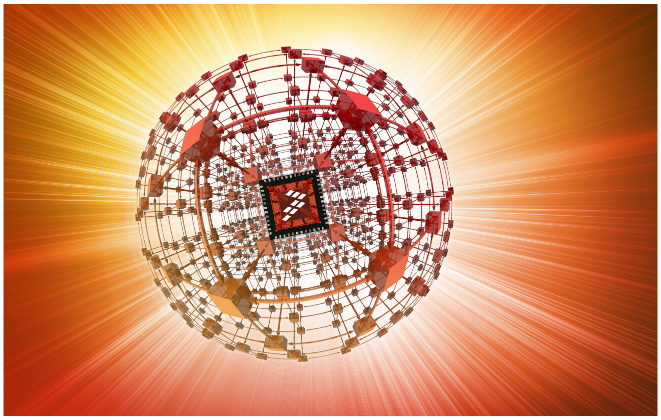
K20 Sub-Family Reference Manual

Supports: MK20DN32VLH5, MK20DX32VLH5, MK20DN64VLH5,
MK20DX64VLH5, MK20DN128VLH5, MK20DX128VLH5,
MK20DN32VMP5, MK20DX32VMP5, MK20DN64VMP5,
MK20DX64VMP5, MK20DN128VMP5, MK20DX128VMP5



Document Number: K20P64M50SF0RM
Rev. 2, Feb 2012

Contents

Section Number **Title** **Page**

Chapter 1
About This Document

1.1 Overview.....45

1.1.1 Purpose.....45

1.1.2 Audience.....45

1.2 Conventions.....45

1.2.1 Numbering systems.....45

1.2.2 Typographic notation.....46

1.2.3 Special terms.....46

Chapter 2
Introduction

2.1 Overview.....47

2.2 Kinetis Portfolio.....47

2.3 K20 Family Introduction.....50

2.4 Module Functional Categories.....50

2.4.1 ARM Cortex-M4 Core Modules.....51

2.4.2 System Modules.....52

2.4.3 Memories and Memory Interfaces.....53

2.4.4 Clocks.....53

2.4.5 Security and Integrity modules.....54

2.4.6 Analog modules.....54

2.4.7 Timer modules.....54

2.4.8 Communication interfaces.....56

2.4.9 Human-machine interfaces.....56

2.5 Orderable part numbers.....57

Chapter 3
Chip Configuration

3.1 Introduction.....59

Section Number	Title	Page

3.2 Core modules....................59

3.2.1 ARM Cortex-M4 Core Configuration....................59

3.2.2 Nested Vectored Interrupt Controller (NVIC) Configuration....................61

3.2.3 Asynchronous Wake-up Interrupt Controller (AWIC) Configuration....................66

3.2.4 JTAG Controller Configuration....................68

3.3 System modules....................68

3.3.1 SIM Configuration....................68

3.3.2 System Mode Controller (SMC) Configuration....................69

3.3.3 PMC Configuration....................69

3.3.4 Low-Leakage Wake-up Unit (LLWU) Configuration....................70

3.3.5 MCM Configuration....................72

3.3.6 Crossbar-Light Switch Configuration....................73

3.3.7 Peripheral Bridge Configuration....................75

3.3.8 DMA request multiplexer configuration....................75

3.3.9 DMA Controller Configuration....................78

3.3.10 External Watchdog Monitor (EWM) Configuration....................79

3.3.11 Watchdog Configuration....................81

3.4 Clock Modules....................82

3.4.1 MCG Configuration....................82

3.4.2 OSC Configuration....................83

3.4.3 RTC OSC configuration....................84

3.5 Memories and Memory Interfaces....................84

3.5.1 Flash Memory Configuration....................84

3.5.2 Flash Memory Controller Configuration....................88

3.5.3 SRAM Configuration....88
3.5.4 System Register File Configuration....91
3.5.5 VBAT Register File Configuration....91
3.5.6 EzPort Configuration....92
3.6 Security....93
3.6.1 CRC Configuration....93
3.7 Analog....94
3.7.1 16-bit SAR ADC Configuration....94
3.7.2 CMP Configuration....98
3.7.3 VREF Configuration....100
3.8 Timers....101
3.8.1 PDB Configuration....101
3.8.2 FlexTimer Configuration....104
3.8.3 PIT Configuration....107
3.8.4 Low-power timer configuration....108
3.8.5 CMT Configuration....110
3.8.6 RTC configuration....111
3.9 Communication interfaces....112
3.9.1 Universal Serial Bus (USB) FS Subsystem....112
3.9.2 SPI configuration....117
3.9.3 I2C Configuration....120
3.9.4 UART Configuration....121
3.9.5 I2S configuration....123
3.10 Human-machine interfaces (HMI)....127
3.10.1 GPIO configuration....127
3.10.2 TSI Configuration....128

Chapter 4
Memory Map

4.1 Introduction....131

Section Number **Title** **Page**

4.2 System memory map....131

4.2.1 Aliased bit-band regions....132

4.3 Flash Memory Map....133

4.3.1 Alternate Non-Volatile IRC User Trim Description....134

4.4 SRAM memory map....134

4.5 Peripheral bridge (AIPS-Lite) memory map....135

4.5.1 Peripheral Bridge 0 (AIPS-Lite 0) Memory Map....135

4.6 Private Peripheral Bus (PPB) memory map....139

Chapter 5
Clock Distribution

5.1 Introduction....141

5.2 Programming model....141

5.3 High-Level device clocking diagram....141

5.4 Clock definitions....142

5.4.1 Device clock summary....143

5.5 Internal clocking requirements....144

5.5.1 Clock divider values after reset....145

5.5.2 VLPR mode clocking....145

5.6 Clock Gating....146

5.7 Module clocks....146

5.7.1 PMC 1-kHz LPO clock....148

5.7.2 WDOG clocking....148

5.7.3 Debug trace clock....148

5.7.4 PORT digital filter clocking....149

5.7.5 LPTMR clocking....149

5.7.6 USB FS OTG Controller clocking....150

5.7.7 UART clocking....150

5.7.8 I2S/SAI clocking....151

5.7.9 TSI clocking....151

Chapter 6
Reset and Boot

6.1 Introduction......153

6.2 Reset......154

  6.2.1 Power-on reset (POR)......154

  6.2.2 System reset sources......154

  6.2.3 MCU Resets......158

  6.2.4 Reset Pin 160

  6.2.5 Debug resets......160

6.3 Boot......161

  6.3.1 Boot sources......161

  6.3.2 Boot options......161

  6.3.3 FOPT boot options......162

  6.3.4 Boot sequence......163

Chapter 7
Power Management

7.1 Introduction......165

7.2 Power modes......165

7.3 Entering and exiting power modes......167

7.4 Power mode transitions......168

7.5 Power modes shutdown sequencing......169

7.6 Module Operation in Low Power Modes......170

7.7 Clock Gating......173

Chapter 8
Security

8.1 Introduction......175

8.2 Flash Security......175

8.3 Security Interactions with other Modules......176

  8.3.1 Security Interactions with EzPort......176

  8.3.2 Security Interactions with Debug......176

Chapter 9
Debug

9.1 Introduction....177

9.1.1 References....179

9.2 The Debug Port....179

9.2.1 JTAG-to-SWD change sequence....180

9.2.2 JTAG-to-cJTAG change sequence....180

9.3 Debug Port Pin Descriptions....181

9.4 System TAP connection....181

9.4.1 IR Codes....181

9.5 JTAG status and control registers....182

9.5.1 MDM-AP Control Register....183

9.5.2 MDM-AP Status Register....185

9.6 Debug Resets....187

9.7 AHB-AP....187

9.8 ITM....188

9.9 Core Trace Connectivity....188

9.10 TPIU....188

9.11 DWT....188

9.12 Debug in Low Power Modes....189

9.12.1 Debug Module State in Low Power Modes....190

9.13 Debug & Security....190

Chapter 10
Signal Multiplexing and Signal Descriptions

10.1 Introduction....191

10.2 Signal Multiplexing Integration....191

10.2.1 Port control and interrupt module features....192

10.2.2 PCRn reset values for port A....192

10.2.3 Clock gating....192

10.2.4 Signal multiplexing constraints....193

10.3 Pinout....193

10.3.1 K20 Signal Multiplexing and Pin Assignments....193

10.3.2 K20 Pinouts....196

10.4 Module Signal Description Tables....198

10.4.1 Core Modules....198

10.4.2 System Modules....199

10.4.3 Clock Modules....199

10.4.4 Memories and Memory Interfaces....200

10.4.5 Analog....200

10.4.6 Communication Interfaces....201

10.4.7 Human-Machine Interfaces (HMI)....203

Chapter 11
Port control and interrupts (PORT)

11.1 Introduction....205

11.1.1 Overview....205

11.1.2 Features....205

11.1.3 Modes of operation....206

11.2 External signal description....206

11.3 Detailed signal description....207

11.4 Memory map and register definition....207

11.4.1 Pin Control Register n (PORT*x*_PCR*n*)....213

11.4.2 Global Pin Control Low Register (PORT*x*_GPCLR)....216

11.4.3 Global Pin Control High Register (PORT*x*_GPCHR)....216

11.4.4 Interrupt Status Flag Register (PORT*x*_ISFR)....217

11.5 Functional description....218

11.5.1 Pin control....218

11.5.2 Global pin control....218

11.5.3 External interrupts....219

Chapter 12
System Integration Module (SIM)

12.1 Introduction......221

12.1.1 Features......221

12.2 Memory map and register definition......222

12.2.1 System Options Register 1 (SIM_SOPT1)......223

12.2.2 SOPT1 Configuration Register (SIM_SOPT1CFG)......225

12.2.3 System Options Register 2 (SIM_SOPT2)......226

12.2.4 System Options Register 4 (SIM_SOPT4)......229

12.2.5 System Options Register 5 (SIM_SOPT5)......232

12.2.6 System Options Register 7 (SIM_SOPT7)......234

12.2.7 System Device Identification Register (SIM_SDID)......235

12.2.8 System Clock Gating Control Register 4 (SIM_SCGC4)......237

12.2.9 System Clock Gating Control Register 5 (SIM_SCGC5)......239

12.2.10 System Clock Gating Control Register 6 (SIM_SCGC6)......241

12.2.11 System Clock Gating Control Register 7 (SIM_SCGC7)......243

12.2.12 System Clock Divider Register 1 (SIM_CLKDIV1)......244

12.2.13 System Clock Divider Register 2 (SIM_CLKDIV2)......246

12.2.14 Flash Configuration Register 1 (SIM_FCFG1)......247

12.2.15 Flash Configuration Register 2 (SIM_FCFG2)......249

12.2.16 Unique Identification Register High (SIM_UIDH)......250

12.2.17 Unique Identification Register Mid-High (SIM_UIDMH)......250

12.2.18 Unique Identification Register Mid Low (SIM_UIDML)......251

12.2.19 Unique Identification Register Low (SIM_UIDL)......251

12.3 Functional description......251

Chapter 13
Reset Control Module (RCM)

13.1 Introduction......253

13.2 Reset memory map and register descriptions....253

13.2.1 System Reset Status Register 0 (RCM_SRS0)....253

13.2.2 System Reset Status Register 1 (RCM_SRS1)....255

13.2.3 Reset Pin Filter Control Register (RCM_RPFC)....257

13.2.4 Reset Pin Filter Width Register (RCM_RPFW)....258

13.2.5 Mode Register (RCM_MR)....259

Chapter 14
System Mode Controller

14.1 Introduction....261

14.2 Modes of operation....261

14.3 Memory map and register descriptions....263

14.3.1 Power Mode Protection Register (SMC_PMPROT)....264

14.3.2 Power Mode Control Register (SMC_PMCTRL)....265

14.3.3 VLLS Control Register (SMC_VLLSCTRL)....267

14.3.4 Power Mode Status Register (SMC_PMSTAT)....268

14.4 Functional Description....268

14.4.1 Power mode transitions....268

14.4.2 Power mode entry/exit sequencing....271

14.4.3 Run modes....274

14.4.4 Wait modes....275

14.4.5 Stop modes....276

14.4.6 Debug in low power modes....279

Chapter 15
Power Management Controller

15.1 Introduction....281

15.2 Features....281

15.3 Low-voltage detect (LVD) system....281

15.3.1 LVD reset operation....282

15.3.2 LVD interrupt operation....282

15.3.3 Low-voltage warning (LVW) interrupt operation......282

15.4 I/O retention......283

15.5 Memory map and register descriptions......283

15.5.1 Low Voltage Detect Status And Control 1 register (PMC_LVDSC1)......283

15.5.2 Low Voltage Detect Status And Control 2 register (PMC_LVDSC2)......285

15.5.3 Regulator Status And Control register (PMC_REGSC)......286

Chapter 16
Low-Leakage Wakeup Unit (LLWU)

16.1 Introduction......289

16.1.1 Features......289

16.1.2 Modes of operation......290

16.1.3 Block diagram......291

16.2 LLWU signal descriptions......292

16.3 Memory map/register definition......293

16.3.1 LLWU Pin Enable 1 register (LLWU_PE1)......294

16.3.2 LLWU Pin Enable 2 register (LLWU_PE2)......295

16.3.3 LLWU Pin Enable 3 register (LLWU_PE3)......296

16.3.4 LLWU Pin Enable 4 register (LLWU_PE4)......297

16.3.5 LLWU Module Enable register (LLWU_ME)......298

16.3.6 LLWU Flag 1 register (LLWU_F1)......300

16.3.7 LLWU Flag 2 register (LLWU_F2)......301

16.3.8 LLWU Flag 3 register (LLWU_F3)......303

16.3.9 LLWU Pin Filter 1 register (LLWU_FILT1)......305

16.3.10 LLWU Pin Filter 2 register (LLWU_FILT2)......306

16.3.11 LLWU Reset Enable register (LLWU_RST)......307

16.4 Functional description......308

16.4.1 LLS mode......308

16.4.2 VLLS modes......308

16.4.3 Initialization......309

Chapter 17
Miscellaneous Control Module (MCM)

17.1 Introduction....311

17.1.1 Features....311

17.2 Memory Map/Register Descriptions....311

17.2.1 Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)....312

17.2.2 Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)....312

17.2.3 Crossbar Switch (AXBS) Control Register (MCM_PLACR)....313

Chapter 18
Crossbar Switch Lite (AXBS-Lite)

18.1 Introduction....315

18.1.1 Features....315

18.2 Memory Map / Register Definition....316

18.3 Functional Description....316

18.3.1 General operation....316

18.3.2 Arbitration....317

18.4 Initialization/application information....318

Chapter 19
Peripheral Bridge (AIPS-Lite)

19.1 Introduction....319

19.1.1 Features....319

19.1.2 General operation....319

19.2 Functional description....320

19.2.1 Access support....320

Chapter 20
Direct Memory Access Multiplexer (DMAMUX)

20.1 Introduction....321

20.1.1 Overview....321

20.1.2 Features....322

20.1.3 Modes of operation....322

20.2 External signal description....323

20.3 Memory map/register definition....323

20.3.1 Channel Configuration register (DMAMUX*x*_CHCFG*n*)....324

20.4 Functional description....325

20.4.1 DMA channels with periodic triggering capability....325

20.4.2 DMA channels with no triggering capability....327

20.4.3 "Always enabled" DMA sources....327

20.5 Initialization/application information....329

20.5.1 Reset....329

20.5.2 Enabling and configuring sources....329

Chapter 21
Direct Memory Access Controller (eDMA)

21.1 Introduction....333

21.1.1 Block diagram....333

21.1.2 Block parts....334

21.1.3 Features....336

21.2 Modes of operation....337

21.3 Memory map/register definition....337

21.3.1 Control Register (DMA_CR)....342

21.3.2 Error Status Register (DMA_ES)....344

21.3.3 Enable Request Register (DMA_ERQ)....346

21.3.4 Enable Error Interrupt Register (DMA_EEI)....347

21.3.5 Clear Enable Error Interrupt Register (DMA_CEEI)....348

21.3.6 Set Enable Error Interrupt Register (DMA_SEEI)....349

21.3.7 Clear Enable Request Register (DMA_CERQ)....350

21.3.8 Set Enable Request Register (DMA_SERQ)....351

21.3.9 Clear DONE Status Bit Register (DMA_CDNE)....352

21.3.10 Set START Bit Register (DMA_SSRT)....353

21.3.11 Clear Error Register (DMA_CERR)....354

21.3.12 Clear Interrupt Request Register (DMA_CINT)....................355

21.3.13 Interrupt Request Register (DMA_INT)....................355

21.3.14 Error Register (DMA_ERR)....................357

21.3.15 Hardware Request Status Register (DMA_HRS)....................358

21.3.16 Channel n Priority Register (DMA_DCHPRI*n*)....................359

21.3.17 TCD Source Address (DMA_TCD*n*_SADDR)....................360

21.3.18 TCD Signed Source Address Offset (DMA_TCD*n*_SOFF)....................360

21.3.19 TCD Transfer Attributes (DMA_TCD*n*_ATTR)....................361

21.3.20 TCD Minor Byte Count (Minor Loop Disabled) (DMA_TCD*n*_NBYTES_MLNO)....................362

21.3.21 TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA_TCD*n*_NBYTES_MLOFFNO)....................363

21.3.22 TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA_TCD*n*_NBYTES_MLOFFYES)....................364

21.3.23 TCD Last Source Address Adjustment (DMA_TCD*n*_SLAST)....................365

21.3.24 TCD Destination Address (DMA_TCD*n*_DADDR)....................365

21.3.25 TCD Signed Destination Address Offset (DMA_TCD*n*_DOFF)....................366

21.3.26 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD*n*_CITER_ELINKYES)....................367

21.3.27 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD*n*_CITER_ELINKNO)....................368

21.3.28 TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD*n*_DLASTSGA)..........369

21.3.29 TCD Control and Status (DMA_TCD*n*_CSR)....................370

21.3.30 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD*n*_BITER_ELINKYES)....................372

21.3.31 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD*n*_BITER_ELINKNO)....................373

21.4 Functional description....................374

21.4.1 eDMA basic data flow....................374

21.4.2 Error reporting and handling....................377

21.4.3 Channel preemption....................379

21.4.4 Performance....................379

21.5 Initialization/application information....................383

21.5.1 eDMA initialization....................383

21.5.2 Programming errors....................385

21.5.3 Arbitration mode considerations....................386

21.5.4 Performing DMA transfers (examples)....................386

21.5.5 Monitoring transfer descriptor status....................390

21.5.6 Channel Linking....................392

21.5.7 Dynamic programming....................393

Chapter 22
External Watchdog Monitor (EWM)

22.1 Introduction....................397

22.1.1 Features....................397

22.1.2 Modes of Operation....................398

22.1.3 Block Diagram....................399

22.2 EWM Signal Descriptions....................400

22.3 Memory Map/Register Definition....................400

22.3.1 Control Register (EWM_CTRL)....................400

22.3.2 Service Register (EWM_SERV)....................401

22.3.3 Compare Low Register (EWM_CMPL)....................402

22.3.4 Compare High Register (EWM_CMPH)....................402

22.4 Functional Description....................403

22.4.1 The EWM_out Signal....................403

22.4.2 The EWM_in Signal....................404

22.4.3 EWM Counter....................404

22.4.4 EWM Compare Registers....................404

22.4.5 EWM Refresh Mechanism....................405

22.4.6 EWM Interrupt....................405

Chapter 23
Watchdog Timer (WDOG)

23.1 Introduction....................407

23.2 Features....................407

23.3 Functional overview....................409

23.3.1 Unlocking and updating the watchdog....................410

23.3.2 Watchdog configuration time (WCT)....................411

23.3.3 Refreshing the watchdog....................412

23.3.4 Windowed mode of operation....................412

23.3.5 Watchdog disabled mode of operation....................412

23.3.6 Low-power modes of operation....................413

23.3.7 Debug modes of operation....................413

23.4 Testing the watchdog....................414

23.4.1 Quick test....................414

23.4.2 Byte test....................415

23.5 Backup reset generator....................416

23.6 Generated resets and interrupts....................416

23.7 Memory map and register definition....................417

23.7.1 Watchdog Status and Control Register High (WDOG_STCTRLH)....................418

23.7.2 Watchdog Status and Control Register Low (WDOG_STCTRLL)....................420

23.7.3 Watchdog Time-out Value Register High (WDOG_TOVALH)....................420

23.7.4 Watchdog Time-out Value Register Low (WDOG_TOVALL)....................421

23.7.5 Watchdog Window Register High (WDOG_WINH)....................421

23.7.6 Watchdog Window Register Low (WDOG_WINL)....................422

23.7.7 Watchdog Refresh register (WDOG_REFRESH)....................422

23.7.8 Watchdog Unlock register (WDOG_UNLOCK)....................423

23.7.9 Watchdog Timer Output Register High (WDOG_TMROUTH)....................423

23.7.10 Watchdog Timer Output Register Low (WDOG_TMROUTL)....................423

23.7.11 Watchdog Reset Count register (WDOG_RSTCNT)....................424

23.7.12 Watchdog Prescaler register (WDOG_PRESC)....424

23.8 Watchdog operation with 8-bit access....424

23.8.1 General guideline....425

23.8.2 Refresh and unlock operations with 8-bit access....425

23.9 Restrictions on watchdog operation....426

Chapter 24
Multipurpose Clock Generator (MCG)

24.1 Introduction....429

24.1.1 Features....429

24.1.2 Modes of Operation....432

24.2 External Signal Description....433

24.3 Memory Map/Register Definition....433

24.3.1 MCG Control 1 Register (MCG_C1)....434

24.3.2 MCG Control 2 Register (MCG_C2)....435

24.3.3 MCG Control 3 Register (MCG_C3)....436

24.3.4 MCG Control 4 Register (MCG_C4)....437

24.3.5 MCG Control 5 Register (MCG_C5)....438

24.3.6 MCG Control 6 Register (MCG_C6)....439

24.3.7 MCG Status Register (MCG_S)....441

24.3.8 MCG Status and Control Register (MCG_SC)....442

24.3.9 MCG Auto Trim Compare Value High Register (MCG_ATCVH)....444

24.3.10 MCG Auto Trim Compare Value Low Register (MCG_ATCVL)....444

24.3.11 MCG Control 7 Register (MCG_C7)....444

24.3.12 MCG Control 8 Register (MCG_C8)....445

24.4 Functional Description....446

24.4.1 MCG mode state diagram....446

24.4.2 Low Power Bit Usage....451

24.4.3 MCG Internal Reference Clocks....451

24.4.4 External Reference Clock....452

Section Number	Title	Page

24.4.5 MCG Fixed frequency clock 452

24.4.6 MCG PLL clock 453

24.4.7 MCG Auto TRIM (ATM)....453

24.5 Initialization / Application information....454

24.5.1 MCG module initialization sequence....454

24.5.2 Using a 32.768 kHz reference....456

24.5.3 MCG mode switching....457

Chapter 25
Oscillator (OSC)

25.1 Introduction....467

25.2 Features and Modes....467

25.3 Block Diagram....468

25.4 OSC Signal Descriptions....468

25.5 External Crystal / Resonator Connections....469

25.6 External Clock Connections....470

25.7 Memory Map/Register Definitions....471

25.7.1 OSC Memory Map/Register Definition....471

25.8 Functional Description....472

25.8.1 OSC Module States....473

25.8.2 OSC Module Modes....474

25.8.3 Counter....476

25.8.4 Reference Clock Pin Requirements....476

25.9 Reset....476

25.10 Low Power Modes Operation....477

25.11 Interrupts....477

Chapter 26
RTC Oscillator

26.1 Introduction....479

26.1.1 Features and Modes....479

26.1.2 Block Diagram....479

26.2 RTC Signal Descriptions....480

26.2.1 EXTAL32 — Oscillator Input....480

26.2.2 XTAL32 — Oscillator Output....480

26.3 External Crystal Connections....481

26.4 Memory Map/Register Descriptions....481

26.5 Functional Description....481

26.6 Reset Overview....482

26.7 Interrupts....482

Chapter 27
Flash Memory Controller (FMC)

27.1 Introduction....483

27.1.1 Overview....483

27.1.2 Features....484

27.2 Modes of operation....484

27.3 External signal description....484

27.4 Memory map and register descriptions....485

27.4.1 Flash Access Protection Register (FMC_PFAPR)....487

27.4.2 Flash Control Register (FMC_PFB0CR)....489

27.4.3 Cache Tag Storage (FMC_TAGVDW0S*n*)....491

27.4.4 Cache Tag Storage (FMC_TAGVDW1S*n*)....492

27.4.5 Cache Tag Storage (FMC_TAGVDW2S*n*)....492

27.4.6 Cache Tag Storage (FMC_TAGVDW3S*n*)....493

27.4.7 Cache Data Storage (FMC_DATAW0S*n*)....494

27.4.8 Cache Data Storage (FMC_DATAW1S*n*)....494

27.4.9 Cache Data Storage (FMC_DATAW2S*n*)....495

27.4.10 Cache Data Storage (FMC_DATAW3S*n*)....495

27.5 Functional description....496

Chapter 28
Flash Memory Module (FTFL)

28.1 Introduction....................497

28.1.1 Features....................498

28.1.2 Block Diagram....................499

28.1.3 Glossary....................500

28.2 External Signal Description....................502

28.3 Memory Map and Registers....................502

28.3.1 Flash Configuration Field Description....................503

28.3.2 Program Flash IFR Map....................503

28.3.3 Data Flash IFR Map....................504

28.3.4 Register Descriptions....................506

28.4 Functional Description....................518

28.4.1 Flash Protection....................518

28.4.2 FlexNVM Description....................520

28.4.3 Interrupts....................524

28.4.4 Flash Operation in Low-Power Modes....................525

28.4.5 Functional Modes of Operation....................525

28.4.6 Flash Reads and Ignored Writes....................525

28.4.7 Read While Write (RWW)....................526

28.4.8 Flash Program and Erase....................526

28.4.9 Flash Command Operations....................526

28.4.10 Margin Read Commands....................533

28.4.11 Flash Command Description....................534

28.4.12 Security....................555

28.4.13 Reset Sequence....................557

Chapter 29
EzPort

29.1 Overview....559

29.1.1 Introduction....559

29.1.2 Features....560

29.1.3 Modes of operation....560

29.2 External signal description....561

29.2.1 EzPort Clock (EZP_CK)....561

29.2.2 EzPort Chip Select (EZP_CS)....561

29.2.3 EzPort Serial Data In (EZP_D)....562

29.2.4 EzPort Serial Data Out (EZP_Q)....562

29.3 Command definition....562

29.3.1 Command descriptions....563

29.4 Flash memory map for EzPort access....569

Chapter 30
Cyclic Redundancy Check (CRC)

30.1 Introduction....571

30.1.1 Features....571

30.1.2 Block diagram....571

30.1.3 Modes of operation....572

30.2 Memory map and register descriptions....572

30.2.1 CRC Data register (CRC_CRC)....573

30.2.2 CRC Polynomial register (CRC_GPOLY)....574

30.2.3 CRC Control register (CRC_CTRL)....575

30.3 Functional description....576

30.3.1 CRC initialization/reinitialization....576

30.3.2 CRC calculations....576

30.3.3 Transpose feature....577

30.3.4 CRC result complement....579

Chapter 31
Analog-to-Digital Converter (ADC)

31.1 Introduction....581

31.1.1 Features....581

31.1.2 Block diagram....582

31.2 ADC Signal Descriptions....583

31.2.1 Analog power (VDDA)....584

31.2.2 Analog ground (VSSA)....584

31.2.3 Voltage reference select....584

31.2.4 Analog channel inputs (ADx)....585

31.2.5 Differential analog channel inputs (DADx)....585

31.3 Register Definition....585

31.3.1 ADC status and control registers 1 (ADC*x*_SC1*n*)....587

31.3.2 ADC configuration register 1 (ADC*x*_CFG1)....590

31.3.3 Configuration register 2 (ADC*x*_CFG2)....592

31.3.4 ADC data result register (ADC*x*_R*n*)....593

31.3.5 Compare value registers (ADC*x*_CV*n*)....594

31.3.6 Status and control register 2 (ADC*x*_SC2)....595

31.3.7 Status and control register 3 (ADC*x*_SC3)....597

31.3.8 ADC offset correction register (ADC*x*_OFS)....598

31.3.9 ADC plus-side gain register (ADC*x*_PG)....599

31.3.10 ADC minus-side gain register (ADC*x*_MG)....599

31.3.11 ADC plus-side general calibration value register (ADC*x*_CLPD)....600

31.3.12 ADC plus-side general calibration value register (ADC*x*_CLPS)....601

31.3.13 ADC plus-side general calibration value register (ADC*x*_CLP4)....601

31.3.14 ADC plus-side general calibration value register (ADC*x*_CLP3)....602

31.3.15 ADC plus-side general calibration value register (ADC*x*_CLP2)....602

31.3.16 ADC plus-side general calibration value register (ADC*x*_CLP1)....603

31.3.17 ADC plus-side general calibration value register (ADC*x*_CLP0)....603

31.3.18 ADC minus-side general calibration value register (ADC*x*_CLMD)....................604

31.3.19 ADC minus-side general calibration value register (ADC*x*_CLMS)....................604

31.3.20 ADC minus-side general calibration value register (ADC*x*_CLM4)....................605

31.3.21 ADC minus-side general calibration value register (ADC*x*_CLM3)....................605

31.3.22 ADC minus-side general calibration value register (ADC*x*_CLM2)....................606

31.3.23 ADC minus-side general calibration value register (ADC*x*_CLM1)....................606

31.3.24 ADC minus-side general calibration value register (ADC*x*_CLM0)....................607

31.4 Functional description....................607

31.4.1 Clock select and divide control....................608

31.4.2 Voltage reference selection....................608

31.4.3 Hardware trigger and channel selects....................609

31.4.4 Conversion control....................610

31.4.5 Automatic compare function....................617

31.4.6 Calibration function....................618

31.4.7 User defined offset function....................620

31.4.8 Temperature sensor....................621

31.4.9 MCU wait mode operation....................621

31.4.10 MCU Normal Stop mode operation....................622

31.4.11 MCU Low Power Stop mode operation....................623

31.5 Initialization information....................623

31.5.1 ADC module initialization example....................624

31.6 Application information....................626

31.6.1 External pins and routing....................626

31.6.2 Sources of error....................628

Chapter 32
Comparator (CMP)

32.1 Introduction....................633

32.2 CMP features....................633

32.3 6-bit DAC key features....................634

32.4 ANMUX key features....635

32.5 CMP, DAC and ANMUX diagram....635

32.6 CMP block diagram....636

32.7 Memory map/register definitions....638

- 32.7.1 CMP Control Register 0 (CMP*x*_CR0)....639
- 32.7.2 CMP Control Register 1 (CMP*x*_CR1)....640
- 32.7.3 CMP Filter Period Register (CMP*x*_FPR)....641
- 32.7.4 CMP Status and Control Register (CMP*x*_SCR)....641
- 32.7.5 DAC Control Register (CMP*x*_DACCR)....643
- 32.7.6 MUX Control Register (CMP*x*_MUXCR)....643

32.8 CMP functional description....644

- 32.8.1 CMP functional modes....644
- 32.8.2 Power modes....654
- 32.8.3 Startup and operation....655
- 32.8.4 Low-pass filter....656

32.9 CMP interrupts....658

32.10 CMP DMA support....658

32.11 Digital-to-analog converter block diagram....659

32.12 DAC functional description....659

- 32.12.1 Voltage reference source select....659

32.13 DAC resets....660

32.14 DAC clocks....660

32.15 DAC interrupts....660

Chapter 33
Voltage Reference (VREFV1)

33.1 Introduction....661

- 33.1.1 Overview....662
- 33.1.2 Features....662
- 33.1.3 Modes of Operation....663

33.1.4 VREF Signal Descriptions....663
33.2 Memory Map and Register Definition....664
33.2.1 VREF Trim Register (VREF_TRM)....664
33.2.2 VREF Status and Control Register (VREF_SC)....665
33.3 Functional Description....666
33.3.1 Voltage Reference Disabled, SC[VREFEN] = 0....667
33.3.2 Voltage Reference Enabled, SC[VREFEN] = 1....667
33.4 Initialization/Application Information....668

Chapter 34
Programmable Delay Block (PDB)

34.1 Introduction....669
34.1.1 Features....669
34.1.2 Implementation....670
34.1.3 Back-to-back Acknowledgement Connections....671
34.1.4 Block Diagram....671
34.1.5 Modes of Operation....673
34.2 PDB Signal Descriptions....673
34.3 Memory Map and Register Definition....673
34.3.1 Status and Control Register (PDB*x*_SC)....674
34.3.2 Modulus Register (PDB*x*_MOD)....677
34.3.3 Counter Register (PDB*x*_CNT)....677
34.3.4 Interrupt Delay Register (PDB*x*_IDLY)....678
34.3.5 Channel n Control Register 1 (PDB*x*_CH*n*C1)....678
34.3.6 Channel n Status Register (PDB*x*_CH*n*S)....679
34.3.7 Channel n Delay 0 Register (PDB*x*_CH*n*DLY0)....680
34.3.8 Channel n Delay 1 Register (PDB*x*_CH*n*DLY1)....680
34.3.9 Pulse-Out n Enable Register (PDB*x*_POEN)....681
34.3.10 Pulse-Out n Delay Register (PDB*x*_PO*n*DLY)....681

34.4 Functional Description....682

34.4.1 PDB Pre-trigger and Trigger Outputs....682

34.4.2 PDB Trigger Input Source Selection....684

34.4.3 Pulse-Out's....684

34.4.4 Updating the Delay Registers....684

34.4.5 Interrupts....686

34.4.6 DMA....686

34.5 Application Information....686

34.5.1 Impact of Using the Prescaler and Multiplication Factor on Timing Resolution....686

Chapter 35
FlexTimer Module (FTM)

35.1 Introduction....687

35.1.1 FlexTimer philosophy....687

35.1.2 Features....688

35.1.3 Modes of operation....689

35.1.4 Block diagram....690

35.2 FTM signal descriptions....692

35.3 Memory map and register definition....692

35.3.1 Memory map....692

35.3.2 Register descriptions....693

35.3.3 Status And Control (FTM*x*_SC)....697

35.3.4 Counter (FTM*x*_CNT)....698

35.3.5 Modulo (FTM*x*_MOD)....699

35.3.6 Channel (n) Status And Control (FTM*x*_C*n*SC)....700

35.3.7 Channel (n) Value (FTM*x*_C*n*V)....703

35.3.8 Counter Initial Value (FTM*x*_CNTIN)....704

35.3.9 Capture And Compare Status (FTM*x*_STATUS)....704

35.3.10 Features Mode Selection (FTM*x*_MODE)....706

35.3.11 Synchronization (FTM*x*_SYNC)....708

35.3.12 Initial State For Channels Output (FTM*x*_OUTINIT)....711

35.3.13 Output Mask (FTM*x*_OUTMASK)....712

35.3.14 Function For Linked Channels (FTM*x*_COMBINE)....714

35.3.15 Deadtime Insertion Control (FTM*x*_DEADTIME)....719

35.3.16 FTM External Trigger (FTM*x*_EXTTRIG)....720

35.3.17 Channels Polarity (FTM*x*_POL)....722

35.3.18 Fault Mode Status (FTM*x*_FMS)....724

35.3.19 Input Capture Filter Control (FTM*x*_FILTER)....726

35.3.20 Fault Control (FTM*x*_FLTCTRL)....727

35.3.21 Quadrature Decoder Control And Status (FTM*x*_QDCTRL)....729

35.3.22 Configuration (FTM*x*_CONF)....731

35.3.23 FTM Fault Input Polarity (FTM*x*_FLTPOL)....733

35.3.24 Synchronization Configuration (FTM*x*_SYNCONF)....734

35.3.25 FTM Inverting Control (FTM*x*_INVCTRL)....736

35.3.26 FTM Software Output Control (FTM*x*_SWOCTRL)....737

35.3.27 FTM PWM Load (FTM*x*_PWMLOAD)....740

35.4 Functional description....741

35.4.1 Clock source....742

35.4.2 Prescaler....743

35.4.3 Counter....743

35.4.4 Input Capture mode....748

35.4.5 Output Compare mode....751

35.4.6 Edge-Aligned PWM (EPWM) mode....752

35.4.7 Center-Aligned PWM (CPWM) mode....754

35.4.8 Combine mode....756

35.4.9 Complementary mode....764

35.4.10 Registers updated from write buffers....765

35.4.11 PWM synchronization....767

35.4.12 Inverting....783

35.4.13 Software output control....784

35.4.14 Deadtime insertion....786

35.4.15 Output mask....789

35.4.16 Fault control....790

35.4.17 Polarity control....793

35.4.18 Initialization....794

35.4.19 Features priority....794

35.4.20 Channel trigger output....795

35.4.21 Initialization trigger....796

35.4.22 Capture Test mode....798

35.4.23 DMA....799

35.4.24 Dual Edge Capture mode....800

35.4.25 Quadrature Decoder mode....807

35.4.26 BDM mode....812

35.4.27 Intermediate load....813

35.4.28 Global time base (GTB)....815

35.5 Reset overview....816

35.6 FTM Interrupts....818

35.6.1 Timer Overflow Interrupt....818

35.6.2 Channel (n) Interrupt....818

35.6.3 Fault Interrupt....818

Chapter 36
Periodic Interrupt Timer (PIT)

36.1 Introduction....819

36.1.1 Block diagram....819

36.1.2 Features....820

36.2 Signal description....820

36.3 Memory map/register description....821

36.3.1 PIT Module Control Register (PIT_MCR)....822

36.3.2 Timer Load Value Register (PIT_LDVAL*n*)....823

36.3.3 Current Timer Value Register (PIT_CVAL*n*)....823

36.3.4 Timer Control Register (PIT_TCTRL*n*)....824

36.3.5 Timer Flag Register (PIT_TFLG*n*)....824

36.4 Functional description....825

36.4.1 General operation....825

36.4.2 Interrupts....826

36.5 Initialization and application information....827

Chapter 37
Low-Power Timer (LPTMR)

37.1 Introduction....829

37.1.1 Features....829

37.1.2 Modes of operation....829

37.2 LPTMR signal descriptions....830

37.2.1 Detailed signal descriptions....830

37.3 Memory map and register definition....831

37.3.1 Low Power Timer Control Status Register (LPTMR*x*_CSR)....831

37.3.2 Low Power Timer Prescale Register (LPTMR*x*_PSR)....833

37.3.3 Low Power Timer Compare Register (LPTMR*x*_CMR)....834

37.3.4 Low Power Timer Counter Register (LPTMR*x*_CNR)....835

37.4 Functional description....835

37.4.1 LPTMR power and reset....835

37.4.2 LPTMR clocking....835

37.4.3 LPTMR prescaler/glitch filter....836

37.4.4 LPTMR compare....837

37.4.5 LPTMR counter....837

37.4.6 LPTMR hardware trigger....838

37.4.7 LPTMR interrupt....838

Chapter 38
Carrier Modulator Transmitter (CMT)

38.1 Introduction.....839
38.2 Features.....839
38.3 Block diagram.....840
38.4 Modes of operation.....841
38.4.1 Wait mode operation.....842
38.4.2 Stop mode operation.....843
38.5 CMT external signal descriptions.....843
38.5.1 CMT_IRO — Infrared Output.....844
38.6 Memory map/register definition.....844
38.6.1 CMT Carrier Generator High Data Register 1 (CMT_CGH1).....845
38.6.2 CMT Carrier Generator Low Data Register 1 (CMT_CGL1).....846
38.6.3 CMT Carrier Generator High Data Register 2 (CMT_CGH2).....846
38.6.4 CMT Carrier Generator Low Data Register 2 (CMT_CGL2).....847
38.6.5 CMT Output Control Register (CMT_OC).....848
38.6.6 CMT Modulator Status and Control Register (CMT_MSC).....849
38.6.7 CMT Modulator Data Register Mark High (CMT_CMD1).....851
38.6.8 CMT Modulator Data Register Mark Low (CMT_CMD2).....851
38.6.9 CMT Modulator Data Register Space High (CMT_CMD3).....852
38.6.10 CMT Modulator Data Register Space Low (CMT_CMD4).....852
38.6.11 CMT Primary Prescaler Register (CMT_PPS).....853
38.6.12 CMT Direct Memory Access Register (CMT_DMA).....854
38.7 Functional description.....854
38.7.1 Clock divider.....854
38.7.2 Carrier generator.....855
38.7.3 Modulator.....857
38.7.4 Extended space operation.....861
38.8 CMT interrupts and DMA.....863

Chapter 39
Real Time Clock (RTC)

39.1 Introduction....865

39.1.1 Features....865

39.1.2 Modes of operation....865

39.1.3 RTC signal descriptions....866

39.2 Register definition....866

39.2.1 RTC Time Seconds Register (RTC_TSR)....867

39.2.2 RTC Time Prescaler Register (RTC_TPR)....868

39.2.3 RTC Time Alarm Register (RTC_TAR)....868

39.2.4 RTC Time Compensation Register (RTC_TCR)....869

39.2.5 RTC Control Register (RTC_CR)....870

39.2.6 RTC Status Register (RTC_SR)....871

39.2.7 RTC Lock Register (RTC_LR)....872

39.2.8 RTC Interrupt Enable Register (RTC_IER)....874

39.2.9 RTC Write Access Register (RTC_WAR)....875

39.2.10 RTC Read Access Register (RTC_RAR)....876

39.3 Functional description....877

39.3.1 Power, clocking and reset....877

39.3.2 Time counter....879

39.3.3 Compensation....879

39.3.4 Time alarm....880

39.3.5 Update mode....880

39.3.6 Register lock....880

39.3.7 Access control....881

39.3.8 Interrupt....881

Chapter 40
Universal Serial Bus OTG Controller (USBOTG)

40.1 Introduction....883

40.1.1 USB....883

40.1.2 USB On-The-Go....884

40.1.3 USB-FS Features....885

40.2 Functional description....885

40.2.1 Data Structures....885

40.3 Programmers interface....886

40.3.1 Buffer Descriptor Table....886

40.3.2 RX vs. TX as a USB target device or USB host....887

40.3.3 Addressing BDT entries....888

40.3.4 Buffer Descriptors (BDs)....888

40.3.5 USB transaction....891

40.4 Memory map/Register definitions....893

40.4.1 Peripheral ID register (USB*x*_PERID)....895

40.4.2 Peripheral ID Complement register (USB*x*_IDCOMP)....896

40.4.3 Peripheral Revision register (USB*x*_REV)....896

40.4.4 Peripheral Additional Info register (USB*x*_ADDINFO)....897

40.4.5 OTG Interrupt Status register (USB*x*_OTGISTAT)....897

40.4.6 OTG Interrupt Control Register (USB*x*_OTGICR)....898

40.4.7 OTG Status register (USB*x*_OTGSTAT)....899

40.4.8 OTG Control Register (USB*x*_OTGCTL)....900

40.4.9 Interrupt Status Register (USB*x*_ISTAT)....901

40.4.10 Interrupt Enable Register (USB*x*_INTEN)....902

40.4.11 Error Interrupt Status Register (USB*x*_ERRSTAT)....903

40.4.12 Error Interrupt Enable Register (USB*x*_ERREN)....904

40.4.13 Status Register (USB*x*_STAT)....906

40.4.14 Control Register (USB*x*_CTL)....907

40.4.15 Address Register (USB*x*_ADDR)....908
40.4.16 BDT Page Register 1 (USB*x*_BDTPAGE1)....909
40.4.17 Frame Number Register Low (USB*x*_FRMNUML)....909
40.4.18 Frame Number Register High (USB*x*_FRMNUMH)....910
40.4.19 Token Register (USB*x*_TOKEN)....910
40.4.20 SOF Threshold Register (USB*x*_SOFTHLD)....911
40.4.21 BDT Page Register 2 (USB*x*_BDTPAGE2)....912
40.4.22 BDT Page Register 3 (USB*x*_BDTPAGE3)....912
40.4.23 Endpoint Control Register (USB*x*_ENDPT*n*)....912
40.4.24 USB Control Register (USB*x*_USBCTRL)....913
40.4.25 USB OTG Observe Register (USB*x*_OBSERVE)....914
40.4.26 USB OTG Control Register (USB*x*_CONTROL)....915
40.4.27 USB Transceiver Control Register 0 (USB*x*_USBTRC0)....915
40.4.28 Frame Adjust Register (USB*x*_USBFRMADJUST)....916
40.5 OTG and Host Mode Operation....917
40.6 Host Mode Operation Examples....917
40.7 On-The-Go Operation....920
40.7.1 OTG Dual Role A Device Operation....921
40.7.2 OTG Dual Role B Device Operation....922

Chapter 41
USB Device Charger Detection Module (USBDCD)

41.1 Preface....925
41.1.1 References....925
41.1.2 Acronyms and abbreviations....925
41.1.3 Glossary....926
41.2 Introduction....926
41.2.1 Block diagram....926
41.2.2 Features....927
41.2.3 Modes of operation....927

41.3 Module signal descriptions....928

41.4 Memory map/Register definition....929

- 41.4.1 Control register (USBDCD_CONTROL)....930
- 41.4.2 Clock register (USBDCD_CLOCK)....932
- 41.4.3 Status register (USBDCD_STATUS)....933
- 41.4.4 TIMER0 register (USBDCD_TIMER0)....934
- 41.4.5 TIMER1 register (USBDCD_TIMER1)....935
- 41.4.6 TIMER2 register (USBDCD_TIMER2)....936

41.5 Functional description....937

- 41.5.1 The charger detection sequence....938
- 41.5.2 Interrupts and events....947
- 41.5.3 Resets....949

41.6 Initialization information....950

41.7 Application information....950

- 41.7.1 External pullups....950
- 41.7.2 Dead or weak battery....950
- 41.7.3 Handling unplug events....951

Chapter 42
USB Voltage Regulator

42.1 Introduction....953

- 42.1.1 Overview....953
- 42.1.2 Features....954
- 42.1.3 Modes of Operation....955

42.2 USB Voltage Regulator Module Signal Descriptions....955

Chapter 43
SPI (DSPI)

43.1 Introduction....957

- 43.1.1 Block Diagram....957
- 43.1.2 Features....958

Section Number	Title	Page

43.1.3 DSPI Configurations....959

43.1.4 Modes of Operation....960

43.2 DSPI signal descriptions....962

43.2.1 PCS0/SS — Peripheral Chip Select/Slave Select....962

43.2.2 PCS1 – PCS3 — Peripheral Chip Selects 1 – 3....962

43.2.3 PCS4 — Peripheral Chip Select 4....963

43.2.4 SIN — Serial Input....963

43.2.5 SOUT — Serial Output....963

43.2.6 SCK — Serial Clock....963

43.3 Memory Map/Register Definition....963

43.3.1 DSPI Module Configuration Register (SPI*x*_MCR)....965

43.3.2 DSPI Transfer Count Register (SPI*x*_TCR)....968

43.3.3 DSPI Clock and Transfer Attributes Register (In Master Mode) (SPI*x*_CTAR*n*)....968

43.3.4 DSPI Clock and Transfer Attributes Register (In Slave Mode) (SPI*x*_CTAR*n*_SLAVE)....973

43.3.5 DSPI Status Register (SPI*x*_SR)....974

43.3.6 DSPI DMA/Interrupt Request Select and Enable Register (SPI*x*_RSER)....977

43.3.7 DSPI PUSH TX FIFO Register In Master Mode (SPI*x*_PUSHR)....979

43.3.8 DSPI PUSH TX FIFO Register In Slave Mode (SPI*x*_PUSHR_SLAVE)....981

43.3.9 DSPI POP RX FIFO Register (SPI*x*_POPR)....981

43.3.10 DSPI Transmit FIFO Registers (SPI*x*_TXFR*n*)....982

43.3.11 DSPI Receive FIFO Registers (SPI*x*_RXFR*n*)....983

43.4 Functional description....983

43.4.1 Start and Stop of DSPI transfers....984

43.4.2 Serial Peripheral Interface (SPI) configuration....985

43.4.3 DSPI baud rate and clock delay generation....988

43.4.4 Transfer formats....991

43.4.5 Continuous Serial Communications Clock....996

43.4.6 Slave Mode Operation Constraints....997

43.4.7 Interrupts/DMA requests....998

43.4.8 Power saving features....1000

43.5 Initialization/application information....1001

43.5.1 How to manage DSPI queues....1001

43.5.2 Switching Master and Slave mode....1002

43.5.3 Initializing DSPI in Master/Slave Modes....1003

43.5.4 Baud rate settings....1003

43.5.5 Delay settings....1004

43.5.6 Calculation of FIFO pointer addresses....1005

Chapter 44
Inter-Integrated Circuit (I2C)

44.1 Introduction....1009

44.1.1 Features....1009

44.1.2 Modes of operation....1010

44.1.3 Block diagram....1010

44.2 I2C signal descriptions....1011

44.3 Memory map and register descriptions....1011

44.3.1 I2C Address Register 1 (I2C*x*_A1)....1012

44.3.2 I2C Frequency Divider register (I2C*x*_F)....1013

44.3.3 I2C Control Register 1 (I2C*x*_C1)....1014

44.3.4 I2C Status register (I2C*x*_S)....1016

44.3.5 I2C Data I/O register (I2C*x*_D)....1017

44.3.6 I2C Control Register 2 (I2C*x*_C2)....1018

44.3.7 I2C Programmable Input Glitch Filter register (I2C*x*_FLT)....1019

44.3.8 I2C Range Address register (I2C*x*_RA)....1020

44.3.9 I2C SMBus Control and Status register (I2C*x*_SMB)....1020

44.3.10 I2C Address Register 2 (I2C*x*_A2)....1022

44.3.11 I2C SCL Low Timeout Register High (I2C*x*_SLTH)....1022

44.3.12 I2C SCL Low Timeout Register Low (I2C*x*_SLTL)....1023

44.4 Functional description....1023

44.4.1 I2C protocol....1023

44.4.2 10-bit address....1028

44.4.3 Address matching....1030

44.4.4 System management bus specification....1030

44.4.5 Resets....1033

44.4.6 Interrupts....1033

44.4.7 Programmable input glitch filter....1035

44.4.8 Address matching wakeup....1036

44.4.9 DMA support....1036

44.5 Initialization/application information....1037

Chapter 45
Universal Asynchronous Receiver/Transmitter (UART)

45.1 Introduction....1041

45.1.1 Features....1041

45.1.2 Modes of operation....1043

45.2 UART signal descriptions....1044

45.2.1 Detailed signal descriptions....1045

45.3 Memory map and registers....1046

45.3.1 UART Baud Rate Registers: High (UART*x*_BDH)....1053

45.3.2 UART Baud Rate Registers: Low (UART*x*_BDL)....1054

45.3.3 UART Control Register 1 (UART*x*_C1)....1055

45.3.4 UART Control Register 2 (UART*x*_C2)....1057

45.3.5 UART Status Register 1 (UART*x*_S1)....1058

45.3.6 UART Status Register 2 (UART*x*_S2)....1061

45.3.7 UART Control Register 3 (UART*x*_C3)....1063

45.3.8 UART Data Register (UART*x*_D)....1065

45.3.9 UART Match Address Registers 1 (UART*x*_MA1)....1066

45.3.10 UART Match Address Registers 2 (UART*x*_MA2)....1067

45.3.11 UART Control Register 4 (UART*x*_C4)....1067

45.3.12 UART Control Register 5 (UART*x*_C5)....1068

45.3.13 UART Extended Data Register (UART*x*_ED)....1069

45.3.14 UART Modem Register (UART*x*_MODEM)....1070

45.3.15 UART Infrared Register (UART*x*_IR)....1071

45.3.16 UART FIFO Parameters (UART*x*_PFIFO)....1072

45.3.17 UART FIFO Control Register (UART*x*_CFIFO)....1074

45.3.18 UART FIFO Status Register (UART*x*_SFIFO)....1075

45.3.19 UART FIFO Transmit Watermark (UART*x*_TWFIFO)....1076

45.3.20 UART FIFO Transmit Count (UART*x*_TCFIFO)....1077

45.3.21 UART FIFO Receive Watermark (UART*x*_RWFIFO)....1077

45.3.22 UART FIFO Receive Count (UART*x*_RCFIFO)....1078

45.3.23 UART 7816 Control Register (UART*x*_C7816)....1078

45.3.24 UART 7816 Interrupt Enable Register (UART*x*_IE7816)....1080

45.3.25 UART 7816 Interrupt Status Register (UART*x*_IS7816)....1081

45.3.26 UART 7816 Wait Parameter Register (UART*x*_WP7816T0)....1083

45.3.27 UART 7816 Wait Parameter Register (UART*x*_WP7816T1)....1083

45.3.28 UART 7816 Wait N Register (UART*x*_WN7816)....1084

45.3.29 UART 7816 Wait FD Register (UART*x*_WF7816)....1084

45.3.30 UART 7816 Error Threshold Register (UART*x*_ET7816)....1085

45.3.31 UART 7816 Transmit Length Register (UART*x*_TL7816)....1086

45.3.32 UART CEA709.1-B Control Register 6 (UART*x*_C6)....1086

45.3.33 UART CEA709.1-B Packet Cycle Time Counter High (UART*x*_PCTH)....1087

45.3.34 UART CEA709.1-B Packet Cycle Time Counter Low (UART*x*_PCTL)....1088

45.3.35 UART CEA709.1-B Beta1 Timer (UART*x*_B1T)....1088

45.3.36 UART CEA709.1-B Secondary Delay Timer High (UART*x*_SDTH)....1089

45.3.37 UART CEA709.1-B Secondary Delay Timer Low (UART*x*_SDTL)....1089

45.3.38 UART CEA709.1-B Preamble (UART*x*_PRE)....1090

45.3.39 UART CEA709.1-B Transmit Packet Length (UART*x*_TPL)....1090

45.3.40 UART CEA709.1-B Interrupt Enable Register (UART*x*_IE)....1091

45.3.41 UART CEA709.1-B WBASE (UART*x*_WB)....1092

45.3.42 UART CEA709.1-B Status Register (UART*x*_S3)....1092

45.3.43 UART CEA709.1-B Status Register (UART*x*_S4)....1094

45.3.44 UART CEA709.1-B Received Packet Length (UART*x*_RPL)....1095

45.3.45 UART CEA709.1-B Received Preamble Length (UART*x*_RPREL)....1095

45.3.46 UART CEA709.1-B Collision Pulse Width (UART*x*_CPW)....1096

45.3.47 UART CEA709.1-B Receive Indeterminate Time (UART*x*_RIDT)....1096

45.3.48 UART CEA709.1-B Transmit Indeterminate Time (UART*x*_TIDT)....1097

45.4 Functional description....1097

45.4.1 CEA709.1-B....1097

45.4.2 Transmitter....1107

45.4.3 Receiver....1113

45.4.4 Baud rate generation....1122

45.4.5 Data format (non ISO-7816)....1124

45.4.6 Single-wire operation....1127

45.4.7 Loop operation....1128

45.4.8 ISO-7816/smartcard support....1128

45.4.9 Infrared interface....1133

45.5 Reset....1134

45.6 System level interrupt sources....1134

45.6.1 RXEDGIF description....1135

45.7 DMA operation....1136

45.8 Application information....1136

45.8.1 Transmit/receive data buffer operation....1136

45.8.2 ISO-7816 initialization sequence....1137

45.8.3 Initialization sequence (non ISO-7816)....1139

45.8.4 Overrun (OR) flag implications....1140

45.8.5 Overrun NACK considerations....1141

45.8.6 Match address registers....1142

45.8.7 Modem feature....1142

45.8.8 IrDA minimum pulse width....1143

45.8.9 Clearing 7816 wait timer (WT, BWT, CWT) interrupts....1143

45.8.10 Legacy and reverse compatibility considerations....1144

Chapter 46
Synchronous Audio Interface (SAI)

46.1 Introduction....1145

46.1.1 Features....1145

46.1.2 Block diagram....1145

46.1.3 Modes of operation....1146

46.2 External signals....1147

46.3 Memory map and register definition....1148

46.3.1 SAI Transmit Control Register (I2S*x*_TCSR)....1149

46.3.2 SAI Transmit Configuration 1 Register (I2S*x*_TCR1)....1152

46.3.3 SAI Transmit Configuration 2 Register (I2S*x*_TCR2)....1152

46.3.4 SAI Transmit Configuration 3 Register (I2S*x*_TCR3)....1154

46.3.5 SAI Transmit Configuration 4 Register (I2S*x*_TCR4)....1155

46.3.6 SAI Transmit Configuration 5 Register (I2S*x*_TCR5)....1156

46.3.7 SAI Transmit Data Register (I2S*x*_TDR*n*)....1157

46.3.8 SAI Transmit FIFO Register (I2S*x*_TFR*n*)....1157

46.3.9 SAI Transmit Mask Register (I2S*x*_TMR)....1158

46.3.10 SAI Receive Control Register (I2S*x*_RCSR)....1158

46.3.11 SAI Receive Configuration 1 Register (I2S*x*_RCR1)....1161

46.3.12 SAI Receive Configuration 2 Register (I2S*x*_RCR2)....1162

46.3.13 SAI Receive Configuration 3 Register (I2S*x*_RCR3)....1163

46.3.14 SAI Receive Configuration 4 Register (I2S*x*_RCR4)....1164

46.3.15 SAI Receive Configuration 5 Register (I2S*x*_RCR5)....1165

46.3.16 SAI Receive Data Register (I2S*x*_RDR*n*)....1166

Section Number Title Page

46.3.17 SAI Receive FIFO Register (I2S*x*_RFR*n*)....1166

46.3.18 SAI Receive Mask Register (I2S*x*_RMR)....1167

46.3.19 SAI MCLK Control Register (I2S*x*_MCR)....1168

46.3.20 SAI MCLK Divide Register (I2S*x*_MDR)....1169

46.4 Functional description....1169

46.4.1 SAI clocking....1169

46.4.2 SAI resets....1171

46.4.3 Synchronous modes....1171

46.4.4 Frame sync configuration....1172

46.4.5 Data FIFO....1173

46.4.6 Word mask register....1174

46.4.7 Interrupts and DMA requests....1175

Chapter 47
General-Purpose Input/Output (GPIO)

47.1 Introduction....1177

47.1.1 Features....1177

47.1.2 Modes of operation....1177

47.1.3 GPIO signal descriptions....1178

47.2 Memory map and register definition....1179

47.2.1 Port Data Output Register (GPIO*x*_PDOR)....1182

47.2.2 Port Set Output Register (GPIO*x*_PSOR)....1182

47.2.3 Port Clear Output Register (GPIO*x*_PCOR)....1183

47.2.4 Port Toggle Output Register (GPIO*x*_PTOR)....1183

47.2.5 Port Data Input Register (GPIO*x*_PDIR)....1184

47.2.6 Port Data Direction Register (GPIO*x*_PDDR)....1185

47.3 Functional description....1185

47.3.1 General-purpose input....1185

47.3.2 General-purpose output....1185

Chapter 48
Touch sense input (TSI)

48.1 Introduction....1187

48.2 Features....1187

48.3 Overview....1188

48.3.1 Electrode capacitance measurement unit....1188

48.3.2 Electrode scan unit....1189

48.3.3 Touch detection unit....1190

48.4 Modes of operation....1190

48.4.1 TSI disabled mode....1191

48.4.2 TSI active mode....1191

48.4.3 TSI low power mode....1191

48.4.4 Block diagram....1192

48.5 TSI signal descriptions....1193

48.5.1 TSI_IN[15:0]....1193

48.6 Memory map and register definition....1193

48.6.1 General Control and Status Register (TSI*x*_GENCS)....1195

48.6.2 SCAN Control Register (TSI*x*_SCANC)....1198

48.6.3 Pin Enable Register (TSI*x*_PEN)....1200

48.6.4 Wake-Up Channel Counter Register (TSI*x*_WUCNTR)....1202

48.6.5 Counter Register (TSI*x*_CNTR*n*)....1203

48.6.6 Low Power Channel Threshold Register (TSI*x*_THRESHOLD)....1203

48.7 Functional descriptions....1204

48.7.1 Capacitance measurement....1204

48.7.2 TSI measurement result....1207

48.7.3 Electrode scan unit....1208

48.7.4 Touch detection unit....1211

48.8 Application information....1212

48.8.1 TSI module sensitivity....1212

48.9 TSI module initialization....1212

48.9.1 Initialization Sequence....1213

Chapter 49
JTAG Controller (JTAGC)

49.1 Introduction....1215

49.1.1 Block diagram....1215

49.1.2 Features....1216

49.1.3 Modes of operation....1216

49.2 External signal description....1218

49.2.1 TCK—Test clock input....1218

49.2.2 TDI—Test data input....1218

49.2.3 TDO—Test data output....1218

49.2.4 TMS—Test mode select....1218

49.3 Register description....1219

49.3.1 Instruction register....1219

49.3.2 Bypass register....1219

49.3.3 Device identification register....1219

49.3.4 Boundary scan register....1220

49.4 Functional description....1221

49.4.1 JTAGC reset configuration....1221

49.4.2 IEEE 1149.1-2001 (JTAG) Test Access Port....1221

49.4.3 TAP controller state machine....1221

49.4.4 JTAGC block instructions....1223

49.4.5 Boundary scan....1226

49.5 Initialization/Application information....1226

Chapter 1
About This Document

1.1 Overview

1.1.1 Purpose

This document describes the features, architecture, and programming model of the Freescale K20 microcontroller.

1.1.2 Audience

This document is primarily for system architects and software application developers who are using or considering using the K20 microcontroller in a system.

1.2 Conventions

1.2.1 Numbering systems

The following suffixes identify different numbering systems:

This suffix	Identifies a
b	Binary number. For example, the binary equivalent of the number 5 is written 101b. In some cases, binary numbers are shown with the prefix *0b*.
d	Decimal number. Decimal numbers are followed by this suffix only when the possibility of confusion exists. In general, decimal numbers are shown without a suffix.
h	Hexadecimal number. For example, the hexadecimal equivalent of the number 60 is written 3Ch. In some cases, hexadecimal numbers are shown with the prefix *0x*.

1.2.2 Typographic notation

The following typographic notation is used throughout this document:

Example	Description
placeholder, x	Items in italics are placeholders for information that you provide. Italicized text is also used for the titles of publications and for emphasis. Plain lowercase letters are also used as placeholders for single letters and numbers.
`code`	Fixed-width type indicates text that must be typed exactly as shown. It is used for instruction mnemonics, directives, symbols, subcommands, parameters, and operators. Fixed-width type is also used for example code. Instruction mnemonics and directives in text and tables are shown in all caps; for example, BSR.
SR[SCM]	A mnemonic in brackets represents a named field in a register. This example refers to the Scaling Mode (SCM) field in the Status Register (SR).
REVNO[6:4], XAD[7:0]	Numbers in brackets and separated by a colon represent either: • A subset of a register's named field For example, REVNO[6:4] refers to bits 6–4 that are part of the COREREV field that occupies bits 6–0 of the REVNO register. • A continuous range of individual signals of a bus For example, XAD[7:0] refers to signals 7–0 of the XAD bus.

1.2.3 Special terms

The following terms have special meanings:

Term	Meaning
asserted	Refers to the state of a signal as follows: • An active-high signal is asserted when high (1). • An active-low signal is asserted when low (0).
deasserted	Refers to the state of a signal as follows: • An active-high signal is deasserted when low (0). • An active-low signal is deasserted when high (1). In some cases, deasserted signals are described as *negated*.
reserved	Refers to a memory space, register, or field that is either reserved for future use or for which, when written to, the module or chip behavior is unpredictable.

Chapter 2
Introduction

2.1 Overview

This chapter provides an overview of the Kinetis portfolio and K20 family of products. It also presents high-level descriptions of the modules available on the devices covered by this document.

2.2 Kinetis Portfolio

Kinetis is the most scalable portfolio of low power, mixed-signal ARM®Cortex™-M4 MCUs in the industry. Phase 1 of the portfolio consists of five MCU families with over 200 pin-, peripheral- and software-compatible devices. Each family offers excellent performance, memory and feature scalability with common peripherals, memory maps, and packages providing easy migration both within and between families.

Kinetis MCUs are built from Freescale's innovative 90nm Thin Film Storage (TFS) flash technology with unique FlexMemory. Kinetis MCU families combine the latest low-power innovations and high performance, high precision mixed-signal capability with a broad range of connectivity, human-machine interface, and safety & security peripherals. Kinetis MCUs are supported by a market-leading enablement bundle from Freescale and numerous ARM 3rd party ecosystem partners.

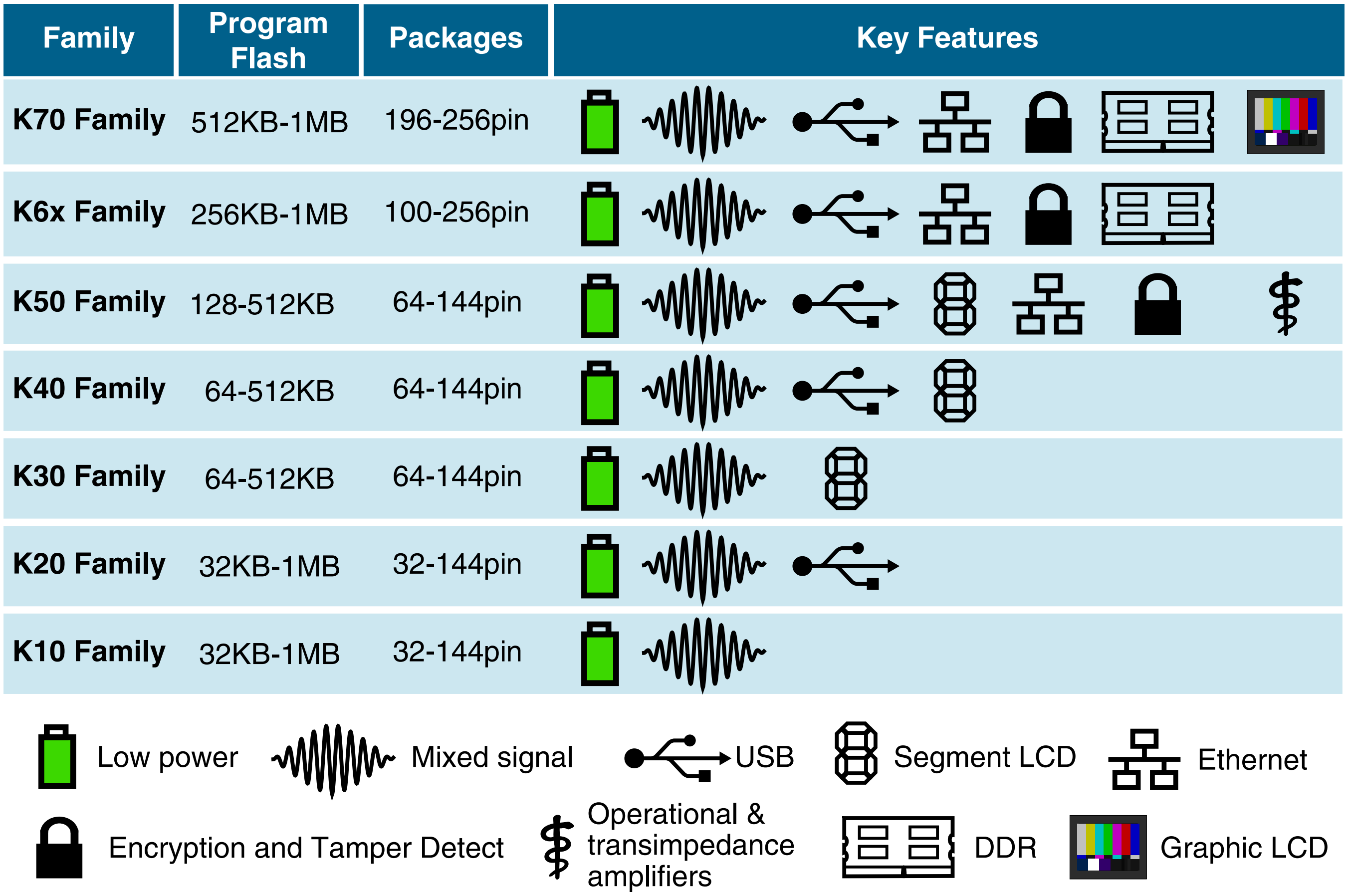

Family
Program Flash
Packages
Key Features
K70 Family
512KB-1MB
196-256pin
K6x Family
256KB-1MB
100-256pin
K50 Family
128-512KB
64-144pin
K40 Family
64-512KB
64-144pin
K30 Family
64-512KB
64-144pin
K20 Family
32KB-1MB
32-144pin
K10 Family
32KB-1MB
32-144pin
Low power
Mixed signal
USB
Segment LCD
Ethernet
Encryption and Tamper Detect
Operational & transimpedance amplifiers
DDR
Graphic LCD

Figure 2-1. Kinetis MCU portfolio

All Kinetis families include a powerful array of analog, communication and timing and control peripherals with the level of feature integration increasing with flash memory size and the number of inputs/outputs. Some of the available features in Kinetis families include:

- Core:
  - ARM Cortex-M4 Core delivering 1.25 DMIPS/MHz with DSP instructions (floating-point unit available on certain Kinetis families)
  - Up to 32-channel DMA for peripheral and memory servicing with minimal CPU intervention
  - Broad range of performance levels rated at maximum CPU frequencies of 50 MHz, 72 MHz, 100 MHz, 120 MHz, and 150 MHz
- Ultra-low power:
  - Multiple low power operating modes for optimizing peripheral activity and wake-up times for extended battery life.
  - Low–leakage wake-up unit, low power timer, and low power RTC for additional low power flexibility
  - Industry-leading fast wake-up times
- Memory:

  - Scalable memory footprints from 32 KB flash / 8 KB RAM to 1 MB flash / 128 KB RAM. Independent flash banks enable concurrent code execution and firmware updates
  - Optional 16 KB cache memory for optimizing bus bandwidth and flash execution performance. Offered on K10, K20, and K60 family devices with CPU performance of up to 150 MHz.
  - FlexMemory with up to 512 KB FlexNVM and up to 16 KB FlexRAM. FlexNVM can be partitioned to support additional program flash memory (ex. bootloader), data flash (ex. storage for large tables), or EEPROM backup. FlexRAM supports EEPROM byte-write/byte-erase operations and dictates the maximum EEPROM size.
  - EEPROM endurance capable of exceeding 10 million cycles
  - EEPROM erase/write times an order of magnitude faster than traditional EEPROM
  - Multi-function external bus interface capable of interfacing to external memories, gate-array logic
- Mixed-signal analog:
  - Fast, high precision 16-bit ADCs, 12-bit DACs, high speed comparators and an internal voltage reference. Powerful signal conditioning, conversion and analysis capability with reduced system cost
- Human Machine Interface (HMI):
  - Capacitive Touch Sensing Interface with full low-power support and minimal current adder when enabled
- Connectivity and Communications:
  - UARTs with ISO7816, CEA709.1-B (LON), and IrDA support, I2S, CAN, I2C and DSPI
  - Full-speed USB OTG controller with on-chip transceiver
- Reliability, Safety and Security:
  - Hardware cyclic redundancy check engine for validating memory contents/ communication data and increased system reliability
  - Independent-clocked computer operating properly (COP) for protection against code runaway in fail-safe applications
  - External watchdog monitor
- Timing and Control:
  - Powerful FlexTimers which support general purpose, PWM, and motor control functions
  - Carrier Modulator Transmitter for IR waveform generation
  - Programmable Interrupt Timer for RTOS task scheduler time base or trigger source for ADC conversion and programmable delay block
- System:
  - 5 V tolerant GPIO with pin interrupt functionality

- Wide operating voltage range from 1.71 V to 3.6 V with flash programmable down to 1.71 V with fully functional flash and analog peripherals
- Ambient operating temperature ranges from -40 °C to 105 °C

2.3 K20 Family Introduction

The K20 MCU family is pin, peripheral and software compatible with the K10 MCU family and adds full and high-speed USB 2.0 On-The-Go with device charger detect capability. Devices start from 32 KB of flash in 5 x 5 mm 32QFN packages extending up to 1 MB in a 144MAPBGA package with a rich suite of analog, communication, timing and control peripherals. High memory density K20 family devices include a single precision floating point unit and NAND flash controller.

2.4 Module Functional Categories

The modules on this device are grouped into functional categories. The following sections describe the modules assigned to each category in more detail.

Table 2-1. Module functional categories

Module category	Description
ARM Cortex-M4 core	• 32-bit MCU core from ARM's Cortex-M class adding DSP instructions, 1.25 DMIPS/MHz, based on ARMv7 architecture
System	• System integration module • Power management and mode controllers • Multiple power modes available based on run, wait, stop, and power-down modes • Low-leakage wakeup unit • Miscellaneous control module • Crossbar switch • Peripheral bridge • Direct memory access (DMA) controller with multiplexer to increase available DMA requests • External watchdog monitor • Watchdog

Memories	• Internal memories include:
	    • Program flash memory
	    • FlexMemory
	        • FlexNVM
	        • FlexRAM
	    • SRAM
	• Serial programming interface: EzPort

Table continues on the next page...

Table 2-1. Module functional categories (continued)

Module category	Description
Clocks	• Multiple clock generation options available from internally- and externally-generated clocks • System oscillator to provide clock source for the MCU • RTC oscillator to provide clock source for the RTC
Security	• Cyclic Redundancy Check module for error detection
Analog	• High speed analog-to-digital converter with integrated programmable gain amplifier • Comparator • Internal voltage reference
Timers	• Programmable delay block • FlexTimers • Periodic interrupt timer • Low power timer • Carrier modulator transmitter • Independent real time clock
Communications	• USB OTG controller with built-in FS/LS transceiver • USB device charger detect • USB voltage regulator • Serial peripheral interface • Inter-integrated circuit (I^2C) • UART • Integrated interchip sound (I^2S)
Human-Machine Interfaces (HMI)	• General purpose input/output controller • Capacitive touch sense input interface enabled in hardware

2.4.1 ARM Cortex-M4 Core Modules

The following core modules are available on this device.

Table 2-2. Core modules

Module	Description
ARM Cortex-M4	The ARM Cortex-M4 is the newest member of the Cortex M Series of processors targeting microcontroller cores focused on very cost sensitive, deterministic, interrupt driven environments. The Cortex M4 processor is based on the ARMv7 Architecture and Thumb®-2 ISA and is upward compatible with the Cortex M3, Cortex M1, and Cortex M0 architectures. Cortex M4 improvements include an ARMv7 Thumb-2 DSP (ported from the ARMv7-A/R profile architectures) providing 32-bit instructions with SIMD (single instruction multiple data) DSP style multiply-accumulates and saturating arithmetic.

Table continues on the next page...

Table 2-2. Core modules (continued)

Module	Description
NVIC	The ARMv7-M exception model and nested-vectored interrupt controller (NVIC) implement a relocatable vector table supporting many external interrupts, a single non-maskable interrupt (NMI), and priority levels. The NVIC replaces shadow registers with equivalent system and simplified programmability. The NVIC contains the address of the function to execute for a particular handler. The address is fetched via the instruction port allowing parallel register stacking and look-up. The first sixteen entries are allocated to ARM internal sources with the others mapping to MCU-defined interrupts.
AWIC	The primary function of the Asynchronous Wake-up Interrupt Controller (AWIC) is to detect asynchronous wake-up events in stop modes and signal to clock control logic to resume system clocking. After clock restart, the NVIC observes the pending interrupt and performs the normal interrupt or event processing.
Debug interfaces	Most of this device's debug is based on the ARM CoreSight™ architecture. Four debug interfaces are supported: • IEEE 1149.1 JTAG • IEEE 1149.7 JTAG (cJTAG) • Serial Wire Debug (SWD) • ARM Real-Time Trace Interface

2.4.2 System Modules

The following system modules are available on this device.

Table 2-3. System modules

Module	Description
System integration module (SIM)	The SIM includes integration logic and several module configuration settings.
System mode controller	The SMC provides control and protection on entry and exit to each power mode, control for the Power management controller (PMC), and reset entry and exit for the complete MCU.
Power management controller (PMC)	The PMC provides the user with multiple power options. Ten different modes are supported that allow the user to optimize power consumption for the level of functionality needed. Includes power-on-reset (POR) and integrated low voltage detect (LVD) with reset (brownout) capability and selectable LVD trip points.
Low-leakage wakeup unit (LLWU)	The LLWU module allows the device to wake from low leakage power modes (LLS and VLLS) through various internal peripheral and external pin sources.
Miscellaneous control module (MCM)	The MCM includes integration logic and embedded trace buffer details.
Crossbar switch (XBS)	The XBS connects bus masters and bus slaves, allowing all bus masters to access different bus slaves simultaneously and providing arbitration among the bus masters when they access the same slave.
Peripheral bridges	The peripheral bridge converts the crossbar switch interface to an interface to access a majority of peripherals on the device.
DMA multiplexer (DMAMUX)	The DMA multiplexer selects from many DMA requests down to a smaller number for the DMA controller.

Table continues on the next page...

Table 2-3. System modules (continued)

Module	Description
Direct memory access (DMA) controller	The DMA controller provides programmable channels with transfer control descriptors for data movement via dual-address transfers for 8-, 16-, 32- and 128-bit data values.
External watchdog monitor (EWM)	The EWM is a redundant mechanism to the software watchdog module that monitors both internal and external system operation for fail conditions.
Software watchdog (WDOG)	The WDOG monitors internal system operation and forces a reset in case of failure. It can run from an independent 1 KHz low power oscillator with a programmable refresh window to detect deviations in program flow or system frequency.

2.4.3 Memories and Memory Interfaces

The following memories and memory interfaces are available on this device.

Table 2-4. Memories and memory interfaces

Module	Description

Flash memory	• Program flash memory — non-volatile flash memory that can execute program code
	• FlexMemory — encompasses the following memory types:
	  • FlexNVM — Non-volatile flash memory that can execute program code, store data, or backup EEPROM data
	  • FlexRAM — RAM memory that can be used as traditional RAM or as high-endurance EEPROM storage, and also accelerates flash programming

Flash memory controller	Manages the interface between the device and the on-chip flash memory.
SRAM	Internal system RAM. Partial SRAM kept powered in VLLS2 low leakage mode.
SRAM controller	Manages simultaneous accesses to system RAM by multiple master peripherals and core.
System register file	32-byte register file that is accessible during all power modes and is powered by VDD.
VBAT register file	32-byte register file that is accessible during all power modes and is powered by VBAT.
Serial programming interface (EzPort)	Same serial interface as, and subset of, the command set used by industry-standard SPI flash memories. Provides the ability to read, erase, and program flash memory and reset command to boot the system after flash programming.

2.4.4 Clocks

The following clock modules are available on this device.

Table 2-5. Clock modules

Module	Description
Multi-clock generator (MCG)	The MCG provides several clock sources for the MCU that include: • Phase-locked loop (PLL) — Voltage-controlled oscillator (VCO) • Frequency-locked loop (FLL) — Digitally-controlled oscillator (DCO) • Internal reference clocks — Can be used as a clock source for other on-chip peripherals
System oscillator	The system oscillator, in conjunction with an external crystal or resonator, generates a reference clock for the MCU.
Real-time clock oscillator	The RTC oscillator has an independent power supply and supports a 32 kHz crystal oscillator to feed the RTC clock. Optionally, the RTC oscillator can replace the system oscillator as the main oscillator source.

2.4.5 Security and Integrity modules

The following security and integrity modules are available on this device:

Table 2-6. Security and integrity modules

Module	Description
Cyclic Redundancy Check (CRC)	Hardware CRC generator circuit using 16/32-bit shift register. Error detection for all single, double, odd, and most multi-bit errors, programmable initial seed value, and optional feature to transpose input data and CRC result via transpose register.

2.4.6 Analog modules

The following analog modules are available on this device:

Table 2-7. Analog modules

Module	Description
16-bit analog-to-digital converters (ADC)	16-bit successive-approximation ADC
Analog comparators	Compares two analog input voltages across the full range of the supply voltage.
6-bit digital-to-analog converters (DAC)	64-tap resistor ladder network which provides a selectable voltage reference for applications where voltage reference is needed.
Voltage reference (VREF)	Supplies an accurate voltage output that is trimmable in 0.5 mV steps. The VREF can be used in medical applications, such as glucose meters, to provide a reference voltage to biosensors or as a reference to analog peripherals, such as the ADC, DAC, or CMP.

2.4.7 Timer modules

The following timer modules are available on this device:

Table 2-8. Timer modules

Module	Description
Programmable delay block (PDB)	• 16-bit resolution • 3-bit prescaler • Positive transition of trigger event signal initiates the counter • Supports two triggered delay output signals, each with an independently-controlled delay from the trigger event • Outputs can be OR'd together to schedule two conversions from one input trigger event and can schedule precise edge placement for a pulsed output. This feature is used to generate the control signal for the CMP windowing feature and output to a package pin if needed for applications, such as critical conductive mode power factor correction. • Continuous-pulse output or single-shot mode supported, each output is independently enabled, with possible trigger events • Supports bypass mode • Supports DMA
Flexible timer modules (FTM)	• Selectable FTM source clock, programmable prescaler • 16-bit counter supporting free-running or initial/final value, and counting is up or up-down • Input capture, output compare, and edge-aligned and center-aligned PWM modes • Operation of FTM channels as pairs with equal outputs, pairs with complimentary outputs, or independent channels with independent outputs • Deadtime insertion is available for each complementary pair • Generation of hardware triggers • Software control of PWM outputs • Up to 4 fault inputs for global fault control • Configurable channel polarity • Programmable interrupt on input capture, reference compare, overflowed counter, or detected fault condition • Quadrature decoder with input filters, relative position counting, and interrupt on position count or capture of position count on external event • DMA support for FTM events
Periodic interrupt timers (PIT)	• Four general purpose interrupt timers • Interrupt timers for triggering ADC conversions • 32-bit counter resolution • Clocked by system clock frequency • DMA support
Low-power timer (LPTimer)	• Selectable clock for prescaler/glitch filter of 1 kHz (internal LPO), 32.768 kHz (external crystal), or internal reference clock • Configurable Glitch Filter or Prescaler with 16-bit counter • 16-bit time or pulse counter with compare • Interrupt generated on Timer Compare • Hardware trigger generated on Timer Compare

Table continues on the next page...

Table 2-8. Timer modules (continued)

Module	Description

Carrier modulator timer (CMT)	• Four CMT modes of operation:
	  • Time with independent control of high and low times
	  • Baseband
	  • Frequency shift key (FSK)
	  • Direct software control of CMT_IRO pin
	• Extended space operation in time, baseband, and FSK modes
	• Selectable input clock divider
	• Interrupt on end of cycle with the ability to disable CMT_IRO pin and use as timer interrupt
	• DMA support

Real-time clock (RTC)	• Independent power supply, POR, and 32 kHz Crystal Oscillator • 32-bit seconds counter with 32-bit Alarm • 16-bit Prescaler with compensation that can correct errors between 0.12 ppm and 3906 ppm

2.4.8 Communication interfaces

The following communication interfaces are available on this device:

Table 2-9. Communication modules

Module	Description
USB OTG (low-/full-speed)	USB 2.0 compliant module with support for host, device, and On-The-Go modes. Includes an on-chip transceiver for full and low speeds.
USB Device Charger Detect (USBDCD)	The USBDCD monitors the USB data lines to detect a smart charger meeting the USB Battery Charging Specification Rev1.1. This information allows the MCU to better manage the battery charging IC in a portable device.
USB voltage regulator	Up to 5 V regulator input typically provided by USB VBUS power with 3.3 V regulated output that powers on-chip USB subsystem, capable of sourcing 120 mA to external board components.
Serial peripheral interface (SPI)	Synchronous serial bus for communication to an external device
Inter-integrated circuit (I2C)	Allows communication between a number of devices. Also supports the System Management Bus (SMBus) Specification, version 2.
Universal asynchronous receiver/ transmitters (UART)	Asynchronous serial bus communication interface with programmable 8- or 9-bit data format and support of CEA709.1-B (LON), ISO 7816 smart card interface
I2S	The I^2S is a full-duplex, serial port that allows the chip to communicate with a variety of serial devices, such as standard codecs, digital signal processors (DSPs), microprocessors, peripherals, and audio codecs that implement the inter-IC sound bus (I^2S) and the Intel® AC97 standards

2.4.9 Human-machine interfaces

The following human-machine interfaces (HMI) are available on this device:

Table 2-10. HMI modules

Module	Description
General purpose input/output (GPIO)	All general purpose input or output (GPIO) pins are capable of interrupt and DMA request generation. All GPIO pins have 5 V tolerance.
Capacitive touch sense input (TSI)	Contains up to 16 channel inputs for capacitive touch sensing applications. Operation is available in low-power modes via interrupts.

2.5 Orderable part numbers

The following table summarizes the part numbers of the devices covered by this document.

Table 2-11. Orderable part numbers summary

Freescale part number	CPU frequency	Pin count	Package	Total flash memory	Program flash	EEPROM	SRAM	GPIO
MK20DN32VLH5	50 MHz	64	LQFP	32 KB	32 KB	—	8 KB	40
MK20DX32VLH5	50 MHz	64	LQFP	64 KB	32 KB	2 KB	8 KB	40
MK20DN64VLH5	50 MHz	64	LQFP	64 KB	64 KB	—	16 KB	40
MK20DX64VLH5	50 MHz	64	LQFP	96 KB	64 KB	2 KB	16 KB	40
MK20DN128VLH5	50 MHz	64	LQFP	128 KB	128 KB	—	16 KB	40
MK20DX128VLH5	50 MHz	64	LQFP	160 KB	128 KB	2 KB	16 KB	40
MK20DN32VMP5	50 MHz	64	BGA	32 KB	32 KB	—	8 KB	40
MK20DX32VMP5	50 MHz	64	BGA	64 KB	32 KB	2 KB	8 KB	40
MK20DN64VMP5	50 MHz	64	BGA	64 KB	64 KB	—	16 KB	40
MK20DX64VMP5	50 MHz	64	BGA	96 KB	64 KB	2 KB	16 KB	40
MK20DN128VMP5	50 MHz	64	BGA	128 KB	128 KB	—	16 KB	40
MK20DX128VMP5	50 MHz	64	BGA	160 KB	128 KB	2 KB	16 KB	40

Chapter 3
Chip Configuration

3.1 Introduction

This chapter provides details on the individual modules of the microcontroller. It includes:

- module block diagrams showing immediate connections within the device,
- specific module-to-module interactions not necessarily discussed in the individual module chapters, and
- links for more information.

3.2 Core modules

3.2.1 ARM Cortex-M4 Core Configuration

This section summarizes how the module has been configured in the chip. Full documentation for this module is provided by ARM and can be found at http://www.arm.com.

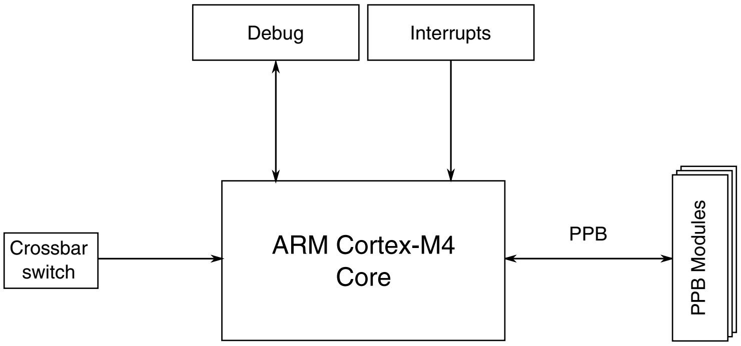


Figure 3-1. Core configuration

Table 3-1. Reference links to related information

Topic	Related module	Reference
Full description	ARM Cortex-M4 core, r0p1	http://www.arm.com
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
System/instruction/data bus module	Crossbar switch	Crossbar switch
Debug	IEEE 1149.1 JTAG Serial Wire Debug (SWD) ARM Real-Time Trace Interface	Debug
Interrupts	Nested Vectored Interrupt Controller (NVIC)	NVIC
Private Peripheral Bus (PPB) module	Miscellaneous Control Module (MCM)	MCM

3.2.1.1 Buses, interconnects, and interfaces

The ARM Cortex-M4 core has four buses as described in the following table.

Bus name	Description
Instruction code (ICODE) bus	The ICODE and DCODE buses are muxed. This muxed bus is called the CODE bus and is connected to the crossbar switch via a single master port.
Data code (DCODE) bus	
System bus	The system bus is connected to a separate master port on the crossbar.
Private peripheral (PPB) bus	The PPB provides access to these modules: • ARM modules such as the NVIC, ITM, DWT, FBP, and ROM table • Freescale Miscellaneous Control Module (MCM)

3.2.1.2 System Tick Timer

The System Tick Timer's clock source is always the core clock, FCLK. This results in the following:

- The CLKSOURCE bit in SysTick Control and Status register is always set to select the core clock.
- Because the timing reference (FCLK) is a variable frequency, the TENMS bit in the SysTick Calibration Value Register is always zero.
- The NOREF bit in SysTick Calibration Value Register is always set, implying that FCLK is the only available source of reference timing.

3.2.1.3 Debug facilities

This device has extensive debug capabilities including run control and tracing capabilities. The standard ARM debug port that supports JTAG and SWD interfaces. Also the cJTAG interface is supported on this device.

3.2.1.4 Core privilege levels

The ARM documentation uses different terms than this document to distinguish between privilege levels.

If you see this term...	it also means this term...
Privileged	Supervisor
Unprivileged or user	User

3.2.2 Nested Vectored Interrupt Controller (NVIC) Configuration

This section summarizes how the module has been configured in the chip. Full documentation for this module is provided by ARM and can be found at http://www.arm.com.

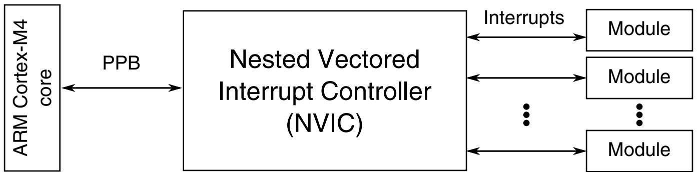


Figure 3-2. NVIC configuration

Table 3-2. Reference links to related information

Topic	Related module	Reference
Full description	Nested Vectored Interrupt Controller (NVIC)	http://www.arm.com
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Private Peripheral Bus (PPB)	ARM Cortex-M4 core	ARM Cortex-M4 core

3.2.2.1 Interrupt priority levels

This device supports 16 priority levels for interrupts. Therefore, in the NVIC each source in the IPR registers contains 4 bits. For example, IPR0 is shown below:

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IRQ3				0	0	0	0	IRQ2				0	0	0	0	IRQ1				0	0	0	0	IRQ0				0	0	0	0
W																																

3.2.2.2 Non-maskable interrupt

The non-maskable interrupt request to the NVIC is controlled by the external $\overline{\text{NMI}}$ signal. The pin the $\overline{\text{NMI}}$ signal is multiplexed on, must be configured for the $\overline{\text{NMI}}$ function to generate the non-maskable interrupt request.

3.2.2.3 Interrupt channel assignments

The interrupt source assignments are defined in the following table.

- Vector number — the value stored on the stack when an interrupt is serviced.
- IRQ number — non-core interrupt source count, which is the vector number minus 16.

The IRQ number is used within ARM's NVIC documentation.

Table 3-4. Interrupt vector assignments

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
ARM Core System Handler Vectors						
0x0000_0000	0	–	–	–	ARM core	Initial Stack Pointer
0x0000_0004	1	–	–	–	ARM core	Initial Program Counter
0x0000_0008	2	–	–	–	ARM core	Non-maskable Interrupt (NMI)
0x0000_000C	3	–	–	–	ARM core	Hard Fault
0x0000_0010	4	–	–	–	ARM core	MemManage Fault
0x0000_0014	5	–	–	–	ARM core	Bus Fault
0x0000_0018	6	–	–	–	ARM core	Usage Fault
0x0000_001C	7	–	–	–	—	—
0x0000_0020	8	–	–	–	—	—
0x0000_0024	9	–	–	–	—	—
0x0000_0028	10	–	–	–	—	—
0x0000_002C	11	–	–	–	ARM core	Supervisor call (SVCall)
0x0000_0030	12	–	–	–	ARM core	Debug Monitor
0x0000_0034	13	–	–	–	—	—
0x0000_0038	14	–	–	–	ARM core	Pendable request for system service (PendableSrvReq)
0x0000_003C	15	–	–	–	ARM core	System tick timer (SysTick)
Non-Core Vectors						
0x0000_0040	16	0	0	0	DMA	DMA channel 0 transfer complete
0x0000_0044	17	1	0	0	DMA	DMA channel 1 transfer complete
0x0000_0048	18	2	0	0	DMA	DMA channel 2 transfer complete
0x0000_004C	19	3	0	0	DMA	DMA channel 3 transfer complete
0x0000_0050	20	4	0	1	DMA	DMA error interrupt channel
0x0000_0054	21	5	0	1	DMA	–
0x0000_0058	22	6	0	1	Flash memory	Command complete
0x0000_005C	23	7	0	1	Flash memory	Read collision
0x0000_0060	24	8	0	2	Mode Controller	Low-voltage detect, low-voltage warning

Table continues on the next page...

Table 3-4. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_0064	25	9	0	2	LLWU	Low Leakage Wakeup **NOTE:** The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit stop mode on an LLS recovery .
0x0000_0068	26	10	0	2	WDOG	Both EWM and WDOG interrupt sources set this IRQ
0x0000_006C	27	11	0	2	I^{2}C0	–
0x0000_0070	28	12	0	3	SPI0	Single interrupt vector for all sources
0x0000_0074	29	13	0	3	I^{2}S0	Transmit
0x0000_0078	30	14	0	3	I^{2}S1	Receive
0x0000_007C	31	15	0	3	UART0	Single interrupt vector for CEA709.1-B (LON) status sources
0x0000_0080	32	16	0	4	UART0	Single interrupt vector for UART status sources
0x0000_0084	33	17	0	4	UART0	Single interrupt vector for UART error sources
0x0000_0088	34	18	0	4	UART1	Single interrupt vector for UART status sources
0x0000_008C	35	19	0	4	UART1	Single interrupt vector for UART error sources
0x0000_0090	36	20	0	5	UART2	Single interrupt vector for UART status sources
0x0000_0094	37	21	0	5	UART2	Single interrupt vector for UART error sources
0x0000_0098	38	22	0	5	ADC0	
0x0000_009C	39	23	0	5	CMP0	—
0x0000_00A0	40	24	0	6	CMP1	—
0x0000_00A4	41	25	0	6	FTM0	—
0x0000_00A8	42	26	0	6	FTM1	—
0x0000_00AC	43	27	0	6	CMT	—
0x0000_00B0	44	28	0	7	RTC	Alarm interrupt
0x0000_00B4	45	29	0	7	RTC	Seconds interrupt
0x0000_00B8	46	30	0	7	PIT	Channel 0

Table continues on the next page...

Table 3-4. Interrupt vector assignments (continued)

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_00BC	47	31	0	7	PIT	Channel 1
0x0000_00C0	48	32	1	8	PIT	Channel 2
0x0000_00C4	49	33	1	8	PIT	Channel 3
0x0000_00C8	50	34	1	8	PDB	—
0x0000_00CC	51	35	1	8	USB OTG	—
0x0000_00D0	52	36	1	9	USB Charger Detect	—
0x0000_00D4	53	37	1	9	TSI	—
0x0000_00D8	54	38	1	9	MCG	—
0x0000_00DC	55	39	1	9	Low Power Timer	—
0x0000_00E0	56	40	1	10	Port control module	Pin detect (Port A)
0x0000_00E4	57	41	1	10	Port control module	Pin detect (Port B)
0x0000_00E8	58	42	1	10	Port control module	Pin detect (Port C)
0x0000_00EC	59	43	1	10	Port control module	Pin detect (Port D)
0x0000_00F0	60	44	1	11	Port control module	Pin detect (Port E)
0x0000_00F4	61	45	1	11	Software initiated interrupt[4]	—

1. Indicates the NVIC's interrupt source number.
2. Indicates the NVIC's ISER, ICER, ISPR, ICPR, and IABR register number used for this IRQ. The equation to calculate this value is: IRQ div 32
3. Indicates the NVIC's IPR register number used for this IRQ. The equation to calculate this value is: IRQ div 4
4. This interrupt can only be pended or cleared via the NVIC registers.

3.2.2.3.1 Determining the bitfield and register location for configuring a particular interrupt

Suppose you need to configure the low-power timer (LPTMR) interrupt. The following table is an excerpt of the LPTMR row from Interrupt channel assignments.

Table 3-5. LPTMR interrupt vector assignment

Address	Vector	IRQ[1]	NVIC non-IPR register number [2]	NVIC IPR register number [3]	Source module	Source description
0x0000_00DC	55	39	1	9	Low Power Timer	—

1. Indicates the NVIC's interrupt source number.
2. Indicates the NVIC's ISER, ICER, ISPR, ICPR, and IABR register number used for this IRQ. The equation to calculate this value is: IRQ div 32
3. Indicates the NVIC's IPR register number used for this IRQ. The equation to calculate this value is: IRQ div 4

- The NVIC registers you would use to configure the interrupt are:
    - NVICISER1
    - NVICICER1
    - NVICISPR1
    - NVICICPR1
    - NVICIABR1
    - NVICIPR9
- To determine the particular IRQ's bitfield location within these particular registers:
    - NVICISER1, NVICICER1, NVICISPR1, NVICICPR1, NVICIABR1 bit location = IRQ mod 32 = 7
    - NVICIPR9 bitfield starting location = 8 * (IRQ mod 4) + 4 = 28

      Since the NVICIPR bitfields are 4-bit wide (16 priority levels), the NVICIPR9 bitfield range is 28-31

Therefore, the following bitfield locations are used to configure the LPTMR interrupts:

- NVICISER1[7]
- NVICICER1[7]
- NVICISPR1[7]
- NVICICPR1[7]
- NVICIABR1[7]
- NVICIPR9[31:28]

3.2.3 Asynchronous Wake-up Interrupt Controller (AWIC) Configuration

This section summarizes how the module has been configured in the chip. Full documentation for this module is provided by ARM and can be found at http://www.arm.com.

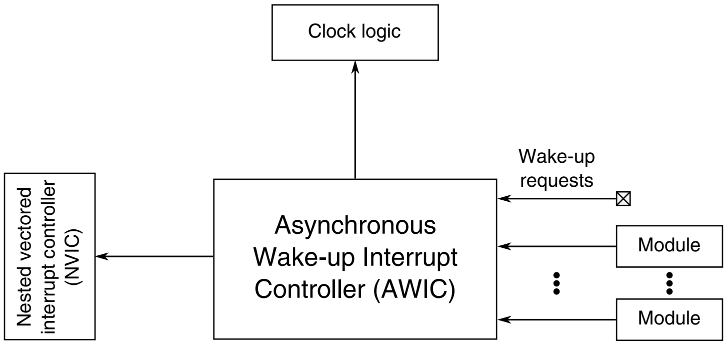

Clock logic
Wake-up requests
Nested vectored interrupt controller (NVIC)
Asynchronous Wake-up Interrupt Controller (AWIC)
Module
Module

Figure 3-3. Asynchronous Wake-up Interrupt Controller configuration

Table 3-6. Reference links to related information

Topic	Related module	Reference
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
	Nested Vectored Interrupt Controller (NVIC)	NVIC
Wake-up requests		AWIC wake-up sources

3.2.3.1 Wake-up sources

The device uses the following internal and external inputs to the AWIC module.

Table 3-7. AWIC Stop and VLPS Wake-up Sources

Wake-up source	Description
Available system resets	$\overline{\text{RESET}}$ pin and WDOG when LPO is its clock source, and JTAG
Low-voltage detect	Mode Controller
Low-voltage warning	Mode Controller
Pin interrupts	Port Control Module - Any enabled pin interrupt is capable of waking the system
ADCx	The ADC is functional when using internal clock source
CMPx	Since no system clocks are available, functionality is limited
I^{2}C	Address match wakeup
UART	Active edge on RXD
USB	Wakeup
LPTMR	Functional in Stop/VLPS modes
RTC	Functional in Stop/VLPS modes
I2S	Functional when using an external bit clock or external master clock
TSI	

3.2.4 JTAG Controller Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

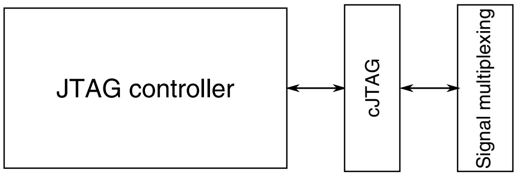

JTAG controller
cJTAG
Signal multiplexing

Figure 3-4. JTAGC Controller configuration

Table 3-8. Reference links to related information

Topic	Related module	Reference
Full description	JTAGC	JTAGC
Signal multiplexing	Port control	Signal multiplexing

3.3 System modules

3.3.1 SIM Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

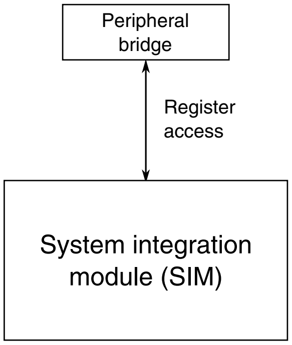

Peripheral bridge
Register access
System integration module (SIM)

Figure 3-5. SIM configuration

Table 3-9. Reference links to related information

Topic	Related module	Reference
Full description	SIM	SIM

Table continues on the next page...

Table 3-9. Reference links to related information (continued)

Topic	Related module	Reference
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management

3.3.2 System Mode Controller (SMC) Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

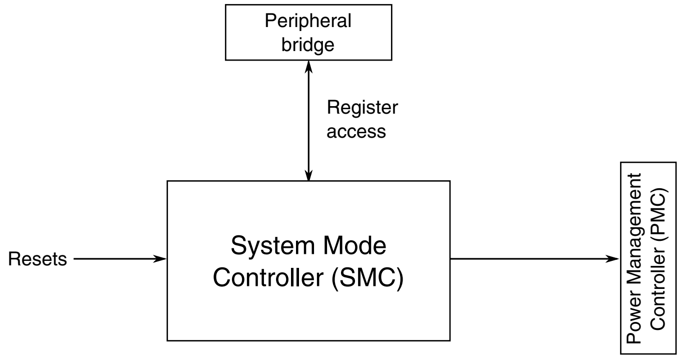

Peripheral bridge
Register access
Resets
System Mode Controller (SMC)
Power Management Controller (PMC)

Figure 3-6. System Mode Controller configuration

Table 3-10. Reference links to related information

Topic	Related module	Reference
Full description	System Mode Controller (SMC)	SMC
System memory map		System memory map
Power management		Power management
	Power management controller (PMC)	PMC
	Low-Leakage Wakeup Unit (LLWU)	LLWU
	Reset Control Module (RCM)	Reset

3.3.3 PMC Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

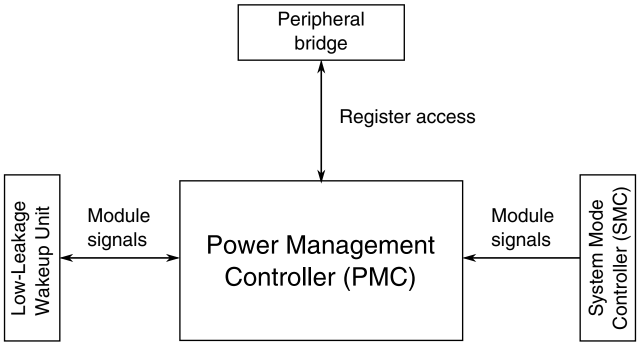

Peripheral bridge
Register access
Low-Leakage Wakeup Unit
Module signals
Power Management Controller (PMC)
Module signals
System Mode Controller (SMC)

Figure 3-7. PMC configuration

Table 3-11. Reference links to related information

Topic	Related module	Reference
Full description	PMC	PMC
System memory map		System memory map
Power management		Power management
Full description	System Mode Controller (SMC)	System Mode Controller
	Low-Leakage Wakeup Unit (LLWU)	LLWU
	Reset Control Module (RCM)	Reset

3.3.4 Low-Leakage Wake-up Unit (LLWU) Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

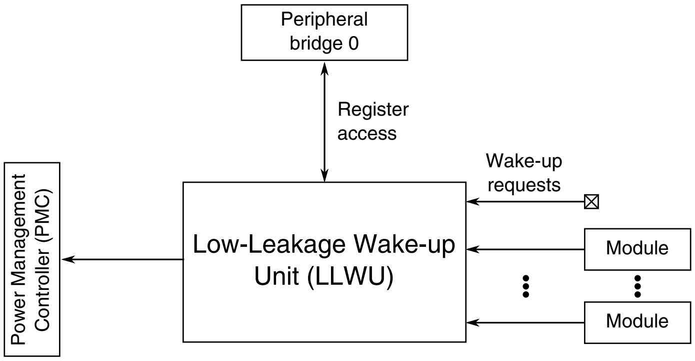

Peripheral bridge 0
Register access
Power Management Controller (PMC)
Low-Leakage Wake-up Unit (LLWU)
Wake-up requests
Module
Module

Figure 3-8. Low-Leakage Wake-up Unit configuration

Table 3-12. Reference links to related information

Topic	Related module	Reference
Full description	LLWU	LLWU
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management chapter
	Power Management Controller (PMC)	Power Management Controller (PMC)
	Mode Controller	Mode Controller
Wake-up requests		LLWU wake-up sources

3.3.4.1 Wake-up Sources

This chip uses the following internal peripheral and external pin inputs as wakeup sources to the LLWU module:

- LLWU_P0-15 are external pin inputs. Any digital function multiplexed on the pin can be selected as the wakeup source. See the chip's signal multiplexing table for the digital signal options.
- LLWU_M0IF-M7IF are connections to the internal peripheral interrupt flags.

NOTE

$\overline{\text{RESET}}$ is also a wakeup source, depending on the bit setting in the LLWU_RST register. On devices where $\overline{\text{RESET}}$ is not a dedicated pin, it must also be enabled in the explicit port mux control.

Table 3-13. Wakeup sources for LLWU inputs

Input	Wakeup source		Input	Wakeup source
LLWU_P0	PTE1/LLWU_P0 pin		LLWU_P12	PTD0/LLWU_P12 pin
LLWU_P1	PTE2/LLWU_P1 pin		LLWU_P13	PTD2/LLWU_P13 pin
LLWU_P2	PTE4/LLWU_P2 pin		LLWU_P14	PTD4/LLWU_P14 pin
LLWU_P3	PTA4/LLWU_P3 pin[1]		LLWU_P15	PTD6/LLWU_P15 pin
LLWU_P4	PTA13/LLWU_P4 pin		LLWU_M0IF	LPTMR[2]
LLWU_P5	PTB0/LLWU_P5 pin		LLWU_M1IF	CMP0[2]
LLWU_P6	PTC1/LLWU_P6 pin		LLWU_M2IF	CMP1[2]
LLWU_P7	PTC3/LLWU_P7 pin		LLWU_M3IF	Reserved
LLWU_P8	PTC4/LLWU_P8 pin		LLWU_M4IF	TSI[2]
LLWU_P9	PTC5/LLWU_P9 pin		LLWU_M5IF	RTC Alarm[2]
LLWU_P10	PTC6/LLWU_P10 pin		LLWU_M6IF	Reserved
LLWU_P11	PTC11/LLWU_P11 pin		LLWU_M7IF	RTC Seconds[2]

1. The $\overline{\text{EZP\_CS}}$ signal is checked only on *Chip Reset not VLLS*, so a VLLS wakeup via a non-reset source does not cause EzPort mode entry. If NMI was enabled on entry to LLS/VLLS, asserting the NMI pin generates an NMI interrupt on exit from the low power mode. NMI can also be disabled via the FOPT[NMI_DIS] bit.
2. Requires the peripheral and the peripheral interrupt to be enabled. The LLWU's WUME bit enables the internal module flag as a wakeup input. After wakeup, the flags are cleared based on the peripheral clearing mechanism.

3.3.5 MCM Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

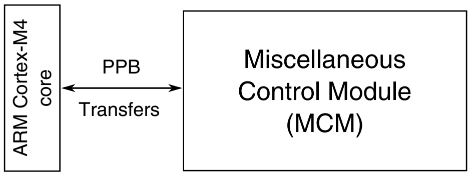


Figure 3-9. MCM configuration

Table 3-14. Reference links to related information

Topic	Related module	Reference
Full description	Miscellaneous control module (MCM)	MCM
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management

Table continues on the next page...

Table 3-14. Reference links to related information (continued)

Topic	Related module	Reference
Transfers Private Peripheral Bus (PPB)	ARM Cortex-M4 core	ARM Cortex-M4 core

3.3.6 Crossbar-Light Switch Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

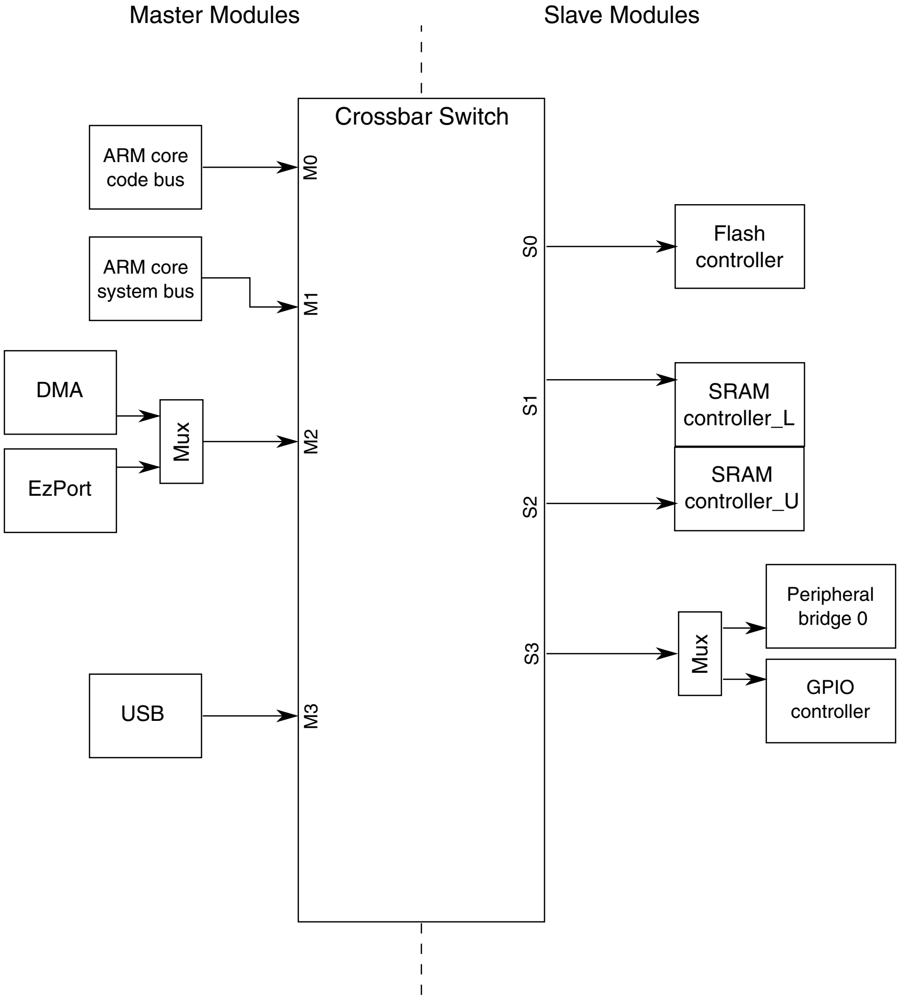

Master Modules
Slave Modules
Crossbar Switch
ARM core code bus
ARM core system bus
DMA
EzPort
Mux
USB
M0
M1
M2
M3
S0
S1
S2
S3
Flash controller
SRAM controller_L
SRAM controller_U
Mux
Peripheral bridge 0
GPIO controller

Figure 3-10. Crossbar-Light switch integration

Table 3-15. Reference links to related information

Topic	Related module	Reference
Full description	Crossbar switch	Crossbar Switch
System memory map		System memory map
Clocking		Clock Distribution
Crossbar switch master	ARM Cortex-M4 core	ARM Cortex-M4 core
Crossbar switch master	DMA controller	DMA controller
Crossbar switch master	EzPort	EzPort
Crossbar switch master	USB FS/LS	USB FS/LS
Crossbar switch slave	Flash	Flash
Crossbar switch slaves	SRAM controllers	SRAM configuration
Crossbar switch slave	Peripheral bridges	Peripheral bridge
Crossbar switch slave	GPIO controller	GPIO controller

3.3.6.1 Crossbar-Light Switch Master Assignments

The masters connected to the crossbar switch are assigned as follows:

Master module	Master port number
ARM core code bus	0
ARM core system bus	1
DMA/EzPort	2
USB OTG	3

NOTE

The DMA and EzPort share a master port. Since these modules never operate at the same time, no configuration or arbitration explanations are necessary.

3.3.6.2 Crossbar-Light Switch Slave Assignments

The slaves connected to the crossbar switch are assigned as follows:

Slave module	Slave port number	Protected by MPU?
Flash memory controller	0	No
SRAM controllers	1,2	No
Peripheral bridge 0/GPIO[1]	3	No. Protection built into bridge.

[1] See System memory map for access restrictions.

3.3.6.3 PRS register reset values

The AXBS_PRS*n* registers reset to 0000_3210h.

3.3.7 Peripheral Bridge Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

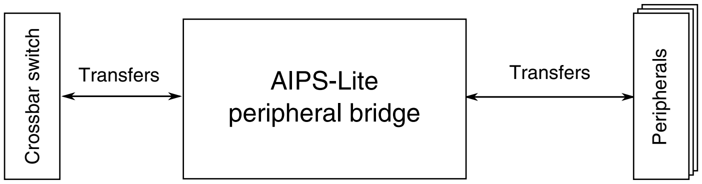


Figure 3-11. Peripheral bridge configuration

Table 3-16. Reference links to related information

Topic	Related module	Reference
Full description	Peripheral bridge (AIPS-Lite)	Peripheral bridge (AIPS-Lite)
System memory map		System memory map
Clocking		Clock Distribution
Crossbar switch	Crossbar switch	Crossbar switch

3.3.7.1 Number of peripheral bridges

This device contains one peripheral bridge.

3.3.7.2 Memory maps

The peripheral bridges are used to access the registers of most of the modules on this device. See AIPS0 Memory Map for the memory slot assignment for each module.

3.3.8 DMA request multiplexer configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

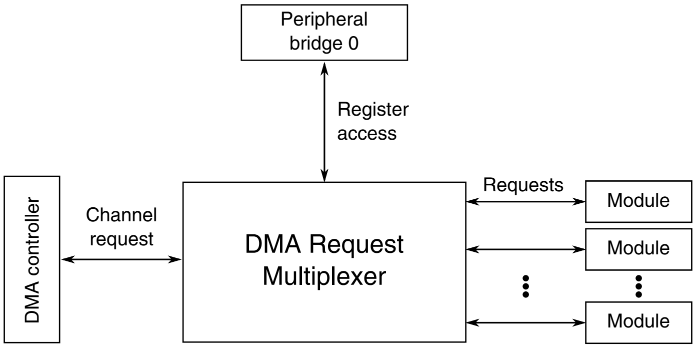

Peripheral bridge 0
Register access
DMA controller
Channel request
DMA Request Multiplexer
Requests
Module
Module
Module

Figure 3-12. DMA request multiplexer configuration

Table 3-17. Reference links to related information

Topic	Related module	Reference
Full description	DMA request multiplexer	DMA Mux
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Channel request	DMA controller	DMA Controller
Requests		DMA request sources

3.3.8.1 DMA MUX request sources

This device includes a DMA request mux that allows up to 63 DMA request signals to be mapped to any of the 4 DMA channels.

Because of the mux there is not a hard correlation between any of the DMA request sources and a specific DMA channel.

Table 3-18. DMA request sources - MUX 0

Source number	Source module	Source description
0	—	Channel disabled[1]
1	Reserved	Not used
2	UART0	Receive
3	UART0	Transmit
4	UART1	Receive
5	UART1	Transmit
6	UART2	Receive
7	UART2	Transmit

Table continues on the next page...

Table 3-18. DMA request sources - MUX 0 (continued)

Source number	Source module	Source description
8	Reserved	—
9	Reserved	—
10	Reserved	—
11	Reserved	—
12	Reserved	—
13	Reserved	—
14	I^2S0	Receive
15	I^2S0	Transmit
16	SPI0	Receive
17	SPI0	Transmit
18	Reserved	—
19	Reserved	—
20	Reserved	—
21	Reserved	—
22	I^2C0	—
23	Reserved	—
24	FTM0	Channel 0
25	FTM0	Channel 1
26	FTM0	Channel 2
27	FTM0	Channel 3
28	FTM0	Channel 4
29	FTM0	Channel 5
30	FTM0	Channel 6
31	FTM0	Channel 7
32	FTM1	Channel 0
33	FTM1	Channel 1
34	Reserved	—
35	Reserved	—
36	Reserved	—
37	Reserved	—
38	Reserved	—
39	Reserved	—
40	ADC0	—
41	Reserved	—

Table continues on the next page...

Table 3-18. DMA request sources - MUX 0 (continued)

Source number	Source module	Source description
42	CMP0	—
43	CMP1	—
44	Reserved	—
45	Reserved	—
46	Reserved	—
47	CMT	—
48	PDB	—
49	Port control module	Port A
50	Port control module	Port B
51	Port control module	Port C
52	Port control module	Port D
53	Port control module	Port E
54	DMA MUX	Always enabled
55	DMA MUX	Always enabled
56	DMA MUX	Always enabled
57	DMA MUX	Always enabled
58	DMA MUX	Always enabled
59	DMA MUX	Always enabled
60	DMA MUX	Always enabled
61	DMA MUX	Always enabled
62	DMA MUX	Always enabled
63	DMA MUX	Always enabled

1. Configuring a DMA channel to select source 0 or any of the reserved sources disables that DMA channel.

3.3.8.2 DMA transfers via PIT trigger

The PIT module can trigger a DMA transfer on the first four DMA channels. The assignments are detailed at PIT/DMA Periodic Trigger Assignments .

3.3.9 DMA Controller Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

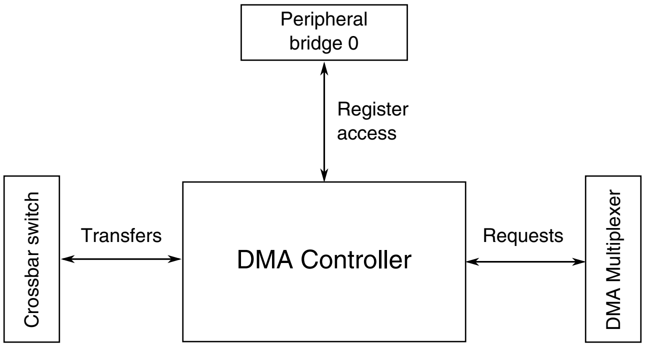

Peripheral bridge 0
Register access
Crossbar switch
Transfers
DMA Controller
Requests
DMA Multiplexer

Figure 3-13. DMA Controller configuration

Table 3-19. Reference links to related information

Topic	Related module	Reference
Full description	DMA Controller	DMA Controller
System memory map		System memory map
Register access	Peripheral bridge (AIPS-Lite 0)	AIPS-Lite 0
Clocking		Clock distribution
Power management		Power management
Transfers	Crossbar switch	Crossbar switch

3.3.10 External Watchdog Monitor (EWM) Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

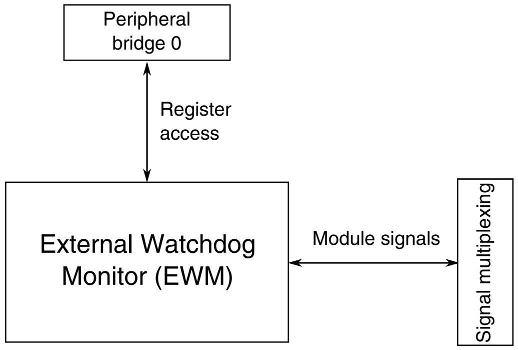

Peripheral bridge 0
Register access
External Watchdog Monitor (EWM)
Module signals
Signal multiplexing

Figure 3-14. External Watchdog Monitor configuration

Table 3-20. Reference links to related information

Topic	Related module	Reference
Full description	External Watchdog Monitor (EWM)	EWM
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port Control Module	Signal multiplexing

3.3.10.1 EWM clocks

This table shows the EWM clocks and the corresponding chip clocks.

Table 3-21. EWM clock connections

Module clock	Chip clock
Low Power Clock	1 kHz LPO Clock

3.3.10.2 EWM low-power modes

This table shows the EWM low-power modes and the corresponding chip low-power modes.

Table 3-22. EWM low-power modes

Module mode	Chip mode
Wait	Wait, VLPW
Stop	Stop, VLPS, LLS
Power Down	VLLS3, VLLS2, VLLS1

3.3.10.3 $\overline{\text{EWM\_OUT}}$ pin state in low power modes

During Wait, Stop and Power Down modes the $\overline{\text{EWM\_OUT}}$ pin enters a high-impedance state. A user has the option to control the logic state of the pin using an external pull device or by configuring the internal pull device. When the CPU enters a Run mode from Wait or Stop recovery, the pin resumes its previous state before entering Wait or Stop mode. When the CPU enters Run mode from Power Down, the pin returns to its reset state.

3.3.11 Watchdog Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

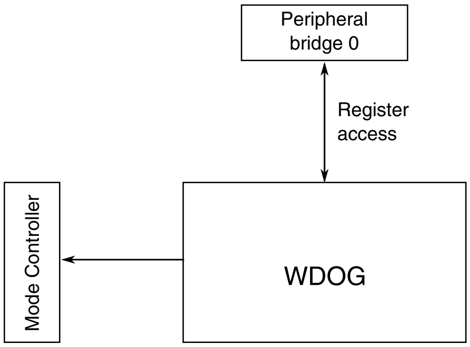


Figure 3-15. Watchdog configuration

Table 3-23. Reference links to related information

Topic	Related module	Reference
Full description	Watchdog	Watchdog
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
	Mode Controller (MC)	System Mode Controller

3.3.11.1 WDOG clocks

This table shows the WDOG module clocks and the corresponding chip clocks.

Table 3-24. WDOG clock connections

Module clock	Chip clock
LPO Oscillator	1 kHz LPO Clock
Alt Clock	Bus Clock
Fast Test Clock	Bus Clock
System Bus Clock	Bus Clock

3.3.11.2 WDOG low-power modes

This table shows the WDOG low-power modes and the corresponding chip low-power modes.

Table 3-25. WDOG low-power modes

Module mode	Chip mode
Wait	Wait, VLPW
Stop	Stop, VLPS
Power Down	LLS, VLLSx

3.4 Clock Modules

3.4.1 MCG Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

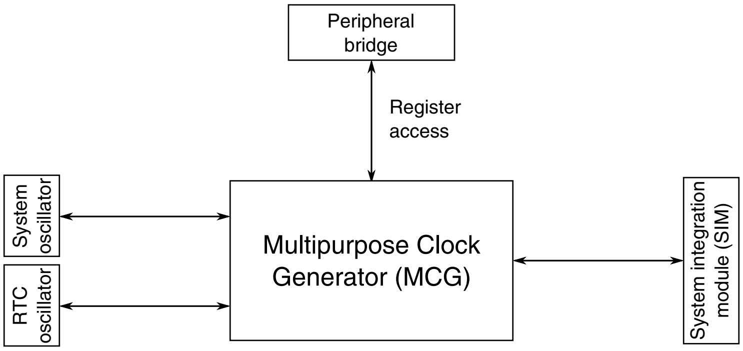

Peripheral bridge
Register access
System oscillator
RTC oscillator
Multipurpose Clock Generator (MCG)
System integration module (SIM)

Figure 3-16. MCG configuration

Table 3-26. Reference links to related information

Topic	Related module	Reference
Full description	MCG	MCG
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

3.4.2 OSC Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

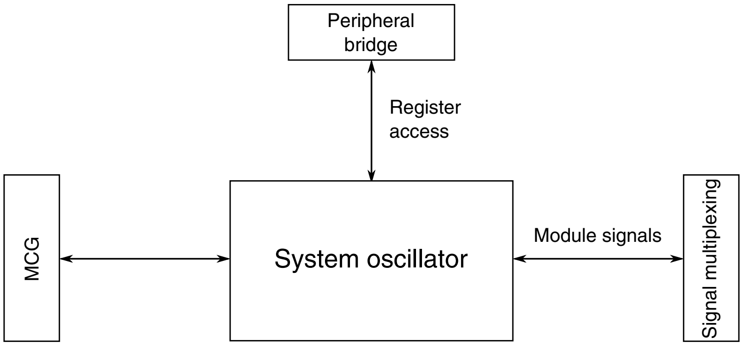

Peripheral bridge
Register access
MCG
System oscillator
Module signals
Signal multiplexing

Figure 3-17. OSC configuration

Table 3-27. Reference links to related information

Topic	Related module	Reference
Full description	OSC	OSC
System memory map		System memory map
Clocking		Clock distribution

Table continues on the next page...

Table 3-27. Reference links to related information (continued)

Topic	Related module	Reference
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing
Full description	MCG	MCG

3.4.2.1 OSC modes of operation with MCG

The MCG's C2 register bits configure the oscillator frequency range. See the OSC and MCG chapters for more details.

3.4.3 RTC OSC configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

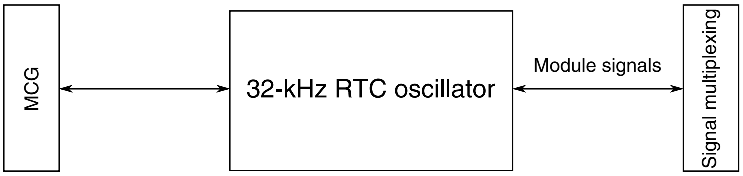


Figure 3-18. RTC OSC configuration

Table 3-28. Reference links to related information

Topic	Related module	Reference
Full description	RTC OSC	RTC OSC
Signal multiplexing	Port control	Signal multiplexing
Full description	MCG	MCG

3.5 Memories and Memory Interfaces

3.5.1 Flash Memory Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

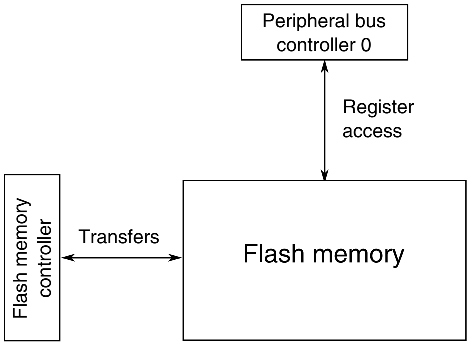


Figure 3-19. Flash memory configuration

Table 3-29. Reference links to related information

Topic	Related module	Reference
Full description	Flash memory	Flash memory
System memory map		System memory map
Clocking		Clock Distribution
Transfers	Flash memory controller	Flash memory controller
Register access	Peripheral bridge	Peripheral bridge

3.5.1.1 Flash memory types

This device contains the following types of flash memory:

- Program flash memory — non-volatile flash memory that can execute program code
- FlexMemory — encompasses the following memory types:
  - FlexNVM — Non-volatile flash memory that can execute program code, store data, or backup EEPROM data
  - FlexRAM — RAM memory that can be used as traditional RAM or as high-endurance EEPROM storage, and also accelerates flash programming

3.5.1.2 Flash Memory Sizes

The devices covered in this document contain:

- 1 block of program flash consisting of 1 KB sectors
- 1 block of FlexNVM consisting of 1 KB sectors
- 1 block of FlexRAM

The amounts of flash memory for the devices covered in this document are:

Device	Program flash (KB)	Block 0 (P-Flash) address range	FlexNVM (KB)	FlexRAM (KB)	FlexRAM address range
MK20DN32VLH5	32	0x0000_0000 – 0x0001_FFFF	—	—	—
MK20DX32VLH5	32	0x0000_0000 – 0x0001_FFFF	32	2	0x1400_0000 – 0x1400_0FFF
MK20DN64VLH5	64	0x0000_0000 – 0x0001_FFFF	—	—	—
MK20DX64VLH5	64	0x0000_0000 – 0x0001_FFFF	32	2	0x1400_0000 – 0x1400_0FFF
MK20DN128VLH5	128	0x0000_0000 – 0x0001_FFFF	—	—	—
MK20DX128VLH5	128	0x0000_0000 – 0x0001_FFFF	32	2	0x1400_0000 – 0x1400_0FFF
MK20DN32VMP5	32	0x0000_0000 – 0x0001_FFFF	—	—	—
MK20DX32VMP5	32	0x0000_0000 – 0x0001_FFFF	32	2	0x1400_0000 – 0x1400_0FFF
MK20DN64VMP5	64	0x0000_0000 – 0x0001_FFFF	—	—	—
MK20DX64VMP5	64	0x0000_0000 – 0x0001_FFFF	32	2	0x1400_0000 – 0x1400_0FFF
MK20DN128VMP5	128	0x0000_0000 – 0x0001_FFFF	—	—	—
MK20DX128VMP5	128	0x0000_0000 – 0x0001_FFFF	32	2	0x1400_0000 – 0x1400_0FFF

3.5.1.3 Flash Memory Map

The various flash memories and the flash registers are located at different base addresses as shown in the following figure. The base address for each is specified in System memory map.

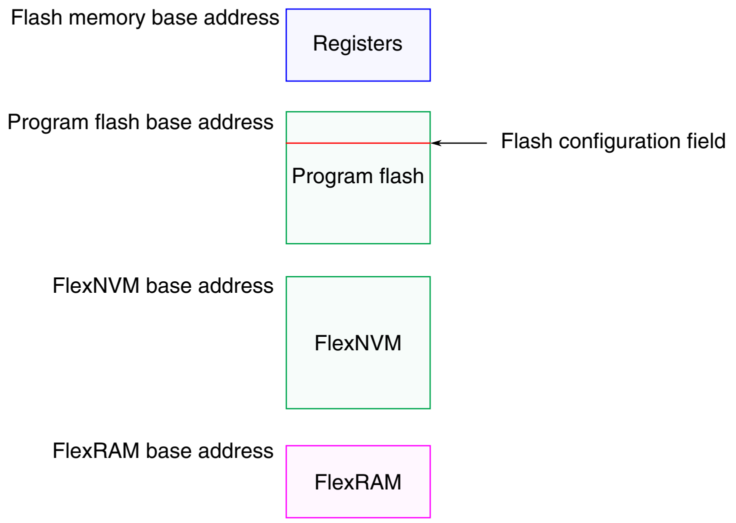

Flash memory base address
Registers
Program flash base address
Flash configuration field
Program flash
FlexNVM base address
FlexNVM
FlexRAM base address
FlexRAM

Figure 3-20. Flash memory map

3.5.1.4 Flash Security

How flash security is implemented on this device is described in Chip Security.

3.5.1.5 Flash Modes

The flash memory operates in NVM normal and NVM special modes. The flash memory enters NVM special mode when the EzPort is enabled ($\overline{\text{EZP\_CS}}$ asserted during reset), or the system is under debug mode. Otherwise, flash memory operates in NVM normal mode.

3.5.1.6 Erase All Flash Contents

In addition to software, the entire flash memory may be erased external to the flash memory in two ways:

1. Via the EzPort by issuing a bulk erase (BE) command. See the EzPort chapter for more details.
2. Via the SWJ-DP debug port by setting DAP_CONTROL[0]. DAP_STATUS[0] is set to indicate the mass erase command has been accepted. DAP_STATUS[0] is cleared when the mass erase completes.

3.5.1.7 FTFL_FOPT Register

The flash memory's FTFL_FOPT register allows the user to customize the operation of the MCU at boot time. See FOPT boot options for details of its definition.

3.5.2 Flash Memory Controller Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

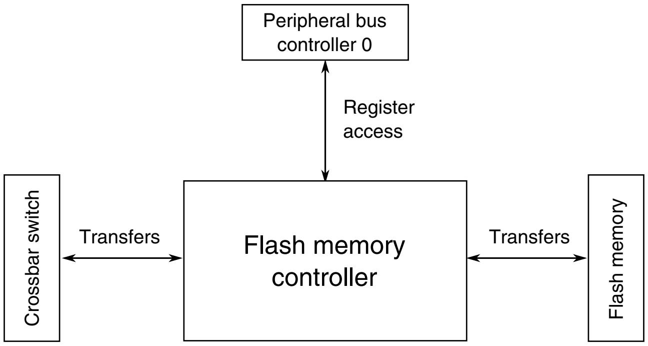

Peripheral bus controller 0
Register access
Crossbar switch
Transfers
Flash memory controller
Transfers
Flash memory

Figure 3-21. Flash memory controller configuration

Table 3-30. Reference links to related information

Topic	Related module	Reference
Full description	Flash memory controller	Flash memory controller
System memory map		System memory map
Clocking		Clock Distribution
Transfers	Flash memory	Flash memory
Transfers	Crossbar switch	Crossbar Switch
Register access	Peripheral bridge	Peripheral bridge

3.5.3 SRAM Configuration

This section summarizes how the module has been configured in the chip.

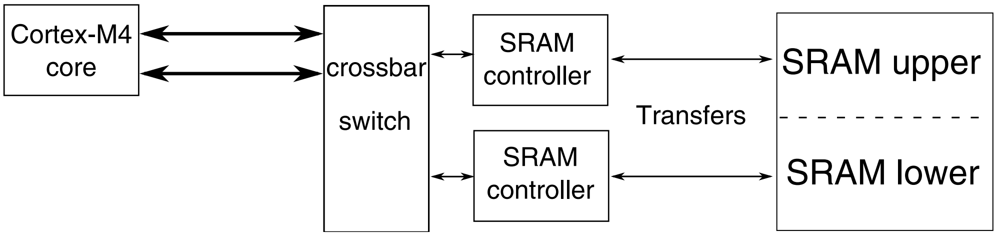

Cortex-M4 core
crossbar switch
SRAM controller
SRAM controller
Transfers
SRAM upper
SRAM lower

Figure 3-22. SRAM configuration

Table 3-31. Reference links to related information

Topic	Related module	Reference
Full description	SRAM	SRAM
System memory map		System memory map
Clocking		Clock Distribution
Transfers	SRAM controller	SRAM controller
	ARM Cortex-M4 core	ARM Cortex-M4 core

3.5.3.1 SRAM sizes

This device contains SRAM which could be accessed by bus masters through cross-bar switch. The amount of SRAM for the devices covered in this document is shown in the following table.

Device	SRAM (KB)
MK20DN32VLH5	8
MK20DX32VLH5	8
MK20DN64VLH5	16
MK20DX64VLH5	16
MK20DN128VLH5	16
MK20DX128VLH5	16
MK20DN32VMP5	8
MK20DX32VMP5	8
MK20DN64VMP5	16
MK20DX64VMP5	16
MK20DN128VMP5	16
MK20DX128VMP5	16

3.5.3.2 SRAM Arrays

The on-chip SRAM is split into two equally-sized logical arrays, SRAM_L and SRAM_U.

The on-chip RAM is implemented such that the SRAM_L and SRAM_U ranges form a contiguous block in the memory map. As such:

- SRAM_L is anchored to 0x1FFF_FFFF and occupies the space before this ending address.
- SRAM_U is anchored to 0x2000_0000 and occupies the space after this beginning address.

Valid address ranges for SRAM_L and SRAM_U are then defined as:

- SRAM_L = [0x2000_0000–(SRAM_size/2)] to 0x1FFF_FFFF
- SRAM_U = 0x2000_0000 to [0x2000_0000+(SRAM_size/2)-1]

This is illustrated in the following figure.

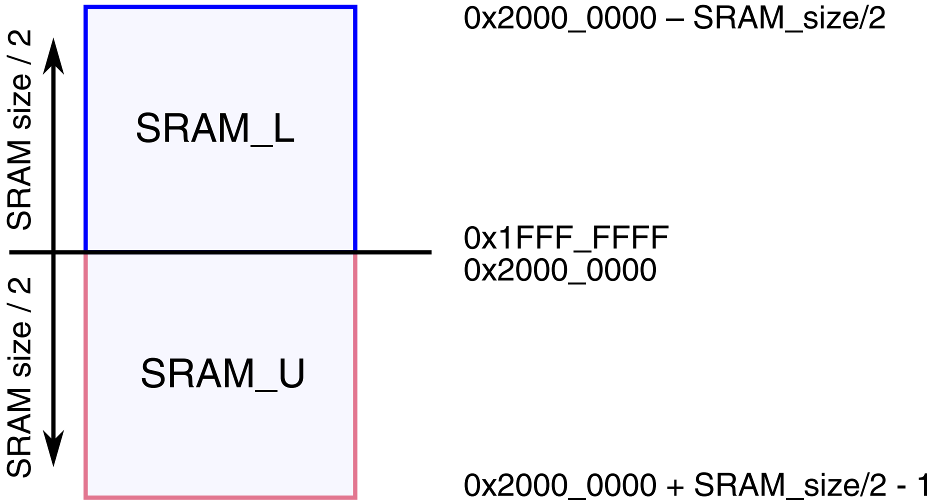


Figure 3-23. SRAM blocks memory map

For example, for a device containing 64 KB of SRAM the ranges are:

- SRAM_L: 0x1FFF_8000 – 0x1FFF_FFFF
- SRAM_U: 0x2000_0000 – 0x2000_7FFF

3.5.3.3 SRAM retention in low power modes

The SRAM is retained down to VLLS3 mode.

In VLLS2 the entire KB region of SRAM_U from 0x2000_0000 is powered.

In VLLS1 and VLLS0 no SRAM is retained; however, the 32-byte register file is available.

3.5.4 System Register File Configuration

This section summarizes how the module has been configured in the chip.

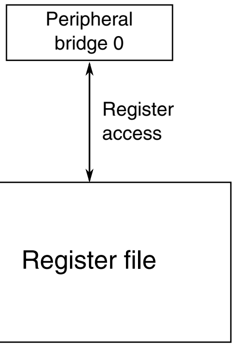


Figure 3-24. System Register file configuration

Table 3-32. Reference links to related information

Topic	Related module	Reference
Full description	Register file	Register file
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management

3.5.4.1 System Register file

This device includes a 32-byte register file that is powered in all power modes.

Also, it retains contents during low-voltage detect (LVD) events and is only reset during a power-on reset.

3.5.5 VBAT Register File Configuration

This section summarizes how the module has been configured in the chip.

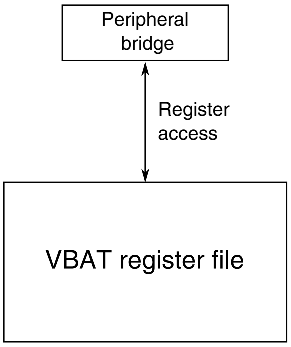

Peripheral bridge
Register access
VBAT register file

Figure 3-25. VBAT Register file configuration

Table 3-33. Reference links to related information

Topic	Related module	Reference
Full description	VBAT register file	VBAT register file
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management

3.5.5.1 VBAT register file

This device includes a 32-byte register file that is powered in all power modes and is powered by VBAT.

It is only reset during VBAT power-on reset.

3.5.6 EzPort Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

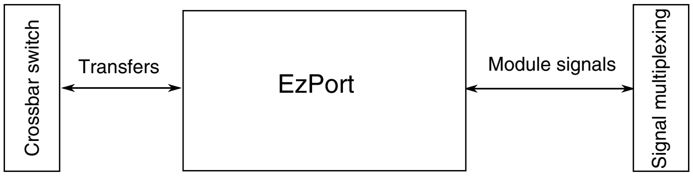

Crossbar switch
Transfers
EzPort
Module signals
Signal multiplexing

Figure 3-26. EzPort configuration

Table 3-34. Reference links to related information

Topic	Related module	Reference
Full description	EzPort	EzPort

Table continues on the next page...

Table 3-34. Reference links to related information (continued)

Topic	Related module	Reference
System memory map		System memory map
Clocking		Clock Distribution
Transfers	Crossbar switch	Crossbar switch
Signal Multiplexing	Port control	Signal Multiplexing

3.5.6.1 JTAG instruction

The system JTAG controller implements an EZPORT instruction. When executing this instruction, the JTAG controller resets the core logic and asserts the EzPort chip select signal to force the processor into EzPort mode.

3.5.6.2 Flash Option Register (FOPT)

The FOPT[EZPORT_DIS] bit can be used to prevent entry into EzPort mode during reset. If the FOPT[EZPORT_DIS] bit is cleared, then the state of the chip select signal ($\overline{\text{EZP\_CS}}$) is ignored and the MCU always boots in normal mode.

This option is useful for systems that use the $\overline{\text{EZP\_CS}}$/NMI signal configured for its NMI function. Disabling EzPort mode prevents possible unwanted entry into EzPort mode if the external circuit that drives the NMI signal asserts it during reset.

The FOPT register is loaded from the flash option byte. If the flash option byte is modified the new value takes effect for any subsequent resets, until the value is changed again.

3.6 Security

3.6.1 CRC Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

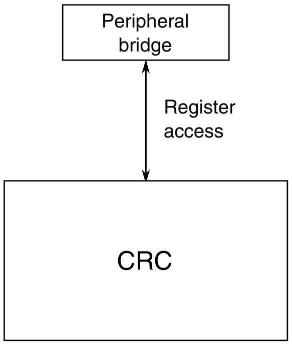

Peripheral bridge
Register access
CRC

Figure 3-27. CRC configuration

Table 3-35. Reference links to related information

Topic	Related module	Reference
Full description	CRC	CRC
System memory map		System memory map
Power management		Power management

3.7 Analog

3.7.1 16-bit SAR ADC Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

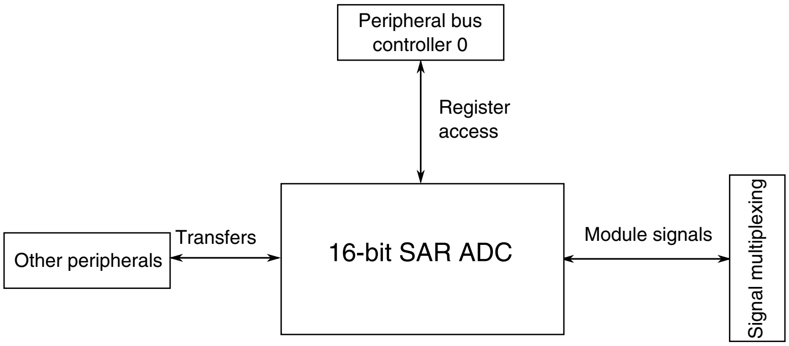

Peripheral bus controller 0
Register access
Other peripherals
Transfers
16-bit SAR ADC
Module signals
Signal multiplexing

Figure 3-28. 16-bit SAR ADC configuration

Table 3-36. Reference links to related information

Topic	Related module	Reference
Full description	16-bit SAR ADC	16-bit SAR ADC
System memory map		System memory map

Table continues on the next page...

Table 3-36. Reference links to related information (continued)

Topic	Related module	Reference
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

3.7.1.1 ADC instantiation information

This device contains one ADC.

3.7.1.1.1 Number of ADC channels

The number of ADC channels present on the device is determined by the pinout of the specific device package. For details regarding the number of ADC channel available on a particular package, refer to the signal multiplexing chapter of this MCU.

3.7.1.2 DMA Support on ADC

Applications may require continuous sampling of the ADC (4K samples/sec) that may have considerable load on the CPU. Though using PDB to trigger ADC may reduce some CPU load, The ADC supports DMA request functionality for higher performance when the ADC is sampled at a very high rate or cases were PDB is bypassed. The ADC can trigger the DMA (via DMA req) on conversion completion.

3.7.1.3 Connections/Channel Assignment

3.7.1.3.1 ADC0 Connections/Channel Assignment

NOTE

> As indicated by the following sections, each ADCx_DPx input and certain ADCx_DMx inputs may operate as single-ended ADC channels in single-ended mode.

3.7.1.3.1.1 ADC0 Channel Assignment for 64-Pin Package

ADC Channel (SC1n[ADCH])	Channel	Input signal (SC1n[DIFF]= 1)	Input signal (SC1n[DIFF]= 0)
00000	DAD0	ADC0_DP0 and ADC0_DM0	ADC0_DP0
00001	DAD1	Reserved	Reserved
00010	DAD2	Reserved	Reserved
00011	DAD3	ADC0_DP3 and ADC0_DM3	ADC0_DP3
00100[1]	AD4a	Reserved	Reserved
00101[1]	AD5a	Reserved	Reserved
00110[1]	AD6a	Reserved	Reserved
00111[1]	AD7a	Reserved	Reserved
00100[1]	AD4b	Reserved	ADC0_SE4b
00101[1]	AD5b	Reserved	ADC0_SE5b
00110[1]	AD6b	Reserved	ADC0_SE6b
00111[1]	AD7b	Reserved	ADC0_SE7b
01000	AD8	Reserved	ADC0_SE8
01001	AD9	Reserved	ADC0_SE9
01010	AD10	Reserved	Reserved
01011	AD11	Reserved	Reserved
01100	AD12	Reserved	ADC0_SE12
01101	AD13	Reserved	ADC0_SE13
01110	AD14	Reserved	ADC0_SE14
01111	AD15	Reserved	ADC0_SE15
10000	AD16	Reserved	Reserved
10001	AD17	Reserved	Reserved
10010	AD18	Reserved	Reserved
10011	AD19	Reserved	ADC0_DM0
10100	AD20	Reserved	Reserved
10101	AD21	Reserved	ADC0_DM3
10110	AD22	Reserved	VREF Output
10111	AD23	Reserved	/ADC0_SE23
11000	AD24	Reserved	Reserved
11001	AD25	Reserved	Reserved
11010	AD26	Temperature Sensor (Diff)	Temperature Sensor (S.E)
11011	AD27	Bandgap (Diff)[2]	Bandgap (S.E)[2]
11100	AD28	Reserved	Reserved
11101	AD29	-VREFH (Diff)	VREFH (S.E)
11110	AD30	Reserved	VREFL
11111	AD31	Module Disabled	Module Disabled

1. ADCx_CFG2[MUXSEL] bit selects between ADCx_SEn channels a and b. Refer to MUXSEL description in ADC chapter for details.

2. This is the PMC bandgap 1V reference voltage not the VREF module 1.2 V reference voltage. Prior to reading from this ADC channel, ensure that you enable the bandgap buffer by setting the PMC_REGSC[BGBE] bit. Refer to the device data sheet for the bandgap voltage (V_{BG}) specification.

3.7.1.4 ADC Channels MUX Selection

The following figure shows the assignment of ADCx_SEn channels a and b through a MUX selection to ADC. To select between alternate set of channels, refer to ADCx_CFG2[MUXSEL] bit settings for more details.

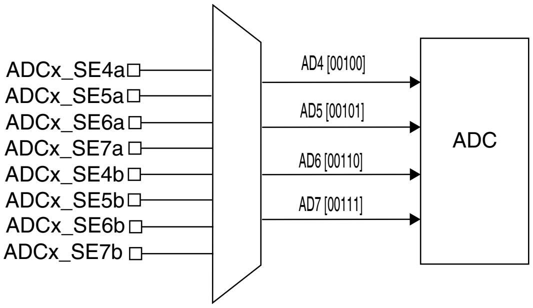


Figure 3-29. ADCx_SEn channels a and b selection

3.7.1.5 ADC Reference Options

The ADC supports the following references:

- VREFH/VREFL - connected as the primary reference option
- 1.2 V VREF_OUT - connected as the V_{ALT} reference option

ADCx_SC2[REFSEL] bit selects the voltage reference sources for ADC. Refer to REFSEL description in ADC chapter for more details.

3.7.1.6 ADC triggers

The ADC supports both software and hardware triggers. The primary hardware mechanism for triggering the ADC is the PDB. The PDB itself can be triggered by other peripherals. For example: RTC (Alarm, Seconds) signal is connected to the PDB. The PDB trigger can receive the RTC (alarm/seconds) trigger input forcing ADC conversions in run mode (where PDB is enabled). On the other hand, the ADC can conduct conversions in low power modes, not triggered by PDB. This allows the ADC to do conversions in low power mode and store the output in the result register. The ADC generates interrupt when the data is ready in the result register that wakes the system from low power mode. The PDB can also be bypassed by using the ADCxTRGSEL bits in the SOPT7 register.

For operation of triggers in different modes, refer to Power Management chapter.

3.7.1.7 Alternate clock

For this device, the alternate clock is connected to OSCERCLK.

NOTE

> This clock option is only usable when OSCERCLK is in the MHz range. A system with OSCERCLK in the kHz range has the optional clock source below minimum ADC clock operating frequency.

3.7.1.8 ADC low-power modes

This table shows the ADC low-power modes and the corresponding chip low-power modes.

Table 3-37. ADC low-power modes

Module mode	Chip mode
Wait	Wait, VLPW
Normal Stop	Stop, VLPS
Low Power Stop	LLS, VLLS3, VLLS2, VLLS1, VLLS0

3.7.2 CMP Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

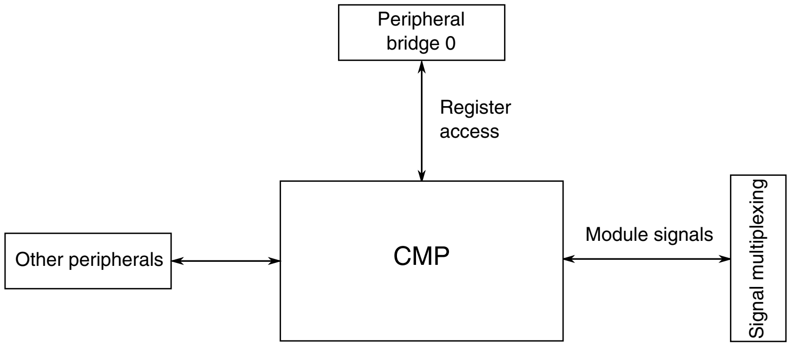


Figure 3-30. CMP configuration

Table 3-38. Reference links to related information

Topic	Related module	Reference
Full description	Comparator (CMP)	Comparator
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

3.7.2.1 CMP input connections

The following table shows the fixed internal connections to the CMP.

Table 3-39. CMP input connections

CMP Inputs	CMP0	CMP1
IN0	CMP0_IN0	CMP1_IN0
IN1	CMP0_IN1	CMP1_IN1
IN2	CMP0_IN2	—
IN3	CMP0_IN3	—
IN4	CMP0_IN4	—
IN5	VREF Output/CMP0_IN5	VREF Output/CMP1_IN5
IN6	Bandgap	Bandgap
IN7	6b DAC0 Reference	6b DAC1 Reference

3.7.2.2 CMP external references

The 6-bit DAC sub-block supports selection of two references. For this device, the references are connected as follows:

- VREF_OUT - V_{in1} input
- VDD - V_{in2} input

3.7.2.3 External window/sample input

Individual PDB pulse-out signals control each CMP Sample/Window timing.

3.7.3 VREF Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

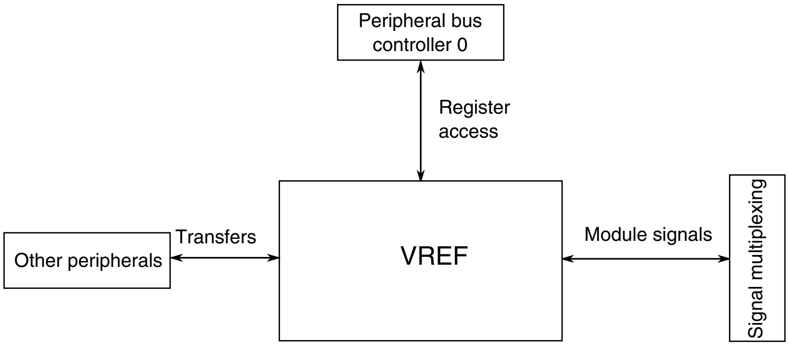

Peripheral bus controller 0
Register access
Transfers
Other peripherals
VREF
Module signals
Signal multiplexing

Figure 3-31. VREF configuration

Table 3-40. Reference links to related information

Topic	Related module	Reference
Full description	VREF	VREF
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

3.7.3.1 VREF Overview

This device includes a voltage reference (VREF) to supply an accurate 1.2 V voltage output.

The voltage reference can provide a reference voltage to external peripherals or a reference to analog peripherals, such as the ADC or CMP.

NOTE

PMC_REGSC[BGEN] bit must be set if the VREF regulator is required to remain operating in VLPx modes.

NOTE

For either an internal or external reference if the VREF_OUT functionality is being used, VREF_OUT signal must be connected to an output load capacitor. Refer the device data sheet for more details.

3.8 Timers

3.8.1 PDB Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

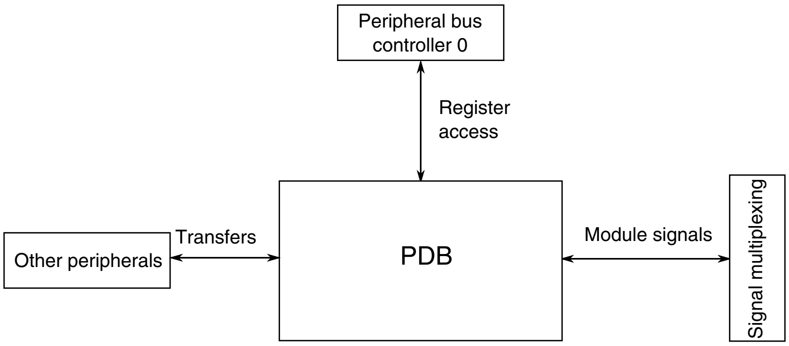

Peripheral bus controller 0
Register access
Transfers
Other peripherals
PDB
Module signals
Signal multiplexing

Figure 3-32. PDB configuration

Table 3-41. Reference links to related information

Topic	Related module	Reference
Full description	PDB	PDB
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

3.8.1.1 PDB Instantiation

3.8.1.1.1 PDB Output Triggers

Table 3-42. PDB output triggers

Number of PDB channels for ADC trigger	1
Number of pre-triggers per PDB channel	2
Number of PulseOut	2

NOTE

There is an additional channel 1 to inter-connect with FTM0.

3.8.1.1.2 PDB Input Trigger Connections

Table 3-43. PDB Input Trigger Options

PDB Trigger	PDB Input
0000	External Trigger (PDB0_EXTRG)
0001	CMP 0
0010	CMP 1
0011	Reserved
0100	PIT Ch 0 Output
0101	PIT Ch 1 Output
0110	PIT Ch 2 Output
0111	PIT Ch 3 Output
1000	FTM0 Init and Ext Trigger Outputs
1001	FTM1 Init and Ext Trigger Outputs
1010	Reserved
1011	Reserved
1100	RTC Alarm
1101	RTC Seconds
1110	LPTMR Output
1111	Software Trigger

3.8.1.2 PDB Module Interconnections

PDB trigger outputs	Connection
Channel 0 triggers	ADC0 trigger
Channel 1 triggers	synchronous input 1 of FTM0
Pulse-out	Pulse-out connected to each CMP module's sample/window input to control sample operation

3.8.1.3 Back-to-back acknowledgement connections

In this MCU, PDB back-to-back operation acknowledgment connections are implemented as follows:

- PDB channel 0 pre-trigger 0 acknowledgement input: ADC1SC1B_COCO
- PDB channel 0 pre-trigger 1 acknowledgement input: ADC0SC1A_COCO
- PDB channel 1 pre-trigger 0 acknowledgement input: ADC0SC1B_COCO
- PDB channel 1 pre-trigger 1 acknowledgement input: ADC1SC1A_COCO

So, the back-to-back chain is connected as a ring:

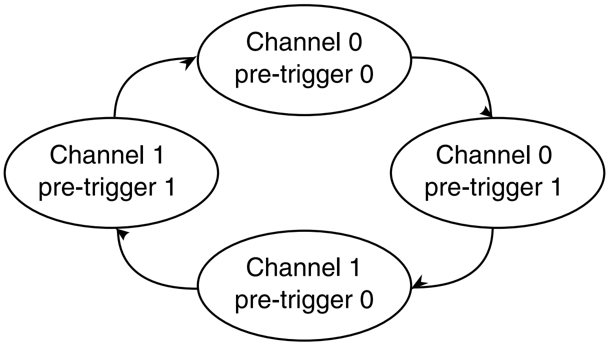


Figure 3-33. PDB back-to-back chain

The application code can set the PDB*x*_CH*n*C1[BB] bits to configure the PDB pre-triggers as a single chain or several chains.

3.8.1.4 Pulse-Out Connection

Individual PDB Pulse-Out signals are connected to each CMP block and used for sample window.

3.8.1.5 Pulse-Out Enable Register Implementation

The following table shows the comparison of pulse-out enable register at the module and chip level.

Table 3-44. PDB pulse-out enable register

Register	Module implementation	Chip implementation
POnEN	7:0 - POEN 31:8 - Reserved	0 - POEN[0] for CMP0 1 - POEN[1] for CMP1 31:2 - Reserved

3.8.2 FlexTimer Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

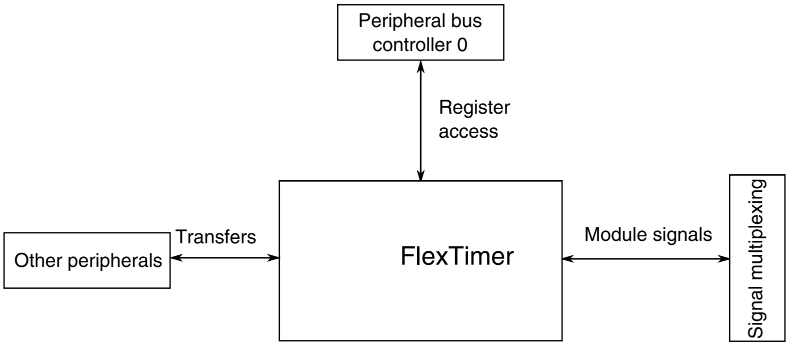

Peripheral bus controller 0
Register access
Other peripherals
Transfers
FlexTimer
Module signals
Signal multiplexing

Figure 3-34. FlexTimer configuration

Table 3-45. Reference links to related information

Topic	Related module	Reference
Full description	FlexTimer	FlexTimer
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

3.8.2.1 Instantiation Information

This device contains two FlexTimer modules.

The following table shows how these modules are configured.

Table 3-46. FTM Instantiations

FTM instance	Number of channels	Features/usage
FTM0	8	3-phase motor + 2 general purpose or stepper motor
FTM1	2[1]	Quadrature decoder or general purpose

[1] Only channels 0 and 1 are available.

Compared with the FTM0 configuration, the FTM1 configuration adds the Quadrature decoder feature and reduces the number of channels.

3.8.2.2 External Clock Options

By default each FTM is clocked by the internal bus clock (the FTM refers to it as system clock). Each module contains a register setting that allows the module to be clocked from an external clock instead. There are two external FTM_CLKINx pins that can be selected by any FTM module via the SOPT4 register in the SIM module.

3.8.2.3 Fixed frequency clock

The fixed frequency clock for each FTM is MCGFFCLK.

3.8.2.4 FTM Interrupts

The FlexTimer has multiple sources of interrupt. However, these sources are OR'd together to generate a single interrupt request per FTM module to the interrupt controller. When an FTM interrupt occurs, read the FTM status registers (FMS, SC, and STATUS) to determine the exact interrupt source.

3.8.2.5 FTM Fault Detection Inputs

The following fault detection input options for the FTM modules are selected via the SOPT4 register in the SIM module. The external pin option is selected by default.

- FTM0 FAULT0 = FTM0_FLT0 pin or CMP0 output
- FTM0 FAULT1 = FTM0_FLT1 pin or CMP1 output
- FTM0 FAULT2 = FTM0_FLT2 pin
- FTM0 FAULT3 = FTM0_FLT3 pin

- FTM1 FAULT0 = FTM1_FLT0 pin or CMP0 output
- FTM1 FAULT1 = CMP1 output

3.8.2.6 FTM Hardware Triggers

The FTM synchronization hardware triggers are connected in the chip as follows:

- FTM0 hardware trigger 0 = CMP0 Output or FTM1 Match (when enabled in the FTM1 External Trigger (EXTTRIG) register)
- FTM0 hardware trigger 1 = PDB channel 1 Trigger Output
- FTM0 hardware trigger 2 = FTM0_FLT0 pin

- FTM1 hardware trigger 0 = CMP0 Output
- FTM1 hardware trigger 1 = CMP1 Output
- FTM1 hardware trigger 2 = FTM1_FLT0 pin

For the triggers with more than one option, the SOPT4 register in the SIM module controls the selection.

3.8.2.7 Input capture options for FTM module instances

The following channel 0 input capture source options are selected via the SOPT4 register in the SIM module. The external pin option is selected by default.

- FTM1 channel 0 input capture = FTM1_CH0 pin or CMP0 output or CMP1 output or USB start of frame pulse

NOTE

> When the USB start of frame pulse option is selected as an FTM channel input capture, disable the USB SOF token interrupt in the USB Interrupt Enable register (INTEN[SOFTOKEN]) to avoid USB enumeration conflicts.

3.8.2.8 FTM output triggers for other modules

FTM output triggers can be selected as input triggers for the PDB and ADC modules. See PDB Instantiation and ADC triggers.

3.8.2.9 FTM Global Time Base

This chip provides the optional FTM global time base feature (see Global time base (GTB)).

FTM0 provides the only source for the FTM global time base. The other FTM modules can share the time base as shown in the following figure:

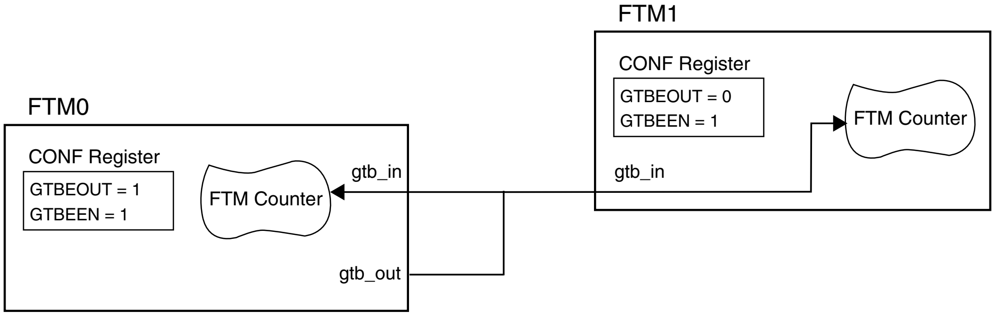

FTM1
CONF Register
GTBEOUT = 0
GTBEEN = 1
FTM Counter
gtb_in
FTM0
CONF Register
GTBEOUT = 1
GTBEEN = 1
FTM Counter
gtb_in
gtb_out

Figure 3-35. FTM Global Time Base Configuration

3.8.2.10 FTM BDM and debug halt mode

In the FTM chapter, references to the chip being in "BDM" are the same as the chip being in “debug halt mode".

3.8.3 PIT Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

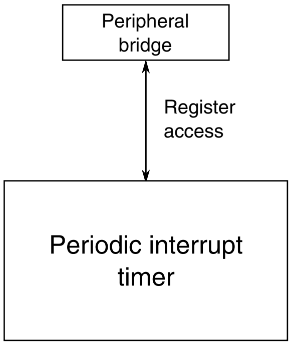


Figure 3-36. PIT configuration

Table 3-47. Reference links to related information

Topic	Related module	Reference
Full description	PIT	PIT
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management

3.8.3.1 PIT/DMA Periodic Trigger Assignments

The PIT generates periodic trigger events to the DMA Mux as shown in the table below.

Table 3-48. PIT channel assignments for periodic DMA triggering

DMA Channel Number	PIT Channel
DMA Channel 0	PIT Channel 0
DMA Channel 1	PIT Channel 1
DMA Channel 2	PIT Channel 2
DMA Channel 3	PIT Channel 3

3.8.3.2 PIT/ADC Triggers

PIT triggers are selected as ADCx trigger sources using the SOPT7[ADCxTRGSEL] bits in the SIM module. For more details, refer to SIM chapter.

3.8.4 Low-power timer configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

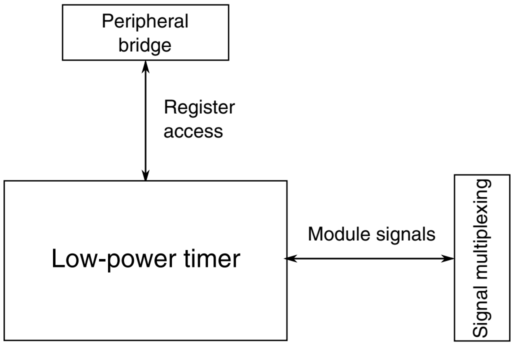


Figure 3-37. LPT configuration

Table 3-49. Reference links to related information

Topic	Related module	Reference
Full description	Low-power timer	Low-power timer
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management
Signal Multiplexing	Port control	Signal Multiplexing

3.8.4.1 LPTMR prescaler/glitch filter clocking options

The prescaler and glitch filter of the LPTMR module can be clocked from one of four sources determined by the LPTMR0_PSR[PCS] bitfield. The following table shows the chip-specific clock assignments for this bitfield.

NOTE

The chosen clock must remain enabled if the LPTMR is to continue operating in all required low-power modes.

LPTMR0_PSR[PCS]	Prescaler/glitch filter clock number	Chip clock
00	0	MCGIRCLK — internal reference clock (not available in VLPS/LLS/VLLS modes)
01	1	LPO — 1 kHz clock (not available in VLLS0 mode)
10	2	ERCLK32K — secondary external reference clock
11	3	OSCERCLK — external reference clock (not available in VLLS0 mode)

See Clock Distribution for more details on these clocks.

3.8.4.2 LPTMR pulse counter input options

The LPTMR_CSR[TPS] bitfield configures the input source used in pulse counter mode. The following table shows the chip-specific input assignments for this bitfield.

LPTMR_CSR[TPS]	Pulse counter input number	Chip input
00	0	CMP0 output
01	1	LPTMR_ALT1 pin
10	2	LPTMR_ALT2 pin
11	3	Reserved

3.8.5 CMT Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

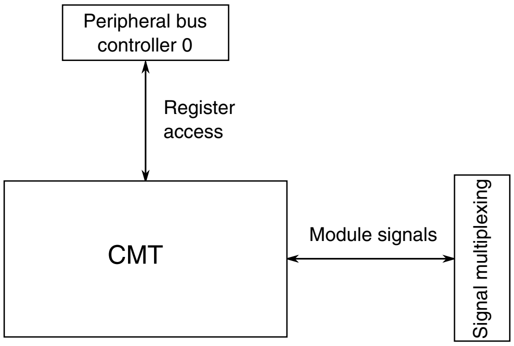


Figure 3-38. CMT configuration

Table 3-50. Reference links to related information

Topic	Related module	Reference
Full description	Carrier modulator transmitter (CMT)	CMT
System memory map		System memory map
Clocking		Clock distribution
Power management		Power management
Signal multiplexing	Port control	Signal multiplexing

3.8.5.1 Instantiation Information

This device contains one CMT module.

3.8.5.2 IRO Drive Strength

The IRO pad requires higher current drive than can be obtained from a single pad. For this device, the pin associated with the CMT_IRO signal is doubled bonded to two pads.

The SOPT2[PTD7PAD] field in SIM module can be used to configure the pin associated with the CMT_IRO signal as a higher current output port pin.

3.8.6 RTC configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

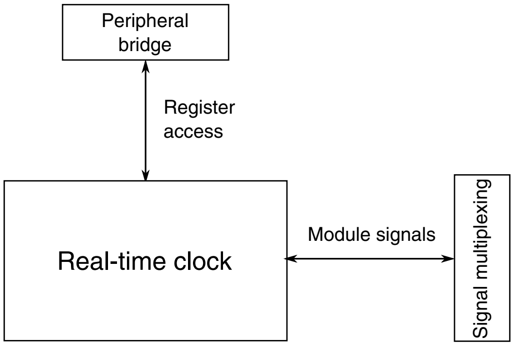


Figure 3-39. RTC configuration

Table 3-51. Reference links to related information

Topic	Related module	Reference
Full description	RTC	RTC
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management

3.8.6.1 RTC_CLKOUT signal

When the RTC is enabled and the port control module selects the RTC_CLKOUT function, the RTC_CLKOUT signal outputs a 1 Hz or 32 kHz output derived from RTC oscillator as shown below.

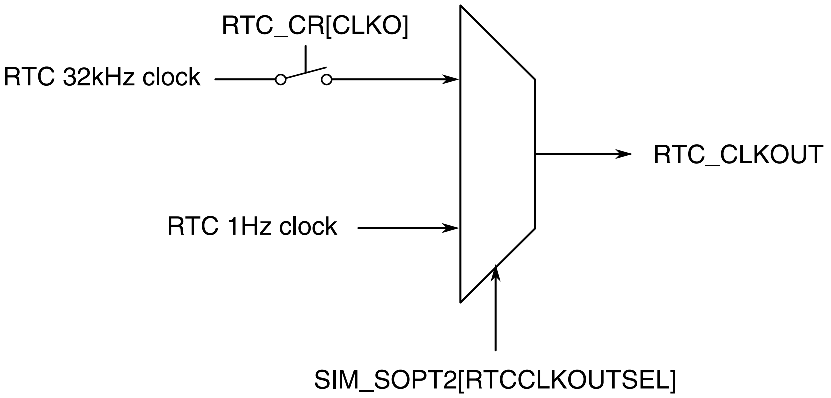

RTC_CR[CLKO]
RTC 32kHz clock
RTC_CLKOUT
RTC 1Hz clock
SIM_SOPT2[RTCCLKOUTSEL]

Figure 3-40. RTC_CLKOUT generation

3.8.6.2 RTC_WAKEUP signal

The RTC_WAKEUP pin is not supported on this device.

3.9 Communication interfaces

3.9.1 Universal Serial Bus (USB) FS Subsystem

The USB FS subsystem includes these components:

- Dual-role USB OTG-capable (On-The-Go) controller that supports a full-speed (FS) device or FS/LS host. The module complies with the USB 2.0 specification.
- USB transceiver that includes internal 15 kΩ pulldowns on the D+ and D- lines for host mode functionality.
- A 3.3 V regulator.
- USB device charger detection module.
- VBUS detect signal: To detect a valid VBUS in device mode, use a GPIO signal that can wake the chip in all power modes.

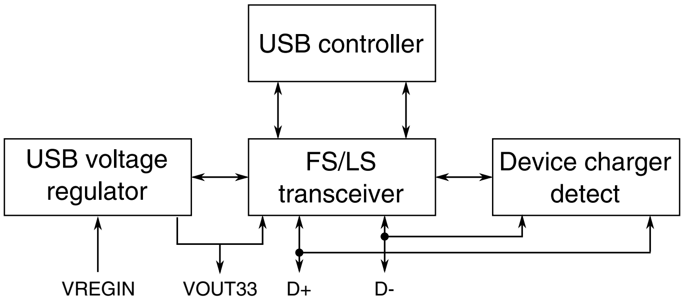

USB controller
USB voltage regulator
FS/LS transceiver
Device charger detect
VREGIN
VOUT33
D+
D-

Figure 3-41. USB Subsystem Overview

3.9.1.1 USB Wakeup

When the USB detects that there is no activity on the USB bus for more than 3 ms, the INT_STAT[SLEEP] bit is set. This bit can cause an interrupt and software decides the appropriate action.

Waking from a low power mode (except in LLS/VLLS mode where USB is not powered) occurs through an asynchronous interrupt triggered by activity on the USB bus. Setting the USBTRC0[USBRESMEN] bit enables this function.

3.9.1.2 USB Power Distribution

This chip includes an internal 5 V to 3.3 V USB regulator that powers the USB transceiver or the MCU (depending on the application).

3.9.1.2.1 AA/AAA cells power supply

The chip can be powered by two AA/AAA cells. In this case, the MCU is powered through VDD which is within the 1.8 to 3.0 V range. After USB cable insertion is detected, the USB regulator is enabled to power the USB transceiver.

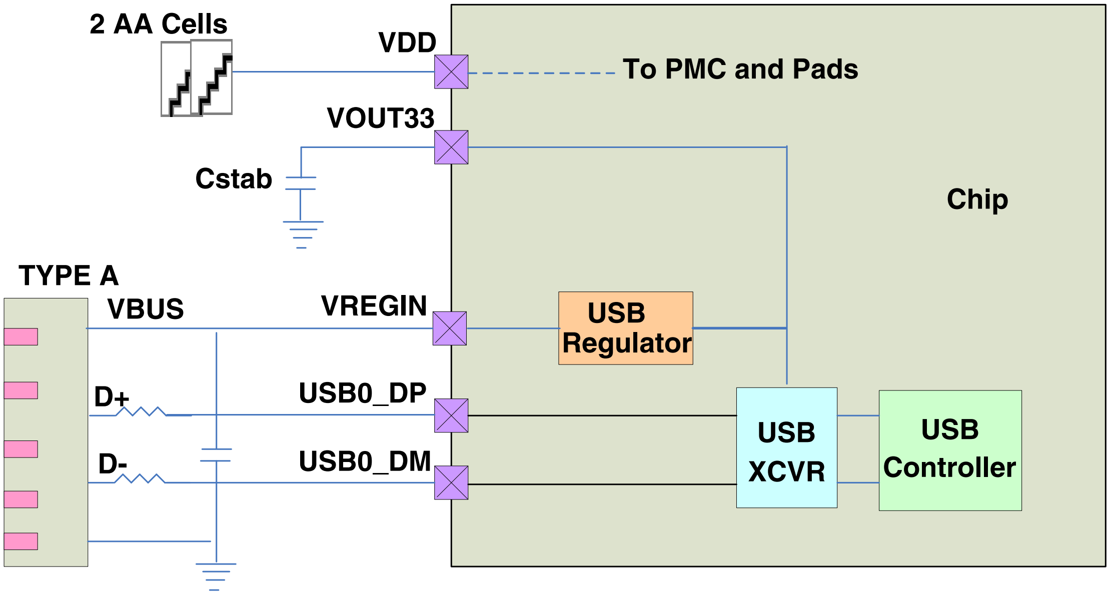

2 AA Cells
VDD
To PMC and Pads
VOUT33
Cstab
Chip
TYPE A
VBUS
VREGIN
USB Regulator
D+
USB0_DP
D-
USB0_DM
USB XCVR
USB Controller

Figure 3-42. USB regulator AA cell usecase

3.9.1.2.2 Li-Ion battery power supply

The chip can also be powered by a single Li-ion battery. In this case, VOUT33 is connected to VDD. The USB regulator must be enabled by default to power the MCU. When connected to a USB host, the input source of this regulator is switched to the USB bus supply from the Li-ion battery. To charge the battery, the MCU can configure the battery charger according to the charger detection information.

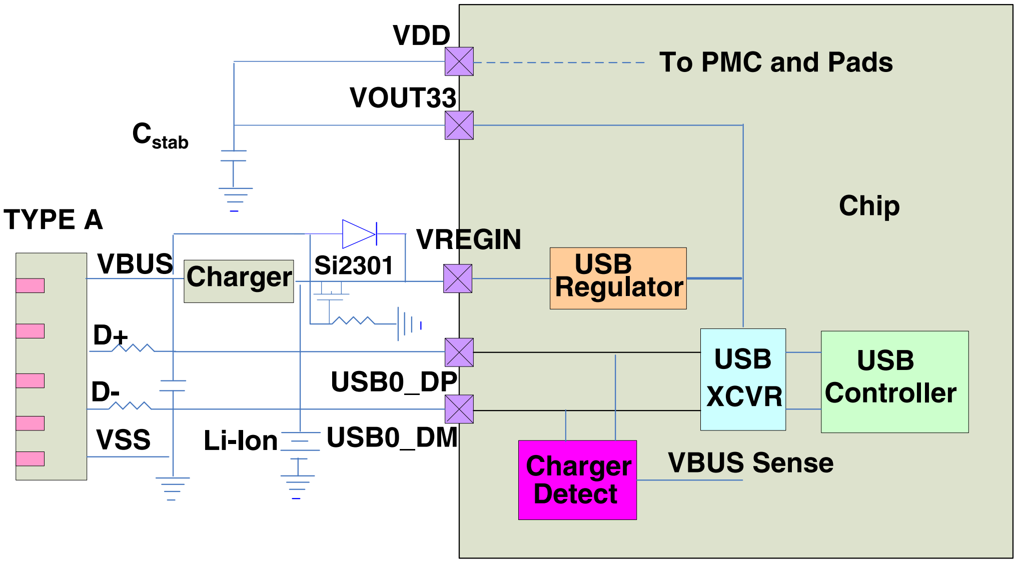

VDD
To PMC and Pads
VOUT33
C_stab
Chip
TYPE A
VREGIN
VBUS
Charger
Si2301
USB Regulator
D+
USB XCVR
USB Controller
D-
USB0_DP
VSS
Li-Ion
USB0_DM
Charger Detect
VBUS Sense

Figure 3-43. USB regulator Li-ion usecase

3.9.1.2.3 USB bus power supply

The chip can also be powered by the USB bus directly. In this case, VOUT33 is connected to VDD. The USB regulator must be enabled by default to power the MCU, then to power USB transceiver or external sensor.

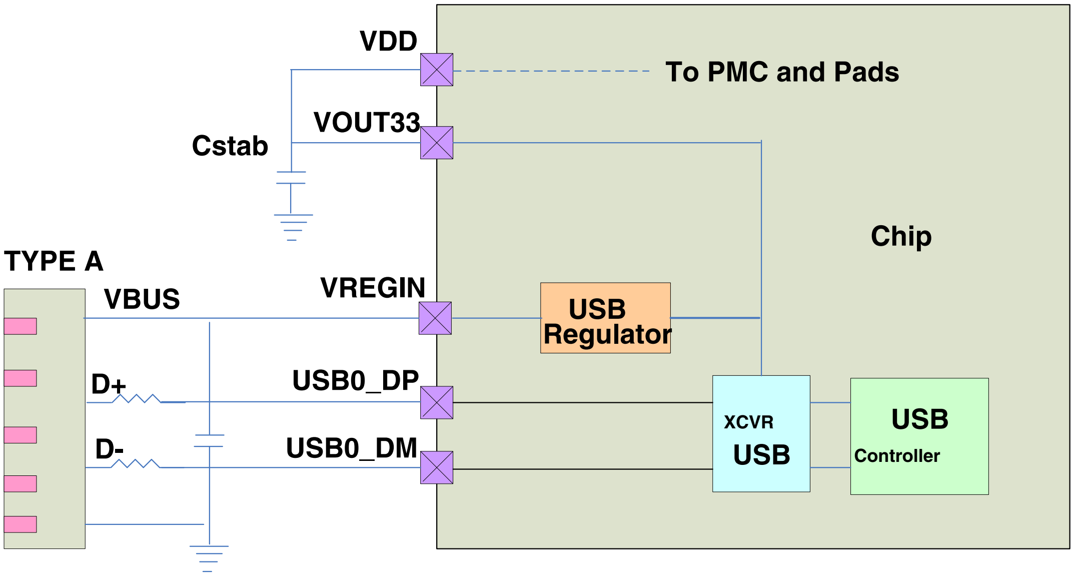

VDD
To PMC and Pads
VOUT33
Cstab
Chip
TYPE A
VBUS
VREGIN
USB Regulator
D+
USB0_DP
D-
USB0_DM
XCVR
USB
USB
Controller

Figure 3-44. USB regulator bus supply

3.9.1.3 USB power management

The regulator should be put into STANDBY mode whenever the chip is in Stop mode. This can be done by setting the two control bits on SIM_SOPT1.

3.9.1.4 USB controller configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

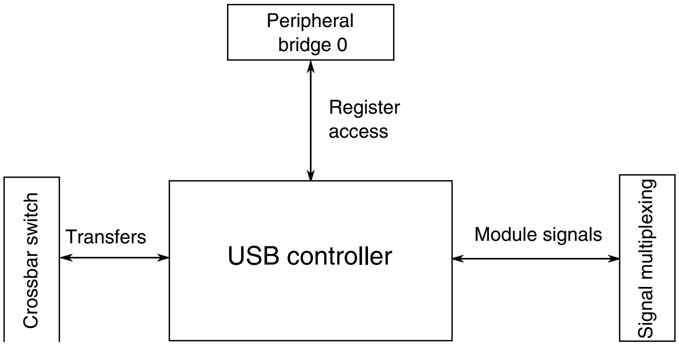

Peripheral bridge 0
Register access
Crossbar switch
Transfers
USB controller
Module signals
Signal multiplexing

Figure 3-45. USB controller configuration

Table 3-52. Reference links to related information

Topic	Related module	Reference
Full description	USB controller	USB controller

Table continues on the next page...

Table 3-52. Reference links to related information (continued)

Topic	Related module	Reference
System memory map		System memory map
Clocking		Clock Distribution
Transfers	Crossbar switch	Crossbar switch
Signal Multiplexing	Port control	Signal Multiplexing

NOTE

When USB is not used in the application, it is recommended that the USB regulator VREGIN and VOUT33 pins remain floating.

3.9.1.5 USB DCD Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

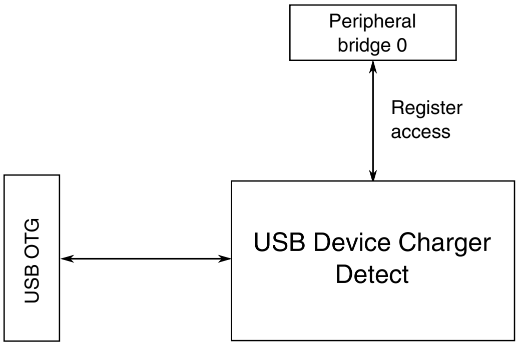


Figure 3-46. USB DCD configuration

Table 3-53. Reference links to related information

Topic	Related module	Reference
Full description	USB DCD	USB DCD
System memory map		System memory map
Clocking		Clock Distribution
	USB controller	USB controller

3.9.1.6 USB Voltage Regulator Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

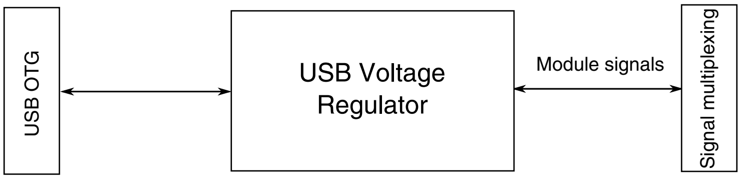

USB OTG
USB Voltage Regulator
Module signals
Signal multiplexing

Figure 3-47. USB Voltage Regulator configuration

Table 3-54. Reference links to related information

Topic	Related module	Reference
Full description	USB Voltage Regulator	USB Voltage Regulator
System memory map		System memory map
Clocking		Clock Distribution
	USB controller	USB controller
Signal Multiplexing	Port control	Signal Multiplexing

NOTE

When USB is not used in the application, it is recommended that the USB regulator VREGIN and VOUT33 pins remain floating.

3.9.2 SPI configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

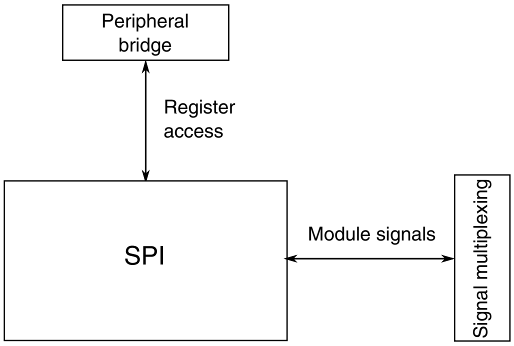

Peripheral bridge
Register access
SPI
Module signals
Signal multiplexing

Figure 3-48. SPI configuration

Table 3-55. Reference links to related information

Topic	Related module	Reference
Full description	SPI	SPI
System memory map		System memory map

Table continues on the next page...

Table 3-55. Reference links to related information (continued)

Topic	Related module	Reference
Clocking		Clock Distribution
Signal Multiplexing	Port control	Signal Multiplexing

3.9.2.1 SPI Modules Configuration

This device contains one SPI module.

3.9.2.2 SPI clocking

The SPI module is clocked by the internal bus clock (the DSPI refers to it as system clock). The module has an internal divider, with a minimum divide is two. So, the SPI can run at a maximum frequency of bus clock/2.

3.9.2.3 Number of CTARs

SPI CTAR registers define different transfer attribute configurations. The SPI module supports up to eight CTAR registers. This device supports two CTARs on all instances of the SPI.

In master mode, the CTAR registers define combinations of transfer attributes, such as frame size, clock phase, clock polarity, data bit ordering, baud rate, and various delays. In slave mode only CTAR0 is used, and a subset of its bitfields sets the slave transfer attributes.

3.9.2.4 TX FIFO size

Table 3-56. SPI transmit FIFO size

SPI Module	Transmit FIFO size
SPI0	4

3.9.2.5 RX FIFO Size

SPI supports up to 16-bit frame size during reception.

Table 3-57. SPI receive FIFO size

SPI Module	Receive FIFO size
SPI0	4

3.9.2.6 Number of PCS signals

The following table shows the number of peripheral chip select signals available per SPI module.

Table 3-58. SPI PCS signals

SPI Module	PCS Signals
SPI0	SPI_PCS[4:0]
SPI1	Not available

3.9.2.7 SPI Operation in Low Power Modes

In VLPR and VLPW modes the SPI is functional; however, the reduced system frequency also reduces the max frequency of operation for the SPI. In VLPR and VLPW modes the max SPI_CLK frequency is 2MHz.

In stop and VLPS modes, the clocks to the SPI module are disabled. The module is not functional, but it is powered so that it retains state.

There is one way to wake from stop mode via the SPI, which is explained in the following section.

3.9.2.7.1 Using GPIO Interrupt to Wake from stop mode

Here are the steps to use a GPIO to create a wakeup upon reception of SPI data in slave mode:

1. Point the GPIO interrupt vector to the desired interrupt handler.
2. Enable the GPIO input to generate an interrupt on either the rising or falling edge (depending on the polarity of the chip select signal).
3. Enter Stop or VLPS mode and Wait for the GPIO interrupt.

NOTE

It is likely that in using this approach the first word of data from the SPI host might not be received correctly. This is dependent on the transfer rate used for the SPI, the delay between chip select assertion and presentation of data, and the system interrupt latency.

3.9.2.8 SPI Doze Mode

The Doze mode for the SPI module is the same as the Wait and VLPW modes for the chip.

3.9.2.9 SPI Interrupts

The SPI has multiple sources of interrupt requests. However, these sources are OR'd together to generate a single interrupt request to the interrupt controller. When an SPI interrupt occurs, read the SPI_SR to determine the exact interrupt source.

3.9.2.10 SPI clocks

This table shows the SPI module clocks and the corresponding chip clocks.

Table 3-59. SPI clock connections

Module clock	Chip clock
System Clock	Bus Clock

3.9.3 I2C Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

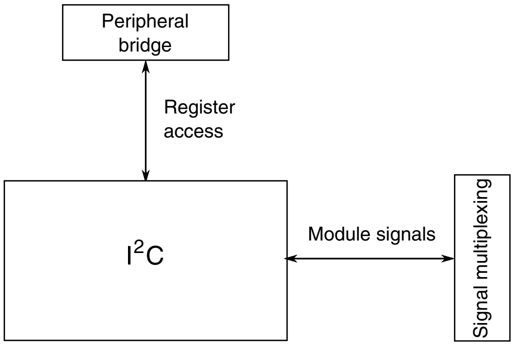

Peripheral bridge
Register access
I²C
Module signals
Signal multiplexing

Figure 3-49. I2C configuration

Table 3-60. Reference links to related information

Topic	Related module	Reference
Full description	I²C	I²C
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management
Signal Multiplexing	Port control	Signal Multiplexing

3.9.4 UART Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

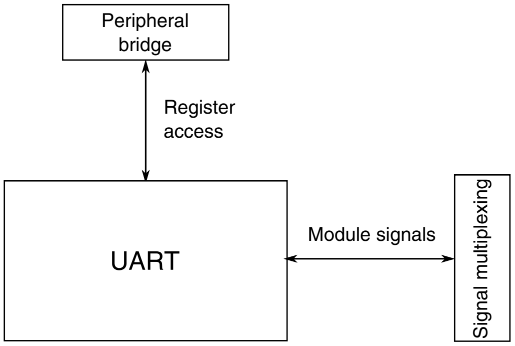

Peripheral bridge
Register access
UART
Module signals
Signal multiplexing

Figure 3-50. UART configuration

Table 3-61. Reference links to related information

Topic	Related module	Reference
Full description	UART	UART
System memory map		System memory map
Clocking		Clock Distribution

Table continues on the next page...

Table 3-61. Reference links to related information (continued)

Topic	Related module	Reference
Power management		Power management
Signal Multiplexing	Port control	Signal Multiplexing

3.9.4.1 UART configuration information

This device contains three UART modules. This section describes how each module is configured on this device.

1. Standard features of all UARTs:
    - RS-485 support
    - Hardware flow control (RTS/CTS)
    - 9-bit UART to support address mark with parity
    - MSB/LSB configuration on data
2. UART0 and UART1 are clocked from the core clock, the remaining UARTs are clocked on the bus clock. The maximum baud rate is 1/16 of related source clock frequency.
3. IrDA is available on all UARTs
4. UART0 contains the standard features plus ISO7816
5. UART0 contain 8-entry transmit and 8-entry receive FIFOs
6. All other UARTs contain a 1-entry transmit and receive FIFOs
7. CEA709.1-B (LON) is available in UART0

3.9.4.2 UART wakeup

The UART can be configured to generate an interrupt/wakeup on the first active edge that it receives.

3.9.4.3 UART interrupts

The UART has multiple sources of interrupt requests. However, some of these sources are OR'd together to generate a single interrupt request. See below for the mapping of the individual interrupt sources to the interrupt request:

The status interrupt combines the following interrupt sources:

Source	UART 0	UART 1	UART 2
Transmit data empty	x	x	x
Transmit complete	x	x	x
Idle line	x	x	x
Receive data full	x	x	x
LIN break detect	x	x	x
RxD pin active edge	x	x	x
Initial character detect	x	—	—

The error interrupt combines the following interrupt sources:

Source	UART 0	UART 1	UART 2
Receiver overrun	x	x	x
Noise flag	x	x	x
Framing error	x	x	x
Parity error	x	x	x
Transmitter buffer overflow	x	x	x
Receiver buffer underflow	x	x	x
Transmit threshold (ISO7816)	x	—	—
Receiver threshold (ISO7816)	x	—	—
Wait timer (ISO7816)	x	—	—
Character wait timer (ISO7816)	x	—	—
Block wait timer (ISO7816)	x	—	—
Guard time violation (ISO7816)	x	—	—

The LON status interrupt combines the following interrupt sources:

Source	UART 0	UART 1	UART 2
Wbase expire after beta1 time slots (LON)	x	—	—
Package received (LON)	x	—	—
Package transmitted (LON)	x	—	—
Package cycle time expired (LON)	x	—	—
Preamble start (LON)	x	—	—
Transmission fail (LON)	x	—	—

3.9.5 I^{2}S configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

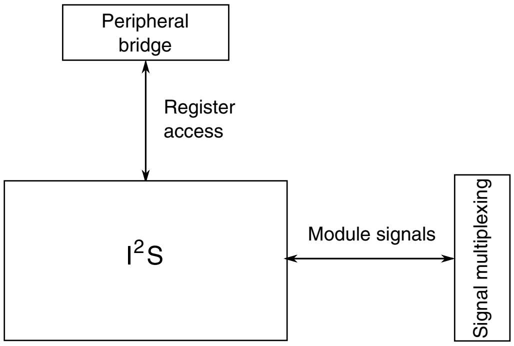


Figure 3-51. I^{2}S configuration

Table 3-62. Reference links to related information

Topic	Related module	Reference
Full description	I^{2}S	I2S
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management
Signal multiplexing	Port control	Signal Multiplexing

3.9.5.1 Instantiation information

This device contains one I^{2}S module.

As configured on the device, module features include:
- TX data lines: 1
- RX data lines: 1
- FIFO size (words): 4
- Maximum words per frame: 16
- Maximum bit clock divider: 512

3.9.5.2 I^{2}S/SAI clocking

3.9.5.2.1 Audio Master Clock

The audio master clock (MCLK) is used to generate the bit clock when the receiver or transmitter is configured for an internally generated bit clock. The audio master clock can also be output to or input from a pin. The transmitter and receiver have the same audio master clock inputs.

3.9.5.2.2 Bit Clock

The I^{2}S/SAI transmitter and receiver support asynchronous bit clocks (BCLKs) that can be generated internally from the audio master clock or supplied externally. The module also supports the option for synchronous operation between the receiver and transmitter or between two separate I^{2}S/SAI peripherals.

3.9.5.2.3 Bus Clock

The bus clock is used by the control registers and to generate synchronous interrupts and DMA requests.

3.9.5.2.4 I^{2}S/SAI clock generation

Each SAI peripheral can control the input clock selection, pin direction and divide ratio of one audio master clock.

The MCLK Input Clock Select bit of the MCLK Control Register (MCR[MICS]) selects the clock input to the I^{2}S/SAI module's MCLK divider.

The following table shows the input clock selection options on this device.

Table 3-63. I2S0 MCLK input clock selection

MCR[MICS]	Clock Selection
00	System clock
01	OSC0ERCLK
10	Not supported
11	MCGPLLCLK or MCGFLLCLK

The module's MCLK Divide Register (MDR) configures the MCLK divide ratio.

The module's MCLK Output Enable bit of the MCLK Control Register (MCR[MOE]) controls the direction of the MCLK pin. The pin is the input from the pin when MOE is 0, and the pin is the output from the clock divider when MOE is 1.

The transmitter and receiver can independently select between the bus clock and the audio master clock to generate the bit clock. Each module's Clocking Mode field of the Transmit Configuration 2 Register and Receive Configuration 2 Register (TCR2[MSEL] and RCR2[MSEL]) selects the master clock.

The following table shows the TCR2[MSEL] and RCR2[MSEL] field settings for this device.

Table 3-64. I2S0 master clock settings

TCR2[MSEL], RCR2[MSEL]	Master Clock
00	Bus Clock
01	I2S0_MCLK
10	Not supported
11	Not supported

3.9.5.2.5 Clock gating and I^{2}S/SAI initialization

The clock to the I^{2}S/SAI module can be gated using a bit in the SIM. To minimize power consumption, these bits are cleared after any reset, which disables the clock to the corresponding module. The clock enable bit should be set by software at the beginning of the module initialization routine to enable the module clock before initialization of any of the I^{2}S/SAI registers.

3.9.5.3 I^{2}S/SAI operation in low power modes

3.9.5.3.1 Stop and very low power modes

In Stop mode, the SAI transmitter and/or receiver can continue operating provided the appropriate Stop Enable bit is set (TCSR[STOPE] and/or RCSR[STOPE], respectively), and provided the transmitter and/or receiver is/are using an externally generated bit clock or an Audio Master Clock that remains operating in Stop mode. The SAI transmitter and/or receiver can generate an asynchronous interrupt to wake the CPU from Stop mode.

In VLPS mode, the module behaves as it does in stop mode if VLPS mode is entered from run mode. However, if VLPS mode is entered from VLPR mode, the FIFO might underflow or overflow before wakeup from stop mode due to the limits in bus bandwidth. In VLPW and VLPR modes, the module is limited by the maximum bus clock frequencies.

When operating from an internally generated bit clock or Audio Master Clock that is disabled in stop modes:

In Stop mode, if the Transmitter Stop Enable (TCSR[STOPE]) bit is clear, the transmitter is disabled after completing the current transmit frame, and, if the Receiver Stop Enable (RCSR[STOPE]) bit is clear, the receiver is disabled after completing the current receive frame. Entry into Stop mode is prevented–not acknowledged–while waiting for the transmitter and receiver to be disabled at the end of the current frame.

3.9.5.3.2 Low-leakage modes

When entering low-leakage modes, the Stop Enable (TCSR[STOPE] and RCSR[STOPE]) bits are ignored and the SAI is disabled after completing the current transmit and receive Frames. Entry into stop mode is prevented (not acknowledged) while waiting for the transmitter and receiver to be disabled at the end of the current frame.

3.10 Human-machine interfaces (HMI)

3.10.1 GPIO configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module’s dedicated chapter.

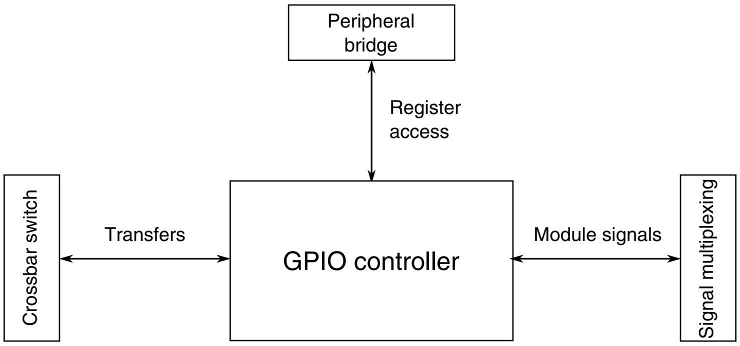


Figure 3-52. GPIO configuration

Table 3-65. Reference links to related information

Topic	Related module	Reference
Full description	GPIO	GPIO
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management

Table continues on the next page...

Table 3-65. Reference links to related information (continued)

Topic	Related module	Reference
Transfers	Crossbar switch	Clock Distribution
Signal Multiplexing	Port control	Signal Multiplexing

3.10.1.1 GPIO access protection

The GPIO module does not have access protection because it is not connected to a peripheral bridge slot.

3.10.1.2 Number of GPIO signals

The number of GPIO signals available on the devices covered by this document are detailed in Orderable part numbers.

3.10.2 TSI Configuration

This section summarizes how the module has been configured in the chip. For a comprehensive description of the module itself, see the module's dedicated chapter.

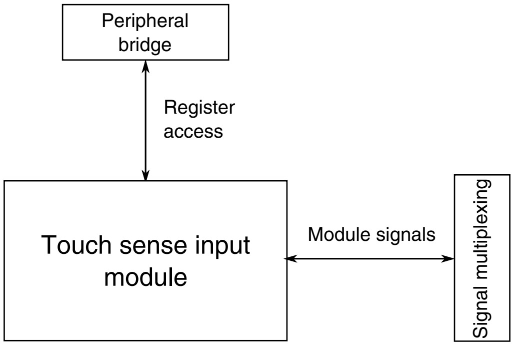


Figure 3-53. TSI configuration

Table 3-66. Reference links to related information

Topic	Related module	Reference
Full description	TSI	TSI
System memory map		System memory map
Clocking		Clock Distribution
Power management		Power management
Signal Multiplexing	Port control	Signal Multiplexing

3.10.2.1 Number of inputs

This device includes one TSI module containing 16 inputs. In low-power modes, one selectable pin is active.

3.10.2.2 TSI module functionality in MCU operation modes

Table 3-67. TSI module functionality in MCU operation modes

MCU operation mode	TSI clock sources	TSI operation mode when GENCS[TSIEN] is 1	Functional electrode pins	Required GENCS[STPE] state
Run	BUSCLK, MCGIRCLK, OSCERCLK	Active mode	All	Don't care
Wait	BUSCLK, MCGIRCLK, OSCERCLK	Active mode	All	Don't care
Stop	MCGIRCLK, OSCERCLK	Active mode	All	1
VLPR	BUSCLK, MCGIRCLK, OSCERCLK	Active mode	All	Don't care
VLPW	BUSCLK, MCGIRCLK, OSCERCLK	Active mode	All	Don't care
VLPS	OSCERCLK	Active mode	All	1
LLS	LPOCLK, VLPOSCCLK	Low power mode	Determined by PEN[LPSP]	1
VLLS3	LPOCLK, VLPOSCCLK	Low power mode	Determined by PEN[LPSP]	1
VLLS2	LPOCLK, VLPOSCCLK	Low power mode	Determined by PEN[LPSP]	1
VLLS1	LPOCLK, VLPOSCCLK	Low power mode	Determined by PEN[LPSP]	1
VLLS0	VLPOSCCLK[1]	Low power mode	Determined by PEN[LPSP]	1

1. This clock must be 32 kHz RTC.

3.10.2.3 TSI clocks

This table shows the TSI clocks and the corresponding chip clocks.

Table 3-68. TSI clock connections

Module clock	Chip clock
BUSCLK	Bus clock
MCGIRCLK	MCGIRCLK
OSCERCLK	OSCERCLK
LPOCLK	1 kHz LPO clock
VLPOSCCLK	ERCLK32K

3.10.2.4 TSI Interrupts

The TSI has multiple sources of interrupt requests. However, these sources are OR'd together to generate a single interrupt request. When a TSI interrupt occurs, read the TSI status register to determine the exact interrupt source.

3.10.2.5 Shield drive signal

The shield drive signal is not supported on this device. Ignore this feature in the TSI chapter.

Chapter 4
Memory Map

4.1 Introduction

This device contains various memories and memory-mapped peripherals which are located in one 32-bit contiguous memory space. This chapter describes the memory and peripheral locations within that memory space.

4.2 System memory map

The following table shows the high-level device memory map.

Table 4-1. System memory map

System 32-bit Address Range	Destination Slave	Access
0x0000_0000–0x07FF_FFFF	Program flash and read-only data (Includes exception vectors in first 1024 bytes)	All masters
0x0800_0000–0x0FFF_FFFF	Reserved	—
0x1000_0000–0x13FF_FFFF	• For MK20DN32VLH5: Reserved • For MK20DX32VLH5: FlexNVM • For MK20DN64VLH5: Reserved • For MK20DX64VLH5: FlexNVM • For MK20DN128VLH5: Reserved • For MK20DX128VLH5: FlexNVM • For MK20DN32VMP5: Reserved • For MK20DX32VMP5: FlexNVM • For MK20DN64VMP5: Reserved • For MK20DX64VMP5: FlexNVM • For MK20DN128VMP5: Reserved • For MK20DX128VMP5: FlexNVM	All masters
0x1400_0000–0x17FF_FFFF	FlexRAM	All masters
0x1800_0000–0x1BFF_FFFF	Reserved	—
0x1C00_0000–0x1FFF_FFFF	SRAM_L: Lower SRAM (ICODE/DCODE)	All masters
0x2000_0000–0x200F_FFFF	SRAM_U: Upper SRAM bitband region	All masters

Table continues on the next page...

Table 4-1. System memory map (continued)

System 32-bit Address Range	Destination Slave	Access
0x2010_0000–0x21FF_FFFF	Reserved	–
0x2200_0000–0x23FF_FFFF	Aliased to SRAM_U bitband	Cortex-M4 core only
0x2400_0000–0x3FFF_FFFF	Reserved	–
0x4000_0000–0x4007_FFFF	Bitband region for AIPS	Cortex-M4 core & DMA/EzPort
0x4008_0000–0x400F_EFFF	Reserved	–
0x400F_F000–0x400F_FFFF	Bitband region for general purpose input/output (GPIO)	Cortex-M4 core & DMA/EzPort
0x4010_0000–0x41FF_FFFF	Reserved	–
0x4200_0000–0x43FF_FFFF	Aliased to AIPS and GPIO bitband	Cortex-M4 core only
0x4400_0000–0xDFFF_FFFF	Reserved	–
0xE000_0000–0xE00F_FFFF	Private Peripherals	Cortex-M4 core only
0xE010_0000–0xFFFF_FFFF	Reserved	–

NOTE

1. EzPort master port is statically muxed with DMA master port. Access rights to AIPS-Lite peripheral bridge and general purpose input/output (GPIO) module address space is limited to the core, DMA, and EzPort.
2. ARM Cortex-M4 core access privileges also includes accesses via the debug interface.

4.2.1 Aliased bit-band regions

The SRAM_U, AIPS-Lite, and general purpose input/output (GPIO) module resources reside in the Cortex-M4 processor bit-band regions.

The processor also includes two 32 MB aliased bit-band regions associated with the two 1 MB bit-band spaces. Each 32-bit location in the 32 MB space maps to an individual bit in the bit-band region. A 32-bit write in the alias region has the same effect as a read-modify-write operation on the targeted bit in the bit-band region.

Bit 0 of the value written to the alias region determines what value is written to the target bit:

- Writing a value with bit 0 set writes a 1 to the target bit.
- Writing a value with bit 0 clear writes a 0 to the target bit.

A 32-bit read in the alias region returns either:

- a value of 0x0000_0000 to indicate the target bit is clear
- a value of 0x0000_0001 to indicate the target bit is set

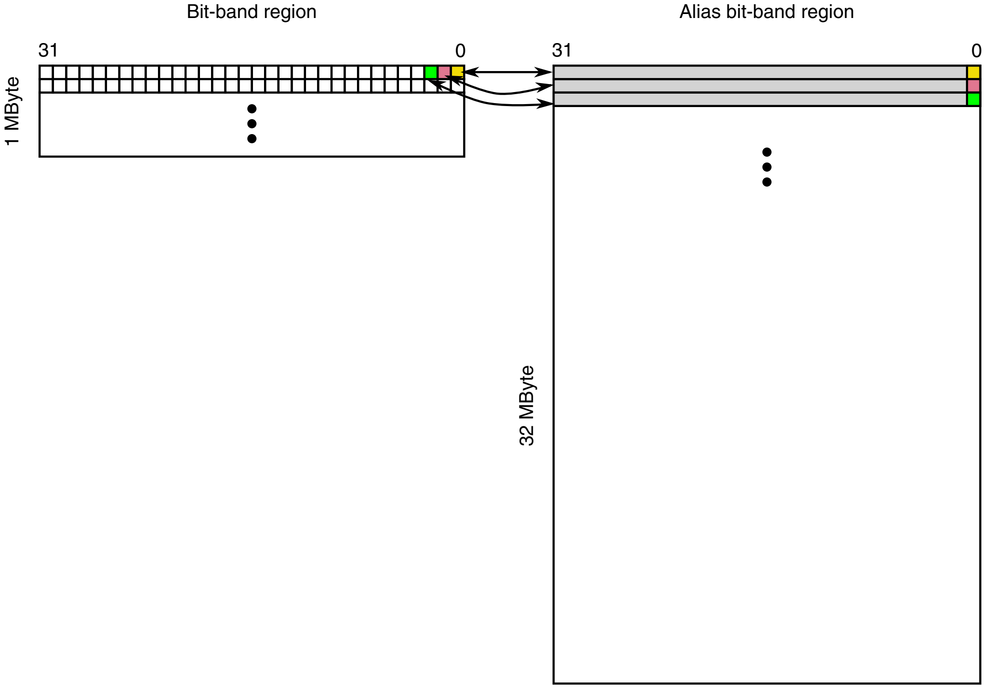


Figure 4-1. Alias bit-band mapping

NOTE

Each bit in bit-band region has an equivalent bit that can be manipulated through bit 0 in a corresponding long word in the alias bit-band region.

4.3 Flash Memory Map

The various flash memories and the flash registers are located at different base addresses as shown in the following figure. The base address for each is specified in System memory map.

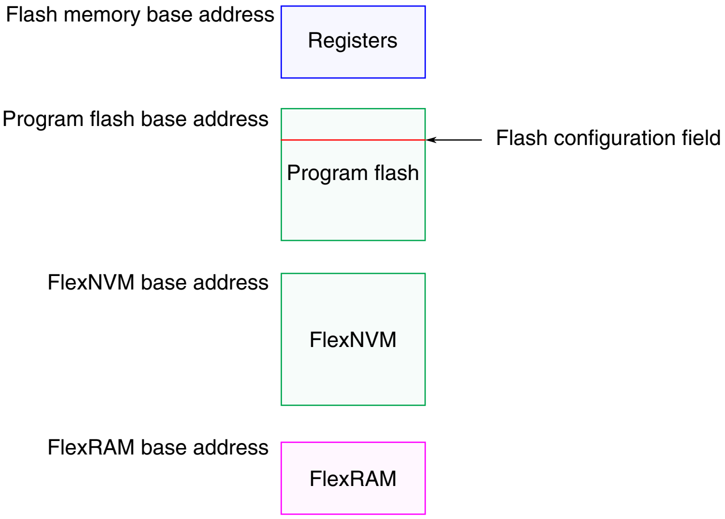

Flash memory base address
Registers
Program flash base address
Flash configuration field
Program flash
FlexNVM base address
FlexNVM
FlexRAM base address
FlexRAM

Figure 4-2. Flash memory map

4.3.1 Alternate Non-Volatile IRC User Trim Description

The following non-volatile locations (4 bytes) are reserved for custom IRC user trim supported by some development tools. An alternate IRC trim to the factory loaded trim can be stored at this location. To override the factory trim, user software must load new values into the MCG trim registers.

Non-Volatile Byte Address	Alternate IRC Trim Value
0x0000_03FC	Reserved
0x0000_03FD	Reserved
0x0000_03FE (bit 0)	SCFTRIM
0x0000_03FE (bit 4:1)	FCTRIM
0x0000_03FF	SCTRIM

4.4 SRAM memory map

The on-chip RAM is split evenly among SRAM_L and SRAM_U. The RAM is also implemented such that the SRAM_L and SRAM_U ranges form a contiguous block in the memory map. See SRAM Arrays for details.

Accesses to the SRAM_L and SRAM_U memory ranges outside the amount of RAM on the device causes the bus cycle to be terminated with an error followed by the appropriate response in the requesting bus master.

4.5 Peripheral bridge (AIPS-Lite) memory map

The peripheral memory map is accessible via one slave port on the crossbar in the 0x4000_0000–0x4007_FFFF region. The device implements one peripheral bridge that defines a 512 KB address space.

Modules that are disabled via their clock gate control bits in the SIM registers disable the associated AIPS slots. Access to any address within an unimplemented or disabled peripheral bridge slot results in a transfer error termination.

For programming model accesses via the peripheral bridges, there is generally only a small range within the 4 KB slots that is implemented. Accessing an address that is not implemented in the peripheral results in a transfer error termination.

4.5.1 Peripheral Bridge 0 (AIPS-Lite 0) Memory Map

Table 4-2. Peripheral bridge 0 slot assignments

System 32-bit base address	Slot number	Module
0x4000_0000	0	Peripheral bridge 0 (AIPS-Lite 0)
0x4000_1000	1	—
0x4000_2000	2	—
0x4000_3000	3	—
0x4000_4000	4	Crossbar switch
0x4000_5000	5	—
0x4000_6000	6	—
0x4000_7000	7	—
0x4000_8000	8	DMA controller
0x4000_9000	9	DMA controller transfer control descriptors
0x4000_A000	10	—
0x4000_B000	11	—
0x4000_C000	12	—
0x4000_D000	13	—
0x4000_E000	14	—
0x4000_F000	15	—
0x4001_0000	16	—
0x4001_1000	17	—

Table continues on the next page...

Table 4-2. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4001_2000	18	—
0x4001_3000	19	—
0x4001_4000	20	—
0x4001_5000	21	—
0x4001_6000	22	—
0x4001_7000	23	—
0x4001_8000	24	—
0x4001_9000	25	—
0x4001_A000	26	—
0x4001_B000	27	—
0x4001_C000	28	—
0x4001_D000	29	—
0x4001_E000	30	—
0x4001_F000	31	Flash memory controller
0x4002_0000	32	Flash memory
0x4002_1000	33	DMA channel mutiplexer 0
0x4002_2000	34	—
0x4002_3000	35	—
0x4002_4000	36	—
0x4002_5000	37	—
0x4002_6000	38	—
0x4002_7000	39	—
0x4002_8000	40	—
0x4002_9000	41	—
0x4002_A000	42	—
0x4002_B000	43	—
0x4002_C000	44	SPI 0
0x4002_D000	45	—
0x4002_E000	46	—
0x4002_F000	47	I2S 0
0x4003_0000	48	—
0x4003_1000	49	—
0x4003_2000	50	CRC
0x4003_3000	51	—

Table continues on the next page...

Table 4-2. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4003_4000	52	—
0x4003_5000	53	USB DCD
0x4003_6000	54	Programmable delay block (PDB)
0x4003_7000	55	Periodic interrupt timers (PIT)
0x4003_8000	56	FlexTimer (FTM) 0
0x4003_9000	57	FlexTimer (FTM) 1
0x4003_A000	58	—
0x4003_B000	59	Analog-to-digital converter (ADC) 0
0x4003_C000	60	—
0x4003_D000	61	Real-time clock (RTC)
0x4003_E000	62	VBAT register file
0x4003_F000	63	—
0x4004_0000	64	Low-power timer (LPTMR)
0x4004_1000	65	System register file
0x4004_2000	66	—
0x4004_3000	67	—
0x4004_4000	68	—
0x4004_5000	69	Touch sense interface (TSI)
0x4004_6000	70	—
0x4004_7000	71	SIM low-power logic
0x4004_8000	72	System integration module (SIM)
0x4004_9000	73	Port A multiplexing control
0x4004_A000	74	Port B multiplexing control
0x4004_B000	75	Port C multiplexing control
0x4004_C000	76	Port D multiplexing control
0x4004_D000	77	Port E multiplexing control
0x4004_E000	78	—
0x4004_F000	79	—
0x4005_0000	80	—
0x4005_1000	81	—
0x4005_2000	82	Software watchdog
0x4005_3000	83	—
0x4005_4000	84	—
0x4005_5000	85	—

Table continues on the next page...

Table 4-2. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4005_6000	86	—
0x4005_7000	87	—
0x4005_8000	88	—
0x4005_9000	89	—
0x4005_A000	90	—
0x4005_B000	91	—
0x4005_C000	92	—
0x4005_D000	93	—
0x4005_E000	94	—
0x4005_F000	95	—
0x4006_0000	96	—
0x4006_1000	97	External watchdog
0x4006_2000	98	Carrier modulator timer (CMT)
0x4006_3000	99	—
0x4006_4000	100	Multi-purpose Clock Generator (MCG)
0x4006_5000	101	System oscillator (OSC)
0x4006_6000	102	I^2C 0
0x4006_7000	103	—
0x4006_8000	104	—
0x4006_9000	105	—
0x4006_A000	106	UART 0
0x4006_B000	107	UART 1
0x4006_C000	108	UART 2
0x4006_D000	109	
0x4006_E000	110	—
0x4006_F000	111	—
0x4007_0000	112	—
0x4007_1000	113	—
0x4007_2000	114	USB OTG FS/LS
0x4007_3000	115	Analog comparator (CMP) / 6-bit digital-to-analog converter (DAC)
0x4007_4000	116	Voltage reference (VREF)
0x4007_5000	117	—
0x4007_6000	118	—
0x4007_7000	119	—

Table continues on the next page...

Table 4-2. Peripheral bridge 0 slot assignments (continued)

System 32-bit base address	Slot number	Module
0x4007_8000	120	—
0x4007_9000	121	—
0x4007_A000	122	—
0x4007_B000	123	—
0x4007_C000	124	Low-leakage wakeup unit (LLWU)
0x4007_D000	125	Power management controller (PMC)
0x4007_E000	126	System Mode controller (SMC)
0x4007_F000	127	Reset Control Module (RCM)
0x400F_F000		GPIO controller

4.6 Private Peripheral Bus (PPB) memory map

The PPB is part of the defined ARM bus architecture and provides access to select processor-local modules. These resources are only accessible from the core; other system masters do not have access to them.

Table 4-3. PPB memory map

System 32-bit Address Range	Resource
0xE000_0000–0xE000_0FFF	Instrumentation Trace Macrocell (ITM)
0xE000_1000–0xE000_1FFF	Data Watchpoint and Trace (DWT)
0xE000_2000–0xE000_2FFF	Flash Patch and Breakpoint (FPB)
0xE000_3000–0xE000_DFFF	Reserved
0xE000_E000–0xE000_EFFF	System Control Space (SCS) (for NVIC)
0xE000_F000–0xE003_FFFF	Reserved
0xE004_0000–0xE004_0FFF	Trace Port Interface Unit (TPIU)
0xE004_1000–0xE004_1FFF	Reserved
0xE004_2000–0xE004_2FFF	Reserved
0xE004_3000–0xE004_3FFF	Reserved
0xE004_4000–0xE007_FFFF	Reserved
0xE008_0000–0xE008_0FFF	Miscellaneous Control Module (MCM)
0xE008_1000–0xE008_1FFF	Reserved
0xE008_2000–0xE00F_EFFF	Reserved
0xE00F_F000–0xE00F_FFFF	ROM Table - allows auto-detection of debug components

Chapter 5
Clock Distribution

5.1 Introduction

The MCG module controls which clock source is used to derive the system clocks. The clock generation logic divides the selected clock source into a variety of clock domains, including the clocks for the system bus masters, system bus slaves, and flash memory. The clock generation logic also implements module-specific clock gating to allow granular shutoff of modules.

The primary clocks for the system are generated from the MCGOUTCLK clock. The clock generation circuitry provides several clock dividers that allow different portions of the device to be clocked at different frequencies. This allows for trade-offs between performance and power dissipation.

Various modules, such as the USB OTG Controller, have module-specific clocks that can be generated from the MCGPLLCLK or MCGFLLCLK clock. In addition, there are various other module-specific clocks that have other alternate sources. Clock selection for most modules is controlled by the SOPT registers in the SIM module.

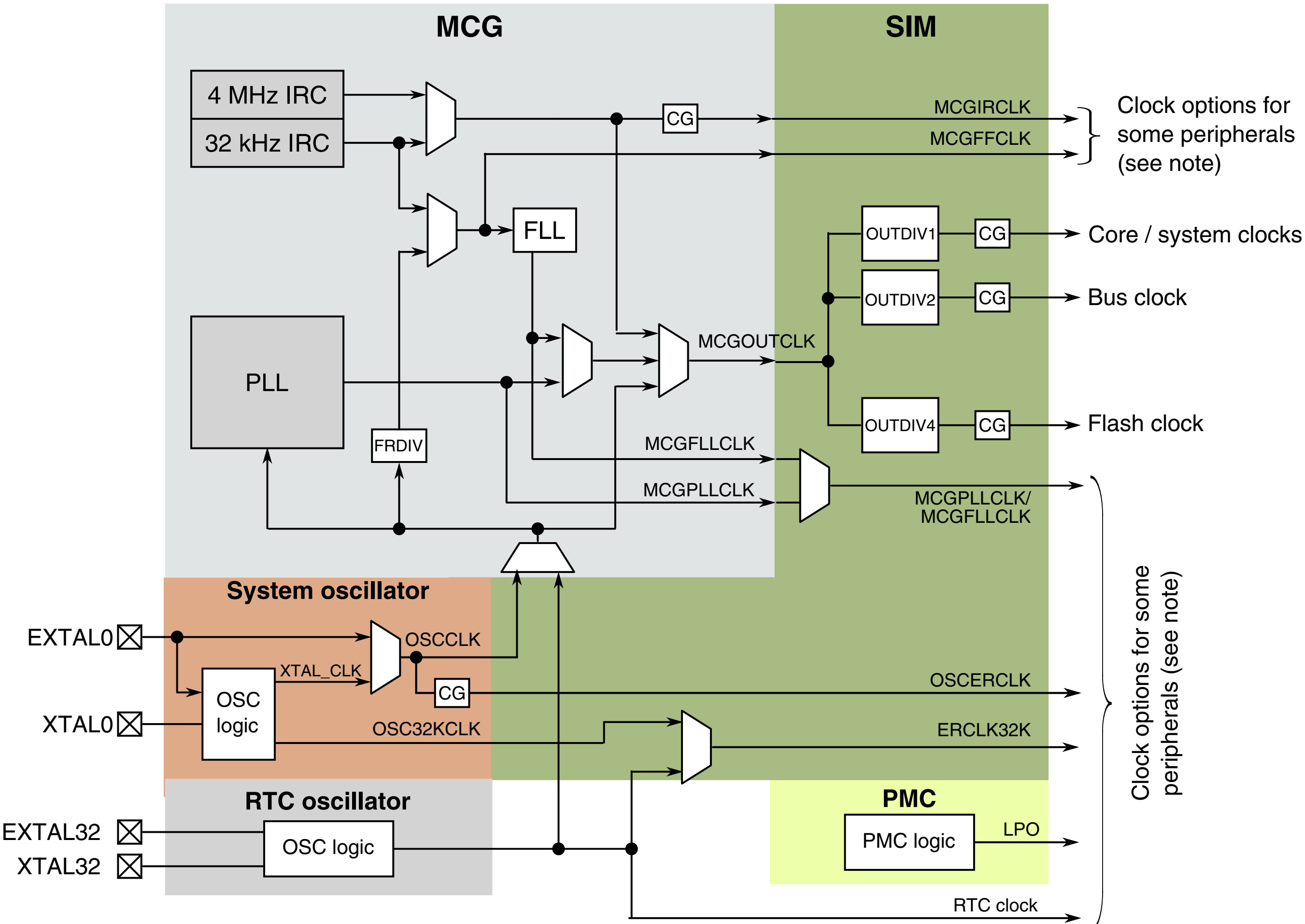
5.2 Programming model

The selection and multiplexing of system clock sources is controlled and programmed via the MCG module. The setting of clock dividers and module clock gating for the system are programmed via the SIM module. Reference those sections for detailed register and bit descriptions.

5.3 High-Level device clocking diagram

The following system oscillator, MCG, and SIM module registers control the multiplexers, dividers, and clock gates shown in the below figure:

	OSC	MCG	SIM
Multiplexers	MCG_C*x*	MCG_C*x*	SIM_SOPT1, SIM_SOPT2
Dividers	—	MCG_C*x*	SIM_CLKDIV*x*
Clock gates	OSC_CR	MCG_C1	SIM_SCGC*x*


MCG
SIM
4 MHz IRC
32 kHz IRC
CG
MCGIRCLK
MCGFFCLK
Clock options for some peripherals (see note)
FLL
OUTDIV1
CG
Core / system clocks
OUTDIV2
CG
Bus clock
MCGOUTCLK
PLL
FRDIV
OUTDIV4
CG
Flash clock
MCGFLLCLK
MCGPLLCLK
MCGPLLCLK/ MCGFLLCLK
System oscillator
EXTAL0
XTAL0
OSCCLK
XTAL_CLK
OSC logic
CG
OSCERCLK
OSC32KCLK
ERCLK32K
Clock options for some peripherals (see note)
RTC oscillator
EXTAL32
XTAL32
OSC logic
PMC
PMC logic
LPO
RTC clock

CG — Clock gate
Note: See subsequent sections for details on where these clocks are used.

Figure 5-1. Clocking diagram

5.4 Clock definitions

The following table describes the clocks in the previous block diagram.

Clock name	Description
Core clock	MCGOUTCLK divided by OUTDIV1 clocks the ARM Cortex-M4 core

Table continues on the next page...

Clock name	Description
System clock	MCGOUTCLK divided by OUTDIV1 clocks the crossbar switch and bus masters directly connected to the crossbar. In addition, this clock is used for UART0 and UART1.
Bus clock	MCGOUTCLK divided by OUTDIV2 clocks the bus slaves and peripheral (excluding memories)
Flash clock	MCGOUTCLK divided by OUTDIV4 clocks the flash memory
MCGIRCLK	MCG output of the slow or fast internal reference clock
MCGFFCLK	MCG output of the slow internal reference clock or a divided MCG external reference clock.
MCGOUTCLK	MCG output of either IRC, MCGFLLCLK, MCGPLLCLK, or MCG's external reference clock that sources the core, system, bus, and flash clock. It is also an option for the debug trace clock.
MCGFLLCLK	MCG output of the FLL. MCGFLLCLK or MCGPLLCLK may clock some modules.
MCGPLLCLK	MCG output of the PLL. MCGFLLCLK or MCGPLLCLK may clock some modules.
MCG external reference clock	Input clock to the MCG sourced by the system oscillator (OSCCLK) or RTC oscillator
OSCCLK	System oscillator output of the internal oscillator or sourced directly from EXTAL
OSCERCLK	System oscillator output sourced from OSCCLKthat may clock some on-chip modules
OSC32KCLK	System oscillator 32kHz output
ERCLK32K	Clock source for some modules that is chosen as OSC32KCLK or the RTC clock
RTC clock	RTC oscillator output for the RTC module
LPO	PMC 1kHz output

5.4.1 Device clock summary

The following table provides more information regarding the on-chip clocks.

Table 5-1. Clock Summary

Clock name	Run mode clock frequency	VLPR mode clock frequency	Clock source	Clock is disabled when…
MCGOUTCLK	Up to 100 MHz	Up to 4 MHz	MCG	In all stop modes
Core clock	Up to 50 MHz	Up to 4 MHz	MCGOUTCLK clock divider	In all wait and stop modes
System clock	Up to 50 MHz	Up to 4 MHz	MCGOUTCLK clock divider	In all stop modes
Bus clock	Up to 50 MHz	Up to 4 MHz	MCGOUTCLK clock divider	In all stop modes

Table continues on the next page...

Table 5-1. Clock Summary (continued)

Clock name	Run mode clock frequency	VLPR mode clock frequency	Clock source	Clock is disabled when…
Flash clock	Up to 25 MHz	Up to 1 MHz	MCGOUTCLK clock divider	In all stop modes
Internal reference (MCGIRCLK)	30-40 kHz or 4 MHz	4 MHz only	MCG	MCG_C1[IRCLKEN] cleared, Stop mode and MCG_C1[IREFSTEN] cleared, or VLPS/LLS/VLLS mode
External reference (OSCERCLK)	Up to 50 MHz (bypass), 30-40 kHz, or 3-32 MHz (crystal)	Up to 16 MHz (bypass), 30-40 kHz (low-range crystal) or Up to 16 MHz (high-range crystal)	System OSC	System OSC's OSC_CR[ERCLKEN] cleared, or Stop mode and OSC_CR[EREFSTEN] cleared
External reference 32kHz (ERCLK32K)	30-40 kHz	30-40 kHz	System OSC or RTC OSC depending on SIM_SOPT1[OSC32KSEL]	System OSC's OSC_CR[ERCLKEN] cleared or RTC's RTC_CR[OSCE] cleared
RTC_CLKOUT	1 Hz or 32 kHz	1 Hz or 32 kHz	RTC clock	Clock is disabled in LLS and VLLSx modes
LPO	1 kHz	1 kHz	PMC	in VLLS0
USB FS clock	48 MHz	N/A	MCGPLLCLK or MCGFLLCLK with fractional clock divider, or USB_CLKIN	USB FS OTG is disabled
I2S master clock	Up to 25 MHz	Up to 12.5 MHz	System clock, MCGPLLCLK, OSCERCLK with fractional clock divider, or I2S_CLKIN	I^2S is disabled
TRACE clock	Up to 50 MHz	Up to 4 MHz	System clock or MCGOUTCLK	Trace is disabled

5.5 Internal clocking requirements

The clock dividers are programmed via the SIM module's CLKDIV registers. Each divider is programmable from a divide-by-1 through divide-by-16 setting. The following requirements must be met when configuring the clocks for this device:

1. The core and system clock frequencies must be 50 MHz or slower.
2. The bus clock frequency must be programmed to 50 MHz or less and an integer divide of the core clock.
3. The flash clock frequency must be programmed to 25 MHz or less, less than or equal to the bus clock, and an integer divide of the core clock.

The following are a few of the more common clock configurations for this device:

Option 1:

Clock	Frequency
Core clock	50 MHz
System clock	50 MHz
Bus clock	50 MHz
Flash clock	25 MHz

5.5.1 Clock divider values after reset

Each clock divider is programmed via the SIM module's CLKDIV*n* registers. The flash memory's FTFL_FOPT[LPBOOT] bit controls the reset value of the core clock, system clock, bus clock, and flash clock dividers as shown below:

FTFL_FOPT [LPBOOT]	Core/system clock	Bus clock	Flash clock	Description
0	0x7 (divide by 8)	0x7 (divide by 8)	0xF (divide by 16)	Low power boot
1	0x0 (divide by 1)	0x0 (divide by 1)	0x1 (divide by 2)	Fast clock boot

This gives the user flexibility for a lower frequency, low-power boot option. The flash erased state defaults to fast clocking mode, since where the low power boot (FTFL_FOPT[LPBOOT]) bit resides in flash is logic 1 in the flash erased state.

To enable the low power boot option program FTFL_FOPT[LPBOOT] to zero. During the reset sequence, if LPBOOT is cleared, the system is in a slow clock configuration. Upon any system reset, the clock dividers return to this configurable reset state.

5.5.2 VLPR mode clocking

The clock dividers cannot be changed while in VLPR mode. They must be programmed prior to entering VLPR mode to guarantee:

- the core/system and bus clocks are less than or equal to 4 MHz, and
- the flash memory clock is less than or equal to 1 MHz

5.6 Clock Gating

The clock to each module can be individually gated on and off using the SIM module's SCGC*x* registers. These bits are cleared after any reset, which disables the clock to the corresponding module to conserve power. Prior to initializing a module, set the corresponding bit in SCGC*x* register to enable the clock. Before turning off the clock, make sure to disable the module.

Any bus access to a peripheral that has its clock disabled generates an error termination.

5.7 Module clocks

The following table summarizes the clocks associated with each module.

Table 5-2. Module clocks

Module	Bus interface clock	Internal clocks	I/O interface clocks
Core modules			
ARM Cortex-M4 core	System clock	Core clock	—
NVIC	System clock	—	—
DAP	System clock	—	—
ITM	System clock	—	—
cJTAG, JTAGC	—	—	JTAG_CLK
System modules			
DMA	System clock	—	—
DMA Mux	Bus clock	—	—
Port control	Bus clock	LPO	—
Crossbar Switch	System clock	—	—
Peripheral bridges	System clock	Bus clock, Flash clock	—
LLWU, PMC, SIM, RCM	Flash clock	LPO	—
Mode controller	Flash clock	—	—
MCM	System clock	—	—
EWM	Bus clock	LPO	—
Watchdog timer	Bus clock	LPO	—
Clocks			

Table continues on the next page...

Table 5-2. Module clocks (continued)

Module	Bus interface clock	Internal clocks	I/O interface clocks
MCG	Bus clock	MCGOUTCLK, MCGPLLCLK, MCGFLLCLK, MCGIRCLK, OSCERCLK, EXTAL32K	—
OSC	Bus clock	OSCERCLK	—
Memory and memory interfaces			
Flash Controller	System clock	Flash clock	—
Flash memory	Flash clock	—	—
EzPort	System clock	—	EZP_CLK
Security			
CRC	Bus clock	—	—
Analog			
ADC	Bus clock	OSCERCLK	—
CMP	Bus clock	—	—
VREF	Bus clock	—	—
Timers			
PDB	Bus clock	—	—
FlexTimers	Bus clock	MCGFFCLK	FTM_CLKIN*x*
PIT	Bus clock	—	—
LPTMR	Flash clock	LPO, OSCERCLK, MCGIRCLK, ERCLK32K	—
CMT	Bus clock	—	—
RTC	Flash clock	EXTAL32	—
Communication interfaces			
USB FS OTG	System clock	USB FS clock	—
USB DCD	Bus clock	—	—
DSPI	Bus clock	—	DSPI_SCK
I^{2}C	Bus clock	—	I2C_SCL
UART0, UART1	System clock	—	—
UART2	Bus clock	—	—
I^{2}S	Bus clock	I^{2}S master clock	I2S_TX_BCLK, I2S_RX_BCLK
Human-machine interfaces			
GPIO	System clock	—	—
TSI	Flash clock	LPO, ERCLK32K, MCGIRCLK	—

5.7.1 PMC 1-kHz LPO clock

The Power Management Controller (PMC) generates a 1-kHz clock that is enabled in all modes of operation, including all low power modes except VLLS0. This 1-kHz source is commonly referred to as LPO clock or 1-kHz LPO clock.

5.7.2 WDOG clocking

The WDOG may be clocked from two clock sources as shown in the following figure.

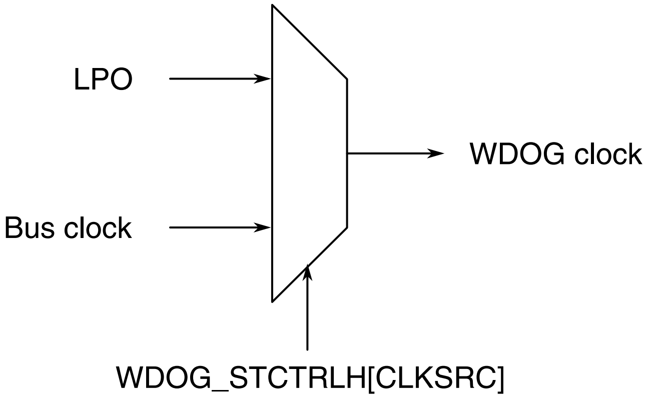

LPO
Bus clock
WDOG clock
WDOG_STCTRLH[CLKSRC]

Figure 5-2. WDOG clock generation

5.7.3 Debug trace clock

The debug trace clock source can be clocked as shown in the following figure.

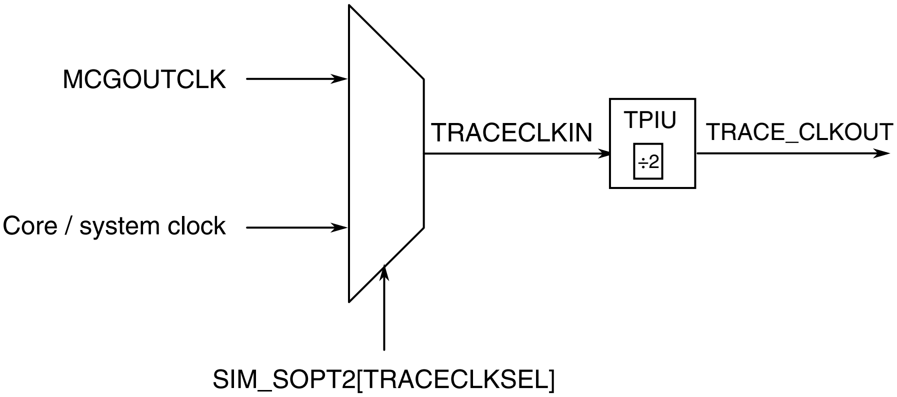

MCGOUTCLK
Core / system clock
TRACECLKIN
TPIU
÷2
TRACE_CLKOUT
SIM_SOPT2[TRACECLKSEL]

Figure 5-3. Trace clock generation

NOTE

The trace clock frequency observed at the TRACE_CLKOUT pin will be half that of the selected clock source.

5.7.4 PORT digital filter clocking

The digital filters in each of the PORT*x* modules can be clocked as shown in the following figure.

NOTE

In stop mode, the digital input filters are bypassed unless they are configured to run from the 1 kHz LPO clock source.

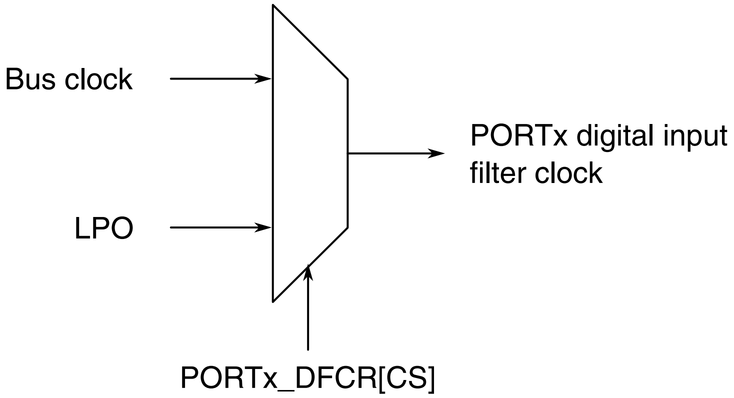


Figure 5-4. PORTx digital input filter clock generation

5.7.5 LPTMR clocking

The prescaler and glitch filters in each of the LPTMR*x* modules can be clocked as shown in the following figure.

NOTE

The chosen clock must remain enabled if the LPTMR*x* is to continue operating in all required low-power modes.

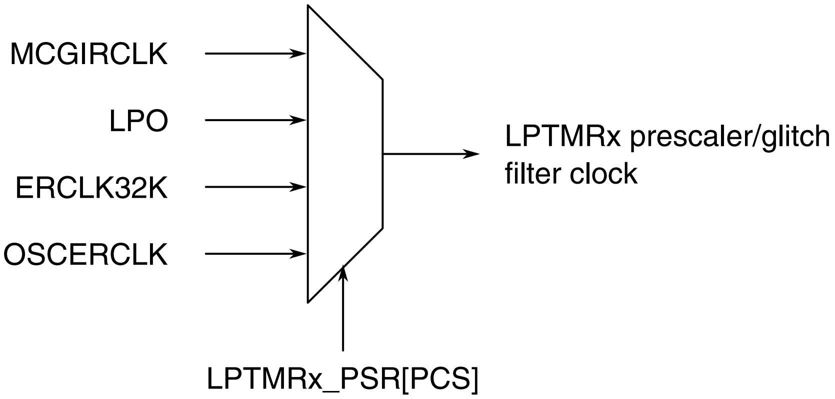

MCGIRCLK
LPO
ERCLK32K
OSCERCLK
LPTMRx prescaler/glitch filter clock
LPTMRx_PSR[PCS]

Figure 5-5. LPTMRx prescaler/glitch filter clock generation

5.7.6 USB FS OTG Controller clocking

The USB FS OTG controller is a bus master attached to the crossbar switch. As such, its clock is connected to the system clock.

NOTE

For the USB FS OTG controller to operate, the minimum system clock frequency is 20 MHz.

The USB OTG controller also requires a 48 MHz clock. The clock source options are shown below.

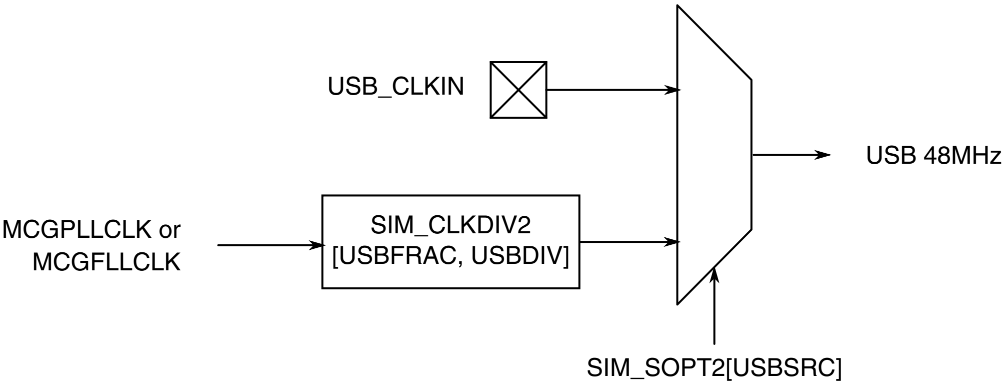

USB_CLKIN
USB 48MHz
MCGPLLCLK or MCGFLLCLK
SIM_CLKDIV2 [USBFRAC, USBDIV]
SIM_SOPT2[USBSRC]

Figure 5-6. USB 48 MHz clock source

NOTE

The MCGFLLCLK does not meet the USB jitter specifications for certification.

5.7.7 UART clocking

UART0 and UART1 modules operate from the core/system clock, which provides higher performance level for these modules. All other UART modules operate from the bus clock.

5.7.8 I^{2}S/SAI clocking

The audio master clock (MCLK) is used to generate the bit clock when the receiver or transmitter is configured for an internally generated bit clock. The audio master clock can also be output to or input from a pin. The transmitter and receiver have the same audio master clock inputs.

Each SAI peripheral can control the input clock selection, pin direction and divide ratio of one audio master clock.

The I^{2}S/SAI transmitter and receiver support asynchronous bit clocks (BCLKs) that can be generated internally from the audio master clock or supplied externally. The module also supports the option for synchronous operation between the receiver and transmitter or between two separate I^{2}S/SAI peripherals.

The transmitter and receiver can independently select between the bus clock and the audio master clock to generate the bit clock.

The MCLK and BCLK source options appear in the following figure.

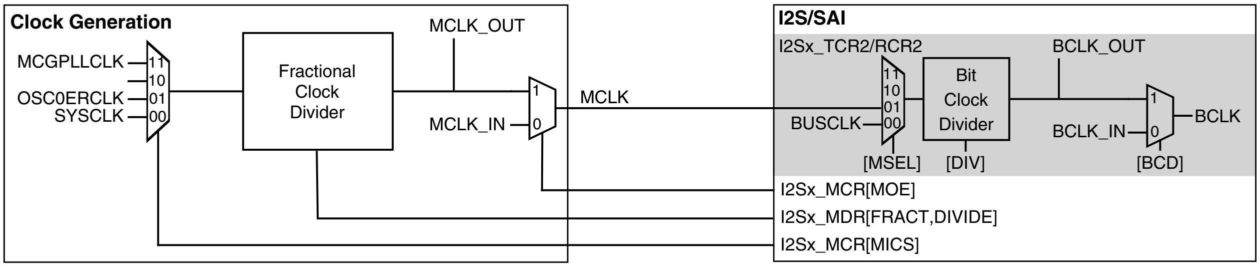

Clock Generation
MCGPLLCLK
OSC0ERCLK
SYSCLK
11
10
01
00
Fractional Clock Divider
MCLK_OUT
MCLK_IN
1
0
MCLK
I2S/SAI
I2Sx_TCR2/RCR2
BUSCLK
11
10
01
00
Bit Clock Divider
BCLK_OUT
BCLK_IN
1
0
BCLK
[MSEL]
[DIV]
[BCD]
I2Sx_MCR[MOE]
I2Sx_MDR[FRACT,DIVIDE]
I2Sx_MCR[MICS]

Figure 5-7. I^{2}S/SAI clock generation

5.7.9 TSI clocking

In active mode, the TSI can be clocked as shown in the following figure.

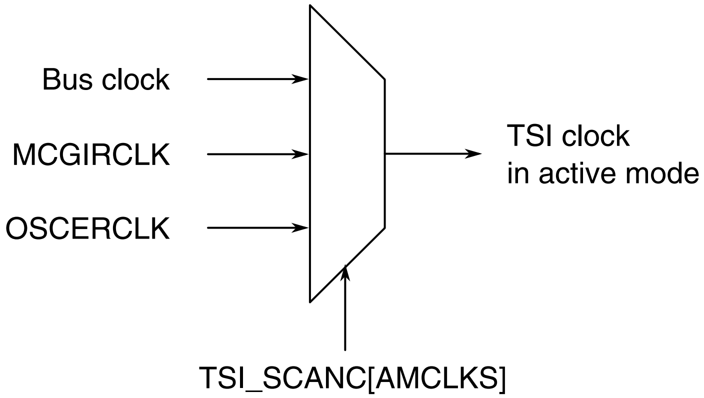

Bus clock
MCGIRCLK
OSCERCLK
TSI clock
in active mode
TSI_SCANC[AMCLKS]

Figure 5-8. TSI clock generation

In low-power mode, the TSI can be clocked as shown in the following figure.

NOTE

In the TSI chapter, these two clocks are referred to as LPOCLK and VLPOSCCLK.

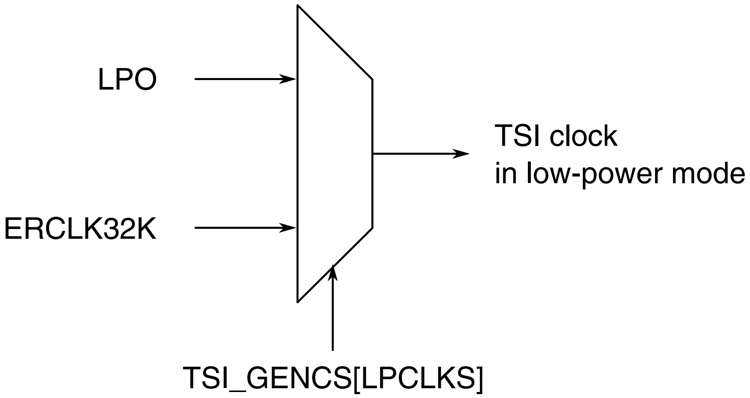

LPO
ERCLK32K
TSI clock
in low-power mode
TSI_GENCS[LPCLKS]

Figure 5-9. TSI low-power clock generation

Chapter 6
Reset and Boot

6.1 Introduction

The following reset sources are supported in this MCU:

Table 6-1. Reset sources

Reset sources	Description
POR reset	• Power-on reset (POR)
System resets	• External pin reset (PIN) • Low-voltage detect (LVD) • Computer operating properly (COP) watchdog reset • Low leakage wakeup (LLWU) reset • Multipurpose clock generator loss of clock (LOC) reset • Multipurpose clock generator loss of lock (LOL) reset • Stop mode acknowledge error (SACKERR) • Software reset (SW) • Lockup reset (LOCKUP) • EzPort reset • MDM DAP system reset
Debug reset	• JTAG reset • nTRST reset

Each of the system reset sources, with the exception of the EzPort and MDM-AP reset, has an associated bit in the system reset status (SRS) registers. See the Reset Control Module for register details.

The MCU exits reset in functional mode that is controlled by $\overline{\text{EZP\_CS}}$ pin to select between the single chip (default) or serial flash programming (EzPort) modes. See Boot options for more details.

6.2 Reset

This section discusses basic reset mechanisms and sources. Some modules that cause resets can be configured to cause interrupts instead. Consult the individual peripheral chapters for more information.

6.2.1 Power-on reset (POR)

When power is initially applied to the MCU or when the supply voltage drops below the power-on reset re-arm voltage level (V_{POR}), the POR circuit causes a POR reset condition.

As the supply voltage rises, the LVD circuit holds the MCU in reset until the supply has risen above the LVD low threshold (V_{LVDL}). The POR and LVD bits in SRS0 register are set following a POR.

6.2.2 System reset sources

Resetting the MCU provides a way to start processing from a known set of initial conditions. System reset begins with the on-chip regulator in full regulation and system clocking generation from an internal reference. When the processor exits reset, it performs the following:

- Reads the start SP (SP_main) from vector-table offset 0
- Reads the start PC from vector-table offset 4
- LR is set to 0xFFFF_FFFF

The on-chip peripheral modules are disabled and the non-analog I/O pins are initially configured as disabled. The pins with analog functions assigned to them default to their analog function after reset.

During and following a reset, the JTAG pins have their associated input pins configured as:

- TDI in pull-up (PU)
- TCK in pull-down (PD)
- TMS in PU

and associated output pin configured as:

- TDO with no pull-down or pull-up

Note that the nTRST signal is initially configured as disabled, however once configured to its JTAG functionality its associated input pin is configured as:

- nTRST in PU

6.2.2.1 External pin reset (PIN)

On this device, $\overline{\text{RESET}}$ is a dedicated pin. This pin is open drain and has an internal pullup device. Asserting $\overline{\text{RESET}}$ wakes the device from any mode. During a pin reset, the RCM's SRS0[PIN] bit is set.

6.2.2.1.1 Reset pin filter

The $\overline{\text{RESET}}$ pin filter supports filtering from both the 1 kHz LPO clock and the bus clock. A separate filter is implemented for each clock source. In stop and VLPS mode operation, this logic either switches to bypass operation or has continued filtering operation depending on the filtering mode selected. In low leakage stop modes, a separate LPO filter in the LLWU can continue filtering the $\overline{\text{RESET}}$ pin.

The RPFC[RSTFLTSS], RPFC[RSTFLTSRW], and RPFW[RSTFLTSEL] fields in the reset control (RCM) register set control this functionality; see the RCM chapter. The filters are asynchronously reset by Chip POR. The reset value for each filter assumes the RESET pin is negated.

The two clock options for the $\overline{\text{RESET}}$ pin filter when the chip is not in low leakage modes are the LPO (1 kHz) and bus clock. For low leakage modes VLLS3, VLLS2, VLLS1, the LLWU provides control (in the LLWU_RST register) of an optional fixed digital filter running the LPO. When entering VLLS0, the RESET pin filter is disabled and bypassed.

The LPO filter has a fixed filter value of 3. Due to a synchronizer on the input data, there is also some associated latency (2 cycles). As a result, 5 cycles are required to complete a transition from low to high or high to low.

The bus filter initializes to off (logic 1) when the bus filter is not enabled. The bus clock is used when the filter selects bus clock, and the number of counts is controlled by the RCM's RPFW[RSTFLTSEL] field.

6.2.2.2 Low-voltage detect (LVD)

The chip includes a system for managing low voltage conditions to protect memory contents and control MCU system states during supply voltage variations. The system consists of a power-on reset (POR) circuit and an LVD circuit with a user-selectable trip voltage. The LVD system is always enabled in normal run, wait, or stop mode. The LVD system is disabled when entering VLPx, LLS, or VLLSx modes.

The LVD can be configured to generate a reset upon detection of a low voltage condition by setting the PMC's LVDSC1[LVDRE] bit to 1. The low voltage detection threshold is determined by the PMC's LVDSC1[LVDV] field. After an LVD reset has occurred, the LVD system holds the MCU in reset until the supply voltage has risen above the low voltage detection threshold. The RCM's SRS0[LVD] bit is set following either an LVD reset or POR.

6.2.2.3 Computer operating properly (COP) watchdog timer

The computer operating properly (COP) watchdog timer (WDOG) monitors the operation of the system by expecting periodic communication from the software. This communication is generally known as servicing (or refreshing) the COP watchdog. If this periodic refreshing does not occur, the watchdog issues a system reset. The COP reset causes the RCM's SRS0[WDOG] bit to set.

6.2.2.4 Low leakage wakeup (LLWU)

The LLWU module provides the means for a number of external pins, the $\overline{\text{RESET}}$ pin, and a number of internal peripherals to wake the MCU from low leakage power modes. The LLWU module is functional only in low leakage power modes.

- In LLS mode, only the $\overline{\text{RESET}}$ pin via the LLWU can generate a system reset.
- In VLLSx modes, all enabled inputs to the LLWU can generate a system reset.

After a system reset, the LLWU retains the flags indicating the input source of the last wakeup until the user clears them.

NOTE

> Some flags are cleared in the LLWU and some flags are required to be cleared in the peripheral module. Refer to the individual peripheral chapters for more information.

6.2.2.5 Multipurpose clock generator loss-of-clock (LOC)

The MCG module supports an external reference clock.

If the C6[CME] bit in the MCG module is set, the clock monitor is enabled. If the external reference falls below f_{loc_low} or f_{loc_high}, as controlled by the C2[RANGE] field in the MCG module, the MCU resets. The RCM's SRS0[LOC] bit is set to indicate this reset source.

NOTE

To prevent unexpected loss of clock reset events, all clock monitors should be disabled before entering any low power modes, including VLPR and VLPW.

6.2.2.6 MCG loss-of-lock (LOL) reset

The MCG includes a PLL loss-of-lock detector. The detector is enabled when configured for PEE and lock has been achieved. If the MCG_C8[LOLRE] bit in the MCG module is set and the PLL lock status bit (MCG_S[LOLS0]) becomes set, the MCU resets. The RCM_SRS0[LOL] bit is set to indicate this reset source.

NOTE

This reset source does not cause a reset if the chip is in any stop mode.

6.2.2.7 Stop mode acknowledge error (SACKERR)

This reset is generated if the core attempts to enter stop mode, but not all modules acknowledge stop mode within 1025 cycles of the 1 kHz LPO clock.

A module might not acknowledge the entry to stop mode if an error condition occurs. The error can be caused by a failure of an external clock input to a module.

6.2.2.8 Software reset (SW)

The SYSRESETREQ bit in the NVIC application interrupt and reset control register can be set to force a software reset on the device. (See ARM's NVIC documentation for the full description of the register fields, especially the VECTKEY field requirements.) Setting SYSRESETREQ generates a software reset request. This reset forces a system reset of all major components except for the debug module. A software reset causes the RCM's SRS1[SW] bit to set.

6.2.2.9 Lockup reset (LOCKUP)

The LOCKUP gives immediate indication of seriously errant kernel software. This is the result of the core being locked because of an unrecoverable exception following the activation of the processor's built in system state protection hardware.

The LOCKUP condition causes a system reset and also causes the RCM's SRS1[LOCKUP] bit to set.

6.2.2.10 EzPort reset

The EzPort supports a system reset request via EzPort signaling. The EzPort generates a system reset request following execution of a Reset Chip (RESET) command via the EzPort interface. This method of reset allows the chip to boot from flash memory after it has been programmed by an external source. The EzPort is enabled or disabled by the $\overline{\text{EZP\_CS}}$ pin.

An EzPort reset causes the RCM's SRS1[EZPT] bit to set.

6.2.2.11 MDM-AP system reset request

Set the system reset request bit in the MDM-AP control register to initiate a system reset. This is the primary method for resets via the JTAG/SWD interface. The system reset is held until this bit is cleared.

Set the core hold reset bit in the MDM-AP control register to hold the core in reset as the rest of the chip comes out of system reset.

6.2.3 MCU Resets

A variety of resets are generated by the MCU to reset different modules.

6.2.3.1 VBAT POR

The VBAT POR asserts on a VBAT POR reset source. It affects only the modules within the VBAT power domain: RTC and VBAT Register File. These modules are not affected by the other reset types.

6.2.3.2 POR Only

The POR Only reset asserts on the POR reset source only. It resets the PMC and System Register File.

The POR Only reset also causes all other reset types (except VBAT POR) to occur.

6.2.3.3 Chip POR not VLLS

The Chip POR not VLLS reset asserts on POR and LVD reset sources. It resets parts of the SMC and SIM. It also resets the LPTMR.

The Chip POR not VLLS reset also causes these resets to occur: Chip POR, Chip Reset not VLLS, and Chip Reset (including Early Chip Reset).

6.2.3.4 Chip POR

The Chip POR asserts on POR, LVD, and VLLS Wakeup reset sources. It resets the Reset Pin Filter registers and parts of the SIM and MCG.

The Chip POR also causes the Chip Reset (including Early Chip Reset) to occur.

6.2.3.5 Chip Reset not VLLS

The Chip Reset not VLLS reset asserts on all reset sources except a VLLS Wakeup that does not occur via the $\overline{\text{RESET}}$ pin. It resets parts of the SMC, LLWU, and other modules that remain powered during VLLS mode.

The Chip Reset not VLLS reset also causes the Chip Reset (including Early Chip Reset) to occur.

6.2.3.6 Early Chip Reset

The Early Chip Reset asserts on all reset sources. It resets only the flash memory module. It negates before flash memory initialization begins ("earlier" than when the Chip Reset negates).

6.2.3.7 Chip Reset

Chip Reset asserts on all reset sources and only negates after flash initialization has completed and the $\overline{\text{RESET}}$ pin has also negated. It resets the remaining modules (the modules not reset by other reset types).

6.2.4 Reset Pin

For all reset sources except a VLLS Wakeup that does not occur via the $\overline{\text{RESET}}$ pin, the $\overline{\text{RESET}}$ pin is driven low by the MCU for at least 128 bus clock cycles and until flash initialization has completed.

After flash initialization has completed, the $\overline{\text{RESET}}$ pin is released, and the internal Chip Reset negates after the $\overline{\text{RESET}}$ pin is pulled high. Keeping the $\overline{\text{RESET}}$ pin asserted externally delays the negation of the internal Chip Reset.

6.2.5 Debug resets

The following sections detail the debug resets available on the device.

6.2.5.1 JTAG reset

The JTAG module generate a system reset when certain IR codes are selected. This functional reset is asserted when EzPort, EXTEST, HIGHZ and CLAMP instructions are active. The reset source from the JTAG module is released when any other IR code is selected. A JTAG reset causes the RCM's SRS1[JTAG] bit to set.

6.2.5.2 nTRST reset

The nTRST pin causes a reset of the JTAG logic when asserted. Asserting the nTRST pin allows the debugger to gain control of the TAP controller state machine (after exiting LLS or VLLSx) without resetting the state of the debug modules.

The nTRST pin does not cause a system reset.

6.2.5.3 Resetting the Debug subsystem

Use the CDBGRSTREQ bit within the SWJ-DP CTRL/STAT register to reset the debug modules. However, as explained below, using the CDBGRSTREQ bit does not reset all debug-related registers.

CDBGRSTREQ resets the debug-related registers within the following modules:

- SWJ-DP
- AHB-AP
- TPIU
- MDM-AP (MDM control and status registers)

CDBGRSTREQ does not reset the debug-related registers within the following modules:

- CM4 core (core debug registers: DHCSR, DCRSR, DCRDR, DEMCR)
- FPB
- DWT
- ITM
- NVIC
- Crossbar bus switch[1]
- AHB-AP[1]
- Private peripheral bus[1]

6.3 Boot

This section describes the boot sequence, including sources and options.

6.3.1 Boot sources

This device only supports booting from internal flash. Any secondary boot must go through an initialization sequence in flash.

6.3.2 Boot options

The device's functional mode is controlled by the state of the EzPort chip select ($\overline{\text{EZP\_CS}}$) pin during reset.

---

[1] CDBGRSTREQ does not affect AHB resources so that debug resources on the private peripheral bus are available during System Reset.

The device can be in single chip (default) or serial flash programming mode (EzPort). While in single chip mode the device can be in run or various low power modes mentioned in Power mode transitions.

Table 6-2. Mode select decoding

EzPort chip select ($\overline{\text{EZP\_CS}}$)	Description
0	Serial flash programming mode (EzPort)
1	Single chip (default)

6.3.3 FOPT boot options

The flash option register (FOPT) in flash memory module (FTFL) allows the user to customize the operation of the MCU at boot time. The register contains read-only bits that are loaded from the NVM's option byte in the flash configuration field. The user can reprogram the option byte in flash to change the FOPT values that are used for subsequent resets. For more details on programming the option byte, refer to the flash memory chapter.

The MCU uses the FTFL_FOPT register bits to configure the device at reset as shown in the following table.

Table 6-3. Flash Option Register (FTFL_FOPT) Bit Definitions

Bit Num	Field	Value	Definition
7-3	Reserved		Reserved for future expansion.
2	NMI_DIS	0	NMI interrupts are always blocked. The associated pin continues to default to NMI pin controls with internal pullup enabled.
		1	NMI pin/interrupts reset default to enabled.
1	EZPORT_DIS	0	EzPort operation is disabled. The device always boots to normal CPU execution and the state of $\overline{\text{EZP\_CS}}$ signal during reset is ignored. This option avoids inadvertent resets into EzPort mode if the $\overline{\text{EZP\_CS}}$/NMI pin is used for its NMI function.
		1	EzPort operation is enabled. The state of $\overline{\text{EZP\_CS}}$ pin during reset determines if device enters EzPort mode.

Table continues on the next page...

Table 6-3. Flash Option Register (FTFL_FOPT) Bit Definitions (continued)

Bit Num	Field	Value	Definition
0	LPBOOT	0	Low-power boot: OUTDIVx values in SIM_CLKDIV1 register are auto-configured at reset exit for higher divide values that produce lower power consumption at reset exit. • Core and system clock divider (OUTDIV1) and bus clock divider (OUTDIV2) are 0x7 (divide by 8) • Flash clock divider (OUTDIV4)is 0xF (divide by 16)
		1	Normal boot: OUTDIVx values in SIM_CLKDIV1 register are auto-configured at reset exit for higher frequency values that produce faster operating frequencies at reset exit. • Core and system clock divider (OUTDIV1) and bus clock divider (OUTDIV2) are 0x0 (divide by 1) • Flash clock divider (OUTDIV4)is 0x1 (divide by 2)

6.3.4 Boot sequence

At power up, the on-chip regulator holds the system in a POR state until the input supply is above the POR threshold. The system continues to be held in this static state until the internally regulated supplies have reached a safe operating voltage as determined by the LVD. The Mode Controller reset logic then controls a sequence to exit reset.

1. A system reset is held on internal logic, the $\overline{\text{RESET}}$ pin is driven out low, and the MCG is enabled in its default clocking mode.
2. Required clocks are enabled (Core Clock, System Clock, Flash Clock, and any Bus Clocks that do not have clock gate control).
3. The system reset on internal logic continues to be held, but the Flash Controller is released from reset and begins initialization operation while the Mode Control logic continues to drive the $\overline{\text{RESET}}$ pin out low for a count of ~128 Bus Clock cycles.
4. The $\overline{\text{RESET}}$ pin is released, but the system reset of internal logic continues to be held until the Flash Controller finishes initialization. EzPort mode is selected instead of the normal CPU execution if $\overline{\text{EZP\_CS}}$ is low when the internal reset is deasserted. EzPort mode can be disabled by programming the FOPT[EZPORT_DIS] field in the Flash Memory module.
5. When Flash Initialization completes, the $\overline{\text{RESET}}$ pin is observed. If $\overline{\text{RESET}}$ continues to be asserted (an indication of a slow rise time on the $\overline{\text{RESET}}$ pin or external drive in low), the system continues to be held in reset. Once the $\overline{\text{RESET}}$ pin is detected high, the system is released from reset.
6. At release of system reset, clocking is switched to a slow clock if the FOPT[LPBOOT] field in the Flash Memory module is configured for Low Power Boot

7. When the system exits reset, the processor sets up the stack, program counter (PC), and link register (LR). The processor reads the start SP (SP_main) from vector-table offset 0. The core reads the start PC from vector-table offset 4. LR is set to 0xFFFF_FFFF. The CPU begins execution at the PC location. EzPort mode is entered instead of the normal CPU execution if Ezport mode was latched during the sequence.
8. If FlexNVM is enabled, the flash controller continues to restore the FlexNVM data. This data is not available immediately out of reset and the system should not access this data until the flash controller completes this initialization step as indicated by the EEERDY flag.

Subsequent system resets follow this reset flow beginning with the step where system clocks are enabled.

Chapter 7
Power Management

7.1 Introduction

This chapter describes the various chip power modes and functionality of the individual modules in these modes.

7.2 Power modes

The power management controller (PMC) provides multiple power options to allow the user to optimize power consumption for the level of functionality needed.

Depending on the stop requirements of the user application, a variety of stop modes are available that provide state retention, partial power down or full power down of certain logic and/or memory. I/O states are held in all modes of operation. The following table compares the various power modes available.

For each run mode there is a corresponding wait and stop mode. Wait modes are similar to ARM sleep modes. Stop modes (VLPS, STOP) are similar to ARM sleep deep mode. The very low power run (VLPR) operating mode can drastically reduce runtime power when the maximum bus frequency is not required to handle the application needs.

The three primary modes of operation are run, wait and stop. The WFI instruction invokes both wait and stop modes for the chip. The primary modes are augmented in a number of ways to provide lower power based on application needs.

Table 7-1. Chip power modes

Chip mode	Description	Core mode	Normal recovery method
Normal run	Allows maximum performance of chip. Default mode out of reset; on-chip voltage regulator is on.	Run	-

Table continues on the next page...

Table 7-1. Chip power modes (continued)

Chip mode	Description	Core mode	Normal recovery method
Normal Wait - via WFI	Allows peripherals to function while the core is in sleep mode, reducing power. NVIC remains sensitive to interrupts; peripherals continue to be clocked.	Sleep	Interrupt
Normal Stop - via WFI	Places chip in static state. Lowest power mode that retains all registers while maintaining LVD protection. NVIC is disabled; AWIC is used to wake up from interrupt; peripheral clocks are stopped.	Sleep Deep	Interrupt
VLPR (Very Low Power Run)	On-chip voltage regulator is in a low power mode that supplies only enough power to run the chip at a reduced frequency. Reduced frequency Flash access mode (1 MHz); LVD off; internal oscillator provides a low power 4 MHz source for the core, the bus and the peripheral clocks.	Run	Interrupt
VLPW (Very Low Power Wait) -via WFI	Same as VLPR but with the core in sleep mode to further reduce power; NVIC remains sensitive to interrupts (FCLK = ON). On-chip voltage regulator is in a low power mode that supplies only enough power to run the chip at a reduced frequency.	Sleep	Interrupt
VLPS (Very Low Power Stop)-via WFI	Places chip in static state with LVD operation off. Lowest power mode with ADC and pin interrupts functional. Peripheral clocks are stopped, but LPTimer, RTC, CMP, TSI can be used. NVIC is disabled (FCLK = OFF); AWIC is used to wake up from interrupt. On-chip voltage regulator is in a low power mode that supplies only enough power to run the chip at a reduced frequency. All SRAM is operating (content retained and I/O states held).	Sleep Deep	Interrupt
LLS (Low Leakage Stop)	State retention power mode. Most peripherals are in state retention mode (with clocks stopped), but LLWU, LPTimer, RTC, CMP, TSI can be used. NVIC is disabled; LLWU is used to wake up. **NOTE:** The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit stop mode on an LLS recovery. All SRAM is operating (content retained and I/O states held).	Sleep Deep	Wakeup Interrupt[1]
VLLS3 (Very Low Leakage Stop3)	Most peripherals are disabled (with clocks stopped), but LLWU, LPTimer, RTC, CMP, TSI can be used. NVIC is disabled; LLWU is used to wake up. SRAM_U and SRAM_L remain powered on (content retained and I/O states held).	Sleep Deep	Wakeup Reset[2]
VLLS2 (Very Low Leakage Stop2)	Most peripherals are disabled (with clocks stopped), but LLWU, LPTimer, RTC, CMP, TSI can be used. NVIC is disabled; LLWU is used to wake up. SRAM_L is powered off. A portion of SRAM_U remains powered on (content retained and I/O states held).	Sleep Deep	Wakeup Reset[2]
VLLS1 (Very Low Leakage Stop1)	Most peripherals are disabled (with clocks stopped), but LLWU, LPTimer, RTC, CMP, TSI can be used. NVIC is disabled; LLWU is used to wake up. All of SRAM_U and SRAM_L are powered off. The 32-byte system register file and the 32-byte VBAT register file remain powered for customer-critical data.	Sleep Deep	Wakeup Reset[2]

Table continues on the next page...

Table 7-1. Chip power modes (continued)

Chip mode	Description	Core mode	Normal recovery method
VLLS0 (Very Low Leakage Stop 0)	Most peripherals are disabled (with clocks stopped), but LLWU and RTC can be used. NVIC is disabled; LLWU is used to wake up. All of SRAM_U and SRAM_L are powered off. The 32-byte system register file and the 32-byte VBAT register file remain powered for customer-critical data. The POR detect circuit can be optionally powered off.	Sleep Deep	Wakeup Reset[2]
BAT (backup battery only)	The chip is powered down except for the VBAT supply. The RTC and the 32-byte VBAT register file for customer-critical data remain powered.	Off	Power-up Sequence

1. Resumes normal run mode operation by executing the LLWU interrupt service routine.
2. Follows the reset flow with the LLWU interrupt flag set for the NVIC.

7.3 Entering and exiting power modes

The WFI instruction invokes wait and stop modes for the chip. The processor exits the low-power mode via an interrupt. The Nested Vectored Interrupt Controller (NVIC) describes interrupt operation and what peripherals can cause interrupts.

NOTE

The WFE instruction can have the side effect of entering a low-power mode, but that is not its intended usage. See ARM documentation for more on the WFE instruction.

Recovery from VLLSx is through the wake-up Reset event. The chip wake-ups from VLLSx by means of reset, an enabled pin or enabled module. See the table "LLWU inputs" in the LLWU configuration section for a list of the sources.

The wake-up flow from VLLSx is through reset. The wakeup bit in the SRS registers in the RCM is set indicating that the chip is recovering from a low power mode. Code execution begins; however, the I/O pins are held in their pre low power mode entry states, and the system oscillator and MCG registers are reset (even if EREFSTEN had been set before entering VLLSx). Software must clear this hold by writing a 1 to the ACKISO bit in the Regulator Status and Control Register in the PMC module.

NOTE

To avoid unwanted transitions on the pins, software must re-initialize the I/O pins to their pre-low-power mode entry states *before* releasing the hold.

If the oscillator was configured to continue running during VLLSx modes, it must be re-configured before the ACKISO bit is cleared. The oscillator configuration within the MCG is cleared after VLLSx recovery and the oscillator will stop when ACKISO is cleared unless the register is re-configured.

7.4 Power mode transitions

The following figure shows the power mode transitions. Any reset always brings the chip back to the normal run state. In run, wait, and stop modes active power regulation is enabled. The VLPx modes are limited in frequency, but offer a lower power operating mode than normal modes. The LLS and VLLSx modes are the lowest power stop modes based on amount of logic or memory that is required to be retained by the application.

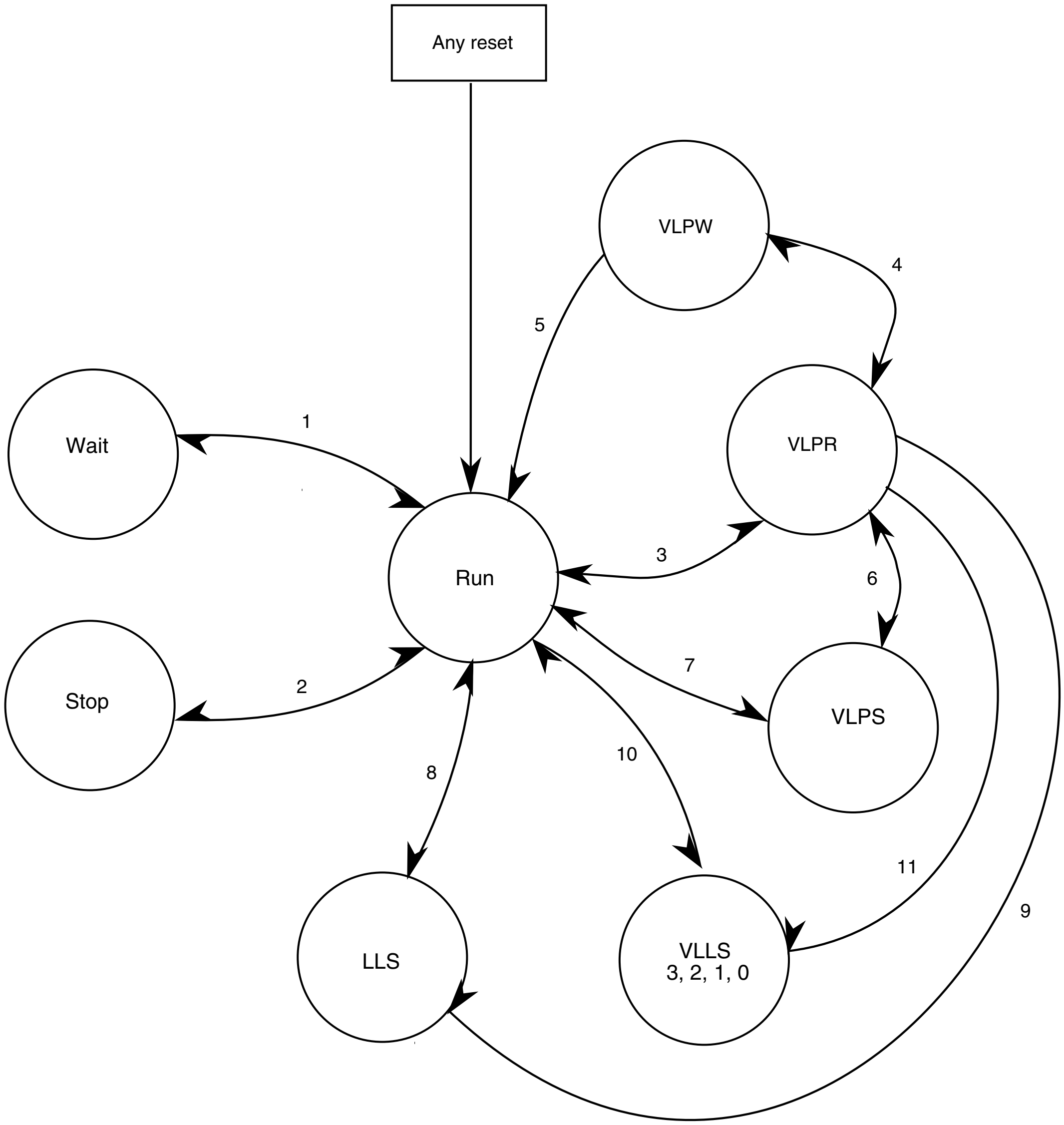


Figure 7-1. Power mode state transition diagram

7.5 Power modes shutdown sequencing

When entering stop or other low-power modes, the clocks are shut off in an orderly sequence to safely place the chip in the targeted low-power state. All low-power entry sequences are initiated by the core executing an WFI instruction. The ARM core's outputs, SLEEPDEEP and SLEEPING, trigger entry to the various low-power modes:

- System level wait and VLPW modes equate to: SLEEPING & $\overline{\text{SLEEPDEEP}}$
- All other low power modes equate to: SLEEPING & SLEEPDEEP

When entering the non-wait modes, the chip performs the following sequence:

- Shuts off Core Clock and System Clock to the ARM Cortex-M4 core immediately.
- Polls stop acknowledge indications from the non-core crossbar masters (DMA), supporting peripherals (SPI, PIT) and the Flash Controller for indications that System Clocks, Bus Clock and/or Flash Clock need to be left enabled to complete a previously initiated operation, effectively stalling entry to the targeted low power mode. When all acknowledges are detected, System Clock, Bus Clock and Flash Clock are turned off at the same time.
- MCG and Mode Controller shut off clock sources and/or the internal supplies driven from the on-chip regulator as defined for the targeted low power mode.

In wait modes, most of the system clocks are not affected by the low power mode entry. The Core Clock to the ARM Cortex-M4 core is shut off. Some modules support stop-in-wait functionality and have their clocks disabled under these configurations.

The debugger modules support a transition from stop, wait, VLPS, and VLPW back to a halted state when the debugger is enabled. This transition is initiated by setting the Debug Request bit in MDM-AP control register. As part of this transition, system clocking is re-established and is equivalent to normal run/VLPR mode clocking configuration.

7.6 Module Operation in Low Power Modes

The following table illustrates the functionality of each module while the chip is in each of the low power modes. (Debug modules are discussed separately; see Debug in Low Power Modes.) Number ratings (such as 2 MHz and 1 Mbps) represent the maximum frequencies or maximum data rates per mode. Also, these terms are used:

- FF = Full functionality. In VLPR and VLPW the system frequency is limited, but if a module does not have a limitation in its functionality, it is still listed as FF.
- static = Module register states and associated memories are retained.
- powered = Memory is powered to retain contents.
- low power = Flash has a low power state that retains configuration registers to support faster wakeup.
- OFF = Modules are powered off; module is in reset state upon wakeup.
- wakeup = Modules can serve as a wakeup source for the chip.

Table 7-2. Module operation in low power modes

Modules	Stop	VLPR	VLPW	VLPS	LLS	VLLSx
Core modules						
NVIC	static	FF	FF	static	static	OFF
System modules						

Table continues on the next page...

Table 7-2. Module operation in low power modes (continued)

Modules	Stop	VLPR	VLPW	VLPS	LLS	VLLSx
Mode Controller	FF	FF	FF	FF	FF	FF
LLWU[1]	static	static	static	static	FF	FF[2]
Regulator	ON	low power	low power	low power	low power	low power in VLLS2/3, OFF in VLLS0/1
LVD	ON	disabled	disabled	disabled	disabled	disabled
Brown-out Detection	ON	ON	ON	ON	ON	ON in VLLS1/2/3, optionally disabled in VLLS0[3]
DMA	static	FF	FF	static	static	OFF
Watchdog	FF	FF	FF	FF	static	OFF
EWM	static	FF	static	static	static	OFF
Clocks						
1kHz LPO	ON	ON	ON	ON	ON	ON in VLLS1/2/3, OFF in VLLS0
System oscillator (OSC)	OSCERCLK optional	OSCERCLK max of 4MHz crystal	OSCERCLK max of 4MHz crystal	OSCERCLK max of 4MHz crystal	limited to low range/low power	limited to low range/low power in VLLS1/2/3, OFF in VLLS0
MCG	static - MCGIRCLK optional; PLL optionally on but gated	4 MHz IRC	4 MHz IRC	static - no clock output	static - no clock output	OFF
Core clock	OFF	4 MHz max	OFF	OFF	OFF	OFF
System clock	OFF	4 MHz max	4 MHz max	OFF	OFF	OFF
Bus clock	OFF	4 MHz max	4 MHz max	OFF	OFF	OFF
Memory and memory interfaces						
Flash	powered	1 MHz max access - no pgm	low power	low power	OFF	OFF
Portion of SRAM_U[4]	low power	low power	low power	low power	low power	low power in VLLS3,2
Remaining SRAM_U and all of SRAM_L	low power	low power	low power	low power	low power	low power in VLLS3
FlexMemory	low power	low power[5]	low power	low power	low power	OFF
Register files[6]	powered	powered	powered	powered	powered	powered
EzPort	disabled	disabled	disabled	disabled	disabled	disabled

Table continues on the next page...

Table 7-2. Module operation in low power modes (continued)

Modules	Stop	VLPR	VLPW	VLPS	LLS	VLLSx
Communication interfaces						
USB FS/LS	static	static	static	static	static	OFF
USB DCD	static	FF	FF	static	static	OFF
USB Voltage Regulator	optional	optional	optional	optional	optional	optional
UART	static, wakeup on edge	125 kbps	125 kbps	static, wakeup on edge	static	OFF
SPI	static	1 Mbps	1 Mbps	static	static	OFF
I^2C	static, address match wakeup	100 kbps	100 kbps	static, address match wakeup	static	OFF
I^2S	FF with external clock[7]	FF	FF	FF with external clock[7]	static	OFF
Security						
CRC	static	FF	FF	static	static	OFF
Timers						
FTM	static	FF	FF	static	static	OFF
PIT	static	FF	FF	static	static	OFF
PDB	static	FF	FF	static	static	OFF
LPTMR	FF	FF	FF	FF	FF	FF[8]
RTC - 32kHz OSC[6]	FF	FF	FF	FF	FF[9]	FF[9]
CMT	static	FF	FF	static	static	OFF
Analog						
16-bit ADC	ADC internal clock only	FF	FF	ADC internal clock only	static	OFF
CMP[10]	HS or LS compare	FF	FF	HS or LS compare	LS compare	LS compare in VLLS1/2/3, OFF in VLLS0
6-bit DAC	static	FF	FF	static	static	static
VREF	FF	FF	FF	FF	static	OFF
Human-machine interfaces						
GPIO	wakeup	FF	FF	wakeup	static, pins latched	OFF, pins latched
TSI	wakeup	FF	FF	wakeup	wakeup[11]	wakeup[11, 12]

1. Using the LLWU module, the external pins available for this chip do not require the associated peripheral function to be enabled. It only requires the function controlling the pin (GPIO or peripheral) to be configured as an input to allow a transition to occur to the LLWU.
2. Since LPO clock source is disabled, filters will be bypassed during VLLS0
3. The VLLSCTRL[PORPO] bit in the SMC module controls this option.
4. A 8KB portion of SRAM_U block is left powered on in low power mode VLLS2.
5. FlexRAM enabled as EEPROM is not writable in VLPR and writes are ignored. Read accesses to FlexRAM as EEPROM while in VLPR are allowed. There are no access restrictions for FlexRAM configured as traditional RAM.

6. These components remain powered in BAT power mode.
7. Use an externally generated bit clock or an externally generated audio master clock (including EXTAL).
8. System OSC and LPO clock sources are not available in VLLS0
9. RTC_CLKOUT is not available.
10. CMP in stop or VLPS supports high speed or low speed external pin to pin or external pin to DAC compares. CMP in LLS or VLLSx only supports low speed external pin to pin or external pin to DAC compares. Windowed, sampled & filtered modes of operation are not available while in stop, VLPS, LLS, or VLLSx modes.
11. TSI wakeup from LLS and VLLSx modes is limited to a single selectable pin.
12. System OSC and LPO clock sources are not available in VLLS0

7.7 Clock Gating

To conserve power, the clocks to most modules can be turned off using the SCGCx registers in the SIM module. These bits are cleared after any reset, which disables the clock to the corresponding module. Prior to initializing a module, set the corresponding bit in the SCGCx register to enable the clock. Before turning off the clock, make sure to disable the module. For more details, refer to the clock distribution and SIM chapters.

Chapter 8
Security

8.1 Introduction

This device implements security based on the mode selected from the flash module. The following sections provide an overview of flash security and details the effects of security on non-flash modules.

8.2 Flash Security

The flash module provides security information to the MCU based on the state held by the FSEC[SEC] bits. The MCU, in turn, confirms the security request and limits access to flash resources. During reset, the flash module initializes the FSEC register using data read from the security byte of the flash configuration field.

NOTE

> The security features apply only to external accesses: debug and EzPort. CPU accesses to the flash are not affected by the status of FSEC.

In the unsecured state all flash commands are available to the programming interfaces (JTAG and EzPort), as well as user code execution of Flash Controller commands. When the flash is secured (FSEC[SEC] = 00, 01, or 11), programmer interfaces are only allowed to launch mass erase operations and have no access to memory locations.

Further information regarding the flash security options and enabling/disabling flash security is available in the Flash Memory Module.

8.3 Security Interactions with other Modules

The flash security settings are used by the SoC to determine what resources are available. The following sections describe the interactions between modules and the flash security settings or the impact that the flash security has on non-flash modules.

8.3.1 Security Interactions with EzPort

When flash security is active the MCU can still boot in EzPort mode. The EzPort holds the flash logic in NVM special mode and thus limits flash operation when flash security is active. While in EzPort mode and security is active, flash bulk erase (BE) can still be executed. The write FCCOB registers (WRFCCOB) command is limited to the mass erase (Erase All Blocks) and verify all 1s (Read 1s All Blocks) commands. Read accesses to internal memories via the EzPort are blocked when security is enabled.

The mass erase can be used to disable flash security, but all of the flash contents are lost in the process. A mass erase via the EzPort is allowed even when some memory locations are protected.

When mass erase has been disabled, mass erase via the EzPort is blocked and cannot be defeated.

8.3.2 Security Interactions with Debug

When flash security is active the JTAG port cannot access the memory resources of the MCU. Boundary scan chain operations work, but debugging capabilities are disabled so that the debug port cannot read flash contents.

Although most debug functions are disabled, the debugger can write to the Flash Mass Erase in Progress bit in the MDM-AP Control register to trigger a mass erase (Erase All Blocks) command. A mass erase via the debugger is allowed even when some memory locations are protected.

When mass erase is disabled, mass erase via the debugger is blocked.

Chapter 9
Debug

9.1 Introduction

This device's debug is based on the ARM coresight architecture and is configured in each device to provide the maximum flexibility as allowed by the restrictions of the pinout and other available resources.

Four debug interfaces are supported:

- IEEE 1149.1 JTAG
- IEEE 1149.7 JTAG (cJTAG)
- Serial Wire Debug (SWD)
- ARM Real-Time Trace Interface(1-pin asynchronous mode only)

The basic Cortex-M4 debug architecture is very flexible. The following diagram shows the topology of the core debug architecture and its components.

