



DP32G030 Reference book

32bit microcontroller

Version: 1.23



Action Dynamic Tech.(HK) Trading Co.

Shenzhen Kinetic Century Technology Co., Ltd.

Company Address: Room 1111, Block B, Phase 6, International Innovation Valley, Dashi 1st Road, Nanshan District,

Shenzhen Tel: 0755-83134419

Fax: 0755-82519160

Company website: www.dnsj88.com

EMAIL: dnsj@dn-ic.com

Zip code: 518031

Table of contents

Table of contents.....	1
Figure Catalog.....	13
table of contents	18
Revision History.....	19
1.Introduction.....	20
1.1Overview.....	20
1.2Product Features.....	twenty one
2.Selection Guide.....	26
3.Chip Structure Block Diagram.....	27
4.Pin definition.....	28
4.1Package form.....	28
4.2Multiplexed pins.....	30
4.3Pin Alternate Function	36
5.Functional description.....	39
5.1Address Space Mapping.....	39
5.2Memory partitioning and access control.....	41
5.3Interrupt Vector Table	41
5.4 ARM Cortex-M0kernel.....	43
5.4.1Overview.....	43
5.4.2characteristic.....	43
5.4.3system timer (SYSTICK)	45
SYST_CTRLStatus Register (0x00)	47
SYST_LOADreload register (0x04)	47
SYST_VALCurrent Value Register (0x08)	48
5.5power management (PMU)	49
5.5.1Overview.....	49
5.5.2characteristic.....	49
5.5.3Block Diagram of Module Structure.....	49
5.5.4Functional description.....	51
power supply.....	51
Reset.....	52
Low Power Mode	55
Register Map	62
Register Description.....	63
LPOW_MDregister(0x00)	63
LPMD_WKENregister(0x04)	64
LPMD_WKSTregister(0x08)	64
CHIP_RST_STregister(0x0C)	65
SRC_CFGregister(0x10)	65
TRIM_POW0register(0x20)	66
TRIM_POW1register(0x24)	66

TRIM_POW2register(0x28)	67
TRIM_POW3register(0x2C)	67
TRIM_RCHFregister(0x30)	67
TRIM_RCLFregister(0x34)	68
TRIM_OPAregister(0x38)	68
TRIM_PLLregister(0x3C)	69
TRIM_LOCKregister(0x80)	69
DATA_BAK0register(0x100)	69
DATA_BAK1register(0x104)	69
DATA_BAK2register(0x108)	70
DATA_BAK3register(0x10C)	70
5.6 System Control (SYSCON)	71
5.6.1 Overview.....	71
5.6.2 characteristic.....	71
5.6.3 Block Diagram of Module Structure.....	71
5.6.4 Functional description.....	73
RCHFclock.....	74
PLLsclock.....	75
XTALclock.....	76
RCLFclock.....	77
system clock (sys_clk)choose.....	78
RTCclock.....	78
Independent Watchdog (IWDT)clock.....	79
window watchdog (WWDT)clock.....	79
SARADC Sampling Clock	79
Low Power Mode Wakeup Clock	79
128Bit chip unique identification code.....	79
Register Map	80
Register Description.....	81
CLK_SELregister(0x00)	81
DIV_CLK_GATEregister(0x04)	82
DEV_CLK_GATEregister(0x08)	82
RC_FREQ_DELTAregister(0x78)	84
VREF_VOLT_DELTAregister(0x7C)	84
CHIP_ID0register(0x80)	85
CHIP_ID1register(0x84)	85
CHIP_ID2register(0x88)	85
CHIP_ID3register(0x8C)	85
PLL_CTRLregister(0x180)	85
PLL_STregister(0x184)	86
5.7 I/O Function configuration (PORTCON)	87
5.7.1 Overview.....	87
5.7.2 characteristic.....	87
5.7.3 Block Diagram of Module Structure.....	88

5.7.4Functional description.....	89
Register Map Register Map	94
Register Description.....	95
PORTA_SEL0register(0x00)	95
PORTA_SEL1register(0x04)	97
PORTB_SEL0register(0x08)	99
PORTB_SEL1register(0x0C)	100
PORTC_SEL0register(0x10)	102
PORTA_IERegister(0x100)	104
PORTB_IERegister(0x104)	104
PORTC_IERegister(0x108)	104
PORTA_PUregister(0x200)	105
PORTB_PUregister(0x204)	105
PORTC_PUregister(0x208)	105
PORTA_PDregister(0x300)	105
PORTB_PDregister(0x304)	106
PORTC_PDregister(0x308)	106
PORTA_ODregister(0x400)	106
PORTB_ODregister(0x404)	107
PORTC_ODregister(0x408)	107
PORTA_WKEregister(0x500)	107
PORTB_WKEregister(0x504)	107
PORTC_WKEregister(0x508)	108
PORT_CFGregister(0x600)	108
PORTA_WK_SELregister(0x700)	109
PORTB_WK_SELregister(0x704)	109
PORTC_WK_SELregister(0x708)	109
5.8universalIO (GPIO).....	110
5.8.1Overview.....	110
5.8.2characteristic.....	110
5.8.3Block Diagram of Module Structure.....	111
5.8.4Functional description.....	111
register mapGPIODATAREgister(0x00)	119
GPIODIRRegister(0x04)	119
INTLVLTRGregister(0x08)	119
INTBEEregister(0x0C)	119
INTRISEENregister(0x10)	120
INTENregister(0x14)	120
INTRAWSTAUSregister(0x18)	120
INTSTAUSregister(0x1C)	121
INTCLRregister(0x20)	121
5.9Basic Timer (TIMERBASE)	122
5.9.1Overview.....	122
5.9.2characteristic.....	122

5.9.3Block Diagram of Module Structure.....	123
5.9.4Functional description.....	124
Register Map	129
Register Description.....	130
TIMERBASE_ENregister(0x00)	130
TIMERBASE_DIVregister(0x04)	130
TIMERBASE_IERegister(0x10)	130
TIMERBASE_IFregister(0x14)	131
HIGH_LOADregister(0x20)	131
HIGH_CNTregister(0x24)	131
LOW_LOADregister(0x30)	131
LOW_CNTregister(0x34)	132
5.10Advanced Timer (TIMERPLUS)	133
5.10.1Overview.....	133
5.10.2characteristic.....	134
5.10.3Block Diagram of Module Structure.....	135
5.10.4Functional description.....	137
Counting clock source selection and prescaler.....	137
Timing Mode.....	138
Counting mode.....	140
Enter Capture Mode	142
HALLmodel.....	144
Interruption.....	146
Operating procedures.....	147
Register Map	158
Register Description.....	159
TIMERPLUS_ENregister(0x00)	159
TIMERPLUS_DIVregister(0x04)	159
TIMERPLUS_CTRregister(0x08)	159
TIMERPLUS_IERegister(0x10)	161
TIMERPLUS_IFregister(0x14)	162
TIMERPLUS_HIGH_LOADregister(0x20)	163
TIMERPLUS_HIGH_CNTregister(0x24)	163
TIMERPLUS_HIGH_CVALregister(0x28)	164
TIMERPLUS_LOW_LOADregister(0x30)	164
TIMERPLUS_LOW_CNTregister(0x34)	164
TIMERPLUS_LOW_CVALregister(0x38)	164
HALL_VALregister(0x40)	165
5.11Independent watchdog clock (IWDT)	165
5.11.1Overview.....	165
5.11.2characteristic.....	166
5.11.3Block Diagram of Module Structure.....	167
5.11.4Functional description.....	167
Register Map	171

Register Description.....	171
IWDT_LOADregister(0x00)	171
IWDT_CTRLregister(0x08)	172
IWDT_IFregister(0x0C)	172
IWDT_FEEDregister(0x10)	172
5.12Window watchdog clock (WWDT)	173
5.12.1Overview.....	173
5.12.2characteristic.....	173
5.12.3Block Diagram of Module Structure.....	174
5.12.4Functional description.....	175
WWDTInterrupt generation and dog feeding interval.....	175
WWDTFeed Dog and Reset Generation	176
Register Map	177
Register Description.....	177
WWDT_LOADregister(0x00)	177
WWDT_VALUEregister(0x04)	177
WWDT_CTRLregister(0x08)	178
WWDT_IFregister(0x0C)	178
WWDT_FEEDregister(0x10)	179
5.13Basic PWM Generator (PWMBASE)	180
5.13.1Overview.....	180
5.13.2characteristic.....	180
5.13.3Block Diagram of Module Structure.....	181
5.13.4Functional description.....	183
Register Map	189
Register Description.....	189
PWMBASE_ENregister(0x00)	189
PWMBASE_DIVregister(0x04)	190
PWMBASE_CONregister(0x08)	190
PWMBASE_PERIODregister(0x0C)	191
PWMBASE_INTEregister(0x10)	191
PWMBASE_IFregister(0x14)	192
PWMBASE_CNTregister(0x18)	192
PWMBASE_CH0_COMPregister(0x20)	192
PWMBASE_CH1_COMPregister(0x30)	192
PWMBASE_CH2_COMPregister(0x40)	193
5.14Advanced Pulse Width Modulation Generator (PWMPLUS)	194
5.14.1Overview.....	194
5.14.2characteristic.....	194
5.14.3Block Diagram of Module Structure.....	195
5.14.4Functional description.....	198
Register Description.....	221
PWMPLUS_CFGregister(0x00)	221
PWMPLUS_GENregister(0x04)	222

PWMPLUS_CLKSRCregister(0x08)	224
PWMPLUS_BRAKE_CFGregister(0x0c)	225
PWMPLUS_MASK_LEVregister(0x10)	227
PWMPLUS_PERIODregister(0x1C)	228
PWMPLUS_CH0_COMPregister(0x20)	228
PWMPLUS_CH1_COMPregister(0x24)	229
PWMPLUS_CH2_COMPregister(0x28)	229
PWMPLUS_CH0_DTregister(0x30)	229
PWMPLUS_CH1_DTregister(0x34)	230
PWMPLUS_CH2_DTregister(0x38)	230
PWMPLUS_TRIG_COMPregister(0x40)	230
PWMPLUS_TRIG_CFGregister(0x44)	231
PWMPLUS_IERegister(0x60)	231
PWMPLUS_IFregister(0x64)	232
PWMPLUS_SWLOADregister(0x84)	234
PWMPLUS_MASK_ENregister(0x88)	235
PWMPLUS_CNT_STregister(0xE0)	235
PWMPLUS_BRAKE_STregister(0xE4)	236
5.15 Real Time Clock(RTC)	237
5.15.1 Overview.....	237
5.15.2 characteristic.....	237
5.15.3 Module Structure Diagram.....	238
5.15.4 Functional description.....	239
Counting clock source selection.....	239
Second mark function.....	239
Calendar function	240
Alarm function.....	241
Interrupt function	241
Low power wake-up function.....	242
Operating procedures.....	242
Register Map	244
Register Description.....	244
RTC_CFGregister(0x00)	244
RTC_IERegister(0x04)	245
RTC_IFregister(0x08)	246
RTC_PREGregister(0x10)	247
RTC_TRregister(0x14)	247
RTC_DRregister(0x18)	248
RTC_ARregister(0x1C)	249
RTC_TSTRregister(0x20)	250
RTC_TSDRregister(0x24)	251
RTC_CNTregister(0x28)	251
RTC_VALIDregister(0x2C)	251
5.16 UARTscontroller (UART)	253

5.16.1 Overview.....	253
5.16.2 characteristic.....	254
5.16.3 Block Diagram of Module Structure.....	255
Functional description.....	256
Receive Timing and Related Status Signals.....	256
Transmit timing and related status signals.....	257
Baud Rate Calculation.....	257
Auto baud rate detection	258
Data Sampling	259
Hardware Automatic Flow Control	259
Receiving Timeout Instructions.....	261
Interruption.....	262
Register Map	263
Register Description.....	264
UART_CTRLregister(0x00)	264
UART_BAUDregister(0x04)	265
UART_TDRregister(0x08)	266
UART_RDRregister(0x0C)	266
UART_IERegister(0x10)	266
UART_IFregister(0x14)	267
UART_FIFOREgister(0x18)	269
UART_FCregister(0x1C)	270
UART_RXTOregister(0x20)	271
5.17 SPIbus controller (SPI)	272
5.17.1 Overview.....	272
5.17.2 characteristic.....	272
5.17.3 Block Diagram of Module Structure.....	273
5.17.4 Functional description.....	274
SPIInterface Timing.....	274
I/O configuration.....	275
Data Transfer Configuration	277
Interrupt Generation.....	278
DMAcontrol.....	284
Operating procedures.....	285
Register Map	286
Register Description.....	286
SPICRregister(0x00)	286
SPIWDRregister(0x04)	288
SPIRDRregister(0x08)	289
SPIIERegister(0x10)	289
SPIIFregister(0x14)	289
SPIFIFOregister(0x18)	290
5.18 IICcontroller (IIC)	292
5.18.1 Overview.....	292

5.18.2characteristic.....	293
5.18.3Block Diagram of Module Structure.....	294
5.18.4Functional description.....	294
Introduction to the protocol.....	294
SCLandSDA.....	298
Interrupt function	299
Operating procedures.....	300
Register Map	312
Register Description.....	312
IIC_CCFGregister(0x00)	312
IIC_CSTregister(0x04)	313
IIC_CTRANSregister(0x08)	314
IIC_RXDATAreigister(0x0C)	315
IIC_TXDATAreigister(0x10)	315
IIC_IERegister(0x14)	316
IIC_IFregister(0x18)	316
IIC_MCTRLregister(0x20)	318
IIC_MSPCregister(0x24)	319
IIC_SCTRLregister(0x30)	320
IIC_SADDRregister(0x34)	321
5.19ADC (ADC)	323
5.19.1Overview.....	323
5.19.2characteristic.....	324
5.19.3Block Diagram of Module Structure.....	326
5.19.4Functional description.....	327
Channel selection	327
Sampling Once vs. Sampling Many Times	327
Single Sampling vs. Continuous Sampling	328
channel storage withFIFOs.....	333
Internal Sampling Clock Mode and External Sampling Clock Mode.....	334
CPUTriggering and external signal triggering.....	334
Interruption.....	335
Operating procedures.....	336
Register Map	337
Register Description.....	339
ADC_CFGregister(0x00)	339
ADC_STARTregister(0x04)	341
ADC_IERegister(0x08)	342
ADC_IFregister(0x0C)	343
ADC_CHx_STATregister.....	343
ADC_CHx_DATAreigister.....	343
ADC_FIFO_STATregister(0xA0)	344
ADC_FIFO_DATAreigister(0xA4)	345
ADC_EXTRIG_SELregister(0xB0)	345

ADC_CALIB_OFFSETregister(0xF0)	345
ADC_CALIB_KDregister(0xF4)	346
5.20Comparators(COMP)	347
5.20.1Overview.....	347
5.20.2characteristic.....	347
5.20.3Block Diagram of Module Structure.....	348
5.20.4Functional description.....	349
Comparator input pins and internal output signals.....	349
Hysteresis.....	349
Filtering.....	350
Interruption.....	351
Register Map	351
Register Description.....	352
CMP_CFGregister(0x120)	352
CMP_STregister(0x124)	353
5.21Operational Amplifier(OPAMP)	354
5.21.1Overview.....	354
5.21.2characteristic.....	354
5.21.3Module Structure Diagram.....	355
Register Map	355
Register Description.....	356
OPA_CFG.....	356
5.22 FLASHcontroller (FLASH CTRL)	357
5.22.1Overview.....	357
5.22.2characteristic.....	357
5.22.3Block Diagram of Module Structure.....	358
5.22.4Functional description.....	359
Low Power Mode	359
Read Rate Configuration	359
NVRDistrict andMAINDistrict Selection	360
Mode of operation.....	360
FLASHOperating address.....	360
Sector Size.....	361
STARTOperation Launch Control	361
Operation lock.....	361
MASKFunction.....	361
Erase Time and Program Time Configuration	362
FLASHInitialize busy flag	362
Controller Busy Flag	362
Program Data Cache Register Empty Status Flag.....	362
FLASHProgram initialization process.....	363
FLASHSector Erase Operation Flow.....	363
FLASHSingle-word programming operation flow.....	365
FLASHMulti-word continuous programming operation flow.....	367



Register Map	369
Register Description.....	369
FLASH_CFGregister(0x00)	369
FLASH_ADDRregister(0x04)	370
FLASH_WDATAreregister(0x08)	371
FLASH_STARTregister(0x10)	371
FLASH_STregister(0x14)	371
FLASH_LOCKregister(0x18)	372
FLASH_UNLOCKregister(0x1C)	372
FLASH_MASKregister(0x20)	372
FLASH_ERASETIMEregister(0x24)	373
FLASH_PROGTIMEregister(0x28)	373
5.23Cyclic Redundancy Check (CRC)	375
5.23.1Overview.....	375
5.23.2characteristic.....	375
5.23.3Module Structure Diagram.....	376
5.23.4Functional description.....	377
CRCCode Generation Rules	377
CRCpolynomial	378
Operating procedures.....	378
Register Map	379
Register Description.....	379
CRC_CRregister(0x00)	379
CRC_IVregister(0x04)	380
CRC_DATAINregister(0x08)	381
CRC_DATAOUTregister(0x0C)	381
5.24DMAcontroller (DMA)	382
5.24.1Overview.....	382
5.24.2characteristic.....	382
5.24.3Schematic diagram of the module structure.....	383
5.24.4Functional description.....	384
DMAProcessing flow.....	384
DMAArbiter Processing	385
DMAChannel configuration information.....	386
Source address and destination address data format description.....	388
Interrupt Description	390
DMARequest mapping relationship.....	390
DMAwork process.....	393
Register Map	395
Register Description.....	396
DMA_CTRregister(0x00)	396
DMA_INTENregister(0x04)	396
DMA_INTSTregister(0x08)	397
DMA_CHnCTRregister(0x100 + 0x20*(n))	398

DMA_CHnMODRegister(0x100 + 0x20*(n) + 0x04)	399
DMA_CHnMSADDRRegister(0x100 + 0x20*(n) + 0x08)	401
DMA_CHnMDADDRRegister(0x100 + 0x20*(n) + 0x0C)	401
DMA_CHn_STregister(0x100 + 0x20*(n) + 0x10)	401
5.25 AES arithmetic unit (AES)	402
5.25.1 Overview.....	402
5.25.2 characteristic.....	403
5.25.3 Block Diagram of Module Structure.....	403
5.25.4 Functional description.....	404
Electronic Codebook (ECB)	404
Cipher block chaining (CBC)	406
counter mode (CTR)	407
Data Schema.....	409
Operating procedures.....	411
Register Map	414
Register Description.....	414
AES_CRRegister(0x00)	414
AES_SRRegister(0x04)	416
AES_DINRRegister(0x08)	416
AES_DOUTRRegister(0x0C)	417
AES_KEYR0register(0x10)	417
AES_KEYR1register(0x14)	418
AES_KEYR2register(0x18)	418
AES_KEYR3register(0x1C)	418
AES_IVR0register(0x20)	419
AES_IVR1register(0x24)	419
AES_IVR2register(0x28)	419
AES_IVR3register(0x2C)	420
6.Electrical Characteristics	421
6.1 Test Conditions.....	421
6.2 Absolute Maximum Ratings	421
6.2.1 Voltage characteristics.....	421
6.2.2 Current Characteristics	421
6.2.3 Temperature Characteristics	422
6.3 Working conditions	422
6.3.1 General working conditions.....	422
6.3.2 Power-On and Power-Down Operating Conditions.....	422
6.3.3 RESET AND POWER CONTROL MODULE CHARACTERISTICS.....	422
6.3.4 Supply Current Characteristics	423
6.3.5 Embedded voltage reference.....	424
6.3.6 Low Power Mode Wake Up Time.....	424
6.3.7 Internal High-Frequency Clock Source Characteristics	425
6.3.8 Internal Low Frequency Clock Source Characteristics.....	425
6.3.9 External high-frequency clock source characteristics.....	425

6.3.10 External low frequency clock source characteristics.....	426
6.3.11 PLLs characteristic.....	426
6.3.12 FLASHMemory Features	426
6.3.13 EMC characteristic.....	427
6.3.14 IOPort Features.....	427
6.3.15 EXTRSTPort Features.....	428
6.3.16 ADC characteristic.....	429
6.3.17 ADC Accuracy.....	429
6.3.18 Temperature Sensor Characteristics	430
6.3.19 Operational Amplifier Characteristics	430
6.3.20 Comparator Characteristics	431

Figure catalog

picture5-1 Cortex-M0Processor Functional Block Diagram.....	43
picture5-2 SystickModule Structure Diagram.....	45
picture5-3 systickCounting Timing Diagram.....	46
picture5-4Block diagram of power module.....	50
picture5-5Waveform schematic diagram of power-on reset and power-off reset.....	53
picture5-6Schematic diagram of external pin reset waveform.....	54
picture5-7 SLEEPschematic diagram of mode entry and exit.....	59
picture5-8 DEEPSLEEPschematic diagram of mode entry and exit.....	61
picture5-9 STOPschematic diagram of mode entry and exit.....	62
picture5-10Chip clock structure diagram.....	72
picture5-11External high-frequency crystal oscillator related hardware configuration diagram.....	73
picture5-12External clock related hardware configuration diagram.....	74
picture5-13External low-frequency crystal oscillator related hardware configuration diagram.....	77
picture5-14 PORTCONBlock Diagram of Module Structure.....	88
picture5-15 IOSchematic diagram of input function selection.....	89
picture5-16 IOSchematic diagram of output function selection.....	90
picture5-17Analog signal connectionIOSchematic	91
picture5-18 IOMultiplexed asana_signal_bFunctional Schematic of Analog Signals.....	91
picture5-19 IOSchematic diagram of push-pull output mode.....	92
picture5-20 IOSchematic diagram of open-drain output mode.....	93
picture5-21 GPIOsModule System Block Diagram.....	110
picture5-22 GPIOsBlock Diagram of Module Structure.....	111
picture5-23 GPIOsOutput Timing Diagram	112
picture5-24 GPIOsHigh Level Interrupt Timing Diagram.....	113
picture5-25 GPIOsTiming Diagram of Low Level Triggered Interrupt.....	114
picture5-26 GPIOsRising edge trigger interrupt timing diagram.....	115
picture5-27 GPIOsTiming Diagram of Falling Edge Triggered Interrupt.....	116
picture5-28 GPIOsTiming Diagram for Dual-Edge Triggered Interrupt.....	117
picture5-29 GPIOsOperation flow chart.....	118
picture5-30 TIMERBASEModule System Block Diagram.....	122
picture5-31 TIMERBASEBlock Diagram of Module Structure.....	123
picture5-32 TIMERBASEHigh Counter Count Timing Diagram.....	124
picture5-33 TIMERBASELow Counter Count Timing Diagram.....	125
picture5-34 TIMERBASETiming diagram of counter frequency division counting.....	126
picture5-35 TIMERBASEInterrupt flag and interrupt diagram.....	127
picture5-36 TIMERBASEOperating procedures.....	128
picture5-37 TIMERPLUSSystem Block Diagram.....	133
picture5-38 TIMERPLUSStructure diagram.....	136
picture5-39 TIMERPLUSschematic diagram of clock source selection.....	138
picture5-40 TIMERPLUS pcIkCounter Timing Diagram under Divided Clock	138
picture5-41 TIMERPLUSCounter Timing Diagram in Timing Mode.....	139

picture5-42 TIMERPLUSRising edge valid timing diagram in counting mode.....	140
picture5-43 TIMERPLUSTiming diagram of falling edge valid in counting mode.....	141
picture5-44 TIMERPLUSDouble-edge active timing diagram in counting mode.....	142
picture5-45 TIMERPLUSTiming Diagram for Active Rising Edge of Input Capture.....	142
picture5-46 TIMERPLUSInput Capture Falling Edge Valid Timing Diagram.....	143
picture5-47 TIMERPLUSTiming diagram of double-edge active under input capture.....	144
picture5-48 TIMERPLUSexistHALLMode Timing Diagram	145
picture5-49 TIMERPLUSInterrupt flag and interrupt diagram.....	146
picture5-50 TIMERPLUSTimer Mode Operation Flowchart.....	148
picture5-51 TIMERPLUSFlow chart of counting mode operation.....	151
picture5-52 TIMERPLUSInput Capture Mode Operation Flowchart.....	154
picture5-53 TIMERPLUS HALLMode Operation Flowchart.....	157
picture5-54 IWDTSystem Block Diagram.....	166
picture5-55 IWDTBlock Diagram of Module Structure.....	167
picture5-56 IWDTInterrupt Generation Diagram	168
picture5-57 IWDTFeed the dog and reset before the first interruption.....	168
picture5-58 IWDTFeed the dog and reset before the second interruption.....	169
picture5-59 IWDTInterrupt Generation Operation Flowchart.....	170
picture5-60 WWDTModule System Block Diagram.....	173
picture5-61 WWDTBlock Diagram of Module Structure.....	174
picture5-62 WWDTInterrupt generation and dog feeding interval.....	175
picture5-63 WWDTFeed the dog and reset.....	176
picture5-64 PWMBASEModule System Block Diagram.....	180
picture5-65 PWMBASEBlock Diagram of Module Structure.....	183
picture5-66 PWMBASETiming Diagrams for Different Cycles and Turnover Points.....	184
picture5-67 PWMBASE 1Frequency division timing diagram.....	184
picture5-68 PWMBASE 2Frequency division timing diagram.....	185
picture5-69 PWMBASE 6Frequency division timing diagram.....	185
picture5-70 PWMBASEOutput Enable Timing Diagram.....	185
picture5-71 PWMBASEOutput Toggle Timing Diagram	186
picture5-72 PWMBASEOutput Interrupt Timing Diagram.....	187
picture5-73 PWMBASEOperation flow chart.....	188
picture5-74 PWMPLUSSystem Block Diagram.....	194
picture5-75 PWMPLUSStructure diagram.....	197
picture5-76 PWMPLUSIn edge-aligned mode, count up, single output, the initial level is0Timing diagram of199 picture 5-77	199
PWMPLUSIn edge-aligned mode, count up, single output, the initial level is1Timing diagram of200 picture5-78	200
PWMPLUSIn edge-aligned mode, count up and output circularly, the initial level is0Timing diagram of201 picture5-79	201
PWMPLUSIn edge-aligned mode, count up and output circularly, the initial level is1Timing diagram of201 picture5-80	201
PWMPLUSIn edge-aligned mode, count down, single output, the initial level is0Timing diagram of202 picture5-81	202
PWMPLUSIn edge-aligned mode, count down, single output, the initial level is1Timing diagram of202 picture5-82	202
PWMPLUSIn edge-aligned mode, count down and output circularly, the initial level is0Timing diagram of203 picture5-83	203
PWMPLUSIn edge-aligned mode, count down and output circularly, the initial level is1Timing diagram of203 picture5-84	203
PWMPLUSIn center-aligned mode, count up, single output, the initial level is0Timing diagram of204 picture5-85	204
PWMPLUSIn center-aligned mode, count up, single output, the initial level is1Timing diagram of205	205

picture5-86 PWMPLUSIn center-aligned mode, count up and output circularly, the initial level is0Timing diagram of205 picture	
5-87 PWMPLUSIn center-aligned mode, count up and output circularly, the initial level is0Timing diagram of206 picture5-88	
PWMPLUSIn center-aligned mode, count down, single output, the initial level is0Timing diagram of207 picture5-89 PWMPLUSIn	
center-aligned mode, count down, single output, the initial level is1Timing diagram of207 picture5-90 PWMPLUSIn center-aligned	
mode, count down and output circularly, the initial level is0Timing diagram of208 picture5-91 PWMPLUSIn center-aligned mode,	
count down and output circularly, the initial level is1Timing diagram of208 picture5-92 PWMPLUSComplementary Output Timing	
Diagram with Dead Time Insertion.....	209
picture5-93 PWMPLUSDead zone length is greater than the positive pulse and smaller than the negative pulse timing diagram.....	209
picture5-94 PWMPLUSDead zone length is greater than the negative pulse and smaller than the positive pulse timing diagram.....	210
picture5-95 PWMPLUSTiming Diagram for Brake Case in Edge-Aligned Mode.....	210
picture5-96 PWMPLUSTiming Diagram for Braking in Center-Aligned Mode.....	211
picture5-97 PWMPLUSedge-aligned modeMASKSituation Sequence Diagram	211
picture5-98 PWMPLUSCenter Alignment ModeMASKSituation Sequence Diagram	212
picture5-99 PWMPLUSEdge-aligned mode, count up, the starting level is0Case flip point and cycle timing diagram	
.....	213
picture5-100 PWMPLUSEdge-aligned mode, count up, the starting level is1Case Toggle Point and Cycle Timing	
picture.....	213
picture5-101 PWMPLUSEdge-aligned mode, counting down, the starting level is0Case Toggle Point and Cycle Timing	
picture.....	214
picture5-102 PWMPLUSEdge-aligned mode, counting down, the starting level is1Case Toggle Point and Cycle Timing	
picture.....	214
picture5-103 PWMPLUSCenter-aligned mode, count up, the starting level is0Case Toggle Point and Cycle Timing	
picture.....	215
picture5-104 PWMPLUSCenter-aligned mode, count up, the starting level is1Case Toggle Point and Cycle Timing	
picture.....	215
picture5-105 PWMPLUSCenter-aligned mode, counting down, the starting level is0Case Toggle Point and Cycle Timing	
picture.....	216
picture5-106 PWMPLUSCenter-aligned mode, counting down, the starting level is1Case Toggle Point and Cycle Timing	
picture.....	216
picture5-107 PWMPLUSOutput Priority Map	217
picture5-108 PWMPLUSBrake Interrupt Timing Diagram.....	217
picture5-109 PWMPLUSToggle point, specific trigger point, cycle overflow interrupt and trigger signal timing diagram.....	218
picture5-110 PWMPLUSModule operation flow chart.....	220
picture5-111 RTCModule System Block Diagram.....	237
picture5-112 RTCBlock Diagram of Module Structure.....	238
picture5-113 RTCCCLKSelect a schematic	239
picture5-114 RTCInterrupt flag and interrupt diagram.....	241
picture5-115 UARTModule System Block Diagram.....	253
picture5-116 UARTBlock Diagram of Module Structure.....	255
picture5-117 UARTReceive timing and related status signal diagram.....	256
picture5-118 UARTSend timing and related status signal diagram.....	257
picture5-119 UARTSchematic diagram of automatic baud rate calculation.....	258
picture5-120 UARTSchematic diagram of data sampling.....	259
picture5-121 UARTAutomatic flow control connection diagram.....	260

picture5-122 UARTSchematic diagram of flow control reception.....	260
picture5-123 UARTSchematic diagram of flow control sending.....	261
picture5-124 UARTFirst Receive Timeout Interrupt	261
picture5-125 UARTSecond Receive Timeout Interrupt	262
picture5-126 UARTInterrupt flag and interrupt diagram.....	263
picture5-127 SPIModule System Block Diagram.....	272
picture5-128 SPIBlock Diagram of Module Structure.....	273
picture5-129 SPIData/Clock Timing Diagram (CPHA=0)	274
picture5-130 SPIData/Clock Timing Diagram (CPHA=1)	275
picture5-131 SPI SSNtiming diagram (CPHA=0)	275
picture5-132 SPI Master/SPI Slaveinterconnection.....	277
picture5-133 SPIInterrupt flag and interrupt diagram.....	278
picture5-134 SPItake overFIFOFull Interrupt	279
picture5-135 SPItake overFIFOHalf-full Interrupt	281
picture5-136 SPIsendFIFOHalf-full Interrupt	282
picture5-137 SPIsendFIFOEmpty Interrupt	283
picture5-138 SPIOperating procedures.....	285
picture5-139 IICBlock Diagram of Module Structure.....	294
picture5-140 IICData Transfer Diagram	295
picture5-141 IICFor host sending7bitAddress Addressed Slave Receive.....	296
picture5-142 IICThe master reads the slave immediately after the first byte.....	297
picture5-143 IICCombined format.....	297
picture5-144 IICFor host sending10bitAddress Addressed Slave Receive.....	297
picture5-145 IIChost receiving10bitAddress Addressed Slave Transmitter.....	298
picture5-146 IIC SCLOpen Drain Diagram	298
picture5-147 IICInterrupt flag and interrupt diagram.....	299
picture5-148 IIChost sending operation flow chart.....	301
picture5-149 IIChost receiving operation flow chart.....	303
picture5-150 IICSlave sending operation flow chart.....	306
picture5-151 IICSlave Receive Operation Flowchart.....	309
picture5-152 SARADCSystem Block Diagram.....	324
picture5-153 SARADCModule Structure Diagram.....	326
picture5-154 SARADCSchematic diagram of channel selection.....	327
picture5-155 SARADCSingle Channel Single Sampling Schematic	328
picture5-156 SARADCSchematic diagram of single-channel continuous sampling.....	329
picture5-157 SARADCMulti-Channel Single Sampling Schematic	330
picture5-158 SARADCSchematic diagram of multi-channel continuous sampling.....	332
picture5-159 SARADCExternal Trigger Selection	335
picture5-160 SARADCSchematic diagram of interrupt flags and interrupt signals.....	335
picture5-161 SARADCOperation flow chart.....	336
picture5-162Comparator Block Diagram.....	348
picture5-163Comparator Hysteresis Schematic	350
picture5-164Schematic diagram of the filter function of the comparator to eliminate glitches.....	350
picture5-165Comparator interrupt flag and interrupt diagram.....	351

picture5-166 Operational Amplifier Block Diagram.....	355
picture5-167 FLASHController structure diagram.....	358
picture5-168 Sector Erase Operation Flowchart.....	364
picture5-169 Single-word programming operation flow chart.....	366
picture5-170 Multi-word continuous programming operation flow chart.....	369
picture5-171 CRCModule System Block Diagram.....	375
picture5-172 CRCModule Structure Diagram.....	376
picture5-173 CRC Schematic diagram of code composition.....	377
picture5-174 CRC Operation flow chart.....	379
picture5-175 DMA Schematic diagram of controller system connection.....	382
picture5-176 DMA Schematic diagram of the module structure.....	384
picture5-177 DMA Data format configuration table of source address and destination address.....	389
picture5-178 DMA Interrupt Description Table	390
picture5-179 DMA Each channel corresponds to the peripheral request mapping relationship.....	390
picture5-180 DMA Source Side Peripheral Request Mapping Diagram.....	392
picture5-181 DMA Destination Peripheral Request Mapping Diagram.....	393
picture5-182 DMA work flow chart.....	394
picture5-183 AES System Block Diagram.....	402
picture5-184 AES Block Diagram of Module Structure.....	404
picture5-185 AES Encryption schematic diagram of the electronic codebook algorithm.....	405
picture5-186 AES Decryption schematic diagram of the electronic codebook algorithm.....	405
picture5-187 AES Encryption Schematic of Cipher Block Chaining Algorithm.....	406
picture5-188 AES Schematic diagram of the decryption of the cipher block chaining algorithm.....	407
picture5-189 AES Encryption Schematic of Counter Mode Chaining Algorithm.....	408
picture5-190 AES Decryption Schematic of Counter Mode Chaining Algorithm.....	409
picture5-191 AES Displacement Schematic	411

table directory

surface2-1 32G030seriesMCUSelection table.....	26
surface4-1Pin definition.....	30
surface4-2Pin Alternate Function	36
surface5-1Address Space Mapping.....	39
surface5-6 32G030Interrupt Vector Table	41
surface5-8 GPIOsHigh level trigger interrupt timing parameters.....	113
surface5-9 GPIOsLow level trigger interrupt timing parameters.....	114
surface5-10 GPIOsRising edge trigger interrupt timing parameters.....	115
surface5-11 GPIOsFalling edge trigger interrupt timing parameters.....	116
surface1Voltage characteristics.....	421
surface2Current Characteristics	421
surface3Temperature Characteristics	422
surface4Shared Working Conditions Sheet.....	422
surface5Power-On and Power-Down Operating Conditions.....	422
surface6RESET AND POWER CONTROL MODULE CHARACTERISTICS.....	422
surface7Supply Current Characteristics	423

revision history

Version	date	author	Remark
1.0	2021/11/20	Zhou Liye	first edition
1.1	2021/12/15	Zhou Liye	Optimize instructions for use
1.2	2022/01/20	Yang Weijia	Further optimize the instruction manual and structure diagram
1.21	2022/02/16	Wu Weinan	Format correction content review
1.22	2022/02/21	Sun Haojun	Corrected formatting, updated CM related information
1.23	2022/02/21	Sun Haojun	Update the correct register table



1. Introduction

1.1 overview

This product is embedded ARM Cortex-M0 core, the highest operating frequency can reach 72MHz, support PLLs, built-in high-speed storage, rich enhanced IO ports and various peripherals. This product includes up to 40 general IO, 1 individual (14 pins) 12-bit ADC, 4 individual 16-bit advanced timers with functions such as input capture and periodic pulse output, 6 individual 16-bit Basic Timing of Bits device, 1 individual 20-bit independent watchdog, 1 individual 7-bit window watchdog, 6 Road Independence Basic PWM waveform generator, 6 Road Independent Senior PWM waveform generator with support for dead zone and complementary functions, 2 individual SPI, 2 individual IIC, 3 individual UART, 1 mark Accurate RTC, 1 individual CRC, 3 analog comparators, 1 temperature sensor, 2 analog operational amplifiers, 1 individual 128-bit AES.

The working voltage of this product series is 2.0V-3.6V, the operating temperature is -40°C-105°C. A variety of power-saving working modes to ensure low power application requirements.

This product is suitable for the following applications: Internet of Things, smart home, industrial control, instrumentation, wearable devices, products such as small home appliances, motor control, medical and handheld devices.

1.2 Product Features

-Kernel and System

- 32bitARM Cortex-M0processor core
- The highest operating frequency is72MHz
- twenty fourbit system tick timer
- Integrated Nested Vectored Interrupt Controller (NVIC)providing up to32interruption
- passSWDInterface burning program

-memory

- built-in64KbyteFLASHmemory as program memory
- built-in16KbyteRAMas a data store

-Clock, Reset and Power Management

- 2.0V-3.6Vsupply voltage
- Power-on/Power-off reset (POR/PDR), watchdog reset, off-chip dedicated pin reset (EXTRST)
- built-in4-32MHzHigh Frequency Crystal Oscillator Driver
- built-in32768HzLow Frequency Crystal Oscillator Driver
- Built-in factory-tuned48MHzhigh frequencyRCoscillator
- Built-in factory-tuned32768Hzlow frequencyRCoscillator
- built-inPLLscircuit that supports up to72MHzfrequency
- low power consumption

supportSLEEPmodel,DEEPSLEEPmode andSTOPMode, a total of three system low power consumption modes

-SARADC

- 14aisle12bit SARADC
- The sampling rate can reach2.4M
- For supply voltage, temperature and external signal sampling
- Support single mode and continuous mode

- Support hardware averaging function, can be configured as 1,2,4,8 averaging
- support ADC. The clock is configurable, and the system clock can be selected 1,2,4,8 frequency division
- Software trigger and hardware trigger optional
- Supports multiple interrupts

- GPIOs

- up to 40 individual IO mouth
- Configurable for the following modes: floating input, pull-up input, pull-down input, push-pull output, open-drain output, analog

IO

- Flexible interrupt configuration, can be configured as level trigger and edge trigger, level trigger can be set as low level or high level
Level, edge trigger can be set to rising edge, falling edge and double edge
- each GPIO supports both button wake-up function, which can be configured as rising edge wake-up and falling edge wake-up
- most IOs support 5V level input tolerance

- CRC

- support pair 8, 16, 32 bit data CRC operation
- Support for input data and output CRC values are processed in multiple data formats
- support CRC polynomial:

$$\begin{aligned} &x^{8+x^2+x+1}, x^{16+x^{12}+x^5+1}, x^{16+x^{15}+x^2+1}, x^{32+x^{26}+x^{23}+x^{22}+x^{16}+} \\ &x^{12+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x^1} \end{aligned}$$

- AES

- 128 bit AES, supports ECB, CBC and CTR model

- DMA

- maximum support 4 aisle
- The source address device and the destination address device can configure the transmission data width separately 8, 16 and 32 bits, source address and destination address must be aligned according to the data transfer width

- Each channel can be configured to transfer up to 4096 bytes of data. Every time each channel transmits a piece of data, the channel releases the total line, the circuit will re-judge the priority of each channel, and the channel with higher priority will be transmitted first
- Channels can be configured for incremental or fixed address mode
- Support memory-to-memory, memory-to-peripheral, peripheral-to-memory
- Supports cyclic restart DMA
- Support configurable priority for each channel, independent interrupt for each channel
- Comparators
 - 3-wide comparator
 - Rail to Rail
 - Different hysteresis voltages can be configured
 - Can be combined into a window comparator
- Operational Amplifier
 - 2 operational amplifier
 - Rail to Rail
 - 3MHz bandwidth
 - low offset voltage
 - The output pins can optionally be configured to ADC channel direct acquisition
- Communication Interface
 - 2 individual SPI Interface, configurable master-slave mode, programmable clock polarity and phase, configurable master mode rate, up to 4 times the system clock frequency division, configurable data transmission sequence, independent read and write data registers, receive and send adopt respectively 8-class FIFO caching mechanism, with DMA transfer function

- 3 individual UART Interface, support full-duplex mode, support 8/9 Bit data format selection, configurable parity bit odd parity parity parity often 1, support 1 Bit stop bit, send delay time configurable, support send completion interrupt, receive complete interrupt, receive timeout interrupt, support automatic baud rate detection, with hardware flow control function, use 8 class FIFO caching mechanism, with DMA transfer function
- 2 individual I2C Interface, support master-slave mode, support 3 modes: Standard-mode(100kbps), Fast-mode (400kbps), Fast-mode Plus(1Mbps) , support in host mode clock synchronization, SCL Clock duty cycle is configurable, slave mode supports slave address mask, supports 7bit and 10bit two address modes

-timer

- 6 individual 16-bit basic timer, count clock support 1-65536 Frequency division, only supports timing function
- 4 individual 16-bit advanced timer, count clock support 1-65536 divider with timing, counting, input capture and Periodic pulse output function, support HALL Function
- 1 individual 20-bit independent watchdog timer, 1 individual 7 Windowed Watchdog Timer
- 6 broad independent 16-bit basic PWM Waveform generator, count clock support 1-65536 Frequency division, support flip point Interrupt and cycle overflow interrupt
- 6 broad independent 16-bit Senior PWM Waveform generator, count clock support 1-65536 Frequency division, support dead zone, Complementary and brake functions, support rising edge counting or falling edge counting, support edge-aligned or center-aligned waveform output Output, support idle level, counting start level and output level inversion configurable, support inversion point interrupt, period Overflow interrupt and specific trigger point interrupt
- 1 standard RTC, with real-time clock, alarm clock and calendar functions, and can automatically solve the problem of leap year, can input out with RTC synchronized half-second, second, minute, hour, day, alarm, etc. interrupts, RTC The counting clock can choose two kinds of time Zhong Yuan RCL F and XTAL.

-UUID 128 chip unique identifier

-environment

- Operating temperature: -40°C-105°C

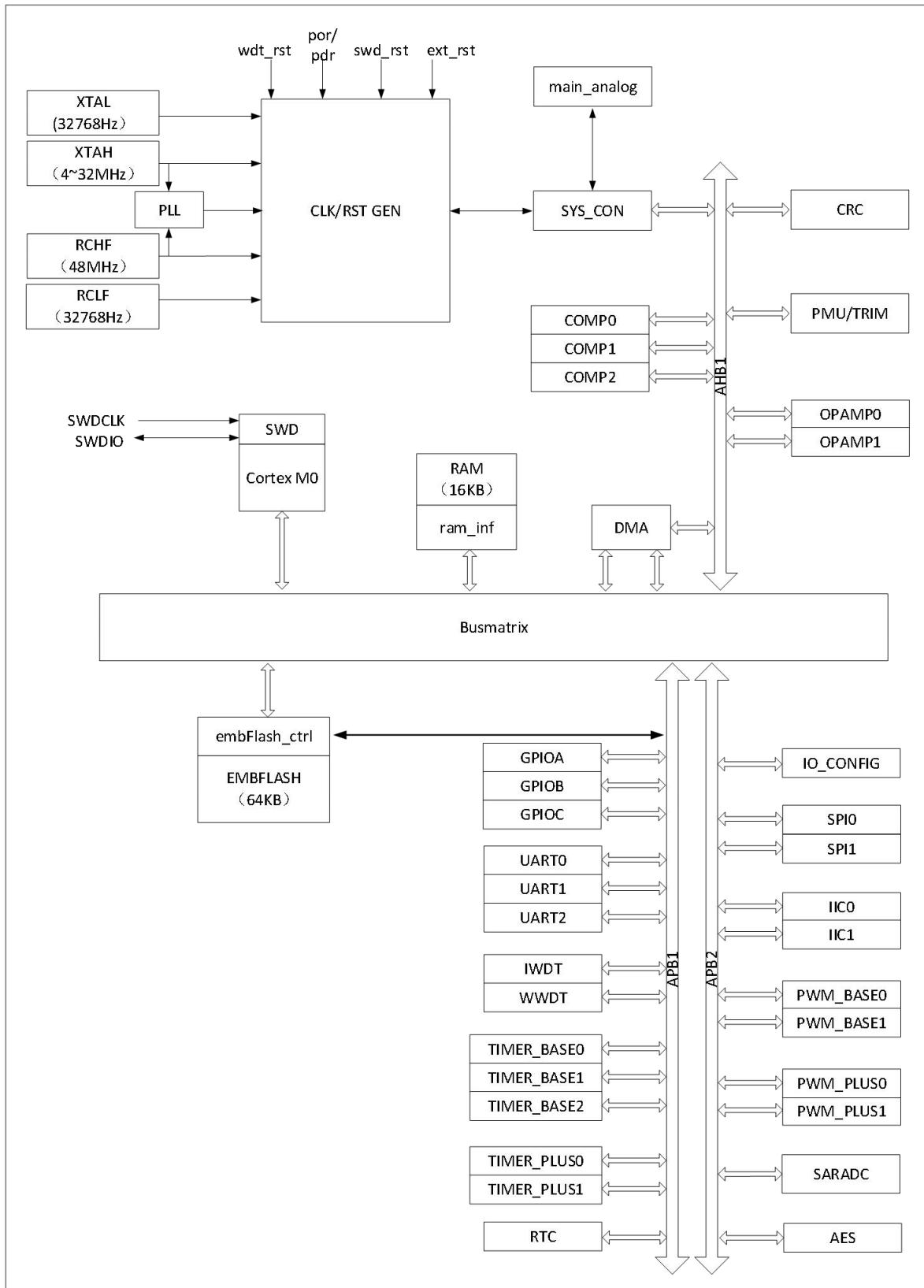
- Storage temperature:-50°C-150°C
 - Humidity level:MSL3
- encapsulation
- LQFP48,LQFP32,TSSOP20wait

2.Selection Guide

surface2-1 32G030seriesMCUSelection table

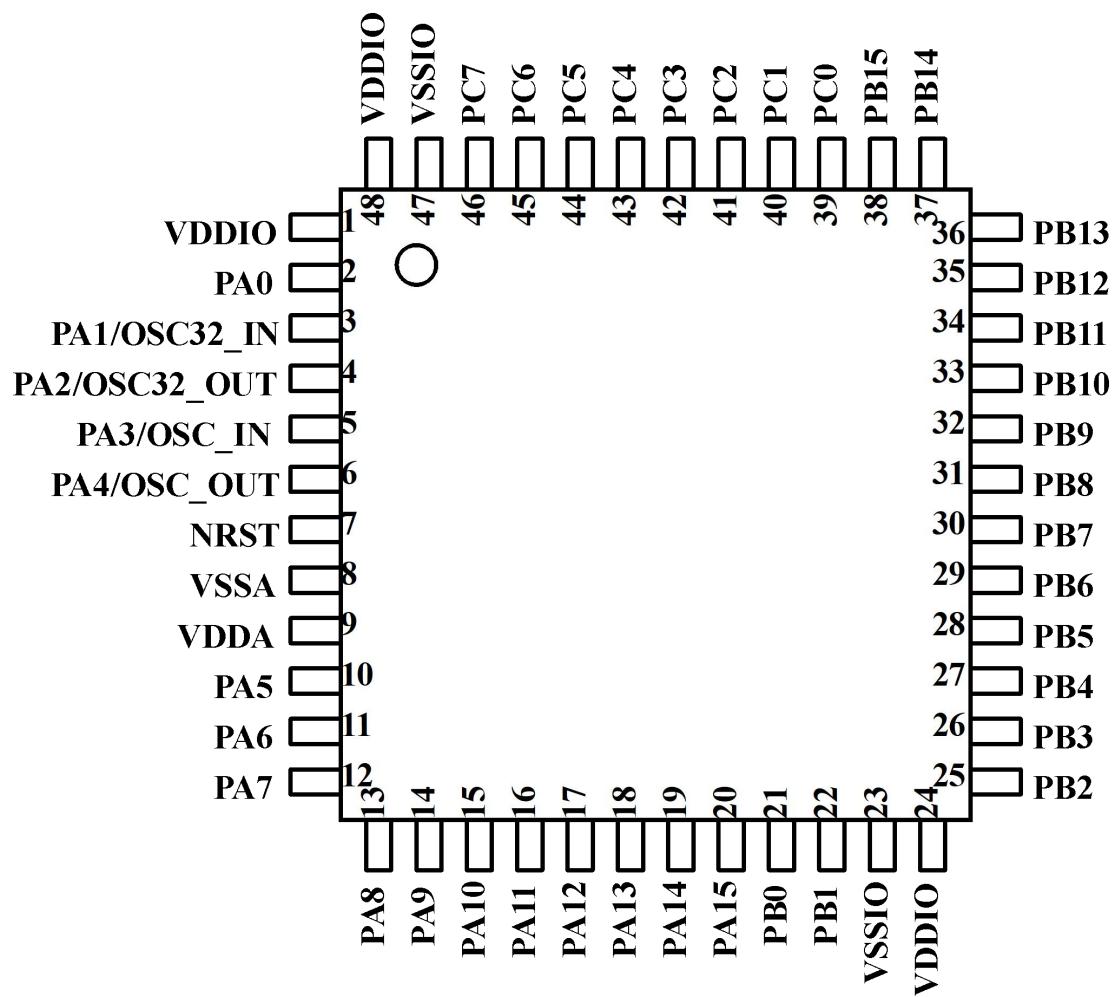
part Number	Voltage (V)	FLASH (KB)	RAM (KB)	IO	Timer ERB ASE	Timer ERP LUS	PW MB ASE	PW MPL US	IWDT	ADC	UART	SPI	IIC	package
DP32G030LQ48	2.0V-3.6V	64	16	40	3	2	2	2	1	1	3	2	2	LQFP48
DP32G030LQ32	2.0V-3.6V	64	16	26	3	2	2	2	1	1	3	2	2	LQFP32
DP32G030TS20	2.0V-3.6V	64	16	16	3	2	2	2	1	1	3	2	2	TSSOP20

3.Chip Structure Block Diagram

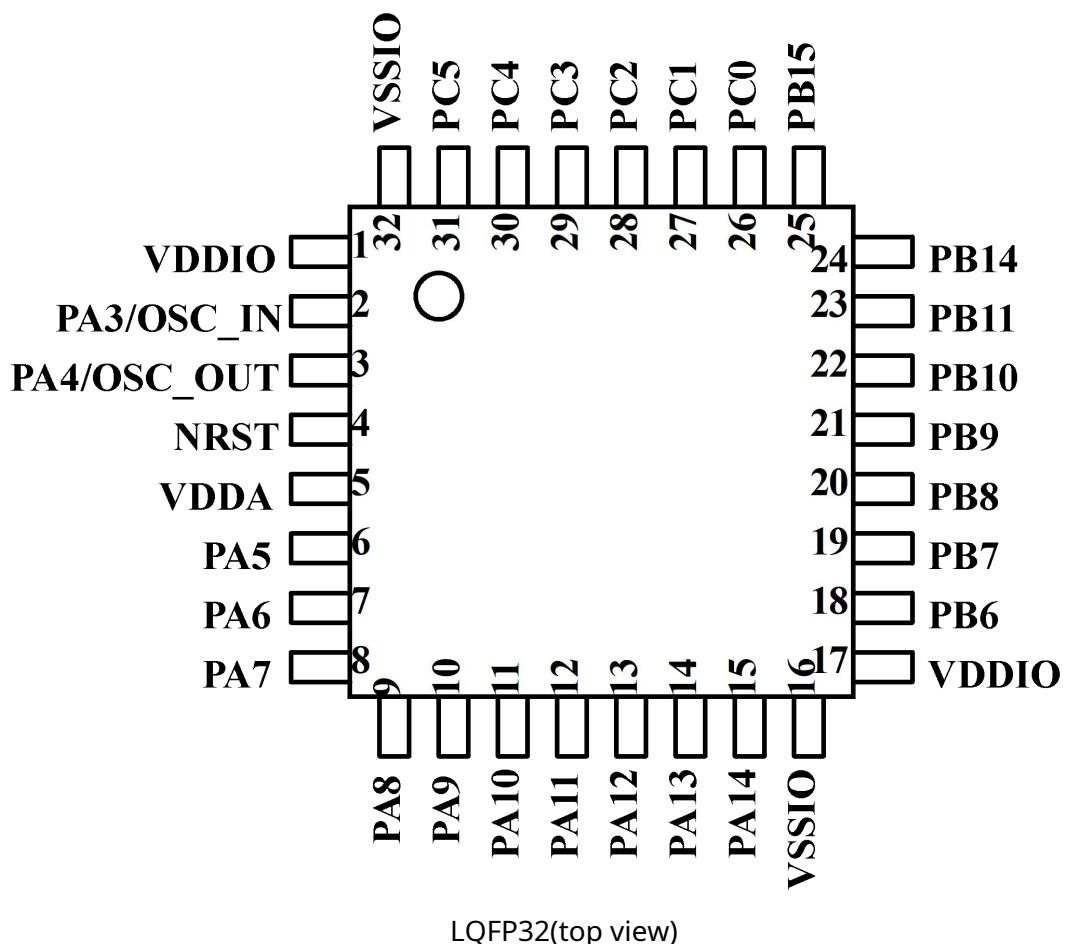


4.pin definition

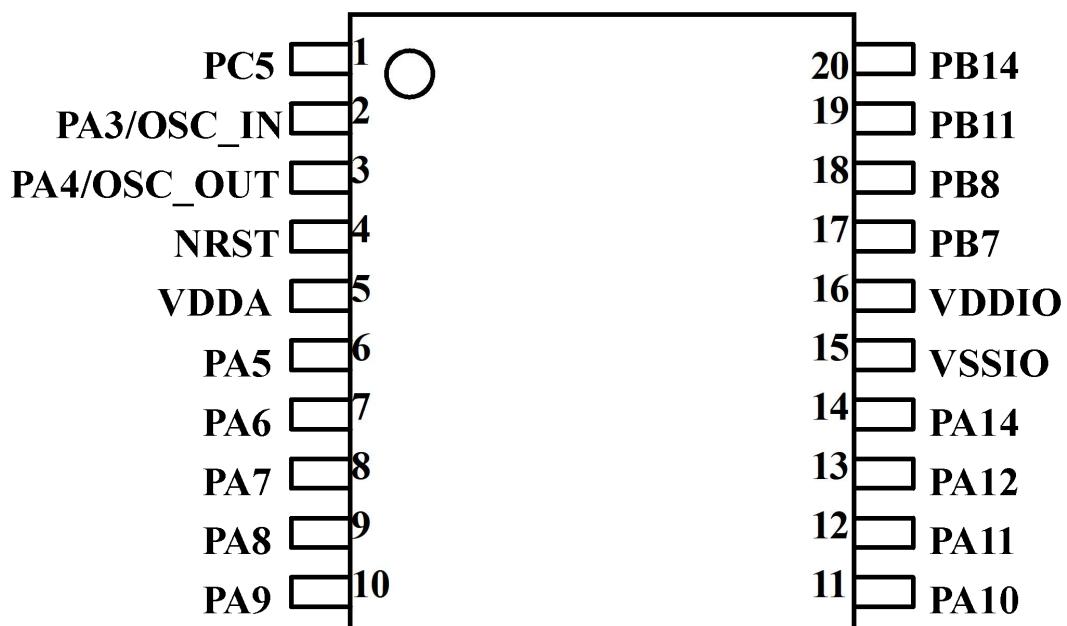
4.1 Package form



LQFP48(top view)



LQFP32(top view)



TSSOP20(top view)



4.2multiplexing pin

surface4-1pin definition

pin serial number	pin serial number	pin serial number	pin definition	kind type	multiplexing function	describe
LQ48	LQ32	TS20				
1 twenty four 48	1 17	16	VDDIO	S		2.0V~3.6Vpower input pin
20 twenty three	16 32	15	VSSIO	S		power ground
9	5	5	VDDA	S		2.3V~3.6Vpower input pin
8			VSSA	S		power ground
7	4	4	NRST	I/O		hardware reset pin
2			PA0	I/O	GPIO_PA0	PA0:numberGPIOsFunction pin
					PWMP1_PLUS0	PWMP1_PLUS0:advancedPWM1The input pulse pin of the0
					PWMP0_PLUS1	PWMP0_PLUS1:advancedPWM0The input pulse pin of the1
					tm	tm:RTC 1/2Second mark output pin
					WAKEUP0	WAKEUP0: Wake-up pin0
3			PA1	I/O	GPIO_PA1	PA1:numberGPIOsFunction pin
4			PA2	I/O	XTAL_XI	XTAL_XI: Input pin of low frequency crystal oscillator
5	2	2	PA3	I/O	GPIO_PA2	PA2:numberGPIOsFunction pin
					XTAL_XO	XTAL_XO: Output pin of low frequency crystal oscillator
					GPIO_PA3	PA3:numberGPIOsFunction pin
					CMP0_VN	CMP0_VN:Comparators0ofNinput pin
					XTAH_XI	XTAH_XI: Input pin of high frequency crystal oscillator
6	3	3	PA4	I/O	GPIO_PA4	PA4:numberGPIOsFunction pin
					CMP0_VP	CMP0_VP:Comparators0ofPinput pin
					XTAH_XO	XTAH_XO: Output pin of high frequency crystal oscillator
10	6	6	PA5	I/O	GPIO_PA5	PA5:numberGPIOsFunction pin
					UART1_CTS	UART1_CTS: serial port1ofCTSpin
					PWMP1_PLUS1	PWMP1_PLUS1:advancedPWM1The input pulse pin of the1
					TIMERP1_IN0	TIMERP1_IN0: Advanced Timer1input of0pin
					TIMERP1_OUT_L	TIMERP1_OUT_L: Advanced Timer1ofLoutput pin
					WAKEUP1	WAKEUP1: Wake-up pin1

					SARADC_CH0	SARADC_CH0:SARADCchannel0pin
11	7	7	PA6	I/O	GPIO_PA6	PA6:numberGPIOsFunction pin
					UART1_RTS	UART1_RTS: serial port1ofRTSpin
					TIMERP1_IN1	TIMERP1_IN1: Advanced Timer1input of1pin
					TIMERP1_OUT_H	TIMERP1_OUT_H: Advanced Timer1ofhoutput pin
					SARADC_CH1	SARADC_CH1:SARADCchannel1pin
					OPA0_OUT	OPA0_OUT:Operational Amplifier0output pin
12	8	8	PA7	I/O	GPIO_PA7	PA7:numberGPIOsFunction pin
					UART1_TX	UART1_TX: serial port1the send pin of
					TIMERP0_IN0	TIMERP0_IN0: Advanced Timer0input of0pin
					TIMERP0_OUT_L	TIMERP0_OUT_L: Advanced Timer0ofLoutput pin
					SARADC_CH2	SARADC_CH2:SARADCchannel2pin
					OPA0_VP	OPA0_VP:Operational Amplifier0ofPinput pin
13	9	9	PA8	I/O	GPIO_PA8	PA8:numberGPIOsFunction pin
					UART1_RX	UART1_RX: serial port1The receive pin of
					TIMERP0_IN1	TIMERP0_IN1: Advanced Timer0input of1pin
					TIMERP0_OUT_H	TIMERP0_OUT_H: Advanced Timer0ofhoutput pin
					SARADC_CH3	SARADC_CH3:SARADCchannel3pin
					OPA0_VN	OPA0_VN:Operational Amplifier0ofNinput pin
14	10	10	PA9	I/O	GPIO_PA9	PA9:numberGPIOsFunction pin
					SPI0_SS_N	SPI0_SS_N:SPI0chip select pin
					TIMERP1_IN0	TIMERP1_IN0: Advanced Timer1input of0pin
					TIMERP1_OUT_L	TIMERP1_OUT_L: Advanced Timer1ofLoutput pin
					tm	tm:RTC 1/2Second mark output pin
					SARADC_CH4	SARADC_CH4:SARADCchannel4pin
					CMP1_VN	CMP1_VN:Comparators1ofNinput pin
15	11	11	PA10	I/O	GPIO_PA10	PA10:numberGPIOsFunction pin
					SPI0_CLK	SPI0_CLK:SPI0the clock pin
					SARADC_CH5	SARADC_CH5:SARADCchannel5pin
					CMP1_VP	CMP1_VP:Comparators1ofPinput pin
16	12	12	PA11	I/O	GPIO_PA11	PA11:numberGPIOsFunction pin
					SPI0_MISO	SPI0_MISO:SPI0The host receive pin of the
					PWMB0_CH0	PWMB0_CH0: basicPWM0channel0pin
					PWMP0_BRK0	PWMP0_BRK0:advancedPWM0ofBRAKE0pin
					TIMERP1_IN1	TIMERP1_IN1: Advanced Timer1input of1pin
					TIMERP1_OUT_H	TIMERP1_OUT_H: Advanced Timer1ofhoutput pin
					SARADC_CH6	SARADC_CH6:SARADCchannel6pin
17	13	13	PA12	I/O	GPIO_PA12	PA12:numberGPIOsFunction pin
					SPI0_MOSI	SPI0_MOSI:SPI0The master transmit pin of the
					PWMB0_CH1	PWMB0_CH1: basicPWM0channel1pin
					PWMP0_CH0N	PWMP0_CH0N:advancedPWM0channel0Npin

					TIMERP0_IN0	TIMERP0_IN0: Advanced Timer0input of0pin
					TIMERP0_OUT_L	TIMERP0_OUT_L: Advanced Timer0ofLoutput pin
					SARADC_CH7	SARADC_CH7:SARADCchannel7pin
18	14	PA13	I/O	GPIO_PA13	PA13:numberGPIOsFunction pin	
				PWMB0_CH2	PWMB0_CH2: basicPWM0channel2pin	
				PWMP0_CH1N	PWMP0_CH1N:advancedPWM0channel1Npin	
				TIMERP0_IN1	TIMERP0_IN1: Advanced Timer0input of1pin	
				TIMERP0_OUT_H	TIMERP0_OUT_H: Advanced Timer0ofhoutput pin	
				SARADC_CH8	SARADC_CH8:SARADCchannel8pin	
19	15	PA14	I/O	GPIO_PA14	PA14:numberGPIOsFunction pin	
				PWMB1_CH0	PWMB1_CH0: basicPWM1channel0pin	
				PWMP0_CH2N	PWMP0_CH2N:advancedPWM0channel2Npin	
				TIMERP1_IN0	TIMERP1_IN0: Advanced Timer1input of0pin	
				TIMERP1_OUT_L	TIMERP1_OUT_L: Advanced Timer1ofLoutput pin	
				SARADC_CH9	SARADC_CH9:SARADCchannel9pin	
20		PA15	I/O	GPIO_PA15	PA15:numberGPIOsFunction pin	
				PWMB1_CH1	PWMB1_CH1: basicPWM1channel1pin	
				PWMP0_CH0	PWMP0_CH0:advancedPWM0channel0pin	
				TIMERP1_IN1	TIMERP1_IN1: Advanced Timer1input of1pin	
				TIMERP1_OUT_H	TIMERP1_OUT_H: Advanced Timer1ofhoutput pin	
twenty one		PB0	I/O	GPIO_PB0	PB0:numberGPIOsFunction pin	
				UART2_TX	UART2_TX: serial port2the send pin of	
				IIC0_SCL	IIC0_SCL:IIC0the clock pin	
				PWMB1_CH2	PWMB1_CH2: basicPWM1channel2pin	
				PWMP0_CH1	PWMP0_CH1:advancedPWM0channel1pin	
twenty two		PB1	I/O	GPIO_PB1	PB1:numberGPIOsFunction pin	
				UART2_RX	UART2_RX: serial port2The receive pin of	
				IIC0_SDA	IIC0_SDA:IIC0The data pin	
				PWMP0_CH2	PWMP0_CH2:advancedPWM0channel2pin	
25		PB2	I/O	GPIO_PB2	PB2:numberGPIOsFunction pin	
				SPI1_SS	SPI1_SS:SPI1chip select pin	
				PWMP0_BRK1	PWMP0_BRK1:advancedPWM0ofBRAKE1pin	
				TIMERP1_HALL0	TIMERP1_HALL0: Advanced Timer1ofHALL0pin	
26		PB3	I/O	GPIO_PB3	PB3:numberGPIOsFunction pin	
				SPI1_CLK	SPI1_CLK:SPI1the clock pin	
				IIC1_SDA	IIC1_SDA:IIC1The data pin	
				PWMP0_CH0N	PWMP0_CH0N:advancedPWM0channel0Npin	
				TIMERP1_HALL1	TIMERP1_HALL1: Advanced Timer1ofHALL1pin	
27		PB4	I/O	GPIO_PB4	PB4:numberGPIOsFunction pin	
				SPI1_MISO	SPI1_MISO:SPI1The host receive pin of the	
				IIC1_SCL	IIC1_SCL:IIC1the clock pin	
				PWMP1_CH0	PWMP1_CH0:advancedPWM1channel0pin	

					PWMP0_CH1N	PWMP0_CH1N:advancedPWM0channel1Npin
					TIMERP1_HALL2	TIMERP1_HALL2: Advanced Timer1ofHALL2pin
28			PB5	I/O	GPIO_PB5	PB5:numberGPIOsFunction pin
					SPI1_MOSI	SPI1_MOSI:SPI1The master transmit pin of the
					PWMP1_CH0N	PWMP1_CH0N:advancedPWM1channel0Npin
					PWMP0_CH2N	PWMP0_CH2N:advancedPWM0channel2Npin
					TIMERP0_IN0	TIMERP0_IN0: Advanced Timer0input of0pin
					TIMERP0_OUT_L	TIMERP0_OUT_L: Advanced Timer0ofLoutput pin
29	18		PB6	I/O	GPIO_PB6	PB6:numberGPIOsFunction pin
					PWMP0_CH0	PWMP0_CH0:advancedPWM0channel0pin
					TIMERP0_IN1	TIMERP0_IN1: Advanced Timer0input of1pin
					TIMERP0_OUT_H	TIMERP0_OUT_H: Advanced Timer0ofhoutput pin
30	19	17	PB7	I/O	GPIO_PB7	PB7:numberGPIOsFunction pin
					SPI0_SSN	SPI0_SSN:SPI0chip select pin
					UART0_TX	UART0_TX: serial port0the send pin of
					IIC0_SCL	IIC0_SCL:IIC0the clock pin
					PWMP1_BRK0	PWMP1_BRK0:advancedPWM1ofBRAKE0pin
					PWMP0_CH1	PWMP1_BRK0:advancedPWM1ofBRAKE0pin
31	20	18	PB8	I/O	GPIO_PB8	PB8:numberGPIOsFunction pin
					SPI0_CLK	SPI0_CLK:SPI0the clock pin
					UART0_RX	UART0_RX: serial port0The receive pin of
					IIC0_SDA	IIC0_SDA:IIC0The data pin PWMB0_CH0:
					PWMB0_CH0	basicPWM0channel0pin
					PWMP1_BRK1	PWMP1_BRK1:advancedPWM1ofBRAKE1pin
32	twenty one		PB9	I/O	PWMP0_CH2	PWMP0_CH2:advancedPWM0channel2pin
					GPIO_PB9	PB9:numberGPIOsFunction pin
					SPI0_MISO	SPI0_MISO:SPI0The host receive pin of the
					UART0_CTS	UART0_CTS: serial port0ofCTSpin
					PWMB0_CH1	PWMB0_CH1: basicPWM0channel1pin
					PWMP1_CH0	PWMP1_CH0:advancedPWM1channel0pin
					TIMERP1_IN1	TIMERP1_IN1: Advanced Timer1input of1pin
33	twenty two		PB10	I/O	TIMERP1_OUT_H	TIMERP1_OUT_H: Advanced Timer1ofhoutput pin
					GPIO_PB10	PB10:numberGPIOsFunction pin
					SPI0_MOSI	SPI0_MOSI:SPI0The master transmit pin of the
					UART0_RTS	UART0_RTS: serial port1ofRTSpin
					PWMB0_CH2	PWMB0_CH2: basicPWM0channel2pin
					PWMP1_CH1	PWMP1_CH1:advancedPWM1channel1pin
					PWMP0_PLUS0	PWMP0_PLUS0:advancedPWM0The input pulse pin of the0
					TIMERP1_IN0	TIMERP1_IN0: Advanced Timer1input of0pin
34	twenty three	19	PB11	I/O	TIMERP1_OUT_L	TIMERP1_OUT_L: Advanced Timer1ofLoutput pin
					GPIO_PB11	PB11:numberGPIOsFunction pin
					SWDIO	SWDIO:chipSWport data pin

					PWMP1_CH2	PWMP1_CH2:advancedPWM1channel2pin
					PWMP0_BRK2	PWMP0_BRK2:advancedPWM0ofBRAKE2pin
35		PB12	I/O	GPIO_PB12	PB12:numberGPIOsFunction pin	
				UART1_TX	UART1_TX: serial port1the send pin of	
				IIC1_SCL	IIC1_SCL:IIC1the clock pin	
				PWMP1_CH0N	PWMP1_CH0N:advancedPWM1channel0Npin	
36		PB13	I/O	GPIO_PB13	PB13:numberGPIOsFunction pin	
				UART1_RX	UART1_RX: serial port1The receive pin of	
				IIC1_SDA	IIC1_SDA:IIC1The data pin	
				PWMP1_CH1N	PWMP1_CH1N:advancedPWM1channel1Npin	
37	twenty four	20	PB14	GPIO_PB14	PB14:numberGPIOsFunction pin	
				SWCLK	SWCLK:chipSWport clock pin	
				UART2_TX	UART2_TX: serial port1the send pin of	
				PWMP1_CH2N	PWMP1_CH2N:advancedPWM1channel2Npin	
38	25		PB15	I/O	GPIO_PB15	PB15:numberGPIOsFunction pin
				SPI1_SS	SPI1_SS:SPI1chip select pin	
				UART2_RX	UART2_RX: serial port2The receive pin of	
39	26		PC0	I/O	GPIO_PC0	PC0:numberGPIOsFunction pin
				SPI1_CLK	SPI1_CLK:SPI1the clock pin	
				UART2_CTS	UART2_CTS: serial port2ofCTSpin	
				PWMB1_CH0	PWMB1_CH0: basicPWM1channel0pin	
40	27		PC1	I/O	GPIO_PC1	PC1:numberGPIOsFunction pin
				SPI1_MISO	SPI1_MISO:SPI1The host receive pin of the	
				UART2_RTS	UART2_RTS: serial port2ofRTSpin	
				PWMB1_CH1	PWMB1_CH1: basicPWM1channel1pin	
				TIMERPO_IN0	TIMERPO_IN0: Advanced Timer0input of0pin	
				TIMERPO_OUT_L	TIMERPO_OUT_L: Advanced Timer0ofLoutput pin	
41	28		PC2	I/O	GPIO_PC2	PC2:numberGPIOsFunction pin
				SPI1_MOSI	SPI1_MOSI:SPI1The master transmit pin of the	
				PWMB1_CH2	PWMB1_CH2: basicPWM1channel2pin	
				PWMP1_BRK2	PWMP1_BRK2:advancedPWM1ofBRAKE2pin	
				TIMERPO_IN1	TIMERPO_IN1: Advanced Timer0input of1pin	
				TIMERPO_OUT_H	TIMERPO_OUT_H: Advanced Timer0ofhoutput pin	
42	29		PC3	I/O	GPIO_PC3	PC3:numberGPIOsFunction pin
				UART0_TX	UART0_TX: serial port0the send pin of	
				IIC0_SCL	IIC0_SCL:IIC0the clock pin	
				PWMP1_CH1N	PWMP1_CH1N:advancedPWM1channel1Npin	
				TIMERPO_HALLO	TIMERPO_HALLO: Advanced Timer0ofHALLOpin	
				CMP2_VN	CMP2_VN:Comparators2ofNinput pin	
43	30		PC4	I/O	GPIO_PC4	PC4:numberGPIOsFunction pin
				UART0_RX	UART0_RX: serial port0The receive pin of	
				IIC0_SDA	IIC0_SDA:IIC0The data pin	

					PWMP1_CH2N	PWMP1_CH2N:advancedPWM1channel2Npin
					TIMERP0_HALL1	TIMERP0_HALL1: Advanced Timer0ofHALL1pin
					CMP2_VP	CMP2_VP:Comparators2ofPinput pin
44	31	1	PC5	I/O	GPIO_PC5	PC5:numberGPIOsFunction pin
					TIMERP0_HALL2	TIMERP0_HALL2: Advanced Timer0ofHALL2pin
					OPA1_VP	OPA1_VP:Operational Amplifier1ofPinput pin
45			PC6	I/O	GPIO_PC6	PC6:numberGPIOsFunction pin
					IIC1_SCL	IIC1_SCL:IIC1the clock pin
					PWMP1_CH1	PWMP1_CH1:advancedPWM1channel1pin
					TIMERP1_IN1	TIMERP1_IN1: Advanced Timer1input of1pin
					TIMERP1_OUT_H	TIMERP1_OUT_H: Advanced Timer1ofhoutput pin
					OPA1_VN	OPA1_VN:Operational Amplifier1ofNinput pin
46			PC7	I/O	GPIO_PC7	PC7:numberGPIOsFunction pin
					IIC1_SDA	IIC1_SDA:IIC1The data pin
					PWMP1_CH2	PWMP1_CH2:advancedPWM1channel2pin
					TIMERP1_IN0	TIMERP1_IN0: Advanced Timer1input of0pin
					TIMERP1_OUT_L	TIMERP1_OUT_L: Advanced Timer1ofLoutput pin
					OPA1_OUT	OPA1_OUT:Operational Amplifier1output pin

4.3Pin Alternate Function

surface4-2Pin Alternate Function

LQ48	pin name	SEL000	SEL001	SEL010	SEL011	SEL100	SEL101	SEL110	SEL111
2	PA0	GPIOA0	PWMP1_PLUS0	PWMP0_PLUS1	tm	WAKEUP0		---	
3	PA1	GPIOA1	XTAL_XI				---		
4	PA2	GPIOA2	XTAL_XO				---		
5	PA3	GPIOA3	CMP0_VN	XTAH_XI	---	---	---		
6	PA4	GPIOA4	CMP0_VP	XTAH_XO	---	---	---		
10	PA5	GPIOA5	UART1_CTS	PWMP1_PLUS1	TIMERP1_IN0	TIMERP1_OUT_L	WAKEUP1	SARADC_CH0	
11	PA6	GPIOA6	UART1_RTS	TIMERP0_IN0	TIMERP1_OUT_H	SARADC_CH1	OPA0_OUT		
12	PA7	GPIOA7	UART1_TX	BREAK_IN1	TIMERP0_OUT_L	SARADC_CH2	OPA0_VP		
13	PA8	GPIOA8	UART1_RX	TIMERP0_IN1	TIMERP0_OUT_H	SARADC_CH3	OPA0_VN	---	
14	PA9	GPIOA9	SPI0_SS_N	TIMERP1_IN0	TIMERP1_OUT_L	tm	SARADC_CH4	CMP1_VN	
15	PA10	GPIOA10	SPI0_CLK	SARADC_CH5	CMP1_VP			---	

16	PA11	GPIOA11	SPI0_MISO	PWMB0_CH0	PWMP0_BRAKE0	TIMERP1_IN1	TIMERP1_OUT_H	SARADC_CH6	
17	PA12	GPIOA12	SPI0_MOSI	PWMB0_CH1	PWMP0_CH0N	TIMERP0_IN0	TIMERP0_OUT_L	SARADC_CH7	
18	PA13	GPIOA13	PWMB0_CH2	PWMP0_CH1N	TIMERP0_IN1	TIMERP0_OUT_H	SARADC_CH8	---	
19	PA14	GPIOA14	PWMB1_CH0	PWMP0_CH2N	TIMERP1_IN0	TIMERP1_OUT_L	SARADC_CH9		
20	PA15	GPIOA15	PWMB1_CH1	PWMP0_CH0	TIMERP1_IN1	TIMERP1_OUT_H	---		
twenty one	PB0	GPIOB0	UART2_TX	IIC0_SCL	PWMB1_CH2	PWMP0_CH1			
twenty two	PB1	GPIOB1	UART2_RX	IIC0_SDA	PWMP0_CH2				
25	PB2	GPIOB2	SPI1_SSN	PWMP0_BRAKE1	TIMERP1_HALL0				
26	PB3	GPIOB3	SPI1_CLK	IIC1_SDA	PWMP0_CH0N	TIMERP1_HALL1			
27	PB4	GPIOB4	SPI1_MISO	IIC1_SCL	PWMP1_CH0	PWMP0_CH1N	TIMERP1_HALL2		
28	PB5	GPIOB5	SPI1_MOSI	PWMP1_CH0N	PWMP0_CH2N	TIMERP0_IN0	TIMERP0_OUT_L		
29	PB6	GPIOB6	PWMP0_CH0	TIMERP0_IN1	TIMERP0_OUT_H				
30	PB7	GPIOB7	SPI0_SSN	UART0_TX	IIC0_SCL	PWMP1_BRAKE0	PWMP0_CH1		
31	PB8	GPIOB8	SPI0_CLK	UART0_RX	IIC0_SDA	PWMB0_CH0	PWMP1_BRAKE1	PWMP0_CH2	
32	PB9	GPIOB9	SPI0_MISO	UART0_CTS	PWMB0_CH1	PWMP1_CH0	TIMERP1_IN1	TIMERP1_OUT_H	
33	PB10	GPIOB10	SPI0_MOSI	UART0_RTS	PWMB0_CH2	PWMP1_CH1	PWMP0_PLUS0	TIMERP1_IN0	TIMERP1_OUT_L
34	PB11	GPIOB11	SWDIO	PWMP1_CH2	PWMP0_BRAKE2				

35	PB12	GPIOB12	UART1_TX	IIC1_SCL	PWMP1_CH0N				
36	PB13	GPIOB13	UART1_RX	IIC1_SDA	PWMP1_CH1N				
37	PB14	GPIOB14	SWCLK	UART2_TX	PWMP1_CH2N				
38	PB15	GPIOB15	SPI1_SS_N	UART2_RX					
39	PC0	GPIOC0	SPI1_CLK	UART2_CTS	PWMB1_CH0				
40	PC1	GPIOC1	SPI1_MISO	UART2_RTS	PWMB1_CH1	TIMERP0_IN0	TIMERP0_OUT_L		
41	PC2	GPIOC2	SPI1_MOSI	PWMB1_CH2	PWMP1_BRAKE2	TIMERP0_IN1	TIMERP0_OUT_H		
42	PC3	GPIOC3	UART0_TX	IIC0_SCL	PWMP1_CH1N	TIMERP0_HALL0	CMP2_VN		
43	PC4	GPIOC4	UART0_RX	IIC0_SDA	PWMP1_CH2N	TIMERP0_HALL1	CMP2_VP		
44	PC5	GPIOC5	TIMERP0_HALL2	OPA1_VP					
45	PC6	GPIOC6	IIC1_SCL	PWMP1_CH1	TIMERP1_IN1	TIMERP1_OUT_H	OPA1_VN		
46	PC7	GPIOC7	IIC1_SDA	PWMP1_CH2	TIMERP1_IN0	TIMERP1_OUT_L	OPA1_OUT		

5.Functional description

5.1address space mapping

The controller is 32bit general-purpose controller that provides 4GByte addressing space, as shown in the table below. data grid

The format only supports little-endian mode, and the specific register layout and operation instructions of each module are described in detail in the following chapters.

surface5-1address space mapping

start	Finish	module
memory		
0x00000000	0x0000FFFF	FLASHdistrict
0x20000000	0x20003FFF	RAMdistrict
AHBbus peripherals		
0x40000000	0x400007FF	SYSCON
0x40000800	0x40000FFF	PMU
0x40001000	0x400017FF	DMA
0x40003000	0x400037FF	CRC
APB1bus peripherals		
0x40060000	0x400607FF	GPIOA
0x40060800	0x40060FFF	GPIOB
0x40061000	0x400617FF	GPIOC
0x40064000	0x400647FF	TIMER_BASE0
0x40064800	0x40064FFF	TIMER_BASE1
0x40065000	0x400657FF	TIMER_BASE2
0x40067000	0x400677FF	TIMER_PLUS0
0x40067800	0x40067FFF	TIMER_PLUS1
0x40069000	0x400697FF	RTC
0x4006A000	0x4006A7FF	IWDT
0x4006A800	0x4006AFFF	WWDT
0x4006B000	0x4006B7FF	UART0
0x4006B800	0x4006BFFF	UART1



0x4006C000	0x4006C7FF	UART2
0x4006F000	0x4006F7FF	FLASH_CTRL
APB2bus peripherals		
0x400B0000	0x400B07FF	PORTCON
0x400B1000	0x400B17FF	PWM_BASE0
0x400B1800	0x400B1FFF	PWM_BASE1
0x400B4000	0x400B47FF	PWM_PLUS0
0x400B4800	0x400B4FFF	PWM_PLUS1
0x400B8000	0x400B87FF	SPI0
0x400B8800	0x400B8FFF	SPI1
0x400B9000	0x400B97FF	IIC0
0x400B9800	0x400B9FFF	IIC1
0x400BA000	0x400BA7FF	SARADC
0x400BD000	0x400BD7FF	AES128

5.2 Memory division and authority control

There are two storage areas in this chip: 64KB FLASH and 16KB RAM.

64KB FLASH Used as program memory area with 2KB of NVRD district and 64KB of MAIN district. NVR The area is mainly used to store some data specific to our company, such as factory code information, TRIM data and product configuration information, etc. MAIN The area is used to store user programs and is completely open to users.

16KB RAM All are used as data area.

5.3 interrupt vector table

The interrupt vector table is as follows:

surface5-6 32G030 interrupt vector table

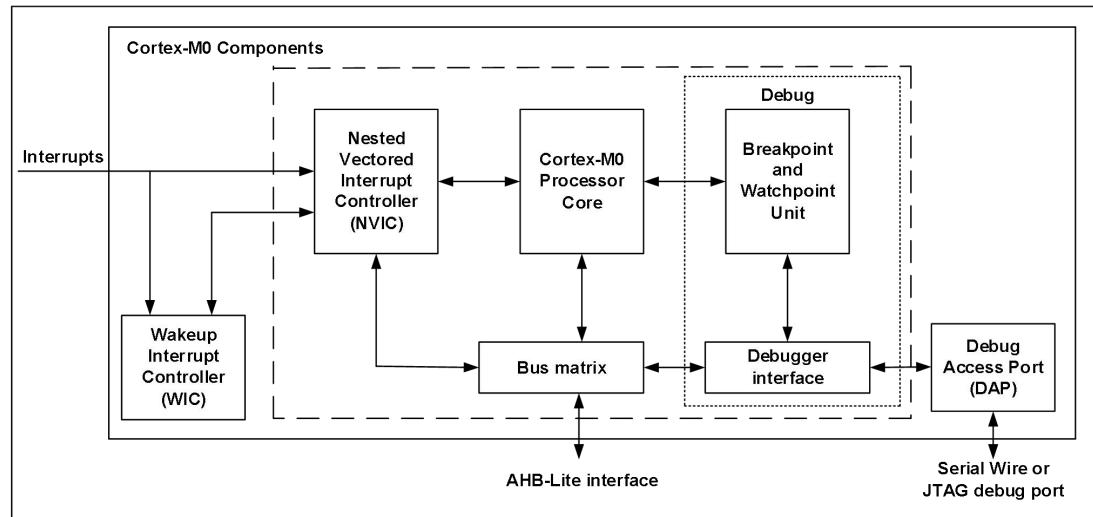
interrupt source	peripheral interrupt
0	WWDT
1	IWDT
2	RTC
3	DMA
4	SARADC
5	TIMER_BASE0
6	TIMER_BASE1
7	TIMER_PLUS0
8	TIMER_PLUS1
9	PWM_BASE0
10	PWM_BASE1
11	PWM_PLUS0
12	PWM_PLUS1
13	UART0
14	UART1
15	UART2
16	SPI0
17	SPI1
18	IIC0

19	IIC1
20	CMP
twenty one	TIMER_BASE2
twenty two	GPIOA5
twenty three	GPIOA6
twenty four	GPIOA7
25	GPIOB0
26	GPIOB1
27	GPIOC0
28	GPIOC1
29	GPIOA
30	GPIOB
31	GPIOC

5.4 ARM Cortex-M0kernel

5.4.1 overview

Cortex-M0processor is32bit multi-level configurableRISCprocessor. it hasAMBA AHB-LiteInterfaces and Nesting Vectored Interrupt Controller (NVIC)With an optional hardware debug function, it is possible to performThumbinstruction, and with otherCortex-M series compatible. This family of processors supports two modes of operationThreadmode andHandlermodel. When an exception occurs, Processor entersHandlermodel. Exception returns can only be made inHandlermode occurs. When reset, the processor will enter the Threadmode, the processor can also enter theThreadmodel. The figure below shows the various parts of the processor core functional module.



picture5-1 Cortex-M0Processor Functional Block Diagram

5.4.2 characteristic

- Processor Features:
 - ThumbInstruction Set
 - Thumb-2technology
 - 24-bit SYSTICKtimer
 - 32-bit hardware multiplier

-The system interface supports little endian (little-endian) data access

- Deterministic, fixed-latency interrupt handling
- Multiple load/store and multicycle multiply instructions can be discarded and restarted for fast interrupt handling
- and C Application Binary Interface Compatible Exception Mode (C-ABI) ARMv6-M(C-ABI) compatibility exception mode allows users to use pureCFunction to implement interrupt handling
- Use the wait for interrupt (WFI) go directly to sleep when returning from an interrupt sleep on exit characteristics can be entered into the low

Sleep Mode for Power Consumption

- NVIC characteristic:
 - 32 external interrupt inputs, each with 4 class priority
 - Non-maskable interrupt input (NMI)
 - Support level and pulse trigger interrupt
 - interrupt wake-up controller (WIC), supports very low power sleep mode
- debugging
 - Four hardware breakpoints
 - two observation points
- Program Count Sampling Register for Non-Intrusive Code (PCSR)
 - Single step and vector capture capability
- Bus interface:

- single 32-bit AMBA-3 AHB-Lite System interface for all system peripherals and memory

integration of

- support DAP (Debug Access Port) single 32-bit slave side

5.4.3 system timer (SYSTICK)

overview

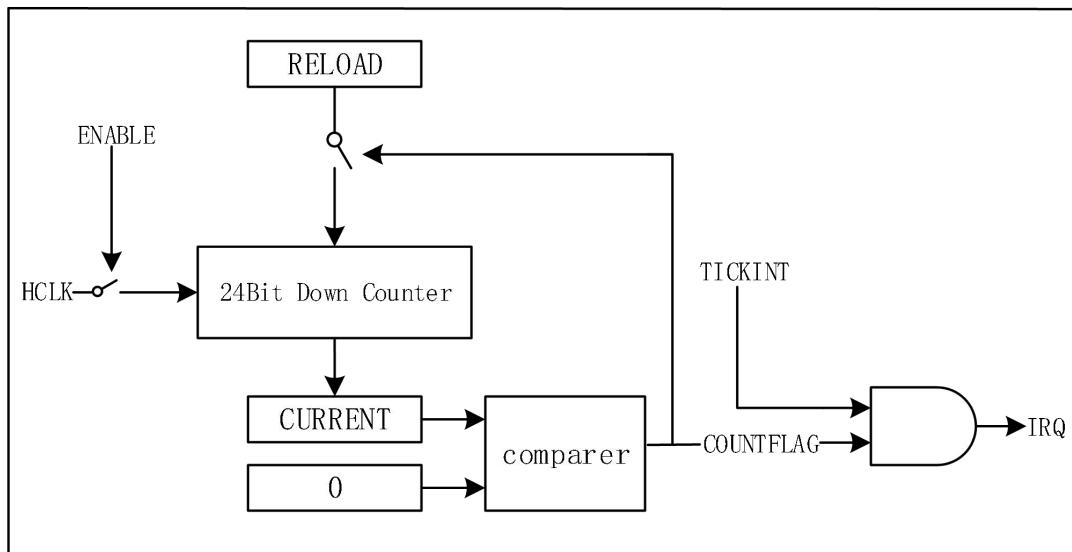
Cortex-M0The core provides a twenty fourbit system timer. Load the current value register after the timer is enabled (SYST_VAL) and decrements down to 0, and reloads the reload register on the next clock edge (SYST_LOAD) internal value. The counter decrements again to 0, the counter status register (SYST_CTRL) in the identification bitCOUNTFLAG place bit, read this bit to clear it.

After reset, SYST_VAL register with SYST_LOAD register values of the registers are unknown, so they need to be initialized before use. SYST_VAL Write any value, clear and reset the status register at the same time, and ensure that the loaded value is SYST_LOAD value in the register. when SYST_LOAD register value is 0, the timer remains at 0, and stop reloading.

characteristic

- twenty fourbit system timer
- decrease
- write clear

Block Diagram of Module Structure



picture5-2 SystickModule Structure Diagram

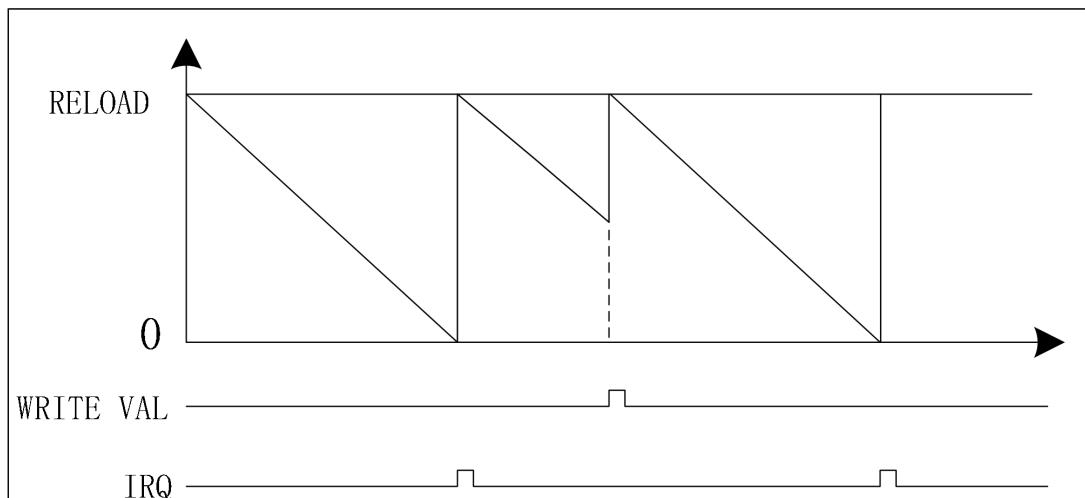
Functional description

After the timer is enabled, load the current value register (SYST_VAL) and decrements down to 0, and at the next clock to reload the reload register (SYST_LOAD) within the value. The counter decrements again to 0, the counter status register device (SYST_CTRL) flags in COUNTFLAGSet, read this bit to clear it.

After reset, SYST_VAL register with SYST_LOADThe values of the registers are unknown, so they need to be initialized before use. SYST_VALWrite any value, clear and reset the status register at the same time, and ensure that the loaded value is SYST_LOADvalue in the register when SYST_LOADThe register value is 0, the timer remains at 0, and stop reloading.

This counter can be used as a tick timer for real-time systems or as a simple counter.

SysTickThe counting sequence diagram is shown in the figure below:



picture5-3 systickCounting Timing Diagram

register map

name	offset	type	reset value	describe
SYSTICK	BASE: 0xE000E010			
SYST_CTRL	0x00	R/W	0x00	status register
SYST_LOAD	0x04	R/W	0x00	reload register
SYST_VAL	0x08	R/W	0x00	current value register

name	offset	type	reset value	describe
SYSTICK	BASE: 0xE000E010			
SYST_CTRL	0x00	R/W	0x00	status register
SYST_LOAD	0x04	R/W	0x00	reload register
SYST_VAL	0x08	R/W	0x00	current value register

Register description
SYST_CTRLStatus Register (0x00)

bit field	name	type	reset value	describe
31:17	RESERVED	R	0	reserved bit
16	COUNTFLAG	R	0	The counter counts to0, the location1
15:2	RESERVED	R	0	reserved bit
1	TINKINT	R/W	0	interrupt enable bit 0: disabled 1:Enable
0	ENABLE	R/W	0	Timer enable bit 0: disabled 1:Enable

SYST_LOADreload register (0x04)

bit field	name	type	reset value	describe
31:24	RESERVED	R	0	reserved bit
23:0	RELOAD	R/W	0	The counter counts to0Load the value of this register, restart start count

SYST_VALCurrent Value Register (0x08)

bit field	name	type	reset value	describe
31:24	RESERVED	R	0	reserved bit
23:0	VAL	R/W	0	A read operation returns the current count value, a write operation will clear the register memory, while clearing COUNTFLAG flag bit

5.power management (PMU)

5.5.1 overview

The supply voltage of this chip (AVDD) for 2.0-3.6V. Provided to the internal digital circuit through the built-in voltage regulator needed 1.2V power supply.

inside the chip A/D converter can be independently AVDD_ADC power supply, the supply voltage is 2.0-3.6V, and can be obtained by slice provides an independent VREF reference voltage. For chips in different packages, A/D converter's power supply may be connected with the chip main Power supply (AVDD) double bonding in the same PIN. For details, please refer to the pin description of each packaged chip model. A/D converter's VREF The reference voltage may also be different from the AVDD common bonding in the same PIN. For details, please refer to the pin description of each packaged chip model.

The chip also provides a variety of ways to reduce power consumption, including: reducing the system clock frequency, peripheral clock control, and various low power mode, etc.

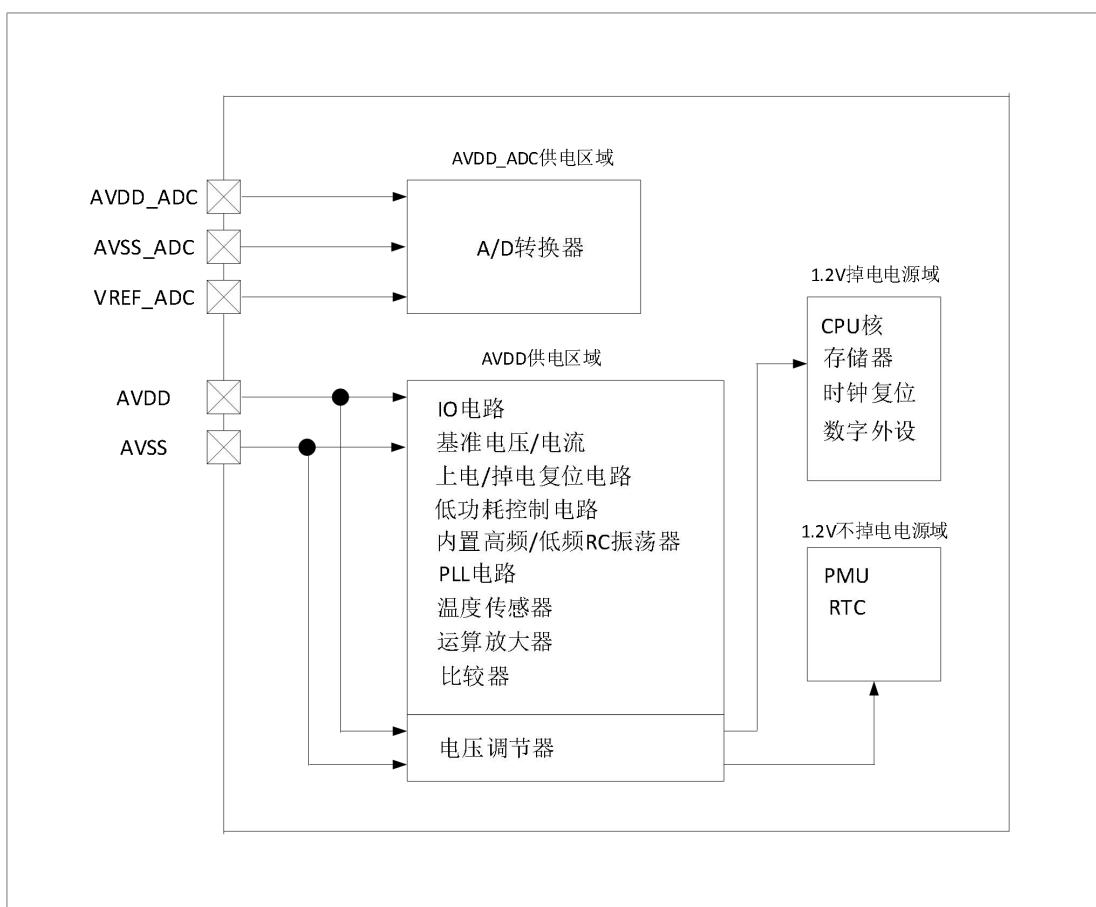
In addition to power-on reset and power-off reset, the chip also includes a watchdog (WDT) reset, pin reset, CPU soft reset.

5.5.2 characteristic

- Working voltage is 2.0V-3.6V
- ADC Separate Power Supply and Reference Supply
- Power-on reset (POR) and brown-out reset (PDR)
- External power-on reset pin
- watchdog reset

The working modes are: normal working mode (normal), WFI mode (WFI), sleep mode (sleep), Deep Sleep Mode (deep sleep), stop mode (stop)

5.5.3 Block Diagram of Module Structure



picture5-4Block Diagram of Power Module

As shown in the figure above, there are two groups of power/ground: one group is the main power/ground that supplies power to the chip analog circuit (AVDD/AVSS),

Another group is for A/D Converter power supply/ground (AVDD_ADC/AVSS_ADC).

Depending on the package, it is possible to AVDD and AVDD_ADC connected in the same PIN superior.

A/D The converter has an independent power supply (AVDD_ADC/AVSS_ADC), and an independent external voltage reference input

PAD(VREF_ADC).

Most of the analog circuits inside the chip are composed of AVDD power supply, including: multiple voltage regulators, IO circuit, reference voltage

Voltage/current, power-on/power-off reset circuit, low power consumption control circuit, built-in high frequency/low frequency RC oscillator circuit, PLLs, temperature

Sensors, operational amplifiers, comparators, etc.

The digital circuits in the chip are all working in the 1.2V voltage, divided into two different power domains: 1.2V power down power domain and

1.2V The power domain is not powered off, and the working voltage can be adjusted by a voltage regulator to meet different low-power application scenarios.

1.2VThe power-down power domain mainly includesCPUCore, memory, clock reset control circuit and many digital peripherals, etc., and

1.2VThe non-power-off power domain mainly includesPMUmodule,RTCmodules etc.

5.5.4Functional description

power supply

HM1030The operating voltage of the chip is2.0V-3.6V. The built-in voltage regulator provides1.2VThe power supply to the internal digital power road. In order to improve the conversion accuracy,ADCUse separate power supplies and reference supplies.

independentA/DConverter Power Supply and Reference Voltage

To improveADCconversion precision,ADCAAn independent power supply and off-chip reference voltage are used. Separately powered The purpose is to filter and block fromPCBInterference and noise on the board. Provide an independent off-chip reference voltage for convenience during use Provides flexible reference voltages to meet more application scenarios.

ADCThe power supply pin isAVDD_ADC.

ADCThe ground pin isAVSS_ADC.

ADCThe off-chip voltage reference pin isVREF_ADC.

ifVREF_ADCandAVDD_ADCconnected in the samePINon the package, the level can be fixed by internally connecting the The way to deduce the voltage value of the current reference voltage source.

Voltage Regulator

On-chip voltage regulators are used to provide power to all digital circuits. According to the working mode of the chip it uses five different work the same way:

1) normal operating mode: the voltage regulator operates in normal power mode to provide1.2Vpower to the digital circuits, at this time all the

Digital circuits can work normally;

2)WFI mode: onlyCPUcore stopped, other clocks, power supplies, and all peripherals includingCPUcore peripherals,

likeNVIC, tick clock (SysTick) and so on are still running according to the original state.

3)SLEEPMode of Operation: The voltage regulator provides a low-power mode1.2Vpower supply to the digital circuits to ensure that the

The digital circuit in the power domain works normally, and the digital circuit in the power domain saves the current state when the power is off.

4)DEEPSLEEPMode of Operation: The voltage regulator provides a low-power mode1.2Vpower supply (this voltage can be

TRIM_LPLDORegister to further adjust the step-down) to the digital circuit to ensure that the digital

The circuit works normally.

5)STOPMode: The voltage regulator stops supplying power. All digital circuit content is lost.

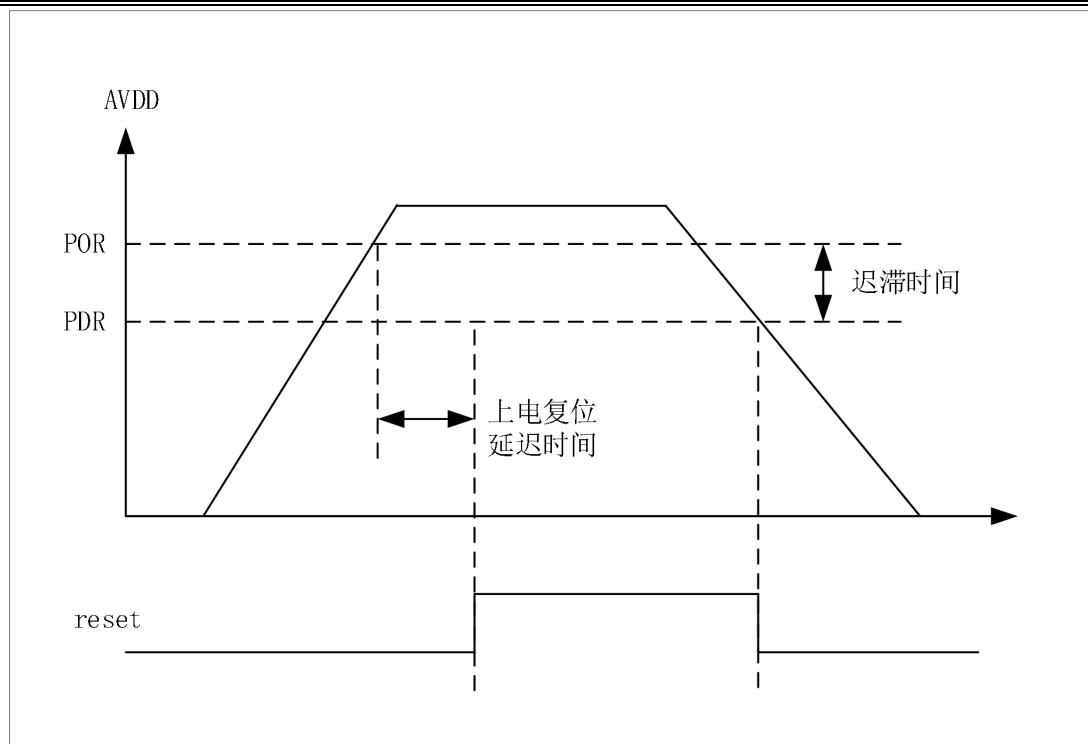
reset

The chip has a total of6Reset sources: power-on reset, brown-out reset, pin reset, watchdog reset, andCPUsoft complex bit, of which there are two types of watchdog reset: independent watchdog (IWDT) reset and the window watchdog (WWDT) reset.

Power-on reset (POR) and brown-out reset (PDR)

The chip consists of a complete set of power-on reset (POR) and brown-out reset (PDR) circuit. When the chip supply voltage is on When the voltage rises to a specific threshold voltage, the system can work normally; when it is lower than the specific threshold voltage, the chip remains in the reset state, without the need for an external reset circuit.

The waveform diagram of power-on reset and power-off reset is as follows:



picture5-5Waveform diagram of power-on reset and power-off reset

in,resetIt is the general reset signal of the digital circuit and resets all the digital circuits.

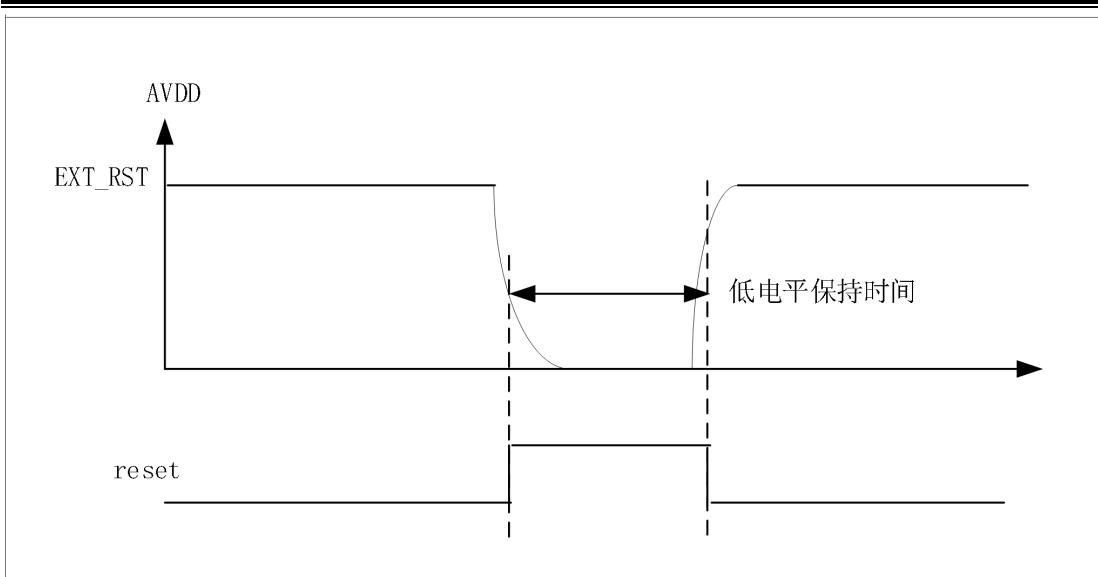
Threshold voltage threshold for power-on reset and brown-out reset, hysteresis for power-on reset and brown-out reset, power-on reset delay

For details such as time, please refer to the relevant chapters of the electrical characteristics in the data sheet.

pin reset

The chip supports a specific external pin reset, and a low level generates a reset.

The waveform diagram of external pin reset is as follows:



picture5-6External pin reset waveform diagram

in,resetIt is the general reset signal of the digital circuit and resets all the digital circuits.

For details about the pin reset low-level hold time, please refer to the relevant chapters of the electrical characteristics of the data sheet.

watchdog reset

This chip has two watchdogs: one is an independent watchdog (IWDT), the other is the window watchdog (WWDT).

Both can generate a watchdog reset and can reset all digital circuits on the chip, includingCPUcore, bus, memory

Memory controllers, peripheral circuits, etc.

CPUsoft reset

passARM CORTEX M0Own soft reset instruction, to0xe000ed0caddress write0x05fa0004, butCPU

The core will issue a software reset operation.

This reset resets most digital circuits. instead of the power-down domainPMU/RTCmodule; power-down domainSYSCONmold within the blockVREF_VOLT_DELTA,RC_FREQ_DELTA,DEVICE_ID0,DEVICE_ID1,DEVICE_ID2,DEVICE_ID3 registers, these circuits will not be reset.

Reset Source Reset Range

surface1Reset Source Reset Range Table

reset source	PMU/RTC	Most digital circuits	Special not to beCPUsoft reset shadow loud circuit
Power-on/power-down reset	reset	reset	reset
pin reset	reset	reset	reset
watchdog reset	reset	reset	reset
CPUsoft reset	reset	reset	not reset

Among them, special is notCPUcircuits affected by a soft reset refer toCPUthe circuit described in the soft reset chapter. the vast majority digital circuit fingerCPUCore, bus, memory controller, clock reset control circuit, peripheral equipment, etc.

low power mode

After the system power supply of the chip is powered on and the reset is released, the chip is in a normal working state. At this time, the system clock is silent recognized by the internal high frequencyRCOscillator clock with an output frequency of24MHz, all circuits operate normally,CPUstart

The instruction is fetched, and the program runs normally.

whenCPUWhen there is no need for continuous operation, various low-power modes can be used to save chip power consumption, such as waiting for a certain External events or long-term timing waits. Users can choose according to the minimum power consumption, start-up time requirements and wake-up sources, etc.,

Select an appropriate low power mode.

