



maxim
integrated™

MAX32550 Secure SOC User's Guide

*UG25H05
Revision E
01/07/2016*

Maxim Integrated, Inc.
160 Rio Robles
San Jose, CA 95134

Disclaimer

LIFE SUPPORT POLICY

MAXIM'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS PRIOR WRITTEN APPROVAL OF THE PRESIDENT AND GENERAL COUNSEL OF MAXIM INTEGRATED PRODUCTS, INC.

As used herein

Life support devices or systems are devices which (a) are intended for surgical implant into the body, or (b) support or sustain life and whose failure to perform when properly used in accordance with instructions for use provided in the labeling can be reasonably expected to result in a significant injury to the user. A critical component is any component in a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system or to affect its safety or effectiveness.

Document Disclaimer

©2014 by Maxim Integrated, Inc. All rights reserved. Information in this publication concerning the devices, applications, or technology described is intended to suggest possible uses and may be superseded. MAXIM INTEGRATED, INC. DOES NOT ASSUME LIABILITY FOR OR PROVIDE A REPRESENTATION OF ACCURACY OF THE INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED IN THIS DOCUMENT. MAXIM ALSO DOES NOT ASSUME LIABILITY FOR INTELLECTUAL PROPERTY INFRINGEMENT RELATED IN ANY MANNER TO USE OF INFORMATION, DEVICES, OR TECHNOLOGY DESCRIBED HEREIN OR OTHERWISE. The information contained within this document has been verified according to the general principles of electrical and mechanical engineering or registered trademarks of Maxim Integrated, Inc. All other product or service names are the property of their respective owners.

ARM® and Thumb® are registered trademarks of ARM Limited in the European Union and other countries. All other product or service names are the property of their respective owners.



Table of Contents

1.0	Overview.....	20
2.0	Memory Map	21
2.1	Memory Protection Unit.....	21
2.2	Memory Encryption Unit.....	21
2.3	Internal Memory Region	21
2.4	Peripheral Bus Region.....	22
2.5	AHB Peripheral Bus Region	24
2.6	System Bus Interconnect	24
3.0	Digital Peripherals and System Control Registers.....	25
4.0	System Clocks, Reset and Power Management	26
4.1	Clock Generation.....	26
4.1.1	Clock Sources	26
4.1.2	Internal Clocks.....	29
4.1.3	Clock Switching.....	29
4.1.4	Clocks in Battery Backed Mode	29
4.2	Power Management.....	30
4.2.1	Battery Back Monitoring, Switching and Control	30
4.2.2	Power Management and System Control	31
4.2.3	System Power Modes.....	32
4.2.4	Power Monitored Reset Generation.....	34
4.2.5	Power Fail Warning	34
4.3	Global Control Registers (GCR)	35
4.3.1	System Control Register (SCON, Offset 0x0000)	36
4.3.2	Reset Register (RSTR, Offset 0x0004).....	37
4.3.3	Clock Control Register (CLKCN, Offset 0x0008)	39
4.3.4	Power Management Register (PM, Offset 0x000C)	40
4.3.5	PLL0 Control Register (PLL0CN, Offset 0x0010).....	42
4.3.6	PLL1 Control Register (PLL1CN, Offset 0x0014).....	43
4.3.7	Peripheral Clock Divider Register (PCKDIV, Offset 0x0018).....	44
4.3.8	Peripheral Clock Disable Register (PERCKCN, Offset 0x0024)	45
4.3.9	Memory Zeroize Control Register (MEMZCN, Offset 0x002C)	47
4.3.10	Smart Card Clock Control Register (SCCK, Offset 0x0034).....	48
5.0	System Initialization	49
5.1	System Initialization Block (SIB)	50
5.2	System Initialization Registers (SIR)	50
5.2.1	System Initialization Status Register (SISTAT, Offset 0x0000)	51
5.2.2	System Initialization Error Address Register (ERRADDR, Offset 0x0004)	52
6.0	Interrupts	53
6.1	Nested Vectored Interrupt Controller	53
7.0	Instruction Cache Controller	54
7.1	Instruction Cache Controller Registers.....	54
7.1.1	Cache ID Register (CACHE_ID, Offset 0x0000).....	55
7.1.2	Memory Configuration Register (MEMCFG, Offset 0x0004)	56
7.1.3	Cache Control and Status Register (CACHE_CTRL, Offset 0x0100)	57
7.1.4	Invalidate All (INVALIDATE, Offset 0x0700).....	58
8.0	Embedded ICE and Debug	59
9.0	General-Purpose Input/Output (GPIO) Pins.....	60
9.1	GPIO Port Register Blocks.....	60
9.2	GPIO Port Initialization.....	60
9.3	Detailed Description.....	61
9.3.1	Configuring GPIO/Peripheral Modes	61
9.3.2	Configuring GPIO (External) Interrupts	61
9.3.3	Handling Interrupts from GPIO Pins	62
9.3.4	Using the GPIO Wake Function	62



9.4	GPIO Registers.....	63
9.4.1	GPIO Function Enable Register (GPIO_EN, Offset 0x0000).....	64
9.4.2	GPIO Output Enable Register (GPIO_OUT_EN, Offset 0x000C).....	65
9.4.3	GPIO Output Register (GPIO_OUT, Offset 0x0018).....	66
9.4.4	GPIO Output Set Register (GPIO_OUT_SET, Offset 0x001C).....	67
9.4.5	GPIO Output Clear Register (GPIO_OUT_CLR, Offset 0x0020).....	68
9.4.6	GPIO Input Register (GPIO_IN, Offset 0x0024).....	69
9.4.7	GPIO Interrupt Mode Register (GPIO_INT_MOD, Offset 0x0028).....	70
9.4.8	GPIO Interrupt Polarity Register (GPIO_INT_POL, Offset 0x002C).....	71
9.4.9	GPIO Interrupt Enable Register (GPIO_INT_EN, Offset 0x0034).....	72
9.4.10	GPIO Interrupt Enable Set Register (GPIO_INT_EN_SET, Offset 0x0038).....	73
9.4.11	GPIO Interrupt Enable Clear Register (GPIO_INT_EN_CLR, Offset 0x003C).....	74
9.4.12	GPIO Interrupt Status Register (GPIO_INT_STAT, Offset 0x0040).....	75
9.4.13	GPIO Interrupt Clear Register (GPIO_INT_CLR, Offset = 0x0048).....	76
9.4.14	GPIO Wake Enable Register (GPIO_WAKE_EN, Offset 0x004C).....	77
9.4.15	GPIO Interrupt Dual Edge Mode Register (GPIO_INT_DUAL_EDGE, Offset 0x005C).....	78
9.4.16	GPIO Input Configuration Register 1 (GPIO_PAD_CFG1, Offset 0x0060).....	79
9.4.17	GPIO Input Configuration Register 2 (GPIO_PAD_CFG2, Offset 0x0064).....	80
9.4.18	GPIO Alternate Function Enable Register (GPIO_EN1, Offset 0x0068).....	81
9.4.19	GPIO Drive Strength Register (GPIO_DS, Offset 0x00B0).....	82
10.0	Analog to Digital Converter (ADC).....	83
10.1	Overview	83
10.2	ADC Operation.....	83
10.3	ADC Interrupts	84
10.4	ADC Temperature Conversion	84
10.5	ADC Registers.....	85
10.5.1	ADC Control Register (ADC_CN, Offset 0x0000)	86
10.5.2	ADC Status Register (ADC_ST, Offset 0x0004).....	88
10.5.3	ADC Data Register (ADC_DATA, Offset 0x0008)	90
10.5.4	ADC Low Threshold Register n (ADC_LT n , Offset 0x000C + $n*4$, $n=0, 1, 2$).....	91
10.5.5	ADC High Threshold Register n (ADC_HT n , Offset 0x0018 + $n*4$, $n=0, 1, 2$)	92
11.0	Cryptographic Accelerator (Rev A Silicon)	93
11.1	Crypto Buffers	93
11.1.1	Direct Memory Access	93
11.1.2	FIFOs	94
11.2	Symmetric Block Cipher Accelerator.....	95
11.3	Hash Function Accelerator	96
11.4	Galois Field Accelerator	97
11.4.1	Cyclic Redundancy Check	97
11.4.2	Linear Feedback Shift Register	98
11.4.3	Entropy	99
11.5	Hamming Code Generator	99
11.6	Modular Arithmetic Accelerator (MAA).....	101
11.6.1	Operation.....	102
11.6.2	Memory Partition	102
11.6.3	Application Note	103
11.6.4	Error	104
11.7	Cryptographic Registers	105
11.7.1	Crypto Control Register (CRYPTO_CTRL, Offset 0x0000)	106
11.7.2	Cipher Control Register (CIPHER_CTRL, Offset 0x0004).....	108
11.7.3	HASH Control Register (HASH_CTRL, Offset 0x0008)	109
11.7.4	CRC Control Register (CRC_CTRL, Offset 0x000C)	110
11.7.5	Crypto DMA Source Register (DMA_SRC, Offset 0x0010).....	111
11.7.6	Crypto DMA Destination Register (DMA_DEST, Offset 0x0014)	112
11.7.7	Crypto DMA Count Register (DMA_CNT, Offset 0x0018)	113
11.7.8	MAA Control Register (MAA_CTRL, Offset 0x001C)	114



11.7.9	Crypto Data In Register 0 (CRYPTO_DIN_0, Offset 0x0020)	116
11.7.10	Crypto Data In Register [3:1].....	116
11.7.11	Crypto Data Out Register 0 (CRYPTO_DOUT_0, Offset 0x0030).....	117
11.7.12	Crypto Data Out Register [3:1]	117
11.7.13	CRC Polynomial Register (CRC_POLY, Offset 0x0040)	118
11.7.14	CRC Value Register (CRC_VAL, Offset 0x0044)	119
11.7.15	Pseudo-Random Number Generator Register (CRC_PRNG, Offset 0x0048)	120
11.7.16	Hamming ECC Register (HAM_ECC, Offset 0x004C).....	121
11.7.17	Cipher Initial Vector Register 0 (CIPHER_INIT_0, Offset 0x0050).....	122
11.7.18	Cipher Initial Vector Register [3:1]	122
11.7.19	Cipher Key Register 0 (CIPHER_KEY_0, Offset 0x0060)	123
11.7.20	Cipher Key Register [7:1]	123
11.7.21	HASH Message Digest Register 0 (HASH_DIGEST_0, Offset 0x0080).....	124
11.7.22	HASH Digest Register [15:1]	124
11.7.23	HASH Message Size Register 0 (HASH_MSG_SZ_0, Offset 0x00C0).....	125
11.7.24	HASH Message Size Register [3:1].....	125
11.7.25	MAA Word Size Register (MAA_MAWS, Offset 0x00D0)	126
12.0	Cryptographic Accelerator (Rev B Silicon)	127
12.1	Crypto Buffers.....	127
12.1.1	Direct Memory Access.....	127
12.1.2	FIFOs	128
12.2	Symmetric Block Cipher Accelerator.....	129
12.3	Hash Function Accelerator	130
12.4	Galois Field Accelerator.....	131
12.4.1	Cyclic Redundancy Check	131
12.4.2	Linear Feedback Shift Register	132
12.4.3	Entropy.....	133
12.5	Hamming Code Generator.....	133
12.6	Modular Arithmetic Accelerator (MAA).....	135
12.6.1	Operation.....	136
12.6.2	Memory Partition	136
12.6.3	Application Note	137
12.6.4	Error	138
12.7	Cryptographic Registers	139
12.7.1	Crypto Control Register (CRYPTO_CTRL, Offset 0x0000).....	140
12.7.2	Cipher Control Register (CIPHER_CTRL, Offset 0x0004).....	142
12.7.3	HASH Control Register (HASH_CTRL, Offset 0x0008).....	143
12.7.4	CRC Control Register (CRC_CTRL, Offset 0x000C)	144
12.7.5	Crypto DMA Source Register (DMA_SRC, Offset 0x0010).....	145
12.7.6	Crypto DMA Destination Register (DMA_DEST, Offset 0x0014)	146
12.7.7	Crypto DMA Count Register (DMA_CNT, Offset 0x0018).....	147
12.7.8	MAA Control Register (MAA_CTRL, Offset 0x001C)	148
12.7.9	Crypto Data In Register 0 (CRYPTO_DIN_0, Offset 0x0020)	150
12.7.10	Crypto Data In Register [3:1].....	150
12.7.11	Crypto Data Out Register 0 (CRYPTO_DOUT_0, Offset 0x0030).....	151
12.7.12	Crypto Data Out Register [3:1]	151
12.7.13	CRC Polynomial Register (CRC_POLY, Offset 0x0040)	152
12.7.14	CRC Value Register (CRC_VAL, Offset 0x0044)	153
12.7.15	Pseudo-Random Number Generator Register (CRC_PRNG, Offset 0x0048)	154
12.7.16	Hamming ECC Register (HAM_ECC, Offset 0x004C).....	155
12.7.17	Cipher Initial Vector Register 0 (CIPHER_INIT_0, Offset 0x0050).....	156
12.7.18	Cipher Initial Vector Register [3:1]	156
12.7.19	Cipher Key Register 0 (CIPHER_KEY_0, Offset 0x0060)	157
12.7.20	Cipher Key Register [7:1]	157
12.7.21	HASH Message Digest Register 0 (HASH_DIGEST_0, Offset 0x0080).....	158
12.7.22	HASH Digest Register [15:1]	158



12.7.23	HASH Message Size Register 0 (HASH_MSG_SZ_0, Offset 0x00C0).....	159
12.7.24	HASH Message Size Register [3:1].....	159
12.7.25	MAA Word Size Register (MAA_MAWS, Offset 0x00D0).....	160
13.0	Digital to Analog Converter (DAC)	161
13.1	Overview	161
13.2	Interrupt.....	161
13.3	Basic DAC initialization.....	161
13.4	Software Management of Conversion Rate (Mode0)	161
13.5	Hardware Management of Conversion Rate (Mode1).....	161
13.6	Interpolation Filter.....	161
13.7	DAC Registers.....	162
13.7.1	DAC Control Register (DAC_CTRL, Offset 0x0000).....	163
13.7.2	DAC Rate Register (DAC_RATE, Offset 0x0004)	165
13.7.3	DAC Interrupt Register (DAC_INT, Offset 0x0008).....	166
13.7.4	DAC AHB FIFO (DAC_FIFO, Offset 0x0000).....	167
14.0	DMA Controller	168
14.1	DMA Channel Operation	168
14.2	DMA Channel Arbitration and DMA Bursts.....	169
14.3	DMA Source and Destination Addressing	169
14.4	Data Movement from Source to DMA FIFO.....	170
14.5	Data Movement from the DMA FIFO to Destination	171
14.6	Memory Buffer Alignment	172
14.7	Count-to-Zero Condition	172
14.8	Chaining Buffers	172
14.9	DMA Interrupts	173
14.10	Channel Time-outs	173
14.10.1	Global 24-bit PreScaler.....	173
14.10.2	8-bit Timer/Counter	174
14.11	Channel and Register Access Restrictions	174
14.12	Memory-to-Memory DMA	174
14.13	DMA Registers	175
14.13.1	DMA Control Register (DMA_CN, Offset 0x0000).....	176
14.13.2	DMA Interrupt Register (DMA_INT, Offset 0x0004).....	177
14.13.3	DMA Channel 0 Configuration Register (DMA0_CFG, Offset 0x0100).....	178
14.13.4	DMA Channel 0 Status Register (DMA0_ST, Offset 0x0104).....	180
14.13.5	DMA Channel 0 Source Register (DMA0_SRC, Offset 0x0108).....	182
14.13.6	DMA Channel 0 Destination Register (DMA0_DST, Offset 0x010C).....	183
14.13.7	DMA Channel 0 Count Register (DMA0_CNT, Offset 0x0110)	184
14.13.8	DMA Channel 0 Source Reload Register (DMA0_SRC_RLD, Offset 0x0114)	185
14.13.9	DMA Channel 0 Destination Reload Register (DMA0_DST_RLD, Offset 0x0118)	186
14.13.10	DMA Channel 0 Count Reload Register (DMA0_CNT_RLD, Offset 0x011C).....	187
15.0	Flash Controller	188
15.1	Flash Memory Operations	188
15.1.1	Flash Write.....	188
15.1.2	Flash Page Erase	188
15.1.3	Flash Mass Erase.....	188
15.1.4	Burst Write	188
15.1.5	Register Access Restriction.....	188
15.1.6	Interrupt	188
15.2	Flash Controller Registers	189
15.2.1	Flash Address Register (FLSH_ADDR, Offset 0x0000)	190
15.2.2	Flash Clock Divide Register (FLSH_CLKDIV, Offset 0x0004)	191
15.2.3	Flash Control Register (FLSH_CN, Offset 0x0008).....	192
15.2.4	Flash Interrupt Register (FLSH_INT, Offset 0x0024)	193
15.2.5	Flash Data Register 0 (FLSH_DATA0, Offset 0x0030)	194
15.2.6	Flash Data Register 1 (FLSH_DATA1, Offset 0x0034)	195



15.2.7	Flash Data Register 2 (FLSH_DATA2, Offset 0x0038)	196
15.2.8	Flash Data Register 3 (FLSH_DATA3, Offset 0x003C)	197
15.2.1	Access Control Register (ACNTL, Offset 0x0040)	198
16.0	Inter-Integrated Circuit (I ² C)	199
16.1	Description	199
16.2	Protocol	199
16.2.1	Bit Transfer	199
16.2.2	Addressing	200
16.2.3	General Call Address	200
16.2.4	START Byte	200
16.2.5	Clock Synchronization	200
16.2.6	Bus Arbitration.....	201
16.3	FIFO	201
16.4	Interrupt.....	202
16.5	DMA.....	202
16.6	I ² C Registers	203
16.6.1	I ² C Control Register (I2C_CR, Offset 0x0000)	204
16.6.2	I ² C Status Register (I2C_SR, Offset 0x0004)	205
16.6.3	I ² C Interrupt Enable Register (I2C_IER, Offset 0x0008)	206
16.6.4	I ² C Interrupt Status Register (I2C_ISR, Offset 0x000C)	207
16.6.5	I ² C Baudrate Register (I2C_BRR, Offset 0x0010)	208
16.6.6	I ² C RX Count Register (I2C_RCR, Offset 0x0014)	209
16.6.7	I ² C TX FIFO Output Register (I2C_TXR, Offset 0x0018).....	210
16.6.8	I ² C Data Register (I2C_DR, Offset 0x001C).....	211
16.6.9	I ² C DMA Register (I2C_DMR, Offset 0x0030)*	212
17.0	Magnetic Stripe Reader	213
17.1	Magnetic Stripe Reader Features.....	213
17.2	General Description and Memory Requirement	213
17.3	Magnetic Stripe Reading Overview	213
17.4	MSR Dynamic Filtering	213
17.5	DMA and Interrupt Operations.....	214
17.5.1	A3 Silicon Revision and Prior.....	214
17.5.2	B1 Silicon Revision and Beyond	214
17.6	Modes of Operations	215
17.7	Magnetic Stripe Reader Registers	217
17.7.1	MSR DSP Address Register (DSPADDR, Offset 0x0000).....	218
17.7.2	Track 1 & 3 Scale Factor Register (T13_SCALE, Index 0x00).....	219
17.7.3	Track 2 Scale Factor Register (T2_SCALE, Index 0x01).....	220
17.7.4	Track 1 & 3 Zero Cross Threshold Fast Register (T13_ZCT_FAST, Index 0x02).....	221
17.7.5	Track 1 & 3 Zero Cross Threshold Middle Register (T13_ZCT_MID, Index 0x03).....	222
17.7.6	Track 1 & 3 Zero Cross Threshold Slow Register (T13_ZCT_SLOW, Index 0x04)	223
17.7.7	Track 2 Zero Cross Threshold Fast Register (T2_ZCT_FAST, Index 0x05).....	224
17.7.8	Track 2 Zero Cross Threshold Middle Register (T2_ZCT_MID, Index 0x06).....	225
17.7.9	Track 2 Zero Cross Threshold Slow Register (T2_ZCT_SLOW, Index 0x07)	226
17.7.10	Leading 0 Detection Count (START_PCNT, Index 0x08)	227
17.7.11	Track 1 Data Count (T1_DATA_CNT, Index 0x09)	228
17.7.12	Track 2 Data Count (T2_DATA_CNT, Index 0x0A)	229
17.7.13	Track 3 Data Count (T3_DATA_CNT, Index 0x0B)	230
17.7.14	ADC Configuration Register (ADCCFG1, Index 0x12)	231
17.7.15	DSP Control Register (DSPCTRL, Offset 0x0004)	232
17.7.16	MSR DSP Data Register (DSPDATA, Offset 0x0008)	233
17.7.17	Track 1 Data Output Register (T1FIFO, Offset 0x000C)	234
17.7.18	Track 2 Data Output Register (T2FIFO, Offset 0x0010)	235
17.7.19	Track 3 Data Output Register (T3FIFO, Offset 0x0014)	236
17.7.20	DSP Interrupt Flag Register (DSPINT, Offset 0x0018)*	237
17.7.21	DSP Interrupt Enable Register (DSPINTEN, Offset 0x001C)*	238



17.7.22	DSP Status Register (DSPSTAT, Offset 0x0020)*	239
18.0	Mono LCD Controller	240
18.1	Operation Overview	240
18.2	Automatic Mode	240
18.2.1	Automatic Read Operation	241
18.2.2	Automatic Write Operation	241
18.3	Manual Mode	242
18.4	Interrupt Operation	242
18.5	Mono LCD Controller Registers	243
18.5.1	LCD Control Register (LCD_EN, Offset 0x0000)	244
18.5.2	LCD Enable Period Register (LCD_EPR, Offset 0x0004)	245
18.5.3	LCD Status Register (LCD_SR, Offset 0x0008)	246
18.5.4	LCD Data Register (LCD_DATA, Offset 0x000C)	247
19.0	Real Time Clock (RTC)	248
19.1	RTC Alarm Functions	248
19.1.1	Time-of-Day Alarm	249
19.1.2	Sub-Second Alarm	249
19.2	RTC Register Access	249
19.2.1	RTC Register Write Protection	249
19.2.2	RTC Register Read Protection	249
19.2.3	RTC Count Register Access	250
19.2.4	RTC Alarm Register Access	250
19.2.5	RTC Trim Register Access	250
19.3	RTC Calibration	250
19.4	RTC Registers	251
19.4.1	RTC Second Counter Register (RTC_SEC, Offset 0x0000)	252
19.4.2	RTC Sub-second Counter Register (RTC_SSEC, Offset 0x0004)	253
19.4.3	RTC Time-of-Day Register (RTC_RAS, Offset 0x0008)	254
19.4.4	RTC Sub-second Alarm Register (RTC_RSSA, Offset 0x000C)	255
19.4.5	RTC Control Register (RTC_CN, Offset 0x0010)	256
19.4.6	RTC Trim Register (RTC_TRIM, Offset 0x0014)	258
20.0	Secure Keyboard	259
20.1	Keyboard Scanning	259
20.2	Drive and Sense Pins	259
20.3	Key Debounce	259
20.4	Key Event Storage	259
20.5	Interrupts	259
20.6	Secure Keyboard Registers	260
20.6.1	Keyboard Control Register 0 (KBD_CR0, Offset 0x0000)	261
20.6.2	Keyboard Control Register 1 (KBD_CR1, Offset 0x0004)	262
20.6.3	Keyboard Status Register (KBD_SR, Offset 0x0008)	263
20.6.4	Keyboard Interrupt Enable Register (KBD_IER, Offset 0x000C)	264
20.6.5	Keyboard Interrupt Status Register (KBD_ISR, Offset 0x0010)	265
20.6.6	Keyboard Key 0 Register (KBD_K0R, Offset 0x0014)	266
20.6.7	Keyboard Key 1 Register (KBD_K1R, Offset 0x0018)	267
20.6.8	Keyboard Key 2 Register (KBD_K2R, Offset 0x001C)	268
20.6.9	Keyboard Key 3 Register (KBD_K3R, Offset 0x0020)	269
21.0	Security Monitor	270
21.1	Destructive Reset Source	270
21.2	Active Die Shield	271
21.3	Temperature Sensor	271
21.4	Battery Voltage Monitor	271
21.5	External Sensor	272
21.5.1	External Sensor Warning	273
21.5.2	Power Fail Warning	273
21.6	Security Monitor Registers	274



21.6.1	External Sensor Control Register (EXTSCN, Offset 0x0000)	275
21.6.2	Internal Sensor Control Register (INTSCN, Offset 0x0004)	277
21.6.3	Security Alarm Register (SECALM, Offset 0x0008)	279
21.6.4	Security Diagnostic Register (SECDIAG, Offset 0x000C)	281
21.6.5	DRS Log RTC Register (DLRTC, Offset 0x0010)	283
21.6.6	Security Monitor Status Register (SECST, Offset 0x0034)	284
22.0	Serial Peripheral Interface (SPI).....	285
22.1	Operation.....	287
22.2	SPI Signals	287
22.2.1	Master-In, Slave-Out.....	287
22.2.2	Master-Out, Slave-In.....	287
22.2.3	Serial Clock.....	287
22.2.4	Slave Select.....	288
22.3	SPI Clock Phase and Polarity Control	288
22.3.1	Transfer Format (PHASE = "0").....	289
22.3.2	Transfer Format (PHASE = "1").....	290
22.4	Data Movement	290
22.5	Configuration for Master, Slave and Multi-Master Modes	291
22.5.1	Single Master Operation	291
22.5.2	Multi-Master Operation	291
22.5.3	Slave Operation.....	291
22.6	I ² S (Inter-IC Sound) Mode	292
22.6.1	Mute	292
22.6.2	Pause	292
22.6.3	Mono	292
22.6.4	Left Justify	293
22.7	Error Detection	293
22.7.1	Transmit Overrun	293
22.7.2	Mode Fault (Multi-Master Collision).....	294
22.7.3	Slave Mode Abort	294
22.7.4	Receive Overrun	294
22.8	SPI Interrupts.....	294
22.9	Data Interrupt	294
22.9.1	Forced Interrupt	294
22.9.2	Error Condition Interrupt	294
22.9.3	Baud Rate Generator Time-out Interrupt	294
22.10	SPI Baud Rate Generator.....	295
22.10.1	Slave Mode	295
22.10.2	Master Mode	295
22.10.3	Timer Mode	295
22.11	SPI Registers	296
22.11.1	SPI Data Register (SPI_DATA, Offset 0x0000).....	297
22.11.2	SPI Control Register (SPI_CNTL, Offset 0x0004).....	298
22.11.3	SPI Status Register (SPI_STATUS, Offset 0x0008).....	300
22.11.4	SPI Mode Register (SPI_MOD, Offset 0x000C)	302
22.11.5	SPI Baud Rate Generator Register (SPI_BRG, Offset 0x0014).....	304
22.11.6	SPI DMA Register (SPI_DMA, Offset 0x0018).....	305
22.11.7	SPI I ² S Control Register (I2S_CNTL, Offset 0x001C).....	307
23.0	Smart Card	308
23.1	UART Clock Source.....	308
23.2	Smart Card Interrupts	309
23.3	Basic Initialization.....	309
23.3.1	Bypass mode (External Transceiver)	309
23.3.2	Normal mode (Internal Transceiver)	309
23.4	Event Timing	310
23.5	Transferring and Receiving Data.....	310



23.6	Emergency Deactivation	311
23.7	Supply Sequence Options during card session	311
23.8	Smart Card Registers	312
23.8.1	Smart Card Control Register (SC_CR, Offset 0x0000)	313
23.8.2	Smart Card Status Register (SC_SR, Offset 0x0004)	315
23.8.3	Smart Card Pin Register (SC_PN, Offset 0x0008)	317
23.8.4	Smart Card ETU Register (SC_ETUR, Offset 0x000C)	319
23.8.5	Smart Card Guard Time Register (SC_GTR, Offset 0x0010)	320
23.8.6	Smart Card Wait Time 0 Register (SC_WT0R, Offset 0x0014)	321
23.8.7	Smart Card Wait Time 1 Register (SC_WT1R, Offset 0x0018)	322
23.8.8	Smart Card Interrupt Enable Register (SC_IER, Offset 0x001C)	323
23.8.9	Smart Card Interrupt Status Register (SC_ISR, Offset 0x0020)	324
23.8.10	Smart Card Transmit Register (SC_TXR, Offset 0x0024)	326
23.8.11	Smart Card Receive Register (SC_RXR, Offset 0x0028)	327
23.8.12	Smart Card Clock Counter Register (SC_CCR, Offset 0x002C)	328
24.0	TFT Controller	329
24.1	Block Diagram	329
24.2	AHB Master Interface and FIFO Operation	330
24.3	Pixel Process Engine	332
24.4	AMBA APB Slave Interface and Palette RAM	334
24.5	Gray Scale Generation on the STN Panel	334
24.5.1	Frame Rate Modulator	334
24.5.2	Gray Scale Decoder	335
24.6	STN Panel Data Output Formatter	335
24.7	Panel/Pixel Clock Generation	337
24.8	LCD Panel Timing Generation and Settings	337
24.8.1	TFT Mode	337
24.8.2	STN Mode	339
24.9	Interrupt Operation	340
24.10	TFT Controller Registers	340
24.10.1	LCD Clock Register (CLCD_CLK, Offset 0x0000)	341
24.10.2	LCD Vertical Timing Register 0 (CLCD_VTIM_0, Offset 0x0004)	343
24.10.3	LCD Vertical Timing Register 1 (CLCD_VTIM_1, Offset 0x0008)	344
24.10.4	LCD Horizontal Timing Register (CLCD_HTIM, Offset 0x000C)	345
24.10.5	LCD Control Register (CLCD_CTRL, Offset 0x0010)	346
24.10.6	LCD Frame Buffer 0 Address Register (CLCD_FRBUF_0, Offset 0x0018)	348
24.10.7	LCD Frame Buffer 1 Address Register (CLCD_FRBUF_1, Offset 0x001C)	349
24.10.8	LCD Interrupt Enable Register (CLCD_INT_EN, Offset 0x0020)	350
24.10.9	LCD Interrupt Status Register (CLCD_STAT, Offset 0x0024)	351
24.10.10	LCD HSYNC-VSYNC Phase Difference Register (CLCD_HV_PHASE, Offset 0x0030)	352
24.10.11	LCD Palette Register (CLCD_PALETTE_RAM, Offset 0x0400 to 0x07FC)	353
25.0	Timers	354
25.1	Basic Operation	356
25.2	Timer Operating Modes	356
25.2.1	One Shot Mode	356
25.2.2	Continuous Mode	357
25.2.3	Counter Mode	358
25.2.4	PWM Mode	358
25.2.5	Capture Mode	359
25.2.6	Compare Mode	360
25.2.7	Gated Mode	362
25.2.8	Capture/Compare Mode	362
25.2.9	PWM Differential Mode	364
25.3	Timer Registers	365
25.3.1	Timer Count Register (TMR_CNT, Offset 0x0000)	366
25.3.2	Timer Compare Register (TMR_CMP, Offset 0x0004)	367



25.3.3	Timer PWM Register (TMR_PWM, Offset = 0x0008)	368
25.3.4	Timer Interrupt Register (TMR_INT, Offset 0x000C)	369
25.3.5	Timer Control Register (TMR_CN, Offset 0x0010).....	370
25.3.6	Timer Non-Overlapping Compare Register (TMR_NOLCMP, Offset 0x0014)	372
26.0	True Random Number Generator	373
26.1	Overview	373
26.2	Reading the Random Number	373
26.3	AES Key Generation	373
26.4	Interrupt Management	373
26.5	TRNG Registers	374
26.5.1	TRNG Control Register (TRNG_CN, Offset 0x0000).....	375
26.5.2	TRNG Data Register (TRNG_DATA, Offset 0x0004).....	376
27.0	UART (Serial Port)	377
27.1	Detailed Description	377
27.1.1	Format and Baud Rate Selection.....	377
27.1.2	Transferring and Receiving Data	377
27.1.3	UART Interrupts	377
27.1.4	DMA Transfers	378
27.1.5	Hardware Flow Control	378
27.2	UART Instance Register Blocks.....	378
27.3	UART Registers	379
27.3.1	UART Control Register (UART_CTRL, Offset 0x0000).....	380
27.3.2	UART Status Register (UART_STAT, Offset 0x0004).....	381
27.3.3	UART Interrupt Enable Register (UART_INT_EN, Offset 0x0008)	382
27.3.4	UART Interrupt Status Register (UART_INT_STAT, Offset 0x000C)	383
27.3.5	UART Baud Rate Register 0 (UART_BAUD0, Offset 0x0010)	384
27.3.6	UART Baud Rate Register 1 (UART_BAUD1, Offset 0x0014)	385
27.3.7	UART Pin Control Register (UART_PIN, Offset 0x001C).....	386
27.3.8	UART Data Buffer Register (UART_DATA, Offset 0x0020)	387
27.3.9	UART DMA Register (UART_DMA, Offset 0x0030).....	388
28.0	USB Controller.....	389
28.1	Device Operation.....	389
28.2	USB Endpoint.....	390
28.2.1	Endpoint Buffer Descriptor.....	391
28.2.2	Endpoint Buffer Address	393
28.3	USB Device Registers	394
28.3.1	USB Control Register (USB_CN, Offset 0x0000).....	395
28.3.2	USB Device Address Register (USB_DEV_ADDR, Offset 0x0200).....	396
28.3.3	USB Device Control Register (USB_DEV_CN, Offset 0x0204)	397
28.3.4	USB Device Interrupt Register (USB_DEV_INT, Offset 0x0208)	400
28.3.5	USB Device Interrupt Enable Register (USB_DEV_INT_EN, Offset 0x020C).....	402
28.3.6	USB Endpoint Descriptor Base Address Register (USB_EP_BASE, Offset 0x0220).....	404
28.3.7	USB Current Buffer Register (USB_CUR_BUF, Offset 0x0224).....	405
28.3.8	USB IN Buffer Owner Register (USB_IN_OWNER, Offset 0x0228)	406
28.3.9	USB OUT Buffer Owner Register (USB_OUT_OWNER, Offset 0x022C).....	407
28.3.10	USB IN Interrupt Register (USB_IN_INT, Offset 0x0230)	408
28.3.11	USB OUT Interrupt Register (USB_OUT_INT, Offset 0x0234).....	409
28.3.12	USB NAK Interrupt Register (USB_NAK_INT, Offset 0x0238).....	410
28.3.13	USB DMA Error Interrupt Register (USB_DMA_ERR_INT, Offset 0x023C).....	411
28.3.14	USB Buffer Overflow Interrupt Register (USB_BUF_OVR_INT, Offset 0x0240)	412
28.3.15	USB SETUP Register 0 (USB_SETUP0, Offset 0x0260)	413
28.3.16	USB SETUP Register 1 (USB_SETUP1, Offset 0x0264)	414
28.3.17	USB Endpoint <i>n</i> Control Registers (USB_EP <i>n</i> , Offset 0x0280+4 <i>n</i>)	415
29.0	Watchdog Timer.....	418
29.1	Enabling and Disabling the Watchdog Timer.....	418
29.2	Time Delay Period Selection	419



29.3	Watchdog Timer Operation	419
29.4	Watchdog Timer Registers	420
29.4.1	Watchdog Timer Control Register (WDT_CTRL, Offset 0x0000)	421
29.4.2	Watchdog Timer Reset Register (WDT_RST, Offset 0x0004).....	423

E/UG25H05/36488764



List of Tables

Table 2-1. Internal Memory Region	21
Table 2-2. Peripheral Bus Region.....	22
Table 2-3. AHB Peripheral Region.....	24
Table 2-4. System Bus Interconnect	24
Table 4-1. Battery-Backed Switching Functionality	30
Table 4-2. Power Modes.....	32
Table 4-3. Global Control Register Addresses (Base ADDR = 0x4000_0000).....	35
Table 4-4. SCON (Offset 0x0000).....	36
Table 4-5. RSTR (Offset 0x0004)	37
Table 4-6. CLKCN (Offset 0x0008).....	39
Table 4-7. PM (Offset 0x000C).....	40
Table 4-8. PLL0CN (Offset 0x0010).....	42
Table 4-9. PLL1CN (Offset 0x0014).....	43
Table 4-10. PCKDIV (Offset 0x0018)	44
Table 4-11. PERCKCN (Offset 0x0024).....	45
Table 4-12. MEMZCN (Offset 0x002C)	47
Table 4-13. SCCK (Offset 0x0034).....	48
Table 5-1. System Initialization Register Addresses (Base ADDR = 0x4000_0400).....	50
Table 5-2. SISTAT (Offset 0x0000).....	51
Table 5-3. ERRADDR (Offset 0x0004)	52
Table 7-1. Instruction Cache Controller Register Address (Base ADDR = 0x4002_A000).....	54
Table 7-2. CACHE_ID (Offset 0x0000).....	55
Table 7-3. MEMCFG (Offset 0x0004)	56
Table 7-4. CACHE_CTRL (Offset 0x0100).....	57
Table 7-5. INVALIDATE (Offset 0x0700)	58
Table 9-1. GPIO Register Addresses (for each port)	63
Table 9-2. GPIO_EN (Offset 0x0000).....	64
Table 9-3. GPIO_OUT_EN (Offset 0x000C)	65
Table 9-4. GPIO_OUT (Offset 0x0018)	66
Table 9-5. GPIO_OUT_SET (Offset 0x001C)	67
Table 9-6. GPIO_OUT_CLR (Offset 0x0020)	68
Table 9-7. GPIO_IN (Offset 0x0024)	69
Table 9-8. GPIO_INT_MOD (Offset 0x0028)	70
Table 9-9. GPIO_INT_POL (Offset 0x002C)	71
Table 9-10. GPIO_INT_EN (Offset 0x0034)	72
Table 9-11. GPIO_INT_EN_SET (Offset 0x0038)	73
Table 9-12. GPIO_INT_EN_CLR (Offset 0x003C)	74
Table 9-13. GPIO_INT_STAT (Offset 0x0040)	75
Table 9-14. GPIO_INT_CLR (Offset 0x0048)	76
Table 9-15. GPIO_WAKE_EN (Offset 0x004C)	77
Table 9-16. GPIO_INT_DUAL_EDGE (Offset 0x005C)	78
Table 9-17. GPIO_PAD_CFG1 (Offset 0x0060)	79
Table 9-18. GPIO_PAD_CFG2 (Offset 0x0064)	80
Table 9-19. GPIO_EN1 (Offset 0x0068).....	81
Table 9-20. GPIO_DS (Offset 0x00B0)	82
Table 10-1. ADC Register Base Address (Base ADDR = 0x4003_4000).....	85
Table 10-2. ADC_CN (Offset 0x0000)	86
Table 10-3. ADC_ST (Offset 0x0004)	88
Table 10-4. ADC_DATA (Offset 0x0008)	90
Table 10-5. ADC_LT n (Offset 0x000C + n^*4)	91
Table 10-6. ADC_HT n (Offset 0x0018 + n^*4)	92
Table 11-1. Symmetric Block Ciphers	95
Table 11-2. Hash Functions	96
Table 11-3. Common CRC Polynomials	98



Table 11-4. MAA Memory Segments	103
Table 11-5. Memory Binding (Logical Segment 0 Shown).....	103
Table 11-6. Cryptographic Accelerator Register Address (Base ADDR = 0x4000_1000)	105
Table 11-7. CRYPTO_CTRL (Offset 0x0000)	106
Table 11-8. CIPHER_CTRL (Offset 0x0004).....	108
Table 11-9. HASH_CTRL (Offset 0x0008)	109
Table 11-10. CRC_CTRL (Offset 0x000C)	110
Table 11-11. DMA_SRC (Offset 0x0010)	111
Table 11-12. DMA_DEST (Offset 0x0014)	112
Table 11-13. DMA_CNT (Offset 0x0018).....	113
Table 11-14. MAA_CTRL (Offset 0x001C)	114
Table 11-15. CRYPTO_DIN_0 (Offset 0x0020).....	116
Table 11-16. CRYPTO_DOUT_0 (Offset 0x0030).....	117
Table 11-17. CRC_POLY (Offset 0x0040)	118
Table 11-18. CRC_VAL (Offset 0x0044)	119
Table 11-19. CRC_PRNG (Offset 0x0048).....	120
Table 11-20. HAM_ECC (Offset 0x004C).....	121
Table 11-21. CIPHER_INIT_0 (Offset 0x0050)	122
Table 11-22. CIPHER_KEY_0 (Offset 0x0060)	123
Table 11-23. HASH_DIGEST_0 (Offset 0x0080)	124
Table 11-24. HASH_MSG_SZ_0 (Offset 0x00C0).....	125
Table 11-25. MAA_MAWS (Offset 0x00D0).....	126
Table 12-1. Symmetric Block Ciphers	129
Table 12-2. Hash Functions	130
Table 12-3. Common CRC Polynomials	132
Table 12-4. MAA Memory Segments	137
Table 12-5. Memory Binding (Logical Segment 0 Shown).....	137
Table 12-6. Cryptographic Accelerator Register Address (Base ADDR = 0x4000_1000)	139
Table 12-7. CRYPTO_CTRL (Offset 0x0000)	140
Table 12-8. CIPHER_CTRL (Offset 0x0004).....	142
Table 12-9. HASH_CTRL (Offset 0x0008)	143
Table 12-10. CRC_CTRL (Offset 0x000C)	144
Table 12-11. DMA_SRC (Offset 0x0010)	145
Table 12-12. DMA_DEST (Offset 0x0014)	146
Table 12-13. DMA_CNT (Offset 0x0018)	147
Table 12-14. MAA_CTRL (Offset 0x001C)	148
Table 12-15. CRYPTO_DIN_0 (Offset 0x0020).....	150
Table 12-16. CRYPTO_DOUT_0 (Offset 0x0030).....	151
Table 12-17. CRC_POLY (Offset 0x0040)	152
Table 12-18. CRC_VAL (Offset 0x0044)	153
Table 12-19. CRC_PRNG (Offset 0x0048).....	154
Table 12-20. HAM_ECC (Offset 0x004C).....	155
Table 12-21. CIPHER_INIT_0 (Offset 0x0050)	156
Table 12-22. CIPHER_KEY_0 (Offset 0x0060)	157
Table 12-23. HASH_DIGEST_0 (Offset 0x0080)	158
Table 12-24. HASH_MSG_SZ_0 (Offset 0x00C0).....	159
Table 12-25. MAA_MAWS (Offset 0x00D0).....	160
Table 13-1. DAC Register Address (Base ADDR = 0x4003_8000)	162
Table 13-2. DAC FIFO Address (Base ADDR = 0x400B_8000)	162
Table 13-3. DAC_CTRL (Offset 0x0000).....	163
Table 13-4. DAC_RATE (Offset 0x0004).....	165
Table 13-5. DAC_INT (Offset 0x0008)	166
Table 13-6. DAC_FIFO (Offset 0x0000)	167
Table 14-1. Source and Destination Address Definition	170
Table 14-2. Inbound Data Alignment	171
Table 14-3. Outbound Data Alignment	172



Table 14-4. DMA Register Address (Base ADDR = 0x4002_8000).....	175
Table 14-5. DMA Channel Registers Offset.....	175
Table 14-6. DMA_CN (Offset 0x0000)	176
Table 14-7. DMA_INT (Offset 0x0004)	177
Table 14-8. DMA0_CFG (Offset 0x0100)	178
Table 14-9. DMA0_ST (Offset 0x0104)	180
Table 14-10. DMA0_SRC (Offset 0x0108)	182
Table 14-11. DMA0_DST (Offset 0x010C).....	183
Table 14-12. DMA0_CNT (Offset 0x0110)	184
Table 14-13. DMA0_SRC_RLD (Offset 0x0114).....	185
Table 14-14. DMA0_DST_RLD (Offset 0x0118).....	186
Table 14-15. DMA0_CNT_RLD (Offset 0x011C).....	187
Table 15-1. Flash Controller Register Address (Base ADDR = 0x4002_9000).....	189
Table 15-2. FLSH_ADDR (Offset 0x0000)	190
Table 15-3. FLSH_CLKDIV (Offset 0x0004).....	191
Table 15-4. FLSH_CN (Offset 0x0008)	192
Table 15-5. FLSH_INT (Offset 0x0024).....	193
Table 15-6. FLSH_DATA0 (Offset 0x0030).....	194
Table 15-7. FLSH_DATA1 (Offset 0x0034).....	195
Table 15-8. FLSH_DATA2 (Offset 0x0038).....	196
Table 15-9. FLSH_DATA3 (Offset 0x003C).....	197
Table 15-10. ACNTL (Offset 0x0040)	198
Table 16-1. Definition of the First Byte	200
Table 16-2. I ² C Register Address (Base ADDR = 0x4000_C000).....	203
Table 16-3. I2C_CR (Offset 0x0000)	204
Table 16-4. I2C_SR (Offset 0x0004)	205
Table 16-5. I2C_IER (Offset 0x0008)	206
Table 16-6. I2C_ISR (Offset 0x000C)	207
Table 16-7. I2C_BRR (Offset 0x0010)	208
Table 16-8. I2C_RCR (Offset 0x0014)	209
Table 16-9. I2C_TXR (Offset 0x0018)	210
Table 16-10. I2C_DR (Offset 0x001C)	211
Table 16-11. I2C_DMR (Offset 0x0030)*	212
Table 17-1. Magnetic Stripe Reader Register Address (Base ADDR = 0x4002_B000)	217
Table 17-2. DSPADDR (Offset 0x0000).....	218
Table 17-3. T13_SCALE (Index 0x0)	219
Table 17-4. T2_SCALE (Index 0x01)	220
Table 17-5. T13_ZCT_FAST (Index 0x02)	221
Table 17-6. T13_ZCT_MID (Index 0x03).....	222
Table 17-7. T13_ZCT_SLOW (Index 0x04)	223
Table 17-8. T2_ZCT_FAST (Index 0x05).....	224
Table 17-9. T2_ZCT_MID (Index 0x06)	225
Table 17-10. T2_ZCT_SLOW (Index 0x07)	226
Table 17-11. START_PCNT (Index 0x08)	227
Table 17-12. T1_DATA_CNT (Index 0x09)	228
Table 17-13. T2_DATA_CNT (Index 0xA)	229
Table 17-14. T3_DATA_CNT (Index 0xB)	230
Table 17-15. ADCCFG1 (Index 0x12)	231
Table 17-16. DSPCTRL (Offset 0x0004)	232
Table 17-17. DSPDATA (Offset 0x0008)	233
Table 17-18. T1FIFO (Offset 0x000C).....	234
Table 17-19. T2FIFO (Offset 0x0010)	235
Table 17-20. T3FIFO (Offset 0x0014)	236
Table 17-21. DSPINT (Offset 0x0018)*	237
Table 17-22. DSPINTEN (Offset 0x001C)*	238
Table 17-23. DSPSTAT (Offset 0x0020)*	239



Table 18-1. Mono LCD Controller Registers (Base ADDR = 0x4003_0000).....	243
Table 18-2. LCD_CN (Offset 0x0000).....	244
Table 18-3. LCD_EPR (Offset 0x0004).....	245
Table 18-4. LCD_SR (Offset 0x0008)	246
Table 18-5. LCD_DATA (Offset 0x000C).....	247
Table 19-1. RTC Register Address (Base ADDR = 0x4000_6000).....	251
Table 19-2. RTC_SEC (Offset 0x0000)	252
Table 19-3. RTC_SSEC (Offset 0x0004).....	253
Table 19-4. RTC_RAS (Offset 0x0008).....	254
Table 19-5. RTC_RSSA (Offset 0x000C).....	255
Table 19-6. RTC_CN (Offset 0x0010).....	256
Table 19-7. RTC_TRIM (Offset 0x0014).....	258
Table 20-1. Secure Keyboard Register Addresses (Base ADDR = 0x4003_2000)	260
Table 20-2. KBD_CR0 (Offset 0x0000).....	261
Table 20-3. KBD_CR1 (Offset 0x0004).....	262
Table 20-4. KBD_SR (Offset 0x0008).....	263
Table 20-5. KBD_IER (Offset 0x000C).....	264
Table 20-6. KBD_ISR (Offset 0x0010).....	265
Table 20-7. KBD_K0R (Offset 0x0014)	266
Table 20-8. KBD_K1R (Offset 0x0018)	267
Table 20-9. KBD_K2R (Offset 0x001C).....	268
Table 20-10. KBD_K3R (Offset 0x0020)	269
Table 21-1. Security Monitor Register Address (Base ADDR = 0x4000_4000).....	274
Table 21-2. EXTSCN (Offset 0x0000).....	275
Table 21-3. INTSCN (Offset 0x0004).....	277
Table 21-4. SECALM (Offset 0x0008)	279
Table 21-5. SECDIAG (Offset 0x000C)	281
Table 21-6. DLRTC (Offset 0x0010)	283
Table 21-7. SECST (Offset 0x0034)	284
Table 22-1. PHASE and CLKPOL Operation.....	288
Table 22-2. SPI Register Addresses (for each instance)	296
Table 22-3. SPI_DATA (Offset 0x0000)	297
Table 22-4. SPI_CNTL (Offset 0x0004)	298
Table 22-5. SPI_STATUS (Offset 0x0008)	300
Table 22-6. SPI_MOD (Qffset 0x000C).....	302
Table 22-7. SPI_BRG (Offset 0x0014)	304
Table 22-8. SPI_DMA (Offset 0x0018)	305
Table 22-9. I2S_CNTL (Offset 0x001C)	307
Table 23-1. Smart Card Clock Selection	308
Table 23-2. Smart Card Register Addresses (Base ADDR = 0x4002_C000).....	312
Table 23-3. SC_CR (Offset 0x0000)	313
Table 23-4. SC_SR (Offset 0x0004).....	315
Table 23-5. SC_PN (Offset 0x0008)	317
Table 23-6. SC_ETUR (Offset 0x000C)	319
Table 23-7. SC_GTR (Offset 0x0010)	320
Table 23-8. SC_WT0R (Offset 0x0014).....	321
Table 23-9. SC_WT1R (Offset 0x0018).....	322
Table 23-10. SC_IER (Offset 0x001C)	323
Table 23-11. SC_ISR (Offset 0x0020)	324
Table 23-12. SC_TXR (Offset 0x0024)	326
Table 23-13. SC_RXR (Offset 0x0028)	327
Table 23-14. SC_CCR (Offset 0x002C)	328
Table 24-1. Palette RAM Data Format	334
Table 24-2. Temporal Dither Sequence	335
Table 24-3. STN Data Output Format	336
Table 24-4. LCD Panel Signals	336



Table 24-5. SYSCLK/PIXEL Clock Divider Ratio	337
Table 24-6. LCD Interface Register Address (Base ADDR = 0x4003_1000).....	340
Table 24-7. CLCD_CLK (Offset 0x0000).....	341
Table 24-8. CLCD_VTIM_0 (Offset 0x0004).....	343
Table 24-9. CLCD_VTIM_1 (Offset 0x0008).....	344
Table 24-10. CLCD_HTIM (Offset 0x000C).....	345
Table 24-11. CLCD_CTRL (Offset 0x0010).....	346
Table 24-12. CLCD_FRBUF_0 (Offset 0x0018)	348
Table 24-13. CLCD_FRBUF_1 (Offset 0x001C)	349
Table 24-14. CLCD_INT_EN (Offset 0x0020).....	350
Table 24-15. CLCD_STAT (Offset 0x0024).....	351
Table 24-16. CLCD_HV_PHASE (Offset 0x0030)	352
Table 24-17. CLCD_PALETTE_RAM (Offset 0x0400 to 0x07FC).....	353
Table 25-1. Timer Register Address Offset (For Each Instance).....	365
Table 25-2. TMR_CNT (Offset 0x0000).....	366
Table 25-3. TMR_CMP (Offset 0x0004)	367
Table 25-4. TMR_PWM (Offset 0x0008)	368
Table 25-5. TMR_INT (Offset 0x000C)	369
Table 25-6. TMR_CN (Offset 0x0010)	370
Table 25-7. TMR_NOLCMP (Offset 0x0014)	372
Table 26-1. TRNG Register Address (Base ADDR = 0x400B_5000)	374
Table 26-2. TRNG_CN (Offset 0x0000)	375
Table 26-3. TRNG_DATA (Offset 0x0004)	376
Table 27-1. UART Register Addresses (for each instance)	379
Table 27-2. UART_CTRL (Offset 0x0000)	380
Table 27-3. UART_STAT (Offset 0x0004).....	381
Table 27-4. UART_INT_EN (Offset 0x0008).....	382
Table 27-5. UART_INT_STAT (Offset 0x000C)	383
Table 27-6. UART_BAUD0 (Offset 0x0010)	384
Table 27-7. UART_BAUD1 (Offset 0x0014)	385
Table 27-8. UART_PIN (Offset 0x001C)	386
Table 27-9. UART_DATA (Offset 0x0020)	387
Table 27-10. UART_DMA (Offset 0x0030)	388
Table 28-1. Endpoint Buffer Descriptor	392
Table 28-2. USB Endpoint Address	393
Table 28-3. USB Device Registers (Base ADDR= 0x400B_0000).....	394
Table 28-4. USB_CN (Offset 0x0000)	395
Table 28-5. USB_DEV_ADDR (Offset 0x0200)	396
Table 28-6. USB_DEV_CN (Offset 0x0204).....	397
Table 28-7. USB_DEV_INT (Offset 0x0208).....	400
Table 28-8. USB_DEV_INT_EN (Offset 0x020C)	402
Table 28-9. USB_EP_BASE (Offset 0x0220)	404
Table 28-10. USB_CUR_BUF (Offset 0x0224)	405
Table 28-11. USB_IN_OWNER (Offset 0x0228).....	406
Table 28-12. USB_OUT_OWNER (Offset 0x022C)	407
Table 28-13. USB_IN_INT (Offset 0x0230)	408
Table 28-14. USB_OUT_INT (Offset 0x0234)	409
Table 28-15. USB_NAK_INT (Offset 0x0238)	410
Table 28-16. USB_DMA_ERR_INT (Offset 0x023C)	411
Table 28-17. USB_BUF_OVR_INT (Offset 0x0240)	412
Table 28-18. USB_SETUP0 (Offset 0x0260)	413
Table 28-19. USB_SETUP1 (Offset 0x0264)	414
Table 28-20. USB_EPn (Offset 0x0280+4n).....	415
Table 29-1. Watchdog Timer Interrupt Period Example	419
Table 29-2. Watchdog Timer Register Addresses	420
Table 29-3. WDT_CTRL (Offset 0x0000)	421



Table 29-4. WDT_RST (Offset 0x0004)	423
-------------------------------------------	-----

E/UG25H05/36488764



List of Figures

Figure 1-1. Block Diagram	20
Figure 4-1. Clock Sources	27
Figure 4-2. On-Chip Oscillator.....	28
Figure 4-3. Power Diagram	31
Figure 4-4. Power Mode Selection	32
Figure 5-1. System Initialization Block Start-up Flow	49
Figure 10-1. ADC Block Diagram.....	83
Figure 10-2. ADC Sample Acquisition.....	84
Figure 11-1. Crypto Accelerator Block Diagram	93
Figure 11-2. DMA Block Diagram.....	94
Figure 11-3. Galois Field CRC and LFSR Architecture.....	98
Figure 11-4. Hamming XOR Calculation	100
Figure 12-1. Crypto Accelerator Block Diagram	127
Figure 12-2. DMA Block Diagram.....	128
Figure 12-3. Galois Field CRC and LFSR Architecture.....	132
Figure 12-4. Hamming XOR Calculation	134
Figure 14-1. DMA Controller Block Diagram	168
Figure 16-1. I ² C Start and Stop Conditions	199
Figure 16-2. General Call Address, Bytes Format.....	200
Figure 16-3. Clock Synchronization.....	201
Figure 16-4. Master Arbitration.....	201
Figure 17-1. Magnetic Stripe Bit Encoding.....	214
Figure 17-2. MSR Peak Detection Example.....	216
Figure 18-1. Mono LCD Controller Block Diagram.....	240
Figure 18-2. LCD Read Operation	241
Figure 18-3. LCD Write Operation	242
Figure 19-1. RTC Block Diagram	248
Figure 21-1. Security Monitor Block Diagram.....	270
Figure 21-2. External Sensors Block Diagram.....	273
Figure 22-1. SPI Block Diagram	285
Figure 22-2. SPI Single-Master, Single-Slave	286
Figure 22-3. SPI Multi-Master, Multi-Slave	286
Figure 22-4. SPI Slave.....	286
Figure 22-5. SPI Timing (PHASE = "0")	289
Figure 22-6. SPI Timing (PHASE = "1")	290
Figure 22-7. I ² S Mode (I ² S_EN=1, I ² S_LJ=0).....	293
Figure 22-8. I ² S Mode (I ² S_EN=1, I ² S_LJ=1).....	293
Figure 23-1. Activation and De-activation Sequence	310
Figure 24-1. Block Diagram of the TFT controller	331
Figure 24-2. TFT Mode	337
Figure 24-3. STN Mode	339
Figure 25-1. 32 bit Timer Block Diagram	355
Figure 25-2 Timer Output Complementary Waveforms.....	364
Figure 28-1. USB Device Block Diagram	390
Figure 28-2. Endpoint Buffer Descriptor Memory	391
Figure 29-1. Watchdog Timer Block Diagram	418

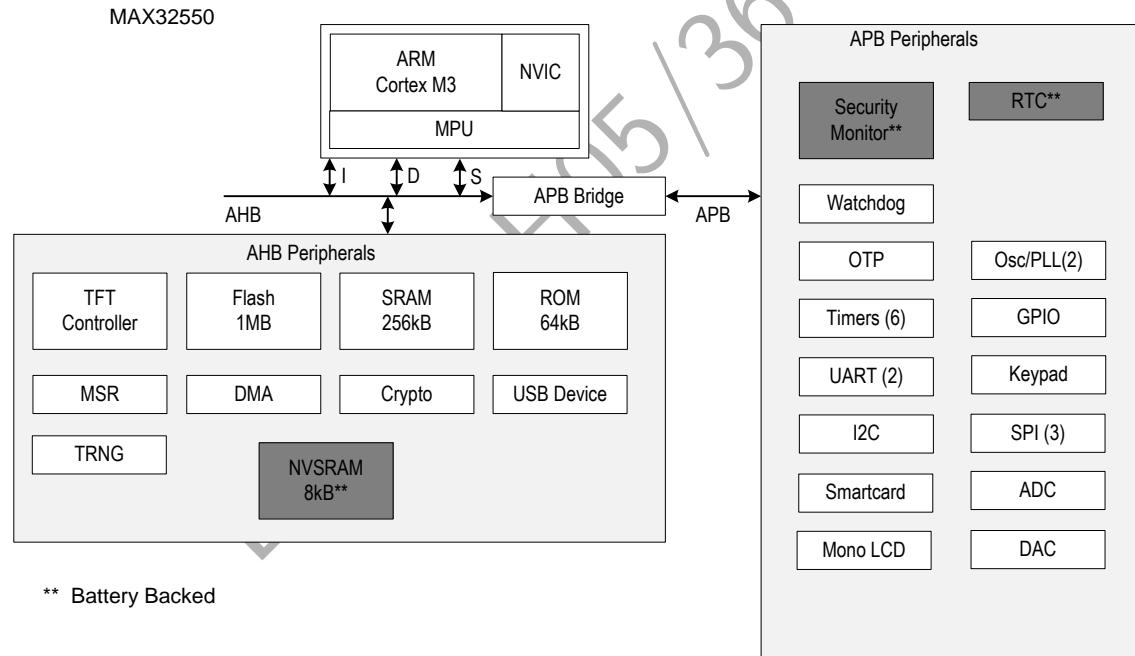
1.0 Overview

The MAX32550 provides an interoperable, secure, and cost-effective solution to build new generations of trusted devices such as mobile chip & PIN solutions. The MAX32550 incorporates a 32-bit ARM Cortex-M3 microcontroller, 1MB of flash, 256KB of system RAM, 4KB of One-Time-Programmable (OTP) memory, 64KB of Boot ROM, 8KB of battery-backed and AES self-encrypted SRAM.

In addition to hardware crypto functions, the MAX32550 provides a true random number generator, battery-backed Real Time Clock, environmental and tamper detection circuitry to facilitate system-level security for the application.

The MAX32550 microcontroller includes multiple communication interfaces. One USB device controller with its respective USB transceiver, one Smart Card controller with an embedded transceiver to directly support 1.8V, 3.3V and 5.0V cards, three SPI ports, two UARTs, and an I₂C bus are also provided. There are six on-chip timers, four of which also support PWM output generation for direct control of external devices. An integrated Secure Keypad, magnetic stripe reader controller, mono LCD interface, color TFT controller, 2-channel 10-bit ADC with an on-chip temperature sensor directly connected to it and one channel 8-bit DAC provide an integrated solution for mobile chip & PIN devices.

Figure 1-1. Block Diagram



** Battery Backed

2.0 Memory Map

The microcontroller has several memory regions. The memory map is divided into internal memory, external memory and peripheral space. The device has 264KB of SRAM.

2.1 Memory Protection Unit

Refer to ARM Cortex-M3 Technical Reference Manual for Memory Protection Unit available at www.arm.com

2.2 Memory Encryption Unit

Memory Encryption Unit (MEU) encrypts the data written to and read from NVSRAM. The encryption uses an AES-256 algorithm in ECB mode. By setting TRNG_CN.AESKG a random key is generated and stored in a secure register. This key is used by MEU to perform crypto functions. For further information, refer to Section 26.3 and Table 26-2 on pages 373 and 375 respectively. MEU is always enabled and its operation is transparent from the user and the software. All other memories are stored in plain text.

2.3 Internal Memory Region

The device's internal memory region contains the program and data memories for the CPU and any peripheral functions that require bus mapped memory. The allocation for this region is shown in Table 2-1.

Table 2-1. Internal Memory Region

PROGRAM	KByte	Start Addr	End Addr
ROM	64	0000 0000	0000 FFFF
Reserved	262080	0001 0000	0FFF FFFF
Flash Block 0	512	1000 0000	1007 FFFF
Flash Block 1	512	1008 0000	100F FFFF
Maxim OTP	4	1010 0000	1010 0FFF
User OTP	4	1010 1000	1010 1FFF
Reserved	523248	1011 1000	1FFF FFFF

DATA	KByte	Start Addr	End Addr
Data SRAM	128	2000 0000	2001 FFFF
Data SRAM	128	2002 0000	2003 FFFF
Reserved	256	2004 0000	2007 FFFF
NVSRAM	8	2008 0000	2008 1FFF
Reserved	504	2008 2000	200F FFFF
Reserved	31744	2010 0000	21FF FFFF
Bit-Band Alias	32768	2200 0000	23FF FFFF
Reserved	458752	2400 0000	3FFF FFFF

2.4 Peripheral Bus Region

Table 2-2 shows the region that contains the peripheral control registers for the device. Note that the addresses shown specify the BASE address for each peripheral function(s). The base address represents the starting address for each group of peripherals (for example, the device watchdog timer control registers start at address 0x4000_3000). In general, the number of peripherals allocated to a section is smaller than the size of the section. Any unused address space is to be considered reserved. The device supports the full APB 3.0 specification. For each peripheral's individual register locations and definition refer to the peripheral's designated chapter.

Table 2-2. Peripheral Bus Region

APB REGION	KByte	Start Addr	End Addr
Global Control Registers	1	4000 0000	4000 03FF
System Initialization Registers	1	4000 0400	4000 07FF
Reserved	1	4000 0800	4000 0BFF
Reserved	1	4000 0C00	4000 0FFF
Crypto (MAA FIFOs & Regs)	4	4000 1000	4000 1FFF
Reserved	4	4000 2000	4000 2FFF
Watch Dog Timer	4	4000 3000	4000 3FFF
Security Monitor	4	4000 4000	4000 4FFF
AES Keys	1	4000 5000	4000 53FF
Reserved	1	4000 5400	4000 57FF
Reserved	2	4000 5800	4000 5FFF
RTC	4	4000 6000	4000 6FFF
Reserved	4	4000 7000	4000 7FFF
GPIO 0	4	4000 8000	4000 8FFF
GPIO 1	4	4000 9000	4000 9FFF
GPIO 2	4	4000 A000	4000 AFFF
Reserved	4	4000 B000	4000 BFFF
I2C	4	4000 C000	4000 CFFF
Reserved	28	4000 D000	4000 FFFF
Timer 0	4	4001 0000	4001 0FFF
Timer 1	4	4001 1000	4001 1FFF
Timer 2	4	4001 2000	4001 2FFF
Timer 3	4	4001 3000	4001 3FFF
Timer 4	4	4001 4000	4001 4FFF
Timer 5	4	4001 5000	4001 5FFF
Reserved	8	4001 6000	4001 7FFF



APB REGION	KByte	Start Addr	End Addr
SPI 0	4	4001 8000	4001 8FFF
SPI 1	4	4001 9000	4001 9FFF
SPI 2	4	4001 A000	4001 AFFF
Reserved	20	4001 B000	4001 FFFF
UART 0	4	4002 0000	4002 0FFF
UART 1	4	4002 1000	4002 1FFF
Reserved	24	4002 2000	4002 7FFF
DMA	4	4002 8000	4002 8FFF
Flash Controller	4	4002 9000	4002 9FFF
Cache Controller	4	4002 A000	4002 AFFF
Magnetic Stripe DSP	4	4002 B000	4002 BFFF
Smart Card	4	4002 C000	4002 CFFF
Reserved	12	4002 D000	4002 FFFF
Mono LCD Controller	4	4003 0000	4003 0FFF
Color LCD Controller	4	4003 1000	4003 1FFF
Secure Keyboard	4	4003 2000	4003 2FFF
Reserved	4	4003 3000	4003 3FFF
ADC	4	4003 4000	4003 4FFF
Reserved	12	4003 5000	4004 7FFF
DAC	4	4003 8000	4003 8FFF
Reserved	476	4003 9000	400A FFFF

2.5 AHB Peripheral Bus Region

Some peripherals are mapped to the higher-speed AHB. Table 2-3 shows the AHB region along with the BASE addresses.

Table 2-3. AHB Peripheral Region

AHB REGION	KByte	Start Addr	End Addr
USB	4	400B 0000	400B 0FFF
Reserved	12	400B 1000	400B 4FFF
TRNG	4	400B 5000	400B 5FFF
Reserved	8	400B 6000	400B 7FFF
DAC FIFO	4	400B 8000	400B 8FFF
Reserved	4	400B 9000	400B 9FFF
Reserved	4	400B A000	400B AFFF
Reserved	276	400B B000	400F FFFF

* MAX32550 only

2.6 System Bus Interconnect

The system bus interconnect is summarized in Table 2-4. Bus arbitration scheme is fixed burst.

Table 2-4. System Bus Interconnect

	DMA	USB	CLCD	System Controller	Crypto DMA	MSR
System RAM1	X	X	X	X	X	X
System RAM2	X	X	X	X	X	X
SNVRAM				X		
APB Bridge	X			X		
USB Slave				X		
TRNG				X		
DAC Controller				X		

3.0 Digital Peripherals and System Control Registers

All registers used to control core system functions and peripheral modules are memory mapped into the AMBA Peripheral Bus (APB) memory region. This device is compliant with version 3.0 of the APB specification as well as the Secure ARM Platform Specification.