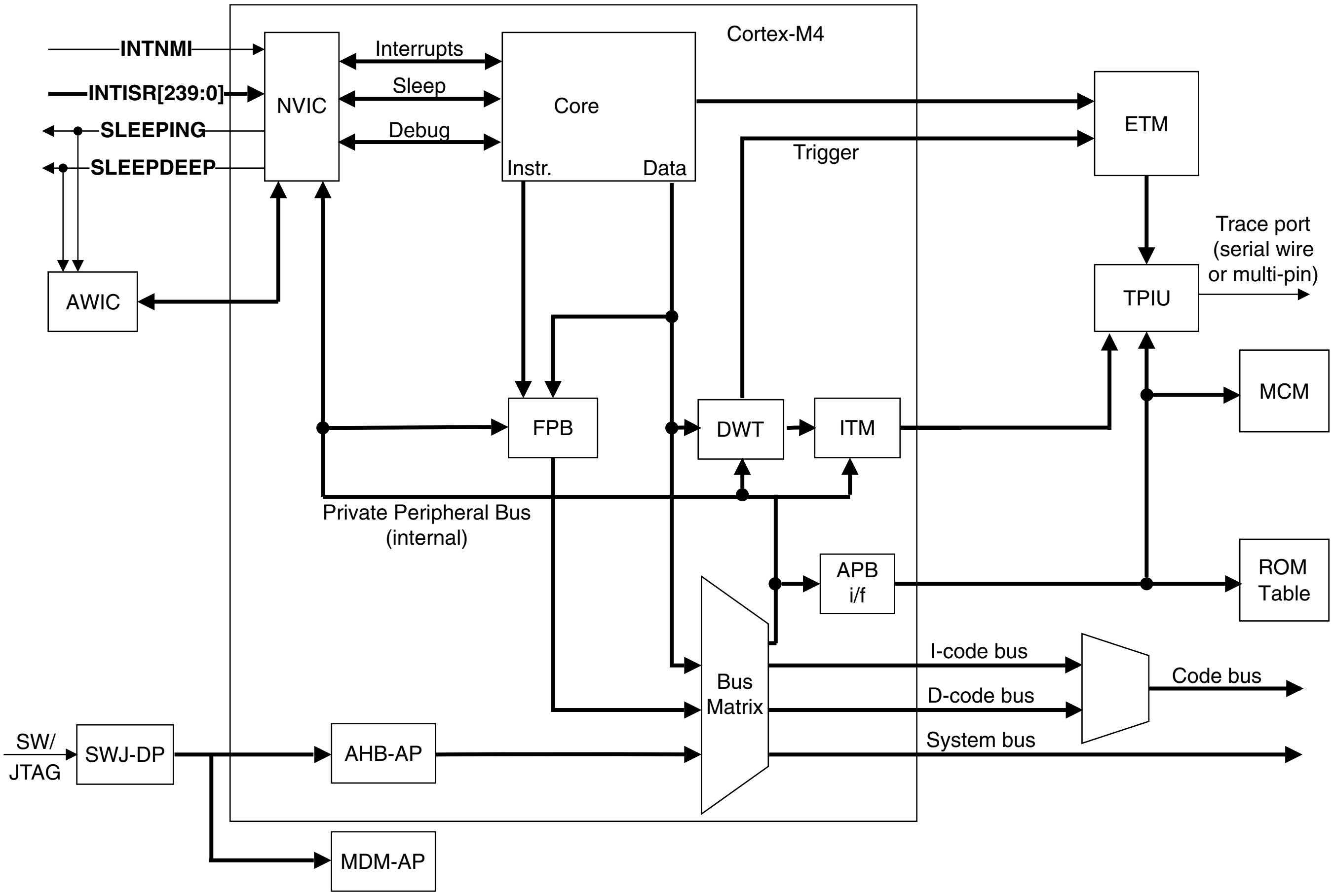

INTNMI
INTISR[239:0]
SLEEPING
SLEEPDEEP
NVIC
Interrupts
Sleep
Debug
Core
Instr.
Data
Cortex-M4
ETM
Trigger
TPIU
Trace port
(serial wire
or multi-pin)
AWIC
FPB
DWT
ITM
MCM
Private Peripheral Bus
(internal)
APB
i/f
ROM
Table
I-code bus
Code bus
Bus
Matrix
D-code bus
System bus
SW/
JTAG
SWJ-DP
AHB-AP
MDM-AP

Figure 9-1. Cortex-M4 Debug Topology

The following table presents a brief description of each one of the debug components.

Table 9-1. Debug Components Description

Module	Description
SWJ-DP+ cJTAG	Modified Debug Port with support for SWD, JTAG, cJTAG
AHB-AP	AHB Master Interface from JTAG to debug module and SOC system memory maps
JTAG-AP	Bridge to DFT/BIST resources.
ROM Table	Identifies which debug IP is available.
Core Debug	Singlestep, Register Access, Run, Core Status
ITM	S/W Instrumentation Messaging + Simple Data Trace Messaging + Watchpoint Messaging
DWT (Data and Address Watchpoints)	4 data and address watchpoints (configurable for less, but 4 seems to be accepted)

Table continues on the next page...

Table 9-1. Debug Components Description (continued)

Module	Description
FPB (Flash Patch and Breakpoints)	The FPB implements hardware breakpoints and patches code and data from code space to system space. The FPB unit contains two literal comparators for matching against literal loads from Code space, and remapping to a corresponding area in System space. The FBP also contains six instruction comparators for matching against instruction fetches from Code space, and remapping to a corresponding area in System space. Alternatively, the six instruction comparators can individually configure the comparators to return a Breakpoint Instruction (BKPT) to the processor core on a match, so providing hardware breakpoint capability.
TPIU (Trace Port Inteface Unit)	Asynchronous Mode (1-pin) = TRACE_SWO (available on JTAG_TDO)

9.1.1 References

For more information on ARM debug components, see these documents:

- ARMv7-M Architecture Reference Manual
- ARM Debug Interface v5.1
- ARM CoreSight Architecture Specification

9.2 The Debug Port

The configuration of the cJTAG module, JTAG controller, and debug port is illustrated in the following figure:

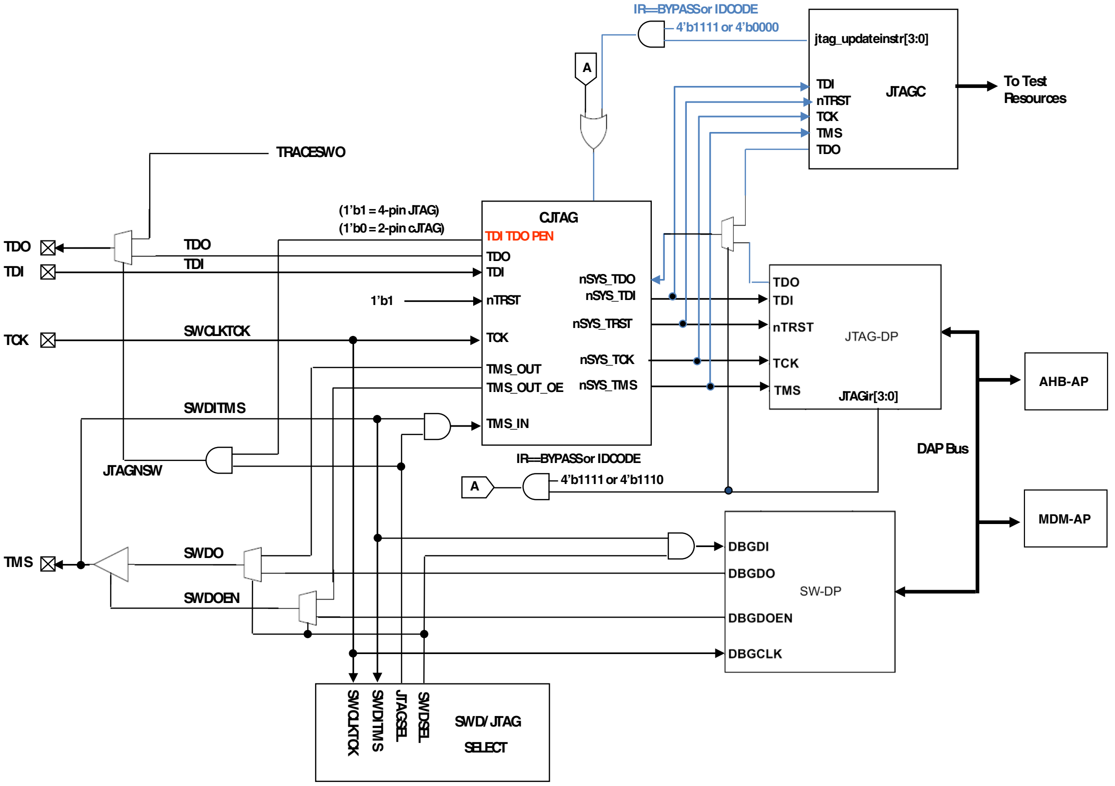

IR==BYPASS or IDCODE
4'b1111 or 4'b0000
jtag_updateinstr[3:0]
A
TDI
nTRST
TCK
TMS
TDO
JTAGC
To Test Resources
TRACESWO
(1'b1 = 4-pin JTAG)
(1'b0 = 2-pin cJTAG)
CJTAG
TDI TDO PEN
TDO
TDI
TDO
TDI
1'b1
nTRST
nSYS_TDO
nSYS_TDI
nSYS_TRST
TCK
SWCLKTCK
TCK
TMS_OUT
TMS_OUT_OE
nSYS_TCK
nSYS_TMS
SWDITMS
TMS_IN
JTAGNSW
IR==BYPASS or IDCODE
A
4'b1111 or 4'b1110
TDO
TDI
nTRST
JTAG-DP
TCK
TMS
JTAGir[3:0]
AHB-AP
DAP Bus
MDM-AP
TMS
SWDO
SWDOEN
DBGDI
DBGDO
SW-DP
DBGDOEN
DBGCLK
SWCLKTCK
SWDITMS
JTAGSEL
SWDSEL
SWD/JTAG SELECT

Figure 9-2. Modified Debug Port

The debug port comes out of reset in standard JTAG mode and is switched into either cJTAG or SWD mode by the following sequences. Once the mode has been changed, unused debug pins can be reassigned to any of their alternative muxed functions.

9.2.1 JTAG-to-SWD change sequence

1. Send more than 50 TCK cycles with TMS (SWDIO) =1
2. Send the 16-bit sequence on TMS (SWDIO) = 0111_1001_1110_0111 (MSB transmitted first)
3. Send more than 50 TCK cycles with TMS (SWDIO) =1

NOTE

See the ARM documentation for the CoreSight DAP Lite for restrictions.

9.2.2 JTAG-to-cJTAG change sequence

1. Reset the debug port

2. Set the control level to 2 via zero-bit scans
3. Execute the Store Format (STFMT) command (00011) to set the scan format register to 1149.7 scan format

9.3 Debug Port Pin Descriptions

The debug port pins default after POR to their JTAG functionality with the exception of JTAG_TRST_b and can be later reassigned to their alternate functionalities. In cJTAG and SWD modes JTAG_TDI and JTAG_TRST_b can be configured to alternate GPIO functions.

Table 9-2. Debug port pins

Pin Name	JTAG Debug Port		cJTAG Debug Port		SWD Debug Port		Internal Pull-up\Down
	Type	**Description**	**Type**	**Description**	**Type**	**Description**	
JTAG_TMS/ SWD_DIO	I/O	JTAG Test Mode Selection	I/O	cJTAG Data	I/O	Serial Wire Data	Pull-up
JTAG_TCLK/ SWD_CLK	I	JTAG Test Clock	I	cJTAG Clock	I	Serial Wire Clock	Pull-down
JTAG_TDI	I	JTAG Test Data Input	-	-	-	-	Pull-up
JTAG_TDO/ TRACE_SWO	O	JTAG Test Data Output	O	Trace output over a single pin	O	Trace output over a single pin	N/C
JTAG_TRST_b	I	JTAG Reset	I	cJTAG Reset	-	-	Pull-up

9.4 System TAP connection

The system JTAG controller is connected in parallel to the ARM TAP controller. The system JTAG controller IR codes overlay the ARM JTAG controller IR codes without conflict. Refer to the IR codes table for a list of the available IR codes. The output of the TAPs (TDO) are muxed based on the IR code which is selected. This design is fully JTAG compliant and appears to the JTAG chain as a single TAP. At power on reset, ARM's IDCODE (IR=4'b1110) is selected.

9.4.1 IR Codes

Table 9-3. JTAG Instructions

Instruction	Code[3:0]	Instruction Summary
IDCODE	0000	Selects device identification register for shift
SAMPLE/PRELOAD	0010	Selects boundary scan register for shifting, sampling, and preloading without disturbing functional operation
SAMPLE	0011	Selects boundary scan register for shifting and sampling without disturbing functional operation
EXTEST	0100	Selects boundary scan register while applying preloaded values to output pins and asserting functional reset
HIGHZ	1001	Selects bypass register while three-stating all output pins and asserting functional reset
CLAMP	1100	Selects bypass register while applying preloaded values to output pins and asserting functional reset
EZPORT	1101	Enables the EZPORT function for the SoC and asserts functional reset.
ARM_IDCODE	1110	ARM JTAG-DP Instruction
BYPASS	1111	Selects bypass register for data operations
Factory debug reserved	0101, 0110, 0111	Intended for factory debug only
ARM JTAG-DP Reserved	1000, 1010, 1011, 1110	These instructions will go the ARM JTAG-DP controller. Please look at ARM JTAG-DP documentation for more information on these instructions.
Reserved [1]	All other opcodes	Decoded to select bypass register

[1] The manufacturer reserves the right to change the decoding of reserved instruction codes in the future

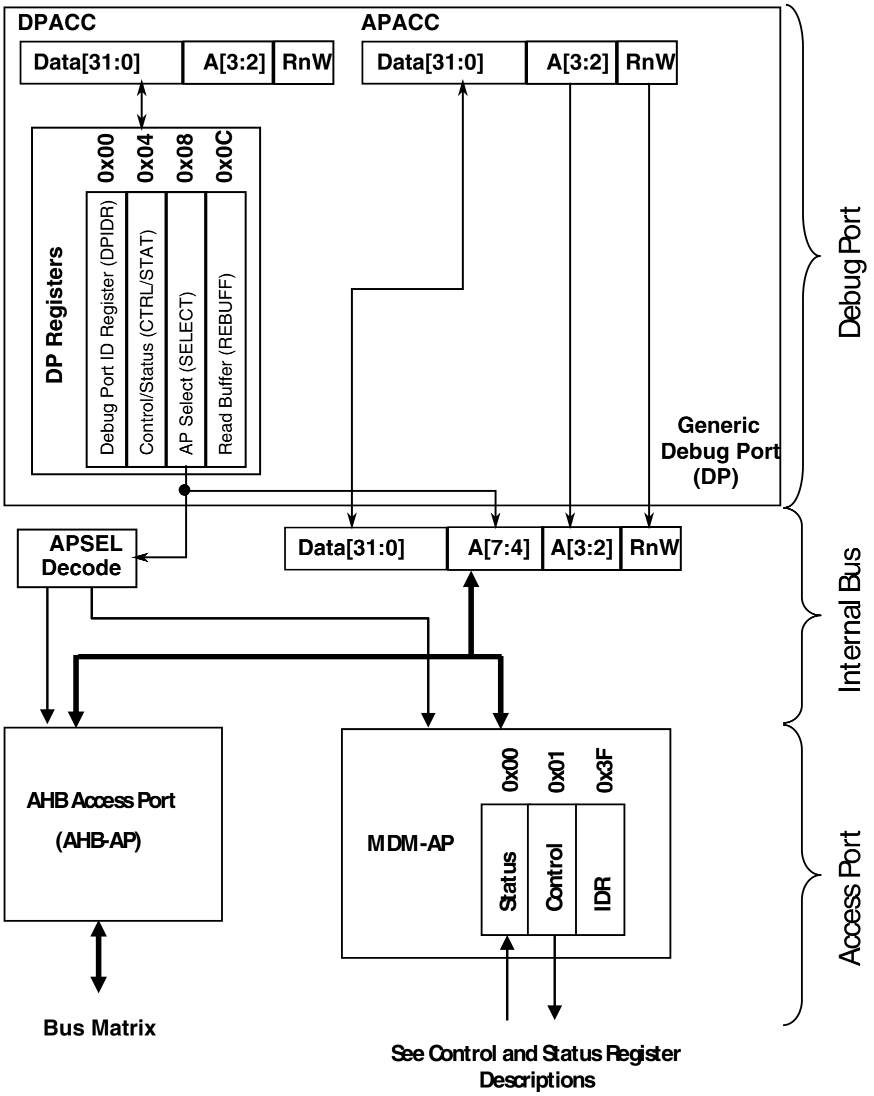
9.5 JTAG status and control registers

Through the ARM Debug Access Port (DAP), the debugger has access to the status and control elements, implemented as registers on the DAP bus as shown in the following figure. These registers provide additional control and status for low power mode recovery and typical run-control scenarios. The status register bits also provide a means for the debugger to get updated status of the core without having to initiate a bus transaction across the crossbar switch, thus remaining less intrusive during a debug session.

It is important to note that these DAP control and status registers are not memory mapped within the system memory map and are only accessible via the Debug Access Port (DAP) using JTAG, cJTAG, or SWD. The MDM-AP is accessible as Debug Access Port 1 with the available registers shown in the table below.

Table 9-4. MDM-AP Register Summary

Address	Register	Description
0x0100_0000	Status	See MDM-AP Status Register
0x0100_0004	Control	See MDM-AP Control Register
0x0100_00FC	ID	Read-only identification register that always reads as 0x001C_0000


DPACC
Data[31:0]
A[3:2]
RnW
APACC
Data[31:0]
A[3:2]
RnW
DP Registers
0x00
0x04
0x08
0x0C
Debug Port ID Register (DPIDR)
Control/Status (CTRL/STAT)
AP Select (SELECT)
Read Buffer (REBUFF)
Generic Debug Port (DP)
Debug Port
APSEL Decode
Data[31:0]
A[7:4]
A[3:2]
RnW
Internal Bus
AHB Access Port (AHB-AP)
Bus Matrix
MDM-AP
0x00
0x01
0x3F
Status
Control
IDR
See Control and Status Register Descriptions
Access Port

SWJ-DP

See the ARM Debug Interface v5p1 Supplement.

SELECT[31:24] (APSEL) selects the AP
SELECT[7:4] (APBANKSEL) selects the bank
A[3:2] from the APACC selects the register
within the bank

AHB-AP

SELECT[31:24] = 0x00 selects the AHB-AP
See ARM documentation for further details

MDM-AP

SELECT[31:24] = 0x01 selects the MDM-AP

SELECT[7:4] = 0x0 selects the bank with Status and Ctrl
A[3:2] = 2’b00 selects the Status Register
A[3:2] = 2’b01 selects the Control Register

SELECT[7:4] = 0xF selects the bank with IDR
A[3:2] = 2’b11 selects the IDR Register
(IDR register reads 0x001C_0000)

Figure 9-3. MDM AP Addressing

9.5.1 MDM-AP Control Register

Table 9-5. MDM-AP Control register assignments

Bit	Name	Secure[1]	Description
0	Flash Mass Erase in Progress	Y	Set to cause mass erase. Cleared by hardware after mass erase operation completes. When mass erase is disabled (via MEEN and SEC settings), the erase request does not occur and the Flash Mass Erase in Progress bit continues to assert until the next system reset.
1	Debug Disable	N	Set to disable debug. Clear to allow debug operation. When set it overrides the C_DEBUGEN bit within the DHCSR and force disables Debug logic.
2	Debug Request	N	Set to force the Core to halt. If the Core is in a stop or wait mode, this bit can be used to wakeup the core and transition to a halted state.
3	System Reset Request	N	Set to force a system reset. The system remains held in reset until this bit is cleared.
4	Core Hold Reset	N	Configuration bit to control Core operation at the end of system reset sequencing. 0 Normal operation - release the Core from reset along with the rest of the system at the end of system reset sequencing. 1 Suspend operation - hold the Core in reset at the end of reset sequencing. Once the system enters this suspended state, clearing this control bit immediately releases the Core from reset and CPU operation begins.
5	VLLSx Debug Request (VLLDBGREQ)	N	Set to configure the system to be held in reset after the next recovery from a VLLSx mode. This bit is ignored on a VLLS wakeup via the Reset pin. During a VLLS wakeup via the Reset pin, the system can be held in reset by holding the reset pin asserted allowing the debugger to re-initialize the debug modules. This bit holds the system in reset when VLLSx modes are exited to allow the debugger time to re-initialize debug IP before the debug session continues. The Mode Controller captures this bit logic on entry to VLLSx modes. Upon exit from VLLSx modes, the Mode Controller will hold the system in reset until VLLDBGACK is asserted. The VLLDBGREQ bit clears automatically due to the POR reset generated as part of the VLLSx recovery.
6	VLLSx Debug Acknowledge (VLLDBGACK)	N	Set to release a system being held in reset following a VLLSx recovery This bit is used by the debugger to release the system reset when it is being held on VLLSx mode exit. The debugger re-initializes all debug IP and then assert this control bit to allow the Mode Controller to release the system from reset and allow CPU operation to begin. The VLLDBGACK bit is cleared by the debugger or can be left set because it clears automatically due to the POR reset generated as part of the next VLLSx recovery.

Table continues on the next page...

Table 9-5. MDM-AP Control register assignments (continued)

Bit	Name	Secure[1]	Description
7	LLS, VLLSx Status Acknowledge	N	Set this bit to acknowledge the DAP LLS and VLLS Status bits have been read. This acknowledge automatically clears the status bits. This bit is used by the debugger to clear the sticky LLS and VLLSx mode entry status bits. This bit is asserted and cleared by the debugger.
8	Timestamp Disable	N	Set this bit to disable the 48-bit global trace timestamp counter during debug halt mode when the core is halted. 0 The timestamp counter continues to count assuming trace is enabled and the ETM is enabled. (default) 1 The timestamp counter freezes when the core has halted (debug halt mode).
9 – 31	Reserved for future use	N	

1. Command available in secure mode

9.5.2 MDM-AP Status Register

Table 9-6. MDM-AP Status register assignments

Bit	Name	Description
0	Flash Mass Erase Acknowledge	The Flash Mass Erase Acknowledge bit is cleared after any system reset. The bit is also cleared at launch of a mass erase command due to write of Flash Mass Erase in Progress bit in MDM AP Control Register. The Flash Mass Erase Acknowledge is set after Flash control logic has started the mass erase operation. When mass erase is disabled (via MEEN and SEC settings), an erase request due to seting of Flash Mass Erase in Progress bit is not acknowledged.
1	Flash Ready	Indicate Flash has been initialized and debugger can be configured even if system is continuing to be held in reset via the debugger.
2	System Security	Indicates the security state. When secure, the debugger does not have access to the system bus or any memory mapped peripherals. This bit indicates when the part is locked and no system bus access is possible.
3	System Reset	Indicates the system reset state. 0 System is in reset 1 System is not in reset
4	Reserved	
5	Mass Erase Enable	Indicates if the MCU can be mass erased or not 0 Mass erase is disabled 1 Mass erase is enabled

Table continues on the next page...

Table 9-6. MDM-AP Status register assignments (continued)

Bit	Name	Description
6	Backdoor Access Key Enable	Indicates if the MCU has the backdoor access key enabled. 0 Disabled 1 Enabled
7	LP Enabled	Decode of LPLLSM control bits to indicate that VLPS, LLS, or VLLSx are the selected power mode the next time the ARM Core enters Deep Sleep. 0 Low Power Stop Mode is not enabled 1 Low Power Stop Mode is enabled Usage intended for debug operation in which Run to VLPS is attempted. Per debug definition, the system actually enters the Stop state. A debugger should interpret deep sleep indication (with SLEEPDEEP and SLEEPING asserted), in conjuntion with this bit asserted as the debugger-VLPS status indication.
8	Very Low Power Mode	Indicates current power mode is VLPx. This bit is not ‘sticky’ and should always represent whether VLPx is enabled or not. This bit is used to throttle JTAG TCK frequency up/down.
9	LLS Mode Exit	This bit indicates an exit from LLS mode has occurred. The debugger will lose communication while the system is in LLS (including access to this register). Once communication is reestablished, this bit indicates that the system had been in LLS. Since the debug modules held their state during LLS, they do not need to be reconfigured. This bit is set during the LLS recovery sequence. The LLS Mode Exit bit is held until the debugger has had a chance to recognize that LLS was exited and is cleared by a write of 1 to the LLS, VLLSx Status Acknowledge bit in MDM AP Control register.
10	VLLSx Modes Exit	This bit indicates an exit from VLLSx mode has occurred. The debugger will lose communication while the system is in VLLSx (including access to this register). Once communication is reestablished, this bit indicates that the system had been in VLLSx. Since the debug modules lose their state during VLLSx modes, they need to be reconfigured. This bit is set during the VLLSx recovery sequence. The VLLSx Mode Exit bit is held until the debugger has had a chance to recognize that a VLLS mode was exited and is cleared by a write of 1 to the LLS, VLLSx Status Acknowledge bit in MDM AP Control register.
11 – 15	Reserved for future use	Always read 0.
16	Core Halted	Indicates the Core has entered debug halt mode
17	Core SLEEPDEEP	Indicates the Core has entered a low power mode SLEEPING==1 and SLEEPDEEP==0 indicates wait or VLPW mode. SLEEPING==1 and SLEEPDEEP==1 indicates stop or VLPS mode.
18	Core SLEEPING	
19 – 31	Reserved for future use	Always read 0.

9.6 Debug Resets

The debug system receives the following sources of reset:

- JTAG_TRST_b from an external signal. This signal is optional and may not be available in all packages.
- Debug reset (CDBGRSTREQ bit within the SWJ-DP CTRL/STAT register) in the TCLK domain that allows the debugger to reset the debug logic.
- TRST asserted via the cJTAG escape command.
- System POR reset

Conversely the debug system is capable of generating system reset using the following mechanism:

- A system reset in the DAP control register which allows the debugger to hold the system in reset.
- SYSRESETREQ bit in the NVIC application interrupt and reset control register
- A system reset in the DAP control register which allows the debugger to hold the Core in reset.

9.7 AHB-AP

AHB-AP provides the debugger access to all memory and registers in the system, including processor registers through the NVIC. System access is independent of the processor status. AHB-AP does not do back-to-back transactions on the bus, so all transactions are non-sequential. AHB-AP can perform unaligned and bit-band transactions. AHB-AP transactions bypass the FPB, so the FPB cannot remap AHB-AP transactions. SWJ/SW-DP-initiated transaction aborts drive an AHB-AP-supported sideband signal called HABORT. This signal is driven into the Bus Matrix, which resets the Bus Matrix state, so that AHB-AP can access the Private Peripheral Bus for last ditch debugging such as read/stop/reset the core. AHB-AP transactions are little endian.

For a short period at the start of a system reset event the system security status is being determined and debugger access to all AHB-AP transactions is blocked. The MDM-AP Status register is accessible and can be monitored to determine when this initial period is completed. After this initial period, if system reset is held via assertion of the RESET pin, the debugger has access via the bus matrix to the private peripheral bus to configure the debug IP even while system reset is asserted. While in system reset, access to other memory and register resources, accessed over the Crossbar Switch, is blocked.

9.8 ITM

The ITM is an application-driven trace source that supports printf style debugging to trace Operating System (OS) and application events, and emits diagnostic system information. The ITM emits trace information as packets. There are four sources that can generate packets. If multiple sources generate packets at the same time, the ITM arbitrates the order in which packets are output. The four sources in decreasing order of priority are:

1. Software trace -- Software can write directly to ITM stimulus registers. This emits packets.
2. Hardware trace -- The DWT generates these packets, and the ITM emits them.
3. Time stamping -- Timestamps are emitted relative to packets. The ITM contains a 21-bit counter to generate the timestamp. The Cortex-M4 clock or the bitclock rate of the Serial Wire Viewer (SWV) output clocks the counter.
4. Global system timestamping. Timestamps can optionally be generated using a system-wide 48-bit count value. The same count value can be used to insert timestamps in the ETM trace stream, allowing coarse-grain correlation.

9.9 Core Trace Connectivity

The ITM can route its data to the TPIU. (See the MCM (Miscellaneous Control Module) for controlling the routing to the TPIU.) This configuration enables the use of trace with low cost tools while maintaining the compatibility with trace probes.

9.10 TPIU

The TPIU acts as a bridge between the on-chip trace data from the Embedded Trace Macrocell (ETM) and the Instrumentation Trace Macrocell (ITM), with separate IDs, to a data stream, encapsulating IDs where required, that is then captured by a Trace Port Analyzer (TPA). The TPIU is specially designed for low-cost debug.

9.11 DWT

The DWT is a unit that performs the following debug functionality:

- It contains four comparators that you can configure as a hardware watchpoint, a PC sampler event trigger, or a data address sampler event trigger. The first comparator, DWT_COMP0, can also compare against the clock cycle counter, CYCCNT. The second comparator, DWT_COMP1, can also be used as a data comparator.
- The DWT contains counters for:
    - Clock cycles (CYCCNT)
    - Folded instructions
    - Load store unit (LSU) operations
    - Sleep cycles
    - CPI (all instruction cycles except for the first cycle)
    - Interrupt overhead

NOTE

An event is emitted each time a counter overflows.

- The DWT can be configured to emit PC samples at defined intervals, and to emit interrupt event information.

9.12 Debug in Low Power Modes

In low power modes in which the debug modules are kept static or powered off, the debugger cannot gather any debug data for the duration of the low power mode. In the case that the debugger is held static, the debug port returns to full functionality as soon as the low power mode exits and the system returns to a state with active debug. In the case that the debugger logic is powered off, the debugger is reset on recovery and must be reconfigured once the low power mode is exited.

Power mode entry logic monitors Debug Power Up and System Power Up signals from the debug port as indications that a debugger is active. These signals can be changed in RUN, VLPR, WAIT and VLPW. If the debug signal is active and the system attempts to enter stop or VLPS, FCLK continues to run to support core register access. In these modes in which FCLK is left active the debug modules have access to core registers but not to system memory resources accessed via the crossbar.

With debug enabled, transitions from Run directly to VLPS are not allowed and result in the system entering Stop mode instead. Status bits within the MDM-AP Status register can be evaluated to determine this pseudo-VLPS state. Note with the debug enabled, transitions from Run--> VLPR --> VLPS are still possible but also result in the system entering Stop mode instead.

In VLLS mode all debug modules are powered off and reset at wakeup. In LLS mode, the debug modules retain their state but no debug activity is possible.

NOTE

When using cJTAG and entering LLS mode, the cJTAG controller must be reset on exit from LLS mode.

Going into a VLLSx mode causes all the debug controls and settings to be reset. To give time to the debugger to sync up with the HW, the MDM-AP Control register can be configured hold the system in reset on recovery so that the debugger can regain control and reconfigure debug logic prior to the system exiting reset and resuming operation.

9.12.1 Debug Module State in Low Power Modes

The following table shows the state of the debug modules in low power modes. These terms are used:

- FF = Full functionality. In VLPR and VLPW the system frequency is limited, but if a module does not have a limitation in its functionality, it is still listed as FF.
- static = Module register states and associated memories are retained.
- OFF = Modules are powered off; module is in reset state upon wakeup.

Table 9-7. Debug Module State in Low Power Modes

Module	STOP	VLPR	VLPW	VLPS	LLS	VLLSx
Debug Port	FF	FF	FF	OFF	static	OFF
AHB-AP	FF	FF	FF	OFF	static	OFF
ITM	FF	FF	FF	OFF	static	OFF
TPIU	FF	FF	FF	OFF	static	OFF
DWT	FF	FF	FF	OFF	static	OFF

9.13 Debug & Security

When security is enabled (FSEC[SEC] != 10), the debug port capabilities are limited in order to prevent exploitation of secure data. In the secure state the debugger still has access to the MDM-AP Status Register and can determine the current security state of the device. In the case of a secure device, the debugger also has the capability of performing a mass erase operation via writes to the MDM-AP Control Register. In the case of a secure device that has mass erase disabled (FSEC[MEEN] = 10), attempts to mass erase via the debug interface are blocked.

Chapter 10
Signal Multiplexing and Signal Descriptions

10.1 Introduction

To optimize functionality in small packages, pins have several functions available via signal multiplexing. This chapter illustrates which of this device's signals are multiplexed on which external pin.

The Port Control block controls which signal is present on the external pin. Reference that chapter to find which register controls the operation of a specific pin.

10.2 Signal Multiplexing Integration

This section summarizes how the module is integrated into the device. For a comprehensive description of the module itself, see the module’s dedicated chapter.

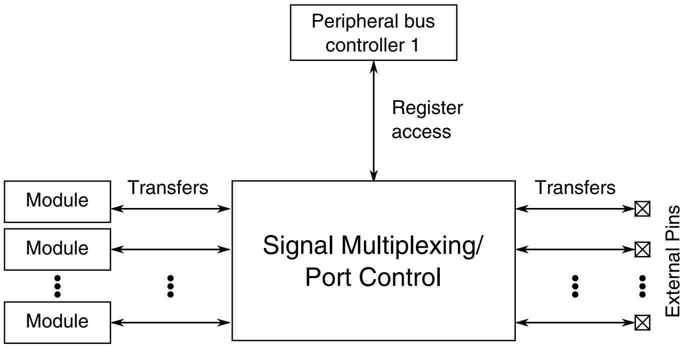

Peripheral bus controller 1
Register access
Module
Module
Module
Transfers
Signal Multiplexing/ Port Control
Transfers
External Pins

Figure 10-1. Signal multiplexing integration

Table 10-1. Reference links to related information

Topic	Related module	Reference
Full description	Port control	Port control
System memory map		System memory map

Table continues on the next page...

Table 10-1. Reference links to related information (continued)

Topic	Related module	Reference
Clocking		Clock Distribution
Register access	Peripheral bus controller	Peripheral bridge

10.2.1 Port control and interrupt module features

- Five 32-pin ports

NOTE

Not all pins are available on the device. See the following section for details.

NOTE

The digital filters are only available on Port D.

- Each 32-pin port is assigned one interrupt.
- The digital filter option has two clock source options: bus clock and 1-kHz LPO. The 1-kHz LPO option gives users this feature in low power modes.
- The digital filter is configurable from 1 to 32 clock cycles when enabled.

10.2.2 PCRn reset values for port A

PCRn bit reset values for port A are 1 for the following bits:

- For PCR0: bits 1, 6, 8, 9, and 10.
- For PCR1 to PCR4: bits 0, 1, 6, 8, 9, and 10.
- For PCR5 : bits 0, 1, and 6.

All other PCRn bit reset values for port A are 0.

10.2.3 Clock gating

The clock to the port control module can be gated on and off using the SCGC5[PORTx] bits in the SIM module. These bits are cleared after any reset, which disables the clock to the corresponding module to conserve power. Prior to initializing the corresponding module, set SCGC5[PORTx] in the SIM module to enable the clock. Before turning off the clock, make sure to disable the module. For more details, refer to the clock distribution chapter.

10.2.4 Signal multiplexing constraints

1. A given peripheral function must be assigned to a maximum of one package pin. Do not program the same function to more than one pin.
2. To ensure the best signal timing for a given peripheral's interface, choose the pins in closest proximity to each other.

10.3 Pinout

10.3.1 K20 Signal Multiplexing and Pin Assignments

The following table shows the signals available on each pin and the locations of these pins on the devices supported by this document. The Port Control Module is responsible for selecting which ALT functionality is available on each pin.

64 MAP BGA	64 LQFP	32 QFN	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
A1	1	—	PTE0	DISABLED		PTE0		UART1_TX				RTC_CLKOUT	
B1	2	—	PTE1/ LLWU_P0	DISABLED		PTE1/ LLWU_P0		UART1_RX					
C5	3	1	VDD	VDD	VDD								
C4	4	2	VSS	VSS	VSS								
E1	5	3	USB0_DP	USB0_DP	USB0_DP								
D1	6	4	USB0_DM	USB0_DM	USB0_DM								
E2	7	5	VOUT33	VOUT33	VOUT33								
D2	8	6	VREGIN	VREGIN	VREGIN								
G1	9	—	ADC0_DP0	ADC0_DP0	ADC0_DP0								
F1	10	—	ADC0_DM0	ADC0_DM0	ADC0_DM0								
G2	11	—	ADC0_DP3	ADC0_DP3	ADC0_DP3								
F2	12	—	ADC0_DM3	ADC0_DM3	ADC0_DM3								
F4	13	7	VDDA	VDDA	VDDA								
G4	14	—	VREFH	VREFH	VREFH								
G3	15	—	VREFL	VREFL	VREFL								
F3	16	8	VSSA	VSSA	VSSA								
H1	17	—	VREF_OUT/ CMP1_IN5/ CMP0_IN5	VREF_OUT/ CMP1_IN5/ CMP0_IN5	VREF_OUT/ CMP1_IN5/ CMP0_IN5								

64 MAP BGA	64 LQFP	32 QFN	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
H2	18	—	CMP1_IN3/ADC0_SE23	CMP1_IN3/ADC0_SE23	CMP1_IN3/ADC0_SE23								
H3	19	9	XTAL32	XTAL32	XTAL32								
H4	20	10	EXTAL32	EXTAL32	EXTAL32								
H5	21	11	VBAT	VBAT	VBAT								
D3	22	12	PTA0	JTAG_TCLK/SWD_CLK/EZP_CLK	TSI0_CH1	PTA0	UART0_CTS_b/UART0_COL_b	FTM0_CH5				JTAG_TCLK/SWD_CLK	EZP_CLK
D4	23	13	PTA1	JTAG_TDI/EZP_DI	TSI0_CH2	PTA1	UART0_RX	FTM0_CH6				JTAG_TDI	EZP_DI
E5	24	14	PTA2	JTAG_TDO/TRACE_SWO/EZP_DO	TSI0_CH3	PTA2	UART0_TX	FTM0_CH7				JTAG_TDO/TRACE_SWO	EZP_DO
D5	25	15	PTA3	JTAG_TMS/SWD_DIO	TSI0_CH4	PTA3	UART0_RTS_b	FTM0_CH0				JTAG_TMS/SWD_DIO	
G5	26	16	PTA4/LLWU_P3	NMI_b/EZP_CS_b	TSI0_CH5	PTA4/LLWU_P3		FTM0_CH1				NMI_b	EZP_CS_b
F5	27	—	PTA5	DISABLED		PTA5	USB_CLKIN	FTM0_CH2			I2S0_TX_BCLK	JTAG_TRST_b	
H6	28	—	PTA12	DISABLED		PTA12		FTM1_CH0			I2S0_TXD0	FTM1_QD_PHA	
G6	29	—	PTA13/LLWU_P4	DISABLED		PTA13/LLWU_P4		FTM1_CH1			I2S0_TX_FS	FTM1_QD_PHB	
G7	30	—	VDD	VDD	VDD								
H7	31	—	VSS	VSS	VSS								
H8	32	17	PTA18	EXTAL0	EXTAL0	PTA18		FTM0_FLT2	FTM_CLKIN0				
G8	33	18	PTA19	XTAL0	XTAL0	PTA19		FTM1_FLT0	FTM_CLKIN1		LPTMR0_ALT1		
F8	34	19	RESET_b	RESET_b	RESET_b								
F7	35	20	PTB0/LLWU_P5	ADC0_SE8/TSI0_CH0	ADC0_SE8/TSI0_CH0	PTB0/LLWU_P5	I2C0_SCL	FTM1_CH0			FTM1_QD_PHA		
F6	36	21	PTB1	ADC0_SE9/TSI0_CH6	ADC0_SE9/TSI0_CH6	PTB1	I2C0_SDA	FTM1_CH1			FTM1_QD_PHB		
E7	37	—	PTB2	ADC0_SE12/TSI0_CH7	ADC0_SE12/TSI0_CH7	PTB2	I2C0_SCL	UART0_RTS_b			FTM0_FLT3		
E8	38	—	PTB3	ADC0_SE13/TSI0_CH8	ADC0_SE13/TSI0_CH8	PTB3	I2C0_SDA	UART0_CTS_b/UART0_COL_b			FTM0_FLT0		
E6	39	—	PTB16	TSI0_CH9	TSI0_CH9	PTB16		UART0_RX			EWM_IN		
D7	40	—	PTB17	TSI0_CH10	TSI0_CH10	PTB17		UART0_TX			EWM_OUT_b		

64 MAP BGA	64 LQFP	32 QFN	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
D6	41	—	PTB18	TSI0_CH11	TSI0_CH11	PTB18			I2S0_TX_BCLK				
C7	42	—	PTB19	TSI0_CH12	TSI0_CH12	PTB19			I2S0_TX_FS				
D8	43	—	PTC0	ADC0_SE14/TSI0_CH13	ADC0_SE14/TSI0_CH13	PTC0	SPI0_PCS4	PDB0_EXTRG					
C6	44	22	PTC1/LLWU_P6	ADC0_SE15/TSI0_CH14	ADC0_SE15/TSI0_CH14	PTC1/LLWU_P6	SPI0_PCS3	UART1_RTS_b	FTM0_CH0		I2S0_TXD0		
B7	45	23	PTC2	ADC0_SE4b/CMP1_IN0/TSI0_CH15	ADC0_SE4b/CMP1_IN0/TSI0_CH15	PTC2	SPI0_PCS2	UART1_CTS_b	FTM0_CH1		I2S0_TX_FS		
C8	46	24	PTC3/LLWU_P7	CMP1_IN1	CMP1_IN1	PTC3/LLWU_P7	SPI0_PCS1	UART1_RX	FTM0_CH2		I2S0_TX_BCLK		
E3	47	—	VSS	VSS	VSS								
E4	48	—	VDD	VDD	VDD								
B8	49	25	PTC4/LLWU_P8	DISABLED		PTC4/LLWU_P8	SPI0_PCS0	UART1_TX	FTM0_CH3		CMP1_OUT		
A8	50	26	PTC5/LLWU_P9	DISABLED		PTC5/LLWU_P9	SPI0_SCK	LPTMR0_ALT2	I2S0_RXD0		CMP0_OUT		
A7	51	27	PTC6/LLWU_P10	CMP0_IN0	CMP0_IN0	PTC6/LLWU_P10	SPI0_SOUT	PDB0_EXTRG	I2S0_RX_BCLK		I2S0_MCLK		
B6	52	28	PTC7	CMP0_IN1	CMP0_IN1	PTC7	SPI0_SIN	USB_SOF_OUT	I2S0_RX_FS				
A6	53	—	PTC8	CMP0_IN2	CMP0_IN2	PTC8			I2S0_MCLK				
B5	54	—	PTC9	CMP0_IN3	CMP0_IN3	PTC9			I2S0_RX_BCLK				
B4	55	—	PTC10	DISABLED		PTC10			I2S0_RX_FS				
A5	56	—	PTC11/LLWU_P11	DISABLED		PTC11/LLWU_P11							
C3	57	—	PTD0/LLWU_P12	DISABLED		PTD0/LLWU_P12	SPI0_PCS0	UART2_RTS_b					
A4	58	—	PTD1	ADC0_SE5b	ADC0_SE5b	PTD1	SPI0_SCK	UART2_CTS_b					
C2	59	—	PTD2/LLWU_P13	DISABLED		PTD2/LLWU_P13	SPI0_SOUT	UART2_RX					
B3	60	—	PTD3	DISABLED		PTD3	SPI0_SIN	UART2_TX					
A3	61	29	PTD4/LLWU_P14	DISABLED		PTD4/LLWU_P14	SPI0_PCS1	UART0_RTS_b	FTM0_CH4		EWM_IN		
C1	62	30	PTD5	ADC0_SE6b	ADC0_SE6b	PTD5	SPI0_PCS2	UART0_CTS_b/UART0_COL_b	FTM0_CH5		EWM_OUT_b		

64 MAP BGA	64 LQFP	32 QFN	Pin Name	Default	ALT0	ALT1	ALT2	ALT3	ALT4	ALT5	ALT6	ALT7	EzPort
B2	63	31	PTD6/ LLWU_P15	ADC0_SE7b	ADC0_SE7b	PTD6/ LLWU_P15	SPI0_PCS3	UART0_RX	FTM0_CH6		FTM0_FLT0		
A2	64	32	PTD7	DISABLED		PTD7	CMT_IRO	UART0_TX	FTM0_CH7		FTM0_FLT1		

10.3.2 K20 Pinouts

The below figure shows the pinout diagram for the devices supported by this document. Many signals may be multiplexed onto a single pin. To determine what signals can be used on which pin, see the previous section.

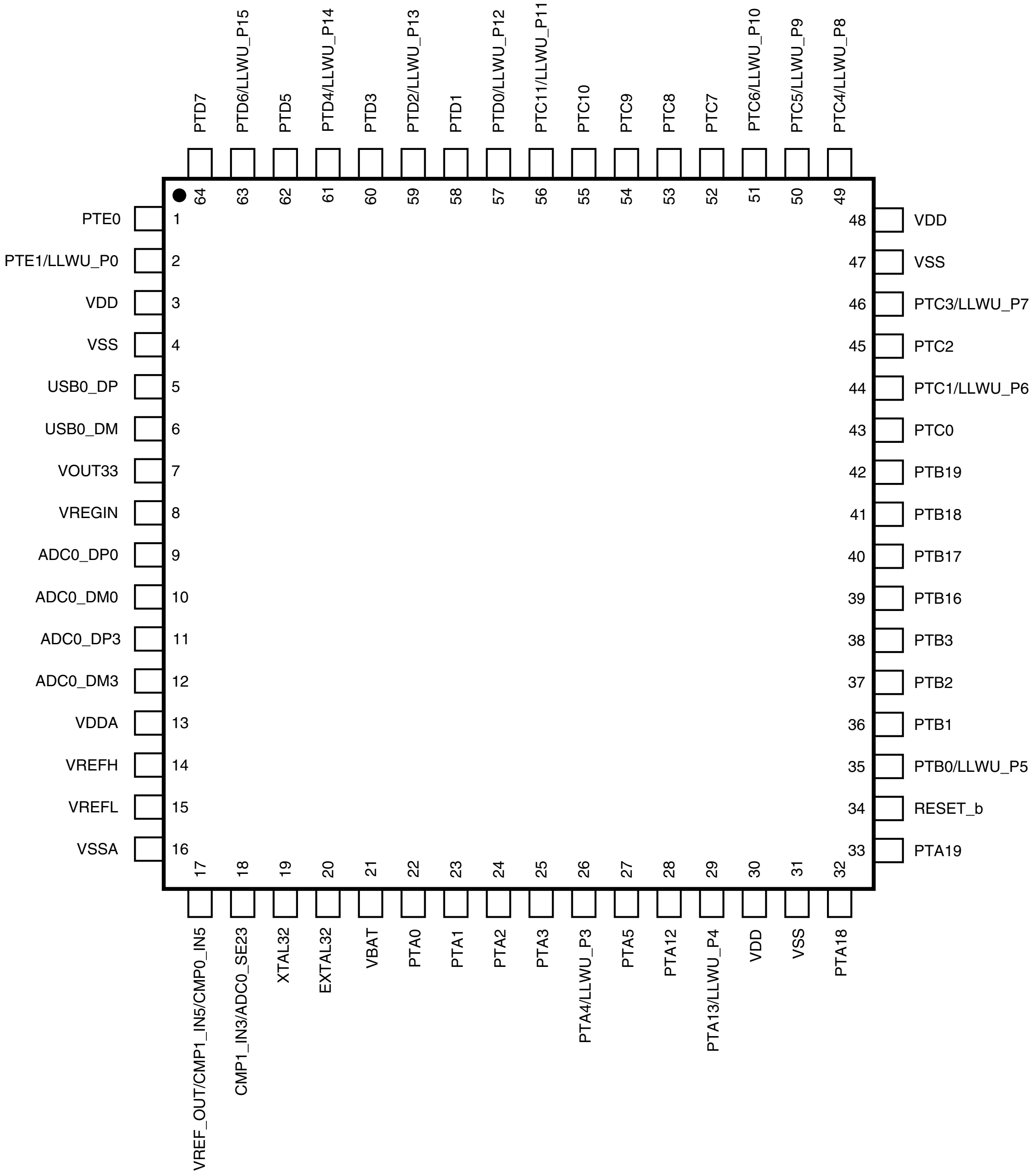

PTD7
PTD6/LLWU_P15
PTD5
PTD4/LLWU_P14
PTD3
PTD2/LLWU_P13
PTD1
PTD0/LLWU_P12
PTC11/LLWU_P11
PTC10
PTC9
PTC8
PTC7
PTC6/LLWU_P10
PTC5/LLWU_P9
PTC4/LLWU_P8
64
63
62
61
60
59
58
57
56
55
54
53
52
51
50
49
PTE0 1
PTE1/LLWU_P0 2
VDD 3
VSS 4
USB0_DP 5
USB0_DM 6
VOUT33 7
VREGIN 8
ADC0_DP0 9
ADC0_DM0 10
ADC0_DP3 11
ADC0_DM3 12
VDDA 13
VREFH 14
VREFL 15
VSSA 16
48 VDD
47 VSS
46 PTC3/LLWU_P7
45 PTC2
44 PTC1/LLWU_P6
43 PTC0
42 PTB19
41 PTB18
40 PTB17
39 PTB16
38 PTB3
37 PTB2
36 PTB1
35 PTB0/LLWU_P5
34 RESET_b
33 PTA19
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
VREF_OUT/CMP1_IN5/CMP0_IN5
CMP1_IN3/ADC0_SE23
XTAL32
EXTAL32
VBAT
PTA0
PTA1
PTA2
PTA3
PTA4/LLWU_P3
PTA5
PTA12
PTA13/LLWU_P4
VDD
VSS
PTA18

Figure 10-2. K20 64 LQFP Pinout Diagram

	1	2	3	4	5	6	7	8	
A	PTE0	PTD7	PTD4/ LLWU_P14	PTD1	PTC11/ LLWU_P11	PTC8	PTC6/ LLWU_P10	PTC5/ LLWU_P9	A
B	PTE1/ LLWU_P0	PTD6/ LLWU_P15	PTD3	PTC10	PTC9	PTC7	PTC2	PTC4/ LLWU_P8	B
C	PTD5	PTD2/ LLWU_P13	PTD0/ LLWU_P12	VSS	VDD	PTC1/ LLWU_P6	PTB19	PTC3/ LLWU_P7	C
D	USB0_DM	VREGIN	PTA0	PTA1	PTA3	PTB18	PTB17	PTC0	D
E	USB0_DP	VOUT33	VSS	VDD	PTA2	PTB16	PTB2	PTB3	E
F	ADC0_DM0	ADC0_DM3	VSSA	VDDA	PTA5	PTB1	PTB0/ LLWU_P5	RESET_b	F
G	ADC0_DP0	ADC0_DP3	VREFL	VREFH	PTA4/ LLWU_P3	PTA13/ LLWU_P4	VDD	PTA19	G
H	VREF_OUT/ CMP1_IN5/ CMP0_IN5	CMP1_IN3/ ADC0_SE23	XTAL32	EXTAL32	VBAT	PTA12	VSS	PTA18	H
	1	2	3	4	5	6	7	8	

Figure 10-3. K20 64 MAPBGA Pinout Diagram

10.4 Module Signal Description Tables

The following sections correlate the chip-level signal name with the signal name used in the module's chapter. They also briefly describe the signal function and direction.

10.4.1 Core Modules

Table 10-2. JTAG Signal Descriptions

Chip signal name	Module signal name	Description	I/O
JTAG_TMS	JTAG_TMS/ SWD_DIO	JTAG Test Mode Selection	I/O
JTAG_TCLK	JTAG_TCLK/ SWD_CLK	JTAG Test Clock	I
JTAG_TDI	JTAG_TDI	JTAG Test Data Input	I
JTAG_TDO	JTAG_TDO/ TRACE_SWO	JTAG Test Data Output	O
JTAG_TRST	JTAG_TRST_b	JTAG Reset	I

Table 10-3. SWD Signal Descriptions

Chip signal name	Module signal name	Description	I/O
SWD_DIO	JTAG_TMS/ SWD_DIO	Serial Wire Data	I/O
SWD_CLK	JTAG_TCLK/ SWD_CLK	Serial Wire Clock	I

Table 10-4. TPIU Signal Descriptions

Chip signal name	Module signal name	Description	I/O
TRACE_SWO	JTAG_TDO/ TRACE_SWO	Trace output data from the ARM CoreSight debug block over a single pin	O

10.4.2 System Modules

Table 10-5. System Signal Descriptions

Chip signal name	Module signal name	Description	I/O
$\overline{\text{NMI}}$	—	Non-maskable interrupt **NOTE:** Driving the $\overline{\text{NMI}}$ signal low forces a non-maskable interrupt, if the $\overline{\text{NMI}}$ function is selected on the corresponding pin.	I
$\overline{\text{RESET}}$	—	Reset bi-directional signal	I/O
VDD	—	MCU power	I
VSS	—	MCU ground	I

Table 10-6. EWM Signal Descriptions

Chip signal name	Module signal name	Description	I/O
EWM_IN	EWM_in	EWM input for safety status of external safety circuits. The polarity of EWM_in is programmable using the EWM_CTRL[ASSIN] bit. The default polarity is active-low.	I
$\overline{\text{EWM\_OUT}}$	$\overline{\text{EWM\_out}}$	EWM reset out signal	O

10.4.3 Clock Modules

Table 10-7. OSC Signal Descriptions

Chip signal name	Module signal name	Description	I/O
EXTAL0	EXTAL	External clock/Oscillator input	I
XTAL0	XTAL	Oscillator output	O

Table 10-8. RTC OSC Signal Descriptions

Chip signal name	Module signal name	Description	I/O
EXTAL32	EXTAL32	32.768 kHz oscillator input	I
XTAL32	XTAL32	32.768 kHz oscillator output	O

10.4.4 Memories and Memory Interfaces

Table 10-9. EzPort Signal Descriptions

Chip signal name	Module signal name	Description	I/O
EZP_CLK	EZP_CK	EzPort Clock	Input
$\overline{\text{EZP\_CS}}$	$\overline{\text{EZP\_CS}}$	EzPort Chip Select	Input
EZP_DI	EZP_D	EzPort Serial Data In	Input
EZP_DO	EZP_Q	EzPort Serial Data Out	Output

10.4.5 Analog

Table 10-10. ADC 0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
ADC0_DPn	DADP[3:0]	Differential analog channel inputs	I
ADC0_DMn	DADM[3:0]	Differential analog channel inputs	I
ADC0_SEn	AD[23:4]	Single-ended analog channel inputs	I
VREFH	V_{REFSH}	Voltage reference select high	I
VREFL	V_{REFSL}	Voltage reference select low	I
VDDA	V_{DDA}	Analog power supply	I
VSSA	V_{SSA}	Analog ground	I

Table 10-11. CMP 0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
CMP0_IN[5:0]	IN[5:0]	Analog voltage inputs	I
CMP0_OUT	CMPO	Comparator output	O

Table 10-12. CMP 1 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
CMP1_IN[5:0]	IN[5:0]	Analog voltage inputs	I
CMP1_OUT	CMPO	Comparator output	O

Table 10-13. VREF Signal Descriptions

Chip signal name	Module signal name	Description	I/O
VREF_OUT	VREF_OUT	Internally-generated Voltage Reference output	O

10.4.6 Communication Interfaces

Table 10-14. USB FS OTG Signal Descriptions

Chip signal name	Module signal name	Description	I/O
USB0_DM	usb_dm	USB D- analog data signal on the USB bus.	I/O
USB0_DP	usb_dp	USB D+ analog data signal on the USB bus.	I/O
USB_CLKIN	—	Alternate USB clock input	I

Table 10-15. USB VREG Signal Descriptions

Chip signal name	Module signal name	Description	I/O
VOUT33	reg33_out	Regulator output voltage	O
VREGIN	reg33_in	Unregulated power supply	I

Table 10-16. SPI 0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
SPI0_PCS0	PCS0/$\overline{\text{SS}}$	Master mode: Peripheral Chip Select 0 output Slave mode: Slave Select input	I/O

Table continues on the next page...

Table 10-16. SPI 0 Signal Descriptions (continued)

Chip signal name	Module signal name	Description	I/O
SPI0_PCS[3:1]	PCS[3:1]	Master mode: Peripheral Chip Select 1 – 3 Slave mode: Unused	O
SPI0_PCS4	PCS4	Master mode: Peripheral Chip Select 4 Slave mode: Unused	O
SPI0_SIN	SIN	Serial Data In	I
SPI0_SOUT	SOUT	Serial Data Out	O
SPI0_SCK	SCK	Master mode: Serial Clock (output) Slave mode: Serial Clock (input)	I/O

Table 10-17. I^{2}C 0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
I2C0_SCL	SCL	Bidirectional serial clock line of the I^{2}C system.	I/O
I2C0_SDA	SDA	Bidirectional serial data line of the I^{2}C system.	I/O

Table 10-18. UART 0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
$\overline{\text{UART0\_CTS}}$	$\overline{\text{CTS}}$	Clear to send	I
$\overline{\text{UART0\_RTS}}$	$\overline{\text{RTS}}$	Request to send	O
UART0_TX	TXD	Transmit data	O
UART0_RX	RXD	Receive data	I
$\overline{\text{UART0\_COL}}$	$\overline{\text{Collision}}$	Collision detect	I

Table 10-19. UART 1 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
$\overline{\text{UART1\_CTS}}$	$\overline{\text{CTS}}$	Clear to send	I
$\overline{\text{UART1\_RTS}}$	$\overline{\text{RTS}}$	Request to send	O
UART1_TX	TXD	Transmit data	O
UART1_RX	RXD	Receive data	I

Table 10-20. UART 2 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
$\overline{\text{UART2\_CTS}}$	$\overline{\text{CTS}}$	Clear to send	I

Table continues on the next page...

Table 10-20. UART 2 Signal Descriptions (continued)

Chip signal name	Module signal name	Description	I/O
$\overline{\text{UART2\_RTS}}$	$\overline{\text{RTS}}$	Request to send	O
UART2_TX	TXD	Transmit data	O
UART2_RX	RXD	Receive data	I

Table 10-21. I^{2}S0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
I2S0_MCLK	SAI_MCLK	Audio Master Clock	I/O
I2S0_RX_BCLK	SAI_RX_BCLK	Receive Bit Clock	I/O
I2S0_RX_FS	SAI_RX_SYNC	Receive Frame Sync	I/O
I2S0_RXD	SAI_RX_DATA	Receive Data	I
I2S0_TX_BCLK	SAI_TX_BCLK	Transmit Bit Clock	I/O
I2S0_TX_FS	SAI_TX_SYNC	Transmit Frame Sync	I/O
I2S0_TXD	SAI_TX_DATA	Transmit Data	O

10.4.7 Human-Machine Interfaces (HMI)

Table 10-22. GPIO Signal Descriptions

Chip signal name	Module signal name	Description	I/O
PTA[31:0][1]	PORTA31–PORTA0	General-purpose input/output	I/O
PTB[31:0][1]	PORTB31–PORTB0	General-purpose input/output	I/O
PTC[31:0][1]	PORTC31–PORTC0	General-purpose input/output	I/O
PTD[31:0][1]	PORTD31–PORTD0	General-purpose input/output	I/O
PTE[31:0][1]	PORTE31–PORTE0	General-purpose input/output	I/O

1. The available GPIO pins depends on the specific package. See the signal multiplexing section for which exact GPIO signals are available.

Table 10-23. TSI 0 Signal Descriptions

Chip signal name	Module signal name	Description	I/O
TSI0_CH[15:0]	TSI_IN[15:0]	TSI pins. Switchable driver that connects directly to the electrode pins TSI[15:0] can operate as GPIO pins	I/O

Chapter 11
Port control and interrupts (PORT)

11.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration chapter.

11.1.1 Overview

The port control and interrupt (PORT) module provides support for port control, and external interrupt functions. Most functions can be configured independently for each pin in the 32-bit port and affect the pin regardless of its pin muxing state.

There is one instance of the PORT module for each port. Not all pins within each port are implemented on a specific device.

11.1.2 Features

The PORT module has the following features:

- Pin interrupt
  - Interrupt flag and enable registers for each pin
  - Support for edge sensitive (rising, falling, both) or level sensitive (low, high) configured per pin
  - Support for interrupt or DMA request configured per pin
  - Asynchronous wakeup in Low-Power modes
  - Pin interrupt is functional in all digital Pin Muxing modes
- Port control
  - Individual pull control fields with pullup, pulldown, and pull-disable support
  - Individual drive strength field supporting high and low drive strength

- Individual slew rate field supporting fast and slow slew rates
- Individual input passive filter field supporting enable and disable of the individual input passive filter
- Individual open drain field supporting enable and disable of the individual open drain output
- Individual mux control field supporting analog or pin disabled, GPIO, and up to six chip-specific digital functions
- Pad configuration fields are functional in all digital Pin Muxing modes

11.1.3 Modes of operation

11.1.3.1 Run mode

In Run mode, the PORT operates normally.

11.1.3.2 Wait mode

In Wait mode, PORT continues to operate normally and may be configured to exit the Low-Power mode if an enabled interrupt is detected. DMA requests are still generated during the Wait mode, but do not cause an exit from the Low-Power mode.

11.1.3.3 Stop mode

In Stop mode, the PORT can be configured to exit the Low-Power mode via an asynchronous wakeup signal if an enabled interrupt is detected.

11.1.3.4 Debug mode

In Debug mode, PORT operates normally.

11.2 External signal description

The following table describes the PORT external signal.

Table 11-1. Signal properties

Name	Function	I/O	Reset	Pull
PORTx[31:0]	External interrupt	I/O	0	-

NOTE

Not all pins within each port are implemented on each device.

11.3 Detailed signal description

The following table contains the detailed signal description for the PORT interface.

Table 11-2. PORT interface—detailed signal description

Signal	I/O	Description	
PORTx[31:0]	I/O	External interrupt.	
		State meaning	Asserted—pin is logic one. Negated—pin is logic zero.
		Timing	Assertion—may occur at any time and can assert asynchronously to the system clock. Negation—may occur at any time and can assert asynchronously to the system clock.

11.4 Memory map and register definition

Any read or write access to the PORT memory space that is outside the valid memory map results in a bus error. All register accesses complete with zero wait states.

PORT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_9000	Pin Control Register n (PORTA_PCR0)	32	R/W	See section	11.4.1/213
4004_9004	Pin Control Register n (PORTA_PCR1)	32	R/W	See section	11.4.1/213
4004_9008	Pin Control Register n (PORTA_PCR2)	32	R/W	See section	11.4.1/213
4004_900C	Pin Control Register n (PORTA_PCR3)	32	R/W	See section	11.4.1/213
4004_9010	Pin Control Register n (PORTA_PCR4)	32	R/W	See section	11.4.1/213
4004_9014	Pin Control Register n (PORTA_PCR5)	32	R/W	See section	11.4.1/213

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_9018	Pin Control Register n (PORTA_PCR6)	32	R/W	See section	11.4.1/213
4004_901C	Pin Control Register n (PORTA_PCR7)	32	R/W	See section	11.4.1/213
4004_9020	Pin Control Register n (PORTA_PCR8)	32	R/W	See section	11.4.1/213
4004_9024	Pin Control Register n (PORTA_PCR9)	32	R/W	See section	11.4.1/213
4004_9028	Pin Control Register n (PORTA_PCR10)	32	R/W	See section	11.4.1/213
4004_902C	Pin Control Register n (PORTA_PCR11)	32	R/W	See section	11.4.1/213
4004_9030	Pin Control Register n (PORTA_PCR12)	32	R/W	See section	11.4.1/213
4004_9034	Pin Control Register n (PORTA_PCR13)	32	R/W	See section	11.4.1/213
4004_9038	Pin Control Register n (PORTA_PCR14)	32	R/W	See section	11.4.1/213
4004_903C	Pin Control Register n (PORTA_PCR15)	32	R/W	See section	11.4.1/213
4004_9040	Pin Control Register n (PORTA_PCR16)	32	R/W	See section	11.4.1/213
4004_9044	Pin Control Register n (PORTA_PCR17)	32	R/W	See section	11.4.1/213
4004_9048	Pin Control Register n (PORTA_PCR18)	32	R/W	See section	11.4.1/213
4004_904C	Pin Control Register n (PORTA_PCR19)	32	R/W	See section	11.4.1/213
4004_9050	Pin Control Register n (PORTA_PCR20)	32	R/W	See section	11.4.1/213
4004_9054	Pin Control Register n (PORTA_PCR21)	32	R/W	See section	11.4.1/213
4004_9058	Pin Control Register n (PORTA_PCR22)	32	R/W	See section	11.4.1/213
4004_905C	Pin Control Register n (PORTA_PCR23)	32	R/W	See section	11.4.1/213
4004_9060	Pin Control Register n (PORTA_PCR24)	32	R/W	See section	11.4.1/213
4004_9064	Pin Control Register n (PORTA_PCR25)	32	R/W	See section	11.4.1/213
4004_9068	Pin Control Register n (PORTA_PCR26)	32	R/W	See section	11.4.1/213
4004_906C	Pin Control Register n (PORTA_PCR27)	32	R/W	See section	11.4.1/213
4004_9070	Pin Control Register n (PORTA_PCR28)	32	R/W	See section	11.4.1/213
4004_9074	Pin Control Register n (PORTA_PCR29)	32	R/W	See section	11.4.1/213
4004_9078	Pin Control Register n (PORTA_PCR30)	32	R/W	See section	11.4.1/213
4004_907C	Pin Control Register n (PORTA_PCR31)	32	R/W	See section	11.4.1/213
4004_9080	Global Pin Control Low Register (PORTA_GPCLR)	32	W (always reads zero)	0000_0000h	11.4.2/216
4004_9084	Global Pin Control High Register (PORTA_GPCHR)	32	W (always reads zero)	0000_0000h	11.4.3/216
4004_90A0	Interrupt Status Flag Register (PORTA_ISFR)	32	w1c	0000_0000h	11.4.4/217
4004_A000	Pin Control Register n (PORTB_PCR0)	32	R/W	See section	11.4.1/213

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_A004	Pin Control Register n (PORTB_PCR1)	32	R/W	See section	11.4.1/213
4004_A008	Pin Control Register n (PORTB_PCR2)	32	R/W	See section	11.4.1/213
4004_A00C	Pin Control Register n (PORTB_PCR3)	32	R/W	See section	11.4.1/213
4004_A010	Pin Control Register n (PORTB_PCR4)	32	R/W	See section	11.4.1/213
4004_A014	Pin Control Register n (PORTB_PCR5)	32	R/W	See section	11.4.1/213
4004_A018	Pin Control Register n (PORTB_PCR6)	32	R/W	See section	11.4.1/213
4004_A01C	Pin Control Register n (PORTB_PCR7)	32	R/W	See section	11.4.1/213
4004_A020	Pin Control Register n (PORTB_PCR8)	32	R/W	See section	11.4.1/213
4004_A024	Pin Control Register n (PORTB_PCR9)	32	R/W	See section	11.4.1/213
4004_A028	Pin Control Register n (PORTB_PCR10)	32	R/W	See section	11.4.1/213
4004_A02C	Pin Control Register n (PORTB_PCR11)	32	R/W	See section	11.4.1/213
4004_A030	Pin Control Register n (PORTB_PCR12)	32	R/W	See section	11.4.1/213
4004_A034	Pin Control Register n (PORTB_PCR13)	32	R/W	See section	11.4.1/213
4004_A038	Pin Control Register n (PORTB_PCR14)	32	R/W	See section	11.4.1/213
4004_A03C	Pin Control Register n (PORTB_PCR15)	32	R/W	See section	11.4.1/213
4004_A040	Pin Control Register n (PORTB_PCR16)	32	R/W	See section	11.4.1/213
4004_A044	Pin Control Register n (PORTB_PCR17)	32	R/W	See section	11.4.1/213
4004_A048	Pin Control Register n (PORTB_PCR18)	32	R/W	See section	11.4.1/213
4004_A04C	Pin Control Register n (PORTB_PCR19)	32	R/W	See section	11.4.1/213
4004_A050	Pin Control Register n (PORTB_PCR20)	32	R/W	See section	11.4.1/213
4004_A054	Pin Control Register n (PORTB_PCR21)	32	R/W	See section	11.4.1/213
4004_A058	Pin Control Register n (PORTB_PCR22)	32	R/W	See section	11.4.1/213
4004_A05C	Pin Control Register n (PORTB_PCR23)	32	R/W	See section	11.4.1/213
4004_A060	Pin Control Register n (PORTB_PCR24)	32	R/W	See section	11.4.1/213
4004_A064	Pin Control Register n (PORTB_PCR25)	32	R/W	See section	11.4.1/213
4004_A068	Pin Control Register n (PORTB_PCR26)	32	R/W	See section	11.4.1/213
4004_A06C	Pin Control Register n (PORTB_PCR27)	32	R/W	See section	11.4.1/213
4004_A070	Pin Control Register n (PORTB_PCR28)	32	R/W	See section	11.4.1/213
4004_A074	Pin Control Register n (PORTB_PCR29)	32	R/W	See section	11.4.1/213
4004_A078	Pin Control Register n (PORTB_PCR30)	32	R/W	See section	11.4.1/213
4004_A07C	Pin Control Register n (PORTB_PCR31)	32	R/W	See section	11.4.1/213
4004_A080	Global Pin Control Low Register (PORTB_GPCLR)	32	W (always reads zero)	0000_0000h	11.4.2/216

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_A084	Global Pin Control High Register (PORTB_GPCHR)	32	W (always reads zero)	0000_0000h	11.4.3/216
4004_A0A0	Interrupt Status Flag Register (PORTB_ISFR)	32	w1c	0000_0000h	11.4.4/217
4004_B000	Pin Control Register n (PORTC_PCR0)	32	R/W	See section	11.4.1/213
4004_B004	Pin Control Register n (PORTC_PCR1)	32	R/W	See section	11.4.1/213
4004_B008	Pin Control Register n (PORTC_PCR2)	32	R/W	See section	11.4.1/213
4004_B00C	Pin Control Register n (PORTC_PCR3)	32	R/W	See section	11.4.1/213
4004_B010	Pin Control Register n (PORTC_PCR4)	32	R/W	See section	11.4.1/213
4004_B014	Pin Control Register n (PORTC_PCR5)	32	R/W	See section	11.4.1/213
4004_B018	Pin Control Register n (PORTC_PCR6)	32	R/W	See section	11.4.1/213
4004_B01C	Pin Control Register n (PORTC_PCR7)	32	R/W	See section	11.4.1/213
4004_B020	Pin Control Register n (PORTC_PCR8)	32	R/W	See section	11.4.1/213
4004_B024	Pin Control Register n (PORTC_PCR9)	32	R/W	See section	11.4.1/213
4004_B028	Pin Control Register n (PORTC_PCR10)	32	R/W	See section	11.4.1/213
4004_B02C	Pin Control Register n (PORTC_PCR11)	32	R/W	See section	11.4.1/213
4004_B030	Pin Control Register n (PORTC_PCR12)	32	R/W	See section	11.4.1/213
4004_B034	Pin Control Register n (PORTC_PCR13)	32	R/W	See section	11.4.1/213
4004_B038	Pin Control Register n (PORTC_PCR14)	32	R/W	See section	11.4.1/213
4004_B03C	Pin Control Register n (PORTC_PCR15)	32	R/W	See section	11.4.1/213
4004_B040	Pin Control Register n (PORTC_PCR16)	32	R/W	See section	11.4.1/213
4004_B044	Pin Control Register n (PORTC_PCR17)	32	R/W	See section	11.4.1/213
4004_B048	Pin Control Register n (PORTC_PCR18)	32	R/W	See section	11.4.1/213
4004_B04C	Pin Control Register n (PORTC_PCR19)	32	R/W	See section	11.4.1/213
4004_B050	Pin Control Register n (PORTC_PCR20)	32	R/W	See section	11.4.1/213
4004_B054	Pin Control Register n (PORTC_PCR21)	32	R/W	See section	11.4.1/213
4004_B058	Pin Control Register n (PORTC_PCR22)	32	R/W	See section	11.4.1/213
4004_B05C	Pin Control Register n (PORTC_PCR23)	32	R/W	See section	11.4.1/213
4004_B060	Pin Control Register n (PORTC_PCR24)	32	R/W	See section	11.4.1/213
4004_B064	Pin Control Register n (PORTC_PCR25)	32	R/W	See section	11.4.1/213
4004_B068	Pin Control Register n (PORTC_PCR26)	32	R/W	See section	11.4.1/213
4004_B06C	Pin Control Register n (PORTC_PCR27)	32	R/W	See section	11.4.1/213
4004_B070	Pin Control Register n (PORTC_PCR28)	32	R/W	See section	11.4.1/213
4004_B074	Pin Control Register n (PORTC_PCR29)	32	R/W	See section	11.4.1/213

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_B078	Pin Control Register n (PORTC_PCR30)	32	R/W	See section	11.4.1/213
4004_B07C	Pin Control Register n (PORTC_PCR31)	32	R/W	See section	11.4.1/213
4004_B080	Global Pin Control Low Register (PORTC_GPCLR)	32	W (always reads zero)	0000_0000h	11.4.2/216
4004_B084	Global Pin Control High Register (PORTC_GPCHR)	32	W (always reads zero)	0000_0000h	11.4.3/216
4004_B0A0	Interrupt Status Flag Register (PORTC_ISFR)	32	w1c	0000_0000h	11.4.4/217
4004_C000	Pin Control Register n (PORTD_PCR0)	32	R/W	See section	11.4.1/213
4004_C004	Pin Control Register n (PORTD_PCR1)	32	R/W	See section	11.4.1/213
4004_C008	Pin Control Register n (PORTD_PCR2)	32	R/W	See section	11.4.1/213
4004_C00C	Pin Control Register n (PORTD_PCR3)	32	R/W	See section	11.4.1/213
4004_C010	Pin Control Register n (PORTD_PCR4)	32	R/W	See section	11.4.1/213
4004_C014	Pin Control Register n (PORTD_PCR5)	32	R/W	See section	11.4.1/213
4004_C018	Pin Control Register n (PORTD_PCR6)	32	R/W	See section	11.4.1/213
4004_C01C	Pin Control Register n (PORTD_PCR7)	32	R/W	See section	11.4.1/213
4004_C020	Pin Control Register n (PORTD_PCR8)	32	R/W	See section	11.4.1/213
4004_C024	Pin Control Register n (PORTD_PCR9)	32	R/W	See section	11.4.1/213
4004_C028	Pin Control Register n (PORTD_PCR10)	32	R/W	See section	11.4.1/213
4004_C02C	Pin Control Register n (PORTD_PCR11)	32	R/W	See section	11.4.1/213
4004_C030	Pin Control Register n (PORTD_PCR12)	32	R/W	See section	11.4.1/213
4004_C034	Pin Control Register n (PORTD_PCR13)	32	R/W	See section	11.4.1/213
4004_C038	Pin Control Register n (PORTD_PCR14)	32	R/W	See section	11.4.1/213
4004_C03C	Pin Control Register n (PORTD_PCR15)	32	R/W	See section	11.4.1/213
4004_C040	Pin Control Register n (PORTD_PCR16)	32	R/W	See section	11.4.1/213
4004_C044	Pin Control Register n (PORTD_PCR17)	32	R/W	See section	11.4.1/213
4004_C048	Pin Control Register n (PORTD_PCR18)	32	R/W	See section	11.4.1/213
4004_C04C	Pin Control Register n (PORTD_PCR19)	32	R/W	See section	11.4.1/213
4004_C050	Pin Control Register n (PORTD_PCR20)	32	R/W	See section	11.4.1/213
4004_C054	Pin Control Register n (PORTD_PCR21)	32	R/W	See section	11.4.1/213
4004_C058	Pin Control Register n (PORTD_PCR22)	32	R/W	See section	11.4.1/213
4004_C05C	Pin Control Register n (PORTD_PCR23)	32	R/W	See section	11.4.1/213
4004_C060	Pin Control Register n (PORTD_PCR24)	32	R/W	See section	11.4.1/213

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_C064	Pin Control Register n (PORTD_PCR25)	32	R/W	See section	11.4.1/213
4004_C068	Pin Control Register n (PORTD_PCR26)	32	R/W	See section	11.4.1/213
4004_C06C	Pin Control Register n (PORTD_PCR27)	32	R/W	See section	11.4.1/213
4004_C070	Pin Control Register n (PORTD_PCR28)	32	R/W	See section	11.4.1/213
4004_C074	Pin Control Register n (PORTD_PCR29)	32	R/W	See section	11.4.1/213
4004_C078	Pin Control Register n (PORTD_PCR30)	32	R/W	See section	11.4.1/213
4004_C07C	Pin Control Register n (PORTD_PCR31)	32	R/W	See section	11.4.1/213
4004_C080	Global Pin Control Low Register (PORTD_GPCLR)	32	W (always reads zero)	0000_0000h	11.4.2/216
4004_C084	Global Pin Control High Register (PORTD_GPCHR)	32	W (always reads zero)	0000_0000h	11.4.3/216
4004_C0A0	Interrupt Status Flag Register (PORTD_ISFR)	32	w1c	0000_0000h	11.4.4/217
4004_D000	Pin Control Register n (PORTE_PCR0)	32	R/W	See section	11.4.1/213
4004_D004	Pin Control Register n (PORTE_PCR1)	32	R/W	See section	11.4.1/213
4004_D008	Pin Control Register n (PORTE_PCR2)	32	R/W	See section	11.4.1/213
4004_D00C	Pin Control Register n (PORTE_PCR3)	32	R/W	See section	11.4.1/213
4004_D010	Pin Control Register n (PORTE_PCR4)	32	R/W	See section	11.4.1/213
4004_D014	Pin Control Register n (PORTE_PCR5)	32	R/W	See section	11.4.1/213
4004_D018	Pin Control Register n (PORTE_PCR6)	32	R/W	See section	11.4.1/213
4004_D01C	Pin Control Register n (PORTE_PCR7)	32	R/W	See section	11.4.1/213
4004_D020	Pin Control Register n (PORTE_PCR8)	32	R/W	See section	11.4.1/213
4004_D024	Pin Control Register n (PORTE_PCR9)	32	R/W	See section	11.4.1/213
4004_D028	Pin Control Register n (PORTE_PCR10)	32	R/W	See section	11.4.1/213
4004_D02C	Pin Control Register n (PORTE_PCR11)	32	R/W	See section	11.4.1/213
4004_D030	Pin Control Register n (PORTE_PCR12)	32	R/W	See section	11.4.1/213
4004_D034	Pin Control Register n (PORTE_PCR13)	32	R/W	See section	11.4.1/213
4004_D038	Pin Control Register n (PORTE_PCR14)	32	R/W	See section	11.4.1/213
4004_D03C	Pin Control Register n (PORTE_PCR15)	32	R/W	See section	11.4.1/213
4004_D040	Pin Control Register n (PORTE_PCR16)	32	R/W	See section	11.4.1/213
4004_D044	Pin Control Register n (PORTE_PCR17)	32	R/W	See section	11.4.1/213
4004_D048	Pin Control Register n (PORTE_PCR18)	32	R/W	See section	11.4.1/213
4004_D04C	Pin Control Register n (PORTE_PCR19)	32	R/W	See section	11.4.1/213

Table continues on the next page...

PORT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_D050	Pin Control Register n (PORTE_PCR20)	32	R/W	See section	11.4.1/213
4004_D054	Pin Control Register n (PORTE_PCR21)	32	R/W	See section	11.4.1/213
4004_D058	Pin Control Register n (PORTE_PCR22)	32	R/W	See section	11.4.1/213
4004_D05C	Pin Control Register n (PORTE_PCR23)	32	R/W	See section	11.4.1/213
4004_D060	Pin Control Register n (PORTE_PCR24)	32	R/W	See section	11.4.1/213
4004_D064	Pin Control Register n (PORTE_PCR25)	32	R/W	See section	11.4.1/213
4004_D068	Pin Control Register n (PORTE_PCR26)	32	R/W	See section	11.4.1/213
4004_D06C	Pin Control Register n (PORTE_PCR27)	32	R/W	See section	11.4.1/213
4004_D070	Pin Control Register n (PORTE_PCR28)	32	R/W	See section	11.4.1/213
4004_D074	Pin Control Register n (PORTE_PCR29)	32	R/W	See section	11.4.1/213
4004_D078	Pin Control Register n (PORTE_PCR30)	32	R/W	See section	11.4.1/213
4004_D07C	Pin Control Register n (PORTE_PCR31)	32	R/W	See section	11.4.1/213
4004_D080	Global Pin Control Low Register (PORTE_GPCLR)	32	W (always reads zero)	0000_0000h	11.4.2/216
4004_D084	Global Pin Control High Register (PORTE_GPCHR)	32	W (always reads zero)	0000_0000h	11.4.3/216
4004_D0A0	Interrupt Status Flag Register (PORTE_ISFR)	32	w1c	0000_0000h	11.4.4/217

11.4.1 Pin Control Register n (PORTx_PCR*n*)

Addresses: 4004_9000h base + 0h offset + (4d × *n*), where *n* = 0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0							ISF	0				IRQC				LK	0				MUX			0	DSE	ODE	PFE	0	SRE	PE	PS
W								w1c																								
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- Refer to the chip configuration chapter for the reset value of this device.x = Undefined at reset.

PORTx_PCR*n* field descriptions

Field	Description
31–25 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

PORTx_PCR*n* field descriptions (continued)

Field	Description
24 ISF	Interrupt Status Flag The pin interrupt configuration is valid in all digital pin muxing modes. 0 Configured interrupt is not detected. 1 Configured interrupt is detected. If the pin is configured to generate a DMA request, then the corresponding flag will be cleared automatically at the completion of the requested DMA transfer. Otherwise, the flag remains set until a logic one is written to the flag. If the pin is configured for a level sensitive interrupt and the pin remains asserted, then the flag is set again immediately after it is cleared.
23–20 Reserved	This read-only field is reserved and always has the value zero.
19–16 IRQC	Interrupt Configuration The pin interrupt configuration is valid in all digital pin muxing modes. The corresponding pin is configured to generate interrupt/DMA request as follows: 0000 Interrupt/DMA request disabled. 0001 DMA request on rising edge. 0010 DMA request on falling edge. 0011 DMA request on either edge. 0100 Reserved. 1000 Interrupt when logic zero. 1001 Interrupt on rising edge. 1010 Interrupt on falling edge. 1011 Interrupt on either edge. 1100 Interrupt when logic one. Others Reserved.
15 LK	Lock Register 0 Pin Control Register fields [15:0] are not locked. 1 Pin Control Register fields [15:0] are locked and cannot be updated until the next system reset.
14–11 Reserved	This read-only field is reserved and always has the value zero.
10–8 MUX	Pin Mux Control Not all pins support all pin muxing slots. Unimplemented pin muxing slots are reserved and may result in configuring the pin for a different pin muxing slot. The corresponding pin is configured in the following pin muxing slot as follows: 000 Pin disabled (analog). 001 Alternative 1 (GPIO). 010 Alternative 2 (chip-specific). 011 Alternative 3 (chip-specific). 100 Alternative 4 (chip-specific). 101 Alternative 5 (chip-specific). 110 Alternative 6 (chip-specific). 111 Alternative 7 (chip-specific).

Table continues on the next page...

PORTx_PCR*n* field descriptions (continued)

Field	Description
7 Reserved	This read-only field is reserved and always has the value zero.
6 DSE	Drive Strength Enable Drive strength configuration is valid in all digital pin muxing modes. 0 Low drive strength is configured on the corresponding pin, if pin is configured as a digital output. 1 High drive strength is configured on the corresponding pin, if pin is configured as a digital output.
5 ODE	Open Drain Enable Open drain configuration is valid in all digital pin muxing modes. 0 Open drain output is disabled on the corresponding pin. 1 Open drain output is enabled on the corresponding pin, if the pin is configured as a digital output.
4 PFE	Passive Filter Enable Passive filter configuration is valid in all digital pin muxing modes. 0 Passive input filter is disabled on the corresponding pin. 1 Passive input filter is enabled on the corresponding pin, if the pin is configured as a digital input. A low pass filter of 10 MHz to 30 MHz bandwidth is enabled on the digital input path. Disable the passive input filter when high speed interfaces of more than 2 MHz are supported on the pin.
3 Reserved	This read-only field is reserved and always has the value zero.
2 SRE	Slew Rate Enable Slew rate configuration is valid in all digital pin muxing modes. 0 Fast slew rate is configured on the corresponding pin, if the pin is configured as a digital output. 1 Slow slew rate is configured on the corresponding pin, if the pin is configured as a digital output.
1 PE	Pull Enable Pull configuration is valid in all digital pin muxing modes. 0 Internal pullup or pulldown resistor is not enabled on the corresponding pin. 1 Internal pullup or pulldown resistor is enabled on the corresponding pin, if the pin is configured as a digital input.
0 PS	Pull Select Pull configuration is valid in all digital pin muxing modes. 0 Internal pulldown resistor is enabled on the corresponding pin, if the corresponding Port Pull Enable field is set. 1 Internal pullup resistor is enabled on the corresponding pin, if the corresponding Port Pull Enable field is set.

11.4.2 Global Pin Control Low Register (PORTx_GPCLR)

Only 32-bit writes are supported to this register.

Addresses: PORTA_GPCLR is 4004_9000h base + 80h offset = 4004_9080h
PORTB_GPCLR is 4004_A000h base + 80h offset = 4004_A080h
PORTC_GPCLR is 4004_B000h base + 80h offset = 4004_B080h
PORTD_GPCLR is 4004_C000h base + 80h offset = 4004_C080h
PORTE_GPCLR is 4004_D000h base + 80h offset = 4004_D080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W	GPWE																GPWD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_GPCLR field descriptions

Field	Description
31–16 GPWE	Global Pin Write Enable Selects which Pin Control Registers (15 through 0) bits [15:0] update with the value in GPWD. If a selected Pin Control Register is locked then the write to that register is ignored. 0 Corresponding Pin Control Register is not updated with the value in GPWD. 1 Corresponding Pin Control Register is updated with the value in GPWD.
15–0 GPWD	Global Pin Write Data Write value that is written to all Pin Control Registers bits [15:0] that are selected by GPWE.

11.4.3 Global Pin Control High Register (PORTx_GPCHR)

Only 32-bit writes are supported to this register.

Addresses: PORTA_GPCHR is 4004_9000h base + 84h offset = 4004_9084h
PORTB_GPCHR is 4004_A000h base + 84h offset = 4004_A084h
PORTC_GPCHR is 4004_B000h base + 84h offset = 4004_B084h
PORTD_GPCHR is 4004_C000h base + 84h offset = 4004_C084h
PORTE_GPCHR is 4004_D000h base + 84h offset = 4004_D084h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W	GPWE																GPWD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_GPCHR field descriptions

Field	Description
31–16 GPWE	Global Pin Write Enable Selects which Pin Control Registers (31 through 16) bits [15:0] update with the value in GPWD. If a selected Pin Control Register is locked then the write to that register is ignored. 0 Corresponding Pin Control Register is not updated with the value in GPWD. 1 Corresponding Pin Control Register is updated with the value in GPWD.
15–0 GPWD	Global Pin Write Data Write value that is written to all Pin Control Registers bits [15:0] that are selected by GPWE.

11.4.4 Interrupt Status Flag Register (PORTx_ISFR)

The pin interrupt configuration is valid in all digital pin muxing modes. The Interrupt Status Flag for each pin is also visible in the corresponding Pin Control Register, and each flag can be cleared in either location.

Addresses: PORTA_ISFR is 4004_9000h base + A0h offset = 4004_90A0h
PORTB_ISFR is 4004_A000h base + A0h offset = 4004_A0A0h
PORTC_ISFR is 4004_B000h base + A0h offset = 4004_B0A0h
PORTD_ISFR is 4004_C000h base + A0h offset = 4004_C0A0h
PORTE_ISFR is 4004_D000h base + A0h offset = 4004_D0A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISF																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PORTx_ISFR field descriptions

Field	Description
31–0 ISF	Interrupt Status Flag Each bit in the field indicates the detection of the configured interrupt of the same number as the field. 0 Configured interrupt is not detected. 1 Configured interrupt is detected. If the pin is configured to generate a DMA request, then the corresponding flag will be cleared automatically at the completion of the requested DMA transfer. Otherwise, the flag remains set until a logic one is written to the flag. If the pin is configured for a level sensitive interrupt and the pin remains asserted, then the flag is set again immediately after it is cleared.

11.5 Functional description

11.5.1 Pin control

Each port pin has a corresponding pin control register, PORT_PCRn, associated with it.

The upper half of the pin control register configures the pin's capability to either interrupt the CPU or request a DMA transfer, on a rising/falling edge or both edges as well as a logic level occurring on the port pin. It also includes a flag to indicate that an interrupt has occurred.

The lower half of the pin control register configures the following functions for each pin within the 32-bit port.

- Pullup or pulldown enable
- Drive strength and slew rate configuration
- Open drain enable
- Passive input filter enable
- Pin Muxing mode

The functions apply across all digital Pin Muxing modes and individual peripherals do not override the configuration in the pin control register. For example, if an I^2C function is enabled on a pin, that does not override the pullup or open drain configuration for that pin.

When the Pin Muxing mode is configured for analog or is disabled, all the digital functions on that pin are disabled. This includes the pullup and pulldown enables, digital output buffer enable, digital input buffer enable, and passive filter enable.

A lock field also exists that allows the configuration for each pin to be locked until the next system reset. When locked, writes to the lower half of that pin control register are ignored, although a bus error is not generated on an attempted write to a locked register.

The configuration of each pin control register is retained when the PORT module is disabled.

11.5.2 Global pin control

The two global pin control registers allow a single register write to update the lower half of the pin control register on up to sixteen pins, all with the same value. Registers that are locked cannot be written using the global pin control registers.

The global pin control registers are designed to enable software to quickly configure multiple pins within the one port for the same peripheral function. However, the interrupt functions cannot be configured using the global pin control registers.

The global pin control registers are write-only registers, that always read as zero.

11.5.3 External interrupts

The external interrupt capability of the PORT module is available in all digital pin muxing modes provided the PORT module is enabled.

Each pin can be individually configured for any of the following external interrupt modes:

- Interrupt disabled, default out of reset
- Active high level sensitive interrupt
- Active low level sensitive interrupt
- Rising edge sensitive interrupt
- Falling edge sensitive interrupt
- Rising and falling edge sensitive interrupt
- Rising edge sensitive DMA request
- Falling edge sensitive DMA request
- Rising and falling edge sensitive DMA request

The interrupt status flag is set when the configured edge or level is detected on the output of the pin. When not in Stop mode, the input is first synchronized to the bus clock to detect the configured level or edge transition.

The PORT module generates a single interrupt that asserts when the interrupt status flag is set for any enabled interrupt for that port. The interrupt negates after the interrupt status flags for all enabled interrupts have been cleared by writing a logic 0 to the ISF flag in the PORT_PCRn register.

The PORT module generates a single DMA request that asserts when the interrupt status flag is set for any enabled DMA request in that port. The DMA request negates after the DMA transfer is completed, because that clears the interrupt status flags for all enabled DMA requests.

During Stop mode, the interrupt status flag for any enabled interrupt is asynchronously set if the required level or edge is detected. This also generates an asynchronous wakeup signal to exit the Low-Power mode.

Chapter 12
System Integration Module (SIM)

12.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The system integration module (SIM) provides system control and chip configuration registers.

12.1.1 Features

- System clocking configuration
  - System clock divide values
  - Architectural clock gating control
  - USB clock selection and divide values
- Flash and System RAM size configuration
- USB regulator configuration
- FlexTimer external clock, hardware trigger and fault source selection
- UART0 and UART1 receive/transmit source selection/configuration

12.2 Memory map and register definition

The SIM module contains many bitfields for selecting the clock source and dividers for various module clocks. See the Clock Distribution chapter for more information including block diagrams and clock definitions.

NOTE

The SIM_SOPT1 and SIM_SOPT1CFG registers are located at a different base address than the other SIM registers.

SIM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_7000	System Options Register 1 (SIM_SOPT1)	32	R/W	See section	12.2.1/223
4004_7004	SOPT1 Configuration Register (SIM_SOPT1CFG)	32	R/W	0000_0000h	12.2.2/225
4004_8004	System Options Register 2 (SIM_SOPT2)	32	R/W	0000_1000h	12.2.3/226
4004_800C	System Options Register 4 (SIM_SOPT4)	32	R/W	0000_0000h	12.2.4/229
4004_8010	System Options Register 5 (SIM_SOPT5)	32	R/W	0000_0000h	12.2.5/232
4004_8018	System Options Register 7 (SIM_SOPT7)	32	R/W	0000_0000h	12.2.6/234
4004_8024	System Device Identification Register (SIM_SDID)	32	R	Undefined	12.2.7/235
4004_8034	System Clock Gating Control Register 4 (SIM_SCGC4)	32	R/W	F010_0030h	12.2.8/237
4004_8038	System Clock Gating Control Register 5 (SIM_SCGC5)	32	R/W	0004_0182h	12.2.9/239
4004_803C	System Clock Gating Control Register 6 (SIM_SCGC6)	32	R/W	4000_0001h	12.2.10/ 241
4004_8040	System Clock Gating Control Register 7 (SIM_SCGC7)	32	R/W	0000_0002h	12.2.11/ 243
4004_8044	System Clock Divider Register 1 (SIM_CLKDIV1)	32	R/W	See section	12.2.12/ 244
4004_8048	System Clock Divider Register 2 (SIM_CLKDIV2)	32	R/W	0000_0000h	12.2.13/ 246
4004_804C	Flash Configuration Register 1 (SIM_FCFG1)	32	R	See section	12.2.14/ 247
4004_8050	Flash Configuration Register 2 (SIM_FCFG2)	32	R	See section	12.2.15/ 249
4004_8054	Unique Identification Register High (SIM_UIDH)	32	R	See section	12.2.16/ 250
4004_8058	Unique Identification Register Mid-High (SIM_UIDMH)	32	R	See section	12.2.17/ 250

Table continues on the next page...

SIM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_805C	Unique Identification Register Mid Low (SIM_UIDML)	32	R	See section	12.2.18/ 251
4004_8060	Unique Identification Register Low (SIM_UIDL)	32	R	See section	12.2.19/ 251

12.2.1 System Options Register 1 (SIM_SOPT1)

NOTE

The SOPT1 register is only reset on POR or LVD.

Address: SIM_SOPT1 is 4004_7000h base + 0h offset = 4004_7000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	USBREGEN	USBSSTBY	USBVSTBY	0									OSC32KSEL		0	
W																
Reset	1*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RAMSIZE				0						Reserved					
W																
Reset	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:
- Reset value loaded during System Reset from Flash IFR.

SIM_SOPT1 field descriptions

Field	Description
31 USBREGEN	USB voltage regulator enable Controls whether the USB voltage regulator is enabled. 0 USB voltage regulator is disabled. 1 USB voltage regulator is enabled.
30 USBSSTBY	USB voltage regulator in standby mode during Stop, VLPS, LLS and VLLS modes. Controls whether the USB voltage regulator is placed in standby mode during Stop, VLPS, LLS and VLLS modes.

Table continues on the next page...

SIM_SOPT1 field descriptions (continued)

Field	Description
	0 USB voltage regulator not in standby during Stop, VLPS, LLS and VLLS modes. 1 USB voltage regulator in standby during Stop, VLPS, LLS and VLLS modes.
29 USBVSTBY	USB voltage regulator in standby mode during VLPR and VLPW modes Controls whether the USB voltage regulator is placed in standby mode during VLPR and VLPW modes. 0 USB voltage regulator not in standby during VLPR and VLPW modes. 1 USB voltage regulator in standby during VLPR and VLPW modes.
28–20 Reserved	This read-only field is reserved and always has the value zero.
19–18 OSC32KSEL	32K oscillator clock select Selects the 32 kHz clock source (ERCLK32K) for TSI and LPTMR. This bit is reset only for POR/LVD. 00 System oscillator (OSC32KCLK) 01 Reserved 10 RTC 32.768kHz oscillator 11 LPO 1 kHz
17–16 Reserved	This read-only field is reserved and always has the value zero.
15–12 RAMSIZE	RAM size This field specifies the amount of system RAM available on the device. 0000 Undefined 0001 8 KBytes 0010 Undefined 0011 16 KBytes 0100 Undefined 0101 Undefined 0110 Undefined 0111 Undefined 1000 Undefined 1001 Undefined 1010 Undefined 1011 Undefined 1100 Undefined 1101 Undefined 1110 Undefined 1111 Undefined
11–6 Reserved	This read-only field is reserved and always has the value zero.
5–0 Reserved	This field is reserved.

12.2.2 SOPT1 Configuration Register (SIM_SOPT1CFG)

NOTE

The SOPT1CFG register is reset on System Reset not VLLS.

Address: SIM_SOPT1CFG is 4004_7000h base + 4h offset = 4004_7004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					USSWE	UVSWE	URWE	0														0		0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SIM_SOPT1CFG field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value zero.
26 USSWE	USB voltage regulator stop standby write enable Writing one to the USSWE bit allows the SOPT1 USBSSTBY bit to be written. This register bit clears after a write to USBSSTBY. 0 SOPT1 USBSSTBY cannot be written. 1 SOPT1 USBSSTBY can be written.
25 UVSWE	USB voltage regulator VLP standby write enable Writing one to the UVSWE bit allows the SOPT1 USBVSTBY bit to be written. This register bit clears after a write to USBVSTBY. 0 SOPT1 USBVSTBY cannot be written. 1 SOPT1 USBVSTBY can be written.
24 URWE	USB voltage regulator enable write enable Writing one to the URWE bit allows the SOPT1 USBREGEN bit to be written. This register bit clears after a write to USBREGEN. 0 SOPT1 USBREGEN cannot be written. 1 SOPT1 USBREGEN can be written.
23–10 Reserved	This read-only field is reserved and always has the value zero.
9–8 Reserved	This read-only field is reserved and always has the value zero.
7–0 Reserved	This read-only field is reserved and always has the value zero.

12.2.3 System Options Register 2 (SIM_SOPT2)

SOPT2 contains the controls for selecting many of the module clock source options on this device. See the Clock Distribution chapter for more information including clocking diagrams and definitions of device clocks.

Address: SIM_SOPT2 is 4004_7000h base + 1004h offset = 4004_8004h

Bit	31	30	29	28	27	26	25	24
R	0		0		0			
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0		0			USBSRC	0	PLLFLLSEL
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0			TRACECLKSEL	PTD7PAD	0	0	
W								
Reset	0	0	0	1	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	CLKOUTSEL			RTCCLKOUTSEL	0			
W								
Reset	0	0	0	0	0	0	0	0

SIM_SOPT2 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero.
29–28 Reserved	This read-only field is reserved and always has the value zero.
27–22 Reserved	This read-only field is reserved and always has the value zero.
21–19 Reserved	This read-only field is reserved and always has the value zero.
18 USBSRC	USB clock source select Selects the clock source for the USB 48 MHz clock. 0 External bypass clock (USB_CLKIN). 1 MCGPLLCLK/MCGFLLCLK clock divided by the USB fractional divider. See the SIM_CLKDIV2[USBFRAC, USBDIV] descriptions.
17 Reserved	This read-only field is reserved and always has the value zero.
16 PLLFLLSEL	PLL/FLL clock select Selects the MCGPLLCLK or MCGFLLCLK clock for various peripheral clocking options. 0 MCGFLLCLK clock 1 MCGPLLCLK clock
15–13 Reserved	This read-only field is reserved and always has the value zero.
12 TRACECLKSEL	Debug trace clock select Selects the core/system clock or MCG output clock (MCGOUTCLK) as the trace clock source. 0 MCGOUTCLK 1 Core/system clock
11 PTD7PAD	PTD7 pad drive strength Controls the output drive strength of the PTD7 pin by selecting either one or two pads to drive it. 0 Single-pad drive strength for PTD7. 1 Double pad drive strength for PTD7.
10 Reserved	This read-only field is reserved and always has the value zero.
9–8 Reserved	This read-only field is reserved and always has the value zero.
7–5 CLKOUTSEL	CLKOUT select Selects the clock to output on the CLKOUT pin. 000 Reserved 001 Reserved 010 Flash clock

Table continues on the next page...

SIM_SOPT2 field descriptions (continued)

Field	Description
	011 LPO clock (1 kHz) 100 MCGIRCLK 101 RTC 32.768kHz clock 110 OSCERCLK0 111 Reserved
4 RTCCLKOUTSEL	RTC clock out select Selects either the RTC 1 Hz clock or the 32.768kHz clock to be output on the RTC_CLKOUT pin. 0 RTC 1 Hz clock is output on the RTC_CLKOUT pin. 1 RTC 32.768kHz clock is output on the RTC_CLKOUT pin.
3–0 Reserved	This read-only field is reserved and always has the value zero.

12.2.4 System Options Register 4 (SIM_SOPT4)

Address: SIM_SOPT4 is 4004_7000h base + 100Ch offset = 4004_800Ch

Bit	31	30	29	28	27	26	25	24
R	0		0	FTM0TRG0SRC	0	0	FTM1CLKSEL	FTM0CLKSEL
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0		0		FTM1CH0SRC		0	
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0							0
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	0			FTM1FLT0	0	0	FTM0FLT1	FTM0FLT0
W								
Reset	0	0	0	0	0	0	0	0

SIM_SOPT4 field descriptions

Field	Description
31–30 Reserved	This read-only field is reserved and always has the value zero.
29 Reserved	This read-only field is reserved and always has the value zero.
28 FTM0TRG0SRC	FlexTimer 0 Hardware Trigger 0 Source Select Selects the source of FTM0 hardware trigger 0.

Table continues on the next page...

SIM_SOPT4 field descriptions (continued)

Field	Description
	0 HSCMP0 output drives FTM0 hardware trigger 0 1 FTM1 channel match drives FTM0 hardware trigger 0
27 Reserved	This read-only field is reserved and always has the value zero.
26 Reserved	This read-only field is reserved and always has the value zero.
25 FTM1CLKSEL	FTM1 External Clock Pin Select Selects the external pin used to drive the clock to the FTM1 module. **NOTE:** The selected pin must also be configured for the FTM external clock function through the appropriate pin control register in the port control module. 0 FTM_CLK0 pin 1 FTM_CLK1 pin
24 FTM0CLKSEL	FlexTimer 0 External Clock Pin Select Selects the external pin used to drive the clock to the FTM0 module. **NOTE:** The selected pin must also be configured for the FTM external clock function through the appropriate pin control register in the port control module. 0 FTM_CLK0 pin 1 FTM_CLK1 pin
23–22 Reserved	This read-only field is reserved and always has the value zero.
21–20 Reserved	This read-only field is reserved and always has the value zero.
19–18 FTM1CH0SRC	FTM1 channel 0 input capture source select Selects the source for FTM1 channel 0 input capture. **NOTE:** When the FTM is not in input capture mode, clear this field. 00 FTM1_CH0 signal 01 CMP0 output 10 CMP1 output 11 USB start of frame pulse
17–9 Reserved	This read-only field is reserved and always has the value zero.
8 Reserved	This read-only field is reserved and always has the value zero.
7–5 Reserved	This read-only field is reserved and always has the value zero.
4 FTM1FLT0	FTM1 Fault 0 Select Selects the source of FTM1 fault 0.

Table continues on the next page...

SIM_SOPT4 field descriptions (continued)

Field	Description
	NOTE: The pin source for fault 0 must be configured for the FTM module fault function through the appropriate pin control register in the port control module. 0 FTM1_FLT0 pin 1 CMP0 out
3 Reserved	This read-only field is reserved and always has the value zero.
2 Reserved	This read-only field is reserved and always has the value zero.
1 FTM0FLT1	FTM0 Fault 1 Select Selects the source of FTM0 fault 1. **NOTE:** The pin source for fault 1 must be configured for the FTM module fault function through the appropriate pin control register in the port control module. 0 FTM0_FLT1 pin 1 CMP1 out
0 FTM0FLT0	FTM0 Fault 0 Select Selects the source of FTM0 fault 0. **NOTE:** The pin source for fault 0 must be configured for the FTM module fault function through the appropriate pin control register in the port control module. 0 FTM0_FLT0 pin 1 CMP0 out

12.2.5 System Options Register 5 (SIM_SOPT5)

Address: SIM_SOPT5 is 4004_7000h base + 1010h offset = 4004_8010h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	UART1RXSRC		0	UART1TXSRC	UART0RXSRC		0	UART0TXSRC
W								
Reset	0	0	0	0	0	0	0	0

SIM_SOPT5 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7–6 UART1RXSRC	UART 1 receive data source select Selects the source for the UART 1 receive data. 00 UART1_RX pin 01 CMP0 10 CMP1 11 Reserved
5 Reserved	This read-only field is reserved and always has the value zero.
4 UART1TXSRC	UART 1 transmit data source select Selects the source for the UART 1 transmit data. 0 UART1_TX pin 1 UART1_TX pin modulated with FTM1 channel 0 output
3–2 UART0RXSRC	UART 0 receive data source select Selects the source for the UART 0 receive data. 00 UART0_RX pin

Table continues on the next page...

SIM_SOPT5 field descriptions (continued)

Field	Description
	01 CMP0 10 CMP1 11 Reserved
1 Reserved	This read-only field is reserved and always has the value zero.
0 UART0TXSRC	UART 0 transmit data source select Selects the source for the UART 0 transmit data. 0 UART0_TX pin 1 UART0_TX pin modulated with FTM1 channel 0 output

12.2.6 System Options Register 7 (SIM_SOPT7)

Address: SIM_SOPT7 is 4004_7000h base + 1018h offset = 4004_8018h

Bit	31	30	29	28
R	0			
W				
Reset	0	0	0	0
Bit	27	26	25	24
R	0			
W				
Reset	0	0	0	0
Bit	23	22	21	20
R	0			
W				
Reset	0	0	0	0
Bit	19	18	17	16
R	0			
W				
Reset	0	0	0	0
Bit	15	14	13	12
R	0			
W				
Reset	0	0	0	0
Bit	11	10	9	8
R	0			
W				
Reset	0	0	0	0
Bit	7	6	5	4
R	ADC0ALTTRGEN	0		ADC0PRETRGSEL
W				
Reset	0	0	0	0
Bit	3	2	1	0
R	ADC0TRGSEL			
W				
Reset	0	0	0	0

SIM_SOPT7 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–8 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

SIM_SOPT7 field descriptions (continued)

Field	Description
7 ADC0ALTTRGEN	ADC0 alternate trigger enable Enable alternative conversion triggers for ADC0. 0 PDB trigger selected for ADC0. 1 Alternate trigger selected for ADC0.
6–5 Reserved	This read-only field is reserved and always has the value zero.
4 ADC0PRETRGSEL	ADC0 pretrigger select Selects the ADC0 pre-trigger source when alternative triggers are enabled through ADC0ALTTRGEN. 0 Pre-trigger A 1 Pre-trigger B
3–0 ADC0TRGSEL	ADC0 trigger select Selects the ADC0 trigger source when alternative triggers are functional in stop and VLPS modes. . 0000 PDB external trigger pin input (PDB0_EXTRG) 0001 High speed comparator 0 output 0010 High speed comparator 1 output 0011 Reserved 0100 PIT trigger 0 0101 PIT trigger 1 0110 PIT trigger 2 0111 PIT trigger 3 1000 FTM0 trigger 1001 FTM1 trigger 1010 Unused 1011 Unused 1100 RTC alarm 1101 RTC seconds 1110 Low-power timer trigger 1111 Unused

12.2.7 System Device Identification Register (SIM_SDID)

Address: SIM_SDID is 4004_7000h base + 1024h offset = 4004_8024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																REVID				0	0	0	0	0	FAMID			PINID			
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

SIM_SDID field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–12 REVID	Device revision number Specifies the silicon implementation number for the device.
11 Reserved	This read-only field is reserved and always has the value zero.
10 Reserved	This read-only field is reserved and always has the value zero.
9 Reserved	This read-only field is reserved and always has the value zero.
8 Reserved	This read-only field is reserved and always has the value zero.
7 Reserved	This read-only field is reserved and always has the value zero.
6–4 FAMID	Kinetis family identification Specifies the Kinetis family of the device. 000 K10 001 K20 010 Reserved 011 Reserved 100 Reserved 101 Reserved 110 Reserved 111 Reserved
3–0 PINID	Pincount identification Specifies the pincount of the device. 0000 Reserved 0001 Reserved 0010 32-pin 0011 Reserved 0100 48-pin 0101 64-pin 0110 Reserved 0111 Reserved 1000 Reserved 1001 Reserved 1010 Reserved 1011 Reserved 1100 Reserved 1101 Reserved 1110 Reserved 1111 Reserved

12.2.8 System Clock Gating Control Register 4 (SIM_SCGC4)

Address: SIM_SCGC4 is 4004_7000h base + 1034h offset = 4004_8034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	1				0							VREF	CMP	USBOTG	0	
W	▨				▨										▨	
Reset	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0		0	UART2	UART1	UART0	0		0	I2C0	1		0	CMT	EWM	0
W	▨		▨				▨		▨		▨		▨			▨
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

SIM_SCGC4 field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value one.
27–21 Reserved	This read-only field is reserved and always has the value zero.
20 VREF	VREF Clock Gate Control This bit controls the clock gate to the VREF module. 0 Clock disabled 1 Clock enabled
19 CMP	Comparator Clock Gate Control This bit controls the clock gate to the comparator module. 0 Clock disabled 1 Clock enabled
18 USBOTG	USB Clock Gate Control This bit controls the clock gate to the USB module. 0 Clock disabled 1 Clock enabled
17–14 Reserved	This read-only field is reserved and always has the value zero.
13 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

SIM_SCGC4 field descriptions (continued)

Field	Description
12 UART2	UART2 Clock Gate Control This bit controls the clock gate to the UART2 module. 0 Clock disabled 1 Clock enabled
11 UART1	UART1 Clock Gate Control This bit controls the clock gate to the UART1 module. 0 Clock disabled 1 Clock enabled
10 UART0	UART0 Clock Gate Control This bit controls the clock gate to the UART0 module. 0 Clock disabled 1 Clock enabled
9–8 Reserved	This read-only field is reserved and always has the value zero.
7 Reserved	This read-only field is reserved and always has the value zero.
6 I2C0	I2C0 Clock Gate Control This bit controls the clock gate to the I ^{2} C0 module. 0 Clock disabled 1 Clock enabled
5–4 Reserved	This read-only field is reserved and always has the value one.
3 Reserved	This read-only field is reserved and always has the value zero.
2 CMT	CMT Clock Gate Control This bit controls the clock gate to the CMT module. 0 Clock disabled 1 Clock enabled
1 EWM	EWM Clock Gate Control This bit controls the clock gate to the EWM module. 0 Clock disabled 1 Clock enabled
0 Reserved	This read-only field is reserved and always has the value zero.

12.2.9 System Clock Gating Control Register 5 (SIM_SCGC5)

Address: SIM_SCGC5 is 4004_7000h base + 1038h offset = 4004_8038h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0					1	0	
W								
Reset	0	0	0	0	0	1	0	0

Bit	15	14	13	12	11	10	9	8
R	0		PORTE	PORTD	PORTC	PORTB	PORTA	1
W								
Reset	0	0	0	0	0	0	0	1

Bit	7	6	5	4	3	2	1	0
R	1	0	TSI	0	0		1	LPTIMER
W								
Reset	1	0	0	0	0	0	1	0

SIM_SCGC5 field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero.
18 Reserved	This read-only field is reserved and always has the value one.
17–14 Reserved	This read-only field is reserved and always has the value zero.
13 PORTE	Port E Clock Gate Control This bit controls the clock gate to the Port E module. 0 Clock disabled 1 Clock enabled
12 PORTD	Port D Clock Gate Control This bit controls the clock gate to the Port D module. 0 Clock disabled 1 Clock enabled
11 PORTC	Port C Clock Gate Control This bit controls the clock gate to the Port C module.

Table continues on the next page...

SIM_SCGC5 field descriptions (continued)

Field	Description
	0 Clock disabled 1 Clock enabled
10 PORTB	Port B Clock Gate Control This bit controls the clock gate to the Port B module. 0 Clock disabled 1 Clock enabled
9 PORTA	Port A Clock Gate Control This bit controls the clock gate to the Port A module. 0 Clock disabled 1 Clock enabled
8–7 Reserved	This read-only field is reserved and always has the value one.
6 Reserved	This read-only field is reserved and always has the value zero.
5 TSI	TSI Clock Gate Control This bit controls the clock gate to the TSI module. 0 Clock disabled 1 Clock enabled
4 Reserved	This read-only field is reserved and always has the value zero.
3–2 Reserved	This read-only field is reserved and always has the value zero.
1 Reserved	This read-only field is reserved and always has the value one.
0 LPTIMER	Low Power Timer Access Control This bit controls software access to the Low Power Timer module. 0 Access disabled 1 Access enabled

12.2.10 System Clock Gating Control Register 6 (SIM_SCGC6)

Address: SIM_SCGC6 is 4004_7000h base + 103Ch offset = 4004_803Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	1	RTC	0	ADC0	0	FTM1	FTM0	PIT	PDB	USBDCD	0		CRC	0	
W																
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	I2S	0	0	SPI0	0		0	0				0	0		DMAMUX	FTFL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

SIM_SCGC6 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero.
30 Reserved	This read-only field is reserved and always has the value one.
29 RTC	RTC Access Control This bit controls software access and interrupts to the RTC module. 0 Access and interrupts disabled 1 Access and interrupts enabled
28 Reserved	This read-only field is reserved and always has the value zero.
27 ADC0	ADC0 Clock Gate Control This bit controls the clock gate to the ADC0 module. 0 Clock disabled 1 Clock enabled
26 Reserved	This read-only field is reserved and always has the value zero.
25 FTM1	FTM1 Clock Gate Control This bit controls the clock gate to the FTM1 module. 0 Clock disabled 1 Clock enabled

Table continues on the next page...

SIM_SCGC6 field descriptions (continued)

Field	Description
24 FTM0	FTM0 Clock Gate Control This bit controls the clock gate to the FTM0 module. 0 Clock disabled 1 Clock enabled
23 PIT	PIT Clock Gate Control This bit controls the clock gate to the PIT module. 0 Clock disabled 1 Clock enabled
22 PDB	PDB Clock Gate Control This bit controls the clock gate to the PDB module. 0 Clock disabled 1 Clock enabled
21 USBDCD	USB DCD Clock Gate Control This bit controls the clock gate to the USB DCD module. 0 Clock disabled 1 Clock enabled
20–19 Reserved	This read-only field is reserved and always has the value zero.
18 CRC	CRC Clock Gate Control This bit controls the clock gate to the CRC module. 0 Clock disabled 1 Clock enabled
17–16 Reserved	This read-only field is reserved and always has the value zero.
15 I2S	I2S Clock Gate Control This bit controls the clock gate to the I ^{2} S module. 0 Clock disabled 1 Clock enabled
14 Reserved	This read-only field is reserved and always has the value zero.
13 Reserved	This read-only field is reserved and always has the value zero.
12 SPI0	SPI0 Clock Gate Control This bit controls the clock gate to the SPI0 module.

Table continues on the next page...

SIM_SCGC6 field descriptions (continued)

Field	Description
	0 Clock disabled 1 Clock enabled
11–10 Reserved	This read-only field is reserved and always has the value zero.
9 Reserved	This read-only field is reserved and always has the value zero.
8–5 Reserved	This read-only field is reserved and always has the value zero.
4 Reserved	This read-only field is reserved and always has the value zero.
3–2 Reserved	This read-only field is reserved and always has the value zero.
1 DMAMUX	DMA Mux Clock Gate Control This bit controls the clock gate to the DMA Mux module. 0 Clock disabled 1 Clock enabled
0 FTFL	Flash Memory Clock Gate Control This bit controls the clock gate to the flash memory. Flash reads are still supported while the flash memory is clock gated, but entry into low power modes is blocked. 0 Clock disabled 1 Clock enabled

12.2.11 System Clock Gating Control Register 7 (SIM_SCGC7)

Address: SIM_SCGC7 is 4004_7000h base + 1040h offset = 4004_8040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																													0	DMA	0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

SIM_SCGC7 field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value zero.
2 Reserved	This read-only field is reserved and always has the value zero.
1 DMA	DMA Clock Gate Control

Table continues on the next page...

SIM_SCGC7 field descriptions (continued)

Field	Description
	This bit controls the clock gate to the DMA module. 0 Clock disabled 1 Clock enabled
0 Reserved	This read-only field is reserved and always has the value zero.

12.2.12 System Clock Divider Register 1 (SIM_CLKDIV1)

NOTE

The CLKDIV1 register cannot be written to when the device is in VLPR mode.

Address: SIM_CLKDIV1 is 4004_7000h base + 1044h offset = 4004_8044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OUTDIV1				OUTDIV2				0				OUTDIV4				0															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- Reset value loaded during Syetem Reset from FTFL_FOPT[LPBOOT].x = Undefined at reset.

SIM_CLKDIV1 field descriptions

Field	Description
31–28 OUTDIV1	Clock 1 output divider value This field sets the divide value for the core/system clock. At the end of reset, it is loaded with either 0000 or 0111 depending on FTFL_FOPT[LPBOOT]. 0000 Divide-by-1. 0001 Divide-by-2. 0010 Divide-by-3. 0011 Divide-by-4. 0100 Divide-by-5. 0101 Divide-by-6. 0110 Divide-by-7. 0111 Divide-by-8. 1000 Divide-by-9. 1001 Divide-by-10. 1010 Divide-by-11. 1011 Divide-by-12. 1100 Divide-by-13.

Table continues on the next page...

SIM_CLKDIV1 field descriptions (continued)

Field	Description
	1101 Divide-by-14. 1110 Divide-by-15. 1111 Divide-by-16.
27–24 OUTDIV2	Clock 2 output divider value This field sets the divide value for the peripheral clock. At the end of reset, it is loaded with either 0000 or 0111 depending on FTFL_FOPT[LPBOOT]. 0000 Divide-by-1. 0001 Divide-by-2. 0010 Divide-by-3. 0011 Divide-by-4. 0100 Divide-by-5. 0101 Divide-by-6. 0110 Divide-by-7. 0111 Divide-by-8. 1000 Divide-by-9. 1001 Divide-by-10. 1010 Divide-by-11. 1011 Divide-by-12. 1100 Divide-by-13. 1101 Divide-by-14. 1110 Divide-by-15. 1111 Divide-by-16.
23–20 Reserved	This read-only field is reserved and always has the value zero.
19–16 OUTDIV4	Clock 4 output divider value This field sets the divide value for the flash clock. At the end of reset, it is loaded with either 0001 or 1111 depending on FTFL_FOPT[LPBOOT]. 0000 Divide-by-1. 0001 Divide-by-2. 0010 Divide-by-3. 0011 Divide-by-4. 0100 Divide-by-5. 0101 Divide-by-6. 0110 Divide-by-7. 0111 Divide-by-8. 1000 Divide-by-9. 1001 Divide-by-10. 1010 Divide-by-11. 1011 Divide-by-12. 1100 Divide-by-13. 1101 Divide-by-14. 1110 Divide-by-15. 1111 Divide-by-16.

Table continues on the next page...

SIM_CLKDIV1 field descriptions (continued)

Field	Description
15–0 Reserved	This read-only field is reserved and always has the value zero.

12.2.13 System Clock Divider Register 2 (SIM_CLKDIV2)

Address: SIM_CLKDIV2 is 4004_7000h base + 1048h offset = 4004_8048h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	0				USBDIV			USBFRAC
W								
Reset	0	0	0	0	0	0	0	0

SIM_CLKDIV2 field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero.
3–1 USBDIV	USB clock divider divisor This field sets the divide value for the fractional clock divider when the MCGFLLCLK/MCGPLLCLK clock is the USB clock source (SOPT2[USBSRC] = 1). Divider output clock = Divider input clock × [(USBFRAC+1) / (USBDIV+1)]
0 USBFRAC	USB clock divider fraction This field sets the fraction multiply value for the fractional clock divider when the MCGFLLCLK/MCGPLLCLK clock is the USB clock source (SOPT2[USBSRC] = 1). Divider output clock = Divider input clock × [(USBFRAC+1) / (USBDIV+1)]

12.2.14 Flash Configuration Register 1 (SIM_FCFG1)

The reset value of EESIZE and DEPART are based on user programming in user IFR via the PGMPART flash command.

Address: SIM_FCFG1 is 4004_7000h base + 104Ch offset = 4004_804Ch

Bit	31	30	29	28	27	26	25	24
R	NVMSIZE				PFSIZE			
W								
Reset	1*	1*	1*	1*	1*	1*	1*	1*

Bit	23	22	21	20	19	18	17	16
R	0				EESIZE			
W								
Reset	0*	0*	0*	0*	1*	1*	1*	1*

Bit	15	14	13	12	11	10	9	8
R	0				DEPART			
W								
Reset	0*	0*	0*	0*	1*	1*	1*	1*

Bit	7	6	5	4	3	2	1	0
R	0						FLASHDOZE	FLASHDIS
W								
Reset	0*	0*	0*	0*	0*	0*	0*	0*

* Notes:

- Reset value loaded during System Reset from Flash IFR.

SIM_FCFG1 field descriptions

Field	Description
31–28 NVMSIZE	FlexNVM size This field specifies the amount of FlexNVM memory available on the device . Undefined values are reserved. 0000 0 KB of FlexNVM 0011 32 KB of FlexNVM, 4 KB protection region
27–24 PFSIZE	Program flash size This field specifies the amount of program flash memory available on the device . Undefined values are reserved. 0011 32 KB of program flash memory, 1 KB protection region 0101 64 KB of program flash memory, 2 KB protection region 0111 128 KB of program flash, 4 KB protection region

Table continues on the next page...

SIM_FCFG1 field descriptions (continued)

Field	Description
23–20 Reserved	This read-only field is reserved and always has the value zero.
19–16 EESIZE	EEPROM size EEPROM data size . 0000 Reserved 0001 Reserved 0010 Reserved 0011 2 KB 0100 1 KB 0101 512 Bytes 0110 256 Bytes 0111 128 Bytes 1000 64 Bytes 1001 32 Bytes 1010-1110 Reserved 1111 0 Bytes
15–12 Reserved	This read-only field is reserved and always has the value zero.
11–8 DEPART	FlexNVM partition Data flash / EEPROM backup split . See DEPART bit description in FTFL chapter.
7–2 Reserved	This read-only field is reserved and always has the value zero.
1 FLASHDOZE	Flash Doze When set, Flash memory is disabled for the duration of Wait mode. An attempt by the DMA or other bus master to access the Flash when the Flash is disabled will result in a bus error. This bit should be clear during VLP modes. The Flash will be automatically enabled again at the end of Wait mode so interrupt vectors do not need to be relocated out of Flash memory. The wakeup time from Wait mode is extended when this bit is set. 0 Flash remains enabled during Wait mode 1 Flash is disabled for the duration of Wait mode
0 FLASHDIS	Flash Disable Flash accesses are disabled (and generate a bus error) and the Flash memory is placed in a low power state. This bit should not be changed during VLP modes. Relocate the interrupt vectors out of Flash memory before disabling the Flash. 0 Flash is enabled 1 Flash is disabled

12.2.15 Flash Configuration Register 2 (SIM_FCFG2)

Address: SIM_FCFG2 is 4004_7000h base + 1050h offset = 4004_8050h

Bit	31	30	29	28	27	26	25	24
R	0	MAXADDR0						
W								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

Bit	23	22	21	20	19	18	17	16
R	PFLSH	MAXADDR1						
W								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

Bit	15	14	13	12	11	10	9	8
R	0							
W								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

Bit	7	6	5	4	3	2	1	0
R	0							
W								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- Reset value loaded during System Reset from Flash IFR.x = Undefined at reset.

SIM_FCFG2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero.
30–24 MAXADDR0	Max address block 0 This field concatenated with leading zeros indicates the first invalid address of flash block 0 (program flash 0). For example, if MAXADDR0 = 0x20 the first invalid address of flash block 0 is 0x0004_0000. This would be the MAXADDR0 value for a device with 256 KB program flash in flash block 0.
23 PFLSH	Program flash For devices with FlexNVM, this bit is always clear. 0 Physical flash block 1 is used as FlexNVM 1 Physical flash block 1 is used as program flash
22–16 MAXADDR1	Max address block 1 This field concatenated with leading zeros plus the FlexNVM base address indicates the first invalid address of the FlexNVM (flash block 1). For example, if MAXADDR1 = 0x20 the first invalid address of flash block 1 is 0x4_0000 + 0x1000_0000 . This would be the MAXADDR1 value for a device with 256 KB FlexNVM.

Table continues on the next page...

SIM_FCFG2 field descriptions (continued)

Field	Description
15–0 Reserved	This read-only field is reserved and always has the value zero.

12.2.16 Unique Identification Register High (SIM_UIDH)

Address: SIM_UIDH is 4004_7000h base + 1054h offset = 4004_8054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UID																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- Reset value loaded during System Reset from Flash IFR.x = Undefined at reset.

SIM_UIDH field descriptions

Field	Description
31–0 UID	Unique Identification Unique identification for the device.

12.2.17 Unique Identification Register Mid-High (SIM_UIDMH)

Address: SIM_UIDMH is 4004_7000h base + 1058h offset = 4004_8058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UID																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- Reset value loaded during System Reset from Flash IFR.x = Undefined at reset.

SIM_UIDMH field descriptions

Field	Description
31–0 UID	Unique Identification Unique identification for the device.

12.2.18 Unique Identification Register Mid Low (SIM_UIDML)

Address: SIM_UIDML is 4004_7000h base + 105Ch offset = 4004_805Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UID																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- Reset value loaded during System Reset from Flash IFR.x = Undefined at reset.

SIM_UIDML field descriptions

Field	Description
31–0 UID	Unique Identification Unique identification for the device.

12.2.19 Unique Identification Register Low (SIM_UIDL)

Address: SIM_UIDL is 4004_7000h base + 1060h offset = 4004_8060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	UID																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- Reset value loaded during System Reset from Flash IFR.x = Undefined at reset.

SIM_UIDL field descriptions

Field	Description
31–0 UID	Unique Identification Unique identification for the device.

12.3 Functional description

See Introduction section.

Chapter 13
Reset Control Module (RCM)

13.1 Introduction

This chapter describes the registers of the Reset Control Module (RCM). The RCM implements many of the reset functions for the chip. See the chip's reset chapter for more information.

13.2 Reset memory map and register descriptions

The reset control module (RCM) registers provide reset status information and reset filter control.

RCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_F000	System Reset Status Register 0 (RCM_SRS0)	8	R	82h	13.2.1/253
4007_F001	System Reset Status Register 1 (RCM_SRS1)	8	R	00h	13.2.2/255
4007_F004	Reset Pin Filter Control Register (RCM_RPFC)	8	R/W	00h	13.2.3/257
4007_F005	Reset Pin Filter Width Register (RCM_RPFW)	8	R/W	00h	13.2.4/258
4007_F007	Mode Register (RCM_MR)	8	R	00h	13.2.5/259

13.2.1 System Reset Status Register 0 (RCM_SRS0)

This register includes read-only status flags to indicate the source of the most recent reset. The reset state of these bits depends on what caused the MCU to reset.

NOTE

The reset value of this register depends on the reset source:

- POR (including LVD) — 0x82

- LVD (without POR) — 0x02
- VLLS mode wakeup due to $\overline{\text{RESET}}$ pin assertion — 0x41
- VLLS mode wakeup due to other wakeup sources — 0x01
- Other reset — a bit is set if its corresponding reset source caused the reset

Address: RCM_SRS0 is 4007_F000h base + 0h offset = 4007_F000h

Bit	7	6	5	4	3	2	1	0
Read	POR	PIN	WDOG	0	LOL	LOC	LVD	WAKEUP
Write								
Reset	1	0	0	0	0	0	1	0

RCM_SRS0 field descriptions

Field	Description
7 POR	Power-on reset Indicates a reset was caused by the power-on detection logic. Because the internal supply voltage was ramping up at the time, the low-voltage reset (LVD) status bit is also set to indicate that the reset occurred while the internal supply was below the LVD threshold. 0 Reset not caused by POR 1 Reset caused by POR
6 PIN	External reset pin Indicates a reset was caused by an active-low level on the external $\overline{\text{RESET}}$ pin. 0 Reset not caused by external reset pin 1 Reset caused by external reset pin
5 WDOG	Watchdog Indicates a reset was caused by the watchdog timer timing out. This reset source can be blocked by disabling the watchdog. 0 Reset not caused by watchdog timeout 1 Reset caused by watchdog timeout
4 Reserved	This read-only field is reserved and always has the value zero.
3 LOL	Loss-of-lock reset Indicates a reset was caused by a loss of lock in the MCG PLL. See the MCG description for information on the loss-of-clock event. 0 Reset not caused by a loss of lock in the PLL 1 Reset caused by a loss of lock in the PLL
2 LOC	Loss-of-clock reset Indicates a reset was caused by a loss of external clock. The MCG clock monitor must be enabled for a loss of clock to be detected. Refer to the detailed MCG description for information on enabling the clock monitor.

Table continues on the next page...

RCM_SRS0 field descriptions (continued)

Field	Description
	0 Reset not caused by a loss of external clock. 1 Reset caused by a loss of external clock.
1 LVD	Low-voltage detect reset If the LVDRE bit is set and the supply drops below the LVD trip voltage, an LVD reset occurs. This bit is also set by POR. 0 Reset not caused by LVD trip or POR 1 Reset caused by LVD trip or POR
0 WAKEUP	Low leakage wakeup reset Indicates a reset was caused by an enabled LLWU module wakeup source while the chip was in a low leakage mode. In LLS mode, the $\overline{\text{RESET}}$ pin is the only wakeup source that can cause this reset. Any enabled wakeup source in a VLLSx mode causes a reset. This bit is cleared by any reset except WAKEUP. 0 Reset not caused by LLWU module wakeup source 1 Reset caused by LLWU module wakeup source

13.2.2 System Reset Status Register 1 (RCM_SRS1)

This register includes read-only status flags to indicate the source of the most recent reset. The reset state of these bits depends on what caused the MCU to reset.

NOTE

The reset value of this register depends on the reset source:

- POR (including LVD) — 0x00
- LVD (without POR) — 0x00
- VLLS mode wakeup — 0x00
- Other reset — a bit is set if its corresponding reset source caused the reset

Address: RCM_SRS1 is 4007_F000h base + 1h offset = 4007_F001h

Bit	7	6	5	4	3	2	1	0
Read	0	0	SACKERR	EZPT	MDM_AP	SW	LOCKUP	JTAG
Write	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

RCM_SRS1 field descriptions

Field	Description
7 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

RCM_SRS1 field descriptions (continued)

Field	Description
6 Reserved	This read-only field is reserved and always has the value zero.
5 SACKERR	Stop Mode Acknowledge Error Reset Indicates a reset was caused, after an attempt to enter stop mode, by a failure of one or more peripherals to acknowledge within approximately one second to enter stop mode. 0 Reset not caused by peripheral failure to acknowledge attempt to enter stop mode 1 Reset caused by peripheral failure to acknowledge attempt to enter stop mode
4 EZPT	EzPort Reset Indicates a reset was caused by EzPort receiving the RESET command while the device is in EzPort mode. 0 Reset not caused by EzPort receiving the RESET command while the device is in EzPort mode 1 Reset caused by EzPort receiving the RESET command while the device is in EzPort mode
3 MDM_AP	MDM-AP system reset request Indicates a reset was caused by the host debugger system setting of the System Reset Request bit in the MDM-AP Control Register. 0 Reset not caused by host debugger system setting of the System Reset Request bit 1 Reset caused by host debugger system setting of the System Reset Request bit
2 SW	Software Indicates a reset was caused by software setting of SYSRESETREQ bit in Application Interrupt and Reset Control Register in the ARM core. 0 Reset not caused by software setting of SYSRESETREQ bit 1 Reset caused by software setting of SYSRESETREQ bit
1 LOCKUP	Core Lockup Indicates a reset was caused by the ARM core indication of a LOCKUP event. 0 Reset not caused by core LOCKUP event 1 Reset caused by core LOCKUP event
0 JTAG	JTAG generated reset Indicates a reset was caused by JTAG selection of certain IR codes (EZPORT, EXTEST, HIGHZ, and CLAMP). 0 Reset not caused by JTAG 1 Reset caused by JTAG

13.2.3 Reset Pin Filter Control Register (RCM_RPFC)

NOTE

The reset values of bits 2-0 are for Chip POR only. They are unaffected by other reset types.

NOTE

The bus clock filter is reset when disabled or when entering stop mode. The LPO filter is reset when disabled or when entering any low leakage stop mode .

Address: RCM_RPFC is 4007_F000h base + 4h offset = 4007_F004h

Bit	7	6	5	4	3	2	1	0
Read	0					RSTFLTSS	RSTFLTSRW	
Write								
Reset	0	0	0	0	0	0	0	0

RCM_RPFC field descriptions

Field	Description
7–3 Reserved	This read-only field is reserved and always has the value zero.
2 RSTFLTSS	Reset pin filter select in stop mode Selects how the reset pin filter is enabled in STOP and VLPS modes . 0 All filtering disabled 1 LPO clock filter enabled
1–0 RSTFLTSRW	Reset pin filter select in run and wait modes Selects how the reset pin filter is enabled in run and wait modes. 00 All filtering disabled 01 Bus clock filter enabled for normal operation 10 LPO clock filter enabled for normal operation 11 Reserved (all filtering disabled)

13.2.4 Reset Pin Filter Width Register (RCM_RPFW)

NOTE

The reset values of the bits in the RSTFLTSEL field are for Chip POR only. They are unaffected by other reset types.

Address: RCM_RPFW is 4007_F000h base + 5h offset = 4007_F005h

Bit	7	6	5	4	3	2	1	0
Read	0			RSTFLTSEL				
Write								
Reset	0	0	0	0	0	0	0	0

RCM_RPFW field descriptions

Field	Description
7–5 Reserved	This read-only field is reserved and always has the value zero.
4–0 RSTFLTSEL	Reset pin filter bus clock select Selects the reset pin bus clock filter width. 00000 Bus clock filter count is 1 00001 Bus clock filter count is 2 00010 Bus clock filter count is 3 00011 Bus clock filter count is 4 00100 Bus clock filter count is 5 00101 Bus clock filter count is 6 00110 Bus clock filter count is 7 00111 Bus clock filter count is 8 01000 Bus clock filter count is 9 01001 Bus clock filter count is 10 01010 Bus clock filter count is 11 01011 Bus clock filter count is 12 01100 Bus clock filter count is 13 01101 Bus clock filter count is 14 01110 Bus clock filter count is 15 01111 Bus clock filter count is 16 10000 Bus clock filter count is 17 10001 Bus clock filter count is 18 10010 Bus clock filter count is 19 10011 Bus clock filter count is 20 10100 Bus clock filter count is 21 10101 Bus clock filter count is 22 10110 Bus clock filter count is 23 10111 Bus clock filter count is 24

Table continues on the next page...

RCM_RPFW field descriptions (continued)

Field	Description
	11000 Bus clock filter count is 25 11001 Bus clock filter count is 26 11010 Bus clock filter count is 27 11011 Bus clock filter count is 28 11100 Bus clock filter count is 29 11101 Bus clock filter count is 30 11110 Bus clock filter count is 31 11111 Bus clock filter count is 32

13.2.5 Mode Register (RCM_MR)

This register includes read-only status flags to indicate the state of the mode pins during the last Chip Reset.

Address: RCM_MR is 4007_F000h base + 7h offset = 4007_F007h

Bit	7	6	5	4	3	2	1	0
Read	0						EZP_MS	0
Write								
Reset	0	0	0	0	0	0	0	0

RCM_MR field descriptions

Field	Description
7–2 Reserved	This read-only field is reserved and always has the value zero.
1 EZP_MS	EZP_MS_B pin state Reflects the state of the $\overline{\text{EZP\_MS}}$ pin during the last Chip Reset 0 Pin negated (logic 1) 1 Pin asserted (logic 0)
0 Reserved	This read-only field is reserved and always has the value zero.

Chapter 14
System Mode Controller

14.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The system mode controller (SMC) is responsible for sequencing the system into and out of all low power stop and run modes. Specifically, it monitors events to trigger transitions between power modes while controlling the power, clocks, and memories of the system to achieve the power consumption and functionality of that mode.

This chapter describes all the available low power modes, the sequence followed to enter/exit each mode, and the functionality available while in each of the modes.

The SMC is able to function during even the deepest low power modes.

14.2 Modes of operation

The ARM CPU has three primary modes of operation:

- Run
- Sleep
- Deep Sleep

The WFI or WFE instruction is used to invoke Sleep and Deep Sleep modes. Run, wait and stop are the common terms used for the primary operating modes of Freescale microcontrollers. The following table shows the translation between the ARM CPU modes and the Freescale MCU power modes.

ARM CPU mode	MCU mode
Sleep	Wait
Deep Sleep	Stop

Accordingly, the ARM CPU documentation refers to sleep and deep sleep, while the Freescale MCU documentation normally uses wait and stop.

In addition, Freescale MCUs also augment stop, wait, and run modes in a number of ways. The power management controller (PMC) contains a run and a stop mode regulator. Run regulation is used in normal run, wait and stop modes. Stop mode regulation is used during all very low power and low leakage modes. During stop mode regulation, the bus frequencies are limited in the very low power modes.

The SMC provides the user with multiple power options. The Very Low Power Run (VLPR) mode can drastically reduce run time power when maximum bus frequency is not required to handle the application needs. From Normal Run mode, the Run Mode (RUNM) field can be modified to change the MCU into VLPR mode when limited frequency is sufficient for the application. From VLPR mode, a corresponding wait (VLPW) and stop (VLPS) mode can be entered.

Depending on the needs of the user application, a variety of stop modes are available that allow the state retention, partial power down or full power down of certain logic and/or memory. I/O states are held in all modes of operation. Several registers are used to configure the various modes of operation for the device.

The following table describes the power modes available for the device.

Table 14-1. Power modes

Mode	Description
RUN	The MCU can be run at full speed and the internal supply is fully regulated, that is, in run regulation. This mode is also referred to as Normal Run mode.
WAIT	The core clock to the ARM Cortex-M4 core is shut off. The system clock continues to operate. Bus clocks, if enabled, continue to operate. Run regulation is maintained.
STOP	The core clock and system clock to the ARM Cortex-M4 core are shut off. System clock to other masters and bus clocks are stopped after all stop acknowledge signals from supporting peripherals are valid.
VLPR	The core, system, bus, and flash clock maximum frequencies are restricted in this mode. See the Power Management chapter for details about the maximum allowable frequencies.
VLPW	The core clock to the ARM Cortex-M4 core is shut off. The system, bus, and flash clocks continue to operate, although their maximum frequency is restricted. See the Power Management chapter for details on the maximum allowable frequencies.
VLPS	The core clock and system clock to the ARM Cortex-M4 core is shut off. System clock to other masters and bus clocks are stopped after all stop acknowledge signals from supporting peripherals are valid.

Table continues on the next page...

Table 14-1. Power modes (continued)

Mode	Description
LLS	The core clock and system clock to the ARM Cortex-M4 core is shut off. System clock and bus clocks are stopped after all stop acknowledge signals from supporting peripherals are valid. The MCU is placed in a low leakage mode by reducing the voltage to internal logic. Internal logic states are retained.
VLLS3	The core clock and system clock to the ARM Cortex-M4 core is shut off. System clock to other masters and bus clocks are stopped after all stop acknowledge signals from supporting peripherals are valid. The MCU is placed in a low leakage mode by powering down the internal logic. All system RAM contents are retained and I/O states are held. FlexRAM contents are not retained. Internal logic states are not retained.
VLLS2	The core clock and system clock to the ARM Cortex-M4 core is shut off. System clock to other masters and bus clocks are stopped after all stop acknowledge signals from supporting peripherals are valid. The MCU is placed in a low leakage mode by powering down the internal logic and the system RAM2 partition. The system RAM1 partition contents are retained in this mode. FlexRAM contents are not retained. Internal logic states are not retained. [1]
VLLS1	In ARM architectures, core clock and system clock to the ARM Cortex-M4 core is shut off. System clock to other masters and bus clocks are stopped after all stop acknowledge signals from supporting peripherals are valid. The MCU is placed in a low leakage mode by powering down the internal logic and all system RAM. A 32-byte register file (available in all modes) contents are retained and I/O states held. FlexRAM contents are not retained. Internal logic states are not retained.
VLLS0	In ARM architectures, core clock and system clock to the ARM Cortex-M4 core is shut off. System clock to other masters and bus clocks are stopped after all stop acknowledge signals from supporting peripherals are valid. The MCU is placed in a low leakage mode by powering down the internal logic and all system RAM. A 32-byte register file (available in all modes) contents are retained and I/O states held. FlexRAM contents are not retained. Internal logic states are not retained. The 1kHz LPO clock is disabled and the power on reset (POR) circuit can be optionally enabled using VLLSCTRL[PORPO].

1. See the devices' chip configuration details for the size and location of the system RAM partitions.

14.3 Memory map and register descriptions

Details follow about the registers related to the system mode controller.

Different SMC registers reset on different reset types. Each register's description provides details. For more information about the types of reset on this chip, refer to the Reset section details.

SMC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_E000	Power Mode Protection Register (SMC_PMPROT)	8	R/W	00h	14.3.1/264
4007_E001	Power Mode Control Register (SMC_PMCTRL)	8	R/W	00h	14.3.2/265
4007_E002	VLLS Control Register (SMC_VLLSCTRL)	8	R/W	03h	14.3.3/267
4007_E003	Power Mode Status Register (SMC_PMSTAT)	8	R	01h	14.3.4/268

14.3.1 Power Mode Protection Register (SMC_PMPROT)

This register provides protection for entry into any low power run or stop mode. The actual enabling of the low power run or stop mode occurs by configuring the power mode control register (PMCTRL).

The PMPROT register can be written only once after any system reset.

If the MCU is configured for a disallowed or reserved power mode, the MCU remains in its current power mode. For example, if the MCU is in normal RUN mode and AVLP is 0, an attempt to enter VLPR mode using PMCTRL[RUNM] is blocked and the RUNM bits remain 00b, indicating the MCU is still in normal run mode.

NOTE

> This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Reset section details for more information.

Address: SMC_PMPROT is 4007_E000h base + 0h offset = 4007_E000h

Bit	7	6	5	4	3	2	1	0
Read	0		AVLP	0	ALLS	0	AVLLS	0
Write	▨	▨		▨		▨		▨
Reset	0	0	0	0	0	0	0	0

SMC_PMPROT field descriptions

Field	Description
7–6 Reserved	This read-only field is reserved and always has the value zero.
5 AVLP	Allow very low power modes Provided the appropriate control bits are set up in PMCTRL, this write-once bit allows the MCU to enter any very low power modes: VLPR, VLPW, and VLPS. 0 VLPR, VLPW and VLPS are not allowed 1 VLPR, VLPW and VLPS are allowed
4 Reserved	This read-only field is reserved and always has the value zero.
3 ALLS	Allow low leakage stop mode This write once bit allows the MCU to enter any low leakage stop mode (LLS) provided the appropriate control bits are set up in PMCTRL. 0 LLS is not allowed 1 LLS is allowed

Table continues on the next page...

SMC_PMPROT field descriptions (continued)

Field	Description
2 Reserved	This read-only field is reserved and always has the value zero.
1 AVLLS	Allow very low leakage stop mode Provided the appropriate control bits are set up in PMCTRL, this write once bit allows the MCU to enter any very low leakage stop mode (VLLSx). 0 Any VLLSx mode is not allowed 1 Any VLLSx mode is allowed
0 Reserved	This read-only field is reserved and always has the value zero.

14.3.2 Power Mode Control Register (SMC_PMCTRL)

The PMCTRL register controls entry into low power run and stop modes, provided that the selected power mode is allowed via an appropriate setting of the protection (PMPROT) register.

NOTE

This register is reset on Chip POR not VLLS and by reset types that trigger Chip POR not VLLS. It is unaffected by reset types that do not trigger Chip POR not VLLS. See the Reset section details. for more information.

Address: SMC_PMCTRL is 4007_E000h base + 1h offset = 4007_E001h

Bit	7	6	5	4	3	2	1	0
Read	LPWUI	RUNM		0	STOPA	STOPM		
Write								
Reset	0	0	0	0	0	0	0	0

SMC_PMCTRL field descriptions

Field	Description
7 LPWUI	Low Power Wake Up on Interrupt Causes the SMC to exit to normal RUN mode when any active MCU interrupt occurs while in a VLP mode (VLPR, VLPW or VLPS). **NOTE:** If VLPS mode was entered directly from RUN mode, the SMC will always exit back to normal RUN mode regardless of the LPWUI setting. **NOTE:** LPWUI should only be modified while the system is in RUN mode i.e. when PMSTAT=RUN. 0 The system remains in a VLP mode on an interrupt 1 The system exits to normal RUN mode on an interrupt

Table continues on the next page...

SMC_PMCTRL field descriptions (continued)

Field	Description
6–5 RUNM	Run Mode Control When written, this field causes entry into the selected run mode. Writes to this field are blocked if the protection level has not been enabled using the PMPROT register. This field is cleared by hardware on any exit to normal RUN mode. **NOTE:** RUNM should only be set to VLPR when PMSTAT=RUN. Once written to VLPR, RUNM should not be written back to RUN until PMSTAT=VLPR. **NOTE:** RUNM should only be set to RUN when PMSTAT=VLPR. Once written to RUN, RUNM should not be written back to VLPR until PMSTAT=RUN. 00 Normal run mode (RUN) 01 Reserved 10 Very low power run mode (VLPR) 11 Reserved
4 Reserved	This read-only field is reserved and always has the value zero.
3 STOPA	Stop Aborted When set, this read-only status bit indicates an interrupt or reset occured during the previous stop mode entry sequence, preventing the system from entering that mode. This bit is cleared by hardware at the beginning of any stop mode entry sequence and is set if the sequence was aborted. 0 The previous stop mode entry was successsful. 1 The previous stop mode entry was aborted.
2–0 STOPM	Stop Mode Control When written, this field controls entry into the selected stop mode when sleep-now or sleep-on-exit mode is entered with SLEEPDEEP=1 . Writes to this field are blocked if the protection level has not been enabled using the PMPROT register. After any system reset, this field is cleared by hardware on any successful write to the PMPROT register. **NOTE:** When set to VLLSx, the VLLSM bits in the VLLSCTRL register is used to further select the particular VLLS sub-mode which will be entered. **NOTE:** 000 Normal stop (STOP) 001 Reserved 010 Very low power stop (VLPS) 011 Low leakage stop (LLS) 100 Very low leakage stop (VLLSx) 101 Reserved 110 Reseved 111 Reserved

14.3.3 VLLS Control Register (SMC_VLLSCTRL)

The VLLSCTRL register selects which VLLSx mode is entered if STOPM=VLLS and controls power to FlexRAM during VLLS2.

NOTE

> This register is reset on Chip POR not VLLS and by reset types that trigger Chip POR not VLLS. It is unaffected by reset types that do not trigger Chip POR not VLLS. See the Reset section details. for more information.

Address: SMC_VLLSCTRL is 4007_E000h base + 2h offset = 4007_E002h

Bit	7	6	5	4	3	2	1	0
Read	0		PORPO	0	0	VLLSM		
Write	▨	▨		▨	▨			
Reset	0	0	0	0	0	0	1	1

SMC_VLLSCTRL field descriptions

Field	Description
7–6 Reserved	This read-only field is reserved and always has the value zero.
5 PORPO	POR Power Option This bit controls whether the POR detect circuit is enabled in VLLS0 mode. 0 POR detect circuit is enabled in VLLS0 1 POR detect circuit is disabled in VLLS0
4 Reserved	This read-only field is reserved and always has the value zero.
3 Reserved	This read-only field is reserved and always has the value zero.
2–0 VLLSM	VLLS Mode Control. This field controls which VLLS sub-mode to enter if STOPM=VLLS. 000 VLLS0 001 VLLS1 010 VLLS2 011 VLLS3 100 Reserved 101 Reserved 110 Reserved 111 Reserved

14.3.4 Power Mode Status Register (SMC_PMSTAT)

PMSTAT is a read-only, one-hot register which indicates the current power mode of the system.

NOTE

This register is reset on Chip POR not VLLS and by reset types that trigger Chip POR not VLLS. It is unaffected by reset types that do not trigger Chip POR not VLLS. See the Reset section details. for more information.

Address: SMC_PMSTAT is 4007_E000h base + 3h offset = 4007_E003h

Bit	7	6	5	4	3	2	1	0
Read	0	PMSTAT						
Write								
Reset	0	0	0	0	0	0	0	1

SMC_PMSTAT field descriptions

Field	Description
7 Reserved	This read-only field is reserved and always has the value zero.
6–0 PMSTAT	**NOTE:** When debug is enabled, the PMSTAT will not update to STOP or VLPS 000_0001 Current power mode is RUN 000_0010 Current power mode is STOP 000_0100 Current power mode is VLPR 000_1000 Current power mode is VLPW 001_0000 Current power mode is VLPS 010_0000 Current power mode is LLS 100_0000 Current power mode is VLLS

14.4 Functional Description

14.4.1 Power mode transitions

The following figure shows the power mode state transitions available on the chip. Any reset always brings the MCU back to the normal run state.

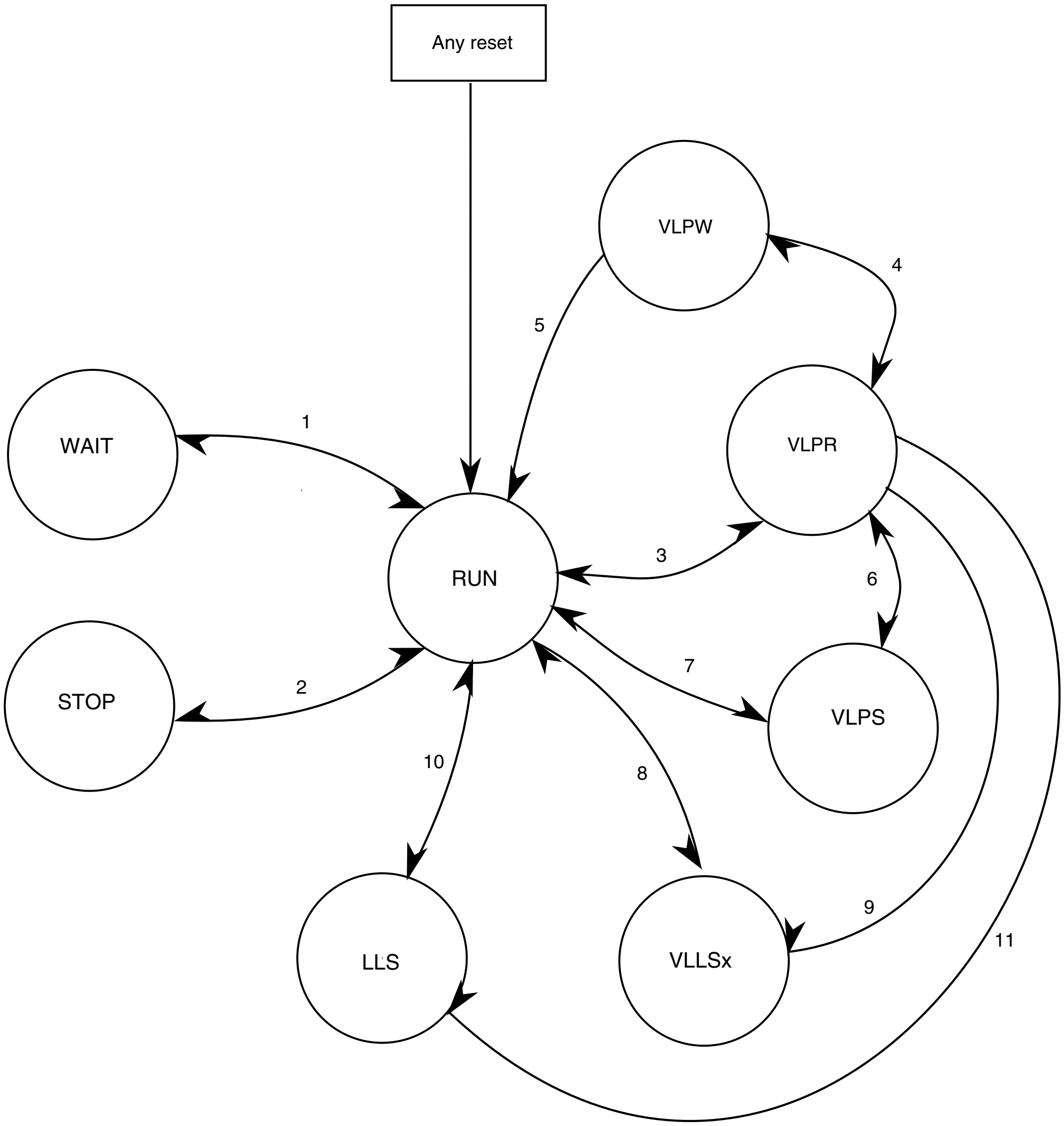

Any reset
VLPW
4
5
WAIT
1
VLPR
RUN
3
6
STOP
2
7
VLPS
10
8
9
11
LLS
VLLSx

Figure 14-5. Power mode state diagram

The following table defines triggers for the various state transitions shown in the previous figure.

Table 14-7. Power mode transition triggers

Transition #	From	To	Trigger conditions
1	RUN	WAIT	Sleep-now or sleep-on-exit modes entered with SLEEPDEEP clear, controlled in System Control Register in ARM core. See note.[1]
	WAIT	RUN	Interrupt or Reset

Table continues on the next page...

Table 14-7. Power mode transition triggers (continued)

Transition #	From	To	Trigger conditions
2	RUN	STOP	PMCTRL[RUNM]=00, PMCTRL[STOPM]=000 Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core. See note.[1]
	STOP	RUN	Interrupt or Reset
3	RUN	VLPR	Reduce system, bus and core frequency to 2 MHz or less, Flash access limited to 1 MHz. Set PMPROT[AVLP]=1, PMCTRL[RUNM]=10.
	VLPR	RUN	Set PMCTRL[RUNM]=00 or Interrupt with PMCTRL[LPWUI] =1 or Reset.
4	VLPR	VLPW	Sleep-now or sleep-on-exit modes entered with SLEEPDEEP clear, which is controlled in System Control Register in ARM core. See note.[1]
	VLPW	VLPR	Interrupt with PMCTRL[LPWUI]=0
5	VLPW	RUN	Interrupt with PMCTRL[LPWUI]=1 or Reset
6	VLPR	VLPS	PMCTRL[STOPM]=000 or 010, Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core. See note.[1]
	VLPS	VLPR	Interrupt with PMCTRL[LPWUI]=0 **NOTE:** If VLPS was entered directly from RUN, hardware will not allow this transition and will force exit back to RUN
7	RUN	VLPS	PMPROT[AVLP]=1, PMCTRL[STOPM]=010, Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core. See note.[1]
	VLPS	RUN	Interrupt with PMCTRL[LPWUI]=1 or Interrupt with PMCTRL[LPWUI]=0 and VLPS mode was entered directly from RUN or Reset

Table continues on the next page...

Table 14-7. Power mode transition triggers (continued)

Transition #	From	To	Trigger conditions
8	RUN	VLLSx	PMPROT[AVLLS]=1, PMCTRL[STOPM]=100, VLLSCTRL[VLLSM]=x (VLLSx), Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core.
	VLLSx	RUN	Wakeup from enabled LLWU input source or RESET pin
9	VLPR	VLLSx	PMPROT[AVLLS]=1, PMCTRL[STOPM]=100, VLLSCTRL[VLLSM]=x (VLLSx), Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core.
10	RUN	LLS	PMPROT[ALLS]=1, PMCTRL[STOPM]=011, Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core.
	LLS	RUN	Wakeup from enabled LLWU input source or RESET pin.
11	VLPR	LLS	PMPROT[ALLS]=1, PMCTRL[STOPM]=011, Sleep-now or sleep-on-exit modes entered with SLEEPDEEP set, which is controlled in System Control Register in ARM core.

1. If debug is enabled, the core clock remains to support debug.

14.4.2 Power mode entry/exit sequencing

When entering or exiting low-power modes, the system must conform to an orderly sequence to manage transitions safely. The SMC manages the system's entry into and exit from all power modes. The following diagram illustrates the connections of the SMC with other system components in the chip that are necessary to sequence the system through all power modes.

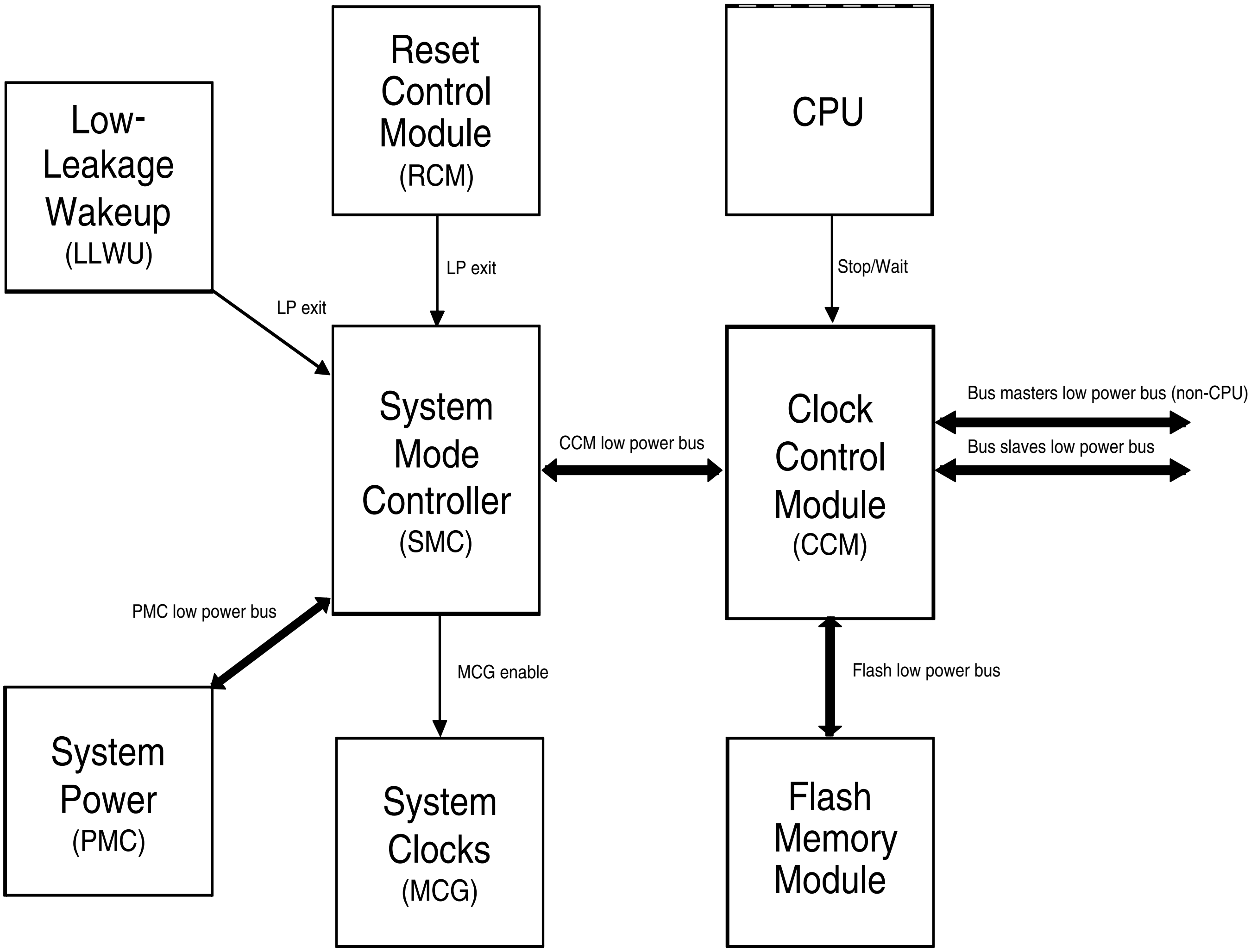

Low-Leakage Wakeup (LLWU)
Reset Control Module (RCM)
CPU
LP exit
LP exit
Stop/Wait
System Mode Controller (SMC)
CCM low power bus
Clock Control Module (CCM)
Bus masters low power bus (non-CPU)
Bus slaves low power bus
PMC low power bus
MCG enable
Flash low power bus
System Power (PMC)
System Clocks (MCG)
Flash Memory Module

Figure 14-6. Low-power system components and connections

14.4.2.1 Stop mode entry sequence

Entry into a low-power stop mode (Stop, VLPS, LLS, VLLSx) is initiated by CPU execution of the WFI instruction. After the instruction is executed, the following sequence occurs:

1. The CPU clock is gated off immediately.
2. Requests are made to all non-CPU bus masters to enter Stop mode.
3. After all masters have acknowledged they are ready to enter Stop mode, requests are made to all bus slaves to enter Stop mode.
4. After all slaves have acknowledged they are ready to enter Stop mode, all system and bus clocks are gated off.
5. Clock generators are disabled in the MCG.
6. The on-chip regulator in the PMC and internal power switches are configured to meet the power consumption goals for the targeted low-power mode.

14.4.2.2 Stop mode exit sequence

Exit from a low-power stop mode is initiated either by a reset or an interrupt event. The following sequence then executes to restore the system to a run mode (RUN or VLPR):

1. The on-chip regulator in the PMC and internal power switches are restored.
2. Clock generators are enabled in the MCG.
3. System and bus clocks are enabled to all masters and slaves.
4. The CPU clock is enabled and the CPU begins servicing the reset or interrupt that initiated the exit from the low-power stop mode.

14.4.2.3 Aborted stop mode entry

If an interrupt or a reset occurs during a stop entry sequence, the SMC can abort the transition early and return to RUN mode without completely entering the stop mode. An aborted entry is possible only if the reset or interrupt occurs before the PMC begins the transition to stop mode regulation. After this point, the interrupt or reset is ignored until the PMC has completed its transition to stop mode regulation. When an aborted stop mode entry sequence occurs, the SMC's PMCTRL[STOPA] is set to 1.

Restriction

Aborted entry to a stop mode is not supported when an interrupt occurs during a transition from VLPR mode to any stop mode.

14.4.2.4 Transition to wait modes

For wait modes (WAIT and VLPW), the CPU clock is gated off while all other clocking continues, as in RUN and VLPR mode operation. Some modules that support stop-in-wait functionality have their clocks disabled in these configurations.

14.4.2.5 Transition from stop modes to Debug mode

The debugger module supports a transition from STOP, WAIT, VLPS, and VLPW back to a Halted state when the debugger has been enabled, that is, ENBDM is 1. As part of this transition, system clocking is re-established and is equivalent to the normal RUN and VLPR mode clocking configuration.

14.4.3 Run modes

The device contains two different run modes:

- Run
- Very Low-Power Run (VLPR)

14.4.3.1 RUN mode

This is the normal operating mode for the device.

This mode is selected after any reset. When the ARM processor exits reset, it sets up the stack, program counter (PC), and link register (LR):

- The processor reads the start SP (SP_main) from vector-table offset 0x000
- The processor reads the start PC from vector-table offset 0x004
- LR is set to 0xFFFF_FFFF.

To reduce power in this mode, disable the clocks to unused modules using their corresponding clock gating control bits in the SIM's registers.

14.4.3.2 Very-Low Power Run (VLPR) mode

In VLPR mode, the on-chip voltage regulator is put into a stop mode regulation state. In this state, the regulator is designed to supply enough current to the MCU over a reduced frequency. To further reduce power in this mode, disable the clocks to unused modules using their corresponding clock gating control bits in the SIM's registers.

Before entering this mode, the following conditions must be met:

- The MCG must be configured in a mode which is supported during VLPR. See the Power Management details for information about these MCG modes.
- All clock monitors in the MCG must be disabled.
- The maximum frequencies of the system, bus, flash, and core are restricted. See the Power Management details about which frequencies are supported.
- Mode protection must be set to allow VLP modes, that is, PMPROT[AVLP] is 1.
- PMCTRL[RUNM] is set to 10b to enter VLPR.
- Flash programming/erasing is not allowed.

NOTE

Do not change the clock frequency while in VLPR mode, because the regulator is slow responding and cannot manage fast load transitions. In addition, do not modify the clock source in the MCG module, the module clock enables in the SIM, or any clock divider registers.

To reenter Normal Run mode, clear RUNM. The PMSTAT register is a read-only status register that can be used to determine when the system has completed an exit to RUN mode. When PMSTAT=RUN, the system is in run regulation and the MCU can run at full speed in any clock mode. If a higher execution frequency is desired, poll the PMSTAT register until it is set to RUN when returning from VLPR mode.

VLPR mode also provides the option to return to run regulation if any interrupt occurs. Implement this option by setting Low-Power Wakeup On Interrupt (LPWUI) in the PMCTRL register. Any reset always causes an exit from VLPR and returns the device to RUN mode after the MCU exits its reset flow. The RUNM bits are cleared by hardware on any interrupt when LPWUI is set or on any reset.

14.4.4 Wait modes

This device contains two different wait modes:

- Wait
- Very-Low Power Wait (VLPW)

14.4.4.1 WAIT mode

WAIT mode is entered when the ARM core enters the Sleep-Now or Sleep-On-Exit modes while SLEEDEEP is cleared. The ARM CPU enters a low-power state in which it is not clocked, but peripherals continue to be clocked provided they are enabled. Clock gating to the peripheral is enabled via the SIM..

When an interrupt request occurs, the CPU exits WAIT mode and resumes processing in RUN mode, beginning with the stacking operations leading to the interrupt service routine.

A system reset will cause an exit from WAIT mode, returning the device to normal RUN mode.

14.4.4.2 Very-Low-Power Wait (VLPW) mode

VLPW is entered by the entering the Sleep-Now or Sleep-On-Exit mode while SLEEPDEEP is cleared and the MCU is in VLPR mode.

In VLPW, the on-chip voltage regulator remains in its stop regulation state. In this state, the regulator is designed to supply enough current to the MCU over a reduced frequency. To further reduce power in this mode, disable the clocks to unused modules by clearing the peripherals' corresponding clock gating control bits in the SIM.

VLPR mode restrictions also apply to VLPW.

VLPW mode provides the option to return to fully-regulated normal RUN mode if any enabled interrupt occurs. This is done by setting PMCTRL[LPWUI]. Wait for the PMSTAT register to set to RUN before increasing the frequency.

If the LPWUI bit is clear, when an interrupt from VLPW occurs, the device returns to VLPR mode to execute the interrupt service routine.

A system reset will cause an exit from WAIT mode, returning the device to normal RUN mode.

14.4.5 Stop modes

This device contains a variety of stop modes to meet your application needs. The stop modes range from:

- a stopped CPU, with all I/O, logic, and memory states retained, and certain asynchronous mode peripherals operating

to:

- a powered down CPU, with only I/O and a small register file retained, very few asynchronous mode peripherals operating, while the remainder of the MCU is powered down.

The choice of stop mode depends upon the user's application, and how power usage and state retention versus functional needs may be traded off.

The various stop modes are selected by setting the appropriate fields in PMPROT and PMCTRL. The selected stop mode mode is entered during the sleep-now or sleep-on-exit entry with the SLEEPDEEP bit set in the System Control Register in the ARM core.

The available stop modes are:

- Normal Stop (STOP)
- Very-Low Power Stop (VLPS)
- Low-Leakage Stop (LLS)
- Very-Low-Leakage Stop (VLLSx)

14.4.5.1 STOP mode

STOP mode is entered via the sleep-now or sleep-on-exit with the SLEEPDEEP bit set in the System Control Register in the ARM core.

The MCG module can be configured to leave the reference clocks running.

A module capable of providing an asynchronous interrupt to the device takes the device out of STOP mode and returns the device to normal RUN mode. Refer to the device's Power Management chapter for peripheral, I/O, and memory operation in STOP mode. When an interrupt request occurs, the CPU exits STOP mode and resumes processing, beginning with the stacking operations leading to the interrupt service routine.

A system reset will cause an exit from STOP mode, returning the device to normal RUN mode via an MCU reset.

14.4.5.2 Very-Low-Power Stop (VLPS) mode

VLPS mode can be entered in one of two ways:

- Entry into stop via the sleep-now or sleep-on-exit with the SLEEPDEEP bit set in the System Control Register in the ARM core while the MCU is in VLPR mode and STOPM=010 or 000 in the PMCTRL register.
- Entry into stop via the sleep-now or sleep-on-exit with the SLEEPDEEP bit set in the System Control Register in the ARM core while the MCU is in normal RUN mode and STOPM=010 in the PMCTRL register. When VLPS is entered directly from RUN mode, exit to VLPR is disabled by hardware and the system will always exit back to RUN.

In VLPS, the on-chip voltage regulator remains in its stop regulation state as in VLPR.

A module capable of providing an asynchronous interrupt to the device takes the device out of VLPS and returns the device to VLPR mode, provided LPWUI is clear.

If LPWUI is set, the device returns to normal RUN mode upon an interrupt request. PMSTAT must be set to RUN before allowing the system to return to a frequency higher than that allowed in VLPR mode.

A system reset will also cause a VLPS exit, returning the device to normal RUN mode.

14.4.5.3 Low-Leakage Stop (LLS) mode

Low-Leakage Stop (LLS) mode can be entered from normal RUN or VLPR modes.

The MCU enters LLS mode if:

- In Sleep-Now or Sleep-On-Exit mode, SLEEPDEEP is set in the System Control Register in the ARM core, and
- The device is configured as shown in Table 14-7.

In LLS, the on-chip voltage regulator is in stop regulation. Most of the peripherals are put in a state-retention mode that does not allow them to operate while in LLS.

Before entering LLS mode, the user should configure the low-leakage wakeup (LLWU) module to enable the desired wakeup sources. The available wakeup sources in LLS are detailed in the chip configuration details for this device.

After wakeup from LLS, the device returns to normal RUN mode with a pending LLWU module interrupt. In the LLWU interrupt service routine (ISR), the user can poll the LLWU module wakeup flags to determine the source of the wakeup.

NOTE

The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit stop mode on an LLS recovery.

An asserted $\overline{\text{RESET}}$ pin will cause an exit from LLS mode, returning the device to normal RUN mode. When LLS is exiting via the $\overline{\text{RESET}}$ pin, the PIN and WAKEUP bits are set in the SRS0 register of the reset control module (RCM).

14.4.5.4 Very-Low-Leakage Stop (VLLSx) modes

This device contains these very low leakage modes:

- VLLS3
- VLLS2
- VLLS1
- VLLS0

VLLSx is often used in this document to refer to all of these modes.

All VLLSx modes can be entered from normal RUN or VLPR modes.

The MCU enters the configured VLLS mode if:

- In Sleep-Now or Sleep-On-Exit mode, the SLEEPDEEP bit is set in the System Control Register in the ARM core, and
- The device is configured as shown in Table 14-7.

In VLLS, the on-chip voltage regulator is in its stop-regulation state while most digital logic is powered off.

In VLLS, configure the LLWU module to enable the desired wakeup sources. The available wakeup sources in VLLS are detailed LLWU's chip configuration details for this device.

When entering VLLS, each I/O pin is latched as configured before executing VLLS. Because all digital logic in the MCU is powered off, all port and peripheral data is lost during VLLS. This information must be restored before ACKISO bit in the PMC is set.

An asserted $\overline{\text{RESET}}$ pin will cause an exit from any VLLS mode, returning the device to normal RUN mode. When exiting VLLS via the $\overline{\text{RESET}}$ pin, the PIN and WAKEUP bits are set in the SRS0 register of the reset control module (RCM).

14.4.6 Debug in low power modes

When the MCU is secure, the device disables/limits debugger operation. When the MCU is unsecure, the ARM debugger can assert two power-up request signals:

- System power up, via SYSPWR in the Debug Port Control/Stat register
- Debug power up, via CDBGPWRUPREQ in the Debug Port Control/Stat register

When asserted while in RUN, WAIT, VLPR, or VLPW, the mode controller drives a corresponding acknowledge for each signal, that is, both CDBGPWRUPACK and CSYSPWRUPACK. When both requests are asserted, the mode controller handles attempts to enter STOP and VLPS by entering an emulated stop state. In this emulated stop state:

- the regulator is in run regulation,
- the MCG-generated clock source is enabled,
- all system clocks, except the core clock, are disabled,
- the debug module has access to core registers, and
- access to the on-chip peripherals is blocked.

No debug is available while the MCU is in LLS or VLLS modes. LLS is a state-retention mode and all debug operation can continue after waking from LLS, even in cases where system wakeup is due to a system reset event.

Entering into a VLLS mode causes all of the debug controls and settings to be powered off. To give time to the debugger to sync with the MCU, the MDM AP Control Register includes a Very-Low-Leakage Debug Request (VLLDBGREQ) bit that is set to configure the Reset Controller logic to hold the system in reset after the next recovery from a VLLS mode. This bit allows the debugger time to reinitialize the debug module before the debug session continues.

The MDM AP Control Register also includes a Very Low Leakage Debug Acknowledge (VLLDBGACK) bit that is set to release the ARM core being held in reset following a VLLS recovery. The debugger reinitializes all debug IP, and then asserts the VLLDBGACK control bit to allow the RCM to release the ARM core from reset and allow CPU operation to begin.

The VLLDBGACK bit is cleared by the debugger (or can be left set as is) or clears automatically due to the reset generated as part of the next VLLS recovery.

Chapter 15
Power Management Controller

15.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The power management controller (PMC) contains the internal voltage regulator, power on reset (POR), and low voltage detect system.

15.2 Features

The PMC features include:

- Internal voltage regulator
- Active POR providing brown-out detect
- Low-voltage detect supporting two low-voltage trip points with four warning levels per trip point

15.3 Low-voltage detect (LVD) system

This device includes a system to guard against low-voltage conditions. This protects memory contents and controls MCU system states during supply voltage variations. The system is comprised of a power-on reset (POR) circuit and a LVD circuit with a user-selectable trip voltage: high (V_{LVDH}) or low (V_{LVDL}). The trip voltage is selected by the LVDSC1[LVDV] bits. The LVD is disabled upon entering VLPx, LLS, and VLLSx modes.

Two flags are available to indicate the status of the low-voltage detect system:

- The low voltage detect flag (LVDF) operates in a level sensitive manner. The LVDF bit is set when the supply voltage falls below the selected trip point (VLVD). The LVDF bit is cleared by writing one to the LVDACK bit, but only if the internal supply has returned above the trip point; otherwise, the LVDF bit remains set.
- The low voltage warning flag (LVWF) operates in a level sensitive manner. The LVWF bit is set when the supply voltage falls below the selected monitor trip point (VLVW). The LVWF bit is cleared by writing one to the LVWACK bit, but only if the internal supply has returned above the trip point; otherwise, the LVWF bit remains set.

15.3.1 LVD reset operation

By setting the LVDRE bit, the LVD generates a reset upon detection of a low voltage condition. The low voltage detection threshold is determined by the LVDV bits. After an LVD reset occurs, the LVD system holds the MCU in reset until the supply voltage rises above this threshold. The LVD bit in the SRS register is set following an LVD or power-on reset.

15.3.2 LVD interrupt operation

By configuring the LVD circuit for interrupt operation (LVDIE set and LVDRE clear), LVDSC1[LVDF] is set and an LVD interrupt request occurs upon detection of a low voltage condition. The LVDF bit is cleared by writing one to the LVDSC1[LVDACK] bit.

15.3.3 Low-voltage warning (LVW) interrupt operation

The LVD system contains a low-voltage warning flag (LVWF) to indicate that the supply voltage is approaching, but is above, the LVD voltage. The LVW also has an interrupt, which is enabled by setting the LVDSC2[LVWIE] bit. If enabled, an LVW interrupt request occurs when the LVWF is set. LVWF is cleared by writing one to the LVDSC2[LVWACK] bit.

The LVDSC2[LVWV] bits select one of four trip voltages:

- Highest: V_{LVW4}
- Two mid-levels: V_{LVW3} and V_{LVW2}
- Lowest: V_{LVW1}

15.4 I/O retention

When in LLS mode, the I/O pins are held in their input or output state. Upon wakeup, the PMC is re-enabled, goes through a power up sequence to full regulation, and releases the logic from state retention mode. The I/O are released immediately after a wakeup or reset event. In the case of LLS exit via a RESET pin, the I/O default to their reset state.

When in VLLS modes, the I/O states are held on a wakeup event (with the exception of wakeup by reset event) until the wakeup has been acknowledged via a write to the ACKISO bit. In the case of VLLS exit via a RESET pin, the I/O are released and default to their reset state. In this case, no write to the ACKISO is needed.

15.5 Memory map and register descriptions

PMC register details follow.

NOTE

Different portions of PMC registers are reset only by particular reset types. Each register's description provides details. For more information about the types of reset on this chip, refer to the Reset section details.

PMC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_D000	Low Voltage Detect Status And Control 1 register (PMC_LVDSC1)	8	R/W	10h	15.5.1/283
4007_D001	Low Voltage Detect Status And Control 2 register (PMC_LVDSC2)	8	R/W	00h	15.5.2/285
4007_D002	Regulator Status And Control register (PMC_REGSC)	8	R/W	04h	15.5.3/286

15.5.1 Low Voltage Detect Status And Control 1 register (PMC_LVDSC1)

This register contains status and control bits to support the low voltage detect function. This register should be written during the reset initialization program to set the desired controls even if the desired settings are the same as the reset settings.

While the device is in the very low power or low leakage modes, the LVD system is disabled regardless of LVDSC1 settings. To protect systems that must have LVD always on, configure the SMC's power mode protection register (PMPROT) to disallow any very low power or low leakage modes from being enabled.

See the device's data sheet for the exact LVD trip voltages.

NOTE

> The LVDV bits are reset solely on a POR Only event. The register's other bits are reset on Chip Reset Not VLLS. For more information about these reset types, refer to the Reset section details.

Address: PMC_LVDSC1 is 4007_D000h base + 0h offset = 4007_D000h

Bit	7	6	5	4	3	2	1	0
Read	LVDF	0	LVDIE	LVDRE	0		LVDV	
Write	▨	LVDACK			▨			
Reset	0	0	0	1	0	0	0	0

PMC_LVDSC1 field descriptions

Field	Description
7 LVDF	Low-Voltage Detect Flag This read-only status bit indicates a low-voltage detect event. 0 Low-voltage event not detected 1 Low-voltage event detected
6 LVDACK	Low-Voltage Detect Acknowledge This write-only bit is used to acknowledge low voltage detection errors. Write 1 to clear LVDF. Reads always return 0.
5 LVDIE	Low-Voltage Detect Interrupt Enable Enables hardware interrupt requests for LVDF. 0 Hardware interrupt disabled (use polling) 1 Request a hardware interrupt when LVDF = 1
4 LVDRE	Low-Voltage Detect Reset Enable This write-once bit enables LVDF events to generate a hardware reset. Additional writes are ignored. 0 LVDF does not generate hardware resets 1 Force an MCU reset when LVDF = 1
3–2 Reserved	This read-only field is reserved and always has the value zero.
1–0 LVDV	Low-Voltage Detect Voltage Select Selects the LVD trip point voltage (V_{LVD}).

Table continues on the next page...

PMC_LVDSC1 field descriptions (continued)

Field	Description
	00 Low trip point selected ($V_{LVD} = V_{LVDL}$) 01 High trip point selected ($V_{LVD} = V_{LVDH}$) 10 Reserved 11 Reserved

15.5.2 Low Voltage Detect Status And Control 2 register (PMC_LVDSC2)

This register contains status and control bits to support the low voltage warning function.

While the device is in the very low power or low leakage modes, the LVD system is disabled regardless of LVDSC2 settings.

See the device's data sheet for the exact LVD trip voltages.

> **NOTE**
>
> The LVW trip voltages depend on LVWV and LVDV bits.

> **NOTE**
>
> The LVWV bits are reset solely on a POR Only event. The register's other bits are reset on Chip Reset Not VLLS. For more information about these reset types, refer to the Reset section details.

Address: PMC_LVDSC2 is 4007_D000h base + 1h offset = 4007_D001h

<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Read</td><td>LVWF</td><td>0</td><td rowspan="2">LVWIE</td><td colspan="3">0</td><td colspan="2" rowspan="2">LVWV</td></tr>
<tr><td>Write</td><td></td><td>LVWACK</td><td colspan="3"></td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

PMC_LVDSC2 field descriptions

Field	Description
7 LVWF	Low-Voltage Warning Flag This read-only status bit indicates a low-voltage warning event. LVWF is set when V_{Supply} transitions below the trip point, or after reset and V_{Supply} is already below V_{LVW}. 0 Low-voltage warning event not detected 1 Low-voltage warning event detected
6 LVWACK	Low-Voltage Warning Acknowledge This write-only bit is used to acknowledge low voltage warning errors. Write 1 to clear LVWF. Reads always return 0.

Table continues on the next page...

PMC_LVDSC2 field descriptions (continued)

Field	Description
5 LVWIE	Low-Voltage Warning Interrupt Enable Enables hardware interrupt requests for LVWF. 0 Hardware interrupt disabled (use polling) 1 Request a hardware interrupt when LVWF = 1
4–2 Reserved	This read-only field is reserved and always has the value zero.
1–0 LVWV	Low-Voltage Warning Voltage Select Selects the LVW trip point voltage (V_{LVW}). The actual voltage for the warning depends on LVDSC1[LVDV]. 00 Low trip point selected ($V_{LVW} = V_{LVW1}$) 01 Mid 1 trip point selected ($V_{LVW} = V_{LVW2}$) 10 Mid 2 trip point selected ($V_{LVW} = V_{LVW3}$) 11 High trip point selected ($V_{LVW} = V_{LVW4}$)

15.5.3 Regulator Status And Control register (PMC_REGSC)

The PMC contains an internal voltage regulator. The voltage regulator design uses a bandgap reference that is also available through a buffer as input to certain internal peripherals, such as the CMP and ADC. The internal regulator provides a status bit (REGONS) indicating the regulator is in run regulation.

NOTE

This register is reset on Chip Reset Not VLLS and by reset types that trigger Chip Reset not VLLS. See the Reset section for more information.

Address: PMC_REGSC is 4007_D000h base + 2h offset = 4007_D002h

Bit	7	6	5	4	3	2	1	0
Read	0			BGEN	ACKISO	REGONS	Reserved	BGBE
Write					w1c			
Reset	0	0	0	0	0	1	0	0

PMC_REGSC field descriptions

Field	Description
7–5 Reserved	This read-only field is reserved and always has the value zero.
4 BGEN	Bandgap Enable In VLPx Operation

Table continues on the next page...

PMC_REGSC field descriptions (continued)

Field	Description
	BGEN controls whether the bandgap is enabled in lower power modes of operation (VLPx, LLS, and VLLSx). When on-chip peripherals require the bandgap voltage reference in low power modes of operation, set BGEN to continue to enable the bandgap operation. **NOTE:** When the bandgap voltage reference is not needed in low power modes, clear BGEN to avoid excess power consumption. 0 Bandgap voltage reference is disabled in VLPx , LLS , and VLLSx modes 1 Bandgap voltage reference is enabled in VLPx , LLS , and VLLSx modes
3 ACKISO	Acknowledge Isolation Reading this bit indicates whether certain peripherals and the I/O pads are in a latched state as a result of having been in a VLLS mode. Writing one to this bit when it is set releases the I/O pads and certain peripherals to their normal run mode state. **NOTE:** After recovering from a VLLS mode, user should restore chip configuration before clearing ACKISO. In particular, pin configuration for enabled LLWU wakeup pins should be restored to avoid any LLWU flag from being falsely set when ACKISO is cleared. 0 Peripherals and I/O pads are in normal run state 1 Certain peripherals and I/O pads are in an isolated and latched state
2 REGONS	Regulator In Run Regulation Status This read-only bit provides the current status of the internal voltage regulator. 0 Regulator is in stop regulation or in transition to/from it 1 Regulator is in run regulation
1 Reserved	This field is reserved.
0 BGBE	Bandgap Buffer Enable Enables the bandgap buffer. 0 Bandgap buffer not enabled 1 Bandgap buffer enabled

Chapter 16
Low-Leakage Wakeup Unit (LLWU)

16.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The LLWU module allows the user to select up to 16 external pin sources and up to 8 internal modules as a wakeup source from low-leakage power modes. The input sources are described in the device's chip configuration details. Each of the available wakeup sources can be individually enabled.

The $\overline{\text{RESET}}$ pin is an additional source for triggering an exit from low-leakage power modes, and causes the MCU to exit both LLS and VLLS through a reset flow. The LLWU_RST[LLRSTE] bit must be set to allow an exit from low-leakage modes via the $\overline{\text{RESET}}$ pin. On a device where the $\overline{\text{RESET}}$ pin is shared with other functions, the explicit port mux control register must be set for the $\overline{\text{RESET}}$ pin before the $\overline{\text{RESET}}$ pin can be used as a low-leakage reset source.

The LLWU module also includes three optional digital pin filters: two for the external wakeup pins and one for the $\overline{\text{RESET}}$ pin.

16.1.1 Features

The LLWU module features include:

- Support for up to 16 external input pins and up to 8 internal modules with individual enable bits
- Input sources may be external pins or from internal peripherals capable of running in LLS or VLLS. See the chip configuration information for wakeup input sources for this device.

- External pin wakeup inputs, each of which is programmable as falling-edge, rising-edge, or any change
- Wakeup inputs that are activated if enabled after MCU enters a low-leakage power mode
- Optional digital filters provided to qualify an external pin detect and $\overline{\text{RESET}}$ pin detect. When entering VLLS0, the filters are disabled and bypassed.

16.1.2 Modes of operation

The LLWU module becomes functional on entry into a low-leakage power mode. After recovery from LLS, the LLWU is immediately disabled. After recovery from VLLS, the LLWU continues to detect wakeup events until the user has acknowledged the wakeup via a write to the PMC_REGSC[ACKISO] bit.

16.1.2.1 LLS mode

The LLWU module provides up to 16 external wakeup inputs and up to 8 internal module wakeup inputs. An LLS reset event can be initiated via assertion of the $\overline{\text{RESET}}$ pin.

Wakeup events due to external wakeup inputs and internal module wakeup inputs result in an interrupt flow when exiting LLS. A reset event due to $\overline{\text{RESET}}$ pin assertion results in a reset flow when exiting LLS.

NOTE

The LLWU interrupt must not be masked by the interrupt controller to avoid a scenario where the system does not fully exit Stop mode on an LLS recovery.

16.1.2.2 VLLS modes

The LLWU module provides up to 16 external wakeup inputs and up to 8 internal module wakeup inputs. A VLLS reset event can be initiated via assertion of the $\overline{\text{RESET}}$ pin. All wakeup and reset events result in VLLS exit via a reset flow.

16.1.2.3 Non-low leakage modes

The LLWU is not active in all non-low leakage modes where detection and control logic are in a static state. The LLWU registers are accessible in non-low leakage modes and are available for configuring and reading status when bus transactions are possible.

When the$\overline{\text{RESET}}$ pin filter or wakeup pin filters are enabled, filter operation begins immediately. If a low leakage mode is entered within 5 LPO clock cycles of an active edge, the edge event will be detected by the LLWU. For $\overline{\text{RESET}}$ pin filtering, this means that there is no restart to the minimum LPO cycle duration as the filtering transitions from a non-low leakage filter, which is implemented in the RCM, to the LLWU filter.

16.1.2.4 Debug mode

When the chip is in Debug mode and then enters LLS or a VLLSx mode, no debug logic works in the fully-functional low-leakage mode. Upon an exit from the LLS or VLLSx mode, the LLWU becomes inactive.

16.1.3 Block diagram

The following figure is the block diagram for the LLWU module.

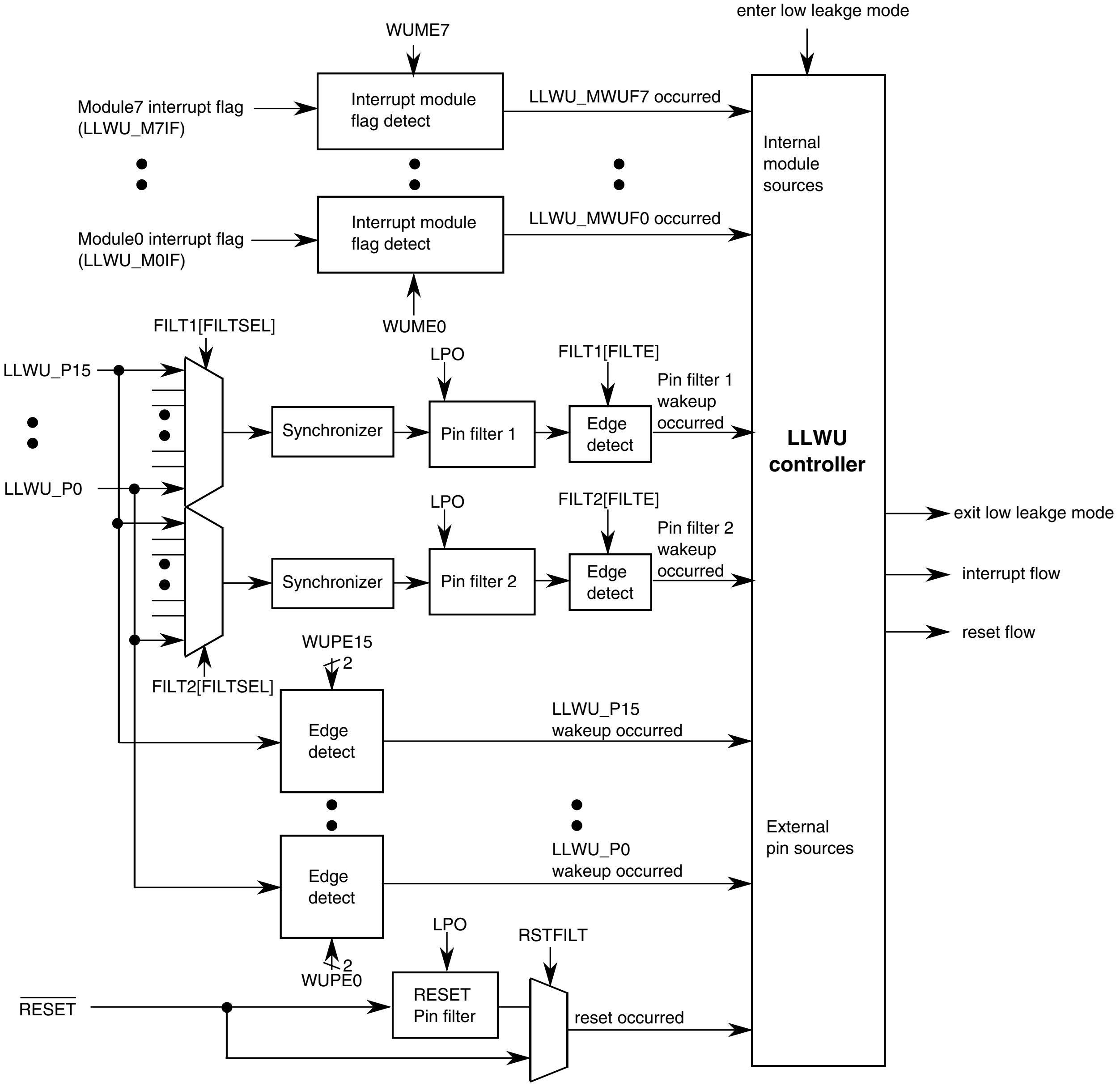

enter low leakge mode
WUME7
Module7 interrupt flag (LLWU_M7IF)
Interrupt module flag detect
LLWU_MWUF7 occurred
Internal module sources
Module0 interrupt flag (LLWU_M0IF)
Interrupt module flag detect
LLWU_MWUF0 occurred
WUME0
FILT1[FILTSEL]
LLWU_P15
LLWU_P0
LPO
FILT1[FILTE]
Synchronizer
Pin filter 1
Edge detect
Pin filter 1 wakeup occurred
LLWU controller
exit low leakge mode
interrupt flow
reset flow
LPO
FILT2[FILTE]
Synchronizer
Pin filter 2
Edge detect
Pin filter 2 wakeup occurred
WUPE15
2
FILT2[FILTSEL]
Edge detect
LLWU_P15 wakeup occurred
Edge detect
LLWU_P0 wakeup occurred
External pin sources
WUPE0
2
LPO
RSTFILT
RESET
RESET Pin filter
reset occurred

Figure 16-1. LLWU block diagram

16.2 LLWU signal descriptions

The signal properties of LLWU are shown in the following table. The external wakeup input pins can be enabled to detect either rising-edge, falling-edge, or on any change.

Table 16-1. LLWU signal descriptions

Signal	Description	I/O
LLWU_Pn	Wakeup inputs (n = 0-15)	I

16.3 Memory map/register definition

The LLWU includes the following registers:

- Five 8-bit wakeup source enable registers
  - Enable external pin input sources
  - Enable internal peripheral sources
- Three 8-bit wakeup flag registers
  - Indication of wakeup source that caused exit from a low-leakage power mode includes external pin or internal module interrupt
- Two 8-bit wakeup pin filter enable registers
- One 8-bit $\overline{\text{RESET}}$ pin filter enable register

NOTE

All LLWU registers are reset by Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. Each register's displayed reset value represents this subset of reset types. LLWU registers are unaffected by reset types that do not trigger Chip Reset not VLLS. For more information about the types of reset on this chip, refer to the Introduction details.

LLWU memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_C000	LLWU Pin Enable 1 register (LLWU_PE1)	8	R/W	00h	16.3.1/294
4007_C001	LLWU Pin Enable 2 register (LLWU_PE2)	8	R/W	00h	16.3.2/295
4007_C002	LLWU Pin Enable 3 register (LLWU_PE3)	8	R/W	00h	16.3.3/296
4007_C003	LLWU Pin Enable 4 register (LLWU_PE4)	8	R/W	00h	16.3.4/297
4007_C004	LLWU Module Enable register (LLWU_ME)	8	R/W	00h	16.3.5/298
4007_C005	LLWU Flag 1 register (LLWU_F1)	8	R/W	00h	16.3.6/300
4007_C006	LLWU Flag 2 register (LLWU_F2)	8	R/W	00h	16.3.7/301
4007_C007	LLWU Flag 3 register (LLWU_F3)	8	R/W	00h	16.3.8/303
4007_C008	LLWU Pin Filter 1 register (LLWU_FILT1)	8	R/W	00h	16.3.9/305
4007_C009	LLWU Pin Filter 2 register (LLWU_FILT2)	8	R/W	00h	16.3.10/ 306
4007_C00A	LLWU Reset Enable register (LLWU_RST)	8	R/W	02h	16.3.11/ 307

16.3.1 LLWU Pin Enable 1 register (LLWU_PE1)

LLWU_PE1 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P3-LLWU_P0.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: LLWU_PE1 is 4007_C000h base + 0h offset = 4007_C000h

Bit	7	6	5	4	3	2	1	0
Read Write	WUPE3		WUPE2		WUPE1		WUPE0	
Reset	0	0	0	0	0	0	0	0

LLWU_PE1 field descriptions

Field	Description
7–6 WUPE3	Wakeup Pin Enable For LLWU_P3 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE2	Wakeup Pin Enable For LLWU_P2 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE1	Wakeup Pin Enable For LLWU_P1 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
1–0 WUPE0	Wakeup Pin Enable For LLWU_P0 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input

Table continues on the next page...

LLWU_PE1 field descriptions (continued)

Field	Description
	01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

16.3.2 LLWU Pin Enable 2 register (LLWU_PE2)

LLWU_PE2 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P7-LLWU_P4.

NOTE

> This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: LLWU_PE2 is 4007_C000h base + 1h offset = 4007_C001h

Bit	7	6	5	4	3	2	1	0
Read Write	WUPE7		WUPE6		WUPE5		WUPE4	
Reset	0	0	0	0	0	0	0	0

LLWU_PE2 field descriptions

Field	Description
7–6 WUPE7	Wakeup Pin Enable For LLWU_P7 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE6	Wakeup Pin Enable For LLWU_P6 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE5	Wakeup Pin Enable For LLWU_P5 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input

Table continues on the next page...

LLWU_PE2 field descriptions (continued)

Field	Description
	01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
1–0 WUPE4	Wakeup Pin Enable For LLWU_P4 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

16.3.3 LLWU Pin Enable 3 register (LLWU_PE3)

LLWU_PE3 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P11-LLWU_P8.

NOTE

> This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: LLWU_PE3 is 4007_C000h base + 2h offset = 4007_C002h

Bit	7	6	5	4	3	2	1	0
Read Write	WUPE11		WUPE10		WUPE9		WUPE8	
Reset	0	0	0	0	0	0	0	0

LLWU_PE3 field descriptions

Field	Description
7–6 WUPE11	Wakeup Pin Enable For LLWU_P11 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE10	Wakeup Pin Enable For LLWU_P10 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input

Table continues on the next page...

LLWU_PE3 field descriptions (continued)

Field	Description
	01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE9	Wakeup Pin Enable For LLWU_P9 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
1–0 WUPE8	Wakeup Pin Enable For LLWU_P8 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

16.3.4 LLWU Pin Enable 4 register (LLWU_PE4)

LLWU_PE4 contains the field to enable and select the edge detect type for the external wakeup input pins LLWU_P15-LLWU_P12.

NOTE

> This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: LLWU_PE4 is 4007_C000h base + 3h offset = 4007_C003h

Bit	7	6	5	4	3	2	1	0
Read Write	WUPE15		WUPE14		WUPE13		WUPE12	
Reset	0	0	0	0	0	0	0	0

LLWU_PE4 field descriptions

Field	Description
7–6 WUPE15	Wakeup Pin Enable For LLWU_P15 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection

Table continues on the next page...

LLWU_PE4 field descriptions (continued)

Field	Description
	10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
5–4 WUPE14	Wakeup Pin Enable For LLWU_P14 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
3–2 WUPE13	Wakeup Pin Enable For LLWU_P13 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection
1–0 WUPE12	Wakeup Pin Enable For LLWU_P12 Enables and configures the edge detection for the wakeup pin. 00 External input pin disabled as wakeup input 01 External input pin enabled with rising edge detection 10 External input pin enabled with falling edge detection 11 External input pin enabled with any change detection

16.3.5 LLWU Module Enable register (LLWU_ME)

LLWU_ME contains the bits to enable the internal module flag as a wakeup input source for inputs MWUF7-MWUF0.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: LLWU_ME is 4007_C000h base + 4h offset = 4007_C004h

Bit	7	6	5	4	3	2	1	0
Read Write	WUME7	WUME6	WUME5	WUME4	WUME3	WUME2	WUME1	WUME0
Reset	0	0	0	0	0	0	0	0

LLWU_ME field descriptions

Field	Description
7 WUME7	Wakeup Module Enable For Module 7 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
6 WUME6	Wakeup Module Enable For Module 6 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
5 WUME5	Wakeup Module Enable For Module 5 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
4 WUME4	Wakeup Module Enable For Module 4 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
3 WUME3	Wakeup Module Enable For Module 3 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
2 WUME2	Wakeup Module Enable For Module 2 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
1 WUME1	Wakeup Module Enable for Module 1 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source
0 WUME0	Wakeup Module Enable For Module 0 Enables an internal module as a wakeup source input. 0 Internal module flag not used as wakeup source 1 Internal module flag used as wakeup source

16.3.6 LLWU Flag 1 register (LLWU_F1)

LLWU_F1 contains the wakeup flags indicating which wakeup source caused the MCU to exit LLS or VLLS mode. For LLS, this is the source causing the CPU interrupt flow. For VLLS, this is the source causing the MCU reset flow.

The external wakeup flags are read-only and clearing a flag is accomplished by a write of a 1 to the corresponding WUFx bit. The wakeup flag (WUFx), if set, will remain set if the associated WUPEx bit is cleared.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: LLWU_F1 is 4007_C000h base + 5h offset = 4007_C005h

Bit	7	6	5	4	3	2	1	0
Read	WUF7	WUF6	WUF5	WUF4	WUF3	WUF2	WUF1	WUF0
Write	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

LLWU_F1 field descriptions

Field	Description
7 WUF7	Wakeup Flag For LLWU_P7 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF7. 0 LLWU_P7 input was not a wakeup source 1 LLWU_P7 input was a wakeup source
6 WUF6	Wakeup Flag For LLWU_P6 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF6. 0 LLWU_P6 input was not a wakeup source 1 LLWU_P6 input was a wakeup source
5 WUF5	Wakeup Flag For LLWU_P5 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF5. 0 LLWU_P5 input was not a wakeup source 1 LLWU_P5 input was a wakeup source

Table continues on the next page...

LLWU_F1 field descriptions (continued)

Field	Description
4 WUF4	Wakeup Flag For LLWU_P4 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF4. 0 LLWU_P4 input was not a wakeup source 1 LLWU_P4 input was a wakeup source
3 WUF3	Wakeup Flag For LLWU_P3 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF3. 0 LLWU_P3 input was not a wakeup source 1 LLWU_P3 input was a wakeup source
2 WUF2	Wakeup Flag For LLWU_P2 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF2. 0 LLWU_P2 input was not a wakeup source 1 LLWU_P2 input was a wakeup source
1 WUF1	Wakeup Flag For LLWU_P1 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF1. 0 LLWU_P1 input was not a wakeup source 1 LLWU_P1 input was a wakeup source
0 WUF0	Wakeup Flag For LLWU_P0 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF0. 0 LLWU_P0 input was not a wakeup source 1 LLWU_P0 input was a wakeup source

16.3.7 LLWU Flag 2 register (LLWU_F2)

LLWU_F2 contains the wakeup flags indicating which wakeup source caused the MCU to exit LLS or VLLS mode. For LLS, this is the source causing the CPU interrupt flow. For VLLS, this is the source causing the MCU reset flow.

The external wakeup flags are read-only and clearing a flag is accomplished by a write of a 1 to the corresponding WUFx bit. The wakeup flag (WUFx), if set, will remain set if the associated WUPEx bit is cleared.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: LLWU_F2 is 4007_C000h base + 6h offset = 4007_C006h

Bit	7	6	5	4	3	2	1	0
Read	WUF15	WUF14	WUF13	WUF12	WUF11	WUF10	WUF9	WUF8
Write	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

LLWU_F2 field descriptions

Field	Description
7 WUF15	Wakeup Flag For LLWU_P15 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF15. 0 LLWU_P15 input was not a wakeup source 1 LLWU_P15 input was a wakeup source
6 WUF14	Wakeup Flag For LLWU_P14 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF14. 0 LLWU_P14 input was not a wakeup source 1 LLWU_P14 input was a wakeup source
5 WUF13	Wakeup Flag For LLWU_P13 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF13. 0 LLWU_P13 input was not a wakeup source 1 LLWU_P13 input was a wakeup source
4 WUF12	Wakeup Flag For LLWU_P12 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF12. 0 LLWU_P12 input was not a wakeup source 1 LLWU_P12 input was a wakeup source
3 WUF11	Wakeup Flag For LLWU_P11 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF11. 0 LLWU_P11 input was not a wakeup source 1 LLWU_P11 input was a wakeup source

Table continues on the next page...

LLWU_F2 field descriptions (continued)

Field	Description
2 WUF10	Wakeup Flag For LLWU_P10 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF10. 0 LLWU_P10 input was not a wakeup source 1 LLWU_P10 input was a wakeup source
1 WUF9	Wakeup Flag For LLWU_P9 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF9. 0 LLWU_P9 input was not a wakeup source 1 LLWU_P9 input was a wakeup source
0 WUF8	Wakeup Flag For LLWU_P8 Indicates that an enabled external wakeup pin was a source of exiting a low-leakage power mode. To clear the flag write a one to WUF8. 0 LLWU_P8 input was not a wakeup source 1 LLWU_P8 input was a wakeup source

16.3.8 LLWU Flag 3 register (LLWU_F3)

LLWU_F3 contains the wakeup flags indicating which internal wakeup source caused the MCU to exit LLS or VLLS mode. For LLS, this is the source causing the CPU interrupt flow. For VLLS, this is the source causing the MCU reset flow.

For internal peripherals that are capable of running in a low-leakage power mode, such as RTC or CMP modules, the flag from the associated peripheral is accessible as the MWUFx bit. The flag will need to be cleared in the peripheral instead of writing a 1 to the MWUFx bit.

NOTE

> This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: LLWU_F3 is 4007_C000h base + 7h offset = 4007_C007h

Bit	7	6	5	4	3	2	1	0
Read	MWUF7	MWUF6	MWUF5	MWUF4	MWUF3	MWUF2	MWUF1	MWUF0
Write								
Reset	0	0	0	0	0	0	0	0

LLWU_F3 field descriptions

Field	Description
7 MWUF7	Wakeup flag For module 7 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 7 input was not a wakeup source 1 Module 7 input was a wakeup source
6 MWUF6	Wakeup flag For module 6 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 6 input was not a wakeup source 1 Module 6 input was a wakeup source
5 MWUF5	Wakeup flag For module 5 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 5 input was not a wakeup source 1 Module 5 input was a wakeup source
4 MWUF4	Wakeup flag For module 4 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 4 input was not a wakeup source 1 Module 4 input was a wakeup source
3 MWUF3	Wakeup flag For module 3 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 3 input was not a wakeup source 1 Module 3 input was a wakeup source
2 MWUF2	Wakeup flag For module 2 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 2 input was not a wakeup source 1 Module 2 input was a wakeup source
1 MWUF1	Wakeup flag For module 1 Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 1 input was not a wakeup source 1 Module 1 input was a wakeup source
0 MWUF0	Wakeup flag For module 0

Table continues on the next page...

LLWU_F3 field descriptions (continued)

Field	Description
	Indicates that an enabled internal peripheral was a source of exiting a low-leakage power mode. To clear the flag, follow the internal peripheral flag clearing mechanism. 0 Module 0 input was not a wakeup source 1 Module 0 input was a wakeup source

16.3.9 LLWU Pin Filter 1 register (LLWU_FILT1)

LLWU_FILT1 is a control and status register that is used to enable/disable the digital filter 1 features for an external pin.

NOTE

> This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: LLWU_FILT1 is 4007_C000h base + 8h offset = 4007_C008h

Bit	7	6	5	4	3	2	1	0
Read	FILTF	FILTE		0	FILTSEL			
Write	w1c			▨				
Reset	0	0	0	0	0	0	0	0

LLWU_FILT1 field descriptions

Field	Description
7 FILTF	Filter Detect Flag Indicates that the filtered external wakeup pin, selected by FILTSEL, was a source of exiting a low-leakage power mode. To clear the flag write a one to FILTF. 0 Pin Filter 1 was not a wakeup source 1 Pin Filter 1 was a wakeup source
6–5 FILTE	Digital Filter On External Pin Controls the digital filter options for the external pin detect. 00 Filter disabled 01 Filter posedge detect enabled 10 Filter negedge detect enabled 11 Filter any edge detect enabled
4 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

LLWU_FILT1 field descriptions (continued)

Field	Description
3–0 FILTSEL	Filter Pin Select Selects 1 out of the 16 wakeup pins to be muxed into the filter. 0000 Select LLWU_P0 for filter ... ... 1111 Select LLWU_P15 for filter

16.3.10 LLWU Pin Filter 2 register (LLWU_FILT2)

LLWU_FILT2 is a control and status register that is used to enable/disable the digital filter 2 features for an external pin.

NOTE

This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: LLWU_FILT2 is 4007_C000h base + 9h offset = 4007_C009h

Bit	7	6	5	4	3	2	1	0
Read	FILTF	FILTE		0	FILTSEL			
Write	w1c							
Reset	0	0	0	0	0	0	0	0

LLWU_FILT2 field descriptions

Field	Description
7 FILTF	Filter Detect Flag Indicates that the filtered external wakeup pin, selected by FILTSEL, was a source of exiting a low-leakage power mode. To clear the flag write a one to FILTF. 0 Pin Filter 2 was not a wakeup source 1 Pin Filter 2 was a wakeup source
6–5 FILTE	Digital Filter On External Pin Controls the digital filter options for the external pin detect. 00 Filter disabled 01 Filter posedge detect enabled 10 Filter negedge detect enabled 11 Filter any edge detect enabled
4 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

LLWU_FILT2 field descriptions (continued)

Field	Description
3–0 FILTSEL	Filter Pin Select Selects 1 out of the 16 wakeup pins to be muxed into the filter. 0000 Select LLWU_P0 for filter ... ... 1111 Select LLWU_P15 for filter

16.3.11 LLWU Reset Enable register (LLWU_RST)

LLWU_RST is a control register that is used to enable/disable the digital filter for the external pin detect and RESET pin.

NOTE

> This register is reset on Chip Reset not VLLS and by reset types that trigger Chip Reset not VLLS. It is unaffected by reset types that do not trigger Chip Reset not VLLS. See the Introduction details for more information.

Address: LLWU_RST is 4007_C000h base + Ah offset = 4007_C00Ah

Bit	7	6	5	4	3	2	1	0
Read	0						LLRSTE	RSTFILT
Write								
Reset	0	0	0	0	0	0	1	0

LLWU_RST field descriptions

Field	Description
7–2 Reserved	This read-only field is reserved and always has the value zero.
1 LLRSTE	Low-Leakage Mode RESET Enable This bit must be set to allow the device to be reset while in a low-leakage power mode. On devices where Reset is not a dedicated pin, the RESET pin must also be enabled in the explicit port mux control. 0 RESET pin not enabled as a leakage mode exit source 1 RESET pin enabled as a low leakage mode exit source
0 RSTFILT	Digital Filter On RESET Pin Enables the digital filter for the RESET pin during LLS, VLLS3, VLLS2, or VLLS1 modes. 0 Filter not enabled 1 Filter enabled

16.4 Functional description

This on-chip peripheral module is called a low-leakage wakeup unit (LLWU) module because it allows internal peripherals and external input pins as a source of wakeup from low-leakage modes. It is operational only in LLS and VLLSx modes.

The LLWU module contains pin enables for each external pin and internal module. For each external pin, the user can disable or select the edge type for the wakeup. Type options are:

- Falling-edge
- Rising-edge
- Either-edge

When an external pin is enabled as a wakeup source, the pin must be configured as an input pin.

The LLWU implements optional 3-cycle glitch filters, based on the LPO clock. A detected external pin, either wakeup or $\overline{\text{RESET}}$, is required to remain asserted until the enabled glitch filter times out. Additional latency of up to 2 cycles is due to synchronization, which results in a total of up to 5 cycles of delay before the detect circuit alerts the system to the wakeup or reset event when the filter function is enabled. Two wakeup detect filters are available to detect up to two external pins. A separate reset filter is on the $\overline{\text{RESET}}$ pin. Glitch filtering is not provided on the internal modules.

For internal module wakeup operation, the WUMEx bit enables the associated module as a wakeup source.

16.4.1 LLS mode

Wakeup events triggered from either an external pin input or an internal module input result in a CPU interrupt flow to begin user code execution.

An LLS reset event due to RESET pin assertion causes an exit via a system reset. State retention data is lost, and the I/O states return to their reset state. The RCM_SRS[WAKEUP] and RCM_SRS[PIN] bits are set and the system executes a reset flow before CPU operation begins with a reset vector fetch.

16.4.2 VLLS modes

In the case of a wakeup due to external pin or internal module wakeup, recovery is always via a reset flow and the RCM_SRS[WAKEUP] is set indicating the low-leakage mode was active. State retention data is lost and I/O will be restored after PMC_REGSC[ACKISO] has been written.

A VLLS exit event due to $\overline{\text{RESET}}$ pin assertion causes an exit via a system reset. State retention data is lost and the I/O states immediately return to their reset state. The RCM_SRS[WAKEUP] and RCM_SRS[PIN] bits are set and the system executes a reset flow before CPU operation begins with a reset vector fetch.

16.4.3 Initialization

For an enabled peripheral wakeup input, the peripheral flag must be cleared by software before entering LLS or VLLSx mode to avoid an immediate exit from the mode.

Flags associated with external input pins, filtered and unfiltered, must also be cleared by software prior to entry to LLS or VLLSx mode.

After enabling an external pin filter or changing the source pin, wait at least 5 LPO clock cycles before entering LLS or VLLSx mode to allow the filter to initialize.

NOTE

The signal selected as a wakeup source pin must be a digital pin, as selected in the pin mux control.

Chapter 17
Miscellaneous Control Module (MCM)

17.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The Miscellaneous Control Module (MCM) provides a myriad of miscellaneous control functions.

17.1.1 Features

The MCM includes the following features:

- Program-visible information on the platform configuration and revision

17.2 Memory Map/Register Descriptions

The memory map and register descriptions below describe the registers using byte addresses.

MCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
E008_0008	Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)	16	R	000Fh	17.2.1/312

Table continues on the next page...

MCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
E008_000A	Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)	16	R	000Fh	17.2.2/312
E008_000C	Crossbar Switch (AXBS) Control Register (MCM_PLACR)	32	R/W	0000_0000h	17.2.3/313

17.2.1 Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)

PLASC is a 16-bit read-only register identifying the presence/absence of bus slave connections to the device’s crossbar switch.

Address: MCM_PLASC is E008_0000h base + 8h offset = E008_0008h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								ASC							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

MCM_PLASC field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value zero.
7–0 ASC	Each bit in the ASC field indicates whether there is a corresponding connection to the crossbar switch's slave input port. 0 A bus slave connection to AXBS input port *n* is absent 1 A bus slave connection to AXBS input port *n* is present

17.2.2 Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)

PLAMC is a 16-bit read-only register identifying the presence/absence of bus master connections to the device's crossbar switch.

Address: MCM_PLAMC is E008_0000h base + Ah offset = E008_000Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								AMC							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

MCM_PLAMC field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value zero.
7–0 AMC	Each bit in the AMC field indicates whether there is a corresponding connection to the AXBS master input port. 0 A bus master connection to AXBS input port *n* is absent 1 A bus master connection to AXBS input port *n* is present

17.2.3 Crossbar Switch (AXBS) Control Register (MCM_PLACR)

The PLACR register selects the arbitration policy for the crossbar masters.

Address: MCM_PLACR is E008_0000h base + Ch offset = E008_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																						ARB	0								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCM_PLACR field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero.
9 ARB	Arbitration select 0 Fixed-priority arbitration for the crossbar masters 1 Round-robin arbitration for the crossbar masters
8–0 Reserved	This read-only field is reserved and always has the value zero.

Chapter 18
Crossbar Switch Lite (AXBS-Lite)

18.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration chapter.

This chapter provides information on the layout, configuration, and programming of the crossbar switch. The crossbar switch connects bus masters and bus slaves using a crossbar switch structure. This structure allows up to four bus masters to access different bus slaves simultaneously, while providing arbitration among the bus masters when they access the same slave.

18.1.1 Features

The crossbar switch includes these distinctive features:

- Symmetric crossbar bus switch implementation
    - Allows concurrent accesses from different masters to different slaves
    - Slave arbitration attributes configured on a slave-by-slave basis
- 32-bit width and support for byte, 2-byte, 4-byte, and 16-byte burst transfers
- Operation at a 1-to-1 clock frequency with the bus masters
- Low-Power Park mode support

18.2 Memory Map / Register Definition

This design was meant to be as small as possible. To help achieve this, the crossbar switch contains no memory-mapped registers for configuring.

18.3 Functional Description

18.3.1 General operation

When a master accesses the crossbar switch the access is immediately taken. If the targeted slave port of the access is available, then the access is immediately presented on the slave port. Single-clock, or -zero-wait state, accesses are possible through the crossbar. If the targeted slave port of the access is busy or parked on a different master port, the requesting master simply sees wait states inserted until the targeted slave port can service the master's request. The latency in servicing the request depends on each master's priority level and the responding peripheral's access time.

Because the crossbar switch appears to be just another slave to the master device, the master device has no knowledge of whether it actually owns the slave port it is targeting. While the master does not have control of the slave port it is targeting, it simply waits.

A master is given control of the targeted slave port only after a previous access to a different slave port completes, regardless of its priority on the newly targeted slave port. This prevents deadlock from occurring when:

- A higher priority master has:
  - An outstanding request to one slave port that has a long response time and
  - A pending access to a different slave port, and
- A lower priority master is also making a request to the same slave port as the pending access of the higher priority master.

After the master has control of the slave port it is targeting, the master remains in control of that slave port until it gives up the slave port by running an IDLE cycle or by leaving that slave port for its next access.

The master could also lose control of the slave port if another higher priority master makes a request to the slave port; however, if the master is running a fixed- or undefined-length burst transfer it retains control of the slave port until that transfer completes. Based on MGPCR[AULB], the master either retains control of the slave port when doing undefined length incrementing burst transfers or loses the bus to a higher priority master.

The crossbar terminates all master IDLE transfers, as opposed to allowing the termination to come from one of the slave buses. Additionally, when no master is requesting access to a slave port, the crossbar drives IDLE transfers onto the slave bus, even though a default master may be granted access to the slave port.

When a slave bus is being idled by the crossbar, it remains parked with the last master to use the slave port . This is done to save the initial clock of arbitration delay that otherwise would be seen if the master had to arbitrate to gain control of the slave port.

18.3.2 Arbitration

The crossbar switch supports two arbitration schemes:

- A fixed-priority comparison algorithm
- A round-robin fairness algorithm

18.3.2.1 Fixed-priority operation

When operating in Fixed-Priority mode, each master is assigned a unique priority level with the highest numbered master having the highest priority (master 1 has lower priority than master 3). If two masters request access to a slave port, the master with the highest priority gains control over the slave port.

When a master makes a request to a slave port, the slave port checks whether the new requesting master's priority level is higher than that of the master that currently has control over the slave port, unless the slave port is in a parked state. The slave port performs an arbitration check at every clock edge to ensure that the proper master, if any, has control of the slave port.

The following table describes possible scenarios based on the requesting master port:

Table 18-1. How AXBS grants control of a slave port to a master

When	Then AXBS grants control to the requesting master
Both of the following are true: • The current master is not running a transfer. • The new requesting master's priority level is higher than that of the current master.	At the next clock edge
Both of the following are true: • The current master is running a fixed length burst transfer or a locked transfer. • The requesting master's priority level is higher than that of the current master.	At the end of the burst transfer or locked transfer

Table continues on the next page...

Table 18-1. How AXBS grants control of a slave port to a master (continued)

When	Then AXBS grants control to the requesting master
Both of the following are true: • The current master is running an undefined length burst transfer. • The requesting master's priority level is higher than that of the current master.	At the next arbitration point for the undefined length burst transfer **NOTE:** Arbitration points for an undefined length burst are defined in the MGPCR for each master.
The requesting master's priority level is lower than the current master.	At the conclusion of one of the following cycles: • An IDLE cycle • A non-IDLE cycle to a location other than the current slave port

18.3.2.2 Round-robin priority operation

When operating in Round-Robin mode, each master is assigned a relative priority based on the master port number. This relative priority is compared to the master port number (ID) of the last master to perform a transfer on the slave bus. The highest priority requesting master becomes owner of the slave bus at the next transfer boundary, accounting for locked and fixed-length burst transfers. Priority is based on how far ahead the ID of the requesting master is to the ID of the last master.

After granted access to a slave port, a master may perform as many transfers as desired to that port until another master makes a request to the same slave port. The next master in line is granted access to the slave port at the next transfer boundary, or possibly on the next clock cycle if the current master has no pending access request.

As an example of arbitration in Round-Robin mode, assume the crossbar is implemented with master ports 0, 1, 4, and 5. If the last master of the slave port was master 1, and master 0, 4 and 5 make simultaneous requests, they are serviced in the order 4, 5, and then 0.

Parking may continue to be used in a round-robin mode, but does not affect the round-robin pointer unless the parked master actually performs a transfer. Handoff occurs to the next master in line after one cycle of arbitration.

18.4 Initialization/application information

No initialization is required by or for the crossbar switch.

Chapter 19
Peripheral Bridge (AIPS-Lite)

19.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The peripheral bridge (AIPS-Lite) converts the crossbar switch interface to an interface to access a majority of peripherals on the device.

The peripheral bridge supports up to 128 peripherals. The bridge includes separate clock enable inputs for each of the slots to accommodate slower peripherals.

19.1.1 Features

Key features of the peripheral bridge are:

- Supports up to 128 peripherals and two global external peripheral spaces
- Supports 8-, 16-, and 32-bit width peripheral slots
- Allows each independently configurable peripheral to include a clock enable, which allows peripherals to operate at any speed less than the system clock rate.
- Provides programming model provides memory protection functionality

19.1.2 General operation

The peripherals connected to the peripheral bridge are modules that contain readable/writable control and status registers. The system masters read and write these registers through the peripheral bridge. The peripheral bridge generates the following as inputs to the peripherals:

- Module enables
- The module address
- Transfer attributes
- Byte enables
- Write data

The peripheral bridge captures read data from the peripheral interface and drives it to the crossbar switch.

Each peripheral is allocated one block of the memory map. Two global external module enables are available for the remaining address space to allow for customization and expansion of addressed peripheral devices.

19.2 Functional description

The peripheral bridge serves as an interface between the crossbar switch and the slave peripheral bus. It functions as a protocol translator.

The peripheral bridge generates select signals for modules on the peripheral bus by decoding accesses within the peripheral bridge address space.

19.2.1 Access support

Aligned and misaligned 32-bit and 16-bit accesses, as well as byte accesses are supported for 32-bit peripherals. Misaligned accesses are supported to allow memory to be placed on the slave peripheral bus. Peripheral registers must not be misaligned, although no explicit checking is performed by the peripheral bridge. All accesses are performed with a single transfer.

All accesses to the peripheral slots must be sized less than or equal to the designated peripheral slot size. If an access is attempted which is larger than the targeted port, an error response is generated.

Chapter 20
Direct Memory Access Multiplexer (DMAMUX)

20.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration chapter.

20.1.1 Overview

The direct memory access multiplexer (DMAMUX) routes DMA sources, called slots, to any of the DMA channels. This process is illustrated in the following figure.

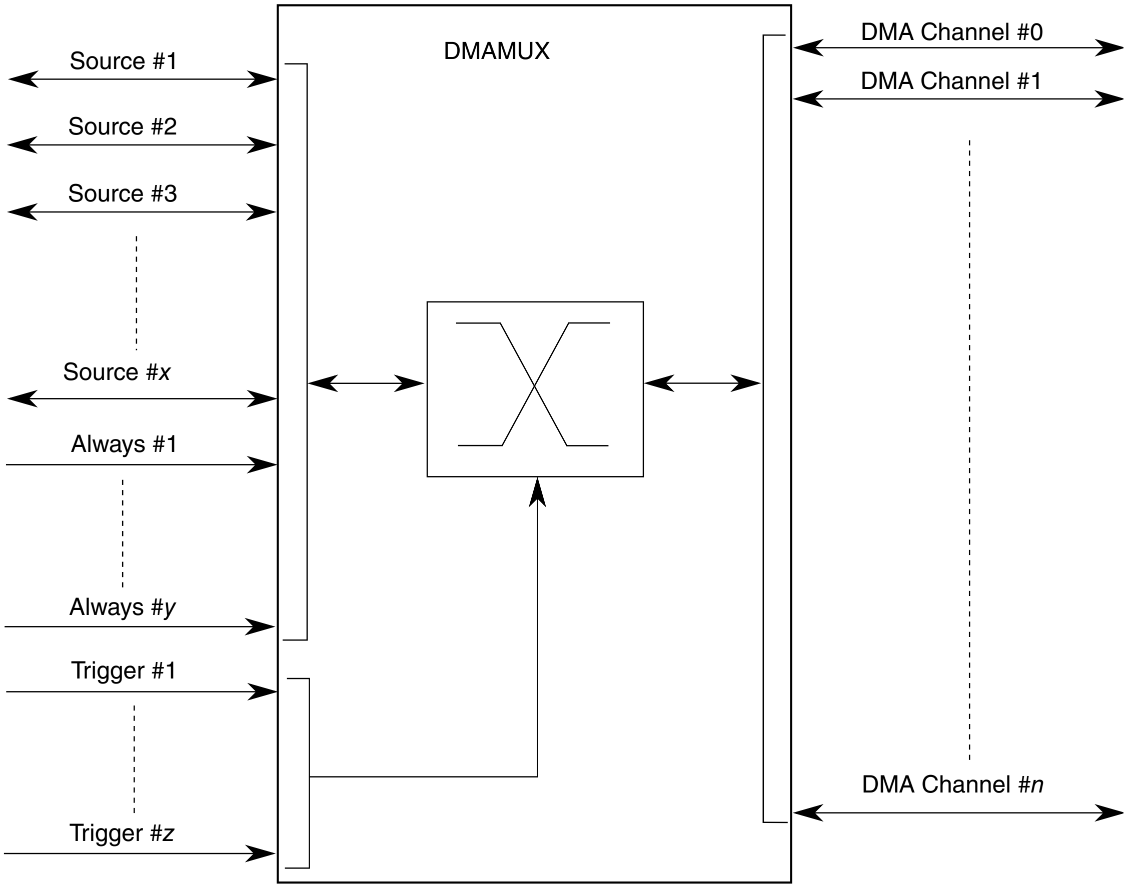

DMAMUX
Source #1
Source #2
Source #3
Source #*x*
Always #1
Always #*y*
Trigger #1
Trigger #*z*
DMA Channel #0
DMA Channel #1
DMA Channel #*n*

Figure 20-1. DMAMUX block diagram

20.1.2 Features

The DMA channel MUX provides these features:

- 52 peripheral slots and 10 always-on slots can be routed to channels.
- independently selectable DMA channel routers.
    - The first 4 channels additionally provide a trigger functionality.
- Each channel router can be assigned to one of the 52 possible peripheral DMA slots or to one of the 10 always-on slots.

20.1.3 Modes of operation

The following operating modes are available:

- Disabled mode

In this mode, the DMA channel is disabled. Because disabling and enabling of DMA channels is done primarily via the DMA configuration registers, this mode is used mainly as the reset state for a DMA channel in the DMA channel MUX. It may also be used to temporarily suspend a DMA channel while reconfiguration of the system takes place, for example, changing the period of a DMA trigger.

- Normal mode

  In this mode, a DMA source is routed directly to the specified DMA channel. The operation of the DMA MUX in this mode is completely transparent to the system.

- Periodic Trigger mode

  In this mode, a DMA source may only request a DMA transfer, such as when a transmit buffer becomes empty or a receive buffer becomes full, periodically. Configuration of the period is done in the registers of the periodic interrupt timer (PIT). This mode is available only for channels 0-3.

20.2 External signal description

The DMA MUX has no external pins.

20.3 Memory map/register definition

This section provides a detailed description of all memory-mapped registers in the DMA MUX.

The following table shows the memory map for the DMA MUX.

All registers are accessible via 8-bit, 16-bit, or 32-bit accesses.

DMAMUX memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_1000	Channel Configuration register (DMAMUX0_CHCFG0)	8	R/W	00h	20.3.1/324
4002_1001	Channel Configuration register (DMAMUX0_CHCFG1)	8	R/W	00h	20.3.1/324
4002_1002	Channel Configuration register (DMAMUX0_CHCFG2)	8	R/W	00h	20.3.1/324
4002_1003	Channel Configuration register (DMAMUX0_CHCFG3)	8	R/W	00h	20.3.1/324

Table continues on the next page...

DMAMUX memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_1004	Channel Configuration register (DMAMUX0_CHCFG4)	8	R/W	00h	20.3.1/324
4002_1005	Channel Configuration register (DMAMUX0_CHCFG5)	8	R/W	00h	20.3.1/324
4002_1006	Channel Configuration register (DMAMUX0_CHCFG6)	8	R/W	00h	20.3.1/324
4002_1007	Channel Configuration register (DMAMUX0_CHCFG7)	8	R/W	00h	20.3.1/324
4002_1008	Channel Configuration register (DMAMUX0_CHCFG8)	8	R/W	00h	20.3.1/324
4002_1009	Channel Configuration register (DMAMUX0_CHCFG9)	8	R/W	00h	20.3.1/324
4002_100A	Channel Configuration register (DMAMUX0_CHCFG10)	8	R/W	00h	20.3.1/324
4002_100B	Channel Configuration register (DMAMUX0_CHCFG11)	8	R/W	00h	20.3.1/324
4002_100C	Channel Configuration register (DMAMUX0_CHCFG12)	8	R/W	00h	20.3.1/324
4002_100D	Channel Configuration register (DMAMUX0_CHCFG13)	8	R/W	00h	20.3.1/324
4002_100E	Channel Configuration register (DMAMUX0_CHCFG14)	8	R/W	00h	20.3.1/324
4002_100F	Channel Configuration register (DMAMUX0_CHCFG15)	8	R/W	00h	20.3.1/324

20.3.1 Channel Configuration register (DMAMUXx_CHCFG*n*)

Each of the DMA channels can be independently enabled/disabled and associated with one of the DMA slots (peripheral slots or always-on slots) in the system.

NOTE

Setting multiple CHCFG registers with the same Source value will result in unpredictable behavior.

NOTE

Before changing the trigger or source settings a DMA channel must be disabled via the CHCFGn[ENBL] bit.

Addresses: 4002_1000h base + 0h offset + (1d × *n*), where *n* = 0d to 15d

Bit	7	6	5	4	3	2	1	0
Read Write	ENBL	TRIG	SOURCE					
Reset	0	0	0	0	0	0	0	0

DMAMUXx_CHCFG*n* field descriptions

Field	Description
7 ENBL	DMA Channel Enable Enables the DMA channel.

Table continues on the next page...

DMAMUXx_CHCFG*n* field descriptions (continued)

Field	Description
	0 DMA channel is disabled. This mode is primarily used during configuration of the DMA Mux. The DMA has separate channel enables/disables, which should be used to disable or re-configure a DMA channel. 1 DMA channel is enabled
6 TRIG	DMA Channel Trigger Enable Enables the periodic trigger capability for the triggered DMA channel. 0 Triggering is disabled. If triggering is disabled, and the ENBL bit is set, the DMA Channel will simply route the specified source to the DMA channel. (Normal mode) 1 Triggering is enabled. If triggering is enabled, and the ENBL bit is set, the DMAMUX is in Periodic Trigger mode.
5–0 SOURCE	DMA Channel Source (Slot) Specifies which DMA source, if any, is routed to a particular DMA channel. See your device's chip configuration details for further details about the peripherals and their slot numbers.

20.4 Functional description

The primary purpose of the DMA MUX is to provide flexibility in the system's use of the available DMA channels. As such, configuration of the DMA MUX is intended to be a static procedure done during execution of the system boot code. However, if the procedure outlined in Enabling and configuring sources is followed, the configuration of the DMA MUX may be changed during the normal operation of the system.

Functionally, the DMA MUX channels may be divided into two classes:

- Channels which implement the normal routing functionality plus periodic triggering capability
- Channels which implement only the normal routing functionality

20.4.1 DMA channels with periodic triggering capability

Besides the normal routing functionality, the first four channels of the DMA MUX provide a special periodic triggering capability that can be used to provide an automatic mechanism to transmit bytes, frames, or packets at fixed intervals without the need for processor intervention. The trigger is generated by the periodic interrupt timer (PIT); as such, the configuration of the periodic triggering interval is done via configuration registers in the PIT. See the section on periodic interrupt timer for more information on this topic.

Note

Because of the dynamic nature of the system (i.e. DMA channel priorities, bus arbitration, interrupt service routine lengths, etc.), the number of clock cycles between a trigger and the actual DMA transfer cannot be guaranteed.

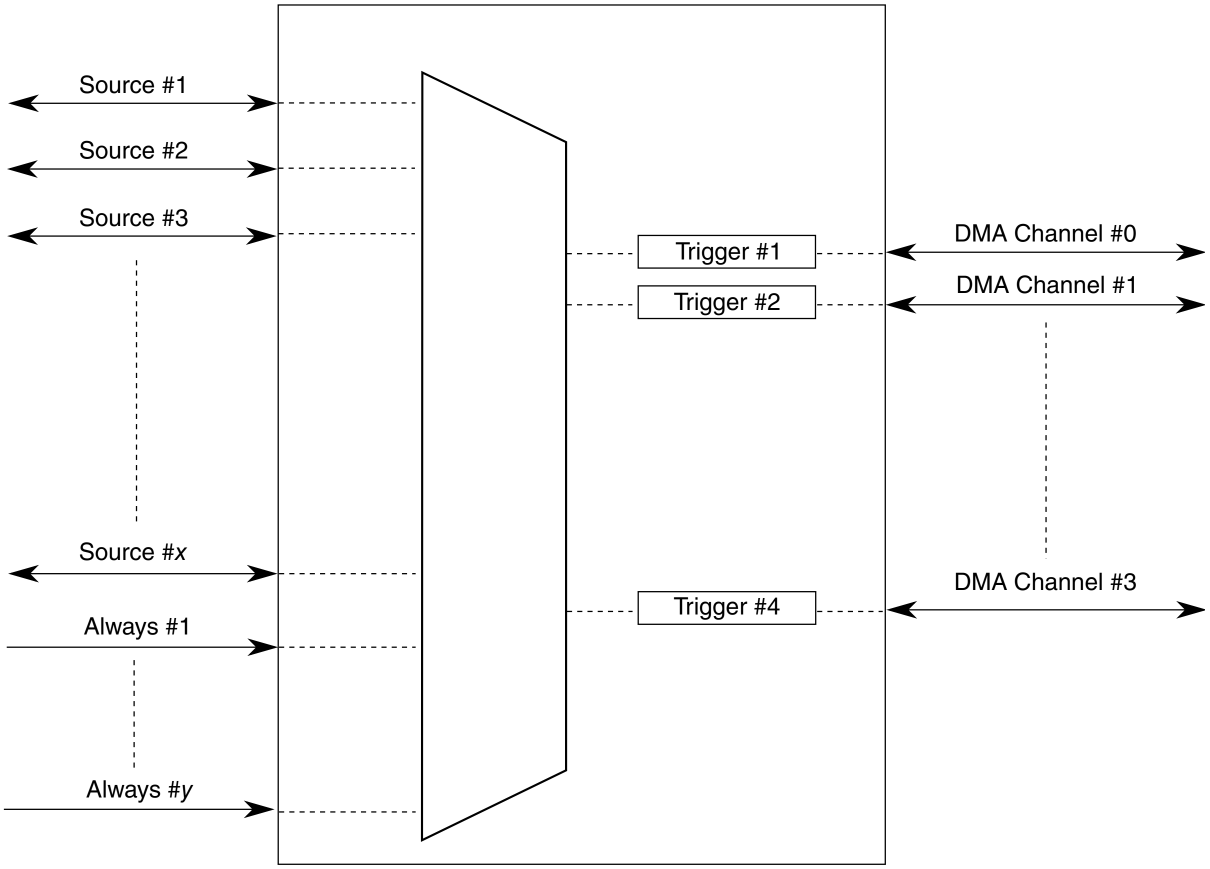

Source #1
Source #2
Source #3
Source #*x*
Always #1
Always #*y*
Trigger #1
Trigger #2
Trigger #4
DMA Channel #0
DMA Channel #1
DMA Channel #3

Figure 20-36. DMA MUX triggered channels

The DMA channel triggering capability allows the system to "schedule" regular DMA transfers, usually on the transmit side of certain peripherals, without the intervention of the processor. This trigger works by gating the request from the peripheral to the DMA until a trigger event has been seen. This is illustrated in the following figure.

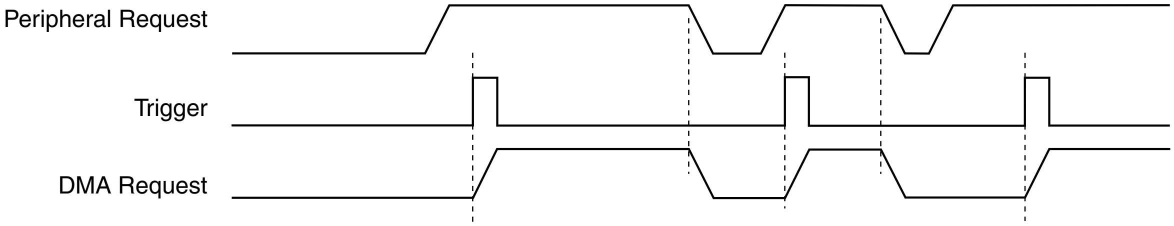

Peripheral Request
Trigger
DMA Request

Figure 20-37. DMA MUX channel triggering: normal operation

After the DMA request has been serviced, the peripheral will negate its request, effectively resetting the gating mechanism until the peripheral re-asserts its request AND the next trigger event is seen. This means that if a trigger is seen, but the peripheral is not requesting a transfer, then that trigger will be ignored. This situation is illustrated in the following figure.

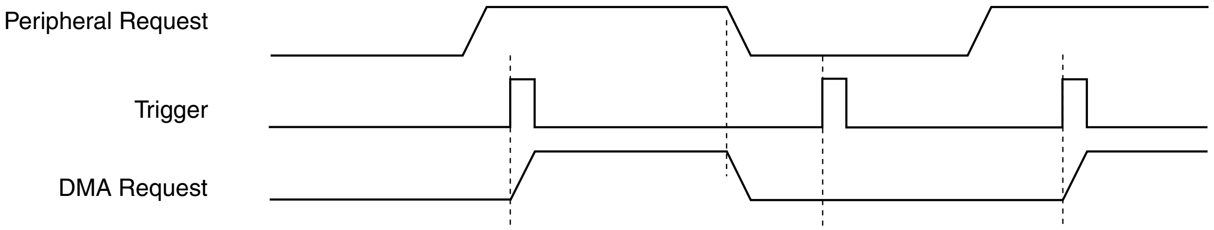


Figure 20-38. DMA MUX channel triggering: ignored trigger

This triggering capability may be used with any peripheral that supports DMA transfers, and is most useful for two types of situations:

- Periodically polling external devices on a particular bus. As an example, the transmit side of an SPI is assigned to a DMA channel with a trigger, as described above. After it has been setup, the SPI will request DMA transfers, presumably from memory, as long as its transmit buffer is empty. By using a trigger on this channel, the SPI transfers can be automatically performed every 5μs (as an example). On the receive side of the SPI, the SPI and DMA can be configured to transfer receive data into memory, effectively implementing a method to periodically read data from external devices and transfer the results into memory without processor intervention.
- Using the GPIO ports to drive or sample waveforms. By configuring the DMA to transfer data to one or more GPIO ports, it is possible to create complex waveforms using tabular data stored in on-chip memory. Conversely, using the DMA to periodically transfer data from one or more GPIO ports, it is possible to sample complex waveforms and store the results in tabular form in on-chip memory.

A more detailed description of the capability of each trigger, including resolution, range of values, and so on, may be found in the periodic interrupt timer section.

20.4.2 DMA channels with no triggering capability

The other channels of the DMA MUX provide the normal routing functionality as described in Modes of operation.

20.4.3 "Always enabled" DMA sources

In addition to the peripherals that can be used as DMA sources, there are 10 additional DMA sources that are "always enabled". Unlike the peripheral DMA sources, where the peripheral controls the flow of data during DMA transfers, the "always enabled" sources provide no such "throttling" of the data transfers. These sources are most useful in the following cases:

- Doing DMA transfers to/from GPIO—Moving data from/to one or more GPIO pins, either unthrottled (that is as fast as possible), or periodically (using the DMA triggering capability).
- Doing DMA transfers from memory to memory—Moving data from memory to memory, typically as fast as possible, sometimes with software activation.
- Doing DMA transfers from memory to the external bus, or vice-versa—Similar to memory to memory transfers, this is typically done as quickly as possible.
- Any DMA transfer that requires software activation—Any DMA transfer that should be explicitly started by software.

In cases where software should initiate the start of a DMA transfer, an "always enabled" DMA source can be used to provide maximum flexibility. When activating a DMA channel via software, subsequent executions of the minor loop require a new "start" event be sent. This can either be a new software activation, or a transfer request from the DMA channel MUX. The options for doing this are:

- Transfer all data in a single minor loop. By configuring the DMA to transfer all of the data in a single minor loop (that is major loop counter = 1), no reactivation of the channel is necessary. The disadvantage to this option is the reduced granularity in determining the load that the DMA transfer will incur on the system. For this option, the DMA channel must be disabled in the DMA channel MUX.
- Use explicit software reactivation. In this option, the DMA is configured to transfer the data using both minor and major loops, but the processor is required to reactivate the channel by writing to the DMA registers *after every minor loop*. For this option, the DMA channel must be disabled in the DMA channel MUX.
- Use an "always enabled" DMA source. In this option, the DMA is configured to transfer the data using both minor and major loops, and the DMA channel MUX does the channel re-activation. For this option, the DMA channel should be enabled and pointing to an "always enabled" source. Note that the reactivation of the channel can

be continuous (DMA triggering is disabled) or can use the DMA triggering capability. In this manner, it is possible to execute periodic transfers of packets of data from one source to another, without processor intervention.

20.5 Initialization/application information

This section provides instructions for initializing the DMA channel MUX.

20.5.1 Reset

The reset state of each individual bit is shown in Memory map/register definition. In summary, after reset, all channels are disabled and must be explicitly enabled before use.

20.5.2 Enabling and configuring sources

To enable a source with periodic triggering:

1. Determine with which DMA channel the source will be associated. Note that only the first 4 DMA channels have periodic triggering capability.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] bits of the DMA channel.
3. Ensure that the DMA channel is properly configured in the DMA. The DMA channel may be enabled at this point.
4. Configure the corresponding timer.
5. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] and CHCFG[TRIG] bits are set.

To configure source #5 transmit for use with DMA channel 2, with periodic triggering capability:

1. Write 0x00 to CHCFG2 (base address + 0x02).
2. Configure channel 2 in the DMA, including enabling the channel.
3. Configure a timer for the desired trigger interval.
4. Write 0xC5 to CHCFG2 (base address + 0x02).

The following code example illustrates steps 1 and 4 above:

```
In File registers.h:
#define DMAMUX_BASE_ADDR     0xFC084000/* Example only ! */
/* Following example assumes char is 8-bits */
volatile unsigned char *CHCONFIG0 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCONFIG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCONFIG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCONFIG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0003);
volatile unsigned char *CHCONFIG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCONFIG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCONFIG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCONFIG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCONFIG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCONFIG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCONFIG10= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCONFIG11= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCONFIG12= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned char *CHCONFIG13= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
volatile unsigned char *CHCONFIG14= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000E);
volatile unsigned char *CHCONFIG15= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000F);

In File main.c:
#include "registers.h"
:
:
*CHCONFIG2 = 0x00;
*CHCONFIG2 = 0xC5;
```

To enable a source without periodic triggering:

1. Determine with which DMA channel the source will be associated. Note that only the first 4 DMA channels have periodic triggering capability.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] bits of the DMA channel.
3. Ensure that the DMA channel is properly configured in the DMA. The DMA channel may be enabled at this point.
4. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] is set while the CHCFG[TRIG] bit is cleared.

To configure source #5 Transmit for use with DMA channel 2, with no periodic triggering capability:

1. Write 0x00 to CHCFG2 (base address + 0x02).
2. Configure channel 2 in the DMA, including enabling the channel.
3. Write 0x85 to CHCFG2 (base address + 0x02).

The following code example illustrates steps 1 and 3 above:

```
In File registers.h:
#define DMAMUX_BASE_ADDR     0xFC084000/* Example only ! */
/* Following example assumes char is 8-bits */
volatile unsigned char *CHCONFIG0 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCONFIG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCONFIG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCONFIG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0003);
volatile unsigned char *CHCONFIG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCONFIG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCONFIG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCONFIG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCONFIG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCONFIG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCONFIG10= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCONFIG11= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCONFIG12= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned char *CHCONFIG13= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
volatile unsigned char *CHCONFIG14= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000E);
volatile unsigned char *CHCONFIG15= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000F);

In File main.c:
```

```
#include "registers.h"
:
:
*CHCONFIG2 = 0x00;
*CHCONFIG2 = 0x85;
```

Disabling a source

A particular DMA source may be disabled by not writing the corresponding source value into any of the CHCFG registers. Additionally, some module-specific configuration may be necessary. See the appropriate section for more details.

To switch the source of a DMA channel:

1. Disable the DMA channel in the DMA and re-configure the channel for the new source.
2. Clear the CHCFG[ENBL] and CHCFG[TRIG] bits of the DMA channel.
3. Select the source to be routed to the DMA channel. Write to the corresponding CHCFG register, ensuring that the CHCFG[ENBL] and CHCFG[TRIG] bits are set.

To switch DMA channel 8 from source #5 transmit to source #7 transmit:

1. In the DMA configuration registers, disable DMA channel 8 and re-configure it to handle the transfers to peripheral slot 7. This example assumes channel 8 doesn't have triggering capability.
2. Write 0x00 to CHCFG8 (base address + 0x08).
3. Write 0x87 to CHCFG8 (base address + 0x08). (In this example, setting the CHCFG[TRIG] bit would have no effect, due to the assumption that channels 8 does not support the periodic triggering functionality).

The following code example illustrates steps 2 and 3 above:

```
In File registers.h:
#define DMAMUX_BASE_ADDR     0xFC084000/* Example only ! */
/* Following example assumes char is 8-bits */
volatile unsigned char *CHCONFIG0 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0000);
volatile unsigned char *CHCONFIG1 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0001);
volatile unsigned char *CHCONFIG2 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0002);
volatile unsigned char *CHCONFIG3 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0003);
volatile unsigned char *CHCONFIG4 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0004);
volatile unsigned char *CHCONFIG5 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0005);
volatile unsigned char *CHCONFIG6 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0006);
volatile unsigned char *CHCONFIG7 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0007);
volatile unsigned char *CHCONFIG8 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0008);
volatile unsigned char *CHCONFIG9 = (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x0009);
volatile unsigned char *CHCONFIG10= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000A);
volatile unsigned char *CHCONFIG11= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000B);
volatile unsigned char *CHCONFIG12= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000C);
volatile unsigned char *CHCONFIG13= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000D);
volatile unsigned char *CHCONFIG14= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000E);
volatile unsigned char *CHCONFIG15= (volatile unsigned char *) (DMAMUX_BASE_ADDR+0x000F);

In File main.c:
#include "registers.h"
:
:
*CHCONFIG8 = 0x00;
*CHCONFIG8 = 0x87;
```

Chapter 21
Direct Memory Access Controller (eDMA)

21.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The enhanced direct memory access (eDMA) controller is a second-generation module capable of performing complex data transfers with minimal intervention from a host processor. The hardware microarchitecture includes:

- A DMA engine that performs:
  - Source- and destination-address calculations
  - Data-movement operations
- Local memory containing transfer control descriptors for each of the 4 channels

21.1.1 Block diagram

This diagram illustrates the eDMA module.

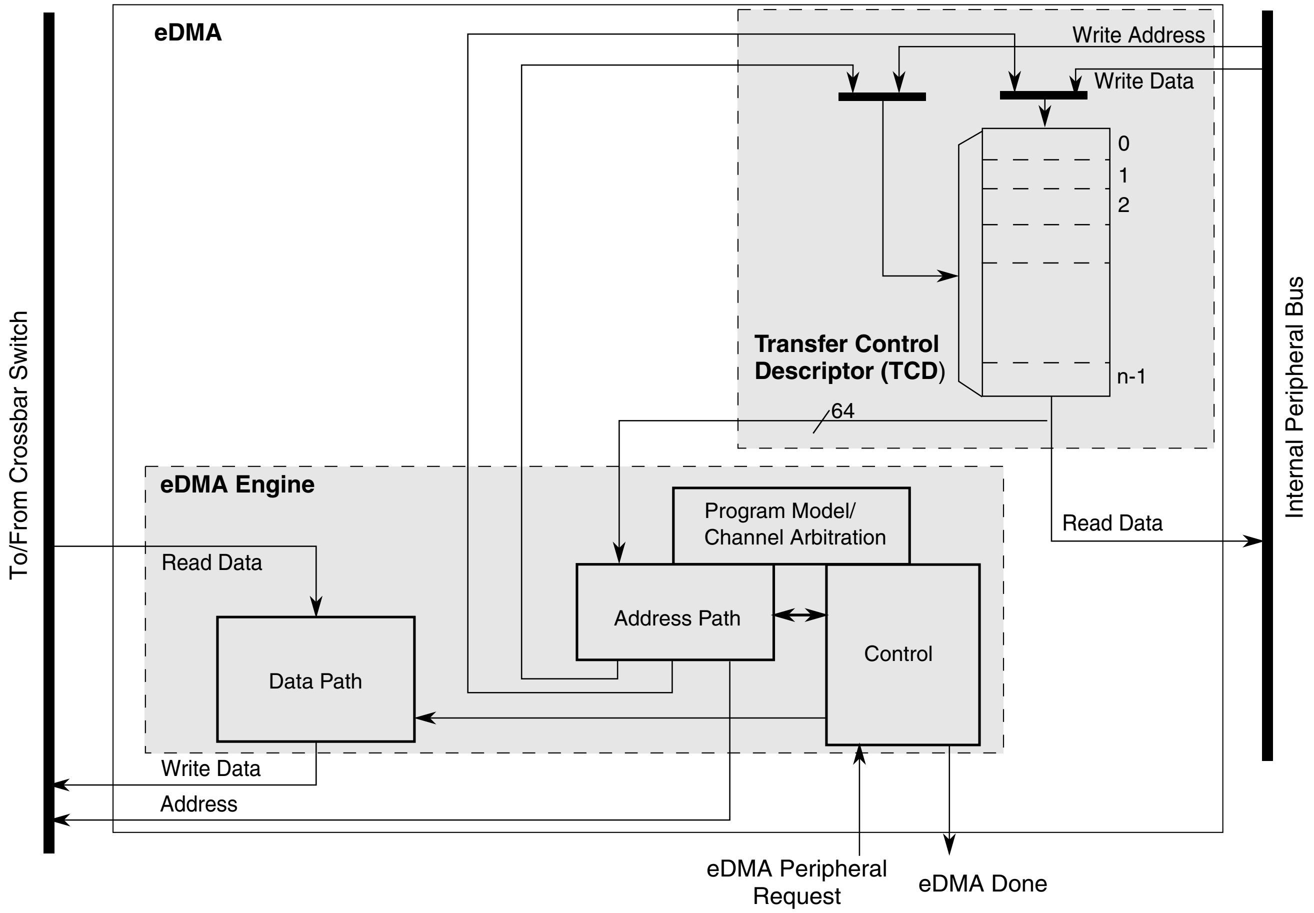

eDMA
Write Address
Write Data
0
1
2
n-1
Transfer Control Descriptor (TCD)
64
Internal Peripheral Bus
To/From Crossbar Switch
eDMA Engine
Program Model/ Channel Arbitration
Read Data
Read Data
Address Path
Control
Data Path
Write Data
Address
eDMA Peripheral Request
eDMA Done

Figure 21-1. eDMA block diagram

21.1.2 Block parts

The eDMA module is partitioned into two major modules: the eDMA engine and the transfer-control descriptor local memory.

The eDMA engine is further partitioned into four submodules:

Table 21-1. eDMA engine submodules

Submodule	Function
Address path	This block implements registered versions of two channel transfer control descriptors, channel x and channel y, and manages all master bus-address calculations. All the channels provide the same functionality. This structure allows data transfers associated with one channel to be preempted after the completion of a read/write sequence if a higher priority channel activation is asserted while the first channel is active. After a channel is activated, it runs until the minor loop is completed, unless preempted by a higher priority channel. This provides a mechanism (enabled by DCHPRI*n*[ECP]) where a large data move operation can be preempted to minimize the time another channel is blocked from execution. When any channel is selected to execute, the contents of its TCD are read from local memory and loaded into the address path channel x registers for a normal start and into channel y registers for a preemption start. After the minor loop completes execution, the address path hardware writes the new values for the TCD*n*_{SADDR, DADDR, CITER} back to local memory. If the major iteration count is exhausted, additional processing is performed, including the final address pointer updates, reloading the TCD*n*_CITER field, and a possible fetch of the next TCD*n* from memory as part of a scatter/gather operation.
Data path	This block implements the bus master read/write datapath. It includes 16 bytes of register storage and the necessary multiplex logic to support any required data alignment. The internal read data bus is the primary input, and the internal write data bus is the primary output. The address and data path modules directly support the 2-stage pipelined internal bus. The address path module represents the 1st stage of the bus pipeline (address phase), while the data path module implements the 2nd stage of the pipeline (data phase).
Program model/channel arbitration	This block implements the first section of the eDMA programming model as well as the channel arbitration logic. The programming model registers are connected to the internal peripheral bus. The eDMA peripheral request inputs and interrupt request outputs are also connected to this block (via control logic).
Control	This block provides all the control functions for the eDMA engine. For data transfers where the source and destination sizes are equal, the eDMA engine performs a series of source read/destination write operations until the number of bytes specified in the minor loop byte count has moved. For descriptors where the sizes are not equal, multiple accesses of the smaller size data are required for each reference of the larger size. As an example, if the source size references 16-bit data and the destination is 32-bit data, two reads are performed, then one 32-bit write.

The transfer-control descriptor local memory is further partitioned into:

Table 21-2. Transfer control descriptor memory

Submodule	Description
Memory controller	This logic implements the required dual-ported controller, managing accesses from the eDMA engine as well as references from the internal peripheral bus. As noted earlier, in the event of simultaneous accesses, the eDMA engine is given priority and the peripheral transaction is stalled.
Memory array	TCD storage is implemented using a single-port, synchronous RAM array.

21.1.3 Features

The eDMA is a highly-programmable data-transfer engine optimized to minimize the required intervention from the host processor. It is intended for use in applications where the data size to be transferred is statically known and not defined within the data packet itself. The eDMA module features:

- All data movement via dual-address transfers: read from source, write to destination
    - Programmable source and destination addresses and transfer size
    - Support for enhanced addressing modes
- 4-channel implementation that performs complex data transfers with minimal intervention from a host processor
    -
    - Connections to the crossbar switch for bus mastering the data movement
- Transfer control descriptor (TCD) organized to support two-deep, nested transfer operations
    - 32-byte TCD stored in local memory for each channel
    - An inner data transfer loop defined by a minor byte transfer count
    - An outer data transfer loop defined by a major iteration count
- Channel activation via one of three methods:
    - Explicit software initiation
    - Initiation via a channel-to-channel linking mechanism for continuous transfers
    - Peripheral-paced hardware requests, one per channel
- Fixed-priority and round-robin channel arbitration
- Channel completion reported via optional interrupt requests

  - One interrupt per channel, optionally asserted at completion of major iteration count
  - Optional error terminations per channel and logically summed together to form one error interrupt to the interrupt controller
- Optional support for scatter/gather DMA processing
- Support for complex data structures
- Support to cancel transfers via software

In the discussion of this module, *n* is used to reference the channel number.

21.2 Modes of operation

The eDMA operates in the following modes:

Table 21-3. Modes of operation

Mode	Description
Normal	In Normal mode, the eDMA transfers data between a source and a destination. The source and destination can be a memory block or an I/O block capable of operation with the eDMA. A service request initiates a transfer of a specific number of bytes (NBYTES) as specified in the transfer control descriptor (TCD). The minor loop is the sequence of read-write operations that transfers these NBYTES per service request. Each service request executes one iteration of the major loop, which transfers NBYTES of data.
Debug	DMA operation is configurable in Debug mode via the control register: • If CR[EDBG] is cleared, the DMA continues to operate. • If CR[EDBG] is set, the eDMA stops transferring data. If Debug mode is entered while a channel is active, the eDMA continues operation until the channel retires.
Wait	Before entering Wait mode, the DMA attempts to complete its current transfer. After the transfer completes, the device enters Wait mode.

21.3 Memory map/register definition

The eDMA's programming model is partitioned into two regions:

- The first region defines a number of registers providing control functions
- The second region corresponds to the local transfer control descriptor memory

Each channel requires a 32-byte transfer control descriptor for defining the desired data movement operation. The channel descriptors are stored in the local memory in sequential order: channel 0, channel 1,... channel 3 . Each TCD*n* definition is presented as 11 registers of 16 or 32 bits.

Reading reserved bits in a register returns the value of zero. Writes to reserved bits in a register are ignored. Reading or writing a reserved memory location generates a bus error.

DMA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_8000	Control Register (DMA_CR)	32	R/W	0000_0000h	21.3.1/342
4000_8004	Error Status Register (DMA_ES)	32	R	0000_0000h	21.3.2/344
4000_800C	Enable Request Register (DMA_ERQ)	32	R/W	0000_0000h	21.3.3/346
4000_8014	Enable Error Interrupt Register (DMA_EEI)	32	R/W	0000_0000h	21.3.4/347
4000_8018	Clear Enable Error Interrupt Register (DMA_CEEI)	8	W (always reads zero)	00h	21.3.5/348
4000_8019	Set Enable Error Interrupt Register (DMA_SEEI)	8	W (always reads zero)	00h	21.3.6/349
4000_801A	Clear Enable Request Register (DMA_CERQ)	8	W (always reads zero)	00h	21.3.7/350
4000_801B	Set Enable Request Register (DMA_SERQ)	8	W (always reads zero)	00h	21.3.8/351
4000_801C	Clear DONE Status Bit Register (DMA_CDNE)	8	W (always reads zero)	00h	21.3.9/352
4000_801D	Set START Bit Register (DMA_SSRT)	8	W (always reads zero)	00h	21.3.10/ 353

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_801E	Clear Error Register (DMA_CERR)	8	W (always reads zero)	00h	21.3.11/ 354
4000_801F	Clear Interrupt Request Register (DMA_CINT)	8	W (always reads zero)	00h	21.3.12/ 355
4000_8024	Interrupt Request Register (DMA_INT)	32	R/W	0000_0000h	21.3.13/ 355
4000_802C	Error Register (DMA_ERR)	32	R/W	0000_0000h	21.3.14/ 357
4000_8034	Hardware Request Status Register (DMA_HRS)	32	R/W	0000_0000h	21.3.15/ 358
4000_8100	Channel n Priority Register (DMA_DCHPRI3)	8	R/W	Undefined	21.3.16/ 359
4000_8101	Channel n Priority Register (DMA_DCHPRI2)	8	R/W	Undefined	21.3.16/ 359
4000_8102	Channel n Priority Register (DMA_DCHPRI1)	8	R/W	Undefined	21.3.16/ 359
4000_8103	Channel n Priority Register (DMA_DCHPRI0)	8	R/W	Undefined	21.3.16/ 359
4000_9000	TCD Source Address (DMA_TCD0_SADDR)	32	R/W	Undefined	21.3.17/ 360
4000_9004	TCD Signed Source Address Offset (DMA_TCD0_SOFF)	16	R/W	Undefined	21.3.18/ 360
4000_9006	TCD Transfer Attributes (DMA_TCD0_ATTR)	16	R/W	Undefined	21.3.19/ 361
4000_9008	TCD Minor Byte Count (Minor Loop Disabled) (DMA_TCD0_NBYTES_MLNO)	32	R/W	Undefined	21.3.20/ 362
4000_9008	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA_TCD0_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.21/ 363
4000_9008	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA_TCD0_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.22/ 364
4000_900C	TCD Last Source Address Adjustment (DMA_TCD0_SLAST)	32	R/W	Undefined	21.3.23/ 365
4000_9010	TCD Destination Address (DMA_TCD0_DADDR)	32	R/W	Undefined	21.3.24/ 365
4000_9014	TCD Signed Destination Address Offset (DMA_TCD0_DOFF)	16	R/W	Undefined	21.3.25/ 366
4000_9016	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD0_CITER_ELINKYES)	16	R/W	Undefined	21.3.26/ 367

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_9016	DMA_TCD0_CITER_ELINKNO	16	R/W	Undefined	21.3.27/ 368
4000_9018	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD0_DLASTSGA)	32	R/W	Undefined	21.3.28/ 369
4000_901C	TCD Control and Status (DMA_TCD0_CSR)	16	R/W	Undefined	21.3.29/ 370
4000_901E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD0_BITER_ELINKYES)	16	R/W	Undefined	21.3.30/ 372
4000_901E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD0_BITER_ELINKNO)	16	R/W	Undefined	21.3.31/ 373
4000_9020	TCD Source Address (DMA_TCD1_SADDR)	32	R/W	Undefined	21.3.17/ 360
4000_9024	TCD Signed Source Address Offset (DMA_TCD1_SOFF)	16	R/W	Undefined	21.3.18/ 360
4000_9026	TCD Transfer Attributes (DMA_TCD1_ATTR)	16	R/W	Undefined	21.3.19/ 361
4000_9028	TCD Minor Byte Count (Minor Loop Disabled) (DMA_TCD1_NBYTES_MLNO)	32	R/W	Undefined	21.3.20/ 362
4000_9028	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA_TCD1_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.21/ 363
4000_9028	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA_TCD1_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.22/ 364
4000_902C	TCD Last Source Address Adjustment (DMA_TCD1_SLAST)	32	R/W	Undefined	21.3.23/ 365
4000_9030	TCD Destination Address (DMA_TCD1_DADDR)	32	R/W	Undefined	21.3.24/ 365
4000_9034	TCD Signed Destination Address Offset (DMA_TCD1_DOFF)	16	R/W	Undefined	21.3.25/ 366
4000_9036	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD1_CITER_ELINKYES)	16	R/W	Undefined	21.3.26/ 367
4000_9036	DMA_TCD1_CITER_ELINKNO	16	R/W	Undefined	21.3.27/ 368
4000_9038	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD1_DLASTSGA)	32	R/W	Undefined	21.3.28/ 369
4000_903C	TCD Control and Status (DMA_TCD1_CSR)	16	R/W	Undefined	21.3.29/ 370
4000_903E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD1_BITER_ELINKYES)	16	R/W	Undefined	21.3.30/ 372

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_903E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD1_BITER_ELINKNO)	16	R/W	Undefined	21.3.31/ 373
4000_9040	TCD Source Address (DMA_TCD2_SADDR)	32	R/W	Undefined	21.3.17/ 360
4000_9044	TCD Signed Source Address Offset (DMA_TCD2_SOFF)	16	R/W	Undefined	21.3.18/ 360
4000_9046	TCD Transfer Attributes (DMA_TCD2_ATTR)	16	R/W	Undefined	21.3.19/ 361
4000_9048	TCD Minor Byte Count (Minor Loop Disabled) (DMA_TCD2_NBYTES_MLNO)	32	R/W	Undefined	21.3.20/ 362
4000_9048	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA_TCD2_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.21/ 363
4000_9048	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA_TCD2_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.22/ 364
4000_904C	TCD Last Source Address Adjustment (DMA_TCD2_SLAST)	32	R/W	Undefined	21.3.23/ 365
4000_9050	TCD Destination Address (DMA_TCD2_DADDR)	32	R/W	Undefined	21.3.24/ 365
4000_9054	TCD Signed Destination Address Offset (DMA_TCD2_DOFF)	16	R/W	Undefined	21.3.25/ 366
4000_9056	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD2_CITER_ELINKYES)	16	R/W	Undefined	21.3.26/ 367
4000_9056	DMA_TCD2_CITER_ELINKNO	16	R/W	Undefined	21.3.27/ 368
4000_9058	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD2_DLASTSGA)	32	R/W	Undefined	21.3.28/ 369
4000_905C	TCD Control and Status (DMA_TCD2_CSR)	16	R/W	Undefined	21.3.29/ 370
4000_905E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD2_BITER_ELINKYES)	16	R/W	Undefined	21.3.30/ 372
4000_905E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD2_BITER_ELINKNO)	16	R/W	Undefined	21.3.31/ 373
4000_9060	TCD Source Address (DMA_TCD3_SADDR)	32	R/W	Undefined	21.3.17/ 360
4000_9064	TCD Signed Source Address Offset (DMA_TCD3_SOFF)	16	R/W	Undefined	21.3.18/ 360
4000_9066	TCD Transfer Attributes (DMA_TCD3_ATTR)	16	R/W	Undefined	21.3.19/ 361

Table continues on the next page...

DMA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4000_9068	TCD Minor Byte Count (Minor Loop Disabled) (DMA_TCD3_NBYTES_MLNO)	32	R/W	Undefined	21.3.20/ 362
4000_9068	TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA_TCD3_NBYTES_MLOFFNO)	32	R/W	Undefined	21.3.21/ 363
4000_9068	TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA_TCD3_NBYTES_MLOFFYES)	32	R/W	Undefined	21.3.22/ 364
4000_906C	TCD Last Source Address Adjustment (DMA_TCD3_SLAST)	32	R/W	Undefined	21.3.23/ 365
4000_9070	TCD Destination Address (DMA_TCD3_DADDR)	32	R/W	Undefined	21.3.24/ 365
4000_9074	TCD Signed Destination Address Offset (DMA_TCD3_DOFF)	16	R/W	Undefined	21.3.25/ 366
4000_9076	TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD3_CITER_ELINKYES)	16	R/W	Undefined	21.3.26/ 367
4000_9076	DMA_TCD3_CITER_ELINKNO	16	R/W	Undefined	21.3.27/ 368
4000_9078	TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD3_DLASTSGA)	32	R/W	Undefined	21.3.28/ 369
4000_907C	TCD Control and Status (DMA_TCD3_CSR)	16	R/W	Undefined	21.3.29/ 370
4000_907E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD3_BITER_ELINKYES)	16	R/W	Undefined	21.3.30/ 372
4000_907E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD3_BITER_ELINKNO)	16	R/W	Undefined	21.3.31/ 373
4000_909E	TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD4_BITER_ELINKYES)	16	R/W	Undefined	21.3.30/ 372

21.3.1 Control Register (DMA_CR)

The CR defines the basic operating configuration of the DMA.

Arbitration can be configured to use either a fixed-priority or a round-robin scheme. For fixed-priority arbitration, the highest priority channel requesting service is selected to execute. The channel priority registers assign the priorities; see the DCHPRIn registers. For round-robin arbitration, the channel priorities are ignored and channels are cycled through without regard to priority.

NOTE

For proper operation, writes to the CR register must be performed only when the DMA channels are inactive; that is, when TCDn_CSR[ACTIVE] bits are cleared.

Address: DMA_CR is 4000_8000h base + 0h offset = 4000_8000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0														CX	ECX	0								EMLM	CLM	HALT	HOE	0	ERCA	EDBG	0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_CR field descriptions

Field	Description
31–18 Reserved	This read-only field is reserved and always has the value zero.
17 CX	Cancel Transfer 0 Normal operation 1 Cancel the remaining data transfer. Stop the executing channel and force the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. The CX bit clears itself after the cancel has been honored. This cancel retires the channel normally as if the minor loop was completed.
16 ECX	Error Cancel Transfer 0 Normal operation 1 Cancel the remaining data transfer in the same fashion as the CX bit. Stop the executing channel and force the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. The ECX bit clears itself after the cancel is honored. In addition to cancelling the transfer, ECX treats the cancel as an error condition, thus updating the ES register and generating an optional error interrupt.
15–8 Reserved	This read-only field is reserved and always has the value zero.
7 EMLM	Enable Minor Loop Mapping 0 Disabled. TCDn.word2 is defined as a 32-bit NBYTES field. 1 Enabled. TCDn.word2 is redefined to include individual enable fields, an offset field, and the NBYTES field. The individual enable fields allow the minor loop offset to be applied to the source address, the destination address, or both. The NBYTES field is reduced when either offset is enabled.
6 CLM	Continuous Link Mode 0 A minor loop channel link made to itself goes through channel arbitration before being activated again. 1 A minor loop channel link made to itself does not go through channel arbitration before being activated again. Upon minor loop completion, the channel activates again if that channel has a minor loop channel link enabled and the link channel is itself. This effectively applies the minor loop offsets and restarts the next minor loop.
5 HALT	Halt DMA Operations

Table continues on the next page...

DMA_CR field descriptions (continued)

Field	Description
	0 Normal operation 1 Stall the start of any new channels. Executing channels are allowed to complete. Channel execution resumes when this bit is cleared.
4 HOE	Halt On Error 0 Normal operation 1 Any error causes the HALT bit to set. Subsequently, all service requests are ignored until the HALT bit is cleared.
3 Reserved	This read-only field is reserved and always has the value zero.
2 ERCA	Enable Round Robin Channel Arbitration 0 Fixed priority arbitration is used for channel selection. 1 Round robin arbitration is used for channel selection.
1 EDBG	Enable Debug 0 When in debug mode, the DMA continues to operate. 1 When in debug mode, the DMA stalls the start of a new channel. Executing channels are allowed to complete. Channel execution resumes when the system exits debug mode or the EDBG bit is cleared.
0 Reserved	This read-only field is reserved and always has the value zero.

21.3.2 Error Status Register (DMA_ES)

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- A configuration error, that is:
  - An illegal setting in the transfer-control descriptor, or
  - An illegal priority register setting in fixed-arbitration
- An error termination to a bus master read or write cycle

See the Error Reporting and Handling section for more details.

Address: DMA_ES is 4000_8000h base + 4h offset = 4000_8004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VLD	0														ECX
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	CPE	0				ERRCHN		SAE	SOE	DAE	DOE	NCE	SGE	SBE	DBE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_ES field descriptions

Field	Description
31 VLD	Logical OR of all ERR status bits 0 No ERR bits are set 1 At least one ERR bit is set indicating a valid error exists that has not been cleared
30–17 Reserved	This read-only field is reserved and always has the value zero.
16 ECX	Transfer Cancelled 0 No cancelled transfers 1 The last recorded entry was a cancelled transfer by the error cancel transfer input
15 Reserved	This read-only field is reserved and always has the value zero.
14 CPE	Channel Priority Error 0 No channel priority error 1 The last recorded error was a configuration error in the channel priorities. Channel priorities are not unique.
13–10 Reserved	This read-only field is reserved and always has the value zero.
9–8 ERRCHN	Error Channel Number or Cancelled Channel Number The channel number of the last recorded error (excluding CPE errors) or last recorded error cancelled transfer.
7 SAE	Source Address Error 0 No source address configuration error. 1 The last recorded error was a configuration error detected in the TCDn_SADDR field. TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE].
6 SOE	Source Offset Error 0 No source offset configuration error 1 The last recorded error was a configuration error detected in the TCDn_SOFF field. TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE].
5 DAE	Destination Address Error 0 No destination address configuration error 1 The last recorded error was a configuration error detected in the TCDn_DADDR field. TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE].
4 DOE	Destination Offset Error 0 No destination offset configuration error 1 The last recorded error was a configuration error detected in the TCDn_DOFF field. TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE].
3 NCE	NBYTES/CITER Configuration Error 0 No NBYTES/CITER configuration error 1 The last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields.

Table continues on the next page...

DMA_ES field descriptions (continued)

Field	Description
	• TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE], or • TCDn_CITER[CITER] is equal to zero, or • TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK]
2 SGE	Scatter/Gather Configuration Error 0 No scatter/gather configuration error 1 The last recorded error was a configuration error detected in the TCDn_DLASTSGA field. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG] is enabled. TCDn_DLASTSGA is not on a 32 byte boundary.
1 SBE	Source Bus Error 0 No source bus error 1 The last recorded error was a bus error on a source read
0 DBE	Destination Bus Error 0 No destination bus error 1 The last recorded error was a bus error on a destination write

21.3.3 Enable Request Register (DMA_ERQ)

The ERQ register provides a bit map for the 4 implemented channels to enable the request signal for each channel. The state of any given channel enable is directly affected by writes to this register; it is also affected by writes to the SERQ and CERQ. The {S,C}ERQ registers are provided so the request enable for a single channel can easily be modified without needing to perform a read-modify-write sequence to the ERQ.

DMA request input signals and this enable request flag must be asserted before a channel's hardware service request is accepted. The state of the DMA enable request flag does not affect a channel service request made explicitly through software or a linked channel request.

Address: DMA_ERQ is 4000_8000h base + Ch offset = 4000_800Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																												ERQ3	ERQ2	ERQ1	ERQ0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_ERQ field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

DMA_ERQ field descriptions (continued)

Field	Description
3 ERQ3	Enable DMA Request 3 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
2 ERQ2	Enable DMA Request 2 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
1 ERQ1	Enable DMA Request 1 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled
0 ERQ0	Enable DMA Request 0 0 The DMA request signal for the corresponding channel is disabled 1 The DMA request signal for the corresponding channel is enabled

21.3.4 Enable Error Interrupt Register (DMA_EEI)

The EEI register provides a bit map for the 4 channels to enable the error interrupt signal for each channel. The state of any given channel's error interrupt enable is directly affected by writes to this register; it is also affected by writes to the SEEI and CEEI. The {S,C}EEI are provided so the error interrupt enable for a single channel can easily be modified without the need to perform a read-modify-write sequence to the EEI register.

The DMA error indicator and the error interrupt enable flag must be asserted before an error interrupt request for a given channel is asserted to the interrupt controller.

Address: DMA_EEI is 4000_8000h base + 14h offset = 4000_8014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																												EEI3	EEI2	EEI1	EEI0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_EEI field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero.
3 EEI3	Enable Error Interrupt 3

Table continues on the next page...

DMA_EEI field descriptions (continued)

Field	Description
	0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
2 EEI2	Enable Error Interrupt 2 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
1 EEI1	Enable Error Interrupt 1 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request
0 EEI0	Enable Error Interrupt 0 0 The error signal for corresponding channel does not generate an error interrupt 1 The assertion of the error signal for corresponding channel generates an error interrupt request

21.3.5 Clear Enable Error Interrupt Register (DMA_CEEI)

The CEEI provides a simple memory-mapped mechanism to clear a given bit in the EEI to disable the error interrupt for a given channel. The data value on a register write causes the corresponding bit in the EEI to be cleared. Setting the CAEE bit provides a global clear function, forcing the EEI contents to be cleared, disabling all DMA request inputs. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: DMA_CEEI is 4000_8000h base + 18h offset = 4000_8018h

Bit	7	6	5	4	3	2	1	0
Read	0	0					0	
Write	NOP	CAEE	0				CEEI	
Reset	0	0	0	0	0	0	0	0

DMA_CEEI field descriptions

Field	Description
7 NOP	0 Normal operation 1 No operation, ignore the other bits in this register
6 CAEE	Clear All Enable Error Interrupts 0 Clear only the EEI bit specified in the CEEI field 1 Clear all bits in EEI
5–2 Reserved	This field is reserved.

Table continues on the next page...

DMA_CEEI field descriptions (continued)

Field	Description
1–0 CEEI	Clear Enable Error Interrupt Clears the corresponding bit in EEI

21.3.6 Set Enable Error Interrupt Register (DMA_SEEI)

The SEEI provides a simple memory-mapped mechanism to set a given bit in the EEI to enable the error interrupt for a given channel. The data value on a register write causes the corresponding bit in the EEI to be set. Setting the SAEE bit provides a global set function, forcing the entire EEI contents to be set. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: DMA_SEEI is 4000_8000h base + 19h offset = 4000_8019h

Bit	7	6	5	4	3	2	1	0
Read	0	0					0	
Write	NOP	SAEE	0				SEEI	
Reset	0	0	0	0	0	0	0	0

DMA_SEEI field descriptions

Field	Description
7 NOP	0 Normal operation 1 No operation, ignore the other bits in this register
6 SAEE	Sets All Enable Error Interrupts 0 Set only the EEI bit specified in the SEEI field. 1 Sets all bits in EEI
5–2 Reserved	This field is reserved.
1–0 SEEI	Set Enable Error Interrupt Sets the corresponding bit in EEI

21.3.7 Clear Enable Request Register (DMA_CERQ)

The CERQ provides a simple memory-mapped mechanism to clear a given bit in the ERQ to disable the DMA request for a given channel. The data value on a register write causes the corresponding bit in the ERQ to be cleared. Setting the CAER bit provides a global clear function, forcing the entire contents of the ERQ to be cleared, disabling all DMA request inputs. If NOP is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: DMA_CERQ is 4000_8000h base + 1Ah offset = 4000_801Ah

Bit	7	6	5	4	3	2	1	0
Read	0	0					0	
Write	NOP	CAER	0				CERQ	
Reset	0	0	0	0	0	0	0	0

DMA_CERQ field descriptions

Field	Description
7 NOP	0 Normal operation 1 No operation, ignore the other bits in this register
6 CAER	Clear All Enable Requests 0 Clear only the ERQ bit specified in the CERQ field 1 Clear all bits in ERQ
5–2 Reserved	This field is reserved.
1–0 CERQ	Clear Enable Request Clears the corresponding bit in ERQ

21.3.8 Set Enable Request Register (DMA_SERQ)

The SERQ provides a simple memory-mapped mechanism to set a given bit in the ERQ to enable the DMA request for a given channel. The data value on a register write causes the corresponding bit in the ERQ to be set. Setting the SAER bit provides a global set function, forcing the entire contents of ERQ to be set. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: DMA_SERQ is 4000_8000h base + 1Bh offset = 4000_801Bh

Bit	7	6	5	4	3	2	1	0
Read	0	0					0	
Write	NOP	SAER	0				SERQ	
Reset	0	0	0	0	0	0	0	0

DMA_SERQ field descriptions

Field	Description
7 NOP	0 Normal operation 1 No operation, ignore the other bits in this register
6 SAER	Set All Enable Requests 0 Set only the ERQ bit specified in the SERQ field 1 Set all bits in ERQ
5–2 Reserved	This field is reserved.
1–0 SERQ	Set enable request Sets the corresponding bit in ERQ

21.3.9 Clear DONE Status Bit Register (DMA_CDNE)

The CDNE provides a simple memory-mapped mechanism to clear the DONE bit in the TCD of the given channel. The data value on a register write causes the DONE bit in the corresponding transfer control descriptor to be cleared. Setting the CADN bit provides a global clear function, forcing all DONE bits to be cleared. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: DMA_CDNE is 4000_8000h base + 1Ch offset = 4000_801Ch

Bit	7	6	5	4	3	2	1	0
Read	0	0					0	
Write	NOP	CADN	0				CDNE	
Reset	0	0	0	0	0	0	0	0

DMA_CDNE field descriptions

Field	Description
7 NOP	0 Normal operation 1 No operation, ignore the other bits in this register
6 CADN	Clears All DONE Bits 0 Clears only the TCDn_CSR[DONE] bit specified in the CDNE field 1 Clears all bits in TCDn_CSR[DONE]
5–2 Reserved	This field is reserved.
1–0 CDNE	Clear DONE Bit Clears the corresponding bit in TCDn_CSR[DONE]

21.3.10 Set START Bit Register (DMA_SSRT)

The SSRT provides a simple memory-mapped mechanism to set the START bit in the TCD of the given channel. The data value on a register write causes the START bit in the corresponding transfer control descriptor to be set. Setting the SAST bit provides a global set function, forcing all START bits to be set. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: DMA_SSRT is 4000_8000h base + 1Dh offset = 4000_801Dh

Bit	7	6	5	4	3	2	1	0
Read	0	0					0	
Write	NOP	SAST	0				SSRT	
Reset	0	0	0	0	0	0	0	0

DMA_SSRT field descriptions

Field	Description
7 NOP	0 Normal operation 1 No operation, ignore the other bits in this register
6 SAST	Set All START Bits (activates all channels) 0 Set only the TCDn_CSR[START] bit specified in the SSRT field 1 Set all bits in TCDn_CSR[START]
5–2 Reserved	This field is reserved.
1–0 SSRT	Set START Bit Sets the corresponding bit in TCDn_CSR[START]

21.3.11 Clear Error Register (DMA_CERR)

The CERR provides a simple memory-mapped mechanism to clear a given bit in the ERR to disable the error condition flag for a given channel. The given value on a register write causes the corresponding bit in the ERR to be cleared. Setting the CAEI bit provides a global clear function, forcing the ERR contents to be cleared, clearing all channel error indicators. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: DMA_CERR is 4000_8000h base + 1Eh offset = 4000_801Eh

Bit	7	6	5	4	3	2	1	0
Read	0	0					0	
Write	NOP	CAEI	0				CERR	
Reset	0	0	0	0	0	0	0	0

DMA_CERR field descriptions

Field	Description
7 NOP	0 Normal operation 1 No operation, ignore the other bits in this register
6 CAEI	Clear All Error Indicators 0 Clear only the ERR bit specified in the CERR field 1 Clear all bits in ERR
5–2 Reserved	This field is reserved.
1–0 CERR	Clear Error Indicator Clears the corresponding bit in ERR

21.3.12 Clear Interrupt Request Register (DMA_CINT)

The CINT provides a simple, memory-mapped mechanism to clear a given bit in the INT to disable the interrupt request for a given channel. The given value on a register write causes the corresponding bit in the INT to be cleared. Setting the CAIR bit provides a global clear function, forcing the entire contents of the INT to be cleared, disabling all DMA interrupt requests. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: DMA_CINT is 4000_8000h base + 1Fh offset = 4000_801Fh

Bit	7	6	5	4	3	2	1	0
Read	0	0					0	
Write	NOP	CAIR	0				CINT	
Reset	0	0	0	0	0	0	0	0

DMA_CINT field descriptions

Field	Description
7 NOP	0 Normal operation 1 No operation, ignore the other bits in this register
6 CAIR	Clear All Interrupt Requests 0 Clear only the INT bit specified in the CINT field 1 Clear all bits in INT
5–2 Reserved	This field is reserved.
1–0 CINT	Clear Interrupt Request Clears the corresponding bit in INT

21.3.13 Interrupt Request Register (DMA_INT)

The INT register provides a bit map for the 4 channels signaling the presence of an interrupt request for each channel. Depending on the appropriate bit setting in the transfer-control descriptors, the eDMA engine generates an interrupt on data transfer completion. The outputs of this register are directly routed to the interrupt controller (INTC). During the interrupt-service routine associated with any given channel, it is the software's responsibility to clear the appropriate bit, negating the interrupt request. Typically, a write to the CINT register in the interrupt service routine is used for this purpose.

The state of any given channel's interrupt request is directly affected by writes to this register; it is also affected by writes to the CINT register. On writes to INT, a 1 in any bit position clears the corresponding channel's interrupt request. A zero in any bit position has no affect on the corresponding channel's current interrupt status. The CINT register is provided so the interrupt request for a single channel can easily be cleared without the need to perform a read-modify-write sequence to the INT register.

Address: DMA_INT is 4000_8000h base + 24h offset = 4000_8024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												INT3	INT2	INT1	INT0
W													w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_INT field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero.
3 INT3	Interrupt Request 3 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
2 INT2	Interrupt Request 2 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
1 INT1	Interrupt Request 1 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active
0 INT0	Interrupt Request 0 0 The interrupt request for corresponding channel is cleared 1 The interrupt request for corresponding channel is active

21.3.14 Error Register (DMA_ERR)

The ERR provides a bit map for the 4 channels, signaling the presence of an error for each channel. The eDMA engine signals the occurrence of an error condition by setting the appropriate bit in this register. The outputs of this register are enabled by the contents of the EEI, and then routed to the interrupt controller. During the execution of the interrupt-service routine associated with any DMA errors, it is software's responsibility to clear the appropriate bit, negating the error-interrupt request. Typically, a write to the CERR in the interrupt-service routine is used for this purpose. The normal DMA channel completion indicators (setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request) are not affected when an error is detected.

The contents of this register can also be polled because a non-zero value indicates the presence of a channel error regardless of the state of the EEI. The state of any given channel's error indicators is affected by writes to this register; it is also affected by writes to the CERR. On writes to the ERR, a one in any bit position clears the corresponding channel's error status. A zero in any bit position has no affect on the corresponding channel's current error status. The CERR is provided so the error indicator for a single channel can easily be cleared.

Address: DMA_ERR is 4000_8000h base + 2Ch offset = 4000_802Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												ERR3	ERR2	ERR1	ERR0
W													w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_ERR field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero.
3 ERR3	Error In Channel 3 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
2 ERR2	Error In Channel 2 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred

Table continues on the next page...

DMA_ERR field descriptions (continued)

Field	Description
1 ERR1	Error In Channel 1 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred
0 ERR0	Error In Channel 0 0 An error in the corresponding channel has not occurred 1 An error in the corresponding channel has occurred

21.3.15 Hardware Request Status Register (DMA_HRS)

The HRS provides a bit map for the DMA channels, signaling the presence of a hardware request for each channel. The hardware request status bits reflect the current state of the register and qualified (via the ERQ fields) DMA request signals as seen by the DMA's arbitration logic. This view into the hardware request signals may be used for debug purposes.

NOTE

These bits reflect the state of the request as seen by the arbitration logic. Therefore, this status is affected by the ERQ bits.