There are four main low-power working modes of the chip:WFImodel,SLEEPmodel,DEEPSLEEPpattern coreSTOP model.

- WFImode: onlyCPUcore stopped, other clocks, power supplies, and all peripherals includingCPUcore peripherals, likeNVIC, tick clock (SYSTICK) and so on are still running according to the original state.
- SLEEPMode: All digital circuits are not powered down. Voltage Regulator(LDOs) will also operate in low power mode, And the output voltage can be adjusted, which can further reduce the overall power consumption. At this time, the internal high frequencyRCoscillator clock The source is turned off, the system clock is turned off, and only the internal low frequency is keptRCThe oscillator clock is running forRTCan wake up circuit.

- DEEPSLEEPmodel:1.2VPower-down domain digital circuits are powered down, includingCPUcore, memory, and most peripherals

and other digital circuits.1.2VThe digital circuit in the non-power-down domain is in a powered state, and the power supply voltage regulator at this time

(LDOs) will also work in low power consumption mode, and the output voltage can be adjusted to further reduce the overall

power consumption. At this time, the internal high frequencyRCOscillator clock source is off, only internal low frequencyRCThe oscillator clock is running,

forRTCAnd wake-up circuit.

- STOPmodel:1.2VPower is off and all clock sources are off.

Additionally, under normal operating conditions, power consumption can be reduced by:

- Lower the system clock.
- Shut down unused peripheral clocks on the bus.

The following table gives the detailed information in low power mode:

surface2Low Power Mode Details Table

model	Enter	wake	for the digital domain Influence	clock source Impact	to the voltage regulator Impact
WFImodel	WFI	CPUto interrupt	CPUclock off, system clock and other peripherals Clock has no effect	none	none
SLEEP	WillIPOW_MD registerSLEEP Location1	allIOOr RTC	when shutting down the system Clock and Peripheral Time bell	RCHFclosure	Enable low power mode format, and can be adjusted output voltage (TRIM_LPLDO)
DEEPSLEEP	WillIPOW_MD register DEEPSLEEPbit place1	specificIOOr RTC	1.2Vpower down domain no power. close the system system clock and external Set the clock	RCHFclosure	Enable low power mode format, and can be adjusted output voltage (TRIM_LPLDO)
STOP	WillIPOW_MD registerSTOP Location1	specificIO	all numeric fields no power	RCHFand RCLFclosure	closure

The details of each low power consumption mode are as follows:

lower system clock

In normal working mode, through the CLK_SEL in the register DIV_CLK_SEL be programmed to reduce the system frequency of the clock. See details 11.6.1 Section: Clock Select Register (CLK_SEL).

Peripheral Clock Control

The chip provides independent peripheral clock control for each peripheral. In normal working mode, it can be turned off at any time.

Turn off peripheral clocks to reduce power consumption.

by right DEV_CLK_GATE Each peripheral clock control bit in the register is programmed to switch the clock speed of each peripheral module clock.

WFI model

by executing ARM CPUNuclear WFI command, which can be used directly CPU The core enters the standby state. In this mode, only CPU is the stopped state, all other IOPins and peripheral blocks maintain their status in normal operating mode. appoint

It means that a peripheral interrupt that is responded to by the nested vectored interrupt controller can be CPU Core wakes up from standby and continues as normal Execute instructions.

This mode takes the shortest time to wake up, and since there is no time lost on interrupt entry or exit, only a few system clock cycle to complete.

SLEEP model

Through the software will LPOW_MDI in the register SLEEP Location 1, allowing the chip to actively enter SLEEP model. for

Make the overall power consumption of the chip in SLEEP The mode should be as low as possible, and the modules that do not need to work can be turned off before entering (such as ADC modules, operational amplifier modules, and digital function peripherals that do not need to work, etc.), save power consumption, and can also pass TRIM_POW3 register LPLDO voltage adjustment bit (TRIM_LPLDO) will output the voltage regulator in low power mode The output voltage is lowered to further reduce power consumption.

existSLEEPIn mode, the analog power consumption control circuit will turn off all the analog modules with relatively large power consumption (such as RCHFetc.), keeping only the basicBG, voltage regulator, and built-in low-frequencyRCoscillator (RCLF) clock, and enables power-up

The voltage regulator works in low power consumption mode, the high-frequency clock in the chip and the system clock are also due to the built-in high-frequencyRCoscillator (RCHF) is turned off to stop. At this time, the entire digital circuit has onlyPMUmodule,RTCThe work in the module is inRCLFclock

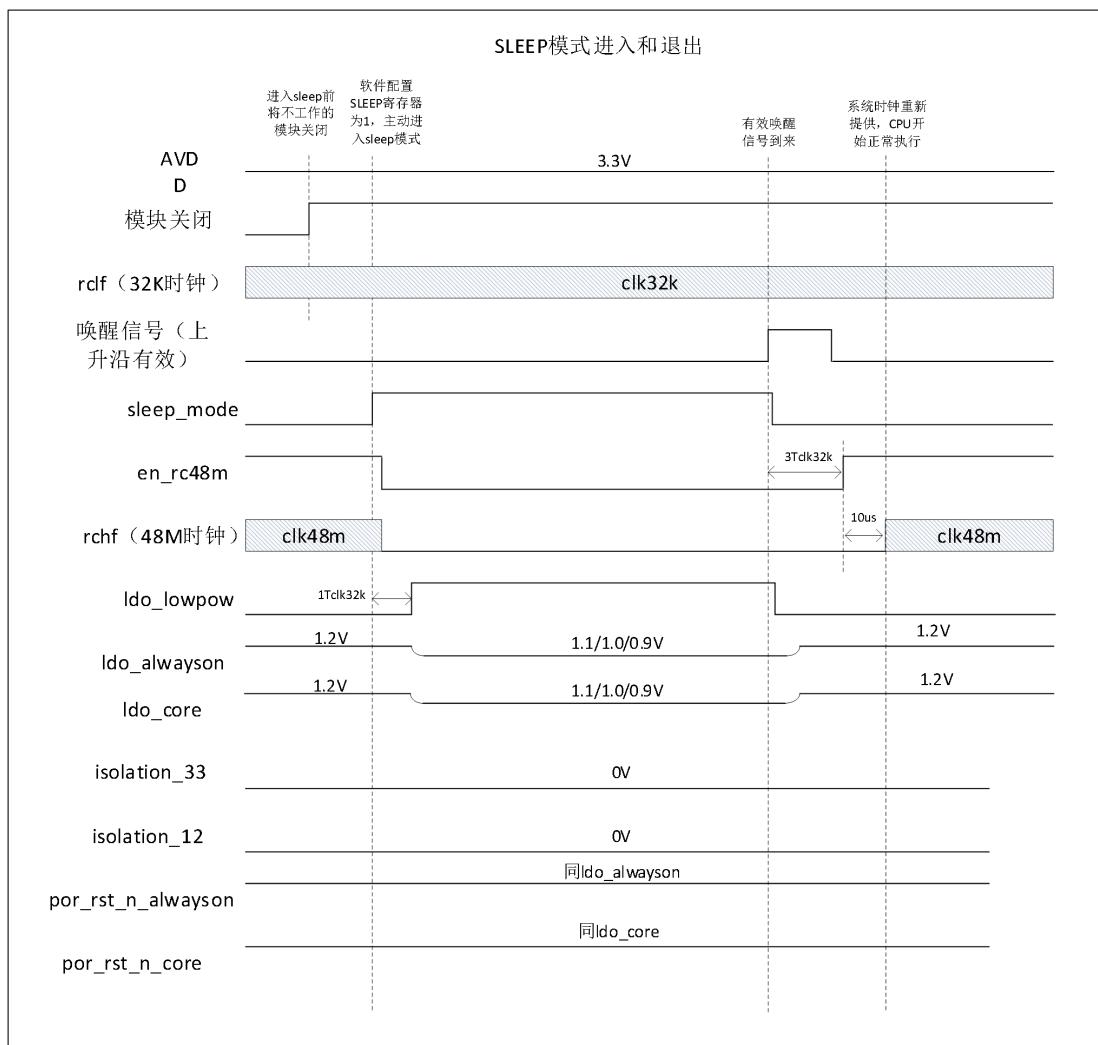
Except the circuit below is working normally, all other digital circuits stop because there is no clock.

SLEEPmode can be exited by an externalIOsignal wake-up or viaRTCTiming signal wakes up.PMUdetected corresponding ofIOorRTCAfter a valid wake-up signal, theSLEEP_MODESignal is cleared to0, means exitSLEEPmode, analog consumption control circuit detectsSLEEP_MODEsignal becomes0After that, the wake-up process will be started, so that the chip will start fromSLEEPmode recovery to normal operating mode.

of which allIOBoth supportSLEEPmode wake-up function.

The time required to wake up from this mode is about100usabout.

SLEEPThe schematic diagram of mode entry and exit is as follows:



picture5-7 SLEEPMode entry and exit diagram

DEEPSLEEPmode

Through the software willLPOW_MDi in the registerDEEPSLEEPlocation1, allowing the chip to actively enterDEEPSLEEPmode

Mode. In order to make the overall power consumption of the chip inDEEPSLEEPThe mode should be as low as possible, and the modules that do not need to work can be turned off before entering

closed (e.g. ADCmodules, operational amplifier modules, and digital function peripherals that do not need to work, etc.), saving power consumption, and

Also available throughTRIM_LPLDoregister to adjust the output voltage of the voltage regulator down in low-power mode, thereby further

further reduce power consumption.

existDEEPSLEEPIn mode, the analog power consumption control circuit will shut down all the analog modules with relatively large power consumption (for example likeRCHFetc.), keeping only the basicBG, voltage regulator, and built-in low-frequencyRCoscillator (RCLF) clock, and



Make the voltage regulator work in low power consumption mode, the high-frequency clock in the chip and the system clock are also due to the built-in high-frequency RC vibration

Oscillator (RCHF) is turned off to stop.

1.2VThe power supply of the digital circuit in the power-down domain will also be disconnected, so that all the circuits in the power-down domain will be powered off to achieve further savings

power consumption purposes.

1.2VThe non-power-down domain is still powered by the voltage regulator, the PMU, RTC module works in RCLF hour

Except the circuit under the clock works normally, other digital circuits stop because there is no clock.

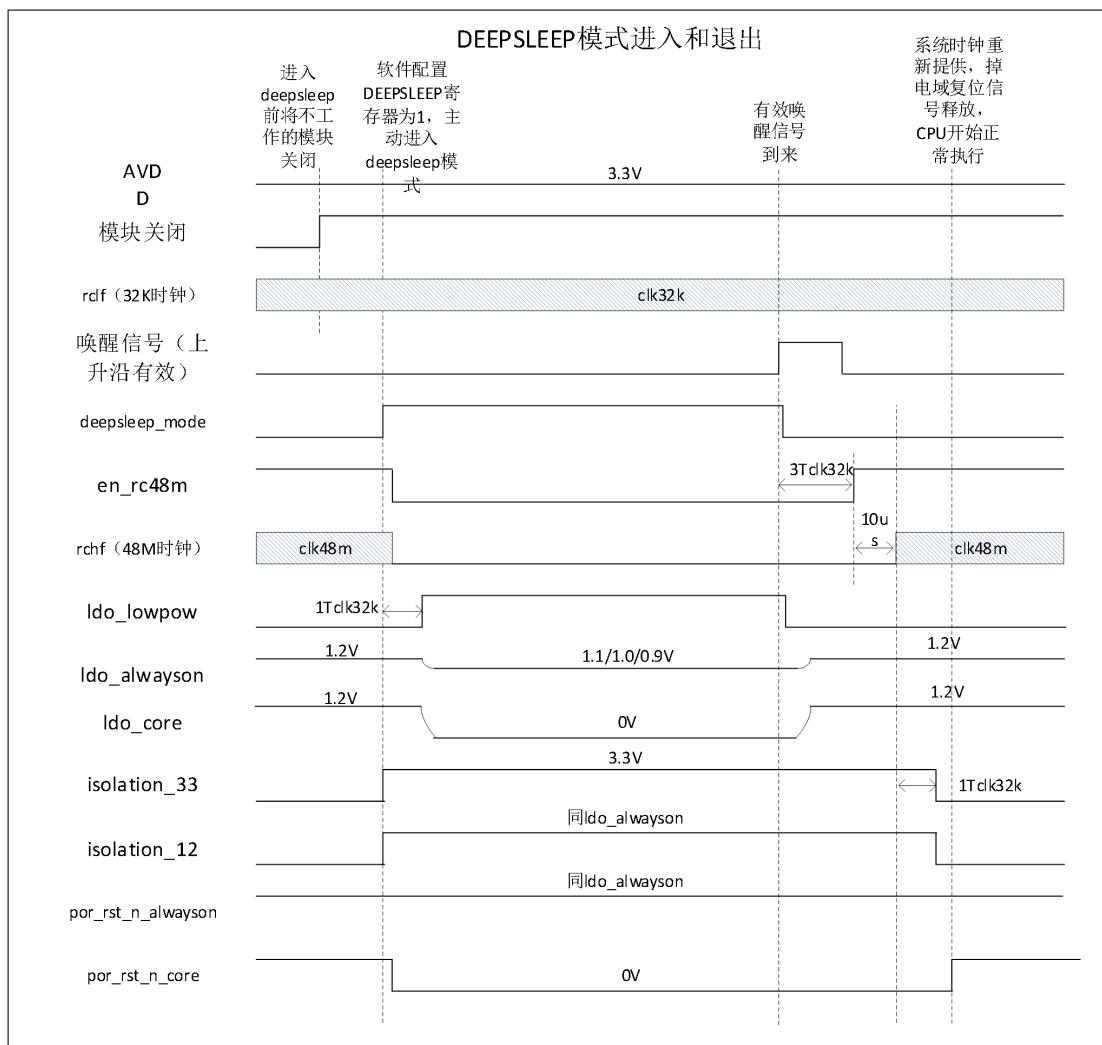
This chip, RAM in the power-down domain, data is not saved due to power-down.

DEEPSLEEP mode can be exited by an external IO signal wake-up or via RTC timing signal wakes up. PMU detected corresponding IO or RTC. After a valid wake-up signal, the DEEPSLEEP_MODE signal is cleared to 0, means exit DEEPSLEEP mode, the analog power control circuitry detects the DEEPSLEEP_MODE signal becomes 0. After that, the wake-up process will be started, so that the chip piece from DEEPSLEEP Mode returns to normal working mode.

This chip only has specific IO support DEEPSLEEP mode wake-up function.

The time required to wake up from this mode is about 140us about.

DEEPSLEEP The schematic diagram of mode entry and exit is as follows:



picture5-8 DEEPSLEEPMode entry and exit diagram

STOPmodel

Through the software willLPOW_Mdin the registerSTOPLocation1, allowing the chip to actively enterSTOPmodel.

existSTOPmode, the voltage regulator is turned off,1.2VAll digital circuits are powered down, allAVDDUnder the power domain

All analog modules are turned off, and only a small part of the analog wake-up circuit is kept working.

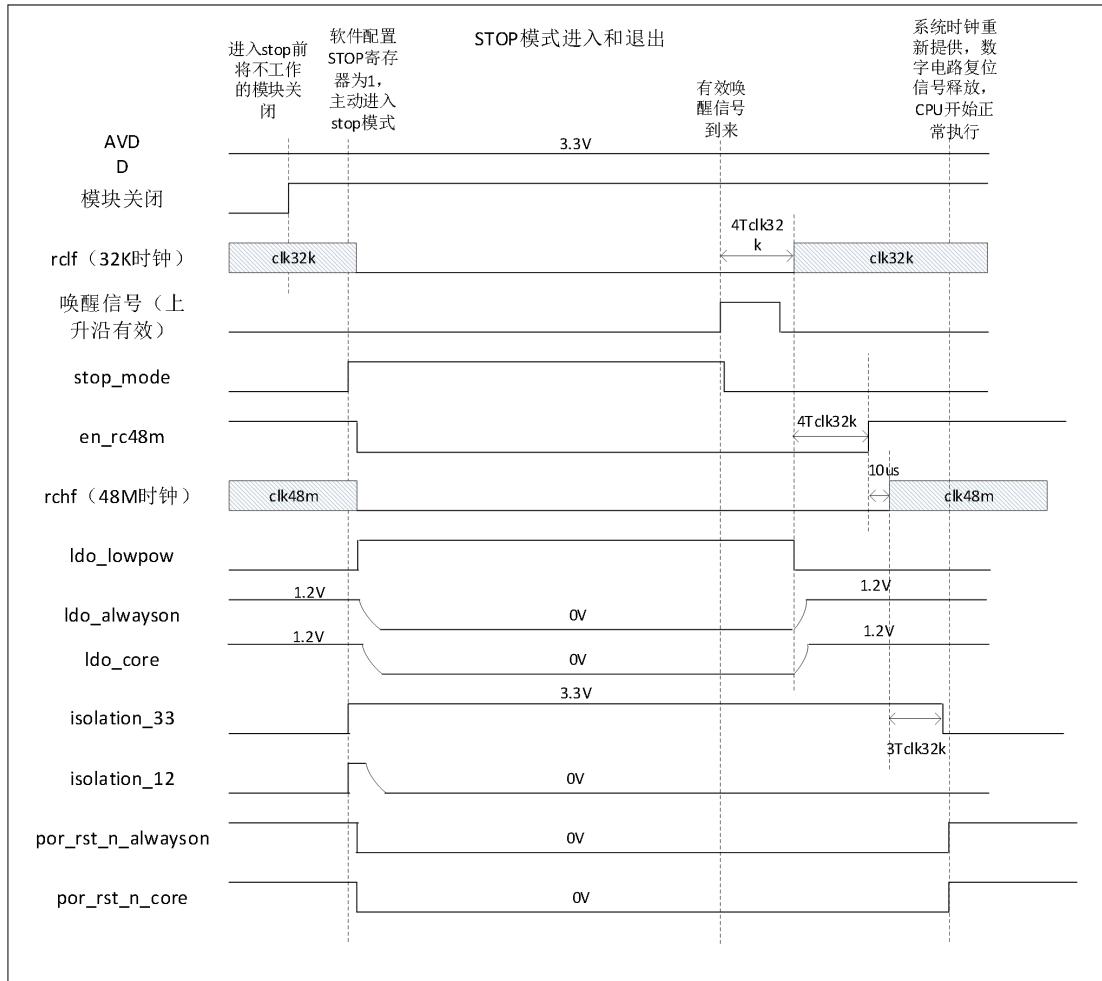
STOPmode can be exited by an externalIosignal to wake up. When the analog wake-up circuit detects the corresponding wake-up signal, it will

Start the wake-up process, so that the chip fromSTOPThe mode returns to normal working mode.

This chip only has specificIosupportSTOPmode wake-up function.

The time required to wake up from this mode is about 350us about.

STOPThe schematic diagram of mode entry and exit is as follows:



picture5-9 STOPMode entry and exit diagram

register map

Control which low-power mode the entire chip enters:sleep,deep sleep and stop.

This module is a power consumption control module, including generating various low power consumption control circuits, wake-up circuits, and power-down

Data held (including simulatedTRIMvalue), etc.

name	offset	type	reset value	describe
PMU	BASE:0x40000800			
LPOW_MD	0x00	R/W	0x00	Low Power Mode Select Register
LPMD_WKEN	0x04	R/W	0x00	Low Power Wakeup Source Enable Register
LPMD_WKST	0x08	R/W	0x00	Low Power Wakeup Source Status Register
CHIP_RST_ST	0x0C	R/W	0x01	Chip Reset Status Register
SRC_CFG	0x10	R/W	0x03	Clock Source Configuration Register
TRIM_POW0	0x20	R/W	0x00	POW0Related Analog ModulesTRIMregister
TRIM_POW1	0x24	R/W	0x00	POW1Related Analog ModulesTRIMregister
TRIM_POW2	0x28	R/W	0x00	POW2Related Analog ModulesTRIMregister
TRIM_POW3	0x2C	R/W	0x00	POW3Related Analog ModulesTRIMregister
TRIM_RCHF	0x30	R/W	0x808	RCHFclock moduleTRIMregister
TRIM_RCLF	0x34	R/W	0x810	RCLFclock moduleTRIMregister
TRIM_OPA	0x38	R/W	0x00	OPAmoduleTRIMregister
TRIM_PLL	0x3C	R/W	0x00	PLLsmoduleTRIMregister
TRIM_LOCK	0x80	R/W	0x00	TRIMlock register
DATA_BAKE0	0x100	R/W	0x00	Non-power-off domain data backup register0
DATA_BAKE1	0x104	R/W	0x00	Non-power-off domain data backup register1
DATA_BAKE2	0x108	R/W	0x00	Non-power-off domain data backup register2
DATA_BAKE3	0x10C	R/W	0x00	Non-power-off domain data backup register3

Register description

LPOW_MDregister(0x00)

bit field	name	type	reset value	describe
-----------	-------------	-------------	--------------------	-----------------

31:4	RESERVED	R	0	reserved bit
3	STOP	R/W	0	write to this register1, the chip entersSTOPpattern software write1, , the hardware automatically clears the
2	DEEPSLEEP	R/W	0	write to this register1, the chip entersDEEPSLEEPpattern software write1, the hardware automatically clears the
1	SLEEP	R/W	0	write to this register1, the chip entersSLEEPpattern software write1, the hardware automatically clears the
0	RESERVED	R	0	reserved bit

Note: The chip enters low-power mode from normal working mode, and can only enter one low-power mode at a time.

The power consumption mode exits and returns to the normal working mode, and is reconfigured by software to enter another low power consumption mode.

LPMD_WKENregister(0x04)

bit field	name	type	reset value	describe
31:3	RESERVED	R	0	reserved bit
2	IO_WKEN	R/W	0	In low power mode, theIOWake up enable 0: disabled 1:Enable Note 1which specificIOWith wake-up function available viaPORTA_WKE, PORTB_WKE,PORTC_WKERegisters for configuration
1	RTC_TIM_WK EN	R/W	0	In low power mode, theRTCTime signal wake-up enable 1:RTCTime signal with low power wake-up
0	RTC_ALA_WK EN	R/W	0	In low power mode, theRTCAlarm clock signal wake-up enable 1:RTCAlarm signal with low power wake-up function

LPMD_WKSTregister(0x08)

bit field	name	type	reset value	describe
31:3	RESERVED	R	0	reserved bit

2	IO_WKST	R/W	0	In low power mode, the IO wake up flag 1: occur IO event wakeup 0: Yet to happen IO event wakeup hardware settings 1, the software writes 1 to clear
1	RTC_TIM_WKST	R/W	0	In low power mode, the RTC time wake up flag 1: occur RTC time event wakes up hardware reset 1, the software writes 1 to clear
0	RTC_ALA_WKST	R/W	0	In low power mode, the RTC alarm clock wake up sign 1: occur RTC alarm clock event wakes up the hardware 1, the software writes 1 to clear

CHIP_RST_ST Register(0x0C)

bit field	name	type	reset value	describe
31:3	RESERVED	R	0	reserved bit
2	WWDT_RST_ST	R/W	0	WWDTReset Status Flags Register 0: Indicates that there is no WWDTreset 1: Indicates that there is WWDTreset Write 1 clear
1	IWDT_RST_ST	R/W	0	IWDTReset Status Flags Register 0: Indicates that there is no IWDTreset 1: Indicates that there is IWDTreset Write 1 clear
0	POR_RST_ST	R/W	1	Power-on reset status flag register 0: Indicates that there is no power-on reset 1: Indicates that a power-on reset has occurred Write 1 clear

SRC_CFGregister(0x10)

bit field	name	type	Reset value	description

31:5	RESERVED	R	0	reserved bit
4	RTC_CLK_SEL	R/W	0	RTClock selection 0:RCLF 1:XTAL
3	XTAL_EN	R/W	0	XTALenable control bit 0:closureXTAL 1: openXTAL
2	XTAH_EN	R/W	0	XTAHenable control bit 0:closureXTAH 1: openXTAH
1	RCHF_FSEL	R/W	1	RCHFFrequency Select Control Bits 0:48MHz 1:24MHz
0	RCHF_EN	R/W	1	RCHFenable control bit 0:closureRCHF 1: openRCHF

TRIM_POW0register(0x20)

bit field	name	type	reset value	describe
31:11	RESERVED	R	0	reserved bit
10:8	TRIM_TEMP_C_O_HPBG	R/W	0	HPBGtemperaturetrimbit
7:4	TRIM_I_HP	R/W	0	HPBGelectric currenttrimbit
3:0	TRIM_V_HP	R/W	0	HPBGVoltagetrimbit

TRIM_POW1register(0x24)

bit field	name	type	reset value	describe
31:8	RESERVED	R	0	reserved bit

7:4	TRIM_V_LP	R/W	0	LPBGVoltagetrimbit
3:0	TRIM_TEMPC_O_LPBG	R/W	0	LPBGTemperaturetrimbit

TRIM_POW2register(0x28)

bit field	name	type	reset value	describe
31:0	RESERVED	R	0	reserved bit

TRIM_POW3register(0x2C)

bit field	name	type	reset value	describe
31:4	RESERVED	R	0	reserved bit
3	TRIM_HPLDO_H	R/W	0	HPLDOvoltage adjusted to1.264v 0:constant 1: adjust up to1.264v
2:1	TRIM_LPLDO	R/W	0	LPLDOvoltage outputtrimbit 00:1.1V 01:1.0V 10:0.9V 11:0.8V
0	TRIM_PD_UVLO	R/W	0	UVLO33 trimbit 0:existSLEEPdown, the supply voltage drops to1.8V, the chip is reset; in this case the analog circuit consumes additional0.6uApower consumption; 1:existSLEEPdown, the supply voltage drops to1.3V(+500mVdeviation), the chip resets; the analog circuitry saves0.6uApower consumption;

TRIM_RCHFregister(0x30)

bit field	name	type	Reset value	description

31:12	RESERVED	R	0	reserved bit
11:8	TRIM_N	R/W	0x8	RCHF N trimbit
7:4	RESERVED	R	0	reserved bit
3:0	TRIM_P	R/W	0x8	RCHF P trimbit

TRIM_RCLFregister(0x34)

bit field	name	type	reset value	describe
31:12	RESERVED	R	0	reserved bit
11:8	TRIM_CS	R/W	0x8	RCLF CS trimbit (coarse adjustment bit)
7:5	RESERVED	R	0	reserved bit
4:0	TRIM_FINE	R/W	0x10	RCLF FINE trimbits (fine bits)

TRIM_OPAreregister(0x38)

bit field	name	type	reset value	describe
31:20	RESERVED	R	0	reserved bit
19:15	OPA1_TRIMP	R/W	0	OPA1ofPendTRIMbit
14:10	OPA1_TRIMN	R/W	0	OPA1ofNendTRIMbit
9:5	OPA0_TRIMP	R/W	0	OPA0ofPendTRIMbit
4:0	OPA0_TRIMN	R/W	0	OPA0ofNendTRIMbit

TRIM_PLLregister(0x3C)

bit field	name	type	reset value	describe
31:4	RESERVED	R	0	reserved bit
3:0	PLL_R_TRSIM	R/W	0	PLLs of R value TRIM bit

TRIM_LOCKregister(0x80)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:8	TRIM_UNLOCK	W	0	write this register to 0xAAfter, the corresponding TRIM configuration registry device can be rewritten.
7:0	TRIM_LOCK	W	0	write this register to 0x55back, corresponding TRIM configuration register cannot be overwritten and is used to protect the TRIM incorrect rewriting of registers.

DATA_BAK0register(0x100)

bit field	name	type	reset value	describe
31:0	DATA_BAK0	R/W	0	data backup0

DATA_BAK1register(0x104)

bit field	name	type	reset value	describe
31:0	DATA_BAK1	R/W	0	data backup1



DATA_BAK2register(0x108)

bit field	name	type	reset value	describe
31:0	DATA_BAK2	R/W	0	data backup2

DATA_BAK3register(0x10C)

bit field	name	type	reset value	describe
31:0	DATA_BAK3	R/W	0	data backup3

5.6 System Control (SYSCON)

5.6.1 overview

The chip manages and controls the clock network of the overall chip system through the system control unit, which is the clock network for each module in the chip and provides clocks, including system clocks, peripheral clocks for each module, and clocks for special functions. This unit also enables independent control of the clock on or off for each module, the selection and frequency division of the system clock source, and the frequency division of special function clocks to achieve reasonable power consumption.

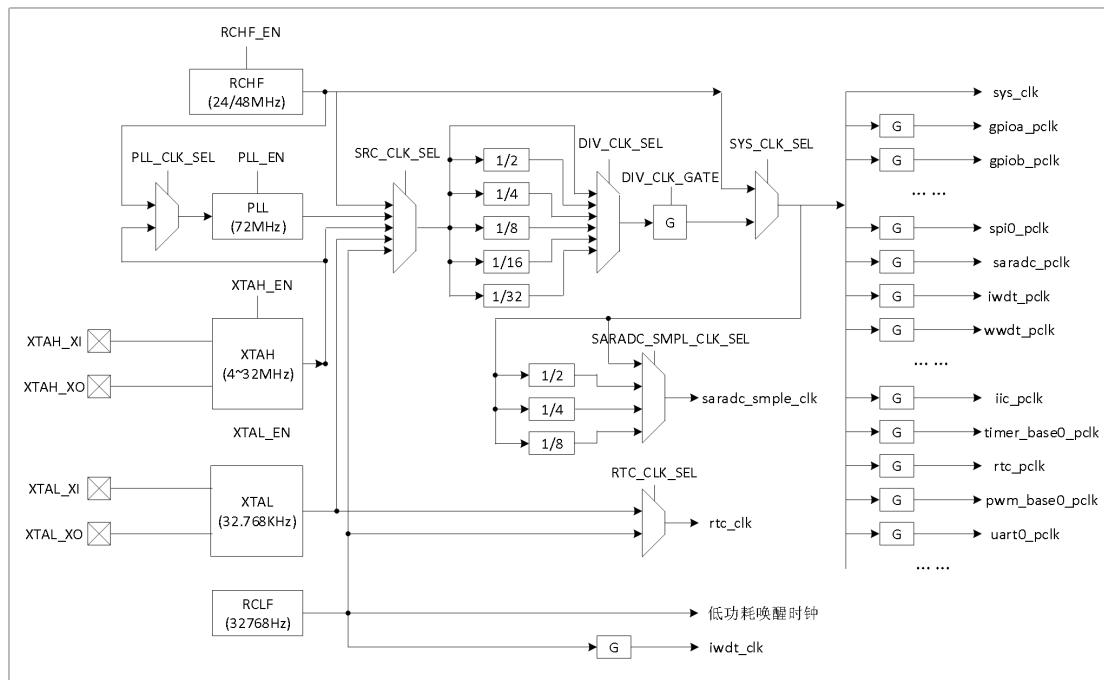
5.6.2 characteristic

- System clock source selection, frequency division and control
- Individual switches for each peripheral
- have128bitChip unique identification code

5.6.3 Block Diagram of Module Structure

in total 5 clock sources: RCHF(48MHz internal RC oscillator), RCLF(32768Hz internal RC oscillator), PLLs(Highest 72MHz), XTAH(4-32MHz crystal oscillator), XTAL(32768Hz crystal oscillator).

The chip clock network structure is as follows



picture5-10Chip clock structure diagram

sys_clk(system clock) can select one of five clock sources, and can also be 1/2/4/8/16/32 point

frequency.

CPU Cortex-M0 of FCLK, HCLK as well as SCLK all by sys_clk supply clock, so M0 Internal SYSTICK

The clock source for sys_clk.

All peripheral bus clocks have their own clock control registers to control their on or off, so that when not in use

The clock of the module can be turned off to save power consumption; and the bus clock frequency is consistent with the system clock frequency.

SARADC The sampling clock passes through the CLK_SEL register SARADC_SMPL_CLK_SEL bits can be selected as sys_clk of 1/2/4/8 Frequency division, according to the different usage scenarios of the sampling signal, different sampling clock frequencies can be selected.

Independent Watchdog (IWDT) has a low-frequency clock source for counting, the low-frequency clock is fixed RCLF clock supply, and is subject to DEV_CLK_GATE register. IWDT_CLK_GATE bit controls whether the clock is turned on or off.

window watchdog (WWDT) uses the system clock as a count and is subject to DEV_CLK_GATE register WWDT_CLK_GATE bit controls whether the clock is turned on or off.

RTC The count clock can be selected as RCLF or XTAL Two clock sources, via DEV_CLK_GATE register RTC_CLK_SEL bit to configure the selection.

SLEEPmode and DEEPSLEEPMode fixed use RCLF as a wake-up clock, and RCLF The clock cannot be turned off.

5.6.4 Functional description

External High Frequency Crystal Clock (XTAH) can be generated from two clock sources: an external crystal/ceramic resonator or the user's external clock.

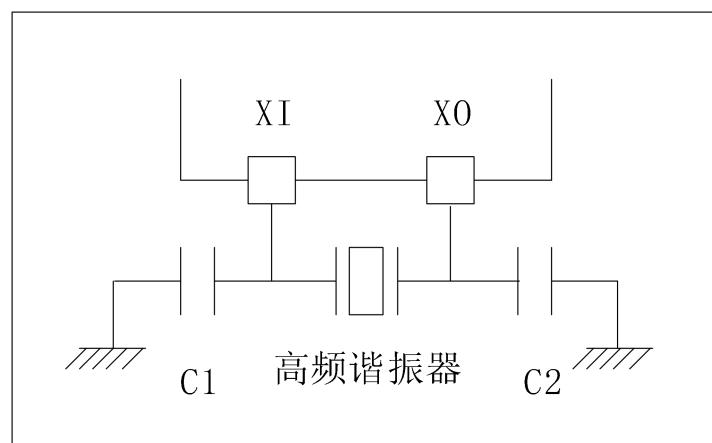
In order to reduce the distortion of the clock output and shorten the start-up settling time, the crystal/ceramic resonator and the load capacitor must be as small as possible.

possible close to the oscillator pins. And the load capacitance must be matched and adjusted according to the selected oscillator.

1, external crystal/ceramic resonator

XTAH supports plug-in 4-32MHz A crystal/ceramic resonator that provides a very accurate clock to the chip. related hardware configuration

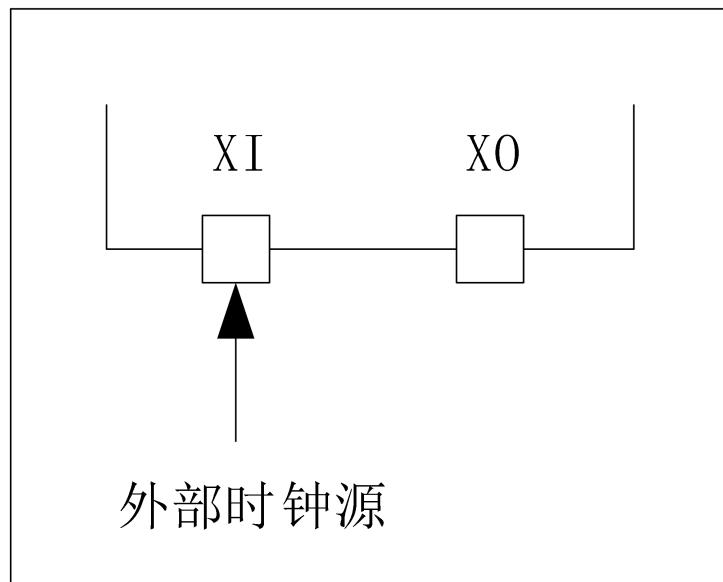
The settings can be referred to as shown in the figure below, and further information can be found in the electrical characteristics section of the data sheet.



picture5-11 External high-frequency crystal oscillator related hardware configuration diagram

2, external clock

XTAH It also supports the connection method of direct input of external clock, and its frequency can reach up to 48MHz. The external clock signal must be connected to XI pin, while ensuring that the X0 pin floating. The related hardware configuration can refer to the figure below.



picture5-12 External clock related hardware configuration diagram

XTAHThe clock can be configured byPMUmiddleSRC_CFGregisterXTAH_ENbit to enable or disable. start up After, wait at least10msto use normally.

RCHFclock

RCHFThe clock signal is provided by the chip internal high frequencyRCoscillator (48MHz)produce. can be configured byPMUmiddle SRC_CFGregisterRCHF_ENbit to enable or disable, and can be configured byPMUmiddleSRC_CFGregister ofRCHF_FSELbit selection48MHzThe output is still24MHzoutput.

RCHF RCThe oscillator can provide the clock required by the system without any external components. its start time CompareXTAHA crystal oscillator is shorter, but it still has poor clock frequency accuracy even after calibration. Further information can be found in Refer to the electrical characteristics section of the data sheet.

calibration

The manufacturing process determines the different chipRCThe oscillator frequency will vary, which is why each chip'sRCHF The clock frequency has been calibrated to $\pm 1\%$ (25°C) or less. After the system is reset, the software can

Calibration values written to specific memory addresses during production testing are loaded into theTRIM_RCHFthe corresponding bit in the register, and can also No.74total pages432Page

Loads the actual frequency value written to a specific address in memory during production testing into `RC_FREQ_DELTA` register

`RCHF_DELTA` and `RCHF_SIG` in the bit, the deviation between the current frequency and the ideal frequency can be known during use, so as to obtain more

Precise clock frequency.

If the user uses different voltage or ambient temperature in the use scene, it will affect RC oscillator precision. user also available through `TRIM_RCHF` in the register `TRIM_P` bit to adjust `RCHF` frequency.

PLLs clock

PLLs Two clock sources can be used (XTA or RCHF) One of them is used as a reference clock input to generate a multiplied clock output.

See Clock Network Block Diagram and `CLK_SEL` in the register `PLL_CLK_SEL` bit description.

PLLs Before enabling, it must be completed PLLs Related configurations (clock source selection, prescaler coefficient and multiplication coefficient etc.), and should not be enabled until their input clocks are stable. and once enabled PLLs, these parameters will no longer be Was changed.

when needed to change PLLs When configuring the , you must first activate the PLLs After closing, make changes to the corresponding configuration, and then re-enable PLLs.

PLLs After enabling, the `PLL_ST` register `PLL_LOCK` Status bit judgment PLLs is locked, PLLs hour The clock can only be used when locked. PLLs It takes about 30us to lock. Further information can be found in the data sheet in the Electrical Characteristics section.

PLLs The clock frequency is mainly determined by the input clock through the `PLL_CTRL` register `PLL_M` bit and `PLL_N` bit these two parameters Obtained by multiplication. Its calculation formula is as follows:

$$f_{pll} = \frac{f_{in}}{M} \times N$$

in `fpll` for PLLs output clock frequency, `f` is the input clock frequency. `m` for the register `PLL_M` configuration values, `N` for register `PLL_N` configuration value.

This chip PLLs The module is available through the register `CLK_SEL` of `PLL_CLK_SEL` Bits select two input clock sources, respectively `RCHF` or `XTAH`. in PLLs The reference clock frequency is input by the `clock/PLL_M` obtained, and the reference clock frequency range for 3MHz-6MHz.

This chipPLLSThe highest output frequency is72MHz, the input clock,PLL_Mas well asPLL_NThe configuration relationship is recommended to make

Use the following table:

input source	input frequency	PLL_M	PLLsreference clock frequency	PLL_N	PLLsclock
RCHF	twenty four	8	3	twenty four	72
RCHF	twenty four	6	4	18	72
RCHF	twenty four	4	6	12	72
RCHF	twenty four	6	4	16	64
RCHF	twenty four	6	4	14	56
RCHF	twenty four	6	4	12	48
XTAH	4	1	4	18	72
XTAH	8	2	4	18	72
XTAH	12	2	6	12	72
XTAH	12	4	3	twenty four	72
XTAH	16	4	4	18	72
XTAH	twenty four	4	6	12	72
XTAH	twenty four	6	4	18	72
XTAH	twenty four	8	3	twenty four	72
XTAH	32	8	4	18	72
XTAH	32	8	4	16	64

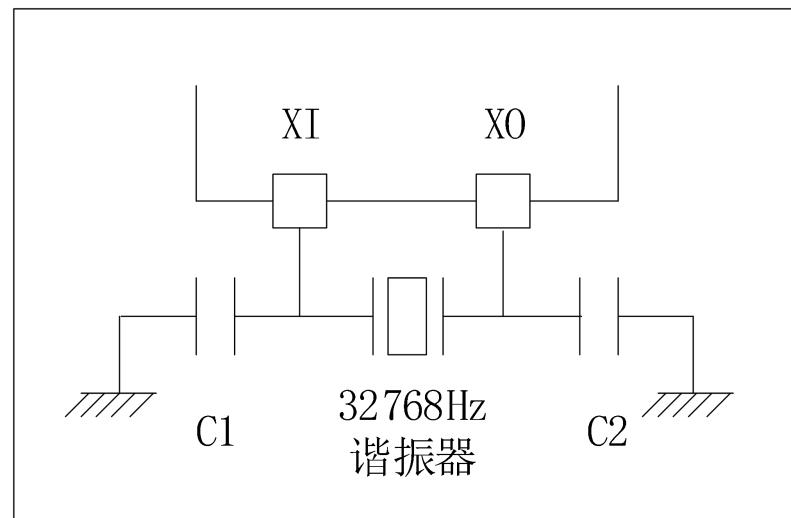
XTALclock

External low frequency crystal clock (XTAL) can be plugged in a32768Hzcrystal/ceramic resonators. It can be real time clock or other circuits to provide a low power and accurate32768Hzthe clock source.

XTALThe clock can be configured bySRC_CFGmiddleXTAL_ENbit to enable or disable. After starting, wait at least 2sto use normally.

XTALSupport plug-ins32768HzA crystal/ceramic resonator that provides a very accurate clock to the chip. related hardware configuration

The configuration can be referred to as shown in the figure below, and further information can be found in the electrical characteristics section of the data sheet.



picture5-13External low frequency crystal oscillator related hardware configuration diagram

RCLFclock

RCLFThe clock signal is provided by the chip internal low frequencyRCoscillator (32768Hz) generation, which acts as a low power low frequency clock source role, which can be found inSLEEPandDEEPSLEEPmode to keep running, forIWDT, real-time clock and wake-up circuit Clock provided.

RCLFThe clock frequency is32768Hz, and the clock cannot be turned off. Further information can be found in the data sheet in the Electrical Properties section content.

calibration

The manufacturing process determines the different chipRCThe oscillator frequency will vary, which is why each chip'sRCLF The clock frequency has been calibrated to $\pm 1\%$ (25°C) or less. After the system is reset, the software can Calibration values written to specific memory addresses during production testing are loaded into theTRIM_RCLFthe corresponding bit in the register, and can also Loads the actual frequency value written to a specific address in memory during production testing intoRC_FREQ_DELTAregister RCLF_DELTAandRCLF_SIGIn the bit, the deviation between the current frequency and the ideal frequency can be known during use, so as to obtain more Precise clock frequency.

If the user uses different voltage or ambient temperature in the use scene, it will affect RC oscillator precision. user also available through TRIM_RCLF in the register TRIM_FINE bit to adjust RCLF frequency.

system clock (sys_clk)choose

After a system reset, the system clock (sys_clk) by RCHF as the default clock source after power-up, and RCHF clock The output frequency is 24MHz.

Refer to the clock network structure, the system clock can pass through CLK_SEL in the register SYS_CLK_SEL at RCHF clock and DIV_CLK switch between clocks at any time to select.

and DIV_CLK If the clock source and frequency division of the clock are changed, the system clock must be changed to SYS_CLK_SEL configuration for 0, the system clock is selected as RCHF clock, so as to ensure that the system clock will not be affected during the process of switching the clock source ring. Next pass the DIV_CLK_GATE The register is configured as 0 closure DIV_CLK clock, and then through CLK_SEL in the register SRC_CLK_SEL bit and DIV_CLK_SEL bit selection DIV_CLK The frequency to be output by the clock, after selection Will DIV_CLK_GATE configured as 1, open DIV_CLK clock output, and finally configure the SYS_CLK_SEL for 1, making the system clock selected as DIV_CLK output.

RTCclock

by setting SRC_CFG in the register RTC_CLK_SEL bit, RTCCLK The clock source can be given by XTAL or RCLF hour clock provided.

exist RTC Before enabling, it must be completed RTC Related configurations (clock source selection, etc.), and should be changed at the same time RTC_CLK_SEL at least 70us Only then can it be enabled.

Whether in normal working mode, SLEEP mode or DEEPSLEEP In both modes, you can choose flexibly according to the application.

Independent Watchdog (IWDT)clock

The counting clock source of the independent watchdog is provided by RCLFClock is provided, this clock cannot be turned off, and is subject to DEV_CLK_GATE in the register IWDT_CLK_GATE bit control, only when the bit is configured as 1hour, IWDTCLK to use normally.

window watchdog (WWDT)clock

The counting clock of the window watchdog is the same frequency as the module bus clock and system clock, and is controlled by DEV_CLK_GATE in the register WWDT_CLK_GATE bit control, only when the bit is configured as 1hour, WWDTCLK to use normally.

SARADC sampling clock

SARADC The sampling clock can be passed through the CLK_SEL in the register SARADC_SMPL_CLK_SEL choose from the following four Frequency: system clock, system clock 2 Frequency division, system clock 4 Divider and System Clock 8 crossover.

Low Power Mode Wakeup Clock

exist SLEEP and DEEPSLEEP In both low power modes, only RCLF The clock is working, and the low power mode The wake-up is also driven by this clock.

128bit chip unique identification code

This chip has 128bit The unique identification code of the chip, each chip is unique.

Unique product identification is ideal for:



-Used as a serial number (e.g.USBcharacter serial number or other terminal applications);

- It is used as a password. When writing flash memory, use this unique identifier in combination with software encryption and decryption algorithms to improve generation code security in flash memory;

-Used to activate the bootstrap process with security mechanisms;

The identification code can be registered through the CHIP_ID0,CHIP_ID1,CHIP_ID2 and CHIP_ID3. These four registers should be 128bit. The device readout is used.

register map

name	Offset	Type	reset value	description
SYSCON	BASE: 0x40000000			
CLK_SEL	0x00	R/W	0x02	Clock Select Register
DIV_CLK_GATE	0x04	R/W	0x01	Divided Clock Gating Register
DEV_CLK_GATE	0x08	R/W	0x00	Peripheral Clock Gating Registers
RC_FREQ_DELTA	0x78	R/W	0x00	RCHF/RCLF True Frequency Value Difference Register
VREF_VOLT_DELTA	0x7C	R/W	0x00	VREF True Voltage Difference Register
CHIP_ID0	0x80	R/W	0x00	equipmentIDregister0
CHIP_ID1	0x84	R/W	0x00	equipmentIDregister1
CHIP_ID2	0x88	R/W	0x00	equipmentIDregister2
CHIP_ID3	0x8C	R/W	0x00	equipmentIDregister3
PLL_CTRL	0x180	R/W	0x2c8	PLLscontrol register
PLL_ST	0x184	R/W	0x00	PLLsstatus register

Register description

CLK_SELregister(0x00)

bit field	name	type	reset value	describe
31:12	RESERVED	R	0	reserved bit
11	PLL_CLK_SEL	R	0	PLLsInput Clock Selection 0:RCHF 1:XTAH
11:10	SARADC_SMPL_CLK_SEL	W	0	SARADC Sampling Clock Selection 00: of the system clock1frequency division 01: of the system clock2frequency division 10: of the system clock4frequency division 11: of the system clock8frequency division
10:9	SARADC_SMPL_CLK_SEL	R	0	SARADC Sampling Clock Selection 00: of the system clock1frequency division 01: of the system clock2frequency division 10: of the system clock4frequency division 11: of the system clock8frequency division
8	RESERVED	R	0	reserved bit
7	RESERVED	R	0	reserved bit
7	PLL_CLK_SEL	W	0	PLLsInput Clock Selection 0:RCHF 1:XTAH
6:4	SRC_CLK_SEL	R/W	0	source clock (SRC_CLK)choose 000:RCHF 001:RCLF 010:XTAH 011:XTAL 100:PLLs Other: reserved

3:1	DIV_CLK_SEL	R/W	01	Divided Clock (DIV_CLK)choose 000:SRC_CLKof1frequency division 001:SRC_CLKof2frequency division 010:SRC_CLKof4frequency division 011:SRC_CLKof8frequency division 100:SRC_CLKof16frequency division 101:SRC_CLKof32Crossover Other: Reserved
0	SYS_CLK_SEL	R/W	0	System Clock Selection 0:RCHFclock 1:DIV_CLKclock

DIV_CLK_GATEregister(0x04)

bit field	name	type	reset value	describe
31:1	RESERVED	R	0	reserved bit
0	DIV_CLK_GATE	R/W	0x1	Divided Clock Gating 1: Divided clock output 0: Divider clock disabled NOTE: When required to change DIV_CLK_SEL or SRC_CLK_SEL register, the system clock needs to be switched to RCHF, then set this register to 0, make DIV_CLK close, and finally change DIV_CLK_SEL or SRC_CLK_SEL The value of , so as to ensure the reliability of the clock.

DEV_CLK_GATEregister(0x08)

bit field	name	type	reset value	describe
31:29	RESERVED	R	0	reserved bit
28	AES_CLK_GATE	R/W	0	AES128Module Clock Gating

27	CRC_CLK_GATE	R/W	0	CRCModule Clock Gating
26	RESERVED	R	0	reserved bit
25	SARADC_CLK_GATE	R/W	0	SARADC_CTRLModule Clock Gating
twenty four	WWDT_CLK_GATE	R/W	0	WWDTModule Clock Gating
twenty three	IWDT_CLK_GATE	R/W	0	IWDTModule Clock Gating
twenty two	RTC_CLK_GATE	R/W	0	RTCModule Clock Gating
twenty one	PWM_PLUS1_CLK_GATE	R/W	0	PWM_PLUS1Module Clock Gating
20	PWM_PLUS0_CLK_GATE	R/W	0	PWM_PLUS0Module Clock Gating
19	RESERVED	R	0	reserved bit
18	PWM_BASE1_CLK_GATE	R/W	0	PWM_BASE1Module Clock Gating
17	PWM_BASE0_CLK_GATE	R/W	0	PWM_BASE0Module Clock Gating
16	TIMER_PLUS1_CLK_GATE	R/W	0	TIMER_PLUS1Module Clock Gating
15	TIMER_PLUS0_CLK_GATE	R/W	0	TIMER_PLUS0Module Clock Gating
14	TIMER_BASE2_CLK_GATE	R/W	0	TIMER_BASE2Module Clock Gating
13	TIMER_BASE1_CLK_GATE	R/W	0	TIMER_BASE1Module Clock Gating
12	TIMER_BASE0_CLK_GATE	R/W	0	TIMER_BASE0Module Clock Gating
11	SPI1_CLK_GATE	R/W	0	SPI1Module Clock Gating
10	SPI0_CLK_GATE	R/W	0	SPI0Module Clock Gating
9	RESERVED	R	0	reserved bit
8	UART2_CLK_GATE	R/W	0	UART2Module Clock Gating
7	UART1_CLK_GATE	R/W	0	UART1Module Clock Gating
6	UART0_CLK_GATE	R/W	0	UART0Module Clock Gating
5	IIC1_CLK_GATE	R/W	0	IIC1Module Clock Gating

4	IIC0_CLK_GATE	R/W	0	IIC0Module Clock Gating
3	RESERVED	R	0	reserved bit
2	GPIOC_CLK_GATE	R/W	0	GPIOCModule Clock Gating
1	GPIOB_CLK_GATE	R/W	0	GPIOBModule Clock Gating
0	GPIOA_CLK_GATE	R/W	0	GPIOAModule Clock Gating

RC_FREQ_DELTAregister(0x78)

bit field	name	type	reset value	describe
31	RCHF_SIG	R/W	0	1:expressRCHF_DELTAis a positive value 0 :expressRCHF_DELTAnegative value
30:11	RCHF_DELTA	R/W	0	RCHFThe actual test frequency and48MHzNote: The real frequency is48MHzand the sum of the differences.
10	RCLF_SIG	R/W	0	1:expressRCLF_DELTAis a positive value 0 :expressRCLF_DELTAnegative value
9:0	RCLF_DELTA	R/W	0	RCLFThe actual test frequency and32.768KHzNote: The real frequency is32.768KHzand the sum of the differences.

VREF_VOLT_DELTAregister(0x7C)

bit field	name	type	reset value	describe
31:7	RESERVED	R	0	reserved bit
6	VREF_SIG	R/W	0	1:expressVREF_DELTAis a positive value 0 :expressVREF_DELTAnegative value
5:0	VREF_DELTA	R/W	0	VREFThe difference between the actual test reference voltage value and the theoretical value (Unit ismv) Note: The actual voltage value is the sum of the theoretical value and the difference.

CHIP_ID0register(0x80)

bit field	name	type	reset value	describe
31:0	CHIP_ID0	R/W	0	equipmentIDregister0

CHIP_ID1register(0x84)

bit field	name	type	reset value	describe
31:0	CHIP_ID1	R/W	0	equipmentIDregister1

CHIP_ID2register(0x88)

bit field	name	type	reset value	describe
31:0	CHIP_ID2	R/W	0	equipmentIDregister2

CHIP_ID3register(0x8C)

bit field	name	type	reset value	describe
31:0	CHIP_ID3	R/W	0	equipmentIDregister3

PLL_CTRLregister(0x180)

bit field	name	type	Reset value	description

31:11	RESERVED	R	0	reserved bit
10:6	PLL_M	R/W	0xb	PLLsReference clock frequency division 00000:1frequency division 00001:2frequency division 00010:3frequency division 00011:4frequency division ... 11110:31frequency division 11111:32frequency division
5:1	PLL_N	R/W	0x4	PLL Feedbackclock frequency division 00000:2frequency division 00001:4frequency division 00010:6frequency division 00011:8frequency division ... 11110:62frequency division 11111:64frequency division
0	PLL_EN	R/W	0	PLLenable control bit 0:closurePLLs 1: openPLLs Note:PLLsWhen enabled, at least30usto lock.

PLL_STregister(0x184)

bit field	name	type	reset value	describe
31:1	RESERVED	R	0	reserved bit
0	PLL_LOCK	R	0	PLLslock status bit 0: unlocked 1: locked, ready to usePLLsclock

5.7 I/O Function configuration (PORTCON)

5.7.1 Overview

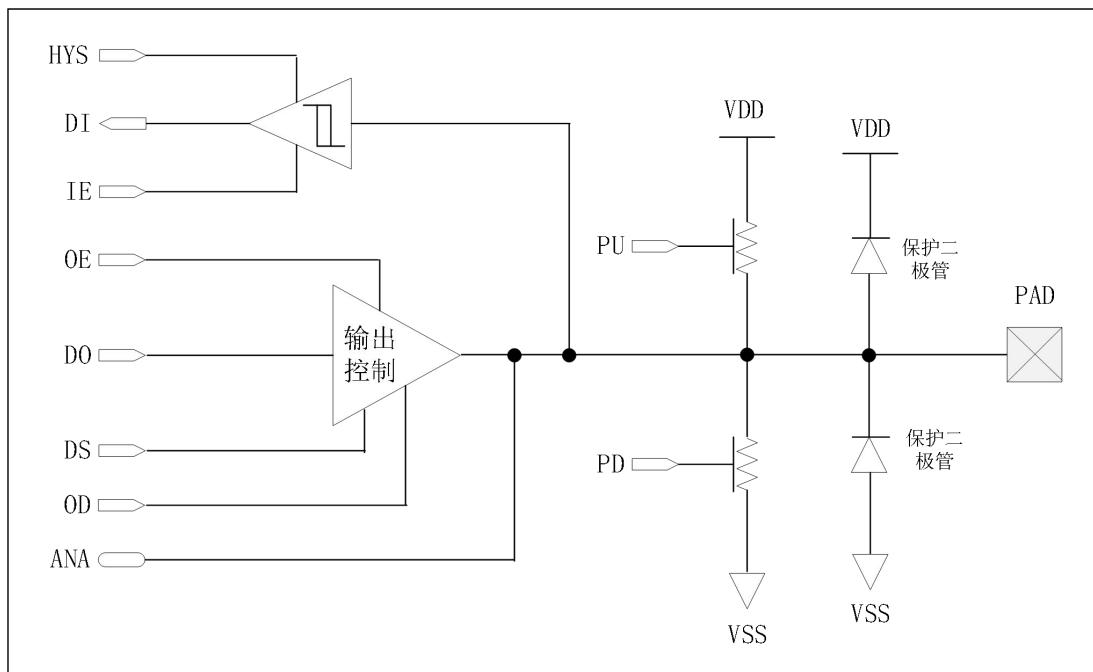
eachIOThe corresponding functions can be configured independently, throughPORTx_SELxregister to select theIOWhat is the function of multiplexing can, and according to eachIOThe specific hardware characteristics of the port are configured by software setting corresponding registers into various modes.

5.7.2 Characteristics

- eachIODepending on the multiplexed digital functions, it can be configured as a specific digital function
- eachIOAccording to the different analog functions it multiplexes, it can be configured as a specific analog function
- Independent input pull-up enable control
- Independent input pull-down enable control
- Independent open-drain output or push-pull output mode
- Independent Input Enable Control
- independentIOWake up enable
- specificIOWake-up active edge selection control
- Input hysteresis selection control
- Output drive capability selection control
- Input pull-up resistor value selection control

The above functions, most functions eachIOPorts can be programmed independently and freely, and some function controls are global, to allIOr it works, so it is selected and used according to the description of the corresponding function register during configuration.

5.7.3 Block Diagram of Module Structure



picture5-14 PORTCON Block Diagram of Module Structure

in:

HYSSelect the control terminal for input hysteresis, which can be uniformly configured by a specific register;

IEEnable the control terminal for the input, which can be controlled byIODirect configuration of independent registers;

DSIt is the output drive capability control terminal, which can be uniformly configured by specific registers;

ODIt is the output mode control terminal, which can be controlled byIODirect configuration of independent registers;

PUEnable the control terminal for the input pull-up, which can be controlled byIODirect configuration of independent registers;

PDEnable control pin for input pull-down, can be controlled byIODirect configuration of independent registers;

ANAFor the analog signal channel, directly communicate with theIOEach custom analog signal connection;

DIfor the reasonPADInput to the input signal terminal inside the chip. Selected through digital function multiplexing, can be used as the corresponding function

input signal;

dois output by the chip internally to thePADoutput signal terminal. Selected through digital function multiplexing, can be used as the corresponding function

output signal;

OEEnable control pin for output. its withoco-exist, shouldIOWhen the configured function is output, the hardware will

OEautomatically open, otherwise theOEclosure;

5.7.4 Functional description

pin input enable

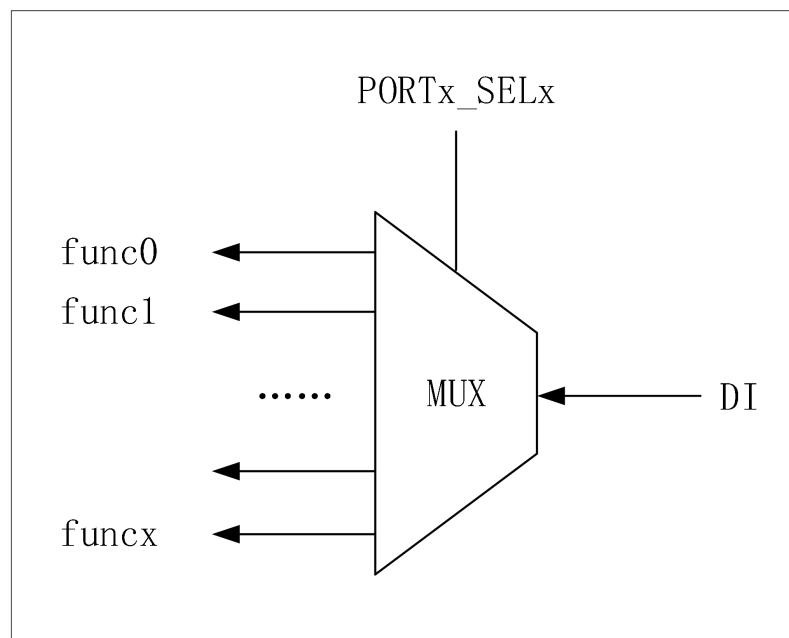
When the pins of this chip are used as inputs or peripherals that require input, the corresponding IOThe input enable register of (PORTx_IE) configuration is valid. whenIOThe corresponding input enable register is configured as1whenIEvalid, input enable Open,PADThe level state on the chip can be input into the chip toDIterminal to obtain the current external state of the chip.

Digital multiplexing function selection

differentIOCan be multiplexed into different digital functions such asGPIOs,SPI,UARTwait,IOThe direction of the multiplexed specific functional requirements. available byPORTx_SELxRegisters directly configure eachIOfunction to be implemented.

-Enter function description:

The input function selection diagram is shown in the figure below:



picture5-15 IOInput function selection diagram

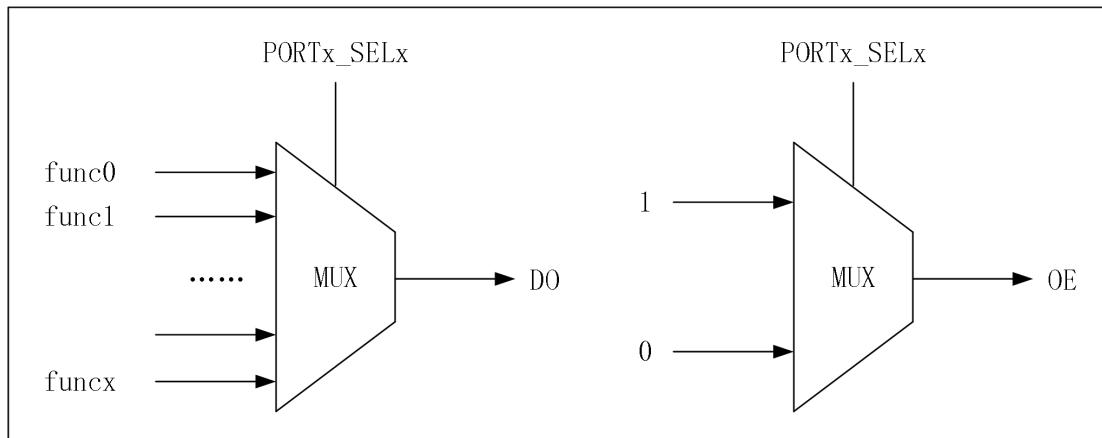
func0,func1,...,funcxetc. are digital signals input to the chip.



passPORTx_SELxregister will be aIOConfigured as a specific digital function, if the function is an input, through PORTx_SELxchoose toDI signal input to one of thefuncsignal, otherfuncsignal level0. pass PORTx_SELxCan get through fromDito specific digital signal paths, corresponding toIOofIEWhen turned on, the input signal fromPADThe terminal directly enters the chip to give the selected digital signal.

-Output function description:

The schematic diagram of output function selection is shown in the figure below:



picture5-16 IOSchematic diagram of output function selection

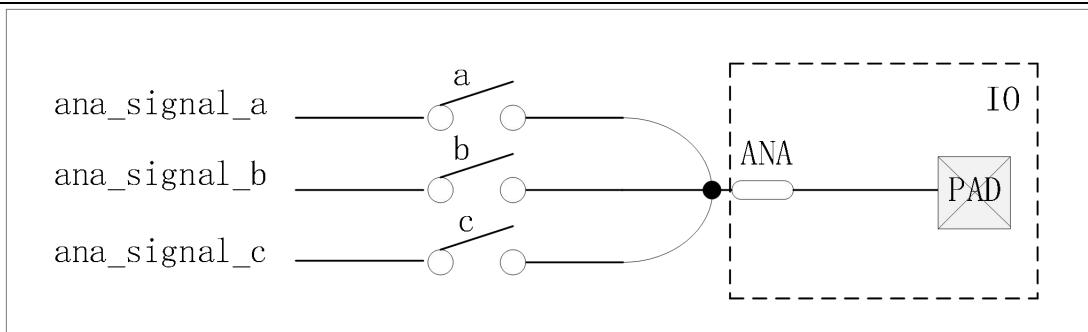
func0,func1,...,funcxWaiting is the internal digital signal of the chip to be output.
passPORTx_SELxregister will be aIOConfigured as a specific digital function, if the function is an output, through PORTx_SELxSelect to output the selected digital signal to do, otherfuncsignal will not strobe the output, and the corresponding IO ofOEend configuration is valid. It can realize the output of the digital signal level inside the chip to the off-chipPAD, for off-chip acquisition.

Analog multiplexing function selection

when passedPORTx_SELxregister will be specificIOWhen selected as an analog function, IOofOterminal will be closed directly, and the same need to set the corresponding input enable register (PORTx_IE) configured as0, WillIOofIEterminal closed. In addition, in order to protect link toPADthe analog signal integrity is not disturbed, the software needs to ensure that theIOThe pull-up and pull-down resistors have no effect (by closing the correspondingIOThe pull-up enable (PORTx_PU) and pull-down enable (PORTx_PD) to control).

The following is an example to illustrate the analog signal andIOconnections and control relationships.

Analog signal connectionIOThe schematic diagram is as follows:

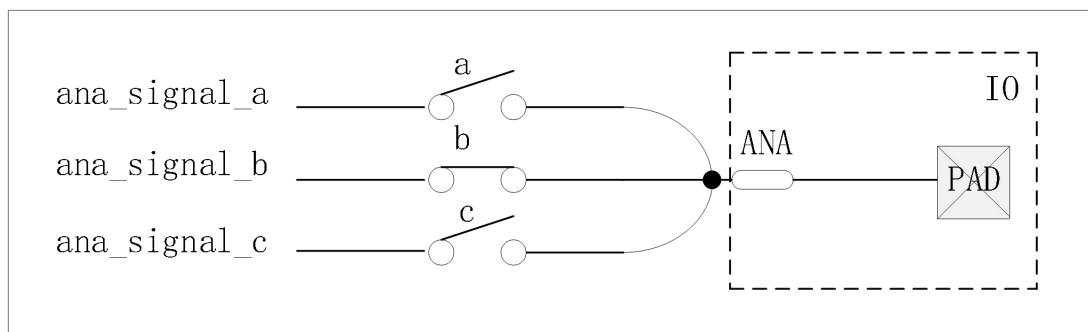


picture5-17 Analog signal connection|Schematic diagram

In the picture above `ana_signal_a`, `ana_signal_b`, `ana_signal_c` for multiplexing to the same IO. The three analog signals on the

The three analog signals are controlled by their respective analog switches `a`, `b`, `c`, these switches are controlled within their respective modules and are not in the Unified control within the module. When the analog signal is not multiplexed, the switch is turned off and the analog signal will not be connected to `ANAend`, thereby `PAD`. The level and behavior on the chip will not affect the analog signal inside the chip.

when needed will `IO`. When it is multiplexed as one of the analog functions, close the corresponding switch so that the analog signal can be connected through to `IO` of `ANAend`. As shown below, `IO` Multiplexed as `ana_signal_b` function of the analog signal.



picture5-18 `IO` Multiplexed as `ana_signal_b` Functional diagram of the analog signal

Pull-up/down resistor enable

All of this chip `IO` Both have independent input pull-up or input pull-down configurable functions.

as shown in the picture `IO`s shown in the schematic diagram of the basic structure of the port, the pull-up resistor passes through the `PUport` control, the pull-down resistor is passed through the `PDport` control.

when `IO` of `I` terminal through the input enable register (`PORTx_IE`) After valid configuration, enable the `can register` (`PORTx_PU`) configured as 1. After that, the `IO` of `PUport` is set to 1, then even if `PAD` for the floating state, `DI` terminal is also logic level 1.

when IO of I terminal through the input enable register (PORTx_IE) After the configuration is valid, enable the can register (PORTx_PD) configured as 1. After that, the IO of PD port is set to 1, then even if PAD for the floating state, DI terminal is also logic level 0.

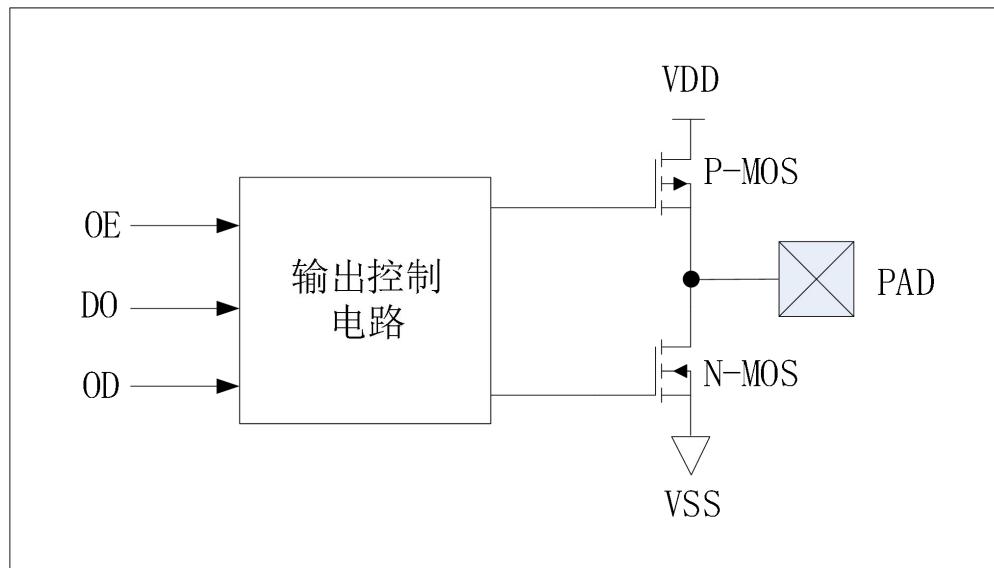
Note: same IO. The pull-up enable and pull-down enable cannot be configured as valid at the same time. If the chip is configured as a If it is valid, the pull-down resistor works, but the pull-up resistor is invalid.

Output mode selection

All of this chip IO with independent output mode selection control, it can be independently configured as output open-drain mode or output push-pull model.

when IO When used as output, OE is high, can be configured by the open-drain enable register (PORTx_OD) for 0 or 1, make PA The output state can be configured as push-pull output or open-drain output.

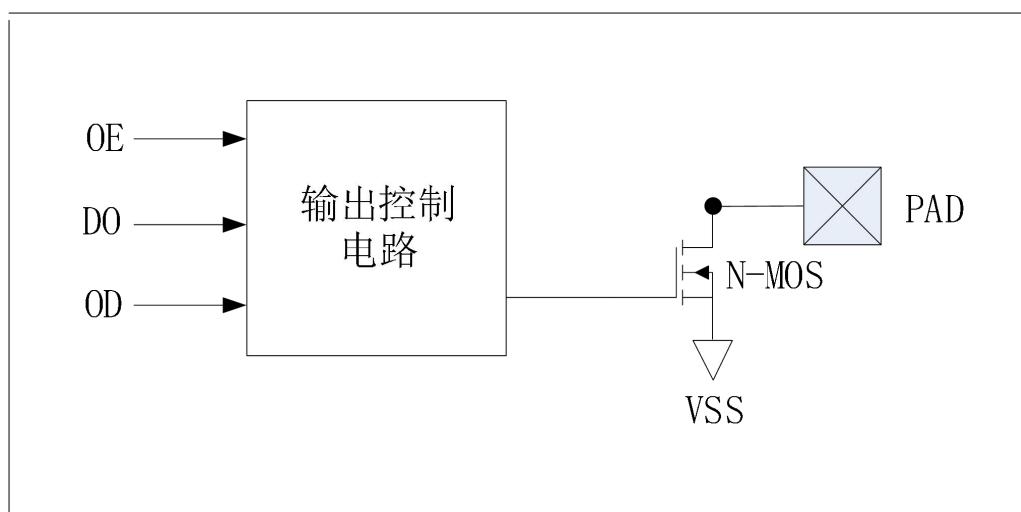
when IO The open-drain enable register (PORTx_OD) configured as 0, it is the push-pull output mode, at this time IO with power pull ability to source/sink current. As shown below:



picture5-19 IO Schematic diagram of push-pull output mode

when OD = 0, and OE = 1 when do for 0 hour, PAD To be strong 0, when do for 1 hour, PAD To be strong 1.

when IO The open-drain enable register (PORTx_OD) configured as 1, it is open-drain output mode, at this time IO only has irrigation Current capability, not pull current capability. As shown below:



picture5-20 IOschematic diagram of open-drain output mode

whenOD = 1, andOE = 1 when do for 0 hour, PADTo be strong0, when do for 1 hour, PADis high-impedance,

If you need to output high level, you need to connect the pull-up resistor to the power supply on the external pin, and realize high-level output through external pull-up.

Pin wake-up function

This chip supports pin wake-up function in low power consumption mode. provides eachIOIndependent wake-up control for on-chip sheet provides flexibility and convenience in use!Wake up method.

eachIOare enabled via the wake-up enable register (PORTx_WKE) independently configured as a wake-up pin. and

And can support wake-up edge or level configurable.

passPORT_RISRegisters are available for allIOIndependently configured as rising edge wake-up or falling edge wake-up.

Note0: In this chip,RISis defined as when configured as0, it is active low; when configured as1time, to rise valid along;

Output drive capability selection

different groups!Ohave independent output drive capability configuration control, which can be controlled by the respectivePORT_DSRegister to select.

Generally speaking!OThe output drive capability can be configured into the following four modes:5mA,14mA,22mAand30mA.

NOTE: This register is!Oglobal control register.

Input Hysteresis Selection

BookIOProvides two kinds of input hysteresis gear selection control, the user can select according to the system conditions and the level of the external signal

Make a reasonable selection configuration.

Input hysteresis is available through thePORT_HYSRegister control. When this register is configured as0, selectable low input hysteresis configuration set; when this register is configured as1, a high input hysteresis configuration can be selected.

NOTE: This register isIOglobal control register.

Pull-up resistor value selection

For the convenience of customers, theIOThe option of configurable pull-up resistor value is provided so that the user can choose a more

Add reasonable pull-up resistors.

The pull-up resistor value can be set byPORT_PURRegister configuration. There are three resistance values to choose from:32kΩ,40kΩand150kΩ.

NOTE: This register isIOglobal control register.

register map register map

name	offset	bit width	type	reset value	describe
port BASE: 0x400B0000					
PORTA_SEL0	0x00	32	R/W	0x00	PORTAFunction Select Register0
PORTA_SEL1	0x04	32	R/W	0x00	PORTAFunction Select Register1
PORTB_SEL0	0x08	32	R/W	0xffff0000	PORTBFunction Select Register0
PORTB_SEL1	0x0C	32	R/W	0x01001000	PORTBFunction Select Register1
PORTC_SEL0	0x10	32	R/W	0x00	PORTCFunction Select Register0
PORTA_IE	0x100	32	R/W	0x00	PORTAinput enable register
PORTB_IE	0x104	32	R/W	0x4800	PORTBinput enable register
PORTC_IE	0x108	32	R/W	0x20	PORTCinput enable register
PORTA_PU	0x200	32	R/W	0x00	PORTApull-up enable register
PORTB_PU	0x204	32	R/W	0x00	PORTBpull-up enable register

PORTC_PU	0x208	32	R/W	0x00	PORTCpull-up enable register
PORTA_PD	0x300	32	R/W	0x00	PORTApull-down enable register
PORTB_PD	0x304	32	R/W	0x00	PORTBpull-down enable register
PORTC_PD	0x308	32	R/W	0x20	PORTCpull-down enable register
PORTA_OD	0x400	32	R/W	0x00	PORTAOpen Drain Enable Register
PORTB_OD	0x404	32	R/W	0x00	PORTBOpen Drain Enable Register
PORTC_OD	0x408	32	R/W	0x00	PORTCOpen Drain Enable Register
PORTA_WKE	0x500	32	R/W	0x00	PORTAwakeup enable register
PORTB_WKE	0x504	32	R/W	0x00	PORTBwakeup enable register
PORTC_WKE	0x508	32	R/W	0x00	PORTCwakeup enable register
PORT_CFG	0x600	32	R/W	0x15	portconfiguration register
PORTA_WK_SEL	0x700	32	R/W	0x00	PORTAwakeIOedge selection register
PORTB_WK_SEL	0x704	32	R/W	0x00	PORTBwakeIOedge selection register
PORTC_WK_SEL	0x708	32	R/W	0x00	PORTCwakeIOedge selection register

Register description

PORTA_SEL0register(0x00)

bit field	name	type	reset value	describe
31:28	PORTA7	R/W	0	0000:GPIOA7 0001:UART1_TX 0010:TIMERPO_IN0 0011:TIMERPO_OUT_L 0100:SARADC_CH2 0101:OPA0_VP Other: reserved

27:24	PORATA6	R/W	0	0000:GPIOA6 0001:UART1_RTS 0010:TIMERP1_IN1 0011:TIMERP1_OUT_H 0100:SARADC_CH1 0101:OPA0_OUT Other: reserved
23:20	PORATA5	R/W	0	0000:GPIOA5 0001:UART1_CTS 0010:PWMP1_PLUS1 0011:TIMERP1_IN0 0100:TIMERP1_OUT_L 0101:WAKEUP1 0110:SARADC_CH0 Other: reserved
19:16	PORATA4	R/W	0	000:GPIOA4 001:CMP0_VP 010:XTAH_XO Other: reserved
15:12	PORATA3	R/W	0	000:GPIOA3 001:CMP0_VN 010:XTAH_XI Other: reserved
11:8	PORATA2	R/W	0	000:GPIOA2 001:XTAL_XO Other: reserved
7:4	PORATA1	R/W	0	000:GPIOA1 001:XTAL_XI Other: reserved

3:0	PORATA0	R/W	0	0000:GPIOA0 0001:PWMP1_PLUS0 0010:PWMP0_PLUS1 0011:tm 0100:WAKEUP0 Other: reserved
-----	---------	-----	---	---

PORATA_SEL1register(0x04)

bit field	name	type	reset value	describe
31:28	PORATA15	R/W	0	0000:GPIOA15 0001:PWMB1_CH1 0010:PWMP0_CH0 0011:TIMERP1_IN1 0100:TIMERP1_OUT_H Other: reserved
27:24	PORATA14	R/W	0	0000:GPIOA14 0001:PWMB1_CH0 0010:PWMP0_CH2N 0011:TIMERP1_IN0 0100:TIMERP1_OUT_L 0101:SARADC_CH9 Other: reserved
23:20	PORATA13	R/W	0	0000:GPIOA13 0001:PWMB0_CH2 0010:PWMP0_CH1N 0011:TIMERP0_IN1 0100:TIMERP0_OUT_H 0101:SARADC_CH8 Other: reserved

19:16	PORTA12	R/W	0	0000:GPIOA12 0001:SPI0_MOSI 0010:PWMB0_CH1 0011:PWMP0_CH0N 0100:TIMERP0_IN0 0101:TIMERP0_OUT_L 0110:SARADC_CH7 Other: reserved
15:12	PORTA11	R/W	0	0000:GPIOA11 0001:SPI0_MISO 0010:PWMB0_CH0 0011:PWMP0_BRAKE0 0100:TIMERP1_IN1 0101:TIMERP1_OUT_H 0110:SARADC_CH6 Other: reserved
11:8	PORTA10	R/W	0	0000:GPIOA10 0001:SPI0_CLK 0010:SARADC_CH5 0011:CMP1_VP Other: reserved
7:4	PORTA9	R/W	0	0000:GPIOA9 0001:SPI0_SSN 0010:TIMERP1_IN0 0011:TIMERP1_OUT_L 0100:tm 0101:SARADC_CH4 0110:CMP1_VN Other: reserved

3:0	PORTA8	R/W	0	0000:GPIOA8 0001:UART1_RX 0010:TIMERP0_IN1 0011:TIMERP0_OUT_H 0100:SARADC_CH3 0101:OPA0_VN Other: reserved
-----	--------	-----	---	--

PORTE_SEL0register(0x08)

bit field	name	type	reset value	describe
31:28	PORTE7	R/W	0xf	0000:GPIOB7 0001:SPI0_SS_N 0010:UART0_TX 0011:IIC0_SCL 0100:PWMP1_BRAKE0 0101:PWMP0_CH1 Other: reserved
27:24	PORTE6	R/W	0xf	0000:GPIOB6 0001:PWMP0_CH0 0010:TIMERP0_IN1 0011:TIMERP0_OUT_H Other: reserved
23:20	PORTE5	R/W	0xf	0000:GPIOB5 0001:SPI1_MOSI 0010:PWMP1_CH0_N 0011:PWMP0_CH2_N 0100:TIMERP0_IN0 0101:TIMERP0_OUT_L Other: reserved



bit field	name	type	Reset value	description
19:16	PORTB4	R/W	0xf	0000:GPIOB4 0001:SPI1_MISO 0010:IIC1_SCL 0011:PWMP1_CH0 0100:PWMP0_CH1N 0101:TIMERP1_HALL2 Other: reserved
15:12	PORTB3	R/W	0	0000:GPIOB3 0001:SPI1_CLK 0010:IIC1_SDA 0011:PWMP0_CH0N 0100:TIMERP1_HALL1 Other: reserved
11:8	PORTB2	R/W	0	0000:GPIOB2 0001:SPI1_SS_N 0010:PWMP0_BRAKE1 0011:TIMERP1_HALL0 Other: reserved
7:4	PORTB1	R/W	0	0000:GPIOB1 0001:UART2_RX 0010:IIC0_SDA 0011:PWMP0_CH2 Other: reserved
3:0	PORTB0	R/W	0	0000:GPIOB0 0001:UART2_TX 0010:IIC0_SCL 0011:PWMB1_CH2 0100:PWMP0_CH1 Other: reserved

PORTB_SEL1register(0x0C)

bit field	name	type	Reset value	description

31:28	PORTB15	R/W	0	0000:GPIOB15 0001:SPI1_SS_N 0010:UART2_RX Other: reserved
27:24	PORTB14	R/W	0x1	0000:GPIOB14 0001:SWCLK 0010:UART2_TX 0011:PWMP1_CH2N Other: reserved
23:20	PORTB13	R/W	0	0000:GPIOB13 0001:UART1_RX 0010:IIC1_SDA 0011:PWMP1_CH1N Other: reserved
19:16	PORTB12	R/W	0	0000:GPIOB12 0001:UART1_TX 0010:IIC1_SCL 0011:PWMP1_CH0N Other: reserved
15:12	PORTB11	R/W	0x1	0000:GPIOB11 0001:SWDIO 0010:PWMP1_CH2 0011:PWMP0_BRAKE2 Other: reserved
11:8	PORTB10	R/W	0	0000:GPIOB10 0001:SPI0_MOSI 0010:UART0_RTS 0011:PWMB0_CH2 0100:PWMP1_CH1 0101:PWMP0_PLUS0 0110:TIMERP1_IN0 0111:TIMERP1_OUT_L Other: reserved

7:4	PORTB9	R/W	0	0000:GPIOB9 0001:SPI0_MISO 0010:UART0_CTS 0011:PWMB0_CH1 0100:PWMP1_CH0 0101:TIMERP1_IN1 0110:TIMERP1_OUT_H Other: reserved
3:0	PORTB8	R/W	0	0000:GPIOB8 0001:SPI0_CLK 0010:UART0_RX 0011:IIC0_SDA 0100:PWMB0_CH0 0101:PWMP1_BRAKE1 0110:PWMP0_CH2 Other: reserved

PORTC_SEL0register(0x10)

bit field	name	type	reset value	describe
31:28	PORTC7	R/W	0	0000:GPIOC7 0001:IIC1_SDA 0010:PWMP1_CH2 0011:TIMERP1_IN0 0100:TIMERP1_OUT_L 0101:OPA1_OUT Other: reserved
27:24	PORTC6	R/W	0	0000:GPIOC6 0001:IIC1_SCL 0010:PWMP1_CH1 0011:TIMERP1_IN1 0100:TIMERP1_OUT_H 0101:OPA1_VN Other: reserved

23:20	PORTC5	R/W	0	0000:GPIOC5 0001:TIMERPO_HALL2 0010:tm 0011:OPA1_VP Other: reserved
19:16	PORTC4	R/W	0	0000:GPIOC4 0001:UART0_RX 0010:IIC0_SDA 0011:PWMP1_CH2N 0100:TIMERPO_HALL1 0101:CMP2_VP Other: reserved
15:12	PORTC3	R/W	0	0000:GPIOC3 0001:UART0_TX 0010:IIC0_SCL 0011:PWMP1_CH1N 0100:TIMERPO_HALL0 0101:CMP2_VN Other: reserved
11:8	PORTC2	R/W	0	0000:GPIOC2 0001:SPI1_MOSI 0010:PWMB1_CH2 0011:PWMP1_BRAKE2 0100:TIMERPO_IN1 0101:TIMERPO_OUT_H Other: reserved
7:4	PORTC1	R/W	0	0000:GPIOC1 0001:SPI1_MISO 0010:UART2_RTS 0011:PWMB1_CH1 0100:TIMERPO_IN0 0101:TIMERPO_OUT_L Other: reserved

3:0	PORTC0	R/W	0	0000:GPIOC0 0001:SPI1_CLK 0010:UART2_CTS 0011:PWMB1_CH0 Other: reserved
-----	--------	-----	---	---

POR TA _IE register(0x100)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	POR TA _IE	R/W	0x00	POR TA input enable register 0: disabled 1:Enable (each bit correspond 1 individual IO, bit 0 correspond A0, bit 1 correspond A1)

POR TB _IE register(0x104)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	POR TB _IE	R/W	0x4800	POR TB input enable register 0: disabled 1:Enable (each bit correspond 1 individual IO, bit 0 correspond B0, bit 1 correspond B1)

POR TC _IE register(0x108)

bit field	name	type	reset value	describe
31:8	RESERVED	R	0	reserved bit
7:0	POR TC _IE	R/W	0x20	POR TC input enable register 0: disabled 1:Enable (each bit correspond 1 individual IO, bit 0 correspond C0, bit 1 correspond C1)

PORTA_PUregister(0x200)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	PORTA_PU	R/W	0	PORTApull-up enable register 0: disabled 1:Enable (eachbitcorrespond1indivualIO,bit0correspondA0,bit1correspondA1)

PORTB_PUregister(0x204)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	PORTB_PU	R/W	0	PORTBpull-up enable register 0: disabled 1:Enable (eachbitcorrespond1indivualIO,bit0correspondB0,bit1correspondB1)

PORTC_PUregister(0x208)

bit field	name	type	reset value	describe
31:8	RESERVED	R	0	reserved bit
7:0	PORTC_PU	R/W	0	PORTCpull-up enable register 0: disabled 1:Enable (eachbitcorrespond1indivualIO,bit0correspondC0,bit1correspondC1)

PORTA_PDregister(0x300)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit

15:0	PORTA_PD	R/W	0	PORTApull-down enable register 0: disabled 1:Enable (eachbitcorrespond1indivualIO,bit0correspondA0,bit1correspondA1)
------	----------	-----	---	--

PORTB_PDregister(0x304)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	PORTB_PD	R/W	0	PORTBpull-down enable register 0: disabled 1:Enable (eachbitcorrespond1indivualIO,bit0correspondB0,bit1correspondB1)

PORTC_PDregister(0x308)

bit field	name	type	reset value	describe
31:8	RESERVED	R	0	reserved bit
7:0	PORTC_PD	R/W	0x20	PORTCpull-down enable register 0: disabled 1:Enable (eachbitcorrespond1indivualIO,bit0correspondC0,bit1correspondC1)

PORTA_ODregister(0x400)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	PORTA_OD	R/W	0	PORTAOpen Drain Enable Register 0: disabled 1:Enable (eachbitcorrespond1indivualIO,bit0correspondA0,bit1correspondA1)

PORTB_ODregister(0x404)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	PORTB_OD	R/W	0	PORTBOpen Drain Enable Register 0: disabled 1:Enable (eachbitcorrespond1indivualIO,bit0correspondB0,bit1correspondB1)

PORTC_ODregister(0x408)

bit field	name	type	reset value	describe
31:8	RESERVED	R	0	reserved bit
7:0	PORTC_OD	R/W	0	PORTCOpen Drain Enable Register 0: disabled 1:Enable (eachbitcorrespond1indivualIO,bit0correspondC0,bit1correspondC1)

PORTA_WKEregister(0x500)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	PORTA_WKE	R/W	0	PORTAwakeup enable register 0: disabled 1:Enable (eachbitcorrespond1indivualIO,bit0correspondA0,bit1correspondA1)

PORTB_WKEregister(0x504)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit

15:0	PORTB_WKE	R/W	0	PORTBwakeup enable register 0: disabled 1:Enable (eachbitcorrespond1indivualIO,bit0correspondB0,bit1correspondB1)
------	-----------	-----	---	---

PORTC_WKEregister(0x508)

bit field	name	type	reset value	describe
31:8	RESERVED	R	0	reserved bit
7:0	PORTC_WKE	R/W	0	PORTCwakeup enable register 0: disabled 1:Enable (eachbitcorrespond1indivualIO,bit0correspondC0,bit1correspondC1)

PORT_CFGregister(0x600)

bit field	name	type	reset value	describe
31:11	RESERVED	R	0	reserved bit
10	PORT_HYS	R/W	0	portInput hysteresis level selection 0: Low input hysteresis (input signal greater than 0.7VDD and less than 0.3VDD) 1: High input hysteresis (input signal greater than 0.85VDD and less than 0.15VDD)
9:6	RESERVED	R	0	reserved bit
5:4	PORTC_DS	R/W	0x01	PORTCDrive Capability Selection Register 00:5mA 01:10mA 10:15mA 11:20mA
3:2	PORTB_DS	R/W	0x01	PORTBDrive Capability Selection Register 00:5mA 01:10mA 10:15mA 11:20mA

1:0	PORTA_DS	R/W	0x01	PORTADrive Capability Selection Register 00:5mA 01:10mA 10:15mA 11:20mA
-----	----------	-----	------	---

PORTA_WK_SELregister(0x700)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	PORTA_WK_SEL	R/W	0	PORTAThe wake-up function is configured along the 0:PORTAThe falling edge of the wake-up function is valid 1: PORTAThe rising edge of the wake-up function is valid

PORTB_WK_SELregister(0x704)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	PORTB_WK_SEL	R/W	0	PORTBThe wake-up function is configured along the 0:PORTBThe falling edge of the wake-up function is valid 1: PORTBThe rising edge of the wake-up function is valid

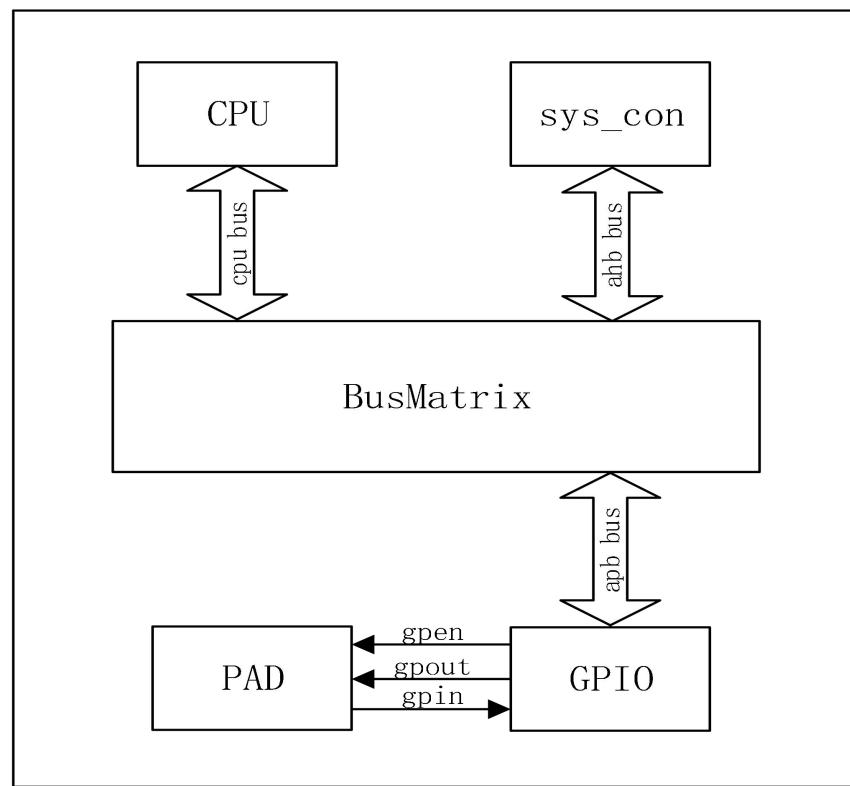
PORTC_WK_SELregister(0x708)

bit field	name	type	reset value	describe
31:8	RESERVED	R	0	reserved bit
7:0	PORTC_WK_SEL	R/W	0	PORCThe wake-up function is configured along the 0:PORCThe falling edge of the wake-up function is valid 1: PORCThe rising edge of the wake-up function is valid

5.8 universalIO (GPIO)

5.8.1 overview

GPIO module implements a general-purpose programmable IO interface, supports configurable input and output modes, can be used to realize serial data communication. Correspondence needs to be enabled before use GPIOs. The module's clock. The system diagram is as follows:



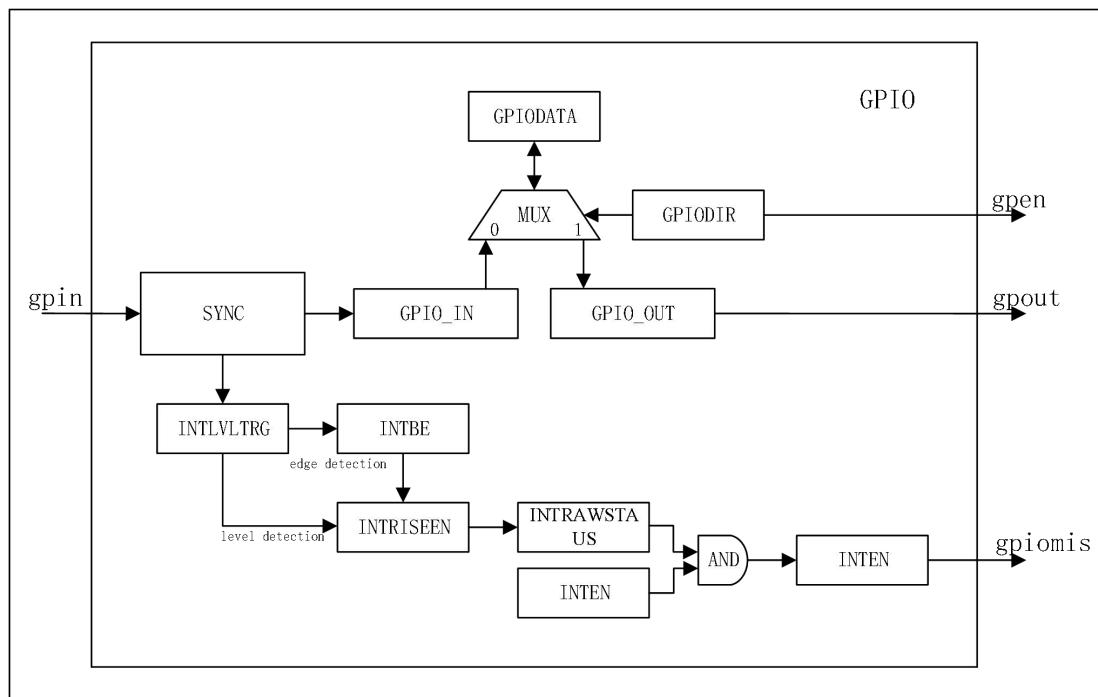
picture5-21 GPIOsModule System Block Diagram

5.8.2 characteristic

- most 40 independent IO
- each IO with interrupt entry
- Configurable interrupt trigger conditions, support level trigger and edge trigger
- Level trigger supports high level and low level
- Edge trigger supports rising edge, falling edge and double edge trigger

eachIOAll support pull-up, pull-down, push-pull, open-drain functions

5.8.3 Block Diagram of Module Structure



picture5-22 GPIOsBlock Diagram of Module Structure

Pictured above isGPIOsSchematic diagram of the internal structure of the module. As shown in the figure above, the external input datagpinAfter synchronous processing circuit syncprocessing, the detection of edge-triggered interrupts is also used forAPBThe bus reads the input data.`gpin`The signal does not pass through Generate interrupt status with interrupt setting`INTRAWSTA US`.`INTRAWSTA US`The signal is output after being controlled by the interrupt enable register. out interrupt signal`gpiomis`to the system. The clock source in this module is`pcclk`, same source as the system clock, available through`SYS CON` in the module`DEV_CLK_GATE`Register to configure the enable of the clock of this module.

5.8.4 Functional description

direction control

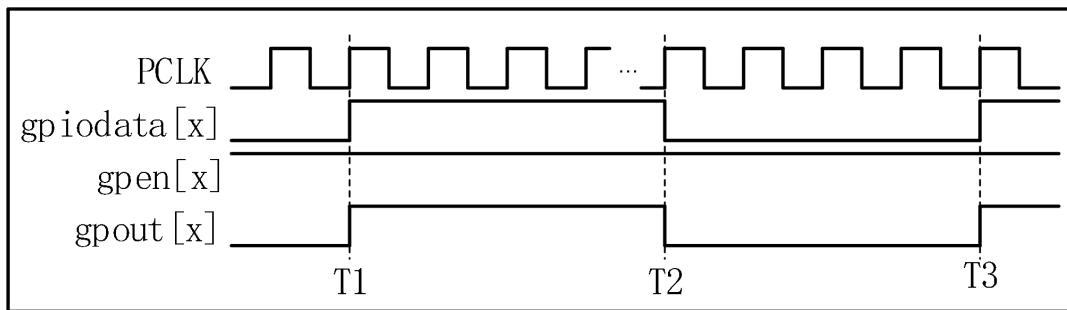
- removeSWDThe default state of all pins after power-on isGPIOsFloating input (`DIR = 0`).
- GPIOsdirection register (`DIRx`) is used to configure each individual pin as input mode or output mode
- When the data direction is set to0hour,GPIOsThe corresponding pin is configured as an input

By reading the corresponding data register (DATA) corresponding bit gets specifiedGPIOsPort current state value

- When the data direction is set to 1hour,GPIOsThe corresponding pin is configured as an output

By writing to the corresponding port data register (DATA) corresponding bit write value changes the specified pin output,0output low level, 1output high level.

GPIOsThe output timing diagram of is as follows:



picture5-23 GPIOsThe output timing diagram of

set upGPIOsmodule correspondingGPIOsbit is an output pin.

T1moment, writegpidata[x]value is 1,GPIOsoutputgout[x]goes high.

T2moment, writegpidata[x]value is 0,GPIOsoutputgout[x]goes low.

T3moment, writegpidata[x]value is 1,GPIOsoutputgout[x]goes high.

GPIOsoutputgout[x]withgpidata[x]change with changes.

Interrupt configuration and clearing

According to the demand willGPIOsThe corresponding pin of the port is configured as interrupt mode, and the interrupt polarity is configured through the relevant register

and trigger mode.

There are two trigger modes: edge trigger and level trigger.

- For edge-triggered interrupts, it can be set to trigger on rising edge, falling edge or both edges. After an interrupt occurs,

The flag bit has a retention feature, and the interrupt flag bit must be cleared by software

- For level-triggered interrupts, when the external pin input is at a specified level, the interrupt occurs. When the level flips, the interrupt signal

The number disappears and no software is required to clear it. When using level-triggered interrupts, it is necessary to ensure that the external signal source maintains a stable level to

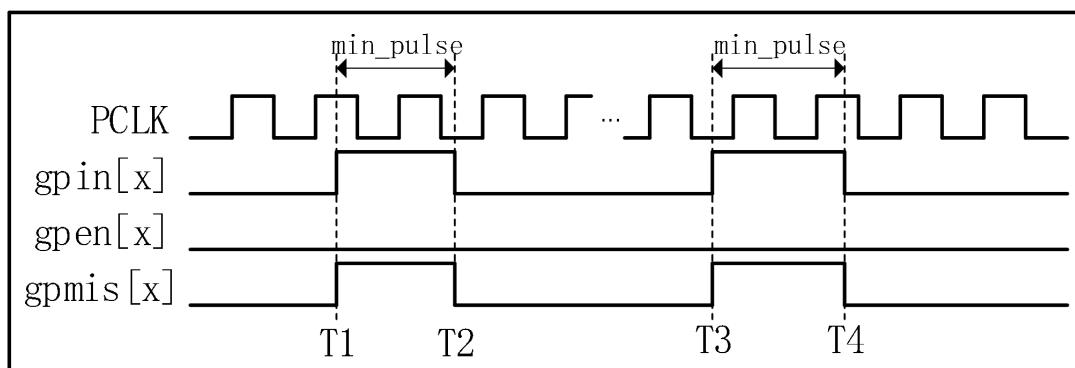
The valid interrupt level can be recognized by the port

Use the following registers to define the interrupt trigger mode and polarity:

- GPIOsInterrupt Trigger Mode Register (INTVLTRG), used to configure level-triggered or edge-triggered

- GPIOsInterrupt Trigger Polarity Register (INTRISEEN), for configuring level or edge triggered polarity
 - GPIOsInterrupt Edge Triggered Configuration Register (INTBE), after selecting as edge trigger, it is used to configure single edge trigger or Dual edge trigger
- passGPIOsInterrupt Enable Register (INTEN) can enable or disable the corresponding bit interrupt of the corresponding port,GPIOsOriginal start interrupt state (INTRAWSTAUS) are not affected by the enable bit. When an interrupt is generated, it can be GPIOsraw interrupt status (RAWINT STAUS) to get the status of the interrupt signal. When the interrupt enable register (INTEN) corresponding to 1 when, interrupt state(INTSTAUS) register can read the corresponding interrupt signal, and the interrupt signal will enter the interrupt configuration module and NVIC module, execute the interrupt routine.
- by writing 1 arrive GPIOsInterrupt Clear Register (INTCLR) to specify a bit to clear the corresponding bit interrupt.

1, High level trigger interrupt timing



picture5-24 GPIOsHigh level interrupt timing diagram

The time parameters in the figure are explained as follows:

surface5-8 GPIOsHigh level trigger interrupt timing time parameters

Parameter	describe	min
min_PLUS	Interrupt signals that the interrupt module can handle INTRAWSTAUS The minimum pulse width of	>=1pcclk

set up GPIOs module corresponding GPIOs Bits are input pins, high-level detection and enable interrupts. when T1 time, gpin[x] input high level signal, INTSTAUS register INTSTAUS [x] will be set by the hardware 1, while outputting to the system INTSTAUS [x] interrupt signal. when T2 time, gpin[x] input low level signal, INTSTAUS register INTSTAUS [x] by hardware clear 0, the hardware clears the INTSTAUS [x] interrupt signal.

Note that since the interrupt signal INTSTATUS needs to be handled by the interrupt module in the system, so the gpin

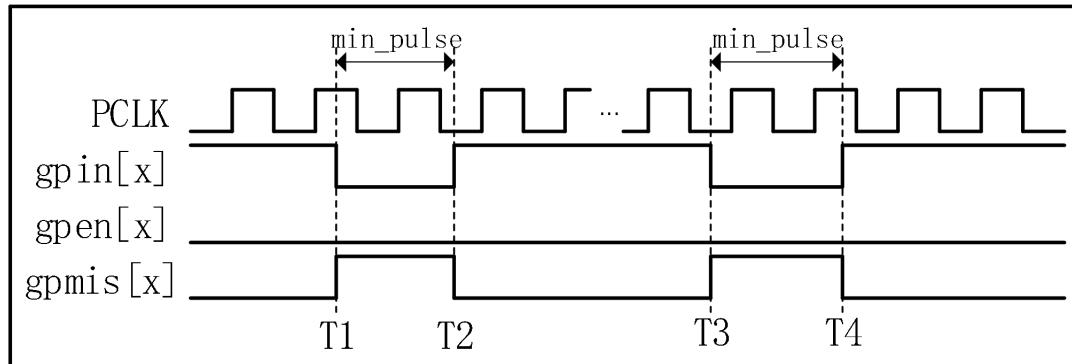
The high-level pulse width of the signal has a minimum limit:min_PLUSWidth requires at least one pclk clock cycle. if high

If the interrupt is masked when the level appears, the interrupt will be ignored and no corresponding interrupt flag signal will be generated.INTSTATUS. Other

Besides, due to GPIOsThe module does not latch the interrupt triggered by the high level, if the interrupt signal generated by the high level interrupt event is in

If the interrupt module can recognize that the high level disappears before, the interrupt flag will be ignored.

2, Low level trigger interrupt timing



picture5-25 GPIOsTiming diagram of low level trigger interrupt

The time parameters in the figure are explained as follows:

surface5-9 GPIOsLow level trigger interrupt timing time parameters

Parameter	describe	min
min_PLUS	Interrupt signals that the interrupt module can handleINTSTATUSThe minimum pulse width of	> =1pclk

set upGPIOsmodule correspondingGPIOsBits are input pins, low-level detection and enable interrupts. whenT1time,gpin[x]

input low level signal,INTSTATUSregisterINTSTATUS [x]will be set by the hardware1, while outputting to the systemINTSTATUS[x]

interrupt signal. whenT2time,gpin[x]input high level signal,INTSTATUSregisterINTSTATUS [x]cleared by hardware0,

hardware clearINTSTATUS[x]interrupt signal.

Note that since the interrupt signal INTSTATUS needs to be handled by the interrupt module in the system, so the gpin

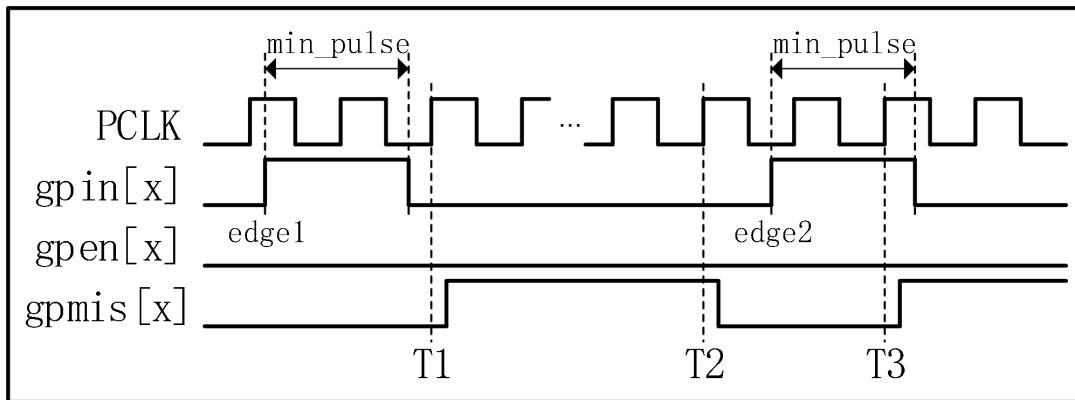
The high-level pulse width of the signal has a minimum limit:min_PLUSWidth requires at least one pclk clock cycle. if low

If the interrupt is masked when the level appears, the interrupt will be ignored and no corresponding interrupt flag signal will be generated.INTSTATUS. Other

Besides, due to GPIOsThe module does not latch the interrupt triggered by the low level, if the interrupt signal generated by the low level interrupt event is in

If the interrupt module can recognize that the low level disappears before, the interrupt flag will be ignored.

3, rising edge trigger interrupt timing



picture5-26 GPIOsRising edge trigger interrupt timing diagram

The time parameters in the figure are explained as follows:

surface5-10 GPIOsRising edge trigger timing time parameters

Parameter	describe	min
min_PLUS	Interrupt signals that the interrupt module can handleINTSTAUSThe minimum pulse width of gpin[x]	> =1pclk

set upGPIOsmodule correspondingGPIOsBits are input pins, rising edge detection and interrupt enable.edge1time,gpin[x]

Input a rising edge, the hardware detects the rising edge and processes it, the distanceedge1Rising edge at least2individualpclkAfter a cycle, the output

INTSTAUS[x]interrupt signal, seeT1time. existT2The time system clears the interrupt by software. existedge2time,

gpin[x]Another rising edge is input, the hardware detects the rising edge and processes it, and the distanceedge2Rising edge at least2individualpclkafter cycle,

output byedge2moment the rising edge event generated by theINTSTAUS[x]interrupt signal, seeT3time.