Reserved register bits are designated by “RFU”. Reserved bits should only be written as 0 unless otherwise stated.

Reset Value is normally specified. However, some registers may have different reset value depending on the reset conditions. The following defines other possible reset values:

- N/A – Not affected by this reset condition. This normally applies to registers with limited reset sources, such as Battery On Reset (BOR), etc.
- S – Not specified. This applies to registers with no specify initial value or uninitialized.

The Access field of each register bit is defined as followed:

- R – Read allowed. Additional restriction may apply. See register definition for details.
- W – Write allowed. Additional restriction may apply. See register definition for details.
- S – Set to 1 only. Clear to 0 ignored.
- C – Clear to 0 only. Writes to 1 ignored.

4.0 System Clocks, Reset and Power Management

4.1 Clock Generation

All functional units in the MAX32550 are synchronized to the system clock, except for the USB, DAC and the Smart Card. The basic unit of time is the system clock period, which is derived from PLL0. All storage logic blocks are triggered by the rising edge of the system clock. The USB, DAC and the Smart Card are synchronized to a second internal clock derived from PLL1.

4.1.1 Clock Sources

The internal clock circuitry generates the system clock from one of the following clock sources:

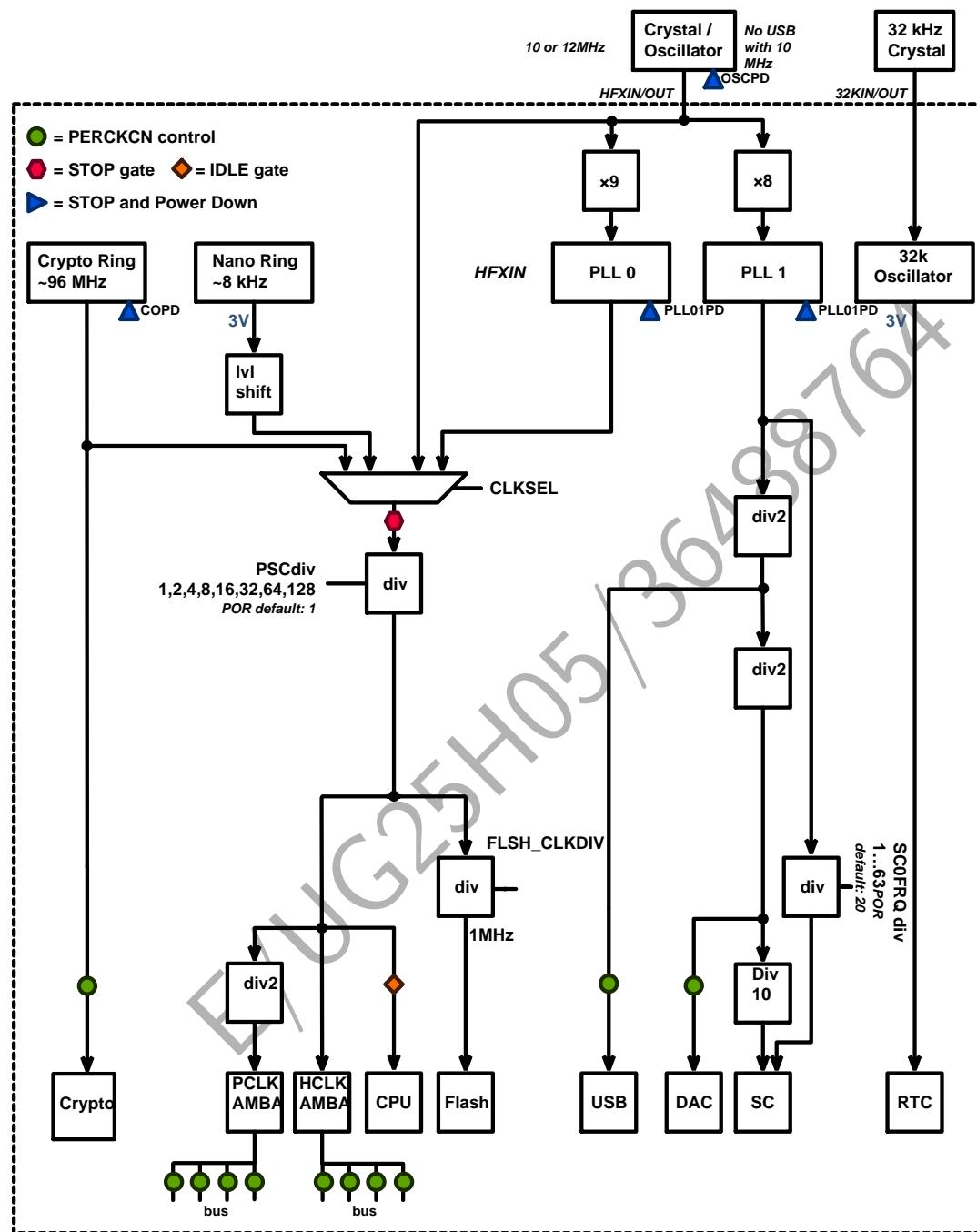
- External clock
- Internal oscillator with an external crystal or resonator
- Internal Nano Ring
- Internal Crypto Oscillator
- PLL0 and PLL1 Output

The external clock and crystal are mutually exclusive since they are input via the same clock pin. Each time the crystal oscillator is started, the following sequence occurs:

1. reset the crystal warm-up counter, and
2. allow the required 16384 external clock cycle warm-up delay

Regardless of the source of the clock, the maximum clock frequency for the ARM core is 108MHz. The system bus is connected to AHB and APB peripherals. The AHB frequency is the same as the core frequency whereas the APB frequency is core or AHB frequency divided by 2.

The core and bus frequencies can both be set lower than their maximums, but the bus frequency must never exceed the core frequency. A simplified clock distribution block diagram is shown in Figure 4-1.

Figure 4-1. Clock Sources

4.1.1.1 External Clock

The PLL can obtain the system clock signal directly from an external clock source. In this configuration, the clock generation circuitry is driven directly by an external clock.

To operate the core from an external clock, connect the clock source to the HFXIN pin and ground the HFXOUT pin while the part is powered up. The clock source should be driven through a CMOS driver. If the clock driver is a TTL gate, its output must be connected to VDD through a pull-up resistor to ensure a satisfactory logic level for active



clock pulses. To minimize system noise on the clock circuitry, the external clock source must meet the maximum rise and fall times and the minimum high and low times specified for the clock source. The external noise can affect clock generation circuit if these parameters do not meet the specification.

4.1.1.2 External Crystal/Resonator

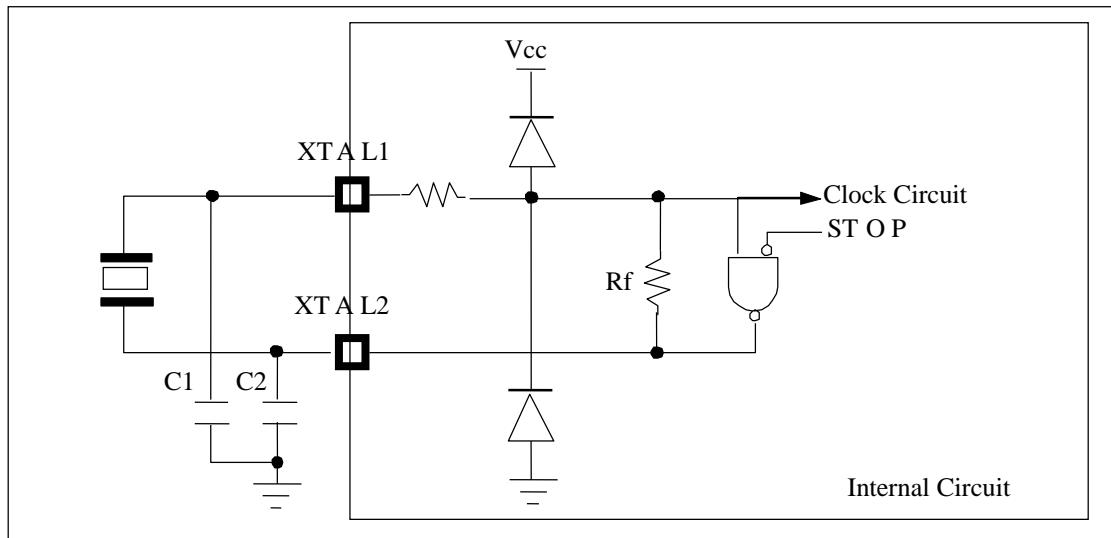
An external quartz crystal or a ceramic resonator can be connected from HFXIN to HFXOUT as the device determining the frequency, as illustrated in Figure 4-2. On-Chip Oscillator.

Crystal specifications, operating temperature, operating voltage, and parasitic capacitance must be considered when designing the internal ring oscillator. The core is designed to operate at a maximum frequency of 108MHz, however the oscillator doesn't need to run at this frequency since the 8X and 9X on PLL1 and PLL0 respectively are used to achieve higher frequencies. In fact, it is expected that an external 12 MHz crystal is used, which is multiplied up to 108 MHz by the PLL0. USB clock is based on the PLL1 which has a 8X multiplication factor to generate the 48MHz clock needed for this block by further dividing it by 2. To further reduce the effects of external noise, a guard ring can be placed around the oscillator circuitry.

Pins HFXIN and HFXOUT are protected by clamping devices against on-chip electrostatic discharge. These clamping devices are diodes parasitic to the feedback resistor Rf in the inverter circuit of the oscillator. The inverter circuit is presented as a NAND gate which can disable clock generation in IDLE mode.

Noise at HFXIN and HFXOUT can adversely affect on-chip clock timing. It is good design practice to place the crystal and capacitors near the oscillator circuitry and connect to HFXIN, HFXOUT and ground with direct short trace. The typical values of external capacitors vary with the type of crystal to be used and should be initially selected based on the load capacitance as suggested by the crystal manufacturer.

Figure 4-2. On-Chip Oscillator



For cost-sensitive applications, a ceramic resonator can be used instead of a crystal. Using the ceramic resonator may require a different circuit configuration and capacitance value.

4.1.1.3 Phase Locked Loop

Two on-chip PLLs are provided to allow for generation of much higher internal system clock frequencies using lower value external quartz crystal or ceramic resonators. There is a 3 μ s start-up time when the clock multiplier is enabled. When the frequency has been locked, the PLLCN.PLL0LOCK or PLLCN.PLL1LOCK flag will be set. Software should poll this flag before switching to the PLL source. The PLL0 has a 9X multiplication factor, and has



a nominal output frequency of 108 MHz. Smart Card, USB and DAC clock is based on PLL1 which has a 8X multiplication factor. The rest of the clocks are based on PLL0.

4.1.1.4 Internal Ring Oscillator

The main clock can also be directly sourced from an internal 96 MHz +/-15% (crypto oscillator) ring oscillator or from the internal 8kHz nano-ring oscillator. Four cycle of warm-up delay is associated with the internal ring oscillator when the system is going through a power-on reset. The 96MHz internal ring oscillator also provides the clock for the cryptographic operations (MAA or cipher). This allows these operations to be "decoupled" from any external clock source so as to complicate any power analysis attacks that could be performed by attempting to vary the parts clock frequency externally.

4.1.2 Internal Clocks

Internal clocks are generated directly from the system clock. The system clock is sourced from one of four clock sources, the internal crypto Ring Oscillator, the internal nano-ring oscillator, External Crystal or External Clock. If the clock source is an external crystal or external clock, then the system clock is always sourced after the PLL has multiplied it.

In addition the core clock is taken from the output of a prescaler that can divide the output of the PLL0 by 1, 2, 4, 8, 16 or 32 as follows:

PSC[2:0]	CPU Clock
000	/1 (108MHz)
001	/2 (54MHz)
010	/4 (27MHz)
011	/8 (13.5MHz)
100	/16 (6.75MHz)
101	/32 (3.375MHz)
110	/64 (1.6875MHz)
111	/128 (0.84375MHz)

The prescaler is controlled via the CLKCN register. See Section 4.3.3 on page 39. The output of the prescaler provides the AHB bus clock. The AHB bus clock is then divided by 2 to generate the APB bus clock.

4.1.3 Clock Switching

Any time the clock source is switched, if there is a switchover time, the Clock Ready (CLKCN.CKRDY) bit will be cleared by hardware to indicate the switchover has not occurred yet. Upon switchover, the Clock Ready bit will be set by hardware. An example of this is switching from an internal oscillator to an external crystal driving a PLL. This bit will be cleared until the crystal warm up has expired and the PLL has locked, at which time the system clock will be switched to the new clock and this bit will be set.

This same mechanism is used on parts that allow running from the ring on startup, the ring shall be used as the system clock source until the selected system clock source is ready, at which time the ring will be switched off and the selected source switched in.

In systems where there is no designated ring oscillator for system startup or there are no delays associated with any clock source switch (i.e.: clocks are halted for switchover) this bit will be reserved.

4.1.4 Clocks in Battery Backed Mode

In battery backed mode, the high frequency oscillator, PLL and crypto oscillators are all switched off to conserve battery power. The only clock sources active in battery back mode are the 32kHz oscillator to run the RTC counter, the internal nano-ring oscillators to run the TRNG (on demand) and the security monitors. Note that in battery backed mode, the TRNG is only enabled when needed by the security monitoring block.

4.2 Power Management

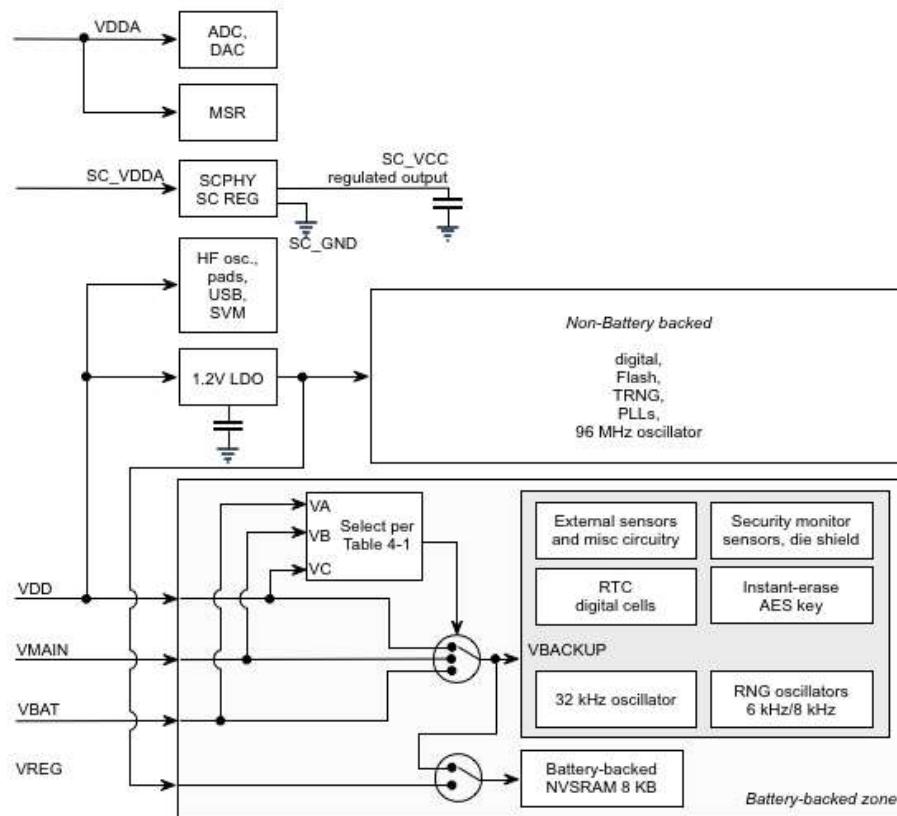
4.2.1 Battery Back Monitoring, Switching and Control

This device includes a battery backup switch in order to automatically and safely switch between different power sources while maintaining non-volatile SRAM content. Many mobile pin-pad implementations cut the main power supply from the secure microcontroller when in stop mode. The internal battery backup switch has two dedicated input pins, VBAT that connects to a traditional non-rechargeable lithium battery and another one called VMAIN that connects to the main rechargeable battery via an external low drop voltage regulator. Tamper detection applies to whatever input pin is currently selected by the internal battery switch. The switch gives the priority over to the VMAIN so the system drains as much current as possible from the main rechargeable battery before switching to the hard-to-replace lithium battery. It allows the end devices to be stored on shelves for long periods of time and without the fear of losing NVSRAM content because of exhausted lithium battery. This is summarized in Table 4-1.

Table 4-1. Battery-Backed Switching Functionality

VDD condition	VMAIN condition	Other condition	VBACKUP
VDD > VRST	N/A	N/A	VDD
VDD < VRST	VMAIN > 2.5V(approx.)	VDD > VMAIN	VDD
VDD < VRST	VMAIN > 2.5V(approx.)	VDD < VMAIN	VMAIN
VDD < VRST	VMAIN < 2.5V(approx.)	VMAIN > VBAT VDD > VMAIN	VDD
VDD < VRST	VMAIN < 2.5V(approx.)	VMAIN < VBAT VDD > VBAT	VDD
VDD < VRST	VMAIN < 2.5V(approx.)	VMAIN > VBAT VDD < VMAIN	VMAIN
VDD < VRST	VMAIN < 2.5V(approx.)	VMAIN < VBAT VDD < VBAT	VBAT

Note that it is also possible to connect VMAIN to VDD in order to switch to the lithium battery as soon as the power is removed from this device. The Power Diagram is shown in Figure 4-3. When in battery back mode, the battery backed RAMs are put into retention mode and only the array is powered. Similarly, the RTC counter continues to run, but any unnecessary digital support circuitry is powered off, so only the counter itself consumes power.

Figure 4-3. Power Diagram

4.2.2 Power Management and System Control

The core incorporates power management features that monitor the power supply voltage levels and support low power operation with three power saving modes, an automatic battery backup mode and a shutdown mode. This chip is developed for low power applications and provides three different levels of operating modes. Further power reduction can be achieved in software by lowering the operating frequencies of the core and AHB/APB busses, independent of the defined modes. Also, the ARM core provides two low power modes internally that can be activated by software in addition to the system operating modes set with the Global Control Register (GCR).

Wake from IDLE and STANDBY modes using the GPIO wake-up feature requires the GPIO pin to be properly configured prior to entering the power saving mode. The following steps should be taken to prepare any GPIO pins that are to be used as wake-up sources:

1. Set the polarity (rising or falling edge) by writing to the GPIO_INT_POL register. The wake function relies upon rising and falling edge detectors to detect and latch the transition on the GPIO pin. These flops operate asynchronously and do not require the GPIO clock to be active.
2. Clear the edge detector flops by writing to the GPIO_INT_CLR register.
3. Activate the GPIO wake function by writing to the GPIO_WAKE_EN.
4. Configure the PMU to wake upon GPIO per System Clocks and Power Management

Because the GPIO Wake function does not require the use of a clock, the clocks to the GPIO modules may be disabled within the Power Management Unit (PMU) during this step.

4.2.3 System Power Modes

The system power mode is set using the MODE[1:0] bits in the Power Management Register (PM). The default is ACTIVE mode (MODE[1:0] = 00) and after power-up, the system starts in this mode. This two bit field selects the current operating mode for the device. Note that code execution only occurs during ACTIVE mode. There are four power modes supported by the chip. These modes are shown in Figure 4-4 and compared in Table 4-2.

Figure 4-4. Power Mode Selection

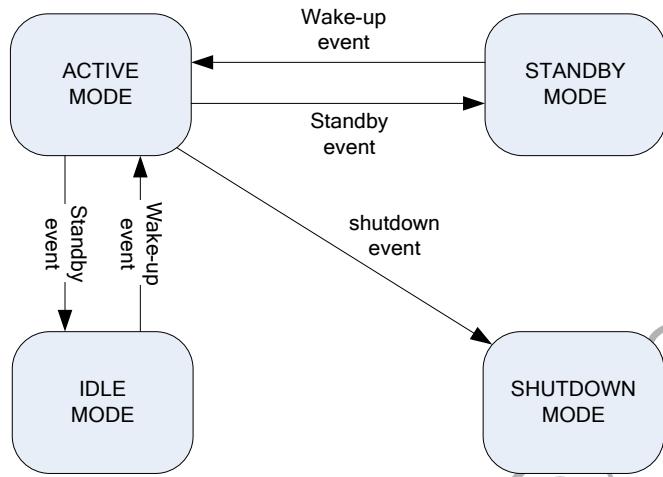


Table 4-2. Power Modes

	ACTIVE	IDLE	STANDBY	SHUTDOWN
VCORE	On	On	On	On
VDD	On	On	On	On
VDDA	On	On	On	On
HF Oscillator	On	On	Configurable	On
Crypto Oscillator	On	On	Configurable	On
PLL	Configurable	Configurable	Configurable	*
Voltage Sensor	On	On	On	On
Memories	On	On	On	On
Core Clock	On	Off	Off	Off
Peripherals Clocks	Configurable	Configurable	Configurable	Off
Wakeup Source	N/A	Configurable	Configurable	N/A
<i>Note*: PLL retains the state prior to SHUTDOWN condition</i>				

4.2.3.1 Active Mode

Active mode is the mode in which the core and all peripherals are on (as configured by PERCKCN), and clocked normally. This is the normal operation mode. All supplies are powered VBACKUP is on VDD, the memories and oscillators are all on.

4.2.3.2 Idle Mode

In Idle Mode the MCU core clock is gated off and program code execution is halted. The peripheral clocks are still on (as configured by PERCKCN) and all analog supplies are powered on.

Idle mode is exited on activity from the following sources:

- GPIO port activity, if configured as a wake up source in the GPIO control register
- RTC alarm
- USB activity

4.2.3.3 Standby Mode

Standby Mode is a low power sleep mode characterized by shutting off as many clocks and clock sources as possible. The internal nano-ring oscillator is used to poll for wakeup sources. The NVSRAM memories are placed in **retention mode** in the same manner as they would be for battery back mode Refer to sections 4.1.4 Clocks in Battery Backed Mode and 4.2.1 Battery Back Monitoring, Switching and Control for more details. Non-volatile memories (flash or ROM) are powered down.

The following sequence must be followed to enter and exit the STANDBY mode at any time or after receiving a suspend order in the USB bus. Exiting STANDBY mode can be achieved either via a USB Resume, RTC alarm or an external wakeup event on the GPIO pin. This external wake-up event is very useful for USB devices that supports USB Remote wake-up (e.g. smart card insertion).

1. Activate STANDBY mode exit by USB event and/or RTC alarm and/or external wake-up: set the PM.GPIOWKEN and/or PM.RTCWKEN and/or PM.USBWKEN bits.
2. Activate GPIO wakeup mode for desired bits and configure polarity of external wake-up event (if relevant).
3. Configure USB data flow (Low speed or Full speed, if relevant).
4. Select STANDBY mode: clear the PM.MODE0 and set the PM.MODE1 bits.
5. Execute assembly 10 “nop” instructions. At this point, the core enters in sleep status. All clocks are disabled and the PLL is powered-down. The chip is in STANDBY mode. When the USB bus receives a resume (or a reset) or RTC Alarm or an external wake-up occur, the Power system exits the STANDBY mode and returns into ACTIVE or IDLE mode.
6. PLL is running, all clocks are enabled. The chip is in ACTIVE. The program restarts after the “nop” sequence.
7. Deactivate USB Wake-Up and/or RTC alarm wake-up and/or external wake-up: clear the PM.GPIOWKEN and/or PM.RTCWKEN and/or PM.USBWKEN bits.

Also note that some of the STANDBY mode behavior can be over-ridden to keep higher speed clocks available in STANDBY mode. By default the nano-ring is used as the system clock, but if faster wakeup is required by the application the crypto oscillator or system oscillator and PLL can be left active at the cost of much higher power consumption in STANDBY mode. This is accomplished by setting bits in the Power Management register (PM).

A Standby wake-up event can be

- Secure Keypad (via GPIO mode)
- Smart Card (via non GPIO mode)
- A reset (external reset or USB reset),
- A USB Resume token on the USB bus,
- A RTC alarm, or

- A configurable level on the GPIO pin.

STANDBY events, Change mode events and Shutdown events are controlled via the PM.MODE field. It is not necessary to change the operating mode to activate or de-activate a peripheral.

In addition to setting the PM.MODE bit to 01, STANDBY mode may also be entered by setting PM.CM3PMUEN (CM3 Power Management Enable) bit to 1 and invoking the Cortex-M3 Sleep or DeepSleep Mode.

Modifying the content of the PERCKCN register automatically changes the states of the associated peripherals in either ACTIVE or IDLE mode. When the operating mode is changed, there is no change to the clocking of the peripherals.

Notes:

Before switching in the STANDBY mode, software must ensure that the UARTs are not busy (the UARTs do not perform a TX/RX). If the UARTs are busy during the ultra-low-power mode, the data will be corrupted upon wake-up.

In order for the suspended USB device to recognize the host resume condition, the chip has to enter the low-power STANDBY state within 2ms after the suspend interrupt is received. If the chip does not enter this low-power mode in the specified time window, only software can wake up the USB device by setting the USB Signal Remote Wakeup register bit, bit 2 of USB_DEV_CN register. A USB software reset caused by setting bit 23 of the Reset Register, RSTR, will correctly reset the entire USB controller. If the chip wakes up for any other reason (RTC, GPIO, etc), the chip must signal a remote wakeup to the host.

4.2.3.4 Shutdown Mode

Shutdown Mode is similar to Standby Mode, but the clock to the wakeup logic is off. This means that all clocks are turned off, power is turned off to as many blocks as possible, and all wakeup sources are disabled. This is a persistent shutdown mode that is only exited when the device undergoes a power-on reset (POR).

4.2.4 Power Monitored Reset Generation

The power monitor in this chip monitors the VDD and the VCORE pins in relation to the on-chip voltage reference. Following power-up of the supplies, a power-on reset will initiate a power-on reset time-out before starting program execution. When power is first applied to the chip, the processor will be held in reset until both supplies have risen above their threshold value and a delay of 16384 oscillator cycles has elapsed, to ensure that power is within tolerance and the clock source has had time to stabilize. Once the reset time-out period has elapsed, the reset condition is removed automatically and software execution will begin at the reset vector location of 0x0000_0000.

The power monitor will invoke a brown-out reset state to halt program execution when either supply drops below the threshold condition. This reset condition will remain while power is below the minimum voltage level. When power returns above the reset threshold, a full power on reset will be performed. Thus a brown-out behaves the same and appears to be the same as a power up.

4.2.5 Power Fail Warning

In addition to Power Monitor Reset Thresholds, high and low Power Fail Warning Thresholds are also available on both VDD and VCORE. Interrupt #16 will fire, if enabled, when either of these supplies are out of range. Note that the interrupt request flag is self-cleared as soon as the fault condition disappears, i.e. VCORE and VDD go back to within the range.

User software can read the status of this bit indirectly via the Security Alarm Register (SECALM).



4.3 Global Control Registers (GCR)

The Global Control Register (GCR) set of registers controls global functions of the MAX32550.

Address assignments for the Global Control registers are outlined in Table 4-3. Reserved register bits should only be written as 0.

Table 4-3. Global Control Register Addresses (Base ADDR = 0x4000_0000)

Offset	Access	Register	Description
0x0000	RW	SCON	System Control Register
0x0004	RW	RSTR	Reset Register
0x0008	RW	CLKCN	Clock Control Register
0x000C	RW	PM	Power Management Register
0x0010	RW	PLL0CN	PLL0 Control Register
0x0014	RW	PLL1CN	PLL1 Control Register
0x0018	RW	PCKDIV	Peripheral Clock Divider Register
0x0024	RW	PERCKCN	Peripheral Clock Disable Register
0x002C	RW	MEMZCN	Memory Zeroize Control Register
0x0030	R	RFU	Reserved for future use
0x0034	RW	SCCK	Smart Card Clock Control Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



4.3.1 System Control Register (SCON, Offset 0x0000)

Table 4-4. SCON (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	CHKRES	CHKRD	CCHK	RFU[4:0]				
Reset	0	0	0	0 0000				
Access	R	R	RS	R				
Position	7	6	5	4	3	2	1	0
Field	RFU[2:0]			FLASH_PAGE_FLIP	ONVMAIN	RFU	RFU	BSTAPEN
Reset	000			0	0	0	0	*
Access	R			RW	R	R	R	*

Name	Bits	Description	Settings									
BSTAPEN	0	Boundary Scan TAP enable. When enabled, the JTAG port is connected to the Boundary Scan TAP. Otherwise, the port is connected to the ARM ICE function. This bit is reset by the POR. Reset value and access depend on the part number: <table border="1" style="margin-left: 20px;"> <tr> <th>Part</th><th>Reset Value</th><th>Access</th></tr> <tr> <td>MAX32550-LNS+</td><td>1</td><td>RW</td></tr> <tr> <td>MAX32550-LNJ+</td><td>0</td><td>R</td></tr> </table>	Part	Reset Value	Access	MAX32550-LNS+	1	RW	MAX32550-LNJ+	0	R	0: Boundary Scan TAP port disabled. 1: Boundary Scan TAP port enabled.
Part	Reset Value	Access										
MAX32550-LNS+	1	RW										
MAX32550-LNJ+	0	R										
RFU	2:1	Reserved	N/A									
ONVMAIN	3	Battery supply source flag	0: VBAT is selected to supply battery supply. 1: VMAIN is selected to supply battery supply.									
FLASH_PAGE_FLIP	4	Flips the Flash bottom and top halves. Initiating a flash page flip will cause a flush of both the data buffer on the DCODE bus and the internal instruction buffer. This bit is controlled by ROM and user should not change the state of this bit during the normal operation. It can be used as a flag to identify the logical mapping of the flash.	0: Physical layout matches logical layout. 1: Bottom half mapped to logical top half and vice versa									
RFU	12:5	Reserved	N/A									
CCHK	13	Compute ROM Checksum. This bit is self-cleared when calculation is completed. Once set, software clearing this bit is ignored and the bit will remain set until the operation is completed.	0: No operation. 1: Starts ROM checksum calculation.									
CHKRD	14	ROM Checksum Ready. This bit is self-cleared when a new checksum calculation is started (CCHK is set).	0: Result not ready. 1: Checksum complete and result ready.									
CHKRES	15	ROM Checksum Result. This bit is only valid when CHKRD=1.	0: Pass. 1: Fail.									
RFU	31:16	Reserved.	N/A									



4.3.2 Reset Register (RSTR, Offset 0x0004)

Table 4-5. RSTR (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	SYSTEM	PRST	SRST	RFU[1:0]		DAC	ADC	RFU
Reset	0	0	0	00		0	0	0
Access	RS	RS	RS	R		RS	RS	R
Position	23	22	21	20	19	18	17	16
Field	USB	LCD/TFT	KBD	SC	MAGDSP	CRYPTO	RTC	I2C
Reset	0	0	0	0	0	0	0	0
Access	RS	RS	RS	RS	RS	RS	RS	RS
Position	15	14	13	12	11	10	9	8
Field	SPI2	SPI1	SPI0	UART1*	UART0	TIMER5	TIMER4	TIMER3
Reset	0	0	0	0	0	0	0	0
Access	RS	RS	RS	RS	RS	RS	RS	RS
Position	7	6	5	4	3	2	1	0
Field	TIMER2	TIMER1	TIMER0	GPIO2	GPIO1	GPIO0	WDT	DMA
Reset	0	0	0	0	0	0	0	0
Access	RS	RS	RS	RS	RS	RS	RS	RS

Setting a bit to 1 in this register causes a reset of the corresponding peripheral or set of peripherals. Once the reset has completed, the bit will be automatically cleared to 0 by hardware. Software writes of 0 to bits in this register have no effect.

Name	Bits	Description
DMA	0	DMA Reset.
WDT	1	Watchdog Timer Reset.
GPIO0	2	GPIO0 Reset. Setting this bit to 1 resets GPIO0 pins to their default states.
GPIO1	3	GPIO1 Reset. Setting this bit to 1 resets GPIO1 pins to their default states.
GPIO2	4	GPIO2 Reset. Setting this bit to 1 resets GPIO2 pins to their default states.
TIMER0	5	Timer0 Reset. Setting this bit to 1 resets Timer 0 blocks.
TIMER1	6	Timer1 Reset. Setting this bit to 1 resets Timer 1 blocks.
TIMER2	7	Timer2 Reset. Setting this bit to 1 resets Timer 2 blocks.
TIMER3	8	Timer3 Reset. Setting this bit to 1 resets Timer 3 blocks.
TIMER4	9	Timer4 Reset. Setting this bit to 1 resets Timer 4 blocks.
TIMER5	10	Timer5 Reset. Setting this bit to 1 resets Timer 5 blocks.
UART0*	11	UART0 Reset. Setting this bit to 1 resets all UART 0 blocks.
UART1**	12	UART1 Reset. Setting this bit to 1 resets all UART1 blocks.
SPI0	13	SPI0 Reset. Setting this bit to 1 resets all SPI 0 blocks.
SPI1	14	SPI1 Reset. Setting this bit to 1 resets all SPI 1 blocks.
SPI2	15	SPI2 Reset. Setting this bit to 1 resets all SPI 2 blocks.
I2C	16	I2C Reset.
RTC	17	Real Time Clock Reset.
CRYPTO	18	Cryptographic Reset. Setting this bit to 1 resets the AES block, the SHA block and the DES block.
MAGDSP	19	Magnetic Strip Reader
SC	20	Smart Card Reader Reset.
KBD	21	Secure Keypad Reset.

Name	Bits	Description
TFT/LCD	22	TFT/LCD Reset. Setting this bit to 1 resets both the TFT and mono LCD blocks.
USB	23	USB Reset. Setting this bit resets both USB blocks.
RFU	24	Reserved
ADC	25	Analog to Digital Reset.
DAC	26	Digital to Analog Reset
RFU	28:27	Reserved.
SRST	29	Soft Reset. Setting this bit to 1 resets everything except the CPU and the watchdog timer.
PRST	30	Peripheral Reset. Setting this bit to 1 resets all peripherals. The CPU core, the watchdog timer, and all GPIO pins are unaffected by this reset.
SYSTEM	31	System Reset. Setting this bit to 1 resets the CPU core and all peripherals, including the watchdog timer.

* This bit also resets UART1 on revision A silicon.

** Not applicable on revision A silicon.



4.3.3 Clock Control Register (CLKCN, Offset 0x0008)

Table 4-6. CLKCN (Offset 0x0008)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU	RFU	CKRDY	RFU	CLKSEL[2:0]		PSC2	
Reset	0	0	0	0	000		0	
Access	R	R	R	R	RW		RW	
Position	7	6	5	4	3	2	1	0
Field	PSC1	PSC0	RFU[5:0]					
Reset	0	0	00 1000					
Access	RW	RW	R					

Name	Bits	Description	Settings
RFU	5:0	Reserved	N/A
PSC	8:6	Prescaler Select. This 3 bit field sets the system operating frequency by controlling the prescaler that divides the output of the PLL0.	000: /1 (108MHz) 001: /2 (54MHz) 010: /4 (27MHz) 011: /8 (13.5MHz) 100: /16 (6.75MHz) 101: /32 (3.375MHz) 110: /64 (1.6875MHz) 111: /128 (0.84375MHz) Output clock shown assume an input PLL clock of 12MHz.
CLKSEL	11:9	Clock Source Select. This 3 bit field selects the source for the system clock.	000: Crypto oscillator 001: PLL Output 010: HFXIN 011: Nano-ring Others: Reserved. Use Crypto oscillator.
RFU	12	Reserved.	N/A
CKRDY	13	Clock Ready. This read only bit reflects whether the currently selected system clock source is running.	0: Switchover from the previously active clock source to the new clock source (as selected by CLKSEL) has not yet occurred. 1: System clock running from CLKSEL clock source.
RFU	31:14	Reserved.	N/A



4.3.4 Power Management Register (PM, Offset 0x000C)

Table 4-7. PM (Offset 0x000C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[5:0]						CM3WICACK	CM3PMUEN
Reset	00 0000						0	0
Access	R						R	RW
Position	15	14	13	12	11	10	9	8
Field	RFU	COPD	PLL01PD	OSCPD	RFU[3:0]			
Reset	0	0	0	0	0000			
Access	R	RW	RW	RW	R			
Position	7	6	5	4	3	2	1	0
Field	SCWKEN	USBWKEN	RTCWKEN	GPIOWKEN	RFU	RFU	MODE1	MODE0
Reset	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	R	R	RW	RW

Name	Bits	Description	Settings
MODE	1:0	Operating Mode. This two bit field selects the current operating mode for the device. Note that code execution only occurs during ACTIVE mode.	00: ACTIVE. 01: IDLE. 10: STANDBY. 11: SHUTDOWN.
RFU	3:2	Reserved.	N/A
GPIOWKEN	4	GPIO Wake Up Enable. This bit enables all GPIO pins as potential wakeup sources. Any GPIO configured for wakeup is capable of causing an exit from IDLE or STANDBY modes when this bit is set.	0: Wakeup from GPIO pin activity is disabled. 1: Wakeup is enabled. Appropriate signals on GPIO pins configured to act as wakeup sources will cause the device to wake up from IDLE or STANDBY modes.
RTCWKEN	5	RTC Alarm Wake Up Enable. This bit enables RTC alarm as wakeup source. If enabled, the desired RTC alarm must be configured via the RTC control registers.	0: Wakeup from an RTC alarm is disabled. 1: Wakeup is enabled. An RTC alarm will wake the device from IDLE or STANDBY modes.
USBWKEN	6	USB Wake Up Enable. This bit enables USB activity as wakeup source.	0: Wakeup disabled. 1: Wakeup enabled. Activity on USB will wake the device from IDLE or STANDBY modes.
SCWKEN	7	Smart Card Wake Up Enable. This bit enables Smart Card activity as a wakeup source	0: Wakeup disabled. 1: Wakeup enabled.
RFU	11:8	Reserved.	N/A
OSCPD	12	External Oscillator Power Down. Setting this bit prior to entering the STANDBY mode, powers down the External Oscillator circuit.	0: The External Oscillator does not power down upon entering the STANDBY mode. 1: The External Oscillator powers down upon entering the STANDBY mode.
PLL01PD	13	PLL0/1 Power Down. This bit selects PLL power state in STANDBY mode.	0: PLL0/1 active. 1: PLL0/1 in standby mode.
COPD	14	Crypto Oscillator Power Down. This bit selects crypto oscillator power state in STANDBY mode.	0: Crypto oscillator active. 1: Crypto oscillator in standby mode.
RFU	15	Reserved	N/A



Name	Bits	Description	Settings
CM3PMUEN	16	CM3 Power Management Enable. Setting this bit will allow the CM3_DEEPSLEEP or CM3_SLEEPING state to put the device into STANDBY mode.	0: Device stays in current power mode despite CM3_SLEEPING or CM3_DEEPSLEEP being asserted. 1: CM3_SLEEPING or CM3_DEEPSLEEP causes the device to enter STANDBY mode
CM3WICACK	17	CM3 WIC Acknowledge. This status bit reflects the state of the CM3 WIC acknowledge to PMU request	
RFU	31:18	Reserved.	N/A



4.3.5 PLL0 Control Register (PLL0CN, Offset 0x0010)

Table 4-8. PLL0CN (Offset 0x0010)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[5:0]						PLL0LOCK	PLL0EN
Reset	0000 00						0	0
Access	R						R	RW

Name	Bits	Description	Settings
PLL0EN	0	PLL0 Enable. This bit enables system PLL0 for clock multiplication.	0: PLL0 disabled. 1: PLL0 enabled.
PLL0LOCK	1	PLL0 Lock. This bit is set by hardware when the PLL0 is locked to the input clock. It is cleared when the PLL0 is disabled, lock is not yet achieved or when the lock is lost.	0: PLL0 is not locked or is disabled. 1: PLL0 is locked.
RFU	31:2	Reserved.	N/A



4.3.6 PLL1 Control Register (PLL1CN, Offset 0x0014)

Table 4-9. PLL1CN (Offset 0x0014)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[5:0]						PLL1LOCK	PLL1EN
Reset	0000 00						0	0
Access	R						R	RW

Name	Bits	Description	Settings
PLL1EN	0	PLL1 Enable. This bit enables system PLL1 for clock multiplication.	0: PLL1 disabled. 1: PLL1 enabled.
PLL1LOCK	1	PLL1 Lock. This bit is set by hardware when the PLL1 is locked to the input clock. It is cleared when the PLL1 is disabled, lock is not yet achieved or when the lock is lost.	0: PLL1 is not locked or is disabled. 1: PLL1 is locked.
RFU	31:2	Reserved.	N/A



4.3.7 Peripheral Clock Divider Register (PCKDIV, Offset 0x0018)

Table 4-10. PCKDIV (Offset 0x0018)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[4:0]					PCF		
Reset	0000 0					011		
Access	R					RW		

Name	Bits	Description	Settings
PCF	2:0	These bits determine the clock frequency for the UART, I2C and Key Pad peripherals. These peripherals have an adaptive clock generator that dynamically adjusts the peripheral frequency based on the main system bus frequency. These bits are dynamically updated when the PLL0 is selected as the system clock source and are set by hardware.	These bits are set by hardware, any write to this register must preserve these bits values.
RFU	31:3	Reserved	N/A



4.3.8 Peripheral Clock Disable Register (PERCKCN, Offset 0x0024)

Table 4-11. PERCKCN (Offset 0x0024)

Position	31	30	29	28	27	26	25	24
Field	RFU[6:0]						DACD	
Reset	0000 000						0	
Access	R						RW	
Position	23	22	21	20	19	18	17	16
Field	ADCD	KBDD	MLCDD	T5D	T4D	T3D	T2D	T1D
Reset	0	0	0	0	0	0	0	0
Access	R	RW	RW	RW	RW	RW	RW	R
Position	15	14	13	12	11	10	9	8
Field	T0D	CRYPTOD	I2CD	SCD	MAGDSPD	UART1D	UART0D	SPI2D
Reset	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW
Position	7	6	5	4	3	2	1	0
Field	SPI1D	SPI0D	DMAD	CLCDD	USBD	GPIO2D	GPIO1D	GPIO0D
Reset	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

Setting a bit to 1 in this register disables the clock for the associated peripheral(s). If the bit is cleared to 0 (which is the default state), the peripheral operates normally.

Name	Bits	Description
GPIO0D	0	GPIO0 Disable
GPIO1D	1	GPIO1 Disable.
GPIO2D	2	GPIO2 Disable.
USBD	3	USB Disable.
CLCD	4	Color LCD controller Disable
DMA	5	DMA Disable
SPI0D	6	SPI 0 Disable.
SPI1D	7	SPI 1 Disable.
SPI2D	8	SPI 2 Disable.
UART0D	9	UART 0 Disable.
UART1D	10	UART 1 Disable.
MAGDSPD	11	Magnetic Stripe DSP disable
SCD	12	Smart Card Disable
I2CD	13	I2C Disable
CRYPTOD	14	Crypto Disable.
T0D	15	Timer 0 Disable.
T1D	16	Timer 1 Disable.
T2D	17	Timer 2 Disable.
T3D	18	Timer 3 Disable.
T4D	19	Timer 4 Disable.
T5D	20	Timer 5 Disable.
MLCDD	21	Mono LCD Disable.
KBDD	22	Keypad Disable.
ADCD	23	ADC Disable



Name	Bits	Description
DACD	24	DAC Disable.
RFU	31:25	Reserved.

E/UG25H05/36488764



4.3.9 Memory Zeroize Control Register (MEMZCN, Offset 0x002C)

Table 4-12. MEMZCN (Offset 0x002C)

Position	31	30	29	28	27	26	25	24
Field				RFU[7:0]				
Reset				0000 0000				
Access				R				
Position	23	22	21	20	19	18	17	16
Field				RFU[7:0]				
Reset				0000 0000				
Access				R				
Position	15	14	13	12	11	10	9	8
Field				RFU[7:0]				
Reset				0000 0000				
Access				R				
Position	7	6	5	4	3	2	1	0
Field	RFU[1:0]	CRYPTO	RFU	ICACHE	RFU	SRAM1	SRAM0	
Reset	00	0	0	0	0	0	0	
Access	R	RW	R	RW	R	RW	RW	

Setting any bit to 1 will initiate zeroize operation for the associated memory. This bit is automatically cleared by hardware on zeroization completion.

Name	Bits	Description
SRAM0	0	System RAM block 0
SRAM1	1	System RAM block 1
RFU	2	Reserved
ICACHE	3	Instruction Cache
RFU	4	Reserved
CRYPTO	5	Crypto (MAA) Memory
RFU	31:6	Reserved.



4.3.10 Smart Card Clock Control Register (SCCK, Offset 0x0034)

System clock source can be any one of the defined clock sources.

Table 4-13. SCCK (Offset 0x0034)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[1:0]		SC0FRQ[5:0]					
Reset	00		01 0100					
Access	R		RW					

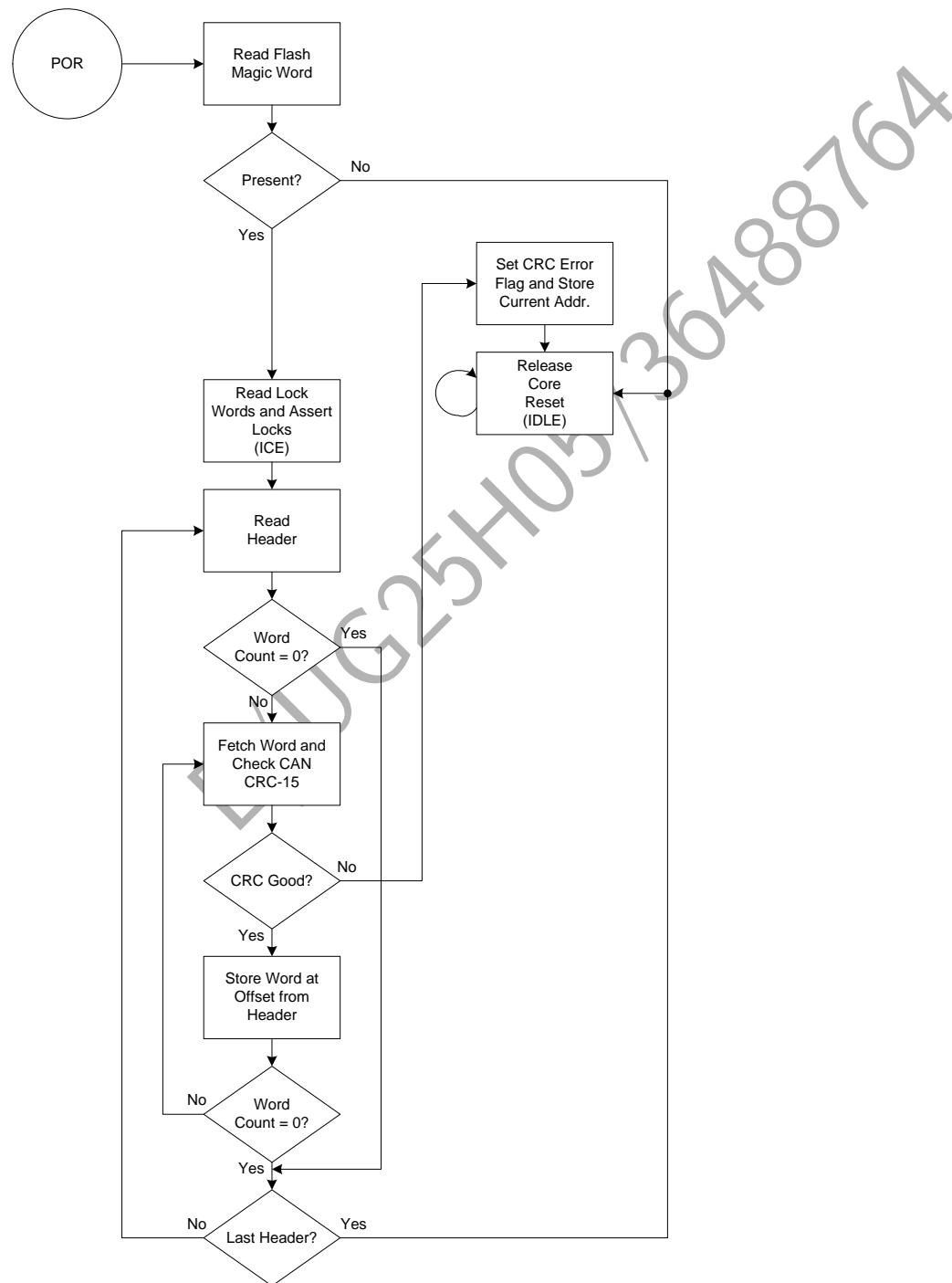
Name	Bits	Description	Settings
SC0FRQ	5:0	Smart Card 0 Clock Frequency. These bits define Smart Card 0 clock frequency. $f_{SC0} = f_{PLL1} / (SC0FRQ)$	Output in Hz: 00: Invalid 01: 96,000,000 02: 48,000,000 03: 32,000,000 04: 24,000,000 30: 32,00,000 31: 3,096,774 32: 3,000,000 33: 2,909,091 34: 2,823,529 59: 1,627,119 60: 1,600,000 61: 1,573,770 62: 1,548,387 63: 1,523,810 The above timing assumes a 12MHz input clock to the PLL1 and x8 multiplication.
RFU	31:6	Reserved.	N/A

Where f_{PLL1} is derived from the frequency of PLL1

5.0 System Initialization

Following initial power-up, the device performs certain actions to initialize its starting state before it begins to execute application code. These actions include retrieving information from nonvolatile memory to configure the operation of the device. This non-volatile data (NVD) includes analog trim information, configuration and locking bits, and may also include customer- and application-specific information such as serial numbers and cryptographic keys. The system initialization start-up flow is shown in Figure 5-1.

Figure 5-1. System Initialization Block Start-up Flow



5.1 System Initialization Block (SIB)

The System Initialization Block (or SIB) is a functional block on the device which is responsible for performing initialization functions following initial power-up. The SIB is activated automatically following a power-on reset, and it holds the main CPU in a reset state while it performs its initialization functions.

While the CPU is held in a reset state, the SIB takes control of the AHB bus and reads trim and configuration data from the on-chip one-time-programmable (OTP) memory. This memory is loaded with appropriate values during the manufacturing and test cycle. The SIB reads values from the OTP block and loads trim registers and other configuration registers as appropriate. Once the initialization process is complete, the SIB disengages from the AHB bus and releases the CPU core from reset, allowing it to begin code execution.

5.2 System Initialization Registers (SIR)

In general, the SIB block performs initialization of the device automatically, and the user does not need to be concerned with the precise details of which values are located in the OTP memory block and how they are interpreted. The only potential issue comes if the OTP values have not been loaded, or if the OTP values are misread or somehow corrupted. In this case, certain registers in the SIR register area may be checked by application code to verify the successful completion of the system initialization process.

Address assignments for the System Initialization registers are outlined in Table 5-1. Reserved register bits should only be written as 0.

Table 5-1. System Initialization Register Addresses (Base ADDR = 0x4000 0400)

Offset	Access	Register	Description
0x0000	R	SISTAT	System Initialization Status Register
0x0004	R	ERRADDR	System Initialization Error Address Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



5.2.1 System Initialization Status Register (SISTAT, Offset 0x0000)

Table 5-2. SISTAT (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[5:0]						CRCERR	MAGIC
Reset	0000 00						0	0
Access	R						R	R

Name	Bits	Description	Settings
MAGIC	0	Read-only status bit set by the system initialization block following power-up.	0: Magic word was not set (OTP has not been initialized properly). 1: Magic word was set – OTP contains valid settings.
CRCERR	1	Read-only status bit set by the system initialization block following power-up.	0: No CRC errors occurred during the read of the OTP memory block 1: A CRC error occurred while reading the OTP. The address of the failure location in the OTP memory is stored in the ERRADDR register.
RFU	31:2	Reserved	N/A



5.2.2 System Initialization Error Address Register (ERRADDR, Offset 0x0004)

Table 5-3. ERRADDR (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	ERRADDR[31:24]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	ERRADDR[23:16]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	ERRADDR[15:8]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	ERRADDR[7:0]							
Reset	0000 0000							
Access	R							

Name	Bits	Description	Settings
ERRADDR	31:0	Read-only field set by the SIB block if a CRC error occurs during the read of the OTP memory.	Contains the failing address in OTP memory (when CRCERR equals 1).



6.0 Interrupts

Interrupt Number	Offset	Interrupt Source	Interrupt Number	Offset	Interrupt Source
16	0040	Power Fail Interrupt	41	00A4	GPIO 1
17	0044	Watchdog Timer	42	00A8	GPIO 2
18	0048	USB	43	00AC	Crypto Engine
19	004C	RTC	44	00B0	DMA 0
20	0050	TRNG	45	00B4	DMA 1
21	0054	Timer 0	46	00B8	DMA 2
22	0058	Timer 1	47	00BC	DMA 3
23	005C	Timer 2			
24	0060	Timer 3			
25	0064	Timer 4			
26	0068	Timer 5			
27	006C	Smart Card			
28	0070	Mono LCD/TFT			
29	0074	I2C			
30	0078	UART 0			
31	007C	UART 1			
32	0080	SPI 0			
33	0084	SPI 1			
34	0088	SPI 2			
35	008C	Secure Keypad			
36	0090	ADC			
37	0094	DAC			
38	0098	MSR DSP Done			
39	009C	Flash Controller			
40	00A0	GPIO 0			

Note: The interrupt vector table should be 256-byte (64 word) aligned otherwise an incorrect handler may be vectored to upon an interrupt. Refer to the ARM Cortex-M3 documentation for further information on interrupt vector table alignment. For MAX32550, there are 32 interrupts and 16 Cortex-M3 exceptions. The total table size is 48, and should only need 64-word alignment.

6.1 Nested Vectored Interrupt Controller

Refer to ARM Cortex-M3 Technical Reference Manual for Nested Vectored Interrupt Controller available at www.arm.com

7.0 Instruction Cache Controller

Please refer to the Instruction Cache Controller register section.

7.1 Instruction Cache Controller Registers

Address assignments for the Instruction Cache Controller Registers are outlined in Table 7-1.

Table 7-1. Instruction Cache Controller Register Address (Base ADDR = 0x4002_A000)

Offset	Access	Register	Description
0x0000	R	CACHE_ID	Cache ID Register
0x0004	RW	MEMCFG	Memory Configuration Register
0x0008 ... 0x00FC	R	RFU	
0x0100	RW	CACHE_CTRL	Cache Control and Status Register
0x0104 ... 0x06FC	R	RFU	
0x0700	R	INVALIDATE	Invalidate All Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



7.1.1 Cache ID Register (CACHE_ID, Offset 0x0000)

Table 7-2. CACHE_ID (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	CCHID[5:0]						PARTNUM[3:2]	
Reset	00 0000						00	
Access	R						R	
Position	7	6	5	4	3	2	1	0
Field	PARTNUM[1:0]		RELENUM[5:0]					
Reset	00		00 0000					
Access	R		R					

Name	Bits	Description	Settings
RELENUM	5:0	Release Number. Identifies the RTL release version.	
PARTNUM	9:6	Part Number. This field reflects the value of C_ID_PART_NUMBER configuration parameter.	
CCHID	15:10	Cache ID. This field reflects the value of the C_ID_CACHEID configuration parameter.	
RFU	31:16	Reserved	N/A



7.1.2 Memory Configuration Register (MEMCFG, Offset 0x0004)

Table 7-3. MEMCFG (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	MEMSZ[15:8]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	MEMSZ[7:0]							
Reset	0000 1000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	CCHSZ[15:8]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	CCHSZ[7:0]							
Reset	0000 1000							
Access	R							
Name	Bits	Description					Settings	
CCHSZ	15:0	Cache Size. Indicates total size in Kbytes of cache						
MEMSZ	31:16	Main Memory Size. Indicates the total size, in units of 128 Kbytes, of code memory accessible to the cache controller						



7.1.3 Cache Control and Status Register (CACHE_CTRL, Offset 0x0100)

Table 7-4. CACHE_CTRL (Offset 0x0100)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[6:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[6:0]							
Reset	0000 0000							
Access	R							

Name	Bits	Description	Settings
CACHE_EN	0	Cache Enable. Controls whether the cache is bypassed or is in use. Changing the state of this bit will cause the instruction cache to be flushed and its contents invalidated.	0: Cache Bypassed. Instruction data is stored in the line fill buffer but is not written to main cache memory array. 1: Cache enabled
RFU	15:1	Reserved	N/A
CACHE_RDY	16	Cache Ready flag. Cleared by hardware when at any time the cache as a whole is invalidated (including a system reset). When this bit is 0, the cache is effectively in bypass mode (instruction fetches will come from main memory or from the line fill buffer). Set by hardware when the invalidate operation is complete and the cache is ready.	
RFU	31:17	Reserved	N/A



7.1.4 Invalidate All (INVALIDATE, Offset 0x0700)

Table 7-5. INVALIDATE (Offset 0x0700)

Position	31	30	29	28	27	26	25	24
Field	IA[31:24]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	IA[23:16]							
Reset	000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	IA[15:8]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	IA[7:0]							
Reset	0000 0000							
Access	R							

Name	Bits	Description	Settings
IA	31:0	Invalidate All Cache Contents. Any time this register location is written (regardless of the data value), the cache controller immediately begins invalidating the entire contents of the cache memory. The cache will be in bypass mode until the invalidate operation is complete. System software can examine the Cache Ready bit (CACHE_CTRL.CACHE_RDY) to determine when the invalidate operation is complete. Note that it is not necessary to disable the cache controller prior to beginning this operation. Reads from this register always return 0.	

8.0 Embedded ICE and Debug

The ARM Cortex-M3 contains a JTAG interface and an embedded In Circuit Emulator (ICE) interface. Refer to ARM Cortex-M3 Technical Reference Manual for Debug available at www.arm.com

E/UG25H05/36488764

9.0 General-Purpose Input/Output (GPIO) Pins

Individual I/O controls are provided for all pins which have GPIO capability. Most of these pins are shared with one or more peripheral functions and can be used as GPIO when the peripheral is not in use by the application. The multiplexing between peripheral and GPIO mode is performed by software on a pin-by-pin basis and can be set either statically (once at the beginning of the application) or dynamically (switching between modes during application execution) as required by the system design.

Features of GPIO pins on the device are listed below.

- Up to 70 GPIO pins are available
- Each GPIO pin may be configured individually to operate in input, output or I/O mode
- Pins may be switched between GPIO and peripheral (primary/secondary) functions on a pin-by-pin basis
- All GPIO pins have pull-up/pull-down capability
- Any GPIO pin can be configured to generate an interrupt
- GPIO pins can be configured to generate interrupts for a high or low level, or on a rising or falling edge, or on both rising and falling edges
- Any GPIO pin can be configured to generate a wake condition to the PMU (power management unit) on a rising or falling edge

9.1 GPIO Port Register Blocks

GPIO pins are organized into 3 ports. The input/output state and configuration modes for all pins in a given port are controlled by a block of registers assigned to that port. Each port contains an identical set of GPIO registers which are used to control the following functions:

- Multiplexing between GPIO mode and primary/secondary functions (GPIO_EN, GPIO_EN1)
- Setting pins to input or output mode (GPIO_OUT_EN)
- Setting output mode to drive high or drive low (GPIO_OUT, GPIO_OUT_SET, GPIO_OUT_CLR)
- Reading the input state of pin(s) (GPIO_IN)
- Interrupt enable/disable and mode configuration (GPIO_INT_EN, GPIO_INT_EN_SET, GPIO_INT_EN_CLR, GPIO_INT_MOD, GPIO_INT_POL, GPIO_INT_STAT, GPIO_INT_CLR, GPIO_INT_DUAL_EDGE)
- Enabling/disabling wakeup mode (GPIO_WAKE_EN, also uses GPIO_INT_POL and GPIO_INT_CLR)
- Setting optional pull-up or pull-down when the GPIO is in input mode (GPIO_PAD_CFG1, GPIO_PAD_CFG2)

The addresses for each register block are listed in Table 2-1. Internal Memory Region.

9.2 GPIO Port Initialization

During the Power-on-Reset (POR) all GPIOs are in input mode with weak pull-ups. Following the reset or initial power-up, all GPIO pins are reset and default to the following conditions:

- GPIO mode is enabled (no peripheral primary or secondary functions are selected) with the exception of the JTAG pins on Port 2.
- Output mode is disabled
- Interrupts are disabled
- PMU wake mode is disabled
- Weak pull-up/pull-down mode is disable

During the execution of the Boot ROM on B1 and later silicon the state will remain the same. On A3 silicon, however, the state of the GPIOs are changed as follows:

- GPIO0[15:0]: Input with weak pull-up
- GPIO0[23:16]: Output low
- GPIO1[4:0]: Input with weak pull-up
- GPIO1[11:8]: Output low



9.3 Detailed Description

The following sections provide more details on configuring GPIO operating modes.

9.3.1 Configuring GPIO/Peripheral Modes

GPIO pins typically share their functionality with one or two other on-chip peripheral functions. If these alternate functions are fully or partially unused, the user is free to assign the pins for use as GPIOs.

By default, the pins are configured to operate in GPIO mode. If an alternate function will be used, the pins should be configured to use the function by setting the appropriate GPIO_EN bits to 0, and also setting the GPIO_EN1 bits if necessary to select between primary and secondary function assignments.

Setting the GPIO_EN bit for a pin to 1 (GPIO mode active) will enable software control of the pin using the bits in the appropriate GPIO_OUT_EN and GPIO_OUT registers. Also, a default input value will be driven to the internal input of the primary and/or secondary functions that would be connected to that GPIO pin.

Switching pins from GPIO mode to peripheral mode (by clearing the appropriate GPIO_EN and, if necessary, setting GPIO_EN1 bits) will automatically configure the pins to input and/or output mode as needed as required by their associated peripheral functions. The GPIO_OUT_EN register does not need to be modified to match the peripheral input and output connections; this is handled directly by hardware. However, if the peripheral requires that certain pins be configured to operate with weak pull-up or weak pull-down mode enabled (refer to the peripheral documentation), then this must be handled in software (by setting GPIO_PAD_CFG1 and GPIO_PAD_CFG2) as it will not be done automatically.

The input function for GPIO pins (level read from the GPIO_IN register) as well as the interrupt and/or PMU wake functions will operate whether the pins are configured to operate in alternate function mode (GPIO_EN=0) or GPIO mode (GPIO_EN=1).

For more details about GPIO pin assignments, refer to the datasheet.

Note: The JTAG port (found on Port 2 of JTAG-enabled parts) is an exception to these rules, and will always be enabled after a Power-On-Reset. To use GPIO ports on the pins associated with JTAG port on JTAG-enabled parts, user has to set GPIO_EN.1 through GPIO_EN.5 to one.

On the JTAG-disabled parts on the other hand, user can still use the JTAG port for boundary scan testing, however, user cannot use JTAG to access the debugger. To use GPIO ports on the pins associated with JTAG port on JTAG-disabled parts, user has to set GPIO_EN.1 through GPIO_EN.5 of port 2 to one and clear SCON.BSTAPEN bit to zero.

9.3.2 Configuring GPIO (External) Interrupts

Below is the suggested procedure to use when initially configuring GPIO interrupts. This can be done for any number of GPIO bits. The GPIO interrupt function can be used regardless of the GPIO_EN and GPIO_OUT_EN settings.

1. Disable interrupts by clearing the GPIO_INT_EN register. This will prevent any new interrupts from triggering but will not clear previously triggered (pending) interrupts. Bits in GPIO_INT_EN can also be cleared by writing to the GPIO_INT_EN_CLR register.
2. Clear pending interrupts by writing to the GPIO_INT_CLR register as needed.
3. Write to GPIO_INT_MOD to select either level or edge triggered interrupts. For edge triggered interrupts, the interrupt triggers on a transition from low to high and/or high to low (as selected by GPIO_INT_POL or GPIO_INT_DUAL_EDGE). For level triggered interrupts, the interrupt triggers on a continuous high or low level (selected by GPIO_INT_POL) with a duration of approximately 100ns or three PCLK cycles regardless of GPIO_INT_DUAL_EDGE register value.



4. If edge triggered interrupt mode is selected, the interrupt can be set to trigger on either edge (rising or falling) by setting the bit(s) in the GPIO_INT_DUAL_EDGE register to 1. The GPIO_INT_POL value is ignored in this case.
5. Write to GPIO_INT_EN to re-enable the interrupt(s) that will be used. Once the GPIO_INT_EN bit(s) are set, interrupts may trigger in the usual manner (if enabled by the interrupt controller). The GPIO_INT_EN bits may also be set by writing to the GPIO_INT_EN_SET register.
6. Configure the Interrupt Controller to enable GPIO interrupts. (The user may want to do this step prior to setting the GPIO_INT_EN bits).

9.3.3 Handling Interrupts from GPIO Pins

GPIO interrupts can be handled/processed using the following procedure.

1. Read GPIO_INT_STAT to determine the GPIO interrupt source.
2. Perform interrupt tasks associated with the interrupt source pin (application defined).
3. Clear the interrupt flag (in GPIO_INT_STAT) by writing to the GPIO_INT_CLR register. This will also clear and re-arm the rising and falling edge triggers.
4. Signal an end-of-interrupt to the interrupt controller by writing to the End-of-Interrupt register.
5. Return from the ISR.

9.3.4 Using the GPIO Wake Function

The suggested procedure below can be used to enable PMU wakeup using a GPIO interrupt trigger. Like the standard interrupt function, this is independent of the GPIO_EN and GPIO_OUT_EN settings.

1. Set the polarity (rising or falling edge) by writing to the GPIO_INT_POL register. The wake function relies on the rising and falling edge detectors, which can operate asynchronously and do not require an active clock. The dual edge mode (rising and falling edge triggers) can also be used if desired. For Level detection, the GPIO input will be sampled by PCLK and for this reason, the level triggered mode cannot be used for PMU wakeup.
2. Clear pending interrupt flags by writing to GPIO_INT_CLR.
3. Activate the GPIO wake function by writing 1 to the appropriate bits in the GPIO_WAKE_EN register.
4. Configure the PMU to use the GPIO as a wakeup source by writing to the appropriate Global Control register (GCR).



9.4 GPIO Registers

Address assignments for GPIO registers within each Port block are outlined in Table 9-1. Reserved register bits should only be written as 0.