Address: DMA_HRS is 4000_8000h base + 34h offset = 4000_8034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																												HRS3	HRS2	HRS1	HRS0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

DMA_HRS field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero.
3 HRS3	Hardware Request Status Channel 3 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
2 HRS2	Hardware Request Status Channel 2 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present

Table continues on the next page...

DMA_HRS field descriptions (continued)

Field	Description
1 HRS1	Hardware Request Status Channel 1 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present
0 HRS0	Hardware Request Status Channel 0 0 A hardware service request for the corresponding channel is not present 1 A hardware service request for the corresponding channel is present

21.3.16 Channel n Priority Register (DMA_DCHPRI*n*)

When fixed-priority channel arbitration is enabled (CR[ERCA] = 0), the contents of these registers define the unique priorities associated with each channel. The channel priorities are evaluated by numeric value; for example, 0 is the lowest priority, 1 is the next priority, then 2, then 3. Software must program the channel priorities with unique values. Otherwise, a configuration error is reported. The range of the priority value is limited to the values of 0 through 3.

Addresses: DCHPRI3 is 4000_8000h base + 100h offset = 4000_8100h
DCHPRI2 is 4000_8000h base + 101h offset = 4000_8101h
DCHPRI1 is 4000_8000h base + 102h offset = 4000_8102h
DCHPRI0 is 4000_8000h base + 103h offset = 4000_8103h

Bit	7	6	5	4	3	2	1	0
Read	ECP	DPA	0				CHPRI	
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_DCHPRI*n* field descriptions

Field	Description
7 ECP	Enable Channel Preemption 0 Channel n cannot be suspended by a higher priority channel's service request 1 Channel n can be temporarily suspended by the service request of a higher priority channel
6 DPA	Disable Preempt Ability 0 Channel n can suspend a lower priority channel 1 Channel n cannot suspend any channel, regardless of channel priority

Table continues on the next page...

DMA_DCHPRI*n* field descriptions (continued)

Field	Description
5–2 Reserved	This read-only field is reserved and always has the value zero.
1–0 CHPRI	Channel n Arbitration Priority Channel priority when fixed-priority arbitration is enabled **NOTE:** Reset value for the channel priority fields, CHPRI, is equal to the corresponding channel number for each priority register, i.e., DCHPRI3[CHPRI] equals 0b11.

21.3.17 TCD Source Address (DMA_TCD_SADDR)

Addresses: TCD0_SADDR is 4000_8000h base + 1000h offset = 4000_9000h
TCD1_SADDR is 4000_8000h base + 1020h offset = 4000_9020h
TCD2_SADDR is 4000_8000h base + 1040h offset = 4000_9040h
TCD3_SADDR is 4000_8000h base + 1060h offset = 4000_9060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SADDR																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

DMA_TCD*n*_SADDR field descriptions

Field	Description
31–0 SADDR	Source Address Memory address pointing to the source data.

21.3.18 TCD Signed Source Address Offset (DMA_TCD_SOFF)

Addresses: TCD0_SOFF is 4000_8000h base + 1004h offset = 4000_9004h
TCD1_SOFF is 4000_8000h base + 1024h offset = 4000_9024h
TCD2_SOFF is 4000_8000h base + 1044h offset = 4000_9044h
TCD3_SOFF is 4000_8000h base + 1064h offset = 4000_9064h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SOFF															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

DMA_TCD*n*_SOFF field descriptions

Field	Description
15–0 SOFF	Source address signed offset Sign-extended offset applied to the current source address to form the next-state value as each source read is completed.

21.3.19 TCD Transfer Attributes (DMA_TCD_ATTR)

Addresses: TCD0_ATTR is 4000_8000h base + 1006h offset = 4000_9006h
TCD1_ATTR is 4000_8000h base + 1026h offset = 4000_9026h
TCD2_ATTR is 4000_8000h base + 1046h offset = 4000_9046h
TCD3_ATTR is 4000_8000h base + 1066h offset = 4000_9066h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	SMOD					SSIZE			DMOD					DSIZE		
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_ATTR field descriptions

Field	Description
15–11 SMOD	Source Address Modulo. 0 Source address modulo feature is disabled ≠0 This value defines a specific address range specified to be the value after SADDR + SOFF calculation is performed or the original register value. The setting of this field provides the ability to implement a circular data queue easily. For data queues requiring power-of-2 size bytes, the queue should start at a 0-modulo-size address and the SMOD field should be set to the appropriate value for the queue, freezing the desired number of upper address bits. The value programmed into this field specifies the number of lower address bits allowed to change. For a circular queue application, the SOFF is typically set to the transfer size to implement post-increment addressing with the SMOD function constraining the addresses to a 0-modulo-size range.
10–8 SSIZE	Source data transfer size The attempted use of a Reserved encoding causes a configuration error. 000 8-bit 001 16-bit 010 32-bit 011 Reserved 100 16-byte 101 32-byte 110 Reserved 111 Reserved

Table continues on the next page...

DMA_TCD*n*_ATTR field descriptions (continued)

Field	Description
7–3 DMOD	Destination Address Modulo See the SMOD definition
2–0 DSIZE	Destination Data Transfer Size See the SSIZE definition

21.3.20 TCD Minor Byte Count (Minor Loop Disabled) (DMA_TCD_NBYTES_MLNO)

TCD word 2's register definition depends on the status of minor loop mapping. If minor loop mapping is disabled (CR[EMLM] = 0), TCD word 2 is defined as follows. If minor loop mapping is enabled, see the TCD_NBYTES_MLOFFNO and TCD_NBYTES_MLOFFYES register descriptions for TCD word 2's register definition.

Addresses: TCD0_NBYTES_MLNO is 4000_8000h base + 1008h offset = 4000_9008h
TCD1_NBYTES_MLNO is 4000_8000h base + 1028h offset = 4000_9028h
TCD2_NBYTES_MLNO is 4000_8000h base + 1048h offset = 4000_9048h
TCD3_NBYTES_MLNO is 4000_8000h base + 1068h offset = 4000_9068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NBYTES																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_NBYTES_MLNO field descriptions

Field	Description
31–0 NBYTES	Minor Byte Transfer Count Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes perform until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. (Although, it may be stalled by using the bandwidth control field, or via preemption.) After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed. **NOTE:** An NBYTES value of 0x0000_0000 is interpreted as a 4 GB transfer.

21.3.21 TCD Signed Minor Loop Offset (Minor Loop Enabled and Offset Disabled) (DMA_TCD_NBYTES_MLOFFNO)

TCD word 2 is defined as follows if:

- Minor loop mapping is enabled (CR[EMLM] = 1) and
- SMLOE = 0 and DMLOE = 0

If minor loop mapping is enabled and SMLOE or DMLOE is set then refer to the TCD_NBYTES_MLOFFYES register description.

Addresses: TCD0_NBYTES_MLOFFNO is 4000_8000h base + 1008h offset = 4000_9008h
TCD1_NBYTES_MLOFFNO is 4000_8000h base + 1028h offset = 4000_9028h
TCD2_NBYTES_MLOFFNO is 4000_8000h base + 1048h offset = 4000_9048h
TCD3_NBYTES_MLOFFNO is 4000_8000h base + 1068h offset = 4000_9068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SMLOE	DMLOE	NBYTES																													
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_NBYTES_MLOFFNO field descriptions

Field	Description
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0 The minor loop offset is not applied to the SADDR 1 The minor loop offset is applied to the SADDR
30 DMLOE	Destination Minor Loop Offset enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0 The minor loop offset is not applied to the DADDR 1 The minor loop offset is applied to the DADDR
29–0 NBYTES	Minor Byte Transfer Count Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes perform until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted; although, it may be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed.

21.3.22 TCD Signed Minor Loop Offset (Minor Loop and Offset Enabled) (DMA_TCD_NBYTES_MLOFFYES)

TCD word 2 is defined as follows if:

- Minor loop mapping is enabled (CR[EMLM] = 1) and
- Minor loop offset enabled (SMLOE or DMLOE = 1)

If minor loop mapping is enabled and SMLOE and DMLOE are cleared then refer to the TCD_NBYTES_MLOFFNO register description.

Addresses: TCD0_NBYTES_MLOFFYES is 4000_8000h base + 1008h offset = 4000_9008h
TCD1_NBYTES_MLOFFYES is 4000_8000h base + 1028h offset = 4000_9028h
TCD2_NBYTES_MLOFFYES is 4000_8000h base + 1048h offset = 4000_9048h
TCD3_NBYTES_MLOFFYES is 4000_8000h base + 1068h offset = 4000_9068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SMLOE	DMLOE	MLOFF																				NBYTES									
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_NBYTES_MLOFFYES field descriptions

Field	Description
31 SMLOE	Source Minor Loop Offset Enable Selects whether the minor loop offset is applied to the source address upon minor loop completion. 0 The minor loop offset is not applied to the SADDR 1 The minor loop offset is applied to the SADDR
30 DMLOE	Destination Minor Loop Offset enable Selects whether the minor loop offset is applied to the destination address upon minor loop completion. 0 The minor loop offset is not applied to the DADDR 1 The minor loop offset is applied to the DADDR
29–10 MLOFF	If SMLOE or DMLOE is set, this field represents a sign-extended offset applied to the source or destination address to form the next-state value after the minor loop completes.
9–0 NBYTES	Minor Byte Transfer Count Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes perform until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. (Although, it may be stalled by using the bandwidth control field, or via

Table continues on the next page...

DMA_TCD*n*_NBYTES_MLOFFYES field descriptions (continued)

Field	Description
	preemption.) After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed.

21.3.23 TCD Last Source Address Adjustment (DMA_TCD_SLAST)

Addresses: TCD0_SLAST is 4000_8000h base + 100Ch offset = 4000_900Ch
TCD1_SLAST is 4000_8000h base + 102Ch offset = 4000_902Ch
TCD2_SLAST is 4000_8000h base + 104Ch offset = 4000_904Ch
TCD3_SLAST is 4000_8000h base + 106Ch offset = 4000_906Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SLAST																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_SLAST field descriptions

Field	Description
31–0 SLAST	Last source Address Adjustment Adjustment value added to the source address at the completion of the major iteration count. This value can be applied to restore the source address to the initial value, or adjust the address to reference the next data structure.

21.3.24 TCD Destination Address (DMA_TCD_DADDR)

Addresses: TCD0_DADDR is 4000_8000h base + 1010h offset = 4000_9010h
TCD1_DADDR is 4000_8000h base + 1030h offset = 4000_9030h
TCD2_DADDR is 4000_8000h base + 1050h offset = 4000_9050h
TCD3_DADDR is 4000_8000h base + 1070h offset = 4000_9070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DADDR																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_DADDR field descriptions

Field	Description
31–0 DADDR	Destination Address Memory address pointing to the destination data.

21.3.25 TCD Signed Destination Address Offset (DMA_TCD_DOFF)

Addresses: TCD0_DOFF is 4000_8000h base + 1014h offset = 4000_9014h
TCD1_DOFF is 4000_8000h base + 1034h offset = 4000_9034h
TCD2_DOFF is 4000_8000h base + 1054h offset = 4000_9054h
TCD3_DOFF is 4000_8000h base + 1074h offset = 4000_9074h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	DOFF															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_DOFF field descriptions

Field	Description
15–0 DOFF	Destination Address Signed offset Sign-extended offset applied to the current destination address to form the next-state value as each destination write is completed.

21.3.26 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD_CITER_ELINKYES)

If TCDn_CITER[ELINK] is set, the TCDn_CITER register is defined as follows.

Addresses: TCD0_CITER_ELINKYES is 4000_8000h base + 1016h offset = 4000_9016h
TCD1_CITER_ELINKYES is 4000_8000h base + 1036h offset = 4000_9036h
TCD2_CITER_ELINKYES is 4000_8000h base + 1056h offset = 4000_9056h
TCD3_CITER_ELINKYES is 4000_8000h base + 1076h offset = 4000_9076h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	ELINK	0				LINKCH		CITER								
Write																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_CITER_ELINKYES field descriptions

Field	Description
15 ELINK	Enable channel-to-channel linking on minor-loop complete As the channel completes the minor loop, this flag enables linking to another channel, defined by the LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** This bit must be equal to the BITER[ELINK] bit. Otherwise, a configuration error is reported. 0 The channel-to-channel linking is disabled 1 The channel-to-channel linking is enabled
14–11 Reserved	This read-only field is reserved and always has the value zero.
10–9 LINKCH	Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request to the channel defined by these four bits by setting that channel's TCDn_CSR[START] bit.
8–0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the minor loop is completed and updated in the transfer control descriptor memory. After the major iteration count is exhausted, the channel performs a number of operations (e.g.,

Table continues on the next page...

DMA_TCD*n*_CITER_ELINKYES field descriptions (continued)

Field	Description
	final source and destination address calculations), optionally generating an interrupt to signal channel completion before reloading the CITER field from the beginning iteration count (BITER) field. **NOTE:** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE:** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

21.3.27 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD_CITER_ELINKNO)

If TCDn_CITER[ELINK] is cleared, the TCDn_CITER register is defined as follows.

Addresses: TCD0_CITER_ELINKNO is 4000_8000h base + 1016h offset = 4000_9016h
TCD1_CITER_ELINKNO is 4000_8000h base + 1036h offset = 4000_9036h
TCD2_CITER_ELINKNO is 4000_8000h base + 1056h offset = 4000_9056h
TCD3_CITER_ELINKNO is 4000_8000h base + 1076h offset = 4000_9076h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	ELINK	CITER														
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_CITER_ELINKNO field descriptions

Field	Description
15 ELINK	Enable channel-to-channel linking on minor-loop complete As the channel completes the minor loop, this flag enables linking to another channel, defined by the LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** This bit must be equal to the BITER[ELINK] bit. Otherwise, a configuration error is reported. 0 The channel-to-channel linking is disabled 1 The channel-to-channel linking is enabled

Table continues on the next page...

DMA_TCD*n*_CITER_ELINKNO field descriptions (continued)

Field	Description
14–0 CITER	Current Major Iteration Count This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the minor loop is completed and updated in the transfer control descriptor memory. After the major iteration count is exhausted, the channel performs a number of operations (e.g., final source and destination address calculations), optionally generating an interrupt to signal channel completion before reloading the CITER field from the beginning iteration count (BITER) field. **NOTE:** When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field. **NOTE:** If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

21.3.28 TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD_DLASTSGA)

Addresses: TCD0_DLASTSGA is 4000_8000h base + 1018h offset = 4000_9018h
TCD1_DLASTSGA is 4000_8000h base + 1038h offset = 4000_9038h
TCD2_DLASTSGA is 4000_8000h base + 1058h offset = 4000_9058h
TCD3_DLASTSGA is 4000_8000h base + 1078h offset = 4000_9078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DLASTSGA																															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_DLASTSGA field descriptions

Field	Description

31–0
DLASTSGA
	Destination last address adjustment or the memory address for the next transfer control descriptor to be loaded into this channel (scatter/gather).
	If (TCDn_CSR[ESG] = 0) then
	• Adjustment value added to the destination address at the completion of the major iteration count. This value can apply to restore the destination address to the initial value or adjust the address to reference the next data structure.
	else
	• This address points to the beginning of a 0-modulo-32-byte region containing the next transfer control descriptor to be loaded into this channel. This channel reload is performed as the major iteration count completes. The scatter/gather address must be 0-modulo-32-byte, else a configuration error is reported.

21.3.29 TCD Control and Status (DMA_TCD_CSR)

Addresses: TCD0_CSR is 4000_8000h base + 101Ch offset = 4000_901Ch
TCD1_CSR is 4000_8000h base + 103Ch offset = 4000_903Ch
TCD2_CSR is 4000_8000h base + 105Ch offset = 4000_905Ch
TCD3_CSR is 4000_8000h base + 107Ch offset = 4000_907Ch

Bit	15	14	13	12	11	10	9	8
Read	BWC		0				MAJORLINKCH	
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

Bit	7	6	5	4	3	2	1	0
Read Write	DONE	ACTIVE	MAJORELINK	ESG	DREQ	INTHALF	INTMAJOR	START
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_CSR field descriptions

Field	Description
15–14 BWC	Bandwidth Control Throttles the amount of bus bandwidth consumed by the eDMA. In general, as the eDMA processes the minor loop, it continuously generates read/write sequences until the minor count is exhausted. This field forces the eDMA to stall after the completion of each read/write access to control the bus request bandwidth seen by the crossbar switch. **NOTE:** If the source and destination sizes are equal, this field is ignored between the first and second transfers and after the last write of each minor loop. This behavior is a side effect of reducing start-up latency. 00 No eDMA engine stalls 01 Reserved 10 eDMA engine stalls for 4 cycles after each r/w 11 eDMA engine stalls for 8 cycles after each r/w
13–10 Reserved	This read-only field is reserved and always has the value zero.

9–8
MAJORLINKCH
	Link Channel Number
	If (MAJORELINK = 0) then
	• No channel-to-channel linking (or chaining) is performed after the major loop counter is exhausted.
	else
	• After the major loop counter is exhausted, the eDMA engine initiates a channel service request at the channel defined by these six bits by setting that channel's TCDn_CSR[START] bit.

7 DONE	Channel Done

Table continues on the next page...

DMA_TCD*n*_CSR field descriptions (continued)

Field	Description
	This flag indicates the eDMA has completed the major loop. The eDMA engine sets it as the CITER count reaches zero; The software clears it, or the hardware when the channel is activated. **NOTE:** This bit must be cleared to write the MAJORELINK or ESG bits.
6 ACTIVE	Channel Active This flag signals the channel is currently in execution. It is set when channel service begins, and the eDMA clears it as the minor loop completes or if any error condition is detected. This bit resets to zero.
5 MAJORELINK	Enable channel-to-channel linking on major loop complete As the channel completes the major loop, this flag enables the linking to another channel, defined by MAJORLINKCH. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. **NOTE:** To support the dynamic linking coherency model, this field is forced to zero when written to while the TCDn_CSR[DONE] bit is set. 0 The channel-to-channel linking is disabled 1 The channel-to-channel linking is enabled
4 ESG	Enable Scatter/Gather Processing As the channel completes the major loop, this flag enables scatter/gather processing in the current channel. If enabled, the eDMA engine uses DLASTSGA as a memory pointer to a 0-modulo-32 address containing a 32-byte data structure loaded as the transfer control descriptor into the local memory. **NOTE:** To support the dynamic scatter/gather coherency model, this field is forced to zero when written to while the TCDn_CSR[DONE] bit is set. 0 The current channel's TCD is normal format. 1 The current channel's TCD specifies a scatter gather format. The DLASTSGA field provides a memory pointer to the next TCD to be loaded into this channel after the major loop completes its execution.
3 DREQ	Disable Request If this flag is set, the eDMA hardware automatically clears the corresponding ERQ bit when the current major iteration count reaches zero. 0 The channel's ERQ bit is not affected 1 The channel's ERQ bit is cleared when the major loop is complete
2 INTHALF	Enable an interrupt when major counter is half complete. If this flag is set, the channel generates an interrupt request by setting the appropriate bit in the INT register when the current major iteration count reaches the halfway point. Specifically, the comparison performed by the eDMA engine is (CITER == (BITER >> 1)). This halfway point interrupt request is provided to support double-buffered (aka ping-pong) schemes or other types of data movement where the processor needs an early indication of the transfer's progress. If BITER is set, do not use INTHALF. Use INTMAJOR instead. 0 The half-point interrupt is disabled 1 The half-point interrupt is enabled
1 INTMAJOR	Enable an interrupt when major iteration count completes

Table continues on the next page...

DMA_TCD*n*_CSR field descriptions (continued)

Field	Description
	If this flag is set, the channel generates an interrupt request by setting the appropriate bit in the INT when the current major iteration count reaches zero. 0 The end-of-major loop interrupt is disabled 1 The end-of-major loop interrupt is enabled
0 START	Channel Start If this flag is set, the channel is requesting service. The eDMA hardware automatically clears this flag after the channel begins execution. 0 The channel is not explicitly started 1 The channel is explicitly started via a software initiated service request

21.3.30 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD_BITER_ELINKYES)

If the TCDn_BITER[ELINK] bit is set, the TCDn_BITER register is defined as follows.

Addresses: TCD0_BITER_ELINKYES is 4000_8000h base + 101Eh offset = 4000_901Eh
TCD1_BITER_ELINKYES is 4000_8000h base + 103Eh offset = 4000_903Eh
TCD2_BITER_ELINKYES is 4000_8000h base + 105Eh offset = 4000_905Eh
TCD3_BITER_ELINKYES is 4000_8000h base + 107Eh offset = 4000_907Eh
TCD4_BITER_ELINKYES is 4000_8000h base + 109Eh offset = 4000_909Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	ELINK	0				LINKCH		BITER								
Write																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_BITER_ELINKYES field descriptions

Field	Description
15 ELINK	Enables channel-to-channel linking on minor loop complete As the channel completes the minor loop, this flag enables the linking to another channel, defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking disables, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field. Otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field is reloaded into the CITER field.

Table continues on the next page...

DMA_TCD*n*_BITER_ELINKYES field descriptions (continued)

Field	Description
	0 The channel-to-channel linking is disabled 1 The channel-to-channel linking is enabled
14–11 Reserved	This read-only field is reserved and always has the value zero.
10–9 LINKCH	Link Channel Number If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request at the channel defined by these four bits by setting that channel's TCDn_CSR[START] bit. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field. Otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field is reloaded into the CITER field.
8–0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be equal to the value in the CITER field. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field. Otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field is reloaded into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

21.3.31 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD_BITER_ELINKNO)

If the TCDn_BITER[ELINK] bit is cleared, the TCDn_BITER register is defined as follows.

Addresses: TCD0_BITER_ELINKNO is 4000_8000h base + 101Eh offset = 4000_901Eh
TCD1_BITER_ELINKNO is 4000_8000h base + 103Eh offset = 4000_903Eh
TCD2_BITER_ELINKNO is 4000_8000h base + 105Eh offset = 4000_905Eh
TCD3_BITER_ELINKNO is 4000_8000h base + 107Eh offset = 4000_907Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	ELINK	BITER														
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

DMA_TCD*n*_BITER_ELINKNO field descriptions

Field	Description
15 ELINK	Enables channel-to-channel linking on minor loop complete As the channel completes the minor loop, this flag enables the linking to another channel, defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking is disabled, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field. Otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field is reloaded into the CITER field. 0 The channel-to-channel linking is disabled 1 The channel-to-channel linking is enabled
14–0 BITER	Starting Major Iteration Count As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be equal to the value in the CITER field. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. **NOTE:** When the software loads the TCD, this field must be set equal to the corresponding CITER field. Otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field is reloaded into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.

21.4 Functional description

21.4.1 eDMA basic data flow

The basic flow of a data transfer can be partitioned into three segments.

As shown in the following diagram, the first segment involves the channel activation:

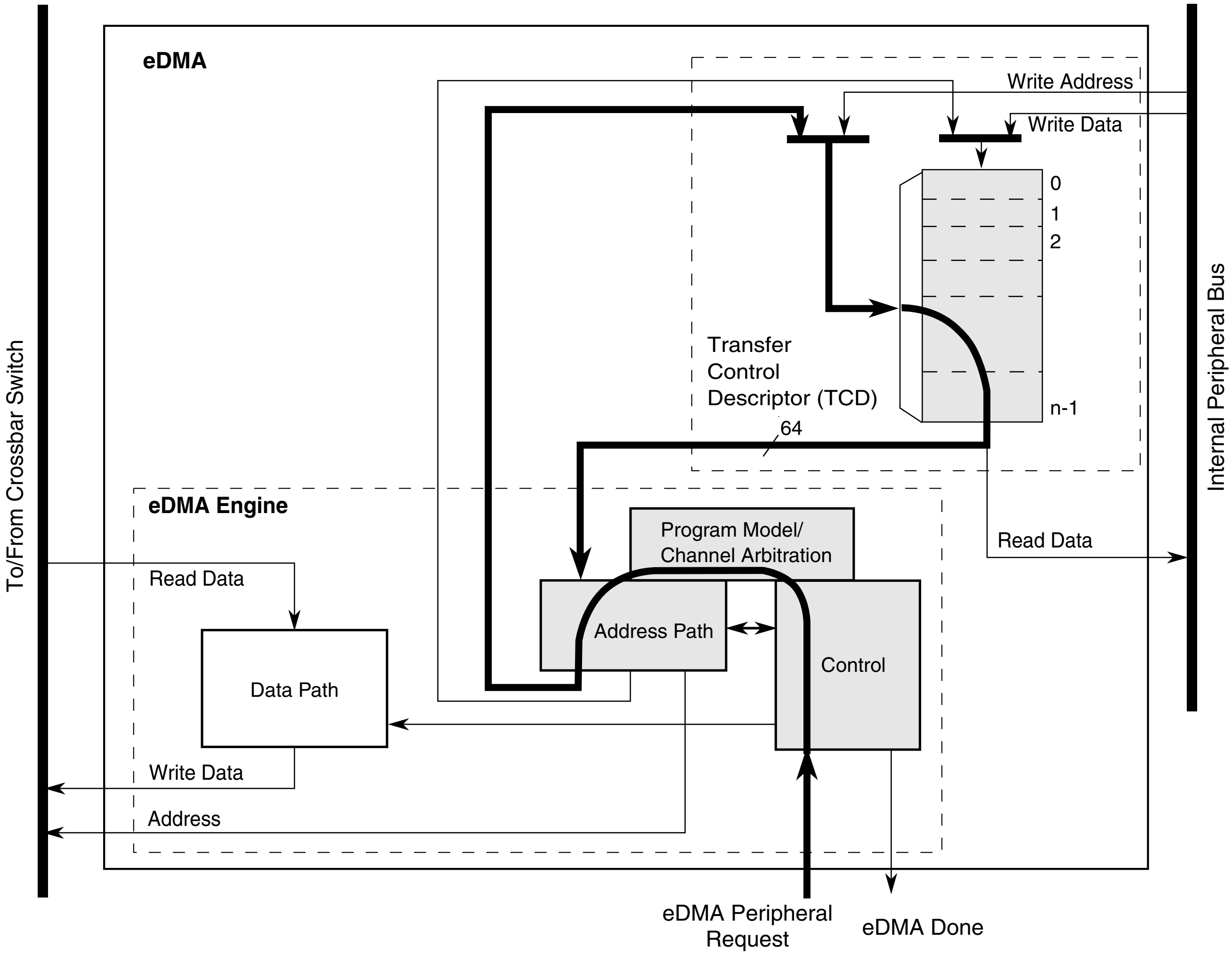

eDMA
Write Address
Write Data
0
1
2
n-1
Transfer
Control
Descriptor (TCD)
64
To/From Crossbar Switch
Internal Peripheral Bus
eDMA Engine
Read Data
Program Model/
Channel Arbitration
Read Data
Address Path
Control
Data Path
Write Data
Address
eDMA Peripheral
Request
eDMA Done

Figure 21-98. eDMA operation, part 1

This example uses the assertion of the eDMA peripheral request signal to request service for channel *n*. Channel activation via software and the TCD*n*_CSR[START] bit follows the same basic flow as peripheral requests. The eDMA request input signal is registered internally and then routed through the eDMA engine: first through the control module, then into the program model and channel arbitration. In the next cycle, the channel arbitration performs, using the fixed-priority or round-robin algorithm. After arbitration is complete, the activated channel number is sent through the address path and converted into the required address to access the local memory for TCD*n*. Next, the TCD memory is accessed and the required descriptor read from the local memory and loaded into the eDMA engine address path channel x or y registers. The TCD memory is 64 bits wide to minimize the time needed to fetch the activated channel descriptor and load it into the address path channel x or y registers.

The following diagram illustrates the second part of the basic data flow:

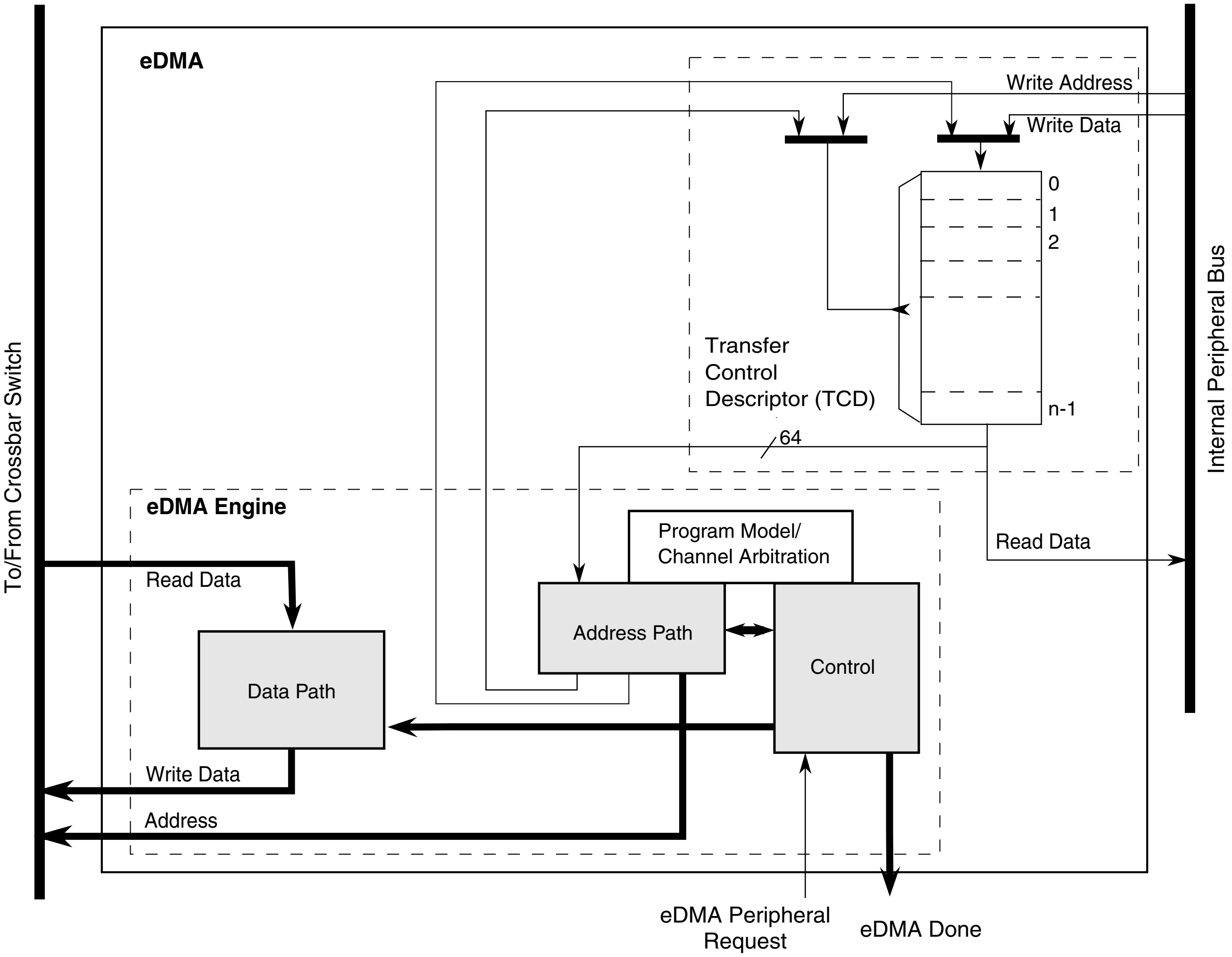

eDMA
Write Address
Write Data
0
1
2
n-1
Transfer Control Descriptor (TCD)
64
eDMA Engine
Program Model/ Channel Arbitration
Read Data
Address Path
Control
Data Path
Write Data
Address
Read Data
To/From Crossbar Switch
Internal Peripheral Bus
eDMA Peripheral Request
eDMA Done

Figure 21-99. eDMA operation, part 2

The modules associated with the data transfer (address path, data path, and control) sequence through the required source reads and destination writes to perform the actual data movement. The source reads are initiated and the fetched data is temporarily stored in the data path block until it is gated onto the internal bus during the destination write. This source read/destination write processing continues until the minor byte count has transferred.

After the minor byte count has moved, the final phase of the basic data flow is performed. In this segment, the address path logic performs the required updates to certain fields in the appropriate TCD, e.g., SADDR, DADDR, CITER. If the major iteration count is exhausted, additional operations are performed. These include the final address adjustments and reloading of the BITER field into the CITER. Assertion of an optional interrupt request also occurs at this time, as does a possible fetch of a new TCD from memory using the scatter/gather address pointer included in the descriptor (if scatter/gather is enabled). The updates to the TCD memory and the assertion of an interrupt request are shown in the following diagram.

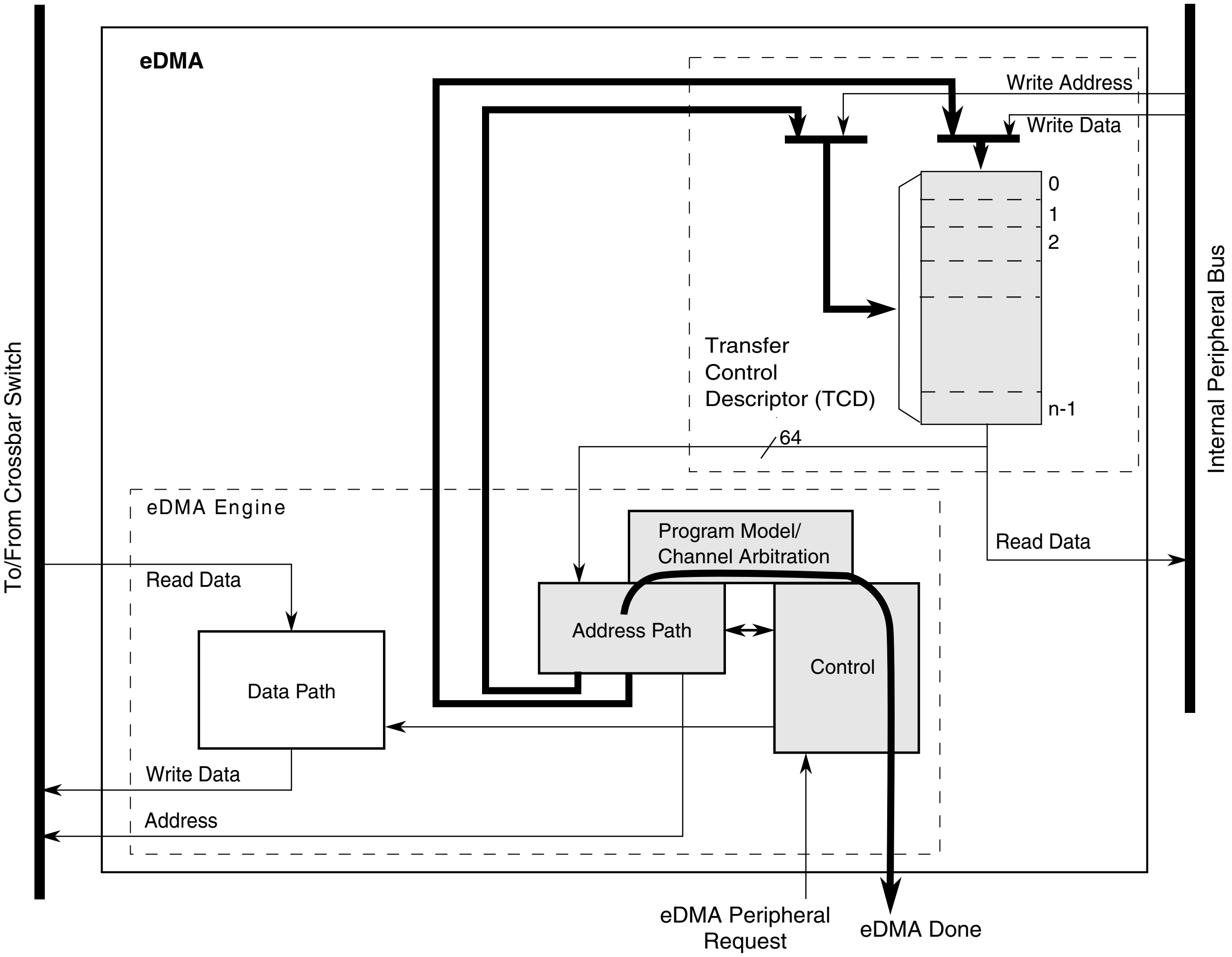

eDMA
Write Address
Write Data
0
1
2
n-1
Transfer Control Descriptor (TCD)
64
eDMA Engine
Program Model/ Channel Arbitration
Read Data
Address Path
Control
Data Path
Write Data
Address
Read Data
To/From Crossbar Switch
Internal Peripheral Bus
eDMA Peripheral Request
eDMA Done

Figure 21-100. eDMA operation, part 3

21.4.2 Error reporting and handling

Channel errors are reported in the ES register and can be caused by:

- A configuration error, which is an illegal setting in the transfer-control descriptor or an illegal priority register setting in Fixed-Arbitration mode, or
- An error termination to a bus master read or write cycle

A configuration error is reported when the starting source or destination address, source or destination offsets, minor loop byte count, or the transfer size represent an inconsistent state. Each of these possible causes are detailed below:

- The addresses and offsets must be aligned on 0-modulo-transfer-size boundaries.
- The minor loop byte count must be a multiple of the source and destination transfer sizes.

- All source reads and destination writes must be configured to the natural boundary of the programmed transfer size respectively.
- In fixed arbitration mode, a configuration error is caused by any two channel priorities being equal. All channel priority levels must be unique when fixed arbitration mode is enabled.
- If a scatter/gather operation is enabled upon channel completion, a configuration error is reported if the scatter/gather address (DLAST_SGA) is not aligned on a 32-byte boundary.
- If minor loop channel linking is enabled upon channel completion, a configuration error is reported when the link is attempted if the TCDn_CITER[E_LINK] bit does not equal the TCDn_BITER[E_LINK] bit.

If enabled, all configuration error conditions, except the scatter/gather and minor-loop link errors, report as the channel activates and asserts an error interrupt request. A scatter/gather configuration error is reported when the scatter/gather operation begins at major loop completion when properly enabled. A minor loop channel link configuration error is reported when the link operation is serviced at minor loop completion.

If a system bus read or write is terminated with an error, the data transfer is stopped and the appropriate bus error flag set. In this case, the state of the channel's transfer control descriptor is updated by the eDMA engine with the current source address, destination address, and current iteration count at the point of the fault. When a system-bus error occurs, the channel terminates after the read or write transaction, which is already pipelined after errant access, has completed. If a bus error occurs on the last read prior to beginning the write sequence, the write executes using the data captured during the bus error. If a bus error occurs on the last write prior to switching to the next read sequence, the read sequence executes before the channel terminates due to the destination bus error.

A transfer may be cancelled by software with the CR[CX] bit. When a cancel transfer request is recognized, the DMA engine stops processing the channel. The current read-write sequence is allowed to finish. If the cancel occurs on the last read-write sequence of a major or minor loop, the cancel request is discarded and the channel retires normally.

The error cancel transfer is the same as a cancel transfer except the ES register is updated with the cancelled channel number and ECX is set. The TCD of a cancelled channel contains the source and destination addresses of the last transfer saved in the TCD. If the channel needs to be restarted, you must re-initialize the TCD because the aforementioned fields no longer represent the original parameters. When a transfer is cancelled by the error cancel transfer mechanism, the channel number is loaded into DMA_ES[ERRCHN] and ECX and VLD are set. In addition, an error interrupt may be generated if enabled.

The occurrence of any error causes the eDMA engine to stop the active channel immediately, and the appropriate channel bit in the eDMA error register is asserted. At the same time, the details of the error condition are loaded into the ES register. The major loop complete indicators, setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request, are not affected when an error is detected. After the error status has been updated, the eDMA engine continues operating by servicing the next appropriate channel. A channel that experiences an error condition is not automatically disabled. If a channel is terminated by an error and then issues another service request before the error is fixed, that channel executes and terminates with the same error condition.

21.4.3 Channel preemption

Channel preemption is enabled on a per-channel basis by setting the DCHPRIn[ECP] bit. Channel preemption allows the executing channel's data transfers to temporarily suspend in favor of starting a higher priority channel. After the preempting channel has completed all its minor loop data transfers, the preempted channel is restored and resumes execution. After the restored channel completes one read/write sequence, it is again eligible for preemption. If any higher priority channel is requesting service, the restored channel is suspended and the higher priority channel is serviced. Nested preemption, that is, attempting to preempt a preempting channel, is not supported. After a preempting channel begins execution, it cannot be preempted. Preemption is available only when fixed arbitration is selected.

A channel's ability to preempt another channel can be disabled by setting DCHPRIn[DPA]. When a channel's preempt ability is disabled, that channel cannot suspend a lower priority channel's data transfer, regardless of the lower priority channel's ECP setting. This allows for a pool of low priority, large data-moving channels to be defined. These low priority channels can be configured to not preempt each other, thus preventing a low priority channel from consuming the preempt slot normally available to a true, high priority channel.

21.4.4 Performance

This section addresses the performance of the eDMA module, focusing on two separate metrics:

- In the traditional data movement context, performance is best expressed as the peak data transfer rates achieved using the eDMA. In most implementations, this transfer rate is limited by the speed of the source and destination address spaces.
- In a second context where device-paced movement of single data values to/from peripherals is dominant, a measure of the requests that can be serviced in a fixed time is a more relevant metric. In this environment, the speed of the source and destination

address spaces remains important. However, the microarchitecture of the eDMA also factors significantly into the resulting metric.

21.4.4.1 Peak transfer rates

The peak transfer rates for several different source and destination transfers are shown in the following tables. These tables assume:

- Internal SRAM can be accessed with zero wait-states when viewed from the system bus data phase
- All internal peripheral bus reads require two wait-states, and internal peripheral bus writes three wait-states, when viewed from the system bus data phase
- All internal peripheral bus accesses are 32-bits in size

This table presents a peak transfer rate comparison.

Table 21-101. eDMA peak transfer rates (Mbytes/sec)

System Speed, Width	Internal SRAM-to- Internal SRAM	32b internal peripheral bus-to- Internal SRAM	Internal SRAM-to- 32b internal peripheral bus
66.7 MHz, 32b	133.3	66.7	53.3
83.3 MHz, 32b	166.7	83.3	66.7
100.0 MHz, 32b	200.0	100.0	80.0
133.3 MHz, 32b	266.7	133.3	106.7
150.0 MHz, 32b	300.0	150.0	120.0

Internal-SRAM-to-internal-SRAM transfers occur at the core's datapath width. For all transfers involving the internal peripheral bus, 32-bit transfer sizes are used. In all cases, the transfer rate includes the time to read the source plus the time to write the destination.

21.4.4.2 Peak request rates

The second performance metric is a measure of the number of DMA requests that can be serviced in a given amount of time. For this metric, assume that the peripheral request causes the channel to move a single internal peripheral bus-mapped operand to/from internal SRAM. The same timing assumptions used in the previous example apply to this calculation. In particular, this metric also reflects the time required to activate the channel.

The eDMA design supports the following hardware service request sequence. Note that the exact timing from Cycle 7 is a function of the response times for the channel's read and write accesses. In the case of an internal peripheral bus read and internal SRAM write, the combined data phase time is 4 cycles. For an SRAM read and internal peripheral bus write, it is 5 cycles.

Table 21-102. Hardware service request process

Cycle		Description
With internal peripheral bus read and internal SRAM write	**With SRAM read and internal peripheral bus write**	
1		eDMA peripheral request is asserted.
2		The eDMA peripheral request is registered locally in the eDMA module and qualified. TCD*n*_CSR[START] bit initiated requests start at this point with the registering of the user write to TCD*n* word 7.
3		Channel arbitration begins.
4		Channel arbitration completes. The transfer control descriptor local memory read is initiated.
5–6		The first two parts of the activated channel's TCD is read from the local memory. The memory width to the eDMA engine is 64 bits, so the entire descriptor can be accessed in four cycles
7		The first system bus read cycle is initiated, as the third part of the channel's TCD is read from the local memory. Depending on the state of the crossbar switch, arbitration at the system bus may insert an additional cycle of delay here.
8–11	8–12	The last part of the TCD is read in. This cycle represents the first data phase for the read, and the address phase for the destination write.
12	13	This cycle represents the data phase of the last destination write.
13	14	The eDMA engine completes the execution of the inner minor loop and prepares to write back the required TCD*n* fields into the local memory. The TCD*n* word 7 is read and checked for channel linking or scatter/gather requests.
14	15	The appropriate fields in the first part of the TCD*n* are written back into the local memory.
15	16	The fields in the second part of the TCD*n* are written back into the local memory. This cycle coincides with the next channel arbitration cycle start.
16	17	The next channel to be activated performs the read of the first part of its TCD from the local memory. This is equivalent to Cycle 4 for the first channel's service request.

Assuming zero wait states on the system bus, DMA requests can be processed every 9 cycles. Assuming an average of the access times associated with internal peripheral bus-to-SRAM (4 cycles) and SRAM-to-internal peripheral bus (5 cycles), DMA requests can be processed every 11.5 cycles (4 + (4+5)/2 + 3). This is the time from Cycle 4 to Cycle x +5. The resulting peak request rate, as a function of the system frequency, is shown in the following table.

Table 21-103. eDMA peak request rate (MReq/sec)

System frequency (MHz)	Request rate with zero wait states	Request rate with wait states
66.6	7.4	5.8
83.3	9.2	7.2
100.0	11.1	8.7
133.3	14.8	11.6
150.0	16.6	13.0

A general formula to compute the peak request rate with overlapping requests is:

PEAKreq = freq / [entry + (1 + read_ws) + (1 + write_ws) + exit]

where:

Table 21-104. Peak request formula operands

Operand	Description
PEAKreq	Peak request rate
freq	System frequency
entry	Channel startup (4 cycles)
read_ws	Wait states seen during the system bus read data phase
write_ws	Wait states seen during the system bus write data phase
exit	Channel shutdown (3 cycles)

21.4.4.3 eDMA performance example

Consider a system with the following characteristics:

- Internal SRAM can be accessed with one wait-state when viewed from the system bus data phase
- All internal peripheral bus reads require two wait-states, and internal peripheral bus writes three wait-states viewed from the system bus data phase
- System operates at 150 MHz

For an SRAM to internal peripheral bus transfer, PEAKreq = 150 MHz / [4 + (1 + 1) + (1 + 3) + 3] cycles = 11.5 Mreq/sec

For an internal peripheral bus to SRAM transfer,

PEAKreq = 150 MHz / [4 + (1 + 2) + (1 + 1) + 3] cycles = 12.5 Mreq/sec

Assuming an even distribution of the two transfer types, the average peak request rate would be:

PEAKreq = (11.5 Mreq/sec + 12.5 Mreq/sec) / 2 = 12.0 Mreq/sec

The minimum number of cycles to perform a single read/write, zero wait states on the system bus, from a cold start where no channel is executing and eDMA is idle are:

- 11 cycles for a software, that is, a TCD*n*_CSR[START] bit, request
- 12 cycles for a hardware, that is, an eDMA peripheral request signal, request

Two cycles account for the arbitration pipeline and one extra cycle on the hardware request resulting from the internal registering of the eDMA peripheral request signals. For the peak request rate calculations above, the arbitration and request registering is absorbed in or overlaps the previous executing channel.

Note

When channel linking or scatter/gather is enabled, a two cycle delay is imposed on the next channel selection and startup. This allows the link channel or the scatter/gather channel to be eligible and considered in the arbitration pool for next channel selection.

21.5 Initialization/application information

The following sections discuss initialization of the eDMA and programming considerations.

21.5.1 eDMA initialization

To initialize the eDMA:

1. Write to the CR if a configuration other than the default is desired.
2. Write the channel priority levels to the DCHPRI*n* registers if a configuration other than the default is desired.

3. Enable error interrupts in the EEI register if so desired.
4. Write the 32-byte TCD for each channel that may request service.
5. Enable any hardware service requests via the ERQ register.
6. Request channel service via either:
    - Software: setting the TCD*n*_CSR[START]
    - Hardware: slave device asserting its eDMA peripheral request signal

After any channel requests service, a channel is selected for execution based on the arbitration and priority levels written into the programmer's model. The eDMA engine reads the entire TCD, including the TCD control and status fields, as shown in the following table, for the selected channel into its internal address path module.

As the TCD is read, the first transfer is initiated on the internal bus, unless a configuration error is detected. Transfers from the source, as defined by TCD*n*_SADDR, to the destination, as defined by TCD*n*_DADDR, continue until the number of bytes specified by TCD*n*_NBYTES are transferred.

When the transfer is complete, the eDMA engine's local TCD*n*_SADDR, TCD*n*_DADDR, and TCD*n*_CITER are written back to the main TCD memory and any minor loop channel linking is performed, if enabled. If the major loop is exhausted, further post processing executes, such as interrupts, major loop channel linking, and scatter/gather operations, if enabled.

Table 21-105. TCD Control and Status fields

TCD*n*_CSR field name	Description
START	Control bit to start channel explicitly when using a software initiated DMA service (Automatically cleared by hardware)
ACTIVE	Status bit indicating the channel is currently in execution
DONE	Status bit indicating major loop completion (cleared by software when using a software initiated DMA service)
D_REQ	Control bit to disable DMA request at end of major loop completion when using a hardware initiated DMA service
BWC	Control bits for throttling bandwidth control of a channel
E_SG	Control bit to enable scatter-gather feature
INT_HALF	Control bit to enable interrupt when major loop is half complete
INT_MAJ	Control bit to enable interrupt when major loop completes

The following figure shows how each DMA request initiates one minor-loop transfer, or iteration, without CPU intervention. DMA arbitration can occur after each minor loop, and one level of minor loop DMA preemption is allowed. The number of minor loops in a major loop is specified by the beginning iteration count (BITER).

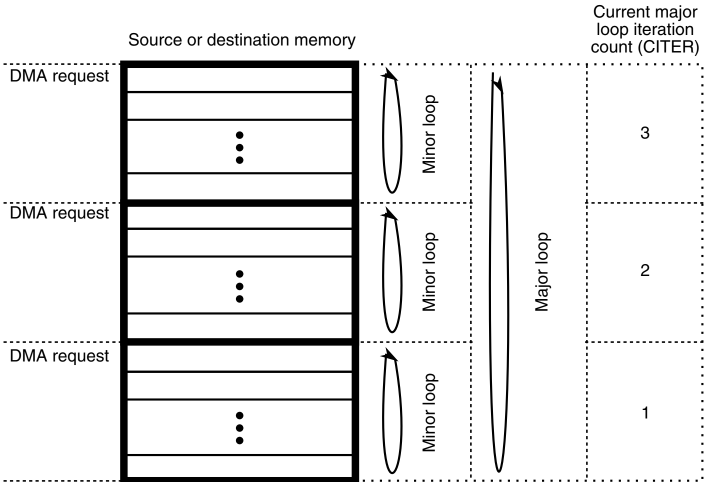

Source or destination memory
Current major loop iteration count (CITER)
DMA request
DMA request
DMA request
Minor loop
Minor loop
Minor loop
Major loop
3
2
1

Figure 21-101. Example of multiple loop iterations

The following figure lists the memory array terms and how the TCD settings interrelate.

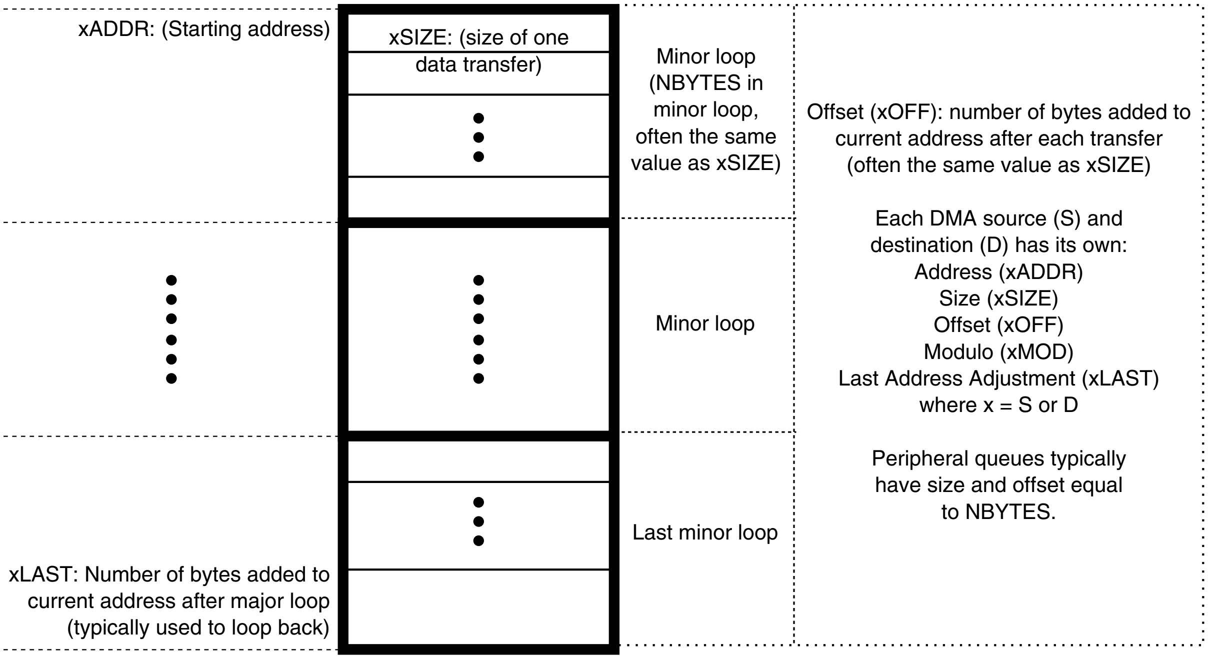

xADDR: (Starting address)
xSIZE: (size of one data transfer)
Minor loop (NBYTES in minor loop, often the same value as xSIZE)
Offset (xOFF): number of bytes added to current address after each transfer (often the same value as xSIZE)
Each DMA source (S) and destination (D) has its own:
Address (xADDR)
Size (xSIZE)
Offset (xOFF)
Modulo (xMOD)
Last Address Adjustment (xLAST)
where x = S or D
Peripheral queues typically have size and offset equal to NBYTES.
Minor loop
Last minor loop
xLAST: Number of bytes added to current address after major loop (typically used to loop back)

Figure 21-102. Memory array terms

21.5.2 Programming errors

The eDMA performs various tests on the transfer control descriptor to verify consistency in the descriptor data. Most programming errors are reported on a per channel basis with the exception of channel priority error (ES[CPE]).

For all error types other than channel priority error, the channel number causing the error is recorded in the ES register. If the error source is not removed before the next activation of the problem channel, the error is detected and recorded again.

If priority levels are not unique, when any channel requests service, a channel priority error is reported. The highest channel priority with an active request is selected, but the lowest numbered channel with that priority is selected by arbitration and executed by the eDMA engine. The hardware service request handshake signals, error interrupts, and error reporting is associated with the selected channel.

21.5.3 Arbitration mode considerations

21.5.3.1 Fixed channel arbitration

In this mode, the channel service request from the highest priority channel is selected to execute.

21.5.3.2 Round-robin channel arbitration

Channels are serviced starting with the highest channel number and rotating through to the lowest channel number without regard to the channel priority levels.

21.5.4 Performing DMA transfers (examples)

21.5.4.1 Single request

To perform a simple transfer of n bytes of data with one activation, set the major loop to one (TCD*n*_CITER = TCD*n*_BITER = 1). The data transfer begins after the channel service request is acknowledged and the channel is selected to execute. After the transfer is complete, the TCD*n*_CSR[DONE] bit is set and an interrupt generates if properly enabled.

For example, the following TCD entry is configured to transfer 16 bytes of data. The eDMA is programmed for one iteration of the major loop transferring 16 bytes per iteration. The source memory has a byte wide memory port located at 0x1000. The destination memory has a 32-bit port located at 0x2000. The address offsets are programmed in increments to match the transfer size: one byte for the source and four bytes for the destination. The final source and destination addresses are adjusted to return to their beginning values.

```
TCDn_CITER = TCDn_BITER = 1
TCDn_NBYTES = 16
TCDn_SADDR = 0x1000
TCDn_SOFF = 1
TCDn_ATTR[SSIZE] = 0
TCDn_SLAST = -16
TCDn_DADDR = 0x2000
TCDn_DOFF = 4
TCDn_ATTR[DSIZE] = 2
TCDn_DLAST_SGA= -16
TCDn_CSR[INT_MAJ] = 1
TCDn_CSR[START] = 1 (Should be written last after all other fields have been initialized)
All other TCDn fields = 0
```

This generates the following event sequence:

1. User write to the TCD*n*_CSR[START] bit requests channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes: TCD*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, TCD*n*_CSR[ACTIVE] = 1.
4. eDMA engine reads: channel TCD data from local memory to internal register file.
5. The source-to-destination transfers are executed as follows:
   a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
   b. Write 32-bits to location 0x2000 → first iteration of the minor loop.
   c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
   d. Write 32-bits to location 0x2004 → second iteration of the minor loop.
   e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
   f. Write 32-bits to location 0x2008 → third iteration of the minor loop.
   g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
   h. Write 32-bits to location 0x200C → last iteration of the minor loop → major loop complete.

6. The eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 1 (TCD*n*_BITER).
7. The eDMA engine writes: TCD*n*_CSR[ACTIVE] = 0, TCD*n*_CSR[DONE] = 1, INT[*n*] = 1.
8. The channel retires and the eDMA goes idle or services the next channel.

21.5.4.2 Multiple requests

The following example transfers 32 bytes via two hardware requests, but is otherwise the same as the previous example. The only fields that change are the major loop iteration count and the final address offsets. The eDMA is programmed for two iterations of the major loop transferring 16 bytes per iteration. After the channel's hardware requests are enabled in the ERQ register, the slave device initiates channel service requests.

```
TCDn_CITER = TCDn_BITER = 2
TCDn_SLAST = -32
TCDn_DLAST_SGA = -32
```

This would generate the following sequence of events:

1. First hardware, that is, eDMA peripheral, request for channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes: TCD*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, TCD*n*_CSR[ACTIVE] = 1.
4. eDMA engine reads: channel TCD*n* data from local memory to internal register file.
5. The source to destination transfers are executed as follows:
    a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
    b. Write 32-bits to location 0x2000 → first iteration of the minor loop.
    c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
    d. Write 32-bits to location 0x2004 → second iteration of the minor loop.
    e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
    f. Write 32-bits to location 0x2008 → third iteration of the minor loop.

   g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
   h. Write 32-bits to location 0x200C → last iteration of the minor loop.
6. eDMA engine writes: TCD*n*_SADDR = 0x1010, TCD*n*_DADDR = 0x2010, TCD*n*_CITER = 1.
7. eDMA engine writes: TCD*n*_CSR[ACTIVE] = 0.
8. The channel retires → one iteration of the major loop. The eDMA goes idle or services the next channel.
9. Second hardware, that is, eDMA peripheral, requests channel service.
10. The channel is selected by arbitration for servicing.
11. eDMA engine writes: TCD*n*_CSR[DONE] = 0, TCD*n*_CSR[START] = 0, TCD*n*_CSR[ACTIVE] = 1.
12. eDMA engine reads: channel TCD data from local memory to internal register file.
13. The source to destination transfers are executed as follows:
    a. Read byte from location 0x1010, read byte from location 0x1011, read byte from 0x1012, read byte from 0x1013.
    b. Write 32-bits to location 0x2010 → first iteration of the minor loop.
    c. Read byte from location 0x1014, read byte from location 0x1015, read byte from 0x1016, read byte from 0x1017.
    d. Write 32-bits to location 0x2014 → second iteration of the minor loop.
    e. Read byte from location 0x1018, read byte from location 0x1019, read byte from 0x101A, read byte from 0x101B.
    f. Write 32-bits to location 0x2018 → third iteration of the minor loop.
    g. Read byte from location 0x101C, read byte from location 0x101D, read byte from 0x101E, read byte from 0x101F.
    h. Write 32-bits to location 0x201C → last iteration of the minor loop → major loop complete.
14. eDMA engine writes: TCD*n*_SADDR = 0x1000, TCD*n*_DADDR = 0x2000, TCD*n*_CITER = 2 (TCD*n*_BITER).
15. eDMA engine writes: TCD*n*_CSR[ACTIVE] = 0, TCD*n*_CSR[DONE] = 1, INT[n] = 1.

16. The channel retires → major loop complete. The eDMA goes idle or services the next channel.

21.5.4.3 Using the modulo feature

The modulo feature of the eDMA provides the ability to implement a circular data queue in which the size of the queue is a power of 2. MOD is a 5-bit field for the source and destination in the TCD, and it specifies which lower address bits increment from their original value after the address+offset calculation. All upper address bits remain the same as in the original value. A setting of 0 for this field disables the modulo feature.

The following table shows how the transfer addresses are specified based on the setting of the MOD field. Here a circular buffer is created where the address wraps to the original value while the 28 upper address bits (0x1234567*x*) retain their original value. In this example the source address is set to 0x12345670, the offset is set to 4 bytes and the MOD field is set to 4, allowing for a 2^4 byte (16-byte) size queue.

Table 21-106. Modulo example

Transfer Number	Address
1	0x12345670
2	0x12345674
3	0x12345678
4	0x1234567C
5	0x12345670
6	0x12345674

21.5.5 Monitoring transfer descriptor status

21.5.5.1 Testing for minor loop completion

There are two methods to test for minor loop completion when using software initiated service requests. The first is to read the TCD*n*_CITER field and test for a change. Another method may be extracted from the sequence shown below. The second method is to test the TCD*n*_CSR[START] bit and the TCD*n*_CSR[ACTIVE] bit. The minor-loop-complete condition is indicated by both bits reading zero after the TCD*n*_CSR[START] was set. Polling the TCD*n*_CSR[ACTIVE] bit may be inconclusive, because the active status may be missed if the channel execution is short in duration.

The TCD status bits execute the following sequence for a software activated channel:

Stage	TCD*n*_CSR bits			State
	START	ACTIVE	DONE	
1	1	0	0	Channel service request via software
2	0	1	0	Channel is executing
3a	0	0	0	Channel has completed the minor loop and is idle
3b	0	0	1	Channel has completed the major loop and is idle

The best method to test for minor-loop completion when using hardware, that is, peripheral, initiated service requests is to read the TCD*n*_CITER field and test for a change. The hardware request and acknowledge handshake signals are not visible in the programmer's model.

The TCD status bits execute the following sequence for a hardware-activated channel:

Stage	TCD*n*_CSR bits			State
	START	ACTIVE	DONE	
1	0	0	0	Channel service request via hardware (peripheral request asserted)
2	0	1	0	Channel is executing
3a	0	0	0	Channel has completed the minor loop and is idle
3b	0	0	1	Channel has completed the major loop and is idle

For both activation types, the major-loop-complete status is explicitly indicated via the TCD*n*_CSR[DONE] bit.

The TCD*n*_CSR[START] bit is cleared automatically when the channel begins execution regardless of how the channel activates.

21.5.5.2 Reading the transfer descriptors of active channels

The eDMA reads back the true TCD*n*_SADDR, TCD*n*_DADDR, and TCD*n*_NBYTES values if read while a channel executes. The true values of the SADDR, DADDR, and NBYTES are the values the eDMA engine currently uses in its internal register file and not the values in the TCD local memory for that channel. The addresses, SADDR and DADDR, and NBYTES, which decrement to zero as the transfer progresses, can give an indication of the progress of the transfer. All other values are read back from the TCD local memory.

21.5.5.3 Checking channel preemption status

Preemption is available only when fixed arbitration is selected as the channel arbitration mode. A preemptive situation is one in which a preempt-enabled channel runs and a higher priority request becomes active. When the eDMA engine is not operating in fixed channel arbitration mode, the determination of the actively running relative priority outstanding requests become undefined. Channel priorities are treated as equal, that is, constantly rotating, when Round-Robin Arbitration mode is selected.

The TCD*n*_CSR[ACTIVE] bit for the preempted channel remains asserted throughout the preemption. The preempted channel is temporarily suspended while the preempting channel executes one major loop iteration. If two TCD*n*_CSR[ACTIVE] bits are set simultaneously in the global TCD map, a higher priority channel is actively preempting a lower priority channel.

21.5.6 Channel Linking

Channel linking (or chaining) is a mechanism where one channel sets the TCD*n*_CSR[START] bit of another channel (or itself), therefore initiating a service request for that channel. When properly enabled, the EDMA engine automatically performs this operation at the major or minor loop completion.

The minor loop channel linking occurs at the completion of the minor loop (or one iteration of the major loop). The TCD*n*_CITER[E_LINK] field determines whether a minor loop link is requested. When enabled, the channel link is made after each iteration of the major loop except for the last. When the major loop is exhausted, only the major loop channel link fields are used to determine if a channel link should be made. For example, the initial fields of:

```
TCDn_CITER[E_LINK] = 1
TCDn_CITER[LINKCH] = 0xC
TCDn_CITER[CITER] value = 0x4
TCDn_CSR[MAJOR_E_LINK] = 1
TCDn_CSR[MAJOR_LINKCH] = 0x7
```

executes as:

1. Minor loop done → set TCD12_CSR[START] bit
2. Minor loop done → set TCD12_CSR[START] bit
3. Minor loop done → set TCD12_CSR[START] bit
4. Minor loop done, major loop done→ set TCD7_CSR[START] bit

When minor loop linking is enabled (TCD*n*_CITER[E_LINK] = 1), the TCD*n*_CITER[CITER] field uses a nine bit vector to form the current iteration count. When minor loop linking is disabled (TCD*n*_CITER[E_LINK] = 0), the TCD*n*_CITER[CITER] field uses a 15-bit vector to form the current iteration count. The bits associated with the TCD*n*_CITER[LINKCH] field are concatenated onto the CITER value to increase the range of the CITER.

Note

> The TCD*n*_CITER[E_LINK] bit and the TCD*n*_BITER[E_LINK] bit must equal or a configuration error is reported. The CITER and BITER vector widths must be equal to calculate the major loop, half-way done interrupt point.

The following table summarizes how a DMA channel can link to another DMA channel, i.e, use another channel's TCD, at the end of a loop.

Table 21-107. Channel Linking Parameters

Desired Link Behavior	TCD Control Field Name	Description
Link at end of Minor Loop	CITER[E_LINK]	Enable channel-to-channel linking on minor loop completion (current iteration)
	CITER[LINKCH]	Link channel number when linking at end of minor loop (current iteration)
Link at end of Major Loop	CSR[MAJOR_E_LINK]	Enable channel-to-channel linking on major loop completion
	CSR[MAJOR_LINKCH]	Link channel number when linking at end of major loop

21.5.7 Dynamic programming

21.5.7.1 Dynamically changing the channel priority

The following two options are recommended for dynamically changing channel priority levels:

1. Switch to Round-Robin Channel Arbitration mode, change the channel priorities, then switch back to Fixed Arbitration mode,
2. Disable all the channels, change the channel priorities, then enable the appropriate channels.

21.5.7.2 Dynamic channel linking

Dynamic channel linking is the process of setting the TCD.major.e_link bit during channel execution. This bit is read from the TCD local memory at the end of channel execution, thus allowing the user to enable the feature during channel execution.

Because the user is allowed to change the configuration during execution, a coherency model is needed. Consider the scenario where the user attempts to execute a dynamic channel link by enabling the TCD.major.e_link bit at the same time the eDMA engine is retiring the channel. The TCD.major.e_link would be set in the programmer's model, but it would be unclear whether the actual link was made before the channel retired.

The following coherency model is recommended when executing a dynamic channel link request.

Step	Action
1	Write 1b to the TCD.major.e_link bit.
2	Read back the TCD.major.e_link bit.
3	Test the TCD.major.e_link request status: • If TCD.major.e_link = 1b, the dynamic link attempt was successful. • If TCD.major.e_link = 0b, the attempted dynamic link did not succeed (the channel was already retiring).

For this request, the TCD local memory controller forces the TCD.major.e_link bit to zero on any writes to a channel's TCD.word7 after that channel's TCD.done bit is set, indicating the major loop is complete.

NOTE

The user must clear the TCD.done bit before writing the TCD.major.e_link bit. The TCD.done bit is cleared automatically by the eDMA engine after a channel begins execution.

21.5.7.3 Dynamic scatter/gather

Dynamic scatter/gather is the process of setting the TCD.e_sg bit during channel execution. This bit is read from the TCD local memory at the end of channel execution, thus allowing the user to enable the feature during channel execution.

Because the user is allowed to change the configuration during execution, a coherency model is needed. Consider the scenario where the user attempts to execute a dynamic scatter/gather operation by enabling the TCD.e_sg bit at the same time the eDMA engine is retiring the channel. The TCD.e_sg would be set in the programmer's model, but it would be unclear whether the actual scatter/gather request was honored before the channel retired.

Two methods for this coherency model are shown in the following subsections. Method 1 has the advantage of reading the major.linkch field and the e_sg bit with a single read. For both dynamic channel linking and scatter/gather requests, the TCD local memory controller forces the TCD.major.e_link and TCD.e_sg bits to zero on any writes to a channel's TCD.word7 if that channel's TCD.done bit is set indicating the major loop is complete.

NOTE

> The user must clear the TCD.done bit before writing the TCD.major.e_link or TCD.e_sg bits. The TCD.done bit is cleared automatically by the eDMA engine after a channel begins execution.

21.5.7.3.1 Method 1 (channel not using major loop channel linking)

For a channel not using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request.

When the TCD.major.e_link bit is zero, the TCD.major.linkch field is not used by the eDMA. In this case, the TCD.major.linkch bits may be used for other purposes. This method uses the TCD.major.linkch field as a TCD indentification (ID).

1. When the descriptors are built, write a unique TCD ID in the TCD.major.linkch field for each TCD associated with a channel using dynamic scatter/gather.
2. Write 1b to theTCD.d_req bit.

   Should a dynamic scatter/gather attempt fail, setting the d_req bit will prevent a future hardware activation of this channel. This stops the channel from executing with a destination address (daddr) that was calculated using a scatter/gather address (written in the next step) instead of a dlast final offest value.

3. Write theTCD.dlast_sga field with the scatter/gather address.
4. Write 1b to the TCD.e_sg bit.
5. Read back the 16 bit TCD control/status field.
6. Test the TCD.e_sg request status and TCD.major.linkch value:

   If e_sg = 1b, the dynamic link attempt was successful.

If e_sg = 0b and the major.linkch (ID) did not change, the attempted dynamic link did not succeed (the channel was already retiring).

If e_sg = 0b and the major.linkch (ID) changed, the dynamic link attempt was successful (the new TCD's e_sg value cleared the e_sg bit).

21.5.7.3.2 Method 2 (channel using major loop channel linking)

For a channel using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request. This method uses the TCD.dlast_sga field as a TCD indentification (ID).

1. Write 1b to theTCD.d_req bit.

   Should a dynamic scatter/gather attempt fail, setting the d_req bit will prevent a future hardware activation of this channel. This stops the channel from executing with a destination address (daddr) that was calculated using a scatter/gather address (written in the next step) instead of a dlast final offest value.

2. Write theTCD.dlast_sga field with the scatter/gather address.
3. Write 1b to the TCD.e_sg bit.
4. Read back the TCD.e_sg bit.
5. Test the TCD.e_sg request status:

   If e_sg = 1b, the dynamic link attempt was successful.

   If e_sg = 0b, read the 32 bit TCD dlast_sga field.

   If e_sg = 0b and the dlast_sga did not change, the attempted dynamic link did not succeed (the channel was already retiring).

   If e_sg = 0b and the dlast_sga changed, the dynamic link attempt was successful (the new TCD's e_sg value cleared the e_sg bit).

Chapter 22
External Watchdog Monitor (EWM)

22.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The watchdog is generally used to monitor the flow and execution of embedded software within an MCU. The watchdog consists of a counter that if allowed to overflow, forces an internal reset (asynchronous) to all on-chip peripherals and optionally assert the $\overline{\text{RESET}}$ pin to reset external devices/circuits. The overflow of the watchdog counter must not occur if the software code works well and services the watchdog to re-start the actual counter.

For safety, a redundant watchdog system, External Watchdog Monitor (EWM), is designed to monitor external circuits, as well as the MCU software flow. This provides a back-up mechanism to the internal watchdog that resets the MCU's CPU and peripherals.

The EWM differs from the internal watchdog in that it does not reset the MCU's CPU and peripherals. The EWM if allowed to time-out, provides an independent $\overline{\text{EWM\_out}}$ pin that when asserted resets or places an external circuit into a safe mode. The CPU resets the EWM counter that is logically ANDed with an external digital input pin. This pin allows an external circuit to influence the reset_out signal.

22.1.1 Features

Features of EWM module include:

- Independent LPO clock source
- Programmable time-out period specified in terms of number of EWM LPO clock cycles.

- Windowed refresh option
    - Provides robust check that program flow is faster than expected.
    - Programmable window.
    - Refresh outside window leads to assertion of $\overline{\text{EWM\_out}}$.
- Robust refresh mechanism
    - Write values of 0xB4 and 0x2C to EWM Refresh Register within 15 (*EWM_service_time*) peripheral bus clock cycles.
- One output port, $\overline{\text{EWM\_out}}$, when asserted is used to reset or place the external circuit into safe mode.
- One Input port, EWM_in, allows an external circuit to control the $\overline{\text{EWM\_out}}$ signal.

22.1.2 Modes of Operation

This section describes the module's operating modes.

22.1.2.1 Stop Mode

When the EWM is in stop mode, the CPU services to the EWM cannot occur. On entry to stop mode, the EWM’s counter freezes.

There are two possible ways to exit from Stop mode:

- On exit from stop mode through a reset, the EWM remains disabled.
- On exit from stop mode by an interrupt, the EWM is re-enabled, and the counter continues to be clocked from the same value prior to entry to stop mode.

Note the following if the EWM enters the stop mode during CPU service mechanism: At the exit from stop mode by an interrupt, refresh mechanism state machine starts from the previous state which means, if first service command is written correctly and EWM enters the stop mode immediately, the next command has to be written within the next 15 (*EWM_service_time*) peripheral bus clocks after exiting from stop mode. User must mask all interrupts prior to executing EWM service instructions.

22.1.2.2 Wait Mode

The EWM module treats the stop and wait modes as the same. EWM functionality remains the same in both of these modes.

22.1.2.3 Debug Mode

Entry to debug mode has no effect on the EWM.

- If the EWM is enabled prior to entry of debug mode, it remains enabled.
- If the EWM is disabled prior to entry of debug mode, it remains disabled.

22.1.3 Block Diagram

This figure shows the EWM block diagram.

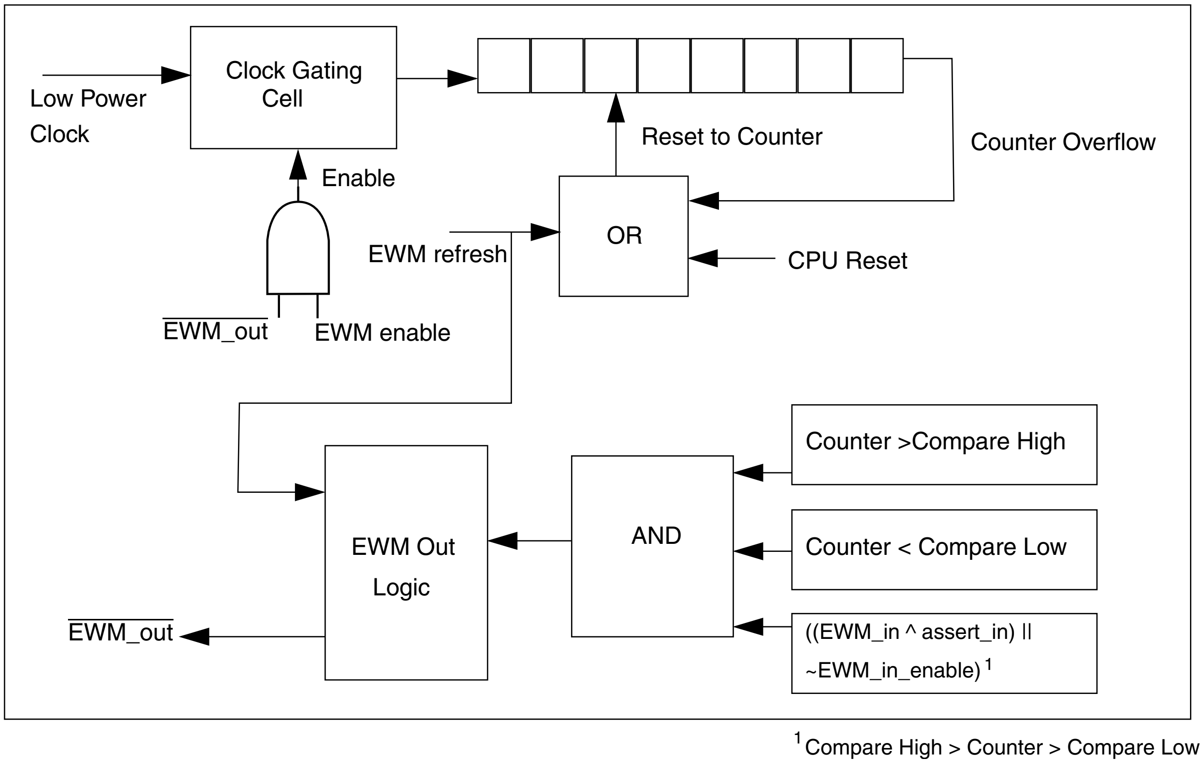

Low Power
Clock
Clock Gating
Cell
Enable
Reset to Counter
Counter Overflow
EWM refresh
OR
CPU Reset
$\overline{\text{EWM\_out}}$
EWM enable
Counter >Compare High
Counter < Compare Low
((EWM_in ^ assert_in) ||
~EWM_in_enable) 1
AND
EWM Out
Logic
$\overline{\text{EWM\_out}}$
1 Compare High > Counter > Compare Low

Figure 22-1. EWM Block Diagram

22.2 EWM Signal Descriptions

The EWM has two external signals, as shown in the following table.

Table 22-1. EWM Signal Descriptions

Signal	Description	I/O
EWM_in	EWM input for safety status of external safety circuits. The polarity of EWM_in is programmable using the EWM_CTRL[ASSIN] bit. The default polarity is active-low.	I
$\overline{\text{EWM\_out}}$	EWM reset out signal	O

22.3 Memory Map/Register Definition

This section contains the module memory map and registers.

EWM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_1000	Control Register (EWM_CTRL)	8	R/W	00h	22.3.1/400
4006_1001	Service Register (EWM_SERV)	8	W (always reads zero)	00h	22.3.2/401
4006_1002	Compare Low Register (EWM_CMPL)	8	R/W	00h	22.3.3/402
4006_1003	Compare High Register (EWM_CMPH)	8	R/W	FFh	22.3.4/402

22.3.1 Control Register (EWM_CTRL)

The CTRL register is cleared by any reset.

NOTE

INEN, ASSIN and EWMEN bits can be written once after a CPU reset. Modifying these bits more than once, generates a bus transfer error.

Address: EWM_CTRL is 4006_1000h base + 0h offset = 4006_1000h

Bit	7	6	5	4	3	2	1	0
Read	0				INTEN	INEN	ASSIN	EWMEN
Write								
Reset	0	0	0	0	0	0	0	0

EWM_CTRL field descriptions

Field	Description
7–4 Reserved	This read-only field is reserved and always has the value zero.
3 INTEN	Interrupt Enable. This bit when set and $\overline{\text{EWM\_out}}$ is asserted, an interrupt request is generated. To de-assert interrupt request, user should clear this bit by writing 0.
2 INEN	Input Enable. This bit when set, enables the EWM_in port.
1 ASSIN	EWM_in's Assertion State Select. Default assert state of the EWM_in signal is logic zero. Setting ASSIN bit inverts the assert state to a logic one.
0 EWMEN	EWM enable. This bit when set, enables the EWM module. This resets the EWM counter to zero and deasserts the $\overline{\text{EWM\_out}}$ signal. Clearing EWMEN bit disables the EWM, and therefore it cannot be enabled until a reset occurs, due to the write-once nature of this bit.

22.3.2 Service Register (EWM_SERV)

The SERV register provides the interface from the CPU to the EWM module. It is write-only and reads of this register return zero.

Address: EWM_SERV is 4006_1000h base + 1h offset = 4006_1001h

Bit	7	6	5	4	3	2	1	0
Read	0							
Write	SERVICE							
Reset	0	0	0	0	0	0	0	0

EWM_SERV field descriptions

Field	Description
7–0 SERVICE	The EWM service mechanism requires the CPU to write two values to the SERV register: a first data byte of 0xB4, followed by a second data byte of 0x2C. The EWM service is illegal if either of the following conditions is true. • The first or second data byte is not written correctly. • The second data byte is not written within a fixed number of peripheral bus cycles of the first data byte. This fixed number of cycles is called *EWM_service_time*.

22.3.3 Compare Low Register (EWM_CMPL)

The CMPL register is reset to zero after a CPU reset. This provides no minimum time for the CPU to service the EWM counter.

NOTE

This register can be written only once after a CPU reset. Writing this register more than once generates a bus transfer error.

Address: EWM_CMPL is 4006_1000h base + 2h offset = 4006_1002h

Bit	7	6	5	4	3	2	1	0
Read Write	COMPAREL							
Reset	0	0	0	0	0	0	0	0

EWM_CMPL field descriptions

Field	Description
7–0 COMPAREL	To prevent runaway code from changing this field, software should write to this field after a CPU reset even if the (default) minimum service time is required.

22.3.4 Compare High Register (EWM_CMPH)

The CMPH register is reset to 0xFF after a CPU reset. This provides a maximum of 256 clocks time, for the CPU to service the EWM counter.

NOTE

This register can be written only once after a CPU reset. Writing this register more than once generates a bus transfer error.

NOTE

The valid values for CMPH are up to 0xFE because the EWM counter never expires when CMPH = 0xFF. The expiration happens only if EWM counter is greater than CMPH.

Address: EWM_CMPH is 4006_1000h base + 3h offset = 4006_1003h

Bit	7	6	5	4	3	2	1	0
Read Write	COMPAREH							
Reset	1	1	1	1	1	1	1	1

EWM_CMPH field descriptions

Field	Description
7–0 COMPAREH	To prevent runaway code from changing this field, software should write to this field after a CPU reset even if the (default) maximum service time is required.

22.4 Functional Description

The following sections describe functional details of the EWM module.

22.4.1 The $\overline{\text{EWM\_out}}$ Signal

The $\overline{\text{EWM\_out}}$ is a digital output signal used to gate an external circuit (application specific) that controls critical safety functions. For example, the $\overline{\text{EWM\_out}}$ could be connected to the high voltage transistors circuits that control an AC motor in a large appliance.

The $\overline{\text{EWM\_out}}$ signal remains deasserted when the EWM is being regularly serviced by the CPU within the programmable service window, indicating that the application code is executed as expected.

The $\overline{\text{EWM\_out}}$ signal is asserted in any of the following conditions:

- Servicing the EWM when the counter value is less than CMPL value.
- If the EWM counter value reaches the CMPH value, and no EWM service has occurred.
- Servicing the EWM when the counter value is more than CMPL and less than CMPH values and EWM_in signal is asserted.
- After any reset (by the virtue of the external pull-down mechanism on the $\overline{\text{EWM\_out}}$ pin)

On a normal reset, the $\overline{\text{EWM\_out}}$ is asserted. To deassert the $\overline{\text{EWM\_out}}$, set EWMEN bit in the CTRL register to enable the EWM.

If the $\overline{\text{EWM\_out}}$ signal shares its pad with a digital I/O pin, on reset this actual pad defers to being an input signal. It takes the $\overline{\text{EWM\_out}}$ output condition only after you enable the EWM by the EWMEN bit in the CTRL register.

When the $\overline{\text{EWM\_out}}$ pin is asserted, it can only be deasserted by forcing a MCU reset.

Note

> EWM_out pad must be in pull down state when EWM functionality is used and when EWM is under Reset.

22.4.2 The EWM_in Signal

The EWM_in is a digital input signal that allows an external circuit to control the EWM_out signal. For example, in the application, an external circuit monitors a critical safety function, and if there is fault with this circuit's behavior, it can then actively initiate the EWM_out signal that controls the gating circuit.

The EWM_in signal is ignored if the EWM is disabled, or if INEN bit of CTRL register is cleared, as after any reset.

On enabling the EWM (setting the CTRL[EWMEN] bit) and enabling EWM_in functionality (setting the CTRL[INEN] bit), the EWM_in signal must be in the deasserted state prior to the CPU servicing the EWM. This ensures that the EWM_out stays in the deasserted state; otherwise, the EWM_out pin is asserted.

Note

> You must update the CMPH and CMPL registers prior to enabling the EWM. After enabling the EWM, the counter resets to zero, therefore providing a reasonable time after a power-on reset for the external monitoring circuit to stabilize and ensure that the EWM_in pin is deasserted.

22.4.3 EWM Counter

It is an 8-bit ripple counter fed from a clock source that is independent of the peripheral bus clock source. As the preferred time-out is between 1 ms and 100 ms the actual clock source should be in the kHz range.

The counter is reset to zero, after a CPU reset, or a EWM refresh cycle. The counter value is not accessible to the CPU.

22.4.4 EWM Compare Registers

The compare registers CMPL and CMPH are write-once after a CPU reset and cannot be modified until another CPU reset occurs.

The EWM compare registers are used to create a service window, which is used by the CPU to service/refresh the EWM module.

- If the CPU services the EWM when the counter value lies between CMPL value and CMPH value, the counter is reset to zero. This is a legal service operation.
- If the CPU executes a EWM service/refresh action outside the legal service window, $\overline{\text{EWM\_out}}$ is asserted.

It is illegal to program CMPL and CMPH with same value. In this case, as soon as counter reaches (CMPL + 1), $\overline{\text{EWM\_out}}$ is asserted.

22.4.5 EWM Refresh Mechanism

Other than the initial configuration of the EWM, the CPU can only access the EWM by the EWM Service Register. The CPU must access the EWM service register with correct write of unique data within the windowed time frame as determined by the CMPL and CMPH registers. Therefore, three possible conditions can occur:

Table 22-7. EWM Refresh Mechanisms

Condition	Mechanism
A unique EWM service occurs when CMPL < Counter < CMPH.	The software behaves as expected and the counter of the EWM is reset to zero, and $\overline{\text{EWM\_out}}$ pin remains in the deasserted state. **Note:** EWM_in pin is also assumed to be in the deasserted state.
A unique EWM service occurs when Counter < CMPL	The software services the EWM and therefore resets the counter to zero and asserts the $\overline{\text{EWM\_out}}$ pin (irrespective of the EWM_in pin). The $\overline{\text{EWM\_out}}$ pin is expected to gate critical safety circuits.
Counter value reaches CMPH prior to a unique EWM service	The counter value reaches the CMPH value and no service of the EWM resets the counter to zero and assert the $\overline{\text{EWM\_out}}$ pin (irrespective of the EWM_in pin). The $\overline{\text{EWM\_out}}$ pin is expected to gate critical safety circuits.

Any illegal service on EWM has no effect on $\overline{\text{EWM\_out}}$.

22.4.6 EWM Interrupt

When $\overline{\text{EWM\_out}}$ is asserted, an interrupt request is generated to indicate the assertion of the EWM reset out signal. This interrupt is enabled when CTRL[INTEN] is set. Clearing this bit clears the interrupt request but does not affect $\overline{\text{EWM\_out}}$. The $\overline{\text{EWM\_out}}$ signal can be deasserted only by forcing a system reset.

Chapter 23
Watchdog Timer (WDOG)

23.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The Watchdog Timer (WDOG) keeps a watch on the system functioning and resets it in case of its failure. Reasons for failure include run-away software code and the stoppage of the system clock that in a safety critical system can lead to serious consequences. In such cases, the watchdog brings the system into a safe state of operation. The watchdog monitors the operation of the system by expecting periodic communication from the software, generally known as servicing or refreshing the watchdog. If this periodic refreshing does not occur, the watchdog resets the system.

23.2 Features

The features of the Watchdog Timer (WDOG) include:

- Clock source input independent from CPU/bus clock. Choice between two clock sources:
  - Low-power oscillator (LPO)
  - External system clock
- Unlock sequence for allowing updates to write-once WDOG control/configuration bits.
- All WDOG control/configuration bits are writable once only within 256 bus clock cycles of being unlocked.

  - You need to always update these bits after unlocking within 256 bus clock cycles. Failure to update these bits resets the system.
- Programmable time-out period specified in terms of number of WDOG clock cycles.
- Ability to test WDOG timer and reset with a flag indicating watchdog test.
  - Quick test—Small time-out value programmed for quick test.
  - Byte test—Individual bytes of timer tested one at a time.
  - Read-only access to the WDOG timer—Allows dynamic check that WDOG timer is operational.

NOTE

Reading the watchdog timer counter while running the watchdog on the bus clock might not give the accurate counter value.

- Windowed refresh option
  - Provides robust check that program flow is faster than expected.
  - Programmable window.
  - Refresh outside window leads to reset.
- Robust refresh mechanism
  - Write values of 0xA602 and 0xB480 to WDOG Refresh Register within 20 bus clock cycles.
- Count of WDOG resets as they occur.
- Configurable interrupt on time-out to provide debug breadcrumbs. This is followed by a reset after 256 bus clock cycles.

23.3 Functional overview

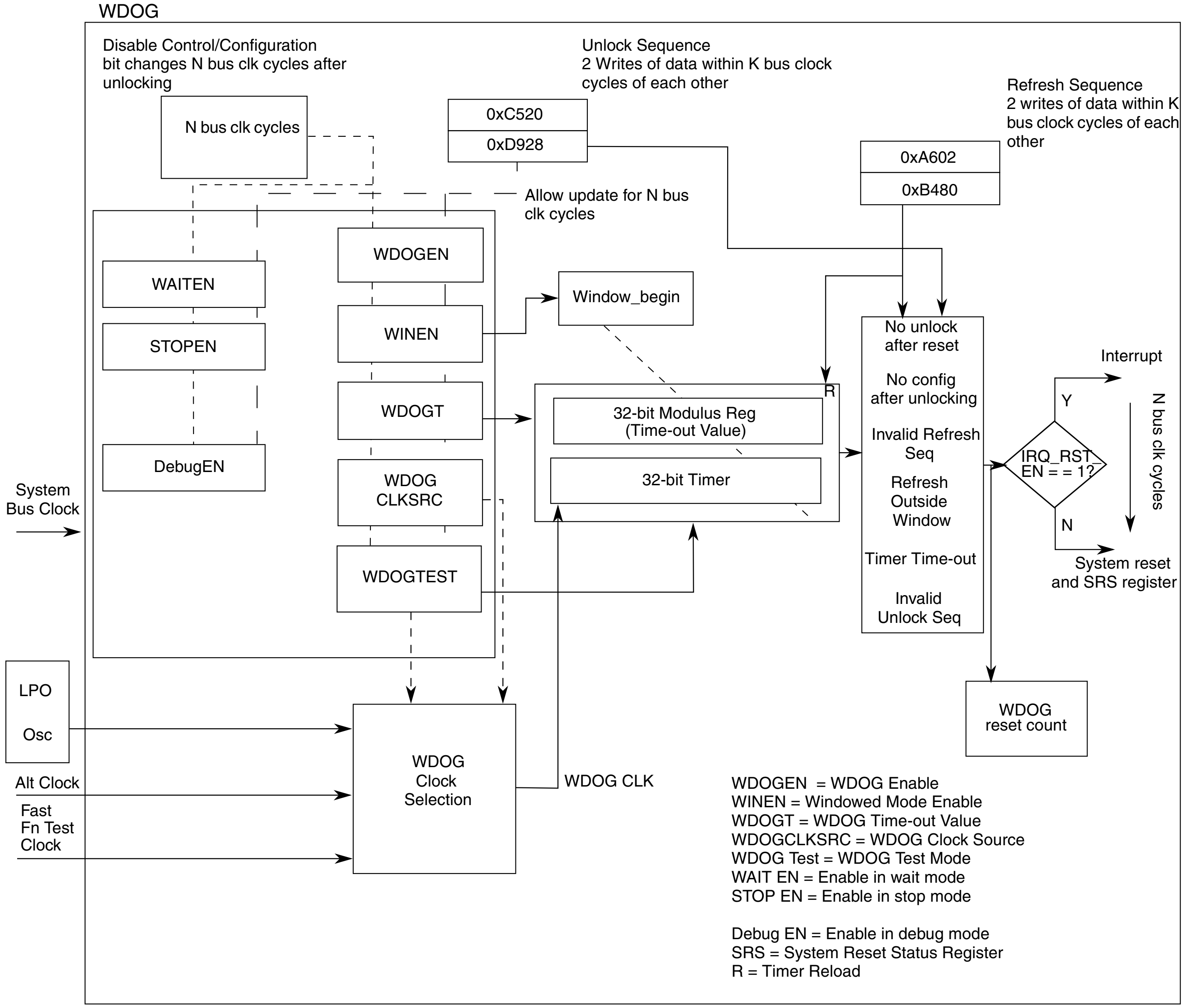

WDOG
Disable Control/Configuration bit changes N bus clk cycles after unlocking
N bus clk cycles
Unlock Sequence
2 Writes of data within K bus clock cycles of each other
0xC520
0xD928
Refresh Sequence
2 writes of data within K bus clock cycles of each other
0xA602
0xB480
Allow update for N bus clk cycles
WAITEN
STOPEN
DebugEN
WDOGEN
WINEN
WDOGT
WDOG CLKSRC
WDOGTEST
Window_begin
R
32-bit Modulus Reg (Time-out Value)
32-bit Timer
No unlock after reset
No config after unlocking
Invalid Refresh Seq
Refresh Outside Window
Timer Time-out
Invalid Unlock Seq
Interrupt
Y
IRQ_RST EN = = 1?
N
N bus clk cycles
System reset and SRS register
WDOG reset count
System Bus Clock
LPO
Osc
Alt Clock
Fast Fn Test Clock
WDOG Clock Selection
WDOG CLK
WDOGEN = WDOG Enable
WINEN = Windowed Mode Enable
WDOGT = WDOG Time-out Value
WDOGCLKSRC = WDOG Clock Source
WDOG Test = WDOG Test Mode
WAIT EN = Enable in wait mode
STOP EN = Enable in stop mode
Debug EN = Enable in debug mode
SRS = System Reset Status Register
R = Timer Reload

Figure 23-1. WDOG operation

The preceding figure shows the operation of the watchdog. The values for N and K are:

- N = 256
- K = 20

The watchdog is a fail safe mechanism that brings the system into a known initial state in case of its failure due to CPU clock stopping or a run-away condition in code execution. In its simplest form, the watchdog timer runs continuously off a clock source and expects to be serviced periodically, failing which it resets the system. This ensures that the software is executing correctly and has not run away in an unintended direction. Software can adjust the period of servicing or the time-out value for the watchdog timer to meet the needs of the application.

You can select a windowed mode of operation that expects the servicing to be done only in a particular window of the time-out period. An attempted servicing of the watchdog outside this window results in a reset. By operating in this mode, you can get an indication of whether the code is running faster than expected. The window length is also user programmable.

If a system fails to update/refresh the watchdog due to an unknown and persistent cause, it will be caught in an endless cycle of resets from the watchdog. To analyze the cause of such conditions, you can program the watchdog to first issue an interrupt, followed by a reset. In the interrupt service routine, the software can analyze the system stack to aid debugging.

To enhance the independence of watchdog from the system, it runs off an independent LPO oscillator clock. You can also switch over to an alternate clock source if required, through a control register bit.

23.3.1 Unlocking and updating the watchdog

As long as ALLOW_UPDATE in the watchdog control register is set, you can unlock and modify the write-once-only control and configuration registers:

1. Write 0xC520 followed by 0xD928 within 20 bus clock cycles to a specific unlock register (WDOG_UNLOCK).
2. Wait one bus clock cycle. You cannot update registers on the bus clock cycle immediately following the write of the unlock sequence.
3. An update window equal in length to the watchdog configuration time (WCT) opens. Within this window, you can update the configuration and control register bits.

These register bits can be modified only once after unlocking.

If none of the configuration and control registers is updated within the update window, the watchdog issues a reset, that is, interrupt-then-reset, to the system. Trying to unlock the watchdog within the WCT after an initial unlock has no effect. During the update operation, the watchdog timer is not paused and continues running in the background. After the update window closes, the watchdog timer restarts and the watchdog functions according to the new configuration.

The update feature is useful for applications that have an initial, non-safety critical part, where the watchdog is kept disabled or with a conveniently long time-out period. This means the application coder does not have to frequently service the watchdog. After the critical part of the application begins, the watchdog can be reconfigured as needed.

The watchdog issues a reset, that is, interrupt-then-reset if enabled, to the system for any of these invalid unlock sequences:

- You write any value other than 0xC520 or 0xD928 to the unlock register.
- ALLOW_UPDATE is set and you allow a gap of more than 20 bus clock cycles between the writing of the unlock sequence values.

An attempted refresh operation between the two writes of the unlock sequence and in the WCT time following a successful unlock, goes undetected. Also, see Watchdog Operation with 8-bit access for guidelines related to 8-bit accesses to the unlock register.

Note

A context switch during unlocking and refreshing may lead to a watchdog reset.

23.3.2 Watchdog configuration time (WCT)

To prevent unintended modification of the watchdog's control and configuration register bits, you are allowed to update them only within a period of 256 bus clock cycles after unlocking. This period is known as the watchdog configuration time (WCT). In addition, these register bits can be modified only once after unlocking them for editing, even after reset.

You must unlock the registers within WCT after system reset, failing which the WDOG issues a reset to the system. In other words, you must write at least the first word of the unlocking sequence within the WCT after reset. After this is done, you have a further 20 bus clock cycles, the maximum allowed gap between the words of the unlock sequence, to complete the unlocking operation. Thereafter, to make sure that you do not forget to configure the watchdog, the watchdog issues a reset if none of the WDOG control and configuration registers is updated in the WCT after unlock. After the close of this window or after the first write, these register bits are locked out from any further changes.

The watchdog timer keeps running according to its default configuration through unlocking and update operations that can extend up to a maximum total of 2xWCT + 20 bus clock cycles. Therefore, it must be ensured that the time-out value for the watchdog is always greater than 2xWCT time + 20 bus clock cycles.

Updates in the write-once registers take effect only after the WCT window closes with the following exceptions for which changes take effect immediately:

- Stop, Wait, and Debug mode enable
- IRQ_RST_EN

The operations of refreshing the watchdog goes undetected during the WCT.

23.3.3 Refreshing the watchdog

A robust refreshing mechanism has been chosen for the watchdog. A valid refresh is a write of 0xA602 followed by 0xB480 within 20 bus clock cycles to watchdog refresh register. If these two values are written more than 20 bus cycles apart or if something other than these two values is written to the register, a watchdog reset, or interrupt-then-reset if enabled, is issued to the system. A valid refresh makes the watchdog timer restart on the next bus clock. Also, an attempted unlock operation in between the two writes of the refresh sequence goes undetected. See Watchdog Operation with 8-bit access for guidelines related to 8-bit accesses to the refresh register.

23.3.4 Windowed mode of operation

In this mode of operation, a restriction is placed on the point in time within the time-out period at which the watchdog can be refreshed. The refresh is considered valid only when the watchdog timer increments beyond a certain count as specified by the watchdog window register. This is known as refreshing the watchdog within a window of the total time-out period. If a refresh is attempted before the timer reaches the window value, the watchdog generates a reset, or interrupt-then-reset if enabled. If there is no refresh at all, the watchdog times out and generates a reset or interrupt-then-reset if enabled.

23.3.5 Watchdog disabled mode of operation

When the watchdog is disabled through the WDOG_EN bit in the watchdog status and control register, the watchdog timer is reset to zero and is disabled from counting until you enable it or it is enabled again by the system reset. In this mode, the watchdog timer cannot be refreshed–there is no requirement to do so while the timer is disabled. However, the watchdog still generates a reset, or interrupt-then-reset if enabled, on a non-time-out exception. See Generated Resets and Interrupts. You need to unlock the watchdog before enabling it. A system reset brings the watchdog out of the disabled mode.

23.3.6 Low-power modes of operation

The low-power modes of operation of the watchdog are described in the following table:

Table 23-1. Low-power modes of operation

Mode	Behavior
Wait	If the WDOG is enabled (WAIT_EN = 1), it can run on bus clock or low-power oscillator clock (CLK_SRC = x) to generate interrupt (IRQ_RST_EN=1) followed by a reset on time-out. After reset the WDOG reset counter increments by one.
Stop	Where the bus clock is gated, the WDOG can run only on low-power oscillator clock (CLK_SRC=0) if it is enabled in stop (STOP_EN=1). In this case, the WDOG runs to time-out twice, and then generates a reset from its backup circuitry. Therefore, if you program the watchdog to time-out after 100 ms and then enter such a stop mode, the reset will occur after 200 ms. Also, in this case, no interrupt will be generated irrespective of the value of IRQ_RST_EN bit. After WDOG reset, the WDOG reset counter will also not increment.
Power-Down	The watchdog is powered off.

23.3.7 Debug modes of operation

You can program the watchdog to disable in debug modes through DBG_EN in the watchdog control register. This results in the watchdog timer pausing for the duration of the mode. Register read/writes are still allowed, which means that operations like refresh, unlock, and so on are allowed. Upon exit from the mode, the timer resumes its operation from the point of pausing.

The entry of the system into the debug mode does not excuse it from compulsorily configuring the watchdog in the WCT time after unlock, unless the system bus clock is gated off, in which case the internal state machine pauses too. Failing to do so still results in a reset, or interrupt-then-reset, if enabled, to the system. Also, all of the exception conditions that result in a reset to the system, as described in Generated Resets and Interrupts, are still valid in this mode. So, if an exception condition occurs and the system bus clock is on, a reset occurs, or interrupt-then-reset, if enabled.

The entry into Debug mode within WCT after reset is treated differently. The WDOG timer is kept reset to zero and there is no need to unlock and configure it within WCT. You must not try to refresh or unlock the WDOG in this state or unknown behavior may result. Upon exit from this mode, the WDOG timer restarts and the WDOG has to be unlocked and configured within WCT.

23.4 Testing the watchdog

For IEC 60730 and other safety standards, the expectation is that anything that monitors a safety function must be tested, and this test is required to be fault tolerant. To test the watchdog, its main timer and its associated compare and reset logic must be tested. To this end, two tests are implemented for the watchdog, as described in Quick Test and Byte Test. A control bit is provided to put the watchdog into functional test mode. There is also an overriding test-disable control bit which allows the functional test mode to be disabled permanently. After it is set, this test-disable bit can only be cleared by a reset.

These two tests achieve the overall aim of testing the counter functioning and the compare and reset logic.

Note

> Do not enable the watchdog interrupt during these tests. If required, you must ensure that the effective time-out value is greater than WCT time. See Generated Resets and Interrupts for more details.

To run a particular test:

1. Select either quick test or byte test..
2. Set a certain test mode bit to put the watchdog in the functional test mode. Setting this bit automatically switches the watchdog timer to a fast clock source. The switching of the clock source is done to achieve a faster time-out and hence a faster test.

In a successful test, the timer times out after reaching the programmed time-out value and generates a system reset.

Note

> After emerging from a reset due to a watchdog test, unlock and configure the watchdog. The refresh and unlock operations and interrupt are not automatically disabled in the test mode.

23.4.1 Quick test

In this test, the time-out value of watchdog timer is programmed to a very low value to achieve quick time-out. The only difference between the quick test and the normal mode of the watchdog is that TESTWDOG is set for the quick test. This allows for a faster test of the watchdog reset mechanism.

23.4.2 Byte test

The byte test is a more thorough a test of the watchdog timer. In this test, the timer is split up into its constituent byte-wide stages that are run independently and tested for time-out against the corresponding byte of the time-out value register. The following figure explains the splitting concept:

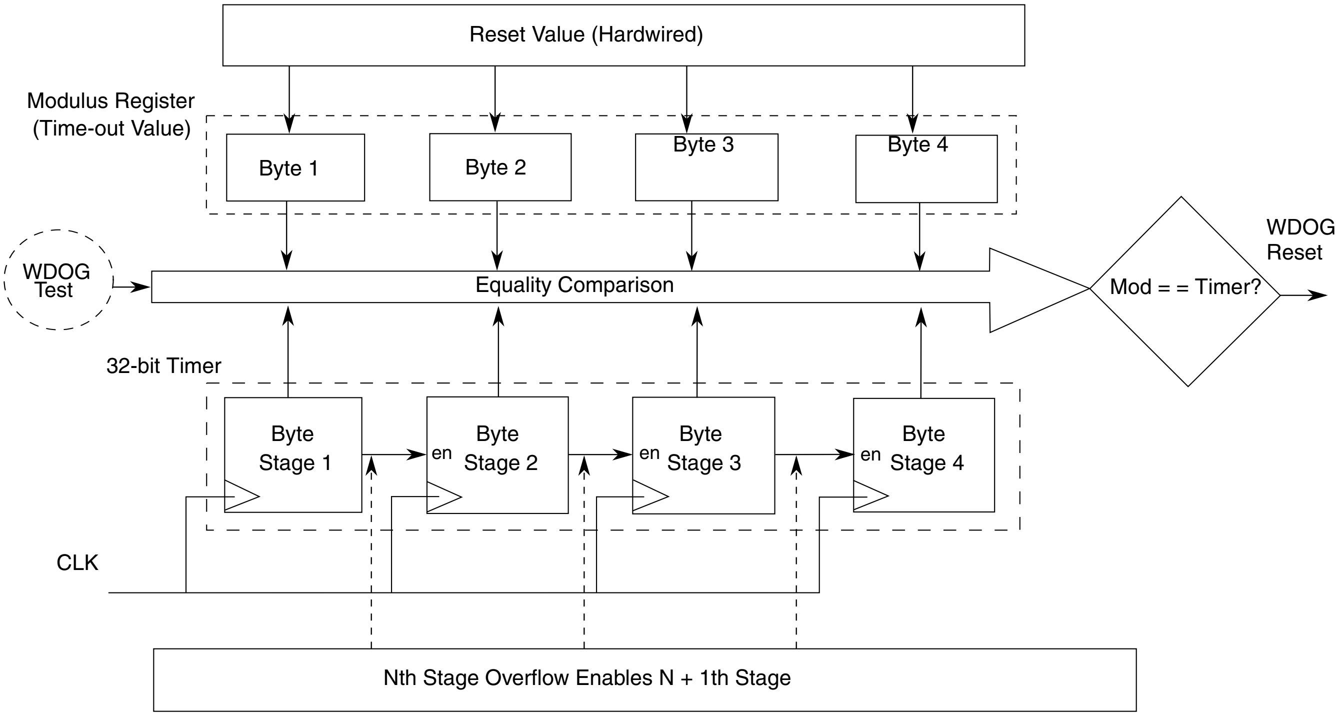

Reset Value (Hardwired)
Modulus Register
(Time-out Value)
Byte 1
Byte 2
Byte 3
Byte 4
WDOG Test
Equality Comparison
Mod = = Timer?
WDOG Reset
32-bit Timer
Byte Stage 1
en
Byte Stage 2
en
Byte Stage 3
en
Byte Stage 4
CLK
Nth Stage Overflow Enables N + 1th Stage

Figure 23-2. Watchdog timer byte splitting

Each stage is an 8-bit synchronous counter followed by combinational logic that generates an overflow signal. The overflow signal acts as an enable to the N + 1th stage.

In the test mode, when an individual byte, N, is tested, byte N – 1 is loaded forcefully with 0xFF, and both these bytes are allowed to run off the clock source. By doing so, the overflow signal from stage N – 1 is generated immediately, enabling counter stage N. The Nth stage runs and compares with the Nth byte of the time-out value register. In this way, the byte N is also tested along with the link between it and the preceding stage. No other stages, N – 2, N – 3... and N + 1, N + 2... are enabled for the test on byte N. These disabled stages, except the most significant stage of the counter, are loaded with a value of 0xFF.

23.5 Backup reset generator

The backup reset generator generates the final reset which goes out to the system. It has a backup mechanism which ensures that in case the bus clock stops and prevents the main state machine from generating a reset exception/interrupt, the watchdog timer's time-out is separately routed out as a reset to the system. Two successive timer time-outs without an intervening system reset result in the backup reset generator routing out the time-out signal as a reset to the system.

23.6 Generated resets and interrupts

The watchdog generates a reset in the following events, also referred to as exceptions:

- A watchdog time-out
- Failure to unlock the watchdog within WCT time after system reset deassertion
- No update of the control and configuration registers within the WCT window after unlocking. At least one of the following registers must be written to within the WCT window to avoid reset:
  - WDOG_ST_CTRL_H, WDOG_ST_CTRL_L
  - WDOG_TO_VAL_H, WDOG_TO_VAL_L
  - WDOG_WIN_H, WDOG_WIN_L
  - WDOG_PRESCALER
- A value other than the unlock sequence or the refresh sequence is written to the unlock and/or refresh registers, respectively.
- A gap of more than 20 bus cycles exists between the writes of two values of the unlock sequence.
- A gap of more than 20 bus cycles exists between the writes of two values of the refresh sequence.

The watchdog can also generate an interrupt. If IRQ_RST_EN is set, then on the above mentioned events WDOG_ST_CTRL_L[INT_FLG] is set, generating an interrupt. A watchdog reset is also generated WCT time later to ensure the watchdog is fault tolerant. The interrupt can be cleared by writing 1 to INT_FLG.

The gap of WCT between interrupt and reset means that the WDOG time-out value must be greater than WCT. Otherwise, if the interrupt was generated due to a time-out, a second consecutive time-out will occur in that WCT gap. This will trigger the backup reset generator to generate a reset to the system, prematurely ending the interrupt service routine execution. Also, jobs such as counting the number of watchdog resets would not be done.

23.7 Memory map and register definition

This section consists of the memory map and register descriptions.

WDOG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4005_2000	Watchdog Status and Control Register High (WDOG_STCTRLH)	16	R/W	01D3h	23.7.1/418
4005_2002	Watchdog Status and Control Register Low (WDOG_STCTRLL)	16	R/W	0001h	23.7.2/420
4005_2004	Watchdog Time-out Value Register High (WDOG_TOVALH)	16	R/W	004Ch	23.7.3/420
4005_2006	Watchdog Time-out Value Register Low (WDOG_TOVALL)	16	R/W	4B4Ch	23.7.4/421
4005_2008	Watchdog Window Register High (WDOG_WINH)	16	R/W	0000h	23.7.5/421
4005_200A	Watchdog Window Register Low (WDOG_WINL)	16	R/W	0010h	23.7.6/422
4005_200C	Watchdog Refresh register (WDOG_REFRESH)	16	R/W	B480h	23.7.7/422
4005_200E	Watchdog Unlock register (WDOG_UNLOCK)	16	R/W	D928h	23.7.8/423
4005_2010	Watchdog Timer Output Register High (WDOG_TMROUTH)	16	R/W	0000h	23.7.9/423
4005_2012	Watchdog Timer Output Register Low (WDOG_TMROUTL)	16	R/W	0000h	23.7.10/ 423
4005_2014	Watchdog Reset Count register (WDOG_RSTCNT)	16	R/W	0000h	23.7.11/ 424
4005_2016	Watchdog Prescaler register (WDOG_PRESC)	16	R/W	0400h	23.7.12/ 424

23.7.1 Watchdog Status and Control Register High (WDOG_STCTRLH)

Address: WDOG_STCTRLH is 4005_2000h base + 0h offset = 4005_2000h

Bit	15	14	13	12
Read	0	DISTESTWDOG	BYTESEL[1:0]	
Write				
Reset	0	0	0	0

Bit	11	10	9	8
Read	TESTSEL	TESTWDOG	0	Reserved
Write				
Reset	0	0	0	1

Bit	7	6	5	4
Read Write	WAITEN	STOPEN	DBGEN	ALLOWUPDATE
Reset	1	1	0	1

Bit	3	2	1	0
Read Write	WINEN	IRQRSTEN	CLKSRC	WDOGEN
Reset	0	0	1	1

WDOG_STCTRLH field descriptions

Field	Description
15 Reserved	This read-only field is reserved and always has the value zero.
14 DISTESTWDOG	Allows the WDOG's functional test mode to be disabled permanently. After it is set, it can only be cleared by a reset. It cannot be unlocked for editing after it is set. 0 WDOG functional test mode is not disabled. 1 WDOG functional test mode is disabled permanently until reset.
13–12 BYTESEL[1:0]	This 2-bit field selects the byte to be tested when the watchdog is in the byte test mode. 00 Byte 0 selected 01 Byte 1 selected 10 Byte 2 selected 11 Byte 3 selected
11 TESTSEL	Effective only if TESTWDOG is set. Selects the test to be run on the watchdog timer. 0 Quick test. The timer runs in normal operation. You can load a small time-out value to do a quick test. 1 Byte test. Puts the timer in the byte test mode where individual bytes of the timer are enabled for operation and are compared for time-out against the corresponding byte of the programmed time-out value. Select the byte through BYTESEL[1:0] for testing.
10 TESTWDOG	Puts the watchdog in the functional test mode. In this mode, the watchdog timer and the associated compare and reset generation logic is tested for correct operation. The clock for the timer is switched from the main watchdog clock to the fast clock input for watchdog functional test. The TESTSEL bit selects the test to be run.

Table continues on the next page...

WDOG_STCTRLH field descriptions (continued)

Field	Description
9 Reserved	This read-only field is reserved and always has the value zero.
8 Reserved	This field is reserved.
7 WAITEN	Enables or disables WDOG in Wait mode. 0 WDOG is disabled in CPU Wait mode. 1 WDOG is enabled in CPU Wait mode.
6 STOPEN	Enables or disables WDOG in Stop mode. 0 WDOG is disabled in CPU Stop mode. 1 WDOG is enabled in CPU Stop mode.
5 DBGEN	Enables or disables WDOG in Debug mode. 0 WDOG is disabled in CPU Debug mode. 1 WDOG is enabled in CPU Debug mode.
4 ALLOWUPDATE	Enables updates to watchdog write-once registers, after the reset-triggered initial configuration window (WCT) closes, through unlock sequence. 0 No further updates allowed to WDOG write-once registers. 1 WDOG write-once registers can be unlocked for updating.
3 WINEN	Enables Windowing mode. 0 Windowing mode is disabled. 1 Windowing mode is enabled.
2 IRQRSTEN	Used to enable the debug breadcrumbs feature. A change in this bit is updated immediately, as opposed to updating after WCT. 0 WDOG time-out generates reset only. 1 WDOG time-out initially generates an interrupt. After WCT, it generates a reset.
1 CLKSRC	Selects clock source for the WDOG timer and other internal timing operations. 0 WDOG clock sourced from LPO . 1 WDOG clock sourced from alternate clock source.
0 WDOGEN	Enables or disables the WDOG's operation. In the disabled state, the watchdog timer is kept in the reset state, but the other exception conditions can still trigger a reset/interrupt. A change in the value of this bit must be held for more than one WDOG_CLK cycle for the WDOG to be enabled or disabled. 0 WDOG is disabled. 1 WDOG is enabled.

23.7.2 Watchdog Status and Control Register Low (WDOG_STCTRLL)

Address: WDOG_STCTRLL is 4005_2000h base + 2h offset = 4005_2002h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	INTFLG	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

WDOG_STCTRLL field descriptions

Field	Description
15 INTFLG	Interrupt flag. It is set when an exception occurs. IRQRSTEN = 1 is a precondition to set this flag. INTFLG = 1 results in an interrupt being issued followed by a reset, WCT later. The interrupt can be cleared by writing 1 to this bit. It also gets cleared on a system reset.
14–0 Reserved	This field is reserved.

23.7.3 Watchdog Time-out Value Register High (WDOG_TOVALH)

Address: WDOG_TOVALH is 4005_2000h base + 4h offset = 4005_2004h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	TOVALHIGH															
Reset	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0

WDOG_TOVALH field descriptions

Field	Description
15–0 TOVALHIGH	Defines the upper 16 bits of the 32-bit time-out value for the watchdog timer. It is defined in terms of cycles of the watchdog clock.

23.7.4 Watchdog Time-out Value Register Low (WDOG_TOVALL)

The time-out value of the watchdog must be set to a minimum of four watchdog clock cycles. This is to take into account the delay in new settings taking effect in the watchdog clock domain.

Address: WDOG_TOVALL is 4005_2000h base + 6h offset = 4005_2006h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	TOVALLOW															
Reset	0	1	0	0	1	0	1	1	0	1	0	0	1	1	0	0

WDOG_TOVALL field descriptions

Field	Description
15–0 TOVALLOW	Defines the lower 16 bits of the 32-bit time-out value for the watchdog timer. It is defined in terms of cycles of the watchdog clock.

23.7.5 Watchdog Window Register High (WDOG_WINH)

NOTE

You must set the Window Register value lower than the Time-out Value Register.

Address: WDOG_WINH is 4005_2000h base + 8h offset = 4005_2008h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	WINHIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOG_WINH field descriptions

Field	Description
15–0 WINHIGH	Defines the upper 16 bits of the 32-bit window for the windowed mode of operation of the watchdog. It is defined in terms of cycles of the watchdog clock. In this mode, the watchdog can be refreshed only when the timer has reached a value greater than or equal to this window length. A refresh outside this window resets the system or if IRQRSTEN is set, it interrupts and then resets the system.

23.7.6 Watchdog Window Register Low (WDOG_WINL)

NOTE

You must set the Window Register value lower than the Time-out Value Register.

Address: WDOG_WINL is 4005_2000h base + Ah offset = 4005_200Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	WINLOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

WDOG_WINL field descriptions

Field	Description
15–0 WINLOW	Defines the lower 16 bits of the 32-bit window for the windowed mode of operation of the watchdog. It is defined in terms of cycles of the pre-scaled watchdog clock. In this mode, the watchdog can be refreshed only when the timer reaches a value greater than or equal to this window length value. A refresh outside of this window resets the system or if IRQRSTEN is set, it interrupts and then resets the system.

23.7.7 Watchdog Refresh register (WDOG_REFRESH)

Address: WDOG_REFRESH is 4005_2000h base + Ch offset = 4005_200Ch

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	WDOGREFRESH															
Reset	1	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0

WDOG_REFRESH field descriptions

Field	Description
15–0 WDOGREFRESH	Watchdog refresh register. A sequence of 0xA602 followed by 0xB480 within 20 bus clock cycles written to this register refreshes the WDOG and prevents it from resetting the system. Writing a value other than the above mentioned sequence or if the sequence is longer than 20 bus cycles, resets the system, or if IRQRSTEN is set, it interrupts and then resets the system.

23.7.8 Watchdog Unlock register (WDOG_UNLOCK)

Address: WDOG_UNLOCK is 4005_2000h base + Eh offset = 4005_200Eh

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	WDOGUNLOCK															
Reset	1	1	0	1	1	0	0	1	0	0	1	0	1	0	0	0

WDOG_UNLOCK field descriptions

Field	Description
15–0 WDOGUNLOCK	Writing the unlock sequence values to this register to makes the watchdog write-once registers writable again. The required unlock sequence is 0xC520 followed by 0xD928 within 20 bus clock cycles. A valid unlock sequence opens a window equal in length to the WCT within which you can update the registers. Writing a value other than the above mentioned sequence or if the sequence is longer than 20 bus cycles, resets the system or if IRQRSTEN is set, it interrupts and then resets the system. The unlock sequence is effective only if ALLOWUPDATE is set.

23.7.9 Watchdog Timer Output Register High (WDOG_TMROUTH)

Address: WDOG_TMROUTH is 4005_2000h base + 10h offset = 4005_2010h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	TIMEROUTHIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOG_TMROUTH field descriptions

Field	Description
15–0 TIMEROUTHIGH	Shows the value of the upper 16 bits of the watchdog timer.

23.7.10 Watchdog Timer Output Register Low (WDOG_TMROUTL)

During Stop mode, the WDOG_TIMER_OUT will be caught at the pre-stop value of the watchdog timer. After exiting Stop mode, a maximum delay of 1 WDOG_CLK cycle + 3 bus clock cycles will occur before the WDOG_TIMER_OUT starts following the watchdog timer.

Address: WDOG_TMROUTL is 4005_2000h base + 12h offset = 4005_2012h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	TIMEROUTLOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOG_TMROUTL field descriptions

Field	Description
15–0 TIMEROUTLOW	Shows the value of the lower 16 bits of the watchdog timer.

23.7.11 Watchdog Reset Count register (WDOG_RSTCNT)

Address: WDOG_RSTCNT is 4005_2000h base + 14h offset = 4005_2014h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	RSTCNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOG_RSTCNT field descriptions

Field	Description
15–0 RSTCNT	Counts the number of times the watchdog resets the system. This register is reset only on a POR. Writing 1 to the bit to be cleared enables you to clear the contents of this register.

23.7.12 Watchdog Prescaler register (WDOG_PRESC)

Address: WDOG_PRESC is 4005_2000h base + 16h offset = 4005_2016h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0					PRESCVAL			0							
Write	▨	▨	▨	▨	▨				▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

WDOG_PRESC field descriptions

Field	Description
15–11 Reserved	This read-only field is reserved and always has the value zero.
10–8 PRESCVAL	3-bit prescaler for the watchdog clock source. A value of zero indicates no division of the input WDOG clock. The watchdog clock is divided by (PRESCVAL + 1) to provide the prescaled WDOG_CLK.
7–0 Reserved	This read-only field is reserved and always has the value zero.

23.8 Watchdog operation with 8-bit access

23.8.1 General guideline

When performing 8-bit accesses to the watchdog's 16-bit registers where the intention is to access both the bytes of a register, place the two 8-bit accesses one after the other in your code.

23.8.2 Refresh and unlock operations with 8-bit access

One exception condition that generates a reset to the system is the write of any value other than those required for a legal refresh/update sequence to the respective refresh and unlock registers.

For an 8-bit access to these registers, writing a correct value requires at least two bus clock cycles, resulting in an invalid value in the registers for one cycle. Therefore, the system is reset even if the intention is to write a correct value to the refresh/unlock register. Keeping this in mind, the exception condition for 8-bit accesses is slightly modified.

Whereas the match for a correct value for a refresh/unlock sequence is as according to the original definition, the match for an incorrect value is done byte-wise on the refresh/unlock rather than for the whole 16-bit value. This means that if the high byte of the refresh/unlock register contains any value other than high bytes of the two values that make up the sequence, it is treated as an exception condition, leading to a reset or interrupt-then-reset. The same holds true for the lower byte of the refresh or unlock register. Take the refresh operation that expects a write of 0xA602 followed by 0xB480 to the refresh register, as an example.

Table 23-15. Refresh for 8-bit access

	WDOG_REFRESH[15:8]	WDOG_REFRESH[7:0]	Sequence value1 or value2 match	Mismatch exception
Current Value	0xB4	0x80	Value2 match	No
Write 1	0xB4	0x02	No match	No
Write 2	0xA6	0x02	Value1 match	No
Write 3	0xB4	0x02	No match	No
Write 4	0xB4	0x80	Value2 match. Sequence complete.	No
Write 5	0x02	0x80	No match	Yes

As shown in the preceding table, the refresh register holds its reset value initially. Thereafter, two 8-bit accesses are performed on the register to write the first value of the refresh sequence. No mismatch exception is registered on the intermediate write, Write1. The sequence is completed by performing two more 8-bit accesses, writing in the second value of the sequence for a successful refresh. It must be noted that the match of value2 takes place only when the complete 16-bit value is correctly written, write4. Hence, the requirement of writing value2 of the sequence within 20 bus clock cycles of value1 is checked by measuring the gap between write2 and write4.

It is reiterated that the condition for matching values 1 and 2 of the refresh or unlock sequence remains unchanged. It is only the criterion for detecting a wrong value in these registers which has been relaxed, as explained, for 8-bit accesses. Any 16-bit access still needs to adhere to the original guidelines, mentioned in the sections Refreshing the Watchdog.

23.9 Restrictions on watchdog operation

This section mentions some exceptions to the watchdog operation that may not be apparent to you.

- Restriction on unlock/refresh operations—In the period between the closure of the WCT window after unlock and the actual reload of the watchdog timer, unlock and refresh operations need not be attempted.
- The update and reload of the watchdog timer happens two to three watchdog clocks after WCT window closes, following a successful configuration on unlock.
- Clock Switching Delay—The watchdog uses glitch-free multiplexers at two places – one to choose between the LPO oscillator input and alternate clock input, and the other to choose between the watchdog functional clock and fast clock input for watchdog functional test. A maximum time period of ~2 clock A cycles plus ~2 clock B cycles elapses from the time a switch is requested to the occurrence of the actual clock switch, where clock A and B are the two input clocks to the clock mux.
- For the windowed mode, there is a two to three bus clock latency between the watchdog counter going past the window value and the same registering in the bus clock domain.
- For proper operation of the watchdog, the watchdog clock must be at least five times slower than the system bus clock at all times. An exception is when the watchdog clock is synchronous to the bus clock wherein the watchdog clock can be as fast as the bus clock.

- WCT must be equivalent to at least three watchdog clock cycles. If not ensured, this means that even after the close of the WCT window, you have to wait for the synchronized system reset to deassert in the watchdog clock domain, before expecting the configuration updates to take effect.
- The time-out value of the watchdog should be set to a minimum of four watchdog clock cycles. This is to take into account the delay in new settings taking effect in the watchdog clock domain.
- You must take care not only to refresh the watchdog within the watchdog timer's actual time-out period, but also provide enough allowance for the time it takes for the refresh sequence to be detected by the watchdog timer, on the watchdog clock.
- Updates cannot be made in the bus clock cycle immediately following the write of the unlock sequence, but one bus clock cycle later.
- It should be ensured that the time-out value for the watchdog is always greater than 2xWCT time + 20 bus clock cycles.
- An attempted refresh operation, in between the two writes of the unlock sequence and in the WCT time following a successful unlock, will go undetected.
- Trying to unlock the watchdog within the WCT time after an initial unlock has no effect.
- The refresh and unlock operations and interrupt are not automatically disabled in the watchdog functional test mode.
- After emerging from a reset due to a watchdog functional test, you are still expected to go through the mandatory steps of unlocking and configuring the watchdog. The watchdog continues to be in its functional test mode and therefore you should pull the watchdog out of the functional test mode within WCT time of reset.
- After emerging from a reset due to a watchdog functional test, you still need to go through the mandatory steps of unlocking and configuring the watchdog.
- You must ensure that both the clock inputs to the glitchless clock multiplexers are alive during the switching of clocks. Failure to do so results in a loss of clock at their outputs.
- There is a gap of two to three watchdog clock cycles from the point that stop mode is entered to the watchdog timer actually pausing, due to synchronization. The same holds true for an exit from the stop mode, this time resulting in a two to three

watchdog clock cycle delay in the timer restarting. In case the duration of the stop mode is less than one watchdog clock cycle, the watchdog timer is not guaranteed to pause.

- Consider the case when the first refresh value is written, following which the system enters stop mode with system bus clk still on. If the second refresh value is not written within 20 bus cycles of the first value, the system is reset, or interrupt-then-reset if enabled.

Chapter 24
Multipurpose Clock Generator (MCG)

24.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The multipurpose clock generator (MCG) module provides several clock source choices for the MCU. The module contains a frequency-locked loop (FLL) and a phase-locked loop (PLL). The FLL is controllable by either an internal or an external reference clock. The PLL is controllable by the external reference clock. The module can select either of the FLL or PLL output clocks, or either of the internal or external reference clocks as a source for the MCU system clock. The MCG operates in conjuction with a crystal oscillator, which allows an external crystal, ceramic resonator, or another external clock source to produce the external reference clock.

24.1.1 Features

Key features of the MCG module are:

- Frequency-locked loop (FLL):
  - Digitally-controlled oscillator (DCO)
  - DCO frequency range is programmable for up to four different frequency ranges.
  - Option to program and maximize DCO output frequency for a low frequency external reference clock source.
  - Option to prevent FLL from resetting its current locked frequency when switching clock modes if FLL reference frequency is not changed.

  - Internal or external reference clock can be used as the FLL source.
  - Can be used as a clock source for other on-chip peripherals.
- Phase-locked loop (PLL):
  - Voltage-controlled oscillator (VCO)
  - External reference clock is used as the PLL source.
  - Modulo VCO frequency divider
  - Phase/Frequency detector
  - Integrated loop filter
  - Can be used as a clock source for other on-chip peripherals.
- Internal reference clock generator:
  - Slow clock with nine trim bits for accuracy
  - Fast clock with four trim bits
  - Can be used as source clock for the FLL. In FEI mode, only the slow Internal Reference Clock (IRC) can be used as the FLL source.
  - Either the slow or the fast clock can be selected as the clock source for the MCU.
  - Can be used as a clock source for other on-chip peripherals.
- Control signals for the MCG external reference low power oscillator clock generators are provided:
  - HGO0, RANGE0, EREFS0
- External clock from the Crystal Oscillator :
  - Can be used as a source for the FLL and/or the PLL.
  - Can be selected as the clock source for the MCU.
- External clock from the Real Time Counter (RTC):
  - Can only be used as a source for the FLL.
  - Can be selected as the clock source for the MCU.
- External clock monitor with reset and interrupt request capability to check for external clock failure when running in FBE, PEE, BLPE, or FEE modes
- Lock detector with interrupt request capability for use with the PLL

- Internal Reference Clocks Auto Trim Machine (ATM) capability using an external clock as a reference
- Reference dividers for both the FLL and PLL are provided
- Reference dividers for the Fast Internal Reference Clock are provided
- MCG PLL Clock (MCGPLLCLK) is provided as a clock source for other on-chip peripherals
- MCG FLL Clock (MCGFLLCLK) is provided as a clock source for other on-chip peripherals
- MCG Fixed Frequency Clock (MCGFFCLK) is provided as a clock source for other on-chip peripherals
- MCG Internal Reference Clock (MCGIRCLK) is provided as a clock source for other on-chip peripherals

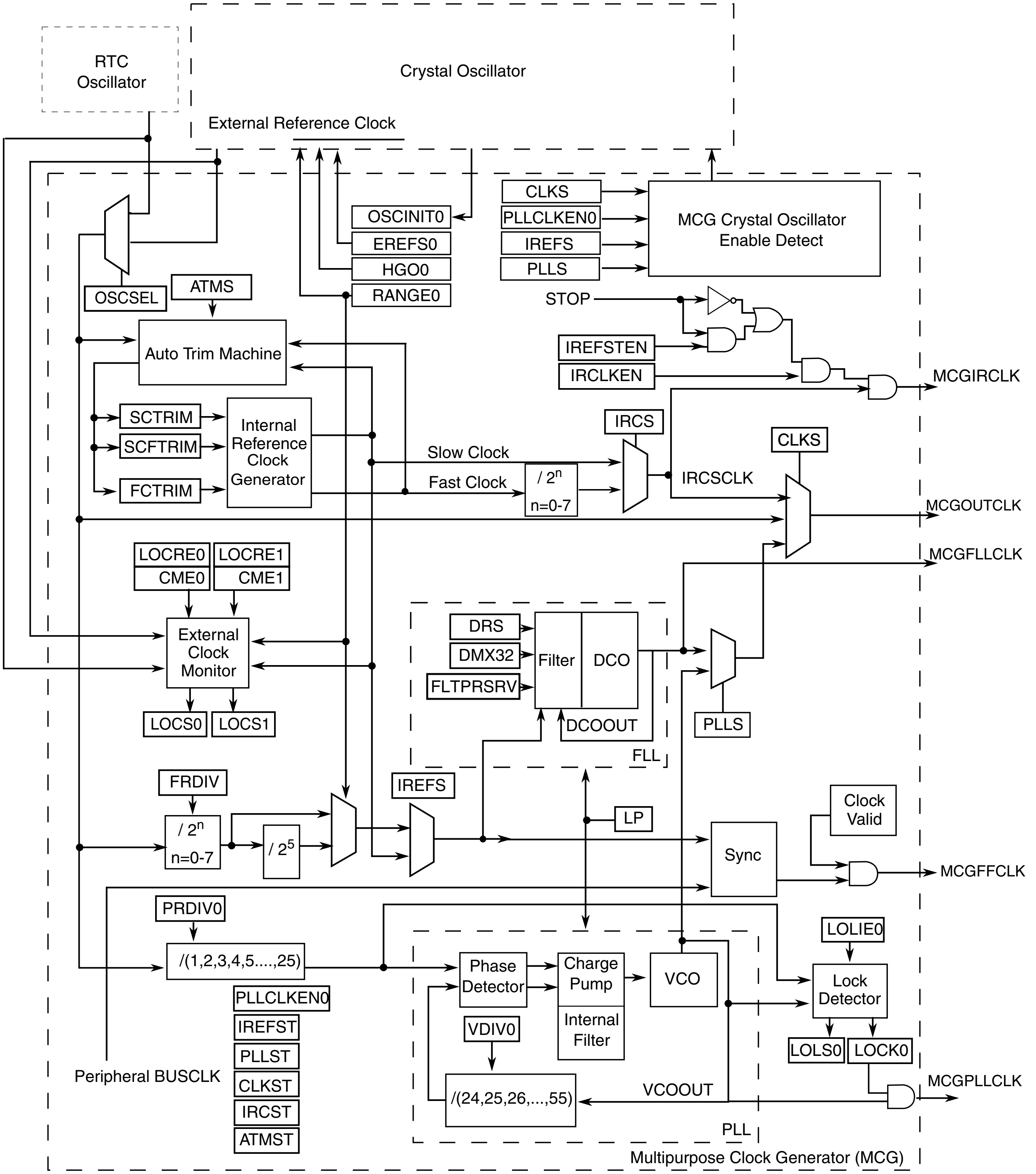

RTC
Oscillator
Crystal Oscillator
External Reference Clock
OSCINIT0
EREFS0
HGO0
RANGE0
CLKS
PLLCLKEN0
IREFS
PLLS
MCG Crystal Oscillator Enable Detect
STOP
IREFSTEN
IRCLKEN
MCGIRCLK
OSCSEL
ATMS
Auto Trim Machine
SCTRIM
SCFTRIM
FCTRIM
Internal Reference Clock Generator
Slow Clock
Fast Clock
/ 2^n
n=0-7
IRCS
IRCSCLK
CLKS
MCGOUTCLK
MCGFLLCLK
LOCRE0
CME0
LOCRE1
CME1
External Clock Monitor
LOCS0
LOCS1
DRS
DMX32
FLTPRSRV
Filter
DCO
DCOOUT
FLL
PLLS
FRDIV
/ 2^n
n=0-7
/ 2^5
IREFS
LP
Sync
Clock Valid
MCGFFCLK
PRDIV0
/(1,2,3,4,5....,25)
LOLIE0
PLLCLKEN0
IREFST
PLLST
CLKST
IRCST
ATMST
Phase Detector
Charge Pump
Internal Filter
VCO
VDIV0
Lock Detector
LOLS0
LOCK0
Peripheral BUSCLK
/(24,25,26,...,55)
VCOOUT
MCGPLLCLK
PLL
Multipurpose Clock Generator (MCG)

Figure 24-1. Multipurpose Clock Generator (MCG) block diagram

24.1.2 Modes of Operation

There are nine modes of operation for the MCG: FEI, FEE, FBI, FBE, PBE, PEE, BLPI, BLPE, and Stop. For details, see MCG modes of operation.

24.2 External Signal Description

There are no MCG signals that connect off chip.

24.3 Memory Map/Register Definition

This section includes the memory map and register definition.

The MCG registers can only be written to when in supervisor mode. Write accesses when in user mode will result in a bus error. Read accesses may be performed in both supervisor and user modes.

MCG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_4000	MCG Control 1 Register (MCG_C1)	8	R/W	04h	24.3.1/434
4006_4001	MCG Control 2 Register (MCG_C2)	8	R/W	80h	24.3.2/435
4006_4002	MCG Control 3 Register (MCG_C3)	8	R/W	Undefined	24.3.3/436
4006_4003	MCG Control 4 Register (MCG_C4)	8	R/W	Undefined	24.3.4/437
4006_4004	MCG Control 5 Register (MCG_C5)	8	R/W	00h	24.3.5/438
4006_4005	MCG Control 6 Register (MCG_C6)	8	R/W	00h	24.3.6/439
4006_4006	MCG Status Register (MCG_S)	8	R	10h	24.3.7/441
4006_4008	MCG Status and Control Register (MCG_SC)	8	R/W	02h	24.3.8/442
4006_400A	MCG Auto Trim Compare Value High Register (MCG_ATCVH)	8	R/W	00h	24.3.9/444
4006_400B	MCG Auto Trim Compare Value Low Register (MCG_ATCVL)	8	R/W	00h	24.3.10/ 444
4006_400C	MCG Control 7 Register (MCG_C7)	8	R/W	00h	24.3.11/ 444
4006_400D	MCG Control 8 Register (MCG_C8)	8	R/W	80h	24.3.12/ 445

24.3.1 MCG Control 1 Register (MCG_C1)

Address: MCG_C1 is 4006_4000h base + 0h offset = 4006_4000h

Bit	7	6	5	4	3	2	1	0
Read Write	CLKS		FRDIV			IREFS	IRCLKEN	IREFSTEN
Reset	0	0	0	0	0	1	0	0

MCG_C1 field descriptions

Field	Description
7–6 CLKS	Clock Source Select Selects the clock source for MCGOUTCLK . 00 Encoding 0 — Output of FLL or PLL is selected (depends on PLLS control bit). 01 Encoding 1 — Internal reference clock is selected. 10 Encoding 2 — External reference clock is selected. 11 Encoding 3 — Reserved.
5–3 FRDIV	FLL External Reference Divider Selects the amount to divide down the external reference clock for the FLL. The resulting frequency must be in the range 31.25 kHz to 39.0625 kHz (This is required when FLL/DCO is the clock source for MCGOUTCLK . In FBE mode, it is not required to meet this range, but it is recommended in the cases when trying to enter a FLL mode from FBE). 000 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 1; for all other RANGE 0 values, Divide Factor is 32. 001 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 2; for all other RANGE 0 values, Divide Factor is 64. 010 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 4; for all other RANGE 0 values, Divide Factor is 128. 011 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 8; for all other RANGE 0 values, Divide Factor is 256. 100 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 16; for all other RANGE 0 values, Divide Factor is 512. 101 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 32; for all other RANGE 0 values, Divide Factor is 1024. 110 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 64; for all other RANGE 0 values, Divide Factor is 1280 . 111 If RANGE 0 = 0 or OSCSEL=1 , Divide Factor is 128; for all other RANGE 0 values, Divide Factor is 1536 .
2 IREFS	Internal Reference Select Selects the reference clock source for the FLL. 0 External reference clock is selected. 1 The slow internal reference clock is selected.
1 IRCLKEN	Internal Reference Clock Enable Enables the internal reference clock for use as MCGIRCLK.

Table continues on the next page...

MCG_C1 field descriptions (continued)

Field	Description
	0 MCGIRCLK inactive. 1 MCGIRCLK active.
0 IREFSTEN	Internal Reference Stop Enable Controls whether or not the internal reference clock remains enabled when the MCG enters Stop mode. 0 Internal reference clock is disabled in Stop mode. 1 Internal reference clock is enabled in Stop mode if IRCLKEN is set or if MCG is in FEI, FBI, or BLPI modes before entering Stop mode.

24.3.2 MCG Control 2 Register (MCG_C2)

Address: MCG_C2 is 4006_4000h base + 1h offset = 4006_4001h

Bit	7	6	5	4	3	2	1	0
Read	LOCRE0	0	RANGE0		HGO0	EREFS0	LP	IRCS
Write		▨						
Reset	1	0	0	0	0	0	0	0

MCG_C2 field descriptions

Field	Description
7 LOCRE0	Loss of Clock Reset Enable Determines whether an interrupt or a reset request is made following a loss of OSC0 external reference clock. The LOCRE0 only has an affect when CME0 is set. 0 Interrupt request is generated on a loss of OSC0 external reference clock. 1 Generate a reset request on a loss of OSC0 external reference clock.
6 Reserved	This read-only field is reserved and always has the value zero.
5–4 RANGE0	Frequency Range Select Selects the frequency range for the crystal oscillator or external clock source. See the Oscillator (OSC) chapter for more details and the device data sheet for the frequency ranges used. 00 Encoding 0 — Low frequency range selected for the crystal oscillator . 01 Encoding 1 — High frequency range selected for the crystal oscillator . 1X Encoding 2 — Very high frequency range selected for the crystal oscillator .
3 HGO0	High Gain Oscillator Select Controls the crystal oscillator mode of operation. See the Oscillator (OSC) chapter for more details. 0 Configure crystal oscillator for low-power operation. 1 Configure crystal oscillator for high-gain operation.
2 EREFS0	External Reference Select

Table continues on the next page...

MCG_C2 field descriptions (continued)

Field	Description
	Selects the source for the external reference clock. See the Oscillator (OSC) chapter for more details. 0 External reference clock requested. 1 Oscillator requested.
1 LP	Low Power Select Controls whether the FLL or PLL is disabled in BLPI and BLPE modes. In FBE or PBE modes, setting this bit to 1 will transition the MCG into BLPE mode; in FBI mode, setting this bit to 1 will transition the MCG into BLPI mode. In any other MCG mode, LP bit has no affect. 0 FLL or PLL is not disabled in bypass modes. 1 FLL or PLL is disabled in bypass modes (lower power)
0 IRCS	Internal Reference Clock Select Selects between the fast or slow internal reference clock source. 0 Slow internal reference clock selected. 1 Fast internal reference clock selected.

24.3.3 MCG Control 3 Register (MCG_C3)

Address: MCG_C3 is 4006_4000h base + 2h offset = 4006_4002h

Bit	7	6	5	4	3	2	1	0
Read Write	SCTRIM							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

MCG_C3 field descriptions

Field	Description
7–0 SCTRIM	Slow Internal Reference Clock Trim Setting SCTRIM [1] controls the slow internal reference clock frequency by controlling the slow internal reference clock period. The SCTRIM bits are binary weighted, that is, bit 1 adjusts twice as much as bit 0. Increasing the binary value increases the period, and decreasing the value decreases the period. An additional fine trim bit is available in C4 register as the SCFTRIM bit. Upon reset, this value is loaded with a factory trim value. If an SCTRIM value stored in nonvolatile memory is to be used, it is your responsibility to copy that value from the nonvolatile memory location to this register.

1. A value for SCTRIM is loaded during reset from a factory programmed location .

24.3.4 MCG Control 4 Register (MCG_C4)

NOTE

Reset values for DRST and DMX32 bits are 0.

Address: MCG_C4 is 4006_4000h base + 3h offset = 4006_4003h

Bit	7	6	5	4	3	2	1	0
Read Write	DMX32	DRST_DRS		FCTRIM				SCFTRIM
Reset	0	0	0	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.
- A value for FCTRIM is loaded during reset from a factory programmed location . x = Undefined at reset.

MCG_C4 field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>7
DMX32</td><td>DCO Maximum Frequency with 32.768 kHz Reference
The DMX32 bit controls whether the DCO frequency range is narrowed to its maximum frequency with a 32.768 kHz reference.
The following table identifies settings for the DCO frequency range.
NOTE: The system clocks derived from this source should not exceed their specified maximums.
<table>
<tr><td>DRST_DRS</td><td>DMX32</td><td>Reference Range</td><td>FLL Factor</td><td>DCO Range</td></tr>
<tr><td rowspan="2">00</td><td>0</td><td>31.25–39.0625 kHz</td><td>640</td><td>20–25 MHz</td></tr>
<tr><td>1</td><td>32.768 kHz</td><td>732</td><td>24 MHz</td></tr>
<tr><td rowspan="2">01</td><td>0</td><td>31.25–39.0625 kHz</td><td>1280</td><td>40–50 MHz</td></tr>
<tr><td>1</td><td>32.768 kHz</td><td>1464</td><td>48 MHz</td></tr>
<tr><td rowspan="2">10</td><td>0</td><td>31.25–39.0625 kHz</td><td>1920</td><td>60–75 MHz</td></tr>
<tr><td>1</td><td>32.768 kHz</td><td>2197</td><td>72 MHz</td></tr>
<tr><td rowspan="2">11</td><td>0</td><td>31.25–39.0625 kHz</td><td>2560</td><td>80–100 MHz</td></tr>
<tr><td>1</td><td>32.768 kHz</td><td>2929</td><td>96 MHz</td></tr>
</table>
0 DCO has a default range of 25%.
1 DCO is fine-tuned for maximum frequency with 32.768 kHz reference.</td></tr>
<tr><td>6–5
DRST_DRS</td><td>DCO Range Select</td></tr>
</table>

Table continues on the next page...

MCG_C4 field descriptions (continued)

Field	Description
	The DRS bits select the frequency range for the FLL output, DCOOUT. When the LP bit is set, writes to the DRS bits are ignored. The DRST read field indicates the current frequency range for DCOOUT. The DRST field does not update immediately after a write to the DRS field due to internal synchronization between clock domains. See the DCO Frequency Range table for more details. 00 Encoding 0 — Low range (reset default). 01 Encoding 1 — Mid range. 10 Encoding 2 — Mid-high range. 11 Encoding 3 — High range.
4–1 FCTRIM	Fast Internal Reference Clock Trim Setting FCTRIM [1] controls the fast internal reference clock frequency by controlling the fast internal reference clock period. The FCTRIM bits are binary weighted, that is, bit 1 adjusts twice as much as bit 0. Increasing the binary value increases the period, and decreasing the value decreases the period. If an FCTRIM[3:0] value stored in nonvolatile memory is to be used, it is your responsibility to copy that value from the nonvolatile memory location to this register.
0 SCFTRIM	Slow Internal Reference Clock Fine Trim SCFTRIM [2] controls the smallest adjustment of the slow internal reference clock frequency. Setting SCFTRIM increases the period and clearing SCFTRIM decreases the period by the smallest amount possible. If an SCFTRIM value stored in nonvolatile memory is to be used, it is your responsibility to copy that value from the nonvolatile memory location to this bit.

1. A value for FCTRIM is loaded during reset from a factory programmed location .
2. A value for SCFTRIM is loaded during reset from a factory programmed location .

24.3.5 MCG Control 5 Register (MCG_C5)

Address: MCG_C5 is 4006_4000h base + 4h offset = 4006_4004h

Bit	7	6	5	4	3	2	1	0
Read	0	PLLCLKEN0	PLLSTEN0	PRDIV0				
Write	▨							
Reset	0	0	0	0	0	0	0	0

MCG_C5 field descriptions

Field	Description
7 Reserved	This read-only field is reserved and always has the value zero.
6 PLLCLKEN0	PLL Clock Enable Enables the PLL independent of PLLS and enables the PLL clock for use as MCGPLLCLK. (PRDIV 0 needs to be programmed to the correct divider to generate a PLL reference clock in the range of 2 - 4 MHz range prior to setting the PLLCLKEN 0 bit). Setting PLLCLKEN 0 will enable the external oscillator if not already enabled. Whenever the PLL is being enabled by means of the PLLCLKEN 0 bit, and the external oscillator is being used as the reference clock, the OSCINIT 0 bit should be checked to make sure it is set.

Table continues on the next page...

MCG_C5 field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>0 MCGPLLCLK is inactive.
1 MCGPLLCLK is active.</td></tr>
<tr><td>5
PLLSTEN0</td><td>PLL Stop Enable

Enables the PLL Clock during Normal Stop. In Low Power Stop mode, the PLL clock gets disabled even if PLLSTEN 0 =1. All other power modes, PLLSTEN 0 bit has no affect and does not enable the PLL Clock to run if it is written to 1.

0 MCGPLLCLK is disabled in any of the Stop modes.
1 MCGPLLCLK is enabled if system is in Normal Stop mode.</td></tr>
<tr><td>4–0
PRDIV0</td><td>PLL External Reference Divider

Selects the amount to divide down the external reference clock for the PLL. The resulting frequency must be in the range of 2 MHz to 4 MHz. After the PLL is enabled (by setting either PLLCLKEN 0 or PLLS), the PRDIV 0 value must not be changed when LOCK 0 is zero.

Table 24-7. PLL External Reference Divide Factor
<table>
<tr><th>PRDIV 0</th><th>Divide Factor</th><th></th><th>PRDIV 0</th><th>Divide Factor</th><th></th><th>PRDIV 0</th><th>Divide Factor</th><th></th><th>PRDIV 0</th><th>Divide Factor</th></tr>
<tr><td>00000</td><td>1</td><td></td><td>01000</td><td>9</td><td></td><td>10000</td><td>17</td><td></td><td>11000</td><td>25</td></tr>
<tr><td>00001</td><td>2</td><td></td><td>01001</td><td>10</td><td></td><td>10001</td><td>18</td><td></td><td>11001</td><td>Reserved</td></tr>
<tr><td>00010</td><td>3</td><td></td><td>01010</td><td>11</td><td></td><td>10010</td><td>19</td><td></td><td>11010</td><td>Reserved</td></tr>
<tr><td>00011</td><td>4</td><td></td><td>01011</td><td>12</td><td></td><td>10011</td><td>20</td><td></td><td>11011</td><td>Reserved</td></tr>
<tr><td>00100</td><td>5</td><td></td><td>01100</td><td>13</td><td></td><td>10100</td><td>21</td><td></td><td>11100</td><td>Reserved</td></tr>
<tr><td>00101</td><td>6</td><td></td><td>01101</td><td>14</td><td></td><td>10101</td><td>22</td><td></td><td>11101</td><td>Reserved</td></tr>
<tr><td>00110</td><td>7</td><td></td><td>01110</td><td>15</td><td></td><td>10110</td><td>23</td><td></td><td>11110</td><td>Reserved</td></tr>
<tr><td>00111</td><td>8</td><td></td><td>01111</td><td>16</td><td></td><td>10111</td><td>24</td><td></td><td>11111</td><td>Reserved</td></tr>
</table></td></tr>
</table>

24.3.6 MCG Control 6 Register (MCG_C6)

Address: MCG_C6 is 4006_4000h base + 5h offset = 4006_4005h

Bit	7	6	5	4	3	2	1	0
Read Write	LOLIE0	PLLS	CME0	VDIV0				
Reset	0	0	0	0	0	0	0	0

MCG_C6 field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>7
LOLIE0</td><td>Loss of Lock Interrrupt Enable

Determines if an interrupt request is made following a loss of lock indication. This bit only has an effect when LOLS 0 is set.

0 No interrupt request is generated on loss of lock.
1 Generate an interrupt request on loss of lock.</td></tr>
<tr><td>6
PLLS</td><td>PLL Select

Controls whether the PLL or FLL output is selected as the MCG source when CLKS[1:0]=00. If the PLLS bit is cleared and PLLCLKEN 0 is not set, the PLL is disabled in all modes. If the PLLS is set, the FLL is disabled in all modes.

0 FLL is selected.
1 PLL is selected (PRDIV 0 need to be programmed to the correct divider to generate a PLL reference clock in the range of 2–4 MHz prior to setting the PLLS bit).</td></tr>
<tr><td>5
CME0</td><td>Clock Monitor Enable

Enables the loss of clock monitoring circuit for the OSC0 external reference mux select. The LOCRE0 bit will determine if a interrupt or a reset request is generated following a loss of OSC0 indication. The CME0 bit should only be set to a logic 1 when the MCG is in an operational mode that uses the external clock (FEE, FBE, PEE, PBE, or BLPE) . Whenever the CME0 bit is set to a logic 1, the value of the RANGE0 bits in the C2 register should not be changed. CME0 bit should be set to a logic 0 before the MCG enters any Stop mode. Otherwise, a reset request may occur while in Stop mode. CME0 should also be set to a logic 0 before entering VLPR or VLPW power modes if the MCG is in BLPE mode.

0 External clock monitor is disabled for OSC0.
1 External clock monitor is enabled for OSC0.</td></tr>
<tr><td>4–0
VDIV0</td><td>VCO 0 Divider

Selects the amount to divide the VCO output of the PLL. The VDIV 0 bits establish the multiplication factor (M) applied to the reference clock frequency. After the PLL is enabled (by setting either PLLCLKEN 0 or PLLS), the VDIV 0 value must not be changed when LOCK 0 is zero.

Table 24-9. PLL VCO Divide Factor
<table>
<tr><th>VDIV 0</th><th>Multiply Factor</th><th></th><th>VDIV 0</th><th>Multiply Factor</th><th></th><th>VDIV 0</th><th>Multiply Factor</th><th></th><th>VDIV 0</th><th>Multiply Factor</th></tr>
<tr><td>00000</td><td>24</td><td></td><td>01000</td><td>32</td><td></td><td>10000</td><td>40</td><td></td><td>11000</td><td>48</td></tr>
<tr><td>00001</td><td>25</td><td></td><td>01001</td><td>33</td><td></td><td>10001</td><td>41</td><td></td><td>11001</td><td>49</td></tr>
<tr><td>00010</td><td>26</td><td></td><td>01010</td><td>34</td><td></td><td>10010</td><td>42</td><td></td><td>11010</td><td>50</td></tr>
<tr><td>00011</td><td>27</td><td></td><td>01011</td><td>35</td><td></td><td>10011</td><td>43</td><td></td><td>11011</td><td>51</td></tr>
<tr><td>00100</td><td>28</td><td></td><td>01100</td><td>36</td><td></td><td>10100</td><td>44</td><td></td><td>11100</td><td>52</td></tr>
<tr><td>00101</td><td>29</td><td></td><td>01101</td><td>37</td><td></td><td>10101</td><td>45</td><td></td><td>11101</td><td>53</td></tr>
<tr><td>00110</td><td>30</td><td></td><td>01110</td><td>38</td><td></td><td>10110</td><td>46</td><td></td><td>11110</td><td>54</td></tr>
<tr><td>00111</td><td>31</td><td></td><td>01111</td><td>39</td><td></td><td>10111</td><td>47</td><td></td><td>11111</td><td>55</td></tr>
</table>
</td></tr>
</table>

24.3.7 MCG Status Register (MCG_S)

Address: MCG_S is 4006_4000h base + 6h offset = 4006_4006h

Bit	7	6	5	4	3	2	1	0
Read	LOLS0	LOCK0	PLLST	IREFST	CLKST		OSCINIT0	IRCST
Write								
Reset	0	0	0	1	0	0	0	0

MCG_S field descriptions

Field	Description
7 LOLS0	Loss of Lock Status This bit is a sticky bit indicating the lock status for the PLL. LOLS 0 is set if after acquiring lock, the PLL output frequency has fallen outside the lock exit frequency tolerance, D_{unl} . LOLIE 0 determines whether an interrupt request is made when LOLS 0 is set. LOLRE determines whether a reset request is made when LOLS0 is set. This bit is cleared by reset or by writing a logic 1 to it when set. Writing a logic 0 to this bit has no effect. 0 PLL has not lost lock since LOLS 0 was last cleared. 1 PLL has lost lock since LOLS 0 was last cleared.
6 LOCK0	Lock Status This bit indicates whether the PLL has acquired lock. Lock detection is disabled when not operating in either PBE or PEE mode unless PLLCLKEN 0 =1 and the MCG is not configured in BLPI or BLPE mode. While the PLL clock is locking to the desired frequency, the MCG PLL clock (MCGPLLCLK) will be gated off until the LOCK 0 bit gets asserted. If the lock status bit is set, changing the value of the PRDIV 0 [4:0] bits in the C5 register or the VDIV0[4:0] bits in the C6 register causes the lock status bit to clear and stay cleared until the PLL has reacquired lock. Loss of PLL reference clock will also cause the LOCK 0 bit to clear until PLL has reacquired lock. Entry into LLS, VLPS, or regular Stop with PLLSTEN 0 =0 also causes the lock status bit to clear and stay cleared until the Stop mode is exited and the PLL has reacquired lock. Any time the PLL is enabled and the LOCK 0 bit is cleared, the MCGPLLCLK will be gated off until the LOCK 0 bit is asserted again. 0 PLL is currently unlocked. 1 PLL is currently locked.
5 PLLST	PLL Select Status This bit indicates the clock source selected by PLLS . The PLLST bit does not update immediately after a write to the PLLS bit due to internal synchronization between clock domains. 0 Source of PLLS clock is FLL clock. 1 Source of PLLS clock is PLL clock.
4 IREFST	Internal Reference Status This bit indicates the current source for the FLL reference clock. The IREFST bit does not update immediately after a write to the IREFS bit due to internal synchronization between clock domains. 0 Source of FLL reference clock is the external reference clock. 1 Source of FLL reference clock is the internal reference clock.

Table continues on the next page...

MCG_S field descriptions (continued)

Field	Description
3–2 CLKST	Clock Mode Status These bits indicate the current clock mode. The CLKST bits do not update immediately after a write to the CLKS bits due to internal synchronization between clock domains. 00 Encoding 0 — Output of the FLL is selected (reset default). 01 Encoding 1 — Internal reference clock is selected. 10 Encoding 2 — External reference clock is selected. 11 Encoding 3 — Output of the PLL is selected.
1 OSCINIT0	OSC Initialization This bit, which resets to 0, is set to 1 after the initialization cycles of the crystal oscillator clock have completed. After being set, the bit is cleared to 0 if the OSC is subsequently disabled. See the OSC module's detailed description for more information.
0 IRCST	Internal Reference Clock Status The IRCST bit indicates the current source for the internal reference clock select clock (IRCSCLK). The IRCST bit does not update immediately after a write to the IRCS bit due to internal synchronization between clock domains. The IRCST bit will only be updated if the internal reference clock is enabled, either by the MCG being in a mode that uses the IRC or by setting the C1[IRCLKEN] bit . 0 Source of internal reference clock is the slow clock (32 kHz IRC). 1 Source of internal reference clock is the fast clock (2 MHz IRC).

24.3.8 MCG Status and Control Register (MCG_SC)

Address: MCG_SC is 4006_4000h base + 8h offset = 4006_4008h

Bit	7	6	5	4	3	2	1	0
Read	ATME	ATMS	ATMF	FLTPRSRV	FCRDIV			LOCS0
Write			▨					▨
Reset	0	0	0	0	0	0	1	0

MCG_SC field descriptions

Field	Description
7 ATME	Automatic Trim Machine Enable Enables the Auto Trim Machine to start automatically trimming the selected Internal Reference Clock. **NOTE:** ATME deasserts after the Auto Trim Machine has completed trimming all trim bits of the IRCS clock selected by the ATMS bit. Writing to C1, C3, C4, and SC registers or entering Stop mode aborts the auto trim operation and clears this bit. 0 Auto Trim Machine disabled. 1 Auto Trim Machine enabled.

Table continues on the next page...

MCG_SC field descriptions (continued)

Field	Description
6 ATMS	Automatic Trim Machine Select Selects the IRCS clock for Auto Trim Test. 0 32 kHz Internal Reference Clock selected. 1 4 MHz Internal Reference Clock selected.
5 ATMF	Automatic Trim Machine Fail Flag Fail flag for the Automatic Trim Machine (ATM). This bit asserts when the Automatic Trim Machine is enabled, ATME=1, and a write to the C1, C3, C4, and SC registers is detected or the MCG enters into any Stop mode. A write to ATMF clears the flag. 0 Automatic Trim Machine completed normally. 1 Automatic Trim Machine failed.
4 FLTPRSRV	FLL Filter Preserve Enable This bit will prevent the FLL filter values from resetting allowing the FLL output frequency to remain the same during clock mode changes where the FLL/DCO output is still valid. (Note: This requires that the FLL reference frequency to remain the same as what it was prior to the new clock mode switch. Otherwise FLL filter and frequency values will change.) 0 FLL filter and FLL frequency will reset on changes to currect clock mode. 1 Fll filter and FLL frequency retain their previous values during new clock mode change.
3–1 FCRDIV	Fast Clock Internal Reference Divider Selects the amount to divide down the fast internal reference clock. The resulting frequency will be in the range 31.25 kHz to 4 MHz (Note: Changing the divider when the Fast IRC is enabled is not supported). 000 Divide Factor is 1 001 Divide Factor is 2. 010 Divide Factor is 4. 011 Divide Factor is 8. 100 Divide Factor is 16 101 Divide Factor is 32 110 Divide Factor is 64 111 Divide Factor is 128.
0 LOCS0	OSC0 Loss of Clock Status The LOCS0 indicates when a loss of OSC0 reference clock has occurred. The LOCS0 bit only has an effect when CME0 is set. This bit is cleared by writing a logic 1 to it when set. 0 Loss of OSC0 has not occurred. 1 Loss of OSC0 has occurred.

24.3.9 MCG Auto Trim Compare Value High Register (MCG_ATCVH)

Address: MCG_ATCVH is 4006_4000h base + Ah offset = 4006_400Ah

Bit	7	6	5	4	3	2	1	0
Read Write	ATCVH							
Reset	0	0	0	0	0	0	0	0

MCG_ATCVH field descriptions

Field	Description
7–0 ATCVH	ATM Compare Value High Values are used by Auto Trim Machine to compare and adjust Internal Reference trim values during ATM SAR conversion.

24.3.10 MCG Auto Trim Compare Value Low Register (MCG_ATCVL)

Address: MCG_ATCVL is 4006_4000h base + Bh offset = 4006_400Bh

Bit	7	6	5	4	3	2	1	0
Read Write	ATCVL							
Reset	0	0	0	0	0	0	0	0

MCG_ATCVL field descriptions

Field	Description
7–0 ATCVL	ATM Compare Value Low Values are used by Auto Trim Machine to compare and adjust Internal Reference trim values during ATM SAR conversion.

24.3.11 MCG Control 7 Register (MCG_C7)

Address: MCG_C7 is 4006_4000h base + Ch offset = 4006_400Ch

Bit	7	6	5	4	3	2	1	0
Read	0							OSCSEL
Write	▨							
Reset	0	0	0	0	0	0	0	0

MCG_C7 field descriptions

Field	Description
7–1 Reserved	This read-only field is reserved and always has the value zero.
0 OSCSEL	MCG OSC Clock Select Selects the MCG FLL external reference clock 0 Selects System Oscillator (OSCCLK). 1 Selects 32 kHz RTC Oscillator.

24.3.12 MCG Control 8 Register (MCG_C8)

Address: MCG_C8 is 4006_4000h base + Dh offset = 4006_400Dh

Bit	7	6	5	4	3	2	1	0
Read	LOCRE1	LOLRE	CME1	0				LOCS1
Write								
Reset	1	0	0	0	0	0	0	0

MCG_C8 field descriptions

Field	Description
7 LOCRE1	Loss of Clock Reset Enable Determines if a interrupt or a reset request is made following a loss of RTC external reference clock. The LOCRE1 only has an affect when CME1 is set. 0 Interrupt request is generated on a loss of RTC external reference clock. 1 Generate a reset request on a loss of RTC external reference clock
6 LOLRE	0 Interrupt request is generated on a PLL loss of lock indication. The PLL loss of lock interrupt enable bit must also be set to generate the interrupt request. 1 Generate a reset request on a PLL loss of lock indication.
5 CME1	Clock Monitor Enable1 Enables the loss of clock monitoring circuit for the output of the RTC external reference clock. The LOCRE1 bit will determine whether an interrupt or a reset request is generated following a loss of RTC clock indication. The CME1 bit should be set to a logic 1 when the MCG is in an operational mode that uses the RTC as its external reference clock or if the RTC is operational. CME1 bit must be set to a logic 0 before the MCG enters any Stop mode. Otherwise, a reset request may occur when in Stop mode. CME1 should also be set to a logic 0 before entering VLPR or VLPW power modes. 0 External clock monitor is disabled for RTC clock. 1 External clock monitor is enabled for RTC clock.
4–1 Reserved	This read-only field is reserved and always has the value zero.
0 LOCS1	RTC Loss of Clock Status This bit indicates when a loss of clock has occurred. This bit is cleared by writing a logic 1 to it when set.

Table continues on the next page...

MCG_C8 field descriptions (continued)

Field	Description
	0 Loss of RTC has not occur. 1 Loss of RTC has occur

24.4 Functional Description

24.4.1 MCG mode state diagram

The nine states of the MCG are shown in the following figure and are described in Table 24-16. The arrows indicate the permitted MCG mode transitions.

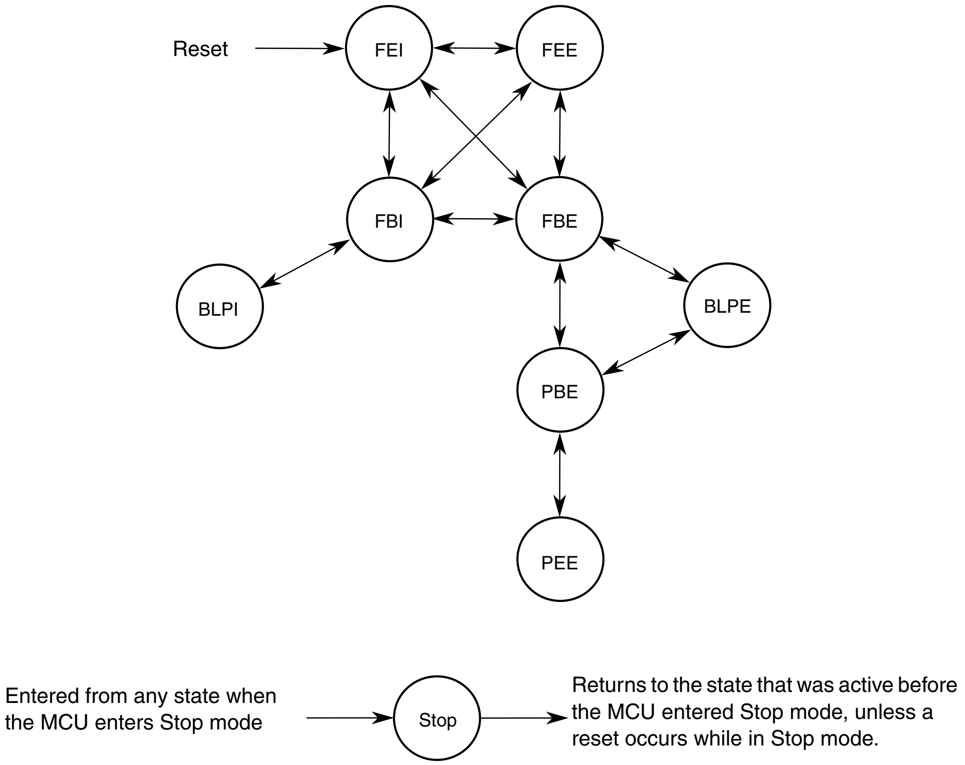

Reset
FEI
FEE
FBI
FBE
BLPI
BLPE
PBE
PEE
Entered from any state when the MCU enters Stop mode
Stop
Returns to the state that was active before the MCU entered Stop mode, unless a reset occurs while in Stop mode.

Figure 24-14. MCG mode state diagram

NOTE

- During exits from LLS or VLPS when the MCG is in PEE mode, the MCG will reset to PBE clock mode and the C1[CLKS] and S[CLKST] will automatically be set to 2’b10.
- If entering Normal Stop mode when the MCG is in PEE mode with C5[PLLSTEN]=0, the MCG will reset to PBE clock mode and C1[CLKS] and S[CLKST] will automatically be set to 2’b10.

24.4.1.1 MCG modes of operation

The MCG operates in one of the following modes.

Note

The MCG restricts transitions between modes. For the permitted transitions, see Figure 24-14.

Table 24-16. MCG modes of operation

Mode	Description
FLL Engaged Internal (FEI)	FLL engaged internal (FEI) is the default mode of operation and is entered when all the following condtions occur: • C1[CLKS] bits are written to 00 • C1[IREFS] bit is written to 1 • C6[PLLS] bit is written to 0 In FEI mode, MCGOUTCLK is derived from the FLL clock (DCOCLK) that is controlled by the 32 kHz Internal Reference Clock (IRC). The FLL loop will lock the DCO frequency to the FLL factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the internal reference frequency. See the C4[DMX32] bit description for more details. In FEI mode, the PLL is disabled in a low-power state unless C5[PLLCLKEN0] is set.

Table continues on the next page...

Table 24-16. MCG modes of operation (continued)

Mode	Description
FLL Engaged External (FEE)	FLL engaged external (FEE) mode is entered when all the following conditions occur: • C1[CLKS] bits are written to 00 • C1[IREFS] bit is written to 0 • C1[FRDIV] must be written to divide external reference clock to be within the range of 31.25 kHz to 39.0625 kHz • C6[PLLS] bit is written to 0 In FEE mode, MCGOUTCLK is derived from the FLL clock (DCOCLK) that is controlled by the external reference clock. The FLL loop will lock the DCO frequency to the FLL factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the external reference frequency, as specified by C1[FRDIV] and C2[RANGE0]. See the C4[DMX32] bit description for more details. In FEE mode, the PLL is disabled in a low-power state unless C5[PLLCLKEN0] is set.
FLL Bypassed Internal (FBI)	FLL bypassed internal (FBI) mode is entered when all the following conditions occur: • C1[CLKS] bits are written to 01 • C1[IREFS] bit is written to 1 • C6[PLLS] is written to 0 • C2[LP] is written to 0 In FBI mode, the MCGOUTCLK is derived either from the slow (32 kHz IRC) or fast (2 MHz IRC) internal reference clock, as selected by the C2[IRCS] bit. The FLL is operational but its output is not used. This mode is useful to allow the FLL to acquire its target frequency while the MCGOUTCLK is driven from the C2[IRCS] selected internal reference clock. The FLL clock (DCOCLK) is controlled by the slow internal reference clock, and the DCO clock frequency locks to a multiplication factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the internal reference frequency. See the C4[DMX32] bit description for more details. In FBI mode, the PLL is disabled in a low-power state unless C5[PLLCLKEN0] is set.
FLL Bypassed External (FBE)	FLL bypassed external (FBE) mode is entered when all the following conditions occur: • C1[CLKS] bits are written to 10 • C1[IREFS] bit is written to 0 • C1[FRDIV] must be written to divide external reference clock to be within the range of 31.25 kHz to 39.0625 kHz. • C6[PLLS] bit is written to 0 • C2[LP] is written to 0 In FBE mode, the MCGOUTCLK is derived from the OSCSEL external reference clock. The FLL is operational but its output is not used. This mode is useful to allow the FLL to acquire its target frequency while the MCGOUTCLK is driven from the external reference clock. The FLL clock (DCOCLK) is controlled by the external reference clock, and the DCO clock frequency locks to a multiplication factor, as selected by C4[DRST_DRS] and C4[DMX32] bits, times the divided external reference frequency. See the C4[DMX32] bit description for more details. In FBI mode the PLL is disabled in a low-power state unless C5[PLLCLKEN0] is set.

Table continues on the next page...

Table 24-16. MCG modes of operation (continued)

Mode	Description
PLL Engaged External (PEE)	PLL Engaged External (PEE) mode is entered when all the following conditions occur: • C1[CLKS] bits are written to 00 • C1[IREFS] bit is written to 0 • C6[PLLS] bit is written to 1 In PEE mode, the MCGOUTCLK is derived from the PLL clock, which is controlled by the external reference clock. The PLL clock frequency locks to a multiplication factor, as specified by C6[VDIV0], times the external reference frequency, as specified by C5[PRDIV0]. The PLL's programmable reference divider must be configured to produce a valid PLL reference clock. The FLL is disabled in a low-power state.
PLL Bypassed External (PBE)	PLL Bypassed External (PBE) mode is entered when all the following conditions occur: • C1[CLKS] bits are written to 10 • C1[IREFS] bit is written to 0 • C6[PLLS] bit is written to 1 • C2[LP] bit is written to 0 In PBE mode, MCGOUTCLK is derived from the OSCSEL external reference clock; the PLL is operational, but its output clock is not used. This mode is useful to allow the PLL to acquire its target frequency while MCGOUTCLK is driven from the external reference clock. The PLL clock frequency locks to a multiplication factor, as specified by its [VDIV], times the PLL reference frequency, as specified by its [PRDIV]. In preparation for transition to PEE, the PLL's programmable reference divider must be configured to produce a valid PLL reference clock. The FLL is disabled in a low-power state.
Bypassed Low Power Internal (BLPI)[1]	Bypassed Low Power Internal (BLPI) mode is entered when all the following conditions occur: • C1[CLKS] bits are written to 01 • C1[IREFS] bit is written to 1 • C6[PLLS] bit is written to 0 • C2[LP] bit is written to 1 In BLPI mode, MCGOUTCLK is derived from the internal reference clock. The FLL is disabled and PLL is disabled even if the C5[PLLCLKEN0] is set to 1.
Bypassed Low Power External (BLPE)	Bypassed Low Power External (BLPE) mode is entered when all the following conditions occur: • C1[CLKS] bits are written to 10 • C1[IREFS] bit is written to 0 • C2[LP] bit is written to 1 In BLPE mode, MCGOUTCLK is derived from the OSCSEL external reference clock. The FLL is disabled and PLL is disabled even if the C5[PLLCLKEN0] is set to 1.

Table continues on the next page...

Table 24-16. MCG modes of operation (continued)

Mode	Description
Stop	Entered whenever the MCU enters a Stop state. The power modes are chip specific. For power mode assignments, see the chapter that describes how modules are configured and MCG behavior during Stop recovery. Entering Stop mode, the FLL is disabled, and all MCG clock signals are static except in the following case: MCGPLLCLK is active in Normal Stop mode when PLLSTEN=1 MCGIRCLK is active in Stop mode when all the following conditions become true: • C1[IRCLKEN] = 1 • C1[IREFSTEN] = 1 **NOTE:** • When entering Low Power Stop modes (LLS or VLPS) from PEE mode, on exit the MCG clock mode is forced to PBE clock mode . C1[CLKS] and S[CLKST] will be configured to 2'b10 and S[LOCK0] bit will be cleared without setting S[LOLS0]. • When entering Normal Stop mode from PEE mode and if C5[PLLSTEN0]=0, on exit the MCG clock mode is forced to PBE mode, the C1[CLKS] and S[CLKST] will be configured to 2'b10 and S[LOCK0] bit will clear without setting S[LOLS0]. If C5[PLLSTEN0]=1, the S[LOCK0] bit will not get cleared and on exit the MCG will continue to run in PEE mode.

1. If entering VLPR mode, MCG has to be configured and enter BLPE mode or BLPI mode with the 4 MHz IRC clock selected (C2[IRCS]=1). After it enters VLPR mode, writes to any of the MCG control registers that can cause an MCG clock mode switch to a non low power clock mode must be avoided.

NOTE

For the chip-specific modes of operation, see the power management chapter of this MCU.

24.4.1.2 MCG mode switching

The C1[IREFS] bit can be changed at any time, but the actual switch to the newly selected reference clocks is shown by the S[IREFST] bit. When switching between engaged internal and engaged external modes, the FLL will begin locking again after the switch is completed.

The C1[CLKS] bits can also be changed at any time, but the actual switch to the newly selected clock is shown by the S[CLKST] bits. If the newly selected clock is not available, the previous clock will remain selected.

The C4[DRST_DRS] write bits can be changed at any time except when C2[LP] bit is 1. If the C4[DRST_DRS] write bits are changed while in FLL engaged internal (FEI) or FLL engaged external (FEE), the MCGOUTCLK will switch to the new selected DCO range within three clocks of the selected DCO clock. After switching to the new DCO, the FLL remains unlocked for several reference cycles. DCO startup time is equal to the FLL acquisition time. After the selected DCO startup time is over, the FLL is locked. The completion of the switch is shown by the C4[DRST_DRS] read bits.

24.4.2 Low Power Bit Usage

The C2[LP] bit is provided to allow the FLL or PLL to be disabled and thus conserve power when these systems are not being used. The C4[DRST_DRS] can not be written while C2[LP] bit is 1. However, in some applications, it may be desirable to enable the FLL or PLL and allow it to lock for maximum accuracy before switching to an engaged mode. Do this by writing C2[LP] to 0.

24.4.3 MCG Internal Reference Clocks

This module supports two internal reference clocks with nominal frequencies of 32 kHz (slow IRC) and 4 MHz (fast IRC). The fast IRC frequency can be divided down by programming of the FCRDIV to produce a frequency range of 32 kHz to 4 MHz.

24.4.3.1 MCG Internal Reference Clock

The MCG Internal Reference Clock (MCGIRCLK) provides a clock source for other on-chip peripherals and is enabled when C1[IRCLKEN]=1. When enabled, MCGIRCLK is driven by either the fast internal reference clock (4 MHz IRC which can be divided down by the FRDIV factors) or the slow internal reference clock (32 kHz IRC). The IRCS clock frequency can be re-targeted by trimming the period of its IRCS selected internal reference clock. This can be done by writing a new trim value to the C3[SCTRIM]:C4[SCFTRIM] bits when the slow IRC clock is selected or by writing a new trim value to the C4[FCTRIM] bits when the fast IRC clock is selected. The internal reference clock period is proportional to the trim value written. C3[SCTRIM]:C4[SCFTRIM] (if C2[IRCS]=0) and C4[FCTRIM] (if C2[IRCS]=1) bits affect the MCGOUTCLK frequency if the MCG is in FBI or BLPI modes. C3[SCTRIM]:C4[SCFTRIM] (if C2[IRCS]=0) bits also affect the MCGOUTCLK frequency if the MCG is in FEI mode.

Additionally, this clock can be enabled in Stop mode by setting C1[IRCLKEN] and C1[IREFSTEN], otherwise this clock is disabled in Stop mode.

24.4.4 External Reference Clock

The MCG module can support an external reference clock in all modes. See the device datasheet for external reference frequency range. When C1[IREFS] is set, the external reference clock will not be used by the FLL or PLL. In these modes, the frequency can be equal to the maximum frequency the chip-level timing specifications will support.

If any of the CME bits are asserted the slow internal reference clock is enabled along with the enabled external clock monitor. For the case when C6[CME0]=1, a loss of clock is detected if the OSC0 external reference falls below a minimum frequency (f_{loc_high} or f_{loc_low} depending on C2[RANGE0]). For the case when C8[CME1]=1, a loss of clock is detected if the RTC external reference falls below a minimum frequency (f_{loc_low}).

All clock monitors must be disabled before VLPR or VLPW power modes are entered.

Upon detect of a loss of clock event, the MCU generates a system reset if the respective LOCRE bit is set. Otherwise the MCG sets the respective LOCS bit and the MCG generates a LOCS interrupt request. In the case where a OSC0 loss of clock is detected, the PLL LOCK status bit is cleared if the OSC clock that is lost was selected as the PLL reference clock.

24.4.5 MCG Fixed frequency clock

The MCG Fixed Frequency Clock (MCGFFCLK) provides a fixed frequency clock source for other on-chip peripherals; see the block diagram. This clock is driven by either the slow clock from the internal reference clock generator or the external reference clock from the Crystal Oscillator, divided by the FLL reference clock divider. The source of MCGFFCLK is selected by C1[IREFS].

This clock is synchronized to the peripheral bus clock and is valid only when its frequency is not more than 1/8 of the MCGOUTCLK frequency. When it is not valid, it is disabled and held high. The MCGFFCLK is not available when the MCG is in BLPI mode. This clock is also disabled in Stop mode. The FLL reference clock must be set within the valid frequency range for the MCGFFCLK.

24.4.6 MCG PLL clock

The MCG PLL Clock (MCGPLLCLK) is available depending on the device's configuration of the MCG module. For more details, see the clock distribution chapter of this MCU. The MCGPLLCLK is prevented from coming out of the MCG until it is enabled and S[LOCK0] is set.

24.4.7 MCG Auto TRIM (ATM)

The MCG Auto Trim (ATM) is a MCG feature that when enabled, it configures the MCG hardware to automatically trim the MCG Internal Reference Clocks using an external clock as a reference. The selection between which MCG IRC clock gets tested and enabled is controlled by the ATC[ATMS] control bit (ATC[ATMS]=0 selects the 32 kHz IRC and ATC[ATMS]=1 selects the 4 MHz IRC). If 4 MHz IRC is selected for the ATM, a divide by 128 is enabled to divide down the 4 MHz IRC to a range of 31.250 kHz.

When MCG ATM is enabled by writing ATC[ATME] bit to 1, The ATM machine will start auto trimming the selected IRC clock. During the autotrim process, ATC[ATME] will remain asserted and will deassert after ATM is completed or an abort occurs. The MCG ATM is aborted if a write to any of the following control registers is detected : C1, C3, C4, or ATC or if Stop mode is entered. If an abort occurs, ATC[ATMF] fail flag is asserted.

The ATM machine uses the bus clock as the external reference clock to perform the IRC auto-trim. Therefore, it is required that the MCG is configured in a clock mode where the reference clock used to generate the system clock is the external reference clock such as FBE clock mode. The MCG must not be configured in a clock mode where selected IRC ATM clock is used to generate the system clock. The bus clock is also required to be running with in the range of 8–16 MHz.

To perform the ATM on the selected IRC, the ATM machine uses the successive approximation technique to adjust the IRC trim bits to generate the desired IRC trimmed frequency. The ATM SARs each of the ATM IRC trim bits starting with the MSB. For each trim bit test, the ATM uses a pulse that is generated by the ATM selected IRC clock to enable a counter that counts number of ATM external clocks. At end of each trim bit, the ATM external counter value is compared to the ATCV[15:0] register value. Based on the comparison result, the ATM trim bit under test will get cleared or stay asserted. This is done until all trim bits have been tested by ATM SAR machine.

Before the ATM can be enabled, the ATM expected count needs to be derived and stored into the ATCV register. The ATCV expected count is derived based on the required target Internal Reference Clock (IRC) frequency, and the frequency of the external reference clock using the following formula:

$$\text{ATCV Expected Count Value} = 21*(Fe/Fr)$$

- Fr = Target Internal Reference Clock (IRC) Trimmed Frequency
- Fe = External Clock Frequency

If the auto trim is being performed on the 4 MHz IRC, the calculated expected count value must be multiplied by 128 before storing it in the ATCV register. Therefore, the ATCV Expected Count Value for trimming the 4 MHz IRC is calculated using the following formula.

$$\text{Expected Count Value} = (Fe/Fr)*21*(128)$$

24.5 Initialization / Application information

This section describes how to initialize and configure the MCG module in an application. The following sections include examples on how to initialize the MCG and properly switch between the various available modes.

24.5.1 MCG module initialization sequence

The MCG comes out of reset configured for FEI mode. The internal reference will stabilize in $t_{irefsts}$ microseconds before the FLL can acquire lock. As soon as the internal reference is stable, the FLL will acquire lock in $t_{fll\_acquire}$ milliseconds.

24.5.1.1 Initializing the MCG

Because the MCG comes out of reset in FEI mode, the only MCG modes that can be directly switched to upon reset are FEE, FBE, and FBI modes (see Figure 24-14). Reaching any of the other modes requires first configuring the MCG for one of these three intermediate modes. Care must be taken to check relevant status bits in the MCG status register reflecting all configuration changes within each mode.

To change from FEI mode to FEE or FBE modes, follow this procedure:

1. Enable the external clock source by setting the appropriate bits in C2 register.

2. Write to C1 register to select the clock mode.
    - If entering FEE mode, set C1[FRDIV] appropriately, clear the C1[IREFS] bit to switch to the external reference, and leave the C1[CLKS] bits at 2'b00 so that the output of the FLL is selected as the system clock source.
    - If entering FBE, clear the C1[IREFS] bit to switch to the external reference and change the C1[CLKS] bits to 2'b10 so that the external reference clock is selected as the system clock source. The C1[FRDIV] bits should also be set appropriately here according to the external reference frequency to keep the FLL reference clock in the range of 31.25 kHz to 39.0625 kHz. Although the FLL is bypassed, it is still on in FBE mode.
    - The internal reference can optionally be kept running by setting the C1[IRCLKEN] bit. This is useful if the application will switch back and forth between internal and external modes. For minimum power consumption, leave the internal reference disabled while in an external clock mode.
3. Once the proper configuration bits have been set, wait for the affected bits in the MCG status register to be changed appropriately, reflecting that the MCG has moved into the proper mode.
    - If the MCG is in FEE, FBE, PEE, PBE, or BLPE mode, and C2[EREFS0] was also set in step 1, wait here for S[OSCINIT0] bit to become set indicating that the external clock source has finished its initialization cycles and stabilized.
    - If in FEE mode, check to make sure the S[IREFST] bit is cleared before moving on.
    - If in FBE mode, check to make sure the S[IREFST] bit is cleared and S[CLKST] bits have changed to 2'b10 indicating the external reference clock has been appropriately selected. Although the FLL is bypassed, it is still on in FBE mode.
4. Write to the C4 register to determine the DCO output (MCGFLLCLK) frequency range.
    - By default, with C4[DMX32] cleared to 0, the FLL multiplier for the DCO output is 640. For greater flexibility, if a mid-low-range FLL multiplier of 1280 is desired instead, set C4[DRST_DRS] bits to 2'b01 for a DCO output frequency of 40 MHz. If a mid high-range FLL multiplier of 1920 is desired instead, set the C4[DRST_DRS] bits to 2'b10 for a DCO output frequency of 60 MHz. If a high-range FLL multiplier of 2560 is desired instead, set the C4[DRST_DRS] bits to 2'b11 for a DCO output frequency of 80 MHz.

   - When using a 32.768 kHz external reference, if the maximum low-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b00 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 732 will be 24 MHz.
   - When using a 32.768 kHz external reference, if the maximum mid-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b01 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 1464 will be 48 MHz.
   - When using a 32.768 kHz external reference, if the maximum mid high-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b10 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 2197 will be 72 MHz.
   - When using a 32.768 kHz external reference, if the maximum high-range DCO frequency that can be achieved with a 32.768 kHz reference is desired, set C4[DRST_DRS] bits to 2'b11 and set C4[DMX32] bit to 1. The resulting DCO output (MCGOUTCLK) frequency with the new multiplier of 2929 will be 96 MHz.

5. Wait for the FLL lock time to guarantee FLL is running at new C4[DRST_DRS] and C4[DMX32] programmed frequency.

To change from FEI clock mode to FBI clock mode, follow this procedure:

1. Change C1[CLKS] bits in C1 register to 2'b01 so that the internal reference clock is selected as the system clock source.
2. Wait for S[CLKST] bits in the MCG status register to change to 2'b01, indicating that the internal reference clock has been appropriately selected.
3. Write to the C2 register to determine the IRCS output (IRCSCLK) frequency range.
   - By default, with C2[IRCS] cleared to 0, the IRCS selected output clock is the slow internal reference clock (32 kHz IRC). If the faster IRC is desired, set C2[IRCS] bit to 1 for a IRCS clock derived from the 4 MHz IRC source.

24.5.2 Using a 32.768 kHz reference

In FEE and FBE modes, if using a 32.768 kHz external reference, at the default FLL multiplication factor of 640, the DCO output (MCGFLLCLK) frequency is 20.97 MHz at low-range. If C4[DRST_DRS] bits are set to 2'b01, the multiplication factor is doubled to 1280, and the resulting DCO output frequency is 41.94 MHz at mid-low-range. If C4[DRST_DRS] bits are set to 2'b10, the multiplication factor is set to 1920, and the resulting DCO output frequency is 62.91 MHz at mid high-range. If C4[DRST_DRS] bits are set to 2'b11, the multiplication factor is set to 2560, and the resulting DCO output frequency is 83.89 MHz at high-range.

In FBI and FEI modes, setting C4[DMX32] bit is not recommended. If the internal reference is trimmed to a frequency above 32.768 kHz, the greater FLL multiplication factor could potentially push the microcontroller system clock out of specification and damage the part.

24.5.3 MCG mode switching

When switching between operational modes of the MCG, certain configuration bits must be changed in order to properly move from one mode to another. Each time any of these bits are changed (C6[PLLS], C1[IREFS], C1[CLKS], C2[IRCS], or C2[EREFS0]), the corresponding bits in the MCG status register (PLLST, IREFST, CLKST, IRCST, or OSCINIT) must be checked before moving on in the application software.

Additionally, care must be taken to ensure that the reference clock divider (C1[FRDIV] and C5[PRDIV0]) is set properly for the mode being switched to. For instance, in PEE mode, if using a 4 MHz crystal, C5[PRDIV0] must be set to 5'b000 (divide-by-1) or 5'b001 (divide -by-2) to divide the external reference down to the required frequency between 2 and 4 MHz.

In FBE, FEE, FBI, and FEI modes, at any time, the application can switch the FLL multiplication factor between 640, 1280, 1920, and 2560 with C4[DRST_DRS] bits. Writes to C4[DRST_DRS] bits will be ignored if C2[LP]=1.

The table below shows MCGOUTCLK frequency calculations using C1[FRDIV], C5[PRDIV0], and C6[VDIV0] settings for each clock mode.

Table 24-17. MCGOUTCLK Frequency Calculation Options

Clock Mode	$f_{MCGOUTCLK}$[1]	Note
FEI (FLL engaged internal)	(f_{int} * F)	Typical $f_{MCGOUTCLK}$ = 20.97 MHz immediately after reset.

Table continues on the next page...

Table 24-17. MCGOUTCLK Frequency Calculation Options (continued)

Clock Mode	$f_{MCGOUTCLK}$[1]	Note
FEE (FLL engaged external)	$(f_{ext} / FLL\_R) * F$	f_{ext} / FLL_R must be specified for f_{fll_ref} in the appropriate device Data Sheet
FBE (FLL bypassed external)	f_{ext}	f_{ext} / FLL_R must be specified for f_{fll_ref} in the appropriate device Data Sheet
FBI (FLL bypassed internal)	f_{int}	Typical f_{int} = 32 kHz
PEE (PLL engaged external)	$(f_{ext} / PLL\_R) * M$	f_{ext} / PLL_R must be in the range specified for f_{pll_ref} in the appropriate device Data Sheet
PBE (PLL bypassed external)	f_{ext}	f_{ext} / PLL_R must be in the range specified for f_{pll_ref} in the appropriate device Data Sheet
BLPI (Bypassed low power internal)	f_{int}	
BLPE (Bypassed low power external)	f_{ext}	

1. FLL_R is the reference divider selected by the C1[FRDIV] bits, PLL_R is the reference divider selected by C5[PRDIV0] bits, F is the FLL factor selected by C4[DRST_DRS] and C4[DMX32] bits, and M is the multiplier selected by C6[VDIV0] bits.

This section will include three mode switching examples using an 4 MHz external crystal. If using an external clock source less than 2 MHz, the MCG must not be configured for any of the PLL modes (PEE and PBE).

24.5.3.1 Example 1: Moving from FEI to PEE mode: External Crystal = 4 MHz, MCGOUTCLK frequency = 48 MHz

In this example, the MCG will move through the proper operational modes from FEI to PEE to achieve 48 MHz MCGOUTCLK frequency from 4 MHz external crystal reference. First, the code sequence will be described. Then there is a flowchart that illustrates the sequence.

1. First, FEI must transition to FBE mode:
   a. C2 = 0x1C
      - C2[RANGE0] set to 2'b01 because the frequency of 4 MHz is within the high frequency range.
      - C2[HGO0] set to 1 to configure the crystal oscillator for high gain operation.
      - C2[EREFS0] set to 1, because a crystal is being used.
   b. C1 = 0x90

        - C1[CLKS] set to 2'b10 to select external reference clock as system clock source
        - C1[FRDIV] set to 3'b010, or divide-by-128 because 4 MHz / 128 = 31.25 kHz which is in the 31.25 kHz to 39.0625 kHz range required by the FLL
        - C1[IREFS] cleared to 0, selecting the external reference clock and enabling the external oscillator.
    c. Loop until S[OSCINIT0] is 1, indicating the crystal selected by C2[EREFS0] has been initialized.
    d. Loop until S[IREFST] is 0, indicating the external reference is the current source for the reference clock.
    e. Loop until S[CLKST] is 2'b10, indicating that the external reference clock is selected to feed MCGOUTCLK.
2. Then configure C5[PRDIV0] to generate correct PLL reference frequency.
    a. C5 = 0x01
        - C5[PRDIV0] set to 5'b001, or divide-by-2 resulting in a pll reference frequency of 4 MHz/2 = 2 MHz.
3. Then, FBE must transition either directly to PBE mode or first through BLPE mode and then to PBE mode:
    a. BLPE: If a transition through BLPE mode is desired, first set C2[LP] to 1.
    b. BLPE/PBE: C6 = 0x40
        - C6[PLLS] set to 1, selects the PLL. At this time, with a C1[PRDIV] value of 2'b001, the PLL reference divider is 2 (see PLL External Reference Divide Factor table), resulting in a reference frequency of 4 MHz/ 2 = 2 MHz. In BLPE mode, changing the C6[PLLS] bit only prepares the MCG for PLL usage in PBE mode.
        - C6[VDIV0] set to 5'b0000, or multiply-by-24 because 2 MHz reference * 24 = 48 MHz. In BLPE mode, the configuration of the VDIV bits does not matter because the PLL is disabled. Changing them only sets up the multiply value for PLL usage in PBE mode.
    c. BLPE: If transitioning through BLPE mode, clear C2[LP] to 0 here to switch to PBE mode.
    d. PBE: Loop until S[PLLST] is set, indicating that the current source for the PLLS clock is the PLL.

   e. PBE: Then loop until S[LOCK0] is set, indicating that the PLL has acquired lock.

4. Lastly, PBE mode transitions into PEE mode:
   a. C1 = 0x10
      - C1[CLKS] set to 2'b00 to select the output of the PLL as the system clock source.
   b. Loop until S[CLKST] are 2'b11, indicating that the PLL output is selected to feed MCGOUTCLK in the current clock mode.
      - Now, with PRDIV0 of divide-by-2, and C6[VDIV0] of multiply-by-24, MCGOUTCLK = [(4 MHz / 2) * 24] = 48 MHz.

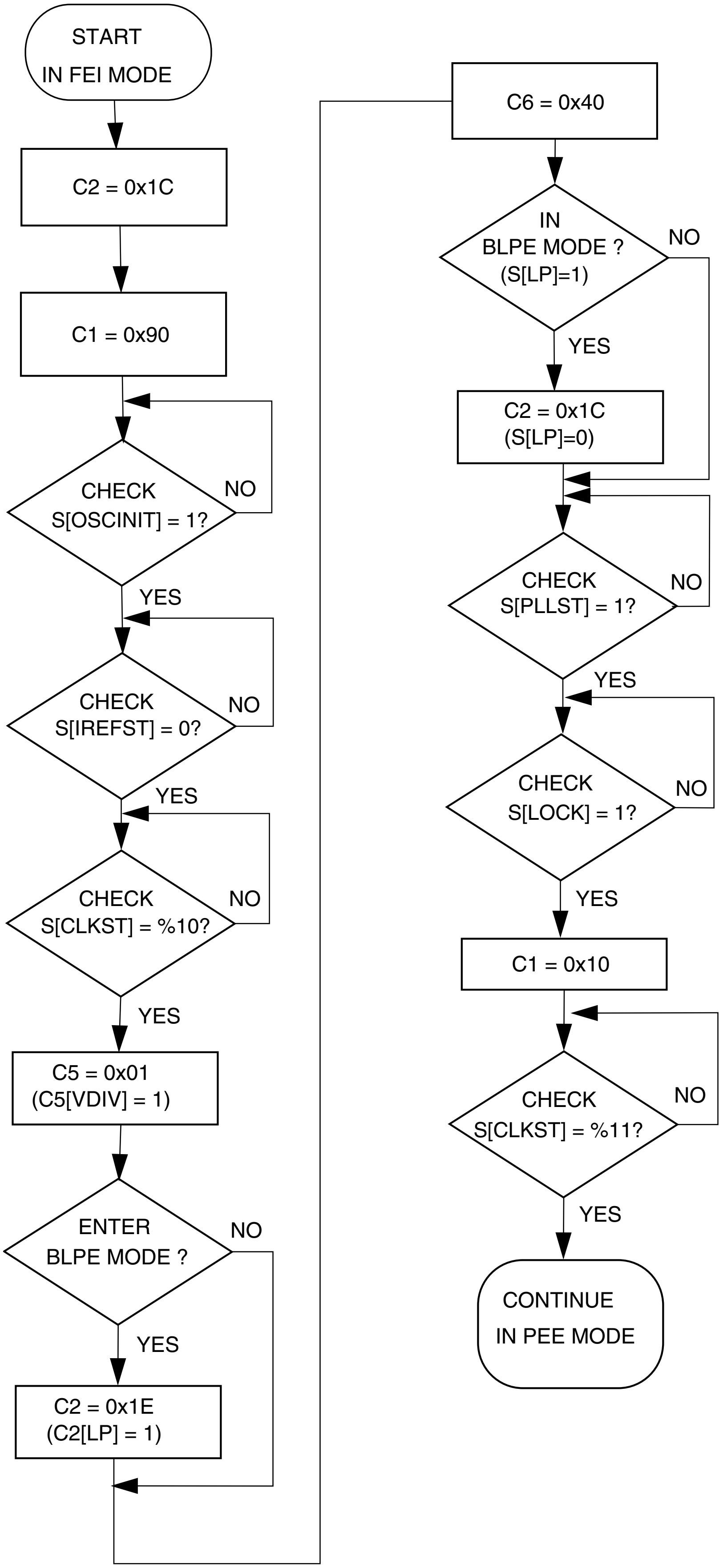

START
IN FEI MODE
C2 = 0x1C
C1 = 0x90
CHECK
S[OSCINIT] = 1?
NO
YES
CHECK
S[IREFST] = 0?
NO
YES
CHECK
S[CLKST] = %10?
NO
YES
C5 = 0x01
(C5[VDIV] = 1)
ENTER
BLPE MODE ?
NO
YES
C2 = 0x1E
(C2[LP] = 1)
C6 = 0x40
IN
BLPE MODE ?
(S[LP]=1)
NO
YES
C2 = 0x1C
(S[LP]=0)
CHECK
S[PLLST] = 1?
NO
YES
CHECK
S[LOCK] = 1?
NO
YES
C1 = 0x10
CHECK
S[CLKST] = %11?
NO
YES
CONTINUE
IN PEE MODE

Figure 24-15. Flowchart of FEI to PEE mode transition using an 4 MHz crystal

24.5.3.2 Example 2: Moving from PEE to BLPI mode: MCGOUTCLK frequency =32 kHz

In this example, the MCG will move through the proper operational modes from PEE mode with a 4 MHz crystal configured for a 48 MHz MCGOUTCLK frequency (see previous example) to BLPI mode with a 32 kHz MCGOUTCLK frequency. First, the code sequence will be described. Then there is a flowchart that illustrates the sequence.

1. First, PEE must transition to PBE mode:
   a. C1 = 0x90
      - C1[CLKS] set to 2'b10 to switch the system clock source to the external reference clock.

   b. Loop until S[CLKST] are 2'b10, indicating that the external reference clock is selected to feed MCGOUTCLK.
2. Then, PBE must transition either directly to FBE mode or first through BLPE mode and then to FBE mode:
   a. BLPE: If a transition through BLPE mode is desired, first set C2[LP] to 1.

   b. BLPE/FBE: C6 = 0x00
      - C6[PLLS] clear to 0 to select the FLL. At this time, with C1[FRDIV] value of 3'b010, the FLL divider is set to 128, resulting in a reference frequency of 4 MHz / 128 = 31.25 kHz. If C1[FRDIV] was not previously set to 3'b010 (necessary to achieve required 31.25–39.06 kHz FLL reference frequency with an 4 MHz external source frequency), it must be changed prior to clearing C6[PLLS] bit. In BLPE mode,changing this bit only prepares the MCG for FLL usage in FBE mode. With C6[PLLS] = 0, the C6[VDIV0] value does not matter.

   c. BLPE: If transitioning through BLPE mode, clear C2[LP] to 0 here to switch to FBE mode.

   d. FBE: Loop until S[PLLST] is cleared, indicating that the current source for the PLLS clock is the FLL.
3. Next, FBE mode transitions into FBI mode:
   a. C1 = 0x54
      - C1[CLKS] set to 2'b01 to switch the system clock to the internal reference clock.

        - C1[IREFS] set to 1 to select the internal reference clock as the reference clock source.
        - C1[FRDIV] remain unchanged because the reference divider does not affect the internal reference.
    b. Loop until S[IREFST] is 1, indicating the internal reference clock has been selected as the reference clock source.
    c. Loop until S[CLKST] are 2'b01, indicating that the internal reference clock is selected to feed MCGOUTCLK.
4. Lastly, FBI transitions into BLPI mode.
    a. C2 = 0x02
        - C2[LP] is 1
        - C2[RANGE0], C2[HGO0], C2[EREFS0], C1[IRCLKEN], and C1[IREFSTEN] bits are ignored when the C1[IREFS] bit is set. They can remain set, or be cleared at this point.

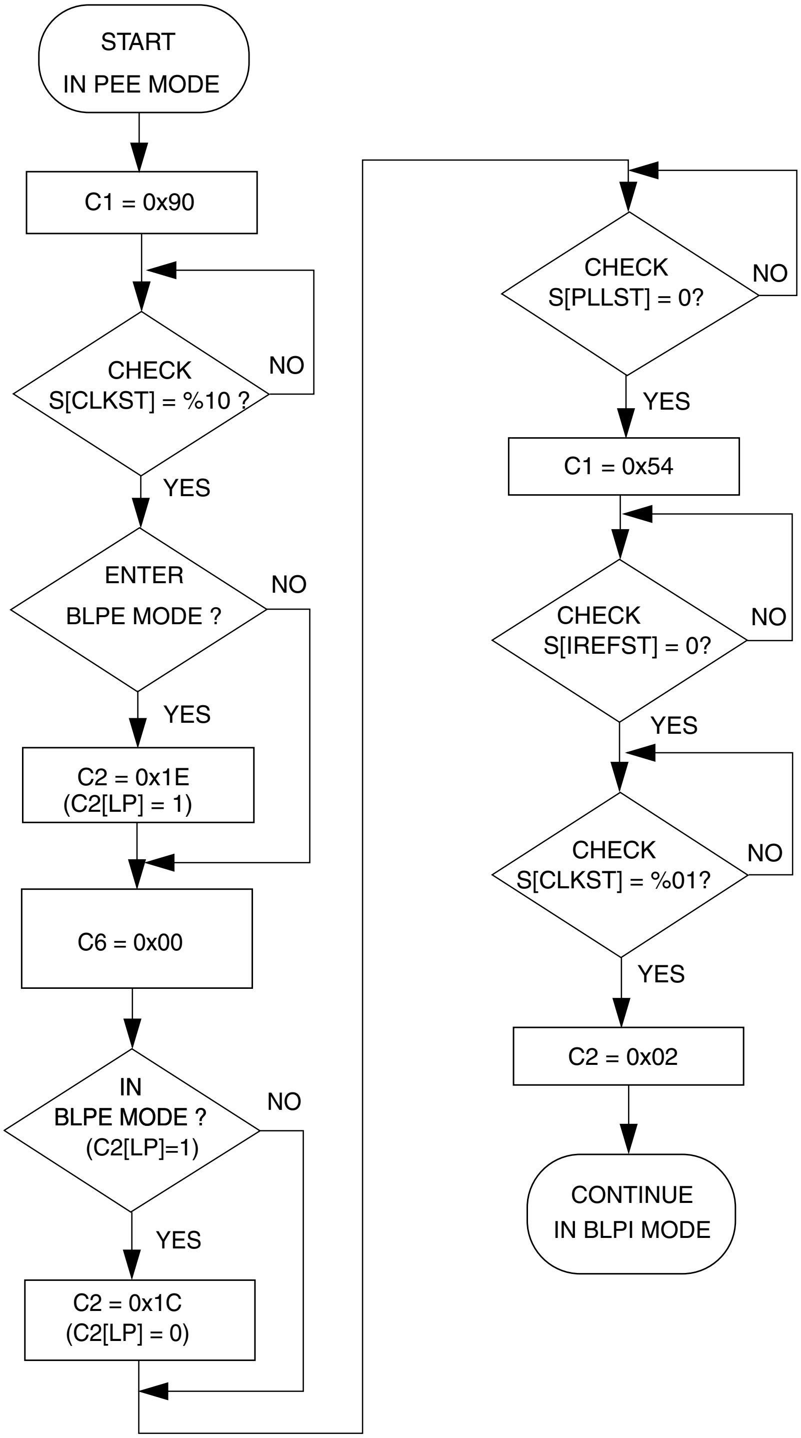

START
IN PEE MODE
C1 = 0x90
CHECK
S[CLKST] = %10 ?
NO
YES
ENTER
BLPE MODE ?
NO
YES
C2 = 0x1E
(C2[LP] = 1)
C6 = 0x00
IN
BLPE MODE ?
(C2[LP]=1)
NO
YES
C2 = 0x1C
(C2[LP] = 0)
CHECK
S[PLLST] = 0?
NO
YES
C1 = 0x54
CHECK
S[IREFST] = 0?
NO
YES
CHECK
S[CLKST] = %01?
NO
YES
C2 = 0x02
CONTINUE
IN BLPI MODE

Figure 24-16. Flowchart of PEE to BLPI mode transition using an 4 MHz crystal

24.5.3.3 Example 3: Moving from BLPI to FEE mode

In this example, the MCG will move through the proper operational modes from BLPI mode at a 32 kHz MCGOUTCLK frequency running off the internal reference clock (see previous example) to FEE mode using a 4 MHz crystal configured for a 20 MHz MCGOUTCLK frequency. First, the code sequence will be described. Then there is a flowchart that illustrates the sequence.

1. First, BLPI must transition to FBI mode.
    a. C2 = 0x00
        - C2[LP] is 0
2. Next, FBI will transition to FEE mode.
    a. C2 = 0x1C
        - C2[RANGE0] set to 2'b01 because the frequency of 4 MHz is within the high frequency range.
        - C2[HGO0] set to 1 to configure the crystal oscillator for high gain operation.
        - C2[EREFS0] set to 1, because a crystal is being used.
    b. C1 = 0x10
        - C1[CLKS] set to 2'b00 to select the output of the FLL as system clock source.
        - C1[FRDIV] remain at 3'b010, or divide-by-128 for a reference of 4 MHz / 128 = 31.25 kHz.
        - C1[IREFS] cleared to 0, selecting the external reference clock.
    c. Loop until S[OSCINIT0] is 1, indicating the crystal selected by the C2[EREFS0] bit has been initialized.
    d. Loop until S[IREFST] is 0, indicating the external reference clock is the current source for the reference clock.
    e. Loop until S[CLKST] are 2'b00, indicating that the output of the FLL is selected to feed MCGOUTCLK.
    f. Now, with a 31.25 kHz reference frequency, a fixed DCO multiplier of 640, MCGOUTCLK = 31.25 kHz * 640 / 1 = 20 MHz.
    g. At this point, by default, the C4[DRST_DRS] bits are set to 2'b00 and C4[DMX32] is cleared to 0. If the MCGOUTCLK frequency of 40 MHz is desired instead, set the C4[DRST_DRS] bits to 0x01 to switch the FLL

multiplication factor from 640 to 1280. To return the MCGOUTCLK frequency to 20 MHz, set C4[DRST_DRS] bits to 2'b00 again, and the FLL multiplication factor will switch back to 640.

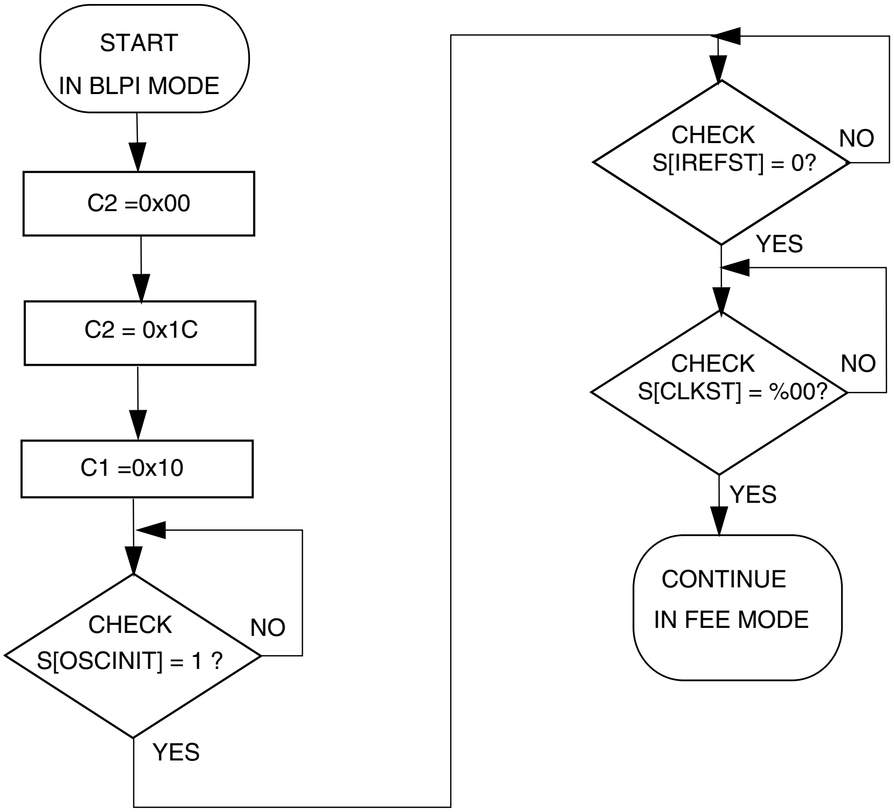

START
IN BLPI MODE
C2 =0x00
C2 = 0x1C
C1 =0x10
CHECK
S[OSCINIT] = 1 ?
NO
YES
CHECK
S[IREFST] = 0?
NO
YES
CHECK
S[CLKST] = %00?
NO
YES
CONTINUE
IN FEE MODE

Figure 24-17. Flowchart of BLPI to FEE mode transition using an 4 MHz crystal

Chapter 25
Oscillator (OSC)

25.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The OSC module is a crystal oscillator. The module, in conjunction with an external crystal or resonator, generates a reference clock for the MCU.

25.2 Features and Modes

Key features of the module are:

- Supports 32 kHz crystals (Low Range mode)
- Supports 3–8 MHz, 8–32 MHz crystals and resonators (High Range mode)
- Automatic Gain Control (AGC) to optimize power consumption in high frequency ranges 3–8 MHz, 8–32 MHz using low-power mode
- High gain option in frequency ranges: 32 kHz, 3–8 MHz, and 8–32 MHz
- Voltage and frequency filtering to guarantee clock frequency and stability
- Optionally external input bypass clock from EXTAL signal directly
- One clock for MCU clock system
- Two clocks for on-chip peripherals that can work in Stop modes

Functional Description describes the module's operation in more detail.

25.3 Block Diagram

The OSC module uses a crystal or resonator to generate three filtered oscillator clock signals. Three clocks are output from OSC module: OSCCLK for MCU system, OSCERCLK for on-chip peripherals, and OSC32KCLK. The OSCCLK can only work in run mode. OSCERCLK and OSC32KCLK can work in low power modes. For the clock source assignments, refer to the clock distribution information of this MCU.

Refer to the chip configuration chapter for the external reference clock source in this MCU.

The following figure shows the block diagram of the OSC module.

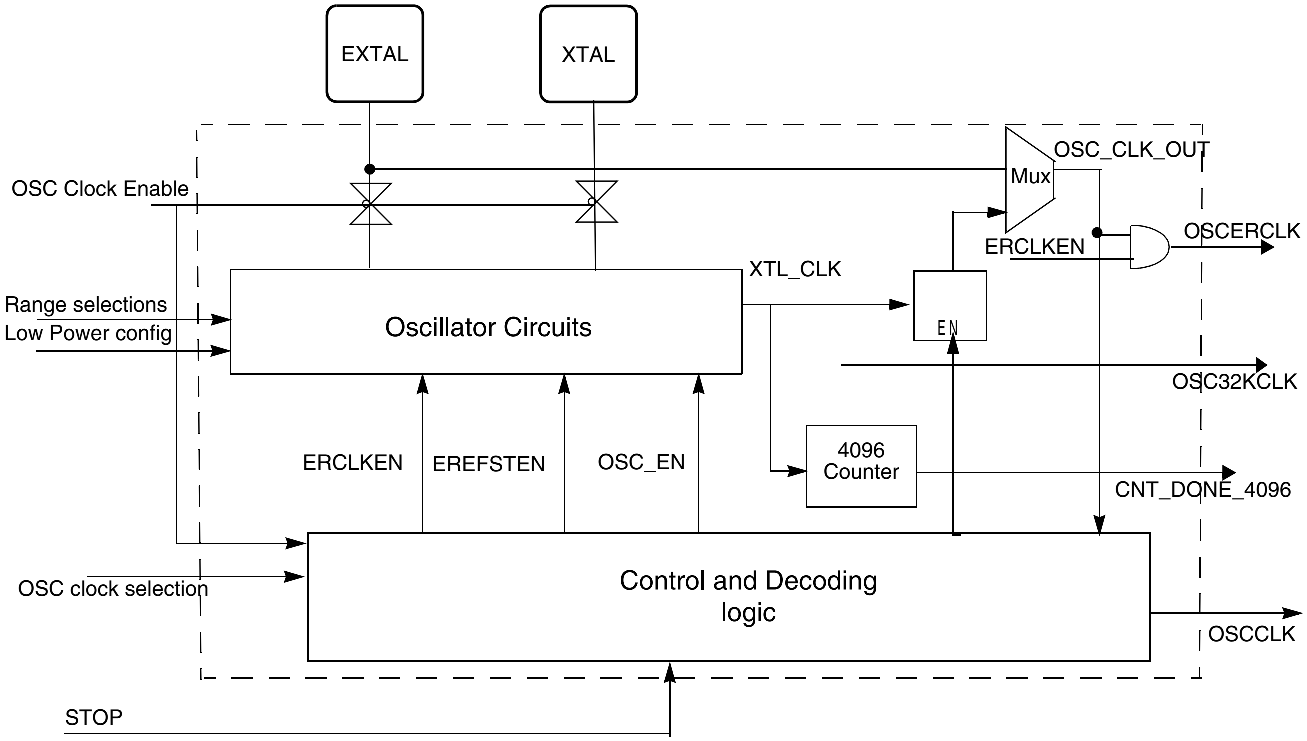

EXTAL
XTAL
OSC Clock Enable
Mux
OSC_CLK_OUT
ERCLKEN
OSCERCLK
XTL_CLK
Range selections
Low Power config
Oscillator Circuits
EN
OSC32KCLK
ERCLKEN
EREFSTEN
OSC_EN
4096
Counter
CNT_DONE_4096
OSC clock selection
Control and Decoding
logic
OSCCLK
STOP

Figure 25-1. OSC Module Block Diagram

25.4 OSC Signal Descriptions

The following table shows the user-accessible signals available for the OSC module. Refer to signal multiplexing information for this MCU for more details.

Table 25-1. OSC Signal Descriptions

Signal	Description	I/O
EXTAL	External clock/Oscillator input	I
XTAL	Oscillator output	O

25.5 External Crystal / Resonator Connections

The connections for a crystal/resonator frequency reference are shown in the following figures. When using low-frequency, low-power mode, the only external component is the crystal or ceramic resonator itself. In the other oscillator modes, load capacitors (C_x, C_y) and feedback resistor (R_F) are required. The following table shows all possible connections.

Table 25-2. External Caystal/Resonator Connections

Oscillator Mode	Connections
Low-frequency (32 kHz), low-power	Connection 1
Low-frequency (32 kHz), high-gain	Connection 2/Connection 3[1]
High-frequency (3~32 MHz), low-power	Connection 1/Connection 3[2,2]
High-frequency (3~32 MHz), high-gain	Connection 2/Connection 3[2]

1. When the load capacitors (Cx, Cy) are greater than 30 pF, use Connection 3.
2. With the low-power mode, the oscillator has the internal feedback resistor R_F. Therefore, the feedback resistor must not be externally with the Connection 3.

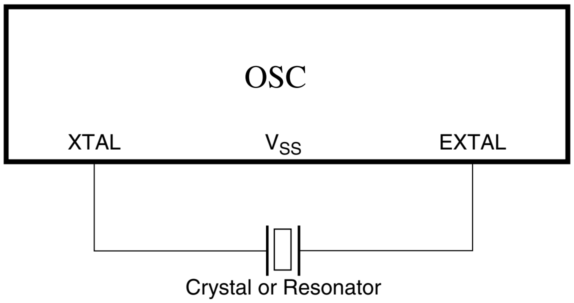


Figure 25-2. Crystal/Ceramic Resonator Connections - Connection 1

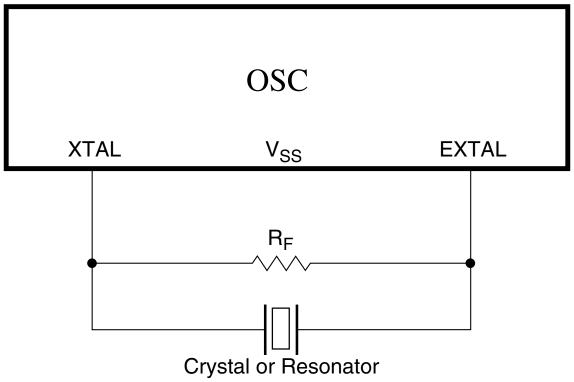

OSC
XTAL
V_{SS}
EXTAL
R_F
Crystal or Resonator

Figure 25-3. Crystal/Ceramic Resonator Connections - Connection 2

NOTE

Connection 1 and Connection 2 should use internal capacitors as the load of the oscillator by configuring the CR[SCxP] bits.

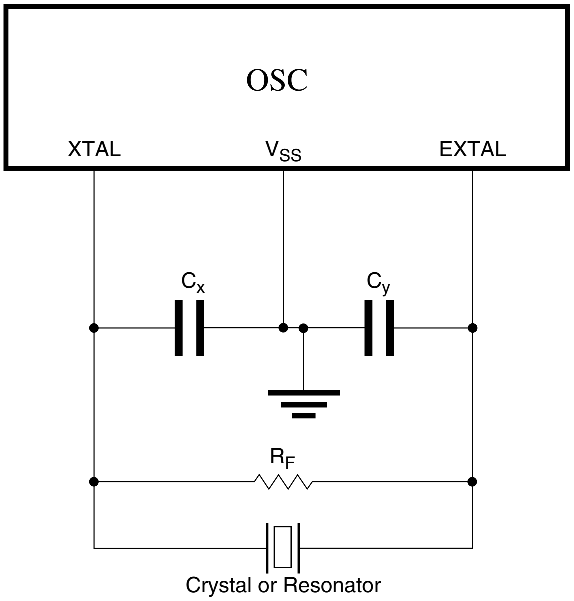

OSC
XTAL
V_{SS}
EXTAL
C_x
C_y
R_F
Crystal or Resonator

Figure 25-4. Crystal/Ceramic Resonator Connections - Connection 3

25.6 External Clock Connections

In external clock mode, the pins can be connected as shown below.

NOTE

XTAL can be used as a GPIO when the GPIO alternate function is configured for it.

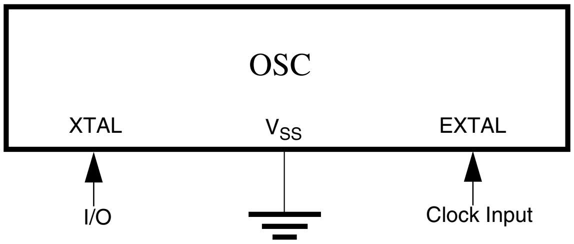


Figure 25-5. External Clock Connections

25.7 Memory Map/Register Definitions

Some oscillator module register bits are typically incorporated into other peripherals such as MCG or SIM.

25.7.1 OSC Memory Map/Register Definition

OSC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_5000	OSC Control Register (OSC0_CR)	8	R/W	00h	25.71.1/ 471

25.71.1 OSC Control Register (OSCx_CR)

NOTE

After OSC is enabled and starts generating the clocks, the configurations such as low power and frequency range, must not be changed.

Addresses: OSC0_CR is 4006_5000h base + 0h offset = 4006_5000h

Bit	7	6	5	4	3	2	1	0
Read	ERCLKEN	0	EREFSTEN	0	SC2P	SC4P	SC8P	SC16P
Write								
Reset	0	0	0	0	0	0	0	0

OSCx_CR field descriptions

Field	Description
7 ERCLKEN	External Reference Enable Enables external reference clock (OSCERCLK). 0 External reference clock is inactive. 1 External reference clock is enabled.
6 Reserved	This read-only field is reserved and always has the value zero.
5 EREFSTEN	External Reference Stop Enable Controls whether or not the external reference clock (OSCERCLK) remains enabled when MCU enters Stop mode. 0 External reference clock is disabled in Stop mode. 1 External reference clock stays enabled in Stop mode if ERCLKEN is set before entering Stop mode.
4 Reserved	This read-only field is reserved and always has the value zero.
3 SC2P	Oscillator 2 pF Capacitor Load Configure Configures the oscillator load. 0 Disable the selection. 1 Add 2 pF capacitor to the oscillator load.
2 SC4P	Oscillator 4 pF Capacitor Load Configure Configures the oscillator load. 0 Disable the selection. 1 Add 4 pF capacitor to the oscillator load.
1 SC8P	Oscillator 8 pF Capacitor Load Configure Configures the oscillator load. 0 Disable the selection. 1 Add 8 pF capacitor to the oscillator load.
0 SC16P	Oscillator 16 pF Capacitor Load Configure Configures the oscillator load. 0 Disable the selection. 1 Add 16 pF capacitor to the oscillator load.

25.8 Functional Description

This following sections provide functional details of the module.

25.8.1 OSC Module States

The states of the OSC module are shown in the following figure. The states and their transitions between each other are described in this section.

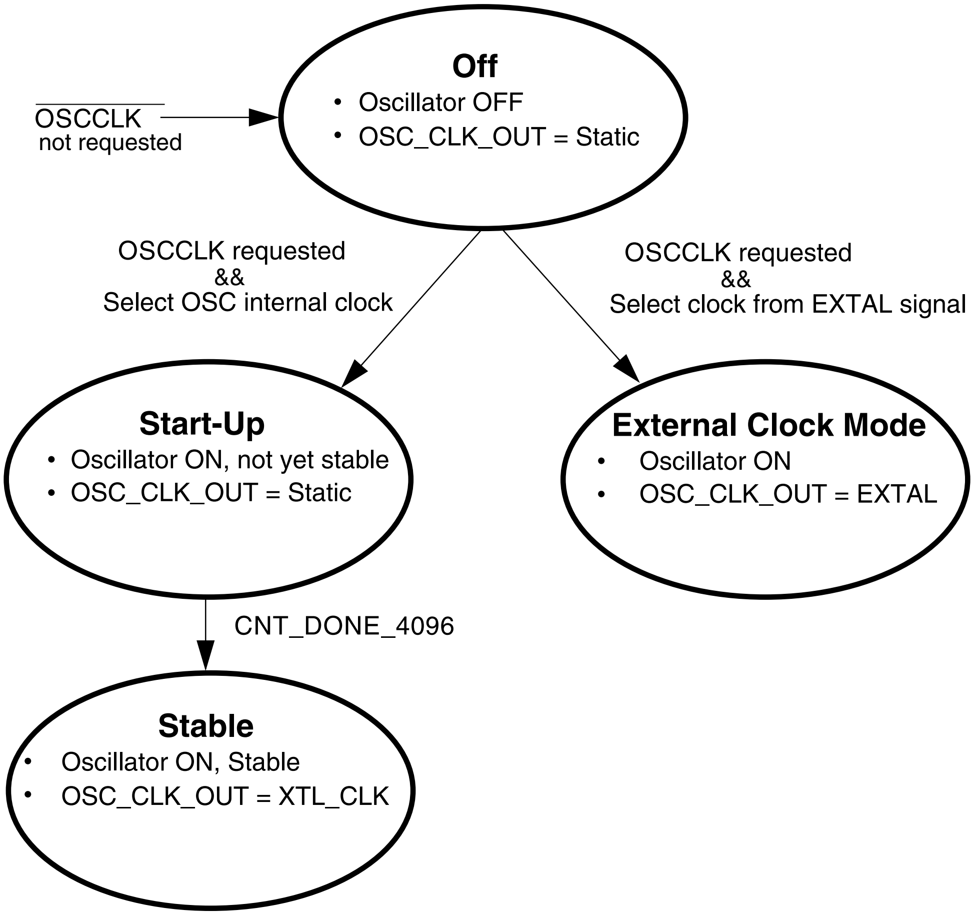

Off
• Oscillator OFF
• OSC_CLK_OUT = Static
$\overline{\text{OSCCLK}}$ not requested
OSCCLK requested && Select OSC internal clock
OSCCLK requested && Select clock from EXTAL signal
Start-Up
• Oscillator ON, not yet stable
• OSC_CLK_OUT = Static
External Clock Mode
• Oscillator ON
• OSC_CLK_OUT = EXTAL
CNT_DONE_4096
Stable
• Oscillator ON, Stable
• OSC_CLK_OUT = XTL_CLK

Figure 25-8. OSC Module State Diagram

NOTE

XTL_CLK is the clock generated internally from OSC circuits.

25.8.1.1 Off

The OSC enters the Off state when the system does not require OSC clocks. Upon entering this state, XTL_CLK is static unless OSC is configured to select the clock from the EXTAL pad by clearing the external reference clock selection bit. For details regarding the external reference clock source in this MCU, refer to the chip configuration chapter. The EXTAL and XTAL pins are also decoupled from all other oscillator circuitry in this state. The OSC module circuitry is configured to draw minimal current.

25.8.1.2 Oscillator Start-Up

The OSC enters start-up state when it is configured to generate clocks (internally the OSC_EN transitions high) using the internal oscillator circuits by setting the external reference clock selection bit. In this state, the OSC module is enabled and oscillations are starting up, but have not yet stabilized. When the oscillation amplitude becomes large enough to pass through the input buffer, XTL_CLK begins clocking the counter. When the counter reaches 4096 cycles of XTL_CLK, the oscillator is considered stable and XTL_CLK is passed to the output clock OSC_CLK_OUT.

25.8.1.3 Oscillator Stable

The OSC enters stable state when it is configured to generate clocks (internally the OSC_EN transitions high) using the internal oscillator circuits by setting the external reference clock selection bit and the counter reaches 4096 cycles of XTL_CLK (when CNT_DONE_4096 is high). In this state, the OSC module is producing a stable output clock on OSC_CLK_OUT. Its frequency is determined by the external components being used.

25.8.1.4 External Clock Mode

The OSC enters external clock state when it is enabled and external reference clock selection bit is cleared. For details regarding external reference clock source in this MCU, refer to the chip configuration chapter. In this state, the OSC module is set to buffer (with hysteresis) a clock from EXTAL onto the OSC_CLK_OUT. Its frequency is determined by the external clock being supplied.

25.8.2 OSC Module Modes

The OSC is a Pierce-type oscillator that supports external crystals or resonators operating over the frequency ranges shown in Table 25-7. These modes assume the following conditions: OSC is enabled to generate clocks (OSC_EN=1), configured to generate clocks internally (MCG_C2[EREFS] = 1), and some or one of the other peripherals (MCG, Timer, and so on) is configured to use the oscillator output clock (OSC_CLK_OUT).

Table 25-7. Oscillator Modes

Mode	Frequency Range
Low-frequency, high-gain	f_{osc_lo} (1 kHz) up to f_{osc_lo} (32.768 kHz)
Low-frequency, low-power (VLP)	
High-frequency mode1, high-gain	$f_{osc\_hi\_1}$ (3 MHz) up to $f_{osc\_hi\_1}$ (8 MHz)
High-frequency mode1, low-power	
High-frequency mode2, high-gain	$f_{osc\_hi\_2}$ (8 MHz) up to $f_{osc\_hi\_2}$ (32 MHz)
High-frequency mode2, low-power	

NOTE

For information about low power modes of operation used in this chip and their alignment with some OSC modes, refer to the chip's Power Management details.

25.8.2.1 Low-Frequency, High-Gain Mode

In Low-frequency, high-gain mode, the oscillator uses a simple inverter-style amplifier. The gain is set to achieve rail-to-rail oscillation amplitudes.

The oscillator input buffer in this mode is single-ended. It provides low pass frequency filtering as well as hysteresis for voltage filtering and converts the output to logic levels. In this mode, the internal capacitors could be used.

25.8.2.2 Low-Frequency, Low-Power Mode

In low-frequency, low-power mode, the oscillator uses a gain control loop to minimize power consumption. As the oscillation amplitude increases, the amplifier current is reduced. This continues until a desired amplitude is achieved at steady-state. This mode provides low pass frequency filtering as well as hysteresis for voltage filtering and converts the output to logic levels. In this mode, the internal capacitors could be used, the internal feedback resistor is connected, and no external resistor should be used.

In this mode, the amplifier inputs, gain-control input, and input buffer input are all capacitively coupled for leakage tolerance (not sensitive to the DC level of EXTAL).

Also in this mode, all external components except for the resonator itself are integrated, which includes the load capacitors and feeback resistor that biases EXTAL.

25.8.2.3 High-Frequency, High-Gain Mode

In high-frequency, high-gain mode, the oscillator uses a simple inverter-style amplifier. The gain is set to achieve rail-to-rail oscillation amplitudes. This mode provides low pass frequency filtering as well as hysteresis for voltage filtering and converts the output to logic levels. In this mode, the internal capacitors could be used.

25.8.2.4 High-Frequency, Low-Power Mode

In high-frequency, low-power mode, the oscillator uses a gain control loop to minimize power consumption. As the oscillation amplitude increases, the amplifier current is reduced. This continues until a desired amplitude is achieved at steady-state. In this mode, the internal capacitors could be used, the internal feedback resistor is connected, and no external resistor should be used.

The oscillator input buffer in this mode is differential. It provides low pass frequency filtering as well as hysteresis for voltage filtering and converts the output to logic levels.

25.8.3 Counter

The oscillator output clock (OSC_CLK_OUT) is gated off until the counter has detected 4096 cycles of its input clock (XTL_CLK). After 4096 cycles are completed, the counter passes XTL_CLK onto OSC_CLK_OUT. This counting time-out is used to guarantee output clock stability.

25.8.4 Reference Clock Pin Requirements

The OSC module requires use of both the EXTAL and XTAL pins to generate an output clock in Oscillator mode, but requires only the EXTAL pin in External clock mode. The EXTAL and XTAL pins are available for I/O. For the implementation of these pins on this device, refer to the Signal Multiplexing chapter.

25.9 Reset

There is no reset state associated with the OSC module. The counter logic is reset when the OSC is not configured to generate clocks.

There are no sources of reset requests for the OSC module.

25.10 Low Power Modes Operation

When the MCU enters Stop modes, the OSC is functional depending on ERCLKEN and EREFSETN bit settings. If both these bits are set, the OSC is in operation. In Low Leakage Stop (LLS) modes, the OSC holds all register settings. If ERCLKEN and EREFSTEN bits are set before entry to Low Leakage Stop modes, the OSC is still functional in these modes. After waking up from Very Low Leakage Stop (VLLSx) modes, all OSC register bits are reset and initialization is required through software.

25.11 Interrupts

The OSC module does not generate any interrupts.

Chapter 26
RTC Oscillator

26.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The RTC oscillator module provides the clock source for the RTC. The RTC oscillator module, in conjunction with an external crystal, generates a reference clock for the RTC.

26.1.1 Features and Modes

The key features of the RTC oscillator are as follows:

- Supports 32 kHz crystals with very low power
- Consists of internal feed back resistor
- Consists of internal programmable capacitors as the Cload of the oscillator
- Automatic Gain Control (AGC) to optimize power consumption

The RTC oscillator operations are described in detail in Functional Description .

26.1.2 Block Diagram

The following is the block diagram of the RTC oscillator.

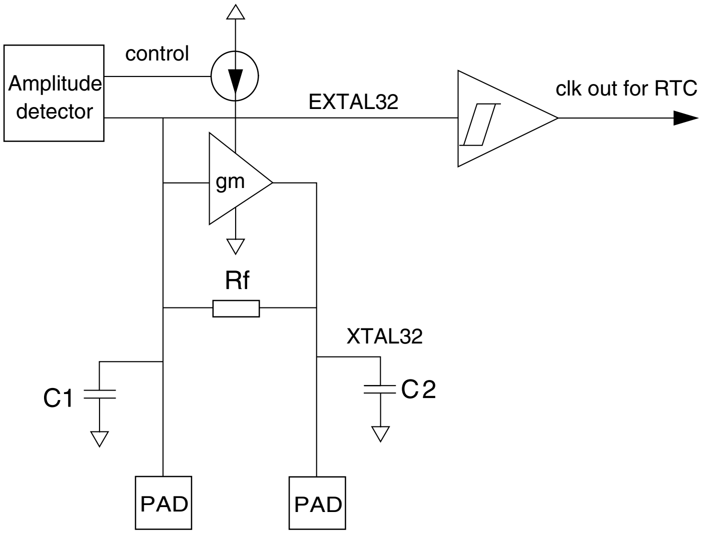

Amplitude detector
control
EXTAL32
clk out for RTC
gm
Rf
XTAL32
C1
C2
PAD
PAD

Figure 26-1. RTC Oscillator Block Diagram

26.2 RTC Signal Descriptions

The following table shows the user-accessible signals available for the RTC oscillator. See the chip-level specification to find out which signals are actually connected to the external pins.

Table 26-1. RTC Signal Descriptions

Signal	Description	I/O
EXTAL32	Oscillator Input	I
XTAL32	Oscillator Output	O

26.2.1 EXTAL32 — Oscillator Input

This signal is the analog input of the RTC oscillator.

26.2.2 XTAL32 — Oscillator Output

This signal is the analog output of the RTC oscillator module.

26.3 External Crystal Connections

The connections with a crystal is shown in the following figure. External load capacitors and feedback resistor are not required.

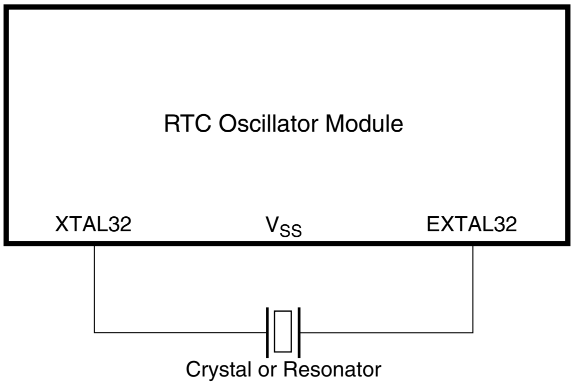


Figure 26-2. Crystal Connections

26.4 Memory Map/Register Descriptions

RTC oscillator control bits are part of the RTC registers. Refer to RTC_CR for more details.

26.5 Functional Description

As shown in Figure 26-1, the module includes an amplifier which supplies the negative resistor for the RTC oscillator. The gain of the amplifier is controlled by the amplitude detector, which optimizes the power consumption. A schmitt trigger is used to translate the sine-wave generated by this oscillator to a pulse clock out, which is a reference clock for the RTC digital core.

The oscillator includes an internal feedback resistor of approximately 100 MΩ between EXTAL32 and XTAL32.

In addition, there are two programmable capacitors with this oscillator, which can be used as the Cload of the oscillator. The programmable range is from 0pF to 30pF.

26.6 Reset Overview

There is no reset state associated with the RTC oscillator.

26.7 Interrupts

The RTC oscillator does not generate any interrupts.

Chapter 27
Flash Memory Controller (FMC)

27.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The Flash Memory Controller (FMC) is a memory acceleration unit that provides:

- an interface between the device and the program flash memory and FlexNVM.
- buffers that can accelerate flash memory and FlexNVM data transfers.

27.1.1 Overview

The Flash Memory Controller manages the interface between the device and the flash memory. The FMC receives status information detailing the configuration of the memory and uses this information to ensure a proper interface. The following table shows the supported 8-bit, 16-bit, and 32-bit read/write operations.

Flash memory type	Read	Write
Program flash memory	x	—[1]
FlexNVM used as data flash memory	x	—[1]
FlexNVM and FlexRAM used as EEPROM	x	x

1. A write operation to program flash memory or to FlexNVM used as data flash memory results in a bus error.

In addition, the FMC provides three separate mechanisms for accelerating the interface between the device and the flash memory. A 32-bit speculation buffer can prefetch the next 32-bit flash memory location, and both a 4-way, 2-set cache and a single-entry 32-bit buffer can store previously accessed flash memory or FlexNVM data for quick access times.

27.1.2 Features

The FMC's features include:

- Interface between the device and the flash memory and FlexMemory:
  - 8-bit, 16-bit, and 32-bit read operations to program flash memory and FlexNVM used as data flash memory.
  - 8-bit, 16-bit, and 32-bit read and write operations to FlexNVM and FlexRAM used as EEPROM.
  - Read accesses to consecutive 32-bit spaces in memory return the second read data with no wait states. The memory returns 32 bits via the 32-bit bus access.
  - Crossbar master access protection for setting no access, read only access, write only access, or read/write access for each crossbar master.
- Acceleration of data transfer from program flash memory and FlexMemory to the device:
  - 32-bit prefetch speculation buffer with controls for instruction/data access per master
  - 4-way, 2-set, 32-bit line size cache for a total of eight 32-bit entries with controls for replacement algorithm and lock per way
  - Single-entry buffer with enable
  - Invalidation control for the speculation buffer and the single-entry buffer

27.2 Modes of operation

The FMC only operates when the device accesses the flash memory or FlexMemory.

In terms of device power modes, the FMC only operates in run and wait modes, including VLPR and VLPW modes.

For any device power mode where the flash memory or FlexMemory cannot be accessed, the FMC is disabled.

27.3 External signal description

The FMC has no external signals.

27.4 Memory map and register descriptions

The programming model consists of the FMC control registers and the program visible cache (data and tag/valid entries).

NOTE

> Program the registers only while the flash controller is idle (for example, execute from RAM). Changing configuration settings while a flash access is in progress can lead to non-deterministic behavior.

Table 27-2. FMC register access

Registers	Read access		Write access	
	Mode	Length	Mode	Length
Control registers: PFAPR, PFB0CR	Supervisor (privileged) mode or user mode	32 bits	Supervisor (privileged) mode only	8, 16, or 32 bits
Cache registers	Supervisor (privileged) mode or user mode	32 bits	Supervisor (privileged) mode only	32 bits

NOTE

> Accesses to unimplemented registers within the FMC's address space return a bus error.

The cache entries, both data and tag/valid, can be read at any time.

NOTE

> System software is required to maintain memory coherence when any segment of the flash cache is programmed. For example, all buffer data associated with the reprogrammed flash should be invalidated. Accordingly, cache program visible writes must occur after a programming or erase event is completed and before the new memory image is accessed.

The cache is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. The following table elaborates on the tag/valid and data entries.

Table 27-3. Program visible cache registers

Cache storage	Based at offset	Contents of 32-bit read	Nomenclature	Nomenclature example
Tag	100h	13'h0, tag[18:6], 5'h0, valid	In TAGVDWxSy, x denotes the way, and y denotes the set.	TAGVDW1S1 is the 13-bit tag and 1-bit valid for cache entry way 1, set 1.
Data	200h	Data word	In DATAWxSy, x denotes the way, and y denotes the set.	DATAW1S1 represents bits [31:0] of data entry way 1, set 1.

FMC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4001_F000	Flash Access Protection Register (FMC_PFAPR)	32	R/W	00F8_003Fh	27.4.1/487
4001_F004	Flash Control Register (FMC_PFB0CR)	32	R/W	3000_001Fh	27.4.2/489
4001_F100	Cache Tag Storage (FMC_TAGVDW0S0)	32	R/W	0000_0000h	27.4.3/491
4001_F104	Cache Tag Storage (FMC_TAGVDW0S1)	32	R/W	0000_0000h	27.4.3/491
4001_F108	Cache Tag Storage (FMC_TAGVDW1S0)	32	R/W	0000_0000h	27.4.4/492
4001_F10C	Cache Tag Storage (FMC_TAGVDW1S1)	32	R/W	0000_0000h	27.4.4/492
4001_F110	Cache Tag Storage (FMC_TAGVDW2S0)	32	R/W	0000_0000h	27.4.5/492
4001_F114	Cache Tag Storage (FMC_TAGVDW2S1)	32	R/W	0000_0000h	27.4.5/492
4001_F118	Cache Tag Storage (FMC_TAGVDW3S0)	32	R/W	0000_0000h	27.4.6/493
4001_F11C	Cache Tag Storage (FMC_TAGVDW3S1)	32	R/W	0000_0000h	27.4.6/493
4001_F200	Cache Data Storage (FMC_DATAW0S0)	32	R/W	0000_0000h	27.4.7/494
4001_F204	Cache Data Storage (FMC_DATAW0S1)	32	R/W	0000_0000h	27.4.7/494
4001_F208	Cache Data Storage (FMC_DATAW1S0)	32	R/W	0000_0000h	27.4.8/494
4001_F20C	Cache Data Storage (FMC_DATAW1S1)	32	R/W	0000_0000h	27.4.8/494
4001_F210	Cache Data Storage (FMC_DATAW2S0)	32	R/W	0000_0000h	27.4.9/495
4001_F214	Cache Data Storage (FMC_DATAW2S1)	32	R/W	0000_0000h	27.4.9/495
4001_F218	Cache Data Storage (FMC_DATAW3S0)	32	R/W	0000_0000h	27.4.10/ 495
4001_F21C	Cache Data Storage (FMC_DATAW3S1)	32	R/W	0000_0000h	27.4.10/ 495

27.4.1 Flash Access Protection Register (FMC_PFAPR)

Address: FMC_PFAPR is 4001_F000h base + 0h offset = 4001_F000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0								Reserved				M3PFD	M2PFD	M1PFD	M0PFD
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								M3AP[1:0]		M2AP[1:0]		M1AP[1:0]		M0AP[1:0]	
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

FMC_PFAPR field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero.
23–20 Reserved	This field is reserved.
19 M3PFD	Master 3 Prefetch Disable These bits control whether prefetching is enabled based on the logical number of the requesting crossbar switch master. This field is further qualified by the PFBnCR[BxDPE,BxIPE] bits. 0 Prefetching for this master is enabled. 1 Prefetching for this master is disabled.
18 M2PFD	Master 2 Prefetch Disable These bits control whether prefetching is enabled based on the logical number of the requesting crossbar switch master. This field is further qualified by the PFBnCR[BxDPE,BxIPE] bits. 0 Prefetching for this master is enabled. 1 Prefetching for this master is disabled.
17 M1PFD	Master 1 Prefetch Disable These bits control whether prefetching is enabled based on the logical number of the requesting crossbar switch master. This field is further qualified by the PFBnCR[BxDPE,BxIPE] bits. 0 Prefetching for this master is enabled. 1 Prefetching for this master is disabled.
16 M0PFD	Master 0 Prefetch Disable These bits control whether prefetching is enabled based on the logical number of the requesting crossbar switch master. This field is further qualified by the PFBnCR[BxDPE,BxIPE] bits. 0 Prefetching for this master is enabled. 1 Prefetching for this master is disabled.

Table continues on the next page...

FMC_PFAPR field descriptions (continued)

Field	Description
15–8 Reserved	This field is reserved.
7–6 M3AP[1:0]	Master 3 Access Protection This field controls whether read and write access to the flash are allowed based on the logical master number of the requesting crossbar switch master. 00 No access may be performed by this master 01 Only read accesses may be performed by this master 10 Only write accesses may be performed by this master 11 Both read and write accesses may be performed by this master
5–4 M2AP[1:0]	Master 2 Access Protection This field controls whether read and write access to the flash are allowed based on the logical master number of the requesting crossbar switch master. 00 No access may be performed by this master 01 Only read accesses may be performed by this master 10 Only write accesses may be performed by this master 11 Both read and write accesses may be performed by this master
3–2 M1AP[1:0]	Master 1 Access Protection This field controls whether read and write access to the flash are allowed based on the logical master number of the requesting crossbar switch master. 00 No access may be performed by this master 01 Only read accesses may be performed by this master 10 Only write accesses may be performed by this master 11 Both read and write accesses may be performed by this master
1–0 M0AP[1:0]	Master 0 Access Protection This field controls whether read and write access to the flash are allowed based on the logical master number of the requesting crossbar switch master. 00 No access may be performed by this master 01 Only read accesses may be performed by this master 10 Only write accesses may be performed by this master 11 Both read and write accesses may be performed by this master

27.4.2 Flash Control Register (FMC_PFB0CR)

Address: FMC_PFB0CR is 4001_F000h base + 4h offset = 4001_F004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	B0RWSC[3:0]				CLCK_WAY[3:0]				0				0	B0MW[1:0]		0
W									CINV_WAY[3:0]				S_B_INV			
Reset	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CRC[2:0]			B0DCE	B0ICE	B0DPE	B0IPE	B0SEBE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

FMC_PFB0CR field descriptions

Field	Description
31–28 B0RWSC[3:0]	Read Wait State Control This read-only field defines the number of wait states required to access the flash memory. The relationship between the read access time of the flash array (expressed in system clock cycles) and RWSC is defined as: Access time of flash array [system clocks] = RWSC + 1 The FMC automatically calculates this value based on the ratio of the system clock speed to the flash clock speed. For example, when this ratio is 4:1, the field's value is 3h.
27–24 CLCK_WAY[3:0]	Cache Lock Way x These bits determine if the given cache way is locked such that its contents will not be displaced by future misses. The bit setting definitions are for each bit in the field. 0 Cache way is unlocked and may be displaced 1 Cache way is locked and its contents are not displaced
23–20 CINV_WAY[3:0]	Cache Invalidate Way x These bits determine if the given cache way is to be invalidated (cleared). When a bit within this field is written, the corresponding cache way is immediately invalidated: the way's tag, data, and valid contents are cleared. This field always reads as zero. Cache invalidation takes precedence over locking. The cache is invalidated by system reset. System software is required to maintain memory coherency when any segment of the flash memory is programmed or erased. Accordingly, cache invalidations must occur after a programming or erase event is completed and before the new memory image is accessed. The bit setting definitions are for each bit in the field.

Table continues on the next page...

FMC_PFB0CR field descriptions (continued)

Field	Description
	0 No cache way invalidation for the corresponding cache 1 Invalidate cache way for the corresponding cache: clear the tag, data, and vld bits of ways selected
19 S_B_INV	Invalidate Prefetch Speculation Buffer This bit determines if the FMC's prefetch speculation buffer and the single entry page buffer are to be invalidated (cleared). When this bit is written, the speculation buffer and single entry buffer are immediately cleared. This bit always reads as zero. 0 Speculation buffer and single entry buffer are not affected. 1 Invalidate (clear) speculation buffer and single entry buffer.
18–17 B0MW[1:0]	Memory Width This read-only field defines the width of the memory. 00 32 bits 01 64 bits 1x Reserved
16 Reserved	This read-only field is reserved and always has the value zero.
15–8 Reserved	This read-only field is reserved and always has the value zero.
7–5 CRC[2:0]	Cache Replacement Control This 3-bit field defines the replacement algorithm for accesses that are cached. 000 LRU replacement algorithm per set across all four ways 001 Reserved 010 Independent LRU with ways [0-1] for ifetches, [2-3] for data 011 Independent LRU with ways [0-2] for ifetches, [3] for data 1xx Reserved
4 B0DCE	Data Cache Enable This bit controls whether data references are loaded into the cache. 0 Do not cache data references. 1 Cache data references.
3 B0ICE	Instruction Cache Enable This bit controls whether instruction fetches are loaded into the cache. 0 Do not cache instruction fetches. 1 Cache instruction fetches.
2 B0DPE	Data Prefetch Enable This bit controls whether prefetches (or speculative accesses) are initiated in response to data references. 0 Do not prefetch in response to data references. 1 Enable prefetches in response to data references.

Table continues on the next page...

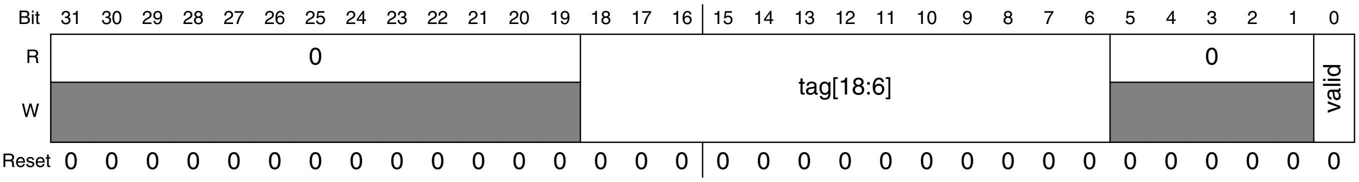
FMC_PFB0CR field descriptions (continued)

Field	Description
1 B0IPE	Instruction Prefetch Enable This bit controls whether prefetches (or speculative accesses) are initiated in response to instruction fetches. 0 Do not prefetch in response to instruction fetches. 1 Enable prefetches in response to instruction fetches.
0 B0SEBE	Single Entry Buffer Enable This bit controls whether the single entry page buffer is enabled in response to flash read accesses. A high-to-low transition of this enable forces the page buffer to be invalidated. 0 Single entry buffer is disabled. 1 Single entry buffer is enabled.

27.4.3 Cache Tag Storage (FMC_TAGVDW0S*n*)

The 32-entry cache is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In TAGVDWxSy, x denotes the way, and y denotes the set. This section represents tag/vld information for the 2 sets (n=0-1) in way 0.

Addresses: TAGVDW0S0 is 4001_F000h base + 100h offset = 4001_F100h
TAGVDW0S1 is 4001_F000h base + 104h offset = 4001_F104h



FMC_TAGVDW0S*n* field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero.
18–6 tag[18:6]	13-bit tag for cache entry
5–1 Reserved	This read-only field is reserved and always has the value zero.
0 valid	1-bit valid for cache entry

27.4.4 Cache Tag Storage (FMC_TAGVDW1S*n*)

The 32-entry cache is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In TAGVDWxSy, x denotes the way, and y denotes the set. This section represents tag/vld information for the 2 sets (n=0-1) in way 1.

Addresses: TAGVDW1S0 is 4001_F000h base + 108h offset = 4001_F108h
TAGVDW1S1 is 4001_F000h base + 10Ch offset = 4001_F10Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													tag[18:6]													0					valid
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_TAGVDW1S*n* field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero.
18–6 tag[18:6]	13-bit tag for cache entry
5–1 Reserved	This read-only field is reserved and always has the value zero.
0 valid	1-bit valid for cache entry

27.4.5 Cache Tag Storage (FMC_TAGVDW2S*n*)

The 32-entry cache is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In TAGVDWxSy, x denotes the way, and y denotes the set. This section represents tag/vld information for the 2 sets (n=0-1) in way 2.

Addresses: TAGVDW2S0 is 4001_F000h base + 110h offset = 4001_F110h
TAGVDW2S1 is 4001_F000h base + 114h offset = 4001_F114h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													tag[18:6]													0					valid
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

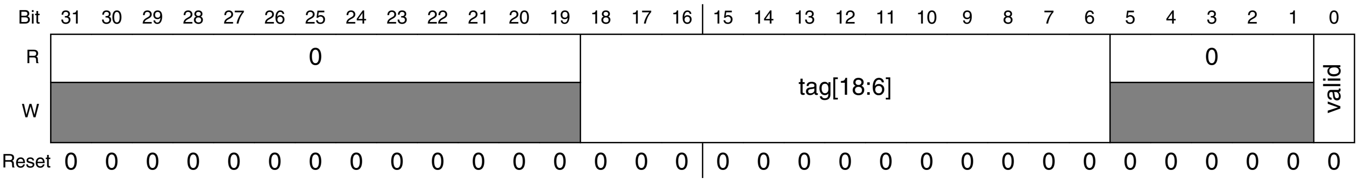
FMC_TAGVDW2S*n* field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero.
18–6 tag[18:6]	13-bit tag for cache entry
5–1 Reserved	This read-only field is reserved and always has the value zero.
0 valid	1-bit valid for cache entry

27.4.6 Cache Tag Storage (FMC_TAGVDW3S*n*)

The 32-entry cache is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In TAGVDWxSy, x denotes the way, and y denotes the set. This section represents tag/vld information for the 2 sets (n=0-1) in way 3.

Addresses: TAGVDW3S0 is 4001_F000h base + 118h offset = 4001_F118h
TAGVDW3S1 is 4001_F000h base + 11Ch offset = 4001_F11Ch



Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													tag[18:6]													0					valid
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_TAGVDW3S*n* field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero.
18–6 tag[18:6]	13-bit tag for cache entry
5–1 Reserved	This read-only field is reserved and always has the value zero.
0 valid	1-bit valid for cache entry

27.4.7 Cache Data Storage (FMC_DATAW0S*n*)

The cache of eight 32-bit entries is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In DATAWxSy, x denotes the way, and y denotes the set. This section represents data for bits [31:0] of sets 0-1 in way 0.

Addresses: DATAW0S0 is 4001_F000h base + 200h offset = 4001_F200h
DATAW0S1 is 4001_F000h base + 204h offset = 4001_F204h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	data[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_DATAW0S*n* field descriptions

Field	Description
31–0 data[31:0]	Bits [31:0] of data entry

27.4.8 Cache Data Storage (FMC_DATAW1S*n*)

The cache of eight 32-bit entries is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In DATAWxSy, x denotes the way, and y denotes the set. This section represents data for bits [31:0] of sets 0-1 in way 1.

Addresses: DATAW1S0 is 4001_F000h base + 208h offset = 4001_F208h
DATAW1S1 is 4001_F000h base + 20Ch offset = 4001_F20Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	data[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_DATAW1S*n* field descriptions

Field	Description
31–0 data[31:0]	Bits [31:0] of data entry

27.4.9 Cache Data Storage (FMC_DATAW2S*n*)

The cache of eight 32-bit entries is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In DATAWxSy, x denotes the way, and y denotes the set. This section represents data for bits [31:0] of sets 0-1 in way 2.

Addresses: DATAW2S0 is 4001_F000h base + 210h offset = 4001_F210h
DATAW2S1 is 4001_F000h base + 214h offset = 4001_F214h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	data[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_DATAW2S*n* field descriptions

Field	Description
31–0 data[31:0]	Bits [31:0] of data entry

27.4.10 Cache Data Storage (FMC_DATAW3S*n*)

The cache of eight 32-bit entries is a 4-way, set-associative cache with 2 sets. The ways are numbered 0-3 and the sets are numbered 0-1. In DATAWxSy, x denotes the way, and y denotes the set. This section represents data for bits [31:0] of sets 0-1 in way 3.

Addresses: DATAW3S0 is 4001_F000h base + 218h offset = 4001_F218h
DATAW3S1 is 4001_F000h base + 21Ch offset = 4001_F21Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	data[31:0]																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FMC_DATAW3S*n* field descriptions

Field	Description
31–0 data[31:0]	Bits [31:0] of data entry

27.5 Functional description

The FMC is a flash acceleration unit with flexible buffers for user configuration. Besides managing the interface between the device and the flash memory and FlexMemory, the FMC can be used to restrict access from crossbar switch masters and customize the cache and buffers to provide single-cycle system-clock data-access times. Whenever a hit occurs for the prefetch speculation buffer, the cache, or the single-entry buffer, the requested data is transferred within a single system clock.

Upon system reset, the FMC is configured to provide a significant level of buffering for transfers from the flash memory or FlexMemory:

- Crossbar masters 0, 1, 2 have read access to the memory.
- When FlexNVM is used with FlexRAM as EEPROM, these crossbar masters have write access to the EEPROM.
- Prefetch support for data and instructions is enabled for crossbar masters 0, 1, 2.
- The cache is configured for least recently used (LRU) replacement for all four ways.
- The cache is configured for data or instruction replacement.
- The single-entry buffer is enabled.

Though the default configuration provides a high degree of flash acceleration, advanced users may desire to customize the FMC buffer configurations to maximize throughput for their use cases. When reconfiguring the FMC for custom use cases, do not program the FMC's control registers while the flash memory or FlexMemory is being accessed. Instead, change the control registers with a routine executing from RAM in supervisor mode.

The FMC's cache and buffering controls within PFB0CR allow the tuning of resources to suit particular applications' needs. The cache and two buffers are each controlled individually. The register controls enable buffering and prefetching per access type (instruction fetch or data reference). The cache also supports three types of LRU replacement algorithms:

- LRU per set across all four ways,
- LRU with ways [0-1] for instruction fetches and ways [2-3] for data fetches, and
- LRU with ways [0-2] for instruction fetches and way [3] for data fetches.

As an application example: if both instruction fetches and data references are accessing the flash memory, control is available to send instruction fetches, data references, or both to the cache or the single-entry buffer. Likewise, speculation can be enabled or disabled for either type of access. If both instruction fetches and data references are cached, the cache's way resources may be divided in several ways between the instruction fetches and data references.

Chapter 28
Flash Memory Module (FTFL)

28.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The flash memory module includes the following accessible memory regions:

- Program flash memory for vector space and code store
- FlexNVM for data store and additional code store
- FlexRAM for high-endurance data store or traditional RAM

Flash memory is ideal for single-supply applications, permitting in-the-field erase and reprogramming operations without the need for any external high voltage power sources.

The flash memory module includes a memory controller that executes commands to modify flash memory contents. An erased bit reads '1' and a programmed bit reads '0'. The programming operation is unidirectional; it can only move bits from the '1' state (erased) to the '0' state (programmed). Only the erase operation restores bits from '0' to '1'; bits cannot be programmed from a '0' to a '1'.

CAUTION

> A flash memory location must be in the erased state before being programmed. Cumulative programming of bits (back-to-back program operations without an intervening erase) within a flash memory location is not allowed. Re-programming of existing 0s to 0 is not allowed as this overstresses the device.
>
> The standard shipping condition for flash memory is erased with security disabled. Data loss over time may occur due to degradation of the erased ('1') states and/or programmed ('0')

states. Therefore, it is recommended that each flash block or sector be re-erased immediately prior to factory programming to ensure that the full data retention capability is achieved.

28.1.1 Features

The flash memory module includes the following features.

NOTE

See the device's Chip Configuration details for the exact amount of flash memory available on your device.

28.1.1.1 Program Flash Memory Features

- Sector size of 1 Kbyte
- Program flash protection scheme prevents accidental program or erase of stored data
- Automated, built-in, program and erase algorithms with verify
- Section programming for faster bulk programming times
- Read access to program flash memory possible while programming or erasing data in the data flash memory or FlexRAM

28.1.1.2 FlexNVM Memory Features

When FlexNVM is partitioned for data flash memory:

- Sector size of 1 Kbyte
- Protection scheme prevents accidental program or erase of stored data
- Automated, built-in program and erase algorithms with verify
- Section programming for faster bulk programming times
- Read access to data flash memory possible while programming or erasing data in the program flash memory

28.1.1.3 FlexRAM Features

- Memory that can be used as traditional RAM or as high-endurance EEPROM storage
- Up to 2 Kbytes of FlexRAM configured for EEPROM or traditional RAM operations
- When configured for EEPROM:
    - Protection scheme prevents accidental program or erase of data written for EEPROM
    - Built-in hardware emulation scheme to automate EEPROM record maintenance functions
    - Programmable EEPROM data set size and FlexNVM partition code facilitating EEPROM memory endurance trade-offs
    - Supports FlexRAM aligned writes of 1, 2, or 4 bytes at a time
    - Read access to FlexRAM possible while programming or erasing data in the program or data flash memory
- When configured for traditional RAM:
    - Read and write access possible to the FlexRAM while programming or erasing data in the program or data flash memory

28.1.1.4 Other Flash Memory Module Features

- Internal high-voltage supply generator for flash memory program and erase operations
- Optional interrupt generation upon flash command completion
- Supports MCU security mechanisms which prevent unauthorized access to the flash memory contents

28.1.2 Block Diagram

The block diagram of the flash memory module is shown in the following figure.

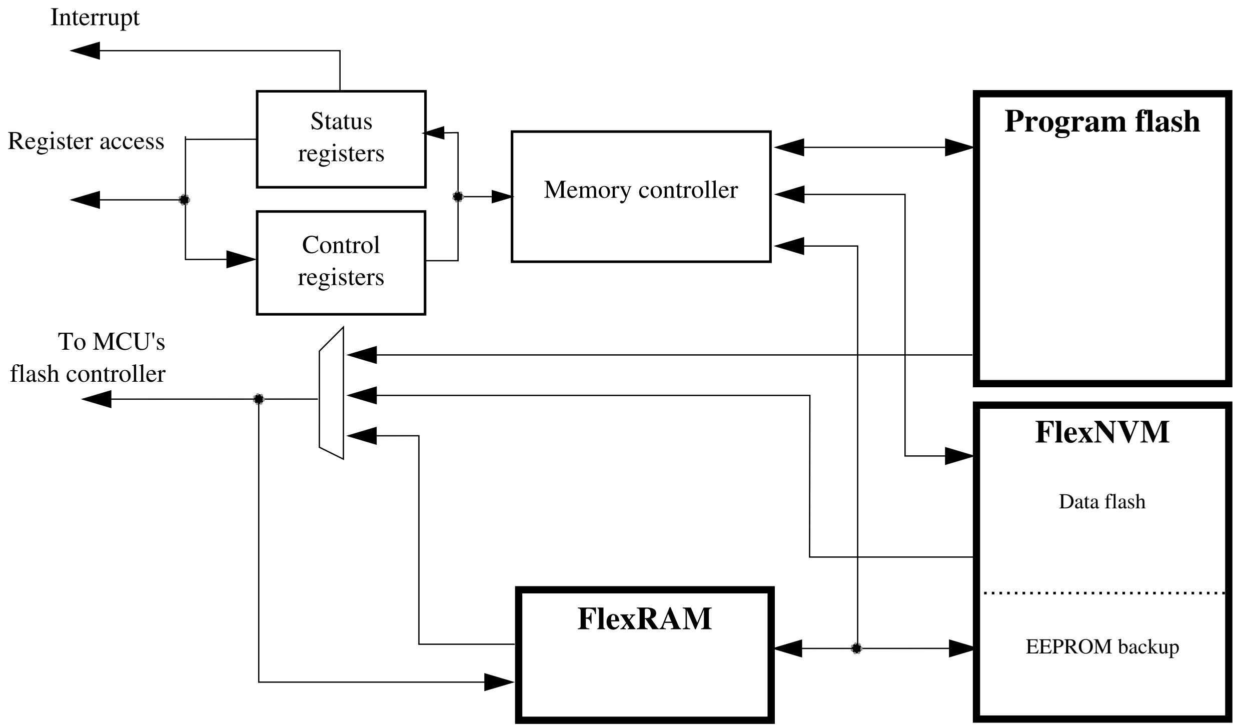

Interrupt
Register access
Status registers
Control registers
Memory controller
Program flash
To MCU's flash controller
FlexNVM
Data flash
FlexRAM
EEPROM backup

Figure 28-1. Flash Block Diagram

28.1.3 Glossary

Command write sequence — A series of MCU writes to the flash FCCOB register group that initiates and controls the execution of flash algorithms that are built into the flash memory module.

Data flash memory — Partitioned from the FlexNVM block, the data flash memory provides nonvolatile storage for user data, boot code, and additional code store.

Data flash sector — The data flash sector is the smallest portion of the data flash memory that can be erased.

EEPROM — Using a built-in filing system, the flash memory module emulates the characteristics of an EEPROM by effectively providing a high-endurance, byte-writeable (program and erase) NVM.

EEPROM backup data header — The EEPROM backup data header is comprised of a 32-bit field found in EEPROM backup data memory which contains information used by the EEPROM filing system to determine the status of a specific EEPROM backup flash sector.

EEPROM backup data record — The EEPROM backup data record is comprised of a 2-bit status field, a 14-bit address field, and a 16-bit data field found in EEPROM backup data memory which is used by the EEPROM filing system. If the status field indicates a record is valid, the data field is mirrored in the FlexRAM at a location determined by the address field.

EEPROM backup data memory — Partitioned from the FlexNVM block, EEPROM backup data memory provides nonvolatile storage for the EEPROM filing system representing data written to the FlexRAM requiring highest endurance.

EEPROM backup data sector — The EEPROM backup data sector contains one EEPROM backup data header and up to 255 EEPROM backup data records, which are used by the EEPROM filing system.

Endurance — The number of times that a flash memory location can be erased and reprogrammed.

FCCOB (Flash Common Command Object) — A group of flash registers that are used to pass command, address, data, and any associated parameters to the memory controller in the flash memory module.

Flash block — A macro within the flash memory module which provides the nonvolatile memory storage.

FlexMemory — Flash configuration that supports data flash, EEPROM, and FlexRAM.

FlexNVM Block — The FlexNVM block can be configured to be used as data flash memory, EEPROM backup flash memory, or a combination of both.

FlexRAM — The FlexRAM refers to a RAM, dedicated to the flash memory module, that can be configured to store EEPROM data or as traditional RAM. When configured for EEPROM, valid writes to the FlexRAM generate new EEPROM backup data records stored in the EEPROM backup flash memory.

Flash Memory Module — All flash blocks plus a flash management unit providing high-level control and an interface to MCU buses.

IFR — Nonvolatile information register found in each flash block, separate from the main memory array.

NVM — Nonvolatile memory. A memory technology that maintains stored data during power-off. The flash array is an NVM using NOR-type flash memory technology.

NVM Normal Mode — An NVM mode that provides basic user access to flash memory module resources. The CPU or other bus masters initiate flash program and erase operations (or other flash commands) using writes to the FCCOB register group in the flash memory module.

NVM Special Mode — An NVM mode enabling external, off-chip access to the memory resources in the flash memory module. A reduced flash command set is available when the MCU is secured. See the Chip Configuration details for information on when this mode is used.

Phrase — 64 bits of data with an aligned phrase having byte-address[2:0] = 000.

Longword — 32 bits of data with an aligned longword having byte-address[1:0] = 00.

Word — 16 bits of data with an aligned word having byte-address[0] = 0.

Program flash — The program flash memory provides nonvolatile storage for vectors and code store.

Program flash Sector — The smallest portion of the program flash memory (consecutive addresses) that can be erased.

Retention — The length of time that data can be kept in the NVM without experiencing errors upon readout. Since erased (1) states are subject to degradation just like programmed (0) states, the data retention limit may be reached from the last erase operation (not from the programming time).

RWW— Read-While-Write. The ability to simultaneously read from one memory resource while commanded operations are active in another memory resource.

Section Program Buffer — Lower half of the FlexRAM allocated for storing large amounts of data for programming via the Program Section command.

Secure — An MCU state conveyed to the flash memory module as described in the Chip Configuration details for this device. In the secure state, reading and changing NVM contents is restricted.

28.2 External Signal Description

The flash memory module contains no signals that connect off-chip.

28.3 Memory Map and Registers

This section describes the memory map and registers for the flash memory module. Data read from unimplemented memory space in the flash memory module is undefined. Writes to unimplemented or reserved memory space (registers) in the flash memory module are ignored.

28.3.1 Flash Configuration Field Description

The program flash memory contains a 16-byte flash configuration field that stores default protection settings (loaded on reset) and security information that allows the MCU to restrict access to the flash memory module.

Flash Configuration Field Byte Address	Size (Bytes)	Field Description
0x0_0400 - 0x0_0407	8	Backdoor Comparison Key. Refer to Verify Backdoor Access Key Command and Unsecuring the Chip Using Backdoor Key Access.
0x0_0408 - 0x0_040B	4	Program flash protection bytes. Refer to the description of the Program Flash Protection Registers (FPROT0-3).
0x0_040F	1	Data flash protection byte. Refer to the description of the Data Flash Protection Register (FDPROT).
0x0_040E	1	EEPROM protection byte. Refer to the description of the EEPROM Protection Register (FEPROT).
0x0_040D	1	Flash nonvolatile option byte. Refer to the description of the Flash Option Register (FOPT).
0x0_040C	1	Flash security byte. Refer to the description of the Flash Security Register (FSEC).

28.3.2 Program Flash IFR Map

The program flash IFR is nonvolatile information memory that can be read freely, but the user has no erase and limited program capabilities (see the Read Once, Program Once, and Read Resource commands in Read Once Command, Program Once Command and Read Resource Command). The contents of the program flash IFR are summarized in the following table and further described in the subsequent paragraphs.

Address Range	Size (Bytes)	Field Description
0x00 – 0xBF	192	Reserved
0xC0 – 0xFF	64	Program Once Field

28.3.2.1 Program Once Field

The Program Once Field in the program flash IFR provides 64 bytes of user data storage separate from the program flash main array. The user can program the Program Once Field one time only as there is no program flash IFR erase mechanism available to the user. The Program Once Field can be read any number of times. This section of the program flash IFR is accessed in 4-Byte records using the Read Once and Program Once commands (see Read Once Command and Program Once Command).

28.3.3 Data Flash IFR Map

The data flash IFR is a 256 byte nonvolatile information memory that can be read and erased, but the user has limited program capabilities in the data flash IFR (see the Program Partition command in Program Partition Command, the Erase All Blocks command in Erase All Blocks Command, and the Read Resource command in Read Resource Command). The contents of the data flash IFR are summarized in the following table and further described in the subsequent paragraphs.

Address Range	Size (Bytes)	Field Description
0x00 – 0xFB, 0xFE – 0xFF	254	Reserved
0xFD	1	EEPROM data set size
0xFC	1	FlexNVM partition code

28.3.3.1 EEPROM Data Set Size

The EEPROM data set size byte in the data flash IFR supplies information which determines the amount of FlexRAM used in each of the available EEPROM subsystems. To program the EEESIZE value, see the Program Partition command described in Program Partition Command.

Table 28-1. EEPROM Data Set Size

Data flash IFR: 0x00FD							
7	6	5	4	3	2	1	0
1	1	1	1	EEESIZE			
▨	= Unimplemented or Reserved						

Table 28-2. EEPROM Data Set Size Field Description

Field	Description
7-4 Reserved	This read-only bitfield is reserved and must always be written as one.
3-0 EEESIZE	**EEPROM Size** — Encoding of the total available FlexRAM for EEPROM use. **NOTE:** EEESIZE must be 0 bytes (1111b) when the FlexNVM partition code (FlexNVM Partition Code) is set to 'No EEPROM'. '0000' = Reserved '0001' = Reserved '0010' = Reserved '0011' = 2,048 Bytes '0100' = 1,024 Bytes '0101' = 512 Bytes '0110' = 256 Bytes '0111' = 128 Bytes '1000' = 64 Bytes '1001' = 32 Bytes '1010' = Reserved '1011' = Reserved '1100' = Reserved '1101' = Reserved '1110' = Reserved '1111' = 0 Bytes

28.3.3.2 FlexNVM Partition Code

The FlexNVM Partition Code byte in the data flash IFR supplies a code which specifies how to split the FlexNVM block between data flash memory and EEPROM backup memory supporting EEPROM functions. To program the DEPART value, see the Program Partition command described in Program Partition Command.

Table 28-3. FlexNVM Partition Code

Data Flash IFR: 0x00FC							
7	6	5	4	3	2	1	0
1	1	1	1	DEPART			
▨	= Unimplemented or Reserved						

Table 28-4. FlexNVM Partition Code Field Description

Field	Description
7-4 Reserved	This read-only bitfield is reserved and must always be written as one.
3-0 DEPART	**FlexNVM Partition Code** — Encoding of the data flash / EEPROM backup split within the FlexNVM memory block. FlexNVM memory not partitioned for data flash will be used to store EEPROM records. 0000 = 32 Kbytes of data flash, No EEPROM backup (No EEPROM) 0001 = 24 Kbytes of data flash, 8 Kbytes of EEPROM backup 0010 = 16 Kbytes of data flash, 16 Kbytes of EEPROM backup 0011 = No data flash, 32 Kbytes of EEPROM backup 0100 = Reserved 0101 = Reserved 0110 = Reserved 0111 = Reserved 1000 = No data flash, 32 Kbytes of EEPROM backup 1001 = 8 Kbytes of data flash, 24 Kbytes of EEPROM backup 1010 = 16 Kbytes of data flash, 16 Kbytes of EEPROM backup 1011 = 32 Kbytes of data flash, No EEPROM backup (No EEPROM) 1100 = Reserved 1101 = Reserved 1110 = Reserved 1111 = Reserved (defaults to 32 Kbytes of data flash, No EEPROM)

28.3.4 Register Descriptions

The flash memory module contains a set of memory-mapped control and status registers.

NOTE

While a command is running (FSTAT[CCIF]=0), register writes are not accepted to any register except FCNFG and FSTAT. The no-write rule is relaxed during the start-up reset sequence, prior to the initial rise of CCIF. During this initialization period the user may write any register. All register writes are also disabled (except for registers FCNFG and FSTAT) whenever an erase suspend request is active (FCNFG[ERSSUSP]=1).

FTFL memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_0000	Flash Status Register (FTFL_FSTAT)	8	R/W	00h	28.34.1/ 508
4002_0001	Flash Configuration Register (FTFL_FCNFG)	8	R/W	00h	28.34.2/ 509
4002_0002	Flash Security Register (FTFL_FSEC)	8	R	Undefined	28.34.3/ 511
4002_0003	Flash Option Register (FTFL_FOPT)	8	R	Undefined	28.34.4/ 512
4002_0004	Flash Common Command Object Registers (FTFL_FCCOB3)	8	R/W	00h	28.34.5/ 513
4002_0005	Flash Common Command Object Registers (FTFL_FCCOB2)	8	R/W	00h	28.34.5/ 513
4002_0006	Flash Common Command Object Registers (FTFL_FCCOB1)	8	R/W	00h	28.34.5/ 513
4002_0007	Flash Common Command Object Registers (FTFL_FCCOB0)	8	R/W	00h	28.34.5/ 513
4002_0008	Flash Common Command Object Registers (FTFL_FCCOB7)	8	R/W	00h	28.34.5/ 513
4002_0009	Flash Common Command Object Registers (FTFL_FCCOB6)	8	R/W	00h	28.34.5/ 513
4002_000A	Flash Common Command Object Registers (FTFL_FCCOB5)	8	R/W	00h	28.34.5/ 513
4002_000B	Flash Common Command Object Registers (FTFL_FCCOB4)	8	R/W	00h	28.34.5/ 513
4002_000C	Flash Common Command Object Registers (FTFL_FCCOBB)	8	R/W	00h	28.34.5/ 513
4002_000D	Flash Common Command Object Registers (FTFL_FCCOBA)	8	R/W	00h	28.34.5/ 513
4002_000E	Flash Common Command Object Registers (FTFL_FCCOB9)	8	R/W	00h	28.34.5/ 513
4002_000F	Flash Common Command Object Registers (FTFL_FCCOB8)	8	R/W	00h	28.34.5/ 513
4002_0010	Program Flash Protection Registers (FTFL_FPROT3)	8	R/W	Undefined	28.34.6/ 514
4002_0011	Program Flash Protection Registers (FTFL_FPROT2)	8	R/W	Undefined	28.34.6/ 514
4002_0012	Program Flash Protection Registers (FTFL_FPROT1)	8	R/W	Undefined	28.34.6/ 514
4002_0013	Program Flash Protection Registers (FTFL_FPROT0)	8	R/W	Undefined	28.34.6/ 514

Table continues on the next page...

FTFL memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_0016	EEPROM Protection Register (FTFL_FEPROT)	8	R/W	Undefined	28.34.7/ 516
4002_0017	Data Flash Protection Register (FTFL_FDPROT)	8	R/W	Undefined	28.34.8/ 517

28.34.1 Flash Status Register (FTFL_FSTAT)

The FSTAT register reports the operational status of the flash memory module.

The CCIF, RDCOLERR, ACCERR, and FPVIOL bits are readable and writable. The MGSTAT0 bit is read only. The unassigned bits read 0 and are not writable.

NOTE

> When set, the Access Error (ACCERR) and Flash Protection Violation (FPVIOL) bits in this register prevent the launch of any more commands or writes to the FlexRAM (when EEERDY is set) until the flag is cleared (by writing a one to it).

Address: FTFL_FSTAT is 4002_0000h base + 0h offset = 4002_0000h

Bit	7	6	5	4	3	2	1	0
Read	CCIF	RDCOLERR	ACCERR	FPVIOL	0			MGSTAT0
Write	w1c	w1c	w1c	w1c				
Reset	0	0	0	0	0	0	0	0

FTFL_FSTAT field descriptions

Field	Description
7 CCIF	Command Complete Interrupt Flag The CCIF flag indicates that a flash command or EEPROM file system operation has completed. The CCIF flag is cleared by writing a 1 to CCIF to launch a command, and CCIF stays low until command completion or command violation. The CCIF flag is also cleared by a successful write to FlexRAM while enabled for EEE, and CCIF stays low until the EEPROM file system has created the associated EEPROM data record. The CCIF bit is reset to 0 but is set to 1 by the memory controller at the end of the reset initialization sequence. Depending on how quickly the read occurs after reset release, the user may or may not see the 0 hardware reset value. 0 Flash command or EEPROM file system operation in progress 1 Flash command or EEPROM file system operation has completed
6 RDCOLERR	Flash Read Collision Error Flag The RDCOLERR error bit indicates that the MCU attempted a read from a flash memory resource that was being manipulated by a flash command (CCIF=0). Any simultaneous access is detected as a collision

Table continues on the next page...

FTFL_FSTAT field descriptions (continued)

Field	Description
	error by the block arbitration logic. The read data in this case cannot be guaranteed. The RDCOLERR bit is cleared by writing a 1 to it. Writing a 0 to RDCOLERR has no effect. 0 No collision error detected 1 Collision error detected
5 ACCERR	Flash Access Error Flag The ACCERR error bit indicates an illegal access has occurred to a flash memory resource caused by a violation of the command write sequence or issuing an illegal flash command. While ACCERR is set, the CCIF flag cannot be cleared to launch a command. The ACCERR bit is cleared by writing a 1 to it. Writing a 0 to the ACCERR bit has no effect. 0 No access error detected 1 Access error detected
4 FPVIOL	Flash Protection Violation Flag The FPVIOL error bit indicates an attempt was made to program or erase an address in a protected area of program flash or data flash memory during a command write sequence or a write was attempted to a protected area of the FlexRAM while enabled for EEPROM . While FPVIOL is set, the CCIF flag cannot be cleared to launch a command. The FPVIOL bit is cleared by writing a 1 to it. Writing a 0 to the FPVIOL bit has no effect. 0 No protection violation detected 1 Protection violation detected
3–1 Reserved	This read-only field is reserved and always has the value zero.
0 MGSTAT0	Memory Controller Command Completion Status Flag The MGSTAT0 status flag is set if an error is detected during execution of a flash command or during the flash reset sequence. As a status flag, this bit cannot (and need not) be cleared by the user like the other error flags in this register. The value of the MGSTAT0 bit for "command-N" is valid only at the end of the "command-N" execution when CCIF=1 and before the next command has been launched. At some point during the execution of "command-N+1," the previous result is discarded and any previous error is cleared.

28.34.2 Flash Configuration Register (FTFL_FCNFG)

This register provides information on the current functional state of the flash memory module.

The erase control bits (ERSAREQ and ERSSUSP) have write restrictions. PFLSH, RAMRDY, and EEERDYare read-only status bits . The unassigned bits read as noted and are not writable. The reset values for the PFLASH, RAMRDY, and EEERDY bits are determined during the reset sequence.

Address: FTFL_FCNFG is 4002_0000h base + 1h offset = 4002_0001h

Bit	7	6	5	4	3	2	1	0
Read	CCIE	RDCOLLIE	ERSAREQ	ERSSUSP	0	PFLSH	RAMRDY	EEERDY
Write			▨		▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

FTFL_FCNFG field descriptions

Field	Description
7 CCIE	Command Complete Interrupt Enable The CCIE bit controls interrupt generation when a flash command completes. 0 Command complete interrupt disabled 1 Command complete interrupt enabled. An interrupt request is generated whenever the FSTAT[CCIF] flag is set.
6 RDCOLLIE	Read Collision Error Interrupt Enable The RDCOLLIE bit controls interrupt generation when a flash memory read collision error occurs. 0 Read collision error interrupt disabled 1 Read collision error interrupt enabled. An interrupt request is generated whenever a flash memory read collision error is detected (see the description of FSTAT[RDCOLERR]).
5 ERSAREQ	Erase All Request This bit issues a request to the memory controller to execute the Erase All Blocks command and release security. ERSAREQ is not directly writable but is under indirect user control. Refer to the device's Chip Configuration details on how to request this command. The ERSAREQ bit sets when an erase all request is triggered external to the flash memory module and CCIF is set (no command is currently being executed). ERSAREQ is cleared by the flash memory module when the operation completes. 0 No request or request complete 1 Request to: 1. run the Erase All Blocks command, 2. verify the erased state, 3. program the security byte in the Flash Configuration Field to the unsecure state, and 4. release MCU security by setting the FSEC[SEC] field to the unsecure state.
4 ERSSUSP	Erase Suspend The ERSSUSP bit allows the user to suspend (interrupt) the Erase Flash Sector command while it is executing. 0 No suspend requested 1 Suspend the current Erase Flash Sector command execution.
3 Reserved	This read-only field is reserved and always has the value zero.
2 PFLSH	Flash memory configuration 0 Flash memory module configured for FlexMemory that supports data flash and/or EEPROM 1 Reserved
1 RAMRDY	RAM Ready This flag indicates the current status of the FlexRAM .

Table continues on the next page...

FTFL_FCNFG field descriptions (continued)

Field	Description
	The state of the RAMRDY flag is normally controlled by the Set FlexRAM Function command. During the reset sequence, the RAMRDY flag is cleared if the FlexNVM block is partitioned for EEPROM and is set if the FlexNVM block is not partitioned for EEPROM. The RAMRDY flag is cleared if the Program Partition command is run to partition the FlexNVM block for EEPROM. The RAMRDY flag sets after completion of the Erase All Blocks command or execution of the erase-all operation triggered external to the flash memory module . 0 FlexRAM is not available for traditional RAM access. 1 FlexRAM is available as traditional RAM only; writes to the FlexRAM do not trigger EEPROM operations.
0 EEERDY	This flag indicates if the EEPROM backup data has been copied to the FlexRAM and is therefore available for read access. During the reset sequence, the EEERDY flag will remain cleared while CCIF is clear and will only set if the FlexNVM block is partitioned for EEPROM. 0 FlexRAM is not available for EEPROM operation. 1 FlexRAM is available for EEPROM operations where: • reads from the FlexRAM return data previously written to the FlexRAM in EEPROM mode and • writes to the FlexRAM clear EEERDY and launch an EEPROM operation to store the written data in the FlexRAM and EEPROM backup.

28.34.3 Flash Security Register (FTFL_FSEC)

This read-only register holds all bits associated with the security of the MCU and flash memory module.

During the reset sequence, the register is loaded with the contents of the flash security byte in the Flash Configuration Field located in program flash memory. The flash basis for the values is signified by X in the reset value.

Address: FTFL_FSEC is 4002_0000h base + 2h offset = 4002_0002h

Bit	7	6	5	4	3	2	1	0
Read	KEYEN		MEEN		FSLACC		SEC	
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FTFL_FSEC field descriptions

Field	Description
7–6 KEYEN	Backdoor Key Security Enable These bits enable and disable backdoor key access to the flash memory module. 00 Backdoor key access disabled

Table continues on the next page...