It should be noted that due toGPIOsThe module needs to detect the rising edge, so thegpinSignal high level pulse width
Degree has a minimum limit:min_PLUSWidth requires at least onepclkclock cycle to ensureGPIOsmodule can detect rising

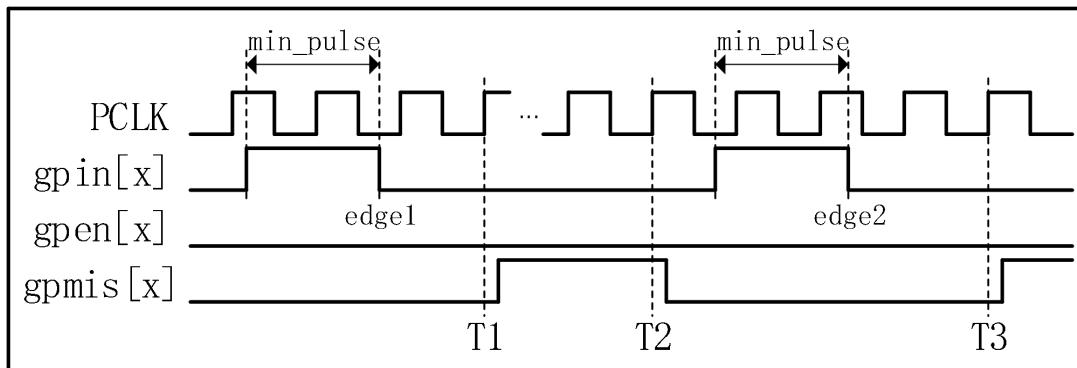
along. If the interrupt is masked when the rising edge occurs, the interrupt will be ignored and no corresponding interrupt flag signal will be generated

INTSTAUS. In addition due to theT2moment to clear the current interrupt, so theT2-2*TpclkNo new ones can appear before time

interrupt event (rising edge), otherwise a new interrupt event generates the interrupt flag signalINTSTAUSexistT2time will also be the same

When cleared, the new interrupt flag signal will be lost.

4, falling edge trigger interrupt timing



picture5-27 GPIOsTiming diagram of falling edge trigger interrupt

The time parameters in the figure are explained as follows:

surface5-11 GPIOsFalling edge trigger interrupt timing time parameters

Parameter	describe	min
min_PLUS	Interrupt signals that the interrupt module can handleINTSTAUSThe minimum pulse width of	> =1pclk

set upGPIOsmodule correspondingGPIOsBits are input pins, falling edge detection and interrupt enable.edge1time,gpin[x]lose

Input a falling edge, the hardware detects the falling edge and processes it, the distanceedge1Rising edge at least2individualpclkAfter a cycle, the outputINTSTAUS[x]

interrupt signal, seeT1time. existT2The time system clears the interrupt by software. existedge2time,gpin[x]Enter again

A falling edge, the hardware detects the falling edge and processes it, the distanceedge2Rising edge at least2individualpclkcycle, the output consists ofedge2

The moment the falling edge event is generated by theINTSTAUS[x]interrupt signal, seeT3time.

It should be noted that due toGPIOsThe module needs to detect the falling edge, so thegpinThe low-level pulse width of the signal has

Minimum limit:min_PLUSWidth requires at least onepclkclock cycle to ensureGPIOsThe module can detect the falling edge. if

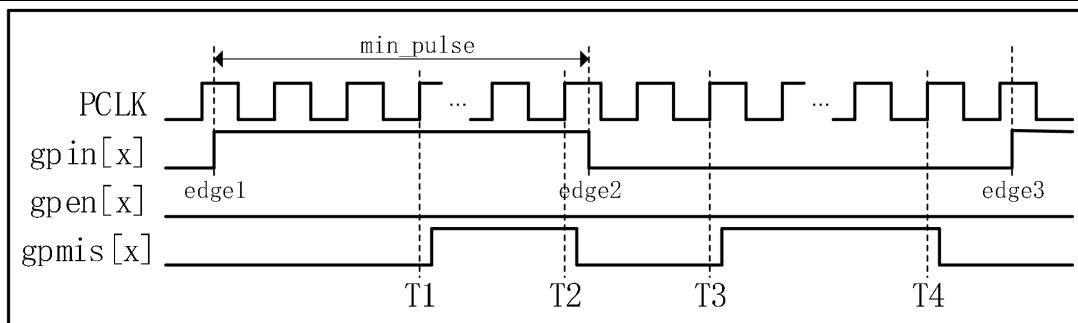
When the interrupt is masked when the falling edge occurs, the interrupt will be ignored and no corresponding interrupt flag signal will be generatedINTSTAUS. Other

external due toT2moment to clear the current interrupt, so theT2-2*TpclkNo new interrupt event (falling edge) can occur before time,

Otherwise a new interrupt event generates the interrupt flag signalINTSTAUSexistT2Time will also be cleared at the same time, the new interrupt flag signal

number will be lost.

5, Double-edge trigger interrupt timing



picture5-28 GPIOsTiming diagram of double-edge trigger interrupt

set upGPIOsmodule correspondingGPIOsBits are input pins, double-edge detection and enable interrupts.

edge1time,gpin[x]Input a rising edge, the hardware detects the rising edge and processes it, the distanceedge1Rising edge at least

2individualpclkafter the cycle, atT1time outputINTSTAUS[x]Interrupt flag signal. system inT2Instant software cleanupedge1

The interrupt is generated by the rising edge of time.

edge2time,gpin[x]Input a falling edge, the hardware detects the falling edge and processes it, the distanceedge2falling edge at least

2individualpclkafter the cycle, atT3time outputINTSTAUS[x]Interrupt flag signal. system inT4Instant software cleanupedge2

The interrupt is generated by the falling edge of time.

existedge3time,gpin[x]Another rising edge is input,GPIOsThe module processes the rising edge in the same way as above

Line processing and interrupt generation.

It should be noted that if the interrupt is masked when the interrupt event (rising edge or falling edge) occurs, the interrupt will be

Ignored, no corresponding interrupt flag signal will be generatedINTSTAUS. Also, as shown in the timing diagram, since theT2time is clear

removeedge1generated interrupt, so theT2-2*TpclkA new interrupt event (falling edge) cannot occur before time, otherwise

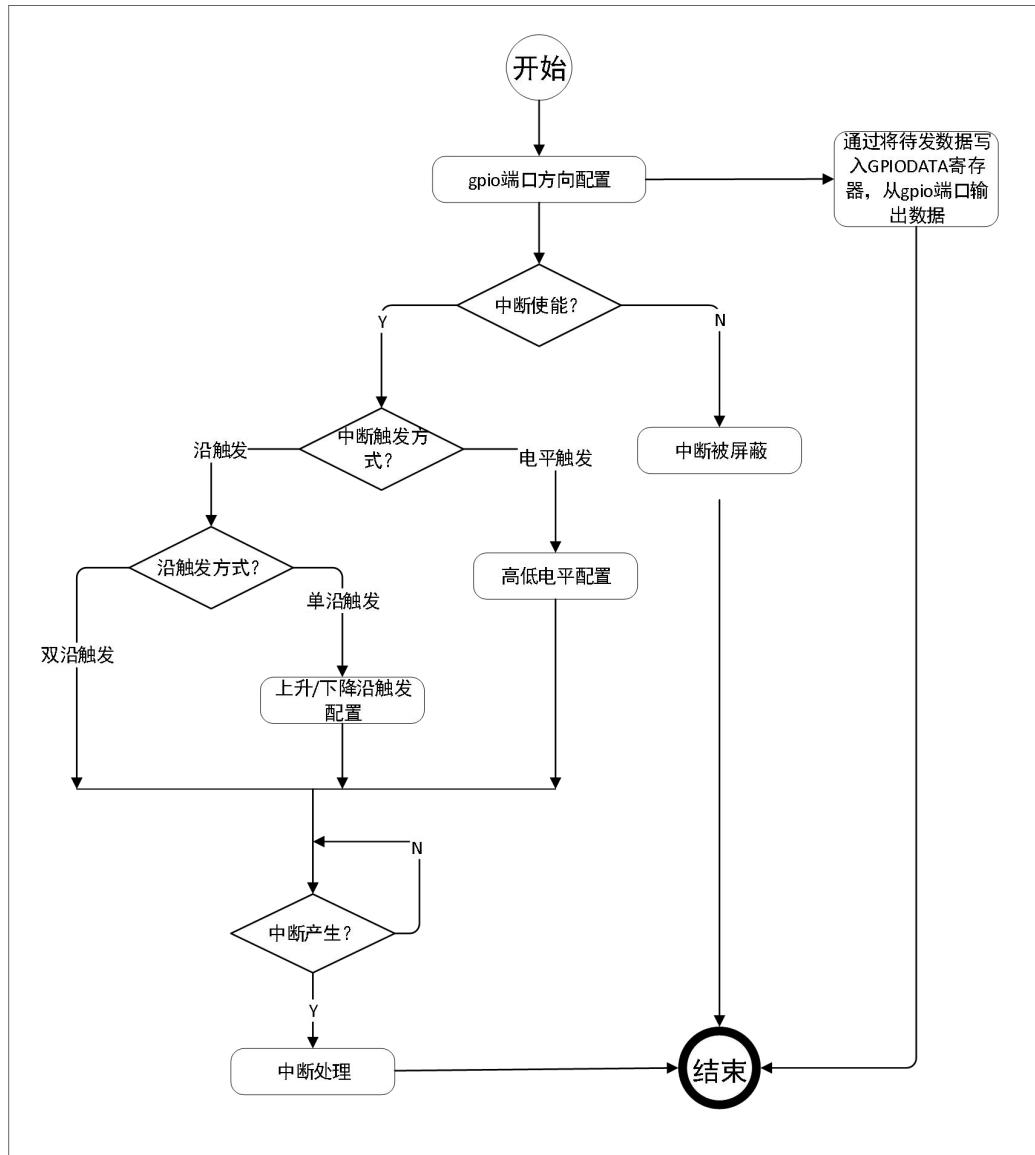
Interrupt flag signal generated by a new interrupt eventINTSTAUSexistT2Time will also be cleared at the same time, the new interrupt flag signal

number will be lost. foredge2event generated by the interrupt flag signal, theT4time is cleared, so inT4-2*Tpclk

A new interrupt event (rising edge) cannot occur before the time, otherwise the interrupt flag signal generated by the new interrupt eventINTSTAUS

existT4Time will also be cleared at the same time, the new interrupt flag signal will be lost.

Operating procedures



picture5-29 GPIOsOperation flow chart

- GPIOsModule clock enable
- portThe port is configured asGPIOsFunction
- Configure the port direction (GPIODIR)register
- If the port is configured as an output, the data to be sent is written to theGPIODATAreregister, fromgpioNumber of port outputs according to
- If the port is configured as an input, configure the interrupt enable and interrupt trigger mode, wait for the interrupt and handle the interrupt

register mapGPIODATAreregister(0x00)

bit field	name	type	reset value	describe
31:GPIO_WIDTH	RESERVED	RO	0	reserved bit
GPIO_WIDTH	GPIODATA	R/W	0	data register

GPIODIRregister(0x04)

bit field	name	type	reset value	describe
31:GPIO_WIDTH	RESERVED	RO	0	reserved bit
GPIO_WIDTH	GPIODIR	R/W	0	set upGPIOsPin direction: 1: Set the corresponding bit of theGPIOsPins are output pins 0: Set the corresponding bit of theGPIOsThe pin is an input pin

INTVLTRGregister(0x08)

bit field	name	type	reset value	describe
31:GPIO_WIDTH	RESERVED	RO	0	reserved bit
GPIO_WIDTH	INTVLTRG	R/W	0	set upGPIOsPin interrupt sensitive conditions: 1: Set the corresponding bit of theGPIOspin for level detection 0 : Set the corresponding bit of theGPIOsedge detection

INTBregister(0x0C)

bit field	name	type	reset value	describe
31:GPIO_WIDTH	RESERVED	RO	0	reserved bit



GPIO_WIDTH	INTBE	R/W	0	<p>set upGPIOsPin edge trigger mode:</p> <p>1: Set the corresponding bit of theGPIOsThe pin is double-edge triggered Interrupt, that is, both rising and falling edges are triggered broken</p> <p>0: Set the corresponding bit of theGPIOsThe pin is single-edge triggered interrupted byINTRISEENThe corresponding bit of the register is correct Must be rising edge/falling edge trigger</p>
------------	-------	-----	---	--

INTRISEENregister(0x10)

bit field	name	type	reset value	describe
31:GPIO_WIDTH	RESERVED	RO	0	reserved bit
GPIO_WIDTH	INTRISEEN	R/W	0	<p>set upGPIOsPin interrupt event mode:</p> <p>1: Set the corresponding bit of theGPIOspin is rising edge/high level triggered interrupt</p> <p>0: Set the corresponding bit of theGPIOspin is falling edge/low level triggered interrupt</p>

INTENregister(0x14)

bit field	name	type	reset value	describe
31:GPIO_WIDTH	RESERVED	RO	0	reserved bit
GPIO_WIDTH	INTEN	R/W	0	<p>set upGPIOsPin interrupt enable:</p> <p>1: Set the corresponding bit of theGPIOspin interrupt enable 0: Set the corresponding bit of theGPIOspin interrupt disabled</p>

INTRAWSTATUSregister(0x18)

bit field	name	type	reset value	describe
31:GPIO_WIDTH	RESERVED	RO	0	reserved bit

GPIO_WIDTH	INTRAWSTAUS	R	0	<p>whenGPIOsWhen configured as an input mode, according to the setting</p> <p>The trigger condition generates an interrupt flag, regardless of the interrupt enable</p> <p>Register effects. Set by hardware, software to GPIOICWrite1clear.</p> <p>1: Indicates that the corresponding bit is detectedGPIOinterrupt trigger condition(raw, before mask)</p> <p>0: Indicates that the corresponding bit is not detectedGPIOto interrupt</p> <p>Triggering conditions</p>
------------	-------------	---	---	---

INTSTAUSregister(0x1C)

bit field	name	type	reset value	describe
31:GPIO_WIDTH	RESERVED	RO	0	reserved bit
GPIO_WIDTH	INTSTAUS	R	0	<p>whenGPIOis configured as an input mode, and the corresponding bits of the</p> <p>When the interrupt is enabled, it will be generated according to the set trigger condition</p> <p>interrupt flag. Set by hardware, software toGPIOICWrite1clear.</p> <p>1: Indicates that the corresponding bit is detectedGPIOspin generation</p> <p>of interrupts (after the mask)</p> <p>0: Indicates that the corresponding bit is not detectedGPIOspin</p> <p>generated interrupt</p>

INTCLRregister(0x20)

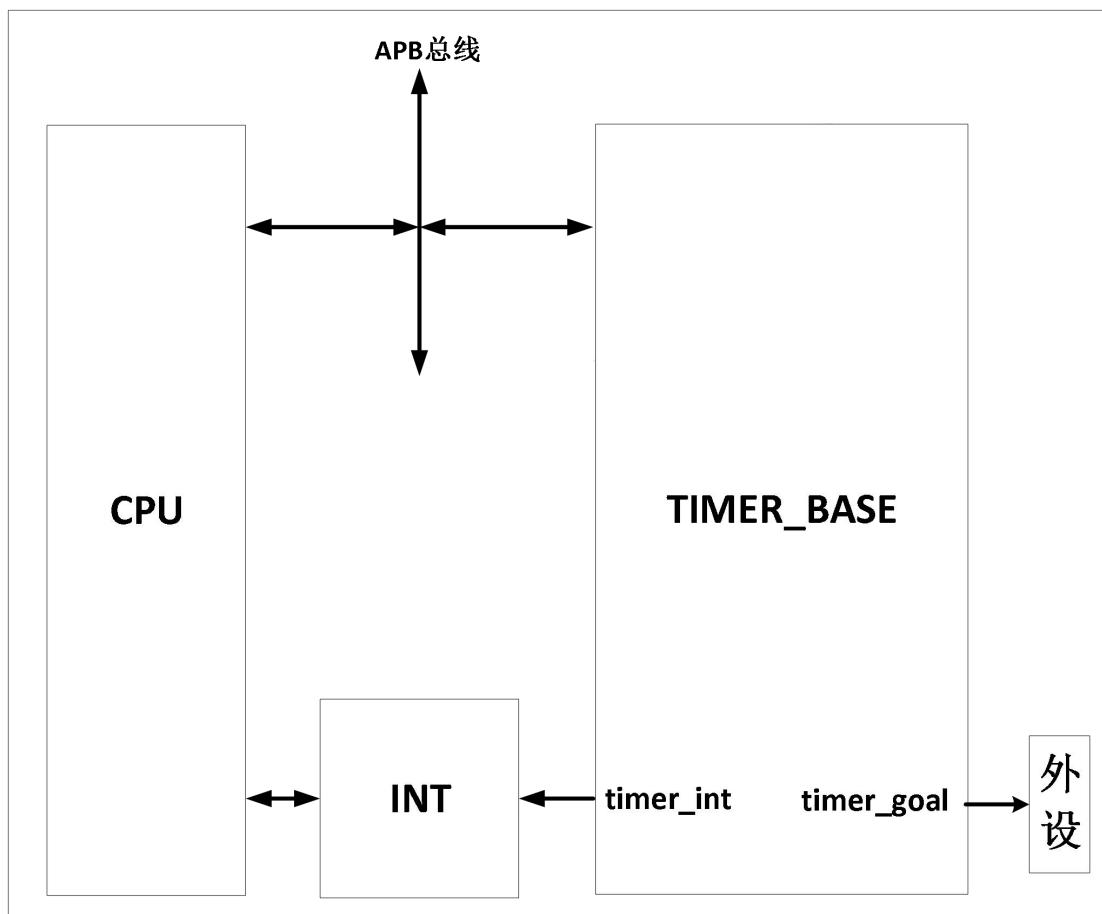
bit field	name	type	reset value	describe
31:GPIO_WIDTH	RESERVED	RO	0	reserved bit
GPIO_WIDTH	INTCLR	W	0	<p>Write1: Clear the corresponding bitGPIOPin edge triggered interrupt flag</p> <p>INTRAWSTAUSandINTSTAUS. After clearing the interrupt</p> <p>flag, theINTCLRCorresponding bit hardware auto recovery</p> <p>for0.</p> <p>Write0:No effect.</p> <p>Read the register, the return value is0.</p>

5.9 Basic Timer (TIMERBASE)

5.9.1 overview

The basic timer module has a timing function and has a 16Bit prescaler, supports interrupt, needs to be enabled before use

Timer module clock. The system block diagram of the basic timer module is as follows:



picture5-30 TIMERBASEModule System Block Diagram

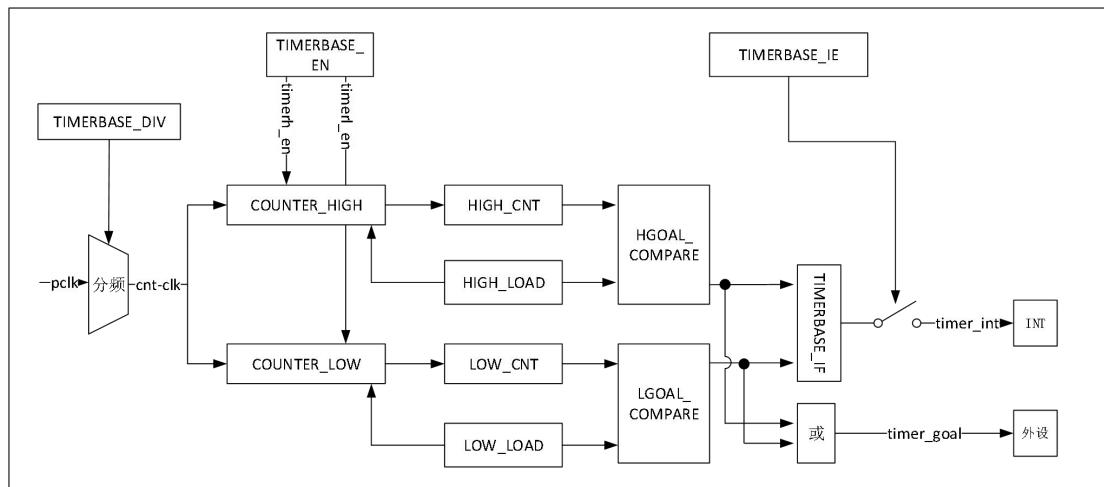
5.9.2 characteristic

-support 2 independent count-up 16bit counter (HIGH,LOW)

-support 16bit prescaler

-Can output interrupt and reach the target value flag signal

5.9.3 Block Diagram of Module Structure



picture5-31 TIMERBASE Block Diagram of Module Structure

Two independent high and low counters in this module share one 16bit Frequency division clock, frequency division value range 1-65536,

By frequency divider register TIMERBASE_DIV configuration to provide the count clock for the counter. sent by the high-low timer target

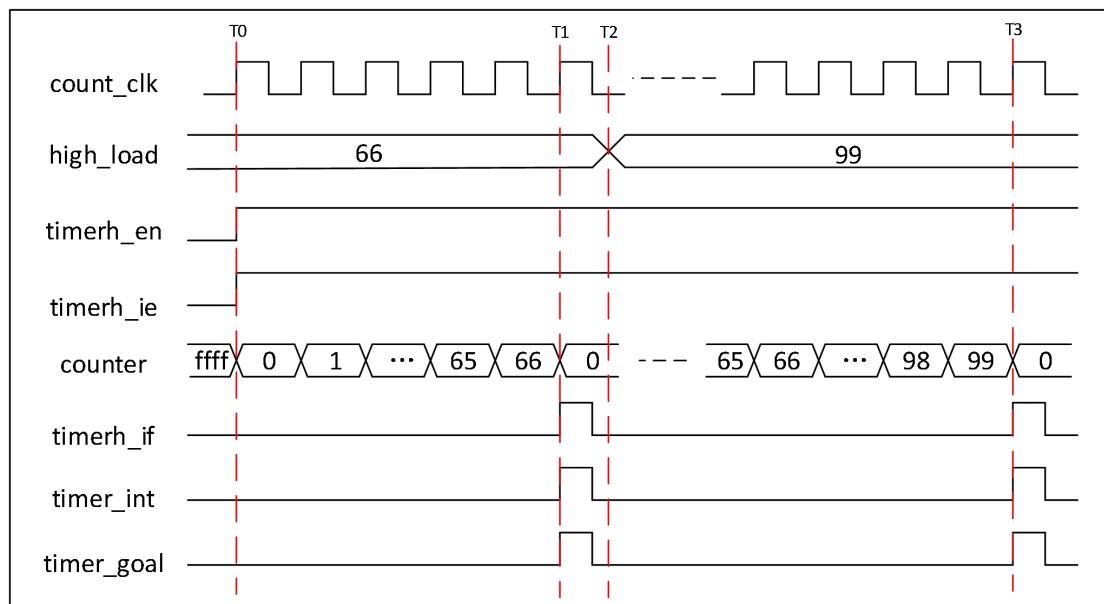
memory HIGH_LOAD/LOW_LOAD You can configure the counting target value for the high and low counters, and then turn on the counting enable to count

The counter starts counting, when the count value of the counter reaches the target value, a corresponding interrupt status will be generated, and the target value of the counter will be output

sign signal timer_goal, when the interrupt enable is turned on, the corresponding interrupt signal will also be generated timer_int.

5.9.4 Functional description

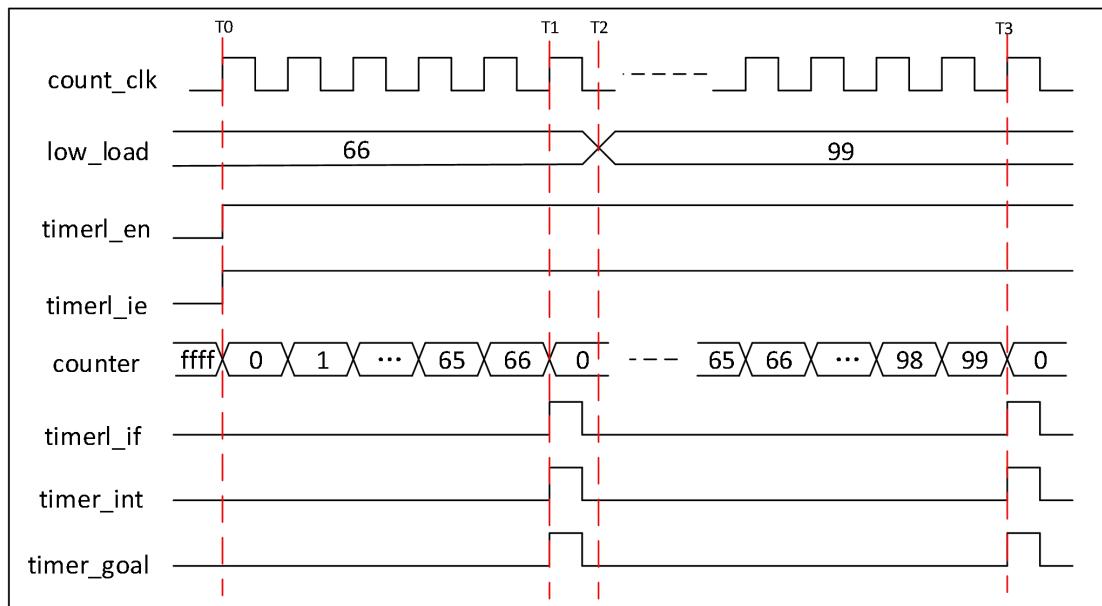
high counter count



picture5-32 TIMERBASEHigh Counter Count Timing Diagram

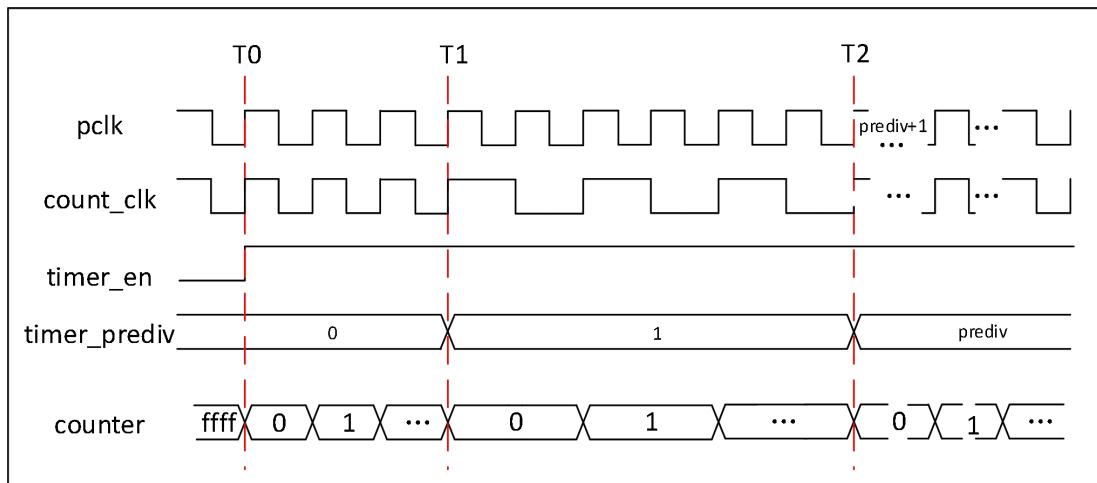
As shown in the figure above, it is the counting sequence diagram of the high-bit counter. After configuring the counting target value, the T0 always open high count register count enable and interrupt enable, the T1 After the time counter reaches the target value, a high counter interrupt status is generated, and the interrupt signal number and reached the target value flag signal. T2 Change the target value at any time, the counter will count to the new target value and generate a high Counter interrupt status, interrupt signal and target value flag signal. The clock source in this module is pclk, same as the system clock source, available through SYSCON in the module DEV_CLK_GATE Register to configure the enable of the clock of this module.

low counter count



picture5-33 TIMERBASELow Counter Count Timing Diagram

As shown in the figure above, it is the counting sequence diagram of the low-bit counter. When the counting target value is configured, the T0 always open low count register count enable and interrupt enable, the T1 After the time counter reaches the target value, a low counter interrupt status is generated, and the interrupt signal number and reached the target value flag signal. T2 Change the target value at any time, the counter will count to the new target value and generate a low count Counter interrupt status, interrupt signal and target value flag signal.

Frequency division counting


picture5-34 TIMERBASETiming diagram of counter frequency division counting

As shown in the figure above, the counter is configuring the frequency division register **TIMERBASE_DIV** respectively 0, 1 and **prediv** time division

Counting timing diagram.

Timing duration calculation formula

Basic timer as two independent 16bitWhen it is a bit timer, it counts up, and the counting source is the system clock Sys.

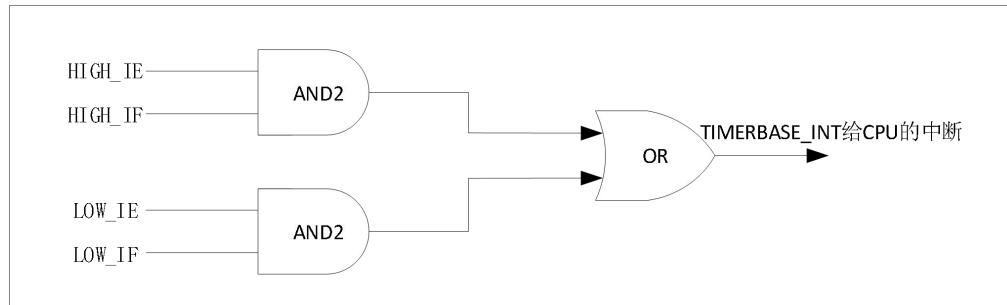
Timing duration T_{out} Calculated as follows:

$$T_{out} = (T_{pre} + 1) * (T_{load} + 1) / \text{Sys}$$

Note: T_{pre} is the frequency division coefficient, T_{load} is the height of the loading value 16bit or low 16bit, Sys for the system clock.

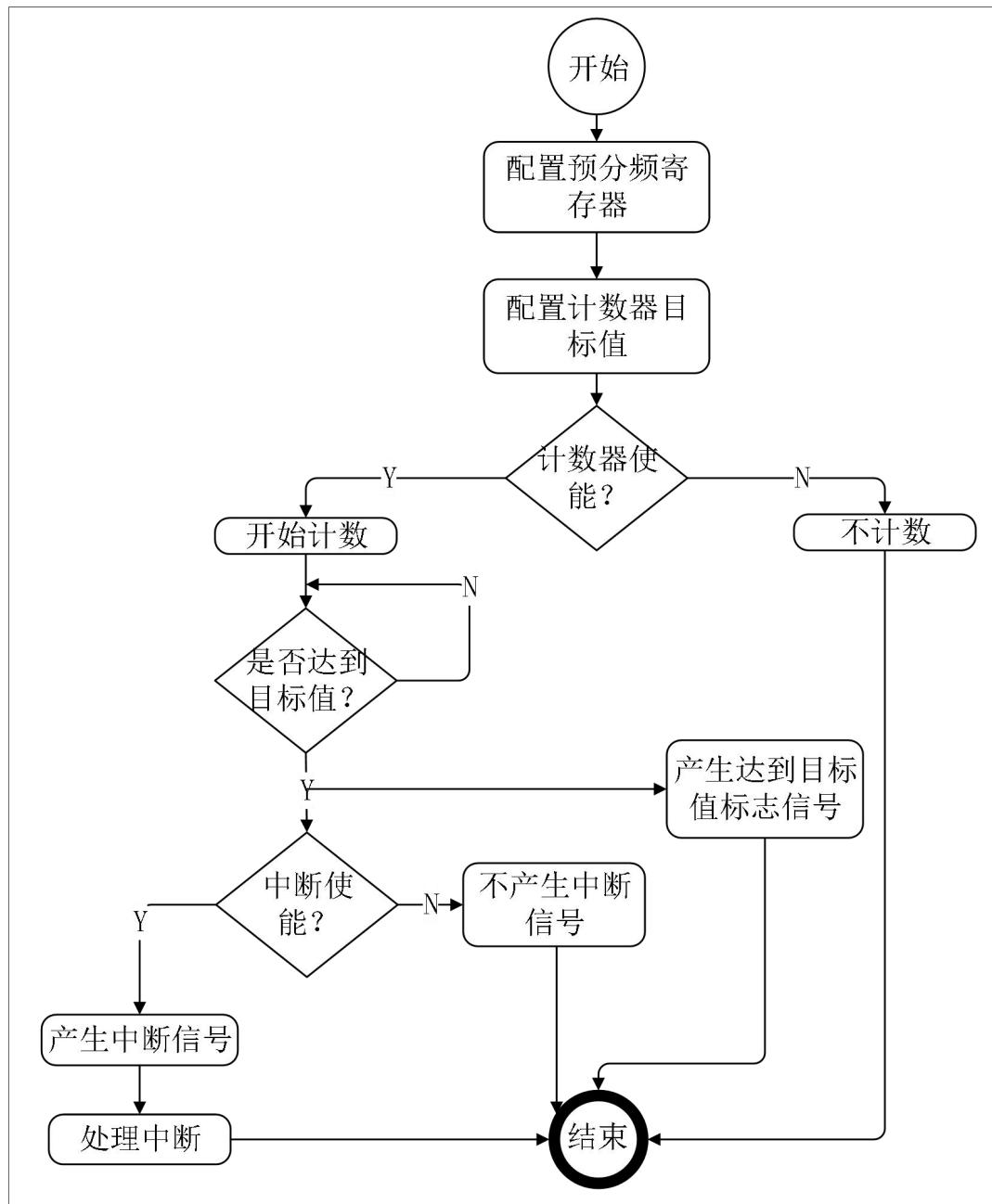
to interrupt

TIMERBASE provided 2 There are two kinds of interrupt sources, and their relationship is shown in the figure below:



picture5-35 TIMERBASE interrupt flag and interrupt diagram

Operating procedures



picture5-36 TIMERBASEOperating procedures

- Through the prescaler register (TIMERBASE_DIV) to set the prescaler target value (1-65536), for the system clock

Carry out frequency division.

- By loading the value register (HIGH_LOAD or LOW_LOAD) to set the count target value (16bit) . here divided into high16bit and low16bit. There are two timers, which can be configured according to the corresponding needs.
- Through the interrupt enable register (TIMERBASE_IE) to configure the interrupt enable.
- Through the enable register (TIMERBASE_EN) to enable the corresponding timer.
- The corresponding timer starts from 0. Count up, when the count reaches the load value, an interrupt is generated, and at the same time restart from 0. Start counting and enter the counting of the next cycle.
- Interrupts can be accessed through the Interrupt Status Register (TIMERBASE_IF) to query (when the interrupt is enabled). Registers can also be written to. The operation clears the interrupt status.
- During the counting process, the current count value register (HIGH_CNT or LOW_CNT) to read. Fetch, get the current count value.

register map

name	Offset	bit width	type	reset value	describe
TIMERBASE0	BASE: 0x40064000				
TIMERBASE1	BASE: 0x40064800				
TIMERBASE_EN	0x00	32	R/W	0x00	TIMERenable register
TIMERBASE_DIV	0x04	32	R/W	0x00	TIMERCount Clock Divider Register
TIMERBASE_IE	0x10	32	R/W	0x00	TIMERinterrupt enable register
TIMERBASE_IF	0x14	32	R/W	0x00	TIMERInterrupt Status Register
HIGH_LOAD	0x20	32	R/W	0xffff	TIMER HIGHtarget configuration register
HIGH_CNT	0x24	32	R	0x00	TIMER HIGHCurrent count value register
LOW_LOAD	0x30	32	R/W	0xffff	TIMER LOWtarget configuration register
LOW_CNT	0x34	32	R	0x00	TIMER LOWCurrent count value register

**Register description****TIMERBASE_ENregister(0x00)**

bit field	name	type	reset value	describe
31:2	RESERVED	R	0	reserve
1	HIGH_EN	R/W	0	TIMERBASE HIGHTimer Enable Register
0	LOW_EN	R/W	0	TIMERBASE LOWTimer Enable Register

TIMERBASE_DIVregister(0x04)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserve
15:0	DIV	R/W	0	TIMERBASECount Clock Prescaler Register 0x0000:express1frequency division 0x0001:express2frequency division ... 0xFFFF:express65536frequency division

TIMERBASE_IERegister(0x10)

bit field	name	type	reset value	describe
31:2	RESERVED	R	0	reserved bit
1	HIGH_IE	R/W	0	TIMERBASE HIGHTimer interrupt enable
0	LOW_IE	R/W	0	TIMERBASE LOWTimer interrupt enable

TIMERBASE_IFregister(0x14)

bit field	name	type	reset value	describe
31:2	RESERVED	R	0	reserved bit
1	HIGH_IF	R/W	0	TIMERBASE HIGHTimer Interrupt Status Write1clear
0	LOW_IF	R/W	0	TIMERBASE LOWTimer Interrupt Status Write1clear

HIGH_LOADregister(0x20)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	HIGH_LOAD	R/W	0xffff	TIMERBASE HIGHTimer Target Configuration Register when high16bitAfter the counter counts up and reaches the set value, it will generate a corresponding status signal

HIGH_CNTregister(0x24)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	HIGH_CNT	R	0	TIMERBASE HIGHThe current count value of the timer

LOW_LOADregister(0x30)

bit field	name	type	reset value	describe

31:16	RESERVED	R	0	reserved bit
15:0	LOW_LOAD	R/W	0xffff	TIMERBASE LOWThe timer target configuration register is low when the16bitAfter the counter counts up and reaches the set value, it will generate a corresponding status signal

LOW_CNTregister(0x34)

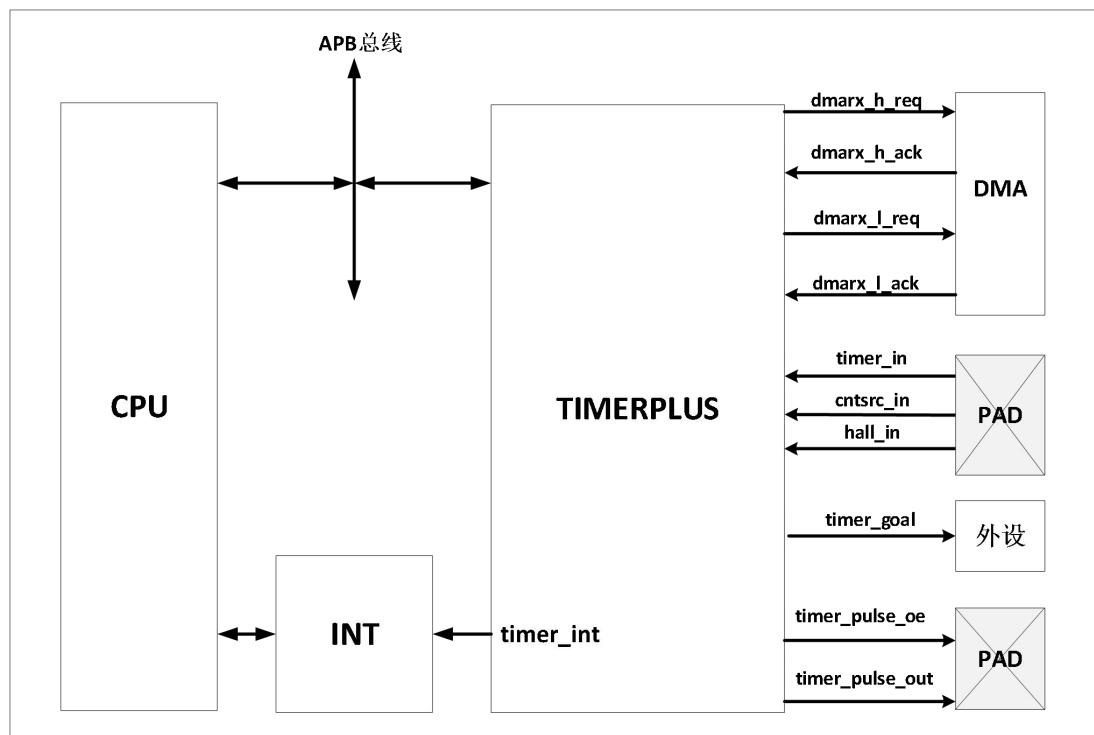
bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	LOW_CNT	R	0	TIMERBASE LOWThe current count value of the timer

5.10 Advanced Timer (TIMERPLUS)

5.10.1 overview

The advanced timer module has functions such as timing, counting, capture, and periodic pulse output, and has a 16bit prescaler device, supports interrupts, 2an independent 16bit timer (HIGH and LOW), where low 16The bit timer also supports HALL function, the advanced timer module clock needs to be enabled before use.

The system block diagram of the advanced timer module is shown in the following figure:



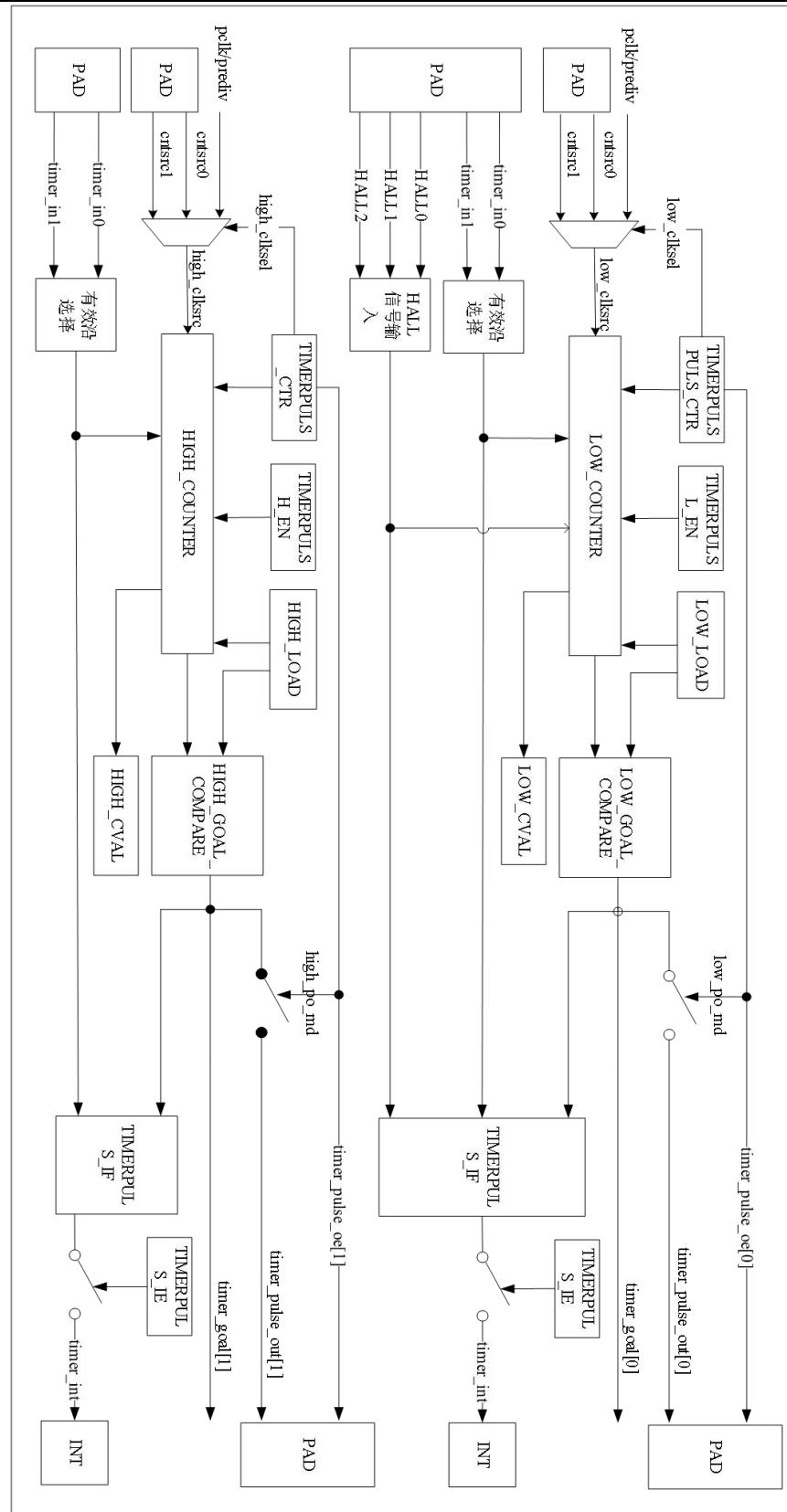
picture5-37 TIMERPLUS System Block Diagram

As shown in the figure above, in this chip TIMERPLUS module through APB bus with CPU complete correspondence and support with DMA communication, can generate a variety of interrupt signals, and can also output periodic pulse signals and periodic pulse signal output enable outputs to PAD Port, which transmits the counter counting to the target value flag signal to other peripherals; at the same time, TIMERPLUS can also pass Pass PAD The port inputs the off-chip timer_in, cntsrc_in, hall_in Wait for the signal.

5.10.2 characteristic

- 2an independent16bitcounter(HIGHandLOW)
- have16bitPrescaler Counter
- Two timers have functions of timing, counting, input capture and periodic pulse output respectively
- LOWcounter supportHALLFunction
- Separate support for count clock selection
- Support multiple interrupt functions
- The two timers have their own target value shadow registers, and the new target value will take effect in the next cycle
- supportDMAInterface to read input pulse capture value

5.10.3 Block Diagram of Module Structure



picture5-38 TIMERPLUSStructure diagram

Pictured above is TIMERPLUS. The structure diagram of the module, this module contains two independent 16-bit counter, two counters. Counting clock can choose internal prescaler clock or off-chip counting clock cntsrc, where the internal prescaler clock can be passed pclkClock Configuration Prescaler Register TIMERPLUS_PREDIV of PCLK_PREDIV bits for frequency division, frequency division value range 1-65536, off-chip count clock cntsrc is through PAD port input. The clock source of this module is pclk, with the system

The clock source is the same, which can be passed through SCS CON in the module DEV_CLK_GATE Register to configure the enable of the clock of this module.

Both the high-order counter and the low-order counter support timing functions, counting functions and input capture functions, and the low-order counter also supports holdHALL function. The function of the high and low counters and related configuration can be set through the register TIMERPLUS_CTR to configure.

The interrupt signals of different states in this module are controlled by the interrupt enable register TIMERPLUS_IEn control, only in the corresponding When the interrupt enable is enabled, the interrupt signal will not be generated until the corresponding interrupt status is generated.

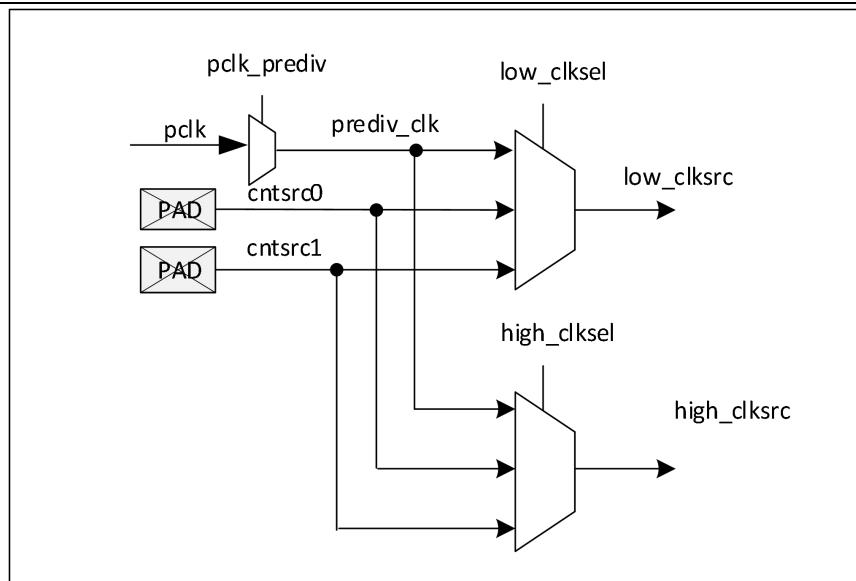
This module also supports DMA. The read of the input capture, in input capture mode, when DMA read enable as 1 hour, DMA will replace CPU read the capture value from the counter capture value register.

generated by this module timer_goal[1:0] Two signals are available as SAR ADC trigger source for the module.

5.10.4 Functional description

Counting clock source selection and prescaler

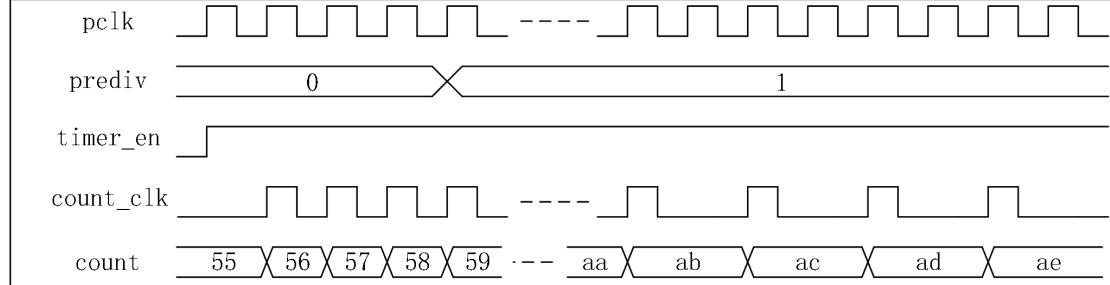
The two internal timers in this module can respectively select the counting clock source and share a prescaler. counter. The schematic diagram of counting clock source selection is as follows, in which the off-chip clock cntsrc0 and cntsrc1 is through PAD port input, it is associated with the off-chip input signal timer_in share the same PAD port at IO Select in function configuration PAD Port is timer_in hour, PAD port can also be used as an external clock cntsrc0 and cntsrc1 input port.



picture5-39 TIMERPLUS Schematic diagram of clock source selection

The prescaler is only used for pclk, the prescaler register in the prescaler PCLK_PREDIV can be controlled by bit pclkclock in Row frequency division, the frequency division value range is 1-65536, the divided clock and the off-chip clock cntsrc0 and cntsrc1 when passing the counter

The selection of the clock register provides the counting clock for the counter. The timing diagram under the divided clock is as follows:



picture5-40 TIMERPLUS pclkCounter Timing Diagram under Divided Clock

timing mode

The timing mode of this module means that the counting clock can be selected as the internal prescaler clock, off-chip cntsrc[0] and off-chip cntsrc[1]

In the case of three clock sources, the counter counts, and the timing time is configured through the counter target value configuration register.

After counting to the target value, the counter reaches the target value flag signal. Periodic pulse output to pin can be configured in timing mode

, which can be output as a simple square wave. During the working process of the timer, modifying the period value will not take effect immediately, but will

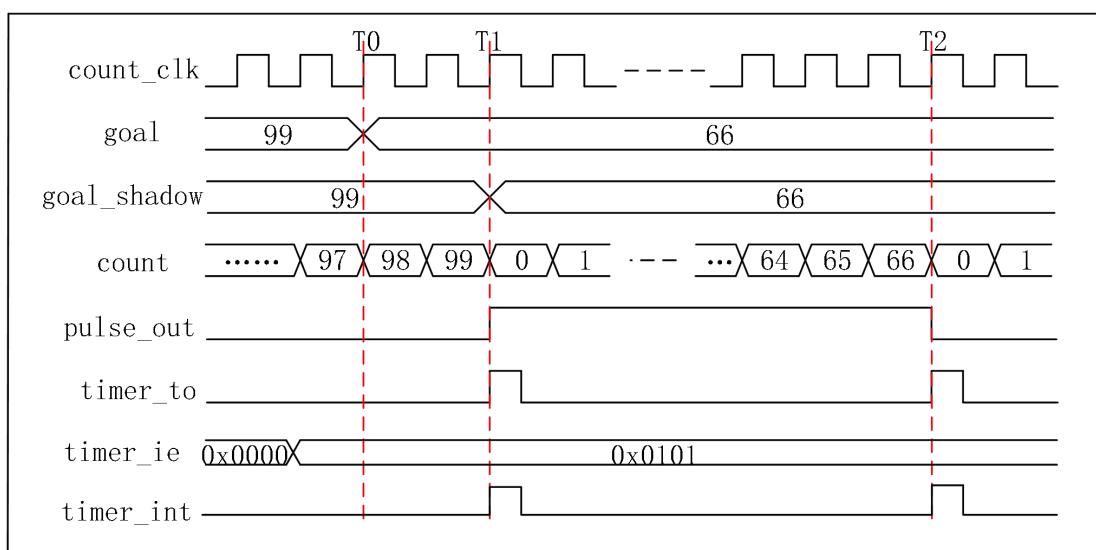
It takes effect in the next cycle after the end of the cycle.

advanced timer as two independent16When the bit timer is used, it counts up, and the counting clock source is the system clockpclk

For example, timing durationToutCalculated as follows:

$$T_{out} = (T_{pre} + 1) * (T_{load} + 1) / pclk.$$

Note:Tpreis the frequency division coefficient,T loadis the height of the loading value16bit or low16bit,pclkfor the system clock.



picture5-41 TIMERPLUSCounter Timing Diagram in Timing Mode

in,count_clkis the count clock of the counter,goalforCPUconfigured target value,goal_shadowfor the shadow register value,countis the count value,PLUS_outis the original value of periodic pulse output,timer_tofor the count value to reach the target Flag signal when value. existT0Changing the target value at all times, the target value of the shadow register does not change immediately, but is counted

The counter does not change until it reaches the previous target value (T1time),T1When the time reaches the target value, the periodic pulse output signal will be

The number is reversed, and the flag signal reaching the target value is generated, and the counter restarts from0Start counting.T2The time counter reaches again

After the new target value is reached, the flag signal reaching the target value will be generated again, the periodic pulse output signal will flip again, and the counter will re-counting times, and when the target value is reached, the interrupt enable is turned onT1,T2A target value reached interrupt signal will be generated.

TIMERPLUS_CTRRegister high-low counter period pulse output enable bitHIGH_PO_MD,LOW_PO_MDrespectively

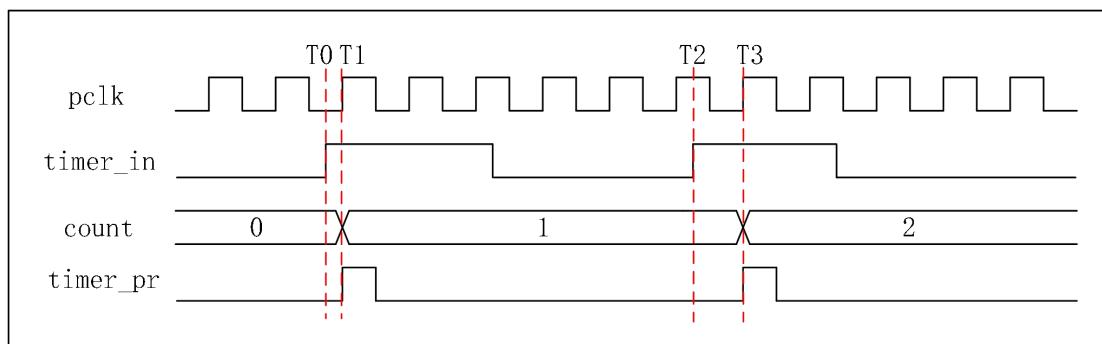


The control cycle pulse signal is different bit pair PAD output, when the enable is enabled, the periodic pulse signal will be output to the corresponding of PAD port.

counting mode

The counting mode of this module refers to the off-chiptimer_in[0] or off-chiptimer_in[1] Signal rising edge, falling edge or bilateral Counting along the edge, when the count reaches the target value, the counter reaches the target value flag signal. passpclkdetect rising edge, The counter counts on the falling edge or both edges, and the specific timing diagram is as follows:

Rising edge valid

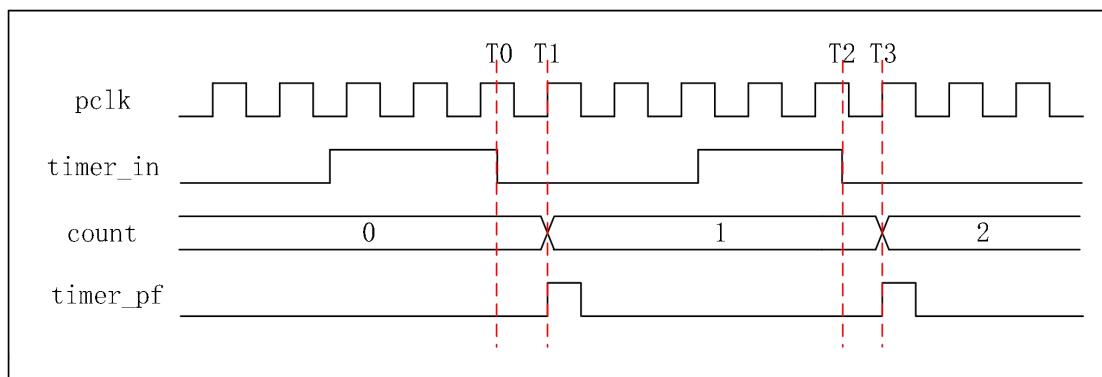


picture5-42 TIMERPLUSRising edge valid timing diagram in counting mode

where the count mode uses pclk is the count clock of the counter, timer_in is an off-chip input signal, countFor the sake of value, timer_pr not detected timer_in Rising edge flag signal. exist T0 time input signal timer_in generate rise edge, on the next rising edge of the count clock T1 The time counter is increased by one, and a rising edge flag signal is generated, T2 and T3 hour

The same is true for the moment.

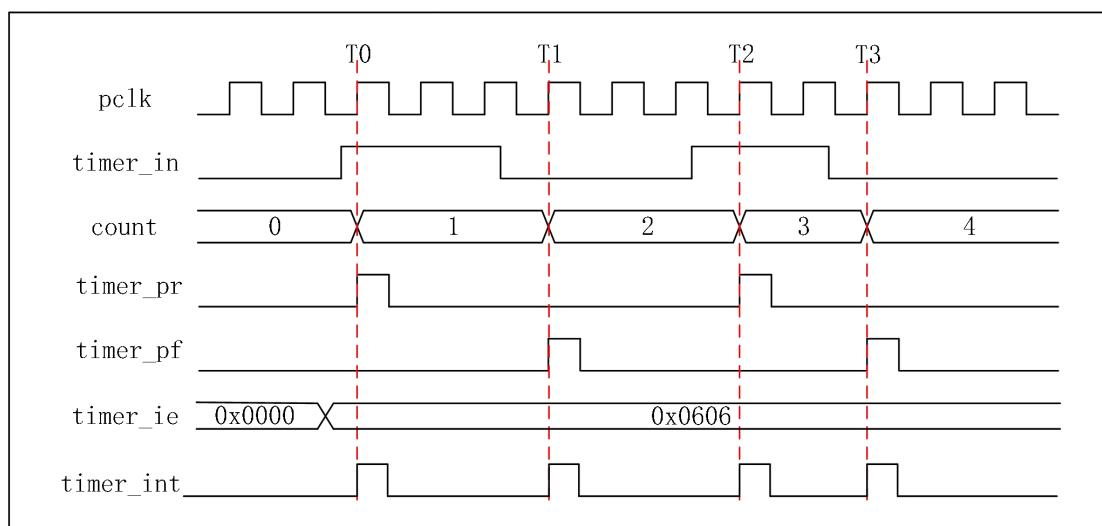
Falling edge valid



picture5-43 TIMERPLUSTiming diagram of falling edge valid in counting mode

inpclk is the count clock of the counter,**timer_in** is an off-chip input signal,**count** is the count value,**timer_pf** is not detected.
timer_in Falling edge flag signal. exist **T0** time input signal **timer_in** generates a falling edge at the count clock
the next rising edge of **T1** The time counter is increased by one, and a falling edge flag signal is generated, **T2** and **T3** The same is true of moments.

Dual edge valid



picture5-44 TIMERPLUSDouble-edge active timing diagram in counting mode

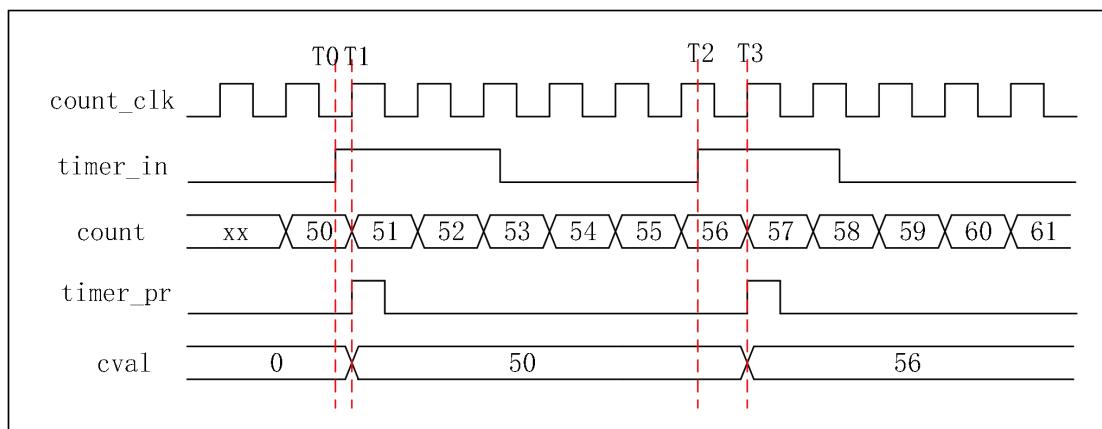
inpclk is the count clock of the counter, timer_in is an off-chip input signal, count is the count value, timer_pr not detected timer_inRising edge flag signal, timer_pf not detected timer_inFalling edge flag signal. Dual edge valid. The situation is the same as the counter counting when the rising edge and the falling edge are valid, that is, after the valid edge arrives, the count of the counter will be valid on the next rising edge of the count clock, and when the corresponding interrupt enable is enabled, a corresponding interrupt signal will be generated, such as in the picture above T0, T1, T2, T3 time.

input capture mode

The input capture mode means that the count clock can be selected as the internal prescaler clock, off-chipcntsrc[0] and off-chipcntsrc[1] three. In the case of two clock sources, the counter counts. When an off-chip timer_in[0] or off-chip timer_in[1] rising signal register through the edge or falling edge, it saves the current counter count value.

The timing diagram of different effective edges is as follows:

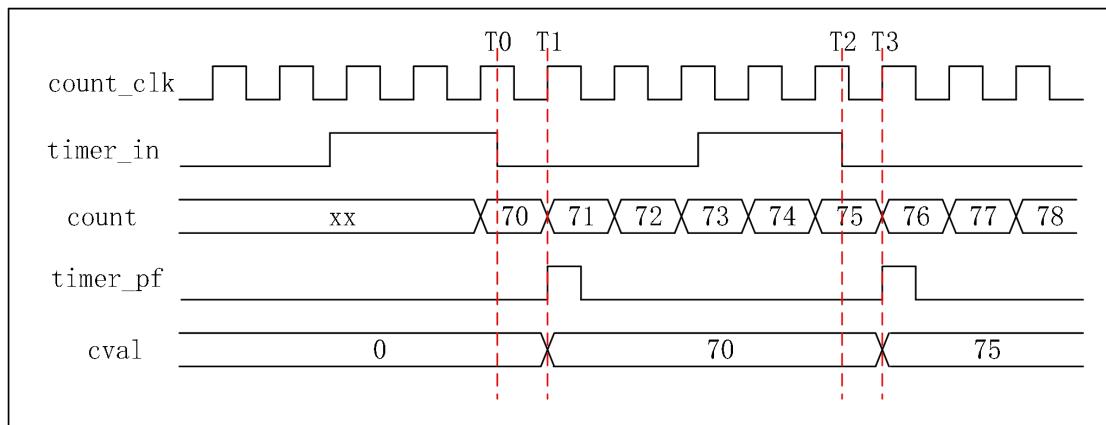
Rising edge valid



picture5-45 TIMERPLUSTiming diagram of valid rising edge under input capture

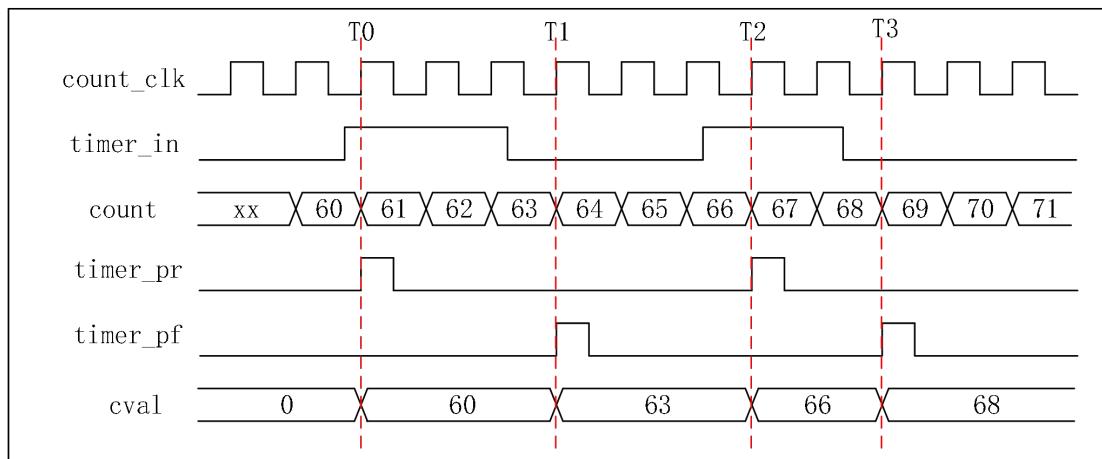
incount_clk is the count clock of the counter, timer_in is an off-chip input signal, count is the count value, timer_pf is not detected. timer_inRising edge flag signal, cval Counts the value for the captured counter. When the rising edge is valid, T0 time off-chip input signal timer_inThe rising edge comes, T1 Save the current count value of the counter to the register at all times CVAL, and generate a corresponding rising edge detected flag signal. T2 time off-chip input signal timer_inThe rising edge of coming in T3 The value of the moment counter will be saved to the register again CVAL.

Falling edge valid



picture5-46 TIMERPLUSInput Capture Falling Edge Valid Timing Diagram

incount_clk is the count clock of the counter, timer_in is an off-chip input signal, count is the count value, timer_pf is not detected. timer_inFalling edge flag signal, cval Counts the value for the captured counter. When the falling edge is active, the T0 time off-chip input signal timer_inThe falling edge comes, T1 Save the current count value of the counter to the register at all times cval, A corresponding falling edge detected flag signal is generated. T2 time off-chip input signal timer_inThe falling edge comes again, at T3 The value of the moment counter will be saved to the register again cval.

Dual edge valid


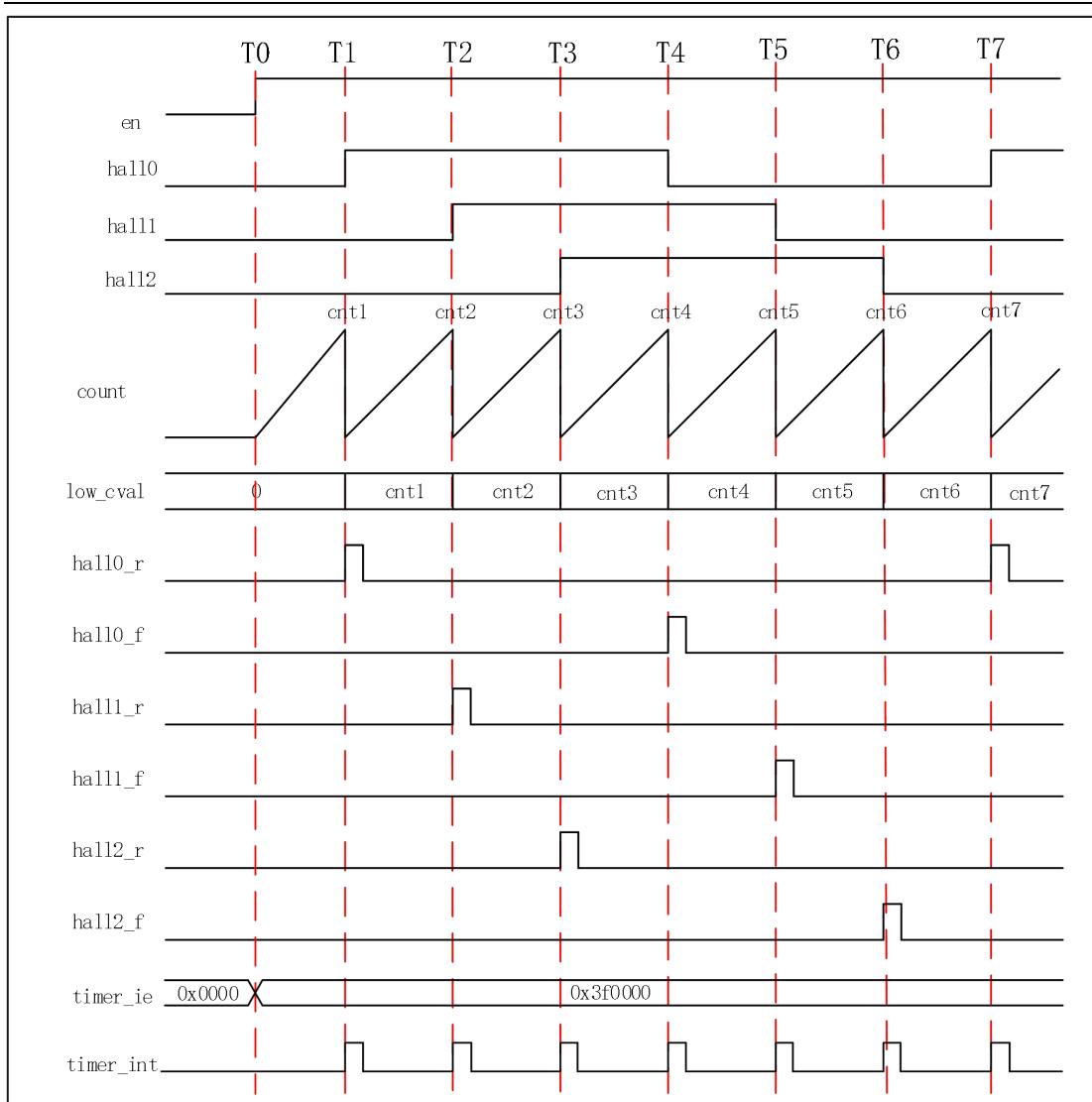
picture5-47 TIMERPLUSTiming diagram of double-edge active under input capture

incount_clkis the count clock of the counter,timer_inis an off-chip input signal,countis the count value,timer_pr
not detectedtimer_inRising edge flag signal,timer_pfnot detectedtimer_inFalling edge flag signal,cvalto catch

The obtained counter count value. In the case of double-edge valid, when the rising edge or falling edge arrives, the current count of the counter will be
save value to registercval, and generate the corresponding flag signal, as shown in the figureT0,T1,T2,T3time shown.

HALLmodel

In this module, onlyLOWtimer hasHALLMode function, can be used for Hall signal acquisition. This mode refers to the
The count clock can be selected as internal prescaler clock, off-chipcntsrc[0]and off-chipcntsrc[1]In the case of three clock sources, the counter
to count. When an off-chiphall_in[0], off-chiphall_in[1]and off-chiphall_in[2]The rising and falling edges of the signal
pass registercvalsaves the current counter count value and willLOWThe counter is cleared.

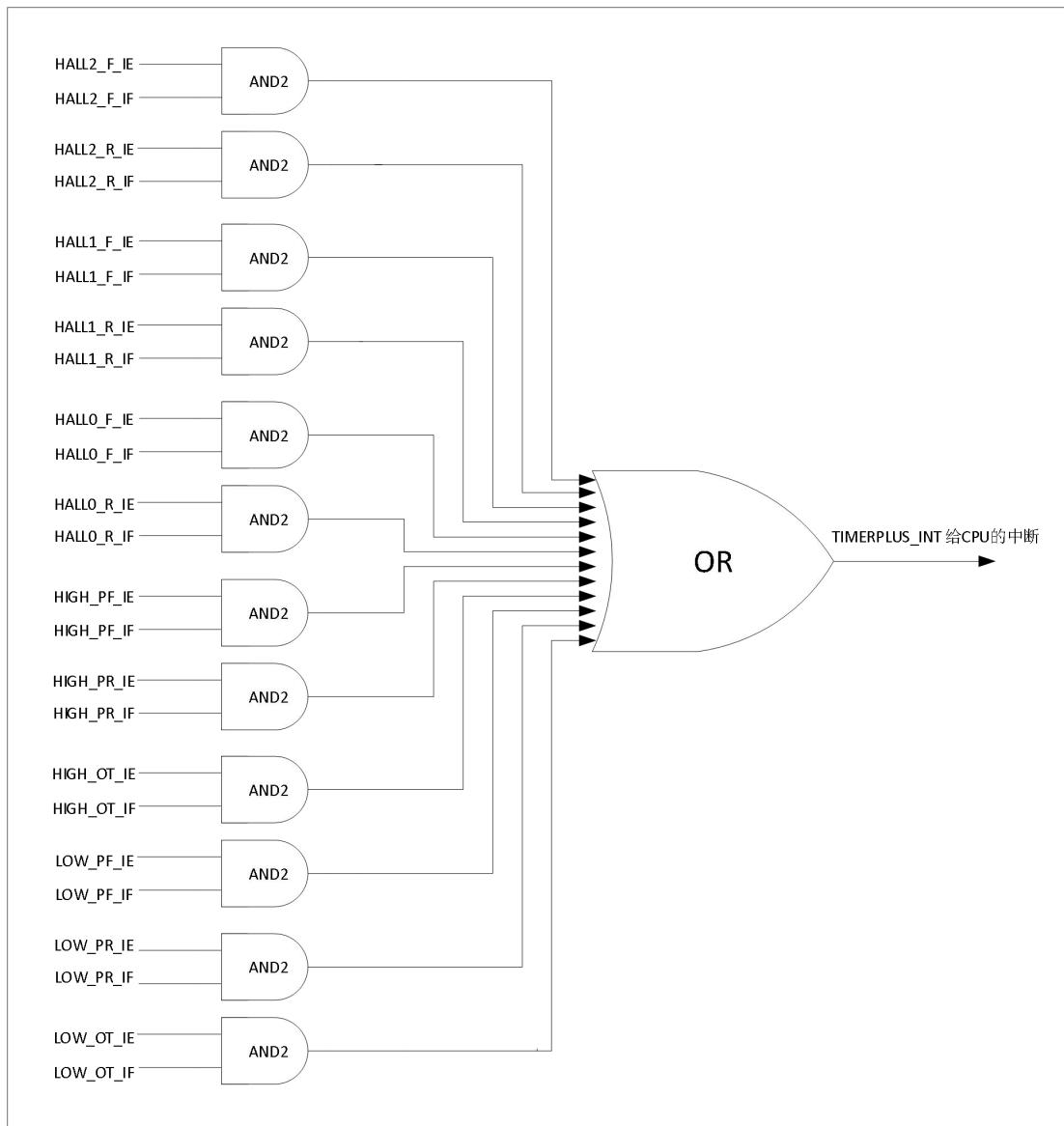


picture5-48 TIMERPLUSexistHALLMode Timing Diagram

in hall0, hall1 and hall2 for three input HALL Signal, count is the count value, hall0_r, hall1_r and hall2_r not detected HALL signal rising edge flag signal, hall0_f, hall1_f and hall2_f not detected HALL letter Sign signal falling edge, low_cval is the captured counter count value, timer_ie is the interrupt enable signal, timer_int for the generated interrupt signal. The circuit will clear the counter every time it detects a rising edge and a falling edge, and will clear the previous count save the value to low_cval register, while generating each HALL The rising or falling edge of the signal marks the signal. when open When the corresponding interrupt is enabled, the corresponding interrupt signal will also be generated after the corresponding status flag signal is generated, as shown in the figure above T1-T7 time shown.

to interrupt

TIMERPLUS provided 12 interrupt sources, the schematic diagram is as follows:

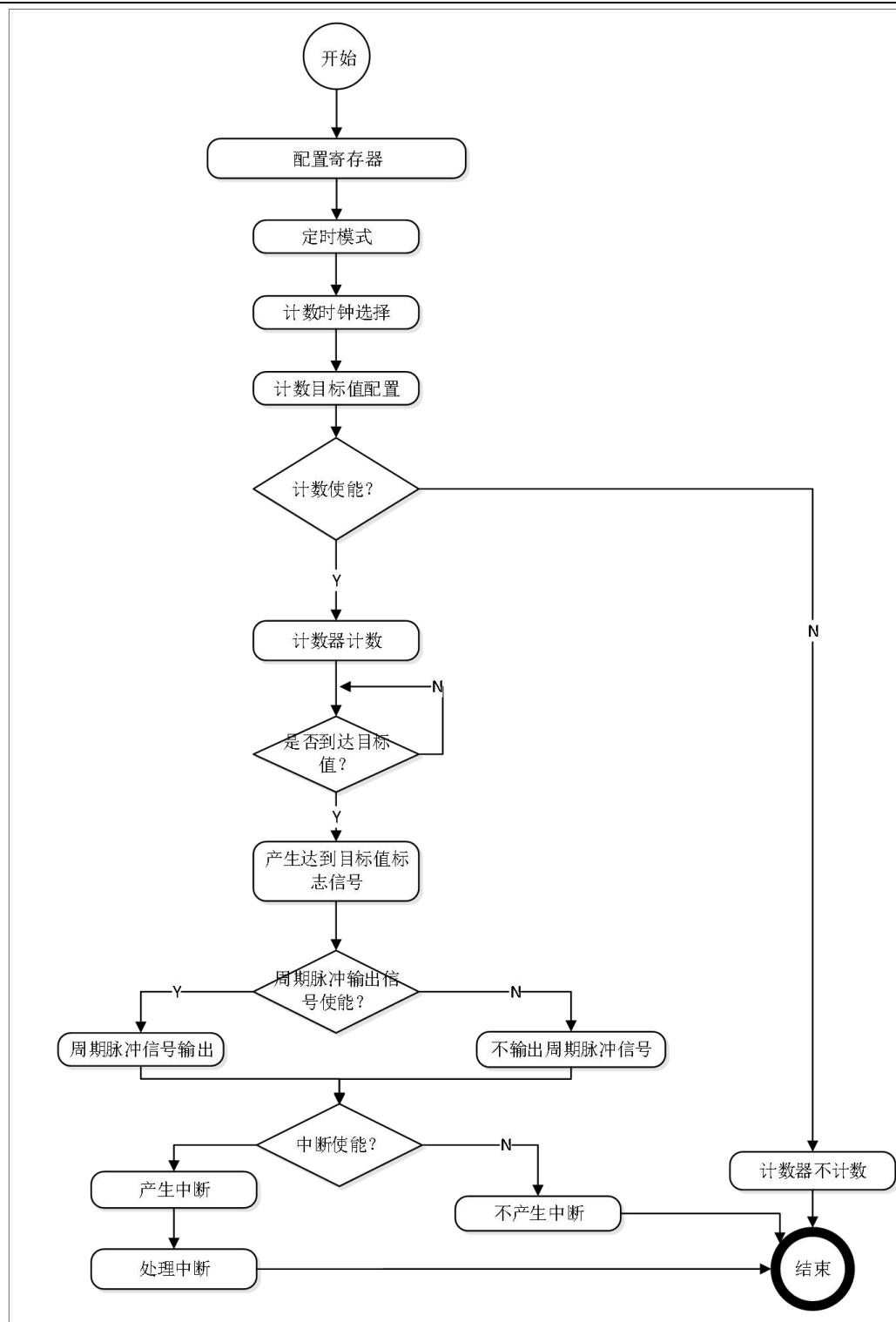


picture5-49 TIMERPLUS interrupt flag and interrupt diagram



Operating procedures

Timing mode operation flow



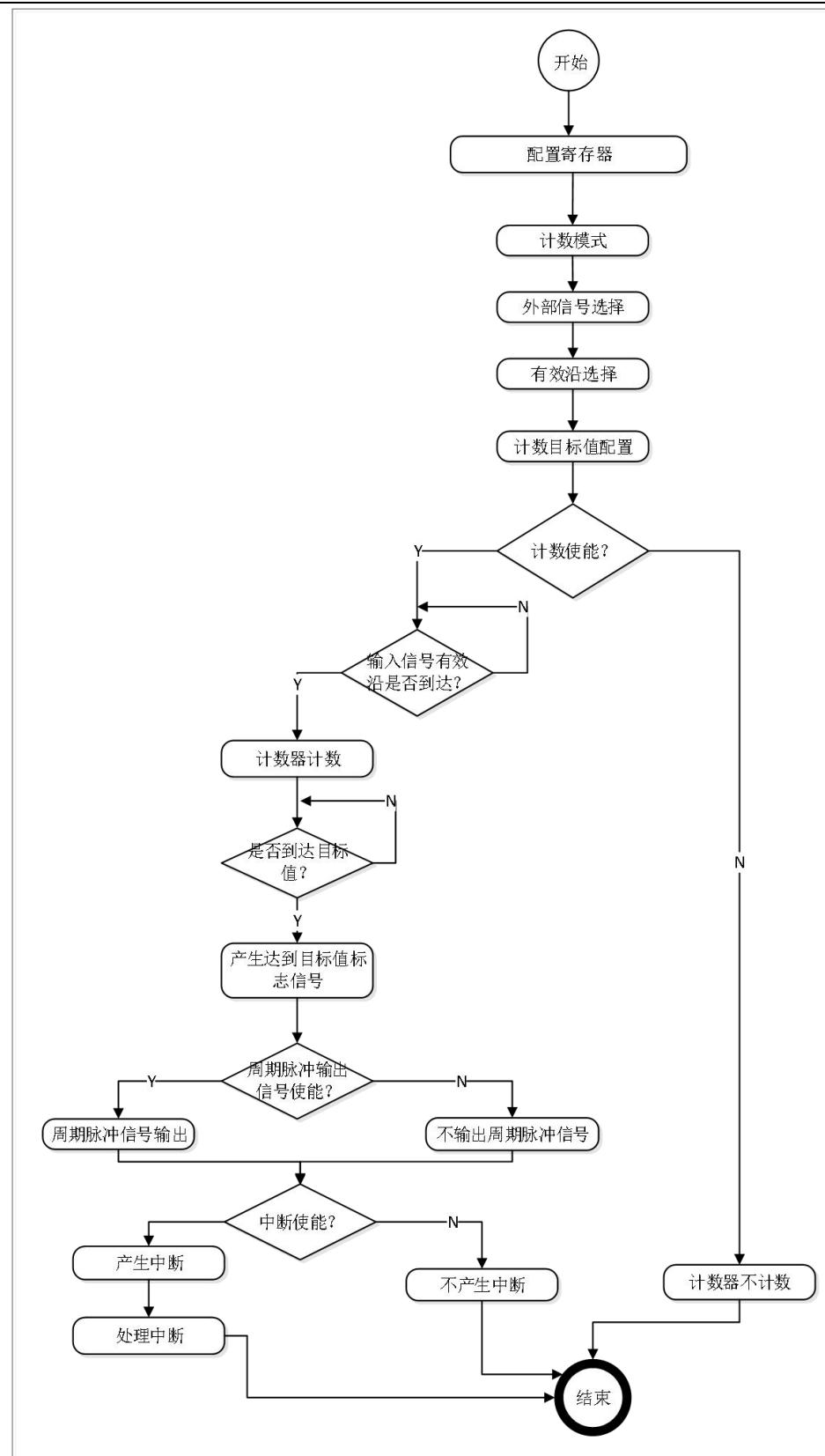
picture5-50 TIMERPLUS Timing Mode Operation Flowchart

-TIMER PLUSclock enable

- portThe port is configured as TIMER PLUSFunction
- Configure the counter working mode register to select the timing mode
- Configure Counter Clock
- If you choose internalpclkIf the clock is the counter clock, you need to configure the prescaler register to divide the clock
- Configure the counter count target value
- Configure interrupt enable
- Configure periodic pulse signal enable
- Configure the counter to enable and start counting



Counting mode operation flow

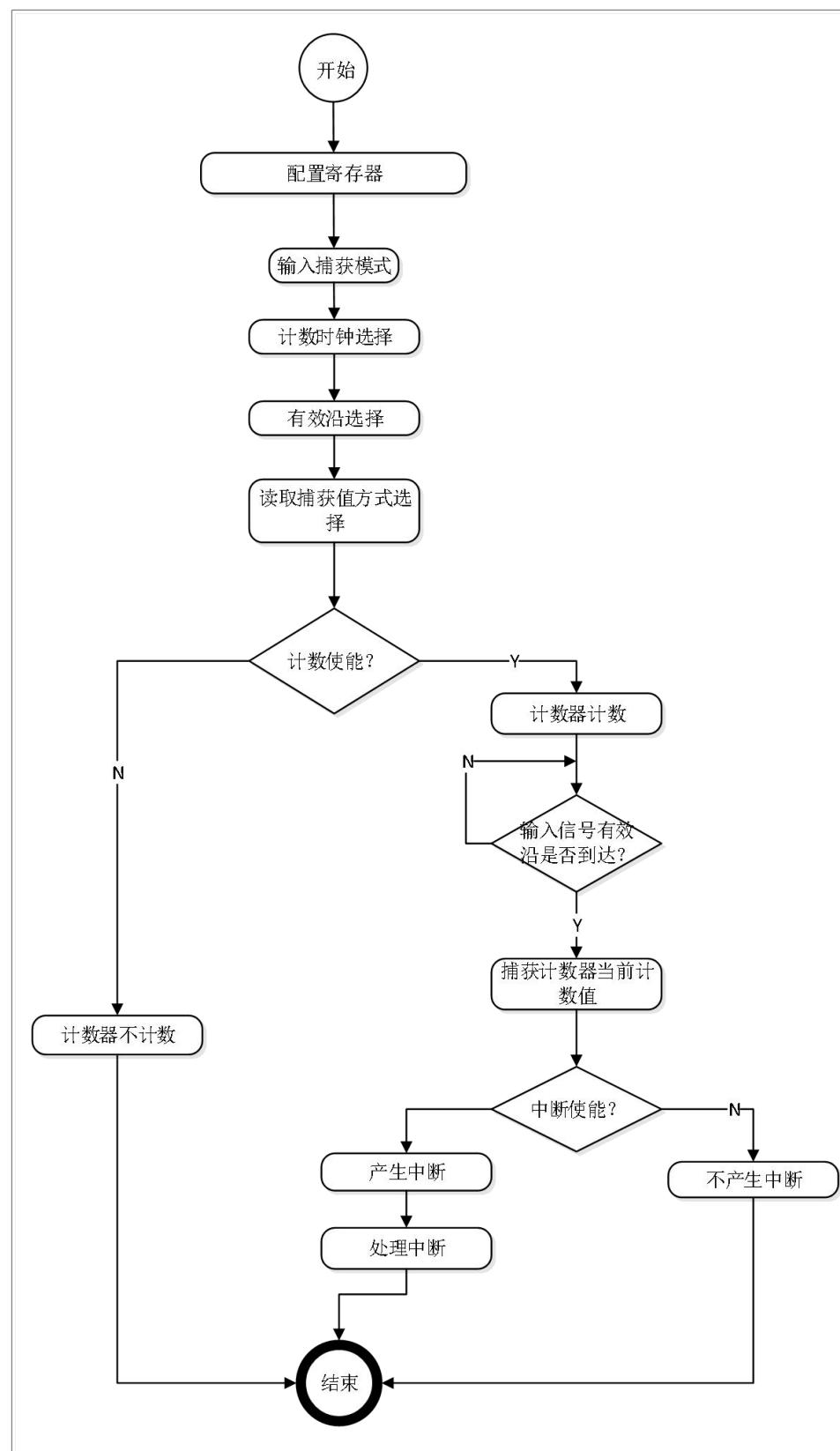


picture5-51 TIMERPLUSOperation flow chart of counting mode

- TIMERPLUSclock enable
- portThe port is configured asTIMERPLUSFunction
- Configure the counter working mode register to select the counting mode
- Configure off-chip input signals and valid edges
- Configure the prescaler register to divide the clock frequency
- Configure the counter count target value
- Configure interrupt enable
- Configure periodic pulse signal enable
- Configure the counter to enable and start counting



Input Capture Mode Operation Flow

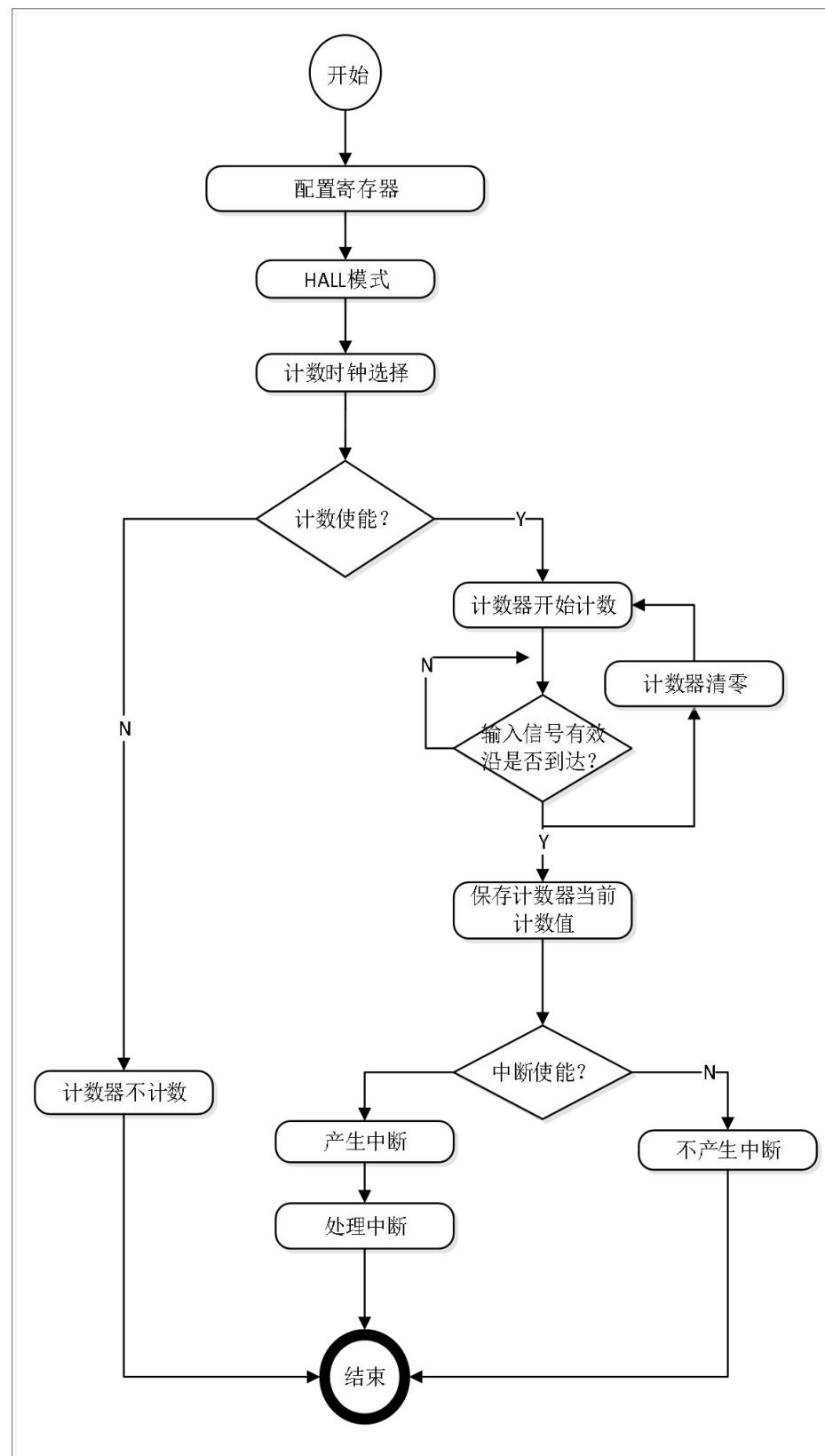


picture5-52 TIMERPLUS Input Capture Mode Operation Flowchart

- TIMER PLUSclock enable
- portThe port is configured asTIMER PLUSFunction
- Configure the counter operating mode register to select the input capture mode
- Configure Counter Clock
- If you choose internalpclkIf the clock is a counter clock, you need to configure a prescaler counter to divide the clock
- Configure the input signal and the valid edge of the input signal
- Configure interrupt enable
- Configure the counter to enable and start counting



HALLMode operation flow



picture5-53 TIMERPLUS HALLMode Operation Flowchart

- TIMER PLUSclock enable
- portThe port is configured asTIMER PLUSFunction
- choose low16bit counter
- Configure the counter operating mode register, selectHALLmodel
- Configure Counter Clock
- If you choose internalpclkIf the clock is a counter clock, you need to configure a prescaler counter to divide the clock
- Configure interrupt enable
- Configure the counter to enable and start counting

register map

name	offset	bit	width	type	reset	value	describe
TIMERPLUS0	BASE: 0x40067000						
TIMERPLUS1	BASE: 0x40067800						
TIMERPLUS_EN	0x00		32	R/W	0x00		TIMERPLUSenable register
TIMERPLUS_DIV	0x04		32	R/W	0x00		TIMERPLUSCount Clock Prescaler Register
TIMERPLUS_CTR	0x08		32	R/W	0x600060		TIMERPLUSconfiguration register
TIMERPLUS_IE	0x10		32	R/W	0x00		TIMERPLUSinterrupt enable register
TIMERPLUS_IF	0x14		32	R/W	0x00		TIMERPLUSinterrupt Status Register
HIGH_GOAL	0x20		32	R/W	0xffff		TIMERPLUS HIGHtarget configuration register
HIGH_CNT	0x24		32	R	0x00		TIMERPLUS HIGHCurrent count value register
HIGH_CVAL	0x28		32	R	0x00		TIMERPLUS HIGHcapture value register
LOW_GOAL	0x30		32	R/W	0xffff		TIMERPLUS LOWtarget configuration register
LOW_CNT	0x34		32	R	0x00		TIMERPLUS LOWCurrent count value register
LOW_CVAL	0x38		32	R	0x00		TIMERPLUS HIGHcapture value register
HALL_VAL	0x40		32	R	0x00		HALLSignal Raw Value Register

Register description
TIMERPLUS_ENregister(0x00)

bit field	name	type	reset value	describe
31:2	RESERVED	R	0	reserved bit
1	TIMERPLUS_HI GH_EN	R/W	0	TIMERPLUShigh16bitTimer Enable Register 0: disabled 1:Enable
0	TIMERPLUS_LO W_EN	R/W	0	TIMERPLUSLow16bitTimer Enable Register 0: disabled 1:Enable

TIMERPLUS_DIVregister(0x04)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	TIMERPLUS_DIV	R/W	0	TIMERPLUSCount Clock Divider Register 0x0000:express1frequency division 0x0001:express2frequency division ... 0xFFFF:express65536frequency division

TIMERPLUS_CTRregister(0x08)

bit field	name	type	reset value	describe
31:25	RESERVED	R	0	reserve



twenty four	HIGH_DMA_EN	R/W	0	DMARreadTIMER HIGHcapture value enable 0 :CPUread capture value 1:DMAread capture value
twenty three	HIGH_PO_MD	R/W	0	TIMER HIGHPeriodic pulse output enable 0: output off 1: output enable
22:21	HIGH_EXT_EDGE	R/W	0x3	TIMER HIGHCounting mode or input capture mode Input signal active edge selection 00: Rising edge is valid 01: Falling edge is valid 10: Rising edge or falling edge is valid 11:reserve
20	HIGH_EXT_SEL	R/W	0	TIMER HIGHCount mode or input capture mode input signal selection 0:timer_in0 1:timer_in1
19:18	HIGH_CLKSEL	R/W	0	TIMER HIGHCount clock source selection 00:PCLK/PREDIV(choosepclkprescaled clock) 01:CNTSRC0 10:CNTSRC1 11:reserve
17:16	HIGH_MODE	R/W	0	TIMER HIGHOperating Mode Register 00: Timing mode (can generate periodic pulse output signal) 01: Counting mode (counting clock can only be selectedpclk) 10: Enter capture mode 11:reserve
15:9	RESERVED	R	0	reserve
8	LOW_DMA_EN	R/W	0	DMARreadTIMER LOWcapture value enable 0 :CPUread capture value 1:DMAread capture value
7	LOW_PO_MD	R/W	0	TIMER LOWPeriodic pulse output enable 0: output off 1: output enable

6:5	LOW_EXT_EDGE	R/W	0x3	TIMER LOWCounting mode or input capture mode Input signal active edge selection 00: Rising edge is valid 01: Falling edge is valid 10: Rising edge or falling edge is valid 11:reserve
4	LOW_EXT_SEL	R/W	0	TIMER LOWCount mode or input capture mode input signal selection 0:timer_in0 1:timer_in1
3:2	LOW_CLKSEL	R/W	0	TIMER LOWCount clock source selection 00:PCLK/PREDIV(choosepclkprescaled clock) 01:CNTSRC0 10:CNTSRC1 11:reserve
1:0	LOW_MODE	R/W	0	TIMER LOWOperating Mode Register 00: Timing mode (can generate periodic pulse output signal) 01: Counting mode (counting clock can only be selectedpclk) 10: Enter capture mode 11:HALLmodel

TIMERPLUS_IERegister(0x10)

bit field	name	type	reset value	describe
31:22	RESERVED	R	0	reserved bit
twenty one	HALL2_F_IE	R/W	0	HALL2Falling edge interrupt enable
20	HALL2_R_IE	R/W	0	HALL2Rising edge interrupt enable
19	HALL1_F_IE	R/W	0	HALL1Falling edge interrupt enable
18	HALL1_R_IE	R/W	0	HALL1Rising edge interrupt enable
17	HALLO_F_IE	R/W	0	HALLOFalling edge interrupt enable

16	HALLO_R_IE	R/W	0	HALLORising edge interrupt enable
15:11	RESERVED	R	0	reserved bit
10	HIGH_PF_IE	R/W	0	TIMERPLUS HIGHInput pulse falling edge interrupt enable
9	HIGH_PR_IE	R/W	0	TIMERPLUS HIGHInput pulse rising edge interrupt enable
8	HIGH_TO_IE	R/W	0	TIMERPLUS HIGHTarget reached interrupt enable
7:3	RESERVED	R	0	reserved bit
2	LOW_PF_IE	R/W	0	TIMERPLUS LOWInput pulse falling edge interrupt enable
1	LOW_PR_IE	R/W	0	TIMERPLUS LOWInput pulse rising edge interrupt enable
0	LOW_TO_IE	R/W	0	TIMERPLUS LOWTarget reached interrupt enable

TIMERPLUS_IFregister(0x14)

bit field	name	type	reset value	describe
31:22	RESERVED	R	0	reserved bit
twenty one	HALL2_F_IF	R/W	0	HALL2Falling edge interrupt status Write1clear
20	HALL2_R_IF	R/W	0	HALL2Rising edge interrupt status Write1clear
19	HALL1_F_IF	R/W	0	HALL1Falling edge interrupt status Write1clear
18	HALL1_R_IF	R/W	0	HALL1Rising edge interrupt status Write1clear
17	HALLO_F_IF	R/W	0	HALLOFalling edge interrupt status Write1clear
16	HALLO_R_IF	R/W	0	HALLORising edge interrupt status Write1clear
15:11	RESERVED	R	0	reserved bit

10	HIGH_PF_IF	R/W	0	TIMERPLUS HIGHInput pulse falling edge interrupt status write1clear
9	HIGH_PR_IF	R/W	0	TIMERPLUS HIGHInput pulse rising edge interrupt status write1clear
8	HIGH_TO_IF	R/W	0	TIMERPLUS HIGHTarget Value Interrupt Status Write1clear
7:3	RESERVED	R	0	reserved bit
2	LOW_PF_IF	R/W	0	TIMERPLUS LOWInput pulse falling edge interrupt status write1clear
1	LOW_PR_IF	R/W	0	TIMERPLUS LOWInput pulse rising edge interrupt status write1clear
0	LOW_TO_IF	R/W	0	TIMERPLUS LOWTarget Value Interrupt Status Write1clear

TIMERPLUS_HIGH_LOADregister(0x20)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	HIGH_LOAD	R/W	0xffff	TIMERPLUS HIGHTimer Target Configuration Register when high16bitAfter the counter counts up and reaches the set value, it will generate a corresponding status signal

Note:HIGHAfter the counter reaches the value configured by this register, the corresponding interrupt status can be generated (HIGH_TO_IF) and external

trigger signal (as in the block diagram of the module structuretimer_goal[1]Signal).

TIMERPLUS_HIGH_CNTregister(0x24)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	HIGH_CNT	R	0	TIMERPLUS HIGHThe current count value of the timer

TIMERPLUS_HIGH_CVALregister(0x28)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	HIGH_CVAL	R	0	TIMERPLUS HIGHcapture value count value

TIMERPLUS_LOW_LOADregister(0x30)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	LOW_LOAD	R/W	0xffff	TIMERPLUS LOWTimer Target Configuration Register when low16bitAfter the counter counts up and reaches the set value, it will generate a corresponding status signal

Note:LOWAfter the counter reaches the value configured by this register, the corresponding interrupt status can be generated (LOW_TO_IF) and external trigger signal (as in the block diagram of the module structuretimer_goal[0]Signal).

TIMERPLUS_LOW_CNTregister(0x34)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	LOW_CNT	R	0	TIMERPLUS LOWThe current count value of the timer

TIMERPLUS_LOW_CVALregister(0x38)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	LOW_CVAL	R	0	TIMERPLUS LOWcapture value count value

HALL_VALregister(0x40)

bit field	name	type	reset value	describe
31:3	RESERVED	R	0	reserved bit
2	HALL2_VAL	R	0	HALL2The original signal level of
1	HALL1_VAL	R	0	HALL1The original signal level of
0	HALLO_VAL	R	0	HALL0The original signal level of

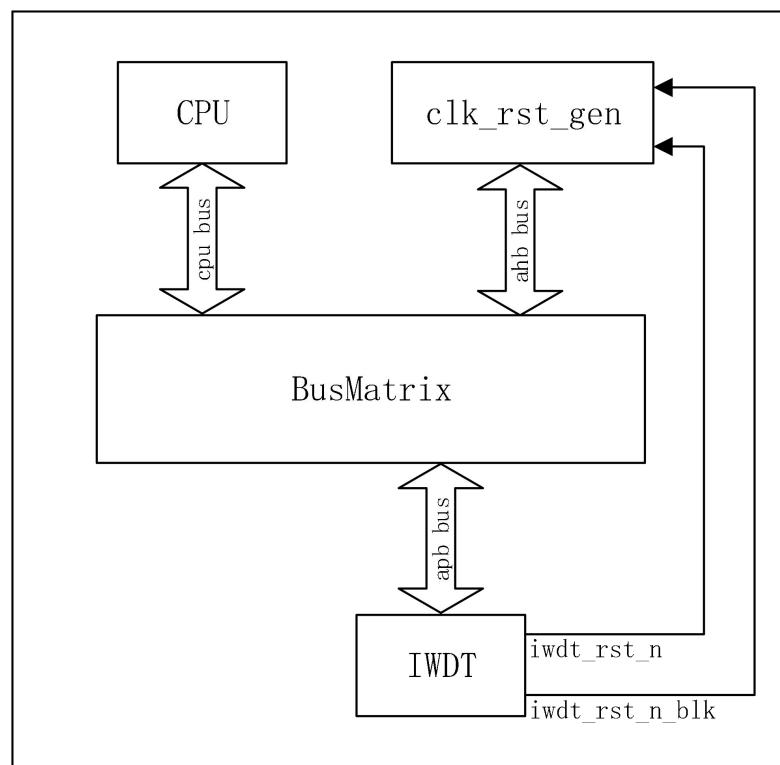
5.11Independent watchdog clock (IWDT)

5.11.1overview

Independent Watchdog Timer (IWDT) is mainly used to control the correct flow of the program.

Reset the chip when the program executes the specified program. Correspondence needs to be enabled before useIWDTModule clock. Its system block diagram is as follows

As shown in the figure:

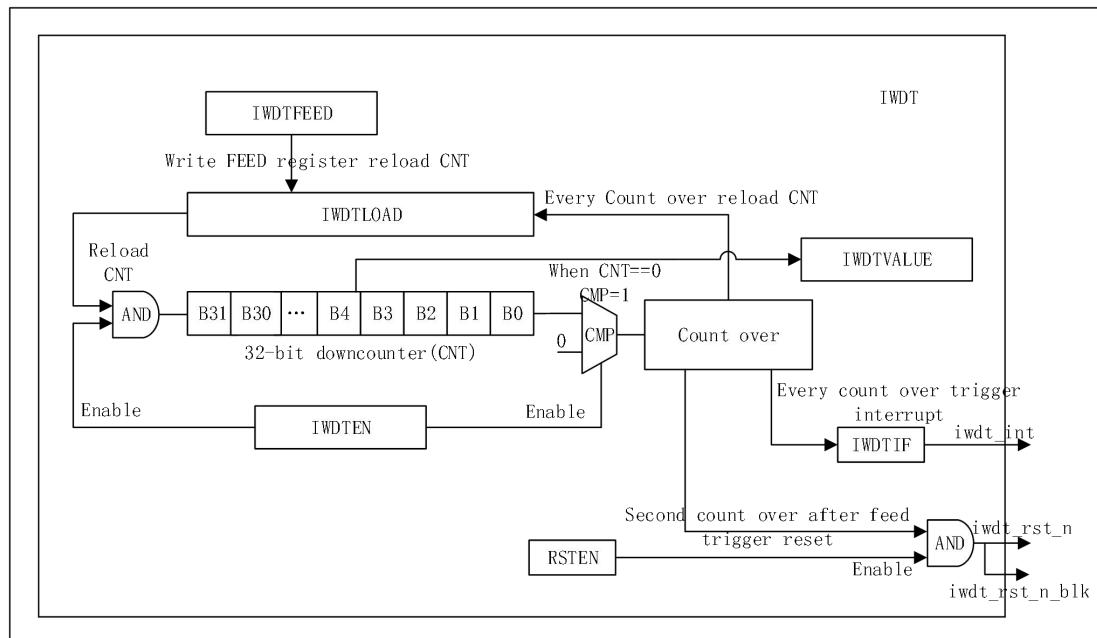


picture5-54 IWDTSystem Block Diagram

5.11.2 characteristic

- Generate counter overflow reset signal, reset signal enable can be configured
- have32Bit count bit width, configurable flexible, wide-range overflow period
- With interrupt function, it provides periodic count overflow interrupt signal, if the interrupt signal is generated in the next interrupt signal
If it is not cleared before birth, it will generate watchdogreset signal.
- With dog feeding function

5.11.3 Block Diagram of Module Structure



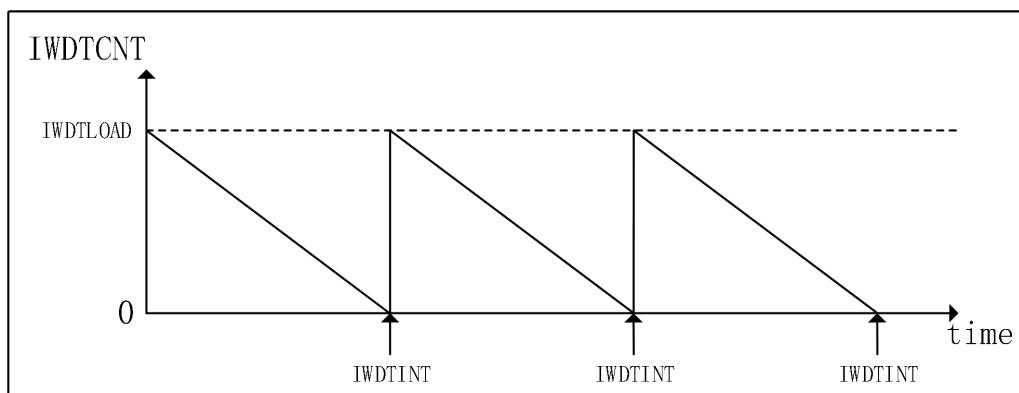
picture5-55 IWDTBlock Diagram of Module Structure

The above figure is the structural block diagram of the independent watchdog clock module. This module has a 32bitA count-down counter of the

The initial value of the counter can be passed through the registerIWDTLOADconfigured and enabled by the counterIWDTENControl the start of the counter and stop when the counter counts to0, an interrupt signal is generated,iwdt_intAfter the interrupt signal is generated, if there has been no get a response, then when the counter counts up again to0,ifIWDTControlin registerRSTENBit is1when reset signal is valid, otherwise whenRSTENfor0, the reset signal is invalid, and this signal can be used to generate a system reset.