Table 9-1. GPIO Register Addresses (for each port)

Offset	Access	Register	Description
0x0000	RW	GPIO_EN	GPIO Function Enable Register
0x000C	RW	GPIO_OUT_EN	GPIO Output Enable Register
0x0018	RW	GPIO_OUT	GPIO Output Register
0x001C	RS	GPIO_OUT_SET	GPIO Output Set
0x0020	RS	GPIO_OUT_CLR	GPIO Output Clear
0x0024	R	GPIO_IN	GPIO Input Register
0x0028	RW	GPIO_INT_MOD	GPIO Interrupt Mode Register
0x002C	RW	GPIO_INT_POL	GPIO Interrupt Polarity Register
0x0034	RW	GPIO_INT_EN	GPIO Interrupt Enable Register
0x0038	RS	GPIO_INT_EN_SET	GPIO Interrupt Enable Set
0x003C	RS	GPIO_INT_EN_CLR	GPIO Interrupt Enable Clear
0x0040	R	GPIO_INT_STAT	GPIO Interrupt Status Register
0x0048	RS	GPIO_INT_CLR	GPIO Interrupt Status Clear
0x004C	RW	GPIO_WAKE_EN	GPIO Wake Enable Register
0x005C	RW	GPIO_INT_DUAL_EDGE	GPIO Interrupt Dual Edge Mode Register
0x0060	RW	GPIO_PAD_CFG1	GPIO Input Mode Config 1
0x0064	RW	GPIO_PAD_CFG2	GPIO Input Mode Config 2
0x0068	RW	GPIO_EN1	GPIO Alternate Function Enable Register
0x00B0	RW	GPIO_DS	GPIO Drive Strength Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



9.4.1 GPIO Function Enable Register (GPIO_EN, Offset 0x0000)

Table 9-2. GPIO_EN (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	GPIO_EN[31:24]							
Reset	1111 1111							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	GPIO_EN[23:16]							
Reset	1111 1111							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	GPIO_EN[15:8]							
Reset	1111 1111							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	GPIO_EN[7:0]							
Reset	1111 1111*							
Access	RW							

Name	Bits	Description	Settings
GPIO_EN	31:0	Each bit controls the GPIO_EN setting for one GPIO pin on the associated port.	<p>0: Alternate function enabled (select between primary/secondary with GPIO_EN1). 1: GPIO function is enabled.</p> <p>Input, interrupt and PMU wake functions can always be used regardless of the setting of this bit. *Note, GPIO_EN.1 to GPIO_EN.5 on port 2 are reset to zero. For more information see the note under 9.3.1.</p>



9.4.2 GPIO Output Enable Register (GPIO_OUT_EN, Offset 0x000C)

Table 9-3. GPIO_OUT_EN (Offset 0x000C)

Position	31	30	29	28	27	26	25	24
Field	GPIO_OUT_EN[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	GPIO_OUT_EN[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	GPIO_OUT_EN[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	GPIO_OUT_EN[7:0]							
Reset	0000 0000*							
Access	RW							

Name	Bits	Description	Settings
GPIO_OUT_EN	31:0	Each bit controls the GPIO_OUT_EN setting for one GPIO pin in the associated port.	<p>0: Disable GPIO output (default mode). The pad state is (high impedance/weak pull-up/weak pull-down) is controlled by the GPIO_PAD_CFG1 and GPIO_PAD_CFG2 bits).</p> <p>1: Enable GPIO output.</p> <p>This bit setting has no effect when GPIO_EN=0.</p> <p>*Note: GPIO_OUT_EN.2 on port 2 resets to 1 (output).</p>



9.4.3 GPIO Output Register (GPIO_OUT, Offset 0x0018)

Table 9-4. GPIO_OUT (Offset 0x0018)

Position	31	30	29	28	27	26	25	24
Field	GPIO_OUT[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	GPIO_OUT[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	GPIO_OUT[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	GPIO_OUT[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
GPIO_OUT	31:0	Each bit controls the GPIO_OUT setting for one pin in the associated port. This register can be written either directly, or by using the GPIO_OUT_SET and GPIO_OUT_CLR registers.	<p>0: Drive logic 0 (low) on GPIO output. 1: Drive logic 1 (high) on GPIO output</p> <p>This setting only takes effect when the GPIO_EN bit is set to 1 and GPIO_OUT_EN is set to 1 for the associated GPIO pin.</p>



9.4.4 GPIO Output Set Register (GPIO_OUT_SET, Offset 0x001C)

Table 9-5. GPIO_OUT_SET (Offset 0x001C)

Position	31	30	29	28	27	26	25	24
Field	GPIO_OUT_SET[31:24]							
Reset	0000 0000							
Access	RS							
Position	23	22	21	20	19	18	17	16
Field	GPIO_OUT_SET[23:16]							
Reset	0000 0000							
Access	RS							
Position	15	14	13	12	11	10	9	8
Field	GPIO_OUT_SET[15:8]							
Reset	0000 0000							
Access	RS							
Position	7	6	5	4	3	2	1	0
Field	GPIO_OUT_SET[7:0]							
Reset	0000 0000							
Access	RS							

Name	Bits	Description	Settings
GPIO_OUT_SET	31:0	Writing a 1 to one or more bits in this register sets the bits in the same positions in GPIO_OUT to '1', without affecting other bits in that register.	Write: 0: No effect 1: Set GPIO_OUT bit in this position to '1' Read: All reads return 0.



9.4.5 GPIO Output Clear Register (GPIO_OUT_CLR, Offset 0x0020)

Table 9-6. GPIO_OUT_CLR (Offset 0x0020)

Position	31	30	29	28	27	26	25	24
Field	GPIO_OUT_CLR[31:24]							
Reset	0000 0000							
Access	RS							
Position	23	22	21	20	19	18	17	16
Field	GPIO_OUT_CLR[23:16]							
Reset	0000 0000							
Access	RS							
Position	15	14	13	12	11	10	9	8
Field	GPIO_OUT_CLR[15:8]							
Reset	0000 0000							
Access	RS							
Position	7	6	5	4	3	2	1	0
Field	GPIO_OUT_CLR[7:0]							
Reset	0000 0000							
Access	RS							

Name	Bits	Description	Settings
GPIO_OUT_CLR	31:0	Writing a 1 to one or more bits in this register clears the bits in the same positions in GPIO_OUT to '0', without affecting other bits in that register.	<p>Write: 0: No effect 1: Clear GPIO_OUT bit in this position to '0'</p> <p>Read: All reads return 0.</p>



9.4.6 GPIO Input Register (GPIO_IN, Offset 0x0024)

Table 9-7. GPIO_IN (Offset 0x0024)

Position	31	30	29	28	27	26	25	24
Field	GPIO_IN[31:24]							
Reset	N/A							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	GPIO_IN[23:16]							
Reset	N/A							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	GPIO_IN[15:8]							
Reset	N/A							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	GPIO_IN[7:0]							
Reset	N/A							
Access	R							

Name	Bits	Description	Settings
GPIO_IN	31:0	Read-only register to read from the logic states of the GPIO pins on this port.	Each bit in this register reads '0' or '1' depending on the logic state currently present on the associated GPIO pin. This register can be read regardless of the configuration/mode of the pin (GPIO input, GPIO output, alternate function, etc.)



9.4.7 GPIO Interrupt Mode Register (GPIO_INT_MOD, Offset 0x0028)

Table 9-8. GPIO_INT_MOD (Offset 0x0028)

Position	31	30	29	28	27	26	25	24
Field	GPIO_INT_MOD[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	GPIO_INT_MOD[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	GPIO_INT_MOD[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	GPIO_INT_MOD[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
GPIO_INT_MOD	31:0	Each bit in this register controls the interrupt mode setting for the associated GPIO pin on this port.	0: Interrupts for this pin are level triggered 1: Interrupts for this pin are edge triggered This setting has no effect unless GPIO_INT_EN is set to 1 for the associated pin(s).



9.4.8 GPIO Interrupt Polarity Register (GPIO_INT_POL, Offset 0x002C)

Table 9-9. GPIO_INT_POL (Offset 0x002C)

Position	31	30	29	28	27	26	25	24
Field	GPIO_INT_POL[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	GPIO_INT_POL[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	GPIO_INT_POL[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	GPIO_INT_POL[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
GPIO_INT_POL	31:0	Each bit in this register controls the interrupt polarity setting for one GPIO pin in the associated port.	<p>0: Interrupts are latched on a falling edge (if GPIO_INT_MOD=1) or low level (GPIO_INT_MOD=0) condition for this pin.</p> <p>1: Interrupts are latched on a rising edge (if GPIO_INT_MOD=1) or high level (GPIO_INT_MOD=0) condition for this pin.</p> <p>This setting has no effect unless GPIO_INT_EN is set to 1 for the associated pin(s).</p> <p>If GPIO_INT_MOD=1 and GPIO_INT_DUAL_EDGE=1 for this pin, this setting has no effect since both rising and falling edges will be used for triggering.</p>



9.4.9 GPIO Interrupt Enable Register (GPIO_INT_EN, Offset 0x0034)

Table 9-10. GPIO_INT_EN (Offset 0x0034)

Position	31	30	29	28	27	26	25	24
Field	GPIO_INT_EN[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	GPIO_INT_EN[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	GPIO_INT_EN[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	GPIO_INT_EN[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
GPIO_INT_EN	31:0	Each bit in this register controls the GPIO interrupt enable for the associated pin on the GPIO port.	0: Interrupts are disabled for this GPIO pin. 1: Interrupts are enabled for this GPIO pin. Clearing this bit to 0 will not clear pending interrupts (GPIO_INT_STAT); these must be cleared separately.



9.4.10 GPIO Interrupt Enable Set Register (GPIO_INT_EN_SET, Offset 0x0038)

Table 9-11. GPIO_INT_EN_SET (Offset 0x0038)

Position	31	30	29	28	27	26	25	24
Field	GPIO_INT_EN_SET[31:24]							
Reset	0000 0000							
Access	RS							
Position	23	22	21	20	19	18	17	16
Field	GPIO_INT_EN_SET[23:16]							
Reset	0000 0000							
Access	RS							
Position	15	14	13	12	11	10	9	8
Field	GPIO_INT_EN_SET[15:8]							
Reset	0000 0000							
Access	RS							
Position	7	6	5	4	3	2	1	0
Field	GPIO_INT_EN_SET[7:0]							
Reset	0000 0000							
Access	RS							

Name	Bits	Description	Settings
GPIO_INT_EN_SET	31:0	Writing a 1 to one or more bits in this register sets the bits in the same positions in GPIO_INT_EN to '1', without affecting other bits in that register.	<p>Write:</p> <p>0: No effect</p> <p>1: Set GPIO_INT_EN bit in this position to '1'</p> <p>Read:</p> <p>All reads return 0.</p>



9.4.11 GPIO Interrupt Enable Clear Register (GPIO_INT_EN_CLR, Offset 0x003C)

Table 9-12. GPIO_INT_EN_CLR (Offset 0x003C)

Position	31	30	29	28	27	26	25	24
Field	GPIO_INT_EN_CLR[31:24]							
Reset	0000 0000							
Access	RS							
Position	23	22	21	20	19	18	17	16
Field	GPIO_INT_EN_CLR[23:16]							
Reset	0000 0000							
Access	RS							
Position	15	14	13	12	11	10	9	8
Field	GPIO_INT_EN_CLR[15:8]							
Reset	0000 0000							
Access	RS							
Position	7	6	5	4	3	2	1	0
Field	GPIO_INT_EN_CLR[7:0]							
Reset	0000 0000							
Access	RS							

Name	Bits	Description	Settings
GPIO_INT_EN_CLR	31:0	Writing a 1 to one or more bits in this register clears the bits in the same positions in GPIO_INT_EN to '0', without affecting other bits in that register.	<p>Write:</p> <p>0: No effect</p> <p>1: Clear GPIO_INT_EN bit in this position to '0'</p> <p>Read:</p> <p>All reads return 0.</p>



9.4.12 GPIO Interrupt Status Register (GPIO_INT_STAT, Offset 0x0040)

Table 9-13. GPIO_INT_STAT (Offset 0x0040)

Position	31	30	29	28	27	26	25	24
Field	GPIO_INT_STAT[31:24]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	GPIO_INT_STAT[23:16]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	GPIO_INT_STAT[15:8]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	GPIO_INT_STAT[7:0]							
Reset	0000 0000							
Access	R							

Name	Bits	Description	Settings
GPIO_INT_STAT	31:0	Each bit in this register contains the pending interrupt status for the associated GPIO pin in this port.	0: No interrupt is pending on this GPIO pin. 1: An interrupt is pending (and has not yet been serviced) on this GPIO pin. This register cannot be written directly; write to GPIO_INT_CLR to clear bits in this register.



9.4.13 GPIO Interrupt Clear Register (GPIO_INT_CLR, Offset = 0x0048)

Table 9-14. GPIO_INT_CLR (Offset 0x0048)

Position	31	30	29	28	27	26	25	24
Field	GPIO_INT_CLR[31:24]							
Reset	0000 0000							
Access	RS							
Position	23	22	21	20	19	18	17	16
Field	GPIO_INT_CLR[23:16]							
Reset	0000 0000							
Access	RS							
Position	15	14	13	12	11	10	9	8
Field	GPIO_INT_CLR[15:8]							
Reset	0000 0000							
Access	RS							
Position	7	6	5	4	3	2	1	0
Field	GPIO_INT_CLR[7:0]							
Reset	0000 0000							
Access	RS							

Name	Bits	Description	Settings
GPIO_INT_CLR	31:0	Writing a 1 to one or more bits in this register clears the bits in the same positions in GPIO_INT_STAT to '0', without affecting other bits in that register.	<p>Write: 0: No effect 1: Clear GPIO_INT_STAT bit in this position to '0'. This clears the pending interrupt (if any) and resets the edge detectors for this pin.</p> <p>Read: All reads return 0.</p>



9.4.14 GPIO Wake Enable Register (GPIO_WAKE_EN, Offset 0x004C)

Table 9-15. GPIO_WAKE_EN (Offset 0x004C)

Position	31	30	29	28	27	26	25	24
Field	GPIO_WAKE_EN[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	GPIO_WAKE_EN[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	GPIO_WAKE_EN[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	GPIO_WAKE_EN[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
GPIO_WAKE_EN	31:0	Each bit in this register controls the PMU wakeup enable for the associated GPIO pin in this port.	0: PMU wakeup for this GPIO is disabled. 1: PMU wakeup for this GPIO is enabled. The PMU must be configured to accept GPIO as a wakeup source for this to have any effect.



9.4.15 GPIO Interrupt Dual Edge Mode Register (GPIO_INT_DUAL_EDGE, Offset 0x005C)

Table 9-16. GPIO INT DUAL EDGE (Offset 0x005C)

Position	31	30	29	28	27	26	25	24
Field	GPIO_INT_DUAL_EDGE[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	GPIO_INT_DUAL_EDGE[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	GPIO_INT_DUAL_EDGE[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	GPIO_INT_DUAL_EDGE[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
GPIO_INT_DUAL_E DGE	31:0	Each bit in this register selects dual edge mode for the associated GPIO pin in this port.	0: No effect. 1: Dual edge mode is enabled. If edge-triggered interrupts are enabled on this GPIO pin, then both rising and falling edges will trigger interrupts regardless of the GPIO_INT_POL setting.



9.4.16 GPIO Input Configuration Register 1 (GPIO_PAD_CFG1, Offset 0x0060)

Table 9-17. GPIO_PAD_CFG1 (Offset 0x0060)

Position	31	30	29	28	27	26	25	24
Field	GPIO_PAD_CFG1[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	GPIO_PAD_CFG1[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	GPIO_PAD_CFG1[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	GPIO_PAD_CFG1[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	CFG2:CFG1	Effect
GPIO_PAD_CFG1	31:0	The two bits in GPIO_PAD_CFG1 and GPIO_PAD_CFG2 for each GPIO pin work together to determine the pad mode when the GPIO is set to input mode.	00	High impedance
GPIO_PAD_CFG2	31:0		01	Weak pull-up mode
			10	Weak pull-down mode
			11	[Reserved]



9.4.17 GPIO Input Configuration Register 2 (GPIO_PAD_CFG2, Offset 0x0064)

Table 9-18. GPIO_PAD_CFG2 (Offset 0x0064)

Position	31	30	29	28	27	26	25	24
Field	GPIO_PAD_CFG2[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	GPIO_PAD_CFG2[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	GPIO_PAD_CFG2[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	GPIO_PAD_CFG2[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
GPIO_PAD_CFG2	31:0	See description for GPIO_PAD_CFG1.	



9.4.18 GPIO Alternate Function Enable Register (GPIO_EN1, Offset 0x0068)

Table 9-19. GPIO_EN1 (Offset 0x0068)

Position	31	30	29	28	27	26	25	24
Field	GPIO_EN1[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	GPIO_EN1[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	GPIO_EN1[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	GPIO_EN1[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
GPIO_EN1	31:0	Each bit in this register selects between primary/secondary functions for the associated GPIO pin in this port.	0: Primary function selected 1: Secondary function selected This setting takes effect only when the associated GPIO pin is set to alternate function mode (GPIO_EN=0).



9.4.19 GPIO Drive Strength Register (GPIO_DS, Offset 0x00B0)

Table 9-20. GPIO_DS (Offset 0x00B0)

Position	31	30	29	28	27	26	25	24
Field	DS[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	DS[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	DS[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	DS[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
DS	31:0	Each bit in this register selects the drive strength for the associated GPIO pin in this port. Refer to the Datasheet for sink/source current of GPIO pins in each mode.	0: GPIO port pin is in low-drive mode 1: GPIO port pin is in high-drive mode



10.0 Analog to Digital Converter (ADC)

This device has a 10 bit sigma-delta Analog-to-Digital Converter (ADC).

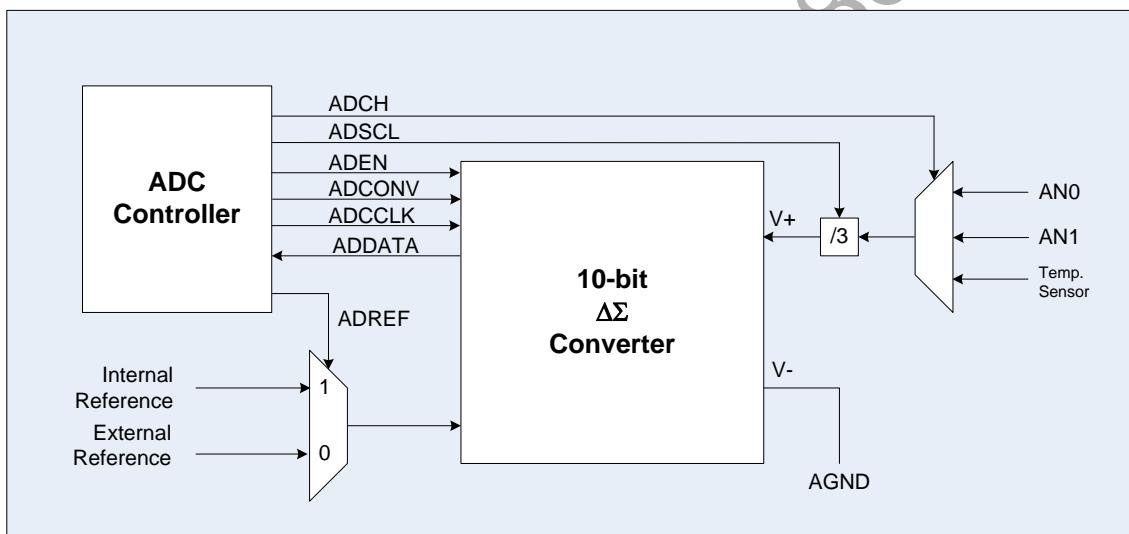
10.1 Overview

The ADC is a 10-bit analog to-digital-converter. This ADC has the following features:

- Three channels
- Single shot mode
- Conversion rate 6K samples/s
- Power-down mode performing minimal power dissipation
- Programmable threshold interrupts—each channel has a high and low reading register. If any ADC reading on that channel falls outside the thresholds, interrupt is triggered if enabled.

Reference voltage for the ADC can either be internal reference or external. User may select the reference voltage through software by setting ADREF bit of the ADC Control Register.

Figure 10-1. ADC Block Diagram



10.2 ADC Operation

The ADC operation is controlled by a set of registers. Refer to section Table 10-1 for complete list of registers.

The ADC must be enabled prior to any ADC conversion. The ADC can be enabled by setting the ADC Enable bit in ADC Control Register to 1 (ADEN=1). In order to obtain accurate data, the application software should wait for the ADC to start up before start of a conversion. This is shown as t_{ADC_SETUP} in . To start a conversion, set the ADC Start Conversion bit in ADC Status Register to 1 (ADCONV=1).

The ADCONV bit will remain set until the conversion process is finished. The ADCONV bit will be cleared to 0 when data is available in the ADC Data Register (ADC_DATA). Clearing the ADCONV bit to 0 will stop ADC operation immediately and the previous data value is preserved. Writing a '1' to the ADCONV bit when ADCONV bit is already set to 1 will be ignored by the ADC controller. After ADCONV is cleared to 0, software must wait at least 1 ADCCLK period before initiating another ADC conversion.

By using the ADC Channel Select bits (ADCH) in the ADC Control Register, the user can select the desired input channel. In addition, each conversion can choose its own reference from external or internal reference. ADC data can be retrieved in either left justified or right justified format (ADALGN), giving application direct control on data format.



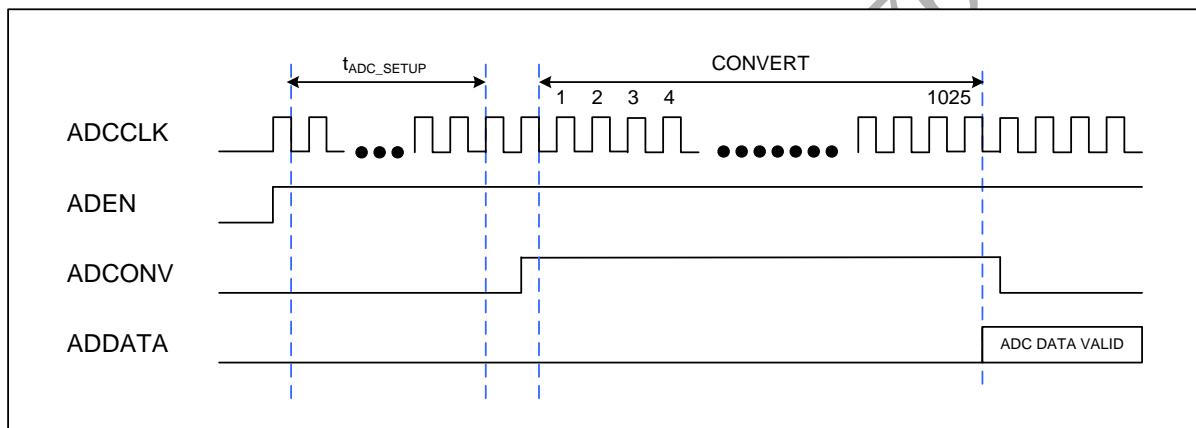
The voltage at AN0 and AN1 inputs can further be divided down by 3 by using the ADSCL bit along with internal reference. This allows direct measurement of supply voltage.

An internal reference is provided for ADC conversion. The internal reference will be enabled when selected by writing a 1 to ADREF field of ADC Control Register (ADREF=1). Software should wait for reference to be stable (after t_{REF_SETUP}) before initiating conversion.

Note that anytime the channel number, scale or reference is changed, user application must wait for t_{ADC_SETTLE} for the ADC to settle before initiating a conversion.

A conversion takes 1025 ADCCLK cycles to complete. The ADCCLK is derived from the system clock with divide ratio defined by ADC Clock Divider Bits (ADCCLK) in ADC Control Register. Therefore, with 1025 ADCCLK to acquire one data, the fastest ADC rate = (PCLK / 16) / 1025 (ADCCLK=0h).

Figure 10-2. ADC Sample Acquisition



10.3 ADC Interrupts

An optional interrupt the CPU can be generated when a data conversion is finished. The ADC Data Interrupt Flag (ADDAI) in ADC Status Register will be set to 1 at the completion of a data conversion. If the ADC Data Available Interrupt is enabled (ADDAIE=1), an interrupt will be generated to the CPU when ADDAI=1. Once set, the ADDAI flag can be cleared by software writing a '0' or automatically cleared at the start of a conversion process when ADCONV is set to 1.

10.4 ADC Temperature Conversion

If the temperature sensor is selected by setting ADC_CN.TSEN bit to one and ADC_CN.ADCH to 11, the temperature measurement formula is:

$$\text{Temp}(X) [\text{°C}] = 25 [\text{°C}] - (\text{ADC_DATA}(X) - A) \times 0.68[\text{°C}]$$

Where A is 620 (typical). To accurately find the value of A , user may calculate the value of A at room temperature based on the following formula:

$$A = \text{ADC_DATA}(\text{room}) + (T(\text{room}) - 25 [\text{°C}]) / 0.68$$

10.5 ADC Registers

Address assignments for the ADC are outlined below. Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.

Table 10-1. ADC Register Base Address (Base ADDR = 0x4003_4000)

Offset	Access	Register Name	Description
0x0000	RW	ADC_CN	ADC Control Register
0x0004	RW	ADC_ST	ADC Status Register
0x0008	R	ADC_DATA	ADC Data Register
0x000C	RW	ADC_LT0	ADC Low Threshold Register 0
0x0010	RW	ADC_LT1	ADC Low Threshold Register 1
0x0014	RW	ADC_LT2	ADC Low Threshold Register 2
0x0018	RW	ADC_HT0	ADC High Threshold Register 0
0x001C	RW	ADC_HT1	ADC High Threshold Register 1
0x0020	RW	ADC_HT2	ADC High Threshold Register 2



10.5.1 ADC Control Register (ADC_CN, Offset 0x0000)

This register can only be written to when ADCONV = 0.

Table 10-2. ADC CN (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	ADOVFDIS	RFU[1:0]		TSEN	ADHTIE	ADLTIE	ADCCLK[1:0]	
Reset	0	00		0	0	0	00	
Access	RW	R		RW	RW	RW	RW	
Position	7	6	5	4	3	2	1	0
Field	IREFEN	ADREF	ADSCL	ADCH[1:0]		ADALGN	ADDAIE	ADEN
Reset	0	0	0	00		0	0	0
Access	R	RW	RW	RW		RW	RW	RW

Name	Bits	Description	Settings
ADEN	0	ADC Enable.	0: Disable ADC operation 1: Enable ADC operation
ADDAIE	1	ADC Data Available Interrupt Enable.	0: Disable Interrupt to CPU (when ADDAI=1) 1: Enable Interrupt to CPU (when ADDAI=1)
ADALGN	2	ADC Data Alignment Select. This bit selects the ADC data alignment mode.	0: ADC data is in right alignment format in ADDDATA[9:0] while ADDATA[15:10] is filled with zero. 1: ADC data is in left alignment format in ADDDATA[15:6] while ADDATA[5:0] is filled with zero.
ADCH	4:3	ADC Input Channel Select. These bits select the input channel to the ADC converter.	00: AN0 01: AN1 10: Reserved 11: Internal Temp Sensor
ADSCL	5	ADC Input Scale. This bit selects the scale factor applies to AN0 and AN1 inputs when using the internal reference. Note that the ADSCL setting is ignored when AVDD is selected as the reference.	0: The AN0 and AN1 inputs are not scaled. 1: The AN0 and AN1 inputs are scaled down by 1/3 prior to ADC conversion.
ADREF	6	ADC Reference Select. This bit selects the reference voltage used in the ADC conversion.	0: External Reference (Default) 1: Internal Reference
IREFEN	7	Internal Reference Enable. This bit has the same value as ADREF. This bit is read only.	
ADCCLK	9:8	ADC Clock Divider. These bits select the ADC conversion clock in relationship to the APB Clock. The APB is first passed through a divide by 16 prescaler and passed to programmable divider so that ADC clock would be in the range of 8MHz with APB clock frequency of 100MHz.	00: PCLK / 16 (Default) 01: PCLK / 32 10: PCLK / 64 11: PCLK / 128
ADLTIE	10	ADC Low Threshold Interrupt Enable.	0: Disable Interrupt to CPU (when ADLTIF=1) 1: Enable Interrupt to CPU (when ADLTIF=1)

Name	Bits	Description	Settings
ADHTIE	11	ADC High Threshold Interrupt Enable.	0: Disable Interrupt to CPU (when ADHTIF=1) 1: Enable Interrupt to CPU (when ADHTIF=1)
TSEN	12	Temp Sensor Enable. This bit enables the internal temp sensor available when ADC_CN.ADCH = b11.	0: Internal Temp Sensor disabled 1: Internal Temp Sensor enabled
RFU	13:14	Reserved	N/A
ADOVFDIS	15	Overflow Disable.	0: ADST.ADOVF bit will be set when the ADC overflows. In this case ADDATA will return a data of 0x3FF if an overflow occurs. 1: ADST.ADOVF bit will not be set when an overflow occurs. In this case ADDATA will return the overflowed data.
RFU	31:16	Reserved	N/A



10.5.2 ADC Status Register (ADC_ST, Offset 0x0004)

Unrestricted write access to ADDAI, ADLTIF and ADHTIF only. For write restriction on other bits, refer to individual bit description.

Table 10-3. ADC_ST (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU	RFU	ADHTIF	ADLTIF	ADOVF	ADDAI	ADCONV	
Reset	1	00	0	0	0	0	0	
Access	R	R	RW	RW	RW	RW	RW	

Name	Bits	Description	Settings
ADCONV	0	ADC Start Conversion. Note that since this bit is not cleared on entry to STOP mode, it is up to the user software to clear ADCONV before entering STOP mode. If this bit is not cleared, the ADC will be held in its current state during STOP mode, and will resume operation at the point where it left off when STOP mode is exited. To avoid undesirable current draw by this block it is recommended to clear this bit prior to entering the STOP mode.	0: Stop ADC conversion. The ADC will stop acquiring data immediately. This bit is not cleared to '0' on entry of STOP mode. Write to ADCONV is ignored when ADC is disabled. 1: starts an ADC conversion when the ADC is enabled (ADEN=1). This bit will remain set until the conversion process is finished.
ADDAL	1	ADC Data Available Interrupt Flag.	0: This bit is cleared by software writing a '0' or automatically cleared at the start of a conversion process when ADCONV is set to 1. 1: This bit is set to 1 when an ADC conversion result is available in the ADDATA register. This flag will cause an interrupt if the ADDAIE is enabled.
ADOVF	2	ADC Overflow Flag. Once set this bit will remain set unless cleared by software.	0: No ADC overflow 1: ADC overflow has occurred
ADLTIF	3	ADC Low Threshold Interrupt Flag. Setting this bit to 1 by software will generate an interrupt if enabled.	0: This bit is cleared by software or by starting a conversion (ADCONV set to 1). 1: This bit is set to 1 when an ADC conversion result is less than ADC Low Threshold value (ADLTn) of the selected channel. This flag will cause an interrupt if the ADLTIE bit is set.
ADHTIF	4	ADC High Threshold Interrupt Flag. Setting this bit to 1 by software will generate an interrupt if enabled.	0: This bit is cleared by software or by starting a conversion (ADCONV set to 1). 1: This bit is set to 1 when an ADC conversion result is more than ADC high Threshold value (ADHTn) of the selected channel. This flag will cause an interrupt if the ADHTIE bit is set.



Name	Bits	Description	Settings
RFU	6:5	Reserved	N/A
RFU	7	Reserved.	Read returns 1.
RFU	31:8	Reserved	N/A

E/UG25H05/36488764



10.5.3 ADC Data Register (ADC_DATA, Offset 0x0008)

Table 10-4. ADC DATA (Offset 0x0008)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	DATA[15:8]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	DATA[7:0]							
Reset	0000 0000							
Access	R							

Name	Bits	Description	Settings
DATA	15:0	Data. When read, this register returns the ADC conversion data. The ADC conversion result will be aligned according to the ADALGN setting.	
RFU	31:16	Reserved	N/A



10.5.4 ADC Low Threshold Register n (ADC_LTn, Offset 0x000C + $n*4$, $n=0, 1, 2$)

Table 10-5. ADC_LTn (Offset 0x000C + $n*4$)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[5:0]						ADLT[9:8]	
Reset	R						00	
Access	0000 00						RW	
Position	7	6	5	4	3	2	1	0
Field	ADLT[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
ADLT	9:0	ADC channel n (AN n) Low Threshold Value, where n is 0, 1 or 2. When an ADC conversion result of AN n channel is less than ADC Low Threshold value (ADLT), ADLTIF flag will be set.	
RFU	31:10	Reserved	N/A



10.5.5 ADC High Threshold Register n (ADC-HT n , Offset 0x0018 + $n*4$, $n=0, 1, 2$)

Table 10-6. ADC HT n (Offset 0x0018 + $n*4$)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[5:0]						ADHT[9:8]	
Reset	R						11	
Access	0000 00						RW	
Position	7	6	5	4	3	2	1	0
Field	ADHT[7:0]							
Reset	1111 1111							
Access	RW							
Name	Bits	Description						Settings
ADHT	9:0	ADC channel n (AN n) High Threshold Value, where n is 0, 1 or 2. When an ADC conversion result of AN n channel is greater than ADC High Threshold value (ADHT), ADHTIF flag will be set.						
RFU	31:10	Reserved						N/A

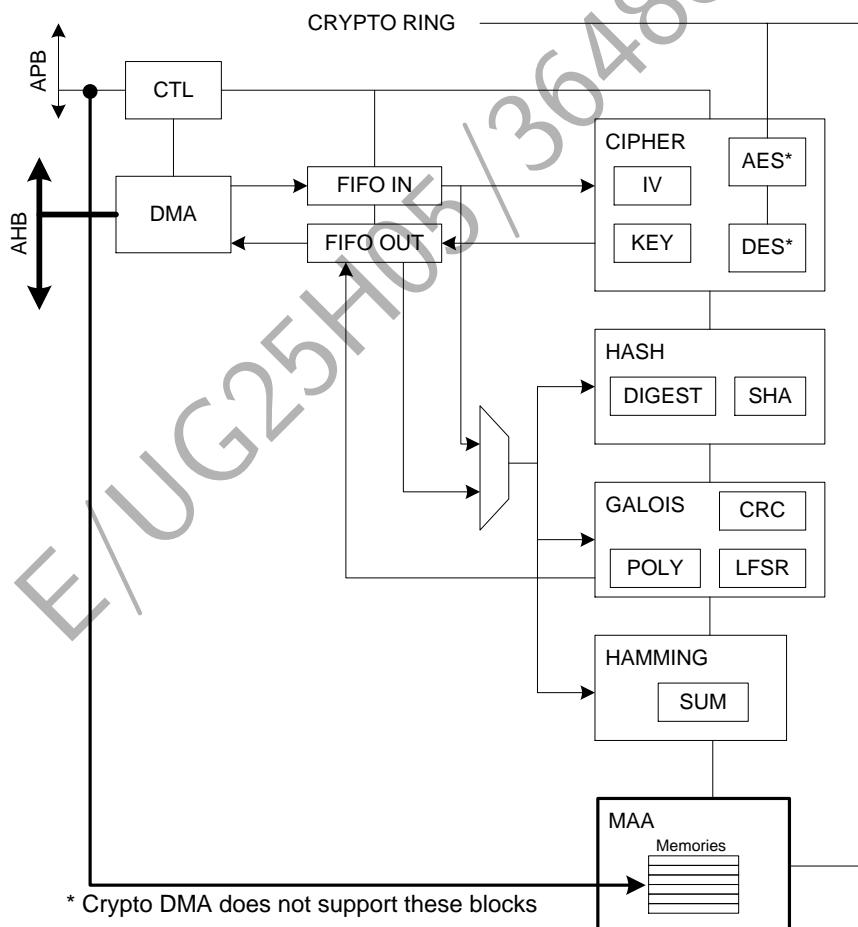
11.0 Cryptographic Accelerator (Rev A Silicon)

The cryptographic accelerator is used to assist the computationally intensive operations of several common algorithms. Supported algorithms include:

- AES-128, 192, and 256 (FIPS 197)
- DES and 3DES (NIST SP800-67)
- SHA-1, 224, 256, 384, and 512 (FIPS 180-3)
- Modulo Arithmetic Hardware Accelerator (MAA)
- Programmable CRC generator
- Hamming code generator
- Support for NIST approved block modes (SP800-38A)

The cryptographic accelerator is configured via an APB register interface.

Figure 11-1. Crypto Accelerator Block Diagram



11.1 Crypto Buffers

11.1.1 Direct Memory Access

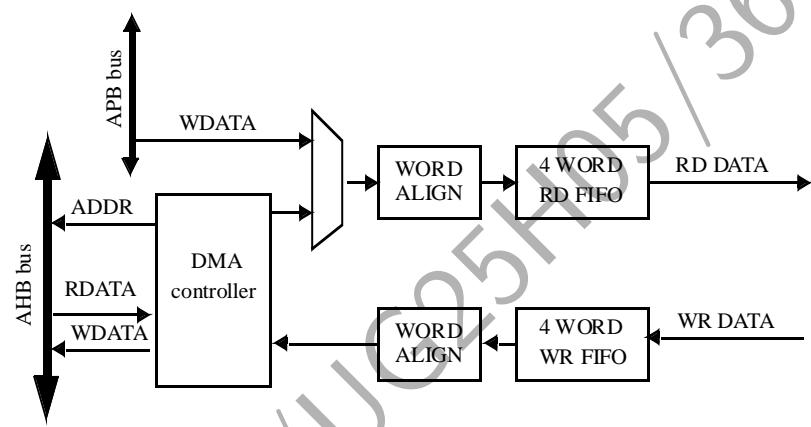
To efficiently hash data, a DMA is integrated with the crypto accelerator. The DMA starts operating when the DMA count is written with a non-zero value. Data is read from the source address and written to the destination address.

The address generators automatically increment. The source and destination of the DMA engine can point to the same memory location where the data is placed. The FIFOs have word alignment buffers to allow data to begin or end on byte boundaries. The DMA bursts data four words at a time with the exception of the beginning or end of the data if it is unaligned.

While the crypto accelerator is busy hashing data, the DMA will prefetch the data for the next operation and store it in the read FIFO. Once the cipher or hash generator is done, the data for the next operation is immediately available. Data output is buffered in the write FIFO so the next hash operation can immediately begin calculating on the next block. This keeps the hash generator running continuously without having to wait for data to be written to or read from the bus. The hash generator input can come from the input or output FIFO.

Only one cipher/hash engine can be used at any time. User should disable all other unused cipher/hash operations.

Figure 11-2. DMA Block Diagram



11.1.2 FIFOs

The read FIFO and write FIFO have programmable sources to allow flexibility in their operation. Data written to the write FIFO is always written out the AHB DMA. The read and write FIFOs can come from the following sources.

Read FIFO

- APB
- AHB DMA
- Random Number Generator

Write FIFO

- None
- Cipher output
- Read FIFO



During crypto operation, a typical setup is to use the AHB DMA as the read FIFO source and the cipher output as the write FIFO source. This reads data from memory and writes the encrypted or decrypted result back out to memory.

A Cipher based Message Authentication Code (CMAC) is similar to digital signature or a Keyed-Hash Message Authentication Code (HMAC). It uses a cipher in a block chaining mode to form a cryptographic checksum. In this mode of operation, the cipher output is not written back to memory. Only the final cipher block is of interest, so the write FIFO source should be set to none.

The DMA can be used to copy memory, similar to the `memcpy` standard C function by setting the write FIFO source to the read FIFO.

Memory can be filled with a block of data, similar to the `memset()` standard C function by pointing the write FIFO source to the read FIFO and setting the read FIFO to the APB. Similarly, memory can be filled with random data by pointing the read FIFO source to the random number generator and the write FIFO source to the read FIFO.

Deterministic Random Number Generators (DRNG) such as those specified in NIST SP 800-90 can be implemented by feeding a source of entropy into a hash function or cipher based MAC. This can be done by setting the read FIFO source to the random number generator, disabling the write FIFO, and enabling the hash generator or cipher.

11.2 Symmetric Block Cipher Accelerator

The symmetric block cipher accelerator supports the Advanced Encryption Standard (AES) family of block ciphers (AES-128, AES-192, and AES-256), the Data Encryption Standard (DES) and Triple Data Encryption Algorithm (TDEA). The DMA cannot be used with any of the block cipher accelerator.

The symmetric block ciphers encrypt or decrypt data in blocks. The block sizes for each cipher is given in the table below.

Table 11-1. Symmetric Block Ciphers

Cipher	Key Size	Used Key Bits	Effective Strength*	Block Size	Throughput @ 96MHz
TDEA	192-bits	168-bits	112-bits	64-bits	16 MBytes/sec
AES-128	128-bits	128-bits	128-bits	128-bits	19 MBytes/sec
AES-192	192-bits	192-bits	192-bits	128-bits	16 MBytes/sec
AES-256	256-bits	256-bits	256-bits	128-bits	14 MBytes/sec

*NIST SP800-57

The block ciphers can be used in various modes of operation. In the simplest mode (ECB), each data block is simply encrypted or decrypted using the cipher. A side effect of this is that identical data blocks will encrypt to the same ciphertext. Various modes of operation are used which chain or feedback ciphertext from the previous block to seed the next encryption operation. This causes identical plaintext data blocks to encrypt to different ciphertexts. NIST SP800-38A specifies five modes of operation.

- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- Cipher Feedback (CFB) – only in AES operation
- Output Feedback (OFB) – only in AES operation
- Counter (CTR) – only in AES operation



For the CFB mode of operation, the mode size is equal to the block size. 128-bit CFB is supported for AES. 1-bit CFB and 8-bit CFB are not supported. For the CTR mode of operation, the lower 32-bits of the initial vector increment.

The cipher algorithm and mode of operation are set in the cipher control register. The cipher will start operating once the FIFO is full.

11.3 Hash Function Accelerator

The Cryptographic Hash Function accelerator supports SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 hash algorithms. It takes an input message of arbitrary length and summarizes it in a fixed length message digest.

Data is processed in 512-bit (16 word, for SHA-1, SHA-224 and SHA-256) or 1024-bit (32 word, for SHA-384 and SHA-512) blocks. After every 16 or 32 words are written to the hash block, several cycles are needed to complete the hash of that block. The number of cycles needed to compute the hash is dependent upon the number of rounds in the algorithm. The first 16 rounds are calculated as the 16 or 32 words are written to the hash block. The remaining rounds are calculated at a rate of one clock cycle per round.

Table 11-2. Hash Functions

Algorithm	Digest Length	Effective Strength*	Block Size	Throughput @ 96MHz
SHA-1	160 bits	63 bits	512 bits	66 MB/sec
SHA-224	224 bits	112 bits	512 bits	80 MB/sec
SHA-256	256 bits	128 bits	512 bits	80 MB/sec
SHA-384	384 bits	192 bits	1024 bits	133 MB/sec
SHA-512	512 bits	256 bits	1024 bits	133 MB/sec

* NIST SP800-57

The integrated DMA is capable of fetching data for the hash accelerator. When using the DMA, the user shall use hash operation only on word-size aligned data. DMA operation is also possible on non-word-size (32-bit for SHA-1/256, 64-bit for SHA-384/512) aligned data but the user shall hash the last message block (512-bit or 1024 bit block of the entire message size) without using the DMA operation.

After the crypto block reset, the CRYPTO_CTRL.RDY must be polled in the software before any other actions can start. Since the crypto DMA does not interact with the CPU caches, prior to any DMA transfer, the caches storing the source and destination memory ranges must be ‘cleaned’. Also after the DMA transfer, caches storing the destination memory range must be ‘invalidated’.

Once all data has been written to the CRYPTO Data register, the final message block must be padded according to the FIPS-180-2 standard. The standard requires that the bit length be appended to the end of the message. The length of the message in binary representation is contained in the HASH Message Size Registers (HASH_MSG_SZ). Prior to hashing the last message data, the Last Message Bit (HASH_CTRL.LAST) should be set along with the HASH_MSG_SZ registers. The HASH_MSG_SZ value will be automatically padded to the last message block.

For SHA-1 and SHA-256, a single bit equal to ‘1’ is appended to the end of the message. The message is then padded using software with zeros until 64 bits remain in the last 512-bit block. If less than 64 bits remain, then zeros are appended to the message until 64 bits remain in the next 512-bit block. The HASH_CTRL.LAST bit is set along with HASH_MSG_SZ[1:0]. Hardware will append values in these registers to the last 64 bits of the final message block.

For SHA-384 and SHA-512, a single bit equal to ‘1’ is appended to the end of the message. The message is then padded using software with zeros until 128 bits remain in the last 1024-bit block. If less than 128 bits remain, then



zeros are appended to the message until 128 bits remain in the next 1024-bit block. The HASH_CTRL.LAST bit is set along with HASH_MSG_SZ_[3:0]. Hardware will append values in these registers to the last 128 bits of the final message block.

11.4 Galois Field Accelerator

The Galois (gal-wah) accelerator can be used to generate Cyclic Redundancy Checks (CRC) of messages or to generate pseudo random numbers using a Linear Feedback Shift Register (LFSR). Galois Fields are sometimes denoted $GF(2^n)$ where n denotes the highest order term of the reduction polynomial and 2^n denotes the size of the field.

11.4.1 Cyclic Redundancy Check

Cyclic Redundancy Checks (CRC) are commonly used for error detection. An n -bit CRC is capable of detecting the following types of errors.

- all single bit errors
- all two bit errors for block lengths less than 2^k where k is the order of the longest irreducible factor of the polynomial
- all odd number of errors for polynomials with the parity polynomial $(x+1)$ as one of its factors (polynomials with an even number of terms)
- all burst errors less than n bits
- overall, all except 1 out of 2^n errors are detected
 1. 99.998% for a 16-bit CRC
 2. 99.9999998% for a 32-bit CRC

The crypto accelerator can calculate the CRC of a block of data. Data is written to the CRYPTO Data register.

In practice, the starting initial CRC value is typically preset to all ones. If the CRC was preset to all zeros and an initial stream of all zeros is received, the CRC does not change. Therefore, most implementations preset the CRC to all ones or some constant value.

Historically, CRCs were calculated on serial bit streams. Most serial bit streams were sent least significant bit first. The CRC was calculated as each bit was transmitted or received. This resulted in the CRC being calculated on the least significant bit of the data first.

The CRC is typically appended to the end of the data. If the receiver calculates the CRC on both the data and received CRC, the result should be all zeros if the data and CRC was received error free. Most implementations do not like to check against an all zero check-sum. Therefore, most implementations invert the CRC before transmitting it. By inverting the CRC on the transmitting end, the resulting CRC on the receiving end should be a constant. The specific constant is dependent upon the CRC polynomial. This works because the non-inverted CRC calculated at the end of the data XOR'd with the received inverted CRC is all ones ($CRC \wedge \sim CRC = 1s$). Shifting all ones through the polynomial results in the same constant for each message, the constant is dependent upon the polynomial.

Since the receiving end calculates a new CRC on both the data and received CRC, the received CRC must be sent in the correct order so the highest order term of the CRC is shifted through the generator first. Since data is typically shifted through the generator LSB first, this means the CRC is reversed bit-wise so the highest order term of the remainder is in the LSB position. Software CRC algorithms typically handle this by calculating everything backwards. They will reverse the polynomial and do right shifts on the data. The resulting CRC ends up being bit swapped and in the correct format.

The CRYPTO_CRC register will be preset to all ones if the crypto block is reset. The initial CRC state can be written to any value. The final inversion must be done by software if required.

The CRC generator has a programmable polynomial up to 32-bits. The polynomial should be written to the CRC_POLY register. The largest term x^n defines the length of the CRC. When calculating the CRC on data LSB first, the polynomial should be reversed so that the coefficient of the highest power term is in the LSB position. The



largest term x^n is implied (always one) and should be omitted when writing to the CRC_POLY register. This is necessary because the polynomial is always one bit larger than the resulting CRC, so a 32-bit CRC has a polynomial with 33 terms (x^0 to x^{32}).

CRC polynomials with good error detection properties should be irreducible (the polynomial should not be factorable). Therefore, the constant term x^0 or 1 should always be present, otherwise the polynomial would be factorable by x . If the constant term x^0 or 1 were not present, the resulting CRC would be cyclic with a subgroup smaller than x^n . The effective length of the CRC would be the difference between the highest and lowest order terms. Therefore, the highest and lowest order terms x^n and x^0 should always appear in the polynomial.

When found in literature, sometimes the LSB or MSB of the polynomial is omitted when the polynomial is written in binary. It is more common to see CRC polynomials with the MSB implied because that is the bit that is shifted off, XOR'd with the data, and tested to see if the result is set. Some literature assumes the reader knows that an n-bit CRC must have the x^n term set, or else it would be a smaller length CRC.

Some common CRC polynomials and their check constants are shown in the table below. The polynomial register resets to the 32-bit CRC polynomial used by Ethernet, PPP and file compression utilities such as zip or gzip.

Table 11-3. Common CRC Polynomials

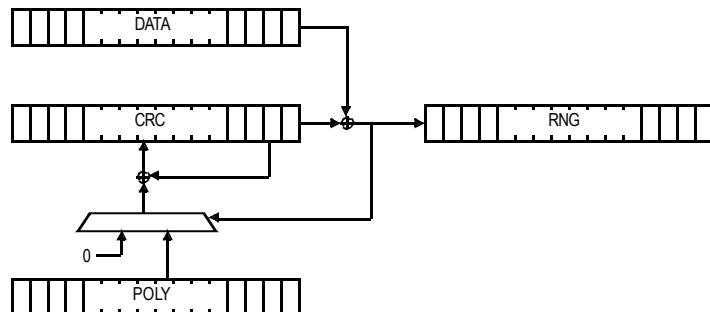
Algorithm	Polynomial Expression	Order	Polynomial	Check
CRC-32 Ethernet	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x^1+x^0$	LSB	0xEDB88320	0xDEBB20E3
CRC-CCITT	$x^{16}+x^{12}+x^5+x^0$	LSB	0x00008408	0x0000F0B8
CRC-16	$x^{16}+x^{15}+x^2+x^0$	LSB	0x0000A001	0x0000B001
USB data	$x^{16}+x^{15}+x^2+x^0$	MSB	0x80050000	0x800D0000
Parity	x^1+x^0	LSB	0x00000001	

By default the CRC accelerator does right shifts and calculates the CRC on the LSB of the data first. The CRC can be calculated on the MSB of the data first by setting the bit-swap control bit. To calculate the CRC MSB first, you must left justify the polynomial in the CRC_POLY register. The hardware implies the MSB of the polynomial just as it did when shifting the LSB first. The LSB of the polynomial should be set, this defines the length of the CRC. The initial state of the CRC should also be left justified. When the CRC calculation is complete, it is necessary to right shift the CRC to right justify it if the polynomial is less than 32-bits.

11.4.2 Linear Feedback Shift Register

Linear Feedback Shift Registers (LFSR) are commonly used to implement Pseudo Random Number Generators (PRNG). A LFSR polynomial can be written to the CRC_POLY register to generate pseudo-random data. The starting state or seed for the pseudo random sequence should be written to the CRC register. The lockup state of all zeros is detected and the LFSR will substitute the value 1 to prevent lockup.

Figure 11-3. Galois Field CRC and LFSR Architecture



Different polynomials will generate different sequences of random data. Ideally, an n-bit polynomial will generate a random sequence of $2^n - 1$ bits. Not all polynomials are maximal length, some repeat before the theoretical maximum



length of $2^n - 1$. There are thousands of different maximal length 32-bit LFSR polynomials. Any length of an LFSR polynomial can be used up to 32-bits. Some tables of maximal length LFSR polynomials omit the MSB x^n term or the LSB $x^0 = 1$ terms. Fibonacci LFSRs feedback the XOR of all the taps to the constant term $x^0 = 1$ so it is often implied when listing the taps, but must be present when writing the polynomial to the CRC_POLY register.

The crypto accelerator automatically generates the next sequence of 32-bits whenever the CRYPTO_LFSR register is read. If the PRNG control bit is set, the incoming data will be forced to zero. The DMA can be used to quickly fill a block of memory with pseudo-random data.

11.4.3 Entropy

The crypto block has a high frequency ring oscillator (approximately 1GHz) which is sampled every system clock to generate entropy.

The raw output of the ring oscillator can be observed to determine how much entropy it provides. Given the probability of predicting the ring oscillator output, the amount of entropy provided per bit can be calculated according to:

$$\text{entropy} = -\log_2(\text{predictability})$$

If the output can be predicted with 99% certainty, then 0.0145 bits of entropy are provided per bit of output. In order to obtain 32 bits of entropy, a minimum of 2207 bits of output from the ring oscillator would need to be mixed. If the probability of predicting the output is 50%, 1 bit of entropy is provided per bit of output.

11.5 Hamming Code Generator

The Hamming code generator can be used to calculate an Error Correction Code (ECC) on a block of data. One application of Hamming codes is for Single Level Cell (SLC) NAND Flash memories. Multi-Level Cell (MLC) Flash memories require Error Correction Codes that can correct multiple bit errors since a corrupt cell can have multiple bit errors.

Hamming codes are capable of correcting single bit errors. An extra parity bit can be included to detect two bit errors. This is commonly referred to as Signal Error Correction, Double Error Detection (SECDED). Three errors will masquerade as a correctable single bit error.

The Hamming code generator calculates the ECC on a block of data up to 2^{16} bits in length (8k bytes). Software implementations can extend this to any length. Since the Hamming code can only correct a single bit error, increasing the block size increases the likelihood of multiple errors which are uncorrectable. The Hamming code generator in the crypto accelerator generates even parity on even halves of bit groups.

If the user wants the parity of the odd halves of the bit groups, they can XOR the parity of the even halves with the parity of the entire array. If the parity of the entire array is odd, the parity of the odd halves is the inverse of the parity of the even halves. If the parity of the entire array is even, the parity of the odd halves is identical to the parity of the even halves.

**Figure 11-4. Hamming XOR Calculation**

b7	b6	b5	b4	b3	b2	b1	b0	byte 5
b7	b6	b5	b4	b3	b2	b1	b0	byte 4
b7	b6	b5	b4	b3	b2	b1	b0	byte 3
b7	b6	b5	b4	b3	b2	b1	b0	byte 2
b7	b6	b5	b4	b3	b2	b1	b0	byte 1
b7	b6	b5	b4	b3	b2	b1	b0	byte 0

ECC bit0 – xor every other bit

b7	b6	b5	b4	b3	b2	b1	b0	byte 5
b7	b6	b5	b4	b3	b2	b1	b0	byte 4
b7	b6	b5	b4	b3	b2	b1	b0	byte 3
b7	b6	b5	b4	b3	b2	b1	b0	byte 2
b7	b6	b5	b4	b3	b2	b1	b0	byte 1
b7	b6	b5	b4	b3	b2	b1	b0	byte 0

ECC bit1 – xor every other 2 bits

b7	b6	b5	b4	b3	b2	b1	b0	byte 5
b7	b6	b5	b4	b3	b2	b1	b0	byte 4
b7	b6	b5	b4	b3	b2	b1	b0	byte 3
b7	b6	b5	b4	b3	b2	b1	b0	byte 2
b7	b6	b5	b4	b3	b2	b1	b0	byte 1
b7	b6	b5	b4	b3	b2	b1	b0	byte 0

ECC bit2 – xor every other 4 bits

b7	b6	b5	b4	b3	b2	b1	b0	byte 5
b7	b6	b5	b4	b3	b2	b1	b0	byte 4
b7	b6	b5	b4	b3	b2	b1	b0	byte 3
b7	b6	b5	b4	b3	b2	b1	b0	byte 2
b7	b6	b5	b4	b3	b2	b1	b0	byte 1
b7	b6	b5	b4	b3	b2	b1	b0	byte 0

ECC bit3 – xor every other 8 bits

b7	b6	b5	b4	b3	b2	b1	b0	byte 5
b7	b6	b5	b4	b3	b2	b1	b0	byte 4
b7	b6	b5	b4	b3	b2	b1	b0	byte 3
b7	b6	b5	b4	b3	b2	b1	b0	byte 2
b7	b6	b5	b4	b3	b2	b1	b0	byte 1
b7	b6	b5	b4	b3	b2	b1	b0	byte 0

ECC bit4 – xor every other 16 bits

b7	b6	b5	b4	b3	b2	b1	b0	byte 5
b7	b6	b5	b4	b3	b2	b1	b0	byte 4
b7	b6	b5	b4	b3	b2	b1	b0	byte 3
b7	b6	b5	b4	b3	b2	b1	b0	byte 2
b7	b6	b5	b4	b3	b2	b1	b0	byte 1
b7	b6	b5	b4	b3	b2	b1	b0	byte 0

ECC bit5 – xor every other 32 bits

For a block of data 2^n bits in length, $n+1$ ECC bits are needed for single bit error correction (ECC B0 through Bn = $n+1$ ECC bits). ECC bit n indicates parity of the entire array. If it is different than the stored ECC bit n, it indicates an error in the data array. The location of the error can be determined using the lower n ECC bits (B0 to Bn-1).

To determine if an error occurred, the user should XOR the saved and calculated ECC bits. If the result is zero, no error occurred. If the result of the ECC XOR operation is not zero, the location of the failing bit is given by the inverse of the ECC XOR result. If the failing bit location is greater than or equal to the size of the data block (bit n of the inverted ECC XOR is set), then the error is in the ECC bits. The ECC bit that is corrupt is the bit that is set as a result of the XOR of the two sets of ECC bits.

$$\text{failing_bit_location} = \sim(\text{ecc_saved} \wedge \text{ecc_calculated})$$

An error in the most significant ECC bit n will masquerade as an error in the most significant bit of the data. To properly detect and correct an error in the MSB of the ECC, the user can include an extra ECC bit ($n+2$ ECC bits). These two most significant ECC bits should be identical if they are error free. If they are different, the failing ECC bit can be determined by XORing the saved and calculated ECC bits.

This extra ECC bit can be saved at the cost of an additional parity calculation. If the MSB of the data is set, the ECC parity bits should be XOR'd with $2^{n+1}-1$ (n 1s). This effectively swaps the MSB of the data with the MSB of the ECC for purposes of parity calculation. When examining the result, if the ECC calculation indicates the MSB of the data is corrupt (the result is $2^{n+1}-1$), the error is really in the MSB of the ECC. If the result indicates the error is in the MSB of the ECC (the result is 2^{n+1}), the error is really in the MSB of the data.

ECC bits n and above are all identical. They all indicate the parity of the entire array. To use a block size of 64k bits (8k bytes), the parity bit of the entire array (bit 16) can be duplicated to obtain the higher order ECC bits.

Hamming codes can be generated over larger blocks using software. After the Hamming code is generated for an 8k block of data, software should examine the parity of the block of data just calculated (bit 16 of the ECC register). If the parity of the block is odd (parity bit is set), software should XOR additional software maintained ECC bits with the inverse of the block address. If the parity of the block is even (parity bit of the block is clear), soft-ware does not need to modify the additional ECC bits. The parity of the block (bit 16 of the ECC register) should be reset for each block, but the remaining ECC bits (B0-B15) should remain unchanged and accumulate their respective XORs throughout the entire array.

To achieve Double Error Detection (but not correction), another ECC parity bit can be included. If the result of the ECC XOR is not zero, then this extra ECC bit should also be set indicating an error has occurred. If this bit is clear

after the ECC XOR operation, it means an even number of errors has occurred and the data is not correctable. This does not mean this extra ECC bit will detect all even number of bit errors, it just indicates an even number of errors have occurred. It is possible for an even number of bit errors to masquerade as valid data. This extra ECC bit will be capable of detecting two bit errors. If an odd number of bit errors occur, they will always masquerade as a single correctable bit error. This is a limitation of using parity to indicate error. Parity can only detect an odd number of errors. For stronger error detection, a Cyclic Redundancy Check (CRC) can be used.

The lower n ECC bits (B_0 to B_{n-1}) are all that is needed to determine the location of a single bit error in the data. The next ECC bit (which is the parity of the entire array) is simply used to indicate there is an error in the data that needs to be corrected. Subsequent ECC bits are only needed to detect an error in this added ECC bit or to detect double bit errors. If an alternative error detection scheme is used, such as a CRC, then only n ECC bits are needed to find the location of a single bit error in the data array.

Since all CRCs with at least two terms in the generator polynomial will detect all single bit errors, ECC bit n which is used to determine if there is a single bit error in the array by checking parity is unnecessary. A CRC polynomial with an even number of terms has the parity polynomial $(x+1)$ as one of its factors and checks parity as well. A CRC of sufficient length is also capable of detecting all two bit errors, making an ECC bit added for double error detection unnecessary as well.

Since a single bit error in the CRC would be catastrophic, the CRC should also be protected with ECC bits. If the CRC is appended to the data for the Hamming code calculation, it can be protected with the same set of ECC bits used to protect the data by including one additional ECC bit to account for the increase in block size.

Most manufacturers of NAND Flash memories have application notes that describe the calculation of a variation of Hamming codes. These application notes calculate parity on both the odd and even halves of the bit groups. As a result, they require $2*n$ ECC bits, almost twice as many as necessary. If the even half is XOR'd with the odd half, the result is the parity of the entire array. Instead of storing parity for both the even halves and odd halves of all bit groups (which requires $2*n$ ECC bits), an implementation only really needs to store the parity of the entire array ($n+1$ ECC bits). The odd half of ECC bits can be determined by XORing the even half of ECC bits with the parity of the entire array. The parity of the entire array is just the next bit in the ECC register, so saving $n+1$ bits from this register inherently saves the parity of the entire array. Storing both the odd and even parity halves offers no more error protection than the methods described above. For both methods, an odd number of errors will masquerade as a correctable single bit error and all double bit errors will be detected.

To calculate the ECC of a new block of data, the RST bit of the CRC Control register should be set to reset the Hamming parity bits to zero and to also reset the internal byte counter.

11.6 Modular Arithmetic Accelerator (MAA)

The Modular Arithmetic Accelerator (MAA) supports IEEE Public Key Cryptography Standard (P1363) for asymmetric cryptographic operations based on DSA, RSA, and ECDSA algorithms.

The accelerator allows the user to compute a set of operations that are important parts of many cryptographic algorithms. The operations include:

- Supports up to a 2048-bit operand size
- Performs up to 2048-bit modular exponentiation ($a^e \bmod m$)
- Performs up to 2048-bit modular multiplication ($a*b \bmod m$)
- Performs up to 2048-bit modular square ($b^2 \bmod m$)
- Performs up to 2048-bit modular square followed by modular multiply ($((b^2 \bmod m) * a \bmod m)$)
- Performs up to 2048-bit modular addition ($a+b \bmod m$)
- Performs up to 2048-bit modular subtraction ($a-b \bmod m$)
- Optimized calculation mode to maximize speed
- Non-optimized calculation mode to maximize security

Using the above operations the following can also be computed:



- Performs modular inversion
- Performs modular reduction

The MAA is a peripheral designed to operate independently from the processor except when the processor is reading or writing the control register, or when it is used to load/unload the data in the specified data memory segment.

11.6.1 Operation

The MAA is controlled and accessed via the Modular Arithmetic Accelerator Control register (MAA_CTRL). The control register provides option control on arithmetic operations, data partition and status bits for start/busy. Operands and parameters are stored in a 768x16 data memory segment. This memory segment is configurable via the Memory Assignment (AMA, BMA, RMA and TMA). Per FIPS' big-endian data convention, the most significant byte or sub-word of a multi-byte word is loaded first and stored at the lowest storage location.

The calculation configuration bits (CLC) are used for MAA operation selection. Operation starts when the Start Calculation (STC) bit is set to 1. The STC bit also serves as the status bit for software monitoring during operation. Once the STC bit is set, it will remain set until either the operation is completed or an error is detected.

Software can stop an MAA operation by clearing STC to 0. If an MAA operation is terminated prematurely by software, no error or interrupt will be generated. In addition, no valid data is available.

The Optimized Calculation Control bit (OCALC) allows the user software to optimize the speed of an exponentiation by skipping unnecessary multiply operations when the corresponding exponent bit is a "0". The OCALC bit defaults to 0, forcing the MAA to operate in a non-optimized mode. The non-optimized mode skips the leading zeros of the exponent and starts square/multiply operations when a "1" has been detected. From this point on, multiply operation will be performed for every exponent bit, regardless of its logical value.

Generating encryption keywords with optimized calculations will result in fluctuation in device power dissipation. Observation of the fluctuation in power may allow observers to deduce the value of the keyword. For maximum security, non-optimized mode is recommended.

The Modular Accelerator Word Size register (MAA_MAWS) is used to define the calculation size of a modular operation. The content of the register presents the number of bits for the modular operation. For example, to perform a 1007-bit (03EFh) modular operation, this register would have to be set to 03EFh prior to setting STC.

Valid word size is from 1 to 2048. The accelerator will not start if MAWS is invalid.

The MAA Done (CRYPTO_CTRL.MAA_DONE) bit will be set to 1 after the completion of an MAA operation or when an error occurs. The MAA_DONE bit can be cleared by software writing a 0. Setting MAA_DONE to 1 by software will generate an interrupt if enabled. If Crypto Interrupt Enable (CRYPTO_CTRL.INT) bit is set to 1, an interrupt will be generated to the CPU when the MAA_DONE = 1.

11.6.2 Memory Partition

The MAA divides the crypto memory into six 256 bytes logical segments to store operands and intermediate values. These logical segments are mapped to location starting from 0x4000_1000 with offsets shown in Table 11-5. The write access to these memory locations are Word only.

The segment allocation is as followed:

- **a** – Multiplier/Operand A, selected by Multipler / Operand A Memory Assignment (AMA)
- **b** – Multiplicand/Operand B, selected by Multipler / Operand B Memory Assignment (BMA)
- **e** – Exponent
- **m** – Modulus
- **t** – Temporary Storage , selected by Temporary Memory Assignment (TMA)
- **r** – Result value , selected by Result Memory Assignment (RMA)



While user can allocate segment for the various parameters, the following restrictions apply:

- Only one parameter can be stored in each memory instance.
- The modulus (***m***) is always stored in memory instance 5.
- When an exponentiation operation is selected, the exponent (***e***) is always stored in memory instance 4. If another operation is selected, the exponent is not stored in any memory instance and memory instance 4 is free to hold another parameter.
- Parameters ***m***, ***b***, ***t*** and ***r*** must be stored in different memory instances (not just different segments). For example, if ***b*** is stored in memory instance 0, then neither ***t*** nor ***r*** may be stored in memory instance 1 when word size is smaller than 1024. Each memory instance is 256 bytes.

Table 11-4. MAA Memory Segments

	Memory Instance	Memory Segment (MAWS \geq 1024)	Memory Segment (MAWS < 1024)	Dedicated Function	Address Offset
xMA[3:0]	0	0	0	None	0x0100 – 0x017F
			1		0x0180 – 0x01FF
	1	1	2	None	0x0200 – 0x027F
			3		0x0280 – 0x02FF
	2	2	4	None	0x0300 – 0x037F
			5		0x0380 – 0x03FF
	3	3	6	None	0x0400 – 0x047F
			7		0x0480 – 0x04FF
	4	4	8	Exponent (if needed)	0x0500 – 0x057F
			9		0x0580 – 0x05FF
	5	5	-	Modulus	0x0600 – 0x06FF

Four of the parameters (***a***, ***b***, ***e***, and ***m***) may also be shifted in starting position within their respective memory instances by setting the appropriate Memory Select bits (AMS, BMS, EMS, MMS). Although the starting position may be altered, the entire parameter is still stored within a single memory instance, wrapping around as follows.

Table 11-5. Memory Binding (Logical Segment 0 Shown)

xMS[1:0]	Address Offset
00	0x0100 ... 0x01FF
01	0x0140 ... 0x01FF, 0x0100 ... 0x013F
10	0x0180 ... 0x01FF, 0x0100 ... 0x017F
11	0x01C0 ... 0x01FF, 0x0100 ... 0x01BF

11.6.3 Application Note

The MAA is running on internal crypto oscillator. Generating encryption keywords with optimized calculations will result in fluctuation in device power dissipation. Observation of the power fluctuation may allow observers to deduce keyword value. For maximum security, non-optimized mode is recommended.