FTFL_FSEC field descriptions (continued)

Field	Description
	01 Backdoor key access disabled (preferred KEYEN state to disable backdoor key access) 10 Backdoor key access enabled 11 Backdoor key access disabled
5–4 MEEN	Mass Erase Enable Bits Enables and disables mass erase capability of the flash memory module. The state of the MEEN bits is only relevant when the SEC bits are set to secure outside of NVM Normal Mode. When the SEC field is set to unsecure, the MEEN setting does not matter. 00 Mass erase is enabled 01 Mass erase is enabled 10 Mass erase is disabled 11 Mass erase is enabled
3–2 FSLACC	Freescale Failure Analysis Access Code These bits enable or disable access to the flash memory contents during returned part failure analysis at Freescale. When SEC is secure and FSLACC is denied, access to the program flash contents is denied and any failure analysis performed by Freescale factory test must begin with a full erase to unsecure the part. When access is granted (SEC is unsecure, or SEC is secure and FSLACC is granted), Freescale factory testing has visibility of the current flash contents. The state of the FSLACC bits is only relevant when the SEC bits are set to secure. When the SEC field is set to unsecure, the FSLACC setting does not matter. 00 Freescale factory access granted 01 Freescale factory access denied 10 Freescale factory access denied 11 Freescale factory access granted
1–0 SEC	Flash Security These bits define the security state of the MCU. In the secure state, the MCU limits access to flash memory module resources. The limitations are defined per device and are detailed in the Chip Configuration details. If the flash memory module is unsecured using backdoor key access, the SEC bits are forced to 10b. 00 MCU security status is secure 01 MCU security status is secure 10 MCU security status is unsecure (The standard shipping condition of the flash memory module is unsecure.) 11 MCU security status is secure

28.34.4 Flash Option Register (FTFL_FOPT)

The flash option register allows the MCU to customize its operations by examining the state of these read-only bits, which are loaded from NVM at reset. The function of the bits is defined in the device's Chip Configuration details.

All bits in the register are read-only .

During the reset sequence, the register is loaded from the flash nonvolatile option byte in the Flash Configuration Field located in program flash memory. The flash basis for the values is signified by X in the reset value.

Address: FTFL_FOPT is 4002_0000h base + 3h offset = 4002_0003h

Bit	7	6	5	4	3	2	1	0
Read	OPT							
Write								
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FTFL_FOPT field descriptions

Field	Description
7–0 OPT	Nonvolatile Option These bits are loaded from flash to this register at reset. Refer to the device's Chip Configuration details for the definition and use of these bits.

28.34.5 Flash Common Command Object Registers (FTFL_FCCOB*n*)

The FCCOB register group provides 12 bytes for command codes and parameters. The individual bytes within the set append a 0-B hex identifier to the FCCOB register name: FCCOB0, FCCOB1, ..., FCCOBB.

Addresses: 4002_0000h base + 4h offset + (1d × *n*), where *n* = 0d to 11d

Bit	7	6	5	4	3	2	1	0
Read Write	CCOBn							
Reset	0	0	0	0	0	0	0	0

FTFL_FCCOB*n* field descriptions

Field	Description
7–0 CCOBn	The FCCOB register provides a command code and relevant parameters to the memory controller. The individual registers that compose the FCCOB data set can be written in any order, but you must provide all needed values, which vary from command to command. First, set up all required FCCOB fields and then initiate the command's execution by writing a 1 to the FSTAT[CCIF] bit. This clears the CCIF bit, which locks all FCCOB parameter fields and they cannot be changed by the user until the command completes (CCIF returns to 1). No command buffering or queueing is provided; the next command can be loaded only after the current command completes. Some commands return information to the FCCOB registers. Any values returned to FCCOB are available for reading after the FSTAT[CCIF] flag returns to 1 by the memory controller.

FTFL_FCCOB*n* field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>
<p>The following table shows a generic flash command format. The first FCCOB register, FCCOB0, always contains the command code. This 8-bit value defines the command to be executed. The command code is followed by the parameters required for this specific flash command, typically an address and/or data values.</p>
<p>NOTE: The command parameter table is written in terms of FCCOB Number (which is equivalent to the byte number). This number is a reference to the FCCOB register name and is not the register address.</p>
<table>
<tr><th>FCCOB Number</th><th>Typical Command Parameter Contents [7:0]</th></tr>
<tr><td>0</td><td>FCMD (a code that defines the flash command)</td></tr>
<tr><td>1</td><td>Flash address [23:16]</td></tr>
<tr><td>2</td><td>Flash address [15:8]</td></tr>
<tr><td>3</td><td>Flash address [7:0]</td></tr>
<tr><td>4</td><td>Data Byte 0</td></tr>
<tr><td>5</td><td>Data Byte 1</td></tr>
<tr><td>6</td><td>Data Byte 2</td></tr>
<tr><td>7</td><td>Data Byte 3</td></tr>
<tr><td>8</td><td>Data Byte 4</td></tr>
<tr><td>9</td><td>Data Byte 5</td></tr>
<tr><td>A</td><td>Data Byte 6</td></tr>
<tr><td>B</td><td>Data Byte 7</td></tr>
</table>
<p>FCCOB Endianness and Multi-Byte Access :</p>
<p>The FCCOB register group uses a big endian addressing convention. For all command parameter fields larger than 1 byte, the most significant data resides in the lowest FCCOB register number. The FCCOB register group may be read and written as individual bytes, aligned words (2 bytes) or aligned longwords (4 bytes).</p>
</td></tr>
</table>

28.34.6 Program Flash Protection Registers (FTFL_FPROT*n*)

The FPROT registers define which logical program flash regions are protected from program and erase operations. Protected flash regions cannot have their content changed; that is, these regions cannot be programmed and cannot be erased by any flash command. Unprotected regions can be changed by program and erase operations.

The four FPROT registers allow 32 protectable regions. Each bit protects a 1/32 region of the program flash memory. The bitfields are defined in each register as follows:

Program flash protection register	Program flash protection bits
FPROT0	PROT[31:24]
FPROT1	PROT[23:16]
FPROT2	PROT[15:8]
FPROT3	PROT[7:0]

During the reset sequence, the FPROT registers are loaded with the contents of the program flash protection bytes in the Flash Configuration Field as indicated in the following table.

Program flash protection register	Flash Configuration Field offset address
FPROT0	0x0008
FPROT1	0x0009
FPROT2	0x000A
FPROT3	0x000B

To change the program flash protection that is loaded during the reset sequence, unprotect the sector of program flash memory that contains the Flash Configuration Field. Then, reprogram the program flash protection byte.

Addresses: FPROT3 is 4002_0000h base + 10h offset = 4002_0010h
FPROT2 is 4002_0000h base + 11h offset = 4002_0011h
FPROT1 is 4002_0000h base + 12h offset = 4002_0012h
FPROT0 is 4002_0000h base + 13h offset = 4002_0013h

Bit	7	6	5	4	3	2	1	0
Read Write	PROT							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FTFL_FPROT*n* field descriptions

Field	Description
7–0 PROT	Program Flash Region Protect Each program flash region can be protected from program and erase operations by setting the associated PROT bit. **In NVM Normal mode:** The protection can only be increased, meaning that currently unprotected memory can be protected, but currently protected memory cannot be unprotected. Since unprotected regions are marked with a 1 and protected regions use a 0, only writes changing 1s to 0s are accepted. This 1-to-0 transition check is performed on a bit-by-bit basis. Those FPROT bits with 1-to-0 transitions are accepted while all bits with 0-to-1 transitions are ignored . **In NVM Special mode:** All bits of FPROT are writable without restriction. Unprotected areas can be protected and protected areas can be unprotected. **Restriction:** The user must never write to any FPROT register while a command is running (CCIF=0).

FTFL_FPROT*n* field descriptions (continued)

Field	Description
	Trying to alter data in any protected area in the program flash memory results in a protection violation error and sets the FSTAT[FPVIOL] bit. A full block erase of a program flash block is not possible if it contains any protected region. Each bit in the 32-bit protection register represents 1/32 of the total program flash . 0 Program flash region is protected. 1 Program flash region is not protected

28.34.7 EEPROM Protection Register (FTFL_FEPROT)

The FEPROT register defines which EEPROM regions of the FlexRAM are protected against program and erase operations. Protected EEPROM regions cannot have their content changed by writing to it. Unprotected regions can be changed by writing to the FlexRAM.

Address: FTFL_FEPROT is 4002_0000h base + 16h offset = 4002_0016h

Bit	7	6	5	4	3	2	1	0
Read Write	EPROT							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

FTFL_FEPROT field descriptions

Field	Description
7–0 EPROT	EEPROM Region Protect Individual EEPROM regions can be protected from alteration by setting the associated EPROT bit. The EPROT bits are not used when the FlexNVM Partition Code is set to data flash only. When the FlexNVM Partition Code is set to data flash and EEPROM or EEPROM only, each EPROT bit covers one-eighth of the configured EEPROM data (see the EEPROM Data Set Size parameter description). **In NVM Normal mode:** The protection can only be increased. This means that currently-unprotected memory can be protected, but currently-protected memory cannot be unprotected. Since unprotected regions are marked with a 1 and protected regions use a 0, only writes changing 1s to 0s are accepted. This 1-to-0 transition check is performed on a bit-by-bit basis. Those FEPROT bits with 1-to-0 transitions are accepted while all bits with 0-to-1 transitions are ignored . **In NVM Special mode :** All bits of the FEPROT register are writable without restriction. Unprotected areas can be protected and protected areas can be unprotected. **Restriction:** Never write to the FEPROT register while a command is running (CCIF=0).

FTFL_FEPROT field descriptions (continued)

Field	Description
	Reset: During the reset sequence, the FEPROT register is loaded with the contents of the FlexRAM protection byte in the Flash Configuration Field located in program flash. The flash basis for the reset values is signified by X in the register diagram. To change the EEPROM protection that will be loaded during the reset sequence, the sector of program flash that contains the Flash Configuration Field must be unprotected; then the EEPROM protection byte must be erased and reprogrammed. Trying to alter data by writing to any protected area in the EEPROM results in a protection violation error and sets the FPVIOL bit in the FSTAT register. 0 EEPROM region is protected 1 EEPROM region is not protected

28.34.8 Data Flash Protection Register (FTFL_FDPROT)

The FDPROT register defines which data flash regions are protected against program and erase operations. Protected Flash regions cannot have their content changed; that is, these regions cannot be programmed and cannot be erased by any flash command. Unprotected regions can be changed by both program and erase operations.

Address: FTFL_FDPROT is 4002_0000h base + 17h offset = 4002_0017h

Bit	7	6	5	4	3	2	1	0
Read Write	DPROT							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

FTFL_FDPROT field descriptions

Field	Description
7–0 DPROT	Data Flash Region Protect Individual data flash regions can be protected from program and erase operations by setting the associated DPROT bit. Each DPROT bit protects one-eighth of the partitioned data flash memory space. The granularity of data flash protection cannot be less than the data flash sector size. If an unused DPROT bit is set, the Erase all Blocks command does not execute and the FSTAT[FPVIOL] flag is set. **In NVM Normal mode:** The protection can only be increased, meaning that currently unprotected memory can be protected but currently protected memory cannot be unprotected. Since unprotected regions are marked with a 1 and protected regions use a 0, only writes changing 1s to 0s are accepted. This 1-to-0 transition check is performed on a bit-by-bit basis. Those FDPROT bits with 1-to-0 transitions are accepted while all bits with 0-to-1 transitions are ignored . **In NVM Special mode:** All bits of the FDPROT register are writable without restriction. Unprotected areas can be protected and protected areas can be unprotected. **Restriction:** The user must never write to the FDPROT register while a command is running (CCIF=0).

FTFL_FDPROT field descriptions (continued)

Field	Description
	Reset: During the reset sequence, the FDPROT register is loaded with the contents of the data flash protection byte in the Flash Configuration Field located in program flash memory. The flash basis for the reset values is signified by X in the register diagram. To change the data flash protection that will be loaded during the reset sequence, unprotect the sector of program flash that contains the Flash Configuration Field. Then, erase and reprogram the data flash protection byte. Trying to alter data with the program and erase commands in any protected area in the data flash memory results in a protection violation error and sets the FSTAT[FPVIOL] bit. A full block erase of the data flash memory (see the Erase Flash Block command description) is not possible if the data flash memory contains any protected region or if the FlexNVM block has been partitioned for EEPROM. 0 Data Flash region is protected 1 Data Flash region is not protected

28.4 Functional Description

The following sections describe functional details of the flash memory module.

28.4.1 Flash Protection

Individual regions within the flash memory can be protected from program and erase operations. Protection is controlled by the following registers:

- FPROT*n* — Four registers that protect 32 regions of the program flash memory as shown in the following figure

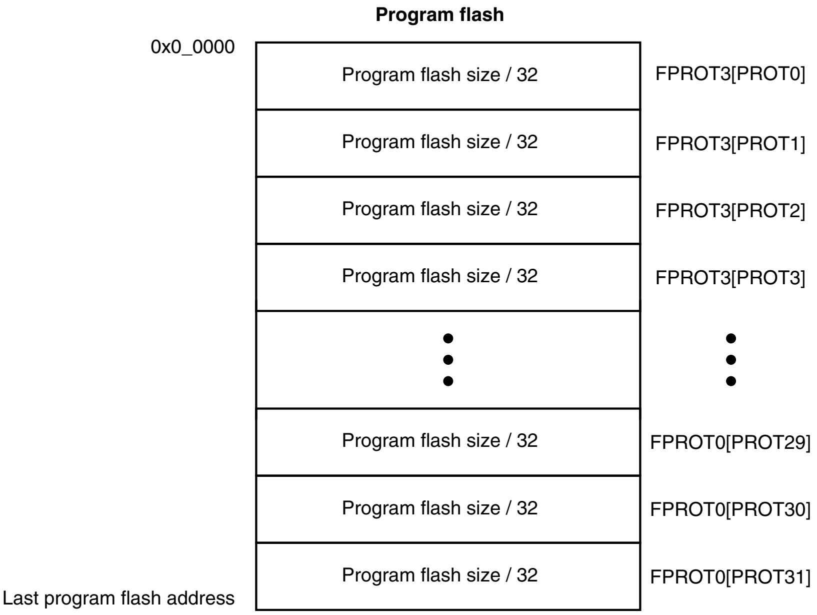

Program flash
0x0_0000
Program flash size / 32
FPROT3[PROT0]
Program flash size / 32
FPROT3[PROT1]
Program flash size / 32
FPROT3[PROT2]
Program flash size / 32
FPROT3[PROT3]
Program flash size / 32
FPROT0[PROT29]
Program flash size / 32
FPROT0[PROT30]
Program flash size / 32
FPROT0[PROT31]
Last program flash address

Figure 28-26. Program flash protection

- FDPROT —
  - protects eight regions of the data flash memory as shown in the following figure

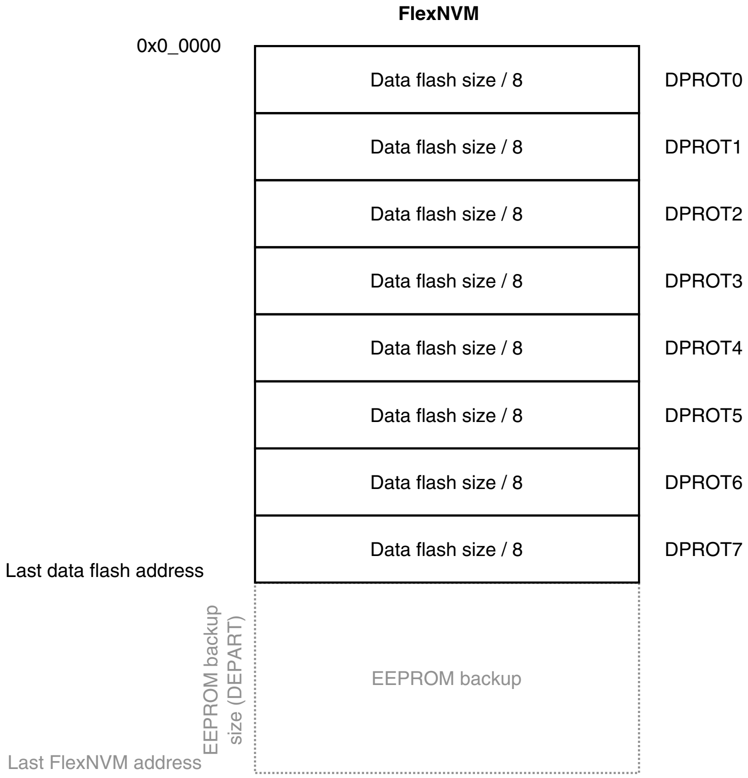

FlexNVM
0x0_0000
Data flash size / 8
DPROT0
Data flash size / 8
DPROT1
Data flash size / 8
DPROT2
Data flash size / 8
DPROT3
Data flash size / 8
DPROT4
Data flash size / 8
DPROT5
Data flash size / 8
DPROT6
Data flash size / 8
DPROT7
Last data flash address
EEPROM backup size (DEPART)
EEPROM backup
Last FlexNVM address

Figure 28-27. Data flash protection

- FEPROT — Protects eight regions of the EEPROM memory as shown in the following figure

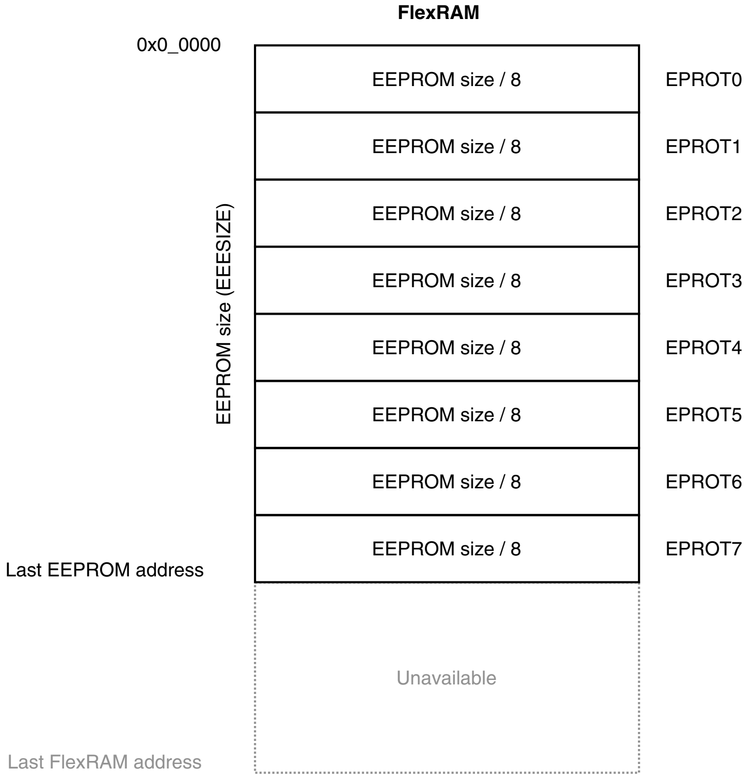

FlexRAM
0x0_0000
EEPROM size (EEESIZE)
EEPROM size / 8
EEPROM size / 8
EEPROM size / 8
EEPROM size / 8
EEPROM size / 8
EEPROM size / 8
EEPROM size / 8
EEPROM size / 8
EPROT0
EPROT1
EPROT2
EPROT3
EPROT4
EPROT5
EPROT6
EPROT7
Last EEPROM address
Unavailable
Last FlexRAM address

Figure 28-28. EEPROM protection

28.4.2 FlexNVM Description

This section describes the FlexNVM memory.

28.4.2.1 FlexNVM Block Partitioning for FlexRAM

The user can configure the FlexNVM block as either:

- Basic data flash,
- EEPROM flash records to support the built-in EEPROM feature, or
- A combination of both.

The user's FlexNVM configuration choice is specified using the Program Partition command described in Program Partition Command.

CAUTION

While different partitions of the FlexNVM block are available, the intention is that a single partition choice is used throughout the entire lifetime of a given application. The FlexNVM partition code choices affect the endurance and data retention characteristics of the device.

28.4.2.2 EEPROM User Perspective

The EEPROM system is shown in the following figure.

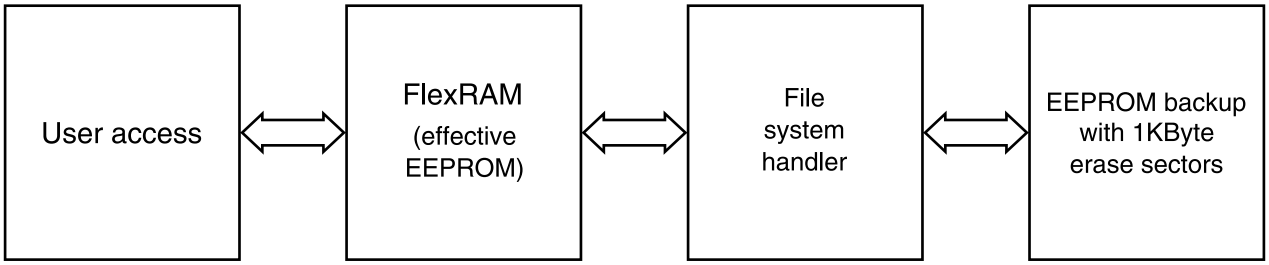


Figure 28-29. Top Level EEPROM Architecture

To handle varying customer requirements, the FlexRAM and FlexNVM blocks can be split into partitions as shown in the figure below.

1. **EEPROM partition** (EEESIZE) — The amount of FlexRAM used for EEPROM can be set from 0 Bytes (no EEPROM) to the maximum FlexRAM size (see Table 28-2). The remainder of the FlexRAM is not accessible while the FlexRAM is configured for EEPROM (see Set FlexRAM Function Command). The EEPROM partition grows upward from the bottom of the FlexRAM address space.
2. **Data flash partition** (DEPART) — The amount of FlexNVM memory used for data flash can be programmed from 0 bytes (all of the FlexNVM block is available for EEPROM backup) to the maximum size of the FlexNVM block (see Table 28-4).
3. **FlexNVM EEPROM partition** — The amount of FlexNVM memory used for EEPROM backup, which is equal to the FlexNVM block size minus the data flash memory partition size. The EEPROM backup size must be at least 16 times the EEPROM partition size in FlexRAM.

The partition information (EEESIZE, DEPART) is stored in the data flash IFR and is programmed using the Program Partition command (see Program Partition Command). Typically, the Program Partition command is executed only once in the lifetime of the device.

Data flash memory is useful for applications that need to quickly store large amounts of data or store data that is static. The EEPROM partition in FlexRAM is useful for storing smaller amounts of data that will be changed often.

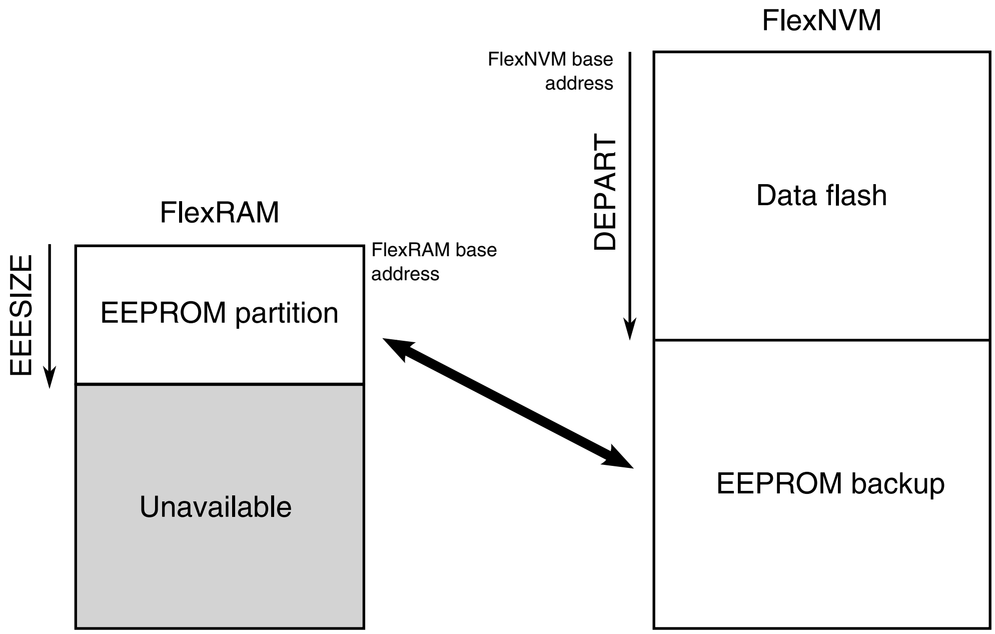


Figure 28-30. FlexRAM to FlexNVM Memory Mapping

28.4.2.3 EEPROM Implementation Overview

Out of reset with the FSTAT[CCIF] bit clear, the partition settings (EEESIZE, DEPART) are read from the data flash IFR and the EEPROM file system is initialized accordingly. The EEPROM file system locates all valid EEPROM data records in EEPROM backup and copies the newest data to FlexRAM. The FSTAT[CCIF] and FCNFG[EEERDY] bits are set after data from all valid EEPROM data records is copied to the FlexRAM. After the CCIF bit is set, the FlexRAM is available for read or write access.

When configured for EEPROM use, writes to an unprotected location in FlexRAM invokes the EEPROM file system to program a new EEPROM data record in the EEPROM backup memory in a round-robin fashion. As needed, the EEPROM file system identifies the EEPROM backup sector that is being erased for future use and partially erases that EEPROM backup sector. After a write to the FlexRAM, the FlexRAM is not accessible until the FSTAT[CCIF] bit is set. The FCNFG[EEERDY] bit will also be set. If enabled, the interrupt associated with the FSTAT[CCIF] bit can be used to determine when the FlexRAM is available for read or write access.

After a sector in EEPROM backup is full of EEPROM data records, EEPROM data records from the sector holding the oldest data are gradually copied over to a previously-erased EEPROM backup sector. When the sector copy completes, the EEPROM backup sector holding the oldest data is tagged for erase.

28.4.2.4 Write endurance to FlexRAM for EEPROM

When the FlexNVM partition code is not set to full data flash, the EEPROM data set size can be set to any of several non-zero values.

The bytes not assigned to data flash via the FlexNVM partition code are used by the FTFL to obtain an effective endurance increase for the EEPROM data. The built-in EEPROM record management system raises the number of program/erase cycles that can be attained prior to device wear-out by cycling the EEPROM data through a larger EEPROM NVM storage space.

While different partitions of the FlexNVM are available, the intention is that a single choice for the FlexNVM partition code and EEPROM data set size is used throughout the entire lifetime of a given application. The EEPROM endurance equation and graph shown below assume that only one configuration is ever used.

$$\text{Writes\_FlexRAM} = \frac{\text{EEPROM} - 2 \times \text{EEESIZE}}{\text{EEESIZE}} \times \text{Write\_efficiency} \times n_{\text{nvmcycd}}$$

where

- Writes_FlexRAM — minimum number of writes to each FlexRAM location
- EEPROM — allocated FlexNVM based on DEPART; entered with Program Partition command
- EEESIZE — allocated FlexRAM based on DEPART; entered with Program Partition command
- Write_efficiency —
  - 0.25 for 8-bit writes to FlexRAM
  - 0.50 for 16-bit or 32-bit writes to FlexRAM
- n_{nvmcycd} — data flash cycling endurance

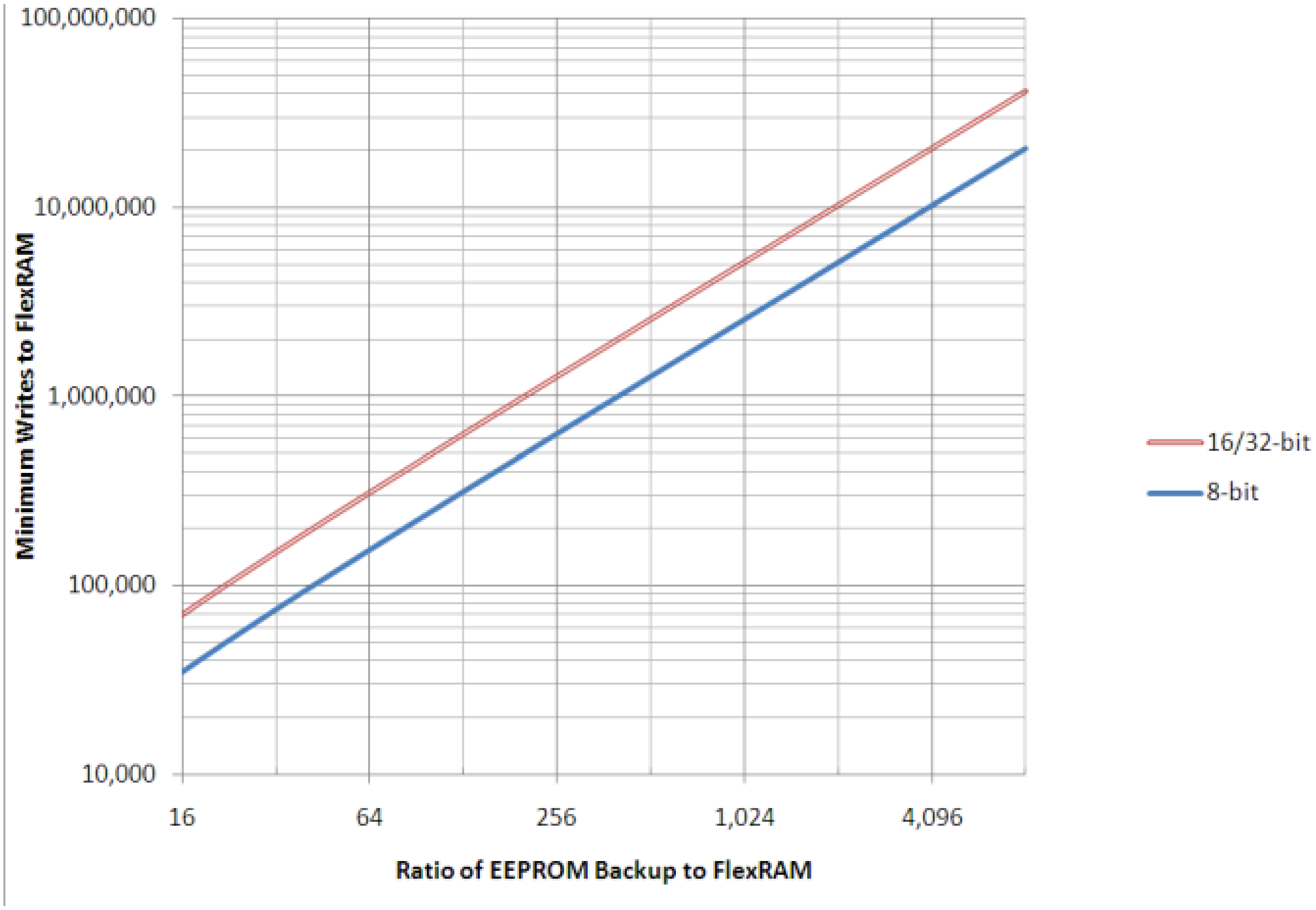


Figure 28-31. EEPROM backup writes to FlexRAM

28.4.3 Interrupts

The flash memory module can generate interrupt requests to the MCU upon the occurrence of various flash events. These interrupt events and their associated status and control bits are shown in the following table.

Table 28-30. Flash Interrupt Sources

Flash Event	Readable Status Bit	Interrupt Enable Bit
Flash Command Complete	FSTAT[CCIF]	FCNFG[CCIE]
Flash Read Collision Error	FSTAT[RDCOLERR]	FCNFG[RDCOLLIE]

Note

Vector addresses and their relative interrupt priority are determined at the MCU level.

28.4.4 Flash Operation in Low-Power Modes

28.4.4.1 Wait Mode

When the MCU enters wait mode, the flash memory module is not affected. The flash memory module can recover the MCU from wait via the command complete interrupt (see Interrupts).

28.4.4.2 Stop Mode

When the MCU requests stop mode, if a flash command is active (CCIF = 0) the command execution completes before the MCU is allowed to enter stop mode.

CAUTION

The MCU should never enter stop mode while any flash command is running (CCIF = 0).

NOTE

While the MCU is in very-low-power modes (VLPR, VLPW, VLPS), the flash memory module does not accept flash commands.

28.4.5 Functional Modes of Operation

The flash memory module has two operating modes: NVM Normal and NVM Special. The operating mode affects the command set availability (see Table 28-31). Refer to the Chip Configuration details of this device for how to activate each mode.

28.4.6 Flash Reads and Ignored Writes

The flash memory module requires only the flash address to execute a flash memory read. MCU read access is available to all flash blocks.

The MCU must not read from the flash memory while commands are running (as evidenced by CCIF=0) on that block. Read data cannot be guaranteed from a flash block while any command is processing within that block. The block arbitration logic detects any simultaneous access and reports this as a read collision error (see the FSTAT[RDCOLERR] bit).

28.4.7 Read While Write (RWW)

The following simultaneous accesses are allowed:

- The user may read from the program flash memory while commands (typically program and erase operations) are active in the data flash and FlexRAM memory space.
- The MCU can fetch instructions from program flash during both data flash program and erase operations and while EEPROM backup data is maintained by the EEPROM commands.
- Conversely, the user may read from data flash and FlexRAM while program and erase commands are executing on the program flash.
- When configured as traditional RAM, writes to the FlexRAM are allowed during program and data flash operations.

Simultaneous data flash operations and FlexRAM writes, when FlexRAM is used for EEPROM, are not possible.

Simultaneous operations are further discussed in Allowed Simultaneous Flash Operations.

28.4.8 Flash Program and Erase

All flash functions except read require the user to setup and launch a flash command through a series of peripheral bus writes. The user cannot initiate any further flash commands until notified that the current command has completed. The flash command structure and operation are detailed in Flash Command Operations.

28.4.9 Flash Command Operations

Flash command operations are typically used to modify flash memory contents. The next sections describe:

- The command write sequence used to set flash command parameters and launch execution

- A description of all flash commands available

28.4.9.1 Command Write Sequence

Flash commands are specified using a command write sequence illustrated in Figure 28-32. The flash memory module performs various checks on the command (FCCOB) content and continues with command execution if all requirements are fulfilled.

Before launching a command, the ACCERR and FPVIOL bits in the FSTAT register must be zero and the CCIF flag must read 1 to verify that any previous command has completed. If CCIF is zero, the previous command execution is still active, a new command write sequence cannot be started, and all writes to the FCCOB registers are ignored.

28.4.9.1.1 Load the FCCOB Registers

The user must load the FCCOB registers with all parameters required by the desired flash command. The individual registers that make up the FCCOB data set can be written in any order.

28.4.9.1.2 Launch the Command by Clearing CCIF

Once all relevant command parameters have been loaded, the user launches the command by clearing the FSTAT[CCIF] bit by writing a '1' to it. The CCIF flag remains zero until the flash command completes.

The FSTAT register contains a blocking mechanism, which prevents a new command from launching (can't clear CCIF) if the previous command resulted in an access error (FSTAT[ACCERR]=1) or a protection violation (FSTAT[FPVIOL]=1). In error scenarios, two writes to FSTAT are required to initiate the next command: the first write clears the error flags, the second write clears CCIF.

28.4.9.1.3 Command Execution and Error Reporting

The command processing has several steps:

1. The flash memory module reads the command code and performs a series of parameter checks and protection checks, if applicable, which are unique to each command.

   If the parameter check fails, the FSTAT[ACCERR] (access error) flag is set. ACCERR reports invalid instruction codes and out-of bounds addresses. Usually, access errors suggest that the command was not set-up with valid parameters in the FCCOB register group.

Program and erase commands also check the address to determine if the operation is requested to execute on protected areas. If the protection check fails, the FSTAT[FPVIOL] (protection error) flag is set.

Command processing never proceeds to execution when the parameter or protection step fails. Instead, command processing is terminated after setting the FSTAT[CCIF] bit.

2. If the parameter and protection checks pass, the command proceeds to execution. Run-time errors, such as failure to erase verify, may occur during the execution phase. Run-time errors are reported in the FSTAT[MGSTAT0] bit. A command may have access errors, protection errors, and run-time errors, but the run-time errors are not seen until all access and protection errors have been corrected.
3. Command execution results, if applicable, are reported back to the user via the FCCOB and FSTAT registers.
4. The flash memory module sets the FSTAT[CCIF] bit signifying that the command has completed.

The flow for a generic command write sequence is illustrated in the following figure.

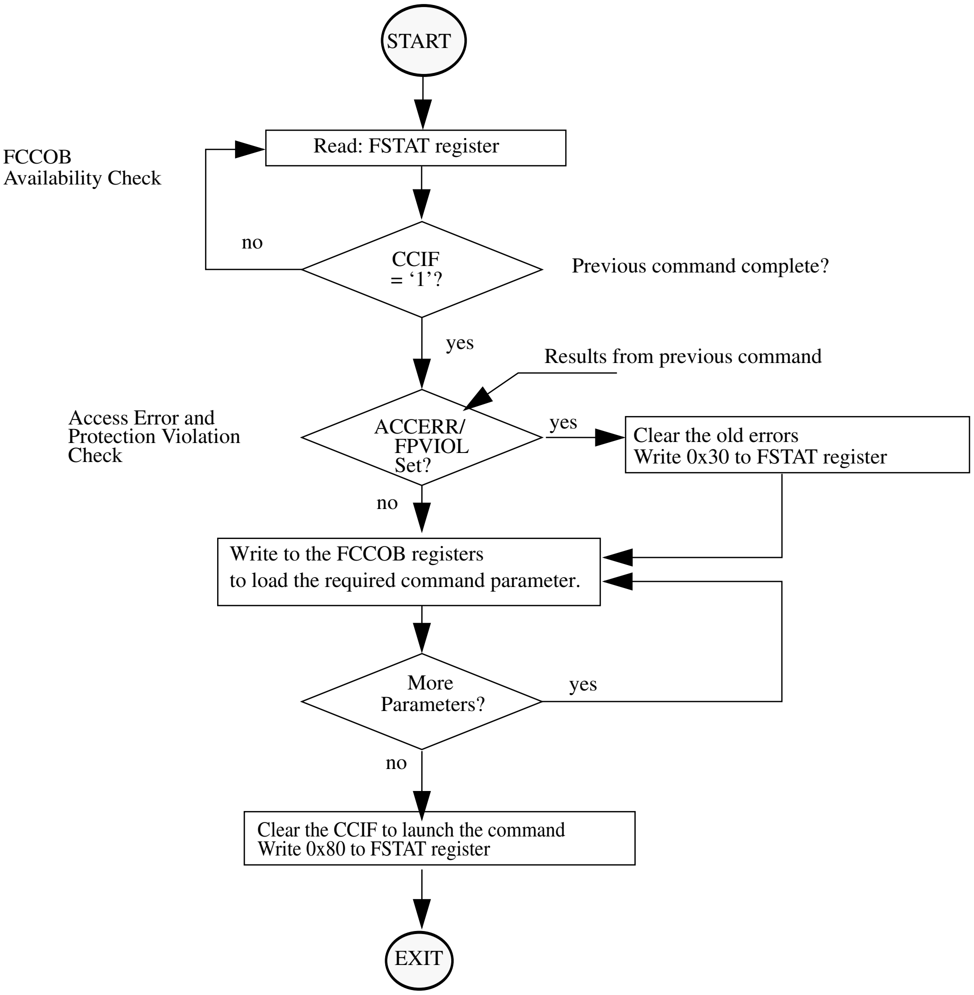

START
Read: FSTAT register
FCCOB
Availability Check
no
CCIF
= ‘1’?
Previous command complete?
yes
Results from previous command
Access Error and
Protection Violation
Check
ACCERR/
FPVIOL
Set?
yes
Clear the old errors
Write 0x30 to FSTAT register
no
Write to the FCCOB registers
to load the required command parameter.
More
Parameters?
yes
no
Clear the CCIF to launch the command
Write 0x80 to FSTAT register
EXIT

Figure 28-32. Generic Flash Command Write Sequence Flowchart

28.4.9.2 Flash Commands

The following table summarizes the function of all flash commands. If the program flash, data flash, or FlexRAM column is marked with an 'X', the flash command is relevant to that particular memory resource.

FCMD	Command	Program flash	Data flash	FlexRAM	Function
0x00	Read 1s Block	×	×		Verify that a program flash or data flash block is erased. FlexNVM block must not be partitioned for EEPROM.

Table continues on the next page...

FCMD	Command	Program flash	Data flash	FlexRAM	Function
0x01	Read 1s Section	×	×		Verify that a given number of program flash or data flash locations from a starting address are erased.
0x02	Program Check	×	×		Tests previously-programmed locations at margin read levels.
0x03	Read Resource	IFR, ID	IFR		Read 4 bytes from program flash IFR, data flash IFR, or version ID.
0x06	Program Longword	×	×		Program 4 bytes in a program flash block or a data flash block.
0x08	Erase Flash Block	×	×		Erase a program flash block or data flash block. An erase of any flash block is only possible when unprotected. FlexNVM block must not be partitioned for EEPROM.
0x09	Erase Flash Sector	×	×		Erase all bytes in a program flash or data flash sector.
0x0B	Program Section	×	×	×	Program data from the Section Program Buffer to a program flash or data flash block.
0x40	Read 1s All Blocks	×	×		Verify that all program flash, data flash blocks, EEPROM backup data records, and data flash IFR are erased then release MCU security.
0x41	Read Once	IFR			Read 4 bytes of a dedicated 64 byte field in the program flash IFR.

Table continues on the next page...

FCMD	Command	Program flash	Data flash	FlexRAM	Function
0x43	Program Once	IFR			One-time program of 4 bytes of a dedicated 64-byte field in the program flash IFR.
0x44	Erase All Blocks	×	×	×	Erase all program flash blocks, data flash blocks, FlexRAM, EEPROM backup data records, and data flash IFR. Then, verify-erase and release MCU security. **NOTE:** An erase is only possible when all memory locations are unprotected.
0x45	Verify Backdoor Access Key	×			Release MCU security after comparing a set of user-supplied security keys to those stored in the program flash.
0x80	Program Partition		IFR	×	Program the FlexNVM Partition Code and EEPROM Data Set Size into the data flash IFR. Format all EEPROM backup data sectors allocated for EEPROM. Initialize the FlexRAM.
0x81	Set FlexRAM Function		x	×	Switches FlexRAM function between RAM and EEPROM. When switching to EEPROM, FlexNVM is not available while valid data records are being copied from EEPROM backup to FlexRAM.

28.4.9.3 Flash Commands by Mode

The following table shows the flash commands that can be executed in each flash operating mode.

Table 28-31. Flash Commands by Mode

FCMD	Command	NVM Normal			NVM Special		
		Unsecure	Secure	MEEN=10	Unsecure	Secure	MEEN=10
0x00	Read 1s Block	×	×	×	×	—	—
0x01	Read 1s Section	×	×	×	×	—	—
0x02	Program Check	×	×	×	×	—	—
0x03	Read Resource	×	×	×	×	—	—
0x06	Program Longword	×	×	×	×	—	—
0x08	Erase Flash Block	×	×	×	×	—	—
0x09	Erase Flash Sector	×	×	×	×	—	—
0x0B	Program Section	×	×	×	×	—	—
0x40	Read 1s All Blocks	×	×	×	×	×	—
0x41	Read Once	×	×	×	×	—	—
0x43	Program Once	×	×	×	×	—	—
0x44	Erase All Blocks	×	×	×	×	×	—
0x45	Verify Backdoor Access Key	×	×	×	×	—	—
0x80	Program Partition	×	×	×	×	—	—
0x81	Set FlexRAM Function	×	×	×	×	—	—

28.4.9.4 Allowed Simultaneous Flash Operations

Only the operations marked 'OK' in the following table are permitted to run simultaneously on the program flash, data flash, and FlexRAM memories. Some operations cannot be executed simultaneously because certain hardware resources are shared by the memories. The priority has been placed on permitting program flash reads while program and erase operations execute on the FlexNVM and FlexRAM. This provides read (program flash) while write (FlexNVM, FlexRAM) functionality.

Table 28-32. Allowed Simultaneous Memory Operations

		Program Flash			Data Flash			FlexRAM		
		Read	Program	Sector Erase	Read	Program	Sector Erase	Read	E-Write[1]	R-Write[2]
Program flash	**Read**	—				OK	OK		OK	
	Program		—		OK			OK		OK[3]
	Sector Erase			—	OK			OK		OK
Data flash	**Read**		OK	OK	—					
	Program	OK				—		OK		OK
	Sector Erase	OK					—	OK		OK
FlexRAM	**Read**		OK	OK		OK	OK	—		
	E-Write[1]	OK							—	
	R-Write[2]		OK	OK		OK	OK			—

1. When FlexRAM configured for EEPROM (writes are effectively multi-cycle operations).
2. When FlexRAM configured as traditional RAM (writes are single-cycle operations).
3. When FlexRAM configured as traditional RAM, writes to the RAM are ignored while the Program Section command is active (CCIF = 0).

28.4.10 Margin Read Commands

The Read-1s commands (Read 1s All Blocks, Read 1s Block, and Read 1s Section) and the Program Check command have a margin choice parameter that allows the user to apply non-standard read reference levels to the program flash and data flash array reads performed by these commands. Using the preset 'user' and 'factory' margin levels, these commands perform their associated read operations at tighter tolerances than a 'normal' read. These non-standard read levels are applied only during the command execution. All simple (uncommanded) flash array reads to the MCU always use the standard, un-margined, read reference level.

Only the 'normal' read level should be employed during normal flash usage. The non-standard, 'user' and 'factory' margin levels should be employed only in special cases. They can be used during special diagnostic routines to gain confidence that the device is not suffering from the end-of-life data loss customary of flash memory devices.

Erased ('1') and programmed ('0') bit states can degrade due to elapsed time and data cycling (number of times a bit is erased and re-programmed). The lifetime of the erased states is relative to the last erase operation. The lifetime of the programmed states is measured from the last program time.

The 'user' and 'factory' levels become, in effect, a minimum safety margin; i.e. if the reads pass at the tighter tolerances of the 'user' and 'factory' margins, then the 'normal' reads have at least this much safety margin before they experience data loss.

The 'user' margin is a small delta to the normal read reference level. 'User' margin levels can be employed to check that flash memory contents have adequate margin for normal level read operations. If unexpected read results are encountered when checking flash memory contents at the 'user' margin levels, loss of information might soon occur during 'normal' readout.

The 'factory' margin is a bigger deviation from the norm, a more stringent read criteria that should only be attempted immediately (or very soon) after completion of an erase or program command, early in the cycling life. 'Factory' margin levels can be used to check that flash memory contents have adequate margin for long-term data retention at the normal level setting. If unexpected results are encountered when checking flash memory contents at 'factory' margin levels, the flash memory contents should be erased and reprogrammed.

CAUTION

> Factory margin levels must only be used during verify of the initial factory programming.

28.4.11 Flash Command Description

This section describes all flash commands that can be launched by a command write sequence. The flash memory module sets the FSTAT[ACCERR] bit and aborts the command execution if any of the following illegal conditions occur:

- There is an unrecognized command code in the FCCOB FCMD field.
- There is an error in a FCCOB field for the specific commands. Refer to the error handling table provided for each command.

Ensure that the ACCERR and FPVIOL bits in the FSTAT register are cleared prior to starting the command write sequence. As described in Launch the Command by Clearing CCIF, a new command cannot be launched while these error flags are set.

Do not attempt to read a flash block while the flash memory module is running a command (CCIF = 0) on that same block. The flash memory module may return invalid data to the MCU with the collision error flag (FSTAT[RDCOLERR]) set.

When required by the command, address bit 23 selects between:

- program flash (=0) block
- data flash (=1) block

CAUTION

> Flash data must be in the erased state before being programmed. Cumulative programming of bits (adding more zeros) is not allowed.

28.4.11.1 Read 1s Block Command

The Read 1s Block command checks to see if an entire program flash or data flash block has been erased to the specified margin level. The FCCOB flash address bits determine which logical block is erase-verified.

Table 28-33. Read 1s Block Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x00 (RD1BLK)
1	Flash address [23:16] in the flash block to be verified
2	Flash address [15:8] in the flash block to be verified
3	Flash address [7:0][1] in the flash block to be verified
4	Read-1 Margin Choice

[1] Must be longword aligned (Flash address [1:0] = 00).

After clearing CCIF to launch the Read 1s Block command, the flash memory module sets the read margin for 1s according to Table 28-34 and then reads all locations within the selected program flash or data flash block.

When the data flash is targeted, DEPART must be set for no EEPROM, else the Read 1s Block command aborts setting the FSTAT[ACCERR] bit. If the flash memory module fails to read all 1s (i.e. the flash block is not fully erased), the FSTAT[MGSTAT0] bit is set. The CCIF flag sets after the Read 1s Block operation has completed.

Table 28-34. Margin Level Choices for Read 1s Block

Read Margin Choice	Margin Level Description
0x00	Use the 'normal' read level for 1s
0x01	Apply the 'User' margin to the normal read-1 level
0x02	Apply the 'Factory' margin to the normal read-1 level

Table 28-35. Read 1s Block Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid margin choice is specified	FSTAT[ACCERR]
Program flash is selected and the address is out of program flash range	FSTAT[ACCERR]
Data flash is selected and the address is out of data flash range	FSTAT[ACCERR]
Data flash is selected with EEPROM enabled	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]
Read-1s fails	FSTAT[MGSTAT0]

28.4.11.2 Read 1s Section Command

The Read 1s Section command checks if a section of program flash or data flash memory is erased to the specified read margin level. The Read 1s Section command defines the starting address and the number of phrases or longwords to be verified.

Table 28-36. Read 1s Section Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x01 (RD1SEC)
1	Flash address [23:16] of the first longword to be verified
2	Flash address [15:8] of the first longword to be verified
3	Flash address [7:0][1] of the first longword to be verified
4	Number of longwords to be verified [15:8]
5	Number of longwords to be verified [7:0]
6	Read-1 Margin Choice

[1] Must be longword aligned (Flash address [1:0] = 00).

Upon clearing CCIF to launch the Read 1s Section command, the flash memory module sets the read margin for 1s according to Table 28-37 and then reads all locations within the specified section of flash memory. If the flash memory module fails to read all 1s (i.e. the flash section is not erased), the FSTAT(MGSTAT0) bit is set. The CCIF flag sets after the Read 1s Section operation completes.

Table 28-37. Margin Level Choices for Read 1s Section

Read Margin Choice	Margin Level Description
0x00	Use the 'normal' read level for 1s
0x01	Apply the 'User' margin to the normal read-1 level
0x02	Apply the 'Factory' margin to the normal read-1 level

Table 28-38. Read 1s Section Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid margin code is supplied	FSTAT[ACCERR]
An invalid flash address is supplied	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]
The requested section crosses a Flash block boundary	FSTAT[ACCERR]
The requested number of longwords is zero	FSTAT[ACCERR]
Read-1s fails	FSTAT[MGSTAT0]

28.4.11.3 Program Check Command

The Program Check command tests a previously programmed program flash or data flash longword to see if it reads correctly at the specified margin level.

Table 28-39. Program Check Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x02 (PGMCHK)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0][1]
4	Margin Choice
8	Byte 0 expected data
9	Byte 1 expected data
A	Byte 2 expected data
B	Byte 3 expected data

[1] Must be longword aligned (Flash address [1:0] = 00).

Upon clearing CCIF to launch the Program Check command, the flash memory module sets the read margin for 1s according to Table 28-40, reads the specified longword, and compares the actual read data to the expected data provided by the FCCOB. If the comparison at margin-1 fails, the MGSTAT0 bit is set.

The flash memory module then sets the read margin for 0s, re-reads, and compares again. If the comparison at margin-0 fails, the MGSTAT0 bit is set. The CCIF flag is set after the Program Check operation completes.

The supplied address must be longword aligned (the lowest two bits of the byte address must be 00):

- Byte 0 data is expected at the supplied address ('start'),

- Byte 1 data is expected at byte address start + 0b01,
- Byte 2 data is expected at byte address start + 0b10, and
- Byte 3 data is expected at byte address start + 0b11.

NOTE

See the description of margin reads, Margin Read Commands

Table 28-40. Margin Level Choices for Program Check

Read Margin Choice	Margin Level Description
0x01	Read at 'User' margin-1 and 'User' margin-0
0x02	Read at 'Factory' margin-1 and 'Factory' margin-0

Table 28-41. Program Check Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid flash address is supplied	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]
An invalid margin choice is supplied	FSTAT[ACCERR]
Either of the margin reads does not match the expected data	FSTAT[MGSTAT0]

28.4.11.4 Read Resource Command

The Read Resource command allows the user to read data from special-purpose memory resources located within the flash memory module. The special-purpose memory resources available include program flash IFR space, data flash IFR space, and the Version ID field. Each resource is assigned a select code as shown in Table 28-43.

Table 28-42. Read Resource Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x03 (RDRSRC)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0][1]
Returned Values	
4	Read Data [31:24]
5	Read Data [23:16]
6	Read Data [15:8]
7	Read Data [7:0]

Table continues on the next page...

Table 28-42. Read Resource Command FCCOB Requirements (continued)

FCCOB Number	FCCOB Contents [7:0]
User-provided values	
8	Resource Select Code (see Table 28-43)

1. Must be longword aligned (Flash address [1:0] = 00).

Table 28-43. Read Resource Select Codes

Resource Select Code[1]	Description	Resource Size	Local Address Range
0x00	IFR	256 Bytes	0x0000 - 0x00FF
0x01[2]	Version ID	8 Bytes	0x0000 - 0x0007

1. Flash address [23] selects between program flash (=0) and data flash (=1) resources.
2. Located in program flash 0 reserved space; Flash address [23] = 0

After clearing CCIF to launch the Read Resource command, four consecutive bytes are read from the selected resource at the provided relative address and stored in the FCCOB register. The CCIF flag sets after the Read Resource operation completes. The Read Resource command exits with an access error if an invalid resource code is provided or if the address for the applicable area is out-of-range.

Table 28-44. Read Resource Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid resource code is entered	FSTAT[ACCERR]
Flash address is out-of-range for the targeted resource.	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]

28.4.11.5 Program Longword Command

The Program Longword command programs four previously-erased bytes in the program flash memory or in the data flash memory using an embedded algorithm.

CAUTION

A flash memory location must be in the erased state before being programmed. Cumulative programming of bits (back-to-back program operations without an intervening erase) within a flash memory location is not allowed. Re-programming of existing 0s to 0 is not allowed as this overstresses the device.

Table 28-45. Program Longword Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x06 (PGM4)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0][1]
4	Byte 0 program value
5	Byte 1 program value
6	Byte 2 program value
7	Byte 3 program value

[1] Must be longword aligned (Flash address [1:0] = 00).

Upon clearing CCIF to launch the Program Longword command, the flash memory module programs the data bytes into the flash using the supplied address. The targeted flash locations must be currently unprotected (see the description of the FPROT and FDPROT registers) to permit execution of the Program Longword operation.

The programming operation is unidirectional. It can only move NVM bits from the erased state ('1') to the programmed state ('0'). Erased bits that fail to program to the '0' state are flagged as errors in MGSTAT0. The CCIF flag is set after the Program Longword operation completes.

The supplied address must be longword aligned (flash address [1:0] = 00):

- Byte 0 data is written to the supplied address ('start'),
- Byte 1 data is programmed to byte address start+0b01,
- Byte 2 data is programmed to byte address start+0b10, and
- Byte 3 data is programmed to byte address start+0b11.

Table 28-46. Program Longword Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid flash address is supplied	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]
Flash address points to a protected area	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

28.4.11.6 Erase Flash Block Command

The Erase Flash Block operation erases all addresses in a single program flash or data flash block.

Table 28-47. Erase Flash Block Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x08 (ERSBLK)
1	Flash address [23:16] in the flash block to be erased
2	Flash address [15:8] in the flash block to be erased
3	Flash address [7:0][1] in the flash block to be erased

[1] Must be longword aligned (Flash address [1:0] = 00).

Upon clearing CCIF to launch the Erase Flash Block command, the flash memory module erases the main array of the selected flash block and verifies that it is erased. When the data flash is targeted, DEPART must be set for no EEPROM (see Table 28-4) else the Erase Flash Block command aborts setting the FSTAT[ACCERR] bit. The Erase Flash Block command aborts and sets the FSTAT[FPVIOL] bit if any region within the block is protected (see the description of the FPROT and FDPROT registers). If the erase verify fails, the MGSTAT0 bit in FSTAT is set. The CCIF flag will set after the Erase Flash Block operation has completed.

Table 28-48. Erase Flash Block Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
Program flash is selected and the address is out of program flash range	FSTAT[ACCERR]
Data flash is selected and the address is out of data flash range	FSTAT[ACCERR]
Data flash is selected with EEPROM enabled	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]
Any area of the selected flash block is protected	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

28.4.11.7 Erase Flash Sector Command

The Erase Flash Sector operation erases all addresses in a flash sector.

Table 28-49. Erase Flash Sector Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x09 (ERSSCR)

Table continues on the next page...

Table 28-49. Erase Flash Sector Command FCCOB Requirements (continued)

FCCOB Number	FCCOB Contents [7:0]
1	Flash address [23:16] in the flash sector to be erased
2	Flash address [15:8] in the flash sector to be erased
3	Flash address [7:0][1] in the flash sector to be erased

[1] Must be longword aligned (flash address [1:0] = 00).

After clearing CCIF to launch the Erase Flash Sector command, the flash memory module erases the selected program flash or data flash sector and then verifies that it is erased. The Erase Flash Sector command aborts if the selected sector is protected (see the description of the FPROT and FDPROT registers). If the erase-verify fails the FSTAT[MGSTAT0] bit is set. The CCIF flag is set after the Erase Flash Sector operation completes. The Erase Flash Sector command is suspendable (see the FCNFG[ERSSUSP] bit and Figure 28-33).

Table 28-50. Erase Flash Sector Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid Flash address is supplied	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]
The selected program flash or data flash sector is protected	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

28.4.11.7.1 Suspending an Erase Flash Sector Operation

To suspend an Erase Flash Sector operation set the FCNFG[ERSSUSP] bit (see Flash Configuration Field Description) when CCIF is clear and the CCOB command field holds the code for the Erase Flash Sector command. During the Erase Flash Sector operation (see Erase Flash Sector Command), the flash memory module samples the state of the ERSSUSP bit at convenient points. If the flash memory module detects that the ERSSUSP bit is set, the Erase Flash Sector operation is suspended and the flash memory module sets CCIF. While ERSSUSP is set, all writes to flash registers are ignored except for writes to the FSTAT and FCNFG registers.

If an Erase Flash Sector operation effectively completes before the flash memory module detects that a suspend request has been made, the flash memory module clears the ERSSUSP bit prior to setting CCIF. When an Erase Flash Sector operation has been successfully suspended, the flash memory module sets CCIF and leaves the ERSSUSP bit set. While CCIF is set, the ERSSUSP bit can only be cleared to prevent the withdrawal of a suspend request before the flash memory module has acknowledged it.

28.4.11.7.2 Resuming a Suspended Erase Flash Sector Operation

If the ERSSUSP bit is still set when CCIF is cleared to launch the next command, the previous Erase Flash Sector operation resumes. The flash memory module acknowledges the request to resume a suspended operation by clearing the ERSSUSP bit. A new suspend request can then be made by setting ERSSUSP. A single Erase Flash Sector operation can be suspended and resumed multiple times.

There is a minimum elapsed time limit between the request to resume the Erase Flash Sector operation (CCIF is cleared) and the request to suspend the operation again (ERSSUSP is set). This minimum time period is required to ensure that the Erase Flash Sector operation will eventually complete. If the minimum period is continually violated, i.e. the suspend requests come repeatedly and too quickly, no forward progress is made by the Erase Flash Sector algorithm. The resume/suspend sequence runs indefinitely without completing the erase.

28.4.11.7.3 Aborting a Suspended Erase Flash Sector Operation

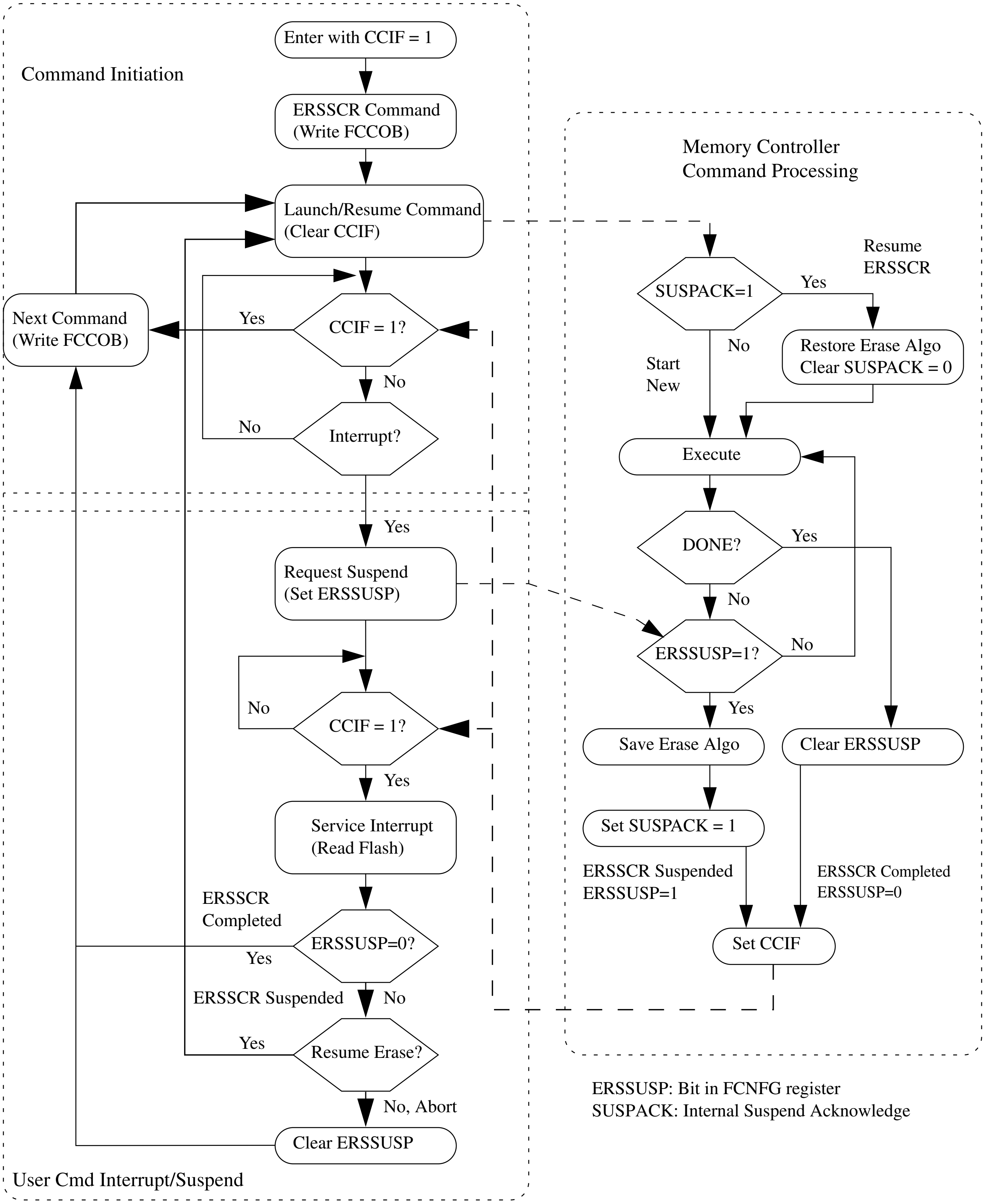
The user may choose to abort a suspended Erase Flash Sector operation by clearing the ERSSUSP bit prior to clearing CCIF for the next command launch. When a suspended operation is aborted, the flash memory module starts the new command using the new FCCOB contents.

While FCNFG[ERSSUSP] is set, a write to the FlexRAM while FCNFG[EEERDY] is set clears ERSSUSP and aborts the suspended operation. The FlexRAM write operation is executed by the flash memory module.

Note

> Aborting the erase leaves the bitcells in an indeterminate, partially-erased state. Data in this sector is not reliable until a new erase command fully completes.

The following figure shows how to suspend and resume the Erase Flash Sector operation.


Command Initiation
Enter with CCIF = 1
ERSSCR Command (Write FCCOB)
Launch/Resume Command (Clear CCIF)
Next Command (Write FCCOB)
Yes
CCIF = 1?
No
No
Interrupt?
Yes
Request Suspend (Set ERSSUSP)
No
CCIF = 1?
Yes
Service Interrupt (Read Flash)
ERSSCR Completed
ERSSUSP=0?
Yes
ERSSCR Suspended
No
Yes
Resume Erase?
No, Abort
Clear ERSSUSP
User Cmd Interrupt/Suspend
Memory Controller Command Processing
Resume ERSSCR
SUSPACK=1
Yes
No
Start New
Restore Erase Algo Clear SUSPACK = 0
Execute
DONE?
Yes
No
ERSSUSP=1?
No
Yes
Save Erase Algo
Clear ERSSUSP
Set SUSPACK = 1
ERSSCR Suspended ERSSUSP=1
ERSSCR Completed ERSSUSP=0
Set CCIF

ERSSUSP: Bit in FCNFG register
SUSPACK: Internal Suspend Acknowledge

Figure 28-33. Suspend and Resume of Erase Flash Sector Operation

28.4.11.8 Program Section Command

The Program Section operation programs the data found in the section program buffer to previously erased locations in the flash memory using an embedded algorithm. Data is preloaded into the section program buffer by writing to the FlexRAM while it is set to function as traditional RAM (see Flash Sector Programming).

The section program buffer is limited to the lower half of the RAM. Data written to the upper half of the RAM is ignored and may be overwritten during Program Section command execution.

CAUTION

> A flash memory location must be in the erased state before being programmed. Cumulative programming of bits (back-to-back program operations without an intervening erase) within a flash memory location is not allowed. Re-programming of existing 0s to 0 is not allowed as this overstresses the device.

Table 28-51. Program Section Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x0B (PGMSEC)
1	Flash address [23:16]
2	Flash address [15:8]
3	Flash address [7:0][1]
4	Number of longwords to program [15:8]
5	Number of longwords to program [7:0]

[1] Must be longword aligned (Flash address [1:0] = 00).

After clearing CCIF to launch the Program Section command, the flash memory module blocks access to the FlexRAM and programs the data residing in the section program buffer into the flash memory starting at the flash address provided.

The starting address must be unprotected (see the description of the FPROT and FDPROT registers) to permit execution of the Program Section operation. Programming, which is not allowed to cross a flash sector boundary, continues until all requested longwords have been programmed. The Program Section command also verifies that after programming, all bits requested to be programmed are programmed.

After the Program Section operation completes, the CCIF flag is set and normal access to the FlexRAM is restored. The contents of the section program buffer may be changed by the Program Section operation.

Table 28-52. Program Section Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid flash address is supplied	FSTAT[ACCERR]
Flash address is not longword aligned	FSTAT[ACCERR]
The requested section crosses a program flash sector boundary	FSTAT[ACCERR]
The requested number of longwords is zero	FSTAT[ACCERR]
The space required to store data for the requested number of longwords is more than half the size of the FlexRAM	FSTAT[ACCERR]
The FlexRAM is not set to function as a traditional RAM, i.e. set if RAMRDY=0	FSTAT[ACCERR]
The flash address falls in a protected area	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

28.4.11.8.1 Flash Sector Programming

The process of programming an entire flash sector using the Program Section command is as follows:

1. If required, execute the Set FlexRAM Function command to make the FlexRAM available as traditional RAM and initialize the FlexRAM to all ones.
2. Launch the Erase Flash Sector command to erase the flash sector to be programmed.
3. Beginning with the starting address of the FlexRAM, sequentially write enough data to the RAM to fill an entire flash sector or half the FlexRAM, whichever is less. This area of the RAM serves as the section program buffer.

   NOTE

   In step 1, the section program buffer was initialized to all ones, the erased state of the flash memory.

   The section program buffer can be written to while the operation launched in step 2 is executing, i.e. while CCIF = 0.
4. Execute the Program Section command to program the contents of the section program buffer into the selected flash sector.
5. If a flash sector is larger than half the FlexRAM, repeat steps 3 and 4 until the sector is completely programmed.
6. To program additional flash sectors, repeat steps 2 through 4.
7. To restore EEPROM functionality, execute the Set FlexRAM Function command to make the FlexRAM available as EEPROM.

28.4.11.9 Read 1s All Blocks Command

The Read 1s All Blocks command checks if the program flash blocks, data flash blocks, EEPROM backup records, and data flash IFR have been erased to the specified read margin level, if applicable, and releases security if the readout passes, i.e. all data reads as '1'.

Table 28-53. Read 1s All Blocks Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x40 (RD1ALL)
1	Read-1 Margin Choice

After clearing CCIF to launch the Read 1s All Blocks command, the flash memory module :

- sets the read margin for 1s according to Table 28-54,
- checks the contents of the program flash, data flash, EEPROM backup records, and data flash IFR are in the erased state.

If the flash memory module confirms that these memory resources are erased, security is released by setting the FSEC[SEC] field to the unsecure state. The security byte in the flash configuration field (see Flash Configuration Field Description) remains unaffected by the Read 1s All Blocks command. If the read fails, i.e. all memory resources are not in the fully erased state, the FSTAT[MGSTAT0] bit is set.

The EEERDY and RAMRDY bits are clear during the Read 1s All Blocks operation and are restored at the end of the Read 1s All Blocks operation.

The CCIF flag sets after the Read 1s All Blocks operation has completed.

Table 28-54. Margin Level Choices for Read 1s All Blocks

Read Margin Choice	Margin Level Description
0x00	Use the 'normal' read level for 1s
0x01	Apply the 'User' margin to the normal read-1 level
0x02	Apply the 'Factory' margin to the normal read-1 level

Table 28-55. Read 1s All Blocks Command Error Handling

Error Condition	Error Bit
An invalid margin choice is specified	FSTAT[ACCERR]
Read-1s fails	FSTAT[MGSTAT0]

28.4.11.10 Read Once Command

The Read Once command provides read access to a reserved 64-byte field located in the program flash IFR (see Program Flash IFR Map and Program Once Field). Access to this field is via 16 records, each 4 bytes long. The Read Once field is programmed using the Program Once command described in Program Once Command.

Table 28-56. Read Once Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x41 (RDONCE)
1	Read Once record index (0x00 - 0x0F)
2	Not used
3	Not used
Returned Values	
4	Read Once byte 0 value
5	Read Once byte 1 value
6	Read Once byte 2 value
7	Read Once byte 3 value

After clearing CCIF to launch the Read Once command, a 4-byte Read Once record is read from the program flash IFR and stored in the FCCOB register. The CCIF flag is set after the Read Once operation completes. Valid record index values for the Read Once command range from 0x00 to 0x0F. During execution of the Read Once command, any attempt to read addresses within the program flash block containing this 64-byte field returns invalid data. The Read Once command can be executed any number of times.

Table 28-57. Read Once Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid record index is supplied	FSTAT[ACCERR]

28.4.11.11 Program Once Command

The Program Once command enables programming to a reserved 64-byte field in the program flash IFR (see Program Flash IFR Map and Program Once Field). Access to the Program Once field is via 16 records, each 4 bytes long. The Program Once field can be read using the Read Once command (see Read Once Command) or using the Read Resource command (see Read Resource Command). Each Program Once record can be programmed only once since the program flash IFR cannot be erased.

Table 28-58. Program Once Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x43 (PGMONCE)
1	Program Once record index (0x00 - 0x0F)
2	Not Used
3	Not Used
4	Program Once Byte 0 value
5	Program Once Byte 1 value
6	Program Once Byte 2 value
7	Program Once Byte 3 value

After clearing CCIF to launch the Program Once command, the flash memory module first verifies that the selected record is erased. If erased, then the selected record is programmed using the values provided. The Program Once command also verifies that the programmed values read back correctly. The CCIF flag is set after the Program Once operation has completed.

The reserved program flash IFR location accessed by the Program Once command cannot be erased and any attempt to program one of these records when the existing value is not Fs (erased) is not allowed. Valid record index values for the Program Once command range from 0x00 to 0x0F. During execution of the Program Once command, any attempt to read addresses within program flash returns invalid data.

Table 28-59. Program Once Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
An invalid record index is supplied	FSTAT[ACCERR]
The requested record has already been programmed to a non-FFFF value[1]	FSTAT[ACCERR]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

[1] If a Program Once record is initially programmed to 0xFFFF_FFFF, the Program Once command is allowed to execute again on that same record.

28.4.11.12 Erase All Blocks Command

The Erase All Blocks operation erases all flash memory, initializes the FlexRAM, verifies all memory contents, and releases MCU security.

Table 28-60. Erase All Blocks Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x44 (ERSALL)

After clearing CCIF to launch the Erase All Blocks command, the flash memory module erases all program flash memory, data flash memory, data flash IFR space, EEPROM backup memory, and FlexRAM, then verifies that all are erased.

If the flash memory module verifies that all flash memories and the FlexRAM were properly erased, security is released by setting the FSEC[SEC] field to the unsecure state and the FCNFG[RAMRDY] bit is set. The Erase All Blocks command aborts if any flash or FlexRAM region is protected. The security byte and all other contents of the flash configuration field (see Flash Configuration Field Description) are erased by the Erase All Blocks command. If the erase-verify fails, the FSTAT[MGSTAT0] bit is set. The CCIF flag is set after the Erase All Blocks operation completes.

Table 28-61. Erase All Blocks Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
Any region of the program flash memory, data flash memory, or FlexRAM is protected	FSTAT[FPVIOL]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

28.4.11.12.1 Triggering an Erase All External to the Flash Memory Module

The functionality of the Erase All Blocks command is also available in an uncommanded fashion outside of the flash memory. Refer to the device's Chip Configuration details for information on this functionality.

Before invoking the external erase all function, the FSTAT[ACCERR and PVIOL] flags must be cleared and the FCCOB0 register must not contain 0x44. When invoked, the erase-all function erases all program flash memory, data flash memory, data flash IFR space, EEPROM backup, and FlexRAM regardless of the protection settings. If the post-erase verify passes, the routine then releases security by setting the FSEC[SEC] field register to the unsecure state and the FCNFG[RAMRDY] bit sets. The security byte in the Flash Configuration Field is also programmed to the unsecure state. The status of the erase-all request is reflected in the FCNFG[ERSAREQ] bit. The FCNFG[ERSAREQ] bit is cleared once the operation completes and the normal FSTAT error reporting is available as described in Erase All Blocks Command.

28.4.11.13 Verify Backdoor Access Key Command

The Verify Backdoor Access Key command only executes if the mode and security conditions are satisfied (see Flash Commands by Mode). Execution of the Verify Backdoor Access Key command is further qualified by the FSEC[KEYEN] bits. The Verify Backdoor Access Key command releases security if user-supplied keys in the

FCCOB match those stored in the Backdoor Comparison Key bytes of the Flash Configuration Field (see Flash Configuration Field Description). The column labelled Flash Configuration Field offset address shows the location of the matching byte in the Flash Configuration Field.

Table 28-62. Verify Backdoor Access Key Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]	Flash Configuration Field Offset Address
0	0x45 (VFYKEY)	
1-3	Not Used	
4	Key Byte 0	0x0_0000
5	Key Byte 1	0x0_0001
6	Key Byte 2	0x0_0002
7	Key Byte 3	0x0_0003
8	Key Byte 4	0x0_0004
9	Key Byte 5	0x0_0005
A	Key Byte 6	0x0_0006
B	Key Byte 7	0x0_0007

After clearing CCIF to launch the Verify Backdoor Access Key command, the flash memory module checks the FSEC[KEYEN] bits to verify that this command is enabled. If not enabled, the flash memory module sets the FSTAT[ACCERR] bit and terminates. If the command is enabled, the flash memory module compares the key provided in FCCOB to the backdoor comparison key in the Flash Configuration Field. If the backdoor keys match, the FSEC[SEC] field is changed to the unsecure state and security is released. If the backdoor keys do not match, security is not released and all future attempts to execute the Verify Backdoor Access Key command are immediately aborted and the FSTAT[ACCERR] bit is (again) set to 1 until a reset of the flash memory module module occurs. If the entire 8-byte key is all zeros or all ones, the Verify Backdoor Access Key command fails with an access error. The CCIF flag is set after the Verify Backdoor Access Key operation completes.

Table 28-63. Verify Backdoor Access Key Command Error Handling

Error Condition	Error Bit
The supplied key is all-0s or all-Fs	FSTAT[ACCERR]
An incorrect backdoor key is supplied	FSTAT[ACCERR]
Backdoor key access has not been enabled (see the description of the FSEC register)	FSTAT[ACCERR]
This command is launched and the backdoor key has mismatched since the last power down reset	FSTAT[ACCERR]

28.4.11.14 Program Partition Command

The Program Partition command prepares the FlexNVM block for use as data flash, EEPROM backup, or a combination of both and initializes the FlexRAM. The Program Partition command must not be launched from flash memory, since flash memory resources are not accessible during Program Partition command execution.

CAUTION

> While different partitions of the FlexNVM are available, the intention is that a single partition choice is used throughout the entire lifetime of a given application. The FlexNVM Partition Code choices affect the endurance and data retention characteristics of the device.

Table 28-64. Program Partition Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x80 (PGMPART)
1	Not Used
2	Not Used
3	Not Used
4	EEPROM Data Size Code[1]
5	FlexNVM Partition Code[2]

1. See Table 28-65 and EEPROM Data Set Size
2. See Table 28-66 and

Table 28-65. Valid EEPROM Data Set Size Codes

EEPROM Data Size Code (FCCOB4)[1]		EEPROM Data Set Size (Bytes)
FCCOB4[5:4]	FCCOB4[EEESIZE]	
11	0xF	0[2]
11	0x9	32
11	0x8	64
11	0x7	128
11	0x6	256
11	0x5	512
11	0x4	1024
11	0x3	2048

1. FCCOB4[7:6] = 00
2. EEPROM Data Set Size must be set to 0 bytes when the FlexNVM Partition Code is set for no EEPROM.

Table 28-66. Valid FlexNVM Partition Codes

FlexNVM Partition Code (FCCOB5[DEPART])[1]	Data flash Size (Kbytes)	EEPROM backup Size (Kbytes)
0000	32	0
0001	24	8
0010	16	16
0011	0	32
1000	0	32
1001	8	24
1010	16	16
1011	32	0

1. FCCOB5[7:4] = 0000

After clearing CCIF to launch the Program Partition command, the flash memory module first verifies that the EEPROM Data Size Code and FlexNVM Partition Code in the data flash IFR are erased. If erased, the Program Partition command erases the contents of the FlexNVM memory. If the FlexNVM is to be partitioned for EEPROM backup, the allocated EEPROM backup sectors are formatted for EEPROM use. Finally, the partition codes are programmed into the data flash IFR using the values provided. The Program Partition command also verifies that the partition codes read back correctly after programming. If the FlexNVM is partitioned for EEPROM, the allocated EEPROM backup sectors are formatted for EEPROM use. The CCIF flag is set after the Program Partition operation completes.

Prior to launching the Program Partition command, the data flash IFR must be in an erased state, which can be accomplished by executing the Erase All Blocks command or by an external request (see Erase All Blocks Command). The EEPROM Data Size Code and FlexNVM Partition Code are read using the Read Resource command (see Read Resource Command).

Table 28-67. Program Partition Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
The EEPROM data size and FlexNVM partition code bytes are not initially 0xFFFF	FSTAT[ACCERR]
Invalid EEPROM Data Size Code is entered (see Table 28-65 for valid codes)	FSTAT[ACCERR]
Invalid FlexNVM Partition Code is entered (see Table 28-66 for valid codes)	FSTAT[ACCERR]
FlexNVM Partition Code = full data flash (no EEPROM) and EEPROM Data Size Code allocates FlexRAM for EEPROM	FSTAT[ACCERR]
FlexNVM Partition Code allocates space for EEPROM backup, but EEPROM Data Size Code allocates no FlexRAM for EEPROM	FSTAT[ACCERR]
FCCOB4[7:6] != 00	FSTAT[ACCERR]

Table continues on the next page...

Table 28-67. Program Partition Command Error Handling (continued)

Error Condition	Error Bit
FCCOB5[7:4] != 0000	FSTAT[ACCERR]
Any errors have been encountered during the verify operation	FSTAT[MGSTAT0]

28.4.11.15 Set FlexRAM Function Command

The Set FlexRAM Function command changes the function of the FlexRAM:

- When not partitioned for EEPROM, the FlexRAM is typically used as traditional RAM.
- When partitioned for EEPROM, the FlexRAM is typically used to store EEPROM data.

Table 28-68. Set FlexRAM Function Command FCCOB Requirements

FCCOB Number	FCCOB Contents [7:0]
0	0x81 (SETRAM)
1	FlexRAM Function Control Code (see Table 28-69)

Table 28-69. FlexRAM Function Control

FlexRAM Function Control Code	Action
0xFF	Make FlexRAM available as RAM: • Clear the FCNFG[EEERDY] and FCNFG[RAMRDY] flags • Write a background of ones to all FlexRAM locations • Set the FCNFG[RAMRDY] flag
0x00	Make FlexRAM available for EEPROM: • Clear the FCNFG[EEERDY] and FCNFG[RAMRDY] flags • Write a background of ones to all FlexRAM locations • Copy-down existing EEPROM data to FlexRAM • Set the FCNFG[EEERDY] flag

After clearing CCIF to launch the Set FlexRAM Function command, the flash memory module sets the function of the FlexRAM based on the FlexRAM Function Control Code.

When making the FlexRAM available as traditional RAM, the flash memory module clears the FCNFG[EEERDY] and FCNFG[RAMRDY] flags, overwrites the contents of the entire FlexRAM with a background pattern of all ones, and sets the FCNFG[RAMRDY] flag. The state of the FEPROT register does not prevent the FlexRAM from being overwritten. When the FlexRAM is set to function as a RAM, normal read and write accesses to the FlexRAM are available. When large sections of flash memory need to be programmed, e.g. during factory programming, the FlexRAM can be used as the Section Program Buffer for the Program Section command (see Program Section Command).

When making the FlexRAM available for EEPROM, the flash memory module clears the FCNFG[EEERDY] and FCNFG[RAMRDY] flags, overwrites the contents of the FlexRAM allocated for EEPROM with a background pattern of all ones, and copies the existing EEPROM data from the EEPROM backup record space to the FlexRAM. After completion of the EEPROM copy-down, the FCNFG[EEERDY] flag is set. When the FlexRAM is set to function as EEPROM, normal read and write access to the FlexRAM is available, but writes to the FlexRAM also invoke EEPROM activity. The CCIF flag is set after the Set FlexRAM Function operation completes.

Table 28-70. Set FlexRAM Function Command Error Handling

Error Condition	Error Bit
Command not available in current mode/security	FSTAT[ACCERR]
FlexRAM Function Control Code is not defined	FSTAT[ACCERR]
FlexRAM Function Control Code is set to make the FlexRAM available for EEPROM, but FlexNVM is not partitioned for EEPROM	FSTAT[ACCERR]

28.4.12 Security

The flash memory module provides security information to the MCU based on contents of the FSEC security register. The MCU then limits access to flash memory resources as defined in the device's Chip Configuration details. During reset, the flash memory module initializes the FSEC register using data read from the security byte of the Flash Configuration Field (see Flash Configuration Field Description).

The following fields are available in the FSEC register. The settings are described in the Flash Security Register (FTFL_FSEC) details.

Table 28-71. FSEC register fields

FSEC field	Description
KEYEN	Backdoor Key Access
MEEN	Mass Erase Capability
FSLACC	Freescale Factory Access
SEC	MCU security

28.4.12.1 Flash Memory Access by Mode and Security

The following table summarizes how access to the flash memory module is affected by security and operating mode.

Table 28-72. Flash Memory Access Summary

Operating Mode	Chip Security State	
	Unsecure	Secure
NVM Normal	Full command set	
NVM Special	Full command set	Only the Erase All Blocks and Read 1s All Blocks commands.

28.4.12.2 Changing the Security State

The security state out of reset can be permanently changed by programming the security byte of the flash configuration field. This assumes that you are starting from a mode where the necessary program flash erase and program commands are available and that the region of the program flash containing the flash configuration field is unprotected. If the flash security byte is successfully programmed, its new value takes affect after the next chip reset.

28.4.12.2.1 Unsecuring the Chip Using Backdoor Key Access

The chip can be unsecured by using the backdoor key access feature, which requires knowledge of the contents of the 8-byte backdoor key value stored in the Flash Configuration Field (see Flash Configuration Field Description). If the FSEC[KEYEN] bits are in the enabled state, the Verify Backdoor Access Key command (see Verify Backdoor Access Key Command) can be run; it allows the user to present prospective keys for comparison to the stored keys. If the keys match, the FSEC[SEC] bits are changed to unsecure the chip. The entire 8-byte key cannot be all 0s or all 1s; that is, 0000_0000_0000_0000h and FFFF_FFFF_FFFF_FFFFh are not accepted by the Verify Backdoor Access Key command as valid comparison values. While the Verify Backdoor Access Key command is active, program flash memory is not available for read access and returns invalid data.

The user code stored in the program flash memory must have a method of receiving the backdoor keys from an external stimulus. This external stimulus would typically be through one of the on-chip serial ports.

If the KEYEN bits are in the enabled state, the chip can be unsecured by the following backdoor key access sequence:

1. Follow the command sequence for the Verify Backdoor Access Key command as explained in Verify Backdoor Access Key Command
2. If the Verify Backdoor Access Key command is successful, the chip is unsecured and the FSEC[SEC] bits are forced to the unsecure state

An illegal key provided to the Verify Backdoor Access Key command prohibits further use of the Verify Backdoor Access Key command. A reset of the chip is the only method to re-enable the Verify Backdoor Access Key command when a comparison fails.

After the backdoor keys have been correctly matched, the chip is unsecured by changing the FSEC[SEC] bits. A successful execution of the Verify Backdoor Access Key command changes the security in the FSEC register only. It does not alter the security byte or the keys stored in the Flash Configuration Field (Flash Configuration Field Description). After the next reset of the chip, the security state of the flash memory module reverts back to the flash security byte in the Flash Configuration Field. The Verify Backdoor Access Key command sequence has no effect on the program and erase protections defined in the program flash protection registers.

If the backdoor keys successfully match, the unsecured chip has full control of the contents of the Flash Configuration Field. The chip may erase the sector containing the Flash Configuration Field and reprogram the flash security byte to the unsecure state and change the backdoor keys to any desired value.

28.4.13 Reset Sequence

On each system reset the flash memory module executes a sequence which establishes initial values for the flash block configuration parameters, FPROT, FDPROT, FEPROT, FOPT, and FSEC registers and the FCNFG[RAMRDY, EEERDY] bits.

CCIF is cleared throughout the reset sequence. The flash memory module holds off all CPU access for a portion of the reset sequence. Flash reads are possible when the hold is removed. Completion of the reset sequence is marked by setting CCIF which enables flash user commands.

If a reset occurs while any flash command is in progress, that command is immediately aborted. The state of the word being programmed or the sector/block being erased is not guaranteed. Commands and operations do not automatically resume after exiting reset.

Chapter 29
EzPort

29.1 Overview

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The EzPort module is a serial flash programming interface that allows In-System Programming (ISP) of flash memory contents on a 32 bit general-purpose microcontroller. Memory contents can be read, erased, and programmed from an external source in a format that is compatible with many stand-alone flash memory chips, without necessitating the removal of the microcontroller from the system.

29.1.1 Introduction

The following figure is a high level block diagram of the EzPort.

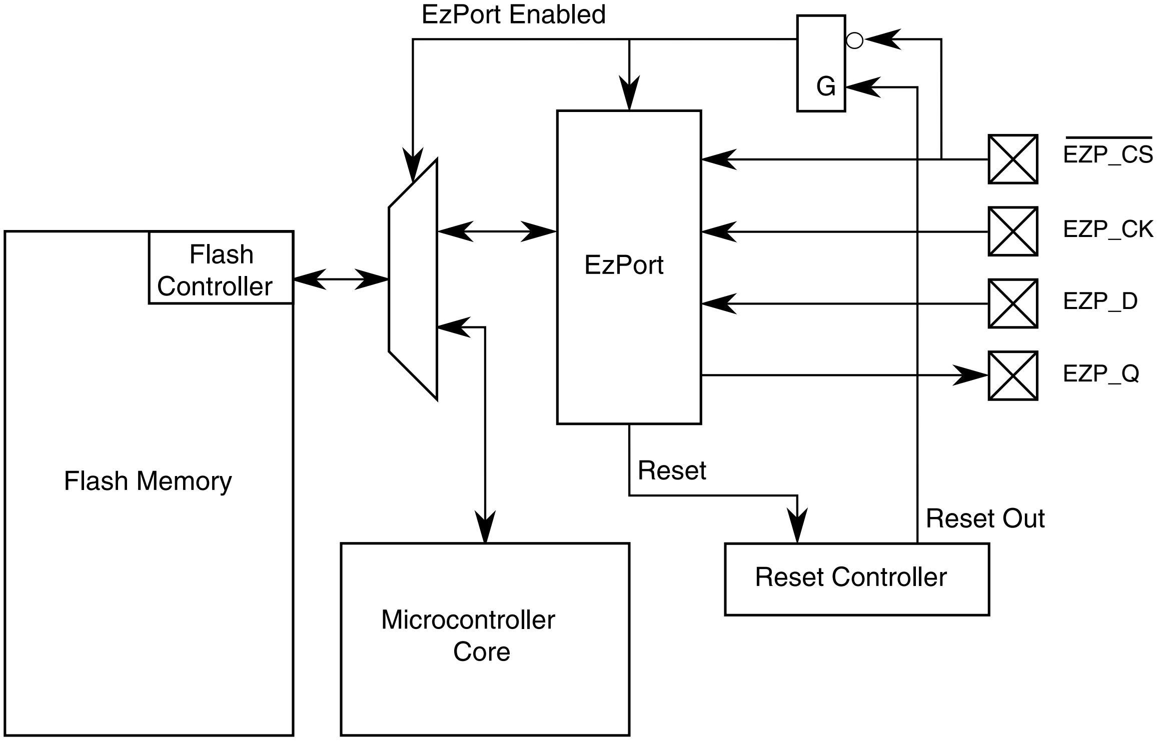

EzPort Enabled
G
$\overline{\text{EZP\_CS}}$
EZP_CK
EZP_D
EZP_Q
Flash Controller
EzPort
Flash Memory
Reset
Reset Out
Reset Controller
Microcontroller Core

Figure 29-1. EzPort block diagram

29.1.2 Features

EzPort includes the following features:

- Serial interface that is compatible with a subset of the SPI format.
- Ability to read, erase, and program flash memory.
- Ability to reset the microcontroller, allowing it to boot from the flash memory after the memory has been configured.

29.1.3 Modes of operation

The EzPort can operate in one of two modes, enabled or disabled.

- Enabled — When enabled, the EzPort steals access to the flash memory, preventing access from other cores or peripherals. The rest of the microcontroller is disabled to avoid conflicts. The flash is configured for NVM Special mode.
- Disabled — When the EzPort is disabled, the rest of the microcontroller can access flash memory as normal.

The EzPort provides a simple interface to connect an external device to the flash memory on board a 32 bit microcontroller. The interface itself is compatible with the SPI interface, with the EzPort operating as a slave, running in either of the two following modes. The data is transmitted with the most significant bit first.

- CPOL = 0, CPHA = 0
- CPOL = 1, CPHA = 1

Commands are issued by the external device to erase, program, or read the contents of the flash memory. The serial data out from the EzPort is tri-stated unless data is being driven. This allows the signal to be shared among several different EzPort (or compatible) devices in parallel, as long as they have different chip-selects.

29.2 External signal description

The following table contains a list of EzPort external signals, and the following sections explain the signals in detail.

Table 29-1. EzPort external signal description

Name	Description	I/O
EZP_CK	EzPort Clock	Input
$\overline{\text{EZP\_CS}}$	EzPort Chip Select	Input
EZP_D	EzPort Serial Data In	Input
EZP_Q	EzPort Serial Data Out	Output

29.2.1 EzPort Clock (EZP_CK)

EZP_CK is the serial clock for data transfers. The serial data in (EZP_D) and chip select ($\overline{\text{EZP\_CS}}$) are registered on the rising edge of EZP_CK, while serial data out (EZP_Q) is driven on the falling edge of EZP_CK.

The maximum frequency of the EzPort clock is half the system clock frequency for all commands except when executing the Read Data or Read FlexRAM commands. When executing these commands, the EzPort clock has a maximum frequency of one-eighth the system clock frequency.

29.2.2 EzPort Chip Select ($\overline{\text{EZP\_CS}}$)

$\overline{\text{EZP\_CS}}$ is the chip select for signaling the start and end of serial transfers. If, while $\overline{\text{EZP\_CS}}$ is asserted, the microcontroller's reset out signal is negated, EzPort is enabled out of reset; otherwise it is disabled. After EzPort is enabled, asserting $\overline{\text{EZP\_CS}}$ commences a serial data transfer, which continues until $\overline{\text{EZP\_CS}}$ is negated again. The negation of $\overline{\text{EZP\_CS}}$ indicates the current command is finished and resets the EzPort state machine so that it is ready to receive the next command.

29.2.3 EzPort Serial Data In (EZP_D)

EZP_D is the serial data in for data transfers. EZP_D is registered on the rising edge of EZP_CK. All commands, addresses, and data are shifted in most significant bit first. When the EzPort is driving output data on EZP_Q, the data shifted in EZP_D is ignored.

29.2.4 EzPort Serial Data Out (EZP_Q)

EZP_Q is the serial data out for data transfers. EZP_Q is driven on the falling edge of EZP_CK. It is tri-stated unless $\overline{\text{EZP\_CS}}$ is asserted and the EzPort is driving data out. All data is shifted out most significant bit first.

29.3 Command definition

The EzPort receives commands from an external device and translates the commands into flash memory accesses. The following table lists the supported commands.

Table 29-2. EzPort commands

Command	Description	Code	Address Bytes	Data Bytes	Accepted when secure?
WREN	Write Enable	0x06	0	0	Yes
WRDI	Write Disable	0x04	0	0	Yes
RDSR	Read Status Register	0x05	0	1	Yes
READ	Flash Read Data	0x03	3[1]	1+	No
FAST_READ	Flash Read Data at High Speed	0x0B	3[1]	1+[2]	No
SP	Flash Section Program	0x02	3[3]	8 - SECTION[4]	No
SE	Flash Sector Erase	0xD8	3[3]	0	No
BE	Flash Bulk Erase	0xC7	0	0	Yes[5]

Table continues on the next page...

Table 29-2. EzPort commands (continued)

Command	Description	Code	Address Bytes	Data Bytes	Accepted when secure?
RESET	Reset Chip	0xB9	0	0	Yes
WRFCCOB	Write FCCOB Registers	0xBA	0	12	Yes[6]
FAST_RDFCCOB	Read FCCOB registers at high speed	0xBB	0	1 - 12[2]	No
WRFLEXRAM	Write FlexRAM	0xBC	3[1]	4	No
RDFLEXRAM	Read FlexRAM	0xBD	3[1]	1+	No
FAST_RDFLEXRAM	Read FlexRAM at high speed	0xBE	3[1]	1+[2]	No

1. Address must be 32-bit aligned (two LSBs must be zero).
2. One byte of dummy data must be shifted in before valid data is shifted out.
3. Address must be 64-bit aligned (three LSBs must be zero).
4. A section is defined as the smaller of either half the size of FlexRAM or the flash sector size. Total number of data bytes programmed must be a multiple of 8.
5. Bulk Erase is accepted when security is set and only when the BEDIS status field is not set.
6. The flash will be in NVM Special mode, restricting the type of commands that can be executed through WRITE_FCCOB when security is enabled.

29.3.1 Command descriptions

This section describes the module commands.

29.3.1.1 Write Enable

The Write Enable (WREN) command sets the write enable register field in the EzPort status register. The write enable field must be set for a write command (SP, SE, BE, WRFCCOB, or WRFLEXRAM) to be accepted. The write enable register field clears on reset, on a Write Disable command, and at the completion of write command. This command must not be used if a write is already in progress.

29.3.1.2 Write Disable

The Write Disable (WRDI) command clears the write enable register field in the status register. This command must not be used if a write is already in progress.

29.3.1.3 Read Status Register

The Read Status Register (RDSR) command returns the contents of the EzPort status register.

Table 29-3. EzPort status register

	7	6	5	4	3	2	1	0
R	FS	WEF			FLEXRAM	BEDIS	WEN	WIP
W								
Reset:	0/1[1]	0	0	0	0/1[2]	0/1[3]	0	1[4]

1. Reset value reflects the status of flash security out of reset.
2. Reset value reflects FlexNVM flash partitioning. If FlexNVM flash has been paritioned for EEPROM, this field is set immediately after reset. Note that FLEXRAM is cleared after the EzPort initialization sequence completes, as indicated by clearing of WIP.
3. Reset value reflects whether bulk erase is enabled or disabled out of reset.
4. Initial value of WIP is 1, but the value clears to 0 after EzPort initialization is complete.

Table 29-4. EzPort status register field description

Field	Description
0 WIP	Write in progress. Sets after a write command (SP, SE, BE, WRFCCOB, or WRFLEXRAM) is accepted and clears after the flash memory has completed all operations associated with the write command, as indicated by the Command Complete Interrupt Flag (CCIF) inside the flash. This field is also asserted on reset and cleared when EzPort initialization is complete. Only the Read Status Register (RDSR) command is accepted while a write is in progress. 0 = Write is not in progress. Accept any command. 1 = Write is in progress. Only accept RDSR command.
1 WEN	Write enable Enables the write comman that follows. It is a control field that must be set before a write command (SP, SE, BE, WRFCCOB, or WRFLEXRAM) is accepted. Is set by the Write Enable (WREN) command and cleared by reset or a Write Disable (WRDI) command. This field also clears when the flash memory has completed all operations associated with the command. 0 = Disables the following write command. 1 = Enables the following write command.
2 BEDIS	Bulk erase disable Indicates whether bulk erase (BE) is disabled when flash is secure. 0 = BE is enabled. 1 = BE is disabled if FS is also set. Attempts to issue a BE command will result in the WEF flag being set.
3 FLEXRAM	FlexRAM mode Indicates the current mode of the FlexRAM. Valid only when WIP is cleared. 0 = FlexRAM is in RAM mode. RD/WRFLEXRAM command can be used to read/write data in FlexRAM. 1 = FlexRAM is in EEPROM mode. SP command is not accepted. RD/WRFLEXRAM command can be used to read/write data in the FlexRAM.

Table continues on the next page...

Table 29-4. EzPort status register field description (continued)

Field	Description
6 WEF	Write error flag Indicates whether there has been an error while executing a write command (SP, SE, BE, WRFCCOB, or WRFLEXRAM). The WEF flag will set if Flash Access Error Flag (ACCERR), Flash Protection Violation (FPVIOL), or Memory Controller Command Completion Status (MGSTAT0) inside the flash memory is set at the completion of the write command. See the flash memory chapter for further description of these flags and their sources. The WEF flag clears after a Read Status Register (RDSR) command. 0 = No error on previous write command. 1 = Error on previous write command.
7 FS	Flash security Indicates whether the flash is secure. See Table 29-2 for the list of commands that will be accepted when flash is secure. Flash security can be disabled by performing a BE command. 0 = Flash is not secure. 1 = Flash is secure.

29.3.1.4 Read Data

The Read Data (READ) command returns data from the flash memory or FlexNVM, depending on the initial address specified in the command word. The initial address must be 32-bit aligned with the two LSBs being zero.

Data continues being returned for as long as the EzPort chip select ($\overline{\text{EZP\_CS}}$) is asserted, with the address automatically incrementing. In this way, the entire contents of flash can be returned by one command. Attempts to read from an address which does not fall within the valid address range for the flash memory regions returns unknown data. See Flash memory map for EzPort access.

For this command to return the correct data, the EzPort clock (EZP_CK) must run at the internal system clock divided by eight or slower. This command is not accepted if the WEF, WIP, or FS field in the EzPort status register is set.

29.3.1.5 Read Data at High Speed

The Read Data at High Speed (FAST_READ) command is identical to the READ command, except for the inclusion of a dummy byte following the address bytes and before the first data byte is returned.

This command can be run with an EzPort clock (EZP_CK) frequency of half the internal system clock frequency of the microcontroller or slower. This command is not accepted if the WEF, WIP, or FS field in the EzPort status register is set.

29.3.1.6 Section Program

The Section Program (SP) command programs up to one section of flash memory that has previously been erased. A section is defined as the smaller of the flash sector size or half the size of the FlexRAM. The starting address of the memory to program is sent after the command word and must be a 64-bit aligned address with the three LSBs being zero).

As data is shifted in, the EzPort buffers the data in FlexRAM before executing an SP command within the flash. For this reason, the number of bytes to be programmed must be a multiple of 8 and up to one flash section can be programmed at a time. For more details, see the Flash Block Guide.

Attempts to program more than one section, across a sector boundary or from an initial address which does not fall within the valid address range for the flash causes the WEF flag to set. See Flash memory map for EzPort access.

This command requires the FlexRAM to be configured for traditional RAM operation. By default, after entering EzPort mode, the FlexRAM is configured for traditional RAM operation. If the user reconfigures FlexRAM for EEPROM operation, then the user should use the WRFCCOB command to configure FlexRAM back to traditional RAM operation before issuing an SP command. See the Flash Memory chapter for details on how the FlexRAM function is modified.

This command is not accepted if the WEF, WIP, FLEXRAM, or FS field is set or if the WEN field is not set in the EzPort status register.

29.3.1.7 Sector Erase

The Sector Erase (SE) command erases the contents of one sector of flash memory. The three byte address sent after the command byte can be any address within the sector to erase, but must be a 64-bit aligned address (the three LSBs must be zero). Attempts to erase from an initial address which does not fall within the valid address range (see Flash memory map for EzPort access) for the flash results in the WEF flag being set.

This command is not accepted if the WEF, WIP or FS field is set or if the WEN field is not set in the EzPort status register.

29.3.1.8 Bulk Erase

The Bulk Erase (BE) command erases the entire contents of flash memory, ignoring any protected sectors or flash security. Flash security is disabled upon successful completion of the BE command.

Attempts to issue a BE command while the BEDIS and FS fields are set results in the WEF flag being set in the EzPort status register. Also, this command is not accepted if the WEF or WIP field is set or if the WEN field is not set in the EzPort status register.

29.3.1.9 EzPort Reset Chip

The Reset Chip (RESET) command forces the chip into the reset state. If the EzPort chip select ($\overline{\text{EZP\_CS}}$) pin is asserted at the end of the reset period, EzPort is enabled; otherwise, it is disabled. This command allows the chip to boot up from flash memory after being programmed by an external source.

This command is not accepted if the WIP field is set in the EzPort status register.

29.3.1.10 Write FCCOB Registers

The Write FCCOB Registers (WRFCCOB) command allows the user to write to the flash common command object registers and execute any command allowed by the flash.

NOTE

> When security is enabled, the flash is configured in NVM Special mode, restricting the commands that can be executed by the flash.

After receiving 12 bytes of data, EzPort writes the data to the FCCOB 0-B registers in the flash and then automatically launches the command within the flash. If greater or less than 12 bytes of data is received, this command has unexpected results and may result in the WEF flag being set.

This command is not accepted if the WEF or WIP field is set or if the WEN field is not set in the EzPort status register.

29.3.1.11 Read FCCOB Registers at High Speed

The Read FCCOB Registers at High Speed (FAST_RDFCCOB) command allows the user to read the contents of the flash common command object registers. After receiving the command, EzPort waits for one dummy byte of data before returning FCCOB register data starting at FCCOB 0 and ending with FCCOB B.

This command can be run with an EzPort clock (EZP_CK) frequency half the internal system clock frequency of the microcontroller or slower. Attempts to read greater than 12 bytes of data returns unknown data. This command is not accepted if the WEF, WIP, or FS fields in the EzPort status register are 1.

29.3.1.12 Write FlexRAM

The Write FlexRAM (WRFLEXRAM) command allows the user to write four bytes of data to the FlexRAM. If the FlexRAM is configured for EEPROM operation, the WRFLEXRAM command can effectively be used to create data records in the EEPROM flash memory.

By default, after entering EzPort mode, the FlexRAM is configured for traditional RAM operation and functions as direct RAM. The user can alter the FlexRAM configuration by using WRFCCOB to execute a Set FlexRAM or Program Partition command within the flash.

The address of the FlexRAM location to be written is sent after the command word and must be a 32-bit aligned address (the two LSBs must be zero). Attempts to write an address which does not fall within the valid address range for the FlexRAM results in the value of the WEF flag being 1. See Flash memory map for EzPort access for more information.

After receiving four bytes of data, EzPort writes the data to the FlexRAM. If greater or less than four bytes of data is received, this command has unexpected results and may result in the value of the WEF flag being 1.

This command is not accepted if the WEF, WIP or FS fields are 1 or if the WEN field is 0 in the EzPort status register.

29.3.1.13 Read FlexRAM

The Read FlexRAM (RDFLEXRAM) command returns data from the FlexRAM. If the FlexRAM is configured for EEPROM operation, the RDFLEXRAM command can effectively be used to read data from EEPROM flash memory.

Data continues being returned for as long as the EzPort chip select ($\overline{\text{EZP\_CS}}$) is asserted, with the address automatically incrementing. In this way, the entire contents of FlexRAM can be returned by one command.

The initial address must be 32-bit aligned (the two LSBs must be zero). Attempts to read from an address which does not fall within the valid address range for the FlexRAM returns unknown data. See Flash memory map for EzPort access for more information.

For this command to return the correct data, the EzPort clock (EZP_CK) must run at the internal system clock divided by eight or slower. This command is not accepted if the WEF, WIP, or FS fields in the EzPort status register are set.

29.3.1.14 Read FlexRAM at High Speed

The Read FlexRAM at High Speed (FAST_RDFLEXRAM) command is identical to the RDFLEXRAM command, except for the inclusion of a dummy byte following the address bytes and before the first data byte is returned.

This command can be run with an EzPort clock (EZP_CK) frequency up to and including half the internal system clock frequency of the microcontroller. This command is not accepted if the WEF, WIP, or FS fields in the EzPort status register are set.

29.4 Flash memory map for EzPort access

The following table shows the flash memory map for access through EzPort.

NOTE

The flash block address map for access through EzPort may not conform to the system memory map. Changes are made to allow the EzPort address width to remain 24 bits.

Table 29-5. Flash Memory Map for EzPort Access

Valid start address	Size	Flash block	Valid commands
0x0000_0000	See device's chip configuration details	Flash	READ, FAST_READ, SP, SE, BE
0x0080_0000	See device's chip configuration details	FlexNVM	READ, FAST_READ, SP, SE, BE
0x0000_0000	See device's chip configuration details	FlexRAM	RDFLEXRAM, FAST_RDFLEXRAM, WRFLEXRAM, BE

Chapter 30
Cyclic Redundancy Check (CRC)

30.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The cyclic redundancy check (CRC) module generates 16/32-bit CRC code for error detection.

The CRC module provides a programmable polynomial, WAS, and other parameters required to implement a 16-bit or 32-bit CRC standard.

The 16/32-bit code is calculated for 32 bits of data at a time.

30.1.1 Features

Features of the CRC module include:

- Hardware CRC generator circuit using a 16-bit or 32-bit programmable shift register
- Programmable initial seed value and polynomial
- Option to transpose input data or output data (the CRC result) bitwise or bytewise. This option is required for certain CRC standards. A bytewise transpose operation is not possible when accessing the CRC data register via 8-bit accesses. In this case, the user's software must perform the bytewise transpose function.
- Option for inversion of final CRC result
- 32-bit CPU register programming interface

30.1.2 Block diagram

The following is a block diagram of the CRC.

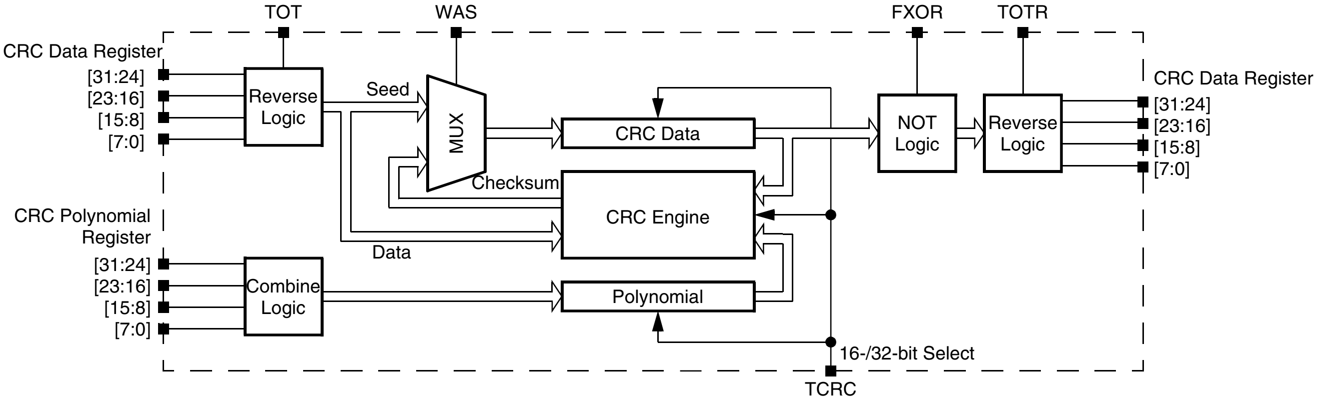

TOT
WAS
FXOR
TOTR
CRC Data Register
[31:24]
[23:16]
[15:8]
[7:0]
Reverse Logic
Seed
MUX
Checksum
CRC Data
CRC Engine
Data
CRC Polynomial Register
[31:24]
[23:16]
[15:8]
[7:0]
Combine Logic
Polynomial
NOT Logic
Reverse Logic
CRC Data Register
[31:24]
[23:16]
[15:8]
[7:0]
16-/32-bit Select
TCRC

Figure 30-1. Programmable cyclic redundancy check (CRC) block diagram

30.1.3 Modes of operation

Various MCU modes affect the CRC module's functionality.

30.1.3.1 Run mode

This is the basic mode of operation.

30.1.3.2 Low-power modes (Wait or Stop)

Any CRC calculation in progress stops when the MCU enters a low-power mode that disables the module clock. It resumes after the clock is enabled or via the system reset for exiting the low-power mode. Clock gating for this module is MCU dependent.

30.2 Memory map and register descriptions

CRC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_2000	CRC Data register (CRC_CRC)	32	R/W	FFFF_ FFFFh	30.2.1/ 573
4003_2004	CRC Polynomial register (CRC_GPOLY)	32	R/W	0000_1021h	30.2.2/ 574
4003_2008	CRC Control register (CRC_CTRL)	32	R/W	0000_0000h	30.2.3/ 575

30.2.1 CRC Data register (CRC_CRC)

The CRC Data register contains the value of the seed, data, and checksum. When CTRL[WAS] is set, any write to the data register is regarded as the seed value. When CTRL[WAS] is cleared, any write to the data register is regarded as data for general CRC computation.

In 16-bit CRC mode, the HU and HL fields are not used for programming the seed value, and reads of these fields return an indeterminate value. In 32-bit CRC mode, all fields are used for programming the seed value.

When programming data values, the values can be written 8 bits, 16 bits, or 32 bits at a time, provided all bytes are contiguous; with MSB of data value written first.

After all data values are written, the CRC result can be read from this data register. In 16-bit CRC mode, the CRC result is available in the LU and LL fields. In 32-bit CRC mode, all fields contain the result. Reads of this register at any time return the intermediate CRC value, provided the CRC module is configured.

Address: CRC_CRC is 4003_2000h base + 0h offset = 4003_2000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HU								HL								LU								LL							
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

CRC_CRC field descriptions

Field	Description
31–24 HU	CRC High Upper Byte In 16-bit CRC mode (CTRL[TCRC] is 0) this field is not used for programming a seed value. In 32-bit CRC mode (CTRL[TCRC] is 1) values written to this field are part of the seed value when CTRL[WAS] is 1. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation in both 16-bit and 32-bit CRC modes.

Table continues on the next page...

CRC_CRC field descriptions (continued)

Field	Description
23–16 HL	CRC High Lower Byte In 16-bit CRC mode (CTRL[TCRC] is 0), this field is not used for programming a seed value. In 32-bit CRC mode (CTRL[TCRC] is 1), values written to this field are part of the seed value when CTRL[WAS] is 1. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation in both 16-bit and 32-bit CRC modes.
15–8 LU	CRC Low Upper Byte When CTRL[WAS] is 1, values written to this field are part of the seed value. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation.
7–0 LL	CRC Low Lower Byte When CTRL[WAS] is 1, values written to this field are part of the seed value. When CTRL[WAS] is 0, data written to this field is used for CRC checksum generation.

30.2.2 CRC Polynomial register (CRC_GPOLY)

This register contains the value of the polynomial for the CRC calculation. The HIGH field contains the upper 16 bits of the CRC polynomial, which are used only in 32-bit CRC mode. Writes to the HIGH field are ignored in 16-bit CRC mode. The LOW field contains the lower 16 bits of the CRC polynomial, which are used in both 16- and 32-bit CRC modes.

Address: CRC_GPOLY is 4003_2000h base + 4h offset = 4003_2004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HIGH																LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1

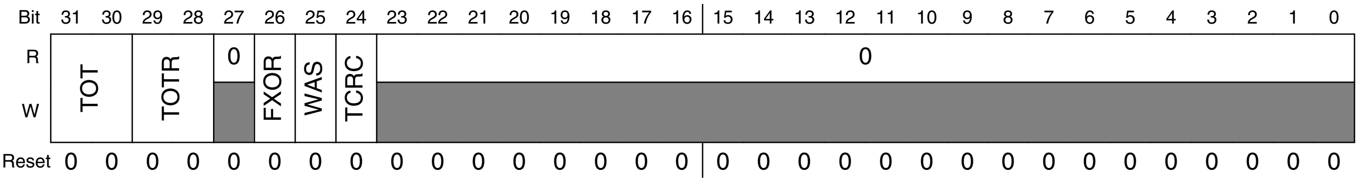
CRC_GPOLY field descriptions

Field	Description
31–16 HIGH	High Polynominal Half-word Writable and readable in 32-bit CRC mode (CTRL[TCRC] is 1). This field is not writable in 16-bit CRC mode (CTRL[TCRC] is 0).
15–0 LOW	Low Polynominal Half-word Writable and readable in both 32-bit and 16-bit CRC modes.

30.2.3 CRC Control register (CRC_CTRL)

This register controls the configuration and working of the CRC module. Appropriate bits must be set before starting a new CRC calculation. A new CRC calculation is initialized by asserting CTRL[WAS] and then writing the seed into the CRC data register.

Address: CRC_CTRL is 4003_2000h base + 8h offset = 4003_2008h



Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TOT		TOTR		0	FXOR	WAS	TCRC	0																							
W					▨				▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRC_CTRL field descriptions

Field	Description
31–30 TOT	Type Of Transpose For Writes Define the transpose configuration of the data written to the CRC data register. See the description of the transpose feature for the available transpose options. 00 No transposition. 01 Bits in bytes are transposed; bytes are not transposed. 10 Both bits in bytes and bytes are transposed. 11 Only bytes are transposed; no bits in a byte are transposed.
29–28 TOTR	Type Of Transpose For Read Identify the transpose configuration of the value read from the CRC Data register. See the description of the transpose feature for the available transpose options. 00 No transposition. 01 Bits in bytes are transposed; bytes are not transposed. 10 Both bits in bytes and bytes are transposed. 11 Only bytes are transposed; no bits in a byte are transposed.
27 Reserved	This read-only field is reserved and always has the value zero.
26 FXOR	Complement Read Of CRC Data Register Some CRC protocols require the final checksum to be XORed with 0xFFFFFFFF or 0xFFFF. Asserting this bit enables on the fly complementing of read data. 0 No XOR on reading. 1 Invert or complement the read value of the CRC Data register.
25 WAS	Write CRC Data Register As Seed When asserted, a value written to the CRC data register is considered a seed value. When deasserted, a value written to the CRC data register is taken as data for CRC computation.

Table continues on the next page...

CRC_CTRL field descriptions (continued)

Field	Description
	0 Writes to the CRC data register are data values. 1 Writes to the CRC data register are seed values.
24 TCRC	Width of CRC protocol. 0 16-bit CRC protocol. 1 32-bit CRC protocol.
23–0 Reserved	This read-only field is reserved and always has the value zero.

30.3 Functional description

30.3.1 CRC initialization/reinitialization

To enable the CRC calculation, the user must program the WAS, POLYNOMIAL, and necessary parameters for transpose and CRC result inversion in the applicable registers. Asserting CTRL[WAS] enables the programming of the seed value into the CRC data register.

After a completed CRC calculation, reasserting CTRL[WAS] and programming a seed, whether the value is new or a previously used seed value, reinitialize the CRC module for a new CRC computation. All other parameters must be set before programming the seed value and subsequent data values.

30.3.2 CRC calculations

In 16-bit and 32-bit CRC modes, data values can be programmed 8 bits, 16 bits, or 32 bits at a time, provided all bytes are contiguous. Noncontiguous bytes can lead to an incorrect CRC computation.

30.3.2.1 16-bit CRC

To compute a 16-bit CRC:

1. Clear CTRL[TCRC] to enable 16-bit CRC mode.
2. Program the transpose and complement options in the CTRL register as required for the CRC calculation. See Transpose feature and CRC result complement for details.

3. Write a 16-bit polynomial to the GPOLY[LOW] field. The GPOLY[HIGH] field is not usable in 16-bit CRC mode.
4. Set CTRL[WAS] to program the seed value.
5. Write a 16-bit seed to CRC[LU:LL]. CRC[HU:HL] are not used.
6. Clear CTRL[WAS] to start writing data values.
7. Write data values into CRC[HU:HL:LU:LL]. A CRC is computed on every data value write, and the intermediate CRC result is stored back into CRC[LU:LL].
8. When all values have been written, read the final CRC result from CRC[LU:LL].

Transpose and complement operations are performed on the fly while reading or writing values. See Transpose feature and CRC result complement for details.

30.3.2.2 32-bit CRC

To compute a 32-bit CRC:

1. Set CTRL[TCRC] to enable 32-bit CRC mode.
2. Program the transpose and complement options in the CTRL register as required for the CRC calculation. See Transpose feature and CRC result complement for details.
3. Write a 32-bit polynomial to GPOLY[HIGH:LOW].
4. Set CTRL[WAS] to program the seed value.
5. Write a 32-bit seed to CRC[HU:HL:LU:LL].
6. Clear CTRL[WAS] to start writing data values.
7. Write data values into CRC[HU:HL:LU:LL]. A CRC is computed on every data value write, and the intermediate CRC result is stored back into CRC[HU:HL:LU:LL].
8. When all values have been written, read the final CRC result from CRC[HU:HL:LU:LL]. The CRC is calculated bytewise, and two clocks are required to complete one CRC calculation.

Transpose and complement operations are performed on the fly while reading or writing values. See Transpose feature and CRC result complement for details.

30.3.3 Transpose feature

By default, the transpose feature is not enabled. However, some CRC standards require the input data and/or the final checksum to be transposed. The user software has the option to configure each transpose operation separately, as desired by the CRC standard. The data is transposed on the fly while being read or written.

Some protocols use little endian format for the data stream to calculate a CRC. In this case, the transpose feature usefully flips the bits. This transpose option is one of the types supported by the CRC module.

30.3.3.1 Types of transpose

The CRC module provides several types of transpose functions to flip the bits and/or bytes, for both writing input data and reading the CRC result, separately using the CTRL[TOT] or CTRL[TOTR] fields, according to the CRC calculation being used.

The following types of transpose functions are available for writing to and reading from the CRC data register:

1. CTRL[TOT] or CTRL[TOTR] is 00

   No transposition occurs.

2. CTRL[TOT] or CTRL[TOTR] is 01

   Bits in a byte are transposed, while bytes are not transposed.

   reg[31:0] becomes {reg[24:31], reg[16:23], reg[8:15], reg[0:7]}

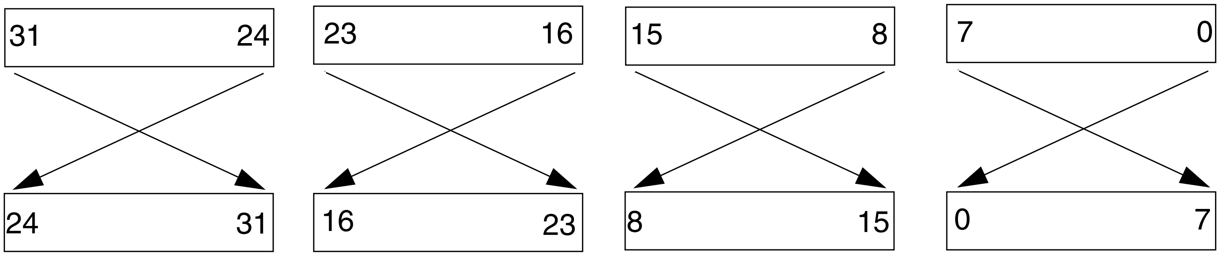


Figure 30-5. Transpose type 01

3. CTRL[TOT] or CTRL[TOTR] is 10

   Both bits in bytes and bytes are transposed.

   reg[31:0] becomes = {reg[0:7], reg[8:15],reg[16:23], reg[24:31]}

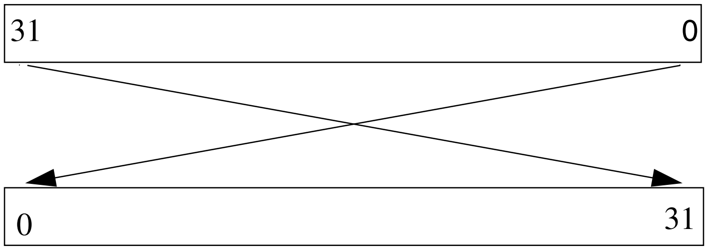


Figure 30-6. Transpose type 10

4. CTRL[TOT] or CTRL[TOTR] is 11

Bytes are transposed, but bits are not transposed.

reg[31:0] becomes {reg[7:0], reg[15:8], reg[23:16], reg[31:24]}

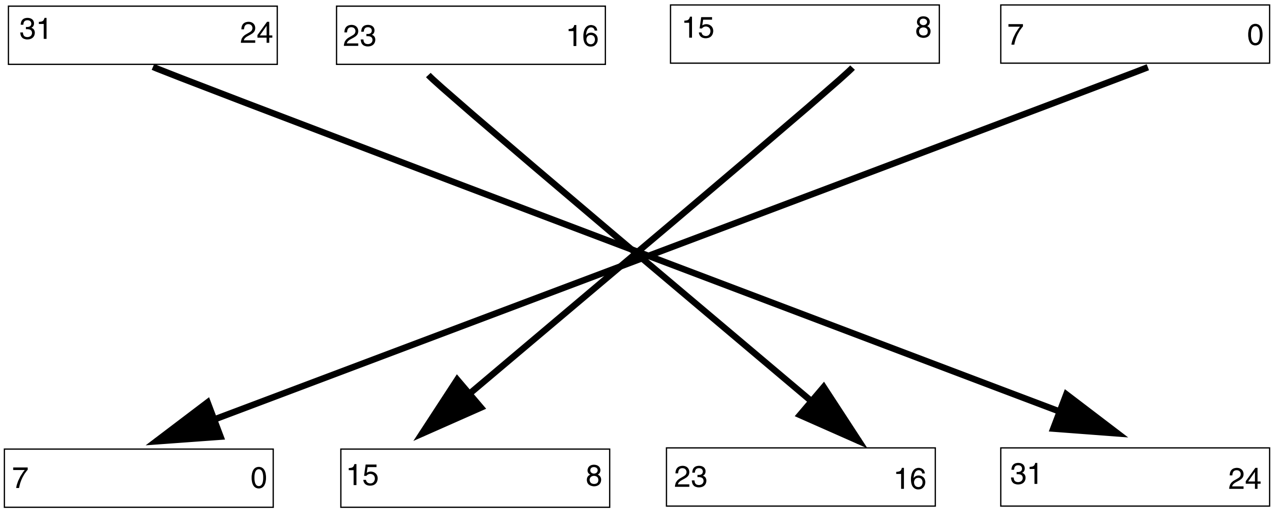


Figure 30-7. Transpose type 11

NOTE

For 8-bit and 16-bit write accesses to the CRC data register, the data is transposed with zeros on the unused byte or bytes (taking 32 bits as a whole), but the CRC is calculated on the valid byte(s) only. When reading the CRC data register for a 16-bit CRC result and using transpose options 10 and 11, the resulting value after transposition resides in the CRC[HU:HL] fields. The user software must account for this situation when reading the 16-bit CRC result, so reading 32 bits is preferred.

30.3.4 CRC result complement

When CTRL[FXOR] is set, the checksum is complemented. The CRC result complement function outputs the complement of the checksum value stored in the CRC data register every time the CRC data register is read. When CTRL[FXOR] is cleared, reading the CRC data register accesses the raw checksum value.

Chapter 31
Analog-to-Digital Converter (ADC)

31.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The 16-bit analog-to-digital converter (ADC) is a successive approximation ADC designed for operation within an integrated microcontroller system-on-chip.

NOTE

For the chip specific modes of operation, refer to the Power Management information for the device.

31.1.1 Features

Features of the ADC module include:

- Linear successive approximation algorithm with up to 16-bit resolution
- Up to 4 pairs of differential and 24 single-ended external analog inputs
- Output modes: differential 16-bit, 13-bit, 11-bit and 9-bit modes, or single-ended 16-bit, 12-bit, 10-bit and 8-bit modes
- Output formatted in 2's complement 16-bit sign extended for differential modes
- Output in right-justified unsigned format for single-ended
- Single or continuous conversion (automatic return to idle after single conversion)
- Configurable sample time and conversion speed/power
- Conversion complete / hardware average complete flag and interrupt

- Input clock selectable from up to four sources
- Operation in low power modes for lower noise operation
- Asynchronous clock source for lower noise operation with option to output the clock
- Selectable hardware conversion trigger with hardware channel select
- Automatic compare with interrupt for less-than, greater-than or equal-to, within range, or out-of-range, programmable value
- Temperature sensor
- Hardware average function
- Selectable voltage reference: external or alternate
- Self-calibration mode

31.1.2 Block diagram

The following figure is the ADC module block diagram.

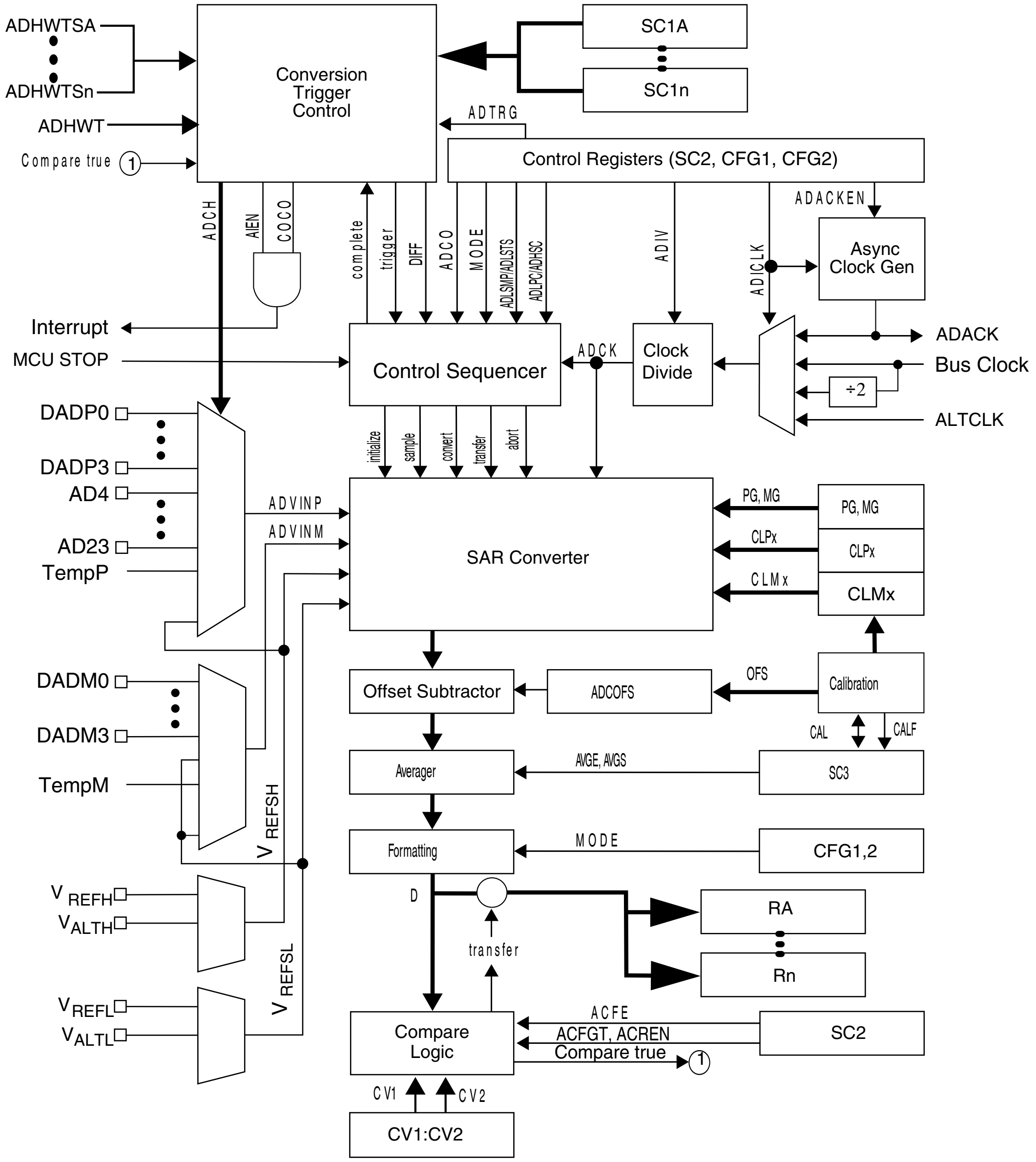

ADHWTSA
ADHWTSn
ADHWT
Compare true
1
Conversion Trigger Control
SC1A
SC1n
ADTRG
Control Registers (SC2, CFG1, CFG2)
ADACKEN
Async Clock Gen
ADCH
AIEN
COCO
complete
trigger
DIFF
ADCO
MODE
ADLSMP/ADLSTS
ADLPC/ADHSC
ADIV
ADICLK
Interrupt
MCU STOP
Control Sequencer
ADCK
Clock Divide
ADACK
Bus Clock
÷2
ALTCLK
DADP0
DADP3
AD4
AD23
TempP
initialize
sample
convert
transfer
abort
ADVINP
ADVINM
SAR Converter
PG, MG
CLPx
CLMx
DADM0
DADM3
TempM
Offset Subtractor
ADCOFS
OFS
Calibration
CAL
CALF
Averager
AVGE, AVGS
SC3
V REFSH
Formatting
MODE
CFG1,2
V REFH
V ALTH
D
transfer
RA
Rn
V REFL
V ALTL
V REFSL
Compare Logic
ACFE
ACFGT, ACREN
Compare true
SC2
CV1
CV2
CV1:CV2

Figure 31-1. ADC block diagram

31.2 ADC Signal Descriptions

The ADC module supports up to 4 pairs of differential inputs and up to 24 single-ended inputs. Each differential pair requires two inputs, DADPx and DADMx. The ADC also requires four supply/reference/ground connections.

Table 31-1. ADC Signal Descriptions

Signal	Description	I/O
DADP[3:0]	Differential analog channel inputs	I

Table continues on the next page...

Table 31-1. ADC Signal Descriptions (continued)

Signal	Description	I/O
DADM[3:0]	Differential analog channel inputs	I
AD[23:4]	Single-ended analog channel inputs	I
V_{REFSH}	Voltage reference select high	I
V_{REFSL}	Voltage reference select low	I
V_{DDA}	Analog power supply	I
V_{SSA}	Analog ground	I

31.2.1 Analog power (V_{DDA})

The ADC analog portion uses V_{DDA} as its power connection. In some packages, V_{DDA} is connected internally to V_{DD}. If externally available, connect the V_{DDA} pin to the same voltage potential as V_{DD}. External filtering may be necessary to ensure clean V_{DDA} for good results.

31.2.2 Analog ground (V_{SSA})

The ADC analog portion uses V_{SSA} as its ground connection. In some packages, V_{SSA} is connected internally to V_{SS}. If externally available, connect the V_{SSA} pin to the same voltage potential as V_{SS}.

31.2.3 Voltage reference select

V_{REFSH} and V_{REFSL} are the high and low reference voltages for the converter.

The ADC can be configured to accept one of two voltage reference pairs for V_{REFSH} and V_{REFSL}. Each pair contains a positive reference that must be between the minimum Ref Voltage High and V_{DDA}, and a ground reference that must be at the same potential as V_{SSA}. The two pairs are external (V_{REFH} and V_{REFL}) and alternate (V_{ALTH} and V_{ALTL}). These voltage references are selected using the REFSEL bits in the SC2 register. The alternate (V_{ALTH} and V_{ALTL}) voltage reference pair may select additional external pins or internal sources depending on MCU configuration. Refer to the Chip Configuration information on the Voltage References specific to this MCU.

In some packages, V_{REFH} is connected in the package to V_{DDA} and V_{REFL} to V_{SSA}. If externally available, the positive reference(s) may be connected to the same potential as V_{DDA} or may be driven by an external source to a level between the minimum Ref Voltage High and the V_{DDA} potential (V_{REFH} must never exceed V_{DDA}). Connect the ground references to the same voltage potential as V_{SSA}.

31.2.4 Analog channel inputs (ADx)

The ADC module supports up to 24 single-ended analog inputs. A single-ended input is selected for conversion through the ADCH channel select bits when the DIFF bit in the SC1n register is low.

31.2.5 Differential analog channel inputs (DADx)

The ADC module supports up to 4 differential analog channel inputs. Each differential analog input is a pair of external pins (DADPx and DADMx) referenced to each other to provide the most accurate analog to digital readings. A differential input is selected for conversion through the ADCH channel select bits when the DIFF bit in the SC1n register bit is high. All DADPx inputs may be used as single-ended inputs if the DIFF bit is low. In certain MCU configurations, some DADMx inputs may also be used as single-ended inputs if the DIFF bit is low. Refer to the Chip Configuration chapter for ADC connections specific to this MCU.

31.3 Register Definition

This section describes the ADC registers.

ADC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_B000	ADC status and control registers 1 (ADC0_SC1A)	32	R/W	0000_001Fh	31.3.1/ 587
4003_B004	ADC status and control registers 1 (ADC0_SC1B)	32	R/W	0000_001Fh	31.3.1/ 587
4003_B008	ADC configuration register 1 (ADC0_CFG1)	32	R/W	0000_0000h	31.3.2/ 590

Table continues on the next page...

ADC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_B00C	Configuration register 2 (ADC0_CFG2)	32	R/W	0000_0000h	31.3.3/ 592
4003_B010	ADC data result register (ADC0_RA)	32	R	0000_0000h	31.3.4/ 593
4003_B014	ADC data result register (ADC0_RB)	32	R	0000_0000h	31.3.4/ 593
4003_B018	Compare value registers (ADC0_CV1)	32	R/W	0000_0000h	31.3.5/ 594
4003_B01C	Compare value registers (ADC0_CV2)	32	R/W	0000_0000h	31.3.5/ 594
4003_B020	Status and control register 2 (ADC0_SC2)	32	R/W	0000_0000h	31.3.6/ 595
4003_B024	Status and control register 3 (ADC0_SC3)	32	R/W	0000_0000h	31.3.7/ 597
4003_B028	ADC offset correction register (ADC0_OFS)	32	R/W	0000_0004h	31.3.8/ 598
4003_B02C	ADC plus-side gain register (ADC0_PG)	32	R/W	0000_8200h	31.3.9/ 599
4003_B030	ADC minus-side gain register (ADC0_MG)	32	R/W	0000_8200h	31.3.10/ 599
4003_B034	ADC plus-side general calibration value register (ADC0_CLPD)	32	R/W	0000_000Ah	31.3.11/ 600
4003_B038	ADC plus-side general calibration value register (ADC0_CLPS)	32	R/W	0000_0020h	31.3.12/ 601
4003_B03C	ADC plus-side general calibration value register (ADC0_CLP4)	32	R/W	0000_0200h	31.3.13/ 601
4003_B040	ADC plus-side general calibration value register (ADC0_CLP3)	32	R/W	0000_0100h	31.3.14/ 602
4003_B044	ADC plus-side general calibration value register (ADC0_CLP2)	32	R/W	0000_0080h	31.3.15/ 602
4003_B048	ADC plus-side general calibration value register (ADC0_CLP1)	32	R/W	0000_0040h	31.3.16/ 603
4003_B04C	ADC plus-side general calibration value register (ADC0_CLP0)	32	R/W	0000_0020h	31.3.17/ 603
4003_B054	ADC minus-side general calibration value register (ADC0_CLMD)	32	R/W	0000_000Ah	31.3.18/ 604
4003_B058	ADC minus-side general calibration value register (ADC0_CLMS)	32	R/W	0000_0020h	31.3.19/ 604
4003_B05C	ADC minus-side general calibration value register (ADC0_CLM4)	32	R/W	0000_0200h	31.3.20/ 605

Table continues on the next page...

ADC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_B060	ADC minus-side general calibration value register (ADC0_CLM3)	32	R/W	0000_0100h	31.3.21/ 605
4003_B064	ADC minus-side general calibration value register (ADC0_CLM2)	32	R/W	0000_0080h	31.3.22/ 606
4003_B068	ADC minus-side general calibration value register (ADC0_CLM1)	32	R/W	0000_0040h	31.3.23/ 606
4003_B06C	ADC minus-side general calibration value register (ADC0_CLM0)	32	R/W	0000_0020h	31.3.24/ 607

31.3.1 ADC status and control registers 1 (ADCx_SC1*n*)

The SC1A register is used for both software and hardware trigger modes of operation.

To allow sequential conversions of the ADC to be triggered by internal peripherals, the ADC can have more then one status and control register: one for each conversion. The SC1B-SC1n registers indicate potentially multiple SC1 registers for use only in hardware trigger mode. Refer to the Chip Configuration information about the number of SC1n registers specific to this device. The SC1n registers have identical fields, and are used in a "ping-pong" approach to control ADC operation.

At any one point in time, only one of the SC1n registers is actively controlling ADC conversions. Updating SC1A while SC1n is actively controlling a conversion is allowed (and vice-versa for any of the SC1n registers specific to this MCU).

Writing SC1A while SC1A is actively controlling a conversion aborts the current conversion. In software trigger mode (ADTRG=0), writes to the SC1A register subsequently initiate a new conversion (if the ADCH bits are equal to a value other than all 1s).

Similarly, writing any of the SC1n registers while that specific SC1n register is actively controlling a conversion aborts the current conversion. None of the SC1B-SC1n registers are used for software trigger operation and therefore writes to the SC1B - SC1n registers do not initiate a new conversion.

Addresses: ADC0_SC1A is 4003_B000h base + 0h offset = 4003_B000h
ADC0_SC1B is 4003_B000h base + 4h offset = 4003_B004h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	COCO	AIEN	DIFF	ADCH				
W								
Reset	0	0	0	1	1	1	1	1

ADCx_SC1n field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 COCO	Conversion complete flag The COCO flag is a read-only bit that is set each time a conversion is completed when the compare function is disabled (ACFE=0) and the hardware average function is disabled (AVGE=0). When the compare function is enabled (ACFE=1), the COCO flag is set upon completion of a conversion only if the compare result is true. When the hardware average function is enabled (AVGE=1), the COCO flag is set upon completion of the selected number of conversions (determined by the AVGS bits). The COCO flag in SC1A is also set at the completion of a Calibration sequence. The COCO bit is cleared when the respective SC1n register is written or when the respective Rn register is read. 0 Conversion not completed. 1 Conversion completed.
6 AIEN	Interrupt enable AIEN enables conversion complete interrupts. When COCO becomes set while the respective AIEN is high, an interrupt is asserted. 0 Conversion complete interrupt disabled. 1 Conversion complete interrupt enabled.
5 DIFF	Differential mode enable DIFF configures the ADC to operate in differential mode. When enabled, this mode automatically selects from the differential channels, and changes the conversion algorithm and the number of cycles to complete a conversion.

Table continues on the next page...

ADCx_SC1*n* field descriptions (continued)

Field	Description
	0 Single-ended conversions and input channels are selected. 1 Differential conversions and input channels are selected.
4–0 ADCH	Input channel select The ADCH bits form a 5-bit field that selects one of the input channels. The input channel decode depends on the value of the DIFF bit. DAD0-DAD3 are associated with the input pin pairs DADPx and DADMx. The successive approximation converter subsystem is turned off when the channel select bits are all set (ADCH = 11111). This feature allows for explicit disabling of the ADC and isolation of the input channel from all sources. Terminating continuous conversions this way prevents an additional single conversion from being performed. It is not necessary to set the channel select bits to all ones to place the ADC in a low-power state when continuous conversions are not enabled because the module automatically enters a low-power state when a conversion completes. 00000 When DIFF=0, DADP0 is selected as input; when DIFF=1, DAD0 is selected as input. 00001 When DIFF=0, DADP1 is selected as input; when DIFF=1, DAD1 is selected as input. 00010 When DIFF=0, DADP2 is selected as input; when DIFF=1, DAD2 is selected as input. 00011 When DIFF=0, DADP3 is selected as input; when DIFF=1, DAD3 is selected as input. 00100 When DIFF=0, AD4 is selected as input; when DIFF=1, it is reserved. 00101 When DIFF=0, AD5 is selected as input; when DIFF=1, it is reserved. 00110 When DIFF=0, AD6 is selected as input; when DIFF=1, it is reserved. 00111 When DIFF=0, AD7 is selected as input; when DIFF=1, it is reserved. 01000 When DIFF=0, AD8 is selected as input; when DIFF=1, it is reserved. 01001 When DIFF=0, AD9 is selected as input; when DIFF=1, it is reserved. 01010 When DIFF=0, AD10 is selected as input; when DIFF=1, it is reserved. 01011 When DIFF=0, AD11 is selected as input; when DIFF=1, it is reserved. 01100 When DIFF=0, AD12 is selected as input; when DIFF=1, it is reserved. 01101 When DIFF=0, AD13 is selected as input; when DIFF=1, it is reserved. 01110 When DIFF=0, AD14 is selected as input; when DIFF=1, it is reserved. 01111 When DIFF=0, AD15 is selected as input; when DIFF=1, it is reserved. 10000 When DIFF=0, AD16 is selected as input; when DIFF=1, it is reserved. 10001 When DIFF=0, AD17 is selected as input; when DIFF=1, it is reserved. 10010 When DIFF=0, AD18 is selected as input; when DIFF=1, it is reserved. 10011 When DIFF=0, AD19 is selected as input; when DIFF=1, it is reserved. 10100 When DIFF=0, AD20 is selected as input; when DIFF=1, it is reserved. 10101 When DIFF=0, AD21 is selected as input; when DIFF=1, it is reserved. 10110 When DIFF=0, AD22 is selected as input; when DIFF=1, it is reserved. 10111 When DIFF=0, AD23 is selected as input; when DIFF=1, it is reserved. 11000 Reserved. 11001 Reserved. 11010 When DIFF=0, Temp sensor (single-ended) is selected as input; when DIFF=1, Temp sensor (differential) is selected as input. 11011 When DIFF=0,Bandgap (single-ended) is selected as input; when DIFF=1, Bandgap (differential) is selected as input. 11100 Reserved. 11101 When DIFF=0, V_{REFSH} is selected as input; when DIFF=1, $-V_{REFSH}$ (differential) is selected as input. Voltage reference selected is determined by the REFSEL bits in the SC2 register. 11110 When DIFF=0, V_{REFSL} is selected as input; when DIFF=1, it is reserved. Voltage reference selected is determined by the REFSEL bits in the SC2 register. 11111 Module disabled.

31.3.2 ADC configuration register 1 (ADCx_CFG1)

CFG1 register selects the mode of operation, clock source, clock divide, and configure for low power or long sample time.

Addresses: ADC0_CFG1 is 4003_B000h base + 8h offset = 4003_B008h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R W	ADLPC	ADIV		ADLSMP	MODE		ADICLK	
Reset	0	0	0	0	0	0	0	0

ADCx_CFG1 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 ADLPC	Low-power configuration ADLPC controls the power configuration of the successive approximation converter. This optimizes power consumption when higher sample rates are not required. 0 Normal power configuration. 1 Low power configuration. The power is reduced at the expense of maximum clock speed.
6–5 ADIV	Clock divide select ADIV selects the divide ratio used by the ADC to generate the internal clock ADCK. 00 The divide ratio is 1 and the clock rate is input clock. 01 The divide ratio is 2 and the clock rate is (input clock)/2. 10 The divide ratio is 4 and the clock rate is (input clock)/4. 11 The divide ratio is 8 and the clock rate is (input clock)/8.
4 ADLSMP	Sample time configuration ADLSMP selects between different sample times based on the conversion mode selected. This bit adjusts the sample period to allow higher impedance inputs to be accurately sampled or to maximize conversion

Table continues on the next page...

ADCx_CFG1 field descriptions (continued)

Field	Description
	speed for lower impedance inputs. Longer sample times can also be used to lower overall power consumption if continuous conversions are enabled and high conversion rates are not required. When ADLSMP=1, the long sample time select bits, (ADLSTS[1:0]), can select the extent of the long sample time. 0 Short sample time. 1 Long sample time.
3–2 MODE	Conversion mode selection MODE bits are used to select the ADC resolution mode. 00 When DIFF=0: It is single-ended 8-bit conversion; when DIFF=1, it is differential 9-bit conversion with 2's complement output. 01 When DIFF=0: It is single-ended 12-bit conversion; when DIFF=1, it is differential 13-bit conversion with 2's complement output. 10 When DIFF=0: It is single-ended 10-bit conversion; when DIFF=1, it is differential 11-bit conversion with 2's complement output. 11 When DIFF=0: It is single-ended 16-bit conversion; when DIFF=1, it is differential 16-bit conversion with 2's complement output.
1–0 ADICLK	Input clock select ADICLK bits select the input clock source to generate the internal clock, ADCK. Note that when the ADACK clock source is selected, it is not required to be active prior to conversion start. When it is selected and it is not active prior to a conversion start (ADACKEN=0), the asynchronous clock is activated at the start of a conversion and shuts off when conversions are terminated. In this case, there is an associated clock startup delay each time the clock source is re-activated. 00 Bus clock. 01 Bus clock divided by 2. 10 Alternate clock (ALTCLK). 11 Asynchronous clock (ADACK).

31.3.3 Configuration register 2 (ADCx_CFG2)

CFG2 register selects the special high speed configuration for very high speed conversions and selects the long sample time duration during long sample mode.

Addresses: ADC0_CFG2 is 4003_B000h base + Ch offset = 4003_B00Ch

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	0			MUXSEL	ADACKEN	ADHSC	ADLSTS	
W								
Reset	0	0	0	0	0	0	0	0

ADCx_CFG2 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7–5 Reserved	This read-only field is reserved and always has the value zero.
4 MUXSEL	ADC Mux select ADC Mux select bit is used to change the ADC mux setting to select between alternate sets of ADC channels. 0 ADxxa channels are selected. 1 ADxxb channels are selected.
3 ADACKEN	Asynchronous clock output enable ADACKEN enables the ADC's asynchronous clock source and the clock source output regardless of the conversion and input clock select (ADICLK bits) status of the ADC. Based on MCU configuration, the asynchronous clock may be used by other modules (see Chip Configuration information). Setting this bit allows the clock to be used even while the ADC is idle or operating from a different clock source. Also, latency of initiating a single or first-continuous conversion with the asynchronous clock selected is reduced since the ADACK clock is already operational.

Table continues on the next page...

ADCx_CFG2 field descriptions (continued)

Field	Description
	0 Asynchronous clock output disabled; Asynchronous clock only enabled if selected by ADICLK and a conversion is active. 1 Asynchronous clock and clock output enabled regardless of the state of the ADC.
2 ADHSC	High speed configuration ADHSC configures the ADC for very high speed operation. The conversion sequence is altered (2 ADCK cycles added to the conversion time) to allow higher speed conversion clocks. 0 Normal conversion sequence selected. 1 High speed conversion sequence selected (2 additional ADCK cycles to total conversion time).
1–0 ADLSTS	Long sample time select ADLSTS selects between the extended sample times when long sample time is selected (ADLSMP=1). This allows higher impedance inputs to be accurately sampled or to maximize conversion speed for lower impedance inputs. Longer sample times can also be used to lower overall power consumption when continuous conversions are enabled if high conversion rates are not required. 00 Default longest sample time (20 extra ADCK cycles; 24 ADCK cycles total). 01 12 extra ADCK cycles; 16 ADCK cycles total sample time. 10 6 extra ADCK cycles; 10 ADCK cycles total sample time. 11 2 extra ADCK cycles; 6 ADCK cycles total sample time.

31.3.4 ADC data result register (ADCx_R*n*)

The data result registers (Rn) contain the result of an ADC conversion of the channel selected by the corresponding status and channel control register (SC1A:SC1n). For every status and channel control register, there is a corresponding data result register.

Unused bits in the Rn register are cleared in unsigned right justified modes and carry the sign bit (MSB) in sign extended 2's complement modes. For example, when configured for 10-bit single-ended mode, D[15:10] are cleared. When configured for 11-bit differential mode, D[15:10] carry the sign bit (bit 10 extended through bit 15).

The following table describes the behavior of the data result registers in the different modes of operation.

Table 31-43. Data result register description

Conversion mode	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Format
16-bit differential	S	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Signed 2's complement
16-bit single-ended	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right justified

Table continues on the next page...

Table 31-43. Data result register description (continued)

Conversion mode	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	Format
13-bit differential	S	S	S	S	D	D	D	D	D	D	D	D	D	D	D	D	Sign extended 2's complement
12-bit single-ended	0	0	0	0	D	D	D	D	D	D	D	D	D	D	D	D	Unsigned right justified
11-bit differential	S	S	S	S	S	S	D	D	D	D	D	D	D	D	D	D	Sign extended 2's complement
10-bit single-ended	0	0	0	0	0	0	D	D	D	D	D	D	D	D	D	D	Unsigned right justified
9-bit differential	S	S	S	S	S	S	S	S	D	D	D	D	D	D	D	D	Sign extended 2's complement
8-bit single-ended	0	0	0	0	0	0	0	0	D	D	D	D	D	D	D	D	Unsigned right justified

NOTE

S: Sign bit or sign bit extension;

D: Data (2's complement data if indicated)

Addresses: ADC0_RA is 4003_B000h base + 10h offset = 4003_B010h

ADC0_RB is 4003_B000h base + 14h offset = 4003_B014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																D															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADC*x*_R*n* field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 D	Data result

31.3.5 Compare value registers (ADC*x*_CV*n*)

The compare value registers (CV1 and CV2) contain a compare value used to compare with the conversion result when the compare function is enabled (ACFE=1). This register is formatted the same for both bit position definition and value format (unsigned or sign-extended 2's complement) as the data result registers (Rn) in the different modes of operation. Therefore, the compare function only uses the compare value register bits that are related to the ADC mode of operation.

The compare value 2 register (CV2) is utilized only when the compare range function is enabled (ACREN=1).

Addresses: ADC0_CV1 is 4003_B000h base + 18h offset = 4003_B018h
ADC0_CV2 is 4003_B000h base + 1Ch offset = 4003_B01Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																CV															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_CVn field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 CV	Compare value

31.3.6 Status and control register 2 (ADCx_SC2)

The SC2 register contains the conversion active, hardware/software trigger select, compare function and voltage reference select of the ADC module.

Addresses: ADC0_SC2 is 4003_B000h base + 20h offset = 4003_B020h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	ADACT	ADTRG	ACFE	ACFGT	ACREN	DMAEN	REFSEL	
W								
Reset	0	0	0	0	0	0	0	0

ADCx_SC2 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 ADACT	Conversion active ADACT indicates that a conversion or hardware averaging is in progress. ADACT is set when a conversion is initiated and cleared when a conversion is completed or aborted. 0 Conversion not in progress. 1 Conversion in progress.
6 ADTRG	Conversion trigger select ADTRG selects the type of trigger used for initiating a conversion. Two types of trigger are selectable: software trigger and hardware trigger. When software trigger is selected, a conversion is initiated following a write to SC1A. When hardware trigger is selected, a conversion is initiated following the assertion of the ADHWT input after a pulse of the ADHWTSn input. 0 Software trigger selected. 1 Hardware trigger selected.
5 ACFE	Compare function enable ACFE enables the compare function. 0 Compare function disabled. 1 Compare function enabled.
4 ACFGT	Compare function greater than enable ACFGT configures the compare function to check the conversion result relative to the compare value register(s) (CV1 and CV2) based upon the value of ACREN. The ACFE bit must be set for ACFGT to have any effect. 0 Configures less than threshold, outside range not inclusive and inside range not inclusive functionality based on the values placed in the CV1 and CV2 registers. 1 Configures greater than or equal to threshold, outside range inclusive and inside range inclusive functionality based on the values placed in the CV1 and CV2 registers.
3 ACREN	Compare function range enable ACREN configures the compare function to check if the conversion result of the input being monitored is either between or outside the range formed by the compare value registers (CV1 and CV2) determined by the value of ACFGT. The ACFE bit must be set for ACFGT to have any effect. 0 Range function disabled. Only the compare value 1 register (CV1) is compared. 1 Range function enabled. Both compare value registers (CV1 and CV2) are compared.
2 DMAEN	DMA enable 0 DMA is disabled. 1 DMA is enabled and will assert the ADC DMA request during a ADC conversion complete event noted by the assertion of any of the ADC COCO flags.
1–0 REFSEL	Voltage reference selection REFSEL bits select the voltage reference source used for conversions.

Table continues on the next page...

ADCx_SC2 field descriptions (continued)

Field	Description
	00 Default voltage reference pin pair (external pins V_{REFH} and V_{REFL}) 01 Alternate reference pair (V_{ALTH} and V_{ALTL}). This pair may be additional external pins or internal sources depending on MCU configuration. Consult the Chip Configuration information for details specific to this MCU. 10 Reserved 11 Reserved

31.3.7 Status and control register 3 (ADCx_SC3)

The SC3 register controls the calibration, continuous convert, and hardware averaging functions of the ADC module.

Addresses: ADC0_SC3 is 4003_B000h base + 24h offset = 4003_B024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CAL	CALF	0		ADCO	AVGE	AVGS	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADCx_SC3 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 CAL	Calibration CAL begins the calibration sequence when set. This bit stays set while the calibration is in progress and is cleared when the calibration sequence is completed. The CALF bit must be checked to determine the result of the calibration sequence. Once started, the calibration routine cannot be interrupted by writes to the ADC registers or the results will be invalid and the CALF bit will set. Setting the CAL bit will abort any current conversion.
6 CALF	Calibration failed flag CALF displays the result of the calibration sequence. The calibration sequence will fail if ADTRG = 1, any ADC register is written, or any stop mode is entered before the calibration sequence completes. The CALF bit is cleared by writing a 1 to this bit. 0 Calibration completed normally. 1 Calibration failed. ADC accuracy specifications are not guaranteed.

Table continues on the next page...

ADCx_SC3 field descriptions (continued)

Field	Description
5–4 Reserved	This read-only field is reserved and always has the value zero.
3 ADCO	Continuous conversion enable ADCO enables continuous conversions. 0 One conversion or one set of conversions if the hardware average function is enabled (AVGE=1) after initiating a conversion. 1 Continuous conversions or sets of conversions if the hardware average function is enabled (AVGE=1) after initiating a conversion.
2 AVGE	Hardware average enable AVGE enables the hardware average function of the ADC. 0 Hardware average function disabled. 1 Hardware average function enabled.
1–0 AVGS	Hardware average select AVGS determines how many ADC conversions will be averaged to create the ADC average result. 00 4 samples averaged. 01 8 samples averaged. 10 16 samples averaged. 11 32 samples averaged.

31.3.8 ADC offset correction register (ADCx_OFS)

The ADC offset correction register (OFS) contains the user selected or calibration generated offset error correction value. This register is a 2's complement, left justified, 16-bit value. The value in the offset correction registers (OFS) is subtracted from the conversion and the result is transferred into the result registers (Rn). If the result is above the maximum or below the minimum result value, it is forced to the appropriate limit for the current mode of operation.

Addresses: ADC0_OFS is 4003_B000h base + 28h offset = 4003_B028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																OFS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

ADCx_OFS field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 OFS	Offset error correction value

31.3.9 ADC plus-side gain register (ADCx_PG)

The plus-side gain register (PG) contains the gain error correction for the plus-side input in differential mode or the overall conversion in single-ended mode. PG, a 16-bit real number in binary format, is the gain adjustment factor, with the radix point fixed between ADPG15 and ADPG14. This register must be written by the user with the value described in the calibration procedure or the gain error specifications may not be met.

Addresses: ADC0_PG is 4003_B000h base + 2Ch offset = 4003_B02Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																PG															
W	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

ADCx_PG field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 PG	Plus-side gain

31.3.10 ADC minus-side gain register (ADCx_MG)

The minus-side gain register (MG) contains the gain error correction for the minus-side input in differential mode. This register is ignored in single-ended mode. MG, a 16-bit real number in binary format, is the gain adjustment factor, with the radix point fixed between ADMG15 and ADMG14. This register must be written by the user with the value described in the calibration procedure or the gain error specifications may not be met.

Addresses: ADC0_MG is 4003_B000h base + 30h offset = 4003_B030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																MG															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

ADCx_MG field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 MG	Minus-side gain

31.3.11 ADC plus-side general calibration value register (ADCx_CLPD)

The plus-side general calibration value registers (CLPx) contain calibration information that is generated by the calibration function. These registers contain seven calibration values of varying widths: CLP0[5:0], CLP1[6:0], CLP2[7:0], CLP3[8:0], CLP4[9:0], CLPS[5:0], and CLPD[5:0]. CLPx are automatically set once the self calibration sequence is done (CAL is cleared). If these registers are written by the user after calibration, the linearity error specifications may not be met.

Addresses: ADC0_CLPD is 4003_B000h base + 34h offset = 4003_B034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLPD					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

ADCx_CLPD field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero.
5–0 CLPD	Calibration value

31.3.12 ADC plus-side general calibration value register (ADCx_CLPS)

For more information, refer to CLPD register description.

Addresses: ADC0_CLPS is 4003_B000h base + 38h offset = 4003_B038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLPS					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

ADCx_CLPS field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero.
5–0 CLPS	Calibration value

31.3.13 ADC plus-side general calibration value register (ADCx_CLP4)

For more information, refer to CLPD register description.

Addresses: ADC0_CLP4 is 4003_B000h base + 3Ch offset = 4003_B03Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																						CLP4									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

ADCx_CLP4 field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero.
9–0 CLP4	Calibration value

31.3.14 ADC plus-side general calibration value register (ADCx_CLP3)

For more information, refer to CLPD register description.

Addresses: ADC0_CLP3 is 4003_B000h base + 40h offset = 4003_B040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																							CLP3								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

ADCx_CLP3 field descriptions

Field	Description
31–9 Reserved	This read-only field is reserved and always has the value zero.
8–0 CLP3	Calibration value

31.3.15 ADC plus-side general calibration value register (ADCx_CLP2)

For more information, refer to CLPD register description.

Addresses: ADC0_CLP2 is 4003_B000h base + 44h offset = 4003_B044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								CLP2							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

ADCx_CLP2 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7–0 CLP2	Calibration value

31.3.16 ADC plus-side general calibration value register (ADCx_CLP1)

For more information, refer to CLPD register description.

Addresses: ADC0_CLP1 is 4003_B000h base + 48h offset = 4003_B048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																									CLP1						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

ADCx_CLP1 field descriptions

Field	Description
31–7 Reserved	This read-only field is reserved and always has the value zero.
6–0 CLP1	Calibration value

31.3.17 ADC plus-side general calibration value register (ADCx_CLP0)

For more information, refer to CLPD register description.

Addresses: ADC0_CLP0 is 4003_B000h base + 4Ch offset = 4003_B04Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLP0					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

ADCx_CLP0 field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero.
5–0 CLP0	Calibration value

31.3.18 ADC minus-side general calibration value register (ADCx_CLMD)

CLMx contain calibration information that is generated by the calibration function. These registers contain seven calibration values of varying widths: CLM0[5:0], CLM1[6:0], CLM2[7:0], CLM3[8:0], CLM4[9:0], CLMS[5:0], and CLMD[5:0]. CLMx are automatically set once the self calibration sequence is done (CAL is cleared). If these registers are written by the user after calibration, the linearity error specifications may not be met.

Addresses: ADC0_CLMD is 4003_B000h base + 54h offset = 4003_B054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLMD					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

ADCx_CLMD field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero.
5–0 CLMD	Calibration value

31.3.19 ADC minus-side general calibration value register (ADCx_CLMS)

For more information, refer to CLMD register description.

Addresses: ADC0_CLMS is 4003_B000h base + 58h offset = 4003_B058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLMS					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

ADCx_CLMS field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero.
5–0 CLMS	Calibration value

31.3.20 ADC minus-side general calibration value register (ADCx_CLM4)

For more information, refer to CLMD register description.

Addresses: ADC0_CLM4 is 4003_B000h base + 5Ch offset = 4003_B05Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																						CLM4									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

ADCx_CLM4 field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero.
9–0 CLM4	Calibration value

31.3.21 ADC minus-side general calibration value register (ADCx_CLM3)

For more information, refer to CLMD register description.

Addresses: ADC0_CLM3 is 4003_B000h base + 60h offset = 4003_B060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																							CLM3								
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

ADCx_CLM3 field descriptions

Field	Description
31–9 Reserved	This read-only field is reserved and always has the value zero.
8–0 CLM3	Calibration value

31.3.22 ADC minus-side general calibration value register (ADCx_CLM2)

For more information, refer to CLMD register description.

Addresses: ADC0_CLM2 is 4003_B000h base + 64h offset = 4003_B064h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								CLM2							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

ADCx_CLM2 field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7–0 CLM2	Calibration value

31.3.23 ADC minus-side general calibration value register (ADCx_CLM1)

For more information, refer to CLMD register description.

Addresses: ADC0_CLM1 is 4003_B000h base + 68h offset = 4003_B068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																									CLM1						
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

ADCx_CLM1 field descriptions

Field	Description
31–7 Reserved	This read-only field is reserved and always has the value zero.
6–0 CLM1	Calibration value

31.3.24 ADC minus-side general calibration value register (ADCx_CLM0)

For more information, refer to CLMD register description.

Addresses: ADC0_CLM0 is 4003_B000h base + 6Ch offset = 4003_B06Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																										CLM0					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

ADCx_CLM0 field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value zero.
5–0 CLM0	Calibration value

31.4 Functional description

The ADC module is disabled during reset, in low power stop mode (refer to the Power Management information for details), or when the ADCH bits in SC1n are all high. The module is idle when a conversion has completed and another conversion has not been initiated. When it is idle and the asynchronous clock output enable is disabled (ADACKEN is 0), the module is in its lowest power state. The ADC can perform an analog-to-digital conversion on any of the software selectable channels. All modes perform conversion by a successive approximation algorithm.

To meet accuracy specifications, the ADC module must be calibrated using the on chip calibration function. See Calibration function for details on how to perform calibration.

When the conversion is completed, the result is placed in the data registers (Rn). The respective conversion complete flag (COCO) is then set and an interrupt is generated if the respective conversion complete interrupt has been enabled (AIEN=1).

The ADC module has the capability of automatically comparing the result of a conversion with the contents of the compare value registers. The compare function is enabled by setting the ACFE bit and operates with any of the conversion modes and configurations.

The ADC module has the capability of automatically averaging the result of multiple conversions. The hardware average function is enabled by setting the AVGE bit and operates with any of the conversion modes and configurations.

NOTE

> For the chip specific modes of operation, refer to the Power Management information of this MCU.

31.4.1 Clock select and divide control

One of four clock sources can be selected as the clock source for the ADC module. This clock source is then divided by a configurable value to generate the input clock to the converter (ADCK). The clock is selected from one of the following sources by means of the ADICLK bits.

- The bus clock. This is the default selection following reset.
- The bus clock divided by two. For higher bus clock rates, this allows a maximum divide by 16 of the bus clock with using the ADIV bits.
- ALTCLK, as defined for this MCU. Refer to the Chip Configuration information.
- The asynchronous clock (ADACK). This clock is generated from a clock source within the ADC module. Note that when the ADACK clock source is selected, it is not required to be active prior to conversion start. When it is selected and it is not active prior to a conversion start (ADACKEN=0), the asynchronous clock is activated at the start of a conversion and shuts off when conversions are terminated. In this case, there is an associated clock startup delay each time the clock source is re-activated. To avoid the conversion time variability and latency associated with the ADACK clock startup, set ADACKEN=1 and wait the worst case startup time of 5 µs prior to initiating any conversions using the ADACK clock source. Conversions are possible using ADACK as the input clock source while the MCU is in Normal Stop mode. Refer to Power Control for more information.

Whichever clock is selected, its frequency must fall within the specified frequency range for ADCK. If the available clocks are too slow, the ADC may not perform according to specifications. If the available clocks are too fast, the clock must be divided to the appropriate frequency. This divider is specified by the ADIV bits and can be divide-by 1, 2, 4, or 8.

31.4.2 Voltage reference selection

The ADC can be configured to accept one of the two voltage reference pairs as the reference voltage (V_{REFSH} and V_{REFSL}) used for conversions. Each pair contains a positive reference that must be between the minimum Ref Voltage High and V_{DDA}, and a ground reference that must be at the same potential as V_{SSA}. The two pairs are external (V_{REFH} and V_{REFL}) and alternate (V_{ALTH} and V_{ALTL}). These voltage references are selected using the REFSEL bits in the SC2 register. The alternate (V_{ALTH} and V_{ALTL}) voltage reference pair may select additional external pins or internal sources depending on MCU configuration. Refer to the Chip Configuration information on the Voltage References specific to this MCU.

31.4.3 Hardware trigger and channel selects

The ADC module has a selectable asynchronous hardware conversion trigger, ADHWT, that is enabled when the ADTRG bit is set and a hardware trigger select event (ADHWTSn) has occurred. This source is not available on all MCUs. Refer to the Chip Configuration chapter for information on the ADHWT source and the ADHWTSn configurations specific to this MCU.

When a ADHWT source is available and hardware trigger is enabled (ADTRG=1), a conversion is initiated on the rising edge of ADHWT after a hardware trigger select event (ADHWTSn) has occurred. If a conversion is in progress when a rising edge of a trigger occurs, the rising edge is ignored. In continuous convert configuration, only the initial rising edge to launch continuous conversions is observed, and until conversion gets aborted the ADC continues to do conversions on the same ADC status and control register that initiated the conversion. The hardware trigger function operates in conjunction with any of the conversion modes and configurations.

The hardware trigger select event (ADHWTSn) must be set prior to the receipt of the ADHWT signal. If these conditions are not met, the converter may ignore the trigger or use the incorrect configuration. If a hardware trigger select event gets asserted during a conversion, it must stay asserted until the end of current conversion and remain set until the receipt of the ADHWT signal to trigger a new conversion. The channel and status fields selected for the conversion depend on the active trigger select signal (ADHWTSA active selects SC1A; ADHWTSn active selects SC1n).

Note

> Asserting more than one hardware trigger select signal (ADHWTSn) at the same time results in unknown results. To avoid this, select only one hardware trigger select signal (ADHWTSn) prior to the next intended conversion.

When the conversion is completed, the result is placed in the data registers associated with the ADHWTSn received (ADHWTSA active selects RA register; ADHWTSn active selects Rn register). The conversion complete flag associated with the ADHWTSn received (the COCO bit in SC1n register) is then set and an interrupt is generated if the respective conversion complete interrupt has been enabled (AIEN=1).

31.4.4 Conversion control

Conversions can be performed as determined by the CFG1[MODE] bits and the SC1n[DIFF] bit as shown in the description of CFG1[MODE].

Conversions can be initiated by a software or hardware trigger. In addition, the ADC module can be configured for low power operation, long sample time, continuous conversion, hardware average, and automatic compare of the conversion result to a software determined compare value.

31.4.4.1 Initiating conversions

A conversion is initiated:

- Following a write to SC1A register (with ADCH bits not all 1's) if software triggered operation is selected (ADTRG=0).
- Following a hardware trigger (ADHWT) event if hardware triggered operation is selected (ADTRG=1) and a hardware trigger select event (ADHWTSn) has occurred. The channel and status fields selected depend on the active trigger select signal (ADHWTSA active selects SC1A register; ADHWTSn active selects SC1n register; if neither is active, the off condition is selected).

Note

> Selecting more than one hardware trigger select signal (ADHWTSn) prior to a conversion completion will result in unknown results. To avoid this, select only one hardware trigger select signal (ADHWTSn) prior to a conversion completion.

- Following the transfer of the result to the data registers when continuous conversion is enabled (ADCO=1).

If continuous conversions are enabled, a new conversion is automatically initiated after the completion of the current conversion. In software triggered operation (ADTRG=0), continuous conversions begin after SC1A register is written and continue until aborted. In hardware triggered operation (ADTRG=1 and one ADHWTSn event has occurred), continuous conversions begin after a hardware trigger event and continue until aborted.

If hardware averaging is enabled, a new conversion is automatically initiated after the completion of the current conversion until the correct number of conversions is completed. In software triggered operation, conversions begin after SC1A register is written. In hardware triggered operation, conversions begin after a hardware trigger. If continuous conversions are also enabled, a new set of conversions to be averaged are initiated following the last of the selected number of conversions.

31.4.4.2 Completing conversions

A conversion is completed when the result of the conversion is transferred into the data result registers, Rn. If the compare functions are disabled, this is indicated by the setting of the COCO bit in the respective SC1n register. If hardware averaging is enabled, the respective COCO bit sets only if the last of the selected number of conversions is completed. If the compare function is enabled, the respective COCO bit sets and conversion result data is transferred only if the compare condition is true. If both hardware averaging and compare functions are enabled then the respective COCO bit sets only if the last of the selected number of conversions is completed and the compare condition is true. An interrupt is generated if the respective AIEN bit is high at the time that the respective COCO bit is set.

31.4.4.3 Aborting conversions

Any conversion in progress is aborted when:

- Writing to SC1A register while it is actively controlling a conversion, aborts the current conversion. In software trigger mode (ADTRG=0), a write to SC1A register initiates a new conversion (if the ADCH field in SC1A is equal to a value other than all 1s). Writing to any of the SC1(B-n) registers while that specific SC1(B-n) register is actively controlling a conversion aborts the current conversion.The SC1(B-n) registers are not used for software trigger operation and therefore writes to the SC1(B-n) registers do not initiate a new conversion.
- A write to any ADC register besides the SC1A:SC1n registers occurs. This indicates a mode of operation change has occurred and the current conversion is therefore invalid.

- The MCU is reset or enters Low Power Stop modes.
- The MCU enters Normal Stop mode with ADACK not enabled.

When a conversion is aborted, the contents of the data registers, Rn, are not altered. The data registers continue to be the values transferred after the completion of the last successful conversion. If the conversion was aborted by a reset or Low Power Stop modes, RA and R n return to their reset states.

31.4.4.4 Power control

The ADC module remains in its idle state until a conversion is initiated. If ADACK is selected as the conversion clock source, but the asynchronous clock output is disabled (ADACKEN=0), the ADACK clock generator also remains in its idle state (disabled) until a conversion is initiated. If the asynchronous clock output is enabled (ADACKEN=1), it remains active regardless of the state of the ADC or the MCU power mode.

Power consumption when the ADC is active can be reduced by setting ADLPC. This results in a lower maximum value for f_{ADCK} .

31.4.4.5 Sample time and total conversion time

For short sample (ADLSMP=0), there is a 2-cycle adder for first conversion over the base sample time of 4 ADCK cycles. For high speed conversions (ADHSC=1), there is an additional 2-cycle adder on any conversion. The table below summarizes sample times for the possible ADC configurations.

ADC Configuration			**Sample time (ADCK cycles)**	
ADLSMP	ADLSTS	ADHSC	First or Single	Subsequent
0	X	0	6	4
1	00	0	24	
1	01	0	16	
1	10	0	10	
1	11	0	6	
0	X	1	8	6
1	00	1	26	
1	01	1	18	
1	10	1	12	

Table continues on the next page...

ADC Configuration			Sample time (ADCK cycles)
1	11	1	8

The total conversion time depends upon: the sample time (as determined by ADLSMP and ADLSTS bits), the MCU bus frequency, the conversion mode (as determined by MODE and SC1n[DIFF] bits), the high speed configuration (ADHSC bit), and the frequency of the conversion clock (f_{ADCK}).

The ADHSC bit is used to configure a higher clock input frequency. This will allow faster overall conversion times. To meet internal ADC timing requirements, the ADHSC bit adds additional ADCK cycles. Conversions with ADHSC = 1 take two more ADCK cycles. ADHSC should be used when the ADCLK exceeds the limit for ADHSC = 0.

After the module becomes active, sampling of the input begins. ADLSMP and ADLSTS select between sample times based on the conversion mode that is selected. When sampling is completed, the converter is isolated from the input channel and a successive approximation algorithm is performed to determine the digital value of the analog signal. The result of the conversion is transferred to Rn upon completion of the conversion algorithm.

If the bus frequency is less than the f_{ADCK} frequency, precise sample time for continuous conversions cannot be guaranteed when short sample is enabled (ADLSMP=0).

The maximum total conversion time is determined by the clock source chosen and the divide ratio selected. The clock source is selectable by the ADICLK bits, and the divide ratio is specified by the ADIV bits.

The maximum total conversion time for all configurations is summarized in the equation below. Refer to the following tables for the variables referenced in the equation.

$$\mathrm{ConversionTime} = \mathrm{SFCAdder} + \mathrm{AverageNum} \times \left(\mathrm{BCT} + \mathrm{LSTAdder} + \mathrm{HSCAdder}\right)$$

Figure 31-62. Conversion time equation

Table 31-70. Single or first continuous time adder (SFCAdder)

ADLSMP	ADACKEN	ADICLK	Single or first continuous time adder (SFCAdder)
1	x	0x, 10	3 ADCK cycles + 5 bus clock cycles
1	1	11	3 ADCK cycles + 5 bus clock cycles[1]
1	0	11	5 μs + 3 ADCK cycles + 5 bus clock cycles
0	x	0x, 10	5 ADCK cycles + 5 bus clock cycles
0	1	11	5 ADCK cycles + 5 bus clock cycles[1]
0	0	11	5 μs + 5 ADCK cycles + 5 bus clock cycles

1. To achieve this time, ADACKEN must be 1 for at least 5 μs prior to the conversion is initiated.

Table 31-71. Average number factor (AverageNum)

AVGE	AVGS[1:0]	Average number factor (AverageNum)
0	xx	1
1	00	4
1	01	8
1	10	16
1	11	32

Table 31-72. Base Conversion Time (BCT)

Mode	Base conversion time (BCT)
8b s.e.	17 ADCK cycles
9b diff	27 ADCK cycles
10b s.e.	20 ADCK cycles
11b diff	30 ADCK cycles
12b s.e.	20 ADCK cycles
13b diff	30 ADCK cycles
16b s.e.	25 ADCK cycles
16b diff	34 ADCK cycles

Table 31-73. Long sample time adder (LSTAdder)

ADLSMP	ADLSTS	Long sample time adder (LSTAdder)
0	xx	0 ADCK cycles
1	00	20 ADCK cycles
1	01	12 ADCK cycles
1	10	6 ADCK cycles
1	11	2 ADCK cycles

Table 31-74. High Speed Conversion time Adder (HSCAdder)

ADHSC	High Speed Conversion Time Adder (HSCAdder)
0	0 ADCK cycles
1	2 ADCK cycles

Note

The ADCK frequency must be between f_{ADCK} minimum and f_{ADCK} maximum to meet ADC specifications.

31.4.4.6 Conversion time examples

The following examples use Figure 31-62 and the information provided in Table 31-70 through Table 31-74.

31.4.4.6.1 Typical conversion time configuration

A typical configuration for ADC conversion is: 10-bit mode, with the bus clock selected as the input clock source, the input clock divide-by-1 ratio selected, and a bus frequency of 8 MHz, long sample time disabled and high speed conversion disabled. The conversion time for a single conversion is calculated by using Figure 31-62 and the information provided in Table 31-70 through Table 31-74. The table below list the variables of Figure 31-62.

Table 31-75. Typical conversion time

Variable	Time
SFCAdder	5 ADCK cycles + 5 bus clock cycles
AverageNum	1
BCT	20 ADCK cycles
LSTAdder	0
HSCAdder	0

The resulting conversion time is generated using the parameters listed in the proceeding table. Therefore, for a bus clock equal to 8 MHz and an ADCK equal to 8 MHz the resulting conversion time is 3.75 µs.

31.4.4.6.2 Long conversion time configuration

A configuration for long ADC conversion is: 16-bit differential mode with the bus clock selected as the input clock source, the input clock divide-by-8 ratio selected, a bus frequency of 8 MHz, long sample time enabled, configured for longest adder, high speed conversion disabled, and average enabled for 32 conversions. The conversion time for this conversion is calculated by using Figure 31-62 and the information provided in Table 31-70 through Table 31-74. The following table lists the variables of the Figure 31-62.

Table 31-76. Typical conversion time

Variable	Time
SFCAdder	3 ADCK cycles + 5 bus clock cycles
AverageNum	32
BCT	34 ADCK cycles
LSTAdder	20 ADCK cycles

Table continues on the next page...

Table 31-76. Typical conversion time (continued)

Variable	Time
HSCAdder	0

The resulting conversion time is generated using the parameters listed in the preceding table. Therefore, for bus clock equal to 8 MHz and ADCK equal to 1 MHz, the resulting conversion time is 57.625 µs (AverageNum). This results in a total conversion time of 1.844 ms.

31.4.4.6.3 Short conversion time configuration

A configuration for short ADC conversion is: 8-bit single ended mode with the bus clock selected as the input clock source, the input clock divide-by-1 ratio selected, a bus frequency of 20 MHz, long sample time disabled, and high speed conversion enabled. The conversion time for this conversion is calculated by using Figure 31-62 and the information provided in Table 31-70 through Table 31-74. The table below list the variables of Figure 31-62.

Table 31-77. Typical conversion time

Variable	Time
SFCAdder	5 ADCK cycles + 5 bus clock cycles
AverageNum	1
BCT	17 ADCK cycles
LSTAdder	0 ADCK cycles
HSCAdder	2

The resulting conversion time is generated using the parameters listed in in the preceding table. Therefore, for bus clock equal to 20 MHz and ADCK equal to 20 MHz, the resulting conversion time is 1.45 µs.

31.4.4.7 Hardware average function

The hardware average function can be enabled (AVGE=1) to perform a hardware average of multiple conversions. The number of conversions is determined by the AVGS[1:0] bits, which select 4, 8, 16, or 32 conversions to be averaged. While the hardware average function is in progress, the ADACT bit will be set.

After the selected input is sampled and converted, the result is placed in an accumulator from which an average is calculated once the selected number of conversions has been completed. When hardware averaging is selected, the completion of a single conversion will not set the COCO bit.

If the compare function is either disabled or evaluates true, after the selected number of conversions are completed, the average conversion result is transferred into the data result registers, Rn, and the COCO bit is set. An ADC interrupt is generated upon the setting of COCO if the respective ADC interrupt is enabled (AIEN=1).

Note

> The hardware average function can perform conversions on a channel while the MCU is in Wait or Normal Stop modes. The ADC interrupt wakes the MCU when the hardware average is completed if SC1n[AIEN] bit was set.

31.4.5 Automatic compare function

The compare function can be configured to check if the result is less than or greater-than-or-equal-to a single compare value, or if the result falls within or outside a range determined by two compare values. The compare mode is determined by ACFGT, ACREN, and the values in the compare value registers (CV1 and CV2). After the input is sampled and converted, the compare values (CV1 and CV2) are used as described in the following table.There are six compare modes as shown in the following table.

Table 31-78. Compare modes

ACFGT	ACREN	ADCCV1 relative to ADCCV2	Function	Compare mode description
0	0	—	Less than threshold	Compare true if the result is less than the CV1 registers.
1	0	—	Greater than or equal to threshold	Compare true if the result is greater than or equal to CV1 registers.
0	1	Less than or equal	Outside range, not inclusive	Compare true if the result is less than CV1 **Or** the result is greater than CV2.
0	1	Greater than	Inside range, not inclusive	Compare true if the result is less than CV1 **And** the result is greater than CV2.
1	1	Less than or equal	Inside range, inclusive	Compare true if the result is greater than or equal to CV1 **And** the result is less than or equal to CV2.
1	1	Greater than	Outside range, inclusive	Compare true if the result is greater than or equal to CV1 **Or** the result is less than or equal to CV2.

With the ADC range enable bit set, ACREN =1, and if compare value register 1 (CV1 value) is less than or equal to the compare value register 2 (CV2 value), then setting ACFGT will select a trigger-if-inside-compare-range inclusive-of-endpoints function. Clearing ACFGT will select a trigger-if-outside-compare-range, not-inclusive-of-endpoints function.

If CV1 is greater than CV2, setting ACFGT will select a trigger-if-outside-compare-range, inclusive-of-endpoints function. Clearing ACFGT will select a trigger-if-inside-compare-range, not-inclusive-of-endpoints function.

If the condition selected evaluates true, COCO is set.

Upon completion of a conversion while the compare function is enabled, if the compare condition is not true, COCO is not set and the conversion result data will not be transferred to the result register. If the hardware averaging function is enabled, the compare function compares the averaged result to the compare values. The same compare function definitions apply. An ADC interrupt is generated upon the setting of COCO if the respective ADC interrupt is enabled (AIEN=1).

Note

> The compare function can monitor the voltage on a channel while the MCU is in Wait or Normal Stop modes. The ADC interrupt wakes the MCU when the compare condition is met.

31.4.6 Calibration function

The ADC contains a self-calibration function that is required to achieve the specified accuracy. Calibration must be run, or valid calibration values written, after any reset and before a conversion is initiated. The calibration function sets the offset calibration value, the minus-side calibration values, and the plus-side calibration values. The offset calibration value is automatically stored in the ADC offset correction register (OFS), and the plus-side and minus-side calibration values are automatically stored in the ADC plus-side and minus-side calibration (CLPx and CLMx) registers. The user must configure the ADC correctly prior to calibration, and must generate the plus-side and minus-side gain calibration results and store them in the ADC plus-side gain register (PG) after the calibration function completes.

Prior to calibration, the user must configure the ADC's clock source and frequency, low power configuration, voltage reference selection, sample time, and high speed configuration according to the application's clock source availability and needs. If the application uses the ADC in a wide variety of configurations, the configuration for which the highest accuracy is required should be selected, or multiple calibrations can be done for the different configurations. For best calibration results, it is recommended to set hardware averaging to maximum (AVGE=1, AVGS=11 for average of 32), ADC clock frequency f_{ADCK} less than or equal to 4 MHz, $V_{REFH}=V_{DDA}$, and to calibrate at nominal voltage and temperature. The input channel, conversion mode continuous function, compare function, resolution mode, and differential/single-ended mode are all ignored during the calibration function.

To initiate calibration, the user sets the CAL bit and the calibration will automatically begin if the ADTRG bit is 0. If ADTRG is 1, the CAL bit will not get set and the calibration fail flag (CALF) will be set. While calibration is active, no ADC register can be written and no stop mode may be entered, or the calibration routine will be aborted causing the CAL bit to clear and the CALF bit to set. At the end of a calibration sequence, the COCO bit of the SC1A register will be set. The AIEN bit can be used to allow an interrupt to occur at the end of a calibration sequence. At the end of the calibration routine, if the CALF bit is not set, the automatic calibration routine completed successfully.

To complete calibration, the user must generate the gain calibration values using the following procedure:

1. Initialize (clear) a 16-bit variable in RAM.
2. Add the plus-side calibration results CLP0, CLP1, CLP2, CLP3, CLP4, and CLPS to the variable.
3. Divide the variable by two.
4. Set the MSB of the variable.
5. The previous two steps can be achieved by setting the carry bit, rotating to the right through the carry bit on the high byte and again on the low byte.
6. Store the value in the plus-side gain calibration register (PG).
7. Repeat the procedure for the minus-side gain calibration value.

When calibration is complete, the user may reconfigure and use the ADC as desired. A second calibration may also be performed if desired by clearing and again setting the CAL bit.

Overall, the calibration routine may take as many as 14k ADCK cycles and 100 bus cycles, depending on the results and the clock source chosen. For an 8 MHz clock source, this length amounts to about 1.7 ms. To reduce this latency, the calibration values (offset, plus-side and minus-side gain, and plus-side and minus-side calibration values) may be stored in flash memory after an initial calibration and recovered prior to the first ADC conversion. This method should reduce the calibration latency to 20 register store operations on all subsequent power, reset, or Low Power Stop mode recoveries.

31.4.7 User defined offset function

The ADC offset correction register (OFS) contains the user selected or calibration generated offset error correction value. This register is a 2’s complement, left justified. The value in the offset correction register (OFS) is subtracted from the conversion and the result is transferred into the result registers (Rn). If the result is above the maximum or below the minimum result value, it is forced to the appropriate limit for the current mode of operation.

The formatting of the ADC offset correction register is different from the data result register (Rn) to preserve the resolution of the calibration value regardless of the conversion mode selected. Lower order bits are ignored in lower resolution modes. For example, in 8-bit single-ended mode, the bits OFS[14:7] are subtracted from D[7:0]; bit OFS[15] indicates the sign (negative numbers are effectively added to the result) and bits OFS[6:0] are ignored. The same bits are used in 9-bit differential mode since bit OFS[15] indicates the sign bit, which maps to bit D[8]. For 16-bit differential mode, all bits OFS[15:0] are directly subtracted from the conversion result data D[15:0]. In 16-bit single-ended mode, there is no bit in the offset correction register corresponding to the least significant result bit D[0], so odd values (-1 or +1, and so on) cannot be subtracted from the result.

OFS is automatically set according to calibration requirements once the self calibration sequence is done (CAL is cleared). The user may write to OFS to override the calibration result if desired. If the offset correction register is written by the user to a value that is different from the calibration value, the ADC error specifications may not be met. It is recommended that the value generated by the calibration function be stored in memory before overwriting with a user specified value.

Note

> There is an effective limit to the values of offset that can be set by the user. If the magnitude of the offset is too great, the results of the conversions will cap off at the limits.

The offset calibration function may be employed by the user to remove application offsets or DC bias values. The offset correction register, OFS may be written with a number in 2's complement format and this offset will be subtracted from the result (or hardware averaged value). To add an offset, store the negative offset in 2's complement format and the effect will be an addition. An offset correction that results in an out-of-range value will be forced to the minimum or maximum value (the minimum value for single-ended conversions is 0x0000; for a differential conversion it is 0x8000).

To preserve accuracy, the calibrated offset value initially stored in the OFS register must be added to the user defined offset. For applications that may change the offset repeatedly during operation, it is recommended to store the initial offset calibration value in flash so it can be recovered and added to any user offset adjustment value and the sum stored in the OFS register.

31.4.8 Temperature sensor

The ADC module includes a temperature sensor whose output is connected to one of the ADC analog channel inputs. The following equation provides an approximate transfer function of the temperature sensor.

$$\mathrm{Temp} = 25 - \left(\left(V_{TEMP} - V_{TEMP25}\right) \div m\right)$$

Figure 31-63. Approximate transfer function of the temperature sensor

where:

- V_{TEMP} is the voltage of the temperature sensor channel at the ambient temperature.
- V_{TEMP25} is the voltage of the temperature sensor channel at 25 °C.
- m is referred as temperature sensor slope in the device data sheet. It is the hot or cold voltage versus temperature slope in V/°C.

For temperature calculations, use the V_{TEMP25} and temperature sensor slope values from the ADC Electricals table.

In application code, the user reads the temperature sensor channel, calculates V_{TEMP}, and compares to V_{TEMP25}. If V_{TEMP} is greater than V_{TEMP25} the cold slope value is applied in the preceding equation. If V_{TEMP} is less than V_{TEMP25}, the hot slope value is applied in the preceding equation. ADC Electricals table may only specify one temperature sensor slope value. In that case, the user could use the same slope for the calculation across the operational temperature range.

For more information on using the temperature sensor, see the application note titled *Temperature Sensor for the HCS08 Microcontroller Family* (document AN3031).

31.4.9 MCU wait mode operation

Wait mode is a lower power-consumption standby mode from which recovery is fast because the clock sources remain active. If a conversion is in progress when the MCU enters Wait mode, it continues until completion. Conversions can be initiated while the MCU is in Wait mode by means of the hardware trigger or if continuous conversions are enabled.

The bus clock, bus clock divided by two, and ADACK are available as conversion clock sources while in Wait mode. The use of ALTCLK as the conversion clock source in Wait is dependent on the definition of ALTCLK for this MCU. Refer to the Chip Configuration information on ALTCLK specific to this MCU.

If the compare and hardware averaging functions are disabled, a conversion complete event sets the COCO and generates an ADC interrupt to wake the MCU from Wait mode if the respective ADC interrupt is enabled (AIEN=1). If the hardware averaging function is enabled, the COCO will set (and generate an interrupt if enabled) when the selected number of conversions are completed. If the compare function is enabled, the COCO will set (and generate an interrupt if enabled) only if the compare conditions are met. If a single conversion is selected and the compare trigger is not met, the ADC will return to its idle state and cannot wake the MCU from Wait mode unless a new conversion is initiated by the hardware trigger.

31.4.10 MCU Normal Stop mode operation

Stop mode is a low power-consumption standby mode during which most or all clock sources on the MCU are disabled.

31.4.10.1 Normal Stop mode with ADACK disabled

If the asynchronous clock, ADACK, is not selected as the conversion clock, executing a stop instruction aborts the current conversion and places the ADC in its idle state. The contents of the ADC registers, including Rn, are unaffected by Normal Stop mode. After exiting from Normal Stop mode, a software or hardware trigger is required to resume conversions.

31.4.10.2 Normal Stop mode with ADACK enabled

If ADACK is selected as the conversion clock, the ADC continues operation during Normal Stop mode. Refer to the Chip Configuration chapter for configuration information for this MCU.

If a conversion is in progress when the MCU enters Normal Stop mode, it continues until completion. Conversions can be initiated while the MCU is in Normal Stop mode by means of the hardware trigger or if continuous conversions are enabled.

If the compare and hardware averaging functions are disabled, a conversion complete event sets the COCO and generates an ADC interrupt to wake the MCU from Normal Stop mode if the respective ADC interrupt is enabled (AIEN = 1). The result register will contain the data from the first completed conversion that occurred during Normal Stop mode. If the hardware averaging function is enabled, the COCO will set (and generate an interrupt if enabled) when the selected number of conversions are completed. If the compare function is enabled, the COCO will set (and generate an interrupt if enabled) only if the compare conditions are met. If a single conversion is selected and the compare is not true, the ADC will return to its idle state and cannot wake the MCU from Normal Stop mode unless a new conversion is initiated by another hardware trigger.

31.4.11 MCU Low Power Stop mode operation

The ADC module is automatically disabled when the MCU enters Low Power Stop mode. All module registers contain their reset values following exit from Low Power Stop mode. Therefore, the module must be re-enabled and re-configured following exit from Low Power Stop mode.

NOTE

For the chip specific modes of operation, refer to the Power Management information for the device.

31.5 Initialization information

This section gives an example that provides some basic direction on how to initialize and configure the ADC module. You can configure the module for 16-bit, 12-bit, 10-bit, or 8-bit single-ended resolution or 16-bit, 13-bit, 11-bit, or 9-bit differential resolution, single or continuous conversion, and a polled or interrupt approach, among many other options. Refer to Table 31-73, Table 31-74, and Table 31-75 for information used in this example.

Note

Hexadecimal values are designated by a preceding 0x, binary values designated by a preceding %, and decimal values have no preceding character.

31.5.1 ADC module initialization example

This section provides details about the ADC module initialization.

31.5.1.1 Initialization sequence

Before the ADC module can be used to complete conversions, an initialization procedure must be performed. A typical sequence is as follows:

1. Calibrate the ADC by following the calibration instructions in Calibration function.
2. Update the configuration register (CFG) to select the input clock source and the divide ratio used to generate the internal clock, ADCK. This register is also used for selecting sample time and low-power configuration.
3. Update status and control register 2 (SC2) to select the conversion trigger (hardware or software) and compare function options, if enabled.
4. Update status and control register 3 (SC3) to select whether conversions will be continuous or completed only once (ADCO) and to select whether to perform hardware averaging.
5. Update the status and control register (SC1:SC1n) to select whether conversions will be single-ended or differential and to enable or disable conversion complete interrupts. Also, select the input channel on which to perform conversions.

31.5.1.2 Pseudo-code example

In this example, the ADC module is set up with interrupts enabled to perform a single 10-bit conversion at low power with a long sample time on input channel 1, where the internal ADCK clock is derived from the bus clock divided by 1.

CFG1 = 0x98 (%10011000)

```
Bit 7   ADLPC    1      Configures for low power (lowers maximum clock speed.
Bit 6:5 ADIV     00     Sets the ADCK to the input clock ÷ 1.
Bit 4   ADLSMP   1      Configures for long sample time.
```

```
        Bit 3:2   MODE   10    Selects the single-ended 10-bit conversion, differential 11-
bit conversion.
        Bit 1:0   ADICLK   00    Selects the bus clock.
```

SC2 = 0x00 (%00000000)

```
        Bit 7   ADACT   0       Flag indicates if a conversion is in progress.
        Bit 6   ADTRG   0       Software trigger selected.
        Bit 5   ACFE    0       Compare function disabled.
        Bit 4   ACFGT   0       Not used in this example.
        Bit 3   ACREN   0      Compare range disabled.
        Bit 2   DMAEN   0      DMA request disabled.
        Bit 1:0 REFSEL  00     Selects default voltage reference pin pair (External pins VREFH
and VREFL).
```

SC1A = 0x41 (%01000001)

```
        Bit 7   COCO   0       Read-only flag which is set when a conversion completes.
        Bit 6   AIEN   1       Conversion complete interrupt enabled.
        Bit 5   DIFF   0       Single-ended conversion selected.
        Bit 4:0 ADCH   00001   Input channel 1 selected as ADC input channel.
```

RA = 0xxx

```
        Holds results of conversion.
```

CV = 0xxx

```
        Holds compare value when compare function enabled.
```

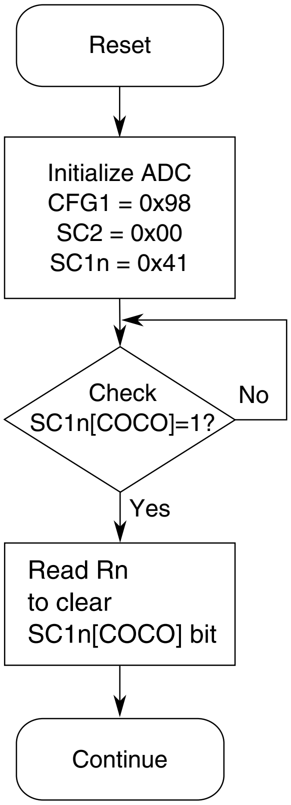

Reset
Initialize ADC
CFG1 = 0x98
SC2 = 0x00
SC1n = 0x41
Check
SC1n[COCO]=1?
No
Yes
Read Rn
to clear
SC1n[COCO] bit
Continue

Figure 31-64. Initialization Flowchart for Example

31.6 Application information

This section contains information for using the ADC module in applications. The ADC has been designed to be integrated into a microcontroller for use in embedded control applications requiring an ADC.

31.6.1 External pins and routing

The following sections discuss the external pins associated with the ADC module and how they should be used for best results.

31.6.1.1 Analog supply pins

The ADC module has analog power and ground supplies (V_{DDA} and V_{SSA}) available as separate pins on some devices. V_{SSA} is shared on the same pin as the MCU digital VSS on some devices. On other devices, V_{SSA} and V_{DDA} are shared with the MCU digital supply pins. In these cases, there are separate pads for the analog supplies bonded to the same pin as the corresponding digital supply so that some degree of isolation between the supplies is maintained.

When available on a separate pin, both V_{DDA} and V_{SSA} must be connected to the same voltage potential as their corresponding MCU digital supply (V_{DD} and V_{SS}) and must be routed carefully for maximum noise immunity and bypass capacitors placed as near as possible to the package.

If separate power supplies are used for analog and digital power, the ground connection between these supplies must be at the V_{SSA} pin. This should be the only ground connection between these supplies if possible. The V_{SSA} pin makes a good single point ground location.

31.6.1.2 Analog voltage reference pins

In addition to the analog supplies, the ADC module has connections for two reference voltage inputs used by the converter, V_{REFSH} and V_{REFSL}. V_{REFSH} is the high reference voltage for the converter. V_{REFSL} is the low reference voltage for the converter.

The ADC can be configured to accept one of two voltage reference pairs for V_{REFSH} and V_{REFSL}. Each pair contains a positive reference and a ground reference. The two pairs are external (V_{REFH} and V_{REFL}) and alternate (V_{ALTH} and V_{ALTL}). These voltage references are selected using the REFSEL bits in the SC2 register. The alternate (V_{ALTH} and V_{ALTL}) voltage reference pair may select additional external pins or internal sources depending on MCU configuration. Refer to the Chip Configuration information on the Voltage References specific to this MCU.

In some packages, the external or alternate pairs are connected in the package to V_{DDA} and V_{SSA}, respectively. One of these positive references may be shared on the same pin as V_{DDA} on some devices. One of these ground references may be shared on the same pin as V_{SSA} on some devices.

If externally available, the positive reference may be connected to the same potential as V_{DDA} or may be driven by an external source to a level between the minimum Ref Voltage High and the V_{DDA} potential (the positive reference must never exceed V_{DDA}). If externally available, the ground reference must be connected to the same voltage potential as V_{SSA}. The voltage reference pairs must be routed carefully for maximum noise immunity and bypass capacitors placed as near as possible to the package.

AC current in the form of current spikes required to supply charge to the capacitor array at each successive approximation step is drawn through the V_{REFH} and V_{REFL} loop. The best external component to meet this current demand is a 0.1 μF capacitor with good high frequency characteristics. This capacitor is connected between V_{REFH} and V_{REFL} and must be placed as near as possible to the package pins. Resistance in the path is not recommended because the current causes a voltage drop that could result in conversion errors. Inductance in this path must be minimum (parasitic only).

31.6.1.3 Analog input pins

The external analog inputs are typically shared with digital I/O pins on MCU devices.

Empirical data shows that capacitors on the analog inputs improve performance in the presence of noise or when the source impedance is high. Use of 0.01 μF capacitors with good high-frequency characteristics is sufficient. These capacitors are not necessary in all cases, but when used they must be placed as near as possible to the package pins and be referenced to V_{SSA}.

For proper conversion, the input voltage must fall between V_{REFH} and V_{REFL}. If the input is equal to or exceeds V_{REFH}, the converter circuit converts the signal to 0xFFF (full scale 12-bit representation), 0x3FF (full scale 10-bit representation) or 0xFF (full scale 8-bit representation). If the input is equal to or less than V_{REFL}, the converter circuit converts it to 0x000. Input voltages between V_{REFH} and V_{REFL} are straight-line linear conversions. There is a brief current associated with V_{REFL} when the sampling capacitor is charging.

For minimal loss of accuracy due to current injection, pins adjacent to the analog input pins should not be transitioning during conversions.

31.6.2 Sources of error

Several sources of error exist for A/D conversions. These are discussed in the following sections.

31.6.2.1 Sampling error

For proper conversions, the input must be sampled long enough to achieve the proper accuracy.

$$RAS + RADIN = SC / (FMAX * NUMTAU * CADIN)$$

Figure 31-65. Sampling equation

Where:

RAS = External analog source resistance

SC = Number of ADCK cycles used during sample window

CADIN = Internal ADC input capacitance

NUMTAU = -ln(LSBERR / 2^N)

LSBERR = value of acceptable sampling error in LSBs

N = 8 in 8-bit mode, 10 in 10-bit mode, 12 in 12-bit mode or 16 in 16-bit mode

Higher source resistances or higher-accuracy sampling is possible by setting ADLSMP and changing the ADLSTS bits (to increase the sample window) or decreasing ADCK frequency to increase sample time.

31.6.2.2 Pin leakage error

Leakage on the I/O pins can cause conversion error if the external analog source resistance (R_{AS}) is high. If this error cannot be tolerated by the application, keep R_{AS} lower than $V_{REFH} / (4 \times I_{LEAK} \times 2^N)$ for less than 1/4 LSB leakage error (N = 8 in 8-bit mode, 10 in 10-bit mode, 12 in 12-bit mode, or 16 in 16-bit mode).

31.6.2.3 Noise-induced errors

System noise that occurs during the sample or conversion process can affect the accuracy of the conversion. The ADC accuracy numbers are guaranteed as specified only if the following conditions are met:

- There is a 0.1 μF low-ESR capacitor from V_{REFH} to V_{REFL}.
- There is a 0.1 μF low-ESR capacitor from V_{DDA} to V_{SSA}.
- If inductive isolation is used from the primary supply, an additional 1 μF capacitor is placed from V_{DDA} to V_{SSA}.
- V_{SSA} (and V_{REFL}, if connected) is connected to V_{SS} at a quiet point in the ground plane.
- Operate the MCU in Wait or Normal Stop mode before initiating (hardware triggered conversions) or immediately after initiating (hardware or software triggered conversions) the ADC conversion.
  - For software triggered conversions, immediately follow the write to the SC1 register with a wait instruction or stop instruction.
  - For Normal Stop mode operation, select ADACK as the clock source. Operation in Normal Stop reduces V_{DD} noise but increases effective conversion time due to stop recovery.
- There is no I/O switching, input or output, on the MCU during the conversion.

There are some situations where external system activity causes radiated or conducted noise emissions or excessive V_{DD} noise is coupled into the ADC. In these situations, or when the MCU cannot be placed in Wait or Normal Stop or I/O activity cannot be halted, these recommended actions may reduce the effect of noise on the accuracy:

- Place a 0.01 μF capacitor (C_{AS}) on the selected input channel to V_{REFL} or V_{SSA} (this improves noise issues, but affects the sample rate based on the external analog source resistance).
- Average the result by converting the analog input many times in succession and dividing the sum of the results. Four samples are required to eliminate the effect of a 1 LSB, one-time error.
- Reduce the effect of synchronous noise by operating off the asynchronous clock (ADACK) and averaging. Noise that is synchronous to ADCK cannot be averaged out.

31.6.2.4 Code width and quantization error

The ADC quantizes the ideal straight-line transfer function into 65536 steps (in 16-bit mode). Each step ideally has the same height (1 code) and width. The width is defined as the delta between the transition points to one code and the next. The ideal code width for an N bit converter (in this case N can be 16, 12, 10, or 8), defined as 1 LSB, is:

$$1\text{LSB} = \left(V_{REFH} \right) / 2^N$$

Figure 31-66. Ideal code width for an N bit converter

There is an inherent quantization error due to the digitization of the result. For 8-bit, 10-bit, or 12-bit conversions, the code transitions when the voltage is at the midpoint between the points where the straight line transfer function is exactly represented by the actual transfer function. Therefore, the quantization error will be ± 1/2 LSB in 8-bit, 10-bit, or 12-bit modes. As a consequence, however, the code width of the first (0x000) conversion is only 1/2 LSB and the code width of the last (0xFF or 0x3FF) is 1.5 LSB.

For 16-bit conversions, the code transitions only after the full code width is present, so the quantization error is -1 LSB to 0 LSB and the code width of each step is 1 LSB.

31.6.2.5 Linearity errors

The ADC may also exhibit non-linearity of several forms. Every effort has been made to reduce these errors, but the system designers should be aware of them because they affect overall accuracy. These errors are:

- Zero-scale error (E_{ZS}) (sometimes called offset): This error is defined as the difference between the actual code width of the first conversion and the ideal code width (1/2 LSB in 8-bit, 10-bit, or 12-bit modes and 1 LSB in 16-bit mode). If the first conversion is 0x001, the difference between the actual 0x001 code width and its ideal (1 LSB) is used.
- Full-scale error (E_{FS}): This error is defined as the difference between the actual code width of the last conversion and the ideal code width (1.5 LSB in 8-bit, 10-bit, or 12-bit modes and 1 LSB in 16-bit mode). If the last conversion is 0x3FE, the difference between the actual 0x3FE code width and its ideal (1 LSB) is used.
- Differential non-linearity (DNL): This error is defined as the worst-case difference between the actual code width and the ideal code width for all conversions.

- Integral non-linearity (INL): This error is defined as the highest-value the (absolute value of the) running sum of DNL achieves. More simply, this is the worst-case difference of the actual transition voltage to a given code and its corresponding ideal transition voltage, for all codes.
- Total unadjusted error (TUE): This error is defined as the difference between the actual transfer function and the ideal straight-line transfer function and includes all forms of error.

31.6.2.6 Code jitter, non-monotonicity, and missing codes

Analog-to-digital converters are susceptible to three special forms of error. These are code jitter, non-monotonicity, and missing codes.

Code jitter is when, at certain points, a given input voltage converts to one of two values when sampled repeatedly. Ideally, when the input voltage is infinitesimally smaller than the transition voltage, the converter yields the lower code (and vice-versa). However, even small amounts of system noise can cause the converter to be indeterminate (between two codes) for a range of input voltages around the transition voltage.

This error may be reduced by repeatedly sampling the input and averaging the result. Additionally, the techniques discussed in Noise-induced errors reduces this error.

Non-monotonicity is defined as when, except for code jitter, the converter converts to a lower code for a higher input voltage. Missing codes are those values never converted for any input value.

In 8-bit or 10-bit mode, the ADC is guaranteed to be monotonic and have no missing codes.

Chapter 32
Comparator (CMP)

32.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The comparator (CMP) module provides a circuit for comparing two analog input voltages. The comparator circuit is designed to operate across the full range of the supply voltage, known as rail-to-rail operation.

The Analog MUX (ANMUX) provides a circuit for selecting an analog input signal from eight channels. One signal is provided by the 6-bit digital-to-analog converter (DAC). The mux circuit is designed to operate across the full range of the supply voltage.

The 6-bit DAC is 64-tap resistor ladder network which provides a selectable voltage reference for applications where voltage reference is needed. The 64-tap resistor ladder network divides the supply reference V_{in} into 64 voltage levels. A 6-bit digital signal input selects the output voltage level, which varies from V_{in} to $V_{in}/64$. V_{in} can be selected from two voltage sources, V_{in1} and V_{in2}. The 6-bit DAC from a comparator is available as an on-chip internal signal only and is not available externally to a pin.

32.2 CMP features

The CMP has the following features:

- Operational over the entire supply range
- Inputs may range from rail to rail
- Programmable hysteresis control

- Selectable interrupt on rising-edge, falling-edge, or both rising or falling edges of the comparator output
- Selectable inversion on comparator output
- Capability to produce a wide range of outputs such as:
  - Sampled
  - Windowed, which is ideal for certain PWM zero-crossing-detection applications
  - Digitally filtered:
    - Filter can be bypassed
    - Can be clocked via external SAMPLE signal or scaled bus clock
- External hysteresis can be used at the same time that the output filter is used for internal functions
- Two software selectable performance levels:
  - Shorter propagation delay at the expense of higher power
  - Low power, with longer propagation delay
- DMA transfer support
  - A comparison event can be selected to trigger a DMA transfer
- Functional in all modes of operation
- The window and filter functions are not available in the following modes:
  - Stop
  - VLPS
  - LLS
  - VLLSx

32.3 6-bit DAC key features

- 6-bit resolution
- Selectable supply reference source
- Power Down mode to conserve power when not in use
- Option to route the output to internal comparator input

32.4 ANMUX key features

- Two 8-to-1 channel mux
- Operational over the entire supply range

32.5 CMP, DAC and ANMUX diagram

The following figure shows the block diagram for the High-Speed Comparator, DAC, and ANMUX modules.

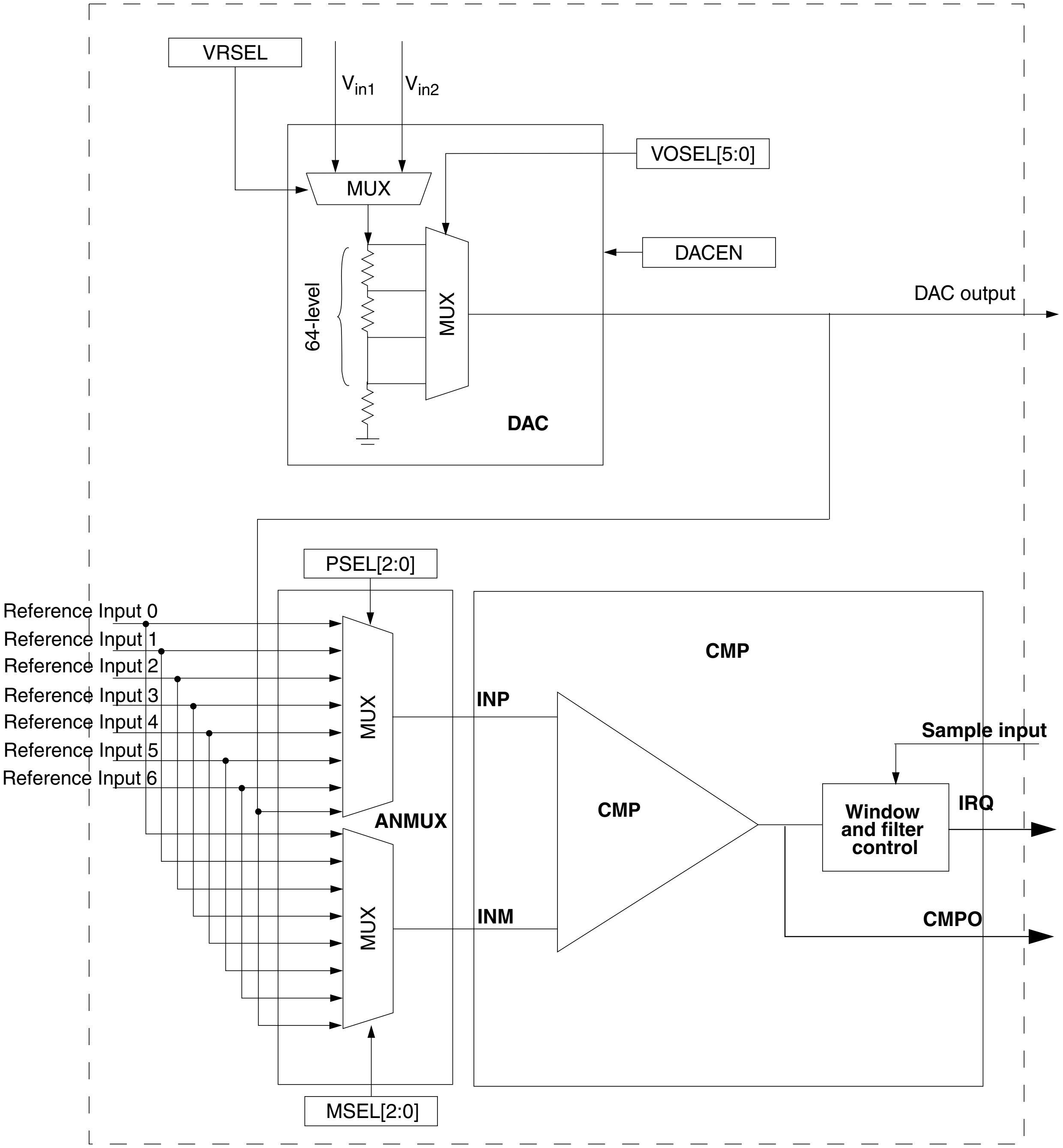

VRSEL
Vin1
Vin2
VOSEL[5:0]
MUX
DACEN
64-level
MUX
DAC output
DAC
PSEL[2:0]
Reference Input 0
Reference Input 1
Reference Input 2
Reference Input 3
Reference Input 4
Reference Input 5
Reference Input 6
MUX
CMP
INP
Sample input
IRQ
ANMUX
CMP
Window and filter control
INM
CMPO
MUX
MSEL[2:0]

Figure 32-1. CMP, DAC and ANMUX block diagram

32.6 CMP block diagram

The following figure shows the block diagram for the CMP module.

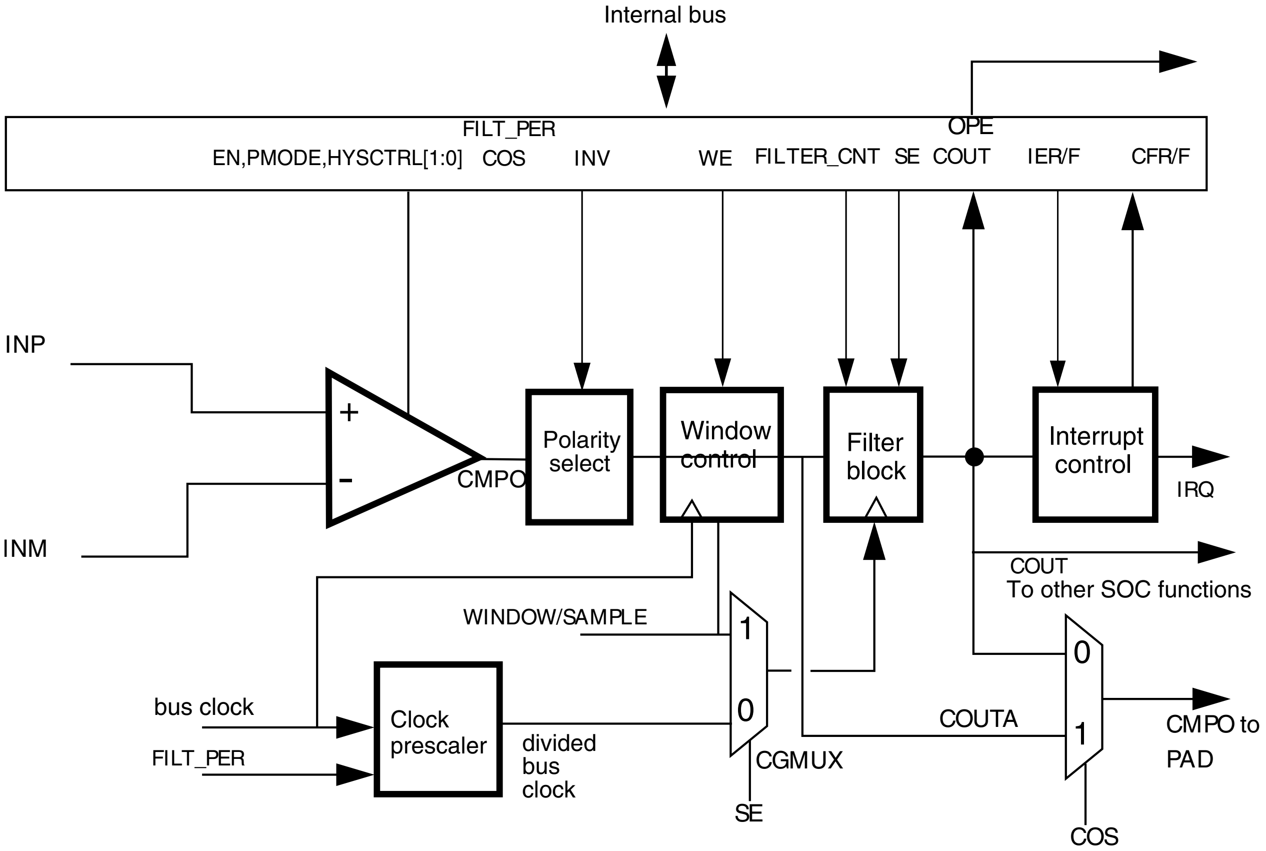

Internal bus
FILT_PER
EN,PMODE,HYSCTRL[1:0]
COS
INV
WE
FILTER_CNT
SE
OPE
COUT
IER/F
CFR/F
INP
INM
+
-
CMPO
Polarity select
Window control
Filter block
Interrupt control
IRQ
COUT
To other SOC functions
WINDOW/SAMPLE
1
0
bus clock
FILT_PER
Clock prescaler
divided bus clock
CGMUX
SE
COUTA
0
1
CMPO to PAD
COS

Figure 32-2. Comparator module block diagram

In the CMP block diagram:

- The Window Control block is bypassed when CR1[WE] = 0
- If CR1[WE] = 1, the comparator output will be sampled on every bus clock when WINDOW=1 to generate COUTA. Sampling does NOT occur when WINDOW = 0.
- The Filter block is bypassed when not in use.
- The Filter block acts as a simple sampler if the filter is bypassed and CR0[FILTER_CNT] is set to 0x01.
- The Filter block filters based on multiple samples when the filter is bypassed and CR0[FILTER_CNT] is set greater than 0x01.
  - If CR1[SE] = 1, the external SAMPLE input is used as sampling clock
  - IF CR1[SE] = 0, the divided bus clock is used as sampling clock

- If enabled, the Filter block will incur up to one bus clock additional latency penalty on COUT due to the fact that COUT, which is crossing clock domain boundaries, must be resynchronized to the bus clock.
- CR1[WE] and CR1[SE] are mutually exclusive.

32.7 Memory map/register definitions

CMP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_3000	CMP Control Register 0 (CMP0_CR0)	8	R/W	00h	32.7.1/ 639
4007_3001	CMP Control Register 1 (CMP0_CR1)	8	R/W	00h	32.7.2/ 640
4007_3002	CMP Filter Period Register (CMP0_FPR)	8	R/W	00h	32.7.3/ 641
4007_3003	CMP Status and Control Register (CMP0_SCR)	8	R/W	00h	32.7.4/ 641
4007_3004	DAC Control Register (CMP0_DACCR)	8	R/W	00h	32.7.5/ 643
4007_3005	MUX Control Register (CMP0_MUXCR)	8	R/W	00h	32.7.6/ 643
4007_3008	CMP Control Register 0 (CMP1_CR0)	8	R/W	00h	32.7.1/ 639
4007_3009	CMP Control Register 1 (CMP1_CR1)	8	R/W	00h	32.7.2/ 640
4007_300A	CMP Filter Period Register (CMP1_FPR)	8	R/W	00h	32.7.3/ 641
4007_300B	CMP Status and Control Register (CMP1_SCR)	8	R/W	00h	32.7.4/ 641
4007_300C	DAC Control Register (CMP1_DACCR)	8	R/W	00h	32.7.5/ 643
4007_300D	MUX Control Register (CMP1_MUXCR)	8	R/W	00h	32.7.6/ 643

32.7.1 CMP Control Register 0 (CMPx_CR0)

Addresses: CMP0_CR0 is 4007_3000h base + 0h offset = 4007_3000h
CMP1_CR0 is 4007_3008h base + 0h offset = 4007_3008h

Bit	7	6	5	4	3	2	1	0
Read	0	FILTER_CNT			0	0	HYSTCTR	
Write	▨				▨	▨		
Reset	0	0	0	0	0	0	0	0

CMPx_CR0 field descriptions

Field	Description
7 Reserved	This read-only field is reserved and always has the value zero.
6–4 FILTER_CNT	Filter Sample Count Represents the number of consecutive samples that must agree prior to the comparator ouput filter accepting a new output state. For information regarding filter programming and latency, see the CMP functional description. 000 Filter is disabled. If SE = 1, then COUT is a logic 0. This is not a legal state, and is not recommended. If SE = 0, COUT = COUTA. 001 One sample must agree. The comparator output is simply sampled. 010 2 consecutive samples must agree. 011 3 consecutive samples must agree. 100 4 consecutive samples must agree. 101 5 consecutive samples must agree. 110 6 consecutive samples must agree. 111 7 consecutive samples must agree.
3 Reserved	This read-only field is reserved and always has the value zero.
2 Reserved	This read-only field is reserved and always has the value zero.
1–0 HYSTCTR	Comparator hard block hysteresis control Defines the programmable hysteresis level. The hysteresis values associated with each level are device-specific. See the Data Sheet of the device for the exact values. 00 Level 0 01 Level 1 10 Level 2 11 Level 3

32.7.2 CMP Control Register 1 (CMPx_CR1)

Addresses: CMP0_CR1 is 4007_3000h base + 1h offset = 4007_3001h
CMP1_CR1 is 4007_3008h base + 1h offset = 4007_3009h

Bit	7	6	5	4	3	2	1	0
Read	SE	WE	0	PMODE	INV	COS	OPE	EN
Write			▨					
Reset	0	0	0	0	0	0	0	0

CMPx_CR1 field descriptions

Field	Description
7 SE	Sample Enable At any given time, either SE or WE can be set. If a write to this register attempts to set both, then SE is set and WE is cleared. However, avoid writing 1s to both field locations because this "11" case is reserved and may change in future implementations. 0 Sampling mode is not selected. 1 Sampling mode is selected.
6 WE	Windowing Enable At any given time, either SE or WE can be set. If a write to this register attempts to set both, then SE is set and WE is cleared. However, avoid writing 1s to both field locations because this "11" case is reserved and may change in future implementations. 0 Windowing mode is not selected. 1 Windowing mode is selected.
5 Reserved	This read-only field is reserved and always has the value zero.
4 PMODE	Power Mode Select See the electrical specifications table in the device Data Sheet for details. 0 Low-Speed (LS) Comparison mode selected. In this mode, CMP has slower output propagation delay and lower current consumption. 1 High-Speed (HS) Comparison mode selected. In this mode, CMP has faster output propagation delay and higher current consumption.
3 INV	Comparator INVERT Allows selection of the polarity of the analog comparator function. It is also driven to the COUT output, on both the device pin and as SCR[COUT], when OPE=0. 0 Does not invert the comparator output. 1 Inverts the comparator output.
2 COS	Comparator Output Select 0 Set the filtered comparator output (CMPO) to equal COUT. 1 Set the unfiltered comparator output (CMPO) to equal COUTA.

Table continues on the next page...

CMPx_CR1 field descriptions (continued)

Field	Description
1 OPE	Comparator Output Pin Enable 0 CMPO is not available on the associated CMPO output pin. 1 CMPO is available on the associated CMPO output pin.
0 EN	Comparator Module Enable Enables the Analog Comparator module. When the module is not enabled, it remains in the off state, and consumes no power. When the user selects the same input from analog mux to the positive and negative port, the comparator is disabled automatically. 0 Analog Comparator is disabled. 1 Analog Comparator is enabled.

32.7.3 CMP Filter Period Register (CMPx_FPR)

Addresses: CMP0_FPR is 4007_3000h base + 2h offset = 4007_3002h
CMP1_FPR is 4007_3008h base + 2h offset = 4007_300Ah

Bit	7	6	5	4	3	2	1	0
Read Write	FILT_PER							
Reset	0	0	0	0	0	0	0	0

CMPx_FPR field descriptions

Field	Description
7–0 FILT_PER	Filter Sample Period Specifies the sampling period, in bus clock cycles, of the comparator output filter, when CR1[SE]=0. Setting FILT_PER to 0x0 disables the filter. Filter programming and latency details appear in the CMP functional description. This field has no effect when CR1[SE]=1. In that case, the external SAMPLE signal is used to determine the sampling period.

32.7.4 CMP Status and Control Register (CMPx_SCR)

Addresses: CMP0_SCR is 4007_3000h base + 3h offset = 4007_3003h
CMP1_SCR is 4007_3008h base + 3h offset = 4007_300Bh

Bit	7	6	5	4	3	2	1	0
Read	0	DMAEN	0	IER	IEF	CFR	CFF	COUT
Write	▨		▨			w1c	w1c	▨
Reset	0	0	0	0	0	0	0	0

CMPx_SCR field descriptions

Field	Description
7 Reserved	This read-only field is reserved and always has the value zero.
6 DMAEN	DMA Enable Control Enables the DMA transfer triggered from the CMP module. When this field is set, a DMA request is asserted when CFR or CFF is set. 0 DMA is disabled. 1 DMA is enabled.
5 Reserved	This read-only field is reserved and always has the value zero.
4 IER	Comparator Interrupt Enable Rising Enables the CFR interrupt from the CMP. When this field is set, an interrupt will be asserted when CFR is set. 0 Interrupt is disabled. 1 Interrupt is enabled.
3 IEF	Comparator Interrupt Enable Falling Enables the CFF interrupt from the CMP. When this field is set, an interrupt will be asserted when CFF is set. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 CFR	Analog Comparator Flag Rising Detects a rising-edge on COUT, when set, during normal operation. CFR is cleared by writing 1 to it. During Stop modes, CFR is level sensitive . 0 Rising-edge on COUT has not been detected. 1 Rising-edge on COUT has occurred.
1 CFF	Analog Comparator Flag Falling Detects a falling-edge on COUT, when set, during normal operation. CFF is cleared by writing 1 to it. During Stop modes, CFF is level senstive . 0 Falling-edge on COUT has not been detected. 1 Falling-edge on COUT has occurred.
0 COUT	Analog Comparator Output Returns the current value of the Analog Comparator output, when read. The field is reset to 0 and will read as CR1[INV] when the Analog Comparator module is disabled, that is, when CR1[EN] = 0. Writes to this field are ignored.

32.7.5 DAC Control Register (CMPx_DACCR)

Addresses: CMP0_DACCR is 4007_3000h base + 4h offset = 4007_3004h
CMP1_DACCR is 4007_3008h base + 4h offset = 4007_300Ch

Bit	7	6	5	4	3	2	1	0
Read Write	DACEN	VRSEL	VOSEL					
Reset	0	0	0	0	0	0	0	0

CMPx_DACCR field descriptions

Field	Description
7 DACEN	DAC Enable Enables the DAC. When the DAC is disabled, it is powered down to conserve power. 0 DAC is disabled. 1 DAC is enabled.
6 VRSEL	Supply Voltage Reference Source Select 0 V is selected as resistor ladder network supply reference V. $_{in1in}$ 1 V is selected as resistor ladder network supply reference V. $_{in2in}$
5–0 VOSEL	DAC Output Voltage Select Selects an output voltage from one of 64 distinct levels. `DACO = (V` $_{in}$ `/64) * (VOSEL[5:0] + 1)` , so the DACO range is from V $_{in}$ /64 to V $_{in}$.

32.7.6 MUX Control Register (CMPx_MUXCR)

Addresses: CMP0_MUXCR is 4007_3000h base + 5h offset = 4007_3005h
CMP1_MUXCR is 4007_3008h base + 5h offset = 4007_300Dh

Bit	7	6	5	4	3	2	1	0
Read	0		PSEL			MSEL		
Write	▨	▨						
Reset	0	0	0	0	0	0	0	0

CMPx_MUXCR field descriptions

Field	Description
7–6 Reserved	This read-only field is reserved and always has the value zero.
5–3 PSEL	Plus Input Mux Control Determines which input is selected for the plus input of the comparator. For INx inputs, see CMP, DAC, and ANMUX block diagrams.

Table continues on the next page...

CMPx_MUXCR field descriptions (continued)

Field	Description
	NOTE: When an inappropriate operation selects the same input for both muxes, the comparator automatically shuts down to prevent itself from becoming a noise generator. 000 IN0 001 IN1 010 IN2 011 IN3 100 IN4 101 IN5 110 IN6 111 IN7
2–0 MSEL	Minus Input Mux Control Determines which input is selected for the minus input of the comparator. For INx inputs, see CMP, DAC, and ANMUX block diagrams. **NOTE:** When an inappropriate operation selects the same input for both muxes, the comparator automatically shuts down to prevent itself from becoming a noise generator. 000 IN0 001 IN1 010 IN2 011 IN3 100 IN4 101 IN5 110 IN6 111 IN7

32.8 CMP functional description

The CMP module can be used to compare two analog input voltages applied to INP and INM. CMPO is high when the non-inverting input is greater than the inverting input, and is low when the non-inverting input is less than the inverting input. This signal can be selectively inverted by setting CR1[INV] = 1.

SCR[IER] and SCR[IEF] are used to select the condition which will cause the CMP module to assert an interrupt to the processor. SCR[CFF] is set on a falling-edge and SCR[CFR] is set on rising-edge of the comparator output. The optionally filtered CMPO can be read directly through SCR[COUT].

32.8.1 CMP functional modes

There are three main sub-blocks to the CMP module:

- The comparator itself
- The window function
- The filter function

The filter, CR0[FILTER_CNT], can be clocked from an internal or external clock source. The filter is programmable with respect to the number of samples that must agree before a change in the output is registered. In the simplest case, only one sample must agree. In this case, the filter acts as a simple sampler.

The external sample input is enabled using CR1[SE]. When set, the output of the comparator is sampled only on rising edges of the sample input.

The "windowing mode" is enabled by setting CR1[WE]. When set, the comparator output is sampled only when WINDOW=1. This feature can be used to ignore the comparator output during time periods in which the input voltages are not valid. This is especially useful when implementing zero-crossing-detection for certain PWM applications.

The comparator filter and sampling features can be combined as shown in the following table. Individual modes are discussed below.

Table 32-22. Comparator sample/filter controls

Mode #	CR1[EN]	CR1[WE]	CR1[SE]	CR0[FILTER_CNT]	FPR[FILT_PER]	Operation
1	0	X	X	X	X	**Disabled** See the Disabled mode (# 1).
2A	1	0	0	0x00	X	**Continuous Mode** See the Continuous mode (#s 2A & 2B).
2B	1	0	0	X	0x00	
3A	1	0	1	0x01	X	**Sampled, Non-Filtered mode** See the Sampled, Non-Filtered mode (#s 3A & 3B).
3B	1	0	0	0x01	> 0x00	
4A	1	0	1	> 0x01	X	**Sampled, Filtered mode** See the Sampled, Filtered mode (#s 4A & 4B).
4B	1	0	0	> 0x01	> 0x00	
5A	1	1	0	0x00	X	**Windowed mode** Comparator output is sampled on every rising bus clock edge when SAMPLE=1 to generate COUTA. See the Windowed mode (#s 5A & 5B).
5B	1	1	0	X	0x00	

Table continues on the next page...

Table 32-22. Comparator sample/filter controls (continued)

Mode #	CR1[EN]	CR1[WE]	CR1[SE]	CR0[FILTER_CNT]	FPR[FILT_PER]	Operation
6	1	1	0	0x01	0x01–0xFF	**Windowed/Resampled mode** Comparator output is sampled on every rising bus clock edge when SAMPLE=1 to generate COUTA, which is then resampled on an interval determined by FILT_PER to generate COUT. See the Windowed/Resampled mode (# 6).
7	1	1	0	> 0x01	0x01–0xFF	**Windowed/Filtered mode** Comparator output is sampled on every rising bus clock edge when SAMPLE=1 to generate COUTA, which is then resampled and filtered to generate COUT. See the Windowed/Filtered mode (#7).
All other combinations of CR1[EN], CR1[WE], CR1[SE], CR0[FILTER_CNT], and FPR[FILT_PER] are illegal.						

For cases where a comparator is used to drive a fault input, for example, for a motor-control module such as FTM, it must be configured to operate in Continuous mode so that an external fault can immediately pass through the comparator to the target fault circuitry.

Note

> Filtering and sampling settings must be changed only after setting CR1[SE]=0 and CR0[FILTER_CNT]=0x00. This resets the filter to a known state.

32.8.1.1 Disabled mode (# 1)

In Disabled mode, the analog comparator is non-functional and consumes no power. CMPO is 0 in this mode.

32.8.1.2 Continuous mode (#s 2A & 2B)

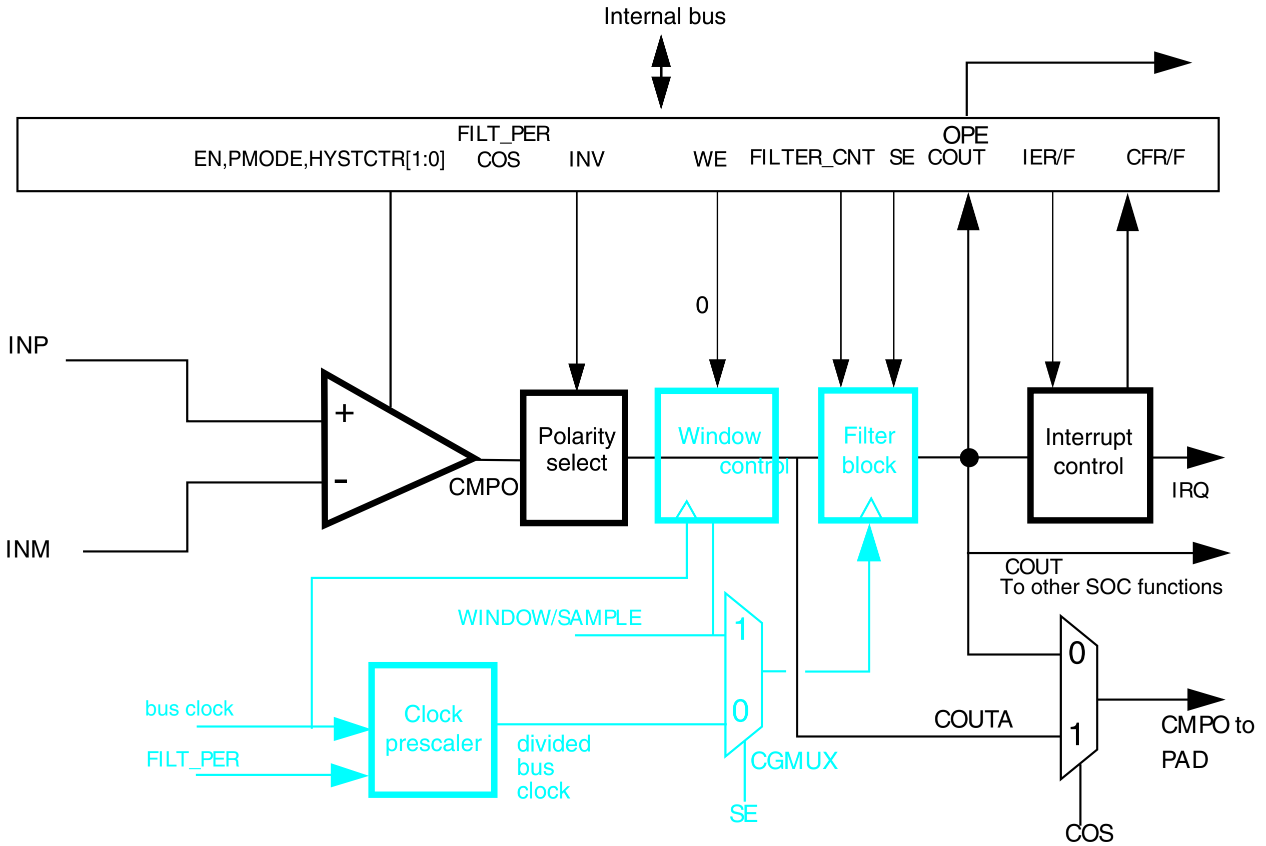

Internal bus
FILT_PER
EN,PMODE,HYSTCTR[1:0]
COS
INV
WE
FILTER_CNT
SE
OPE
COUT
IER/F
CFR/F
0
INP
INM
+
-
CMPO
Polarity select
Window control
Filter block
Interrupt control
IRQ
COUT
To other SOC functions
WINDOW/SAMPLE
1
0
bus clock
FILT_PER
Clock prescaler
divided bus clock
CGMUX
SE
COUTA
0
1
CMPO to PAD
COS

Figure 32-21. Comparator operation in Continuous mode

NOTE

See the chip configuration section for the source of sample/ window input.

The analog comparator block is powered and active. CMPO may be optionally inverted, but is not subject to external sampling or filtering. Both window control and filter blocks are completely bypassed. SCR[COUT] is updated continuously. The path from comparator input pins to output pin is operating in combinational unclocked mode. COUT and COUTA are identical.

For control configurations which result in disabling the filter block, see the Filter Block Bypass Logic diagram.

32.8.1.3 Sampled, Non-Filtered mode (#s 3A & 3B)

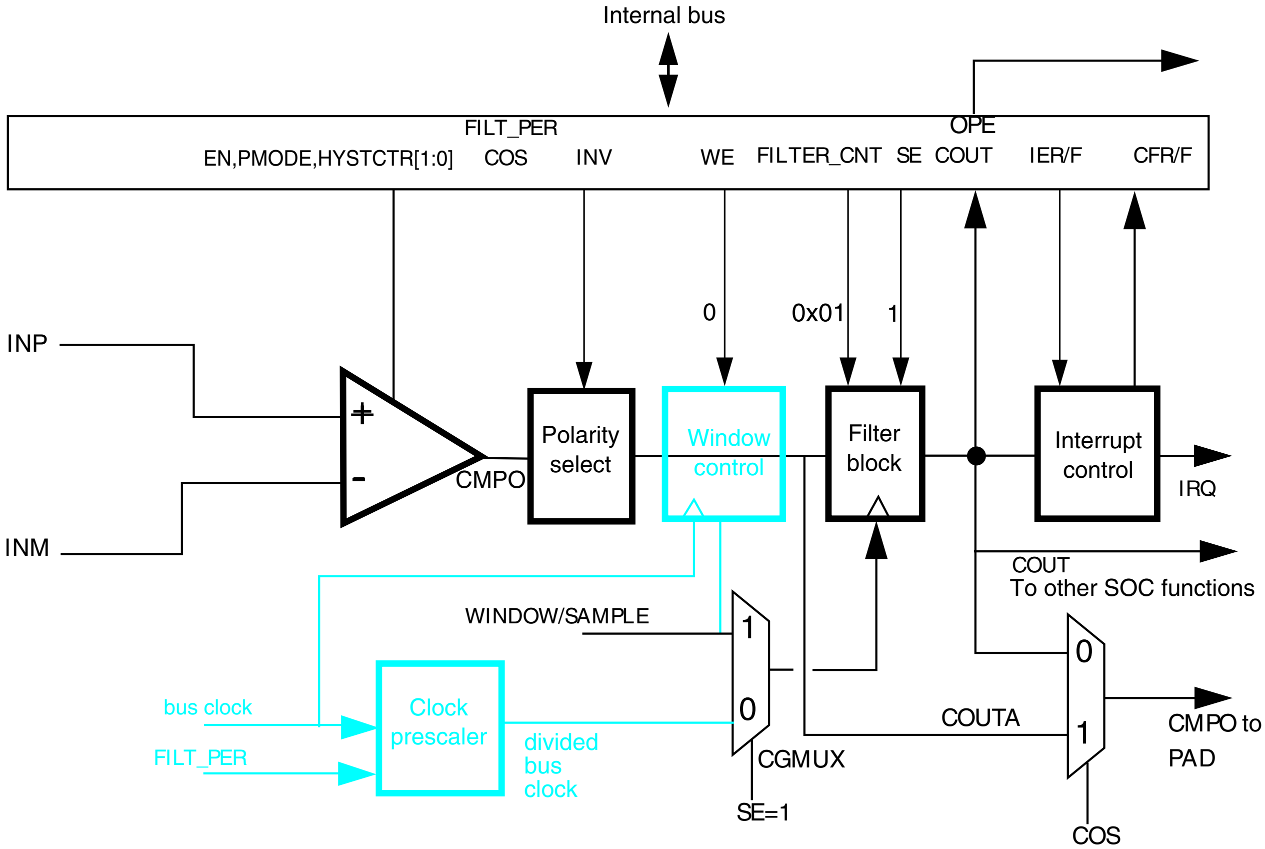

Internal bus
FILT_PER
EN,PMODE,HYSTCTR[1:0]
COS
INV
WE
FILTER_CNT
SE
OPE
COUT
IER/F
CFR/F
0
0x01
1
INP
INM
+
-
CMPO
Polarity select
Window control
Filter block
Interrupt control
IRQ
COUT
To other SOC functions
WINDOW/SAMPLE
1
0
CGMUX
SE=1
bus clock
FILT_PER
Clock prescaler
divided bus clock
COUTA
0
1
CMPO to PAD
COS

Figure 32-22. Sampled, Non-Filtered (# 3A): sampling point externally driven

In Sampled, Non-Filtered mode, the analog comparator block is powered and active. The path from analog inputs to COUTA is combinational unclocked. Windowing control is completely bypassed. COUTA is sampled whenever a rising-edge is detected on the filter block clock input.

The only difference in operation between Sampled, Non-Filtered (# 3A) and Sampled, Non-Filtered (# 3B) is in how the clock to the filter block is derived. In #3A, the clock to filter block is externally derived while in #3B, the clock to filter block is internally derived.

The comparator filter has no other function than sample/hold of the comparator output in this mode (# 3B).

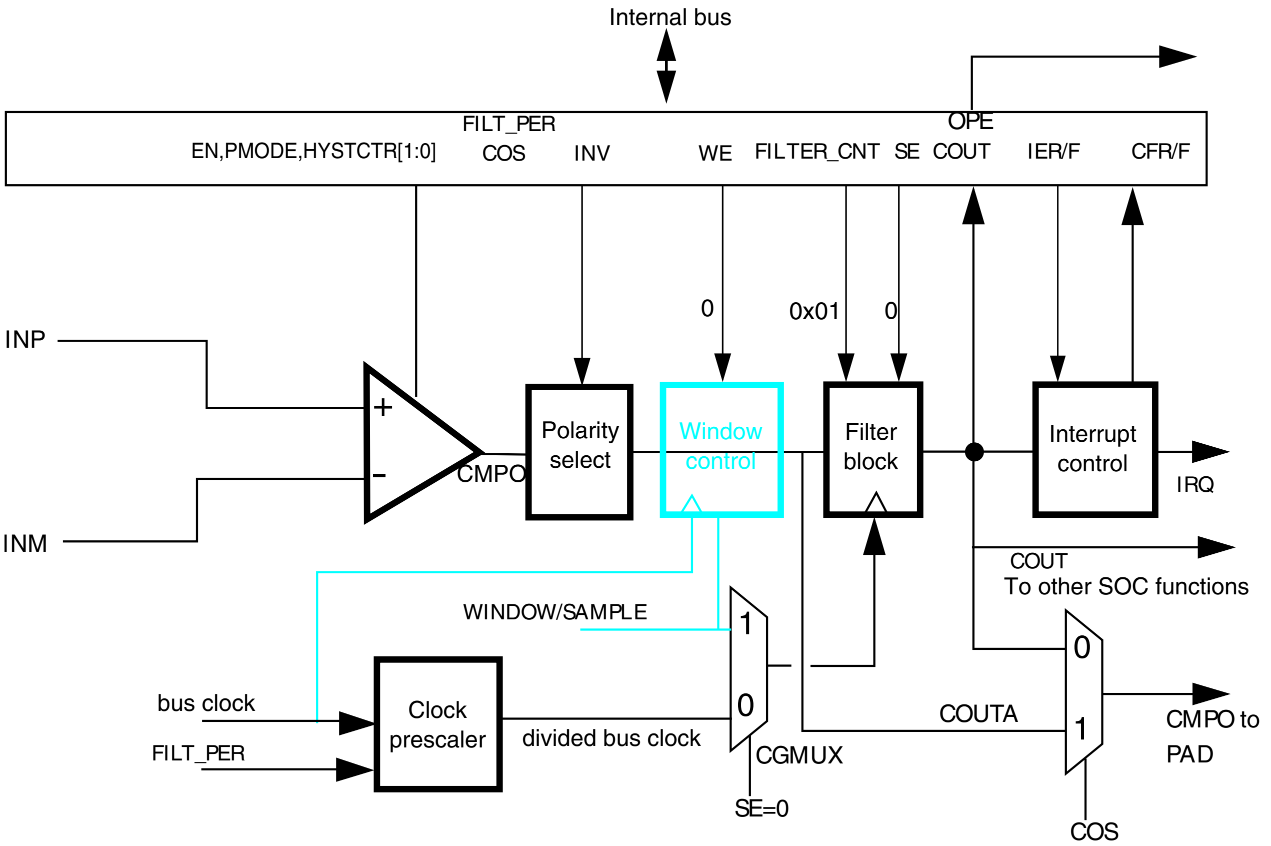

Internal bus
FILT_PER
EN,PMODE,HYSTCTR[1:0]
COS
INV
WE
FILTER_CNT
SE
OPE
COUT
IER/F
CFR/F
0
0x01
0
INP
INM
+
-
CMPO
Polarity select
Window control
Filter block
Interrupt control
IRQ
COUT
To other SOC functions
WINDOW/SAMPLE
1
0
bus clock
FILT_PER
Clock prescaler
divided bus clock
CGMUX
SE=0
COUTA
0
1
CMPO to PAD
COS

Figure 32-23. Sampled, Non-Filtered (# 3B): sampling interval internally derived

32.8.1.4 Sampled, Filtered mode (#s 4A & 4B)

In Sampled, Filtered mode, the analog comparator block is powered and active. The path from analog inputs to COUTA is combinational unclocked. Windowing control is completely bypassed. COUTA is sampled whenever a rising edge is detected on the filter block clock input.

The only difference in operation between Sampled, Non-Filtered (# 3A) and Sampled, Filtered (# 4A) is that, now, CR0[FILTER_CNT]>1, which activates filter operation.

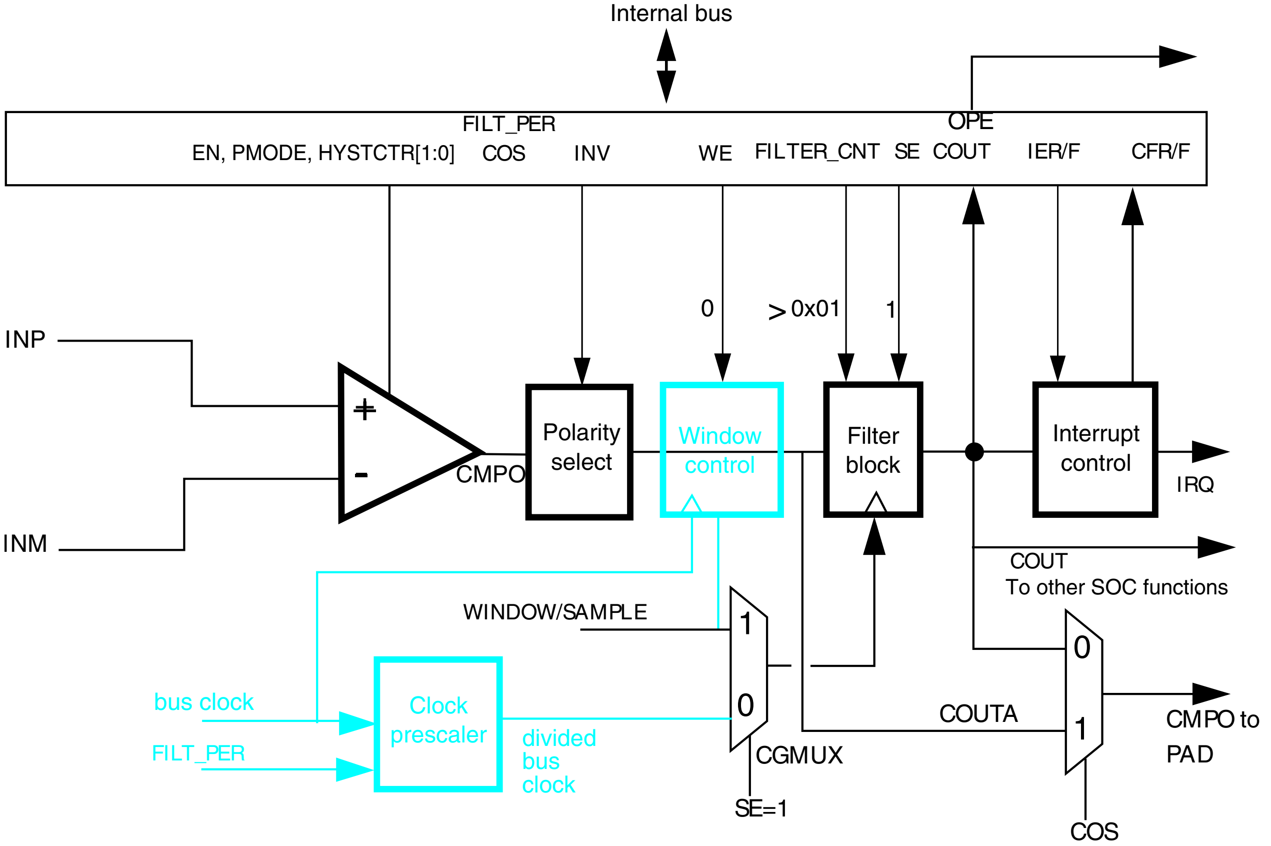

Internal bus
FILT_PER
EN, PMODE, HYSTCTR[1:0]
COS
INV
WE
FILTER_CNT
SE
OPE
COUT
IER/F
CFR/F
0
> 0x01
1
INP
INM
+
-
CMPO
Polarity select
Window control
Filter block
Interrupt control
IRQ
COUT
To other SOC functions
WINDOW/SAMPLE
1
0
CGMUX
SE=1
bus clock
FILT_PER
Clock prescaler
divided bus clock
COUTA
0
1
CMPO to PAD
COS

Figure 32-24. Sampled, Filtered (# 4A): sampling point externally driven

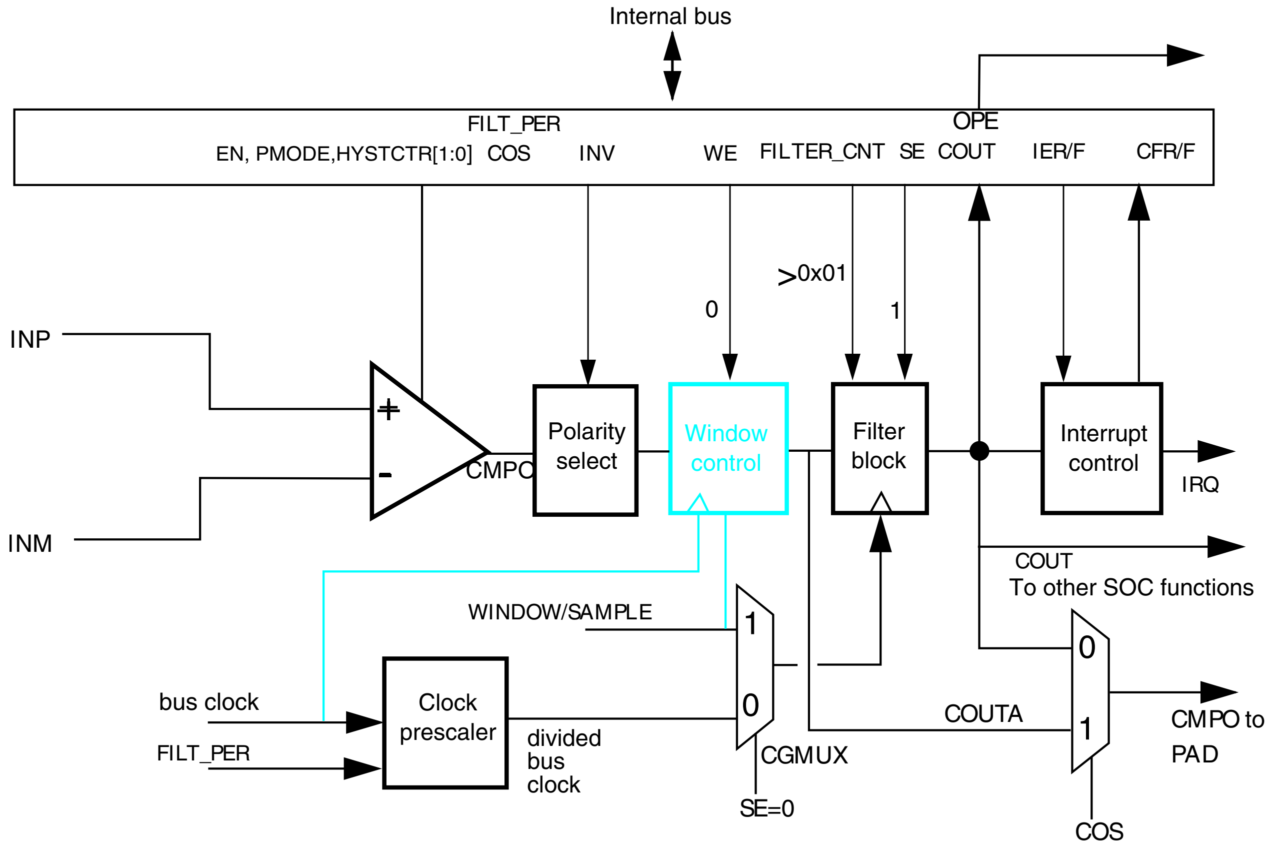

Internal bus
FILT_PER
EN, PMODE,HYSTCTR[1:0]
COS
INV
WE
FILTER_CNT
SE
OPE
COUT
IER/F
CFR/F
>0x01
0
1
INP
INM
+
-
CMPO
Polarity select
Window control
Filter block
Interrupt control
IRQ
COUT
To other SOC functions
WINDOW/SAMPLE
1
0
bus clock
FILT_PER
Clock prescaler
divided bus clock
CGMUX
SE=0
COUTA
0
1
CMPO to PAD
COS

Figure 32-25. Sampled, Filtered (# 4B): sampling point internally derived

The only difference in operation between Sampled, Non-Filtered (# 3B) and Sampled, Filtered (# 4B) is that now, CR0[FILTER_CNT]>1, which activates filter operation.

32.8.1.5 Windowed mode (#s 5A & 5B)

The following figure illustrates comparator operation in the Windowed mode, ignoring latency of the analog comparator, polarity select, and window control block. It also assumes that the polarity select is set to non-inverting state.

NOTE

> The analog comparator output is passed to COUTA only when the WINDOW signal is high.

In actual operation, COUTA may lag the analog inputs by up to one bus clock cycle plus the combinational path delay through the comparator and polarity select logic.

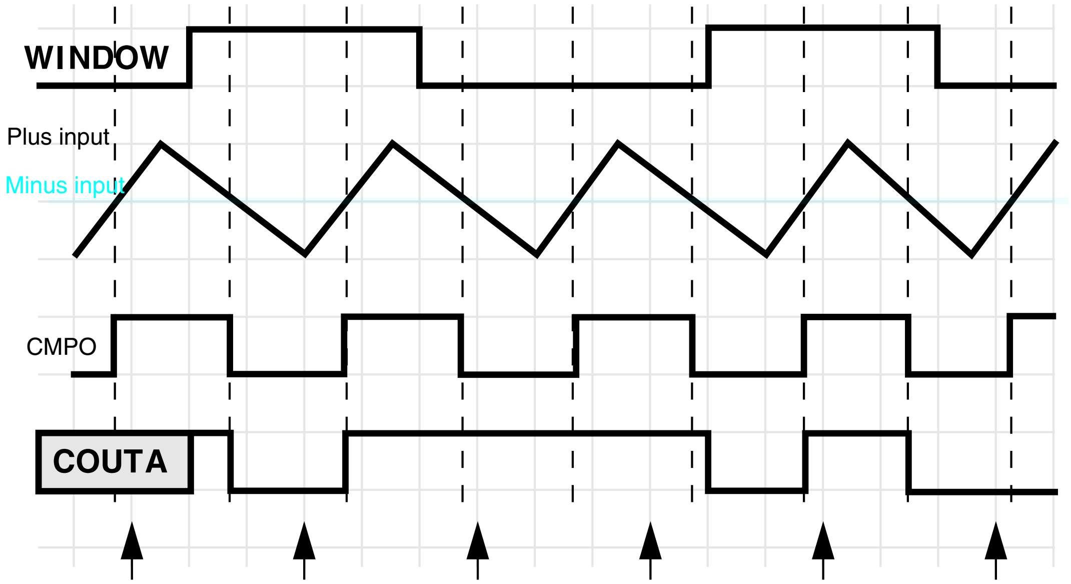


Figure 32-26. Windowed mode operation

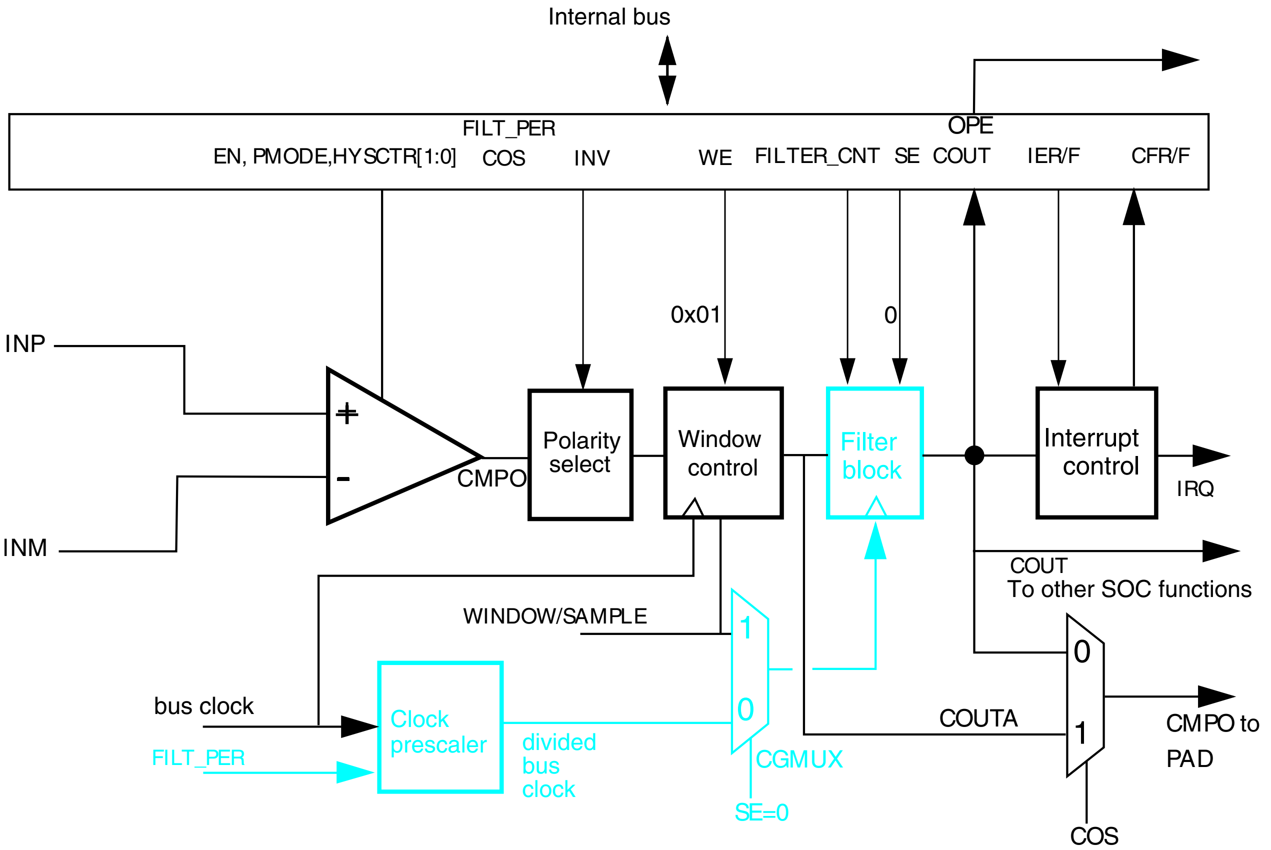

Internal bus
EN, PMODE,HYSCTR[1:0]
FILT_PER
COS
INV
WE
FILTER_CNT
SE
OPE
COUT
IER/F
CFR/F
0x01
0
INP
INM
+
-
CMPO
Polarity select
Window control
Filter block
Interrupt control
IRQ
COUT
To other SOC functions
WINDOW/SAMPLE
1
0
CGMUX
SE=0
bus clock
FILT_PER
Clock prescaler
divided bus clock
COUTA
0
1
CMPO to PAD
COS

Figure 32-27. Windowed mode

For control configurations which result in disabling the filter block, see Filter Block Bypass Logic diagram.

When any windowed mode is active, COUTA is clocked by the bus clock whenever WINDOW = 1. The last latched value is held when WINDOW = 0.

32.8.1.6 Windowed/Resampled mode (# 6)

The following figure uses the same input stimulus shown in Figure 32-26, and adds resampling of COUTA to generate COUT. Samples are taken at the time points indicated by the arrows in the figure. Again, prop delays and latency are ignored for the sake of clarity.

This example was generated solely to demonstrate operation of the comparator in windowed/resampled mode, and does not reflect any specific application. Depending upon the sampling rate and window placement, COUT may not see zero-crossing events detected by the analog comparator. Sampling period and/or window placement must be carefully considered for a given application.

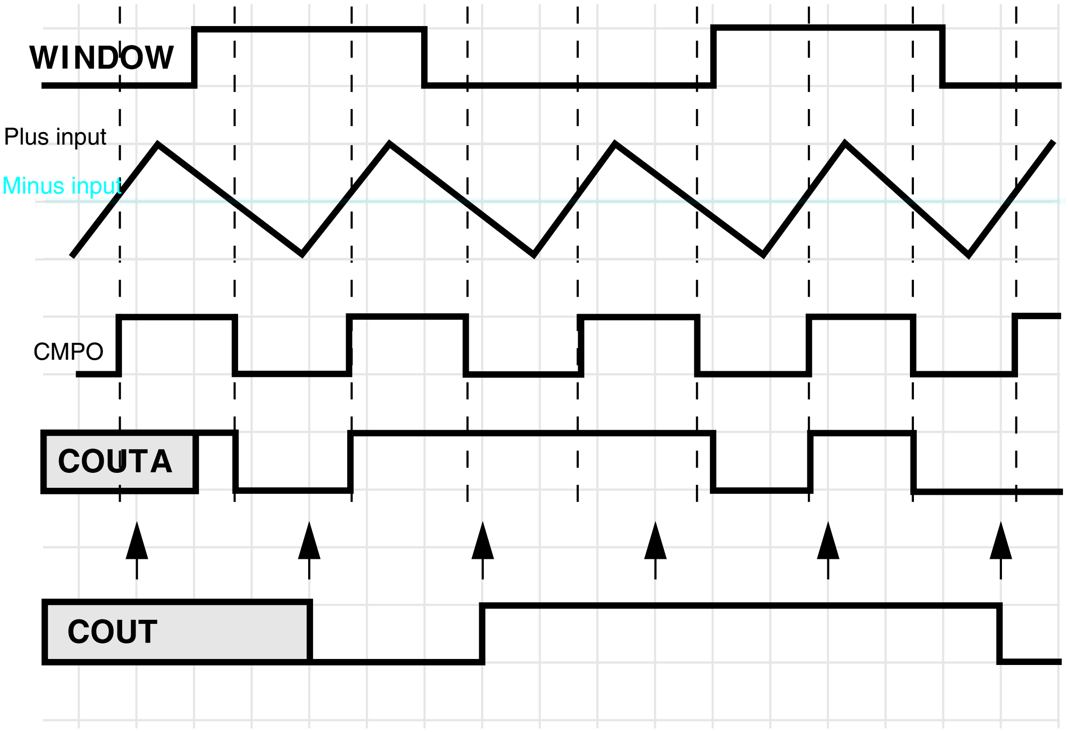


Figure 32-28. Windowed/resampled mode operation

This mode of operation results in an unfiltered string of comparator samples where the interval between the samples is determined by FPR[FILT_PER] and the bus clock rate. Configuration for this mode is virtually identical to that for the Windowed/Filtered Mode shown in the next section. The only difference is that the value of CR0[FILTER_CNT] must be 1.

32.8.1.7 Windowed/Filtered mode (#7)

This is the most complex mode of operation for the comparator block, as it uses both windowing and filtering features. It also has the highest latency of any of the modes. This can be approximated: up to 1 bus clock synchronization in the window function + ((CR0[FILTER_CNT] * FPR[FILT_PER]) + 1) * bus clock for the filter function.

When any windowed mode is active, COUTA is clocked by the bus clock whenever WINDOW = 1. The last latched value is held when WINDOW = 0.

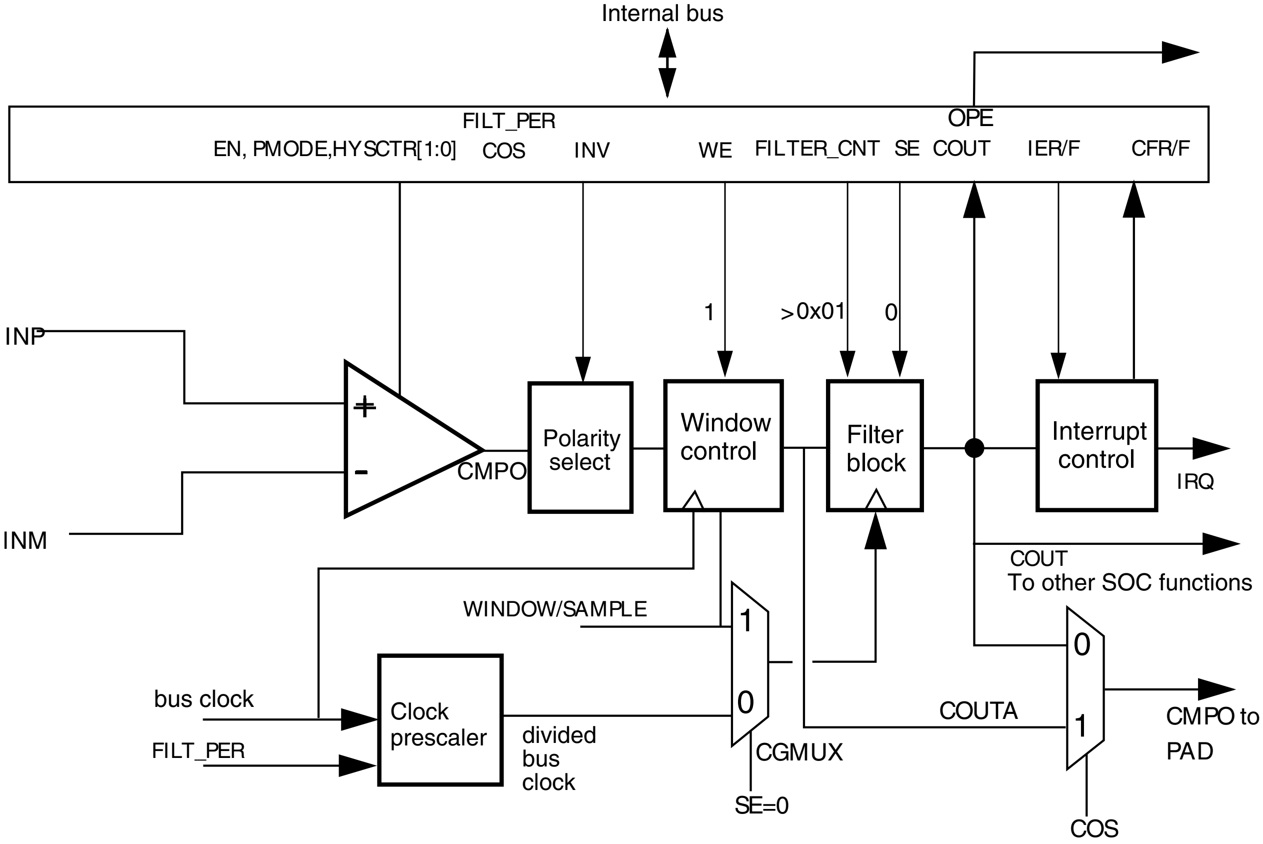

Internal bus
FILT_PER
EN, PMODE,HYSCTR[1:0]
COS
INV
WE
FILTER_CNT
SE
OPE
COUT
IER/F
CFR/F
1
>0x01
0
INP
INM
+
-
CMPO
Polarity select
Window control
Filter block
Interrupt control
IRQ
COUT
To other SOC functions
WINDOW/SAMPLE
1
0
CGMUX
SE=0
bus clock
FILT_PER
Clock prescaler
divided bus clock
COUTA
0
1
CMPO to PAD
COS

Figure 32-29. Windowed/Filtered mode

32.8.2 Power modes

32.8.2.1 Wait mode operation

During Wait and VLPW modes, the CMP, if enabled, continues to operate normally and a CMP interrupt can wake the MCU.

32.8.2.2 Stop mode operation

Subject to platform-specific clock restrictions, the MCU is brought out of stop when a compare event occurs and the corresponding interrupt is enabled. Similarly, if CR1[OPE] is enabled, the comparator output operates as in the normal operating mode and comparator output is placed onto the external pin. In Stop modes, the comparator can be operational in both:

- High-Speed (HS) Comparison mode when CR1[PMODE] = 1
- Low-Speed (LS) Comparison mode when CR1[PMODE] = 0

It is recommended to use the LS mode to minimize power consumption.

If stop is exited with a reset, all comparator registers are put into their reset state.

32.8.2.3 Low-Leakage mode operation

When the chip is in Low-Leakage modes:

- The CMP module is partially functional and is limited to Low-Speed mode, regardless of CR1[PMODE] setting
- Windowed, Sampled, and Filtered modes are not supported
- The CMP output pin is latched and does not reflect the compare output state.

The positive- and negative-input voltage can be supplied from external pins or the DAC output. The MCU can be brought out of the Low-Leakage mode if a compare event occurs and the CMP interrupt is enabled. After wakeup from low-leakage modes, the CMP module is in the reset state except for SCR[CFF] and SCR[CFR].

32.8.3 Startup and operation

A typical startup sequence is as follows.

The time required to stabilize COUT will be the power-on delay of the comparators plus the largest propagation delay from a selected analog source through the analog comparator, windowing function and filter. See the Data Sheets for power-on delays of the comparators. The windowing function has a maximum of one bus clock period delay. The filter delay is specified in the Low-pass filter.

During operation, the propagation delay of the selected data paths must always be considered. It may take many bus clock cycles for COUT and SCR[CFR]/SCR[CFF] to reflect an input change or a configuration change to one of the components involved in the data path.

When programmed for filtering modes, COUT will initially be equal to 0, until sufficient clock cycles have elapsed to fill all stages of the filter. This occurs even if COUTA is at a logic 1.

32.8.4 Low-pass filter

The low-pass filter operates on the unfiltered and unsynchronized and optionally inverted comparator output COUTA and generates the filtered and synchronized output COUT. Both COUTA and COUT can be configured as module outputs and are used for different purposes within the system.

Synchronization and edge detection are always used to determine status register bit values. They also apply to COUT for all sampling and windowed modes. Filtering can be performed using an internal timebase defined by FPR[FILT_PER], or using an external SAMPLE input to determine sample time.

The need for digital filtering and the amount of filtering is dependent on user requirements. Filtering can become more useful in the absence of an external hysteresis circuit. Without external hysteresis, high-frequency oscillations can be generated at COUTA when the selected INM and INP input voltages differ by less than the offset voltage of the differential comparator.

32.8.4.1 Enabling filter modes

Filter modes can be enabled by:

- Setting CR0[FILTER_CNT] > 0x01 and
- Setting FPR[FILT_PER] to a nonzero value or setting CR1[SE]=1

If using the divided bus clock to drive the filter, it will take samples of COUTA every FPR[FILT_PER] bus clock cycles.

The filter output will be at logic 0 when first initalized, and will subsequently change when all the consecutive CR0[FILTER_CNT] samples agree that the output value has changed. In other words, SCR[COUT] will be 0 for some initial period, even when COUTA is at logic 1.

Setting both CR1[SE] and FPR[FILT_PER] to 0 disables the filter and eliminates switching current associated with the filtering process.

Note

> Always switch to this setting prior to making any changes in filter parameters. This resets the filter to a known state. Switching CR0[FILTER_CNT] on the fly without this intermediate step can result in unexpected behavior.

If CR1[SE]=1, the filter takes samples of COUTA on each positive transition of the sample input. The output state of the filter changes when all the consecutive CR0[FILTER_CNT] samples agree that the output value has changed.

32.8.4.2 Latency issues

The value of FPR[FILT_PER] or SAMPLE period must be set such that the sampling period is just longer than the period of the expected noise. This way a noise spike will corrupt only one sample. The value of CR0[FILTER_CNT] must be chosen to reduce the probability of noisy samples causing an incorrect transition to be recognized. The probability of an incorrect transition is defined as the probability of an incorrect sample raised to the power of CR0[FILTER_CNT].

The values of FPR[FILT_PER] or SAMPLE period and CR0[FILTER_CNT] must also be traded off against the desire for minimal latency in recognizing actual comparator output transitions. The probability of detecting an actual output change within the nominal latency is the probability of a correct sample raised to the power of CR0[FILTER_CNT].

The following table summarizes maximum latency values for the various modes of operation *in the absence of noise*. Filtering latency is restarted each time an actual output transition is masked by noise.

Table 32-23. Comparator sample/filter maximum latencies

Mode #	CR1[EN]	CR1[WE]	CR1[SE]	CR0[FILTER_CNT]	FPR[FILT_PER]	Operation	Maximum latency[1]
1	0	X	X	X	X	Disabled	N/A
2A	1	0	0	0x00	X	Continuous Mode	T_{PD}
2B	1	0	0	X	0x00		
3A	1	0	1	0x01	X	Sampled, Non-Filtered mode	$T_{PD} + T_{SAMPLE} + T_{per}$
3B	1	0	0	0x01	> 0x00		T_{PD} + (FPR[FILT_PER] * T_{per}) + T_{per}

Table continues on the next page...

Table 32-23. Comparator sample/filter maximum latencies (continued)

Mode #	CR1[EN]	CR1[WE]	CR1[SE]	CR0[FILTER_CNT]	FPR[FILT_PER]	Operation	Maximum latency[1]
4A	1	0	1	> 0x01	X	Sampled, Filtered mode	T_{PD} + (CR0[FILTER_CNT] * T_{SAMPLE}) + T_{per}
4B	1	0	0	> 0x01	> 0x00		T_{PD} + (CR0[FILTER_CNT] * FPR[FILT_PER] x T_{per}) + T_{per}
5A	1	1	0	0x00	X	Windowed mode	T_{PD} + T_{per}
5B	1	1	0	X	0x00		T_{PD} + T_{per}
6	1	1	0	0x01	0x01 - 0xFF	Windowed / Resampled mode	T_{PD} + (FPR[FILT_PER] * T_{per}) + 2T_{per}
7	1	1	0	> 0x01	0x01 - 0xFF	Windowed / Filtered mode	T_{PD} + (CR0[FILTER_CNT] * FPR[FILT_PER] x T_{per}) + 2T_{per}

1. T_{PD} represents the intrinsic delay of the analog component plus the polarity select logic. T_{SAMPLE} is the clock period of the external sample clock. T_{per} is the period of the bus clock.

32.9 CMP interrupts

The CMP module is capable of generating an interrupt on either the rising- or falling-edge of the comparator output, or both. The following table gives the conditions in which the interrupt request is asserted and deasserted.

When	Then
SCR[IER] and SCR[CFR] are set	The interrupt request is asserted
SCR[IEF] and SCR[CFF] are set	The interrupt request is asserted
SCR[IER] and SCR[CFR] are cleared for a rising-edge interrupt	The interrupt request is deasserted
SCR[IEF] and SCR[CFF] are cleared for a falling-edge interrupt	The interrupt request is deasserted

32.10 CMP DMA support

Normally, the CMP generates a CPU interrupt if there is a change on the COUT. When DMA support is enabled by setting SCR[DMAEN] and the interrupt is enabled by setting SCR[IER], SCR[IEF], or both, the corresponding change on COUT forces a DMA transfer request rather than a CPU interrupt instead. When the DMA has completed the transfer, it sends a dma_done signal that deasserts the dma_request and clears the flag to allow a subsequent change on comparator output to occur and force another DMA request.

32.11 Digital-to-analog converter block diagram

The following figure shows the block diagram of the DAC module. It contains a 64-tap resistor ladder network and a 64-to-1 multiplexer, which selects an output voltage from one of 64 distinct levels that outputs from DACO. It is controlled through the DAC Control Register (DACCR). Its supply reference source can be selected from two sources V_{in1} and V_{in2}. The module can be powered down or disabled when not in use. When in Disabled mode, DACO is connected to the analog ground.

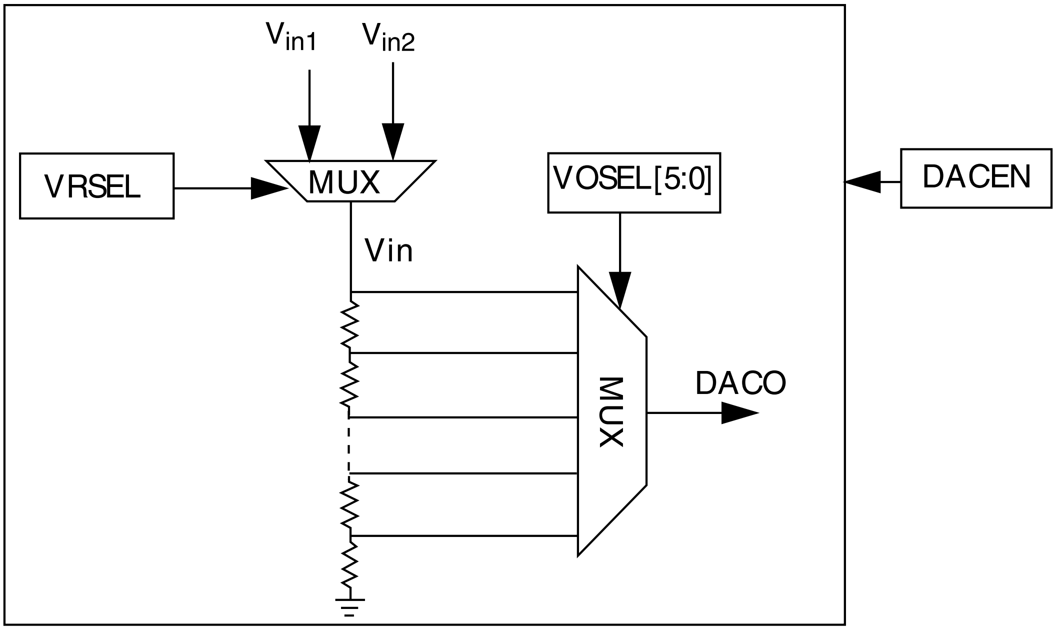


Figure 32-30. 6-bit DAC block diagram

32.12 DAC functional description

This section provides DAC functional description.

32.12.1 Voltage reference source select

- V_{in1} must be used to connect to the primary voltage source as supply reference of 64 tap resistor ladder
- V_{in2} must be used to connect to an alternate voltage source, or primary source, if an alternate voltage source is not available

32.13 DAC resets

This module has a single reset input, corresponding to the chip-wide peripheral reset.

32.14 DAC clocks

This module has a single clock input, the bus clock.

32.15 DAC interrupts

This module has no interrupts.

Chapter 33
Voltage Reference (VREFV1)

33.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The Voltage Reference(VREF) is intended to supply an accurate voltage output that can be trimmed in 0.5 mV steps. The VREF can be used in applications to provide a reference voltage to external devices or used internally as a reference to analog peripherals such as the ADC, DAC, or CMP. The voltage reference has three operating modes that provide different levels of supply rejection and power consumption..

The following figure is a block diagram of the Voltage Reference.

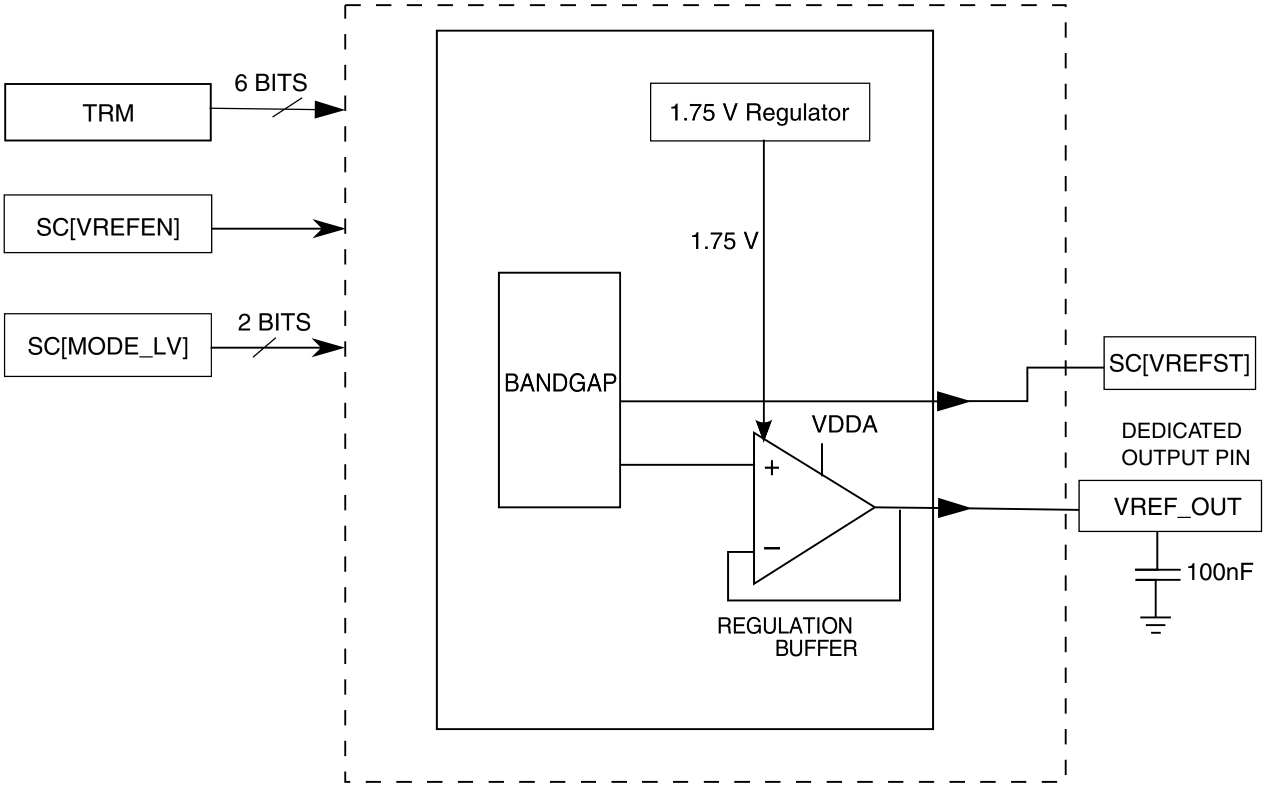

TRM
6 BITS
SC[VREFEN]
SC[MODE_LV]
2 BITS
1.75 V Regulator
1.75 V
BANDGAP
VDDA
+
−
REGULATION
BUFFER
SC[VREFST]
DEDICATED
OUTPUT PIN
VREF_OUT
100nF

Figure 33-1. Voltage reference block diagram

33.1.1 Overview

The Voltage Reference provides a buffered reference voltage for use as an external reference. In addition, the buffered reference is available internally for use with on chip peripherals such as ADCs and DACs. Refer to the chip configuration chapter for a description of these options. The reference voltage is output on a dedicated output pin when the VREF is enabled. The Voltage Reference output can be trimmed with a resolution of 0.5mV by means of the TRM register TRIM[5:0] bitfield.

33.1.2 Features

The Voltage Reference has the following features:

- Programmable trim register with 0.5 mV steps, automatically loaded with factory trimmed value upon reset
- Programmable buffer mode selection:
  - Off

  - Bandgap enabled/standby (output buffer disabled)
  - Low power buffer mode (output buffer enabled)
  - High power buffer mode (output buffer enabled)
- 1.2 V output at room temperature
- Dedicated output pin, VREF_OUT

33.1.3 Modes of Operation

The Voltage Reference continues normal operation in Run, Wait, and Stop modes. The Voltage Reference can also run in Very Low Power Run (VLPR), Very Low Power Wait (VLPW) and Very Low Power Stop (VLPS). If it is desired to use the VREF regulator in the very low power modes, the system reference voltage must be enabled in these modes. Refer to the chip configuration chapter for information on enabling this mode of operation. Having the VREF regulator enabled does increase current consumption. In very low power modes it may be desirable to disable the VREF regulator to minimize current consumption. Note however that the accuracy of the output voltage will be reduced (by as much as several mVs) when the VREF regulator is not used. .