5.11.4 Functional description

interrupt generated



picture5-56 IWDTinterrupt generation diagram

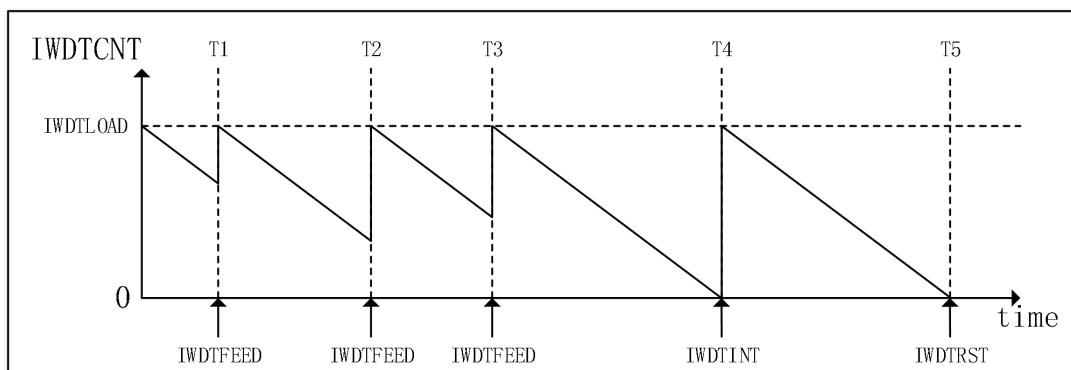
IWDT built-in 32BIT counter IWDTCNT, when the watchdog is enabled IWDTEN when valid, IWDTCNT will load the set up IWDTLOAD value, and starts counting down, whenever IWDTCNT the count value arrives 0 will generate IWDT interrupt status WDT_IF signal, and will directly generate a system interrupt.

Note that the interrupt status is only generated with the IWDTCNT the current count value is concerned, whenever the count value reaches the 0 will produce interrupted state.

When the interrupt has not been cleared, wait until the count value is counted again 0 when, if at this time RSTEN bit is valid, it will produce a look at the watchdog reset, at this time IWDTCNT will be cleared 0.

IWDTFeed dog and reset generation

1. Feed the dog before the first interruption

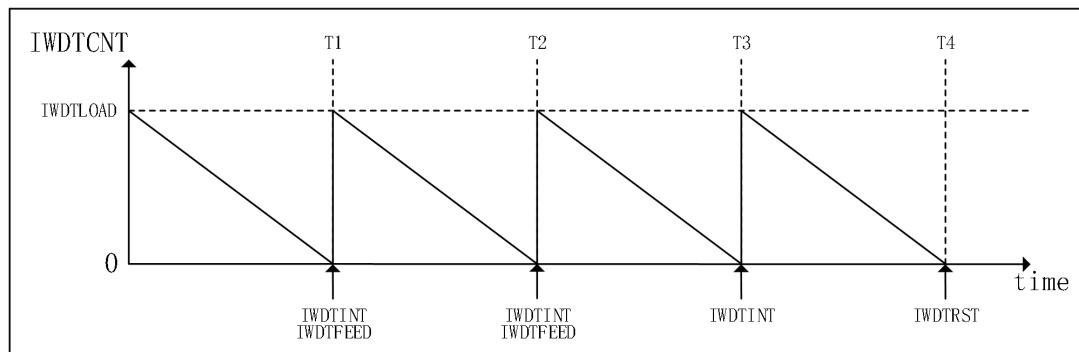


picture5-57 IWDTfeed the dog and reset before the first interruption

As shown in the figure above, when IWDTCNTThe counter has not yet reached for the first time 0 When the watchdog feeds the dog, the counter will reset from IWDTLOADstart counting (T1,T2,T3time) . After feeding the dog for the last time (T3time) IWDTCNTproduce When interrupted twice (T4,T5time), and RSTENvalid, a watchdog reset will be generated IWDT_RST(T5time).

The watchdog reset is only generated when the second interrupt is generated after feeding the dog, if the dog feeding operation is performed before the second interrupt is generated operation, re-record the number of interrupts.

2, Feed the dog before the second interruption

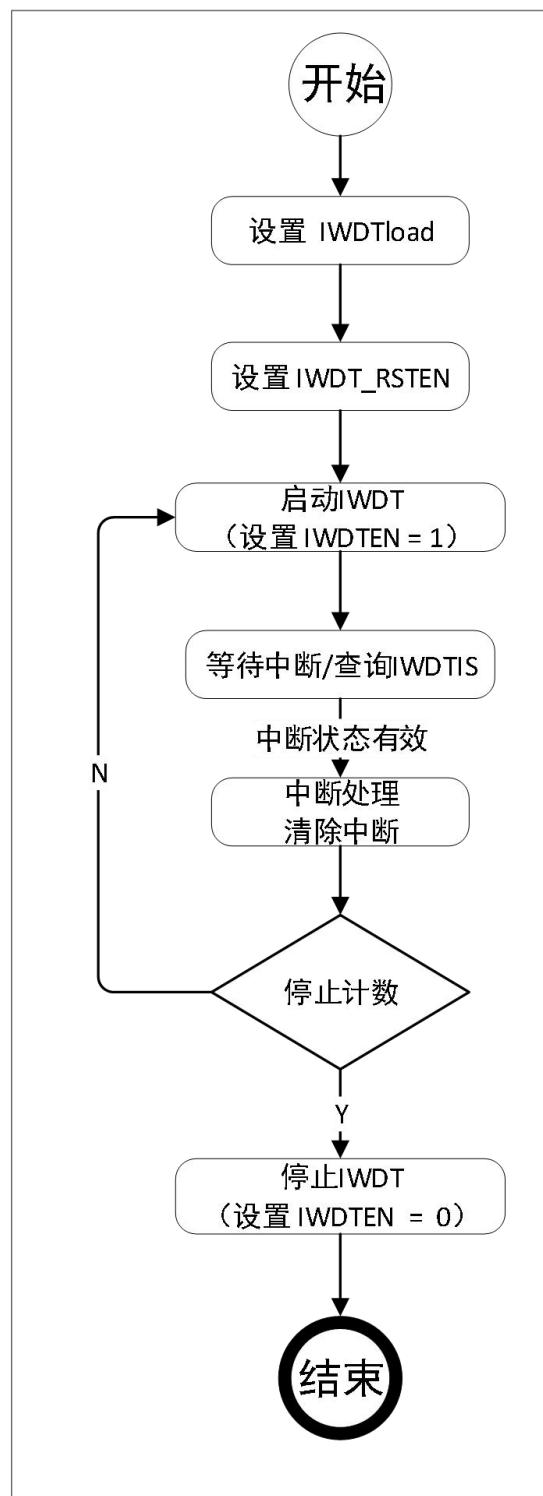


picture5-58 IWDTFeed the dog and reset before the second interruption

As shown in the figure above, when IWDTCNTThe counter arrives once 0 After the watchdog feeds the dog, the count will restart from the IWDTLOAD start counting (T1,T2time) . After feeding the dog for the last time (T2time), IWDTCNTWhen two interrupts are generated (T3, T4time) , and RSTENvalid, a watchdog reset is generated IWDT_RST(T4time).



Operating procedures



- IWDTclock enable
- Configuration Initial Value Register (IWDTLOAD)
- Configuration Reset Enable Register (RSTEN)
- Configuration Enable Register (IWDTEN)
- Wait for an interrupt to occur and handle the interrupt
- If interrupted at IWDTValuecount again to 0 has not been processed yet, a watchdog reset will be generated

register map

name	Offset	bit width	type	reset value	describe
IWDT BASE: 0x4006A000					
IWDT_LOAD	0x00	32	R/W	0xfffff	IWDTinitial value register
IWDT_CTRL	0x08	32	R/W	0x00	IWDTcontrol register
IWDT_IF	0x0C	32	R/W	0x00	IWDTstatus register
IWDT_FEED	0x10	32	R/W	0x00	IWDTfeeding dog register

Register description

IWDT_LOADregister(0x00)

bit field	name	type	reset value	describe
31:20	RESERVED	RO	0	reserved bit
19:0	IWDTLOAD	R/W	0xfffff	<p>IWDTThe initial value of the counter configures the register.</p> <p>IWDTAt startup, the counter is automatically loaded with IWDTLOADvalue, start counting down. When the counter value reaches 0, the hardware will automaticallyIWDTLOADThe value in the register is reloaded into the counter and counting down continues.</p> <p>when first counted to 0When, an interrupt is generated, if the dog is not fed, count down again to 0, a reset occurs.</p> <p>This register must be in IWDTENConfigure when invalid.</p>

IWDT_CTRLregister(0x08)

bit field	name	type	reset value	describe
31:2	RESERVED	RO	0	reserved bit
1	INTEN	R/W	0	IWDTinterrupt enable bit 1:Enable 0:prohibit
0	IWDTEN	R/W	0	IWDTstart bit 1:start upIWDTcount 0: stop counting

IWDT_IFregister(0x0C)

bit field	name	type	reset value	describe
31:1	RESERVED	RO	0	reserved bit
0	IWDT_IF	R/W	0	IWDTstatus bit, counting up to0, active high hardware set, software write1to clear

IWDT_FEEDregister(0x10)

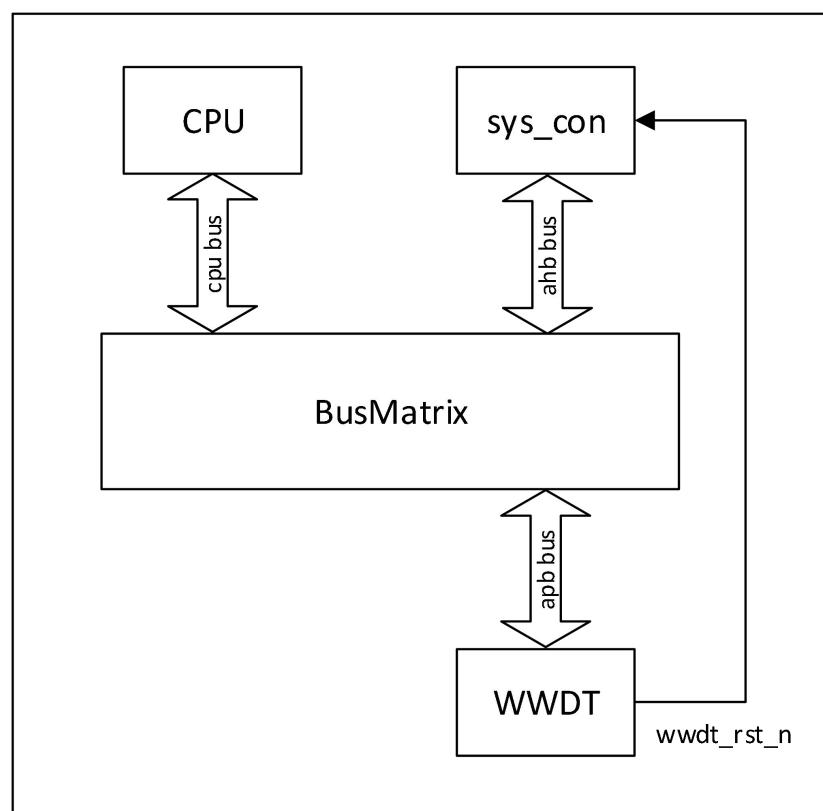
bit field	name	type	reset value	describe
31:8	RESERVED	RO	0	reserved bit
7:0	FEED	R/W	0	IWDTRestart Counter Register write to this register0x55will restart laterIWDTCounter (dog feeding operation)

5.12 Window watchdog clock (WWDT)

5.12.1 overview

Window Watchdog Timer (WWDT) is mainly used to control the correct flow of the program.

Reset the chip when the program executes the specified program. Correspondence needs to be enabled before useWWDTModule clock.



picture5-60 WWDTModule System Block Diagram

5.12.2 characteristic

-Watchdog count clock source is system clock4096frequency division

- Features: support7Bit down counting, providing window interrupt with pre-reset interrupt, if one after pre-reset interrupt

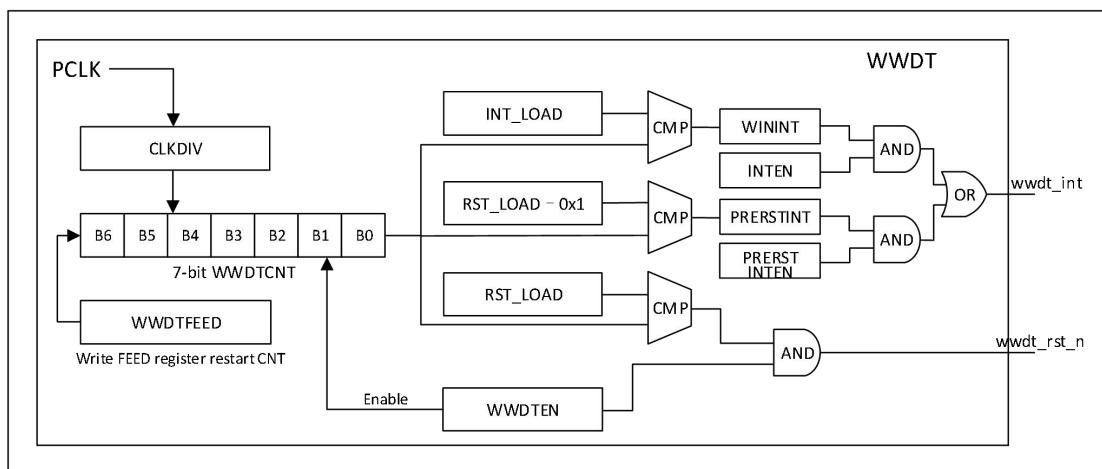
If there is still no dog feeding operation within the watchdog counting period, a reset signal will be generated.

-Support interrupt window value and reset window value independently configurable

-Watchdog count clock frequency configurable

-Support pre-reset interrupt alarm function

5.12.3 Block Diagram of Module Structure



picture5-61 WWDTBlock Diagram of Module Structure

WWDT built-in 7bit counter, the count clock is PCLK go through CLKDIV. The divided clock, when WWDTEN place

1, enable the counter, the counter loads the counting initial value from the register 0x7f, start counting down.

When the count reaches INT_LOAD When setting the value, WWDT produce WININT Interrupt status, if the interrupt enable is enabled

INTEN, a system interruption occurs wwdt_int.

When the count reaches RST_LOAD - 1 When setting the value, WWDT produce PRERSTINT Interrupt status, if the interrupt is enabled

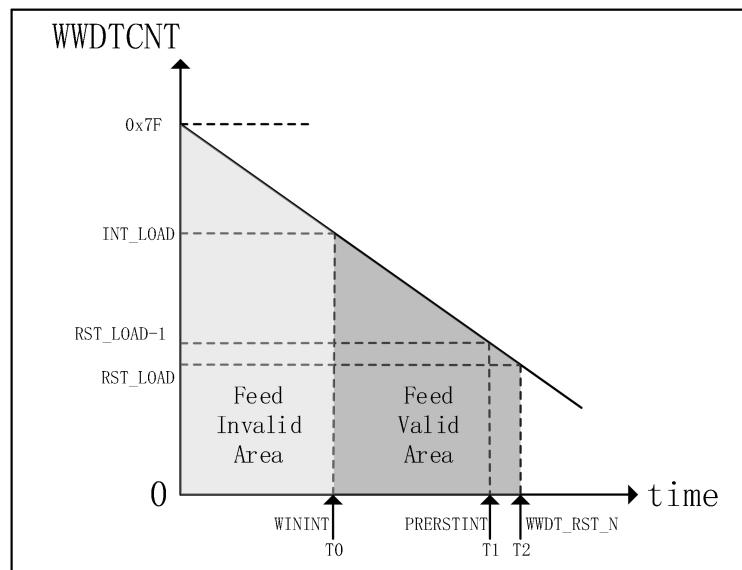
able PRERSTINTEN, a system interruption occurs wwdt_int.

When the count reaches RST_LOAD A watchdog reset is generated when the value is set wwdt_rst_n.

to write WWDTCNT register 0x55A watchdog feed occurs, causing the counter to go from 0x7f. The value restarts the count.

5.12.4 Functional description

WWDT Interrupt generation and dog feeding interval



picture5-62 WWDT interrupt generation and dog feeding interval

As shown in FIG:

when WWDTCNTThe count value arrives INT_LOADhour(T0time), resulting in WININTinterrupted state;

when WWDTCNTThe count value arrives RST_LOAD-1hour(T1time), resulting in PRERSTINTinterrupted state;

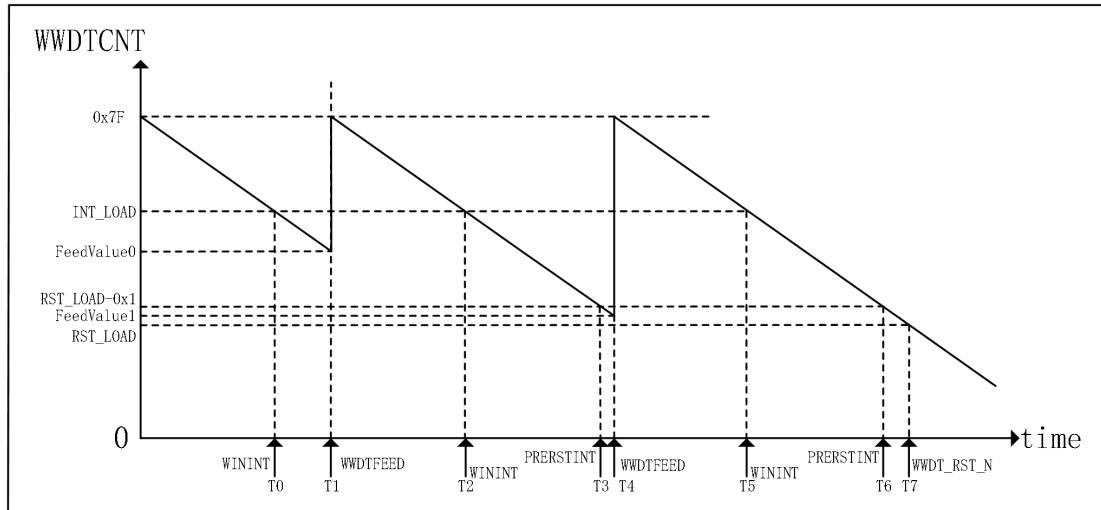
when WWDTCNTThe count value arrives RST_LOADhour(T2time) ,produce WWDT_RST_Nwatchdog reset;

WWDTInterrupt generation is only related to the current count value.

from 0time to T0Time is an invalid dog feeding interval, if the dog is fed at this time, a watchdog reset will be generated immediately;

from T0time to T2Time is the effective dog feeding interval, if the dog is fed at this time, it will be reset WWDTCNTWatchdog counter.

WWDTFeed dog and reset generation



picture5-63 WWDTFeed the dog and reset

As shown in FIG:

whenWWDTCNTThe count value arrivesINT_LOADhour(T0time), resulting inWININTinterrupted state;

whenWWDTCNTThe count value arrivesFeedValue0hour(T1moment), the watchdog feeds the dog, and the counter restarts
count;

whenWWDTCNTThe count value arrivesINT_LOADhour(T2time), resulting inWININTinterrupted state;

whenWWDTCNTThe count value arrivesRST_LOAD-1hour(T3time), resulting inPRERSTINTinterrupted state;

whenWWDTCNTThe count value arrivesFeedValue1hour(T4moment), the watchdog feeds the dog, and the counter restarts
count;

whenWWDTCNTThe count value arrivesINT_LOADhour(T5time), resulting inWININTinterrupted state;

whenWWDTCNTThe count value arrivesRST_LOAD-1hour(T6time), resulting inPRERSTINTinterrupted state;

whenWWDTCNTThe count value arrivesRST_LOADhour(T7time) ,produceWWDT_RST_Nwatchdog reset.

register map

name	Offset	bit width	type	reset value	describe
WWDT		BASE: 0x4006A800			
WWDT_LOAD	0x00	32	R/W	0x40	WWDTinitial value register
WWDT_VALUE	0x04	32	R/W	0x7f	WWDTCurrent count value register
WWDT_CTRL	0x08	32	R/W	0x00	WWDTcontrol register
WWDT_IF	0x0C	32	R/W	0x00	WWDTInterrupt Status Register
WWDT_FEED	0x10	32	R/W	0x00	WWDTfeeding dog register

Register description

WWDT_LOADregister(0x00)

bit field	name	type	reset value	describe
31:14	RESERVED	RO	0	reserved bit
13:8	RST_LOAD	R/W	0x0	Window Reset Compare Value Register Note1: When the watchdog counts down to this value, a reset can be generated Note2: Watchdog counts down to this value plus1interrupt can be generated when
7	RESERVED	RO	0	reserved bit
6:0	INT_LOAD	R/W	0x40	Window Interrupt Compare Value Register Note1: When configuring, the window interrupt comparison value must be greater than the window reset comparison value Note2: When the watchdog counts down to this value, an interrupt can be generated

WWDT_VALUEregister(0x04)

bit field	name	type	reset value	describe
31:7	RESERVED	RO	0	reserved bit

6:0	VALUE	R	0x7f	This register is a read-only register, the reset value is 0x7f. When this register is read, the current count value of the counter is returned.
-----	-------	---	------	---

WWDT_CTRLregister(0x08)

bit field	name	type	reset value	describe
31:5	RESERVED	RO	0	reserved bit
4	PRERSTINTEN	R/W	0	Pre-reset interrupt enable bit 1:Enable 0:prohibit
3	INTEN	R/W	0	Window interrupt enable bit 1:Enable 0:prohibit
2:1	CLKDIV	R/W	0	Watchdog counter count clock prescaler 00: Watchdog count clock1frequency division 01: Watchdog count clock2 frequency division 10: Watchdog count clock4frequency division 11: Watchdog count clock8Frequency division Note: The watchdog counting clock is the system clock4096frequency division
0	EN	R/W	0	WWDTstart bit 1:start upWWDTcount 0: stop counting

WWDT_IFregister(0x0C)

bit field	name	type	reset value	describe
31:2	RESERVED	RO	0	reserved bit
1	PRERSTINT	R/W	0	Pre-reset interrupt flag bits (i.e. VALUE = RST_LOAD + 1 time), high effective Set by hardware, write by software1to clear
0	WININT	R/W	0	window interrupt flag bit (i.e. VALUE = INT_LOADtime), high effective Set by hardware, write by software1to clear

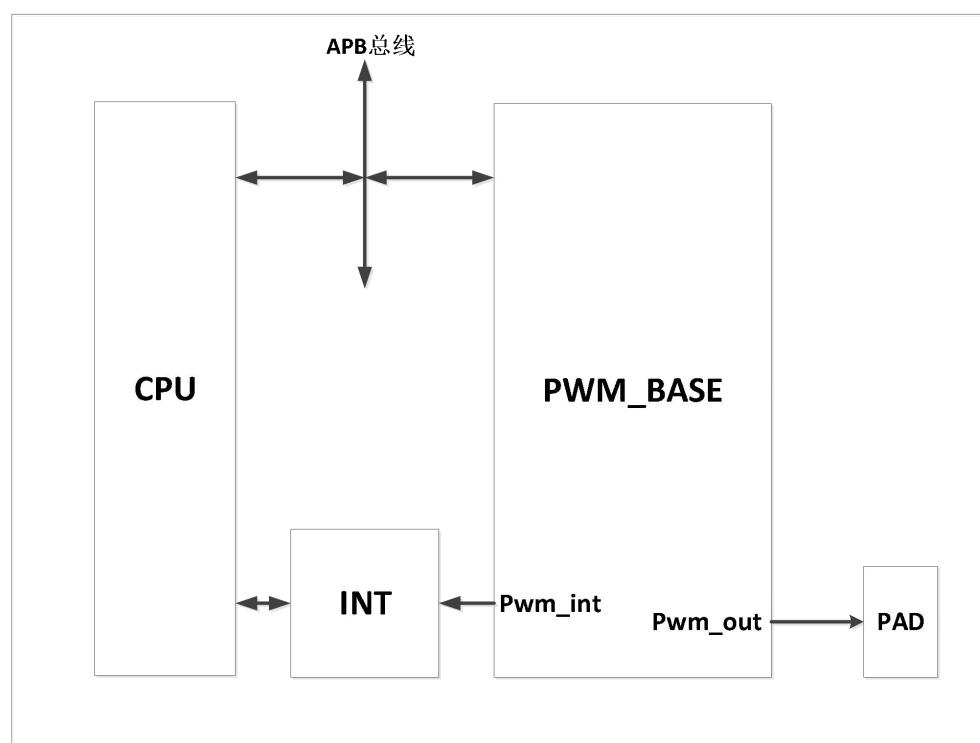
WWDT_FEEDregister(0x10)

bit field	name	type	reset value	describe
31:8	RESERVED	RO	0	reserved bit
7:0	FEED	W	0	<p>WWDTRestart Counter Register</p> <p>write to this register 0x55 will restart later WWDTCounter (dog feeding operation)</p> <p>Note: Only during the window period (INT_LOAD > VALUE > RST_LOAD) to feed the dog to take effect, otherwise the dog feed operation will generate a watchdog reset.</p>

5.13Basic PWM Generator (PWMBASE)

5.13.1 overview

PWMBASEAs a basic pulse width modulation generator, this chipPWMBASEmodule provides3road(CH0,CH1, CH2) Independent channel, supports prescaler function, supports output level inversion, supports reaching the inversion point interrupt and cycle end broken. usePWMBASEThe module needs to be enabled beforePWMBASEclock.PWMNASEmodule is passedAPBthe bus comes achieve withCPUconnected, and can generate interrupts and outputPWMWave toPADport, its system block diagram is shown in the figure below Show:



picture5-64 PWMBASEModule System Block Diagram

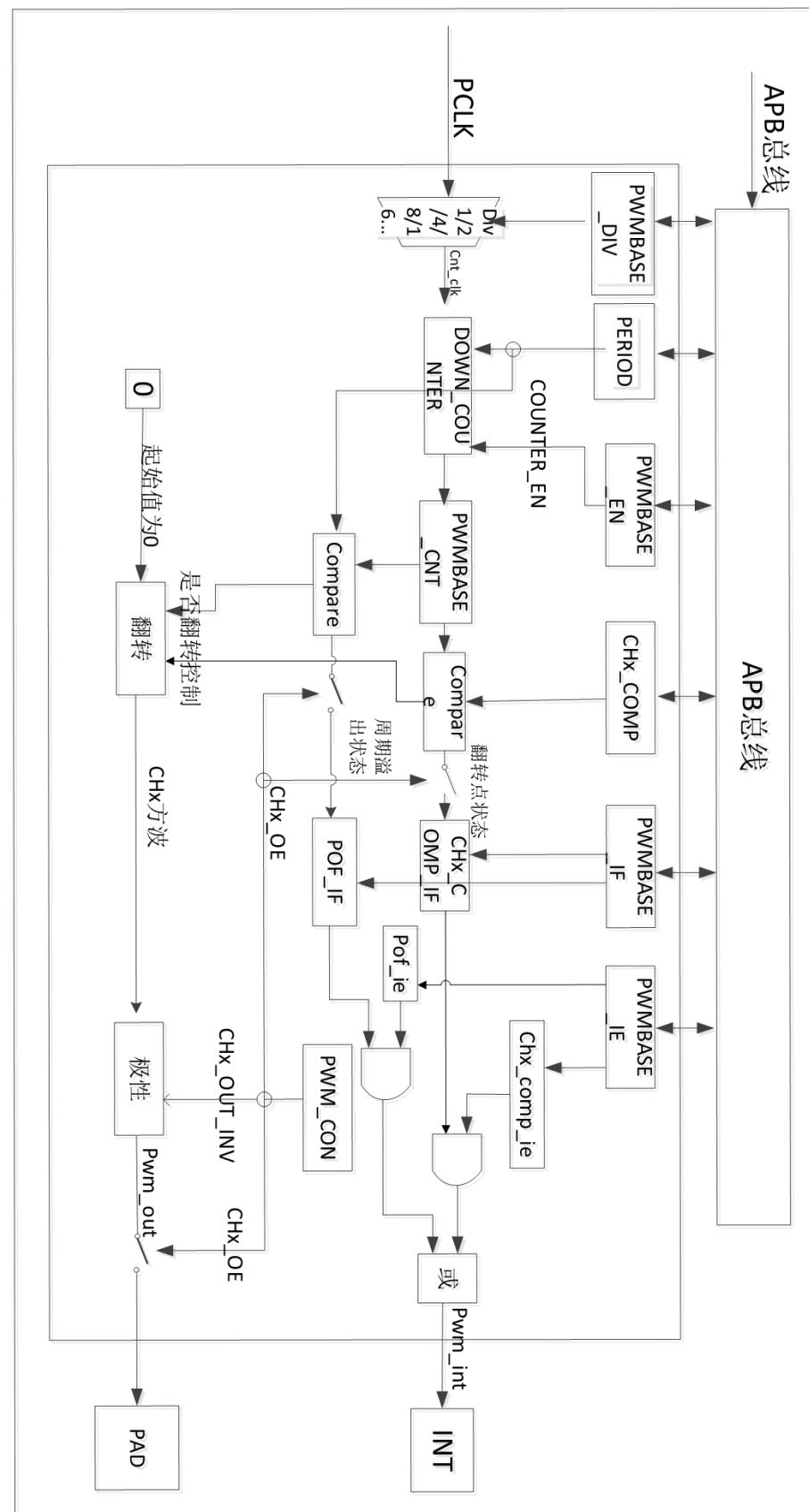
5.13.2 characteristic

- 3individual16bit PWM, can output different duty cyclePWMwaveform
- 8bitPrescaler Counter
- Whether the output level is flipped

-Supports tipping point reached interrupt and cycle end interrupt

-Internally a counter that counts down

5.13.3 Block Diagram of Module Structure



picture5-65 PWMBASEBlock Diagram of Module Structure

Pictured above isPWMBASEThe frame diagram of the module structure, as can be seen from the figure, the system clockclkable to pass8bitpre frequency division registerdivCarry out frequency division, and provide counting clock for the counter counting down after frequency division. Through the cycle configuration register periodConfigure the counting period for the counter, and the counter is enabledcounter_enControls the start and end of the count. counting starts Later, the current count value will existPWM_CNTRegister, the current count value will be compared with the configured flip point and period value, Because the initial level of the channel output is0, and it is a counter that counts down, so when the count value is less than the flip point value, the output for1, when greater than or equal to the flip point value, the output is0, when the counting period is full, the waveform will reverse again. This module passesPWMBASE output configuration register to configurerPWMBASEThe output of the waveform, wherechx_out_invControls the polarity of the channel waveform, chx_out_invfor1When the output is opposite to the original waveform, chx_out_invfor0When the output is consistent with the original waveform;chx_oe Control channel waveform output enable, whenchx_oe for1hour,PWMBASEoutput channel waveform, whenchx_oe for0hour PWMBASEOutput high impedance state.

The generation of the interrupt signal is controlled by the interrupt status and the interrupt enable. The interrupt status is divided into the flip point status and the cycle overflow status. state, and the interrupt state is only in thechx_oe for1It can only be generated when there is an interrupt status, and the corresponding middle The interrupt signal is generated only when the interrupt enable is turned on, that is, the toggle point interrupt enable and cycle overflow interrupt enable of each channel.

5.13.4Functional description

Period value and rollover point

When the channel output enable registerchx_oe for1, through the registerPWMBASE_PERIODand PWM_CHX_COMPDifferent period values and flip point values of the output waveform can be configured, as shown in the figure below, period value configuration for7, the timing diagram for different flip point values.

An example of the relationship between the configuration period value and the flip point value (the idle start level is0):

1) the user wants to generatePWMThe waveform is: period8, the duty cycle is25%, you need to configurer it as:PERIOD=0x07,

CHx_COMP=0x02.

2) the user wants to generatePWMThe waveform is: period8, the duty cycle is87.5%, you need to configurer it as:PERIOD=0x07,

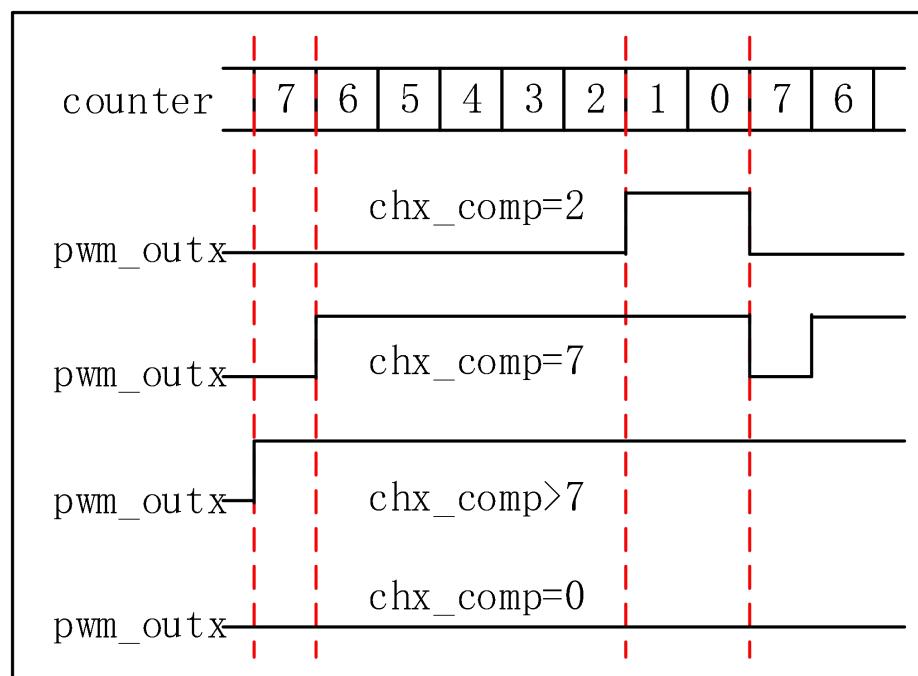
CHx_COMP=0x07.

3) the user wants to generatePWMThe waveform is: period8, the duty cycle is100%, you need to configurer it as:PERIOD=0x07,

CHx_COMP=0x08(more than the7can be).

4) the user wants to generate PWMThe waveform is: period8, the duty cycle is0%, you need to configure it as:PERIOD=0x07,
CHx_COMP=0x00.

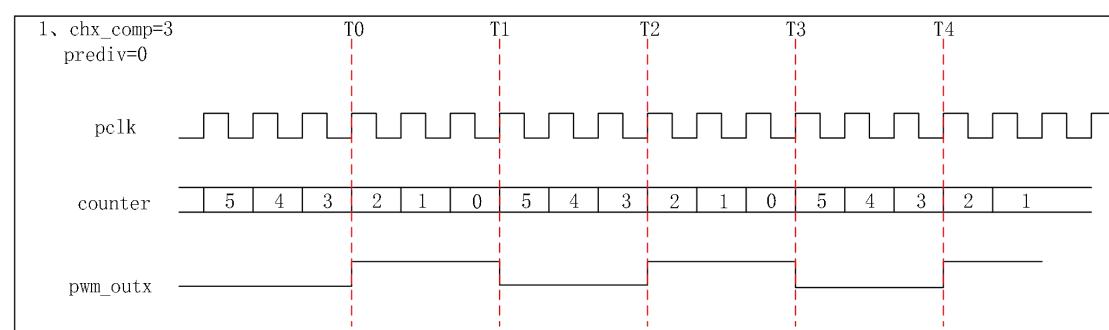
The output waveform under the relationship between the above period value and the flip point value is realized in the following schematic example.



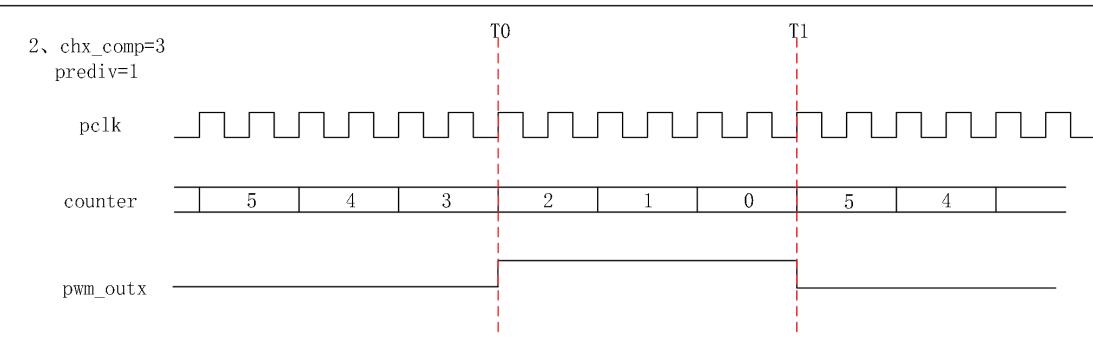
picture5-66 PWMBASETiming Diagram of Different Periods and Turnover Points

prescaler count

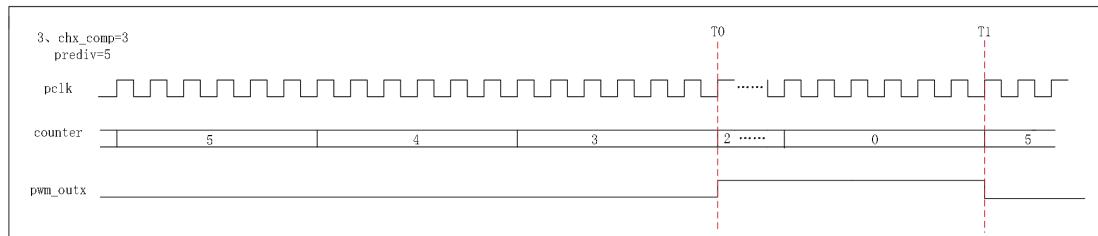
This module can be8bitPrescaler Count Registerpwmbase_divConfigure different clock frequency division values to achieve frequency division counting, the frequency division value range is1-256, in the figure below, an example of flip point value is3, the period is6, the frequency division value is1,2, 6time sequence diagram.



picture5-67 PWMBASE 1Frequency division timing diagram

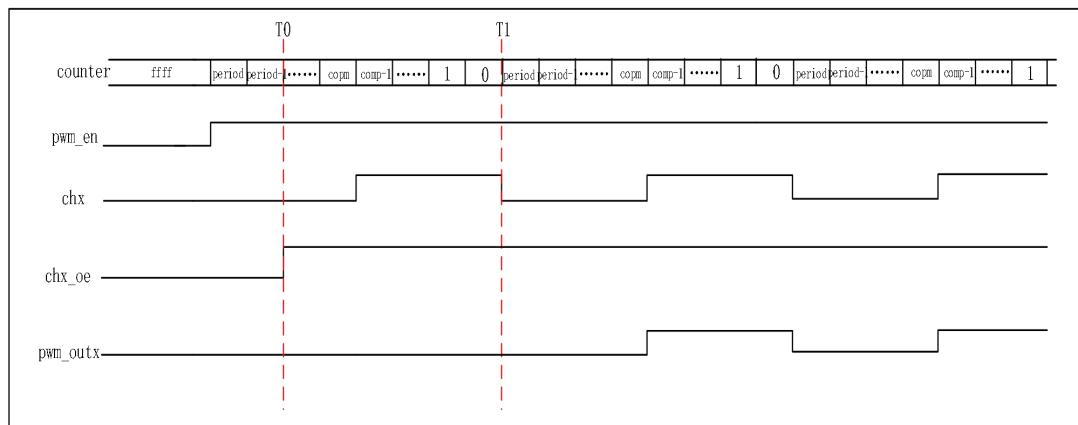


picture5-68 PWMBASE 2Frequency division timing diagram



picture5-69 PWMBASE 6Frequency division timing diagram

output enable



picture5-70 PWMBASEOutput Enable Timing Diagram

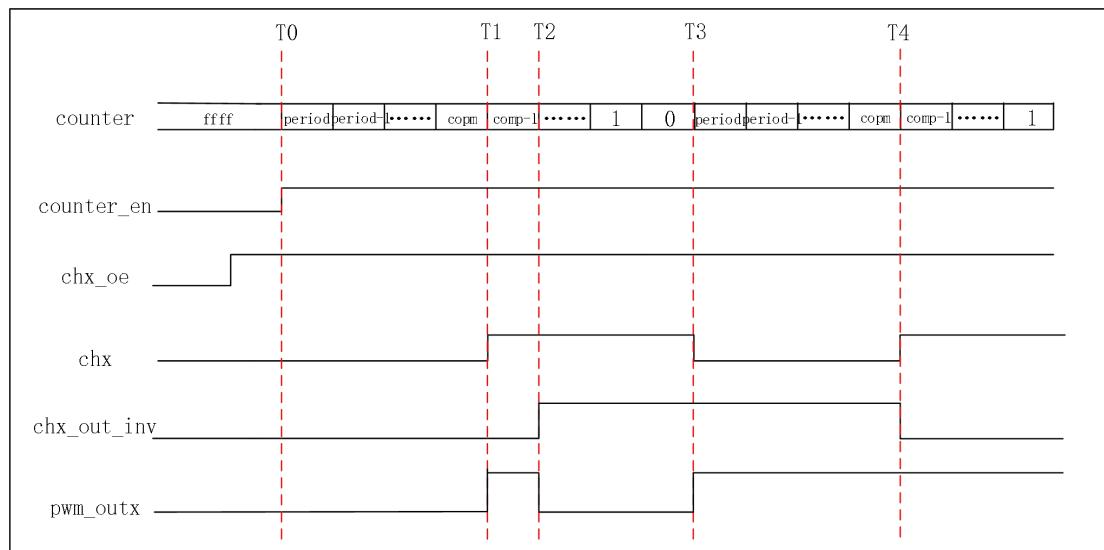
Pictured above is PWMBASEThe output enable timing diagram, where chx is the original waveform, chx_oeFor channel waveform output use able,pwm_outx for pwmOutput waveform, before the channel waveform is enabledT0before the moment,pwmoutput start level0, when T0Time Channel Waveform Enable Register chx_oe becomes1hour,pwmThe waveform is not output immediately, but the output pass

channel initial level0, from the next cycleT1The output of the channel waveform is only started at the moment to ensure the output of a complete waveform. Note required

It means that before the channel waveform is enabled, the correspondingPADThe output of the port is in a high-impedance state, and only after the channel waveform is enabled, the corresponding

answerPADThe port only outputswaveform.

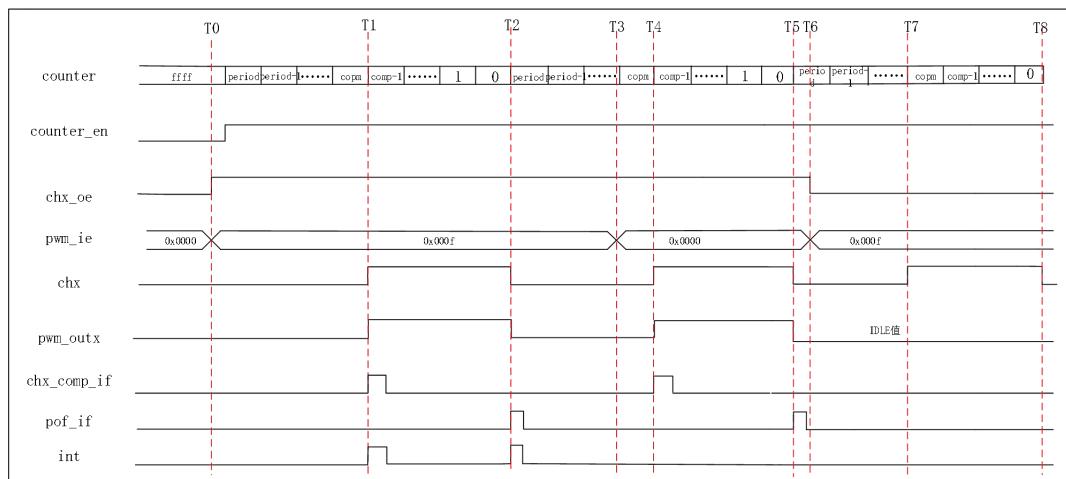
output flip



picture5-71 PWMBASEOutput Toggle Timing Diagram

As shown in the figure above, when the counter is enabled and the channel waveform output is enabled at the same time, when the polarity inversion register is configured memorychx_out_invfor0, you can see theT0-T2time periodThe output waveform is consistent with the original waveform, whenT2 time configurationchx_out_invfor1back,pwmThe output waveform will immediately reverse its polarity, opposite to the original waveform, when T4time againchx_out_invconfigured as0back,pwmThe output waveform will again match the original waveform.

to interrupt



picture5-72 PWMBASEOutput Interrupt Timing Diagram

This module supports a variety of interrupts, as shown in the figure above, inT0Always enable channel waveform outputchx_oeconfigured as1,

While setting the interrupt enable registerpwm_ieconfigured as0x000f, that is, the flip point interrupt and cycle overflow interrupt of each channel

all open. Turn on the counter enable, theT1time, when the count value reaches the flip, the flip point interrupt status register

chx_comp_ifwill be pulled high, and an interrupt signal is generated at the same time, and then the interrupt is processed, and the interrupt status register is written1cleared; when

The count value is0When , it means counting one cycle, at this time the cycle overflow state becomes1, as shown in the figureT2moment, and producing

interrupt signal, then handle the interrupt and clear the interrupt status. when inT3moment, willpwm_ieconfigured as0x0000, turn off

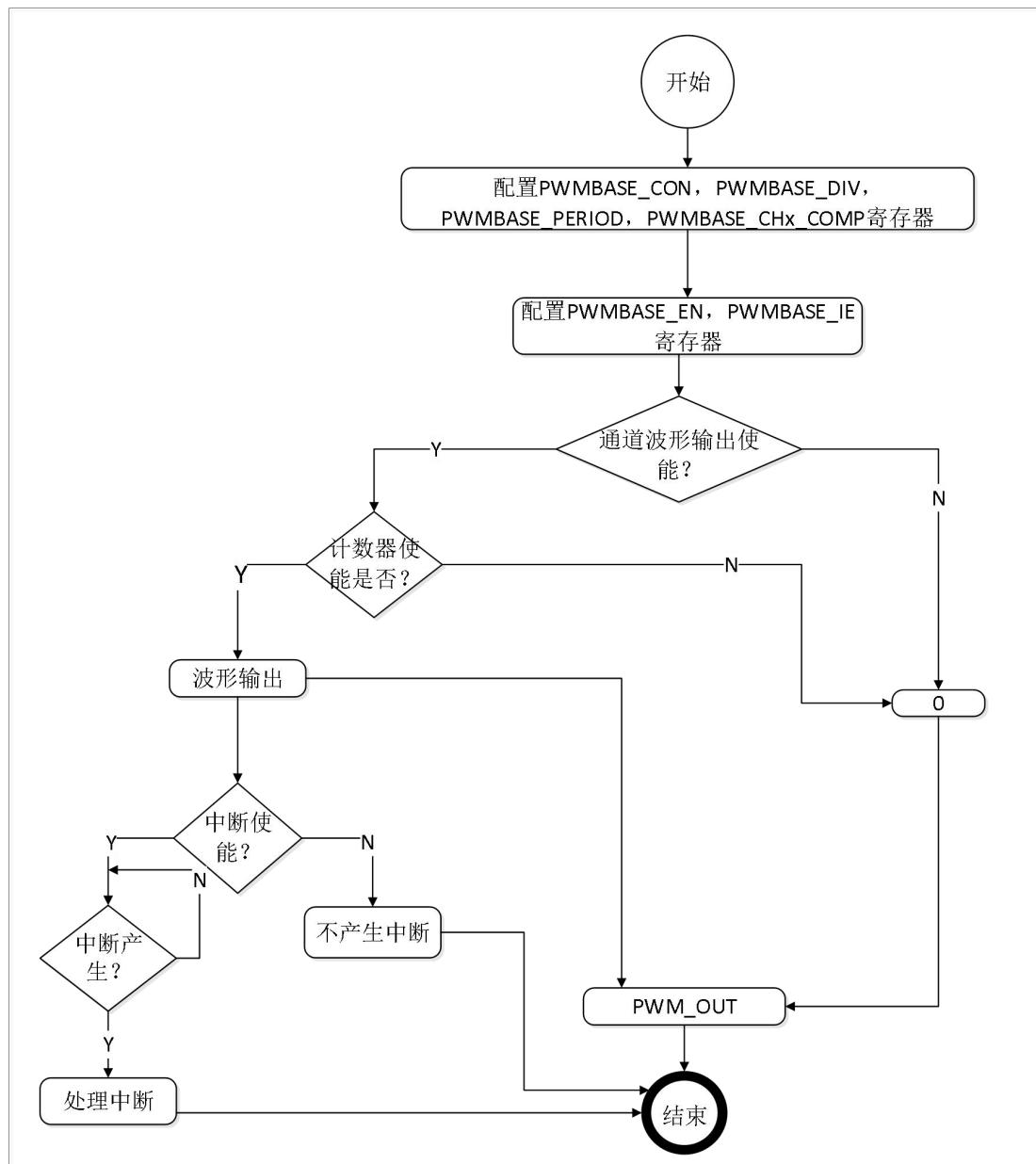
Turn-over and period overflow interrupts are enabled atT4andT5Only the corresponding interrupt status will be generated at the moment, and the interrupt signal will not be generated.

existT6Turn on the flip point and period overflow interrupt enable again at all times, and enable the channel waveform outputchx_oeconfigured as0hour,

Even if the interrupt enable is turned on, but inT7,T8Moments toggling point and cycle overflow interrupt status are not generated, therefore, no

An interrupt signal will be generated.

Operating procedures



picture5-73 PWMBASEOperation flow chart

- configurationPWMBASEclock enable
- portThe port is configured asPWMBASEFunction
- Configure the prescaler (PWMBASE_DIV)register
- Configure the output to flip (PWMBASE_CON)register
- Configure interrupt enable (PWMBASE_INT_EN)register

- configuration cycle (PWMBASEEx_PERIOD) and flip point (PWMBASEEx_CHx_COMP)register
- If interrupts are configured, enablePWMBASEto interrupt
- Configure Waveform Output Enable (PWMBASE_CON)register
- configurationPWMBASEenable bit (PWMBASE_EN), turn onPWMBASE

register map

name	Offset	bit width	type	reset value	describe
PWMBASE0					BASE: 0x400B1000
PWMBASE1					BASE: 0x400B1800
PWMBASE_EN	0x00	32	R/W	0x00	PWMBASEenable register
PWMBASE_DIV	0x04	32	R/W	0x00	PWMBASEClock Prescaler Register
PWMBASE_CON	0x08	32	R/W	0x00	PWMBASEOutput Configuration Register
PWMBASE_PERIOD	0x0C	32	R/W	0x00	PWMBASEPeriod configuration register
PWMBASE_IE	0x10	32	R/W	0x00	PWMBASEinterrupt enable register
PWMBASE_IF	0x14	32	R/W	0x00	PWMBASEInterrupt Status Register
PWMBASE_CNT	0x18	32	R/W	0xffff	PWMBASECurrent count value register
PWMBASE_CH0_COMP	0x20	32	R/W	0x00	PWMBASEaisle0Toggle Point Configuration Register
PWMBASE_CH1_COMP	0x30	32	R/W	0x00	PWMBASEaisle1Toggle Point Configuration Register
PWMBASE_CH2_COMP	0x40	32	R/W	0x00	PWMBASEaisle2Toggle Point Configuration Register

Register description

PWMBASE_ENregister(0x00)

bit field	name	type	reset value	describe
31:1	RESERVED	R	0	reserved bit



0	COUNTER_EN	R/W	0	PWMBASECounter Enable Register When this bit is configured as1, it means that the counter starts counting, CH0/CH1/CH2According to the pre-configured period value and comparison value, the corresponding channel will be generated.PWMoutput waveform. When this bit is configured as0, it means that the counter stops counting, and the output waveform level returns to the initial level after a complete counting cycle is completed.0 level status
---	------------	-----	---	--

PWMBASE_DIVregister(0x04)

bit field	name	type	Reset value	description
31:16	RESERVED	R	0	reserved bit
15:0	PWMBASE_DIV	R/W	0	PWMBASECount Clock Prescaler Register 0x0000:express1frequency division 0x0001:express2frequency division 0x0002:express3frequency division ... 0xFFFF:express65536frequency division

PWMBASE_CONregister(0x08)

bit field	name	type	reset value	describe
31:7	RESERVED	R	0	reserved bit
6	CH2_OE	R/W	0	CH2Channel waveform output enable 0: The output is off, and the pin is in a high-impedance state 1: outputCH2square wave
5	CH1_OE	R/W	0	CH1Channel waveform output enable 0: The output is off, and the pin is in a high-impedance state 1: outputCH1square wave
4	CH0_OE	R/W	0	CH0Channel waveform output enable 0: The output is off, and the pin is in a high-impedance state 1: outputCH0square wave
3	RESERVED	RO	0	reserved bit

2	CH2_OUT_INV	R/W	0	CH2output polarity flip register 0:do not change,CH2The output waveform is the original waveform 1:polarity flipped, CH2The output waveform is inverted for the original waveform
1	CH1_OUT_INV	R/W	0	CH1output polarity flip register 0:do not change,CH1The output waveform is the original waveform 1:polarity flipped, CH1The output waveform is inverted for the original waveform
0	CH0_OUT_INV	R/W	0	CH0output polarity flip register 0:do not change,CH0The output waveform is the original waveform 1:polarity flipped, CH0The output waveform is inverted for the original waveform

PWMBASE_PERIODregister(0x0C)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	PWMx_PERIOD	R/W	0	PWMxOutput Period Configuration Register The actual counting period is the period value configured for this register plus1. Note0: period cannot be configured as0. Example: configure as decimal199, then thinkPWMThe waveform period is200.

PWMBASE_INTENregister(0x10)

bit field	name	type	reset value	describe
31:4	RESERVED	R	0	reserved bit
3	POF_IE	R/W	0	Cycle overflow interrupt enable
2	CH2_COMP_IE	R/W	0	CH2Interrupt enable on reaching the tipping point
1	CH1_COMP_IE	R/W	0	CH1Interrupt enable on reaching the tipping point
0	CH0_COMP_IE	R/W	0	CH0Interrupt enable on reaching the tipping point

PWMBASE_IFregister(0x14)

bit field	name	type	reset value	describe
31:4	RESERVED	R	0	reserved bit
3	POF_IF	R/W	0	Cycle overflow interrupt status Write1clear
2	CH2_COMP_IF	R/W	0	CH2Reaching the flip point state Write1clear
1	CH1_COMP_IF	R/W	0	CH1Reaching the flip point state Write1clear
0	CH0_COMP_IF	R/W	0	CH0Reaching the flip point state Write1clear

PWMBASE_CNTregister(0x18)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	PWMBASE_CNT	R	0xffff	PWMBASECounter current count value register

PWMBASE_CH0_COMPregister(0x20)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	CH1_COMP	R/W	0	CH0Toggle Point Configuration Register Note: the count value is less than the flip point value, output1; Greater than or equal to the flip point value, output0.

PWMBASE_CH1_COMPregister(0x30)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit

15:0	CH1_COMP	R/W	0	CH1Toggle Point Configuration Register Note: the count value is less than the flip point value, output1; Greater than or equal to the flip point value, output0.
------	----------	-----	---	---

PWMBASE_CH2_COMPRegister(0x40)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	CH2_COMP	R/W	0	CH2Toggle Point Configuration Register Note: the count value is less than the flip point value, output1; Greater than or equal to the flip point value, output0.

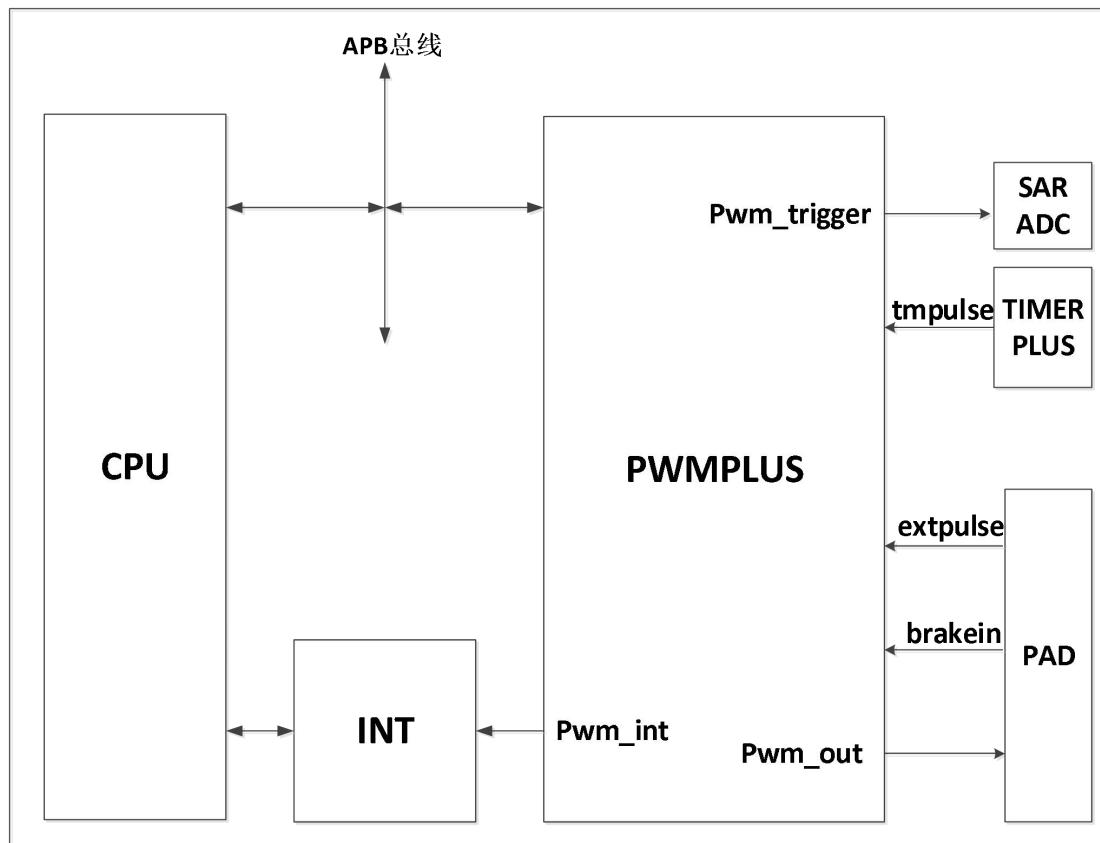
5.14 Advanced Pulse Width Modulation Generator (PWMPLUS)

5.14.1 overview

PWMPLUS for advanced PWM module, the chip's PWMPLUS support dead zone length configurable, braking, shielding,

Multiple functions such as polarity inversion, counting mode, and symmetrical mode can output flexible waveforms with different duty ratios to realize control system external devices. use PWMPLUS The module needs to be enabled before PWMPLUS clock.

Its system frame diagram is shown in the figure below:



picture5-74 PWMPLUS System Block Diagram

5.14.2 characteristic

- support 3 independent channel 16bit PWM channel output (CH0/CH1/CH2), can output different duty cycle PWM waveform
- 16bit Prescaler counter, support on-chip timer or an external signal as a count clock function

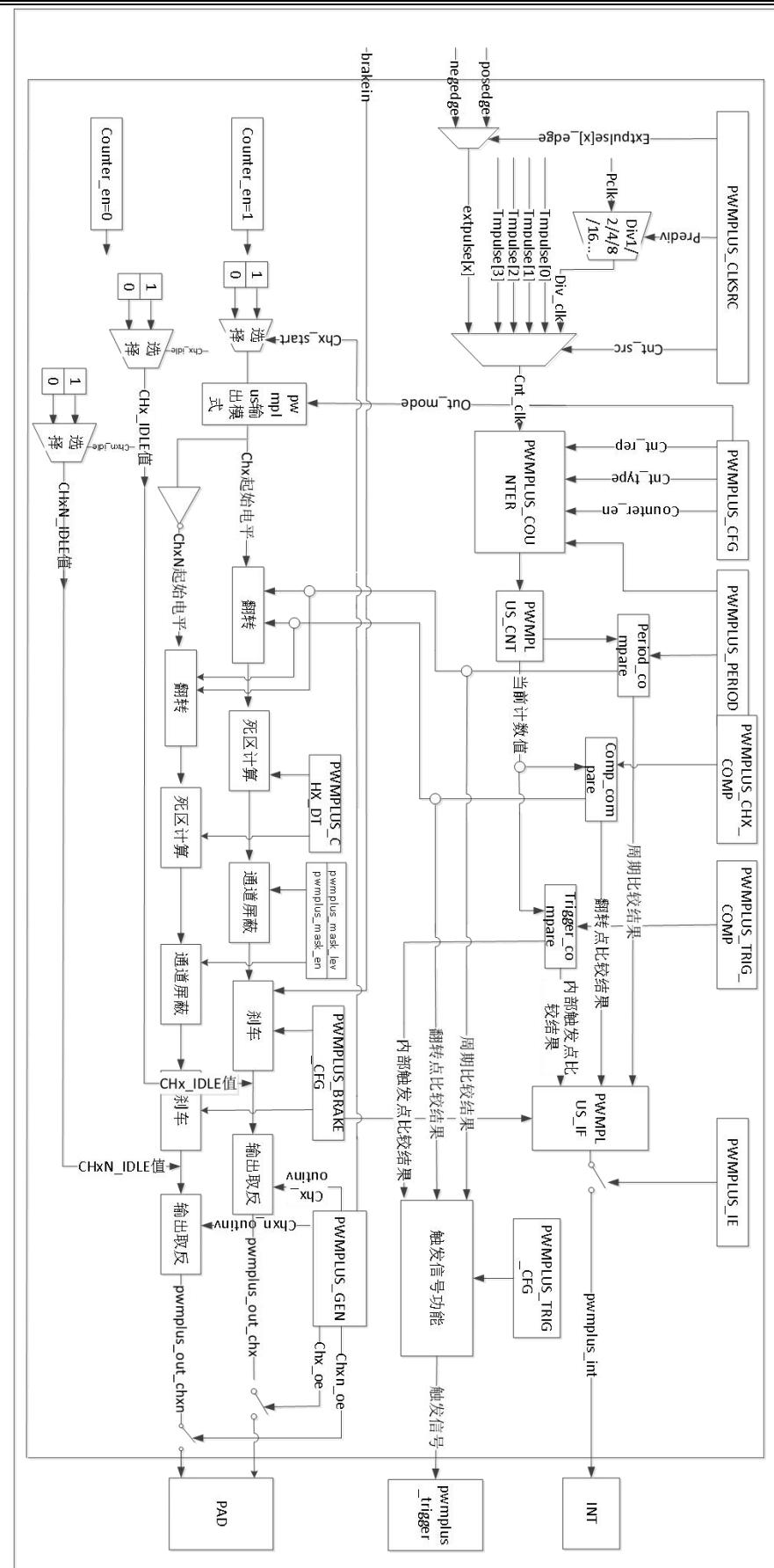
-Each channel supports configurable dead zone length

- per channel PWM Both support their own complementary output (CH0/CH0N, CH1/CH1N, CH2/CH2N)
- PWM The counter supports up counting or down counting, supports edge-aligned mode or center-aligned mode configurable
- All channels support the configuration of idle level, counting start level, and whether the output is flipped
- Support brake function, brake effective level can be configured
- Support software forced output fixed level function, level polarity can be configured
- Supports internal specific trigger mechanism, which can generate three kinds of pulses: cycle end, channel flip point and specific trigger point for external output

Signal

- Support single or loop mode selection
- Configurable to generate edge-aligned or center-aligned waveforms
- Support period value, channel flip point value, specific trigger point value fixed cycle automatic loading or software loading function

5.14.3 Block Diagram of Module Structure



picture5-75 PWMPLUSStructure diagram

As shown above isPWMPLUSThe structural frame diagram of the module, its input signal mainly includes the clockpclk, the external brake input incoming signalbrake, external input pulse signalextPLUS,internaltimerinput pulse signalmPLUS,inpclk, extPLUS,tmPLUSAll three signals can be selected asPWMPLUSCount clock usage. The output signal is mainly interrupted Signalpwmplus_int, the trigger signalpwmplus_trigger, and the output of the channel waveformpwmplus_out.

There are three options for the counting clock in this module, which are the internal system clockpclk, external input pulse signalextPLUS and insidetimerinput pulse signalmPLUS, three clocks can be registered through thepwmplus_clksrcto configure the selection, where the internal clockpclkIt can be prescaled, and the frequency division range is1-256, while the external input pulse signalextPLUSthen you can choose Select rising edge valid or falling edge valid. also,pwmplusThe output mode of the counter, the counting behavior of the counter, the counting cycle Both ring mode and count enable can be set by registerpwmplus_cfgto configure.

The channel waveform output is controlled by the channel waveform enable register, whenchx_oefor1when, toPADport output channel waveform, chx_oefor0Time directionPADThe port outputs a high-impedance state. The output of the channel waveform is divided into idle stateoutput and Channel square wave output. When the count is enabledcounter_enfor0hourpwmoutput idle stateidlevalue,idlevalue can pass pass registerpwmplus_genconfigured as1or0, you can further control whether to flip when outputting. When the count is enabled counter_enfor1hourpwmoutput channel square wave, whilePWMPLUSThe output mode can be registered through thepwmplus_cfg To configure, the initial voltage of the channel square wave can also be configured as1or0. Then, the count period value can be configured by software, flipping the The comparison between the turning point value and the current count value controls the flipping of the waveform, and the automatic loading register can also be configured to indicate the cycle overflow How many times to automatically load the period value, comparison value, dead zone value andTRIGGERvalue. Then by configuring the dead zone length, Channel masking, braking and flipping to create the desired waveform.

The generation of interrupt signal is controlled by interrupt enable and interrupt status.pwmplus_iefor1when, in phase The corresponding interrupt signal will be generated after the corresponding interrupt status is generated. When the interrupt is enabledpwmplus_iefor0, even with interrupted Status also does not generate an interrupt signal. The trigger signal is controlled by the internal trigger configuration registerpwmplus_trig_cfgcontrol, can Configure the function selection of the trigger signal, the trigger signal will be generated only after the corresponding trigger function is configured and there is a corresponding trigger state. number, and the trigger signal is a high level for one clock cycle.

This module also supports period value, channel flip point value, specific trigger point value fixed cycle automatic loading or software loading function, Among them, the automatic loading function is to automatically load the period value, comparison value, dead zone value and TRIGGERvalue, via the autoload registerauto_reloadto configure the number of cycle overflows (auto_reload+1times); soft The file loading function can be accessed viapwmplusConfiguration Register Softwareloadcontrol bits to achieve. In this modulepwm_period,

pwm_ch0_comp,pwm_ch1_comp,pwm_ch2_comp,pwm_ch0_dt,pwm_ch1_dt,pwm_ch2_dt,

trig_comphis8Each register has its own shadow register. Software writes to this bit1After that, the hardware at the end of this cycle

will this8The latest value held in a registerloadto their respective shadow registers and take effect in the next cycle. Book

The module can also read the working status of the counter through the counter working status register, and register the status of the brake input signal

The controller reads the current state of the brake input signal.

5.14.4 Functional description

Edge-Aligned Mode Output

The counter of this modulePWMThe output mode can be configured as edge-aligned mode and center-aligned mode. First, the edge

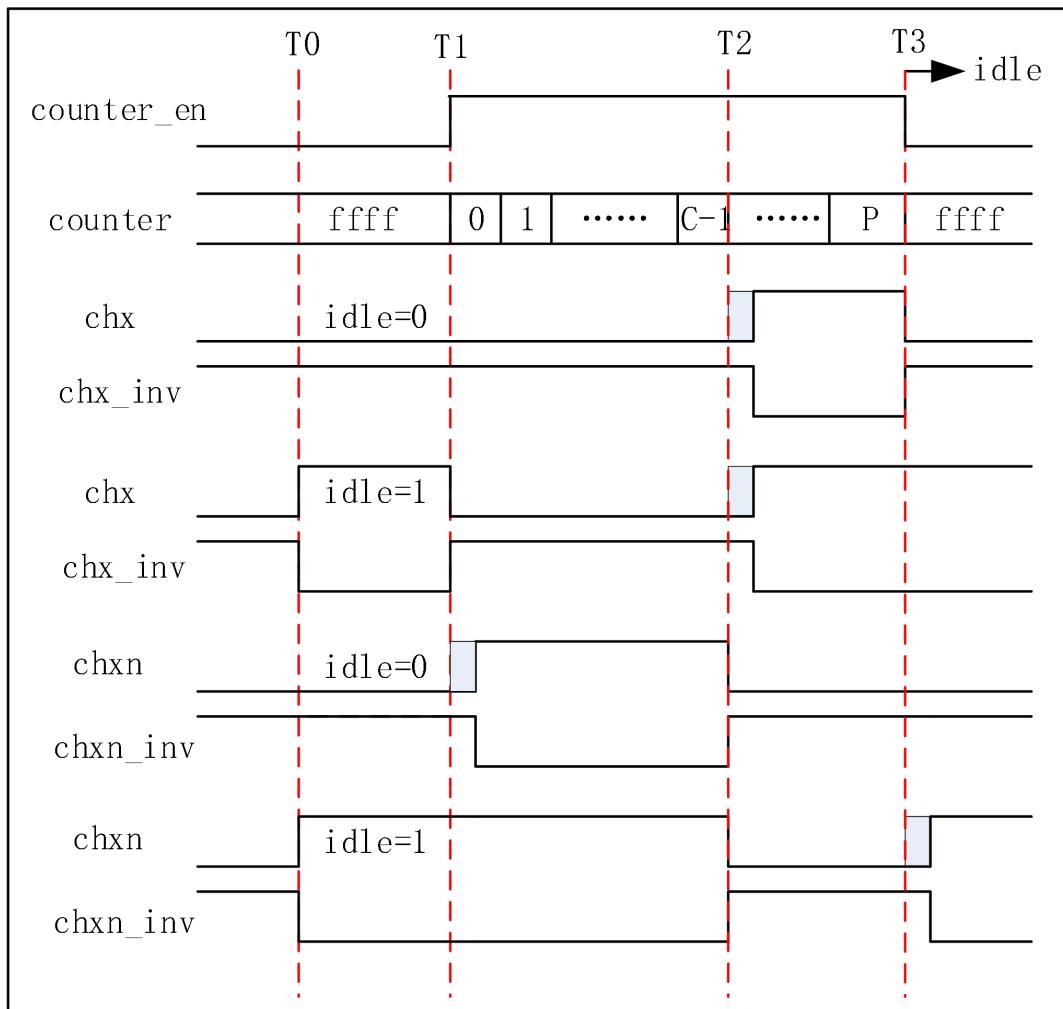
In edge-aligned output mode, the counter counts up or down, single output, circular output and different starting levels, etc.

The timing diagram is shown below.

- counter counts up

Counting up means that the counter starts counting from zero and adds one every counting clock cycle until it reaches the configured period value. the following
in the picturechx,chnx,chn_inv,chnx_invThe meanings indicated by the gray parts are as follows:chxIndicates the output status register
chx_outinvfor0hour,chnxchannelpwmplusoutput waveform,chnx_invIndicates the output status registerchx_outinv
for1hour,chnxchannelpwmplusOutput waveform. At this time, the output waveform of the channel is opposite to the original waveform of the channel.chxn
andchnx_invIndicates its complementary output. The gray part in the figure represents the dead zone, and the length of the dead zone can be configured.

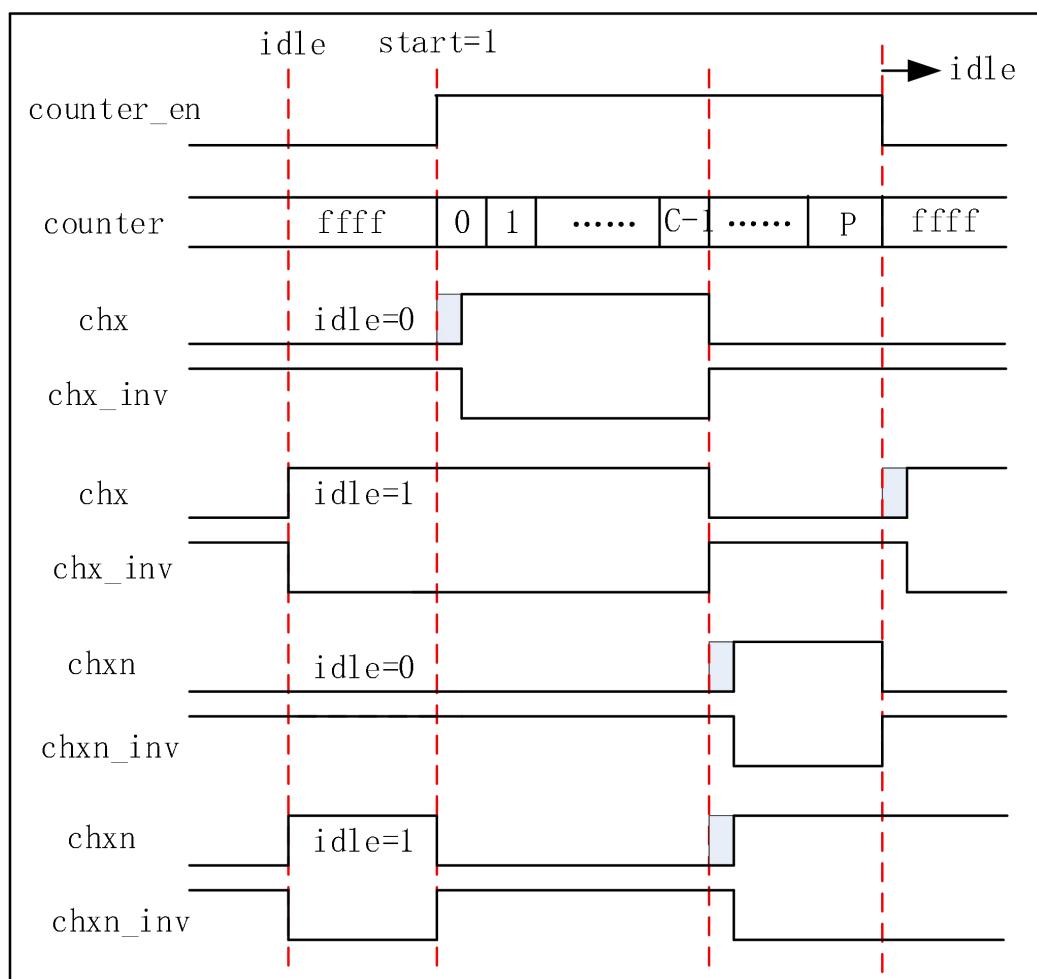
1, a single output,start=0Case



picture5-76 PWMPLUSIn edge-aligned mode, count up, single output, the initial level is 0Timing diagram of

As shown in the figure above, when the count is enabled `counter_en` before enabling (T1 time before), `pwmplusOutputIDLE` value when T0 time change `IDLE` value up `pwmplusOutput` follows `IDLE` changes in value, when the T1 Always open the counter count enable, `pwmplusOutput` channel waveform. After configuring the corresponding interrupt enable, the T2A flip point interrupt is generated at all times, exist T3A cycle overflow interrupt is generated at any time.

2, a single output,start=1Case

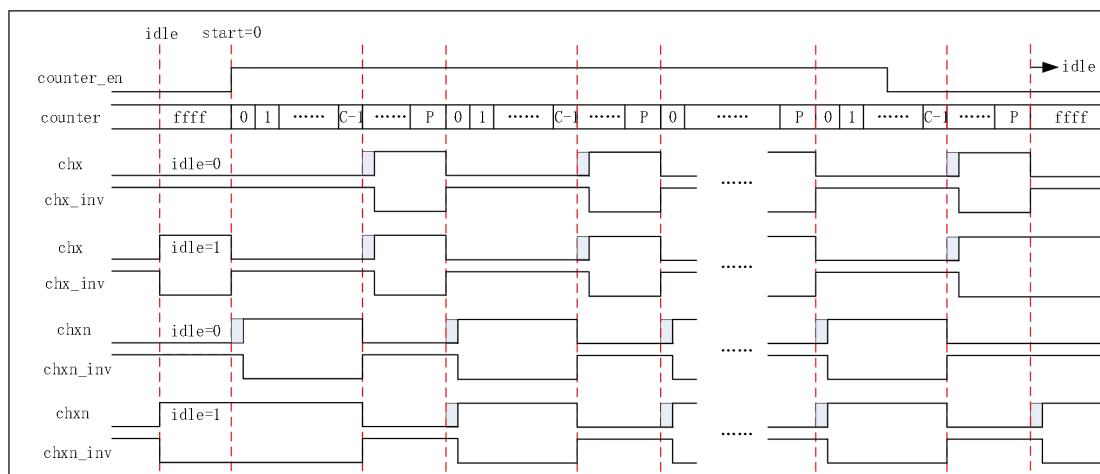


picture5-77 PWMPLUSIn edge-aligned mode, count up, single output, the initial level is1Timing diagram of

As shown in the figure above, configure the flip point interrupt enable and the cycle end interrupt enable to generate a flip point at the third vertical line

Interrupt, an end-of-cycle interrupt is generated at the fourth vertical bar.

3, loop output,start=0Case

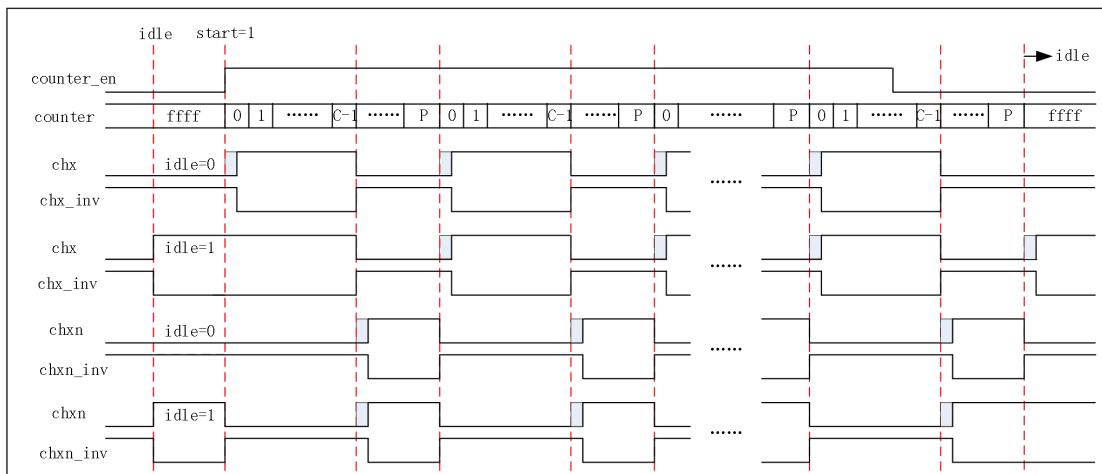


picture5-78 PWMPLUSIn edge-aligned mode, count up and output circularly, the initial level is0Timing diagram of

As shown in the figure above, the configuration of the toggle point interrupt enable and the cycle end interrupt enable can be performed in each "C-1" The location can be

A toggle point interrupt is generated at each "P" Any position can generate an end-of-cycle interrupt.

4, loop output,start=1Case



picture5-79 PWMPLUSIn edge-aligned mode, count up and output circularly, the initial level is1Timing diagram of

As shown in the figure above, the configuration of the toggle point interrupt enable and the cycle end interrupt enable can be performed in each "C-1" The location can be

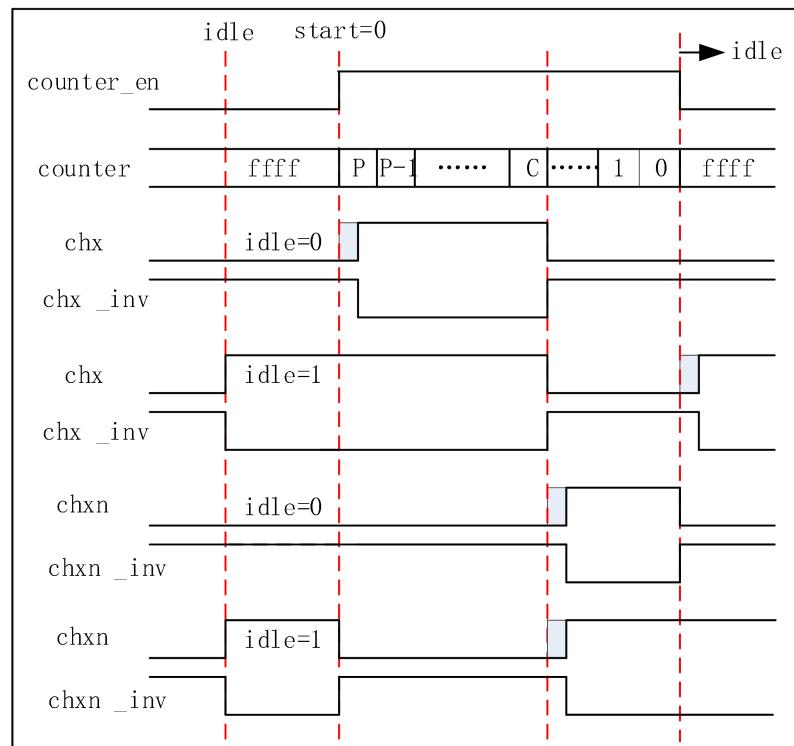
A toggle point interrupt is generated at each "P" Any position can generate an end-of-cycle interrupt.

- counter counts down

Counting down means that the counter starts counting from the configured period value, and decreases by one every count clock cycle until it reaches zero. by

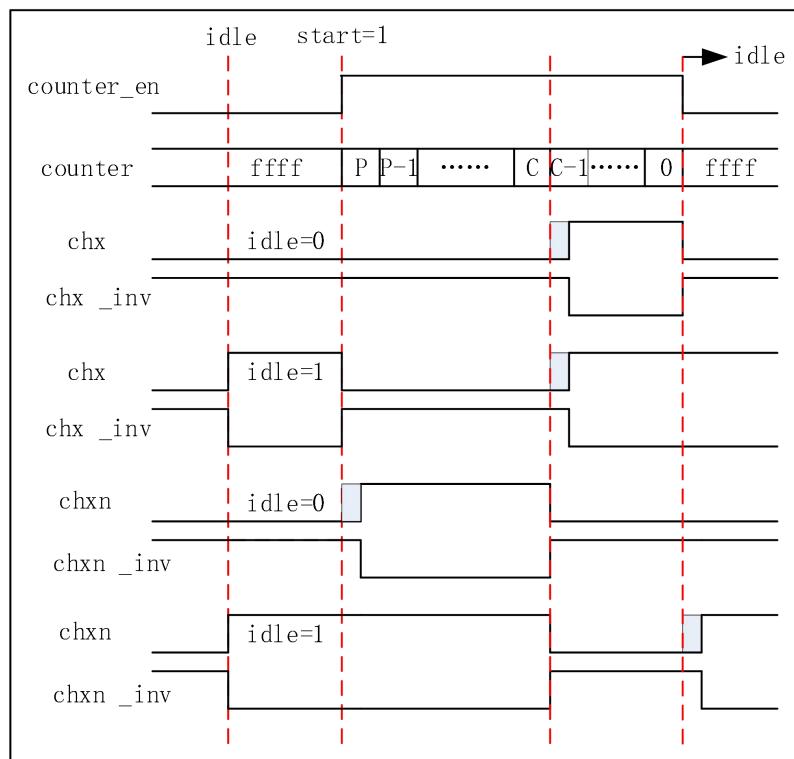
The figure below chx, chxn, chx_inv, chxn_inv. The meanings indicated by the gray parts are as follows: chx indicates the output status register device chx_outinv for 0 hour, chx channel pwm output waveform, chx_inv indicates the output status register chx_outinv for 1 hour, chx channel pwmplus output waveform. At this time, the output waveform of the channel is opposite to the original waveform of the channel. chxn and chxn_inv indicates its complementary output. The gray part in the figure represents the dead zone, and the length of the dead zone can be configured.

1, a single output,start=0Case



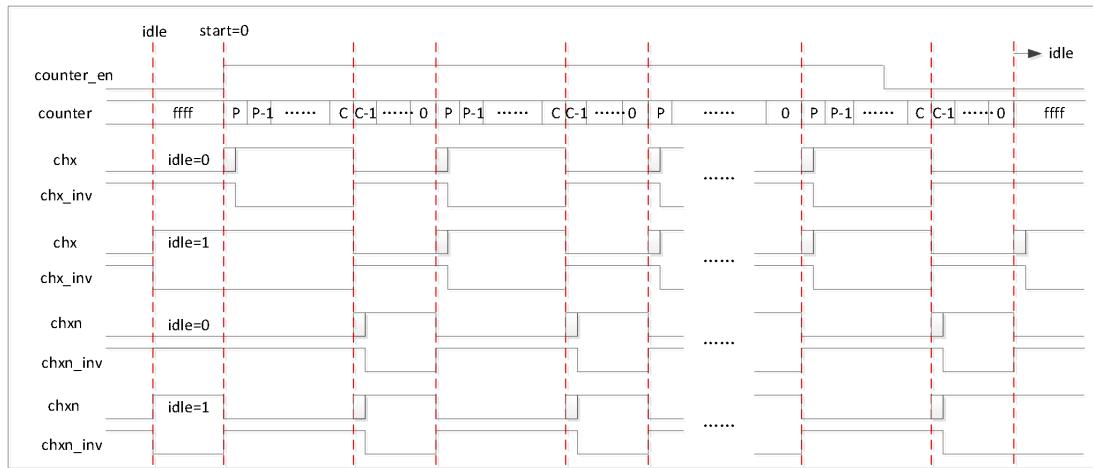
picture5-80 PWMPLUSIn edge-aligned mode, count down, single output, the initial level is0Timing diagram of

2, a single output,start=1Case



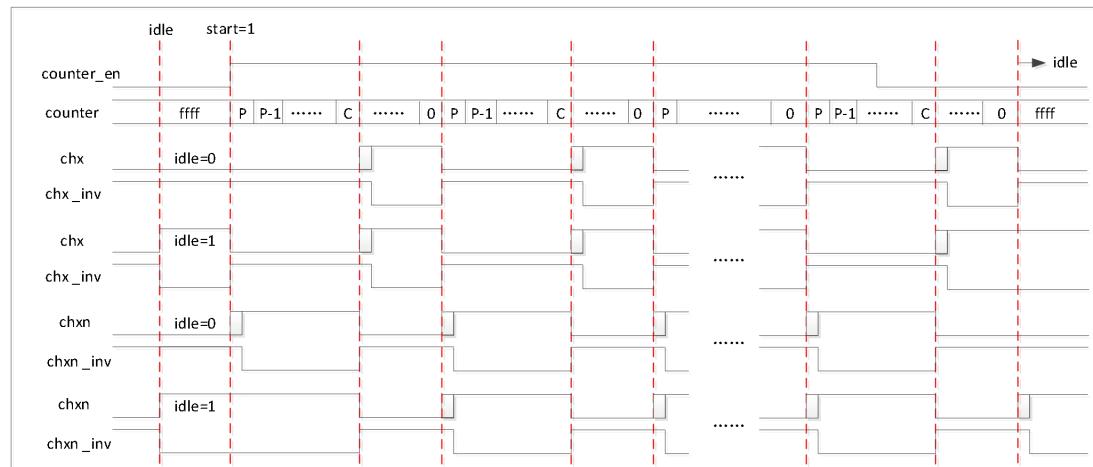
picture5-81 PWMPLUSIn edge-aligned mode, count down, single output, the initial level is1Timing diagram of

3, loop output,start=0Case



picture5-82 PWMPLUSIn edge-aligned mode, count down and output circularly, the initial level is0Timing diagram of

4, loop output,start=1Case



picture5-83 PWMPLUSIn edge-aligned mode, count down and output circularly, the initial level is1Timing diagram of

Center-aligned mode output

In center-aligned mode, the counter counts up or down, one-shot output, loop output and different start levels, etc.

The timing diagram of the situation is as follows:

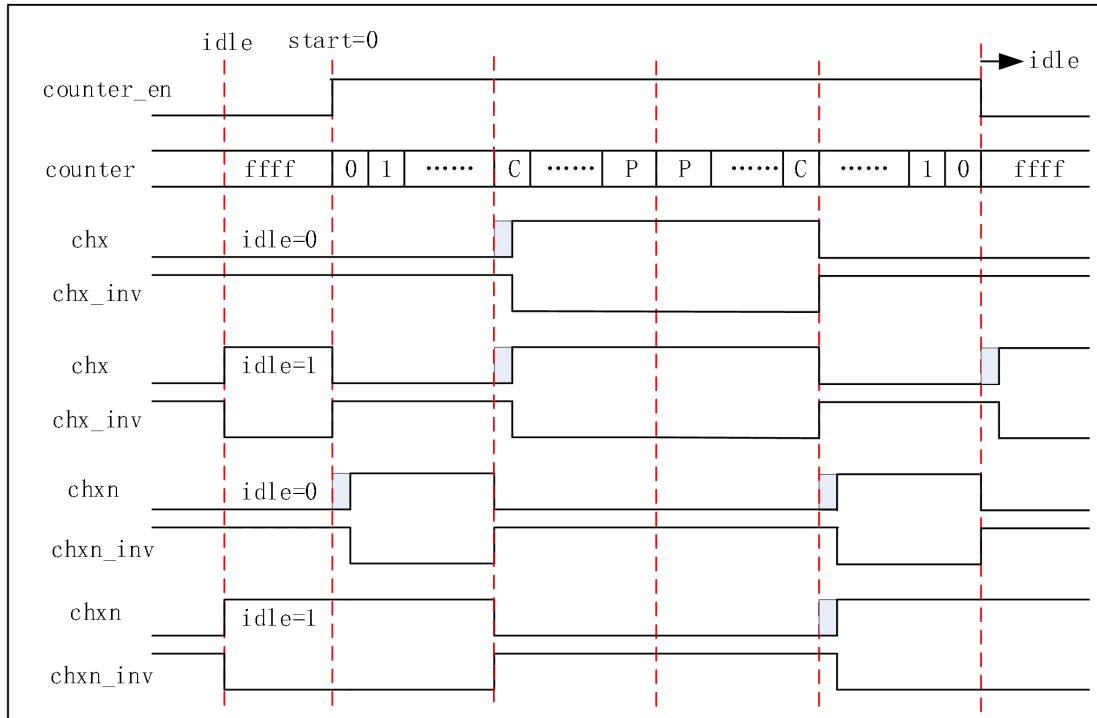
- counter counts up

Counting up means that the counter starts counting from zero and adds one every counting clock cycle until it reaches the configured period value. by

The figure below chx, chxn, chx_inv, chxn_inv The meanings indicated by the gray parts are as follows: chx Indicates the output status register

device chx_outinv for 0 hour, chx channel pwmplus output waveform, chx_inv indicates the output status register chx_outinv for 1 hour, chx channel pwmplus output waveform. At this time, the output waveform of the channel is opposite to the original waveform of the channel. chxn and chxn_inv indicates its complementary output. The gray part in the figure represents the dead zone, and the length of the dead zone can be configured

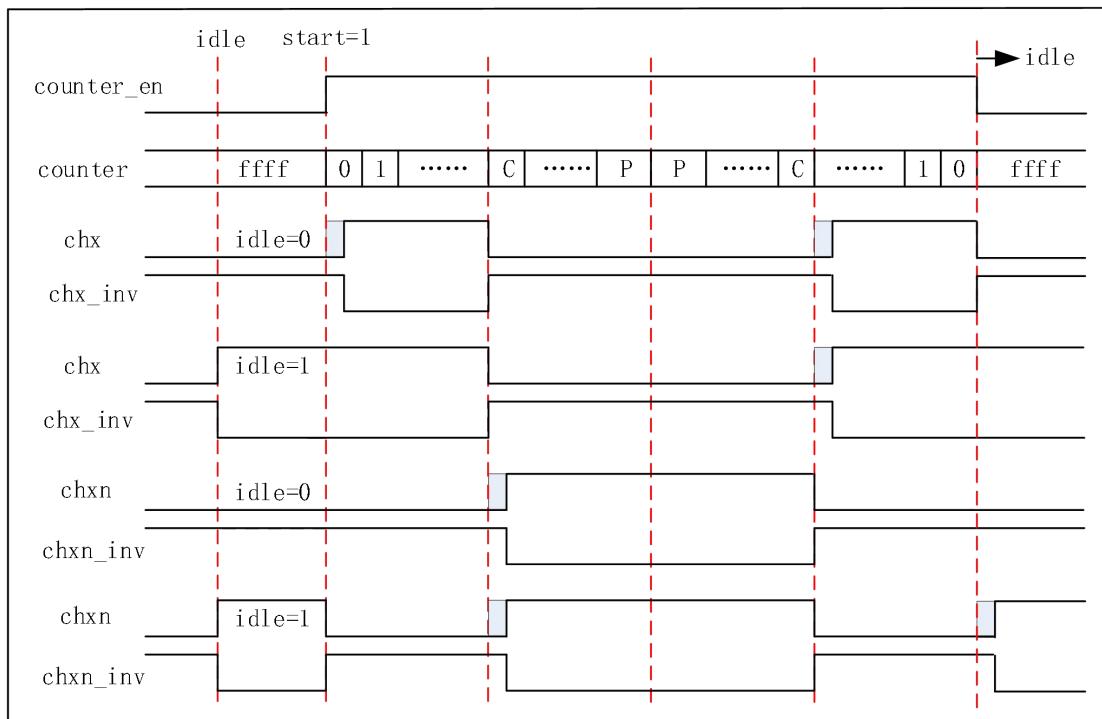
1, a single output,start=0 Case



picture5-84 PWMPLUS in center-aligned mode, count up, single output, the initial level is 0 Timing diagram of

As can be seen from the figure above, in center-aligned mode pwmplus The output waveform is pThe position where the point ends is center-aligned, The first half cycle is counting up, and the second half cycle is counting down. Configurable to generate rollover point interrupts at third and fifth vertical bars, End-of-cycle interrupts are generated at the fourth and sixth vertical bars.

2, a single output,start=1 Case

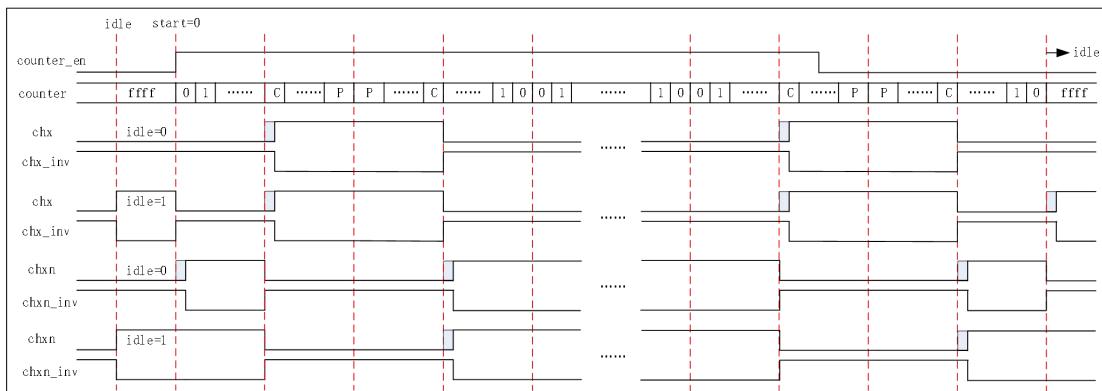


picture5-85 PWMPLUSIn center-aligned mode, count up, single output, the initial level is1Timing diagram of

As shown in the figure above, configuring the toggle point interrupt enable and the cycle end interrupt enable can generate toggle at the third and fifth vertical bars.

Turn point interrupts, end-of-cycle interrupts are generated at the fourth and sixth vertical bars.

3, loop output,start=0Case

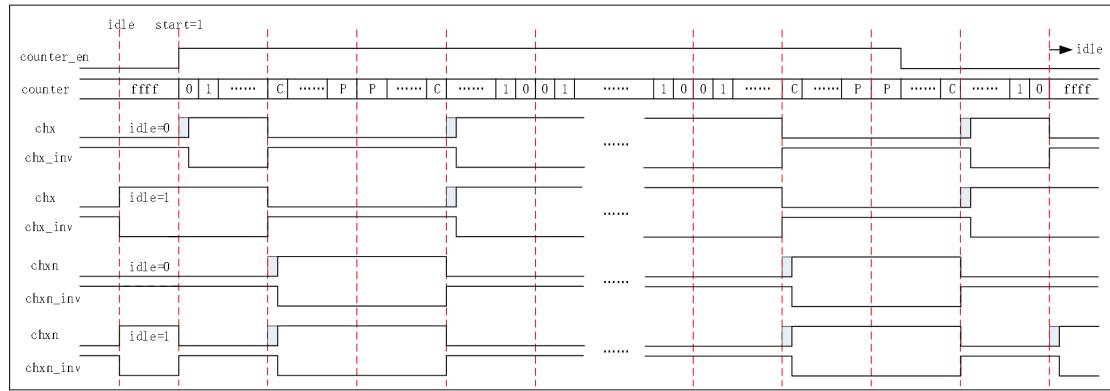


picture5-86 PWMPLUSIn center-aligned mode, count up and output circularly, the initial level is0Timing diagram of

As shown in the figure above, the configuration of the toggle point interrupt enable and the cycle end interrupt enable can be performed in each "C" can be produced in any position

The raw flip point is interrupted at every two "P" position or every two "0" Any position can generate an end-of-cycle interrupt.

4, loop output,start=1Case



picture5-87 PWMPLUSIn center-aligned mode, count up and output circularly, the initial level is0Timing diagram of

As shown in the figure above, configure the toggle point interrupt enable and cycle end interrupt enable can be configured in each "C"position can be generated

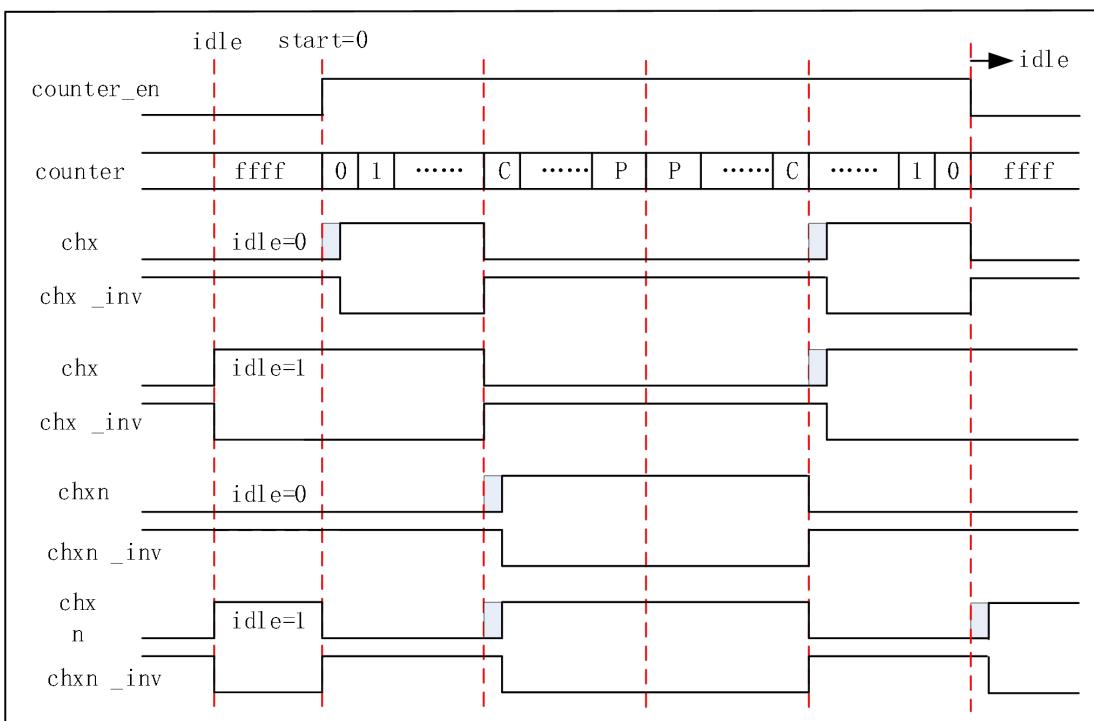
flip point breaks at every two"P"position or every two "0"Any position can generate an end-of-cycle interrupt.

- counter counts down

Counting down means that the counter starts counting from the configured period value, and decreases by one every count clock cycle until it reaches zero. by

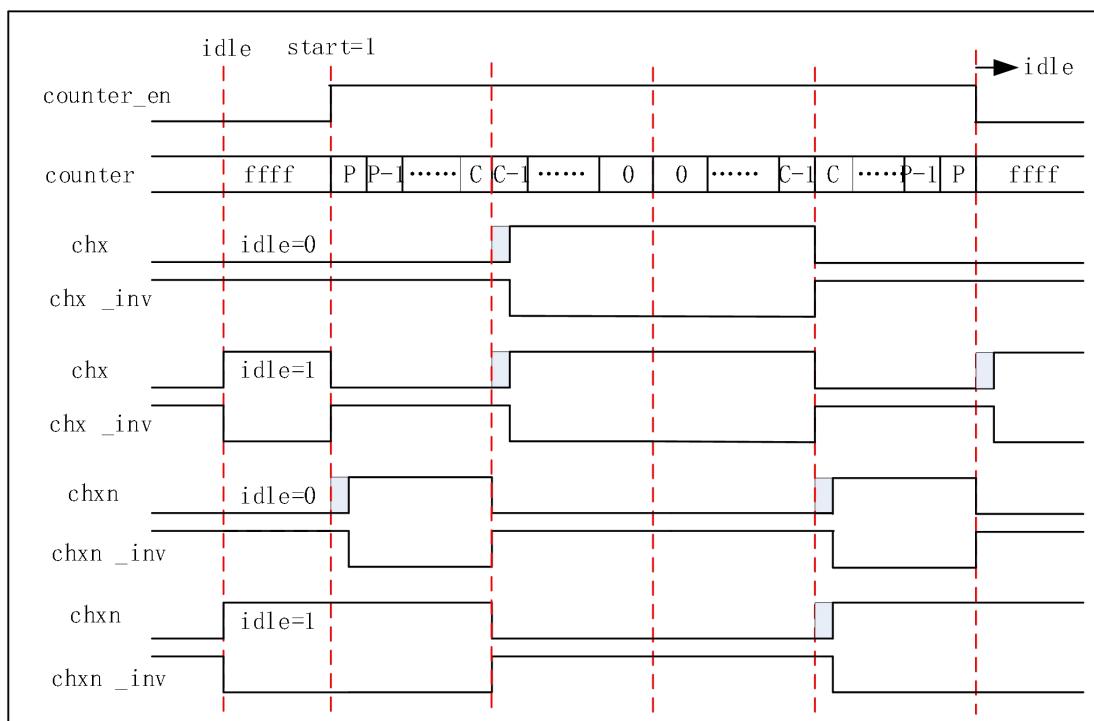
The figure belowchx,chxn,chx_inv,chxn_invThe meanings indicated by the gray parts are as follows:chxIndicates the output status register devicechx_outinvfor0hour,chxchannelpwmplusoutput waveform, chx_invIndicates the output status registerchx_outinv for1hour,chxchannelpwmplusOutput waveform. At this time, the output waveform of the channel is opposite to the original waveform of the channel.chxn andchxn_invIndicates its complementary output. The gray part in the figure represents the dead zone, and the length of the dead zone can be configured

1, a single output,start=0Case



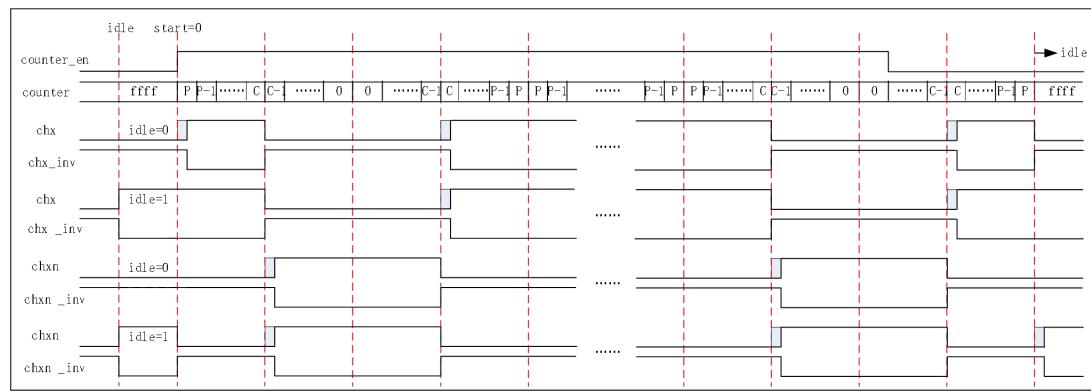
picture5-88 PWMPLUSIn center-aligned mode, count down, single output, the initial level is0Timing diagram of

2, a single output,start=1Case



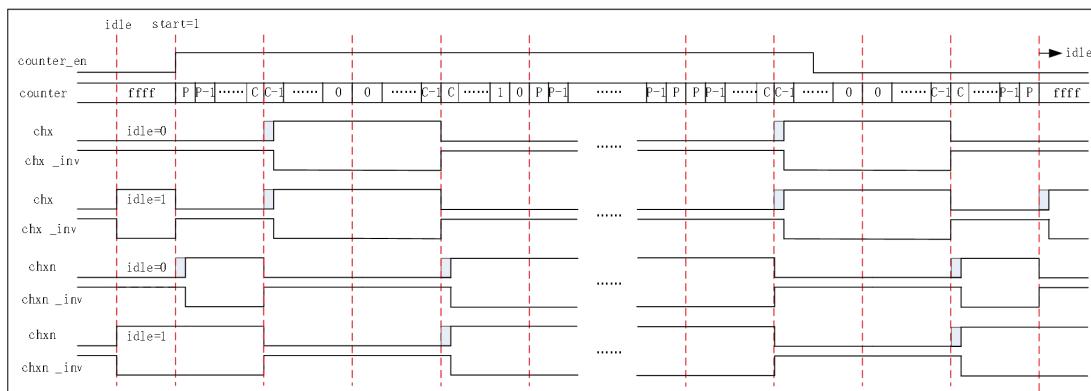
picture5-89 PWMPLUSIn center-aligned mode, count down, single output, the initial level is1Timing diagram of

3, loop output,start=0Case



picture5-90 PWMPLUSIn center-aligned mode, count down and output circularly, the initial level is0Timing diagram of

4, loop output,start=1Case

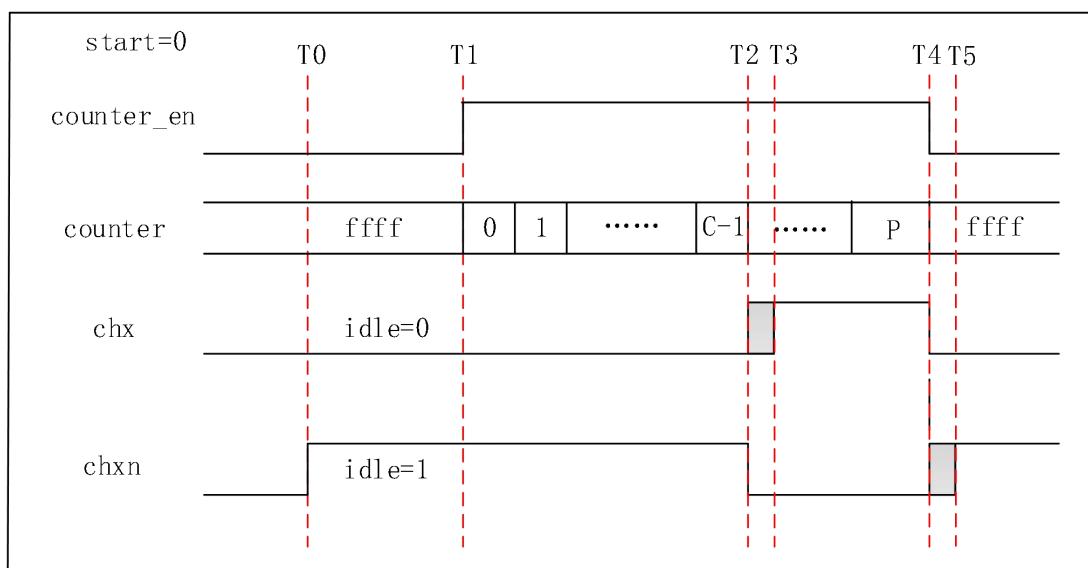


picture5-91 PWMPLUSIn center-aligned mode, count down and output circularly, the initial level is1Timing diagram of

Complementary Outputs with Dead Time Insertion

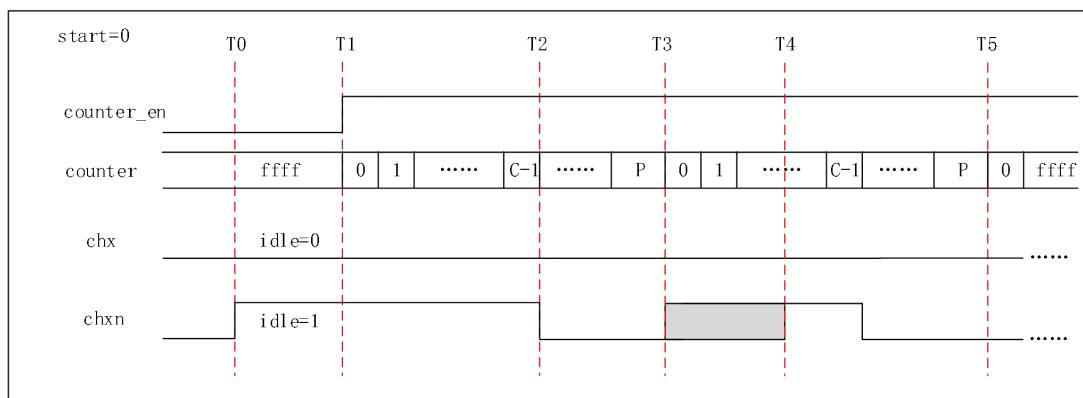
PWMPLUSCapable of outputting two complementary signals, and capable of managing momentary turn-off and turn-on of the output, during which time the

Often referred to as the dead zone, the user should determine which output devices are connected and their characteristics (delay of level transitions, power switching delay, etc.) to adjust the dead time.