MAA operand size is specified by the MAA Word Size Register (MAA_MAWS). Valid values are from 1 to 2048. This value specifies valid ranges for ***a***, ***b***, ***e***, and ***m***.

a, ***b***, and ***e***, can have any values between 0 and $2^{\text{MAWS}} - 1$ inclusive as long as their number of bits are less than those of ***m***. For exponentiation operation, however, ***b*** memory must contain the value of 1 and ***e*** cannot be 0.

The ***m*** memory, which holds the modulus, will have a value from $2^{\text{MAWS}-1}$ up to $2^{\text{MAWS}} - 1$. For example, for a 16-bit MAWS=0x10, it will have a value from 0x8000 (32768) up to 0xFFFF (65535)

If any parameter exceeds Word Size (MAWS), an invalid result may be generated without issuing an MAA error status. It is up to application to check for invalid word sizes.



The MAA operands are stored as 64-bit blocks. For all the memories, operands must be zero padded to the 64-bit block boundary. There is no restriction on values stored in the memories beyond this boundary. For example, MAWS=65, and operand=01 xxxx xxxx xxxx xxxxh. In this example, MAA operands are stored as two 64-bit blocks. Bit[63:0] contains the lower 64 bits. Bit[127:66] are zero padded while bit[65]=1. Data in the words above where the 0000 0000 0000 0001h is stored may have any value.

The multiplier supports unsigned operation. For modular subtraction, operand **b** must be equal to or less than operand **a**.

For most calculations, memory segments **t** and **r** will be used to store intermediate values during round of operations and may contain the same value at the final operation.

For square-multiply and exponentiation, memory segment **b** is used as an additional temporary storage area during round operations.

11.6.4 Error

The MAA Error bit (MAAER) is used to signify an error condition. MAAER is set to 1 if user attempts to: Write to MAA_CTRL or MAA_MAWS when MAA is in progress. This will also terminate the MAA operation

In the above cases, setting STC to 1 does not initiate a new MAA operation and STC remains cleared at 0. The MAAER must be cleared by software once set, otherwise no new operation can be initiated.



11.7 Cryptographic Registers

Address assignments for the Cryptographic Accelerator registers are outlined in Table 11-6.

Table 11-6. Cryptographic Accelerator Register Address (Base ADDR = 0x4000_1000)

Offset	Access	Register	Description
0x0000	RW	CRYPTO_CTRL	Crypto Control Register
0x0004	RW	CIPHER_CTRL	Cipher Control Register
0x0008	RW	HASH_CTRL	HASH Control Register
0x000C	RW	CRC_CTRL	CRC Control Register
0x0010	RW	DMA_SRC	Crypto DMA Source Register
0x0014	RW	DMA_DEST	Crypto DMA Destination Register
0x0018	RW	DMA_CNT	Crypto DMA Count Register
0x001C	RW	MAA_CTRL	MAA Control Register
0x0020	W	CRYPTO_DIN_[3:0]	Crypto Data In Register [3:0]
...			
0x002C			
0x0030	R	CRYPTO_DOUT_[3:0]	Crypto Data Out Register [3:0]
...			
0x003C			
0x0040	RW	CRC_POLY	CRC Polynomial Register
0x0044	RW	CRC_VAL	CRC Value Register
0x0048	R	CRC_PRNG	Pseudo-Random Number Generator Register
0x004C	RW	HAM_ECC	Hamming ECC Register
0x0050	RW	CIPHER_INIT_[3:0]	Cipher Initial Vector Register [3:0]
...			
0x005C			
0x0060	W	CIPHER_KEY_[7:0]	Cipher Key Register [7:0]
...			
0x007C			
0x0080	RW	HASH_DIGEST_[15:0]	HASH Message Digest Register [15:0]
...			
0x00BC			
0x00C0	RW	HASH_MSG_SZ_[3:0]	HASH Message Size Register [3:0]
...			
0x00CC			
0x00D0	RW	MAA_MAWS	MAA Word Size Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



11.7.1 Crypto Control Register (CRYPTO_CTRL, Offset 0x0000)

Table 11-7. CRYPTO_CTRL (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	DONE	RDY	ERR	MAA_DONE	CPH_DONE	HSH_DONE	GLS_DONE	DMA_DONE
Reset	0	1	0	0	0	0	0	0
Access	R	R	R	RW	RW	RW	RW	RW
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[3:0]				RDSRC[1:0]		WRSRC[1:0]	
Reset	0000				00		00	
Access	R				RW		RW	
Position	7	6	5	4	3	2	1	0
Field	WAIT_POL	WAIT_EN	BSI	BSO	RFU	SRC	INT	RST
Reset	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
RST	0	Reset. This bit is used to reset the crypto accelerator. All crypto internal states and related registers are reset to their default reset values. Control register such as CRYPTO_CTRL, CIPHER_CTRL, HASH_CTRL, CRC_CTRL, MAA_CTRL (with the exception of the STC bit), HASH_MSG_SZ[3:0] and MAA_MAWS will retain their values. This bit will automatically clear itself after one cycle.	0: No effect 1: Reset crypto accelerator
INT	1	Interrupt Enable. Generates an interrupt when done or error set.	0: Interrupt disabled. 1: Interrupt enabled.
SRC	2	Source Select. This bit selects the hash function and CRC generator input source.	0: Input FIFO 1: Output FIFO
RFU	3	Reserved	N/A
BSO	4	Byte Swap Output. Note. No byte swap will occur if there is not a full word.	0: No effect. 1: Byte swap output.
BSI	5	Byte Swap Input. Note. No byte swap will occur if there is not a full word.	0: No effect. 1: Byte swap input.
WAIT_EN	6	Wait Pin Enable. This can be used to hold off the crypto DMA until an external memory is ready. This is useful for transferring pages from NAND flash which may take several microseconds to become ready.	0: Disabled 1: Enabled
WAIT_POL	7	Wait Pin Polarity. When the wait pin is enabled, this bit selects its active state.	0: Active low 1: Active high
WRSRC	9:8	Write FIFO Source Select. This field determines where data written to the write FIFO comes from. When data is written to the write FIFO, it is always written out the DMA. To decrypt or encrypt data, the write FIFO source should be set to the cipher output. To implement memcpy() or memset() functions, or to fill memory with random data, the write FIFO source should be set to the read FIFO. When calculating a HASH or CMAC, the write FIFO should be disabled.	00: None 01: Cipher Output 10: Read FIFO 11: Reserved

Name	Bits	Description	Settings
RDSRC	11:10	Read FIFO Source Select. This field selects the source of the read FIFO. Typically, it is set to use the DMA. To implement a memset() function, the read FIFO DMA should be disabled. To fill memory with random data or to hash random numbers, the read FIFO source should be set to the random number generator.	00: DMA disabled 01: DMA or APB 10: RNG 11: Reserved
RFU	23:12	Reserved	N/A
DMA_DONE	24	DMA Done. DMA write/read operation is complete. This bit must be cleared before starting a DMA operation. (See Warning)	0: Not Done 1: Operation Done
GLS_DONE	25	Galois Done. FIFO is full and CRC or Hamming Code Generator is enabled. This bit must be cleared before starting a CRC operation Note that DMA_DONE must be polled instead of this bit to determine the end of DMA operation during the utilization of Hamming Code Generator. (See Warning)	0: Not Done 1: Operation Done
HSH_DONE	26	Hash Done. SHA operation is complete. This bit must be cleared before starting a HASH operation. (See Warning)	0: Not Done 1: Operation Done
CPH_DONE	27	Cipher Done. Either AES or DES encryption/decryption operation is complete. This bit must be cleared before starting a cipher operation. (See Warning)	0: Not Done 1: Operation Done
MAA_DONE	28	MAA Done. MAA operation is complete. This bit must be cleared before starting a new MAA operation. This bit is read only while the MAA is in progress. This bit is negate of MAA_CTRL.STC. (See Warning)	0: Not Done 1: Operation Done
ERR	29	AHB Bus Error. This bit is set when the DMA encounters a bus error during a read or write operation. Once this bit is set, the DMA will stop. This bit can only be cleared by resetting the crypto block.	0: No Error 1: Error
RDY	30	Ready. Crypto block ready for more data	0: crypto busy 1: ready for data
DONE	31	Done. One or more cryptographic calculations complete. (logical 'OR' of done flags)	0: Not Done 1: Operation Done

Warning: The values written by software to DMA_DONE, GLS_DONE, HSH_DONE and CPH_DONE bits become the new values of these bits therefore it is impossible for software to reset (clear) one bit without affecting the values of the other bits.



11.7.2 Cipher Control Register (CIPHER_CTRL, Offset 0x0004)

Table 11-8. CIPHER_CTRL (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU	RFU	RFU[2:0]			RFU[2:0]		
Reset	0	0	000			000		
Access	R	R	RW			RW		
Position	15	14	13	12	11	10	9	8
Field	RFU[4:0]					MODE[2:0]		
Reset	0 0000					0		
Access	R					RW		
Position	7	6	5	4	3	2	1	0
Field	RFU	CIPHER[2:0]			SRC	RFU	KEY	ENC
Reset	0	000			0	0	0	0
Access	RW	RW			RW	RW	RW	RW

Name	Bits	Description	Settings
ENC	0	Encrypt. Select encryption or decryption of input data.	0: Encrypt* 1: Decrypt
KEY	1	Load Key from crypto DMA. This bit is automatically cleared by hardware after the DMA has completed loading the key. When the DMA operation is done, it sets the appropriate crypto DMA Done flag.	0: NOP 1: Initiate key loading from DMA
RFU	2	Reserved	
SRC	3	Source of Random key	0: User cipher key (0x4000_1060) 1: Key from battery-backed register file (0x4000_5000 to 0x4000_501F)*
CIPHER	6:4	Cipher Operation Select. Symmetric Block Cipher algorithm selection or memory operation.	000: Disabled 001: AES-128 010: AES-192 011: AES-256 100: DES 101: TDEA
RFU	7	Reserved	N/A
MODE	10:8	Mode Select. Mode of operation for block cipher or memory operation. DES/TDES cannot be used in CFB, OFB or CTR modes.	000: ECB 001: CBC 010: CFB (only for AES) 011: OFB (only for AES) 100: CTR (only for AES)
RFU	15:11	Reserved	N/A
RFU	18:16	Reserved	These bits must be written to zero
RFU	21:19	Reserved	These bits must be written to zero
RFU	31:22	Reserved	N/A



11.7.3 HASH Control Register (HASH_CTRL, Offset 0x0008)

Table 11-9. HASH_CTRL (Offset 0x0008)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU	RFU	LAST	HASH[2:0]			XOR	INIT
Reset	0	0	0	000			0	0
Access	R	R	RW	RW			RW	RW

Name	Bits	Description	Settings
INIT	0	Initialize. Initializes hash registers with standard constants.	0: NOP 1: Initialize Hash Values.
XOR	1	XOR. XOR data with IV from cipher block. Useful when calculating HMAC to XOR the input pad and output pad.	0 - no XOR 1 - XOR input with IV
HASH	4:2	Hash function selection.	000 - HASH disabled 001 - SHA-1 010 - SHA-224 011 - SHA-256 100 - SHA-384 101 - SHA-512 Others: Reserved.
LAST	5	Last Message Bit. This bit shall be set along with the HASH_MSG_SZ register prior to hashing the last 512 or 1024-bit block of the message data. It will allow automatic preprocessing of the last message padding, which includes the trailing bit "1", followed by the respective number of zero bits for the last block size and finally the message length represented in bytes. The bit will be automatically cleared at the same time the HASH DONE is set, designating the completion of the last message hash.	0: no effect 1: Last message data.
RFU	31:6	Reserved	N/A

Note: The automatic padding feature can only be used in terms of message bytes, not bits. Therefore, the HASH_MSG_SZ registers are in terms of bytes. In addition, the feature will automatically generate an additional "padding-only" block, if the last block of message data cannot accommodate the 64 or 128-bit padding block.

As an exception, attempting to hash a "0" message size block must include a "dummy" write to the HASH message digest register.



11.7.4 CRC Control Register (CRC_CTRL, Offset 0x000C)

Table 11-10. CRC_CTRL (Offset 0x000C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU	RFU	HRST	HAM	ENT	PRNG	MSB	CRC
Reset	0	0	0	0	0	0	0	0
Access	R	R	W	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
CRC	0	Cyclic Redundancy Check Enable. The CRC cannot be enabled if the PRNG is enabled.	0: CRC disabled 1: CRC enabled
MSB	1	MSB select. This bit selects the order of calculating CRC on data.	0: LSB data first 1: MSB data first
PRNG	2	Pseudo Random Number Generator Enable. If entropy is disabled, this outputs one byte of pseudo random data per clock cycle. If entropy is enabled, data is output at a rate of one bit per clock cycle.	0: PRNG disabled 1: PRNG enabled
ENT	3	Entropy Enable. If the PRNG is enabled, this mixes the high frequency ring oscillator with the LFSR. If the PRNG is disabled, the raw entropy data is output at a rate of 1 bit per clock. This makes it possible to characterize the quality of the entropy source.	0: No entropy 1: Add entropy
HAM	4	Hamming Code Enable. Enable hamming code calculation	0: Hamming disabled 1: Hamming enabled
HRST	5	Hamming Reset. Reset Hamming code ECC generator for next block	0: NOP 1: Reset Hamming Register
RFU	31:6	Reserved	N/A



11.7.5 Crypto DMA Source Register (DMA_SRC, Offset 0x0010)

Table 11-11. DMA_SRC (Offset 0x0010)

Position	31	30	29	28	27	26	25	24
Field	ADDR[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	ADDR[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	ADDR[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	ADDR[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description					Settings	
ADDR	31:0	DMA Source Address.						



11.7.6 Crypto DMA Destination Register (DMA_DEST, Offset 0x0014)

Table 11-12. DMA_DEST (Offset 0x0014)

Position	31	30	29	28	27	26	25	24
Field	ADDR[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	ADDR[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	ADDR[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	ADDR[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description					Settings	
ADDR	31:0	DMA Destination Address.						



11.7.7 Crypto DMA Count Register (DMA_CNT, Offset 0x0018)

Table 11-13. DMA_CNT (Offset 0x0018)

Position	31	30	29	28	27	26	25	24
Field	CNT[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	CNT[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	CNT[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	CNT[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description					Settings	
CNT	31:0	DMA Byte Count.						



11.7.8 MAA Control Register (MAA_CTRL, Offset 0x001C)

This register can only be written to when MAA is idle (STC=0). See STC bit for its write access limitation.

For each segment assignment field below, all four bits (xMA[3:0]) select the segment (0 to 9) if the word size is less than 1024 bits). If the word size is greater than or equal to 1024 bits, only the highest three bits (xMA[3:1]) is used to select segment 0 to 4, which means that the values of bits AMA0, BMA0, RMA0, and TMA0 are ignored.

Table 11-14. MAA_CTRL (Offset 0x001C)

Position	31	30	29	28	27	26	25	24
Field	TMA[3:0]				RMA[3:0]			
	0000				0000			
	RW				RW			
Position	23	22	21	20	19	18	17	16
Field	BMA[3:0]				AMA[3:0]			
	0000				0000			
	RW				RW			
Position	15	14	13	12	11	10	9	8
Field	MMS[1:0]		EMS[1:0]		BMS[1:0]		AMS[1:0]	
	00		00		00		00	
	RW		RW		RW		RW	
Position	7	6	5	4	3	2	1	0
Field	MAAER	RFU	RFU	O CALC	CLC[2:0]			STC
	0	0	0	0	000			0
	RW	RW	RW	RW	RW			RS

Name	Bits	Description	Settings
STC	0	Start Calculation. This bit functions as both the control and the status of the MAA. If the size value in the MAWS register is invalid, the STC bit will be cleared by hardware immediately. Otherwise, the STC bit is automatically cleared following the completion of each calculation or detecting an error. Clearing the STC bit resets the controller to its default state.	0: No operation. 1: Start calculation specified by CLC.
CLC	3:1	Calculation Configuration. These bits select desired calculation.	000: Exponentiation (default) 001: Square operation 010: Multiplication 011: Square followed by a multiplication 100: Addition 101: Subtraction 11x: Reserved, no operation
O CALC	4	Optimized Calculation Control. For optimized calculation, unnecessary multiply operations after normalizing the exponent is skipped.	0: No optimization 1: Optimize calculation
RFU	6:5	Reserved	N/A
MAAER	7	MAA Error. The MAAER bit defaults to 0 and can only be set by hardware. Once set, it must be cleared by software otherwise no new operation can be initiated. Software writes 1 to this bit has no effect and MAAER will maintain its original state.	0: No error. 1: Error occurs.
AMS	9:8	Multiplier A Memory Select. These bits select the starting position the parameter α within the logical segment specified by AMA.	

Name	Bits	Description	Settings
BMS	11:10	Multiplicand B Memory Select. These bits select the starting position the parameter b within the logical segment specified by BMA.	
EMS	13:12	Exponent Memory Select. These bits select the starting position the parameter e within the logical segment specified by EMA.	
MMS	15:14	Modulus Memory Select. These bits select the starting position the parameter m within the logical segment 5.	
AMA	19:16	Multiplier / Operand A Memory Assignment. These bits select the logical cryptographic RAM segment for the parameter a .	
BMA	23:20	Multiplicand / Operand B Memory Assignment. These bits select the logical cryptographic RAM segment for the parameter b .	
RMA	27:24	Result Memory Assignment. These bits select the logical cryptographic RAM segment for the parameter r .	
TMA	31:28	Temporary Memory Assignment. These bits select the logical cryptographic RAM segment for the parameter t .	



11.7.9 Crypto Data In Register 0 (CRYPTO_DIN_0, Offset 0x0020)

Crypto Data In Registers are write only and reading of any of these registers will return 0.

Table 11-15. CRYPTO_DIN_0 (Offset 0x0020)

Position	31	30	29	28	27	26	25	24
Field	DATA[31:24]							
Reset	0000 0000							
Access	W							
Position	23	22	21	20	19	18	17	16
Field	DATA[23:16]							
Reset	0000 0000							
Access	W							
Position	15	14	13	12	11	10	9	8
Field	DATA[15:8]							
Reset	0000 0000							
Access	W							
Position	7	6	5	4	3	2	1	0
Field	DATA[7:0]							
Reset	0000 0000							
Access	W							
Name	Bits	Description	Settings					
DATA	31:0	Crypto Data Input. Data input can be written to this register instead of using the DMA. This register writes to the FIFO. This register occupies four successive words to allow the use of multi-store instructions. Words can be written to any location, they will be placed in the FIFO in the order they are written. The endian swap input control bit (CRYPTO_CTRL.BSI) affects this register.						

11.7.10 Crypto Data In Register [3:1]

For details, refer to Crypto Data In Register 0.

Register	Offset	Description
CRYPTO_DIN_1	0x0024	DATA[63:32]
CRYPTO_DIN_2	0x0028	DATA[95:64]
CRYPTO_DIN_3	0x002C	DATA[127:96]



11.7.11 Crypto Data Out Register 0 (CRYPTO_DOUT_0, Offset 0x0030)

Table 11-16. CRYPTO_DOUT_0 (Offset 0x0030)

Position	31	30	29	28	27	26	25	24
Field	DATA[31:24]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	DATA[23:16]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	DATA[15:8]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	DATA[7:0]							
Reset	0000 0000							
Access	R							

Name	Bits	Description	Settings
DATA	31:0	Crypto Data Output. Resulting data from cipher calculation. Data is placed in the lower words of these four registers depending on the algorithm. For block cipher modes, this register holds the result of most recent encryption or decryption operation. These registers are affected by the endian swap bit (CRYPTO_CTRL.BSO).	

11.7.12 Crypto Data Out Register [3:1]

For details, refer to Crypto Data Out Register 0.

Register	Offset	Description
CRYPTO_DOUT_1	0x0034	DATA[63:32]
CRYPTO_DOUT_2	0x0038	DATA[95:64]
CRYPTO_DOUT_3	0x003C	DATA[127:96]

11.7.13 CRC Polynomial Register (CRC_POLY, Offset 0x0040)

Table 11-17. CRC POLY (Offset 0x0040)

Position	31	30	29	28	27	26	25	24
Field	POLY[31:24]							
Reset	1110 1101							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	POLY[23:16]							
Reset	1011 1000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	POLY[15:8]							
Reset	1000 0011							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	POLY[7:0]							
Reset	0010 0000							
Access	RW							
Name	Bits	Description					Settings	
POLY	31:0	CRC Polynomial. The polynomial to be used for Galois Field calculations (CRC or LFSR) should be written to this register. This register is affected by the MSB control bit.						

11.7.14 CRC Value Register (CRC_VAL, Offset 0x0044)

Table 11-18. CRC_VAL (Offset 0x0044)

Position	31	30	29	28	27	26	25	24
Field	VAL[31:24]							
Reset	1111 1111							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	VAL[23:16]							
Reset	1111 1111							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	VAL[15:8]							
Reset	1111 1111							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	VAL[7:0]							
Reset	1111 1111							
Access	RW							
Name	Bits	Description					Settings	
VAL	31:0	CRC Value. This is the state for the Galois Field. This register holds the result of a CRC calculation or the current state of the LFSR. This register is affected by the MSB control bit.						



11.7.15 Pseudo-Random Number Generator Register (CRC_PRNG, Offset 0x0048)

Table 11-19. CRC PRNG (Offset 0x0048)

Position	31	30	29	28	27	26	25	24
Field	PRNG[31:24]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	PRNG[23:16]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	PRNG[15:8]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	PRNG[7:0]							
Reset	0000 0000							
Access	R							
Name	Bits	Description					Settings	
PRNG	31:0	Pseudo Random Value. Output of the Galois Field shift register. This holds the resulting pseudo-random number if entropy is disabled or true random number if entropy is enabled.						



11.7.16 Hamming ECC Register (HAM_ECC, Offset 0x004C)

Table 11-20. HAM_ECC (Offset 0x004C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[6:0]							
Reset	000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	ECC[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	ECC[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description	Settings					
ECC	15:0	Hamming ECC Value. These bits are the even parity of their corresponding bit groups.						
PAR	16	Parity. This is the parity of the entire array.	0: Even parity 1: Odd parity					
RFU	31:17	Reserved	N/A					



11.7.17 Cipher Initial Vector Register 0 (CIPHER_INIT_0, Offset 0x0050)

Table 11-21. CIPHER INIT_0 (Offset 0x0050)

Position	31	30	29	28	27	26	25	24
Field	IVEC[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	IVEC[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	IVEC[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	IVEC[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description					Settings	
IVEC	31:0	Initial Vector. For block cipher operations that use CBC, CFB, OFB, or CNTR modes, this register holds the initial value. This register is updated with each encryption or decryption operation. This register is affected by the endian swap bits.						

11.7.18 Cipher Initial Vector Register [3:1]

For details, refer to Cipher Initial Vector Register 0.

Register	Offset	Description
CIPHER_INIT_1	0x0054	IVEC[63:32]
CIPHER_INIT_2	0x0058	IVEC[95:64]
CIPHER_INIT_3	0x005C	IVEC[127:96]



11.7.19 Cipher Key Register 0 (CIPHER_KEY_0, Offset 0x0060)

Cipher Key Registers are write only and reading of any of these registers will return 0.

Table 11-22. CIPHER KEY 0 (Offset 0x0060)

Position	31	30	29	28	27	26	25	24
Field	KEY[31:24]							
Reset	0000 0000							
Access	W							
Position	23	22	21	20	19	18	17	16
Field	KEY[23:16]							
Reset	0000 0000							
Access	W							
Position	15	14	13	12	11	10	9	8
Field	KEY[15:8]							
Reset	0000 0000							
Access	W							
Position	7	6	5	4	3	2	1	0
Field	KEY[7:0]							
Reset	0000 0000							
Access	W							
Name	Bits	Description					Settings	
KEY	31:0	Cipher Key. This register holds the key used for block cipher operations. The lower words are used for block ciphers that use shorter key lengths. This register is affected by the endian swap input control bits.						

11.7.20 Cipher Key Register [7:1]

For details, refer to Cipher Key Register 0.

Register	Offset	Description
CIPHER_KEY_1	0x0064	KEY[63:32]
CIPHER_KEY_2	0x0068	KEY[95:64]
CIPHER_KEY_3	0x006C	KEY[127:96]
CIPHER_KEY_4	0x0070	KEY[159:128]
CIPHER_KEY_5	0x0074	KEY[191:160]
CIPHER_KEY_6	0x0078	KEY[223:192]
CIPHER_KEY_7	0x007C	KEY[255:224]



11.7.21 HASH Message Digest Register 0 (HASH_DIGEST_0, Offset 0x0080)

Table 11-23. HASH DIGEST 0 (Offset 0x0080)

Position	31	30	29	28	27	26	25	24
Field	HASH[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	HASH[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	HASH[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	HASH[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
HASH	31:0	This register holds the calculated hash value. This register is affected by the endian swap bits.	

11.7.22 HASH Digest Register [15:1]

For details, refer to HASH Digest Register 0.

Register	Offset	Description
HASH_DIGEST_1	0x0084	HASH[63:32]
HASH_DIGEST_2	0x0088	HASH[95:64]
HASH_DIGEST_3	0x008C	HASH[127:96]
HASH_DIGEST_4	0x0090	HASH[159:128]
HASH_DIGEST_5	0x0094	HASH[191:160]
HASH_DIGEST_6	0x0098	HASH[223:192]
HASH_DIGEST_7	0x009C	HASH[255:224]
HASH_DIGEST_8	0x00A0	HASH[287:256]
HASH_DIGEST_9	0x00A4	HASH[319:288]
HASH_DIGEST_10	0x00A8	HASH[351:320]
HASH_DIGEST_11	0x00AC	HASH[383:352]
HASH_DIGEST_12	0x00B0	HASH[415:384]
HASH_DIGEST_13	0x00B4	HASH[447:416]
HASH_DIGEST_14	0x00B8	HASH[479:448]
HASH_DIGEST_15	0x00BC	HASH[511:480]



11.7.23 HASH Message Size Register 0 (HASH_MSG_SZ_0, Offset 0x00C0)

Table 11-24. HASH_MSG_SZ_0 (Offset 0x00C0)

Position	31	30	29	28	27	26	25	24
Field	MSGSZ[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	MSGSZ[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	MSGSZ[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	MSGSZ[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
MSGSZ	31:0	Message Size. This register holds the lowest 32-bit of message size in bytes.	

11.7.24 HASH Message Size Register [3:1]

For details, refer to HASH Message Size Register 0.

Register	Offset	Description
HASH_MSG_SZ_1	0x00C4	MSGSZ[63:32]
HASH_MSG_SZ_2	0x00C8	MSGSZ [95:64]
HASH_MSG_SZ_3	0x00CC	MSGSZ [124:96]



11.7.25 MAA Word Size Register (MAA_MAWS, Offset 0x00D0)

This register can only be written to when MAA is idle (STC=0).

Table 11-25. MAA_MAWS (Offset 0x00D0)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[3:0]				MAWS[11:8]			
Reset	0000				0000			
Access	R				RW			
Position	7	6	5	4	3	2	1	0
Field	MAWS [7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description	Settings					
MAWS	11:0	MAA Word Size. This register defines the number of bits for a modular operation. This register must be set to a valid value prior to the MAA operation start. Valid values are from 1 to 2048. Invalid values are ignored and will not initiate a MAA operation.						
RFU	31:12	Reserved	N/A					

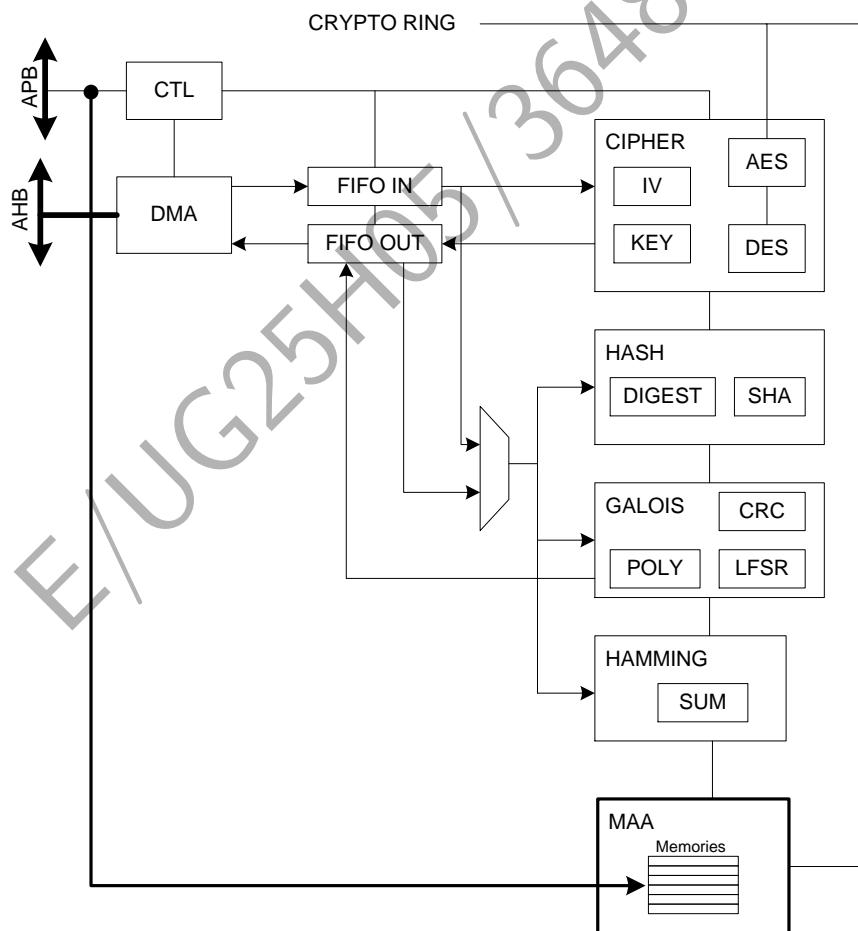
12.0 Cryptographic Accelerator (Rev B Silicon)

The cryptographic accelerator is used to assist the computationally intensive operations of several common algorithms. Supported algorithms include:

- AES-128, 192, and 256 (FIPS 197)
- DES and 3DES (NIST SP800-67)
- SHA-1, 224, 256, 384, and 512 (FIPS 180-3)
- Modulo Arithmetic Hardware Accelerator (MAA)
- Programmable CRC generator
- Hamming code generator
- Support for NIST approved block modes (SP800-38A)
- Integrated DMA with read ahead and write buffers for high throughput
- Parallel calculation of blockcipher and hash functions

The cryptographic accelerator is configured via an APB register interface.

Figure 12-1. Crypto Accelerator Block Diagram



12.1 Crypto Buffers

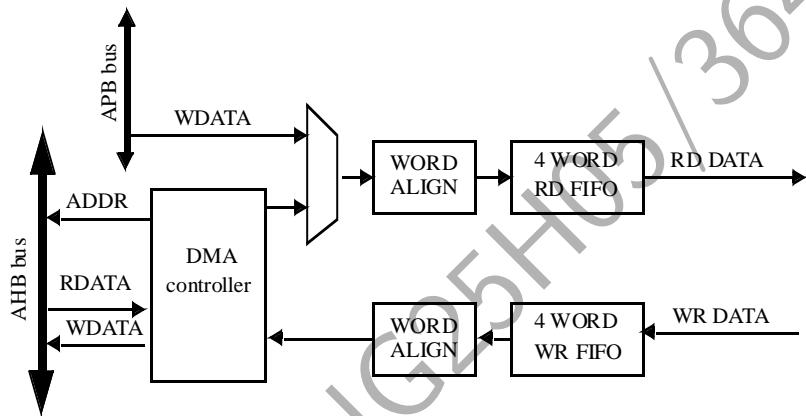
12.1.1 Direct Memory Access

To efficiently encrypt or hash data, a DMA is integrated with the crypto accelerator. The DMA starts operating when the DMA count is written with a non-zero value. Data is read from the source address and written to the

destination address. The address generators automatically increment. The source and destination of the DMA engine can point to the same memory location to encrypt or decrypt where the data is placed. The FIFOs have word alignment buffers to allow data to begin or end on byte boundaries. The DMA bursts data four words at a time with the exception of the beginning or end of the data if it is unaligned.

While the crypto accelerator is busy encrypting or hashing data, the DMA will prefetch the data for the next operation and store it in the read FIFO. Once the cipher or hash generator is done, the data for the next operation is immediately available. Data output is buffered in the write FIFO so the next cipher or hash operation can immediately begin calculating on the next block. This keeps the cipher and hash generator running continuously without having to wait for data to be written to or read from the bus. The block cipher can operate in parallel with the hash accelerator as long as only one operation uses the DMA. The hash generator input can come from the input or output FIFO.

Figure 12-2. DMA Block Diagram



12.1.2 FIFOs

The read FIFO and write FIFO have programmable sources to allow flexibility in their operation. Data written to the write FIFO is always written out the AHB DMA. The read and write FIFOs can come from the following sources.

Read FIFO

- APB
- AHB DMA
- Random Number Generator

Write FIFO

- None
- Cipher output
- Read FIFO



During crypto operation, a typical setup is to use the AHB DMA as the read FIFO source and the cipher output as the write FIFO source. This reads data from memory and writes the encrypted or decrypted result back out to memory.

A Cipher based Message Authentication Code (CMAC) is similar to digital signature or a Keyed-Hash Message Authentication Code (HMAC). It uses a cipher in a block chaining mode to form a cryptographic checksum. In this mode of operation, the AHB DMA is used as the read FIFO source, but the cipher output is not written back to memory. Only the final cipher block is of interest, so the write FIFO source should be set to none.

The DMA can be used to copy memory, similar to the memcpy standard C function by setting the write FIFO source to the read FIFO. If the Hamming ECC generator is enabled, Flash pages can be copied to memory while simultaneously calculating the error correction code.

Memory can be filled with a block of data, similar to the memset() standard C function by pointing the write FIFO source to the read FIFO and setting the read FIFO to the APB. Similarly, memory can be filled with random data by pointing the read FIFO source to the random number generator and the write FIFO source to the read FIFO.

Deterministic Random Number Generators (DRNG) such as those specified in NIST SP 800-90 can be implemented by feeding a source of entropy into a hash function or cipher based MAC. This can be done by setting the read FIFO source to the random number generator, disabling the write FIFO, and enabling the hash generator or cipher.

12.2 Symmetric Block Cipher Accelerator

The symmetric block cipher accelerator supports the Advanced Encryption Standard (AES) family of block ciphers (AES-128, AES-192, and AES-256), the Data Encryption Standard (DES) and Triple Data Encryption Algorithm (TDEA).

The symmetric block ciphers encrypt or decrypt data in blocks. The block sizes for each cipher is given in the table below.

Table 12-1. Symmetric Block Ciphers

Cipher	Key Size	Used Key Bits	Effective Strength*	Block Size	Throughput @ 96MHz
TDEA	192-bits	168-bits	112-bits	64-bits	16 MBytes/sec
AES-128	128-bits	128-bits	128-bits	128-bits	19 MBytes/sec
AES-192	192-bits	192-bits	192-bits	128-bits	16 MBytes/sec
AES-256	256-bits	256-bits	256-bits	128-bits	14 MBytes/sec

*NIST SP800-57

The block ciphers can be used in various modes of operation. In the simplest mode (ECB), each data block is simply encrypted or decrypted using the cipher. A side effect of this is that identical data blocks will encrypt to the same ciphertext. Various modes of operation are used which chain or feedback ciphertext from the previous block to seed the next encryption operation. This causes identical plaintext data blocks to encrypt to different ciphertexts. NIST SP800-38A specifies five modes of operation.

- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- Cipher Feedback (CFB)
- Output Feedback (OFB)
- Counter (CTR)



For the CFB mode of operation, the mode size is equal to the block size. 128-bit CFB is supported for AES and 64 bit CFB is supported for TDEA. 1-bit CFB and 8-bit CFB are not supported. For the CTR mode of operation, the lower 32-bits of the initial vector increment.

The cipher algorithm and mode of operation are set in the cipher control register. The cipher will start operating once the FIFO is full.

12.3 Hash Function Accelerator

The Cryptographic Hash Function accelerator supports SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 hash algorithms. It takes an input message of arbitrary length and summarizes it in a fixed length message digest.

Data is processed in 512-bit (16 word, for SHA-1, SHA-224 and SHA-256) or 1024-bit (32 word, for SHA-384 and SHA-512) blocks. After every 16 or 32 words are written to the hash block, several cycles are needed to complete the hash of that block. The number of cycles needed to compute the hash is dependent upon the number of rounds in the algorithm. The first 16 rounds are calculated as the 16 or 32 words are written to the hash block. The remaining rounds are calculated at a rate of one clock cycle per round.

Table 12-2. Hash Functions

Algorithm	Digest Length	Effective Strength*	Block Size	Throughput @ 96MHz
SHA-1	160 bits	63 bits	512 bits	66 MB/sec
SHA-224	224 bits	112 bits	512 bits	80 MB/sec
SHA-256	256 bits	128 bits	512 bits	80 MB/sec
SHA-384	384 bits	192 bits	1024 bits	133 MB/sec
SHA-512	512 bits	256 bits	1024 bits	133 MB/sec

* NIST SP800-57

The integrated DMA is capable of fetching data for the hash accelerator. Calculation of the hash can occur in parallel with the block cipher as long as only one of the operations uses the DMA.

Cache coherency needs to be managed by user software. After the crypto block reset, the CRYPTO_CTRL.RDY must be polled in the software before any other actions can start. Since the crypto DMA does not interact with the CPU caches, prior to any DMA transfer, the caches storing the source and destination memory ranges must be ‘cleaned’. Also after the DMA transfer, caches storing the destination memory range must be ‘invalidated’.

Once all data has been written to the CRYPTO Data register, the final message block must be padded according to the FIPS-180-2 standard. The standard requires that the bit length be appended to the end of the message. The length of the message in binary representation is contained in the HASH Message Size Registers (HASH_MSG_SZ). Prior to hashing the last message data, the Last Message Bit (HASH_CTRL.LAST) should be set along with the HASH_MSG_SZ registers. The HASH_MSG_SZ value will be automatically padded to the last message block.

For SHA-1 and SHA-256, a single bit equal to ‘1’ is appended to the end of the message. The message is then padded using software with zeros until 64 bits remain in the last 512-bit block. If less than 64 bits remain, then zeros are appended to the message until 64 bits remain in the next 512-bit block. The HASH_CTRL.LAST bit is set along with HASH_MSG_SZ_[1:0]. Hardware will append values in these registers to the last 64 bits of the final message block.

For SHA-384 and SHA-512, a single bit equal to ‘1’ is appended to the end of the message. The message is then padded using software with zeros until 128 bits remain in the last 1024-bit block. If less than 128 bits remain, then zeros are appended to the message until 128 bits remain in the next 1024-bit block. The HASH_CTRL.LAST bit is



set along with HASH_MSG_SZ_[3:0]. Hardware will append values in these registers to the last 128 bits of the final message block.

12.4 Galois Field Accelerator

The Galois (gal-wah) accelerator can be used to generate Cyclic Redundancy Checks (CRC) of messages or to generate pseudo random numbers using a Linear Feedback Shift Register (LFSR). Galois Fields are sometimes denoted $GF(2^n)$ where n denotes the highest order term of the reduction polynomial and 2^n denotes the size of the field.

12.4.1 Cyclic Redundancy Check

Cyclic Redundancy Checks (CRC) are commonly used for error detection. An n -bit CRC is capable of detecting the following types of errors.

- all single bit errors
- all two bit errors for block lengths less than 2^k where k is the order of the longest irreducible factor of the polynomial
- all odd number of errors for polynomials with the parity polynomial $(x+1)$ as one of its factors (polynomials with an even number of terms)
- all burst errors less than n bits
- overall, all except 1 out of 2^n errors are detected
 3. 99.998% for a 16-bit CRC
 4. 99.9999998% for a 32-bit CRC

The crypto accelerator can calculate the CRC of a block of data. Data is written to the CRYPTO Data register.

In practice, the starting initial CRC value is typically preset to all ones. If the CRC was preset to all zeros and an initial stream of all zeros is received, the CRC does not change. Therefore, most implementations preset the CRC to all ones or some constant value.

Historically, CRCs were calculated on serial bit streams. Most serial bit streams were sent least significant bit first. The CRC was calculated as each bit was transmitted or received. This resulted in the CRC being calculated on the least significant bit of the data first.

The CRC is typically appended to the end of the data. If the receiver calculates the CRC on both the data and received CRC, the result should be all zeros if the data and CRC was received error free. Most implementations do not like to check against an all zero check-sum. Therefore, most implementations invert the CRC before transmitting it. By inverting the CRC on the transmitting end, the resulting CRC on the receiving end should be a constant. The specific constant is dependent upon the CRC polynomial. This works because the non-inverted CRC calculated at the end of the data XOR'd with the received inverted CRC is all ones ($CRC \wedge \sim CRC = 1s$). Shifting all ones through the polynomial results in the same constant for each message, the constant is dependent upon the polynomial.

Since the receiving end calculates a new CRC on both the data and received CRC, the received CRC must be sent in the correct order so the highest order term of the CRC is shifted through the generator first. Since data is typically shifted through the generator LSB first, this means the CRC is reversed bit-wise so the highest order term of the remainder is in the LSB position. Software CRC algorithms typically handle this by calculating everything backwards. They will reverse the polynomial and do right shifts on the data. The resulting CRC ends up being bit swapped and in the correct format.

The CRYPTO_CRC register will be preset to all ones if the crypto block is reset. The initial CRC state can be written to any value. The final inversion must be done by software if required.

The CRC generator has a programmable polynomial up to 32-bits. The polynomial should be written to the CRC_POLY register. The largest term x^n defines the length of the CRC. When calculating the CRC on data LSB first, the polynomial should be reversed so that the coefficient of the highest power term is in the LSB position. The largest term x^n is implied (always one) and should be omitted when writing to the CRC_POLY register. This is



necessary because the polynomial is always one bit larger than the resulting CRC, so a 32-bit CRC has a polynomial with 33 terms (x^0 to x^{32}).

CRC polynomials with good error detection properties should be irreducible (the polynomial should not be factorable). Therefore, the constant term x^0 or 1 should always be present, otherwise the polynomial would be factorable by x. If the constant term x^0 or 1 were not present, the resulting CRC would be cyclic with a subgroup smaller than x^n . The effective length of the CRC would be the difference between the highest and lowest order terms. Therefore, the highest and lowest order terms x^n and x^0 should always appear in the polynomial.

When found in literature, sometimes the LSB or MSB of the polynomial is omitted when the polynomial is written in binary. It is more common to see CRC polynomials with the MSB implied because that is the bit that is shifted off, XOR'd with the data, and tested to see if the result is set. Some literature assumes the reader knows that an n-bit CRC must have the x^n term set, or else it would be a smaller length CRC.

Some common CRC polynomials and their check constants are shown in the table below. The polynomial register resets to the 32-bit CRC polynomial used by Ethernet, PPP and file compression utilities such as zip or gzip.

Table 12-3. Common CRC Polynomials

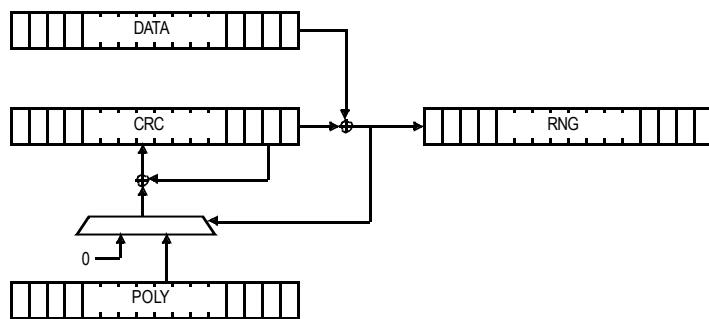
Algorithm	Polynomial Expression	Order	Polynomial	Check
CRC-32 Ethernet	$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x^1+x^0$	LSB	0xEDB88320	0xDEBB20E3
CRC-CCITT	$x^{16}+x^{12}+x^5+x^0$	LSB	0x00008408	0x0000F0B8
CRC-16	$x^{16}+x^{15}+x^2+x^0$	LSB	0x0000A001	0x0000B001
USB data	$x^{16}+x^{15}+x^2+x^0$	MSB	0x80050000	0x800D0000
Parity	x^1+x^0	LSB	0x00000001	

By default the CRC accelerator does right shifts and calculates the CRC on the LSB of the data first. The CRC can be calculated on the MSB of the data first by setting the bit-swap control bit. To calculate the CRC MSB first, you must left justify the polynomial in the CRC_POLY register. The hardware implies the MSB of the polynomial just as it did when shifting the LSB first. The LSB of the polynomial should be set, this defines the length of the CRC. The initial state of the CRC should also be left justified. When the CRC calculation is complete, it is necessary to right shift the CRC to right justify it if the polynomial is less than 32-bits.

12.4.2 Linear Feedback Shift Register

Linear Feedback Shift Registers (LFSR) are commonly used to implement Pseudo Random Number Generators (PRNG). A LFSR polynomial can be written to the CRC_POLY register to generate pseudo-random data. The starting state or seed for the pseudo random sequence should be written to the CRC register. The lockup state of all zeros is detected and the LFSR will substitute the value 1 to prevent lockup.

Figure 12-3. Galois Field CRC and LFSR Architecture



Different polynomials will generate different sequences of random data. Ideally, an n-bit polynomial will generate a random sequence of $2^n - 1$ bits. Not all polynomials are maximal length, some repeat before the theoretical maximum length of $2^n - 1$. There are thousands of different maximal length 32-bit LFSR polynomials. Any length of an LFSR



polynomial can be used up to 32-bits. Some tables of maximal length LFSR polynomials omit the MSB x^n term or the LSB $x^0=1$ terms. Fibonacci LFSRs feedback the XOR of all the taps to the constant term $x^0 = 1$ so it is often implied when listing the taps, but must be present when writing the polynomial to the CRC_POLY register.

The crypto accelerator automatically generates the next sequence of 32-bits whenever the CRYPTO_LFSR register is read. If the PRNG control bit is set, the incoming data will be forced to zero. The DMA can be used to quickly fill a block of memory with pseudo-random data.

12.4.3 Entropy

The crypto block has a high frequency ring oscillator (approximately 1GHz) which is sampled every system clock to generate entropy.

The raw output of the ring oscillator can be observed to determine how much entropy it provides. Given the probability of predicting the ring oscillator output, the amount of entropy provided per bit can be calculated according to:

$$\text{entropy} = -\log_2(\text{predictability})$$

If the output can be predicted with 99% certainty, then 0.0145 bits of entropy are provided per bit of output. In order to obtain 32 bits of entropy, a minimum of 2207 bits of output from the ring oscillator would need to be mixed. If the probability of predicting the output is 50%, 1 bit of entropy is provided per bit of output.

12.5 Hamming Code Generator

The Hamming code generator can be used to calculate an Error Correction Code (ECC) on a block of data. One application of Hamming codes is for Single Level Cell (SLC) NAND Flash memories. Multi-Level Cell (MLC) Flash memories require Error Correction Codes that can correct multiple bit errors since a corrupt cell can have multiple bit errors.

Hamming codes are capable of correcting single bit errors. An extra parity bit can be included to detect two bit errors. This is commonly referred to as Signal Error Correction, Double Error Detection (SECDED). Three errors will masquerade as a correctable single bit error.

The Hamming code generator calculates the ECC on a block of data up to 2^{16} bits in length (8k bytes). Software implementations can extend this to any length. Since the Hamming code can only correct a single bit error, increasing the block size increases the likelihood of multiple errors which are uncorrectable. The Hamming code generator in the crypto accelerator generates even parity on even halves of bit groups.

If the user wants the parity of the odd halves of the bit groups, they can XOR the parity of the even halves with the parity of the entire array. If the parity of the entire array is odd, the parity of the odd halves is the inverse of the parity of the even halves. If the parity of the entire array is even, the parity of the odd halves is identical to the parity of the even halves.

**Figure 12-4. Hamming XOR Calculation**

b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0

ECC bit0 – xor every other bit

b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0

ECC bit1 – xor every other 2 bits

b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0

ECC bit2 – xor every other 4 bits

b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0

ECC bit3 – xor every other 8 bits

b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0

ECC bit4 – xor every other 16 bits

b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0
b7	b6	b5	b4	b3	b2	b1	b0

ECC bit5 – xor every other 32 bits

For a block of data 2^n bits in length, $n+1$ ECC bits are needed for single bit error correction (ECC B0 through Bn = $n+1$ ECC bits). ECC bit n indicates parity of the entire array. If it is different than the stored ECC bit n, it indicates an error in the data array. The location of the error can be determined using the lower n ECC bits (B0 to Bn-1).

To determine if an error occurred, the user should XOR the saved and calculated ECC bits. If the result is zero, no error occurred. If the result of the ECC XOR operation is not zero, the location of the failing bit is given by the inverse of the ECC XOR result. If the failing bit location is greater than or equal to the size of the data block (bit n of the inverted ECC XOR is set), then the error is in the ECC bits. The ECC bit that is corrupt is the bit that is set as a result of the XOR of the two sets of ECC bits.

$$\text{failing_bit_location} = \sim(\text{ecc_saved} \wedge \text{ecc_calculated})$$

An error in the most significant ECC bit n will masquerade as an error in the most significant bit of the data. To properly detect and correct an error in the MSB of the ECC, the user can include an extra ECC bit ($n+2$ ECC bits). These two most significant ECC bits should be identical if they are error free. If they are different, the failing ECC bit can be determined by XORing the saved and calculated ECC bits.

This extra ECC bit can be saved at the cost of an additional parity calculation. If the MSB of the data is set, the ECC parity bits should be XOR'd with $2^{n+1}-1$ (n 1s). This effectively swaps the MSB of the data with the MSB of the ECC for purposes of parity calculation. When examining the result, if the ECC calculation indicates the MSB of the data is corrupt (the result is $2^{n+1}-1$), the error is really in the MSB of the ECC. If the result indicates the error is in the MSB of the ECC (the result is 2^{n+1}), the error is really in the MSB of the data.

ECC bits n and above are all identical. They all indicate the parity of the entire array. To use a block size of 64k bits (8k bytes), the parity bit of the entire array (bit 16) can be duplicated to obtain the higher order ECC bits.

Hamming codes can be generated over larger blocks using software. After the Hamming code is generated for an 8k block of data, software should examine the parity of the block of data just calculated (bit 16 of the ECC register). If the parity of the block is odd (parity bit is set), software should XOR additional software maintained ECC bits with the inverse of the block address. If the parity of the block is even (parity bit of the block is clear), soft-ware does not need to modify the additional ECC bits. The parity of the block (bit 16 of the ECC register) should be reset for each block, but the remaining ECC bits (B0-B15) should remain unchanged and accumulate their respective XORs throughout the entire array.

To achieve Double Error Detection (but not correction), another ECC parity bit can be included. If the result of the ECC XOR is not zero, then this extra ECC bit should also be set indicating an error has occurred. If this bit is clear



after the ECC XOR operation, it means an even number of errors has occurred and the data is not correctable. This does not mean this extra ECC bit will detect all even number of bit errors, it just indicates an even number of errors have occurred. It is possible for an even number of bit errors to masquerade as valid data. This extra ECC bit will be capable of detecting two bit errors. If an odd number of bit errors occur, they will always masquerade as a single correctable bit error. This is a limitation of using parity to indicate error. Parity can only detect an odd number of errors. For stronger error detection, a Cyclic Redundancy Check (CRC) can be used.

The lower n ECC bits (B_0 to B_{n-1}) are all that is needed to determine the location of a single bit error in the data. The next ECC bit (which is the parity of the entire array) is simply used to indicate there is an error in the data that needs to be corrected. Subsequent ECC bits are only needed to detect an error in this added ECC bit or to detect double bit errors. If an alternative error detection scheme is used, such as a CRC, then only n ECC bits are needed to find the location of a single bit error in the data array.

Since all CRCs with at least two terms in the generator polynomial will detect all single bit errors, ECC bit n which is used to determine if there is a single bit error in the array by checking parity is unnecessary. A CRC polynomial with an even number of terms has the parity polynomial $(x+1)$ as one of its factors and checks parity as well. A CRC of sufficient length is also capable of detecting all two bit errors, making an ECC bit added for double error detection unnecessary as well.

Since a single bit error in the CRC would be catastrophic, the CRC should also be protected with ECC bits. If the CRC is appended to the data for the Hamming code calculation, it can be protected with the same set of ECC bits used to protect the data by including one additional ECC bit to account for the increase in block size.

Most manufacturers of NAND Flash memories have application notes that describe the calculation of a variation of Hamming codes. These application notes calculate parity on both the odd and even halves of the bit groups. As a result, they require $2*n$ ECC bits, almost twice as many as necessary. If the even half is XOR'd with the odd half, the result is the parity of the entire array. Instead of storing parity for both the even halves and odd halves of all bit groups (which requires $2*n$ ECC bits), an implementation only really needs to store the parity of the entire array ($n+1$ ECC bits). The odd half of ECC bits can be determined by XORing the even half of ECC bits with the parity of the entire array. The parity of the entire array is just the next bit in the ECC register, so saving $n+1$ bits from this register inherently saves the parity of the entire array. Storing both the odd and even parity halves offers no more error protection than the methods described above. For both methods, an odd number of errors will masquerade as a correctable single bit error and all double bit errors will be detected.

To calculate the ECC of a new block of data, the RST bit of the CRC Control register should be set to reset the Hamming parity bits to zero and to also reset the internal byte counter.

12.6 Modular Arithmetic Accelerator (MAA)

The Modular Arithmetic Accelerator (MAA) supports IEEE Public Key Cryptography Standard (P1363) for asymmetric cryptographic operations based on DSA, RSA, and ECDSA algorithms.

The accelerator allows the user to compute a set of operations that are important parts of many cryptographic algorithms. The operations include:

- Supports up to a 2048-bit operand size
- Performs up to 2048-bit modular exponentiation ($a^e \bmod m$)
- Performs up to 2048-bit modular multiplication ($a*b \bmod m$)
- Performs up to 2048-bit modular square ($b^2 \bmod m$)
- Performs up to 2048-bit modular square followed by modular multiply ($((b^2 \bmod m) * a \bmod m)$)
- Performs up to 2048-bit modular addition ($a+b \bmod m$)
- Performs up to 2048-bit modular subtraction ($a-b \bmod m$)
- Optimized calculation mode to maximize speed
- Non-optimized calculation mode to maximize security

Using the above operations the following can also be computed:



- Performs modular inversion
- Performs modular reduction

The MAA is a peripheral designed to operate independently from the processor except when the processor is reading or writing the control register, or when it is used to load/unload the data in the specified data memory segment.

12.6.1 Operation

The MAA is controlled and accessed via the Modular Arithmetic Accelerator Control register (MAA_CTRL). The control register provides option control on arithmetic operations, data partition and status bits for start/busy. Operands and parameters are stored in a 768x16 data memory segment. This memory segment is configurable via the Memory Assignment (AMA, BMA, RMA and TMA). Per FIPS' big-endian data convention, the most significant byte or sub-word of a multi-byte word is loaded first and stored at the lowest storage location.

The calculation configuration bits (CLC) are used for MAA operation selection. Operation starts when the Start Calculation (STC) bit is set to 1. The STC bit also serves as the status bit for software monitoring during operation. Once the STC bit is set, it will remain set until either the operation is completed or an error is detected.

Software can stop an MAA operation by clearing STC to 0. If an MAA operation is terminated prematurely by software, no error or interrupt will be generated. In addition, no valid data is available.

The Optimized Calculation Control bit (OCALC) allows the user software to optimize the speed of an exponentiation by skipping unnecessary multiply operations when the corresponding exponent bit is a "0". The OCALC bit defaults to 0, forcing the MAA to operate in a non-optimized mode. The non-optimized mode skips the leading zeros of the exponent and starts square/multiply operations when a "1" has been detected. From this point on, multiply operation will be performed for every exponent bit, regardless of its logical value.

Generating encryption keywords with optimized calculations will result in fluctuation in device power dissipation. Observation of the fluctuation in power may allow observers to deduce the value of the keyword. For maximum security, non-optimized mode is recommended.

The Modular Accelerator Word Size register (MAA_MAWS) is used to define the calculation size of a modular operation. The content of the register presents the number of bits for the modular operation. For example, to perform a 1007-bit (03EFh) modular operation, this register would have to be set to 03EFh prior to setting STC.

Valid word size is from 1 to 2048. The accelerator will not start if MAWS is invalid.

The MAA Done (CRYPTO_CTRL.MAA_DONE) bit will be set to 1 after the completion of an MAA operation or when an error occurs. The MAA_DONE bit can be cleared by software writing a 0. Setting MAA_DONE to 1 by software will generate an interrupt if enabled. If Crypto Interrupt Enable (CRYPTO_CTRL.INT) bit is set to 1, an interrupt will be generated to the CPU when the MAA_DONE = 1.

12.6.2 Memory Partition

The MAA divides the crypto memory into six 256 bytes logical segments to store operands and intermediate values. These logical segments are mapped to location starting from 0x4000_1000 with offsets shown in Table 12-5.

The segment allocation is as followed:

- **a** – Multiplier/Operand A, selected by Multipler / Operand A Memory Assignment (AMA)
- **b** – Multiplicand/Operand B, selected by Multipler / Operand B Memory Assignment (BMA)
- **e** – Exponent
- **m** – Modulus
- **t** – Temporary Storage , selected by Temporary Memory Assignment (TMA)
- **r** – Result value , selected by Result Memory Assignment (RMA)

While user can allocate segment for the various parameters, the following restrictions apply:



- Only one parameter can be stored in each memory instance.
- The modulus (***m***) is always stored in memory instance 5.
- When an exponentiation operation is selected, the exponent (***e***) is always stored in memory instance 4. If another operation is selected, the exponent is not stored in any memory instance and memory instance 4 is free to hold another parameter.
- Parameters ***m***, ***b***, ***t*** and ***r*** must be stored in different memory instances (not just different segments). For example, if ***b*** is stored in memory instance 0, then neither ***t*** nor ***r*** may be stored in memory instance 1 when word size is smaller than 1024. Each memory instance is 256 bytes.

Table 12-4. MAA Memory Segments

	Memory Instance	Memory Segment (MAWS \geq 1024)	Memory Segment (MAWS < 1024)	Dedicated Function	Address Offset
xMA[3:0]	0	0	0	None	0x0100 – 0x017F
			1		0x0180 – 0x01FF
	1	1	2	None	0x0200 – 0x027F
			3		0x0280 – 0x02FF
	2	2	4	None	0x0300 – 0x037F
			5		0x0380 – 0x03FF
	3	3	6	None	0x0400 – 0x047F
			7		0x0480 – 0x04FF
	4	4	8	Exponent (if needed)	0x0500 – 0x057F
			9		0x0580 – 0x05FF
	5	5	-	Modulus	0x0600 – 0x06FF

Four of the parameters (***a***, ***b***, ***e***, and ***m***) may also be shifted in starting position within their respective memory instances by setting the appropriate Memory Select bits (AMS, BMS, EMS, MMS). Although the starting position may be altered, the entire parameter is still stored within a single memory instance, wrapping around as follows.

Table 12-5. Memory Binding (Logical Segment 0 Shown)

xMS[1:0]	Address Offset
00	0x0100 ... 0x01FF
01	0x0140 ... 0x01FF, 0x0100 ... 0x013F
10	0x0180 ... 0x01FF, 0x0100 ... 0x017F
11	0x01C0 ... 0x01FF, 0x0100 ... 0x01BF

12.6.3 Application Note

The MAA is running on internal crypto oscillator. Generating encryption keywords with optimized calculations will result in fluctuation in device power dissipation. Observation of the power fluctuation may allow observers to deduce keyword value. For maximum security, non-optimized mode is recommended.

MAA operand size is specified by the MAA Word Size Register (MAA_MAWS). Valid values are from 1 to 2048. This value specifies valid ranges for ***a***, ***b***, ***e***, and ***m***.

a, ***b***, and ***e***, can have any values between 0 and $2^{\text{MAWS}} - 1$ inclusive as long as their number of bits are less than those of ***m***. For exponentiation operation, however, ***b*** memory must contain the value of 1 and ***e*** cannot be 0.

The ***m*** memory, which holds the modulus, will have a value from $2^{\text{MAWS}-1}$ up to $2^{\text{MAWS}} - 1$. For example, for a 16-bit MAWS=0x10, it will have a value from 0x8000 (32768) up to 0xFFFF (65535)

If any parameter exceeds Word Size (MAWS), an invalid result may be generated without issuing an MAA error status. It is up to application to check for invalid word sizes.

The MAA operands are stored as 64-bit blocks. For all the memories, operands must be zero padded to the 64-bit block boundary. There is no restriction on values stored in the memories beyond this boundary. For example, MAWS=65, and operand=01 xxxx xxxx xxxx xxxxh. In this example, MAA operands are stored as two 64-bit blocks. Bit[63:0] contains the lower 64 bits. Bit[127:66] are zero padded while bit[65]=1. Data in the words above where the 0000 0000 0000 0001h is stored may have any value.

The multiplier supports unsigned operation. For modular subtraction, operand **b** must be equal to or less than operand **a**.

For most calculations, memory segments **t** and **r** will be used to store intermediate values during round of operations and may contain the same value at the final operation.

For square-multiply and exponentiation, memory segment **b** is used as an additional temporary storage area during round operations.

12.6.4 Error

The MAA Error bit (MAAER) is used to signify an error condition. MAAER is set to 1 if user attempts to: Write to MAA_CTRL or MAA_MAWS when MAA is in progress. This will also terminate the MAA operation

In the above cases, setting STC to 1 does not initiate a new MAA operation and STC remains cleared at 0. The MAAER must be cleared by software once set, otherwise no new operation can be initiated.



12.7 Cryptographic Registers

Address assignments for the Cryptographic Accelerator registers are outlined in Table 12-6.

Table 12-6. Cryptographic Accelerator Register Address (Base ADDR = 0x4000_1000)

Offset	Access	Register	Description
0x0000	RW	CRYPTO_CTRL	Crypto Control Register
0x0004	RW	CIPHER_CTRL	Cipher Control Register
0x0008	RW	HASH_CTRL	HASH Control Register
0x000C	RW	CRC_CTRL	CRC Control Register
0x0010	RW	DMA_SRC	Crypto DMA Source Register
0x0014	RW	DMA_DEST	Crypto DMA Destination Register
0x0018	RW	DMA_CNT	Crypto DMA Count Register
0x001C	RW	MAA_CTRL	MAA Control Register
0x0020	W	CRYPTO_DIN_[3:0]	Crypto Data In Register [3:0]
...			
0x002C			
0x0030	R	CRYPTO_DOUT_[3:0]	Crypto Data Out Register [3:0]
...			
0x003C			
0x0040	RW	CRC_POLY	CRC Polynomial Register
0x0044	RW	CRC_VAL	CRC Value Register
0x0048	R	CRC_PRNG	Pseudo-Random Number Generator Register
0x004C	RW	HAM_ECC	Hamming ECC Register
0x0050	RW	CIPHER_INIT_[3:0]	Cipher Initial Vector Register [3:0]
...			
0x005C			
0x0060	W	CIPHER_KEY_[7:0]	Cipher Key Register [7:0]
...			
0x007C			
0x0080	RW	HASH_DIGEST_[15:0]	HASH Message Digest Register [15:0]
...			
0x00BC			
0x00C0	RW	HASH_MSG_SZ_[3:0]	HASH Message Size Register [3:0]
...			
0x00CC			
0x00D0	RW	MAA_MAWS	MAA Word Size Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



12.7.1 Crypto Control Register (CRYPTO_CTRL, Offset 0x0000)

Table 12-7. CRYPTO_CTRL (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	DONE	RDY	ERR	MAA_DONE	CPH_DONE	HSH_DONE	GLS_DONE	DMA_DONE
Reset	0	1	0	0	0	0	0	0
Access	R	R	R	RW/RW1C*	RW/RW1C*	RW/RW1C*	RW/RW1C*	RW/RW1C*
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	DMADNEMSK	FLAG_MODE	RFU[1:0]		RDSRC[1:0]		WRSRC[1:0]	
Reset	0	0	00		00		00	
Access	RW	RW	R		RW		RW	
Position	7	6	5	4	3	2	1	0
Field	WAIT_POL	WAIT_EN	BSI	BSO	RFU	SRC	INT	RST
Reset	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
RST	0	Reset. This bit is used to reset the crypto accelerator. All crypto internal states and related registers are reset to their default reset values. Control register such as CRYPTO_CTRL, CIPHER_CTRL, HASH_CTRL, CRC_CTRL, MAA_CTRL (with the exception of the STC bit), HASH_MSG_SZ[3:0] and MAA_MAWS will retain their values. This bit will automatically clear itself after one cycle.	0: No effect 1: Reset crypto accelerator
INT	1	Interrupt Enable. Generates an interrupt when done or error set.	0: Interrupt disabled. 1: Interrupt enabled.
SRC	2	Source Select. This bit selects the hash function and CRC generator input source.	0: Input FIFO 1: Output FIFO
RFU	3	Reserved	N/A
BSO	4	Byte Swap Output. Note. No byte swap will occur if there is not a full word.	0: No effect. 1: Byte swap output.
BSI	5	Byte Swap Input. Note. No byte swap will occur if there is not a full word.	0: No effect. 1: Byte swap input.
WAIT_EN	6	Wait Pin Enable. This can be used to hold off the crypto DMA until an external memory is ready. This is useful for transferring pages from NAND flash which may take several microseconds to become ready.	0: Disabled 1: Enabled
WAIT_POL	7	Wait Pin Polarity. When the wait pin is enabled, this bit selects its active state.	0: Active low 1: Active high
WRSRC	9:8	Write FIFO Source Select. This field determines where data written to the write FIFO comes from. When data is written to the write FIFO, it is always written out the DMA. To decrypt or encrypt data, the write FIFO source should be set to the cipher output. To implement memcpy() or memset() functions, or to fill memory with random data, the write FIFO source should be set to the read FIFO. When calculating a HASH or CMAC, the write FIFO should be disabled.	00: None 01: Cipher Output 10: Read FIFO 11: Reserved



Name	Bits	Description	Settings
RDSRC	11:10	Read FIFO Source Select. This field selects the source of the read FIFO. Typically, it is set to use the DMA. To implement a memset() function, the read FIFO DMA should be disabled. To fill memory with random data or to hash random numbers, the read FIFO source should be set to the random number generator.	00: DMA disabled 01: DMA or APB 10: RNG 11: Reserved
RFU	13:12	Reserved	N/A
FLAG_MODE	14	Done Flag Mode. This bit configures the access behavior of the individual CRYPTO_CTRL Done flags (CRYPTO_CTRL[27:24]). This bit is cleared only on reset to limit upkeep, i.e. once set, it will remain set until a reset occurs.	0: Unrestricted write(0 or 1) of CRYPTO_CTRL[27:24] flags 1: Access to CRYPTO_CTRL[27:24] are "write 1 to clear/write 0 no effect"
DMADNEMSK	15	DMA Done Flag Mask. This bit masks the DMA_DONE flag from being used to generate the CRYPTO_CTRL.DONE flag, and thus disables a DMA_DONE condition from generating an interrupt. The DMA_DONE flag itself is unaffected and still may be monitored. This allows more optimal interrupt-driven crypto operations using DMA..	0: DMA_DONE not used in setting CRYPTO_CTRL.DONE bit. 1: DMA_DONE used in setting CRYPTO_CTRL.DONE bit.
RFU	23:16	Reserved	N/A
DMA_DONE	24	DMA Done. DMA write/read operation is complete. This bit must be cleared before starting a DMA operation.	0: Not Done 1: Operation Done (See Note)
GLS_DONE	25	Galois Done. FIFO is full and CRC or Hamming Code Generator is enabled. This bit must be cleared before starting a CRC operation	0: Not Done 1: Operation Done (See Note)
HSH_DONE	26	Hash Done. SHA operation is complete. This bit must be cleared before starting a HASH operation.	0: Not Done 1: Operation Done (See Note)
CPH_DONE	27	Cipher Done. Either AES or DES encryption/decryption operation is complete. This bit must be cleared before starting a cipher operation.	0: Not Done 1: Operation Done (See Note)
MAA_DONE	28	MAA Done. MAA operation is complete. This bit must be cleared before starting a new MAA operation. This bit is read only while the MAA is in progress. This bit is negate of MAA_CTRL.STC.	0: Not Done 1: Operation Done (See Note)
ERR	29	AHB Bus Error. This bit is set when the DMA encounters a bus error during a read or write operation. Once this bit is set, the DMA will stop. This bit can only be cleared by resetting the crypto block.	0: No Error 1: Error
RDY	30	Ready. Crypto block ready for more data	0: crypto busy 1: ready for data
DONE	31	Done. One or more cryptographic calculations complete. (logical 'OR' of done flags). See the CRYPTO_CTRL.DMADNEMSK bit description for further detail.	0: Not Done 1: Operation Done

Note: If FLAG_MODE = 0, the DMA_DONE, GLS_DONE, HSH_DONE and CPH_DONE bits functionality allows each flag to be written to either 0 or 1. If FLAG_MODE = 1, the DMA_DONE, GLS_DONE, HSH_DONE and CPH_DONE bits are cleared when writing a 1 to that flag, and writing a 0 has no effect on that flag.