NOTE

The assignment of module modes to core modes is chip-specific. For module-to-core mode assignments, see the chapter that describes how modules are configured.

33.1.4 VREF Signal Descriptions

The following table shows the Voltage Reference signals properties.

Table 33-1. VREF Signal Descriptions

Signal	Description	I/O
VREF_OUT	Internally-generated Voltage Reference output	O

NOTE

When the VREF output buffer is disabled, the status of the VREF_OUT signal is high-impedence.

33.2 Memory Map and Register Definition

VREF memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_4000	VREF Trim Register (VREF_TRM)	8	R/W	See section	33.2.1/ 664
4007_4001	VREF Status and Control Register (VREF_SC)	8	R/W	00h	33.2.2/ 665

33.2.1 VREF Trim Register (VREF_TRM)

This register contains bits that contain the trim data for the Voltage Reference.

Address: VREF_TRM is 4007_4000h base + 0h offset = 4007_4000h

Bit	7	6	5	4	3	2	1	0
Read Write	Reserved	CHOPEN	TRIM					
Reset	x*	0	x*	x*	x*	x*	x*	x*

* Notes:

- x = Undefined at reset.

VREF_TRM field descriptions

Field	Description
7 Reserved	This field is reserved.
6 CHOPEN	Chop oscillator enable. When set, internal chopping operation is enabled and the internal analog offset will be minimized. This bit is set during factory trimming of the VREF voltage. This bit should be written to 1 to achieve the performance stated in the data sheet. 0 Chop oscillator is disabled. 1 Chop oscillator is enabled.
5–0 TRIM	Trim bits These bits change the resulting VREF by approximately ± 0.5 mV for each step. **NOTE:** Min = minimum and max = maximum voltage reference output. For minimum and maximum voltage reference output values, refer to the Data Sheet for this chip.

Table continues on the next page...

VREF_TRM field descriptions (continued)

Field	Description
	000000 Min .... 111111 Max

33.2.2 VREF Status and Control Register (VREF_SC)

This register contains the control bits used to enable the internal voltage reference and to select the buffer mode to be used.

Address: VREF_SC is 4007_4000h base + 1h offset = 4007_4001h

Bit	7	6	5	4	3	2	1	0
Read	VREFEN	REGEN	ICOMPEN	0	0	VREFST	MODE_LV	
Write				▨	▨	▨		
Reset	0	0	0	0	0	0	0	0

VREF_SC field descriptions

Field	Description
7 VREFEN	Internal Voltage Reference enable This bit is used to enable the bandgap reference within the Voltage Reference module. **NOTE:** After the VREF is enabled, turning off the clock to the VREF module via the corresponding clock gate register will not disable the VREF. VREF must be disabled via this VREFEN bit. 0 The module is disabled. 1 The module is enabled.
6 REGEN	Regulator enable This bit is used to enable the internal 1.75 V regulator to produce a constant internal voltage supply in order to reduce the sensitivity to external supply noise and variation. If it is desired to keep the regulator enabled in very low power modes, refer to the Chip Configuration chapter for a description on how this can be achieved. This bit is set during factory trimming of the VREF voltage. This bit should be written to 1 to achieve the performance stated in the data sheet. 0 Internal 1.75 V regulator is disabled. 1 Internal 1.75 V regulator is enabled.
5 ICOMPEN	Second order curvature compensation enable This bit is set during factory trimming of the VREF voltage. This bit should be written to 1 to achieve the performance stated in the data sheet. 0 Disabled 1 Enabled

Table continues on the next page...

VREF_SC field descriptions (continued)

Field	Description
4 Reserved	This read-only field is reserved and always has the value zero.
3 Reserved	This read-only field is reserved and always has the value zero.
2 VREFST	Internal Voltage Reference stable This bit indicates that the bandgap reference within the Voltage Reference module has completed its startup and stabilization. 0 The module is disabled or not stable. 1 The module is stable.
1–0 MODE_LV	Buffer Mode selection These bits select the buffer modes for the Voltage Reference module. 00 Bandgap on only, for stabilization and startup 01 High power buffer mode enabled 10 Low-power buffer mode enabled 11 Reserved

33.3 Functional Description

The Voltage Reference is a bandgap buffer system. Unity gain amplifiers are used.

The VREF_OUT signal can be used by both internal and external peripherals in low and high power buffer mode. A 100 nF capacitor must always be connected between VREF_OUT and VSSA if the VREF is being used.

The following table shows all possible function configurations of the Voltage Reference.

Table 33-5. Voltage Reference function configurations

SC[VREFEN]	SC[MODE_LV]	Configuration	Functionality
0	X	Voltage Reference disabled	Off
1	00	Voltage Reference enabled, bandgap on only	Startup and standby
1	01	Voltage Reference enabled, high-power buffer on	VREF_OUT available for internal and external use. 100 nF capacitor is required.
1	10	Voltage Reference enabled, low power buffer on	VREF_OUT available for internal and external use. 100 nF capacitor is required.
1	11	Reserved	Reserved

33.3.1 Voltage Reference Disabled, SC[VREFEN] = 0

When SC[VREFEN] = 0, the Voltage Reference is disabled, the VREF bandgap and the output buffers are disabled. The Voltage Reference is in off mode.

33.3.2 Voltage Reference Enabled, SC[VREFEN] = 1

When SC[VREFEN] = 1, the Voltage Reference is enabled, and different modes should be set by the SC[MODE_LV] bits.

33.3.2.1 SC[MODE_LV]=00

The internal VREF bandgap is enabled to generate an accurate 1.2 V output that can be trimmed with the TRM register's TRIM[5:0] bitfield. The bandgap requires some time for startup and stabilization. SC[VREFST] can be monitored to determine if the stabilization and startup is complete.

The output buffer is disabled in this mode, and there is no buffered voltage output. The Voltage Reference is in standby mode. If this mode is first selected and the low power or high power buffer mode is subsequently enabled, there will be a delay before the buffer output is settled at the final value. This is the buffer start up delay (Tstup) and the value is specified in the appropriate device data sheet.

33.3.2.2 SC[MODE_LV] = 01

The internal VREF bandgap is on. The high power buffer is enabled to generate a buffered 1.2 V voltage to VREF_OUT. It can also be used as a reference to internal analog peripherals such as an ADC channel or analog comparator input.

If this mode is entered from the standby mode (SC[MODE_LV] = 00, SC[VREFEN] = 1) there will be a delay before the buffer output is settled at the final value. This is the buffer start up delay (Tstup) and the value is specified in the appropriate device data sheet. If this mode is entered when the VREF module is enabled then you must wait the longer of Tstup or until SC[VREFST] = 1.

In this mode, a 100 nF capacitor is required to connect between the VREF_OUT pin and VSSA.

33.3.2.3 SC[MODE_LV] = 10

The internal VREF bandgap is on. The low power buffer is enabled to generate a buffered 1.2 V voltage to VREF_OUT. It can also be used as a reference to internal analog peripherals such as an ADC channel or analog comparator input.

If this mode is entered from the standby mode (SC[MODE_LV] = 00, SC[VREFEN] = 1) there will be a delay before the buffer output is settled at the final value. This is the buffer start up delay (Tstup) and the value is specified in the appropriate device data sheet. If this mode is entered when the VREF module is enabled then you must wait the longer of Tstup or until SC[VREFST] = 1.

In this mode, a 100 nF capacitor is required to connect between the VREF_OUT pin and VSSA.

33.3.2.4 SC[MODE_LV] = 11

Reserved

33.4 Initialization/Application Information

The Voltage Reference requires some time for startup and stabilization. After SC[VREFEN] = 1, SC[VREFST] can be monitored to determine if the stabilization and startup is completed.

When the Voltage Reference is already enabled and stabilized, changing SC[MODE_LV] will not clear SC[VREFST] but there will be some startup time before the output voltage at the VREF_OUT pin has settled. This is the buffer start up delay (Tstup) and the value is specified in the appropriate device data sheet. Also, there will be some settling time when a step change of the load current is applied to the VREF_OUT pin. When the 1.75V VREF regulator is disabled, the VREF_OUT voltage will be more sensitive to supply voltage variation. It is recommended to use this regulator to achieve optimum VREF_OUT performance.

The TRM[CHOPEN], SC[REGEN] and SC[ICOMPEN] bits are written to 1 during factory trimming of the VREF voltage. These bits should be written to 1 to achieve the perfromance stated in the device data sheet.

Chapter 34
Programmable Delay Block (PDB)

34.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The programmable delay block (PDB) provides controllable delays from either an internal or an external trigger, or a programmable interval tick, to the hardware trigger inputs of ADCs and/or generates the interval triggers to DACs, so that the precise timing between ADC conversions and/or DAC updates can be achieved. The PDB can optionally provides pulse outputs (Pulse-Out's) that are used as the sample window in the CMP block.

34.1.1 Features

- Up to 15 trigger input sources and software trigger source
- Up to eight configurable PDB channels for ADC hardware trigger
    - One PDB channel is associated with one ADC.
    - One trigger output for ADC hardware trigger and up to eight pre-trigger outputs for ADC trigger select per PDB channel
    - Trigger outputs can be enabled or disabled independently.
    - One 16-bit delay register per pre-trigger output
    - Optional bypass of the delay registers of the pre-trigger outputs
    - Operation in One-Shot or Continuous modes

  - Optional back-to-back mode operation, which enables the ADC conversions complete to trigger the next PDB channel
  - One programmable delay interrupt
  - One sequence error interrupt
  - One channel flag and one sequence error flag per pre-trigger
  - DMA support
- Up to eight pulse outputs (pulse-out's)
  - Pulse-out's can be enabled or disabled independently.
  - Programmable pulse width

NOTE

The number of PDB input and output triggers are chip-specific. Refer to the Chip Configuration information for details.

34.1.2 Implementation

In this chapter, the following letters refers to the number of output triggers.

- N — Total available number of PDB channels.
- n — PDB channel number, valid from 0 to N-1.
- M — Total available pre-trigger per PDB channel.
- m — Pre-trigger number, valid from 0 to M-1.
- X — Total number of DAC interval triggers.
- x — DAC interval trigger output number, valid from 0 to X-1.
- Y — Total number of Pulse-Out's.
- y — Pulse-Out number, valid value is 0 to Y-1.

NOTE

The number of module output triggers to core are chip-specific. For module to core output triggers implementation, refer to the Chip Configuration information.

34.1.3 Back-to-back Acknowledgement Connections

PDB back-to-back operation acknowledgment connections are chip-specific. For implementation, refer to the Chip Configuration information.

34.1.4 Block Diagram

This diagram illustrates the major components of the PDB.

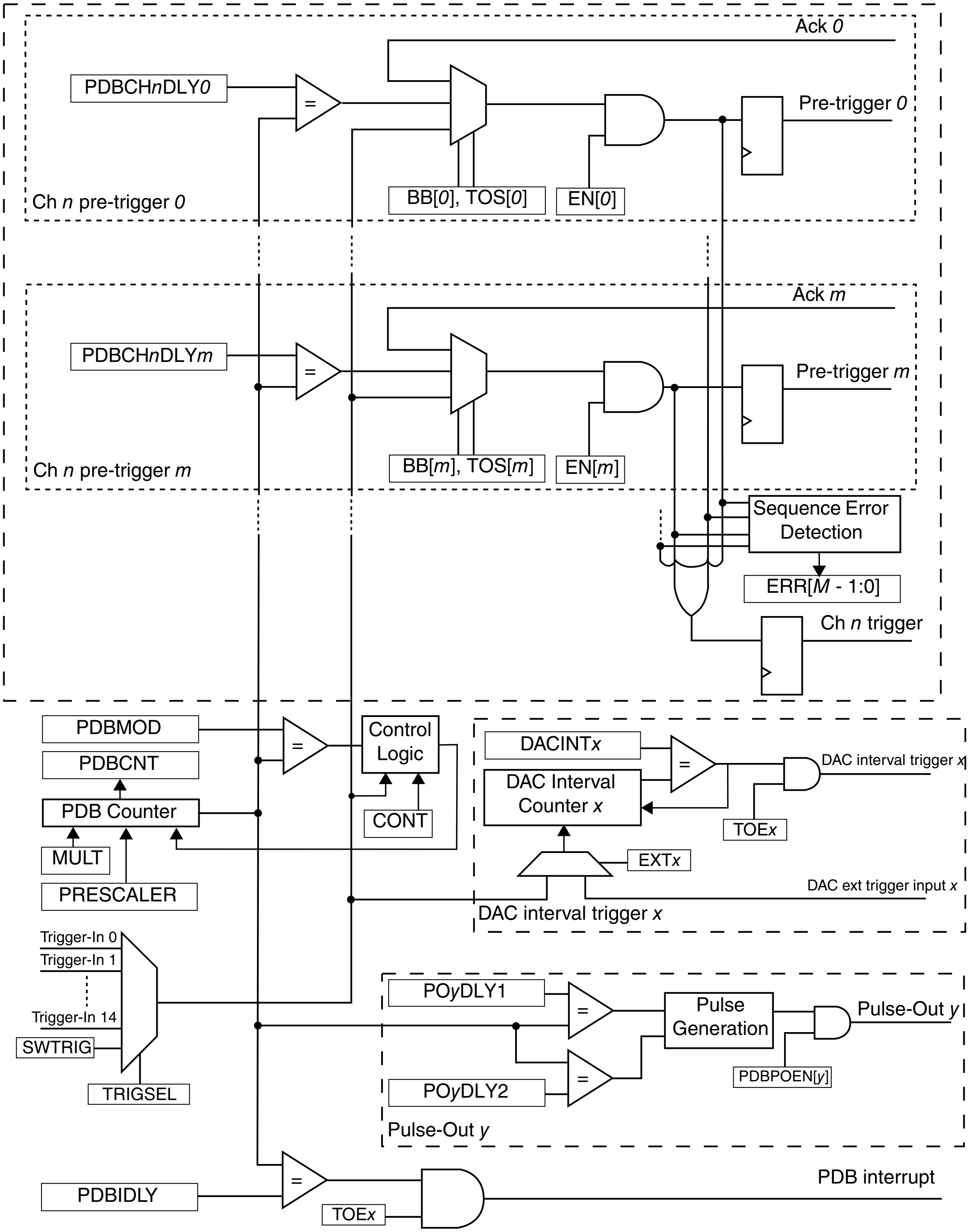

Ack 0
PDBCHnDLY0
=
Pre-trigger 0
Ch n pre-trigger 0
BB[0], TOS[0]
EN[0]
Ack m
PDBCHnDLYm
=
Pre-trigger m
Ch n pre-trigger m
BB[m], TOS[m]
EN[m]
Sequence Error Detection
ERR[M - 1:0]
Ch n trigger
PDBMOD
PDBCNT
PDB Counter
MULT
PRESCALER
=
Control Logic
CONT
DACINTx
DAC Interval Counter x
=
DAC interval trigger x
TOEx
EXTx
DAC ext trigger input x
DAC interval trigger x
Trigger-In 0
Trigger-In 1
Trigger-In 14
SWTRIG
TRIGSEL
POyDLY1
POyDLY2
=
=
Pulse Generation
Pulse-Out y
PDBPOEN[y]
Pulse-Out y
PDBIDLY
=
TOEx
PDB interrupt

Figure 34-1. PDB Block Diagram

In this diagram, only one PDB channel *n*, one DAC interval trigger *x*, and one Pulse-Out *y* is shown. The PDB enable control logic and the sequence error interrupt logic is not shown.

34.1.5 Modes of Operation

PDB ADC trigger operates in the following modes.

Disabled: Counter is off, all pre-trigger and trigger outputs are low if PDB is not in back-to-back operation of Bypass mode.

Debug: Counter is paused when processor is in debug mode, the counter for dac trigger also paused in Debug mode.

Enabled One-Shot: Counter is enabled and restarted at count zero upon receiving a positive edge on the selected trigger input source or software trigger is selected and SC[SWTRIG] is written with 1. In each PDB channel, an enabled pre-trigger asserts once per trigger input event; the trigger output asserts whenever any of pre-triggers is asserted.

Enabled Continuous: Counter is enabled and restarted at count zero. The counter is rolled over to zero again when the count reaches the value specified in the modulus register, and the counting is restarted. This enables a continuous stream of pre-triggers/ trigger outputs as a result of a single trigger input event.

Enabled Bypassed: The pre-trigger and trigger outputs assert immediately after a positive edge on the selected trigger input source or software trigger is selected and SC[SWTRIG] is written with 1, that is the delay registers are bypassed. It is possible to bypass any one or more of the delay registers; therefore this mode can be used in conjunction with One-Shot or Continuous mode.

34.2 PDB Signal Descriptions

This table shows the detailed description of the external signal.

Table 34-1. PDB Signal Descriptions

Signal	Description	I/O
EXTRG	External trigger input source. If the PDB is enabled and external trigger input source is selected, a positive edge on the EXTRG signal resets and starts the counter.	I

34.3 Memory Map and Register Definition

PDB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_6000	Status and Control Register (PDB0_SC)	32	R/W	0000_0000h	34.3.1/ 674
4003_6004	Modulus Register (PDB0_MOD)	32	R/W	0000_FFFFh	34.3.2/ 677
4003_6008	Counter Register (PDB0_CNT)	32	R	0000_0000h	34.3.3/ 677
4003_600C	Interrupt Delay Register (PDB0_IDLY)	32	R/W	0000_FFFFh	34.3.4/ 678
4003_6010	Channel n Control Register 1 (PDB0_CH0C1)	32	R/W	0000_0000h	34.3.5/ 678
4003_6014	Channel n Status Register (PDB0_CH0S)	32	w1c	0000_0000h	34.3.6/ 679
4003_6018	Channel n Delay 0 Register (PDB0_CH0DLY0)	32	R/W	0000_0000h	34.3.7/ 680
4003_601C	Channel n Delay 1 Register (PDB0_CH0DLY1)	32	R/W	0000_0000h	34.3.8/ 680
4003_6190	Pulse-Out n Enable Register (PDB0_POEN)	32	R/W	0000_0000h	34.3.9/ 681
4003_6194	Pulse-Out n Delay Register (PDB0_PO0DLY)	32	R/W	0000_0000h	34.3.10/ 681
4003_6198	Pulse-Out n Delay Register (PDB0_PO1DLY)	32	R/W	0000_0000h	34.3.10/ 681

34.3.1 Status and Control Register (PDBx_SC)

Addresses: PDB0_SC is 4003_6000h base + 0h offset = 4003_6000h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0				LDMOD		PDBEIE	0
W								SWTRIG
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R W	DMAEN	PRESCALER			TRGSEL			
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	PDBEN	PDBIF	PDBIE	0	MULT		CONT	LDOK
W								
Reset	0	0	0	0	0	0	0	0

PDBx_SC field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value zero.
19–18 LDMOD	Load Mode Select Selects the mode to load the MOD, IDLY, CH*n*DLY*m*, INT*x*, and PO*y*DLY registers, after 1 is written to LDOK. 00 The internal registers are loaded with the values from their buffers immediately after 1 is written to LDOK. 01 The internal registers are loaded with the values from their buffers when the PDB counter reaches the MOD register value after 1 is written to LDOK. 10 The internal registers are loaded with the values from their buffers when a trigger input event is detected after 1 is written to LDOK. 11 The internal registers are loaded with the values from their buffers when either the PDB counter reaches the MOD register value or a trigger input event is detected, after 1 is written to LDOK.
17 PDBEIE	PDB Sequence Error Interrupt Enable This bit enables the PDB sequence error interrupt. When this bit is set, any of the PDB channel sequence error flags generates a PDB sequence error interrupt. 0 PDB sequence error interrupt disabled. 1 PDB sequence error interrupt enabled.
16 SWTRIG	Software Trigger When PDB is enabled and the software trigger is selected as the trigger input source, writing 1 to this bit reset and restarts the counter. Writing 0 to this bit has no effect. Reading this bit results 0.
15 DMAEN	DMA Enable When DMA is enabled, the PDBIF flag generates a DMA request instead of an interrupt. 0 DMA disabled 1 DMA enabled
14–12 PRESCALER	Prescaler Divider Select 000 Counting uses the peripheral clock divided by multiplication factor selected by MULT. 001 Counting uses the peripheral clock divided by twice of the multiplication factor selected by MULT. 010 Counting uses the peripheral clock divided by four times of the multiplication factor selected by MULT. 011 Counting uses the peripheral clock divided by eight times of the multiplication factor selected by MULT. 100 Counting uses the peripheral clock divided by 16 times of the multiplication factor selected by MULT. 101 Counting uses the peripheral clock divided by 32 times of the multiplication factor selected by MULT. 110 Counting uses the peripheral clock divided by 64 times of the multiplication factor selected by MULT. 111 Counting uses the peripheral clock divided by 128 times of the multiplication factor selected by MULT.
11–8 TRGSEL	Trigger Input Source Select

Table continues on the next page...

PDBx_SC field descriptions (continued)

Field	Description
	Selects the trigger input source for the PDB. The trigger input source can be internal or external (EXTRG pin), or the software trigger. Please refer to Chip Configuration chapter for the actual PDB input trigger connections. 0000 Trigger-In 0 is selected 0001 Trigger-In 1 is selected 0010 Trigger-In 2 is selected 0011 Trigger-In 3 is selected 0100 Trigger-In 4 is selected 0101 Trigger-In 5 is selected 0110 Trigger-In 6 is selected 0111 Trigger-In 7 is selected 1000 Trigger-In 8 is selected 1001 Trigger-In 9 is selected 1010 Trigger-In 10 is selected 1011 Trigger-In 11 is selected 1100 Trigger-In 12 is selected 1101 Trigger-In 13 is selected 1110 Trigger-In 14 is selected 1111 Software trigger is selected
7 PDBEN	PDB Enable 0 PDB disabled. Counter is off. 1 PDB enabled
6 PDBIF	PDB Interrupt Flag This bit is set when the counter value is equal to the IDLY register. Writing zero clears this bit.
5 PDBIE	PDB Interrupt Enable. This bit enables the PDB interrupt. When this bit is set and DMAEN is cleared, PDBIF generates a PDB interrupt. 0 PDB interrupt disabled 1 PDB interrupt enabled
4 Reserved	This read-only field is reserved and always has the value zero.
3–2 MULT	Multiplication Factor Select for Prescaler This bit selects the multiplication factor of the prescaler divider for the counter clock. 00 Multiplication factor is 1 01 Multiplication factor is 10 10 Multiplication factor is 20 11 Multiplication factor is 40
1 CONT	Continuous Mode Enable This bit enables the PDB operation in Continuous mode.

Table continues on the next page...

PDBx_SC field descriptions (continued)

Field	Description
	0 PDB operation in One-Shot mode 1 PDB operation in Continuous mode
0 LDOK	Load OK Writing 1 to this bit updates the internal registers of MOD, IDLY, CHnDLYm, DACINTx, and POyDLY with the values written to their buffers. The MOD, IDLY, CHnDLYm, DACINTx, and POyDLY will take effect according to the LDMOD. After 1 is written to LDOK bit, the values in the buffers of above registers are not effective and the buffers cannot be written until the values in buffers are loaded into their internal registers. LDOK can be written only when PDBEN is set or it can be written at the same time with PDBEN being written to 1. It is automatically cleared when the values in buffers are loaded into the internal registers or the PDBEN is cleared. Writing 0 to it has no effect.

34.3.2 Modulus Register (PDBx_MOD)

Addresses: PDB0_MOD is 4003_6000h base + 4h offset = 4003_6004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																MOD															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

PDBx_MOD field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 MOD	PDB Modulus. These bits specify the period of the counter. When the counter reaches this value, it will be reset back to zero. If the PDB is in Continuous mode, the count begins anew. Reading these bits returns the value of internal register that is effective for the current cycle of PDB.

34.3.3 Counter Register (PDBx_CNT)

Addresses: PDB0_CNT is 4003_6000h base + 8h offset = 4003_6008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																CNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_CNT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 CNT	PDB Counter These read-only bits contain the current value of the counter.

34.3.4 Interrupt Delay Register (PDBx_IDLY)

Addresses: PDB0_IDLY is 4003_6000h base + Ch offset = 4003_600Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																IDLY															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

PDBx_IDLY field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 IDLY	PDB Interrupt Delay These bits specify the delay value to schedule the PDB interrupt. It can be used to schedule an independent interrupt at some point in the PDB cycle. If enabled, a PDB interrupt is generated, when the counter is equal to the IDLY. Reading these bits returns the value of internal register that is effective for the current cycle of the PDB.

34.3.5 Channel n Control Register 1 (PDBx_CHC1)

Each PDB channel has one Control Register, CHnC1. The bits in this register control the functionality of each PDB channel operation.

Addresses: PDB0_CH0C1 is 4003_6000h base + 10h offset = 4003_6010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								BB								TOS								EN							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_CH*n*C1 field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

PDBx_CH*n*C1 field descriptions (continued)

Field	Description
23–16 BB	PDB Channel Pre-Trigger Back-to-Back Operation Enable These bits enable the PDB ADC pre-trigger operation as back-to-back mode. Only lower M pre-trigger bits are implemented in this MCU. Back-to-back operation enables the ADC conversions complete to trigger the next PDB channel pre-trigger and trigger output, so that the ADC conversions can be triggered on next set of configuration and results registers. Application code must only enable the back-to-back operation of the PDB pre-triggers at the leading of the back-to-back connection chain. 0 PDB channel's corresponding pre-trigger back-to-back operation disabled. 1 PDB channel's corresponding pre-trigger back-to-back operation enabled.
15–8 TOS	PDB Channel Pre-Trigger Output Select These bits select the PDB ADC pre-trigger outputs. Only lower M pre-trigger bits are implemented in this MCU. 0 PDB channel's corresponding pre-trigger is in bypassed mode. The pre-trigger asserts one peripheral clock cycle after a rising edge is detected on selected trigger input source or software trigger is selected and SWTRIG is written with 1. 1 PDB channel's corresponding pre-trigger asserts when the counter reaches the channel delay register plus one peripheral clock cycle after a rising edge is detected on selected trigger input source or software trigger is selected and SETRIG is written with 1.
7–0 EN	PDB Channel Pre-Trigger Enable These bits enable the PDB ADC pre-trigger outputs. Only lower M pre-trigger bits are implemented in this MCU. 0 PDB channel's corresponding pre-trigger disabled. 1 PDB channel's corresponding pre-trigger enabled.

34.3.6 Channel n Status Register (PDBx_CHS)

Addresses: PDB0_CH0S is 4003_6000h base + 14h offset = 4003_6014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CF								0								ERR							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_CH*n*S field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value zero.
23–16 CF	PDB Channel Flags The CF[*m*] bit is set when the PDB counter matches the CH*n*DLY*m*. Write 0 to clear these bits.
15–8 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

PDBx_CHnS field descriptions (continued)

Field	Description
7–0 ERR	PDB Channel Sequence Error Flags Only the lower M bits are implemented in this MCU. 0 Sequence error not detected on PDB channel's corresponding pre-trigger. 1 Sequence error detected on PDB channel's corresponding pre-trigger. ADCn block can be triggered for a conversion by one pre-trigger from PDB channel *n*. When one conversion, which is triggered by one of the pre-triggers from PDB channel *n*, is in progress, new trigger from PDB channel's corresponding pre-trigger m cannot be accepted by ADCn, and ERR[m] is set. Writing 1's to clear the sequence error flags.

34.3.7 Channel n Delay 0 Register (PDBx_CHDLY0)

Addresses: PDB0_CH0DLY0 is 4003_6000h base + 18h offset = 4003_6018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																DLY															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_CHnDLY0 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 DLY	PDB Channel Delay These bits specify the delay value for the channel's corresponding pre-trigger. The pre-trigger asserts when the counter is equal to DLY. Reading these bits returns the value of internal register that is effective for the current PDB cycle.

34.3.8 Channel n Delay 1 Register (PDBx_CHDLY1)

Addresses: PDB0_CH0DLY1 is 4003_6000h base + 1Ch offset = 4003_601Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																DLY															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_CHnDLY1 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

PDBx_CH*n*DLY1 field descriptions (continued)

Field	Description
15–0 DLY	PDB Channel Delay These bits specify the delay value for the channel's corresponding pre-trigger. The pre-trigger asserts when the counter is equal to DLY. Reading these bits returns the value of internal register that is effective for the current PDB cycle.

34.3.9 Pulse-Out n Enable Register (PDBx_POEN)

Addresses: PDB0_POEN is 4003_6000h base + 190h offset = 4003_6190h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								POEN							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_POEN field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7–0 POEN	PDB Pulse-Out Enable These bits enable the pulse output. Only lower Y bits are implemented in this MCU. 0 PDB Pulse-Out disabled 1 PDB Pulse-Out enabled

34.3.10 Pulse-Out n Delay Register (PDBx_PODLY)

Addresses: PDB0_PO0DLY is 4003_6000h base + 194h offset = 4003_6194h
PDB0_PO1DLY is 4003_6000h base + 198h offset = 4003_6198h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DLY1																DLY2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PDBx_PO*n*DLY field descriptions

Field	Description
31–16 DLY1	PDB Pulse-Out Delay 1 These bits specify the delay 1 value for the PDB Pulse-Out. Pulse-Out goes high when the PDB counter is equal to the DLY1. Reading these bits returns the value of internal register that is effective for the current PDB cycle.

Table continues on the next page...

PDBx_PO*n*DLY field descriptions (continued)

Field	Description
15–0 DLY2	PDB Pulse-Out Delay 2 These bits specify the delay 2 value for the PDB Pulse-Out. Pulse-Out goes low when the PDB counter is equal to the DLY2. Reading these bits returns the value of internal register that is effective for the current PDB cycle.

34.4 Functional Description

34.4.1 PDB Pre-trigger and Trigger Outputs

The PDB contains a counter whose output is compared against several different digital values. If the PDB is enabled, a trigger input event will reset the counter and make it start to count. A trigger input event is defined as a rising edge being detected on selected trigger input source or software trigger being selected and SC[SWTRIG] is written with 1. For each channel, delay *m* determines the time between assertion of the trigger input event to the point at which changes in the pre-trigger *m* output signal is initiated. The time is defined as:

- Trigger input event to pre-trigger *m* = (prescaler X multiplication factor X delay *m*) + 2 peripheral clock cycles
- Add one additional peripheral clock cycle to determine the time at which the channel trigger output change.

Each channel is associated with one ADC block. PDB channel n pre-trigger outputs 0 to *M* and trigger output is connected to ADC hardware trigger select and hardware trigger inputs. The pre-triggers are used to precondition the ADC block prior to the actual trigger. The ADC contains *M* sets of configuration and result registers, allowing it to operate in a ping-pong fashion, alternating conversions between *M* different analog sources. The pre-trigger outputs are used to specify which signal will be sampled next. When pre-trigger *m* is asserted, the ADC conversion is triggered with set *m* of the configuration and result registers.

The waveforms shown in the following diagram illuminate the pre-trigger and trigger outputs of PDB channel *n*. The delays can be independently set via the CH*n*DLY*m* registers. And the pre-triggers can be enabled or disabled in CH*n*C1[EN[*m*]].

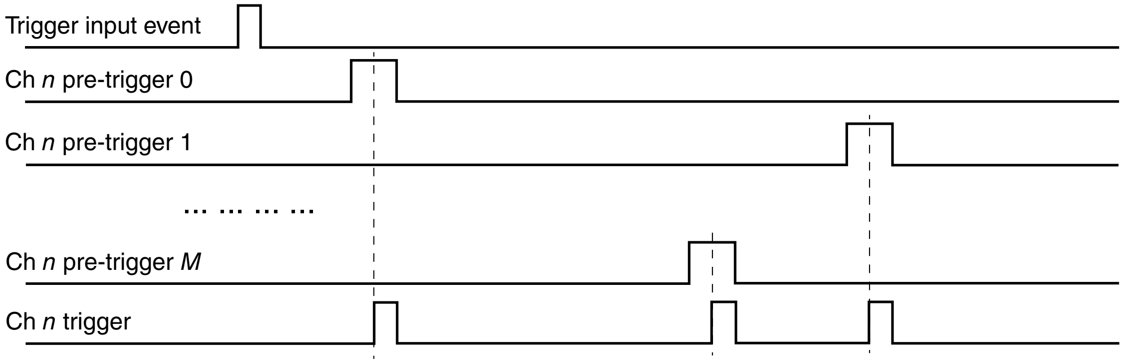


Figure 34-34. Pre-trigger and Trigger Outputs

The delay in CH*n*DLY*m* register can be optionally bypassed, if CH*n*C1[TOS[*m*]] is cleared. In this case, when the trigger input event occurs, the pre-trigger *m* is asserted after two peripheral clock cycles.

The PDB can be configured in back-to-back (B2B) operation. B2B operation enables the ADC conversions complete to trigger the next PDB channel pre-trigger and trigger outputs, so that the ADC conversions can be triggered on next set of configuration and results registers. When B2B is enabled by setting CH*n*C1[BB[*m*]], the delay *m* is ignored and the pre-trigger *m* is asserted two peripheral cycles after the acknowledgment *m* is received. The acknowledgment connections in this MCU is described in Back-to-back Acknowledgement Connections.

When an ADC conversion, which is triggered by one of the pre-triggers from PDB channel *n*, is in progress and ADC*n*SC1[COCO] is not set, a new trigger from PDB channel *n* pre-trigger *m* cannot be accepted by ADC*n*. Therefore every time when one PDB channel *n* pre-trigger and trigger output starts an ADC conversion, an internal lock associated with the corresponding pre-trigger is activated. The lock becomes inactive when the corresponding ADC*n*SC1[COCO] is set, or the corresponding PDB pre-trigger is disabled, or the PDB is disabled. The channel *n* trigger output is suppressed when any of the locks of the pre-triggers in channel *n* is active. If a new pre-trigger *m* asserts when there is active lock in the PDB channel *n*, a register flag bit, CH*n*S[ERR[*m*]], associated with the pre-trigger *m* is set. If SC[PDBEIE] is set, the sequence error interrupt is generated. Sequence error is typically happened because the delay *m* is set too short and the pre-trigger *m* asserts before the previous triggered ADC conversion is completed.

When the PDB counter reaches the value set in IDLY register, the SC[PDBIF] flag is set. A PDB interrupt can be generated if SC[PDBIE] is set and SC[DMAEN] is cleared. If SC[DMAEN] is set, PDB requests a DMA transfer when SC[PDBIF] is set.

The modulus value in MOD register, is used to reset the counter back to zero at the end of the count. If SC[CONT] bit is set, the counter will then resume a new count. Otherwise, the counter operation will cease until the next trigger input event occurs.

34.4.2 PDB Trigger Input Source Selection

The PDB has up to 15 trigger input sources, namely Trigger-In 0 to 14. They are connected to on-chip or off-chip event sources. The PDB can be triggered by software through the SC[SWTRIG]. SC[TRIGSEL] bits select the active trigger input source or software trigger.

For the trigger input sources implemented in this MCU, refer to Chip Configuration information.

34.4.3 Pulse-Out's

PDB can generate pulse outputs of configurable width. When PDB counter reaches the value set in PO*y*DLY[DLY1], the Pulse-Out goes high; when the counter reaches PO*y*DLY[DLY2], it goes low. PO*y*DLY[DLY2] can be set either greater or less than PO*y*DLY[DLY1].

Because the PDB counter is shared by both ADC pre-trigger/trigger outputs and Pulse-Out generation, they have the same time base.

The pulse-out connections implemented in this MCU are described in the device's Chip Configuration details.

34.4.4 Updating the Delay Registers

The following registers control the timing of the PDB operation; and in some of the applications, they may need to become effective at the same time.

- PDB Modulus Register (MOD)
- PDB Interrupt Delay Register (IDLY)
- PDB Channel *n* Delay *m* Register (CH*n*DLY*m*)
- DAC Interval *x* Register (DACINT*x*)
- PDB Pulse-Out *y* Delay Register (PO*y*DLY)

The internal registers of them are buffered and any values written to them are written first to their buffers. The circumstances that cause their internal registers to be updated with the values from the buffers are summarized as below table.

Table 34-36. Circumstances of Update to the Delay Registers

SC[LDMOD]	Update to the Delay Registers
00	The internal registers are loaded with the values from their buffers immediately after 1 is written to SC[LDOK].
01	The PDB counter reaches the MOD register value after 1 is written to SC[LDOK].
10	A trigger input event is detected after 1 is written to SC[LDOK].
11	Either the PDB counter reaches the MOD register value, or a trigger input event is detected, after 1 is written to SC[LDOK].

After 1 is written to SC[LDOK], the buffers cannot be written until the values in buffers are loaded into their internal registers. SC[LDOK] is self-cleared when the internal registers are loaded, so the application code can read it to determine the updates of the internal registers.

The following diagrams show the cases of the internal registers being updated with SC[LDMOD] is 00 and x1.

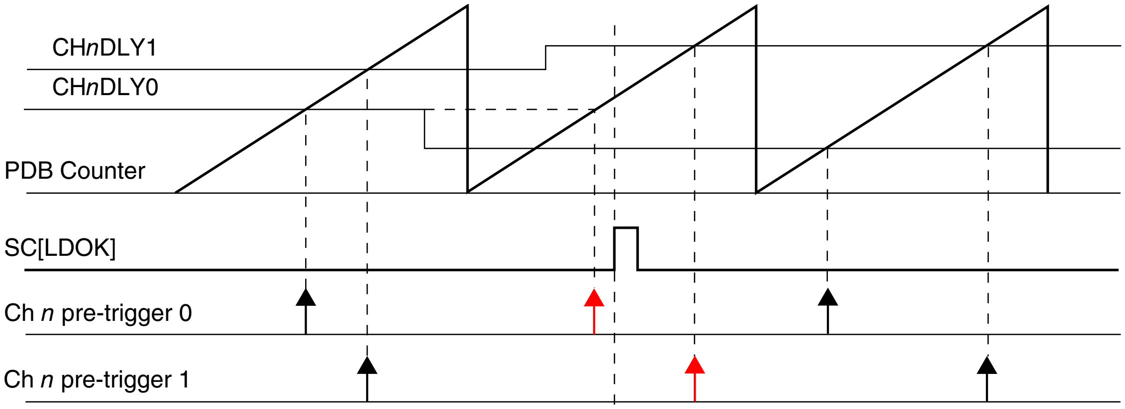

CHnDLY1
CHnDLY0
PDB Counter
SC[LDOK]
Ch n pre-trigger 0
Ch n pre-trigger 1

Figure 34-35. Registers Update with SC[LDMOD] = 00

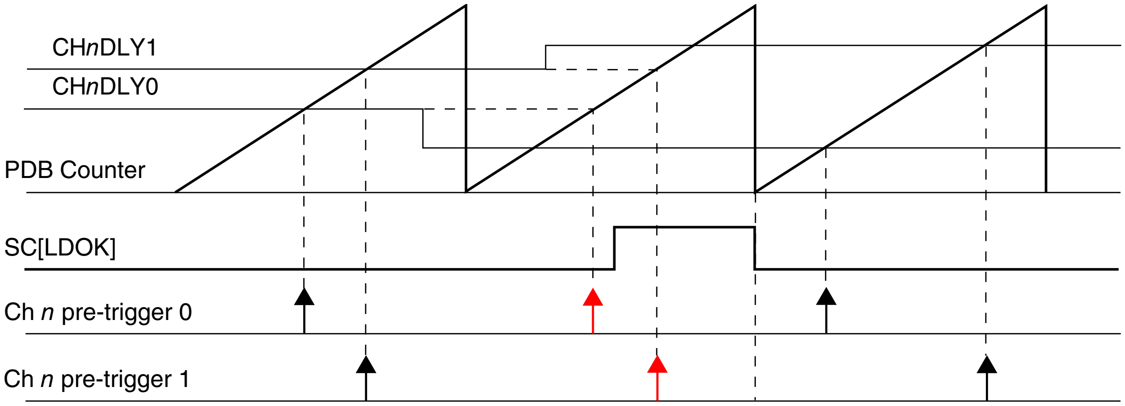

CHnDLY1
CHnDLY0
PDB Counter
SC[LDOK]
Ch n pre-trigger 0
Ch n pre-trigger 1

Figure 34-36. Registers Update with SC[LDMOD] = x1

34.4.5 Interrupts

PDB can generate two interrupts, PDB interrupt and PDB sequence error interrupt. The following table summarizes the interrupts.

Table 34-37. PDB Interrupt Summary

Interrupt	Flags	Enable Bit
PDB Interrupt	SC[PDBIF]	SC[PDBIE] = 1 and SC[DMAEN] = 0
PDB Sequence Error Interrupt	CH*n*S[ERR*m*]	SC[PDBEIE] = 1

34.4.6 DMA

If SC[DMAEN] is set, PDB can generate DMA transfer request when SC[PDBIF] is set. When DMA is enabled, the PDB interrupt will not be issued.

34.5 Application Information

34.5.1 Impact of Using the Prescaler and Multiplication Factor on Timing Resolution

Use of prescaler and multiplication factor greater than 1 limits the count/delay accuracy in terms of peripheral clock cycles (to the modulus of the prescaler X multiplication factor). If the multiplication factor is set to 1 and the prescaler is set to 2 then the only values of total peripheral clocks that can be detected are even values; if prescaler is set to 4 then the only values of total peripheral clocks that can be decoded as detected are mod(4) and so forth. If the applications need a really long delay value and use 128, then the resolution would be limited to 128 peripheral clock cycles.

Therefore, use the lowest possible prescaler and multiplication factor for a given application.

Chapter 35
FlexTimer Module (FTM)

35.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The FlexTimer module (FTM) is a two-to-eight channel timer that supports input capture, output compare, and the generation of PWM signals to control electric motor and power management applications. The FTM time reference is a 16-bit counter that can be used as an unsigned or signed counter.

35.1.1 FlexTimer philosophy

The FlexTimer is built upon a simple timer, the HCS08 Timer PWM Module – TPM, used for many years on Freescale's 8-bit microcontrollers. The FlexTimer extends the functionality to meet the demands of motor control, digital lighting solutions, and power conversion, while providing low cost and backwards compatibility with the TPM module.

Several key enhancements are made:

- Signed up counter
- Deadtime insertion hardware
- Fault control inputs
- Enhanced triggering functionality
- Initialization and polarity control

All of the features common with the TPM have fully backwards compatible register assignments. The FlexTimer can also use code on the same core platform without change to perform the same functions.

Motor control and power conversion features have been added through a dedicated set of registers and defaults turn off all new features. The new features, such as hardware deadtime insertion, polarity, fault control, and output forcing and masking, greatly reduce loading on the execution software and are usually each controlled by a group of registers.

FlexTimer input triggers can be from comparators, ADC, or other submodules to initiate timer functions automatically. These triggers can be linked in a variety of ways during integration of the sub modules so please note the options available for used FlexTimer configuration.

Several FlexTimers may be synchronized to provide a larger timer with their counters incrementing in unison, assuming the initialization, the input clocks, the initial and final counting values are the same in each FlexTimer.

All main user access registers are buffered to ease the load on the executing software. A number of trigger options exist to determine which registers are updated with this user defined data.

35.1.2 Features

The FTM features include:

- FTM source clock is selectable
  - Source clock can be the system clock, the fixed frequency clock, or an external clock
  - Fixed frequency clock is an additional clock input to allow the selection of an on chip clock source other than the system clock
  - Selecting external clock connects FTM clock to a chip level input pin therefore allowing to synchronize the FTM counter with an off chip clock source
- Prescaler divide-by 1, 2, 4, 8, 16, 32, 64, or 128
- 16-bit counter
  - It can be a free-running counter or a counter with initial and final value
  - The counting can be up or up-down
- Each channel can be configured for input capture, output compare, or edge-aligned PWM mode
- In Input Capture mode:

  - The capture can occur on rising edges, falling edges or both edges
  - An input filter can be selected for some channels
- In Output Compare mode the output signal can be set, cleared, or toggled on match
- All channels can be configured for center-aligned PWM mode
- Each pair of channels can be combined to generate a PWM signal with independent control of both edges of PWM signal
- The FTM channels can operate as pairs with equal outputs, pairs with complementary outputs, or independent channels with independent outputs
- The deadtime insertion is available for each complementary pair
- Generation of match triggers
- Software control of PWM outputs
- Up to 4 fault inputs for global fault control
- The polarity of each channel is configurable
- The generation of an interrupt per channel
- The generation of an interrupt when the counter overflows
- The generation of an interrupt when the fault condition is detected
- Synchronized loading of write buffered FTM registers
- Write protection for critical registers
- Backwards compatible with TPM
- Testing of input captures for a stuck at zero and one conditions
- Dual edge capture for pulse and period width measurement
- Quadrature decoder with input filters, relative position counting, and interrupt on position count or capture of position count on external event

35.1.3 Modes of operation

When the MCU is in an active BDM mode, the FTM temporarily suspends all counting until the MCU returns to normal user operating mode. During Stop mode, all FTM input clocks are stopped, so the FTM is effectively disabled until clocks resume. During Wait mode, the FTM continues to operate normally. If the FTM does not need to produce a real time reference or provide the interrupt sources needed to wake the MCU from Wait mode, the power can then be saved by disabling FTM functions before entering Wait mode.

35.1.4 Block diagram

The FTM uses one input/output (I/O) pin per channel, CHn (FTM channel (n)) where n is the channel number (0–7).

The following figure shows the FTM structure. The central component of the FTM is the 16-bit counter with programmable initial and final values and its counting can be up or up-down.

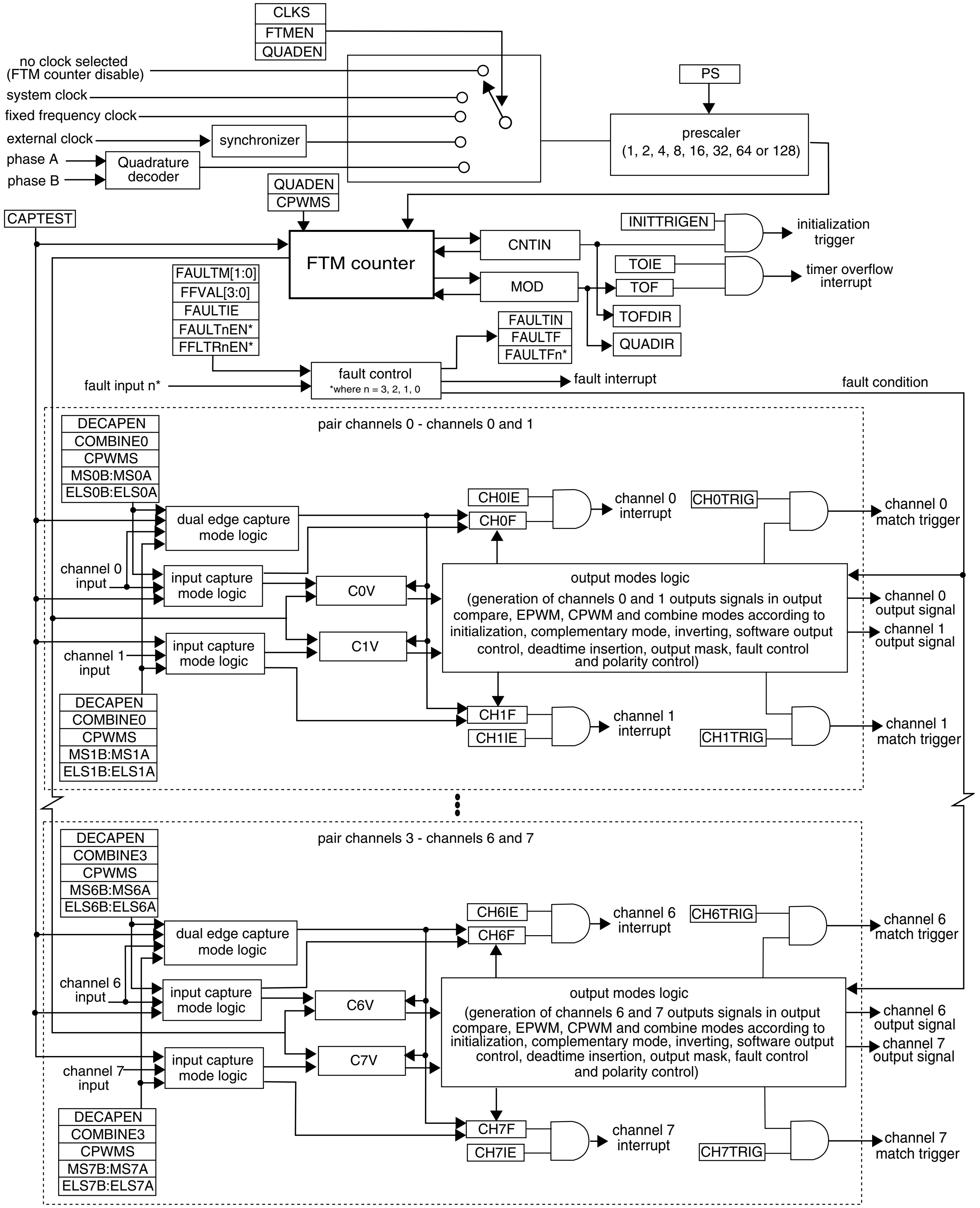

CLKS
FTMEN
QUADEN
no clock selected (FTM counter disable)
system clock
fixed frequency clock
external clock
synchronizer
phase A
phase B
Quadrature decoder
PS
prescaler (1, 2, 4, 8, 16, 32, 64 or 128)
QUADEN
CPWMS
CAPTEST
FTM counter
CNTIN
MOD
INITTRIGEN
initialization trigger
TOIE
TOF
timer overflow interrupt
TOFDIR
QUADIR
FAULTM[1:0]
FFVAL[3:0]
FAULTIE
FAULTnEN*
FFLTRnEN*
FAULTIN
FAULTF
FAULTFn*
fault control
*where n = 3, 2, 1, 0
fault input n*
fault interrupt
fault condition
pair channels 0 - channels 0 and 1
DECAPEN
COMBINE0
CPWMS
MS0B:MS0A
ELS0B:ELS0A
dual edge capture mode logic
channel 0 input
input capture mode logic
C0V
C1V
channel 1 input
input capture mode logic
DECAPEN
COMBINE0
CPWMS
MS1B:MS1A
ELS1B:ELS1A
CH0IE
CH0F
channel 0 interrupt
CH0TRIG
channel 0 match trigger
output modes logic
(generation of channels 0 and 1 outputs signals in output compare, EPWM, CPWM and combine modes according to initialization, complementary mode, inverting, software output control, deadtime insertion, output mask, fault control and polarity control)
channel 0 output signal
channel 1 output signal
CH1F
CH1IE
channel 1 interrupt
CH1TRIG
channel 1 match trigger
pair channels 3 - channels 6 and 7
DECAPEN
COMBINE3
CPWMS
MS6B:MS6A
ELS6B:ELS6A
dual edge capture mode logic
channel 6 input
input capture mode logic
C6V
C7V
channel 7 input
input capture mode logic
DECAPEN
COMBINE3
CPWMS
MS7B:MS7A
ELS7B:ELS7A
CH6IE
CH6F
channel 6 interrupt
CH6TRIG
channel 6 match trigger
output modes logic
(generation of channels 6 and 7 outputs signals in output compare, EPWM, CPWM and combine modes according to initialization, complementary mode, inverting, software output control, deadtime insertion, output mask, fault control and polarity control)
channel 6 output signal
channel 7 output signal
CH7F
CH7IE
channel 7 interrupt
CH7TRIG
channel 7 match trigger

Figure 35-1. FTM block diagram

35.2 FTM signal descriptions

Table 35-1 shows the user-accessible signals for the FTM.

Table 35-1. FTM signal descriptions

Signal	Description	I/O	Function
EXTCLK	External clock. FTM external clock can be selected to drive the FTM counter.	I	The external clock input signal is used as the FTM counter clock if selected by CLKS[1:0] bits in the SC register. This clock signal must not exceed 1/4 of system clock frequency. The FTM counter prescaler selection and settings are also used when an external clock is selected.
CHn	FTM channel (n), where n can be 7-0	I/O	Each FTM channel can be configured to operate either as input or output. The direction associated with each channel, input or output, is selected according to the mode assigned for that channel.
FAULTj	Fault input (j), where j can be 3-0	I	The fault input signals are used to control the CHn channel output state. If a fault is detected, the FAULTj signal is asserted and the channel output is put in a safe state. The behavior of the fault logic is defined by the FAULTM[1:0] control bits in the MODE register and FAULTEN bit in the COMBINEm register. Note that each FAULTj input may affect all channels selectively since FAULTM[1:0] and FAULTEN control bits are defined for each pair of channels. Because there are several FAULTj inputs, maximum of 4 for the FTM module, each one of these inputs is activated by the FAULTjEN bit in the FLTCTRL register.
PHA	Quadrature decoder phase A input. Input pin associated with quadrature decoder phase A.	I	The quadrature decoder phase A input is used as the Quadrature Decoder mode is selected. The phase A input signal is one of the signals that control the FTM counter increment or decrement in the Quadrature Decoder mode.
PHB	Quadrature decoder phase B input. Input pin associated with quadrature decoder phase B.	I	The quadrature decoder phase B input is used as the Quadrature Decoder mode is selected. The phase B input signal is one of the signals that control the FTM counter increment or decrement in the Quadrature Decoder mode.

35.3 Memory map and register definition

35.3.1 Memory map

This section presents a high-level summary of the FTM registers and how they are mapped.

The first set has the original TPM registers.

The second set has the FTM specific registers. Any second set registers, or bits within these registers, that are used by an unavailable function in the FTM configuration remain in the memory map and in the reset value, so they have no active function.

Note

Do not write to the FTM specific registers (second set registers) when FTMEN = 0.

35.3.2 Register descriptions

Accesses to reserved addresses result in transfer errors. Registers for absent channels are considered reserved.

FTM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_8000	Status And Control (FTM0_SC)	32	R/W	0000_0000h	35.3.3/ 697
4003_8004	Counter (FTM0_CNT)	32	R/W	0000_0000h	35.3.4/ 698
4003_8008	Modulo (FTM0_MOD)	32	R/W	0000_0000h	35.3.5/ 699
4003_800C	Channel (n) Status And Control (FTM0_C0SC)	32	R/W	0000_0000h	35.3.6/ 700
4003_8010	Channel (n) Value (FTM0_C0V)	32	R/W	0000_0000h	35.3.7/ 703
4003_8014	Channel (n) Status And Control (FTM0_C1SC)	32	R/W	0000_0000h	35.3.6/ 700
4003_8018	Channel (n) Value (FTM0_C1V)	32	R/W	0000_0000h	35.3.7/ 703
4003_801C	Channel (n) Status And Control (FTM0_C2SC)	32	R/W	0000_0000h	35.3.6/ 700
4003_8020	Channel (n) Value (FTM0_C2V)	32	R/W	0000_0000h	35.3.7/ 703
4003_8024	Channel (n) Status And Control (FTM0_C3SC)	32	R/W	0000_0000h	35.3.6/ 700
4003_8028	Channel (n) Value (FTM0_C3V)	32	R/W	0000_0000h	35.3.7/ 703
4003_802C	Channel (n) Status And Control (FTM0_C4SC)	32	R/W	0000_0000h	35.3.6/ 700

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_8030	Channel (n) Value (FTM0_C4V)	32	R/W	0000_0000h	35.3.7/ 703
4003_8034	Channel (n) Status And Control (FTM0_C5SC)	32	R/W	0000_0000h	35.3.6/ 700
4003_8038	Channel (n) Value (FTM0_C5V)	32	R/W	0000_0000h	35.3.7/ 703
4003_803C	Channel (n) Status And Control (FTM0_C6SC)	32	R/W	0000_0000h	35.3.6/ 700
4003_8040	Channel (n) Value (FTM0_C6V)	32	R/W	0000_0000h	35.3.7/ 703
4003_8044	Channel (n) Status And Control (FTM0_C7SC)	32	R/W	0000_0000h	35.3.6/ 700
4003_8048	Channel (n) Value (FTM0_C7V)	32	R/W	0000_0000h	35.3.7/ 703
4003_804C	Counter Initial Value (FTM0_CNTIN)	32	R/W	0000_0000h	35.3.8/ 704
4003_8050	Capture And Compare Status (FTM0_STATUS)	32	R/W	0000_0000h	35.3.9/ 704
4003_8054	Features Mode Selection (FTM0_MODE)	32	R/W	0000_0004h	35.3.10/ 706
4003_8058	Synchronization (FTM0_SYNC)	32	R/W	0000_0000h	35.3.11/ 708
4003_805C	Initial State For Channels Output (FTM0_OUTINIT)	32	R/W	0000_0000h	35.3.12/ 711
4003_8060	Output Mask (FTM0_OUTMASK)	32	R/W	0000_0000h	35.3.13/ 712
4003_8064	Function For Linked Channels (FTM0_COMBINE)	32	R/W	0000_0000h	35.3.14/ 714
4003_8068	Deadtime Insertion Control (FTM0_DEADTIME)	32	R/W	0000_0000h	35.3.15/ 719
4003_806C	FTM External Trigger (FTM0_EXTTRIG)	32	R/W	0000_0000h	35.3.16/ 720
4003_8070	Channels Polarity (FTM0_POL)	32	R/W	0000_0000h	35.3.17/ 722
4003_8074	Fault Mode Status (FTM0_FMS)	32	R/W	0000_0000h	35.3.18/ 724
4003_8078	Input Capture Filter Control (FTM0_FILTER)	32	R/W	0000_0000h	35.3.19/ 726
4003_807C	Fault Control (FTM0_FLTCTRL)	32	R/W	0000_0000h	35.3.20/ 727

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_8080	Quadrature Decoder Control And Status (FTM0_QDCTRL)	32	R/W	0000_0000h	35.3.21/ 729
4003_8084	Configuration (FTM0_CONF)	32	R/W	0000_0000h	35.3.22/ 731
4003_8088	FTM Fault Input Polarity (FTM0_FLTPOL)	32	R/W	0000_0000h	35.3.23/ 733
4003_808C	Synchronization Configuration (FTM0_SYNCONF)	32	R/W	0000_0000h	35.3.24/ 734
4003_8090	FTM Inverting Control (FTM0_INVCTRL)	32	R/W	0000_0000h	35.3.25/ 736
4003_8094	FTM Software Output Control (FTM0_SWOCTRL)	32	R/W	0000_0000h	35.3.26/ 737
4003_8098	FTM PWM Load (FTM0_PWMLOAD)	32	R/W	0000_0000h	35.3.27/ 740
4003_9000	Status And Control (FTM1_SC)	32	R/W	0000_0000h	35.3.3/ 697
4003_9004	Counter (FTM1_CNT)	32	R/W	0000_0000h	35.3.4/ 698
4003_9008	Modulo (FTM1_MOD)	32	R/W	0000_0000h	35.3.5/ 699
4003_900C	Channel (n) Status And Control (FTM1_C0SC)	32	R/W	0000_0000h	35.3.6/ 700
4003_9010	Channel (n) Value (FTM1_C0V)	32	R/W	0000_0000h	35.3.7/ 703
4003_9014	Channel (n) Status And Control (FTM1_C1SC)	32	R/W	0000_0000h	35.3.6/ 700
4003_9018	Channel (n) Value (FTM1_C1V)	32	R/W	0000_0000h	35.3.7/ 703
4003_901C	Channel (n) Status And Control (FTM1_C2SC)	32	R/W	0000_0000h	35.3.6/ 700
4003_9020	Channel (n) Value (FTM1_C2V)	32	R/W	0000_0000h	35.3.7/ 703
4003_9024	Channel (n) Status And Control (FTM1_C3SC)	32	R/W	0000_0000h	35.3.6/ 700
4003_9028	Channel (n) Value (FTM1_C3V)	32	R/W	0000_0000h	35.3.7/ 703
4003_902C	Channel (n) Status And Control (FTM1_C4SC)	32	R/W	0000_0000h	35.3.6/ 700
4003_9030	Channel (n) Value (FTM1_C4V)	32	R/W	0000_0000h	35.3.7/ 703

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_9034	Channel (n) Status And Control (FTM1_C5SC)	32	R/W	0000_0000h	35.3.6/ 700
4003_9038	Channel (n) Value (FTM1_C5V)	32	R/W	0000_0000h	35.3.7/ 703
4003_903C	Channel (n) Status And Control (FTM1_C6SC)	32	R/W	0000_0000h	35.3.6/ 700
4003_9040	Channel (n) Value (FTM1_C6V)	32	R/W	0000_0000h	35.3.7/ 703
4003_9044	Channel (n) Status And Control (FTM1_C7SC)	32	R/W	0000_0000h	35.3.6/ 700
4003_9048	Channel (n) Value (FTM1_C7V)	32	R/W	0000_0000h	35.3.7/ 703
4003_904C	Counter Initial Value (FTM1_CNTIN)	32	R/W	0000_0000h	35.3.8/ 704
4003_9050	Capture And Compare Status (FTM1_STATUS)	32	R/W	0000_0000h	35.3.9/ 704
4003_9054	Features Mode Selection (FTM1_MODE)	32	R/W	0000_0004h	35.3.10/ 706
4003_9058	Synchronization (FTM1_SYNC)	32	R/W	0000_0000h	35.3.11/ 708
4003_905C	Initial State For Channels Output (FTM1_OUTINIT)	32	R/W	0000_0000h	35.3.12/ 711
4003_9060	Output Mask (FTM1_OUTMASK)	32	R/W	0000_0000h	35.3.13/ 712
4003_9064	Function For Linked Channels (FTM1_COMBINE)	32	R/W	0000_0000h	35.3.14/ 714
4003_9068	Deadtime Insertion Control (FTM1_DEADTIME)	32	R/W	0000_0000h	35.3.15/ 719
4003_906C	FTM External Trigger (FTM1_EXTTRIG)	32	R/W	0000_0000h	35.3.16/ 720
4003_9070	Channels Polarity (FTM1_POL)	32	R/W	0000_0000h	35.3.17/ 722
4003_9074	Fault Mode Status (FTM1_FMS)	32	R/W	0000_0000h	35.3.18/ 724
4003_9078	Input Capture Filter Control (FTM1_FILTER)	32	R/W	0000_0000h	35.3.19/ 726
4003_907C	Fault Control (FTM1_FLTCTRL)	32	R/W	0000_0000h	35.3.20/ 727
4003_9080	Quadrature Decoder Control And Status (FTM1_QDCTRL)	32	R/W	0000_0000h	35.3.21/ 729

Table continues on the next page...

FTM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_9084	Configuration (FTM1_CONF)	32	R/W	0000_0000h	35.3.22/ 731
4003_9088	FTM Fault Input Polarity (FTM1_FLTPOL)	32	R/W	0000_0000h	35.3.23/ 733
4003_908C	Synchronization Configuration (FTM1_SYNCONF)	32	R/W	0000_0000h	35.3.24/ 734
4003_9090	FTM Inverting Control (FTM1_INVCTRL)	32	R/W	0000_0000h	35.3.25/ 736
4003_9094	FTM Software Output Control (FTM1_SWOCTRL)	32	R/W	0000_0000h	35.3.26/ 737
4003_9098	FTM PWM Load (FTM1_PWMLOAD)	32	R/W	0000_0000h	35.3.27/ 740

35.3.3 Status And Control (FTMx_SC)

SC contains the overflow status flag and control bits used to configure the interrupt enable, FTM configuration, clock source, and prescaler factor. These controls relate to all channels within this module.

Addresses: FTM0_SC is 4003_8000h base + 0h offset = 4003_8000h

FTM1_SC is 4003_9000h base + 0h offset = 4003_9000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TOF	TOIE	CPWMS	CLKS		PS		
W									0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_SC field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 TOF	Timer Overflow Flag Set by hardware when the FTM counter passes the value in the MOD register. The TOF bit is cleared by reading the SC register while TOF is set and then writing a 0 to TOF bit. Writing a 1 to TOF has no effect.

Table continues on the next page...

FTMx_SC field descriptions (continued)

Field	Description
	If another FTM overflow occurs between the read and write operations, the write operation has no effect; therefore, TOF remains set indicating an overflow has occurred. In this case, a TOF interrupt request is not lost due to the clearing sequence for a previous TOF. 0 FTM counter has not overflowed. 1 FTM counter has overflowed.
6 TOIE	Timer Overflow Interrupt Enable Enables FTM overflow interrupts. 0 Disable TOF interrupts. Use software polling. 1 Enable TOF interrupts. An interrupt is generated when TOF equals one.
5 CPWMS	Center-Aligned PWM Select Selects CPWM mode. This mode configures the FTM to operate in Up-Down Counting mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 FTM counter operates in Up Counting mode. 1 FTM counter operates in Up-Down Counting mode.
4–3 CLKS	Clock Source Selection Selects one of the three FTM counter clock sources. This field is write protected. It can be written only when MODE[WPDIS] = 1. 00 No clock selected. This in effect disables the FTM counter. 01 System clock 10 Fixed frequency clock 11 External clock
2–0 PS	Prescale Factor Selection Selects one of 8 division factors for the clock source selected by CLKS. The new prescaler factor affects the clock source on the next system clock cycle after the new value is updated into the register bits. This field is write protected. It can be written only when MODE[WPDIS] = 1. 000 Divide by 1 001 Divide by 2 010 Divide by 4 011 Divide by 8 100 Divide by 16 101 Divide by 32 110 Divide by 64 111 Divide by 128

35.3.4 Counter (FTMx_CNT)

The CNT register contains the FTM counter value.

Reset clears the CNT register. Writing any value to COUNT updates the counter with its initial value, CNTIN.

When BDM is active, the FTM counter is frozen. This is the value that you may read.

Addresses: FTM0_CNT is 4003_8000h base + 4h offset = 4003_8004h
FTM1_CNT is 4003_9000h base + 4h offset = 4003_9004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_CNT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 COUNT	Counter Value

35.3.5 Modulo (FTMx_MOD)

The Modulo register contains the modulo value for the FTM counter. After the FTM counter reaches the modulo value, the overflow flag (TOF) becomes set at the next clock, and the next value of FTM counter depends on the selected counting method; see Counter.

Writing to the MOD register latches the value into a buffer. The MOD register is updated with the value of its write buffer according to Registers updated from write buffers.

If FTMEN = 0, this write coherency mechanism may be manually reset by writing to the SC register whether BDM is active or not.

Initialize the FTM counter, by writing to CNT, before writing to the MOD register to avoid confusion about when the first counter overflow will occur.

Addresses: FTM0_MOD is 4003_8000h base + 8h offset = 4003_8008h
FTM1_MOD is 4003_9000h base + 8h offset = 4003_9008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																MOD															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_MOD field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–0 MOD	Modulo Value

35.3.6 Channel (n) Status And Control (FTMx_CSC)

CnSC contains the channel-interrupt-status flag and control bits used to configure the interrupt enable, channel configuration, and pin function.

Table 35-67. Mode, edge, and level selection

DECAPEN	COMBINE	CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
X	X	X	XX	0	None	Pin not used for FTM

Table continues on the next page...

Table 35-67. Mode, edge, and level selection (continued)

DECAPEN	COMBINE	CPWMS	MSnB:MSnA	ELSnB:ELSnA	Mode	Configuration
0	0	0	0	1	Input Capture	Capture on Rising Edge Only
				10		Capture on Falling Edge Only
				11		Capture on Rising or Falling Edge
			1	1	Output Compare	Toggle Output on match
				10		Clear Output on match
				11		Set Output on match
			1X	10	Edge-Aligned PWM	High-true pulses (clear Output on match)
				X1		Low-true pulses (set Output on match)
		1	XX	10	Center-Aligned PWM	High-true pulses (clear Output on match-up)
				X1		Low-true pulses (set Output on match-up)
	1	0	XX	10	Combine PWM	High-true pulses (set on channel (n) match, and clear on channel (n+1) match)
				X1		Low-true pulses (clear on channel (n) match, and set on channel (n +1) match)
1	0	0	X0	See the following table (Table 35-8).	Dual Edge Capture	One-Shot Capture mode
			X1			Continuous Capture mode

Table 35-68. Dual Edge Capture mode — edge polarity selection

ELSnB	ELSnA	Channel Port Enable	Detected Edges
0	0	Disabled	No edge
0	1	Enabled	Rising edge
1	0	Enabled	Falling edge
1	1	Enabled	Rising and falling edges

Addresses: FTM0_C0SC is 4003_8000h base + Ch offset = 4003_800Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CHF	CHIE	MSB	MSA	ELSB	ELSA	0	DMA
W									0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_CnSC field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 CHF	Channel Flag Set by hardware when an event occurs on the channel. CHF is cleared by reading the CSC register while CHnF is set and then writing a 0 to the CHF bit. Writing a 1 to CHF has no effect. If another event occurs between the read and write operations, the write operation has no effect; therefore, CHF remains set indicating an event has occurred. In this case a CHF interrupt request is not lost due to the clearing sequence for a previous CHF. 0 No channel event has occurred. 1 A channel event has occurred.
6 CHIE	Channel Interrupt Enable Enables channel interrupts. 0 Disable channel interrupts. Use software polling. 1 Enable channel interrupts.
5 MSB	Channel Mode Select Used for further selections in the channel logic. Its functionality is dependent on the channel mode. See Table 35-7. This field is write protected. It can be written only when MODE[WPDIS] = 1.
4 MSA	Channel Mode Select

Table continues on the next page...

FTMx_C*n*SC field descriptions (continued)

Field	Description
	Used for further selections in the channel logic. Its functionality is dependent on the channel mode. See Table 35-7. This field is write protected. It can be written only when MODE[WPDIS] = 1.
3 ELSB	Edge or Level Select The functionality of ELSB and ELSA depends on the channel mode. See Table 35-7. This field is write protected. It can be written only when MODE[WPDIS] = 1.
2 ELSA	Edge or Level Select The functionality of ELSB and ELSA depends on the channel mode. See Table 35-7. This field is write protected. It can be written only when MODE[WPDIS] = 1.
1 Reserved	This read-only field is reserved and always has the value zero.
0 DMA	DMA Enable Enables DMA transfers for the channel. 0 Disable DMA transfers. 1 Enable DMA transfers.

35.3.7 Channel (n) Value (FTMx_CV)

These registers contain the captured FTM counter value for the input modes or the match value for the output modes.

In Input Capture, Capture Test, and Dual Edge Capture modes, any write to a CnV register is ignored.

In output modes, writing to a CnV register latches the value into a buffer. A CnV register is updated with the value of its write buffer according to Registers updated from write buffers.

If FTMEN = 0, this write coherency mechanism may be manually reset by writing to the CnSC register whether BDM mode is active or not.

Addresses: FTM0_C0V is 4003_8000h base + 10h offset = 4003_8010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																VAL															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_C*n*V field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 VAL	Channel Value Captured FTM counter value of the input modes or the match value for the output modes

35.3.8 Counter Initial Value (FTMx_CNTIN)

The Counter Initial Value register contains the initial value for the FTM counter.

Writing to the CNTIN register latches the value into a buffer. The CNTIN register is updated with the value of its write buffer according to Registers updated from write buffers.

When the FTM clock is initially selected, by writing a non-zero value to the CLKS bits, the FTM counter starts with the value 0x0000. To avoid this behavior, before the first write to select the FTM clock, write the new value to the the CNTIN register and then initialize the FTM counter by writing any value to the CNT register.

Addresses: FTM0_CNTIN is 4003_8000h base + 4Ch offset = 4003_804Ch
FTM1_CNTIN is 4003_9000h base + 4Ch offset = 4003_904Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																INIT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_CNTIN field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–0 INIT	Initial Value Of The FTM Counter

35.3.9 Capture And Compare Status (FTMx_STATUS)

The STATUS register contains a copy of the status flag CHnF bit in CnSC for each FTM channel for software convenience.

Each CHnF bit in STATUS is a mirror of CHnF bit in CnSC. All CHnF bits can be checked using only one read of STATUS. All CHnF bits can be cleared by reading STATUS followed by writing 0x00 to STATUS.

Hardware sets the individual channel flags when an event occurs on the channel. CHF is cleared by reading STATUS while CHnF is set and then writing a 0 to the CHF bit. Writing a 1 to CHF has no effect.

If another event occurs between the read and write operations, the write operation has no effect; therefore, CHF remains set indicating an event has occurred. In this case, a CHF interrupt request is not lost due to the clearing sequence for a previous CHF.

NOTE

The STATUS register should be used only in Combine mode.

Addresses: FTM0_STATUS is 4003_8000h base + 50h offset = 4003_8050h
FTM1_STATUS is 4003_9000h base + 50h offset = 4003_9050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								CH7F	CH6F	CH5F	CH4F	CH3F	CH2F	CH1F	CH0F
W									0	0	0	0	0	0	0	0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_STATUS field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 CH7F	Channel 7 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
6 CH6F	Channel 6 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
5 CH5F	Channel 5 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.

Table continues on the next page...

FTMx_STATUS field descriptions (continued)

Field	Description
4 CH4F	Channel 4 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
3 CH3F	Channel 3 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
2 CH2F	Channel 2 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
1 CH1F	Channel 1 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.
0 CH0F	Channel 0 Flag See the register description. 0 No channel event has occurred. 1 A channel event has occurred.

35.3.10 Features Mode Selection (FTMx_MODE)

This register contains the global enable bit for FTM-specific features and the control bits used to configure:

- Fault control mode and interrupt
- Capture Test mode
- PWM synchronization
- Write protection
- Channel output initialization

These controls relate to all channels within this module.

Addresses: FTM0_MODE is 4003_8000h base + 54h offset = 4003_8054h
FTM1_MODE is 4003_9000h base + 54h offset = 4003_9054h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R W	FAULTIE	FAULTM		CAPTEST	PWMSYNC	WPDIS	INIT	FTMEN
Reset	0	0	0	0	0	1	0	0

FTMx_MODE field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 FAULTIE	Fault Interrupt Enable Enables the generation of an interrupt when a fault is detected by FTM and the FTM fault control is enabled. 0 Fault control interrupt is disabled. 1 Fault control interrupt is enabled.
6–5 FAULTM	Fault Control Mode Defines the FTM fault control mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 00 Fault control is disabled for all channels. 01 Fault control is enabled for even channels only (channels 0, 2, 4, and 6), and the selected mode is the manual fault clearing. 10 Fault control is enabled for all channels, and the selected mode is the manual fault clearing. 11 Fault control is enabled for all channels, and the selected mode is the automatic fault clearing.
4 CAPTEST	Capture Test Mode Enable Enables the capture test mode. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Capture test mode is disabled. 1 Capture test mode is enabled.

Table continues on the next page...

FTMx_MODE field descriptions (continued)

Field	Description
3 PWMSYNC	PWM Synchronization Mode Selects which triggers can be used by MOD, CnV, OUTMASK, and FTM counter synchronization. See PWM synchronization. The PWMSYNC bit configures the synchronization when SYNCMODE is zero. 0 No restrictions. Software and hardware triggers can be used by MOD, CnV, OUTMASK, and FTM counter synchronization. 1 Software trigger can only be used by MOD and CnV synchronization, and hardware triggers can only be used by OUTMASK and FTM counter synchronization.
2 WPDIS	Write Protection Disable When write protection is enabled (WPDIS = 0), write protected bits cannot be written. When write protection is disabled (WPDIS = 1), write protected bits can be written. The WPDIS bit is the negation of the WPEN bit. WPDIS is cleared when 1 is written to WPEN. WPDIS is set when WPEN bit is read as a 1 and then 1 is written to WPDIS. Writing 0 to WPDIS has no effect. 0 Write protection is enabled. 1 Write protection is disabled.
1 INIT	Initialize The Channels Output When a 1 is written to INIT bit the channels output is initialized according to the state of their corresponding bit in the OUTINIT register. Writing a 0 to INIT bit has no effect. The INIT bit is always read as 0.
0 FTMEN	FTM Enable This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Only the TPM-compatible registers (first set of registers) can be used without any restriction. Do not use the FTM-specific registers. 1 All registers including the FTM-specific registers (second set of registers) are available for use with no restrictions.

35.3.11 Synchronization (FTMx_SYNC)

This register configures the PWM synchronization.

A synchronization event can perform the synchronized update of MOD, CV, and OUTMASK registers with the value of their write buffer and the FTM counter initialization.

NOTE

> The software trigger, SWSYNC bit, and hardware triggers TRIG0, TRIG1, and TRIG2 bits have a potential conflict if used together when SYNCMODE = 0. Use only hardware or software triggers but not both at the same time, otherwise unpredictable behavior is likely to happen.

The selection of the loading point, CNTMAX and CNTMIN bits, is intended to provide the update of MOD, CNTIN, and CnV registers across all enabled channels simultaneously. The use of the loading point selection together with SYNCMODE = 0 and hardware trigger selection, TRIG0, TRIG1, or TRIG2 bits, is likely to result in unpredictable behavior.

The synchronization event selection also depends on the PWMSYNC (MODE register) and SYNCMODE (SYNCONF register) bits. See PWM synchronization.

Addresses: FTM0_SYNC is 4003_8000h base + 58h offset = 4003_8058h
FTM1_SYNC is 4003_9000h base + 58h offset = 4003_9058h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R W	SWSYNC	TRIG2	TRIG1	TRIG0	SYNCHOM	REINIT	CNTMAX	CNTMIN
Reset	0	0	0	0	0	0	0	0

FTMx_SYNC field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 SWSYNC	PWM Synchronization Software Trigger Selects the software trigger as the PWM synchronization trigger. The software trigger happens when a 1 is written to SWSYNC bit. 0 Software trigger is not selected. 1 Software trigger is selected.
6 TRIG2	PWM Synchronization Hardware Trigger 2 Enables hardware trigger 2 to the PWM synchronization. Hardware trigger 2 happens when a rising edge is detected at the trigger 2 input signal.

Table continues on the next page...

FTMx_SYNC field descriptions (continued)

Field	Description
	0 Trigger is disabled. 1 Trigger is enabled.
5 TRIG1	PWM Synchronization Hardware Trigger 1 Enables hardware trigger 1 to the PWM synchronization. Hardware trigger 1 happens when a rising edge is detected at the trigger 1 input signal. 0 Trigger is disabled. 1 Trigger is enabled.
4 TRIG0	PWM Synchronization Hardware Trigger 0 Enables hardware trigger 0 to the PWM synchronization. Hardware trigger 0 happens when a rising edge is detected at the trigger 0 input signal. 0 Trigger is disabled. 1 Trigger is enabled.
3 SYNCHOM	Output Mask Synchronization Selects when the OUTMASK register is updated with the value of its buffer. 0 OUTMASK register is updated with the value of its buffer in all rising edges of the system clock. 1 OUTMASK register is updated with the value of its buffer only by the PWM synchronization.
2 REINIT	FTM Counter Reinitialization By Synchronization (FTM counter synchronization) Determines if the FTM counter is reinitialized when the selected trigger for the synchronization is detected. The REINIT bit configures the synchronization when SYNCMODE is zero. 0 FTM counter continues to count normally. 1 FTM counter is updated with its initial value when the selected trigger is detected.
1 CNTMAX	Maximum Loading Point Enable Selects the maximum loading point to PWM synchronization. See Boundary cycle and loading points. If CNTMAX is one, the selected loading point is when the FTM counter reaches its maximum value (MOD register). 0 The maximum loading point is disabled. 1 The maximum loading point is enabled.
0 CNTMIN	Minimum Loading Point Enable Selects the minimum loading point to PWM synchronization. See Boundary cycle and loading points. If CNTMIN is one, the selected loading point is when the FTM counter reaches its minimum value (CNTIN register). 0 The minimum loading point is disabled. 1 The minimum loading point is enabled.

35.3.12 Initial State For Channels Output (FTMx_OUTINIT)

Addresses: FTM0_OUTINIT is 4003_8000h base + 5Ch offset = 4003_805Ch
FTM1_OUTINIT is 4003_9000h base + 5Ch offset = 4003_905Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								CH7OI	CH6OI	CH5OI	CH4OI	CH3OI	CH2OI	CH1OI	CH0OI
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_OUTINIT field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 CH7OI	Channel 7 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
6 CH6OI	Channel 6 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
5 CH5OI	Channel 5 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
4 CH4OI	Channel 4 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
3 CH3OI	Channel 3 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
2 CH2OI	Channel 2 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs.

Table continues on the next page...

FTMx_OUTINIT field descriptions (continued)

Field	Description
	0 The initialization value is 0. 1 The initialization value is 1.
1 CH1OI	Channel 1 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.
0 CH0OI	Channel 0 Output Initialization Value Selects the value that is forced into the channel output when the initialization occurs. 0 The initialization value is 0. 1 The initialization value is 1.

35.3.13 Output Mask (FTMx_OUTMASK)

This register provides a mask for each FTM channel. The mask of a channel determines if its output responds, that is, it is masked or not, when a match occurs. This feature is used for BLDC control where the PWM signal is presented to an electric motor at specific times to provide electronic commutation.

Any write to the OUTMASK register, stores the value in its write buffer. The register is updated with the value of its write buffer according to PWM synchronization.

Addresses: FTM0_OUTMASK is 4003_8000h base + 60h offset = 4003_8060h
FTM1_OUTMASK is 4003_9000h base + 60h offset = 4003_9060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								CH7OM	CH6OM	CH5OM	CH4OM	CH3OM	CH2OM	CH1OM	CH0OM
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_OUTMASK field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 CH7OM	Channel 7 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.

Table continues on the next page...

FTMx_OUTMASK field descriptions (continued)

Field	Description
6 CH6OM	Channel 6 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
5 CH5OM	Channel 5 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
4 CH4OM	Channel 4 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
3 CH3OM	Channel 3 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
2 CH2OM	Channel 2 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
1 CH1OM	Channel 1 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.
0 CH0OM	Channel 0 Output Mask Defines if the channel output is masked or unmasked. 0 Channel output is not masked. It continues to operate normally. 1 Channel output is masked. It is forced to its inactive state.

35.3.14 Function For Linked Channels (FTMx_COMBINE)

This register contains the control bits used to configure the fault control, synchronization, deadtime insertion, Dual Edge Capture mode, Complementary, and Combine mode for each pair of channels (n) and (n+1), where n equals 0, 2, 4, and 6.

Addresses: FTM0_COMBINE is 4003_8000h base + 64h offset = 4003_8064h
FTM1_COMBINE is 4003_9000h base + 64h offset = 4003_9064h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	FAULTEN3	SYNCEN3	DTEN3	DECAP3	DECAPEN3	COMP3	COMBINE3	0	FAULTEN2	SYNCEN2	DTEN2	DECAP2	DECAPEN2	COMP2	COMBINE2
W	▨								▨							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	FAULTEN1	SYNCEN1	DTEN1	DECAP1	DECAPEN1	COMP1	COMBINE1	0	FAULTEN0	SYNCEN0	DTEN0	DECAP0	DECAPEN0	COMP0	COMBINE0
W	▨								▨							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_COMBINE field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value zero.
30 FAULTEN3	Fault Control Enable For n = 6 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
29 SYNCEN3	Synchronization Enable For n = 6 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
28 DTEN3	Deadtime Enable For n = 6 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
27 DECAP3	Dual Edge Capture Mode Captures For n = 6 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when FTMEN = 1 and DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
26 DECAPEN3	Dual Edge Capture Mode Enable For n = 6 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 35-7. This field applies only when FTMEN = 1. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
25 COMP3	Complement Of Channel (n) for n = 6 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
24 COMBINE3	Combine Channels For n = 6 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.
23 Reserved	This read-only field is reserved and always has the value zero.
22 FAULTEN2	Fault Control Enable For n = 4 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
21 SYNCEN2	Synchronization Enable For n = 4 Enables PWM synchronization of registers C(n)V and C(n+1)V.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
	0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
20 DTEN2	Deadtime Enable For n = 4 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
19 DECAP2	Dual Edge Capture Mode Captures For n = 4 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when FTMEN = 1 and DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
18 DECAPEN2	Dual Edge Capture Mode Enable For n = 4 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 35-7. This field applies only when FTMEN = 1. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
17 COMP2	Complement Of Channel (n) For n = 4 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
16 COMBINE2	Combine Channels For n = 4 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.
15 Reserved	This read-only field is reserved and always has the value zero.
14 FAULTEN1	Fault Control Enable For n = 2 Enables the fault control in channels (n) and (n+1).

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
	This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
13 SYNCEN1	Synchronization Enable For n = 2 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
12 DTEN1	Deadtime Enable For n = 2 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
11 DECAP1	Dual Edge Capture Mode Captures For n = 2 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when FTMEN = 1 and DECAPEN = 1. DECAP bit is cleared automatically by hardware if Dual Edge Capture – One-Shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
10 DECAPEN1	Dual Edge Capture Mode Enable For n = 2 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 35-7. This field applies only when FTMEN = 1. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
9 COMP1	Complement Of Channel (n) For n = 2 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
8 COMBINE1	Combine Channels For n = 2 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
	0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.
7 Reserved	This read-only field is reserved and always has the value zero.
6 FAULTEN0	Fault Control Enable For n = 0 Enables the fault control in channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault control in this pair of channels is disabled. 1 The fault control in this pair of channels is enabled.
5 SYNCEN0	Synchronization Enable For n = 0 Enables PWM synchronization of registers C(n)V and C(n+1)V. 0 The PWM synchronization in this pair of channels is disabled. 1 The PWM synchronization in this pair of channels is enabled.
4 DTEN0	Deadtime Enable For n = 0 Enables the deadtime insertion in the channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The deadtime insertion in this pair of channels is disabled. 1 The deadtime insertion in this pair of channels is enabled.
3 DECAP0	Dual Edge Capture Mode Captures For n = 0 Enables the capture of the FTM counter value according to the channel (n) input event and the configuration of the dual edge capture bits. This field applies only when FTMEN = 1 and DECAPEN = 1. DECAP bit is cleared automatically by hardware if dual edge capture – one-shot mode is selected and when the capture of channel (n+1) event is made. 0 The dual edge captures are inactive. 1 The dual edge captures are active.
2 DECAPEN0	Dual Edge Capture Mode Enable For n = 0 Enables the Dual Edge Capture mode in the channels (n) and (n+1). This bit reconfigures the function of MSnA, ELSnB:ELSnA and ELS(n+1)B:ELS(n+1)A bits in Dual Edge Capture mode according to Table 35-7. This field applies only when FTMEN = 1. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The Dual Edge Capture mode in this pair of channels is disabled. 1 The Dual Edge Capture mode in this pair of channels is enabled.
1 COMP0	Complement Of Channel (n) For n = 0 Enables Complementary mode for the combined channels. In Complementary mode the channel (n+1) output is the inverse of the channel (n) output. This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

FTMx_COMBINE field descriptions (continued)

Field	Description
	0 The channel (n+1) output is the same as the channel (n) output. 1 The channel (n+1) output is the complement of the channel (n) output.
0 COMBINE0	Combine Channels For n = 0 Enables the combine feature for channels (n) and (n+1). This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Channels (n) and (n+1) are independent. 1 Channels (n) and (n+1) are combined.

35.3.15 Deadtime Insertion Control (FTMx_DEADTIME)

This register selects the deadtime prescaler factor and deadtime value. All FTM channels use this clock prescaler and this deadtime value for the deadtime insertion.

Addresses: FTM0_DEADTIME is 4003_8000h base + 68h offset = 4003_8068h
FTM1_DEADTIME is 4003_9000h base + 68h offset = 4003_9068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								DTPS		DTVAL					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_DEADTIME field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7–6 DTPS	Deadtime Prescaler Value Selects the division factor of the system clock. This prescaled clock is used by the deadtime counter. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0x Divide the system clock by 1. 10 Divide the system clock by 4. 11 Divide the system clock by 16.
5–0 DTVAL	Deadtime Value Selects the deadtime insertion value for the deadtime counter. The deadtime counter is clocked by a scaled version of the system clock. See the description of DTPS. Deadtime insert value = (DTPS × DTVAL). DTVAL selects the number of deadtime counts inserted as follows: When DTVAL is 0, no counts are inserted. When DTVAL is 1, 1 count is inserted.

Table continues on the next page...

FTMx_DEADTIME field descriptions (continued)

Field	Description
	When DTVAL is 2, 2 counts are inserted. This pattern continues up to a possible 63 counts. This field is write protected. It can be written only when MODE[WPDIS] = 1.

35.3.16 FTM External Trigger (FTMx_EXTTRIG)

This register:

- Indicates when a channel trigger was generated
- Enables the generation of a trigger when the FTM counter is equal to its initial
- Selects which channels are used in the generation of the channel triggers

Several channels can be selected to generate multiple triggers in one PWM period.

Channels 6 and 7 are not used to generate channel triggers.

Addresses: FTM0_EXTTRIG is 4003_8000h base + 6Ch offset = 4003_806Ch
FTM1_EXTTRIG is 4003_9000h base + 6Ch offset = 4003_906Ch

Bit	31	30	29	28	27	26	25	24
R W	Reserved[0:16]							
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R W	Reserved[15:0]							
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R W	Reserved[7:0]							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R W	TRIGF	INITTRIGEN	CH1TRIG	CH0TRIG	CH5TRIG	CH4TRIG	CH3TRIG	CH2TRIG
Reset	0	0	0	0	0	0	0	0

FTMx_EXTTRIG field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7 TRIGF	Channel Trigger Flag

Table continues on the next page...

FTMx_EXTTRIG field descriptions (continued)

Field	Description
	Set by hardware when a channel trigger is generated. Clear TRIGF by reading EXTTRIG while TRIGF is set and then writing a 0 to TRIGF. Writing a 1 to TRIGF has no effect. If another channel trigger is generated before the clearing sequence is completed, the sequence is reset so TRIGF remains set after the clear sequence is completed for the earlier TRIGF. 0 No channel trigger was generated. 1 A channel trigger was generated.
6 INITTRIGEN	Initialization Trigger Enable Enables the generation of the trigger when the FTM counter is equal to the CNTIN register. 0 The generation of initialization trigger is disabled. 1 The generation of initialization trigger is enabled.
5 CH1TRIG	Channel 1 Trigger Enable Enable the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
4 CH0TRIG	Channel 0 Trigger Enable Enable the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
3 CH5TRIG	Channel 5 Trigger Enable Enable the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
2 CH4TRIG	Channel 4 Trigger Enable Enable the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
1 CH3TRIG	Channel 3 Trigger Enable Enable the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.
0 CH2TRIG	Channel 2 Trigger Enable Enable the generation of the channel trigger when the FTM counter is equal to the CnV register. 0 The generation of the channel trigger is disabled. 1 The generation of the channel trigger is enabled.

35.3.17 Channels Polarity (FTMx_POL)

This register defines the output polarity of the FTM channels.

NOTE

> The safe value that is driven in a channel output when the fault control is enabled and a fault condition is detected is the inactive state of the channel. That is, the safe value of a channel is the value of its POL bit.

Addresses: FTM0_POL is 4003_8000h base + 70h offset = 4003_8070h
FTM1_POL is 4003_9000h base + 70h offset = 4003_9070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																								POL7	POL6	POL5	POL4	POL3	POL2	POL1	POL0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_POL field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7 POL7	Channel 7 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
6 POL6	Channel 6 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
5 POL5	Channel 5 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
4 POL4	Channel 4 Polarity Defines the polarity of the channel output.

Table continues on the next page...

FTMx_POL field descriptions (continued)

Field	Description
	This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
3 POL3	Channel 3 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
2 POL2	Channel 2 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
1 POL1	Channel 1 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.
0 POL0	Channel 0 Polarity Defines the polarity of the channel output. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The channel polarity is active high. 1 The channel polarity is active low.

35.3.18 Fault Mode Status (FTMx_FMS)

This register contains the fault detection flags, write protection enable bit, and the logic OR of the enabled fault inputs.

Addresses: FTM0_FMS is 4003_8000h base + 74h offset = 4003_8074h
FTM1_FMS is 4003_9000h base + 74h offset = 4003_9074h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	FAULTF	WPEN	FAULTIN	0	FAULTF3	FAULTF2	FAULTF1	FAULTF0
W	0				0	0	0	0
Reset	0	0	0	0	0	0	0	0

FTMx_FMS field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 FAULTF	Fault Detection Flag Represents the logic OR of the individual FAULTFj bits where j = 3, 2, 1, 0. Clear FAULTF by reading the FMS register while FAULTF is set and then writing a 0 to FAULTF while there is no existing fault condition at the enabled fault inputs. Writing a 1 to FAULTF has no effect. If another fault condition is detected in an enabled fault input before the clearing sequence is completed, the sequence is reset so FAULTF remains set after the clearing sequence is completed for the earlier fault condition. FAULTF is also cleared when FAULTFj bits are cleared individually. 0 No fault condition was detected. 1 A fault condition was detected.
6 WPEN	Write Protection Enable The WPEN bit is the negation of the WPDIS bit. WPEN is set when 1 is written to it. WPEN is cleared when WPEN bit is read as a 1 and then 1 is written to WPDIS. Writing 0 to WPEN has no effect.

Table continues on the next page...

FTMx_FMS field descriptions (continued)

Field	Description
	0 Write protection is disabled. Write protected bits can be written. 1 Write protection is enabled. Write protected bits cannot be written.
5 FAULTIN	Fault Inputs Represents the logic OR of the enabled fault inputs after their filter (if their filter is enabled) when fault control is enabled. 0 The logic OR of the enabled fault inputs is 0. 1 The logic OR of the enabled fault inputs is 1.
4 Reserved	This read-only field is reserved and always has the value zero.
3 FAULTF3	Fault Detection Flag 3 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF3 by reading the FMS register while FAULTF3 is set and then writing a 0 to FAULTF3 while there is no existing fault condition at the the corresponding fault input. Writing a 1 to FAULTF3 has no effect. FAULTF3 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF3 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.
2 FAULTF2	Fault Detection Flag 2 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF2 by reading the FMS register while FAULTF2 is set and then writing a 0 to FAULTF2 while there is no existing fault condition at the the corresponding fault input. Writing a 1 to FAULTF2 has no effect. FAULTF2 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF2 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.
1 FAULTF1	Fault Detection Flag 1 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF1 by reading the FMS register while FAULTF1 is set and then writing a 0 to FAULTF1 while there is no existing fault condition at the the corresponding fault input. Writing a 1 to FAULTF1 has no effect. FAULTF1 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF1 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.

Table continues on the next page...

FTMx_FMS field descriptions (continued)

Field	Description
0 FAULTF0	Fault Detection Flag 0 Set by hardware when fault control is enabled, the corresponding fault input is enabled and a fault condition is detected at the fault input. Clear FAULTF0 by reading the FMS register while FAULTF0 is set and then writing a 0 to FAULTF0 while there is no existing fault condition at the the corresponding fault input. Writing a 1 to FAULTF0 has no effect. FAULTF0 bit is also cleared when FAULTF bit is cleared. If another fault condition is detected at the corresponding fault input before the clearing sequence is completed, the sequence is reset so FAULTF0 remains set after the clearing sequence is completed for the earlier fault condition. 0 No fault condition was detected at the fault input. 1 A fault condition was detected at the fault input.

35.3.19 Input Capture Filter Control (FTMx_FILTER)

This register selects the filter value for the inputs of channels.

Channels 4, 5, 6 and 7 do not have an input filter.

NOTE

> Writing to the FILTER register has immediate effect and must be done only when the channels 0, 1, 2, and 3 are not in input modes. Failure to do this could result in a missing valid signal.

Addresses: FTM0_FILTER is 4003_8000h base + 78h offset = 4003_8078h

FTM1_FILTER is 4003_9000h base + 78h offset = 4003_9078h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																CH3FVAL				CH2FVAL				CH1FVAL				CH0FVAL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_FILTER field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–12 CH3FVAL	Channel 3 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.
11–8 CH2FVAL	Channel 2 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.

Table continues on the next page...

FTMx_FILTER field descriptions (continued)

Field	Description
7–4 CH1FVAL	Channel 1 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.
3–0 CH0FVAL	Channel 0 Input Filter Selects the filter value for the channel input. The filter is disabled when the value is zero.

35.3.20 Fault Control (FTMx_FLTCTRL)

This register selects the filter value for the fault inputs, enables the fault inputs and the fault inputs filter.

Addresses: FTM0_FLTCTRL is 4003_8000h base + 7Ch offset = 4003_807Ch
FTM1_FLTCTRL is 4003_9000h base + 7Ch offset = 4003_907Ch

Bit	31	30	29	28	27	26	25	24
R	0							
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W	▨	▨	▨	▨	▨	▨	▨	▨
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0				FFVAL			
W	▨	▨	▨	▨				
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R W	FFLTR3EN	FFLTR2EN	FFLTR1EN	FFLTR0EN	FAULT3EN	FAULT2EN	FAULT1EN	FAULT0EN
Reset	0	0	0	0	0	0	0	0

FTMx_FLTCTRL field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero.
11–8 FFVAL	Fault Input Filter

Table continues on the next page...

FTMx_FLTCTRL field descriptions (continued)

Field	Description
	Selects the filter value for the fault inputs. The fault filter is disabled when the value is zero. **NOTE:** Writing to this field has immediate effect and must be done only when the fault control or all fault inputs are disabled. Failure to do this could result in a missing fault detection.
7 FFLTR3EN	Fault Input 3 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
6 FFLTR2EN	Fault Input 2 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
5 FFLTR1EN	Fault Input 1 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
4 FFLTR0EN	Fault Input 0 Filter Enable Enables the filter for the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input filter is disabled. 1 Fault input filter is enabled.
3 FAULT3EN	Fault Input 3 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input is disabled. 1 Fault input is enabled.
2 FAULT2EN	Fault Input 2 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input is disabled. 1 Fault input is enabled.
1 FAULT1EN	Fault Input 1 Enable Enables the fault input.

Table continues on the next page...

FTMx_FLTCTRL field descriptions (continued)

Field	Description
	This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input is disabled. 1 Fault input is enabled.
0 FAULT0EN	Fault Input 0 Enable Enables the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Fault input is disabled. 1 Fault input is enabled.

35.3.21 Quadrature Decoder Control And Status (FTMx_QDCTRL)

This register has the control and status bits for the Quadrature Decoder mode.

Addresses: FTM0_QDCTRL is 4003_8000h base + 80h offset = 4003_8080h
FTM1_QDCTRL is 4003_9000h base + 80h offset = 4003_9080h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	PHAFLTREN	PHBFLTREN	PHAPOL	PHBPOL	QUADMODE	QUADIR	TOFDIR	QUADEN
W								
Reset	0	0	0	0	0	0	0	0

FTMx_QDCTRL field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 PHAFLTREN	Phase A Input Filter Enable

Table continues on the next page...

FTMx_QDCTRL field descriptions (continued)

Field	Description
	Enables the filter for the quadrature decoder phase A input. The filter value for the phase A input is defined by the CH0FVAL field of FILTER. The phase A filter is also disabled when CH0FVAL is zero. 0 Phase A input filter is disabled. 1 Phase A input filter is enabled.
6 PHBFLTREN	Phase B Input Filter Enable Enables the filter for the quadrature decoder phase B input. The filter value for the phase B input is defined by the CH1FVAL field of FILTER. The phase B filter is also disabled when CH1FVAL is zero. 0 Phase B input filter is disabled. 1 Phase B input filter is enabled.
5 PHAPOL	Phase A Input Polarity Selects the polarity for the quadrature decoder phase A input. 0 Normal polarity. Phase A input signal is not inverted before identifying the rising and falling edges of this signal. 1 Inverted polarity. Phase A input signal is inverted before identifying the rising and falling edges of this signal.
4 PHBPOL	Phase B Input Polarity Selects the polarity for the quadrature decoder phase B input. 0 Normal polarity. Phase B input signal is not inverted before identifying the rising and falling edges of this signal. 1 Inverted polarity. Phase B input signal is inverted before identifying the rising and falling edges of this signal.
3 QUADMODE	Quadrature Decoder Mode Selects the encoding mode used in the Quadrature Decoder mode. 0 Phase A and phase B encoding mode. 1 Count and direction encoding mode.
2 QUADIR	FTM Counter Direction In Quadrature Decoder Mode Indicates the counting direction. 0 Counting direction is decreasing (FTM counter decrement). 1 Counting direction is increasing (FTM counter increment).
1 TOFDIR	Timer Overflow Direction In Quadrature Decoder Mode Indicates if the TOF bit was set on the top or the bottom of counting. 0 TOF bit was set on the bottom of counting. There was an FTM counter decrement and FTM counter changes from its minimum value (CNTIN register) to its maximum value (MOD register). 1 TOF bit was set on the top of counting. There was an FTM counter increment and FTM counter changes from its maximum value (MOD register) to its minimum value (CNTIN register).
0 QUADEN	Quadrature Decoder Mode Enable

Table continues on the next page...

FTMx_QDCTRL field descriptions (continued)

Field	Description
	Enables the Quadrature Decoder mode. In this mode, the phase A and B input signals control the FTM counter direction. The Quadrature Decoder mode has precedence over the other modes. See Table 35-7. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 Quadrature Decoder mode is disabled. 1 Quadrature Decoder mode is enabled.

35.3.22 Configuration (FTMx_CONF)

This register selects the number of times that the FTM counter overflow should occur before the TOF bit to be set, the FTM behavior in BDM modes, the use of an external global time base, and the global time base signal generation.

Addresses: FTM0_CONF is 4003_8000h base + 84h offset = 4003_8084h
FTM1_CONF is 4003_9000h base + 84h offset = 4003_9084h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0
Bit	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
R	0							
W								
Reset	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**
R	0					GTBEOUT	GTBEEN	0
W								
Reset	0	0	0	0	0	0	0	0
Bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R	BDMMODE		0	NUMTOF				
W								
Reset	0	0	0	0	0	0	0	0

FTMx_CONF field descriptions

Field	Description
31–11 Reserved	This read-only field is reserved and always has the value zero.
10 GTBEOUT	Global Time Base Output Enables the global time base signal generation to other FTMs.

Table continues on the next page...

FTMx_CONF field descriptions (continued)

Field	Description
	0 A global time base signal generation is disabled. 1 A global time base signal generation is enabled.
9 GTBEEN	Global Time Base Enable Configures the FTM to use an external global time base signal that is generated by another FTM. 0 Use of an external global time base is disabled. 1 Use of an external global time base is enabled.
8 Reserved	This read-only field is reserved and always has the value zero.
7–6 BDMMODE	BDM Mode Selects the FTM behavior in BDM mode. See BDM mode.
5 Reserved	This read-only field is reserved and always has the value zero.
4–0 NUMTOF	TOF Frequency Selects the ratio between the number of counter overflows to the number of times the TOF bit is set. NUMTOF = 0: The TOF bit is set for each counter overflow. NUMTOF = 1: The TOF bit is set for the first counter overflow but not for the next overflow. NUMTOF = 2: The TOF bit is set for the first counter overflow but not for the next 2 overflows. NUMTOF = 3: The TOF bit is set for the first counter overflow but not for the next 3 overflows. This pattern continues up to a maximum of 31.

35.3.23 FTM Fault Input Polarity (FTMx_FLTPOL)

This register defines the fault inputs polarity.

Addresses: FTM0_FLTPOL is 4003_8000h base + 88h offset = 4003_8088h
FTM1_FLTPOL is 4003_9000h base + 88h offset = 4003_9088h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	0				FLT3POL	FLT2POL	FLT1POL	FLT0POL
W								
Reset	0	0	0	0	0	0	0	0

FTMx_FLTPOL field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero.
3 FLT3POL	Fault Input 3 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault input polarity is active high. A one at the fault input indicates a fault. 1 The fault input polarity is active low. A zero at the fault input indicates a fault.
2 FLT2POL	Fault Input 2 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault input polarity is active high. A one at the fault input indicates a fault. 1 The fault input polarity is active low. A zero at the fault input indicates a fault.
1 FLT1POL	Fault Input 1 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1.

Table continues on the next page...

FTMx_FLTPOL field descriptions (continued)

Field	Description
	0 The fault input polarity is active high. A one at the fault input indicates a fault. 1 The fault input polarity is active low. A zero at the fault input indicates a fault.
0 FLT0POL	Fault Input 0 Polarity Defines the polarity of the fault input. This field is write protected. It can be written only when MODE[WPDIS] = 1. 0 The fault input polarity is active high. A one at the fault input indicates a fault. 1 The fault input polarity is active low. A zero at the fault input indicates a fault.

35.3.24 Synchronization Configuration (FTMx_SYNCONF)

This register selects the PWM synchronization configuration, SWOCTRL, INVCTRL and CNTIN registers synchronization, if FTM clears the TRIGj bit, where j = 0, 1, 2, when the hardware trigger j is detected.

Addresses: FTM0_SYNCONF is 4003_8000h base + 8Ch offset = 4003_808Ch
FTM1_SYNCONF is 4003_9000h base + 8Ch offset = 4003_908Ch

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0			HWSOC	HWINVC	HWOM	HWWRBUF	HWRSTCNT
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0			SWSOC	SWINVC	SWOM	SWWRBUF	SWRSTCNT
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	SYNCMODE	0	SWOC	INVC	0	CNTINC	0	HWTRIGMODE
W								
Reset	0	0	0	0	0	0	0	0

FTMx_SYNCONF field descriptions

Field	Description
31–21 Reserved	This read-only field is reserved and always has the value zero.
20 HWSOC	Software output control synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the SWOCTRL register synchronization. 1 A hardware trigger activates the SWOCTRL register synchronization.
19 HWINVC	Inverting control synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the INVCTRL register synchronization. 1 A hardware trigger activates the INVCTRL register synchronization.
18 HWOM	Output mask synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the OUTMASK register synchronization. 1 A hardware trigger activates the OUTMASK register synchronization.
17 HWWRBUF	MOD, CNTIN, and CV registers synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate MOD, CNTIN, and CV registers synchronization. 1 A hardware trigger activates MOD, CNTIN, and CV registers synchronization.
16 HWRSTCNT	FTM counter synchronization is activated by a hardware trigger. 0 A hardware trigger does not activate the FTM counter synchronization. 1 A hardware trigger activates the FTM counter synchronization.
15–13 Reserved	This read-only field is reserved and always has the value zero.
12 SWSOC	Software output control synchronization is activated by the software trigger. 0 The software trigger does not activate the SWOCTRL register synchronization. 1 The software trigger activates the SWOCTRL register synchronization.
11 SWINVC	Inverting control synchronization is activated by the software trigger. 0 The software trigger does not activate the INVCTRL register synchronization. 1 The software trigger activates the INVCTRL register synchronization.
10 SWOM	Output mask synchronization is activated by the software trigger. 0 The software trigger does not activate the OUTMASK register synchronization. 1 The software trigger activates the OUTMASK register synchronization.
9 SWWRBUF	MOD, CNTIN, and CV registers synchronization is activated by the software trigger. 0 The software trigger does not activate MOD, CNTIN, and CV registers synchronization. 1 The software trigger activates MOD, CNTIN, and CV registers synchronization.
8 SWRSTCNT	FTM counter synchronization is activated by the software trigger. 0 The software trigger does not activate the FTM counter synchronization. 1 The software trigger activates the FTM counter synchronization.
7 SYNCMODE	Synchronization Mode Selects the PWM Synchronization mode.

Table continues on the next page...

FTMx_SYNCONF field descriptions (continued)

Field	Description
	0 Legacy PWM synchronization is selected. 1 Enhanced PWM synchronization is selected.
6 Reserved	This read-only field is reserved and always has the value zero.
5 SWOC	SWOCTRL Register Synchronization 0 SWOCTRL register is updated with its buffer value at all rising edges of system clock. 1 SWOCTRL register is updated with its buffer value by the PWM synchronization.
4 INVC	INVCTRL Register Synchronization 0 INVCTRL register is updated with its buffer value at all rising edges of system clock. 1 INVCTRL register is updated with its buffer value by the PWM synchronization.
3 Reserved	This read-only field is reserved and always has the value zero.
2 CNTINC	CNTIN Register Synchronization 0 CNTIN register is updated with its buffer value at all rising edges of system clock. 1 CNTIN register is updated with its buffer value by the PWM synchronization.
1 Reserved	This read-only field is reserved and always has the value zero.
0 HWTRIGMODE	Hardware Trigger Mode 0 FTM clears the TRIGj bit when the hardware trigger j is detected. 1 FTM does not clear the TRIGj bit when the hardware trigger j is detected.

35.3.25 FTM Inverting Control (FTMx_INVCTRL)

This register controls when the channel (n) output becomes the channel (n+1) output, and channel (n+1) output becomes the channel (n) output. Each INVmEN bit enables the inverting operation for the corresponding pair channels m.

This register has a write buffer. The INVmEN bit is updated by the INVCTRL register synchronization.

Addresses: FTM0_INVCTRL is 4003_8000h base + 90h offset = 4003_8090h
FTM1_INVCTRL is 4003_9000h base + 90h offset = 4003_9090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																												INV3EN	INV2EN	INV1EN	INV0EN
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FTMx_INVCTRL field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value zero.
3 INV3EN	Pair Channels 3 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.
2 INV2EN	Pair Channels 2 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.
1 INV1EN	Pair Channels 1 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.
0 INV0EN	Pair Channels 0 Inverting Enable 0 Inverting is disabled. 1 Inverting is enabled.

35.3.26 FTM Software Output Control (FTMx_SWOCTRL)

This register enables software control of channel (n) output and defines the value forced to the channel (n) output:

- The CHnOC bits enable the control of the corresponding channel (n) output by software.
- The CHnOCV bits select the value that is forced at the corresponding channel (n) output.

This register has a write buffer. The fields are updated by the SWOCTRL register synchronization.

Addresses: FTM0_SWOCTRL is 4003_8000h base + 94h offset = 4003_8094h
FTM1_SWOCTRL is 4003_9000h base + 94h offset = 4003_9094h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R W	CH7OCV	CH6OCV	CH5OCV	CH4OCV	CH3OCV	CH2OCV	CH1OCV	CH0OCV
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R W	CH7OC	CH6OC	CH5OC	CH4OC	CH3OC	CH2OC	CH1OC	CH0OC
Reset	0	0	0	0	0	0	0	0

FTMx_SWOCTRL field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15 CH7OCV	Channel 7 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
14 CH6OCV	Channel 6 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
13 CH5OCV	Channel 5 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
12 CH4OCV	Channel 4 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
11 CH3OCV	Channel 3 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
10 CH2OCV	Channel 2 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.

Table continues on the next page...

FTMx_SWOCTRL field descriptions (continued)

Field	Description
9 CH1OCV	Channel 1 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
8 CH0OCV	Channel 0 Software Output Control Value 0 The software output control forces 0 to the channel output. 1 The software output control forces 1 to the channel output.
7 CH7OC	Channel 7 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
6 CH6OC	Channel 6 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
5 CH5OC	Channel 5 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
4 CH4OC	Channel 4 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
3 CH3OC	Channel 3 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
2 CH2OC	Channel 2 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
1 CH1OC	Channel 1 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.
0 CH0OC	Channel 0 Software Output Control Enable 0 The channel output is not affected by software output control. 1 The channel output is affected by software output control.

35.3.27 FTM PWM Load (FTMx_PWMLOAD)

Enables the loading of the MOD, CNTIN, C(n)V, and C(n+1)V registers with the values of their write buffers when the FTM counter changes from the MOD register value to its next value or when a channel (j) match occurs. A match occurs for the channel (j) when FTM counter = C(j)V.

Addresses: FTM0_PWMLOAD is 4003_8000h base + 98h offset = 4003_8098h
FTM1_PWMLOAD is 4003_9000h base + 98h offset = 4003_9098h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0						LDOK	0
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R W	CH7SEL	CH6SEL	CH5SEL	CH4SEL	CH3SEL	CH2SEL	CH1SEL	CH0SEL
Reset	0	0	0	0	0	0	0	0

FTMx_PWMLOAD field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value zero.
9 LDOK	Load Enable Enables the loading of the MOD, CNTIN, and CV registers with the values of their write buffers. 0 Loading updated values is disabled. 1 Loading updated values is enabled.
8 Reserved	This read-only field is reserved and always has the value zero.
7 CH7SEL	Channel 7 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
6 CH6SEL	Channel 6 Select

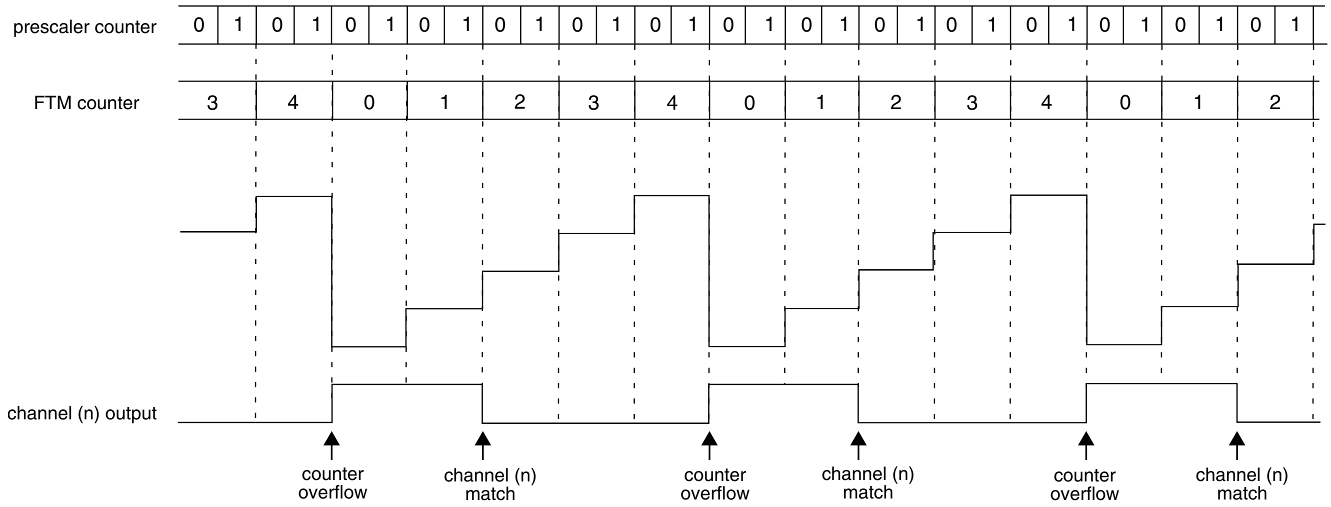
Table continues on the next page...

FTMx_PWMLOAD field descriptions (continued)

Field	Description
	0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
5 CH5SEL	Channel 5 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
4 CH4SEL	Channel 4 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
3 CH3SEL	Channel 3 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
2 CH2SEL	Channel 2 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
1 CH1SEL	Channel 1 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.
0 CH0SEL	Channel 0 Select 0 Do not include the channel in the matching process. 1 Include the channel in the matching process.

35.4 Functional description

The notation used in this document to represent the counters and the generation of the signals is shown in the following figure.



FTM counting is up.
Channel (n) is in high-true EPWM mode.

PS[2:0] = 001
CNTIN = 0x0000
MOD = 0x0004
CnV = 0x0002

prescaler counter
0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
FTM counter
3 4 0 1 2 3 4 0 1 2 3 4 0 1 2
channel (n) output
counter overflow
channel (n) match
counter overflow
channel (n) match
counter overflow
channel (n) match

Figure 35-125. Notation used

35.4.1 Clock source

The FTM has only one clock domain: the system clock..

35.4.1.1 Counter clock source

The CLKS[1:0] bits in the SC register select one of three possible clock sources for the FTM counter or disable the FTM counter. After any MCU reset, CLKS[1:0] = 0:0 so no clock source is selected.

The CLKS[1:0] bits may be read or written at any time. Disabling the FTM counter by writing 0:0 to the CLKS[1:0] bits does not affect the FTM counter value or other registers.

The fixed frequency clock is an alternative clock source for the FTM counter that allows the selection of a clock other than the system clock or an external clock. This clock input is defined by chip integration. Refer to the chip specific documentation for further information. Due to FTM hardware implementation limitations, the frequency of the fixed frequency clock must not exceed 1/2 of the system clock frequency.

The external clock passes through a synchronizer clocked by the system clock to assure that counter transitions are properly aligned to system clock transitions.Therefore, to meet Nyquist criteria considering also jitter, the frequency of the external clock source must not exceed 1/4 of the system clock frequency.

35.4.2 Prescaler

The selected counter clock source passes through a prescaler that is a 7-bit counter. The value of the prescaler is selected by the PS[2:0] bits. The following figure shows an example of the prescaler counter and FTM counter.

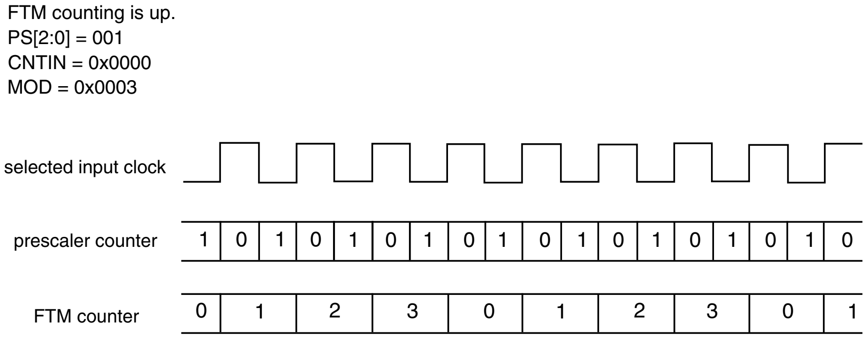


Figure 35-126. Example of the prescaler counter

35.4.3 Counter

The FTM has a 16-bit counter that is used by the channels either for input or output modes. The FTM counter clock is the selected clock divided by the prescaler.

The FTM counter has these modes of operation:

- Up counting
- Up-down counting
- Quadrature Decoder mode

35.4.3.1 Up counting

Up counting is selected when (QUADEN = 0) and (CPWMS = 0).

CNTIN defines the starting value of the count and MOD defines the final value of the count, see the following figure. The value of CNTIN is loaded into the FTM counter, and the counter increments until the value of MOD is reached, at which point the counter is reloaded with the value of CNTIN.

The FTM period when using up counting is (MOD – CNTIN + 0x0001) × period of the FTM counter clock.

The TOF bit is set when the FTM counter changes from MOD to CNTIN.

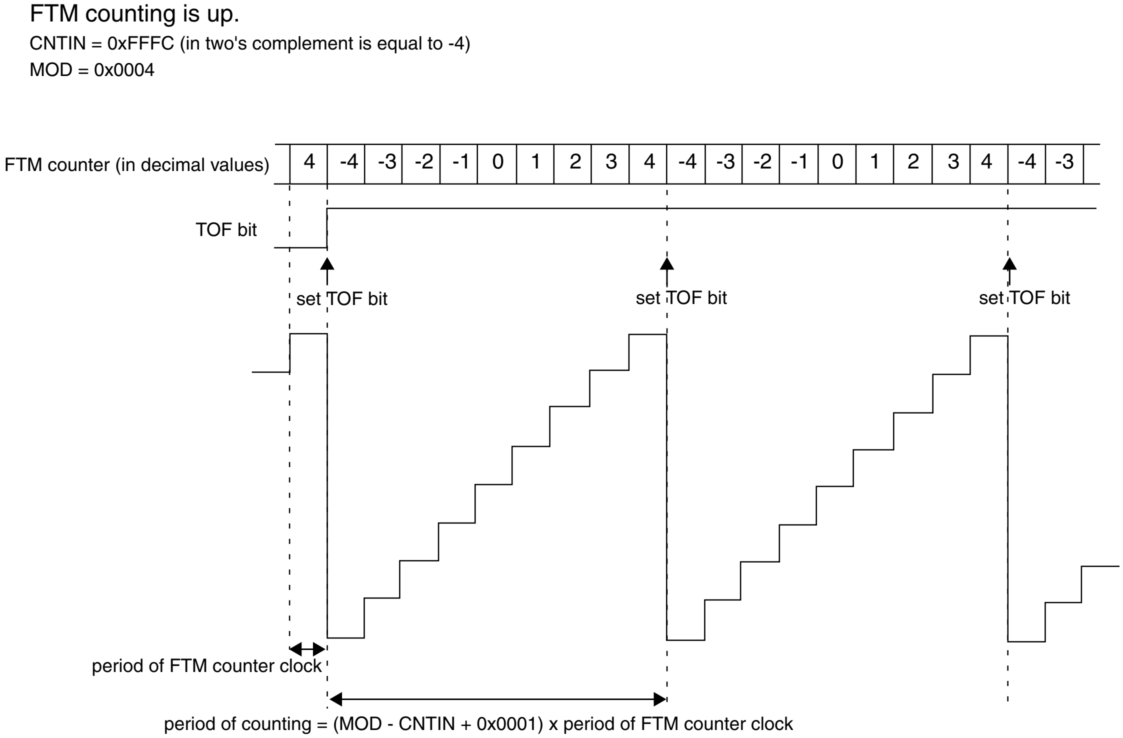

FTM counting is up.
CNTIN = 0xFFFC (in two's complement is equal to -4)
MOD = 0x0004
FTM counter (in decimal values)
4 -4 -3 -2 -1 0 1 2 3 4 -4 -3 -2 -1 0 1 2 3 4 -4 -3
TOF bit
set TOF bit
set TOF bit
set TOF bit
period of FTM counter clock
period of counting = (MOD - CNTIN + 0x0001) x period of FTM counter clock

Figure 35-127. Example of FTM up and signed counting

Table 35-182. FTM counting based on CNTIN value

When	Then
CNTIN = 0x0000	The FTM counting is equivalent to TPM up counting, that is, up and unsigned counting. See the following figure.
CNTIN[15] = 1	The initial value of the FTM counter is a negative number in two's complement, so the FTM counting is up and signed.
CNTIN[15] = 0 and CNTIN ≠ 0x0000	The initial value of the FTM counter is a positive number, so the FTM counting is up and unsigned.

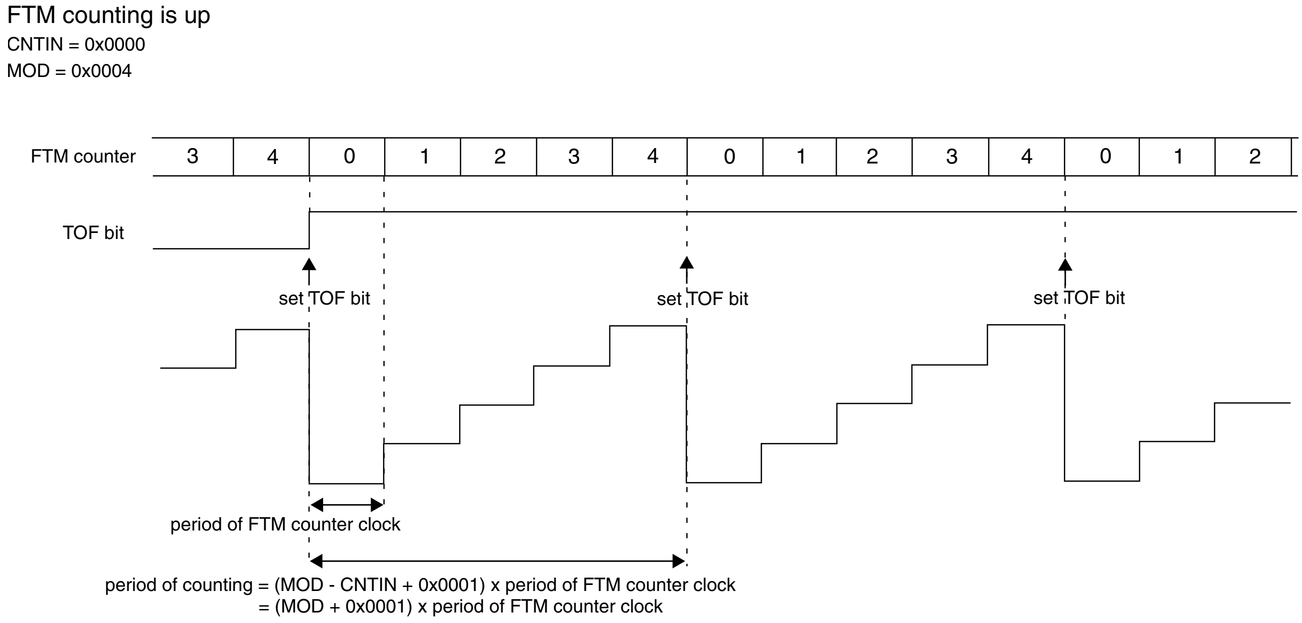

FTM counting is up
CNTIN = 0x0000
MOD = 0x0004
FTM counter
3
4
0
1
2
3
4
0
1
2
3
4
0
1
2
TOF bit
set TOF bit
set TOF bit
set TOF bit
period of FTM counter clock
period of counting = (MOD - CNTIN + 0x0001) x period of FTM counter clock
= (MOD + 0x0001) x period of FTM counter clock

Figure 35-128. Example of FTM up counting with CNTIN = 0x0000

Note

- FTM operation is only valid when the value of the CNTIN register is less than the value of the MOD register, either in the unsigned counting or signed counting. It is the responsibility of the software to ensure that the values in the CNTIN and MOD registers meet this requirement. Any values of CNTIN and MOD that do not satisfy this criteria can result in unpredictable behavior.
- MOD = CNTIN is a redundant condition. In this case, the FTM counter is always equal to MOD and the TOF bit is set in each rising edge of the FTM counter clock.
- When MOD = 0x0000, CNTIN = 0x0000, for example after reset, and FTMEN = 1, the FTM counter remains stopped at 0x0000 until a non-zero value is written into the MOD or CNTIN registers.
- Setting CNTIN to be greater than the value of MOD is not recommended as this unusual setting may make the FTM operation difficult to comprehend. However, there is no restriction on this configuration, and an example is shown in the following figure.

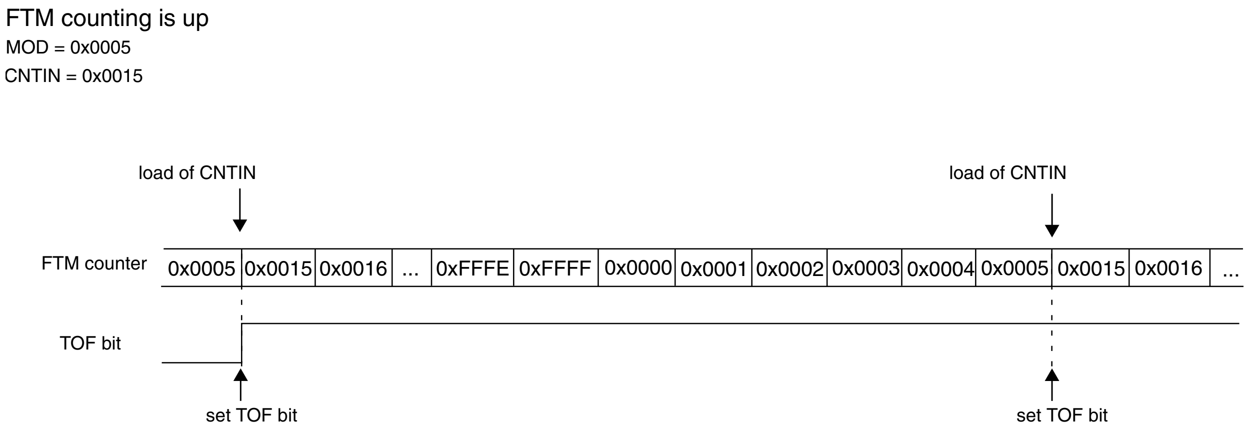

FTM counting is up
MOD = 0x0005
CNTIN = 0x0015
load of CNTIN
load of CNTIN
FTM counter
0x0005
0x0015
0x0016
...
0xFFFE
0xFFFF
0x0000
0x0001
0x0002
0x0003
0x0004
0x0005
0x0015
0x0016
...
TOF bit
set TOF bit
set TOF bit

Figure 35-129. Example of up counting when the value of CNTIN is greater than the value of MOD

35.4.3.2 Up-down counting

Up-down counting is selected when (QUADEN= 0) and (CPWMS = 1).

CNTIN defines the starting value of the count and MOD defines the final value of the count. The value of CNTIN is loaded into the FTM counter, and the counter increments until the value of MOD is reached, at which point the counter is decremented until it returns to the value of CNTIN and the up-down counting restarts.

The FTM period when using up-down counting is 2 × (MOD – CNTIN) × period of the FTM counter clock.

The TOF bit is set when the FTM counter changes from MOD to (MOD – 1).

If (CNTIN = 0x0000), the FTM counting is equivalent to TPM up-down counting, that is, up-down and unsigned counting. See the following figure.

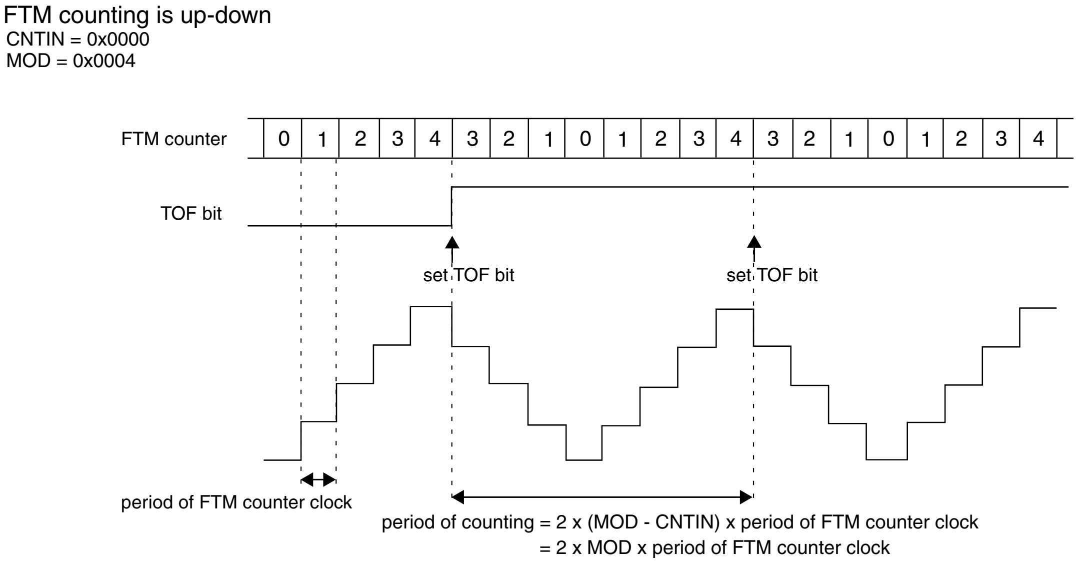

FTM counting is up-down
CNTIN = 0x0000
MOD = 0x0004
FTM counter
0 1 2 3 4 3 2 1 0 1 2 3 4 3 2 1 0 1 2 3 4
TOF bit
set TOF bit
set TOF bit
period of FTM counter clock
period of counting = 2 x (MOD - CNTIN) x period of FTM counter clock
= 2 x MOD x period of FTM counter clock

Figure 35-130. Example of up-down counting when CNTIN = 0x0000

Note

It is expected that the up-down counting be used only with CNTIN = 0x0000.

35.4.3.3 Free running counter

If (FTMEN = 0) and (MOD = 0x0000 or MOD = 0xFFFF), the FTM counter is a free running counter. In this case, the FTM counter runs free from 0x0000 through 0xFFFF and the TOF bit is set when the FTM counter changes from 0xFFFF to 0x0000. See the following figure..

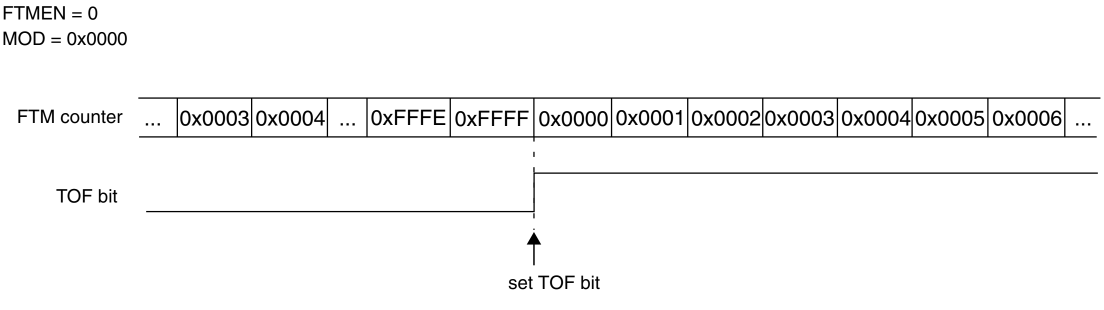

FTMEN = 0
MOD = 0x0000
FTM counter
... 0x0003 0x0004 ... 0xFFFE 0xFFFF 0x0000 0x0001 0x0002 0x0003 0x0004 0x0005 0x0006 ...
TOF bit
set TOF bit

Figure 35-131. Example when the FTM counter is free running

If (FTMEN = 1), (QUADEN = 0), (CPWMS = 0), (CNTIN = 0x0000), and (MOD = 0xFFFF), the FTM counter is a free running counter. In this case, the FTM counter runs free from 0x0000 through 0xFFFF and the TOF bit is set when the FTM counter changes from 0xFFFF to 0x0000.

35.4.3.4 Counter reset

Any write to CNT resets the FTM counter to the value in the CNTIN register and the channels output to its initial value, except for channels in Output Compare mode.

The FTM counter synchronization can also be used to force the value of CNTIN into the FTM counter and the channels output to its initial value, except for channels in Output Compare mode.

35.4.3.5 When the TOF bit is set

The NUMTOF[4:0] bits define the number of times that the FTM counter overflow should occur before the TOF bit to be set. If NUMTOF[4:0] = 0x00, then the TOF bit is set at each FTM counter overflow.

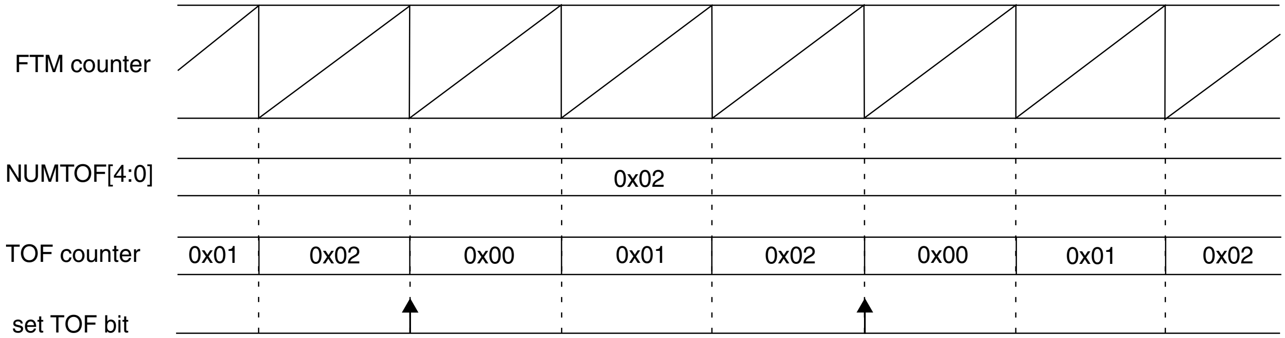

FTM counter
NUMTOF[4:0]
0x02
TOF counter
0x01
0x02
0x00
0x01
0x02
0x00
0x01
0x02
set TOF bit

Figure 35-132. Periodic TOF when NUMTOF = 0x02

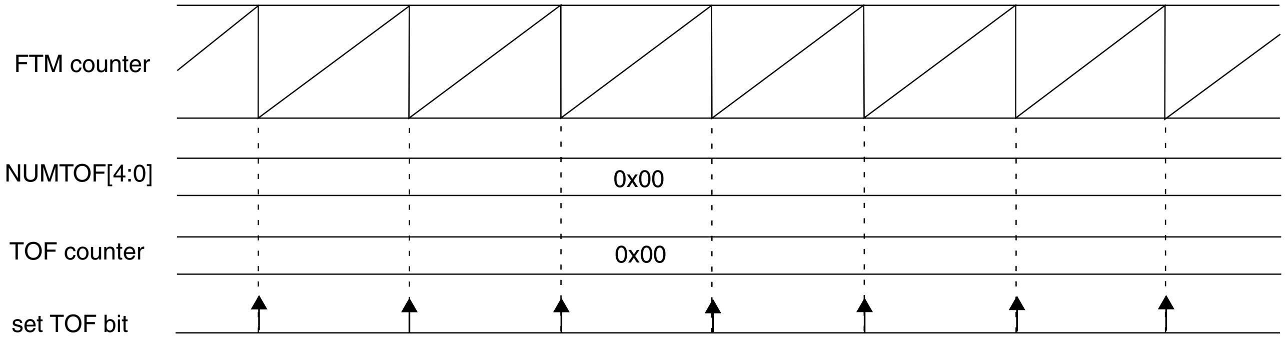

FTM counter
NUMTOF[4:0]
0x00
TOF counter
0x00
set TOF bit

Figure 35-133. Periodic TOF when NUMTOF = 0x00

35.4.4 Input Capture mode

The Input Capture mode is selected when:

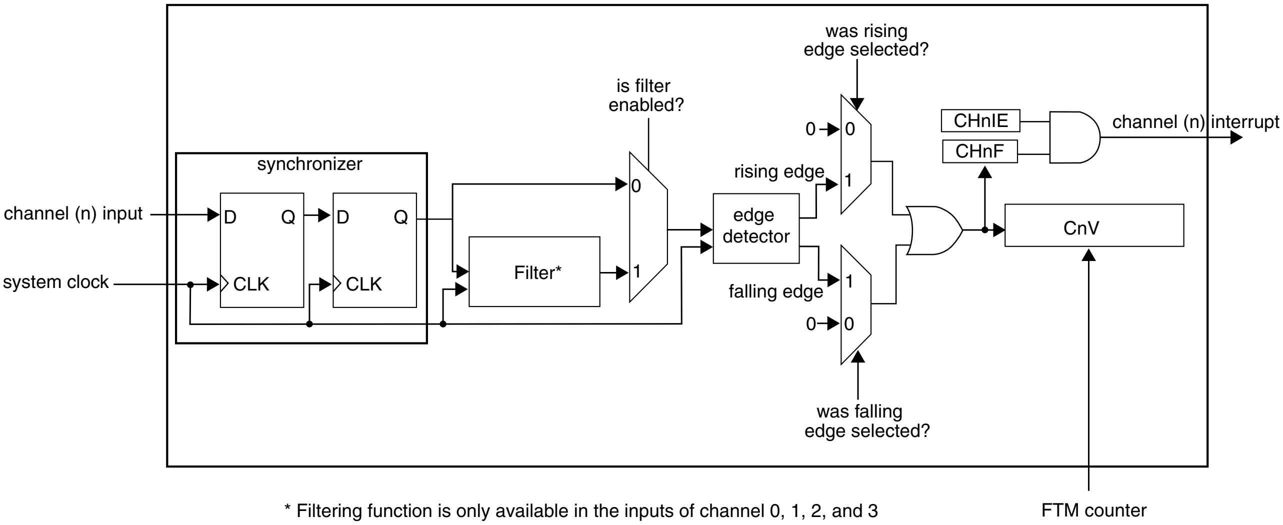
- DECAPEN = 0
- COMBINE = 0
- CPWMS = 0
- MSnB:MSnA = 0:0
- ELSnB:ELSnA ≠ 0:0

When a selected edge occurs on the channel input, the current value of the FTM counter is captured into the CnV register, at the same time the CHnF bit is set and the channel interrupt is generated if enabled by CHnIE = 1. See the following figure.

When a channel is configured for input capture, the FTMxCHn pin is an edge-sensitive input. ELSnB:ELSnA control bits determine which edge, falling or rising, triggers input-capture event. Note that the maximum frequency for the channel input signal to be detected correctly is system clock divided by 4, which is required to meet Nyquist criteria for signal sampling.

Writes to the CnV register is ignored in Input Capture mode.

While in BDM, the input capture function works as configured. When a selected edge event occurs, the FTM counter value, which is frozen because of BDM, is captured into the CnV register and the CHnF bit is set.


was rising
edge selected?
is filter
enabled?
0
0
1
CHnIE
CHnF
channel (n) interrupt
synchronizer
rising edge
channel (n) input
D
Q
D
Q
0
edge
detector
CnV
system clock
CLK
CLK
Filter*
1
1
falling edge
0
0
was falling
edge selected?
FTM counter

* Filtering function is only available in the inputs of channel 0, 1, 2, and 3

Figure 35-134. Input Capture mode

If the channel input does not have a filter enabled, then the input signal is always delayed 3 rising edges of the system clock, that is, two rising edges to the synchronizer plus one more rising edge to the edge detector. In other words, the CHnF bit is set on the third rising edge of the system clock after a valid edge occurs on the channel input.

Note

The Input Capture mode must be used only with CNTIN = 0x0000.

35.4.4.1 Filter for Input Capture mode

The filter function is only available on channels 0, 1, 2, and 3.

First, the input signal is synchronized by the system clock. Following synchronization, the input signal enters the filter block. See the following figure.

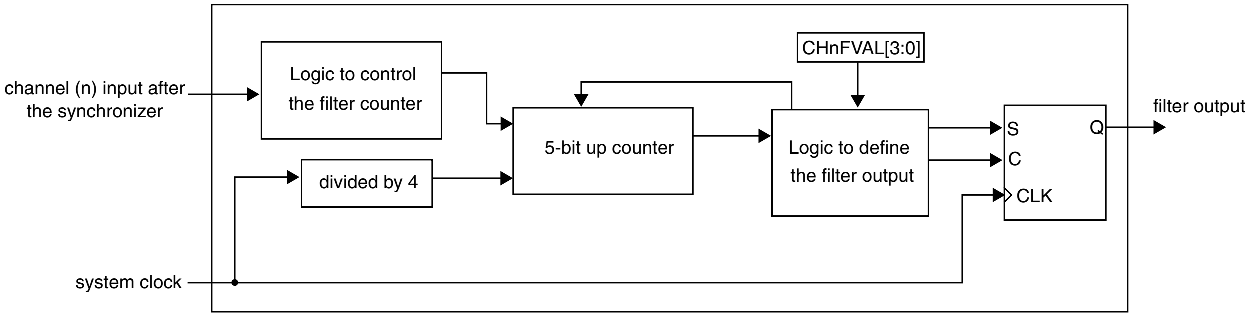

channel (n) input after the synchronizer
Logic to control the filter counter
divided by 4
5-bit up counter
CHnFVAL[3:0]
Logic to define the filter output
S
Q
C
CLK
filter output
system clock

Figure 35-135. Channel input filter

When there is a state change in the input signal, the 5-bit counter is reset and starts counting up. As long as the new state is stable on the input, the counter continues to increment. If the 5-bit counter overflows, that is, the counter exceeds the value of CHnFVAL[3:0], the state change of the input signal is validated. It is then transmitted as a pulse edge to the edge detector.

If the opposite edge appears on the input signal before it can be validated, the counter is reset. At the next input transition, the counter starts counting again. Any pulse that is shorter than the minimum value selected by CHnFVAL[3:0] (× 4 system clocks) is regarded as a glitch and is not passed on to the edge detector. A timing diagram of the input filter is shown in the following figure.

The filter function is disabled when CHnFVAL[3:0] bits are zero. In this case, the input signal is delayed 3 rising edges of the system clock. If (CHnFVAL[3:0] ≠ 0000), then the input signal is delayed by the minimum pulse width (CHnFVAL[3:0] × 4 system clocks) plus a further 4 rising edges of the system clock: two rising edges to the synchronizer, one rising edge to the filter output, plus one more to the edge detector. In other words, CHnF is set (4 + 4 × CHnFVAL[3:0]) system clock periods after a valid edge occurs on the channel input.

The clock for the 5-bit counter in the channel input filter is the system clock divided by 4.

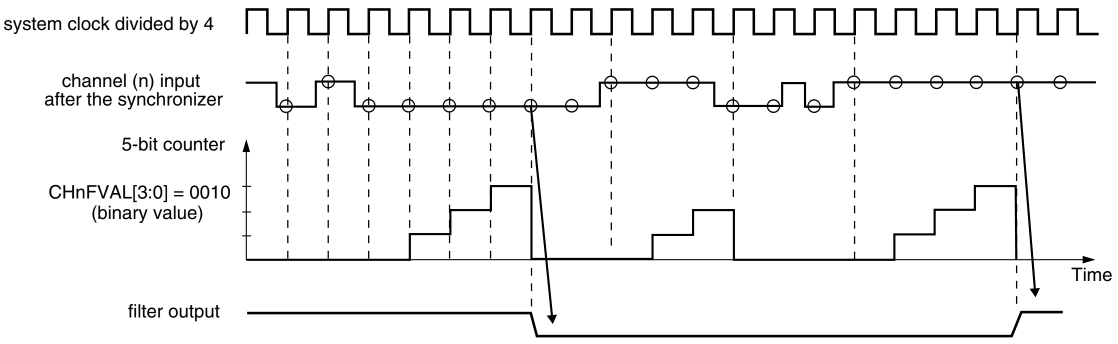

system clock divided by 4
channel (n) input after the synchronizer
5-bit counter
CHnFVAL[3:0] = 0010 (binary value)
Time
filter output

Figure 35-136. Channel input filter example

35.4.5 Output Compare mode

The Output Compare mode is selected when:

- DECAPEN = 0
- COMBINE = 0
- CPWMS = 0, and
- MSnB:MSnA = 0:1

In Output Compare mode, the FTM can generate timed pulses with programmable position, polarity, duration, and frequency. When the counter matches the value in the CnV register of an output compare channel, the channel (n) output can be set, cleared, or toggled.

When a channel is initially configured to Toggle mode, the previous value of the channel output is held until the first output compare event occurs.

The CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1 at the channel (n) match (FTM counter = CnV).

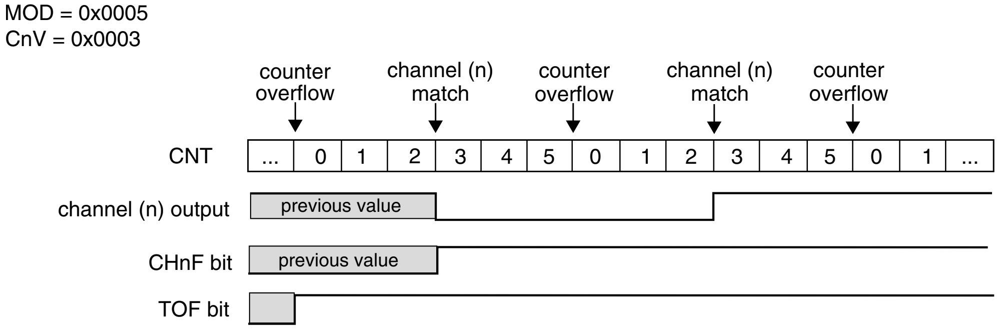

MOD = 0x0005
CnV = 0x0003
counter overflow
channel (n) match
counter overflow
channel (n) match
counter overflow
CNT
...
0
1
2
3
4
5
0
1
2
3
4
5
0
1
...
channel (n) output
previous value
CHnF bit
previous value
TOF bit

Figure 35-137. Example of the Output Compare mode when the match toggles the channel output

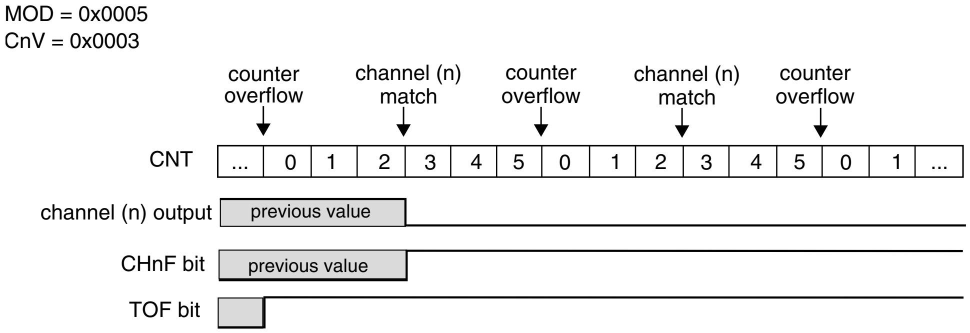

MOD = 0x0005
CnV = 0x0003
counter overflow
channel (n) match
counter overflow
channel (n) match
counter overflow
CNT
... 0 1 2 3 4 5 0 1 2 3 4 5 0 1 ...
channel (n) output
previous value
CHnF bit
previous value
TOF bit

Figure 35-138. Example of the Output Compare mode when the match clears the channel output

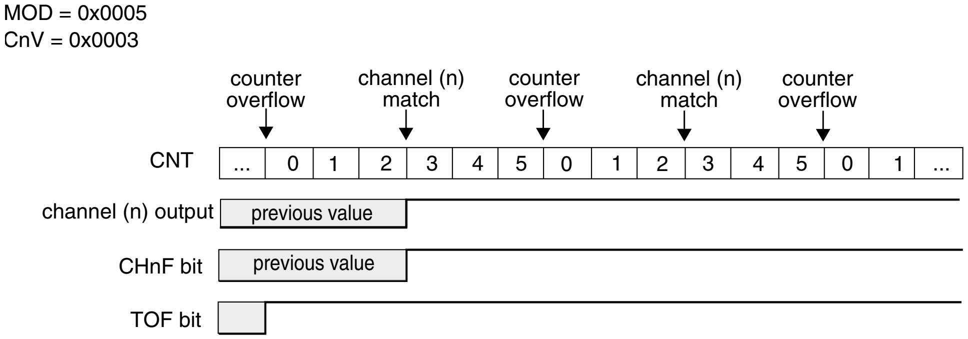

MOD = 0x0005
CnV = 0x0003
counter overflow
channel (n) match
counter overflow
channel (n) match
counter overflow
CNT
... 0 1 2 3 4 5 0 1 2 3 4 5 0 1 ...
channel (n) output
previous value
CHnF bit
previous value
TOF bit

Figure 35-139. Example of the Output Compare mode when the match sets the channel output

Using the Output Compare mode is possible with (ELSnB:ELSnA = 0:0). In this case, when the counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1, however the channel (n) output is not modified and controlled by FTM.

Note

The Output Compare mode must be used only with CNTIN = 0x0000.

35.4.6 Edge-Aligned PWM (EPWM) mode

The Edge-Aligned mode is selected when:

- QUADEN = 0
- DECAPEN = 0
- COMBINE = 0
- CPWMS = 0, and
- MSnB = 1

The EPWM period is determined by (MOD − CNTIN + 0x0001) and the pulse width (duty cycle) is determined by (CnV − CNTIN).

The CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1 at the channel (n) match (FTM counter = CnV), that is, at the end of the pulse width.

This type of PWM signal is called edge-aligned because the leading edges of all PWM signals are aligned with the beginning of the period, which is the same for all channels within an FTM.

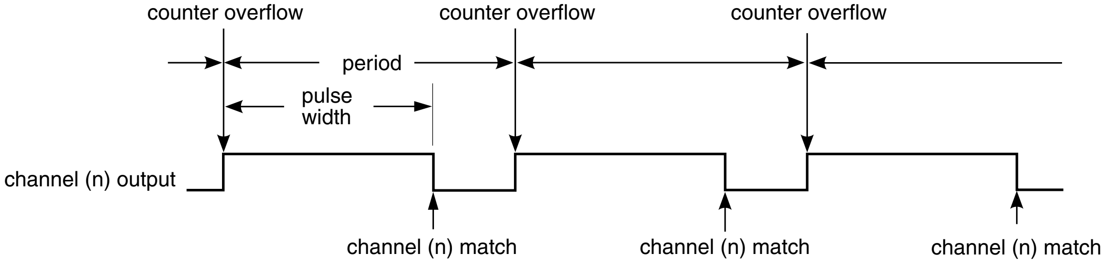

counter overflow
counter overflow
counter overflow
period
pulse width
channel (n) output
channel (n) match
channel (n) match
channel (n) match

Figure 35-140. EPWM period and pulse width with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = 0:0) when the counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated if CHnIE = 1, however the channel (n) output is not controlled by FTM.

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the counter overflow when the CNTIN register value is are loaded into the FTM counter, and it is forced low at the channel (n) match (FTM counter = CnV). See the following figure.

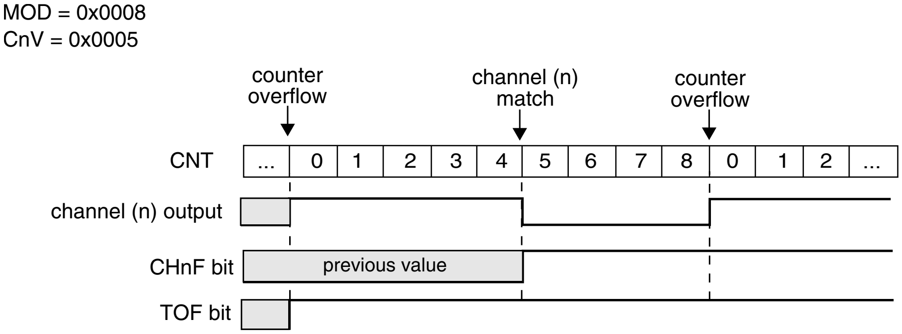

MOD = 0x0008
CnV = 0x0005
counter overflow
channel (n) match
counter overflow
CNT
... 0 1 2 3 4 5 6 7 8 0 1 2 ...
channel (n) output
CHnF bit
previous value
TOF bit

Figure 35-141. EPWM signal with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the counter overflow when the CNTIN register value is loaded into the FTM counter, and it is forced high at the channel (n) match (FTM counter = CnV). See the following figure.

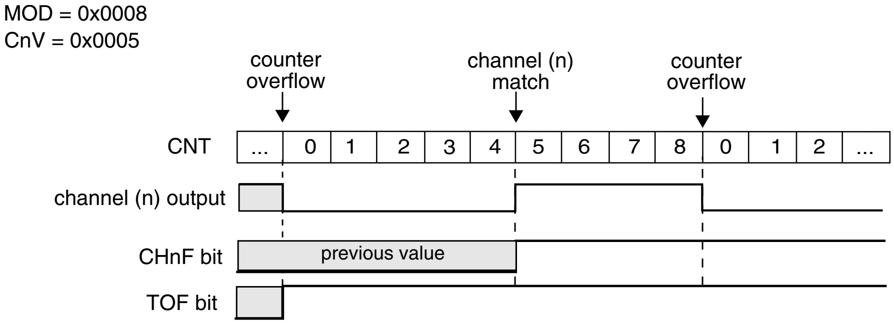


Figure 35-142. EPWM signal with ELSnB:ELSnA = X:1

If (CnV = 0x0000), then the channel (n) output is a 0% duty cycle EPWM signal and CHnF bit is not set even when there is the channel (n) match. If (CnV > MOD), then the channel (n) output is a 100% duty cycle EPWM signal and CHnF bit is not set even when there is the channel (n) match. Therefore, MOD must be less than 0xFFFF in order to get a 100% duty cycle EPWM signal.

Note

The EPWM mode must be used only with CNTIN = 0x0000.

35.4.7 Center-Aligned PWM (CPWM) mode

The Center-Aligned mode is selected when:

- QUADEN = 0
- DECAPEN = 0
- COMBINE = 0, and
- CPWMS = 1

The CPWM pulse width (duty cycle) is determined by 2 × (CnV − CNTIN) and the period is determined by 2 × (MOD − CNTIN). See the following figure. MOD must be kept in the range of 0x0001 to 0x7FFF because values outside this range can produce ambiguous results.

In the CPWM mode, the FTM counter counts up until it reaches MOD and then counts down until it reaches CNTIN.

The CHnF bit is set and channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (FTM counter = CnV) when the FTM counting is down (at the begin of the pulse width) and when the FTM counting is up (at the end of the pulse width).

This type of PWM signal is called center-aligned because the pulse width centers for all channels are aligned with the value of CNTIN.

The other channel modes are not compatible with the up-down counter (CPWMS = 1). Therefore, all FTM channels must be used in CPWM mode when (CPWMS = 1).

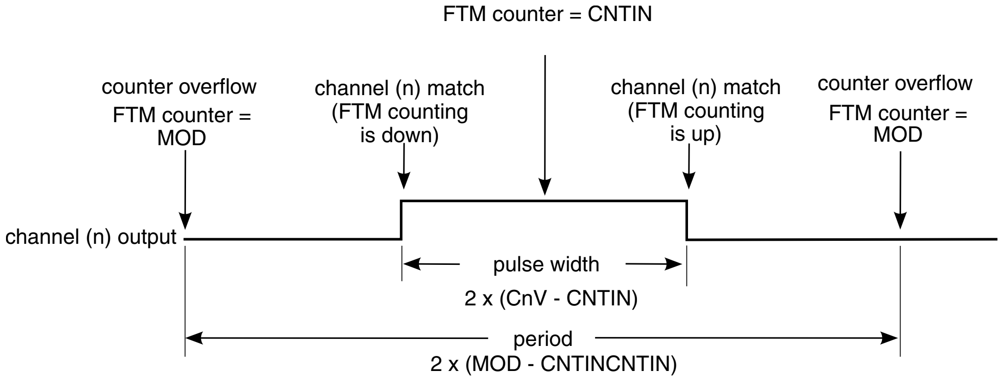

FTM counter = CNTIN
counter overflow
FTM counter = MOD
channel (n) match (FTM counting is down)
channel (n) match (FTM counting is up)
counter overflow
FTM counter = MOD
channel (n) output
pulse width
2 x (CnV - CNTIN)
period
2 x (MOD - CNTINCNTIN)

Figure 35-143. CPWM period and pulse width with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = 0:0) when the FTM counter reaches the value in the CnV register, the CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1), however the channel (n) output is not controlled by FTM.

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced high at the channel (n) match (FTM counter = CnV) when counting down, and it is forced low at the channel (n) match when counting up. See the following figure.

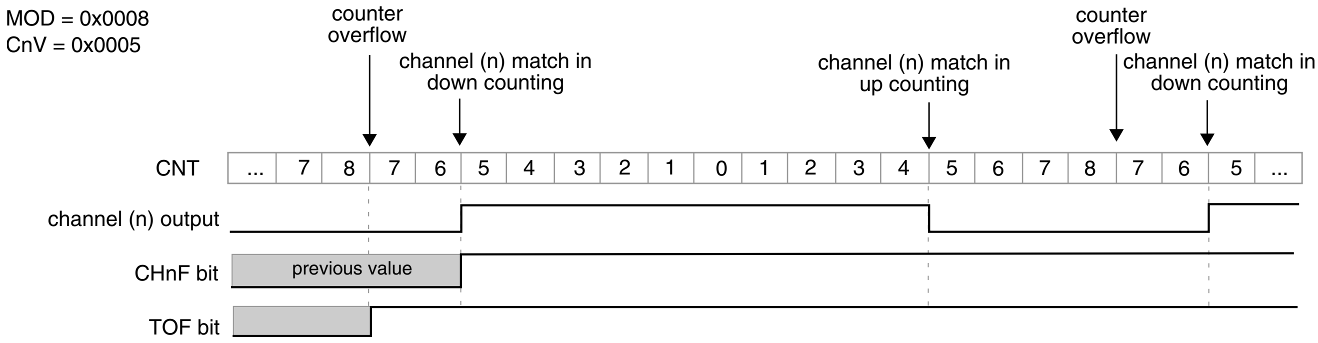

MOD = 0x0008
CnV = 0x0005
counter overflow
channel (n) match in down counting
channel (n) match in up counting
counter overflow
channel (n) match in down counting
CNT
... 7 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 7 6 5 ...
channel (n) output
CHnF bit
previous value
TOF bit

Figure 35-144. CPWM signal with ELSnB:ELSnA = 1:0

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced low at the channel (n) match (FTM counter = CnV) when counting down, and it is forced high at the channel (n) match when counting up. See the following figure.

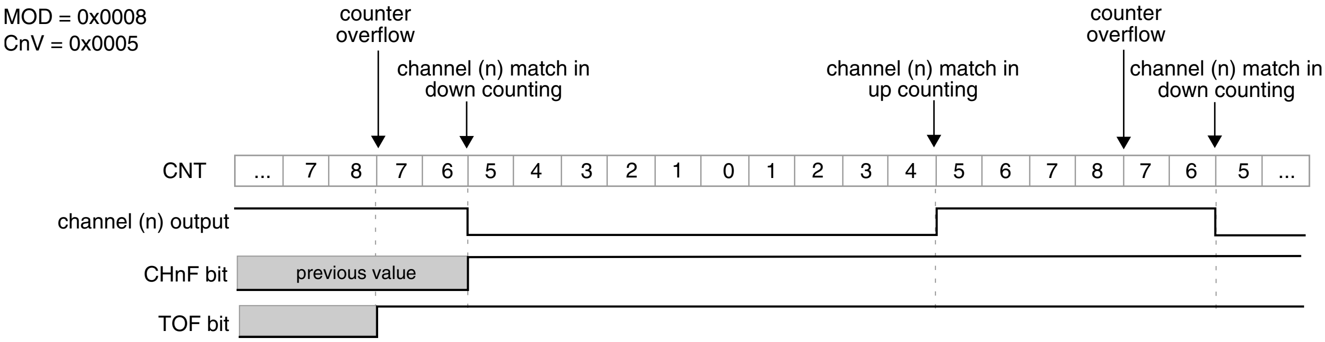

MOD = 0x0008
CnV = 0x0005
counter overflow
channel (n) match in down counting
channel (n) match in up counting
counter overflow
channel (n) match in down counting
CNT
... 7 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 7 6 5 ...
channel (n) output
CHnF bit
previous value
TOF bit

Figure 35-145. CPWM signal with ELSnB:ELSnA = X:1

If (CnV = 0x0000) or CnV is a negative value, that is (CnV[15] = 1), then the channel (n) output is a 0% duty cycle CPWM signal and CHnF bit is not set even when there is the channel (n) match.

If CnV is a positive value, that is (CnV[15] = 0), (CnV ≥ MOD), and (MOD ≠ 0x0000), then the channel (n) output is a 100% duty cycle CPWM signal and CHnF bit is not set even when there is the channel (n) match. This implies that the usable range of periods set by MOD is 0x0001 through 0x7FFE, 0x7FFF if you do not need to generate a 100% duty cycle CPWM signal. This is not a significant limitation because the resulting period is much longer than required for normal applications.

The CPWM mode must not be used when the FTM counter is a free running counter.

Note

The CPWM mode must be used only with CNTIN = 0x0000.

35.4.8 Combine mode

The Combine mode is selected when:

- FTMEN = 1
- QUADEN = 0
- DECAPEN = 0
- COMBINE = 1, and
- CPWMS = 0

In Combine mode, an even channel (n) and adjacent odd channel (n+1) are combined to generate a PWM signal in the channel (n) output.

In the Combine mode, the PWM period is determined by (MOD − CNTIN + 0x0001) and the PWM pulse width (duty cycle) is determined by (|C(n+1)V − C(n)V|).

The CHnF bit is set and the channel (n) interrupt is generated (if CHnIE = 1) at the channel (n) match (FTM counter = C(n)V). The CH(n+1)F bit is set and the channel (n +1) interrupt is generated, if CH(n+1)IE = 1, at the channel (n+1) match (FTM counter = C(n+1)V).

If (ELSnB:ELSnA = 1:0), then the channel (n) output is forced low at the beginning of the period (FTM counter = CNTIN) and at the channel (n+1) match (FTM counter = C(n +1)V). It is forced high at the channel (n) match (FTM counter = C(n)V). See the following figure.

If (ELSnB:ELSnA = X:1), then the channel (n) output is forced high at the beginning of the period (FTM counter = CNTIN) and at the channel (n+1) match (FTM counter = C(n +1)V). It is forced low at the channel (n) match (FTM counter = C(n)V). See the following figure.

In Combine mode, the ELS(n+1)B and ELS(n+1)A bits are not used in the generation of the channels (n) and (n+1) output. However, if (ELSnB:ELSnA = 0:0) then the channel (n) output is not controlled by FTM, and if (ELS(n+1)B:ELS(n+1)A = 0:0) then the channel (n+1) output is not controlled by FTM.

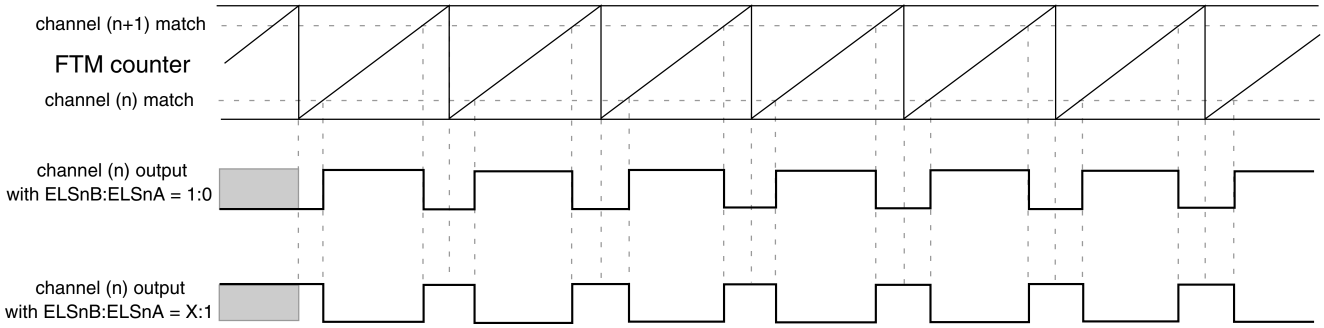


Figure 35-146. Combine mode

The following figures illustrate the PWM signals generation using Combine mode.

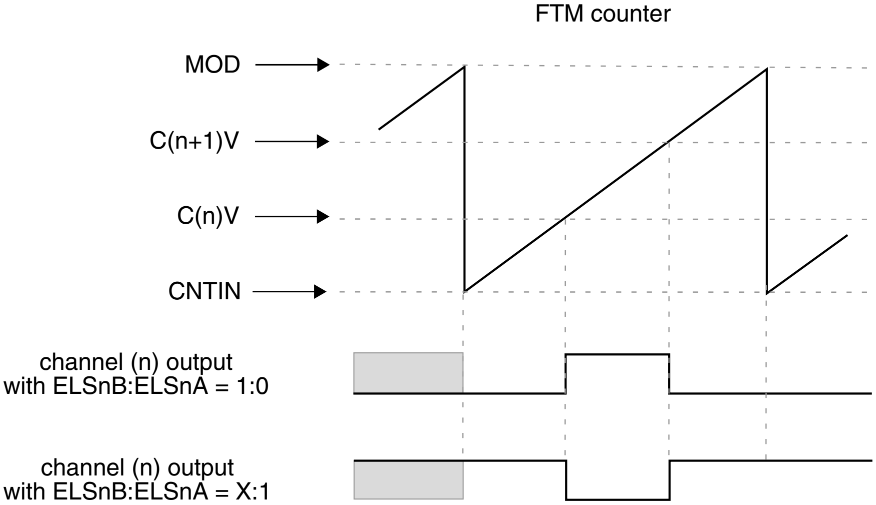

FTM counter
MOD
C(n+1)V
C(n)V
CNTIN
channel (n) output
with ELSnB:ELSnA = 1:0
channel (n) output
with ELSnB:ELSnA = X:1

Figure 35-147. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V < C(n+1)V)

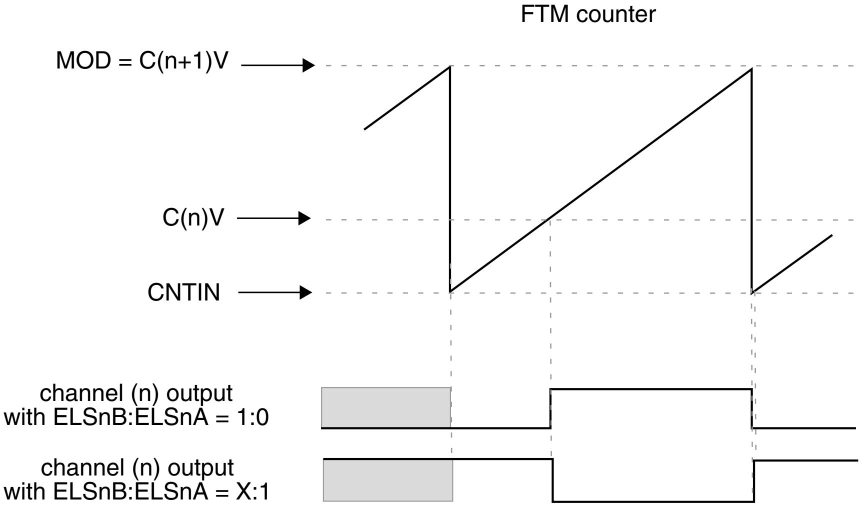

FTM counter
MOD = C(n+1)V
C(n)V
CNTIN
channel (n) output
with ELSnB:ELSnA = 1:0
channel (n) output
with ELSnB:ELSnA = X:1

Figure 35-148. Channel (n) output if (CNTIN < C(n)V < MOD) and (C(n+1)V = MOD)

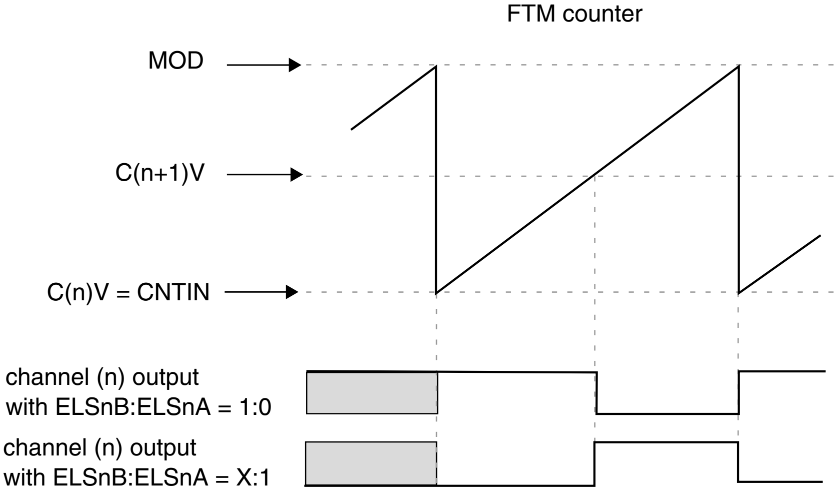

FTM counter
MOD
C(n+1)V
C(n)V = CNTIN
channel (n) output
with ELSnB:ELSnA = 1:0
channel (n) output
with ELSnB:ELSnA = X:1

Figure 35-149. Channel (n) output if (C(n)V = CNTIN) and (CNTIN < C(n+1)V < MOD)

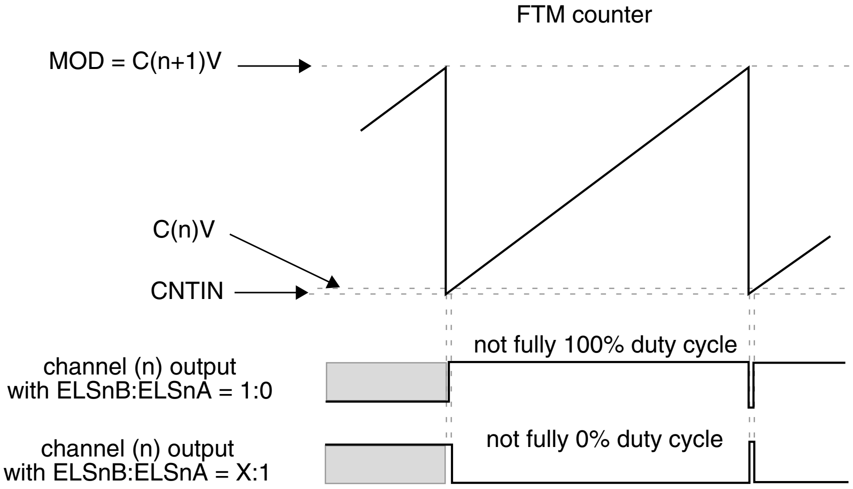

FTM counter
MOD = C(n+1)V
C(n)V
CNTIN
not fully 100% duty cycle
channel (n) output
with ELSnB:ELSnA = 1:0
not fully 0% duty cycle
channel (n) output
with ELSnB:ELSnA = X:1

Figure 35-150. Channel (n) output if (CNTIN < C(n)V < MOD) and (C(n)V is Almost Equal to CNTIN) and (C(n+1)V = MOD)

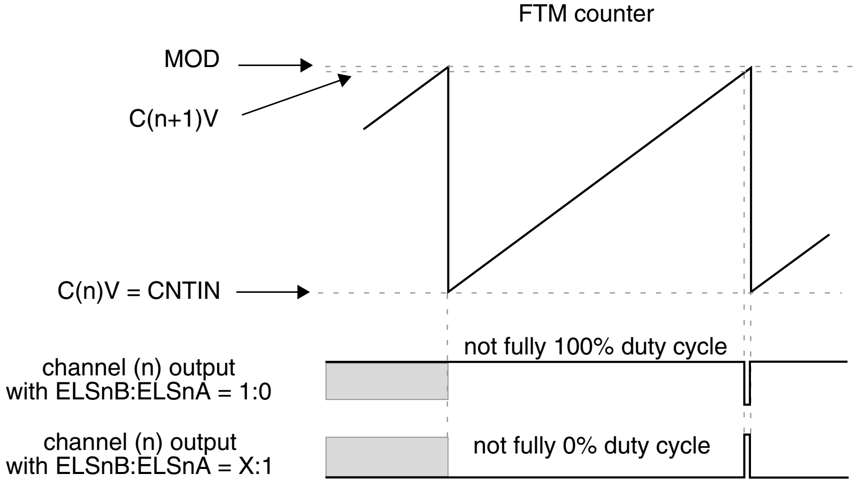

FTM counter
MOD
C(n+1)V
C(n)V = CNTIN
not fully 100% duty cycle
channel (n) output
with ELSnB:ELSnA = 1:0
not fully 0% duty cycle
channel (n) output
with ELSnB:ELSnA = X:1

Figure 35-151. Channel (n) output if (C(n)V = CNTIN) and (CNTIN < C(n+1)V < MOD) and (C(n+1)V is Almost Equal to MOD)

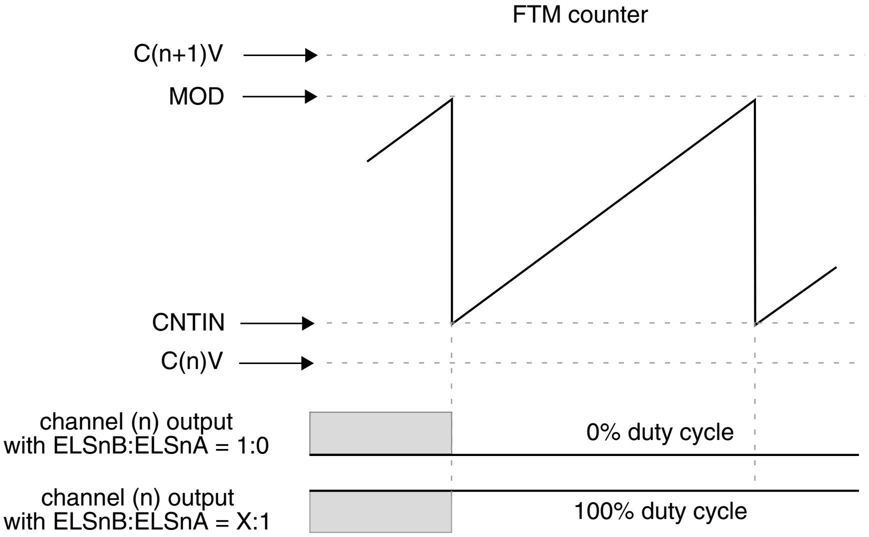

FTM counter
C(n+1)V
MOD
CNTIN
C(n)V
channel (n) output
with ELSnB:ELSnA = 1:0
0% duty cycle
channel (n) output
with ELSnB:ELSnA = X:1
100% duty cycle

Figure 35-152. Channel (n) output if C(n)V and C(n+1)V are not between CNTIN and MOD

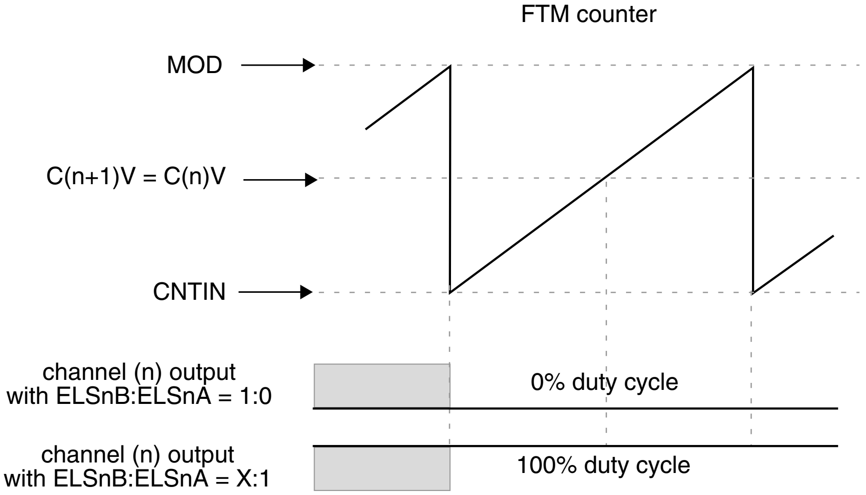

FTM counter
MOD
C(n+1)V = C(n)V
CNTIN
channel (n) output
with ELSnB:ELSnA = 1:0
0% duty cycle
channel (n) output
with ELSnB:ELSnA = X:1
100% duty cycle

Figure 35-153. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V = C(n+1)V)

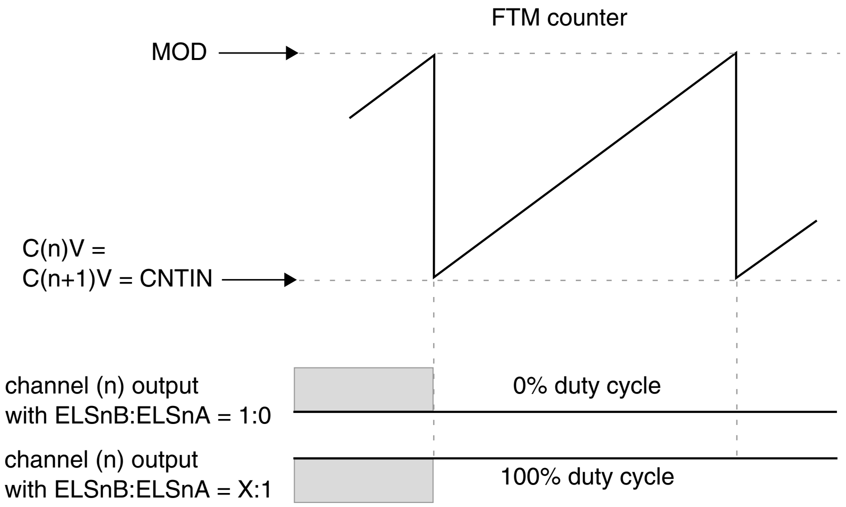

FTM counter
MOD
C(n)V =
C(n+1)V = CNTIN
channel (n) output
with ELSnB:ELSnA = 1:0
0% duty cycle
channel (n) output
with ELSnB:ELSnA = X:1
100% duty cycle

Figure 35-154. Channel (n) output if (C(n)V = C(n+1)V = CNTIN)

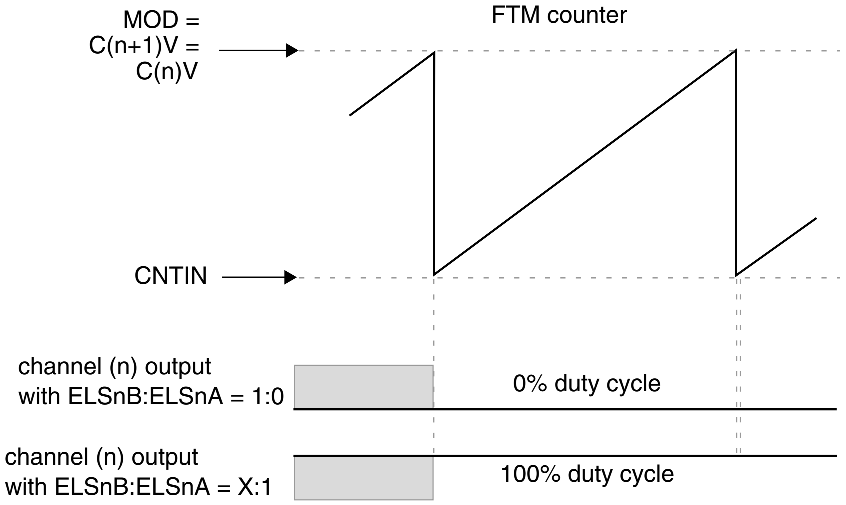

FTM counter
MOD =
C(n+1)V =
C(n)V
CNTIN
channel (n) output
with ELSnB:ELSnA = 1:0
0% duty cycle
channel (n) output
with ELSnB:ELSnA = X:1
100% duty cycle

Figure 35-155. Channel (n) output if (C(n)V = C(n+1)V = MOD)

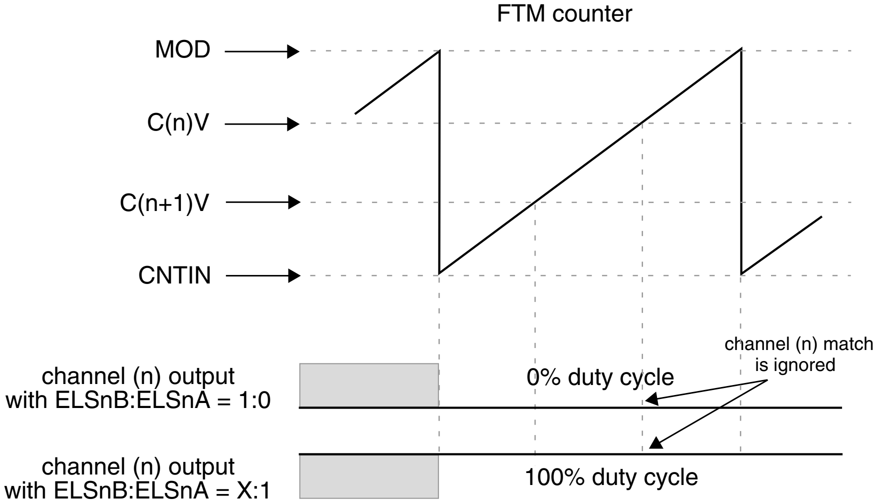

FTM counter
MOD
C(n)V
C(n+1)V
CNTIN
channel (n) match
is ignored
channel (n) output
with ELSnB:ELSnA = 1:0
0% duty cycle
channel (n) output
with ELSnB:ELSnA = X:1
100% duty cycle

Figure 35-156. Channel (n) output if (CNTIN < C(n)V < MOD) and (CNTIN < C(n+1)V < MOD) and (C(n)V > C(n+1)V)

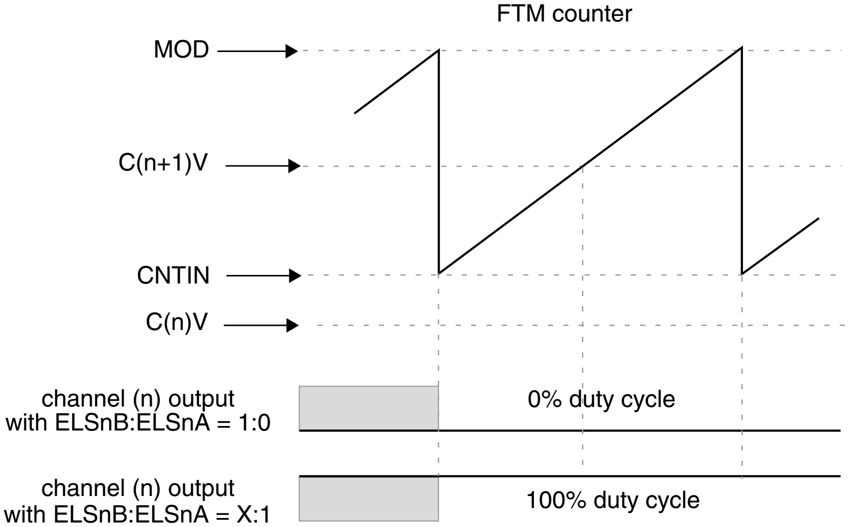

FTM counter
MOD
C(n+1)V
CNTIN
C(n)V
channel (n) output
with ELSnB:ELSnA = 1:0
0% duty cycle
channel (n) output
with ELSnB:ELSnA = X:1
100% duty cycle

Figure 35-157. Channel (n) output if (C(n)V < CNTIN) and (CNTIN < C(n+1)V < MOD)

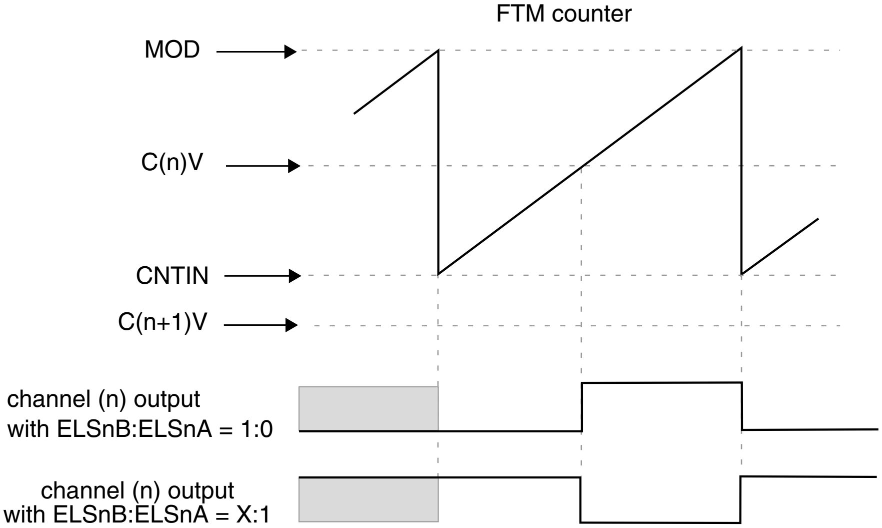

FTM counter
MOD
C(n)V
CNTIN
C(n+1)V
channel (n) output
with ELSnB:ELSnA = 1:0
channel (n) output
with ELSnB:ELSnA = X:1

Figure 35-158. Channel (n) output if (C(n+1)V < CNTIN) and (CNTIN < C(n)V < MOD)

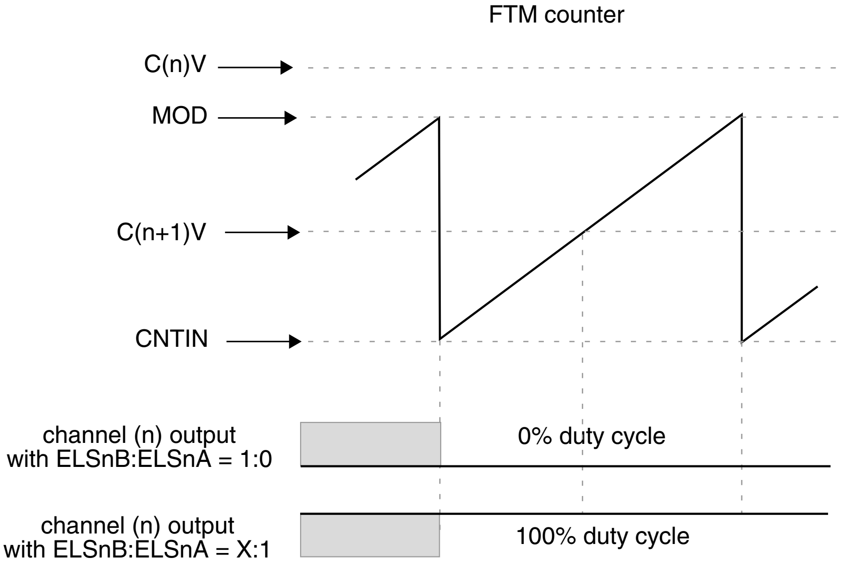

FTM counter
C(n)V
MOD
C(n+1)V
CNTIN
channel (n) output
with ELSnB:ELSnA = 1:0
0% duty cycle
channel (n) output
with ELSnB:ELSnA = X:1
100% duty cycle

Figure 35-159. Channel (n) output if (C(n)V > MOD) and (CNTIN < C(n+1)V < MOD)

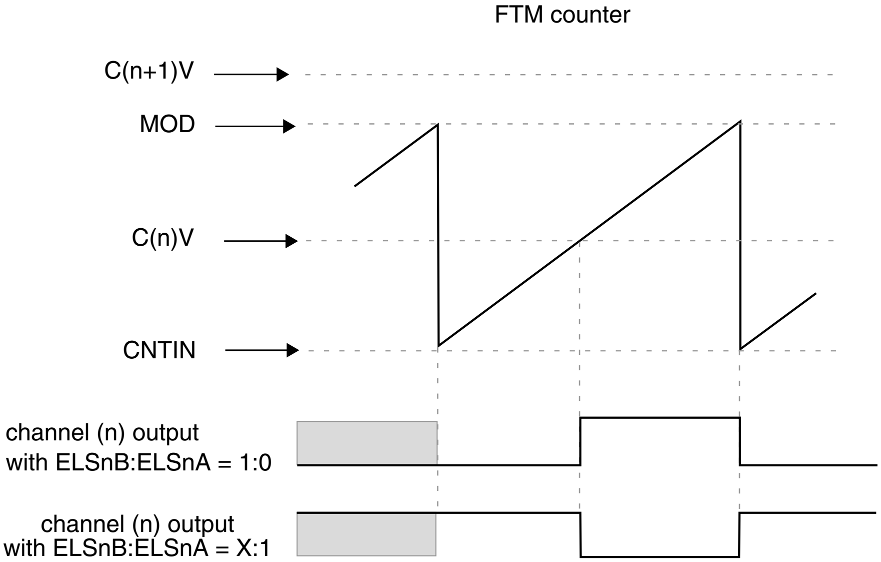

FTM counter
C(n+1)V
MOD
C(n)V
CNTIN
channel (n) output
with ELSnB:ELSnA = 1:0
channel (n) output
with ELSnB:ELSnA = X:1

Figure 35-160. Channel (n) output if (C(n+1)V > MOD) and (CNTIN < C(n)V < MOD)

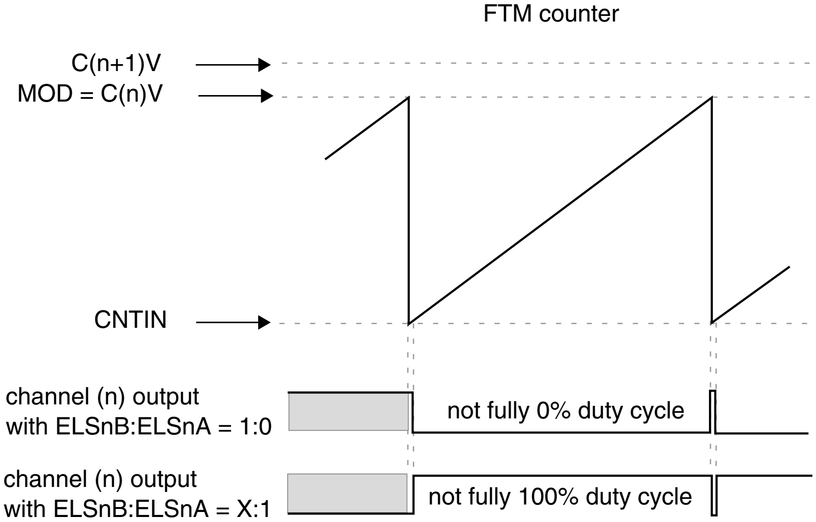

FTM counter
C(n+1)V
MOD = C(n)V
CNTIN
channel (n) output
with ELSnB:ELSnA = 1:0
not fully 0% duty cycle
channel (n) output
with ELSnB:ELSnA = X:1
not fully 100% duty cycle

Figure 35-161. Channel (n) output if (C(n+1)V > MOD) and (CNTIN < C(n)V = MOD)

35.4.8.1 Asymmetrical PWM

In Combine mode, the control of the PWM signal first edge, when the channel (n) match occurs, that is, FTM counter = C(n)V, is independent of the control of the PWM signal second edge, when the channel (n+1) match occurs, that is, FTM counter = C(n+1)V. So, Combine mode allows the generation of asymmetrical PWM signals.

35.4.9 Complementary mode

The Complementary mode is selected when:

- FTMEN = 1
- QUADEN = 0
- DECAPEN = 0
- COMBINE = 1
- CPWMS = 0, and
- COMP = 1

In Complementary mode, the channel (n+1) output is the inverse of the channel (n) output.

If (FTMEN = 1), (QUADEN = 0), (DECAPEN = 0), (COMBINE = 1), (CPWMS = 0), and (COMP = 0), then the channel (n+1) output is the same as the channel (n) output.

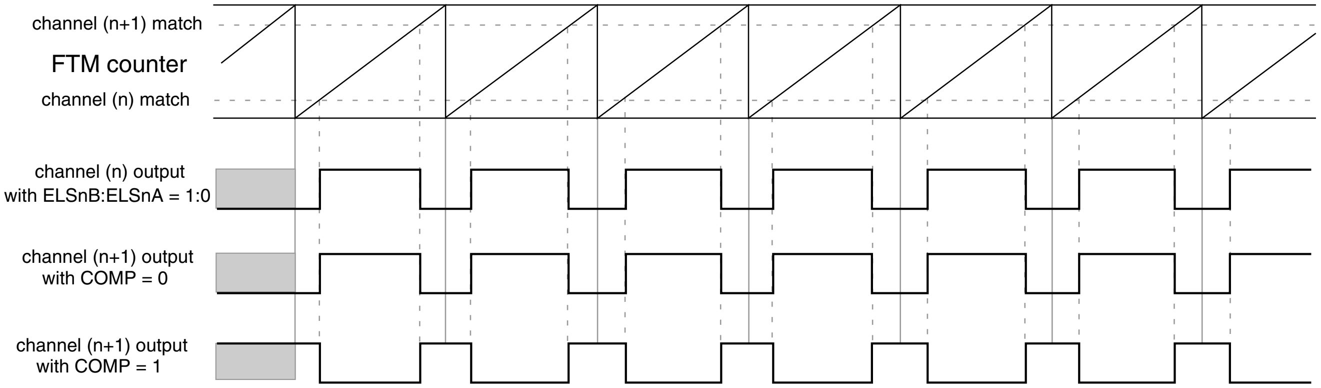

channel (n+1) match
FTM counter
channel (n) match
channel (n) output with ELSnB:ELSnA = 1:0
channel (n+1) output with COMP = 0
channel (n+1) output with COMP = 1

Figure 35-162. Channel (n+1) output in Complementary mode with (ELSnB:ELSnA = 1:0)

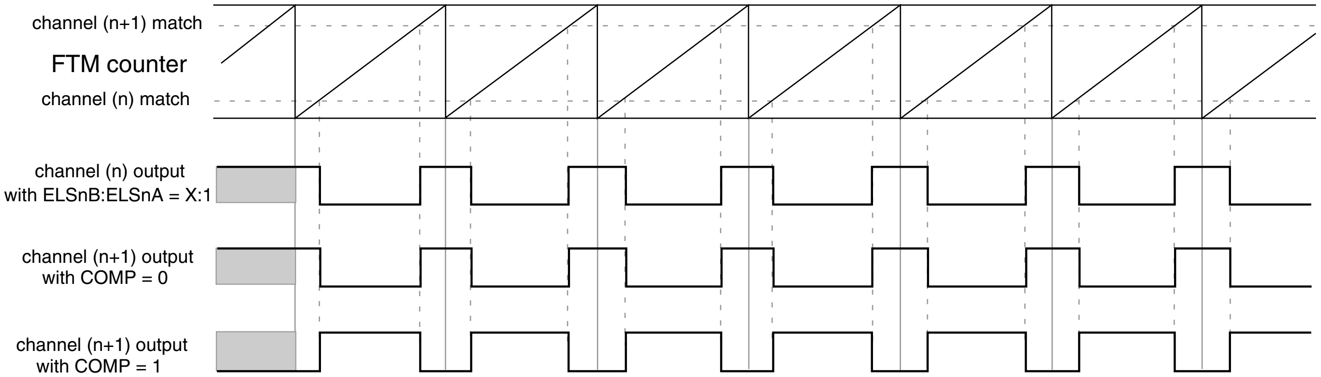

channel (n+1) match
FTM counter
channel (n) match
channel (n) output with ELSnB:ELSnA = X:1
channel (n+1) output with COMP = 0
channel (n+1) output with COMP = 1

Figure 35-163. Channel (n+1) output in Complementary mode with (ELSnB:ELSnA = X:1)

35.4.10 Registers updated from write buffers

35.4.10.1 CNTIN register update

The following table describes when CNTIN register is updated:

Table 35-183. CNTIN register update

When	Then CNTIN register is updated
CLKS[1:0] = 0:0	When CNTIN register is written, independent of FTMEN bit.
• FTMEN = 0, or • CNTINC = 0	At the next system clock after CNTIN was written.
• FTMEN = 1, • SYNCMODE = 1, and • CNTINC = 1	By the CNTIN register synchronization.

35.4.10.2 MOD register update

The following table describes when MOD register is updated:

Table 35-184. MOD register update

When	Then MOD register is updated
CLKS[1:0] = 0:0	When MOD register is written, independent of FTMEN bit.
• CLKS[1:0] ≠ 0:0, and • FTMEN = 0	According to the CPWMS bit, that is: • If the selected mode is not CPWM then MOD register is updated after MOD register was written and the FTM counter changes from MOD to CNTIN. If the FTM counter is at free-running counter mode then this update occurs when the FTM counter changes from 0xFFFF to 0x0000. • If the selected mode is CPWM then MOD register is updated after MOD register was written and the FTM counter changes from MOD to (MOD – 0x0001).
• CLKS[1:0] ≠ 0:0, and • FTMEN = 1	By the MOD register synchronization.

35.4.10.3 CnV register update

The following table describes when CnV register is updated:

Table 35-185. CnV register update

When	Then CnV register is updated
CLKS[1:0] = 0:0	When CnV register is written, independent of FTMEN bit.
• CLKS[1:0] ≠ 0:0, and • FTMEN = 0	According to the selected mode, that is: • If the selected mode is Output Compare, then CnV register is updated on the next FTM counter change, end of the prescaler counting, after CnV register was written. • If the selected mode is EPWM, then CnV register is updated after CnV register was written and the FTM counter changes from MOD to CNTIN. If the FTM counter is at free-running counter mode then this update occurs when the FTM counter changes from 0xFFFF to 0x0000. • If the selected mode is CPWM, then CnV register is updated after CnV register was written and the FTM counter changes from MOD to (MOD – 0x0001).
• CLKS[1:0] ≠ 0:0, and • FTMEN = 1	According to the selected mode, that is: • If the selected mode is output compare then CnV register is updated according to the SYNCEN bit. If (SYNCEN = 0) then CnV register is updated after CnV register was written at the next change of the FTM counter, the end of the prescaler counting. If (SYNCEN = 1) then CnV register is updated by the C(n)V and C(n+1)V register synchronization. • If the selected mode is not output compare and (SYNCEN = 1) then CnV register is updated by the C(n)V and C(n+1)V register synchronization.

35.4.11 PWM synchronization

The PWM synchronization provides an opportunity to update the MOD, CNTIN, CnV, OUTMASK, INVCTRL and SWOCTRL registers with their buffered value and force the FTM counter to the CNTIN register value.

Note

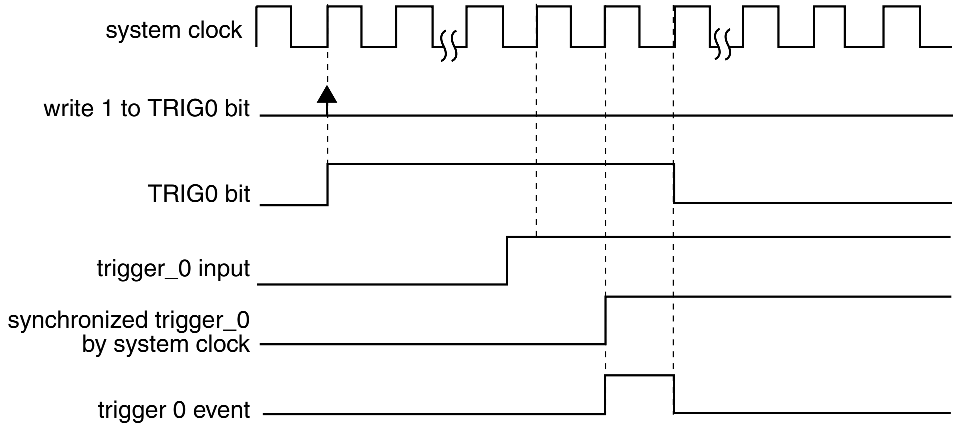
- The PWM synchronization must be used only in Combine mode.
- The legacy PWM synchronization (SYNCMODE = 0) is a subset of the enhanced PWM synchronization (SYNCMODE = 1). Thus, only the enhanced PWM synchronization must be used.

35.4.11.1 Hardware trigger

Three hardware trigger signal inputs of the FTM module are enabled when TRIGn = 1, where n = 0, 1 or 2 corresponding to each one of the input signals, respectively. The hardware trigger input n is synchronized by the system clock. The PWM synchronization with hardware trigger is initiated when a rising edge is detected at the enabled hardware trigger inputs.

If (HWTRIGMODE = 0) then the TRIGn bit is cleared when 0 is written to it or when the trigger n event is detected.

In this case, if two or more hardware triggers are enabled (for example, TRIG0 and TRIG1 = 1) and only trigger 1 event occurs, then only TRIG1 bit is cleared. If a trigger n event occurs together with a write setting TRIGn bit, then the synchronization is initiated, but TRIGn bit remains set due to the write operation.



Note
All hardware trigger inputs have the same behavior.

Figure 35-164. Hardware trigger event with HWTRIGMODE = 0

If HWTRIGMODE = 1, then the TRIGn bit is only cleared when 0 is written to it.

NOTE

The HWTRIGMODE bit must be 1 only with enhanced PWM synchronization (SYNCMODE = 1).

35.4.11.2 Software trigger

A software trigger event occurs when 1 is written to the SYNC[SWSYNC] bit. The SWSYNC bit is cleared when 0 is written to it or when the PWM synchronization, initiated by the software event, is completed.

If another software trigger event occurs (by writing another 1 to the SWSYNC bit) at the same time the PWM synchronization initiated by the previous software trigger event is ending, a new PWM synchronization is started and the SWSYNC bit remains equal to 1.

If SYNCMODE = 0 then the SWSYNC bit is also cleared by FTM according to PWMSYNC and REINIT bits. In this case if (PWMSYNC = 1) or (PWMSYNC = 0 and REINIT = 0) then SWSYNC bit is cleared at the next selected loading point after that the software trigger event occurred; see Boundary cycle and loading points and the following figure. If (PWMSYNC = 0) and (REINIT = 1) then SWSYNC bit is cleared when the software trigger event occurs.

If SYNCMODE = 1 then the SWSYNC bit is also cleared by FTM according to the SWRSTCNT bit. If SWRSTCNT = 0 then SWSYNC bit is cleared at the next selected loading point after that the software trigger event occurred; see the following figure. If SWRSTCNT = 1 then SWSYNC bit is cleared when the software trigger event occurs.

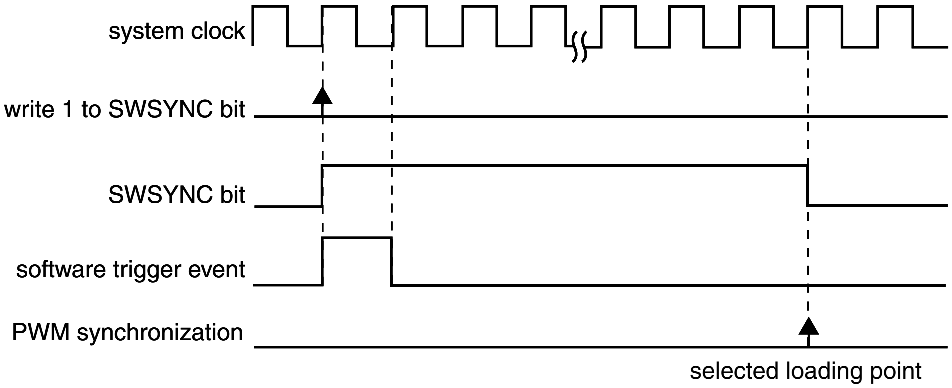


Figure 35-165. Software trigger event

35.4.11.3 Boundary cycle and loading points

The boundary cycle definition is important for the loading points for the registers MOD, CNTIN, and C(n)V.

In Up counting mode, the boundary cycle is defined as when the counter wraps to its initial value (CNTIN). If in Up-down counting mode, then the boundary cycle is defined as when the counter turns from down to up counting and when from up to down counting.

The following figure shows the boundary cycles and the loading points for the registers. In the Up Counting mode, the loading points are enabled if one of CNTMIN or CTMAX bits are 1. In the Up-Down Counting mode, the loading points are selected by CNTMIN and CNTMAX bits, as indicated in the figure. These loading points are safe places for register updates thus allowing a smooth transitions in PWM waveform generation.

For both counting modes, if neither CNTMIN nor CNTMAX are 1, then the boundary cycles are not used as loading points for registers updates. See the register synchronization descriptions in the following sections for details.

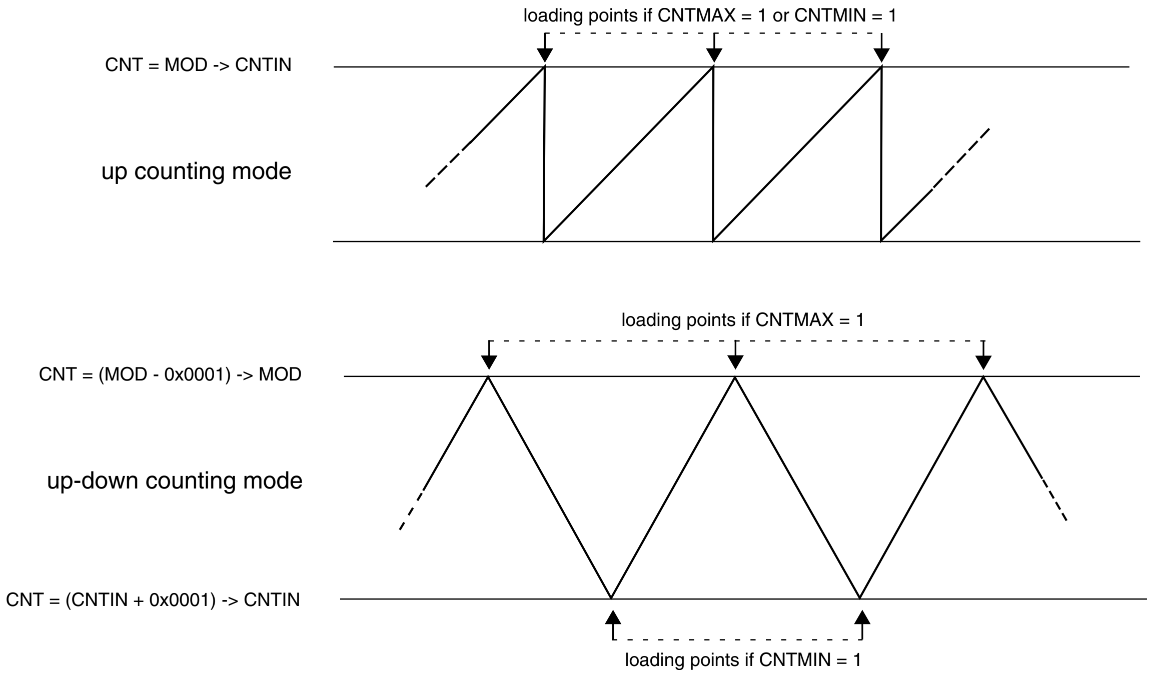

loading points if CNTMAX = 1 or CNTMIN = 1
CNT = MOD -> CNTIN
up counting mode
loading points if CNTMAX = 1
CNT = (MOD - 0x0001) -> MOD
up-down counting mode
CNT = (CNTIN + 0x0001) -> CNTIN
loading points if CNTMIN = 1

Figure 35-166. Boundary cycles and loading points

35.4.11.4 MOD register synchronization

The MOD register synchronization updates the MOD register with its buffer value. This synchronization is enabled if (FTMEN = 1).

The MOD register synchronization can be done by either the enhanced PWM synchronization (SYNCMODE = 1) or the legacy PWM synchronization (SYNCMODE = 0). However, it is expected that the MOD register be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the MOD register synchronization depends on SWWRBUF, SWRSTCNT, HWWRBUF, and HWRSTCNT bits according to this flowchart:

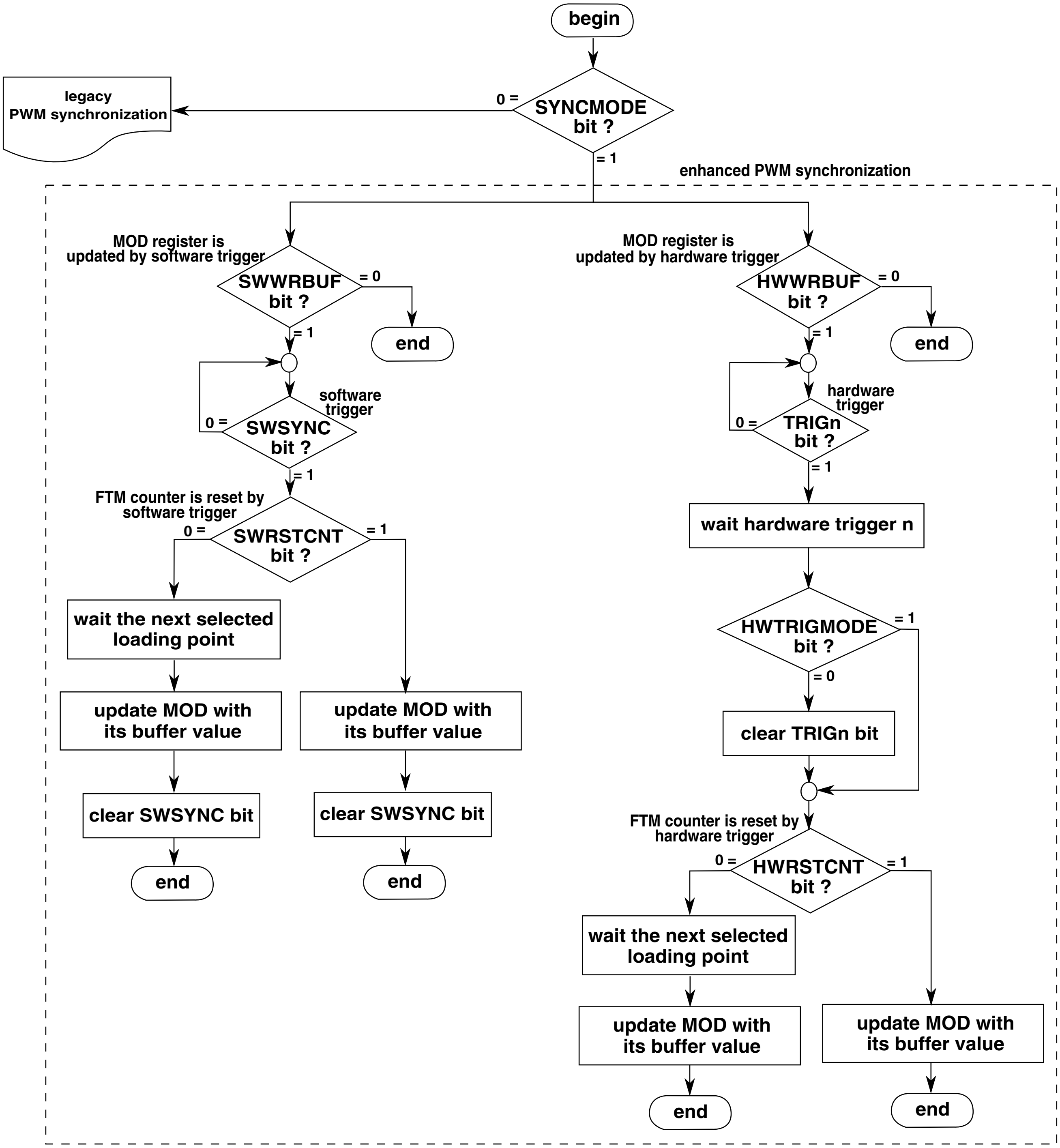

begin
legacy PWM synchronization
0 =
SYNCMODE bit ?
= 1
enhanced PWM synchronization
MOD register is updated by software trigger
SWWRBUF bit ?
= 0
end
= 1
software trigger
0 =
SWSYNC bit ?
= 1
FTM counter is reset by software trigger
0 =
SWRSTCNT bit ?
= 1
wait the next selected loading point
update MOD with its buffer value
clear SWSYNC bit
end
update MOD with its buffer value
clear SWSYNC bit
end
MOD register is updated by hardware trigger
HWWRBUF bit ?
= 0
end
= 1
hardware trigger
0 =
TRIGn bit ?
= 1
wait hardware trigger n
HWTRIGMODE bit ?
= 1
= 0
clear TRIGn bit
FTM counter is reset by hardware trigger
0 =
HWRSTCNT bit ?
= 1
wait the next selected loading point
update MOD with its buffer value
end
update MOD with its buffer value
end

Figure 35-167. MOD register synchronization flowchart

In the case of legacy PWM synchronization, the MOD register synchronization depends on PWMSYNC and REINIT bits according to the following description.

If (SYNCMODE = 0), (PWMSYNC = 0), and (REINIT = 0), then this synchronization is made on the next selected loading point after an enabled trigger event takes place. If the trigger event was a software trigger, then the SWSYNC bit is cleared on the next selected loading point. If the trigger event was a hardware trigger, then the trigger enable bit (TRIGn) is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

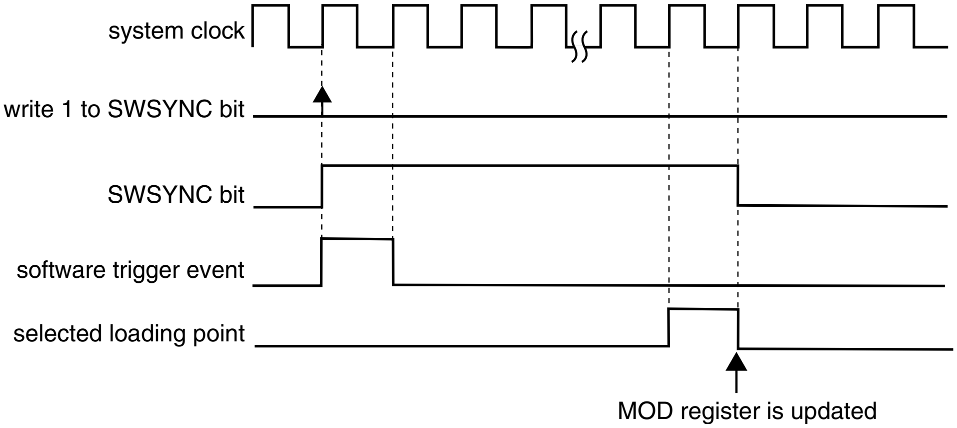

system clock
write 1 to SWSYNC bit
SWSYNC bit
software trigger event
selected loading point
MOD register is updated

Figure 35-168. MOD synchronization with (SYNCMODE = 0), (PWMSYNC = 0), (REINIT = 0), and software trigger was used

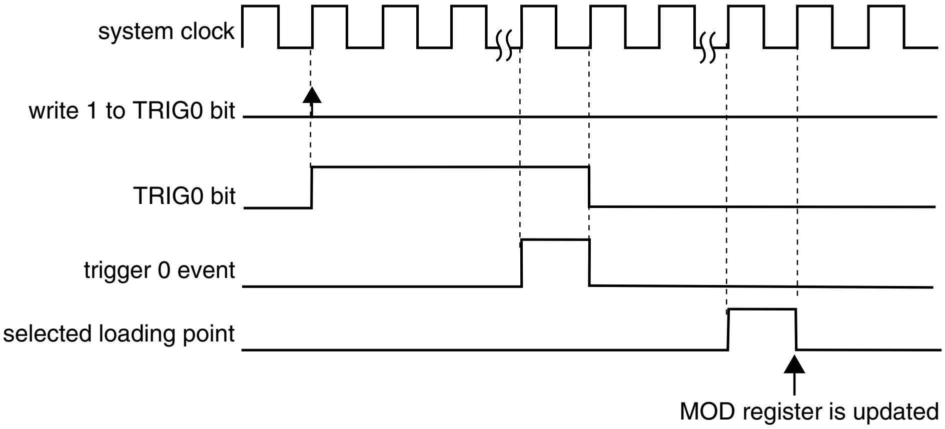

system clock
write 1 to TRIG0 bit
TRIG0 bit
trigger 0 event
selected loading point
MOD register is updated

Figure 35-169. MOD synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (PWMSYNC = 0), (REINIT = 0), and a hardware trigger was used

If (SYNCMODE = 0), (PWMSYNC = 0), and (REINIT = 1), then this synchronization is made on the next enabled trigger event. If the trigger event was a software trigger, then the SWSYNC bit is cleared according to the following example. If the trigger event was a hardware trigger, then the TRIGn bit is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

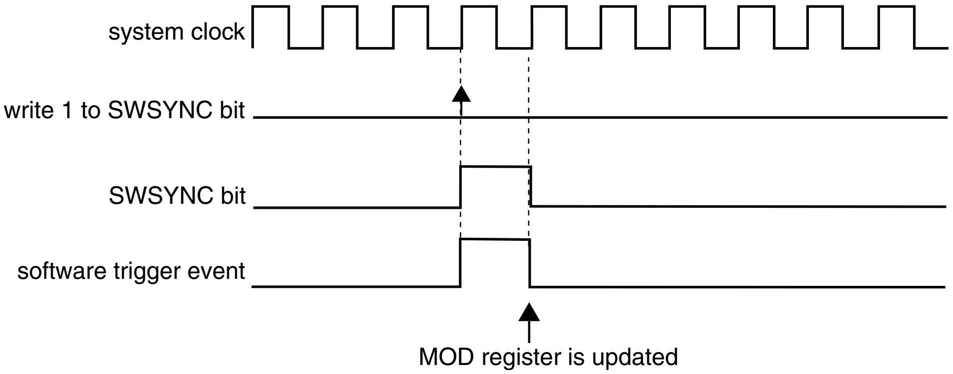

system clock
write 1 to SWSYNC bit
SWSYNC bit
software trigger event
MOD register is updated

Figure 35-170. MOD synchronization with (SYNCMODE = 0), (PWMSYNC = 0), (REINIT = 1), and software trigger was used

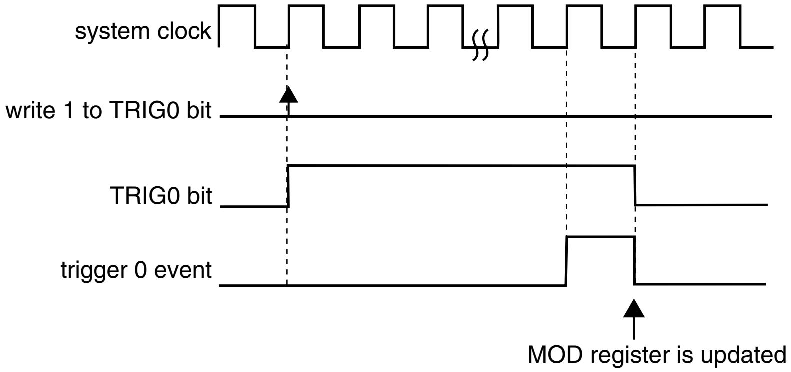

system clock
write 1 to TRIG0 bit
TRIG0 bit
trigger 0 event
MOD register is updated

Figure 35-171. MOD synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (PWMSYNC = 0), (REINIT = 1), and a hardware trigger was used

If (SYNCMODE = 0) and (PWMSYNC = 1), then this synchronization is made on the next selected loading point after the software trigger event takes place. The SWSYNC bit is cleared on the next selected loading point:

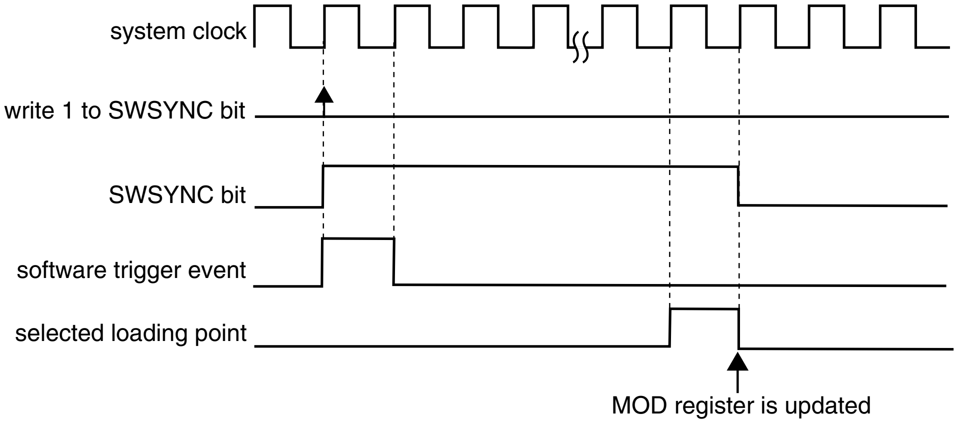

system clock
write 1 to SWSYNC bit
SWSYNC bit
software trigger event
selected loading point
MOD register is updated

Figure 35-172. MOD synchronization with (SYNCMODE = 0) and (PWMSYNC = 1)

35.4.11.5 CNTIN register synchronization

The CNTIN register synchronization updates the CNTIN register with its buffer value.

This synchronization is enabled if (FTMEN = 1), (SYNCMODE = 1), and (CNTINC = 1). The CNTIN register synchronization can be done only by the enhanced PWM synchronization (SYNCMODE = 1). The synchronization mechanism is the same as the MOD register synchronization done by the enhanced PWM synchronization; see MOD register synchronization.

35.4.11.6 C(n)V and C(n+1)V register synchronization

The C(n)V and C(n+1)V registers synchronization updates the C(n)V and C(n+1)V registers with their buffer values.

This synchronization is enabled if (FTMEN = 1) and (SYNCEN = 1). The synchronization mechanism is the same as the MOD register synchronization. However, it is expected that the C(n)V and C(n+1)V registers be synchronized only by the enhanced PWM synchronization (SYNCMODE = 1).

35.4.11.7 OUTMASK register synchronization

The OUTMASK register synchronization updates the OUTMASK register with its buffer value.

The OUTMASK register can be updated at each rising edge of system clock (SYNCHOM = 0), by the enhanced PWM synchronization (SYNCHOM = 1 and SYNCMODE = 1) or by the legacy PWM synchronization (SYNCHOM = 1 and SYNCMODE = 0). However, it is expected that the OUTMASK register be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the OUTMASK register synchronization depends on SWOM and HWOM bits. See the following flowchart:

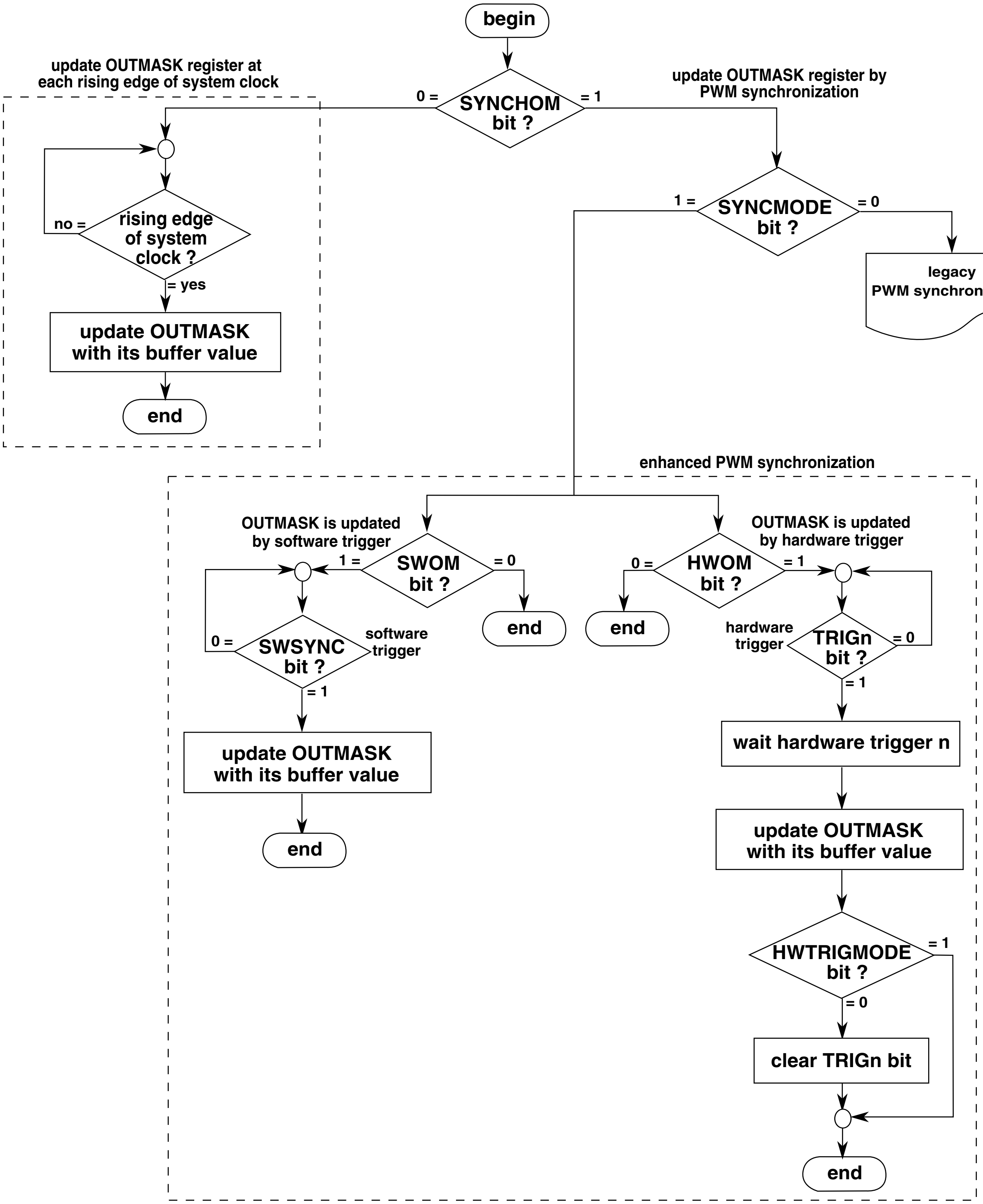

begin
SYNCHOM bit ?
0 =
= 1
update OUTMASK register at each rising edge of system clock
update OUTMASK register by PWM synchronization
no =
rising edge of system clock ?
= yes
update OUTMASK with its buffer value
end
SYNCMODE bit ?
1 =
= 0
legacy PWM synchronization
enhanced PWM synchronization
OUTMASK is updated by software trigger
SWOM bit ?
1 =
= 0
end
0 =
SWSYNC bit ?
software trigger
= 1
update OUTMASK with its buffer value
end
OUTMASK is updated by hardware trigger
HWOM bit ?
0 =
= 1
end
hardware trigger
TRIGn bit ?
= 0
= 1
wait hardware trigger n
update OUTMASK with its buffer value
HWTRIGMODE bit ?
= 1
= 0
clear TRIGn bit
end

Figure 35-173. OUTMASK register synchronization flowchart

In the case of legacy PWM synchronization, the OUTMASK register synchronization depends on PWMSYNC bit according to the following description.

If (SYNCMODE = 0), (SYNCHOM = 1), and (PWMSYNC = 0), then this synchronization is done on the next enabled trigger event. If the trigger event was a software trigger, then the SWSYNC bit is cleared on the next selected loading point. If the trigger event was a hardware trigger, then the TRIGn bit is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

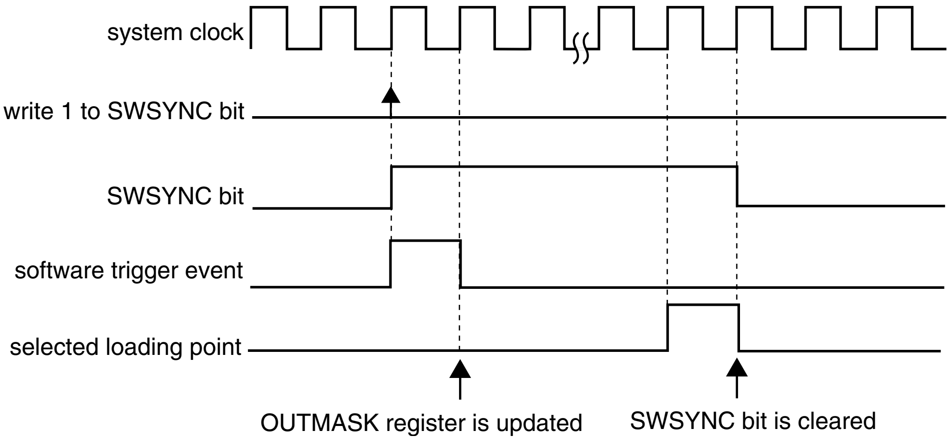

system clock
write 1 to SWSYNC bit
SWSYNC bit
software trigger event
selected loading point
OUTMASK register is updated
SWSYNC bit is cleared

Figure 35-174. OUTMASK synchronization with (SYNCMODE = 0), (SYNCHOM = 1), (PWMSYNC = 0) and software trigger was used

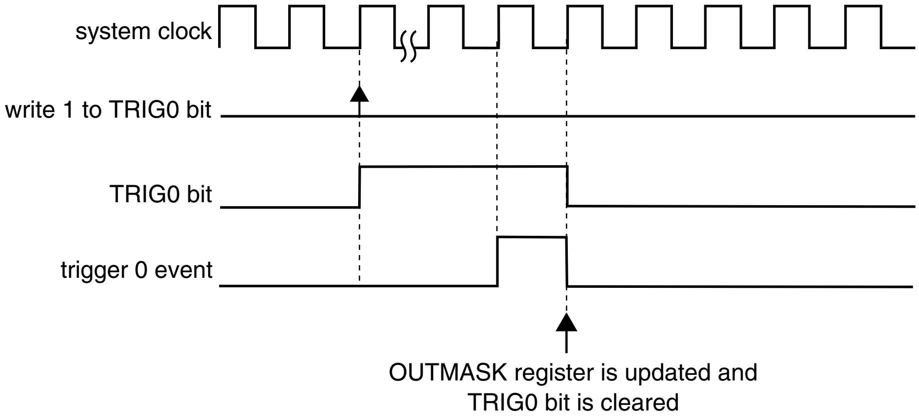

system clock
write 1 to TRIG0 bit
TRIG0 bit
trigger 0 event
OUTMASK register is updated and
TRIG0 bit is cleared

Figure 35-175. OUTMASK synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (SYNCHOM = 1), (PWMSYNC = 0), and a hardware trigger was used

If (SYNCMODE = 0), (SYNCHOM = 1), and (PWMSYNC = 1), then this synchronization is made on the next enabled hardware trigger. The TRIGn bit is cleared according to Hardware trigger. An example with a hardware trigger follows.

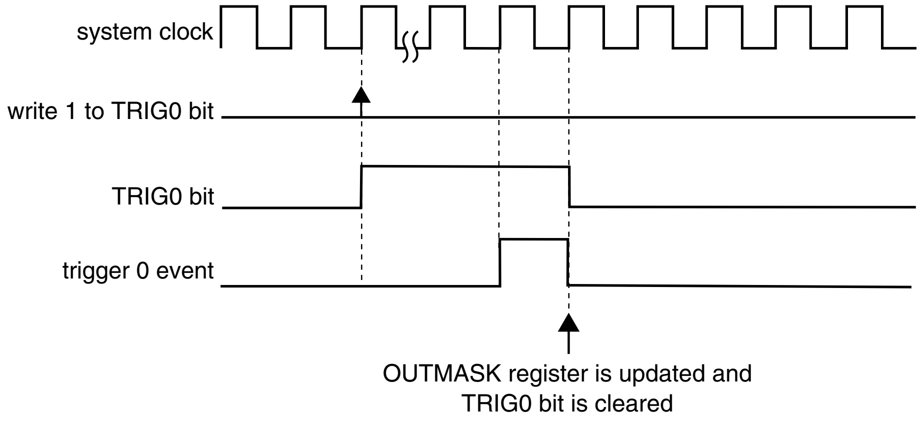

system clock
write 1 to TRIG0 bit
TRIG0 bit
trigger 0 event
OUTMASK register is updated and
TRIG0 bit is cleared

Figure 35-176. OUTMASK synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (SYNCHOM = 1), (PWMSYNC = 1), and a hardware trigger was used

35.4.11.8 INVCTRL register synchronization

The INVCTRL register synchronization updates the INVCTRL register with its buffer value.

The INVCTRL register can be updated at each rising edge of system clock (INVC = 0) or by the enhanced PWM synchronization (INVC = 1 and SYNCMODE = 1) according to the following flowchart.

In the case of enhanced PWM synchronization, the INVCTRL register synchronization depends on SWINVC and HWINVC bits.

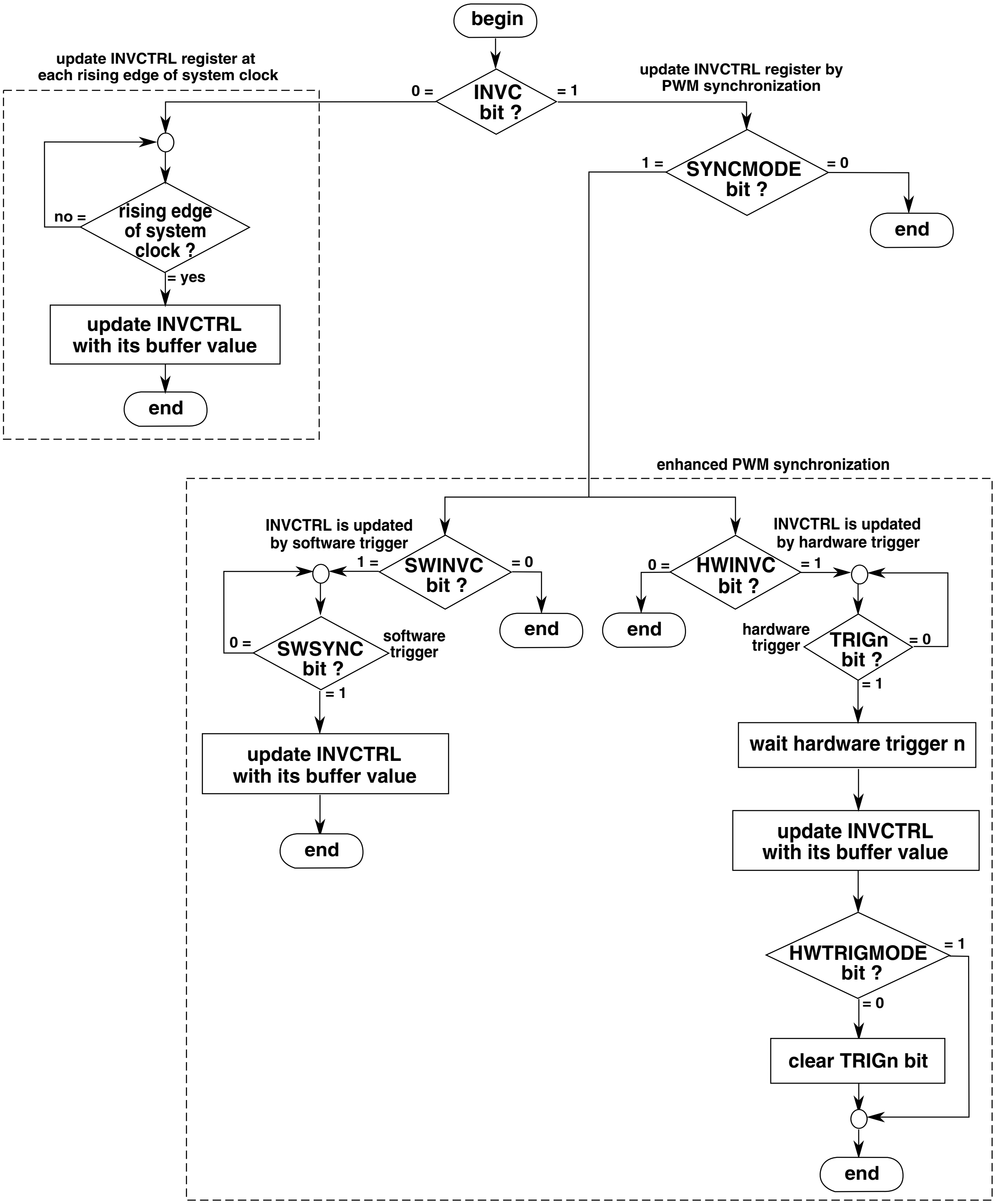

begin
INVC bit ?
0 =
= 1
update INVCTRL register at each rising edge of system clock
update INVCTRL register by PWM synchronization
no =
rising edge of system clock ?
= yes
update INVCTRL with its buffer value
end
SYNCMODE bit ?
1 =
= 0
end
enhanced PWM synchronization
INVCTRL is updated by software trigger
SWINVC bit ?
1 =
= 0
end
0 =
SWSYNC bit ?
software trigger
= 1
update INVCTRL with its buffer value
end
INVCTRL is updated by hardware trigger
HWINVC bit ?
0 =
= 1
end
hardware trigger
TRIGn bit ?
= 0
= 1
wait hardware trigger n
update INVCTRL with its buffer value
HWTRIGMODE bit ?
= 1
= 0
clear TRIGn bit
end

Figure 35-177. INVCTRL register synchronization flowchart

35.4.11.9 SWOCTRL register synchronization

The SWOCTRL register synchronization updates the SWOCTRL register with its buffer value.

The SWOCTRL register can be updated at each rising edge of system clock (SWOC = 0) or by the enhanced PWM synchronization (SWOC = 1 and SYNCMODE = 1) according to the following flowchart.

In the case of enhanced PWM synchronization, the SWOCTRL register synchronization depends on SWSOC and HWSOC bits.

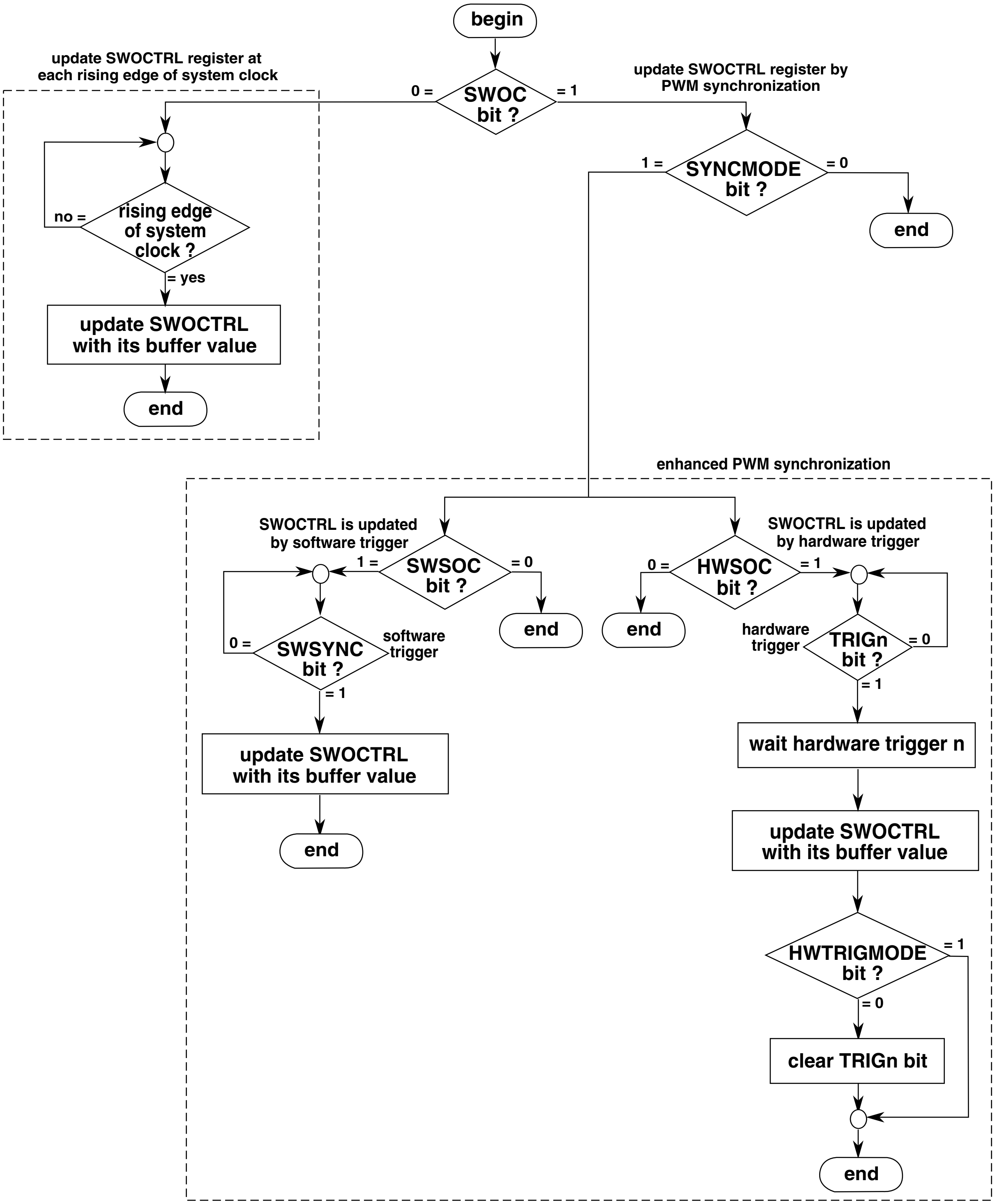

begin
update SWOCTRL register at each rising edge of system clock
0 =
SWOC bit ?
= 1
update SWOCTRL register by PWM synchronization
1 =
SYNCMODE bit ?
= 0
end
no =
rising edge of system clock ?
= yes
update SWOCTRL with its buffer value
end
enhanced PWM synchronization
SWOCTRL is updated by software trigger
1 =
SWSOC bit ?
= 0
end
0 =
SWSYNC bit ?
software trigger
= 1
update SWOCTRL with its buffer value
end
SWOCTRL is updated by hardware trigger
0 =
HWSOC bit ?
= 1
end
hardware trigger
TRIGn bit ?
= 0
= 1
wait hardware trigger n
update SWOCTRL with its buffer value
HWTRIGMODE bit ?
= 1
= 0
clear TRIGn bit
end

Figure 35-178. SWOCTRL register synchronization flowchart

35.4.11.10 FTM counter synchronization

The FTM counter synchronization is a mechanism that allows the FTM to restart the PWM generation at a certain point in the PWM period. The channels outputs are forced to their initial value, except for channels in Output Compare mode, and the FTM counter is forced to its initial counting value defined by CNTIN register.

The following figure shows the FTM counter synchronization. Note that after the synchronization event occurs, the channel (n) is set to its initial value and the channel (n +1) is not set to its initial value due to a specific timing of this figure in which the deadtime insertion prevents this channel output from transitioning to 1. If no deadtime insertion is selected, then the channel (n+1) transitions to logical value 1 immediately after the synchronization event occurs.

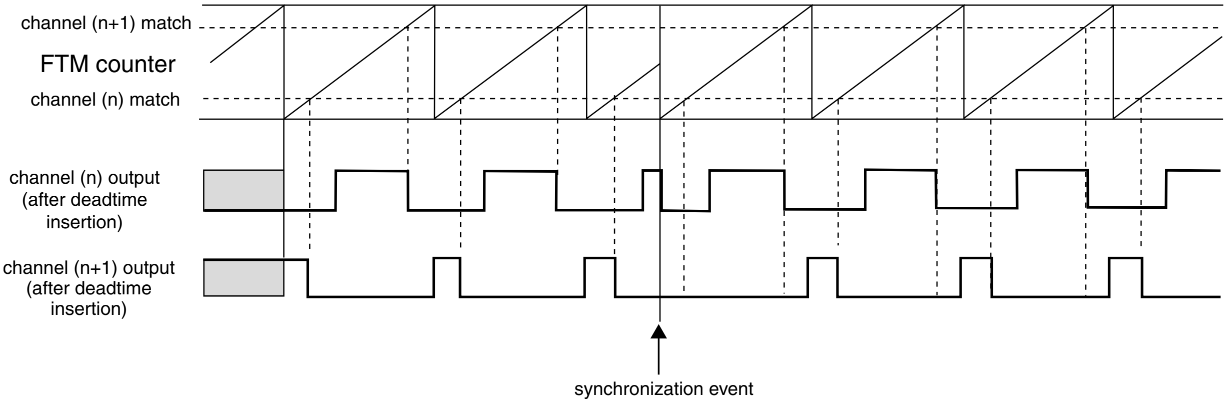

channel (n+1) match
FTM counter
channel (n) match
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)
synchronization event

Figure 35-179. FTM counter synchronization

The FTM counter synchronization can be done by either the enhanced PWM synchronization (SYNCMODE = 1) or the legacy PWM synchronization (SYNCMODE = 0). However, the FTM counter must be synchronized only by the enhanced PWM synchronization.

In the case of enhanced PWM synchronization, the FTM counter synchronization depends on SWRSTCNT and HWRSTCNT bits according to the following flowchart.

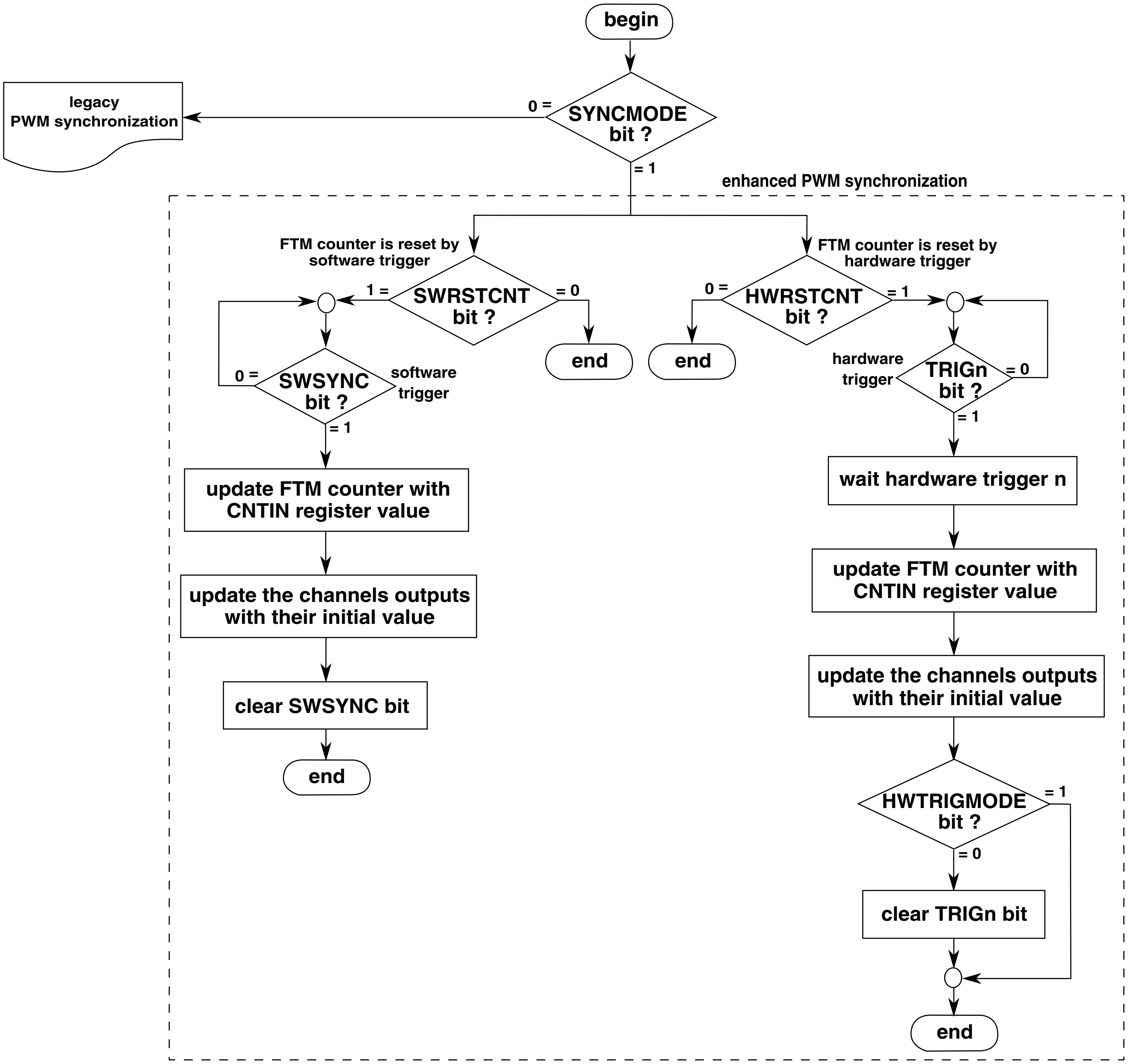

begin
SYNCMODE bit ?
0 =
legacy PWM synchronization
= 1
enhanced PWM synchronization
FTM counter is reset by software trigger
SWRSTCNT bit ?
1 =
= 0
end
SWSYNC bit ?
0 =
software trigger
= 1
update FTM counter with CNTIN register value
update the channels outputs with their initial value
clear SWSYNC bit
end
FTM counter is reset by hardware trigger
HWRSTCNT bit ?
0 =
= 1
end
hardware trigger
TRIGn bit ?
= 0
= 1
wait hardware trigger n
update FTM counter with CNTIN register value
update the channels outputs with their initial value
HWTRIGMODE bit ?
= 1
= 0
clear TRIGn bit
end

Figure 35-180. FTM counter synchronization flowchart

In the case of legacy PWM synchronization, the FTM counter synchronization depends on REINIT and PWMSYNC bits according to the following description.

If (SYNCMODE = 0), (REINIT = 1), and (PWMSYNC = 0) then this synchronization is made on the next enabled trigger event. If the trigger event was a software trigger then the SWSYNC bit is cleared according to the following example. If the trigger event was a hardware trigger then the TRIGn bit is cleared according to Hardware trigger. Examples with software and hardware triggers follow.

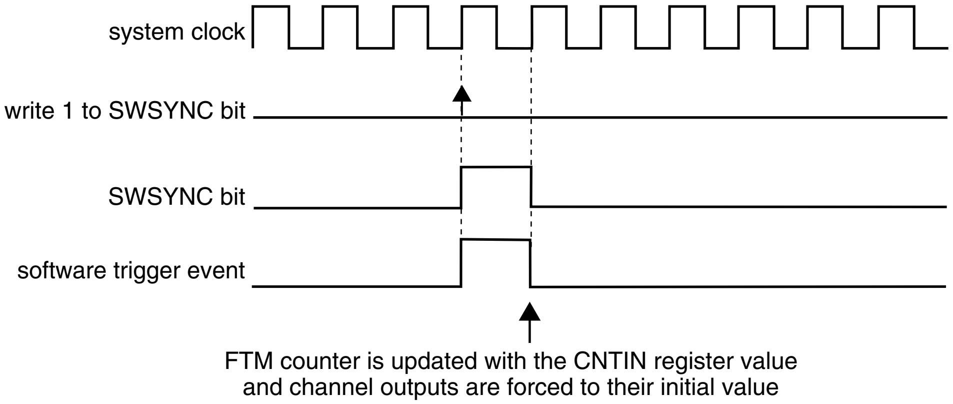

system clock
write 1 to SWSYNC bit
SWSYNC bit
software trigger event
FTM counter is updated with the CNTIN register value
and channel outputs are forced to their initial value

Figure 35-181. FTM counter synchronization with (SYNCMODE = 0), (REINIT = 1), (PWMSYNC = 0), and software trigger was used

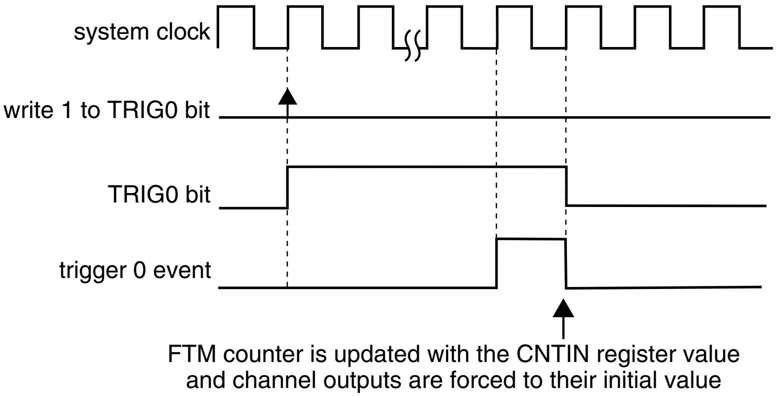

system clock
write 1 to TRIG0 bit
TRIG0 bit
trigger 0 event
FTM counter is updated with the CNTIN register value
and channel outputs are forced to their initial value

Figure 35-182. FTM counter synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (REINIT = 1), (PWMSYNC = 0), and a hardware trigger was used

If (SYNCMODE = 0), (REINIT = 1), and (PWMSYNC = 1) then this synchronization is made on the next enabled hardware trigger. The TRIGn bit is cleared according to Hardware trigger.

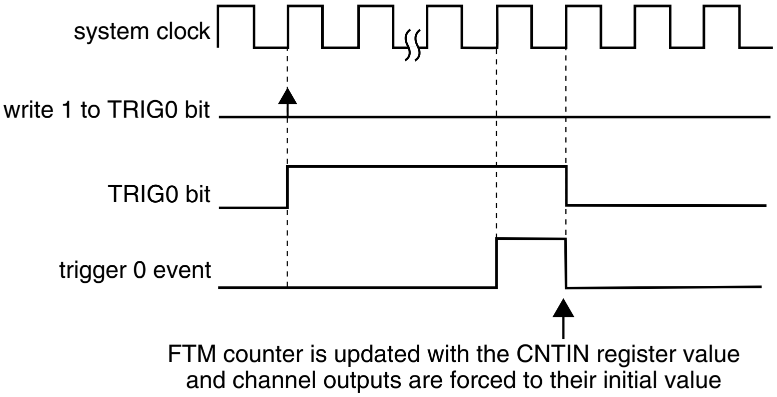
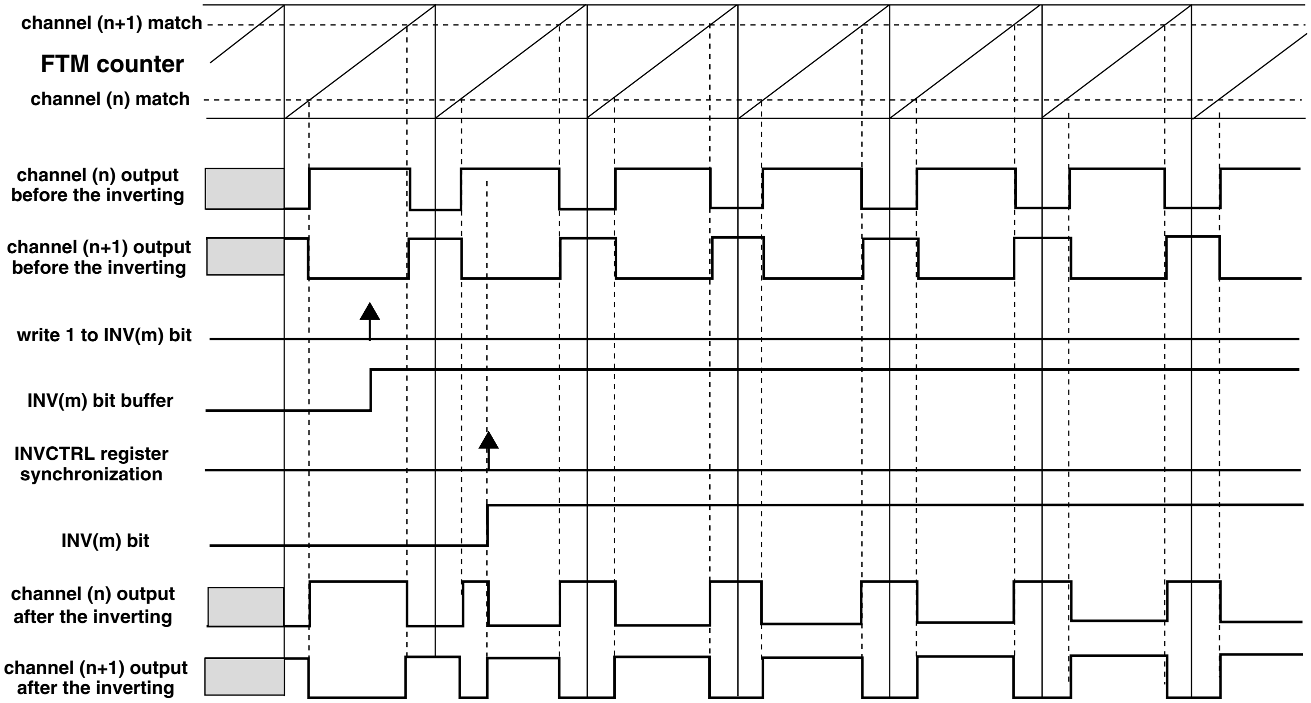

system clock
write 1 to TRIG0 bit
TRIG0 bit
trigger 0 event
FTM counter is updated with the CNTIN register value
and channel outputs are forced to their initial value

Figure 35-183. FTM counter synchronization with (SYNCMODE = 0), (HWTRIGMODE = 0), (REINIT = 1), (PWMSYNC = 1), and a hardware trigger was used

35.4.12 Inverting

The invert functionality swaps the signals between channel (n) and channel (n+1) outputs. The inverting operation is selected when (FTMEN = 1), (QUADEN = 0), (DECAPEN = 0), (COMBINE = 1), (COMP = 1), (CPWMS = 0), and (INVm = 1), where m represents a channel pair. The INVm bit in INVCTRL register is updated with its buffer value according to INVCTRL register synchronization

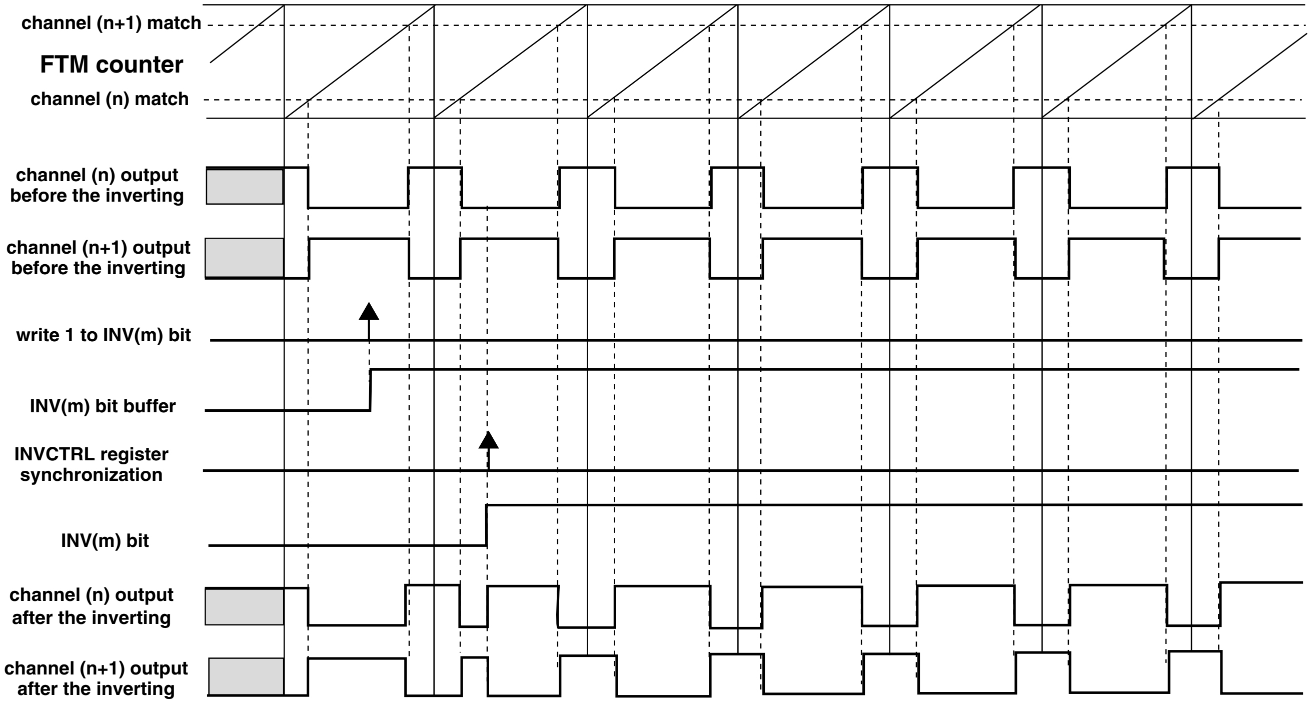
In High-True (ELSnB:ELSnA = 1:0) Combine mode, the channel (n) output is forced low at the beginning of the period (FTM counter = CNTIN), forced high at the channel (n) match and forced low at the channel (n+1) match. If the inverting is selected, the channel (n) output behavior is changed to force high at the beginning of the PWM period, force low at the channel (n) match and force high at the channel (n+1) match. See the following figure.


channel (n+1) match
FTM counter
channel (n) match
channel (n) output before the inverting
channel (n+1) output before the inverting
write 1 to INV(m) bit
INV(m) bit buffer
INVCTRL register synchronization
INV(m) bit
channel (n) output after the inverting
channel (n+1) output after the inverting

NOTE
INV(m) bit selects the inverting to the pair channels (n) and (n+1).

Figure 35-184. Channels (n) and (n+1) outputs after the inverting in High-True (ELSnB:ELSnA = 1:0) Combine mode

Note that the ELSnB:ELSnA bits value should be considered because they define the active state of the channels outputs. In Low-True (ELSnB:ELSnA = X:1) Combine mode, the channel (n) output is forced high at the beginning of the period, forced low at the channel (n) match and forced high at the channel (n+1) match. When inverting is selected, the channels (n) and (n+1) present waveforms as shown in the following figure.


channel (n+1) match
FTM counter
channel (n) match
channel (n) output before the inverting
channel (n+1) output before the inverting
write 1 to INV(m) bit
INV(m) bit buffer
INVCTRL register synchronization
INV(m) bit
channel (n) output after the inverting
channel (n+1) output after the inverting

NOTE
INV(m) bit selects the inverting to the pair channels (n) and (n+1).

Figure 35-185. Channels (n) and (n+1) outputs after the inverting in Low-True (ELSnB:ELSnA = X:1) Combine mode

Note

The inverting feature must be used only in Combine mode.

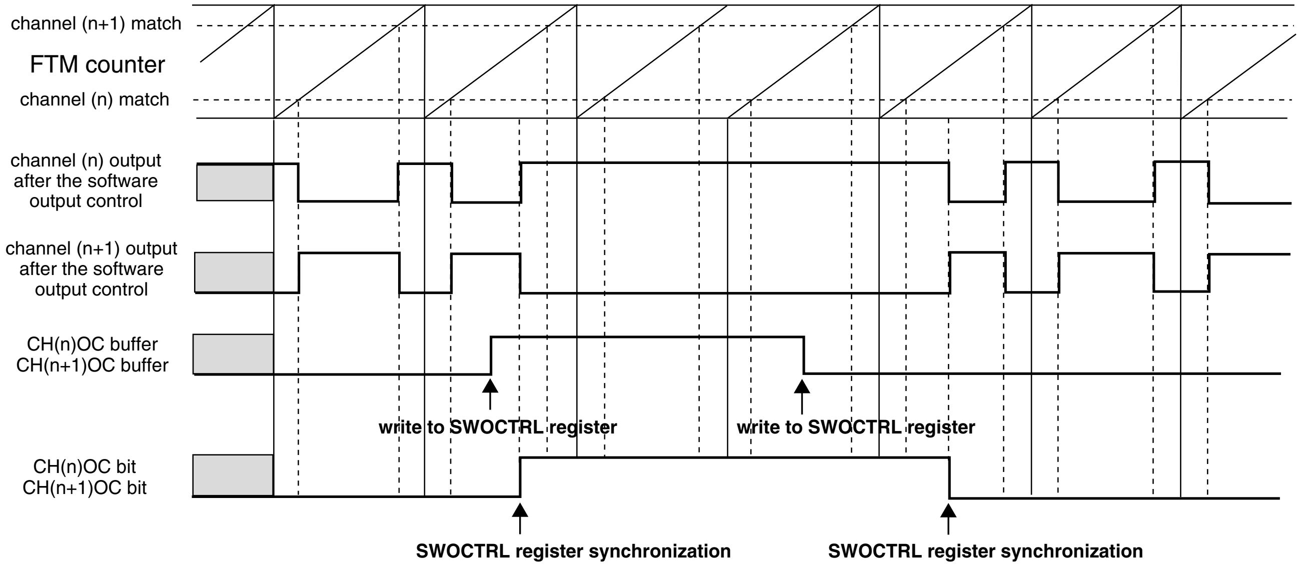
35.4.13 Software output control

The software output control forces the channel output according to software defined values at a specific time in the PWM generation.

The software output control is selected when (FTMEN = 1), (QUADEN = 0), (DECAPEN = 0), (COMBINE = 1), (CPWMS = 0), and (CHnOC = 1). The CHnOC bit enables the software output control for a specific channel output and the CHnOCV selects the value that is forced to this channel output.

Both CHnOC and CHnOCV bits in SWOCTRL register are buffered and updated with their buffer value according to SWOCTRL register synchronization.

The following figure shows the channels (n) and (n+1) outputs signals when the software output control is used. In this case the channels (n) and (n+1) are set to Combine and Complementary mode.


channel (n+1) match
FTM counter
channel (n) match
channel (n) output after the software output control
channel (n+1) output after the software output control
CH(n)OC buffer
CH(n+1)OC buffer
write to SWOCTRL register
write to SWOCTRL register
CH(n)OC bit
CH(n+1)OC bit
SWOCTRL register synchronization
SWOCTRL register synchronization

NOTE
CH(n)OCV = 1 and CH(n+1)OCV = 0.

Figure 35-186. Example of software output control in Combine and Complementary mode

Software output control forces the following values on channels (n) and (n+1) when the COMP bit is zero.

Table 35-186. Software ouput control behavior when (COMP = 0)

CH(n)OC	CH(n+1)OC	CH(n)OCV	CH(n+1)OCV	Channel (n) Output	Channel (n+1) Output
0	0	X	X	is not modified by SWOC	is not modified by SWOC
1	1	0	0	is forced to zero	is forced to zero
1	1	0	1	is forced to zero	is forced to one
1	1	1	0	is forced to one	is forced to zero
1	1	1	1	is forced to one	is forced to one

Software output control forces the following values on channels (n) and (n+1) when the COMP bit is one.

Table 35-187. Software ouput control behavior when (COMP = 1)

CH(n)OC	CH(n+1)OC	CH(n)OCV	CH(n+1)OCV	Channel (n) Output	Channel (n+1) Output
0	0	X	X	is not modified by SWOC	is not modified by SWOC
1	1	0	0	is forced to zero	is forced to zero
1	1	0	1	is forced to zero	is forced to one
1	1	1	0	is forced to one	is forced to zero
1	1	1	1	is forced to one	is forced to zero

Note

- The software output control feature must be used only in Combine mode.
- The CH(n)OC and CH(n+1)OC bits should be equal.
- The COMP bit must not be modified when software output control is enabled, that is, CH(n)OC = 1 and/or CH(n+1)OC = 1.
- Software output control has the same behavior with disabled or enabled FTM counter (see the CLKS bitfield description in the Status and Control register).

35.4.14 Deadtime insertion

The deadtime insertion is enabled when (DTEN = 1) and (DTVAL[5:0] is non- zero).

DEADTIME register defines the deadtime delay that can be used for all FTM channels. The DTPS[1:0] bits define the prescaler for the system clock and the DTVAL[5:0] bits define the deadtime modulo, that is, the number of the deadtime prescaler clocks.

The deadtime delay insertion ensures that no two complementary signals (channels (n) and (n+1)) drive the active state at the same time.

If POL(n) = 0, POL(n+1) = 0, and the deadtime is enabled, then when the channel (n) match (FTM counter = C(n)V) occurs, the channel (n) output remains at the low value until the end of the deadtime delay when the channel (n) output is set. Similarly, when the channel (n+1) match (FTM counter = C(n+1)V) occurs, the channel (n+1) output remains at the low value until the end of the deadtime delay when the channel (n+1) output is set. See the following figures.

If POL(n) = 1, POL(n+1) = 1, and the deadtime is enabled, then when the channel (n) match (FTM counter = C(n)V) occurs, the channel (n) output remains at the high value until the end of the deadtime delay when the channel (n) output is cleared. Similarly, when the channel (n+1) match (FTM counter = C(n+1)V) occurs, the channel (n+1) output remains at the high value until the end of the deadtime delay when the channel (n+1) output is cleared.

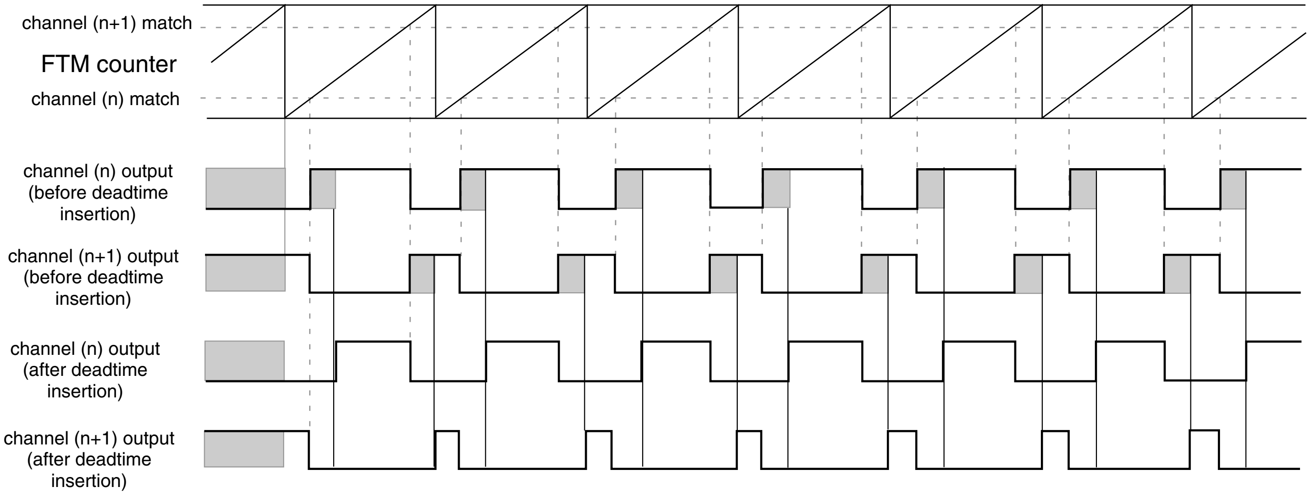

channel (n+1) match
FTM counter
channel (n) match
channel (n) output (before deadtime insertion)
channel (n+1) output (before deadtime insertion)
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)

Figure 35-187. Deadtime insertion with ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 0

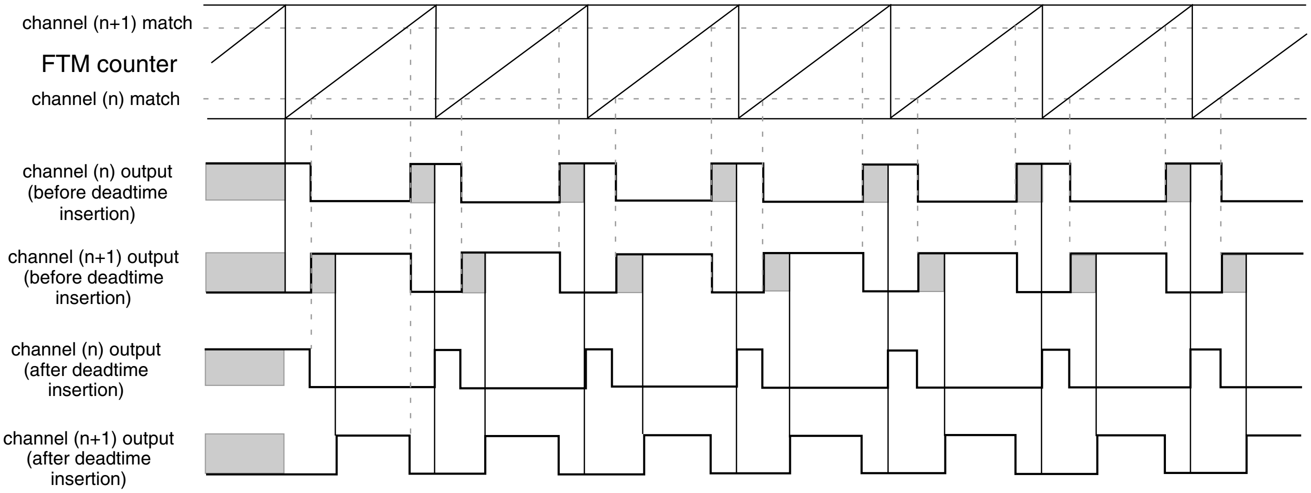

channel (n+1) match
FTM counter
channel (n) match
channel (n) output (before deadtime insertion)
channel (n+1) output (before deadtime insertion)
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)

Figure 35-188. Deadtime insertion with ELSnB:ELSnA = X:1, POL(n) = 0, and POL(n+1) = 0

NOTE

The deadtime feature must be used only in Combine and Complementary modes.

35.4.14.1 Deadtime insertion corner cases

If (PS[2:0] is cleared), (DTPS[1:0] = 0:0 or DTPS[1:0] = 0:1):

- and the deadtime delay is greater than or equal to the channel (n) duty cycle ((C(n+1)V – C(n)V) × system clock), then the channel (n) output is always the inactive value (POL(n) bit value).
- and the deadtime delay is greater than or equal to the channel (n+1) duty cycle ((MOD – CNTIN + 1 – (C(n+1)V – C(n)V)) × system clock), then the channel (n+1) output is always the inactive value (POL(n+1) bit value).

Although, in most cases the deadtime delay is not comparable to channels (n) and (n+1) duty cycle, the following figures show examples where the deadtime delay is comparable to the duty cycle.

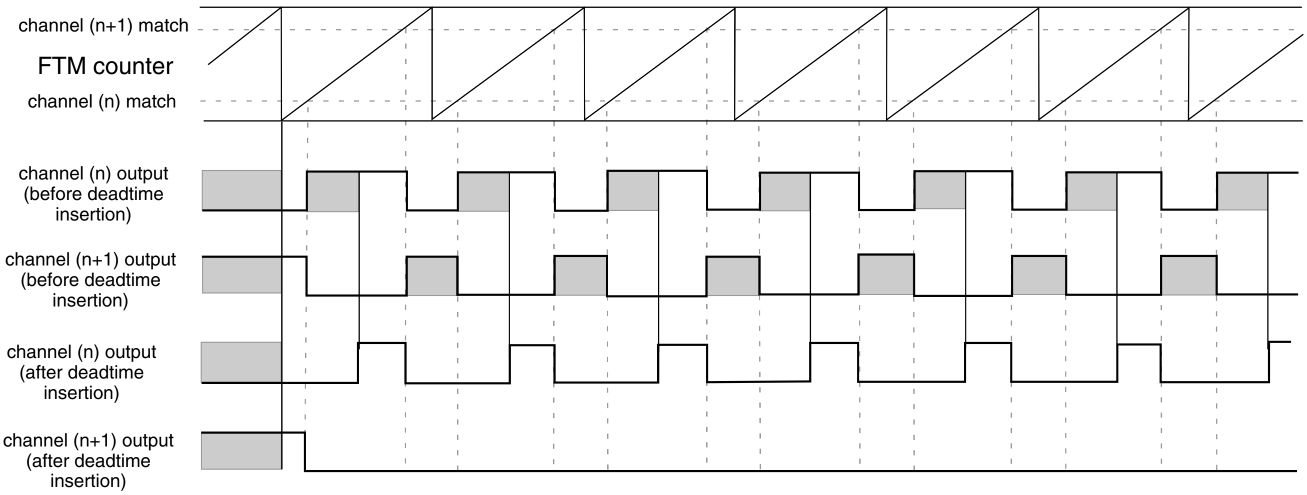

channel (n+1) match
FTM counter
channel (n) match
channel (n) output (before deadtime insertion)
channel (n+1) output (before deadtime insertion)
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)

Figure 35-189. Example of the deadtime insertion (ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 0) when the deadtime delay is comparable to channel (n+1) duty cycle

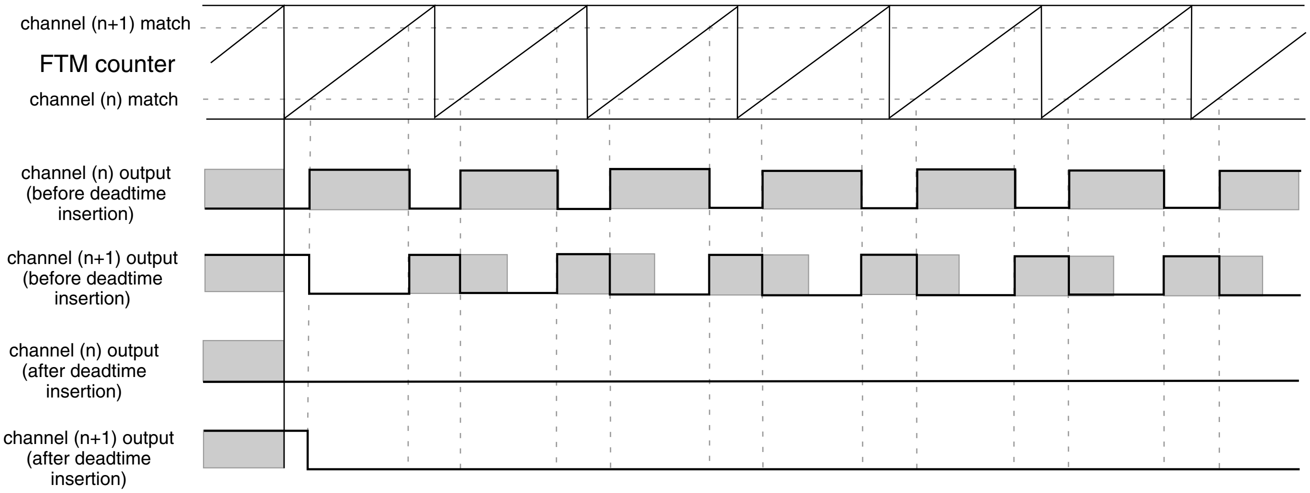

channel (n+1) match
FTM counter
channel (n) match
channel (n) output (before deadtime insertion)
channel (n+1) output (before deadtime insertion)
channel (n) output (after deadtime insertion)
channel (n+1) output (after deadtime insertion)

Figure 35-190. Example of the deadtime insertion (ELSnB:ELSnA = 1:0, POL(n) = 0, and POL(n+1) = 0) when the deadtime delay is comparable to channels (n) and (n+1) duty cycle

35.4.15 Output mask

The output mask can be used to force channels output to their inactive state through software. For example: to control a BLDC motor.

Any write to the OUTMASK register updates its write buffer. The OUTMASK register is updated with its buffer value by PWM synchronization; see OUTMASK register synchronization.

If CHnOM = 1, then the channel (n) output is forced to its inactive state (POLn bit value). If CHnOM = 0, then the channel (n) output is unaffected by the output mask. See the following figure.

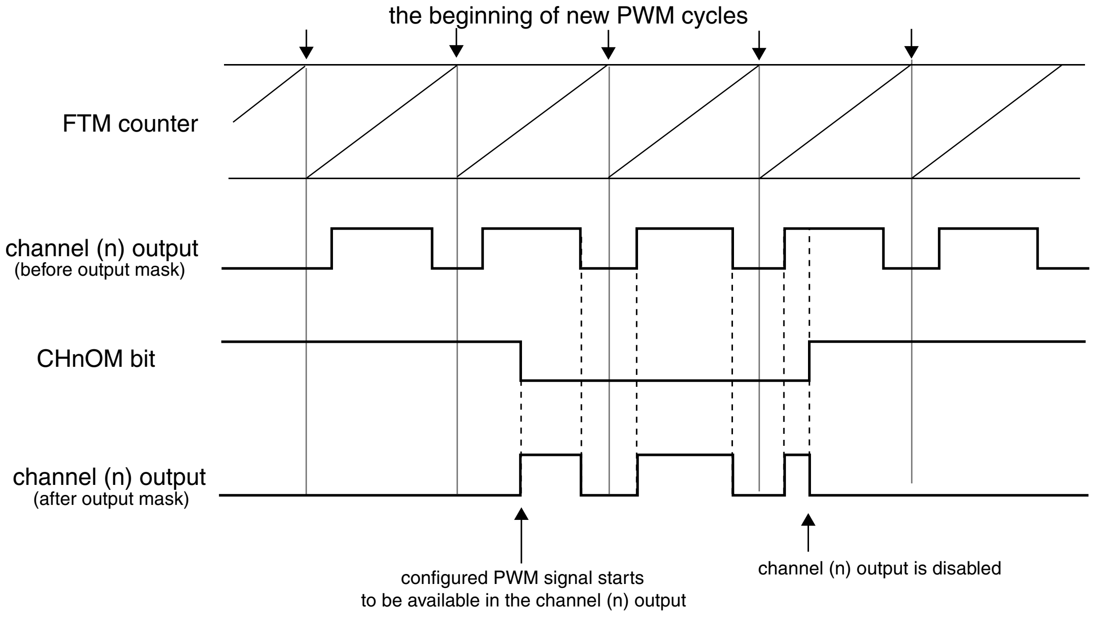

the beginning of new PWM cycles
FTM counter
channel (n) output
(before output mask)
CHnOM bit
channel (n) output
(after output mask)
configured PWM signal starts
to be available in the channel (n) output
channel (n) output is disabled

Figure 35-191. Output mask with POLn = 0

The following table shows the output mask result before the polarity control.

Table 35-188. Output mask result for channel (n) before the polarity control

CHnOM	Output Mask Input	Output Mask Result
0	inactive state	inactive state
	active state	active state
1	inactive state	inactive state
	active state	

Note

The output mask feature must be used only in Combine mode.

35.4.16 Fault control

The fault control is enabled if (FTMEN = 1) and (FAULTM[1:0] ≠ 0:0).

FTM can have up to four fault inputs. FAULTnEN bit (where n = 0, 1, 2, 3) enables the fault input n and FFLTRnEN bit enables the fault input n filter. FFVAL[3:0] bits select the value of the enabled filter in each enabled fault input.