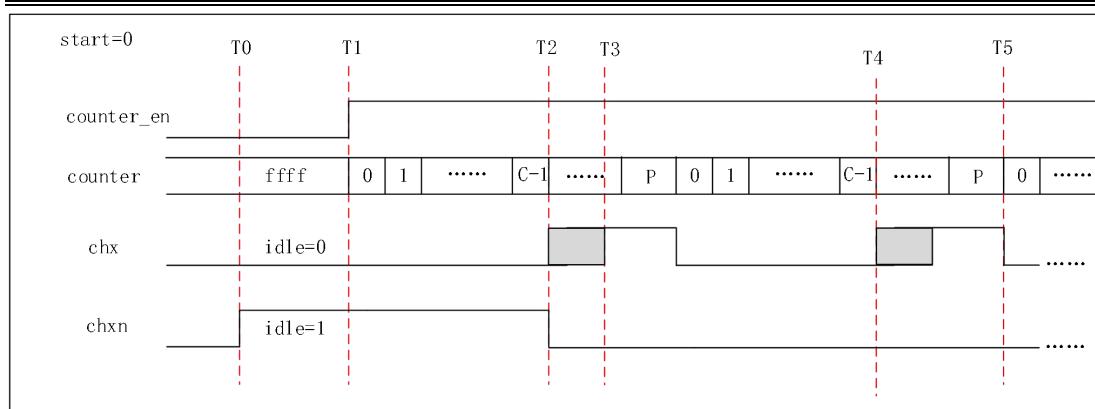
picture5-92 PWMPLUSComplementary Output Timing Diagram with Dead Time Insertion

Pictured above is PWMPLUSComplementary output with dead-time insertion, where chxn for chx The complementary output of , in the figure T2-T3 and T4-T5 The gray part between is the dead zone, the length of the dead zone can be passed CHx_DT register to configure, when configuring the deadband length need to be compared with the period value, CHx_START value, CHx_COMP value, otherwise the output waveform may not be as expected.



picture5-93 PWMPLUSDead zone length is greater than the positive pulse and smaller than the negative pulse timing diagram

Pictured above is PWMPLUSThe starting level is 0When the configured dead zone length is greater than the timing diagram of the positive pulse and smaller than the negative pulse, in the picture T3-T4The gray part is the dead zone length.



picture5-94 PWMPLUSDead zone length is greater than the negative pulse and smaller than the positive pulse timing diagram

Pictured above is PWMPLUSThe starting level is 0When the configured dead zone length is greater than the timing diagram of the positive pulse and smaller than the negative pulse,

in the pictureT2-T3The gray part is the dead zone length.

output under braking

This module supports different channels to select different braking signals, the effective braking level can be configured, and the level value of each channel during braking

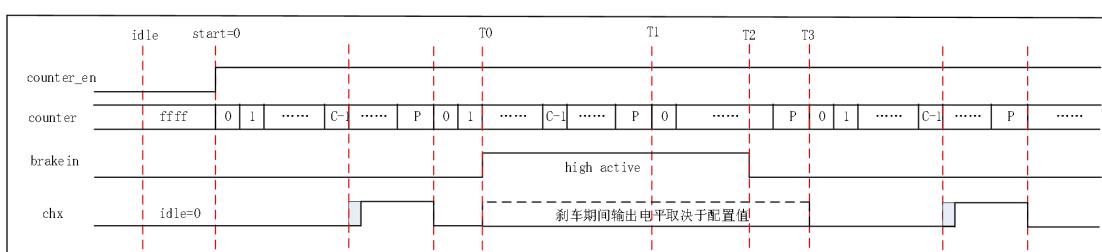
Configurable. During braking, the counter counts normally and is not affected by braking.

After the brake signal arrives, the output signal of the channel will be effective immediately. After the brake is cancelled, it will be output at the beginning of the next cycle.

normal waveform.

Count up with the counter, the initial level is 0Example when:

1, edge-aligned mode



picture5-95 PWMPLUSTiming Diagram of Braking in Edge-Aligned Mode

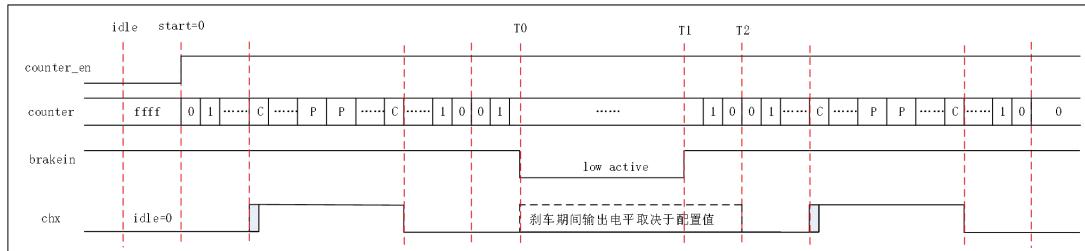
As shown in the figure above, in the edge-aligned mode, taking the active high brake level as an example, inT0Brake level input at all times

When it becomes high, the signal is valid immediately, and enters the braking period. During the braking period, each channel does not output normal waveforms, as shown in the figureT1hour

The output waveform will not be reversed at the moment, and the channel waveform output depends on the configuration value; whenT2After the end of moment braking, the channel waveform

does not return to normal immediately, but starts in the next cycle (T3moment) will output the normal waveform.

2, center-aligned mode



picture5-96 PWMPLUSTiming diagram of braking in center-aligned mode

As shown in the figure above, in the center-aligned mode, taking the low effective brake level as an example, inT0Brake level input at all times

When it becomes low, the signal is valid immediately and enters the braking period. During the braking period, each channel does not output normal waveforms, but channel waveforms output depends on the configuration value; whenT1After the end of the time brake, the channel waveform did not return to normal immediately, but in the next week period starts (T2moment) will output the normal waveform.

In addition, this chip also supports the brake filter function, which can control the digital filter of the brake signal, and can perform2,4and8individual

Internal prescaler clock filtering through registerBRAKE_FILTERcontrol.

MASKcase output

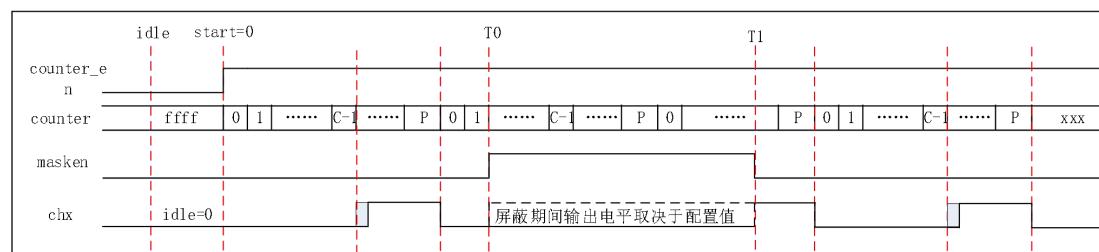
This module supports independentMASKFunction configuration, each channelMASKThe period level value is configurable.MASKExpect

During this period, the counter counts normally without being affected.

MASKAfter the function is valid, the channel output signal is valid immediately,MASKAfter the function is canceled, the normal waveform will be output immediately.

Count up with the counter, the initial level is0Example when:

1, edge-aligned mode



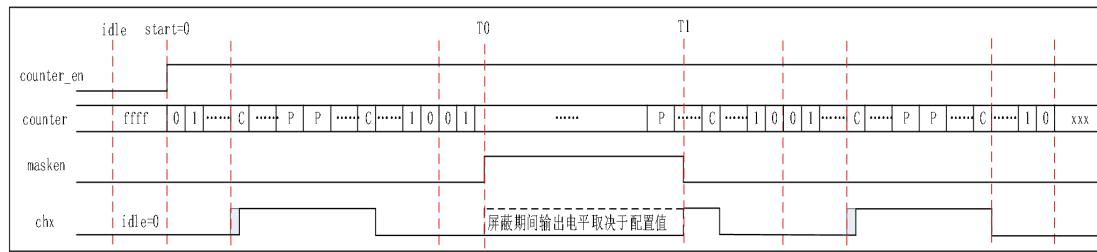
picture5-97 PWMPLUSEdge-aligned modeMASKSituation sequence diagram

As shown in the figure above, in edge-aligned mode, theT0The shielding enable becomes high at all times, and the shielding enable is immediately effective.

During the shielding period, each channel does not output normal waveforms during the shielding period, and the channel waveform output depends on the configuration value; whenT1hour

After the masking is completed, the channel waveform returns to normal immediately, and the normal waveform is output.

2, center-aligned mode



picture5-98 PWMPLUSCenter Alignment ModeMASKSituation sequence diagram

As shown in the figure above, in center-aligned mode, the T0The shielding enable becomes high at all times, and the shielding enable is immediately effective.

During the shielding period, each channel does not output normal waveforms during the shielding period, and the channel waveform output depends on the configuration value; when T1hour

After the masking is completed, the channel waveform returns to normal immediately, and the normal waveform is output.

Period value and rollover point

Example of configuring the relationship between period value and flip point value (IDLEfor0):

user wants to generatePWMPLUSThe waveform is: period8, the counting start level is0, the duty cycle is75%, you need

To configure as:PERIOD=0x7,CHx_COMP=0x2.

user wants to generatePWMPLUSThe waveform is: period8, the counting start level is0, the duty cycle is12.5%, you need

To configure as:PERIOD=0x7,CHx_COMP=0x7.

user wants to generatePWMPLUSThe waveform is: period8, the counting start level is0, the duty cycle is0%, you need

Configured as:PERIOD=0x7,CHx_COMP=0x8(more than the7can be).

user wants to generatePWMPLUSThe waveform is: period8, the counting start level is0, the duty cycle is100%, you need

To configure as:PERIOD=0x7,CHx_COMP=0x0.

user wants to generatePWMPLUSThe waveform is: period8, the counting start level is1, the duty cycle is25%, you need

To configure as:PERIOD=0x7,CHx_COMP=0x2.

user wants to generatePWMPLUSThe waveform is: period8, the counting start level is1, the duty cycle is87.5%, you need

To configure as:PERIOD=0x7,CHx_COMP=0x7.

user wants to generatePWMPLUSThe waveform is: period8, the counting start level is1, the duty cycle is100%, you need

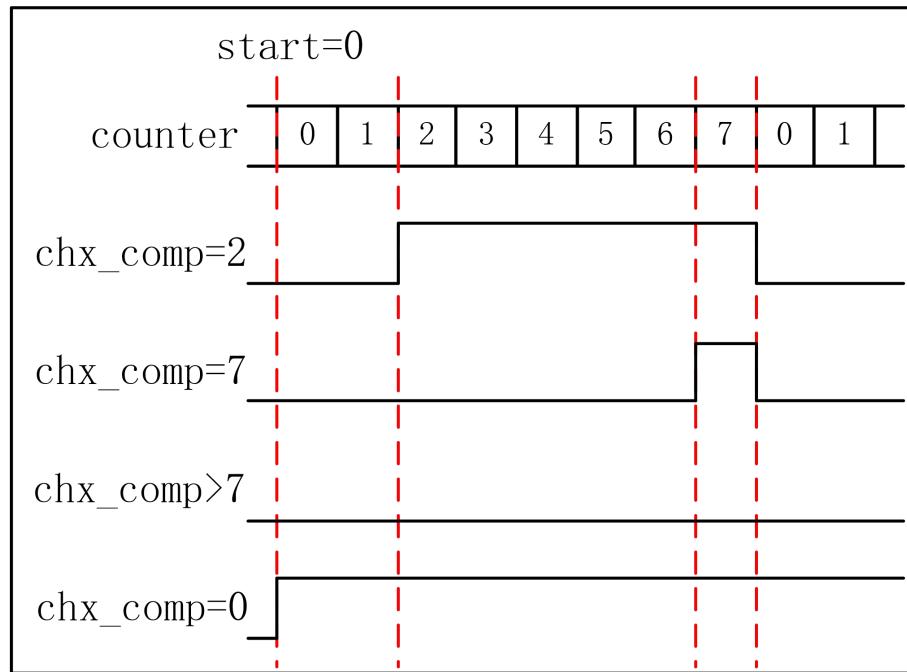
To configure as:PERIOD=0x7,CHx_COMP=0x8(more than the7ok) .

user wants to generatePWMPLUSThe waveform is: period8, the counting start level is1, the duty cycle is0%, you need

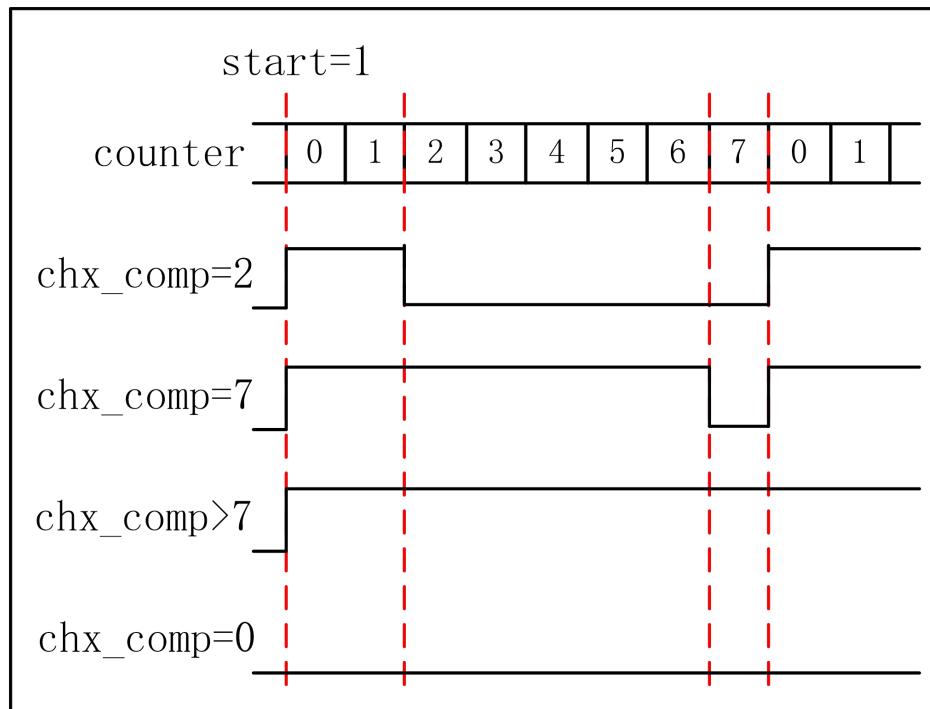
Configured as:PERIOD=0x7,CHx_COMP=0x0.

In the following schematic example, the period value is configured as 7.

1, Edge-aligned mode, counting up:

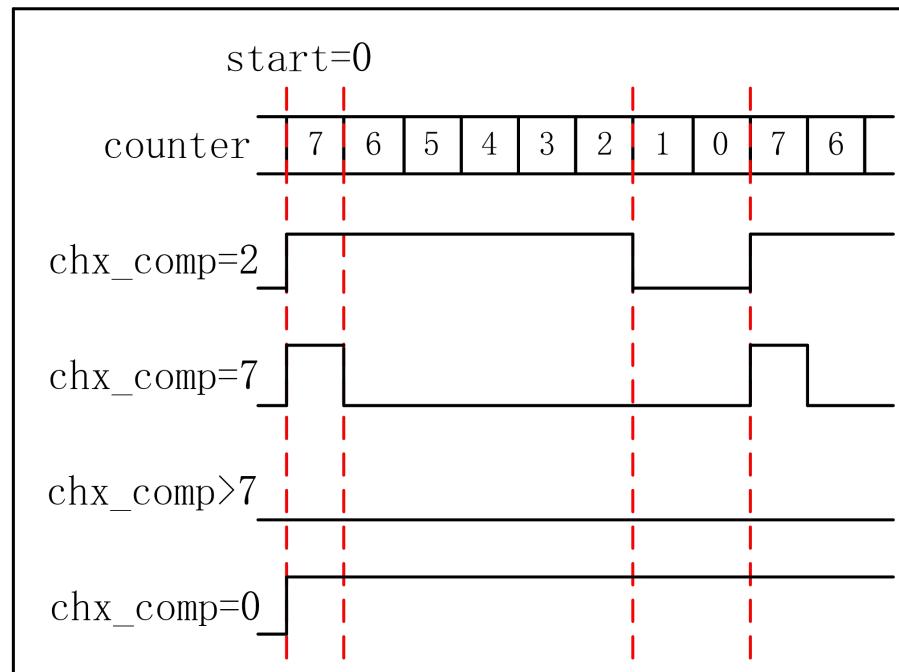


picture5-99 PWMPLUSEdge-aligned mode, count up, the starting level is0Case flip point and cycle timing diagram

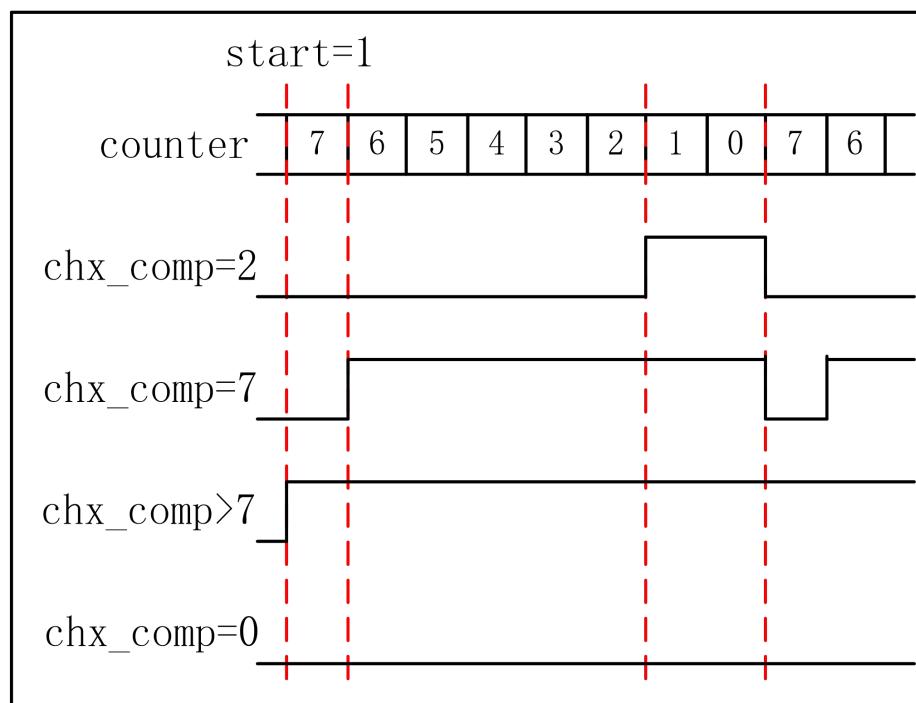


picture5-100 PWMPLUSEdge-aligned mode, count up, the starting level is1Case flip point and cycle timing diagram

2, Edge-aligned mode, counting down:

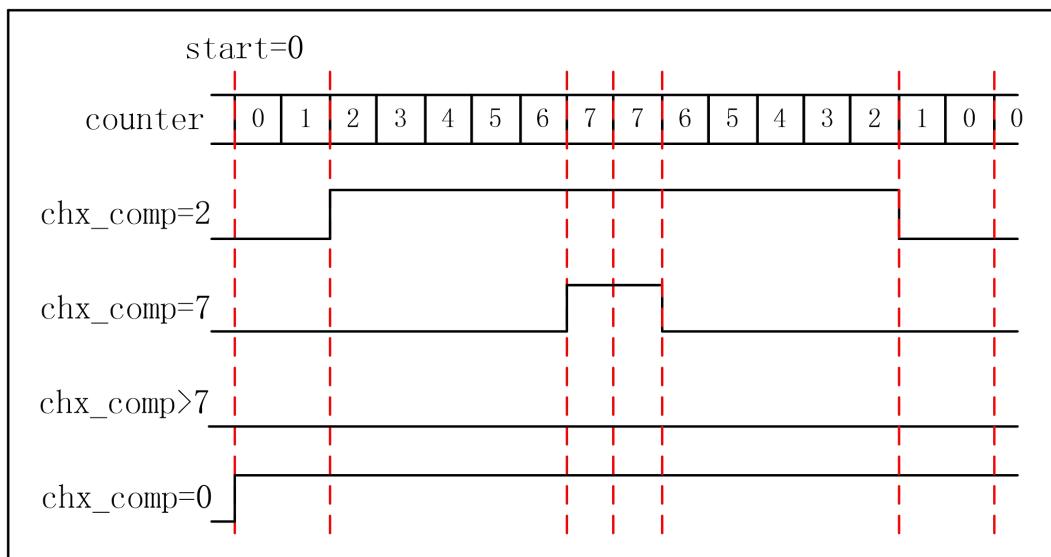


picture5-101 PWMPLUSEdge-aligned mode, counting down, the starting level is0Case flip point and cycle timing diagram

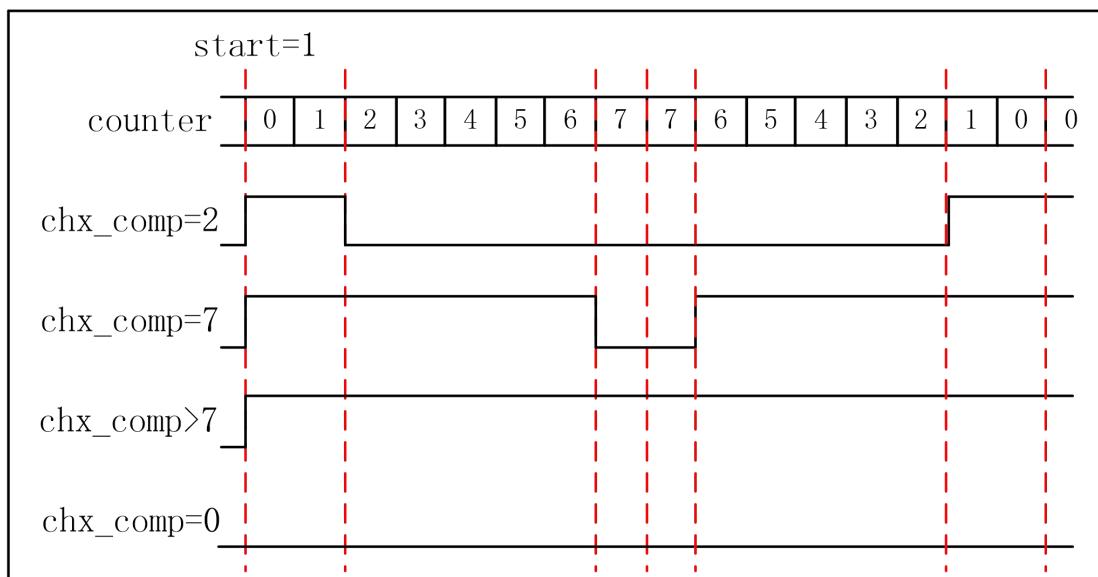


picture5-102 PWMPLUSEdge-aligned mode, counting down, the starting level is1Case flip point and cycle timing diagram

3, center-aligned mode, counting up:

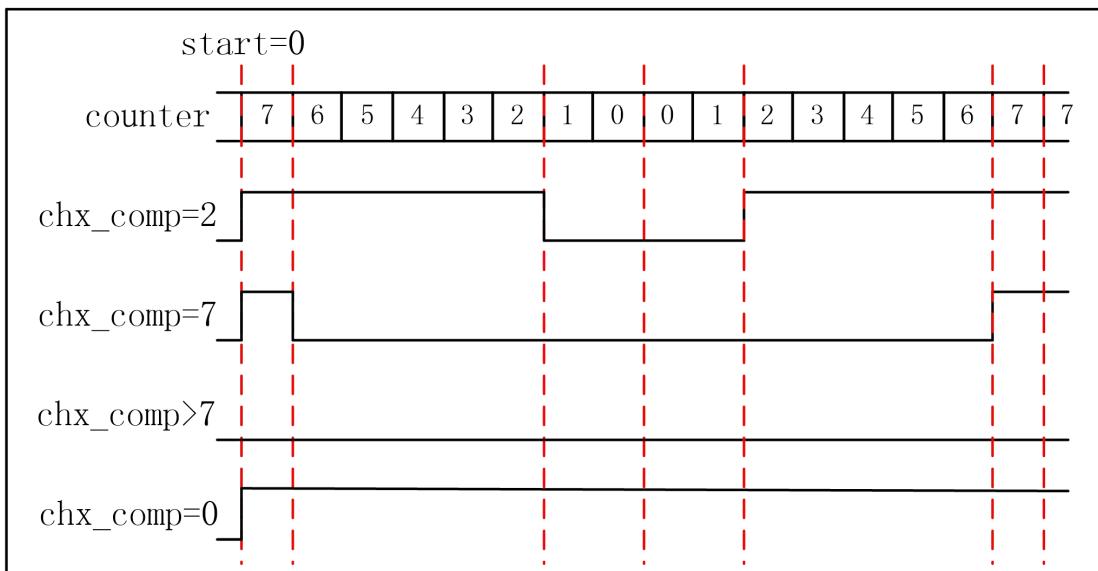


picture5-103 PWMPLUSCenter-aligned mode, count up, the starting level is0Case flip point and cycle timing diagram

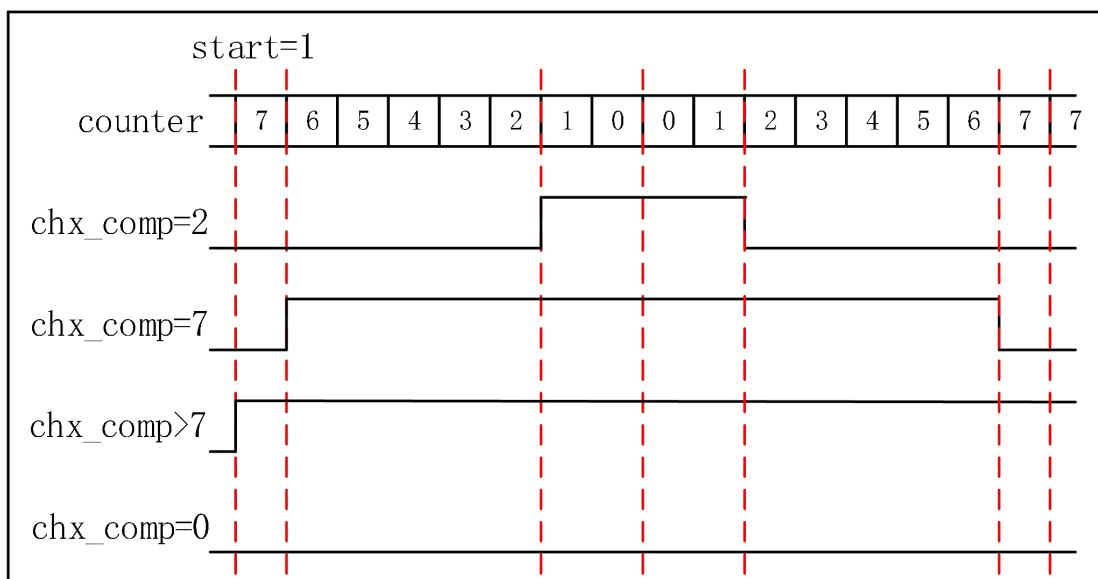


picture5-104 PWMPLUSCenter-aligned mode, count up, the starting level is1Case flip point and cycle timing diagram

4, center-aligned mode, counting down:



picture5-105 PWMPLUSCenter-aligned mode, counting down, the starting level is0Case flip point and cycle timing diagram

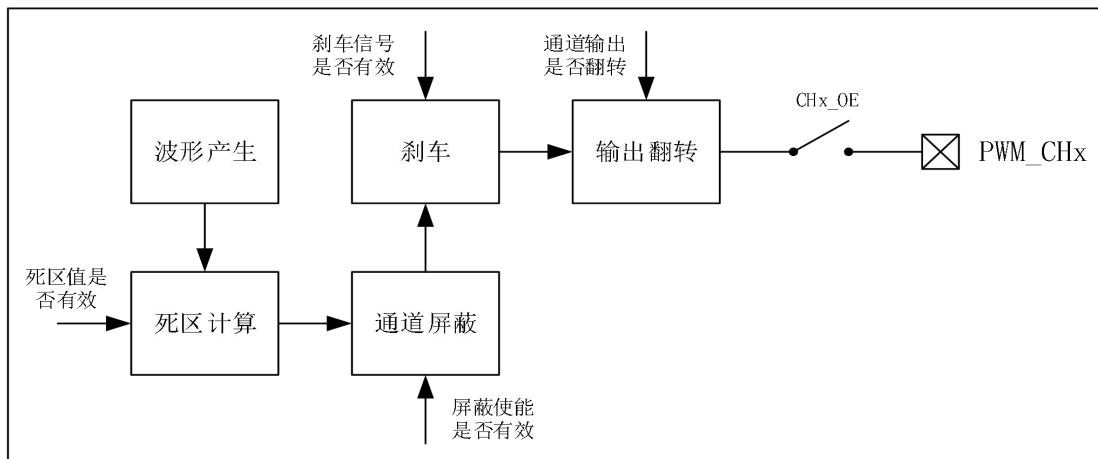


picture5-106 PWMPLUSCenter-aligned mode, counting down, the starting level is1Case flip point and cycle timing diagram

Output Priority Relations

The five conditions of dead zone value calculation, channel masking, brake signal, output reversal, and output enable will affect each channel PWM

Waveform output, the output waveform priority relationship is as follows:



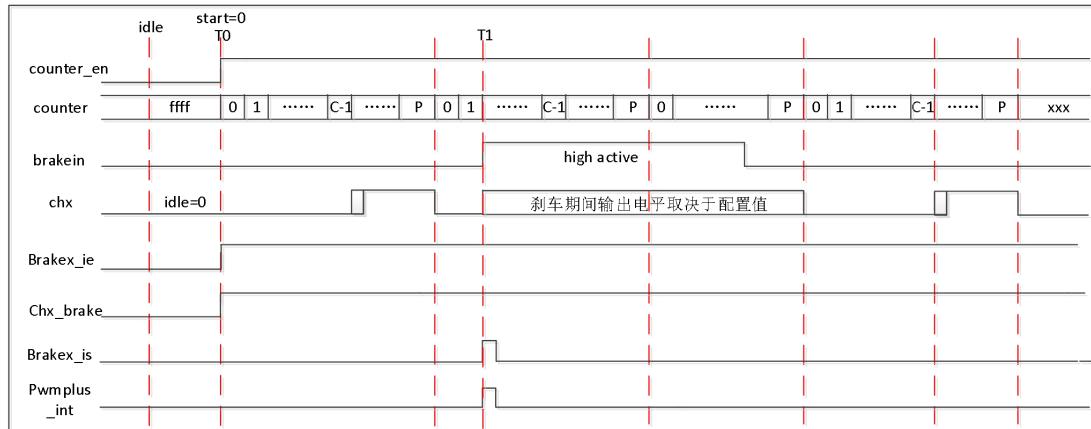
picture5-107 PWMPLUS output priority map

Trigger Signals and Interrupts

This module can realize the trigger signal of the three situations of the three channel flip points, the end of the period and the specific trigger point. The trigger signal for individualclkThe high level of the clock; it can also realize the flip point, period overflow, and characteristic trigger point in different counting modes. And the generation of the brake interrupt signal, in which the specific trigger point can only generate the corresponding trigger state and interrupt state, and will not change ChangePWMoutput waveform. The following figure is in edge-aligned mode, and the starting value is 0, counting up as an example, showing various cases.

Next trigger signal and interrupt generation.

1, brake interruption

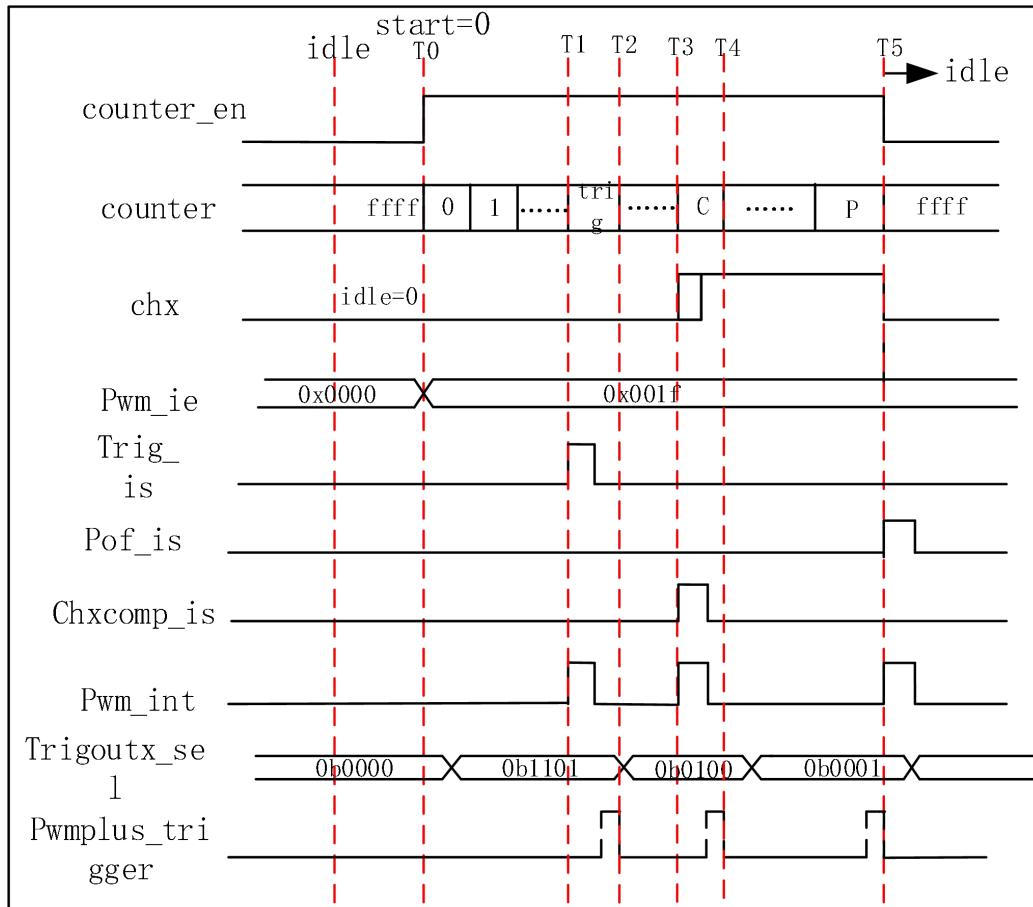


picture5-108 PWMPLUS Brake Interrupt Timing Diagram

As shown in the figure above, the brake interrupt signal is controlled by the brake interrupt enable register brake_ie and brake signal control register ch_brakecontrol, only any channel is subject to anybitcontrol of the input brake signal and when the corresponding bit in input brake

When the effective level of the car signal arrives, the corresponding interrupt status can be generated, and when the interrupt enable is1, the corresponding interrupt signal. As shown above, inT0time, enable the brake interrupt enable, and configure the brake signal control registerch_break, Make any channel controlled by brake signal, whenT1When the effective level of the input brake signal arrives at any time, a corresponding interrupt status and interrupt signal.

2, toggle point, specific trigger point, period overflow interrupt and trigger signal



picture5-109 PWMPLUSTurnover point, specific trigger point, cycle overflow interrupt and trigger signal timing diagram

As shown above, inT0Open the corresponding interrupt enable at all times, that is, the internal trigger point interrupt under counting up, flip Turn point interrupt and cycle overflow interrupt are enabled. existT1When the time reaches the internal trigger point, an interrupt state of the internal trigger point will be generated. At the same time, an interrupt signal is generated, and the internal trigger point only generates an interrupt state and does not affectPWMWaveform generation. in the same way existT3andT5After reaching the turning point and period overflow at time respectively, the corresponding turning point interrupt status and period overflowing will be generated. interrupt status and generate a corresponding interrupt signal. Through the configuration registerPWMPLUS_TRIG_CFGcan choosetriggerletter

The output function of the number, after reaching the corresponding state, it will generate the correspondingtrigger. The above figure takes an up counter as an example, showing

in differencettriggerThe timing diagram under the signal function will appear in theT2,T4,T5Generated before time

Output signal for one system clock cycle. However, the trigger signals of different functions cannot be selected at the same time, so eachbitbit

The trigger signal can only select one output function at a time, and it can be selected through differentbitbits to configure different output functions, this module

pwm_triggerin total4bit.

autoload

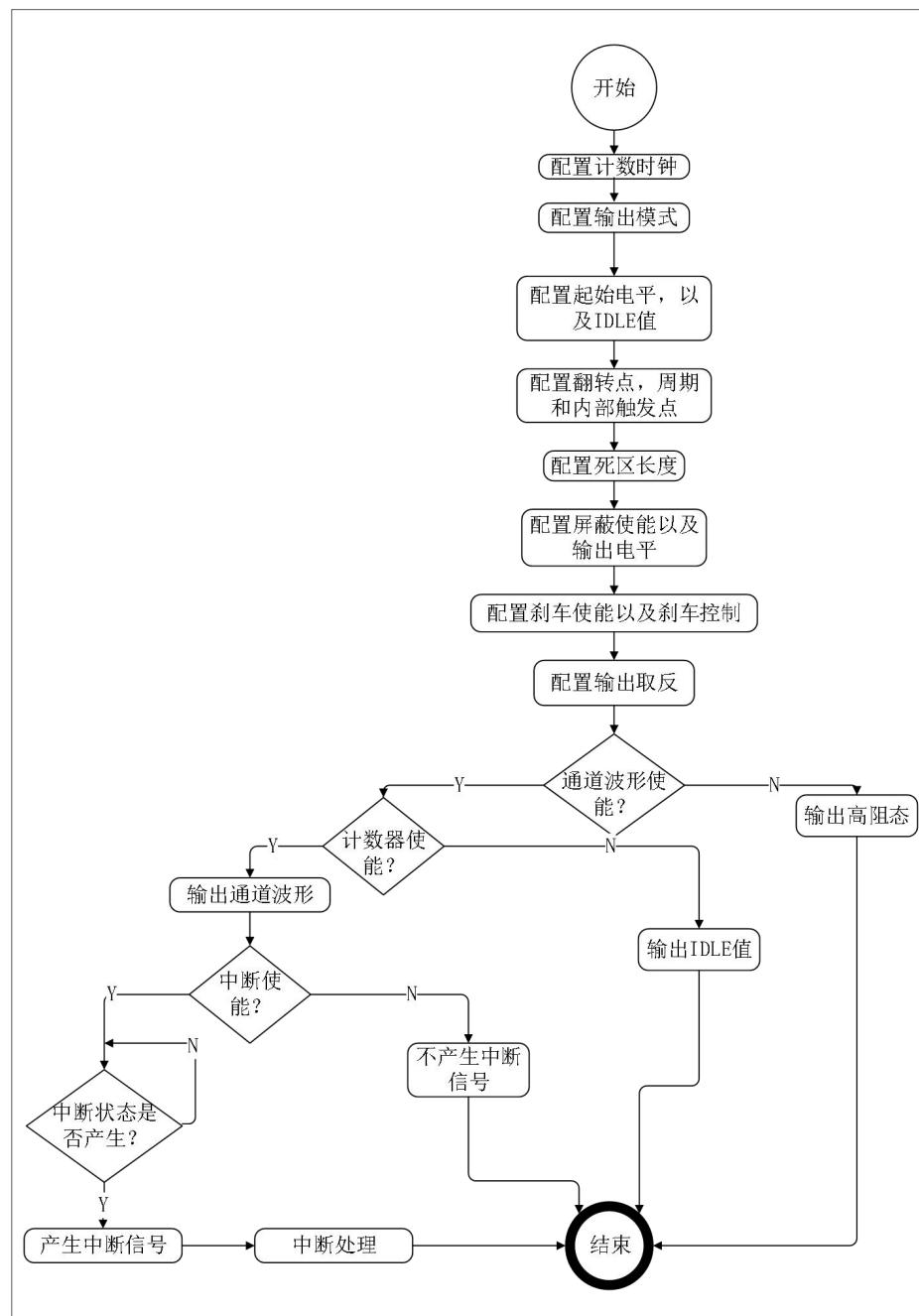
PWMPLUSThe module supports the autoload function, which can automatically load registers by configuringAUTO_RELOAD to achieve, indicating how many times the period overflows and automatically loads the period value, comparison value, dead zone value andTRIGGERvalue. since

The dynamic loading action is in (AUTO_RELOAD+1) is autoloaded after a cycle overflow.

software loading

PWMPLUSThe module supports software loading function, in this modulePWM_PERIOD,PWM_CH0_COMP, PWM_CH1_COMP,PWM_CH2_COMP,PWM_CH0_DT,PWM_CH1_DT,PWM_CH2_DT, TRIG_COMPthis8Each register has its own shadow register. software to softwareLOADcontrol bitSWLOADWrite1 After this period, the hardware resets this8The latest value held in a registerloadinto their respective shadow registers, and Take effect in the next cycle.

Operating procedures



picture5-110 PWMPLUSModule operation flow chart

- configurationPWMPLUSclock enable
- portThe port is configured asPWMPLUSFunction
- Configure count clock

- Configure output mode, counting cycle mode, counting behavior mode
- Configure the starting level and IDLEValue
- Configure period value, flip point value and internal trigger point value
- Configure autoload registers
- Configure dead zone length
- Configure mask enable and output level
- Configure brake enable, brake filter, active edge of brake signal and brake output level
- Configure Output Inversion
- If interrupts are configured, enable PWMPLUS interrupt
- Configure channel waveform output enable, if enable is not enabled then PADPort output high impedance state
- Configure the counter to enable, if the enable is not turned on, it will output IDLEvalue

Register description

PWMPLUS_CFGregister(0x00)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserve
15:8	AUTO_RELOAD	R/W	0	Automatic loading register, indicating how many times the cycle overflows automatically Load once period value, compare value, deadband value and TRIGGER value. The autoload action is in (AUTO_RELOAD+1) is autoloaded after a cycle overflow.
7:4	RESERVED	R	0	reserve
3	OUT_MODE	R/W	0	PWMoutput mode 0: Edge-aligned mode output 1: center-aligned mode output
2	CNT_REP	R/W	0	PWMCounter loop mode 0: Single, stop after one counting cycle 1: loop, loop counting after startup until software stops

1	CNT_TYPE	R/W	0	<p>PWMCounter Behavior</p> <p>0: count up 1: count down</p> <p>If it is center-aligned mode, it means that the first half cycle of the counter counting behavior.</p>
0	COUNTER_EN	R/W	0	<p>Counter Enable Register</p> <p>When this bit is configured as 1, it means that the counter starts counting, CH0/CH1/CH2According to the pre-configured period value, comparison value and dead zone value to generate the corresponding channel PWM output waveform.</p> <p>When configured in one-shot mode, this bit</p> <p>Hardware will be automatically cleared.</p> <p>When configured as a loop mode, this bit needs to be cleared by software, To stop the counter counting, when a complete counting cycle is completed period after the output waveform level returns to the IDLE state.</p>

Note: CH0, CH1 and CH2 share a cycle configuration register.

PWMPLUS_GENregister(0x04)

bit field	name	type	reset value	describe
31:30	RESERVED	R	0	reserve
29	CH2N_OE	R/W	0	<p>CH2NChannel waveform output enable</p> <p>0: The output is off, and the pin is in a high-impedance state 1: output CH2Nsquare wave</p>
28	CH2_OE	R/W	0	<p>CH2Channel waveform output enable</p> <p>0: The output is off, and the pin is in a high-impedance state 1: output CH2square wave</p>
27	CH1N_OE	R/W	0	<p>CH1NChannel waveform output enable</p> <p>0: The output is off, and the pin is in a high-impedance state 1: output CH1Nsquare wave</p>
26	CH1_OE	R/W	0	<p>CH1Channel waveform output enable</p> <p>0: The output is off, and the pin is in a high-impedance state 1: output CH1square wave</p>

25	CH0N_OE	R/W	0	CH0NChannel waveform output enable 0: The output is off, and the pin is in a high-impedance state 1: outputCH0Nsquare wave
twenty four	CH0_OE	R/W	0	CH0Channel waveform output enable 0: The output is off, and the pin is in a high-impedance state 1: outputCH0square wave
23:22	RESERVED	R	0	reserve
twenty one	CH2N_OUTINV	R/W	0	CH2NChannel output state selection 0: No change in level 1: Output level reversed
20	CH2_OUTINV	R/W	0	CH2Channel output state selection 0: No change in level 1: Output level reversed
19	CH1N_OUTINV	R/W	0	CH1NChannel output state selection 0: No change in level 1: Output level reversed
18	CH1_OUTINV	R/W	0	CH1Channel output state selection 0: No change in level 1: Output level reversed
17	CH0N_OUTINV	R/W	0	CH0NChannel output state selection 0: No change in level 1: Output level reversed
16	CH0_OUTINV	R/W	0	CH0Channel output state selection 0: No change in level 1: Output level reversed
15:11	RESERVED	R	0	reserve
10	CH2_START	R/W	0	originalCH2Output status value at start of channel count 0:originalCH2channel output0level 1 :originalCH2channel output1After level counting starts, theCH2NforCH2the anti

9	CH1_START	R/W	0	originalCH1Output status value at start of channel count 0:originalCH1channel output0level 1 :originalCH1channel output1After level counting starts, theCH1NforCH1the anti
8	CH0_START	R/W	0	originalCH0Output status value when the channel starts counting 0:originalCH0channel output0level 1 :originalCH0channel output1After level counting starts, theCH0NforCH0the anti
7:6	RESERVED	R	0	reserve
5	CH2N_IDLE	R/W	1	originalCH2NOutput status value when the channel is idle 0:originalCH2Nchannel output0level 1 :originalCH2Nchannel output1level
4	CH2_IDLE	R/W	0	originalCH1Output status value when the channel is idle 0:originalCH1channel output0level 1 :originalCH1channel output1level
3	CH1N_IDLE	R/W	1	originalCH1NOutput status value when the channel is idle 0:originalCH1Nchannel output0level 1 :originalCH1Nchannel output1level
2	CH1_IDLE	R/W	0	originalCH1Output status value when the channel is idle 0:originalCH1channel output0level 1 :originalCH1channel output1level
1	CH0N_IDLE	R/W	1	originalCH0NOutput status value when the channel is idle 0:originalCH0Nchannel output0level 1 :originalCH0Nchannel output1level
0	CH0_IDLE	R/W	0	originalCH0Output status value when the channel is idle 0:originalCH0channel output0level 1 :originalCH0channel output1level

PWMPLUS_CLKSRCregister(0x08)

bit field	name	type	reset value	describe

31:16	PREDIV	R/W	0	Internal prescaler clock frequency selection. with this modulepclkAs the clock source of the prescaler clock. 0x0000:expresspclkof1frequency division 0x0001:expresspclkof2frequency division 0x0002:expresspclkof3frequency division ... 0xFFFF:expresspclkof65536frequency division
15:6	RESERVED	R	0	reserved bit
5	EXTPLUS1_EDGE	R/W	0	ExtPLUS1Edge selection control when used as counting clock 0: Falling edge trigger counting 1: Rising edge trigger counting
4	EXTPLUS0_EDGE	R/W	0	ExtPLUS0Edge selection control when used as counting clock 0: Falling edge trigger counting 1: Rising edge trigger counting
3	RESERVED	R	0	reserved bit
2:0	CNT_SRC	R/W	0	PWMCounter count clock selection 000: Select the internal prescaler clock as the count clock 001:chooseextPLUS[0]as a counting clock 010 :chooseextPLUS[1]as a counting clock 011 :choosetmPLUS[0]as a counting clock 100 :choosetmPLUS[1]as a counting clock 101 :choosetmPLUS[2]as a counting clock 110 :choosetmPLUS[3]as a counting clock 111 :reserve

PWMPLUS_BRAKE_CFGregister(0x0c)

bit field	name	type	reset value	describe
31:26	RESERVED	R	0	reserved bit

25:24	BRAKE_FILTER	R/W	0	Brake signal digital filter control 00: no filtering 01:conduct2an internal prescaler clock filter 10 :conduct4an internal prescaler clock filter 11 :conduct8an internal prescaler clock filter
23:22	RESERVED	R	0	reserved bit
twenty one	BRAKE_CH2NPOL	R/W	0	when brakingCH2NOutput level selection 0: Output when braking0 1: Output when braking1
20	BRAKE_CH2POL	R/W	0	when brakingCH2Output level selection 0: Output when braking0 1: Output when braking1
19	BRAKE_CH1NPOL	R/W	0	when brakingCH1NOutput level selection 0: Output when braking0 1: Output when braking1
18	BRAKE_CH1POL	R/W	0	when brakingCH1Output level selection 0: Output when braking0 1: Output when braking1
17	BRAKE_CH0NPOL	R/W	0	when brakingCH0NOutput level selection 0: Output when braking0 1: Output when braking1
16	BRAKE_CH0POL	R/W	0	when brakingCH0Output level selection 0: Output when braking0 1: Output when braking1
15	RESERVED	R	0	reserved bit
14:12	BRAKE_LEV	R/W	0	Brake effective level selection BRAKELEVofBit2correspondbrake in[2],Bit1correspond brake in[1],Bit0correspondbrake in[0]. 0: Indicates that the brake input is active at low level 1: Indicates that the brake input is active at high level
11:9	RESERVED	R	0	reserved bit

8:6	CH2_BRAKE	R/W	0	CH2/CH2Nbrake control selection CH2_BRAKEofBit2correspondbrake in[2],Bit1correspond brake in[1],Bit0correspondbrake in[0]. 0: Indicates that it is not controlled by the brake signal 1: Indicates that it is controlled by the brake signal
5:3	CH1_BRAKE	R/W	0	CH1/CH1Nbrake control selection CH1_BRAKEofBit2correspondbrake in[2],Bit1correspond brake in[1],Bit0correspondbrake in[0]. 0: Indicates that it is not controlled by the brake signal 1: Indicates that it is controlled by the brake signal
2:0	CH0_BRAKE	R/W	0	CH0/CH0Nbrake control selection CH0_BRAKEofBit2correspondbrake in[2],Bit1correspond brake in[1],Bit0correspondbrake in[0]. 0: Indicates that it is not controlled by the brake signal 1: Indicates that it is controlled by the brake signal

Note: When the brake starts, it will take effect immediately on the output waveform; during the brake process, the counter counts normally;

The channel outputs a waveform after a complete cycle.

PWMPLUS_MASKLEVregister(0x10)

bit field	name	type	reset value	describe
31:6	RESERVED	R	0	reserved bit
5	CH2N_MASKLEV	R/W	0	CH2NChannel mask level selection 0: Force output during masking period0 1: Force output during masking period1
4	CH2_MASKLEV	R/W	0	CH2Channel mask level selection 0: Force output during masking period0 1: Force output during masking period1
3	CH1N_MASKLEV	R/W	0	CH1NChannel mask level selection 0: Force output during masking period0 1: Force output during masking period1

2	CH1_MASK_LEV	R/W	0	CH1Channel mask level selection 0: Force output during masking period0 1: Force output during masking period1
1	CH0N_MASKLEV	R/W	0	CH0NChannel mask level selection 0: Force output during masking period0 1: Force output during masking period1
0	CH0_MASKLEV	R/W	0	CH0Channel mask level selection 0: Force output during masking period0 1: Force output during masking period1

PWMPLUS_PERIODregister(0x1C)

bit field	name	type	reset value	describe
31:16	RESERVED	RO	0	reserved bit
15:0	PERIOD	R/W	0xffff	PWMPLUSPeriod configuration register. The actual counting period is the period value configured for this register plus1. Note0: period cannot be configured as0. Example: configure as decimal199, then thinkPWMThe waveform period is200. Note1: In center-aligned mode, the actual period is the configured value plus 1of2times.

PWMPLUS_CH0_COMPRegister(0x20)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	CH0_COMP	R/W	0	CH0/CH0Nrollover point configuration register. Note: the count value is less than the flip point value, outputstart[0]; Greater than or equal to the flip point value, outputstart[0]of non.

PWMPLUS_CH1_COMPRegister(0x24)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	CH1_COMP	R/W	0	CH1/CH1Nrollover point configuration register. Note: the count value is less than the flip point value, outputstart[1]; Greater than or equal to the flip point value, outputstart[1]of non.

PWMPLUS_CH2_COMPRegister(0x28)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	CH2_COMP	R/W	0	CH2/CH2Nrollover point configuration register. Note: the count value is less than the flip point value, outputstart[2]; Greater than or equal to the flip point value, outputstart[2]of non.

PWMPLUS_CH0_DTregister(0x30)

bit field	name	type	reset value	describe
31:10	RESERVED	R	0	reserved bit
9:0	CH0_DT	R/W	0	CH0/CH0NDead Band Length Configuration Register Note0: configured as0Indicates no dead zone; configured as1Indicates that the length of the dead zone is1, configured as2Indicates that the length of the dead zone is2, and so on. Note1: When configuring this value, it needs to be related to the period value, CH0_STARTvalue, CH0_COMPvalue, otherwise the output waveform may not be as expected.



PWMPLUS_CH1_DTregister(0x34)

bit field	name	type	reset value	describe
31:10	RESERVED	R	0	reserved bit
9:0	CH1_DT	R/W	0	<p>CH1/CH1NDead Band Length Configuration Register</p> <p>Note0: configured as0Indicates no dead zone; configured as1Indicates that the length of the dead zone is1, configured as2Indicates that the length of the dead zone is2, and so on.</p> <p>Note1: When configuring this value, it needs to be related to the period value, CH1_STARTvalue, CH1_COMPvalue, otherwise the output waveform may not be as expected.</p>

PWMPLUS_CH2_DTregister(0x38)

bit field	name	type	reset value	describe
31:10	RESERVED	R	0	reserved bit
9:0	CH2_DT	R/W	0	<p>CH2/CH2NDead Band Length Configuration Register</p> <p>Note0: configured as0Indicates no dead zone; configured as1Indicates that the length of the dead zone is1, configured as2Indicates that the length of the dead zone is2, and so on.</p> <p>Note1: When configuring this value, it needs to be related to the period value, CH2_STARTvalue, CH2_COMPvalue, otherwise the output waveform may not be as expected.</p>

PWMPLUS_TRIG_COMPreregister(0x40)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit
15:0	TRIG_COMP	R/W	0	<p>Internal trigger point configuration register</p> <p>Note: The internal trigger point value must be less than the period value</p>

PWMPLUS_TRIG_CFGregister(0x44)

bit field	name	type	reset value	describe
31:4	RESERVED	R	0	reserved bit
3:0	TIRGOUT_L_SEL	R/W	0	outputtrigger0Signal function selection 0000: No signal output; 0001: The overflow point of the upward counting cycle; 0010: overflow point of down counting cycle; 0011: Up or down counting period overflow point; 0100:CH0Count up flip points; 0101: CH0Count down flip points; 0110: CH0Flip points up or down; 0111: CH1Count up flip points; 1000:CH1 Count down flip points; 1001:CH1Flip points up or down; 1010:CH2Count up flip points; 1011:CH2Count down flip points; 1100:CH2Flip points up or down; 1101: count up internal trigger points; 1110: Count down internal trigger points; 1111: Up or down the internal trigger point;

PWMPLUS_IERegister(0x60)

bit field	name	type	reset value	describe
31:20	RESERVED	R	0	reserved bit
19	AUTORELOAD_IE	R/W	0	Autoload Interrupt Enable
18	BRAK2_IE	R/W	0	brake2interrupt enable
17	BRAK1_IE	R/W	0	brake1interrupt enable

16	BRAK0_IE	R/W	0	brake0interrupt enable
15:13	RESERVED	R	0	reserved bit
12	DOWN_TRIG_IE	R/W	0	count down toTRIGGERTrigger point interrupt enable
11	DOWN_POF_IE	R/W	0	Down count period overflow interrupt enable
10	DOWN_CH2COMP_IE	R/W	0	count downCH2Interrupt enable on reaching the tipping point
9	DOWN_CH1COMP_IE	R/W	0	count downCH1Interrupt enable on reaching the tipping point
8	DOWN_CH0COMP_IE	R/W	0	count downCH0Interrupt enable on reaching the tipping point
7:5	RESERVED	R	0	reserved bit
4	UP_TRIG_IE	R/W	0	count up toTRIGGERTrigger point interrupt enable
3	UP_POF_IE	R/W	0	Up count period overflow interrupt enable
2	UP_CH2COMP_IE	R/W	0	count upCH2Interrupt enable on reaching the tipping point
1	UP_CH1COMP_IE	R/W	0	count upCH1Interrupt enable on reaching the tipping point
0	UP_CH0COMP_IE	R/W	0	count upCH0Interrupt enable on reaching the tipping point

PWMPLUS_IFregister(0x64)

bit field	name	type	reset value	describe
31:20	RESERVED	R	0	reserved bit
19	AUTORELOAD_IF	R/W	0	autoload interrupt status Write1clear
18	BRAK2_IF	R/W	0	brake2interrupt status Write1clear Note: Only any channel is braked2signal control, the state will be produced.

17	BRAK1_IF	R/W	0	<p>brake1interrupt status Write1clear Note: Only any channel is braked1signal control, the state will be produced.</p>
16	BRAKO_IF	R/W	0	<p>brake0interrupt status Write1clear Note: Only any channel is braked0signal control, the state will be produced.</p>
15:13	RESERVED	R	0	reserved bit
12	DOWN_TRIG_IF	R/W	0	<p>count down toTRIGGERTrigger point interrupt status Write1clear Note: Edge-aligned mode means configured to count down when to up to the internal trigger point; center-aligned mode, means configured to When counting down, the internal trigger point is reached during the first half cycle or the configured When counting up, the internal trigger point is reached in the second half of the cycle</p>
11	DOWN_POF_IF	R/W	0	<p>Down count period overflow interrupt status Write1clear Note: Edge-aligned mode means configured to count down when to up to the cycle overflow point; center-aligned mode, means configured to When counting down, the period overflow point is reached in the first half cycle or the configured When counting up, the period overflow point is reached in the second half of the cycle</p>
10	DOWN_CH2COM_P_IF	R/W	0	<p>count downCH2Reached the break point interrupt state Write1clear</p>
9	DOWN_CH1COM_P_IF	R/W	0	<p>count downCH1Reached the break point interrupt state Write1clear</p>
8	DOWN_CH0COM_P_IF	R/W	0	<p>count downCH0Reached the break point interrupt state Write1clear Note: Edge-aligned mode means configured to count down when to up to the rollover point; center-aligned mode, indicating that the configuration is counted down When counting, the rollover point is reached in the first half cycle or configured to count up Counting, the flipping point is reached in the second half cycle. other channel behavior Same with it.</p>
7:5	RESERVED	R	0	reserved bit

4	UP_TRIG_IF	R/W	0	<p>count up toTRIGGERTrigger point interrupt status Write1clear Note: Edge-aligned mode means configured to count up when to up to the internal trigger point; center-aligned mode, means configured to When counting up, the internal trigger point is reached in the first half cycle or the configured When counting down, the internal trigger point is reached in the second half of the cycle</p>
3	UP_POF_IF	R/W	0	<p>Up count cycle overflow interrupt status Write1clear Note: Edge-aligned mode means configured to count up when to up to the cycle overflow point; center-aligned mode, means configured to When counting up, the period overflow point is reached in the first half cycle or the configured When counting down, the period overflow point is reached in the second half of the cycle</p>
2	UP_CH2COMP_IF	R/W	0	<p>count upCH2Reached the break point interrupt state Write1clear</p>
1	UP_CH1COMP_IF	R/W	0	<p>count upCH1Reached the break point interrupt state Write1clear</p>
0	UP_CH0COMP_IF	R/W	0	<p>count upCH0Reached the break point interrupt state Write1clear Note: Edge-aligned mode means configured to count up when to up to the flip point; center-aligned mode, indicating that the configuration is counting up When counting, the rollover point is reached in the first half cycle or configured to count down Counting, the flipping point is reached in the second half cycle. other channel behavior Same with it.</p>

PWMPLUS_SWLOADregister(0x84)

bit field	name	type	reset value	describe
31:1	RESERVED	R	0	reserved bit
0	SWLOAD	R/W	0	<p>PWMConfiguration Register SoftwareLOADcontrol bit In this modulePWM_PERIOD,PWM_CH0_COMP, PWM_CH1_COMP, PWM_CH2_COMP,PWM_CH0_DT, PWM_CH1_DT,PWM_CH2_DT, TRIG_COMPthis8Each register has its own shadow register. Software writes to this bit1After this period, the hardware resets this8The latest value held in a register loadto their respective shadow registers and take effect on the next cycle.</p>

PWMPLUS_MASK_ENregister(0x88)

bit field	name	type	reset value	describe
31:6	RESERVED	R	0	reserved bit
5	CH2N_MASK_EN	R/W	0	CH2NChannel Mask Enable write the bit1, the forced output shielding function is activated, and the forced output level passesMASK_CFGchoose.
4	CH2_MASK_EN	R/W	0	CH2Channel Mask Enable write the bit1, the forced output shielding function is activated, and the forced output level passesMASK_CFGchoose.
3	CH1N_MASK_EN	R/W	0	CH1NChannel Mask Enable write the bit1, the forced output shielding function is activated, and the forced output level passesMASK_CFGchoose.
2	CH1_MASK_EN	R/W	0	CH1Channel Mask Enable write the bit1, the forced output shielding function is activated, and the forced output level passesMASK_CFGchoose.
1	CH0N_MASK_EN	R/W	0	CH0NChannel Mask Enable write the bit1, the forced output shielding function is activated, and the forced output level passesMASK_CFGchoose.
0	CH0_MASK_EN	R/W	0	CH0Channel Mask Enable write the bit1, the forced output shielding function is activated, and the forced output level passesMASK_CFGchoose.

Note: When the shielding function of each channel is enabled, the forced output takes effect immediately. When the shielding function is cancelled, it will output normally according to the original waveform immediately.

PWMPLUS_CNT_STregister(0xE0)

bit field	name	type	reset value	describe
31:18	RESERVED	R	0	reserved bit

17	CNT_ST	R	0	PWMCounter working status 0: Indicates that the counter is not working 1: Indicates that the counter is counting
16	CNT_DIR	R	0	PWMThe current counting direction of the counter 0: Indicates that the counter is currently counting up 1: Indicates that the counter is currently counting down
15:0	PWMPLUS_CNT	R	0	PWMCounter current count value register

PWMPLUS_BRAKE_STregister(0xE4)

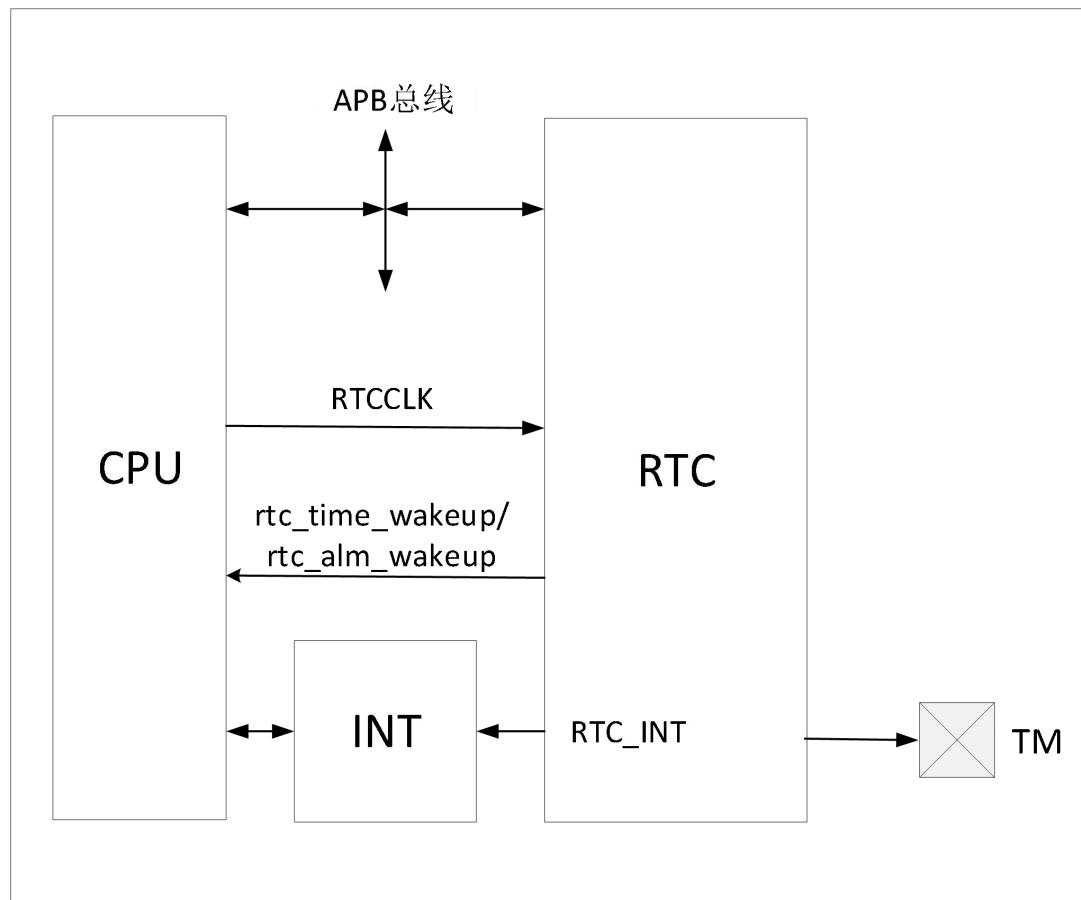
bit field	name	type	reset value	describe
31:2	RESERVED	R	0	reserved bit
1:0	BRAKE_ST	R	0	Current status of brake input signal BRAKE_ST[1]correspondbrake1,BRAKE_ST[0]correspond brake0.

5.15 Real Time Clock(RTC)

5.15.1 overview

The real time clock is an independent timer. The module has a real-time clock function, which can automatically solve the problem of leap year, input Clock source selectable RCL or XTAL, can output 1/2 seconds and its interrupt, using the RTC. The corresponding clock needs to be turned on before.

Its system block diagram is as follows:



picture5-111 RTCModule System Block Diagram

5.15.2 characteristic

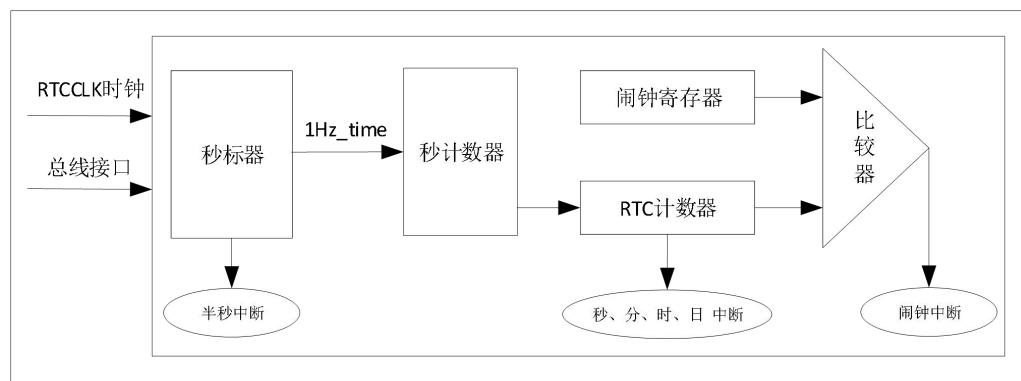
-With real-time clock function

-Alarm function

-With calendar function, timing range0-99Year

- Automatically resolve leap year issues
- Can output with RTC Synchronized seconds, minutes, hours, days, alarm interrupt
- RTCTwo clock sources can be selected for counting clock
- can output 1/2seconds and their interruptions
- can output RTC Time wake-up and alarm clock wake-up signals are used for low-power wake-up modules
- Data validity detection for clock setting and alarm clock setting
- haveBCDEncoded Alarm Clock Register
- haveBCDdecoded time register

5.15.3 Module Structure Diagram



picture5-112RTCBlock Diagram of Module Structure

5.15.4 Functional description

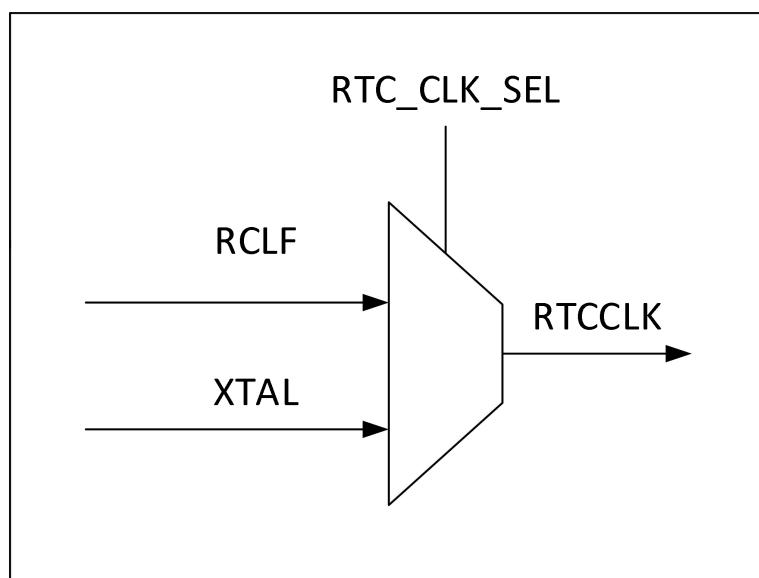
Count clock source selection

This chip RTC counter clock (RTCCLK) can select two clock sources: one for the on-chip low frequency RC oscillator (RCLF),

The other is an off-chip low frequency crystal oscillator (XTAL).

by setting SRC_CFG in the register RTC_CLK_SEL bit, RTCCLK The clock source can be given by XTAL or RCLF hour clock provided, and in RTC before enabling, it must be completed RTC related configurations (clock source selection, etc.).

RTCCLK Select the schematic as shown below:



picture5-113 RTCCLK select schematic

Second mark function

The seconds marker produces the seconds marker time used to update the calendar (1Hz_time), can also be configured by RTC_PRE register generation 1/2 seconds.

RTC_PRE register PRE_ROUND bits are the integer part of the prescaler, PRE_PERIOD refers to how many seconds each A fractional part calculation, PRE_DECIMAL is the fractional part of the prescaler, according to PRE_PERIOD and PRE_DECIMAL Compare which precision is better to determine the specific configuration value, 1 seconds have a margin of error of ±1ppm.

For example1:

No.239 total pages 432 Page



RTCThe clock frequency is32767.62Hz,butPRE_ROUNDconfigured as32766; Decimal parts are calculated separately
 $0.62 \times 8 = 4.96$ and $0.62 \times 16 = 9.92$, select8seconds need to be configured as5,choose16Need to configure in seconds10, it can be seen that two
The accuracy of the method is the same, and it is preferred8Second,PRE_DECIMALconfigured as5,PRE_PERIODconfigured as0.

but1The error in seconds is $((32767+5/8)/32767.62)-1 = 0.15\text{ppm}$

For example2:

RTCThe clock frequency is32767.71Hz,butPRE_ROUNDconfigured as32766; Decimal parts are calculated separately
 $0.71 \times 8 = 5.68$ and $0.71 \times 16 = 11.36$, select8seconds need to be configured as6,choose16Need to configure in seconds11, it can be seen that the selected
select16Higher precision in seconds, preferred16Second,PRE_DECIMALconfigured as11,PRE_PERIODconfigured as1.

but1The error in seconds is $((32767+11/16)/32767.71)-1 = -0.68\text{ppm}$

For example3:

RTCThe clock frequency is32770.094Hz,butPRE_ROUNDconfigured as32769; Decimal parts are calculated separately
 $0.094 \times 8 = 0.752$ and $0.094 \times 16 = 1.504$, select8seconds need to be configured as1,choose16Need to configure in seconds1,As can be seen
choose8Higher precision in seconds, preferred8Second,PRE_DECIMALconfigured as1,PRE_PERIODconfigured as0.

but1The error in seconds is $((32770+1/8)/32770.094)-1 = 0.946\text{ppm}$

calendar function

The real-time clock is generated based on the second marker1Hz_time, to accurately time the year, month, day, hour, minute, and second, and
Output timing time.RTCAccept the time setting, and you can set the time to start counting.

The calendar function has a set of configuration time registers and current time registers, usingBCDCode encoding format. configuration time
The register has the function of automatically checking the time format (except the week), and can judge the set time according to the value written by the user
is it legal. If the time format is invalid (such as1moon34day,2moon30day,twenty fourhour01grading), will be registered in the state
deviceRTC_IFmiddleTIME_ERRBit display setting time data format error. even if openLOAD_EN, configuration time
value will not take effect.

Alarm function

RTCThe module provides an alarm clock function, and when the counter counts to the alarm clock set value, an interrupt can be generated.

Alarm clock set byRTC_ARRegisters only support day, hour, minute, and second. The alarm clock register has an automatic detection

The function of checking the time format (except the week) can judge whether the set time is legal according to the value written by the user. If write

The format of the alarm clock register is illegal (such as20hour20point63Second,twenty fourhour01grading), will be in the status registerRTC_IF

middleALM_ERRIf the alarm clock format is wrong, even if the alarm clock function enable bit is turned on, the alarm clock will not take effect. when

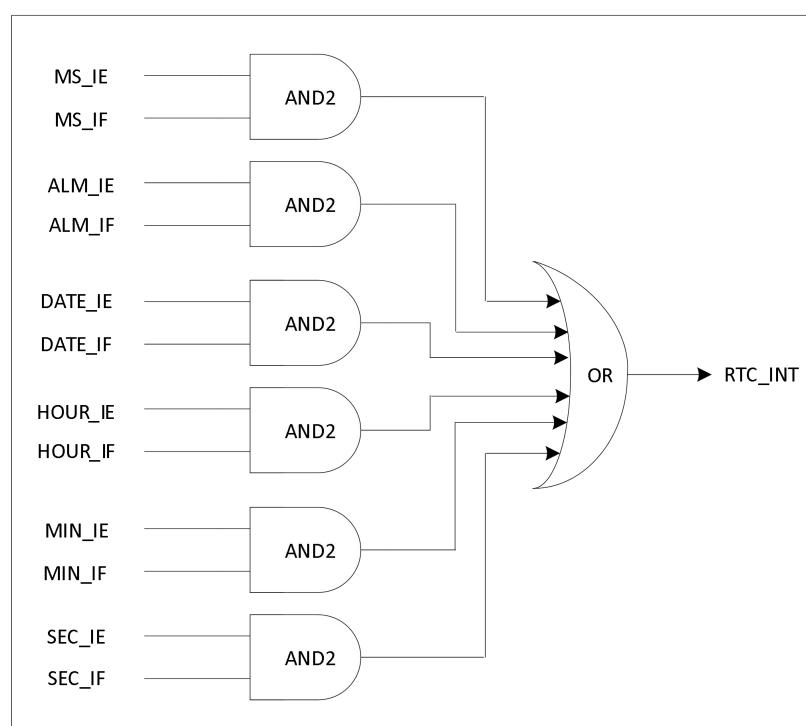
week set to full0, the alarm clock will only respond once, otherwise the alarm clock will respond in a loop until it is turned off.

interrupt function

This module provides six kinds of interrupt functions, including second, minute, hour, day, alarm clock and half-second interrupt. accessibleRTC_IE

The register configures the corresponding interrupt enable bit, throughRTC_IFregister to view each corresponding interrupt status bit.

The interrupt flag and interrupt diagram are as follows:



picture5-114RTCInterrupt flag and interrupt diagram

Low power wake-up function

This module provides two low-power wake-up functions, which can support the SLEEP mode and DEEPSLEEP mode to wake up the system.

They are time wake-up and alarm clock wake-up respectively.

by configurationRTC The time and alarm clock can generate corresponding wake-up signal to PMU wake-up circuit, by PMU call

Wake up circuit configurationLPMD_WKENregister can turn on the corresponding wake-up enable by LPMD_WKSTregister query wake-up state.

Operating procedures

Initialization settings

1, according to RTCCLK, configure RTC_PREGISTER so that the output 1Hz_time;

2, configurationRTC_TRandRTC_DRREGISTER;

3, read status register RTC_IFoFTIME_ERRbit, confirm that the format is correct;

4, configurationRTC_CFGofLOAD_ENBit is 1, to open the time setting function;

5, configurationRTC_ARREGISTER;

6, read status register RTC_IFoFALM_ERRbit, confirm that the format is correct;

7, configurationRTC_CFGofALM_ENBit is 1, turn on the alarm clock function;

8, configurationRTC_CFGofRTC_ENBit is 1, OpenRTC, start the timer;

Note: Skip if no time setting function 2-4; Skip if the alarm function is not required 5-7.

time setting

- 1, configurationRTC_TRandRTC_DRregister;
- 2, read status registerRTC_IFoFTIME_ERRbit, confirm that the format is correct;
- 3, configurationRTC_CFGofLOAD_ENBit is1, to open the time setting function;
- 4, RTCThe timer is on the nextRTCCLKThe clock cycle loads the new value and starts timing.

alarm clock setting

- 1, configurationRTC_ARregister;
- 2, read status registerRTC_IFoFALM_ERRbit, confirm that the format is correct;
- 3, configurationRTC_CFGofALM_ENBit is1, turn on the alarm clock function;
- 4, Wait for the alarm clock to respond.

read time

- 1, read and judgeRTC_VALIDfor1;
- 2, read registerRTC_TSTR,RTC_TSDR, get the current time value;
- 3, readRTC_CNTGet the current count value of the second mark;

interrupt response setting

- 1, configurationRTC_IE,have6Each interrupt can be configured separately;

2, Waiting for interrupt response;

3, readRTC_IFRegister, query the current interrupt status

4, Towards RTC_IFThe corresponding bit of the register is written1, to clear the corresponding interrupt.

register map

name	offset	bit width	type	reset value	describe
RTC BASE: 0x40069000					
RTC_CFG	0x00	32	R/W	0x00	RTCconfiguration register
RTC_IE	0x04	32	R/W	0x00	RTCinterrupt enable register
RTC_IF	0x08	32	R/W	0x00	RTCstatus register
RTC_PRE	0x10	32	R/W	0x7fff	RTCprescaler register
RTC_TR	0x14	32	R/W	0x00	RTCTime register
RTC_DR	0x18	32	R/W	0x101	RTCdate register
RTC_AR	0x1C	32	R/W	0x00	RTCalarm clock register
RTC_TSTR	0x20	32	RO	0x00	RTCcurrent time register
RTC_TSDR	0x24	32	RO	0x101	RTCcurrent date register
RTC_CNT	0x28	32	RO	0x00	RTCThe current count value of the second scale
RTC_VALID	0x2C	32	RO	0x00	RTCCurrent Time Valid Flag Register

Register description

RTC_CFGregister(0x00)

bit field	name	type	reset value	describe
31:3	RESERVED	RO	0x0	reserved bit

2	LOAD_EN	R/W, AC	0x0	RTCload registerRTC_TRandRTC_DRtime set value 1: Load the value set by the register, and the hardware will automatically clear 0: Do not load the value set by the register Note: recommended RTC_TRandRTC_DRAfter configuration, checkRTC_IF of TIME_ERRBit is 0, and then turn on this bit
1	ALM_EN	R/W	0x0	RTCAalarm function enable bit 1: Turn on the alarm function, only when ALM_WEEKDAY is 0, this bit is automatically cleared by hardware 0: Turn off the alarm function Note: recommended RTC_ARA After configuration, check RTC_IF of ALM_ERRBit is 0, and then turn on this bit
0	RTC_EN	R/W	0x0	RTCeable bit 1: OpenRTC 0: closureRTC

RTC_IERegister(0x04)

bit field	name	type	reset value	describe
31:6	RESERVED	RO	0x0	reserved bit
5	MS_IE	R/W	0x0	1/2Second interrupt enable bit 1: Enable 0: disable
4	ALM_IE	R/W	0x0	Alarm interrupt enable bit 1: Enable 0: disable
3	DATE_IE	R/W	0x0	day interrupt enable bit 1: Enable 0: disable
2	HOUR_IE	R/W	0x0	Hour interrupt enable bit 1: Enable 0: disable

1	MIN_IE	R/W	0x0	minute interrupt enable bit 1:Enable 0: disable
0	SEC_IE	R/W	0x0	Second interrupt enable bit 1:Enable 0: disable

RTC_IFregister(0x08)

bit field	name	type	reset value	describe
31:10	RESERVED	RO	0x0	reserved bit
9	ALM_ERR	RO	0x0	Alarm clock setting valid flag bit 1: Alarm setting error 0: The alarm clock is set correctly Note: This bit does not judge the alarm clock ALM_WEEKDAY
8	TIME_ERR	RO	0x0	Time setting valid flag bit 1: Time setting error 0: The time is set correctly Note: if BCD_WEEKDAY configured as 7 This bit is also valid
7:6	RESERVED	RO	0x0	reserved bit
5	MS_IF	R,W1C	0x0	1/2seconds interrupt response, active high, write1to clear
4	ALM_IF	R,W1C	0x0	Alarm clock interrupt response, active high, write1to clear
3	DATE_IF	R,W1C	0x0	th interrupt response, active high, write1to clear
2	HOUR_IF	R,W1C	0x0	Hour Interrupt Response, Active High, Write1to clear
1	MIN_IF	R,W1C	0x0	Min Interrupt Response, Active High, Write1to clear
0	SEC_IF	R,W1C	0x0	seconds interrupt response, active high, write1to clear

RTC_PREregister(0x10)

bit field	name	type	reset value	describe
31:25	RESERVED	RO	0x0	reserved bit
twenty four	PRE_PERIOD	R/W	0x0	Decimal calculation cycle selection 0:8Second 1:16Second Note: Calculate the fractional part every second
23:20	PRE_DECIMAL	R/W	0x0	Prescaler fractional part 0: Indicates no decimals 1:express1/8or 1/16... 7:express7/8or7/16 8 :express8/16 ... 15:express15/16
19:0	PRE_ROUND	R/W	0x7fff	prescaler integer part The count value is the configured value plus1

RTC_TRregister(0x14)

bit field	name	type	reset value	describe
31:27	RESERVED	RO	0x0	reserved bit

				Set the day of the week to which the time belongs,0-6for legitimate data 0:Sunday 1:Monday 2:Tuesday 3:Wednesday 4:Thursday 5:Friday 6:Saturday
26:24	BCD_WEEK	R/W	0x0	
23:22	RESERVED	RO	0x0	reserved bit
21:20	BCD_HOUR_DEC	R/W	0x0	Set the tens digit of the hour to which the time belongs (decimal) ,0-2efficient
19:16	BCD_HOUR	R/W	0x0	Set the unit digit of the hour to which the time belongs (decimal) ,0-9efficient
15	RESERVED	RO	0x0	reserved bit
14:12	BCD_MIN_DEC	R/W	0x0	Set the tens digit of the minute to which the time belongs (decimal),0-5efficient
11:8	BCD_MIN	R/W	0x0	Set the unit digit of the minute to which the time belongs (decimal),0-9efficient
7	RESERVED	RO	0x0	reserved bit
6:4	BCD_SEC_DEC	R/W	0x0	Set the tens digit of the second of the time (decimal) ,0-5efficient
3:0	BCD_SEC	R/W	0x0	Set the unit digit of the second of the time (decimal) ,0-9efficient

RTC_DRregister(0x18)

bit field	name	type	reset value	describe
31:24	RESERVED	R/W	0x0	reserved bit
23:20	BCD_YEAR_DEC	R/W	0x0	Set the tens digit of the year to which the time belongs (decimal) ,0-9efficient
19:16	BCD_YEAR	R/W	0x0	Set the unit digit of the year to which the time belongs (decimal) ,0-9efficient
15:13	RESERVED	RO	0x0	reserved bit.

12	BCD_MONTH_DEC	R/W	0x0	Set the tens digit of the month to which the time belongs (decimal), 0-1efficient
11:8	BCD_MONTH	R/W	0x1	Set the single digit of the month to which the time belongs (decimal) 0-9efficient
7:6	RESERVED	RO	0x0	reserved bit.
5:4	BCD_DATE_DEC	R/W	0x0	Set the tens digit of the date to which the time belongs (decimal) , 0-3efficient
3:0	BCD_DATE	R/W	0x1	Set the unit digit of the date to which the time belongs (decimal) ,0-9efficient

RTC_ARregister(0x1C)

bit field	name	type	reset value	describe
31	RESERVED	RO	0x0	reserved bit
30:24	ALM_WEEKDAY	R/W	0x0	<p>The week the alarm time belongs to, everybitrepresents a day of the week bit0=1:Sunday bit1=1:Monday bit2=1:Tuesday bit3=1:Wednesday bit4=1:Thursday bit5=1:Friday bit6=1:Saturday</p> <p>Note: for all0When , the alarm clock responds only once; for other values, the alarm clock responds multiple times untilALM_ENconfiguration off</p>
23:22	RESERVED	RO	0x0	reserved bit
21:20	ALM_HOUR_DEC	R/W	0x0	The tens digit of the hour to which the alarm time belongs (decimal) 0-2efficient
19:16	ALM_HOUR	R/W	0x0	The unit digit of the hour of the alarm time (decimal),0-9efficient
15	RESERVED	RO	0x0	reserved bit
14:12	ALM_MIN_DEC	R/W	0x0	The tens digit of the minute to which the alarm time belongs (decimal) ,0-5efficient
11:8	ALM_MIN	R/W	0x0	The unit digit of the minute to which the alarm time belongs (decimal) ,0-9efficient
7	RESERVED	RO	0x0	reserved bit



6:4	ALM_SEC_DEC	R/W	0x0	The tens digit of the second of the alarm time (decimal), 0-5 efficient
3:0	ALM_SEC	R/W	0x0	The single digit of the second of the alarm time (decimal), 0-9 efficient

RTC_TSTRregister(0x20)

bit field	name	type	reset value	describe
31:27	RESERVED	RO	0x0	reserved bit.
26:24	WEEKDAY	RO	0x0	the current day of the week, 0-6 for legitimate data 0:Sunday 1:Monday 2:Tuesday 3:Wednesday 4:Thursday 5:Friday 6:Saturday
23:22	RESERVED	RO	0x0	reserved bit.
21:20	HOUR_DEC	RO	0x0	the tens digit (decimal) of the hour to which the current time belongs, 0-2 efficient
19:16	HOUR	RO	0x0	the unit digit (decimal) of the hour to which the current time belongs, 0-9 efficient
15	RESERVED	RO	0x0	reserved bit.
14:12	MIN_DEC	RO	0x0	the tens digit (decimal) of the minute to which the current time belongs, 0-5 efficient
11:8	MIN	RO	0x0	the unit digit (decimal) of the minute to which the current time belongs, 0-9 efficient
7	RESERVED	RO	0x0	reserved bit.
6:4	SEC_DEC	RO	0x0	the tens digit (decimal) of the second to which the current time belongs, 0-5 efficient
3:0	SEC	RO	0x0	the unit digit (decimal) of the second to which the current time belongs, 0-9 efficient

RTC_TSDRregister(0x24)

bit field	name	type	reset value	describe
31:25	RESERVED	RO	0x0	reserved bit
twenty four	LEAP YEAR	RO	0x0	
23:20	YEAR_DEC	RO	0x0	Tens digit (decimal) of the epoch to which the current time belongs ,0-9efficient
19:16	YEAR	RO	0x0	The unit digit (decimal) of the epoch to which the current time belongs ,0-9efficient
15:13	RESERVED	RO	0x0	reserved bit
12	MONTH_DEC	RO	0x0	The tens digit of the month to which the current time belongs (decimal),0-1efficient
11:8	MONTH	RO	0x1	The single digit of the month to which the current time belongs (decimal) ,0-9efficient
7:6	RESERVED	RO	0x0	reserved bit
5:4	DATE_DEC	RO	0x0	The tens digit (decimal) of the date to which the current time belongs ,0-3efficient
3:0	DATE	RO	0x1	The single digit (decimal) of the date to which the current time belongs ,0-9efficient

RTC_CNTregister(0x28)

bit field	name	type	reset value	describe
31:20	RESERVED	RO	0x0	reserved bit
19:0	CNT_20	RO	0x0	20bitCount bit

RTC_VALIDregister(0x2C)

bit field	name	type	Reset value	description

31:1	RESERVED	RO	0x0	reserved bit
0	CUR_VALID	RO	0x0	<p>Current time valid flag</p> <p>When judging that the bit is 1 When, the current time register can be read (RTC_TSTR, RTC_TSDR, RTC_CNT)</p>

5.16UARTcontroller (UART)

5.16.1 overview

Universal Asynchronous Transceiver (Universal Asynchronous Receiver/Transmitter, UART) is a serial communication

Communication technology, often used for board-level communication between microcontrollers and computers, and between microcontrollers and microcontrollers, which is asynchronous communication,

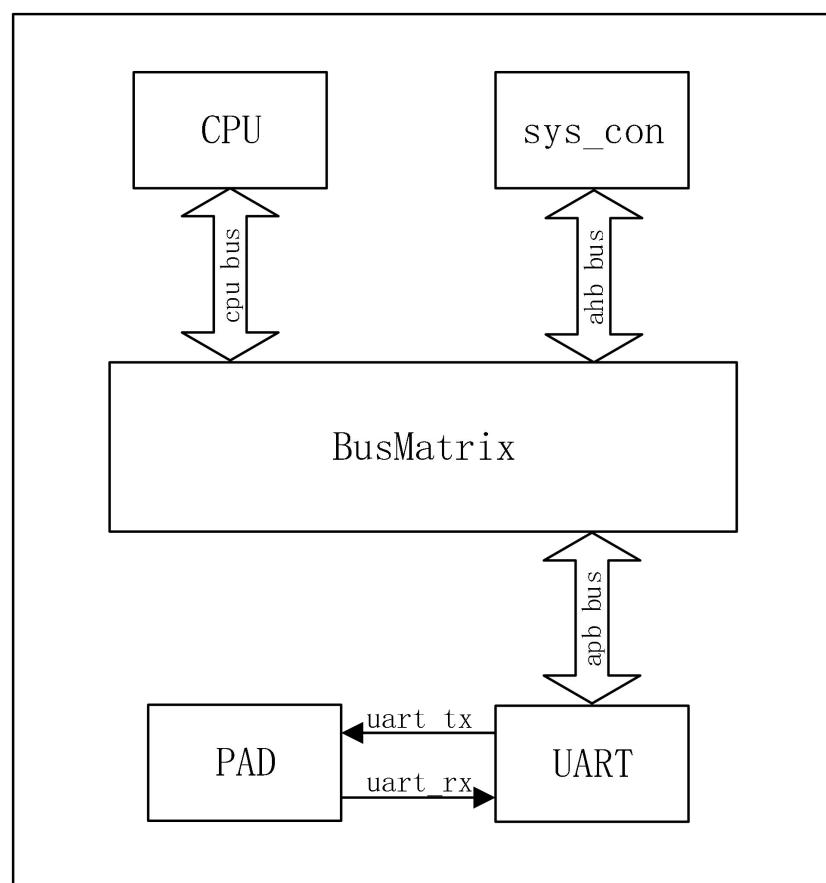
A port is capable of sending data on one wire and receiving data on another wire at the same time. The most important parameter of serial communication is baud

rate, data bits, stop bits, and parity. This chip has 3 Serial port, support baud rate configuration, data length, checksum

Configurable bits and stop bits, support multiple interrupts, support DMA transmission mechanism with hardware automatic flow control, receiving

and sends independently with a depth of 8 of FIFO, and FIFO The water level is configurable, you need to enable the corresponding UART

clock. The schematic diagram of the module system connection is as follows:



picture5-115 UARTModule System Block Diagram

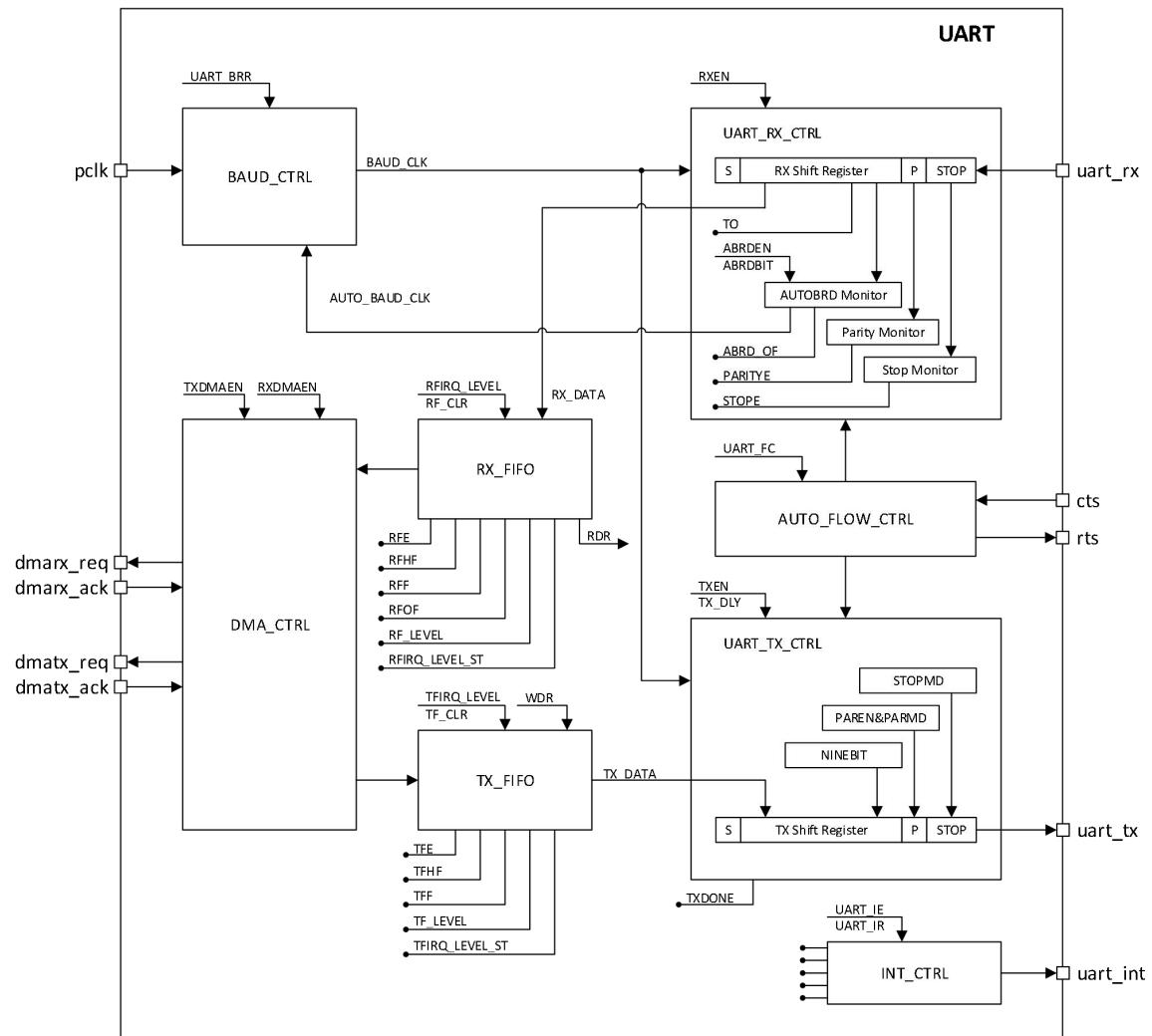
Pictured above isUARTSchematic diagram of the connection of the module system. inCPUaccessibleAPBBus interface direct controlUARTmodule,

UARTmodule throughPADRealize the communication between the chip and the external equipment.

5.16.2characteristic

- Support standardUARTprotocol
 - Support full duplex mode
 - Support baud rate configuration
 - Programmable data bit length (8bitor9bit)
- Configurable parity bit odd parity even parity constant0often1
- support1Bit stop bit, and supports sending delay time configurable
 - Support baud rate automatic detection function
 - With hardware automatic flow control function
 - supportDMAtransport mechanism
 - receive and transmit independently with a depth of8ofFIFO, andFIFOwater level configurable
 - Support send complete interrupt, receive complete interrupt, receive timeout interrupt function

5.16.3 Block Diagram of Module Structure



picture5-116 UARTBlock Diagram of Module Structure

UART via the internal baud rate controller **BAUD_CTRL**, according to **UART_BAUD** The set baud rate parameter produces raw sampling clock **BAUD_CLK**, available for the receiver **UART_RX_CTRL** Data sampling, sending end **UART_TX_CTRL** Send data.

UART Built in two depths of 8 of FIFO, used to store **UART_RX_CTRL** The valid data sampled **RX_FIFO** and for storage CPU or DMA written data to be sent **TX_FIFO**.

data receiver **UART_RX_CTRL** right **UART_RX** The data is synchronized and the data is sampled at the same time, and the sampled data is Analysis, check whether the data parity bit and stop bit are the same as the set mode, so as to generate the corresponding status bit, and put no Valid data with parity error, stop bit error **RX_DATA** deposit **RX_FIFO**.

data senderUART_RX_CTRLreceiveRX_FIFOThe data to be sent, according to the set data bit, parity bit and stop bit according to BAUD_CLKone by oneBITsend.

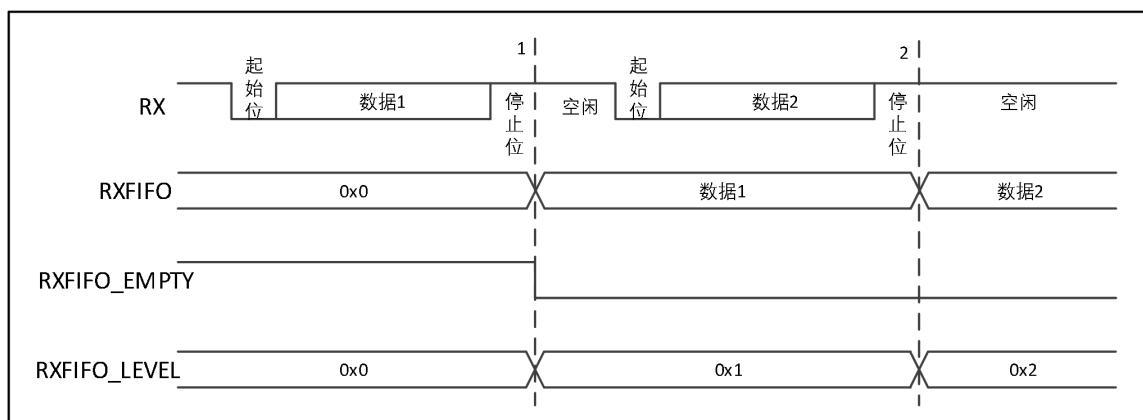
UART support DMA function, through the internal DMA_CTRL as well as DMA The corresponding interface implementation DMA control UART

Data receiving and sending.

UART through internal AUTO_FLOW_CTRL implement hardware flow control.

Functional description

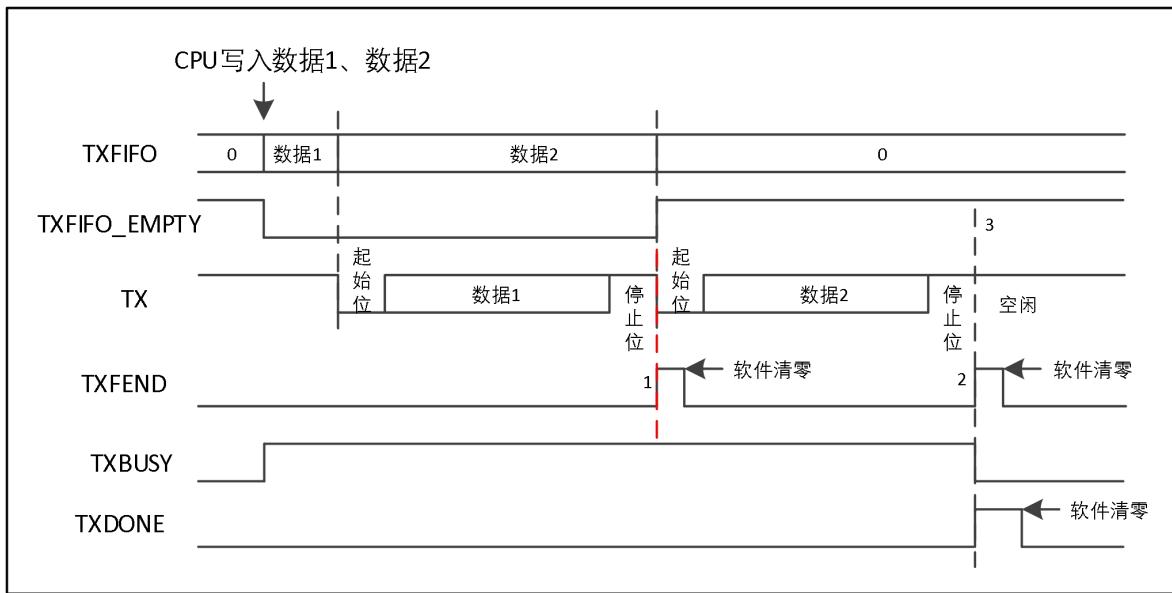
Receive timing and related status signals



picture5-117 UART Receive timing and related status signal diagram

Pictured above is UART receive a simple timing diagram, at the sign1 place, UART save received data to RXFIFO middle, RXFIFO water level increases, RXFIFO null is deasserted; flag2 place, UART save received data to RXFIFO middle, RXFIFO The water level increases.

Send timing and related status signals



picture5-118 UARTSend timing and related status signal diagram

Pictured above is UARTSend a simple sequence diagram, each time a set of data is sent, TXFEND will be set to 1, see above 1 and 2 shown here. When all data transmission is complete, TXDONE will be set to 1.

when TXFIFO There is data to be sent in , and the transmit shift register is in the process of transmission, then TXBUSY will be set to efficient.

Baud rate calculation

The formula for calculating the baud rate is as follows:

$$f_{baudrate} = \frac{f_{pclk}}{UARTDIV}$$

For example: $f_{pclk} = 48M$, the baud rate is set to 115200, but $UARTDIV = 48000000 / 115200 = 416.6$,

Can be selected by rounding 417.

Auto baud rate detection

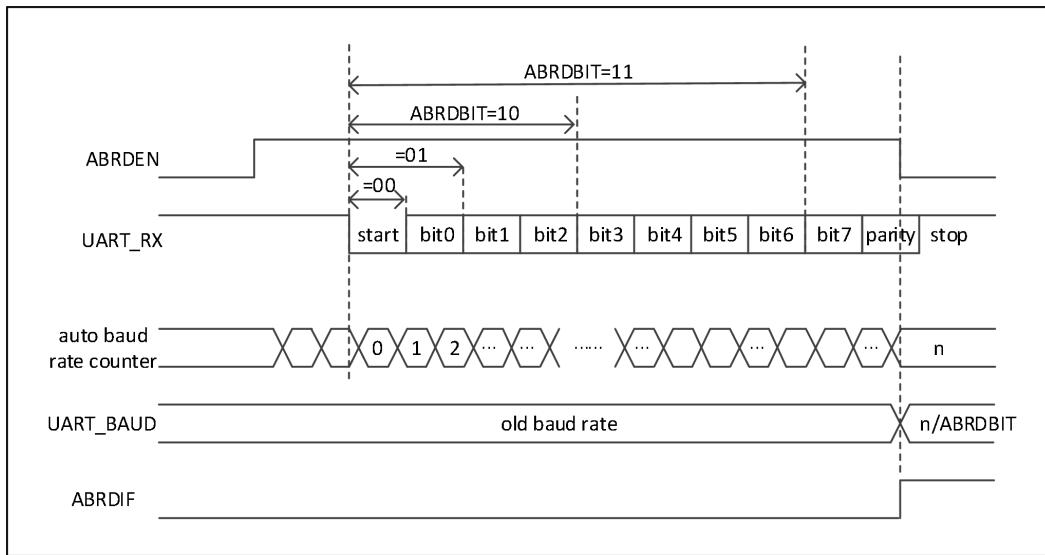
The auto-baud rate detection function can automatically measureuart_rxThe data input by the pin is used to calculate the baud rate. when the automatic wave

After the baud rate measurement is completed, the measurement result will be put intoUART_BAUDregister.

In auto-baud rate detection, will detectuart_rxData starts from the falling edge of the start bit and ends with the first rising edge time, which can be determined byABRDBITto make sure.

by configurationABRDENThe automatic baud rate detection function can be turned on. In the initial stage,uart_rxkeep as1, a section When the falling edge is detected, the start bit is received, the auto-baud counter is started and starts counting, when the first On the rising edge, the auto-baud counter will stop counting. Then, the auto-baud counter value is divided by theABRDBITthe knot automatic depositUART_BAUDregister, andABRDENwill be cleared and an auto-baud end flag will be generated Chi.

The schematic diagram of automatic baud rate calculation is as follows:



picture5-119 UARTSchematic diagram of automatic baud rate calculation

data sampling

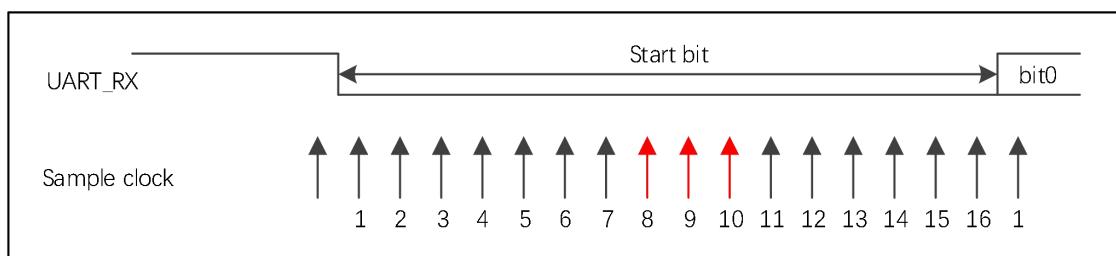
UARTAs an asynchronous communication interface, there is no independent operating clock, so the received data needs to be processed synchronously.

In order to be able to correctly obtain the received character data, UARTuse16Clock pair times the data baud rateUART_RXEnter

row data samples, eachbitdata bits have16sampling clock, take the middle8,9,10subsampled data as the actual

The received data is used.

The schematic diagram of data sampling is as follows:



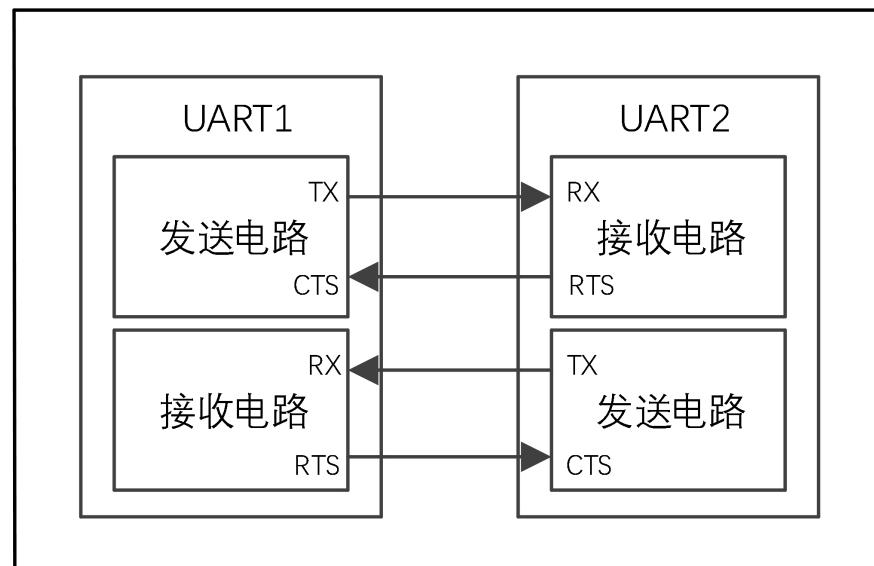
picture5-120 UARTSchematic diagram of data sampling

for everybitFor data, it will be performed according to the baud rate setting16sampling, take the sampling8sequence9 second and second10Second data to judge: if there are at least two samples in the three samples0, then thebitBit is0; If the three sampling contains at least two1, then thebitBit is1.