In addition, for cipher, hash or galois operation, when RDSRC is configured for DMA, the associated DONE flags will only set after the entire DMA operation is complete with the end cipher, hash or galois block of data, i.e. intermediate operation on cipher, hash or galois blocks will not continually set the associated DONE flag. For cipher, when WRSRC is configured for cipher output, the CPH_DONE will set only after the last cipher text has completed the DMA transfer out to memory. Therefore, when using the crypto DMA for these operations, the DMA_DONE does not normally need processing and can be left masked from interrupting the CPU.



12.7.2 Cipher Control Register (CIPHER_CTRL, Offset 0x0004)

Table 12-8. CIPHER_CTRL (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field			RFU[7:0]					
Reset			0000 0000					
Access			R					
Position	15	14	13	12	11	10	9	8
Field	RFU[4:0]					MODE[2:0]		
Reset	0 0000					0		
Access	R					RW		
Position	7	6	5	4	3	2	1	0
Field	RFU	CIPHER[2:0]			SRC	RFU	KEY	ENC
Reset	0	000			0	0	0	0
Access	R	RW			RW	RW	RW	RW

Name	Bits	Description	Settings
ENC	0	Encrypt. Select encryption or decryption of input data.	0: Encrypt* 1: Decrypt
KEY	1	Load Key from crypto DMA. This bit is automatically cleared by hardware after the DMA has completed loading the key. When the DMA operation is done, it sets the appropriate crypto DMA Done flag.	0: NOP 1: Initiate key loading from DMA
RFU	2	Reserved	
SRC	3	Source of Random key	0: User cipher key (0x4000_1060) 1: Key from battery-backed register file (0x4000_5000 to 0x4000_501F)*
CIPHER	6:4	Cipher Operation Select. Symmetric Block Cipher algorithm selection or memory operation.	000: Disabled 001: AES-128 010: AES-192 011: AES-256 100: DES 101: TDEA
RFU	7	Reserved	N/A
MODE	10:8	Mode Select. Mode of operation for block cipher or memory operation.	000: ECB 001: CBC 010: CFB 011: OFB 100: CTR
RFU	31:11	Reserved	N/A



12.7.3 HASH Control Register (HASH_CTRL, Offset 0x0008)

Table 12-9. HASH_CTRL (Offset 0x0008)

Position	31	30	29	28	27	26	25	24			
Field	RFU[7:0]										
Reset	0000 0000										
Access	R										
Position	23	22	21	20	19	18	17	16			
Field	RFU[7:0]										
Reset	0000 0000										
Access	R										
Position	15	14	13	12	11	10	9	8			
Field	RFU[7:0]										
Reset	0000 0000										
Access	R										
Position	7	6	5	4	3	2	1	0			
Field	RFU	RFU	LAST	HASH[2:0]			XOR	INIT			
Reset	0	0	0	000			0	0			
Access	R	R	RW	RW			RW	RW			
Name	Bits	Description					Settings				
INIT	0	Initialize. Initializes hash registers with standard constants.					0: NOP 1: Initialize Hash Values.				
XOR	1	XOR. XOR data with IV from cipher block. Useful when calculating HMAC to XOR the input pad and output pad.					0 - no XOR 1 - XOR input with IV				
HASH	4:2	Hash function selection.					000 - HASH disabled 001 - SHA-1 010 - SHA-224 011 - SHA-256 100 - SHA-384 101 - SHA-512 Others: Reserved.				
LAST	5	Last Message Bit. This bit shall be set along with the HASH_MSG_SZ register prior to hashing the last 512 or 1024-bit block of the message data. It will allow automatic preprocessing of the last message padding, which includes the trailing bit "1", followed by the respective number of zero bits for the last block size and finally the message length represented in bytes. The bit will be automatically cleared at the same time the HASH DONE is set, designating the completion of the last message hash.					0: no effect 1: Last message data.				
RFU	31:6	Reserved					N/A				

Note: The automatic padding feature can only be used in terms of message bytes, not bits. Therefore, the HASH_MSG_SZ registers are in terms of bytes. In addition, the feature will automatically generate an additional "padding-only" block, if the last block of message data cannot accommodate the 64 or 128-bit padding block.

As an exception, attempting to hash a "0" message size block must include a "dummy" write to the HASH message digest register.



12.7.4 CRC Control Register (CRC_CTRL, Offset 0x000C)

Table 12-10. CRC_CTRL (Offset 0x000C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU	RFU	HRST	HAM	ENT	PRNG	MSB	CRC
Reset	0	0	0	0	0	0	0	0
Access	R	R	W	RW	RW	RW	RW	RW
Name	Bits	Description					Settings	
CRC	0	Cyclic Redundancy Check Enable. The CRC cannot be enabled if the PRNG is enabled.					0: CRC disabled 1: CRC enabled	
MSB	1	MSB select. This bit selects the order of calculating CRC on data.					0: LSB data first 1: MSB data first	
PRNG	2	Pseudo Random Number Generator Enable. If entropy is disabled, this outputs one byte of pseudo random data per clock cycle. If entropy is enabled, data is output at a rate of one bit per clock cycle.					0: PRNG disabled 1: PRNG enabled	
ENT	3	Entropy Enable. If the PRNG is enabled, this mixes the high frequency ring oscillator with the LFSR. If the PRNG is disabled, the raw entropy data is output at a rate of 1 bit per clock. This makes it possible to characterize the quality of the entropy source.					0: No entropy 1: Add entropy	
HAM	4	Hamming Code Enable. Enable hamming code calculation					0: Hamming disabled 1: Hamming enabled	
HRST	5	Hamming Reset. Reset Hamming code ECC generator for next block					0: NOP 1: Reset Hamming Register	
RFU	31:6	Reserved					N/A	



12.7.5 Crypto DMA Source Register (DMA_SRC, Offset 0x0010)

Table 12-11. DMA_SRC (Offset 0x0010)

Position	31	30	29	28	27	26	25	24
Field	ADDR[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	ADDR[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	ADDR[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	ADDR[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description					Settings	
ADDR	31:0	DMA Source Address.						



12.7.6 Crypto DMA Destination Register (DMA_DEST, Offset 0x0014)

Table 12-12. DMA_DEST (Offset 0x0014)

Position	31	30	29	28	27	26	25	24
Field	ADDR[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	ADDR[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	ADDR[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	ADDR[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description					Settings	
ADDR	31:0	DMA Destination Address.						



12.7.7 Crypto DMA Count Register (DMA_CNT, Offset 0x0018)

Table 12-13. DMA_CNT (Offset 0x0018)

Position	31	30	29	28	27	26	25	24
Field	CNT[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	CNT[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	CNT[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	CNT[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description					Settings	
CNT	31:0	DMA Byte Count.						



12.7.8 MAA Control Register (MAA_CTRL, Offset 0x001C)

This register can only be written to when MAA is idle (STC=0). See STC bit for its write access limitation.

For each segment assignment field below, all four bits (xMA[3:0]) select the segment (0 to 9) if the word size is less than 1024 bits). If the word size is greater than or equal to 1024 bits, only the highest three bits (xMA[3:1]) is used to select segment 0 to 4, which means that the values of bits AMA0, BMA0, RMA0, and TMA0 are ignored.

Table 12-14. MAA_CTRL (Offset 0x001C)

Position	31	30	29	28	27	26	25	24
Field	TMA[3:0]				RMA[3:0]			
	0000				0000			
	RW				RW			
Position	23	22	21	20	19	18	17	16
Field	BMA[3:0]				AMA[3:0]			
	0000				0000			
	RW				RW			
Position	15	14	13	12	11	10	9	8
Field	MMS[1:0]		EMS[1:0]		BMS[1:0]		AMS[1:0]	
	00		00		00		00	
	RW		RW		RW		RW	
Position	7	6	5	4	3	2	1	0
Field	MAAER	RFU	RFU	O CALC	CLC[2:0]			STC
	0	0	0	0	000			0
	RW	RW	RW	RW	RW			RS

Name	Bits	Description	Settings
STC	0	Start Calculation. This bit functions as both the control and the status of the MAA. If the size value in the MAWS register is invalid, the STC bit will be cleared by hardware immediately. Otherwise, the STC bit is automatically cleared following the completion of each calculation or detecting an error. Clearing the STC bit resets the controller to its default state.	0: No operation. 1: Start calculation specified by CLC.
CLC	3:1	Calculation Configuration. These bits select desired calculation.	000: Exponentiation (default) 001: Square operation 010: Multiplication 011: Square followed by a multiplication 100: Addition 101: Subtraction 11x: Reserved, no operation
O CALC	4	Optimized Calculation Control. For optimized calculation, unnecessary multiply operations after normalizing the exponent is skipped.	0: No optimization 1: Optimize calculation
RFU	6:5	Reserved	N/A
MAAER	7	MAA Error. The MAAER bit defaults to 0 and can only be set by hardware. Once set, it must be cleared by software otherwise no new operation can be initiated. Software writes 1 to this bit has no effect and MAAER will maintain its original state.	0: No error. 1: Error occurs.
AMS	9:8	Multiplier A Memory Select. These bits select the starting position the parameter α within the logical segment specified by AMA.	

Name	Bits	Description	Settings
BMS	11:10	Multiplicand B Memory Select. These bits select the starting position the parameter b within the logical segment specified by BMA.	
EMS	13:12	Exponent Memory Select. These bits select the starting position the parameter e within the logical segment specified by EMA.	
MMS	15:14	Modulus Memory Select. These bits select the starting position the parameter m within the logical segment 5.	
AMA	19:16	Multiplier / Operand A Memory Assignment. These bits select the logical cryptographic RAM segment for the parameter a .	
BMA	23:20	Multiplicand / Operand B Memory Assignment. These bits select the logical cryptographic RAM segment for the parameter b .	
RMA	27:24	Result Memory Assignment. These bits select the logical cryptographic RAM segment for the parameter r .	
TMA	31:28	Temporary Memory Assignment. These bits select the logical cryptographic RAM segment for the parameter t .	



12.7.9 Crypto Data In Register 0 (CRYPTO_DIN_0, Offset 0x0020)

Crypto Data In Registers are write only and reading of any of these registers will return 0.

Table 12-15. CRYPTO_DIN_0 (Offset 0x0020)

Position	31	30	29	28	27	26	25	24
Field	DATA[31:24]							
Reset	0000 0000							
Access	W							
Position	23	22	21	20	19	18	17	16
Field	DATA[23:16]							
Reset	0000 0000							
Access	W							
Position	15	14	13	12	11	10	9	8
Field	DATA[15:8]							
Reset	0000 0000							
Access	W							
Position	7	6	5	4	3	2	1	0
Field	DATA[7:0]							
Reset	0000 0000							
Access	W							
Name	Bits	Description	Settings					
DATA	31:0	Crypto Data Input. Data input can be written to this register instead of using the DMA. This register writes to the FIFO. This register occupies four successive words to allow the use of multi-store instructions. Words can be written to any location, they will be placed in the FIFO in the order they are written. The endian swap input control bit (CRYPTO_CTRL.BSI) affects this register.						

12.7.10 Crypto Data In Register [3:1]

For details, refer to Crypto Data In Register 0.

Register	Offset	Description
CRYPTO_DIN_1	0x0024	DATA[63:32]
CRYPTO_DIN_2	0x0028	DATA[95:64]
CRYPTO_DIN_3	0x002C	DATA[127:96]



12.7.11 Crypto Data Out Register 0 (CRYPTO_DOUT_0, Offset 0x0030)

Table 12-16. CRYPTO_DOUT_0 (Offset 0x0030)

Position	31	30	29	28	27	26	25	24
Field	DATA[31:24]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	DATA[23:16]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	DATA[15:8]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	DATA[7:0]							
Reset	0000 0000							
Access	R							

Name	Bits	Description	Settings
DATA	31:0	Crypto Data Output. Resulting data from cipher calculation. Data is placed in the lower words of these four registers depending on the algorithm. For block cipher modes, this register holds the result of most recent encryption or decryption operation. These registers are affected by the endian swap bit (CRYPTO_CTRL.BSO).	

12.7.12 Crypto Data Out Register [3:1]

For details, refer to Crypto Data Out Register 0.

Register	Offset	Description
CRYPTO_DOUT_1	0x0034	DATA[63:32]
CRYPTO_DOUT_2	0x0038	DATA[95:64]
CRYPTO_DOUT_3	0x003C	DATA[127:96]

12.7.13 CRC Polynomial Register (CRC_POLY, Offset 0x0040)

Table 12-17. CRC POLY (Offset 0x0040)

Position	31	30	29	28	27	26	25	24
Field	POLY[31:24]							
Reset	1110 1101							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	POLY[23:16]							
Reset	1011 1000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	POLY[15:8]							
Reset	1000 0011							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	POLY[7:0]							
Reset	0010 0000							
Access	RW							
Name	Bits	Description					Settings	
POLY	31:0	CRC Polynomial. The polynomial to be used for Galois Field calculations (CRC or LFSR) should be written to this register. This register is affected by the MSB control bit.						



12.7.14 CRC Value Register (CRC_VAL, Offset 0x0044)

Table 12-18. CRC_VAL (Offset 0x0044)

Position	31	30	29	28	27	26	25	24
Field	VAL[31:24]							
Reset	1111 1111							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	VAL[23:16]							
Reset	1111 1111							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	VAL[15:8]							
Reset	1111 1111							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	VAL[7:0]							
Reset	1111 1111							
Access	RW							
Name	Bits	Description					Settings	
VAL	31:0	CRC Value. This is the state for the Galois Field. This register holds the result of a CRC calculation or the current state of the LFSR. This register is affected by the MSB control bit.						



12.7.15 Pseudo-Random Number Generator Register (CRC_PRNG, Offset 0x0048)

Table 12-19. CRC PRNG (Offset 0x0048)

Position	31	30	29	28	27	26	25	24
Field	PRNG[31:24]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	PRNG[23:16]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	PRNG[15:8]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	PRNG[7:0]							
Reset	0000 0000							
Access	R							
Name	Bits	Description					Settings	
PRNG	31:0	Pseudo Random Value. Output of the Galois Field shift register. This holds the resulting pseudo-random number if entropy is disabled or true random number if entropy is enabled.						



12.7.16 Hamming ECC Register (HAM_ECC, Offset 0x004C)

Table 12-20. HAM_ECC (Offset 0x004C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[6:0]							
Reset	000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	ECC[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	ECC[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description	Settings					
ECC	15:0	Hamming ECC Value. These bits are the even parity of their corresponding bit groups.						
PAR	16	Parity. This is the parity of the entire array.	0: Even parity 1: Odd parity					
RFU	31:17	Reserved	N/A					



12.7.17 Cipher Initial Vector Register 0 (CIPHER_INIT_0, Offset 0x0050)

Table 12-21. CIPHER INIT_0 (Offset 0x0050)

Position	31	30	29	28	27	26	25	24
Field	IVEC[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	IVEC[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	IVEC[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	IVEC[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description					Settings	
IVEC	31:0	Initial Vector. For block cipher operations that use CBC, CFB, OFB, or CNTR modes, this register holds the initial value. This register is updated with each encryption or decryption operation. This register is affected by the endian swap bits.						

12.7.18 Cipher Initial Vector Register [3:1]

For details, refer to Cipher Initial Vector Register 0.

Register	Offset	Description
CIPHER_INIT_1	0x0054	IVEC[63:32]
CIPHER_INIT_2	0x0058	IVEC[95:64]
CIPHER_INIT_3	0x005C	IVEC[127:96]



12.7.19 Cipher Key Register 0 (CIPHER_KEY_0, Offset 0x0060)

Cipher Key Registers are write only and reading of any of these registers will return 0.

Table 12-22. CIPHER KEY 0 (Offset 0x0060)

Position	31	30	29	28	27	26	25	24
Field	KEY[31:24]							
Reset	0000 0000							
Access	W							
Position	23	22	21	20	19	18	17	16
Field	KEY[23:16]							
Reset	0000 0000							
Access	W							
Position	15	14	13	12	11	10	9	8
Field	KEY[15:8]							
Reset	0000 0000							
Access	W							
Position	7	6	5	4	3	2	1	0
Field	KEY[7:0]							
Reset	0000 0000							
Access	W							
Name	Bits	Description					Settings	
KEY	31:0	Cipher Key. This register holds the key used for block cipher operations. The lower words are used for block ciphers that use shorter key lengths. This register is affected by the endian swap input control bits.						

12.7.20 Cipher Key Register [7:1]

For details, refer to Cipher Key Register 0.

Register	Offset	Description
CIPHER_KEY_1	0x0064	KEY[63:32]
CIPHER_KEY_2	0x0068	KEY[95:64]
CIPHER_KEY_3	0x006C	KEY[127:96]
CIPHER_KEY_4	0x0070	KEY[159:128]
CIPHER_KEY_5	0x0074	KEY[191:160]
CIPHER_KEY_6	0x0078	KEY[223:192]
CIPHER_KEY_7	0x007C	KEY[255:224]



12.7.21 HASH Message Digest Register 0 (HASH_DIGEST_0, Offset 0x0080)

Table 12-23. HASH DIGEST 0 (Offset 0x0080)

Position	31	30	29	28	27	26	25	24
Field	HASH[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	HASH[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	HASH[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	HASH[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
HASH	31:0	This register holds the calculated hash value. This register is affected by the endian swap bits.	

12.7.22 HASH Digest Register [15:1]

For details, refer to HASH Digest Register 0.

Register	Offset	Description
HASH_DIGEST_1	0x0084	HASH[63:32]
HASH_DIGEST_2	0x0088	HASH[95:64]
HASH_DIGEST_3	0x008C	HASH[127:96]
HASH_DIGEST_4	0x0090	HASH[159:128]
HASH_DIGEST_5	0x0094	HASH[191:160]
HASH_DIGEST_6	0x0098	HASH[223:192]
HASH_DIGEST_7	0x009C	HASH[255:224]
HASH_DIGEST_8	0x00A0	HASH[287:256]
HASH_DIGEST_9	0x00A4	HASH[319:288]
HASH_DIGEST_10	0x00A8	HASH[351:320]
HASH_DIGEST_11	0x00AC	HASH[383:352]
HASH_DIGEST_12	0x00B0	HASH[415:384]
HASH_DIGEST_13	0x00B4	HASH[447:416]
HASH_DIGEST_14	0x00B8	HASH[479:448]
HASH_DIGEST_15	0x00BC	HASH[511:480]



12.7.23 HASH Message Size Register 0 (HASH_MSG_SZ_0, Offset 0x00C0)

Table 12-24. HASH_MSG_SZ_0 (Offset 0x00C0)

Position	31	30	29	28	27	26	25	24
Field	MSGSZ[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	MSGSZ[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	MSGSZ[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	MSGSZ[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
MSGSZ	31:0	Message Size. This register holds the lowest 32-bit of message size in bytes.	

12.7.24 HASH Message Size Register [3:1]

For details, refer to HASH Message Size Register 0.

Register	Offset	Description
HASH_MSG_SZ_1	0x00C4	MSGSZ[63:32]
HASH_MSG_SZ_2	0x00C8	MSGSZ [95:64]
HASH_MSG_SZ_3	0x00CC	MSGSZ [124:96]



12.7.25 MAA Word Size Register (MAA_MAWS, Offset 0x00D0)

This register can only be written to when MAA is idle (STC=0).

Table 12-25. MAA_MAWS (Offset 0x00D0)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[3:0]				MAWS[11:8]			
Reset	0000				0000			
Access	R				RW			
Position	7	6	5	4	3	2	1	0
Field	MAWS [7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description	Settings					
MAWS	11:0	MAA Word Size. This register defines the number of bits for a modular operation. This register must be set to a valid value prior to the MAA operation start. Valid values are from 1 to 2048. Invalid values are ignored and will not initiate a MAA operation.						
RFU	31:12	Reserved	N/A					



13.0 Digital to Analog Converter (DAC)

13.1 Overview

The design includes an 8-bit DAC. The DAC interface logic provides an AMBA APB/AHB slave interface and control of the DAC. The DAC controls include:

- DAC Output sample rate
- DAC Interpolation Filter

The general operating speed of the DAC is up to 333kHz and the clock source for the DAC is PLL1. Input data is 8-bit binary format. Care must be taken that the compiler interprets the write as an 8-bit write.

13.2 Interrupt

The DAC Interface generates the following interrupts if they are enabled:

- AHB FIFO Almost Empty (set when the number of values in the FIFO is less than the DAC_CTRL.DACFAECNT value).
- AHB FIFO Underflow, (set when FIFO empty)
- DAC Data Output Pattern Done (Used in Mode 1, set when DAC_RATE.DACSMPLCNT values have been outputted)

13.3 Basic DAC initialization

To initialize the DAC, the following steps need to be taken

- 1) Power up PLL1 and wait until the PLL is ready (PLL1CN.PLLRDY = 1).
- 2) Power up the DAC. (DAC_CTRL.DACPU = 1)
- 3) Wait 200 µs for the DAC to power up.

From this point, the DAC is ready for conversion.

13.4 Software Management of Conversion Rate (Mode0)

In DAC mode 0 (FIFO immediate mode), software must manage the speed of the data writes to the DAC interface. The data writes must be at least 3.0us apart.

13.5 Hardware Management of Conversion Rate (Mode1)

For DAC mode 1, the DAC output conversion rate is controlled by DAC_RATE.DACRATECNT. The calculated sampling time must be greater than 3.0us.

13.6 Interpolation Filter

When DAC_CTRL.DACMODE = 01, an interpolation filter may be enabled to enhance dynamic performance and reduce AHB bus bandwidth by lowering the input data rate to the DAC FIFO. The filter utilizes linear interpolation and supports 1:2, 1:4 and 1:8 interpolation.

Please see the DAC_CTRL.DACINTERPMODE description for enabling and selection of interpolation value. Selecting a valid interpolation value will enable the interpolation filter.

13.7 DAC Registers

Address assignments for the DAC Registers and DAC FIFO are outlined in Table 13-1 and Table 13-2 respectively.

Table 13-1. DAC Register Address (Base ADDR = 0x4003_8000)

Offset	Access	Register	Description
0x0000	RW	DAC_CTRL	DAC Control Register
0x0004	RW	DAC_RATE	DAC Output Rate and Sample Control
0x0008	RW	DAC_INT	DAC Interrupt Register

Table 13-2. DAC FIFO Address (Base ADDR = 0x400B_8000)

Offset	Access	Register	Description
0x0000	W	DAC_FIFO	DAC AHB FIFO

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



13.7.1 DAC Control Register (DAC_CTRL, Offset 0x0000)

Table 13-3. DAC_CTRL (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	DACRST	RFU[1:0]			DACPU	RFU[1:0]		
Reset	0	00			0	00		
Access	W1A	R			RW	R		
Position	23	22	21	20	19	18	17	16
Field	RFU[2:0]			CPUDACST	RFU[1:0]			DACSTMODE[1:0]
Reset	000			00	00			00
Access	R			W1A	R			RW
Position	15	14	13	12	11	10	9	8
Field	DACFFCNT[3:0]					RFU	DACINTERPMODE[2:0]	
Reset	0111					0	000	
Access	RW					R	RW	
Position	7	6	5	4	3	2	1	0
Field	DACFAE	DACFE	DACFAF	RFU	DACFAECNT[3:0]			
Reset	1	1	0	0	0100			
Access	R	R	R	R	RW			

Name	Bits	Description	Settings
DACFAECNT	3:0	DAC FIFO Almost Empty Level. This value serves as a threshold- if the write data pointer in the FIFO exceeds this count, then the DACFAE flag is set.	
RFU	4	Reserved	N/A
DACFAF	5	FIFO Almost Full	
DACFE	6	FIFO Empty	
DACFAE	7	FIFO Almost Empty	
DACINTERPMODE	10:8	DAC Interpolation Mode	000 : Disabled 001 : 1:2 Interpolation 010 : 1:4 Interpolation 011 : 1:8 Interpolation else : Reserved (default to 1:2 mode)
RFU	11	Reserved	N/A
DACFFCNT	15:12	DAC FIFO Almost Full Level. This value serves as a threshold- if the write data pointer in the FIFO exceeds this count, then the DACFAF flag is set.	
DACSTMODE	17:16	DAC Output Start Mode	00 : Start on FIFO not empty else : Reserved
RFU	19:18	Reserved	N/A
CPUDACST	20	DAC Start Strobe. The CPU may write this bit to a 1 to start a conversion.	
RFU	23:21	Reserved	NA



Name	Bits	Description	Settings
DACMODE	25:24	DAC Operation Mode	00: Output Data in FIFO as soon as available 01: Output DacSmplCnt[15:0] Data points from FIFO at Ts defined by DacRateCnt[15:0] 10: Reserved 11: Output DacSmplCnt[15:0] Data points from FIFO at Ts until mode changed
RFU	27:26	Reserved	N/A
DACPU	28	DAC Power Up. Writing this bit to a 1 will power up the DAC.	
RFU	30:29	Reserved	N/A
DACRST	31	DAC Timing Generator Reset. Setting this bit to a 1 resets the FIFO and all state machines	



13.7.2 DAC Rate Register (DAC_RATE, Offset 0x0004)

Table 13-4. DAC RATE (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	DACSMPLEN[15:8]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	DACSMPLEN[7:0]							
Reset	0000 0001							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	DACRATECNT[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	DACRATECNT[7:0]							
Reset	0000 0001							
Access	RW							

Name	Bits	Description	Settings
DACRATECNT	15:0	Delay between output samples in DACMODE=01 Ts = (DACRATECNT + 2)*(1/24MHz) based on a 12MHz crystal	
DACSAMPLECNT	31:16	Number of data points to output in DACMODE=01 if DACINTERPMODE[2:0] = 3'b000 DACSMPLEN = Number samples to output if DACINTERPMODE[2:0] > 3'b000 DACSMPLEN = (Number uninterpolated samples -1)*2^DACINTERPMODE + 1	



13.7.3 DAC Interrupt Register (DAC_INT, Offset 0x0008)

Table 13-5. DAC INT (Offset 0x0008)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[4:0]					DACFAEIE	DACFUFIE	DACDONEIE
Reset	0 0000					0	0	0
Access	R					RW	RW	RW
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[3:0]				DACFUFST	DACFAEISR	DACFUFISR	DACDONEISR
Reset	0000				0	0	0	0
Access	R				RW1C	RW1C	RW1C	RW1C
Name	Bits	Description						Settings
DACDONEISR	0	DAC output Done Interrupt status. This bit is set to a 1 by hardware when an interrupt is triggered by the DAC output Done condition.						
DACFUFISR	1	FIFO underflow interrupt status. This bit is set to a 1 by hardware when an interrupt is triggered by the DAC FIFO Underflow condition.						
DACFAEISR	2	FIFO almost empty interrupt status. This bit is set to a 1 by hardware when an interrupt is triggered by the DAC FIFO Almost Empty condition.						
DACFUFST	3	FIFO underflow status.						
RFU	15:4	Reserved						N/A
DACDONEIE	16	DAC Output Done Interrupt enable. Setting this bit to a 1 allows the DAC output condition to trigger an interrupt.						
DACFUFIE	17	FIFO underflow interrupt enable. Setting this bit to a 1 allows the FIFO underflow condition to trigger an interrupt.						
DACFAEIE	18	FIFO almost empty interrupt enable. Setting this bit to a 1 allows the FIFO almost empty condition to trigger an interrupt.						
RFU	31:19	Reserved						N/A



13.7.4 DAC AHB FIFO (DAC_FIFO, Offset 0x0000)

Table 13-6. DAC FIFO (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	DACPFO[7:0]							
Reset	0000 0000							
Access	W							

Name	Bits	Description	Settings
DACFIFO	7:0	DAC FIFO. The FIFO is located on AHB bus at 0x400B8000. The write to this location should be 8-bit write.	
RFU	31:8	Reserved	N/A

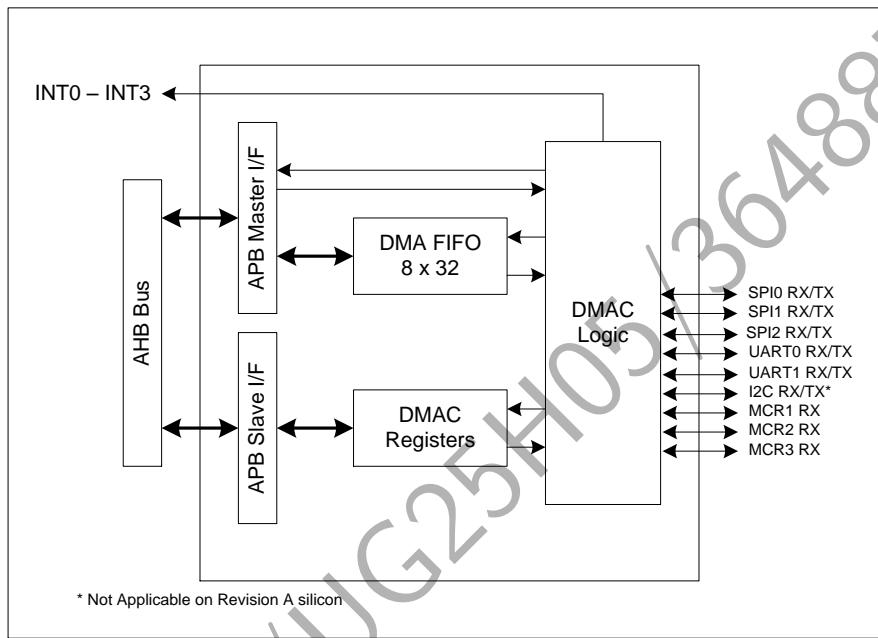
14.0 DMA Controller

The DMA Controller (DMAC) provides fully programmable, chaining capable DMA channels that support the following categories of transfer:

- 4-channel
- Peripheral-to-Memory
- Memory-to-Peripheral
- Memory-to-Memory

All DMA transactions consist of an AHB burst read into the DMA FIFO followed immediately by an AHB burst write from the FIFO.

Figure 14-1. DMA Controller Block Diagram



14.1 DMA Channel Operation

Each channel of the DMA channel is governed by a set of registers:

- Channel Configuration Register
- Channel Status Register
- Channel Source Register
- Channel Destination Register
- Channel Count Register

Additionally, each channel has a set of reload registers which allows for easy chaining of DMA buffers on count-to-zero (CTZ) condition:

- Channel Source Reload Register
- Channel Destination Reload Register
- Channel Count Reload Register

Using these registers allows each channel the following features:

- Full 32-bit source and destination addresses with 24-bit (16M-byte) address increment capability
- Up to 16Mbytes for each DMA buffer



- Programmable burst size
- Programmable priority
- Interrupt upon count-to-zero.
- Abort on error

14.2 DMA Channel Arbitration and DMA Bursts

The DMA Controller contains an internal arbiter to allow any enabled channels to access the AHB and move data. A DMA channel must be enabled before it can request from the arbiter. The Channel Enable (DMA_CFG.CHEN) bit is used to enable a channel.

When disabling a channel, the programmer must poll Channel Status (DMA_ST.CH_ST) to determine if the channel is truly disabled. In general, CH_ST will follow the setting of the DMA_CFG.CHEN. However, it will be automatically cleared under the following conditions:

- Bus error (cleared immediately)
- Count-to-zero with RLDEN=0 (cleared at the end of the AHB R/W burst).
- CHEN cleared by programmer (cleared at the end of the AHB R/W burst).

Whenever CH_ST automatically transitions from 1 to 0, the corresponding DMA_CFG.CHEN bit is also cleared (if not cleared already). During an AHB read/write burst, attempting to disable an active channel will be delayed until burst completion.

Once a channel has been programmed and enabled, it will generate a request to the arbiter either immediately (for Memory-to-Memory DMA) or whenever its associated peripheral requests DMA (for Memory-to-Peripheral or Peripheral-to-Memory DMA).

The arbiter grants requests to a single channel at a time. Granting is done first on the basis of priority – a higher priority request is always granted. Within a given priority level, requests are granted on a round-robin basis.

Once a channel's request has been granted, it will execute a DMA access in two steps:

1. Burst movement of data from the source into the DMA FIFO.
2. Burst movement of data from the DMA FIFO to the destination.

Any required data alignment is achieved during the process of moving the data in and out of the DMA FIFO.

Once granted, only an error condition will interrupt the execution of the two steps listed above. Clearing the enable bit or the occurrence of a higher priority request will not interrupt an on-going DMA transfer.

Once the two steps are complete, the channel will relinquish its grant.

Request Select (DMA_CFG.REQSEL) determines which request is used to initiate a DMA burst. In the case of memory-to- memory DMA, the channel is treated as always requesting DMA access. The Priority (PRI) bits set the DMA channel priority.

14.3 DMA Source and Destination Addressing

For memory addresses, the DMA_SRC and DMA_DST fields are used to program the addresses of the source and destination. For peripherals, however, all or part of the address is fixed based on the settings of the Request Select Field. (DMA_CFG.REQSEL). This was done to prevent data movement to/from secure parts of the device.

Table 14-1. Source and Destination Address Definition shows how the source and destination addresses as well as the address increment controls are constructed based on the “REQ SEL” setting.

“P” in the “SRCINC” or “DSTINC” field indicates that the field is programmable and set according to the bit field in the DMA_CFG. If there is a zero in the column, the field will be forced to zero.

**Table 14-1. Source and Destination Address Definition**

Request Select	Transfer	Source Address	SRC INC	Destination Address	DST INC
000000	Mem-to-Mem	DMA_SRC	P	DMA_DST	P
000001	SPI0 RX	SPI0 Data Register 0x4001_8000	0	DMA_DST	P
000010	SPI1 RX	SPI1 Data Register 0x4001_9000	0	DMA_DST	P
000011	SPI2 RX	SPI2 Data Register 0x4001_A000	0	DMA_DST	P
000100	UART0 RX	UART0 Data Register 0x4002_0020	0	DMA_DST	P
000101	UART1 RX	UART1 Data Register 0x4002_1020	0	DMA_DST	P
000110	Reserved				
000111*	I2C RX*	I2C Data Register 0x4000_C01C	0	DMA_DST	P
001000 – 001010	Reserved				
001011	MSR_DSP Track1 RX	MSR Track 1 FIFO Output Register 0x4002_B00C	0	DMA_DST	P
001100	MSR_DSP Track2 RX	MSR Track 1 FIFO Output Register 0x4002_B010	0	DMA_DST	P
001101	MSR_DSP Track3 RX	MSR Track 1 FIFO Output Register 0x4002_B014	0	DMA_DST	P
001000 – 100000	Reserved				
100001	SPI0 TX	DMA_SRC	P	SPI0 Data Register 0x4001_8000	0
100010	SPI1 TX	DMA_SRC	P	SPI1 Data Register 0x4001_9000	0
100011	SPI2 TX	DMA_SRC	P	SPI2 Data Register 0x4001_A000	0
100100	UART0 TX	DMA_SRC	P	UART0 Data Register 0x4002_0020	0
100101	UART1 TX	DMA_SRC	P	UART1 Data Register 0x4002_1020	0
100110	Reserved				
100111*	I2C TX*	DMA_SRC	P	I2C Data Register 0x4000_C01C	0
101000 – 111111	Reserved				

* Not applicable on revision A silicon

14.4 Data Movement from Source to DMA FIFO

There are several registers which are used to control the movement of data into the DMA FIFO. Source can be either a peripheral or memory.

DMA_SRC: Source address. If the increment enable is set, this increments on every read cycle of the burst.

DMA_CNT: Number of bytes to transfer before a “count-to-zero” condition occurs. This register is decremented on each read of the burst.



BRST: Burst Size (1-32). This determines the maximum number of bytes to be moved during the burst read. For the final burst, the actual number moved will never be more than the current value of DMA_CNT.

SRCWD: Source Width. This determines the maximum data width to be used during each read of the AHB burst (byte, half-word, or word). The actual AHB width may be less if DMA_CNT is not great enough to supply all of the needed bytes.

SRCINC: Source Increment Enable. This enables the incrementing of DMA_SRC. See Table 14-1 for restriction.

Table 14-2 depicts how data is moved into the DMA FIFO based on the settings of the DMA_SRC[1:0] and the “Source Width” bits. In cases where the width of the device is larger than the current value of DMA_CNT, the DMA Controller will perform an AHB cycle corresponding to a smaller width.

Table 14-2. Inbound Data Alignment

Source Device Characteristics				Register Field Values				Resulting AHB Burst	
Data Word Size	Active Byte Lanes			SRC Width		DMA_SRC[1:0]			
	3	2	1	0	1	0	0		
8			X	0	0	0	0	One AHB byte read for each byte moved. Only data from the indicated byte lane will be moved into the DMA FIFO (LSByte first).	
			X			0	1		
		X				1	0		
	X					1	1		
16			X X	0	1	0	0	One AHB half-word read for each half-word moved. Only data from the indicated byte lanes will be moved into the DMA FIFO (least significant half-word first).	
	X	X				1	0		
32	X	X	X X	1	0	0	0	One AHB word read for each word moved. The entire word is moved into the DMA FIFO.	

14.5 Data Movement from the DMA FIFO to Destination

There are several registers which are used to control the burst movement of data out of the DMA FIFO. Destination can be either a peripheral or memory.

DMA_DST: Destination Address. If the increment enable is set, this increments on every write cycle of the burst.

DMA FIFO: Counter (Internal and not accessible). THIS IS AN INTERNAL REGISTER WHICH CANNOT BE ACCESSED. This indicates the number of bytes currently held within the DMA FIFO. This register is automatically incremented on AHB reads and decremented on AHB writes.

BRST: Burst Size (1-32). This determines the maximum number of bytes to be moved during a single AHB read/write burst. The actual number moved will never be more than the current value of DMA FIFO Counter.

DSTWD: Destination Width. This determines the maximum data width to be used during each write of the AHB burst (byte, half-word, or word). The actual width will never represent a greater number of bytes than the current value of DMA FIFO Counter.

DSTINC: Destination Increment Enable. This enables the incrementing of DMA_DST. See Table 14-1 for restriction.

Table 14-3 depicts how data is moved out of the DMA FIFO based on the settings of DMA FIFO COUNTER, DMA_DST[1:0] and the “DST WIDTH”. In cases where the width of the device is larger than the number of bytes left in the DMA FIFO, the DMA Controller will perform an AHB cycle corresponding to a smaller width.

**Table 14-3. Outbound Data Alignment**

Destination Device Characteristics				Register Field Values				Resulting AHB Burst	
Data Word Size	Active Byte Lanes			DST Width		DMA_DST[1:0]			
	3	2	1	0	1	0	0		
8			X	0	0	0	0	One AHB byte write for each byte moved. Bytes will be moved out of the DMA FIFO (LSByte first) onto the indicated byte lane.	
			X			0	1		
		X				1	0		
	X					1	1		
16		X	X	0	1	0	0	One AHB half-word write for each half-word moved. Half-words will be moved out of the DMA FIFO (least significant half-word first) onto the indicated byte lanes.	
	X	X				1	0		
32	X	X	X	1	0	0	0	One AHB word write for each word moved. The entire word is moved out of the DMA FIFO.	

14.6 Memory Buffer Alignment

The DMA will dynamically adjust the transfer size to provide correct buffer alignment. There are no restrictions on the buffer alignment.

14.7 Count-to-Zero Condition

When an AHB channel burst complete, the DMAC checks to see if DMA_CNT has been decremented to 0. If so, a count-to-zero (CTZ) condition occurs.

Upon count-to-zero, depending on RLDEN, there are two possible responses:

1. RLDEn=1. DMA_SRC, DMA_DST, and DMA_CNT will be loaded from the reload registers and the channel will remain active and continue operating using the newly loaded address/count values and the previously programmed configuration values.
2. RLDEn=0. The channel will be disabled and DMA_ST.CH_ST will be cleared to 0.

14.8 Chaining Buffers

The reload registers may be used for chaining buffers together. This eases the DMA ISR response time requirements and allows the DMA to continue to service a request without immediate processing from the CPU.

The Reload Enable Bit (RLDEN=1) automate a reload of DMA_SRC, DMA_DST, and DMA_CNT upon a count-to-zero condition. The channel will remain active, the DMA_SRC, DMA_DST, and DMA_CNT will be loaded with values from the reload registers, and the DMA operation will continue with the new DMA buffer. The RDLEN bit can be set either in the DMA_CFG or DMA_CNT_RLD registers.

To configure a channel for buffer chaining both the base and reload DMA registers must be initialized prior to starting the DMA. These are the DMA_CFG, DMA_SRC, DMA_DST, DMA_CNT, DMA_SRC_RLD, DMA_DST_RLD and DMA_CNT_RLD. When the DMA_CNT_RLD register is written, the RLDEn bit must not be set. Any writes to the DMA_CFG register prior to completion of this initialization must not set the CHEN and RLDEn bits. After all registers are initialized, the last operation in the initialization is a write to the DMA_CFG register which sets both the CHEN and RLDEn bits. This will start the DMA. Subsequent DMA interrupts are serviced by writing only to the reload registers, setting DMA_CNT_RLD.RLDEN.

The CTZIEN bit must be set in the DMA_CFG register to receive an interrupt after each buffer is DMA'd. The Channel Disable Interrupt Enable bit should also be set to provide an interrupt in the case of a bus error.

Caution: Setting the CHEN via a write to the DMA_CFG register and separately setting the RLDEN bit via a write to the DMA_RLD_CNT register risks a race condition between a DMA completion interrupt service routine initializing the reload registers (for the third buffer) before the software initialization of these registers for the second buffer.

When the first DMA transfer completes (based on DMA_CNT value), a count to zero interrupt will occur, the DMA_SRC, DMA_DST and DMA_CNT registers will be reloaded from the corresponding reload registers. The DMA_ST register will indicate both “reload” and “count to zero” events occurred. The Channel Status bit (CH_ST) will typically be a 1, indicating the DMA is now busy with the second DMA transfer defined in the reload registers. If CH_ST=0, this indicates both the initial and second DMA transfers have completed. If there are additional buffers to chain, the interrupt service routine will initialize the DMA_SRC_RLD, DMA_DST_RLD, and DMA_CNT_RLD registers, setting DMA_CNT_RLD.RLDEN bit as the last operation. The interrupt service routine will not write to the DMA_CFG, DMA_SRC, DMA_DST and DMA_CNT registers, just the reload registers.

To prevent improper operation, program the address bits before setting the RLDEN bit.

14.9 DMA Interrupts

Interrupts from each channel can be individually enabled by setting the corresponding interrupt enable bits (DMA_CN.CHIEN). When an interrupt is pending, the corresponding Channel Interrupt Pending (DMA_INT.IPEND) bit is set to 1. This is a read-only status. The interrupt must be cleared in the DMA_ST.

A channel interrupt (DMS_ST.IPEND=1) can be caused by:

- Count-to-zero interrupt (DMA_CFG.CTZIEN=1). If enabled, all count-to-zero occurrences set IPEND.
- Channel Disable interrupt (DMA_CFG.CHDIEN=1). If enabled, any clearing of CH_ST will set PEND. The DMA_ST can be examined to see which reason(s) caused the disable. The CHDIEN also enables Time-out interrupt. The Time-out interrupt does not clear the CH_ST bit

To clear the channel interrupt, write 1 to the cause of the interrupt (CTZ_ST, RLD_ST, BUS_ERR, TO_ST).

When running in normal mode without buffer chaining (RLDEN = 0), it is recommended to set only the DMA_CFG.CHDIEN. An interrupt will be generated upon DMA completion or an error condition (bus error or time-out error).

When running in buffer chaining mode (RLDEN=1), it is recommended to set both the CHDIEN and CTZIEN bits. The count-to-zero interrupts will occur on completion of each DMA (count reaches zero and reload occurs). The setting of CHDIEN insures that an error condition will also generate an interrupt. If the CTZIEN=0, the only DMA completion interrupt will be when the DMA completes and RLDEn=0 (final DMA).

14.10 Channel Time-outs

Each channel can be configured to generate an interrupt when its associated request line is inactive for a given period of time. An example use of this feature is to determine an idle UART receive channel. The time-out mechanism consists of a single global 24-bit prescaler and per-channel programmable 8-bit timer/counters.

The Channel Disable Interrupt Enable bit must be set for the Time-out to generate an interrupt.

14.10.1 Global 24-bit PreScaler

A single global 24-bit clock prescaler is used to divide hclk. There are three taps provided from this divider:

- hclk / 256
- hclk / 64k
- hclk / 16M



14.10.2 8-bit Timer/Counter

Each channel has an individual 8-bit timer/counter. The Prescale Select (DMA_CFG.PSSEL) bits select the prescaler taps used as timer clock source.

The Time-out Select (DMA_CFG.TOSEL) bits set the limit at which the 8-bit timer/counter should count before generating an interrupt.

The 8-bit timer/counter is reset whenever any of the following conditions occurs:

1. The DMA request line programmed for the channel is activated.
2. The channel is disabled for any reason (DMA_ST.CH_ST=0).

Any of the 8-bit timer can be disabled by setting the PSSEL bits to 00b. When all channels of the DMAC are disabled, the global prescaler is also disabled.

Normally, the 8-bit timer starts counting as soon as the channel is enabled and the PSSEL bits are non-zero. But if Request Wait is set (DMA_CFG.REQWAIT=1), the timer starts counting only after the first DMA request is received from the peripheral.

The time-out period can be calculated using the following equation:

$$T_{\text{timeout}} = T_{\text{hclk}} * N_{\text{pssel}} * N_{\text{tosel}}$$

For an hclk frequency of 90MHz, and PSSEL=10b and TOSEL=100b, the calculation would be:

$$T_{\text{timeout}} = (1/90\text{MHz}) * 65536 * 32 = 23.3\text{ms}$$

14.11 Channel and Register Access Restrictions

The programmer is free to write to any registers while a channel is disabled, but there are certain restrictions when a channel is enabled. The Channel Status bit indicates whether the channel is enabled or not.

Because an active channel may be in the middle of an AHB read/write burst, the user should not write to DMA_SRC, DMA_DST, or DMA_CNT while a channel is active (CH_ST=1).

The programmer can disable any DMA channel by clearing CHEN. The user must then poll CH_ST to verify that the channel is disabled. When clearing Channel Enable bit, the user should perform a read-modify-write to ensure that other bits of the DMA_CFG are not modified.

CAUTION: When disabling a DMA channel, ensure that system interrupts are configured appropriately.

14.12 Memory-to-Memory DMA

Memory-to-memory transfers are performed as if the request is always active. This means that the DMA channel will generate an almost constant request for the bus until its transfer is complete. For this reason, the user should assign a lower priority to channels executing memory-to-memory transfers to prevent starvation of other DMA channels.



14.13 DMA Registers

Address assignments for the DMA registers are outlined in Table 14-4 and Table 14-5.

Table 14-4. DMA Register Address (Base ADDR = 0x4002_8000)

Offset	Access	Register	Description
0x0000	RW	DMA_CN	DMA Control Register
0x0004	RW	DMA_INT	DMA Interrupt Status Register
0x0100	RW	DMA0	DMA Channel 0 Base Address
0x0120	RW	DMA1	DMA Channel 1 Base Address
0x0140	RW	DMA2	DMA Channel 2 Base Address
0x0160	RW	DMA3	DMA Channel 3 Base Address

Each channel has a set of associated registers. The relative offsets for these registers with respect to the channel base address are outlined in Table 14-5.

Table 14-5. DMA Channel Registers Offset

Offset	Access	Register	Description
0x0000	RW	DMAx_CFG	DMA Channel x Configuration Register
0x0004	RW	DMAx_ST	DMA Channel x Status Register
0x0008	RW	DMAx_SRC	DMA Channel x Source Register
0x000C	RW	DMAx_DST	DMA Channel x Destination Register
0x0010	RW	DMAx_CNT	DMA Channel x Count Register
0x0014	RW	DMAx_SRC_RLD	DMA Channel x Source Reload Register
0x0018	RW	DMAx_DST_RLD	DMA Channel x Destination Reload Register
0x001C	RW	DMAx_CNT_RLD	DMA Channel x Count Reload Register

Since the registers are identical for all the channels, only registers associated with DMA Channel 0 is described in this document as the same definition applies to all channels.

All DMA channels operate identically. Only Channel 0 registers are described. For all other channels operation, please refer to Channel 0 registers description.

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



14.13.1 DMA Control Register (DMA_CN, Offset 0x0000)

Table 14-6. DMA_CN (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[3:0]				CHIEN[3:0]			
Reset	0000				0000			
Access	R				RW			
Name	Bits	Description					Settings	
CHIEN	3:0	Channel Interrupt Enable. Each bit enables the corresponding channel interrupt.					0: Disable 1: Enable	
RFU	31:4	Reserved					N/A	



14.13.2 DMA Interrupt Register (DMA_INT, Offset 0x0004)

Table 14-7. DMA_INT (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[3:0]				IPEND[3:0]			
Reset	0000				0000			
Access	R				R			
Name	Bits	Description					Settings	
IPEND	3:0	Channel Interrupt. Each bit represents interrupt from the corresponding channel. To clear an interrupt, all active interrupt bits of the DMA_ST must be cleared. The interrupt bits are set only if their corresponding interrupt enable bits are set in DMA_CN.					0: No Interrupt 1: Interrupt Pending	
RFU	31:4	Reserved					N/A	



14.13.3 DMA Channel 0 Configuration Register (DMA0_CFG, Offset 0x0100)

Table 14-8. DMA0_CFG (Offset 0x0100)

Position	31	30	29	28	27	26	25	24
Field	CTZIEN	CHDIEN	RFU	BRST [4:0]				
Reset	0	0	0	0 0000				
Access	RW	RW	R	RW				
Position	23	22	21	20	19	18	17	16
Field	RFU	DSTINC	DSTWD[1:0]		RFU	SRCINC	SRCWD[1:0]	
Reset	0	0	00		0	0	00	
Access	R	RW	RW		R	RW	RW	
Position	15	14	13	12	11	10	9	8
Field	PSSEL[1:0]		TOSEL[2:0]			REQWAIT	REQSEL[5:4]	
Reset	00		000			0	00	
Access	RW		RW			RW	RW	
Position	7	6	5	4	3	2	1	0
Field	REQSEL[3:0]				PRI[1:0]		RLDEN	CHEN
Reset	0000				00		0	0
Access	RW				RW		RW	RW

Name	Bits	Description	Settings
CHEN	0	Channel Enable. This bit is automatically cleared when DMA_ST.CH_ST changes from 1 to 0.	0: Disable this channel. 1: Enable this channel.
RLDEN	1	Reload Enable. Setting this bit to 1 enables DMA_SRC, DMA_DST and DMA_CNT to be reloaded with their corresponding reload registers upon count-to-zero. This bit is also writeable in the Count Reload Register. Refer to the description on Buffer Chaining for use of this bit. If buffer chaining is not used this bit must be written with a 0. This bit should be set after the reload registers have been programmed.	0: Disabled 1: Enabled.
PRI	3:2	DMA Priority.	00: Highest Priority 11: Lowest Priority
REQSEL	9:4	Request Select. Select DMA request line for this channel. If memory-to-memory is selected, the channel operates as if the request is always active. Refer to Table 14-1 for the decode for this field.	
REQWAIT	10	Request Wait Enable. When enabled, delay timer start until DMA request transitions from active to inactive.	0: Start timer normally. 1: Delay timer start.
TOSEL	13:11	Time-Out Select. Selects the number of prescale clocks seen by the channel timer before a time-out conditions is generated for this channel. Important note: since the prescaler runs independent of the individual channel timers, the actual number of Pre-Scale clock edges seen has a margin of error equal to a single Pre-Scale clock. This is reflected in the settings column (e.g. 3-4, not just 4).	000: 3-4 001: 7-8 010: 15-16 011: 31-32 100: 63-64 101: 127-128 110: 255-256 111: 511-512
PSSEL	15:14	Pre-Scale Select. Selects the Pre-Scale divider for timer clock input	00: Disable timer 01: hclk / 256 10: hclk / 64k 11: hclk / 16M

Name	Bits	Description	Settings
SRCWD	17:16	Source Width. In most cases, this will be the data width of each AHB transactions. However, the width will be reduced in the cases where DMA_CNT indicates a smaller value.	00: Byte 01: Half-word 10: Word 11: Reserved (Byte width if set)
SRCINC	18	Source Increment Enable. This bit enables DMA_SRC increment upon every AHB transaction. This bit is forced to 0 for DMA receive from peripherals.	0: Increment disabled. 1: Increment enabled.
RFU	19	Reserved.	N/A
DSTWD	21:20	Destination Width. Indicates the width of the each AHB transactions to the destination peripheral or memory. (The actual width may be less than this if there are insufficient bytes in the DMA FIFO for the full width).	00: Byte 01: Half-word 10: Word 11: Reserved (Byte width if set)
DSTINC	22	Destination Increment Enable. This bit enables DMA_DST increment upon every AHB transaction. This bit is forced to 0 for DMA transmit to peripherals.	0: Increment disabled. 1: Increment enabled.
RFU	23	Reserved.	N/A
BRST	28:24	Burst Size. The number of bytes to be transferred into and out of the DMA FIFO in a single burst.	00000: 1 byte 00001: 2 bytes 00010: 3 bytes ... 11111: 32 bytes
RFU	29	Reserved.	N/A
CHDIEN	30	Channel Disable Interrupt Enable. When enabled, the IPEND will be set to 1 whenever CH_ST changes from 1 to 0.	0: Interrupt disabled. 1: Interrupt enabled.
CTZIEN	31	Count-to-zero Interrupts Enable. When enabled, the IPEND will be set to 1 whenever a count-to-zero event occurs.	0: Interrupt disabled. 1: Interrupt enabled.

14.13.4 DMA Channel 0 Status Register (DMA0_ST, Offset 0x0104)

Table 14-9. DMA0_ST (Offset 0x0104)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU	TO_ST	RFU	BUS_ERR	RLD_ST	CTZ_ST	IPEND	CH_ST
Reset	0	0	0	0	0	0	0	0
Access	R	RW1C	R	RW1C	RW1C	RW1C	R	R

Name	Bits	Description	Settings
CH_ST	0	Channel Status. This bit is used to indicate to the programmer when it is safe to change the configuration, address, and count registers for the channel. Whenever this bit is cleared by hardware, the DMA_CFG.CHEN bit is also cleared (if not cleared already).	0: Channel disabled. 1: Channel enabled.
IPEND	1	Channel Interrupt.	0: No interrupt 1: Interrupt pending
CTZ_ST	2	Count-to-Zero (CTZ) Status.	Read: 0: CTZ has not occurred. 1: CTZ has occurred. Write: 0: No effect 1: Write 1 to clear
RLD_ST	3	Reload Status.	Read: 0: Reload has not occurred. 1: Reload has occurred. Write: 0: No effect 1: Write 1 to clear
BUS_ERR	4	Bus Error. Indicates that an AHB abort was received and the channel has been disabled.	Read: 0: No error found 1: An AHB bus error occurred Write: 0: No effect 1: Write 1 to clear
RFU	5	Reserved	N/A



Name	Bits	Description	Settings
TO_ST	6	Time-Out Status.	Read: 0: No time-out 1: A time-out has occurred Write: 0: No effect 1: Write 1 to clear
RFU	31:7	Reserved	N/A

E/UG25H05/36488764



14.13.5 DMA Channel 0 Source Register (DMA0_SRC, Offset 0x0108)

Table 14-10. DMA0_SRC (Offset 0x0108)

Position	31	30	29	28	27	26	25	24
Field	SRC[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	SRC[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	SRC[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	SRC[7:0]							
Reset	000 0000							
Access	RW							

Name	Bits	Description	Settings
SRC	31:0	Source Device Address. If SRCINC=1, the counter bits are incremented by 1,2, or 4, depending on the data width of each AHB cycle. For peripheral transfers, some or all of the actual address bits are fixed (see Table 14-1). If SRCINC=0, this register remains constant. In the case where a count-to-zero condition occurs while RLDEN=1, the register is reloaded with the contents of DMA_SRC_RLD.	

14.13.6 DMA Channel 0 Destination Register (DMA0_DST, Offset 0x010C)

Table 14-11. DMA0_DST (Offset 0x010C)

Position	31	30	29	28	27	26	25	24
Field	DST[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	DST[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	DST[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	DST[7:0]							
Reset	000 0000							
Access	RW							

Name	Bits	Description	Settings
DST	31:0	Destination Device Address. For peripheral transfers, some or all of the actual address bits are fixed (see Table 14-1). If DSTINC=1, this register is incremented on every AHB write out of the DMA FIFO. They are incremented by 1,2, or 4, depending on the data width of each AHB cycle. In the case where a count-to-zero condition occurs while RLDEN=1, the register is reloaded with DMA_DST_RLD.	



14.13.7 DMA Channel 0 Count Register (DMA0_CNT, Offset 0x0110)

Table 14-12. DMA0_CNT (Offset 0x0110)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	CNT[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	CNT[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	CNT[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description	Settings					
CNT	23:0	DMA Counter. The user loads this register with the number of bytes to transfer. This counter decreases on every AHB cycle into the DMA FIFO. The decrement will be 1, 2, or 4 depending on the data width of each AHB cycle. When the counter reaches '0', a count-to-zero condition is triggered.	0x000000: 0 Byte 0x000001: 1 Byte 0x000002: 2 Bytes ... 0xFFFFFFF: 16,777,215 Bytes					
RFU	31:24	Reserved	N/A					



14.13.8 DMA Channel 0 Source Reload Register (DMA0_SRC_RLD, Offset 0x0114)

Table 14-13. DMA0_SRC_RLD (Offset 0x0114)

Position	31	30	29	28	27	26	25	24
Field	RFU			SRC_RLD[30:24]				
Reset	0			000 0000				
Access	R			RW				
Position	23	22	21	20	19	18	17	16
Field				SRC_RLD[23:16]				
Reset				0000 0000				
Access				RW				
Position	15	14	13	12	11	10	9	8
Field				SRC_RLD[15:8]				
Reset				0000 0000				
Access				RW				
Position	7	6	5	4	3	2	1	0
Field				SRC_RLD[7:0]				
Reset				0000 0000				
Access				RW				
Name	Bits	Description					Settings	
SRC_RLD	30:0	Source Address Reload Value. The value of this register is loaded into DMA0_SRC upon a count-to-zero condition.						
RFU	31	Reserved					N/A	



14.13.9 DMA Channel 0 Destination Reload Register (DMA0_DST_RLD, Offset 0x0118)

Table 14-14. DMA0_DST_RLD (Offset 0x0118)

Position	31	30	29	28	27	26	25	24
Field	RFU			DST_RLD[30:24]				
Reset	0			000 0000				
Access	R			RW				
Position	23	22	21	20	19	18	17	16
Field				DST_RLD[23:16]				
Reset				0000 0000				
Access				RW				
Position	15	14	13	12	11	10	9	8
Field				DST_RLD[15:8]				
Reset				0000 0000				
Access				RW				
Position	7	6	5	4	3	2	1	0
Field				DST_RLD[7:0]				
Reset				0000 0000				
Access				RW				

Name	Bits	Description	Settings
DST_RLD	30:0	Destination Address Reload Value. The value of this register is loaded into DMA0_DST upon a count-to-zero condition.	
RFU	31	Reserved	N/A

14.13.10 DMA Channel 0 Count Reload Register (DMA0_CNT_RLD, Offset 0x011C)

Table 14-15. DMA0_CNT_RLD (Offset 0x011C)

Position	31	30	29	28	27	26	25	24
Field	RLDEN				RFU[6:0]			
Reset	0				000 0000			
Access	RW				R			
Position	23	22	21	20	19	18	17	16
Field					CNT_RLD[23:16]			
Reset					0000 0000			
Access					RW			
Position	15	14	13	12	11	10	9	8
Field					CNT_RLD[15:8]			
Reset					0000 0000			
Access					RW			
Position	7	6	5	4	3	2	1	0
Field					CNT_RLD[7:0]			
Reset					0000 0000			
Access					RW			

Name	Bits	Description	Settings
CNT_RLD	23:0	Count Reload Value. The value of this register is loaded into DMA0_CNT upon a count-to-zero condition.	
RFU	30:24	Reserved	N/A
RLDEN	31	Reload Enable. This bit should be set after the address reload registers have been programmed. This bit is automatically cleared to 0 when reload occurs.	0: Reload disabled. 1: Reload enabled.



15.0 Flash Controller

MAX32550 has 1MB of Flash memory divided into 256 pages respectively. Size of each page is 4096B. The address span of the Flash memory is shown in Table 2-1. Please refer to the Flash Controller register section.

15.1 Flash Memory Operations

15.1.1 Flash Write

Flash write address is written to the Flash Address Register (FLSH_ADDR) and the data into the Flash Data Register (FLSH_DATA). Application then sets the Write bit (WR) to initiates the write. The WR bit clears itself when the write operation completes.

There are two data write formats: 32-bit and 128-bit. This is selected by the Data Width bit (WDTH).

When WDTH=1, data length is 32-bit and FLSH_ADDR[31:2] are valid address bits. Write data is to be placed in FLSH_DATA0.

When WDTH=0, data length is 128-bit and FLSH_ADDR[31:4] are valid address bits. Write data is to be placed in FLSH_DATA0 to FLSH_DATA3. FLSH_DATA0 will correspond to the least significant 32-bit of data whereas FLSH_DATA3 will correspond to the most significant 32-bit of data. In this mode, after the write operation is initiated, the Flash controller will program the data in sequence and will only clear the WR bit when all of the 128-bits are programmed.

15.1.2 Flash Page Erase

Flash page erase address is written to the Flash Address Register (FLSH_ADDR). Only FLSH_ADDR[19:12] are valid. Application then sets the Page Erase bit (PGE) to initiates the erase. The PGE bit clears itself when the operation completes.

15.1.3 Flash Mass Erase

Application sets the Mass Erase bit (ME) to initiates the erase. The ME bit clears itself when the operation completes.

15.1.4 Burst Write

To reduce the total programming time for the entire Flash, Burst Write is supported by the Flash controller. Burst mode is enabled by Burst Mode bit (FLSH_CN.BRST). In this mode, data length is always 128-bit.

The burst write cycle is very similar to a single write cycle from programmer's point of view. The FLSH_ADDR register is first set to the Flash address to be programmed. The data to be programmed is then written to the FLSH_DATAx register followed by setting the WR bit. The WR bit must then be polled until it is self-cleared. This indicates that the next data in the burst can be written to the data registers. FLSH_ADDR is auto incremented at the end of the write operation.

15.1.5 Register Access Restriction

Write to any of the Flash registers are ignored when the Flash is busy as indicated by any of the PGE, ME or WR bits being set, only one of which can be set at any one time.

15.1.6 Interrupt

The Flash Done Interrupt (FLSH_INT.DONE) bit is set at the end of a Flash operation. The DONE bit is set whether the operation succeed or not. In the case of an error, the Access Fail bit (FLSH_CN.AF) bit will also be set to 1.

15.2 Flash Controller Registers

Address assignments for the Flash Controller Registers are outlined in Table 15-1.

Table 15-1. Flash Controller Register Address (Base ADDR = 0x4002_9000)

Offset	Access	Register	Description
0x0000	RW	FLSH_ADDR	Flash Address Register
0x0004	RW	FLSH_CLKDIV	Flash Clock Divide Register
0x0008	RW	FLSH_CN	Flash Control Register
0x000C ... 0x0020	RW	RFU	
0x0024	R	FLSH_INT	Flash Interrupt Register
0x0028 ... 0x002C	R	RFU	
0x0030	RW	FLSH_DATA0	Flash Data Register 0
0x0034	RW	FLSH_DATA1	Flash Data Register 1
0x0038	RW	FLSH_DATA2	Flash Data Register 2
0x003C	RW	FLSH_DATA3	Flash Data Register 3
0x0040	RW	ACNTL	Access Control Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



15.2.1 Flash Address Register (FLSH_ADDR, Offset 0x0000)

Table 15-2. FLSH_ADDR (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	ADDR[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	ADDR[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	ADDR[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	ADDR[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description					Settings	
ADDR	31:0	Flash Write Address.						