First, each fault input signal is synchronized by the system clock; see the synchronizer block in the following figure. Following synchronization, the fault input n signal enters the filter block. When there is a state change in the fault input n signal, the 5-bit counter is reset and starts counting up. As long as the new state is stable on the fault input n, the counter continues to increment. If the 5-bit counter overflows, that is, the counter exceeds the value of the FFVAL[3:0] bits, the new fault input n value is validated. It is then transmitted as a pulse edge to the edge detector.

If the opposite edge appears on the fault input n signal before validation (counter overflow), the counter is reset. At the next input transition, the counter starts counting again. Any pulse that is shorter than the minimum value selected by FFVAL[3:0] bits (× system clock) is regarded as a glitch and is not passed on to the edge detector.

The fault input n filter is disabled when the FFVAL[3:0] bits are zero or when FAULTnEN = 0. In this case, the fault input n signal is delayed 2 rising edges of the system clock and the FAULTFn bit is set on 3th rising edge of the system clock after a rising edge occurs on the fault input n.

If FFVAL[3:0] ≠ 0000 and FAULTnEN = 1, then the fault input n signal is delayed (3 + FFVAL[3:0]) rising edges of the system clock, that is, the FAULTFn bit is set (4 + FFVAL[3:0]) rising edges of the system clock after a rising edge occurs on the fault input n.

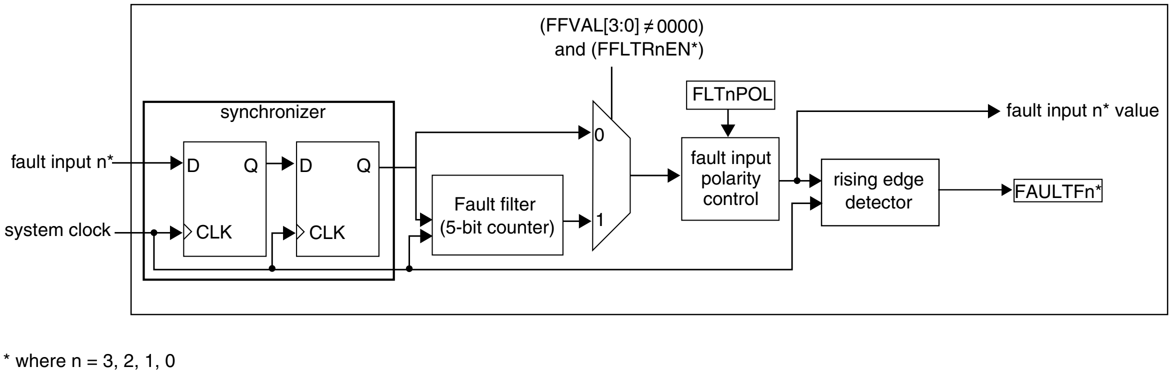

(FFVAL[3:0] ≠ 0000)
and (FFLTRnEN*)
FLTnPOL
fault input n* value
synchronizer
fault input n*
D
Q
D
Q
0
fault input
polarity
control
rising edge
detector
FAULTFn*
Fault filter
(5-bit counter)
system clock
CLK
CLK
1
* where n = 3, 2, 1, 0

Figure 35-192. Fault input n control block diagram

If the fault control and fault input n are enabled and a rising edge at the fault input n signal is detected, a fault condition has occurred and the FAULTFn bit is set. The FAULTF bit is the logic OR of FAULTFn[3:0] bits. See the following figure.

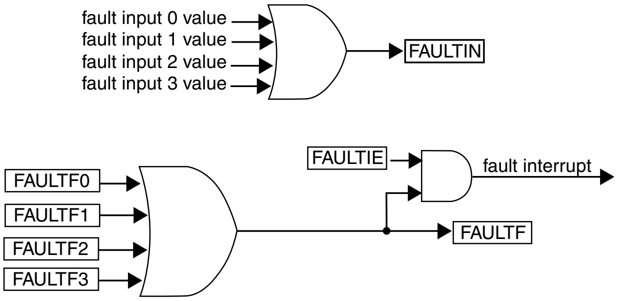

fault input 0 value
fault input 1 value
fault input 2 value
fault input 3 value
FAULTIN
FAULTIE
fault interrupt
FAULTF0
FAULTF1
FAULTF2
FAULTF3
FAULTF

Figure 35-193. FAULTF and FAULTIN bits and fault interrupt

If the fault control is enabled (FAULTM[1:0] ≠ 0:0), a fault condition has occurred and (FAULTEN = 1), then outputs are forced to their safe values:

- Channel (n) output takes the value of POL(n)
- Channel (n+1) takes the value of POL(n+1)

The fault interrupt is generated when (FAULTF = 1) and (FAULTIE = 1). This interrupt request remains set until:

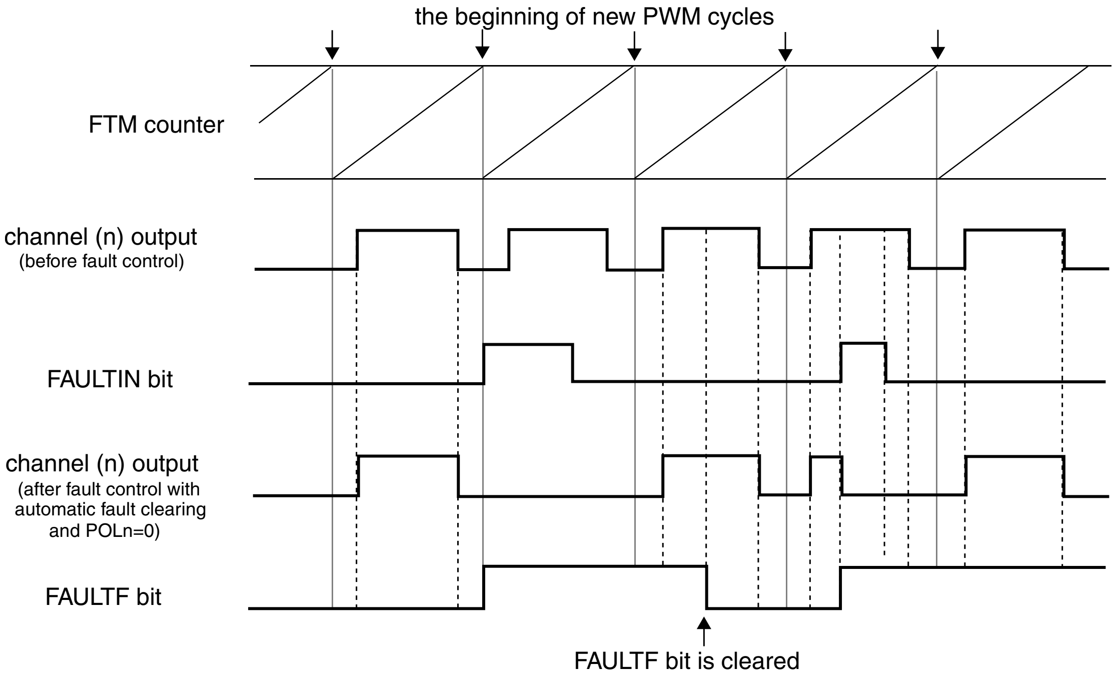
- Software clears the FAULTF bit by reading FAULTF bit as 1 and writing 0 to it
- Software clears the FAULTIE bit
- A reset occurs

Note

The fault control must be used only in Combine mode.

35.4.16.1 Automatic fault clearing

If the automatic fault clearing is selected (FAULTM[1:0] = 1:1), then the channels output disabled by fault control is again enabled when the fault input signal (FAULTIN) returns to zero and a new PWM cycle begins. See the following figure.

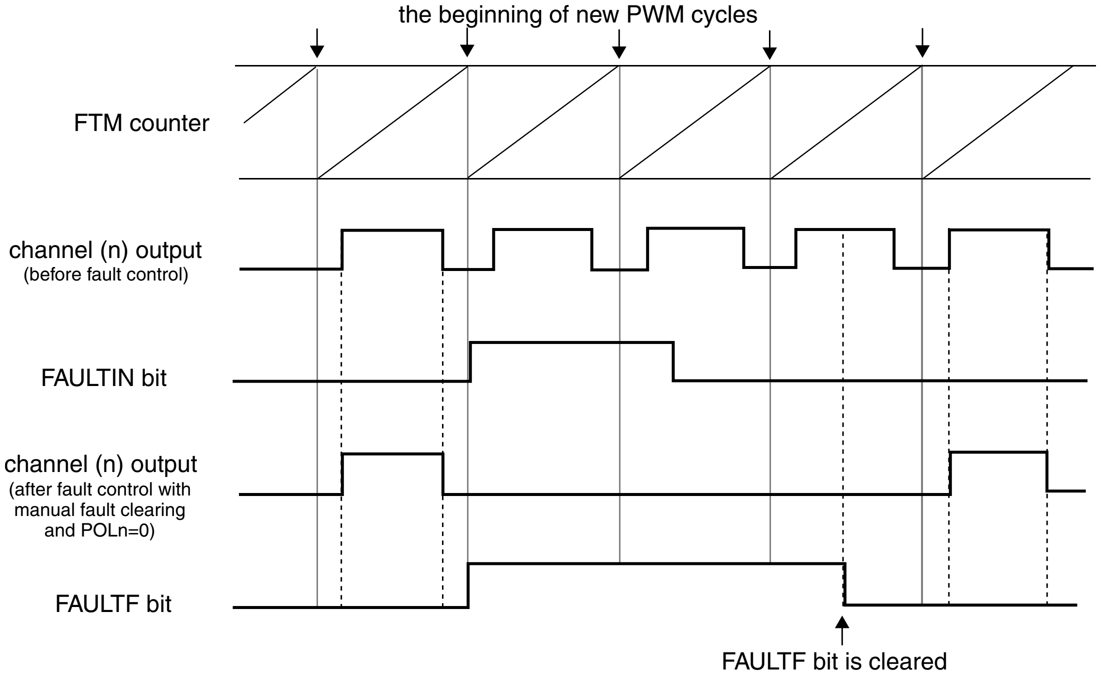

the beginning of new PWM cycles
FTM counter
channel (n) output
(before fault control)
FAULTIN bit
channel (n) output
(after fault control with automatic fault clearing and POLn=0)
FAULTF bit
FAULTF bit is cleared

NOTE
The channel (n) output is after the fault control with automatic fault clearing and POLn = 0.

Figure 35-194. Fault control with automatic fault clearing

35.4.16.2 Manual fault clearing

If the manual fault clearing is selected (FAULTM[1:0] = 0:1 or 1:0), then the channels output disabled by fault control is again enabled when the FAULTF bit is cleared and a new PWM cycle begins. See the following figure.


the beginning of new PWM cycles
FTM counter
channel (n) output
(before fault control)
FAULTIN bit
channel (n) output
(after fault control with
manual fault clearing
and POLn=0)
FAULTF bit
FAULTF bit is cleared

NOTE
The channel (n) output is after the fault control with manual fault clearing and POLn = 0.

Figure 35-195. Fault control with manual fault clearing

35.4.16.3 Fault inputs polarity control

The FLTjPOL bit selects the fault input j polarity, where j = 0, 1, 2, 3:

- If FLTjPOL = 0, the fault j input polarity is high, so the logical one at the fault input j indicates a fault.
- If FLTjPOL = 1, the fault j input polarity is low, so the logical zero at the fault input j indicates a fault.

35.4.17 Polarity control

The POLn bit selects the channel (n) output polarity:

- If POLn = 0, the channel (n) output polarity is high, so the logical one is the active state and the logical zero is the inactive state.
- If POLn = 1, the channel (n) output polarity is low, so the logical zero is the active state and the logical one is the inactive state.

Note

The polarity control must be used only in Combine mode.

35.4.18 Initialization

The initialization forces the CHnOI bit value to the channel (n) output when a one is written to the INIT bit.

The initialization depends on COMP and DTEN bits. The following table shows the values that channels (n) and (n+1) are forced by initialization when the COMP and DTEN bits are zero.

Table 35-189. Initialization behavior when (COMP = 0 and DTEN = 0)

CH(n)OI	CH(n+1)OI	Channel (n) Output	Channel (n+1) Output
0	0	is forced to zero	is forced to zero
0	1	is forced to zero	is forced to one
1	0	is forced to one	is forced to zero
1	1	is forced to one	is forced to one

The following table shows the values that channels (n) and (n+1) are forced by initialization when (COMP = 1) or (DTEN = 1).

Table 35-190. Initialization behavior when (COMP = 1 or DTEN = 1)

CH(n)OI	CH(n+1)OI	Channel (n) Output	Channel (n+1) Output
0	X	is forced to zero	is forced to one
1	X	is forced to one	is forced to zero

Note

The initialization feature must be used only in Combine mode and with disabled FTM counter. See the description of the ../dil/FTM.xml#ftm_sc_clks field in the Status and Control register.

35.4.19 Features priority

The following figure shows the priority of the features used at the generation of channels (n) and (n+1) outputs signals.

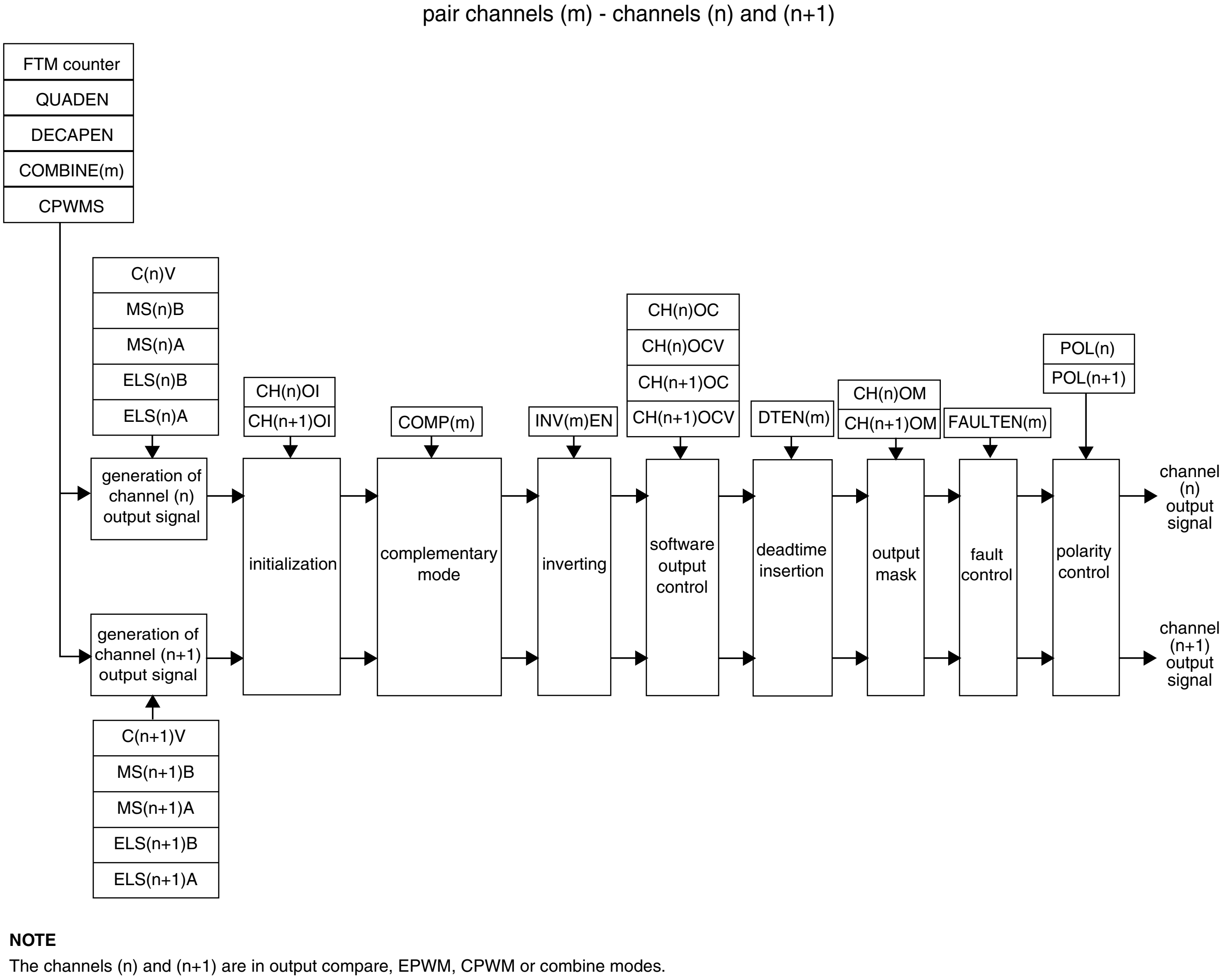

pair channels (m) - channels (n) and (n+1)
FTM counter
QUADEN
DECAPEN
COMBINE(m)
CPWMS
C(n)V
MS(n)B
MS(n)A
ELS(n)B
ELS(n)A
generation of channel (n) output signal
generation of channel (n+1) output signal
C(n+1)V
MS(n+1)B
MS(n+1)A
ELS(n+1)B
ELS(n+1)A
CH(n)OI
CH(n+1)OI
initialization
COMP(m)
complementary mode
INV(m)EN
inverting
CH(n)OC
CH(n)OCV
CH(n+1)OC
CH(n+1)OCV
software output control
DTEN(m)
deadtime insertion
CH(n)OM
CH(n+1)OM
output mask
FAULTEN(m)
fault control
POL(n)
POL(n+1)
polarity control
channel (n) output signal
channel (n+1) output signal
NOTE
The channels (n) and (n+1) are in output compare, EPWM, CPWM or combine modes.

Figure 35-196. Priority of the features used at the generation of channels (n) and (n+1) outputs signals

Note

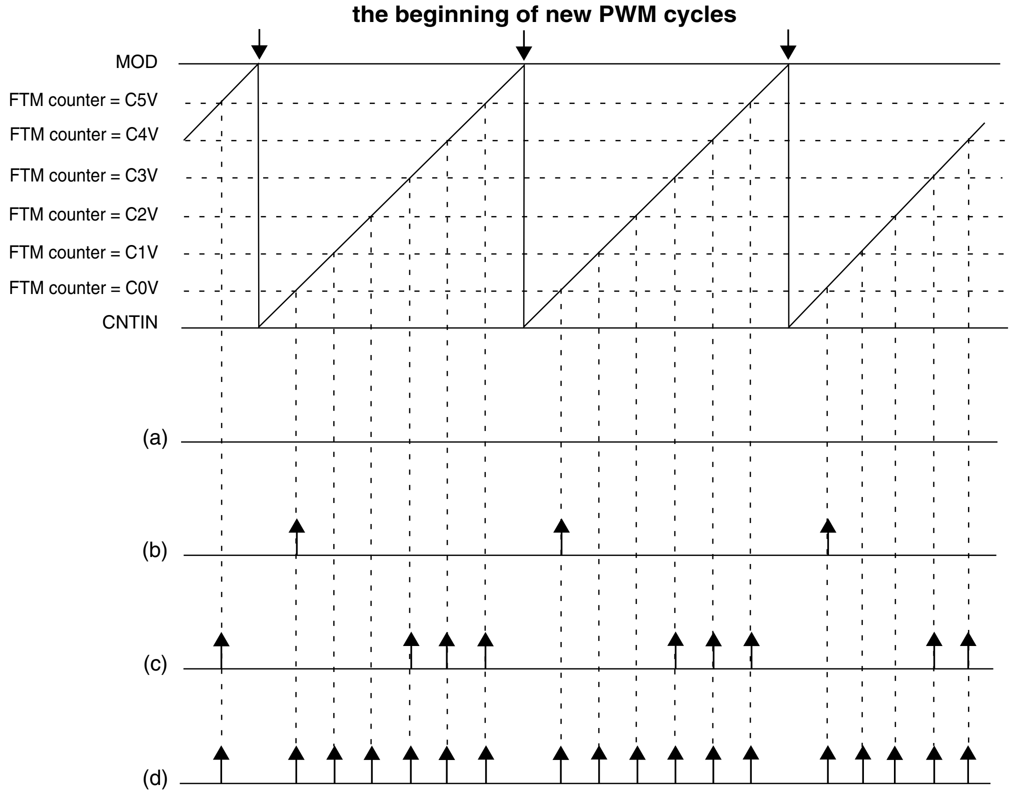
The Initialization feature must not be used with Inverting and Software output control features.

35.4.20 Channel trigger output

If CHjTRIG = 1, where j = 0, 1, 2, 3, 4, or 5, then the FTM generates a trigger when the channel (j) match occurs (FTM counter = C(j)V).

The channel trigger output provides a trigger signal that is used for on-chip modules.

The FTM is able to generate multiple triggers in one PWM period. Because each trigger is generated for a specific channel, several channels are required to implement this functionality. This behavior is described in the following figure.


the beginning of new PWM cycles
MOD
FTM counter = C5V
FTM counter = C4V
FTM counter = C3V
FTM counter = C2V
FTM counter = C1V
FTM counter = C0V
CNTIN
(a)
(b)
(c)
(d)

NOTE

(a) CH0TRIG = 0, CH1TRIG = 0, CH2TRIG = 0, CH3TRIG = 0, CH4TRIG = 0, CH5TRIG = 0
(b) CH0TRIG = 1, CH1TRIG = 0, CH2TRIG = 0, CH3TRIG = 0, CH4TRIG = 0, CH5TRIG = 0
(c) CH0TRIG = 0, CH1TRIG = 0, CH2TRIG = 0, CH3TRIG = 1, CH4TRIG = 1, CH5TRIG = 1
(d) CH0TRIG = 1, CH1TRIG = 1, CH2TRIG = 1, CH3TRIG = 1, CH4TRIG = 1, CH5TRIG = 1

Figure 35-197. Channel match trigger

Note

The channel match trigger must be used only in Combine mode.

35.4.21 Initialization trigger

If INITTRIGEN = 1, then the FTM generates a trigger when the FTM counter is updated with the CNTIN register value in the following cases.

- The FTM counter is automatically updated with the CNTIN register value by the selected counting mode.
- When there is a write to CNT register

- When there is the FTM counter synchronization
- If (CNT = CNTIN), (CLKS[1:0] = 0:0), and a value different from zero is written to CLKS[1:0] bits

The following figures show these cases.

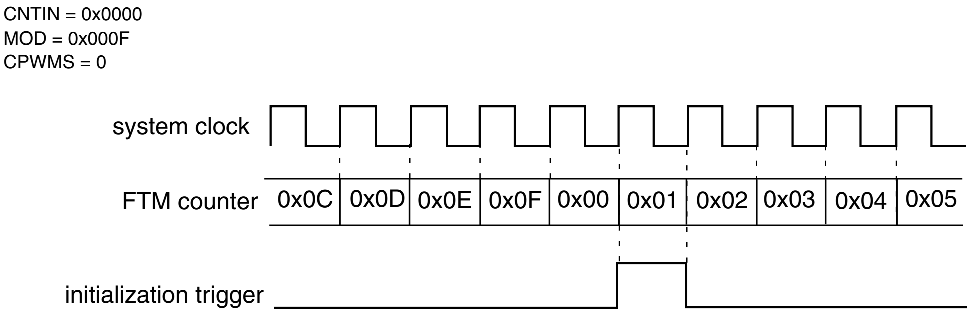

CNTIN = 0x0000
MOD = 0x000F
CPWMS = 0
system clock
FTM counter
0x0C
0x0D
0x0E
0x0F
0x00
0x01
0x02
0x03
0x04
0x05
initialization trigger

Figure 35-198. Initialization trigger is generated when the FTM counting achieves the CNTIN register value

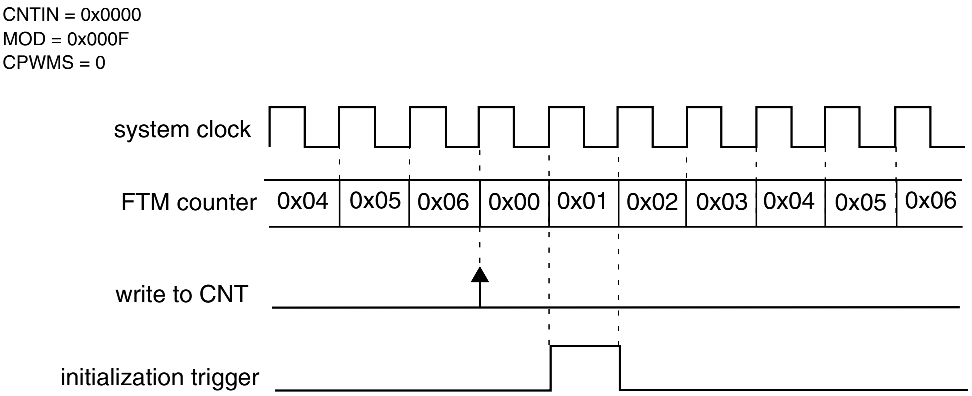

CNTIN = 0x0000
MOD = 0x000F
CPWMS = 0
system clock
FTM counter
0x04
0x05
0x06
0x00
0x01
0x02
0x03
0x04
0x05
0x06
write to CNT
initialization trigger

Figure 35-199. Initialization trigger is generated when there is a write to CNT register

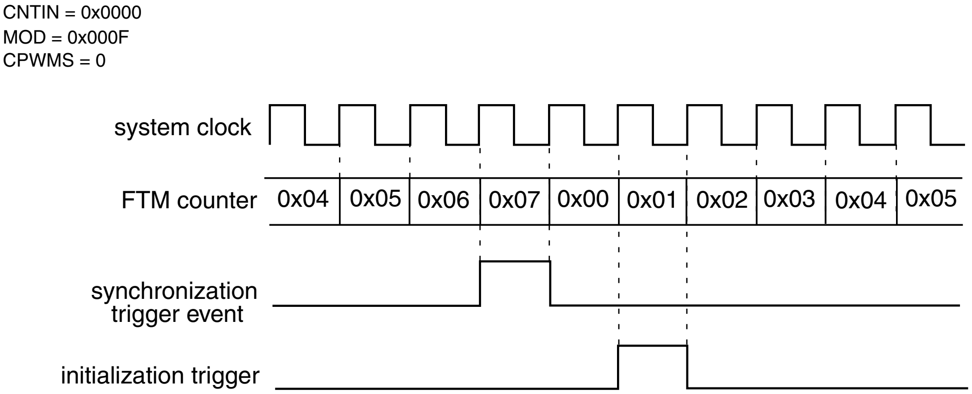

CNTIN = 0x0000
MOD = 0x000F
CPWMS = 0
system clock
FTM counter
0x04
0x05
0x06
0x07
0x00
0x01
0x02
0x03
0x04
0x05
synchronization trigger event
initialization trigger

Figure 35-200. Initialization trigger is generated when there is the FTM counter synchronization

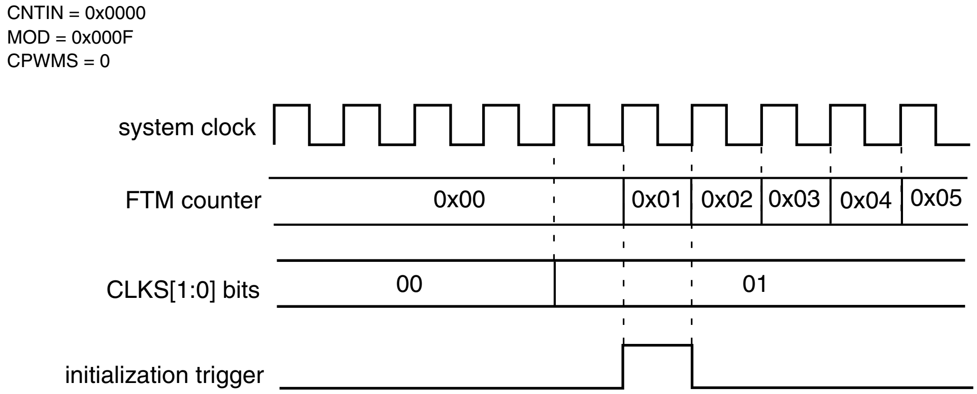


Figure 35-201. Initialization trigger is generated if (CNT = CNTIN), (CLKS[1:0] = 0:0), and a value different from zero is written to CLKS[1:0] bits

The initialization trigger output provides a trigger signal that is used for on-chip modules.

Note

The initialization trigger must be used only in Combine mode.

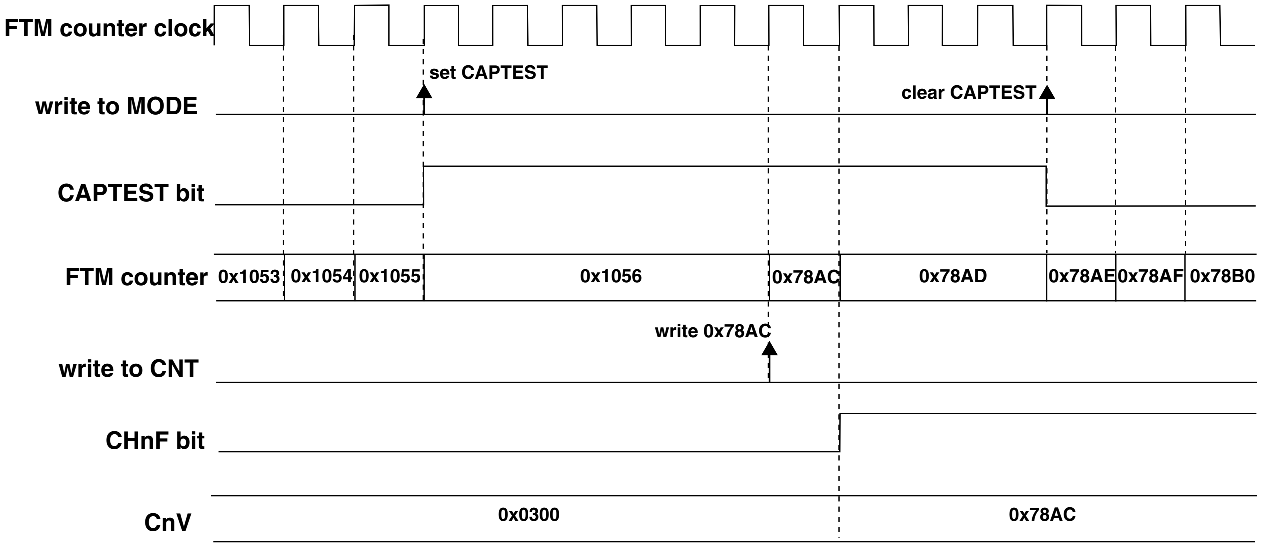
35.4.22 Capture Test mode

The Capture Test mode allows to test the CnV registers, the FTM counter and the interconnection logic between the FTM counter and CnV registers.

In this test mode, all channels must be configured for Input Capture mode and FTM counter must be configured to the Up counting.

When the Capture Test mode is enabled (CAPTEST = 1), the FTM counter is frozen and any write to CNT register updates directly the FTM counter; see the following figure. After it was written, all CnV registers are updated with the written value to CNT register and CHnF bits are set. Therefore, the FTM counter is updated with its next value according to its configuration. Its next value depends on CNTIN, MOD, and the written value to FTM counter.

The next reads of CnV registers return the written value to the FTM counter and the next reads of CNT register return FTM counter next value.


FTM counter clock
write to MODE
set CAPTEST
clear CAPTEST
CAPTEST bit
FTM counter
0x1053
0x1054
0x1055
0x1056
0x78AC
0x78AD
0x78AE
0x78AF
0x78B0
write 0x78AC
write to CNT
CHnF bit
CnV
0x0300
0x78AC

NOTE

- FTM counter configuration: (FTMEN = 1), (QUADEN = 0), (CAPTEST = 1), (CPWMS = 0), (CNTIN = 0x0000), and (MOD = 0xFFFF)
- FTM channel n configuration: input capture mode - (DECAPEN = 0), (COMBINE = 0), and (MSnB:MSnA = 0:0)

Figure 35-202. Capture Test mode

35.4.23 DMA

The channel generates a DMA transfer request according to DMA and CHnIE bits. See the following table.

Table 35-191. Channel DMA transfer request

DMA	CHnIE	Channel DMA Transfer Request	Channel Interrupt
0	0	The channel DMA transfer request is not generated.	The channel interrupt is not generated.
0	1	The channel DMA transfer request is not generated.	The channel interrupt is generated if (CHnF = 1).
1	0	The channel DMA transfer request is not generated.	The channel interrupt is not generated.
1	1	The channel DMA transfer request is generated if (CHnF = 1).	The channel interrupt is not generated.

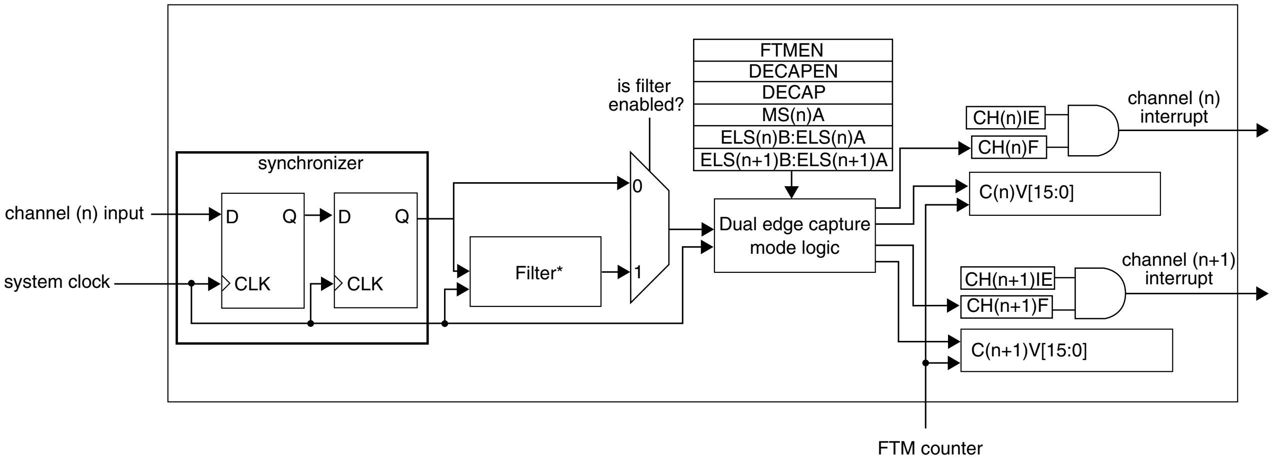
If DMA = 1, the CHnF bit is cleared either by channel DMA transfer done or reading CnSC while CHnF is set and then writing a zero to CHnF bit according to CHnIE bit. See the following table.

Table 35-192. Clear CHnF bit when DMA = 1

CHnIE	How CHnF Bit Can Be Cleared
0	CHnF bit is cleared either when the channel DMA transfer is done or by reading CnSC while CHnF is set and then writing a 0 to CHnF bit.
1	CHnF bit is cleared when the channel DMA transfer is done.

35.4.24 Dual Edge Capture mode

The Dual Edge Capture mode is selected if FTMEN = 1 and DECAPEN = 1. This mode allows to measure a pulse width or period of the signal on the input of channel (n) of a channel pair. The channel (n) filter can be active in this mode when n is 0 or 2.


FTMEN
DECAPEN
DECAP
MS(n)A
ELS(n)B:ELS(n)A
ELS(n+1)B:ELS(n+1)A
is filter enabled?
channel (n) interrupt
CH(n)IE
CH(n)F
synchronizer
channel (n) input
system clock
D
Q
CLK
D
Q
CLK
0
1
Filter*
Dual edge capture mode logic
C(n)V[15:0]
CH(n+1)IE
CH(n+1)F
channel (n+1) interrupt
C(n+1)V[15:0]
FTM counter

* Filtering function for dual edge capture mode is only available in the channels 0 and 2

Figure 35-203. Dual Edge Capture mode block diagram

The MS(n)A bit defines if the Dual Edge Capture mode is one-shot or continuous.

The ELS(n)B:ELS(n)A bits select the edge that is captured by channel (n), and ELS(n+1)B:ELS(n+1)A bits select the edge that is captured by channel (n+1). If both ELS(n)B:ELS(n)A and ELS(n+1)B:ELS(n+1)A bits select the same edge, then it is the period measurement. If these bits select different edges, then it is a pulse width measurement.

In the Dual Edge Capture mode, only channel (n) input is used and channel (n+1) input is ignored.

If the selected edge by channel (n) bits is detected at channel (n) input, then CH(n)F bit is set and the channel (n) interrupt is generated (if CH(n)IE = 1). If the selected edge by channel (n+1) bits is detected at channel (n) input and (CH(n)F = 1), then CH(n+1)F bit is set and the channel (n+1) interrupt is generated (if CH(n+1)IE = 1).

The C(n)V register stores the value of FTM counter when the selected edge by channel (n) is detected at channel (n) input. The C(n+1)V register stores the value of FTM counter when the selected edge by channel (n+1) is detected at channel (n) input.

In this mode, a coherency mechanism ensures coherent data when the C(n)V and C(n +1)V registers are read. The only requirement is that C(n)V must be read before C(n +1)V.

Note

- The CH(n)F, CH(n)IE, MS(n)A, ELS(n)B, and ELS(n)A bits are channel (n) bits.
- The CH(n+1)F, CH(n+1)IE, MS(n+1)A, ELS(n+1)B, and ELS(n+1)A bits are channel (n+1) bits.
- The Dual Edge Capture mode must be used with ELS(n)B:ELS(n)A = 0:1 or 1:0, ELS(n+1)B:ELS(n+1)A = 0:1 or 1:0 and the FTM counter in Free running counter.

35.4.24.1 One-Shot Capture mode

The One-Shot Capture mode is selected when (FTMEN = 1), (DECAPEN = 1), and (MS(n)A = 0). In this capture mode, only one pair of edges at the channel (n) input is captured. The ELS(n)B:ELS(n)A bits select the first edge to be captured, and ELS(n +1)B:ELS(n+1)A bits select the second edge to be captured.

The edge captures are enabled while DECAP bit is set. For each new measurement in One-Shot Capture mode, first the CH(n)F and CH(n+1) bits must be cleared, and then the DECAP bit must be set.

In this mode, the DECAP bit is automatically cleared by FTM when the edge selected by channel (n+1) is captured. Therefore, while DECAP bit is set, the one-shot capture is in process. When this bit is cleared, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers.

Similarly, when the CH(n+1)F bit is set, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers.

35.4.24.2 Continuous Capture mode

The Continuous Capture mode is selected when (FTMEN = 1), (DECAPEN = 1), and (MS(n)A = 1). In this capture mode, the edges at the channel (n) input are captured continuously. The ELS(n)B:ELS(n)A bits select the initial edge to be captured, and ELS(n+1)B:ELS(n+1)A bits select the final edge to be captured.

The edge captures are enabled while DECAP bit is set. For the initial use, first the CH(n)F and CH(n+1)F bits must be cleared, and then DECAP bit must be set to start the continuous measurements.

When the CH(n+1)F bit is set, both edges were captured and the captured values are ready for reading in the C(n)V and C(n+1)V registers. The latest captured values are always available in these registers even after the DECAP bit is cleared.

In this mode, it is possible to clear only the CH(n+1)F bit. Therefore, when the CH(n+1)F bit is set again, the latest captured values are available in C(n)V and C(n+1)V registers.

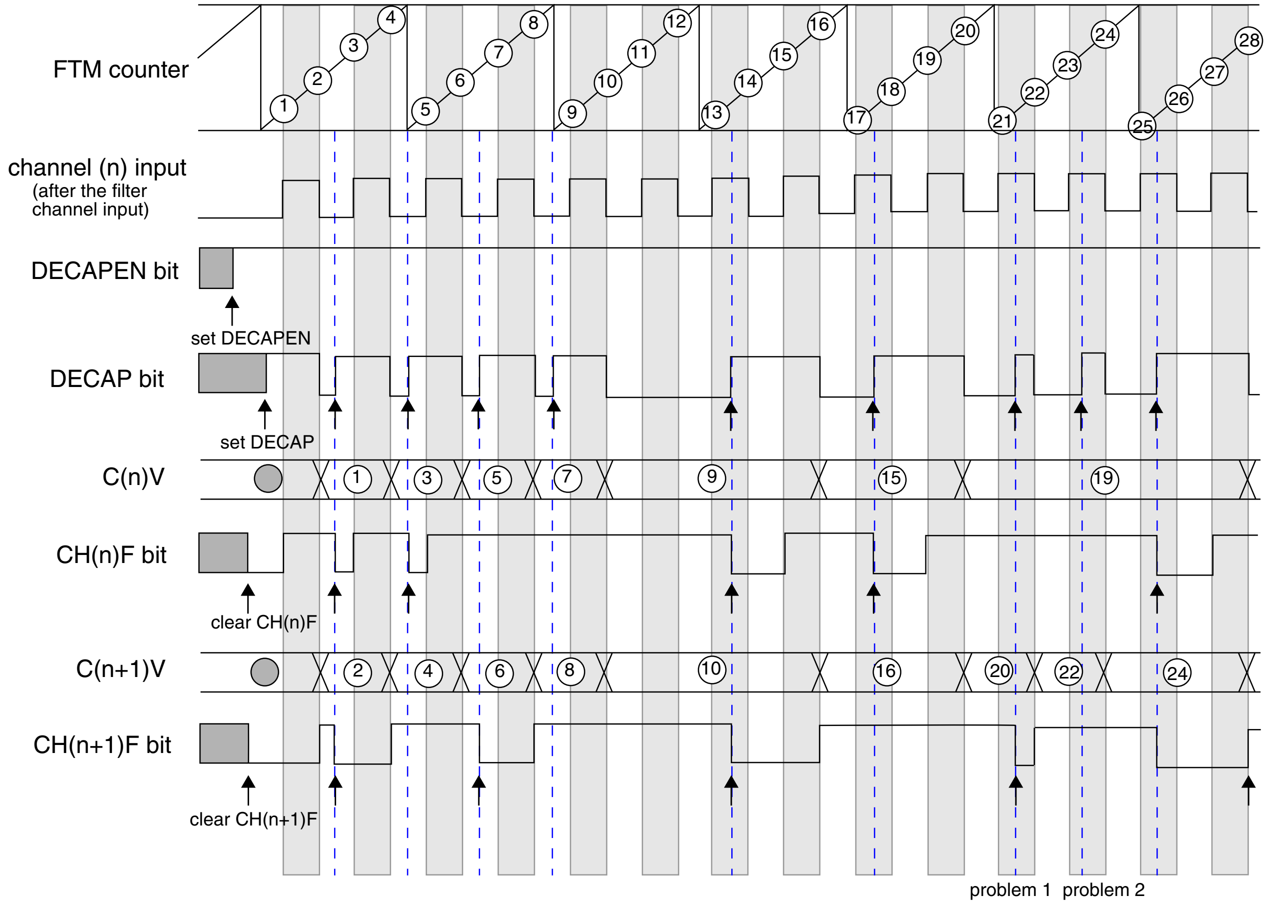
For a new sequence of the measurements in the Dual Edge Capture – Continuous mode, clear the CH(n)F and CH(n+1)F bits to start new measurements.

35.4.24.3 Pulse width measurement

If the channel (n) is configured to capture rising edges (ELS(n)B:ELS(n)A = 0:1) and the channel (n+1) to capture falling edges (ELS(n+1)B:ELS(n+1)A = 1:0), then the positive polarity pulse width is measured. If the channel (n) is configured to capture falling edges (ELS(n)B:ELS(n)A = 1:0) and the channel (n+1) to capture rising edges (ELS(n+1)B:ELS(n+1)A = 0:1), then the negative polarity pulse width is measured.

The pulse width measurement can be made in One-Shot Capture mode or Continuous Capture mode.

The following figure shows an example of the Dual Edge Capture – One-Shot mode used to measure the positive polarity pulse width. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. The DECAP bit is set to enable the measurement of next positive polarity pulse width. The CH(n)F bit is set when the first edge of this pulse is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set and DECAP bit is cleared when the second edge of this pulse is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. Both DECAP and CH(n+1)F bits indicate when two edges of the pulse were captured and the C(n)V and C(n+1)V registers are ready for reading.

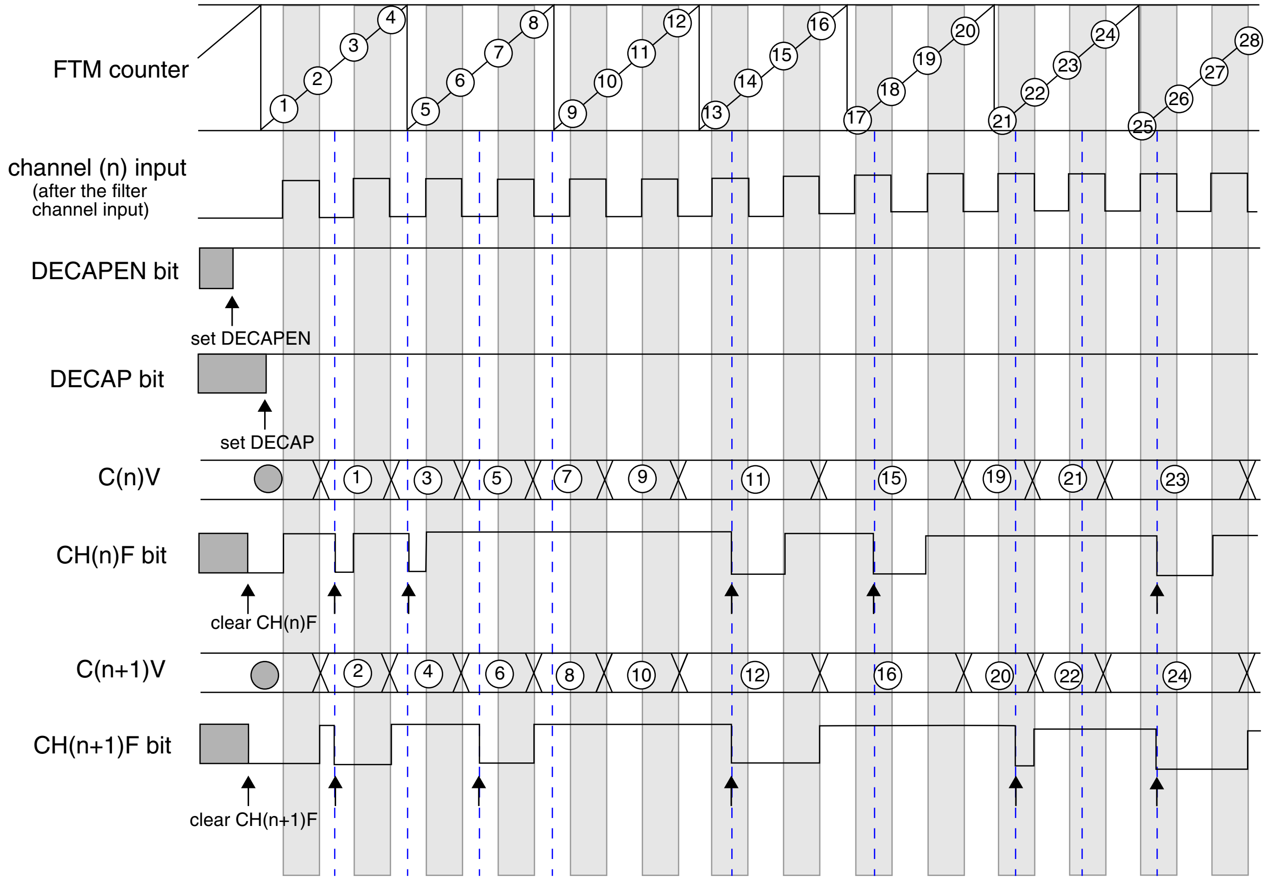

FTM counter
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
channel (n) input
(after the filter channel input)
DECAPEN bit
set DECAPEN
DECAP bit
set DECAP
C(n)V
1
3
5
7
9
15
19
CH(n)F bit
clear CH(n)F
C(n+1)V
2
4
6
8
10
16
20
22
24
CH(n+1)F bit
clear CH(n+1)F
problem 1
problem 2

Note

- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.
- Problem 1: channel (n) input = 1, set DECAP, not clear CH(n)F, and clear CH(n+1)F.
- Problem 2: channel (n) input = 1, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.

Figure 35-204. Dual Edge Capture – One-Shot mode for positive polarity pulse width measurement

The following figure shows an example of the Dual Edge Capture – Continuous mode used to measure the positive polarity pulse width. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. While the DECAP bit is set the configured measurements are made. The CH(n)F bit is set when the first edge of the positive polarity pulse is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set when the second edge of this pulse is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. The CH(n+1)F bit indicates when two edges of the pulse were captured and the C(n)V and C(n+1)V registers are ready for reading.


FTM counter
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
channel (n) input
(after the filter
channel input)
DECAPEN bit
set DECAPEN
DECAP bit
set DECAP
C(n)V
1
3
5
7
9
11
15
19
21
23
CH(n)F bit
clear CH(n)F
C(n+1)V
2
4
6
8
10
12
16
20
22
24
CH(n+1)F bit
clear CH(n+1)F

Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.

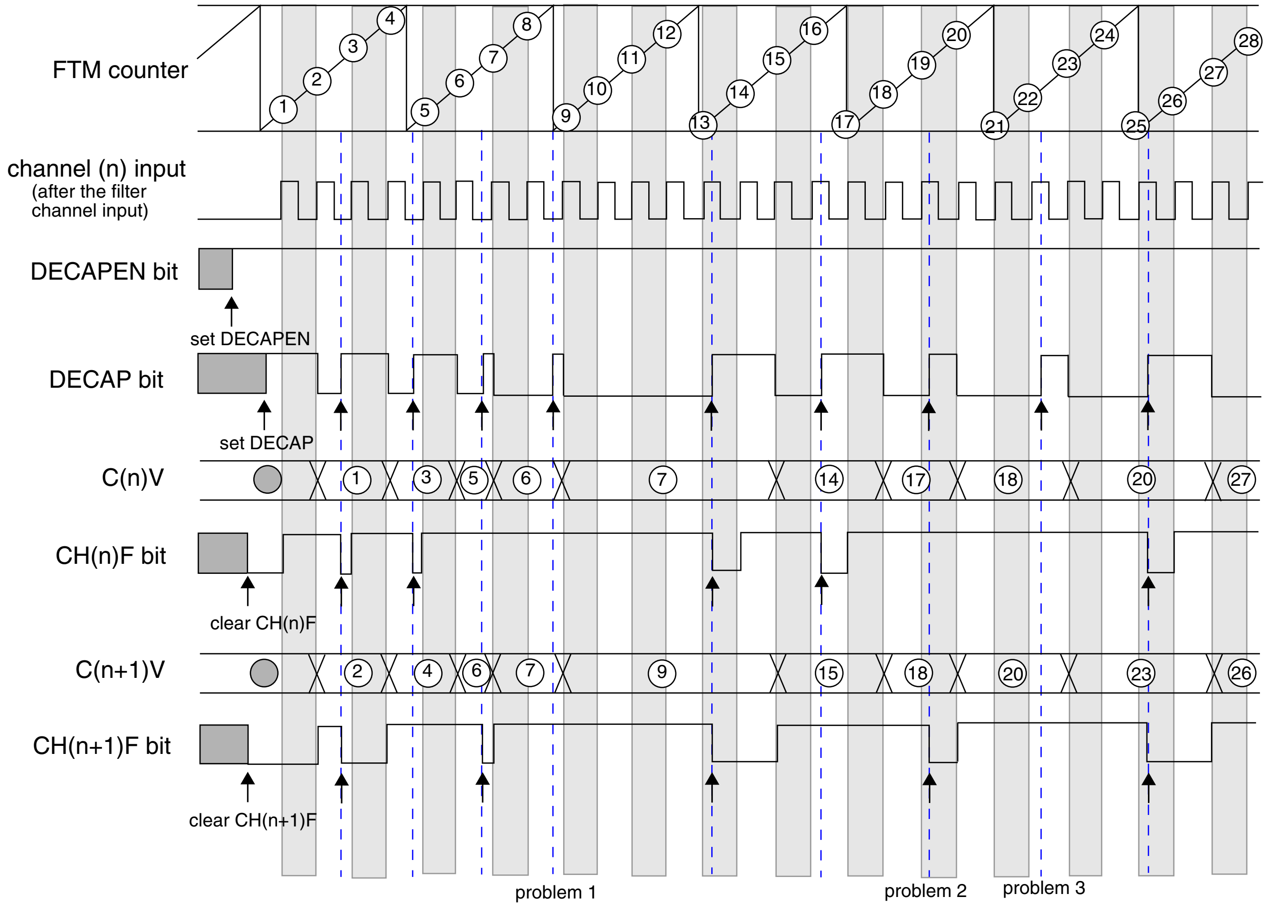
Figure 35-205. Dual Edge Capture – Continuous mode for positive polarity pulse width measurement

35.4.24.4 Period measurement

If the channels (n) and (n+1) are configured to capture consecutive edges of the same polarity, then the period of the channel (n) input signal is measured. If both channels (n) and (n+1) are configured to capture rising edges (ELS(n)B:ELS(n)A = 0:1 and ELS(n+1)B:ELS(n+1)A = 0:1), then the period between two consecutive rising edges is measured. If both channels (n) and (n+1) are configured to capture falling edges (ELS(n)B:ELS(n)A = 1:0 and ELS(n+1)B:ELS(n+1)A = 1:0), then the period between two consecutive falling edges is measured.

The period measurement can be made in One-Shot Capture mode or Continuous Capture mode.

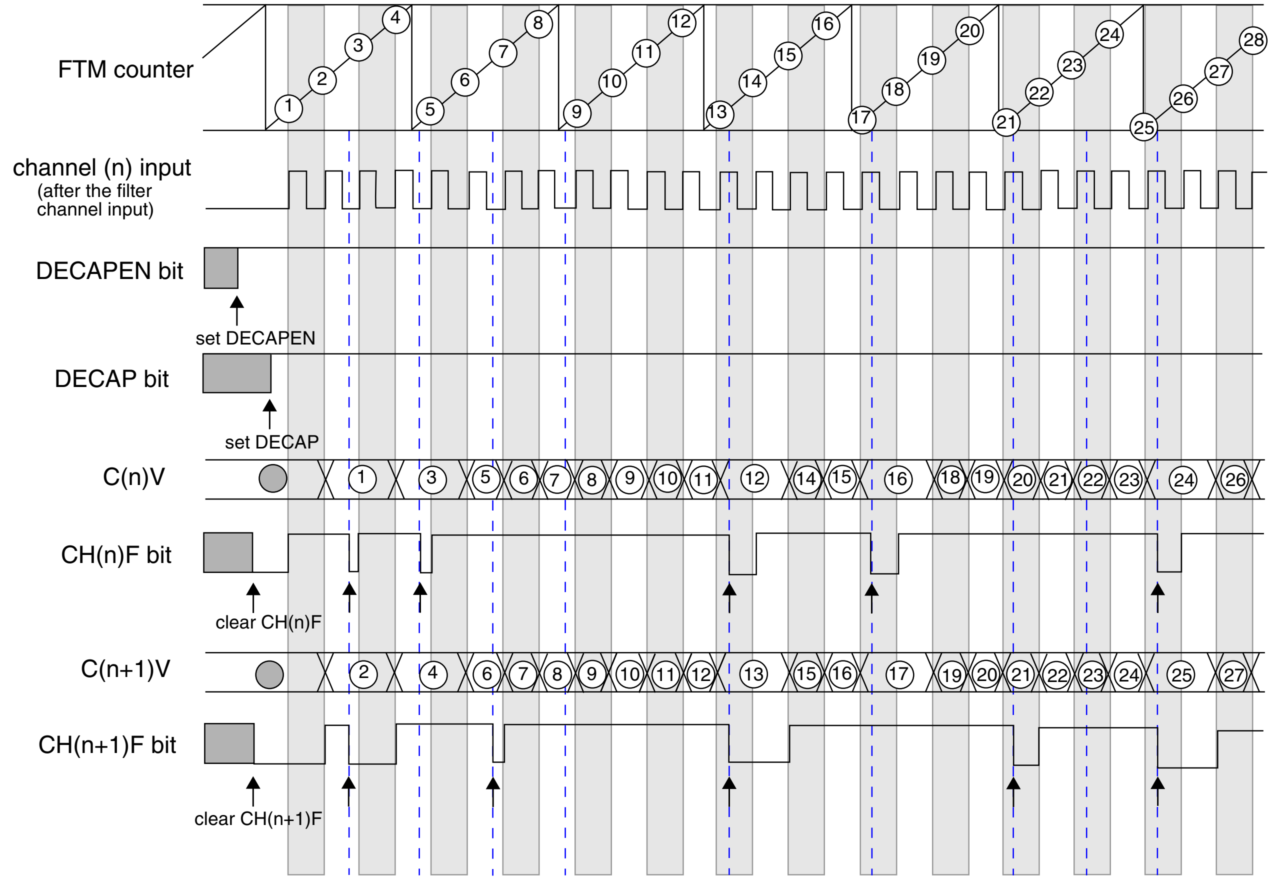
The following figure shows an example of the Dual Edge Capture – One-Shot mode used to measure the period between two consecutive rising edges. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. The DECAP bit is set to enable the measurement of next period. The CH(n)F bit is set when the first rising edge is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set and DECAP bit is cleared when the second rising edge is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. Both DECAP and CH(n+1)F bits indicate when two selected edges were captured and the C(n)V and C(n+1)V registers are ready for reading.


FTM counter
channel (n) input
(after the filter channel input)
DECAPEN bit
set DECAPEN
DECAP bit
set DECAP
C(n)V
CH(n)F bit
clear CH(n)F
C(n+1)V
CH(n+1)F bit
clear CH(n+1)F
problem 1
problem 2
problem 3

Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.
- Problem 1: channel (n) input = 0, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.
- Problem 2: channel (n) input = 1, set DECAP, not clear CH(n)F, and clear CH(n+1)F.
- Problem 3: channel (n) input = 1, set DECAP, not clear CH(n)F, and not clear CH(n+1)F.

Figure 35-206. Dual Edge Capture – One-Shot mode to measure of the period between two consecutive rising edges

The following figure shows an example of the Dual Edge Capture – Continuous mode used to measure the period between two consecutive rising edges. The DECAPEN bit selects the Dual Edge Capture mode, so it remains set. While the DECAP bit is set the configured measurements are made. The CH(n)F bit is set when the first rising edge is detected, that is, the edge selected by ELS(n)B:ELS(n)A bits. The CH(n+1)F bit is set when the second rising edge is detected, that is, the edge selected by ELS(n+1)B:ELS(n+1)A bits. The CH(n+1)F bit indicates when two edges of the period were captured and the C(n)V and C(n+1)V registers are ready for reading.


FTM counter
channel (n) input
(after the filter channel input)
DECAPEN bit
set DECAPEN
DECAP bit
set DECAP
C(n)V
CH(n)F bit
clear CH(n)F
C(n+1)V
CH(n+1)F bit
clear CH(n+1)F

Note
- The commands set DECAPEN, set DECAP, clear CH(n)F, and clear CH(n+1)F are made by the user.

Figure 35-207. Dual Edge Capture – Continuous mode to measure of the period between two consecutive rising edges

35.4.24.5 Read coherency mechanism

The Dual Edge Capture mode implements a read coherency mechanism between the FTM counter value captured in C(n)V and C(n+1)V registers. The read coherency mechanism is illustrated in the following figure. In this example, the channels (n) and (n+1) are in Dual Edge Capture – Continuous mode for positive polarity pulse width measurement. Thus, the channel (n) is configured to capture the FTM counter value when there is a rising edge at channel (n) input signal, and channel (n+1) to capture the FTM counter value when there is a falling edge at channel (n) input signal.

When a rising edge occurs in the channel (n) input signal, the FTM counter value is captured into channel (n) capture buffer. The channel (n) capture buffer value is transferred to C(n)V register when a falling edge occurs in the channel (n) input signal. C(n)V register has the FTM counter value when the previous rising edge occurred, and the channel (n) capture buffer has the FTM counter value when the last rising edge occurred.

When a falling edge occurs in the channel (n) input signal, the FTM counter value is captured into channel (n+1) capture buffer. The channel (n+1) capture buffer value is transferred to C(n+1)V register when the C(n)V register is read.

In the following figure, the read of C(n)V returns the FTM counter value when the event 1 occurred and the read of C(n+1)V returns the FTM counter value when the event 2 occurred.

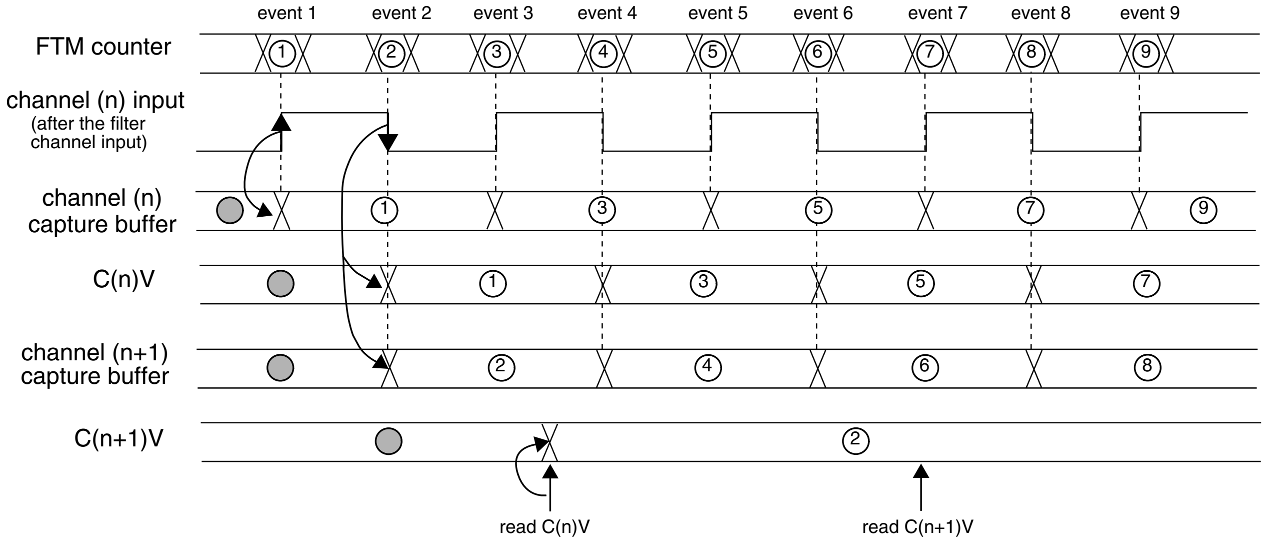

event 1
event 2
event 3
event 4
event 5
event 6
event 7
event 8
event 9
FTM counter
①
②
③
④
⑤
⑥
⑦
⑧
⑨
channel (n) input
(after the filter channel input)
channel (n) capture buffer
C(n)V
channel (n+1) capture buffer
C(n+1)V
read C(n)V
read C(n+1)V

Figure 35-208. Dual Edge Capture mode read coherency mechanism

C(n)V register must be read prior to C(n+1)V register in dual edge capture one-shot and continuous modes for the read coherency mechanism works properly.

35.4.25 Quadrature Decoder mode

The Quadrature Decoder mode is selected if (FTMEN = 1) and (QUADEN = 1). The Quadrature Decoder mode uses the input signals phase A and B to control the FTM counter increment and decrement. The following figure shows the quadrature decoder block diagram.

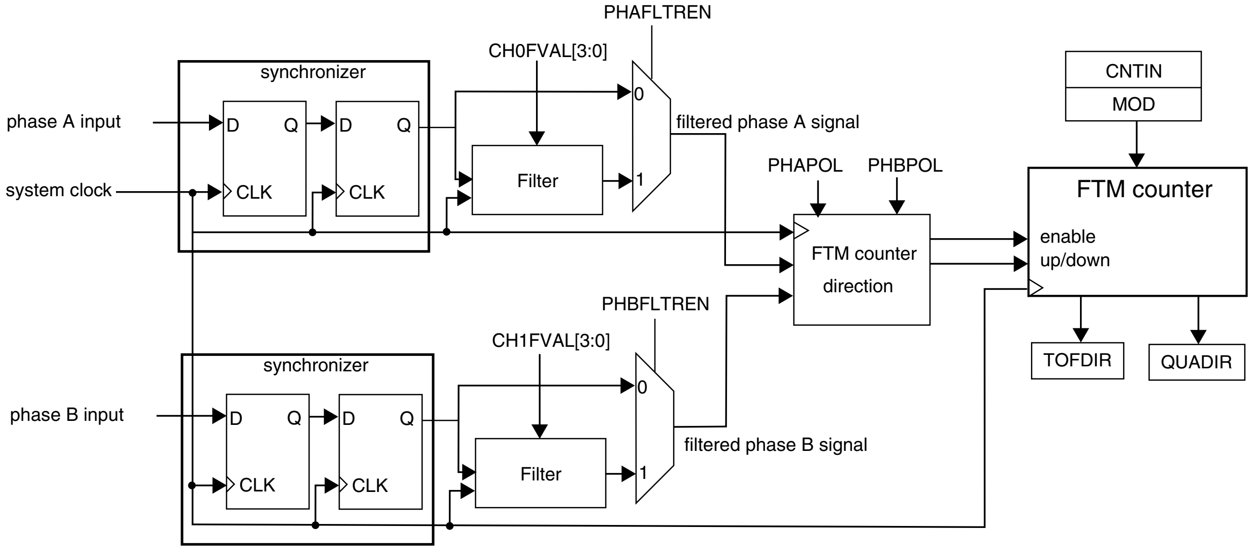

PHAFLTREN
CH0FVAL[3:0]
synchronizer
phase A input
system clock
D
Q
CLK
Filter
0
1
filtered phase A signal
CNTIN
MOD
PHAPOL
PHBPOL
FTM counter
enable
up/down
FTM counter
direction
PHBFLTREN
CH1FVAL[3:0]
TOFDIR
QUADIR
phase B input
filtered phase B signal

Figure 35-209. Quadrature Decoder block diagram

Each one of input signals phase A and B has a filter that is equivalent to the filter used in the channels input; Filter for Input Capture mode. The phase A input filter is enabled by PHAFLTREN bit and this filter's value is defined by CH0FVAL[3:0] bits (CH(n)FVAL[3:0] bits in FILTER0 register). The phase B input filter is enabled by PHBFLTREN bit and this filter's value is defined by CH1FVAL[3:0] bits (CH(n+1)FVAL[3:0] bits in FILTER0 register).

Except for CH0FVAL[3:0] and CH1FVAL[3:0] bits, no channel logic is used in Quadrature Decoder mode.

Note

Notice that the FTM counter is clocked by the phase A and B input signals when quadrature decoder mode is selected. Therefore it is expected that the Quadrature Decoder be used only with the FTM channels in input capture or output compare modes.

The PHAPOL bit selects the polarity of the phase A input, and the PHBPOL bit selects the polarity of the phase B input.

The QUADMODE selects the encoding mode used in the Quadrature Decoder mode. If QUADMODE = 1, then the count and direction encoding mode is enabled; see the following figure. In this mode, the phase B input value indicates the counting direction, and the phase A input defines the counting rate. The FTM counter is updated when there is a rising edge at phase A input signal.

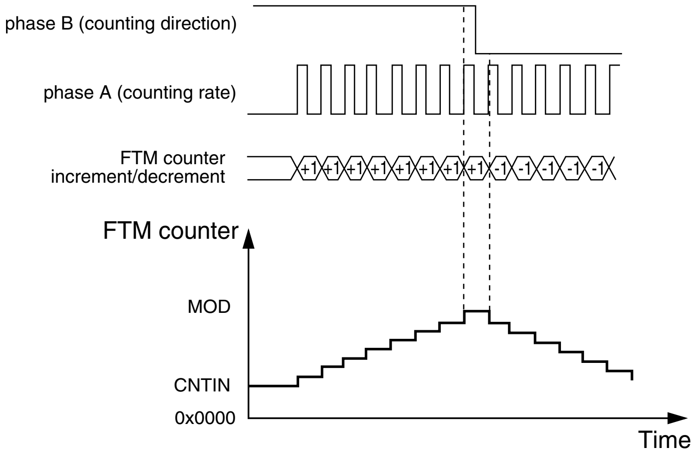

phase B (counting direction)
phase A (counting rate)
FTM counter increment/decrement
+1 +1 +1 +1 +1 +1 +1 +1 -1 -1 -1 -1 -1
FTM counter
MOD
CNTIN
0x0000
Time

Figure 35-210. Quadrature Decoder – Count and Direction Encoding mode

If QUADMODE = 0, then the Phase A and Phase B Encoding mode is enabled; see the following figure. In this mode, the relationship between phase A and B signals indicates the counting direction, and phase A and B signals define the counting rate. The FTM counter is updated when there is an edge either at the phase A or phase B signals.

If PHAPOL = 0 and PHBPOL = 0, then the FTM counter increment happens when:

- there is a rising edge at phase A signal and phase B signal is at logic zero;
- there is a rising edge at phase B signal and phase A signal is at logic one;
- there is a falling edge at phase B signal and phase A signal is at logic zero;
- there is a falling edge at phase A signal and phase B signal is at logic one;

and the FTM counter decrement happens when:

- there is a falling edge at phase A signal and phase B signal is at logic zero;
- there is a falling edge at phase B signal and phase A signal is at logic one;
- there is a rising edge at phase B signal and phase A signal is at logic zero;
- there is a rising edge at phase A signal and phase B signal is at logic one.

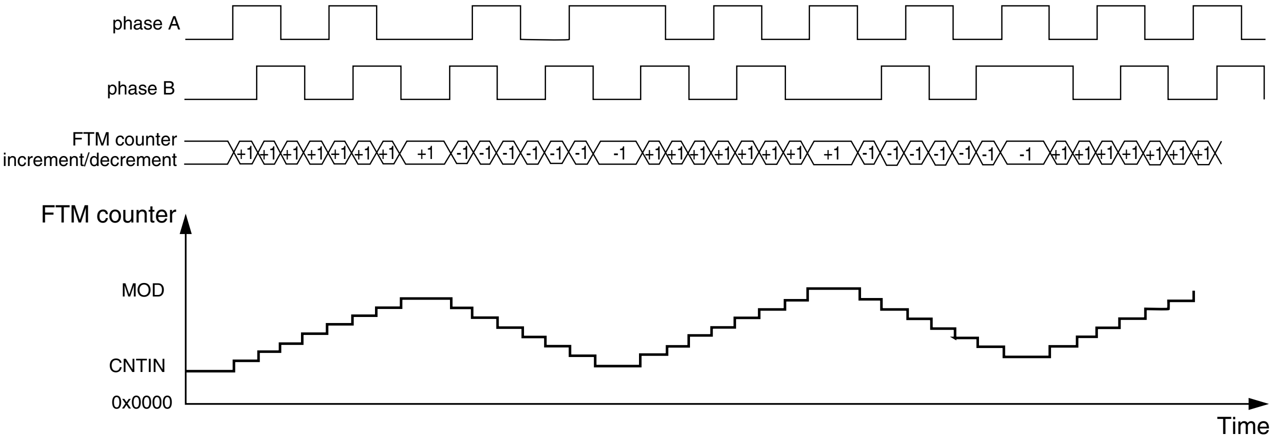

phase A
phase B
FTM counter increment/decrement
+1
-1
FTM counter
MOD
CNTIN
0x0000
Time

Figure 35-211. Quadrature Decoder – Phase A and Phase B Encoding mode

The following figure shows the FTM counter overflow in up counting. In this case, when the FTM counter changes from MOD to CNTIN, TOF and TOFDIR bits are set. TOF bit indicates the FTM counter overflow occurred. TOFDIR indicates the counting was up when the FTM counter overflow occurred.

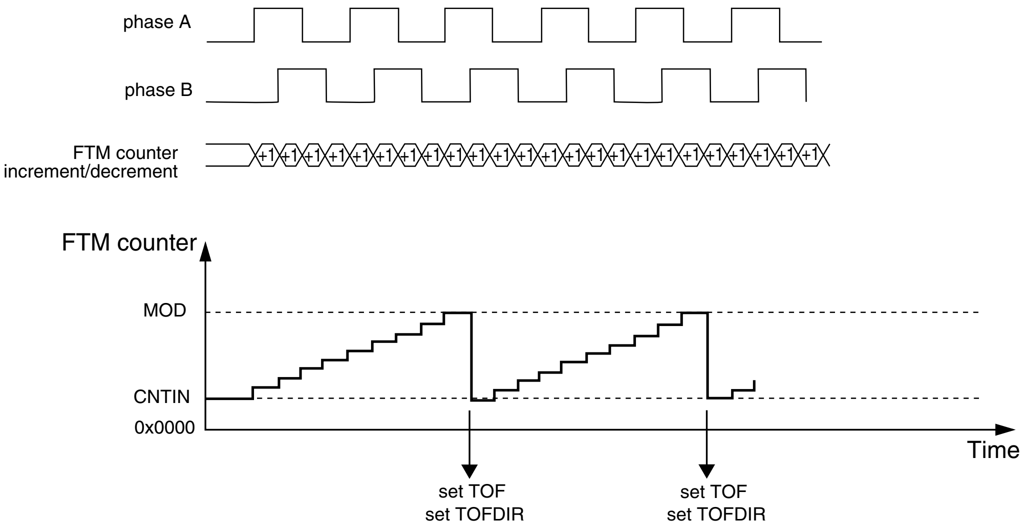

phase A
phase B
FTM counter increment/decrement
+1
FTM counter
MOD
CNTIN
0x0000
Time
set TOF
set TOFDIR
set TOF
set TOFDIR

Figure 35-212. FTM Counter overflow in up counting for Quadrature Decoder mode

The following figure shows the FTM counter overflow in down counting. In this case, when the FTM counter changes from CNTIN to MOD, TOF bit is set and TOFDIR bit is cleared. TOF bit indicates the FTM counter overflow occurred. TOFDIR indicates the counting was down when the FTM counter overflow occurred.

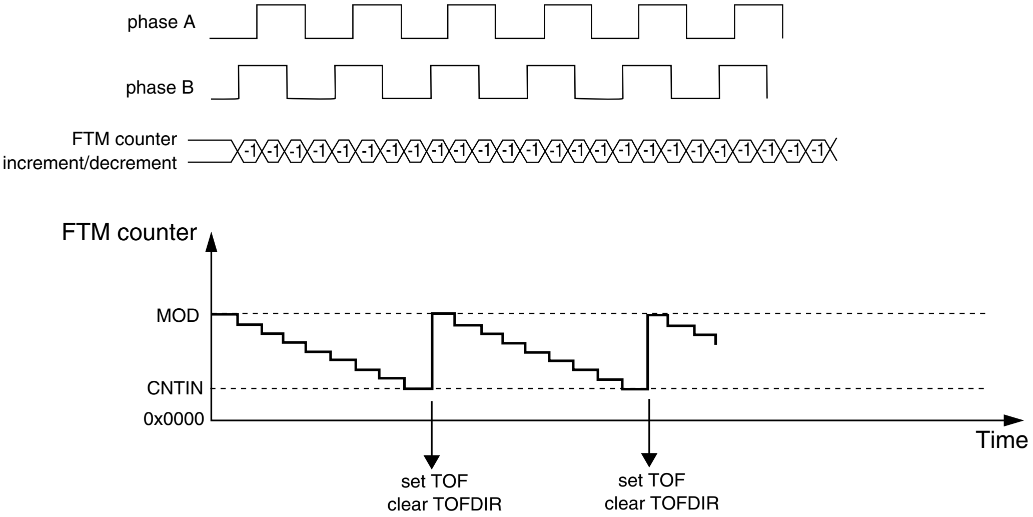

phase A
phase B
FTM counter increment/decrement
-1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
FTM counter
MOD
CNTIN
0x0000
Time
set TOF
clear TOFDIR
set TOF
clear TOFDIR

Figure 35-213. FTM counter overflow in down counting for Quadrature Decoder mode

35.4.25.1 Quadrature Decoder boundary conditions

The following figures show the FTM counter responding to motor jittering typical in motor position control applications.

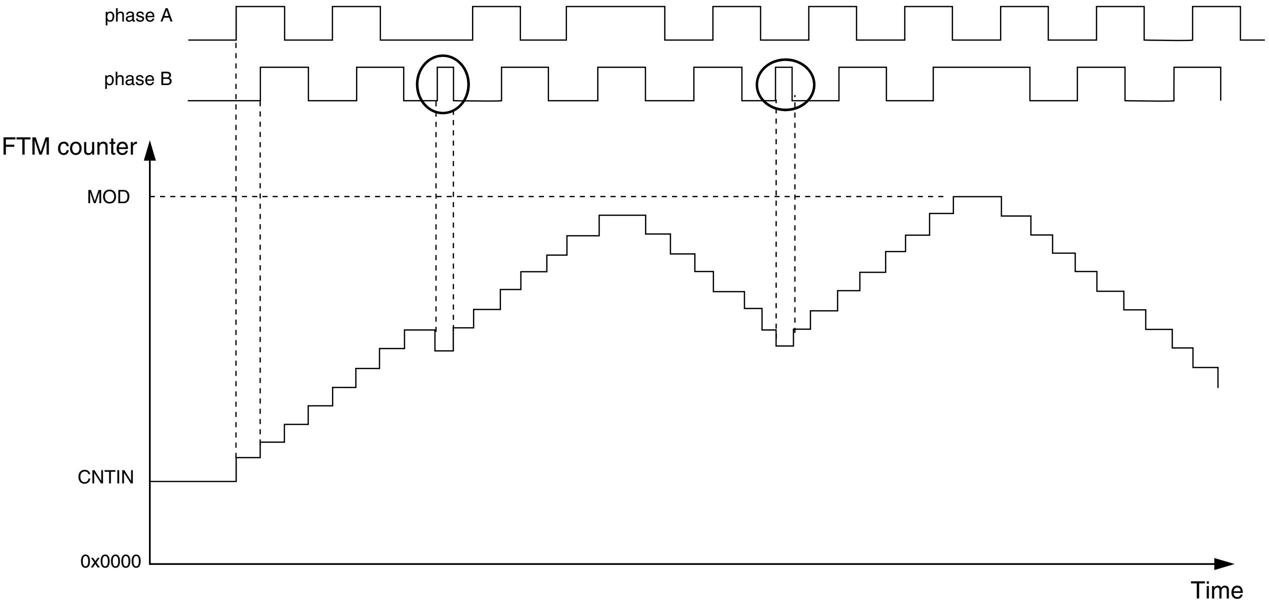

phase A
phase B
FTM counter
MOD
CNTIN
0x0000
Time

Figure 35-214. Motor position jittering in a mid count value

The following figure shows motor jittering produced by the phase B and A pulses respectively:

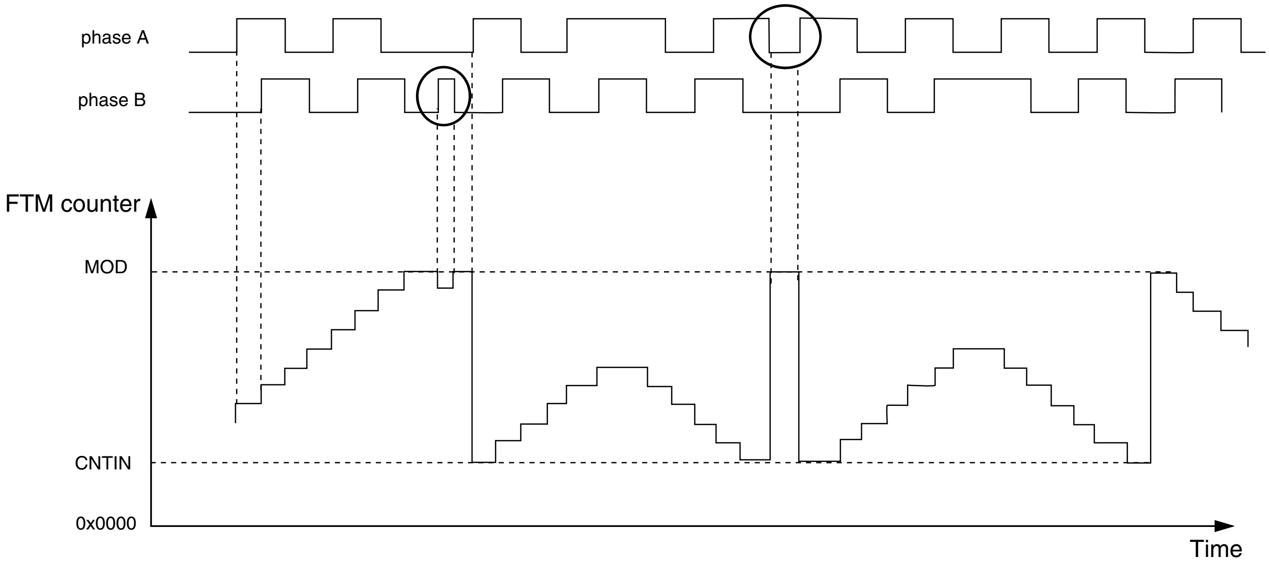


Figure 35-215. Motor position jittering near maximum and minimum count value

The first highlighted transition causes a jitter on the FTM counter value near the maximum count value (MOD). The second indicated transition occurs on phase A and causes the FTM counter transition between the maximum and minimum count values which are defined by MOD and CNTIN registers.

The appropriate settings of the phase A and phase B input filters are important to avoid glitches that may cause oscillation on the FTM counter value. The preceding figures show examples of oscillations that can be caused by poor input filter setup. Thus, it is important to guarantee a minimum pulse width to avoid these oscillations.

35.4.26 BDM mode

When the chip is in BDM mode, the BDMODE[1:0] bits select the behavior of the FTM counter, the CH(n)F bit, the channels output, and the writes to the MOD, CNTIN, and C(n)V registers according to the following table.

Table 35-193. FTM behavior when the chip Is in BDM mode

BDMMODE	FTM Counter	CH(n)F Bit	FTM Channels Output	Writes to MOD, CNTIN, and C(n)V Registers
00	Stopped	can be set	Functional mode	Writes to these registers bypass the registers buffers
01	Stopped	is not set	The channels outputs are forced to their safe value according to POLn bit	Writes to these registers bypass the registers buffers
10	Stopped	is not set	The channels outputs are frozen when the chip enters in BDM mode	Writes to these registers bypass the registers buffers

Table continues on the next page...

Table 35-193. FTM behavior when the chip Is in BDM mode (continued)

BDMMODE	FTM Counter	CH(n)F Bit	FTM Channels Output	Writes to MOD, CNTIN, and C(n)V Registers
11	Functional mode	can be set	Functional mode	Functional mode

Note that if BDMMODE[1:0] = 2'b00 then the channels outputs remain at the value when the chip enters in BDM mode, because the FTM counter is stopped. However, the following situations modify the channels outputs in this BDM mode.

- Write any value to CNT register; see Counter reset. In this case, the FTM counter is updated with the CNTIN register value and the channels outputs are updated to the initial value – except for those channels set to Output Compare mode.
- FTM counter is reset by PWM Synchronization mode; see FTM counter synchronization) In this case, the FTM counter is updated with the CNTIN register value and the channels outputs are updated to the initial value – except for channels in Output Compare mode.
- In the channels outputs initialization, the channel (n) output is forced to the CH(n)OI bit value when the value 1 is written to INIT bit. See Initialization.

Note

The BDMMODE[1:0] = 2'b00 must not be used with the Fault control. Even if the fault control is enabled and a fault condition exists, the channels outputs values are as defined above.

35.4.27 Intermediate load

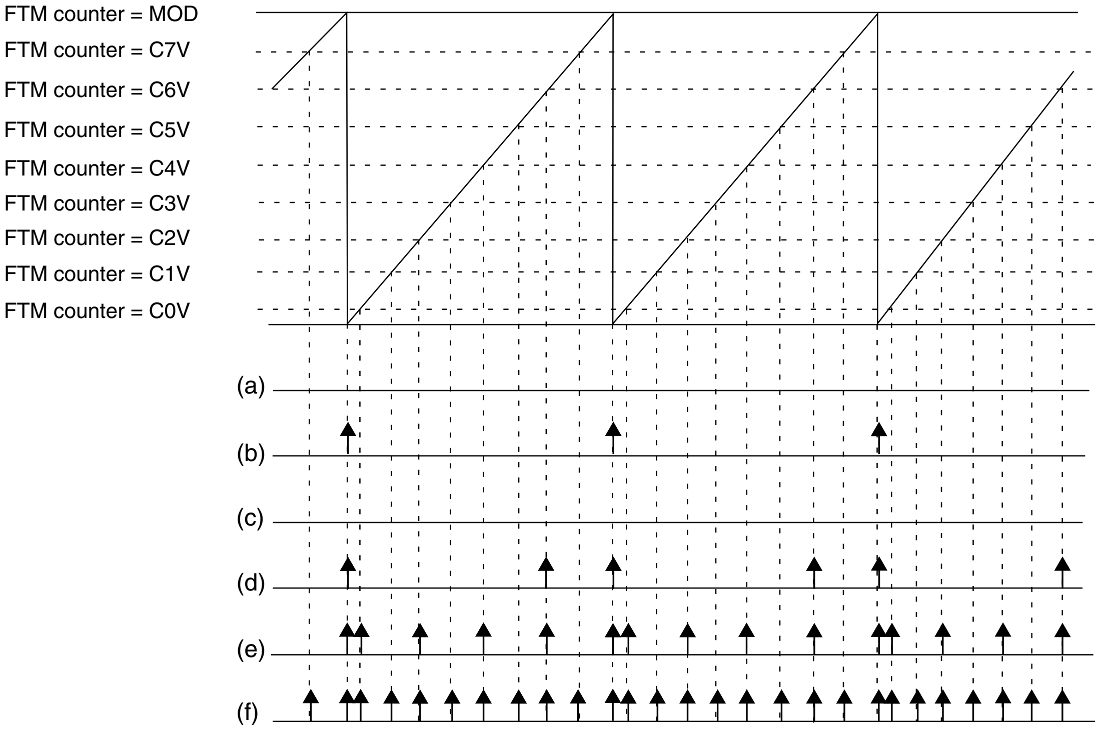
The PWMLOAD register allows software to update the MOD, CNTIN, and C(n)V registers with the content of the register buffer at a defined load point. In this case, it is not required to use the PWM synchronization control.

There are multiple possible loading points for intermediate load:

Table 35-194. When possible loading points are enabled

Loading point	Enabled
When the FTM counter wraps from MOD value to CNTIN value	Always
At the channel (j) match (FTM counter = C(j)V)	When CHjSEL = 1

The following figure shows some examples of enabled loading points.



NOTE

(a) LDOK = 0, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 0, CH4SEL = 0, CH5SEL = 0, CH6SEL = 0, CH7SEL = 0

(b) LDOK = 1, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 0, CH4SEL = 0, CH5SEL = 0, CH6SEL = 0, CH7SEL = 0

(c) LDOK = 0, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 1, CH4SEL = 0, CH5SEL = 0, CH6SEL = 0, CH7SEL = 0

(d) LDOK = 1, CH0SEL = 0, CH1SEL = 0, CH2SEL = 0, CH3SEL = 0, CH4SEL = 0, CH5SEL = 0, CH6SEL = 1, CH7SEL = 0

(e) LDOK = 1, CH0SEL = 1, CH1SEL = 0, CH2SEL = 1, CH3SEL = 0, CH4SEL = 1, CH5SEL = 0, CH6SEL = 1, CH7SEL = 0

(f) LDOK = 1, CH0SEL = 1, CH1SEL = 1, CH2SEL = 1, CH3SEL = 1, CH4SEL = 1, CH5SEL = 1, CH6SEL = 1, CH7SEL = 1

Figure 35-216. Loading points for intermediate load

After enabling the loading points, the LDOK bit must be set for the load to occur. In this case, the load occurs at the next enabled loading point according to the following conditions:

Table 35-195. Conditions for loads occurring at the next enabled loading point

When a new value was written	Then
To the MOD register	The MOD register is updated with its write buffer value.
To the CNTIN register and CNTINC = 1	The CNTIN register is updated with its write buffer value.
To the C(n)V register and SYNCENm = 1 – where m indicates the pair channels (n) and (n+1)	The C(n)V register is updated with its write buffer value.
To the C(n+1)V register and SYNCENm = 1 – where m indicates the pair channels (n) and (n+1)	The C(n+1)V register is updated with its write buffer value.

NOTE

- If ELSjB and ELSjA bits are different from zero, then the channel (j) output signal is generated according to the configured output mode. If ELSjB and ELSjA bits are zero,

then the generated signal is not available on channel (j) output.
- If CHjIE = 1, then the channel (j) interrupt is generated when the channel (j) match occurs.
- At the intermediate load neither the channels outputs nor the FTM counter are changed. Software must set the intermediate load at a safe point in time.
- The intermediate load feature must be used only in Combine mode.

35.4.28 Global time base (GTB)

The global time base (GTB) is a FTM function that allows the synchronization of multiple FTM modules on a chip. The following figure shows an example of the GTB feature used to synchronize two FTM modules. In this case, the FTM A and B channels can behave as if just one FTM module was used, that is, a global time base.

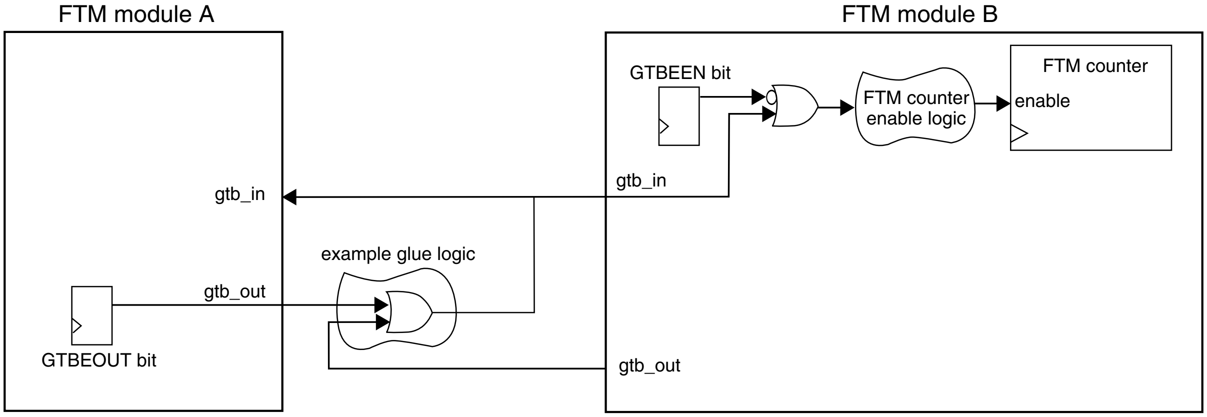


Figure 35-217. Global time base (GTB) block diagram

The GTB functionality is implemented by the GTBEEN and GTBEOUT bits in the CONF register, the internal input signal *gtb_in*, and the internal output signal *gtb_out*. The GTBEEN bit enables gtb_in to control the FTM counter enable signal:

- If GTBEEN = 0, each one of FTM modules works independently according to their configured mode.
- If GTBEEN = 1, the FTM counter update is enabled only when gtb_in is 1.

In the configuration described in the preceding figure, FTM modules A and B have their FTM counters enabled if at least one of the gtb_out signals from one of the FTM modules is 1. There are several possible configurations for the interconnection of the gtb_in and gtb_out signals, represented by the example glue logic shown in the figure. Note that these configurations are chip-dependent and implemented outside of the FTM modules. See the chip configuration details for the chip's specific implementation.

NOTE

- In order to use the internal GTB signals to synchronize the FTM counter of different FTM modules, the configuration of each FTM module should guarantee that its FTM counter starts counting as soon as the gtb_in signal is 1.
- The GTB feature does not provide continuous synchronization of FTM counters, meaning that the FTM counters may lose synchronization during FTM operation. The GTB feature only allows the FTM counters to *start* their operation synchronously.

35.4.28.1 Enabling the global time base (GTB)

To enable the GTB feature, follow these steps for each participating FTM module:

1. Stop the FTM counter: Write 00b to SC[CLKS].
2. Program the FTM module to the intended configuration. The operation mode needs to be consistent across all participating modules.
3. Write 1 to CONF[GTBEEN] and write 0 to CONF[GTBEOUT] at the same time.
4. Select the intended FTM counter clock source in SC[CLKS]. The clock source needs to be consistent across all participating modules.
5. Reset the FTM counter: Write any value to the CNT register.

To initiate the GTB feature, follow these steps for the FTM module used as the time base:

1. Write 1 to CONF[GTBEOUT].
2. If needed, configure the GTB glue logic connecting the FTM modules within the chip. Some chips do not require configuration of glue logic. See the chip configuration details for the chip's specific implementation.

35.5 Reset overview

The FTM is reset whenever any chip reset occurs.

When the FTM exits from reset:

- the FTM counter and the prescaler counter are zero and are stopped (CLKS[1:0] = 00b);

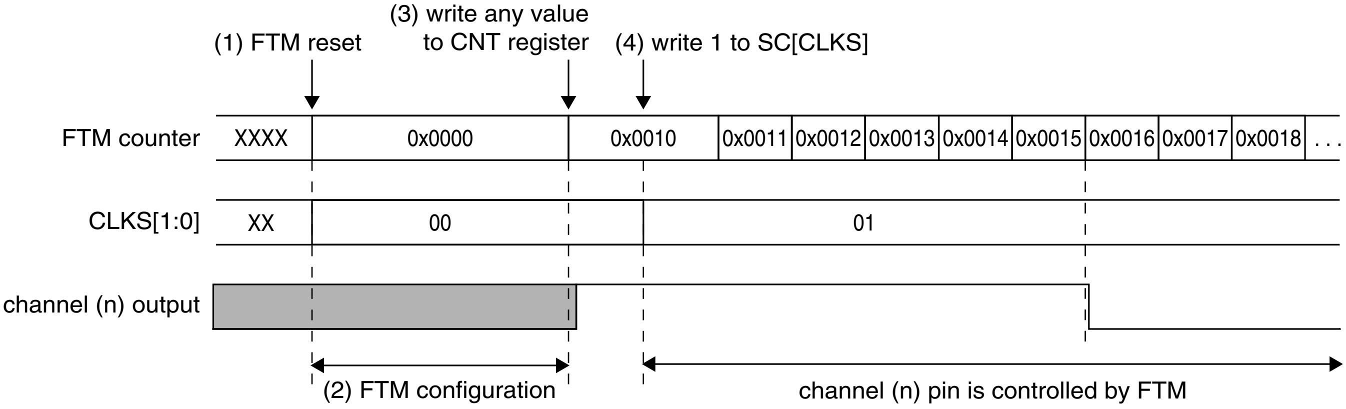
- the timer overflow interrupt is zero, see Timer Overflow Interrupt;
- the channels interrupts are zero, see Channel (n) Interrupt;
- the fault interrupt is zero, see Fault Interrupt;
- the channels are in input capture mode, see Input Capture mode;
- the channels outputs are zero;
- the channels pins are not controlled by FTM (ELS(n)B:ELS(n)A = 0:0) (../dil/FTM.xml#ModeSel1Table).

The following figure shows the FTM behavior after the reset. At the reset (item 1), the FTM counter is disabled (see the description of the ../dil/FTM.xml#ftm_sc_clks field in the Status and Control register), its value is updated to zero and the pins are not controlled by FTM (../dil/FTM.xml#ModeSel1Table).

After the reset, the FTM should be configurated (item 2). It is necessary to define the FTM counter mode, the FTM counting limits (MOD and CNTIN registers value), the channels mode and CnV registers value according to the channels mode.

Thus, it is recommended to write any value to CNT register (item 3). This write updates the FTM counter with the CNTIN register value and the channels output with its initial value (except for channels in output compare mode) (Counter reset).

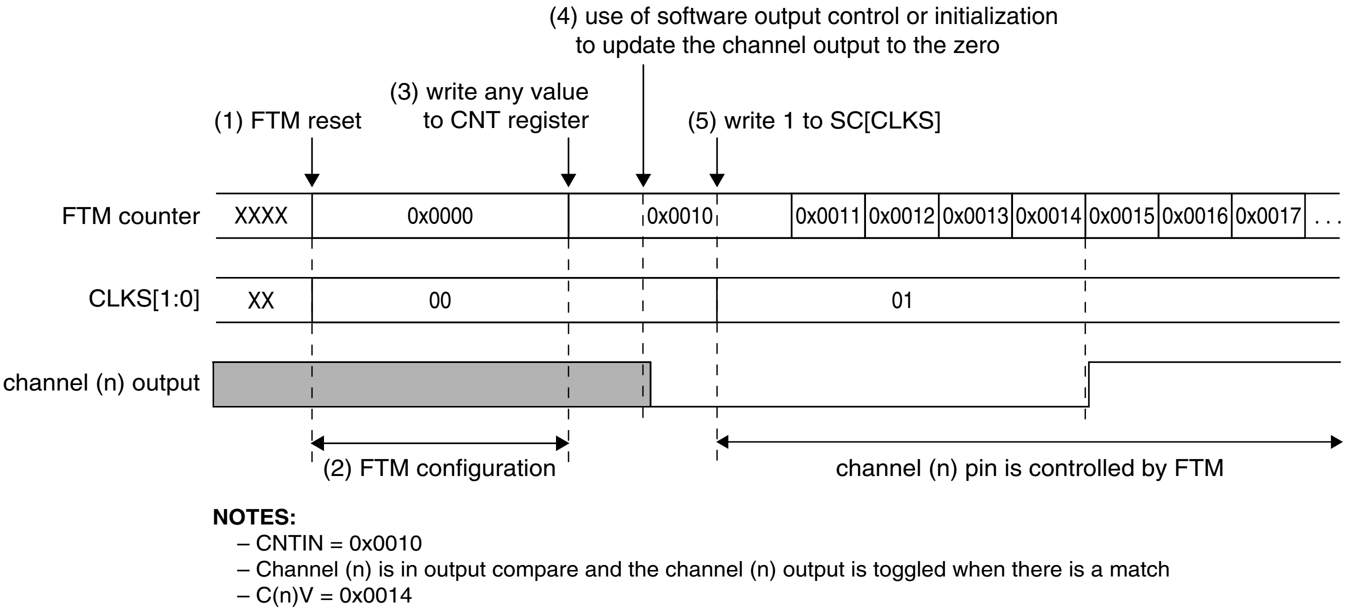
The next step is to select the FTM counter clock by the CLKS[1:0] bits (item 4). It is important to highlight that the pins are only controlled by FTM when CLKS[1:0] bits are different from zero (../dil/FTM.xml#ModeSel1Table).


(1) FTM reset
(3) write any value to CNT register
(4) write 1 to SC[CLKS]
FTM counter
XXXX
0x0000
0x0010
0x0011
0x0012
0x0013
0x0014
0x0015
0x0016
0x0017
0x0018
. . .
CLKS[1:0]
XX
00
01
channel (n) output
(2) FTM configuration
channel (n) pin is controlled by FTM

NOTES:
– CNTIN = 0x0010
– Channel (n) is in low-true combine mode with CNTIN < C(n)V < C(n+1)V < MOD
– C(n)V = 0x0015

Figure 35-218. FTM behavior after reset when the channel (n) is in Combine mode

The following figure shows an example when the channel (n) is in Output Compare mode and the channel (n) output is toggled when there is a match. In the Output Compare mode, the channel output is not updated to its initial value when there is a write to CNT register (item 3). In this case, use the software output control (Software output control) or the initialization (Initialization) to update the channel output to the selected value (item 4).


(4) use of software output control or initialization to update the channel output to the zero
(1) FTM reset
(3) write any value to CNT register
(5) write 1 to SC[CLKS]
FTM counter
XXXX
0x0000
0x0010
0x0011
0x0012
0x0013
0x0014
0x0015
0x0016
0x0017
. . .
CLKS[1:0]
XX
00
01
channel (n) output
(2) FTM configuration
channel (n) pin is controlled by FTM

NOTES:
- CNTIN = 0x0010
- Channel (n) is in output compare and the channel (n) output is toggled when there is a match
- C(n)V = 0x0014

Figure 35-219. FTM behavior after reset when the channel (n) is in Output Compare mode

35.6 FTM Interrupts

35.6.1 Timer Overflow Interrupt

The timer overflow interrupt is generated when (TOIE = 1) and (TOF = 1).

35.6.2 Channel (n) Interrupt

The channel (n) interrupt is generated when (CHnIE = 1) and (CHnF = 1).

35.6.3 Fault Interrupt

The fault interrupt is generated when (FAULTIE = 1) and (FAULTF = 1).

Chapter 36
Periodic Interrupt Timer (PIT)

36.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The PIT module is an array of timers that can be used to raise interrupts and trigger DMA channels.

36.1.1 Block diagram

The following figure shows the block diagram of the PIT module.

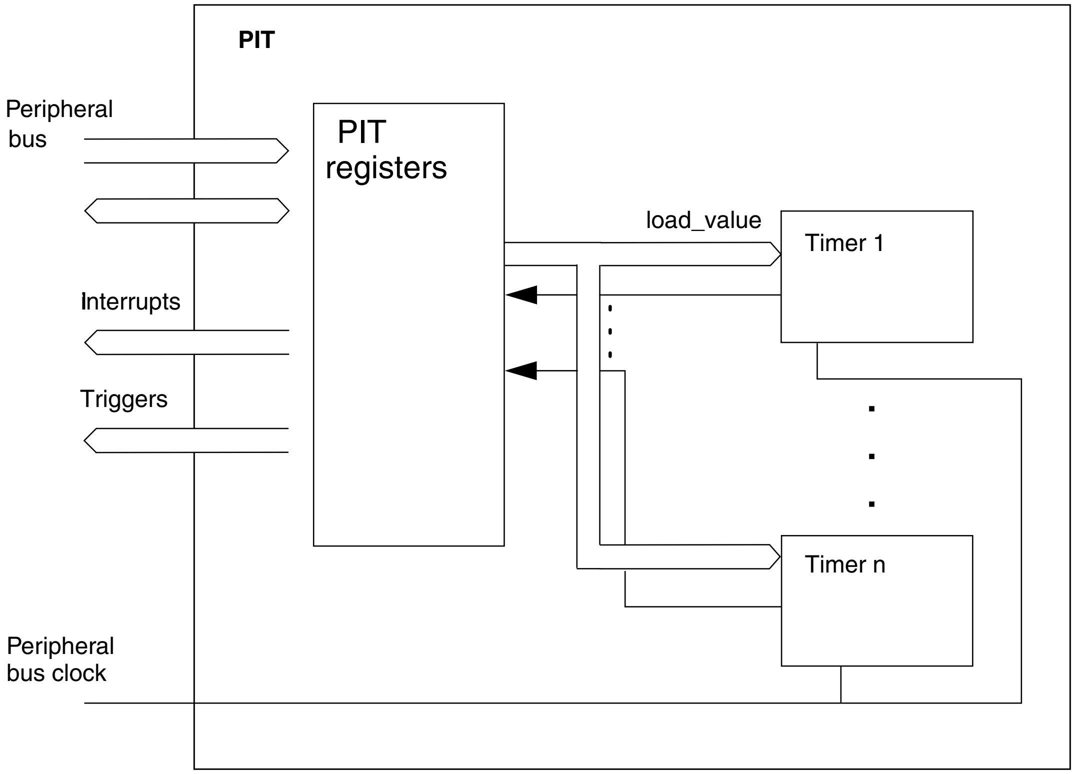

PIT
Peripheral bus
PIT registers
load_value
Timer 1
Interrupts
Triggers
Timer n
Peripheral bus clock

Figure 36-1. Block diagram of the PIT

NOTE

See the chip configuration details for the number of PIT channels used in this MCU.

36.1.2 Features

The main features of this block are:

- Ability of timers to generate DMA trigger pulses
- Ability of timers to generate interrupts
- Maskable interrupts
- Independent timeout periods for each timer

36.2 Signal description

The PIT module has no external pins.

36.3 Memory map/register description

This section provides a detailed description of all registers accessible in the PIT module.

NOTE

- Reserved registers will read as 0, writes will have no effect.
- See the chip configuration details for the number of PIT channels used in this MCU.

PIT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_7000	PIT Module Control Register (PIT_MCR)	32	R/W	0000_0002h	36.3.1/ 822
4003_7100	Timer Load Value Register (PIT_LDVAL0)	32	R/W	0000_0000h	36.3.2/ 823
4003_7104	Current Timer Value Register (PIT_CVAL0)	32	R/W	0000_0000h	36.3.3/ 823
4003_7108	Timer Control Register (PIT_TCTRL0)	32	R/W	0000_0000h	36.3.4/ 824
4003_710C	Timer Flag Register (PIT_TFLG0)	32	R/W	0000_0000h	36.3.5/ 824
4003_7110	Timer Load Value Register (PIT_LDVAL1)	32	R/W	0000_0000h	36.3.2/ 823
4003_7114	Current Timer Value Register (PIT_CVAL1)	32	R/W	0000_0000h	36.3.3/ 823
4003_7118	Timer Control Register (PIT_TCTRL1)	32	R/W	0000_0000h	36.3.4/ 824
4003_711C	Timer Flag Register (PIT_TFLG1)	32	R/W	0000_0000h	36.3.5/ 824
4003_7120	Timer Load Value Register (PIT_LDVAL2)	32	R/W	0000_0000h	36.3.2/ 823
4003_7124	Current Timer Value Register (PIT_CVAL2)	32	R/W	0000_0000h	36.3.3/ 823
4003_7128	Timer Control Register (PIT_TCTRL2)	32	R/W	0000_0000h	36.3.4/ 824
4003_712C	Timer Flag Register (PIT_TFLG2)	32	R/W	0000_0000h	36.3.5/ 824
4003_7130	Timer Load Value Register (PIT_LDVAL3)	32	R/W	0000_0000h	36.3.2/ 823
4003_7134	Current Timer Value Register (PIT_CVAL3)	32	R/W	0000_0000h	36.3.3/ 823

Table continues on the next page...

PIT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_7138	Timer Control Register (PIT_TCTRL3)	32	R/W	0000_0000h	36.3.4/ 824
4003_713C	Timer Flag Register (PIT_TFLG3)	32	R/W	0000_0000h	36.3.5/ 824

36.3.1 PIT Module Control Register (PIT_MCR)

This register enables or disables the PIT timer clocks and controls the timers when the PIT enters the Debug mode.

Address: PIT_MCR is 4003_7000h base + 0h offset = 4003_7000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																														MDIS	FRZ
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

PIT_MCR field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero.
1 MDIS	Module Disable Disables the module clock. This field must be enabled before any other setup is done. 0 Clock for PIT timers is enabled. 1 Clock for PIT timers is disabled.
0 FRZ	Freeze Allows the timers to be stopped when the device enters the Debug mode. 0 Timers continue to run in Debug mode. 1 Timers are stopped in Debug mode.

36.3.2 Timer Load Value Register (PIT_LDVAL*n*)

These registers select the timeout period for the timer interrupts.

Addresses: LDVAL0 is 4003_7000h base + 100h offset = 4003_7100h
LDVAL1 is 4003_7000h base + 110h offset = 4003_7110h
LDVAL2 is 4003_7000h base + 120h offset = 4003_7120h
LDVAL3 is 4003_7000h base + 130h offset = 4003_7130h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSV																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_LDVAL*n* field descriptions

Field	Description
31–0 TSV	Timer Start Value Sets the timer start value. The timer will count down until it reaches 0, then it will generate an interrupt and load this register value again. Writing a new value to this register will not restart the timer; instead the value will be loaded after the timer expires. To abort the current cycle and start a timer period with the new value, the timer must be disabled and enabled again.

36.3.3 Current Timer Value Register (PIT_CVAL*n*)

These registers indicate the current timer position.

Addresses: CVAL0 is 4003_7000h base + 104h offset = 4003_7104h
CVAL1 is 4003_7000h base + 114h offset = 4003_7114h
CVAL2 is 4003_7000h base + 124h offset = 4003_7124h
CVAL3 is 4003_7000h base + 134h offset = 4003_7134h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TVL																															
W	▨																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_CVAL*n* field descriptions

Field	Description
31–0 TVL	Current Timer Value Represents the current timer value, if the timer is enabled. **NOTE:** • If the timer is disabled, do not use this field as its value is unreliable. • The timer uses a downcounter. The timer values are frozen in Debug mode if MCR[FRZ] is set.

36.3.4 Timer Control Register (PIT_TCTRL*n*)

These registers contain the control bits for each timer.

Addresses: TCTRL0 is 4003_7000h base + 108h offset = 4003_7108h
TCTRL1 is 4003_7000h base + 118h offset = 4003_7118h
TCTRL2 is 4003_7000h base + 128h offset = 4003_7128h
TCTRL3 is 4003_7000h base + 138h offset = 4003_7138h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																														TIE	TEN
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_TCTRL*n* field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero.
1 TIE	Timer Interrupt Enable When an interrupt is pending, or, TFLGn[TIF] is set, enabling the interrupt will immediately cause an interrupt event. To avoid this, the associated TFLGn[TIF] must be cleared first. 0 Interrupt requests from Timer n are disabled. 1 Interrupt will be requested whenever TIF is set.
0 TEN	Timer Enable Enables or disables the timer. 0 Timer n is disabled. 1 Timer n is enabled.

36.3.5 Timer Flag Register (PIT_TFLG*n*)

These registers hold the PIT interrupt flags.

Addresses: TFLG0 is 4003_7000h base + 10Ch offset = 4003_710Ch
TFLG1 is 4003_7000h base + 11Ch offset = 4003_711Ch
TFLG2 is 4003_7000h base + 12Ch offset = 4003_712Ch
TFLG3 is 4003_7000h base + 13Ch offset = 4003_713Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															TIF
W																																w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PIT_TFLG*n* field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value zero.
0 TIF	Timer Interrupt Flag Sets to 1 at the end of the timer period. Writing 1 to this flag clears it. Writing 0 has no effect. If enabled, or when TCTRLn[TIE] = 1, TIF causes an interrupt request. 0 Timeout has not yet occurred. 1 Timeout has occurred.

36.4 Functional description

This section provides the functional description of the module.

36.4.1 General operation

This section gives detailed information on the internal operation of the module. Each timer can be used to generate trigger pulses and interrupts. Each interrupt is available on a separate interrupt line.

36.4.1.1 Timers

The timers generate triggers at periodic intervals, when enabled. The timers load the start values as specified in their LDVAL registers, count down to 0 and then load the respective start value again. Each time a timer reaches 0, it will generate a trigger pulse and set the interrupt flag.

All interrupts can be enabled or masked by setting TCTRLn[TIE]. A new interrupt can be generated only after the previous one is cleared.

If desired, the current counter value of the timer can be read via the CVAL registers.

The counter period can be restarted, by first disabling, and then enabling the timer with TCTRLn[TEN]. See the following figure.

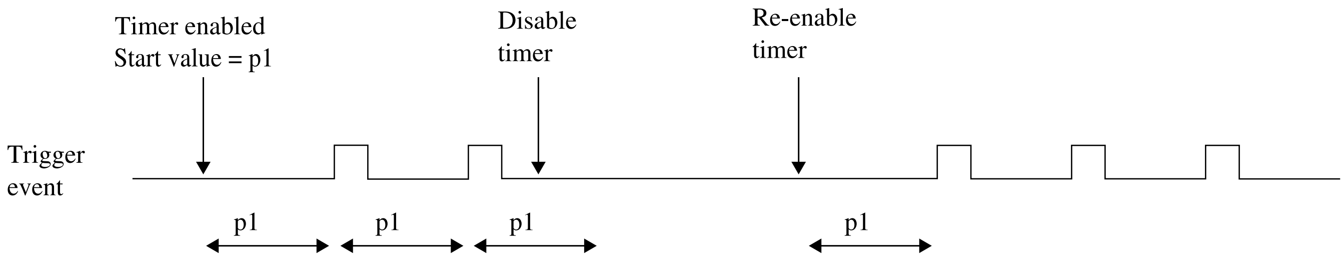

Timer enabled
Start value = p1
Disable
timer
Re-enable
timer
Trigger
event
p1
p1
p1
p1

Figure 36-23. Stopping and starting a timer

The counter period of a running timer can be modified, by first disabling the timer, setting a new load value, and then enabling the timer again. See the following figure.

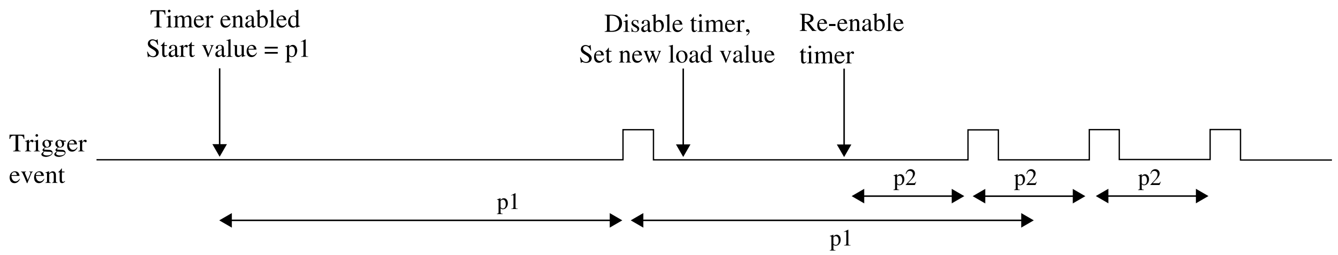

Timer enabled
Start value = p1
Disable timer,
Set new load value
Re-enable
timer
Trigger
event
p2
p2
p2
p1
p1

Figure 36-24. Modifying running timer period

It is also possible to change the counter period without restarting the timer by writing LDVAL with the new load value. This value will then be loaded after the next trigger event. See the following figure.

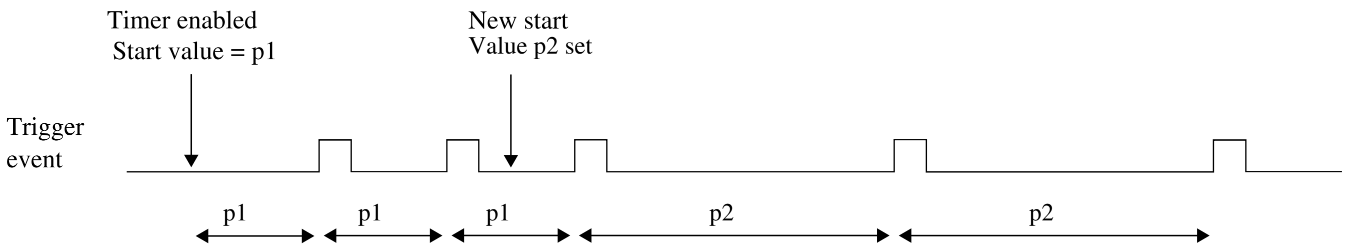

Timer enabled
Start value = p1
New start
Value p2 set
Trigger
event
p1
p1
p1
p2
p2

Figure 36-25. Dynamically setting a new load value

36.4.1.2 Debug mode

In Debug mode, the timers will be frozen based on MCR[FRZ]. This is intended to aid software development, allowing the developer to halt the processor, investigate the current state of the system, for example, the timer values, and then continue the operation.

36.4.2 Interrupts

All the timers support interrupt generation. See the MCU specification for related vector addresses and priorities.

Timer interrupts can be enabled by setting TCTRLn[TIE]. TFLGn[TIF] are set to 1 when a timeout occurs on the associated timer, and are cleared to 0 by writing a 1 to the corresponding TFLGn[TIF].

36.5 Initialization and application information

In the example configuration:

- The PIT clock has a frequency of 50 MHz.
- Timer 1 creates an interrupt every 5.12 ms.
- Timer 3 creates a trigger event every 30 ms.

The PIT module must be activated by writing a 0 to MCR[MDIS].

The 50 MHz clock frequency equates to a clock period of 20 ns. Timer 1 needs to trigger every 5.12 ms/20 ns = 256,000 cycles and Timer 3 every 30 ms/20 ns = 1,500,000 cycles. The value for the LDVAL register trigger is calculated as:

LDVAL trigger = (period / clock period) -1

This means LDVAL1 and LDVAL3 must be written with 0x0003E7FF and 0x0016E35F respectively.

The interrupt for Timer 1 is enabled by setting TCTRL1[TIE]. The timer is started by writing 1 to TCTRL1[TEN].

Timer 3 shall be used only for triggering. Therefore, Timer 3 is started by writing a 1 to TCTRL3[TEN]. TCTRL3[TIE] stays at 0.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;


// Timer 1
PIT_LDVAL1 = 0x0003E7FF; // setup timer 1 for 256000 cycles
PIT_TCTRL1 = TIE; // enable Timer 1 interrupts
PIT_TCTRL1 |= TEN; // start Timer 1