Hardware automatic flow control

UARTpassCTSinput andRTSoutput can be controlled2Asynchronous serial data flow between devices. As shown below

Show:



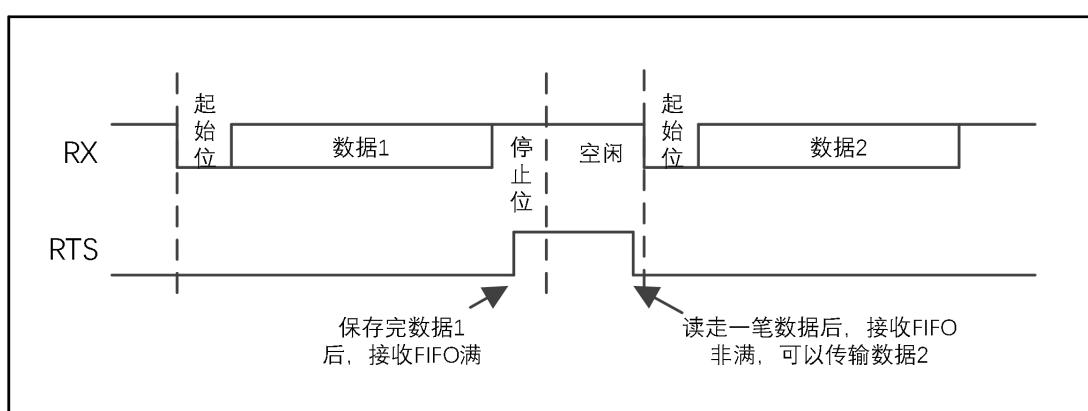
picture5-121 UARTAutomatic flow control connection diagram

RTSflow control:

When configuring RTSEN for 1, after the data is received FIFO, if there is space, then RTS is high level output (the level can be configured).

When data is received FIFO, if full, RTS is low level output, which means that you want to stop the data when the current group of data ends transmission.

An example of flow control communication is as follows:

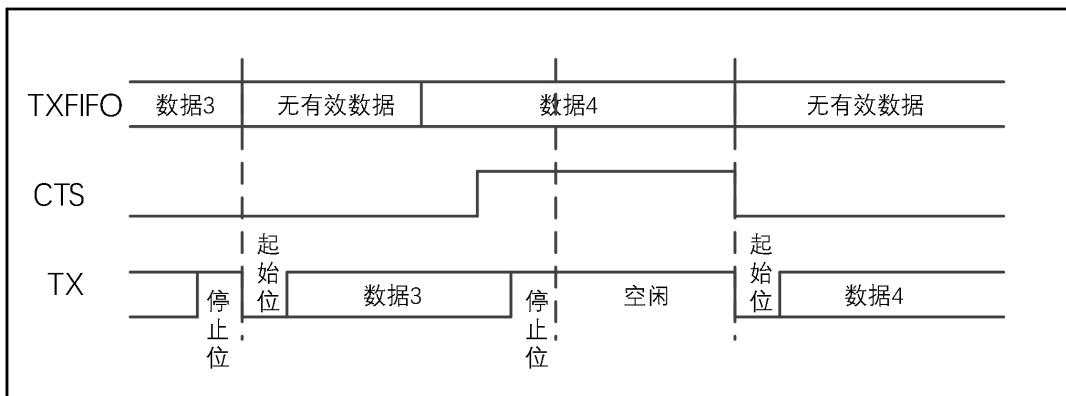


picture5-122 UARTSchematic diagram of flow control reception

CTSflow control:

When configuring CTSEN for 1, after that, the sending circuit checks the input before sending the next set of data. If CTS valid (level configurable), then send FIFO. The next set of data in can be sent, otherwise a set of data cannot be sent. If it becomes invalid during data transmission, the next group of data transmission will stop after the current transmission is completed.

start up CTS. An example of flow control communication is as follows:



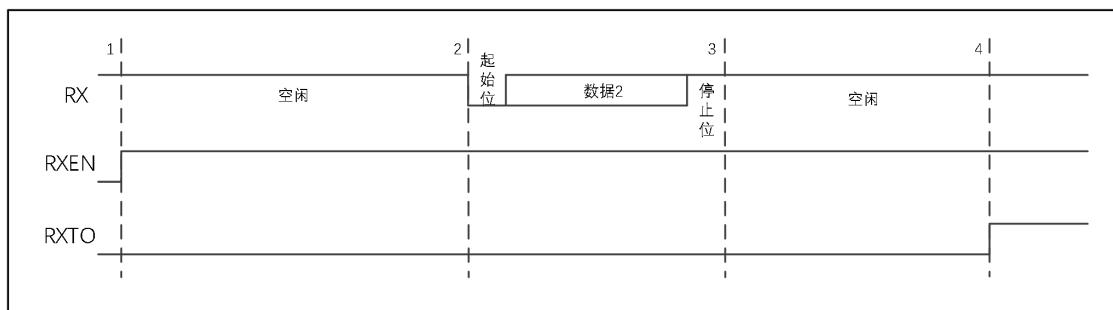
picture5-123 UART schematic diagram of flow control sending

Receive timeout instructions

Receive timeouts occur in two cases:

1. When the receiving enable is enabled, the timeout interrupt will take effect after at least one byte of data is received (the timeout interrupt is enabled open). When the time exceeds the configured timeout time, a timeout interrupt will be generated, and the user can process it in the interrupt service function.

Clear the timeout interrupt flag.



picture5-124 UART first receive timeout interrupt

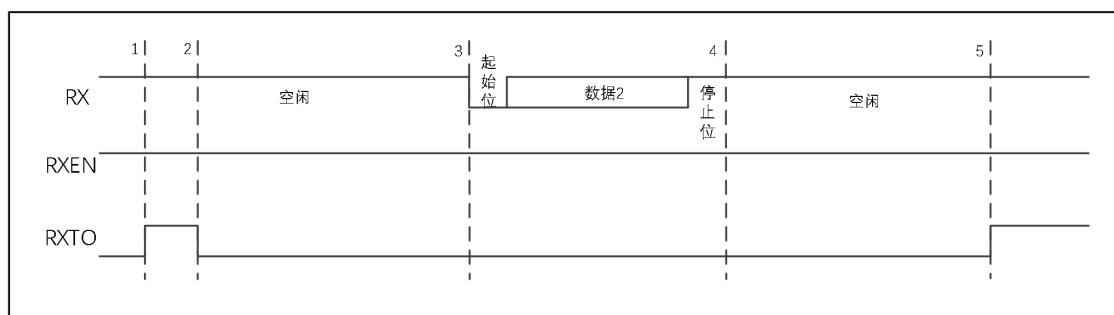
As shown in FIG:

1timeUARTreceive enable is turned on until the2time agoUARTThe reception is always idle, and there will be no timeout interrupt occurs;2timeUARTReceive the first data until3Time data reception is complete, and then enters idle

At this point the timeout counter starts counting until4A timeout interrupt is generated at all times.

2, When a receiving timeout interrupt occurs, the timeout interrupt can only be restarted after at least one byte of data is received again.

takes effect for the second time (when the timeout interrupt enable is enabled).



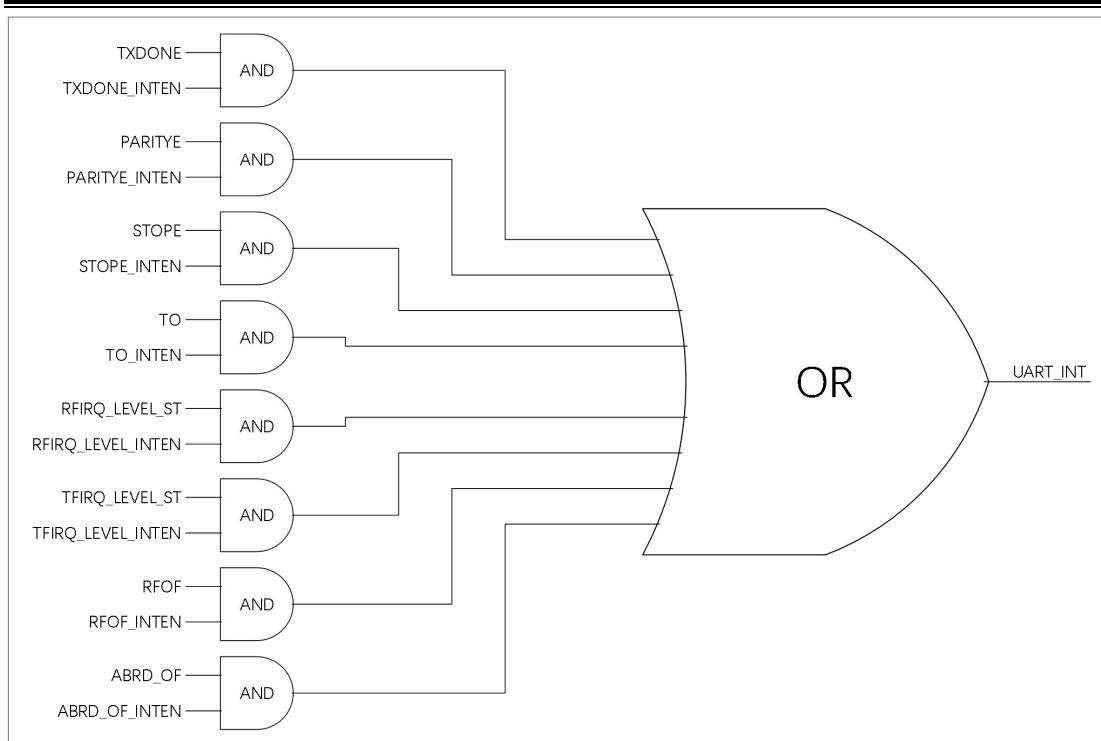
picture5-125 UARTThe second receive timeout interrupt

As shown in FIG:

1timeUARTproduceRXTOreceive time-out interrupt, and at2moment byCPUis cleared until3time agoUART The reception is always idle, but no timeout interrupt will be generated in this interval;3timeUARTReceive the first data, until arrive4After the data is received at the moment, and then enters the idle time, the timeout counter will start counting until5timeout interruption.

to interrupt

UARTprovided8There are two kinds of interrupt sources, and their relationship is shown in the figure below:



picture5-126 UARTInterrupt flag and interrupt diagram

register map

name	Offset	bit width	type	reset value	describe
UART0:	BASE: 0x4006B000				
UART1:	BASE: 0x4006B800				
UART2:	BASE: 0x4006C000				
UART_CTRL	0x00	32	R/W	0x00	UARTcontrol register
UART_BAUD	0x04	32	R/W	0x00	UARTBaud Rate Configuration Register
UART_TDR	0x08	32	W	0x00	UARTwrite data register
UART_RDR	0x0c	32	R	0x00	UARTread data register
UART_IE	0x10	32	R/W	0x00	UARTinterrupt enable register
UART_IF	0x14	32	R/W	0x2400	UARTInterrupt Status Register
UART_FIFO	0x18	32	R/W	0x07	UART FIFOcontrol register
UART_FC	0x1c	32	R/W	0x00	UARTFlow Control Configuration Register
UART_RXTO	0x20	32	R/W	0xff	UARTReceive Timeout Configuration Register

Register description**UART_CTRLregister(0x00)**

bit field	name	type	reset value	describe
31:17	RESERVED	R	0	reserved bit
16:14	TX_DLY	R/W	0	Send delay time setting (used to set the transmission delay time between the last stop bit and the next start bit) 000: no delay 001:1 bitDelay 010:2bitDelay ... 111:7bitDelay
13:12	ABRDBIT	R/W	0	Automatic baud rate detection bit length configuration 11:8Bit duration, from start bit to first rising edge. The input data is0x80. 10:4Bit duration, from start bit to first rising edge. The input data is0x08. 01:2Bit duration, from start bit to first rising edge. The input data is0x02. 00:1Bit duration, from start bit to first rising edge. The input data is0x01.
11	ABRDEN	R/W	0	Auto-baud rate detection enable 1: Auto baud rate detection function enable 0: Automatic baud rate detection function disabled This bit will be automatically cleared to 0 after auto-detection ends.
10:9	RESERVED	R	0	reserved bit

8:7	PARMD	R/W	0	Parity mode selection 11:often0 10:often1 01: even parity 00:Odd parity
6	PAREN	R/W	0	parity bit enable 1: with parity bit 0: without parity bit
5	NINEBIT	R/W	0	9bitdata mode enable 1:9bitdata schema 0:8bitdata schema
4	TXDMAEN	R/W	0	sendDMAtransfer enable 1:expressDMAoperateUARTThe transmit data register 0 :expressCPUoperateUARTThe transmit data register
3	RXDMAEN	R/W	0	take overDMAtransfer enable 1:expressDMAoperateUARTReceive Data Register of 0 :expressCPUoperateUARTReceive Data Register of
2	TXEN	R/W	0	send enable bit 1: Send on. Willtx_fifoThe data saved in theuart_tx send it out. 0: Send off. No data is sent.uart_txsignal hold1.
1	RXEN	R/W	0	receive enable bit 1: Receive is on. accessibleuart_rxreceive incoming data 0: Receive off. do not acceptuart_rxThe data
0	UARTEN	R/W	0	UARTenable bit 1:EnableUARTmodule 0:closureUARTmodule

UART_BAUDregister(0x04)

bit field	name	type	reset value	describe
31:16	RESERVED	R	0	reserved bit

15:0	BAUD	R/W	0	Baud Rate Configuration Data
------	------	-----	---	------------------------------

UART_TDRregister(0x08)

bit field	name	type	reset value	describe
31:9	RESERVED	R	0	reserved bit
8:0	TDR	W	0	write data register The data to be sent is sent by writing to this register FIFO middle. When ready to send data, the shift register directly reads the send FIFO Send data.

UART_RDRregister(0x0C)

bit field	name	type	reset value	describe
31:9	RESERVED	R	0	reserve
8:0	RDR	R	0	read data register The received data is read out through this register. The shift register is in After each transmission is completed, store the data in the receive FIFO, read through this register to receive FIFO data in . Note: If a parity bit error or a stop bit error occurs, the group number will not write RX FIFO. like RX FIFO is not empty, then RX FIFO The data in is valid data.

UART_IERegister(0x10)

bit field	name	type	reset value	describe
31:10	RESERVED	R	0	reserved bit

9	ABRD_OVF	R/W	0	Auto-baud rate detection function Counter overflow interrupt enable
8	RXFIFO_OVF	R/W	0	take overFIFOOverflow interrupt enable
7	TXFIFO	R/W	0	sendFIFOThe data saved in reaches the set water level interrupt enable
6	RXFIFO	R/W	0	take overFIFOThe received data reaches the set water level interrupt enable
5	RXTO	R/W	0	Receive timeout interrupt enable
4	STOPE	R/W	0	Receive data stop bit error interrupt enable
3	PARITYE	R/W	0	Receive data parity error interrupt enable
2	TXDONE	R/W	0	All data transmission complete interrupt enable (indicates that the transmission shift register The data transmission in the middle is completed and the data is sentFIFOThere is no pending data in the
1	RESERVED	R/W	0	reserved bit
0	RESERVED	R/W	0	reserved bit

UART_IFregister(0x14)

bit field	name	type	reset value	describe
31:23	RESERVED	R	0	reserved bit
22:20	TF_LEVEL	R	0	sendFIFOwater level sign 000: Indicates when not fullFIFOThere are0data, when full FIFOhave8data; 001:expressFIFOhave1data; ... 111:expressFIFOhave7data;

19:17	RF_LEVEL	R	0	take overFIFOwater level sign 000: Indicates when not fullFIFOThere are0data, when full FIFOhave8data; 001:expressFIFOhave1data; ... 111:expressFIFOhave7data;
16	TXBUSY	R	0	data sending busy flag 1:sendFIFONot empty, or data is being sent 0 :sendFIFOempty, and no data is being sent
15	TXFIFO_HFULL	R	0	sendFIFOhalf full sign
14	TXFIFO_FULL	R	0	sendFIFOfull sign
13	TXFIFO_EMPTY	R	0x1	sendFIFOempty sign
12	RXFIFO_HFULL	R	0	take overFIFOhalf full sign
11	RXFIFO_FULL	R	0	take overFIFOfull sign
10	RXFIFO_EMPTY	R	0x1	take overFIFOempty sign
9	ABRD_OVF	R/W	0	Auto-baud rate detection function counter overflow flag 1: counter overflow, detection failed 0: The counter has not overflowed Write1clear
8	RXFIFO_OVF	R/W	0	take overFIFOoverflow flag Write1clear Note: The data after overflow will be discarded
7	TXFIFO	R	0	sendFIFOWhen the data saved in reaches the set water level, this bit is 1, otherwise0
6	RXFIFO	R	0	take overFIFOWhen the received data reaches the set water level, this bit 1, otherwise0
5	RXTO	R/W	0	receive timeout flag 1: send and receive timeout Write1clear

4	STOPE	R/W	0	Stop bit error in received data Write1clear Mainly means that the stop bit is not received within the expected time period or recognized, the stop bit error is considered
3	PARITYE	R/W	0	Received data has a parity error Write1clear
2	TXDONE	R/W	0	All data transmission is completed (indicates that the data in the shift register is sent send complete and send dataFIFOThere is no pending data in the Write1clear
1	RESERVED	R/W	0	reserved bit
0	RESERVED	R/W	0	reserved bit

UART_FIFOREgister(0x18)

bit field	name	type	reset value	describe
31:8	RESERVED	R	0	reserved bit
7	TF_CLR	R/W	0	TXFIFOclear enable software write1clear sendFIFO, the hardware automatically clears the
6	RF_CLR	R/W	0	RXFIFOclear enable software write1clear receiveFIFO, the hardware automatically clears the
5:3	TF_LEVEL	R/W	0	for TXFIFOWater level settings that generate interrupts 000:0 001:1 010:2 ... 111:7 means to sendFIFOThe number of data to be sent is not more than the water level setting value For example, set the value to 011, but TXFIFOThe number of data written in is less than or equal to 3 corresponding signal



2:0	RF_LEVEL	R/W	0x7	<p>for RXFIFO Water level settings that generate interrupts</p> <p>000:1 001:2 010:3 ... 111:8</p> <p>express acceptance FIFO The number of data received in at least reaches the water level setting value</p> <p>For example, set the value to 011, but RXFIFO data in at least 4 corresponding signal</p>
-----	----------	-----	-----	--

UART_FCregister(0x1C)

bit field	name	type	reset value	describe
31:6	RESERVED	R	0	reserved bit
5	RTS_SIGNAL	R	0	<p>means online RTS state</p> <p>1:RTS is high level 0:RTS low level</p>
4	CTS_SIGNAL	R	0	<p>means online CTS state</p> <p>1:CTS is high level 0:CTS low level</p>
3	RTSPOL	R/W	0	<p>RTS Signal Polarity Configuration</p> <p>1:RTS signal output is high when the UART can receive data; RTS signal output is low when the UART takes over FIFO when it is full, no more data can be received.</p> <p>0:RTS signal output is low when the UART can receive data; RTS signal output is high when the UART takes over FIFO when it is full, no more data can be received.</p>
2	CTSPOL	R/W	0	<p>CTS Signal Polarity Configuration</p> <p>1:CTS signal input is high when the UART data can be sent; CTS When the signal input is low, the UART No data is emitted.</p> <p>0:CTS When the signal input is low, the UART data can be sent; CTS signal input is high when the UART No data is emitted.</p>

1	RTSEN	R/W	0	RTSflow control enabled 1:RTSSignals play a fluidic role 0 :RTSsignal doesn't work
0	CTSEN	R/W	0	CTSflow control enabled 1:CTSSignals play a fluidic role 0 :CTSsignal doesn't work

UART_RXTOregister(0x20)

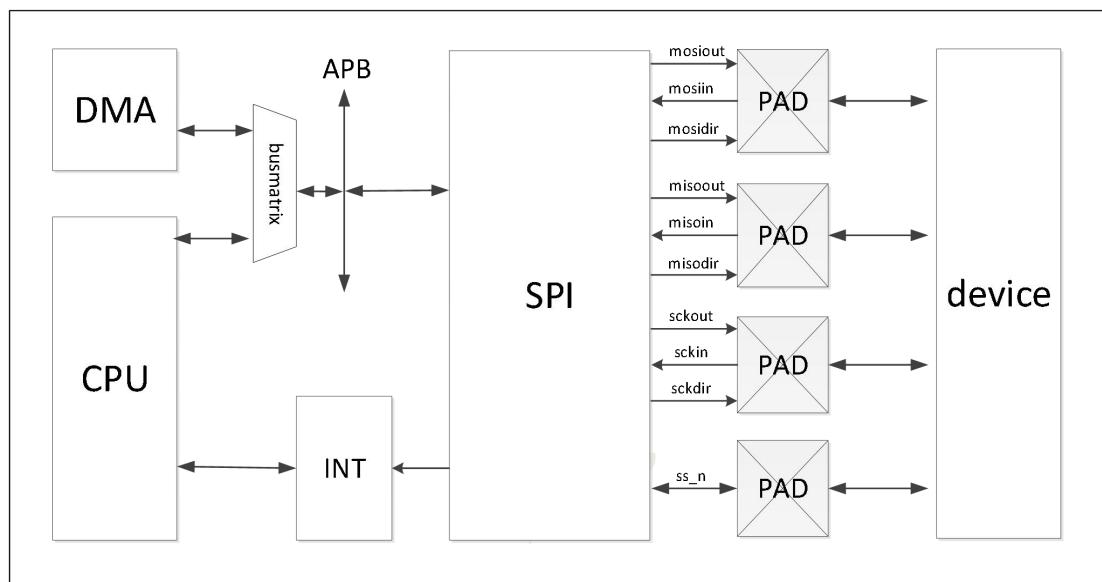
bit field	name	type	reset value	describe
31:8	RESERVED	R	0	reserved bit
7:0	RXTO	R/W	0xff	Receive data timeout trigger comparison value when RXFIFOWhen a new data is received, the timer is cleared and starts counting again (the timing clock is the duration of one data). When the timer exceeds the configured value, no A valid start bit generates a receive timeout flag.

5.17 SPIbus controller (SPI)

5.17.1 overview

Serial Peripheral Interface (Serial Peripheral Interface, SPI) is an external device via 2 line switching 8 string of bits

Synchronous means of communication. chip provides a SPI interface module can be configured as a master device or a slave device to achieve communication with external SPI communication.



picture5-127 SPIModule System Block Diagram

The picture above is SPI schematic diagram of the connection of the module system. in CPU accessible APB Bus interface direct control SPI module,

SPI module through PAD realize the communication between the chip and the external equipment. Depending on the configured mode, SPI modules can be configured for each PAD direction.

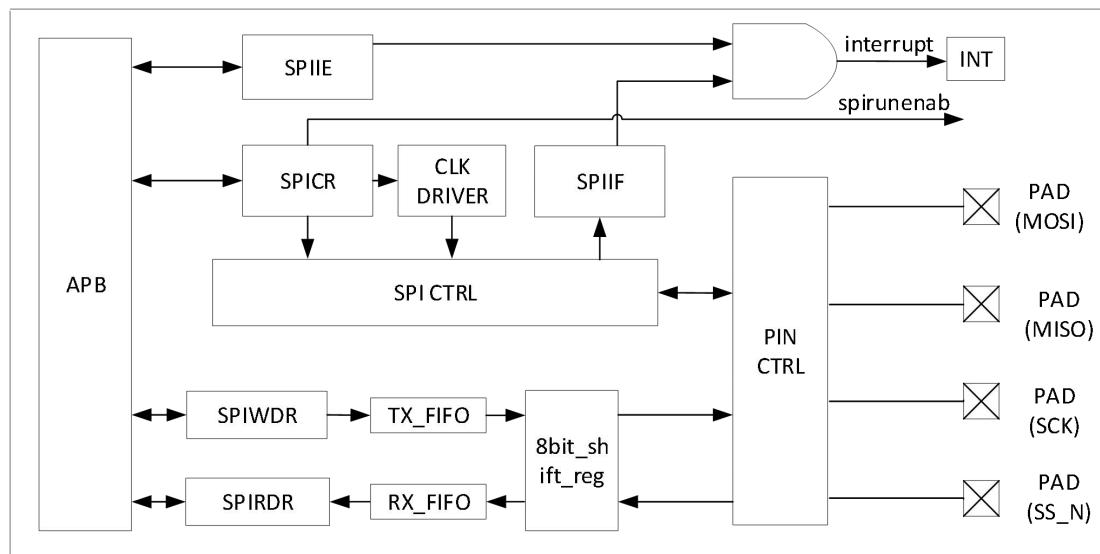
5.17.2 characteristic

- Support master mode and slave mode
- Programmable clock polarity and phase
- The master mode rate is configurable, the maximum frequency is the system clock 4 times frequency division
- Data transfer sequence is configurable

-transfer end interrupt flag

- The read data register and the write data register are separated
- Receiving and sending respectively adopt8classFIFOcaching mechanism
- haveDMAtransmission interface

5.17.3 Block Diagram of Module Structure



picture5-128 SPIBlock Diagram of Module Structure

As shown in the figure above, for SPI schematic diagram of the module structure, available through SPI control register SPICR to configure the main module

Slave mode, transmission mode, FIFO clear and other functions etc. data write through SPIWDRThe operation is complete and ready to send data, SPIThe data to be sent via SPIWDRRegister Write Send FIFO, the shift register reads FIFO of Value, send data; read data through SPIRDRoperation is complete, the shift register is completed after each transfer, store data in receiveFIFOin, through SPIRDRRegister Read Receive FIFO data in . The interrupt signal of the module is affected by the interrupt enable register SPIIIEControl, turn on the interrupt enable, and an interrupt signal will be generated when there is a corresponding interrupt state.

5.17.4 Functional description

SPI interface timing

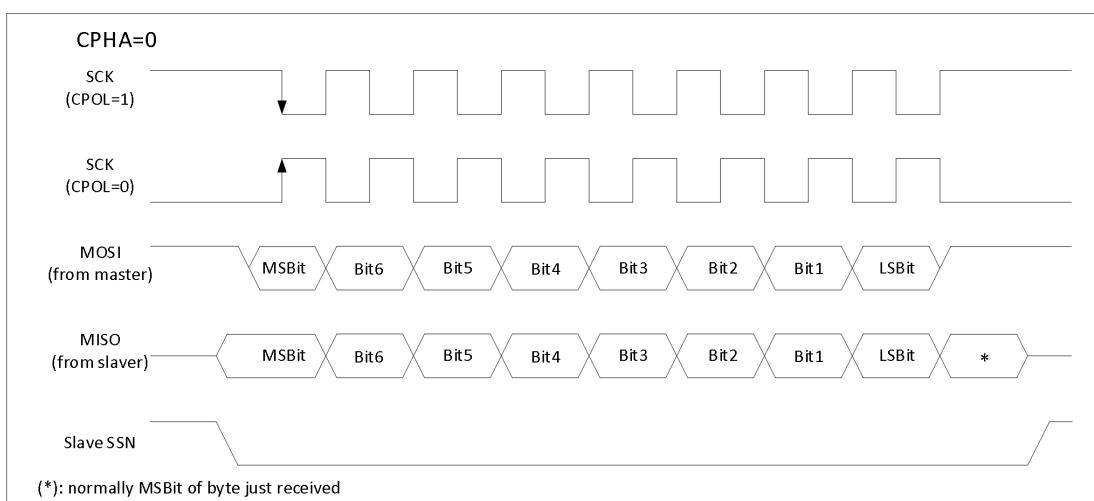
different SPI peripherals, SPI The timing of the serial clock can be controlled by the clock phase select bit (SPICR.CPHA) and the clock pole select bit (SPICR.CPOL) set generate 4 different combinations. In order to ensure the correct transmission of data, the timing configuration of the master and slave devices must be consistent.

device mode or SPI system Enable Bit (SPICR.SPE) bit for 0 hour, SPI of SCK Pin has no serial clock output.

1: CPHA=0 hour, SPI The module samples data on the first jump edge of the serial clock, namely:

like CPOL=1, sample data on the falling edge of the serial clock;

like CPOL=0, data is sampled on the rising edge of the serial clock. As shown below:

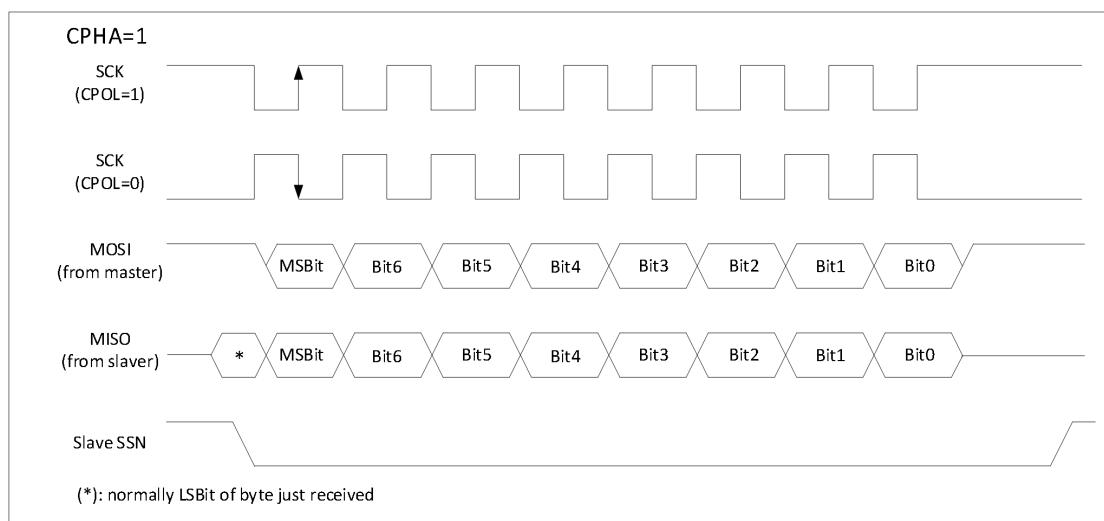


picture5-129 SPI Data/Clock Timing Diagram (CPHA=0)

2: CPHA=1 hour, SPI The module samples data on the second transition edge of the serial clock, namely:

like CPOL=1, sample data on the rising edge of the serial clock;

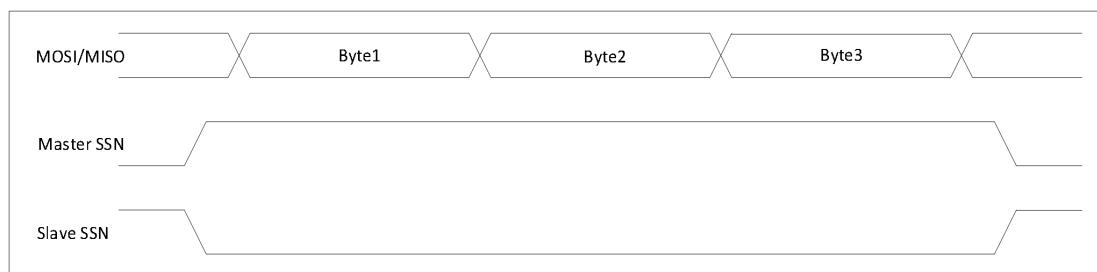
like CPOL=0, data is sampled on the falling edge of the serial clock. As shown below:



picture5-130 SPI Data/Clock Timing Diagram (CPHA=1)

3: SlaveSSN

like SPI for the slave device, the slave device's SSN pin can be held low during continuous data transfers. As shown below:



picture5-131 SPI SSN timing diagram (CPHA=0)

I/O configuration

Master output, slave input (MOSI)

master out slave in (MOSI) pin is the output of the master and the input of the slave for the master-to-slave serial data transmission. When SPI is configured as a master device, this pin is an output. When SPI is configured as a slave, this pin is an input.

Master input, slave output (MISO)

master in slave out (MISO) pin is the output of the slave and the input of the master for the serial connection from the slave to the master data transmission. When configured as a master, this pin is an input when the SPI. When configured as a slave, this pin is an output.

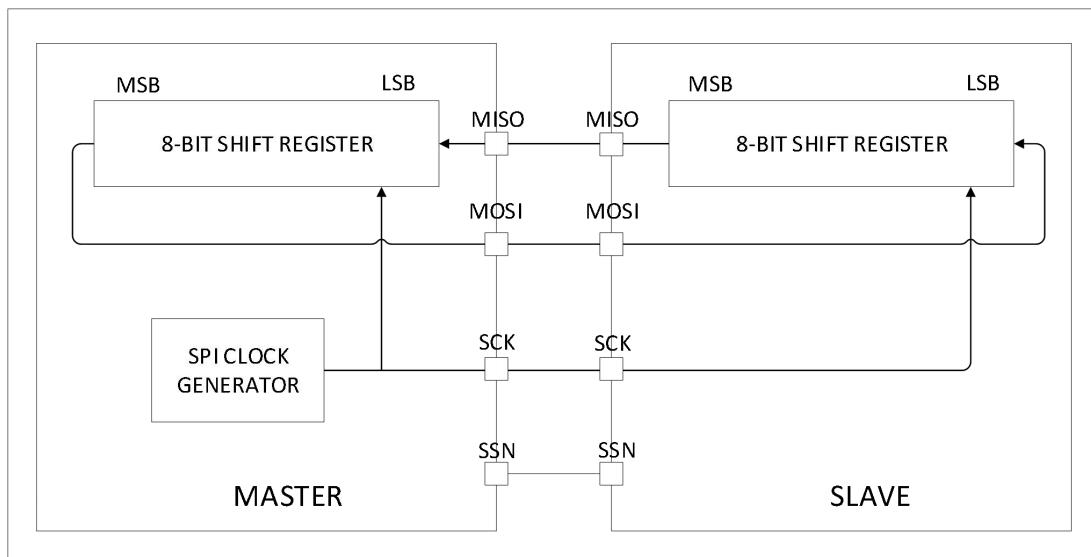
Serial Clock (SCK)

Serial Clock (SCK) pin is the output of the master and the input of the slave, and is used to synchronize the exist MOSI and MISO serial data transmission over the wire. When configured as a master device, this pin outputs the clock when the SPI. When configured as a slave, this pin is an input.

Select from (SSN)

Select from (SSN) pin is used to control slave device selection, when the SPI. When configured as a master device, the SSN pins can be accessed via The way of the register controls whether the slave device is selected or not, when the SPI. When configured as a slave, the SSN pin derived from the control of the master device signal.

SPI. The connection of the master and slave devices is shown in the figure below:



picture5-132 SPI Master/SPI Slaveinterconnection

master-slave deviceMOSI,MISO,SCKandSSNrespectively connected together. master-slave throughMOSI,MISOEven into A loop, the master device outputs the clock, and during data transfer, the master device passesMOSIoutput data from the device through theMISO Output Data. After one byte of data has been transferred, the master and slave devices will exchange8Bit shift register value.

Data transfer configuration

- 1: It needs to be configured before data transmissionSPICR.SPEbit andSPICR.MSTRbit to enableSPIand set the master-slave mode Mode.
- 2: configurationSPICR.CPHAbit andSPICR.CPOLbit to set the serial clock phase and polarity (Master and Slave need to match).
- 3: configurationSPICR.SPR[2:0]bit to set the serial clock baud rate (if it is a slave device mode, it is not necessary to set, the serial The clock rate is determined by the master device).

4: configurationSPICR.LSBbits, set transfer order, configureSPICR.CPHA_DATAHOLD_S, to set the slave mode The number of transfer data retention cycles.

When needed, configure interrupts, configureSPIIEandSPIIRbit.

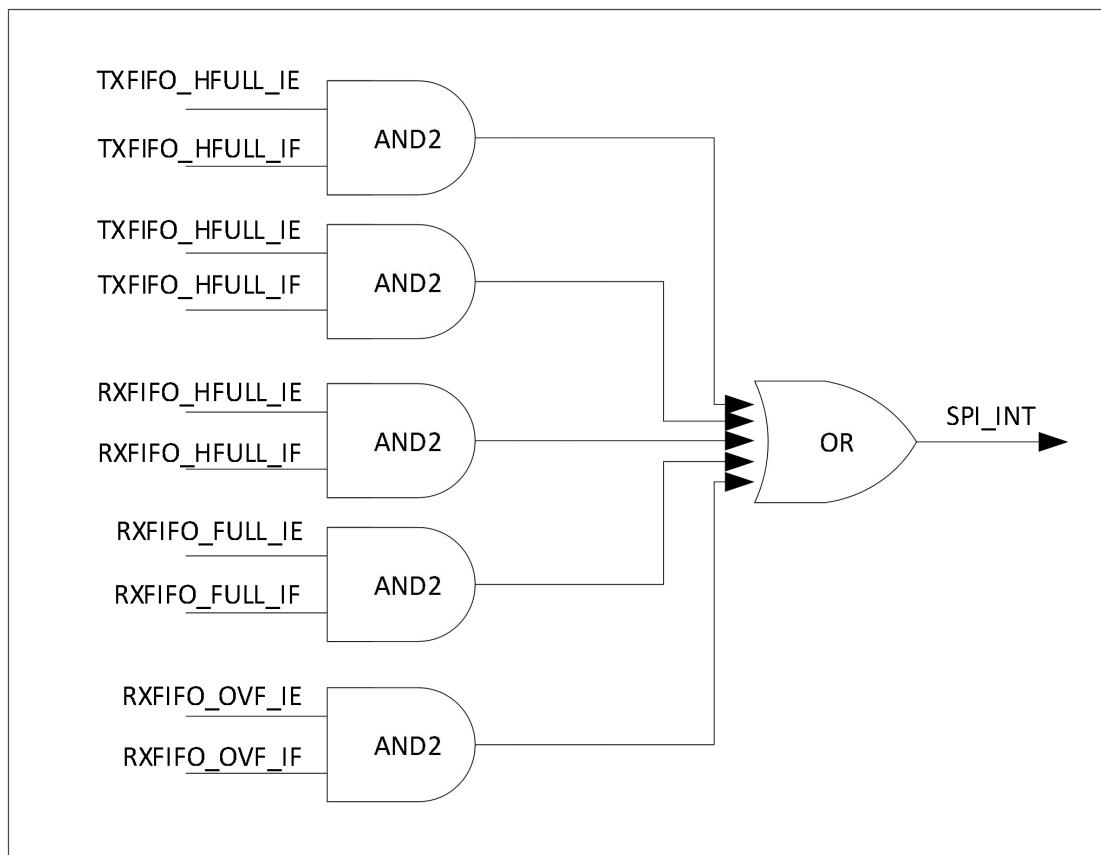
Before data transmission in master device mode, the slave device'sSSNpin pulled low. In master modeMCUwriteSPIWDR Register action initiates data transfer, interrupt flagSPIIFSet up to complete the data transfer.

Slave mode handling is special when CPHA=0, the slave device's SSN pin low to initiate a data transfer, the slave piece of SSN pin high to end the data transfer (even before SPIR.SPIF interrupt has been generated), because the slave device does not know When does the channel transfer start, when SSN After the pin is pulled low, the MISO pin begins data immediately MSB transmission.

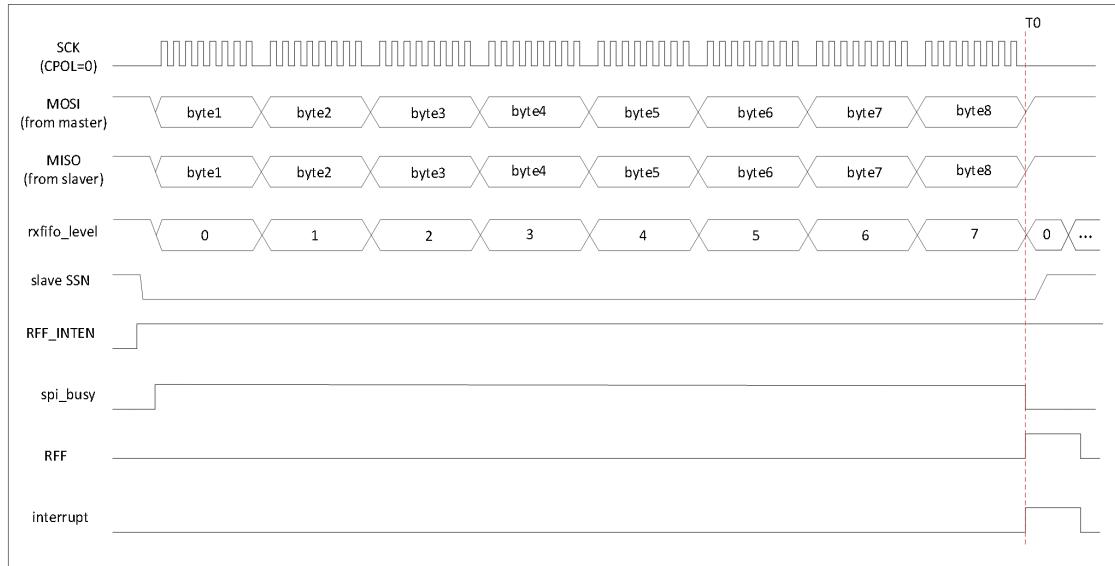
when CPHA=1, the slave device initiates a data transfer on the first edge of the serial clock, at SPIR.SPIF end after set data transmission.

interrupt generated

SPI module provides 5 The schematic diagram of the interrupt source is as follows:



picture5-133 SPI interrupt flag and interrupt diagram

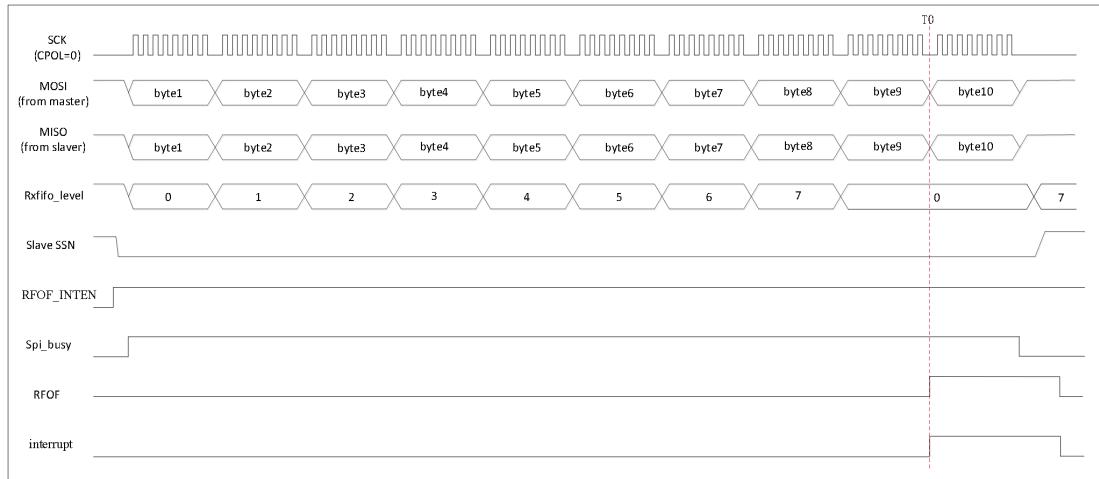
take overFIFOfull interrupt


picture5-134 SPI take overFIFOfull interrupt

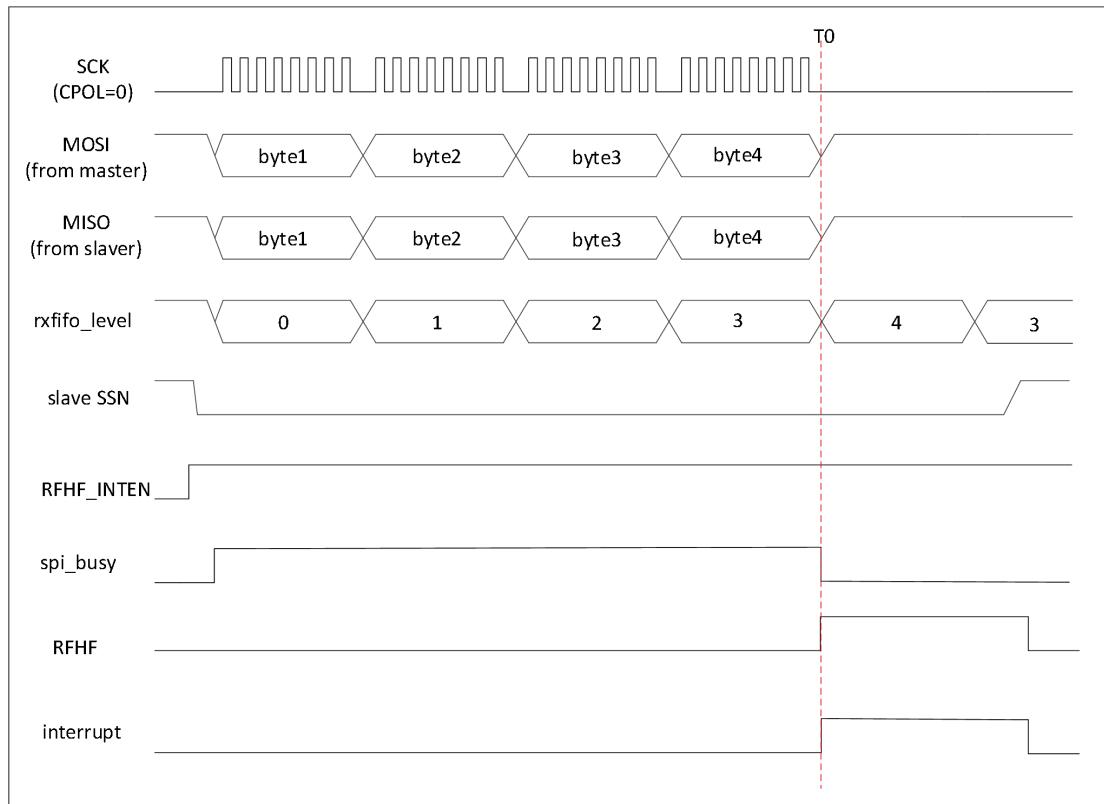
During the data transmission process of this module, the received data will be saved to the receiver through the shift registerFIFOIn, enable to connect receiveFIFOFull Interrupt Enable RegisterRFF_INTEN, this module'sFIFODepth is8, so whenFIFOThe data arrives in 8when receivingFIFOfull state signalRFFwill be set1That is, in the above figureT0time, and generate an interrupt signal. requires attention Yes, whenFIFOThe data in is not read away,FIFOWhen always full, the receive full status signal will always remain high, even if the status After the state is cleared, it will be set again1,onlyFIFOThe data in is read away,FIFOIn the dissatisfied state, the full state will is cleared, and atSPIDuring data transfer,SPIThe transfer busy flag will remain high all the time.



take overFIFOoverflow interrupt



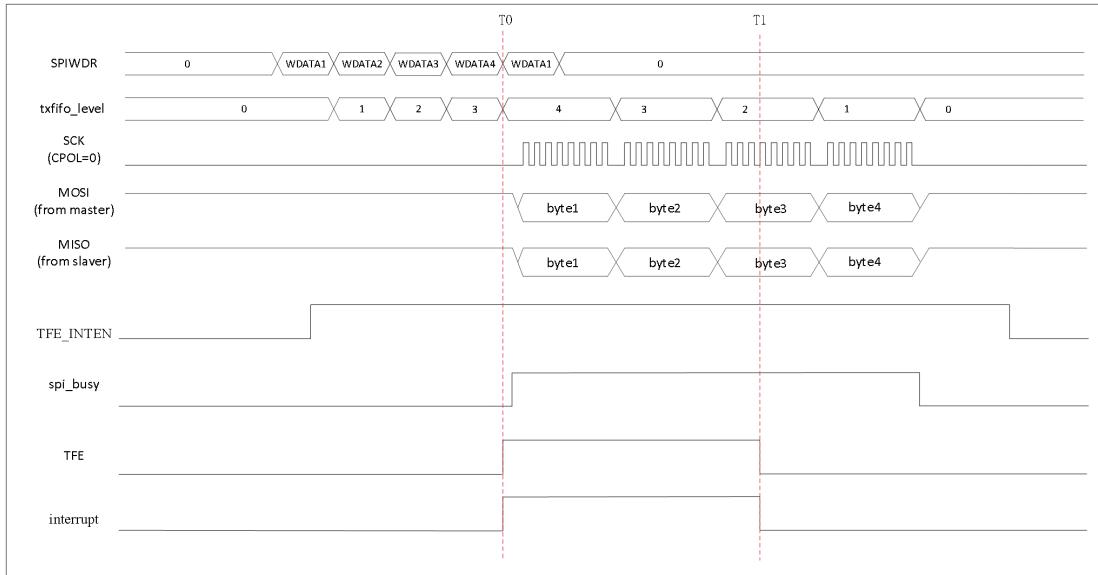
During the data transmission process of this module, the received data will be saved to the receiver through the shift registerFIFOIn, enable to connect receiveFIFOOverflow Interrupt Enable RegisterRFOF_INTEN, this module'sFIFODepth is8, so whenFIFOnumber in According to arrival8After that, continue to receiveFIFOtransfer data inT0Always receiveFIFOOverflow status signalRFOFWill be place1, and generate an interrupt signal. It should be noted that whenFIFOThe data in is not read away,FIFOWhen always full, continue TowardsFIFOTransmit data, the data will not be written, it will be lost, receiveFIFOThe overflow status signal is always maintained high, it will be set again even after the state is cleared1, and inSPIDuring data transfer,SPIThe transfer busy flag will be one stay high all the time.

take overFIFOhalf full interrupt


picture5-135 SPI take overFIFOhalf full interrupt

During the data transmission process of this module, the received data will be saved to the receiver through the shift register FIFOIn, enable to connect receiveFIFOHalf Full Interrupt Enable Register RFHF_INTEN, this module's FIFO Depth is 8, so when FIFO data in arrive4 when receiving FIFO half full status signal RFF will be set1 That is, in the above figure T_0 time, and generate an interrupt signal. need Note that when FIFO The number of data in is greater than or equal to 4 When the receive half-full status signal will remain high, even if the status After the state is cleared, it will be set again1, only FIFO The data in is read away, FIFO When the data in the middle is less than half full, half-full status is cleared, and the SPI During data transfer, SPI The transfer busy flag will remain high all the time.

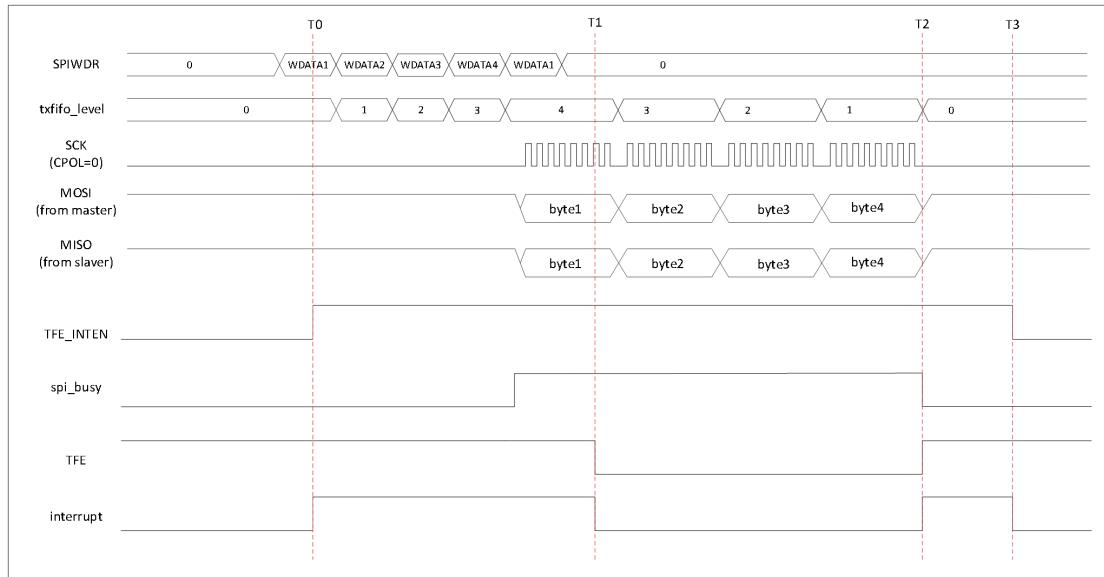
sendFIFOhalf full interrupt



picture5-136 SPIsendFIFOhalf full interrupt

During the data transmission process of this module, the sent data will pass through SPIWDRRegister saved to transmitFIFOin, make can sendFIFOHalf Full Interrupt Enable RegisterTFE_INTEN, this module'sFIFOdepth is8, so when sendingFIFOmiddle The amount of data arrived4when, sendFIFOhalf full status signalTFEwill be set1That is, in the above figureT0moment, and generates an interrupt Signal. Note that when sendingFIFOThe data in is not read away,FIFOWhen it is always greater than or equal to half full, send full The status signal will always remain high, even after the status is cleared it will be set again1,onlyFIFOThe data in is read away,FIFO When the state is less than half full, the full state will be cleared, as shown in the figureT1moment, and atSPIDuring data transfer, SPIThe transfer busy flag will remain high all the time.

sendFIFOempty interrupt



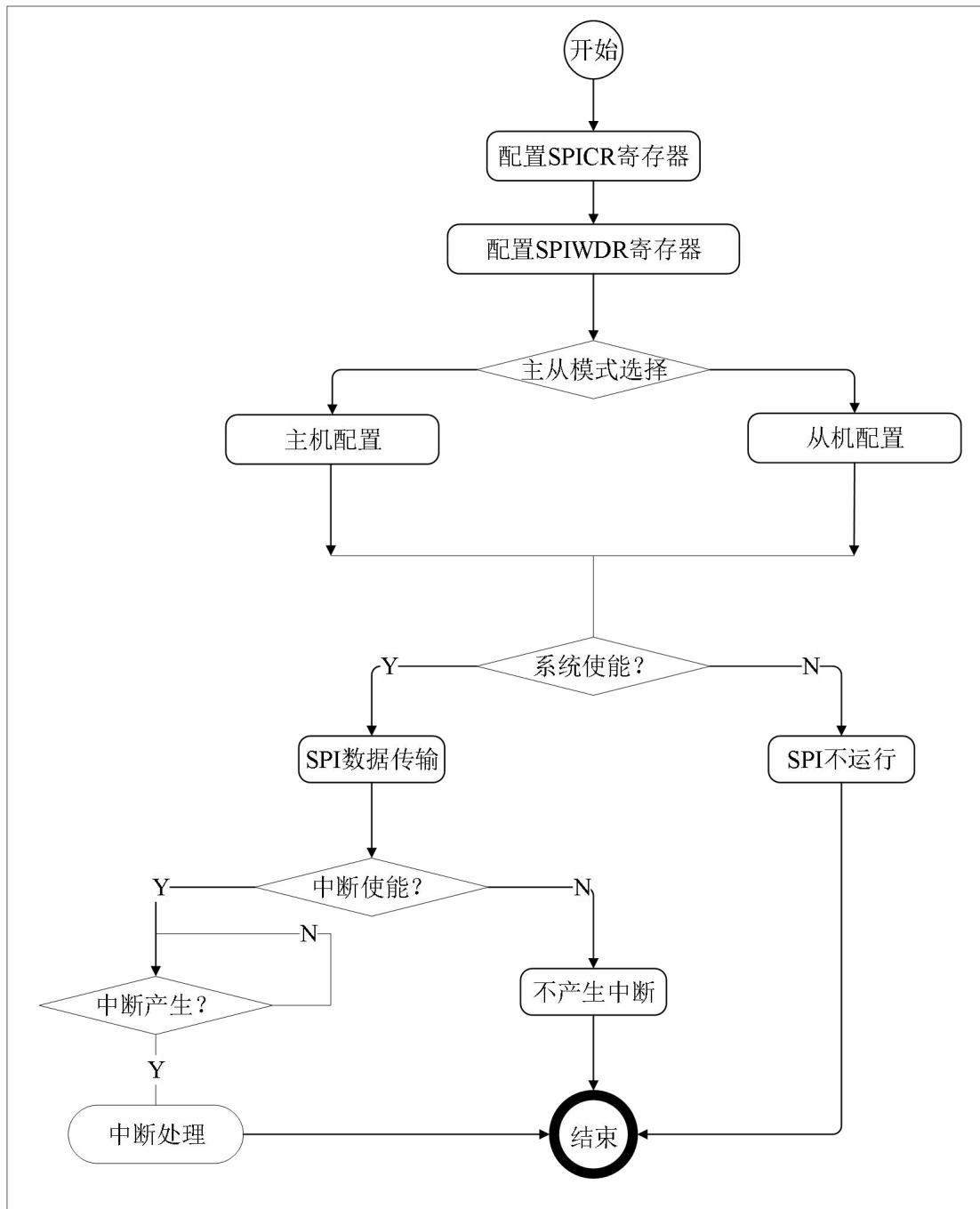
picture5-137 SPI sendFIFOempty interrupt

During the data transmission process of this module, the sent data will pass through SPIWDRRegister saved to transmitFIFOin, pass sent when output starts FIFO. There is no data in, so an empty status signal is sent TFE for 1, T0 Always enable sending FIFO half full interrupt enable register TFE_INTEN, an interrupt signal will be generated; when sending to FIFO After writing data in, and empty state OK cleared, T1 Time empty state and send empty interrupt signal will be set 0; T2 moment, send FIFO data is again Read empty, send empty status will be set again 1, and generate an interrupt signal; T3 Always close the send empty interrupt enable, it will no longer Generate an interrupt signal. Note that when sending FIFO There is no data consistent in FIFO When it is always empty, send a full status The status signal will always remain high, even after the status is cleared, it will be set again 1, only FIFO There is data written in, FIFO The empty state is only cleared in the non-empty state, and in SPI During data transfer, SPI The transfer busy flag will be kept hold high.

DMAcontrol

DMAable to passDMAControl Enable RegisterTXDMAENandRXDMAENTo realize the control of this module, TXDMAENfor1when, meansDMAoperateSPIThe transmit data register;RXDMAENfor1when, meansDMAHold doSPIthe receive data register.

Operating procedures



picture5-138 SPIOperating procedures

-configurationSPImodule clock

-portThe port is configured asSPIFunction

- configurationSPImaster-slave mode
- configurationSPIClock Phase and Polarity
- Configure transfer order
- If the main mode also needs to be configuredSPIbaud rate
- Configure interrupt enable
- configurationSPIsystem enableSPI, start transferring data

register map

name	Offset	bit width	type	reset value	describe
SPI0: BASE: 0x400B8000					
SPI1: BASE: 0x400B8800					
SPICR	0x00	32	R/W	0x1010	SPIcontrol register
SPIWDR	0x04	32	R/W	0x00	SPIwrite data register
SPIRDR	0x08	32	R	0x00	SPIread data register
SPIIE	0x10	32	RW	0x00	SPIinterrupt enable register
SPIIF	0x14	32	R/W	0x8	SPIInterrupt Status Register
SPIFIFO	0x18	32	R/W	0x9	SPIFIFOstatus register

Register description

SPICRregister(0x00)

bit field	name	type	reset value	describe
31:17	RESERVED	R	0	reserved bit
16	TF_CLR	R/W	0	sendFIFOclear bit software write1clear sendFIFO, the hardware automatically clears the
15	RF_CLR	R/W	0	take overFIFOclear bit software write1clear receiveFIFO, the hardware automatically clears the
14	TXDMAEN	R/W	0	sendDMAcontrol enable bit 1:expressDMAoperateSPIThe transmit data register 0 :expressCPUoperateSPIThe transmit data register
13	RXDMAEN	R/W	0	take overDMAcontrol enable bit 1:expressDMAoperateSPIReceive Data Register of 0 :expressCPUoperateSPIReceive Data Register of
12	MSR_SSN	R/W	1	in master modeSSNoutput, by default output1 This register is valid only in master mode
11:8	CPHA_DATA HOLD_S	R/W	0	slave modeCPHAfor1, the data hold time configuration register 0000:1indivualpclk 0001:2indivualpclk ... 1111:16indivualpclk
7	LSB	R/W	0	Data Transfer Sequence Selection 0:MSB 1:LSB
6	MSTR	R/W	0	Master-slave mode selection 0 = SPISystem configured in slave mode 1 = SPISystem Configuration in Master Mode
5	CPOL	R/W	0	Clock Polarity Selection 0 =Serial clock idle state is low level, active level is high level 1 = Serial clock idle state is high level, active level is low level
4	CPHA	R/W	1	Clock Phase Selection 0 =Data is sampled on the first transition edge of the serial clock 1 =Data is sampled on the second transition edge of the serial clock

3	SPE	R/W	0	SPIsystem enable 0 = SPIsystem shutdown 1 = SPIsystem enable
2	SPR2	R/W	0	SPIbaud rate selection bits2
1	SPR1	R/W	0	SPIbaud rate selection bits1
0	SPR0	R/W	0	SPIbaud rate selection bits0

SPR0,SPR1,SPR2Indicates the choice of baud rate:

SPR2	SPR1	SPR0	Fsck	Fsck(Fcpu=48Mhz)
0	0	0	Fpclk/4	12MHz
0	0	1	Fpclk/8	6MHz
0	1	0	Fpclk/16	3MHz
0	1	1	Fpclk/32	1.5MHz
1	0	0	Fpclk/64	750KHz
1	0	1	Fpclk/128	375KHz
1	1	0	Fpclk/256	187.5KHz
1	1	1	Fpclk/512	93.75KHz

SPIWDRregister(0x04)

bit field	name	type	Reset value	description
7:0	SPIWDR	R/W	0	SPIThe data to be sent viaSPIWDRRegister Write SendFIFO middle. When ready to send data, the shift register directly reads the send FIFO Send data.

SPIRDRregister(0x08)

bit field	name	type	Reset value	description
7:0	SPIRDR	R	0	SPIpassSPIRDRThe register reads the received data. After each transmission, the shift register stores the data in the receiveFIFOin, through SPIRDRRegister Read ReceiveFIFOdata in .

SPIIRegister(0x10)

bit field	name	type	reset value	describe
31:5	RESERVED	R	0	reserved bit
4	TXFIFO_HFULL	R/W	0	sendFIFOhalf full interrupt enable
3	TXFIFO_EMPTY	R/W	0	sendFIFOempty interrupt enable
2	RXFIFO_HFULL	R/W	0	take overFIFOhalf full interrupt enable
1	RXFIFO_FULL	R/W	0	take overFIFOfull interrupt enable
0	RXFIFO_OVF	R/W	0	take overFIFOoverflow interrupt enable

SPIIFregister(0x14)

bit field	name	type	reset value	describe
31:5	RESERVED	R	0	reserved bit
4	TXFIFO_HFULL	R/W	0	sendFIFOhalf full sign Write1clear
3	TXFIFO_EMPTY	R/W	1	sendFIFOempty sign Write1clear
2	RXFIFO_HFULL	R/W	0	take overFIFOhalf full sign Write1clear



1	RXFIFO_FULL	R/W	0	take overFIFOfull sign Write1clear
0	RXFIFO_OVF	R/W	0	take overFIFOoverflow flag Write1clear Note: Data after overflow will be discarded.

SPI FIFO ST register(0x18)

bit field	name	type	reset value	describe
31:12	Reserved	R	0	reserved bit
11:9	TF_LEVEL	R	0	sendFIFOwater level status 000: Indicates when not fullFIFO have 0 data, when fullFIFO have 8 data; 001: expressFIFO have 1 data; 010: expressFIFO have 2 data; 011: expressFIFO have 3 data; 100: expressFIFO have 4 data; 101: expressFIFO have 5 data; 110: expressFIFO have 6 data; 111: expressFIFO have 7 data;
8:6	RF_LEVEL	R	0	take overFIFOwater level status 000: Indicates when not fullFIFO have 0 data, when fullFIFO have 8 data; 001: expressFIFO have 1 data; 010: expressFIFO have 2 data; 011: expressFIFO have 3 data; 100: expressFIFO have 4 data; 101: expressFIFO have 5 data; 110: expressFIFO have 6 data; 111: expressFIFO have 7 data;
5	TFHF	R	0	sendFIFO half full sign
4	TFF	R	0	sendFIFO full sign

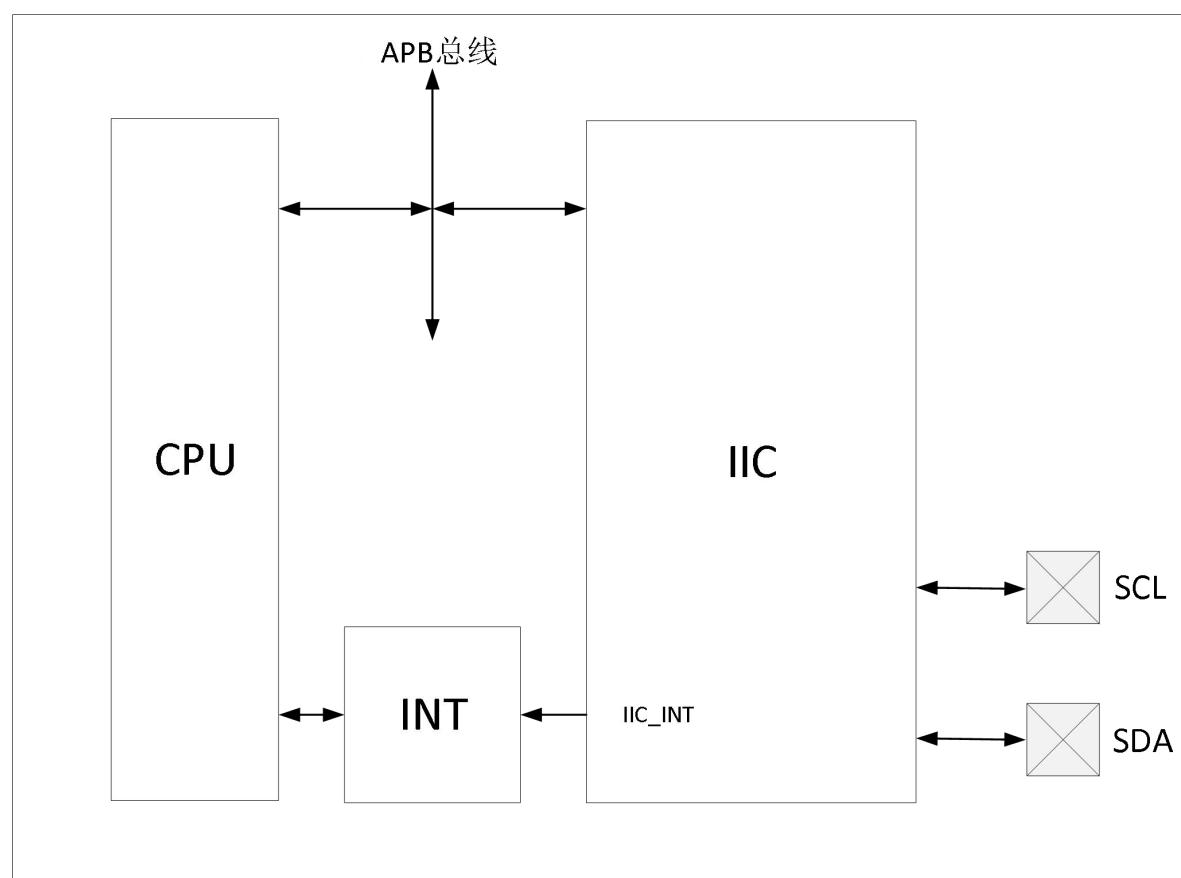
3	TFE	R	1	sendFIFOempty sign
2	RFHF	R	0	take overFIFOhalf full sign
1	RFF	R	0	take overFIFOfull sign
0	RFEs	R	1	take overFIFOempty sign

5.18 IICcontroller (IIC)

5.18.1 overview

IIC (Inter-Integrated Circuit) is a serial communication bus that uses a multi-master-slave architecture. The bus is physically connected very simple, respectively by SDA(serial data line) and SCL(serial clock line) and a pull-up resistor. The principle of communication is through right SCL and SDA line high and low level timing control, to generate the signals required by the bus protocol carry out data transfer. When the bus is idle, these two lines are generally pulled high by the pull-up resistors connected above, and maintain a high level. Communication party The mode is half-duplex, only one SDA line, only one-way communication is possible at the same time. The bus data transfer rate is in standard mode up to 100kbit/s, in fast mode up to 400kbit/s. general pass IIC bus interface programmable clock to implement The adjustment of the output rate is also related to the resistance value of the connected pull-up resistor. chip provides a IIC interface module to implement with the external IIC device communication.

The picture below is the system schematic diagram of the module, this module is through APB bus to implement with CPU connected, and can produce Student interruption.



5.18.2 characteristic

- support the master, slave two modes
- support IIC Input Signal Digital Filtering
- support 3 modes: Standard-mode(100kbps), Fast-mode(400kbps), Fast-mode Plus(1Mbps)
- SCL/SDA online data readable
- Supports multiple interrupts

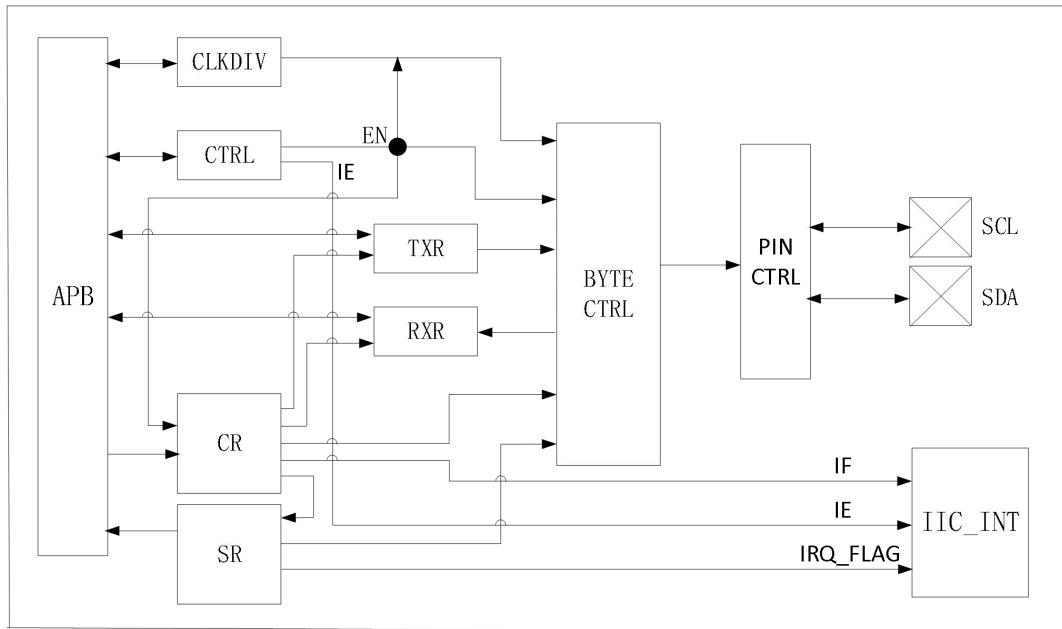
MasterPattern features:

- support SCL LOW Timeout alarm
- issued by support SCL. The clock period is at most $(2^{17}) * \text{system clock}$
- SCL Configurable clock duty cycle

SlavePattern features:

- support 7bit, 10bit two address modes
- support address mask, one slave A device can occupy multiple addresses; 7bit address mode, as a slave device most How much can be occupied 128 addresses; 10bit address mode, as a slave The device can take up to 256 addresses
- support clock stretching, slave The device can be pulled low by SCL Come hold bus

5.18.3 Block Diagram of Module Structure



picture5-139 IIC Block Diagram of Module Structure

IIC
The structure of the module is shown in the figure above.
REG
The module implements register reading and writing and generates interrupts;
IO
module pair SDA, SCL
The input is synchronized and digitally filtered to SDA, SCL output to select;
MASTER
The module implements the host function;
SLAVE
The module implements the slave function;
TIMER
Implement timing functions, including START SCL hold time, SCL low level length timing, SCL high level length timing, when sending data SDA data hold time timing, SCL low level length timeout timing, etc.

5.18.4 Functional description

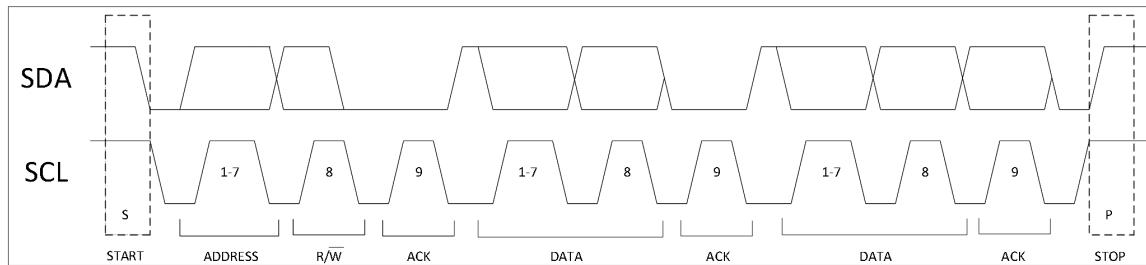
Protocol introduction

IIC
The bus uses serial data lines (SDA) and the serial clock line (SCL) transfer data.

 Data passes between master and slave devices SCL clock signal at SDA synchronous transmission byte by byte on the data line. Every SCL clock pulse sends one bit of data, MSB first. A response signal is generated every time a byte of data is sent. at the time clock line SCL Each bit of data is sampled during the high period. data line SDA on the clock line SCL Change to low level change, in the clock line SCL Remains stable when high.

Usually, a standard communication consists of four parts: start signal, slave address, data transmission, stop signal

Number. As shown below:



picture5-140 IICData Transfer Diagram

start signaling

When the bus is free, it means that no host device occupies the bus (SCLandSDAboth remain high) , the host can

Start the transfer by sending a start signal. start signal, commonly referred to as Sbit.SCLWhen high, the SDA Depend on

Transition from high level to low level. A start signal indicates the start of a new data transfer.

Slave address sending

After the start signal, the first byte of data transmitted by the master is the slave address. Include 7 bits of the slave address and 1 bit R/W indicator bit. R/W The indicator bit signal indicates the direction of data transmission with the slave, 0 indicates a write operation, 1 means read operate. Slaves in the system cannot have the same address. Only when the slave address matches the address sent by the master can generate an acknowledge bit (pulled low on the ninth clock cycle SDA) to respond.

data transmission

Once the slave address has been successfully obtained, the master can pass the R/W bits control sending data byte by byte per transmission.

Each byte requires an acknowledge bit on the ninth clock cycle.

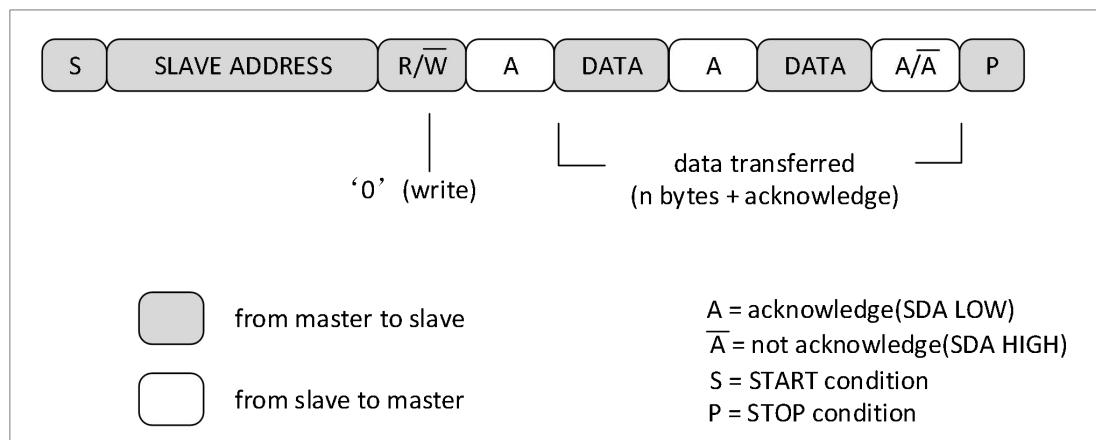
If the slave signal is invalid or the slave returns a NACK signal, the host can generate a STOP signal to abort the number of data transmission.

If the master acts as a receiving device and does not respond to the slave, the slave will release SDA, the host generates a stop signal.

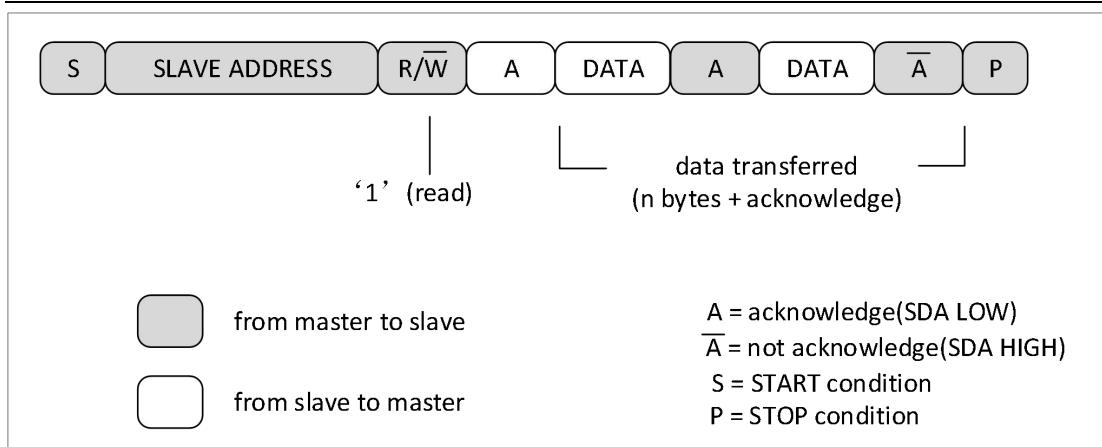
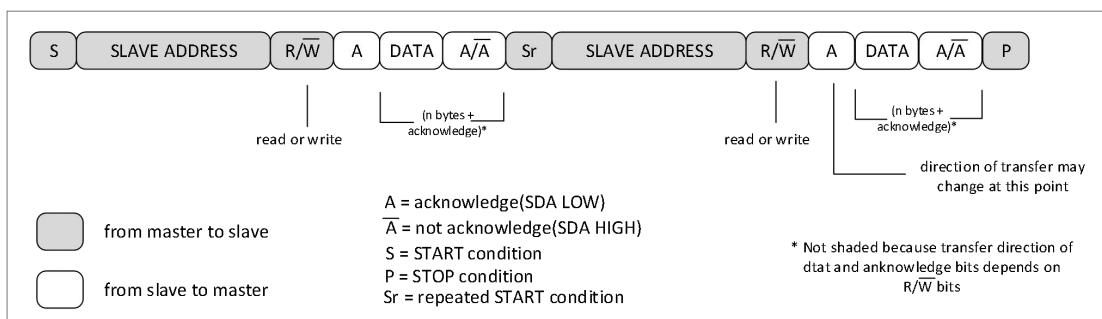
stop signaling

The master can terminate communication by generating a stop signal. The stop signal is often referred to as Pbit, is defined as SCL When high, the SDA transition from low level to high level.

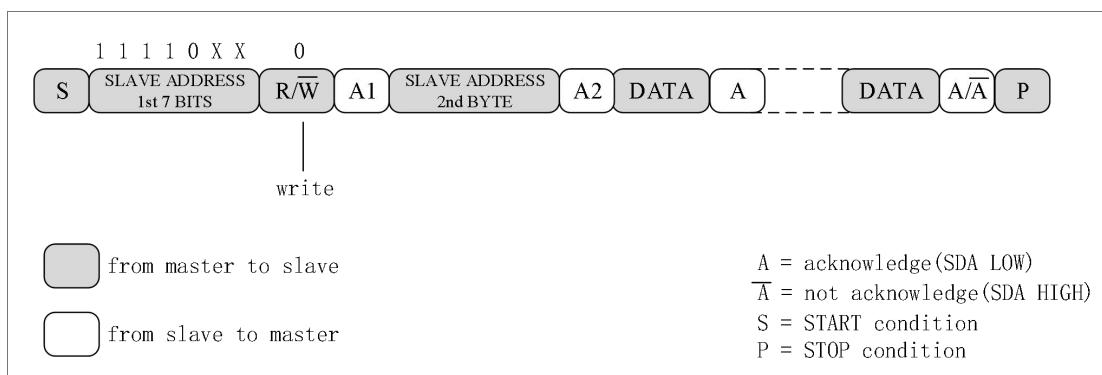
7bitaddress data transmission

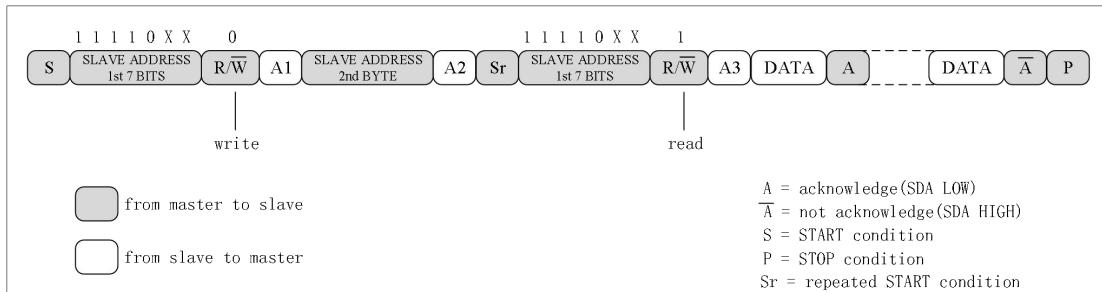


picture5-141 IICFor host sending 7bitAddress Addressed Slave Receiver

picture5-142 I²C The master reads the slave immediately after the first bytepicture5-143 I²C combined format

10bitaddress data transmission

picture5-144 I²C For host sending 10bitAddress Addressed Slave Receiver



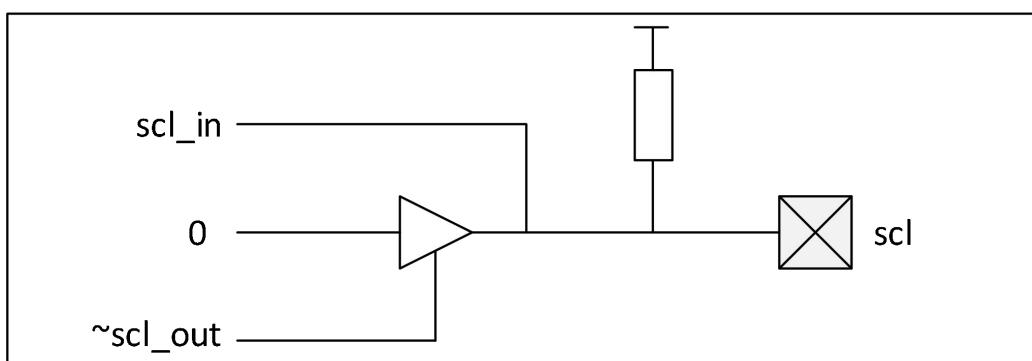
picture5-145 IIC host receiving 10bitAddress Addressed Slave Transmit

SCLandSDA

exist IIC protocol, the port signal SCL and SDA is a bidirectional signal. for Standard-mode, Fast-mode,

SCL and SDA is bidirectional open drain IO, the signal is pulled up to a high level by a resistor, and the rising edge is slow.

SCL and SDA in use IO for ordinary two-way IO, through the processing of the enable signal to achieve bidirectional open-drain IO of function. In the following way, achieve SCL and SDA bi-directional open drain IO:



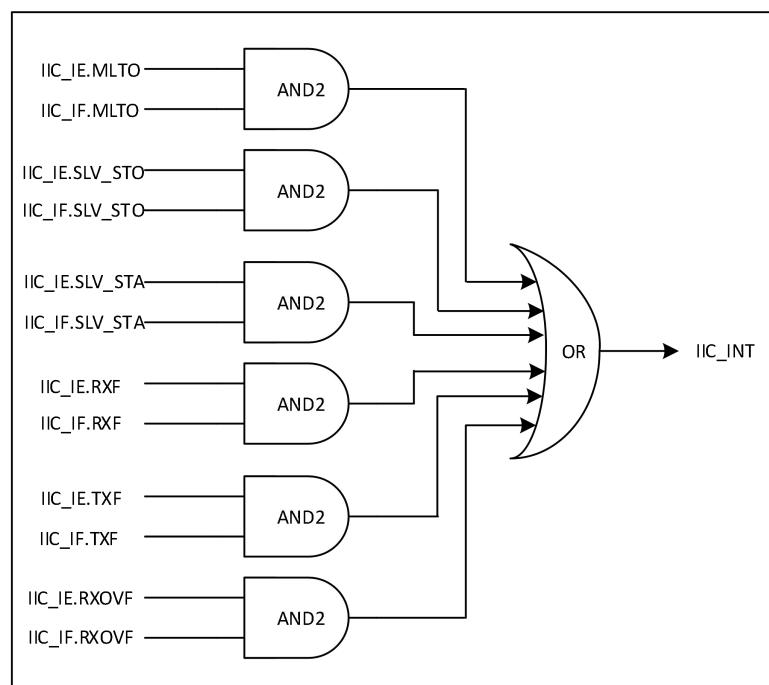
picture5-146 IIC SCL Open Drain Diagram

interrupt function

This module provides six interrupt functions, including receiving data overflow, sending completion, receiving completion, SLAVE detection arriveSTART, SLAVE detected STOP and MASTER SCL LOW Timeout interrupt. accessible IIC_IER register allocation

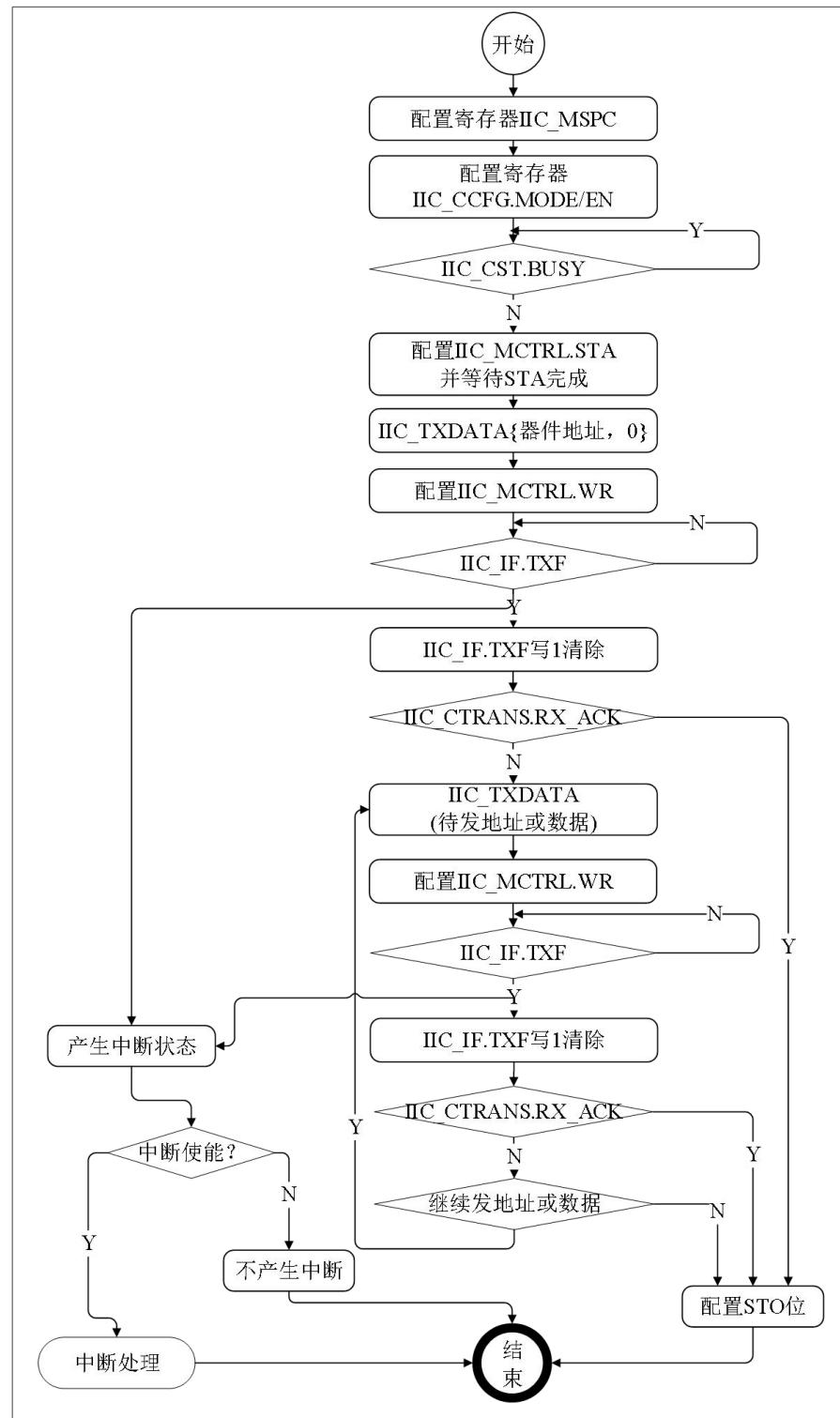
Set the corresponding interrupt enable bit, by IIC_IF register to view each corresponding interrupt status bit.

The interrupt flag and interrupt diagram are as follows:



picture5-147 IIC Interrupt flag and interrupt diagram

Operating procedures

master-transmitter

picture5-148 IICHost sending operation flow chart

1, set the registerIIC_MSPC. supposepclk=60M,hopeIICwork atStandard-mode(100kbps) speed, eachSCL 600indivualpclk,SotLOWfor400indivualpclk,tHIGHfor200indivualpclk, so that to setSCL_LOW=0xC4,SCL_HITH=0x60,CPD=0x01.

2, set the registerIIC_CCFGofMODEBit is1,registerIIC_CCFGofENBit is1.

3, query registerIIC_CSTofBUSYbit, if1, wait until it becomes0; if0, but

Proceed to the next step.

4, hairstart. set registerIIC_MCTRLofSTABit is1, poll the bit until it becomes0.

5, hairslaveaddress byte, the specific steps are as follows:

set registerIIC_TXDATA,inbit7~bit1is the slave address,bit0for0instant write command;

set registerIIC_MCTRLof WRBit is1, poll the bit until it becomes0(or query to registerIIC_IF ofTXFBit is1(Sent successfully) , and write1clear);

read registerIIC_CTRANSofRX_ACKbit, if the bit is0, expressslaveThe address matches successfully

Proceed to the next step, if the bit is1go to step8.

6, TowardsslaveTo write address or data, the specific steps are as follows:

set registerIIC_TXDATA, ready to be writtenslaveaddress or data;

set registerIIC_MCTRLof WRBit is1, poll the bit until it becomes0(or query to registerIIC_IF ofTXFBit is1, and write1clear);

read registerIIC_CTRANSofRX_ACKbit, if the bit is0, indicating that the write address or data is successful and can be entered

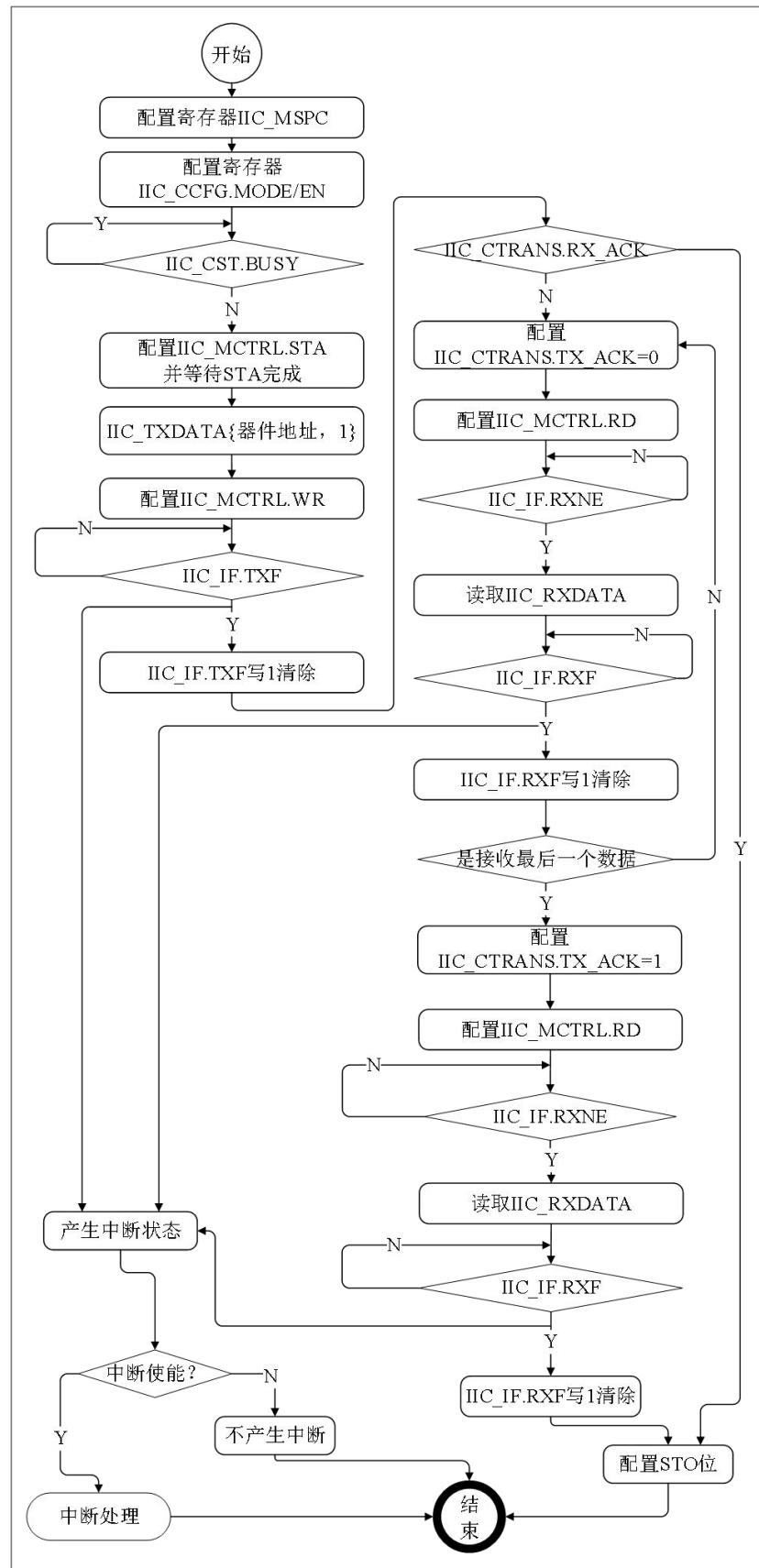
Go to the next step, if the bit is1go to step8.

7, If you want to continue sending address or data, repeat the steps6, otherwise proceed to the next step.

8, hairSTOP. set registerIIC_MCTRLofSTOBit is1, poll the bit until it becomes0.

9, If the interrupt enable is set, it can beTFWhen the transmission is completed, the interrupt status is found, and the interrupt processing can be performed.

master-receiver



picture5-149 IICHost receiving operation flow chart

1, set the registerIIC_MSPC. supposeclk=60M,hopeIICwork atStandard-mode(100kbps) speed, eachSCL 600individualpclk,SotLOWfor400individualpclk,tHIGHfor200individualpclk, so that to setSCL_LOW=0xC4,SCL_HITH=0x60,CPD=0x01.

2, set the registerIIC_CCFGofMODEBit is1,registerIIC_CCFGofENBit is1.

3, query registerIIC_CSTofBUSYbit, if1, wait until it becomes0; if0, but

Proceed to the next step.

4, hairstart. set registerIIC_MCTRLofSTABit is1, poll the bit until it becomes0.

5, hairslaveaddress byte, the specific steps are as follows:

set registerIIC_TXDATA,inbit7~bit1is the slave address,bit0for1Instant read command;

set registerIIC_MCTRLofWRBit is1, poll the bit until it becomes0(or query to registerIIC_IF ofTXFBit is1(Sent successfully) , and write1clear);

read registerIIC_CTRANSofRX_ACKbit, if the bit is0, express slaveThe address matches successfully

Proceed to the next step, if the bit is1go to step8.

6, from slaveTo read data, the specific steps are as follows:

set registerIIC_CTRANSofTX_ACKBit is0;

set registerIIC_MCTRLofRDBit is1, query until the registerIIC_IFofRXNEBit is1;

read registerIIC_RXDATA, get slavedata;

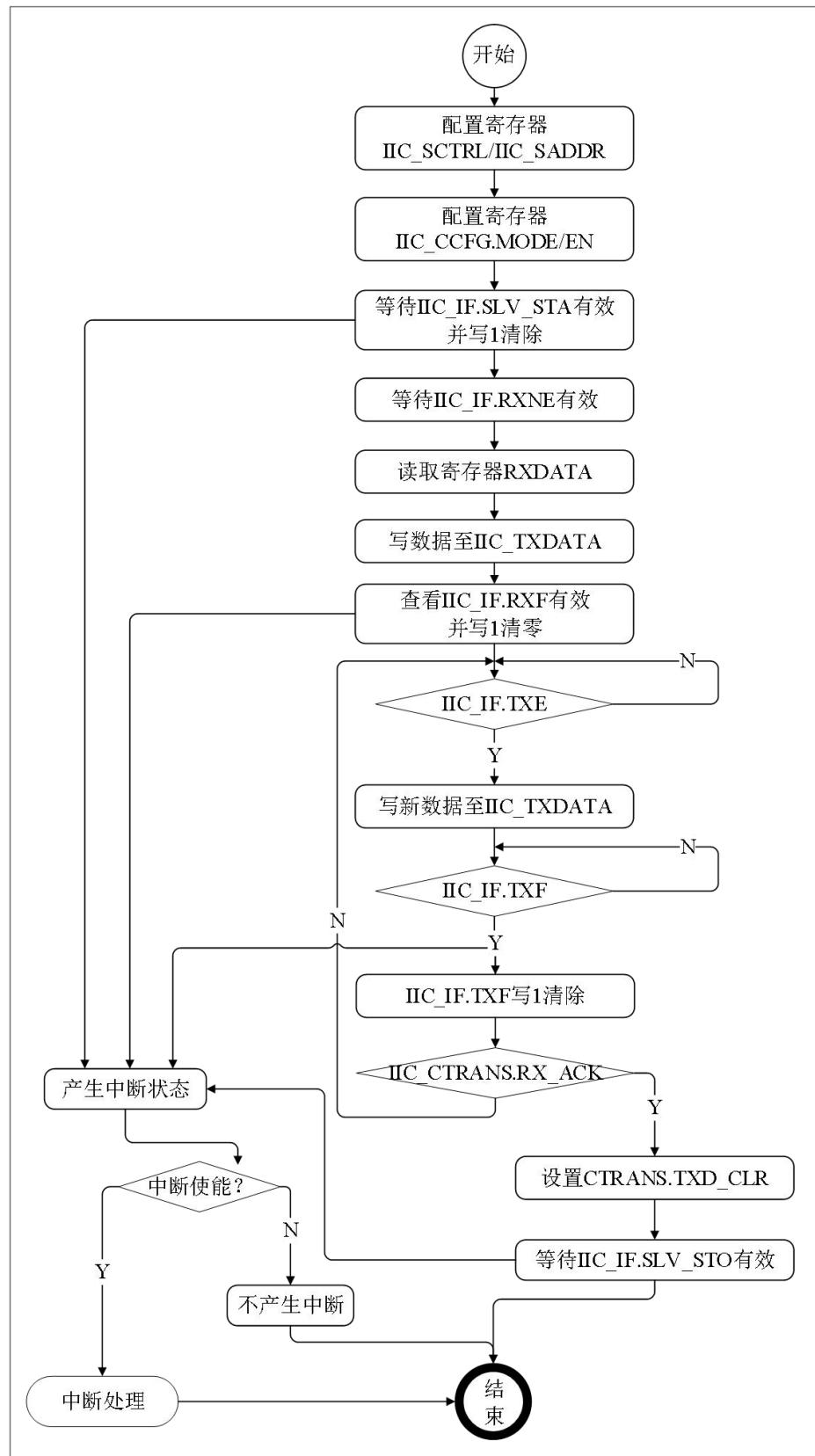
query registerIIC_MCTRLofRDbit until it becomes0(or query to registerIIC_IF.RXFBit is1, and write1Clear).

7, Continue to receive and not the last data received, repeat the steps6; If it is the last data received, set registerIIC_CTRANSofTX_ACKBit is1, other same steps6, then proceed to the next step.

8, hairSTOP. set registerIIC_MCTRLofSTOBit is1, poll the bit until it becomes0.

9, If the interrupt enable is set, it can be TXFwhen sending is complete and RXFWhen the receiving is completed, the interrupt status is detected, and the Line interrupt handling.

slave-transmitter



picture5-150 IICSlave sending operation flow chart

1, set up slave address mode. registerIIC_SCTRL of ADMDBit is 0.

2, set up slave address registerIIC_SADDR.

3, set the registerIIC_CCFG of MODEBit is 0, registerIIC_CCFG of ENBit is 1.

4, query until the registerIIC_IF of SLV_STAbit, indicating that the detected IIC on the bus start issued, and write 1 cleared.

5, query until the registerIIC_IF of RXNEBit is 1. said yes the master selected this device.

6, read registerIIC_RXDATA, if the registerIIC_SADDR address is set in mask, judge the master

The actual address to send from.

7, prepare data, write registerIIC_TXDATA.

8, query until the registerIIC_IF of RXFBit is 1, indicating that after the previous address matches, return ACK finish.

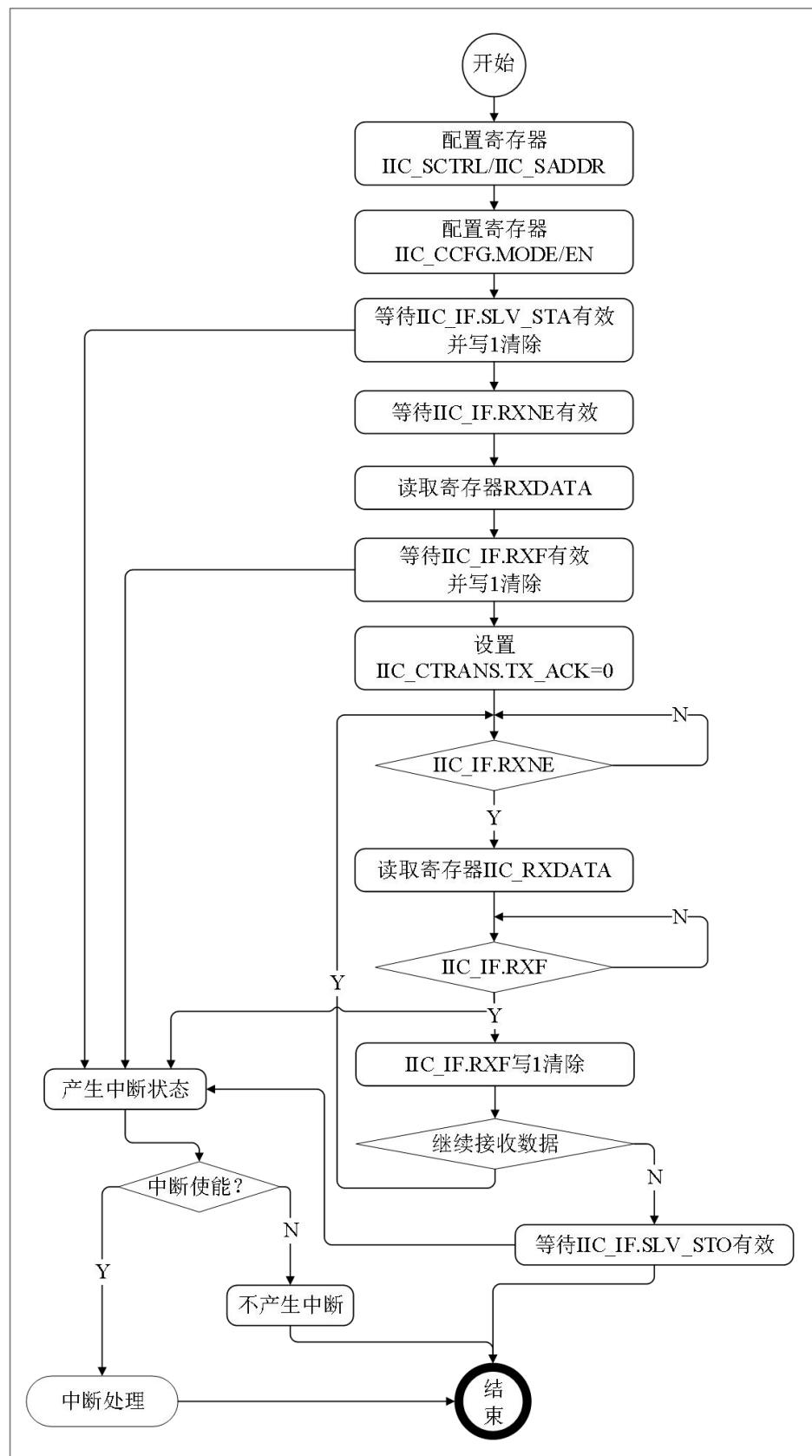
9, query until the registerIIC_IF of TXEBit is 1, you can send to TXDATA New data is written in.

10, query until the registerIIC_IF of TXFBit is 1, indicating that the data transmission is complete. then write 1 clear.

11, query registerIIC_CTRANS of RX_ACK bit, if 0, express the master hope to continue receiving
According to, repeat 9~11; if the registerIIC_CTRANS of RX_ACK Bit is 1, express the master wish to end the read operation,
then set the registerIIC_CTRANS of TXD_CLR bit, pre-arm to register before clearing IIC_TXDATA middle
last data. Go to the next step.

12, Query to the registerIIC_IF of SLV_STO bit, indicating that the detected IIC on the bus STOP issue. this time
Session ended.

13, If the interrupt enable is set, it can be SLAVE detected STA, RXF received complete, TXF send complete and
SLAVE detected STO when the interrupt status is detected, the interrupt processing can be performed.

slave-receiver

picture5-151 IICSlave Receive Operation Flowchart

1, set up slave address mode. registerIIC_SCTRLofADMDBit is0.

2, set up slave address registerIIC_SADDR.

3, set the registerIIC_CCFGofMODEBit is0, registerIIC_CCFGofENBit is1.

4, query until the registerIIC_IFofSLV_STAbit, indicating that the detectedIICOn the busstartissued, and write 1cleared.

5, wait until the registerIIC_IFofRXNEbit is1. said yes the masterSelect this device.

6, read registerIIC_RXDATA, if the registerIIC_SADDRaddress is set inmask, judge the master

The actual address to send from.

7, query until the registerIIC_IFofRXFBit is1, indicating that after the previous address matches, returnACKFinish. However write after 1clear.

8, set the registerIIC_CTRANSofTX_ACKBit is0.

9, query until the registerIIC_IFofRXNEbit is1, express slaveReceive new data, read the register IIC_RXDATA.

10, query until the registerIIC_IFofRXFBit is1, indicating that after receiving data before, returnACKFinish. However write after 1clear.

11, repeat 9~10, continue to receive data until the register is queriedIIC_IFofSLV_STObit, indicating that this time Session ended.

12, If the interrupt enable is set, it can be SLAVEdetectedSTA, RXFreceive complete and SLAVEdetectedSTO

When the interrupt status is detected, the interrupt processing can be performed.



clock-stretching

bymaster-receiverFor example, the details are as follows:

1, set the registerIIC_MSPC. supposepclk=60M,hopeIICwork atStandard-mode(100kbps) speed, eachSCL 600individualpclk,SotLOWfor400individualpclk,tHIGHfor200individualpclk, so that to setSCL_LOW=0xC4,SCL_HITH=0x60,CPD=0x01.

2, set the registerIIC_CCFGofMODEBit is1,registerIIC_CCFGofENBit is1.

3, query registerIIC_CSTofBUSYbit, if1, wait until it becomes0; if0, but

Proceed to the next step.

4, hairstart. set registerIIC_MCTRLofSTABit is1, poll the bit until it becomes0.