15.2.2 Flash Clock Divide Register (FLSH_CLKDIV, Offset 0x0004)

Table 15-3. FLSH_CLKDIV (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	CLKDIV[7:0]							
Reset	0110 0100							
Access	RW							
Name	Bits	Description						Settings
CLKDIV	7:0	Flash Clock Divide. The clock (PLL0) is divided by this value to generate a 1 MHz clock for Flash controller.						
RFU	31:8	Reserved.						N/A



15.2.3 Flash Control Register (FLSH_CN, Offset 0x0008)

Table 15-4. FLSH_CN (Offset 0x0008)

Position	31	30	29	28	27	26	25	24
Field	UNLOCK[3:0]					BRST	RFU[1:0]	
Reset	0000					0	00	
Access	RW					RW	R	
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	ERASE_CODE[7:0]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	RFU[2:0]			WDTH	RFU	PGE	ME	WR
Reset	000			0	0	0	0	0
Access	R			RW	R	RS	RS	RS
Name	Bits	Description					Settings	
WR	0	Write. This bit is automatically cleared after the operation.					0: No operation. 1: Initiate Write Operation.	
ME	1	Mass Erase. This bit is automatically cleared after the operation.					0: No operation. 1: Initiate Mass Erase.	
PGE	2	Page Erase. This bit is automatically cleared after the operation.					0: No operation. 1: Initiate Page Erase.	
RFU	3	Reserved					N/A	
WDTH	4	Data Width. This bits selects write data width.					0: 128-bit 1: 32-bit	
RFU	7:5	Reserved					N/A	
ERASE_CODE	8:15	Erase Code. The ERASE_CODE must be set up properly before erase operation can be initiated. These bits are automatically cleared after the operation is complete.					0x00 - No operation. 0x55 – Enable Page Erase. 0xAA – Enable Mass Erase. The debug port must be enabled.	
RFU	23:16	Reserved					N/A	
PEND	24	Flash Pending. When Flash operation is in progress (busy), Flash reads and writes will fail. When PEND is set, write to all Flash registers, with exception of the Flash interrupt register, are ignored.					0: Flash Idle. 1: Flash Busy.	
RFU	26:25	Reserved					N/A	
BRST	27	Burst Mode Enable.					0: Disabled. 1: Enabled.	
UNLOCK	31:28	Flash Unlock. The correct unlock code must be written to these four bits before any Flash write or erase operation is allowed.					0010: Flash unlocked. Others: Flash locked.	



15.2.4 Flash Interrupt Register (FLSH_INT, Offset 0x0024)

Table 15-5. FLSH_INT (Offset 0x0024)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[5:0]						AFIE	DONEIE
Reset	0000 00						0	0
Access	R						RW	RW
Position	7	6	5	4	3	2	1	0
Field	RFU[5:0]						AF	DONE
Reset	0000 00						0	0
Access	R						RC	RW
Name	Bits	Description						
DONE	0	Flash Done Interrupt. This bit is set to 1 upon Flash write or erase completion.	0: No Interrupt. 1: Interrupt pending.					
AF	1	Flash Access Failure Interrupt. This bit is set when an attempt is made to write the flash while the flash is busy or the flash is locked. This bit can only be set to '1' by hardware. This bit can be cleared by writing a '0'. Writing a '1' to this bit has no effect.	0: No failure. 1: Failure occurs.					
RFU	7:2	Reserved.	N/A					
DONEIE	8	Flash Done Interrupt Enable.	0: Disabled. 1: Enabled.					
AFIE	9	Flash Access Fail Interrupt Enable.	0: Disabled. 1: Enabled.					
RFU	31:10	Reserved.	N/A					



15.2.5 Flash Data Register 0 (FLSH_DATA0, Offset 0x0030)

Table 15-6. FLSH_DATA0 (Offset 0x0030)

Position	31	30	29	28	27	26	25	24
Field	DATA0[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	DATA0[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	DATA0[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	DATA0[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description					Settings	
DATA0	31:0	Flash Write Data.						



15.2.6 Flash Data Register 1 (FLSH_DATA1, Offset 0x0034)

Table 15-7. FLSH_DATA1 (Offset 0x0034)

Position	31	30	29	28	27	26	25	24
Field	DATA1[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	DATA1[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	DATA1[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	DATA1[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description					Settings	
DATA1	31:0	Flash Write Data.						



15.2.7 Flash Data Register 2 (FLSH_DATA2, Offset 0x0038)

Table 15-8. FLSH_DATA2 (Offset 0x0038)

Position	31	30	29	28	27	26	25	24
Field	DATA2[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	DATA2[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	DATA2[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	DATA2[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description						Settings
DATA2	31:0	Flash Write Data.						



15.2.8 Flash Data Register 3 (FLSH_DATA3, Offset 0x003C)

Table 15-9. FLSH_DATA3 (Offset 0x003C)

Position	31	30	29	28	27	26	25	24
Field	DATA3[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	DATA3[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	DATA3[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	DATA3[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description						Settings
DATA3	31:0	Flash Write Data.						



15.2.1 Access Control Register (ACNTL, Offset 0x0040)

Table 15-10. ACNTL (Offset 0x0040)

Position	31	30	29	28	27	26	25	24
Field	ADATA[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	ADATA[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	ADATA[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	ADATA[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
ADATA	31:0	System Info Block Access Data	<p>Writing the ACNTL register with the following values in the order shown, allows read and write access to the system and user Information block.: <code>pflc->acntl = 0x3a7f5ca3;</code> <code>pflc->acntl = 0xa1e34f20;</code> <code>pflc->acntl = 0x9608b2c1;</code></p> <p>When unlocked, a write of any word will disable access to system and user information block.</p> <p>Readback of this register is always zero.</p>



16.0 Inter-Integrated Circuit (I²C)

The I²C host port is compliant with the I²C Bus Specification with features:

- I²C bus specification version 2.1 compliant (100 kHz and 400 kHz)
- Programmable for both normal (100 kHz) and fast bus data rates (400 kHz)
- Clock synchronization and bus arbitration
- Supports arbitration in a multi-master environment
- 8-byte depth 2 FIFOs (RX and TX)
- Supports I²C bus hold for slow host service
- Transfer status interrupts and flags
- Support DMA data transfer (not applicable on Revision A silicon)

16.1 Description

The I²C port is a half-duplex serial port that uses two lines (SDA for data and SCL for clock) for data transmission. Standard 100 kHz and fast 400 kHz transmit modes are supported.

16.2 Protocol

The I²C protocol communication is a 2-wire serial transmission. It is a half-duplex protocol where data can be transferred at various baud rates: up to 100 Kbits in standard mode and up to 400 Kbits in fast mode. Each transfer is made of a start bit, followed by eight data bits, an acknowledge bit sent by the receiver, and a stop bit. Data is sent with the MSB first.

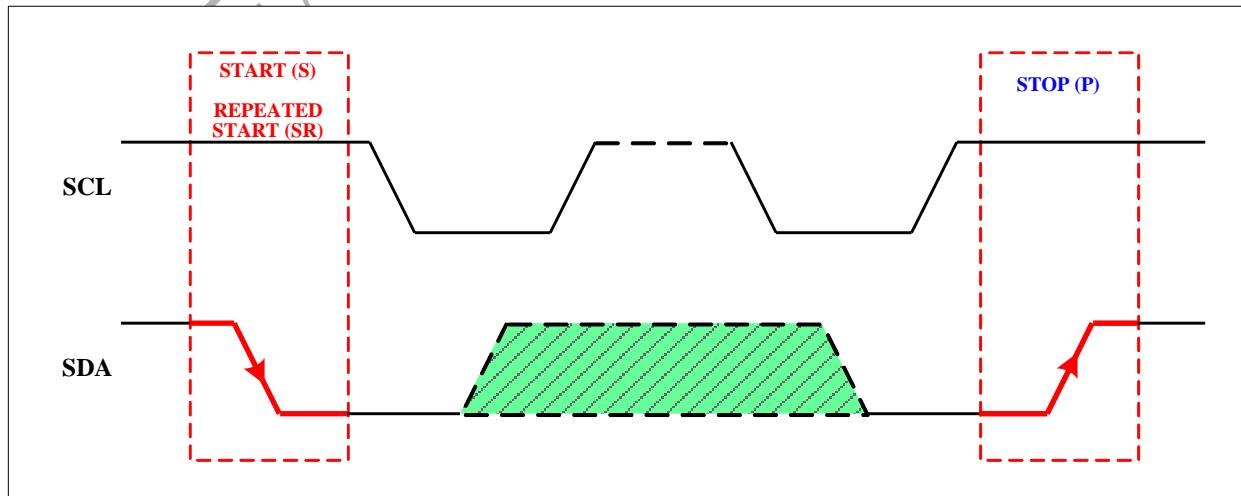
16.2.1 Bit Transfer

Both SDA and SCL lines are bi-directional lines connected to a positive supply voltage via a current source or a pull-up resistor. When the bus is free, the lines are in high state. The data on the SDA line must be stable when the SCL line is high.

Communication starts when the SDA line switches from high to low state and the SCL line is high. Communication stops when the SDA line switches from low to high state and the SCL line is high. Only the master generates the start and stop conditions. During communication, the bus is considered busy.

The figure below shows the signals during communication.

Figure 16-1. I²C Start and Stop Conditions





16.2.2 Addressing

This block can only run in master mode, and in fast or standard mode. Hence, the first byte sent defines the type of addressing (see Table 16-1. Definition of the First Byte):

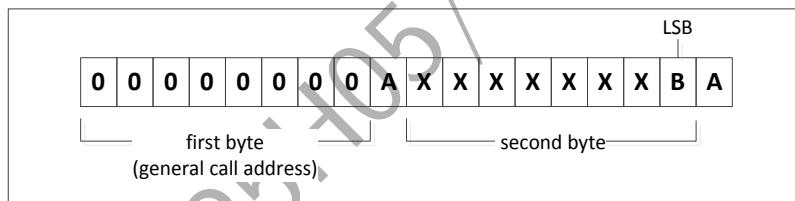
Table 16-1. Definition of the First Byte

7-1 bits	R/W bit	Description
0000 000	0	General Call Address
0000 000	1	START byte
0000 001	X	Reserved (CBUS address)
0000 010	X	Reserved
0000 011	X	Reserved
0000 1XX	X	Hs-mode master code
1111 1XX	X	Reserved

16.2.3 General Call Address

The General Call Address is 0x00. The acknowledge bit is optional if the device does not require the General Call Address data. A General Call Address is followed by another byte, which initiates all the devices. The format is given in General Call Address, Bytes Format.

Figure 16-2. General Call Address, Bytes Format



When bit B is 0, the second byte has the following definition:

- 0x06: reset and write the programmable part of the slave address by hardware.
- 0x04: write the programmable part of the slave address by hardware.

When bit B is set to 1, the hardware address of the master is written in the second byte.

The General Call Address has no effect on this interface.

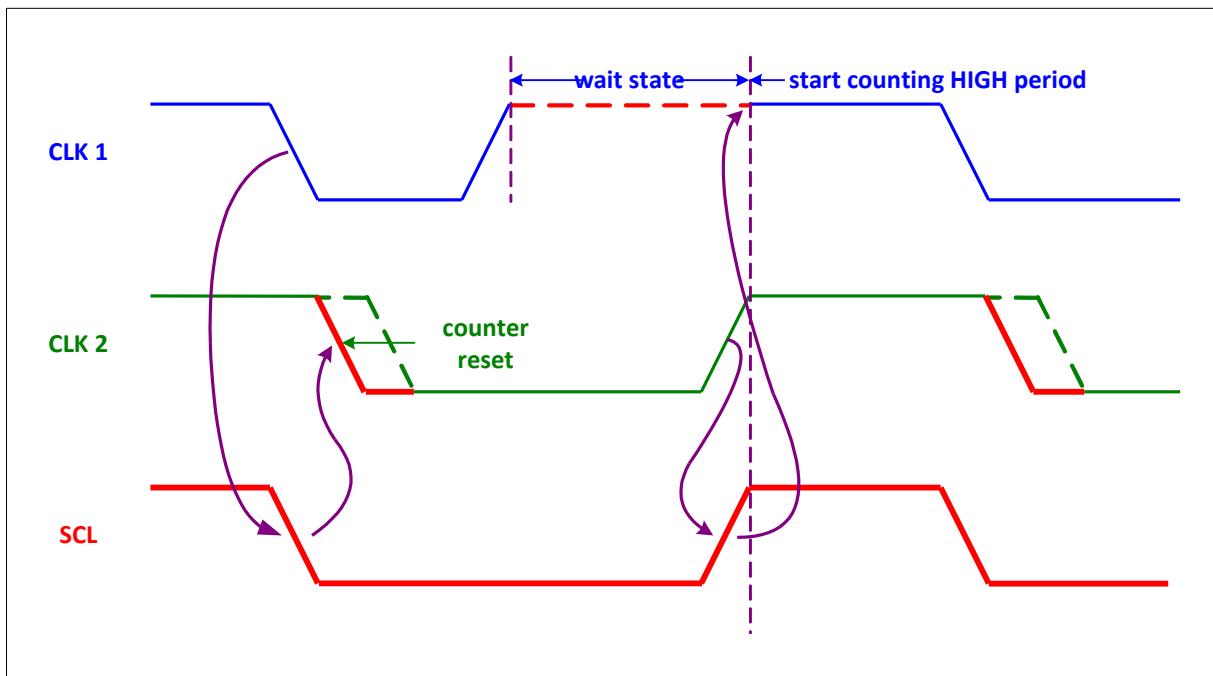
16.2.4 START Byte

The START byte is used to initiate communication with slow devices. The START byte has no effect on this interface.

If two or more slaves are addressed at the same time, only the slave that has the smallest address (described in decimal figures), can exchange data with the master. Communication with the other slaves is cut off.

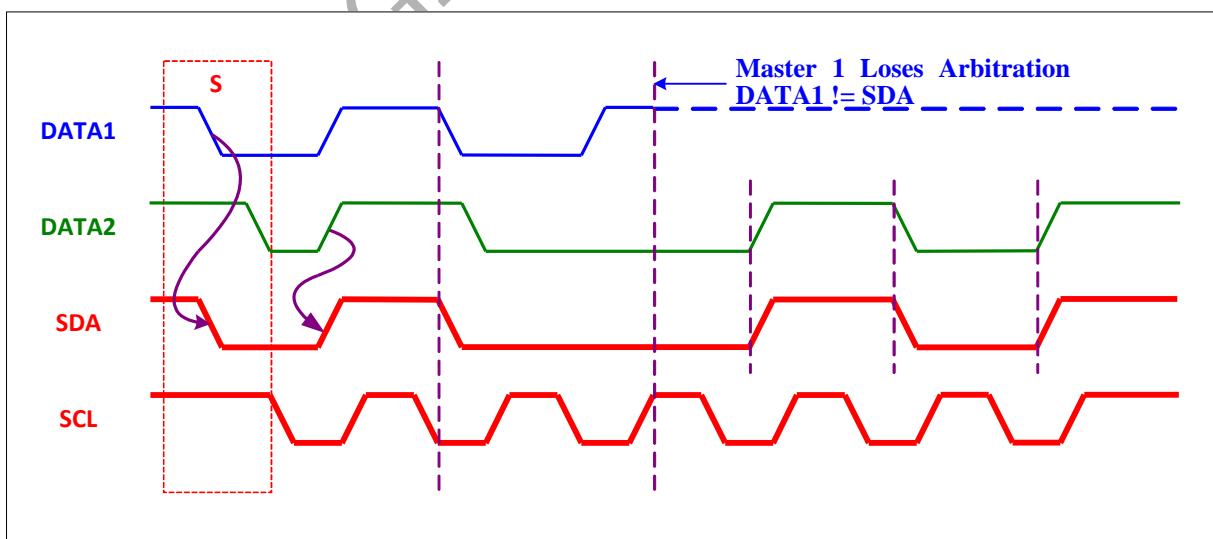
16.2.5 Clock Synchronization

The I²C protocol accepts multiple masters on the bus. This is the reason why synchronization between the masters' clocks is compulsory. Clock synchronization is performed using the wired-AND connection of SCL. Figure 16-3. Clock Synchronization below shows an example of clock synchronization:

**Figure 16-3. Clock Synchronization**

16.2.6 Bus Arbitration

When the bus is free, two or more masters can initiate communication, but only one master can make a valid transmission, and the others must switch to slave mode. Each master compares the data sent to the SDA line. If the data is different, the master which output the HIGH state on the SDA line will switch off its SDA data output. Arbitration occurs until only one master remains. Figure 16-4 shows a master arbitration:

Figure 16-4. Master Arbitration

16.3 FIFO

Both RX and TX FIFOs are 8-byte depth.

The RX FIFO is written by hardware when a data has been received from the device. If the RX FIFO is full all data received from the slave are lost, so do not start a new read operation with the RX FIFO full.



To read the RX FIFO, you need to read the data register.

The TX FIFO is written by software using the data register (a write in the data register writes into the TX FIFO).

The TX FIFO is read by hardware only when the acknowledge bit has been received (meaning just before the stop/restart bit in case of NACK or just before the first bit of the next data in case of ACK).

16.4 Interrupt

The I²C Controller contains multiple interrupt sources that are combined into one interrupt request signal to the interrupt controller. If the I²C Controller is enabled, the source of the interrupt is determined by which bits are set in the I2C_ISR register.

It is recommended to acknowledge an interrupt before enabling it (unmask operation) to avoid a previous event triggering.

16.5 DMA

The I²C supports the DMA peripheral communication. It allows the DMA to read and/or write the FIFOs of the I²C device.

To enable the DMA transfers, refer to the DMA design specification. If you choose to enable the DMA, there is no special configuration for the I²C device: the DMA should be seen as a core (executing software) replacement. The I²C interrupts are not rerouted to the DMA and are still routed to the core (it is up to the software to configure the interrupts according to the DMA usage).

Note that I²C DMA support does not exist on revision A of silicon.

16.6 I²C Registers

Address assignments for the I²C registers are outlined in Table 16-2

Table 16-2. I²C Register Address (Base ADDR = 0x4000_C000)

Offset	Access	Register	Description
0x0000	RW	I2C_CR	Control Register
0x0004	RW	I2C_SR	Status Register
0x0008	RW	I2C_IER	Interrupt Enable Register
0x000C	R	I2C_ISR	Interrupt Status Register
0x0010	RW	I2C_BRR	Baudrate Register
0x0014	RW	I2C_RCR	RX Count Register
0x0018	RW	I2C_TXR	TX FIFO Output Register
0x001C	RW	I2C_DR	Data Register
0x0020	R	Reserved	Reserved
0x0024	R	Reserved	Reserved
0x0028	R	Reserved	Reserved
0x0030*	RW	I2C_DMR*	DMA Register*

* Not applicable on revision A silicon

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



16.6.1 I²C Control Register (I²C_CR, Offset 0x0000)

Table 16-3. I²C CR (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field				RFU[7:0]				
Reset				0000 0000				
Access				R				
Position	23	22	21	20	19	18	17	16
Field				RFU[7:0]				
Reset				0000 0000				
Access				R				
Position	15	14	13	12	11	10	9	8
Field				RFU[3:0]	FREEZE	AFREAD	RXFLUSH	TXFLUSH
Reset				0000	-	0	0	0
Access				R	W	W	W	R/W
Position	7	6	5	4	3	2	1	0
Field	STOP	RESTART	START	AUTOSTT			RXTHD[3:0]	
Reset	0	0	0	0			0000	
Access	RW	W	W	RW			RW	

Name	Bits	Description	Settings
RXTHD	3:0	Receive FIFO Threshold. This field specifies the depth of receive FIFO for interrupt generation.	0000: Ignored 0001: 1 word 0010: 2 words : 1000: 8 words 1001~1111: Ignored
AUTOSTT	4	Auto Start. If this bit is 1, I ² C communication starts automatically when a word is placed in the transmit FIFO.	0: Manual start 1: Automatic start
START	5	Start communication. When 1 is written, I ² C communication is started. This bit is automatically cleared by hardware when the first byte is sent.	0: No effect 1: Start communication
RESTART	6	Restart. When 1 is written, the communication restarts if the slave does not acknowledge the first byte (the address byte).	0: No effect 1: Restart communication
STOP	7	Stop Read communication. No effect on write communication.	0: No effect 1: Stop read
TXFLUSH	8	Flush Transmit FIFO. This bit is cleared by hardware.	0: No effect 1: Flush TX FIFO
RXFLUSH	9	Flush Receive FIFO. This bit is cleared by hardware.	0: No effect 1: Flush RX FIFO
AFREAD	10	Auto Read. If set to 1, this bit indicates that the current transfer should not be stopped but restarted when the transmit FIFO becomes empty (generally used to perform a read after a write transfer). The FREEZE bit should be set to prevent any error on the read operation. Use the I ² C_ISR.RDYRDIS bit to know when you can write the slave address (in read mode) in the transmit FIFO.	0: No effect 1: Auto Start
FREEZE	11	Freeze. If set to 1, the I ² C bus is frozen if the transmit FIFO is empty during a write or if the receive FIFO is full during a read.	0: No effect 1: Freeze I ² C bus
RFU	31:12	Reserved	N/A



16.6.2 I²C Status Register (I²C_SR, Offset 0x0004)

Table 16-4. I²C_SR (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	TXELT[3:0]				RXELT[3:0]			
Reset	0000				0000			
Access	R				R			
Position	7	6	5	4	3	2	1	0
Field	TXFULL	TXEMPTY	RXFULL	RXEMPTY	RFU	RFU	FROZEN	BUSY
Reset	0	1	0	1	0	0	0	0
Access	R	R	R	R	R	R	R	R
Name	Bits	Description					Settings	
BUSY	0	Busy. The I ² C Bus is busy when another master is using the bus or this interface itself is using the bus.					0: The bus is idle. 1: The bus is busy.	
FROZEN	1	Bus Frozen. This occurs when the clock signal is stuck at 0, such as being held by another device.					0: The bus is not frozen. 1: The bus is frozen.	
RFU	3:2	Reserved					N/A	
RXEMPTY	4	Receive FIFO Empty.					0: Receive FIFO is not empty. 1: Receive FIFO is empty.	
RXFULL	5	Receive FIFO Full.					0: Receive FIFO is not full. 1: Receive FIFO is full	
TXEMPTY	6	Transmit FIFO Empty.					0: Transmit FIFO is not empty 1: Transmit FIFO is empty	
TXFULL	7	Transmit FIFO Full.					0: Transmit FIFO is not full 1: Transmit FIFO is full	
RXELET	11:8	Number of byte in the RX FIFO.						
TXELT	15:12	Number of byte in the TX FIFO.						
RFU	31:16	Reserved					N/A	



16.6.3 I²C Interrupt Enable Register (I2C_IER, Offset 0x0008)

Table 16-5. I²C_IER (Offset 0x0008)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU	FFTXHIE	FFTXOIE	FFRXIE	RDYRDIE	COMENDIE	NOANSIE	LOSTIE
Reset	0	0	0	0	0	0	0	0
Access	R	RW	RW	RW	RW	RW	RW	RW
Name	Bits	Description					Settings	
LOSTIE	0	Loss of Transfer Interrupt Enable.					0: Disabled 1: Enabled	
NOANSIE	1	Address NACK (not acknowledge) Interrupt Enable.					0: Disabled 1: Enabled	
COMENDIE	2	Communication End Interrupt Enable.					0: Disabled 1: Enabled	
RDYRDIE	3	Read Ready Interrupt Enable					0: Disabled 1: Enabled	
FFRXIE	4	Receive FIFO Threshold Enable.					0: Disabled 1: Enabled	
FFTXOIE	5	Transmit FIFO Last Word Interrupt Enable.					0: Disabled 1: Enabled	
FFTXHIE	6	Transmit FIFO Last Four Words Interrupt Enable.					0: Disabled 1: Enabled	
RFU	31:7	Reserved					N/A	



16.6.4 I²C Interrupt Status Register (I²C_ISR, Offset 0x000C)

Table 16-6. I²C ISR (Offset 0x000C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU	FFTXHIS	FFTXOIS	FFRXIS	RDYRDIS	COMENDIS	NOANSIS	LOSTIS
Reset	0	0	0	0	0	0	0	0
Access	R	RC	RC	RC	RC	RC	RC	RC

Name	Bits	Description	Settings
LOSTIS	0	Loss of Transfer Interrupt Status. This occurs when the current transfer is lost due to another master.	0: No interrupt is pending 1: Event occurred.
NOANSIS	1	Address NACK (not acknowledge) Interrupt Status. This occurs when no slave responds to the address.	0: No interrupt is pending 1: Event occurred.
COMENDIS	2	Communication End Interrupt Status. This occurs when the current transfer is over.	0: No interrupt is pending 1: Event occurred.
RDYRDIS	3	Read Ready Interrupt Status. This occurs when the current write transfer is done and the next read is ready to be processed (only if AFREAD=1).	0: No interrupt is pending 1: Event occurred.
FFRXIS	4	Receive FIFO Threshold Status. This occurs when FIFO RX reaches the programmed threshold.	0: No interrupt is pending 1: Event occurred.
FFTXOIS	5	Transmit FIFO Last Word Interrupt Status. This occurs when there is only one byte remaining in the TX FIFO.	0: No interrupt is pending 1: Event occurred.
FFTXHIS	6	Transmit FIFO Last Four Words Interrupt Status. This occurs when there are four bytes remaining in the TX FIFO.	0: No interrupt is pending 1: Event occurred.
RFU	31:7	Reserved	N/A

Note: These bits are set by hardware and must be cleared by software to be acknowledged.



16.6.5 I²C Baudrate Register (I²C_BRR, Offset 0x0010)

Table 16-7. I²C BRR (Offset 0x0010)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU	DIV[6:0]						
Reset	0	000 0000						
Access	R	RW						

Name	Bits	Description	Settings
DIV	6:0	Baud Rate Divisor. DIV = "Nominal Frequency"/(32 × Desired Frequency) Where the nominal frequency = 2x f _{PLL0}	
RFU	31:7	Reserved	N/A



16.6.6 I²C RX Count Register (I²C_RCR, Offset 0x0014)

Table 16-8. I²C RCR (Offset 0x0014)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RXCNT[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	RXCNT[7:0]							
Reset	000 0000							
Access	RW							

Name	Bits	Description	Settings
RXCNT	15:0	RX Count. Number of byte to read. This register is used to specify how many bytes should be read from the device before stopping the communication.	
RFU	31:16	Reserved	N/A



16.6.7 I²C TX FIFO Output Register (I²C_TXR, Offset 0x0018)

Table 16-9. I²C TXR (Offset 0x0018)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	DATA[7:0]							
Reset	000 0000							
Access	RW							

Name	Bits	Description	Settings
DATA	7:0	TX FIFO Output. This register is used to read the output of the FIFO, reading this register does not read the FIFO itself. This register is mainly used for debug purpose.	
RFU	31:8	Reserved	N/A



16.6.8 I²C Data Register (I²C_DR, Offset 0x001C)

Table 16-10. I²C DR (Offset 0x001C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	DATA[7:0]							
Reset	000 0000							
Access	RW							

Name	Bits	Description	Settings
DATA	7:0	Data register. This register is used to obtain data from or place data into the FIFO. Read this register to retrieve the oldest data available in the RX FIFO. Write this register to place data in the TX FIFO.	
RFU	31:8	Reserved	N/A



16.6.9 I²C DMA Register (I2C_DMR, Offset 0x0030)*

Table 16-11. I²C DMR (Offset 0x0030)*

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[5:0]						RXEN	RXCNT3
Reset	00 0000						0	0
Access	R						RW	RW
Position	7	6	5	4	3	2	1	0
Field	RXCNT[2:0]			TXEN	TXCNT[3:0]			
Reset	000			0	0000			
Access	RW			RW	RW			
Name	Bits	Description					Settings	
TXCNT	3:0	TX threshold						
TXEN	4	TX channel enable					0: Disable 1: Enable	
RXCNT	8:5	RX threshold						
RXEN	9	RX channel enable					0: Disable 1: Enable	
RFU	31:10	Reserved					N/A	

* Not applicable on revision A silicon.



17.0 Magnetic Stripe Reader

17.1 Magnetic Stripe Reader Features

Magnetic Stripe Reader (MSR) has the following features:

- Support for 3 simultaneous tracks
- Low power DSP
- Direct connection to magnetic heads
- Automatic peak detection
- FIFO and DMA Support

17.2 General Description and Memory Requirement

Each track reader has two differential input pins. Extended ESD protection is only necessary for one of the stripe reader inputs for each channel. The other input will be connected to GND.

The ADCs are designed to operate at a maximum clock rate of 2 MHz. When used by the CPU, the ADC clock divider (DSPCTRL.DIV_RATIO) must be set.

The DSP has three 8×16-bit FIFOs, one for each track. The FIFOs must be continuously read by the CPU or by DMA.

The DSP is a APB slave accessible via the Cortex-M3 or DMA. The DSP core is held in reset until DSPCTRL.DSPEN=1. This reduces system power consumption. The DSP peripheral registers are available to the CPU while the DSP core is held in reset, allowing software to configure the DSP.

The ADC provides data for each track to the DSP. The DSP processes this data, and pushes the data on to the appropriate track FIFO data output register. The DSP outputs to the FIFO one 16-bit value per each detected waveform peak. This value contains a 7-bit peak amplitude at bit position [15:9] and a 9-bit peak-to-peak time interval at bit position [8:0].

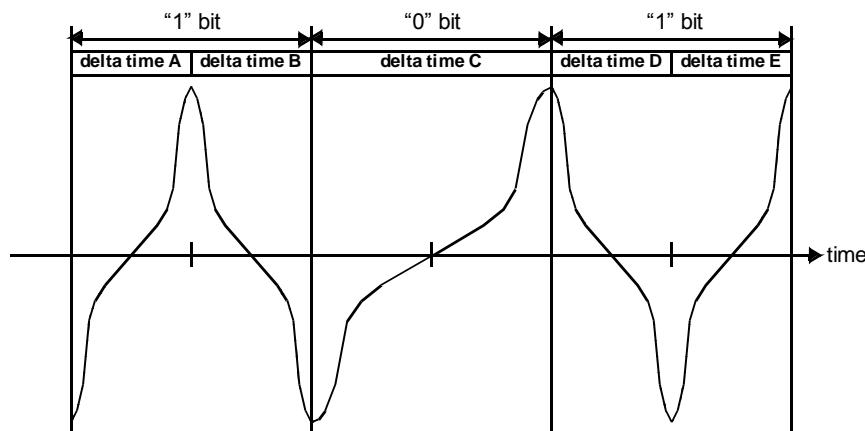
Software must allocate at least 3072 bytes of data memory for each track.

17.3 Magnetic Stripe Reading Overview

Figure 17-1 shows an example of magnetic card bit encoding. This shows the signal derived from a single magnetic card track. The signal contains a series of alternating peaks occurring once or twice within a single bit time. A logic '0' is encoded as one transition per bit; a logic '1' is encoded as two transitions. Although the bit rate will vary and distortions will occur for various reasons, the bit rate is relatively constant from bit-to-bit and the delta time between alternating peaks can be used to decode the series of '1's and '0's.

17.4 MSR Dynamic Filtering

DSP dynamically determines when speed switch is required, and dynamically switches processing to the optimal waveform. This happens on the fly when the swipe speed changes.

Figure 17-1. Magnetic Stripe Bit Encoding

17.5 DMA and Interrupt Operations

17.5.1 A3 Silicon Revision and Prior

Each track has its own DMA request and its own interrupt flag. The DMA request and/or FIFO interrupt flag for a given track will fire when that track's FIFO is half full. The DSP FIFO Interrupt Enable must be set in order for the FIFO interrupt flag to cause an interrupt.

Upon completion of a swipe, the DSP Interrupt Flag is set. This interrupt is enabled when the DSP is enabled (DSPCTRL.DSPEN).

Once the swipe is complete, the DSP also outputs the number of stored peaks (values) for each track via the T1_DATA_CNT/T2_DATA_CNT/T3_DATA_CNT registers. The number of peaks and the data stored in memory must be further processed by software to decode the characters recorded on the magnetic stripe.

17.5.2 B1 Silicon Revision and Beyond

Each track has its own DMA request and its own interrupt flag. The DMA request and/or FIFO interrupt flag for a given track will fire when that track's FIFO is half full. The DSP FIFO Interrupt Enable must be set in order for the FIFO interrupt flag to cause an interrupt.

The track FIFO interrupt flags (T1_INTF, T2_INTF, and T3_INTF) are reflected in both the DSPCTRL and DSPINT registers.

Clearing the FIFO interrupt flag in either DSPCTRL or DSPINT automatically updates the corresponding FIFO interrupt flag in the other register. Writing a 1 to the DSPINT FIFO interrupt flag will prevent other interrupt flags from being accidentally cleared.

The track FIFO interrupts indicating that the FIFO is half full may be globally or individually enabled. In order to enable the FIFO interrupt for a single track, the DSPINTEN.T1F_INTEN/T2F_INTEN/T3F_INTEN bits may be individually set.

Setting DSPCTRL.FIFO_INT_EN will enable interrupts for all 3 tracks, regardless of the setting of the individual track enables (DSPINTEN.T1F_INTEN/T2F_INTEN/T3F_INTEN).

The FIFO empty status flag (DSPSTAT.T1EMPTYF/T2EMPTYF/T3EMPTYF) indicates whether the respective track's FIFO is empty.

Upon completion of a swipe and when all FIFOs are empty, the DSP Interrupt Flag is set. The DSP Interrupt Flag is reflected in both the DSPCTRL and the DSPINT registers.



Clearing the interrupt flag in either DSPCTRL or DSPINT automatically updates the corresponding interrupt flag in the other register. Writing a 1 to the DSPINT.DSP_INTF interrupt flag will prevent other interrupt flags from being accidentally cleared. This interrupt is enabled by default when the DSP is enabled (DSPCTRL.DSPEN) but may be disabled by the user (DSPINTEN.DSP_INTFEN) if needed.

Upon completion of a swipe, the DSP Swipe Done flag (DSPINT.SWDONE) is set. Writing a 1 to the DSPINT.SWDONE interrupt flag will prevent other interrupt flags from being accidentally cleared. This interrupt is enabled by DSPINTEN.SWDONEEN.

An overflow condition occurs when the track's FIFO is full, and the DSP pushes an additional value into that FIFO. An overflow condition indicates that there is data lost for that track, and the user application will need to handle the loss of data with respect to further processing. In this case, the corresponding track overflow interrupt flag is set (DSPINT.T1OVF/T2OVF/T3OVF).

If the overflow interrupt enable is set for that track (DSPINTEN.T1OVFEN/T2OVFEN/T3OVFEN), the MSR DSP also generates an interrupt.

Once the swipe is complete, the DSP also outputs the number of stored peaks (values) for each track via the T1_DATA_CNT/T2_DATA_CNT/T3_DATA_CNT registers. The number of peaks and the data stored in memory must be further processed by software to decode the characters recorded on the magnetic stripe.

17.6 Modes of Operations

This device has two methods to determine the zero threshold of the magnetic stripe waveform. The method used is set by DSPCTRL.DSPPKDETECT.

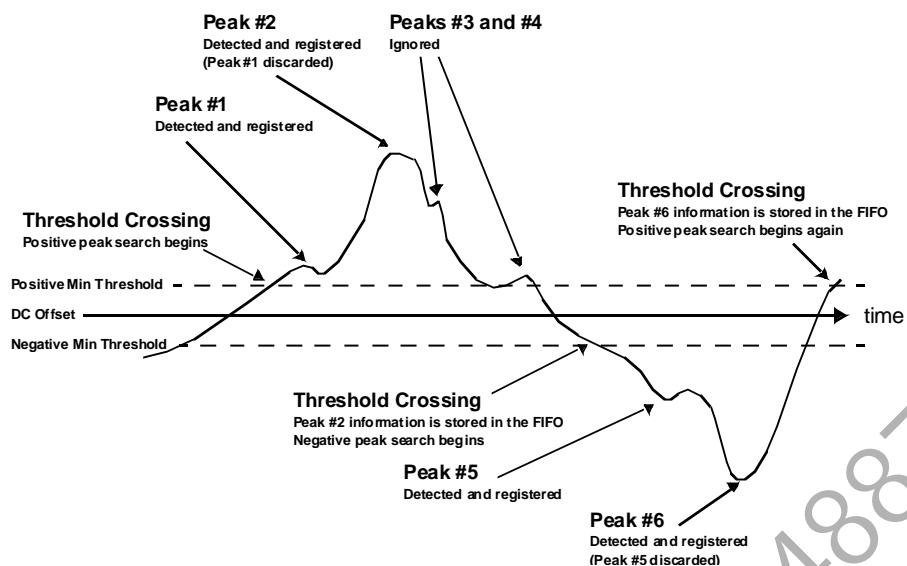
Once the MSR DSP is enabled, the DSP takes a running average of the MSR ADC output data to calculate the zero threshold of the output data.

When the device is configured for Variable Threshold Mode (DSPCTRL.DSP_PKDETECT = 0), the device will continuously monitor and update the zero threshold during the swipe.

When the device is configured for Fix Threshold Mode (DSPCTRL.DSP_PKDETECT = 1), the device will use only the ADC output data prior to swipe to determine the zero threshold. This fixed value will be used once the swipe commences for the entire duration of the swipe.

An example of peak detection is shown in Figure 17-2. The peak detection algorithm of the DSP works as follow:

- 1) The user programs the track zero cross-threshold registers. This sets the positive and negative minimum deltas around the zero threshold determined by the device.
- 2) Once the minimum positive threshold is crossed, the peak detector begins searching for a positive peak. This first value over the minimum is registered.
- 3) Every incoming sample is compared to the registered value.
- 4) If the incoming sample is greater, it is registered and the old value is discarded. Otherwise the incoming sample is ignored.
- 5) If the incoming sample falls below the negative threshold, the peak detector stores the positive peak information in the FIFO, and begins the search for a negative peak.
- 6) The negative peak search is analogous to the positive peak search.
- 7) The peak detector alternates between the negative and positive peak searches until the swipe is finished.

**Figure 17-2. MSR Peak Detection Example**



17.7 Magnetic Stripe Reader Registers

Due to the number of DSP registers, an indexed access method with address and data registers is used to interface with the DSP. For example, in order to access the DSP register T2_ZCT_MID, the value 0x06 has to be written to DSPADDR first, and DSPDATA then can be read or written and will return T2_ZCT_MID, or change it, respectively.

Address assignments for the Magnetic Card Reader Registers are outlined in Table 17-1.

Table 17-1. Magnetic Stripe Reader Register Address (Base ADDR = 0x4002_B000)

Offset	Access	Register	Description
0x0000	RW	DSPADDR	MSR DSP Address Register
0x0004	RW	DSPCTRL	MSR DSP Control Register
0x0008	RW	DSPDATA	MSR DSP Data Register
0x000C	RW	T1FIFO	Track 1 FIFO Data Output Register
0x0010	RW	T2FIFO	Track 2 FIFO Data Output Register
0x0014	RW	T3FIFO	Track 3 FIFO Data Output Register
0x0018*	RW	DSPINT	MSR DSP Interrupt Flag Register
0x001C*	RW	DSPINTEN	MSR DSP Interrupt Register
0x0020*	RW	DSPSTAT	MSR DSP Status Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.

* From Silicon Revision B1. Does not exist on Revision A3 and prior.



17.7.1 MSR DSP Address Register (DSPADDR, Offset 0x0000)

Table 17-2. DSPADDR (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	ADDR[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
ADDR	7:0	MSR Indirect Address Register Access. Indirectly access MSR settings registers by writing the index to this register. The register values are accessible by reading the DSPDATA Register as shown in 17.7.16. The registers names, indices and definitions are described in 17.7.2 through 17.7.14.	
RFU	31:8	Reserved	N/A



17.7.2 Track 1 & 3 Scale Factor Register (T13_SCALE, Index 0x00)

Table 17-3. T13_SCALE (Index 0x00)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[5:0]						T13_SCALE[1:0]	
Reset	0000 00						00	
Access	R						RW	

Name	Bits	Description	Settings
T13_SCALE	1:0	Track 1 and 3 scale factor. Valid values are 00, 01 and 10. Writing 3 sets the register to 2.	
RFU	31:2	Reserved	N/A



17.7.3 Track 2 Scale Factor Register (T2_SCALE, Index 0x01)

Table 17-4. T2_SCALE (Index 0x01)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[5:0]						T2_SCALE[1:0]	
Reset	0000 00						00	
Access	R						RW	

Name	Bits	Description	Settings
T2_SCALE	1:0	Track 2 scale factor. Valid values are 00, 01 and 10. Writing 3 sets the register to 2.	
RFU	31:2	Reserved	N/A



17.7.4 Track 1 & 3 Zero Cross Threshold Fast Register (T13_ZCT_FAST, Index 0x02)

Table 17-5. T13_ZCT_FAST (Index 0x02)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[4:0]					ZCTF[10:8]		
Reset	0 0000					000		
Access	R					RW		
Position	7	6	5	4	3	2	1	0
Field	ZCTF[7:0]							
Reset	1100 1000							
Access	RW							
Name	Bits	Description					Settings	
ZCTF	10:0	Zero Cross Threshold Fast for Tracks 1 and 3						
RFU	31:11	Reserved					N/A	



17.7.5 Track 1 & 3 Zero Cross Threshold Middle Register (T13_ZCT_MID, Index 0x03)

Table 17-6. T13_ZCT_MID (Index 0x03)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[4:0]							
Reset	0 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	ZCTM[7:0]							
Reset	0111 0100							
Access	RW							
Name	Bits	Description						Settings
ZCTM	10:0	Zero Cross Threshold Middle for Tracks 1 and 3						
RFU	31:11	Reserved						N/A



17.7.6 Track 1 & 3 Zero Cross Threshold Slow Register (T13_ZCT_SLOW, Index 0x04)

Table 17-7. T13_ZCT_SLOW (Index 0x04)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[4:0]							
Reset	0 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	ZCTS[7:0]							
Reset	0011 1100							
Access	RW							
Name	Bits	Description						Settings
ZCTS	10:0	Zero Cross Threshold Slow for Tracks 1 and 3						
RFU	31:11	Reserved						N/A



17.7.7 Track 2 Zero Cross Threshold Fast Register (T2_ZCT_FAST, Index 0x05)

Table 17-8. T2_ZCT_FAST (Index 0x05)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[4:0]					ZCTF[10:8]		
Reset	0 0000					000		
Access	R					RW		
Position	7	6	5	4	3	2	1	0
Field	ZCTF[7:0]							
Reset	1000 0100							
Access	RW							
Name	Bits	Description					Settings	
ZCTF	10:0	Zero Cross Threshold Fast for Track 2						
RFU	31:11	Reserved					N/A	



17.7.8 Track 2 Zero Cross Threshold Middle Register (T2_ZCT_MID, Index 0x06)

Table 17-9. T2_ZCT_MID (Index 0x06)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[4:0]					ZCTM[10:8]		
Reset	0 0000					000		
Access	R					RW		
Position	7	6	5	4	3	2	1	0
Field	ZCTM[7:0]							
Reset	0010 1101							
Access	RW							
Name	Bits	Description					Settings	
ZCTM	10:0	Zero Cross Threshold Middle for Track 2						
RFU	31:11	Reserved					N/A	



17.7.9 Track 2 Zero Cross Threshold Slow Register (T2_ZCT_SLOW, Index 0x07)

Table 17-10. T2_ZCT_SLOW (Index 0x07)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[4:0]					ZCTS[10:8]		
Reset	0000 0					000		
Access	R					RW		
Position	7	6	5	4	3	2	1	0
Field	ZCTS[7:0]							
Reset	0010 0011							
Access	RW							
Name	Bits	Description					Settings	
ZCTS	10:0	Zero Cross Threshold Slow for Track 2						
RFU	31:11	Reserved					N/A	



17.7.10 Leading 0 Detection Count (START_PCNT, Index 0x08)

Table 17-11. START PCNT (Index 0x08)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[3:0]				START_PCNT[3:0]			
Reset	0000				1000			
Access	R				RW			
Name	Bits	Description						Settings
START_PCNT	3:0	Leading 0 Detection Count						
RFU	31:4	Reserved						N/A



17.7.11 Track 1 Data Count (T1_DATA_CNT, Index 0x09)

Table 17-12. T1_DATA_CNT (Index 0x09)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[4:0]					COUNT[10:8]		
Reset	0000 0					000		
Access	R					R		
Position	7	6	5	4	3	2	1	0
Field	COUNT[7:0]							
Reset	0000 0000							
Access	R							
Name	Bits	Description					Settings	
COUNT	10:0	Track 1 Data Count						
RFU	31:11	Reserved					N/A	



17.7.12 Track 2 Data Count (T2_DATA_CNT, Index 0xA)

Table 17-13. T2_DATA_CNT (Index 0xA)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[4:0]					COUNT[10:8]		
Reset	0000 0					000		
Access	R					R		
Position	7	6	5	4	3	2	1	0
Field	COUNT[7:0]							
Reset	0000 0000							
Access	R							
Name	Bits	Description					Settings	
COUNT	10:0	Track 2 Data Count						
RFU	31:11	Reserved					N/A	



17.7.13 Track 3 Data Count (T3_DATA_CNT, Index 0x0B)

Table 17-14. T3_DATA_CNT (Index 0x0B)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[4:0]					COUNT[10:8]		
Reset	0000 0					000		
Access	R					R		
Position	7	6	5	4	3	2	1	0
Field	COUNT[7:0]							
Reset	0000 0000							
Access	R							
Name	Bits	Description					Settings	
COUNT	10:0	Track 3 Data Count						
RFU	31:11	Reserved					N/A	



17.7.14 ADC Configuration Register (ADCCFG1, Index 0x12)

Table 17-15. ADCCFG1 (Index 0x12)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[2:0]			RCR2[2:0]			RCR1[1:0]	
Reset	000			110			00	
Access	R			RW			RW	
Position	7	6	5	4	3	2	1	0
Field	ADCOUTEN	OFSAZ[2:0]			VREF_SEL[1:0]		RFU	PUADC
Reset	0	000			11		0	0
Access	RW	RW			RW		R	RW

Name	Bits	Description	Settings
PUADC	0	Power Up ADC. This bit control enable the ADC bandgap, and the 3 dedicated magnetic stripes ADCs.	0: Disable MSR 1: Enable MSR
RFU	1	Reserved	N/A
VREF_SEL[1:0]	3:2	Voltage Reference Selection	00: 1.0V 01: 0.8V 10: 0.6V 11: 0.5V
OFSAZ[2:0]	6:4	Offset Auto Zero for ADC 2/1/0. Setting one of these bits to 1 will ground the corresponding ADC input. Clearing these bits will allow normal operation.	000: Normal Operation xx1: Ground ADC0 input x1x: Ground ADC1 input 1xx: Ground ADC2 input
RFU	7	Reserved	N/A
RCR1[1:0]	9:8	RC filter R1 setting	00: Short 01: 0.5KΩ 01: 1.0KΩ 01: 2.0KΩ
RCR2[2:0]	12:10	RC filter R2 setting	0xx: Open 100: 1.5KΩ 101: 3.0KΩ 110: 6.0KΩ 111: 12KΩ
RFU	31:13	Reserved	N/A

17.7.15 DSP Control Register (DSPCTRL, Offset 0x0004)

Table 17-16. DSPCTRL (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU	DVI_RATIO[6:0]						
Reset	0	001 1010						
Access	R	RW						
Position	7	6	5	4	3	2	1	0
Field	T3_INTF	T2_INTF	T1_INTF	DSP_INTF	RFU	FIFO_INT	DSP_PKDETECT	DSPEN
Reset	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	R	RW	RW	RW

Name	Bits	Description	Settings
DSPEN	0	DSP Enable. Setting this bit to 1 enables the DSP Functionality. Clearing this bit to 0 disables the DSP.	0: Disable DSP 1: Enable DSP
DSP_PKDETECT	1	DSP Peak Detect Threshold Mode.	0: Select Variable Threshold Mode 1: Select Fix Threshold Mode
FIFO_INT_EN	2	DSP FIFO Interrupt Enable. Enable the FIFO To cause an interrupt when ready to be read.	
RFU	3	Reserved	N/A
DSP_INTF	4	DSP Interrupt Flag. Set to 1 by hardware when swipe is complete. Can be set or cleared by software.	
T1_INTF	5	T1 FIFO Interrupt Flag. Set by hardware when FIFO is ready for reading. Can be set or cleared by software.	
T2_INTF	6	T2 FIFO Interrupt Flag. Set by hardware when FIFO is ready for reading. Can be set or cleared by software.	
T3_INTF	7	T3 FIFO Interrupt Flag. Set by hardware when FIFO is ready for reading. Can be set or cleared by software.	
DIV_RATIO	14:8	ADC clock divide ratio. ADC_CLK freq. = PCLK freq. / (DIV_RATIO + 1)	
RFU	31:15	Reserved	N/A



17.7.16 MSR DSP Data Register (DSPDATA, Offset 0x0008)

Table 17-17. DSPDATA (Offset 0x0008)

Position	31	30	29	28	27	26	25	24
Field	DATA[31:24]							
Reset	SSSS SSSS							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	DATA[23:16]							
Reset	SSSS SSSS							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	DATA[15:8]							
Reset	SSSS SSSS							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	DATA[7:0]							
Reset	SSSS SSSS							
Access	RW							

Name	Bits	Description	Settings
DATA	31:0	MSR Indirect Data Register. Indirectly access MSR settings registers by access this register. This register represents an index register whose index is written to DSPADDR register as shown in 17.7.1. The registers names, indices and definitions are described in 17.7.2 through 17.7.14.	



17.7.17 Track 1 Data Output Register (T1FIFO, Offset 0x000C)

Table 17-18. T1FIFO (Offset 0x000C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	T1DATA[15:8]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	T1DATA[7:0]							
Reset	0000 0000							
Access	R							
Name	Bits	Description						Settings
T1DATA	15:0	Track 1 FIFO output data						
RFU	31:16	Reserved						N/A



17.7.18 Track 2 Data Output Register (T2FIFO, Offset 0x0010)

Table 17-19. T2FIFO (Offset 0x0010)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	T2DATA[15:8]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	T2DATA[7:0]							
Reset	0000 0000							
Access	R							
Name	Bits	Description					Settings	
T2DATA	15:0	Track 2 FIFO output data						
RFU	31:16	Reserved					N/A	



17.7.19 Track 3 Data Output Register (T3FIFO, Offset 0x0014)

Table 17-20. T3FIFO (Offset 0x0014)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	T3DATA[15:8]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	T3DATA[7:0]							
Reset	0000 0000							
Access	R							
Name	Bits	Description						Settings
T3DATA	15:0	Track 3 FIFO output data						
RFU	31:16	Reserved						N/A



17.7.20 DSP Interrupt Flag Register (DSPINT, Offset 0x0018)*

Table 17-21. DSPINT (Offset 0x0018)*

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	DSP_T3_INTF	DSP_T2_INTF	DSP_T1_INTF	DSP_INTF	SWDONE	T3OVF	T2OVF	T1OVF
Reset	0	0	0	0	0	0	0	0
Access	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C	RW1C

Name	Bits	Description	Settings
T1OVF	0	T1 FIFO Overflow Interrupt Flag. Hardware sets this bit to a 1 when the Track 1 FIFO is completely full. Writing a 1 to this bit clears the flag. Writing a 0 has no effect.	0: No FIFO overflow condition 1: FIFO overflow condition has occurred
T2OVF	1	T2 FIFO Overflow Interrupt Flag. Hardware sets this bit to a 1 when the Track 2 FIFO is completely full. Writing a 1 to this bit clears the flag. Writing a 0 has no effect.	0: No FIFO overflow condition 1: FIFO overflow condition has occurred
T3OVF	2	T3 FIFO Overflow Interrupt Flag. Hardware sets this bit to a 1 when the Track 3 FIFO is completely full. Writing a 1 to this bit clears the flag. Writing a 0 has no effect.	0: No FIFO overflow condition 1: FIFO overflow condition has occurred
SWDONE	3	DSP Swipe Done Interrupt Flag. Hardware sets this bit to a 1 when the card swipe is complete. Writing a 1 to this bit clears the flag. Writing a 0 has no effect.	0: Swipe not complete 1: Swipe complete
DSP_INTF	4	DSP Interrupt Flag. This is a mirror of DSP_CTRL.DSP_INTF. Writing a 1 to this bit clears the flag. Writing a 0 has no effect.	
DSP_T1_INTF	5	T1 FIFO Interrupt Flag. This is a mirror of DSP_CTRL.DSP_T1_INTF. Writing a 1 to this bit clears the flag. Writing a 0 has no effect.	
DSP_T2_INTF	6	T2 FIFO Interrupt Flag. This is a mirror of DSP_CTRL.DSP_T2_INTF. Writing a 1 to this bit clears the flag. Writing a 0 has no effect.	
DSP_T3_INTF	7	T3 FIFO Interrupt Flag. This is a mirror of DSP_CTRL.DSP_T3_INTF. Writing a 1 to this bit clears the flag. Writing a 0 has no effect.	
RFU	31:8	Reserved	N/A

* From Silicon Revision B1. Does not exist on Revision A3 and prior.



17.7.21 DSP Interrupt Enable Register (DSPINTEN, Offset 0x001C)*

Table 17-22. DSPINTEN (Offset 0x001C)*

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	T3F_INTEN	T2F_INTEN	T1F_INTEN	DSP_INTFEN	SWDONEEN	T3OVFEN	T2OVFEN	T1OVFEN
Reset	0	0	0	1	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
T1OVFEN	0	T1 FIFO Overflow Interrupt Enable	0: Disable T1 FIFO overflow interrupt 1: Enable T1 FIFO overflow interrupt
T2OVFEN	1	T2 FIFO Overflow Interrupt Enable	0: Disable T2 FIFO overflow interrupt 1: Enable T2 FIFO overflow interrupt
T3OVFEN	2	T3 FIFO Overflow Interrupt Enable	0: Disable T3 FIFO overflow interrupt 1: Enable T3 FIFO overflow interrupt
SWDONEEN	3	DSP Swipe Done Interrupt Enable	0: Disable SWDONE interrupt 1: Enable SWDONE interrupt
DSP_INTFEN	4	DSP INTF Interrupt Enable. This interrupt is enabled by default when the DSP is enabled.	0: Disable DSP_INTF interrupt 1: Enable DSP_INTF interrupt
T1F_INTEN	5	T1 FIFO Interrupt Enable. Setting this bit or setting DSPCTRL.FIFO_INT_EN enables Track 1 interrupts for FIFO half full.	0: T1 FIFO half full interrupt disabled if DSPCTRL.FIFO_INT_EN = 0. 1: T1 FIFO half full interrupt enabled.
T2F_INTEN	6	T2 FIFO Interrupt Enable. Setting this bit or setting DSPCTRL.FIFO_INT_EN enables Track 2 interrupts for FIFO half full.	0: T2 FIFO half full interrupt disabled if DSPCTRL.FIFO_INT_EN = 0. 1: T2 FIFO half full interrupt enabled.
T3F_INTEN	7	T3 FIFO Interrupt Enable. Setting this bit or setting DSPCTRL.FIFO_INT_EN enables Track 3 interrupts for FIFO half full.	0: T3 FIFO half full interrupt disabled if DSPCTRL.FIFO_INT_EN = 0. 1: T3 FIFO half full interrupt enabled.
RFU	31:8	Reserved	N/A

* From Silicon Revision B1. Does not exist on Revision A3 and prior.



17.7.22 DSP Status Register (DSPSTAT, Offset 0x0020)*

Table 17-23. DSPSTAT (Offset 0x0020)*

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[4:0]					T3EMPTYF	T2EMPTYF	T1EMPTYF
Reset	0000 0					1	1	1
Access	R					R	R	R

Name	Bits	Description	Settings
T1EMPTYF	0	T1 FIFO Empty Status Flag. This flag is set by hardware when the Track 1 FIFO is empty and cleared when the Track 1 FIFO is not empty.	0: Track 1 FIFO not empty 1: Track 1 FIFO empty
T2EMPTYF	1	T2 FIFO Empty Status Flag. This flag is set by hardware when the Track 2 FIFO is empty and cleared when the Track 2 FIFO is not empty.	0: Track 2 FIFO not empty 1: Track 2 FIFO empty
T3EMPTYF	2	T3 FIFO Empty Status Flag. This flag is set by hardware when the Track 3 FIFO is empty and cleared when the Track 3 FIFO is not empty.	0: Track 3 FIFO not empty 1: Track 3 FIFO empty
RFU	31:3	Reserved	N/A

* From Silicon Revision B1. Does not exist on Revision A3 and prior.

18.0 Mono LCD Controller

The LCD interface allows an external LCD display to be accessed directly over the APB. This interface has the following features:

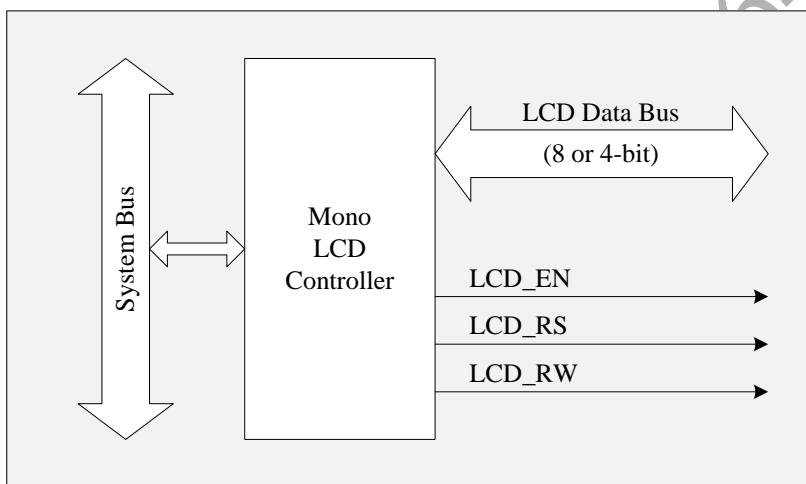
- Directly compatible with popular monochrome LCD Displays, text and graphic modes.
- Supports 4-/8-bit Data interface with 3 control signals.
- Read and writes to external LCD module supported register operations.
- Programmable read and write cycle times.

18.1 Operation Overview

The LCD interface is a bus controller. It can manage data exchange between the device and an LCD module. When automaton is activated, bus signals are generated according to the timing defined in the LCD Enable Period Register (LCD_EPR). In Manual Mode, LCD_EN, LCD_RS and LCD_RW pins are controlled by the corresponding bits in the LCD Control Register (LCD_CN).

shows the simplified block diagram depicts the flow of data for the Mono LCD Controller.

Figure 18-1. Mono LCD Controller Block Diagram



When configured as 4-bit (nibble) data format:

- Read: LCD_DATA register is filled by performing two consecutive 4-bit read accesses from LCD_DATA[7:4] pins. The first 4 bit will be read into LCD_DATA[7:4] register bits, the next 4 bits will fill LCD_DATA[3:0] register bits.
- Write: Send two consecutive 4-bit data through LCD_DATA[7:4] pins (LCD_DATA[3:0] register bits, then LCD_DATA[7:4] register bits).

18.2 Automatic Mode

This mode is switched on when automaton is activated (LCD_CN.AE=1). The controller is responsible for managing bus timing and the type of accesses: 8-bit or 4-bit. The polarity of the LCD_RS and LCD_RW signals must be set by software for each access.

The monochrome LCD controller automatic mode timing is described below. In this document, functional timing is defined as the timing relationships created by the synchronous logic included in the LCD controller, using the PCLK as the time base. This timing does not include combinational logic propagation delays into and out of the device, as these values depend on data gathered after final placement and routing of the design. Final LCD_EPR settings will need to account for the input and output delays, but these are expected to be much less than one PCLK cycle.

The period of the interface LCD Enable signal (LCD_EN) is defined by the value in user programmable LCD Enable Period Register (LCD_EPR). This delay is measured in the number of APB clock cycles (PCLK). All other functional timing is referenced to this value.

Moreover when automaton is activated, the LCD interface can manage either 8 or 4-bit data bus width according to the Bus Width Select (LCD_CN.NIBBLE) bit. In nibble mode, 8-bit data put in the data register LCD_DATA, is sent in two consecutive 4-bit write accesses from LCD_DATA[7:4] pins. In the same way, the LCD interface gets 8-bit data in two consecutive 4-bit read accesses.

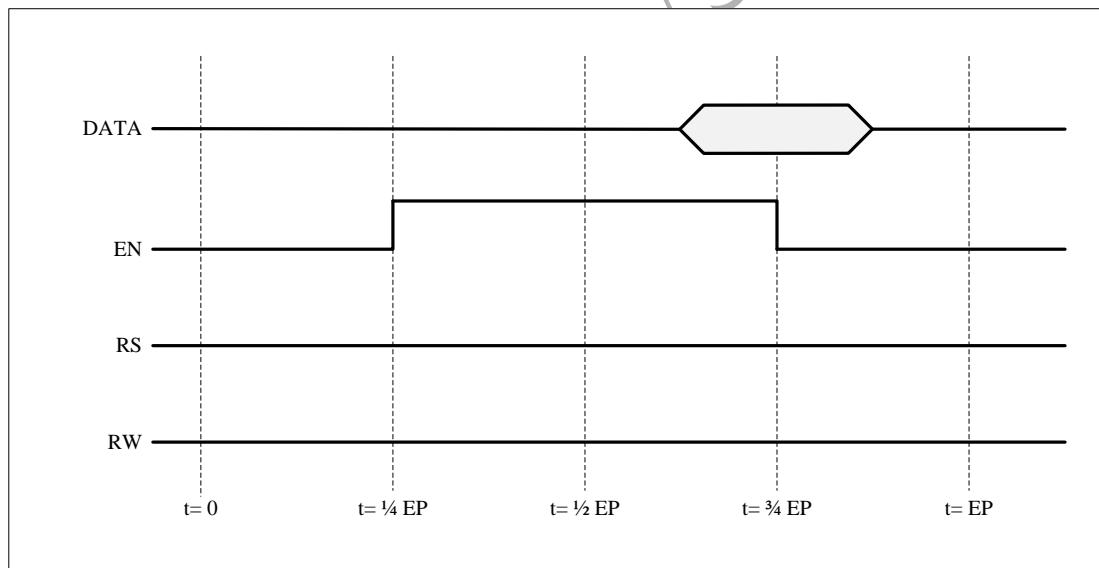
18.2.1 Automatic Read Operation

To perform a read access, the LCD_CN.READ bit must be set to 1. In automatic mode, the controller will automatically update the outputs of the LCD_EN, LCD_RW and LCD_RS pins according to the content of the LCD_CN register. The duration of the read accesses is specified by LCD_EPR register.

Automatic Read Operation:

- $t=0$: Output LCD_RS, LCD_RW.
- $t = \frac{1}{4} EP$: Assert LCD_EN
- $t = \frac{3}{4} EP$: De-assert LCD_EN and sample data.
- LCD_RS, LCD_RW and IO direction will change only at the next transaction (if necessary).

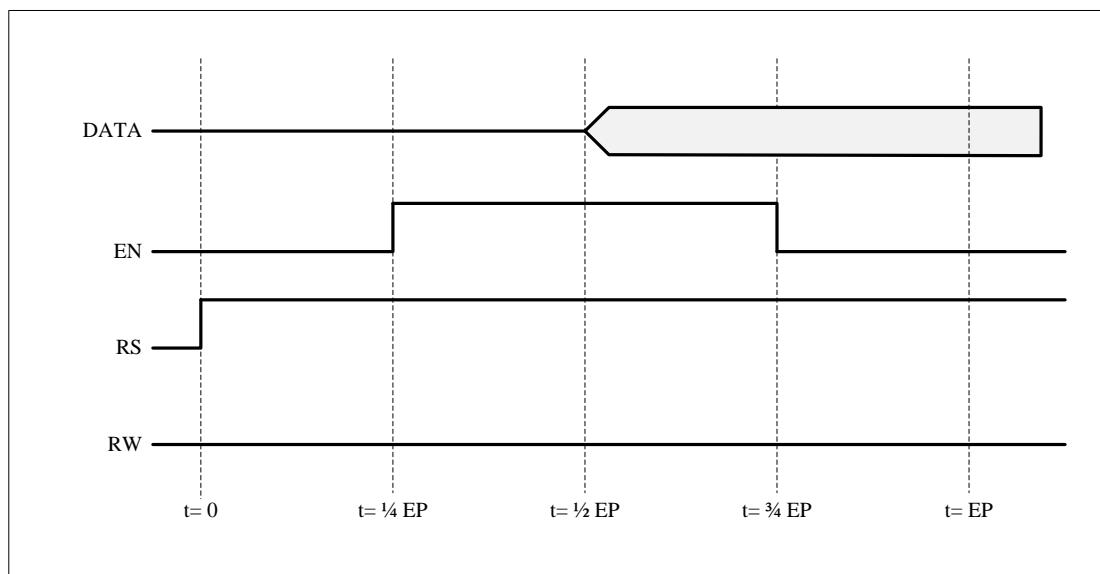
Figure 18-2. LCD Read Operation



18.2.2 Automatic Write Operation

A write access starts when a data is written into LCD_DATA. The correct logic levels of LCD_RW and LCD_RS pins must be set before writing to the data register. Like read accesses the bus timing is defined by the LCD_EPR register.

- $t = 0$: Output LCD_RS, LCD_RW
- $t = \frac{1}{4} EP$: Assert LCD_EN
- $t = \frac{1}{2} EP$: Output Data
- $t = \frac{3}{4} EP$: De-assert LCD_EN
- LCD_RS, LCD_RW, IO direction and IO data will change only at the next transaction (if necessary).

**Figure 18-3. LCD Write Operation**

18.3 Manual Mode

When `LCD_CN.AE=0`, automaton is disabled. In this case the software is responsible for the bus timing and control signal sequencing. The `LCD_EN`, `LCD_RS` and `LCD_RW` pins function as GPIOs.

As in the case of automaton, a read access is performed when the `LCD_CN.READ` bit is set. It is software responsibility to ensure data on bus is steady.

In the same way, a write access starts when data is written into `LCD_DATA`. It is the software responsibility to meet the LCD module write timing requirement.

18.4 Interrupt Operation

The LCD controller provides two individually maskable interrupts:

- Read Complete: an interrupt is asserted when read operation completes
- Write Complete: an interrupt is asserted when write operation completes

The LCD Status Register (`LCD_SR`) indicates the interrupt source. To acknowledge an interrupt, the corresponding bit in `LCD_SR` must be cleared.

It is recommended to acknowledge an interrupt before enabling it (unmask operation) to avoid a previous event triggering.



18.5 Mono LCD Controller Registers

Address assignments for the LCD Interface registers are outlined in .