// Timer 3
```

```
PIT_LDVAL3 = 0x0016E35F; // setup timer 3for 1500000 cycles
PIT_TCTRL3 |= TEN; // start Timer 3
```

Chapter 37
Low-Power Timer (LPTMR)

37.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The low-power timer (LPTMR) can be configured to operate as a time counter with optional prescaler, or as a pulse counter with optional glitch filter, across all power modes, including the low-leakage modes. It can also continue operating through most system reset events, allowing it to be used as a time of day counter.

37.1.1 Features

The features of the LPTMR module include:

- 16-bit time counter or pulse counter with compare
  - Optional interrupt can generate asynchronous wakeup from any low-power mode
  - Hardware trigger output
  - Counter supports free-running mode or reset on compare
- Configurable clock source for prescaler/glitch filter
- Configurable input source for pulse counter
  - Rising-edge or falling-edge

37.1.2 Modes of operation

The following table describes the operation of the LPTMR module in various modes.

Table 37-1. Modes of operation

Modes	Description
Run	The LPTMR operates normally.
Wait	The LPTMR continues to operate normally and may be configured to exit the low-power mode by generating an interrupt request.
Stop	The LPTMR continues to operate normally and may be configured to exit the low-power mode by generating an interrupt request.
Low-Leakage	The LPTMR continues to operate normally and may be configured to exit the low-power mode by generating an interrupt request.
Debug	The LPTMR operates normally.

37.2 LPTMR signal descriptions

Table 37-2. LPTMR signal descriptions

Signal	I/O	Description
LPTMR_ALT*n*	I	Pulse Counter Input pin

37.2.1 Detailed signal descriptions

Table 37-3. LPTMR interface—detailed signal descriptions

Signal	I/O	Description	
LPTMR_ALT*n*	I	Pulse Counter Input The LPTMR can select one of the input pins to be used in Pulse Counter mode.	
		State meaning	Assertion—If configured for pulse counter mode with active-high input, then assertion causes the CNR to increment. Deassertion—If configured for pulse counter mode with active-low input, then deassertion causes the CNR to increment.
		Timing	Assertion or deassertion may occur at any time; input may assert asynchronously to the bus clock.

37.3 Memory map and register definition

NOTE

The LPTMR registers are reset only on a POR or LVD event. See LPTMR power and reset for more details.

LPTMR memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_0000	Low Power Timer Control Status Register (LPTMR0_CSR)	32	R/W	0000_0000h	37.3.1/ 831
4004_0004	Low Power Timer Prescale Register (LPTMR0_PSR)	32	R/W	0000_0000h	37.3.2/ 833
4004_0008	Low Power Timer Compare Register (LPTMR0_CMR)	32	R/W	0000_0000h	37.3.3/ 834
4004_000C	Low Power Timer Counter Register (LPTMR0_CNR)	32	R	0000_0000h	37.3.4/ 835

37.3.1 Low Power Timer Control Status Register (LPTMRx_CSR)

Addresses: LPTMR0_CSR is 4004_0000h base + 0h offset = 4004_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								TCF	TIE	TPS		TPP	TFC	TMS	TEN
W									w1c							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_CSR field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 TCF	Timer Compare Flag TCF is set when the LPTMR is enabled and the CNR equals the CMR and increments. TCF is cleared when the LPTMR is disabled or a logic 1 is written to it.

Table continues on the next page...

LPTMRx_CSR field descriptions (continued)

Field	Description
	0 The value of CNR is not equal to CMR and increments. 1 The value of CNR is equal to CMR and increments.
6 TIE	Timer Interrupt Enable When TIE is set, the LPTMR Interrupt is generated whenever TCF is also set. 0 Timer interrupt disabled. 1 Timer interrupt enabled.
5–4 TPS	Timer Pin Select Configures the input source to be used in Pulse Counter mode. TPS must be altered only when the LPTMR is disabled. The input connections vary by device. See the chip configuration details for information on the connections to these inputs. 00 Pulse counter input 0 is selected. 01 Pulse counter input 1 is selected. 10 Pulse counter input 2 is selected. 11 Pulse counter input 3 is selected.
3 TPP	Timer Pin Polarity Configures the polarity of the input source in Pulse Counter mode. TPP must be changed only when the LPTMR is disabled. 0 Pulse Counter input source is active-high, and the CNR will increment on the rising-edge. 1 Pulse Counter input source is active-low, and the CNR will increment on the falling-edge.
2 TFC	Timer Free-Running Counter When clear, TFC configures the CNR to reset whenever TCF is set. When set, TFC configures the CNR to reset on overflow. TFC must be altered only when the LPTMR is disabled. 0 CNR is reset whenever TCF is set. 1 CNR is reset on overflow.
1 TMS	Timer Mode Select Configures the mode of the LPTMR. TMS must be altered only when the LPTMR is disabled. 0 Time Counter mode. 1 Pulse Counter mode.
0 TEN	Timer Enable When TEN is clear, it resets the LPTMR internal logic, including the CNR and TCF. When TEN is set, the LPTMR is enabled. While writing 1 to this field, CSR[5:1] must not be altered. 0 LPTMR is disabled and internal logic is reset. 1 LPTMR is enabled.

37.3.2 Low Power Timer Prescale Register (LPTMRx_PSR)

Addresses: LPTMR0_PSR is 4004_0000h base + 4h offset = 4004_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																									PRESCALE				PBYP	PCS	
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_PSR field descriptions

Field	Description
31–7 Reserved	This read-only field is reserved and always has the value zero.
6–3 PRESCALE	Prescale Value Configures the size of the Prescaler in Time Counter mode or width of the glitch filter in Pulse Counter mode. PRESCALE must be altered only when the LPTMR is disabled. 0000 Prescaler divides the prescaler clock by 2; glitch filter does not support this configuration. 0001 Prescaler divides the prescaler clock by 4; glitch filter recognizes change on input pin after 2 rising clock edges. 0010 Prescaler divides the prescaler clock by 8; glitch filter recognizes change on input pin after 4 rising clock edges. 0011 Prescaler divides the prescaler clock by 16; glitch filter recognizes change on input pin after 8 rising clock edges. 0100 Prescaler divides the prescaler clock by 32; glitch filter recognizes change on input pin after 16 rising clock edges. 0101 Prescaler divides the prescaler clock by 64; glitch filter recognizes change on input pin after 32 rising clock edges. 0110 Prescaler divides the prescaler clock by 128; glitch filter recognizes change on input pin after 64 rising clock edges. 0111 Prescaler divides the prescaler clock by 256; glitch filter recognizes change on input pin after 128 rising clock edges. 1000 Prescaler divides the prescaler clock by 512; glitch filter recognizes change on input pin after 256 rising clock edges. 1001 Prescaler divides the prescaler clock by 1024; glitch filter recognizes change on input pin after 512 rising clock edges. 1010 Prescaler divides the prescaler clock by 2048; glitch filter recognizes change on input pin after 1024 rising clock edges. 1011 Prescaler divides the prescaler clock by 4096; glitch filter recognizes change on input pin after 2048 rising clock edges. 1100 Prescaler divides the prescaler clock by 8192; glitch filter recognizes change on input pin after 4096 rising clock edges. 1101 Prescaler divides the prescaler clock by 16,384; glitch filter recognizes change on input pin after 8192 rising clock edges. 1110 Prescaler divides the prescaler clock by 32,768; glitch filter recognizes change on input pin after 16,384 rising clock edges. 1111 Prescaler divides the prescaler clock by 65,536; glitch filter recognizes change on input pin after 32,768 rising clock edges.

Table continues on the next page...

LPTMRx_PSR field descriptions (continued)

Field	Description
2 PBYP	Prescaler Bypass When PBYP is set, the selected prescaler clock in Time Counter mode or selected input source in Pulse Counter mode directly clocks the CNR. When PBYP is clear, the CNR is clocked by the output of the prescaler/glitch filter. PBYP must be altered only when the LPTMR is disabled. 0 Prescaler/glitch filter is enabled. 1 Prescaler/glitch filter is bypassed.
1–0 PCS	Prescaler Clock Select Selects the clock to be used by the LPTMR prescaler/glitch filter. PCS must be altered only when the LPTMR is disabled. The clock connections vary by device. **NOTE:** See the chip configuration details for information on the connections to these inputs. 00 Prescaler/glitch filter clock 0 selected. 01 Prescaler/glitch filter clock 1 selected. 10 Prescaler/glitch filter clock 2 selected. 11 Prescaler/glitch filter clock 3 selected.

37.3.3 Low Power Timer Compare Register (LPTMRx_CMR)

Addresses: LPTMR0_CMR is 4004_0000h base + 8h offset = 4004_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COMPARE															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_CMR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 COMPARE	Compare Value When the LPTMR is enabled and the CNR equals the value in the CMR and increments, TCF is set and the hardware trigger asserts until the next time the CNR increments. If the CMR is 0, the hardware trigger will remain asserted until the LPTMR is disabled. If the LPTMR is enabled, the CMR must be altered only when TCF is set.

37.3.4 Low Power Timer Counter Register (LPTMRx_CNR)

Addresses: LPTMR0_CNR is 4004_0000h base + Ch offset = 4004_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNTER															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LPTMRx_CNR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 COUNTER	Counter Value

37.4 Functional description

37.4.1 LPTMR power and reset

The LPTMR remains powered in all power modes, including low-leakage modes. If the LPTMR is not required to remain operating during a low-power mode, then it must be disabled before entering the mode.

The LPTMR is reset only on global Power On Reset (POR) or Low Voltage Detect (LVD). When configuring the LPTMR registers, the CSR must be initially written with the timer disabled, before configuring the PSR and CMR. Then, CSR[TIE] must be set as the last step in the initialization. This ensures the LPTMR is configured correctly and the LPTMR counter is reset to zero following a warm reset.

37.4.2 LPTMR clocking

The LPTMR prescaler/glitch filter can be clocked by one of the four clocks. The clock source must be enabled before the LPTMR is enabled.

NOTE

The clock source selected may need to be configured to remain enabled in low-power modes, otherwise the LPTMR will not operate during low-power modes.

In Pulse Counter mode with the prescaler/glitch filter bypassed, the selected input source directly clocks the CNR and no other clock source is required. To minimize power in this case, configure the prescaler clock source for a clock that is not toggling.

NOTE

The clock source or pulse input source selected for the LPTMR should not exceed the frequency f_{LPTMR} defined in the device datasheet.

37.4.3 LPTMR prescaler/glitch filter

The LPTMR prescaler and glitch filter share the same logic which operates as a prescaler in Time Counter mode and as a glitch filter in Pulse Counter mode.

NOTE

The prescaler/glitch filter configuration must not be altered when the LPTMR is enabled.

37.4.3.1 Prescaler enabled

In Time Counter mode, when the prescaler is enabled, the output of the prescaler directly clocks the CNR. When the LPTMR is enabled, the CNR will increment every 2^2 to 2^{16} prescaler clock cycles. After the LPTMR is enabled, the first increment of the CNR will take an additional one or two prescaler clock cycles due to synchronization logic.

37.4.3.2 Prescaler bypassed

In Time Counter mode, when the prescaler is bypassed, the selected prescaler clock increments the CNR on every clock cycle. When the LPTMR is enabled, the first increment will take an additional one or two prescaler clock cycles due to synchronization logic.

37.4.3.3 Glitch filter

In Pulse Counter mode, when the glitch filter is enabled, the output of the glitch filter directly clocks the CNR. When the LPTMR is first enabled, the output of the glitch filter is asserted, that is, logic 1 for active-high and logic 0 for active-low. The following table shows the change in glitch filter output with the selected input source.

If	Then
The selected input source remains deasserted for at least 2^1 to 2^{15} consecutive prescaler clock rising edges	The glitch filter output will also deassert.
The selected input source remains asserted for at least 2^1 to 2^{15} consecutive prescaler clock rising-edges	The glitch filter output will also assert.

NOTE

The input is only sampled on the rising clock edge.

The CNR will increment each time the glitch filter output asserts. In Pulse Counter mode, the maximum rate at which the CNR can increment is once every 2^2 to 2^{16} prescaler clock edges. When first enabled, the glitch filter will wait an additional one or two prescaler clock edges due to synchronization logic.

37.4.3.4 Glitch filter bypassed

In Pulse Counter mode, when the glitch filter is bypassed, the selected input source increments the CNR every time it asserts. Before the LPTMR is first enabled, the selected input source is forced to be asserted. This prevents the CNR from incrementing if the selected input source is already asserted when the LPTMR is first enabled.

37.4.4 LPTMR compare

When the CNR equals the value of the CMR and increments, the following events occur:

- CSR[TCF] is set.
- LPTMR interrupt is generated if CSR[TIE] is also set.
- LPTMR hardware trigger is generated.
- CNR is reset if CSR[TFC] is clear.

When the LPTMR is enabled, the CMR can be altered only when CSR[TCF] is set. When updating the CMR, the CMR must be written and CSR[TCF] must be cleared before the LPTMR counter has incremented past the new LPTMR compare value.

37.4.5 LPTMR counter

The CNR increments by one on every:

- Prescaler clock in Time Counter mode with prescaler bypassed
- Prescaler output in Time Counter mode with prescaler enabled
- Input source assertion in Pulse Counter mode with glitch filter bypassed
- Glitch filter output in Pulse Counter mode with glitch filter enabled

The CNR is reset when the LPTMR is disabled or if the counter register overflows. If CSR[TFC] is set, then the CNR is also reset whenever CSR[TCF] is set.

The CNR continues incrementing when the core is halted in Debug mode.

The CNR cannot be initialized, but can be read at any time. On each read of the CNR, software must first write to the CNR with any value. This will synchronize and register the current value of the CNR into a temporary register. The contents of the temporary register are returned on each read of the CNR.

When reading the CNR, the bus clock must be at least two times faster than the rate at which the LPTMR counter is incrementing, otherwise incorrect data may be returned.

37.4.6 LPTMR hardware trigger

The LPTMR hardware trigger asserts at the same time the CSR[TCF] is set and can be used to trigger hardware events in other peripherals without software intervention. The hardware trigger is always enabled.

When	Then
The CMR is set to 0 with CSR[TFC] clear	The LPTMR hardware trigger will assert on the first compare and does not deassert.
The CMR is set to a nonzero value, or, if CSR[TFC] is set	The LPTMR hardware trigger will assert on each compare and deassert on the following increment of the CNR.

37.4.7 LPTMR interrupt

The LPTMR interrupt is generated whenever CSR[TIE] and CSR[TCF] are set. CSR[TCF] is cleared by disabling the LPTMR or by writing a logic 1 to it.

CSR[TIE] can be altered and CSR[TCF] can be cleared while the LPTMR is enabled.

The LPTMR interrupt is generated asynchronously to the system clock and can be used to generate a wakeup from any low-power mode, including the low-leakage modes, provided the LPTMR is enabled as a wakeup source.

Chapter 38
Carrier Modulator Transmitter (CMT)

38.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The carrier modulator transmitter (CMT) module provides the means to generate the protocol timing and carrier signals for a wide variety of encoding schemes. The CMT incorporates hardware to off-load the critical and/or lengthy timing requirements associated with signal generation from the CPU, releasing much of its bandwidth to handle other tasks such as:

- Code data generation
- Data decompression, or,
- Keyboard scanning

. The CMT does not include dedicated hardware configurations for specific protocols, but is intended to be sufficiently programmable in its function to handle the timing requirements of most protocols with minimal CPU intervention.

When the modulator is disabled, certain CMT registers can be used to change the state of the infrared output (IRO) signal directly. This feature allows for the generation of future protocol timing signals not readily producible by the current architecture.

38.2 Features

The features of this module include:

- Four modes of operation:
    - Time; with independent control of high and low times

  - Baseband
  - Frequency-shift key (FSK)
  - Direct software control of the IRO signal
- Extended space operation in Time, Baseband, and FSK modes
- Selectable input clock divider
- Interrupt on end-of-cycle
  - Ability to disable the IRO signal and use as timer interrupt

38.3 Block diagram

The following figure presents the block diagram of the CMT module.

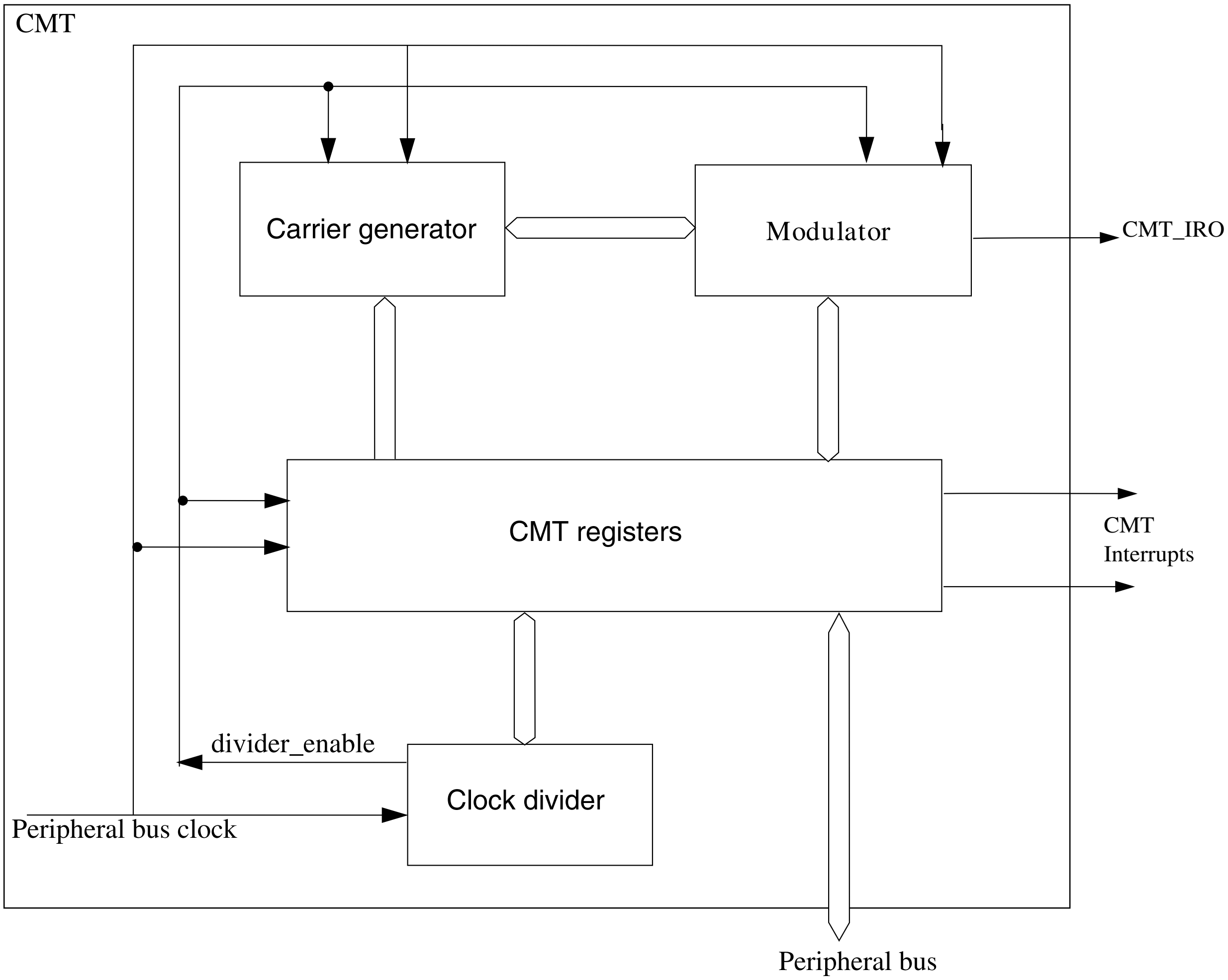

CMT
Carrier generator
Modulator
CMT_IRO
CMT registers
CMT
Interrupts
divider_enable
Clock divider
Peripheral bus clock
Peripheral bus

Figure 38-1. CMT module block diagram

38.4 Modes of operation

The following table describes the operation of the CMT module operates in various modes.

Table 38-1. Modes of operation

Modes	Description
Time	In Time mode, the user independently defines the high and low times of the carrier signal to determine both period and duty cycle
Baseband	When MSC[BASE] is set, the carrier output (f_{cg}) to the modulator is held high continuously to allow for the generation of baseband protocols.

Table continues on the next page...

Table 38-1. Modes of operation (continued)

Modes	Description
Frequency-shift key	This mode allows the carrier generator to alternate between two sets of high and low times. When operating in FSK mode, the generator will toggle between the two sets when instructed by the modulator, allowing the user to dynamically switch between two carrier frequencies without CPU intervention.

The following table summarizes the modes of operation of the CMT module.

Table 38-2. CMT modes of operation

Mode	MSC[MCGEN][1]	MSC[BASE][2]	MSC[FSK][2]	MSC[EXSPC]	Comment
Time	1	0	0	0	f_{cg} controlled by primary high and low registers. f_{cg} transmitted to the IRO signal when modulator gate is open.
Baseband	1	1	X	0	f_{cg} is always high. The IRO signal is high when the modulator gate is open.
FSK	1	0	1	0	f_{cg} control alternates between primary high/low registers and secondary high/low registers. f_{cg} transmitted to the IRO signal when modulator gate is open.
Extended Space	1	X	X	1	Setting MSC[EXSPC] causes subsequent modulator cycles to be spaces (modulator out not asserted) for the duration of the modulator period (mark and space times).
IRO Latch	0	X	X	X	OC[IROL] controls the state of the IRO signal.

1. To prevent spurious operation, initialize all data and control registers before beginning a transmission when MSC[MCGEN]=1.
2. This field is not double-buffered and must not be changed during a transmission while MSC[MCGEN]=1.

NOTE

The assignment of module modes to core modes is chip-specific. For module-to-core mode assignments, see the chapter that describes how modules are configured.

38.4.1 Wait mode operation

During Wait mode, the CMT if enabled, will continue to operate normally. However, there is no change in operating modes of CMT during Wait mode, because the CPU is not operating.

38.4.2 Stop mode operation

This section describes the CMT Stop mode operations.

38.4.2.1 Normal Stop mode operation

During Normal Stop mode, clocks to the CMT module are halted. No registers are affected.

The CMT module will resume upon exit from Normal Stop mode because the clocks are halted. Software must ensure that the Normal Stop mode is not entered while the modulator is still in operation so as to prevent the IRO signal from being asserted while in Normal Stop mode. This may require a timeout period from the time that MSC[MCGEN] is cleared to allow the last modulator cycle to complete.

38.4.2.2 Low-Power Stop mode operation

During Low-Power Stop mode, the CMT module is completely powered off internally and the IRO signal state is latched and held at the time when the CMT enters this mode. To prevent the IRO signal from being asserted during Low-Power Stop mode, the software must assure that the signal is not active when entering Low-Power Stop mode. Upon wakeup from Low-Power Stop mode, the CMT module will be in the reset state.

38.5 CMT external signal descriptions

The following table shows the description of the external signal.

Table 38-3. CMT signal description

Signal	Description	I/O
CMT_IRO	Infrared Output	O

38.5.1 CMT_IRO — Infrared Output

This output signal is driven by the modulator output when MSC[MCGEN] and OC[IROPEN] are set. The IRO signal starts a valid transmission with a delay, after MSC[MCGEN] bit be asserted to high, that can be calculated based on two register bits. Table 38-5 shows how to calculate this delay.

The following table describes conditions for the IRO signal to be active.

If	Then
MSC[MCGEN] is cleared and OC[IROPEN] is set	The signal is driven by OC[IROL] . This enables user software to directly control the state of the IRO signal by writing to OC[IROL] .
OC[IROPEN] is cleared	The signal is disabled and is not driven by the CMT module. Therefore, CMT can be configured as a modulo timer for generating periodic interrupts without causing signal activity.

Table 38-5. CMT_IRO signal delay calculation

Condition	Delay (bus clock cycles)
MSC[CMTDIV] = 0	PPS[PPSDIV] + 2
MSC[CMTDIV] > 0	(PPS[PPSDIV] *2) + 3

38.6 Memory map/register definition

The following registers control and monitor the CMT operation.

The address of a register is the sum of a base address and an address offset. The base address is defined at the chip level. The address offset is defined at the module level.

CMT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_2000	CMT Carrier Generator High Data Register 1 (CMT_CGH1)	8	R/W	Undefined	38.6.1/ 845
4006_2001	CMT Carrier Generator Low Data Register 1 (CMT_CGL1)	8	R/W	Undefined	38.6.2/ 846
4006_2002	CMT Carrier Generator High Data Register 2 (CMT_CGH2)	8	R/W	Undefined	38.6.3/ 846

Table continues on the next page...

CMT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_2003	CMT Carrier Generator Low Data Register 2 (CMT_CGL2)	8	R/W	Undefined	38.6.4/ 847
4006_2004	CMT Output Control Register (CMT_OC)	8	R/W	00h	38.6.5/ 848
4006_2005	CMT Modulator Status and Control Register (CMT_MSC)	8	R/W	00h	38.6.6/ 849
4006_2006	CMT Modulator Data Register Mark High (CMT_CMD1)	8	R/W	Undefined	38.6.7/ 851
4006_2007	CMT Modulator Data Register Mark Low (CMT_CMD2)	8	R/W	Undefined	38.6.8/ 851
4006_2008	CMT Modulator Data Register Space High (CMT_CMD3)	8	R/W	Undefined	38.6.9/ 852
4006_2009	CMT Modulator Data Register Space Low (CMT_CMD4)	8	R/W	Undefined	38.6.10/ 852
4006_200A	CMT Primary Prescaler Register (CMT_PPS)	8	R/W	00h	38.6.11/ 853
4006_200B	CMT Direct Memory Access Register (CMT_DMA)	8	R/W	00h	38.6.12/ 854

38.6.1 CMT Carrier Generator High Data Register 1 (CMT_CGH1)

This data register contains the primary high value for generating the carrier output.

Address: CMT_CGH1 is 4006_2000h base + 0h offset = 4006_2000h

Bit	7	6	5	4	3	2	1	0
Read Write	PH							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CGH1 field descriptions

Field	Description
7–0 PH	Primary Carrier High Time Data Value Contains the number of input clocks required to generate the carrier high time period. When operating in Time mode, this register is always selected. When operating in FSK mode, this register and the secondary register pair are alternately selected under the control of the modulator. The primary carrier high time value is undefined out of reset. This register must

CMT_CGH1 field descriptions (continued)

Field	Description
	be written to nonzero values before the carrier generator is enabled to avoid spurious results.

38.6.2 CMT Carrier Generator Low Data Register 1 (CMT_CGL1)

This data register contains the primary low value for generating the carrier output.

Address: CMT_CGL1 is 4006_2000h base + 1h offset = 4006_2001h

Bit	7	6	5	4	3	2	1	0
Read Write	PL							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CGL1 field descriptions

Field	Description
7–0 PL	Primary Carrier Low Time Data Value Contains the number of input clocks required to generate the carrier low time period. When operating in Time mode, this register is always selected. When operating in FSK mode, this register and the secondary register pair are alternately selected under the control of the modulator. The primary carrier low time value is undefined out of reset. This register must be written to nonzero values before the carrier generator is enabled to avoid spurious results.

38.6.3 CMT Carrier Generator High Data Register 2 (CMT_CGH2)

This data register contains the secondary high value for generating the carrier output.

Address: CMT_CGH2 is 4006_2000h base + 2h offset = 4006_2002h

Bit	7	6	5	4	3	2	1	0
Read Write	SH							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CGH2 field descriptions

Field	Description
7–0 SH	Secondary Carrier High Time Data Value Contains the number of input clocks required to generate the carrier high time period. When operating in Time mode, this register is never selected. When operating in FSK mode, this register and the primary register pair are alternately selected under control of the modulator. The secondary carrier high time value is undefined out of reset. This register must be written to nonzero values before the carrier generator is enabled when operating in FSK mode.

38.6.4 CMT Carrier Generator Low Data Register 2 (CMT_CGL2)

This data register contains the secondary low value for generating the carrier output.

Address: CMT_CGL2 is 4006_2000h base + 3h offset = 4006_2003h

Bit	7	6	5	4	3	2	1	0
Read Write	SL							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CGL2 field descriptions

Field	Description
7–0 SL	Secondary Carrier Low Time Data Value Contains the number of input clocks required to generate the carrier low time period. When operating in Time mode, this register is never selected. When operating in FSK mode, this register and the primary register pair are alternately selected under the control of the modulator. The secondary carrier low time value is undefined out of reset. This register must be written to nonzero values before the carrier generator is enabled when operating in FSK mode.

38.6.5 CMT Output Control Register (CMT_OC)

This register is used to control the IRO signal of the CMT module.

Address: CMT_OC is 4006_2000h base + 4h offset = 4006_2004h

Bit	7	6	5	4
Read	IROL	CMTPOL	IROPEN	0
Write				
Reset	0	0	0	0

Bit	3	2	1	0
Read	0			
Write				
Reset	0	0	0	0

CMT_OC field descriptions

Field	Description
7 IROL	IRO Latch Control Reads the state of the IRO latch. Writing to IROL changes the state of the IRO signal when MSC[MCGEN] is cleared and IROPEN is set.
6 CMTPOL	CMT Output Polarity Controls the polarity of the IRO signal. 0 The IRO signal is active-low. 1 The IRO signal is active-high.
5 IROPEN	IRO Pin Enable Enables and disables the IRO signal. When the IRO signal is enabled, it is an output that drives out either the CMT transmitter output or the state of IROL depending on whether MSC[MCGEN] is set or not. Also, the state of output is either inverted or non-inverted, depending on the state of CMTPOL. When the IRO signal is disabled, it is in a high-impedance state and is unable to draw any current. This signal is disabled during reset. 0 The IRO signal is disabled. 1 The IRO signal is enabled as output.
4–0 Reserved	This read-only field is reserved and always has the value zero.

38.6.6 CMT Modulator Status and Control Register (CMT_MSC)

This register contains the modulator and carrier generator enable (MCGEN), end of cycle interrupt enable (EOCIE), FSK mode select (FSK), baseband enable (BASE), extended space (EXSPC), prescaler (CMTDIV) bits, and the end of cycle (EOCF) status bit.

Address: CMT_MSC is 4006_2000h base + 5h offset = 4006_2005h

Bit	7	6	5	4
Read	EOCF	CMTDIV		EXSPC
Write				
Reset	0	0	0	0

Bit	3	2	1	0
Read Write	BASE	FSK	EOCIE	MCGEN
Reset	0	0	0	0

CMT_MSC field descriptions

Field	Description
7 EOCF	End Of Cycle Status Flag Sets when: • The modulator is not currently active and MCGEN is set to begin the initial CMT transmission. • At the end of each modulation cycle while MCGEN is set. This is recognized when a match occurs between the contents of the space period register and the down counter. At this time, the counter is initialized with, possibly new contents of the mark period buffer, CMD1 and CMD2, and the space period register is loaded with, possibly new contents of the space period buffer, CMD3 and CMD4. This flag is cleared by reading MSC followed by an access of CMD2 or CMD4, or by the DMA transfer. 0 End of modulation cycle has not occured since the flag last cleared. 1 End of modulator cycle has occurred.
6–5 CMTDIV	CMT Clock Divide Prescaler Causes the CMT to be clocked at the IF signal frequency, or the IF frequency divided by 2 ,4, or 8 . This field must not be changed during a transmission because it is not double-buffered. 00 IF ÷ 1 01 IF ÷ 2 10 IF ÷ 4 11 IF ÷ 8

Table continues on the next page...

CMT_MSC field descriptions (continued)

Field	Description
4 EXSPC	Extended Space Enable Enables the extended space operation. 0 Extended space is disabled. 1 Extended space is enabled.
3 BASE	Baseband Enable When set, BASE disables the carrier generator and forces the carrier output high for generation of baseband protocols. When BASE is cleared, the carrier generator is enabled and the carrier output toggles at the frequency determined by values stored in the carrier data registers. This field is cleared by reset. This field is not double-buffered and must not be written to during a transmission. 0 Baseband mode is disabled. 1 Baseband mode is enabled.
2 FSK	FSK Mode Select Enables FSK operation. 0 The CMT operates in Time or Baseband mode. 1 The CMT operates in FSK mode.
1 EOCIE	End of Cycle Interrupt Enable Requests to enable a CPU interrupt when EOCF is set if EOCIE is high. 0 CPU interrupt is disabled. 1 CPU interrupt is enabled.
0 MCGEN	Modulator and Carrier Generator Enable Setting MCGEN will initialize the carrier generator and modulator and will enable all clocks. When enabled, the carrier generator and modulator will function continuously. When MCGEN is cleared, the current modulator cycle will be allowed to expire before all carrier and modulator clocks are disabled to save power and the modulator output is forced low. **NOTE:** To prevent spurious operation, the user should initialize all data and control registers before enabling the system. 0 Modulator and carrier generator disabled 1 Modulator and carrier generator enabled

38.6.7 CMT Modulator Data Register Mark High (CMT_CMD1)

The contents of this register are transferred to the modulator down counter upon the completion of a modulation period.

Address: CMT_CMD1 is 4006_2000h base + 6h offset = 4006_2006h

Bit	7	6	5	4	3	2	1	0
Read Write	MB[15:8]							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CMD1 field descriptions

Field	Description
7–0 MB[15:8]	Controls the upper mark periods of the modulator for all modes.

38.6.8 CMT Modulator Data Register Mark Low (CMT_CMD2)

The contents of this register are transferred to the modulator down counter upon the completion of a modulation period.

Address: CMT_CMD2 is 4006_2000h base + 7h offset = 4006_2007h

Bit	7	6	5	4	3	2	1	0
Read Write	MB[7:0]							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CMD2 field descriptions

Field	Description
7–0 MB[7:0]	Controls the lower mark periods of the modulator for all modes.

38.6.9 CMT Modulator Data Register Space High (CMT_CMD3)

The contents of this register are transferred to the space period register upon the completion of a modulation period.

Address: CMT_CMD3 is 4006_2000h base + 8h offset = 4006_2008h

Bit	7	6	5	4	3	2	1	0
Read Write	SB[15:8]							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CMD3 field descriptions

Field	Description
7–0 SB[15:8]	Controls the upper space periods of the modulator for all modes.

38.6.10 CMT Modulator Data Register Space Low (CMT_CMD4)

The contents of this register are transferred to the space period register upon the completion of a modulation period.

Address: CMT_CMD4 is 4006_2000h base + 9h offset = 4006_2009h

Bit	7	6	5	4	3	2	1	0
Read Write	SB[7:0]							
Reset	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

CMT_CMD4 field descriptions

Field	Description
7–0 SB[7:0]	Controls the lower space periods of the modulator for all modes.

38.6.11 CMT Primary Prescaler Register (CMT_PPS)

This register is used to set the Primary Prescaler Divider field (PPSDIV).

Address: CMT_PPS is 4006_2000h base + Ah offset = 4006_200Ah

Bit	7	6	5	4	3	2	1	0
Read	0				PPSDIV			
Write	▨	▨	▨	▨				
Reset	0	0	0	0	0	0	0	0

CMT_PPS field descriptions

Field	Description
7–4 Reserved	This read-only field is reserved and always has the value zero.
3–0 PPSDIV	Primary Prescaler Divider Divides the CMT clock to generate the Intermediate Frequency clock enable to the secondary prescaler. 0000 Bus clock ÷ 1 0001 Bus clock ÷ 2 0010 Bus clock ÷ 3 0011 Bus clock ÷ 4 0100 Bus clock ÷ 5 0101 Bus clock ÷ 6 0110 Bus clock ÷ 7 0111 Bus clock ÷ 8 1000 Bus clock ÷ 9 1001 Bus clock ÷ 10 1010 Bus clock ÷ 11 1011 Bus clock ÷ 12 1100 Bus clock ÷ 13 1101 Bus clock ÷ 14 1110 Bus clock ÷ 15 1111 Bus clock ÷ 16

38.6.12 CMT Direct Memory Access Register (CMT_DMA)

This register is used to enable/disable direct memory access (DMA).

Address: CMT_DMA is 4006_2000h base + Bh offset = 4006_200Bh

Bit	7	6	5	4	3	2	1	0
Read	0							DMA
Write	▨							
Reset	0	0	0	0	0	0	0	0

CMT_DMA field descriptions

Field	Description
7–1 Reserved	This read-only field is reserved and always has the value zero.
0 DMA	DMA Enable Enables the DMA protocol. 0 DMA transfer request and done are disabled. 1 DMA transfer request and done are enabled.

38.7 Functional description

The CMT module primarily consists of clock divider, carrier generator, and modulator.

38.7.1 Clock divider

The CMT was originally designed to be based on an 8 MHz bus clock that could be divided by 1, 2, 4, or 8 according to the specification. To be compatible with higher bus frequency, the primary prescaler (PPS) was developed to receive a higher frequency and generate a clock enable signal called intermediate frequency (IF). This IF must be approximately equal to 8 MHz and will work as a clock enable to the secondary prescaler. The following figure shows the clock divider block diagram.

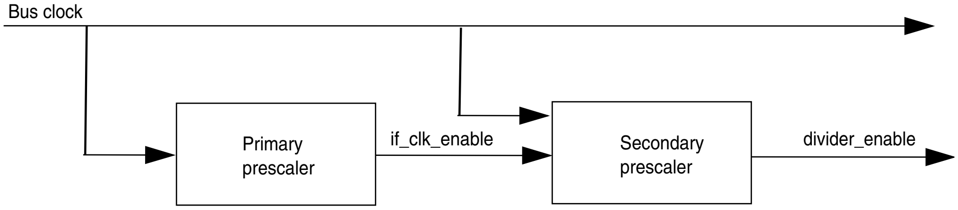


Figure 38-14. Clock divider block diagram

For compatibility with previous versions of CMT, when bus clock = 8 MHz, the PPS must be configured to zero. The PPS counter is selected according to the bus clock to generate an intermediate frequency approximately equal to 8 MHz.

38.7.2 Carrier generator

The carrier generator resolution is 125 ns when operating with an 8 MHz intermediate frequency signal and the secondary prescaler is set to divide by 1, or, when MSC[CMTDIV] = 00. The carrier generator can generate signals with periods between 250 ns (4 MHz) and 127.5 µs (7.84 kHz) in steps of 125 ns. The following table shows the relationship between the clock divide bits and the carrier generator resolution, minimum carrier generator period, and minimum modulator period.

Table 38-19. Clock divider

Bus clock (MHz)	MSC[CMTDIV]	Carrier generator resolution (µs)	Min. carrier generator period (µs)	Min. modulator period (µs)
8	00	0.125	0.25	1.0
8	01	0.25	0.5	2.0
8	10	0.5	1.0	4.0
8	11	1.0	2.0	8.0

The possible duty cycle options depend upon the number of counts required to complete the carrier period. For example, 1.6 MHz signal has a period of 625 ns and will therefore require 5 x 125 ns counts to generate. These counts may be split between high and low times, so the duty cycles available will be:

- 20% with one high and four low times
- 40% with two high and three low times
- 60% with three high and two low times, and
- 80% with four high and one low time

.

For low-frequency signals with large periods, high-resolution duty cycles as a percentage of the total period, are possible.

The carrier signal is generated by counting a register-selected number of input clocks (125 ns for an 8 MHz bus) for both the carrier high time and the carrier low time. The period is determined by the total number of clocks counted. The duty cycle is determined by the ratio of high-time clocks to total clocks counted. The high and low time values are user-programmable and are held in two registers.

An alternate set of high/low count values is held in another set of registers to allow the generation of dual-frequency FSK protocols without CPU intervention.

Note

> Only nonzero data values are allowed. The carrier generator will not work if any of the count values are equal to zero.

MSC[MCGEN] must be set and MSC[BASE] must be cleared to enable carrier generator clocks. When MSC[BASE] is set, the carrier output to the modulator is held high continuously. The following figure represents the block diagram of the clock generator.

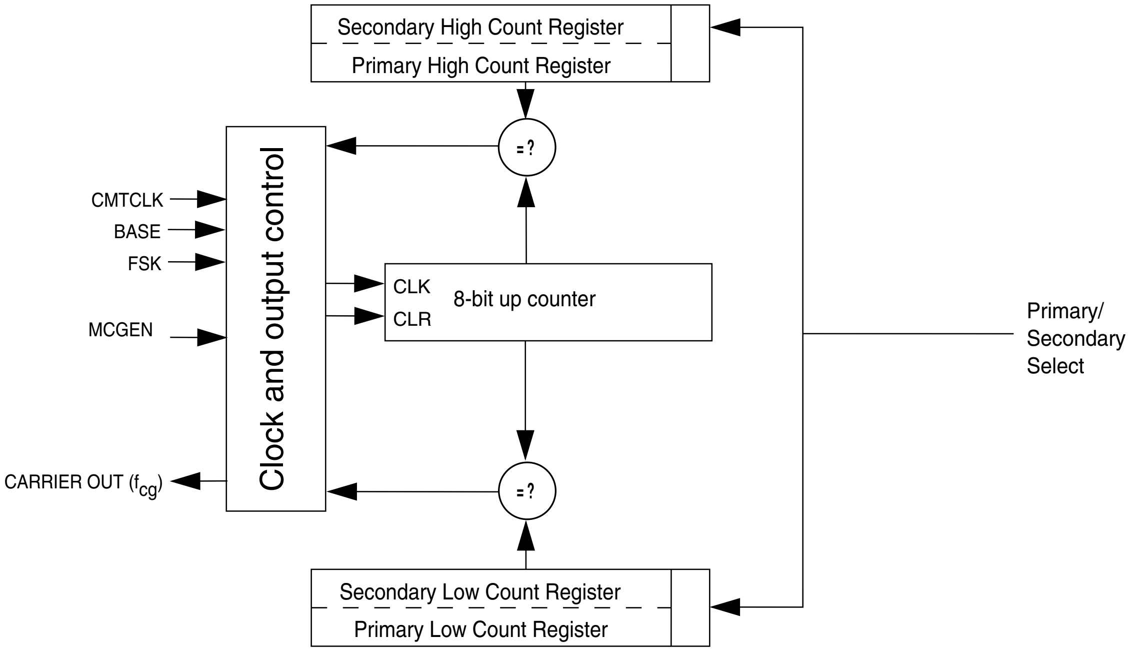

Secondary High Count Register
Primary High Count Register
=?
CMTCLK
BASE
FSK
MCGEN
Clock and output control
CLK
CLR
8-bit up counter
Primary/
Secondary
Select
CARRIER OUT (f_{cg})
=?
Secondary Low Count Register
Primary Low Count Register

Figure 38-15. Carrier generator block diagram

The high/low time counter is an 8-bit up counter. After each increment, the contents of the counter are compared with the appropriate high or low count value register. When the compare value is reached, the counter is reset to a value of 0x01, and the compare is redirected to the other count value register.

Assuming that the high time count compare register is currently active, a valid compare will cause the carrier output to be driven low. The counter will continue to increment starting at the reset value of 0x01. When the value stored in the selected low count value register is reached, the counter will again be reset and the carrier output will be driven high.

The cycle repeats, automatically generating a periodic signal which is directed to the modulator. The lower frequency with maximum period, f_{max}, and highest frequency with minimum period, f_{min}, which can be generated, are defined as:

$f_{max} = f_{CMTCLK} \div (2 * 1)$ Hz

$f_{min} = f_{CMTCLK} \div (2 * (2^8 - 1))$ Hz

In the general case, the carrier generator output frequency is:

$f_{cg} = f_{CMTCLK} \div (\text{High count} + \text{Low count})$ Hz

Where: $0 < \text{High count} < 256$ and

$0 < \text{Low count} < 256$

The duty cycle of the carrier signal is controlled by varying the ratio of high time to low + high time. As the input clock period is fixed, the duty cycle resolution will be proportional to the number of counts required to generate the desired carrier period.

$$\text{DutyCycle} = \frac{\text{Highcount}}{\text{Highcount}+\text{Lowcount}}$$

38.7.3 Modulator

The modulator block controls the state of the infrared out signal (IRO). The modulator output is gated on to the IRO signal when the modulator/carrier generator is enabled. . When the modulator/carrier generator is disabled, the IRO signal is controlled by the state of the IRO latch. OC[CMTPOL] enables the IRO signal to be active-high or active-low.

The following table describes the functions of the modulators in different modes:

Table 38-20. Mode functions

Mode	Function
Time	The modulator can gate the carrier onto the modulator output.
Baseband	The modulator can control the logic level of the modulator output.

Table continues on the next page...

Table 38-20. Mode functions (continued)

Mode	Function
FSK	The modulator can count carrier periods and instruct the carrier generator to alternate between two carrier frequencies whenever a modulation period consisting of mark and space counts, expires.

The modulator provides a simple method to control protocol timing. The modulator has a minimum resolution of 1.0 µs with an 8 MHz. It can count bus clocks to provide real-time control, or carrier clocks for self-clocked protocols.

The modulator includes a 17-bit down counter with underflow detection. The counter is loaded from the 16-bit modulation mark period buffer registers, CMD1 and CMD2. The most significant bit is loaded with a logic 0 and serves as a sign bit.

When	Then
The counter holds a positive value	The modulator gate is open and the carrier signal is driven to the transmitter block.
The counter underflows	The modulator gate is closed and a 16-bit comparator is enabled which compares the logical complement of the value of the down counter with the contents of the modulation space period register which has been loaded from the registers, CMD3 and CMD4.

When a match is obtained, the cycle repeats by opening the modulator gate, reloading the counter with the contents of CMD1 and CMD2, and reloading the modulation space period register with the contents of CMD3 and CMD4.

The modulation space period is activated when the carrier signal is low to prohibit cutting off the high pulse of a carrier signal. If the carrier signal is high, the modulator extends the mark period until the carrier signal becomes low. To deassert the space period and assert the mark period, the carrier signal must have gone low to ensure that a space period is not erroneously shortened.

If the contents of the modulation space period register are all zeroes, the match will be immediate and no space period will be generated, for instance, for FSK protocols that require successive bursts of different frequencies).

MSC[MCGEN] must be set to enable the modulator timer.

The following figure presents the block diagram of the modulator.

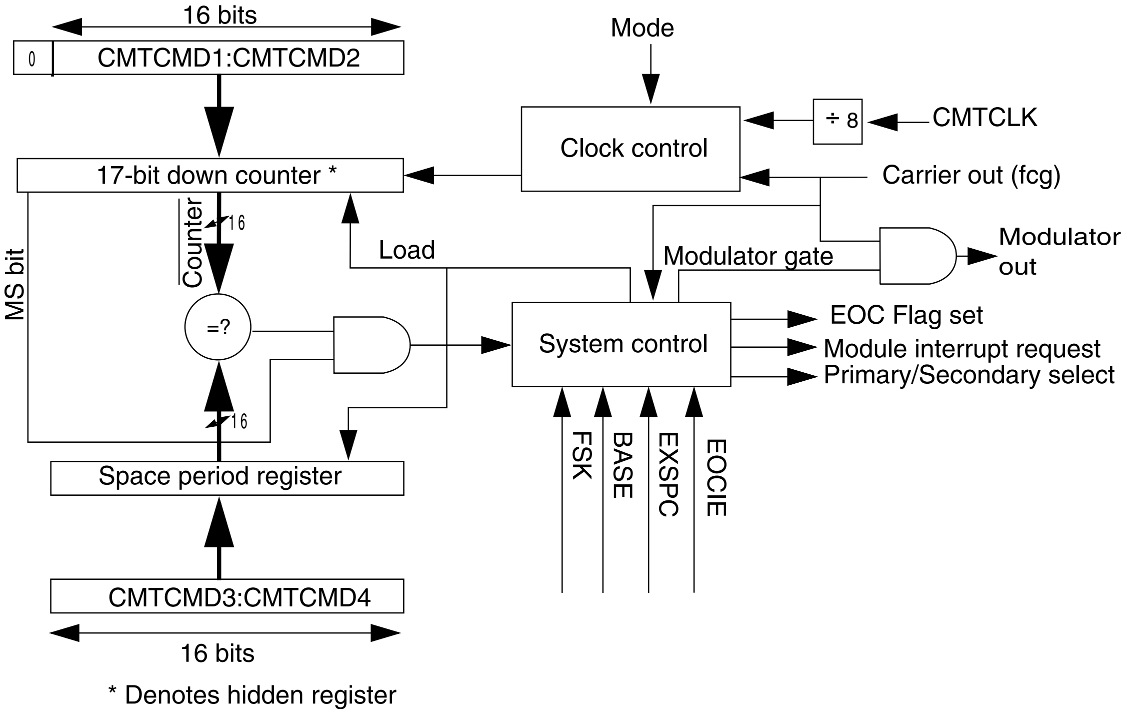

16 bits
0
CMTCMD1:CMTCMD2
Mode
Clock control
÷ 8
CMTCLK
Carrier out (fcg)
17-bit down counter *
MS bit
Counter
16
Load
Modulator gate
Modulator out
=?
System control
EOC Flag set
Module interrupt request
Primary/Secondary select
16
Space period register
FSK
BASE
EXSPC
EOCIE
CMTCMD3:CMTCMD4
16 bits
* Denotes hidden register

Figure 38-16. Modulator block diagram

38.7.3.1 Time mode

When the modulator operates in Time mode, or, when MSC[MCGEN] is set, and MSC[BASE] and MSC[FSK] are cleared:

- The modulation mark period consists of an integer number of (CMTCLK ÷ 8) clock periods.
- The modulation space period consists of 0 or an integer number of (CMTCLK ÷ 8) clock periods.

With an 8 MHz IF and MSC[CMTDIV] = 00, the modulator resolution is 1 μs and has a maximum mark and space period of about 65.535 ms each . See Figure 38-17 for an example of the Time and Baseband mode outputs.

The mark and space time equations for Time and Baseband mode are:

$t_{mark} = (CMD1:CMD2 + 1) \div (f_{CMTCLK} \div 8)$

$t_{space} = CMD3:CMD4 \div (f_{CMTCLK} \div 8)$

where CMD1:CMD2 and CMD3:CMD4 are the decimal values of the concatenated registers.

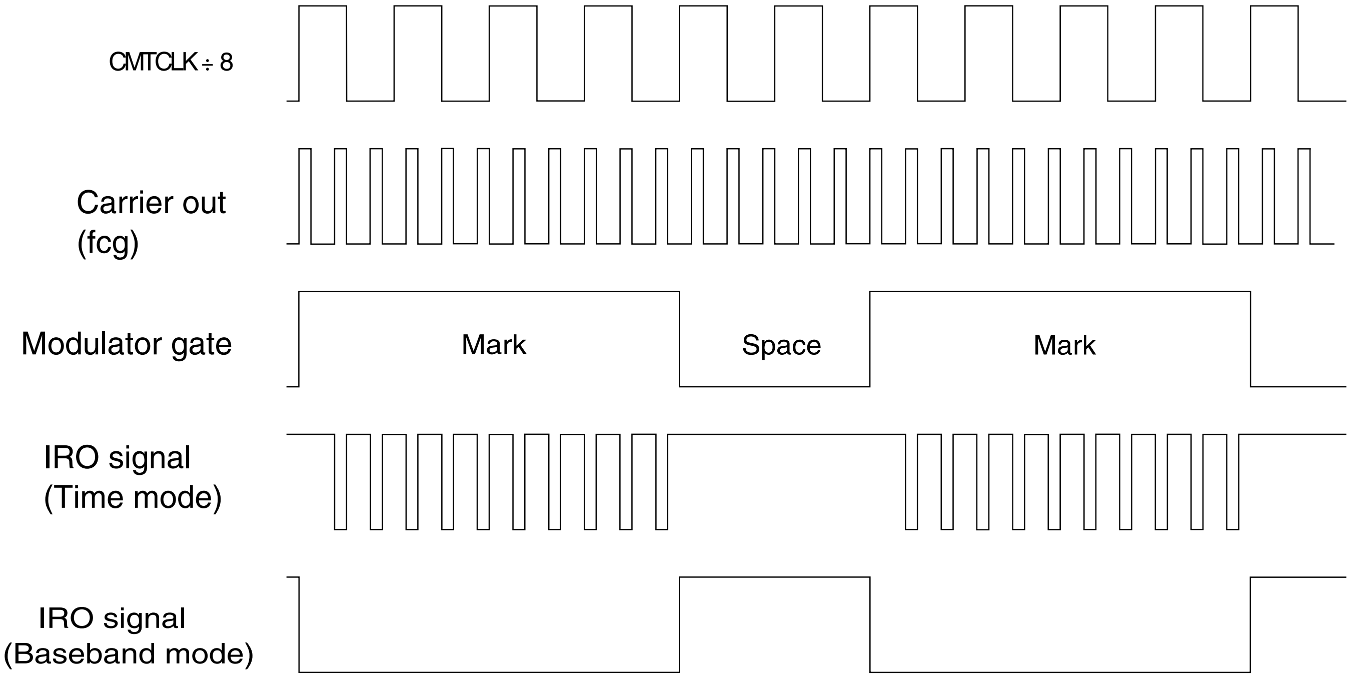


Figure 38-17. Example: CMT output in Time and Baseband modes with OC[CMTPOL]=0

38.7.3.2 Baseband mode

Baseband mode, that is, when MSC[MCGEN] and MSC[BASE] are set, is a derivative of Time mode, where the mark and space period is based on (CMTCLK ÷ 8) counts. The mark and space calculations are the same as in Time mode.

In this mode, the modulator output will be at a logic 1 for the duration of the mark period and at a logic 0 for the duration of a space period. See Figure 38-17 for an example of the output for both Baseband and Time modes. In the example, the carrier out frequency (f_{cg}) is generated with a high count of 0x01 and a low count of 0x02 that results in a divide of 3 of CMTCLK with a 33% duty cycle. The modulator down counter was loaded with the value 0x0003 and the space period register with 0x0002.

Note

The waveforms in Figure 38-17 and Figure 38-18 are for the purpose of conceptual illustration and are not meant to represent precise timing relationships between the signals shown.

38.7.3.3 FSK mode

When the modulator operates in FSK mode, that is, when MSC[MCGEN] and MSC[FSK] are set, and MSC[BASE] is cleared:

- The modulation mark and space periods consist of an integer number of carrier clocks (space period can be zero).
- When the mark period expires, the space period is transparently started as in Time mode.
- The carrier generator toggles between primary and secondary data register values whenever the modulator space period expires.

The space period provides an interpulse gap (no carrier). If CMD3:CMD4 = 0x0000, then the modulator and carrier generator will switch between carrier frequencies without a gap or any carrier glitches (zero space).

Using timing data for carrier burst and interpulse gap length calculated by the CPU, FSK mode can automatically generate a phase-coherent, dual-frequency FSK signal with programmable burst and interburst gaps.

The mark and space time equations for FSK mode are:

$t_{mark} = (\text{CMD1:CMD2} + 1) \div f_{cg}$

$t_{space} = (\text{CMD3:CMD4}) \div f_{cg}$

Where f_{cg} is the frequency output from the carrier generator. The example in Figure 38-18 shows what the IRO signal looks like in FSK mode with the following values:

- CMD1:CMD2 = 0x0003
- CMD3:CMD4 = 0x0002
- Primary carrier high count = 0x01
- Primary carrier low count = 0x02
- Secondary carrier high count = 0x03
- Secondary carrier low count = 0x01

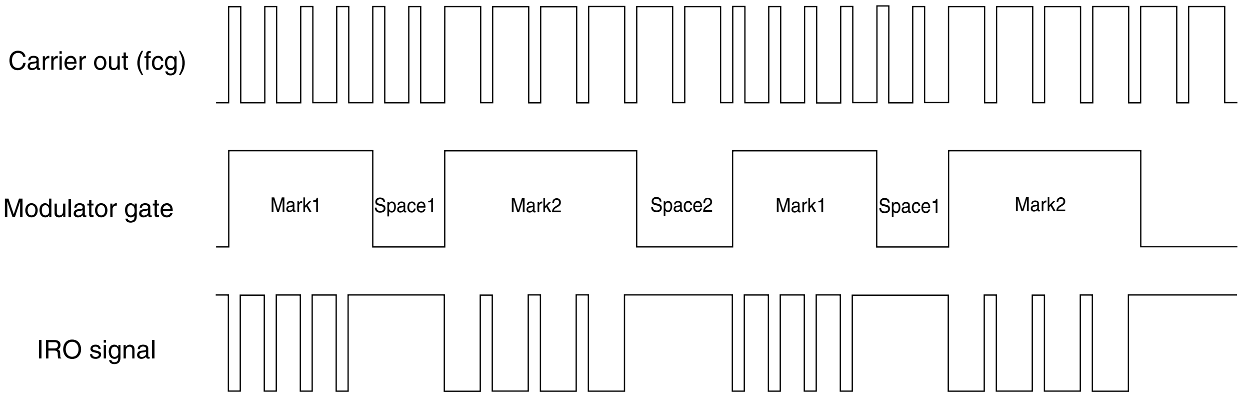


Figure 38-18. Example: CMT output in FSK mode

38.7.4 Extended space operation

In either Time, Baseband, or FSK mode, the space period can be made longer than the maximum possible value of the space period register. Setting MSC[EXSPC] will force the modulator to treat the next modulation period beginning with the next load of the counter and space period register, as a space period equal in length to the mark and space counts combined. Subsequent modulation periods will consist entirely of these extended space periods with no mark periods. Clearing MSC[EXSPC] will return the modulator to standard operation at the beginning of the next modulation period.

38.7.4.1 EXSPC operation in Time mode

To calculate the length of an extended space in Time or Baseband mode, add the mark and space times and multiply by the number of modulation periods when MSC[EXSPC] is set.

$t_{exspace} = (t_{mark} + t_{space})$ * (number of modulation periods)

For an example of extended space operation, see Figure 38-19.

Note

The extended space enable feature can be used to emulate a zero mark event.

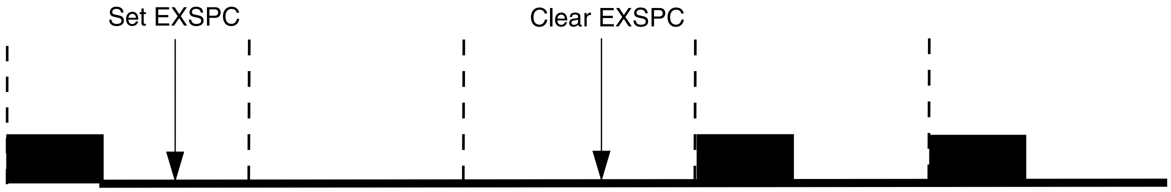


Figure 38-19. Extended space operation

38.7.4.2 EXSPC operation in FSK mode

In FSK mode, the modulator continues to count carrier out clocks, alternating between the primary and secondary registers at the end of each modulation period.

To calculate the length of an extended space in FSK mode, it is required to know whether MSC[EXSPC] was set on a primary or secondary modulation period, and the total number of both primary and secondary modulation periods completed while MSC[EXSPC] is high. A status bit for the current modulation is not accessible to the

CPU. If necessary, software must maintain tracking of the current primary or secondary modulation cycle. The extended space period ends at the completion of the space period time of the modulation period during which MSC[EXSPC]is cleared.

The following table depicts the equations which can be used to calculate the extended space period depending on when MSC[EXSPC] is set.

If	Then
MSC[EXSPC] was set during a primary modulation cycle	Use the equation: $t_{exspace} = (t_{space})_p + (t_{mark} + t_{space})_s + (t_{mark} + t_{space})_p + \ldots$
MSC[EXSPC] bit was set during a secondary modulation cycle	Use the equation: $t_{exspace} = (t_{space})_s + (t_{mark} + t_{space})_p + (t_{mark} + t_{space})_s + \ldots$

Where the subscripts p and s refer to mark and space times for the primary and secondary modulation cycles.

38.8 CMT interrupts and DMA

The CMT generates an interrupt request or a DMA transfer request according to MSC[EOCIE], MSC[EOCF], DMA[DMA] bits.

Table 38-23. DMA transfer request x CMT interrupt request

MSC[EOCF]	DMA[DMA]	MSC[EOCIE]	DMA transfer request	CMT interrupt request
0	X	X	0	0
1	X	0	0	0
1	0	1	0	1
1	1	1	1	0

MSC[EOCF] is set:

- When the modulator is not currently active and MSC[MCGEN] is set to begin the initial CMT transmission.
- At the end of each modulation cycle when the counter is reloaded from CMD1:CMD2, while MSC[MCGEN] is set.

When MSC[MCGEN] is cleared and then set before the end of the modulation cycle, MSC[EOCF] will not be set when MSC[MCGEN] is set, but will become set at the end of the current modulation cycle.

When MSC[MCGEN] becomes disabled, the CMT module does not set MSC[EOCF] at the end of the last modulation cycle.

If MSC[EOCIE] is high when MSC[EOCF] is set, the CMT module will generate an interrupt request or a DMA transfer request.

MSC[EOCF] must be cleared to prevent from being generated by another event like interrupt or DMA request, after exiting the service routine. See the following table.

Table 38-24. How to clear MSC[EOCF]

DMA[DMA]	MSC[EOCIE]	Description
0	X	MSC[EOCF] is cleared by reading MSC followed by an access of CMD2 or CMD4.
1	X	MSC[EOCF] is cleared by the CMT DMA transfer done.

The EOC interrupt is coincident with:

- Loading the down-counter with the contents of CMD1:CMD2
- Loading the space period register with the contents of CMD3:CMD4

The EOC interrupt provides a means for the user to reload new mark/space values into the modulator data registers. Modulator data register updates will take effect at the end of the current modulation cycle.

NOTE

The down-counter and space period register are updated at the end of every modulation cycle, irrespective of interrupt handling and the state of MSC[EOCF].

Chapter 39
Real Time Clock (RTC)

39.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration chapter.

39.1.1 Features

The RTC module features include:

- Independent power supply, POR and 32 kHz crystal oscillator
- 32-bit seconds counter with roll-over protection and 32-bit alarm
- 16-bit prescaler with compensation that can correct errors between 0.12 ppm and 3906 ppm
- Register write protection
  - Lock register requires VBAT POR or software reset to enable write access
  - Access control registers require system reset to enable read and/or write access
- 1 Hz square wave output

39.1.2 Modes of operation

The RTC operates in one of two modes of operation, chip power-up and chip power-down.

During chip power-down, RTC is powered from the backup power supply (VBAT) and is electrically isolated from the rest of the chip but continues to increment the time counter (if enabled) and retain the state of the RTC registers. The RTC registers are not accessible.

During chip power-up, RTC remains powered from the backup power supply (VBAT). All RTC registers are accessible by software and all functions are operational. If enabled, the 32.768 kHz clock can be supplied to the rest of the chip.

39.1.3 RTC signal descriptions

Table 39-1. RTC signal descriptions

Signal	Description	I/O
EXTAL32	32.768 kHz oscillator input	I
XTAL32	32.768 kHz oscillator output	O
RTC_CLKOUT	1Hz square-wave output	O
$\overline{\text{RTC\_WAKEUP}}$	Wakeup for external device	O

39.1.3.1 RTC clock output

The clock to the seconds counter is available on the RTC_CLKOUT signal. It is a 1Hz square wave output.

39.1.3.2 RTC wakeup pin

The RTC wakeup pin is an open drain, active low, output that allows the RTC to wakeup the chip via an external component. The wakeup pin asserts when the wakeup pin enable is set, the RTC interrupt is asserted and the chip is powered down. The wakeup pin does not assert from the RTC seconds interrupt.

The wakeup pin is optional and may not be implemented on all devices.

39.2 Register definition

All registers must be accessed using 32-bit writes and all register accesses incur three wait states.

Write accesses to any register by non-supervisor mode software, when the supervisor access bit in the control register is clear, will terminate with a bus error.

Read accesses by non-supervisor mode software complete as normal.

Writing to a register protected by the write access register or lock register does not generate a bus error, but the write will not complete.

Reading a register protected by the read access register does not generate a bus error, but the register will read zero.

RTC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_D000	RTC Time Seconds Register (RTC_TSR)	32	R/W	0000_0000h	39.2.1/ 867
4003_D004	RTC Time Prescaler Register (RTC_TPR)	32	R/W	0000_0000h	39.2.2/ 868
4003_D008	RTC Time Alarm Register (RTC_TAR)	32	R/W	0000_0000h	39.2.3/ 868
4003_D00C	RTC Time Compensation Register (RTC_TCR)	32	R/W	0000_0000h	39.2.4/ 869
4003_D010	RTC Control Register (RTC_CR)	32	R/W	0000_0000h	39.2.5/ 870
4003_D014	RTC Status Register (RTC_SR)	32	R/W	0000_0001h	39.2.6/ 871
4003_D018	RTC Lock Register (RTC_LR)	32	R/W	0000_00FFh	39.2.7/ 872
4003_D01C	RTC Interrupt Enable Register (RTC_IER)	32	R/W	0000_0007h	39.2.8/ 874
4003_D800	RTC Write Access Register (RTC_WAR)	32	R/W	0000_00FFh	39.2.9/ 875
4003_D804	RTC Read Access Register (RTC_RAR)	32	R/W	0000_00FFh	39.2.10/ 876

39.2.1 RTC Time Seconds Register (RTC_TSR)

Address: RTC_TSR is 4003_D000h base + 0h offset = 4003_D000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TSR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_TSR field descriptions

Field	Description
31–0 TSR	Time Seconds Register When the time counter is enabled, the TSR is read only and increments once a second provided SR[TOF] or SR[TIF] are not set. The time counter will read as zero when SR[TOF] or SR[TIF] are set. When the time counter is disabled, the TSR can be read or written. Writing to the TSR when the time counter is disabled will clear the SR[TOF] and/or the SR[TIF]. Writing to the TSR register with zero is supported, but not recommended since TSR will read as zero when SR[TIF] or SR[TOF] are set (indicating the time is invalid).

39.2.2 RTC Time Prescaler Register (RTC_TPR)

Address: RTC_TPR is 4003_D000h base + 4h offset = 4003_D004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TPR															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_TPR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 TPR	Time Prescaler Register When the time counter is enabled, the TPR is read only and increments every 32.768 kHz clock cycle. The time counter will read as zero when SR[TOF] or SR[TIF] are set. When the time counter is disabled, the TPR can be read or written. The TSR[TSR] increments when bit 14 of the TPR transitions from a logic one to a logic zero.

39.2.3 RTC Time Alarm Register (RTC_TAR)

Address: RTC_TAR is 4003_D000h base + 8h offset = 4003_D008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TAR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_TAR field descriptions

Field	Description
31–0 TAR	Time Alarm Register When the time counter is enabled, the SR[TAF] is set whenever the TAR[TAR] equals the TSR[TSR] and the TSR[TSR] increments. Writing to the TAR clears the SR[TAF].

39.2.4 RTC Time Compensation Register (RTC_TCR)

Address: RTC_TCR is 4003_D000h base + Ch offset = 4003_D00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CIC								TCV								CIR								TCR							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RTC_TCR field descriptions

Field	Description
31–24 CIC	Compensation Interval Counter Current value of the compensation interval counter. If the compensation interval counter equals zero then it is loaded with the contents of the CIR. If the CIC does not equal zero then it is decremented once a second.
23–16 TCV	Time Compensation Value Current value used by the compensation logic for the present second interval. Updated once a second if the CIC equals 0 with the contents of the TCR field. If the CIC does not equal zero then it is loaded with zero (compensation is not enabled for that second increment).
15–8 CIR	Compensation Interval Register Configures the compensation interval in seconds from 1 to 256 to control how frequently the TCR should adjust the number of 32.768 kHz cycles in each second. The value written should be one less than the number of seconds (for example, write zero to configure for a compensation interval of one second). This register is double buffered and writes do not take affect until the end of the current compensation interval.
7–0 TCR	Time Compensation Register Configures the number of 32.768 kHz clock cycles in each second. This register is double buffered and writes do not take affect until the end of the current compensation interval. 80h Time prescaler register overflows every 32896 clock cycles. ... ... FFh Time prescaler register overflows every 32769 clock cycles. 00h Time prescaler register overflows every 32768 clock cycles. 01h Time prescaler register overflows every 32767 clock cycles. ... ... 7Fh Time prescaler register overflows every 32641 clock cycles.

39.2.5 RTC Control Register (RTC_CR)

Address: RTC_CR is 4003_D000h base + 10h offset = 4003_D010h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0	Reserved	SC2P	SC4P	SC8P	SC16P	CLKO	OSCE
W		0						
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	0				UM	SUP	WPE	SWR
W								
Reset	0	0	0	0	0	0	0	0

RTC_CR field descriptions

Field	Description
31–15 Reserved	This read-only field is reserved and always has the value zero.
14 Reserved	This field is reserved.
13 SC2P	Oscillator 2pF load configure 0 Disable the load. 1 Enable the additional load.
12 SC4P	Oscillator 4pF load configure 0 Disable the load. 1 Enable the additional load.
11 SC8P	Oscillator 8pF load configure 0 Disable the load. 1 Enable the additional load.
10 SC16P	Oscillator 16pF load configure 0 Disable the load. 1 Enable the additional load.
9 CLKO	Clock Output

Table continues on the next page...

RTC_CR field descriptions (continued)

Field	Description
	0 The 32kHz clock is output to other peripherals 1 The 32kHz clock is not output to other peripherals
8 OSCE	Oscillator Enable 0 32.768 kHz oscillator is disabled. 1 32.768 kHz oscillator is enabled. After setting this bit, wait the oscillator startup time before enabling the time counter to allow the 32.768 kHz clock time to stabilize.
7–4 Reserved	This read-only field is reserved and always has the value zero.
3 UM	Update Mode Allows the SR[TCE] to be written even when the Status Register is locked. When set, the SR[TCE] can always be written if the SR[TIF] or SR[TOF] are set or if the SR[TCE] is clear. 0 Registers cannot be written when locked. 1 Registers can be written when locked under limited conditions.
2 SUP	Supervisor Access 0 Non-supervisor mode write accesses are not supported and generate a bus error. 1 Non-supervisor mode write accesses are supported.
1 WPE	Wakeup Pin Enable The wakeup pin is optional and not available on all devices. 0 Wakeup pin is disabled. 1 Wakeup pin is enabled and wakeup pin asserts if the RTC interrupt asserts and the chip is powered down.
0 SWR	Software Reset 0 No effect 1 Resets all RTC registers except for the SWR bit and the RTC_WAR and RTC_RAR registers. The SWR bit is cleared after VBAT POR and by software explicitly clearing it.

39.2.6 RTC Status Register (RTC_SR)

Address: RTC_SR is 4003_D000h base + 14h offset = 4003_D014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											TCE	0	TAF	TOF	TIF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

RTC_SR field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value zero.
4 TCE	Time Counter Enable When time counter is disabled the TSR register and TPR register are writeable, but do not increment. When time counter is enabled the TSR register and TPR register are not writeable, but increment. 0 Time counter is disabled. 1 Time counter is enabled.
3 Reserved	This read-only field is reserved and always has the value zero.
2 TAF	Time Alarm Flag Time alarm flag is set when the TAR[TAR] equals the TSR[TSR] and the TSR[TSR] increments. This bit is cleared by writing the TAR register. 0 Time alarm has not occurred. 1 Time alarm has occurred.
1 TOF	Time Overflow Flag Time overflow flag is set when the time counter is enabled and overflows. The TSR and TPR do not increment and read as zero when this bit is set. This bit is cleared by writing the TSR register when the time counter is disabled. 0 Time overflow has not occurred. 1 Time overflow has occurred and time counter is read as zero.
0 TIF	Time Invalid Flag The time invalid flag is set on VBAT POR or software reset. The TSR and TPR do not increment and read as zero when this bit is set. This bit is cleared by writing the TSR register when the time counter is disabled. 0 Time is valid. 1 Time is invalid and time counter is read as zero.

39.2.7 RTC Lock Register (RTC_LR)

Address: RTC_LR is 4003_D000h base + 18h offset = 4003_D018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								1	LRL	SRL	CRL	TCL	1		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

RTC_LR field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 Reserved	This read-only field is reserved and always has the value one.
6 LRL	Lock Register Lock Once cleared, this bit can only be set by VBAT POR or software reset. 0 Lock register is locked and writes are ignored. 1 Lock register is not locked and writes complete as normal.
5 SRL	Status Register Lock Once cleared, this bit can only be set by VBAT POR or software reset. 0 Status register is locked and writes are ignored. 1 Status register is not locked and writes complete as normal.
4 CRL	Control Register Lock Once cleared, this bit can only be set by VBAT POR. 0 Control register is locked and writes are ignored. 1 Control register is not locked and writes complete as normal.
3 TCL	Time Compensation Lock Once cleared, this bit can only be set by VBAT POR or software reset. 0 Time compensation register is locked and writes are ignored. 1 Time compensation register is not locked and writes complete as normal.
2–0 Reserved	This read-only field is reserved and always has the value one.

39.2.8 RTC Interrupt Enable Register (RTC_IER)

Address: RTC_IER is 4003_D000h base + 1Ch offset = 4003_D01Ch

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R W	Reserved			TSIE	Reserved	TAIE	TOIE	TIIE
Reset	0	0	0	0	0	1	1	1

RTC_IER field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7–5 Reserved	This field is reserved.
4 TSIE	Time Seconds Interrupt Enable The seconds interrupt is an edge-sensitive interrupt with a dedicated interrupt vector. It is generated once a second and requires no software overhead (there is no corresponding status flag to clear). 0 Seconds interrupt is disabled. 1 Seconds interrupt is enabled.
3 Reserved	This field is reserved.
2 TAIE	Time Alarm Interrupt Enable 0 Time alarm flag does not generate an interrupt. 1 Time alarm flag does generate an interrupt.
1 TOIE	Time Overflow Interrupt Enable 0 Time overflow flag does not generate an interrupt. 1 Time overflow flag does generate an interrupt.
0 TIIE	Time Invalid Interrupt Enable

Table continues on the next page...

RTC_IER field descriptions (continued)

Field	Description
	0 Time invalid flag does not generate an interrupt. 1 Time invalid flag does generate an interrupt.

39.2.9 RTC Write Access Register (RTC_WAR)

Address: RTC_WAR is 4003_D000h base + 800h offset = 4003_D800h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								IERW	LRW	SRW	CRW	TCRW	TARW	TPRW	TSRW
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

RTC_WAR field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 IERW	Interrupt Enable Register Write Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the interupt enable register are ignored. 1 Writes to the interrupt enable register complete as normal.
6 LRW	Lock Register Write Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the lock register are ignored. 1 Writes to the lock register complete as normal.
5 SRW	Status Register Write Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the status register are ignored. 1 Writes to the status register complete as normal.
4 CRW	Control Register Write Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the control register are ignored. 1 Writes to the control register complete as normal.
3 TCRW	Time Compensation Register Write Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset.

Table continues on the next page...

RTC_WAR field descriptions (continued)

Field	Description
	0 Writes to the time compensation register are ignored. 1 Writes to the time compensation register complete as normal.
2 TARW	Time Alarm Register Write Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the time alarm register are ignored. 1 Writes to the time alarm register complete as normal.
1 TPRW	Time Prescaler Register Write Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the time prescaler register are ignored. 1 Writes to the time prescaler register complete as normal.
0 TSRW	Time Seconds Register Write Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset. 0 Writes to the time seconds register are ignored. 1 Writes to the time seconds register complete as normal.

39.2.10 RTC Read Access Register (RTC_RAR)

Address: RTC_RAR is 4003_D000h base + 804h offset = 4003_D804h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																								IERR	LRR	SRR	CRR	TCRR	TARR	TPRR	TSRR
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

RTC_RAR field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value zero.
7 IERR	Interrupt Enable Register Read Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the interrupt enable register are ignored. 1 Reads to the interrupt enable register complete as normal.
6 LRR	Lock Register Read Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset.

Table continues on the next page...

RTC_RAR field descriptions (continued)

Field	Description
	0 Reads to the lock register are ignored. 1 Reads to the lock register complete as normal.
5 SRR	Status Register Read Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the status register are ignored. 1 Reads to the status register complete as normal.
4 CRR	Control Register Read Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the control register are ignored. 1 Reads to the control register complete as normal.
3 TCRR	Time Compensation Register Read Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset 0 Reads to the time compensation register are ignored. 1 Reads to the time compensation register complete as normal.
2 TARR	Time Alarm Register Read Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the time alarm register are ignored. 1 Reads to the time alarm register complete as normal.
1 TPRR	Time Prescaler Register Read Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the time prescaler register are ignored. 1 Reads to the time prescaler register complete as normal.
0 TSRR	Time Seconds Register Read Once cleared, this bit is only set by system reset. It is not affected by VBAT POR or software reset. 0 Reads to the time seconds register are ignored. 1 Reads to the time seconds register complete as normal.

39.3 Functional description

39.3.1 Power, clocking and reset

The RTC is an always powered block that is powered by the battery power supply (VBAT). The battery power supply ensures that the RTC registers retain their state during chip power-down and that the RTC time counter remains operational.

The time counter within the RTC is clocked by a 32.768 kHz clock and can supply this clock to other peripherals. The 32.768 kHz clock can only be sourced from an external crystal using the oscillator that is part of the RTC module.

The RTC includes its own analog POR block, which generates a power-on-reset signal whenever the RTC module is powered up and initializes all RTC registers to their default state. A software reset bit can also initialize all RTC registers. The RTC also monitors the chip power supply and electrically isolates itself when the rest of the chip is powered down.

Any attempt to access an RTC register (except the access control registers) when VBAT is powered down, when the RTC is electrically isolated, or when VBAT POR is asserted, will result in a bus error.

39.3.1.1 Oscillator control

The 32.768 kHz crystal oscillator is disabled at VBAT POR and must be enabled by software. After enabling the cystal oscillator, wait the oscillator startup time before setting the SR[TCE] bit or using the oscillator clock external to the RTC.

The crystal oscillator includes tunable capacitors that can be configured by software. Do not change the capacitance unless the oscillator is disabled.

39.3.1.2 Software reset

Writing one to the CR[SWR] forces the equivalent of a VBAT POR to the rest of the RTC module. The CR[SWR] is not affected by the software reset and must be cleared by software. The access control registers are not affected by either VBAT POR or the software reset; they are reset by the chip reset.

39.3.1.3 Supervisor access

When the supervisor access control bit is clear, only supervisor mode software can write to the RTC registers, non-supervisor mode software will generate a bus error. Both supervisor and non-supervisor mode software can always read the RTC registers.

39.3.2 Time counter

The time counter consists of a 32-bit seconds counter that increments once every second and a 16-bit prescaler register that increments once every 32.768 kHz clock cycle.

The time seconds register and time prescaler register can only be written when the SR[TCE] bit is clear. Always write to the prescaler register before writing to the seconds register, since the seconds register increments on the falling edge of bit 14 of the prescaler register.

The time prescaler register increments provided the SR[TCE] bit is set, the SR[TIF] is clear, the SR[TOF] is clear and the 32.768 kHz clock source is present. After enabling the oscillator, wait the oscillator startup time before setting the SR[TCE] bit to allow time for the oscillator clock output to stabilize.

If the time seconds register overflows then the SR[TOF] will set and the time prescaler register will stop incrementing. Clear the SR[TOF] by initializing the time seconds register. The time seconds register and time prescaler register read as zero whenever the SR[TOF] is set.

The SR[TIF] is set on VBAT POR and software reset and is cleared by initializing the time seconds register. The time seconds register and time prescaler register read as zero whenever the SR[TIF] is set.

39.3.3 Compensation

The compensation logic provides an accurate and wide compensation range and can correct errors as high as 3906 ppm and as low as 0.12 ppm. Note that the compensation factor must be calculated externally to the RTC and supplied by software to the compensation register. The RTC itself does not calculate the amount of compensation that is required, although the 1 Hz clock is output to an external pin in support of external calibration logic.

Crystal compensation can be supported by using firmware and crystal characteristics to determine the compensation amount. Temperature compensation can be supported by firmware that periodically measures the external temperature (via ADC) and updates the compensation register based on a look-up table that specifies the change in crystal frequency over temperature.

The compensation logic alters the number of 32.768 kHz clock cycles it takes for the prescaler register to overflow and increment the time seconds counter. The time compensation value is used to adjust the number of clock cycles between -127 and +128.

Cycles are added or subtracted from the prescaler register when the prescaler register equals 0x3FFF and then increments. The compensation interval is used to adjust the frequency at which the time compensation value is used (from once a second to once every 256 seconds).

Updates to the time compensation register will not take effect until the next time the time seconds register increments and provided the previous compensation interval has expired. When the compensation interval is set to other than once a second then the compensation is applied in the first second interval and the remaining second intervals receive no compensation.

Compensation is disabled by configuring the time compensation register to zero.

39.3.4 Time alarm

The time alarm register, SR[TAF] and IER[TAIE] allow the RTC to generate an interrupt at a predefined time. The 32-bit time alarm register is compared with the 32-bit time seconds register each time it increments. The SR[TAF] will set when the time alarm register equals the time seconds register and the time seconds register increments.

The time alarm flag is cleared by writing the time alarm register. This will usually be the next alarm value, although writing a value that is less than the time seconds register (such as zero) will prevent the time alarm flag from setting again. The time alarm flag cannot otherwise be disabled, although the interrupt it generates is enabled or disabled by IER[TAIE].

39.3.5 Update mode

The update mode bit (CR[UM]) in the control register configures software write access to the time counter enable (SR[TCE]) bit. When CR[UM] is clear, SR[TCE] can only be written when the LR[SRL] bit is set. When CR[UM] is set, the SR[TCE] can also be written when SR[TCE] is clear or when SR[TIF] or SR[TOF] are set. This allows the time seconds and prescaler registers to be initialized whenever time is invalidated, while preventing the time seconds and prescaler registers from being changed on the fly. When LR[SRL] is set, the CR[UM] bit has no effect on SR[TCE].

39.3.6 Register lock

The lock register can be used to block write accesses to certain registers until the next VBAT POR or software reset. Locking the control register will disable the software reset. Locking the lock register will block future updates to the lock register.

Write accesses to a locked register are ignored and do not generate a bus error.

39.3.7 Access control

The read access and write access registers are implemented in the chip power domain and reset on the chip reset (they are not affected by the VBAT POR or the software reset). They are used to block read or write accesses to each register until the next chip system reset. When accesses are blocked the bus access is not seen in the VBAT power supply and does not generate a bus error.

39.3.8 Interrupt

The RTC Interrupt is asserted whenever a status flag and the corresponding interrupt enable bit are both set. It is always asserted on VBAT POR, software reset and when the VBAT power supply is powered down. The RTC interrupt is enabled at the chip level by enabling the chip-specific RTC clock gate control bit. The RTC Interrupt can be used to wakeup the chip from any low power mode.

The optional RTC seconds interrupt is an edge-sensitive interrupt with a dedicated interrupt vector that is generated once a second and requires no software overhead (there is no corresponding status flag to clear). It is enabled in the RTC by the time seconds interrupt enable bit and enabled at the chip level by setting the chip-specific RTC clock gate control bit. The RTC seconds interrupt does not cause the RTC wakeup pin to assert. This interrupt is optional and may not be implemented on all devices.

Chapter 40
Universal Serial Bus OTG Controller (USBOTG)

40.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

This section describes the USB. The OTG implementation in this module provides limited host functionality and device solutions for implementing a USB 2.0 full-speed/ low-speed compliant peripheral. The OTG implementation supports the On-The-Go (OTG) addendum to the USB 2.0 Specification. Only one protocol can be active at any time. A negotiation protocol must be used to switch to a USB host functionality from a USB device. This is known as the Master Negotiation Protocol (MNP).

40.1.1 USB

The USB is a cable bus that supports data exchange between a host computer and a wide range of simultaneously accessible peripherals. The attached peripherals share USB bandwidth through a host-scheduled, token-based protocol. The bus allows peripherals to be attached, configured, used, and detached while the host and other peripherals are in operation.

USB software provides a uniform view of the system for all application software, hiding implementation details making application software more portable. It manages the dynamic attach and detach of peripherals.

There is only one host in any USB system. The USB interface to the host computer system is referred to as the Host Controller.

There may be multiple USB devices in any system such as joysticks, speakers, printers, etc. USB devices present a standard USB interface in terms of comprehension, response, and standard capability.

The host initiates transactions to specific peripherals, whereas the device responds to control transactions. The device sends and receives data to and from the host using a standard USB data format. USB 2.0 full-speed /low-speed peripherals operate at 12Mbit/s or 1.5 Mbit/s.

For additional information, see the USB 2.0 specification.

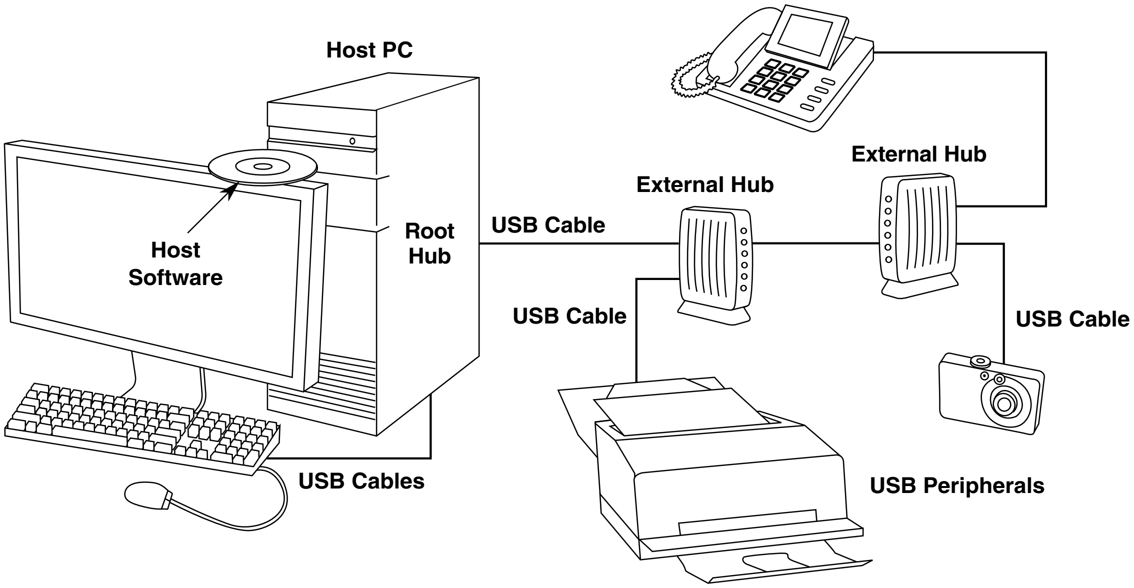


Figure 40-1. Example USB 2.0 system configuration

40.1.2 USB On-The-Go

USB is a popular standard for connecting peripherals and portable consumer electronic devices such as digital cameras and hand-held computers to host PCs. The On-The-Go (OTG) Supplement to the USB Specification extends USB to peer-to-peer application. Using USB OTG technology consumer electronics, peripherals, and portable devices can connect to each other to exchange data. For example, a digital camera can connect directly to a printer, or a keyboard can connect to a Personal Digital Assistant to exchange data.

With the USB On-The-Go product, you can develop a fully USB-compliant peripheral device that can also assume the role of a USB host. Software determines the role of the device based on hardware signals, and then initializes the device in the appropriate mode of operation (host or peripheral) based on how it is connected. After connecting the devices can negotiate using the OTG protocols to assume the role of host or peripheral based on the task to be accomplished.

For additional information, see the *On-The-Go Supplement to the USB 2.0 Specification*.

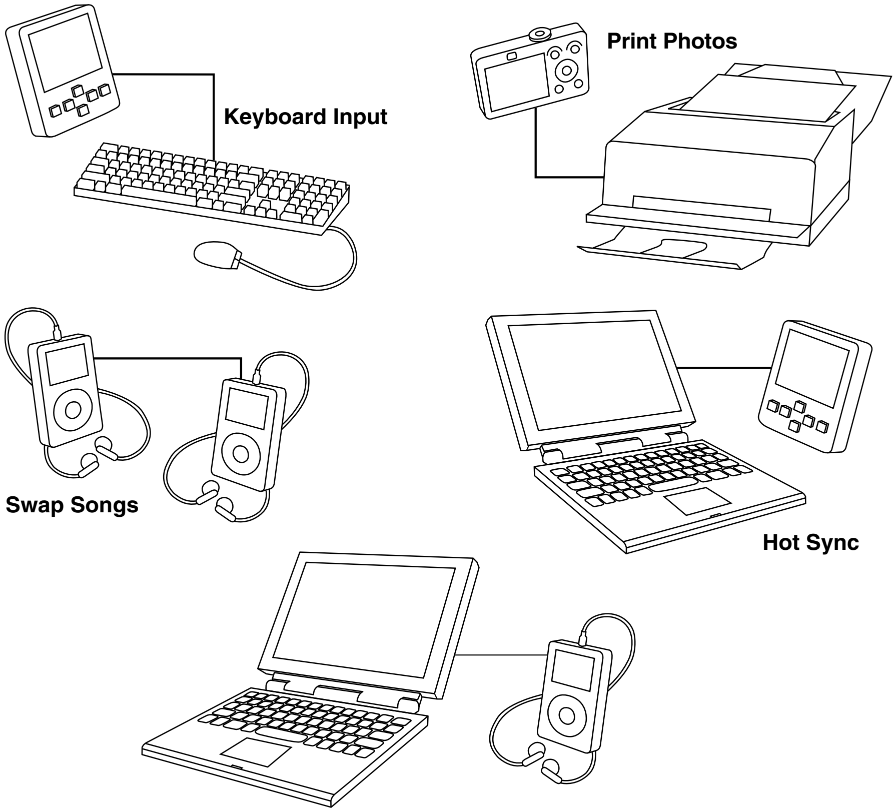


Figure 40-2. Example USB 2.0 On-The-Go configurations

40.1.3 USB-FS Features

- USB 1.1 and 2.0 compliant full-speed device controller
- 16 bidirectional end points
- DMA or FIFO data stream interfaces
- Low-power consumption
- On-The-Go protocol logic

40.2 Functional description

The USB-FS 2.0 full-speed/low-speed module communicates with the processor core through status registers, control registers, and data structures in memory.

40.2.1 Data Structures

The function of the device operation is to transfer a request in the memory image to and from the Universal Serial Bus. To efficiently manage USB endpoint communications the USB-FS implements a Buffer Descriptor Table (BDT) in system memory. See Figure 40-3.

40.3 Programmers interface

This section discusses the major components of the programming model for the USB module.

40.3.1 Buffer Descriptor Table

To efficiently manage USB endpoint communications the USB-FS implements a Buffer Descriptor Table (BDT) in system memory. The BDT resides on a 512-byte boundary in system memory and is pointed to by the BDT Page Registers. Every endpoint direction requires two 8-byte Buffer Descriptor (BD) entries. Therefore, a system with 16 fully bidirectional endpoints would require 512 bytes of system memory to implement the BDT. The two BD entries allows for an EVEN BD and ODD BD entry for each endpoint direction. This allows the microprocessor to process one BD while the USB-FS is processing the other BD. Double buffering BDs in this way allows the USB-FS to transfer data easily at the maximum throughput provided by USB.

The software API intelligently manages buffers for the USB-FS by updating the BDT when needed. This allows the USB-FS to efficiently manage data transmission and reception, while the microprocessor performs communication overhead processing and other function dependent applications. Because the buffers are shared between the microprocessor and the USB-FS, a simple semaphore mechanism is used to distinguish who is allowed to update the BDT and buffers in system memory. A semaphore, the OWN bit, is cleared to 0 when the BD entry is owned by the microprocessor. The microprocessor is allowed read and write access to the BD entry and the buffer in system memory when the OWN bit is 0. When the OWN bit is set to 1, the BD entry and the buffer in system memory are owned by the USB-FS. The USB-FS now has full read and write access and the microprocessor must not modify the BD or its corresponding data buffer. The BD also contains indirect address pointers to where the actual buffer resides in system memory. This indirect address mechanism is shown in the following diagram.

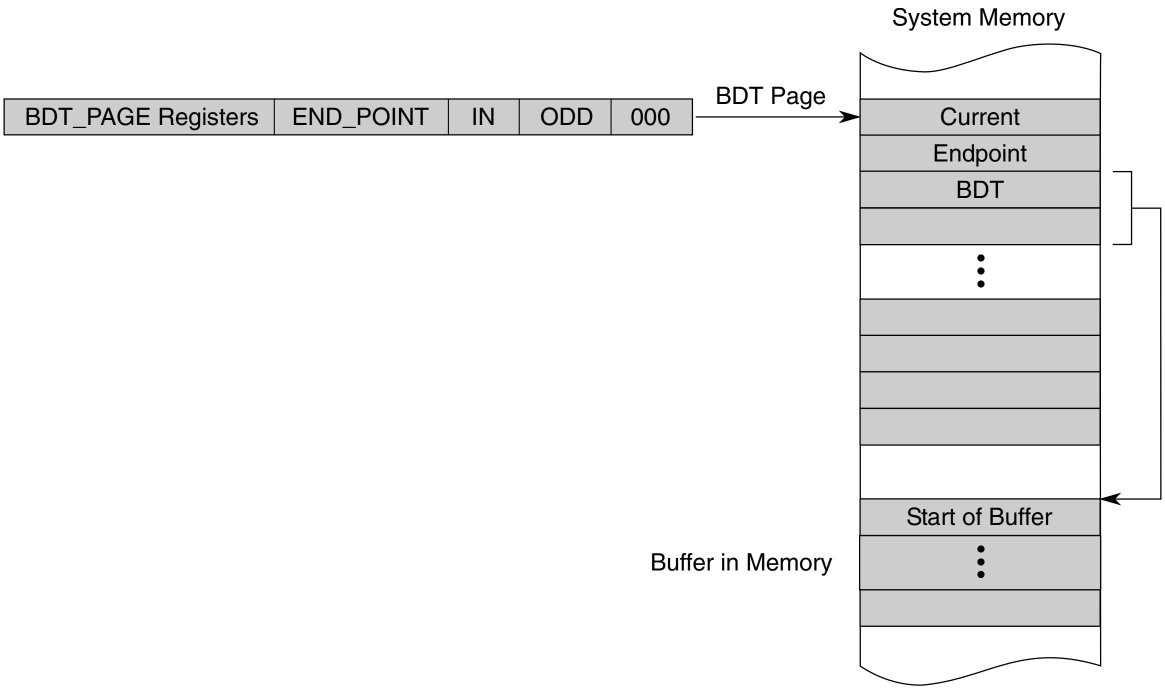

System Memory
BDT_PAGE Registers
END_POINT
IN
ODD
000
BDT Page
Current
Endpoint
BDT
Start of Buffer
Buffer in Memory

Figure 40-3. Buffer descriptor table

40.3.2 RX vs. TX as a USB target device or USB host

The USB-FS core uses software control to switch between two modes of operation:

- USB target device
- USB hosts

In either mode, USB host or USB target device, the same data paths and buffer descriptors are used for the transmission and reception of data. For this reason, a USB-FS core centric nomenclature is used to describe the direction of the data transfer between the USB-FS core and the USB:

- RX (or receive) describes transfers that move data from the USB to memory.
- TX (or transmit) describes transfers that move data from memory to the USB.

The following table shows how the data direction corresponds to the USB token type in host and target device applications.

Table 40-1. Data direction for USB host or USB target

	RX	TX
Device	OUT or Setup	IN
Host	IN	OUT or SETUP

40.3.3 Addressing BDT entries

An understanding of the addressing mechanism of the Buffer Descriptor Table is useful when accessing endpoint data via the USB-FS or microprocessor. Some points of interest are:

- The BDT occupies up to 512 bytes of system memory.
- 16 bidirectional endpoints can be supported with a full BDT of 512 bytes.
- 16 bytes are needed for each USB endpoint direction.
- Applications with less than 16 endpoints require less RAM to implement the BDT.
- The BDT Page Registers (BDT_PAGE) point to the starting location of the BDT.
- The BDT must be located on a 512-byte boundary in system memory.
- All enabled TX and RX endpoint BD entries are indexed into the BDT to allow easy access via the USB-FS or MCU core.

When a USB token on an enabled endpoint is received, the USB-FS uses its integrated DMA controller to interrogate the BDT. The USB-FS reads the corresponding endpoint BD entry to determine whether it owns the BD and corresponding buffer in system memory.

To compute the entry point in to the BDT, the BDT_PAGE registers is concatenated with the current endpoint and the TX and ODD fields to form a 32-bit address. This address mechanism is shown below:

Table 40-2. BDT address calculation fields

Field	Description
BDT_PAGE	BDT_PAGE registers in the Control Register Block
END_POINT	END POINT field from the USB TOKEN
TX	1 for transmit transfers and 0 for receive transfers
ODD	Maintained within the USB-FS SIE. It corresponds to the buffer currently in use. The buffers are used in a ping-pong fashion.

40.3.4 Buffer Descriptors (BDs)

A buffer descriptor provides endpoint buffer control information for the USB-FS and processor. The Buffer Descriptors have different meaning based on whether it is the USB-FS or processor reading the BD in memory.

The USB-FS Controller uses the data stored in the BDs to determine:

- Who owns the buffer in system memory
- Data0 or Data1 PID
- Whether to release ownership upon packet completion
- No address increment (FIFO mode)
- Whether data toggle synchronization is enabled
- How much data is to be transmitted or received
- Where the buffer resides in system memory

While the processor uses the data stored in the BDs to determine:

- Who owns the buffer in system memory
- Data0 or Data1 PID
- The received TOKEN PID
- How much data was transmitted or received
- Where the buffer resides in system memory

The format for the BD is shown in the following figure.

Table 40-3. Buffer descriptor format

31:26	25:16	15:8	7	6	5	4	3	2	1	0
RSVD	BC (10 bits)	RSVD	OWN	DATA0/1	KEEP/ TOK_PID[3]	NINC/ TOK_PID[2]	DTS/ TOK_PID[1]	BDT_STALL/ TOK_PID[0]	0	0
Buffer Address (32-Bits)										

Table 40-4. Buffer descriptor fields

Field	Description
31–26 RSVD	Reserved
25–16 BC	Byte Count Represents the 10-bit byte count. The USB-FS SIE changes this field upon the completion of a RX transfer with the byte count of the data received.
15–8 RSVD	Reserved

Table continues on the next page...

Table 40-4. Buffer descriptor fields (continued)

Field	Description
7 OWN	Determines whether the processor or the USB-FS currently owns the buffer. Except when KEEP=1, the SIE writes a 0 to this bit when it has completed a token. This must always be the last byte of the BD that the processor updates when it initializes a BD. 0 The processor has exclusive access to the BD. The USB-FS ignores all other fields in the BD. 1 USB-FS has exclusive access to the BD. After the BD has been assigned to the USB-FS, the processor should not change it in any way.
6 DATA0/1	Defines whether a DATA0 field (DATA0/1=0) or a DATA1 (DATA0/1=1) field was transmitted or received. It is unchanged by the USB-FS.
5 KEEP/ TOK_PID[3]	Typically, this bit is 1 with ISO endpoints feeding a FIFO. The microprocessor is not informed that a token has been processed, the data is simply transferred to or from the FIFO. When KEEP is set, normally the NINC bit is also set to prevent address increment. 0 Bit 3 of the current token PID is written back to the BD by the USB-FS. Allows the USB-FS to release the BD when a token has been processed. 1 This bit is unchanged by the USB-FS. If the OWN bit also is set, the BD remains owned by the USB-FS forever.
4 NINC/ TOK_PID[2]	No Increment (NINC) Disables the DMA engine address increment. This forces the DMA engine to read or write from the same address. This is useful for endpoints when data needs to be read from or written to a single location such as a FIFO. Typically this bit is set with the KEEP bit for ISO endpoints that are interfacing to a FIFO. 0 The USB-FS writes bit 2 of the current token PID to the BD. 1 This bit is unchanged by the USB-FS.
3 DTS/ TOK_PID[1]	Setting this bit enables the USB-FS to perform Data Toggle Synchronization. • If KEEP=0, bit 1 of the current token PID is written back to the BD. • If KEEP=1, this bit is unchanged by the USB-FS. 0 Data Toggle Synchronization is disabled. 1 Enables the USB-FS to perform Data Toggle Synchronization.
2 BDT_STALL TOK_PID[0]	Setting this bit causes the USB-FS to issue a STALL handshake if a token is received by the SIE that would use the BDT in this location. The BDT is not consumed by the SIE (the owns bit remains set and the rest of the BDT is unchanged) when a BDT-STALL bit is set. • If KEEP=0, bit 0 of the current token PID is written back to the BD. • If KEEP=1, this bit is unchanged by the USB-FS. 0 No stall issued. 1 The BDT is not consumed by the SIE (the OWN bit remains set and the rest of the BDT is unchanged).

Table continues on the next page...

Table 40-4. Buffer descriptor fields (continued)

Field	Description
TOK_PID[n]	Bits [5:2] can also represent the current token PID. The current token PID is written back in to the BD by the USB-FS when a transfer completes. The values written back are the token PID values from the USB specification: • 0x1 for an OUT token. • 0x9 for an IN token. • 0xd for a SETUP token. In host mode, this field is used to report the last returned PID or a transfer status indication. The possible values returned are: • 0x3 DATA0 • 0xb DATA1 • 0x2 ACK • 0xe STALL • 0xa NAK • 0x0 Bus Timeout • 0xf Data Error
1–0 Reserved	Reserved, should read as zeroes.
ADDR[31:0]	Address Represents the 32-bit buffer address in system memory. These bits are unchanged by the USB-FS.

40.3.5 USB transaction

When the USB-FS transmits or receives data, it computes the BDT address using the address generation shown in "Addressing Buffer Descriptor Entries" table.

If OWN =1, the following process occurs:

1. The USB-FS reads the BDT.
2. The SIE transfers the data via the DMA to or from the buffer pointed to by the ADDR field of the BD.
3. When the TOKEN is complete, the USB-FS updates the BDT and, if KEEP=0, changes the OWN bit to 0.
4. The STAT register is updated and the TOK_DNE interrupt is set.
5. When the processor processes the TOK_DNE interrupt, it reads from the status register all the information needed to process the endpoint.
6. At this point, the processor allocates a new BD so that additional USB data can be transmitted or received for that endpoint, and then processes the last BD.

The following figure shows a timeline of how a typical USB token is processed after the BDT is read and OWN=1.

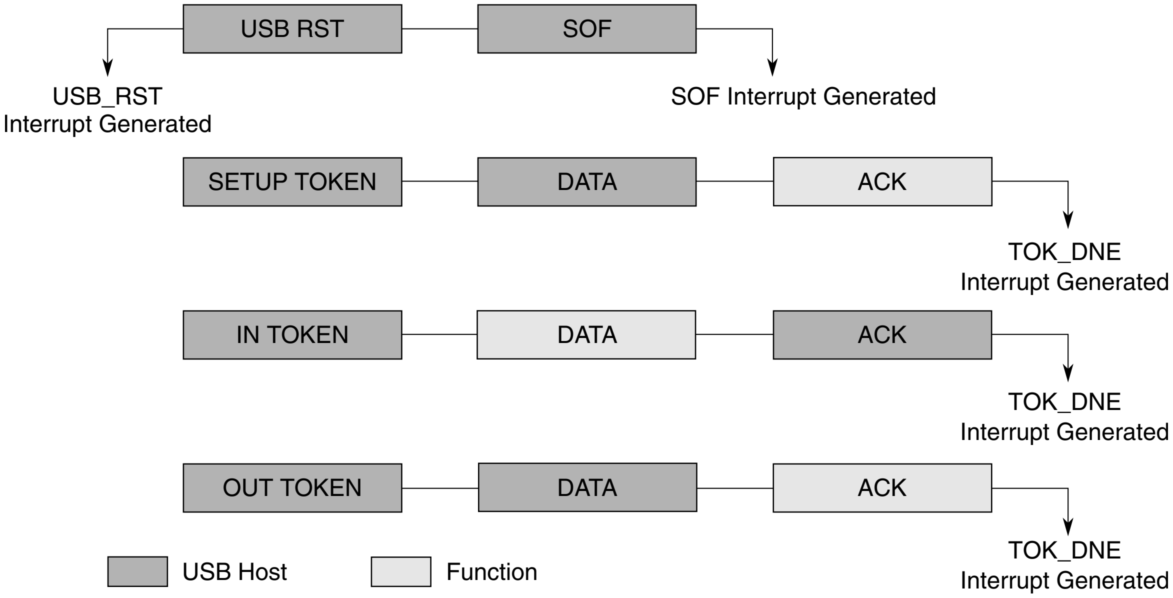

USB RST
SOF
USB_RST
Interrupt Generated
SOF Interrupt Generated
SETUP TOKEN
DATA
ACK
TOK_DNE
Interrupt Generated
IN TOKEN
DATA
ACK
TOK_DNE
Interrupt Generated
OUT TOKEN
DATA
ACK
TOK_DNE
Interrupt Generated
USB Host
Function

Figure 40-4. USB token transaction

The USB has two sources for the DMA overrun error:

Memory Latency

The memory latency may be too high and cause the receive FIFO to overflow. This is predominantly a hardware performance issue, usually caused by transient memory access issues.

Oversized Packets

The packet received may be larger than the negotiated *MaxPacket* size. Typically, this is caused by a software bug. For DMA overrun errors due to oversized data packets, the USB specification is ambiguous. It assumes correct software drivers on both sides. NAKing the packet can result in retransmission of the already oversized packet data. Therefore, in response to oversized packets, the USB core continues ACKing the packet for non-isochronous transfers.

Table 40-5. USB responses to DMA overrun errors

Errors due to Memory Latency	Errors due to Oversized Packets
Non-Acknowledgment (NAK) or Bus Timeout (BTO) — See bit 4 in "Error Interrupt Status Register (ERRSTAT)" as appropriate for the class of transaction.	Continues acknowledging (ACKing) the packet for non-isochronous transfers.
—	The data written to memory is clipped to the MaxPacket size so as not to corrupt system memory.
The DMAERR bit is set in the ERRSTAT register for host and device modes of operation. Depending on the values of the INTENB and ERRENB register, the core may assert an interrupt to notify the processor of the DMA error.	Asserts ERRSTAT[DMAERR] ,which can trigger an interrupt and TOKDNE interrupt fires. Note: The TOK_PID field of the BDT is not 1111 because the DMAERR is not due to latency.

Table continues on the next page...

Table 40-5. USB responses to DMA overrun errors (continued)

Errors due to Memory Latency	Errors due to Oversized Packets
- For host mode, the TOKDNE interrupt is generated and the TOK_PID field of the BDT is 1111 to indicate the DMA latency error. Host mode software can decide to retry or move to next scheduled item. - In device mode, the BDT is not written back nor is the TOKDNE interrupt triggered because it is assumed that a second attempt is queued and will succeed in the future.	The packet length field written back to the BDT is the MaxPacket value that represents the length of the clipped data actually written to memory.
From here, the software can decide an appropriate course of action for future transactions such as stalling the endpoint, canceling the transfer, disabling the endpoint, etc.	

40.4 Memory map/Register definitions

This section provides the memory map and detailed descriptions of all USB interface registers.

USB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_2000	Peripheral ID register (USB0_PERID)	8	R	04h	40.4.1/ 895
4007_2004	Peripheral ID Complement register (USB0_IDCOMP)	8	R	FBh	40.4.2/ 896
4007_2008	Peripheral Revision register (USB0_REV)	8	R	33h	40.4.3/ 896
4007_200C	Peripheral Additional Info register (USB0_ADDINFO)	8	R	01h	40.4.4/ 897
4007_2010	OTG Interrupt Status register (USB0_OTGISTAT)	8	R/W	00h	40.4.5/ 897
4007_2014	OTG Interrupt Control Register (USB0_OTGICR)	8	R/W	00h	40.4.6/ 898
4007_2018	OTG Status register (USB0_OTGSTAT)	8	R/W	00h	40.4.7/ 899
4007_201C	OTG Control Register (USB0_OTGCTL)	8	R/W	00h	40.4.8/ 900
4007_2080	Interrupt Status Register (USB0_ISTAT)	8	R/W	00h	40.4.9/ 901
4007_2084	Interrupt Enable Register (USB0_INTEN)	8	R/W	00h	40.4.10/ 902

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_2088	Error Interrupt Status Register (USB0_ERRSTAT)	8	R/W	00h	40.4.11/ 903
4007_208C	Error Interrupt Enable Register (USB0_ERREN)	8	R/W	00h	40.4.12/ 904
4007_2090	Status Register (USB0_STAT)	8	R	00h	40.4.13/ 906
4007_2094	Control Register (USB0_CTL)	8	R/W	00h	40.4.14/ 907
4007_2098	Address Register (USB0_ADDR)	8	R/W	00h	40.4.15/ 908
4007_209C	BDT Page Register 1 (USB0_BDTPAGE1)	8	R/W	00h	40.4.16/ 909
4007_20A0	Frame Number Register Low (USB0_FRMNUML)	8	R/W	00h	40.4.17/ 909
4007_20A4	Frame Number Register High (USB0_FRMNUMH)	8	R/W	00h	40.4.18/ 910
4007_20A8	Token Register (USB0_TOKEN)	8	R/W	00h	40.4.19/ 910
4007_20AC	SOF Threshold Register (USB0_SOFTHLD)	8	R/W	00h	40.4.20/ 911
4007_20B0	BDT Page Register 2 (USB0_BDTPAGE2)	8	R/W	00h	40.4.21/ 912
4007_20B4	BDT Page Register 3 (USB0_BDTPAGE3)	8	R/W	00h	40.4.22/ 912
4007_20C0	Endpoint Control Register (USB0_ENDPT0)	8	R/W	00h	40.4.23/ 912
4007_20C4	Endpoint Control Register (USB0_ENDPT1)	8	R/W	00h	40.4.23/ 912
4007_20C8	Endpoint Control Register (USB0_ENDPT2)	8	R/W	00h	40.4.23/ 912
4007_20CC	Endpoint Control Register (USB0_ENDPT3)	8	R/W	00h	40.4.23/ 912
4007_20D0	Endpoint Control Register (USB0_ENDPT4)	8	R/W	00h	40.4.23/ 912
4007_20D4	Endpoint Control Register (USB0_ENDPT5)	8	R/W	00h	40.4.23/ 912
4007_20D8	Endpoint Control Register (USB0_ENDPT6)	8	R/W	00h	40.4.23/ 912
4007_20DC	Endpoint Control Register (USB0_ENDPT7)	8	R/W	00h	40.4.23/ 912

Table continues on the next page...

USB memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4007_20E0	Endpoint Control Register (USB0_ENDPT8)	8	R/W	00h	40.4.23/ 912
4007_20E4	Endpoint Control Register (USB0_ENDPT9)	8	R/W	00h	40.4.23/ 912
4007_20E8	Endpoint Control Register (USB0_ENDPT10)	8	R/W	00h	40.4.23/ 912
4007_20EC	Endpoint Control Register (USB0_ENDPT11)	8	R/W	00h	40.4.23/ 912
4007_20F0	Endpoint Control Register (USB0_ENDPT12)	8	R/W	00h	40.4.23/ 912
4007_20F4	Endpoint Control Register (USB0_ENDPT13)	8	R/W	00h	40.4.23/ 912
4007_20F8	Endpoint Control Register (USB0_ENDPT14)	8	R/W	00h	40.4.23/ 912
4007_20FC	Endpoint Control Register (USB0_ENDPT15)	8	R/W	00h	40.4.23/ 912
4007_2100	USB Control Register (USB0_USBCTRL)	8	R/W	C0h	40.4.24/ 913
4007_2104	USB OTG Observe Register (USB0_OBSERVE)	8	R	50h	40.4.25/ 914
4007_2108	USB OTG Control Register (USB0_CONTROL)	8	R/W	00h	40.4.26/ 915
4007_210C	USB Transceiver Control Register 0 (USB0_USBTRC0)	8	R/W	00h	40.4.27/ 915
4007_2114	Frame Adjust Register (USB0_USBFRMADJUST)	8	R/W	00h	40.4.28/ 916

40.4.1 Peripheral ID register (USBx_PERID)

Reads back the value of 0x04. This value is defined for the USB peripheral.

Addresses: USB0_PERID is 4007_2000h base + 0h offset = 4007_2000h

Bit	7	6	5	4	3	2	1	0
Read	0		ID					
Write								
Reset	0	0	0	0	0	1	0	0

USBx_PERID field descriptions

Field	Description
7–6 Reserved	This read-only field is reserved and always has the value zero.
5–0 ID	Peripheral Identification This field always reads 0x4h.

40.4.2 Peripheral ID Complement register (USBx_IDCOMP)

Reads back the complement of the Peripheral ID register. For the USB peripheral, the value is 0xFB.

Addresses: USB0_IDCOMP is 4007_2000h base + 4h offset = 4007_2004h

Bit	7	6	5	4	3	2	1	0
Read	1		NID					
Write								
Reset	1	1	1	1	1	0	1	1

USBx_IDCOMP field descriptions

Field	Description
7–6 Reserved	This read-only field is reserved and always has the value one.
5–0 NID	Ones complement of peripheral identification bits.

40.4.3 Peripheral Revision register (USBx_REV)

Contains the revision number of the USB module.

Addresses: USB0_REV is 4007_2000h base + 8h offset = 4007_2008h

Bit	7	6	5	4	3	2	1	0
Read	REV							
Write								
Reset	0	0	1	1	0	0	1	1

USBx_REV field descriptions

Field	Description
7–0 REV	Revision Indicate the revision number of the USB Core.

40.4.4 Peripheral Additional Info register (USBx_ADDINFO)

Reads back the value of the fixed Interrupt Request Level (IRQNUM) along with the Host Enable bit.

Addresses: USB0_ADDINFO is 4007_2000h base + Ch offset = 4007_200Ch

Bit	7	6	5	4	3	2	1	0
Read	IRQNUM					0		IEHOST
Write								
Reset	0	0	0	0	0	0	0	1

USBx_ADDINFO field descriptions

Field	Description
7–3 IRQNUM	Assigned Interrupt Request Number
2–1 Reserved	This read-only field is reserved and always has the value zero.
0 IEHOST	When this bit is set, the USB peripheral is operating in host mode.

40.4.5 OTG Interrupt Status register (USBx_OTGISTAT)

Records changes of the ID sense and VBUS signals. Software can read this register to determine the event that triggers interrupt. Only bits that have changed since the last software read are set. Writing a one to a bit clears the associated interrupt.

Addresses: USB0_OTGISTAT is 4007_2000h base + 10h offset = 4007_2010h

Bit	7	6	5	4	3	2	1	0
Read	IDCHG	ONEMSEC	LINE_ STATE_ CHG	0	SESSVLDCHG	B_SESS_CHG	0	AVBUSCHG
Write								
Reset	0	0	0	0	0	0	0	0

USBx_OTGISTAT field descriptions

Field	Description
7 IDCHG	This bit is set when a change in the ID Signal from the USB connector is sensed.
6 ONEMSEC	This bit is set when the 1 millisecond timer expires. This bit stays asserted until cleared by software. The interrupt must be serviced every millisecond to avoid losing 1msec counts.
5 LINE_STATE_ CHG	This bit is set when the USB line state changes. The interrupt associated with this bit can be used to detect Reset, Resume, Connect, and Data Line Pulse signaling

Table continues on the next page...

USBx_OTGISTAT field descriptions (continued)

Field	Description
4 Reserved	This read-only field is reserved and always has the value zero.
3 SESSVLDCHG	This bit is set when a change in VBUS is detected indicating a session valid or a session no longer valid.
2 B_SESS_CHG	This bit is set when a change in VBUS is detected on a B device.
1 Reserved	This read-only field is reserved and always has the value zero.
0 AVBUSCHG	This bit is set when a change in VBUS is detected on an A device.

40.4.6 OTG Interrupt Control Register (USBx_OTGICR)

Enables the corresponding interrupt status bits defined in the OTG Interrupt Status Register.

Addresses: USB0_OTGICR is 4007_2000h base + 14h offset = 4007_2014h

Bit	7	6	5	4	3	2	1	0
Read	IDEN	ONEMSECEN	LINESTATEEN	0	SESSVLDEN	BSESSEN	0	AVBUSEN
Write				▨			▨	
Reset	0	0	0	0	0	0	0	0

USBx_OTGICR field descriptions

Field	Description
7 IDEN	ID Interrupt Enable 0 The ID interrupt is disabled 1 The ID interrupt is enabled
6 ONEMSECEN	One Millisecond Interrupt Enable 0 Diables the 1ms timer interrupt. 1 Enables the 1ms timer interrupt.
5 LINESTATEEN	Line State Change Interrupt Enable 0 Disables the LINE_STAT_CHG interrupt. 1 Enables the LINE_STAT_CHG interrupt.
4 Reserved	This read-only field is reserved and always has the value zero.
3 SESSVLDEN	Session Valid Interrupt Enable 0 Disables the SESSVLDCHG interrupt. 1 Enables the SESSVLDCHG interrupt.

Table continues on the next page...

USBx_OTGICR field descriptions (continued)

Field	Description
2 BSESSEN	B Session END Interrupt Enable 0 Disables the B_SESS_CHG interrupt. 1 Enables the B_SESS_CHG interrupt.
1 Reserved	This read-only field is reserved and always has the value zero.
0 AVBUSEN	A VBUS Valid Interrupt Enable 0 Disables the AVBUSCHG interrupt. 1 Enables the AVBUSCHG interrupt.

40.4.7 OTG Status register (USBx_OTGSTAT)

Displays the actual value from the external comparator outputs of the ID pin and VBUS.

Addresses: USB0_OTGSTAT is 4007_2000h base + 18h offset = 4007_2018h

Bit	7	6	5	4
Read	ID	ONEMSECEN	LINESTATESTABLE	0
Write				▨
Reset	0	0	0	0

Bit	3	2	1	0
Read	SESS_VLD	BSESSEND	0	AVBUSVLD
Write			▨	
Reset	0	0	0	0

USBx_OTGSTAT field descriptions

Field	Description
7 ID	Indicates the current state of the ID pin on the USB connector 0 Indicates a Type A cable is plugged into the USB connector. 1 Indicates no cable is attached or a Type B cable is plugged into the USB connector.
6 ONEMSECEN	This bit is reserved for the 1ms count, but it is not useful to software.
5 LINESTATESTABLE	Indicates that the internal signals that control the LINE_STATE_CHG field of OTGISTAT are stable for at least 1 millisecond. First read LINE_STATE_CHG field and then read this field. If this field reads as 1, then the value of LINE_STATE_CHG can be considered stable. 0 The LINE_STAT_CHG bit is not yet stable. 1 The LINE_STAT_CHG bit has been debounced and is stable.
4 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

USBx_OTGSTAT field descriptions (continued)

Field	Description
3 SESS_VLD	Session Valid 0 The VBUS voltage is below the B session valid threshold 1 The VBUS voltage is above the B session valid threshold.
2 BSESSEND	B Session End 0 The VBUS voltage is above the B session end threshold. 1 The VBUS voltage is below the B session end threshold.
1 Reserved	This read-only field is reserved and always has the value zero.
0 AVBUSVLD	A VBUS Valid 0 The VBUS voltage is below the A VBUS Valid threshold. 1 The VBUS voltage is above the A VBUS Valid threshold.

40.4.8 OTG Control Register (USBx_OTGCTL)

The OTG Control Register controls the operation of VBUS and Data Line termination resistors.

Addresses: USB0_OTGCTL is 4007_2000h base + 1Ch offset = 4007_201Ch

Bit	7	6	5	4	3	2	1	0
Read	DPHIGH	0	DPLOW	DMLOW	0	OTGEN	0	
Write		▨			▨		▨	▨
Reset	0	0	0	0	0	0	0	0

USBx_OTGCTL field descriptions

Field	Description
7 DPHIGH	D+ Data Line pullup resistor enable 0 D+ pullup resistor is not enabled 1 D+ pullup resistor is enabled
6 Reserved	This read-only field is reserved and always has the value zero.
5 DPLOW	D+ Data Line pull-down resistor enable This bit should always be enabled together with bit 4 (DMLOW) 0 D+ pulldown resistor is not enabled. 1 D+ pulldown resistor is enabled.
4 DMLOW	D- Data Line pull-down resistor enable

Table continues on the next page...

USBx_OTGCTL field descriptions (continued)

Field	Description
	0 D- pulldown resistor is not enabled. 1 D- pulldown resistor is enabled.
3 Reserved	This read-only field is reserved and always has the value zero.
2 OTGEN	On-The-Go pullup/pulldown resistor enable 0 If USB_EN is set and HOST_MODE is clear in the Control Register (CTL), then the D+ Data Line pull-up resistors are enabled. If HOST_MODE is set the D+ and D- Data Line pull-down resistors are engaged. 1 The pull-up and pull-down controls in this register are used.
1–0 Reserved	This read-only field is reserved and always has the value zero.

40.4.9 Interrupt Status Register (USBx_ISTAT)

The Interrupt Status Register contains bits for each of the interrupt sources within the USB Module. Each of these bits are qualified with their respective interrupt enable bits. All bits of this register are logically OR'd together along with the OTG Interrupt Status Register (OTGSTAT) to form a single interrupt source for the processor's interrupt controller. After an interrupt bit has been set it may only be cleared by writing a one to the respective interrupt bit. This register contains the value of 0x00 after a reset.

Addresses: USB0_ISTAT is 4007_2000h base + 80h offset = 4007_2080h

Bit	7	6	5	4	3	2	1	0
Read	STALL	ATTACH	RESUME	SLEEP	TOKDNE	SOFTOK	ERROR	USBRST
Write	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

USBx_ISTAT field descriptions

Field	Description
7 STALL	Stall Interrupt In Target mode this bit is asserted when a STALL handshake is sent by the SIE. In Host mode this bit is set when the USB Module detects a STALL acknowledge during the handshake phase of a USB transaction. This interrupt can be use to determine is the last USB transaction was completed successfully or if it stalled.
6 ATTACH	Attach Interrupt This bit is set when the USB Module detects an attach of a USB device. This signal is only valid if HOSTMODEEN is true. This interrupt signifies that a peripheral is now present and must be configured.
5 RESUME	This bit is set depending upon the DP/DM signals, and can be used to signal remote wake-up signaling on the USB bus. When not in suspend mode this interrupt should be disabled.

Table continues on the next page...

USBx_ISTAT field descriptions (continued)

Field	Description
4 SLEEP	This bit is set when the USB Module detects a constant idle on the USB bus for 3 milliseconds. The sleep timer is reset by activity on the USB bus.
3 TOKDNE	This bit is set when the current token being processed has completed. The processor should immediately read the STAT register to determine the EndPoint and BD used for this token. Clearing this bit (by writing a one) causes the STAT register to be cleared or the STAT holding register to be loaded into the STAT register.
2 SOFTOK	This bit is set when the USB Module receives a Start Of Frame (SOF) token. In Host mode this bit is set when the SOF threshold is reached, so that software can prepare for the next SOF.
1 ERROR	This bit is set when any of the error conditions within the ERRSTAT register occur. The processor must then read the ERRSTAT register to determine the source of the error.
0 USBRST	This bit is set when the USB Module has decoded a valid USB reset. This informs the Microprocessor that it should write 0x00 into the address register and enable endpoint 0. USBRST is set after a USB reset has been detected for 2.5 microseconds. It is not asserted again until the USB reset condition has been removed and then reasserted.

40.4.10 Interrupt Enable Register (USBx_INTEN)

The Interrupt Enable Register contains enable bits for each of the interrupt sources within the USB Module. Setting any of these bits enables the respective interrupt source in the ISTAT register. This register contains the value of 0x00 after a reset.

Addresses: USB0_INTEN is 4007_2000h base + 84h offset = 4007_2084h

Bit	7	6	5	4	3	2	1	0
Read Write	STALLEN	ATTACHEN	RESUMEEN	SLEEPEN	TOKDNEEN	SOFTOKEN	ERROREN	USBRSTEN
Reset	0	0	0	0	0	0	0	0

USBx_INTEN field descriptions

Field	Description
7 STALLEN	STALL Interrupt Enable 0 The STALL interrupt is not enabled. 1 The STALL interrupt is enabled.
6 ATTACHEN	ATTACH Interrupt Enable 0 The ATTACH interrupt is not enabled. 1 The ATTACH interrupt is enabled.
5 RESUMEEN	RESUME Interrupt Enable 0 The RESUME interrupt is not enabled. 1 The RESUME interrupt is enabled.

Table continues on the next page...

USBx_INTEN field descriptions (continued)

Field	Description
4 SLEEPEN	SLEEP Interrupt Enable 0 The SLEEP interrupt is not enabled. 1 The SLEEP interrupt is enabled.
3 TOKDNEEN	TOKDNE Interrupt Enable 0 The TOKDNE interrupt is not enabled. 1 The TOKDNE interrupt is enabled.
2 SOFTOKEN	SOFTOK Interrupt Enable 0 The SOFTOK interrupt is not enabled. 1 The SOFTOK interrupt is enabled.
1 ERROREN	ERROR Interrupt Enable 0 The ERROR interrupt is not enabled. 1 The ERROR interrupt is enabled.
0 USBRSTEN	USBRST Interrupt Enable 0 The USBRST interrupt is not enabled. 1 The USBRST interrupt is enabled.

40.4.11 Error Interrupt Status Register (USBx_ERRSTAT)

The Error Interrupt Status Register contains enable bits for each of the error sources within the USB Module. Each of these bits are qualified with their respective error enable bits. All bits of this Register are logically OR'd together and the result placed in the ERROR bit of the ISTAT register. After an interrupt bit has been set it may only be cleared by writing a one to the respective interrupt bit. Each bit is set as soon as the error conditions is detected. Therefore, the interrupt does not typically correspond with the end of a token being processed. This register contains the value of 0x00 after a reset.

Addresses: USB0_ERRSTAT is 4007_2000h base + 88h offset = 4007_2088h

Bit	7	6	5	4	3	2	1	0
Read	BTSERR	0	DMAERR	BTOERR	DFN8	CRC16	CRC5EOF	PIDERR
Write	w1c		w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0

USBx_ERRSTAT field descriptions

Field	Description
7 BTSERR	This bit is set when a bit stuff error is detected. If set, the corresponding packet is rejected due to the error.

Table continues on the next page...

USBx_ERRSTAT field descriptions (continued)

Field	Description
6 Reserved	This read-only field is reserved and always has the value zero.
5 DMAERR	This bit is set if the USB Module has requested a DMA access to read a new BDT but has not been given the bus before it needs to receive or transmit data. If processing a TX transfer this would cause a transmit data underflow condition. If processing a RX transfer this would cause a receive data overflow condition. This interrupt is useful when developing device arbitration hardware for the microprocessor and the USB Module to minimize bus request and bus grant latency. This bit is also set if a data packet to or from the host is larger than the buffer size allocated in the BDT. In this case the data packet is truncated as it is put into buffer memory.
4 BTOERR	This bit is set when a bus turnaround timeout error occurs. The USB Module contains a bus turnaround timer that keeps track of the amount of time elapsed between the token and data phases of a SETUP or OUT TOKEN or the data and handshake phases of a IN TOKEN. If more than 16 bit times are counted from the previous EOP before a transition from IDLE, a bus turnaround timeout error occurs.
3 DFN8	This bit is set if the data field received was not 8 bits in length. USB Specification 1.0 requires that data fields be an integral number of bytes. If the data field was not an integral number of bytes, this bit is set.
2 CRC16	This bit is set when a data packet is rejected due to a CRC16 error.
1 CRC5EOF	This error interrupt has two functions. When the USB Module is operating in peripheral mode (HOSTMODEEN=0), this interrupt detects CRC5 errors in the token packets generated by the host. If set the token packet was rejected due to a CRC5 error. When the USB Module is operating in host mode (HOSTMODEEN=1), this interrupt detects End Of Frame (EOF) error conditions. This occurs when the USB Module is transmitting or receiving data and the SOF counter reaches zero. This interrupt is useful when developing USB packet scheduling software to ensure that no USB transactions cross the start of the next frame.
0 PIDERR	This bit is set when the PID check field fails.

40.4.12 Error Interrupt Enable Register (USBx_ERREN)

The Error Interrupt Enable Register contains enable bits for each of the error interrupt sources within the USB Module. Setting any of these bits enables the respective interrupt source in the ERRSTAT register. Each bit is set as soon as the error conditions is detected. Therefore, the interrupt does not typically correspond with the end of a token being processed. This register contains the value of 0x00 after a reset.

Addresses: USB0_ERREN is 4007_2000h base + 8Ch offset = 4007_208Ch

Bit	7	6	5	4	3	2	1	0
Read	BTSERREN	0	DMAERREN	BTOERREN	DFN8EN	CRC16EN	CRC5EOFEN	PIDERREN
Write		▨						
Reset	0	0	0	0	0	0	0	0

USBx_ERREN field descriptions

Field	Description
7 BTSERREN	BTSERR Interrupt Enable 0 The BTSERR interrupt is not enabled. 1 The BTSERR interrupt is enabled.
6 Reserved	This read-only field is reserved and always has the value zero.
5 DMAERREN	DMAERR Interrupt Enable 0 The DMAERR interrupt is not enabled. 1 The DMAERR interrupt is enabled.
4 BTOERREN	BTOERR Interrupt Enable 0 The BTOERR interrupt is not enabled. 1 The BTOERR interrupt is enabled.
3 DFN8EN	DFN8 Interrupt Enable 0 The DFN8 interrupt is not enabled. 1 The DFN8 interrupt is enabled.
2 CRC16EN	CRC16 Interrupt Enable 0 The CRC16 interrupt is not enabled. 1 The CRC16 interrupt is enabled.
1 CRC5EOFEN	CRC5/EOF Interrupt Enable 0 The CRC5/EOF interrupt is not enabled. 1 The CRC5/EOF interrupt is enabled.
0 PIDERREN	PIDERR Interrupt Enable 0 The PIDERR interrupt is not enabled. 1 The PIDERR interrupt is enabled.

40.4.13 Status Register (USBx_STAT)

The Status Register reports the transaction status within the USB Module. When the processor's interrupt controller has received a TOKDNE interrupt the Status Register should be read to determine the status of the previous endpoint communication. The data in the status register is valid when the TOKDNE interrupt bit is asserted. The STAT register is actually a read window into a status FIFO maintained by the USB Module. When the USB Module uses a BD, it updates the Status Register. If another USB transaction is performed before the TOKDNE interrupt is serviced, the USB Module stores the status of the next transaction in the STAT FIFO. Thus the STAT register is actually a four byte FIFO that allows the processor core to process one transaction while the SIE is processing the next transaction. Clearing the TOKDNE bit in the ISTAT register causes the SIE to update the STAT register with the contents of the next STAT value. If the data in the STAT holding register is valid, the SIE immediately reasserts to TOKDNE interrupt.

Addresses: USB0_STAT is 4007_2000h base + 90h offset = 4007_2090h

Bit	7	6	5	4	3	2	1	0
Read	ENDP				TX	ODD	0	
Write								
Reset	0	0	0	0	0	0	0	0

USBx_STAT field descriptions

Field	Description
7–4 ENDP	This four-bit field encodes the endpoint address that received or transmitted the previous token. This allows the processor core to determine which BDT entry was updated by the last USB transaction.
3 TX	Transmit Indicator 0 The most recent transaction was a Receive operation. 1 The most recent transaction was a Transmit operation.
2 ODD	this bit is set if the last Buffer Descriptor updated was in the odd bank of the BDT.
1–0 Reserved	This read-only field is reserved and always has the value zero.

40.4.14 Control Register (USBx_CTL)

The Control Register provides various control and configuration information for the USB Module.

Addresses: USB0_CTL is 4007_2000h base + 94h offset = 4007_2094h

Bit	7	6	5	4
Read Write	JSTATE	SE0	TXSUSPENDTOKENBUSY	RESET
Reset	0	0	0	0

Bit	3	2	1	0
Read Write	HOSTMODEEN	RESUME	ODDRST	USBENSOFEN
Reset	0	0	0	0

USBx_CTL field descriptions

Field	Description
7 JSTATE	Live USB differential receiver JSTATE signal The polarity of this signal is affected by the current state of LSEN .
6 SE0	Live USB Single Ended Zero signal
5 TXSUSPENDTOKENBUSY	When the USB Module is in Host mode TOKEN_BUSY is set when the USB Module is busy executing a USB token and no more token commands should be written to the Token Register. Software should check this bit before writing any tokens to the Token Register to ensure that token commands are not lost. In Target mode TXD_SUSPEND is set when the SIE has disabled packet transmission and reception. Clearing this bit allows the SIE to continue token processing. This bit is set by the SIE when a Setup Token is received allowing software to dequeue any pending packet transactions in the BDT before resuming token processing.
4 RESET	Setting this bit enables the USB Module to generate USB reset signaling. This allows the USB Module to reset USB peripherals. This control signal is only valid in Host mode (HOSTMODEEN=1). Software must set RESET to 1 for the required amount of time and then clear it to 0 to end reset signaling. For more information on RESET signaling see Section 7.1.4.3 of the USB specification version 1.0.
3 HOSTMODEEN	When set to 1, this bit enables the USB Module to operate in Host mode. In host mode, the USB module performs USB transactions under the programmed control of the host processor.
2 RESUME	When set to 1 this bit enables the USB Module to execute resume signaling. This allows the USB Module to perform remote wake-up. Software must set RESUME to 1 for the required amount of time and then clear it to 0. If the HOSTMODEEN bit is set, the USB module appends a Low Speed End of Packet to the Resume signaling when the RESUME bit is cleared. For more information on RESUME signaling see Section 7.1.4.5 of the USB specification version 1.0.
1 ODDRST	Setting this bit to 1 resets all the BDT ODD ping/pong bits to 0, which then specifies the EVEN BDT bank.
0 USBENSOFEN	USB Enable

Table continues on the next page...

USBx_CTL field descriptions (continued)

Field	Description
	Setting this bit causes the SIE to reset all of its ODD bits to the BDTs. Therefore, setting this bit resets much of the logic in the SIE. When host mode is enabled, clearing this bit causes the SIE to stop sending SOF tokens. 0 The USB Module is disabled. 1 The USB Module is enabled.

40.4.15 Address Register (USBx_ADDR)

The Address Register holds the unique USB address that the USB Module decodes when in Peripheral mode (HOSTMODEEN=0). When operating in Host mode (HOSTMODEEN=1) the USB Module transmits this address with a TOKEN packet. This enables the USB Module to uniquely address an USB peripheral. In either mode, the USB_EN bit within the control register must be set. The Address Register is reset to 0x00 after the reset input becomes active or the USB Module decodes a USB reset signal. This action initializes the Address Register to decode address 0x00 as required by the USB specification.

Addresses: USB0_ADDR is 4007_2000h base + 98h offset = 4007_2098h

Bit	7	6	5	4	3	2	1	0
Read Write	LSEN	ADDR						
Reset	0	0	0	0	0	0	0	0

USBx_ADDR field descriptions

Field	Description
7 LSEN	Low Speed Enable bit This bit informs the USB Module that the next token command written to the token register must be performed at low speed. This enables the USB Module to perform the necessary preamble required for low-speed data transmissions.
6–0 ADDR	USB address This 7-bit value defines the USB address that the USB Module decodes in peripheral mode, or transmit when in host mode.

40.4.16 BDT Page Register 1 (USBx_BDTPAGE1)

The Buffer Descriptor Table Page Register 1 provides address bits 15 through 9 of the base address where the current Buffer Descriptor Table (BDT) resides in system memory. The 32-bit BDT Base Address is always aligned on 512-byte boundaries, so bits 8 through 0 of the base address are always taken as zero.

Addresses: USB0_BDTPAGE1 is 4007_2000h base + 9Ch offset = 4007_209Ch

Bit	7	6	5	4	3	2	1	0
Read	BDTBA							0
Write								▨
Reset	0	0	0	0	0	0	0	0

USBx_BDTPAGE1 field descriptions

Field	Description
7–1 BDTBA	This field provides address bits 15 through 9 of the BDT base address.
0 Reserved	This read-only field is reserved and always has the value zero.

40.4.17 Frame Number Register Low (USBx_FRMNUML)

The Frame Number Register (Low and High) contains an 11-bit value used to compute the address where the current Buffer Descriptor Table (BDT) resides in system memory.

Addresses: USB0_FRMNUML is 4007_2000h base + A0h offset = 4007_20A0h

Bit	7	6	5	4	3	2	1	0
Read Write	FRM[7:0]							
Reset	0	0	0	0	0	0	0	0

USBx_FRMNUML field descriptions

Field	Description
7–0 FRM[7:0]	This 8-bit field and the 3-bit field in the Frame Number Register High are used to compute the address where the current Buffer Descriptor Table (BDT) resides in system memory.

40.4.18 Frame Number Register High (USBx_FRMNUMH)

The Frame Number Register (Low and High) contains an 11-bit value used to compute the address where the current Buffer Descriptor Table (BDT) resides in system memory.

Addresses: USB0_FRMNUMH is 4007_2000h base + A4h offset = 4007_20A4h

Bit	7	6	5	4	3	2	1	0
Read	0					FRM[10:8]		
Write								
Reset	0	0	0	0	0	0	0	0

USBx_FRMNUMH field descriptions

Field	Description
7–3 Reserved	This read-only field is reserved and always has the value zero.
2–0 FRM[10:8]	This 3-bit field and the 8-bit field in the Frame Number Register Low are used to compute the address where the current Buffer Descriptor Table (BDT) resides in system memory.

40.4.19 Token Register (USBx_TOKEN)

The Token Register is used to perform USB transactions when in host mode (HOSTMODEEN=1). When the processor core wishes to execute a USB transaction to a peripheral, it writes the TOKEN type and endpoint to this register. After this register has been written, the USB module begins the specified USB transaction to the address contained in the address register. The processor core should always check that the TOKEN_BUSY bit in the control register is not set before performing a write to the Token Register. This ensures token commands are not overwritten before they can be executed. The address register and endpoint control register 0 are also used when performing a token command and therefore must also be written before the Token Register. The address register is used to correctly select the USB peripheral address transmitted by the token command. The endpoint control register determines the handshake and retry policies used during the transfer.

Addresses: USB0_TOKEN is 4007_2000h base + A8h offset = 4007_20A8h

Bit	7	6	5	4	3	2	1	0
Read Write	TOKENPID				TOKENENDPT			
Reset	0	0	0	0	0	0	0	0

USBx_TOKEN field descriptions

Field	Description
7–4 TOKENPID	This 4-bit field contains the token type executed by the USB Module. 0001 OUT Token. USB Module performs an OUT (TX) transaction. 1001 IN Token. USB Module performs an In (RX) transaction. 1101 SETUP Token. USB Module performs a SETUP (TX) transaction
3–0 TOKENENDPT	This 4 bit field holds the Endpoint address for the token command. The four bit value written must be a valid endpoint.

40.4.20 SOF Threshold Register (USBx_SOFTHLD)

The SOF Threshold Register is used only in Hosts mode (HOSTMODEEN=1). When in Host mode, the 14-bit SOF counter counts the interval between SOF frames. The SOF must be transmitted every 1msec so the SOF counter is loaded with a value of 12000. When the SOF counter reaches zero, a Start Of Frame (SOF) token is transmitted. The SOF threshold register is used to program the number of USB byte times before the SOF to stop initiating token packet transactions. This register must be set to a value that ensures that other packets are not actively being transmitted when the SOF time counts to zero. When the SOF counter reaches the threshold value, no more tokens are transmitted until after the SOF ha been transmitted. The value programmed into the threshold register must reserve enough time to ensure the worst case transaction completes. In general the worst case transaction is a IN token followed by a data packet from the target followed by the response from the host. The actual time required is a function of the maximum packet size on the bus. Typical values for the SOF threshold are: 64-byte packets=74; 32-byte packets=42; 16-byte packets=26; 8-byte packets=18.

Addresses: USB0_SOFTHLD is 4007_2000h base + ACh offset = 4007_20ACh

Bit	7	6	5	4	3	2	1	0
Read Write	CNT							
Reset	0	0	0	0	0	0	0	0

USBx_SOFTHLD field descriptions

Field	Description
7–0 CNT	This 8-bit field represents the SOF count threshold in byte times.

40.4.21 BDT Page Register 2 (USBx_BDTPAGE2)

The Buffer Descriptor Table Page Register 2 contains an 8-bit value used to compute the address where the current Buffer Descriptor Table (BDT) resides in system memory.

Addresses: USB0_BDTPAGE2 is 4007_2000h base + B0h offset = 4007_20B0h

Bit	7	6	5	4	3	2	1	0
Read Write	BDTBA							
Reset	0	0	0	0	0	0	0	0

USBx_BDTPAGE2 field descriptions

Field	Description
7–0 BDTBA	This 8 bit field provides address bits 23 through 16 of the BDT base address, which defines where the Buffer Descriptor Table resides in system memory.

40.4.22 BDT Page Register 3 (USBx_BDTPAGE3)

The Buffer Descriptor Table Page Register 3 contains an 8-bit value used to compute the address where the current Buffer Descriptor Table (BDT) resides in system memory.

Addresses: USB0_BDTPAGE3 is 4007_2000h base + B4h offset = 4007_20B4h

Bit	7	6	5	4	3	2	1	0
Read Write	BDTBA							
Reset	0	0	0	0	0	0	0	0

USBx_BDTPAGE3 field descriptions

Field	Description
7–0 BDTBA	This 8 bit field provides address bits 31 through 24 of the BDT base address, which defines where the Buffer Descriptor Table resides in system memory.

40.4.23 Endpoint Control Register (USB*x*_ENDPT*n*)

The Endpoint Control Registers contain the endpoint control bits for each of the 16 endpoints available within the USB Module for a decoded address. The format for these registers is shown in the following figure. Endpoint 0 (ENDPT0) is associated with control pipe 0, which is required for all USB functions. Therefore, after a USBRST interrupt occurs the processor core should set the ENDPT0 register to contain 0x0D.

In Host mode ENDPT0 is used to determine the handshake, retry and low speed characteristics of the host transfer. For Host mode control, bulk and interrupt transfers the EPHSHK bit should be set to 1. For Isochronous transfers it should be set to 0. Common values to use for ENDPT0 in host mode are 0x4D for Control, Bulk, and Interrupt transfers, and 0x4C for Isochronous transfers.

Addresses: 4007_2000h base + C0h offset + (4d × *n*), where *n* = 0d to 15d

Bit	7	6	5	4	3	2	1	0
Read	HOSTWOHUB	RETRYDIS	0	EPCTLDIS	EPRXEN	EPTXEN	EPSTALL	EPHSHK
Write			▨					
Reset	0	0	0	0	0	0	0	0

USBx_ENDPT*n* field descriptions

Field	Description
7 HOSTWOHUB	This is a Host mode only bit and is only present in the control register for endpoint 0 (ENDPT0). When set this bit allows the host to communicate to a directly connected low speed device. When cleared, the host produces the PRE_PID then switch to low speed signaling when sending a token to a low speed device as required to communicate with a low speed device through a hub.
6 RETRYDIS	This is a Host mode only bit and is only present in the control register for endpoint 0 (ENDPT0). When set this bit causes the host to not retry NAK'ed (Negative Acknowledgement) transactions. When a transaction is NAKed, the BDT PID field is updated with the NAK PID, and the TOKEN_DNE interrupt is set. When this bit is cleared NAKed transactions is retried in hardware. This bit must be set when the host is attempting to poll an interrupt endpoint.
5 Reserved	This read-only field is reserved and always has the value zero.
4 EPCTLDIS	This bit, when set, disables control (SETUP) transfers. When cleared, control transfers are enabled. This applies if and only if the EPRXEN and EPTXEN bits are also set.
3 EPRXEN	This bit, when set, enables the endpoint for RX transfers.
2 EPTXEN	This bit, when set, enables the endpoint for TX transfers.
1 EPSTALL	When set this bit indicates that the endpoint is called. This bit has priority over all other control bits in the EndPoint Enable Register, but it is only valid if EPTXEN=1 or EPRXEN=1. Any access to this endpoint causes the USB Module to return a STALL handshake. After an endpoint is stalled it requires intervention from the Host Controller.
0 EPHSHK	When set this bet enables an endpoint to perform handshaking during a transaction to this endpoint. This bit is generally set unless the endpoint is Isochronous.

40.4.24 USB Control Register (USBx_USBCTRL)

Addresses: USB0_USBCTRL is 4007_2000h base + 100h offset = 4007_2100h

Bit	7	6	5	4	3	2	1	0
Read	SUSP	PDE	0					
Write			▨					
Reset	1	1	0	0	0	0	0	0

USBx_USBCTRL field descriptions

Field	Description
7 SUSP	Places the USB transceiver into the suspend state. 0 USB transceiver is not in suspend state. 1 USB transceiver is in suspend state.
6 PDE	Enables the weak pulldowns on the USB transceiver. 0 Weak pulldowns are disabled on D+ and D- 1 Weak pulldowns are enabled on D+ and D-.
5–0 Reserved	This read-only field is reserved and always has the value zero.

40.4.25 USB OTG Observe Register (USBx_OBSERVE)

Provides visibility on the state of the pull-ups and pull-downs at the transceiver. Useful when interfacing to an external OTG control module via a serial interface.

Addresses: USB0_OBSERVE is 4007_2000h base + 104h offset = 4007_2104h

Bit	7	6	5	4	3	2	1	0
Read	DPPU	DPPD	0	DMPD	0			0
Write								
Reset	0	1	0	1	0	0	0	0

USBx_OBSERVE field descriptions

Field	Description
7 DPPU	Provides observability of the D+ Pull Up enable at the USB transceiver . 0 D+ pullup disabled. 1 D+ pullup enabled.
6 DPPD	Provides observability of the D+ Pull Down enable at the USB transceiver . 0 D+ pulldown disabled. 1 D+ pulldown enabled.
5 Reserved	This read-only field is reserved and always has the value zero.
4 DMPD	Provides observability of the D- Pull Down enable at the USB transceiver . 0 D- pulldown disabled. 1 D- pulldown enabled.
3–1 Reserved	This read-only field is reserved and always has the value zero.
0 Reserved	This read-only field is reserved and always has the value zero.

40.4.26 USB OTG Control Register (USBx_CONTROL)

Addresses: USB0_CONTROL is 4007_2000h base + 108h offset = 4007_2108h

Bit	7	6	5	4
Read	0			DPPULLUPNONOTG
Write				
Reset	0	0	0	0

Bit	3	2	1	0
Read	0			
Write				
Reset	0	0	0	0

USBx_CONTROL field descriptions

Field	Description
7–5 Reserved	This read-only field is reserved and always has the value zero.
4 DPPULLUPNONOTG	Provides control of the DP PULLUP in the USB OTG module, if USB is configured in non-OTG device mode. 0 DP Pull up in non-OTG device mode is not enabled. 1 DP Pull up in non-OTG device mode is enabled.
3–0 Reserved	This read-only field is reserved and always has the value zero.

40.4.27 USB Transceiver Control Register 0 (USBx_USBTRC0)

Addresses: USB0_USBTRC0 is 4007_2000h base + 10Ch offset = 4007_210Ch

Bit	7	6	5	4
Read		0	USBRESMEN	0
Write	USBRESET			
Reset	0	0	0	0

Bit	3	2	1	0
Read	0		SYNC_DET	USB_RESUME_INT
Write				
Reset	0	0	0	0

USBx_USBTRC0 field descriptions

Field	Description
7 USBRESET	USB reset Generates a hard reset to the USB_OTG module. After this bit is set and the reset occurs, this bit is automatically cleared.

Table continues on the next page...

USBx_USBTRC0 field descriptions (continued)

Field	Description
	NOTE: It is always read as zero. 0 Normal USB module operation. 1 Returns the USB module to its reset state.
6 Reserved	This read-only field is reserved and always has the value zero.
5 USBRESMEN	Asynchronous Resume Interrupt Enable This bit, when set, allows the USB module to send an asynchronous wakeup event to the MCU upon detection of resume signaling on the USB bus. The MCU then re-enables clocks to the USB module. It is used for low-power suspend mode when USB module clocks are stopped or the USB transceiver is in Suspend mode. Async wakeup only works in device mode. 0 USB asynchronous wakeup from suspend mode disabled. 1 USB asynchronous wakeup from suspend mode enabled. The asynchronous resume interrupt differs from the synchronous resume interrupt in that it asynchronously detects K-state using the unfiltered state of the D+ and D- pins. This interupt should only be enabled when the Transceiver is suspended.
4–2 Reserved	This read-only field is reserved and always has the value zero.
1 SYNC_DET	Synchronous USB Interrupt Detect 0 Synchronous interrupt has not been detected. 1 Synchronous interrupt has been detected.
0 USB_RESUME_INT	USB Asynchronous Interrupt 0 No interrupt was generated. 1 Interrupt was generated because of the USB asynchronous interrupt.

40.4.28 Frame Adjust Register (USBx_USBFRMADJUST)

Addresses: USB0_USBFRMADJUST is 4007_2000h base + 114h offset = 4007_2114h

Bit	7	6	5	4	3	2	1	0
Read Write	ADJ							
Reset	0	0	0	0	0	0	0	0

USBx_USBFRMADJUST field descriptions

Field	Description
7–0 ADJ	Frame Adjustment In Host mode, the frame adjustment is a twos complement number that adjusts the period of each USB frame in 12-MHz clock periods. A SOF is normally generated every 12,000 12-MHz clock cycles. The Frame Adjust Register can adjust this by -128 to +127 to compensate for inaccuracies in the USB 48-MHz clock. Changes to the ADJ bit take effect at the next start of the next frame.

40.5 OTG and Host Mode Operation

The Host Mode logic allows devices such as digital cameras and palmtop computers to function as a USB Host Controller. The OTG logic adds an interface to allow the OTG Host Negotiation and Session Request Protocols (HNP and SRP) to be implemented in software. Host Mode allows a peripheral such as a digital camera to be connected directly to a USB compliant printer. Digital photos can then be easily printed without having to upload them to a PC. In the palmtop computer application, a USB compliant keyboard/ mouse can be connected to the palmtop computer with the obvious advantages of easier interaction.

Host mode is intended for use in handheld-portable devices to allow easy connection to simple HID class devices such as printers and keyboards. It is NOT intended to perform the functions of a full OHCI or UHCI compatible host controller found on PC motherboards. The USB-FS is not supported by Windows 98 as a USB host controller. Host mode allows bulk, Isochronous, interrupt and control transfers. Bulk data transfers are performed at nearly the full USB bus bandwidth. Support is provided for ISO transfers, but the number of ISO streams that can be practically supported is affected by the interrupt latency of the processor servicing the token during interrupts from the SIE. Custom drivers must be written to support Host mode operation.

Setting the HOST_MODE_EN bit in the CTL register enables host Mode. The USB-FS core can only operate as a peripheral device or in Host Mode. It cannot operate in both modes simultaneously. When HOST_MODE is enabled, only endpoint zero is used. All other endpoints should be disabled by software.

40.6 Host Mode Operation Examples

The following sections illustrate the steps required to perform USB host functions using the USB-FS core. While it is useful to understand the interaction of the hardware and the software at a detailed level, an understanding of the interactions at this level is not required to write host applications using the API software.

To enable host mode and discover a connected device:

1. Enable Host Mode (CTL[HOST_MODE_EN]=1). The pull-down resistors are enabled, and pull-up disabled. Start of Frame (SOF) generation begins. SOF counter loaded with 12,000. Disable SOF packet generation to eliminate noise on the USB by writing the USB enable bit to 0 (CTL[USB_EN]=0).

2. Enable the ATTACH interrupt (INT_ENB[ATTACH]=1).
3. Wait for ATTACH interrupt (INT_STAT[ATTACH]). Signaled by USB Target pull-up resistor changing the state of DPLUS or DMINUS from 0 to 1 (SE0 to J or K state).
4. Check the state of the JSTATE and SE0 bits in the control register. If the connecting device is low speed (JSTATE bit is 0), set the low-speed bit in the address registers (ADDR[LS_EN]=1) and the host without hub bit in endpoint 0 register control (EP_CTL0[HOST_WO_HUB]=1).
5. Enable RESET (CTL[RESET]=1) for 10 ms.
6. Enable SOF packet to keep the connected device from going to suspend (CTL[USB_EN=1]).
7. Start enumeration by sending a sequence of device framework commands, device frame work packets to the default control pipe of the connected device. Refer to the *Universal Serial Bus Revision 2.0 specification,* "Chapter 9 USB Device Framework" (http://www.usb.org/developers/docs).

To complete a control transaction to a connected device:

1. Complete all steps discover a connected device
2. Set up the endpoint control register for bidirectional control transfers EP_CTL0[4:0] = 0x0d.
3. Place a copy of the device framework setup command in a memory buffer. Refer to the *Universal Serial Bus Revision 2.0 specification,* "Chapter 9 USB Device Framework" (http://www.usb.org/developers/docs).
4. Initialize current (even or odd) TX EP0 BDT to transfer the 8 bytes of command data for a device framework command (for example, a GET DEVICE DESCRIPTOR).
    - Set the BDT command word to 0x00080080 –Byte count to 8, own bit to 1.
    - Set the BDT buffer address field to the start address of the 8 byte command buffer.
5. Set the USB device address of the target device in the address register (ADDR[6:0]). After the USB bus reset, the device USB address is zero. It is set to some other value (usually 1) by the Set Address device framework command.
6. Write the token register with a SETUP to Endpoint 0 the target device default control pipe (TOKEN=0xD0). This initiates a setup token on the bus followed by a data packet. The device handshake is returned in the BDT PID field after the packets

complete. When the BDT is written a token done (INT_STAT[TOK_DNE]) interrupt is asserted. This completes the setup phase of the setup transaction. Refer to the *Universal Serial Bus Revision 2.0 specification,* "Chapter 9 USB Device Framework" (http://www.usb.org/developers/docs).

7. To initiate the data phase of the setup transaction (for example, get the data for the GET DEVICE descriptor command) set up a buffer in memory for the data to be transferred.
8. Initialize the current (even or odd) TX EP0 BDT to transfer the data.
    - Set the BDT command word to 0x004000C0 –Byte count to the length of the data buffer in this case 64, own bit to 1, Data toggle to Data1.
    - Set the BDT buffer address field to the start address of the data buffer
9. Write the token register with a IN or OUT token to Endpoint 0 the target device default control pipe, an IN token for a GET DEVICE DESCRIPTOR command (TOKEN=0x90). This initiates an IN token on the bus followed by a data packet from the device to the host. When the data packet completes the BDT is written and a token done (INT_STAT[TOK_DNE]) interrupt is asserted. For control transfers with a single packet data phase this completes the data phase of the setup transaction. Refer to the *Universal Serial Bus Revision 2.0 specification,* "Chapter 9 USB Device Framework" (http://www.usb.org/developers/docs).
10. To initiate the Status phase of the setup transaction set up a buffer in memory to receive or send the zero length status phase data packet.
11. Initialize the current (even or odd) TX EP0 BDT to transfer the status data.
    - Set the BDT command word to 0x00000080 –Byte count to the length of the data buffer in this case 0, own bit to 1, Data toggle to Data0.
    - Set the BDT buffer address field to the start address of the data buffer
12. Write the token register with a IN or OUT token to Endpoint 0 the target device default control pipe, an OUT token for a GET DEVICE DESCRIPTOR command (TOKEN=0x10). This initiates an OUT token on the bus followed by a zero length data packet from the host to the device. When the data packet completes the BDT is written with the handshake form the device and a token done (INT_STAT[TOK_DNE]) interrupt is asserted. This completes the data phase of the setup transaction. Refer to the *Universal Serial Bus Revision 2.0 specification,* "Chapter 9 USB Device Framework" (http://www.usb.org/developers/docs).

To send a Full speed bulk data transfer to a target device:

1. Complete all steps discover a connected device and to configure a connected device. Write the ADDR register with the address of the target device. Typically, there is only one other device on the USB bus in host mode so it is expected that the address is 0x01 and should remain constant.
2. Write the ENDPT0 to 0x1D register to enable transmit and receive transfers with handshaking enabled.
3. Setup the Even TX EP0 BDT to transfer up to 64 bytes.
4. Set the USB device address of the target device in the address register (ADDR[6:0]).
5. Write the TOKEN register with an OUT token to the desired endpoint. The write to this register triggers the USB-FS transmit state machines to begin transmitting the TOKEN and the data.
6. Setup the Odd TX EP0 BDT to transfer up to 64 bytes.
7. Write the TOKEN register with an OUT token as in step 4. Two Tokens can be queued at a time to allow the packets to be double buffered to achieve maximum throughput.
8. Wait for the TOK_DNE interrupt. This indicates one of the BDTs has been released back to the microprocessor and that the transfer has completed. If the target device asserts NAKs, the USB-FS continues to retry the transfer indefinitely without processor intervention unless the RETRY_DIS retry disable bit is set in the EP0 control register. If the retry disable bit is set, the handshake (ACK, NAK, STALL, or ERROR (0xf)) is returned in the BDT PID field. If a stall interrupt occurs, the pending packet must be dequeued and the error condition in the target device cleared. If a RESET interrupt occurs (SE0 for more than 2.5us), the target has detached.
9. After the TOK_DNE interrupt occurs, the BDTs can be examined and the next data packet queued by returning to step 2.

40.7 On-The-Go Operation

The USB-OTG core provides sensors and controls to enable On-The-Go (OTG) operation. These sensors are used by the OTG API software to implement the Host Negotiation Protocol (HNP) and Session Request Protocol (SRP). API calls are provided to give access the OTG protocol control signals, and include the OTG capabilities in the device application. The following state machines show the OTG operations involved with HNP and SRP protocols from either end of the USB cable.

40.7.1 OTG Dual Role A Device Operation

A device is considered the A device because of the type of cable attached. If the USB Type A connector or the USB Type Mini A connector is plugged into the device, he is considered the A device.

A dual role A device operates as the following flow diagram and state description table illustrates.

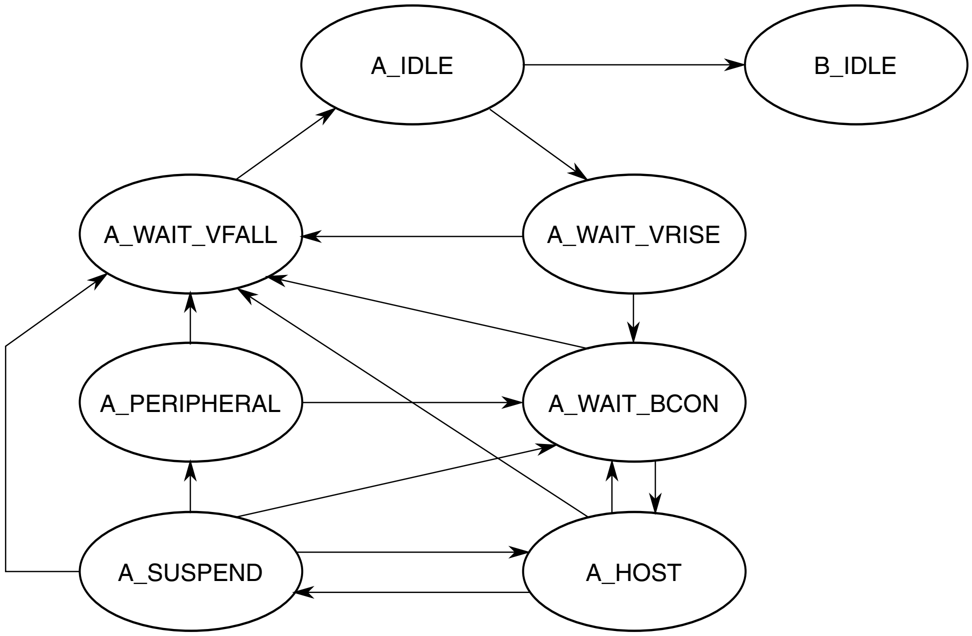


Figure 40-93. Dual Role A Device Flow Diagram

Table 40-96. State Descriptions for the Dual Role A Device Flow

State	Action	Response
A_IDLE	If ID Interrupt. The cable has been un-plugged or a Type B cable has been attached. The device now acts as a Type B device.	Go to B_IDLE
	If the A application wants to use the bus or if the B device is doing an SRP as indicated by an A_SESS_VLD Interrupt or Attach or Port Status Change Interrupt check data line for 5 –10 msec pulsing.	Go to A_WAIT_VRISE Turn on DRV_VBUS
A_WAIT_VRISE	If ID Interrupt or if A_VBUS_VLD is false after 100 msec The cable has been changed or the A device cannot support the current required from the B device.	Go to A_WAIT_VFALL Turn off DRV_VBUS
	If A_VBUS_VLD interrupt	Go to A_WAIT_BCON

Table continues on the next page...

Table 40-96. State Descriptions for the Dual Role A Device Flow (continued)

State	Action	Response
A_WAIT_BCON	After 200 msec without Attach or ID Interrupt. (This could wait forever if desired.)	Go to A_WAIT_FALL Turn off DRV_VBUS
	A_VBUS_VLD Interrupt and B device attaches	Go to A_HOST Turn on Host Mode
A_HOST	Enumerate Device determine OTG Support.	
	If A_VBUS_VLD/ Interrupt or A device is done and doesn't think he wants to do something soon or the B device disconnects	Go to A_WAIT_VFALL Turn off Host Mode Turn off DRV_VBUS
	If the A device is finished with session or if the A device wants to allow the B device to take bus.	Go to A_SUSPEND
	ID Interrupt or the B device disconnects	Go to A_WAIT_BCON
A_SUSPEND	If ID Interrupt, or if 150 msec B disconnect timeout (This timeout value could be longer) or if A_VBUS_VLD\ Interrupt	Go to A_WAIT_VFALL Turn off DRV_VBUS
	If HNP enabled, and B disconnects in 150 msec then B device is becoming the host.	Go to A_PERIPHERAL Turn off Host Mode
	If A wants to start another session	Go to A_HOST
A_PERIPHERAL	If ID Interrupt or if A_VBUS_VLD interrupt	Go to A_WAIT_VFALL Turn off DRV_VBUS.
	If 3 –200 msec of Bus Idle	Go to A_WAIT_BCON Turn on Host Mode
A_WAIT_VFALL	If ID Interrupt or (A_SESS_VLD/ & b_conn/)	Go to A_IDLE

40.7.2 OTG Dual Role B Device Operation

A device is considered a B device if it connected to the bus with a USB Type B cable or a USB Type Mini B cable.

A dual role B device operates as the following flow diagram and state description table illustrates.

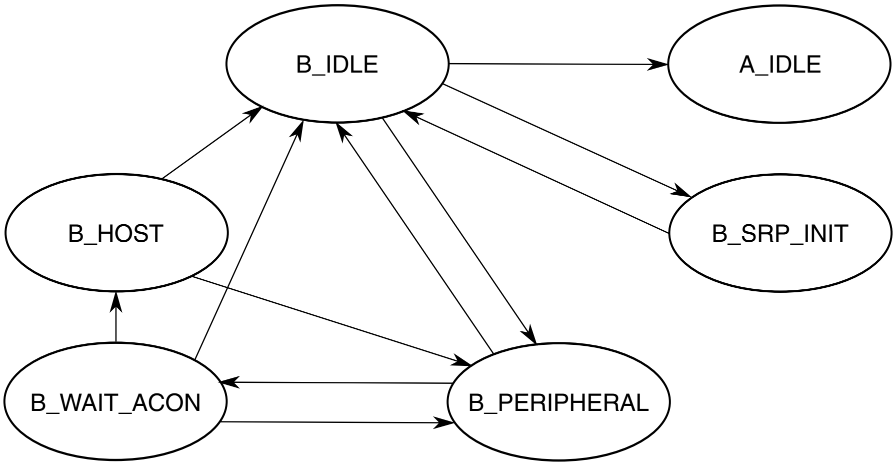


Figure 40-94. Dual Role B Device Flow Diagram

Table 40-97. State Descriptions for the Dual Role B Device Flow

State	Action	Response
B_IDLE	If ID\ Interrupt. A Type A cable has been plugged in and the device should now respond as a Type A device.	Go to A_IDLE
	If B_SESS_VLD Interrupt. The A device has turned on VBUS and begins a session.	Go to B_PERIPHERAL Turn on DP_HIGH
	If B application wants the bus and Bus is Idle for 2 ms and the B_SESS_END bit is set, the B device can perform an SRP.	Go to B_SRP_INIT Pulse CHRG_VBUS Pulse DP_HIGH 5-10 ms
B_SRP_INIT	If ID\ Interrupt or SRP Done (SRP must be done in less than 100 msecs.)	Go to B_IDLE
B_PERIPHERAL	If HNP enabled and the bus is suspended and B wants the bus, the B device can become the host.	Go to B_WAIT_ACON Turn off DP_HIGH
B_WAIT_ACON	If A connects, an attach interrupt is received	Go to B_HOST Turn on Host Mode
	If ID\ Interrupt or B_SESS_VLD/ Interrupt If the cable changes or if VBUS goes away, the host doesn't support us. Go to B_IDLE	Go to B_IDLE
	If 3.125 ms expires or if a Resume occurs	Go to B_PERIPHERAL
B_HOST	If ID\ Interrupt or B_SESS_VLD\ Interrupt If the cable changes or if VBUS goes away, the host doesn't support us.	Go to B_IDLE
	If B application is done or A disconnects	Go to B_PERIPHERAL

Chapter 41
USB Device Charger Detection Module (USBDCD)

41.1 Preface

41.1.1 References

The following publications are referenced in this document. For updates to these specifications, see http://www.usb.org.

- *USB Battery Charging Specification Revision 1.1, USB Implementers Forum*
- *Universal Serial Bus Specification Revision 2.0, USB Implementers Forum*

41.1.2 Acronyms and abbreviations

The following table contains acronyms and abbreviations used in this document.

Table 41-1. Acronyms and abbreviated terms

Term	Meaning
FS	Full speed (12 Mbit/s)
HS	High speed (480 Mbit/s)
I_{DEV_DCHG}	Current drawn when the USB device is connected to a dedicated charging port
$I_{DEV\_HCHG\_LFS}$	Current drawn when the USB device is connected to an FS charging host port
I_{DM_SINK}	Current sink for the D– line
I_{DP_SRC}	Current source for the D+ line
I_{SUSP}	Current drawn when the USB device is suspended
LDO	Low dropout
LS	Low Speed (1.5 Mbit/s)
N/A	Not applicable

Table continues on the next page...

Table 41-1. Acronyms and abbreviated terms (continued)

Term	Meaning
OTG	On-The-Go
R_{DM_DWN}	D– pulldown resistance for data pin contact detect
V_{DAT_REF}	Data detect reference voltage for the voltage comparator
V_{DP_SRC}	Voltage source for the D+ line
V_{LGC}	Threshold voltage for logic high

41.1.3 Glossary

The following table shows a glossary of terms used in this document.

Table 41-2. Glossary of terms

Term	Definition
Transceiver	Module that implements the physical layer of the USB standard (FS or LS only).
PHY	Module that implements the physical layer of the USB standard (HS capable).
Attached	Device is physically plugged into USB port, but has *not enabled* either D+ or D– pullup resistor.
Connected	Device is physically plugged into USB port, and has *enabled* either D+ or D– pullup resistor.
Suspended	After 3 ms of no bus activity, the USB device enters suspend mode.
Component	The hardware and software that make up a subsystem.

41.2 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The USBDCD module works with the USB transceiver to detect whether the USB device is attached to a charging port, either a dedicated charging port or a charging host. System software coordinates the detection activites of the module and controls an off-chip integrated circuit that performs the battery charging.

41.2.1 Block diagram

The following figure is a high level block diagram of the module.

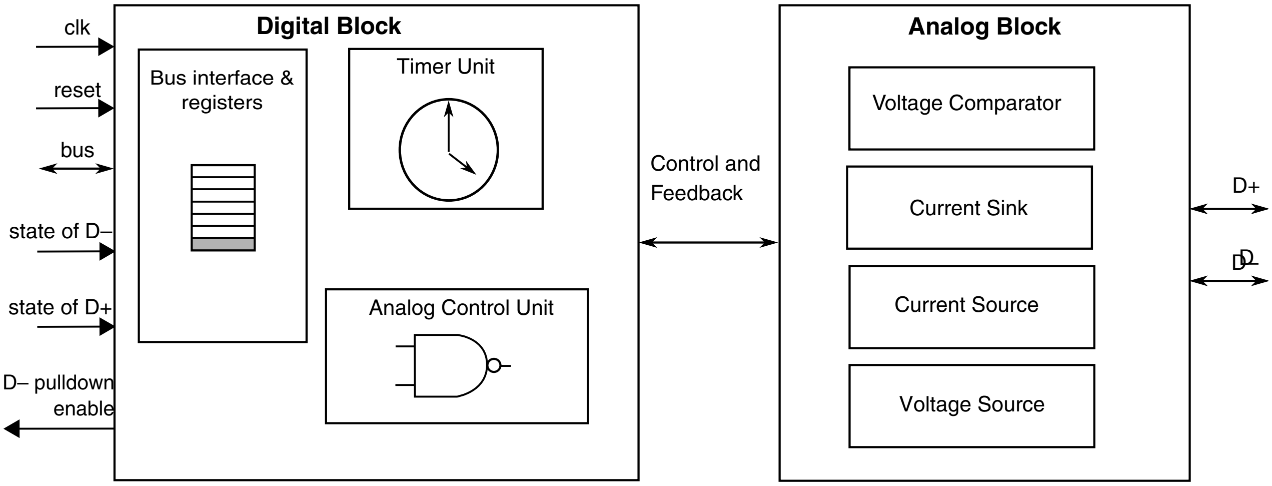

clk
reset
bus
state of D–
state of D+
D– pulldown enable
Digital Block
Bus interface & registers
Timer Unit
Analog Control Unit
Control and Feedback
Analog Block
Voltage Comparator
Current Sink
Current Source
Voltage Source
D+
D–

Figure 41-1. Block diagram

The USBDCD module consists of two main blocks:

- A digital block provides the programming interface (memory-mapped registers) and includes the timer unit and the analog control unit.
- An analog block provides the circuitry for the physical detection of the charger, including the voltage source, current source, current sink, and voltage comparator circuitry.

41.2.2 Features

The USBDCD module offers the following features:

- Compliant with the latest industry standard specification: *USB Battery Charging Specification, Revision 1.1*
- Programmable timing parameters default to values required by the industry standards:
  - Having standard default values allows for easy configuration- simply set the clock frequency before enabling the module.
  - Programmability allows the flexibility to meet future updates of the standards.

41.2.3 Modes of operation

The operating modes of the USBDCD module are shown in the following table.

Table 41-3. Module modes and their conditions

Module mode	Description	Conditions when used
Enabled	The module performs the charger detection sequence.	System software should enable the module only when *all* of the following conditions are true: • The system uses a rechargeable battery. • The device is being used in an FS USB device application. • The device has detected that it is attached to the USB cable.
Disabled	The module is not active and is held in a low power state.	System software should disable the module when *either* of the following conditions is true: • The charger detect sequence is complete. • The conditions for being enabled are not met.
Powered Off	The digital supply voltage dvdd is removed.	Low system performance requirements allow putting the device into a very low-power stop mode.

Operating mode transitions are shown in the following table.

Table 41-4. Entering and exiting module modes

Module mode	Entering	Exiting	Mode after exiting
Enabled	Set CONTROL[START].	Set CONTROL[SR].[1]	Disabled
Disabled	Take *either* of the following actions: • Set CONTROL[SR].[1] • Reset the module. By default, the module is disabled.	Set CONTROL[START].	Enabled
Powered Off	Perform the following actions: 1. Put the device into very low-power stop mode. 2. Adjust the supply voltages.	Perform the following actions: 1. Restore the supply voltages. 2. Take the device out of very low-power stop mode.	Disabled

1. The effect of setting the SR bit is immediate; that is, the module is disabled even if the sequence has not completed.

41.3 Module signal descriptions

This section describes the module signals. The following table shows a summary of module signals that interface with the pins of the device.

Table 41-5. Signal descriptions

Signal	Description	I/O
usb_dm	USB D– analog data signal. The analog block interfaces directly to the D– signal on the USB bus.	I/O
usb_dp	USB D+ analog data signal. The analog block interfaces directly to the D+ signal on the USB bus.	I/O
avdd33[1]	3.3 V regulated analog supply	I
avss	Analog ground	I
dvss	Digital ground	I
dvdd	1.2 V digital supply	I

1. Voltage must be 3.3 V +/- 10% for full functionality of the module. That is, the charger detection function does not work when this voltage is below 3.0 V, and the CONTROL[START] bit should not be set.

NOTE

> The transceiver module also interfaces to the usb_dm and usb_dp signals. Both modules and the USB host/hub use these signals as bidirectional, tristate signals.

Information about the signal integrity aspects of the lines including shielding, isolated return paths, input or output impedance, packaging, suggested external components, ESD, and other protections can be found in the USB 2.0 specification and in Application information.

41.4 Memory map/Register definition

This section describes the memory map and registers for the USBDCD module.

USBDCD memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_5000	Control register (USBDCD_CONTROL)	32	R/W	0001_0000h	41.4.1/ 930
4003_5004	Clock register (USBDCD_CLOCK)	32	R/W	0000_00C1h	41.4.2/ 932
4003_5008	Status register (USBDCD_STATUS)	32	R	0000_0000h	41.4.3/ 933
4003_5010	TIMER0 register (USBDCD_TIMER0)	32	R/W	0010_0000h	41.4.4/ 934

Table continues on the next page...

USBDCD memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4003_5014	TIMER1 register (USBDCD_TIMER1)	32	R/W	000A_0028h	41.4.5/ 935
4003_5018	TIMER2 register (USBDCD_TIMER2)	32	R/W	0028_0001h	41.4.6/ 936

41.4.1 Control register (USBDCD_CONTROL)

Contains the control and interrupt bit fields.

Address: USBDCD_CONTROL is 4003_5000h base + 0h offset = 4003_5000h

Bit	31	30	29	28	27	26	25	24
R	0						0	0
W							SR	START
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							IE
W								
Reset	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
R	Reserved							IF
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	0							0
W								IACK
Reset	0	0	0	0	0	0	0	0

USBDCD_CONTROL field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero.
25 SR	Software Reset Determines whether a software reset is performed. 0 Do not perform a software reset. 1 Perform a software reset.
24 START	Start Change Detection Sequence Determines whether the charger detection sequence is initiated.

Table continues on the next page...

USBDCD_CONTROL field descriptions (continued)

Field	Description
	0 Do not start the sequence. Writes of this value have no effect. 1 Initiate the charger detection sequence. If the sequence is already running, writes of this value have no effect.
23–17 Reserved	This read-only field is reserved and always has the value zero.
16 IE	Interrupt Enable Enables/disables interrupts to the system. 0 Disable interrupts to the system. 1 Enable interrupts to the system.
15–9 Reserved	This field is reserved.
8 IF	Interrupt Flag Determines whether an interrupt is pending. 0 No interrupt is pending. 1 An interrupt is pending.
7–1 Reserved	This read-only field is reserved and always has the value zero.
0 IACK	Interrupt Acknowledge Determines whether the interrupt is cleared. 0 Do not clear the interrupt. 1 Clear the IF bit (interrupt flag).

41.4.2 Clock register (USBDCD_CLOCK)

Address: USBDCD_CLOCK is 4003_5000h base + 4h offset = 4003_5004h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	0				CLOCK_SPEED[2:6]			
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	CLOCK_SPEED[5:0]						0	CLOCK_ UNIT
W								
Reset	1	1	0	0	0	0	0	1

USBDCD_CLOCK field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value zero.
11–2 CLOCK_SPEED	Numerical Value of Clock Speed in Binary The unit of measure is programmed in CLOCK_UNIT. The valid range is from 1 to 1023 when clock unit is MHz and 4 to 1023 when clock unit is kHz. Examples with CLOCK_UNIT = 1: • For 48 MHz: 0b00_0011_0000 (48) (Default) • For 24 MHz: 0b00_0001_1000 (24) Examples with CLOCK_UNIT = 0: • For 100 kHz: 0b00_0110_0100 (100) • For 500 kHz: 0b01_1111_0100 (500)
1 Reserved	This read-only field is reserved and always has the value zero.
0 CLOCK_UNIT	Unit of Measurement Encoding for Clock Speed Specifies the unit of measure for the clock speed. 0 kHz Speed (between 1 kHz and 1023 kHz) 1 MHz Speed (between 1 MHz and 1023 MHz)

41.4.3 Status register (USBDCD_STATUS)

Provides the current state of the module for system software monitoring.

Address: USBDCD_STATUS is 4003_5000h base + 8h offset = 4003_5008h

Bit	31	30	29	28	27	26	25	24
R	0							
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	0	ACTIVE	TO	ERR	SEQ_STAT		SEQ_RES	
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R W	Reserved							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R W	Reserved							
Reset	0	0	0	0	0	0	0	0

USBDCD_STATUS field descriptions

Field	Description
31–23 Reserved	This read-only field is reserved and always has the value zero.
22 ACTIVE	Active Status Indicator Indicates whether the sequence is running. 0 The sequence is not running. 1 The sequence is running.
21 TO	Timeout Flag Indicates whether the detection sequence has passed the timeout threshhold. 0 The detection sequence has not been running for over 1 s. 1 It has been over 1 s since the data pin contact was detected and debounced.
20 ERR	Error Flag Indicates whether there is an error in the detection sequence. 0 No sequence errors. 1 Error in the detection sequence. See the SEQ_STAT field to determine the phase in which the error occurred.

Table continues on the next page...

USBDCD_STATUS field descriptions (continued)

Field	Description
19–18 SEQ_STAT	Charger Detection Sequence Status Indicates the status of the charger detection sequence. 00 The module is either not enabled, or the module is enabled but the data pins have not yet been detected. 01 Data pin contact detection is complete. 10 Charging port detection is complete. 11 Charger type detection is complete.
17–16 SEQ_RES	Charger Detection Sequence Results Reports how the charger detection is attached. 00 No results to report. 01 Attached to a standard host. Must comply with USB 2.0 by drawing only 2.5 mA (max) until connected. 10 Attached to a charging port. The exact meaning depends on bit 18: • 0: Attached to either a charging host or a dedicated charger. The charger type detection has not completed. • 1: Attached to a charging host. The charger type detection has completed. 11 Attached to a dedicated charger.
15–0 Reserved	This field is reserved.

41.4.4 TIMER0 register (USBDCD_TIMER0)

TIMER0 has an TSEQ_INIT field that represents the system latency in ms. Latency is measured from the time when VBUS goes active until the time system software initiates charger detection sequence in USBDCD module. When software sets the CONTROL[START] bit, the Unit Connection Timer (TUNITCON) is initialized with the value of TSEQ_INIT.

Valid values are 0–1023, however the USB Battery Charging Specification requires the entire sequence, including TSEQ_INIT, to be completed in 1s or less.

Address: USBDCD_TIMER0 is 4003_5000h base + 10h offset = 4003_5010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						TSEQ_INIT										0				TUNITCON											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

USBDCD_TIMER0 field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero.
25–16 TSEQ_INIT	Sequence Initiation Time TSEQ_INIT represents the system latency (in ms) measured from the time VBUS goes active to the time system software initiates the charger detection sequence in the USBDCD module. When software sets the CONTROL[START] bit, the Unit Connection Timer (TUNITCON) is initialized with the value of TSEQ_INIT. Valid values are 0-1023, but the USB Battery Charging Specification requires the entire sequence, including TSEQ_INIT, to be completed in 1s or less.
15–12 Reserved	This read-only field is reserved and always has the value zero.
11–0 TUNITCON	Unit Connection Timer Elapse (in ms) Displays the amount of elapsed time since the event of setting the START bit plus the value of TSEQ_INIT. The timer is automatically initialized with the value of TSEQ_INIT This timer enables compliance with the maximum time allowed to connect T_{UNIT_CON} under the USB Battery Charging Specification, v1.1.If the timer reaches the one second limit, the module triggers an interrupt and sets the error flag STATUS[ERR]. The timer continues counting throughout the charger detection sequence, even when control has been passed to software. As long as the module is active, the timer continues to count until it reaches the maximum value of 0xFFF (4095 ms). The timer does not rollover to zero. A software reset clears the timer.

41.4.5 TIMER1 register (USBDCD_TIMER1)

TIMER1 contains timing parameters. Note that register values can be written that are not compliant with the USB Battery Charging Specification v1.1, so care should be taken when overwriting the default values.

Address: USBDCD_TIMER1 is 4003_5000h base + 14h offset = 4003_5014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						TDCD_DBNC										0						TVDPSRC_ON									
W	▨	▨	▨	▨	▨	▨											▨	▨	▨	▨	▨	▨										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0

USBDCD_TIMER1 field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero.
25–16 TDCD_DBNC	Time Period to Debounce D+ Signal Sets the time period (ms) to debounce the D+ signal during the data pin contact detection phase. See "Debouncing the data pin contact" Valid values are 1–1023, but the USB Battery Charging Specification requires a minimum value of 10 ms.

Table continues on the next page...

USBDCD_TIMER1 field descriptions (continued)

Field	Description
15–10 Reserved	This read-only field is reserved and always has the value zero.
9–0 TVDPSRC_ON	Time Period Comparator Enabled This timing parameter is used after detection of the data pin. See "Charging Port Detection". Valid values are 1–1023, but the USB Battery Charging Specification requires a minimum value of 40 ms.

41.4.6 TIMER2 register (USBDCD_TIMER2)

TIMER2 contains timing parameters.

NOTE

> Register values can be written that are not compliant with the USB Battery Charging Specification v1.1, so care should be taken when overwriting the default values.

Address: USBDCD_TIMER2 is 4003_5000h base + 18h offset = 4003_5018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0						TVDPSRC_CON										0												CHECK_DM			
W	▨																▨															
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

USBDCD_TIMER2 field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value zero.
25–16 TVDPSRC_CON	Time Period Before Enabling D+ Pullup Sets the time period (ms) that the module waits after charging port detection before system software must enable the D+ pullup to connect to the USB host. Valid values are 1–1023, but the USB Battery Charging Specification requires a minimum value of 40 ms.
15–4 Reserved	This read-only field is reserved and always has the value zero.
3–0 CHECK_DM	Time Before Check of D– Line Sets the amount of time (in ms) that the module waits after the device connects to the USB bus until checking the state of the D� line to determine the type of charging port. See "Charger Type Detection." Valid values are 1–15ms.

41.5 Functional description

The sequence of detecting the presence of charging port and type of charging port involves several hardware components, coordinated by system software. This collection of interacting hardware and software is called the USB Battery Charging Subsystem. The following figure shows the USBDCD module as a component of the subsystem. The following table describes the components.

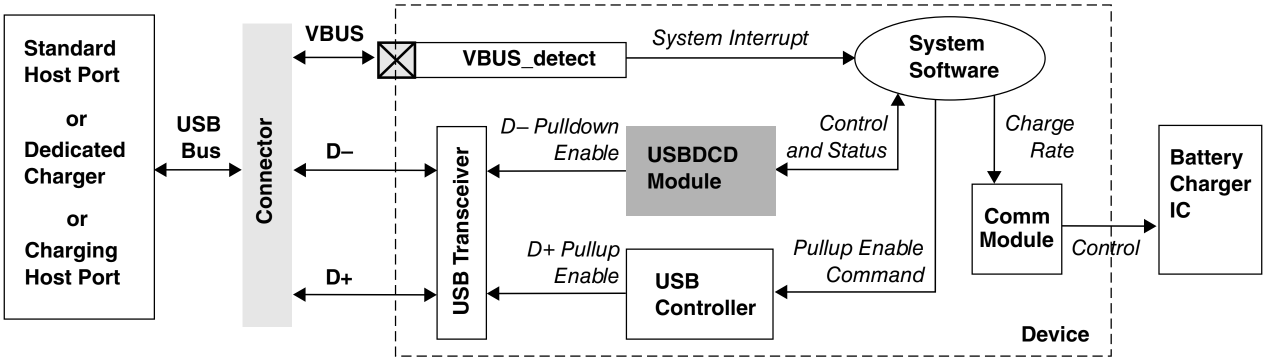

Standard Host Port
or
Dedicated Charger
or
Charging Host Port
USB Bus
Connector
VBUS
D–
D+
VBUS_detect
System Interrupt
System Software
USB Transceiver
D– Pulldown Enable
USBDCD Module
Control and Status
Charge Rate
D+ Pullup Enable
USB Controller
Pullup Enable Command
Comm Module
Control
Battery Charger IC
Device

Figure 41-8. USB battery charging subsystem

Table 41-13. USB battery charger subsystem components

Component	Description
Battery Charger IC	The external battery charger IC regulates the charge rate to the rechargable battery. System software is responsible for communicating the appropriate charge rates. <table><tr><th>Charger</th><th>Maximum current drawn[1]</th></tr><tr><td>Standard host port</td><td>up to 500 mA</td></tr><tr><td>Charging host port</td><td>up to 1500 mA</td></tr><tr><td>Dedicated charging port</td><td>up to 1800 mA</td></tr></table> 1. If the USB host has suspended the USB device, system software must configure the system to limit the current drawn from the USB bus to 2.5 mA or less.
Comm Module	A communications module on the device can be used to control the charge rate of the battery charger IC.
System software	Coordinates the detection activities of the subsystem.
USB Controller	The D+ pullup enable control signal plays a role during the charger type detection phase. System software must issue a command to the USB controller to assert this signal. After this pullup is enabled, the device is considered to be connected to the USB bus. The host then attempts to enumerate it. **NOTE:** The USB controller must be used only for USB device applications when using the USBDCD module. For USB host applications, the USBDCD module must be disabled.

Table continues on the next page...

Table 41-13. USB battery charger subsystem components (continued)

Component	Description
USB Transceiver	The USB transceiver contains the pullup resistor for the USB D+ signal and the pulldown resistors for the USB D+ and D– signals. The D+ pullup and the D– pulldown are both used during the charger detection sequence. The USB transceiver also outputs the digital state of the D+ and D– signals from the USB bus. The pullup and pulldown enable signals are controlled by other modules during the charger detection sequence. The D+ pullup enable is output from the USB controller and is under software control. The USBDCD module controls the D–pulldown enable.
USBDCD Module	Detects whether the device has been plugged into either a standard host port, a charging host port, or a dedicated charger.
VBUS_detect	This interrupt pin connected to the USB VBUS signal detects when the device has been plugged into or unplugged from the USB bus. If the system requires waking up from a low power mode on being plugged into the USB port, this interrupt should also be a low power wake up source. If this pin multiplexes other functions, such as GPIO, the pin can be configured as an interrupt so that the USB plug or unplug event can be detected.

1. If the USB host has suspended the USB device, system software must configure the system to limit the current drawn from the USB bus to 2.5 mA or less.

41.5.1 The charger detection sequence

The following figure illustrates the charger detection sequence in a simplified timing diagram based on the USB Battery Charging Specification v1.1.

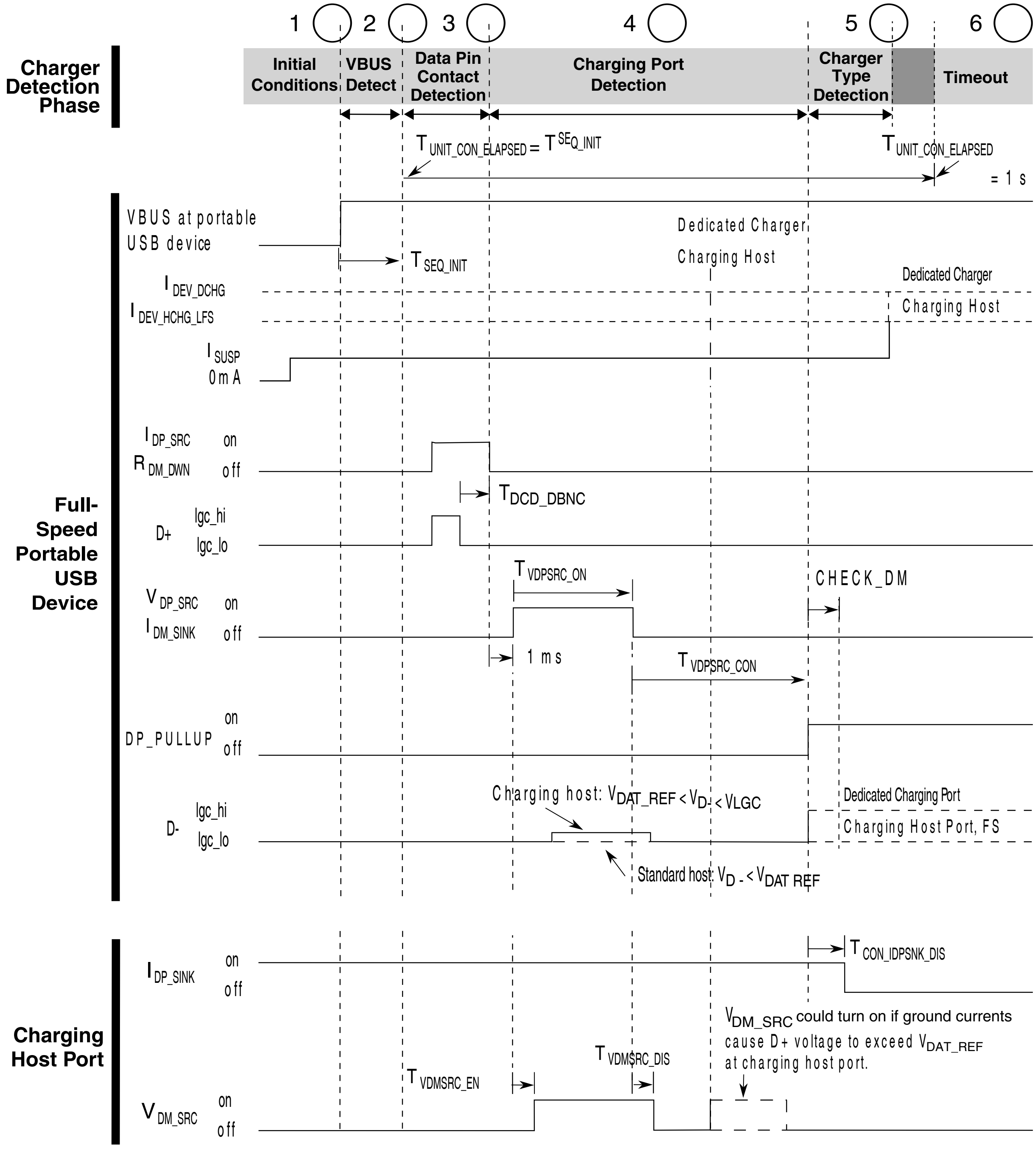

1
2
3
4
5
6
Charger Detection Phase
Initial Conditions
VBUS Detect
Data Pin Contact Detection
Charging Port Detection
Charger Type Detection
Timeout
T UNIT_CON_ELAPSED = T SEQ_INIT
T UNIT_CON_ELAPSED = 1 s
Full-Speed Portable USB Device
VBUS at portable USB device
T SEQ_INIT
Dedicated Charger
Charging Host
I DEV_DCHG
I DEV_HCHG_LFS
Dedicated Charger
Charging Host
I SUSP
0mA
I DP_SRC
R DM_DWN
on
off
T DCD_DBNC
D+
lgc_hi
lgc_lo
T VDPSRC_ON
CHECK_DM
V DP_SRC
I DM_SINK
on
off
1 ms
T VDPSRC_CON
DP_PULLUP
on
off
Charging host: VDAT_REF < VD- < VLGC
Dedicated Charging Port
Charging Host Port, FS
D-
lgc_hi
lgc_lo
Standard host: VD - < VDAT REF
Charging Host Port
I DP_SINK
on
off
T CON_IDPSNK_DIS
VDM_SRC could turn on if ground currents cause D+ voltage to exceed VDAT_REF at charging host port.
T VDMSRC_EN
T VDMSRC_DIS
V DM_SRC
on
off

Figure 41-9. Full speed charger detection timing

The following table provides an overview description of the charger detection sequence shown in the preceding figure.

Table 41-14. Overview of the charger detection sequence

Phase		Overview description	Full description
1	Initial Conditions	Initial system conditions that need to be met before the detection sequence is initiated.	Initial System Conditions
2	VBUS Detection	System software detects contact of the VBUS signal with the system interrupt pin VBUS_detect.	VBUS contact detection
3	Data Pin Contact Detection	The USBDCD module detects that the USB data pins D+ and D– have made contact with the USB port.	Data pin contact detection
4	Charging Port Detection	The USBDCD module detects if the port is a standard host or either type of charging port, that is charging host or dedicated charger.	Charging port detection
5	Charger Type Detection	The USBDCD module detects the type of charging port, if applicable.	Charger type detection
6	Sequence Timeout	The USBDCD module did not finish the detection sequence within the timeout interval. The sequence will continue until halted by software.	Charger detection sequence timeout

Timing parameter values used in this module are listed in the following table.

Table 41-15. Timing parameters for the charger detection sequence

Parameter	USB Battery Charging Spec	Module default	Module programmable range
T_{DCD_DBNC}[1]	10 ms min (no max)	10 ms	0– 1023 ms
T_{VDPSRC_ON}[1]	40 ms min (no max)	40 ms	0 –1023 ms
T_{VDPSRC_CON}[1]	40 ms min (no max)	40 ms	0 –1023 ms
CHECK_DM	N/A	1 ms	0– 15 ms
T_{SEQ_INIT}	N/A	16 ms	0 –1023 ms
T_{UNIT_CON}[1]	1 s	N/A	N/A
T_{VDMSRC_EN}[1]	1– 20 ms	From the USB host	N/A
T_{VDMSRC_DIS}[1]	0 –20 ms	From the USB host	N/A
$T_{CON\_IDPSINK\_DIS}$[1]	0– 20 ms	From the USB host	N/A

[1] This parameter is defined by the *USB Battery Charging Specification, v1.1*.

41.5.1.1 Initial System Conditions

The USBDCD module can be used only with FS USB device applications using a rechargable battery. That is, it cannot be used with USB applications that are HS, LS, host, or OTG. In addition, before the USBDCD module's charger detection sequence can be initiated, the system must be:

- Powered-up and in run mode.

- Recently plugged into a USB port.
- Drawing not more than 2.5 mA total system current from the USB bus.

Examples of allowable precursors to this set of initial conditions include:

- A powered-down device is subsequently powered-up upon being plugged into the USB bus.
- A device in a low power mode subsequently enters run mode upon being plugged into the USB bus.

41.5.1.2 VBUS contact detection

Once the device is plugged into a USB port, the VBUS_detect system interrupt is triggered. System software must do the following to initialize the module and start the charger detection sequence:

1. Restore power if the module is powered-off.
2. Set CONTROL[SR] to initiate a software reset.
3. Configure the USBDCD module by programming the CLOCK register and the timing parameters as needed.
4. Set CONTROL[IE] to enable interrupts, or clear the bit if software polling method is used .
5. Set CONTROL[START] to start the charger detection sequence.

41.5.1.3 Data pin contact detection

The module must ensure that the data pins have made contact because the detection sequence depends upon the state of the USB D+ signal. USB plugs and receptables are designed such that when the plug is inserted into the receptable, the power pins make contact before the data pins make contact. See the following figure.

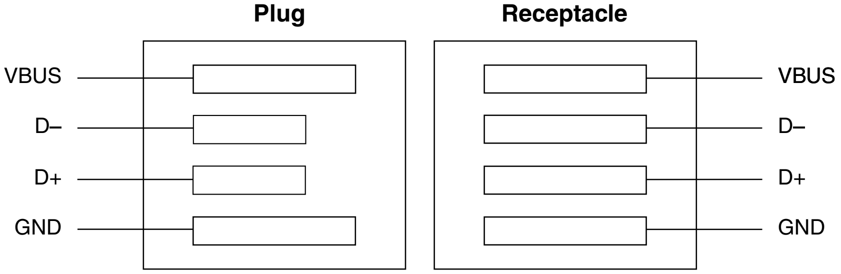


Figure 41-10. Relative pin positions in USB plugs and receptacles

As a result, when a portable USB device is attached to an upstream port, the portable USB device detects VBUS before the data pins have made contact. The time between power pins and data pins making contact depends on how fast the plug is inserted into the receptable. Delays of several hundred milliseconds are possible.

41.5.1.3.1 Debouncing the data pin contact

When system software has initiated the charger detection sequence, as described in Initial System Conditions, the USBDCD module turns on the I_{DP_SRC} current source and enables the R_{DM_DWN} pulldown resistor. If the data pins have not made contact, the D+ line remains high. After the data pins make contact, the D+ line goes low and debouncing begins.

After the D+ line goes low, the module continuously samples the D+ line over the duration of the T_{DCD_DBNC} debounce time interval.By deafult, T_{DCD_DBNC} is 10 ms, but it can be programmed in the TIMER0[TDCD_DBNC] field. See the description of the TIMER0 Register for register information.

When it has remained low for the entire interval, the debouncing is complete. However, if the D+ line returns high during the debounce interval, the module waits until the D+ line goes low again to restart the debouncing. This cycle repeats until either of the following happens:

- The data pin contact has been successfully debounced (see Success in detecting data pin contact (phase completion)).
- A timeout occurs (see Charger detection sequence timeout).

41.5.1.3.2 Success in detecting data pin contact (phase completion)

After successfully debouncing the D+ state, the module does the following:

- Updates the STATUS register to reflect phase completion (See Table 41-18 for field values.)
- Directly proceeds to the next step in the sequence: detection of a charging port (See Charging port detection.)

41.5.1.4 Charging port detection

After it detects that the data pins have made contact, the module waits for a fixed delay of 1 ms, and then attempts to detect whether it is plugged into a charging port. The module connects the following analog units to the USB D+ or D– lines during this phase:

- The voltage source V_{DP_SRC} connects to the D+ line
- The current sink I_{DM_SINK} connects to the D– line
- The voltage comparator connects to the USB D– line, comparing it to the voltage V_{DAT_REF}.

After a time of T_{VDPSRC_ON}, the module samples the D– line. The T_{VDPSRC_ON} parameter is programmable and defaults to 40 ms. After sampling the D– line, the module disconnects the voltage source, current sink, and comparator.

The next steps in the sequence depend on the voltage on the D– line as determined by the voltage comparator. See the following table.

Table 41-16. Sampling D– in the charging port detection phase

If the voltage on D- is...	Then...	See...
Below V_{DAT_REF}	The port is a *standard host* that does not support the USB Battery Charging Specification v1.1.	Standard host port
Above V_{DAT_REF} but below V_{LGC}	The port is a *charging port*.	Charging port
Above V_{LGC}	This is an error condition.	Error in charging port detection

41.5.1.4.1 Standard host port

As part of the charger detection handshake with a standard USB host, the module does the following without waiting for the T_{VDPSRC_CON} interval to elapse:

- Updates the STATUS register to reflect that a standard host has been detected with SEQ_RES = 01. See Table 41-18 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE].

At this point, control has been passed to system software via the interrupt. The rest of the sequence, which detects the type of charging port, is not applicable, so software should perform the following steps:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Set CONTROL[SR] to issue a software reset to the module.
4. Disable the module.
5. Communicate the appropriate charge rate to the external battery charger IC; see Table 41-13.

41.5.1.4.2 Charging port

As part of the charger detection handshake with any type of USB host, the module waits until the T_{VDPSRC_CON} interval has elapsed before it does the following:

- Updates the STATUS register to reflect that a charging port has been detected with SEQ_RES = 10. See Table 41-18 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE].

At this point, control has passed to system software via the interrupt. Software should:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Issue a command to the USB controller to pullup the USB D+ line.
4. Wait for the module to complete the final phase of the sequence. See Charger type detection.

41.5.1.4.3 Error in charging port detection

For this error condition, the module does the following:

- Updates the STATUS register to reflect the error with SEQ_RES = 00. See Table 41-18 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE].

Note that in this case the module does not wait for the T_{VDPSRC_CON} interval to elapse.

At this point, control has been passed to system software via the interrupt. The rest of the sequence (detecting the type of charging port) is not applicable, so software should:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Set CONTROL[SR] to issue a software reset to the module.
4. Disable the module.

41.5.1.5 Charger type detection

After software enables the D+ pullup resistor, the module is notified automatically (via internal signaling) to start the CHECK_DM timer counting down the time interval programmed in the TIMER2[CHECK_DM] field.

After the CHECK_DM time has elapsed, the module samples the USB D– line to determine the type of charger. See the following table.

Table 41-17. Sampling D– in the charger type detection phase

If the voltage on D– is...	Then...	See...
High	The port is a *dedicated charging port*.[1]	Dedicated charging port
Low	The port is a *charging host port*.[2]	Charging host port

1. In a dedicated charger, the D+ and D– lines are shorted together through a small resistor.
2. In a charging host port, the D+ and D– lines are not shorted.

41.5.1.5.1 Dedicated charging port

For a dedicated charger, the module does the following:

- Updates the STATUS register to reflect that a dedicated charger has been detected with SEQ_RES = 11. See Table 41-18 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE] bit.

At this point, control has been passed to system software via the interrupt. Software should:

1. Read the STATUS register.

2. Disable the USB controller to prevent transitions on the USB D+ or D– lines from causing spurious interrupt or wakeup events to the system.
3. Set CONTROL[IACK] to acknowledge the interrupt.
4. Set CONTROL[SR] to issue a software reset to the module.
5. Disable the module.
6. Communicate the appropriate charge rate to the external battery charger IC; see Table 41-13.

41.5.1.5.2 Charging host port

For a charging host port, the module does the following:

- Updates the STATUS register to reflect that a charging host port has been detected with SEQ_RES = 10. See Table 41-18 for field values.
- Sets CONTROL[IF].
- Generates an interrupt if enabled in CONTROL[IE].

At this point, control has been passed to system software via the interrupt. Software should:

1. Read the STATUS register.
2. Set CONTROL[IACK] to acknowledge the interrupt.
3. Set CONTROL[SR] to issue a software reset to the module.
4. Disable the module.
5. Communicate the appropriate charge rate to the external battery charger IC; see Table 41-13.

41.5.1.6 Charger detection sequence timeout

The maximum time allowed to connect according to the *USB Battery Charging Specification, v1.1* is one second. If the Unit Connection Timer reaches the one second limit and the sequence is still running as indicated by the STATUS[ACTIVE] bit, the module does the following:

- Updates the STATUS register to reflect that a timeout error has occured. See Table 41-18 for field values.
- Sets the CONTROL[IF] bit.
- Generates an interrupt if enabled in CONTROL[IE].
- The detection sequence continues until explicitly halted by software setting the CONTROL[SR] bit.
- The Unit Connection Timer continues counting. See the description of the TIMER0 Register.

At this point, control has been passed to system software via the interrupt, which has two options: ignore the interrupt and allow more time for the sequence to complete, or halt the sequence. To halt the sequence, software should:

1. Read the STATUS register.
2. Set the CONTROL[IACK] bit to acknowledge the interrupt.
3. Set the CONTROL[SR] bit to issue a software reset to the module.
4. Disable the module.

This timeout function is also useful in case software does not realize that the USB device is unplugged from USB port during the charger detection sequence. If the interrupt occurs but the V_{BUS_DETECT} input is low, software can disable and reset the module.

System software might allow the sequence to run past the timeout interrupt under these conditions:

1. The USB Battery Charging Spec is amended to allow more time. In this case, software should poll TIMER0[$T_{UNITCON}$] periodically to track elapsed time after 1s; or
2. For debug purposes.

Note that the $T_{UNITCON}$ register field will stop incrementing when it reaches its maximum value so it will not rollover to zero and start counting up again.

41.5.2 Interrupts and events

The USBDCD module has an interrupt to alert system software of certain events, which are listed in the following table. All events except the Phase Complete event for the Data Pin Detection phase can trigger an interrupt.

Table 41-18. Events triggering an interrupt by sequence phase

Sequence phase	Event	Event description	STATUS fields[1]	Phase description
Data Pin Detection	Phase Complete	The module has detected data pin contact. *No interrupt occurs*: CONTROL[IF] = 0.	ERR = 0 SEQ_STAT = 01 SEQ_RES = 00 TO = 0	VBUS contact detection
Charging Port Detection	Phase Complete	The module has completed the process of identifying if the USB port is a charging port or not.	ERR = 0 SEQ_STAT = 10 SEQ_RES = 01 or 10 TO = 0	Charging port detection
	Error	The module cannot identify the type of port because the D– line is above the USB's VLGC threshold.	ERR = 1 SEQ_STAT = 10 SEQ_RES = 00 TO = 0	Error in charging port detection
Charger Type Detection	Phase Complete	The module has completed the process of identifying the charger type detection. **Note:** The ERR flag always reads zero because no known error conditions are checked during this phase.	ERR = 0 SEQ_STAT = 11 SEQ_RES = 11 or 10 TO = 0	Charger type detection
Sequence Timeout	Error	The timeout interval from the time the USB device attaches to a USB port until it connects has elapsed.	ERR = 1 SEQ_STAT = last value[2] SEQ_RES = last value[2] TO = 1	Charger detection sequence timeout.

1. See the description of the Status register for register information.
2. The SEQ_STAT and SEQ_RES fields retain the values held at the time of the timeout error.

41.5.2.1 Interrupt Handling

Software can read which event caused the interrupt from the STATUS register during the interrupt service routine.

An interrupt is generated only if CONTROL[IE] is set. The CONTROL[IF] bit is always set under interrupt conditions, even if CONTROL[IE] is cleared. In this case, software can poll CONTROL[IF] to determine if an interrupt condition is pending.

Writes to CONTROL[IF] are ignored. To reset CONTROL[IF], set CONTROL[IACK] to acknowledge the interrupt. Writing to CONTROL[IACK] when CONTROL[IF] is cleared has no effect.

41.5.3 Resets

There are two ways to reset various register contents in this module: hardware resets and a software reset.

41.5.3.1 Hardware resets

Hardware resets originate at the system or device level and propagate down to the individual module level. They include start up reset, low-voltage reset, and all other hardware reset sources.

Hardware resets cause the register contents to be restored to their default state as listed in the register descriptions.

41.5.3.2 Software reset

A software reset re-initializes the module's status information, but leaves configuration information unchanged. The software reset allows software to prepare the module without needing to reprogram the same configuration each time the USB device is plugged into a USB port.

Setting CONTROL[SR] initiates a software reset. The following table shows all register fields that are reset to their default values by a software reset.

Table 41-19. Software reset and register fields affected

Register	Fields affected	Fields not affected
CONTROL[1]	IF	IE, START
STATUS	All	None
CLOCK	None	All
TIMER*n*	TUNITCON	All other

[1] CONTROL[SR] and CONTROL[IACK] are self-clearing.

A software reset also returns all internal logic, timers, and counters to their reset states. If the module is already active (STATUS[ACTIVE] = 1), a software reset stops the sequence.

Note

> Software must always initiate a software reset before starting the sequence to ensure the module is in a known state.

41.6 Initialization information

This module has been designed for minimal configuration while retaining significant programmability. The CLOCK register needs to be initialized to the actual system clock frequency, unless the default value already matches the system requirements.

The other registers generally do not need to be modified, because they default to values that comply with the USB Battery Charging Specification v1.1. However, several timing parameters can be changed for a great deal of flexibility if a particular system requires it.

All module configuration must occur *before* initiating the charger detection sequence. Configuration changes made *after* setting CONTROL[START] result in undefined behavior.

41.7 Application information

This section provides application information.

41.7.1 External pullups

Any external pullups applied to the USB D+ or D- data lines must be capable of being disabled to prevent incorrect pullup values or incorrect operation of the USB subsystem.

41.7.2 Dead or weak battery

According to the USB Battery Charging Specification v1.1, a USB device with a dead, weak, or missing battery that is attached to a charging port can remain attached indefinitely drawing up to 1.5A, until the battery is charged to the point that the USB device can connect.

The USBDCD module is compatible with systems that do not check the strength of the battery. Therefore, this module assumes that the battery is good, so the USB device must immediately connect to the USB bus by pulling the D+ line high after the USBDCD module has determined that the device is attached to a charging port.

The module is also compatible with systems that do check the strength of the battery. In these systems, if it is known that the battery is weak or dead, software can delay connecting to the USB while charging at 1.5A. Once the battery is charged to the good battery threshold, software can then connect to the USB host by pulling the D+ line high.

41.7.3 Handling unplug events

If the device is unplugged from the USB bus during the charger detection sequence, the contents of the STATUS register must be ignored and the USBDCD module must get a Software Reset, as described in Software reset.

Chapter 42
USB Voltage Regulator

42.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The USB Voltage Regulator module is a LDO linear voltage regulator to provide 3.3V power from an input power supply varying from 2.7 V to 5.5 V. It consists of one 3.3 V power channel. When the input power supply is below 3.6 V, the regulator goes to pass-through mode. The following figure shows the ideal relation between the regulator output and input power supply.

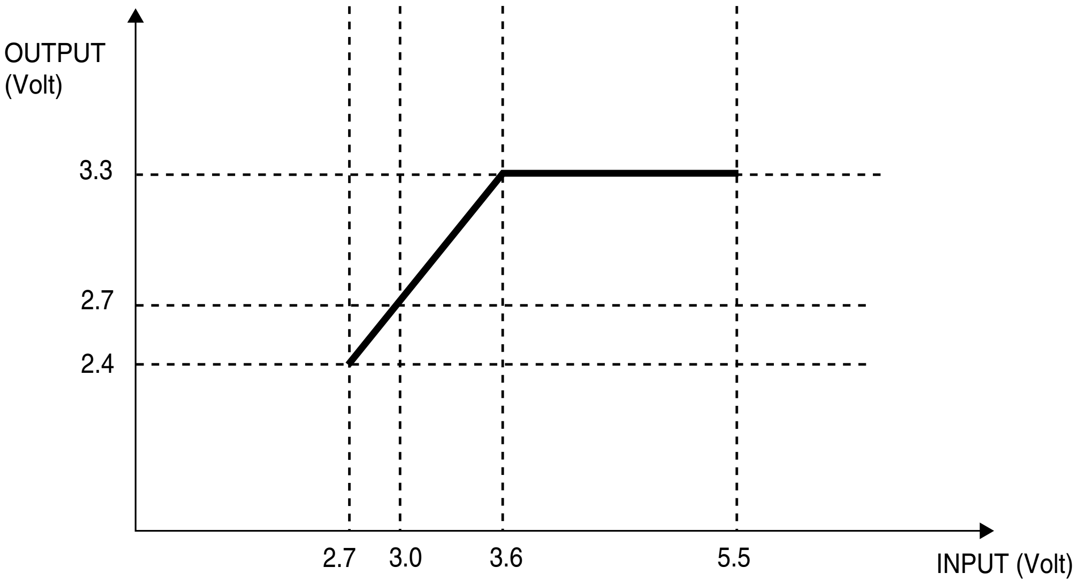


Figure 42-1. Ideal Relation Between the Regulator Output and Input Power Supply

42.1.1 Overview

A simplified block diagram for the USB Voltage Regulator module is shown below.

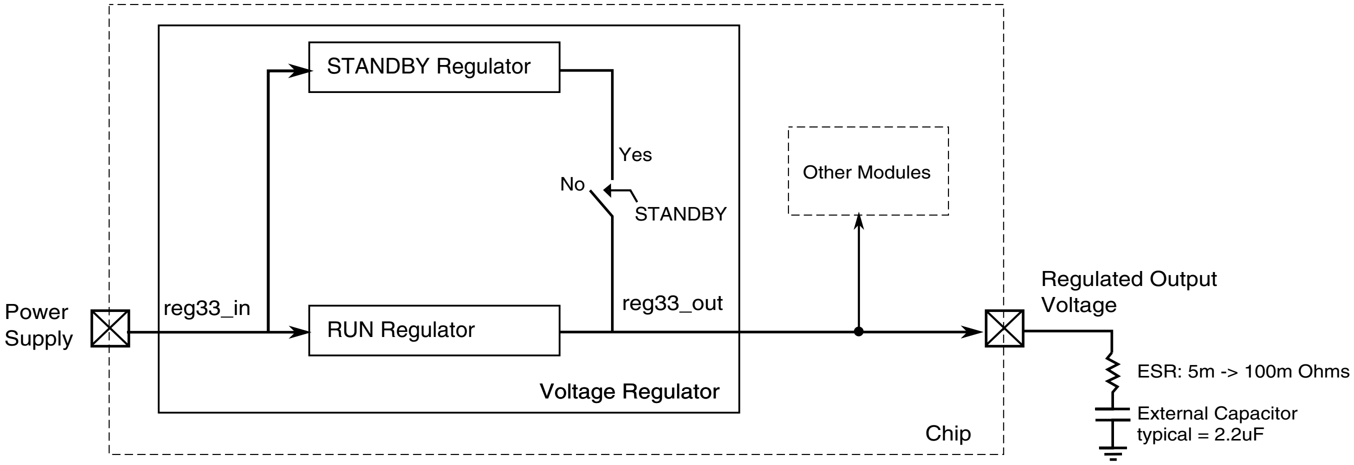

STANDBY Regulator
Yes
No
STANDBY
Other Modules
Power Supply
reg33_in
RUN Regulator
reg33_out
Regulated Output Voltage
ESR: 5m -> 100m Ohms
External Capacitor typical = 2.2uF
Voltage Regulator
Chip

Figure 42-2. USB Voltage Regulator Block Diagram

This module uses 2 regulators in parallel. In run mode, the RUN regulator with the bandgap voltage reference is enabled and can provide up to 120 mA load current. In run mode, the STANDBY regulator and the low power reference are also enabled, but a switch disconnects its output from the external pin. In STANDBY mode, the RUN regulator is disabled and the STANDBY regulator output is connected to the external pin supplying up to 3 mA load current.

Internal power mode signals control whether the module is in RUN or STANDBY mode.

42.1.2 Features

- Low drop-out linear voltage regulator with one power channel (3.3V).
- Low drop-out voltage: 300 mV.
- Output current: 120 mA.
- Three different power modes: RUN, STANDBY and SHUTDOWN.
- Low quiescent current in RUN mode.
  - Typical value is around 120 uA (one thousand times smaller than the maximum load current).
- Very low quiescent current in STANDBY mode.
  - Typical value is around 1 uA.

- Automatic current limiting if the load current is greater than 290 mA.
- Automatic power-up once some voltage is applied to the regulator input.
- Pass-through mode for regulator input voltages less than 3.6 V
- Small output capacitor: 2.2 uF
- Stable with aluminum, tantalum or ceramic capacitors.

42.1.3 Modes of Operation

The regulator has these power modes:

- RUN—The regulating loop of the RUN regulator and the STANDBY regulator are active, but the switch connecting the STANDBY regulator output to the external pin is open.
- STANDBY—The regulating loop of the RUN regulator is disabled and the standby regulator is active. The switch connecting the STANDBY regulator output to the external pin is closed.
- SHUTDOWN—The module is disabled.

The regulator is enabled by default. This means that once the power supply is provided, the module power-up sequence to RUN mode starts.

42.2 USB Voltage Regulator Module Signal Descriptions

The following table shows the external signals for the regulator.

Table 42-1. USB Voltage Regulator Module Signal Descriptions

Signal	Description	I/O
reg33_in	Unregulated power supply	I
reg33_out	Regulator output voltage	O

Chapter 43
SPI (DSPI)

43.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The serial peripheral interface (SPI) module provides a synchronous serial bus for communication between an MCU and an external peripheral device.

43.1.1 Block Diagram

The block diagram of this module is as follows:

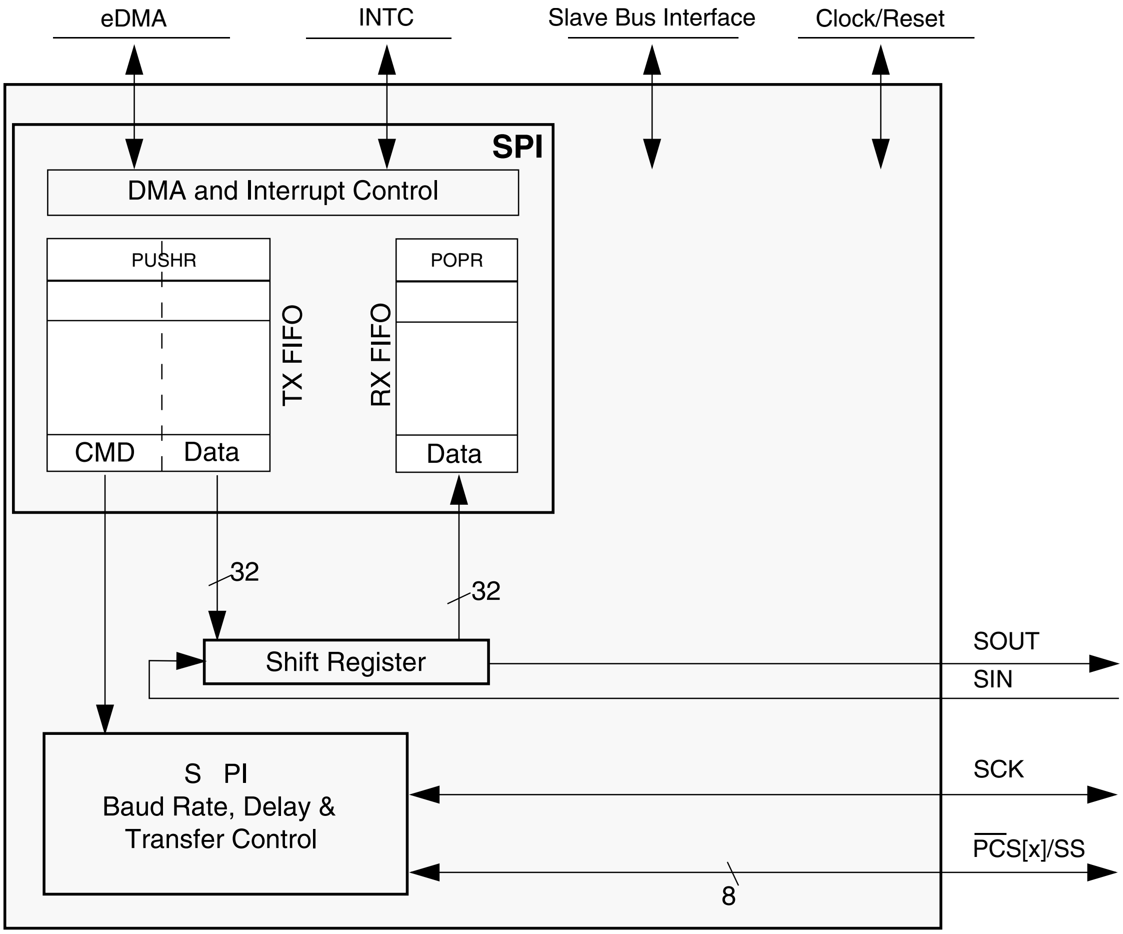

eDMA
INTC
Slave Bus Interface
Clock/Reset
SPI
DMA and Interrupt Control
PUSHR
POPR
TX FIFO
RX FIFO
CMD
Data
Data
32
32
Shift Register
SOUT
SIN
SCK
S PI
Baud Rate, Delay &
Transfer Control
PCS[x]/SS
8

Figure 43-1. DSPI Block Diagram

43.1.2 Features

The DSPI supports the following SPI features:

- Full-duplex, four-wire synchronous transfers
- Master and Slave modes:
    - Data streaming operation in Slave mode with continuous slave selection
- Buffered transmit operation using the transmit first in first out (TX FIFO) with depth of 4 entries
- Buffered receive operation using the receive FIFO (RX FIFO) with depth of 4 entries
- TX and RX FIFOs can be disabled individually for low-latency updates to SPI queues
- Visibility into TX and RX FIFOs for ease of debugging

- Programmable transfer attributes on a per-frame basis:
    - 2 transfer attribute registers
    - Serial clock (SCK) with programmable polarity and phase
    - Various programmable delays
    - Programmable serial frame size of 4–16 bits, expandable by software control
        - SPI frames longer than 16 bits can be supported using the continuous selection format
    - Continuously held chip select capability
- 5 peripheral chip selects (PCSs), expandable to 32 with external demultiplexer
- Deglitching support for up to 16 PCS with external demultiplexer
- DMA support for adding entries to TX FIFO and removing entries from RX FIFO:
    - TX FIFO is not full (TFFF)
    - RX FIFO is not empty (RFDF)
- Interrupt conditions:
    - End of Queue reached (EOQF)
    - TX FIFO is not full (TFFF)
    - Transfer of current frame complete (TCF)
    - Attempt to transmit with an empty Transmit FIFO (TFUF)
    - RX FIFO is not empty (RFDF)
    - Frame received while Receive FIFO is full (RFOF)
- Global interrupt request line
- Modified SPI transfer formats for communication with slower peripheral devices
- Power-saving architectural features:
    - Support for Stop mode
    - Support for Doze mode

43.1.3 DSPI Configurations

The DSPI module always operates in SPI configuration.

43.1.3.1 SPI Configuration

The SPI configuration allows the DSPI to send and receive serial data. This configuration allows the DSPI to operate as a basic SPI block with internal FIFOs supporting external queue operation. Transmitted data and received data reside in separate FIFOs. The host CPU or a DMA controller read the received data from the Receive FIFO and write transmit data to the Transmit FIFO.

For queued operations, the SPI queues can reside in system RAM, external to the DSPI. Data transfers between the queues and the DSPI FIFOs are accomplished by a DMA controller or host CPU. The following figure shows a system example with DMA, DSPI, and external queues in system RAM.

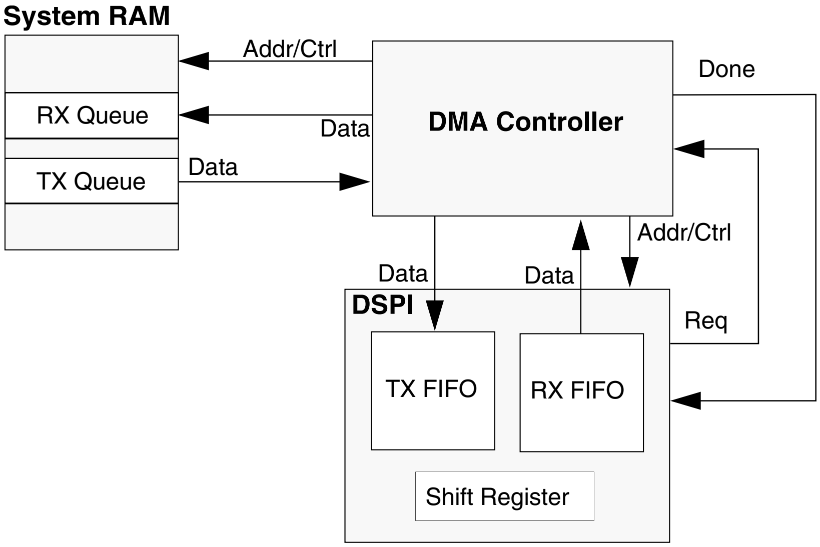

System RAM
Addr/Ctrl
RX Queue
Data
TX Queue
Data
DMA Controller
Done
Addr/Ctrl
Data
Data
DSPI
Req
TX FIFO
RX FIFO
Shift Register

Figure 43-2. DSPI with queues and DMA

43.1.4 Modes of Operation

The DSPI supports the following modes of operation that can be divided into two categories:

- Module-specific modes:
  - Master mode

  - Slave mode
  - Module Disable mode
- MCU-specific modes:
  - External Stop mode
  - Debug mode

The DSPI enters module-specific modes when the host writes a DSPI register. The MCU-specific modes are controlled by signals external to the DSPI. The MCU-specific modes are modes that an MCU may enter in parallel to the DSPI block-specific modes.

43.1.4.1 Master Mode

Master mode allows the DSPI to initiate and control serial communication. In this mode, the SCK signal and the PCS[*x*] signals are controlled by the DSPI and configured as outputs.

43.1.4.2 Slave Mode

Slave mode allows the DSPI to communicate with SPI bus masters. In this mode, the DSPI responds to externally controlled serial transfers. The SCK signal and the PCS[0]/$\overline{\text{SS}}$ signals are configured as inputs and driven by an SPI bus master.

43.1.4.3 Module Disable Mode

The Module Disable mode can be used for MCU power management. The clock to the non-memory mapped logic in the DSPI can be stopped while in the Module Disable mode.

43.1.4.4 External Stop Mode

External Stop mode is used for MCU power management. The DSPI supports the Peripheral Bus Stop mode mechanism. When a request is made to enter External Stop mode, the DSPI block acknowledges the request and completes the transfer that is in progress. When the DSPI reaches the frame boundary, it signals that the system clock to the DSPI module may be shut off.

43.1.4.5 Debug Mode

Debug mode is used for system development and debugging. The MCR[FRZ] bit controls DSPI behavior in the Debug mode:

- If the bit is set, the DSPI stops all serial transfers, when the MCU is in debug mode.
- If the bit is cleared, the MCU debug mode has no effect on the DSPI.

43.2 DSPI signal descriptions

This section provides description of the DSPI signals.

The following table lists the signals that may connect off chip depending on device implementation.

Table 43-1. DSPI signal descriptions

Signal	Description	I/O
PCS0/$\overline{\text{SS}}$	Master mode: Peripheral Chip Select 0 output Slave mode: Slave Select input	I/O
PCS[3:1]	Master mode: Peripheral Chip Select 1 – 3 Slave mode: Unused	O
PCS4	Master mode: Peripheral Chip Select 4 Slave mode: Unused	O
SIN	Serial Data In	I
SOUT	Serial Data Out	O
SCK	Master mode: Serial Clock (output) Slave mode: Serial Clock (input)	I/O

43.2.1 PCS0/$\overline{\text{SS}}$ — Peripheral Chip Select/Slave Select

In Master mode, the PCS0 signal is an output that selects which slave device the current transmission is intended for.

In Slave mode, the active low $\overline{\text{SS}}$ signal is an input signal that allows an SPI master to select the DSPI as the target for transmission.

43.2.2 PCS1 – PCS3 — Peripheral Chip Selects 1 – 3

PCS1 – PCS3 are output signals in Master mode.

In Slave mode, these signals are unused.

43.2.3 PCS4 — Peripheral Chip Select 4

In Master mode, PCS4 is an output signal.

In Slave mode, this signal is unused.

43.2.4 SIN — Serial Input

SIN is a serial data input signal.

43.2.5 SOUT — Serial Output

SOUT is a serial data output signal.

43.2.6 SCK — Serial Clock

SCK is a serial communication clock signal. In Master mode, the DSPI generates the SCK. In Slave mode, SCK is an input from an external bus master.

43.3 Memory Map/Register Definition

Register accesses to memory addresses that are reserved or undefined result in a transfer error. Write access to the POPR also results in a transfer error.

SPI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_C000	DSPI Module Configuration Register (SPI0_MCR)	32	R/W	0000_4001h	43.3.1/ 965

Table continues on the next page...

SPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_C008	DSPI Transfer Count Register (SPI0_TCR)	32	R/W	0000_0000h	43.3.2/ 968
4002_C00C	DSPI Clock and Transfer Attributes Register (In Master Mode) (SPI0_CTAR0)	32	R/W	7800_0000h	43.3.3/ 968
4002_C00C	DSPI Clock and Transfer Attributes Register (In Slave Mode) (SPI0_CTAR0_SLAVE)	32	R/W	7800_0000h	43.3.4/ 973
4002_C010	DSPI Clock and Transfer Attributes Register (In Master Mode) (SPI0_CTAR1)	32	R/W	7800_0000h	43.3.3/ 968
4002_C02C	DSPI Status Register (SPI0_SR)	32	R/W	0201_0000h	43.3.5/ 974
4002_C030	DSPI DMA/Interrupt Request Select and Enable Register (SPI0_RSER)	32	R/W	0000_0000h	43.3.6/ 977
4002_C034	DSPI PUSH TX FIFO Register In Master Mode (SPI0_PUSHR)	32	R/W	0000_0000h	43.3.7/ 979
4002_C034	DSPI PUSH TX FIFO Register In Slave Mode (SPI0_PUSHR_SLAVE)	32	R/W	0000_0000h	43.3.8/ 981
4002_C038	DSPI POP RX FIFO Register (SPI0_POPR)	32	R	0000_0000h	43.3.9/ 981
4002_C03C	DSPI Transmit FIFO Registers (SPI0_TXFR0)	32	R	0000_0000h	43.3.10/ 982
4002_C040	DSPI Transmit FIFO Registers (SPI0_TXFR1)	32	R	0000_0000h	43.3.10/ 982
4002_C044	DSPI Transmit FIFO Registers (SPI0_TXFR2)	32	R	0000_0000h	43.3.10/ 982
4002_C048	DSPI Transmit FIFO Registers (SPI0_TXFR3)	32	R	0000_0000h	43.3.10/ 982
4002_C07C	DSPI Receive FIFO Registers (SPI0_RXFR0)	32	R	0000_0000h	43.3.11/ 983
4002_C080	DSPI Receive FIFO Registers (SPI0_RXFR1)	32	R	0000_0000h	43.3.11/ 983
4002_C084	DSPI Receive FIFO Registers (SPI0_RXFR2)	32	R	0000_0000h	43.3.11/ 983
4002_C088	DSPI Receive FIFO Registers (SPI0_RXFR3)	32	R	0000_0000h	43.3.11/ 983

43.3.1 DSPI Module Configuration Register (SPIx_MCR)

Contains bits to configure various attributes associated with DSPI operations. The HALT and MDIS bits can be changed at any time, but the effect takes place only on the next frame boundary. Only the HALT and MDIS bits in the MCR can be changed, while the DSPI is in the Running state.

Addresses: SPI0_MCR is 4002_C000h base + 0h offset = 4002_C000h

Bit	31	30	29	28	27	26	25	24
R W	MSTR	CONT_SCKE	DCONF		FRZ	MTFE	Reserved	ROOE
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R W	Reserved			PCSIS[4:0]				
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	DOZE	MDIS	DIS_TXF	DIS_RXF	0	0	SMPL_PT	
W					CLR_TXF	CLR_RXF		
Reset	0	1	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	0					0	0	HALT
W								
Reset	0	0	0	0	0	0	0	1

SPIx_MCR field descriptions

Field	Description
31 MSTR	Master/Slave Mode Select Configures the DSPI for either Master mode or Slave mode. 0 DSPI is in Slave mode. 1 DSPI is in Master mode.
30 CONT_SCKE	Continuous SCK Enable Enables the Serial Communication Clock (SCK) to run continuously. 0 Continuous SCK disabled. 1 Continuous SCK enabled.
29–28 DCONF	DSPI Configuration Selects among the different configurations of the DSPI. 00 SPI 01 Reserved

Table continues on the next page...

SPIx_MCR field descriptions (continued)

Field	Description
	10 Reserved 11 Reserved
27 FRZ	Freeze Enables the DSPI transfers to be stopped on the next frame boundary when the device enters Debug mode. 0 Do not halt serial transfers in Debug mode. 1 Halt serial transfers in Debug mode.
26 MTFE	Modified Timing Format Enable Enables a modified transfer format to be used. 0 Modified SPI transfer format disabled. 1 Modified SPI transfer format enabled.
25 Reserved	This field is reserved.
24 ROOE	Receive FIFO Overflow Overwrite Enable In the RX FIFO overflow condition, configures the DSPI to ignore the incoming serial data or overwrite existing data. If the RX FIFO is full and new data is received, the data from the transfer, generating the overflow, is ignored or shifted into the shift register. 0 Incoming data is ignored. 1 Incoming data is shifted into the shift register.
23–21 Reserved	This field is reserved.
20–16 PCSIS[4:0]	Peripheral Chip Select x Inactive State Determines the inactive state of PCSx. 0 The inactive state of PCSx is low. 1 The inactive state of PCSx is high.
15 DOZE	Doze Enable Provides support for an externally controlled Doze mode power-saving mechanism. 0 Doze mode has no effect on DSPI. 1 Doze mode disables DSPI.
14 MDIS	Module Disable Allows the clock to be stopped to the non-memory mapped logic in the DSPI effectively putting the DSPI in a software-controlled power-saving state. The reset value of the MDIS bit is parameterized, with a default reset value of 0. 0 Enables DSPI clocks. 1 Allows external logic to disable DSPI clocks.
13 DIS_TXF	Disable Transmit FIFO

Table continues on the next page...

SPIx_MCR field descriptions (continued)

Field	Description
	When the TX FIFO is disabled, the transmit part of the DSPI operates as a simplified double-buffered SPI. This bit can be written only when the MDIS bit is cleared. 0 TX FIFO is enabled. 1 TX FIFO is disabled.
12 DIS_RXF	Disable Receive FIFO When the RX FIFO is disabled, the receive part of the DSPI operates as a simplified double-buffered SPI. This bit can only be written when the MDIS bit is cleared. 0 RX FIFO is enabled. 1 RX FIFO is disabled.
11 CLR_TXF	Clear TX FIFO Flushes the TX FIFO. Writing a 1 to CLR_TXF clears the TX FIFO Counter. The CLR_TXF bit is always read as zero. 0 Do not clear the TX FIFO counter. 1 Clear the TX FIFO counter.
10 CLR_RXF	Flushes the RX FIFO. Writing a 1 to CLR_RXF clears the RX Counter. The CLR_RXF bit is always read as zero. 0 Do not clear the RX FIFO counter. 1 Clear the RX FIFO counter.
9–8 SMPL_PT	Sample Point Controls when the DSPI master samples SIN in Modified Transfer Format. This field is valid only when CPHA bit in CTARn[CPHA] is 0. 00 0 system clocks between SCK edge and SIN sample 01 1 system clock between SCK edge and SIN sample 10 2 system clocks between SCK edge and SIN sample 11 Reserved
7–3 Reserved	This read-only field is reserved and always has the value zero.
2 Reserved	This read-only field is reserved and always has the value zero.
1 Reserved	This read-only field is reserved and always has the value zero.
0 HALT	Halt Starts and stops DSPI transfers. 0 Start transfers. 1 Stop transfers.

43.3.2 DSPI Transfer Count Register (SPIx_TCR)

TCR contains a counter that indicates the number of SPI transfers made. The transfer counter is intended to assist in queue management. Do not write the TCR when the DSPI is in the Running state.

Addresses: SPI0_TCR is 4002_C000h base + 8h offset = 4002_C008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SPI_TCNT																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_TCR field descriptions

Field	Description
31–16 SPI_TCNT	SPI Transfer Counter Counts the number of SPI transfers the DSPI makes. The SPI_TCNT field increments every time the last bit of an SPI frame is transmitted. A value written to SPI_TCNT presets the counter to that value. SPI_TCNT is reset to zero at the beginning of the frame when the CTCNT field is set in the executing SPI command. The Transfer Counter wraps around; incrementing the counter past 65535 resets the counter to zero.
15–0 Reserved	This read-only field is reserved and always has the value zero.

43.3.3 DSPI Clock and Transfer Attributes Register (In Master Mode) (SPIx_CTAR*n*)

CTARs are used to define different transfer attributes. The number of the CTARs is parameterized in the RTL and can be from two to eight registers. Do not write to the CTARs while the DSPI is in the Running state.

In Master mode, the CTARs define combinations of transfer attributes such as frame size, clock phase and polarity, data bit ordering, baud rate, and various delays. In Slave mode, a subset of the fields in CTAR0 are used to set the slave transfer attributes.

When the DSPI is configured as an SPI master, the CTAS field in the command portion of the TX FIFO entry selects which of the CTAR register is used. When the DSPI is configured as an SPI bus slave, the CTAR0 is used.

Addresses: SPI0_CTAR0 is 4002_C000h base + Ch offset = 4002_C00Ch
SPI0_CTAR1 is 4002_C000h base + 10h offset = 4002_C010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DBR	FMSZ				CPOL	CPHA	LSBFE	PCSSCK		PASC		PDT		PBR	
Reset	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSSCK				ASC				DT				BR			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_CTAR*n* field descriptions

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>31
DBR</td><td>Double Baud Rate

Doubles the effective baud rate of the Serial Communications Clock (SCK). This field is used only in Master mode. It effectively halves the Baud Rate division ratio, supporting faster frequencies, and odd division ratios for the SCK. When the DBR bit is set, the duty cycle of the SCK depends on the value in the Baud Rate Prescaler and the Clock Phase bit as listed in the following table. See the BR field description for details on how to compute the baud rate.

Table 43-32. DSPI SCK duty cycle
<table>
<tr><th>DBR</th><th>CPHA</th><th>PBR</th><th>SCK duty cycle</th></tr>
<tr><td>0</td><td>any</td><td>any</td><td>50/50</td></tr>
<tr><td>1</td><td>0</td><td>00</td><td>50/50</td></tr>
<tr><td>1</td><td>0</td><td>01</td><td>33/66</td></tr>
<tr><td>1</td><td>0</td><td>10</td><td>40/60</td></tr>
<tr><td>1</td><td>0</td><td>11</td><td>43/57</td></tr>
<tr><td>1</td><td>1</td><td>00</td><td>50/50</td></tr>
<tr><td>1</td><td>1</td><td>01</td><td>66/33</td></tr>
<tr><td>1</td><td>1</td><td>10</td><td>60/40</td></tr>
<tr><td>1</td><td>1</td><td>11</td><td>57/43</td></tr>
</table>

0 The baud rate is computed normally with a 50/50 duty cycle.
1 The baud rate is doubled with the duty cycle depending on the Baud Rate Prescaler.</td></tr>
<tr><td>30–27
FMSZ</td><td>Frame Size

The number of bits transferred per frame is equal to the FMSZ field value plus 1. The minimum valid FMSZ field value is 3.</td></tr>
<tr><td>26
CPOL</td><td>Clock Polarity

Selects the inactive state of the SCK. This bit is used in both Master and Slave mode. For successful communication between serial devices, the devices must have identical clock polarities. When the Continuous Selection Format is selected, switching between clock polarities without stopping the DSPI</td></tr>
</table>

Table continues on the next page...

SPIx_CTAR*n* field descriptions (continued)

Field	Description
	can cause errors in the transfer due to the peripheral device interpreting the switch of clock polarity as a valid clock edge. 0 The inactive state value of SCK is low. 1 The inactive state value of SCK is high.
25 CPHA	Clock Phase Selects which edge of SCK causes data to change and which edge causes data to be captured. This bit is used in both Master and Slave mode. For successful communication between serial devices, the devices must have identical clock phase settings. In Continuous SCK mode, the bit value is ignored and the transfers are done as if the CPHA bit is set to 1. 0 Data is captured on the leading edge of SCK and changed on the following edge. 1 Data is changed on the leading edge of SCK and captured on the following edge.
24 LSBFE	LSB First Specifies whether the LSB or MSB of the frame is transferred first. 0 Data is transferred MSB first. 1 Data is transferred LSB first.
23–22 PCSSCK	PCS to SCK Delay Prescaler Selects the prescaler value for the delay between assertion of PCS and the first edge of the SCK. See the CSSCK field description for information on how to compute the PCS to SCK Delay. See PCS to SCK Delay (t_{CSC}) for more details. 00 PCS to SCK Prescaler value is 1. 01 PCS to SCK Prescaler value is 3. 10 PCS to SCK Prescaler value is 5. 11 PCS to SCK Prescaler value is 7.
21–20 PASC	After SCK Delay Prescaler Selects the prescaler value for the delay between the last edge of SCK and the negation of PCS. See the ASC field description for information on how to compute the After SCK Delay.See After SCK Delay (t_{ASC}) for more details. 00 Delay after Transfer Prescaler value is 1. 01 Delay after Transfer Prescaler value is 3. 10 Delay after Transfer Prescaler value is 5. 11 Delay after Transfer Prescaler value is 7.
19–18 PDT	Delay after Transfer Prescaler Selects the prescaler value for the delay between the negation of the PCS signal at the end of a frame and the assertion of PCS at the beginning of the next frame. The PDT field is used only in Master mode. See the DT field description for details on how to compute the Delay after Transfer. See Delay after Transfer (t_{DT}) for more details. 00 Delay after Transfer Prescaler value is 1. 01 Delay after Transfer Prescaler value is 3. 10 Delay after Transfer Prescaler value is 5. 11 Delay after Transfer Prescaler value is 7.

Table continues on the next page...

SPIx_CTAR*n* field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td>17–16
PBR</td><td>Baud Rate Prescaler

Selects the prescaler value for the baud rate. This field is used only in Master mode. The baud rate is the frequency of the SCK. The system clock is divided by the prescaler value before the baud rate selection takes place. See the BR field description for details on how to compute the baud rate.

00 Baud Rate Prescaler value is 2.
01 Baud Rate Prescaler value is 3.
10 Baud Rate Prescaler value is 5.
11 Baud Rate Prescaler value is 7.</td></tr>
<tr><td>15–12
CSSCK</td><td>PCS to SCK Delay Scaler

Selects the scaler value for the PCS to SCK delay. This field is used only in Master mode. The PCS to SCK Delay is the delay between the assertion of PCS and the first edge of the SCK. The delay is a multiple of the system clock period, and it is computed according to the following equation:

$t_{CSC} = (1/f_{SYS})$ x PCSSCK x CSSCK

The following table lists the delay scaler values.

Table 43-33. Delay scaler encoding
<table>
<tr><th>Field value</th><th>Delay scaler value</th></tr>
<tr><td>0000</td><td>2</td></tr>
<tr><td>0001</td><td>4</td></tr>
<tr><td>0010</td><td>8</td></tr>
<tr><td>0011</td><td>16</td></tr>
<tr><td>0100</td><td>32</td></tr>
<tr><td>0101</td><td>64</td></tr>
<tr><td>0110</td><td>128</td></tr>
<tr><td>0111</td><td>256</td></tr>
<tr><td>1000</td><td>512</td></tr>
<tr><td>1001</td><td>1024</td></tr>
<tr><td>1010</td><td>2048</td></tr>
<tr><td>1011</td><td>4096</td></tr>
<tr><td>1100</td><td>8192</td></tr>
<tr><td>1101</td><td>16384</td></tr>
<tr><td>1110</td><td>32768</td></tr>
<tr><td>1111</td><td>65536</td></tr>
</table>

See PCS to SCK Delay (t_{CSC}) for more details.</td></tr>
<tr><td>11–8
ASC</td><td>After SCK Delay Scaler</td></tr>
</table>

Table continues on the next page...

SPIx_CTAR*n* field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>Selects the scaler value for the After SCK Delay. This field is used only in Master mode. The After SCK Delay is the delay between the last edge of SCK and the negation of PCS. The delay is a multiple of the system clock period, and it is computed according to the following equation:
$t_{ASC} = (1/f_{SYS}) \times PASC \times ASC$
See Delay Scaler Encoding table in CTARn[CSSCK] bit field description for scaler values. See After SCK Delay (t_{ASC}) for more details.</td></tr>
<tr><td>7–4
DT</td><td>Delay After Transfer Scaler
Selects the Delay after Transfer Scaler. This field is used only in Master mode. The Delay after Transfer is the time between the negation of the PCS signal at the end of a frame and the assertion of PCS at the beginning of the next frame.
In the Continuous Serial Communications Clock operation, the DT value is fixed to one SCK clock period. The Delay after Transfer is a multiple of the system clock period, and it is computed according to the following equation:
$t_{DT} = (1/f_{SYS}) \times PDT \times DT$
See Delay Scaler Encoding table in CTARn[CSSCK] field description for scaler values.</td></tr>
<tr><td>3–0
BR</td><td>Baud Rate Scaler
Selects the scaler value for the baud rate. This field is used only in Master mode. The prescaled system clock is divided by the Baud Rate Scaler to generate the frequency of the SCK. The baud rate is computed according to the following equation:
SCK baud rate = $(f_{SYS}/PBR) \times [(1+DBR)/BR]$
The following table lists the baud rate scaler values.
Table 43-34. DSPI baud rate scaler
<table>
<tr><th>CTARn[BR]</th><th>Baud rate scaler value</th></tr>
<tr><td>0000</td><td>2</td></tr>
<tr><td>0001</td><td>4</td></tr>
<tr><td>0010</td><td>6</td></tr>
<tr><td>0011</td><td>8</td></tr>
<tr><td>0100</td><td>16</td></tr>
<tr><td>0101</td><td>32</td></tr>
<tr><td>0110</td><td>64</td></tr>
<tr><td>0111</td><td>128</td></tr>
<tr><td>1000</td><td>256</td></tr>
<tr><td>1001</td><td>512</td></tr>
<tr><td>1010</td><td>1024</td></tr>
<tr><td>1011</td><td>2048</td></tr>
<tr><td>1100</td><td>4096</td></tr>
<tr><td>1101</td><td>8192</td></tr>
<tr><td>1110</td><td>16384</td></tr>
</table>
</td></tr>
</table>

Table continues on the next page...

SPIx_CTARn field descriptions (continued)

Field	Description
	Table 43-34. DSPI baud rate scaler (continued) CTARn[BR]: 1111 — Baud rate scaler value: 32768

43.3.4 DSPI Clock and Transfer Attributes Register (In Slave Mode) (SPIx_CTAR_SLAVE)

When the DSPI is configured as an SPI bus slave, the CTAR0 register is used.

Addresses: SPI0_CTAR0_SLAVE is 4002_C000h base + Ch offset = 4002_C00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FMSZ					CPOL	CPHA	0		0	0																					
W																																
Reset	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_CTARn_SLAVE field descriptions

Field	Description
31–27 FMSZ	Frame Size The number of bits transfered per frame is equal to the FMSZ field value plus 1. The minimum value of thisfield is 3.
26 CPOL	Clock Polarity Selects the inactive state of the Serial Communications Clock (SCK). 0 The inactive state value of SCK is low. 1 The inactive state value of SCK is high.
25 CPHA	Clock Phase Selects which edge of SCK causes data to change and which edge causes data to be captured. This bit is used in both master and slave mode. For successful communication between serial devices, the devices must have identical clock phase settings. In Continuous SCK mode, the bit value is ignored and the transfers are done as the CPHA bit is set to 1. 0 Data is captured on the leading edge of SCK and changed on the following edge. 1 Data is changed on the leading edge of SCK and captured on the following edge.
24–23 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

SPIx_CTAR*n*_SLAVE field descriptions (continued)

Field	Description
22 Reserved	This read-only field is reserved and always has the value zero.
21–0 Reserved	This read-only field is reserved and always has the value zero.

43.3.5 DSPI Status Register (SPIx_SR)

SR contains status and flag bits. The bits reflect the status of the DSPI and indicate the occurrence of events that can generate interrupt or DMA requests. Software can clear flag bits in the SR by writing a 1 to them. Writing a 0 to a flag bit has no effect. This register may not be writable in Module Disable mode due to the use of power saving mechanisms.

Addresses: SPI0_SR is 4002_C000h base + 2Ch offset = 4002_C02Ch

Bit	31	30	29	28	27	26	25	24
R	TCF	TXRXS	0	EOQF	TFUF	0	TFFF	0
W	w1c	w1c		w1c	w1c		w1c	
Reset	0	0	0	0	0	0	1	0

Bit	23	22	21	20	19	18	17	16
R	0	0	0	0	RFOF	0	RFDF	0
W					w1c		w1c	
Reset	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8
R	TXCTR				TXNXTPTR			
W								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	RXCTR				POPNXTPTR			
W								
Reset	0	0	0	0	0	0	0	0

SPIx_SR field descriptions

Field	Description
31 TCF	Transfer Complete Flag Indicates that all bits in a frame have been shifted out. TCF remains set until it is cleared by writing a 1 to it.

Table continues on the next page...

SPIx_SR field descriptions (continued)

Field	Description
	0 Transfer not complete. 1 Transfer complete.
30 TXRXS	TX and RX Status Reflects the run status of the DSPI. 0 Transmit and receive operations are disabled (DSPI is in Stopped state). 1 Transmit and receive operations are enabled (DSPI is in Running state).
29 Reserved	This read-only field is reserved and always has the value zero.
28 EOQF	End of Queue Flag Indicates that the last entry in a queue has been transmitted when the DSPI is in Master mode. The EOQF bit is set when the TX FIFO entry has the EOQ bit set in the command halfword and the end of the transfer is reached. The EOQF bit remains set until cleared by writing a 1 to it. When the EOQF bit is set, the TXRXS bit is automatically cleared. 0 EOQ is not set in the executing command. 1 EOQ is set in the executing SPI command.
27 TFUF	Transmit FIFO Underflow Flag Indicates an underflow condition in the TX FIFO. The transmit underflow condition is detected only for DSPI blocks operating in Slave mode and SPI configuration. TFUF is set when the TX FIFO of a DSPI operating in SPI Slave mode is empty and an external SPI master initiates a transfer. The TFUF bit remains set until cleared by writing 1 to it. 0 No TX FIFO underflow. 1 TX FIFO underflow has occurred.
26 Reserved	This read-only field is reserved and always has the value zero.
25 TFFF	Transmit FIFO Fill Flag Provides a method for the DSPI to request more entries to be added to the TX FIFO. The TFFF bit is set while the TX FIFO is not full. The TFFF bit can be cleared by writing 1 to it or by acknowledgement from the DMA controller to the TX FIFO full request. 0 TX FIFO is full. 1 TX FIFO is not full.
24 Reserved	This read-only field is reserved and always has the value zero.
23 Reserved	This read-only field is reserved and always has the value zero.
22 Reserved	This read-only field is reserved and always has the value zero.
21 Reserved	This read-only field is reserved and always has the value zero.
20 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

SPIx_SR field descriptions (continued)

Field	Description
19 RFOF	Receive FIFO Overflow Flag Indicates an overflow condition in the RX FIFO. The field is set when the RX FIFO and shift register are full and a transfer is initiated. The bit remains set until it is cleared by writing a 1 to it. 0 No Rx FIFO overflow. 1 Rx FIFO overflow has occurred.
18 Reserved	This read-only field is reserved and always has the value zero.
17 RFDF	Receive FIFO Drain Flag Provides a method for the DSPI to request that entries be removed from the RX FIFO. The bit is set while the RX FIFO is not empty. The RFDF bit can be cleared by writing 1 to it or by acknowledgement from the DMA controller when the RX FIFO is empty. 0 RX FIFO is empty. 1 RX FIFO is not empty.
16 Reserved	This read-only field is reserved and always has the value zero.
15–12 TXCTR	TX FIFO Counter Indicates the number of valid entries in the TX FIFO. The TXCTR is incremented every time the PUSHR is written. The TXCTR is decremented every time an SPI command is executed and the SPI data is transferred to the shift register.
11–8 TXNXTPTR	Transmit Next Pointer Indicates which TX FIFO entry is transmitted during the next transfer. The TXNXTPTR field is updated every time SPI data is transferred from the TX FIFO to the shift register.
7–4 RXCTR	RX FIFO Counter Indicates the number of entries in the RX FIFO. The RXCTR is decremented every time the POPR is read. The RXCTR is incremented every time data is transferred from the shift register to the RX FIFO.
3–0 POPNXTPTR	Pop Next Pointer Contains a pointer to the RX FIFO entry to be returned when the POPR is read. The POPNXTPTR is updated when the POPR is read.

43.3.6 DSPI DMA/Interrupt Request Select and Enable Register (SPIx_RSER)

RSER controls DMA and interrupt requests. Do not write to the RSER while the DSPI is in the Running state.

Addresses: SPI0_RSER is 4002_C000h base + 30h offset = 4002_C030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TCF_RE	0	0	EOQF_RE	TFUF_RE	0	TFFF_RE	TFFF_DIRS	0	0	0	0	RFOF_RE	0	RFDF_RE	RFDF_DIRS
W		▨	▨			▨			▨	▨	▨	▨		▨		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0													
W	▨	▨	▨													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_RSER field descriptions

Field	Description
31 TCF_RE	Transmission Complete Request Enable Enables TCF flag in the SR to generate an interrupt request. 0 TCF interrupt requests are disabled. 1 TCF interrupt requests are enabled.
30 Reserved	This read-only field is reserved and always has the value zero.
29 Reserved	This read-only field is reserved and always has the value zero.
28 EOQF_RE	DSPI Finished Request Enable Enables the EOQF flag in the SR to generate an interrupt request. 0 EOQF interrupt requests are disabled. 1 EOQF interrupt requests are enabled.
27 TFUF_RE	Transmit FIFO Underflow Request Enable Enables the TFUF flag in the SR to generate an interrupt request. 0 TFUF interrupt requests are disabled. 1 TFUF interrupt requests are enabled.
26 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

SPIx_RSER field descriptions (continued)

Field	Description
25 TFFF_RE	Transmit FIFO Fill Request Enable Enables the TFFF flag in the SR to generate a request. The TFFF_DIRS bit selects between generating an interrupt request or a DMA request. 0 TFFF interrupts or DMA requests are disabled. 1 TFFF interrupts or DMA requests are enabled.
24 TFFF_DIRS	Transmit FIFO Fill DMA or Interrupt Request Select Selects between generating a DMA request or an interrupt request. When SR[TFFF] and RSER[TFFF_RE] are set, this field selects between generating an interrupt request or a DMA request. 0 TFFF flag generates interrupt requests. 1 TFFF flag generates DMA requests.
23 Reserved	This read-only field is reserved and always has the value zero.
22 Reserved	This read-only field is reserved and always has the value zero.
21 Reserved	This read-only field is reserved and always has the value zero.
20 Reserved	This read-only field is reserved and always has the value zero.
19 RFOF_RE	Receive FIFO Overflow Request Enable Enables the RFOF flag in the SR to generate an interrupt request. 0 RFOF interrupt requests are disabled. 1 RFOF interrupt requests are enabled.
18 Reserved	This read-only field is reserved and always has the value zero.
17 RFDF_RE	Receive FIFO Drain Request Enable Enables the RFDF flag in the SR to generate a request. The RFDF_DIRS bit selects between generating an interrupt request or a DMA request. 0 RFDF interrupt or DMA requests are disabled. 1 RFDF interrupt or DMA requests are enabled.
16 RFDF_DIRS	Receive FIFO Drain DMA or Interrupt Request Select Selects between generating a DMA request or an interrupt request. When the RFDF flag bit in the SR is set, and the RFDF_RE bit in the RSER is set, the RFDF_DIRS bit selects between generating an interrupt request or a DMA request. 0 Interrupt request. 1 DMA request.
15 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

SPIx_RSER field descriptions (continued)

Field	Description
14 Reserved	This read-only field is reserved and always has the value zero.
13–0 Reserved	This read-only field is reserved and always has the value zero.

43.3.7 DSPI PUSH TX FIFO Register In Master Mode (SPIx_PUSHR)

PUSHR provides the means to write to the TX FIFO. Data written to this register is transferred to the TX FIFO . 8- or 16-bit write accesses to the Data Field of PUSHR transfers the 16 bit Data field of PUSHR to the TX FIFO. Write accesses to the Command Field of PUSHR transfers the 16 bit Command Field of PUSHR to the TX FIFO. The register structure is different in Master and Slave modes. In Master mode, the register provides 16-bit command and data to the TX FIFO. In Slave mode, the 16 bit Command Field of PUSHR is reserved.

A PUSHR Read Operation returns the topmost TX FIFO entry.

When DSPI Module is disabled, any writes to this register will not update the FIFO. Hence any reads performed during Module disable mode will return the last PUSHR write performed when Module was enabled.

Addresses: SPI0_PUSHR is 4002_C000h base + 34h offset = 4002_C034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CONT	CTAS			EOQ	CTCNT	0		0		PCS[5:0]						TXDATA															
W							▨		▨																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_PUSHR field descriptions

Field	Description
31 CONT	Continuous Peripheral Chip Select Enable Selects a continuous selection format. The bit is used in SPI Master mode. The bit enables the selected PCS signals to remain asserted between transfers. 0 Return PCSn signals to their inactive state between transfers. 1 Keep PCSn signals asserted between transfers.
30–28 CTAS	Clock and Transfer Attributes Select

Table continues on the next page...

SPIx_PUSHR field descriptions (continued)

Field	Description
	Selects which CTAR to use in master mode to specify the transfer attributes for the associated SPI frame. In SPI Slave mode, CTAR0 is used. See the chapter on chip configuration to determine how many CTARs this device has. You should not program a value in this field for a register that is not present. 000 CTAR0 001 CTAR1 010 Reserved 011 Reserved 100 Reserved 101 Reserved 110 Reserved 111 Reserved
27 EOQ	End Of Queue Host software uses this bit to signal to the DSPI that the current SPI transfer is the last in a queue. At the end of the transfer, the EOQF bit in the SR is set. 0 The SPI data is not the last data to transfer. 1 The SPI data is the last data to transfer.
26 CTCNT	Clear Transfer Counter Clears the TCNT field in the TCR register. The TCNT field is cleared before the DSPI starts transmitting the current SPI frame. 0 Do not clear the TCR[TCNT] field. 1 Clear the TCR[TCNT] field.
25–24 Reserved	This read-only field is reserved and always has the value zero.
23–22 Reserved	This read-only field is reserved and always has the value zero.
21–16 PCS[5:0]	Select which PCS signals are to be asserted for the transfer. Refer to the chip configuration chapter for the number of PCS signals used in this MCU. 0 Negate the PCS[x] signal. 1 Assert the PCS[x] signal.
15–0 TXDATA	Transmit Data Holds SPI data to be transferred according to the associated SPI command.

43.3.8 DSPI PUSH TX FIFO Register In Slave Mode (SPIx_PUSHR_SLAVE)

PUSHR provides the means to write to the TX FIFO. Data written to this register is transferred to the TX FIFO. Eight- or sixteen-bit write accesses to the Data Field of PUSHR transfers the 16 bit Data Field of PUSHR to the TX FIFO. The register structure is different in master and slave modes. The register structure is different in master and slave modes. In master mode the register provides 16-bit command and data to the TX FIFO. In slave mode, the 16 bit Command Field of PUSHR is reserved.

Addresses: SPI0_PUSHR_SLAVE is 4002_C000h base + 34h offset = 4002_C034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXDATA															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_PUSHR_SLAVE field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 TXDATA	Transmit Data Holds SPI data to be transferred according to the associated SPI command.

43.3.9 DSPI POP RX FIFO Register (SPIx_POPR)

POPR is used to read the RX FIFO. Eight- or sixteen-bit read accesses to the POPR have the same effect on the RX FIFO as 32-bit read accesses. A write to this register will generate a Transfer Error.

Addresses: SPI0_POPR is 4002_C000h base + 38h offset = 4002_C038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXDATA																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_POPR field descriptions

Field	Description
31–0 RXDATA	Received Data Contains the SPI data from the RX FIFO entry to which the Pop Next Data Pointer points.

43.3.10 DSPI Transmit FIFO Registers (SPIx_TXFR*n*)

TXFRn registers provide visibility into the TX FIFO for debugging purposes. Each register is an entry in the TX FIFO. The registers are read-only and cannot be modified. Reading the TXFRx registers does not alter the state of the TX FIFO.

Addresses: SPI0_TXFR0 is 4002_C000h base + 3Ch offset = 4002_C03Ch
SPI0_TXFR1 is 4002_C000h base + 40h offset = 4002_C040h
SPI0_TXFR2 is 4002_C000h base + 44h offset = 4002_C044h
SPI0_TXFR3 is 4002_C000h base + 48h offset = 4002_C048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TXCMD_TXDATA																TXDATA															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_TXFR*n* field descriptions

Field	Description
31–16 TXCMD_ TXDATA	Transmit Command or Transmit Data In Master mode the TXCMD field contains the command that sets the transfer attributes for the SPI data. In Slave mode, this field is reserved.
15–0 TXDATA	Transmit Data Contains the SPI data to be shifted out.

43.3.11 DSPI Receive FIFO Registers (SPIx_RXFR*n*)

RXFRn provide visibility into the RX FIFO for debugging purposes. Each register is an entry in the RX FIFO. The RXFRs are read-only. Reading the RXFRx registers does not alter the state of the RX FIFO.

Addresses: SPI0_RXFR0 is 4002_C000h base + 7Ch offset = 4002_C07Ch
SPI0_RXFR1 is 4002_C000h base + 80h offset = 4002_C080h
SPI0_RXFR2 is 4002_C000h base + 84h offset = 4002_C084h
SPI0_RXFR3 is 4002_C000h base + 88h offset = 4002_C088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RXDATA																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPIx_RXFR*n* field descriptions

Field	Description
31–0 RXDATA	Receive Data Contains the received SPI data.

43.4 Functional description

The Serial Peripheral Interface (DSPI) block supports full-duplex, synchronous serial communications between MCUs and peripheral devices. All communications are done with SPI-like protocol.

The DSPI has the following configurations:

- SPI Configuration in which the DSPI operates as a basic SPI or a queued SPI.

The DCONF field in the DSPI Module Configuration Register (MCR) determines the DSPI Configuration. See ../dil/DSPI_BG_V5x.xml#MCR for the DSPI configuration values.

The CTARn registers hold clock and transfer attributes. The SPI configuration allows to select which CTAR to use on a frame by frame basis by setting a field in the SPI command.

See ../dil/DSPI_BG_V5x.xml#CTAR_MASTER_2 for information on the fields of CTAR registers.

Typical master to slave connections are shown in the following figure. When a data transfer operation is performed, data is serially shifted a predetermined number of bit positions. Because the modules are linked, data is exchanged between the master and the slave. The data that was in the master shift register is now in the shift register of the slave, and vice versa. At the end of a transfer, the TCF bit in the SR is set to indicate a completed transfer.

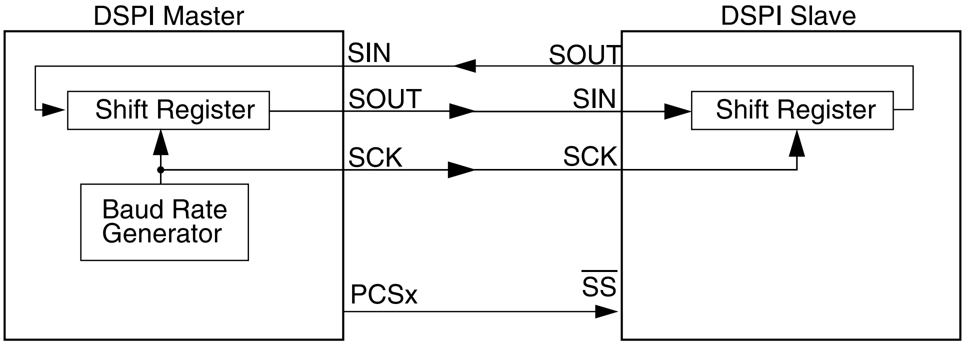


Figure 43-47. SPI serial protocol overview

Generally, more than one slave device can be connected to the DSPI master. 6 Peripheral Chip Select (PCS) signals of the DSPI masters can be used to select which of the slaves to communicate with. Refer to the chip configuration chapter for the number of PCS signals used in this MCU.

The three DSPI configurations share transfer protocol and timing properties which are described independently of the configuration in Transfer formats . The transfer rate and delay settings are described in DSPI baud rate and clock delay generation.

43.4.1 Start and Stop of DSPI transfers

The DSPI has two operating states: Stopped and Running. Both the states are independent of DSPI configuration. The default state of the DSPI is Stopped. In the Stopped state, no serial transfers are initiated in Master mode and no transfers are responded to in Slave mode. The Stopped state is also a safe state for writing the various configuration registers of the DSPI without causing undetermined results. In the Running state serial transfers take place.

The TXRXS bit in the SR indicates the state of DSPI. The bit is set if the module is in Running state.

The DSPI starts or transitions to Running when all of the following conditions are true:

- SR[EOQF] bit is clear
- MCU is not in the Debug mode or the MCR[FRZ] bit is clear
- MCR[HALT] bit is clear

The DSPI stops or transitions from Running to Stopped after the current frame when any one of the following conditions exist:

- SR[EOQF] bit is set
- MCU in the Debug mode and the MCR[FRZ] bit is set
- MCR[HALT] bit is set

State transitions from Running to Stopped occur on the next frame boundary if a transfer is in progress, or immediately if no transfers are in progress.

43.4.2 Serial Peripheral Interface (SPI) configuration

The SPI configuration transfers data serially using a shift register and a selection of programmable transfer attributes. The DSPI is in SPI configuration when the DCONF field in the MCR is 0b00. The SPI frames can be 32 bits long. The host CPU or a DMA controller transfers the SPI data from the external to DSPI RAM queues to a TX FIFO buffer. The received data is stored in entries in the RX FIFO buffer. The host CPU or the DMA controller transfers the received data from the RX FIFO to memory external to the DSPI. The operation of FIFO buffers is described in Transmit First In First Out (TX FIFO) buffering mechanism, Transmit First In First Out (TX FIFO) buffering mechanism and Receive First In First Out (RX FIFO) buffering mechanism. The interrupt and DMA request conditions are described in Interrupts/DMA requests.

The SPI configuration supports two block-specific modes—Master mode and Slave mode. In Master mode the DSPI initiates and controls the transfer according to the fields of the executing SPI Command. In Slave mode, the DSPI responds only to transfers initiated by a bus master external to the DSPI and the SPI command field space is reserved.

43.4.2.1 Master mode

In SPI Master, mode the DSPI initiates the serial transfers by controlling the SCK and the PCS signals. The executing SPI Command determines which CTARs will be used to set the transfer attributes and which PCS signals to assert . The command field also contains various bits that help with queue management and transfer protocol . See ../dil/DSPI_BG_V5x.xml#PUSHR_MASTER for details on the SPI command fields. The data in the executing TX FIFO entry is loaded into the shift register and shifted out on the Serial Out (SOUT) pin. In SPI Master mode, each SPI frame to be transmitted has a command associated with it, allowing for transfer attribute control on a frame by frame basis.

43.4.2.2 Slave mode

In SPI Slave mode the DSPI responds to transfers initiated by an SPI bus master. The DSPI does not initiate transfers. Certain transfer attributes such as clock polarity, clock phase, and frame size must be set for successful communication with an SPI master. The SPI Slave mode transfer attributes are set in the CTAR0 . The data is shifted out with MSB first. Shifting out of LSB is not supported in this mode.

43.4.2.3 FIFO disable operation

The FIFO disable mechanisms allow SPI transfers without using the TX FIFO or RX FIFO. The DSPI operates as a double-buffered simplified SPI when the FIFOs are disabled. The Transmit and Receive side of the FIFOs are disabled separately; setting the MCR[DIS_TXF] bit disables the TX FIFO, and setting the MCR[DIS_RXF] bit disables the RX FIFO.

The FIFO disable mechanisms are transparent to the user and to host software. Transmit data and commands are written to the PUSHR and received data is read from the POPR.

When the TX FIFO is disabled, the fields SR[TFFF], SR[TFUF] and SR[TXCTR] behave as if there is a one-entry FIFO but the contents of TXFRs, SR[TXNXTPTR] are undefined. Similarly, when the RX FIFO is disabled, the RFDF, RFOF, and RXCTR fields in the SR behave as if there is a one-entry FIFO, but the contents of the RXFR registers and POPNXTPTR are undefined.

43.4.2.4 Transmit First In First Out (TX FIFO) buffering mechanism

The TX FIFO functions as a buffer of SPI data for transmission. The TX FIFO holds 4 words, each consisting of SPI data. The number of entries in the TX FIFO is device-specific. SPI data is added to the TX FIFO by writing to the Data Field of DSPI PUSH FIFO Register (PUSHR). TX FIFO entries can only be removed from the TX FIFO by being shifted out or by flushing the TX FIFO.

The TX FIFO Counter field (TXCTR) in the DSPI Status Register (SR) indicates the number of valid entries in the TX FIFO. The TXCTR is updated every time a 8- or 16-bit write takes place to the Data Field of DSPI_PUSHR or SPI data is transferred into the shift register from the TX FIFO.

The TXNXTPTR field indicates the TX FIFO Entry that will be transmitted during the next transfer. The TXNXTPTR field is incremented every time SPI data is transferred from the TX FIFO to the shift register. The maximum value of the field is equal to the maximum implemented TXFR number and it rolls over after reaching the maximum.

43.4.2.4.1 Filling the TX FIFO

Host software or other intelligent blocks can add (push) entries to the TX FIFO by writing to the PUSHR. When the TX FIFO is not full, the TX FIFO Fill Flag (TFFF) in the SR is set. The TFFF bit is cleared when TX FIFO is full and the DMA controller indicates that a write to PUSHR is complete. Writing a '1' to the TFFF bit also clears it. The TFFF can generate a DMA request or an interrupt request. See Transmit FIFO Fill Interrupt or DMA Request for details.

The DSPI ignores attempts to push data to a full TX FIFO, and the state of the TX FIFO does not change and no error condition is indicated.

43.4.2.4.2 Draining the TX FIFO

The TX FIFO entries are removed (drained) by shifting SPI data out through the shift register. Entries are transferred from the TX FIFO to the shift register and shifted out as long as there are valid entries in the TX FIFO. Every time an entry is transferred from the TX FIFO to the shift register, the TX FIFO Counter decrements by one. At the end of a transfer, the TCF bit in the SR is set to indicate the completion of a transfer. The TX FIFO is flushed by writing a '1' to the CLR_TXF bit in MCR.

If an external bus master initiates a transfer with a DSPI slave while the slave's DSPI TX FIFO is empty, the Transmit FIFO Underflow Flag (TFUF) in the slave's SR is set. See Transmit FIFO Underflow Interrupt Request for details.

43.4.2.5 Receive First In First Out (RX FIFO) buffering mechanism

The RX FIFO functions as a buffer for data received on the SIN pin. The RX FIFO holds 4 received SPI data frames. The number of entries in the RX FIFO is device-specific. SPI data is added to the RX FIFO at the completion of a transfer when the received data in the shift register is transferred into the RX FIFO. SPI data are removed (popped) from the RX FIFO by reading the DSPI POP RX FIFO Register (POPR). RX FIFO entries can only be removed from the RX FIFO by reading the POPR or by flushing the RX FIFO.

The RX FIFO Counter field (RXCTR) in the DSPI Status Register (SR) indicates the number of valid entries in the RX FIFO. The RXCTR is updated every time the POPR is read or SPI data is copied from the shift register to the RX FIFO.

The POPNXTPTR field in the SR points to the RX FIFO entry that is returned when the POPR is read. The POPNXTPTR contains the positive offset from RXFR0 in a number of 32-bit registers. For example, POPNXTPTR equal to two means that the RXFR2 contains the received SPI data that will be returned when the POPR is read. The POPNXTPTR field is incremented every time the POPR is read. The maximum value of the field is equal to the maximum implemented RXFR number and it rolls over after reaching the maximum.

43.4.2.5.1 Filling the RX FIFO

The RX FIFO is filled with the received SPI data from the shift register. While the RX FIFO is not full, SPI frames from the shift register are transferred to the RX FIFO. Every time an SPI frame is transferred to the RX FIFO, the RX FIFO Counter is incremented by one.

If the RX FIFO and shift register are full and a transfer is initiated, the RFOF bit in the SR is set indicating an overflow condition. Depending on the state of the ROOE bit in the MCR, the data from the transfer that generated the overflow is either ignored or shifted in to the shift register. If the ROOE bit is set, the incoming data is shifted in to the shift register. If the ROOE bit is cleared, the incoming data is ignored.

43.4.2.5.2 Draining the RX FIFO

Host CPU or a DMA can remove (pop) entries from the RX FIFO by reading the DSPI POP RX FIFO Register (POPR). A read of the POPR decrements the RX FIFO Counter by one. Attempts to pop data from an empty RX FIFO are ignored and the RX FIFO Counter remains unchanged. The data, read from the empty RX FIFO, is undetermined.

When the RX FIFO is not empty, the RX FIFO Drain Flag (RFDF) in the SR is set. The RFDF bit is cleared when the RX_FIFO is empty and the DMA controller indicates that a read from POPR is complete or by writing a 1 to it.

43.4.3 DSPI baud rate and clock delay generation

The SCK frequency and the delay values for serial transfer are generated by dividing the system clock frequency by a prescaler and a scaler with the option for doubling the baud rate. The following figure shows conceptually how the SCK signal is generated.

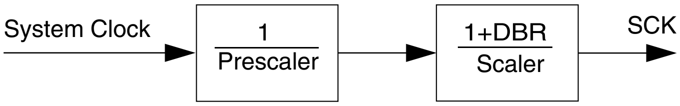


Figure 43-48. Communications clock prescalers and scalers

43.4.3.1 Baud rate generator

The baud rate is the frequency of the SCK. The system clock is divided by a prescaler (PBR) and scaler (BR) to produce SCK with the possibility of halving the scaler division. The DBR, PBR, and BR fields in the CTARs select the frequency of SCK by the formula in the BR field description. The following table shows an example of how to compute the baud rate.

Table 43-54. Baud rate computation example

f_{SYS}	PBR	Prescaler	BR	Scaler	DBR	Baud rate
100 MHz	0b00	2	0b0000	2	0	25 Mb/s
20 MHz	0b00	2	0b0000	2	1	10 Mb/s

NOTE

The clock frequencies mentioned in the preceding table are given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

43.4.3.2 PCS to SCK Delay (t_{CSC})

The PCS to SCK delay is the length of time from assertion of the PCS signal to the first SCK edge. See Figure 43-49 for an illustration of the PCS to SCK delay. The PCSSCK and CSSCK fields in the CTAR*x* registers select the PCS to SCK delay by the formula in the CSSCK field description. The following table shows an example of how to compute the PCS to SCK delay.

Table 43-55. PCS to SCK delay computation example

f_{SYS}	PCSSCK	Prescaler	CSSCK	Scaler	PCS to SCK Delay
100 MHz	0b01	3	0b0100	32	0.96 μs

NOTE

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

43.4.3.3 After SCK Delay (t_{ASC})

The After SCK Delay is the length of time between the last edge of SCK and the negation of PCS. See Figure 43-49 and Figure 43-50 for illustrations of the After SCK delay. The PASC and ASC fields in the CTAR*x* registers select the After SCK Delay by the formula in the ASC field description. The following table shows an example of how to compute the After SCK delay.

Table 43-56. After SCK Delay computation example

f_{SYS}	PASC	Prescaler	ASC	Scaler	After SCK Delay
100 MHz	0b01	3	0b0100	32	0.96 μs

NOTE

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

43.4.3.4 Delay after Transfer (t_{DT})

The Delay after Transfer is the minimum time between negation of the PCS signal for a frame and the assertion of the PCS signal for the next frame. See Figure 43-49 for an illustration of the Delay after Transfer. The PDT and DT fields in the CTAR*x* registers select the Delay after Transfer by the formula in the DT field description. The following table shows an example of how to compute the Delay after Transfer.

Table 43-57. Delay after Transfer computation example

f_{SYS}	PDT	Prescaler	DT	Scaler	Delay after Transfer
100 MHz	0b01	3	0b1110	32768	0.98 ms

NOTE

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

When in Non-Continuous Clock mode the t_{DT} delay is configured according to the equation specified in the CTAR[DT] bitfield description. When in Continuous Clock mode, the delay is fixed at 1 SCK period.

43.4.4 Transfer formats

The SPI serial communication is controlled by the Serial Communications Clock (SCK) signal and the PCS signals. The SCK signal provided by the master device synchronizes shifting and sampling of the data on the SIN and SOUT pins. The PCS signals serve as enable signals for the slave devices.

In Master mode, the CPOL and CPHA bits in the Clock and Transfer Attributes Registers (CTARn) select the polarity and phase of the serial clock, SCK.

- CPOL - Selects the idle state polarity of the SCK
- CPHA - Selects if the data on SOUT is valid before or on the first SCK edge

Even though the bus slave does not control the SCK signal, in Slave mode these values must be identical to the master device settings to ensure proper transmission. In SPI Slave mode, only CTAR0 is used.

The DSPI supports four different transfer formats:

- Classic SPI with CPHA=0
- Classic SPI with CPHA=1
- Modified Transfer Format with CPHA = 0
- Modified Transfer Format with CPHA = 1

A modified transfer format is supported to allow for high-speed communication with peripherals that require longer setup times. The DSPI can sample the incoming data later than halfway through the cycle to give the peripheral more setup time. The MTFE bit in the MCR selects between Classic SPI Format and Modified Transfer Format.

In the SPI configurations, the DSPI provides the option of keeping the PCS signals asserted between frames. See Continuous Selection Format for details.

43.4.4.1 Classic SPI Transfer Format (CPHA = 0)

The transfer format shown in following figure is used to communicate with peripheral SPI slave devices where the first data bit is available on the first clock edge. In this format, the master and slave sample their SIN pins on the odd-numbered SCK edges and change the data on their SOUT pins on the even-numbered SCK edges.

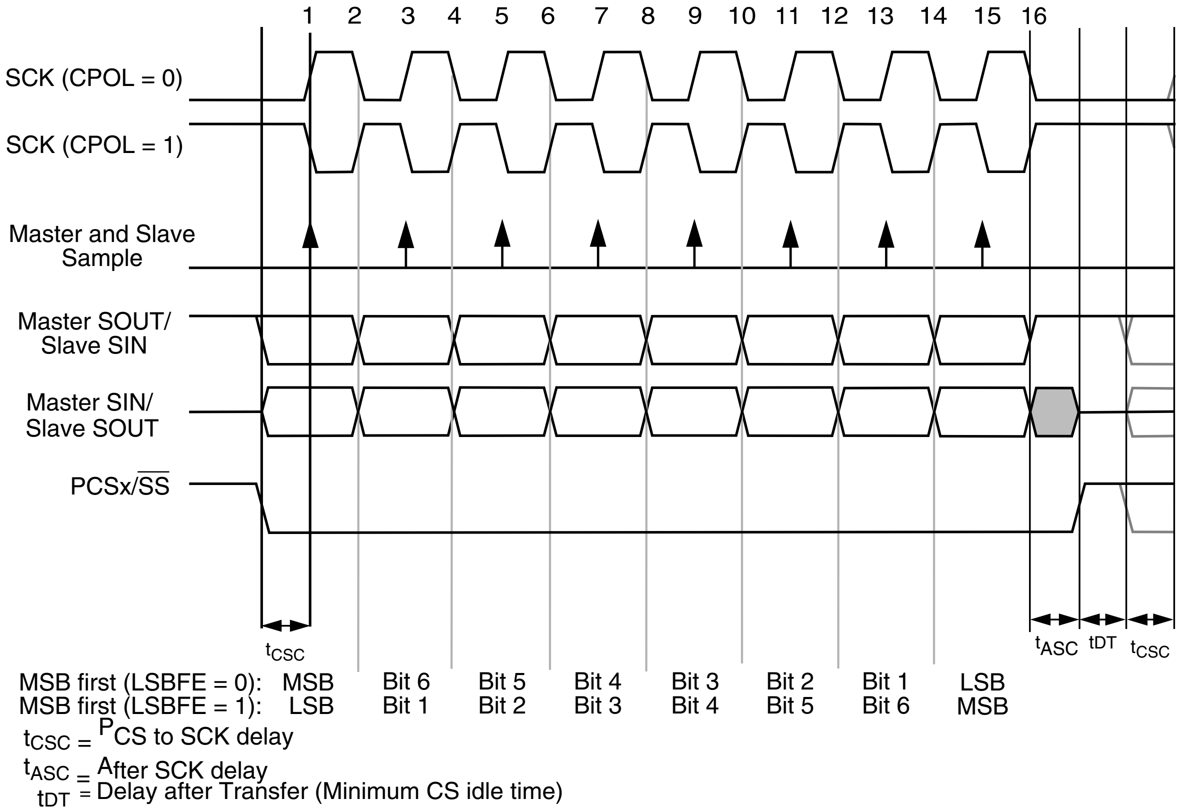
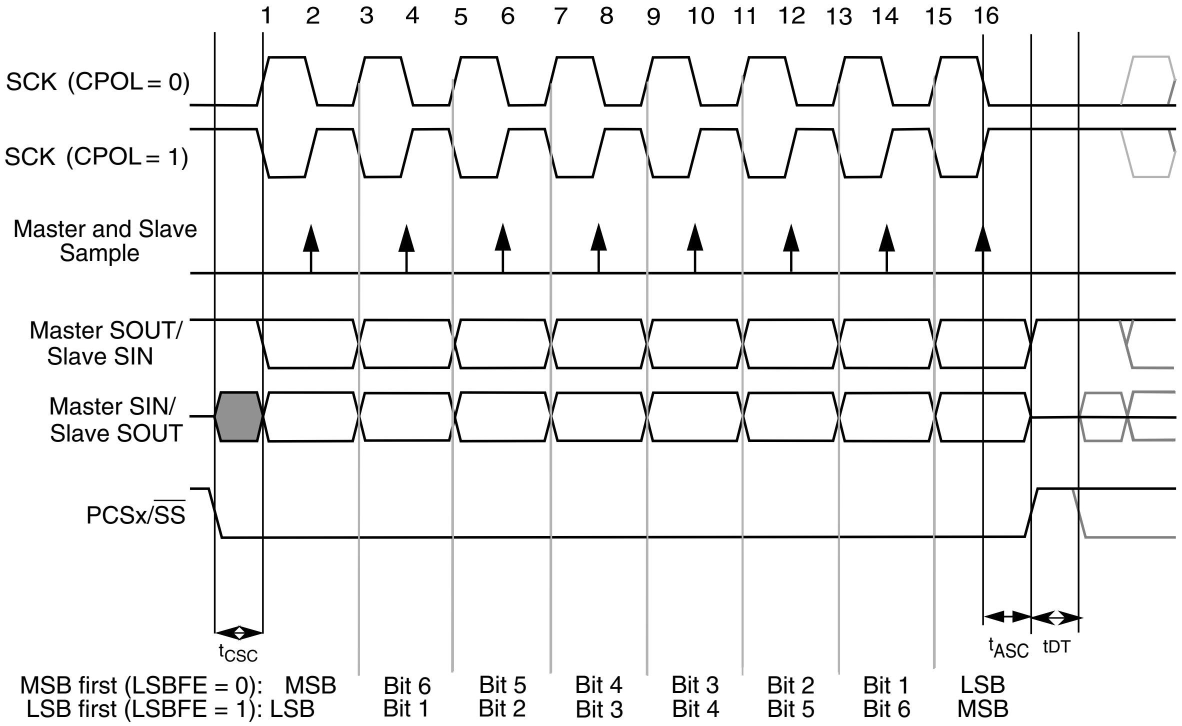

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
SCK (CPOL = 0)
SCK (CPOL = 1)
Master and Slave Sample
Master SOUT/ Slave SIN
Master SIN/ Slave SOUT
PCSx/$\overline{SS}$
t_{CSC}
t_{ASC}
t_{DT}
t_{CSC}
MSB first (LSBFE = 0): MSB Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 LSB
MSB first (LSBFE = 1): LSB Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 MSB
t_{CSC} = PCS to SCK delay
t_{ASC} = After SCK delay
t_{DT} = Delay after Transfer (Minimum CS idle time)

Figure 43-49. DSPI transfer timing diagram (MTFE=0, CPHA=0, FMSZ=8)

The master initiates the transfer by placing its first data bit on the SOUT pin and asserting the appropriate peripheral chip select signals to the slave device. The slave responds by placing its first data bit on its SOUT pin. After the t_{ASC} delay elapses, the master outputs the first edge of SCK. The master and slave devices use this edge to sample the first input data bit on their serial data input signals. At the second edge of the SCK, the master and slave devices place their second data bit on their serial data output signals. For the rest of the frame the master and the slave sample their SIN pins on the odd-numbered clock edges and changes the data on their SOUT pins on the even-numbered clock edges. After the last clock edge occurs, a delay of t_{ASC} is inserted before the master negates the PCS signals. A delay of t_{DT} is inserted before a new frame transfer can be initiated by the master.

43.4.4.2 Classic SPI Transfer Format (CPHA = 1)

This transfer format shown in the following figure is used to communicate with peripheral SPI slave devices that require the first SCK edge before the first data bit becomes available on the slave SOUT pin. In this format, the master and slave devices change the data on their SOUT pins on the odd-numbered SCK edges and sample the data on their SIN pins on the even-numbered SCK edges


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
SCK (CPOL = 0)
SCK (CPOL = 1)
Master and Slave Sample
Master SOUT/ Slave SIN
Master SIN/ Slave SOUT
PCSx/$\overline{SS}$
t_{CSC}
t_{ASC}
tDT
MSB first (LSBFE = 0): MSB Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 LSB
LSB first (LSBFE = 1): LSB Bit 1 Bit 2 Bit 3 Bit 4 Bit 5 Bit 6 MSB

t_{CSC} = PCS to SCK delay
t_{ASC} = After SCK delay
tDT = Delay after Transfer (minimum CS negation time)

Figure 43-50. DSPI transfer timing diagram (MTFE=0, CPHA=1, FMSZ=8)

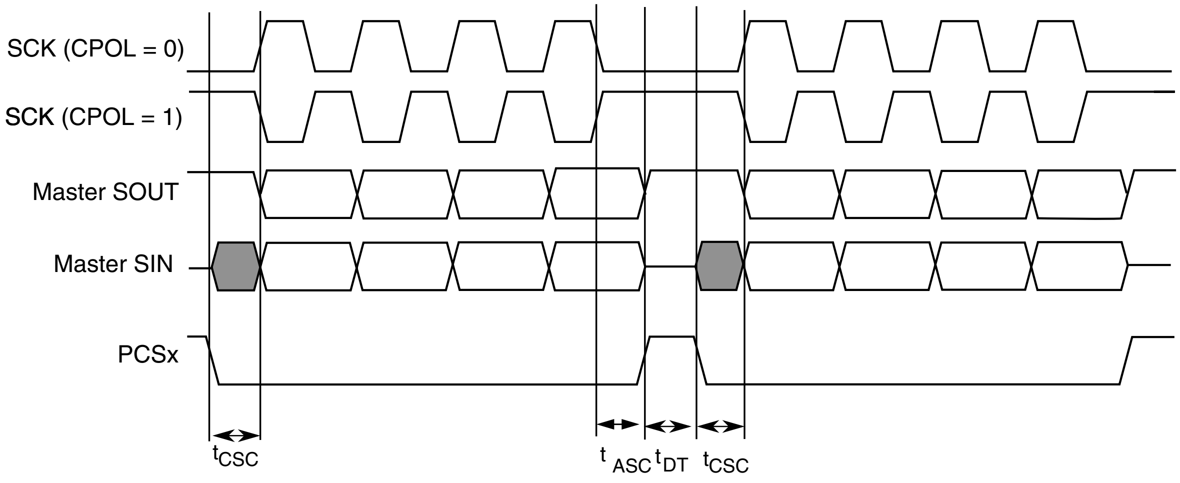
The master initiates the transfer by asserting the PCS signal to the slave. After the t_{CSC} delay has elapsed, the master generates the first SCK edge and at the same time places valid data on the master SOUT pin . The slave responds to the first SCK edge by placing its first data bit on its slave SOUT pin.

At the second edge of the SCK the master and slave sample their SIN pins. For the rest of the frame the master and the slave change the data on their SOUT pins on the odd-numbered clock edges and sample their SIN pins on the even-numbered clock edges. After the last clock edge occurs, a delay of t_{ASC} is inserted before the master negates the PCS signal. A delay of t_{DT} is inserted before a new frame transfer can be initiated by the master.

43.4.4.3 Continuous Selection Format

Some peripherals must be deselected between every transfer. Other peripherals must remain selected between several sequential serial transfers. The Continuous Selection Format provides the flexibility to handle the following case. The Continuous Selection Format is enabled for the SPI configuration by setting the CONT bit in the SPI command. The behavior of the PCS signals in the configurations is identical so only SPI configuration will be described.

When the CONT bit = 0, the DSPI drives the asserted Chip Select signals to their idle states in between frames. The idle states of the Chip Select signals are selected by the PCSISn bits in the MCR. The following timing diagram is for two four-bit transfers with CPHA = 1 and CONT = 0.



t_{CSC} = PCS to SCK dela

t_{ASC} = After SCK delay

t_{DT} = Delay after Transfer (minimum CS negation time)

Figure 43-51. Example of non-continuous format (CPHA=1, CONT=0)

When the CONT bit = 1, the PCS signal remains asserted for the duration of the two transfers. The Delay between Transfers (t_{DT}) is not inserted between the transfers. The following figure shows the timing diagram for two four-bit transfers with CPHA = 1 and CONT = 1.

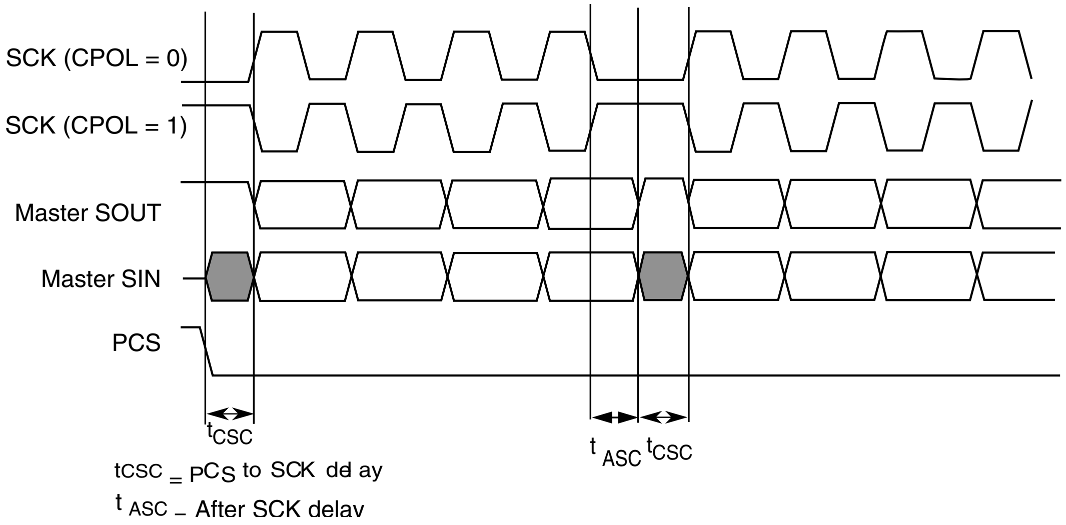


Figure 43-52. Example of continuous transfer (CPHA=1, CONT=1)

When using DSPI with continuous selection follow these rules:

- All transmit commands must have the same PCSn bits programming.
- The CTARs, selected by transmit commands, must be programmed with the same transfer attributes. Only FMSZ field can be programmed differently in these CTARs.
- When transmitting multiple frames in this mode, the user software must ensure that the last frame has the PUSHR[CONT] bit deasserted in Master mode and the user software must provide sufficient frames in the TX_FIFO to be sent out in Slave mode and the master deasserts the PCSn at end of transmission of the last frame.
- The PUSHR[CONT] / DSICR0[DCONT] bits must be deasserted before asserting MCR[HALT] bit in master mode. This will make sure that the PCSn signals are deasserted. Asserting MCR[HALT] bit during continuous transfer will cause the PCSn signals to remain asserted and hence Slave Device cannot transition from Running to Stopped state.

NOTE

User must fill the TX FIFO with the number of entries that will be concatenated together under one PCS assertion for both master and slave before the TX FIFO becomes empty.

When operating in Slave mode, ensure that when the last entry in the TX FIFO is completely transmitted, that is, the corresponding TCF flag is asserted and TXFIFO is empty, the slave is deselected for any further serial communication; otherwise, an underflow error occurs.

43.4.5 Continuous Serial Communications Clock

The DSPI provides the option of generating a Continuous SCK signal for slave peripherals that require a continuous clock.

Continuous SCK is enabled by setting the CONT_SCKE bit in the MCR. Enabling this bit generates the Continuous SCK regardless of the MCR[HALT] bit status. Continuous SCK is valid in all configurations.

Continuous SCK is only supported for CPHA=1. Clearing CPHA is ignored if the CONT_SCKE bit is set. Continuous SCK is supported for Modified Transfer Format.

Clock and transfer attributes for the Continuous SCK mode are set according to the following rules:

- When the DSPI is in SPI configuration, CTAR0 is used initially. At the start of each SPI frame transfer, the CTAR specified by the CTAS for the frame is used.
- In all configurations, the currently selected CTAR remains in use until the start of a frame with a different CTAR specified, or the Continuous SCK mode is terminated.

It is recommended to keep the baud rate the same while using the Continuous SCK. Switching clock polarity between frames while using Continuous SCK can cause errors in the transfer. Continuous SCK operation is not guaranteed if the DSPI is put into the External Stop mode or Module Disable mode.

Enabling Continuous SCK disables the PCS to SCK delay and the Delay after Transfer (t_{DT}) is fixed to one SCK cycle. The following figure is the timing diagram for Continuous SCK format with Continuous Selection disabled.

NOTE

In Continuous SCK mode, for the SPI transfer CTAR0 should always be used, and the TX FIFO must be cleared using the MCR[CLR_TXF] field before initiating transfer.

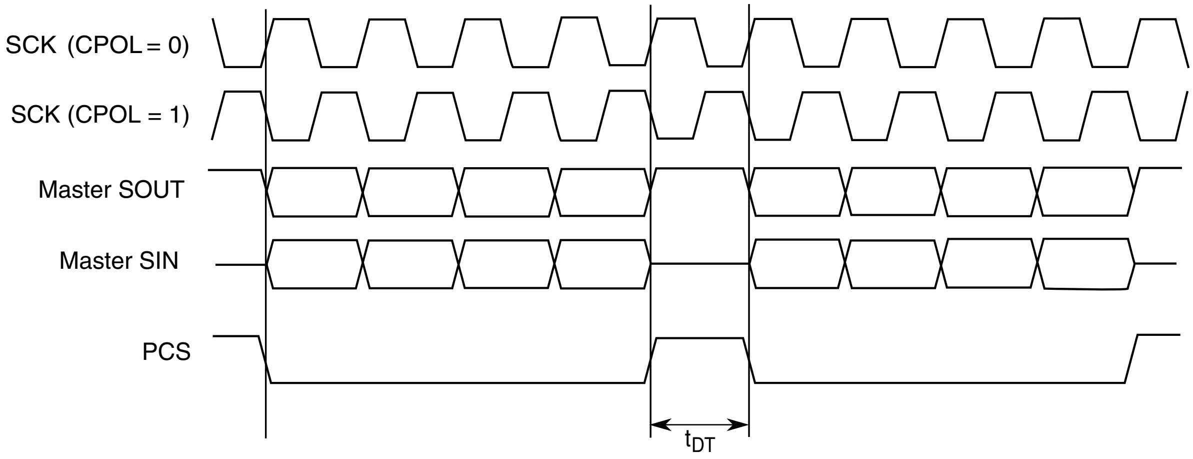

SCK (CPOL = 0)
SCK (CPOL = 1)
Master SOUT
Master SIN
PCS
t_{DT}

Figure 43-53. Continuous SCK Timing Diagram (CONT=0)

If the CONT bit in the TX FIFO entry is set, PCS remains asserted between the transfers. Under certain conditions, SCK can continue with PCS asserted, but with no data being shifted out of SOUT, that is, SOUT pulled high. This can cause the slave to receive incorrect data. Those conditions include:

- Continuous SCK with CONT bit set, but no data in the TX FIFO.
- Continuous SCK with CONT bit set and entering Stopped state (refer to Start and Stop of DSPI transfers).
- Continuous SCK with CONT bit set and entering Stop mode or Module Disable mode.

The following figure shows timing diagram for Continuous SCK format with Continuous Selection enabled.

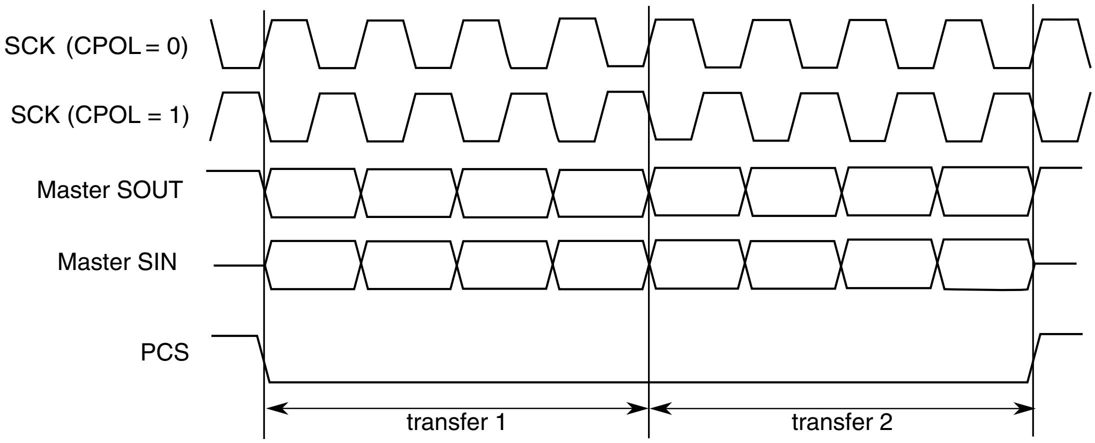

SCK (CPOL = 0)
SCK (CPOL = 1)
Master SOUT
Master SIN
PCS
transfer 1
transfer 2

Figure 43-54. Continuous SCK timing diagram (CONT=1)

43.4.6 Slave Mode Operation Constraints

Slave mode logic shift register is buffered. This allows data streaming operation, when the DSPI is permanently selected and data is shifted in with a constant rate.

The transmit data is transferred at second SCK clock edge of the each frame to the shift register if the $\overline{SS}$ signal is asserted and any time when transmit data is ready and $\overline{SS}$ signal is negated.

Received data is transferred to the receive buffer at last SCK edge of each frame, defined by frame size programmed to the CTAR0/1 register. Then the data from the buffer is transferred to the RXFIFO or DDR register.

If the $\overline{SS}$ negates before that last SCK edge, the data from shift register is lost.

43.4.7 Interrupts/DMA requests

The DSPI has several conditions that can generate only interrupt requests and two conditions that can generate interrupt or DMA requests. The following table lists these conditions.

Table 43-58. Interrupt and DMA request conditions

Condition	Flag	Interrupt	DMA
End of Queue (EOQ)	EOQF	Yes	-
TX FIFO Fill	TFFF	Yes	Yes
Transfer Complete	TCF	Yes	-
TX FIFO Underflow	TFUF	Yes	-
RX FIFO Drain	RFDF	Yes	Yes
RX FIFO Overflow	RFOF	Yes	-

Each condition has a flag bit in the DSPI Status Register (SR) and a Request Enable bit in the DSPI DMA/Interrupt Request Select and Enable Register (RSER). Certain flags (as shown in above table) generate interrupt requests or DMA requests depending on configuration of DSPI_RSER register.

The DSPI module also provides a global interrupt request line, which is asserted when any of individual interrupt requests lines is asserted.

43.4.7.1 End of Queue Interrupt Request

The End of Queue Request indicates that the end of a transmit queue is reached. The End of Queue Request is generated when the EOQ bit in the executing SPI command is set and the EOQF_RE bit in the RSER is set.

NOTE

This interrupt request is generated when the last bit of the SPI frame with EOQ bit set is transmitted.

43.4.7.2 Transmit FIFO Fill Interrupt or DMA Request

The Transmit FIFO Fill Request indicates that the TX FIFO is not full. The Transmit FIFO Fill Request is generated when the number of entries in the TX FIFO is less than the maximum number of possible entries, and the TFFF_RE bit in the RSER is set. The TFFF_DIRS bit in the RSER selects whether a DMA request or an interrupt request is generated.

NOTE

TFFF flag clears automatically when DMA is used to fill TX FIFO.

To clear TFFF when not using DMA, follow these steps for every PUSH performed using CPU to fill TX FIFO:

1. Wait until TFFF = 1.
2. Write data to PUSHR using CPU.
3. Clear TFFF by writing a 1 to its location. If TX FIFO is not full, this flag will not clear.

43.4.7.3 Transfer Complete Interrupt Request

The Transfer Complete Request indicates the end of the transfer of a serial frame. The Transfer Complete Request is generated at the end of each frame transfer when the TCF_RE bit is set in the RSER.

43.4.7.4 Transmit FIFO Underflow Interrupt Request

The Transmit FIFO Underflow Request indicates that an underflow condition in the TX FIFO has occurred. The transmit underflow condition is detected only for the DSPI, operating in Slave mode and SPI configuration . The TFUF bit is set when the TX FIFO of a DSPI is empty, and a transfer is initiated from an external SPI master. If the TFUF bit is set while the TFUF_RE bit in the RSER is set, an interrupt request is generated.

43.4.7.5 Receive FIFO Drain Interrupt or DMA Request

The Receive FIFO Drain Request indicates that the RX FIFO is not empty. The Receive FIFO Drain Request is generated when the number of entries in the RX FIFO is not zero, and the RFDF_RE bit in the RSER is set. The RFDF_DIRS bit in the RSER selects whether a DMA request or an interrupt request is generated.

43.4.7.6 Receive FIFO Overflow Interrupt Request

The Receive FIFO Overflow Request indicates that an overflow condition in the RX FIFO has occurred. A Receive FIFO Overflow request is generated when RX FIFO and shift register are full and a transfer is initiated. The RFOF_RE bit in the RSER must be set for the interrupt request to be generated.

Depending on the state of the ROOE bit in the MCR, the data from the transfer that generated the overflow is either ignored or shifted in to the shift register. If the ROOE bit is set, the incoming data is shifted in to the shift register. If the ROOE bit is cleared, the incoming data is ignored.

43.4.8 Power saving features

The DSPI supports following power-saving strategies:

- External Stop mode
- Module Disable mode – Clock gating of non-memory mapped logic

43.4.8.1 Stop mode (External Stop mode)

The DSPI supports the Stop mode protocol. When a request is made to enter External Stop mode, the DSPI block acknowledges the request . If a serial transfer is in progress, the DSPI waits until it reaches the frame boundary before it is ready to have its clocks shut off . While the clocks are shut off, the DSPI memory-mapped logic is not accessible. This also puts the DSPI in STOPPED state. The SR[TXRXS] bit is cleared to indicate STOPPED state. The states of the interrupt and DMA request signals cannot be changed while in External Stop mode.

43.4.8.2 Module Disable mode

Module Disable mode is a block-specific mode that the DSPI can enter to save power. Host CPU can initiate the Module Disable mode by setting the MDIS bit in the MCR. The Module Disable mode can also be initiated by hardware. A power management block can initiate the Module Disable mode by asserting the DOZE mode signal while the DOZE bit in the MCR is set.

When the MDIS bit is set or the DOZE mode signal is asserted while the DOZE bit is set, the DSPI negates Clock Enable signal at the next frame boundary. Once the Clock Enable signal is negated, DSPI is said to have entered Module Disable Mode. This also puts the DSPI in STOPPED state. The SR[TXRXS] bit is cleared to indicate STOPPED state.If implemented, the Clock Enable signal can stop the clock to the non-memory mapped logic. When Clock Enable is negated, the DSPI is in a dormant state, but the memory mapped registers are still accessible. Certain read or write operations have a different effect when the DSPI is in the Module Disable mode. Reading the RX FIFO Pop Register does not change the state of the RX FIFO. Similarly, writing to the PUSHR Register does not change the state of the TX FIFO. Clearing either of the FIFOs has no effect in the Module Disable mode. Changes to the DIS_TXF and DIS_RXF fields of the MCR have no effect in the Module Disable mode. In the Module Disable mode, all status bits and register flags in the DSPI return the correct values when read, but writing to them has no effect. Writing to the TCR during Module Disable mode has no effect. Interrupt and DMA request signals cannot be cleared while in the Module Disable mode.

43.5 Initialization/application information

This section describes how to initialize the DSPI module.

43.5.1 How to manage DSPI queues

The queues are not part of the DSPI, but the DSPI includes features in support of queue management. Queues are primarily supported in SPI configuration.

1. When DSPI executes last command word from a queue, the EOQ bit in the command word is set to indicate to the DSPI that this is the last entry in the queue.
2. At the end of the transfer, corresponding to the command word with EOQ set is sampled, the EOQ flag (EOQF) in the SR is set.
3. The setting of the EOQF flag disables serial transmission and reception of data, putting the DSPI in the Stopped state. The TXRXS bit is cleared to indicate the Stopped state.
4. The DMA can continue to fill TX FIFO until it is full or step 5 occurs.
5. Disable DSPI DMA transfers by disabling the DMA enable request for the DMA channel assigned to TX FIFO and RX FIFO. This is done by clearing the corresponding DMA enable request bits in the DMA Controller.
6. Ensure all received data in RX FIFO has been transferred to memory receive queue by reading the RXCNT in SR or by checking RFDF in the SR after each read operation of the POPR.
7. Modify DMA descriptor of TX and RX channels for new queues
8. Flush TX FIFO by writing a 1 to the CLR_TXF bit in the MCR. Flush RX FIFO by writing a '1' to the CLR_RXF bit in the MCR.
9. Clear transfer count either by setting CTCNT bit in the command word of the first entry in the new queue or via CPU writing directly to SPI_TCNT field in the TCR.
10. Enable DMA channel by enabling the DMA enable request for the DMA channel assigned to the DSPI TX FIFO, and RX FIFO by setting the corresponding DMA set enable request bit.
11. Enable serial transmission and serial reception of data by clearing the EOQF bit.

43.5.2 Switching Master and Slave mode

When changing modes in the DSPI, follow the steps below to guarantee proper operation.

1. Halt the DSPI by setting MCR[HALT].

2. Clear the transmit and receive FIFOs by writing a 1 to the CLR_TXF and CLR_RXF bits in MCR.
3. Set the appropriate mode in MCR[MSTR] and enable the DSPI by clearing MCR[HALT].

43.5.3 Initializing DSPI in Master/Slave Modes

Once the appropriate mode in MCR[MSTR] is configured, the DSPI is enabled by clearing MCR[HALT]. It should be ensured that DSPI Slave is enabled before enabling DSPI Master. This ensures the Slave is ready to be communicated with, before Master initializes communication.

43.5.4 Baud rate settings

The following table shows the baud rate that is generated based on the combination of the baud rate prescaler PBR and the baud rate scaler BR in the CTARs. The values calculated assume a 100 MHz system frequency and the double baud rate DBR bit is cleared.

NOTE

> The clock frequency mentioned above is given as an example in this chapter. See the clocking chapter for the frequency used to drive this module in the device.

Table 43-59. Baud rate values (bps)

		Baud rate divider prescaler values			
		2	3	5	7
Baud Rate Scaler Values	2	25.0M	16.7M	10.0M	7.14M
	4	12.5M	8.33M	5.00M	3.57M
	6	8.33M	5.56M	3.33M	2.38M
	8	6.25M	4.17M	2.50M	1.79M
	16	3.12M	2.08M	1.25M	893k
	32	1.56M	1.04M	625k	446k
	64	781k	521k	312k	223k
	128	391k	260k	156k	112k
	256	195k	130k	78.1k	55.8k
	512	97.7k	65.1k	39.1k	27.9k
	1024	48.8k	32.6k	19.5k	14.0k
	2048	24.4k	16.3k	9.77k	6.98k
	4096	12.2k	8.14k	4.88k	3.49k
	8192	6.10k	4.07k	2.44k	1.74k
	16384	3.05k	2.04k	1.22k	872
	32768	1.53k	1.02k	610	436

43.5.5 Delay settings

The following table shows the values for the Delay after Transfer (t_{DT}) and CS to SCK Delay (T_{CSC}) that can be generated based on the prescaler values and the scaler values set in the CTARs. The values calculated assume a 100 MHz system frequency.

NOTE

The clock frequency mentioned above is given as an example in this chapter. See the clocking chapter for the frequency used to drive this module in the device.

Table 43-60. Delay values

		Delay prescaler values			
		1	3	5	7
Delay scaler values	2	20.0 ns	60.0 ns	100.0 ns	140.0 ns
	4	40.0 ns	120.0 ns	200.0 ns	280.0 ns
	8	80.0 ns	240.0 ns	400.0 ns	560.0 ns
	16	160.0 ns	480.0 ns	800.0 ns	1.1 μs
	32	320.0 ns	960.0 ns	1.6 μs	2.2 μs
	64	640.0 ns	1.9 μs	3.2 μs	4.5 μs
	128	1.3 μs	3.8 μs	6.4 μs	9.0 μs
	256	2.6 μs	7.7 μs	12.8 μs	17.9 μs
	512	5.1 μs	15.4 μs	25.6 μs	35.8 μs
	1024	10.2 μs	30.7 μs	51.2 μs	71.7 μs
	2048	20.5 μs	61.4 μs	102.4 μs	143.4 μs
	4096	41.0 μs	122.9 μs	204.8 μs	286.7 μs
	8192	81.9 μs	245.8 μs	409.6 μs	573.4 μs
	16384	163.8 μs	491.5 μs	819.2 μs	1.1 ms
	32768	327.7 μs	983.0 μs	1.6 ms	2.3 ms
	65536	655.4 μs	2.0 ms	3.3 ms	4.6 ms

43.5.6 Calculation of FIFO pointer addresses

Complete visibility of the TX and RX FIFO contents is available through the FIFO registers, and valid entries can be identified through a memory-mapped pointer and counter for each FIFO. The pointer to the first-in entry in each FIFO is memory mapped. For the TX FIFO the first-in pointer is the Transmit Next Pointer (TXNXTPTR). For the RX FIFO the first-in pointer is the Pop Next Pointer (POPNXTPTR). The following figure illustrates the concept of first-in and last-in FIFO entries along with the FIFO Counter. The TX FIFO is chosen for the illustration, but the concepts carry over to the RX FIFO. See Transmit First In First Out (TX FIFO) buffering mechanism and Receive First In First Out (RX FIFO) buffering mechanism for details on the FIFO operation.

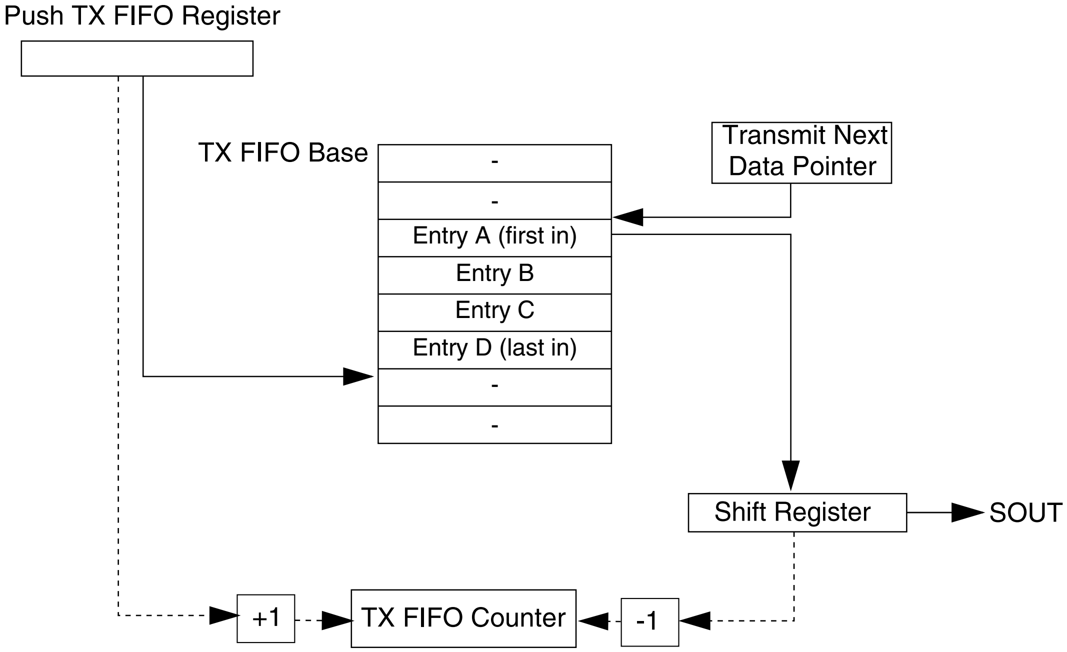

Push TX FIFO Register
TX FIFO Base
-
-
Entry A (first in)
Entry B
Entry C
Entry D (last in)
-
-
Transmit Next Data Pointer
Shift Register
SOUT
+1
TX FIFO Counter
-1

Figure 43-55. TX FIFO pointers and counter

43.5.6.1 Address Calculation for the First-in Entry and Last-in Entry in the TX FIFO

The memory address of the first-in entry in the TX FIFO is computed by the following equation:

$$\text{First-in EntryAddress} = \text{TXFIFOBase} + (4 \times \text{TXNXTPTR})$$

The memory address of the last-in entry in the TX FIFO is computed by the following equation:

$$\text{Last-inEntryaddress} = \text{TXFIFOBase} + 4 \times (\text{TXCTR} + \text{TXNXTPTR} - 1)\text{mod}(\text{TXFIFOdepth})$$

TX FIFO Base - Base address of TX FIFO
TXCTR - TX FIFO Counter
TXNXTPTR - Transmit Next Pointer
TX FIFO Depth - Transmit FIFO depth, implementation specific

43.5.6.2 Address Calculation for the First-in Entry and Last-in Entry in the RX FIFO

The memory address of the first-in entry in the RX FIFO is computed by the following equation:

$$\text{First-in EntryAddress} = \text{RX FIFOBase} + (4 \times \text{POPNXTPTR})$$

The memory address of the last-in entry in the RX FIFO is computed by the following equation:

$$\text{Last-inEntryaddress} = \text{RX FIFO Base} + 4 \times (\text{RXCTR} + \text{POPNXTPTR} - 1)\text{mod}(\text{RXFIFOdepth})$$

RX FIFO Base - Base address of RX FIFO
RXCTR - RX FIFO counter
POPNXTPTR - Pop Next Pointer
RX FIFO Depth - Receive FIFO depth, implementation specific

Chapter 44
Inter-Integrated Circuit (I2C)

44.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The inter-integrated circuit (I^2C, I2C, or IIC) module provides a method of communication between a number of devices. The interface is designed to operate up to 100 kbit/s with maximum bus loading and timing. The I2C device is capable of operating at higher baud rates, up to a maximum of clock/20, with reduced bus loading. The maximum communication length and the number of devices that can be connected are limited by a maximum bus capacitance of 400 pF. The I2C module also complies with the *System Management Bus (SMBus) Specification, version 2*.

44.1.1 Features

The I2C module has the following features:

- Compatible with *The I^2C-Bus Specification*
- Multimaster operation
- Software programmable for one of 64 different serial clock frequencies
- Software-selectable acknowledge bit
- Interrupt-driven byte-by-byte data transfer
- Arbitration-lost interrupt with automatic mode switching from master to slave
- Calling address identification interrupt
- START and STOP signal generation and detection
- Repeated START signal generation and detection
- Acknowledge bit generation and detection
- Bus busy detection
- General call recognition

- 10-bit address extension
- Support for *System Management Bus (SMBus) Specification, version 2*
- Programmable glitch input filter
- Low power mode wakeup on slave address match
- Range slave address support
- DMA support

44.1.2 Modes of operation

The I2C module's operation in various low power modes is as follows:

- Run mode: This is the basic mode of operation. To conserve power in this mode, disable the module.
- Wait mode: The module continues to operate when the core is in Wait mode and can provide a wakeup interrupt.
- Stop mode: The module is inactive in Stop mode for reduced power consumption, except that address matching is enabled in Stop mode. The STOP instruction does not affect the I2C module's register states.

44.1.3 Block diagram

The following figure is a functional block diagram of the I2C module.

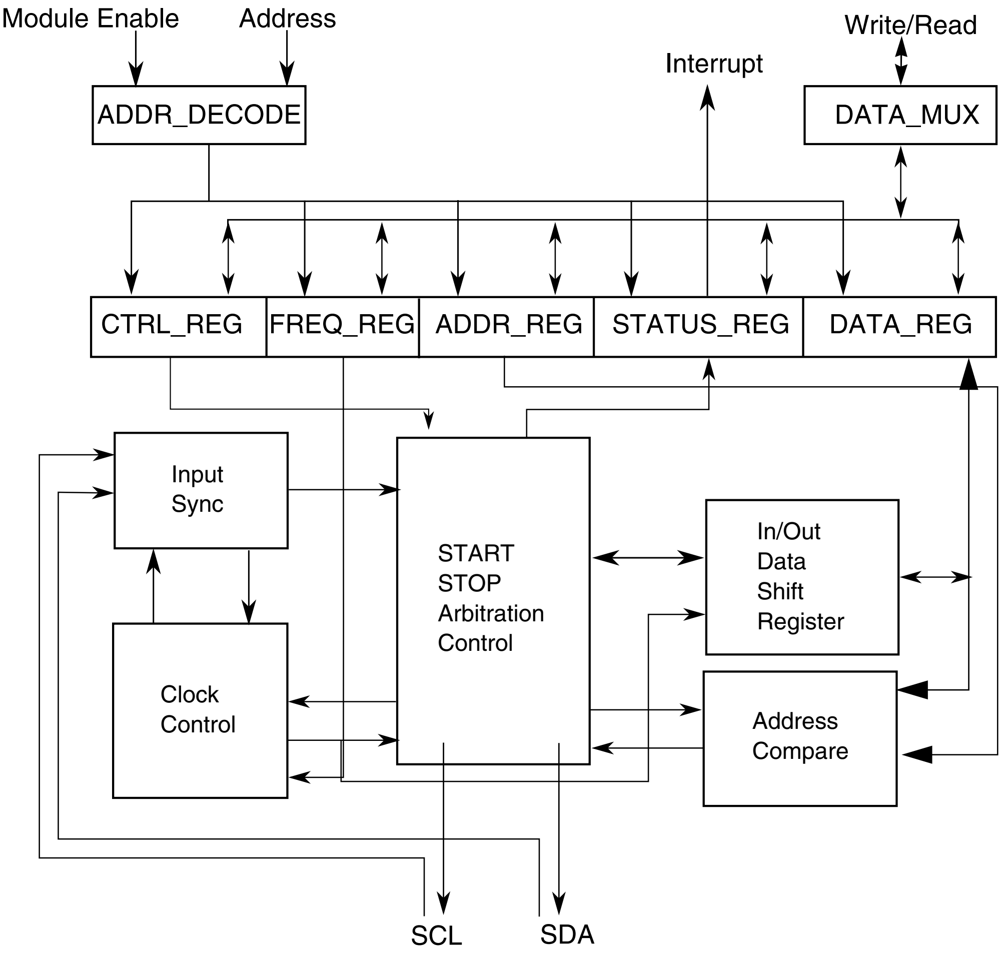

Module Enable
Address
Interrupt
Write/Read
ADDR_DECODE
DATA_MUX
CTRL_REG
FREQ_REG
ADDR_REG
STATUS_REG
DATA_REG
Input Sync
START STOP Arbitration Control
In/Out Data Shift Register
Clock Control
Address Compare
SCL
SDA

Figure 44-1. I2C Functional block diagram

44.2 I²C signal descriptions

The signal properties of I²C are shown in the following table.

Table 44-1. I²C signal descriptions

Signal	Description	I/O
SCL	Bidirectional serial clock line of the I²C system.	I/O
SDA	Bidirectional serial data line of the I²C system.	I/O

44.3 Memory map and register descriptions

This section describes in detail all I2C registers accessible to the end user.

I2C memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_6000	I2C Address Register 1 (I2C0_A1)	8	R/W	00h	44.3.1/ 1012
4006_6001	I2C Frequency Divider register (I2C0_F)	8	R/W	00h	44.3.2/ 1013
4006_6002	I2C Control Register 1 (I2C0_C1)	8	R/W	00h	44.3.3/ 1014
4006_6003	I2C Status register (I2C0_S)	8	R/W	80h	44.3.4/ 1016
4006_6004	I2C Data I/O register (I2C0_D)	8	R/W	00h	44.3.5/ 1017
4006_6005	I2C Control Register 2 (I2C0_C2)	8	R/W	00h	44.3.6/ 1018
4006_6006	I2C Programmable Input Glitch Filter register (I2C0_FLT)	8	R/W	00h	44.3.7/ 1019
4006_6007	I2C Range Address register (I2C0_RA)	8	R/W	00h	44.3.8/ 1020
4006_6008	I2C SMBus Control and Status register (I2C0_SMB)	8	R/W	00h	44.3.9/ 1020
4006_6009	I2C Address Register 2 (I2C0_A2)	8	R/W	C2h	44.3.10/ 1022
4006_600A	I2C SCL Low Timeout Register High (I2C0_SLTH)	8	R/W	00h	44.3.11/ 1022
4006_600B	I2C SCL Low Timeout Register Low (I2C0_SLTL)	8	R/W	00h	44.3.12/ 1023

44.3.1 I2C Address Register 1 (I2Cx_A1)

This register contains the slave address to be used by the I2C module.

Addresses: I2C0_A1 is 4006_6000h base + 0h offset = 4006_6000h

Bit	7	6	5	4	3	2	1	0
Read	AD[7:1]							0
Write								
Reset	0	0	0	0	0	0	0	0

I2Cx_A1 field descriptions

Field	Description
7–1 AD[7:1]	Address Contains the primary slave address used by the I2C module when it is addressed as a slave. This field is used in the 7-bit address scheme and the lower seven bits in the 10-bit address scheme.

Table continues on the next page...

I2Cx_A1 field descriptions (continued)

Field	Description
0 Reserved	This read-only field is reserved and always has the value zero.

44.3.2 I2C Frequency Divider register (I2Cx_F)

Addresses: I2C0_F is 4006_6000h base + 1h offset = 4006_6001h

Bit	7	6	5	4	3	2	1	0
Read Write	MULT		ICR					
Reset	0	0	0	0	0	0	0	0

I2Cx_F field descriptions

Field	Description
7–6 MULT	The MULT bits define the multiplier factor mul. This factor is used along with the SCL divider to generate the I2C baud rate. 00 mul = 1 01 mul = 2 10 mul = 4 11 Reserved
5–0 ICR	ClockRate Prescales the bus clock for bit rate selection. This field and the MULT field determine the I2C baud rate, the SDA hold time, the SCL start hold time, and the SCL stop hold time. For a list of values corresponding to each ICR setting, see I2C divider and hold values. The SCL divider multiplied by multiplier factor (mul) determines the I2C baud rate. $\text{I2C baud rate} = \text{bus speed (Hz)}/(\text{mul} \times \text{SCL divider})$ The SDA hold time is the delay from the falling edge of SCL (I2C clock) to the changing of SDA (I2C data). $\text{SDA hold time} = \text{bus period (s)} \times \text{mul} \times \text{SDA hold value}$ The SCL start hold time is the delay from the falling edge of SDA (I2C data) while SCL is high (start condition) to the falling edge of SCL (I2C clock). $\text{SCL start hold time} = \text{bus period (s)} \times \text{mul} \times \text{SCL start hold value}$ The SCL stop hold time is the delay from the rising edge of SCL (I2C clock) to the rising edge of SDA (I2C data) while SCL is high (stop condition). $\text{SCL stop hold time} = \text{bus period (s)} \times \text{mul} \times \text{SCL stop hold value}$ For example, if the bus speed is 8 MHz, the following table shows the possible hold time values with different ICR and MULT selections to achieve an I^2C baud rate of 100 kbps.

Table continues on the next page...

I2Cx_F field descriptions (continued)

Field	Description				
	MULT	ICR	Hold times (μs)		
			SDA	SCL Start	SCL Stop
	2h	00h	3.500	3.000	5.500
	1h	07h	2.500	4.000	5.250
	1h	0Bh	2.250	4.000	5.250
	0h	14h	2.125	4.250	5.125
	0h	18h	1.125	4.750	5.125

44.3.3 I2C Control Register 1 (I2Cx_C1)

Addresses: I2C0_C1 is 4006_6000h base + 2h offset = 4006_6002h

Bit	7	6	5	4	3	2	1	0
Read	IICEN	IICIE	MST	TX	TXAK	0	WUEN	DMAEN
Write						RSTA		
Reset	0	0	0	0	0	0	0	0

I2Cx_C1 field descriptions

Field	Description
7 IICEN	I2C Enable Enables I2C module operation. 0 Disabled 1 Enabled
6 IICIE	I2C Interrupt Enable Enables I2C interrupt requests. 0 Disabled 1 Enabled
5 MST	Master Mode Select When the MST bit is changed from a 0 to a 1, a START signal is generated on the bus and master mode is selected. When this bit changes from a 1 to a 0, a STOP signal is generated and the mode of operation changes from master to slave. 0 Slave mode 1 Master mode
4 TX	Transmit Mode Select

Table continues on the next page...

I2Cx_C1 field descriptions (continued)

Field	Description
	Selects the direction of master and slave transfers. In master mode this bit must be set according to the type of transfer required. Therefore, for address cycles, this bit is always set. When addressed as a slave this bit must be set by software according to the SRW bit in the status register. 0 Receive 1 Transmit
3 TXAK	Transmit Acknowledge Enable Specifies the value driven onto the SDA during data acknowledge cycles for both master and slave receivers. The value of the FACK bit affects NACK/ACK generation. 0 An acknowledge signal is sent to the bus on the following (if FACK is cleared) or current (if FACK is set) receiving byte. 1 No acknowledge signal is sent to the bus on the following (if FACK is cleared) or current (if FACK is set) receiving data byte. **NOTE:** SCL is held low until TXAK is written.
2 RSTA	Repeat START Writing a one to this bit generates a repeated START condition provided it is the current master. This bit will always be read as zero. Attempting a repeat at the wrong time results in loss of arbitration.
1 WUEN	Wakeup Enable The I2C module can wake the MCU from low power mode with no peripheral bus running when slave address matching occurs. 0 Normal operation. No interrupt generated when address matching in low power mode. 1 Enables the wakeup function in low power mode.
0 DMAEN	DMA Enable The DMAEN bit enables or disables the DMA function. 0 All DMA signalling disabled. 1 DMA transfer is enabled and the following conditions trigger the DMA request: • While FACK = 0, a data byte is received, either address or data is transmitted. (ACK/NACK automatic) • While FACK = 0, the first byte received matches the A1 register or is general call address. If any address matching occurs, IAAS and TCF are set. If the direction of transfer is known from master to slave, then it is not required to check the SRW. With this assumption, DMA can also be used in this case. In other cases, if the master reads data from the slave, then it is required to rewrite the C1 register operation. With this assumption, DMA cannot be used. When FACK = 1, an address or a data byte is transmitted.

44.3.4 I2C Status register (I2Cx_S)

Addresses: I2C0_S is 4006_6000h base + 3h offset = 4006_6003h

Bit	7	6	5	4	3	2	1	0
Read	TCF	IAAS	BUSY	ARBL	RAM	SRW	IICIF	RXAK
Write	▨		▨	w1c		▨	w1c	▨
Reset	1	0	0	0	0	0	0	0

I2Cx_S field descriptions

Field	Description
7 TCF	Transfer Complete Flag This bit sets on the completion of a byte and acknowledge bit transfer. This bit is valid only during or immediately following a transfer to or from the I2C module. The TCF bit is cleared by reading the I2C data register in receive mode or by writing to the I2C data register in transmit mode. 0 Transfer in progress 1 Transfer complete
6 IAAS	Addressed As A Slave This bit is set by one of the following conditions: • The calling address matches the programmed slave primary address in the A1 register or range address in the RA register (which must be set to a nonzero value). • GCAEN is set and a general call is received. • SIICAEN is set and the calling address matches the second programmed slave address. • ALERTEN is set and an SMBus alert response address is received • RMEN is set and an address is received that is within the range between the values of the A1 and RA registers. This bit sets before the ACK bit. The CPU must check the SRW bit and set TX/RX accordingly. Writing the C1 register with any value clears this bit. 0 Not addressed 1 Addressed as a slave
5 BUSY	Bus Busy Indicates the status of the bus regardless of slave or master mode. This bit is set when a START signal is detected and cleared when a STOP signal is detected. 0 Bus is idle 1 Bus is busy
4 ARBL	Arbitration Lost This bit is set by hardware when the arbitration procedure is lost. The ARBL bit must be cleared by software, by writing a one to it. 0 Standard bus operation. 1 Loss of arbitration.
3 RAM	Range Address Match

Table continues on the next page...

I2Cx_S field descriptions (continued)

Field	Description
	This bit is set by any of the following conditions: • Any nonzero calling address is received that matches the address in the RA register. • The RMEN bit is set and the calling address is within the range of values of the A1 and RA registers. Writing the C1 register with any value clears this bit. 0 Not addressed 1 Addressed as a slave
2 SRW	Slave Read/Write When addressed as a slave, SRW indicates the value of the $R/\overline{W}$ command bit of the calling address sent to the master. 0 Slave receive, master writing to slave 1 Slave transmit, master reading from slave
1 IICIF	Interrupt Flag This bit sets when an interrupt is pending. This bit must be cleared by software or by writing a 1 to it in the interrupt routine. One of the following events can set this bit: • One byte transfer including ACK/NACK bit completes if FACK = 0 • One byte transfer excluding ACK/NACK bit completes if FACK = 1. An ACK or NACK is sent on the bus by writing 0 or 1 to TXAK after this bit is set in receive mode • Match of slave address to calling address including primary slave address, range slave address, alert response address, second slave address, or general call address. • Arbitration lost • In SMBus mode, any timeouts except SCL and SDA high timeouts 0 No interrupt pending 1 Interrupt pending
0 RXAK	Receive Acknowledge 0 Acknowledge signal was received after the completion of one byte of data transmission on the bus 1 No acknowledge signal detected

44.3.5 I2C Data I/O register (I2Cx_D)

Addresses: I2C0_D is 4006_6000h base + 4h offset = 4006_6004h

Bit	7	6	5	4	3	2	1	0
Read Write	DATA							
Reset	0	0	0	0	0	0	0	0

I2Cx_D field descriptions

Field	Description
7–0 DATA	Data

I2Cx_D field descriptions (continued)

Field	Description
	In master transmit mode, when data is written to this register, a data transfer is initiated. The most significant bit is sent first. In master receive mode, reading this register initiates receiving of the next byte of data. **NOTE:** When making the transition out of master receive mode, switch the I2C mode before reading the Data register to prevent an inadvertent initiation of a master receive data transfer. In slave mode, the same functions are available after an address match occurs. The C1[TX] bit must correctly reflect the desired direction of transfer in master and slave modes for the transmission to begin. For example, if the I2C module is configured for master transmit but a master receive is desired, reading the Data register does not initiate the receive. Reading the Data register returns the last byte received while the I2C module is configured in master receive or slave receive mode. The Data register does not reflect every byte that is transmitted on the I2C bus, and neither can software verify that a byte has been written to the Data register correctly by reading it back. In master transmit mode, the first byte of data written to the Data register following assertion of MST (start bit) or assertion of RSTA (repeated start bit) is used for the address transfer and must consist of the calling address (in bits 7-1) concatenated with the required $R/\overline{W}$ bit (in position bit 0).

44.3.6 I2C Control Register 2 (I2Cx_C2)

Addresses: I2C0_C2 is 4006_6000h base + 5h offset = 4006_6005h

Bit	7	6	5	4	3	2	1	0
Read Write	GCAEN	ADEXT	HDRS	SBRC	RMEN	AD[10:8]		
Reset	0	0	0	0	0	0	0	0

I2Cx_C2 field descriptions

Field	Description
7 GCAEN	General Call Address Enable Enables general call address. 0 Disabled 1 Enabled
6 ADEXT	Address Extension Controls the number of bits used for the slave address. 0 7-bit address scheme 1 10-bit address scheme
5 HDRS	High Drive Select Controls the drive capability of the I2C pads. 0 Normal drive mode 1 High drive mode

Table continues on the next page...

I2Cx_C2 field descriptions (continued)

Field	Description
4 SBRC	Slave Baud Rate Control Enables independent slave mode baud rate at maximum frequency, which forces clock stretching on SCL in very fast I2C modes. To a slave, an example of a "very fast" mode is when the master transfers at 40 kbps but the slave can capture the master's data at only 10 kbps. 0 The slave baud rate follows the master baud rate and clock stretching may occur 1 Slave baud rate is independent of the master baud rate
3 RMEN	Range Address Matching Enable This bit controls slave address matching for addresses between the values of the A1 and RA registers. When this bit is set, a slave address match occurs for any address greater than the value of the A1 register and less than or equal to the value of the RA register. 0 Range mode disabled. No address match occurs for an address within the range of values of the A1 and RA registers. 1 Range mode enabled. Address matching occurs when a slave receives an address within the range of values of the A1 and RA registers.
2–0 AD[10:8]	Slave Address Contains the upper three bits of the slave address in the 10-bit address scheme. This field is valid only while the ADEXT bit is set.

44.3.7 I2C Programmable Input Glitch Filter register (I2Cx_FLT)

Addresses: I2C0_FLT is 4006_6000h base + 6h offset = 4006_6006h

Bit	7	6	5	4	3	2	1	0
Read	Reserved	0		FLT				
Write		▨						
Reset	0	0	0	0	0	0	0	0

I2Cx_FLT field descriptions

Field	Description
7 Reserved	This field is reserved.
6–5 Reserved	This read-only field is reserved and always has the value zero.
4–0 FLT	I2C Programmable Filter Factor Controls the width of the glitch, in terms of bus clock cycles, that the filter must absorb. For any glitch whose size is less than or equal to this width setting, the filter does not allow the glitch to pass. 00h No filter/bypass 01-1Fh Filter glitches up to width of *n* bus clock cycles, where *n*=1-31d

44.3.8 I2C Range Address register (I2Cx_RA)

Addresses: I2C0_RA is 4006_6000h base + 7h offset = 4006_6007h

Bit	7	6	5	4	3	2	1	0
Read	RAD							0
Write								▨
Reset	0	0	0	0	0	0	0	0

I2Cx_RA field descriptions

Field	Description
7–1 RAD	Range Slave Address This field contains the slave address to be used by the I2C module. The field is used in the 7-bit address scheme. Any nonzero write enables this register. This register's use is similar to that of the A1 register, but in addition this register can be considered a maximum boundary in range matching mode.
0 Reserved	This read-only field is reserved and always has the value zero.

44.3.9 I2C SMBus Control and Status register (I2Cx_SMB)

NOTE

When the SCL and SDA signals are held high for a length of time greater than the high timeout period, the SHTF1 flag sets. Before reaching this threshold, while the system is detecting how long these signals are being held high, a master assumes that the bus is free. However, the SHTF1 bit rises in the bus transmission process with the idle bus state.

NOTE

When the TCKSEL bit is set, there is no need to monitor the SHTF1 bit because the bus speed is too high to match the protocol of SMBus.

Addresses: I2C0_SMB is 4006_6000h base + 8h offset = 4006_6008h

Bit	7	6	5	4	3	2	1	0
Read	FACK	ALERTEN	SIICAEN	TCKSEL	SLTF	SHTF1	SHTF2	SHTF2IE
Write					w1c	▨	w1c	
Reset	0	0	0	0	0	0	0	0

I2Cx_SMB field descriptions

Field	Description
7 FACK	Fast NACK/ACK Enable For SMBus packet error checking, the CPU must be able to issue an ACK or NACK according to the result of receiving data byte. 0 An ACK or NACK is sent on the following receiving data byte 1 Writing 0 to TXAK after receiving a data byte generates an ACK. Writing 1 to TXAK after receiving a data byte generates a NACK.
6 ALERTEN	SMBus Alert Response Address Enable Enables or disables SMBus alert response address matching. **NOTE:** After the host responds to a device that used the alert response address, you must use software to put the device's address on the bus. The alert protocol is described in the SMBus specification. 0 SMBus alert response address matching is disabled 1 SMBus alert response address matching is enabled
5 SIICAEN	Second I2C Address Enable Enables or disables SMBus device default address. 0 I2C address register 2 matching is disabled 1 I2C address register 2 matching is enabled
4 TCKSEL	Timeout Counter Clock Select Selects the clock source of the timeout counter. 0 Timeout counter counts at the frequency of the bus clock / 64 1 Timeout counter counts at the frequency of the bus clock
3 SLTF	SCL Low Timeout Flag This bit is set when the SLT register (consisting of the SLTH and SLTL registers) is loaded with a non-zero value (LoValue) and an SCL low timeout occurs. Software clears this bit by writing a logic 1 to it. **NOTE:** The low timeout function is disabled when the SLT register's value is zero. 0 No low timeout occurs 1 Low timeout occurs
2 SHTF1	SCL High Timeout Flag 1 This read-only bit sets when SCL and SDA are held high more than clock × LoValue / 512, which indicates the bus is free. This bit is cleared automatically. 0 No SCL high and SDA high timeout occurs 1 SCL high and SDA high timeout occurs
1 SHTF2	SCL High Timeout Flag 2 This bit sets when SCL is held high and SDA is held low more than clock × LoValue/512. Software clears this bit by writing a 1 to it.

Table continues on the next page...

I2Cx_SMB field descriptions (continued)

Field	Description
	0 No SCL high and SDA low timeout occurs 1 SCL high and SDA low timeout occurs
0 SHTF2IE	SHTF2 Interrupt Enable Enables SCL high and SDA low timeout interrupt. 0 SHTF2 interrupt is disabled 1 SHTF2 interrupt is enabled

44.3.10 I2C Address Register 2 (I2Cx_A2)

Addresses: I2C0_A2 is 4006_6000h base + 9h offset = 4006_6009h

Bit	7	6	5	4	3	2	1	0
Read	SAD							0
Write								▨
Reset	1	1	0	0	0	0	1	0

I2Cx_A2 field descriptions

Field	Description
7–1 SAD	SMBus Address Contains the slave address used by the SMBus. This field is used on the device default address or other related addresses.
0 Reserved	This read-only field is reserved and always has the value zero.

44.3.11 I2C SCL Low Timeout Register High (I2Cx_SLTH)

Addresses: I2C0_SLTH is 4006_6000h base + Ah offset = 4006_600Ah

Bit	7	6	5	4	3	2	1	0
Read Write	SSLT[15:8]							
Reset	0	0	0	0	0	0	0	0

I2Cx_SLTH field descriptions

Field	Description
7–0 SSLT[15:8]	Most significant byte of SCL low timeout value that determines the timeout period of SCL low.

44.3.12 I2C SCL Low Timeout Register Low (I2Cx_SLTL)

Addresses: I2C0_SLTL is 4006_6000h base + Bh offset = 4006_600Bh

Bit	7	6	5	4	3	2	1	0
Read Write	SSLT[7:0]							
Reset	0	0	0	0	0	0	0	0

I2Cx_SLTL field descriptions

Field	Description
7–0 SSLT[7:0]	Least significant byte of SCL low timeout value that determines the timeout period of SCL low.

44.4 Functional description

This section provides a comprehensive functional description of the I2C module.

44.4.1 I2C protocol

The I2C bus system uses a serial data line (SDA) and a serial clock line (SCL) for data transfers. All devices connected to it must have open drain or open collector outputs. A logic AND function is exercised on both lines with external pull-up resistors. The value of these resistors depends on the system.

Normally, a standard instance of communication is composed of four parts:

1. START signal
2. Slave address transmission
3. Data transfer
4. STOP signal

The STOP signal should not be confused with the CPU STOP instruction. The following figure illustrates I2C bus system communication.

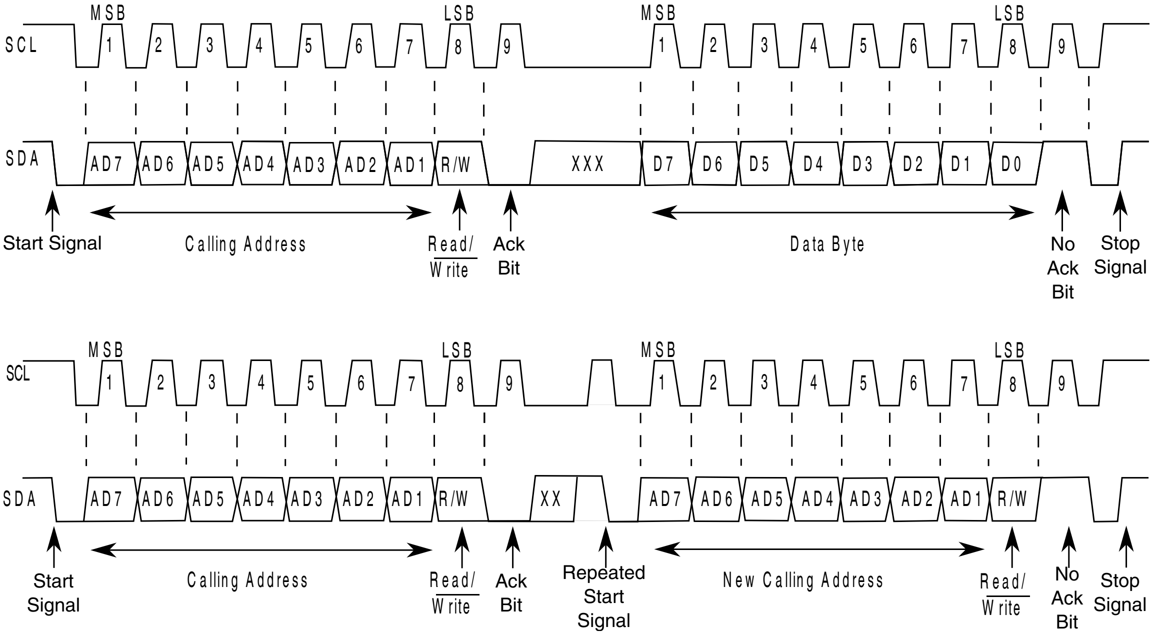

MSB
LSB
MSB
LSB
SCL
1
2
3
4
5
6
7
8
9
SDA
AD7
AD6
AD5
AD4
AD3
AD2
AD1
R/W
XXX
D7
D6
D5
D4
D3
D2
D1
D0
Start Signal
Calling Address
Read/ Write
Ack Bit
Data Byte
No Ack Bit
Stop Signal
XX
Start Signal
Calling Address
Read/ Write
Ack Bit
Repeated Start Signal
New Calling Address
Read/ Write
No Ack Bit
Stop Signal

Figure 44-26. I2C bus transmission signals

44.4.1.1 START signal

The bus is free when no master device is engaging the bus (both SCL and SDA are high). When the bus is free, a master may initiate communication by sending a START signal. A START signal is defined as a high-to-low transition of SDA while SCL is high. This signal denotes the beginning of a new data transfer—each data transfer might contain several bytes of data—and brings all slaves out of their idle states.

44.4.1.2 Slave address transmission

Immediately after the START signal, the first byte of a data transfer is the slave address transmitted by the master. This address is a 7-bit calling address followed by an R/$\overline{\text{W}}$ bit. The R/$\overline{\text{W}}$ bit tells the slave the desired direction of data transfer.

- 1 = Read transfer: The slave transmits data to the master
- 0 = Write transfer: The master transmits data to the slave

Only the slave with a calling address that matches the one transmitted by the master responds by sending an acknowledge bit. The slave sends the acknowledge bit by pulling SDA low at the ninth clock.

No two slaves in the system can have the same address. If the I2C module is the master, it must not transmit an address that is equal to its own slave address. The I2C module cannot be master and slave at the same time. However, if arbitration is lost during an address cycle, the I2C module reverts to slave mode and operates correctly even if it is being addressed by another master.

44.4.1.3 Data transfers

When successful slave addressing is achieved, data transfer can proceed on a byte-by-byte basis in the direction specified by the R/$\overline{W}$ bit sent by the calling master.

All transfers that follow an address cycle are referred to as data transfers, even if they carry subaddress information for the slave device.

Each data byte is 8 bits long. Data may be changed only while SCL is low. Data must be held stable while SCL is high. There is one clock pulse on SCL for each data bit, and the MSB is transferred first. Each data byte is followed by a ninth (acknowledge) bit, which is signaled from the receiving device by pulling SDA low at the ninth clock. In summary, one complete data transfer needs nine clock pulses.

If the slave receiver does not acknowledge the master in the ninth bit, the slave must leave SDA high. The master interprets the failed acknowledgement as an unsuccessful data transfer.

If the master receiver does not acknowledge the slave transmitter after a data byte transmission, the slave interprets it as an end to data transfer and releases the SDA line.

In the case of a failed acknowledgement by either the slave or master, the data transfer is aborted and the master does one of two things:

- Relinquishes the bus by generating a STOP signal.
- Commences a new call by generating a repeated START signal.

44.4.1.4 STOP signal

The master can terminate the communication by generating a STOP signal to free the bus. A STOP signal is defined as a low-to-high transition of SDA while SCL is asserted.

The master can generate a STOP signal even if the slave has generated an acknowledgement, at which point the slave must release the bus.

44.4.1.5 Repeated START signal

The master may generate a START signal followed by a calling command without generating a STOP signal first. This action is called a repeated START. The master uses a repeated START to communicate with another slave or with the same slave in a different mode (transmit/receive mode) without releasing the bus.

44.4.1.6 Arbitration procedure

The I2C bus is a true multimaster bus that allows more than one master to be connected on it.

If two or more masters try to control the bus at the same time, a clock synchronization procedure determines the bus clock. The bus clock's low period is equal to the longest clock low period, and the high period is equal to the shortest one among the masters.

The relative priority of the contending masters is determined by a data arbitration procedure. A bus master loses arbitration if it transmits logic level 1 while another master transmits logic level 0. The losing masters immediately switch to slave receive mode and stop driving SDA output. In this case, the transition from master to slave mode does not generate a STOP condition. Meanwhile, hardware sets a status bit to indicate the loss of arbitration.

44.4.1.7 Clock synchronization

Because wire AND logic is performed on SCL, a high-to-low transition on SCL affects all devices connected on the bus. The devices start counting their low period and, after a device's clock has gone low, that device holds SCL low until the clock reaches its high state. However, the change of low to high in this device clock might not change the state of SCL if another device clock is still within its low period. Therefore, the synchronized clock SCL is held low by the device with the longest low period. Devices with shorter low periods enter a high wait state during this time; see the following diagram. When all applicable devices have counted off their low period, the synchronized clock SCL is released and pulled high. Afterward there is no difference between the device clocks and the state of SCL, and all devices start counting their high periods. The first device to complete its high period pulls SCL low again.

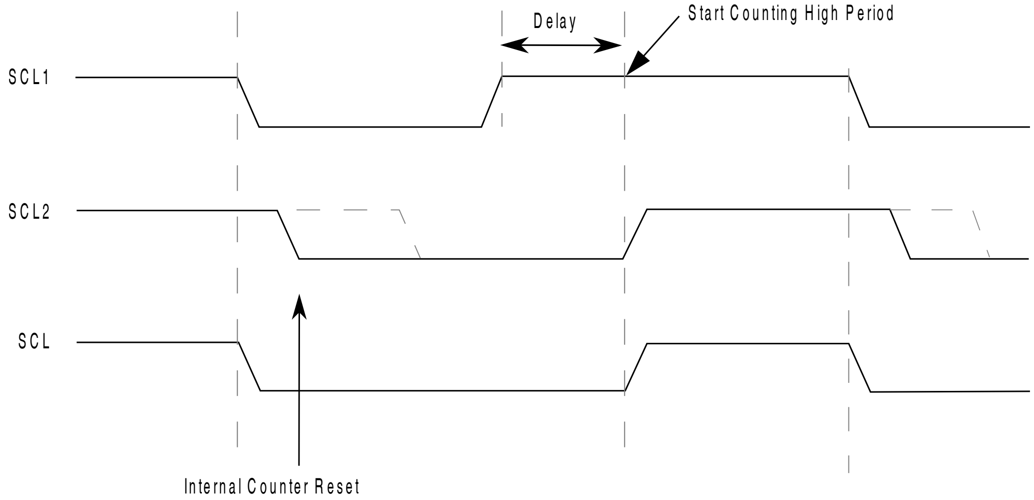


Figure 44-27. I2C clock synchronization

44.4.1.8 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfers. A slave device may hold SCL low after completing a single byte transfer (9 bits). In this case, it halts the bus clock and forces the master clock into wait states until the slave releases SCL.

44.4.1.9 Clock stretching

The clock synchronization mechanism can be used by slaves to slow down the bit rate of a transfer. After the master drives SCL low, a slave can drive SCL low for the required period and then release it. If the slave's SCL low period is greater than the master's SCL low period, the resulting SCL bus signal's low period is stretched. In other words, the SCL bus signal's low period is increased to be the same length as the slave's SCL low period.

44.4.1.10 I2C divider and hold values

Table 44-28. I2C divider and hold values

ICR (hex)	SCL divider	SDA hold value	SCL hold (start) value	SCL hold (stop) value		ICR (hex)	SCL divider (clocks)	SDA hold (clocks)	SCL hold (start) value	SCL hold (stop) value
00	20	7	6	11		20	160	17	78	81
01	22	7	7	12		21	192	17	94	97

Table continues on the next page...

Table 44-28. I2C divider and hold values (continued)

ICR (hex)	SCL divider	SDA hold value	SCL hold (start) value	SCL hold (stop) value		ICR (hex)	SCL divider (clocks)	SDA hold (clocks)	SCL hold (start) value	SCL hold (stop) value
02	24	8	8	13		22	224	33	110	113
03	26	8	9	14		23	256	33	126	129
04	28	9	10	15		24	288	49	142	145
05	30	9	11	16		25	320	49	158	161
06	34	10	13	18		26	384	65	190	193
07	40	10	16	21		27	480	65	238	241
08	28	7	10	15		28	320	33	158	161
09	32	7	12	17		29	384	33	190	193
0A	36	9	14	19		2A	448	65	222	225
0B	40	9	16	21		2B	512	65	254	257
0C	44	11	18	23		2C	576	97	286	289
0D	48	11	20	25		2D	640	97	318	321
0E	56	13	24	29		2E	768	129	382	385
0F	68	13	30	35		2F	960	129	478	481
10	48	9	18	25		30	640	65	318	321
11	56	9	22	29		31	768	65	382	385
12	64	13	26	33		32	896	129	446	449
13	72	13	30	37		33	1024	129	510	513
14	80	17	34	41		34	1152	193	574	577
15	88	17	38	45		35	1280	193	638	641
16	104	21	46	53		36	1536	257	766	769
17	128	21	58	65		37	1920	257	958	961
18	80	9	38	41		38	1280	129	638	641
19	96	9	46	49		39	1536	129	766	769
1A	112	17	54	57		3A	1792	257	894	897
1B	128	17	62	65		3B	2048	257	1022	1025
1C	144	25	70	73		3C	2304	385	1150	1153
1D	160	25	78	81		3D	2560	385	1278	1281
1E	192	33	94	97		3E	3072	513	1534	1537
1F	240	33	118	121		3F	3840	513	1918	1921

44.4.2 10-bit address

For 10-bit addressing, 0x11110 is used for the first 5 bits of the first address byte. Various combinations of read/write formats are possible within a transfer that includes 10-bit addressing.

44.4.2.1 Master-transmitter addresses a slave-receiver

The transfer direction is not changed. When a 10-bit address follows a START condition, each slave compares the first 7 bits of the first byte of the slave address (11110XX) with its own address and tests whether the eighth bit (R/$\overline{W}$ direction bit) is 0. It is possible that more than one device finds a match and generates an acknowledge (A1). Each slave that finds a match compares the 8 bits of the second byte of the slave address with its own address, but only one slave finds a match and generates an acknowledge (A2). The matching slave remains addressed by the master until it receives a STOP condition (P) or a repeated START condition (Sr) followed by a different slave address.

Table 44-29. Master-transmitter addresses slave-receiver with a 10-bit address

S	Slave address first 7 bits 11110 + AD10 + AD9	R/$\overline{W}$ 0	A1	Slave address second byte AD[8:1]	A2	Data	A	...	Data	A/A	P

After the master-transmitter has sent the first byte of the 10-bit address, the slave-receiver sees an I2C interrupt. User software must ensure that for this interrupt, the contents of the Data register are ignored and not treated as valid data.

44.4.2.2 Master-receiver addresses a slave-transmitter

The transfer direction is changed after the second R/$\overline{W}$ bit. Up to and including acknowledge bit A2, the procedure is the same as that described for a master-transmitter addressing a slave-receiver. After the repeated START condition (Sr), a matching slave remembers that it was addressed before. This slave then checks whether the first seven bits of the first byte of the slave address following Sr are the same as they were after the START condition (S), and it tests whether the eighth (R/$\overline{W}$) bit is 1. If there is a match, the slave considers that it has been addressed as a transmitter and generates acknowledge A3. The slave-transmitter remains addressed until it receives a STOP condition (P) or a repeated START condition (Sr) followed by a different slave address.

After a repeated START condition (Sr), all other slave devices also compare the first seven bits of the first byte of the slave address with their own addresses and test the eighth ($R/\overline{W}$) bit. However, none of them are addressed because $R/\overline{W} = 1$ (for 10-bit devices), or the 11110XX slave address (for 7-bit devices) does not match.

Table 44-30. Master-receiver addresses a slave-transmitter with a 10-bit address

S	Slave address first 7 bits 11110 + AD10 + AD9	$R/\overline{W}$ 0	A1	Slave address second byte AD[8:1]	A2	Sr	Slave address first 7 bits 11110 + AD10 + AD9	$R/\overline{W}$ 1	A3	Data	A	...	Data	A	P

After the master-receiver has sent the first byte of the 10-bit address, the slave-transmitter sees an I2C interrupt. User software must ensure that for this interrupt, the contents of the Data register are ignored and not treated as valid data.

44.4.3 Address matching

All received addresses can be requested in 7-bit or 10-bit address format. The Address Register 1, which contains the I2C primary slave address, always participates in the address matching process. If the GCAEN bit is set, general call participates the address matching process. If the ALERTEN bit is set, alert response participates the address matching process. If the SIICAEN bit is set, the Address Register 2 participates in the address matching process. If the Range Address register is programmed to a nonzero value, the range address itself participates in the address matching process. If the RMEN bit is set, any address within the range of values of the Address Register 1 and the Range Address register participates in the address matching process. The Range Address register must be programmed to a value greater than the value of the Address Register 1.

When the I2C module responds to one of these addresses, it acts as a slave-receiver and the IAAS bit is set after the address cycle. Software must read the Data register after the first byte transfer to determine that the address is matched.

44.4.4 System management bus specification

SMBus provides a control bus for system and power management related tasks. A system can use SMBus to pass messages to and from devices instead of tripping individual control lines. Removing the individual control lines reduces pin count. Accepting messages ensures future expandability. With the system management bus, a device can provide manufacturer information, tell the system what its model/part number is, save its state for a suspend event, report different types of errors, accept control parameters, and return its status.

44.4.4.1 Timeouts

The $T_{TIMEOUT,MIN}$ parameter allows a master or slave to conclude that a defective device is holding the clock low indefinitely or a master is intentionally trying to drive devices off the bus. The slave device must release the bus (stop driving the bus and let SCL and SDA float high) when it detects any single clock held low longer than $T_{TIMEOUT,MIN}$. Devices that have detected this condition must reset their communication and be able to receive a new START condition within the timeframe of $T_{TIMEOUT,MAX}$.

SMBus defines a clock low timeout, $T_{TIMEOUT}$, of 35 ms, specifies $T_{LOW:SEXT}$ as the cumulative clock low extend time for a slave device, and specifies $T_{LOW:MEXT}$ as the cumulative clock low extend time for a master device.

44.4.4.1.1 SCL low timeout

If the SCL line is held low by a slave device on the bus, no further communication is possible. Furthermore, the master cannot force the SCL line high to correct the error condition. To solve this problem, the SMBus protocol specifies that devices participating in a transfer must detect any clock cycle held low longer than a timeout value condition. Devices that have detected the timeout condition must reset the communication. When the I2C module is an active master, if it detects that SMBCLK low has exceeded the value of $T_{TIMEOUT,MIN}$, it must generate a stop condition within or after the current data byte in the transfer process. When the I2C module is a slave, if it detects the $T_{TIMEOUT,MIN}$ condition, it resets its communication and is then able to receive a new START condition.

44.4.4.1.2 SCL high timeout

When the I2C module has determined that the SMBCLK and SMBDAT signals have been high for at least $T_{HIGH:MAX}$, it assumes that the bus is idle.

A HIGH timeout occurs after a START condition appears on the bus but before a STOP condition appears on the bus. Any master detecting this scenario can assume the bus is free when either of the following occurs:

- SHTF1 rises.
- The BUSY bit is high and SHTF1 is high.

When the SMBDAT signal is low and the SMBCLK signal is high for a period of time, another kind of timeout occurs. The time period must be defined in software. SHTF2 is used as the flag when the time limit is reached. This flag is also an interrupt resource, so it triggers IICIF.

44.4.4.1.3 CSMBCLK TIMEOUT MEXT and CSMBCLK TIMEOUT SEXT

The following figure illustrates the definition of the timeout intervals $T_{LOW:SEXT}$ and $T_{LOW:MEXT}$. When in master mode, the I2C module must not cumulatively extend its clock cycles for a period greater than $T_{LOW:MEXT}$ within a byte, where each byte is defined as START-to-ACK, ACK-to-ACK, or ACK-to-STOP. When CSMBCLK TIMEOUT MEXT occurs, SMBus MEXT rises and also triggers the SLTF.

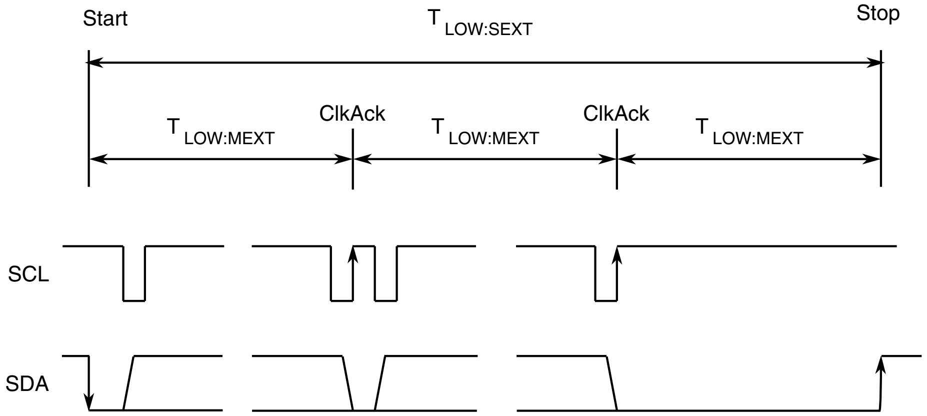


Figure 44-28. Timeout measurement intervals

A master is allowed to abort the transaction in progress to any slave that violates the $T_{LOW:SEXT}$ or $T_{TIMEOUT,MIN}$ specifications. To abort the transaction, the master issues a STOP condition at the conclusion of the byte transfer in progress. When a slave, the I2C module must not cumulatively extend its clock cycles for a period greater than $T_{LOW:SEXT}$ during any message from the initial START to the STOP. When CSMBCLK TIMEOUT SEXT occurs, SEXT rises and also triggers SLTF.

NOTE

CSMBCLK TIMEOUT SEXT and CSMBCLK TIMEOUT MEXT are optional functions that are implemented in the second step.

44.4.4.2 FAST ACK and NACK

To improve reliability and communication robustness, implementation of packet error checking (PEC) by SMBus devices is optional for SMBus devices but required for devices participating in and only during the address resolution protocol (ARP) process.

The PEC is a CRC-8 error checking byte, calculated on all the message bytes. The PEC is appended to the message by the device that supplied the last data byte. If the PEC is present but not correct, a NACK is issued by the receiver. Otherwise an ACK is issued. To calculate the CRC-8 by software, this module can hold the SCL line low after receiving the eighth SCL (8th bit) if this byte is a data byte. So software can determine whether an ACK or NACK should be sent to the bus by setting or clearing the TXAK bit if the FACK (fast ACK/NACK enable) bit is enabled.

SMBus requires a device always to acknowledge its own address, as a mechanism to detect the presence of a removable device (such as a battery or docking station) on the bus. In addition to indicating a slave device busy condition, SMBus uses the NACK mechanism to indicate the reception of an invalid command or invalid data. Because such a condition may occur on the last byte of the transfer, SMBus devices are required to have the ability to generate the not acknowledge after the transfer of each byte and before the completion of the transaction. This requirement is important because SMBus does not provide any other resend signaling. This difference in the use of the NACK signaling has implications on the specific implementation of the SMBus port, especially in devices that handle critical system data such as the SMBus host and the SBS components.

NOTE

In the last byte of master receive slave transmit mode, the master must send a NACK to the bus, so FACK must be switched off before the last byte transmits.

44.4.5 Resets

The I2C module is disabled after a reset. The I2C module cannot cause a core reset.

44.4.6 Interrupts

The I2C module generates an interrupt when any of the events in the following table occur, provided that the IICIE bit is set. The interrupt is driven by the IICIF bit (of the I2C Status Register) and masked with the IICIE bit (of the I2C Control Register 1). The IICIF bit must be cleared (by software) by writing 1 to it in the interrupt routine. The SMBus timeouts interrupt is driven by SLTF and masked with the IICIE bit. The SLTF bit must be cleared by software by writing 1 to it in the interrupt routine. You can determine the interrupt type by reading the Status Register.

NOTE

In master receive mode, the FACK bit must be set to zero before the last byte transfer.

Table 44-31. Interrupt summary

Interrupt source	Status	Flag	Local enable
Complete 1-byte transfer	TCF	IICIF	IICIE
Match of received calling address	IAAS	IICIF	IICIE
Arbitration lost	ARBL	IICIF	IICIE
SMBus SCL low timeout	SLTF	IICIF	IICIE
SMBus SCL high SDA low timeout	SHTF2	IICIF	IICIE & SHTF2IE
Wakeup from stop or wait mode	IAAS	IICIF	IICIE & WUEN

44.4.6.1 Byte transfer interrupt

The Transfer Complete Flag (TCF) bit is set at the falling edge of the ninth clock to indicate the completion of a byte and acknowledgement transfer. When FACK is enabled, TCF is then set at the falling edge of eighth clock to indicate the completion of byte.

44.4.6.2 Address detect interrupt

When the calling address matches the programmed slave address (I2C Address Register) or when the GCAEN bit is set and a general call is received, the IAAS bit in the Status Register is set. The CPU is interrupted, provided the IICIE bit is set. The CPU must check the SRW bit and set its Tx mode accordingly.

44.4.6.3 Exit from low-power/stop modes

The slave receive input detect circuit and address matching feature are still active on low power modes (wait and stop). An asynchronous input matching slave address or general call address brings the CPU out of low power/stop mode if the interrupt is not masked. Therefore, TCF and IAAS both can trigger this interrupt.

44.4.6.4 Arbitration lost interrupt

The I2C is a true multimaster bus that allows more than one master to be connected on it. If two or more masters try to control the bus at the same time, the relative priority of the contending masters is determined by a data arbitration procedure. The I2C module asserts the arbitration-lost interrupt when it loses the data arbitration process and the ARBL bit in the Status Register is set.

Arbitration is lost in the following circumstances:

1. SDA is sampled as low when the master drives high during an address or data transmit cycle.
2. SDA is sampled as low when the master drives high during the acknowledge bit of a data receive cycle.
3. A START cycle is attempted when the bus is busy.
4. A repeated START cycle is requested in slave mode.
5. A STOP condition is detected when the master did not request it.

The ARBL bit must be cleared (by software) by writing 1 to it.

44.4.6.5 Timeout interrupt in SMBus

When the IICIE bit is set, the I2C module asserts a timeout interrupt (outputs SLTF and SHTF2) upon detection of any of the mentioned timeout conditions, with one exception. The SCL high and SDA high TIMEOUT mechanism must not be used to influence the timeout interrupt output, because this timeout indicates an idle condition on the bus. SHTF1 rises when it matches the SCL high and SDA high TIMEOUT and falls automatically just to indicate the bus status. The SHTF2's timeout period is the same as that of SHTF1, which is short compared to that of SLTF, so another control bit, SHTF2IE, is added to enable or disable it.

44.4.7 Programmable input glitch filter

An I2C glitch filter has been added outside legacy I2C modules but within the I2C package. This filter can absorb glitches on the I2C clock and data lines for the I2C module. The width of the glitch to absorb can be specified in terms of the number of (half) bus clock cycles. A single Programmable Input Glitch Filter control register is provided. Effectively, any down-up-down or up-down-up transition on the data line that occurs within the number of clock cycles programmed in this register is ignored by the I2C module. The programmer must specify the size of the glitch (in terms of bus clock cycles) for the filter to absorb and not pass.

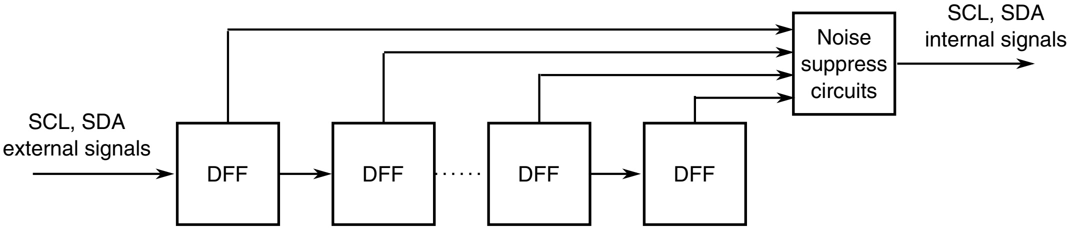


Figure 44-29. Programmable input glitch filter diagram

44.4.8 Address matching wakeup

When a primary, range, or general call address match occurs when the I2C module is in slave receive mode, the MCU wakes from a low power mode with no peripheral bus running. Data sent on the bus that is the same as a target device address might also wake the target MCU.

After the address matching IAAS bit is set, an interrupt is sent at the end of address matching to wake the core. The IAAS bit must be cleared after the clock recovery.

NOTE

After the system recovers and is in Run mode, restart the I2C module if necessary. The SCL line is not held low until the I2C module resets after address matching.

NOTE

The main purpose of this feature is to wake the MCU from Stop mode. The main purpose is not communication.

44.4.9 DMA support

If the DMAEN bit is cleared and the IICIE bit is set, an interrupt condition generates an interrupt request. If the DMAEN bit is set and the IICIE bit is set, an interrupt condition generates a DMA request instead. DMA requests are generated by the transfer complete flag (TCF).

If the DMAEN bit is set, the only arbitration lost is to another I2C module (error), and SCL low timeouts (error) generate CPU interrupts. All other events initiate a DMA transfer.

NOTE

Before the last byte of master receive mode, TXAK must be set to send a NACK after the last byte's transfer. Therefore, the DMA must be disabled before the last byte's transfer.

NOTE

In 10-bit address mode transmission, the addresses to send occupy 2-3 bytes. During this transfer period, the DMA must be disabled because the C1 register is written to send a repeat start or to change the transfer direction.

44.5 Initialization/application information

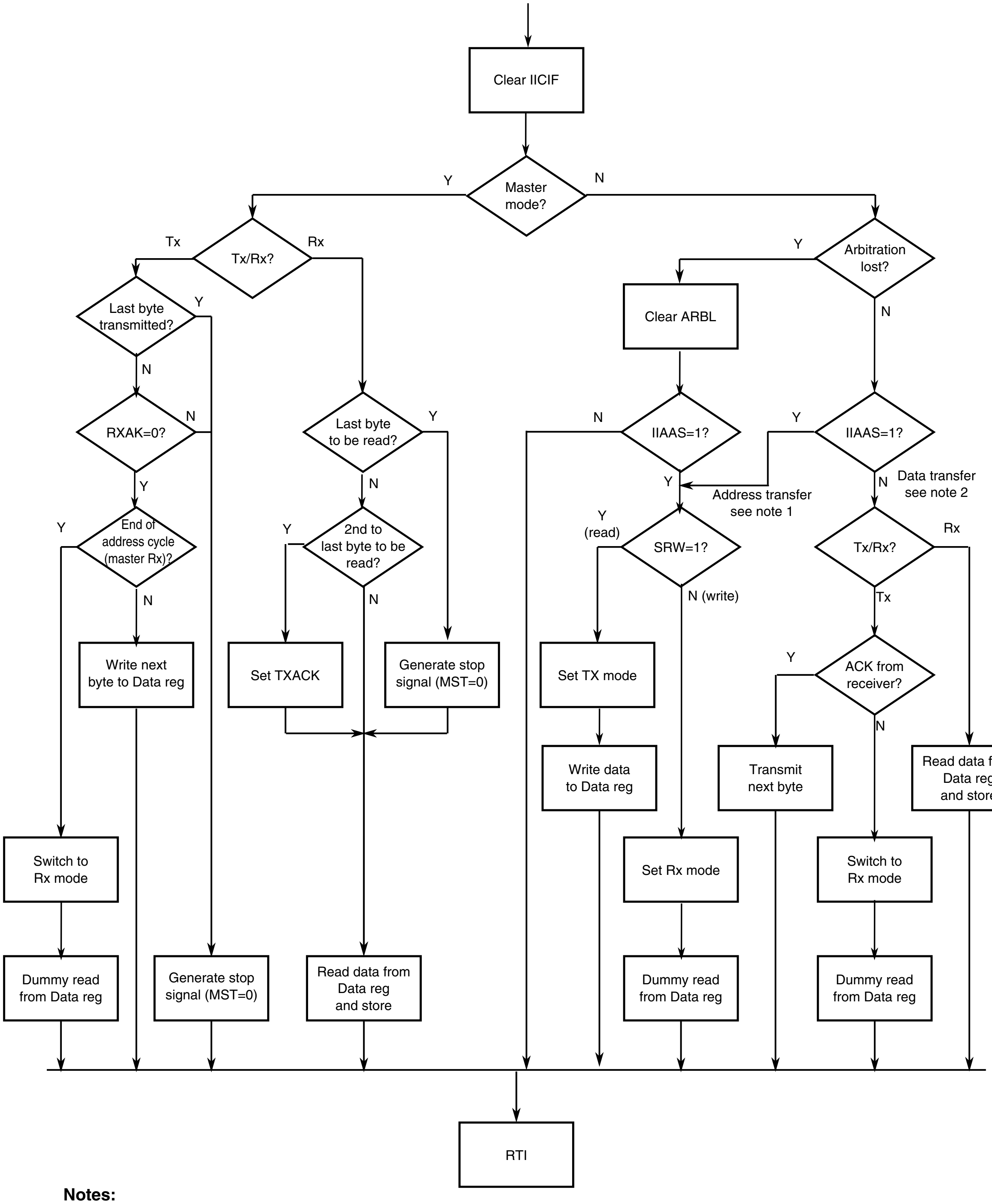
Module Initialization (Slave)

1. Write: Control Register 2
    - to enable or disable general call
    - to select 10-bit or 7-bit addressing mode
2. Write: Address Register 1 to set the slave address
3. Write: Control Register 1 to enable the I2C module and interrupts
4. Initialize RAM variables (IICEN = 1 and IICIE = 1) for transmit data
5. Initialize RAM variables used to achieve the routine shown in the following figure

Module Initialization (Master)

1. Write: Frequency Divider register to set the I2C baud rate (example provided in this chapter)
2. Write: Control Register 1 to enable the I2C module and interrupts
3. Initialize RAM variables (IICEN = 1 and IICIE = 1) for transmit data
4. Initialize RAM variables used to achieve the routine shown in the following figure
5. Write: Control Register 1 to enable TX
6. Write: Control Register 1 to enable MST (master mode)
7. Write: Data register with the address of the target slave (the LSB of this byte determines whether the communication is master receive or transmit)

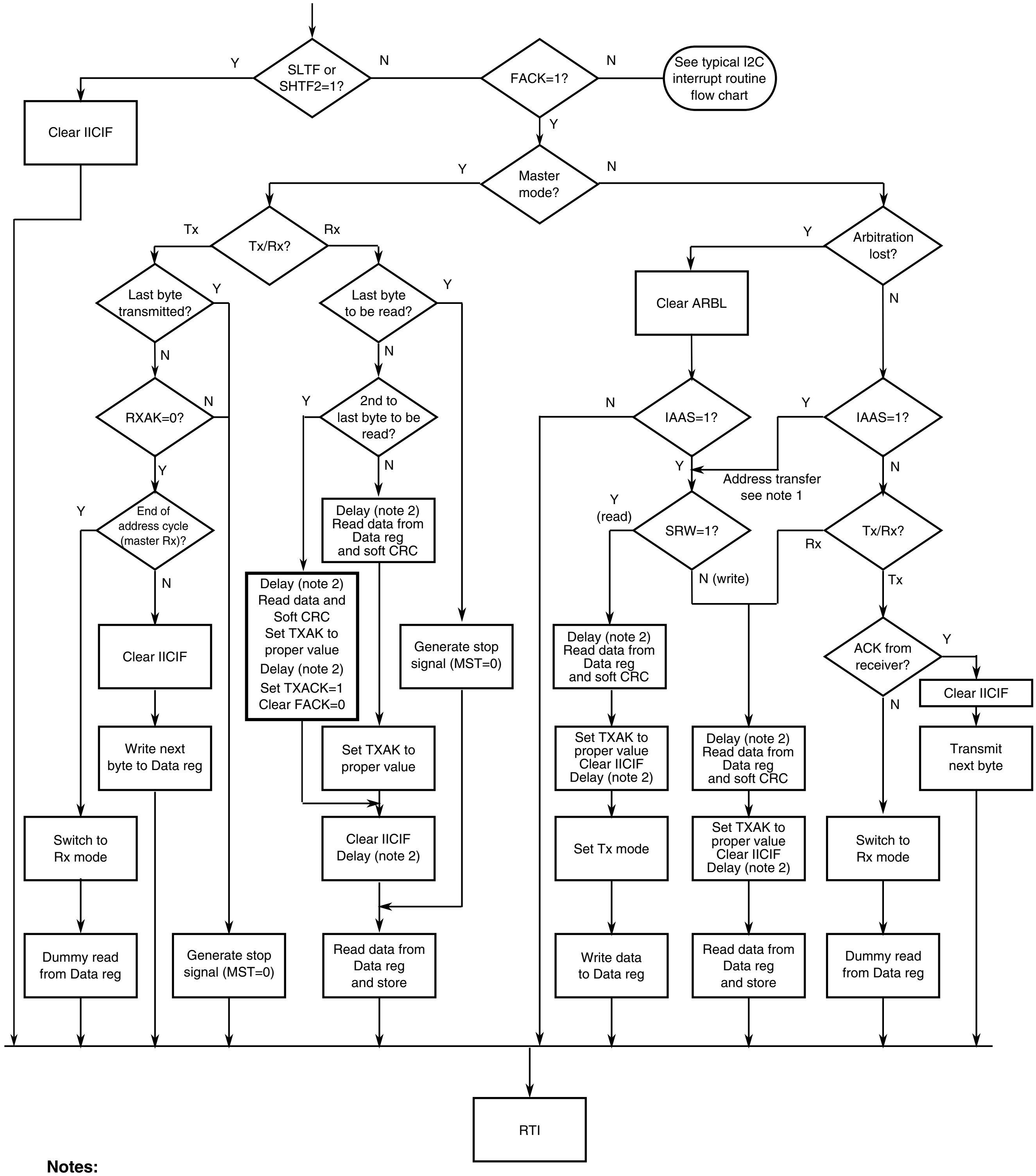
The routine shown in the following figure can handle both master and slave I2C operations. For slave operation, an incoming I2C message that contains the proper address begins I2C communication. For master operation, communication must be initiated by writing the Data register.


Clear IICIF
Master mode?
Y
N
Tx/Rx?
Tx
Rx
Arbitration lost?
Y
N