5, hairslaveaddress byte, the specific steps are as follows:

set registerIIC_TXDATA,inbit7~bit1is the slave address,bit0for1instant read command;

set registerIIC_MCTRLofWRBit is1, poll the bit until it becomes0(or query to registerIIC_IF ofTXFBit is1(Sent successfully), and write1clear);

read registerIIC_CTRANSofRX_ACKbit, if the bit is0, express slaveThe address matches successfully

Proceed to the next step, if the bit is1go to step8.

6, from slaveRead data: if slaveIt takes a long time to prepare the data, the master of SCL Will be slave hold straight to slave data ready if hold too long, will cause the master to generate registerIIC_IF of MLT0bit interrupt, the specific steps are as follows:

set registerIIC_CTRANSofTX_ACKBit is0;

set registerIIC_MCTRLofRDBit is1, query until the registerIIC_IF of RXNEBit is1;

read registerIIC_RXDATA, get slave data;

query registerIIC_MCTRLofRDbit until it becomes0(or query to registerIIC_IF.RXFBit is1, and write1Clear).

7, Continue to receive and not the last data received, repeat the steps6; If it is the last data received, set registerIIC_CTRANSofTX_ACKBit is1, other same steps6, then proceed to the next step.

8,hairSTOP. set registerIIC_MCTRLofSTOBit is1, poll the bit until it becomes0.

9, If the interrupt enable is set, it can beTXFwhen sending is complete andRXFwhen the receiving is completed, the interrupt status is detected, and the Line interrupt handling.

byslave-transmitterFor example, the details are as follows:

1, set upslaveaddress mode. registerIIC_SCTRLofADMDBit is0,STRETCHBit is1.

2, set upslaveaddress registerIIC_SADDR.

3, set the registerIIC_CCFGofMODEBit is0,registerIIC_CCFGofENBit is1.

4, query until the registerIIC_IFofSLV_STAbit, indicating that the detectedIICon the busstartissued, and write 1cleared.

5, query until the registerIIC_IFofRXNEBit is1. said yes the masterSelect this device.

6, read registerIIC_RXDATA, if the registerIIC_SADDRaddress is set inmask,judgethe master

The actual address to send from.

7, prepare data, write registerIIC_TXDATA, if the data preparation time is very long, then slaveThe hardware will automatically pull downSCL, Prevent the master issue a newSCL. when slave complete write registerIIC_TXDATAAfter action, slave The hardware will then wait for the registerIIC_MSPCset upSCL_LOWafter time, releaseSCL, proceed to the next step.

8, query until the registerIIC_IFofRXFBit is1, indicating that after the previous address matches, returnACKFinish.

9, query until the registerIIC_IFofTXEBit is1, you can send toTXDATANew data is written in.

10, query until the registerIIC_IFofTXFBit is1, indicating that the data transmission is complete. then write1clear.

11, query registerIIC_CTRANSofRX_ACKbit, if0, express the masterHope to continue receiving According to, repeat9~11; if the registerIIC_CTRANSofRX_ACKBit is1, express the master wish to end the read operation, then set the registerIIC_CTRANSofTXD_CLRbit, pre-arm to register before clearingIIC_TXDATAmiddle last data. Go to the next step.

12, Query to the register IIC_IFofSLV_STObit, indicating that the detected IIC on the bus STOP issue. this time

Session ended.

13, If the interrupt enable is set, it can be SLAVE detected STA, RX Received complete, TXF send complete and

SLAVE detected STO When the interrupt status is detected, the interrupt processing can be performed.

register map

name	Offset	bit width	type	reset value	describe
IIC0					BASE: 0x400B9000
IIC1					BASE: 0x400B9800
IIC_CCFG	0x0	32	R/W	0x18	General Configuration Register
IIC_CST	0x4	32	RO	0x06	General Status Register
IIC_CTRANS	0x8	32	R/W	0x02	general transfer register
IIC_RXDATA	0xC	32	RO	0x00	Receive Data Register
IIC_TXDATA	0x10	32	R/W	0x00	send data register
IIC_IE	0x14	32	R/W	0x00	interrupt enable register
IIC_IF	0x18	32	R/W	0x01	Interrupt Flag Register
IIC_MCTRL	0x20	32	R/W	0x00	Host Mode Control Register
IIC_MSPC	0x24	32	R/W	0x33f7f	Timing Configuration Register
IIC_SCTRL	0x30	32	R/W	0x08	Slave Mode Control Register
IIC_SADDR	0x34	32	R/W	0x00	Slave Mode Address Register

Register description

IIC_CCFG Register(0x00)

bit field	name	type	reset value	describe
31:7	RESERVED	RO	0	reserved bit

6:3	DNF	R/W	0x3	Receive SDA,SCLDigital Noise Filtering (Digital Noise Filter) 0000: filter disabled 0001: Filtering is enabled, and the filtering capability is maximum1system clock ... 1111: Filtering is enabled, and the filtering capability is maximum15system clock
2	RESERVED	RO	0	reserved bit
1	MODE	R/W	0x0	mode control 0:slavemode 1:the mastermodel
0	EN	R/W	0	IICbus enable 0: disable 1:Enable

IIC_CSTregister(0x04)

bit field	name	type	reset value	describe
31:14	RESERVED	R	0	reserved bit
13:12	SLV_RXDT	RO	0	SlaveThe type of data received. only atSlaveThe pattern is valid. 00:RXDATAis empty. 01: Received is an address. 10: Received is data. 11: Received ismaster code. only ifMCDE=1valid.
11	SLV_STRETCH_BUSY	RO	0	Slave clock stretchingbusy status. only atslaveThe pattern is valid. 0:noneclock stretching. 1:haveclock stretching.
10	SLV_WR	RO	0	Slavewrite status. only atslaveThe pattern is valid. 1:slave receivedthe masterValid after the write request. 0:slavereceivedthe masterA read request orSTOPAfter that, it will be cleared automatically.

9	SLV_RD	RO	0	Slave read status. only at slave. The pattern is valid. 1: slave received the master valid after a read request. 0: slave received the master write request or STOP. After that, it will be cleared automatically.
8	SLV_ACTIVE	RO	0	Slave active state. only at slave. The pattern is valid. 0: slave device is inactive 1: slave device is active. This bit is valid after the address matching is successful; STOP, or Start_repeat. After the address matching is unsuccessful, it will be cleared automatically.
7:3	RESERVED	RO	0	reserved bit.
2	SDA	RO	1	IIC SDA state. Not subject to IIC bus enable effect 0: IIC SDA for low 1: IIC SDA for high
1	SCL	RO	1	IIC SCL state. Not subject to IIC bus enable effect 0: IIC SCL for low 1: IIC SCL for high
0	BUSY	RO	0	The bus is busy. Standard is not subject to IIC_CCFG.EN bit control, when EN When not enabled, still detect bus busy status 0: The bus is not busy 1: bus busy, IIC bus START to STOP valid during

IIC_CTRANSregister(0x08)

bit field	name	type	reset value	describe
31:3	RESERVED	R	0	reserved bit
2	TXD_CLR	W, AC	0	The transmit data register is cleared. Hardware is automatically cleared. 0: Do not clear. 1: clear IF.TXE bit.
1	RX_ACK	RO	1	When acting as the sender, the received ACK/NACK hardware set, IF.TXF. This bit can be queried after it is valid; received Start_repeater STOP will place this location 1. 0: received ACK 1: received NACK

0	TX_ACK	R/W	0	<p>When acting as a receiver, sendACK/NACK. 0:sendACK. 1:sendNACK.</p> <p>Note: In the following cases, ACK/NACK not up to standard:</p> <ul style="list-style-type: none"> A. slave When receiving an address, the hardware automatically sendsACK/NACK. B. slave MCDE valid, received master code, the hardware automatically returns toNACK. C. slave When receiving overflow, the hardware automatically sendsNACK.
---	--------	-----	---	--

IIC_RXDATAregister(0x0C)

bit field	name	type	reset value	describe
31:8	RESERVED	R	0	reserved bit
7:0	RXDATA	R	0	<p>Receive data register. IF.RXNE for 1, indicating that there is valid data in this register.</p> <p>Note: After completing data reception (does not contain ACK/NACK sent), update this register.</p> <p>slave Receive address byte situation, see RXF bit description.</p>

IIC_TXDATAregister(0x10)

bit field	name	type	reset value	describe
31:8	RESERVED	R	0	reserved bit
7:0	TXDATA	R/W	0	Transmit data register. IF.TXE for 0, indicating that there is data to be sent in this register.

IIC_IERegister(0x14)

bit field	name	type	reset value	describe
31:18	RESERVED	R	0	reserved bit
17	MLTO	R/W	0	Master SCL LOWTimeout interrupt enable. 0: Disable. 1:Enable.
16:10	RESERVED	RO	0	reserved bit.
9	SLV_STO	R/W	0	SlavedetectedSTOPInterrupt enable. 0: Disable. 1:Enable.
8	SLV_STA	R/W	0	SlavedetectedSTARTInterrupt enable. 0: Disable. 1:Enable.
7:5	RESERVED	RO	0	reserved bit.
4	RXF	R/W	0	Receive data end interrupt enable. 0: Disable. 1:Enable.
3	TXF	R/W	0	Transmit data end interrupt enable. 0: Disable. 1:Enable.
2	wxya	R/W	0	Receive data register overflow interrupt enable. 0: Disable. 1:Enable.
1:0	RESERVED	RO	0	reserved bit.

IIC_IFRegister(0x18)

bit field	name	type	reset value	describe
31:18	RESERVED	R	0	reserved bit

17	MLTO	R, W1C	0	Master SCL LOWtime out. Write1clear. only atthe masterThe pattern is valid. 0: No timeout. 1:time out.SCL LOWtime exceeded1024byMSPC register set bySCL LOWtime.
16:10	RESERVED	RO	0	reserved bit
9	SLV_STO	R, W1C	0	SlavedetectedSTOP. Write1clear. only atslavemode is valid. 0:slavenot detectedSTOP. 1:slavedetectedSTOP.
8	SLV_STA	R, W1C	0	SlavedetectedSTART. Write1clear. only atslavemode is valid. 0:slavenot detectedSTART. 1:slavedetectedSTART.
7:5	RESERVED	RO	0	reserved bit
4	RXF	R, W1C	0	Reception is complete. Write1clear, containACK/NACKtime. 0: Reception is not completed. 1: Reception ends. Note:slaveReception statement: 1,Slavedevice7In bit address mode, theslaveAddress bytes (including R/Wbit) receive complete, this interrupt is generated if the address matches. 2,Slavedevice10In bit address mode, theslaveaddress of the2Character Festival(ADDR[7:0]) reception is complete, if10bit address match, this interrupt is generated; followed byrepeat STARTAfterslaveaddress no.1 bytes, if address8,9bit match, this interrupt is generated; followed bySTART after the first1After byte reception is complete, even if ADDR[9:8]match, this interrupt will not be generated. 3. Slavemode1,SCTRL.MCDE=1,receivedmaster code , this interrupt is generated.
3	TXF	R, W1C	0	Sending is complete. Write1clear, containACK/NACKtime. 0: Sending is not completed, or not sent. 1: Sending ends.

2	wxya	R, W1C	0	<p>Receive data register overflow. software write1clear. (updated time point, does not containACK/NACKsend)</p> <p>0: No overflow.</p> <p>1:whenRXDATAWhen it is not empty, and a new byte is received, an overflow will occur. When overflow occurs, new data is lost.</p> <p>Note: For slavemode ifSTRETCHbit is valid, when the receive data register is not empty and a new byte is received, slave device will pull lowSCLsignal untilRXDATAThe old data in is read, and the new data is stored inRXDATA, this case does not produce overflow.</p>
1	RXNE	RO	0	<p>Receive data register is not empty.</p> <p>0: The receive data register is empty, and there is no unread receive data. 1: The receive data register is not empty, and there is unread receive data.</p> <p>Note: This bit is updated when the data is received (excludingACK/NACK sending time).</p> <p>If the old data is not read in time when the new data is received, it can be handled in the following cases:</p> <p>Mastermodel:</p> <p>New data is lost. Simultaneously setIF.RXOVFbit.</p> <p>Slavemode:</p> <p>A. STRETCH=0: New data is lost. Simultaneously setIF.RXOVF bit, the hardware automatically sends theNACK.</p> <p>B. STRETCH=1: normal returnACK, then in the master before sending the next byte, slave will SCL hold At low level, update new data toRXDATAregister. finally releasedSCL.</p>
0	TXE	RO	1	<p>The transmit data register is empty.</p> <p>0: The send data register is not empty, writing is not allowedTXDATA register. 1: send data register empty, allow writingTXDATAregister.</p> <p>Note: At the beginning of sending data, after sending data is read by hardware, this bit is updated to1(at this timeIF.TXFstill for 0).</p> <p>TowardsTXDATAThis bit is cleared when new data is written to the register.</p>

IIC_MCTRLregister(0x20)

bit field	name	type	Reset value	description

31:4	RESERVED	RO	0	reserved bit
3	STO	W, AC	0	Write1, produceSTOP, automatically cleared to zero after completion.
2	WR	W, AC	0	Write1, sendTXDATAData in, after completion (includingACK/NACKtime) is automatically cleared. write to standard1before requestTXDATACan not be empty. Otherwise, this bit cannot be set. Notice: WRandRDbits cannot be written simultaneously1.
1	RD	W, AC	0	Write1, receive data toRXDATADuring, after completion (includingACK/NACK time) is automatically cleared.
0	STA	W, AC	0	Write1, produceSTART, automatically cleared to zero after completion. Note: allow STAandWRSet at the same time, send prioritySTART.

IIC_MSPCregister(0x24)

bit field	name	type	reset value	describe
31:28	RESERVED	RO	0	reserved bit
27:24	DAT_HD	R/W	0x00	SDAData retention time configuration. (rightMasterand Slavevalid) for the master:tHD;DAT=(DAT_HD + 4) * Tpclk for slave:tHD;DAT=(DAT_HD + DNF + 6) * Tpclk Note: If the application environment is relatively harsh, it should be noted that theSDA A glitch during data hold may causeSDAChanges along the timing of the advance glitch width (if this timeSCLIf there is no glitch on the bus, there will be unexpectedSTA,STOP). In this case, set the DAT_HD make tHD;DAT greater than the maximum glitch width.
23:16	CPD	R/W	0x03	Clock prescaler, seeSCL_HIandSCL_LOWdescribe. (only for Mastermode valid)
15:8	SCL_HI	R/W	0x3f	SCLClock high time configuration. (only forMastermode valid) tHIGH=((SCL_HI+1) * (CPD+1) + DNF + 6) * Tpclk



7:0	SCL_LOW	R/W	0x7f	<p>SCLClock low time configuration. (rightMastermode is valid; in slave mode, if enabledSTRETCHfunction, andSCTRL.ASDSconfigured as0, you need to configure this register. existsslaveWrite TXDATAAfter that, delay the time set by this register, and then releaseSCL.)</p> <p>$t_{LOW} = (SCL_LOW + 1) * (CPD + 1) + DAT_HD + 5) * T_{pclk}$</p> <p>SCLThe cycle istHIGH+tLOW.</p> <p>recommendtHIGHandtLOWThe ratio of1:2.</p>
-----	---------	-----	------	--

IIC_SCTRLregister(0x30)

bit field	name	type	reset value	describe
31:4	RESERVED	RO	0	reserved bit
3	ASDS	R/W	1	<p>StretchingPost-data setup time adaptation is enabled. (Adaptive Stretching Data Setup)</p> <p>0: Self-adaptation is disabled. Depend onMSPCset up.</p> <p>1: Adaptive enable. receivingthe masteraddress is automatically detected when theSCL low time, astretchingAfter data establishment time.</p> <p>Note:Slave-transmitter,whenSTRETCHregister is set to active and occurs stretchingIn the case of , after the new data is ready,slave will continue to pull downSCLfor a period of time to ensureSDASatisfy the requirement of data creation time online.</p>

				Clock stretchingenable control. 0:Clock stretchingnot enabled. 1:Clock stretchingEnable.
2	STRETCH	R/W	0	<p>Note: slaveasreceiver, when new data is received, but old data is not read in time (IF.RXNE=1): CTRANS. STRETCH_BUSYbecomes valid, returningACKafter that willSCL hold In the low level, until the old data is read, update the new data to the RXDATAin, at the same timeSTRETCH_BUSYbecome invalid, then releaseSCL, start receiving the next data.</p> <p>slaveastransmitter, when sending ends (IF.TXF=1, including receivingACK/NACKtime), but new data is not ready (TXE=1) : CTRANS. STRETCH_BUSYbecomes effective, theSCL holdLOW until new data is ready, delaySCL_LOWafter time, STRETCH_BUSY become invalid, then releaseSCL, start sending new data.</p>
1	MCDE	R/W	0	<p>Master Code Detect Enable. 0: do not detectmaster code. 1: detectmaster code.</p> <p>When the standard is valid, slaveexistSTARTdetected aftermaster code, will generateIF.RXFinterrupt, and the hardware sets theCST.SLV_RXDT for11. Software shall guaranteeslaveThe address setting does not match themaster codeconflict.</p>
0	ADMD	R/W	0	<p>slaveAddress mode control. 0:7bit address mode 1:10bit address mode</p>

IIC_SADDRregister(0x34)

bit field	name	type	reset value	describe
31:24	RESERVED	RO	0	reserved bit

23:17	MASK_ADDR[7:1]	R/W	0	SlaveCorresponding address bitmask. 0: no mask. 1: The mask corresponds to the bit address. After the mask, the hardware matches the slave address, the masked address bits are ignored. for 10bit address mode, RXDATA save only ADDR[7:0], so it is not supported for ADDR[9:8] of mask.
16	MASK_ADDR 0	R/W	0	SlaveCorresponding address bitmask.
15:10	RESERVED	RO	0	reserved bit.
9:8	ADDR[9:8]	R/W	0	7Bit address mode: don't care 10Bit Address Mode: Address bit 9~bit 8
7:1	ADDR[7:1]	R/W	0	address bit 7~bit 1
0	ADDR0	R/W	0	7Bit address mode: don't care 10 Bit Address Mode: Address bit 0

5.19ADC (ADC)

5.19.1 overview

This chip SAR ADC Using the successive approximation structure, the 14 The analog signal of effective channels is sampled to realize A/D convert.

which channel 0 - aisle 10 Used for external analog input signal sampling, and channel 1 You can also choose to OPA 0 simulation output is sampled, the channel 10 You can also choose to OPA 1 The analog output is sampled.

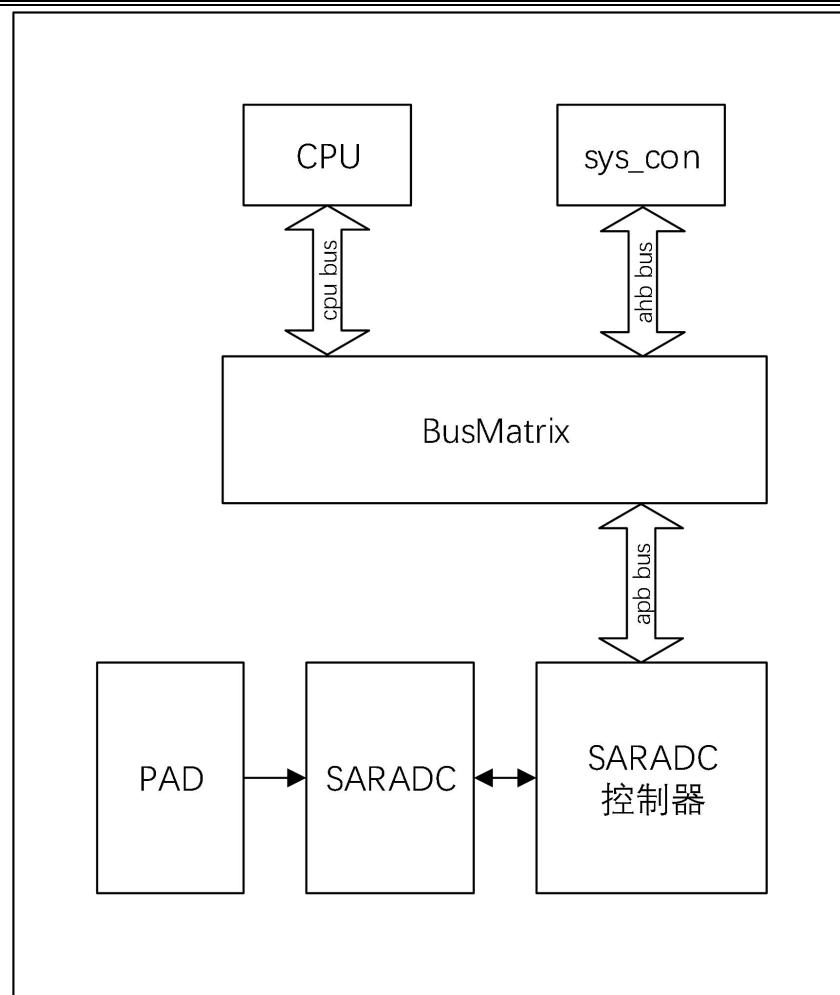
aisle 11 and channel 12 Not used for reservation.

aisle 13 Used to sample the die temperature.

aisle 14 used to pair 1.2V The reference voltage is sampled.

aisle 15 used to pair AVDD/3 The voltage is sampled.

Its system block diagram is shown in the figure below:



picture5-152 SARADCSystem Block Diagram

5.19.2 characteristic

- from the outside I/O port connected to an analog channel input;
- Sampling rates up to 2.4M;
- 0-10The channel is a high-speed channel, 13-15The channel is a low-speed channel; the establishment time of a high-speed channel is short, and the required sampling window is short.

On the contrary, the low-speed channel takes a long time to set up, and the sampling window needs to be increased appropriately to ensure the accuracy of the sampling voltage;

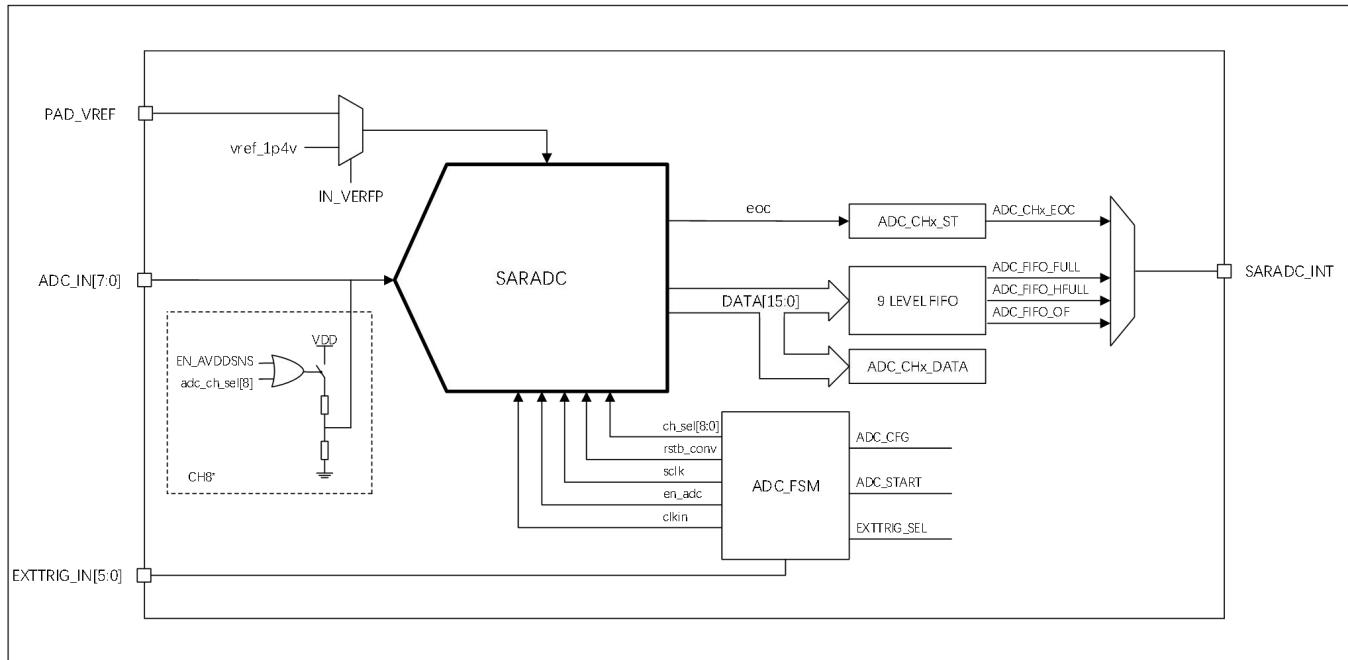
- 13The channel is the voltage input of the on-chip temperature sensor;
- 14Channel is 1.2VReference voltage input;
- 15Channel is AVDD/3voltage input;

-externalADC reference voltage PAD_VREF, some packages PAD_VREF and AVDD_ADC connected together;

-Independent channel data register, shared by all channels 16 class FIFO optional;

- Software trigger and hardware trigger, hardware trigger can be selected TIMERPLUS The overflow signal of the module and PMWPLUS of triggerSignal;
- Support sampling once or sampling multiple times to take the average configurable;
- Support single sampling and continuous sampling configurable;
- After the single sampling is started, traverse all the selected channels once from the low channel to the high channel;
- After the continuous sampling is started, it traverses all the selected channels from the low channel to the high channel without interruption until the software stop sampling;
- Support internal sampling clock, external sampling clock can be configured;
- Support configurable sampling window;
- Conversion completion status of each channel, FIFO full state and FIFO half-full state can generate an interrupt;
- ADC Channel input range: $0 \leq V_{IN} \leq V_{ref} \leq AVDD_ADC$;
- support DMA read FIFO sampling data;

5.19.3 Block Diagram of Module Structure



picture5-153 SARADCModule Structure Diagram

SARADChave11externalADCSampling channel, externalADCThe sampling channel voltage should be lower than the reference voltage

PAD_VREF,ADCcontrol unit according toADCconfiguration to generate corresponding control timing, controlADCTake a sample. sampling junction

The results can be selected to be stored in each channel register or stored inFIFO, and produces a sample complete status orFIFOhalf full and full state.

SARADCsupport12external trigger sources, respectively fromTIMERPLUSmodularTIMER_PLUS1_GOAL[1:0] (TIMERPLUS1module's high counter overflow signal and low counter overflow signal) andTIMER_PLUS0_GOAL[1:0] (TIMERPLUS0module's high counter overflow signal and low counter overflow signal), and fromPWMPLUSmodular PWM_PLUS1_TRIGGER[3:0](PWMPLUS1modulartriggersignal) andPWM_PLUS0_TRIGGER[3:0] (PWMPLUS0modulartriggerSignal).

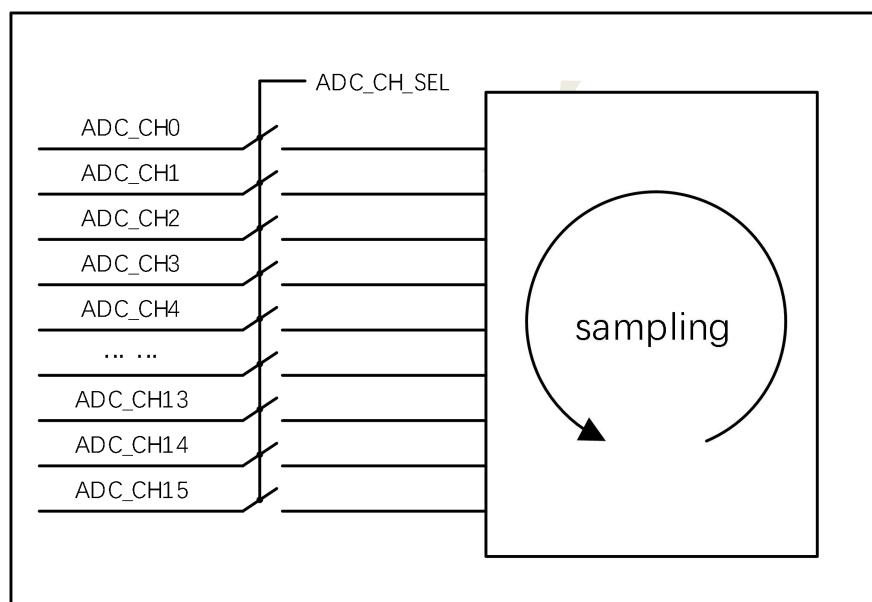
5.19.4 Functional description

channel selection

BookSARADC owned in total 16 sampling channels, where 14 is an effective channel, 2 channel is invalid (channel 11 and channel 12). Do not use via the control register ADC_CFG of ADC_CH_SEL bits to control which channels are in the next takes effect in the next sample.

After the sampling is initiated, the control unit will traverse all the selected effective channels from the low channel to the high channel to complete the sampling.

Sample.



picture5-154 SARADC Channel selection diagram

Sampling once and sampling multiple times to take the average

Through the configuration register SPL_NUM to set the average number of sampling times for each sampling result of each channel of, such as when SPL_NUM set to sample 8 times averaged, then each channel will perform 8 time finished whole sample, and finally SARADC_CTRL will put 8 The sub-sampling results are averaged, and then stored in the channel register or FIFO inside.

The channel sampling completion flag will be generated after averaging.

Single Sampling vs Continuous Sampling

Through the configuration register ADC_M to set SARADC sampling mode. SARADC has single-shot and continuous-sampling

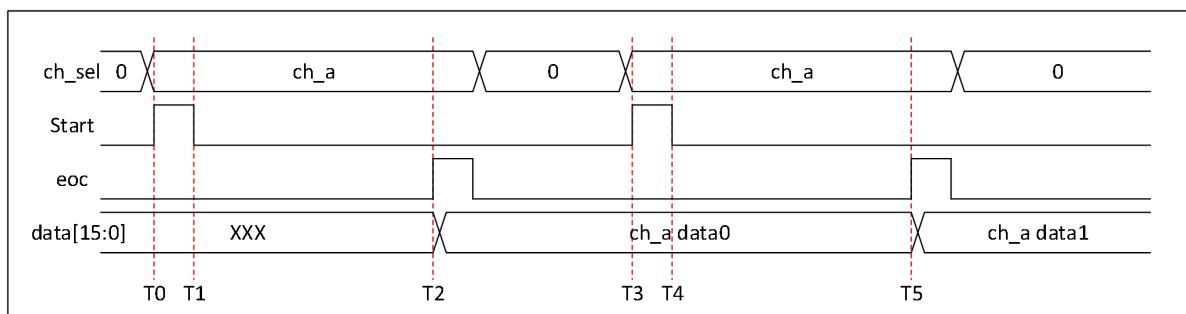
Two modes. In one-shot mode, the SARADC It will traverse the effective channels to sample once, and then stop waiting for the next sampling;

In continuous sampling mode, the SARADC It will cycle through the valid channels for sampling, and continue sampling until the software stops sampling.

Single Channel Single Sampling

Single Channel Single Sampling: The effective channel is only a single channel i.e. ADC_CH_SEL only one is high, SARADC

The sampling mode is single sampling.



picture5-155 SARADC schematic diagram of single channel single sampling

The picture above shows the channel schematic diagram of single sampling timing:

- T0 time CPU programming ADC_START in the register ADC_START Bit is 1, initiated SARADC pair channel a Enter

line sampling;

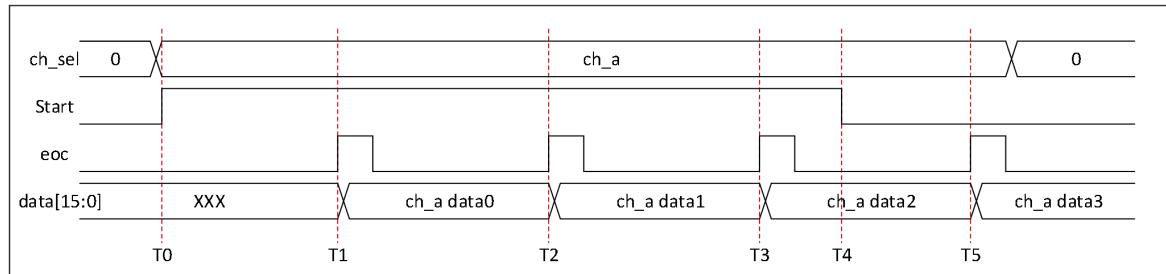
- T1 time ADC_START bit is cleared by digital logic hardware;

- T2timeSARADCSampling is completed once, the output channelasampled datach_a data0,eofflag is pulled high, at this time
CPUcan be read by reading the corresponding channel data register orFIFOThe data register acquires sampled data, as well as by reading
The corresponding status register acquires the sampling completion flag;
 - If you want to resample the channel,CPUneed to againADC_STARTin the registerADC_STARTbit
Write1. as aboveT3moment to initiateSARADCpair channelare-sampling;

-T4timeADC_STARTRegisters are cleared by digital logic hardware;
 - T5timeSARADCSampling is completed once, the output channelasampled datach_a data1,eofFlag pulled high. at this time
CPUcan be read by reading the corresponding channel data register orFIFOThe data register acquires sampled data, as well as by reading
The corresponding status register acquires the sampling completion flag;
- When single-channel single sampling,CPUstart upADC_STARTbit will sample the channel once,ADC_STARTmeeting
Cleared directly by hardware, after samplingSARADCstop sampling, waitCPUInitiate the next sampling.

Single channel continuous sampling

Single-channel continuous sampling: the effective channel is only a single channel that isADC_CH_SELonly one is high,SARADC
The sampling mode is continuous sampling.



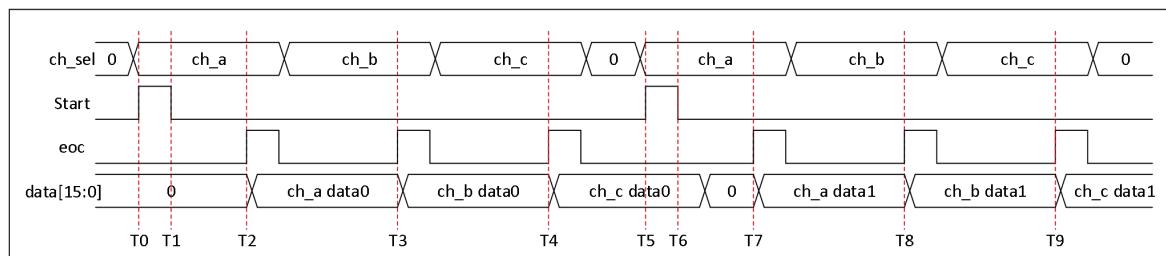
picture5-156 SARADC schematic diagram of single-channel continuous sampling

- The picture above shows the channelSchematic diagram of continuous sampling timing:
- T0timeCPUprogrammingADC_STARTin the registerADC_STARTBit is1, initiatedSARADCpair channelaEnter
line sampling;

- T1timeSARADCSampling is completed once, the output channelasampled datach_a data0,eocFlag pulled high. at this time
CPUcan be read by reading the corresponding channel data register orFIFOThe data register acquires sampled data, as well as by reading
The corresponding status register acquires the sampling completion flag;
 - T2timeSARADCSampling is completed once, the output channelasampled datach_a data1,eocFlag pulled high. at this time
CPUcan be read by reading the corresponding channel data register orFIFOThe data register acquires sampled data, as well as by reading
The corresponding status register acquires the sampling completion flag;
 - T3timeSARADCSampling is completed once, the output channelasampled datach_a data2,eocFlag pulled high. at this time
CPUcan be read by reading the corresponding channel data register orFIFOThe data register acquires sampled data, as well as by reading
The corresponding status register acquires the sampling completion flag;
 - likeCPUexpect to getNSubsampling data, you need to obtain theN-1times data after programming byADC_STARTsend
in memoryADC_STARTBit is0, to stopCPUKeep sampling. see picture aboveT4moment, willADC_STARTclear
for0, the hardware circuit will stop after the current sampling period is completedSARADCContinue sampling;
 - T5timeSARADCSampling is completed once, the output channelasampled datach_a data3,eocFlag pulled high. at this time,
CPUThe last sampling data can be obtained;
- When single-channel continuous sampling,ADC_STARTset asideSARADCThis channel is sampled in a loop until
ADC_STARTCleared by the program, after clearing,SARADCwill stop sampling after the current sampling is complete.

Multi-Channel Single Sampling

Multi-channel single sampling: the effective channel is multiple channelsADC_CH_SELhas a number of high,SARADCsampling mode
for a single sample.



picture5-157 SARADC schematic diagram of multi-channel single-shot sampling

The picture above shows the channela,aisleb,aisleC schematic diagram of continuous sampling timing, where the channel number a<b< c, example channel0<Pass

road5<aisle8:

- T0 time CPU programming ADC_START in the register ADC_START Bit is 1, the hardware initiates SARADC in turn

aisle a, b, c to sample;

- T1 time ADC_START Registers are cleared by digital logic hardware;

- T2 time SARADC sampling channel a Done, output channel a sampled data ch_a data0, eocFlag pulled high. this

hour CPU can be read by reading the corresponding channel data register or FIFO data register to obtain sampled data, and by reading

Get the corresponding status register to get the sampling completion flag;

- T3 time SARADC sampling channel b Done, output channel b sampled data ch_b data0, eocFlag pulled high. this

hour CPU can be read by reading the corresponding channel data register or FIFO data register to obtain sampled data, and by reading

Get the corresponding status register to get the sampling completion flag;

- T4 time SARADC sampling channel c Done, output channel c sampled data ch_c data0, eocFlag pulled high. this

hour CPU can be read by reading the corresponding channel data register or FIFO data register to obtain sampled data, and by reading

Get the corresponding status register to get the sampling completion flag;

- If you want to resample the sequence channel, CPU need to again ADC_START in the register ADC_START

bit write 1. as above T5 moment to initiate SARADC pair channel a, b, c re-sampling;

- T6 time ADC_START Registers are cleared by digital logic hardware;

- T7 time SARADC sampling channel a Done, output channel a sampled data ch_a data1, eocFlag pulled high. this

hour CPU can be read by reading the corresponding channel data register or FIFO data register to obtain sampled data, and by reading

Get the corresponding status register to get the sampling completion flag;

- T8 time SARADC sampling channel b Done, output channel b sampled data ch_b data1, eocFlag pulled high. this

hour CPU can be read by reading the corresponding channel data register or FIFO data register to obtain sampled data, and by reading

Get the corresponding status register to get the sampling completion flag;

- T9 time SARADC sampling channel c Done, output channel c sampled data ch_c data1, eocFlag pulled high. this

hour CPU can be read by reading the corresponding channel data register or FIFO data register to obtain sampled data, and by reading

Get the corresponding status register to get the sampling completion flag;

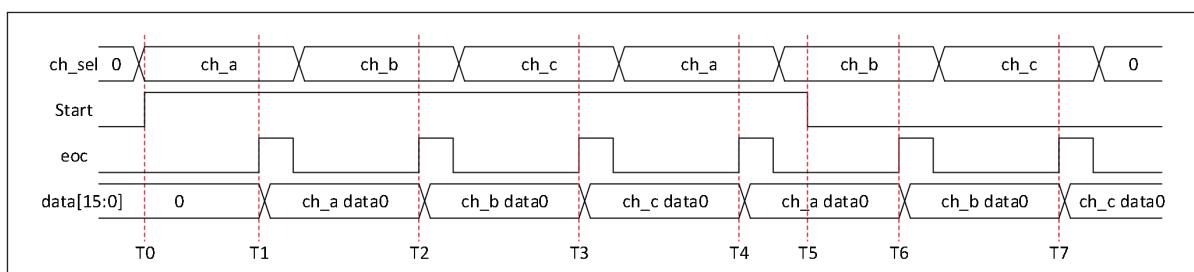
When multi-channel single-shot sampling, startADC_STARTAll selected channels will be sampled once, ADC_START

It will be cleared directly, each channel is sampled once, after all sampling is completed SARADCStop sampling and wait for the next sampling initiation.

Multi-channel continuous sampling

Multi-channel continuous sampling: the effective channel is multiple channels ADC_CH_SEL has a number of high, SARADC sampling mode

for continuous sampling.



picture5-158 SARADC schematic diagram of multi-channel continuous sampling

The picture above shows the channel a, aisle b, aisle c schematic diagram of continuous sampling timing, where the channel number a < b < c, such as channel 0 <

aisle 5 < aisle 8:

- T0 time CPU programming ADC_START in the register ADC_START Bit is 1, initiated SARADC channel in turn a, b, c to sample;
- T1 time SARADC sampling channel a Done, output channel a sampled data ch_a data0, eoc Flag pulled high. this hour CPU can be read by reading the corresponding channel data register or FIFO data register to obtain sampled data, and by reading Get the corresponding status register to get the sampling completion flag;
- T2 time SARADC sampling channel b Done, output channel b sampled data ch_b data0, eoc Flag pulled high. this hour CPU can be read by reading the corresponding channel data register or FIFO data register to obtain sampled data, and by reading Get the corresponding status register to get the sampling completion flag;

- T3timeSARADCsampling channelcDone, output channelcsampled datach_c data0,eocFlag pulled high. this hourCPUcan be read by reading the corresponding channel data register orFIFOdata register to obtain sampled data, and by reading Get the corresponding status register to get the sampling completion flag;
 - likeCPUexpect to getNSubsampling data, you need to obtain theN-1after the data, and at theNpass all times before data conversion is complete, by programming theADC_STARTin the registerADC_STARTBit is0, to stopCPU Keep sampling. see picture aboveT5time as an example, theADC_STARTclear for0, then the hardware circuit is fully will stop when all valid channel conversions are completeSARADCsampling;
 - T4timeSARADCsampling channelaDone, output channelasampled datach_a data1,eocFlag pulled high. this hourCPUcan be read by reading the corresponding channel data register orFIFOdata register to obtain sampled data, and by reading Get the corresponding status register to get the sampling completion flag;
 - T6timeSARADCsampling channelbDone, output channelbsampled datach_b data1,eocFlag pulled high. this hourCPUcan be read by reading the corresponding channel data register orFIFOdata register to obtain sampled data, and by reading Get the corresponding status register to get the sampling completion flag;
 - T7timeSARADCsampling channelcDone, output channelcsampled datach_c data1,eocFlag pulled high. this hourCPUcan be read by reading the corresponding channel data register orFIFOdata register to obtain sampled data, and by reading Get the corresponding status register to get the sampling completion flag;
- When multi-channel continuous sampling,ADC_STARTset asideSARADCIt will cyclically sample all selected channels, untilADC_STARTCleared by the program, after clearing,SARADCwill stop sampling after the current loop sampling is complete.

channel storage withFIFO

SARADCThe sampling result can be stored in the sampling data register of each channel, or the built-in depth is16byte ofFIFOto store the sampling results of all channels. Through the configuration registerADC_MEM_MDto set up sample data storage Location.

When using the channel register, the channel register of each channel will store the sampling value of its own channel, if the data is not read in time Out, the new sampling data will overwrite the old data.

use FIFO when storing. FIFO the sampling value of the selected channel will be stored. If the data is not read out in time, the new sampling number

Data will be lost.

Internal sampling clock method and external sampling clock method

SARADC There are two sampling methods: one is the internal sampling clock method, and the other is the external sampling clock method.
configuration register ADC_IN_SMPL Make a selection.

Different sampling clock methods need to configure corresponding window settings. The window setting of the internal sampling clock mode can be set by
ADC_CFG in the register IN_SMPL_WIN Bit software programming, configurable parameters are 1/3/5/7/9/11/13/15 Eight types
selection; the window setting of the external sampling clock mode can be set by ADC_CFG in the register SMPL_SETUP Bit software programming,
The configurable parameters are 1/2/4/8/16/32/64/128 There are eight options in total.

CPU Triggering and external signal triggering

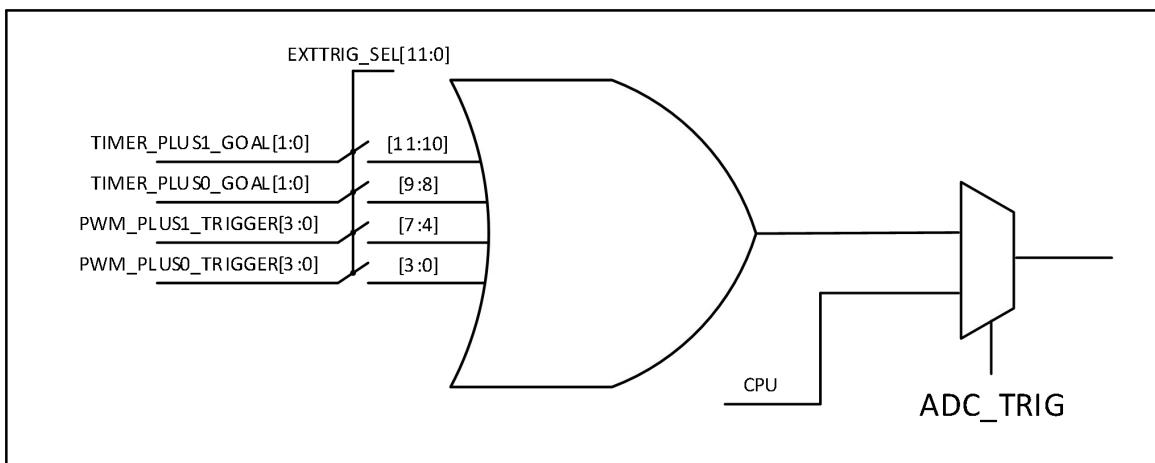
SARADC Provides a selectable external signal trigger source, through the configuration register ADC_TRIG to select the trigger method.
in total 12 an external trigger source signal.

in ADC_EXTTRIG_SEL[3:0] are used to control PWM_PLUS0_TRIGGER[3:0];

ADC_EXTTRIG_SEL[7:4] are used to control PWM_PLUS1_TRIGGER[3:0];

ADC_EXTTRIG_SEL[9:8] are used to control TIMER_PLUS0_GOAL[1:0];

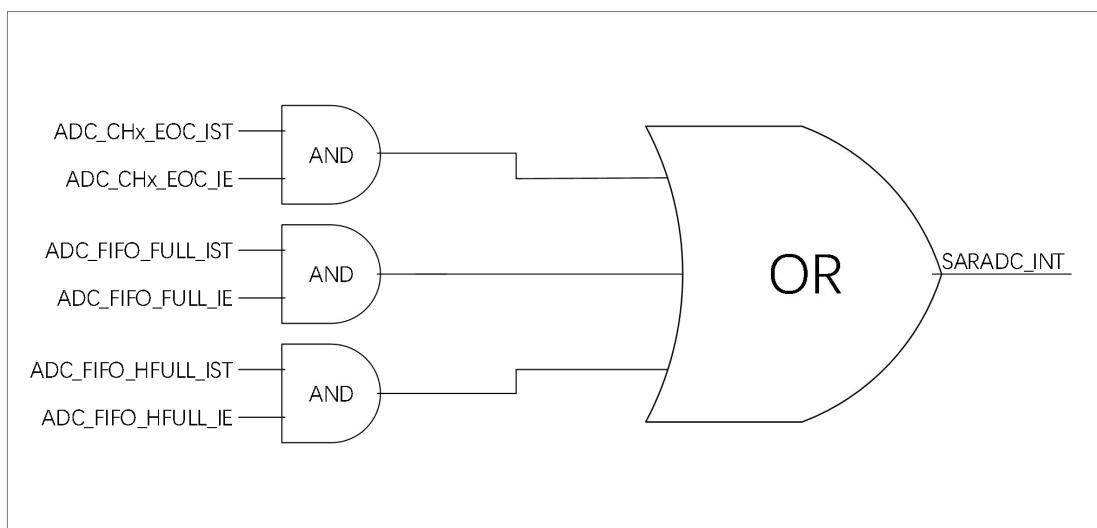
ADC_EXTTRIG_SEL[11:10] are used to control TIMER_PLUS1_GOAL[1:0];



picture5-159 SARADCExternal trigger selection

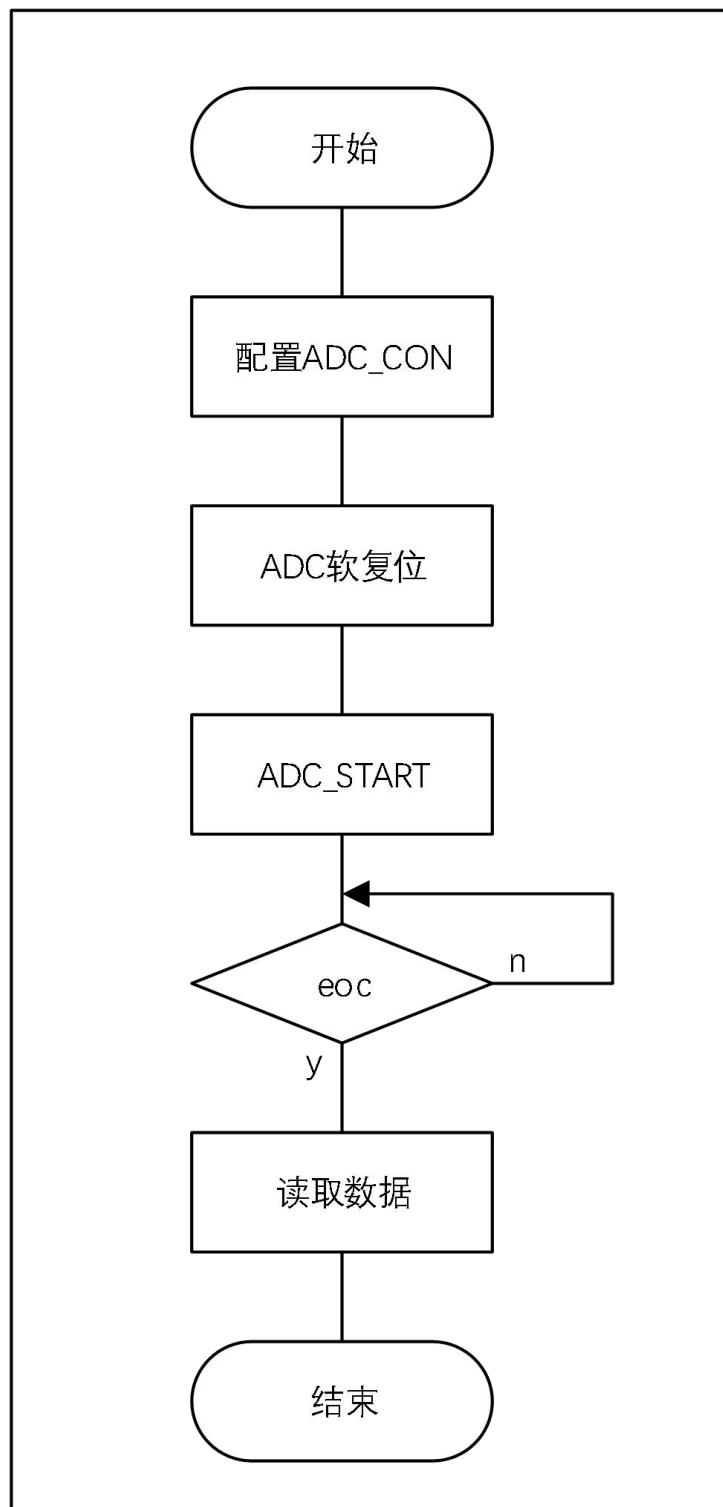
to interrupt

SARADC provided 18 interrupt, as shown in the figure below.



picture5-160 SARADC schematic diagram of interrupt flag and interrupt signal

Operating procedures



picture5-161 SAR ADC Operation flow chart

-OpenSARADCmodule clock

- The configuration pins areADCFunction
- configurationADCThe conversion clock divider value of
- configurationADCconversion channel
- Configure sampling to average
- Configure the conversion mode as single mode or continuous mode
- Configuration data is stored asFIFOor channel
- Configure the sampling clock as internal sampling or external sampling
- Configure internal or external sampling window
- configurationKDIsthe data valid
- configurationoffsetIs the data valid
- Configure whether the channel conversion completion interrupt is enabled
- configurationFIFOwhether the full interrupt is enabled
- configurationFIFOwhether the half-full interrupt is enabled
- initiateADCsoft reset
- startADCsampling
- Wait for sampling to end
- Read sampled data

register map

name	Offset	bit width	type	reset value	describe
SARADC	BASE: 0x400BA000				
ADC_CFG	0x00	32	R/W	0x2100 000	ADCconfiguration register
ADC_START	0x04	32	R/W	0x00	ADCboot register
ADC_IE	0x08	32	R/W	0x00	ADCinterrupt enable register
ADC_IF	0x0c	32	RW	0x00	ADCInterrupt Status Register
ADC_CH0_STAT	0x10	32	R/W	0x00	ADCAisle0status register
ADC_CH0_DATA	0x14	32	R/W	0x00	ADCAisle0data register
ADC_CH1_STAT	0x18	32	R/W	0x00	ADCAisle1status register
ADC_CH1_DATA	0x1C	32	R/W	0x00	ADCAisle1data register
ADC_CH2_STAT	0x20	32	R/W	0x00	ADCAisle2status register
ADC_CH2_DATA	0x24	32	R/W	0x00	ADCAisle2data register
ADC_CH3_STAT	0x28	32	R/W	0x00	ADCAisle3status register
ADC_CH3_DATA	0x2C	32	R/W	0x00	ADCAisle3data register
ADC_CH4_STAT	0x30	32	R/W	0x00	ADCAisle4status register
ADC_CH4_DATA	0x34	32	R/W	0x00	ADCAisle4data register
ADC_CH5_STAT	0x38	32	R/W	0x00	ADCAisle5status register
ADC_CH5_DATA	0x3C	32	R/W	0x00	ADCAisle5data register
ADC_CH6_STAT	0x40	32	R/W	0x00	ADCAisle6status register
ADC_CH6_DATA	0x44	32	R/W	0x00	ADCAisle6data register
ADC_CH7_STAT	0x48	32	R/W	0x00	ADCAisle7status register
ADC_CH7_DATA	0x4C	32	R/W	0x00	ADCAisle7data register
ADC_CH8_STAT	0x50	32	R/W	0x00	ADCAisle8status register
ADC_CH8_DATA	0x54	32	R/W	0x00	ADCAisle8data register
ADC_CH9_ST	0x58	32	R/W	0x00	ADCAisle9status register
ADC_CH9_DATA	0x5C	32	R/W	0x00	ADCAisle9data register
ADC_CH10_ST	0x60	32	R/W	0x00	ADCAisle10status register
ADC_CH10_DATA	0x64	32	R/W	0x00	ADCAisle10data register
ADC_CH11_ST	0x68	32	R/W	0x00	ADCAisle11status register

ADC_CH11_DATA	0x6c	32	R/W	0x00	ADCaisle11data register
ADC_CH12_ST	0x70	32	R/W	0x00	ADCaisle12status register
ADC_CH12_DATA	0x74	32	R/W	0x00	ADCaisle12data register
ADC_CH13_ST	0x78	32	R/W	0x00	ADCaisle13status register
ADC_CH13_DATA	0x7c	32	R/W	0x00	ADCaisle13data register
ADC_CH14_ST	0x80	32	R/W	0x00	ADCaisle14status register
ADC_CH14_DATA	0x84	32	R/W	0x00	ADCaisle14data register
ADC_CH15_ST	0x88	32	R/W	0x00	ADCaisle15status register
ADC_CH15_DATA	0x8c	32	R/W	0x00	ADCaisle15data register
ADC_FIFO_STAT	0xa0	32	R/W	0x04	ADC FIFOstatus register
ADC_FIFO_DATA	0xa4	32	R/W	0x00	ADC FIFOdata register
EXTTRIG_SEL	0xb0	32	R/W	0x00	External signal triggerADCchoose deposit device
ADC_CALIB_OFFSET	0xf0	32	R/W	0x00	ADCcalibrationOFFSETregister
ADC_CALIB_KD	0xf4	32	R/W	0x00	ADCcalibrationKDregister

Register description

ADC_CFGregister(0x00)

bit field	name	type	reset value	describe
31:30	RESERVED	R	0	reserved bit
29	DMA_EN	R/W	0	DMAreadFIFOEnable 0:CPUreadFIFO 1:DMAreadFIFO
28	ADC_TRIG	R/W	0	ADCTrigger source selection 0:chooseCPUtriggerADCsampling 1: select external signal triggerADCsampling



27	ADC_EN	R/W	0	ADCenable bit 0: disabled 1:Enable
26:24	IN_SMPL_WI N	R/W	010	ADCinternal sampling clock mode sampling window selection 000: sample setup time hold1indivualADCclock cycle 001: sample setup time hold3indivualADCclock cycle 010: sample setup time hold5indivualADCclock cycle 011: sample setup time hold7indivualADCclock cycle 100: sample setup time hold9indivualADCclock cycle 101: sample setup time hold11indivualADCclock cycle 110: sample setup time hold13indivualADCclock cycle 111: sample setup time hold15indivualADCclock cycle
twenty three	ADC_SMPL_C LK	R/W	0	ADCSampling mode selection 1:expressADCUsing internal sampling clock mode 0 .expressADCUsing external sampling clock mode
twenty two	ADC_MEM_ MODE	R/W	0	ADCData storage method selection: 0:ADCdata is stored asFIFOmodel; 1: ADCData is stored as channel mode;
21:19	SMPL_SETUP	R/W	010	ADCSampling window selection by external sampling clock mode 000: sample setup time hold1indivualADCclock cycle 001: sample setup time hold2indivualADCclock cycle 010: sample setup time hold4indivualADCclock cycle 011: sample setup time hold8indivualADCclock cycle 100: sample setup time hold16indivualADCclock cycle 101: sample setup time hold32indivualADCclock cycle 110: sample setup time hold64indivualADCclock cycle 111: sample setup time hold128indivualADCclock cycle
18	CONT	R/W	0	ADCSampling working mode 0: single sample 1: continuous sampling
17:16	AVG	R/W	0	one startADCSampling average times configuration register 00:1subsampling average 01:2subsampling average 10:4subsampling average 11:8subsampling average

15:0	ADC_CH_SEL	R/W	0	ADCChannel Select Register Bit15-bit0Corresponding to the channel15-aisle0. The corresponding bits are configured as1It means the channel is valid.
------	------------	-----	---	--

ADC_STARTRegister(0x04)

bit field	name	type	reset value	describe
30:4	RESERVED	R	0	reserved bit
3	FIFO_CLR	R/W	0	FIFOclear enable 0: disabled 1:Enable Write1clearFIFO
2	SOFT_RESET	R/W	0	ADCsoft reset 0Reset valid existADCAfter enabling, it must be reset once before startingADC
1	BUSY	R	0	ADCworking status 1express busy

0	START	R/W	0	<p>ADCstart signal 0: disabled 1:Enable</p> <p>This bit writes1, a conversion is initiated. CanADC_CONTCooperate with use likeADC_CONTin one-shot mode, the position1after that will</p> <p>Sequentially poll the valid channels for sampling conversion, and convert the converted data stored in the corresponding channelFIFOor register. Hardware after conversion will be automatically cleared.</p> <p>likeADC_CONTin continuous sampling mode, the position1Indicates enlightenment moveADCConversion, cleared to indicate stopADCconvert. start upADC</p> <p>After the conversion, the valid channels will be polled sequentially for sampling conversion, and the The converted data is saved in the corresponding channel'sFIFOor register. every time After the conversion is completed, judge whether the bit is1, if1then continue to convert, if for0then stop the conversion.</p>
---	-------	-----	---	---

ADC_IERegister(0x08)

bit field	name	type	reset value	describe
30:18	RESERVED	R	0	reserved bit
17	ADC_FIFO_HFULL_IE	R/W	0	ADC FIFOshalf full interrupt enable 0: disabled 1:Enable
16	ADC_FIFO_FULL_IE	R/W	0	ADC FIFOfull interrupt enable 0: disabled 1:Enable
15:0	ADC_CHx_EOC_IE	R/W	0	ADCaislexData conversion complete interrupt enable

ADC_IFregister(0x0C)

bit field	name	type	reset value	describe
30:18	RESERVED	R	0	reserved bit
17	ADC_FIFO_HF ULL_IF	R/W	0	ADC FIFO half full interrupt status Write1clear
16	ADC_FIFO_FU LL_IF	R/W	0	ADC FIFO full interrupt status Write1clear
15:0	ADC_CHx_EO C_IST	R/W	0	ADCaislexData conversion complete interrupt status write1clear

ADC_CHx_STATregister

bit field	name	type	reset value	describe
30:1	RESERVED	R	0	reserved bit
0	ADC_CH_E OC	R	0	ADCaislexData conversion complete flag 1:express ADCpair channelxA sample conversion is completed by sending the ADC_IF corresponding bit write1Clearable

ADC_CHx_DATAreregister

bit field	name	type	reset value	describe
30:16	RESERVED	R	0	reserved bit
15:12	ADC_CH_NUM	R	0	ADCThe channel number corresponding to the data
11:0	ADC_CH_DATA	R	0	ADCaislexdata register Note: After overflow, the converted data will overwrite the old data

ADC_FIFO_STATregister(0xA0)

bit field	name	type	reset value	describe
30:8	RESERVED	R	0	reserved bit
7:4	ADC_FIFO_LEVEL	R	0	ADCdataFIFOwater level 0000: Indicates when not full FIFO have 0 data; when full FIFO have 16 data; 0001: express FIFO have 1 data; 0010: express FIFO have 2 data; 0011: express FIFO have 3 data; 0100: express FIFO have 4 data; 0101: express FIFO have 5 data; 0110: express FIFO have 6 data; 0111: express FIFO have 7 data; 1000: express FIFO have 8 data; 1001: express FIFO have 9 data; 1010: express FIFO have 10 data; 1011: express FIFO have 11 data; 1100: express FIFO have 12 data; 1101: express FIFO have 13 data; 1110: express FIFO have 14 data; 1111: express FIFO have 15 data;
3	RESERVED	R	0	reserved bit
2	ADC_FIFO_EMPTY	R	1	ADCdataFIFOempty sign 1: express FIFO null 0: express FIFO non empty
1	ADC_FIFO_HFULL	R	0	ADCdataFIFO half full sign 1 : express FIFO half full 0: express FIFO not half full
0	ADC_FIFO_FULL	R	0	ADCdataFIFO full sign 1: express FIFO Full 0: express FIFO not full

ADC_FIFO_DATAregister(0xA4)

bit field	name	type	reset value	describe
30:16	RESERVED	R	0	reserved bit
15:12	ADC_FIFO_NUM	R	0	ADCThe channel number corresponding to the data
11:0	ADC_FIFO_DATA	R	0	ADCdataFIFOregister Note: After overflow, the converted data will be lost

ADC_EXTTRIG_SELregister(0xB0)

bit field	name	type	reset value	describe
30:12	RESERVED	R	0	reserved bit
11:0	EXTTRIG_SEL	R/W	0	external triggerADCsampling signal selection control Bit11-bit0: Indicates that they are used to control exttrig_in[11:0] Whether to triggerADCcontrol bit 0Indicates invalid;1Indicates valid

ADC_CALIB_OFFSETregister(0xF0)

bit field	name	type	reset value	describe
30:17	RESERVED	R	0	reserved bit
16	OFFSET_VALID	R/W	0	OFFSETIs the data valid
15:8	RESERVED	R	0	reserved bit
7:0	OFFSET	R/W	0	ADCdata calibrationOFFSETvalue. CalculatedOFFSETneed to be stored in this register. used in useADCCalibration is performed.

ADC_CALIB_KDregister(0xF4)

bit field	name	type	reset value	describe
30:17	RESERVED	R	0	reserved bit
16	KD_VALID	R/W	0	KDI is the data valid
15:10	RESERVED	R	0	reserved bit
9:0	KD	R/W	0	ADC data calibration. The fractional part of the value. Calculated. The fractional part of the value needs to be stored in this register. used in use ADCCalibration is performed. like more than 1, then the sign bit is 1, like less than 1, then the sign bit is 0.

5.20Comparators(COMP)

5.20.1 overview

This chip provides three general-purpose comparator units, and each comparator can be configured and used independently. use specificIOend

The port is used as the input port, and the specific port is shown in the block diagram of this module. They have a variety of functions, including rail-to-rail comparison

device, independently programmable configuration hysteresis voltage and independent programming configuration output filter function, etc.

5.20.2 characteristic

-Rail-to-Rail Comparator

-Hysteresis voltages are independently programmable for each comparator:0mvhysteresis,24mvhysteresis,40mvHysteresis and60mvhysteresis

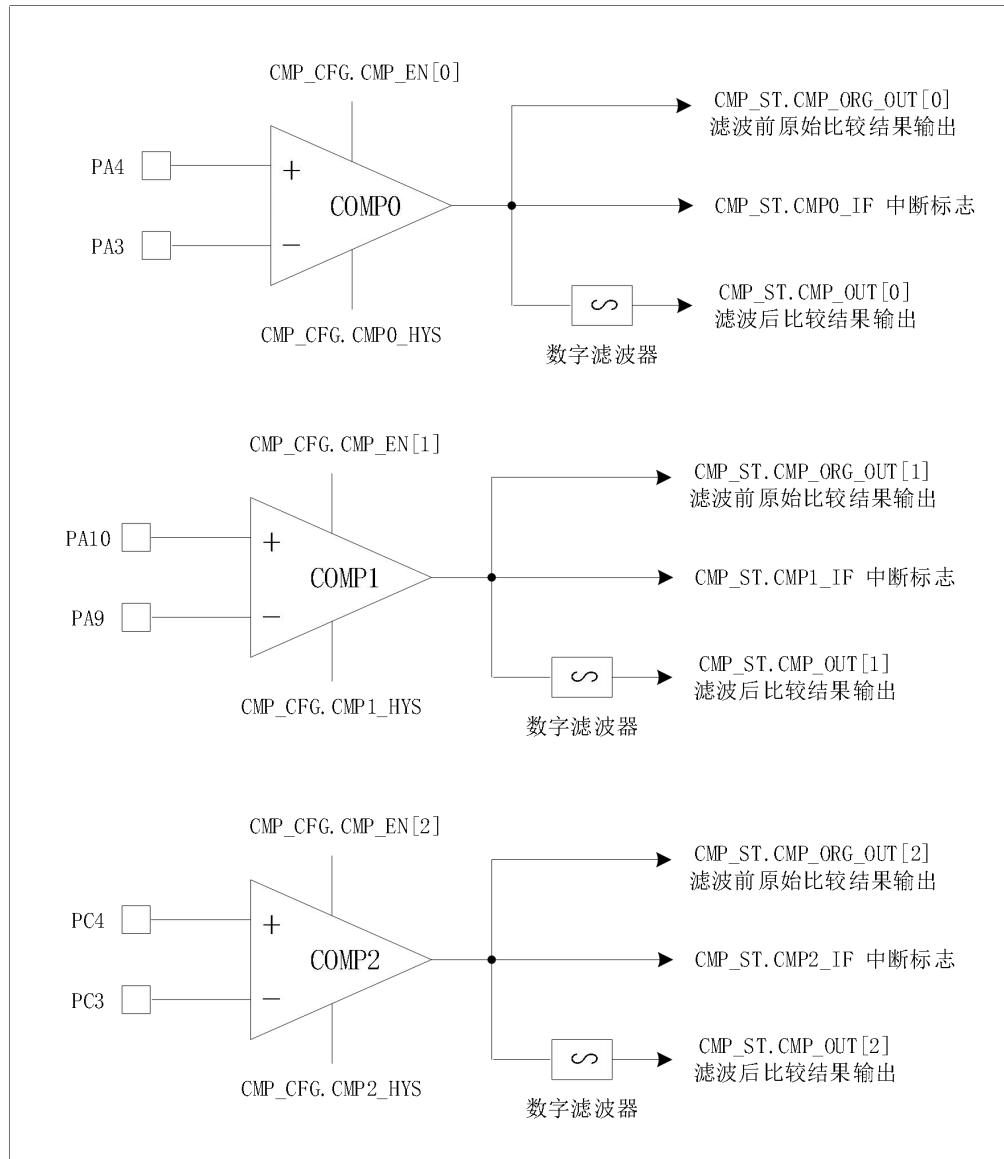
four gears

-Each comparator can be independently programmed to configure the output filter function: no filter,2system clock cycle filtering,4system

clock cycle filtering and8One system clock cycle filtering Four filtering methods

- Each comparator is capable of generating an interrupt
- Can be combined into a window comparator

5.20.3 Block Diagram of Module Structure



picture5-162Comparator Block Diagram

As shown in FIG:

Comparators 0 of VPreason PA4 pin input, VN reason PA3 pin input; and the comparator can be independently
The enable register CMP_CFG.CMP_EN[0] Direct programming.

Comparators 1 of VPreason PA10 pin input, VN reason PA9 pin input; and the comparator can be independently
The enable register CMP_CFG.CMP_EN[1] Direct programming.

Comparators 2 of VPreason PC4 pin input, VNreason PC3 pin input; and the comparator can be independently

The enable register CMP_CFG.CMP_EN[2] Direct programming.

5.20.4 Functional description

Comparator input pin and internal output signal

Three independent comparators each have independent inputs and outputs.

Comparators 0 (COMP0) through the positive end of PA4 input, negative through PA3 enter. passCMP_STregister CMP_ORG_OUT[0] Bits can be queried for raw comparison results, via CMP_STregister CMP_OUT[0] bits can be queried Filtered comparison results, via CMP_STregister CMP_IF[0] bit to poll the interrupt flag.

Comparators 1 (COMP1) through the positive end of PA10 input, negative through PA9 enter. passCMP_STregister of CMP_ORG_OUT[1] Bits can be queried for raw comparison results, via CMP_STregister CMP_OUT[1] bit can Query the filtered comparison results, through CMP_STregister CMP_IF[1] bit can poll the interrupt flag

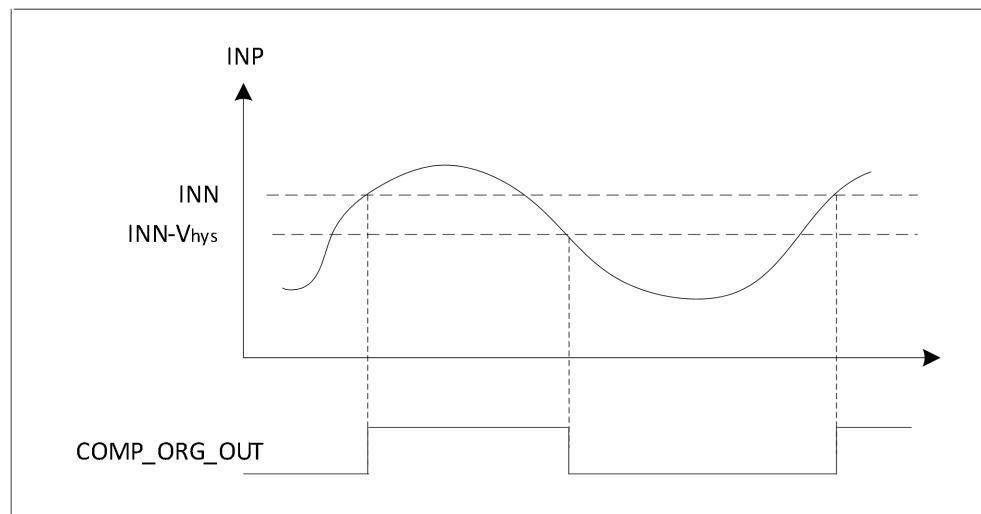
Comparators 2 (COMP2) through the positive end of PC4 input, negative through PC3 enter. passCMP_STregister CMP_ORG_OUT[2] Bits can be queried for raw comparison results, via CMP_STregister CMP_OUT[2] bits can be queried Filtered comparison results, via CMP_STregister CMP_IF[2] bit can poll the interrupt flag

hysteresis

The hysteresis function of the comparators can be controlled by the respective hysteresis register CMPx_HYS standalone configuration for avoiding interference due to resulting in an error in the compare output.

Four hysteresis gears can be selected through the programmable hysteresis register: 0mV, 24mV, 40mV and 60mV.

The diagram of comparator hysteresis is as follows:



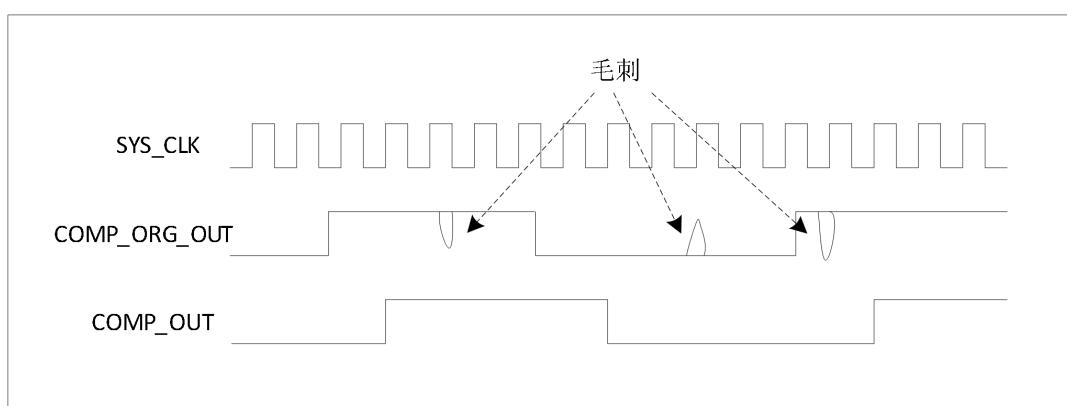
picture5-163Comparator Hysteresis Diagram

filtering

In order to improve the reliability of the comparator and reduce the misjudgment caused by interference, the output result of the comparator is digitally filtered Wave.

Through the programmable filter register CMP_CFG.CMP_FILTER four filter gears can be selected: 0system clock cycle filter, 2system clock cycle filtering, 4system clock cycle filtering and 8system clock cycle filtering.

A schematic diagram of the filtering function used to remove glitches is shown below:



picture5-164Schematic diagram of the filtering function of the comparator to eliminate glitches

This configuration is valid for all comparators.

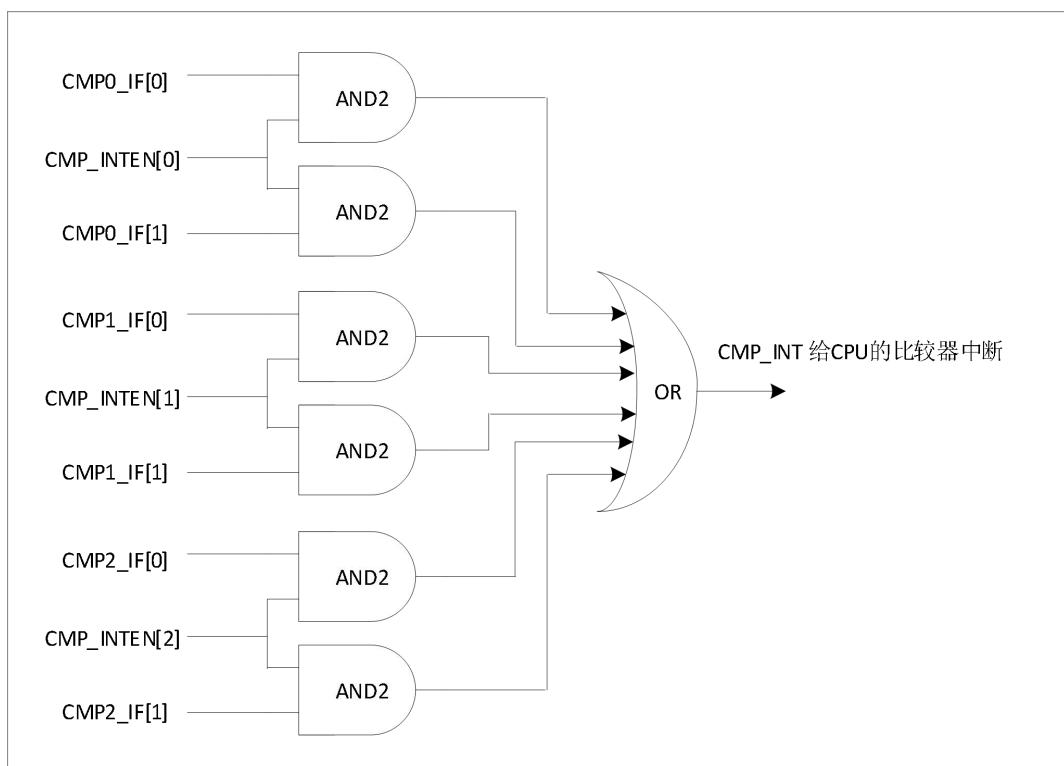
to interrupt

Each comparator has its own two comparator output status flags: one is 0arrive1change status indicator Chi; the other is expressed by 1arrive0The change status flag for .

Three independent comparators are controlled by their respective interrupt enable and output toCPUThe interrupt controller generates the chip's

Comparator interrupt source.

The interrupt flag and interrupt diagram are as follows:



picture5-165Comparator interrupt flag and interrupt diagram

register map

name	Offset	type	reset value	describe
COMPARATOR	BASE: 0x40000000			

CMP_CFG	0x120	R/W	0x200000	Comparator Configuration Register
CMP_ST	0x124	R/W	0x00	Comparator Status Register

Register description

CMP_CFGregister(0x120)

bit field	name	type	reset value	describe
31:26	RESERVED	R	0	reserved bit
25:24	CMP_FILTER	R/W	10	Comparator output filter control 00: no filtering 01:2system clock cycle filtering 10:4system clock cycle filtering 11:8system clock cycle filtering
23:22	RESERVED	R	0	reserved bit
21:20	CMP2_HYS	R/W	0	CMP2hysteresis selection 00:24mv 01:40mv 10:60mv 11:0mv
19:18	CMP1_HYS	R/W	0	CMP1hysteresis selection 00:24mv 01:40mv 10:60mv 11:0mv
17:16	CMP0_HYS	R/W	0	CMP0hysteresis selection 00:24mv 01:40mv 10:60mv 11:0mv
15:11	RESERVED	R	0	reserved bit
10:8	CMP_INTEN	R/W	0	Comparator interrupt enable control bit 1:Enable 0:closure Bit10-bit8separate controlCMP2-CMP0
7:3	RESERVED	R	0	reserved bit

2:0	CMP_EN	R/W	0	Comparator Enable Control Bit 1:Enable 0:closure Bit2-bit0separate controlCMP2-CMP0
-----	--------	-----	---	--

CMP_STregister(0x124)

bit field	name	type	reset value	describe
31:22	RESERVED	R	0	reserved bit
21:20	CMP2_IF	R/W	0	Comparators2interrupt flag Bit21for1Indicates the comparator2output occurs from0arrive1The change Bit20for1Indicates the comparator2output occurs from1arrive0A variation of both is to write1clear
19:18	CMP1_IF	R/W	0	Comparators1interrupt flag Bit19for1Indicates the comparator1output occurs from0arrive1The change Bit18for1Indicates the comparator1output occurs from1arrive0A variation of both is to write1clear
17:16	CMP0_IF	R/W	0	Comparators0interrupt flag Bit17for1Indicates the comparator0output occurs from0arrive1The change Bit16for1Indicates the comparator0output occurs from1arrive0A variation of both is to write1clear
15:11	RESERVED	R	0	reserved bit
10:8	CMP_ORG_out	R	0	Comparator result output (before filtering) 1:Pside>Nend 0:Pend<Nend Bit10~bit8RepresentingCMP2~CMP0Output
7:3	RESERVED	R	0	reserved bit
2:0	CMP_OUT	R	0	Comparator result output (filtered) 1:Pside>Nend 0:Pend<Nend Bit2~bit0RepresentingCMP2~CMP0Output

5.21 Operational Amplifier(OPAMP)

5.21.1 overview

This chip provides two independent operational amplifier units, and each amplifier can be configured and used independently. Each op amp

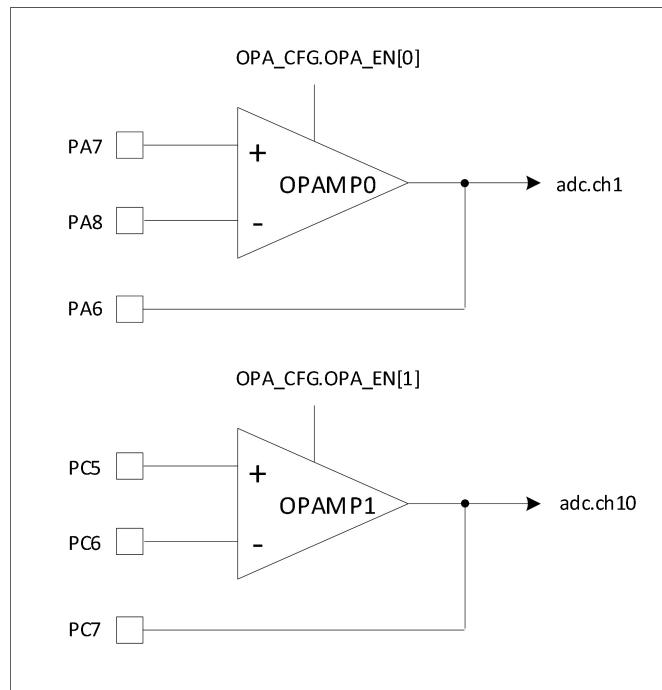
The output of the amplifier is internally connected to a specific ADC input channel, can realize ADC direct sampling measurements. make

with specific I/O. The port is used as the input terminal, and the specific port is shown in the block diagram of this module.

5.21.2 characteristic

- Rail-to-Rail Input/Output
- Positive/Negative Input
- 2 operational amplifier
- 3MHz bandwidth
- low offset voltage
- The output signal can optionally be configured to ADC channel direct acquisition

5.21.3 Module Structure Diagram



picture5-166Operational Amplifier Block Diagram

As shown in FIG:

OPAMP0 of VPreason PA7 pin input, VN reason PA8 pin input, the output can be output directly to PA6 pin or output to SARADC channel 1; and the op amp can be enabled by its independent register `OPA_CFG.OPA_EN[0]` Direct programming.

OPAMP0 of VPreason PC5 pin input, VN reason PC6 pin input, the output can be output directly to PC7 pin or output to SARADC channel 10; and the op amp can be enabled by its independent register `OPA_CFG.OPA_EN[1]` Direct programming.

register map

name	Offset	Type	reset	value	description
OPERATIONAL AMPLIFIER	BASE: 0x40000000				

OPA_CFG	0x140	R/W	0x00	Op Amp Configuration Register
---------	-------	-----	------	-------------------------------

Register description**OPA_CFG**

bit field	name	type	reset value	describe
31:2	RESERVED	R	0	reserved bit
1:0	OPA_EN	R	0	OPAenable control bit 1:Enable 0:closure Bit1-bit0separate controlOPA1-OPA0

5.22 FLASHcontroller (FLASH CTRL)

5.22.1 overview

The chip has a 64Kbyte embedded FLASH Memory, used to store application programs and chip performance

Tuning related parameter data. FLASH There are two areas: one is the main array area for storing applications or data;

another piece for NVR The area is used to store the parameter data related to the performance adjustment of this chip, etc.

for controlling internal FLASH The controller circuitry for operation is located in the CPU and FLASH between, through CPU read, write,

The wipe operation implements the FLASH read, program, and erase operations.

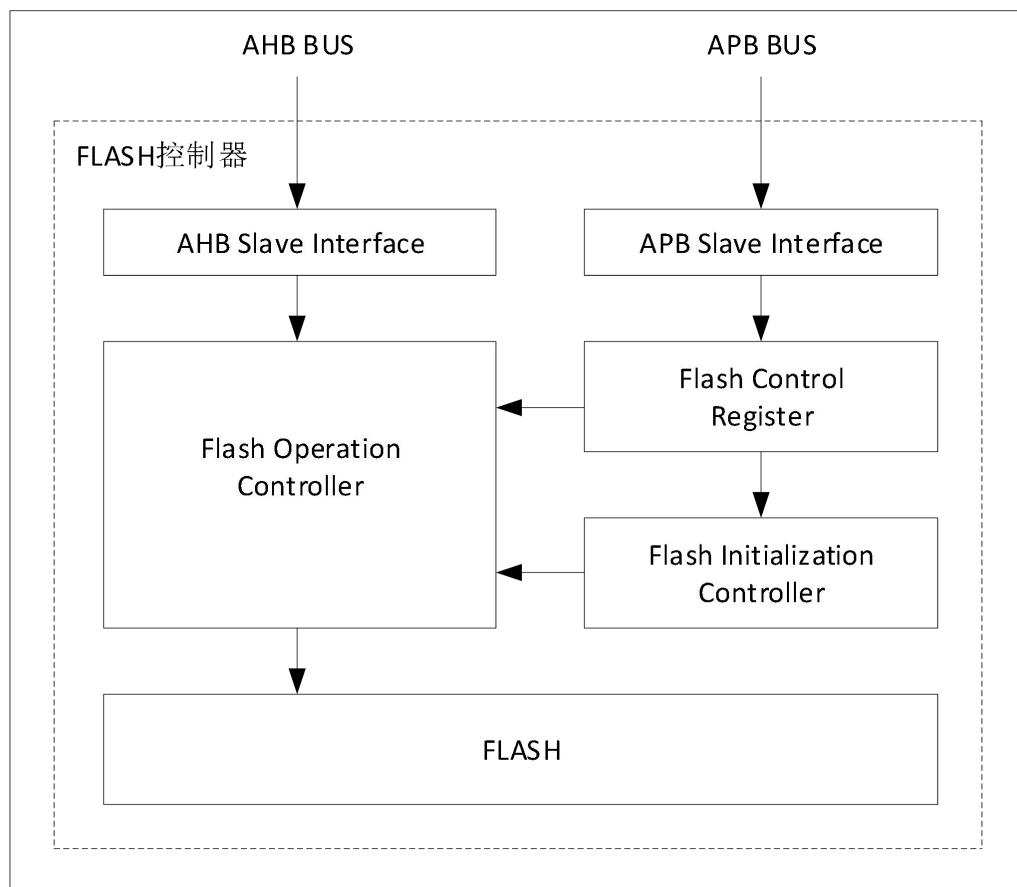
5.22.2 characteristic

- support 64KBytes Application Storage
- Support entry and exit FLASH low power mode
- The read rate is configurable. When the system clock frequency is less than or equal to 56MHz, the read rate is 1 system clock cycle etc.
Wait; when the system clock frequency is greater than 56MHz less than or equal to 84MHz, the read rate is 2 system clock cycle
wait
- configuration area (NVR area) 4 sectors, each sector 512bytes, the 1 sectors are stored in FLASH since
The user cannot operate the configuration information of itself. other 3 sector user can perform erase, read or write operation
 - The operation mode supports read operation, program operation and sector erase operation, etc.
 - Operation lock can be configured to prohibit FLASH for program and erase operations, the protection FLASH data is not modified
 - Support address shielding function, the shielding address can be configured as 2Kbyte, 4Kbyte or 8Kbyte
 - Each sector size is 512byte, support sector erase operation
 - Support continuous programming, in the same sector up to 64words (256byte) consecutively programmed

5.22.3 Block Diagram of Module Structure

FLASHController includes AHB from the interface, APB from the interface, FLASH control registers, FLASH initialize the controller, FLASH operation controller and on-chip FLASH constitute.

The schematic block diagram of the storage controller structure is shown in the figure below:



picture5-167 FLASHController Structure Diagram

AHBSlave interface: for CPU instruction and data read.

APBSlave interface: for CPU right FLASHController-related registers are programmed and configured.

FLASHControl register: used to implement FLASHThe logic of each function register and each operation required in the controller module

The relevant state in the mode generates logic.

FLASHInitialize the controller: when the chip is powered on or reset,FLASHInitializing the controller circuit will begin automatically accessing the FLASH, and readFLASHParameter data, load user configuration information toFLASHregister, completeFLASHinitial change.FLASHAfter initialization is complete, normal operation can begin.

FLASHOperational Control: According toCPUOperation via busFLASHfunctional requirements, to achieve theFLASHcorrelate operation, includingFLASHerase,FLASHprogramming andFLASHread, and generate the corresponding status signal feedback to the control register logic.

ChipFLASHMemory: On-chipFLASHThe memory is mainly used to store user programs and parameters, it includes 2Kbyte configuration area and64Kbyte program area, each sector size is512bytes, total128sectors, the minimum The programming data bit size is32bit.

5.22.4Functional description

low power mode

FLASHSupport to enter the low power consumption mode, after entering the low power consumption mode, the operating current of the chip can be greatly reduced. The way to enter low power is in theRAMConfigure inFLASH_CFGin the registerDEEP_PDLlocation1That's it. exit low power The way to consume is inRAMConfigure inFLASH_CFGin the registerDEEP_PDLlocation0That's it.

Read Rate Configuration

When the system clock is configured to a different frequency, it needs to match the corresponding read rate, otherwiseFLASHThere will be problems reading. when The system clock frequency is less than or equal to56MHz, only need1Just wait for a system clock, that is, inRAMConfigure in FLASH_CFGin the registerREAD_MDLlocation0Just; when the system clock frequency is greater than56MHzand is less than or equal to 84MHzwhen needed2system clock wait, that is, atRAMConfigure inFLASH_CFGin the registerREAD_MD Location1That's it.



NVRDistrict and MAINdistrict selection

Chip voltage and clock, etc. TRIM The value is saved in NVR area, the program needs to be initialized from NVRDistrict will TRIM

The value is read and written to the corresponding register.

NVR The zone is accessed in the RAM Configure in FLASH_CFG in the register NVR_SEL Location 1 That's it. NVR Total area 4 sectors, each sector 512 bytes, the 1 sectors are stored in FLASH configuration information, users cannot operate, the following 3 A sector user can perform erase, read and write operations.

operating mode

FLASH The operation mode supports the following three methods: read operation, program operation, sector erase operation. Configure operation mode must be at RAM in the implementation.

Programming operation: program the data to be programmed into the FLASH_ADDR The middle address is the starting address, and each time at most Program half a sector (256 bytes) and can only be programmed within half a sector.

Sector wipe operation: each sector is 512 byte, the address corresponding to the sector to be erased is given by FLASH_ADDR to specify, every time After the program or sector erase operation is complete, the operating mode must be configured back to read operation.

Read operation: pass CPU direct interview FLASH The address can be realized by FLASH read operation.

FLASH operation address

FLASH_ADDR Registers: Address registers for program/sector erase operations.

The starting address, in words, when starting a programming operation. The maximum programming data for each continuous programming is half a sector, That is, the maximum programmed data is 64 words and can only be programmed within half a sector.

When a sector wipe operation is initiated, the bit 7-bit 13 indicates the sector number to perform wiping; that is, 1 sectors 512 Byte, enter the address OK 512 byte alignment.

sector size

This chipFLASH Each sector is 512Byte, the main program area (MAINarea) a total of 64KBytes, that is 128 sector, configuration area (NVRarea) is 2KBytes, that is 4sectors. The chip supports sector erase, when programming operation Only half a sector can be programmed at most (256bytes) and can only be programmed within half a sector.

STARTOperation Start Control

when FLASH_START in the register START location 1hour, FLASH will start MODE configured in the register Command operation (read, sector erase, programming), this bit is automatically cleared after the current command operation is completed.

operation lock

Towards FLASH_LOCK register write 0x55, the operation start is locked, START can't write 1. used to protect the FLASH In case of misoperation, the power-on defaults to the locked state.

Towards FLASH_UNLOCK register write 0xAA, the operation start is unlocked, START can be written 1.

MASKFunction

MASK The address can be configured as unmasked, 2KBytes, 4KBytes and 8KBytes. can be configured MASK option lock, when configured MASK_LOCK bit configured as 1hour, MASK_SEL cannot be changed. when This bit is configured as 0hour, MASK_SEL can be modified.

Erase Time and Program Time Configuration

Depending on the system clock frequency, the erase time and program time need to be configured to meetFLASHon time corresponding requirements of the sequence.

For specific configuration, please refer to the corresponding register description.

FLASHInitialize the busy flag

whenFLASH_STregisterINIT_BUSYBit is1is in progressflashDuring initialization, for0Timetable

Indicates that the initialization is complete.

After power-on, reset and exit from low-power mode, the program needs to judge this bit, onlyFLASHAfter initialization is complete, the normal operation is possible.

controller busy flag

whenFLASH_STregisterBUSYBit is1means that the controller is in the process of the current command operation, for0 indicates that the current operation has been completed and the controller is inREADYStatus, waiting for new command operations to execute.

Programming Data Cache Register Empty Status Flag

whenFLASH_STregisterPROG_BUF_EMPTYBit is1When the programming data buffer register is empty, it can write the next word data; for0When , it means that the programming data buffer register is not empty, and the next word data cannot be written. at present word programming process (16us) to write the next word data will continue programming, otherwise the continuous programming operation will be terminated.

FLASHProgram initialization process

configurationFLASHEnter normal working mode, configure as read mode, configure read rate, erase time, programming time parameters,FLASHlock, prohibitFLASHprogramming operation.

These operations must be inRAMin the implementation.

FLASHSector wipe operation process

1)InquireFLASHThe controller is busy flag, waiting for the controller to be inREADYstate;

2) The configuration mode is sector wipe operation;

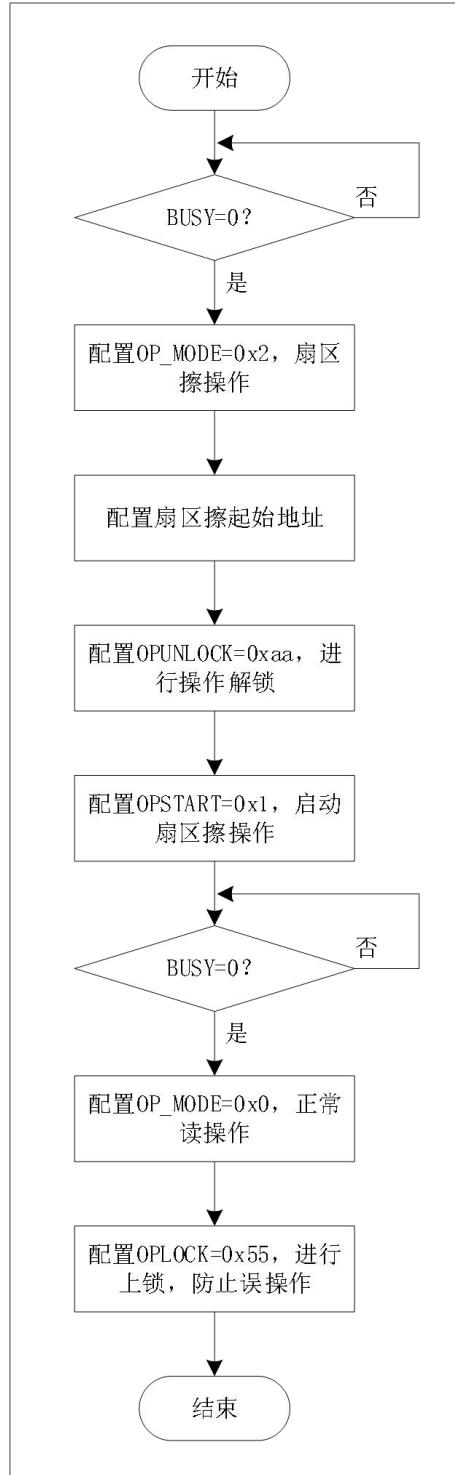
3) Write sector erase start address, in word unit;

4) configurationFLASHunlock, startSTARTOrder;

5)InquireFLASHThe controller is busy flag, waiting for the controller to be inREADYstate;

6) configuration mode for read operation,FLASHLock it and complete the sector wipe operation.

The schematic diagram of the sector wipe operation process is as follows:



picture5-168Sector wipe operation flow chart

These operations must be inRAM in the implementation.

FLASHSingle-word programming operation flow

1) InquireFLASHThe controller is busy flag, waiting for the controller to be inREADYstate;

2) The configuration mode is programming operation;

3) into the programming address, in word units;

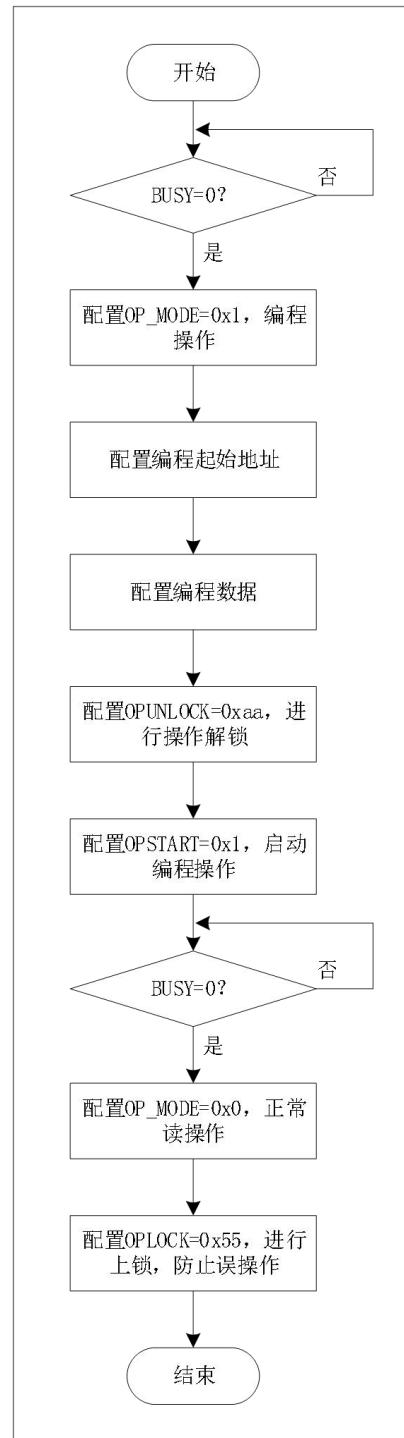
4) Put the data to be programmed into the data register;

5) configurationFLASHunlock, startSTARTOrder;

6) InquireFLASHThe controller is busy flag, waiting for the controller to be inREADYstate;

7) configuration mode for read operation,FLASHLock and complete the programming operation of the word data.

The schematic diagram of the programming operation flow is as follows:



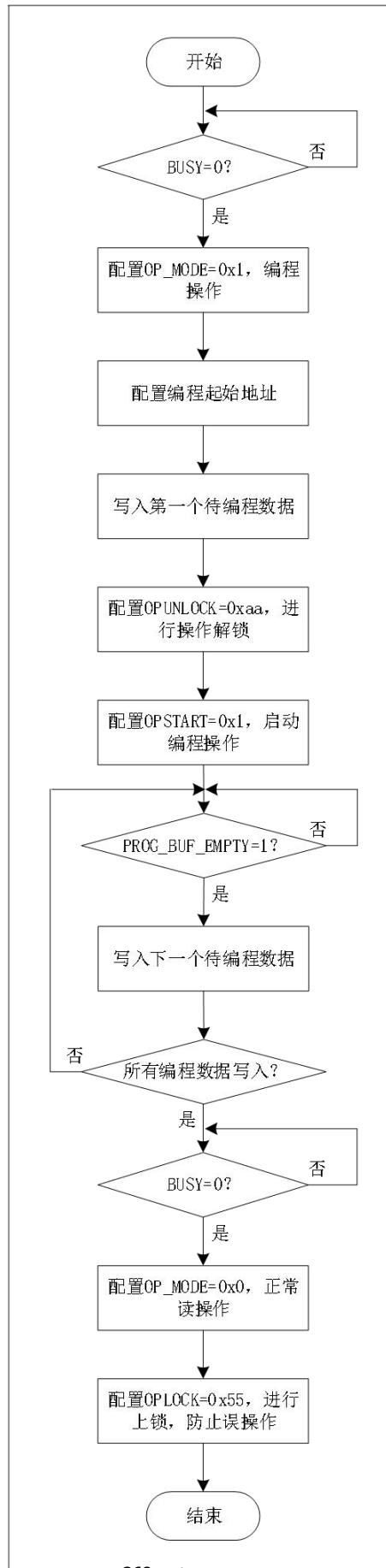
picture5-169Single-word programming operation flow chart

These operations must be inRAMin the implementation.

FLASHMulti-word continuous programming operation flow

- 1) InquireFLASHThe controller is busy flag, waiting for the controller to be inREADYstate;
 - 2) The configuration mode is programming operation;
 - 3) into the programming address, in word units;
 - 4) writes the first word of programming data into the data register;
 - 5) configurationFLASHunlock, startSTARTOrder;
 - 6) If there is still data to be programmed, then judgePROG_BUF_EMPTYIs the bit0,for0then you can write the next words to be programmed (for1wait, cannot write), until all the data to be programmed are written;
 - 7) InquireFLASHThe controller is busy flag, waiting for the controller to be inREADYstate;
 - 8) configuration mode for read operation,FLASHLocked to complete the operation of all the data to be programmed this time, among which multiple words are continuous
- The maximum amount of programming data is 64 words and can only be programmed within half a sector.

The schematic diagram of the programming operation flow is as follows:



picture5-170Multi-word continuous programming operation flow chart

These operations must be inRAM in the implementation.

register map

name	Offset	bit width	type	reset value	describe
FLASH_CTRL	BASE: 0x4006F000				
FLASH_CFG	0x00	32	R/W	0x80000000	configuration register
FLASH_ADDR	0x04	32	R/W	0x0	Address Configuration Register
FLASH_WDATA	0x08	32	W	0x0	write data register
FLASH_START	0x10	32	R/W	0x0	Operation Start Register
FLASH_ST	0x14	32	R	0x5	status register
FLASH_LOCK	0x18	32	W	0x0	Operation Lock Control Register
FLASH_UNLOCK	0x1c	32	W	0x0	Operation Unlock Control Register
FLASH_MASK	0x20	32	R/W	0x2	mask control register
FLASH_ERASETIME	0x24	32	R/W	0x7532a31b	Erase Time Parameter Configuration Register
FLASH_PROGTIME	0x28	32	R/W	0x1f4360	Program Time Parameter Configuration Register

Register description

FLASH_CFGregister(0x00)

bit field	name	type	reset value	describe
31	DEEP_PD	R/W	1	<p>configurationFLASHenter low power mode 1: enter low power mode 0: normal working mode</p> <p>Note: This operation can only be done inRAM in the implementation.</p>



30:5	RESERVED	R	0	reserved bit
4:2	MODE	R/W	0	<p>Operation Mode Configuration Register</p> <p>000: Normal read operation. for AHB mode of operation when reading.</p> <p>001: Programming operation. Program the data to be programmed into the FLASH_ADDR. The middle address is the starting address of the middle. And program up to half a sector at a time (256bytes) and can only be programmed within half a sector.</p> <p>010: sector erase operation (each sector is 512byte). The address corresponding to the erased sector is given by FLASH_ADDR to specify.</p> <p>Other: reserved</p> <p>Note1: All the above operations can only be performed in RAM in the implementation.</p> <p>Note2: After each other operation mode (non-normal read operation mode) is completed, MODE must be sure to configure back to 0.</p>
1	NVR_SEL	R/W	0	<p>NVR district selection</p> <p>0: choose Main Array (common 128 sectors, each sector 512 byte)</p> <p>1: choose NVR sector (total 4 sectors, each sector 512 byte) Note: This register change must be in RAM in the implementation.</p>
0	READ_MD	R/W	0x0	<p>Read Rate Mode Selection</p> <p>1: 2 system clock cycle wait (56MHz < sys_clk < 84MHz) 0: 1 system clock cycle wait (sys_clk <= 56MHz)</p> <p>NOTE: This register change must be in RAM in the implementation.</p>

FLASH_ADDR Register(0x04)

bit field	name	type	reset value	describe
31:14	RESERVED	R	0	reserved bit
13:0	ADDR	R/W	0	<p>Address Register for Program/Sector Erase Operations</p> <p>The starting address, in words, when starting a programming operation. The maximum programming data for each consecutive programming is half a sector, that is, the maximum programming data is 64 words, and can only be programmed within half a sector;</p> <p>When a sector wipe operation is initiated, the bit 7-bit 13 indicates the sector number to perform wiping; that is, 1 sector 512 bytes</p>

FLASH_WDATRegister(0x08)

bit field	name	type	reset value	describe
31:0	WDATA	R/W	0	<p>Data Register for Program Operation</p> <p>When starting a program operation, program the data in this register into the corresponding address of</p>

FLASH_STARTregister(0x10)

bit field	name	type	reset value	describe
31:1	RESERVED	R	0	reserved bit
0	START	R/W	0	<p>Operation start control bit</p> <p>write the bit1, then startMODEThe command operation configured in the register.</p> <p>This bit is automatically cleared to 0 after the current command operation is completed</p>

FLASH_STregister(0x14)

bit field	name	type	reset value	describe
31:3	RESERVED	R	0	reserved bit
2	PROG_BUF_EMPTY	R	1	<p>Programming Data Cache Register Empty Status Flag</p> <p>1: Indicates that the programming data buffer register is empty. The next word data can be written.</p> <p>0: Indicates that the programming data buffer register is not empty, and the next word data cannot be written.</p> <p>Writing the next word data during the current word programming will result in continuous programming, otherwise the continuous programming operation will be terminated.</p>

1	BUSY	R	0	controller busy flag 1: Indicates that the controller is in the process of the current command operation. 0: Indicates that the controller is in the READY Status, waiting for the command operation to execute.
0	INIT_BUSY	R	1	FLASH Initialize the busy flag 1: means in progress flash During initialization. 0: Indicates that the initialization is complete

FLASH_LOCKregister(0x18)

bit field	name	type	reset value	describe
31:8	RESERVED	R	0	reserved bit
7:0	LOCK	W	0x0	Operation lock control write to this register 0x55, the operation start is locked, START can't write 1. used to protect the FLASH misuse. It is locked by default when powered on.

FLASH_UNLOCKregister(0x1C)

bit field	name	type	reset value	describe
31:8	RESERVED	R	0	reserved bit
7:0	UNLOCK	W	0	Operation Unlock Control write to this register 0xAA, the operation start is unlocked, START can be written 1.

FLASH_MASKregister(0x20)

bit field	name	type	Reset value	description

31:3	RESERVED	R	0	reserved bit
2	MASK_LOCK	R/W	1	<p>MASKselect lock control</p> <p>When this bit is configured as1hour,MASK_SELCannot be changed. When this bit is configured for0hour,MASK_SELcan be modified.</p>
1:0	MASK_SEL	R/W	0	<p>MASKchoose</p> <p>00: Unshielded</p> <p>01:lowest2KB(i.e. minimum4sectors) are masked 10</p> <p>:lowest4KB(i.e. minimum8sectors) are masked 11</p> <p>:lowest8KB(i.e. minimum16sectors) are masked</p>

FLASH_ERASETIMEregister(0x24)

bit field	name	type	reset value	describe
31	RESERVED	R	0	reserved bit
30:19	TRCV	R/W	0xea6	<p>FLASHWhen sector is erased,TRCVtime register</p> <p>Depending on the frequency of the system clock, it is necessary toTRCVproperly configured so that its time is at least longer than50usrequirements.</p> <p>Among them, the sector erase operation can be performed when the system clock is up to72MHzcase work</p> <p>Note1: The default value is the system clock is48MHzcase, the configuration value for the sector wipe operation.</p>
18:0	TERASE	R/W	0x2a31b	<p>FLASHWhen sector is erased,TERASEtime register</p> <p>Depending on the frequency of the system clock, it is necessary toTERASEproperly configured so thatTERASEtime at3.2ms-4mswithin range.</p> <p>Among them, the sector erase operation can be performed when the system clock is up to72MHzwork under circumstances;</p> <p>Note1: The default value is the system clock is48MHzcase, the configuration value for the sector wipe operation.</p>

FLASH_PROGTIMErregister(0x28)

bit field	name	type	Reset value	description

31:22	RESERVED	R	0	reserved bit
21:11	TPGS	R/W	0x3e8	<p>FLASH during programming operation TPGS time control register.</p> <p>Depending on the frequency of the system clock, it is necessary to TPGS properly configured so that its time is at least longer than 20us requirements.</p> <p>Note: The default is the system clock at 48MHz the case</p>
10:0	TPROG	R/W	0x360	<p>FLASH during programming operation TPROG time control register.</p> <p>When programming, TPROG time needs to be in 16us-20us scope;</p> <p>Depending on the frequency of the system clock, it is necessary to TPROG properly configure it so that it is within the proper range.</p> <p>Note: The default is the system clock at 48MHz the case</p>

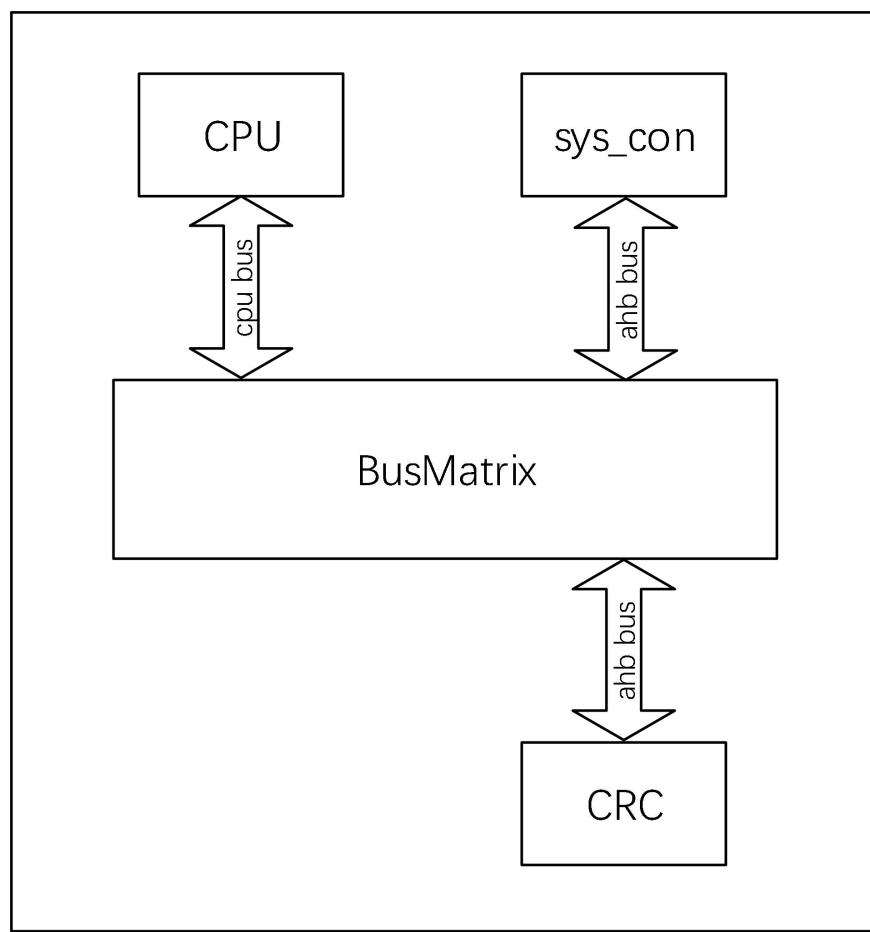
5.23 Cyclic Redundancy Check (CRC)

5.23.1 overview

This chip CRC It is mainly used to reduce the bit error rate of communication lines during data communication. accessible AHB bus interface

The transmitted data is calculated by this module, and the reliability of data transmission is ensured through the result of cyclic redundancy check.