Table 18-1. Mono LCD Controller Registers (Base ADDR = 0x4003_0000)

Offset	Access	Register	Description
0x0000	RW	LCD_CN	LCD Control Register
0x0004	RW	LCD_EPR	LCD Enable Period Register
0x0008	RW	LCD_SR	LCD Status Register
0x000C	RW	LCD_DATA	LCD Data Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



18.5.1 LCD Control Register (LCD_EN, Offset 0x0000)

Table 18-2. LCD CN (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	READ	ENB	RW	RS	NIBBLE	AE	RXIE	TXIE
Reset	1	0	0	0	0	0	0	0
Access	RS	RW	RW	RW	R/W	RW	RW	RW

Name	Bits	Description	Settings
TXIE	0	Data Transmit Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled
RXIE	1	Data Receive Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled
AE	2	Automatic Mode Enable	0: Manual Mode 1: Automatic Mode
NIBBLE	3	Bus Width Select	0: 8 bit bus width 1: 4 bit bus width (This feature is only available when AUTOEN=1)
RS	4	Register Select	This bit sets the logic level of the LCD_RS pin.
RW	5	Read Write	This bit sets the logic level of the LCD_RW pin.
ENB	6	Enable (Active Low)	In Automatic Mode, this bit sets the polarity of the LCD_EN pin. 0: LCD_EN is active high. 1: LCD_EN is active low. In Manual Mode, this bit directly sets the LCD_EN pin state.
READ	7	Read	0: This bit is cleared by hardware when the read has completed. Writing a '0' to this bit has no effect on the read operation. 1: Setting this bit initiates a read access on the bus.
RFU	31:8	Reserved	N/A



18.5.2 LCD Enable Period Register (LCD_EPR, Offset 0x0004)

Table 18-3. LCD_EPR (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	EP[7:0]							
Reset	0001 1000							
Access	RW							

Name	Bits	Description	Settings
EP	7:0	Enable Period	This register specifies the number of system clock cycles for an access. It is twice the asserted time of the enable signal (LCD_EN pin). This register has a minimum value of 4. Any value less than 4 will be set to 4.
RFU	31:8	Reserved	N/A



18.5.3 LCD Status Register (LCD_SR, Offset 0x0008)

Table 18-4. LCD_SR (Offset 0x0008)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[4:0]					READY	RXIS	TXIS
Reset	0 0000					1	0	0
Access	R					R	RW	RW

Name	Bits	Description	Settings
TXIS	0	Transmission Status	0: The content of the data register has not yet been sent. 1: The content of the data register has been sent. This bit is set by hardware and must be cleared by software to be acknowledged.
RXIS	1	Receive Status	0: No data has been received. 1: Data is available in LCD_DATA register. This bit is set by hardware and must be cleared by software to be acknowledged.
READY	2	LCD Ready	0: The interface is busy. 1: The interface is ready for read or write operation This bit is read only.
RFU	31:3	Reserved	N/A



18.5.4 LCD Data Register (LCD_DATA, Offset 0x000C)

Table 18-5. LCD DATA (Offset 0x000C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	DATA[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description	Settings					
DATA	7:0	Data	Data register for LCD read/write operation.					
RFU	31:8	Reserved	N/A					



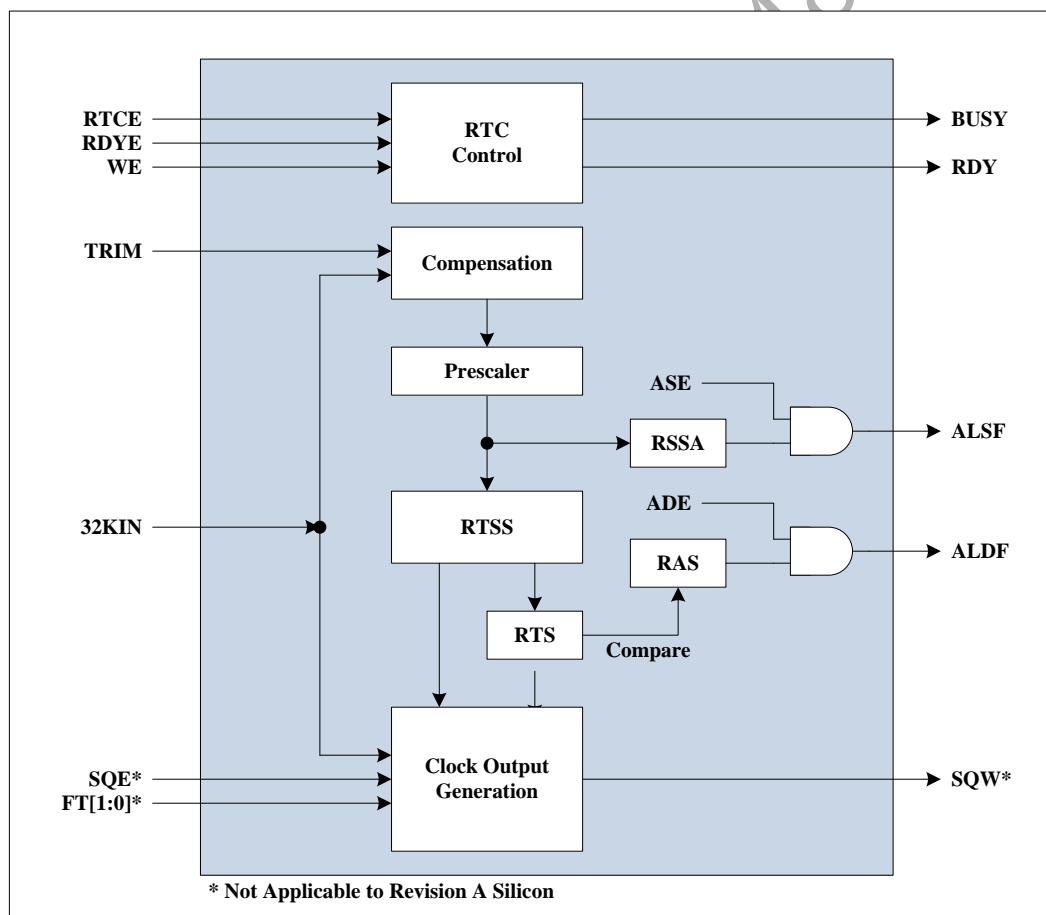
19.0 Real Time Clock (RTC)

The Real Time Clock (RTC) is a binary timer that keeps the time of day and provides time-of-day and sub-second alarm functionality in the form of system interrupts. The RTC timebase is created through a 32768Hz crystal connected between the 32KIN and 32KOUT pins. No external capacitors are required for crystal oscillation.

The RTC counters represent absolute seconds (~136 years) and sub-seconds (in 1/256 second resolution). A separate auto-reload sub-second alarm counter can be used to generate interval alarms with granularity of 1/256 seconds. The 32-bit seconds counter increments with each rollover of the 256Hz sub-second counter. The 32-bit counter can be used with the programmable time-of-day comparison alarm to provide a single event timer. The RTC must be stopped for the counter registers to initially be written, but once enabled, the RTC counts continuously as long as it is enabled and does not stop for reads of the counter registers. The RTC also supports a digital trim option for those applications requiring high accuracy.

The RTC is battery-backed. Once enabled, it will continue running as long as the RTC is enabled and the VBAT supply remains within the acceptable range given in the datasheet.

Figure 19-1. RTC Block Diagram



19.1 RTC Alarm Functions

The RTC provides time-of-day and sub-second interval alarm functions. The time-of-day alarm is accomplished by matching the count values in the counter register with the user configured alarm register value. The sub-second interval alarm provides an auto-reload timer that is driven by the trimmed RTC clock source.



19.1.1 Time-of-Day Alarm

The RTC Time-of-Day Alarm Register (RTC_RAS) is programmed with the desired 20-bit value that represents the specific timer value that matches the RTC Second Register (RTC_SEC) when alarm will trigger. **The time-of-day alarm can be programmed to any future value between 1 second and 12 days relative to the current time with a resolution of 1 second.** The time-of-day alarm must be disabled before the changing RTC_RAS registers.

The time-of-day alarm is a single event that will set the Time-of-Day Alarm Interrupt Flag (ALDF) to 1 when RTC_RAS[19:0] = RTC_SEC[19:0] and sub-second counter rollover.

Setting the ALDF to 1 in software will cause an interrupt request to the processor if the Time of Date Interrupt Enable (ADE) bit is set to 1 and the system interrupt enable is set.

19.1.2 Sub-Second Alarm

The RTC Sub-Second Alarm Register (RTC_RSSA) stores the sub-second interval alarm value for the sub-second alarm function. The RSSA register is 16 bits wide allowing a maximum interval alarm of 16 second and with programming resolution of ~244 us for a 4096 Hz RTC input clock. The sub-second interval alarm must be disabled before changing the interval alarm value. The sub-second alarm can be enabled by setting the Sub-second Alarm Enable (ASE) to 1. The alarm is generated (ALSF=1) when the Sub-Second Alarm register RTC_RSSA rolls over from FFFF to 0000.

The delay (uncertainty) associated with the enabling of the interval sub-second alarm is up to one period of the sub-second clock (for an input of 4096Hz, this is ~244us). Thus, the same uncertainty will be associated with the first interval alarm. Thereafter, if the interval alarm remains enabled, the alarm will trigger after each sub-second interval as defined. This is because the sub-second alarm is constructed as an auto-reload timer; the alarm value (RSSA) is only loaded to the counter by a timer rollover. Note that enabling the sub-second alarm with sub-second interval alarm register programmed to 0 results in the maximum sub-second alarm interval. The sub-second interval counter sources its clock from the trimmed RTC input clock period.

19.2 RTC Register Access

Access to specific registers may be restricted to prevent software reading from or writing to RTC registers while they are being updated by the RTC hardware.

19.2.1 RTC Register Write Protection

The BUSY bit (BUSY) is a read-only status bit. Hardware sets the BUSY bit when any of the following conditions occur:

- 1) System reset,
- 2) Software writes to RTC count registers or trim or
- 3) Software changes RTCE, ADE, or ASE.

When the BUSY bit is set by hardware, writes to the above RTC control bits and count registers are blocked by hardware. The BUSY bit remains active until a synchronized 32kHz version of the register (or bit) is in place. This takes place when the next rising edge of the 32kHz clock occurs, which means that the BUSY bit is set for no longer than one 4kHz clock = ~250µs. Therefore, write should not be considered complete until hardware clears the BUSY bit. This is an indication that a 32kHz synchronized version of the register bit(s) is in place.

Once the BUSY bit is cleared to 0, additional writes can be performed as permitted by individual count or alarm-enable bits.

19.2.2 RTC Register Read Protection

The Ready (RTC_CN.RDY) bit indicates when the RTC count registers contains valid data. The RDY bit is cleared by hardware approximately one 4kHz clock before the ripple occurs through the RTC counter chain (RTS, RTSS) and is set once again immediately after the ripple occurs. The period of the RDY bit set/clear activity (as controlled

by hardware) is therefore $1/256\text{Hz} = 3.9\text{ms}$, providing a very large window during which the RTC count registers can be read. The RDY bit can be cleared by software at any time and remains clear until set by hardware again. A separate Ready Enable (RTC_CN.RDYE) bit is provided for the purpose of generating an interrupt whenever the RDY bit is set by hardware. This interrupt can be used to signal the start of a new RTC read window.

19.2.3 RTC Count Register Access

The RTC Count registers (RTC_SEC, RTC_SSEC) should only be read when RDY=1. Data read from these registers when RDY = 0 should be considered invalid. To write the RTC count registers, the RTC Enable (RTCE) bit must be cleared to 0. Clearing of the RTCE bit is permitted only when the Write Enable (WE) bit is set to 1 and is governed by the BUSY bit signaling process (i.e., the BUSY bit is deasserted once a synchronized 32kHz version of the bit is in place). Writes to each RTC count register should also obey the rules associated with BUSY bit signaling.

19.2.4 RTC Alarm Register Access

The RTC Alarm registers (RTC_RSSA, RTC_RAS) can be read at any time. To write to the alarm register, the corresponding alarm enable must be disabled first (ASE=0 or ADE=0). Clearing these bits requires monitoring the BUSY bit to assess completion of the write. Once the alarm is disabled, the associated RTC alarm register(s) can be updated by user code.

19.2.5 RTC Trim Register Access

The RTC Trim Register (RTC_TRIM) can be read at any time. To write to this register, the Write Enable (WE) bit must be set to 1 and BUSY=0.

19.3 RTC Calibration

The uncompensated accuracy of the RTC is a function of the attached crystal. A digital trim facility allows the device to compensate for up to +/- 127 ppm as designated by the RTC_TRIM register. A square wave output* can be measured and the accuracy of the RTC determined.

The steps are as follows:

1. Set FT[1:0] to the desired output frequency.
2. Set SQE=1 to enable square wave output.
3. Measure square wave output and compare to an accurate reference clock.
4. Set WE = 1 to adjust RTC_TRIM.
5. Repeat the above as necessary until desired accuracy is achieved.

*Note: Not Applicable to Revision A Silicon

19.4 RTC Registers

Address assignments for registers are outlined below. Reserved register bits should only be written as 0.

Table 19-1. RTC Register Address (Base ADDR = 0x4000 6000)

Offset	Access	Register	Description
0x0000	RW	RTC_SEC	RTC Second Counter Register
0x0004	RW	RTC_SSEC	RTC Sub-second Counter Register
0x0008	RW	RTC_RAS	RTC Alarm Time-of-Day Register
0x000C	RW	RTC_RSSA	RTC Sub-second Alarm Register
0x0010	RW	RTC_CN	RTC Control Register
0x0014	RW	RTC_TRIM	RTC Trim Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



19.4.1 RTC Second Counter Register (RTC_SEC, Offset 0x0000)

This register is battery-backed. The value of this register is valid only when RDY=1. This register can only be written to when RTCE=BUSY=0. Change to this register is effective only after BUSY is cleared from 1 to 0.

Table 19-2. RTC_SEC (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RTS[31:24]							
BOR	0000 0000							
Other Reset	N/A							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	RTS[23:16]							
BOR	0000 0000							
Other Reset	N/A							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	RTS[15:8]							
BOR	0000 0000							
Other Reset	N/A							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	RTS[7:0]							
BOR	0000 0000							
Other Reset	N/A							
Access	RW							
Name	Bits	Description					Settings	
RTS	31:0	RTC Second Counter. This register contains the 32-bit second counter.						



19.4.2 RTC Sub-second Counter Register (RTC_SSEC, Offset 0x0004)

This register can only be read when RDY=1. The value of this register is valid only when RDY=1. This register can only be written to when RTCE=BUSY=0. Change to this register is effective only after BUSY is cleared from 1 to 0.

Table 19-3. RTC_SSEC (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field				RFU[7:0]				
BOR				0000 0000				
Other Reset				0000 0000				
Access				R				
Position	23	22	21	20	19	18	17	16
Field				RFU[7:0]				
BOR				0000 0000				
Other Reset				0000 0000				
Access				R				
Position	15	14	13	12	11	10	9	8
Field				RFU[7:0]				
BOR				0000 0000				
Other Reset				0000 0000				
Access				R				
Position	7	6	5	4	3	2	1	0
Field				RTSS[7:0]				
BOR				0000 0000				
Other Reset				N/A				
Access				RW				

Name	Bits	Description	Settings
RTSS	7:0	RTC Sub-second Counter. This counter increments at 256Hz. RTC_SEC is incremented when this register rolls over from 0xFF to 0x00.	
RFU	31:8	Reserved	N/A



19.4.3 RTC Time-of-Day Register (RTC_RAS, Offset 0x0008)

This register is battery-backed. This register can only be written to when BUSY=0 and (RTCE=0 or ADE=0).

Table 19-4. RTC_RAS (Offset 0x0008)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
BOR	0000 0000							
Other Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[3:0]				RAS[19:16]			
BOR	0000				0000			
Other Reset	0000				N/A			
Access	R				RW			
Position	15	14	13	12	11	10	9	8
Field	RAS[15:8]							
BOR	0000 0000							
Other Reset	N/A							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	RAS[7:0]							
BOR	0000 0000							
Other Reset	N/A							
Access	RW							

Name	Bits	Description	Settings
RAS	19:0	Time-of-day Alarm.	
RFU	31:20	Reserved	N/A



19.4.4 RTC Sub-second Alarm Register (RTC_RSSA, Offset 0x000C)

This register is battery-backed. This register can only be written to when BUSY=0 and (ASE=0 or RTCE=0).

Table 19-5. RTC_RSSA (Offset 0x000C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
BOR	0000 0000							
Other Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
BOR	0000 0000							
Other Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RSSA[15:8]							
BOR	0000 0000							
Other Reset	N/A							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	RSSA[7:0]							
BOR	0000 0000							
Other Reset	N/A							
Access	RW							

Name	Bits	Description	Settings
RSSA	15:0	RTC sub-second alarm. This register contains the reload value for the sub-second alarm.	
RFU	31:16	Reserved	N/A



19.4.5 RTC Control Register (RTC_CN, Offset 0x0010)

This register is battery-backed. Unrestricted write to RDY, RDYE and WE. Write to other bits are allowed only when BUSY=0. Additional restriction may apply. See bit description for details.

Table 19-6. RTC CN (Offset 0x0010)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
BOR	0000 0000							
Other Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
BOR	0000 0000							
Other Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	WE	RFU	RFU	X32KMD[1:0]		FT[1:0]*		SQE*
BOR	0	0	0	00		00		0
Other Reset	0	0	0	00		00		0
Access	RW	R	R	RW		RW		RW
Position	7	6	5	4	3	2	1	0
Field	ALSF	ALDF	RDYE	RDY	BUSY	ASE	ADE	RTCE
BOR	0	0	0	0	1	0	0	0
Other Reset	N/A	N/A	0	0	1	N/A	N/A	N/A
Access	R	R	RW	RC	R	R	RW	RW

Name	Bits	Description	Settings
RTCE	0	Real Time Clock Enable. This bit enables the Real Time Clock. This bit can only be written when WE=1 and BUSY =0. Change to this bit is effective only after BUSY is cleared from 1 to 0.	0: Real time clock disabled 1: Real time clock enabled
ADE	1	Alarm Time-of-Day Interrupt Enable. Change to this bit is effective only after BUSY is cleared from 1 to 0.	0: Interrupt disabled. 1: Interrupt enabled.
ASE	2	Alarm Sub-second Interrupt Enable. Change to this bit is effective only after BUSY is cleared from 1 to 0.	0: Interrupt disabled. 1: Interrupt enabled.
BUSY	3	RTC Busy. This bit is set to 1 by hardware when changes to RTC registers required a synchronized version of the register to be in place. This bit is automatically cleared by hardware.	
RDY	4	RTC Ready. This bit is set to 1 by hardware when the RTC count registers update. It can be cleared to 0 by software at any time. It will also be cleared to 0 by hardware just prior to an update of the RTC count register.	
RDYE	5	RTC Ready Interrupt Enable.	0: Interrupt disabled. 1: Interrupt enabled.

Name	Bits	Description	Settings
ALDF	6	Time-of-Day Alarm Interrupt Flag. This alarm is qualified as wake-up source to the processor.	0: No interrupt pending. 1: Interrupt pending.
ALSF	7	Sub-second Alarm Interrupt Flag. This alarm is qualified as wake-up source to the processor.	0: No interrupt pending. 1: Sub-Second register auto-reload occurs.
SQE	8	Square Wave Output Enable. *Note: Not Applicable to Revision A Silicon	0: Square wave output disabled. 1: Square wave specified by FT is outputted on the SQW pin.
FT	10:9	Frequency Output Selection. When SQE=1, these bits specify the output frequency on the SQW pin. *Note: Not Applicable to Revision A Silicon	00: 1 Hz (Compensated) 01: 512 Hz (Compensated) 1x: RTC Input Clock / 8
X32KMD	12:11	32KHz Oscillator Mode.	00: Always operate in Noise Immune Mode. Oscillator warm-up required. 01: Always operate in Quiet Mode. No oscillator warm-up required. 10: Operate in Noise Immune Mode normally, switch to Quiet Mode on Stop Mode entry. Will wait for 32K oscillator warm-up before code execution on Stop Mode exit. 11: Operate in Noise Immune Mode normally, switch to Quiet Mode on Stop Mode entry. Will not wait for 32K oscillator warm-up before code execution on Stop Mode exit.
RFU	14:13	Reserved	N/A
WE	15	Write Enable. This register bit serves as a protection mechanism against unintentional writes to critical RTC bits.	0: Write to RTC_TRIM and RTCE is ignored. 1: Write to RTC_TRIM and RTCE is allowed.
RFU	31:16	Reserved	N/A



19.4.6 RTC Trim Register (RTC_TRIM, Offset 0x0014)

This register is battery-backed. This register can only be written to when BUSY=0 and WE=1. Change to this register is effective only after BUSY is cleared from 1 to 0.

Table 19-7. RTC TRIM (Offset 0x0014)

Position	31	30	29	28	27	26	25	24
Field	VBATTMR[31:24]*							
BOR	0000 0000							
Other Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	VBATTMR[23:16]*							
BOR	0000 0000							
Other Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	VBATTMR[15:8]*							
BOR	0000 0000							
Other Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	TRIM[7:0]							
BOR	0000 0000							
Other Reset	N/A							
Access	RW							

Name	Bits	Description	Settings
TRIM	7:0	RTC Trim. This register contains the 2's complement value that specifies the trim resolution. Each increment or decrement of the bit adds or subtracts 1ppm at each 4KHz clock value, with a maximum correction of +/- 127ppm.	
VBATTMR*	31:8	VBAT Timer Value. When RTC is running off of VBAT, this field is incremented every 32 seconds.	

* Not applicable on revision A silicon.



20.0 Secure Keyboard

The Secure Keyboard provides a tamper resistant keyboard scan methodology which incorporates randomized scan start position, scan direction and pulse duration. Randomized scan parameters are generated by the system True Random Number Generator (TRNG). A maximum of 8 total pins may be selected as output or input to match the row and column layout of the physical keypad. Interrupts signal key press, key release, and input overflow. The hardware supports storage of up to four key events. The Secure Keyboard also provides user selectable keypress debounce time.

20.1 Keyboard Scanning

The Secure Keyboard cycles through all outputs and senses all inputs on each scan cycle. The beginning and end output pin, the direction of the scan and the scan pulse are randomized. Keyboard pins used and key debounce time are set via KBD_CR0 and KBD_CR1 registers.

20.2 Drive and Sense Pins

The rows and columns of a keyboard must be interfaced to the KBDIO0 – KBDIO7 pins. Any number of rows and columns may be used with the limitation that at least one row and one column must be connected. Output pins start at KBDIO0 and must be contiguous. Input pins are allocated following the last output pin. For example:

- 4 row, 2 column keypad
- 4 outputs, 2 input
- KBDIO0-KBDIO3 tied to keypad rows
- KBDIO4-KBDIO5 tied to keypad columns
- KBD_CR0 = 0x000F (Set KBDIO0-3 to outputs)
- KBD_CR1 = 0x0401 (Set 4 output pins, enable scanning)

20.3 Key Debounce

Key debounce is enforced by the keyboard controller to prevent false key push or key release detections. The debouncing time is settable in the KBD_CR1 register and may range between 4.1 milliseconds and 12.3 milliseconds.

20.4 Key Event Storage

Up to four key press and key release events may be stored in registers KBD_K0R through KBD_K3R. When unloading key events during keyboard processing, user code should read the events starting at KBD_K0R and proceed through KBD_K3R. Any register with READ bit set should be skipped. User code should terminate unloading of key registers after KBD_K3R is unloaded or when any NEXT bit is clear.

20.5 Interrupts

Three interrupt sources are provided to signal keyboard events: Push, Release and Overrun. Each interrupt may be individually enabled or disabled via Interrupt Enable Register (IER). When an interrupt occurs, the appropriate flag in Interrupt Status Register (ISR) is set. Once the interrupt service code is run, the interrupt flag should be cleared so that following interrupts may signal new events.



20.6 Secure Keyboard Registers

Address assignments for the Secure Keyboard registers are outlined in Table 20-1.

Table 20-1. Secure Keyboard Register Addresses (Base ADDR = 0x4003_2000)

Offset	Access	Register	Description
0x0000	RW	KBD_CRO	Control Register 0
0x0004	RW	KBD_CR1	Control Register 1
0x0008	RW	KBD_SR	Status Register
0x000C	RW	KBD_IER	Interrupt Enable Register
0x0010	RW	KBD_ISR	Interrupt Status Register
0x0014	R	KBD_K0R	Key 0 Register
0x0018	R	KBD_K1R	Key 1 Register
0x001C	R	KBD_K2R	Key 2 Register
0x0020	R	KBD_K3R	Key 3 Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



20.6.1 Keyboard Control Register 0 (KBD_CR0, Offset 0x0000)

Table 20-2. KBD CR0 (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	7	6	5	4	3	2	1	0
Field	IOSEL[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
IOSEL	7:0	Input Output Select Bits. Each bit of IOSEL selects the pin direction for the corresponding KBDIO pin. If IOSEL[0] = 1, KBDIO0 is an output.	0: Corresponding KBDIO pin is an input. 1: Corresponding KBDIO pin is an output.
RFU	31:8	Reserved	N/A



20.6.2 Keyboard Control Register 1 (KBD_CR1, Offset 0x0004)

Table 20-3. KBD CR1 (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8		
Field	DBTM[2:0]				RFU		OUTNB[2:0]			
Reset	000			00		000				
Access	RW			R		RW				

Position	7	6	5	4	3	2	1	0
Field	RFU[5:0]							CLEAR
Reset	0000 00							0
Access	R							RW

Name	Bits	Description	Settings
AUTOEN	0	Automatic Keyboard Scan Enable.	0: Keyboard scan off. 1: Keyboard scan on.
CLEAR	1	Clear bit.	0: Do not clear keypress event when key is released. 1: Clear keypress event if key is released.
RFU	7:2	Reserved.	N/A
OUTNB	10:8	Output Number. Number of KBDIO pins selected as outputs. NOTE: Output pins must be allocated contiguously starting with KBDIO0 and continuing through to KBDIO7.	
RFU	12:11	Reserved.	N/A
DBTM	15:13	Debounce Time. Number of milliseconds a keypress event must be active before it is considered actual. NOTE: Debounce time values based on system running from an external 12MHz clock source with PLL0 enabled. Other external crystal values will cause the debounce time to scale linearly from this table.	000: 4.1 001: 5.3 010: 6.5 011: 7.6 100: 8.8 101: 10.0 110: 11.2 111: 12.3
RFU	31:16	Reserved.	N/A



20.6.3 Keyboard Status Register (KBD_SR, Offset 0x0008)

Table 20-4. KBD_SR (Offset 0x0008)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	7	6	5	4	3	2	1	0
Field	RFU[6:0]							
Reset	0000 000							
Access	R							

Name	Bits	Description	Settings
BUSY	0	Busy bit. This bit is set by hardware when the automatic keyboard scan is enabled and running. This bit is clear at all other times.	0: Idle. 1: Busy.
RFU	31:1	Reserved	N/A



20.6.4 Keyboard Interrupt Enable Register (KBD_IER, Offset 0x000C)

Table 20-5. KBD_IER (Offset 0x000C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	7	6	5	4	3	2	1	0
Field	RFU[4:0]					OVERIE	RELEASEIE	PUSHIE
Reset	0000 00					0	0	0
Access	R					RW	RW	RW

Name	Bits	Description	Settings
PUSHIE	0	Push Event Enable Bit. When set, this bit enables an interrupt to be generated on a key push event. Automatic keyboard scan must be enabled.	0: Interrupt Disabled. 1: Interrupt Enabled.
RELEASEIE	1	Release Event Enable Bit. When set, this bit enables an interrupt to be generated on a key release event. Automatic keyboard scan must be enabled.	0: Interrupt Disabled. 1: Interrupt Enabled.
OVERIE	2	Overrun Event Enable Bit. When set, this bit enables an interrupt to be generated on an overrun event. Automatic keyboard scan must be enabled.	0: Interrupt Disabled. 1: Interrupt Enabled.
RFU	31:3	Reserved	N/A



20.6.5 Keyboard Interrupt Status Register (KBD_ISR, Offset 0x0010)

Table 20-6. KBD ISR (Offset 0x0010)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Other Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Other Reset	0000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Other Reset	0000 0000							
Access	R							

Position	7	6	5	4	3	2	1	0
Field	RFU[4:0]							
Other Reset	0000 0							
Access	R							

Name	Bits	Description	Settings
PUSHIS	0	Push Interrupt Flag. This bit is set by hardware when a key has been pushed. If the interrupt is enabled for this flag, a system interrupt will be fired. If the interrupt enable is not set, the flag will be set, but no interrupt will fire. This bit must be cleared by software.	
RELEASEIS	1	Release Interrupt Flag. This bit is set by hardware when a key has been released. If the interrupt is enabled for this flag, a system interrupt will be fired. If the interrupt enable is not set, the flag will be set, but no interrupt will fire. This bit must be cleared by software.	
OVERIS	2	Overrun Event Enable Bit. This bit is set by hardware when an overrun event has occurred. If the interrupt is enabled for this flag, a system interrupt will be fired. If the interrupt enable is not set, the flag will be set, but no interrupt will fire. This bit must be cleared by software.	.
RFU	31:3	Reserved	N/A



20.6.6 Keyboard Key 0 Register (KBD_K0R, Offset 0x0014)

Table 20-7. KBD_K0R (Offset 0x0014)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8
Field	RFU[2:0]			NEXT	READ	PUSH	RFU	
Reset	000			0	1	1	00	
Access	R			R	R	R	R	

Position	7	6	5	4	3	2	1	0
Field	IOOUT[2:0]			RFU	IOPIN[2:0]			
Reset	000			00	000			
Access	R			R	R			

Name	Bits	Description	Settings
IOIN	2:0	IO Input. Input pin of key event.	
RFU	4:3	Reserved	N/A
IOOUT	7:5	IO Output. Output pin of key event.	
RFU	9:8	Reserved	N/A
PUSH	10	If set to '1' the key has been released. If set to '0' the key has been pushed.	
READ	11	If set to '1' this register has been read. If set to '0' the key register has not been read since its last change.	
NEXT	12	If set to '1' one of the next key registers (x+1 to 3) contains a key event.	
RFU	31:13	Reserved	N/A



20.6.7 Keyboard Key 1 Register (KBD_K1R, Offset 0x0018)

Table 20-8. KBD_K1R (Offset 0x0018)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8
Field	RFU[2:0]			NEXT	READ	PUSH	RFU	
Reset	000			0	1	1	00	
Access	R			R	R	R	R	

Position	7	6	5	4	3	2	1	0
Field	IOOUT[2:0]			RFU	IOPIN[2:0]			
Reset	000			00	000			
Access	R			R	R			

Name	Bits	Description	Settings
IOIN	2:0	IO Input. Input pin of key event.	
RFU	4:3	Reserved	N/A
IOOUT	7:5	IO Output. Output pin of key event.	
RFU	9:8	Reserved	N/A
PUSH	10	If set to '1' the key has been released. If set to '0' the key has been pushed.	
READ	11	If set to '1' this register has been read. If set to '0' the key register has not been read since its last change.	
NEXT	12	If set to '1' one of the next key registers (x+1 to 3) contains a key event.	
RFU	31:13	Reserved	N/A



20.6.8 Keyboard Key 2 Register (KBD_K2R, Offset 0x001C)

Table 20-9. KBD_K2R (Offset 0x001C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8
Field	RFU[2:0]							
Reset	000							
Access	R							

Position	7	6	5	4	3	2	1	0
Field	IOOUT[2:0]							
Reset	000							
Access	R							

Name	Bits	Description	Settings
IOIN	2:0	IO Input. Input pin of key event.	
RFU	4:3	Reserved	N/A
IOOUT	7:5	IO Output. Output pin of key event.	
RFU	9:8	Reserved	N/A
PUSH	10	If set to '1' the key has been released. If set to '0' the key has been pushed.	
READ	11	If set to '1' this register has been read. If set to '0' the key register has not been read since its last change.	
NEXT	12	If set to '1' one of the next key registers (x+1 to 3) contains a key event.	
RFU	31:13	Reserved	N/A



20.6.9 Keyboard Key 3 Register (KBD_K3R, Offset 0x0020)

Table 20-10. KBD_K3R (Offset 0x0020)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8
Field	RFU[2:0]							
Reset	000							
Access	R							

Position	7	6	5	4	3	2	1	0
Field	IOOUT[2:0]							
Reset	000							
Access	R							

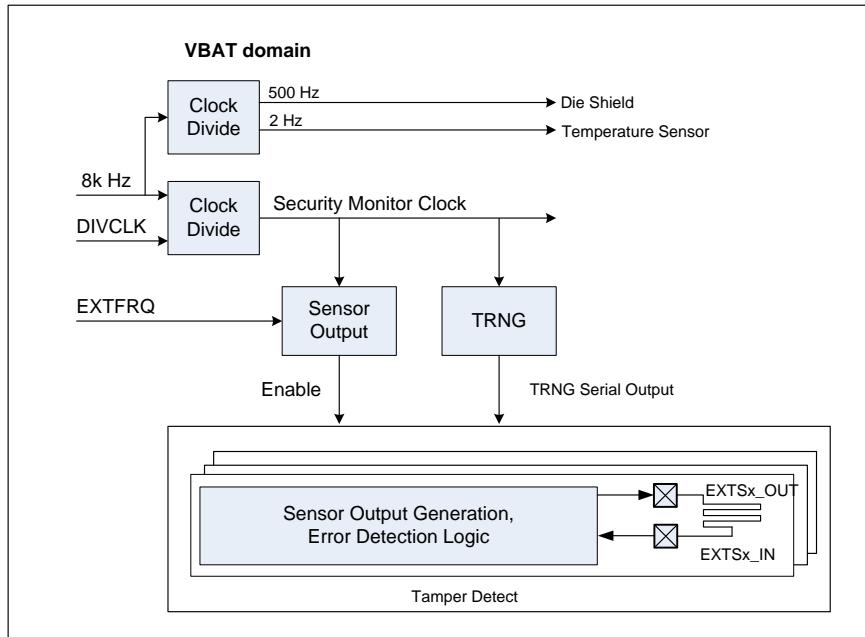
Name	Bits	Description	Settings
IOIN	2:0	IO Input. Input pin of key event.	
RFU	4:3	Reserved	N/A
IOOUT	7:5	IO Output. Output pin of key event.	
RFU	9:8	Reserved	N/A
PUSH	10	If set to '1' the key has been released. If set to '0' the key has been pushed.	
READ	11	If set to '1' this register has been read. If set to '0' the key register has not been read since its last change.	
NEXT	12	If set to '1' one of the next key registers (x+1 to 3) contains a key event.	
RFU	31:13	Reserved	N/A

21.0 Security Monitor

The Security Monitor block is used to monitor system threat conditions. These include external sensor pairs, an internal die shield, temperature, battery and voltage monitors. An internal 8kHz clock provides time base for the security monitor block. This clock can be divided down by setting the Clock Divide bits (EXTSCN.DIVCLK). The resulting divided clock is used for all logic within the Security Monitor Block.

A dedicated Random Number Generator (TRNG) generates the security patterns for the external sensors and die shield.

Figure 21-1. Security Monitor Block Diagram



21.1 Destructive Reset Source

The core of the security response is the Destructive Reset Source (DRS). A DRS can be triggered by any of the defined sources, including tamper detect inputs, environmental monitors or clearing the security lock bit.

DRS sources are:

- Die shield tamper
- Temperature out of range
- Battery out of range
- External tamper
- Software initiated by setting SECALM.DRS

DRS response initiates the following actions:

- It triggers Cortex-M3 NMI
- AES secure key register is cleared.
- The DRS source flag(s) is (are) set

- RTC second count is logged if the RTC is enabled

The DRS RTC logger mechanism creates a timestamp of the last DRS event occurrence, and to identify its source for diagnostic or tamper response purposes. Three battery backed registers are used to record such events, (SECALM, SECDIAG and DLRTC)

The DRS source is recorded as flags in the SECALM and SECDIAG registers. DLRTC contains the 32 bits snapshot of the RTC second counter when the last DRS event occurred. This may be zero if the RTC was not enabled.

The source flag in the SECALM register remains set until software clears the flag. The source flag in the SECDIAG register remains set until a BOR. The timestamp remains in the DLRTC register until the next DRS event or BOR.

There is no priority in the DRS sources and the order in which they are listed have no impact on operation of the DRS system. If multiple DRS sources happen simultaneously, or additional DRS sources are triggered while the device is in DRS reset, all applicable DRS source flags will be set. The last DRS timestamp is kept until a BOR. The device will be placed in DRS reset until the DRS source condition has cleared, or a minimum of 4 core clock cycles, whichever is longer.

21.2 Active Die Shield

The die shield monitors the top metal shield. This network covers the whole chip area. The die shield is enabled by setting the Die Shield Enable (INTSCN.SHIELD_EN) bit to 1.

When enabled, a Destructive NMI (non-maskable interrupt) will be triggered upon physical tampering attack to the chip.

21.3 Temperature Sensor

Device temperature is monitored by temperature sensor and is clocked by a 500 Hz clock.

If the device temperature is outside the allowable range, a DRS will be initiated. A DRS caused by the above temperature fault will set the corresponding flags in SECALM and SECDIAG registers. Once set, SECALM flags are cleared by software or a BOR, whereas SECDIAG flags are cleared by a BOR only.

Temperature sensor can be disabled for power savings by clearing the Temperature Sensor Enable bit (INTSCN.TEMP_EN). While the temperature low threshold is adjustable, the high temperature threshold is set at +125°C.

21.4 Battery Voltage Monitor

The device incorporates both a high battery voltage and a low battery voltage monitor.

The high battery voltage monitor will cause a DRS when the battery voltage attached to the device exceeds the datasheet threshold.

The low battery voltage monitor will cause a DRS when the battery voltage attached to the device is below the datasheet threshold.

A DRS caused by the above battery fault condition will set the corresponding flags in SECALM and SECDIAG registers. Once set, SECALM flags are cleared by software or a BOR, whereas SECDIAG flags are cleared by a BOR only.

Battery voltage monitor can be disabled by clearing the Battery Monitor Enable bit (INTSCN.VBAT_EN).



21.5 External Sensor

There are 6 pairs of External Sensors. Each sensor output uses the TRNG to generate a unique random digital signal which can be fed to an external mesh or contact point and then back to the corresponding sensor input. As long as the mesh remains intact (or the contact remains closed), the signal on the input will be filtered to remove transient glitches and compared to the output signal.

The sensor input sample frequency is 8kHz (default frequency). The rate of change of the sensor output is programmable from 2kHz to 31Hz with the default input clock. This is accomplished by dividing the 8kHz clock by a user defined value (DIVCLK, EXTFRQ) to produce a very slow local clock that is used to switch the sensor outputs. After changing sample frequency or sensor output frequency (DIVCLK, EXTFRQ), there may be a transient time up to about 100 clock cycles when the sensor is activated.

An error counter is used for glitch filtering purposes. At each mismatch between EXTS_x_IN and EXTS_x_OUT, the error counter is incremented. The counter is automatically reset each sensor bit period. If the counter exceeds its programmed threshold (EXTCNT), an alarm signal is raised, the status register is updated and a DRS is fired.

To determine EXTCNT value:

Sample Frequency,	$f_{SAMPLE} = 8\text{kHz} / \text{Clock Divide Factor (DIVCLK)}$
Sensor Output Frequency,	$f_{OUT} = f_{SAMPLE} / \text{Output Frequency Factor (EXTFRQ)}$
Glitch Filter Width (in second),	$GFW = (\text{EXTCNT}+1) * 1/f_{SAMPLE} \Rightarrow \text{EXTCNT} = GFW * f_{SAMPLE} - 1$

Note that $GFW < 1/f_{OUT}$, otherwise tamper will not be detected.

Therefore, $\text{EXTCNT} < (f_{SAMPLE} / f_{OUT}) - 1$

Example 1:

DIVCLK=000b
EXTFRQ=000b

$$\begin{aligned}f_{SAMPLE} &= 8\text{kHz}/1 = 8\text{kHz} \\f_{OUT} &= 8\text{kHz}/4 = 2\text{kHz}\end{aligned}$$

$$\begin{aligned}\text{EXTCNT} &< 8\text{kHz}/2\text{kHz}-1 = 3 \\ \text{Therefore, set EXTCNT from 0 to 2.} \\ \text{Maximum GFW} &< 1/2\text{kHz (0.5ms)} \\ \text{To filter a 0.2ms glitch} \\ \text{GFW} &= 0.2 \text{ ms} \\ \text{EXTCNT} &= 0.2 \text{ ms} * 8\text{kHz} - 1 = 0.6 \\ \text{Therefore , set EXTCNT to 0.}\end{aligned}$$

Example 2:

DIVCLK=001b
EXTFRQ=010b

$$\begin{aligned}f_{SAMPLE} &= 8\text{kHz}/2 = 4\text{kHz} \\f_{OUT} &= 4\text{kHz}/16 = 250\text{Hz}\end{aligned}$$

$$\begin{aligned}\text{EXTCNT} &< 4\text{kHz}/250\text{Hz} - 1 = 15. \\ \text{Therefore, set EXTCNT from 0 to 14.} \\ \text{Maximum GFW} &< 1/250\text{Hz (4ms)}\end{aligned}$$

$$\begin{aligned}\text{To filter a 1ms glitch} \\ \text{GFW} &= 1 \text{ ms}\end{aligned}$$

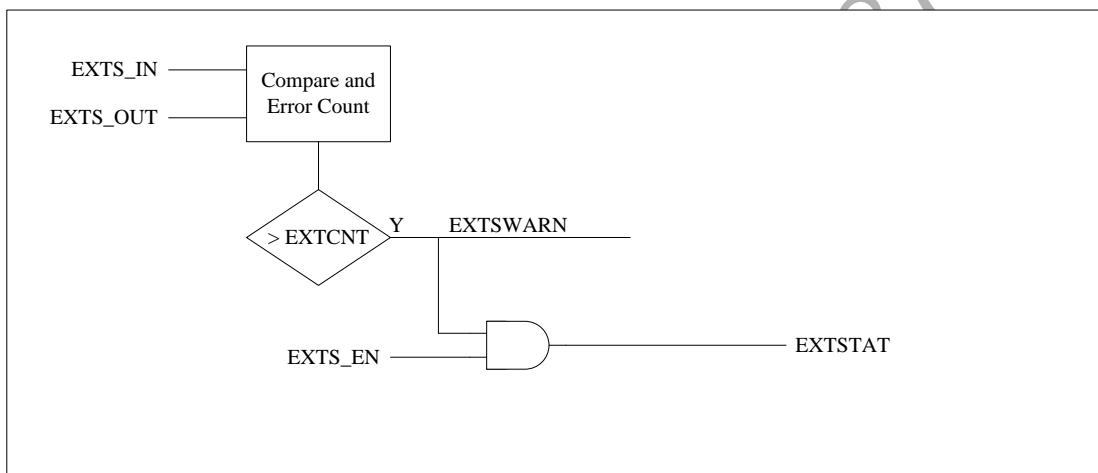
$\text{EXTCNT} = 1 \text{ ms} * 4\text{kHz} - 1 = 3$
 Therefore , set EXTCNT from 0 to 2

21.5.1 External Sensor Warning

On initial power up, EXTS_x_IN and EXTS_x_OUT may not necessarily be the same. To prevent enabling external sensor tamper detect prematurely, software should check the status of each pair before enabling it. To check the status, software can read the External Sensor Warning (SECALM.EXTSWARN). The EXTSWARN bit is set to 1 when the error count exceeds the programmed threshold. It will remain set until software clears it and the error count no longer exceed the programmed threshold. If EXTSWARN=1, then the sensor input and output has not settled to the same initial condition yet.

This is shown in Figure 21-2. For an enabled sensor, if the error count exceeds programmed threshold, the corresponding External Sensor Interrupt (EXTSI) and the External Sensor Tamper Detect (EXTSTAT) flags will be set to 1. When any of the EXTSTAT flags are set, a DRS will be generated, causing a non-maskable interrupt.

Figure 21-2. External Sensors Block Diagram



21.5.2 Power Fail Warning

The interrupt fires when any of the supplies are out of range. For location of the power fail warning interrupt vector, refer to Interrupt vector table at page 53. The power fail warning sources are :

- VCORE Over voltage
- VCORE Under voltage
- VDD Over voltage
- VDD Under voltage
- Voltage Glitch



21.6 Security Monitor Registers

Address assignments for the Security Monitor registers are outlined in Table 21-1.

Table 21-1. Security Monitor Register Address (Base ADDR = 0x4000 4000)

Offset	Access	Register	Description
0x0000	RW	EXTSCN	External Sensor Control Register
0x0004	RW	INTSCN	Internal Sensor Control Register
0x0008	RW	SECALM	Security Alarm Register
0x000C	R	SECDIAG	Security Diagnostic Register
0x0010	R	DLRTC	DRS Log RTC Register
0x0014	W	RFU	Reserved
0x0018	W	RFU	Reserved
0x001C	W	RFU	Reserved
0x0020	W	RFU	Reserved
0x0034	R	SECST	Security Monitor Status Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



21.6.1 External Sensor Control Register (EXTSCN, Offset 0x0000)

This register is battery backed. This register can only be written to when LOCK=0.

Table 21-2. EXTSCN (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	LOCK	BUSY	RFU[2:0]					DIVCLK[2:0]
BOR	0	0	000					000
Other Reset	N/A	N/A	000					N/A
Access	RS	R	R					RW

Position	23	22	21	20	19	18	17	16
Field	EXTFRQ[2:0]					EXTCNT[4:0]		
BOR	000					0 0000		
Other Reset	N/A					N/A		
Access	RW					RW		

Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
BOR	0000 0000							
Other Reset	0000 0000							
Access	R							

Position	7	6	5	4	3	2	1	0
Field	RFU	RFU	EXTS_EN[5:0]					
BOR	0	0	00 0000					
Other Reset	0	0	N/A					
Access	R	R	RW					

Name	Bits	Description	Settings
EXTS_EN	5:0	External Sensor Enable. Each bit enables a corresponding external sensor input/output pair.	0: External sensor disabled 1: External sensor enabled
RFU	15:6	Reserved	N/A
EXTCNT	20:16	External Sensor Error Counter. These bits set the number of external sensor accepted mismatches that have to occur within a single bit period before an external sensor alarm is triggered.	Number of accepted mismatches = EXTCNT
EXTFRQ	23:21	External Sensor Frequency. These bits define the frequency at which the external sensors are clocked to/from the EXTS_IN and EXTS_OUT pair.	000: /4 (2000Hz) 001: /8 (1000 Hz) 010: /16 (500 Hz) 011: /32 (250 Hz) 100: /64 (125 Hz) 101: /128 (63 Hz) 110: /256 (31 Hz) 111: Reserved * The above table assumes an undivided 8KHz reference clock.

Name	Bits	Description	Settings
DIVCLK	26:24	Clock Divide. These bits are used to divide the 8KHz input clock. The resulting divided clock is used for all logic within the Security Monitor Block. Note: If the input clock is divided with these bits, the error count threshold table and output frequency will be affected accordingly with the same divide factor.	000: /1 (8000 Hz) 001: /2 (4000 Hz) 010: /4 (2000 Hz) 011: /8 (1000 Hz) 100: /16 (500 Hz) 101: /32 (250 Hz) 110: /64(125 Hz) 111: Reserved
RFU	29:27	Reserved	N/A
BUSY	30	Busy. This bit is set to 1 by hardware after EXTSCN register is written to. This bit is automatically cleared to 0 after this register information has been transferred to the security monitor domain.	0: Idle 1: Update in progress
LOCK	31	Lock Register. Once locked, the EXTSCN register can no longer be modified. Only a battery disconnect will clear this bit. VBAT powers this register.	0: Not locked. Writes to this register allowed. 1: Locked. Writes to this register ignored.



21.6.2 Internal Sensor Control Register (INTSCN, Offset 0x0004)

This register is battery backed. This register can only be written to when LOCK=0.

Table 21-3. INTSCN (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	LOCK	RFU[6:0]						
BOR	0	000 0000						
Other Reset	N/A	000 0000						
Access	RS	R						

Position	23	22	21	20	19	18	17	16
Field	RFU	VGLEN	VDDHIEN	VDDLOEN	VCOREHIEN	VCORELOEN	RFU	LOTEMP_SEL
BOR	0	0	0	0	0	0	0	0
Other Reset	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Access	R	RW	RW	RW	RW	RW	R	RW

Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
BOR	0000 0000							
Other Reset	0000 0000							
Access	R							

Position	7	6	5	4	3	2	1	0
Field	RFU[4:0]					VBAT_EN	TEMP_EN	SHIELD_EN
BOR	0 0000					0	0	0
Other Reset	0 0000					N/A	N/A	N/A
Access	R					RW	RW	RW

Name	Bits	Description	Settings
SHIELD_EN	0	Die Shield Enable.	0: Die shield sensors powered off 1: Die shield sensors enabled
TEMP_EN	1	Temperature Sensor Enable.	0: Temperature sensor powered off 1: Temperature sensor enabled
VBAT_EN	2	Battery Monitor Enable.	0: Battery voltage monitor powered off 1: Battery voltage monitor enabled
RFU	15:3	Reserved.	N/A
LOTEMP_SEL	16	Low Temperature Detection Select.	0: -50°C 1: -30°C
RFU	17	Reserved.	N/A
VCORELOEN	18	VCORE Undervoltage Detect Enable	0: VCORE Low Warning disabled 1: VCORE Low Warning enabled
VCOREHIEN	19	VCORE Overvoltage Detect Enable	0: VCORE High Warning disabled 1: VCORE High Warning enabled
VDDLOEN	20	VDD Undervoltage Detect Enable	0: VDD Low Warning disabled 1: VDD Low Warning enabled
VDDHIEN	21	VDD Overvoltage Detect Enable	0: VDD High Warning disabled 1: VDD High Warning enabled
VGLEN	22	Voltage Glitch Detection Enable	0: Voltage glitch detection disabled 1: Voltage glitch detection enabled



Name	Bits	Description	Settings
RFU	30:23	Reserved	N/A
LOCK	31	Lock Register. Once locked, the INTSCN register can no longer be modified. Only a battery disconnect will clear this bit. VBAT powers this register.	0: Not locked. Writes to this register allowed. 1: Locked. Writes to this register ignored.

E/UG25H05/36488764



21.6.3 Security Alarm Register (SECALM, Offset 0x0008)

This register is battery backed. This register bits are set by hardware on detection of a DRS and can be cleared by software.

Table 21-4. SECALM (Offset 0x0008)

Position	31	30	29	28	27	26	25	24
Field	RFU[1:0]				EXTSWARN[5:0]			
BOR	00				00 0000			
Other Reset	N/A				N/A			
Access	R				RC			

Position	23	22	21	20	19	18	17	16
Field	RFU[1:0]				EXTSTAT[5:0]			
BOR	00				00 0000			
Other Reset	N/A				N/A			
Access	R				RC			

Position	15	14	13	12	11	10	9	8
Field	RFU[2:0]		VGL	VDDHI	VCOREHI	VCORELO	VDDLO	
BOR	000		0	0	0	0	0	0
Other Reset	N/A		N/A	N/A	N/A	N/A	N/A	N/A
Access	R		RC	RC	RC	RC	RC	RC

Position	7	6	5	4	3	2	1	0
Field	EXTF	BATHI	BATLO	HITEMP	LOTEMP	SHIELDF	KEYWIPE	DRS
BOR	0	0	0	0	0	0	0	0
Other Reset	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
Access	RC	RC	RC	RC	RC	RC	RW	RW

Name	Bits	Description	Settings
DRS	0	Destructive Reset Trigger. Setting this bit will generate a DRS. This bit is self-cleared by hardware.	0: No effect. 1: Set to 1 to initiate a DRS.
KEYWIPE	1	Key Wipe Trigger. Set to 1 to initiate a wipe of the AES key register. It does not reset the part, or log a timestamp. AES and DES registers are not affected by this bit. This bit is automatically cleared to 0 after the keys have been wiped	0: No effect. 1: Initiate key wipe
SHIELDF	2	Die Shield Flag.	0: No fault detected. 1: Die shield fault detected.
LOTEMP	3	Low Temperature Detect.	0: No fault detected. 1: Device temperature below low temperature threshold detected.
HITEMP	4	High Temperature Detect.	0: No fault detected. 1: Device temperature above high temperature threshold detected.
BATLO	5	Battery Undervoltage Detect	0: No fault detected. 1: Battery under voltage detected.
BATHI	6	Battery Overvoltage Detect	0: No fault detected. 1: Battery over voltage detected.

Name	Bits	Description	Settings
EXTF	7	External Sensor Flag. This bit is set to 1 when any of the EXTSTAT bits are set.	0: No fault detected. 1: External tamper detected.
VDDLO	8	VDD Undervoltage Detect Flag	0: No fault detected. 1: VDD undervoltage detected.
VCORELO	9	VCORE Undervoltage Detect Flag	0: No fault detected. 1: VCORE undervoltage detected.
VCOREHI	10	VCORE Overvoltage Detect Flag	0: No fault detected. 1: VCORE overvoltage detected.
VDDHI	11	VDD Overvoltage Flag	0: No fault detected. 1: VDD overvoltage detected.
VGL	12	Voltage Glitch Detection Flag	0: No fault detected. 1: Voltage glitch detected.
RFU	15:13	Reserved	N/A
EXTSTAT	21:16	External Sensor Detect. Each bit corresponds to one external sensor input/output pair. The tamper detect is only active when it is enabled. These bits need to be cleared in software after a tamper event to re-arm the sensor(s).	0: No fault detected. 1: External tamper detected.
RFU	23:22	Reserved	N/A
EXTSWARN	29:24	External Sensor Warning Ready flag. Each bit corresponds to one external sensor input/output pair. The tamper detect warning flags are set, regardless of whether the external sensors are enabled.	0: No fault detected 1: External tamper warning.
RFU	31:30	Reserved	N/A



21.6.4 Security Diagnostic Register (SECDIAG, Offset 0x000C)

This register is battery backed. This register bits are set by hardware on detection of a DRS and can be cleared by BOR only.

Table 21-5. SECDIAG (Offset 0x000C)

Position	31	30	29	28	27	26	25	24
Field					RFU[7:0]			
BOR					0000 0000			
Other Reset					0000 0000			
Access					R			

Position	23	22	21	20	19	18	17	16
Field		RFU[1:0]				EXTSTAT[5:0]		
BOR		00				00 0000		
Other Reset		00				N/A		
Access		R				R		

Position	15	14	13	12	11	10	9	8
Field				RFU[6:0]				AESKT
BOR				000 0000				0
Other Reset				000 0000				N/A
Access				R				R

Position	7	6	5	4	3	2	1	0
Field	DYNF	BATHI	BATLO	HITEMP	LOTEMP	SHIELDF	RFU	BORF*
BOR	0	0	0	0	0	0	0	1
Other Reset	N/A	N/A	N/A	N/A	N/A	N/A	0	N/A
Access	R	R	R	R	R	R	R	R

Name	Bits	Description	Settings
BORF*	0	Battery-On-Reset Flag. This bit is set once the back up battery is connected. * <i>This bit does not exist on A3 silicon and before.</i>	0: BORF has not occurred 1: BORF has occurred
RFU	1	Reserved	N/A
SHIELDF	2	Die Shield Flag.	0: No fault detected. 1: Die shield fault detected.
LOTEMP	3	Low Temperature Detect	0: No fault detected. 1: Device temperature below low temperature threshold detected.
HITEMP	4	High Temperature Detect	0: No fault detected. 1: Device temperature above high temperature threshold detected.
BATLO	5	Battery Undervoltage Detect	0: No fault detected. 1: Battery under voltage detected.
BATHI	6	Battery Overvoltage Detect	0: No fault detected. 1: Battery over voltage detected.
DYNF	7	Dynamic Sensor Flag. This bit is set to 1 when any of the EXTSTAT bits are set.	0: No fault detected. 1: External tamper detected.
AESKT	8	AES Key Transfer. This bit is set to 1 when AES Key has been transferred from the TRNG to the battery backed AES key register. This bit can be reset by a BOR.	0: Key has not been transferred or has been erased by BOR 1: Key has been transferred.



Name	Bits	Description	Settings
RFU	15:9	Reserved	N/A
EXTSTAT	21:16	External Sensor Detect. Each bit corresponds to one external sensor input/output pair.	0: No fault detected. 1: External tamper detected.
RFU	31:22	Reserved	N/A

E/UG25H05/36488764



21.6.5 DRS Log RTC Register (DLRTC, Offset 0x0010)

This register is battery backed.

Table 21-6. DLRTC (Offset 0x0010)

Position	31	30	29	28	27	26	25	24
Field	DLRTC[31:24]							
BOR	0000 0000							
Other Reset	N/A							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	DLRTC[23:16]							
BOR	0000 0000							
Other Reset	N/A							
Access	R							

Position	15	14	13	12	11	10	9	8
Field	DLRTC[15:8]							
BOR	0000 0000							
Other Reset	N/A							
Access	R							

Position	7	6	5	4	3	2	1	0
Field	DLRTC[7:0]							
BOR	0000 0000							
Other Reset	N/A							
Access	R							

Name	Bits	Description	Settings
DLRTC	31:0	DRS Log RTC Value. This register contains the 32 bit value in the RTC second register when the last DRS event occurred.	



21.6.6 Security Monitor Status Register (SECST, Offset 0x0034)

The register reflects the availability of security monitor registers. When the corresponding bit is 0, the associated register has been updated and access (read/write operation) is authorized. When the corresponding bit is 1, the associated register is being updated and access is not authorized. This register should be polled after each write to a security monitor register to avoid reading back the previous value.

Table 21-7. SECST (Offset 0x0034)

Position	31	30	29	28	27	26	25	24
Field				RFU[7:0]				
Reset				0000 0000				
Access				R				
Position	23	22	21	20	19	18	17	16
Field				RFU[7:0]				
Reset				0000 0000				
Access				R				
Position	15	14	13	12	11	10	9	8
Field				RFU[7:0]				
Reset				0000 0000				
Access				R				
Position	7	6	5	4	3	2	1	0
Field			RFU[4:0]			SECALRS	INTSRS	EXTSRS
Reset			0000 0			0	0	0
Access			R			R	R	R
Name	Bits	Description					Settings	
EXTSRS	0	External Sensor Control Register Status					0:Access authorized 1:Access not authorized.	
INTSRS	1	Internal Sensor Control Register Status					0:Access authorized 1:Access not authorized.	
SECALRS	2	Security Alarm Register Status					0:Access authorized 1:Access not authorized.	
RFU	31:3	Reserved					N/A	



22.0 Serial Peripheral Interface (SPI)

The SPI module provides an independent serial communication channel to communicate synchronously with peripheral devices in a multiple master or multiple slave system. The interface is a four-wire full-duplex serial bus that can be operated in either master mode or slave mode.

This device provides three instances of the SPI module.

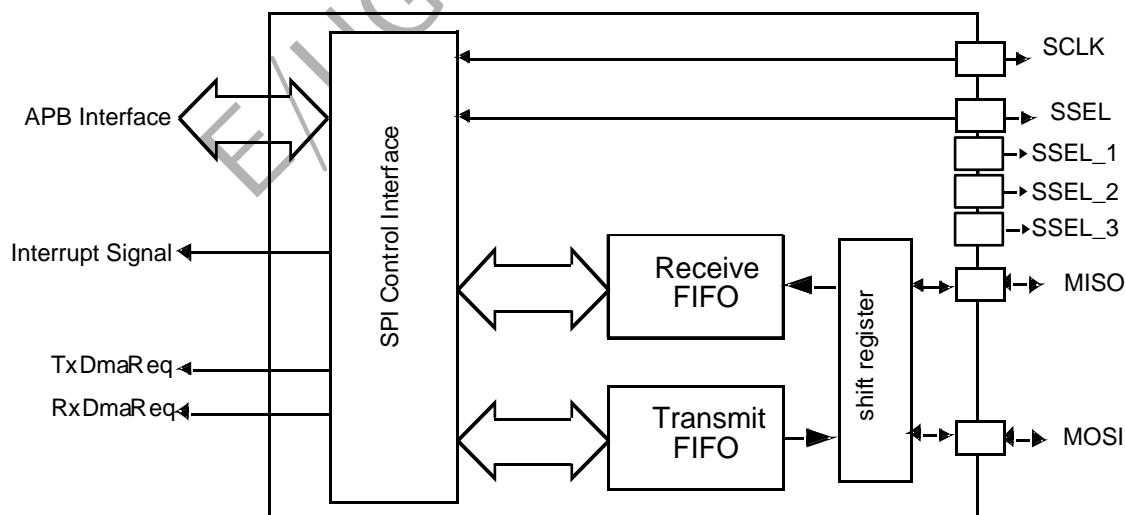
SPI-compatible devices include EEPROMs, printer controllers and contactless smart card controllers. An Inter-IC Sound (I²S) protocol mode supports 16-bit mono or stereo audio transfer to/from an external audio codec.

Feature of the SPI include:

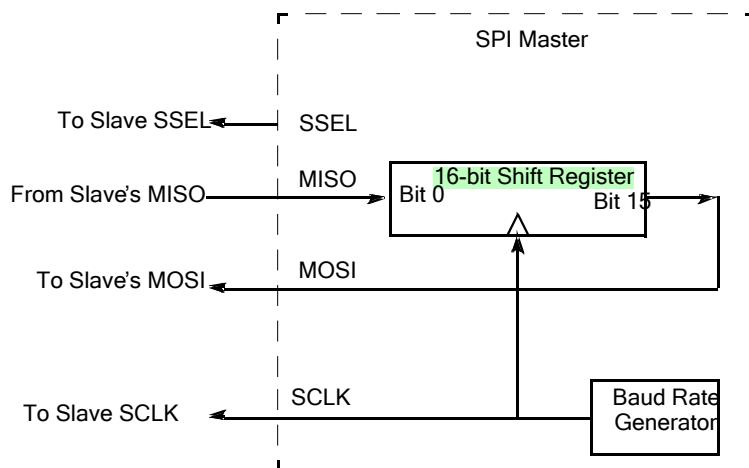
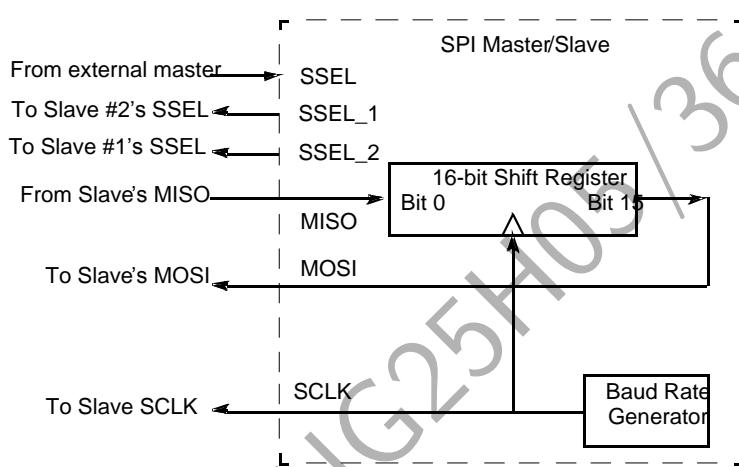
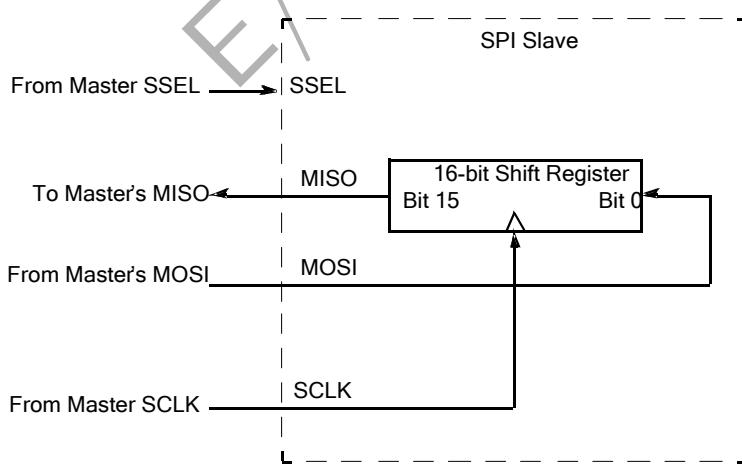
- Full-duplex, synchronous communication of 1 to 16-bit characters
- Four-wire interface
- Data transfers rates up to one-fourth the peripheral clock (PCLK) frequency
- Master, multi-master and slave modes of operation
- Dedicated Baud Rate Generator
- 8 x 16 Transmit and Receive FIFOs
- Transmit and Receive DMA Support
- I²S mode for 16-bit audio transfer, master and slave modes
- Up to 3 additional Slave Select Pins in master mode.

The block diagram shows the SPI external interface signals, control unit, receive and transmit FIFOs, and single shift register common to the transmit and receive data path. Each time that an SPI transfer completes, the received character is transferred to the receive FIFO.

Figure 22-1. SPI Block Diagram



The SPI may be configured as either a master (in single or multi-master systems) or a slave. An SPI system has a single master and one or more slaves for any given transaction. To configure the SPI block as master (slave), set (clear) the MMEN bit. For a multi-master system the WOR bit must also be set.

**Figure 22-2. SPI Single-Master, Single-Slave****Figure 22-3. SPI Multi-Master, Multi-Slave****Figure 22-4. SPI Slave**



22.1 Operation

The SPI is a full-duplex, synchronous, character-oriented channel that supports a four-wire interface (serial clock, transmit data, receive data and slave select). The SPI block consists of a transmit/receive shift register (supported by FIFOs), a Baud Rate Generator and a control unit.

During an SPI transfer, data is sent and received simultaneously by both master and slave devices. When an SPI transfer occurs, a multi-bit (selectable from 1 to 16-bit) character is shifted out on one data pin and a multi-bit character is simultaneously shifted in on a second data pin. A 16-bit shift register in the master and another 16-bit shift register in the slave are connected as a circular buffer with the most significant bit (bit15) sent first. The SPI contains two 8x16 FIFOs to support transmit and receive directions. New data is moved automatically from the transmit FIFO into the shift register at the start of every new SPI transfer as long as there is data in the transmit FIFO. At the end of every SPI transfer, data is moved from the shift register into the receive FIFO.

22.2 SPI Signals

The SPI signals are:

- MISO (Master-In, Slave-Out)
- MOSI (Master-Out, Slave-In)
- SCLK (SPI Serial Clock)
- SSEL (Slave Select)
- SSEL_1, SSEL_2, SSEL_3 (additional slave selects in Master Mode)

These signals may be pinned out through GPIO pins as alternate functions. Refer to the GPIO chapter for information on selecting the SPI mode I/O. An external pull-up resistor should be used to prevent floating input signals when operating the SPI signals in open drain mode (refer to the WOR bit) or high impedance mode (slave MISO is in high impedance mode when the slave is not selected).

22.2.1 Master-In, Slave-Out

The MISO pin is configured as an input in master mode and as an output in slave mode. It is one of two lines that transfer serial data, with the most significant bit sent first. The MISO pin of a slave device is placed in a high-impedance state if the slave is not selected. When the SPI channel is not enabled (SPIEN bit = 0), this signal is in a high-impedance state.

22.2.2 Master-Out, Slave-In

The MOSI pin is configured as an output in master mode and as an input in slave mode. It is one of two lines that transfer serial data, with the most significant bit sent first. When the SPI channel is not enabled, this signal is in a high-impedance state.

22.2.3 Serial Clock

The Serial Clock (SCLK) synchronizes data movement in and out of the device through the MOSI and MISO pins. In master mode, the SPI's Baud Rate Generator creates SCLK. The master drives the serial clock out its SCLK pin to the slave's SCLK pin. When the SPI is configured as a slave, the SCLK pin is an input from the master. Slave devices ignore the SCLK signal, unless their SSEL pin is asserted. When configured as a slave, the minimum SCLK period is 8 times the peripheral clock (PCLK) period. For example if the APB clock (PCLK) is running at 20 MHz in the slave SPI, the master SPI SCLK must be set at a maximum of 2.5 MHz.

The master and slave are each capable of exchanging a character of data during a sequence of NUMBITS clock cycles (refer to NUMBITS field). In both master and slave devices, data is shifted on one edge of the SCLK and is sampled on the opposite edge where data is stable. Edge polarity is determined by the SPI phase and polarity control.