Clear ARBL
Last byte transmitted?
Y
N
RXAK=0?
N
Y
Last byte to be read?
Y
N
IIAAS=1?
N
Y
IIAAS=1?
Y
N
Data transfer see note 2
Address transfer see note 1
End of address cycle (master Rx)?
Y
N
2nd to last byte to be read?
Y
N
Y (read)
SRW=1?
N (write)
Tx/Rx?
Rx
Tx
ACK from receiver?
Y
N
Write next byte to Data reg
Set TXACK
Generate stop signal (MST=0)
Set TX mode
Write data to Data reg
Transmit next byte
Read data from Data reg and store
Switch to Rx mode
Set Rx mode
Switch to Rx mode
Dummy read from Data reg
Generate stop signal (MST=0)
Read data from Data reg and store
Dummy read from Data reg
Dummy read from Data reg
RTI

Notes:

1. If general call is enabled, check to determine if the received address is a general call address (0x00). If the received address is a general call address, the general call must be handled by user software.
2. When 10-bit addressing addresses a slave, the slave sees an interrupt following the first byte of the extended address. Ensure that for this interrupt, the contents of the Data register are ignored and not treated as a valid data transfer.

Figure 44-30. Typical I2C interrupt routine


SLTF or SHTF2=1?
Y
N
FACK=1?
N
See typical I2C interrupt routine flow chart
Y
Clear IICIF
Master mode?
Y
N
Tx/Rx?
Tx
Rx
Arbitration lost?
Y
N
Clear ARBL
Last byte transmitted?
Y
N
Last byte to be read?
Y
N
RXAK=0?
N
Y
2nd to last byte to be read?
Y
N
IAAS=1?
N
Y
IAAS=1?
Y
N
Address transfer see note 1
End of address cycle (master Rx)?
Y
N
Delay (note 2) Read data from Data reg and soft CRC
SRW=1?
Y (read)
N (write)
Tx/Rx?
Rx
Tx
Delay (note 2) Read data and Soft CRC Set TXAK to proper value Delay (note 2) Set TXACK=1 Clear FACK=0
Generate stop signal (MST=0)
Clear IICIF
Delay (note 2) Read data from Data reg and soft CRC
ACK from receiver?
Y
N
Clear IICIF
Write next byte to Data reg
Set TXAK to proper value
Set TXAK to proper value Clear IICIF Delay (note 2)
Delay (note 2) Read data from Data reg and soft CRC
Transmit next byte
Switch to Rx mode
Clear IICIF Delay (note 2)
Set Tx mode
Set TXAK to proper value Clear IICIF Delay (note 2)
Switch to Rx mode
Dummy read from Data reg
Generate stop signal (MST=0)
Read data from Data reg and store
Write data to Data reg
Read data from Data reg and store
Dummy read from Data reg
RTI

Notes:

1. If general call or SIICAEN is enabled, check to determine if the received address is a general call address (0x00) or an SMBus device default address. In either case, they must be handled by user software.
2. In receive mode, one bit time delay may be needed before the first and second data reading.

Figure 44-31. Typical I2C SMBus interrupt routine

Chapter 45
Universal Asynchronous Receiver/Transmitter (UART)

45.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The UART allows asynchronous serial communication with peripheral devices and CPUs.

45.1.1 Features

The UART includes the following features:

- Full-duplex operation
- Standard mark/space non-return-to-zero (NRZ) format
- Selectable IrDA 1.4 return-to-zero-inverted (RZI) format with programmable pulse width
- 13-bit baud rate selection with /32 fractional divide, based on the module clock frequency
- Programmable 8-bit or 9-bit data format
- Separately enabled transmitter and receiver
- Programmable transmitter output polarity
- Programmable receive input polarity
- 13-bit break character option

- 11-bit break character detection option
- Independent FIFO structure for transmit and receive
- Two receiver wakeup methods:
  - Idle line wakeup
  - Address mark wakeup
- Address match feature in the receiver to reduce address mark wakeup ISR overhead
- Ability to select MSB or LSB to be first bit on wire
- Hardware flow control support for request to send (RTS) and clear to send (CTS) signals
- Support for ISO 7816 protocol to interface with SIM cards and smart cards
  - Support for T=0 and T=1 protocols
  - Automatic retransmission of NACK'd packets with programmable retry threshold
  - Support for 11 and 12 ETU transfers
  - Detection of initial packet and automated transfer parameter programming
  - Interrupt-driven operation with seven ISO-7816 specific interrupts:
    - Wait time violated
    - Character wait time violated
    - Block wait time violated
    - Initial frame detected
    - Transmit error threshold exceeded
    - Receive error threshold exceeded
    - Guard time violated
- Support for CEA709.1-B protocol used in building automation and home networking systems
  - Automatic clock resynchronization
  - Support for collision detection
- Interrupt-driven operation with 12 flags, not specific to ISO-7816 support

  - Transmitter data buffer at or below watermark
  - Transmission complete
  - Receiver data buffer at or above watermark
  - Idle receiver input
  - Receiver data buffer overrun
  - Receiver data buffer underflow
  - Transmit data buffer overflow
  - Noise error
  - Framing error
  - Parity error
  - Active edge on receive pin
  - LIN break detect
- Receiver framing error detection
- Hardware parity generation and checking
- 1/16 bit-time noise detection
- DMA interface

45.1.2 Modes of operation

The UART functions in the same way in all the normal modes.

It has the following two low power modes:

- Wait mode
- Stop mode

45.1.2.1 Run mode

This is the normal mode of operation.

45.1.2.2 Wait mode

UART operation in the Wait mode depends on the state of the C1[UARTSWAI] field.

- If C1[UARTSWAI] is cleared, and the CPU is in Wait mode, the UART operates normally.
- If C1[UARTSWAI] is set, and the CPU is in Wait mode, the UART clock generation ceases and the UART module enters a power conservation state.

C1[UARTSWAI] does not initiate any power down or power up procedures for the ISO-7816 smartcard interface.

Setting C1[UARTSWAI] does not affect the state of the C2[RE] or C2[TE].

If C1[UARTSWAI] is set, any ongoing transmission or reception stops at the Wait mode entry. The transmission or reception resumes when either an internal or external interrupt brings the CPU out of Wait mode. Bringing the CPU out of Wait mode by reset aborts any ongoing transmission or reception and resets the UART.

45.1.2.3 Stop mode

The UART is inactive during Stop mode for reduced power consumption. The STOP instruction does not affect the UART register states, but the UART module clock is disabled. The UART operation resumes after an external interrupt brings the CPU out of Stop mode. Bringing the CPU out of Stop mode by reset aborts any ongoing transmission or reception and resets the UART. Entering or leaving Stop mode does not initiate any power down or power up procedures for the ISO-7816 smartcard interface.

45.2 UART signal descriptions

The UART signals are shown in the following table.

Table 45-1. UART signal descriptions

Signal	Description	I/O
$\overline{\text{CTS}}$	Clear to send	I
$\overline{\text{RTS}}$	Request to send	O
RXD	Receive data	I
TXD	Transmit data	O
$\overline{\text{Collision}}$	Collision detect	I

45.2.1 Detailed signal descriptions

The detailed signal descriptions of the UART are shown in the following table.

Table 45-2. UART—Detailed signal descriptions

Signal	I/O	Description	
$\overline{\text{CTS}}$	I	Clear to send. Indicates whether the UART can start transmitting data when flow control is enabled.	
		State meaning	Asserted—Data transmission can start. Negated—Data transmission cannot start.
		Timing	Assertion—When transmitting device's $\overline{\text{RTS}}$ asserts. Negation—When transmitting device's $\overline{\text{RTS}}$ deasserts.
$\overline{\text{RTS}}$	O	Request to send. When driven by the receiver, indicates whether the UART is ready to receive data. When driven by the transmitter, can enable an external transceiver during transmission.	
		State Meaning	Asserted—When driven by the receiver, ready to receive data. When driven by the transmitter, enable the external transmitter. Negated—When driven by the receiver, not ready to receive data. When driven by the transmitter, disable the external transmitter.
		Timing	Assertion—Can occur at any time; can assert asynchronously to the other input signals. Negation—Can occur at any time; can deassert asynchronously to the other input signals.
RXD	I	Receive data. Serial data input to receiver.	
		State meaning	Whether RXD is interpreted as a 1 or 0 depends on the bit encoding method along with other configuration settings.
		Timing	Sampled at a frequency determined by the module clock divided by the baud rate.
TXD	O	Transmit data. Serial data output from transmitter.	
		State meaning	Whether TXD is interpreted as a 1 or 0 depends on the bit encoding method along with other configuration settings.
		Timing	Driven at the beginning or within a bit time according to the bit encoding method along with other configuration settings. Otherwise, transmissions are independent of reception timing.
$\overline{\text{Collision}}$	I	Collision Detect. Indicates if a collision is detected during Data Transmission.	
		State Meaning	Asserted—Indicates a collision detection. UARTxCPW determines the length of this pulse for valid collision detection. Negated—No collision detected.
		Timing	Asserts asynchronously to other input signals.

45.3 Memory map and registers

This section provides a detailed description of all memory and registers.

Accessing reserved addresses within the memory map results in a transfer error. None of the contents of the implemented addresses are modified as a result of that access.

Only byte accesses are supported.

UART memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_A000	UART Baud Rate Registers: High (UART0_BDH)	8	R/W	00h	45.3.1/ 1053
4006_A001	UART Baud Rate Registers: Low (UART0_BDL)	8	R/W	04h	45.3.2/ 1054
4006_A002	UART Control Register 1 (UART0_C1)	8	R/W	00h	45.3.3/ 1055
4006_A003	UART Control Register 2 (UART0_C2)	8	R/W	00h	45.3.4/ 1057
4006_A004	UART Status Register 1 (UART0_S1)	8	R	C0h	45.3.5/ 1058
4006_A005	UART Status Register 2 (UART0_S2)	8	R/W	00h	45.3.6/ 1061
4006_A006	UART Control Register 3 (UART0_C3)	8	R/W	00h	45.3.7/ 1063
4006_A007	UART Data Register (UART0_D)	8	R/W	00h	45.3.8/ 1065
4006_A008	UART Match Address Registers 1 (UART0_MA1)	8	R/W	00h	45.3.9/ 1066
4006_A009	UART Match Address Registers 2 (UART0_MA2)	8	R/W	00h	45.3.10/ 1067
4006_A00A	UART Control Register 4 (UART0_C4)	8	R/W	00h	45.3.11/ 1067
4006_A00B	UART Control Register 5 (UART0_C5)	8	R/W	00h	45.3.12/ 1068
4006_A00C	UART Extended Data Register (UART0_ED)	8	R	00h	45.3.13/ 1069
4006_A00D	UART Modem Register (UART0_MODEM)	8	R/W	00h	45.3.14/ 1070

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_A00E	UART Infrared Register (UART0_IR)	8	R/W	00h	45.3.15/ 1071
4006_A010	UART FIFO Parameters (UART0_PFIFO)	8	R/W	See section	45.3.16/ 1072
4006_A011	UART FIFO Control Register (UART0_CFIFO)	8	R/W	00h	45.3.17/ 1074
4006_A012	UART FIFO Status Register (UART0_SFIFO)	8	R/W	C0h	45.3.18/ 1075
4006_A013	UART FIFO Transmit Watermark (UART0_TWFIFO)	8	R/W	00h	45.3.19/ 1076
4006_A014	UART FIFO Transmit Count (UART0_TCFIFO)	8	R	00h	45.3.20/ 1077
4006_A015	UART FIFO Receive Watermark (UART0_RWFIFO)	8	R/W	01h	45.3.21/ 1077
4006_A016	UART FIFO Receive Count (UART0_RCFIFO)	8	R	00h	45.3.22/ 1078
4006_A018	UART 7816 Control Register (UART0_C7816)	8	R/W	00h	45.3.23/ 1078
4006_A019	UART 7816 Interrupt Enable Register (UART0_IE7816)	8	R/W	00h	45.3.24/ 1080
4006_A01A	UART 7816 Interrupt Status Register (UART0_IS7816)	8	R/W	00h	45.3.25/ 1081
4006_A01B	UART 7816 Wait Parameter Register (UART0_WP7816T0)	8	R/W	0Ah	45.3.26/ 1083
4006_A01B	UART 7816 Wait Parameter Register (UART0_WP7816T1)	8	R/W	0Ah	45.3.27/ 1083
4006_A01C	UART 7816 Wait N Register (UART0_WN7816)	8	R/W	00h	45.3.28/ 1084
4006_A01D	UART 7816 Wait FD Register (UART0_WF7816)	8	R/W	01h	45.3.29/ 1084
4006_A01E	UART 7816 Error Threshold Register (UART0_ET7816)	8	R/W	00h	45.3.30/ 1085
4006_A01F	UART 7816 Transmit Length Register (UART0_TL7816)	8	R/W	00h	45.3.31/ 1086
4006_A021	UART CEA709.1-B Control Register 6 (UART0_C6)	8	R/W	00h	45.3.32/ 1086
4006_A022	UART CEA709.1-B Packet Cycle Time Counter High (UART0_PCTH)	8	R/W	00h	45.3.33/ 1087
4006_A023	UART CEA709.1-B Packet Cycle Time Counter Low (UART0_PCTL)	8	R/W	00h	45.3.34/ 1088

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_A024	UART CEA709.1-B Beta1 Timer (UART0_B1T)	8	R/W	00h	45.3.35/ 1088
4006_A025	UART CEA709.1-B Secondary Delay Timer High (UART0_SDTH)	8	R/W	00h	45.3.36/ 1089
4006_A026	UART CEA709.1-B Secondary Delay Timer Low (UART0_SDTL)	8	R/W	00h	45.3.37/ 1089
4006_A027	UART CEA709.1-B Preamble (UART0_PRE)	8	R/W	00h	45.3.38/ 1090
4006_A028	UART CEA709.1-B Transmit Packet Length (UART0_TPL)	8	R/W	00h	45.3.39/ 1090
4006_A029	UART CEA709.1-B Interrupt Enable Register (UART0_IE)	8	R/W	00h	45.3.40/ 1091
4006_A02A	UART CEA709.1-B WBASE (UART0_WB)	8	R/W	00h	45.3.41/ 1092
4006_A02B	UART CEA709.1-B Status Register (UART0_S3)	8	R/W	00h	45.3.42/ 1092
4006_A02C	UART CEA709.1-B Status Register (UART0_S4)	8	R/W	00h	45.3.43/ 1094
4006_A02D	UART CEA709.1-B Received Packet Length (UART0_RPL)	8	R	00h	45.3.44/ 1095
4006_A02E	UART CEA709.1-B Received Preamble Length (UART0_RPREL)	8	R	00h	45.3.45/ 1095
4006_A02F	UART CEA709.1-B Collision Pulse Width (UART0_CPW)	8	R/W	00h	45.3.46/ 1096
4006_A030	UART CEA709.1-B Receive Indeterminate Time (UART0_RIDT)	8	R/W	00h	45.3.47/ 1096
4006_A031	UART CEA709.1-B Transmit Indeterminate Time (UART0_TIDT)	8	R/W	00h	45.3.48/ 1097
4006_B000	UART Baud Rate Registers: High (UART1_BDH)	8	R/W	00h	45.3.1/ 1053
4006_B001	UART Baud Rate Registers: Low (UART1_BDL)	8	R/W	04h	45.3.2/ 1054
4006_B002	UART Control Register 1 (UART1_C1)	8	R/W	00h	45.3.3/ 1055
4006_B003	UART Control Register 2 (UART1_C2)	8	R/W	00h	45.3.4/ 1057
4006_B004	UART Status Register 1 (UART1_S1)	8	R	C0h	45.3.5/ 1058
4006_B005	UART Status Register 2 (UART1_S2)	8	R/W	00h	45.3.6/ 1061

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_B006	UART Control Register 3 (UART1_C3)	8	R/W	00h	45.3.7/ 1063
4006_B007	UART Data Register (UART1_D)	8	R/W	00h	45.3.8/ 1065
4006_B008	UART Match Address Registers 1 (UART1_MA1)	8	R/W	00h	45.3.9/ 1066
4006_B009	UART Match Address Registers 2 (UART1_MA2)	8	R/W	00h	45.3.10/ 1067
4006_B00A	UART Control Register 4 (UART1_C4)	8	R/W	00h	45.3.11/ 1067
4006_B00B	UART Control Register 5 (UART1_C5)	8	R/W	00h	45.3.12/ 1068
4006_B00C	UART Extended Data Register (UART1_ED)	8	R	00h	45.3.13/ 1069
4006_B00D	UART Modem Register (UART1_MODEM)	8	R/W	00h	45.3.14/ 1070
4006_B00E	UART Infrared Register (UART1_IR)	8	R/W	00h	45.3.15/ 1071
4006_B010	UART FIFO Parameters (UART1_PFIFO)	8	R/W	See section	45.3.16/ 1072
4006_B011	UART FIFO Control Register (UART1_CFIFO)	8	R/W	00h	45.3.17/ 1074
4006_B012	UART FIFO Status Register (UART1_SFIFO)	8	R/W	C0h	45.3.18/ 1075
4006_B013	UART FIFO Transmit Watermark (UART1_TWFIFO)	8	R/W	00h	45.3.19/ 1076
4006_B014	UART FIFO Transmit Count (UART1_TCFIFO)	8	R	00h	45.3.20/ 1077
4006_B015	UART FIFO Receive Watermark (UART1_RWFIFO)	8	R/W	01h	45.3.21/ 1077
4006_B016	UART FIFO Receive Count (UART1_RCFIFO)	8	R	00h	45.3.22/ 1078
4006_B018	UART 7816 Control Register (UART1_C7816)	8	R/W	00h	45.3.23/ 1078
4006_B019	UART 7816 Interrupt Enable Register (UART1_IE7816)	8	R/W	00h	45.3.24/ 1080
4006_B01A	UART 7816 Interrupt Status Register (UART1_IS7816)	8	R/W	00h	45.3.25/ 1081
4006_B01B	UART 7816 Wait Parameter Register (UART1_WP7816T0)	8	R/W	0Ah	45.3.26/ 1083

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_B01B	UART 7816 Wait Parameter Register (UART1_WP7816T1)	8	R/W	0Ah	45.3.27/ 1083
4006_B01C	UART 7816 Wait N Register (UART1_WN7816)	8	R/W	00h	45.3.28/ 1084
4006_B01D	UART 7816 Wait FD Register (UART1_WF7816)	8	R/W	01h	45.3.29/ 1084
4006_B01E	UART 7816 Error Threshold Register (UART1_ET7816)	8	R/W	00h	45.3.30/ 1085
4006_B01F	UART 7816 Transmit Length Register (UART1_TL7816)	8	R/W	00h	45.3.31/ 1086
4006_B021	UART CEA709.1-B Control Register 6 (UART1_C6)	8	R/W	00h	45.3.32/ 1086
4006_B022	UART CEA709.1-B Packet Cycle Time Counter High (UART1_PCTH)	8	R/W	00h	45.3.33/ 1087
4006_B023	UART CEA709.1-B Packet Cycle Time Counter Low (UART1_PCTL)	8	R/W	00h	45.3.34/ 1088
4006_B024	UART CEA709.1-B Beta1 Timer (UART1_B1T)	8	R/W	00h	45.3.35/ 1088
4006_B025	UART CEA709.1-B Secondary Delay Timer High (UART1_SDTH)	8	R/W	00h	45.3.36/ 1089
4006_B026	UART CEA709.1-B Secondary Delay Timer Low (UART1_SDTL)	8	R/W	00h	45.3.37/ 1089
4006_B027	UART CEA709.1-B Preamble (UART1_PRE)	8	R/W	00h	45.3.38/ 1090
4006_B028	UART CEA709.1-B Transmit Packet Length (UART1_TPL)	8	R/W	00h	45.3.39/ 1090
4006_B029	UART CEA709.1-B Interrupt Enable Register (UART1_IE)	8	R/W	00h	45.3.40/ 1091
4006_B02A	UART CEA709.1-B WBASE (UART1_WB)	8	R/W	00h	45.3.41/ 1092
4006_B02B	UART CEA709.1-B Status Register (UART1_S3)	8	R/W	00h	45.3.42/ 1092
4006_B02C	UART CEA709.1-B Status Register (UART1_S4)	8	R/W	00h	45.3.43/ 1094
4006_B02D	UART CEA709.1-B Received Packet Length (UART1_RPL)	8	R	00h	45.3.44/ 1095
4006_B02E	UART CEA709.1-B Received Preamble Length (UART1_RPREL)	8	R	00h	45.3.45/ 1095
4006_B02F	UART CEA709.1-B Collision Pulse Width (UART1_CPW)	8	R/W	00h	45.3.46/ 1096

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_B030	UART CEA709.1-B Receive Indeterminate Time (UART1_RIDT)	8	R/W	00h	45.3.47/ 1096
4006_B031	UART CEA709.1-B Transmit Indeterminate Time (UART1_TIDT)	8	R/W	00h	45.3.48/ 1097
4006_C000	UART Baud Rate Registers: High (UART2_BDH)	8	R/W	00h	45.3.1/ 1053
4006_C001	UART Baud Rate Registers: Low (UART2_BDL)	8	R/W	04h	45.3.2/ 1054
4006_C002	UART Control Register 1 (UART2_C1)	8	R/W	00h	45.3.3/ 1055
4006_C003	UART Control Register 2 (UART2_C2)	8	R/W	00h	45.3.4/ 1057
4006_C004	UART Status Register 1 (UART2_S1)	8	R	C0h	45.3.5/ 1058
4006_C005	UART Status Register 2 (UART2_S2)	8	R/W	00h	45.3.6/ 1061
4006_C006	UART Control Register 3 (UART2_C3)	8	R/W	00h	45.3.7/ 1063
4006_C007	UART Data Register (UART2_D)	8	R/W	00h	45.3.8/ 1065
4006_C008	UART Match Address Registers 1 (UART2_MA1)	8	R/W	00h	45.3.9/ 1066
4006_C009	UART Match Address Registers 2 (UART2_MA2)	8	R/W	00h	45.3.10/ 1067
4006_C00A	UART Control Register 4 (UART2_C4)	8	R/W	00h	45.3.11/ 1067
4006_C00B	UART Control Register 5 (UART2_C5)	8	R/W	00h	45.3.12/ 1068
4006_C00C	UART Extended Data Register (UART2_ED)	8	R	00h	45.3.13/ 1069
4006_C00D	UART Modem Register (UART2_MODEM)	8	R/W	00h	45.3.14/ 1070
4006_C00E	UART Infrared Register (UART2_IR)	8	R/W	00h	45.3.15/ 1071
4006_C010	UART FIFO Parameters (UART2_PFIFO)	8	R/W	See section	45.3.16/ 1072
4006_C011	UART FIFO Control Register (UART2_CFIFO)	8	R/W	00h	45.3.17/ 1074
4006_C012	UART FIFO Status Register (UART2_SFIFO)	8	R/W	C0h	45.3.18/ 1075

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_C013	UART FIFO Transmit Watermark (UART2_TWFIFO)	8	R/W	00h	45.3.19/ 1076
4006_C014	UART FIFO Transmit Count (UART2_TCFIFO)	8	R	00h	45.3.20/ 1077
4006_C015	UART FIFO Receive Watermark (UART2_RWFIFO)	8	R/W	01h	45.3.21/ 1077
4006_C016	UART FIFO Receive Count (UART2_RCFIFO)	8	R	00h	45.3.22/ 1078
4006_C018	UART 7816 Control Register (UART2_C7816)	8	R/W	00h	45.3.23/ 1078
4006_C019	UART 7816 Interrupt Enable Register (UART2_IE7816)	8	R/W	00h	45.3.24/ 1080
4006_C01A	UART 7816 Interrupt Status Register (UART2_IS7816)	8	R/W	00h	45.3.25/ 1081
4006_C01B	UART 7816 Wait Parameter Register (UART2_WP7816T0)	8	R/W	0Ah	45.3.26/ 1083
4006_C01B	UART 7816 Wait Parameter Register (UART2_WP7816T1)	8	R/W	0Ah	45.3.27/ 1083
4006_C01C	UART 7816 Wait N Register (UART2_WN7816)	8	R/W	00h	45.3.28/ 1084
4006_C01D	UART 7816 Wait FD Register (UART2_WF7816)	8	R/W	01h	45.3.29/ 1084
4006_C01E	UART 7816 Error Threshold Register (UART2_ET7816)	8	R/W	00h	45.3.30/ 1085
4006_C01F	UART 7816 Transmit Length Register (UART2_TL7816)	8	R/W	00h	45.3.31/ 1086
4006_C021	UART CEA709.1-B Control Register 6 (UART2_C6)	8	R/W	00h	45.3.32/ 1086
4006_C022	UART CEA709.1-B Packet Cycle Time Counter High (UART2_PCTH)	8	R/W	00h	45.3.33/ 1087
4006_C023	UART CEA709.1-B Packet Cycle Time Counter Low (UART2_PCTL)	8	R/W	00h	45.3.34/ 1088
4006_C024	UART CEA709.1-B Beta1 Timer (UART2_B1T)	8	R/W	00h	45.3.35/ 1088
4006_C025	UART CEA709.1-B Secondary Delay Timer High (UART2_SDTH)	8	R/W	00h	45.3.36/ 1089
4006_C026	UART CEA709.1-B Secondary Delay Timer Low (UART2_SDTL)	8	R/W	00h	45.3.37/ 1089
4006_C027	UART CEA709.1-B Preamble (UART2_PRE)	8	R/W	00h	45.3.38/ 1090

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4006_C028	UART CEA709.1-B Transmit Packet Length (UART2_TPL)	8	R/W	00h	45.3.39/ 1090
4006_C029	UART CEA709.1-B Interrupt Enable Register (UART2_IE)	8	R/W	00h	45.3.40/ 1091
4006_C02A	UART CEA709.1-B WBASE (UART2_WB)	8	R/W	00h	45.3.41/ 1092
4006_C02B	UART CEA709.1-B Status Register (UART2_S3)	8	R/W	00h	45.3.42/ 1092
4006_C02C	UART CEA709.1-B Status Register (UART2_S4)	8	R/W	00h	45.3.43/ 1094
4006_C02D	UART CEA709.1-B Received Packet Length (UART2_RPL)	8	R	00h	45.3.44/ 1095
4006_C02E	UART CEA709.1-B Received Preamble Length (UART2_RPREL)	8	R	00h	45.3.45/ 1095
4006_C02F	UART CEA709.1-B Collision Pulse Width (UART2_CPW)	8	R/W	00h	45.3.46/ 1096
4006_C030	UART CEA709.1-B Receive Indeterminate Time (UART2_RIDT)	8	R/W	00h	45.3.47/ 1096
4006_C031	UART CEA709.1-B Transmit Indeterminate Time (UART2_TIDT)	8	R/W	00h	45.3.48/ 1097

45.3.1 UART Baud Rate Registers: High (UARTx_BDH)

This register, along with the BDL register, controls the prescale divisor for UART baud rate generation. To update the 13-bit baud rate setting (SBR[12:0]), first write to BDH to buffer the high half of the new value and then write to BDL. The working value in BDH does not change until BDL is written.

BDL is reset to a nonzero value, but after reset, the baud rate generator remains disabled until the first time the receiver or transmitter is enabled, that is, when C2[RE] or C2[TE] is set.

Addresses: UART0_BDH is 4006_A000h base + 0h offset = 4006_A000h
UART1_BDH is 4006_B000h base + 0h offset = 4006_B000h
UART2_BDH is 4006_C000h base + 0h offset = 4006_C000h

Bit	7	6	5	4	3	2	1	0
Read	LBKDIE	RXEDGIE	0	SBR				
Write			▨					
Reset	0	0	0	0	0	0	0	0

UARTx_BDH field descriptions

Field	Description
7 LBKDIE	LIN Break Detect Interrupt Enable Enables the LIN break detect flag, LBKDIF, to generate interrupt requests based on the state of LBKDDMAS. 0 LBKDIF interrupt requests disabled. 1 LBKDIF interrupt requests enabled.
6 RXEDGIE	RxD Input Active Edge Interrupt Enable Enables the receive input active edge, RXEDGIF, to generate interrupt requests. 0 Hardware interrupts from RXEDGIF disabled using polling. 1 RXEDGIF interrupt request enabled.
5 Reserved	This read-only field is reserved and always has the value zero.
4–0 SBR	UART Baud Rate Bits The baud rate for the UART is determined by the 13 SBR fields. See Baud rate generation for details. **NOTE:** • The baud rate generator is disabled until C2[TE] or C2[RE] is set for the first time after reset.The baud rate generator is disabled when SBR = 0. • Writing to BDH has no effect without writing to BDL, because writing to BDH puts the data in a temporary location until BDL is written.

45.3.2 UART Baud Rate Registers: Low (UARTx_BDL)

This register, along with the BDH register, controls the prescale divisor for UART baud rate generation. To update the 13-bit baud rate setting, SBR[12:0], first write to BDH to buffer the high half of the new value and then write to BDL. The working value in BDH does not change until BDL is written. BDL is reset to a nonzero value, but after reset, the baud rate generator remains disabled until the first time the receiver or transmitter is enabled, that is, when C2[RE] or C2[TE] is set.

Addresses: UART0_BDL is 4006_A000h base + 1h offset = 4006_A001h
UART1_BDL is 4006_B000h base + 1h offset = 4006_B001h
UART2_BDL is 4006_C000h base + 1h offset = 4006_C001h

Bit	7	6	5	4	3	2	1	0
Read Write	SBR							
Reset	0	0	0	0	0	1	0	0

UARTx_BDL field descriptions

Field	Description
7–0 SBR	UART Baud Rate Bits

UARTx_BDL field descriptions (continued)

Field	Description
	The baud rate for the UART is determined by the 13 SBR fields. See Baud rate generation for details. **NOTE:** • The baud rate generator is disabled until C2[TE] or C2[RE] is set for the first time after reset.The baud rate generator is disabled when SBR = 0. • Writing to BDH has no effect without writing to BDL, because writing to BDH puts the data in a temporary location until BDL is written. • When the 1/32 narrow pulse width is selected for infrared (IrDA), the baud rate fields must be even, the least significant bit is 0. See MODEM register for more details.

45.3.3 UART Control Register 1 (UARTx_C1)

This read/write register controls various optional features of the UART system.

Addresses: UART0_C1 is 4006_A000h base + 2h offset = 4006_A002h
UART1_C1 is 4006_B000h base + 2h offset = 4006_B002h
UART2_C1 is 4006_C000h base + 2h offset = 4006_C002h

Bit	7	6	5	4	3	2	1	0
Read Write	LOOPS	UARTSWAI	RSRC	M	WAKE	ILT	PE	PT
Reset	0	0	0	0	0	0	0	0

UARTx_C1 field descriptions

Field	Description
7 LOOPS	Loop Mode Select When LOOPS is set, the RxD pin is disconnected from the UART and the transmitter output is internally connected to the receiver input. The transmitter and the receiver must be enabled to use the loop function. 0 Normal operation. 1 Loop mode where transmitter output is internally connected to receiver input. The receiver input is determined by RSRC.
6 UARTSWAI	UART Stops in Wait Mode 0 UART clock continues to run in Wait mode. 1 UART clock freezes while CPU is in Wait mode.
5 RSRC	Receiver Source Select This field has no meaning or effect unless the LOOPS field is set. When LOOPS is set, the RSRC field determines the source for the receiver shift register input. 0 Selects internal loop back mode. The receiver input is internally connected to transmitter output. 1 Single wire UART mode where the receiver input is connected to the transmit pin input signal.
4 M	9-bit or 8-bit Mode Select This field must be set when C7816[ISO_7816E] is set/enabled.

Table continues on the next page...

UARTx_C1 field descriptions (continued)

Field	Description
	0 Normal—start + 8 data bits (MSB/LSB first as determined by MSBF) + stop. 1 Use—start + 9 data bits (MSB/LSB first as determined by MSBF) + stop.
3 WAKE	Receiver Wakeup Method Select Determines which condition wakes the UART: • Address mark in the most significant bit position of a received data character, or • An idle condition on the receive pin input signal. 0 Idle line wakeup. 1 Address mark wakeup.
2 ILT	Idle Line Type Select Determines when the receiver starts counting logic 1s as idle character bits. The count begins either after a valid start bit or after the stop bit. If the count begins after the start bit, then a string of logic 1s preceding the stop bit can cause false recognition of an idle character. Beginning the count after the stop bit avoids false idle character recognition, but requires properly synchronized transmissions. **NOTE:** • In case the UART is programmed with ILT = 1, a logic of 1'b0 is automatically shifted after a received stop bit, therefore resetting the idle count. • In case the UART is programmed for IDLE line wakeup (RWU = 1 and WAKE = 0), ILT has no effect on when the receiver starts counting logic 1s as idle character bits. In idle line wakeup, an idle character is recognized at anytime the receiver sees 10, 11, or 12 1s depending on the M, PE, and C4[M10] fields. 0 Idle character bit count starts after start bit. 1 Idle character bit count starts after stop bit.
1 PE	Parity Enable Enables the parity function. When parity is enabled, parity function inserts a parity bit in the bit position immediately preceding the stop bit. This field must be set when C7816[ISO_7816E] is set/enabled. 0 Parity function disabled. 1 Parity function enabled.
0 PT	Parity Type Determines whether the UART generates and checks for even parity or odd parity. With even parity, an even number of 1s clears the parity bit and an odd number of 1s sets the parity bit. With odd parity, an odd number of 1s clears the parity bit and an even number of 1s sets the parity bit. This field must be cleared when C7816[ISO_7816E] is set/enabled. 0 Even parity. 1 Odd parity.

45.3.4 UART Control Register 2 (UARTx_C2)

This register can be read or written at any time.

Addresses: UART0_C2 is 4006_A000h base + 3h offset = 4006_A003h
UART1_C2 is 4006_B000h base + 3h offset = 4006_B003h
UART2_C2 is 4006_C000h base + 3h offset = 4006_C003h

Bit	7	6	5	4	3	2	1	0
Read Write	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
Reset	0	0	0	0	0	0	0	0

UARTx_C2 field descriptions

Field	Description
7 TIE	Transmitter Interrupt or DMA Transfer Enable. Enables S1[TDRE] to generate interrupt requests or DMA transfer requests, based on the state of C5[TDMAS]. **NOTE:** If C2[TIE] and C5[TDMAS] are both set, then TCIE must be cleared, and D[D] must not be written unless servicing a DMA request. 0 TDRE interrupt and DMA transfer requests disabled. 1 TDRE interrupt or DMA transfer requests enabled.
6 TCIE	Transmission Complete Interrupt Enable Enables the transmission complete flag, S1[TC], to generate interrupt requests . 0 TC interrupt requests disabled. 1 TC interrupt requests enabled.
5 RIE	Receiver Full Interrupt or DMA Transfer Enable Enables S1[RDRF] to generate interrupt requests or DMA transfer requests, based on the state of C5[RDMAS]. 0 RDRF interrupt and DMA transfer requests disabled. 1 RDRF interrupt or DMA transfer requests enabled.
4 ILIE	Idle Line Interrupt Enable Enables the idle line flag, S1[IDLE], to generate interrupt requests , based on the state of C5[ILDMAS]. 0 IDLE interrupt requests disabled. 1 IDLE interrupt requests enabled.
3 TE	Transmitter Enable Enables the UART transmitter. TE can be used to queue an idle preamble by clearing and then setting TE. When C7816[ISO_7816E] is set/enabled and C7816[TTYPE] = 1, this field is automatically cleared after the requested block has been transmitted. This condition is detected when TL7816[TLEN] = 0 and four additional characters are transmitted.

Table continues on the next page...

UARTx_C2 field descriptions (continued)

Field	Description
	0 Transmitter off. 1 Transmitter on.
2 RE	Receiver Enable Enables the UART receiver. 0 Receiver off. 1 Receiver on.
1 RWU	Receiver Wakeup Control This field can be set to place the UART receiver in a standby state. RWU automatically clears when an RWU event occurs, that is, an IDLE event when C1[WAKE] is clear or an address match when C1[WAKE] is set. This field must be cleared when C7816[ISO_7816E] is set. **NOTE:** RWU must be set only with C1[WAKE] = 0 (wakeup on idle) if the channel is currently not idle. This can be determined by S2[RAF]. If the flag is set to wake up an IDLE event and the channel is already idle, it is possible that the UART will discard data. This is because the data must be received or a LIN break detected after an IDLE is detected before IDLE is allowed to reasserted. 0 Normal operation. 1 RWU enables the wakeup function and inhibits further receiver interrupt requests. Normally, hardware wakes the receiver by automatically clearing RWU.
0 SBK	Send Break Toggling SBK sends one break character from the following: See for the number of logic 0s for the different configurations. Toggling implies clearing the SBK field before the break character has finished transmitting. As long as SBK is set, the transmitter continues to send complete break characters (10, 11, or 12 bits, or 13 or 14 bits). • 10, 11, or 12 logic 0s if S2[BRK13] is cleared • 13 or 14 logic 0s if S2[BRK13] is set. Transmitting break charactersThis field must be cleared when C7816[ISO_7816E] is set. 0 Normal transmitter operation. 1 Queue break characters to be sent.

45.3.5 UART Status Register 1 (UARTx_S1)

The S1 register provides inputs to the MCU for generation of UART interrupts or DMA requests. This register can also be polled by the MCU to check the status of its fields. To clear a flag, the status register should be read followed by a read or write to D register, depending on the interrupt flag type. Other instructions can be executed between the two steps as long the handling of I/O is not compromised, but the order of operations is important for flag clearing. When a flag is configured to trigger a DMA request, assertion of the associated DMA done signal from the DMA controller clears the flag.

NOTE

- If the condition that results in the assertion of the flag, interrupt, or DMA request is not resolved prior to clearing the flag, the flag, and interrupt/DMA request, reasserts. For example, if the DMA or interrupt service routine fails to write sufficient data to the transmit buffer to raise it above the watermark level, the flag reasserts and generates another interrupt or DMA request.
- Reading an empty data register to clear one of the flags of the S1 register causes the FIFO pointers to become misaligned. A receive FIFO flush reinitializes the pointers.

Addresses: UART0_S1 is 4006_A000h base + 4h offset = 4006_A004h
UART1_S1 is 4006_B000h base + 4h offset = 4006_B004h
UART2_S1 is 4006_C000h base + 4h offset = 4006_C004h

Bit	7	6	5	4	3	2	1	0
Read	TDRE	TC	RDRF	IDLE	OR	NF	FE	PF
Write								
Reset	1	1	0	0	0	0	0	0

UARTx_S1 field descriptions

Field	Description
7 TDRE	Transmit Data Register Empty Flag TDRE will set when the number of datawords in the transmit buffer (D and C3[T8])is equal to or less than the number indicated by TWFIFO[TXWATER]. A character that is in the process of being transmitted is not included in the count. To clear TDRE, read S1 when TDRE is set and then write to the UART data register (D). For more efficient interrupt servicing, all data except the final value to be written to the buffer must be written to D/C3[T8]. Then S1 can be read before writing the final data value, resulting in the clearing of the TRDE flag. This is more efficient because the TDRE reasserts until the watermark has been exceeded. So, attempting to clear the TDRE with every write will be ineffective until sufficient data has been written. 0 The amount of data in the transmit buffer is greater than the value indicated by TWFIFO[TXWATER]. 1 The amount of data in the transmit buffer is less than or equal to the value indicated by TWFIFO[TXWATER] at some point in time since the flag has been cleared.
6 TC	Transmit Complete Flag TC is cleared when there is a transmission in progress or when a preamble or break character is loaded. TC is set when the transmit buffer is empty and no data, preamble, or break character is being transmitted. When TC is set, the transmit data output signal becomes idle (logic 1). TC is cleared by reading S1 with TC set and then doing one of the following: When C7816[ISO_7816E] is set/enabled, this field is set after any NACK signal has been received, but prior to any corresponding guard times expiring.When C6[EN709] is set/enabled, this flag is not set on transmit packet completion. • Writing to D to transmit new data. • Queuing a preamble by clearing and then setting C2[TE]. • Queuing a break character by writing 1 to SBK in C2.

Table continues on the next page...

UARTx_S1 field descriptions (continued)

Field	Description
	0 Transmitter active (sending data, a preamble, or a break). 1 Transmitter idle (transmission activity complete).
5 RDRF	Receive Data Register Full Flag RDRF is set when the number of datawords in the receive buffer is equal to or more than the number indicated by RWFIFO[RXWATER]. A dataword that is in the process of being received is not included in the count. RDRF is prevented from setting while S2[LBKDE] is set. Additionally, when S2[LBKDE] is set, the received datawords are stored in the receive buffer but over-write each other. To clear RDRF, read S1 when RDRF is set and then read D. For more efficient interrupt and DMA operation, read all data except the final value from the buffer, using D/C3[T8]/ED. Then read S1 and the final data value, resulting in the clearing of the RDRF flag. Even if RDRF is set, data will continue to be received until an overrun condition occurs. 0 The number of datawords in the receive buffer is less than the number indicated by RXWATER. 1 The number of datawords in the receive buffer is equal to or greater than the number indicated by RXWATER at some point in time since this flag was last cleared.
4 IDLE	Idle Line Flag After the IDLE flag is cleared, a frame must be received (although not necessarily stored in the data buffer, for example if C2[RWU] is set), or a LIN break character must set the S2[LBKDIF] flag before an idle condition can set the IDLE flag. To clear IDLE, read UART status S1 with IDLE set and then read D. IDLE is set when either of the following appear on the receiver input: • 10 consecutive logic 1s if C1[M] = 0 • 11 consecutive logic 1s if C1[M] = 1 and C4[M10] = 0 • 12 consecutive logic 1s if C1[M] = 1, C4[M10] = 1, and C1[PE] = 1 Idle detection is not supported when7816Eor EN709is set/enabled and hence this flag is ignored. **NOTE:** When RWU is set and WAKE is cleared, an idle line condition sets the IDLE flag if RWUID is set, else the IDLE flag does not become set. 0 Receiver input is either active now or has never become active since the IDLE flag was last cleared. 1 Receiver input has become idle or the flag has not been cleared since it last asserted.
3 OR	Receiver Overrun Flag OR is set when software fails to prevent the receive data register from overflowing with data. The OR bit is set immediately after the stop bit has been completely received for the dataword that overflows the buffer and all the other error flags (FE, NF, and PF) are prevented from setting. The data in the shift register is lost, but the data already in the UART data registers is not affected. If the OR flag is set, no data is stored in the data buffer even if sufficient room exists. Additionally, while the OR flag is set, the RDRF and IDLE flags are blocked from asserting, that is, transition from an inactive to an active state. To clear OR, read S1 when OR is set and then read D. If LBKDE is enabled and a LIN Break is detected, the OR field asserts if S2[LBKDIF] is not cleared before the next data character is received.See for more details regarding the operation of the OR bit. Overrun (OR) flag implicationsIn 7816 mode, it is possible to configure a NACK to be returned by programing C7816[ONACK]. 0 No overrun has occurred since the last time the flag was cleared. 1 Overrun has occurred or the overrun flag has not been cleared since the last overrun occured.
2 NF	Noise Flag NF is set when the UART detects noise on the receiver input. NF does not become set in the case of an overrun or while the LIN break detect feature is enabled (S2[LBKDE] = 1). When NF is set, it indicates only that a dataword has been received with noise since the last time it was cleared. There is no guarantee that the first dataword read from the receive buffer has noise or that there is only one dataword

Table continues on the next page...

UARTx_S1 field descriptions (continued)

Field	Description
	in the buffer that was received with noise unless the receive buffer has a depth of one. To clear NF, read S1 and then read D. When EN709 is set/enabled, noise flag is not set. 0 No noise detected since the last time this flag was cleared. If the receive buffer has a depth greater than 1 then there may be data in the receiver buffer that was received with noise. 1 At least one dataword was received with noise detected since the last time the flag was cleared.
1 FE	Framing Error Flag FE is set when a logic 0 is accepted as the stop bit. FE does not set in the case of an overrun or while the LIN break detect feature is enabled (S2[LBKDE] = 1). FE inhibits further data reception until it is cleared. To clear FE, read S1 with FE set and then read D. The last data in the receive buffer represents the data that was received with the frame error enabled. Framing errors are not supported when 7816E is set/ enabled. However, if this flag is set, data is still not received in 7816 mode.Framing errors are not supported in 709 mode. 0 No framing error detected. 1 Framing error.
0 PF	Parity Error Flag PF is set when PE is set, S2[LBKDE] is disabled, and the parity of the received data does not match its parity bit. The PF is not set in the case of an overrun condition. When PF is set, it indicates only that a dataword was received with parity error since the last time it was cleared. There is no guarantee that the first dataword read from the receive buffer has a parity error or that there is only one dataword in the buffer that was received with a parity error, unless the receive buffer has a depth of one. To clear PF, read S1 and then read D. Within the receive buffer structure the received dataword is tagged if it is received with a parity error. This information is available by reading the ED register prior to reading the D register.When EN709 is set/enabled parity error flag is not set. 0 No parity error detected since the last time this flag was cleared. If the receive buffer has a depth greater than 1, then there may be data in the receive buffer what was received with a parity error. 1 At least one dataword was received with a parity error since the last time this flag was cleared.

45.3.6 UART Status Register 2 (UARTx_S2)

The S2 register provides inputs to the MCU for generation of UART interrupts or DMA requests. Also, this register can be polled by the MCU to check the status of these bits. This register can be read or written at any time, with the exception of the MSBF and RXINV bits, which should be changed by the user only between transmit and receive packets.

Addresses: UART0_S2 is 4006_A000h base + 5h offset = 4006_A005h
UART1_S2 is 4006_B000h base + 5h offset = 4006_B005h
UART2_S2 is 4006_C000h base + 5h offset = 4006_C005h

Bit	7	6	5	4	3	2	1	0
Read	LBKDIF	RXEDGIF	MSBF	RXINV	RWUID	BRK13	LBKDE	RAF
Write								▨
Reset	0	0	0	0	0	0	0	0

UARTx_S2 field descriptions

Field	Description
7 LBKDIF	LIN Break Detect Interrupt Flag LBKDIF is set when LBKDE is set and a LIN break character is detected on the receiver input. The LIN break characters are 11 consecutive logic 0s if C1[M] = 0 or 12 consecutive logic 0s if C1[M] = 1. LBKDIF is set after receiving the last LIN break character. LBKDIF is cleared by writing a 1 to it. 0 No LIN break character detected. 1 LIN break character detected.
6 RXEDGIF	RxD Pin Active Edge Interrupt Flag RXEDGIF is set when an active edge occurs on the RxD pin. The active edge is falling if RXINV = 0, and rising if RXINV=1. RXEDGIF is cleared by writing a 1 to it. See for additional details. RXEDGIF description **NOTE:** The active edge is detected only in two wire mode and on receiving data coming from the RxD pin. 0 No active edge on the receive pin has occurred. 1 An active edge on the receive pin has occurred.
5 MSBF	Most Significant Bit First Setting this field reverses the order of the bits that are transmitted and received on the wire. This field does not affect the polarity of the bits, the location of the parity bit, or the location of the start or stop bits. This field is automatically set when C7816[INIT] and C7816[ISO7816E] are enabled and an initial character is detected in T = 0 protocol mode.In EN709 mode, this field affects the order of bits the same way as it does in normal mode. 0 LSB (bit0) is the first bit that is transmitted following the start bit. Further, the first bit received after the start bit is identified as bit0. 1 MSB (bit8, bit7 or bit6) is the first bit that is transmitted following the start bit, depending on the setting of C1[M] and C1[PE]. Further, the first bit received after the start bit is identified as bit8, bit7, or bit6, depending on the setting of C1[M] and C1[PE].
4 RXINV	Receive Data Inversion Setting this field reverses the polarity of the received data input. In NRZ format, a one is represented by a mark and a zero is represented by a space for normal polarity, and the opposite for inverted polarity. In IrDA format, a zero is represented by short high pulse in the middle of a bit time remaining idle low for a one for normal polarity. A zero is represented by a short low pulse in the middle of a bit time remaining idle high for a one for inverted polarity.This field is automatically set when C7816[INIT] and C7816[ISO7816E] are enabled and an initial character is detected in T = 0 protocol mode.In EN709 mode, this bit affects the polarity of bits the same as it does in normal mode. **NOTE:** Setting RXINV inverts the RxD input for data bits, start and stop bits, break, and idle. When C7816[ISO7816E] is set/enabled, only the data bits and the parity bit are inverted. 0 Receive data is not inverted. 1 Receive data is inverted.
3 RWUID	Receive Wakeup Idle Detect When RWU is set and WAKE is cleared, this field controls whether the idle character that wakes the receiver sets S1[IDLE]. This field must be cleared when C7816[ISO7816E] is set/enabled. 0 S1[IDLE] is not set upon detection of an idle character. 1 S1[IDLE] is set upon detection of an idle character.

Table continues on the next page...

UARTx_S2 field descriptions (continued)

Field	Description
2 BRK13	Break Transmit Character Length Determines whether the transmit break character is 10, 11, or 12 bits long, or 13 or 14 bits long. See for the length of the break character for the different configurations. The detection of a framing error is not affected by this field. Transmitting break characters 0 Break character is 10, 11, or 12 bits long. 1 Break character is 13 or 14 bits long.
1 LBKDE	LIN Break Detection Enable Selects a longer break character detection length. While LBKDE is set, S1[RDRF], S1[NF], S1[FE], and S1[PF] are prevented from setting. When LBKDE is set, see . Overrun operationLBKDE must be cleared when C7816[ISO7816E] is set. 0 Break character is detected at one of the following lengths: • 10 bit times if C1[M] = 0 • 11 bit times if C1[M] = 1 and C4[M10] = 0 • 12 bit times if C1[M] = 1, C4[M10] = 1, and S1[PE] = 1 1 Break character is detected at length of 11 bit times if C1[M] = 0 or 12 bits time if C1[M] = 1.
0 RAF	Receiver Active Flag RAF is set when the UART receiver detects a logic 0 during the RT1 time period of the start bit search. RAF is cleared when the receiver detects an idle character when C7816[ISO7816E] is cleared/disabled. When C7816[ISO7816E] is enabled, the RAF is cleared if the C7816[TTYPE] = 0 expires or the C7816[TTYPE] = 1 expires. **NOTE:** In case C7816[ISO7816E] is set and C7816[TTYPE] = 0, it is possible to configure the guard time to 12. However, if a NACK is required to be transmitted, the data transfer actually takes 13 ETU with the 13th ETU slot being a inactive buffer. Therefore, in this situation, the RAF may deassert one ETU prior to actually being inactive. 0 UART receiver idle/inactive waiting for a start bit. 1 UART receiver active, RxD input not idle.

45.3.7 UART Control Register 3 (UARTx_C3)

Writing R8 does not have any effect. TXDIR and TXINV can be changed only between transmit and receive packets.

Addresses: UART0_C3 is 4006_A000h base + 6h offset = 4006_A006h
UART1_C3 is 4006_B000h base + 6h offset = 4006_B006h
UART2_C3 is 4006_C000h base + 6h offset = 4006_C006h

Bit	7	6	5	4	3	2	1	0
Read	R8	T8	TXDIR	TXINV	ORIE	NEIE	FEIE	PEIE
Write	▨							
Reset	0	0	0	0	0	0	0	0

UARTx_C3 field descriptions

Field	Description
7 R8	Received Bit 8 R8 is the ninth data bit received when the UART is configured for 9-bit data format, that is, if C1[M] = 1 or C4[M10] = 1.
6 T8	Transmit Bit 8 T8 is the ninth data bit transmitted when the UART is configured for 9-bit data format, that is, if C1[M] = 1 or C4[M10] = 1. **NOTE:** If the value of T8 is the same as in the previous transmission, T8 does not have to be rewritten. The same value is transmitted until T8 is rewritten.
5 TXDIR	Transmitter Pin Data Direction in Single-Wire mode Determines whether the TXD pin is used as an input or output in the single-wire mode of operation. This field is relevant only to the single wire mode. When C7816[ISO7816E] is set/enabled and C7816[TTYPE] = 1, this field is automatically cleared after the requested block is transmitted. This condition is detected when TL7816[TLEN] = 0 and 4 additional characters are transmitted. Additionally, if C7816[ISO7816E] is set/enabled and C7816[TTYPE] = 0 and a NACK is being transmitted, the hardware automatically overrides this field as needed. In this situation, TXDIR does not reflect the temporary state associated with the NACK. 0 TXD pin is an input in single wire mode. 1 TXD pin is an output in single wire mode.
4 TXINV	Transmit Data Inversion. Setting this field reverses the polarity of the transmitted data output. In NRZ format, a one is represented by a mark and a zero is represented by a space for normal polarity, and the opposite for inverted polarity. In IrDA format, a zero is represented by short high pulse in the middle of a bit time remaining idle low for a one for normal polarity, and a zero is represented by short low pulse in the middle of a bit time remaining idle high for a one for inverted polarity.This field is automatically set when C7816[INIT] and C7816[ISO7816E] are enabled and an initial character is detected in T = 0 protocol mode. **NOTE:** Setting TXINV inverts all transmitted values, including idle, break, start, and stop bits. In loop mode, if TXINV is set, the receiver gets the transmit inversion bit when RXINV is disabled. When C7816[ISO7816E] is set/enabled then only the transmitted data bits and parity bit are inverted. 0 Transmit data is not inverted. 1 Transmit data is inverted.
3 ORIE	Overrun Error Interrupt Enable Enables the overrun error flag, S1[OR], to generate interrupt requests. 0 OR interrupts are disabled. 1 OR interrupt requests are enabled.
2 NEIE	Noise Error Interrupt Enable Enables the noise flag, S1[NF], to generate interrupt requests. 0 NF interrupt requests are disabled. 1 NF interrupt requests are enabled.
1 FEIE	Framing Error Interrupt Enable

Table continues on the next page...

UARTx_C3 field descriptions (continued)

Field	Description
	Enables the framing error flag, S1[FE], to generate interrupt requests. 0 FE interrupt requests are disabled. 1 FE interrupt requests are enabled.
0 PEIE	Parity Error Interrupt Enable Enables the parity error flag, S1[PF], to generate interrupt requests. 0 PF interrupt requests are disabled. 1 PF interrupt requests are enabled.

45.3.8 UART Data Register (UARTx_D)

This register is actually two separate registers. Reads return the contents of the read-only receive data register and writes go to the write-only transmit data register.

NOTE

- In 8-bit or 9-bit data format, only UART data register (D) needs to be accessed to clear the S1[RDRF] bit (assuming receiver buffer level is less than RWFIFO[RXWATER]). The C3 register needs to be read, prior to the D register, only if the ninth bit of data needs to be captured. Similarly, the ED register needs to be read, prior to the D register, only if the additional flag data for the dataword needs to be captured.
- In the normal 8-bit mode (M bit cleared) if the parity is enabled, you get seven data bits and one parity bit. That one parity bit is loaded into the D register. So, for the data bits, mask off the parity bit from the value you read out of this register.
- When transmitting in 9-bit data format and using 8-bit write instructions, write first to transmit bit 8 in UART control register 3 (C3[T8]), then D. A write to C3[T8] stores the data in a temporary register. If D register is written first, and then the new data on data bus is stored in D, the temporary value written by the last write to C3[T8] gets stored in the C3[T8] register.

Addresses: UART0_D is 4006_A000h base + 7h offset = 4006_A007h
UART1_D is 4006_B000h base + 7h offset = 4006_B007h
UART2_D is 4006_C000h base + 7h offset = 4006_C007h

Bit	7	6	5	4	3	2	1	0
Read Write	RT							
Reset	0	0	0	0	0	0	0	0

UARTx_D field descriptions

Field	Description
7–0 RT	Reads return the contents of the read-only receive data register and writes go to the write-only transmit data register.

45.3.9 UART Match Address Registers 1 (UARTx_MA1)

The MA1 and MA2 registers are compared to input data addresses when the most significant bit is set and the associated C4[MAEN] field is set. If a match occurs, the following data is transferred to the data register. If a match fails, the following data is discarded. These registers can be read and written at anytime.

Addresses: UART0_MA1 is 4006_A000h base + 8h offset = 4006_A008h
UART1_MA1 is 4006_B000h base + 8h offset = 4006_B008h
UART2_MA1 is 4006_C000h base + 8h offset = 4006_C008h

Bit	7	6	5	4	3	2	1	0
Read Write	MA							
Reset	0	0	0	0	0	0	0	0

UARTx_MA1 field descriptions

Field	Description
7–0 MA	Match Address

45.3.10 UART Match Address Registers 2 (UARTx_MA2)

These registers can be read and written at anytime. The MA1 and MA2 registers are compared to input data addresses when the most significant bit is set and the associated C4[MAEN] field is set. If a match occurs, the following data is transferred to the data register. If a match fails, the following data is discarded.

Addresses: UART0_MA2 is 4006_A000h base + 9h offset = 4006_A009h
UART1_MA2 is 4006_B000h base + 9h offset = 4006_B009h
UART2_MA2 is 4006_C000h base + 9h offset = 4006_C009h

Bit	7	6	5	4	3	2	1	0
Read Write	MA							
Reset	0	0	0	0	0	0	0	0

UARTx_MA2 field descriptions

Field	Description
7–0 MA	Match Address

45.3.11 UART Control Register 4 (UARTx_C4)

Addresses: UART0_C4 is 4006_A000h base + Ah offset = 4006_A00Ah
UART1_C4 is 4006_B000h base + Ah offset = 4006_B00Ah
UART2_C4 is 4006_C000h base + Ah offset = 4006_C00Ah

Bit	7	6	5	4	3	2	1	0
Read Write	MAEN1	MAEN2	M10	BRFA				
Reset	0	0	0	0	0	0	0	0

UARTx_C4 field descriptions

Field	Description
7 MAEN1	Match Address Mode Enable 1 See Match address operation for more information. 0 All data received is transferred to the data buffer if MAEN2 is cleared. 1 All data received with the most significant bit cleared, is discarded. All data received with the most significant bit set, is compared with contents of MA1 register. If no match occurs, the data is discarded. If match occurs, data is transferred to the data buffer. This field must be cleared when C7816[ISO7816E] is set/enabled.
6 MAEN2	Match Address Mode Enable 2

Table continues on the next page...

UARTx_C4 field descriptions (continued)

Field	Description
	See Match address operation for more information. 0 All data received is transferred to the data buffer if MAEN1 is cleared. 1 All data received with the most significant bit cleared, is discarded. All data received with the most significant bit set, is compared with contents of MA2 register. If no match occurs, the data is discarded. If a match occurs, data is transferred to the data buffer. This field must be cleared when C7816[ISO7816E] is set/enabled.
5 M10	10-bit Mode select Causes a tenth, non-memory mapped bit to be part of the serial transmission. This tenth bit is generated and interpreted as a parity bit. The M10 field does not affect the LIN send or detect break behavior. If M10 is set, then both C1[M] and C1[PE] must also be set. This field must be cleared when C7816[ISO7816E] is set/enabled. See Data format (non ISO-7816) for more information. 0 The parity bit is the ninth bit in the serial transmission. 1 The parity bit is the tenth bit in the serial transmission.
4–0 BRFA	Baud Rate Fine Adjust This bit field is used to add more timing resolution to the average baud frequency, in increments of 1/32. See Baud rate generation for more information.

45.3.12 UART Control Register 5 (UARTx_C5)

Addresses: UART0_C5 is 4006_A000h base + Bh offset = 4006_A00Bh
UART1_C5 is 4006_B000h base + Bh offset = 4006_B00Bh
UART2_C5 is 4006_C000h base + Bh offset = 4006_C00Bh

Bit	7	6	5	4	3	2	1	0
Read	TDMAS	0	RDMAS	0				
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_C5 field descriptions

Field	Description
7 TDMAS	Transmitter DMA Select Configures the transmit data register empty flag, S1[TDRE], to generate interrupt or DMA requests if C2[TIE] is set. **NOTE:** • If C2[TIE] is cleared, TDRE DMA and TDRE interrupt request signals are not asserted when the TDRE flag is set, regardless of the state of TDMAS. • If C2[TIE] and TDMAS are both set, then C2[TCIE] must be cleared, and D must not be written unless a DMA request is being serviced.

Table continues on the next page...

UARTx_C5 field descriptions (continued)

Field	Description
	0 If C2[TIE] is set and the S1[TDRE] flag is set, the TDRE interrupt request signal is asserted to request interrupt service. 1 If C2[TIE] is set and the S1[TDRE] flag is set, the TDRE DMA request signal is asserted to request a DMA transfer.
6 Reserved	This read-only field is reserved and always has the value zero.
5 RDMAS	Receiver Full DMA Select Configures the receiver data register full flag, S1[RDRF], to generate interrupt or DMA requests if C2[RIE] is set. **NOTE:** If C2[RIE] is cleared, and S1[RDRF] is set, the RDRF DMA and RDFR interrupt request signals are not asserted, regardless of the state of RDMAS. 0 If C2[RIE] and S1[RDRF] are set, the RDFR interrupt request signal is asserted to request an interrupt service. 1 If C2[RIE] and S1[RDRF] are set, the RDRF DMA request signal is asserted to request a DMA transfer.
4–0 Reserved	This read-only field is reserved and always has the value zero.

45.3.13 UART Extended Data Register (UARTx_ED)

This register contains additional information flags that are stored with a received dataword. This register may be read at any time but contains valid data only if there is a dataword in the receive FIFO.

NOTE

- The data contained in this register represents additional information regarding the conditions on which a dataword was received. The importance of this data varies with the application, and in some cases maybe completely optional. These fields automatically update to reflect the conditions of the next dataword whenever D is read.
- If S1[NF] and S1[PF] have not been set since the last time the receive buffer was empty, the NOISY and PARITYE fields will be zero.

Addresses: UART0_ED is 4006_A000h base + Ch offset = 4006_A00Ch
UART1_ED is 4006_B000h base + Ch offset = 4006_B00Ch
UART2_ED is 4006_C000h base + Ch offset = 4006_C00Ch

Bit	7	6	5	4	3	2	1	0
Read	NOISY	PARITYE	0					
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_ED field descriptions

Field	Description
7 NOISY	The current received dataword contained in D and C3[R8] was received with noise. 0 The dataword was received without noise. 1 The data was received with noise.
6 PARITYE	The current received dataword contained in D and C3[R8] was received with a parity error. 0 The dataword was received without a parity error. 1 The dataword was received with a parity error.
5–0 Reserved	This read-only field is reserved and always has the value zero.

45.3.14 UART Modem Register (UARTx_MODEM)

The MODEM register controls options for setting the modem configuration.

NOTE

RXRTSE, TXRTSPOL, TXRTSE, and TXCTSE must all be cleared when C7816[ISO7816EN] is enabled. This will cause the RTS to deassert during ISO-7816 wait times. The ISO-7816 protocol does not use the RTS and CTS signals.

Addresses: UART0_MODEM is 4006_A000h base + Dh offset = 4006_A00Dh
UART1_MODEM is 4006_B000h base + Dh offset = 4006_B00Dh
UART2_MODEM is 4006_C000h base + Dh offset = 4006_C00Dh

Bit	7	6	5	4	3	2	1	0
Read	0				RXRTSE	TXRTSPOL	TXRTSE	TXCTSE
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_MODEM field descriptions

Field	Description
7–4 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

UARTx_MODEM field descriptions (continued)

Field	Description
3 RXRTSE	Receiver request-to-send enable Allows the RTS output to control the CTS input of the transmitting device to prevent receiver overrun. **NOTE:** Do not set both RXRTSE and TXRTSE. 0 The receiver has no effect on RTS. 1 RTS is deasserted if the number of characters in the receiver data register (FIFO) is equal to or greater than RWFIFO[RXWATER]. RTS is asserted when the number of characters in the receiver data register (FIFO) is less than RWFIFO[RXWATER].
2 TXRTSPOL	Transmitter request-to-send polarity Controls the polarity of the transmitter RTS. TXRTSPOL does not affect the polarity of the receiver RTS. RTS will remain negated in the active low state unless TXRTSE is set. 0 Transmitter RTS is active low. 1 Transmitter RTS is active high.
1 TXRTSE	Transmitter request-to-send enable Controls RTS before and after a transmission. 0 The transmitter has no effect on RTS. 1 When a character is placed into an empty transmitter data buffer , RTS asserts one bit time before the start bit is transmitted. RTS deasserts one bit time after all characters in the transmitter data buffer and shift register are completely sent, including the last stop bit. (FIFO)(FIFO)
0 TXCTSE	Transmitter clear-to-send enable TXCTSE controls the operation of the transmitter. TXCTSE can be set independently from the state of TXRTSE and RXRTSE. 0 CTS has no effect on the transmitter. 1 Enables clear-to-send operation. The transmitter checks the state of CTS each time it is ready to send a character. If CTS is asserted, the character is sent. If CTS is deasserted, the signal TXD remains in the mark state and transmission is delayed until CTS is asserted. Changes in CTS as a character is being sent do not affect its transmission.

45.3.15 UART Infrared Register (UARTx_IR)

The IR register controls options for setting the infrared configuration.

Addresses: UART0_IR is 4006_A000h base + Eh offset = 4006_A00Eh
UART1_IR is 4006_B000h base + Eh offset = 4006_B00Eh
UART2_IR is 4006_C000h base + Eh offset = 4006_C00Eh

Bit	7	6	5	4	3	2	1	0
Read	0					IREN	TNP	
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_IR field descriptions

Field	Description
7–3 Reserved	This read-only field is reserved and always has the value zero.
2 IREN	Infrared enable Enables/disables the infrared modulation/demodulation. 0 IR disabled. 1 IR enabled.
1–0 TNP	Transmitter narrow pulse Enables whether the UART transmits a 1/16, 3/16, 1/32, or 1/4 narrow pulse. 00 3/16. 01 1/16. 10 1/32. 11 1/4.

45.3.16 UART FIFO Parameters (UARTx_PFIFO)

This register provides the ability for the programmer to turn on and off FIFO functionality. It also provides the size of the FIFO that has been implemented. This register may be read at any time. This register must be written only when C2[RE] and C2[TE] are cleared/not set and when the data buffer/FIFO is empty.

Addresses: UART0_PFIFO is 4006_A000h base + 10h offset = 4006_A010h
UART1_PFIFO is 4006_B000h base + 10h offset = 4006_B010h
UART2_PFIFO is 4006_C000h base + 10h offset = 4006_C010h

Bit	7	6	5	4	3	2	1	0
Read	TXFE	TXFIFOSIZE			RXFE	RXFIFOSIZE		
Write		▨	▨	▨		▨	▨	▨
Reset	0	*	*	*	0	*	*	*

* Notes:

- TXFIFOSIZE bitfield: The reset value depends on whether the specific UART instance supports the FIFO and on the size of that FIFO. See the Chip Configuration details for more information on the FIFO size supported for each UART instance.
- RXFIFOSIZE bitfield: The reset value depends on whether the specific UART instance supports the FIFO and on the size of that FIFO. See the Chip Configuration details for more information on the FIFO size supported for each UART instance.

UARTx_PFIFO field descriptions

Field	Description
7 TXFE	Transmit FIFO Enable

Table continues on the next page...

UARTx_PFIFO field descriptions (continued)

Field	Description
	When this field is set, the built in FIFO structure for the transmit buffer is enabled. The size of the FIFO structure is indicated by TXFIFOSIZE. If this field is not set, the transmit buffer operates as a FIFO of depth one dataword regardless of the value in TXFIFOSIZE. Both C2[TE] and C2[RE] must be cleared prior to changing this field. Additionally, TXFLUSH and RXFLUSH commands must be issued immediately after changing this field. 0 Transmit FIFO is not enabled. Buffer is depth 1. (Legacy support). 1 Transmit FIFO is enabled. Buffer is depth indicated by TXFIFOSIZE.
6–4 TXFIFOSIZE	Transmit FIFO. Buffer Depth The maximum number of transmit datawords that can be stored in the transmit buffer. This field is read only. 000 Transmit FIFO/Buffer depth = 1 dataword. 001 Transmit FIFO/Buffer depth = 4 datawords. 010 Transmit FIFO/Buffer depth = 8 datawords. 011 Transmit FIFO/Buffer depth = 16 datawords. 100 Transmit FIFO/Buffer depth = 32 datawords. 101 Transmit FIFO/Buffer depth = 64 datawords. 110 Transmit FIFO/Buffer depth = 128 datawords. 111 Reserved.
3 RXFE	Receive FIFO Enable When this field is set, the built in FIFO structure for the receive buffer is enabled. The size of the FIFO structure is indicated by the RXFIFOSIZE field. If this field is not set, the receive buffer operates as a FIFO of depth one dataword regardless of the value in RXFIFOSIZE. Both C2[TE] and C2[RE] must be cleared prior to changing this field. Additionally, TXFLUSH and RXFLUSH commands must be issued immediately after changing this field. 0 Receive FIFO is not enabled. Buffer is depth 1. (Legacy support) 1 Receive FIFO is enabled. Buffer is depth indicted by RXFIFOSIZE.
2–0 RXFIFOSIZE	Receive FIFO. Buffer Depth The maximum number of receive datawords that can be stored in the receive buffer before an overrun occurs. This field is read only. 000 Receive FIFO/Buffer depth = 1 dataword. 001 Receive FIFO/Buffer depth = 4 datawords. 010 Receive FIFO/Buffer depth = 8 datawords. 011 Receive FIFO/Buffer depth = 16 datawords. 100 Receive FIFO/Buffer depth = 32 datawords. 101 Receive FIFO/Buffer depth = 64 datawords. 110 Receive FIFO/Buffer depth = 128 datawords. 111 Reserved.

45.3.17 UART FIFO Control Register (UARTx_CFIFO)

This register provides the ability to program various control fields for FIFO operation. This register may be read or written at any time. Note that writing to TXFLUSH and RXFLUSH may result in data loss and requires careful action to prevent unintended/ unpredictable behavior. Therefore, it is recommended that TE and RE be cleared prior to flushing the corresponding FIFO.

Addresses: UART0_CFIFO is 4006_A000h base + 11h offset = 4006_A011h
UART1_CFIFO is 4006_B000h base + 11h offset = 4006_B011h
UART2_CFIFO is 4006_C000h base + 11h offset = 4006_C011h

Bit	7	6	5	4	3	2	1	0
Read	0	0	0			RXOFE	TXOFE	RXUFE
Write	TXFLUSH	RXFLUSH						
Reset	0	0	0	0	0	0	0	0

UARTx_CFIFO field descriptions

Field	Description
7 TXFLUSH	Transmit FIFO/Buffer Flush Writing to this field causes all data that is stored in the transmit FIFO/buffer to be flushed. This does not affect data that is in the transmit shift register. 0 No flush operation occurs. 1 All data in the transmit FIFO/Buffer is cleared out.
6 RXFLUSH	Receive FIFO/Buffer Flush Writing to this field causes all data that is stored in the receive FIFO/buffer to be flushed. This does not affect data that is in the receive shift register. 0 No flush operation occurs. 1 All data in the receive FIFO/buffer is cleared out.
5–3 Reserved	This read-only field is reserved and always has the value zero.
2 RXOFE	Receive FIFO Overflow Interrupt Enable When this field is set, the RXOF flag generates an interrupt to the host. 0 RXOF flag does not generate an interrupt to the host. 1 RXOF flag generates an interrupt to the host.
1 TXOFE	Transmit FIFO Overflow Interrupt Enable When this field is set, the TXOF flag generates an interrupt to the host. 0 TXOF flag does not generate an interrupt to the host. 1 TXOF flag generates an interrupt to the host.

Table continues on the next page...

UARTx_CFIFO field descriptions (continued)

Field	Description
0 RXUFE	Receive FIFO Underflow Interrupt Enable When this field is set, the RXUF flag generates an interrupt to the host. 0 RXUF flag does not generate an interrupt to the host. 1 RXUF flag generates an interrupt to the host.

45.3.18 UART FIFO Status Register (UARTx_SFIFO)

This register provides status information regarding the transmit and receiver buffers/FIFOs, including interrupt information. This register may be written to or read at any time.

Addresses: UART0_SFIFO is 4006_A000h base + 12h offset = 4006_A012h
UART1_SFIFO is 4006_B000h base + 12h offset = 4006_B012h
UART2_SFIFO is 4006_C000h base + 12h offset = 4006_C012h

Bit	7	6	5	4	3	2	1	0
Read	TXEMPT	RXEMPT	0			RXOF	TXOF	RXUF
Write	▨	▨	▨					
Reset	1	1	0	0	0	0	0	0

UARTx_SFIFO field descriptions

Field	Description
7 TXEMPT	Transmit Buffer/FIFO Empty Asserts when there is no data in the Transmit FIFO/buffer. This field does not take into account data that is in the transmit shift register. 0 Transmit buffer is not empty. 1 Transmit buffer is empty.
6 RXEMPT	Receive Buffer/FIFO Empty Asserts when there is no data in the receive FIFO/Buffer. This field does not take into account data that is in the receive shift register. 0 Receive buffer is not empty. 1 Receive buffer is empty.
5–3 Reserved	This read-only field is reserved and always has the value zero.
2 RXOF	Receiver Buffer Overflow Flag Indicates that more data has been written to the receive buffer than it can hold. This field will assert regardless of the value of CFIFO[RXOFE]. However, an interrupt will be issued to the host only if CFIFO[RXOFE] is set. This flag is cleared by writing a 1.

Table continues on the next page...

UARTx_SFIFO field descriptions (continued)

Field	Description
	0 No receive buffer overflow has occurred since the last time the flag was cleared. 1 At least one receive buffer overflow has occurred since the last time the flag was cleared.
1 TXOF	Transmitter Buffer Overflow Flag Indicates that more data has been written to the transmit buffer than it can hold. This field will assert regardless of the value of CFIFO[TXOFE]. However, an interrupt will be issued to the host only if CFIFO[TXOFE] is set. This flag is cleared by writing a 1. 0 No transmit buffer overflow has occurred since the last time the flag was cleared. 1 At least one transmit buffer overflow has occurred since the last time the flag was cleared.
0 RXUF	Receiver Buffer Underflow Flag Indicates that more data has been read from the receive buffer than was present. This field will assert regardless of the value of CFIFO[RXUFE]. However, an interrupt will be issued to the host only if CFIFO[RXUFE] is set. This flag is cleared by writing a 1. 0 No receive buffer underflow has occurred since the last time the flag was cleared. 1 At least one receive buffer underflow has occurred since the last time the flag was cleared.

45.3.19 UART FIFO Transmit Watermark (UARTx_TWFIFO)

This register provides the ability to set a programmable threshold for notification of needing additional transmit data. This register may be read at any time but must be written only when C2[TE] is not set. Changing the value of the watermark will not clear the S1[TDRE] flag.

Addresses: UART0_TWFIFO is 4006_A000h base + 13h offset = 4006_A013h
UART1_TWFIFO is 4006_B000h base + 13h offset = 4006_B013h
UART2_TWFIFO is 4006_C000h base + 13h offset = 4006_C013h

Bit	7	6	5	4	3	2	1	0
Read Write	TXWATER							
Reset	0	0	0	0	0	0	0	0

UARTx_TWFIFO field descriptions

Field	Description
7–0 TXWATER	Transmit Watermark When the number of datawords in the transmit FIFO/buffer is equal to or less than the value in this register field, an interrupt via S1[TDRE] or a DMA request via C5[TDMAS] is generated as determined by C5[TDMAS] and C2[TIE]. For proper operation, the value in TXWATER must be set to be less than the size of the transmit buffer/FIFO size as indicated by PFIFO[TXFIFOSIZE] and PFIFO[TXFE].

45.3.20 UART FIFO Transmit Count (UARTx_TCFIFO)

This is a read only register that indicates how many datawords are currently in the transmit buffer/FIFO. It may be read at any time.

Addresses: UART0_TCFIFO is 4006_A000h base + 14h offset = 4006_A014h
UART1_TCFIFO is 4006_B000h base + 14h offset = 4006_B014h
UART2_TCFIFO is 4006_C000h base + 14h offset = 4006_C014h

Bit	7	6	5	4	3	2	1	0
Read	TXCOUNT							
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_TCFIFO field descriptions

Field	Description
7–0 TXCOUNT	Transmit Counter The value in this register indicates the number of datawords that are in the transmit FIFO/buffer. If a dataword is being transmitted, that is, in the transmit shift register, it is not included in the count. This value may be used in conjunction with PFIFO[TXFIFOSIZE] to calculate how much room is left in the transmit FIFO/buffer.

45.3.21 UART FIFO Receive Watermark (UARTx_RWFIFO)

This register provides the ability to set a programmable threshold for notification of the need to remove data from the receiver FIFO/buffer. This register may be read at any time but must be written only when C2[RE] is not asserted. Changing the value in this register will not clear S1[RDRF].

Addresses: UART0_RWFIFO is 4006_A000h base + 15h offset = 4006_A015h
UART1_RWFIFO is 4006_B000h base + 15h offset = 4006_B015h
UART2_RWFIFO is 4006_C000h base + 15h offset = 4006_C015h

Bit	7	6	5	4	3	2	1	0
Read Write	RXWATER							
Reset	0	0	0	0	0	0	0	1

UARTx_RWFIFO field descriptions

Field	Description
7–0 RXWATER	Receive Watermark When the number of datawords in the receive FIFO/buffer is equal to or greater than the value in this register field, an interrupt via S1[RDRF] or a DMA request via C5[RDMAS] is generated as determined by C5[RDMAS] and C2[RIE]. For proper operation, the value in RXWATER must be set to be less than the

UARTx_RWFIFO field descriptions (continued)

Field	Description
	receive FIFO/buffer size as indicated by PFIFO[RXFIFOSIZE] and PFIFO[RXFE] and must be greater than 0.

45.3.22 UART FIFO Receive Count (UARTx_RCFIFO)

This is a read only register that indicates how many datawords are currently in the receive FIFO/buffer. It may be read at any time.

Addresses: UART0_RCFIFO is 4006_A000h base + 16h offset = 4006_A016h
UART1_RCFIFO is 4006_B000h base + 16h offset = 4006_B016h
UART2_RCFIFO is 4006_C000h base + 16h offset = 4006_C016h

Bit	7	6	5	4	3	2	1	0
Read	RXCOUNT							
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_RCFIFO field descriptions

Field	Description
7–0 RXCOUNT	Receive Counter The value in this register indicates the number of datawords that are in the receive FIFO/buffer. If a dataword is being received, that is, in the receive shift register, it is not included in the count. This value may be used in conjunction with PFIFO[RXFIFOSIZE] to calculate how much room is left in the receive FIFO/buffer.

45.3.23 UART 7816 Control Register (UARTx_C7816)

The C7816 register is the primary control register for ISO-7816 specific functionality. This register is specific to 7816 functionality and the values in this register have no effect on UART operation and should be ignored if ISO_7816E is not set/enabled. This register may be read at any time but values must be changed only when ISO_7816E is not set.

Addresses: UART0_C7816 is 4006_A000h base + 18h offset = 4006_A018h
UART1_C7816 is 4006_B000h base + 18h offset = 4006_B018h
UART2_C7816 is 4006_C000h base + 18h offset = 4006_C018h

Bit	7	6	5	4	3	2	1	0
Read	0			ONACK	ANACK	INIT	TTYPE	ISO_7816E
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_C7816 field descriptions

Field	Description
7–5 Reserved	This read-only field is reserved and always has the value zero.
4 ONACK	Generate NACK on Overflow When this field is set, the receiver automatically generates a NACK response if a receive buffer overrun occurs, as indicated by S1[OR]. In many systems, this results in the transmitter resending the packet that overflowed until the retransmit threshold for that transmitter is reached. A NACK is generated only if TTYPE=0. This field operates independently of ANACK. See . Overrun NACK considerations 0 The received data does not generate a NACK when the receipt of the data results in an overflow event. 1 If the receiver buffer overflows, a NACK is automatically sent on a received character.
3 ANACK	Generate NACK on Error When this field is set, the receiver automatically generates a NACK response if a parity error occurs or if INIT is set and an invalid initial character is detected. A NACK is generated only if TTYPE = 0. If ANACK is set, the UART attempts to retransmit the data indefinitely. To stop retransmission attempts, clear C2[TE] or ISO_7816E and do not set until S1[TC] sets C2[TE] again. 0 No NACK is automatically generated. 1 A NACK is automatically generated if a parity error is detected or if an invalid initial character is detected.
2 INIT	Detect Initial Character When this field is set, all received characters are searched for a valid initial character. If an invalid initial character is identified, and ANACK is set, a NACK is sent. All received data is discarded and error flags blocked (S1[NF], S1[OR], S1[FE], S1[PF], IS7816[WT], IS7816[CWT], IS7816[BWT], IS7816[GTV]) until a valid initial character is detected. Upon detecting a valid initial character, the configuration values S2[MSBF], C3[TXINV], and S2[RXINV] are automatically updated to reflect the initial character that was received. The actual INIT data value is not stored in the receive buffer. Additionally, upon detection of a valid initial character, IS7816[INITD] is set and an interrupt issued as programmed by IE7816[INITDE]. When a valid initial character is detected, INIT is automatically cleared. This Initial Character Detect feature is supported only in T = 0 protocol mode. 0 Normal operating mode. Receiver does not seek to identify initial character. 1 Receiver searches for initial character.
1 TTYPE	Transfer Type Indicates the transfer protocol being used. See ISO-7816 / smartcard support for more details. 0 T = 0 per the ISO-7816 specification. 1 T = 1 per the ISO-7816 specification.
0 ISO_7816E	ISO-7816 Functionality Enabled Indicates that the UART is operating according to the ISO-7816 protocol. **NOTE:** This field must be modified only when no transmit or receive is occurring. If this field is changed during a data transfer, the data being transmitted or received may be transferred incorrectly. 0 ISO-7816 functionality is turned off/not enabled. 1 ISO-7816 functionality is turned on/enabled.

45.3.24 UART 7816 Interrupt Enable Register (UARTx_IE7816)

The IE7816 register controls which flags result in an interrupt being issued. This register is specific to 7816 functionality, the corresponding flags that drive the interrupts are not asserted when 7816E is not set/enabled. However, these flags may remain set if they are asserted while 7816E was set and not subsequently cleared. This register may be read or written to at any time.

Addresses: UART0_IE7816 is 4006_A000h base + 19h offset = 4006_A019h
UART1_IE7816 is 4006_B000h base + 19h offset = 4006_B019h
UART2_IE7816 is 4006_C000h base + 19h offset = 4006_C019h

Bit	7	6	5	4	3	2	1	0
Read	WTE	CWTE	BWTE	INITDE	0	GTVE	TXTE	RXTE
Write					▨			
Reset	0	0	0	0	0	0	0	0

UARTx_IE7816 field descriptions

Field	Description
7 WTE	Wait Timer Interrupt Enable 0 The assertion of IS7816[WT] does not result in the generation of an interrupt. 1 The assertion of IS7816[WT] results in the generation of an interrupt.
6 CWTE	Character Wait Timer Interrupt Enable 0 The assertion of IS7816[CWT] does not result in the generation of an interrupt. 1 The assertion of IS7816[CWT] results in the generation of an interrupt.
5 BWTE	Block Wait Timer Interrupt Enable 0 The assertion of IS7816[BWT] does not result in the generation of an interrupt. 1 The assertion of IS7816[BWT] results in the generation of an interrupt.
4 INITDE	Initial Character Detected Interrupt Enable 0 The assertion of IS7816[INITD] does not result in the generation of an interrupt. 1 The assertion of IS7816[INITD] results in the generation of an interrupt.
3 Reserved	This read-only field is reserved and always has the value zero.
2 GTVE	Guard Timer Violated Interrupt Enable 0 The assertion of IS7816[GTV] does not result in the generation of an interrupt. 1 The assertion of IS7816[GTV] results in the generation of an interrupt.
1 TXTE	Transmit Threshold Exceeded Interrupt Enable 0 The assertion of IS7816[TXT] does not result in the generation of an interrupt. 1 The assertion of IS7816[TXT] results in the generation of an interrupt.

Table continues on the next page...

UARTx_IE7816 field descriptions (continued)

Field	Description
0 RXTE	Receive Threshold Exceeded Interrupt Enable 0 The assertion of IS7816[RXT] does not result in the generation of an interrupt. 1 The assertion of IS7816[RXT] results in the generation of an interrupt.

45.3.25 UART 7816 Interrupt Status Register (UARTx_IS7816)

The IS7816 register provides a mechanism to read and clear the interrupt flags. All flags/ interrupts are cleared by writing a 1 to the field location. Writing a 0 has no effect. All bits are "sticky", meaning they indicate that only the flag condition that occurred since the last time the bit was cleared, not that the condition currently exists. The status flags are set regardless of whether the corresponding field in the IC7816 is set or cleared. The IC7816 controls only if an interrupt is issued to the host processor. This register is specific to 7816 functionality and the values in this register have no affect on UART operation and should be ignored if 7816E is not set/enabled. This register may be read or written at anytime.

Addresses: UART0_IS7816 is 4006_A000h base + 1Ah offset = 4006_A01Ah
UART1_IS7816 is 4006_B000h base + 1Ah offset = 4006_B01Ah
UART2_IS7816 is 4006_C000h base + 1Ah offset = 4006_C01Ah

Bit	7	6	5	4	3	2	1	0
Read	WT	CWT	BWT	INITD	0	GTV	TXT	RXT
Write					▨			
Reset	0	0	0	0	0	0	0	0

UARTx_IS7816 field descriptions

Field	Description
7 WT	Wait Timer Interrupt Indicates that the wait time, the time between the leading edge of a character being transmitted and the leading edge of the next response character, has exceeded the programmed value. This flag asserts only when C7816[TTYPE] = 0. This interrupt is cleared by writing 1. 0 Wait time (WT) has not been violated. 1 Wait time (WT) has been violated.
6 CWT	Character Wait Timer Interrupt Indicates that the character wait time, the time between the leading edges of two consecutive characters in a block, has exceeded the programmed value. This flag asserts only when C7816[TTYPE] = 1. This interrupt is cleared by writing 1. 0 Character wait time (CWT) has not been violated. 1 Character wait time (CWT) has been violated.

Table continues on the next page...

UARTx_IS7816 field descriptions (continued)

Field	Description
5 BWT	Block Wait Timer Interrupt Indicates that the block wait time, the time between the leading edge of first received character of a block and the leading edge of the last character the previously transmitted block, has exceeded the programmed value. This flag asserts only when C7816[TTYPE] = 1.This interrupt is cleared by writing 1. 0 Block wait time (BWT) has not been violated. 1 Block wait time (BWT) has been violated.
4 INITD	Initial Character Detected Interrupt Indicates that a valid initial character is received. This interrupt is cleared by writing 1. 0 A valid initial character has not been received. 1 A valid initial character has been received.
3 Reserved	This read-only field is reserved and always has the value zero.
2 GTV	Guard Timer Violated Interrupt Indicates that one or more of the character guard time, block guard time, or guard time are violated. This interrupt is cleared by writing 1. 0 A guard time (GT, CGT, or BGT) has not been violated. 1 A guard time (GT, CGT, or BGT) has been violated.
1 TXT	Transmit Threshold Exceeded Interrupt Indicates that the transmit NACK threshold has been exceeded as indicated by ET7816[TXTHRESHOLD]. Regardless of whether this flag is set, the UART continues to retransmit indefinitely. This flag asserts only when C7816[TTYPE] = 0. If 7816E is cleared/disabled, ANACK is cleared/disabled, C2[TE] is cleared/disabled, C7816[TTYPE] = 1, or packet is transferred without receiving a NACK, the internal NACK detection counter is cleared and the count restarts from zero on the next received NACK. This interrupt is cleared by writing 1. 0 The number of retries and corresponding NACKS does not exceed the value in ET7816[TXTHRESHOLD]. 1 The number of retries and corresponding NACKS exceeds the value in ET7816[TXTHRESHOLD].
0 RXT	Receive Threshold Exceeded Interrupt Indicates that there are more than ET7816[RXTHRESHOLD] consecutive NACKS generated in response to parity errors on received data. This flag requires ANACK to be set. Additionally, this flag asserts only when C7816[TTYPE] = 0. Clearing this field also resets the counter keeping track of consecutive NACKS. The UART will continue to attempt to receive data regardless of whether this flag is set. If 7816E is cleared/disabled, RE is cleared/disabled, C7816[TTYPE] = 1, or packet is received without needing to issue a NACK, the internal NACK detection counter is cleared and the count restarts from zero on the next transmitted NACK. This interrupt is cleared by writing 1. 0 The number of consecutive NACKS generated as a result of parity errors and buffer overruns is less than or equal to the value in ET7816[RXTHRESHOLD]. 1 The number of consecutive NACKS generated as a result of parity errors and buffer overruns is greater than the value in ET7816[RXTHRESHOLD].

45.3.26 UART 7816 Wait Parameter Register (UARTx_WP7816T0)

The WP7816 register contains constants used in the generation of various wait timer counters. To save register space, this register is used differently when C7816[TTYPE] = 0 and C7816[TTYPE] = 1. This register may be read at any time. This register must be written to only when C7816[ISO_7816E] is not set.

Addresses: UART0_WP7816T0 is 4006_A000h base + 1Bh offset = 4006_A01Bh
UART1_WP7816T0 is 4006_B000h base + 1Bh offset = 4006_B01Bh
UART2_WP7816T0 is 4006_C000h base + 1Bh offset = 4006_C01Bh

Bit	7	6	5	4	3	2	1	0
Read Write	WI							
Reset	0	0	0	0	1	0	1	0

UARTx_WP7816T0 field descriptions

Field	Description
7–0 WI	Wait Timer Interrupt (C7816[TTYPE] = 0) Used to calculate the value used for the WT counter. It represents a value between 1 and 255. The value of zero is not valid. This value is used only when C7816[TTYPE] = 0. See . Wait time and guard time parameters

45.3.27 UART 7816 Wait Parameter Register (UARTx_WP7816T1)

The WP7816 register contains constants used in the generation of various wait timer counters. To save register space, this register is used differently when C7816[TTYPE] = 0 and C7816[TTYPE] = 1. This register may be read at any time. This register must be written to only when C7816[ISO_7816E] is not set.

Addresses: UART0_WP7816T1 is 4006_A000h base + 1Bh offset = 4006_A01Bh
UART1_WP7816T1 is 4006_B000h base + 1Bh offset = 4006_B01Bh
UART2_WP7816T1 is 4006_C000h base + 1Bh offset = 4006_C01Bh

Bit	7	6	5	4	3	2	1	0
Read Write	CWI				BWI			
Reset	0	0	0	0	1	0	1	0

UARTx_WP7816T1 field descriptions

Field	Description
7–4 CWI	Character Wait Time Integer (C7816[TTYPE] = 1)

Table continues on the next page...

UARTx_WP7816T1 field descriptions (continued)

Field	Description
	Used to calculate the value used for the CWT counter. It represents a value between 0 and 15. This value is used only when C7816[TTYPE] = 1. See Wait time and guard time parameters .
3–0 BWI	Block Wait Time Integer(C7816[TTYPE] = 1) Used to calculate the value used for the BWT counter. It represent a value between 0 and 15. This value is used only when C7816[TTYPE] = 1. See Wait time and guard time parameters .

45.3.28 UART 7816 Wait N Register (UARTx_WN7816)

The WN7816 register contains a parameter that is used in the calculation of the guard time counter. This register may be read at any time. This register must be written to only when C7816[ISO_7816E] is not set.

Addresses: UART0_WN7816 is 4006_A000h base + 1Ch offset = 4006_A01Ch
UART1_WN7816 is 4006_B000h base + 1Ch offset = 4006_B01Ch
UART2_WN7816 is 4006_C000h base + 1Ch offset = 4006_C01Ch

Bit	7	6	5	4	3	2	1	0
Read Write	GTN							
Reset	0	0	0	0	0	0	0	0

UARTx_WN7816 field descriptions

Field	Description
7–0 GTN	Guard Band N Defines a parameter used in the calculation of GT, CGT, and BGT counters. The value represents an integer number between 0 and 255. See Wait time and guard time parameters .

45.3.29 UART 7816 Wait FD Register (UARTx_WF7816)

The WF7816 contains parameters that are used in the generation of various counters including GT, CGT, BGT, WT, and BWT. This register may be read at any time. This register must be written to only when C7816[ISO_7816E] is not set.

Addresses: UART0_WF7816 is 4006_A000h base + 1Dh offset = 4006_A01Dh
UART1_WF7816 is 4006_B000h base + 1Dh offset = 4006_B01Dh
UART2_WF7816 is 4006_C000h base + 1Dh offset = 4006_C01Dh

Bit	7	6	5	4	3	2	1	0
Read Write	GTFD							
Reset	0	0	0	0	0	0	0	1

UARTx_WF7816 field descriptions

Field	Description
7–0 GTFD	FD Multiplier Used as another multiplier in the calculation of WT and BWT. This value represents a number between 1 and 255. The value of 0 is invalid. This value is not used in baud rate generation. See Wait time and guard time parameters and Baud rate generation .

45.3.30 UART 7816 Error Threshold Register (UARTx_ET7816)

The ET7816 register contains fields that determine the number of NACKs that must be received or transmitted before the host processor is notified. This register may be read at anytime. This register must be written to only when C7816[ISO_7816E] is not set.

Addresses: UART0_ET7816 is 4006_A000h base + 1Eh offset = 4006_A01Eh
UART1_ET7816 is 4006_B000h base + 1Eh offset = 4006_B01Eh
UART2_ET7816 is 4006_C000h base + 1Eh offset = 4006_C01Eh

Bit	7	6	5	4	3	2	1	0
Read Write	TXTHRESHOLD				RXTHRESHOLD			
Reset	0	0	0	0	0	0	0	0

UARTx_ET7816 field descriptions

Field	Description
7–4 TXTHRESHOLD	Transmit NACK Threshold The value written to this field indicates the maximum number of failed attempts (NACKs) a transmitted character can have before the host processor is notified. This field is meaningful only when C7816[TTYPE] = 0 and C7816[ANACK] = 1. The value read from this field represents the number of consecutive NACKs that have been received since the last successful transmission. This counter saturates at 4'hF and does not wrap around. Regardless of how many NACKs that are received, the UART continues to retransmit indefinitely. This flag only asserts when C7816[TTYPE] = 0. For additional information see the IS7816[TXT] field description. 0 TXT asserts on the first NACK that is received. 1 TXT asserts on the second NACK that is received.
3–0 RXTHRESHOLD	Receive NACK Threshold The value written to this field indicates the maximum number of consecutive NACKs generated as a result of a parity error or receiver buffer overruns before the host processor is notified. After the counter exceeds that value in the field, the IS7816[RXT] is asserted. This field is meaningful only when C7816[TTYPE] = 0. The value read from this field represents the number of consecutive NACKs that have been transmitted since the last successful reception. This counter saturates at 4'hF and does not wrap around. Regardless of the number of NACKs sent, the UART continues to receive valid packets indefinitely. For additional information, see IS7816[RXT] field description.

45.3.31 UART 7816 Transmit Length Register (UARTx_TL7816)

The TL7816 register is used to indicate the number of characters contained in the block being transmitted. This register is used only when C7816[TTYPE] = 1. This register may be read at anytime. This register must be written only when C2[TE] is not enabled.

Addresses: UART0_TL7816 is 4006_A000h base + 1Fh offset = 4006_A01Fh
UART1_TL7816 is 4006_B000h base + 1Fh offset = 4006_B01Fh
UART2_TL7816 is 4006_C000h base + 1Fh offset = 4006_C01Fh

Bit	7	6	5	4	3	2	1	0
Read Write	TLEN							
Reset	0	0	0	0	0	0	0	0

UARTx_TL7816 field descriptions

Field	Description
7–0 TLEN	Transmit Length This value plus four indicates the number of characters contained in the block being transmitted. This register is automatically decremented by 1 for each character in the information field portion of the block. Additionally, this register is automatically decremented by 1 for the first character of a CRC in the epilogue field. Therefore, this register must be programmed with the number of bytes in the data packet if an LRC is being transmitted, and the number of bytes + 1 if a CRC is being transmitted. This register is not decremented for characters that are assumed to be part of the Prologue field, that is, the first three characters transmitted in a block, or the LRC or last CRC character in the Epilogue field, that is, the last character transmitted. This field must be programed or adjusted only when C2[TE] is cleared.

45.3.32 UART CEA709.1-B Control Register 6 (UARTx_C6)

Addresses: UART0_C6 is 4006_A000h base + 21h offset = 4006_A021h
UART1_C6 is 4006_B000h base + 21h offset = 4006_B021h
UART2_C6 is 4006_C000h base + 21h offset = 4006_C021h

Bit	7	6	5	4	3	2	1	0
Read	EN709	TX709	CE	CP	0			
Write					▨			
Reset	0	0	0	0	0	0	0	0

UARTx_C6 field descriptions

Field	Description
7 EN709	EN709 Enables the CEA709.1-B feature.

Table continues on the next page...

UARTx_C6 field descriptions (continued)

Field	Description
	0 CEA709.1-B is disabled. 1 CEA709.1-B is enabled
6 TX709	CEA709.1-B Transmit Enable Starts CEA709.1-B transmission. 0 CEA709.1-B transmitter is disabled. 1 CEA709.1-B transmitter is enabled.
5 CE	Collision Enable Enables the collision detect functionality. 0 Collision detect feature is disabled. 1 Collision detect feature is enabled.
4 CP	Collision Signal Polarity Indicates the polarity of the collision signal. 0 Collision signal is active low. 1 Collision signal is active high.
3–0 Reserved	This read-only field is reserved and always has the value zero.

45.3.33 UART CEA709.1-B Packet Cycle Time Counter High (UARTx_PCTH)

Addresses: UART0_PCTH is 4006_A000h base + 22h offset = 4006_A022h
UART1_PCTH is 4006_B000h base + 22h offset = 4006_B022h
UART2_PCTH is 4006_C000h base + 22h offset = 4006_C022h

Bit	7	6	5	4	3	2	1	0
Read Write	PCTH							
Reset	0	0	0	0	0	0	0	0

UARTx_PCTH field descriptions

Field	Description
7–0 PCTH	Packet Cycle Time Counter High Indicates the most significant byte of maximum period after the line code violation for which the bus could remain idle without decrementing back log count. If the time elapsed after line code violation is greater than packet cycle time, then packet cycle timer expired interrupt is generated. It is measured in terms of bit times, that is, the time it takes for a single bit or one differential Manchester symbol to be transmitted. This is medium dependent and hence does not usually require adjustment and is programmed only once.

45.3.34 UART CEA709.1-B Packet Cycle Time Counter Low (UARTx_PCTL)

Addresses: UART0_PCTL is 4006_A000h base + 23h offset = 4006_A023h
UART1_PCTL is 4006_B000h base + 23h offset = 4006_B023h
UART2_PCTL is 4006_C000h base + 23h offset = 4006_C023h

Bit	7	6	5	4	3	2	1	0
Read Write	PCTL							
Reset	0	0	0	0	0	0	0	0

UARTx_PCTL field descriptions

Field	Description
7–0 PCTL	Packet Cycle Time Counter Low Indicates the least significant byte of maximum period after the line code violation for which the bus could remain idle without decrementing back log count. If the time elapsed after line code violation is greater than packet cycle time, then packet cycle timer expired interrupt is generated. It is measured in terms of bit times, that is, the time it takes for a single bit or one Differential Manchester symbol to be transmitted. This is medium dependent and therefore does not usually require adjustment and is programmed only once.

45.3.35 UART CEA709.1-B Beta1 Timer (UARTx_B1T)

Addresses: UART0_B1T is 4006_A000h base + 24h offset = 4006_A024h
UART1_B1T is 4006_B000h base + 24h offset = 4006_B024h
UART2_B1T is 4006_C000h base + 24h offset = 4006_C024h

Bit	7	6	5	4	3	2	1	0
Read Write	B1T							
Reset	0	0	0	0	0	0	0	0

UARTx_B1T field descriptions

Field	Description
7–0 B1T	Beta1 Timer Beta1 delay is a value that is system dependent and usually does not require adjustment. It is programmed only once and measured in bit times.

45.3.36 UART CEA709.1-B Secondary Delay Timer High (UARTx_SDTH)

Addresses: UART0_SDTH is 4006_A000h base + 25h offset = 4006_A025h
UART1_SDTH is 4006_B000h base + 25h offset = 4006_B025h
UART2_SDTH is 4006_C000h base + 25h offset = 4006_C025h

Bit	7	6	5	4	3	2	1	0
Read Write	SDTH							
Reset	0	0	0	0	0	0	0	0

UARTx_SDTH field descriptions

Field	Description
7–0 SDTH	Secondary Delay Timer High This is the most significant byte of the secondary delay timer and is set by software. This is generally a variable value that must be set for each data message to be transmitted. It is measured in bit times, that is, the time that it takes for a single bit or one differential Manchester symbol to be transmitted. This value must be between 0 and (BL*Wbase) + (ProritySlots -1), Beta2 timeslots. A value of zero indicates that the queued packet will be sent immediately upon expiration of the beta1 timer.

45.3.37 UART CEA709.1-B Secondary Delay Timer Low (UARTx_SDTL)

Addresses: UART0_SDTL is 4006_A000h base + 26h offset = 4006_A026h
UART1_SDTL is 4006_B000h base + 26h offset = 4006_B026h
UART2_SDTL is 4006_C000h base + 26h offset = 4006_C026h

Bit	7	6	5	4	3	2	1	0
Read Write	SDTL							
Reset	0	0	0	0	0	0	0	0

UARTx_SDTL field descriptions

Field	Description
7–0 SDTL	Secondary Delay Timer Low This is the least significant byte of the secondary delay timer and is set by software. This is generally a variable value that must be set for each data message to be transmitted. It is measured in bit times, that is, the time that it takes for a single bit or one Differential Manchester symbol to be transmitted. This value must be between 0 and (BL*Wbase) + (ProritySlots -1), Beta2 timeslots. A value of zero indicates that the queued packet will be sent immediately upon expiration of the Beta1 timer.

45.3.38 UART CEA709.1-B Preamble (UARTx_PRE)

Addresses: UART0_PRE is 4006_A000h base + 27h offset = 4006_A027h
UART1_PRE is 4006_B000h base + 27h offset = 4006_B027h
UART2_PRE is 4006_C000h base + 27h offset = 4006_C027h

Bit	7	6	5	4	3	2	1	0
Read Write	PREAMBLE							
Reset	0	0	0	0	0	0	0	0

UARTx_PRE field descriptions

Field	Description
7–0 PREAMBLE	CEA709.1-B Preamble Register The number of bit-sync characters that occur prior to the byte-sync character when preamble is transmitted. **NOTE:** The minimum preamble length supported by twisted pair wire is four bit-sync fields.

45.3.39 UART CEA709.1-B Transmit Packet Length (UARTx_TPL)

Addresses: UART0_TPL is 4006_A000h base + 28h offset = 4006_A028h
UART1_TPL is 4006_B000h base + 28h offset = 4006_B028h
UART2_TPL is 4006_C000h base + 28h offset = 4006_C028h

Bit	7	6	5	4	3	2	1	0
Read Write	TPL							
Reset	0	0	0	0	0	0	0	0

UARTx_TPL field descriptions

Field	Description
7–0 TPL	Transmit Packet Length Register Length of the data packet in bytes that is transmitted by CEA709.1-B transmitter. This includes the CRC packet as well.

45.3.40 UART CEA709.1-B Interrupt Enable Register (UARTx_IE)

Addresses: UART0_IE is 4006_A000h base + 29h offset = 4006_A029h
UART1_IE is 4006_B000h base + 29h offset = 4006_B029h
UART2_IE is 4006_C000h base + 29h offset = 4006_C029h

Bit	7	6	5	4	3	2	1	0
Read	0	WBEIE	ISDIE	PRXIE	PTXIE	PCTEIE	PSIE	TXFIE
Write	▨							
Reset	0	0	0	0	0	0	0	0

UARTx_IE field descriptions

Field	Description
7 Reserved	This read-only field is reserved and always has the value zero.
6 WBEIE	WBASE Expired Interrupt Enable Interrupt enable for WBASE expired flag. 0 Interrupt is disabled. 1 Interrupt is enabled.
5 ISDIE	Initial Sync Detection Interrupt Enable Interrupt enable for initial synchronization detection flag. **NOTE:** This field cannot be cleared except by disabling CEA709. Therefore, ISDIE must be cleared when the first initial sync detection interrupt occurs. If the ISD interrupt is not disabled in the interrupt handler, then user will continuously get interrupts. 0 Interrupt is disabled. 1 Interrupt is enabled.
4 PRXIE	Packet Received Interrupt Enable Interrupt enable for packet received flag. 0 Interrupt is disabled. 1 Interrupt is enabled.
3 PTXIE	Packet Transmitted Interrupt Enable Interrupt enable for packet transmitted flag. 0 Interrupt is disabled. 1 Interrupt is enabled.
2 PCTEIE	Packet Cycle Timer Interrupt Enable Interrupt enable for packet cycle time expired flag. 0 Interrupt is disabled. 1 Interrupt is enabled.

Table continues on the next page...

UARTx_IE field descriptions (continued)

Field	Description
1 PSIE	Preamble Start Interrupt Enable Interrupt enable for preamble start flag. 0 Interrupt is disabled. 1 Interrupt is enabled.
0 TXFIE	Transmission Fail Interrupt Enable Interrupt enable for transmission fail flag. 0 Interrupt is disabled. 1 Interrupt is enabled.

45.3.41 UART CEA709.1-B WBASE (UARTx_WB)

Addresses: UART0_WB is 4006_A000h base + 2Ah offset = 4006_A02Ah
UART1_WB is 4006_B000h base + 2Ah offset = 4006_B02Ah
UART2_WB is 4006_C000h base + 2Ah offset = 4006_C02Ah

Bit	7	6	5	4	3	2	1	0
Read Write	WBASE							
Reset	0	0	0	0	0	0	0	0

UARTx_WB field descriptions

Field	Description
7–0 WBASE	CEA709.1-B WBASE register Size of the basic randomizing window in bit periods after Beta1 time period.

45.3.42 UART CEA709.1-B Status Register (UARTx_S3)

Addresses: UART0_S3 is 4006_A000h base + 2Bh offset = 4006_A02Bh
UART1_S3 is 4006_B000h base + 2Bh offset = 4006_B02Bh
UART2_S3 is 4006_C000h base + 2Bh offset = 4006_C02Bh

Bit	7	6	5	4	3	2	1	0
Read	PEF	WBEF	ISD	PRXF	PTXF	PCTEF	PSF	TXFF
Write			▨					
Reset	0	0	0	0	0	0	0	0

UARTx_S3 field descriptions

Field	Description
7 PEF	Preamble Error Flag Indicates that the received preamble has an error. If the received preamble length is greater than or less than the transmit preamble length, the preamble error flag is asserted. This flag is cleared by writing 1. 0 Preamble is correct. 1 Preamble has an error.
6 WBEF	Wbase Expired Flag Indicates that the Wbase time period has expired after beta1 time slots. This flag is cleared by writing 1. 0 WBASE time period has not expired. 1 WBASE time period has expired after beta1 time slots.
5 ISD	Initial Sync Detect Indicates that initially, a valid one and a line code violation is detected. This flag is cleared by deasserting EN709 bit. 0 Initial sync is not detected. 1 Initial sync is detected.
4 PRXF	Packet Received Flag Indicates that complete packet is received. This flag is cleared by writing 1. 0 Packet is not received. 1 Packet is received.
3 PTXF	Packet Transmitted Flag Indicates that complete packet is transmitted. This flag is cleared by writing 1. In case TX packet gets aborted due to FIFO becoming empty or an overflow, packet transmitted flag will still be generated. 0 Packet transmission is not complete. 1 Packet transmission is complete.
2 PCTEF	Packet Cycle Timer Expired Flag Indicates that packet cycle time period has expired with no activity on the line. This flag is cleared by writing 1. 0 Packet cycle time has not expired. 1 Packet cycle time has expired.
1 PSF	Preamble Start Flag Indicates start of the preamble while the packet is being transmitted. This flag is cleared by writing 1. 0 Preamble start is not detected. 1 Preamble start is detected.
0 TXFF	Transmission Fail Flag Indicates that transmission could not proceed. This flag is asserted when the packet is queued for transmission but before the random delay has expired and an incoming receive packet is detected. This flag is also asserted while transmission when the TX FIFO becomes empty or overflows. In these cases,

Table continues on the next page...

UARTx_S3 field descriptions (continued)

Field	Description
	line code violation is transmitted on TX line immediately after the current byte or preamble transmission is finished, without waiting for completion of transmit packet length. If the transmission fail flag is asserted, C6[TX709] is cleared. This flag is cleared by writing 1. 0 Transmission continues normally. 1 Transmission has failed.

45.3.43 UART CEA709.1-B Status Register (UARTx_S4)

Addresses: UART0_S4 is 4006_A000h base + 2Ch offset = 4006_A02Ch
UART1_S4 is 4006_B000h base + 2Ch offset = 4006_B02Ch
UART2_S4 is 4006_C000h base + 2Ch offset = 4006_C02Ch

Bit	7	6	5	4	3	2	1	0
Read	0			INITF	CDET		ILCV	FE
Write								
Reset	0	0	0	0	0	0	0	0

UARTx_S4 field descriptions

Field	Description
7–5 Reserved	This read-only field is reserved and always has the value zero.
4 INITF	Initial Synchronization Fail Flag Indicates that the initial synchronization has failed and the packet cycle time has expired after enabling EN709 register. This flag is cleared if EN709 is cleared. 0 Initial synchronization has not failed. 1 Initial synchronization has failed.
3–2 CDET	CDET Indicates when the collision occurs during transmission. This flag is cleared by writing 2'b11. If the collision flag is not cleared by software and a valid collision pulse is detected during some other phase of transmission, then collision flag continues to indicate the previous value. 00 No collision. 01 Collision occurred during preamble. 10 Collision occurred during data. 11 Collision occurred during line code violation.
1 ILCV	Improper Line Code Violation Indicates that line code violation received is not proper. This flag is cleared by writing 1. 0 Line code violation received is proper. 1 Line code violation received is improper, that is, less than three bit periods.

Table continues on the next page...

UARTx_S4 field descriptions (continued)

Field	Description
0 FE	Framing Error Indicates that the received CEA709.1-B packet has finished at byte boundary. This flag is cleared by writing 1. 0 Received packet is byte bound. 1 Received packet is not byte bound.

45.3.44 UART CEA709.1-B Received Packet Length (UARTx_RPL)

Addresses: UART0_RPL is 4006_A000h base + 2Dh offset = 4006_A02Dh
UART1_RPL is 4006_B000h base + 2Dh offset = 4006_B02Dh
UART2_RPL is 4006_C000h base + 2Dh offset = 4006_C02Dh

Bit	7	6	5	4	3	2	1	0
Read	RPL							
Write	▨							
Reset	0	0	0	0	0	0	0	0

UARTx_RPL field descriptions

Field	Description
7–0 RPL	Received Packet Length Indicates the length of received packet in bytes. If the received packet is not byte aligned, the partial byte received is appended by zeros.

45.3.45 UART CEA709.1-B Received Preamble Length (UARTx_RPREL)

Addresses: UART0_RPREL is 4006_A000h base + 2Eh offset = 4006_A02Eh
UART1_RPREL is 4006_B000h base + 2Eh offset = 4006_B02Eh
UART2_RPREL is 4006_C000h base + 2Eh offset = 4006_C02Eh

Bit	7	6	5	4	3	2	1	0
Read	RPREL							
Write	▨							
Reset	0	0	0	0	0	0	0	0

UARTx_RPREL field descriptions

Field	Description
7–0 RPREL	Received Preamble Length

UARTx_RPREL field descriptions (continued)

Field	Description
	Indicates the number of bit sync fields received in the preamble.

45.3.46 UART CEA709.1-B Collision Pulse Width (UARTx_CPW)

Addresses: UART0_CPW is 4006_A000h base + 2Fh offset = 4006_A02Fh
UART1_CPW is 4006_B000h base + 2Fh offset = 4006_B02Fh
UART2_CPW is 4006_C000h base + 2Fh offset = 4006_C02Fh

Bit	7	6	5	4	3	2	1	0
Read Write	CPW							
Reset	0	0	0	0	0	0	0	0

UARTx_CPW field descriptions

Field	Description
7–0 CPW	CEA709.1-B CPW register Indicates the width of valid collision pulse in terms of IPG clock cycles.

45.3.47 UART CEA709.1-B Receive Indeterminate Time (UARTx_RIDT)

Addresses: UART0_RIDT is 4006_A000h base + 30h offset = 4006_A030h
UART1_RIDT is 4006_B000h base + 30h offset = 4006_B030h
UART2_RIDT is 4006_C000h base + 30h offset = 4006_C030h

Bit	7	6	5	4	3	2	1	0
Read Write	RIDT							
Reset	0	0	0	0	0	0	0	0

UARTx_RIDT field descriptions

Field	Description
7–0 RIDT	CEA709.1-B Receive IDT register Indicates the indeterminate time period after reception during which any activity on RX line will be discarded. Indeterminate time period value should be less than Beta1 timer value.

45.3.48 UART CEA709.1-B Transmit Indeterminate Time (UARTx_TIDT)

Addresses: UART0_TIDT is 4006_A000h base + 31h offset = 4006_A031h
UART1_TIDT is 4006_B000h base + 31h offset = 4006_B031h
UART2_TIDT is 4006_C000h base + 31h offset = 4006_C031h

Bit	7	6	5	4	3	2	1	0
Read Write	TIDT							
Reset	0	0	0	0	0	0	0	0

UARTx_TIDT field descriptions

Field	Description
7–0 TIDT	CEA709.1-B Transmit IDT Register This register indicates the indeterminate time period after transmission during which any activity on TX line will be discarded. Indeterminate time period value should be less than Beta1 timer value.

45.4 Functional description

This section provides a complete functional description of the UART block.

The UART allows full duplex, asynchronous, NRZ serial communication between the CPU and remote devices, including other CPUs. The UART transmitter and receiver operate independently, although they use the same baud rate generator. The CPU monitors the status of the UART, writes the data to be transmitted, and processes received data.

45.4.1 CEA709.1-B

The UART provides support for CEA709.1-B, which is commonly used in building automation, home networking, including all key building automation subsystems such as heating, ventilating, airconditioning, lighting, security, fire detection, access control, and energy monitoring.

45.4.1.1 CEA709.1-B packet cycle

The following figure illustrates the frame format and Differential Manchester encoding. Differential Manchester encoding requires that each transmitted bit includes a clock transition at the start of the bit period. This allows synchronization with the receiver.

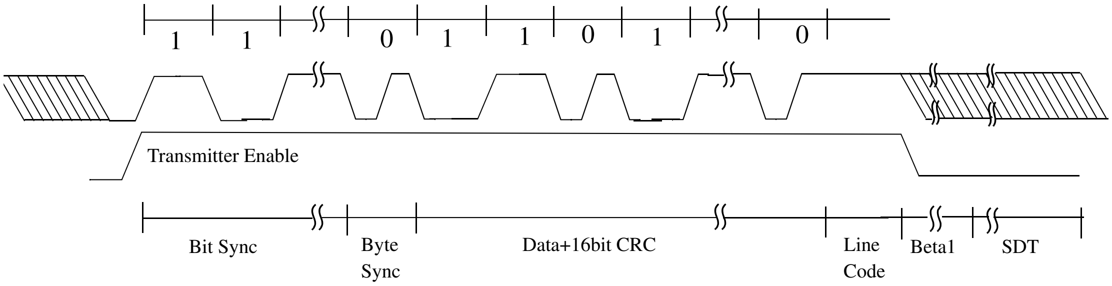


Figure 45-193. Frame format with differential manchester encoding

A logic zero is indicated with the presence of a transition in the middle of the bit period and a logic one is indicated by the absence of any transition. When transitions occur at the start of the bit time, polarity is arbitrary because the last bit of a transmission has no trailing clock edge. A transmitter will transmit a preamble at the beginning of a packet to allow other nodes to synchronize their receiver clocks. The preamble comprises a bit-sync field followed by a byte-sync field. The bit-sync field is a series of differential Manchester logic ones and the byte-sync field is a single differential Manchester logic zero. The byte-sync field marks the end of the preamble and the start of the data field (MPDU/LPDU).

The transmitter terminates the packet by forcing the data output to be transitionless long enough for the receiver to recognize an invalid bit code. This signals the end of the packet. At the end of the packet transmission, the line must remain transitionless for three bit periods after the final clock transition.

The UART is responsible for providing the BitSync and ByteSync fields of the PPDU illustrated below. The layer two software manages all other encapsulating fields and provides these to the UART as part of the packet to be transmitted.

Bit Sync	Byte Sync	Priority	Alt Path	Delta BL	NPDU	CRC

Figure 45-194. Physical protocol data unit structure

45.4.1.2 Packet cycle and delay calculations

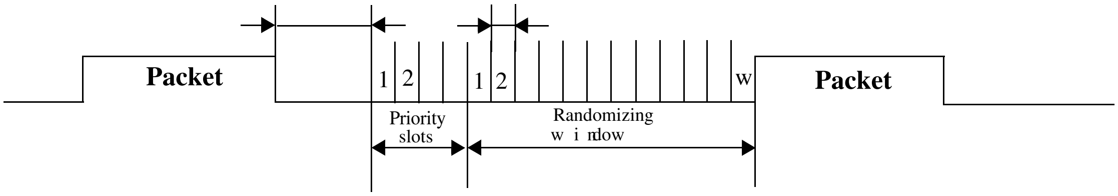


Figure 45-195. CEA709.1-B packet cycle

Predictive p-persistent CSMA is a technique for collision avoidance that randomizes channel access using knowledge of predicted load. It manages software using data and events reported by the hardware.

Beta1 delay is a value set by the software. It is generally a fixed value that is system dependent and hence does not usually require adjustment. It is measured in bit times, that is, the time that it takes for a single bit to be transmitted or one differential Manchester symbol. Beta1 is defined by CEA/EIA-709 specification as:

Beta1 > 1 bit time + $(2 \times T_{aup} + T_{aum})$

Where T_{aup} is the physical propagation delay defined by the media length.

T_{aum} is the detection and turn around-delay within the MAC sublayer; this is the period from the time the idle channel condition is detected, to the point when the first output transition appears on the output. On media where there is a carrier, this time must include the time between turning on the carrier, and it being asserted as a valid carrier on the medium.

The secondary delay timer is a value that is set by software. This is generally a variable value that must be set for each data message to be transmitted. It is measured in bit times, that is, the time that it takes for a single bit to be transmitted or one differential Manchester symbol. This value must be between 0 and (BL × Wbase) + (ProritySlots -1), Beta2 timeslots. A value of zero indicates that the queued packet for transmit is to be sent immediately upon expiration of the beta1 timer. According to the CEA/EIA-709 specification:

- BL is back log
- Wbase is 16 beta2 timeslots
- A priorityslot is the same amount of time as a beta2 timeslot
- Beta2 > $2 \times T_{aup} + T_{aum}$

Priority slots are handled completely by software. When calculating the secondary delay timer value, the software must take into account any priority slot that is included in the design of the system.

Each node must maintain an estimation of the current channel backlog. Backlog calculation is managed by the layer two software. Initially, the backlog is set to one. The backlog is incremented on transmission by a value indicated in the frames backlog increment field.

The backlog decrements under the following conditions:

- On waiting to transmit: If Wbase randomizing slots go by without channel activity.
- On receive: If a packet is received with a backlog increment of 0.
- On transmit: If a packet is transmitted with a backlog increment of 0.
- On idle: If a packet cycle time expires without channel activity.

The following actions need to be completed when a frame is received to prepare an outgoing message for transmission after the channel becomes idle:

- CRC of incoming message needs to be verified by software.
- If the CRC is good, the BL is recalculated, otherwise BL remains the same.
- Transmit delay (secondary delay timer) is calculated and supplied to UART.

45.4.1.3 Clock resynchronization

The UART is transmitting on time base source. Therefore, all receivers keep synchronizing with the node that is transmitting and no clock resynchronization occurs in transmitting.

When the UART is receiving or waiting to transmit, clock resynchronization is vital. Because long streams of data are possible (up to 229 bytes + headers), there exists significant potential for nodes to wander regarding time reference over the course of the message. Therefore, Differential Manchester Encoding (DME) is used. While DME requires twice the bandwidth of non-return to zero (NRZ) encoding schemes, it has the benefit of a guaranteed transition at the start of each bit transmitter. A transition occurring at the middle of the bit is encoded as a logic 0 or the lack of a transition at the middle of the bit time is encoded as a logic 1. By detecting the transition at the start of a bit period, the receiver is able to be resynthesized to the transmitter every bit period. Resynchronization can occur only after the node is already synchronized with the system. Additionally, for resynchronization to be effective, some basic assumptions regarding the system must be made:

1. Only a single channel sample may be in error (noise) over the entire bit (16 samples) period.
2. While a node is drifted from the system time base, with the resynchronization, the node is never shifted by more than 2 data samples in a given bit period.
3. If multiple noise events have occurred, no action is taken.

4. If a single noise event occurs, and it is possible to uniquely identify the noise event, then resynchronization takes place.

Starting at sample 15 of the previous time bit period, five data samples are collected. The number and location of the samples are key to decide if an adjustment in time base is required. Table below lists the possible values and the actions associated with each possibility. In the table, S means the data is the same as the logical value that was received in the second half of the previous bit period. D means that the sample is different from the logical value that was received in the second half of the previous bit period.

Sample Values (15,16,1,2,3)	Action / Event
SSSSS	No start of bit tarnation is detected. Therefore, no adjustment to time base is made.
SSSSD	Two or more error events occurred or the time base was off. In this case, the time base is slowed down by two. Sample 3 becomes sample 1. The next sample is treated as sample 2.
SSSDS	Two or more error events occurred, time base was off along with noise occurrence, or sample 2 is noise and there is no start of bit transition. Therefore, no adjustment to time base is made.
SSSDD	It is possible that either noise was received during sample 1 or the time base needs shifting. In this case, the time base is slowed down by one. Sample 2 becomes sample 1, and sample 3 becomes sample 2. The next sample is treated as sample 3.
SSDSS	It is most likely that sample 1 is noise and there is no start of bit transition. Therefore, no adjustment to time base is made.
SSDSD	It is possible that sample 1 is noise, and time base needs shifting by two, or that sample 2 is noise. It is more likely that sample 2 is noise and therefore no adjustment to time base is made.
SSDDS	It is most likely that sample 3 is noise. Therefore, no adjustment to time base is made.
SSDDD	This is the expected case. Therefore, no adjustment to time base is made.
SDSSS	It is most likely that sample 16 is noise and there is no start of bit transition. Therefore, no adjustment to time base is made.
SDSSD	Either multiple errors occurred or sample 16 is noise and time base is off by two. In this case, the time base is slowed down by two. Sample 3 becomes sample 1. The next sample is treated as sample 2.
SDSDS	In this case, multiple errors have occurred. Therefore, no adjustment to time base is made.
SDSDD	In this case, there must either be multiple noise or one noise at sample 16 or sample 1 with a time shift. Assuming that one noise occurred, it is unclear what direction the time shift is. Therefore, no adjustment to time base is made.

Table continues on the next page...

Sample Values (15,16,1,2,3)	Action / Event
SDDSS	In this case, either multiple errors occurred, two or more noise, or two or more noise and a time shift. The most likely case is that samples 16 and 1 are noise. Therefore, no adjustment to time base is made.
SDDSD	The most likely case is noise for sample 2 and a time shift. Therefore, the time base is sped up by one. Sample 16 becomes sample 1, sample 1 becomes sample 2, sample 2 becomes sample 3, sample 3 becomes sample 4, and the next sample taken is sample 5.
SDDDS	The most likely case is noise for sample 3 and a time shift. Therefore, the time base is sped up by one. Sample 16 becomes sample 1, sample 1 becomes sample 2, sample 2 becomes sample 3, sample 3 becomes sample 4, and the next sample taken is sample 5.
SDDDD	Either sample 16 is noise or the time base has shifted. In this case, it is assumed that a time shift has occurred. Therefore, the time base is sped up by one. Sample 15 becomes sample 1, sample 1 becomes sample 2, sample 2 becomes sample 3, sample 3 becomes sample 4, and the next sample taken is sample 5.
DSSSS	It is most likely that sample 16 is noise and there is no start of bit transition. Therefore, no adjustment to time base is made.
DSSSD	It is most likely that sample 15 is noise along with time shift. In this case, the time base is slowed down by two. Sample 3 becomes sample 1. The next sample is treated as sample 2.
DSSDS	In this case, multiple errors occurred. Therefore, no adjustment to time base is made
DSSDD	Either multiple errors occurred, possibly with time shift, or sample 15 is noise. In this case, the time base is slowed down by one. Sample 2 becomes sample 1, and sample 3 becomes sample 2. The next sample is treated as sample 3.
DSDSS	In this case, multiple errors occurred. Therefore, no adjustment to time base is made.
DSDSD	In this case, multiple errors occurred. Therefore, no adjustment to time base is made.
DSDDS	In this case, multiple errors occurred. Therefore, no adjustment to time base is made.
DSDDD	In this case, either multiple errors occurred or sample 15 is noise and there is no start of bit transition. Therefore, no adjustment to time base is made.
DDSSS	In this case multiple errors occurred. It is most likely that samples 15 and 16 are noise. Therefore, no adjustment to time base is made.
DDSSD	In this case multiple errors occurred. Therefore, no adjustment to time base is made.
DDSDS	In this case multiple errors occurred. Therefore, no adjustment to time base is made.

Table continues on the next page...

Sample Values (15,16,1,2,3)	Action / Event
DDSDD	It is most likely that sample 1 is noise. Therefore, the time base is sped up by two. Sample 15 becomes sample 1, sample 16 becomes sample 2, sample 1 becomes sample 3, sample 2 becomes sample 4, sample 3 becomes sample 5, and the next sample taken is sample 6.
DDDSS	In this case multiple errors occurred along with time shift. Therefore, no adjustment to time base is made.
DDDSD	It is most likely that sample 2 is noise. Therefore, the time base is sped up by two. Sample 15 becomes sample 1, sample 16 becomes sample 2, sample 1 becomes sample 3, sample 2 becomes sample 4, sample 3 becomes sample 5, and the next sample taken is sample 6.
DDDDS	It is most likely that sample 3 is noise. Therefore, the time base is sped up by two. Sample 15 becomes sample 1, sample 16 becomes sample 2, sample 1 becomes sample 3, sample 2 becomes sample 4, sample 3 becomes sample 5, and the next sample taken is sample 6.
DDDDD	Either samples 15 and 16 are noise or the time base has shifted. Therefore, the time base is sped up by two. Sample 15 becomes sample 1, sample 16 becomes sample 2, sample 1 becomes sample 3, sample 2 becomes sample 4, sample 3 becomes sample 5, and the next sample taken is sample 6.

45.4.1.4 Data sampling

The receiver samples the unsynchronized receiver input signal at the RT clock rate. The RT clock is an internal signal with a frequency 16 times the baud rate. To adjust for baud rate mismatch, the RT clock is resynchronized after every bit.

To locate the start of preamble, data recovery logic does an asynchronous search for a logic 0 preceded by three logic 1s or logic 1 preceded by three logic 0s. When the falling edge or rising edge of a possible preamble bit occurs, the RT clock begins to count to 16.

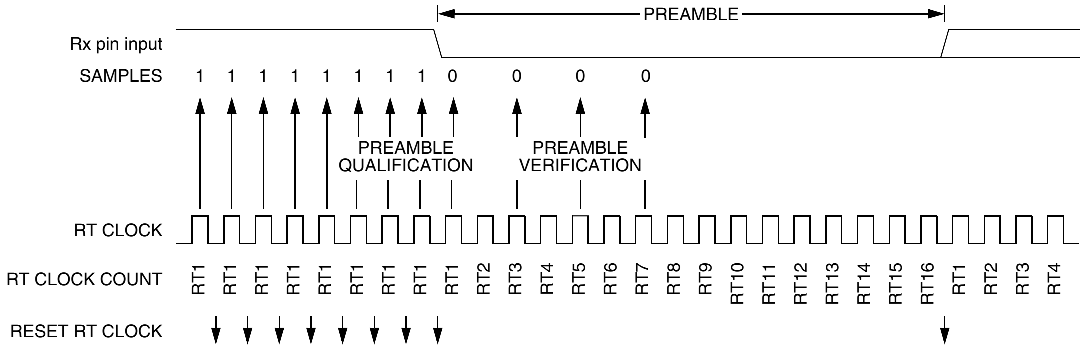

PREAMBLE
Rx pin input
SAMPLES
1 1 1 1 1 1 1 1 0 0 0 0
PREAMBLE QUALIFICATION
PREAMBLE VERIFICATION
RT CLOCK
RT CLOCK COUNT
RT1 RT1 RT1 RT1 RT1 RT1 RT1 RT1 RT1 RT2 RT3 RT4 RT5 RT6 RT7 RT8 RT9 RT10 RT11 RT12 RT13 RT14 RT15 RT16 RT1 RT2 RT3 RT4
RESET RT CLOCK

Figure 45-196. Receiver data sampling

To verify the start bit and to detect noise, data recovery logic takes samples at RT3, RT5, and RT7. The following table summarizes the results of the preamble verification samples.

Table 45-200. Preamble/ Data bit verification

RT3, RT5, and RT7 samples	Preamble verification
000	Yes
001	Yes
010	Yes
011	No
100	Yes
101	No
110	No
111	No

If preamble verification is not successful, the RT clock is reset and a new search for a preamble begins.

To determine the value of a data bit, recovery logic takes samples at RT11, RT12, and RT13. The following table summarizes the results of the data bit samples. If the majority of RT11, RT12, and RT13 samples is the same as the majority of RT3, RT5, and RT7 samples, then the data bit detected is 1, else the data bit detected is 0.

Table 45-201. Data bit recovery

RT11, RT12, and RT13 samples	Data bit determination
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

To signify the end of a data packet, the transmitter causes a line-code violation to occur, that is, the transmitter remains transitionless for at least 3-bit periods after the final clock transition, excluding the final data transition, if it exists. The receiver detects this violation. For the purpose of detecting a line-code violation, the receiver monitors the channel to locate a series of five or six back-to-back half bit periods.

45.4.1.5 Initial clock synchronization

When operating with EN709 set, there are various times when initial clock synchronization is required. When the UART has just been enabled, there is clearly no system clock reference. Additionally, if a channel has remained idle for a significant period of time, such as the arbitration time between packets, substantial clock drift may have occurred in the system between nodes. This is because there have been meaningful clock transitions on the channel to keep nodes synchronized. After these events, the clock may require significant synchronization adjustment; this event is referred to as initial clock synchronization.

There are three situations that may occur when a node attempts to obtain initial clock synchronization.

1. The node enters the system while a data packet is being actively transmitted.
2. The node enters the system while there is no data packet being actively transmitted on the system.
3. The node is already in the system and initial clock synchronization is required due to the end of a packet.

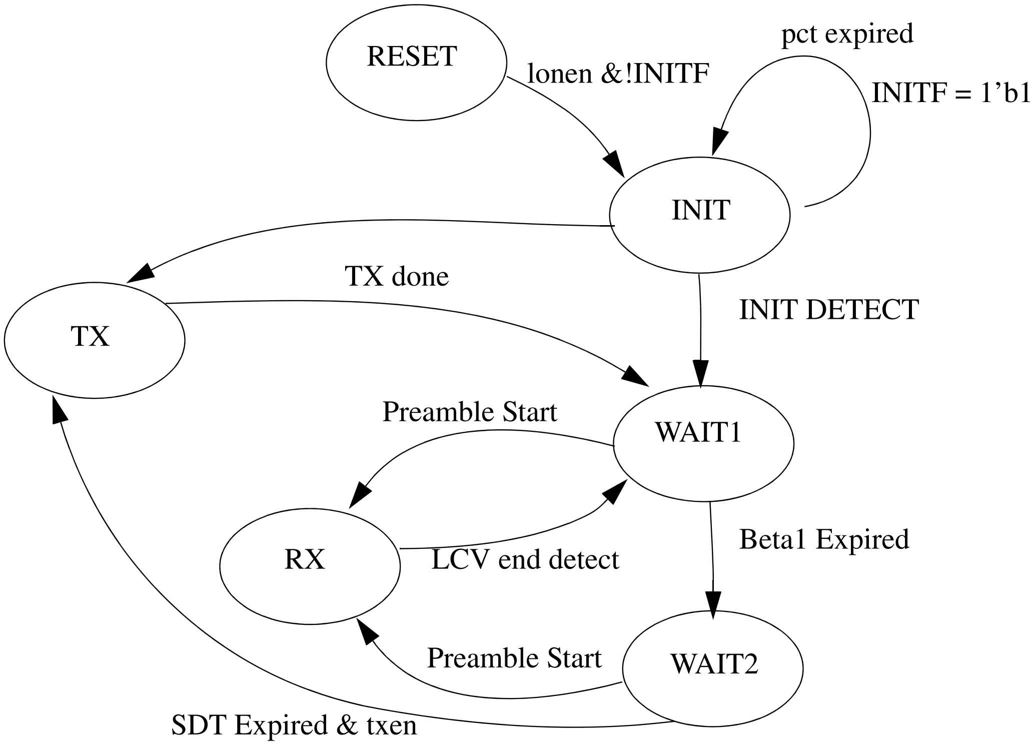
For case 1 and 2, the UART implements the following procedure:

1. The UART attempts to identify a valid edge to synchronize with.
2. While the UART attempts to locate a valid edge, it also tries to identify a line code violation of 8 back-to-back half bit time samples rather than the 6. It is not required to finish the current bit because the clock is not synchronized. If the required line-code violation is detected, the beta1 delay timer will start and the UART will transit to case 3.
3. If an edge is determined to be valid, that node will consider itself synchronized but will not start receiving, or attempt to send data, until a line code violation has been identified.
4. If no valid edge is determined and meanwhile the packet cycle timer expires, it is indicated to the processor that initial synchronization has failed and the processor can choose to transmit the data.

For case 3, it implements the following procedure:

1. Beta1 delay and secondary delay times increment as appropriate, that is Beta1 delay expires before the secondary delay timer starts.
2. While the timers are counting, the UART attempts to identify a valid edge.
3. If a valid edge is identified before the time expires, and data is queued to be transmitted, the transmission failure asserts and the clock is considered synchronized. The incoming data packet is received.

4. If a valid edge is not identified before the delay time expires, and data is queued to be transmitted, the UART considers itself synchronized, and starts the preamble process.
5. If a valid edge is not identified before the delay time expires, and data is not queued to be transmitted, the UART continues attempting to locate a valid edge using the same process, and receives the incoming data packet like in step 3.



45.4.1.6 Priority packet preemption

The first data is fetched from the data buffer immediately after the preamble has completed. Therefore, it is possible to decide which data is sent during transmission until the completion of the preamble. This can be done in two different ways.

- The expected data to be transmitted can be written to the data buffer before or shortly after TE is enabled. In this case, the data is ready before the start of the preamble period. If a high priority packet has been identified for immediate/preemptive transmission, software may flush the data buffer and put the new data into the data buffer. This new data must be put into the data buffer prior to the completion of the preamble. Similarly, the transmit packet length register needs to be updated.
- Software can trigger data to be transmitted by asserting TE before the actual data has been placed in the data buffer. In the end, the software can write data into the data buffer and update the transmit packet length register. This occurs before the preamble completes. To assist in identifying how much time is left before the

preamble completes, the preamble started interrupt is asserted when the UART starts transmitting the preamble.

NOTE

If the data buffer does not contain at least one byte of valid data and the transmit packet length register has been updated prior to the preamble completing, an underflow event will occur and TXEN is deasserted. The packet is terminated by transmitting line code violation.

45.4.1.7 Collision detection

Collision flag is detected only when device is transmitting if C6[CE] is asserted. The collision pulse is valid if it is asserted for CPW number of ipg clock cycles. If the collision signal is already asserted before the start of packet transmission, then the width of the collision pulse is calculated from the start of transmit packet as shown in the figure below. If the collision signal is not cleared by the software by writing 11b, then the flag continues to retain the previous value. After the flag is cleared, the collision pulse width is calculated again, and the flag is asserted, if the width is equal to or more than the programmed CPW value.

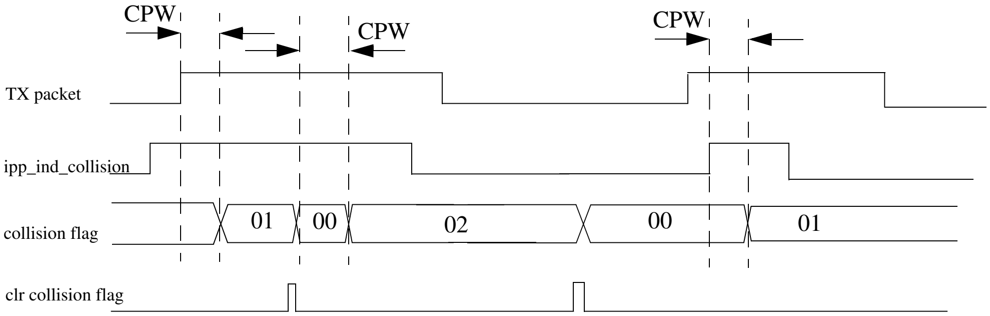


Figure 45-197. Collision pulse detection

The collision signal is asynchronous to the ipg clk, therefore the collision pulse of width exactly equal to CPW may not be detected correctly due to synchronization issue. The collision pulse visible to design may be decreased by one ipg clk cycle due to the asynchronous nature of the collision pulse.

45.4.2 Transmitter

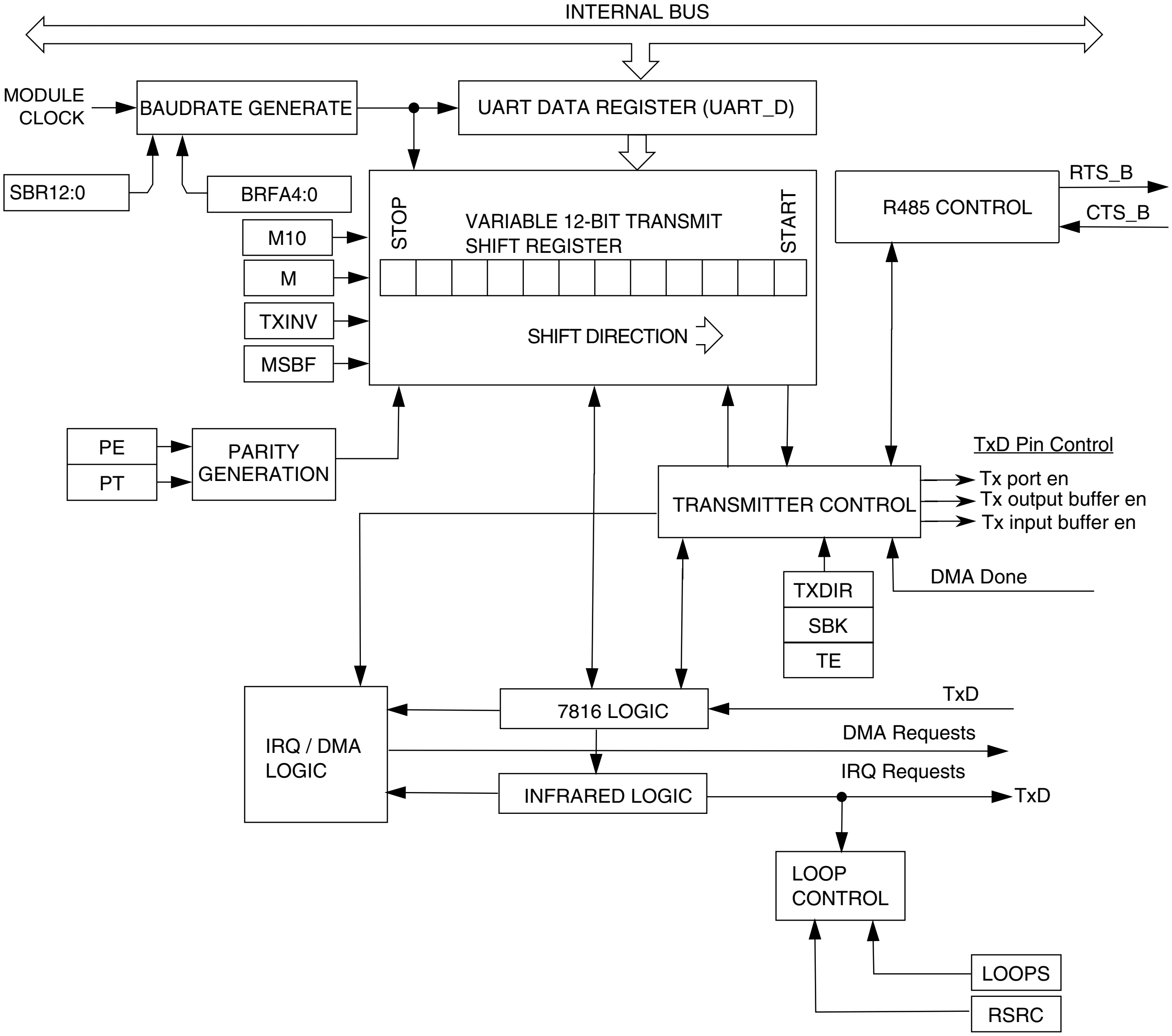

INTERNAL BUS
MODULE CLOCK
BAUDRATE GENERATE
UART DATA REGISTER (UART_D)
SBR12:0
BRFA4:0
M10
M
TXINV
MSBF
STOP
VARIABLE 12-BIT TRANSMIT SHIFT REGISTER
START
SHIFT DIRECTION
R485 CONTROL
RTS_B
CTS_B
PE
PT
PARITY GENERATION
TxD Pin Control
TRANSMITTER CONTROL
Tx port en
Tx output buffer en
Tx input buffer en
TXDIR
SBK
TE
DMA Done
IRQ / DMA LOGIC
7816 LOGIC
TxD
DMA Requests
IRQ Requests
INFRARED LOGIC
TxD
LOOP CONTROL
LOOPS
RSRC

Figure 45-198. Transmitter Block Diagram

45.4.2.1 Transmitter character length

The UART transmitter can accommodate either 8, 9, or 10-bit data characters. The state of the C1[M] and C1[PE] bits and the C4[M10] bit determine the length of data characters. When transmitting 9-bit data, bit C3[T8] is the ninth bit (bit 8).

45.4.2.2 Transmission bit order

When S2[MSBF] is set, the UART automatically transmits the MSB of the data word as the first bit after the start bit. Similarly, the LSB of the data word is transmitted immediately preceding the parity bit, or the stop bit if parity is not enabled. All necessary bit ordering is handled automatically by the module. Therefore, the format of the data written to D for transmission is completely independent of the S2[MSBF] setting.

45.4.2.3 Character transmission

To transmit data, the MCU writes the data bits to the UART transmit buffer using UART data registers C3[T8] and D. Data in the transmit buffer is then transferred to the transmitter shift register as needed. The transmit shift register then shifts a frame out through the transmit data output signal after it has prefaced it with any required start and stop bits. The UART data registers, C3[T8] and D, provide access to the transmit buffer structure.

The UART also sets a flag, the transmit data register empty flag S1[TDRE], and generates an interrupt or DMA request (C5[TDMAS]) whenever the number of datawords in the transmit buffer is equal to or less than the value indicated by TWFIFO[TXWATER]. The transmit driver routine may respond to this flag by writing additional datawords to the transmit buffer using C3[T8]/D as space permits.

See Application information for specific programing sequences.

Setting C2[TE] automatically loads the transmit shift register with the following preamble:

- 10 logic 1s if C1[M] = 0
- 11 logic 1s if C1[M] = 1 and C4[M10] = 0
- 12 logic 1s if C1[M] = 1, C4[M10] = 1, C1[PE] = 1

After the preamble shifts out, control logic transfers the data from the D register into the transmit shift register. The transmitter automatically transmits the correct start bit and stop bit before and after the dataword.

When C7816[ISO_7816E] = 1, setting C2[TE] does not result in a preamble being generated. The transmitter starts transmitting as soon as the corresponding guard time expires. When C7816[TTYPE] = 0, the value in GT is used. When C7816[TTYPE] = 1, the value in BGT is used, because C2[TE] will remain asserted until the end of the block transfer. C2[TE] is automatically cleared when C7816[TTYPE] = 1 and the block being transmitted has completed. When C7816[TTYPE] = 0, the transmitter listens for a NACK indication. If no NACK is received, it is assumed that the character was correctly received. If a NACK is received, the transmitter resends the data, assuming that the number of retries for that character, that is, the number of NACKs received, is less than or equal to the value in ET7816[TXTHRESHOLD].

Hardware supports odd or even parity. When parity is enabled, the bit immediately preceding the stop bit is the parity bit.

When the transmit shift register is not transmitting a frame, the transmit data output signal goes to the idle condition, logic 1. If at any time software clears C2[TE], the transmitter enable signal goes low and the transmit signal goes idle.

If the software clears C2[TE] while a transmission is in progress, the character in the transmit shift register continues to shift out, provided S1[TC] was cleared during the data write sequence. To clear S1[TC], the S1 register must be read followed by a write to D register.

If S1[TC] is cleared during character transmission and C2[TE] is cleared, the transmission enable signal is deasserted at the completion of the current frame. Following this, the transmit data out signal enters the idle state even if there is data pending in the UART transmit data buffer. To ensure that all the data written in the FIFO is transmitted on the link before clearing C2[TE], wait for S1[TC] to set. Alternatively, the same can be achieved by setting TWFIFO[TXWATER] to 0x0 and waiting for S1[TDRE] to set.

45.4.2.4 Transmitting break characters

Setting C2[SBK] loads the transmit shift register with a break character. A break character contains all logic 0s and has no start, stop, or parity bit. Break character length depends on C1[M] and C1[PE], S2[BRK13], and C4[M10]. See the following table.

Table 45-202. Transmit break character length

S2[BRK13]	C1[M]	C4[M10]	C1[PE]	Bits transmitted
0	0	—	—	10
0	1	0	—	11
0	1	1	0	11
0	1	1	1	12
1	0	—	—	13
1	1	—	—	14

As long as C2[SBK] is set, the transmitter logic continuously loads break characters into the transmit shift register. After the software clears C2[SBK], the shift register finishes transmitting the last break character and then transmits at least one logic 1. The automatic logic 1 at the end of a break character guarantees the recognition of the start bit of the next character. Break bits are not supported when C7816[ISO_7816E] is set/enabled.

NOTE

When queuing a break character, it will be transmitted following the completion of the data value currently being shifted out from the shift register. This means that, if data is queued in the data buffer to be transmitted, the break character preempts that queued data. The queued data is then transmitted after the break character is complete.

45.4.2.5 Idle characters

An idle character contains all logic 1s and has no start, stop, or parity bit. Idle character length depends on C1[M], C1[PE], and C4[M10]. The preamble is a synchronizing idle character that begins the first transmission initiated after setting C2[TE]. When C7816[ISO_7816E] is set/enabled, idle characters are not sent or detected. When data is not being transmitted, the data I/O line is in an inactive state.

If C2[TE] is cleared during a transmission, the transmit data output signal becomes idle after completion of the transmission in progress. Clearing and then setting C2[TE] during a transmission queues an idle character to be sent after the dataword currently being transmitted.

Note

When queuing an idle character, the idle character will be transmitted following the completion of the data value currently being shifted out from the shift register. This means that if data is queued in the data buffer to be transmitted, the idle character preempts that queued data. The queued data is then transmitted after the idle character is complete.

If C2[TE] is cleared and the transmission is completed, the UART is not the master of the TXD pin.

45.4.2.6 Hardware flow control

The transmitter supports hardware flow control by gating the transmission with the value of $\overline{\text{CTS}}$. If the clear-to-send operation is enabled, the character is transmitted when $\overline{\text{CTS}}$ is asserted. If $\overline{\text{CTS}}$ is deasserted in the middle of a transmission with characters remaining in the receiver data buffer, the character in the shift register is sent and TXD remains in the mark state until $\overline{\text{CTS}}$ is reasserted.

If the clear-to-send operation is disabled, the transmitter ignores the state of $\overline{\text{CTS}}$. Also, if the transmitter is forced to send a continuous low condition because it is sending a break character, the transmitter ignores the state of $\overline{\text{CTS}}$ regardless of whether the clear-to-send operation is enabled.

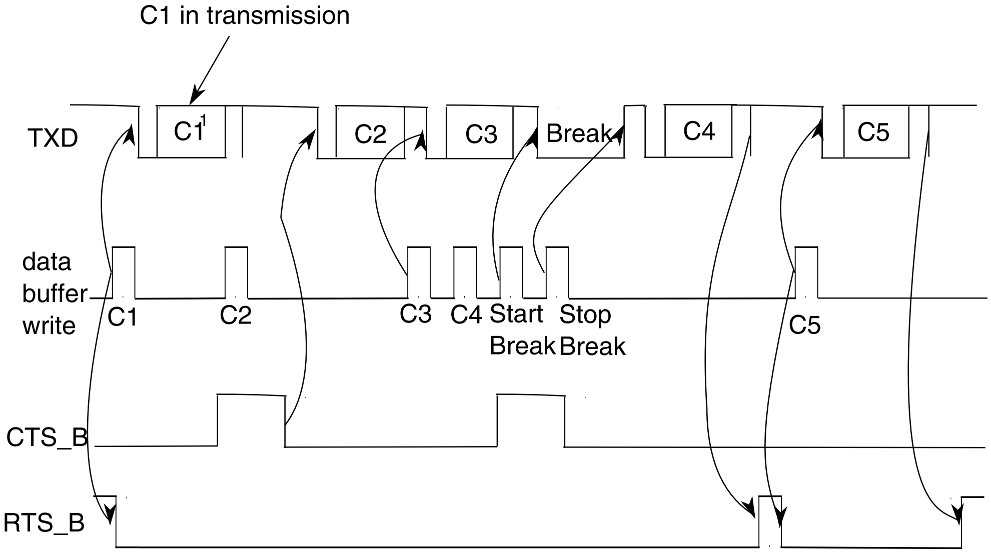
The transmitter's $\overline{\text{CTS}}$ signal can also be enabled even if the same UART receiver's $\overline{\text{RTS}}$ signal is disabled.

45.4.2.7 Transceiver driver enable

The transmitter can use $\overline{\text{RTS}}$ as an enable signal for the driver of an external transceiver. See Transceiver driver enable using $\overline{\text{RTS}}$ for details. If the request-to-send operation is enabled, when a character is placed into an empty transmitter data buffer, $\overline{\text{RTS}}$ asserts one bit time before the start bit is transmitted. $\overline{\text{RTS}}$ remains asserted for the whole time that the transmitter data buffer has any characters. $\overline{\text{RTS}}$ deasserts one bit time after all characters in the transmitter data buffer and shift register are completely sent, including the last stop bit. Transmitting a break character also asserts $\overline{\text{RTS}}$, with the same assertion and deassertion timing as having a character in the transmitter data buffer.

The transmitter's $\overline{\text{RTS}}$ signal asserts only when the transmitter is enabled. However, the transmitter's $\overline{\text{RTS}}$ signal is unaffected by its $\overline{\text{CTS}}$ signal. $\overline{\text{RTS}}$ will remain asserted until the transfer is completed, even if the transmitter is disabled mid-way through a data transfer.

The following figure shows the functional timing information for the transmitter. Along with the actual character itself, TXD shows the start bit. The stop bit is also indicated, with a dashed line if necessary.


C1 in transmission
TXD
C1[1]
C2
C3
Break
C4
C5
data buffer write
C1
C2
C3
C4
Start Break
Stop Break
C5
CTS_B
RTS_B

1. Cn = transmit characters

Figure 45-199. Transmitter RTS and CTS timing diagram

45.4.3 Receiver

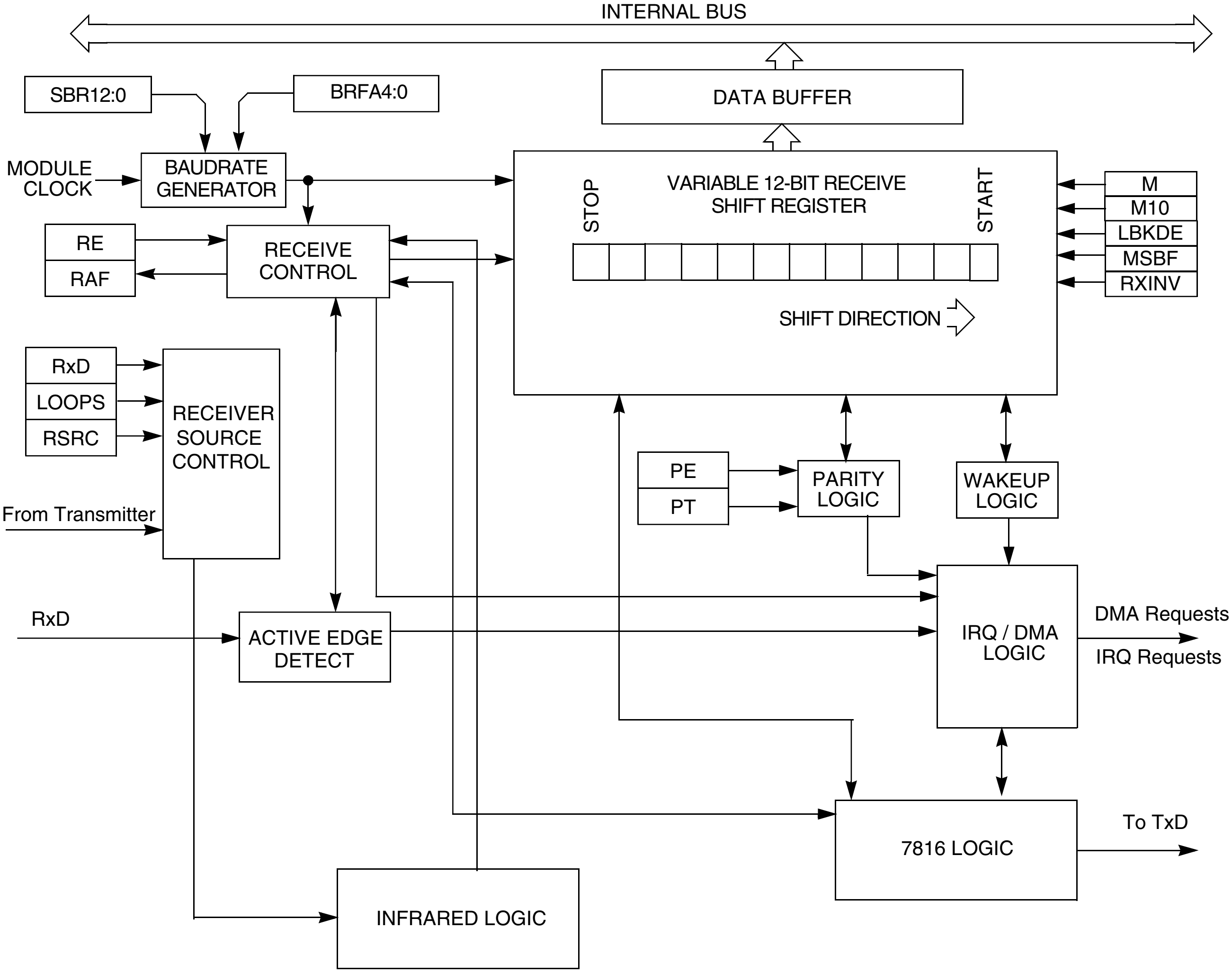

INTERNAL BUS
SBR12:0
BRFA4:0
DATA BUFFER
MODULE CLOCK
BAUDRATE GENERATOR
STOP
VARIABLE 12-BIT RECEIVE SHIFT REGISTER
START
M
M10
LBKDE
MSBF
RXINV
RE
RAF
RECEIVE CONTROL
SHIFT DIRECTION
RxD
LOOPS
RSRC
RECEIVER SOURCE CONTROL
From Transmitter
PE
PT
PARITY LOGIC
WAKEUP LOGIC
RxD
ACTIVE EDGE DETECT
IRQ / DMA LOGIC
DMA Requests
IRQ Requests
7816 LOGIC
To TxD
INFRARED LOGIC

Figure 45-200. UART receiver block diagram

45.4.3.1 Receiver character length

The UART receiver can accommodate 8-, 9-, or 10-bit data characters. The states of C1[M], C1[PE], and C4[M10] determine the length of data characters. When receiving 9 or 10-bit data, C3[R8] is the ninth bit (bit 8).

45.4.3.2 Receiver bit ordering

When S2[MSBF] is set, the receiver operates such that the first bit received after the start bit is the MSB of the dataword. Similarly, the bit received immediately preceding the parity bit, or the stop bit if parity is not enabled, is treated as the LSB for the dataword. All necessary bit ordering is handled automatically by the module. Therefore, the format of the data read from receive data buffer is completely independent of S2[MSBF].

45.4.3.3 Character reception

During UART reception, the receive shift register shifts a frame in from the unsynchronized receiver input signal. After a complete frame shifts into the receive shift register, the data portion of the frame transfers to the UART receive buffer. Additionally, the noise and parity error flags that are calculated during the receive process are also captured in the UART receive buffer. The receive data buffer is accessible via the D and C3[T8] registers. Additional received information flags regarding the receive dataword can be read in ED register. S1[RDRF] is set if the number of resulting datawords in the receive buffer is equal to or greater than the number indicated by RWFIFO[RXWATER]. If the C2[RIE] is also set, RDRF generates an RDRF interrupt request. Alternatively, by programming C5[RDMAS] correctly, a DMA request can be generated.

When C7816[ISO_7816E] is set/enabled and C7816[TTYPE] = 0, character reception operates slightly differently. Upon receipt of the parity bit, the validity of the parity bit is checked. If C7816[ANACK] is set and the parity check fails, or if INIT and the received character is not a valid initial character, then a NACK is sent by the receiver. If the number of consecutive receive errors exceeds the threshold set by ET7816[RXTHRESHOLD], then IS7816[RXT] is set and an interrupt generated if IE7816[RXTE] is set. If an error is detected due to parity or an invalid initial character, the data is not transferred from the receive shift register to the receive buffer. Instead, the data is overwritten by the next incoming data.

When the C7816[ISO_7816E] is set/enabled, C7816[ONACK] is set/enabled, and the received character results in the receive buffer overflowing, a NACK is issued by the receiver. Additionally, S1[OR] is set and an interrupt is issued if required, and the data in the shift register is discarded.

45.4.3.4 Framing errors

If the data recovery logic does not detect a logic 1 where the stop bit should be in an incoming frame, it sets the framing error flag, S1[FE], if S2[LBKDE] is disabled. When S2[LBKDE] is disabled, a break character also sets the S1[FE] because a break character has no stop bit. S1[FE] is set at the same time that received data is placed in the receive data buffer. Framing errors are not supported when C7816[ISO7816E] is set/enabled. However, if S1[FE] is set, data will not be received when C7816[ISO7816E] is set.

45.4.3.5 Receiving break characters

The UART recognizes a break character when a start bit is followed by eight, nine, or ten logic 0 data bits and a logic 0 where the stop bit should be. Receiving a break character has these effects on UART registers:

- Sets the framing error flag, S1[FE].
- Writes an all 0 dataword to the data buffer, which may cause S1[RDRF] to set, depending on the watermark and number of values in the data buffer.
- May set the overrun flag, S1[OR], noise flag, S1[NF], parity error flag, S1[PE], or the receiver active flag, S2[RAF].

The detection threshold for a break character can be adjusted when using an internal oscillator in a LIN system by setting S2[LBKDE]. The UART break character detection threshold depends on C1[M], C1[PE], C4[LBKDE], and C4[M10]. See the following table.

Table 45-203. Receive break character detection threshold

LBKDE	M	M10	PE	Threshold (bits)
0	0	—	—	10
0	1	0	—	11
0	1	1	0	11
0	1	1	1	12
1	0	—	—	11
1	1	—	—	12

While C4[LBKDE] is set, it will have these effects on the UART registers:

- Prevents S1[RDRF], S1[FE], S1[NF], and S1[PF] from being set. However, if they are already set, they will remain set.
- Sets the LIN break detect interrupt flag, S2[LBKDIF], if a LIN break character is received.

Break characters are not detected or supported when C7816[ISO_7816E] is set/enabled.

45.4.3.6 Hardware flow control

To support hardware flow control, the receiver can be programmed to automatically deassert and assert $\overline{\text{RTS}}$.

- $\overline{\text{RTS}}$ remains asserted until the transfer is complete, even if the transmitter is disabled midway through a data transfer. See Transceiver driver enable using $\overline{\text{RTS}}$ for more details.
- If the receiver request-to-send functionality is enabled, the receiver automatically deasserts $\overline{\text{RTS}}$ if the number of characters in the receiver data register is equal to or greater than receiver data buffer's watermark, RWFIFO[RXWATER].
- The receiver asserts $\overline{\text{RTS}}$ when the number of characters in the receiver data register is less than the watermark. It is not affected if RDRF is asserted.
- Even if $\overline{\text{RTS}}$ is deasserted, the receiver continues to receive characters until the receiver data buffer is full or is overrun.
- If the receiver request-to-send functionality is disabled, the receiver $\overline{\text{RTS}}$ remains deasserted.

The following figure shows receiver hardware flow control functional timing. Along with the actual character itself, RXD shows the start bit. The stop bit can also indicated, with a dashed line, if necessary. The watermark is set to 2.

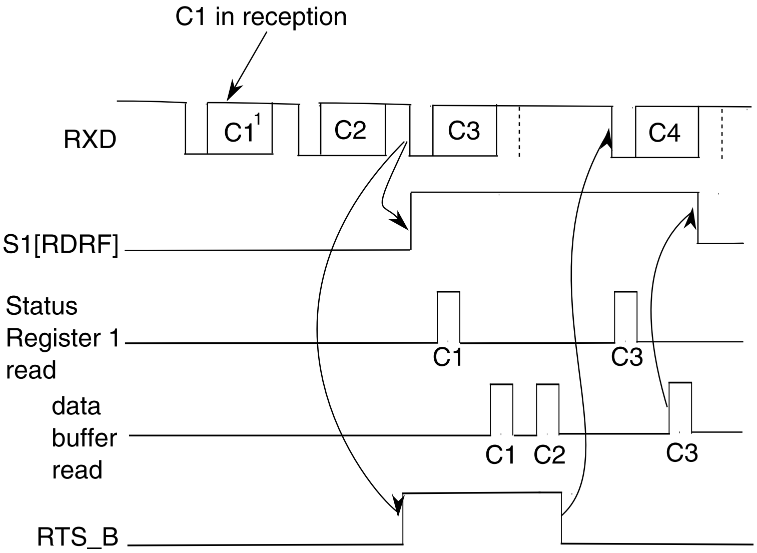


Figure 45-201. Receiver hardware flow control timing diagram

45.4.3.7 Infrared decoder

The infrared decoder converts the received character from the IrDA format to the NRZ format used by the receiver. It also has a 16-RT clock counter that filters noise and indicates when a 1 is received.

45.4.3.7.1 Start bit detection

When S2[RXINV] is cleared, the first rising edge of the received character corresponds to the start bit. The infrared decoder resets its counter. At this time, the receiver also begins its start bit detection process. After the start bit is detected, the receiver synchronizes its bit times to this start bit time. For the rest of the character reception, the infrared decoder's counter and the receiver's bit time counter count independently from each other.

45.4.3.7.2 Noise filtering

Any further rising edges detected during the first half of the infrared decoder counter are ignored by the decoder. Any pulses less than one RT clocks can be undetected by it regardless of whether it is seen in the first or second half of the count.

45.4.3.7.3 Low-bit detection

During the second half of the decoder count, a rising edge is decoded as a 0, which is sent to the receiver. The decoder counter is also reset.

45.4.3.7.4 High-bit detection

At 16-RT clocks after the previous rising edge, if a rising edge is not seen, then the decoder sends a 1 to the receiver.

If the next bit is a 0, which arrives late, then a low-bit is detected according to Low-bit detection. The value sent to the receiver is changed from 1 to a 0. Then, if a noise pulse occurs outside the receiver's bit time sampling period, then the delay of a 0 is not recorded as noise.

45.4.3.8 Baud rate tolerance

A transmitting device may be operating at a baud rate below or above the receiver baud rate. Accumulated bit time misalignment can cause one of the three stop bit data samples (RT8, RT9, and RT10) to fall outside the actual stop bit. A noise error will occur if the

RT8, RT9, and RT10 samples are not all the same logical values. A framing error will occur if the receiver clock is misaligned in such a way that the majority of the RT8, RT9, and RT10 stop bit samples are a logic 0.

As the receiver samples an incoming frame, it resynchronizes the RT clock on any valid falling edge within the frame. Resynchronization within frames corrects a misalignment between transmitter bit times and receiver bit times.

45.4.3.8.1 Slow data tolerance

The following figure shows how much a slow received frame can be misaligned without causing a noise error or a framing error. The slow stop bit begins at RT8 instead of RT1 but arrives in time for the stop bit data samples at RT8, RT9, and RT10.

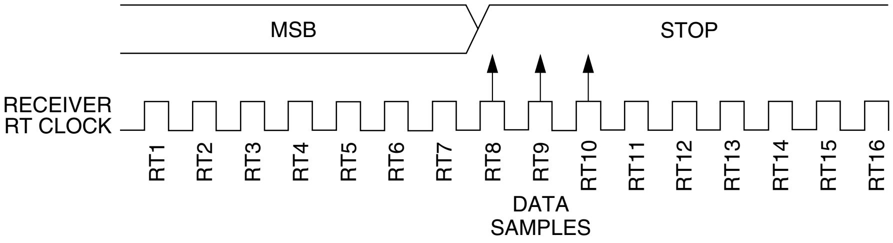


Figure 45-202. Slow data

For an 8-bit data character, data sampling of the stop bit takes the receiver 154 RT cycles (9 bit times × 16 RT cycles + 10 RT cycles).

With the misaligned character shown in the Figure 45-202, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 147 RT cycles (9 bit times × 16 RT cycles + 3 RT cycles).

The maximum percent difference between the receiver count and the transmitter count of a slow 8-bit data character with no errors is:

$$((154 - 147) \div 154) \times 100 = 4.54\%$$

For a 9-bit data character, data sampling of the stop bit takes the receiver 170 RT cycles (10 bit times × 16 RT cycles + 10 RT cycles).

With the misaligned character shown in the Figure 45-202, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 163 RT cycles (10 bit times × 16 RT cycles + 3 RT cycles).

The maximum percent difference between the receiver count and the transmitter count of a slow 9-bit character with no errors is:

$$((170 - 163) \div 170) \times 100 = 4.12\%$$

45.4.3.8.2 Fast data tolerance

The following figure shows how much a fast received frame can be misaligned. The fast stop bit ends at RT10 instead of RT16 but is still sampled at RT8, RT9, and RT10.

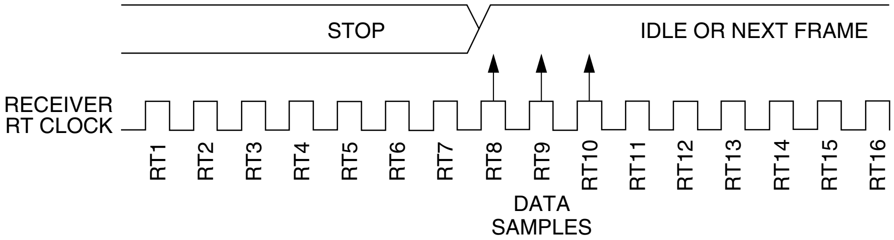


Figure 45-203. Fast data

For an 8-bit data character, data sampling of the stop bit takes the receiver 154 RT cycles (9 bit times × 16 RT cycles + 10 RT cycles).

With the misaligned character shown in the Figure 45-203, the receiver counts 154 RT cycles at the point when the count of the transmitting device is 160 RT cycles (10 bit times × 16 RT cycles).

The maximum percent difference between the receiver count and the transmitter count of a fast 8-bit character with no errors is:

$$((154 - 160) \div 154) \times 100 = 3.90\%$$

For a 9-bit data character, data sampling of the stop bit takes the receiver 170 RT cycles (10 bit times × 16 RT cycles + 10 RT cycles).

With the misaligned character shown in the Figure 45-203, the receiver counts 170 RT cycles at the point when the count of the transmitting device is 176 RT cycles (11 bit times × 16 RT cycles).

The maximum percent difference between the receiver count and the transmitter count of a fast 9-bit character with no errors is:

$$((170 - 176) \div 170) \times 100 = 3.53\%$$

45.4.3.9 Receiver wakeup

C1[WAKE] determines how the UART is brought out of the standby state to process an incoming message. C1[WAKE] enables either idle line wakeup or address mark wakeup.

Receiver wakeup is not supported when C7816[ISO_7816E] is set/enabled because multi-receiver systems are not allowed.

45.4.3.9.1 Idle input line wakeup (C1[WAKE] = 0)

In this wakeup method, an idle condition on the unsynchronized receiver input signal clears C2[RWU] and wakes the UART. The initial frame or frames of every message contain addressing information. All receivers evaluate the addressing information, and receivers for which the message is addressed process the frames that follow. Any receiver for which a message is not addressed can set its C2[RWU] and return to the standby state. C2[RWU] remains set and the receiver remains in standby until another idle character appears on the unsynchronized receiver input signal.

Idle line wakeup requires that messages be separated by at least one idle character and that no message contains idle characters.

When C2[RWU] is 1 and S2[RWUID] is 0, the idle character that wakes the receiver does not set S1[IDLE] or the receive data register full flag, S1[RDRF]. The receiver wakes and waits for the first data character of the next message which is stored in the receive data buffer. When S2[RWUID] and C2[RWU] are set and C1[WAKE] is cleared, any idle condition sets S1[IDLE] and generates an interrupt if enabled.

Idle input line wakeup is not supported when C7816[ISO_7816E] is set/enabled.

45.4.3.9.2 Address mark wakeup (C1[WAKE] = 1)

In this wakeup method, a logic 1 in the bit position immediately preceding the stop bit of a frame clears C2[RWU] and wakes the UART. A logic 1 in the bit position immediately preceeding the stop bit marks a frame as an address frame that contains addressing information. All receivers evaluate the addressing information, and the receivers for which the message is addressed process the frames that follow. Any receiver for which a message is not addressed can set its C2[RWU] and return to the standby state. C2[RWU] remains set and the receiver remains in standby until another address frame appears on the unsynchronized receiver input signal.

A logic 1 in the bit position immediately preceding the stop bit clears the receiver's C2[RWU] before the stop bit is received and places the received data into the receiver data buffer.

Address mark wakeup allows messages to contain idle characters but requires that the bit position immediately preceding the stop bit be reserved for use in address frames.

If module is in standby mode and nothing triggers to wake the UART, no error flag is set even if an invalid error condition is detected on the receiving data line.

Address mark wakeup is not supported when C7816[ISO_7816E] is set/enabled.

45.4.3.9.3 Match address operation

Match address operation is enabled when C4[MAEN1] or C4[MAEN2] is set. In this function, a frame received by the RX pin with a logic 1 in the bit position immediately preceding the stop bit is considered an address and is compared with the associated MA1 or MA2 register. The frame is transferred to the receive buffer, and S1[RDRF] is set, only if the comparison matches. All subsequent frames received with a logic 0 in the bit position immediately preceding the stop bit are considered to be data associated with the address and are transferred to the receive data buffer. If no marked address match occurs, then no transfer is made to the receive data buffer, and all following frames with logic 0 in the bit position immediately preceding the stop bit are also discarded. If both C4[MAEN1] and C4[MAEN2] are negated, the receiver operates normally and all data received is transferred to the receive data buffer.

Match address operation functions in the same way for both MA1 and MA2 registers.

- If only one of C4[MAEN1] and C4[MAEN2] is asserted, a marked address is compared only with the associated match register and data is transferred to the receive data buffer only on a match.
- If C4[MAEN1] and C4[MAEN2] are asserted, a marked address is compared with both match registers and data is transferred only on a match with either register.

Address match operation is not supported when C7816[ISO_7816E] is set/enabled.

45.4.4 Baud rate generation

A 13-bit modulus counter and a 5-bit fractional fine-adjust counter in the baud rate generator derive the baud rate for both the receiver and the transmitter. The value from 1 to 8191 written to SBR[12:0] determines the module clock divisor. The SBR bits are in the UART baud rate registers, BDH and BDL. The baud rate clock is synchronized with the module clock and drives the receiver. The fractional fine-adjust counter adds fractional delays to the baud rate clock to allow fine trimming of the baud rate to match the system baud rate. The transmitter is driven by the baud rate clock divided by 16. The receiver has an acquisition rate of 16 samples per bit time.

Baud rate generation is subject to two sources of error:

- Integer division of the module clock may not give the exact target frequency. This error can be reduced with the fine-adjust counter.
- Synchronization with the module clock can cause phase shift.

The Table 45-204 lists the available baud divisor fine adjust values.

UART baud rate = UART module clock / (16 × (SBR[12:0] + BRFD))

The following table lists some examples of achieving target baud rates with a module clock frequency of 10.2 MHz, with and without fractional fine adjustment.

Table 45-204. Baud rates (example: module clock = 10.2 MHz)

Bits SBR (decimal)	Bits BRFA	BRFD value	Receiver clock (Hz)	Transmitter clock (Hz)	Target Baud rate	Error (%)
17	00000	0	600,000.0	37,500.0	38,400	2.3
16	10011	19/32=0.59375	614,689.3	38,418.08	38,400	0.047
33	00000	0	309,090.9	19,318.2	19,200	0.62
33	00110	6/32=0.1875	307,344.6	19,209.04	19,200	0.047
66	00000	0	154,545.5	9659.1	9600	0.62
133	00000	0	76,691.7	4793.2	4800	0.14
266	00000	0	38,345.9	2396.6	2400	0.14
531	00000	0	19,209.0	1200.6	1200	0.11
1062	00000	0	9604.5	600.3	600	0.05
2125	00000	0	4800.0	300.0	300	0.00
4250	00000	0	2400.0	150.0	150	0.00
5795	00000	0	1760.1	110.0	110	0.00

Table 45-205. Baud rate fine adjust

BRFA	Baud Rate Fractional Divisor (BRFD)
0 0 0 0 0	0/32 = 0
0 0 0 0 1	1/32 = 0.03125
0 0 0 1 0	2/32 = 0.0625
0 0 0 1 1	3/32 = 0.09375
0 0 1 0 0	4/32 = 0.125
0 0 1 0 1	5/32 = 0.15625
0 0 1 1 0	6/32 = 0.1875
0 0 1 1 1	7/32 = 0.21875
0 1 0 0 0	8/32 = 0.25
0 1 0 0 1	9/32 = 0.28125
0 1 0 1 0	10/32 = 0.3125
0 1 0 1 1	11/32 = 0.34375
0 1 1 0 0	12/32 = 0.375
0 1 1 0 1	13/32 = 0.40625

Table continues on the next page...

Table 45-205. Baud rate fine adjust (continued)

BRFA	Baud Rate Fractional Divisor (BRFD)
0 1 1 1 0	14/32 = 0.4375
0 1 1 1 1	15/32 = 0.46875
1 0 0 0 0	16/32 = 0.5
1 0 0 0 1	17/32 = 0.53125
1 0 0 1 0	18/32 = 0.5625
1 0 0 1 1	19/32 = 0.59375
1 0 1 0 0	20/32 = 0.625
1 0 1 0 1	21/32 = 0.65625
1 0 1 1 0	22/32 = 0.6875
1 0 1 1 1	23/32 = 0.71875
1 1 0 0 0	24/32 = 0.75
1 1 0 0 1	25/32 = 0.78125
1 1 0 1 0	26/32 = 0.8125
1 1 0 1 1	27/32 = 0.84375
1 1 1 0 0	28/32 = 0.875
1 1 1 0 1	29/32 = 0.90625
1 1 1 1 0	30/32 = 0.9375
1 1 1 1 1	31/32 = 0.96875

45.4.5 Data format (non ISO-7816)

Each data character is contained in a frame that includes a start bit and a stop bit. The rest of the data format depends upon C1[M], C1[PE], S2[MSBF], and C4[M10].

45.4.5.1 Eight-bit configuration

Clearing C1[M] configures the UART for 8-bit data characters, that is, eight bits are memory mapped in D. A frame with eight data bits has a total of 10 bits. The most significant bit of the eight data bits can be used as an address mark to wake the receiver. If the most significant bit is used in this way, then it serves as an address or data indication, leaving the remaining seven bits as actual data. When C1[PE] is set, the eighth data bit is automatically calculated as the parity bit. See the following table.

Table 45-206. Configuration of 8-bit data format

UART_C1[PE]	Start bit	Data bits	Address bits	Parity bits	Stop bit
0	1	8	0	0	1
0	1	7	1[1]	0	1
1	1	7	0	1	1

1. The address bit identifies the frame as an address character. See Receiver wakeup.

45.4.5.2 Nine-bit configuration

When C1[M] is set and C4[M10] is cleared, the UART is configured for 9-bit data characters. If C1[PE] is enabled, the ninth bit is either C3[T8/R8] or the internally generated parity bit. This results in a frame consisting of a total of 11 bits. In the event that the ninth data bit is selected to be C3[T8], it will remain unchanged after transmission and can be used repeatedly without rewriting it, unless the value needs to be changed. This feature may be useful when the ninth data bit is being used as an address mark.

When C1[M] and C4[M10] are set, the UART is configured for 9-bit data characters, but the frame consists of a total of 12 bits. The 12 bits include the start and stop bits, the 9 data character bits, and a tenth internal data bit. Note that if C4[M10] is set, C1[PE] must also be set. In this case, the tenth bit is the internally generated parity bit. The ninth bit can either be used as an address mark or a ninth data bit.

See the following table.

Table 45-207. Configuration of 9-bit data formats

C1[PE]	UC1[M]	C1[M10]	Start bit	Data bits	Address bits	Parity bits	Stop bit
0	0	0	See Eight-bit configuration				
0	0	1	Invalid configuration				
0	1	0	1	9	0	0	1
0	1	0	1	8	1[1]	0	1
0	1	1	Invalid Configuration				
1	0	0	See Eight-bit configuration				
1	0	1	Invalid Configuration				
1	1	0	1	8	0	1	1
1	1	1	1	9	0	1	1
1	1	1	1	8	1[2]	1	1

1. The address bit identifies the frame as an address character.

2. The address bit identifies the frame as an address character.

Note

Unless in 9-bit mode with M10 set, do not use address mark wakeup with parity enabled.

45.4.5.3 Timing examples

Timing examples of these configurations in the NRZ mark/space data format are illustrated in the following figures. The timing examples show all of the configurations in the following sub-sections along with the LSB and MSB first variations.

45.4.5.3.1 Eight-bit format with parity disabled

The most significant bit can be used for address mark wakeup.

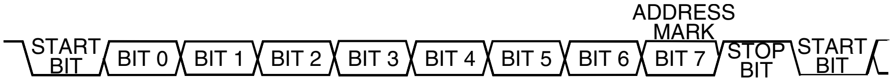

ADDRESS MARK
START BIT
BIT 0
BIT 1
BIT 2
BIT 3
BIT 4
BIT 5
BIT 6
BIT 7
STOP BIT
START BIT

Figure 45-204. Eight bits of data with LSB first

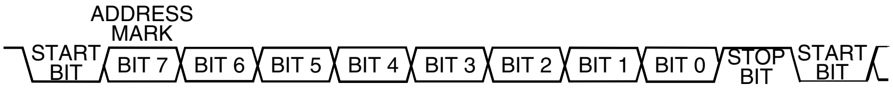

ADDRESS MARK
START BIT
BIT 7
BIT 6
BIT 5
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0
STOP BIT
START BIT

Figure 45-205. Eight bits of data with MSB first

45.4.5.3.2 Eight-bit format with parity enabled

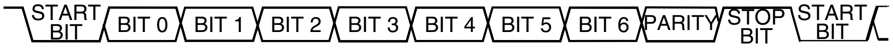

START BIT
BIT 0
BIT 1
BIT 2
BIT 3
BIT 4
BIT 5
BIT 6
PARITY
STOP BIT
START BIT

Figure 45-206. Seven bits of data with LSB first and parity

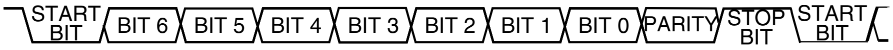

START BIT
BIT 6
BIT 5
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0
PARITY
STOP BIT
START BIT

Figure 45-207. Seven bits of data with MSB first and parity

45.4.5.3.3 Nine-bit format with parity disabled

The most significant bit can be used for address mark wakeup.

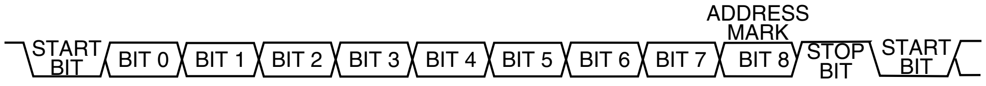

ADDRESS MARK
START BIT
BIT 0
BIT 1
BIT 2
BIT 3
BIT 4
BIT 5
BIT 6
BIT 7
BIT 8
STOP BIT
START BIT

Figure 45-208. Nine bits of data with LSB first

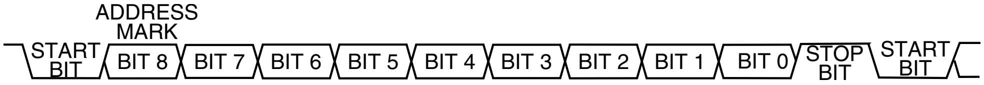

ADDRESS MARK
START BIT
BIT 8
BIT 7
BIT 6
BIT 5
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0
STOP BIT
START BIT

Figure 45-209. Nine bits of data with MSB first

45.4.5.3.4 Nine-bit format with parity enabled

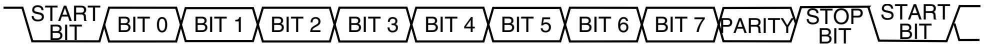

START BIT
BIT 0
BIT 1
BIT 2
BIT 3
BIT 4
BIT 5
BIT 6
BIT 7
PARITY
STOP BIT
START BIT

Figure 45-210. Eight bits of data with LSB first and parity

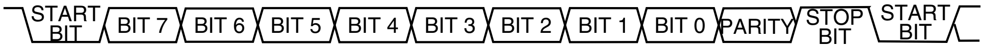

START BIT
BIT 7
BIT 6
BIT 5
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0
PARITY
STOP BIT
START BIT

Figure 45-211. Eight bits of data with MSB first and parity

45.4.5.3.5 Non-memory mapped tenth bit for parity

The most significant memory-mapped bit can be used for address mark wakeup.

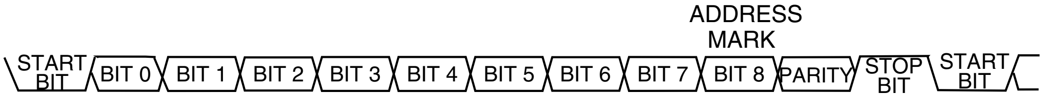

ADDRESS MARK
START BIT
BIT 0
BIT 1
BIT 2
BIT 3
BIT 4
BIT 5
BIT 6
BIT 7
BIT 8
PARITY
STOP BIT
START BIT

Figure 45-212. Nine bits of data with LSB first and parity

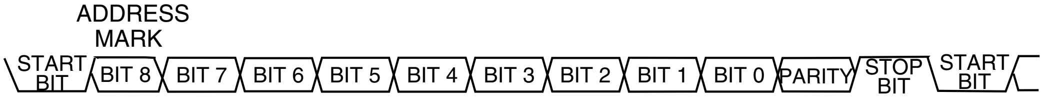

ADDRESS MARK
START BIT
BIT 8
BIT 7
BIT 6
BIT 5
BIT 4
BIT 3
BIT 2
BIT 1
BIT 0
PARITY
STOP BIT
START BIT

Figure 45-213. Nine bits of data with MSB first and parity

45.4.6 Single-wire operation

Normally, the UART uses two pins for transmitting and receiving. In single wire operation, the RXD pin is disconnected from the UART and the UART implements a half-duplex serial connection. The UART uses the TXD pin for both receiving and transmitting.

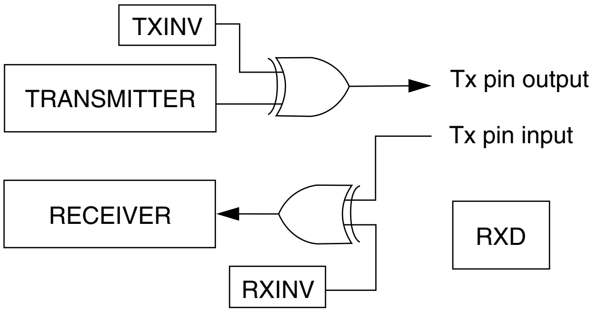

TXINV
TRANSMITTER
Tx pin output
Tx pin input
RECEIVER
RXD
RXINV

Figure 45-214. Single-wire operation (C1[LOOPS] = 1, C1[RSRC] = 1)

Enable single wire operation by setting C1[LOOPS] and the receiver source field, C1[RSRC]. Setting C1[LOOPS] disables the path from the unsynchronized receiver input signal to the receiver. Setting C1[RSRC] connects the receiver input to the output of the TXD pin driver. Both the transmitter and receiver must be enabled (C2[TE] = 1 and C2[RE] = 1). When C7816[ISO_7816EN] is set, it is not required that both C2[TE] and C2[RE] are set.

45.4.7 Loop operation

In loop operation, the transmitter output goes to the receiver input. The unsynchronized receiver input signal is disconnected from the UART.

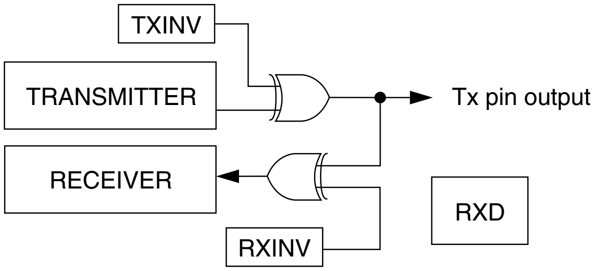


Figure 45-215. Loop operation (C1[LOOPS] = 1, C1[RSRC] = 0)

Enable loop operation by setting C1[LOOPS] and clearing C1[RSRC]. Setting C1[LOOPS] disables the path from the unsynchronized receiver input signal to the receiver. Clearing C1[RSRC] connects the transmitter output to the receiver input. Both the transmitter and receiver must be enabled (C2[TE] = 1 and C2[RE] = 1). When C7816[ISO_7816EN] is set, it is not required that both C2[TE] and C2[RE] are set.

45.4.8 ISO-7816/smartcard support

The UART provides mechanisms to support the ISO-7816 protocol that is commonly used to interface with smartcards. The ISO-7816 protocol is an NRZ, single wire, half-duplex interface. The TxD pin is used in open-drain mode because the data signal is used for both transmitting and receiving. There are multiple subprotocols within the ISO-7816 standard. The UART supports both T = 0 and T = 1 protocols. The module also provides for automated initial character detection and configuration, which allows for support of both direct convention and inverse convention data formats. A variety of interrupts specific to 7816 are provided in addition to the general interrupts to assist software. Additionally, the module is able to provide automated NACK responses and has programmed automated retransmission of failed packets. An assortment of programmable timeouts and guard band times are also supported.

The term elemental time unit (ETU) is frequently used in the context of ISO-7816. This concept is used to relate the frequency that the system (UART) is running at and the frequency that data is being transmitted and received. One ETU represents the time it takes to transmit or receive a single bit. For example, a standard 7816 packet, excluding any guard time or NACK elements is 10 ETUs (start bit, 8 data bits, and a parity bit). Guard times and wait times are also measured in ETUs.,

NOTE

The ISO-7816 specification may have certain configuration options that are reserved. To maintain maximum flexibility to support future 7816 enhancements or devices that may not strictly conform to the specification, the UART does not prevent those options being used. Further, the UART may provide configuration options that exceed the flexibility of options explicitly allowed by the 7816 specification. Failure to correctly configure the UART may result in unexpected behavior or incompatibility with the ISO-7816 specification.

45.4.8.1 Initial characters

In ISO-7816 with T = 0 mode, the UART can be configured to use C7816[INIT] to detect the next valid initial character, referred to by the ISO-7816 specifically as a TS character. When the initial character is detected, the UART provides the host processor with an interrupt if IE7816[INITDE] is set. Additionally, the UART will alter S2[MSBF], C3[TXINV], and S2[RXINV] automatically, based on the initial character. The corresponding initial character and resulting register settings are listed in the following table.

Table 45-208. Initial character automated settings

Initial character (bit 1-10)	Initial character (hex)	MSBF	TXINV	RXINV
LHHL LLL LLH inverse convention	3F	1	1	1
LHHL HHH LLH direct convention	3B	0	0	0

S2[MSBF], C3[TXINV], and S2[RXINV] must be reset to their default values before C7816[INIT] is set. Once C7816[INIT] is set, the receiver searches all received data for the first valid initial character. Detecting a Direct Convention Initial Character will cause no change to S2[MSBF], C3[TXINV], and S2[RXINV], while detecting an Inverse Convention Initial Character will cause these fields to set automatically. All data received, which is not a valid initial character, is ignored and all flags resulting from the invalid data are blocked from asserting. If C7816[ANACK] is set, a NACK is returned for invalid received initial characters and an RXT interrupt is generated as programmed.

45.4.8.2 Protocol T = 0

When T = 0 protocol is selected, a relatively complex error detection scheme is used. Data characters are formatted as illustrated in the following figure. This scheme is also used for answer to reset and Peripheral Pin Select (PPS) formats.

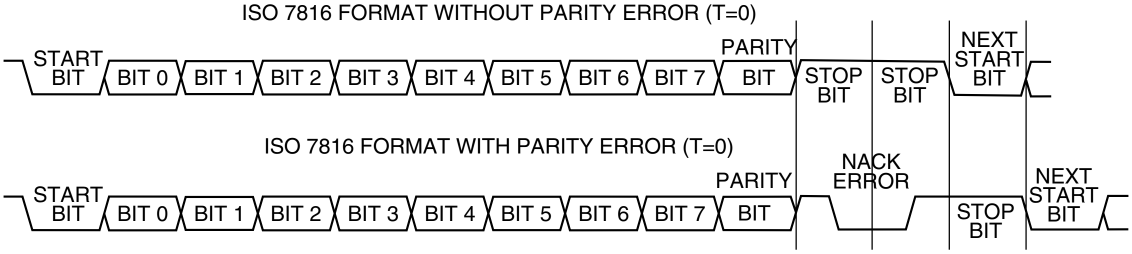

ISO 7816 FORMAT WITHOUT PARITY ERROR (T=0)
START BIT
BIT 0
BIT 1
BIT 2
BIT 3
BIT 4
BIT 5
BIT 6
BIT 7
PARITY BIT
STOP BIT
STOP BIT
NEXT START BIT
ISO 7816 FORMAT WITH PARITY ERROR (T=0)
START BIT
BIT 0
BIT 1
BIT 2
BIT 3
BIT 4
BIT 5
BIT 6
BIT 7
PARITY BIT
NACK ERROR
STOP BIT
NEXT START BIT

Figure 45-216. ISO-7816 T = 0 data format

As with other protocols supported by the UART, the data character includes a start bit. However, in this case, there are two stop bits rather than the typical single stop bit. In addition to a standard even parity check, the receiver has the ability to generate and return a NACK during the second half of the first stop bit period. The NACK must be at least one time period (ETU) in length and no more than two time periods (ETU) in length. The transmitter must wait for at least two time units (ETU) after detection of the error signal before attempting to retransmit the character.

It is assumed that the UART and the device (smartcard) know in advance which device is receiving and which is transmitting. No special mechanism is supplied by the UART to control receive and transmit in the mode other than C2[TE] and C2[RE]. Initial Character Detect feature is also supported in this mode.

45.4.8.3 Protocol T = 1

When T = 1 protocol is selected, the NACK error detection scheme is not used. Rather, the parity bit is used on a character basis and a CRC or LRC is used on the block basis, that is, for each group of characters. In this mode, the data format allows for a single stop bit although additional inactive bit periods may be present between the stop bit and the next start bit. Data characters are formatted as illustrated in the following figure.

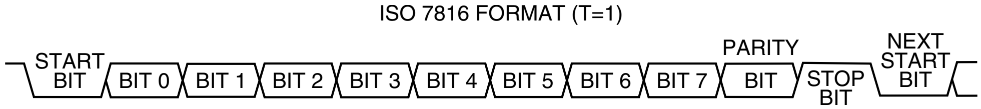


Figure 45-217. ISO 7816 T=1 data format

The smallest data unit that is transferred is a block. A block is made up of several data characters and may vary in size depending on the block type. The UART does not provide a mechanism to decode the block type. As part of the block, an LRC or CRC is included. The UART does not calculate the CRC or LRC for transmitted blocks, nor does it verify the validity of the CRC or LRC for received blocks. The 7816 protocol requires that the initiator and the smartcard (device) takes alternate turns in transmitting and receiving blocks. When the UART detects that the last character in a block has been transmitted it will automatically clear C2[TE] and enter receive mode. Therefore, the software must program the transmit buffer with the next data to be transmitted, and then enable C2[TE], once the software has determined that the last character of the received block has been received. The UART detects that the last character of the transmit block has been sent when TL7816[TLEN] = 0 and four additional characters have been sent. The four additional characters are made up of three prior to TL7816[TLEN] decrementing (prologue) and one after TL7816[TLEN] = 0, the final character of the epilogue.

45.4.8.4 Wait time and guard time parameters

The ISO-7816 specification defines several wait time and guard time parameters. The UART allows for flexible configuration and violation detection of these settings. On reset, the wait time (IS7816[WT]) defaults to 9600 ETUs and guard time (GT) to 12 ETUs. These values are controlled by parameters in the WP7816, WN7816, and WF7816 registers. Additionally, the value of C7816[TTYPE] also factors into the calculation. The formulae used to calculate the number ETUs for each wait time and guard time value are shown in Table 45-209.

Wait time (WT) is defined as the maximum allowable time between the leading edge of a character transmitted by the smartcard device and the leading edge of the previous character that was transmitted by the UART or the device. Similarly, character wait time (CWT) is defined as the maximum allowable time between the leading edge of two characters within the same block. Block wait time (BWT) is defined as the maximum time between the leading edge character of the last block received by the smartcard device and the leading edge of the first character transmitted by the smartcard device.

Guard time (GT) is defined as the minimum allowable time between the leading edge of two consecutive characters. Character guard time (CGT) is the minimum allowable time between the leading edges of two consecutive characters in the same direction, that is, transmission or reception. Block guard time (BGT) is the minimum allowable time between the leading edges of two consecutive characters in opposite directions, that is, transmission then reception or reception then transmission.

The GT and WT counters reset whenever C7816[TTYPE] = 1 or C7816[ISO_7816E] = 0 or a new dataword start bit has been received or transmitted as specified by the counter descriptions. The CWT, CGT, BWT, BGT counters reset whenever C7816[TTYPE] = 0 or C7816[ISO_7816E] = 0 or a new dataword start bit is received or transmitted as specified by the counter descriptions. When C7816[TTYPE] = 1, some of the counter values require an assumption regarding the first data transferred when the UART first starts. This assumption is required when the 7816E is disabled, when transition from C7816[TTYPE] = 0 to C7816[TTYPE] = 1 or when coming out of reset. In this case, it is assumed that the previous non-existent transfer was a received transfer.

The UART will automatically handle GT, CGT, and BGT such that the UART will not send a packet before the corresponding guard time expiring.

Table 45-209. Wait and guard time calculations

Parameter	Reset value [ETU]	C7816[TTYPE] = 0 [ETU]	C7816[TTYPE] = 1 [ETU]
Wait time (WT)	9600	WI × 960 × GTFD	Not used
Character wait time (CWT)	Not used	Not used	$11 + 2^{CWI}$
Block wait time (BWT)	Not used	Not used	$11 + 2^{BWI} \times 960 \times GTFD$
Guard time (GT)	12	**GTN not wqual to 255** 12 + GTN **GTN wqual to 255** 12	Not used
Character guard time (CGT)	Not used	Not used	**GTN not equal to 255** 12 + GTN **GTN equal to 255** 11
Block guard time (BGT)	Not used	Not used	22

45.4.8.5 Baud rate generation

The value in WF7816[GTFD] does not impact the clock frequency. SBR and BRFD are used to generate the clock frequency. This clock frequency is used by the UART only and is not seen by the smartcard device. The transmitter clocks operates at 1/16 the frequency of the receive clock so that the receiver is able to sample the received value 16 times during the ETU.

45.4.8.6 UART restrictions in ISO-7816 operation

Due to the flexibility of the UART module, there are several features and interrupts that are not supported while running in ISO-7816 mode. These restrictions are documented within the register field definitions.

45.4.9 Infrared interface

The UART provides the capability of transmitting narrow pulses to an IR LED and receiving narrow pulses and transforming them to serial bits, which are sent to the UART. The IrDA physical layer specification defines a half-duplex infrared communication link for exchanging data. The full standard includes data rates up to 16 Mbits/s. This design covers data rates only between 2.4 kbits/s and 115.2 kbits/s.

The UART has an infrared transmit encoder and receive decoder. The UART transmits serial bits of data that are encoded by the infrared submodule to transmit a narrow pulse for every zero bit. No pulse is transmitted for every one bit. When receiving data, the IR pulses are detected using an IR photo diode and transformed to CMOS levels by the IR receive decoder, external from the MCU. The narrow pulses are then stretched by the infrared receive decoder to get back to a serial bit stream to be received by the UART. The polarity of transmitted pulses and expected receive pulses can be inverted so that a direct connection can be made to external IrDA transceiver modules that use active low pulses.

The infrared submodule receives its clock sources from the UART. One of these two clocks are selected in the infrared submodule to generate either 3/16, 1/16, 1/32, or 1/4 narrow pulses during transmission.

45.4.9.1 Infrared transmit encoder

The infrared transmit encoder converts serial bits of data from transmit shift register to the TXD signal. A narrow pulse is transmitted for a zero bit and no pulse for a one bit. The narrow pulse is sent in the middle of the bit with a duration of 1/32, 1/16, 3/16, or 1/4 of a bit time. A narrow high pulse is transmitted for a zero bit when C3[TXINV] is cleared, while a narrow low pulse is transmitted for a zero bit when C3[TXINV] is set.

45.4.9.2 Infrared receive decoder

The infrared receive block converts data from the RXD signal to the receive shift register. A narrow pulse is expected for each zero received and no pulse is expected for each one received. A narrow high pulse is expected for a zero bit when S2[RXINV] is cleared, while a narrow low pulse is expected for a zero bit when S2[RXINV] is set. This receive decoder meets the edge jitter requirement as defined by the IrDA serial infrared physical layer specification.

45.5 Reset

All registers reset to a particular value are indicated in Memory map and registers.

45.6 System level interrupt sources

There are several interrupt signals that are sent from the UART. The following table lists the interrupt sources generated by the UART. The local enables for the UART interrupt sources are described in this table. Details regarding the individual operation of each interrupt are contained under various sub-sections of Memory map and registers. However, RXEDGIF description also outlines additional details regarding the RXEDGIF interrupt because of its complexity of operation. Any of the UART interrupt requests listed in the table can be used to bring the CPU out of Wait mode.

Table 45-210. UART interrupt sources

Interrupt Source	Flag	Local enable	DMA select
Transmitter	TDRE	TIE	TDMAS = 0
Transmitter	TC	TCIE	-
Receiver	IDLE	ILIE	-
Receiver	RDRF	RIE	RDMAS = 0
Receiver	LBKDIF	LBKDIE	-
Receiver	RXEDGIF	RXEDGIE	-
Receiver	OR	ORIE	-
Receiver	NF	NEIE	-
Receiver	FE	FEIE	-
Receiver	PF	PEIE	-
Receiver	RXUF	RXUFE	-
Transmitter	TXOF	TXOFE	-

Table continues on the next page...

Table 45-210. UART interrupt sources (continued)

Interrupt Source	Flag	Local enable	DMA select
Receiver	WT	WTWE	-
Receiver	CWT	CWTE	-
Receiver	BWT	BWTE	-
Receiver	INITD	INITDE	-
Receiver	TXT	TXTE	-
Receiver	RXT	RXTE	-
Receiver	GTV	GTVE	-

45.6.1 RXEDGIF description

S2[RXEDGIF] is set when an active edge is detected on the RxD pin. Therefore, the active edge can be detected only when in two wire mode. A RXEDGIF interrupt is generated only when S2[RXEDGIF] is set. If RXEDGIE is not enabled before S2[RXEDGIF] is set, an interrupt is not generated until S2[RXEDGIF] is set.

45.6.1.1 RxD edge detect sensitivity

Edge sensitivity can be software programmed to be either falling or rising. The polarity of the edge sensitivity is selected using S2[RXINV]. To detect the falling edge, S2[RXINV] is programmed to 0. To detect the rising edge, S2[RXINV] is programmed to 1.

Synchronizing logic is used prior to detect edges. Prior to detecting an edge, the receive data on RxD input must be at the deasserted logic level. A falling edge is detected when the RxD input signal is seen as a logic 1 (the deasserted level) during one module clock cycle, and then a logic 0 (the asserted level) during the next cycle. A rising edge is detected when the input is seen as a logic 0 during one module clock cycle and then a logic 1 during the next cycle.

45.6.1.2 Clearing RXEDGIF interrupt request

Writing a logic 1 to S2[RXEDGIF] immediately clears the RXEDGIF interrupt request even if the RxD input remains asserted. S2[RXEDGIF] remains set if another active edge is detected on RxD while attempting to clear S2[RXEDGIF] by writing a 1 to it.

45.6.1.3 Exit from low-power modes

The receive input active edge detect circuit is still active on low power modes (Wait and Stop). An active edge on the receive input brings the CPU out of low power mode if the interrupt is not masked (S2[RXEDGIF]=1).

45.7 DMA operation

In the transmitter, S1[TDRE] can be configured to assert a DMA transfer request. In the receiver, S1[RDRF], can be configured to assert a DMA transfer request. The following table shows the configuration field settings required to configure each flag for DMA operation.

Table 45-211. DMA configuration

Flag	Request enable bit	DMA select bit
TDRE	TIE = 1	TDMAS = 1
RDRF	RIE = 1	RDMAS = 1

When a flag is configured for a DMA request, its associated DMA request is asserted when the flag is set. When S1[RDRF] is configured as a DMA request, the clearing mechanism of reading S1, followed by reading D, does not clear the associated flag. The DMA request remains asserted until an indication is received that the DMA transactions are done. When this indication is received, the flag bit and the associated DMA request is cleared. If the DMA operation failed to remove the situation that caused the DMA request, another request is issued.

45.8 Application information

This section describes the UART application information.

45.8.1 Transmit/receive data buffer operation

The UART has independent receive and transmit buffers. The size of these buffers may vary depending on the implementation of the module. The implemented size of the buffers is a fixed constant via PFIFO[TXFIFOSIZE] and PFIFO[RXFIFOSIZE]. Additionally, legacy support is provided that allows for the FIFO structure to operate as a depth of one. This is the default/reset behavior of the module and can be adjusted using the PFIFO[RXFE] and PFIFO[TXFE] bits. Individual watermark levels are also provided for transmit and receive.

There are multiple ways to ensure that a data block, which is a set of characters, has completed transmission. These methods include:

1. Set TXFIFO[TXWATER] to 0. TDRE asserts when there is no further data in the transmit buffer. Alternatively the S1[TC] flag can be used to indicate when the transmit shift register is also empty.
2. Poll TCFIFO[TXCOUNT]. Assuming that only data for a data block has been put into the data buffer, when TCFIFO[TXCOUNT] = 0, all data has been transmitted or is in the process of transmission.
3. S1[TC] can be monitored. When S1[TC] asserts, it indicates that all data has been transmitted and there is no data currently being transmitted in the shift register.

45.8.2 ISO-7816 initialization sequence

This section outlines how to program the UART for ISO-7816 operation. Elements such as procedures to power up or power down the smartcard, and when to take those actions, are beyond the scope of this description. To set up the UART for ISO-7816 operation:

1. Select a baud rate. Write this value to the UART baud registers (BDH/L) to begin the baud rate generator. Remember that the baud rate generator is disabled when the baud rate is zero. Writing to the BDH has no effect without also writing to BDL. According to the 7816 specification the initial (default) baud rating setting should be Fi = 372 and Di = 1 and a maximum frequency of 5 MHz. In other words, the BDH, BDL, and C4 registers should be programmed such that the transmission frequency provided to the smartcard device must be 1/372th of the clock and must not exceed 5 MHz.
2. Write to set BDH[LBKDIE] = 0.
3. Write to C1 to configure word length, parity, and other configuration fields (LOOPS, RSRC) and set C1[M] = 1, C1[PE] = 1, and C1[PT] = 0.
4. Write to set S2[RWUID] = 0 and S2[LBKDE] = 0.
5. Write to set MODEM[RXRTSE] = 0, MODEM[TXRTSPOL] = 0, MODEM[TXRTSE] = 0, and MODEM[TXCTSE] = 0.

6. Write to set up interrupt enable fields desired (C3[ORIE], C3[NEIE], C3[PEIE], and C3[FEIE])
7. Write to set C4[MAEN1] = 0 and C4[MAEN2] = 0.
8. Write to C5 register and configure DMA control register fields as desired for application.
9. Write to set C7816[INIT] = 1,C7816[TTYPE] = 0, and C7816[ISO_7816E] = 1. Program C7816[ONACK] and C7816[ANACK] as desired.
10. Write to IE7816 to set interrupt enable parameters as desired.
11. Write to ET7816 and set as desired.
12. Write to set C2[ILIE] = 0, C2[RE] = 1, C2[TE] = 1, C2[RWU] = 0, and C2[SBK] = 0. Set up interrupt enables C2[TIE], C2[TCIE], and C2[RIE] as desired.

At this time, the UART will start listening for an initial character. After being identified, it will automatically adjust S2[MSBF], C3[TXINV], and S2[RXINV]. The software must then receive and process an answer to reset. Upon processing the answer to reset, the software must write to set C2[RE] = 0 and C2[TE] = 0. The software should then adjust 7816 specific and UART generic parameters to match and configure data that was received during the answer on reset period. After the new settings have been programmed, including the new baud rate and C7816[TTYPE], C2[RE] and C2[TE] can be reenabled as required.

45.8.2.1 Transmission procedure for (C7816[TTYPE] = 0)

When the protocol selected is C7816[TTYPE] = 0, it is assumed that the software has a prior knowledge of who should be transmitting and receiving. Therefore, no mechanism is provided for automated transmission/receipt control. The software must monitor S1[TDRE], or configure for an interrupt, and provide additional data for transmission, as appropriate. Additionally, software should set C2[TE] = 1 and control TXDIR whenever it is the UART's turn to transmit information. For ease of monitoring, it is suggested that only data be transmitted until the next receiver/transmit switchover is loaded into the transmit FIFO/buffer.

45.8.2.2 Transmission procedure for (C7816[TTYPE] = 1)

When the protocol selected is C7816[TTYPE] = 1, data is transferred in blocks. Before starting a transmission, the software must write the size, in number of bytes, for the Information Field portion of the block into TLEN. If a CRC is being transmitted for the block, the value in TLEN must be one more than the size of the information field. The software must then set C2[TE] = 1 and C2[RE] = 1. The software must then monitor S1[TDRE]/interrupt and write the prologue, information, and epilogue field to the transmit buffer. TLEN automatically decrements, except for prologue bytes and the final epilogue byte. When the final epilogue byte has been transmitted, the UART automatically clears C2[TE] to 0, and the UART automatically starts capturing the response to the block that was transmitted. After the software has detected the receipt of the response, the transmission process must be repeated as needed with sufficient urgency to ensure that the block wait time and character wait times are not violated.

45.8.3 Initialization sequence (non ISO-7816)

To initiate a UART transmission:

1. Configure the UART.
    a. Select a baud rate. Write this value to the UART baud registers (BDH/L) to begin the baud rate generator. Remember that the baud rate generator is disabled when the baud rate is zero. Writing to the BDH has no effect without also writing to BDL.
    b. Write to C1 to configure word length, parity, and other configuration bits (LOOPS, RSRC, M, WAKE, ILT, PE, and PT). Write to C4, MA1, and MA2 to configure.
    c. Enable the transmitter, interrupts, receiver, and wakeup as required, by writing to C2 (TIE, TCIE, RIE, ILIE, TE, RE, RWU, and SBK), S2 (MSBF and BRK13), and C3 (ORIE, NEIE, PEIE, and FEIE). A preamble or idle character is then shifted out of the transmitter shift register.
2. Transmit procedure for each byte.
    a. Monitor S1[TDRE] by reading S1 or responding to the TDRE interrupt. The amount of free space in the transmit buffer directly using TCFIFO[TXCOUNT] can also be monitored.
    b. If the TDRE flag is set, or there is space in the transmit buffer, write the data to be transmitted to (C3[T8]/D). A new transmission will not result until data exists in the transmit buffer.

3. Repeat step 2 for each subsequent transmission.

Note

During normal operation, S1[TDRE] is set when the shift register is loaded with the next data to be transmitted from the transmit buffer and the number of datawords contained in the transmit buffer is less than or equal to the value in TWFIFO[TXWATER]. This occurs 9/16ths of a bit time after the start of the stop bit of the previous frame.

To separate messages with preambles with minimum idle line time, use this sequence between messages.

1. Write the last dataword of the first message to C3[T8]/D.
2. Wait for S1[TDRE] to go high with TWFIFO[TXWATER] = 0, indicating the transfer of the last frame to the transmit shift register.
3. Queue a preamble by clearing and then setting C2[TE].
4. Write the first and subsequent datawords of the second message to C3[T8]/D.

45.8.4 Overrun (OR) flag implications

To be flexible, the overrun flag (OR) operates slight differently depending on the mode of operation. There may be implications that need to be carefully considered. This section clarifies the behavior and the resulting implications. Regardless of mode, if a dataword is received while S1[OR] is set, S1[RDRF] and S1[IDLE] are blocked from asserting. If S1[RDRF] or S1[IDLE] were previously asserted, they will remain asserted until cleared.

45.8.4.1 Overrun operation

The assertion of S1[OR] indicates that a significant event has occurred. The assertion indicates that received data has been lost because there was a lack of room to store it in the data buffer. Therefore, while S1[OR] is set, no further data is stored in the data buffer until S1[OR] is cleared. This ensures that the application will be able to handle the overrun condition.

In most applications, because the total amount of lost data is known, the application will attempt to return the system to a known state. Before S1[OR] is cleared, all received data will be dropped. For this, the software does the following.

1. Remove data from the receive data buffer. This could be done by reading data from the data buffer and processing it if the data in the FIFO was still valuable when the overrun event occurred, or using CFIFO[RXFLUSH] to clear the buffer.
2. Clear S1[OR]. Note that if data was cleared using CFIFO[RXFLUSH], then clearing S1[OR] will result in SFIFO[RXUF] asserting. This is because the only way to clear S1[OR] requires reading additional information from the FIFO. Care should be taken to disable the SFIFO[RXUF] interrupt prior to clearing the OR flag and then clearing SFIFO[RXUF] after the OR flag has been cleared.

Note that, in some applications, if an overrun event is responded to fast enough, the lost data can be recovered. For example, when C7816[ISO_7816E] is asserted, C7816[TTYPE]=1 and C7816[ONACK] = 1, the application may reasonably be able to determine whether the lost data will be resent by the device. In this scenario, flushing the receiver data buffer may not be required. Rather, if S1[OR] is cleared, the lost data may be resent and therefore may be recoverable.

When LIN break detect (LBKDE) is asserted, S1[OR] has significantly different behavior than in other modes. S1[OR] will be set, regardless of how much space is actually available in the data buffer, if a LIN break character has been detected and the corresponding flag, S2[LBKDIF], is not cleared before the first data character is received after S2[LBKDIF] asserted. This behavior is intended to allow the software sufficient time to read the LIN break character from the data buffer to ensure that a break character was actually detected. The checking of the break character was used on some older implementations and is therefore supported for legacy reasons. Applications that do not require this checking can simply clear S2[LBKDIF] without checking the stored value to ensure it is a break character.

45.8.5 Overrun NACK considerations

When C7816[ISO_7816E] is enabled and C7816[TTYPE] = 0, the retransmission feature of the 7816 protocol can be used to help avoid lost data when the data buffer overflows. Using C7816[ONACK], the module can be programmed to issue a NACK on an overflow event. Assuming that the smartcard device has implemented retransmission, the lost data will be retransmitted. While useful, there is a programming implication that may require special consideration. The need to transmit a NACK must be determined and committed to prior to the dataword being fully received. While the NACK is being received, it is possible that the application code will read the data buffer such that sufficient room will be made to store the dataword that is being NACKed. Even if room has been made in the data buffer after the transmission of a NACK is completed, the received data will always be discarded as a result of an overflow and the

ET7816[RXTHRESHOLD] value will be incremented by one. However, if sufficient space now exists to write the received data which was NACK'ed, S1[OR] will be blocked and kept from asserting.

45.8.6 Match address registers

The two match address registers allow a second match address function for a broadcast or general call address to the serial bus, as an example.

45.8.7 Modem feature

This section describes the modem features.

45.8.7.1 Ready-to-receive using $\overline{\text{RTS}}$

To help to stop overrun of the receiver data buffer, the $\overline{\text{RTS}}$ signal can be used by the receiver to indicate to another UART that it is ready to receive data. The other UART can send the data when its $\overline{\text{CTS}}$ signal is asserted. This handshaking conforms to the TIA-232-E standard. A transceiver is necessary if the required voltage levels of the communication link do not match the voltage levels of the UART's $\overline{\text{RTS}}$ and $\overline{\text{CTS}}$ signals.

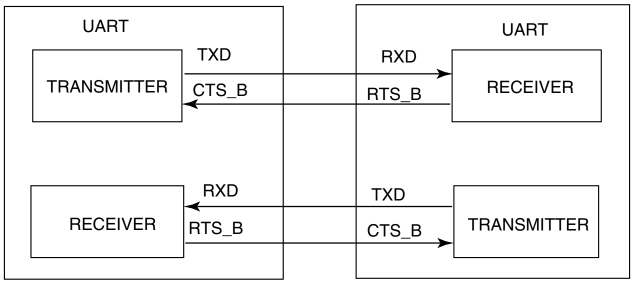


Figure 45-218. Ready-to-receive

The transmitter's $\overline{\text{CTS}}$ signal can be used for hardware flow control whether its $\overline{\text{RTS}}$ signal is used for hardware flow control, transceiver driver enable, or not at all.

45.8.7.2 Transceiver driver enable using $\overline{\text{RTS}}$

RS-485 is a multiple drop communication protocol in which the UART transceiver's driver is 3-stated unless the UART is driving. The $\overline{\text{RTS}}$ signal can be used by the transmitter to enable the driver of a transceiver. The polarity of $\overline{\text{RTS}}$ can be matched to the polarity of the transceiver's driver enable signal. See the following figure.

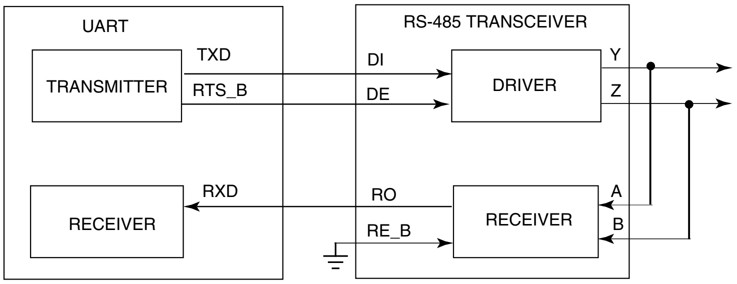


Figure 45-219. Transceiver driver enable using $\overline{\text{RTS}}$

In the figure, the receiver enable signal is asserted. Another option for this connection is to connect RTS_B to both DE and RE_B. The transceiver's receiver is disabled while driving. A pullup can pull RXD to a non-floating value during this time. This option can be refined further by operating the UART in single wire mode, freeing the RXD pin for other uses.

45.8.8 IrDA minimum pulse width

The IrDA specifies a minimum pulse width of 1.6 μs. The UART hardware does not include a mechanism to restrict/force the pulse width to be greater than or equal to 1.6 μs. However, configuring the baud rate to 115.2 kbit/s and the narrow pulse width to 3/16 of a bit time results in a pulse width of 1.6 μs.

45.8.9 Clearing 7816 wait timer (WT, BWT, CWT) interrupts

The 7816 wait timer interrupts associated with IS7816[WT], IS7816[BWT], and IS7816[CWT] will automatically reassert if they are cleared and the wait time is still violated. This behavior is similar to most of the other interrupts on the UART. In most cases, if the condition that caused the interrupt to trigger still exists when the interrupt is cleared, then the interrupt will reassert. For example, consider the following scenario:

1. IS7816[WT] is programmed to assert after 9600 cycles of unresponsiveness.
2. The 9600 cycles pass without a response resulting in the WT interrupt asserting.
3. The IS7816[WT] is cleared at cycle 9700 by the interrupt service routine.
4. After the WT interrupt has been cleared, the smartcard remains unresponsive. At cycle 9701 the WT interrupt will reassert.

If the intent of clearing the interrupt is such that it does not reassert, the interrupt service routine must remove or clear the condition that originally caused the interrupt to assert prior to clearing the interrupt. There are multiple ways that this can be accomplished, including ensuring that an event that results in the wait timer resetting occurs, such as, the transmission of another packet.

45.8.10 Legacy and reverse compatibility considerations

Recent versions of the UART have added several new features. Whenever reasonably possible, reverse compatibility was maintained. However, in some cases this was either not feasible or the behavior was deemed as not intended. This section describes several differences to legacy operation that resulted from these recent enhancements. If application code from previous versions is used, it must be reviewed and modified to take the following items into account. Depending on the application code, additional items that are not listed here may also need to be considered.

1. Various reserved registers and register bits are used, such as, MSFB and M10.
2. This module now generates an error when invalid address spaces are used.
3. While documentation indicated otherwise, in some cases it was possible for S1[IDLE] to assert even if S1[OR] was set.
4. S1[OR] will be set only if the data buffer (FIFO) does not have sufficient room. Previously, the data buffer was always a fixed size of one and the S1[OR] flag would set so long as S1[RDRF] was set even if there was room in the data buffer. While the clearing mechanism has remained the same for S1[RDRF], keeping the OR flag assertion tied to the RDRF event rather than the data buffer being full would have greatly reduced the usefulness of the buffer when its size is larger than one.
5. Previously, when C2[RWU] was set (and WAKE = 0), the IDLE flag could reassert up to every bit period causing an interrupt and requiring the host processor to reassert C2[RWU]. This behavior has been modified. Now, when C2[RWU] is set (and WAKE = 0), at least one non-idle bit must be detected before an idle can be detected.

Chapter 46
Synchronous Audio Interface (SAI)

46.1 Introduction

NOTE

For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The I^2S (or I2S) module provides a synchronous audio interface (SAI) that supports full-duplex serial interfaces with frame synchronization such as I^2S, AC97, and codec/DSP interfaces.

46.1.1 Features

- Transmitter with independent bit clock and frame sync supporting 1 data channel
- Receiver with independent bit clock and frame sync supporting 1 data channel
- Maximum Frame Size of 16 words
- Word size of between 8-bits and 32-bits
- Word size configured separately for first word and remaining words in frame
- Asynchronous 4 × 32-bit FIFO for each transmit and receive channel
- Graceful restart after FIFO error

46.1.2 Block diagram

The following block diagram also shows the module clocks.

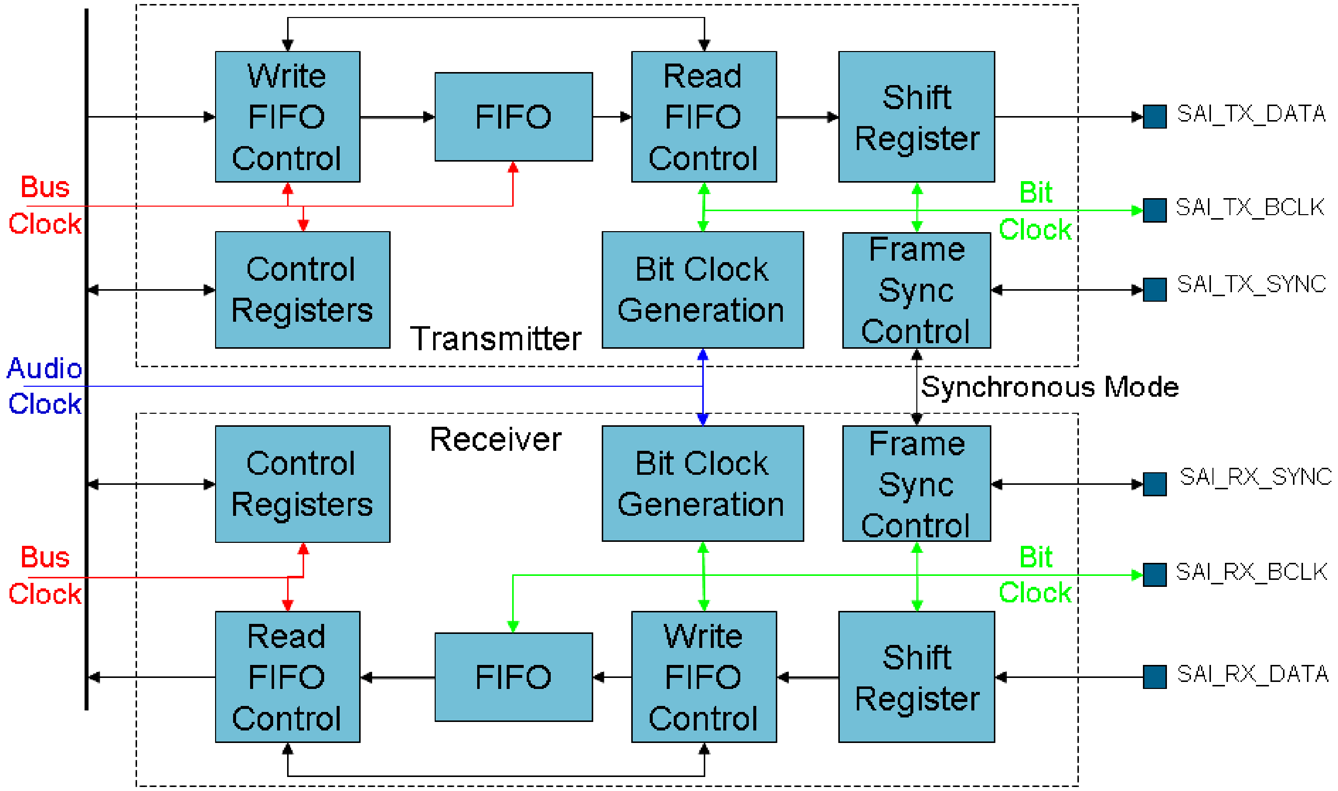

Write FIFO Control
FIFO
Read FIFO Control
Shift Register
SAI_TX_DATA
Bus Clock
Bit Clock
SAI_TX_BCLK
Control Registers
Transmitter
Bit Clock Generation
Frame Sync Control
SAI_TX_SYNC
Audio Clock
Synchronous Mode
Receiver
Control Registers
Bit Clock Generation
Frame Sync Control
SAI_RX_SYNC
Bus Clock
Bit Clock
SAI_RX_BCLK
Read FIFO Control
FIFO
Write FIFO Control
Shift Register
SAI_RX_DATA

Figure 46-1. I^2S/SAI block diagram

46.1.3 Modes of operation

The module operates in these MCU power modes: Run mode, stop modes, low-leakage modes, and Debug mode.

46.1.3.1 Run mode

In Run mode, the SAI transmitter and receiver operate normally.

46.1.3.2 Stop modes

In Stop mode, the SAI transmitter and/or receiver can continue operating provided the appropriate Stop Enable bit is set (TCSR[STOPE] and/or RCSR[STOPE], respectively), and provided the transmitter and/or receiver is/are using an externally generated bit clock or an Audio Master Clock that remains operating in Stop mode. The SAI transmitter and/or receiver can generate an asynchronous interrupt to wake the CPU from Stop mode.

In Stop mode, if the Transmitter Stop Enable (TCSR[STOPE]) bit is clear, the transmitter is disabled after completing the current transmit frame, and, if the Receiver Stop Enable (RCSR[STOPE]) bit is clear, the receiver is disabled after completing the current receive frame. Entry into Stop mode is prevented–not acknowledged–while waiting for the transmitter and receiver to be disabled at the end of the current frame.

46.1.3.3 Low-leakage modes

When entering low-leakage modes, the Stop Enable (TCSR[STOPE] and RCSR[STOPE]) bits are ignored and the SAI is disabled after completing the current transmit and receive Frames. Entry into stop mode is prevented (not acknowledged) while waiting for the transmitter and receiver to be disabled at the end of the current frame.

46.1.3.4 Debug mode

In Debug mode, the SAI transmitter and/or receiver can continue operating provided the Debug Enable bit is set. When TCSR[DBGE] or RCSR[DBGE] bit is clear and Debug mode is entered, the SAI is disabled after completing the current transmit or receive frame. The transmitter and receiver bit clocks are not affected by Debug mode.

46.2 External signals

Name	Function	I/O	Reset	Pull
SAI_TX_BCLK	Transmit Bit Clock	I/O	0	—
SAI_TX_SYNC	Transmit Frame Sync	I/O	0	—
SAI_TX_DATA	Transmit Data	O	0	—
SAI_RX_BCLK	Receive Bit Clock	I/O	0	—
SAI_RX_SYNC	Receive Frame Sync	I/O	0	—
SAI_RX_DATA	Receive Data	I	0	—
SAI_MCLK	Audio Master Clock	I/O	0	—

46.3 Memory map and register definition

A read or write access to an address after the last register will result in a bus error.

I2S memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_F000	SAI Transmit Control Register (I2S0_TCSR)	32	R/W	0000_0000h	46.3.1/ 1149
4002_F004	SAI Transmit Configuration 1 Register (I2S0_TCR1)	32	R/W	0000_0000h	46.3.2/ 1152
4002_F008	SAI Transmit Configuration 2 Register (I2S0_TCR2)	32	R/W	0000_0000h	46.3.3/ 1152
4002_F00C	SAI Transmit Configuration 3 Register (I2S0_TCR3)	32	R/W	0000_0000h	46.3.4/ 1154
4002_F010	SAI Transmit Configuration 4 Register (I2S0_TCR4)	32	R/W	0000_0000h	46.3.5/ 1155
4002_F014	SAI Transmit Configuration 5 Register (I2S0_TCR5)	32	R/W	0000_0000h	46.3.6/ 1156
4002_F020	SAI Transmit Data Register (I2S0_TDR0)	32	W (always reads zero)	0000_0000h	46.3.7/ 1157
4002_F040	SAI Transmit FIFO Register (I2S0_TFR0)	32	R	0000_0000h	46.3.8/ 1157
4002_F060	SAI Transmit Mask Register (I2S0_TMR)	32	R/W	0000_0000h	46.3.9/ 1158
4002_F080	SAI Receive Control Register (I2S0_RCSR)	32	R/W	0000_0000h	46.3.10/ 1158
4002_F084	SAI Receive Configuration 1 Register (I2S0_RCR1)	32	R/W	0000_0000h	46.3.11/ 1161
4002_F088	SAI Receive Configuration 2 Register (I2S0_RCR2)	32	R/W	0000_0000h	46.3.12/ 1162
4002_F08C	SAI Receive Configuration 3 Register (I2S0_RCR3)	32	R/W	0000_0000h	46.3.13/ 1163
4002_F090	SAI Receive Configuration 4 Register (I2S0_RCR4)	32	R/W	0000_0000h	46.3.14/ 1164
4002_F094	SAI Receive Configuration 5 Register (I2S0_RCR5)	32	R/W	0000_0000h	46.3.15/ 1165
4002_F0A0	SAI Receive Data Register (I2S0_RDR0)	32	R	0000_0000h	46.3.16/ 1166

Table continues on the next page...

I2S memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4002_F0C0	SAI Receive FIFO Register (I2S0_RFR0)	32	R	0000_0000h	46.3.17/ 1166
4002_F0E0	SAI Receive Mask Register (I2S0_RMR)	32	R/W	0000_0000h	46.3.18/ 1167
4002_F100	SAI MCLK Control Register (I2S0_MCR)	32	R/W	0000_0000h	46.3.19/ 1168
4002_F104	SAI MCLK Divide Register (I2S0_MDR)	32	R/W	0000_0000h	46.3.20/ 1169

46.3.1 SAI Transmit Control Register (I2Sx_TCSR)

Addresses: I2S0_TCSR is 4002_F000h base + 0h offset = 4002_F000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TE	STOPE	DBGE	BCE	0		0	SR	0			WSF	SEF	FEF	FWF	FRF
W							FR					w1c	w1c	w1c		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WSIE	SEIE	FEIE	FWIE	FRIE	0			0			FWDE	FRDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCSR field descriptions

Field	Description
31 TE	Transmitter Enable Enables/disables the transmitter. When software clears this field, the transmitter remains enabled, and this bit remains set, until the end of the current frame. 0 Transmitter is disabled. 1 Transmitter is enabled, or transmitter has been disabled and has not yet reached end of frame.
30 STOPE	Stop Enable Configures transmitter operation in Stop mode. This field is ignored and the transmitter is disabled in all low-leakage stop modes. 0 Transmitter disabled in Stop mode. 1 Transmitter enabled in Stop mode.
29 DBGE	Debug Enable Enables/disables transmitter operation in Debug mode. The transmit bit clock is not affected by debug mode.

Table continues on the next page...

I2Sx_TCSR field descriptions (continued)

Field	Description
	0 Transmitter is disabled in Debug mode, after completing the current frame. 1 Transmitter is enabled in Debug mode.
28 BCE	Bit Clock Enable Enables the transmit bit clock, separately from the TE. This field is automatically set whenever TE is set. When software clears this field, the transmit bit clock remains enabled, and this bit remains set, until the end of the current frame. 0 Transmit bit clock is disabled. 1 Transmit bit clock is enabled.
27–26 Reserved	This read-only field is reserved and always has the value zero.
25 FR	FIFO Reset Resets the FIFO pointers. Reading this field will always return zero. 0 No effect. 1 FIFO reset.
24 SR	Software Reset When set, resets the internal transmitter logic including the FIFO pointers. Software-visible registers are not affected, except for the status registers. 0 No effect. 1 Software reset.
23–21 Reserved	This read-only field is reserved and always has the value zero.
20 WSF	Word Start Flag Indicates that the start of the configured word has been detected. Write a logic 1 to this field to clear this flag. 0 Start of word not detected. 1 Start of word detected.
19 SEF	Sync Error Flag Indicates that an error in the externally-generated frame sync has been detected. Write a logic 1 to this field to clear this flag. 0 Sync error not detected. 1 Frame sync error detected.
18 FEF	FIFO Error Flag Indicates that an enabled transmit FIFO has underrun. Write a logic 1 to this field to clear this flag. 0 Transmit underrun not detected. 1 Transmit underrun detected.
17 FWF	FIFO Warning Flag Indicates that an enabled transmit FIFO is empty.

Table continues on the next page...

I2Sx_TCSR field descriptions (continued)

Field	Description
	0 No enabled transmit FIFO is empty. 1 Enabled transmit FIFO is empty.
16 FRF	FIFO Request Flag Indicates that the number of words in an enabled transmit channel FIFO is less than or equal to the transmit FIFO watermark. 0 Transmit FIFO watermark has not been reached. 1 Transmit FIFO watermark has been reached.
15–13 Reserved	This read-only field is reserved and always has the value zero.
12 WSIE	Word Start Interrupt Enable Enables/disables word start interrupts. 0 Disables interrupt. 1 Enables interrupt.
11 SEIE	Sync Error Interrupt Enable Enables/disables sync error interrupts. 0 Disables interrupt. 1 Enables interrupt.
10 FEIE	FIFO Error Interrupt Enable Enables/disables FIFO error interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
9 FWIE	FIFO Warning Interrupt Enable Enables/disables FIFO warning interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
8 FRIE	FIFO Request Interrupt Enable Enables/disables FIFO request interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
7–5 Reserved	This read-only field is reserved and always has the value zero.
4–2 Reserved	This read-only field is reserved and always has the value zero.
1 FWDE	FIFO Warning DMA Enable Enables/disables DMA requests.

Table continues on the next page...

I2Sx_TCSR field descriptions (continued)

Field	Description
	0 Disables the DMA request. 1 Enables the DMA request.
0 FRDE	FIFO Request DMA Enable Enables/disables DMA requests. 0 Disables the DMA request. 1 Enables the DMA request.

46.3.2 SAI Transmit Configuration 1 Register (I2Sx_TCR1)

Addresses: I2S0_TCR1 is 4002_F000h base + 4h offset = 4002_F004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																														TFW	
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR1 field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero.
1–0 TFW	Transmit FIFO Watermark Configures the watermark level for all enabled transmit channels.

46.3.3 SAI Transmit Configuration 2 Register (I2Sx_TCR2)

This register must not be altered when TCSR[TE] is set.

Addresses: I2S0_TCR2 is 4002_F000h base + 8h offset = 4002_F008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SYNC		BCS	BCI	MSEL		BCP	BCD	0																DIV							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR2 field descriptions

Field	Description
31–30 SYNC	Synchronous Mode Configures between asynchronous and synchronous modes of operation. When configured for a synchronous mode of operation, the receiver must be configured for asynchronous operation. 00 Asynchronous mode. 01 Synchronous with receiver. 10 Synchronous with another SAI transmitter. 11 Synchronous with another SAI receiver.
29 BCS	Bit Clock Swap When the SAI is in asynchronous mode and this field is set to 1, the transmitter is clocked by the receiver bit clock. When the SAI is in synchronous mode and this field is set to 1, the transmitter is clocked by the transmitter bit clock, but it uses the receiver frame sync. 0 Use the normal bit clock source. 1 Swap the bit clock source.
28 BCI	Bit Clock Input When set in either asynchronous or synchronous mode and using an internally generated bit clock, configures the internal logic to be clocked as if the bit clock was externally generated. This has the effect of decreasing data input setup time, but increasing data output valid time. This bit has no effect when configured for an externally generated bit clock. 0 No effect. 1 Internal logic is clocked by external bit clock.
27–26 MSEL	MCLK Select Selects the Audio Master Clock used to generate an internally generated bit clock. This field has no effect when configured for an externally generated bit clock. 00 Bus Clock selected. 01 Master Clock 1 selected. 10 Master Clock 2 selected. 11 Master Clock 3 selected.
25 BCP	Bit Clock Polarity Configures the polarity of the bit clock. 0 Bit clock is active high with drive outputs on rising edge and sample inputs on falling edge. 1 Bit clock is active low with drive outputs on falling edge and sample inputs on rising edge.
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0 Bit clock is generated externally in Slave mode. 1 Bit clock is generated internally in Master mode.
23–8 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

I2Sx_TCR2 field descriptions (continued)

Field	Description
7–0 DIV	Bit Clock Divide Divides down the audio master clock to generate the bit clock when configured for an internal bit clock. The division value is (DIV + 1) * 2.

46.3.4 SAI Transmit Configuration 3 Register (I2Sx_TCR3)

This register must not be altered when TCSR[TE] is set.

Addresses: I2S0_TCR3 is 4002_F000h base + Ch offset = 4002_F00Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															TCE	0												WDFL			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR3 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value zero.
16 TCE	Transmit Channel Enable Enables a data channel for a transmit operation. A channel must be enabled before its FIFO is accessed. 0 Transmit data channel is disabled. 1 Transmit data channel is enabled.
15–4 Reserved	This read-only field is reserved and always has the value zero.
3–0 WDFL	Word Flag Configuration Configures which word sets the start of word flag. The value written must be one less than the word number. For example, writing 0 configures the first word in the frame. When configured to a value greater than TCR4[FRSZ], then the start of word flag is never set.

46.3.5 SAI Transmit Configuration 4 Register (I2Sx_TCR4)

This register must not be altered when TCSR[TE] is set.

Addresses: I2S0_TCR4 is 4002_F000h base + 10h offset = 4002_F010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												FRSZ				0			SYWD					0			MF	FSE	0	FSP	FSD
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR4 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value zero.
19–16 FRSZ	Frame Size Configures the number of words in each frame. The value written should be one less than the number of words in the frame (for example, write 0 for one word per frame). The maximum supported frame size is 16 words.
15–13 Reserved	This read-only field is reserved and always has the value zero.
12–8 SYWD	Sync Width Configures the length of the frame sync in number of bit clocks. The value written must be one less than the number of bit clocks. For example, write 0 for the frame sync to assert for one bit clock only. The sync width cannot be configured longer than the first word of the frame.
7–5 Reserved	This read-only field is reserved and always has the value zero.
4 MF	MSB First Specifies whether the LSB or the MSB is transmitted/received first. 0 LSB is transmitted/received first. 1 MSB is transmitted/received first.
3 FSE	Frame Sync Early 0 Frame sync asserts with the first bit of the frame. 1 Frame sync asserts one bit before the first bit of the frame.
2 Reserved	This read-only field is reserved and always has the value zero.
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync. 0 Frame sync is active high. 1 Frame sync is active low.

Table continues on the next page...

I2Sx_TCR4 field descriptions (continued)

Field	Description
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0 Frame sync is generated externally in Slave mode. 1 Frame sync is generated internally in Master mode.

46.3.6 SAI Transmit Configuration 5 Register (I2Sx_TCR5)

This register must not be altered when TCSR[TE] is set.

Addresses: I2S0_TCR5 is 4002_F000h base + 14h offset = 4002_F014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WNW					0			W0W					0			FBT					0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TCR5 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero.
28–24 WNW	Word N Width Configures the number of bits in each word, for each word except the first in the frame. The value written must be one less than the number of bits per word. The value of WNW must be greater than or equal to the value of W0W even when there is only one word in each frame. Word width of less than 8 bits is not supported.
23–21 Reserved	This read-only field is reserved and always has the value zero.
20–16 W0W	Word 0 Width Configures the number of bits in the first word in each frame. The value written must be one less than the number of bits in the first word. Word width of less than 8 bits is not supported if there is only one word per frame.
15–13 Reserved	This read-only field is reserved and always has the value zero.
12–8 FBT	First Bit Shifted Configures the bit index for the first bit transmitted for each word in the frame. If configured for MSB First, the index of the next bit transmitted is one less than the current bit transmitted. If configured for LSB First, the index of the next bit transmitted is one more than the current bit transmitted. The value written must be greater than or equal to the word width when configured for MSB First. The value written must be less than or equal to 31-word width when configured for LSB First.
7–0 Reserved	This read-only field is reserved and always has the value zero.

46.3.7 SAI Transmit Data Register (I2Sx_TDR)

Addresses: I2S0_TDR0 is 4002_F000h base + 20h offset = 4002_F020h

<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td colspan="32">0</td></tr>
<tr><td>W</td><td colspan="32">TDR[31:0]</td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

I2Sx_TDR*n* field descriptions

Field	Description
31–0 TDR[31:0]	Transmit Data Register The corresponding TCR3[TCE] bit must be set before accessing the channel's transmit data register. Writes to this register when the transmit FIFO is not full will push the data written into the transmit data FIFO. Writes to this register when the transmit FIFO is full are ignored.

46.3.8 SAI Transmit FIFO Register (I2Sx_TFR)

The MSB of the read and write pointers is used to distinguish between FIFO full and empty conditions. If the read and write pointers are identical, then the FIFO is empty. If the read and write pointers are identical except for the MSB, then the FIFO is full.

Addresses: I2S0_TFR0 is 4002_F000h base + 40h offset = 4002_F040h

<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td colspan="13">0</td><td colspan="3">WFP</td><td colspan="13">0</td><td colspan="3">RFP</td></tr>
<tr><td>W</td><td colspan="13"></td><td colspan="3"></td><td colspan="13"></td><td colspan="3"></td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

I2Sx_TFR*n* field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero.
18–16 WFP	Write FIFO Pointer FIFO write pointer for transmit data channel.
15–3 Reserved	This read-only field is reserved and always has the value zero.
2–0 RFP	Read FIFO Pointer FIFO read pointer for transmit data channel.

46.3.9 SAI Transmit Mask Register (I2Sx_TMR)

This register is double-buffered and updates:

1. When TCSR[TE] is first set
2. At the end of each frame.

This allows the masked words in each frame to change from frame to frame.

Addresses: I2S0_TMR is 4002_F000h base + 60h offset = 4002_F060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TWM															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_TMR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 TWM	Transmit Word Mask Configures whether the transmit word is masked for each word in the frame. 0 Word N is enabled. 1 Word N is masked. The transmit data pins are tri-stated when masked.

46.3.10 SAI Receive Control Register (I2Sx_RCSR)

Addresses: I2S0_RCSR is 4002_F000h base + 80h offset = 4002_F080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RE	STOPE	DBGE	BCE	0		0	SR	0			WSF	SEF	FEF	FWF	FRF
W							FR					w1c	w1c	w1c		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WSIE	SEIE	FEIE	FWIE	FRIE	0			0			FWDE	FRDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCSR field descriptions

Field	Description
31 RE	Receiver Enable Enables/disables the receiver. When software clears this field, the receiver remains enabled, and this bit remains set, until the end of the current frame. 0 Receiver is disabled. 1 Receiver is enabled, or receiver has been disabled and has not yet reached end of frame.
30 STOPE	Stop Enable Configures receiver operation in Stop mode. This bit is ignored and the receiver is disabled in all low-leakage stop modes. 0 Receiver disabled in Stop mode. 1 Receiver enabled in Stop mode.
29 DBGE	Debug Enable Enables/disables receiver operation in Debug mode. The receive bit clock is not affected by Debug mode. 0 Receiver is disabled in Debug mode, after completing the current frame. 1 Receiver is enabled in Debug mode.
28 BCE	Bit Clock Enable Enables the receive bit clock, separately from RE. This field is automatically set whenever RE is set. When software clears this field, the receive bit clock remains enabled, and this field remains set, until the end of the current frame. 0 Receive bit clock is disabled. 1 Receive bit clock is enabled.
27–26 Reserved	This read-only field is reserved and always has the value zero.
25 FR	FIFO Reset Resets the FIFO pointers. Reading this field will always return zero. 0 No effect. 1 FIFO reset.
24 SR	Software Reset Resets the internal receiver logic including the FIFO pointers. Software-visible registers are not affected, except for the status registers. 0 No effect. 1 Software reset.
23–21 Reserved	This read-only field is reserved and always has the value zero.
20 WSF	Word Start Flag Indicates that the start of the configured word has been detected. Write a logic 1 to this field to clear this flag.

Table continues on the next page...

I2Sx_RCSR field descriptions (continued)

Field	Description
	0 Start of word not detected. 1 Start of word detected.
19 SEF	Sync Error Flag Indicates that an error in the externally-generated frame sync has been detected. Write a logic 1 to this field to clear this flag. 0 Sync error not detected. 1 Frame sync error detected.
18 FEF	FIFO Error Flag Indicates that an enabled receive FIFO has overflowed. Write a logic 1 to this field to clear this flag. 0 Receive overflow not detected. 1 Receive overflow detected.
17 FWF	FIFO Warning Flag Indicates that an enabled receive FIFO is full. 0 No enabled receive FIFO is full. 1 Enabled receive FIFO is full.
16 FRF	FIFO Request Flag Indicates that the number of words in an enabled receive channel FIFO is greater than the receive FIFO watermark. 0 Receive FIFO watermark not reached. 1 Receive FIFO watermark has been reached.
15–13 Reserved	This read-only field is reserved and always has the value zero.
12 WSIE	Word Start Interrupt Enable Enables/disables word start interrupts. 0 Disables interrupt. 1 Enables interrupt.
11 SEIE	Sync Error Interrupt Enable Enables/disables sync error interrupts. 0 Disables interrupt. 1 Enables interrupt.
10 FEIE	FIFO Error Interrupt Enable Enables/disables FIFO error interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
9 FWIE	FIFO Warning Interrupt Enable

Table continues on the next page...

I2Sx_RCSR field descriptions (continued)

Field	Description
	Enables/disables FIFO warning interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
8 FRIE	FIFO Request Interrupt Enable Enables/disables FIFO request interrupts. 0 Disables the interrupt. 1 Enables the interrupt.
7–5 Reserved	This read-only field is reserved and always has the value zero.
4–2 Reserved	This read-only field is reserved and always has the value zero.
1 FWDE	FIFO Warning DMA Enable Enables/disables DMA requests. 0 Disables the DMA request. 1 Enables the DMA request.
0 FRDE	FIFO Request DMA Enable Enables/disables DMA requests. 0 Disables the DMA request. 1 Enables the DMA request.

46.3.11 SAI Receive Configuration 1 Register (I2Sx_RCR1)

Addresses: I2S0_RCR1 is 4002_F000h base + 84h offset = 4002_F084h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																														RFW	
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR1 field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value zero.
1–0 RFW	Receive FIFO Watermark Configures the watermark level for all enabled receiver channels.

46.3.12 SAI Receive Configuration 2 Register (I2Sx_RCR2)

This register must not be altered when RCSR[RE] is set.

Addresses: I2S0_RCR2 is 4002_F000h base + 88h offset = 4002_F088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SYNC		BCS	BCI	MSEL		BCP	BCD	0																DIV							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR2 field descriptions

Field	Description
31–30 SYNC	Synchronous Mode Configures between asynchronous and synchronous modes of operation. When configured for a synchronous mode of operation, the transmitter must be configured for asynchronous operation. 00 Asynchronous mode. 01 Synchronous with transmitter. 10 Synchronous with another SAI receiver. 11 Synchronous with another SAI transmitter.
29 BCS	Bit Clock Swap When the SAI is in asynchronous mode and this field is set to 1, the receiver is clocked by the transmitter bit clock. When the SAI is in synchronous mode and this field is set to 1, the receiver is clocked by the receiver bit clock, but it uses the transmitter frame sync. 0 Use the normal bit clock source. 1 Swap the bit clock source.
28 BCI	Bit Clock Input When set in either asynchronous or synchronous mode and the module is using an internally generated bit clock, configures the internal logic to be clocked as if the bit clock was externally generated. This has the effect of decreasing data input setup time, but increasing data output valid time. This bit has no effect when configured for an externally generated bit clock. 0 No effect. 1 Internal logic is clocked as if bit clock was externally generated.
27–26 MSEL	MCLK Select Selects the audio master clock used to generate an internally generated bit clock. This field has no effect when configured for an externally generated bit clock. 00 Bus clock selected. 01 Master clock 1 selected. 10 Master clock 2 selected. 11 Master clock 3 selected.

Table continues on the next page...

I2Sx_RCR2 field descriptions (continued)

Field	Description
25 BCP	Bit Clock Polarity Configures the polarity of the bit clock. 0 Bit Clock is active high with drive outputs on rising edge and sample inputs on falling edge. 1 Bit Clock is active low with drive outputs on falling edge and sample inputs on rising edge.
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0 Bit clock is generated externally in Slave mode. 1 Bit clock is generated internally in Master mode.
23–8 Reserved	This read-only field is reserved and always has the value zero.
7–0 DIV	Bit Clock Divide Divides down the audio master clock to generate the bit clock when configured for an internal bit clock. The division value is (DIV + 1) * 2.

46.3.13 SAI Receive Configuration 3 Register (I2Sx_RCR3)

This register must not be altered when RCSR[RE] is set.

Addresses: I2S0_RCR3 is 4002_F000h base + 8Ch offset = 4002_F08Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															RCE	0												WDFL			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR3 field descriptions

Field	Description
31–17 Reserved	This read-only field is reserved and always has the value zero.
16 RCE	Receive Channel Enable Enables a data channel for a receive operation. A channel should be enabled before its FIFO is accessed. 0 Receive data channel is disabled. 1 Receive data channel is enabled.
15–4 Reserved	This read-only field is reserved and always has the value zero.
3–0 WDFL	Word flag configuration

Table continues on the next page...

I2Sx_RCR3 field descriptions (continued)

Field	Description
	Configures which word the start of word flag is set. The value written should be one less than the word number (for example, write zero to configure for the first word in the frame). When configured to a value greater than the Frame Size field, then the start of word flag is never set.

46.3.14 SAI Receive Configuration 4 Register (I2Sx_RCR4)

This register must not be altered when RCSR[RE] is set.

Addresses: I2S0_RCR4 is 4002_F000h base + 90h offset = 4002_F090h

Bit	31–20	19–16	15–13	12–8	7–5	4	3	2	1	0
R	0	FRSZ	0	SYWD	0	MF	FSE	0	FSP	FSD
W										
Reset	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0	0 0 0	0 0 0 0 0	0 0 0	0	0	0	0	0

I2Sx_RCR4 field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value zero.
19–16 FRSZ	Frame Size Configures the number of words in each frame. The value written must be one less than the number of words in the frame. For example, write 0 for one word per frame. The maximum supported frame size is 16 words.
15–13 Reserved	This read-only field is reserved and always has the value zero.
12–8 SYWD	Sync Width Configures the length of the frame sync in number of bit clocks. The value written must be one less than the number of bit clocks. For example, write 0 for the frame sync to assert for one bit clock only. The sync width cannot be configured longer than the first word of the frame.
7–5 Reserved	This read-only field is reserved and always has the value zero.
4 MF	MSB First Specifies whether the LSB or the MSB is transmitted/received first. 0 LSB is transmitted/received first. 1 MSB is transmitted/received first.
3 FSE	Frame Sync Early 0 Frame sync asserts with the first bit of the frame. 1 Frame sync asserts one bit before the first bit of the frame.

Table continues on the next page...

I2Sx_RCR4 field descriptions (continued)

Field	Description
2 Reserved	This read-only field is reserved and always has the value zero.
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync. 0 Frame sync is active high. 1 Frame sync is active low.
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0 Frame Sync is generated externally in Slave mode. 1 Frame Sync is generated internally in Master mode.

46.3.15 SAI Receive Configuration 5 Register (I2Sx_RCR5)

This register must not be altered when RCSR[RE] is set.

Addresses: I2S0_RCR5 is 4002_F000h base + 94h offset = 4002_F094h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			WNW					0			W0W					0			FBT					0							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RCR5 field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero.
28–24 WNW	Word N Width Configures the number of bits in each word, for each word except the first in the frame. The value written must be one less than the number of bits per word. The value of WNW must be greater than or equal to the value of W0W even when there is only one word in each frame. Word width of less than 8 bits is not supported.
23–21 Reserved	This read-only field is reserved and always has the value zero.
20–16 W0W	Word 0 Width Configures the number of bits in the first word in each frame. The value written must be one less than the number of bits in the first word. Word width of less than 8 bits is not supported if there is only one word per frame.
15–13 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

I2Sx_RCR5 field descriptions (continued)

Field	Description
12–8 FBT	First Bit Shifted Configures the bit index for the first bit received for each word in the frame. If configured for MSB First, the index of the next bit received is one less than the current bit received. If configured for LSB First, the index of the next bit received is one more than the current bit received. The value written must be greater than or equal to the word width when configured for MSB First. The value written must be less than or equal to 31-word width when configured for LSB First.
7–0 Reserved	This read-only field is reserved and always has the value zero.

46.3.16 SAI Receive Data Register (I2Sx_RDR)

Reading this register introduces one additional peripheral clock wait state on each read.

Addresses: I2S0_RDR0 is 4002_F000h base + A0h offset = 4002_F0A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDR[31:0]																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RDR*n* field descriptions

Field	Description
31–0 RDR[31:0]	Receive Data Register The corresponding RCR3[RCE] bit must be set before accessing the channel's receive data register. Reads from this register when the receive FIFO is not empty will return the data from the top of the receive FIFO. Reads from this register when the receive FIFO is empty are ignored.

46.3.17 SAI Receive FIFO Register (I2Sx_RFR)

The MSB of the read and write pointers is used to distinguish between FIFO full and empty conditions. If the read and write pointers are identical, then the FIFO is empty. If the read and write pointers are identical except for the MSB, then the FIFO is full.

Addresses: I2S0_RFR0 is 4002_F000h base + C0h offset = 4002_F0C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0													WFP			0													RFP		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RFR*n* field descriptions

Field	Description
31–19 Reserved	This read-only field is reserved and always has the value zero.
18–16 WFP	Write FIFO Pointer FIFO write pointer for receive data channel.
15–3 Reserved	This read-only field is reserved and always has the value zero.
2–0 RFP	Read FIFO Pointer FIFO read pointer for receive data channel.

46.3.18 SAI Receive Mask Register (I2Sx_RMR)

This register is double-buffered and updates:

1. When RCSR[RE] is first set
2. At the end of each frame

This allows the masked words in each frame to change from frame to frame.

Addresses: I2S0_RMR is 4002_F000h base + E0h offset = 4002_F0E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																RWM															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_RMR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 RWM	Receive Word Mask For each word in the frame, configures whether the receive word is masked. 0 Word N is enabled. 1 Word N is masked.

46.3.19 SAI MCLK Control Register (I2Sx_MCR)

The MCLK Control Register (MCR) controls the clock source and direction of the audio master clock.

Addresses: I2S0_MCR is 4002_F000h base + 100h offset = 4002_F100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DUF	MOE	0				MICS		0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_MCR field descriptions

Field	Description
31 DUF	Divider Update Flag Provides the status of on-the-fly updates to the MCLK divider ratio. 0 MCLK divider ratio is not being updated currently. 1 MCLK divider ratio is updating on-the-fly. Further updates to the MCLK divider ratio are blocked while this flag remains set.
30 MOE	MCLK Output Enable Enables the MCLK divider and configures the MCLK signal pin as an output. When software clears this field, it remains set until the MCLK divider is fully disabled. 0 MCLK signal pin is configured as an input that bypasses the MCLK divider. 1 MCLK signal pin is configured as an output from the MCLK divider and the MCLK divider is enabled.
29–26 Reserved	This read-only field is reserved and always has the value zero.
25–24 MICS	MCLK Input Clock Select Selects the clock input to the MCLK divider. This field cannot be changed while the MCLK divider is enabled. See the chip configuration details for information about the connections to these inputs. 00 MCLK divider input clock 0 selected. 01 MCLK divider input clock 1 selected. 10 MCLK divider input clock 2 selected. 11 MCLK divider input clock 3 selected.
23–0 Reserved	This read-only field is reserved and always has the value zero.

46.3.20 SAI MCLK Divide Register (I2Sx_MDR)

The MCLK Divide Register (MDR) configures the MCLK divide ratio. Although the MDR can be changed when the MCLK divider clock is enabled, additional writes to the MDR are blocked while MCR[DUF] is set. Writes to the MDR when the MCLK divided clock is disabled do not set MCR[DUF].

Addresses: I2S0_MDR is 4002_F000h base + 104h offset = 4002_F104h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												FRACT								DIVIDE											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Sx_MDR field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value zero.
19–12 FRACT	MCLK Fraction Sets the MCLK divide ratio such that: MCLK output = MCLK input * ((FRACT + 1) / (DIVIDE + 1)). FRACT must be set equal or less than the value in the DIVIDE field.
11–0 DIVIDE	MCLK Divide Sets the MCLK divide ratio such that: MCLK output = MCLK input * ((FRACT + 1) / (DIVIDE + 1)). FRACT must be set equal or less than the value in the DIVIDE field.

46.4 Functional description

46.4.1 SAI clocking

The SAI clocks include:

- The audio master clock
- The bit clock
- The bus clock

46.4.1.1 Audio master clock

The audio master clock is used to generate the bit clock when the receiver or transmitter is configured for an internally generated bit clock. The transmitter and receiver can independently select between the bus clock and up to three audio master clocks to generate the bit clock.

Each SAI peripheral can control the input clock selection, pin direction and divide ratio of one audio master clock. The input clock selection and pin direction cannot be altered if an SAI module using that audio master clock has been enabled. The MCLK divide ratio can be altered while an SAI is using that master clock, although the change in the divide ratio takes several cycles. MCR[DUF] can be polled to determine when the divide ratio change has completed.

The audio master clock generation and selection is chip-specific. Refer to chip-specific clocking information about how the audio master clocks are generated. A typical implementation appears in the following figure.

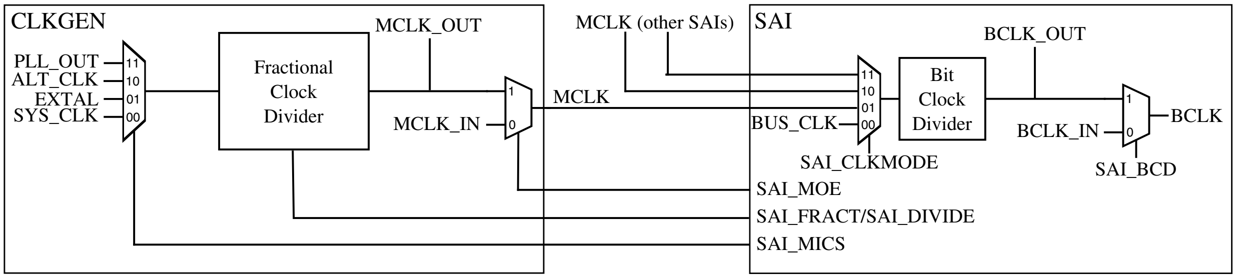

CLKGEN
PLL_OUT
ALT_CLK
EXTAL
SYS_CLK
11
10
01
00
Fractional Clock Divider
MCLK_OUT
MCLK_IN
1
0
MCLK
MCLK (other SAIs)
SAI
BUS_CLK
11
10
01
00
Bit Clock Divider
BCLK_OUT
BCLK_IN
1
0
BCLK
SAI_CLKMODE
SAI_BCD
SAI_MOE
SAI_FRACT/SAI_DIVIDE
SAI_MICS

Figure 46-50. SAI master clock generation

46.4.1.2 Bit clock

The SAI transmitter and receiver support asynchronous free-running bit clocks that can be generated internally from an audio master clock or supplied externally. There is also the option for synchronous bit clock and frame sync operation between the receiver and transmitter or between multiple SAI peripherals.

Externally generated bit clocks must be:

- Enabled before the SAI transmitter or receiver is enabled
- Disabled after the SAI transmitter or receiver is disabled and completes its current frames

46.4.1.3 Bus clock

The bus clock is used by the control and configuration registers and to generate synchronous interrupts and DMA requests.

46.4.2 SAI resets

The SAI is asynchronously reset on system reset. The SAI has a software reset and a FIFO reset.

46.4.2.1 Software reset

The SAI transmitter includes a software reset that resets all transmitter internal logic, including the bit clock generation, status flags, and FIFO pointers. It does not reset the configuration registers. The software reset remains asserted until cleared by software.

The SAI receiver includes a software reset that resets all receiver internal logic, including the bit clock generation, status flags and FIFO pointers. It does not reset the configuration registers. The software reset remains asserted until cleared by software.

46.4.2.2 FIFO reset

The SAI transmitter includes a FIFO reset that synchronizes the FIFO write pointer to the same value as the FIFO read pointer. This empties the FIFO contents and is to be used after TCSR[FEF] is set, and before the FIFO is re-initialized and TCSR[FEF] is cleared. The FIFO reset is asserted for one cycle only.

The SAI receiver includes a FIFO reset that synchronizes the FIFO read pointer to the same value as the FIFO write pointer. This empties the FIFO contents and is to be used after the RCSR[FEF] is set and any remaining data has been read from the FIFO, and before the RCSR[FEF] is cleared. The FIFO reset is asserted for one cycle only.

46.4.3 Synchronous modes

The SAI transmitter and receiver can operate synchronously to each other.

46.4.3.1 Synchronous mode

The SAI transmitter and receiver can be configured to operate with synchronous bit clock and frame sync.

If the transmitter bit clock and frame sync are to be used by both the transmitter and receiver:

- The transmitter must be configured for asynchronous operation and the receiver for synchronous operation.
- In synchronous mode, the receiver is enabled only when both the transmitter and receiver are enabled.
- It is recommended that the transmitter is the last enabled and the first disabled.

If the receiver bit clock and frame sync are to be used by both the transmitter and receiver:

- The receiver must be configured for asynchronous operation and the transmitter for synchronous operation.
- In synchronous mode, the transmitter is enabled only when both the receiver and transmitter are both enabled.
- It is recommended that the receiver is the last enabled and the first disabled.

When operating in synchronous mode, only the bit clock, frame sync, and transmitter/receiver enable are shared. The transmitter and receiver otherwise operate independently, although configuration registers must be configured consistently across both the transmitter and receiver.

46.4.4 Frame sync configuration

The frame sync signal is used to indicate the start of each frame. A valid frame sync requires a rising edge (if active high) or falling edge (if active low) to be detected and the transmitter or receiver cannot be busy with a previous frame. A valid frame sync is also ignored (slave mode) or not generated (master mode) for the first four bit clock cycles after enabling the transmitter or receiver.

The transmitter and receiver frame sync can be configured independently with any of the following options:

- Externally generated or internally generated
- Active high or active low
- Assert with the first bit in frame or asserts one bit early
- Assert for a duration between 1 bit clock and the first word length

- Frame length from 1 to 16 words per frame
- Word length to support 8 to 32 bits per word
  - First word length and remaining word lengths can be configured separately
- Can be configured for MSB first or LSB first

These configuration options cannot be changed after the SAI transmitter or receiver is enabled.

46.4.5 Data FIFO

46.4.5.1 Data alignment

Each transmit and receive channel includes a FIFO of size 4 × 32-bit. The FIFO data is accessed using the SAI Transmit/Receive Data Registers. Data in the FIFO can be aligned anywhere within the 32-bit wide register through the use of the First Bit Shifted configuration field, which selects the bit index (between 31 and 0) of the first bit shifted.

Examples of supported data alignment and the required First Bit Shifted configuration are illustrated in Figure 46-51 for LSB First configurations and Figure 46-52 for MSB First configurations.

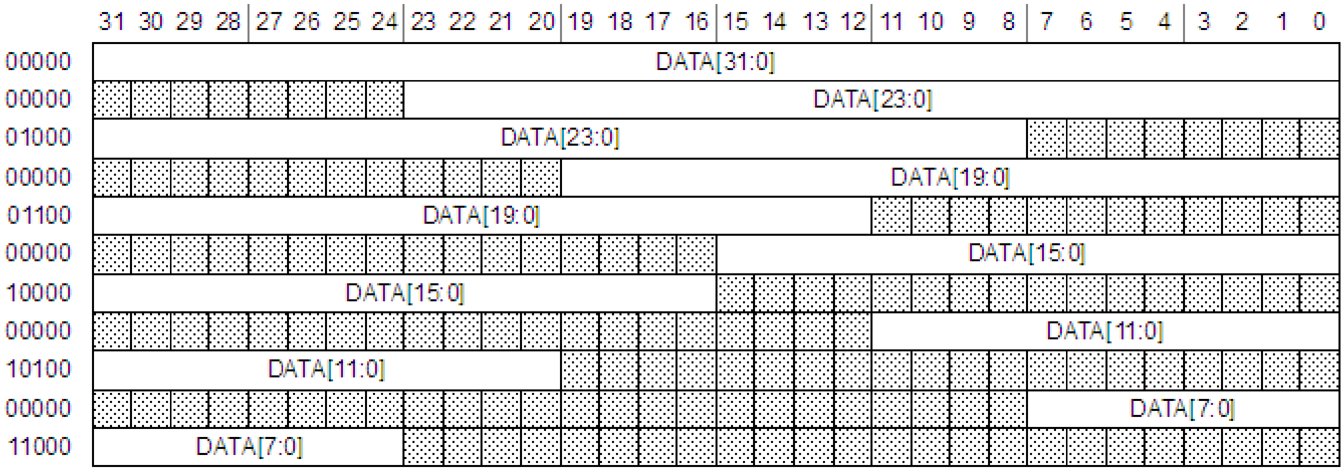

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
00000 DATA[31:0]
00000 DATA[23:0]
01000 DATA[23:0]
00000 DATA[19:0]
01100 DATA[19:0]
00000 DATA[15:0]
10000 DATA[15:0]
00000 DATA[11:0]
10100 DATA[11:0]
00000 DATA[7:0]
11000 DATA[7:0]

Figure 46-51. SAI first bit shifted, LSB first

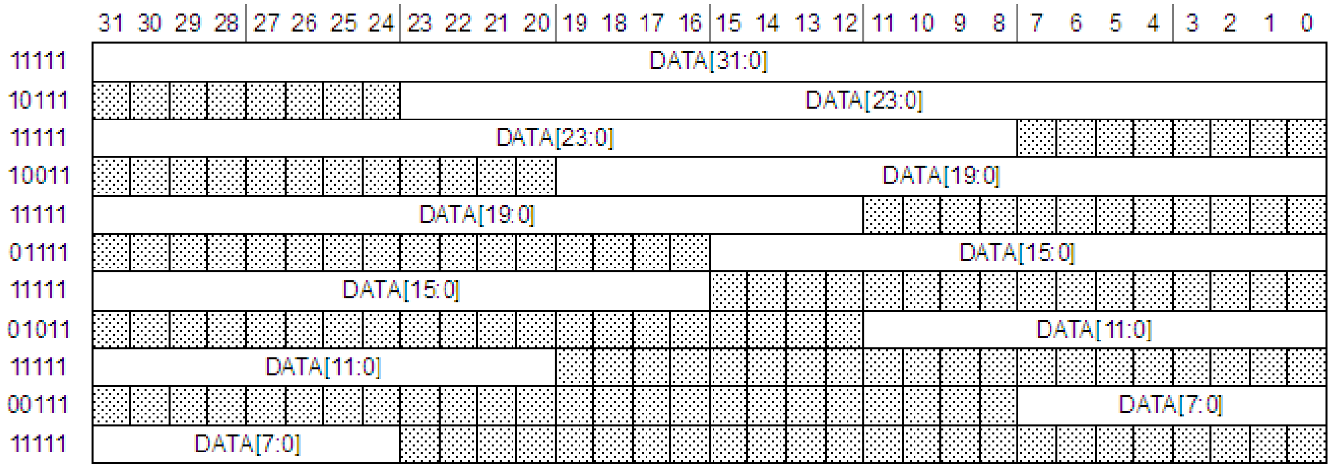

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
11111
DATA[31:0]
10111
DATA[23:0]
11111
DATA[23:0]
10011
DATA[19:0]
11111
DATA[19:0]
01111
DATA[15:0]
11111
DATA[15:0]
01011
DATA[11:0]
11111
DATA[11:0]
00111
DATA[7:0]
11111
DATA[7:0]

Figure 46-52. SAI first bit shifted, MSB first

46.4.5.2 FIFO pointers

When writing to a TDR, the WFP of the corresponding TFR increments after each valid write. The SAI supports 8-bit and 16-bit writes to TDR for transmitting 8-bit and 16-bit data respectively.

Writes to a TDR are ignored if the corresponding bit of TCR3[TCE] is clear or if the FIFO is full. If the Transmit FIFO is empty, the TDR must be written at least three bit clocks before the start of the next unmasked word to avoid a FIFO underrun.

When reading an RDR, the RFP of the corresponding RFR increments after each valid read. The SAI supports 8-bit and 16-bit reads from RDR for receiving 8-bit and 16-bit data respectively.

Reads from an RDR are ignored if the corresponding bit of RCR3[RCE] is clear or if the FIFO is empty. If the Receive FIFO is full, the RDR must be read at least three bit clocks before the end of an unmasked word to avoid a FIFO overrun.

46.4.6 Word mask register

The SAI transmitter and receiver each contain a word mask register, namely TMR and RMR, that can be used to mask any word in the frame. Because the word mask register is double buffered, software can update it before the end of each frame to mask a particular word in the next frame.

The TMR causes the Transmit Data pin to be tri-stated for the length of each selected word and the transmit FIFO is not read for masked words.

The RMR causes the received data for each selected word to be discarded and not written to the receive FIFO.

46.4.7 Interrupts and DMA requests

The SAI transmitter and receiver generate separate interrupts and separate DMA requests, but support the same status flags. Asynchronous versions of the transmitter and receiver interrupts are generated to wake up the CPU from stop mode.

46.4.7.1 FIFO data ready flag

The FIFO data ready flag is set based on the number of entries in the FIFO and the FIFO watermark configuration.

The transmit data ready flag is set when the number of entries in any of the enabled transmit FIFOs is less than or equal to the transmit FIFO watermark configuration and is cleared when the number of entries in each enabled transmit FIFO is greater than the transmit FIFO watermark configuration.

The receive data ready flag is set when the number of entries in any of the enabled receive FIFOs is greater than the receive FIFO watermark configuration and is cleared when the number of entries in each enabled receive FIFO is less than or equal to the receive FIFO watermark configuration.

The FIFO data ready flag can generate an interrupt or a DMA request.

46.4.7.2 FIFO warning flag

The FIFO warning flag is set based on the number of entries in the FIFO.

The transmit warning flag is set when the number of entries in any of the enabled transmit FIFOs is empty and is cleared when the number of entries in each enabled transmit FIFO is not empty.

The receive warning flag is set when the number of entries in any of the enabled receive FIFOs is full and is cleared when the number of entries in each enabled receive FIFO is not full.

The FIFO warning flag can generate an Interrupt or a DMA request.

46.4.7.3 FIFO error flag

The transmit FIFO error flag is set when the any of the enabled transmit FIFOs underflow. After it is set, all enabled transmit channels repeat the last valid word read from the transmit FIFO until TCSR[FEF] is cleared and the next transmit frame starts. All enabled transmit FIFOs must be reset and initialized with new data before TCSR[FEF] is cleared.

RCSR[FEF] is set when the any of the enabled receive FIFOs overflow. After it is set, all enabled receive channels discard received data until RCSR[FEF] is cleared and the next next receive frame starts. All enabled receive FIFOs should be emptied before RCSR[FEF] is cleared.

The FIFO error flag can generate only an interrupt.

46.4.7.4 Sync error flag

The sync error flag, TCSR[SEF] or RCSR[SEF], is set when configured for an externally generated frame sync and the external frame sync asserts when the transmitter or receiver is busy with the previous frame. The external frame sync assertion is ignored and the sync error flag is set. When the sync error flag is set, the transmitter or receiver continues checking for frame sync assertion when idle or at the end of each frame.

The sync error flag can generate an interrupt only.

46.4.7.5 Word start flag

The word start flag is set at the start of the second bit clock for the selected word, as configured by the Word Flag register field.

The word start flag can generate an interrupt only.

Chapter 47
General-Purpose Input/Output (GPIO)

47.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The general-purpose input and output (GPIO) module communicates to the processor core via a zero wait state interface for maximum pin performance. The GPIO registers support 8-bit, 16-bit or 32-bit accesses.

The GPIO data direction and output data registers control the direction and output data of each pin when the pin is configured for the GPIO function. The GPIO input data register displays the logic value on each pin when the pin is configured for any digital function, provided the corresponding Port Control and Interrupt module for that pin is enabled.

Efficient bit manipulation of the general-purpose outputs is supported through the addition of set, clear, and toggle write-only registers for each port output data register.

47.1.1 Features

- Features of the GPIO module include:
  - Pin input data register visible in all digital pin-multiplexing modes
  - Pin output data register with corresponding set/clear/toggle registers
  - Pin data direction register
  - Zero wait state access to GPIO registers

47.1.2 Modes of operation

The following table depicts different modes of operation and the behavior of the GPIO module in these modes.

Table 47-1. Modes of operation

Modes of operation	Description
Run	The GPIO module operates normally.
Wait	The GPIO module operates normally.
Stop	The GPIO module is disabled.
Debug	The GPIO module operates normally.

47.1.3 GPIO signal descriptions

Table 47-2. GPIO signal descriptions

Signal	Description	I/O
PORTA31–PORTA0	General-purpose input/output	I/O
PORTB31–PORTB0	General-purpose input/output	I/O
PORTC31–PORTC0	General-purpose input/output	I/O
PORTD31–PORTD0	General-purpose input/output	I/O
PORTE31–PORTE0	General-purpose input/output	I/O

NOTE

Not all pins within each port are implemented on each device. See the chapter on signal multiplexing for the number of GPIO ports available in the device.

47.1.3.1 Detailed signal description

Table 47-3. GPIO interface-detailed signal descriptions

Signal	I/O	Description	
PORTA31–PORTA0 PORTB31–PORTB0 PORTC31–PORTC0 PORTD31–PORTD0 PORTE31–PORTE0	I/O	General-purpose input/output	
		State meaning	Asserted: The pin is logic 1. Deasserted: The pin is logic 0.
		Timing	Assertion: When output, this signal occurs on the rising-edge of the system clock. For input, it may occur at any time and input may be asserted asynchronously to the system clock. Deassertion: When output, this signal occurs on the rising-edge of the system clock. For input, it may occur at any time and input may be asserted asynchronously to the system clock.

47.2 Memory map and register definition

Any read or write access to the GPIO memory space that is outside the valid memory map results in a bus error. All register accesses complete with zero wait states, except error accesses which complete with one wait state.

GPIO memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400F_F000	Port Data Output Register (GPIOA_PDOR)	32	R/W	0000_0000h	47.2.1/ 1182
400F_F004	Port Set Output Register (GPIOA_PSOR)	32	W (always reads zero)	0000_0000h	47.2.2/ 1182
400F_F008	Port Clear Output Register (GPIOA_PCOR)	32	W (always reads zero)	0000_0000h	47.2.3/ 1183

Table continues on the next page...

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400F_F00C	Port Toggle Output Register (GPIOA_PTOR)	32	W (always reads zero)	0000_0000h	47.2.4/ 1183
400F_F010	Port Data Input Register (GPIOA_PDIR)	32	R	0000_0000h	47.2.5/ 1184
400F_F014	Port Data Direction Register (GPIOA_PDDR)	32	R/W	0000_0000h	47.2.6/ 1185
400F_F040	Port Data Output Register (GPIOB_PDOR)	32	R/W	0000_0000h	47.2.1/ 1182
400F_F044	Port Set Output Register (GPIOB_PSOR)	32	W (always reads zero)	0000_0000h	47.2.2/ 1182
400F_F048	Port Clear Output Register (GPIOB_PCOR)	32	W (always reads zero)	0000_0000h	47.2.3/ 1183
400F_F04C	Port Toggle Output Register (GPIOB_PTOR)	32	W (always reads zero)	0000_0000h	47.2.4/ 1183
400F_F050	Port Data Input Register (GPIOB_PDIR)	32	R	0000_0000h	47.2.5/ 1184
400F_F054	Port Data Direction Register (GPIOB_PDDR)	32	R/W	0000_0000h	47.2.6/ 1185
400F_F080	Port Data Output Register (GPIOC_PDOR)	32	R/W	0000_0000h	47.2.1/ 1182
400F_F084	Port Set Output Register (GPIOC_PSOR)	32	W (always reads zero)	0000_0000h	47.2.2/ 1182
400F_F088	Port Clear Output Register (GPIOC_PCOR)	32	W (always reads zero)	0000_0000h	47.2.3/ 1183
400F_F08C	Port Toggle Output Register (GPIOC_PTOR)	32	W (always reads zero)	0000_0000h	47.2.4/ 1183
400F_F090	Port Data Input Register (GPIOC_PDIR)	32	R	0000_0000h	47.2.5/ 1184
400F_F094	Port Data Direction Register (GPIOC_PDDR)	32	R/W	0000_0000h	47.2.6/ 1185

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
400F_F0C0	Port Data Output Register (GPIOD_PDOR)	32	R/W	0000_0000h	47.2.1/ 1182
400F_F0C4	Port Set Output Register (GPIOD_PSOR)	32	W (always reads zero)	0000_0000h	47.2.2/ 1182
400F_F0C8	Port Clear Output Register (GPIOD_PCOR)	32	W (always reads zero)	0000_0000h	47.2.3/ 1183
400F_F0CC	Port Toggle Output Register (GPIOD_PTOR)	32	W (always reads zero)	0000_0000h	47.2.4/ 1183
400F_F0D0	Port Data Input Register (GPIOD_PDIR)	32	R	0000_0000h	47.2.5/ 1184
400F_F0D4	Port Data Direction Register (GPIOD_PDDR)	32	R/W	0000_0000h	47.2.6/ 1185
400F_F100	Port Data Output Register (GPIOE_PDOR)	32	R/W	0000_0000h	47.2.1/ 1182
400F_F104	Port Set Output Register (GPIOE_PSOR)	32	W (always reads zero)	0000_0000h	47.2.2/ 1182
400F_F108	Port Clear Output Register (GPIOE_PCOR)	32	W (always reads zero)	0000_0000h	47.2.3/ 1183
400F_F10C	Port Toggle Output Register (GPIOE_PTOR)	32	W (always reads zero)	0000_0000h	47.2.4/ 1183
400F_F110	Port Data Input Register (GPIOE_PDIR)	32	R	0000_0000h	47.2.5/ 1184
400F_F114	Port Data Direction Register (GPIOE_PDDR)	32	R/W	0000_0000h	47.2.6/ 1185

47.2.1 Port Data Output Register (GPIOx_PDOR)

This register configures the logic levels that are driven on each general-purpose output pins.

Addresses: GPIOA_PDOR is 400F_F000h base + 0h offset = 400F_F000h
GPIOB_PDOR is 400F_F040h base + 0h offset = 400F_F040h
GPIOC_PDOR is 400F_F080h base + 0h offset = 400F_F080h
GPIOD_PDOR is 400F_F0C0h base + 0h offset = 400F_F0C0h
GPIOE_PDOR is 400F_F100h base + 0h offset = 400F_F100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PDO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PDOR field descriptions

Field	Description
31–0 PDO	Port Data Output Unimplemented pins for a particular device read as zero. 0 Logic level 0 is driven on pin, provided pin is configured for general-purpose output. 1 Logic level 1 is driven on pin, provided pin is configured for general-purpose output.

47.2.2 Port Set Output Register (GPIOx_PSOR)

This register configures whether to set the fields of the PDOR.

Addresses: GPIOA_PSOR is 400F_F000h base + 4h offset = 400F_F004h
GPIOB_PSOR is 400F_F040h base + 4h offset = 400F_F044h
GPIOC_PSOR is 400F_F080h base + 4h offset = 400F_F084h
GPIOD_PSOR is 400F_F0C0h base + 4h offset = 400F_F0C4h
GPIOE_PSOR is 400F_F100h base + 4h offset = 400F_F104h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																															
W	PTSO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PSOR field descriptions

Field	Description
31–0 PTSO	Port Set Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows:

GPIOx_PSOR field descriptions (continued)

Field	Description
	0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to logic 1.

47.2.3 Port Clear Output Register (GPIOx_PCOR)

This register configures whether to clear the fields of PDOR.

Addresses: GPIOA_PCOR is 400F_F000h base + 8h offset = 400F_F008h
GPIOB_PCOR is 400F_F040h base + 8h offset = 400F_F048h
GPIOC_PCOR is 400F_F080h base + 8h offset = 400F_F088h
GPIOD_PCOR is 400F_F0C0h base + 8h offset = 400F_F0C8h
GPIOE_PCOR is 400F_F100h base + 8h offset = 400F_F108h

<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td colspan="32">0</td></tr>
<tr><td>W</td><td colspan="32">PTCO</td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

GPIOx_PCOR field descriptions

Field	Description
31–0 PTCO	Port Clear Output Writing to this register will update the contents of the corresponding bit in the Port Data Output Register (PDOR) as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is cleared to logic 0.

47.2.4 Port Toggle Output Register (GPIOx_PTOR)

Addresses: GPIOA_PTOR is 400F_F000h base + Ch offset = 400F_F00Ch
GPIOB_PTOR is 400F_F040h base + Ch offset = 400F_F04Ch
GPIOC_PTOR is 400F_F080h base + Ch offset = 400F_F08Ch
GPIOD_PTOR is 400F_F0C0h base + Ch offset = 400F_F0CCh
GPIOE_PTOR is 400F_F100h base + Ch offset = 400F_F10Ch

<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td colspan="32">0</td></tr>
<tr><td>W</td><td colspan="32">PTTO</td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

GPIOx_PTOR field descriptions

Field	Description
31–0 PTTO	Port Toggle Output Writing to this register will update the contents of the corresponding bit in the PDOR as follows: 0 Corresponding bit in PDORn does not change. 1 Corresponding bit in PDORn is set to the inverse of its existing logic state.

47.2.5 Port Data Input Register (GPIOx_PDIR)

Addresses: GPIOA_PDIR is 400F_F000h base + 10h offset = 400F_F010h
GPIOB_PDIR is 400F_F040h base + 10h offset = 400F_F050h
GPIOC_PDIR is 400F_F080h base + 10h offset = 400F_F090h
GPIOD_PDIR is 400F_F0C0h base + 10h offset = 400F_F0D0h
GPIOE_PDIR is 400F_F100h base + 10h offset = 400F_F110h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PDI																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PDIR field descriptions

Field	Description
31–0 PDI	Port Data Input Reads 0 at the unimplemented pins for a particular device. Pins that are not configured for a digital function read 0. If the Port Control and Interrupt module is disabled, then the corresponding bit in PDIR does not update. 0 Pin logic level is logic 0, or is not configured for use by digital function. 1 Pin logic level is logic 1.

47.2.6 Port Data Direction Register (GPIOx_PDDR)

The PDDR configures the individual port pins for input or output.

Addresses: GPIOA_PDDR is 400F_F000h base + 14h offset = 400F_F014h
GPIOB_PDDR is 400F_F040h base + 14h offset = 400F_F054h
GPIOC_PDDR is 400F_F080h base + 14h offset = 400F_F094h
GPIOD_PDDR is 400F_F0C0h base + 14h offset = 400F_F0D4h
GPIOE_PDDR is 400F_F100h base + 14h offset = 400F_F114h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PDD																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PDDR field descriptions

Field	Description
31–0 PDD	Port Data Direction Configures individual port pins for input or output. 0 Pin is configured as general-purpose input, for the GPIO function. 1 Pin is configured as general-purpose output, for the GPIO function.

47.3 Functional description

47.3.1 General-purpose input

The logic state of each pin is available via the pin data input registers, provided the pin is configured for a digital function and the corresponding Port Control and Interrupt module is enabled.

The Pin Data Input registers return the synchronized pin state after any enabled digital filter in the Port Control and Interrupt module. The input pin synchronizers are shared with the Port Control and Interrupt module, so that if the corresponding Port Control and Interrupt module is disabled, then synchronizers are also disabled. This reduces power consumption when a port is not required for general-purpose input functionality.

47.3.2 General-purpose output

The logic state of each pin can be controlled via the pin data output registers and port data direction registers, provided the pin is configured for the GPIO function. The following table depicts the conditions for a pin to be configured as input/output.

If	Then
A pin is configured for the GPIO function and the corresponding data output enable register bit is clear.	The pin is configured as an input.
A pin is configured for the GPIO function and the corresponding pin data output enable register bit is set.	The pin is configured as an output and and the logic state of the pin is equal to the corresponding pin data output register.

To facilitate efficient bit manipulation on the general-purpose outputs, pin data set, pin data clear, and pin data toggle registers exist to allow one or more outputs within one port to be set, cleared, or toggled from a single register write.

The corresponding Port Control and Interrupt module does not need to be enabled to update the state of the pin output enable registers and pin data output registers including the set/clear/toggle registers.

Chapter 48
Touch sense input (TSI)

48.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The touch sensing input (TSI) module provides capacitive touch sensing detection with high sensitivity and enhanced robustness. Each TSI pin implements the capacitive measurement of an electrode having individual programmable detection thresholds and result registers. The TSI module can be functional in several low power modes with ultra low current adder and waking up the CPU in a touch event. It provides a solid capacitive measurement module to the implementation of touch keypad, rotaries and sliders.

48.2 Features

TSI module features included:

- Support as many as 16 input capacitive touch sensing pins with individual result registers
- Automatic detection of electrode capacitance change in low power mode with programmable upper and lower threshold
- Automatic periodic scan unit with different duty cycles for run and low power modes
- Fully support with FSL touch sensing SW library suite the implementation of keypads, rotaries and sliders
- Operation across all low power modes: WAIT,STOP, VLPR, VLPW, VLPS, LLS,VLLS{3,2,1}
- Capability to wake up MCU from low power modes.
- Configurable interrupts:
  - End-of-scan or out-of-range interrupt
  - TSI error interrupts: pad short to V_{DD}/V_{SS} or conversion overrun

- Compensate temperature and supply voltage variations
- Stand alone operation not requiring any external crystal even in low power modes
- Configurable integration of each electrode capacitance measurement from 1 to 4096 periods
- Programmable Electrode Oscillator and TSI Reference Oscillator allowing high sensitivity, small scan time and low power functionality.
- Only uses one pin per electrode implementation with no external hardware required

48.3 Overview

This section presents an overview of the TSI module. The following figure presents the simplified TSI module block diagram.

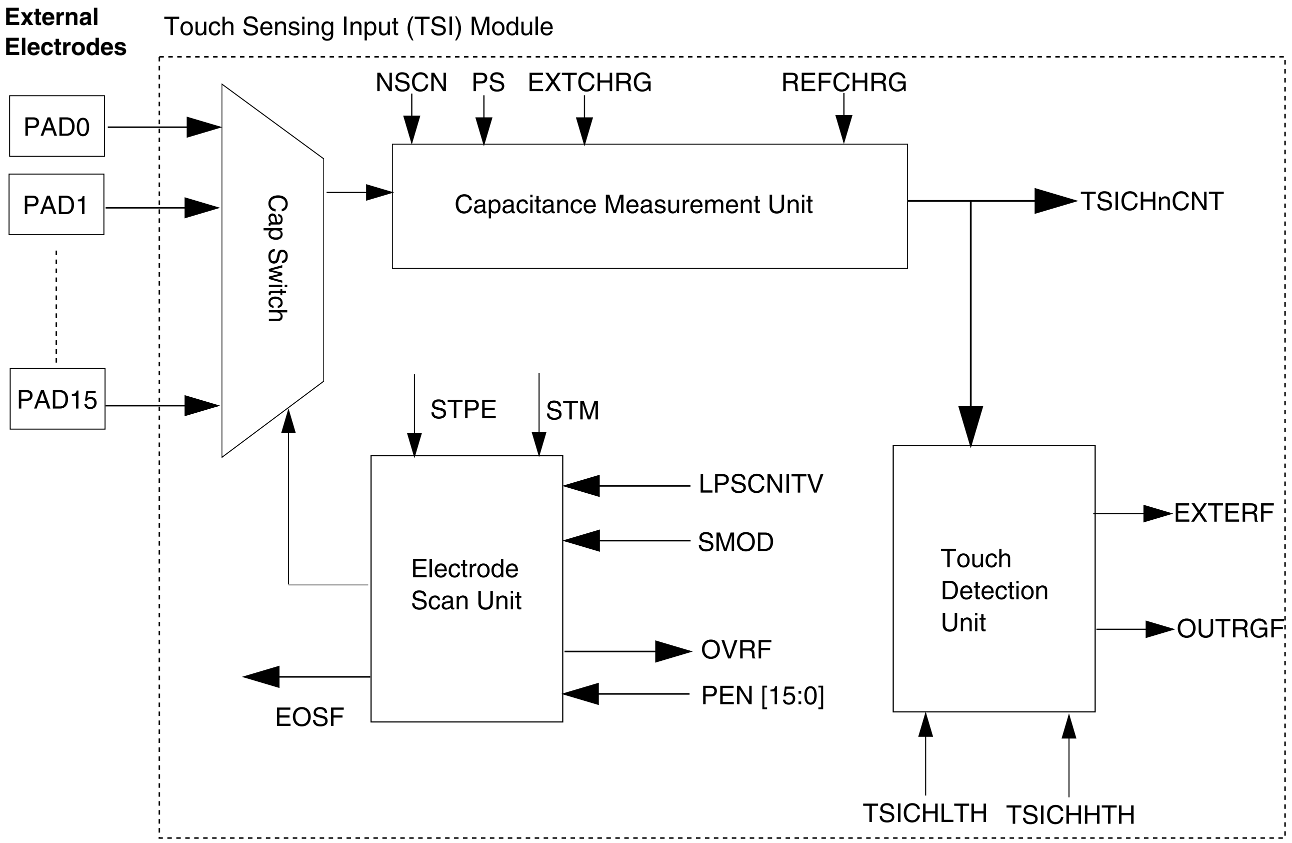

External Electrodes
Touch Sensing Input (TSI) Module
PAD0
PAD1
PAD15
Cap Switch
NSCN
PS
EXTCHRG
REFCHRG
Capacitance Measurement Unit
TSICHnCNT
STPE
STM
Electrode Scan Unit
LPSCNITV
SMOD
OVRF
PEN [15:0]
EOSF
Touch Detection Unit
EXTERF
OUTRGF
TSICHLTH
TSICHHTH

Figure 48-1. Touch sensing input block diagram

48.3.1 Electrode capacitance measurement unit

The electrode capacitance measurement unit senses the capacitance of a TSI pin and outputs a 16-bit result. This module is based in dual oscillator architecture. One oscillator is connected to the external electrode array and oscillates according to the electrode capacitance, while the other according to an internal reference capacitor. The pin capacitance measurement is given by the counted number of periods of the reference oscillator during a pre-defined number of electrode oscillations.

The electrode oscillator charges and discharges the pin capacitance with a programmable current source in order to accommodate several different sizes of electrode capacitances. The electrode oscillator frequency, before being compared to that of the reference oscillator, goes through a prescaler and module counter to decrease its frequency and consecutively increase the measurement resolution and noise robustness.

The following figure presents the simplified block diagram of how the electrode capacitance is measured.

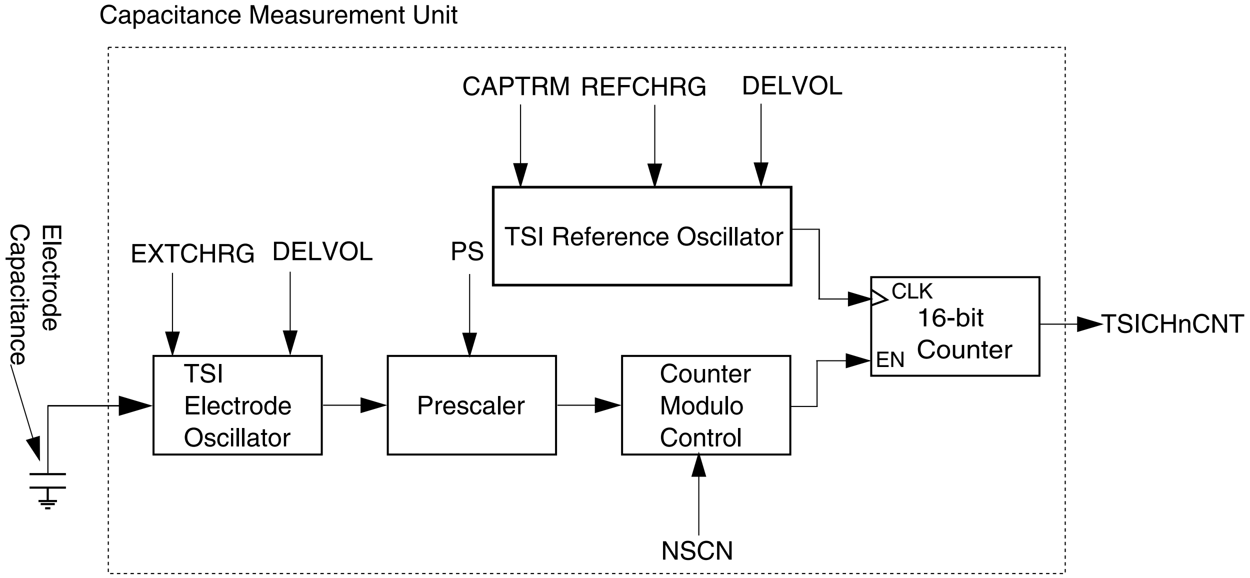

Capacitance Measurement Unit
CAPTRM
REFCHRG
DELVOL
Electrode Capacitance
EXTCHRG
DELVOL
PS
TSI Reference Oscillator
CLK
16-bit Counter
EN
TSICHnCNT
TSI Electrode Oscillator
Prescaler
Counter Modulo Control
NSCN

Figure 48-2. TSI capacitance measurement unit block diagram

48.3.2 Electrode scan unit

This session describes the functionality of the electrode scan unit. It is responsible for triggering the start of the active electrode scan.

The touch sense input module needs to periodically scan all active electrodes to determine if a touch event has occurred. The electrode scan unit is responsible for defining two independent scan periods, one for TSI active mode and the other for TSI low power mode. This independent control allows the application to configure longer scan period during low power mode, so contributing to smaller average power consumption. The TSI, in low power mode, has the capability to wake up the CPU upon an electrode capacitance change. When the CPU wakes, the TSI enters active mode, and a shorter scan period can provide a faster response time and more robust touch detection. Apart from the periodical mode, the electrode scan unit also allows software triggering of the electrode scans. This feature is very useful for initialization of the touch application for detecting the initial electrode capacitances. This module generates configurable end-of-scan interrupt to indicate the application that all electrodes were scanned. In the event starting a new electrode scan while a previous one is still in progress an overrun error flag is generated.

48.3.3 Touch detection unit

The touch detection unit indicates any change in the low power electrode pin capacitance. The purpose of this module is to only wake up th CPU from low power modes in the event of a electrode capacitance change. So, if there is no capacitance change in the electrode, the MCU stays in low power mode indefinitely, while keeping the electrode monitoring, ensuring minimal power consumption.

This module compares the pin capacitance value in the result register with a pre-configured low and high threshold. If the capacitance result register value is outside the ranges defined by upper and lower threshold the touch detection unit generates an out-of-range flag indicating a pin capacitance change.

The upper and lower threshold values are configurable allowing the application to select the magnitude of the capacitance change to trigger the out-of-range flag. With the threshold values programmed properly, the application noise level does not cause frequent CPU interrupts, so minimizes the CPU usage.

48.4 Modes of operation

The TSI module has three operation modes: disabled, active mode and low power mode.

Table 48-1. TSI Module funtionality in MCU operation modes

MCU operation mode	TSI clock sources	TSI operation mode when TSIEN = 1	Functional electrode pins	Required STPE state
Run	LPOCLK, MSGIRCLK, OSCERCLK	Active mode	All	Don’t care
Wait	LPOCLK, MSGIRCLK, OSCERCLK	Active mode	All	Don’t care

Table continues on the next page...

Table 48-1. TSI Module funtionality in MCU operation modes (continued)

MCU operation mode	TSI clock sources	TSI operation mode when TSIEN = 1	Functional electrode pins	Required STPE state
Stop	LPOCLK, MSGIRCLK, OSCERCLK	Active mode	All	1
VLPRun	LPOCLK, MSGIRCLK, OSCERCLK	Active mode	All	Don't care
VLPWait	LPOCLK, MSGIRCLK, OSCERCLK	Active mode	All	Don't care
VLPStop	LPOCLK MSGIRCLK, OSCERCLK	Active mode	All	1
LLS	LPOCLK, VLPOSCCLK	Low power mode	Determined by PEN[LPSP]	1
VLLS3	LPOCLK, VLPOSCCLK	Low power mode	Determined by PEN[LPSP]	1
VLLS2	LPOCLK, VLPOSCCLK	Low power mode	Determined by PEN[LPSP]	1
VLLS1	LPOCLK, VLPOSCCLK	Low power mode	Determined by PEN[LPSP]	1

48.4.1 TSI disabled mode

When GENCS[TSIEN] is cleared, the TSI module is disabled, and does not perform any functionally in any MCU operation mode.

48.4.2 TSI active mode

In active mode, the TSI module has its full functionality, being able to scan up to 16 electrodes. The TSI can be in active mode with the MCU in any of the following operational modes: run, wait, stop, VLPR, VLPW and VLPS.

Three clocks sources can be selected for the TSI module in active mode: LPOCLK, MCGIRCLK and OSCERCLK.

48.4.3 TSI low power mode

The TSI modules enters in low power mode if the GENCS[STPE] is set to one and the MCU enters in one of the following operational modes: LLS, VLLS1, VLLS2 and VLLS3. In low power mode, only one selectable pin is active, being able to perform capacitance measurements. The scan period is defined by GENCS[LPSCNITV] . Two low power clock sources are available in the TSI low power mode, LPOCLK and VLPOSCCLK, being selected by the GENCS[LPCLKS].

In low power mode the TSI interrupt can also be configured as end-of-scan or out-of-range and the GENCS[TSIIEN] must be set in order to generate these interrupts. The TSI interrupt causes the exit of the low power mode and entrance in the active mode, and the MCU also wakes up.

In low power mode the electrode scan unit is always configured to periodical low power scan.

48.4.4 Block diagram

The following figure shows the block diagram of TSI module.[1]

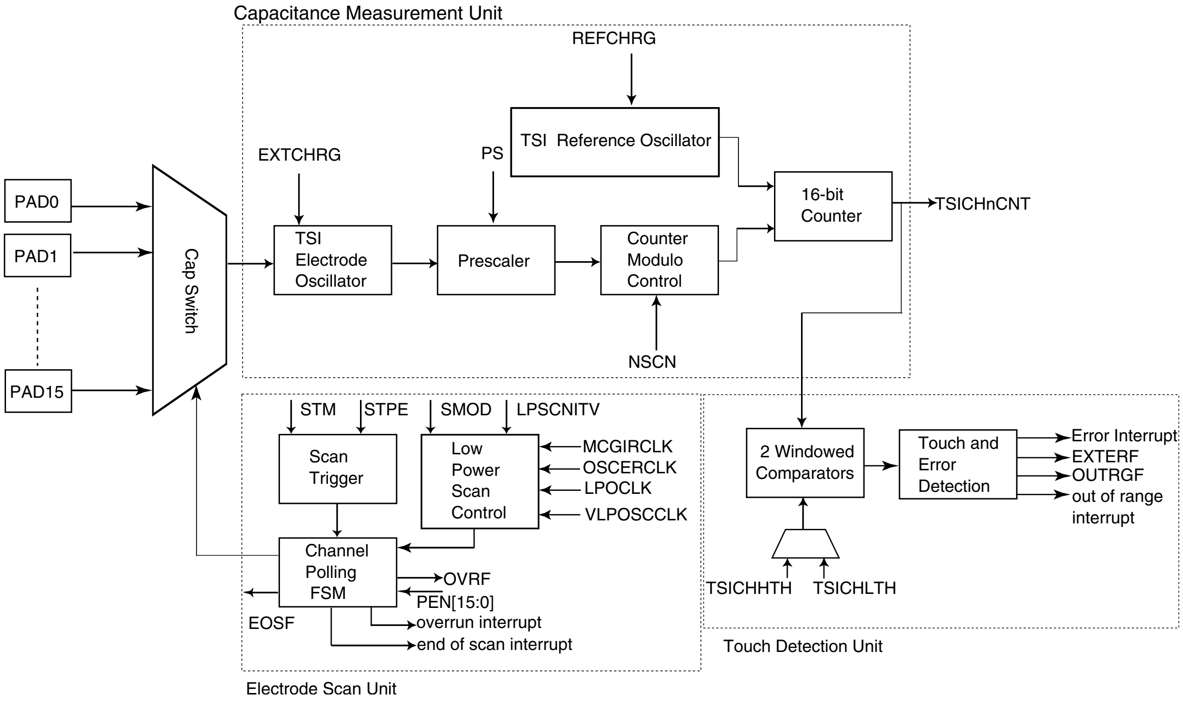

Capacitance Measurement Unit
REFCHRG
TSI Reference Oscillator
EXTCHRG
PS
16-bit Counter
TSICHnCNT
PAD0
PAD1
PAD15
Cap Switch
TSI Electrode Oscillator
Prescaler
Counter Modulo Control
NSCN
STM
STPE
SMOD
LPSCNITV
Scan Trigger
Low Power Scan Control
MCGIRCLK
OSCERCLK
LPOCLK
VLPOSCCLK
Channel Polling FSM
OVRF
PEN[15:0]
EOSF
overrun interrupt
end of scan interrupt
Electrode Scan Unit
2 Windowed Comparators
Touch and Error Detection
Error Interrupt
EXTERF
OUTRGF
out of range interrupt
TSICHHTH
TSICHLTH
Touch Detection Unit

Figure 48-3. TSI block diagram

---

[1] The out of range functinality present in the Touch Detection Unit is only available in low power modes.

48.5 TSI signal descriptions

The TSI module has up to 16 external pins for touch sensing. The table below itemizes all the TSI external pins.

Table 48-2. TSI signal descriptions

Signal	Description	I/O
TSI_IN[15:0]	TSI capacitive pins. Switchable driver that connects directly to the electrode pins TSI[15:0] can operate as GPIO pins	I/O

48.5.1 TSI_IN[15:0]

When TSI functionality is enabled by the PEN[PENn], the TSI analog portion uses corresponding TSI_IN[n] pin to connect the module with the external electrode. The connection between the pin and the touch pad must be kept as short as possible to reduce distribution capacity on board.

48.6 Memory map and register definition

This section presents the touch sensing input module memory map and registers definition.

TSI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_5000	General Control and Status Register (TSI0_GENCS)	32	R/W	0000_0000h	48.6.1/ 1195
4004_5004	SCAN Control Register (TSI0_SCANC)	32	R/W	0000_0000h	48.6.2/ 1198
4004_5008	Pin Enable Register (TSI0_PEN)	32	R/W	0000_0000h	48.6.3/ 1200
4004_500C	Wake-Up Channel Counter Register (TSI0_WUCNTR)	32	R/W	0000_0000h	48.6.4/ 1202
4004_5100	Counter Register (TSI0_CNTR1)	32	R	0000_0000h	48.6.5/ 1203

Table continues on the next page...

TSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4004_5104	Counter Register (TSI0_CNTR3)	32	R	0000_0000h	48.6.5/ 1203
4004_5108	Counter Register (TSI0_CNTR5)	32	R	0000_0000h	48.6.5/ 1203
4004_510C	Counter Register (TSI0_CNTR7)	32	R	0000_0000h	48.6.5/ 1203
4004_5110	Counter Register (TSI0_CNTR9)	32	R	0000_0000h	48.6.5/ 1203
4004_5114	Counter Register (TSI0_CNTR11)	32	R	0000_0000h	48.6.5/ 1203
4004_5118	Counter Register (TSI0_CNTR13)	32	R	0000_0000h	48.6.5/ 1203
4004_511C	Counter Register (TSI0_CNTR15)	32	R	0000_0000h	48.6.5/ 1203
4004_5120	Low Power Channel Threshold Register (TSI0_THRESHOLD)	32	R/W	0000_0000h	48.6.6/ 1203

48.6.1 General Control and Status Register (TSIx_GENCS)

Addresses: TSI0_GENCS is 4004_5000h base + 0h offset = 4004_5000h

Bit	31	30	29	28	27	26	25	24
R	0			LPCLKS	LPSCNITV			
W								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
R	NSCN					PS		
W								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
R	EOSF	OUTRGF	EXTERF	OVRF	0		SCNIP	
W	w1c	w1c	w1c	w1c				SWTS
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
R	TSIEN	TSIIE	ERIE	ESOR	0	Reserved	STM	STPE
W								
Reset	0	0	0	0	0	0	0	0

TSIx_GENCS field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value zero.
28 LPCLKS	Low Power Mode Clock Source Selection. This bit-field can only be changed if the TSI module is disabled (TSIEN bit = 0). 0 LPOCLK is selected to determine the scan period in low power mode 1 VLPOSCCLK is selected to determine the scan period in low power mode
27–24 LPSCNITV	TSI Low Power Mode Scan Interval. This bit-field can only be changed if the TSI module is disabled (TSIEN bit = 0). 0000 1 ms scan interval 0001 5 ms scan interval

Table continues on the next page...

TSIx_GENCS field descriptions (continued)

Field	Description
	0010 10 ms scan interval 0011 15 ms scan interval 0100 20 ms scan interval 0101 30 ms scan interval 0110 40 ms scan interval 0111 50 ms scan interval 1000 75 ms scan interval 1001 100 ms scan interval 1010 125 ms scan interval 1011 150 ms scan interval 1100 200 ms scan interval 1101 300 ms scan interval 1110 400 ms scan interval 1111 500 ms scan interval
23–19 NSCN	Number of Consecutive Scans per Electrode electrode. This bit-field can only be changed if the TSI module is disabled (TSIEN bit = 0). 00000 Once per electrode 00001 Twice per electrode 00010 3 times per electrode 00011 4 times per electrode 00100 5 times per electrode 00101 6 times per electrode 00110 7 times per electrode 00111 8 times per electrode 01000 9 times per electrode 01001 10 times per electrode 01010 11 times per electrode 01011 12 times per electrode 01100 13 times per electrode 01101 14 times per electrode 01110 15 times per electrode 01111 16 times per electrode 10000 17 times per electrode 10001 18 times per electrode 10010 19 times per electrode 10011 20 times per electrode 10100 21 times per electrode 10101 22 times per electrode 10110 23 times per electrode 10111 24 times per electrode 11000 25 times per electrode 11001 26 times per electrode 11010 27 times per electrode 11011 28 times per electrode 11100 29 times per electrode

Table continues on the next page...

TSIx_GENCS field descriptions (continued)

Field	Description
	11101 30 times per electrode 11110 31 times per electrode 11111 32 times per electrode
18–16 PS	Electrode Oscillator prescaler. . This bit-field can only be changed if the TSI module is disabled (TSIEN bit = 0) 000 Electrode Oscillator Frequency divided by 1 001 Electrode Oscillator Frequency divided by 2 010 Electrode Oscillator Frequency divided by 4 011 Electrode Oscillator Frequency divided by 8 100 Electrode Oscillator Frequency divided by 16 101 Electrode Oscillator Frequency divided by 32 110 Electrode Oscillator Frequency divided by 64 111 Electrode Oscillator Frequency divided by 128
15 EOSF	End of Scan Flag. This flag is set when all active electrodes are scanned is ended after a scan trigger. Writing "1" to this bit will clear the flag to 0.
14 OUTRGF	Out of Range Flag. This flag is set if the result register of the low power enabled electrode is outside the range defined by the TSI_THRESHOLD register. This flag is only set when the TSI is in low power mode. It can be read once the CPU wakes up. Writing "1" to this bit will clear the flag to 0.
13 EXTERF	External Electrode error occurred This flag is set when an active electrode has a result register either 0x0000 or 0xFFFF. Writing "1" to this bit will clear the flag to 0. 0 No fault happend on TSI electrodes 1 Short to VDD or VSS was detected on one or more electrodes.
12 OVRF	Overrun error Flag. This flag is set when a scan trigger occurs while a scan is still in progress. Writing "1" to this bit will clear the flag to 0. 0 No over run. 1 Over Run occurred.
11–10 Reserved	This read-only field is reserved and always has the value zero.
9 SCNIP	Scan In Progress status "1" indicates a scanning process is in progress,this bit is read-only and changes automatically by the TSI model.
8 SWTS	Software Trigger Start Write a "1" to this bit will start a scan sequence and write a "0" to this bit has no effect.
7 TSIEN	Touch Sensing Input Module Enable

Table continues on the next page...

TSIx_GENCS field descriptions (continued)

Field	Description
	0 TSI module is disabled 1 TSI module is enabled
6 TSIIE	Touch Sensing Input Interrupt Module Enable 0 Interrupt from TSI is disabled 1 Interrupt from TSI is enabled
5 ERIE	Error Interrupt Enable Caused either by a Short or Overrun Error. 0 Interrupt disabled for error. 1 Interrupt enabled for error.
4 ESOR	End-of-Scan or Out-of-Range Interrupt select 0 Out-of-Range interrupt is allowed. 1 End-of-Scan interrupt is allowed.
3 Reserved	This read-only field is reserved and always has the value zero.
2 Reserved	Reserved This field is reserved.
1 STM	Scan Trigger Mode. This bit-field can only be changed if the TSI module is disabled (TSIEN bit = 0). 0 Software trigger scan. 1 Periodical Scan.
0 STPE	TSI STOP Enable while in Low Power Modes (STOP, VLPS, LLS and VLLS{3,2,1}) 0 Disable TSI when MCU goes into low power modes. 1 Allows TSI to continue running in all low power modes.

48.6.2 SCAN Control Register (TSIx_SCANC)

Addresses: TSI0_SCANC is 4004_5000h base + 4h offset = 4004_5004h

<table>
<tr><td>Bit</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>R</td><td colspan="4">0</td><td colspan="4" rowspan="2">REFCHRG</td><td colspan="4">0</td><td colspan="4" rowspan="2">EXTCHRG</td></tr>
<tr><td>W</td><td colspan="4"></td><td colspan="4"></td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

<table>
<tr><td>Bit</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td colspan="8" rowspan="2">SMOD</td><td colspan="2">0</td><td>0</td><td colspan="2" rowspan="2">AMCLKS</td><td colspan="3" rowspan="2">AMPSC</td></tr>
<tr><td>W</td><td colspan="2"></td><td></td></tr>
<tr><td>Reset</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

TSIx_SCANC field descriptions

Field	Description
31–28 Reserved	This read-only field is reserved and always has the value zero.
27–24 REFCHRG	Ref OSC Charge Current select 0000 2 μA charge current. 0001 4 μA charge current. 0010 6 μA charge current. 0011 8 μA charge current. 0100 10 μA charge current. 0101 12 μA charge current. 0110 14 μA charge current. 0111 16 μA charge current. 1000 18 μA charge current. 1001 20 μA charge current. 1010 22 μA charge current. 1011 24 μA charge current. 1100 26 μA charge current. 1101 28 μA charge current. 1110 30 μA charge current. 1111 32 μA charge current.
23–20 Reserved	This read-only field is reserved and always has the value zero.
19–16 EXTCHRG	External OSC Charge Current select 0000 2 μA charge current. 0001 4 μA charge current. 0010 6 μA charge current. 0011 8 μA charge current. 0100 10 μA charge current. 0101 12 μA charge current. 0110 14 μA charge current. 0111 16 μA charge current. 1000 18 μA charge current. 1001 20 μA charge current. 1010 22 μA charge current. 1011 24 μA charge current. 1100 26 μA charge current. 1101 28 μA charge current. 1110 30 μA charge current. 1111 32 μA charge current.
15–8 SMOD	Scan Module 00000000 Continue Scan. Others Scan Period Modulus.
7–6 Reserved	This read-only field is reserved and always has the value zero.

Table continues on the next page...

TSIx_SCANC field descriptions (continued)

Field	Description
5 Reserved	This read-only field is reserved and always has the value zero.
4–3 AMCLKS	Active Mode Clock Source 00 LPOSCCLK 01 MCGIRCLK. 10 OSCERCLK. 11 Not valid.
2–0 AMPSC	Active Mode Prescaler 000 Input Clock Source divided by 1. 001 Input Clock Source divided by 2. 010 Input Clock Source divided by 4. 011 Input Clock Source divided by 8. 100 Input Clock Source divided by 16. 101 Input Clock Source divided by 32. 110 Input Clock Source divided by 64. 111 Input Clock Source divided by 128.

48.6.3 Pin Enable Register (TSIx_PEN)

Do not change the settings when TSIEN is 1.

Addresses: TSI0_PEN is 4004_5000h base + 8h offset = 4004_5008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0												LPSP				PEN15	PEN14	PEN13	PEN12	PEN11	PEN10	PEN9	PEN8	PEN7	PEN6	PEN5	PEN4	PEN3	PEN2	PEN1	PEN0
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TSIx_PEN field descriptions

Field	Description
31–20 Reserved	This read-only field is reserved and always has the value zero.
19–16 LPSP	Low Power Scan Pin 0000 TSI_IN[0] is active in low power mode. 0001 TSI_IN[1] is active in low power mode. 0010 TSI_IN[2] is active in low power mode. 0011 TSI_IN[3] is active in low power mode. 0100 TSI_IN[4] is active in low power mode. 0101 TSI_IN[5] is active in low power mode. 0110 TSI_IN[6] is active in low power mode.

Table continues on the next page...

TSIx_PEN field descriptions (continued)

Field	Description
	0111 TSI_IN[7] is active in low power mode. 1000 TSI_IN[8] is active in low power mode. 1001 TSI_IN[9] is active in low power mode. 1010 TSI_IN[10] is active in low power mode. 1011 TSI_IN[11] is active in low power mode. 1100 TSI_IN[12] is active in low power mode. 1101 TSI_IN[13] is active in low power mode. 1110 TSI_IN[14] is active in low power mode. 1111 TSI_IN[15] is active in low power mode.
15 PEN15	Touch Sensing Input Pin Enable Register 15 0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.
14 PEN14	Touch Sensing Input Pin Enable Register 14 0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.
13 PEN13	Touch Sensing Input Pin Enable Register 13 0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.
12 PEN12	Touch Sensing Input Pin Enable Register 12 0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.
11 PEN11	Touch Sensing Input Pin Enable Register 11 0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.
10 PEN10	Touch Sensing Input Pin Enable Register 10 0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.
9 PEN9	Touch Sensing Input Pin Enable Register 9 0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.
8 PEN8	Touch Sensing Input Pin Enable Register 8 0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.
7 PEN7	Touch Sensing Input Pin Enable Register 7 0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.
6 PEN6	Touch Sensing Input Pin Enable Register 6

Table continues on the next page...

TSIx_PEN field descriptions (continued)

Field	Description
	0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.
5 PEN5	Touch Sensing Input Pin Enable Register 5 0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.
4 PEN4	Touch Sensing Input Pin Enable Register 4 0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.
3 PEN3	Touch Sensing Input Pin Enable Register 3 0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.
2 PEN2	Touch Sensing Input Pin Enable Register 2 0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.
1 PEN1	Touch Sensing Input Pin Enable Register 1 0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.
0 PEN0	Touch Sensing Input Pin Enable Register 0 0 The corresponding pin is not used by TSI. 1 The corresponding pin is used by TSI.

48.6.4 Wake-Up Channel Counter Register (TSIx_WUCNTR)

Addresses: TSI0_WUCNTR is 4004_5000h base + Ch offset = 4004_500Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																WUCNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TSIx_WUCNTR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value zero.
15–0 WUCNT	TouchSensing wake-up Channel 16bit counter value

48.6.5 Counter Register (TSIx_CNTR)

Addresses: TSI0_CNTR1 is 4004_5000h base + 100h offset = 4004_5100h
TSI0_CNTR3 is 4004_5000h base + 104h offset = 4004_5104h
TSI0_CNTR5 is 4004_5000h base + 108h offset = 4004_5108h
TSI0_CNTR7 is 4004_5000h base + 10Ch offset = 4004_510Ch
TSI0_CNTR9 is 4004_5000h base + 110h offset = 4004_5110h
TSI0_CNTR11 is 4004_5000h base + 114h offset = 4004_5114h
TSI0_CNTR13 is 4004_5000h base + 118h offset = 4004_5118h
TSI0_CNTR15 is 4004_5000h base + 11Ch offset = 4004_511Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CTN																CTN1															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TSIx_CNTR*n* field descriptions

Field	Description
31–16 CTN	TouchSensing Channel n 16-bit counter value
15–0 CTN1	TouchSensing Channel n-1 16-bit counter value

48.6.6 Low Power Channel Threshold Register (TSIx_THRESHOLD)

Addresses: TSI0_THRESHOLD is 4004_5000h base + 120h offset = 4004_5120h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LTHH																HTHH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

TSIx_THRESHOLD field descriptions

Field	Description
31–16 LTHH	Touch Sensing Channel Low Threshold value
15–0 HTHH	Touch Sensing Channel High Threshold value

48.7 Functional descriptions

This section provides functional description of the TSI module.

48.7.1 Capacitance measurement

The electrode pin capacitance measurement uses a dual oscillator approach. The TSI electrode oscillator has its frequency dependable of the external electrode capacitance and of the TSI module configuration. After going to a configurable prescaler, the TSI electrode oscillator signal goes to the input of the module counter. The time for the module counter to reach its module value is measured using the TSI reference oscillator. The measured electrode capacitance is directly proportional to the time.

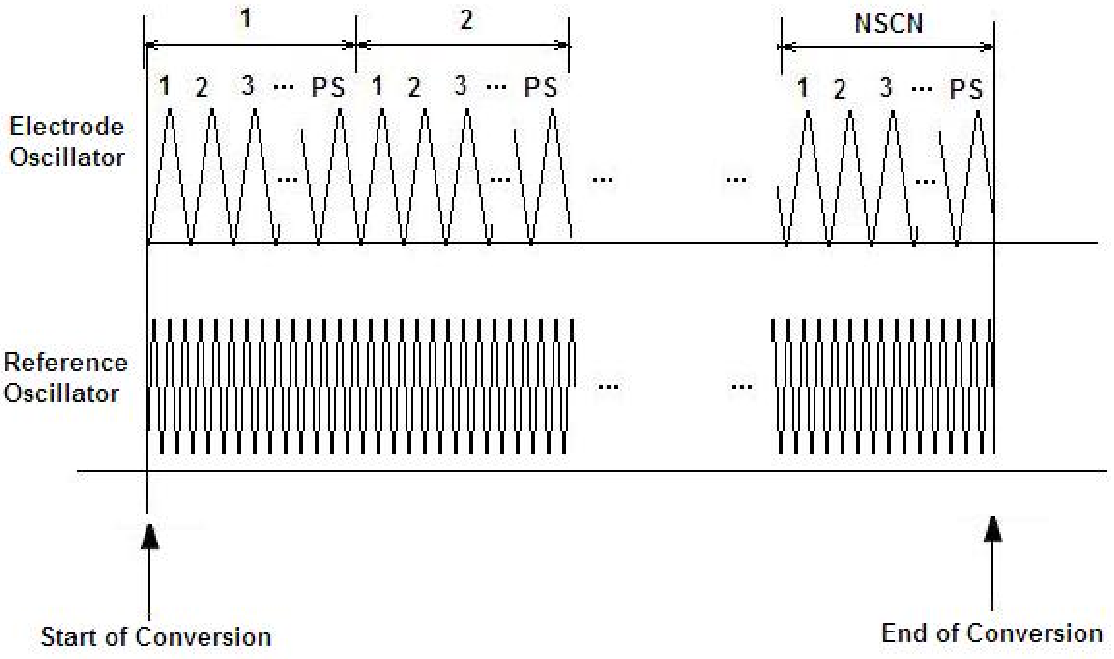

1
2
NSCN
1 2 3 ··· PS 1 2 3 ··· PS
1 2 3 ··· PS
Electrode Oscillator
Reference Oscillator
Start of Conversion
End of Conversion

Figure 48-32. Dual Electrode Capacitance Measurement

48.7.1.1 TSI electrode oscillator

The TSI electrode oscillator circuit is illustrated in the following figure. A configurable constant current source is used to charge and discharge the external electrode capacitance. A buffer hysteresis defines the oscillator delta voltage. The delta voltage defines the margin of high and low voltage which are the reference input of the comparator in different time.

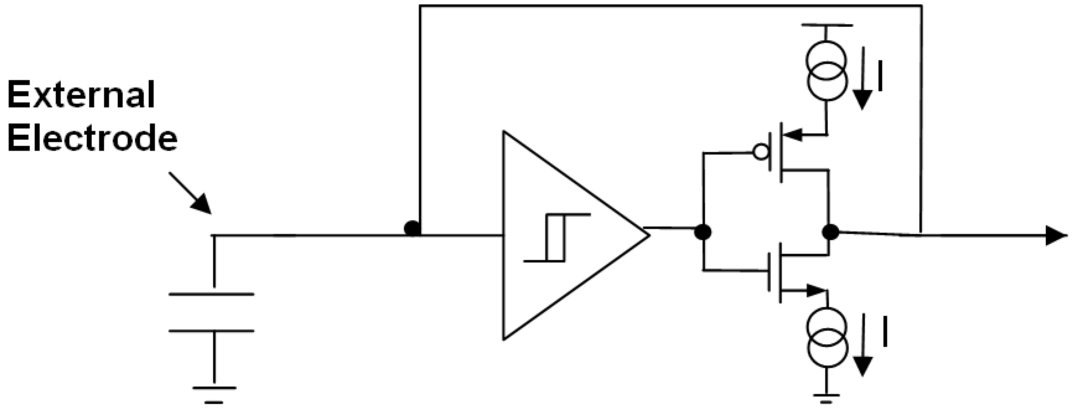


Figure 48-33. TSI electrode oscillator circuit

The current source applied to the pad capacitance is controlled by the SCANC[EXTCHRG]. The hysteresis delta voltage is defined inn the module electrical specifications present in the device DataSheet.. The figure below shows the voltage amplitude waveform of the electrode capacitance charging and discharging with a programmable current.

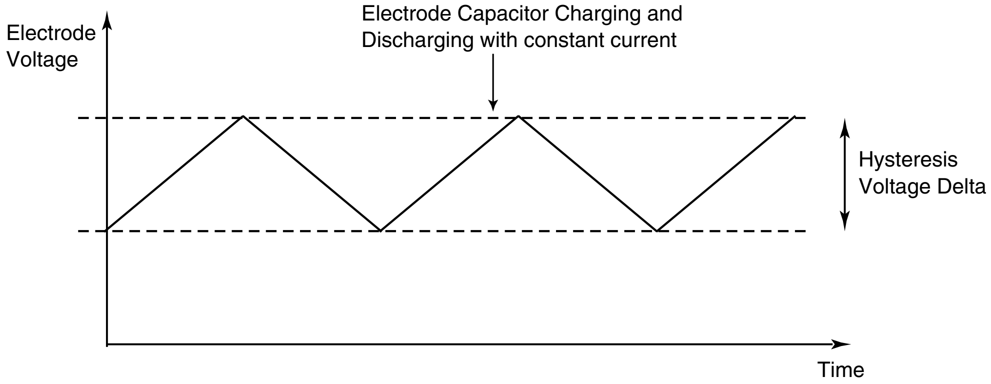


Figure 48-34. TSI electrode oscillator chart

The oscillator frequency is give by the following equation

$$F_{elec} = \frac{I}{2 * C_{elec} * \Delta V}$$

Figure 48-35. Equation 1: TSI electrode oscillator frequency

Where:

I: constant current

C_{elec}: electrode capacitance

ΔV: Hysteresis delta voltage

So by this equation, for example, an electrode with C_{elec}= 20 pF, with a current source of I = 16 μA and ΔV = 600 mV have the following oscillation frequency:

$$F_{elec} = \frac{16\ \mu A}{2 * 20pF * 600mV} = 0.67MHz$$

Figure 48-36. Equation 2: TSI electrode oscillator frequency

The current source is used to accommodate the TSI electrode oscillator frequency with different electrode capacitance sizes.

48.7.1.2 Electrode oscillator and counter module control

The TSI oscillator frequency signal goes through a prescaler defined by the GENCS[PS] and then enters in a module counter. The bit field GENCS[NSCN] defines the maximum count value of the module counter.

The pin capacitance sampling time is given by the time the module counter takes to go from zero to its maximum value, defined by NSCN. The electrode sample time is expressed by the following equation:

$$T_{cap\_samp} = \frac{PS * NSCN}{F_{elec}}$$

Using Equation 1.

$$T_{cap\_samp} = \frac{2 * PS * NSCN * C_{elec} * \Delta V}{I}$$

Figure 48-37. Equation 3: Electrode sampling time

Where:

PS: prescaler value

NSCN: module counter maximum value

I: constant current

C_{elec}: electrode capacitance

ΔV: Hysteresis delta voltage

By this equation we have that an electrode with C = 20 pF, with a current source of I = 16 μA and ΔV = 600 mV, PS = 2 and NSCN = 16 have the following sampling time:

$$T_{cap\_samp} = \frac{2*2*16*20pF*600mV}{16\mu A} = 48\mu s$$

48.7.1.3 TSI reference oscillator

The TSI reference oscillator has the same topology of the TSI electrode oscillator. The TSI reference oscillator instead of using an external capacitor for the electrode oscillator has an internal reference capacitor.

The TSI reference oscillator has an independent programmable current source controlled by the SCANC[REFCHRG].

The reference oscillator frequency is given by the following equation:

$$F_{ref\_osc} = \frac{I_{ref}}{2 * C_{ref} * \Delta V}$$

Figure 48-38. Equation 4: TSI reference oscillator frequency

Where:

C_{ref}: Internal reference capacitor

I_{ref}: Reference oscillator current source

ΔV : Hysteresis delta voltage

Considering C_{ref} = 1.0 pF, I_{ref} = 12 μA and ΔV = 600 mV, follows

$$F_{ref\_osc} = \frac{12\mu A}{2 * 1.0pF * 600mV} = 10.0MHz$$

48.7.2 TSI measurement result

The capacitance measurement result is defined by the number of TSI reference oscillator periods during the sample time and is stored in the TSICHnCNT register.

$TSICHnCNT = T_{cap\_samp} * F_{ref\_osc}$

Using Equation 2 and Equation 1 follows:

$$\mathrm{TSICHnCNT} = \frac{I_{ref} * PS * NSCN}{C_{ref} * I_{elec}} * C_{elec}$$

Figure 48-39. Equation 5: Capacitance result value

In the example where F_{ref_osc} = 10.0MHz and T_{cap_samp} = 48 μs, TSICHnCNT = 480

48.7.3 Electrode scan unit

This session describes the functionality of the electrode scan unit. It is responsible for triggering the start of the active electrode scan.

The touch sense input module needs to periodically scan all active electrodes to determine if a touch event has occurred. The electrode scan unit is responsible for defining two independent scan periods, one for TSI active mode and the other for TSI low power mode. This independent control allows the application to configure longer scan period during low power mode, so contributing to smaller average power consumption. The TSI, in low power mode, has the capability to wake up the CPU upon an electrode capacitance change. When the CPU wakes, the TSI enters active mode, and a shorter scan period can provide a faster response time and more robust touch detection. Apart from the periodical mode, the electrode scan unit also allows software triggering of the electrode scans. This feature is very useful for initialization of the touch application for detecting the initial electrode capacitances. This module generates configurable end-of-scan interrupt to indicate the application that all electrodes were scanned. In the event starting a new electrode scan while a previous one is still in progress an overrun error flag is generated.

48.7.3.1 Active electrodes

The electrode scan unit is responsible to start the capacitance measurement of all active electrodes. Each electrode pin should be activated by writing a 1 to the respective PEN[PEN] bit.

Once an electrode scan is triggered, the electrode scan unit, controls the scanning of all the active electrodes sequentially. It starts the scanning of the electrode pin TSI_IN[0] and goes sequentially scanning until it reaches the electrode pin TSI_IN[15]. The electrode pins that does not have its enable bit (PEN[PEN]) are not scanned and are skipped.

Only one electrode pin is functional in the low power mode scan and it's defined by the bit-field PEN[LPSP]. In low power scan mode the configuration of PEN[PEN] bits are ignored.

48.7.3.2 Scan trigger

The scan trigger can be set to periodical scan or software trigger. The bit GENCS[STM] determines the TSI scan trigger mode. If STM = 1 the trigger mode is selected as continuous. If STM = 0, the software trigger mode is selected. In periodic mode the scan trigger is generated automatically by the electrode scan unit

48.7.3.3 Software trigger mode

The software trigger scan is started by writing 1 to the bit GENCS[SWTS]. A single scan of all active electrodes is performed. The software trigger scan only can be initiated by the GENCS[SWTS] bit if the STM = 0. If STM = 1, any write in the GENCS[SWTS] bit is ignored.

48.7.3.4 Periodic scan control

The electrode scan unit operates both in TSI active mode and TSI low power mode. It has a separate scan period control for each one of these modes. It allows the application to controls the trade-off of the scan frequency and the average TSI module power consumption.

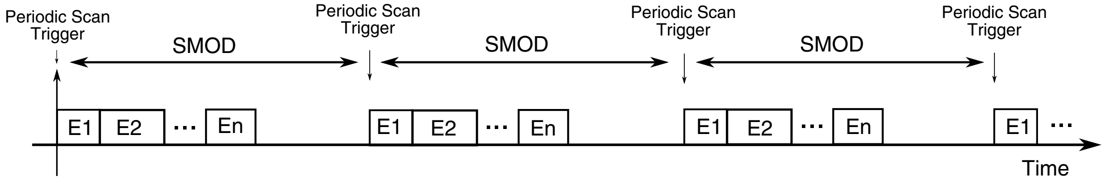


Figure 48-40. Periodical Scan Time Chart

48.7.3.4.1 Active mode periodic scan

In active mode periodic scan the scan following clocks can be selected: LPOOSCCLK, MCGIRCLK and OSCERCLK. The bit field SCANC[AMCLKS] selects the TSI clock source for the active mode scan. The scan period is determined by the SCANC[SMOD] value. SMOD is the module of the counter that determines the scan period.

The following figure presents the scan sequence performed by the TSI module. Every active electrode is scanned sequentially, stating with the TSI_IN[0] and ending with the TSI_IN[15] pin, if they are active.

When the electrode scan unit starts a scan sequence, all the active electrodes will be scanned sequentially with each electrode has the scanned time defined by the GENCS[NSCN]. The counter value is the sum of the total scan times of that electrode.

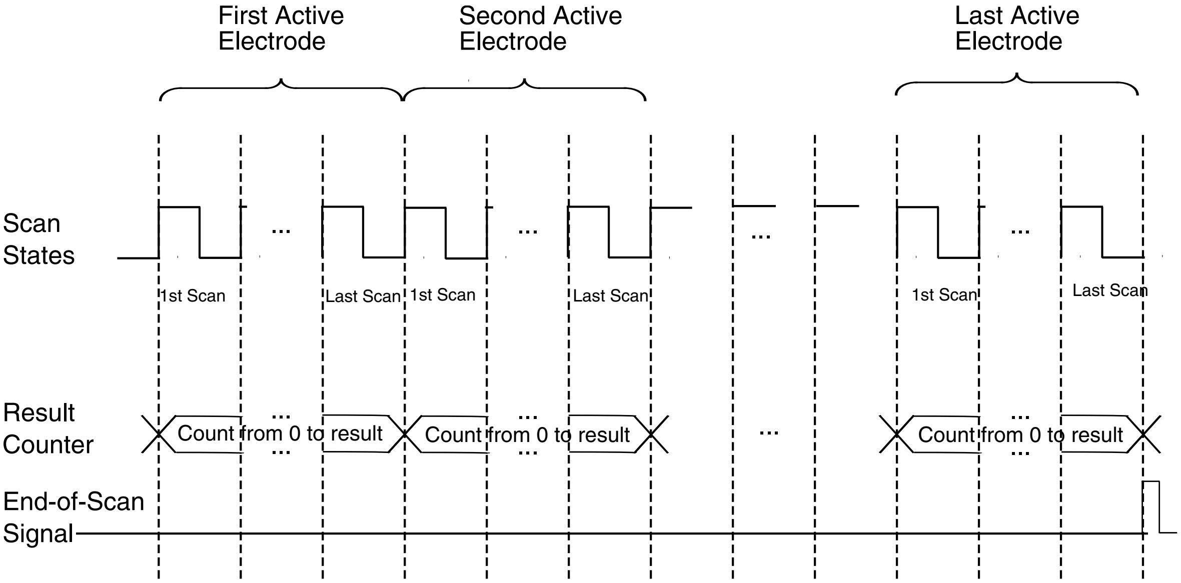

First Active Electrode
Second Active Electrode
Last Active Electrode
Scan States
1st Scan
Last Scan
1st Scan
Last Scan
1st Scan
Last Scan
Result Counter
Count from 0 to result
Count from 0 to result
Count from 0 to result
End-of-Scan Signal
...

Figure 48-41. Scan sequence

48.7.3.4.2 Low power mode scan

In low power periodic scan the scan period is define by the GENCS[LPSCNITV]. The TSI module is only enabled in low power modes only if the bit GENCS[STPE] is 1.

Only one electrode pin is functional in the low power mode scan and it’s defined by the bit-field PEN[LPSP].

48.7.3.4.3 End-of-scan interrupt

The electrode scan unit sets the EOSF flag in the GENCS registers once all the active electrode scan finishes. The EOSF Flag generate an end-of-scan interrupt request if it is enabled.. The interrupt is asserted if enabled by GENCS[TSIIE] and GENCS[ESOR] bits.

The GENCS[EOSF] indicates that all active electrode scans are finished and the respective capacitance results are in the TSICHnCNT registers. The GENCS[EOSF] is cleared by writing one to it.

48.7.3.4.4 Over-run interrupt

If an electrode scan is in progress and there is a scan trigger the electrode scan unit generates and over-run error by asserting the GENCS[OVRF]. If the TSI error interrupt is active by setting the GENCS[ERIE] bit a interrupt request is asserted. The OVRF flag is cleared by writing 1 to it.

48.7.4 Touch detection unit

The touch detection unit is responsible to detect electrode capacitance changes while in low power mode.

It also detects the occurrence of error with the electrode in the case it capacitance result is 0x0000 or 0xFFFF. The errors can be cause by electrode pin short circuit to V_{DD} or V_{SS}. Or by electrode capacitances out of the configuration range of the TSI module.

48.7.4.1 Capacitance change threshold

Each TSI pin has it result register TSICHnCNT. In low power mode only one electrode can be active, at the end of the low power active electrode conversion the touch detection unit compares if the TSICHnCNT result value is inside a configurable range. The comparison range is defined individually registers, TSICHHTH, the upper threshold value and TSICHLTH, the lower threshold value. If the TSICHnCNT happens to be out of the range defined by TSICHLTH and TSICHHTH the GENCS[OUTRGF] flag is set indicating that a capacitance change occurred in the low power active electrode..

48.7.4.1.1 Out-of-range interrupt

The GENCS[OUTRGF] flag generates a TSI interrupt request if the GENCS[TSIIE] bit is set and GENCS[ESOR] bit is cleared. With this configuration, after the end-of-electrode scan, the TSI interrupt is only requested if there is a capacitance change. If the low power electrode capacitance does not vary, the TSI Interrupt do not interrupt the CPU.

48.7.4.2 Error interrupt

The GENCS[EXTERF] is set in the case the capacitance result registers, TSICHnCNT, of a TSI pin is either 0 or 0xFFFF, the two possible extreme values. The EXTERF flag generates a TSI Error Interrupt request if the GENCS[ERIE] bit is set.

48.8 Application information

After enable the TSI module for the first time, it is highly recommended a calibration to all the enabled channels by setting proper high and low threshold value for each active channel. All the channel dedicated counter values can be read from each counter value registers, software suite can then adjust the threshold based on these values.

Follow proper PCB layout guidelines for board design on electrode shapes, sizes, routes, etc. Visit www.freescale.com/touch for application notes and reference designs.

48.8.1 TSI module sensitivity

The TSI module sensitivity is defined by the increment cause in the TSICHnCNT result registers caused by a 1 pF delta in the electrode pin capacitance.

It is given by the following equation:

$$TSI_{sensitivity} = \frac{C_{ref} * I}{I_{ref} * PS * NSCN}$$

For the example provided, $I_{ref} = 2$ µA, PS = 2; NSCN = 16, $C_{ref} = 1.0$ pF and I =2 µA, the

$TSI_{sensitivity} = 0.03125$ pf/count

48.9 TSI module initialization

This section provides the recommended initialization sequence for the TSI module.

Prior to enable TSI module by setting TSI_GENCS[TSIEN] bit, it is required to configure other bits first. The pin enable registers are set to select which channels will be sampled, the dual oscillators configuration bits are set in order to make the scan and conversion more accurate. Also remember not to change the settings while TSI is working in progress. To switch from different scan modes, for instance, it is required to do a software reset to TSI by disabling and then enabling TSI_GENCS[TSIEN].

48.9.1 Initialization Sequence

Freescale TSS library has complete support for TSI, which make the configuration and application much easier. For detailed information on how to work with TSI and TSS together, visit www.freescale.com/touchsensing to get the application notes for details.

Chapter 49
JTAG Controller (JTAGC)

49.1 Introduction

NOTE

> For the chip-specific implementation details of this module's instances see the chip configuration chapter.

The JTAGC block provides the means to test chip functionality and connectivity while remaining transparent to system logic when not in test mode. Testing is performed via a boundary scan technique, as defined in the IEEE 1149.1-2001 standard. All data input to and output from the JTAGC block is communicated in serial format.

49.1.1 Block diagram

The following is a block diagram of the JTAG Controller (JTAGC) block.

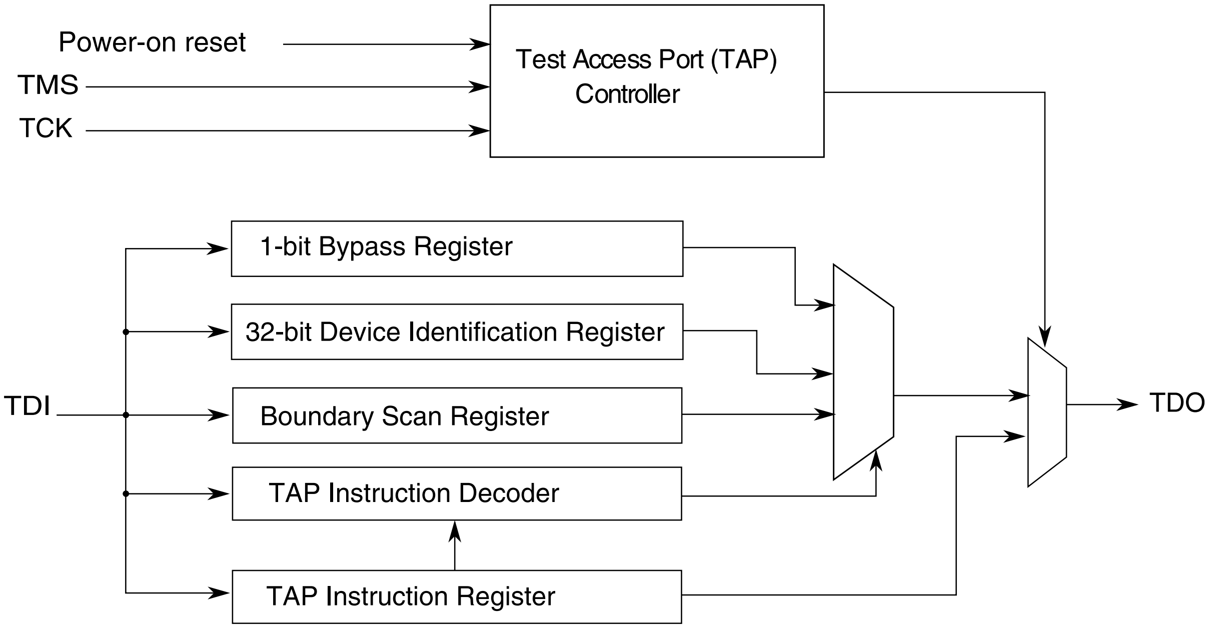

Power-on reset
TMS
TCK
Test Access Port (TAP) Controller
1-bit Bypass Register
32-bit Device Identification Register
TDI
Boundary Scan Register
TAP Instruction Decoder
TAP Instruction Register
TDO

Figure 49-1. JTAG (IEEE 1149.1) block diagram

49.1.2 Features

The JTAGC block is compliant with the IEEE 1149.1-2001 standard, and supports the following features:

- IEEE 1149.1-2001 Test Access Port (TAP) interface
  - 4 pins (TDI, TMS, TCK, and TDO)
- Instruction register that supports several IEEE 1149.1-2001 defined instructions as well as several public and private device-specific instructions. Refer to Table 49-3 for a list of supported instructions.
- Bypass register, boundary scan register, and device identification register.
- TAP controller state machine that controls the operation of the data registers, instruction register and associated circuitry.

49.1.3 Modes of operation

The JTAGC block uses a power-on reset indication as its primary reset signals. Several IEEE 1149.1-2001 defined test modes are supported, as well as a bypass mode.

49.1.3.1 Reset

The JTAGC block is placed in reset when either power-on reset is asserted, or the TMS input is held high for enough consecutive rising edges of TCK to sequence the TAP controller state machine into the Test-Logic-Reset state. Holding TMS high for five consecutive rising edges of TCK guarantees entry into the Test-Logic-Reset state regardless of the current TAP controller state. Asserting power-on reset results in asynchronous entry into the reset state. While in reset, the following actions occur:

- The TAP controller is forced into the Test-Logic-Reset state, thereby disabling the test logic and allowing normal operation of the on-chip system logic to continue unhindered
- The instruction register is loaded with the IDCODE instruction

49.1.3.2 IEEE 1149.1-2001 defined test modes

The JTAGC block supports several IEEE 1149.1-2001 defined test modes. A test mode is selected by loading the appropriate instruction into the instruction register while the JTAGC is enabled. Supported test instructions include EXTEST, HIGHZ, CLAMP, SAMPLE and SAMPLE/PRELOAD. Each instruction defines the set of data register(s) that may operate and interact with the on-chip system logic while the instruction is current. Only one test data register path is enabled to shift data between TDI and TDO for each instruction.

The boundary scan register is enabled for serial access between TDI and TDO when the EXTEST, SAMPLE or SAMPLE/PRELOAD instructions are active. The single-bit bypass register shift stage is enabled for serial access between TDI and TDO when the BYPASS, HIGHZ, CLAMP or reserved instructions are active. The functionality of each test mode is explained in more detail in JTAGC block instructions.

49.1.3.3 Bypass mode

When no test operation is required, the BYPASS instruction can be loaded to place the JTAGC block into bypass mode. While in bypass mode, the single-bit bypass shift register is used to provide a minimum-length serial path to shift data between TDI and TDO.

49.2 External signal description

The JTAGC consists of a set of signals that connect to off chip development tools and allow access to test support functions. The JTAGC signals are outlined in the following table and described in the following sections.

Table 49-1. JTAG signal properties

Name	I/O	Function	Reset State	Pull
TCK	Input	Test Clock	—	Down
TDI	Input	Test Data In	—	Up
TDO	Output	Test Data Out	High Z[1]	—
TMS	Input	Test Mode Select	—	Up

1. TDO output buffer enable is negated when the JTAGC is not in the Shift-IR or Shift-DR states. A weak pull may be implemented at the TDO pad for use when JTAGC is inactive.

49.2.1 TCK—Test clock input

Test Clock Input (TCK) is an input pin used to synchronize the test logic and control register access through the TAP.

49.2.2 TDI—Test data input

Test Data Input (TDI) is an input pin that receives serial test instructions and data. TDI is sampled on the rising edge of TCK.

49.2.3 TDO—Test data output

Test Data Output (TDO) is an output pin that transmits serial output for test instructions and data. TDO is three-stateable and is actively driven only in the Shift-IR and Shift-DR states of the TAP controller state machine, which is described in TAP controller state machine.

49.2.4 TMS—Test mode select

Test Mode Select (TMS) is an input pin used to sequence the IEEE 1149.1-2001 test control state machine. TMS is sampled on the rising edge of TCK.

49.3 Register description

This section provides a detailed description of the JTAGC block registers accessible through the TAP interface, including data registers and the instruction register. Individual bit-level descriptions and reset states of each register are included. These registers are not memory-mapped and can only be accessed through the TAP.

49.3.1 Instruction register

The JTAGC block uses a 4-bit instruction register as shown in the following figure. The instruction register allows instructions to be loaded into the block to select the test to be performed or the test data register to be accessed or both. Instructions are shifted in through TDI while the TAP controller is in the Shift-IR state, and latched on the falling edge of TCK in the Update-IR state. The latched instruction value can only be changed in the Update-IR and Test-Logic-Reset TAP controller states. Synchronous entry into the Test-Logic-Reset state results in the IDCODE instruction being loaded on the falling edge of TCK. Asynchronous entry into the Test-Logic-Reset state results in asynchronous loading of the IDCODE instruction. During the Capture-IR TAP controller state, the instruction shift register is loaded with the value 0001b , making this value the register's read value when the TAP controller is sequenced into the Shift-IR state.

	3	2	1	0
R	0	0	0	1
W	Instruction Code			
Reset:	0	0	0	1

Figure 49-2. Instruction register

49.3.2 Bypass register

The bypass register is a single-bit shift register path selected for serial data transfer between TDI and TDO when the BYPASS, CLAMP, HIGHZ or reserve instructions are active. After entry into the Capture-DR state, the single-bit shift register is set to a logic 0. Therefore, the first bit shifted out after selecting the bypass register is always a logic 0.

49.3.3 Device identification register

The device identification (JTAG ID) register, shown in the following figure, allows the revision number, part number, manufacturer, and design center responsible for the design of the part to be determined through the TAP. The device identification register is selected for serial data transfer between TDI and TDO when the IDCODE instruction is active. Entry into the Capture-DR state while the device identification register is selected loads the IDCODE into the shift register to be shifted out on TDO in the Shift-DR state. No action occurs in the Update-DR state.

<table>
<tr><td></td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td>R</td><td colspan="4">Part Revision Number</td><td colspan="6">Design Center</td><td colspan="6">Part Identification Number</td></tr>
<tr><td>W</td><td colspan="4"></td><td colspan="6"></td><td colspan="6"></td></tr>
<tr><td>RESET:</td><td colspan="4">PRN</td><td colspan="6">DC</td><td colspan="6">PIN</td></tr>
</table>

<table>
<tr><td></td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>R</td><td colspan="4">Part Identification Number</td><td colspan="11">Manufacturer Identity Code</td><td>1</td></tr>
<tr><td>W</td><td colspan="4"></td><td colspan="11"></td><td></td></tr>
<tr><td>RESET:</td><td colspan="4">PIN (contd.)</td><td colspan="11">MIC</td><td>1</td></tr>
</table>

The following table describes the device identification register functions.

Table 49-2. Device identification register field descriptions

Field	Description
PRN	Part Revision Number. Contains the revision number of the part. Value is 0x0.
DC	Design Center. Indicates the design center. Value is 0x2C.
PIN	Part Identification Number. Contains the part number of the device. Value is TBD.
MIC	Manufacturer Identity Code. Contains the reduced Joint Electron Device Engineering Council (JEDEC) ID. Value is 0x00E .
IDCODE ID	IDCODE Register ID. Identifies this register as the device identification register and not the bypass register. Always set to 1.

49.3.4 Boundary scan register

The boundary scan register is connected between TDI and TDO when the EXTEST, SAMPLE or SAMPLE/PRELOAD instructions are active. It is used to capture input pin data, force fixed values on output pins, and select a logic value and direction for bidirectional pins. Each bit of the boundary scan register represents a separate boundary scan register cell, as described in the IEEE 1149.1-2001 standard and discussed in Boundary scan. The size of the boundary scan register and bit ordering is device-dependent and can be found in the device BSDL file.

49.4 Functional description

This section explains the JTAGC functional description.

49.4.1 JTAGC reset configuration

While in reset, the TAP controller is forced into the Test-Logic-Reset state, thus disabling the test logic and allowing normal operation of the on-chip system logic. In addition, the instruction register is loaded with the IDCODE instruction.

49.4.2 IEEE 1149.1-2001 (JTAG) Test Access Port

The JTAGC block uses the IEEE 1149.1-2001 TAP for accessing registers. This port can be shared with other TAP controllers on the MCU. Ownership of the port is determined by the value of the currently loaded instruction.

Data is shifted between TDI and TDO though the selected register starting with the least significant bit, as illustrated in the following figure. This applies for the instruction register, test data registers, and the bypass register.

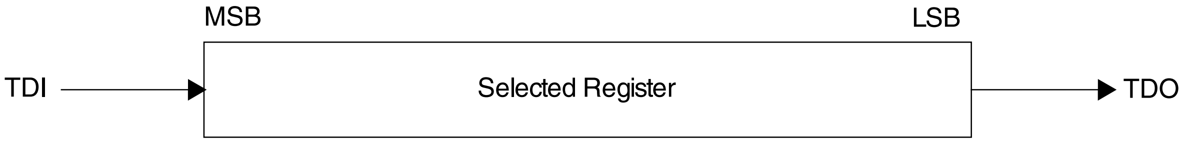
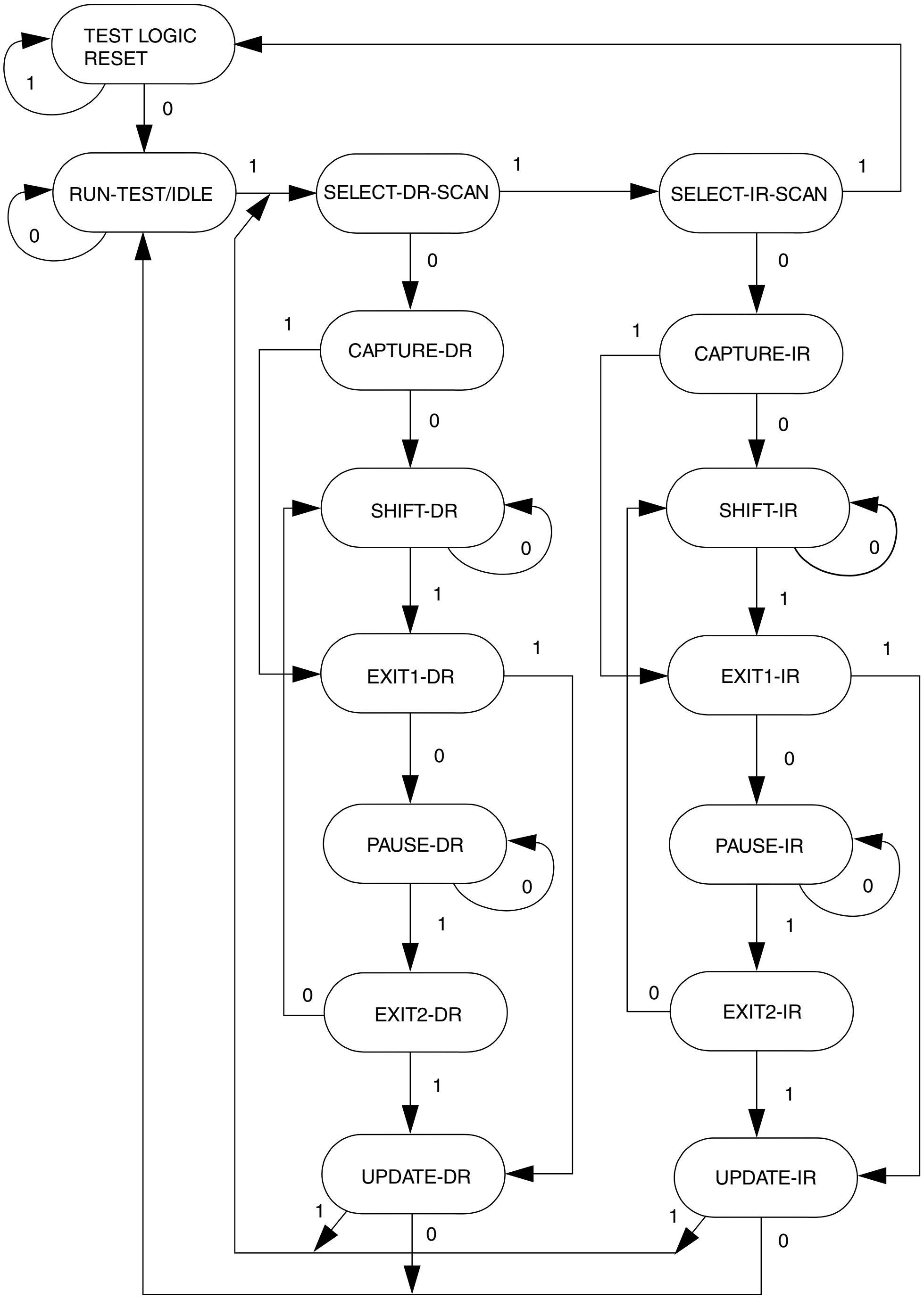


Figure 49-3. Shifting data through a register

49.4.3 TAP controller state machine

The TAP controller is a synchronous state machine that interprets the sequence of logical values on the TMS pin. The following figure shows the machine's states. The value shown next to each state is the value of the TMS signal sampled on the rising edge of the TCK signal. As the following figure shows, holding TMS at logic 1 while clocking TCK through a sufficient number of rising edges also causes the state machine to enter the Test-Logic-Reset state.


TEST LOGIC RESET
RUN-TEST/IDLE
SELECT-DR-SCAN
SELECT-IR-SCAN
CAPTURE-DR
CAPTURE-IR
SHIFT-DR
SHIFT-IR
EXIT1-DR
EXIT1-IR
PAUSE-DR
PAUSE-IR
EXIT2-DR
EXIT2-IR
UPDATE-DR
UPDATE-IR
0
1

The value shown adjacent to each state transition in this figure represents the value of TMS at the time of a rising edge of TCK.

Figure 49-4. IEEE 1149.1-2001 TAP controller finite state machine

49.4.3.1 Enabling the TAP controller

The JTAGC TAP controller is enabled by setting the JTAGC enable to a logic 1 value.

49.4.3.2 Selecting an IEEE 1149.1-2001 register

Access to the JTAGC data registers is achieved by loading the instruction register with any of the JTAGC block instructions while the JTAGC is enabled. Instructions are shifted in via the Select-IR-Scan path and loaded in the Update-IR state. At this point, all data register access is performed via the Select-DR-Scan path.

The Select-DR-Scan path is used to read or write the register data by shifting in the data (LSB first) during the Shift-DR state. When reading a register, the register value is loaded into the IEEE 1149.1-2001 shifter during the Capture-DR state. When writing a register, the value is loaded from the IEEE 1149.1-2001 shifter to the register during the Update-DR state. When reading a register, there is no requirement to shift out the entire register contents. Shifting may be terminated once the required number of bits have been acquired.

49.4.4 JTAGC block instructions

The JTAGC block implements the IEEE 1149.1-2001 defined instructions listed in the following table. This section gives an overview of each instruction; refer to the IEEE 1149.1-2001 standard for more details. All undefined opcodes are reserved.

Table 49-3. 4-bit JTAG instructions

Instruction	Code[3:0]	Instruction Summary
IDCODE	0000	Selects device identification register for shift
EZPORT	0001	Enables the EZPORT function for the SoC
SAMPLE/PRELOAD	0010	Selects boundary scan register for shifting, sampling, and preloading without disturbing functional operation
SAMPLE	0011	Selects boundary scan register for shifting and sampling without disturbing functional operation
EXTEST	0100	Selects boundary scan register while applying preloaded values to output pins and asserting functional reset
Factory debug reserved	0101	Intended for factory debug only
Factory debug reserved	0110	Intended for factory debug only
Factory debug reserved	0111	Intended for factory debug only
ARM JTAG-DP Reserved	1000	This instruction goes the ARM JTAG-DP controller. See the ARM JTAG-DP documentation for more information.
HIGHZ	1001	Selects bypass register while three-stating all output pins and asserting functional reset
ARM JTAG-DP Reserved	1010	This instruction goes the ARM JTAG-DP controller. See the ARM JTAG-DP documentation for more information.

Table continues on the next page...

Table 49-3. 4-bit JTAG instructions (continued)

Instruction	Code[3:0]	Instruction Summary
ARM JTAG-DP Reserved	1011	This instruction goes the ARM JTAG-DP controller. See the ARM JTAG-DP documentation for more information.
CLAMP	1100	Selects bypass register while applying preloaded values to output pins and asserting functional reset
ARM JTAG-DP Reserved	1110	This instruction goes the ARM JTAG-DP controller. See the ARM JTAG-DP documentation for more information.
BYPASS	1111	Selects bypass register for data operations

49.4.4.1 IDCODE instruction

IDCODE selects the 32-bit device identification register as the shift path between TDI and TDO. This instruction allows interrogation of the MCU to determine its version number and other part identification data. IDCODE is the instruction placed into the instruction register when the JTAGC block is reset.

49.4.4.2 EZPORT instruction

The EZPORT instruction allows for the EZPORT module to program the on-chip flash from a simple 4-pin interface. The JTAGC forces the core into a reset state and forces the EZPORT mode select/chip select low. In this mode, the flash can be programmed through the JTAG test port pins, which are connected to the EZPORT module.

49.4.4.3 SAMPLE/PRELOAD instruction

The SAMPLE/PRELOAD instruction has two functions:

- The SAMPLE portion of the instruction obtains a sample of the system data and control signals present at the MCU input pins and just before the boundary scan register cells at the output pins. This sampling occurs on the rising edge of TCK in the Capture-DR state when the SAMPLE/PRELOAD instruction is active. The sampled data is viewed by shifting it through the boundary scan register to the TDO output during the Shift-DR state. Both the data capture and the shift operation are transparent to system operation.
- The PRELOAD portion of the instruction initializes the boundary scan register cells before selecting the EXTEST or CLAMP instructions to perform boundary scan tests. This is achieved by shifting in initialization data to the boundary scan register during the Shift-DR state. The initialization data is transferred to the parallel outputs

of the boundary scan register cells on the falling edge of TCK in the Update-DR state. The data is applied to the external output pins by the EXTEST or CLAMP instruction. System operation is not affected.

49.4.4.4 SAMPLE instruction

The SAMPLE instruction obtains a sample of the system data and control signals present at the MCU input pins and just before the boundary scan register cells at the output pins. This sampling occurs on the rising edge of TCK in the Capture-DR state when the SAMPLE instruction is active. The sampled data is viewed by shifting it through the boundary scan register to the TDO output during the Shift-DR state. There is no defined action in the Update-DR state. Both the data capture and the shift operation are transparent to system operation.

49.4.4.5 EXTEST External test instruction

EXTEST selects the boundary scan register as the shift path between TDI and TDO. It allows testing of off-chip circuitry and board-level interconnections by driving preloaded data contained in the boundary scan register onto the system output pins. Typically, the preloaded data is loaded into the boundary scan register using the SAMPLE/PRELOAD instruction before the selection of EXTEST. EXTEST asserts the internal system reset for the MCU to force a predictable internal state while performing external boundary scan operations.

49.4.4.6 HIGHZ instruction

HIGHZ selects the bypass register as the shift path between TDI and TDO. While HIGHZ is active all output drivers are placed in an inactive drive state (e.g., high impedance). HIGHZ also asserts the internal system reset for the MCU to force a predictable internal state.

49.4.4.7 CLAMP instruction

CLAMP allows the state of signals driven from MCU pins to be determined from the boundary scan register while the bypass register is selected as the serial path between TDI and TDO. CLAMP enhances test efficiency by reducing the overall shift path to a single bit (the bypass register) while conducting an EXTEST type of instruction through the boundary scan register. CLAMP also asserts the internal system reset for the MCU to force a predictable internal state.

49.4.4.8 BYPASS instruction

BYPASS selects the bypass register, creating a single-bit shift register path between TDI and TDO. BYPASS enhances test efficiency by reducing the overall shift path when no test operation of the MCU is required. This allows more rapid movement of test data to and from other components on a board that are required to perform test functions. While the BYPASS instruction is active the system logic operates normally.

49.4.5 Boundary scan

The boundary scan technique allows signals at component boundaries to be controlled and observed through the shift-register stage associated with each pad. Each stage is part of a larger boundary scan register cell, and cells for each pad are interconnected serially to form a shift-register chain around the border of the design. The boundary scan register consists of this shift-register chain, and is connected between TDI and TDO when the EXTEST, SAMPLE, or SAMPLE/PRELOAD instructions are loaded. The shift-register chain contains a serial input and serial output, as well as clock and control signals.

49.5 Initialization/Application information

The test logic is a static logic design, and TCK can be stopped in either a high or low state without loss of data. However, the system clock is not synchronized to TCK internally. Any mixed operation using both the test logic and the system functional logic requires external synchronization.

To initialize the JTAGC block and enable access to registers, the following sequence is required:

1. Place the JTAGC in reset through TAP controller state machine transitions controlled by TMS
2. Load the appropriate instruction for the test or action to be performed

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductors products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claims alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-complaint and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale's Environmental Products program, go to http://www.freescale.com/epp.

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2011–2012 Freescale Semiconductor, Inc.

Document Number: K20P64M50SF0RM
Rev. 2, Feb 2012