Its system block diagram is as follows:



picture5-171CRCModule System Block Diagram

5.23.2 characteristic

-support pair 8, 16, 32bit data CRC operation

-Support the inversion of input data and output data;

- Support input data and output data flipping;

- Support the following four CRC polynomial:

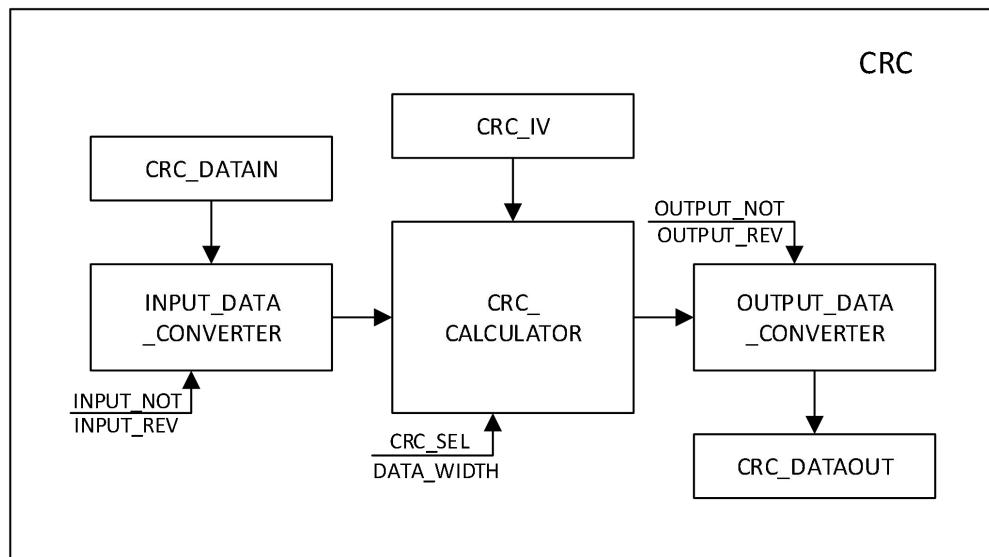
$$x^{16}+x^{12}+x^5+1,$$

$$x^8+x^2+x+1,$$

$$x^{16}+x^{15}+x^2+1,$$

$$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x^1+1.$$

5.23.3 Module Structure Diagram



picture5-172CRCModule Structure Diagram

Pictured above is the schematic diagram of the internal structure of the module. Input data CRC_DATAIN After negation and flipping, enter CRC calculator, counter based on CRC initial value, valid data bit setting and selected CRC polynomial to carry out CRC calculate The calculation result is output to CRC_DATAOUT.

5.23.4 Functional description

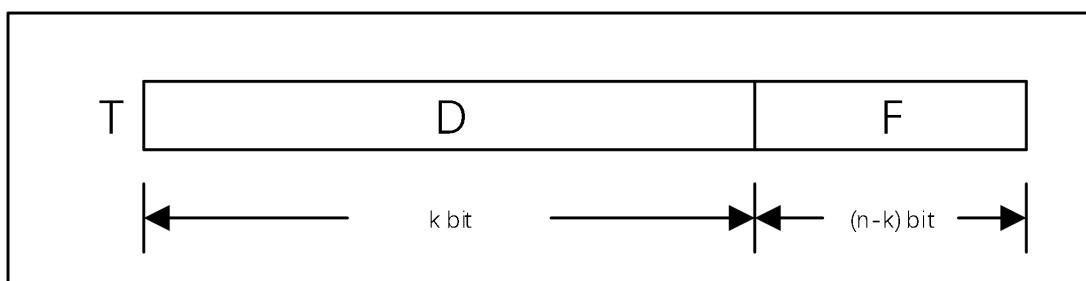
CRC code generation rules

CRCThe code consists of two parts, the first part is the information code, which is the information that needs to be verified, and the latter part is the verification code. exist

The sender (memory write) depends on the k bit binary code sequence, with certain rules to generate a verification rbit supervision code (CRC code), appended to the original information to form a new binary code sequence with a total of k+rbit, then send out the go. At the receiving end (memory read), according to the information code and CRCThe rules followed between the codes are checked to determine the transmission Is there an error in .

if CRC code is long total n bit, information code D is long k bit, the remaining r bit off is the check digit. like (7,3) code:

110 1001, the first three 110 is the information code, 1001 is the check code. As shown below, is a CRC schematic diagram of the code structure.



picture5-173CRC schematic diagram of code composition

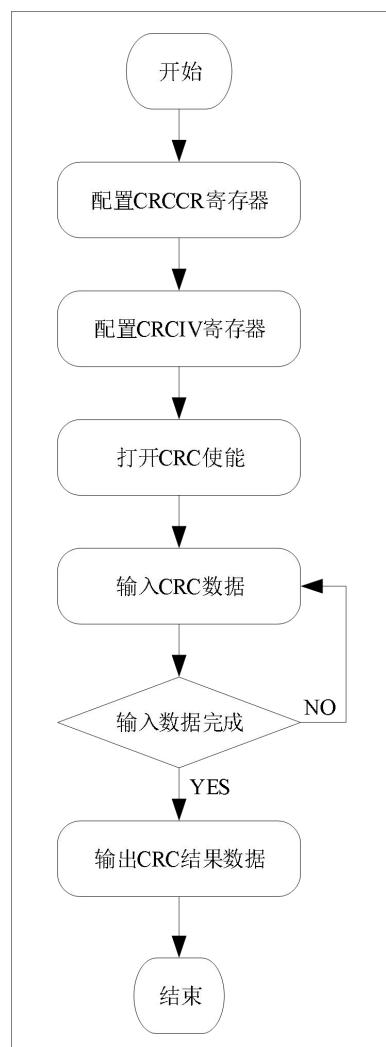
CRCThe generation rules of the check code can be summarized as the following steps:

1. Shift the original information code to the left CRC polynomial bits, zero-padded on the right; e.g. binary data 110 conduct 4 bit CRC school
- Check code generation -> 110 0000;
2. use 110 0000 divide by CRC polynomial (note that the modulo 2 division), the remainder is CRC school
- check code;
3. Continue the check code to the end of the information code to form CRC code.

CRCpolynomial

CRCmodel	statement name	CRCcheck digit	corresponding polynomial
CRC-CCITT	CRC_CCITT	16	$x^{16}+x^{12}+x^5+1$
CRC-8	CRC_8	8	x^8+x^2+x+1
CRC-16	CRC_16	16	$x^{16}+x^{15}+x^2+1$
CRC-32	CRC_32	32	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$

Operating procedures



picture5-174CRCOperation flow chart

register map

name	Offset	bit width	type	reset value	describe
CRC	BASE: 0x40003000				
CRC_CR	0x00	32	R/W	0x00	CRCcontrol register
CRC_IV	0x04	32	W	0x00	CRCinitial value register
CRC_DATAIN	0x08	32	R/W	0x00	CRCinput data register
CRC_DATAOUT	0x0c	32	R	0x00	CRCoutput data register

Register description

CRC_CRregister(0x00)

bit field	name	type	reset value	describe
31:11	RESERVED	RO	0	reserved bit
10:9	CRC_SEL	R/W	0	<p>CRCAlgorthm Selection Register</p> <p>00:x 16+x^12+x 5+1</p> <p>01:x 8+x 2+x+1 10:x</p> <p>16+x^15+x 2+1</p> <p>11:</p> <p>x 32+x^26+x 23+x 22+x^16+x^12+x 11+x 10+x^8+x</p> <p>7+x 5+x^4+x 2+x+1</p>
8:7	DATA_WIDTH	R/W	0	<p>CRCInput Data Significant Bits Register</p> <p>00:32bit input data valid</p> <p>01:Low16bit input data valid</p> <p>10:Low8bit input data valid</p> <p>11:reserve</p>



6:5	OUTPUT_INV	R/W	0	Output Data Toggle Register The flipping rules are the same as INPUT_INV
4	OUTPUT_REV	R/W	0	Whether the output data is reversed 1: Invert the output data 0: The output data is not inverted
3:2	INPUT_INV	R/W	0	Input Data Toggle Register 00:bitorder unchanged 01:bitThe order is completely reversed 32bit data width31:0 -> 0:31; 16bit data width15:0 -> 0:15; 8bit data width7:0 -> 0:7 10:bitThe order is flipped across byte ranges 32bit data width31:0 -> 24:31, 16:23, 8:15, 0:7; 16bit data width15:0 -> 8:15, 0:7; 8Bit data width is the same as01 11: Byte order flip only 32bit data width31:0 -> 7:0,15:8,23:16,31:24; 16bit data width15:0 -> 7:0,15:8; 8Bit data width is the same as00
1	INPUT_REV	R/W	0	Whether the input data is reversed 1: Invert the input data 0: input data is not negated
0	CRC_EN	R/W	0	CRCenable bit 1:CRCEnable 0:CRCnot enabled

CRC_IVregister(0x04)

bit field	name	type	reset value	describe
31:0	CRC_IV	R/W	0	CRCinitial value register

CRC_DATAINregister(0x08)

bit field	name	type	reset value	describe
31:0	CRC_DATAIN	W	0	<p>CRCinput data register</p> <p>Note: whenDATA_WIDTHfor</p> <p>00:32bit valid</p> <p>01:Low16bit valid</p> <p>10:Low8bit valid</p> <p>11:reserve</p>

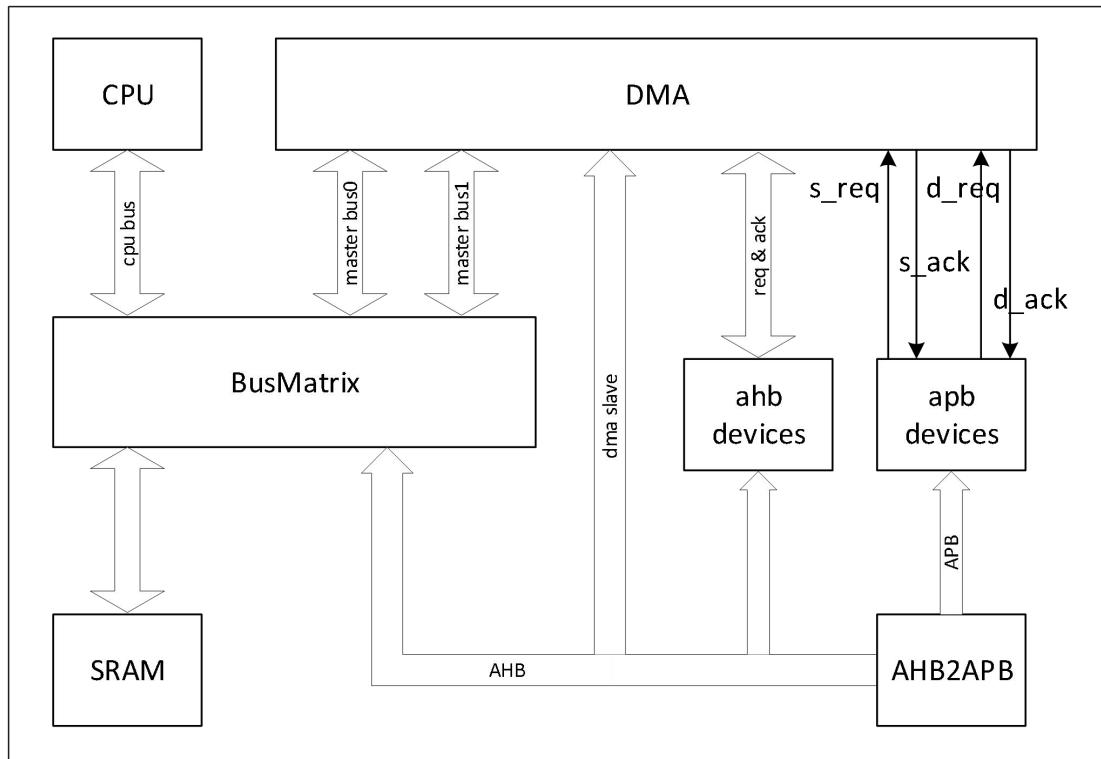
CRC_DATAOUTregister(0x0C)

bit field	name	type	reset value	describe
31:0	CRC_DATAOUT	W	0	<p>CRCoutput data register</p> <p>Note: whenCRC_SELfor</p> <p>00:Low16bit valid</p> <p>01:Low8bit valid</p> <p>10:Low16bit valid</p> <p>11:32bit valid</p>

5.24 DMA controller (DMA)

5.24.1 overview

The module is DMA controller, which is located between two peripherals, is mainly used to realize the connection between the system storage device and the peripherals data transfer. The schematic diagram of its system connection is as follows:



picture5-175 DMAController system connection diagram

5.24.2 characteristic

-This chip DMA The number of channels is 4.

-has two AHB Master console interface: one is used to read data from the source address device, and the other is used to send data to the destination

Write data to the address device;

-The source address device and the destination address device can configure the transmission data width separately 8, 16 and 32 bits, source address and destination

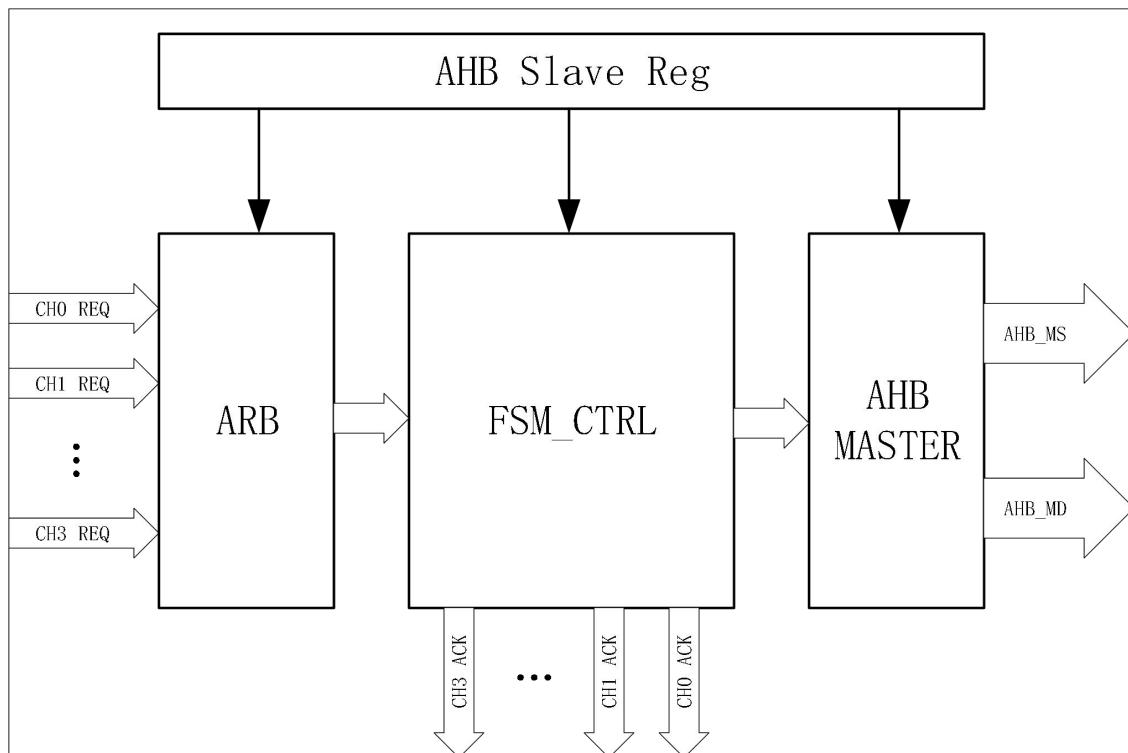
The address must be aligned according to the data transfer width;

-Each channel can be configured to transfer up to 4096 bytes of data. After each channel transmits a piece of data, the channel releases the bus.

The circuit will re-judge the priority of each channel, and the channel with higher priority will be transmitted first.

- Support channels can be configured for incremental or fixed address mode.
- Support three ways between memory to memory, memory to peripheral, and peripheral to memory
- Supports cyclic restart DMA
- Support configurable priority for each channel
- Supports independent interrupts per channel

5.24.3 Schematic diagram of module structure



picture5-176 DMA schematic diagram of module structure

5.24.4 Functional description

DMA controller with CPU. The core shares the system bus matrix and can directly execute memory-to-memory, memory-to-data transfer between peripherals, peripherals, and memory. When CPU and DMA simultaneous access to the same target (RAM or peripherals) hour, DMA request will be paused. CPU access to the bus for several cycles, the bus arbiter will perform an arbitration operation to ensure CPU At least half of the system bus bandwidth is available.

DMA processing flow

memory (RAM) to memory (RAM)

If desired DMA uses CHn channel transfers data from RAM A space moved to RAM another space. match Setting the source side of the channel will CHn MOD. MS_SEL register is configured as 0, the destination side will CHn MOD. MD_SEL register also configured as 0, and both the source and destination addresses are configured as RAM address space, other configuration items for this channel are configured When finished, set the CHn CON. SWREQ configured as 1, then the data transfer will be started, and each data will be configured according to the number SIZE size DMA via the source-side master controller from the RAM. The source address is read out, after DMA After processing, through the main control of the destination side device write RAM in the destination address.

memory (RAM) to the peripheral

If wish DMA uses CHn channel transfers data from RAM Moved to peripherals. Then configure the source side of the channel to be CHn MOD. MS_SEL register is configured as 0, the destination side will CHn MOD. MD_SEL After the register is configured as the corresponding peripheral, and other configuration items of the channel are configured, set CHn CON. SWREQ match Set as 1, then the data transfer will be started, and each data will be configured according to the number SIZE size DMA Through the source side main controller

from RAM. The source address is read and passed through DMA processing, after waiting for the request signal from the destination side, pass the processed data to

Send it to the peripheral through the bus, and at the same time send the response signal corresponding to this peripheral request to the peripheral, which is used for the peripheral to clear this time

A valid request, so that the next request can be initiated normally.

peripheral to memory (RAM)

if wishDMAuseCHnChannels move data from peripherals to RAM.

Then configure the source side of the channel to be CHnMOD.MS_SEL register configuration for the corresponding peripheral, the destination side will

CHnMOD.MD_SEL register is configured as 0, and after other configuration items of this channel are configured, enable this channel,

Waiting for the request signal sent by the peripheral on the source side, after receiving the request signal, it starts to transfer the data, each data according to

Number of configurations SIZEsizeDMARead from the source address of the peripheral by the master controller on the source side, after DMA processing, pass

Destination host controller writes RAM in the destination address, at the same time, the response signal of this request is returned to the source side peripheral for

The peripheral on the source side clears this request so that the next request can be initiated normally.

DMA Arbiter processing

Each channel can independently configure the priority level, and the arbitrator starts the memory or

Peripheral access transfer.

DMA The priority of each channel is passed through CHnCON.PRI to configure, a total of 4 levels:

---low priority

---medium priority

---high priority

---highest priority

If two or more channels initiate requests at the same moment of arbitration, the transmission of this data will be prioritized first.

Higher-level channels perform data transfer. If the priority is the same, the lower-numbered channel has priority over the higher-numbered channel.

Example: channel2priority over channel3.

existDMADuring the data transfer of a certain channel, if it receives requests from other channels, these requests will be put on hold.

It will not be released until the data of this channel has been transferredDMAof control. The next transmission is selected by punching again

Select which channel is transmitted.

DMAChannel configuration information

Each channel has its own channel-related configuration registers and status registers, which can be independently address between the peripheral registers and the memoryDMAdata transmission.DMAThe amount of data transferred is programmable up to a maximum support4096For pen data transmission, the size of each data bit is also programmed. With the register of the number of data transfers, you can know the current How many times data has been transferred, this register is incremented after each transfer completes.

source address

passChnMSADDRRegisters can be configuredDMAsource address of the transferred peripheral or memory. Data transfer occurs , this address will be used as the source of the data transmission.

Destination address

passChnMDADDRRegisters can be configuredDMAthe destination address of the transferred peripheral or memory. Data transmission occurs When sending, this address will be used as the destination of data transmission.

channel priority

passCHnCON.PRIThe register can configure the priority level of the current channel, a total of4levels are selectable. every time

The arbitration period is compared with the priority level of each channel, and the priority level of the higher level is transmitted first.

cycle mode

passCHnCON.CIRMDThe register configures the cycle mode of the current channel. This register is configured as0is not enabled

Cyclic mode, after all the data volume configured by the channel has been transmitted, the transmission will stop; the register is configured as1when

Turn on the loop mode, and after all the data volume configured by the channel is transmitted, it will automatically follow the configuration information of the channel,

DMAData transfers will be restarted until the channel is closed.

amount of data transferred

passCHnCON.LENTHThe register can configure the transmission data volume of the current channel, and the maximum supported4096pen data.

passCHnMOD.MD_SIZEThe register configures the size of the destination bus write data format, throughCHnMOD.MS_SIZEsend

The register configures the size of the bus read data format on the source side. These two registers can be programmed as8bit,16bitand32bit.

channel enable

passCHnCON.CH_ENThe register can be configured to enable or disable the current channel. When the relevant configuration information of the channel is written into the phase

After closing the configuration register, configure the register as1, the configuration information takes effect and waits for a valid request signal from the source side.

When this register is configured as1Afterwards, if the loop mode is not turned on, when a valid request from the source side is received, the data processing starts

transmission, when the configured data volume is completely transmitted, the hardware will automaticallyCHnCON.CH_ENRegisters are cleared as0,closure

At this time, even if a valid request from the source side is received, the channel will not be started. If you want to enable the channel transmission again, then

Software is required to reprogram this register to1.

When this register is configured as1After that, if the loop mode is enabled, data transmission will start after receiving a valid request from the source side.

When the configured data volume is completely transmitted, the transmitted data volume counter starts counting again.DMAaccording to the previous

Configuration information and then data transmission. If data transmission is no longer desired, software programming is required to set this registerCHnCON.CH_EN

configured as0, close the channel, even if the request from the source side is received, no more data transmission will be performed. Note that when the software will

CHnCON.CH_ENRegisters are cleared as0, the channel will not be closed immediately, and the hardware will

The channel will not be closed automatically until all transmissions are completed.

Channel Mapping Selection

Each channel can independently select the corresponding memory or peripherals on the source and destination sides.

The source side can pass throughCHnMOD.MS_SELRegister programming selection, the destination side can be selected byCHnMOD.MD_SELregister programming options. Each of these two registers can select multiple devices (memory or peripherals), where the configuration is0x00choose for memory, usually on-chipRAMmemory. Can also be used for other peripherals that do not have to wait for access without handshaking, configured as0x01-0x07, select as a peripheral with a handshaking function.

Channel address change mode

Each channel can independently select the address change mode of the source side and the destination side. There are two options for address change: address increment and address constant.

Among them, the source side can pass throughCHnMOD.MS_ADDMODRegister programming selection, the destination side can be selected by CHnMOD.MD_ADDMODRegister programming selection.

Source address and destination address data format description

The data format of the source address and destination address can be configured separately, and the corresponding relationship is shown in the following example:

inF0/F1/F2/F3/F4/F5/F6/F7/F8/F9/FA/FB/FC/FD/FE/FF Both represent a byte of data.

picture5-177 DMAData format configuration table of source address and destination address

source Width Spend	head of Width Spend	transmission length	source (address/data)	transfer operation	purpose (address/number according to)
8	8	4	0x0/F0 0x1/F1 0x2/F2 0x3/F3	1:exist0x0readF0[7:0],exist0x0WriteF0[7:0] 2: exist0x1readF1[7:0],exist0x1WriteF1[7:0] 3: exist0x2readF2[7:0],exist0x2WriteF2[7:0] 4: exist0x3readF3[7:0],exist0x3WriteF3[7:0]	0x0/F0 0x1/F1 0x2/F2 0x3/F3
8	16	4	0x0/F0 0x1/F1 0x2/F2 0x3/F3	1:exist0x0readF0[7:0],exist0x0Write00F0[15:0] 2: exist0x1readF1[7:0],exist0x2Write00F1[15:0] 3: exist0x2readF2[7:0],exist0x4Write00F2[15:0] 4: exist0x3readF3[7:0],exist0x6Write00F3[15:0]	0x0/00F0 0x2/00F1 0x4/00F2 0x6/00F3
8	32	4	0x0/F0 0x1/F1 0x2/F2 0x3/F3	1:exist0x0readF0[7:0],exist0x0Write000000F0[31:0] 2: exist0x1readF1[7:0],exist0x4Write000000F1[31:0] 3:exist 0x2readF2[7:0],exist0x8Write000000F2[31:0] 4:exist0x3 readF3[7:0],exist0xCWrite000000F3[31:0]	0x0/000000F0 0x4/000000F1 0x8/000000F2 0xC/000000F3
16	8	4	0x0/F1F0 0x2/F3F2 0x4/F5F4 0x6/F7F6	1:exist0x0readF1F0[15:0],exist0x0WriteF0[7:0] 2: exist0x2readF3F2[15:0],exist0x1WriteF2[7:0] 3: exist0x4readF5F4[15:0],exist0x2WriteF4[7:0] 4: exist0x6readF7F6[15:0],exist0x3WriteF6[7:0]	0x0/F0 0x1/F2 0x2/F4 0x3/F6
16	16	4	0x0/F1F0 0x2/F3F2 0x4/F5F4 0x6/F7F6	1:exist0x0readF1F0[15:0],exist0x0WriteF1F0[15:0] 2: exist0x2readF3F2[15:0],exist0x2WriteF3F2[15:0] 3: exist0x4readF5F4[15:0],exist0x4WriteF5F4[15:0] 4: exist0x6readF7F6[15:0],exist0x6WriteF7F6[15:0]	0x0/F1F0 0x2/F3F2 0x4/F5F4 0x6/F7F6
16	32	4	0x0/F1F0 0x2/F3F2 0x4/F5F4 0x6/F7F6	1:exist0x0readF1F0[15:0],exist0x0Write0000F1F0[31:0] 2: exist0x2readF3F2[15:0],exist0x4Write0000F3F2[31:0] 3: exist0x4readF5F4[15:0],exist0x8Write0000F5F4[31:0] 4: exist0x6readF7F6[15:0],exist0xCWrite0000F7F6[31:0]	0x0/0000F1F0 0x4/0000F3F2 0x8/0000F5F4 0xC/0000F7F6
32	8	4	0x0/ F3F2F1F0 0x4/ F7F6F5F4 0x8/FBFAF9F8 0xC/FFFEDFC	1:exist0x0readF3F2F1F0[31:0],exist0x0WriteF0[7:0] 2: exist0x4readF7F6F5F4[31:0],exist0x1WriteF4[7:0] 3: exist0x8readFBFAF9F8[31:0],exist0x2WriteF8[7:0] 4: exist0xcreadFFFEDFC[31:0],exist0x3WriteFC[7:0]	0x0/F0 0x1/F4 0x2/F8 0x3/FC



32	16	4	0x0/ F3F2F1F0 0x4/ F7F6F5F4 0x8/FBFAF9F8 0xC/FFFEFDFO	1:exist0x0readF3F2F1F0[31:0],exist0x0WriteF1F0[15:0] 2: exist0x4readF7F6F5F4[31:0],exist0x2WriteF5F4[15:0] 3: exist0x8readFBFAF9F8[31:0],exist0x4WriteF9F8[15:0] 4: exist0xcreadFFFEFDFO[31:0],exist0x6WriteFDFO[15:0]	0x0/F1F0 0x2/F5F4 0x4/F9F8 0x6/FFFDFO
32	32	4	0x0/ F3F2F1F0 0x4/ F7F6F5F4 0x8/FBFAF9F8 0xC/FFFEFDFO	1:exist0x0readF3F2F1F0[31:0],exist0x0WriteF3F2F1F0[31:0] 2: exist0x4readF7F6F5F4[31:0],exist0x4WriteF7F6F5F4[31:0] 3:exist 0x8readFBFAF9F8[31:0],exist0x8WriteFBFAF9F8[31:0] 4:exist0xcreadFFFEFDFO[31:0],exist0xCWrite FFFEFDFO[31:0]	0x0/ F3F2F1F0 0x4/ F7F6F5F4 0x8/FBFAF9F8 0xC/FFFEFDFO

Interrupt Description

Each channel can beDMAinterrupts are generated when the transfer is halfway through and when the transfer is complete. In order to meet the application flexibility and real

For usability considerations, these interrupts can be enabled by programming different bits of the register.

picture5-178 DMAInterrupt Description Table

interrupt event	enable control bit	event flag
More than half of the transmission	HTC_INTEN	HTC_INTST
transfer complete	TC_INTEN	TC_INTST

DMAResource mapping

This chipDMAin total4channel, channel0-aile3. support fromUART0,UART1,UART2,SPI0,SPI1, ADC,TIMER_PLUS0andTIMER_PLUS1requests generated by these eight peripherals.

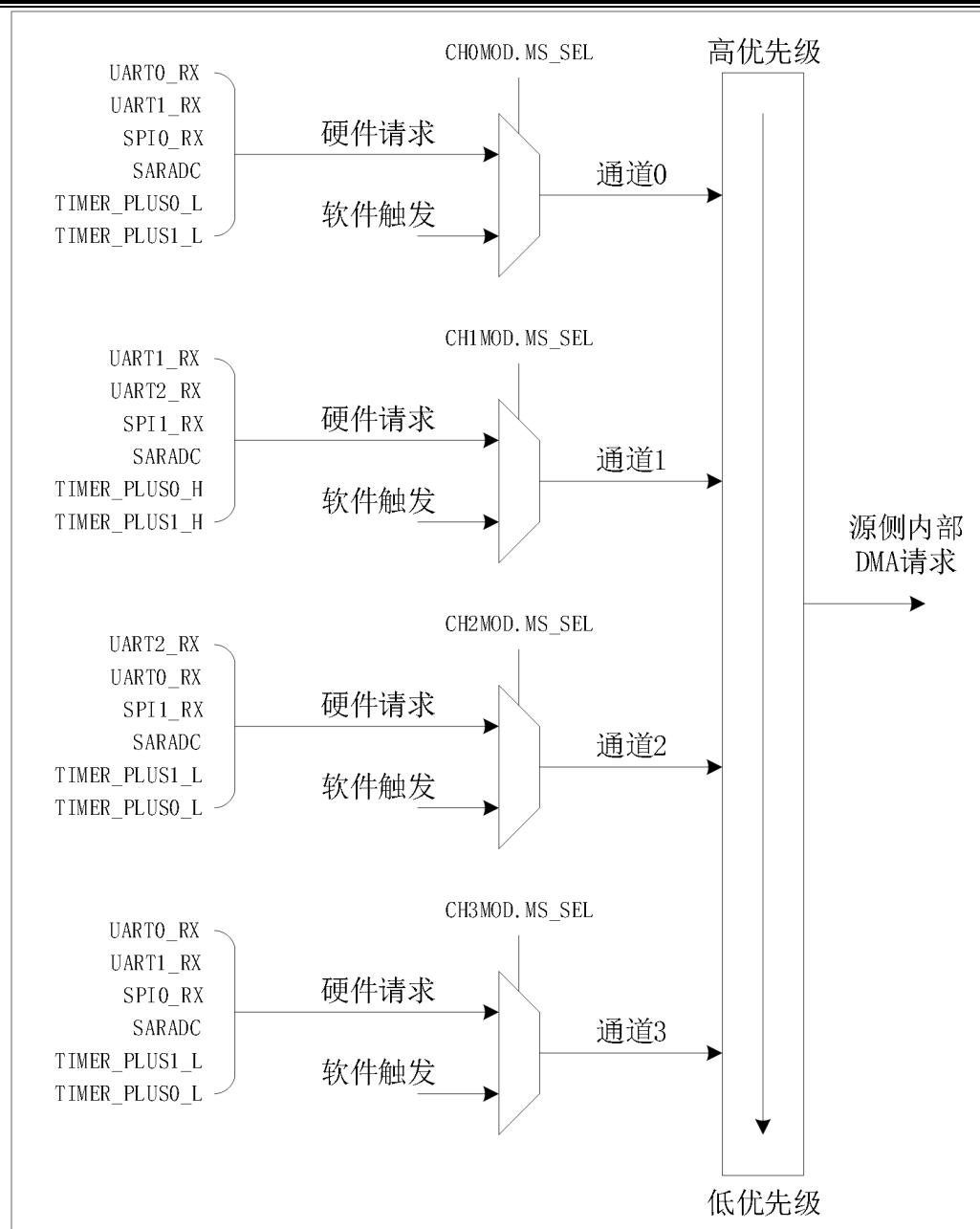
For the peripheral requests corresponding to each channel, please refer to the corresponding relationship shown in the following table and figure.

The corresponding peripheral request mapping relationship of each channel is as follows:

picture5-179 DMAEach channel corresponds to the peripheral request mapping relationship

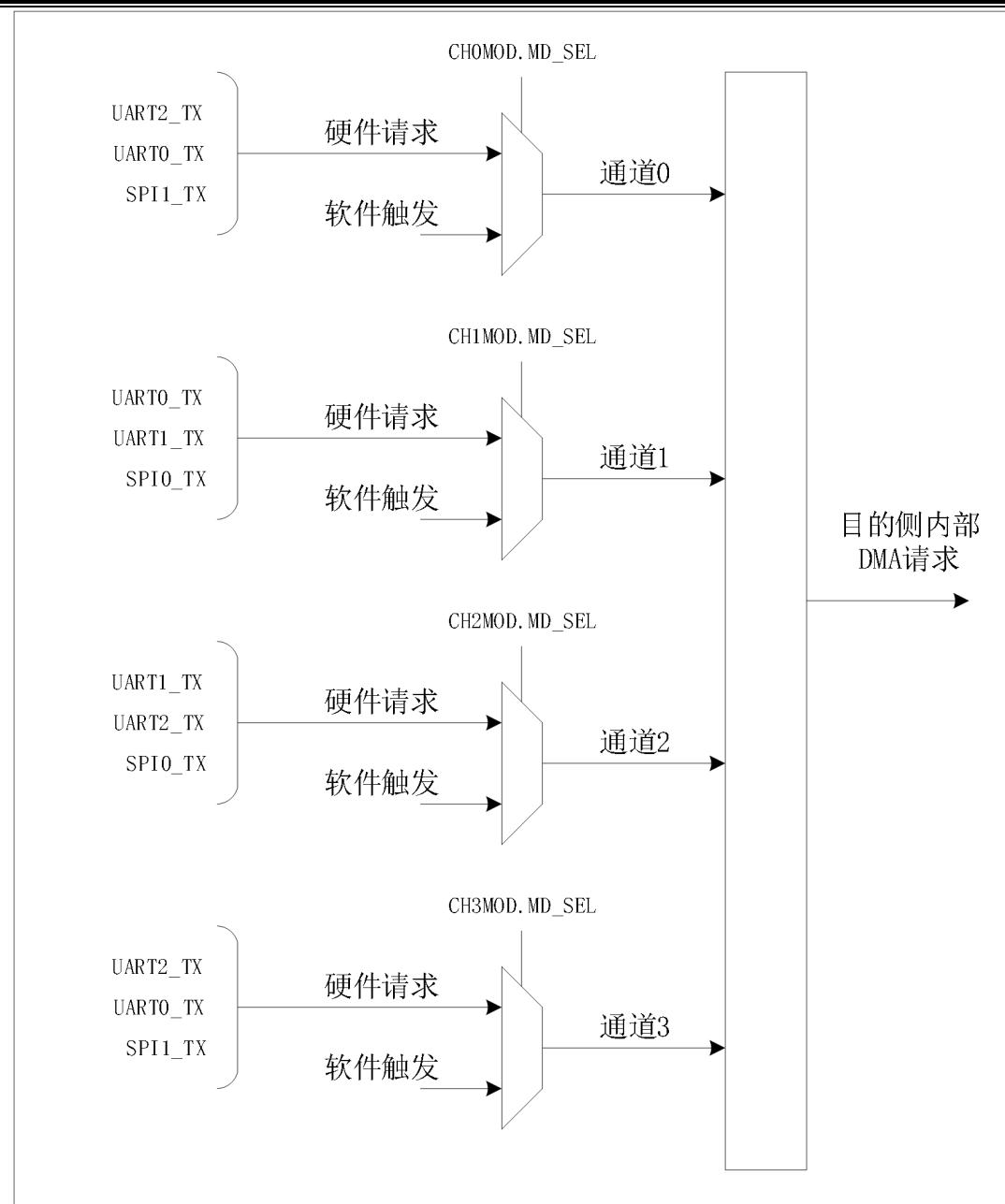
correspondbit	aisle0		aisle1	
	source side	Destination side	source side	Destination side
000	UART0_RX	UART2_TX	UART1_RX	UART0_TX
001	UART1_RX	UART0_TX	UART2_RX	UART1_TX
010	SPI0_RX	SPI1_TX	SPI1_RX	SPI0_TX
011	SARADC	N/A	SARADC	N/A
100	TIMER_PLUS0_L	N/A	TIMER_PLUS0_H	N/A
101	TIMER_PLUS1_L	N/A	TIMER_PLUS1_H	N/A
correspondbit	aisle2		aisle3	
	source side	Destination side	source side	Destination side
000	UART2_RX	UART1_TX	UART0_RX	UART2_TX
001	UART0_RX	UART2_TX	UART1_RX	UART0_TX
010	SPI1_RX	SPI0_TX	SPI0_RX	SPI1_TX
011	SARADC	N/A	SARADC	N/A
100	TIMER_PLUS1_L	N/A	TIMER_PLUS1_H	N/A
101	TIMER_PLUS0_L	N/A	TIMER_PLUS0_H	N/A

The source-side peripheral request mapping diagram is as follows:



picture5-180 DMA Source Side Peripheral Request Mapping Diagram

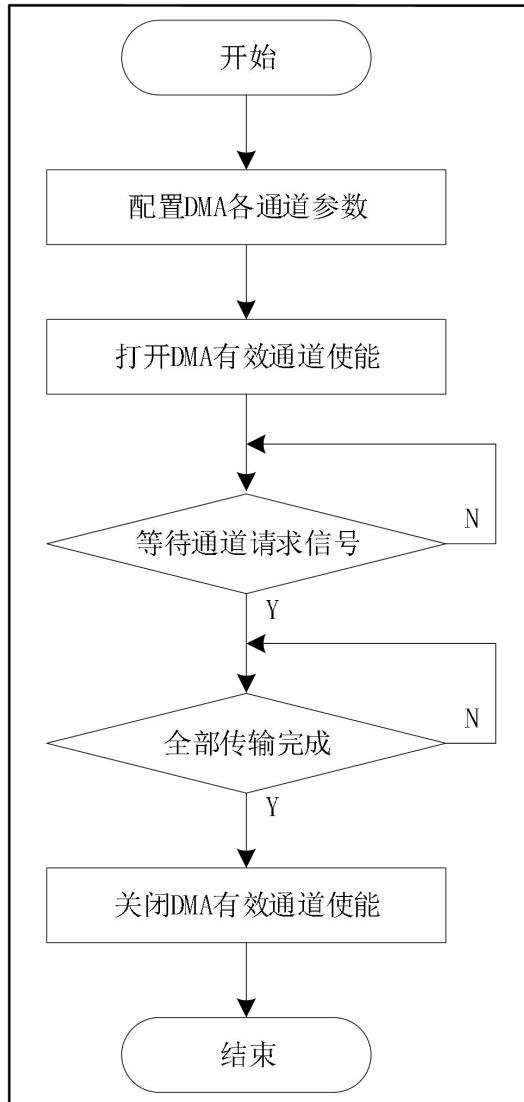
The mapping relationship diagram of the destination peripheral device request is as follows:



picture5-181 DMA Destination Peripheral Request Mapping Diagram

DMA work process

DMAThe workflow is shown in the figure below:



picture5-182 DMA work flow chart

The figure above briefly describes DMA working process. There are mainly the following operations:

1) will register DMA_CON middle DMA_EN bit configured as 1, WillDMAOpen;

2) Turn on the corresponding interrupt enable of each channel;

3) through the registers of each channel DMA_CHnCON, DMA_CHnMOD, DMA_CHnMSADDR and DMA_CHnMADDRProgram the parameter information that each channel needs to configure into the corresponding register bit;

4) to configure the corresponding channel DMA_CHnCON.CH_EN The register is 1, open the channel;

5) waits for a request signal from the source side of the channel.DMAAfter receiving the request signal from the source side, through the bus from the corresponding peripheral set

The source address reads the data and saves it;

6) After reading back the required data from the source side, wait for the request signal from the destination side. whenDMAAfter receiving the request signal from the destination side, pass

Write the processed data to the destination address set by the corresponding peripheral through the bus;

7) According to the configured amount of transmitted data, when half of all data is transmitted or all of it is completely transmitted, a half-transmission will occur

interrupt flag or transfer complete interrupt flag;

8) When all the data has been transferred, the channelCH_ENconfigured as0, to close the channel. At this point the channel can be modified

The value of the configuration register is waiting for the next data transmission to be used.

register map

name	offset	bit width	type	reset value	describe
DMA BASE: 0x40001000					
DMA_CTR	0x00	32	R/W	0x00	DMAcontrol register
DMA_INTEN	0x04	32	R/W	0x00	DMAinterrupt enable register
DMA_INTST	0x08	32	R/W	0x00	DMAInterrupt Status Register
DMA_CH0CTR	0x100	32	R/W	0x1ff	aisle0control register
DMA_CH0MOD	0x104	32	R/W	0x00	aisle0mode register
DMA_CH0MSADDR	0x108	32	R/W	0x00	aisle0source address register
DMA_CH0MDADDR	0x10c	32	R/W	0x00	aisle0Destination Address Register
DMA_CH0_ST	0x110	32	R	0x00	aisle0status register
DMA_CH1CTR	0x120	32	R/W	0x1ff	aisle1control register
DMA_CH1MOD	0x124	32	R/W	0x00	aisle1mode register
DMA_CH1MSADDR	0x128	32	R/W	0x00	aisle1source address register
DMA_CH1MDADDR	0x12c	32	R/W	0x00	aisle1Destination Address Register

DMA_CH1_ST	0x130	32	R	0x00	aisle1status register
DMA_CH2CTR	0x140	32	R/W	0x1ffe	aisle2control register
DMA_CH2MOD	0x144	32	R/W	0x00	aisle2mode register
DMA_CH2MSADDR	0x148	32	R/W	0x00	aisle2source address register
DMA_CH2MDADDR	0x14c	32	R/W	0x00	aisle2Destination Address Register
DMA_CH2_ST	0x150	32	R	0x00	aisle2status register
DMA_CH3CTR	0x160	32	R/W	0x1ffe	aisle3control register
DMA_CH3MOD	0x164	32	R/W	0x00	aisle3mode register
DMA_CH3MSADDR	0x168	32	R/W	0x00	aisle3source address register
DMA_CH3MDADDR	0x16c	32	R/W	0x00	aisle3Destination Address Register
DMA_CH3_ST	0x170	32	R	0x00	aisle3status register

Register description

DMA_CTRregister(0x00)

bit field	name	type	reset value	describe
31: 1	RESERVED	R	0	reserved bit
0	DMA_EN	R/W	0	DMAEnable 0:DMAclosure 1:DMAEnable

DMA_INTEregister(0x04)

bit field	name	type	reset value	describe
31:12	RESERVED	R	0	reserved bit

11	CH3_THC_INTEN	R/W	0	aisle3Transfer Half Done Interrupt Enable Register
10	CH2_THC_INTEN	R/W	0	aisle2Transfer Half Done Interrupt Enable Register
9	CH1_THC_INTEN	R/W	0	aisle1Transfer Half Done Interrupt Enable Register
8	CH0_THC_INTEN	R/W	0	aisle0Transfer Half Done Interrupt Enable Register
7:4	RESERVED	R	0	reserved bit
3	CH3_TC_INTEN	R/W	0	aisle3Transfer Complete Interrupt Enable Register
2	CH2_TC_INTEN	R/W	0	aisle2Transfer Complete Interrupt Enable Register
1	CH1_TC_INTEN	R/W	0	aisle1Transfer Complete Interrupt Enable Register
0	CH0_TC_INTEN	R/W	0	aisle0Transfer Complete Interrupt Enable Register

DMA_INTSTregister(0x08)

bit field	name	type	reset value	describe
31:12	RESERVED	R	0	reserved bit
11	CH3_THC_INTST	R/W	0	aisle3Transfer Half Done Interrupt Status Register Write1cleared.
10	CH2_THC_INTST	R/W	0	aisle2Transfer Half Done Interrupt Status Register Write1cleared.
9	CH1_THC_INTST	R/W	0	aisle1Transfer Half Done Interrupt Status Register Write1cleared.
8	CH0_THC_INTST	R/W	0	aisle0Transfer Half Done Interrupt Status Register Write1cleared.
7:4	RESERVED	R	0	reserved bit
3	CH3_TC_INTST	R/W	0	aisle3Transfer Complete Interrupt Status Register Write1cleared.
2	CH2_TC_INTST	R/W	0	aisle2Transfer Complete Interrupt Status Register Write1cleared.

1	CH1_TC_INTST	R/W	0	aisle1Transfer Complete Interrupt Status Register Write1cleared.
0	CH0_TC_INTST	R/W	0	aisle0Transfer Complete Interrupt Status Register Write1cleared.

DMA_CHnCTRregister(0x100 + 0x20*(n))

bit field	name	type	reset value	describe
31:17	RESERVED	R	0	reserved bit
16	SWREQ	R/W	0	<p>Request this channel to start transmission through software Write1start transfer</p> <p>Note1: Only used for requests initiated when the source address side is configured as a storage device. Note2:LOOPIf configured as0When , a transmission is completed, and the hardware is automatically cleared;LOOPIf configured as1, it is cleared by software control.</p>
15:14	PRI	R/W	0	<p>channel priority 00:Low 01:middle 10:high 11:Highest</p> <p>Note: If different channels are configured with the same priority, the channel with the smaller has higher priority.</p>
13	LOOP	R/W	0	<p>Cycle Mode Control Bits 0: Do not execute the loop mode. Stop when the transfer is complete 1: Execute loop mode. After the transmission is completed, it will automatically reset according to the original configuration. new transmission</p>
12:1	LENTH	R/W	0xffff	<p>The transmission count register indicates the number of requests transmitted by this channel. The actual number of transfers is (LENTH+1)Second-rate. The data unit of each transmission is: the data unit of the source address side is composed of MS_SIZEdecision; the data unit on the destination address side is determined byMD_SIZEDecide.</p>

0	CH_EN	R/W	0	<p>Channel Enable Control</p> <p>0: channel closed</p> <p>1: The channel is valid</p> <p>Note1: Channel enable is written by software1, to start the transfer.LOOPIf match Set as0When , a transmission is completed, and the hardware is automatically cleared;LOOP If configured as1, it is cleared by software control.</p> <p>Note2: All configurations need to be performed with the enable disabled. Should Bit is1, the related configuration cannot be changed.</p>
---	-------	-----	---	---

DMA_CHnMODregister(0x100 + 0x20*(n) + 0x04)

bit field	name	type	reset value	describe
31:14	RESERVED	R	0	reserved bit
13:11	MD_SEL	R/W	0	<p>MDSide Peripheral Selection</p> <p>000: select storage peripheral</p> <p>001: selection yieldshsreq_md[0]Corresponding peripherals</p> <p>010: selection yieldshsreq_md[1]Corresponding peripherals</p> <p>011: selection yieldshsreq_md[2]Corresponding peripherals</p> <p>100: selection yieldshsreq_md[3]Corresponding peripherals</p> <p>101: selection yieldshsreq_md[4]Corresponding peripherals</p> <p>110: selection yieldshsreq_md[5]Corresponding peripherals</p> <p>111: selection yieldshsreq_md[6]Corresponding peripherals</p> <p>Note1: configured as000When , it means that the storage peripheral is selected, usually means on-chipSRAM. Can also be used for no-handshake no-wait reachable other peripherals.</p> <p>Note2: configured as001-111When different chips are connected to different peripherals, The corresponding peripherals need to be determined according to the actual peripherals connected to the specific chip.</p>



10:9	MD_SIZE	R/W	0	MDside bus transfer width 00:8bits 01:16bits 10:32bits 11:reserve
8	MD_ADDMO D.	R/W	0	MDside address change mode selection 0: The address remains unchanged 1: address increment
7:6	RESERVED	R	0	reserved bit
5:3	MS_SEL	R/W	0	MSSide Peripheral Selection 000: select storage peripheral 001: selection yieldshsreq_ms[0]Corresponding peripherals 010: selection yieldshsreq_ms[1]Corresponding peripherals 011: selection yieldshsreq_ms[2]Corresponding peripherals 100: selection yieldshsreq_ms[3]Corresponding peripherals 101: selection yieldshsreq_ms[4]Corresponding peripherals 110: selection yieldshsreq_ms[5]Corresponding peripherals 111: selection yieldshsreq_ms[6]Corresponding peripherals Note1: configured as000When , it means that the storage peripheral is selected, usually means on-chipSRAM. Can also be used for no-handshake no-wait reachable other peripherals. Note2: configured as001-111When different chips are connected to different peripherals, The corresponding peripherals need to be determined according to the actual peripherals connected to the specific chip.
2:1	MS_SIZE	R/W	0	MSSide bus transfer width 00:8bits 01:16bits 10:32bits 11:reserve

0	MS_ADDMO D.	R/W	0	MSSide address change mode selection 0: The address remains unchanged 1: address increment
---	----------------	-----	---	--

DMA_CHnMSADDRregister(0x100 + 0x20*(n) + 0x08)

bit field	name	type	reset value	describe
31:0	MS_ADDR	R/W	0	MSSide address (source address) 32Word alignment for bit operations;16Bit operations are halfword aligned.8 For bit operations, byte-aligned.

DMA_CHnMDADDRregister(0x100 + 0x20*(n) + 0x0C)

bit field	name	type	reset value	describe
31:0	MD_ADDR	R/W	0	MDSide address (destination address) 32Word alignment for bit operations;16Bit operations are halfword aligned.8 For bit operations, byte-aligned.

DMA_CHn_STregister(0x100 + 0x20*(n) + 0x10)

bit field	name	type	reset value	describe
31:12	RESERVED	R	0	reserved bit
11:0	CUR_LENGTH	R	0	The current number of transfers

5.25 AES arithmetic unit (AES)

5.25.1 overview

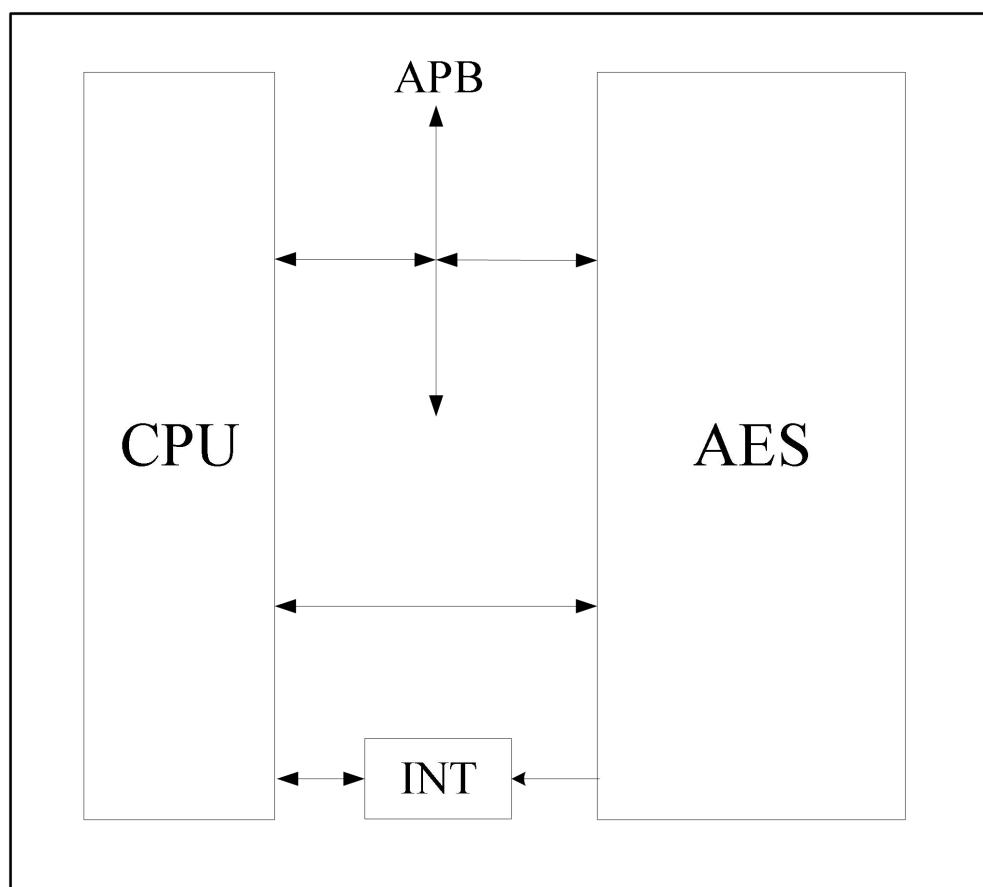
This chip has a AES module, support 128bit key and initial vector can be used to use AES algorithm for data row encryption and decryption.

use 128bit key length pair 128Bit blocks are encrypted and decrypted. It can also perform key derivation, encryption or decryption

The encryption key is stored in internal registers to minimize CPU

write operation.

By default, the electronic codebook mode is selected (ECB), the hardware also supports cipher block chaining (CBC) or counter (CTR mode) chaining algorithm.

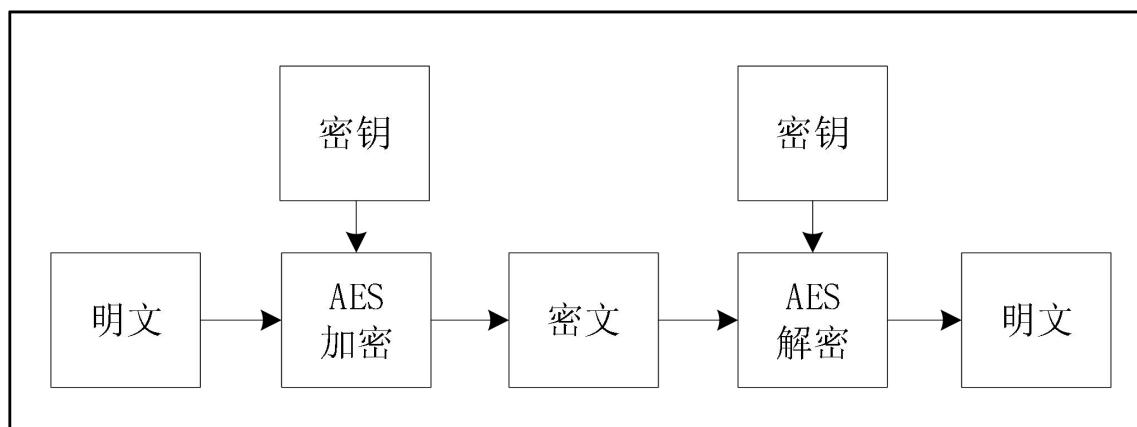


picture5-183 AES System Block Diagram

5.25.2 characteristic

- useAES RijndaelBlock cipher algorithm for encryption/decryption
- internal128bit register for storing encrypted or derived keys (4individual32bit register)
- Support electronic codebook (ECB), cipher block chaining (CBC) and counter mode (CTR)
- Derived key derivation for decryption
- 128bit block processing
- 128bit key length
- 213clock cycles to encrypt or decrypt a128Bitblock (includes input and output stages)
- 1individual32bitINPUTdatabufferand1individual32bitOUTPUTdatabuffer
- only support32Bit Data Width Register Access
- one128bit register for theCBCmode configurationAESwhen used to initialize the vector, or when selectingCTRmold for32bit counter initialization

5.25.3 Block Diagram of Module Structure



picture5-184 AESBlock Diagram of Module Structure

Pictured above isAESThe structural block diagram of the module, the figure only briefly describes the process of encryption and decryption, each of the modules in this module

The encryption and decryption processes and schematic diagrams of different algorithms in this mode will be described in detail in the function description chapter.

5.25.4Functional description

AESThe hardware supports three algorithms, when disabledAES(bitEN = 0), you can passAES_CRin the registerCHMOD [1:0]bits to select:

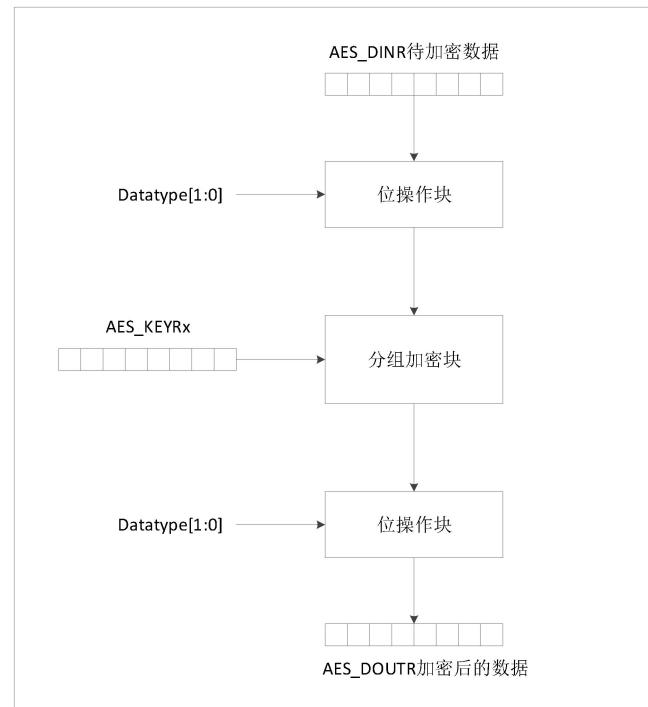
- Electronic codebook (ECB)
- Cipher block chaining (CBC)
- Counter mode (CTR)

Electronic Codebook (ECB)

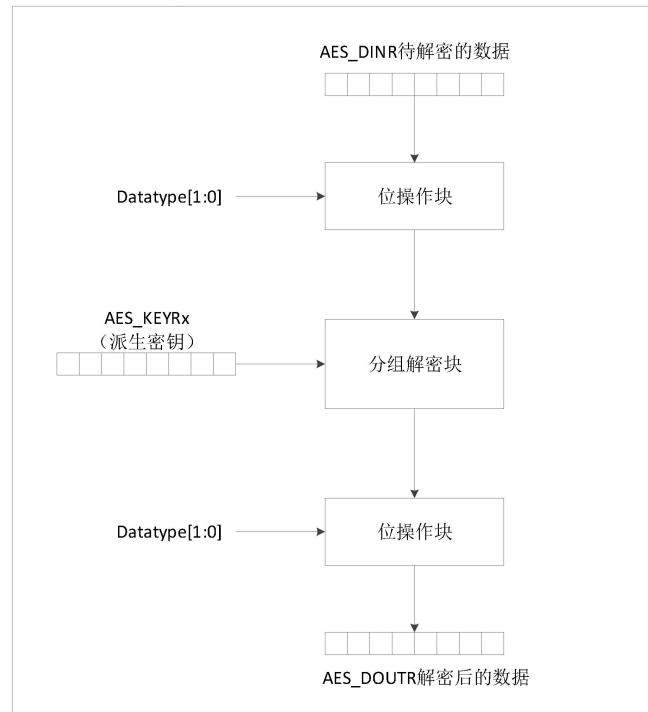
This is the default mode. Not used in this modeAES_IVRegister. There is no link operation. The data is divided into blocks,

Each block is encrypted separately.

The schematic diagram of the encryption and decryption of the electronic codebook algorithm is as follows:



picture5-185 AESEncryption schematic diagram of electronic codebook algorithm



picture5-186 AESDecryption schematic diagram of the electronic codebook algorithm

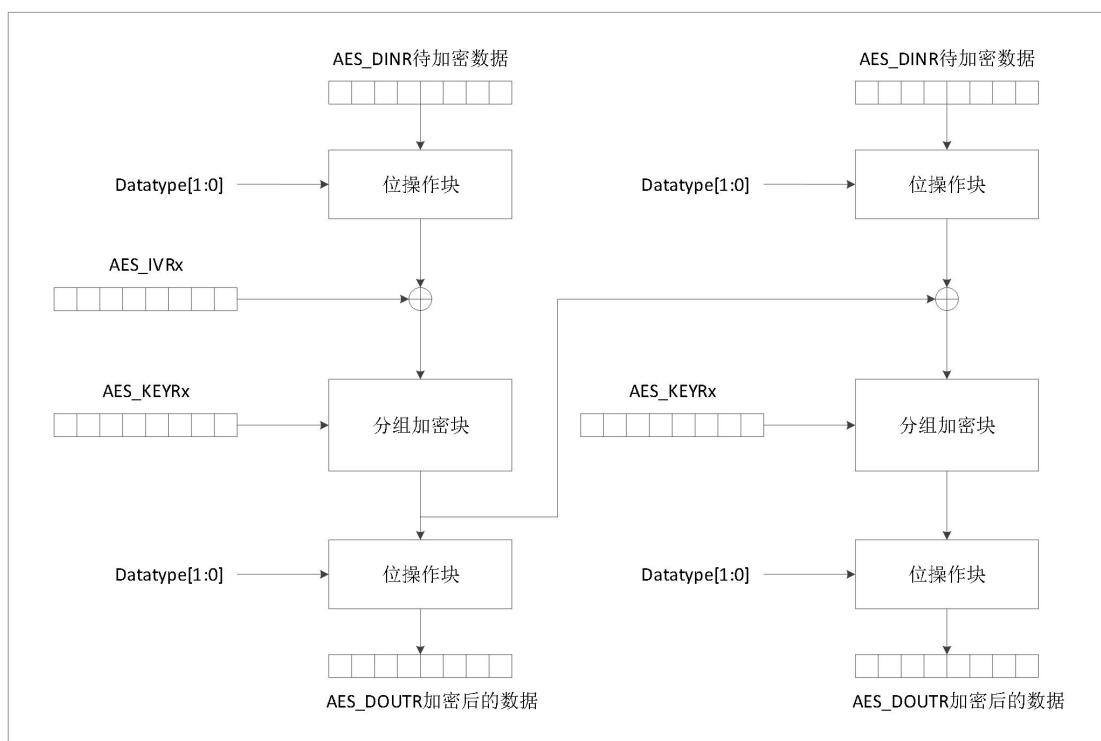
Cipher block chaining (CBC)

In the cipher block chaining (CBC) mode, each data to be encrypted is XORed with the previous encrypted ciphertext before encryption

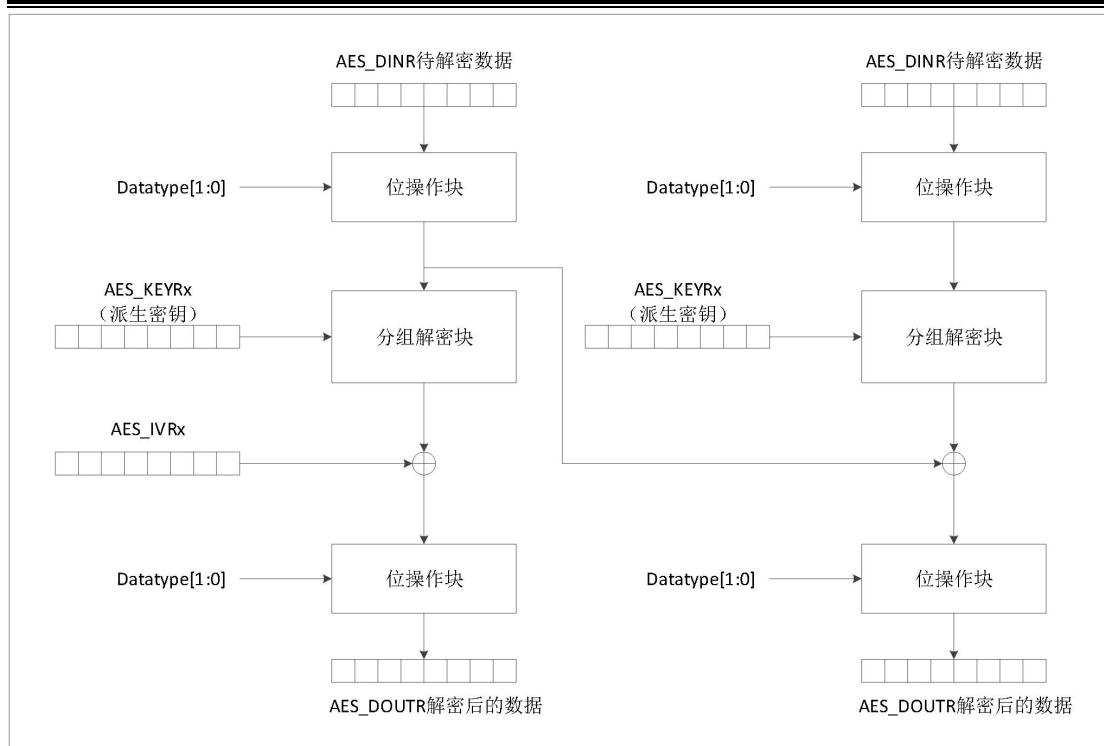
or. To make each message unique, an initialization vector (AES_IVRx).

in encrypted modeIVParticipate in XOR before encryption block, in decryption modeIVParticipates in the XOR after decrypting the block.

The schematic diagram of the encryption and decryption of the cipher block chaining algorithm is as follows:



picture5-187 AESEncryption schematic diagram of the cipher block chaining algorithm



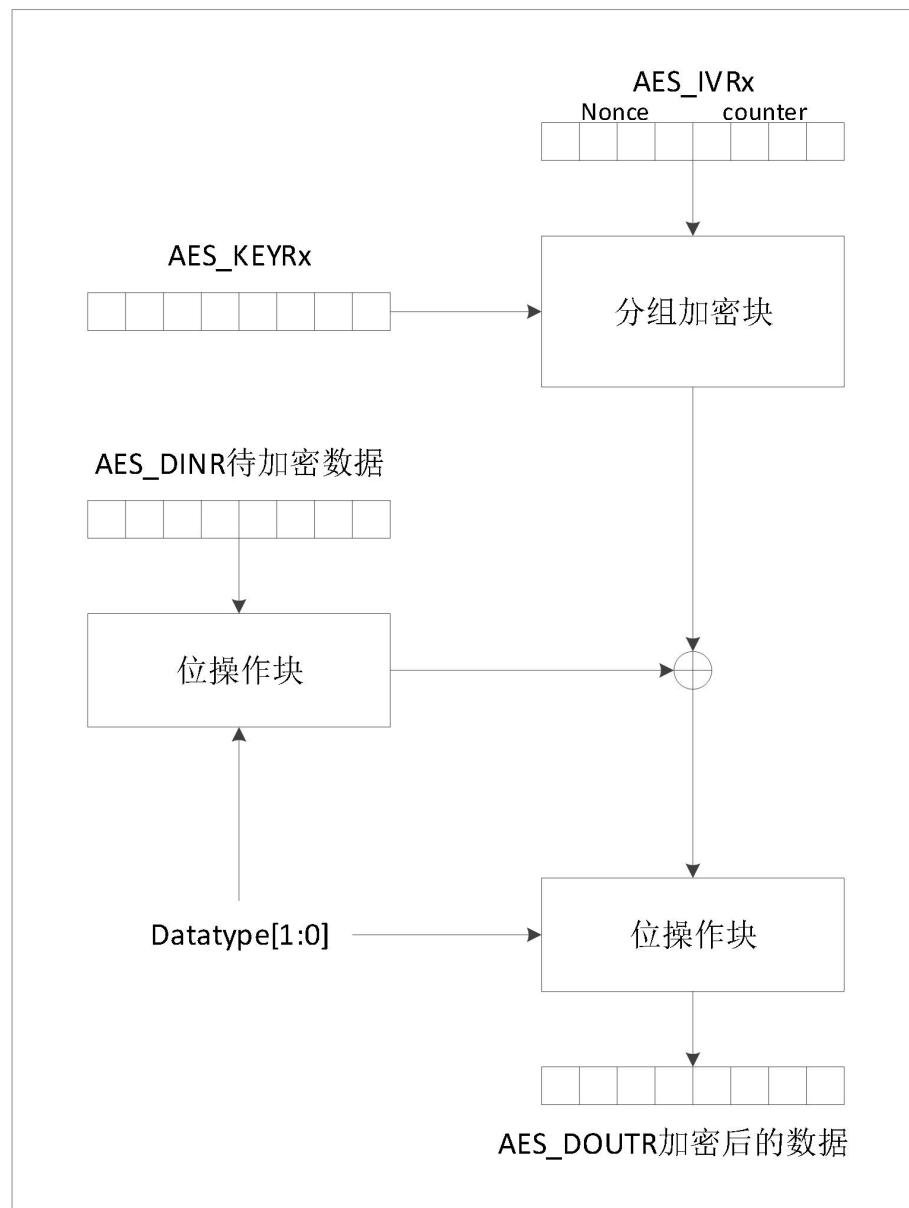
picture5-188 AES schematic diagram of the decryption of the cipher block chaining algorithm

counter mode (CTR)

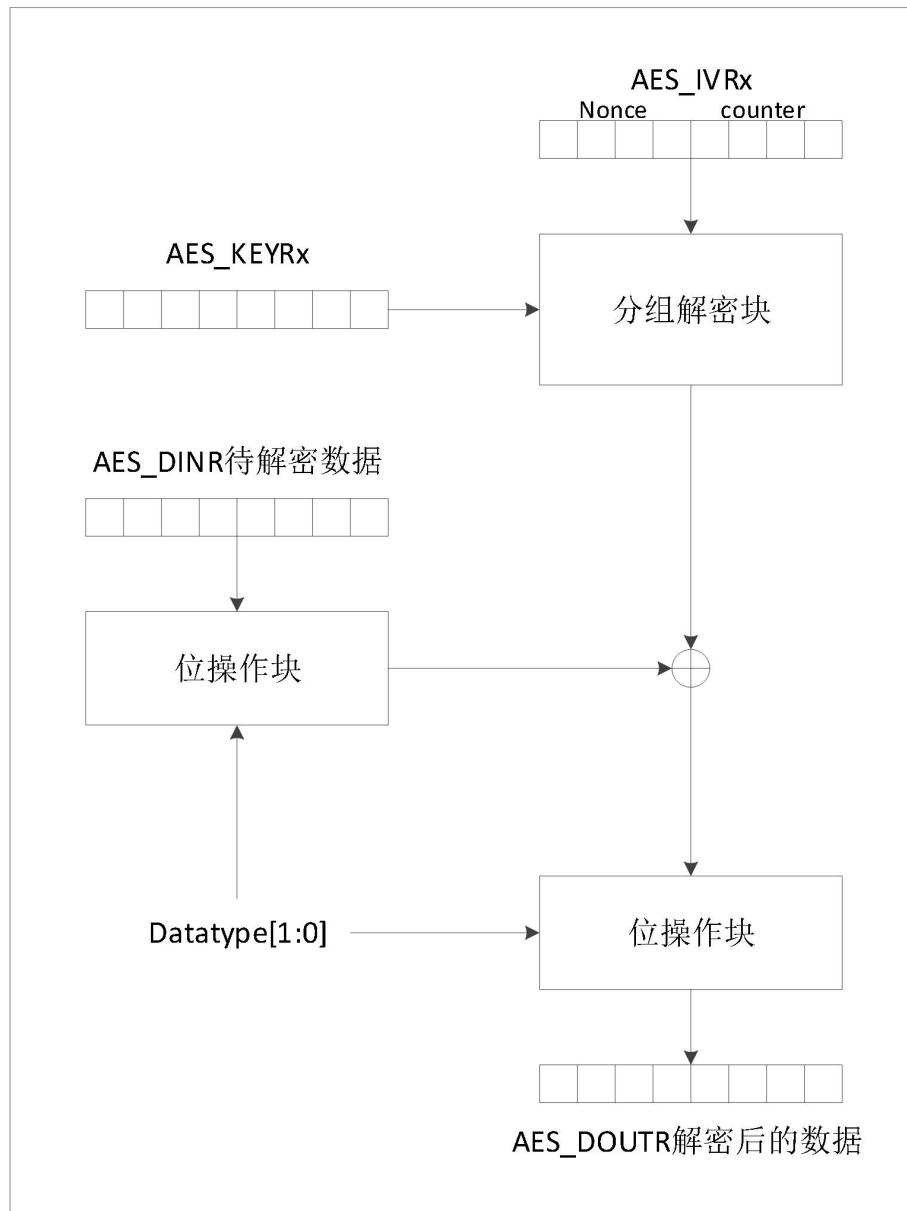
In counter mode, in addition to the random value of the 32-bit counter is used with the ciphertext or the number to be encrypted

According to conduct XOR operation.

The schematic diagram of the encryption and decryption of the counter mode chaining algorithm is as follows:



picture5-189 AESEncryption schematic diagram of the counter mode chaining algorithm



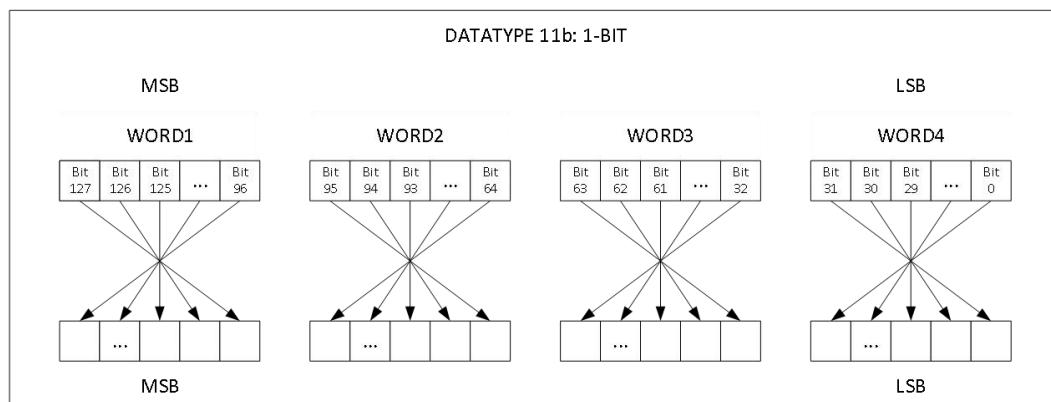
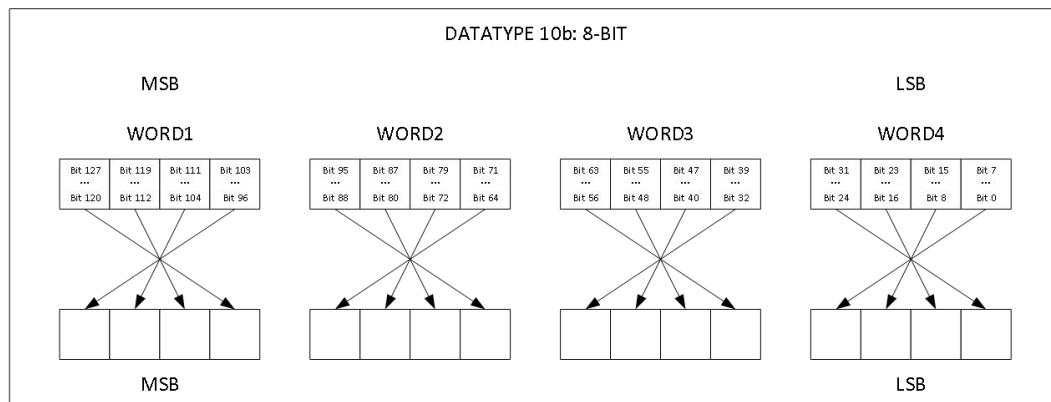
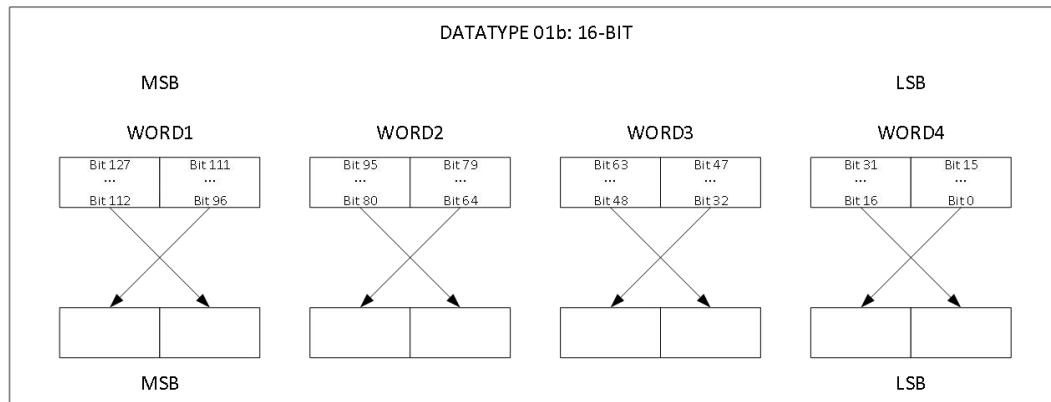
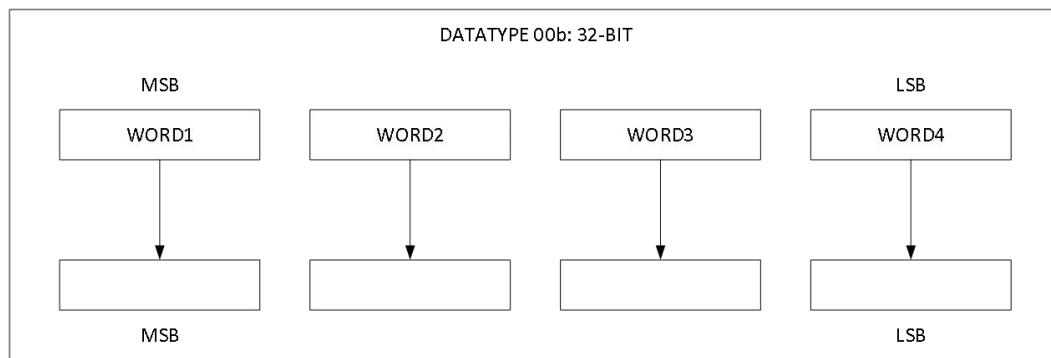
picture5-190 AESDecryption schematic of the counter mode chaining algorithm

data schema

AESFour data formats are provided to perform shift operations on input and output data. For specific shift operations, please refer to the above chapter

The algorithm description of each module in the section can be modified by AES_CRregisterDATATYPEbit is set.

The specific displacement principle is as follows:



picture5-191 AESDisplacement schematic

Operating procedures

encryption

Encryption meansAESUsing the key register (KEY) and the initial vector register (IV) to the original data in different modes

The process of performing encryption processing to obtain a series of ciphertexts. The encryption process is as follows:

1.set upAES_CR·EN=0disabledAES;

2.set upAES_CR·MODE[1:0]=00Configured as encrypted mode, setAES_CR·CHMOD[1:0]configuration chain

connection mode;

3. ECBmode configurationAES_KEYRxregister,CTRorCBCmode also configureAES_IVRxregister;

4.set upAES_CR·EN=1turn onAES;

5.Write the plaintext data to be encrypted in four consecutive timesAES_DINRregister (write firstMSB) ;

6.waitAES_SRin registerCCFflag generation;

7.continuous readAES_DOUTRRegister four times to get the encrypted ciphertext data (read out firstMSB);

8.repeat5,6,7Complete encryption of all data.

Note: the encryption usedKEYInitial key for data encryption;

when encryptingKEYThe register does not change;

CTRWhen the schema is encryptedIVwill change;

key derivation

key derivation meansAESBy utilizing the key register (KEY) for key derivation processing in different modes, thus

The process of getting a sequence of derived keys. The key derivation process is as follows:

1.set upAES_CR·EN=0disabledAES;

2.set upAES_CR·MODE[1:0]=01Configured as the key derivation mode, no need to set when deriving the key

AES_CR·CHMOD[1:0], because the key derivation mode is independent of the selected chaining algorithm;

3.configurationAES_KEYRxregister, configurationAES_IVRxRegister is invalid;

4.set upAES_CR·EN=1turn onAES;

5.waitAES_SRin registerCCFflag generation;

6.The key derived at this point is stored in theAES_KEYRxregister, if necessary please read theAES_KEYRxregister to obtain the derived key;

7.If you need to restart the key derivation operation, repeat the steps3,4,5,6.

Note: Key derivation has nothing to do with the chaining algorithm, only with the original keyKEYrelated.

decrypt

Decryption meansAESBy utilizing the key register (KEY) and the initial vector register (IV) in different modes against the input

The process of decrypting the input ciphertext data to obtain a series of plaintext data. The decryption process is as follows:

1.set upAES_CR·EN=0disabledAES;

2.set upAES_CR·MODE[1:0]=10Configured as decryption mode, setAES_CR·CHMOD[1:0]configuration chain connection mode;

3. ECBmode configurationAES_KEYRxregister,CTRorCBCmode also configureAES_IVRxregister;

4.set upAES_CR·EN=1turn onAES;

5.Write the ciphertext data to be decrypted in four consecutive timesAES_DINRregister (write firstMSB) ;

6.waitAES_SRin registerCCFflag generation;

7.continuous readAES_DOUTRRegister four times to get the decrypted plaintext data (read out firstMSB);

8.repeat5,6,7Complete the decryption of all data.

Note: used in decryption modeKEYTo derive the key, the derived key can be generated through the key derivation mode, and then for decryption mode;

CTRin modeKEYfor initialKEY,CTRin decrypted modeIVfor initialIV.

Key derivation + decryption

Derived key + decryption meansAESThe key derivation and decryption process is carried out successively, so the key register used here (KEY) is the initial key. The process is as follows:

1.set upAES_CR·EN=0disabledAES;

2.set upAES_CR·MODE[1:0]=11Configured as key derivation + decryption mode, setAES_CR·CHMOD[1:0]

Configure link mode (CTRThe mode does not support the key derivation + decryption mode. If this mode is configured, the hardware will forcefully return the decryption mode.

password mode);

3. ECBmode configurationAES_KEYRxregister,CTRorCBCmode also configureAES_IVRxregister;

4.set upAES_CR·EN=1turn onAES;

5.Write the ciphertext data to be decrypted in four consecutive timesAES_DINRregister (write firstMSB) ;

6.waitAES_SRin registerCCFflag generation;

7.continuous readAES_DOUTRRegister four times to get the decrypted plaintext data (read out firstMSB);

8.repeat5,6,7Complete the decryption of all data.

NOTE: This mode usesKEYis the original key,AESThe key will be derived during the working process, and the derived key key stored inAES_KEYin the register;

NoticeCTRThis type of operation is not supported in the mode, and the forced configuration will be returned to the decryption mode by the hardware;

register map

name	Offset	bit width	type	reset value	describe
AES	BASE: 0x400BD000				
AES_CR	0x00	32	R/W	0x00	AEScontrol register
AES_SR	0x04	32	R	0x00	AESstatus register
AES_DINR	0x08	32	R/W	0x00	AESinput data register
AES_DOUTR	0x0c	32	R	0x00	AESoutput data register
AES_KEYR0	0x10	32	R/W	0x00	AESkey register0
AES_KEYR1	0x14	32	R/W	0x00	AESkey register1
AES_KEYR2	0x18	32	R/W	0x00	AESkey register2
AES_KEYR3	0x1C	32	R/W	0x00	AESkey register3
AES_IVR0	0x20	32	R/W	0x00	AESEncryption starting point register0
AES_IVR1	0x24	32	R/W	0x00	AESEncryption starting point register1
AES_IVR2	0x28	32	R/W	0x00	AESEncryption starting point register2
AES_IVR3	0x2C	32	R/W	0x00	AESEncryption starting point register3

Register description

AES_CRregister(0x00)

bit field	name	type	reset value	describe
31:9	RESERVED	R	0	reserved bit
8	ERRC	R/W	0	<p>error flag clear</p> <p>Will1Writing this bit will clear theRDERRandWRERRsign.</p> <p>This bit is read low.</p>

7	CCFC	W	0	Computation complete flag is cleared. This bit is read low.
6:5	CHMOD	R/W	0	AESLink mode selection. 00: electronic codebook (ECB) 01: cipher block chaining (CBC) 10: Counter mode (CTR)) 11:reserve. Only when disabledAEScan only be changed whenAESlink mode. disabled when enabledAESwrite these bits when writing to avoid unpredictableAES Behavior.
4:3	MODE	R/W	0	AESMode selection. 00: Mode 1: Encryption 01: Mode 2: Key Derivation 10: Mode 3: Decryption 11: Mode 4: Key derivation + decryption only disabledAESto change the operating mode. disabled when enabledAESwrite these bits when writing to avoid unpredictableAES Behavior. if chooseCTRmode, disable mode4. If the software attempts to CTRmode configuration mode4, it will be forced into the mode3.
2:1	DATATYPE	R/W	0	Data type selection. 00:32Bit data, no bit swapping. 01:16bit data or halfword, perform halfword swap. For example, raw data 0x764356AB, the value given to the encrypted block is0x56AB7643 10:8Bits of data or bytes, all bytes are swapped. For example, the raw data is 0x764356AB, the value given to the encrypted block is0xAB564376. 11:1bit data. All bits are swapped. For example, the original data is 0x00112233, the value given to the encrypted block is0xCC448800only disabledAES to change the operating mode. disabled when enabledAESwhen writing these bits to avoid some unpredictable AESBehavior.
0	EN	R/W	0	AESenable signal. AESIt can be initialized at any time by resetting this bit: whenENafter settingAESReady to start processing new blocks. whenAESin mode2(Key Derivation) This bit is automatically cleared by hardware when the operation is complete

AES_SRregister(0x04)

bit field	name	type	reset value	describe
31:3	RESERVED	R	0	reserved bit.
2	WRERR	R	0	<p>Write error flag. when detected fromAES_DINRThis bit is set by hardware on an accidental read operation of a register (during calculation or data input phase). software through theAES_CRset in the registerERRCbit to clear. 0: No write errors detected 1: Write error detected This sign is rightAEShas no effect, even if aWRERRinterruption, AESwill continue to run.</p>
1	RDERR	R	0	<p>Read error flags. when detected fromAES_DOUTRThis bit is set by hardware on an accidental read operation of a register (during calculation or data input phase). software through theAES_CRset in the registerERRCbit to clear. 0: No read errors detected 1: Read error detected This sign is rightAEShas no effect, even if aRDERRinterruption, AESwill continue to run.</p>
0	CCF	R	0	<p>Calculation complete flag. ifCCFIEpreviously inAES_CRset in the register. software through theAES_CRset in the registerCCFcbit to clear. 1: calculation complete 0: Computation not complete</p>

AES_DINRregister(0x08)

bit field	name	type	Reset value	description

31:0	DINR	R/W	0	<p>Input data register.</p> <p>During the input phase, this register must be written to4Second-rate:</p> <ul style="list-style-type: none"> - in mode1(encryption), you must write4bytes, indicating from MSBarriveLSBThe data. - in mode2(key derivation), this register is not used because this mode only involvesAES_KEYRxRegister starts the key derivation calculation. - in mode3(decrypted) and mode4(key derivation + decryption), must write4bytes representing the ciphertextMSBarriveLSB. <p>NOTE: This register must end with32Bit data width access.</p>
------	------	-----	---	--

AES_DOUTRregister(0x0C)

bit field	name	type	reset value	describe
31:0	DOUTR	R	0	<p>data output register</p> <p>This register is read-only.</p> <p>once producedCCFflag (computation complete flag) , read the data register4times to access128bit output result:</p> <ul style="list-style-type: none"> - in mode1(encrypted), read the4bytes fromMSB arrive LSBencrypted data. - in mode2(key derivation), there is no need to read this register because the derivation key is located inAES_KEYRxregister. - in mode3(decrypted) and mode4(key derivation + decryption), read the4word means fromMSBarriveLSBof plain text. <p>NOTE: This register must end with32Bit data width access.</p>

AES_KEYR0register(0x10)

bit field	name	type	Reset value	description

31:0	KEYR	R/W	0	<p>key register[31:0]</p> <p>must be openedAESWrite to this register before enabling:</p> <p>in mode1(encrypted), mode2(key derivation) and mode4(key derivation + decryption), the value to be written represents fromLSBThe encryption key of , which means that the key [31:0].</p> <p>in mode3(decryption), the value to be written represents the value fromLSB decryption key for [31:0]. When the register is written with the encryption key in this decryption mode, when theAESReading it before will return the encrypted value. set upCCFReading it after the flag will return the derived key.</p> <p>enableAESReading this register will return an unpredictable value.</p> <p>NOTE: This register does not contain the mode4The derived key in (derived key + decryption). It always contains the encryption key value.</p>
------	------	-----	---	---

AES_KEYR1register(0x14)

bit field	name	type	reset value	describe
31:0	KEYR	R/W	0	key register[63:32]

AES_KEYR2register(0x18)

bit field	name	type	reset value	describe
31:0	KEYR	R/W	0	key register[95:64]

AES_KEYR3register(0x1C)

bit field	name	type	reset value	describe
31:0	KEYR	R/W	0	key register[127:96]

AES_IVR0register(0x20)

bit field	name	type	reset value	describe
31:0	IVR	R/W	0	<p>Initialize vector registers [31:0]</p> <p>gotta beAES_CRregisterENLocation1Before writing to this register:</p> <p>Register values have no meaning in the following cases:</p> <ul style="list-style-type: none"> -ECBmode (electronic codebook). - CTRorCBCkey derivation in mode. <p>existCTRmode (counter mode), this register contains the32bit counter value.</p> <p>in enableAESI simultaneous read of this register will return the value 0x00000000.</p>

AES_IVR1register(0x24)

bit field	name	type	reset value	describe
31:0	IVR	R/W	0	<p>Initialize vector registers [63:32]</p> <p>existCTRmode (counter mode), this register contains a random value.</p> <p>in enableAESI simultaneous read of this register will return the value 0x00000000.</p>

AES_IVR2register(0x28)

bit field	name	type	reset value	describe
31:0	IVR	R/W	0	<p>Initialize vector registers [95:64]</p> <p>existCTRmode (counter mode), this register contains a random value.</p> <p>in enableAESI simultaneous read of this register will return the value 0x00000000.</p>



AES_IVR3register(0x2C)

bit field	name	type	reset value	describe
31:0	IVR	R/W	0	<p>Initialize vector registers (MSB IVR [127:96])</p> <p>existCTRmode (counter mode), this register contains a random value.</p> <p>in enableAESI simultaneous read of this register will return the value 0x00000000.</p>

6.electrical characteristics

This chapter describes the electrical parameters of the chip, including operating voltage, operating temperature, power consumption, analog characteristic parameters and IO characteristic parameters, etc.

6.1Test Conditions

Unless otherwise specified, all voltages are 3.3V as the benchmark. Unless otherwise specified, maximum and minimum parameters are at ambient temperature Spend $T_A=25^{\circ}\text{C}$, $V_{DD}=3.3\text{V}$ tested below.

6.2Absolute Maximum Ratings

6.2.1Voltage characteristics

surface1 Voltage characteristics					
symbol	describe	minimum value	typical value	maximum value	unit
$V_{DD}-V_{SS}$	External mains supply voltage	- 0.3		3.6	V
V_{IN}	The input voltage of the pin	VSS-0.3		3.6	V
$-V_{DD}$	Voltage difference between different supply pins			50	mV
$-V_{SS}$	The voltage difference between different ground pins			50	mV

6.2.2Current characteristics

surface2 Current characteristics					
symbol	describe	minimum value	typical value	maximum value	unit
I_{DD}	The total current through the power line			120	mA
I_{out}	Output current on a single pin	- 30		30	mA
I_{IN}	Injection current on a single pin	- 5		5	mA
$-I_{IN}$	Total injected current on all pins	- 25		25	mA

6.2.3temperature characteristics

surface3temperature characteristics

symbol	describe	minimum value	typical value	maximum value	unit
T_{STG}	storage temperature range	- 45		150	°C
T_J	maximum junction temperature			125	°C

6.3working conditions

6.3.1general working conditions

surface4Shared Working Conditions Sheet

symbol	describe	minimum value	typical value	maximum value	unit
V_{DD}	Standard working voltage	2.0	3.3	3.6	V
f_{CLK}	system clock frequency	-	-	72	MHz
T_A	Operating temperature	- 40	-	85	°C

6.3.2Power-up and power-down operating conditions

surface5Power-up and power-down operating conditions

symbol	describe	minimum value	typical value	maximum value	unit
T_{VR}	VDDrate of ascent	0.3	-	∞	uS/V
T_{VF}	VDDrate of descent	0.3	-	∞	uS/V

6.3.3Reset and Power Control Block Features

Surface6Reset and Power Control Block Features

symbol	describe	minimum value	typical value	maximum value	unit
V_{POR}	Power-on reset threshold	1.87	1.9	1.96	V
V_{PDR}	Brown-out Reset Threshold	1.76	1.8	1.85	V

V_{HYS}	PDRhysteresis	90	100	110	mV
T_{RST}	reset duration	-	0.6	-	ms

Note: This is the theoretical value of circuit simulation.

6.3.4 Supply Current Characteristics

surface7Supply Current Characteristics

symbol	describe	condition	minimum value	typical value	maximum value	unit
I_{DD}	Supply current in run mode	3.3V@48MHz Run in RAM All Peripherals clock OFF				mA
I_{DD}	Supply current in run mode	3.3V@24MHz Run in RAM All Peripherals clock OFF				mA
I_{DD}	Supply current in run mode	3.3V@12MHz Run in RAM All Peripherals clock OFF				-A
I_{DD}	Supply current in run mode	3.3V@32KHz Run in RAM All Peripherals clock OFF				-A
I_{DD}	Supply current in run mode	3.3V@48MHz All Peripherals clock ON				mA
I_{DD}	Supply current in run mode	3.3V@24MHz All Peripherals clock ON				mA
I_{DD}	Supply current in run mode	3.3V@12MHz All Peripherals clock ON				-A
I_{DD}	Supply current in run mode	3.3V@48MHz All Peripherals clock OFF				mA
I_{DD}	Supply current in run mode	3.3V@24MHz All Peripherals clock OFF				mA
I_{DD}	Supply current in run mode	3.3V@12MHz All Peripherals clock OFF				-A
I_{DD}	sleepsupply current in mode	3.3V@48MHz	-	12	-	-A
I_{DD}	sleepsupply current in mode	3.3V@48MHz All Peripherals clock ON				-A
I_{DD}	sleepsupply current in mode	3.3V@24MHz All Peripherals clock ON				-A

I_{DD}	sleepsupply current in mode	3.3V@12MHz All Peripherals clock ON				-A
I_{DD}	sleepsupply current in mode	3.3V@48MHz All Peripherals clock OFF				-A
I_{DD}	sleepsupply current in mode	3.3V@24MHz All Peripherals clock OFF				-A
I_{DD}	sleepsupply current in mode	3.3V@12MHz All Peripherals clock OFF				-A
I_{DD}	deep sleepPower supply in mode flow	3.3V@48MHz	-	5	-	-A
I_{DD}	stopsupply current in mode	3.3V@48MHz	-	0.5	-	-A
I_{DD}	stopsupply current in mode	3.3V@48MHz All Peripherals clock OFF				-A
I_{DD}	stopsupply current in mode	3.3V@24MHz All Peripherals clock OFF				-A
I_{DD}	stopsupply current in mode	3.3V@12MHz All Peripherals clock OFF				-A

Note: All withAll Peripherals clockThe condition item current is measured after linking some basic peripherals.

6.3.5 Embedded reference voltage

symbol	describe	condition	minimum value	typical value	maximum value	unit
V_{REFINT}	internal reference voltage	- 40°C~85°C	1.19	1.2	1.21	V
$tS_vrefint$	ADC sampling time when reading the internal reference voltage	-	-	5	-	us

Note: This is the theoretical value of circuit simulation.

6.3.6 Low Power Mode Wake Up Time

symbol	describe	condition	minimum value	typical value	maximum value	unit
$T_{wusleep}$	sleepmode wakeup	-	-	100	-	us
$T_{wudeeps sleep}$	deep sleepmode wakeup	-	-	140	-	us
T_{wustop}	stopmode wakeup	-	-	350	-	us

Note: This is the theoretical value of circuit simulation.

6.3.7 Internal High Frequency Clock Source Characteristics

symbol	describe	condition	minimum value	typical value	maximum value	unit
f_{HSI}	Clock frequency	-	-	48	-	MHz
TRIM	trim step	-	-	0.8	-	%
ACC	Oscillator Accuracy	VDD=2.0V~3.6V - 40°C~85°C	- 1	-	1	%
DC	duty cycle	VDD=3.3V	45	50	55	%
T_{SU}	Oscillator start-up time	-	-	10	-	-S
I_{DD}	power consumption	-	-	100	85	-A

Note: This is the theoretical value of circuit simulation.

6.3.8 Internal Low Frequency Clock Source Characteristics

symbol	describe	condition	minimum value	typical value	maximum value	unit
f_{LSI}	Clock frequency	-	-	32.768	-	KHz
TRIM	trim step	-	-	-	1.8	%
ACC	Oscillator Accuracy	VDD=2.0V~3.6V - 40°C~85°C	- 3		3	%
DC	duty cycle	VDD=3.3V	45	50	55	%
T_{SU}	Oscillator start-up time	-	-	-	80	-S
I_{DD}	power consumption	-	-	-	1	-A

Note: This is the theoretical value of circuit simulation.

6.3.9 External High Frequency Clock Source Characteristics

symbol	describe	condition	minimum value	typical value	maximum value	unit
f_{HSE}	Clock frequency	-	4	-	32	MHz
T_{SU}	Oscillator start-up time	-	0.3	-	1	ms

I_{DD}	power consumption	VDD = 3.3V, Rm = 45Ω, CL = 10pF@8MHz	-	0.86	1.45	uA
		VDD = 3.3V, Rm = 30Ω, CL = 20pF@16MHz	-	0.8	1.2	uA

Note: This is the theoretical value of circuit simulation.

6.3.10 External Low Frequency Clock Source Characteristics

symbol	describe	condition	minimum value	typical value	maximum value	unit
f_{LSE}	Clock frequency	-	-	32.768	-	KHz
T_{SU}	Oscillator start-up time	CL=20pF	-	600	-	ms
I_{DD}	power consumption	CL = 20pF@8MHz	1.2	-	2	uA

Note: This is the theoretical value of circuit simulation.

6.3.11 PLLscharacteristic

symbol	describe	condition	minimum value	typical value	maximum value	unit
f_{PLL_IN}	input clock frequency	-	4	-	48	MHz
	Input Clock Duty Cycle	-	40	-	60	%
f_{PLL_OUT}	output clock frequency	-	-	72	-	MHz
T_{lock}	PLLslock time	-	-	-	35	us
Jitter	Cycle-to-cycle jitter	-	-		200	ps

Note: This is the theoretical value of circuit simulation.

6.3.12 FLASHmemory characteristics

symbol	describe	condition	minimum value	typical value	maximum value	unit
T_{prog}	32-bitprogramming time	- 40°C~85°C	-	-	20	us
Terase	Sector (0.5KB) wipe time	- 40°C~85°C	-	-	4	ms

IDD	programming current	- 40°C~85°C	-	-	0.9	mA
	Sector Erase Current	- 40°C~85°C	-	-	0.9	mA
	Standbyelectric current	room temperature	-	90	-	uA
	Deep power downelectric current	room temperature	-	3	-	uA
N _{cycle}	Program/erase times	85°C	10	-	-	Ten thousand times
T _{ret}	data retention	85°C	10	-	-	Year

Note: Guaranteed by design, not tested in production.

6.3.13 EMC characteristic

symbol	describe	condition	maximum value	unit
ESD(HBM)	Electrostatic Discharge Human Body Model	TA = 25°C, conform to JEDEC JS-001-2017	2000	V
ESD (CDM)	Electrostatic discharge charging equipment model	TA = 25°C, conform to JEDEC JS-002-2014	500	V
LatchUp	static latch class	TA = 25°C, conform to JEDEC78D	100	mA

Note: This is the theoretical value of circuit simulation.

6.3.14 I/O port characteristics

symbol	describe	condition	minimum value	typical value	maximum value	unit
V _{IL}	Input low level voltage	TC(3.3V IO)	-	-	0.39VDD	V
		FT(5V tolerance IO)	-	-	0.47VDD	V
V _{IH}	Input high level voltage	TC(3.3V IO)	0.55VDD	-	-	V
		FT(5V tolerance IO)	0.51VDD	-	-	V
V _{hys}	Schmitt hysteresis voltage	-	-	240	-	mV
I _{lH}	input leakage current	-	-1	-	1	uA

I _{IL}		-	- 1	-	1	uA
R _{PU}	Weak pull-up equivalent resistance	-	-	40	-	KΩ
R _{PD}	Weak pull-down equivalent resistance	-	-	40	-	KΩ
V _{OL}	output low level	TC(3.3V IO) I _{IO} = 8mA VDD≥2.7V	-	-	0.27	V
V _{oh}	output high level		VDD-0.29	-	-	V
V _{OL}	output low level	TC(3.3V IO) I _{IO} = 20mA VDD≥2.7V	-	-	1	V
V _{oh}	output high level		VDD-0.87	-	-	V
V _{OL_5}	output low level	FT(5V tolerance IO) I _{IO} = 8mA VDD≥2.7V	-	-	0.28	V
V _{OH_5}	output high level		VDD-0.3	-	-	V
V _{OL_5}	output low level	FT(5V tolerance IO) I _{IO} = 20mA VDD≥2.7V	-	-	0.9	V
V _{OH_5}	output high level		VDD-0.91	-	-	V

Note: This is the theoretical value of circuit simulation.

6.3.15 EXTRSTport characteristics

symbol	describe	condition	minimum value	typical value	maximum value	unit
V _{IL}	Input low level voltage	-	-	-	0.3VDD	V
V _{IH}	Input high level voltage	-	0.7VDD	-	-	V
V _{hys}	Schmitt hysteresis voltage	-	0.71	1.19	1.33	V
R _{PU}	Weak pull-up equivalent resistance	V _{IN} =V _{SS}	-	40	-	KΩ
V _f	Input filter pulse voltage	-	-	-	70	ns
V _{NF}	Input unfiltered pulse voltage	-	-	350	-	ns

Note: This is the theoretical value of circuit simulation.

6.3.16 ADC characteristic

symbol	describe	condition	minimum value	typical value	maximum value	unit
V _{DD}	ADC Input voltage	-	2	-	3.6	V
I _{DD}	ADC electric current	V _{DD} = 3.3V, room temperature	-	0.5	-	mV
f _{ADC}	ADC Sampling clock frequency	-	-	48	-	MHz
f _s	Sampling Rate	-	-	2.4	-	MHz
V _A IN	Conversion voltage range	-	0	-	V _{DD}	V
R _A IN	External input impedance	V _{IN} = V _{SS}	-	-	50	kΩ
R _{ADC}	Sampling Switch Resistance	-	-	-	0.65	kΩ
C _{ADC}	Internal sample and hold resistor	-	-	-	3.8	pF
T _{the} s	sampling time	-	1	-	128	1/f _{ADC}
T _{CONV}	total conversion time (including sample time, conversion time, data operation and storage)	f _{ADC} = f _{SYS} = 48MHz	20	-	147	1/f _{ADC}
V _{extvref}	external reference voltage	-	-	-	V _{DD}	V

Note: This is the theoretical value of circuit simulation.

6.3.17 ADC precision

symbol	describe	condition	minimum value	typical value	maximum value	unit
E _T	Total unadjusted error	f _{ADC} = 48M, R _A IN < 3kΩ, V _D DA=2.7V to 3.6V TA = -40 to 85°C	-	-	+/-4	LSB
E _O	Offset error		-	-	+/-1.5	LSB
E _G	Gain error		-	-	+/-2.5	LSB
E _D	Differential linearity error		-	-	+/-1	LSB
E _L	Integral linearity error		-	-	+/-1.7	LSB

Note: This is the theoretical value of circuit simulation.



6.3.18 Temperature Sensor Characteristics

symbol	describe	condition	minimum value	typical value	maximum value	unit
T _L	Linear Accuracy	-	-	-	+/-2.7	°C
Avg_Slope	Slopeaverage	-	3.05	3.13	3.2	mv/°C
V ₂₅	25Voltage at °C	-	1.34	1.543	1.52	V
T _{start}	Start Time	-	4	-	10	us
T _{s_temp}	ADCWhen measuring temperature sample time	-	5	-	-	us

Note: This is the theoretical value of circuit simulation.

6.3.19 Operational Amplifier Characteristics

symbol	describe	condition	minimum value	typical value	maximum value	unit
VDD	voltage range	-	2.0	-	3.6	V
CMIR	input range	-	0	-	VDD	V
V _{opa_offset}	Input offset voltage maximum trim range	25°C, No load output	-	-	8	mV
	Input offset voltage after offset trim	25°C, No load output	-	-	0.4	mV
		2.0V~3.6V, -40°C ~85°C	-	-	0.8	mV
I _{load}	drive current	-	-	-	500	uA
IDD	power consumption	no load, static mode	-	-	320	uA
CMRR	Common mode rejection ratio	-	54	-	196	dB
PSRR	Power supply rejection ratio	-	72	-	138	dB
GBW	bandwidth	-	2.1	4.7	12.6	MHz
SR	Slew Rate	-	2.5	4.1	5.4	V/us
R _{load}	resistive load	-	4	-	-	KΩ

C _{load}	capacitive load	-	-	-	50	pF
V _{OHsat}	High saturation voltage	-	VDD-30	-	-	mV
V _{OLsat}	Low saturation voltage	-	-	-	31	mV
PM	Phase Margin	-	46	67	97	°
GM	Gain Margin	-	6.6	17	30	dB
THD	Total Harmonic Distortion	-	3.1	3.2	3.3	%

Note: This is the theoretical value of circuit simulation.

6.3.20 Comparator Characteristics

symbol	describe	condition	minimum value	typical value	maximum value	unit
VDD	voltage range	-	2.0	-	3.6	V
IDD	power consumption	-	-	30	45	uA
V _{CM}	Input voltage range	-	0.35	0.5VDD	VDD-0.3	V
V _{offset}	enteroffsetVoltage	hysteresis off	-	10	20	mV
V _{hys}	hysteresis window	-	-	60	140	mV
T _{start}	Comparator startup time	-	-	-	400	ns
A _v	DC Voltage Gain	-	45	65	75	dB

Note: This is the theoretical value of circuit simulation.