22.2.4 Slave Select

The Slave Select (SSEL) signal is used to select a specific slave device during SPI transfers or to distinguish left and right channel audio data in I²S mode. In an SPI system with multiple slaves, the master must provide separate SSEL signals to each slave. SSEL must be low prior to all data communication to and from the slave device. SSEL must stay low for the full duration of each character transfer. The SSEL signal may stay low during the transfer of multiple characters or may de-assert between each character. User should not toggle SSEL between words. Though the SSEL signal typically is active low, either polarity can be supported via the SSV bit.

Single Master SPI System – When configured as the only master in an SPI system, the SSEL pin is configured as an output by setting SSIO = 1. The polarity of SSEL is selected via the SSV bit. Other GPIO output pins must be employed to select additional SPI slave devices.

Multi-Master SPI System – When configured as one master in a multi-master SPI system, the SSEL pin is configured as an input by clearing SSIO = 0. SSEL_1, SSEL_2, and SSEL_3 signals are employed to select SPI slave devices via bits SSL1, SSL2, and SSL3 respectively. When acting as the master, the SSEL input signal should be high. If the SSEL input signal goes low (indicating another master is selecting this device as an SPI slave) the Collision error flag is set. The SPI block can be switched between master and slave modes when operating in a multi-master system via the MMEN bit.

Slave SPI System – When configured as a slave in an SPI system, the SSEL pin is configured as an input by clearing SSIO = 0.

I²S System – In I²S mode the SSEL output is controlled by hardware and distinguishes left and right channel audio data. When operating as the I²S master, the SCLK and SSEL signals are outputs. When operating as the I²S slave, the SCLK and SSEL signals are inputs. This SSEL signal is referred to as word select signal (WS) in the I²S protocol. Normally the WS signal transitions one SCLK period before the MSB of the audio data word, however if the I2S_LJ bit is set, the audio data word is “left justified” to be in phase with the WS signal.

22.3 SPI Clock Phase and Polarity Control

The SPI supports four combinations of SCLK phase and polarity:

- Clock Polarity (CLKPOL) selects an active low/high clock and has no effect on the transfer format
- Clock Phase (PHASE) selects one of two fundamentally different transfer formats

For proper data transmission, the clock phase and polarity must be identical for the SPI master and slave. The master always places data on the MOSI line a half-cycle before the SCLK edge in order for the slave to latch the data.

Table 22-1. PHASE and CLKPOL Operation

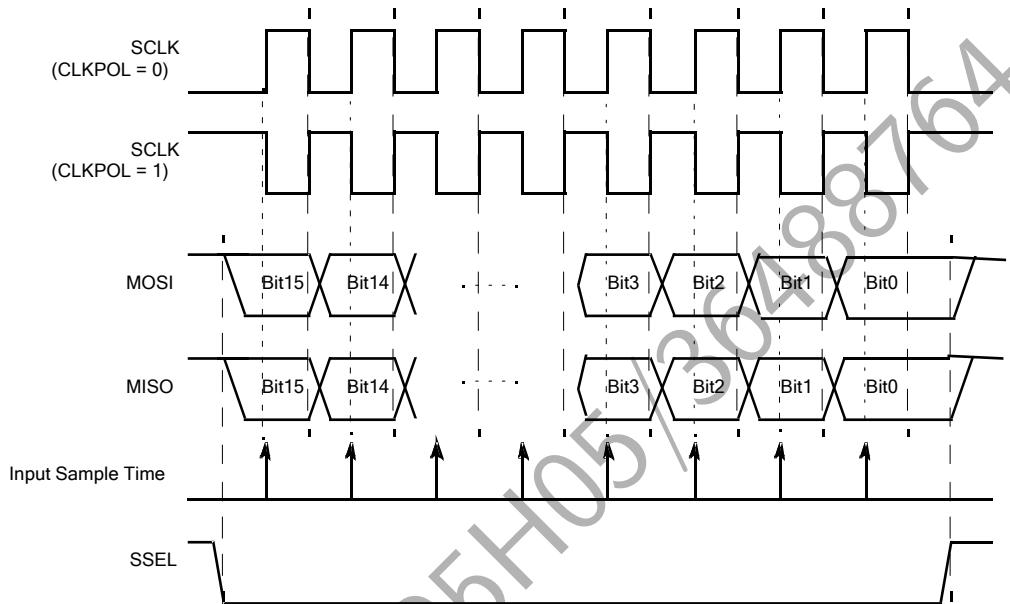
PHASE	CLKPOL	SCLK Transmit Edge	SCLK Receive Edge	SCLK Idle State
0	0	Falling	Rising	Low
0	1	Rising	Falling	High
1	0	Rising	Falling	Low
1	1	Falling	Rising	High

22.3.1 Transfer Format (PHASE = “0”)

Figure 22-5 is the timing diagram for an SPI 16-bit transfer in which the clock phase is cleared (PHASE = 0). The two SCLK waveforms show active low (CLKPOL = 0) and active high (CLKPOL = 1). The diagram may be interpreted as either a master or slave timing diagram since the SCLK, MISO and MOSI pins are directly connected between the master and the slave.

In the case of multi-character transfers with SSEL remaining asserted between characters, the output data will change at the end of the Bit0 (final clock edge) to reflect the output value for Bit15 of the next character.

Figure 22-5. SPI Timing (PHASE = “0”)



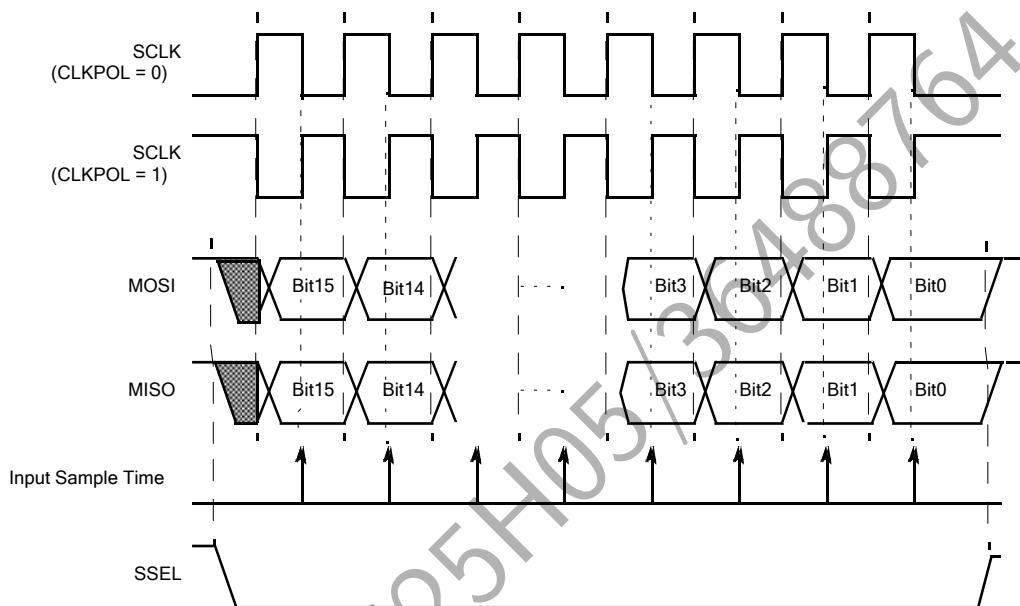


22.3.2 Transfer Format (PHASE = “1”)

Figure 22-6 is the timing diagram for an SPI transfer in which the clock phase is set (PHASE = 1). The two SCLK waveforms show active low (CLKPOL = 0) and active high (CLKPOL = 1). The diagram may be interpreted as either a master or slave timing diagram since the SCLK, MISO and MOSI pins are directly connected between the master and the slave.

In the case of multi-character transfers with SSEL remaining asserted between characters, the Bit0 output data will remain stable until the clock edge which starts Bit15 of the next character or until SSEL deasserts at the end of the transfer.

Figure 22-6. SPI Timing (PHASE = “1”)



22.4 Data Movement

Data movement can be controlled in one of the following ways:

Software polling the TXST bit (transfer one word at a time) or polling the TX_FIFO_LEVEL/RX_FIFO_LEVEL fields (can transfer up to 8 characters at a time).

The IRQE bit can be set to enable data and error interrupts. The STR bit may be used if desired to force a “startup” data interrupt. A data interrupt will be generated on completion of each character transfer.

DMA control of data transferred is enabled via the RX_DMA_EN and/or TX_DMA_EN bits. The TX_FIFO_LEVEL and RX_FIFO_LEVEL control when DMA requests are asserted. When DMA is enabled, data interrupts are disabled (error interrupts will still occur). DMA operation is beneficial for block transfers as the CPU only needs to service one DMA interrupt per block of data versus one interrupt for each character transferred if data interrupt based transfer is used.

The SPI Data register is used for transferring data in both incoming and outgoing directions.

For incoming data, the receive data is shifted into an internal shift register. Once a full character has been shifted in, the character is automatically moved into the Receive FIFO. The Receive FIFO data is read through the SPI Data Register.

For outgoing data, the transmit data written to the SPI Data Register is written into the Transmit FIFO. When the shift register is empty, data is automatically moved into the shift register from the Transmit FIFO.



Note: When SPIEN is cleared to 0, any data written to the SPI Data Register is not discarded. When SPIEN is set to 1, data transmission will begin with data that had been written to the FIFO or left in the FIFO when SPIEN = 0. Data in the transmit FIFO can be flushed at any time by writing a 1 to the TX_FIFO_CLR bit in the SPI_DMA register.

With the SPI configured as a master, writing data to this register initiates the data transmission. With the SPI configured as a slave, writing data to this register loads the shift register in preparation for the next data transfer with the external master. In either the master or slave mode, when the transmit FIFO is full, writes to this register are ignored and the Transmit Overrun error flag, TOVR, is set in the SPI Status register.

Data is shifted out starting with bit 15. The last bit received will reside in bit position 0. When the character length is less than 16 bits (as set by the NUMBITS field in the SPI Mode register), the transmit character must be left justified in the SPI Data register. A received character of less than 16 bits will be right justified (last bit received will be in bit position 0). For example, if the SPI is configured for 4-bit characters, the transmit characters must be written to SPI_DATA[15:12] and the received characters are read from SPI_DATA[3:0].

The software overhead to left justify the transmit data can be eliminated by setting the TX_LJ bit in the SPI_MODE register. When TX_LJ = 1, transmit data is always written by software or DMA to SPI_DATA in right justified form and hardware performs the left justify according to NUMBITS when the shift register is loaded. For the 4-bit character example, when TX_LJ = 1, transmit data is written to SPI_DATA[3:0] and hardware shifts these to bits [15:12] when the shift register is loaded. The TX_LJ bit has no effect on receive data which is always right justified.

22.5 Configuration for Master, Slave and Multi-Master Modes

22.5.1 Single Master Operation

The SPI block is configured as the master in a single master SPI configuration by the following:

```
SPIEN = 1
MMEN = 1
WOR = 0
SSIO = 1
```

The PHASE and CLKPOL bits and the NUMBITS field must be consistent with the slave SPI devices. The SSV bit asserts/deasserts the SSEL output pin (SSEL_1, SSEL_2, and SSEL_3 pins may be used to drive slave selects to multiple slaves). The SPI Baud Rate Register must be initialized.

22.5.2 Multi-Master Operation

The SPI block is configured for master/slave operation in a multi-master SPI configuration by setting:

```
SPIEN = 1
MMEN = 1 or 0. Software controls the master/slave mode dynamically via some bus arbitration algorithm to allow multiple masters to communicate to slave and master/slave devices.
SSIO = 0
WOR = 1 open-drain mode must be enabled to prevent bus contention since all SCLK, MOSI and MISO pins are tied together.
```

At any time, only one SPI device can be configured as the master and all other SPI devices on the bus must be configured as slaves. The master selects a single slave by asserting the SSEL pin to that slave only. Then the master drives data out its SCLK and MOSI pins to all of the slaves' SCLK and MOSI pins (including those which are not selected). The selected slave drives data out its MISO pin to the master's MISO pin. When configured as a master operating in a multi-master system, if the SSEL pin is configured as an input and is driven low by another master, a multi-master collision fault is signaled by COL = 1.

22.5.3 Slave Operation

The SPI block is configured for slave mode operation by setting:



SPIEN = 1
 MMEN = 0
 SSIO = 0
 WOR = 0

The PHASE and CLKPOL bits and the NUMBITS field must be set to be consistent with the other SPI devices. The STR bit may be used if desired to force a “startup” interrupt. The BIRQ bit and the SSV bit are not used in slave mode. The SPI baud rate generator is not used in slave mode, so the SPI Mode Register need not be initialized.

If the slave has data to send to the master, the data should be written before the transaction starts (first edge of SCLK when SSEL is asserted). If the SPI Data Register is not written prior to the slave transaction (the Transmit FIFO is empty), the MISO pin will output whatever value was written last into the SPI Data Register.

Due to the delay resulting from synchronization of the SPI input signals to PCLK, the maximum SCLK baud rate that can be supported in slave mode is the PCLK frequency divided by 8. This rate is controlled by the SPI master.

22.6 I²S (Inter-IC Sound) Mode

The SPI block is configured for I²S mode operation by setting:

I2S_EN = 1
 SPIEN = 1
 PHASE = 0
 CLKPOL = 0
 NUMBITS = 0 (to select 16-bit characters)

The MMEN and SSIO bits are set in accordance with either master or slave mode of operation. The SSV bit is ignored by hardware in I²S mode. In I²S, the master hardware sources SSEL (known as WS in I²S protocol) and SCLK. In this mode SSEL alternates between consecutive audio words. SSEL=0 indicates left channel data and SSEL=1 indicates right channel audio data.

The receive and/or transmit DMA channels must be enabled when operating in I²S mode. Typically audio data will only flow in one direction as defined by the RX_DMA_EN or TX_DMA_EN bits, however audio data may be transferred in both directions simultaneously if desired. Data in the transmit buffer should be initialized with the first 16-bit character containing a left channel audio sample, then alternating right and left channel 16-bit audio samples. When audio data is being received, the first sample written into the receive buffer will be a left channel audio sample.

22.6.1 Mute

The I2S_MUTE bit in the I²S Control Register can be set by software asynchronously to the DMA transfers to silence the transmit output. At the beginning of the next left channel audio sample after I2S_MUTE is asserted, DMA and FIFO accesses will continue, however, the data read from the transmit FIFO will be discarded and replaced with zeroes. When I2S_MUTE is deasserted, the transmit output will resume at the beginning of the next left channel audio sample.

22.6.2 Pause

The I2S_PAUSE bit can be set by software asynchronously to the DMA transfers to halt DMA and FIFO accesses. At the beginning of the next left channel audio sample after I2S_PAUSE is asserted, both transmit and receive DMA and FIFO accesses will halt and the transmit data will be forced to zero. At the beginning of the next left channel audio sample after I2S_PAUSE is deasserted, the DMA accesses will resume from where they were halted. Pause takes precedence over mute.

22.6.3 Mono

The I2S_MONO bit in the I²S Control Register (0x01C) is set to select single channel audio data vs. stereo format. In mono mode each transmit data word read from the transmit FIFO is duplicated for both left and right channel output words.. The receive channel will read the data from the left channel (SSEL = 0) and ignore data in the right channel. This allows DMA buffers for mono mode to be one-half the size of DMA buffers for stereo mode.



22.6.4 Left Justify

The I²S_LJ bit selects the phase of the SSEL signal versus the data. When I²S_LJ = 0 (normal I²S mode), the audio data lags the SSEL signal by one SCLK period. When I²S_LJ = 1, the audio data is “left justified” so that it is in sync with the SSEL signal.

Figure 22-7. I²S Mode (I²S_EN=1, I²S_LJ=0).

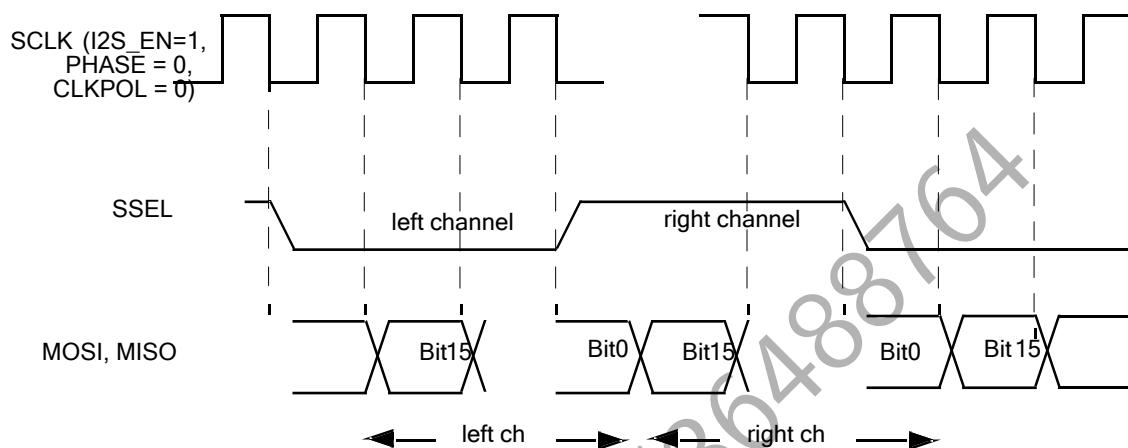
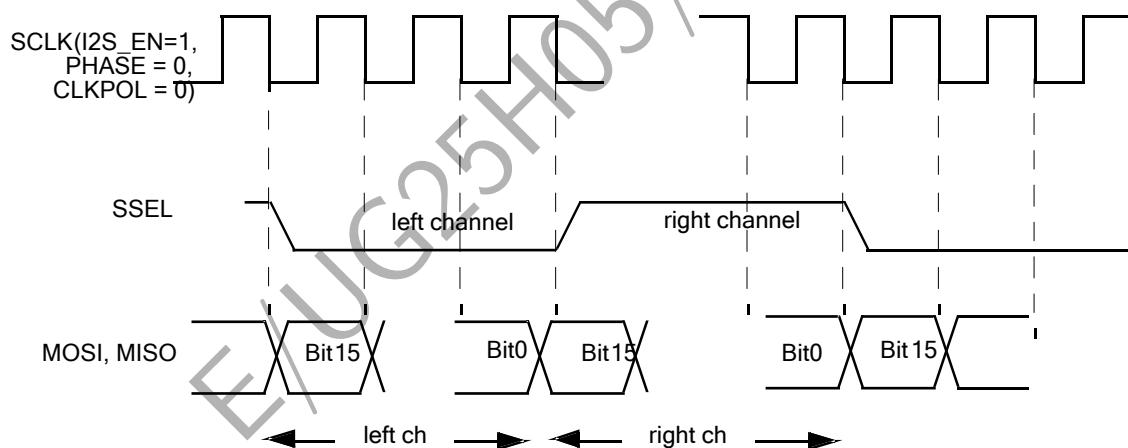


Figure 22-8. I²S Mode (I²S_EN=1, I²S_LJ=1).



22.7 Error Detection

The SPI contains error detection logic to support SPI communication protocols and recognize when communication errors have occurred. If the IRQE bit is set, error conditions will generate an interrupt. The SPI Status Register indicates which error has been detected.

22.7.1 Transmit Overrun

A transmit overrun error indicates a write to the FIFO was attempted when the internal transmit FIFO was full in either master or slave mode. An overrun sets the TOVR bit to 1. Writing a 1 to TOVR clears this error flag.

A transmit FIFO overrun in I²S mode may result in mixing left and right channel data. Software should reinitialize the DMA channel and data buffer and restart the I²S transfer.



22.7.2 Mode Fault (Multi-Master Collision)

A mode fault indicates more than one master is trying to communicate at the same time (a multi-master collision). The mode fault is detected when an enabled master's SSEL input pin is asserted low. A mode fault sets the COL bit to 1. Writing a 1 to COL clears this error flag.

This error interrupt will not occur in I²S mode.

22.7.3 Slave Mode Abort

A slave mode abort indicate that the SSEL pin deasserted before all bits in a character were transferred (while operating in slave mode). The next time SSEL asserts, the MISO pin will output SPI_DATA[15], regardless of where the previous transaction left off. A slave mode abort sets the ABT bit to 1. Writing a 1 to ABT clears this error flag.

This error interrupt will not occur in I²S mode.

22.7.4 Receive Overrun

A receive overrun error indicates a write to the receive FIFO occurred when the internal receive FIFO was full (in either master or slave mode). An overrun sets ROVR = 1. Writing a 1 to ROVR clears this error flag.

A receive FIFO overrun in I²S mode may result in mixing left and right channel data. Software should reinitialize the DMA channel and data buffer and restart the I²S transfer.

22.8 SPI Interrupts

When the SPI interrupt is enabled (IRQE bit = 1 in the SPI Control Register [0x004]), the SPI generates an interrupt when one of the following interrupt conditions occur. The interrupt condition is indicated by the IRQ bit in the SPI Status Register (0x008). Writing a 1 to the IRQ bit clears the pending SPI interrupt request.

22.9 Data Interrupt

A data interrupt occurs when the transmit character has been fully moved out of the shift register AND the Transmit FIFO is empty (in either master or slave mode). Since transmit and receive are always interlocked, there is no need for a separate receive interrupt. If either transmit or receive DMA is enabled via the RX_DMA_EN and TX_DMA_EN bits, the data interrupt will not occur, however error interrupts are still enabled when using DMA. A data interrupt is indicated by IRQ = 1 and no error interrupt bits set.

22.9.1 Forced Interrupt

To start the data transfer process, an SPI interrupt may be forced by software by writing a 1 to the STR bit in the SPI Control Register (0x004).

22.9.2 Error Condition Interrupt

If any of the SPI error conditions occurs as described in the previous section, the corresponding error bit and the IRQ bit are set in the SPI Status register and the SPI interrupt is asserted. The error status bits and the IRQ bit should be cleared at the same time by writing a 1 to those bits.

22.9.3 Baud Rate Generator Time-out Interrupt

If the SPI is disabled, an SPI interrupt can be generated by a Baud Rate Generator time-out. This timer function must be enabled by setting the BIRQ bit in the SPI Control Register (0x004). See SPI Baud Rate Generator.

22.10 SPI Baud Rate Generator

22.10.1 Slave Mode

The Baud Rate Generator is not used in SPI slave mode,. When configured as a slave, the minimum SCLK period is 8 times the PCLK period.

22.10.2 Master Mode

In SPI master mode, the Baud Rate Generator creates a lower frequency serial clock (SCLK) for data transmission synchronization between the master and the external slave. The input to the Baud Rate Generator is the PCLK. The SPI Baud Rate register is a 16-bit reload value, BRG[15:0], for the SPI Baud Rate Generator. The reload value must be greater than or equal to 0002H for SPI operation (maximum baud rate is PCLK frequency divided by 4). The SPI baud rate is calculated using the following equation (for the special case BRG = 0x0000 substitute 2^{16} for BRG in the equation):

$$\text{SPI Baud Rate (bits/s)} = \text{SPI Baud Rate (bps)} = \frac{(\text{PCLK Frequency (Hz)})}{(2 \times \text{BRG}[15:0])}$$

22.10.3 Timer Mode

When the SPI is disabled, the Baud Rate Generator can function as a continuous mode 16-bit timer with interrupt on time-out. To configure the Baud Rate Generator as a timer with interrupt on time-out, complete the following procedure:

- Set SPIEN = 0 (in the Error Detection Register)
- Load the desired 16-bit count value into the SPI Baud Rate Register, BRG[15:0].
- Set BIRQ = 1



22.11 SPI Registers

Address assignments for SPI registers within each block are outlined in Table 2-2. Peripheral Bus Region. Reserved register bits should only be written as 0.

Each SPI instance is controlled by a block of registers assigned to that port; all blocks contain identical control, interrupt, status and FIFO read/write registers.

Register names for different instances can be defined by appending the instance number to the peripheral name, so for instance, the Control Register for SPI 0 would be SPI0_CTRL, while the Control Register for SPI 1 would be SPI1_CTRL, and so on.

Table 22-2. SPI Register Addresses (for each instance)

Offset	Access	Register	Description
0x0000	RW	SPI_DATA	SPI Data Register
0x0004	RW	SPI_CTRL	SPI Control Register
0x0008	RW	SPI_STATUS	SPI Status Register
0x000C	RW	SPI_MOD	SPI Mode Register
0x0010	RW		RFU
0x0014	RW	SPI_BRG	SPI Baud Rate Register
0x0018	RW	SPI_DMA	SPI DMA Register
0x001C	RW	I2S_CTRL	I2S Control Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



22.11.1 SPI Data Register (SPI_DATA, Offset 0x0000)

Table 22-3. SPI DATA (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	DATA[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	DATA[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description					Settings	
DATA	15:0	SPI Data					For a complete description consult the Data Movement section	
RFU	31:16	Reserved						



22.11.2 SPI Control Register (SPI_CNTL, Offset 0x0004)

Table 22-4. SPI_CNTL (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	IRQE	STR	BIRQ	PHASE	CLKPOL	WOR	MMEN	SPIEN
Reset	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
SPIEN	0	SPI Enable.	0: SPI operation is disabled. 1: SPI operation is enabled.
MMEN	1	SPI Master Mode Enable.	0: SPI configured in slave mode. 1: SPI configured in master mode.
WOR	2	Wired OR (open drain) Enable.	0: SPI signal pins not configured for open-drain. 1: All 4 SPI signal pins (SSEL, SCLK, MISO, MOSI) configured for open-drain function. This setting typically used for multi-master and/or multi-slave configurations.
CLKPOL	3	Clock Polarity.	0: SCLK idles Low (0) after character transmission/reception. 1: SCLK idles High (1) after character transmission/reception.
PHASE	4	Phase Select.	Refer to SPI Clock Phase and Polarity Control
BIRQ	5	Baud Rate Generator Timer Interrupt Request.	0: If SPIEN = "0", BIRQ = "0" disables the Baud Rate Generation timer function. If SPIEN = "1", this bit has no effect. 1: If SPIEN = "0", BIRQ = "1" enables the Baud Rate Generation timer function and time-out interrupt. If SPIEN = "1", this bit has no effect.
STR	6	Start SPI Interrupt.	Setting this bit starts a SPI interrupt request. Setting this bit also sets IRQ = "1" in the SPI_STAT. Setting this bit forces the SPI to send an interrupt request to the Interrupt Controller if IRQE = 1. This bit is cleared by writing a 0 to this bit or by writing a 1 to the IRQ bit in the SPI_STAT.



Name	Bits	Description	Settings
IRQE	7	Interrupt Request Enable.	0: SPI interrupts are disabled. 1: SPI interrupts are enabled. Interrupt requests are sent to the Interrupt Controller Note that if transmit or receive DMA is enabled, the transmit data complete interrupt is disabled, but other interrupt sources are enabled.
RFU	31:8	Reserved	

E/UG25H05/36488764



22.11.3 SPI Status Register (SPI_STATUS, Offset 0x0008)

Table 22-5. SPI_STATUS (Offset 0x0008)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	IRQ	TOVR	COL	ABT	ROVR	TUND	TXST	SLAS
Reset	0	0	0	0	0	0	0	1
Access	RW	RW	RW	RW	RW	RW	R	R

Name	Bits	Description	Settings
SLAS	0	Slave Select	If the SPI is in slave mode, this bit indicates if the SPI is selected. If the SPI is in master mode this bit has no meaning. 0 = Slave SPI is selected 1 = Slave SPI is not selected
TXST	1	Transmit Status	0: No data transmission currently in progress. 1: Data transmission currently in progress.
TUND	2	Transmit Underrun	This bit is set by hardware to indicate a transmit FIFO underrun has occurred. Once set, the bit must be written to a '1' by software to clear this flag. 0 = No FIFO underrun has occurred 1 = A FIFO underrun has occurred
ROVR	3	Receive Overrun	This bit is set by hardware when a receive FIFO overrun occurs. Once set, the bit must be written to a '1' by software to clear this flag. 0 = No FIFO overrun has occurred 1 = A FIFO overrun has occurred
ABT	4	Slave Mode Transaction Abort	This bit is set by hardware when a slave mode transaction abort occurs. Once set, the bit must be written to a '1' by software to clear this flag. 0 = No slave mode transaction abort has occurred 1 = A slave mode transaction abort has occurred
COL	5	Collision	This bit is set by hardware when a multi-master collision (mode fault) occurs. Once set, the bit must be written to a '1' by software to clear this flag. 0 = No multi-master collision has occurred 1 = A multi-master collision has occurred
TOVR	6	Transmit Overrun	This bit is set by hardware when a transmit FIFO overrun has occurred. Once set, the bit must be written to a '1' by software to clear this flag. 0 = No FIFO overrun has occurred 1 = A FIFO overrun has occurred



Name	Bits	Description	Settings
IRQ	7	SPI Interrupt Request	This bit is set by hardware when an SPI interrupt request is pending. Once set, the bit can only be cleared if xxxx xxFFh is written to this register, clearing this flag and all other error flags in this register. 0 = No SPI interrupt request is pending 1 = An SPI interrupt request is pending
RFU	31:8	Reserved.	

E/UG25H05/36488764



22.11.4 SPI Mode Register (SPI_MOD, Offset 0x000C)

Table 22-6. SPI_MOD (Offset 0x000C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[4:0]					SSL3	SSL2	SSL1
Reset	0000 0					0	0	0
Access	R					RW	RW	RW
Position	7	6	5	4	3	2	1	0
Field	TX_LJ	RFU	NUMBITS[3:0]				SSIO	SSV
Reset	0	0	00 00				0	0
Access	RW	R	RW				RW	RW

Name	Bits	Description	Settings
SSV	0	Slave Select Value	This indicates the value of the SSEL pin. 0 = If SSIO = 1 and the SPI is configured as a master, the SSEL pin will be driven low. 1 = If SSIO = 1 and the SPI is configured as a master, the SSEL pin will be driven high.
SSIO	1	Slave Select I/O	This indicates the direction of the SSEL pin. 0 = The SSEL pin is configured as an input. 1 = The SSEL pin is configured as an output. This is only valid if the device is operating in master mode.
NUMBITS[3:0]	5:2	Number of Data Bits per Character to Transfer	This field contains the number of bits to shift for each character transfer. Refer to the data movement chapter for information on valid bit positions when the character length is less than 16-bits. NUMBITS Number of bits per character 0000 16 0001 1 0010 2 ... 1110 14 1111 15
RFU	6	Reserved.	
TX_LJ	7	Transmit Left Justify	When NUMBITS[3:0] = 0000b, this bit is ignored. 0 = Transmit data is loaded directly from the transmit FIFO into the shift register. 1 = Transmit data is left justified as it is loaded into the shift register when character length is less than 16. For example, if NUMBITS = 0x1000 (8-bit characters), software or DMA writes the character data into the SPI_DATA[7:0] and hardware shifts the character to bits [15:8] when the shift register is loaded.
SSL1	8	Slave Select 1. If SPI is enabled and in master mode, the SSEL_1 is driven according to this bit.	0: SSEL_1 driven high 1: SSEL_1 driven low

Name	Bits	Description	Settings
SSL2	9	Slave Select 2. If SPI is enabled and in master mode, the SSEL_2 is driven according to this bit.	0: SSEL_2 driven high 1: SSEL_2 driven low
SSL3	10	Slave Select 3. If SPI is enabled and in master mode, the SSEL_3 is driven according to this bit.	0: SSEL_3 driven high. 1: SSEL_3 driven low
RFU	31:11	Reserved.	

E/UG25H05/36488764



22.11.5 SPI Baud Rate Generator Register (SPI_BRG, Offset 0x0014)

Table 22-7. SPI_BRG (Offset 0x0014)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	BRG[15:8]							
Reset	11111111							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	BRG[7:0]							
Reset	11111111							
Access	RW							
Name	Bits	Description					Settings	
BRG	15:0	Baud Rate Reload Value					The SPI Baud Rate register is a 16-bit reload value for the SPI Baud Rate Generator. The reload value must be greater than or equal to 0002H for proper SPI operation (maximum baud rate is PCLK frequency divided by 4).	
RFU	31:16	Reserved						



22.11.6 SPI DMA Register (SPI_DMA, Offset 0x0018)

Table 22-8. SPI DMA (Offset 0x0018)

Position	31	30	29	28	27	26	25	24
Field	RX_DMA_EN		RFU			RX_FIFO_CNT[3:0]		
Reset	0		000			0000		
Access	RW		R			R		
Position	23	22	21	20	19	18	17	16
Field		RFU		RX_FIFO_CLEAR	RFU		RX_FIFO_LEVEL[2:0]	
Reset		000		0	0		111	
Access		R		W	R		RW	
Position	15	14	13	12	11	10	9	8
Field	TX_DMA_EN		RFU			TX_FIFO_CNT[3:0]		
Reset	0		000			0000		
Access	RW		RW			R		
Position	7	6	5	4	3	2	1	0
Field		RFU		TX_FIFO_CLEAR	RFU		TX_FIFO_LEVEL[2:0]	
Reset		000		0	0		111	
Access		R		W	R		RW	

Name	Bits	Description	Settings
TX_FIFO_LEVEL	2:0	Transmit FIFO Level.	000: Request TxDMA when TxFIFO has 1 free entry. 001: Request TxDMA when TxFIFO has 2 free entries. 010: Request TxDMA when TxFIFO has 3 free entries. ... 111: Request TxDMA when TxFIFO has 8 free entries.
RFU	3	Reserved	
TX_FIFO_CLEAR	4	Transmit FIFO Clear	This bit always reads as 0. 0: No action 1: Reset TxFIFO
RFU	7:5	Reserved	
TX FIFO Count	11:8	Transmit FIFO Count.	0000: TxFIFO empty (0 entries) 0001: TxFIFO contains 1 entry 0010: TxFIFO contains 2 entries 0011: TxFIFO contains 3 entries ... 1000: TxFIFO contains 15 entries
RFU	14:12	Reserved	
TX DMA Enable	15	Transmit DMA Enable	Disabling clears any active request to the DMA controller. 0: Disable TX DMA requests 1: Enable TX DMA requests
RX FIFO Level	18:16	Receive FIFO Level	Sets the RX FIFO DMA request threshold. This configures the number of filled RxFIFO entries before activating an RxDMA request. 000: request RxDMA when RxFIFO contains 1 entry 001: request RxDMA when RxFIFO contains 2 entries 010: request RxDMA when RxFIFO contains 3 entries ... 111: request RxDMA when RxFIFO contains 8 entries
RFU	19	Reserved	



Name	Bits	Description	Settings
RX FIFO Clear	20	Receive FIFO Clear	This bit always reads as 0. 0: No action 1: Reset RxFIFO
RFU	23:21	Reserved	
RX FIFO Count	27:24	Receive FIFO Count	0000: RxFIFO empty (0 entries) 0001: RxFIFO contains 1 entry 0010: RxFIFO contains 2 entries 0011: RxFIFO contains 3 entries ... 1000: RxFIFO contains 15 entries
RFU	30:28	Reserved	
RX_DMA_EN	31	Receive DMA Enable	Disabling clears any active request to the DMA controller. 0: Disable RX DMA requests 1: Enable RX DMA requests



22.11.7 SPI I²S Control Register (I²S_CNTL, Offset 0x001C)

Table 22-9. I²S_CNTL (Offset 0x001C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU			I ² S_LJ	I ² S_MONO	I ² S_PAUSE	I ² S_MUTE	I ² S_EN
Reset	000			0	0	0	0	0
Access	R			RW	RW	RW	RW	RW

Name	Bits	Description	Settings
I ² S_EN	0	I ² S Mode Enable	0 - I ² S mode is disabled. 1 - I ² S mode is enabled.
I ² S_MUTE	1	I ² S Mute transmit	0 - Normal transmit. 1 - Transmit data is replaced with 0.
I ² S_PAUSE	2	I ² S Pause transmit/receive	0 - Normal transmission/reception. 1 - Halt transmit and receive FIFO and DMA accesses, transmit 0's
I ² S_MONO	3	I ² S Monophonic Audio Mode	0 - Stereophonic audio. 1 - Monophonic audio format. Each transmit data word is replicated on both left/right channels. Receive data is taken from left channel, right channel receive data is ignored.
I ² S_LJ	4	I ² S Left Justify	0 - Normal I ² S audio protocol - audio data lags left/right channel signal by one SCLK period. 1 - Audio data is synchronized with SSEL (left/right channel signal).
RFU	31:5	Reserved	



23.0 Smart Card

The MAX32550 smart card controller embeds both the digital core and analog transceiver of the smart card interface. The built-in transceiver is responsible of voltage translation according to the EMV or ISO7816-1 standard and can support 1.8V, 3V and 5V cards. A bypass mode is available in order to use an external transceiver. The dedicated card detection input can be used to wake-up the device when in standby mode.

The ISO-7816 UART supports the following features:

- ISO/IEC 7816 standards supported
- Supports 1.8V, 3V and 5V cards
- Interrupts on all Smart Card events
- Bypass mode to use external transceiver.
- Supports both synchronous and asynchronous cards
- Hardware debounce on card insertion
- Hardware fault detection and emergency deactivation on card supply overcurrent and short circuit detection
- Automated control of activation and deactivation sequence.
- Hardware parity handling on transmit and receive including retransmits
- Hardware timers to handle Guard Time and Work Wait Time
- General purpose Smart Card clock counter
- 8 byte FIFOs for transmit and receive with programmable thresholds
- Manual control option for all pins

23.1 UART Clock Source

The clock for this block is based on PLL1 and it can be controlled from Smart Card Clock Register (SCCK, 0x4000_0034).

Smart Card block may be configured in several different ways. For example user may bypass the sequencer or bypass the transceiver independently. Table 23-1 shows all the supported combinations.

Table 23-1. Smart Card Clock Selection

SC_CR.BYPASS_PH Y	SC_CR.BYPASS_SE Q	SC_PN.CLK_E N	SC_PN.CLKSE L	CLK SOURCE	CLK OUTPU T
0	0	x	0	SC_PN.CRDCL K	PHY
0	0	x	1	UART	PHY
0	1	0	x	always 0	PHY
0	1	1	0	SC_PN.CRDCL K	PHY
0	1	1	1	UART	PHY
1	x	x	0	SC_PN.CRDCL K	GPIO
1	x	x	1	UART	GPIO



23.2 Smart Card Interrupts

Twelve interrupt sources are provided to signal following Smart Card UART events.

- Parity Error
- Waiting Time Overflow
- Clock counter Overflow
- Character Transmission Complete
- Receive FIFO Empty
- Receive FIFO Threshold Reached
- Receive FIFO Full
- Transmit FIFO Empty
- Transmit FIFO Threshold Reached
- Protection Card Interface
- Card Presence Detect
- Activation Sequencer

Each interrupt may be individually enabled or disabled via the Interrupt Enable Register (SC_IER). When an interrupt occurs, the appropriate flag in the Interrupt Status Register (SC_ISR) is set. Once the interrupt service code is run, the interrupt flag should be cleared so that following interrupts may signal new events. The FIFO related interrupt flags are unique as they may only be cleared once the event trigger is removed. For example, if the Receive FIFO is full and RXFIS=1, removing data from the FIFO via read of RXR must occur before attempting to clear RXFIS.

23.3 Basic Initialization

23.3.1 Bypass mode (External Transceiver)

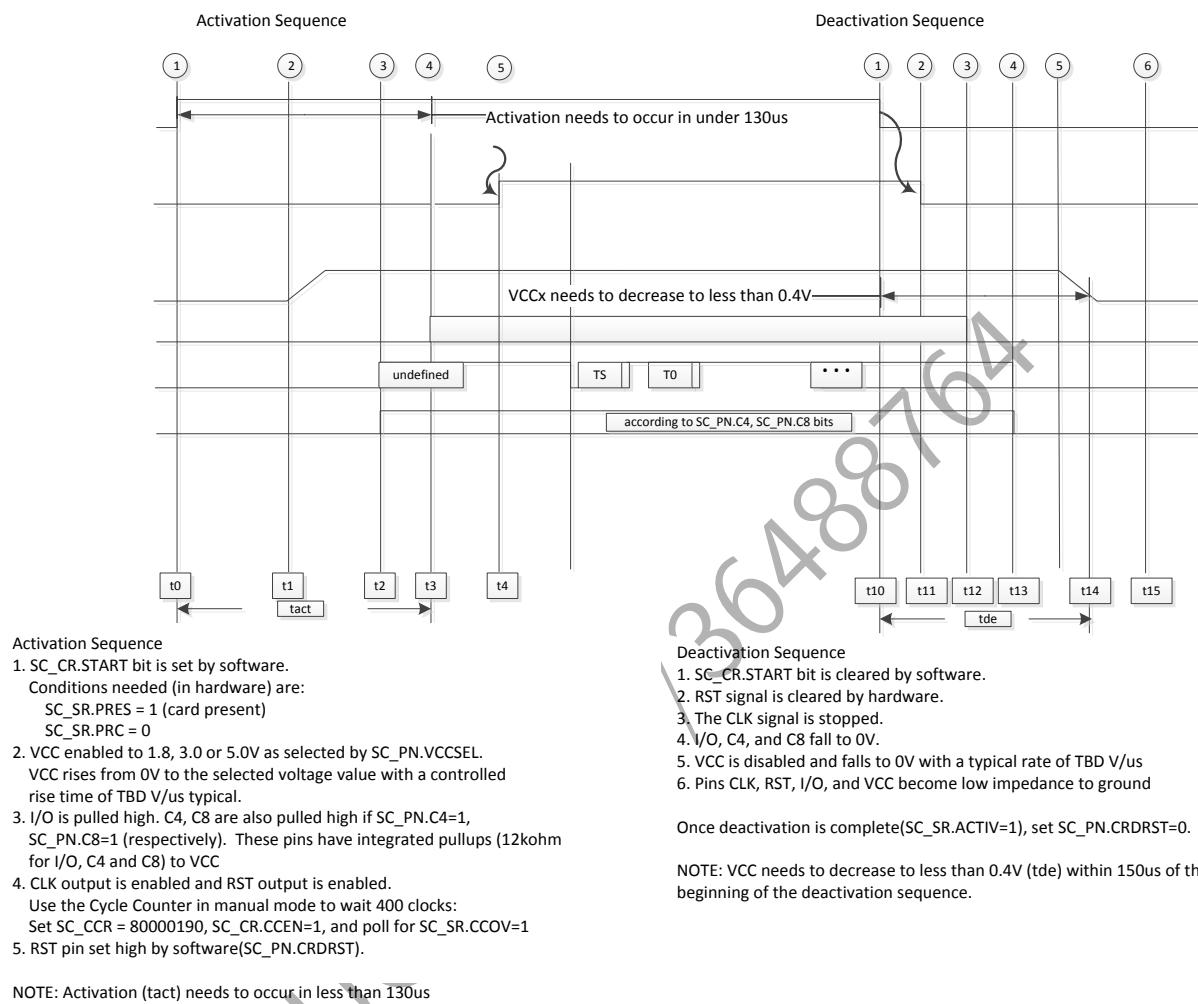
Follow the following steps to initialize the Smart Card peripheral for Smart Card power up.

- 1) Set SCCK to a value which generates a 1-5MHz clock source for the Smart Card
- 2) Enable PLL1 and wait until ready
- 3) Set the ETU counter to the power up default of F/D=372. SC_ETUR=372
- 4) Initialize RST and C4/C8, enable UART control of CLK pin. SC_PN [7:0] = 0x38
- 5) Enable Smart Card UART and bring the UART pins out to GPIO by bypassing the Smart Card transceiver. SC_CR = 0x21108
- 6) Proceed with normal Smart Card power up sequencing.

23.3.2 Normal mode (Internal Transceiver)

Follow the following steps to initialize the Smart Card peripheral for Smart Card power up.

- 1) Set SCCK to a value which generates a 1-5MHz clock source for the Smart Card
- 2) Enable PLL1 and wait until ready
- 3) Set the ETU counter to the power up default of F/D=372. SC_ETUR=372
- 4) Set Card Vcc by setting SC_PN.VCCSEL
- 5) Initialize RST and C4/C8, enable UART control of CLK pin. SC_PN [7:0] = 0x38
- 6) Enable Smart Card UART. SC_CR = 0x1108
- 7) Wait for the card insertion (i.e. polling either SC_SR.PRES or SC_SR.PDL bit or using the SC_ISR.PDLIS interrupt (enable when SC_IE.PDLIE = 1)).
- 8) Use the sequencer to initiate the internal transceiver power-up sequence.

**Figure 23-1. Activation and De-activation Sequence**

23.4 Event Timing

The Smart Card peripheral provides three timers.

SC_GTR – Controls the guard time between falling edges of transmitted bytes. The guard time is defined as a count of ETUs.

SC_WT0R/SC_WT1R – A timer with ETU clock time base. This timer may be used for any timeout or measurement which uses ETU timing units. (e.g. Smart Card work wait time)

SC_CCR – A timer with Smart Card CLK time base. This timer may be used for any timeout or measurement which uses clock cycle timing units. (e.g. Smart Card minimum clocks before release of RST)

23.5 Transferring and Receiving Data

Data to be transferred must be written to the TX FIFO by writing to the SC_DATA register. This data will be transferred out by the hardware automatically a character at a time, in the order that it was written. The status flags and associated Smart Card interrupts can be used to monitor the FIFO status and determine when the transfer cycle or cycles have completed.

As data is received, it is loaded into the RX FIFO, and it can then be unloaded by reading from the SC_DATA register. If 8 characters are received and are not unloaded by the CPU, and another character arrives, an overrun error will occur and the new character will be lost.

23.6 Emergency Deactivation

All emergency deactivations will execute the full deactivation sequence in hardware shown in Figure 23-1. Emergency deactivation may occur if overcurrent detected on RST and Vcc pins. In case of an emergency deactivation, SC_PHYCR.START bit is cleared by the hardware.

23.7 Supply Sequence Options during card session

The Smart Card supply voltage may be switched without restriction. The card may be first deactivated, a new smart card supply setting is chosen, and the card is reactivated. The Smart Card sequencer adjusts the smart card supply start up at activation, accounting for the previous card supply configuration.

23.8 Smart Card Registers

Address assignments for the Smart Card registers are outlined in Table 23-2.

Table 23-2. Smart Card Register Addresses (Base ADDR = 0x4002_C000)

Offset	Access	Register	Description
0x0000	RW	SC_CR	Control Register
0x0004	RW	SC_SR	Status Register
0x0008	RW	SC_PN	Pin Register
0x000C	RW	SC_ETUR	ETU Register
0x0010	RW	SC_GTR	Guard Time Register
0x0014	RW	SC_WT0R	Waiting Time 0 Register
0x0018	RW	SC_WT1R	Waiting Time 1 Register
0x001C	RW	SC_IER	Interrupt Enable Register
0x0020	RW	SC_ISR	Interrupt Status Register
0x0024	RW	SC_TXR	Transmit Register
0x0028	RW	SC_RXR	Receive Register
0x002C	RW	SC_CCR	Clock Counter Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



23.8.1 Smart Card Control Register (SC_CR, Offset 0x0000)

Table 23-3. SC CR (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	DUAL_MODE*	SEQDLY[1:0]*	BYPASS_SEQ	DEBNCE	PRPOL	BYPASS_PHY	START	
Reset	0	10	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

Position	15	14	13	12	11	10	9	8
Field	TXTHD[3:0]					RXTHD[3:0]		
Reset	0000					0000		
Access	RW					RW		

Position	7	6	5	4	3	2	1	0
Field	RFU	TXFLUSH	RXFLUSH	CCEN	UART	WTEN	CREP	CONV
Reset	0	0	0	0	0	0	0	0
Access	R	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
CONV	0	Convention Select Bit.	0: Direct Convention. 1: Indirect Convention.
CREP	1	Character Repeat Enable Bit. Enables character retransmit on parity error.	0: Character repetition disabled. 1: Character repetition enabled.
WTEN	2	Waiting Time Counter Enable Bit.	0: Stop waiting time counter. 1: Start waiting time counter.
UART	3	Smart Card Mode Bit. Selects manual control or smart card UART control of smart card pins.	0: Manual pin control. 1: Smart Card UART pin control.
CCEN	4	Clock Counter Enable Bit.	0: Stop clock counter. 1: Start clock counter.
RXFLUSH	5	Receive FIFO Flush Bit. This bit is cleared automatically by hardware.	1: Flush RX FIFO.
TXFLUSH	6	Transmit FIFO Flush Bit. This bit is cleared automatically by hardware.	1: Flush TX FIFO.
RFU	7	Reserved	N/A
RXTHD	11:8	Receive FIFO depth. Valid range 1 to 7.	0001 - 0111
TXTHD	15:12	Transmit FIFO depth. Valid range 1 to 7.	0001 - 0111
START	16	Start bit. This bit controls software activation/deactivation of the card interface. When this bit is set, the activation sequence for the selected card is performed. When this bit is cleared, the deactivation sequence for the selected card is performed. Hardware should automatically reset the START bit during emergency deactivation.	0: Deactivation performed 1: Activation performed
BYPASS_PHY	17	Bypass mode for smart card PHY.	0: Smart Card PHY used. 1: Smart Card PHY bypassed, UART pins brought out to GPIO



Name	Bits	Description	Settings										
PRPOL	18	Presence detect polarity. When PRPOL = 0, card presence pin shall interpret a logic 1 as card presence. When PRPOL = 1, card presence pin shall interpret a logic 0 as card presence.	0: Logic 1 presence detect 1: Logic 0 presence detect										
DEBNCE	19	Enable Debounce on card insertion. When set to logic 1, this will enable the hardware debounce of the card detect pin. The debounce function shall wait for fixed timer period of stable card detect assertion before setting the SC_ISR.PRIS bit. De-assertion of SC_ISR.PRIS is immediate upon de-assertion of card detect pin.	0: No wait time. 1: 8ms wait time										
BYPASS_SEQ	20	Sequencer bypass. Disable sequencer for activation and deactivation	0: Enable of Smart Card pins are managed by the sequencer 1: manual mode. Enable of the Smart Card pins are managed by individual enable bits in SC_PN register										
SEQDLY*	22:21	Smart Card Sequencer Delay. Setting these bits allow for user control of the sequencer delay so that SC_VDD is stable prior to initiating activity on the Smart Card pins. The user should choose a value which exceeds the minimum delay required, which is dependent on both card selection (SC_PN.VCCSEL) and the Smart Card LDO slew rate: $\text{Min_SeqDly} = (\text{Smart Card SC_VDD desired value}) / (\text{Smart Card LDO slew rate})$ The default setting should be sufficient for 5V card operation	<table border="1"> <thead> <tr> <th>SEQDLY</th> <th>LDO Start to Smart Card Signal Enable</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>52 µs</td> </tr> <tr> <td>01</td> <td>65 µs</td> </tr> <tr> <td>10</td> <td>79 µs (Default)</td> </tr> <tr> <td>11</td> <td>92 µs</td> </tr> </tbody> </table>	SEQDLY	LDO Start to Smart Card Signal Enable	00	52 µs	01	65 µs	10	79 µs (Default)	11	92 µs
SEQDLY	LDO Start to Smart Card Signal Enable												
00	52 µs												
01	65 µs												
10	79 µs (Default)												
11	92 µs												
DUAL_MODE*	23	Dual Mode. If set, the internal Phy is still powered and Card signals are preserved (SC_IO, SC_C4, SC_C8, and SC_RST are latched, SC_CLK is running) when SC_CR.BYPASS_PHY is set, in this way it is possible to alternatively communicate with a smart card connected to the internal Phy and a SAM connected to GPIO (but using the smart card Uart), without closing the communications to Smart card and SAM. Refer to specific application note.	0: Legacy Mode. If SC_CR.BYPASS_PHY is set the internal Phy is powered off 1:SC_CR.BYPASS_PHY is set, the internal Phy is still powered and smart card signals are preserved.										
RFU	31:24	Reserved	N/A										

* Not applicable on revision A silicon.

23.8.2 Smart Card Status Register (SC_SR, Offset 0x0004)

Table 23-4. SC_SR (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[2:0]					ACTIV	PDL	PRC
Reset	000		0	0	0	0	0	0
Access	R		RW	RW	RW	R	RW	

Position	15	14	13	12	11	10	9	8
Field	TXELT[3:0]					RXELT[3:0]		
Reset	0000					0000		
Access	R					R		

Position	7	6	5	4	3	2	1	0
Field	TXFULL	TXEMPTY	RXFULL	RXEMPTY	TXCF	CCOV	WTOV	PAR
Reset	0	1	0	1	0	0	0	0
Access	R	R	R	R	RW	R	R	RW

Name	Bits	Description	Settings
PAR	0	Parity Error Detected Flag. Set to 1 by hardware if parity error is detected. Cleared by software.	
WTOV	1	Waiting Time Counter Overflow Flag. Set to 1 by hardware when waiting time counter overflows.	
CCOV	2	Clock Counter Overflow Flag. Set to 1 by hardware when clock counter has reached its maximum value. See Note.	
TXCF	3	Transmit Complete Flag. Set to 1 by hardware when character has been transmitted. Cleared by software.	
RXEMPTY	4	Receive FIFO Empty Flag.	0: RX FIFO is not empty. 1: RX FIFO is empty (RXELT=0).
RXFULL	5	Receive FIFO Full Flag.	0: RX FIFO is not full. 1: RX FIFO is full (RXELT=8).
TXEMPTY	6	Transmit FIFO Empty Flag.	0: TX FIFO is not empty. 1: TX FIFO is empty (TXELT=0).
TXFULL	7	Transmit FIFO Full Flag.	0: TX FIFO is not full. 1: TX FIFO is full (TXELT=8).
RXELT	11:8	Number of bytes in Receive FIFO.	0000 - 1000
TXELT	15:12	Number of bytes in Transmit FIFO.	0000 – 1000
PRES	16	Card Presence Indication. The presence indication depends on the polarity bit (SC_CR.PRPOL): PRPOL = 0=> PRES=1 on rising edge of SC_DETECT PRPOL = 1=>PRES=1 on falling edge of SC_DETECT pin. Cleared by hardware.	0: Card not present 1: Card presence detected

Name	Bits	Description	Settings
RFU	17	Reserved	N/A
PRC	18	Protection Card Interface Status Flag. This bit is set to a 1 by hardware when a fault is detected on the card reader interface. A fault is defined as a detection of a short-circuit condition on the RST or Vcc pin. This bit may be cleared by software to a 0.	0: Normal current on PHY VCC supply 1: Over current condition on PHY VCC supply.
PDL	19	Presence detect Level Status Flag. This bit is set to a 1 when a level change has been detected on the SC_DETECT pin. If the interrupt is enabled for this flag, a system interrupt will be fired. If the interrupt enable is not set, the flag will be set, but no interrupt will fire. This bit cleared by software.	0: No level change detected on SC_DETECT 1: Level change detected on SC_DETECT
ACTIV	20	Activation Sequencer Status Flag.	0: Activation or deactivation has begun or is ongoing 1: Activation or deactivation sequence is complete.
RFU	31:21	Reserved.	N/A

Note: This bit goes low during counting but due to clock domain differences, there is a delay before it goes low. The delay mainly depends on the SC clock frequency. If this bit is read-back just after starting the counter, in some cases, it will read back as "1" instead of "0".

To check counter overflow, it is recommended to use the CTIS bit in the SC_ISR register, this bit goes high when the counter reaches the counting value. This bit works even if the interrupts TCIE bit in SC_IER is not set. A preferred way however is to enable interrupt by setting TCIE bit in SC_IER register and use CTIS interrupt to check the timer (SC_CCR) overflow.



23.8.3 Smart Card Pin Register (SC_PN, Offset 0x0008)

Table 23-5. SC PN (Offset 0x0008)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[3:0]		RAMP_EN	VCC_EN	RST_EN	CLK_EN	IO_C48_EN	
Reset	0000		0	0	0	0	0	
Access	R		RW	RW	RW	RW	RW	

Position	15	14	13	12	11	10	9	8
Field	RFU[5:0]						VCCSEL[1:0]	
Reset	0000 00						00	
Access	R						RW	

Position	7	6	5	4	3	2	1	0
Field	RFU[1:0]		CLKSEL	CRDC8	CRDC4	CRDIO	CRDCLK	CRDRST
Reset	00		0	0	0	0	0	0
Access	R		RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
CRDRST	0	Smart Card Reset Pin Control. Writes to this bit set the associated pin logic level. Reads from this bit return the logic value present on the associated pin.	0: RST low. 1: RST high.
CRDCLK	1	Smart Card Clock Pin Control. Writes to this bit set the associated pin logic level. Reads from this bit return the logic value present on the associated pin.	0: CLK low. 1: CLK high.
CRDIO	2	Smart Card I/O Pin Control. Writes to this bit set the associated pin logic level. Reads from this bit return the logic value present on the associated pin.	0: IO low. 1: IO high.
CRDC4	3	Smart Card C4 Pin Control. Writes to this bit set the associated pin logic level. Reads from this bit return the logic value present on the associated pin.	0: C4 low. 1: C4 high.
CRDC8	4	Smart Card C8 Pin Control. Writes to this bit set the associated pin logic level. Reads from this bit return the logic value present on the associated pin.	0: C8 low. 1: C8 high.
CLKSEL	5	Smart Card Clock Control Select.	0: Clock Pin controlled by CRDCLK bit. 1: Clock Pin controlled by UART.
RFU	7:6	Reserved	N/A
VCCSEL	9:8	Selects supply for smart card. This value must be set to select a target Vcc during card activation.	00: 1.8V 01: 3.0V 1x: 5.0V
RFU	15:10	Reserved	N/A
IO_C48_EN	16	Smart Card IOs enable control, manual mode only (if SC_CR.BYPASS_SEQ= 1).	0: SC_IO, SC_C4 and SC_C8 are disabled 1: SC_IO, SC_C4 and SC_C8 are enabled
CLK_EN	17	Smart Card CLK enable control, manual mode only (if SC_CR.BYPASS_SEQ= 1).	0: SC_CLK is disabled 1: SC_CLK is enabled



Name	Bits	Description	Settings
RST_EN	18	Smart Card RST enable control, manual mode only (if SC_CR.BYPASS_SEQ= 1).	0: SC_RST is disabled 1: SC_RST is enabled
VCC_EN	19	Smart Card supply enable control, manual mode only (if SC_CR.BYPASS_SEQ= 1).	0: CARD_VCC is disabled 1: CARD_VCC is enabled according to VCCSEL bits
RAMP_EN	20	Smart Card LDO ramp enable control, manual mode only(if SC_CR.BYPASS_SEQ=1).	0: LDO is disabled 1: LDO is enabled
RFU	31:21	Reserved	N/A

E/UG25H05/36488764



23.8.4 Smart Card ETU Register (SC_ETUR, Offset 0x000C)

Table 23-6. SC ETUR (Offset 0x000C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[6:0]							
Reset	000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8	
Field	COMP	ETU[14:8]							
Reset	0	000 0001							
Access	RW	RW							

Position	7	6	5	4	3	2	1	0
Field	ETU[7:0]							
Reset	0111 0100							
Access	RW							

Name	Bits	Description	Settings
ETU	14:0	Elemental Time Unit value. If HALF=0, Number of clocks in one asynchronous bit time. If HALF=1, Number of clocks in $\frac{1}{2}$ of one asynchronous bit time.	
COMP	15	Compensation Mode Enable Bit. Controls 1 clock cycle subtraction of ETU value on odd bits.	0: Compensation Disabled. 1: Compensation Enabled.
HALF	16	Half ETU Count Selection Bit. NOTE: Guard Time (GT) and Wait Time (WT) must be doubled if HALF=1.	0: ETU value represents required clocks for one ETU 1: ETU value represents required clocks for one half ETU
RFU	31:17	Reserved	N/A



23.8.5 Smart Card Guard Time Register (SC_GTR, Offset 0x0010)

The Guard Time Register counts the number of ETU cycles required between start bit edges on transmit.

Table 23-7. SC_GTR (Offset 0x0010)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Other Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Other Reset	0000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8
Field	GT[15:8]							
Other Reset	0000 0000							
Access	RW							

Position	7	6	5	4	3	2	1	0
Field	GT[7:0]							
Other Reset	0000 000C							
Access	RW							

Name	Bits	Description	Settings
GT	15:0	Guard Time: Minimum time between two consecutive start bits in transmit mode. Guard Time is expressed in ETUs. NOTE: Guard Time must be doubled if ETUR.HALF=1. NOTE: Writes to this register may be delayed up to two ETU cycles.	
RFU	31:16	Reserved	N/A



23.8.6 Smart Card Wait Time 0 Register (SC_WT0R, Offset 0x0014)

Writes to this register set the reload value and stop the current Wait Time counter.

Table 23-8. SC_WT0R (Offset 0x0014)

Position	31	30	29	28	27	26	25	24
Field	WT[31:24]							
Reset	0000 0000							
Access	RW							

Position	23	22	21	20	19	18	17	16
Field	WT[23:16]							
Reset	0000 0000							
Access	RW							

Position	15	14	13	12	11	10	9	8
Field	WT[15:8]							
Reset	0010 0101							
Access	RW							

Position	7	6	5	4	3	2	1	0
Field	WT[7:0]							
Reset	1000 0000							
Access	RW							

Name	Bits	Description	Settings
WT	31:0	Wait Time: Least Significant 32 bits of the Wait Time Counter expressed in ETU. NOTE: Wait Time must be doubled if ETUR.HALF=1. NOTE: Writes to this register may be delayed up to two ETU cycles.	



23.8.7 Smart Card Wait Time 1 Register (SC_WT1R, Offset 0x0018)

Writes to this register set the reload value and stop the current Wait Time counter.

Table 23-9. SC_WT1R (Offset 0x0018)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	7	6	5	4	3	2	1	0
Field	WT[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
WT	7:0	Wait Time: Most Significant 8 bits of the Wait Time Counter expressed in ETU. NOTE: Wait Time must be doubled if ETUR.HALF=1. NOTE: Writes to this register may be delayed up to two ETU cycles.	
RFU	31:8	Reserved	N/A



23.8.8 Smart Card Interrupt Enable Register (SC_IER, Offset 0x001C)

Table 23-10. SC_IER (Offset 0x001C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8
Field	RFU[3:0]					ACTIVIE	PDLIE	PRCIE
Reset	0000		0		0	0	0	0
Access	R		RW		RW	RW	RW	RW

Position	7	6	5	4	3	2	1	0
Field	TXEIE	RXFIE	RXTIE	RXEIE	TCIE	CTIE	WTIE	PARIE
Reset	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
PARIE	0	Parity Error Interrupt Enable.	0: Interrupt Disabled. 1: Interrupt Enabled.
WTIE	1	Waiting Time Overflow Interrupt Enable.	0: Interrupt Disabled. 1: Interrupt Enabled.
CTIE	2	Clock Counter Overflow Interrupt Enable.	0: Interrupt Disabled. 1: Interrupt Enabled.
TCIE	3	Character Transmission Completion Interrupt Enable.	0: Interrupt Disabled. 1: Interrupt Enabled.
RXEIE	4	Receive FIFO Empty Interrupt Enable.	0: Interrupt Disabled. 1: Interrupt Enabled.
RXTIE	5	Receive FIFO Threshold Reached Interrupt Enable.	0: Interrupt Disabled. 1: Interrupt Enabled.
RXFIE	6	Receive FIFO Full Interrupt Enable.	0: Interrupt Disabled. 1: Interrupt Enabled.
TXEIE	7	Transmit FIFO Empty Interrupt Enable.	0: Interrupt Disabled. 1: Interrupt Enabled.
TXTIE	8	Transmit FIFO Threshold Reached Interrupt Enable.	0: Interrupt Disabled. 1: Interrupt Enabled.
PRCIE	9	Protection Card Interface Interrupt Enable	0: Interrupt Disabled. 1: Interrupt Enabled.
PDLIE	10	Presence Detect Interrupt Enable.	0: Interrupt Disabled. 1: Interrupt Enabled.
ACTIVIE	11	Activation Sequencer Interrupt Enable	0: Interrupt Disabled. 1: Interrupt Enabled.
RFU	31:12	Reserved	N/A



23.8.9 Smart Card Interrupt Status Register (SC_ISR, Offset 0x0020)

Table 23-11. SC ISR (Offset 0x0020)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8
Field	RFU[3:0]					ACTIVIS	PDLIS	PRCIS
Reset	0000					1	0	0
Access	R					RW	RW	RW

Position	7	6	5	4	3	2	1	0
Field	TXEIS	RXFIS	RXTIS	RXEIS	TCIS	CTIS	WTIS	PARIS
Reset	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
PARIS	0	Parity Error Interrupt Status Flag. This bit is set by hardware. If the interrupt is enabled for this flag, a system interrupt will be fired. If the interrupt enable is not set, the flag will be set, but no interrupt will fire. This bit cleared by software.	
WTIS	1	Waiting Time Overflow Interrupt Status Flag. This bit is set by hardware. If the interrupt is enabled for this flag, a system interrupt will be fired. If the interrupt enable is not set, the flag will be set, but no interrupt will fire. This bit cleared by software.	
CTIS	2	Clock Counter Overflow Interrupt Status Flag. This bit is set by hardware. If the interrupt is enabled for this flag, a system interrupt will be fired. If the interrupt enable is not set, the flag will be set, but no interrupt will fire. This bit cleared by software.	
TCIS	3	Character Transmission Completion Interrupt Status Flag. This bit is set by hardware. If the interrupt is enabled for this flag, a system interrupt will be fired. If the interrupt enable is not set, the flag will be set, but no interrupt will fire. This bit cleared by software.	
RXEIS	4	Receive FIFO Empty Interrupt Status Flag. This bit is set by hardware. If the interrupt is enabled for this flag, a system interrupt will be fired. If the interrupt enable is not set, the flag will be set, but no interrupt will fire. This bit cleared by software.	

Name	Bits	Description	Settings
RXTIS	5	Receive FIFO Threshold Reached Interrupt Status Flag. This bit is set by hardware. If the interrupt is enabled for this flag, a system interrupt will be fired. If the interrupt enable is not set, the flag will be set, but no interrupt will fire. This bit cleared by software.	
RXFIS	6	Receive FIFO Full Interrupt Status Flag. This bit is set by hardware. If the interrupt is enabled for this flag, a system interrupt will be fired. If the interrupt enable is not set, the flag will be set, but no interrupt will fire. This bit cleared by software.	
TXEIS	7	Transmit FIFO Empty Interrupt Status Flag. This bit is set by hardware. If the interrupt is enabled for this flag, a system interrupt will be fired. If the interrupt enable is not set, the flag will be set, but no interrupt will fire. This bit cleared by software.	
TXTIS	8	Transmit FIFO Threshold Reached Interrupt Status Flag. This bit is set by hardware. If the interrupt is enabled for this flag, a system interrupt will be fired. If the interrupt enable is not set, the flag will be set, but no interrupt will fire. This bit cleared by software.	
PRCIS	9	Protection Card Interface Interrupt Status Flag. This bit is set to a 1 by hardware when a fault is detected on the card reader interface. A fault is defined as a detection of a short-circuit condition on the smart card RST or smart card supply pin. This bit may be cleared by software to a 0.	0: Normal current on PHY VCC supply 1: Over current condition on PHY VCC supply.
PDLIS	10	Presence detect Interrupt Status Flag. This bit is set to a 1 when a card has been inserted or removed according to the SC_CR.PRPOL bit. If the interrupt is enabled for this flag, a system interrupt will be fired. If the interrupt enable is not set, the flag will be set, but no interrupt will fire. This bit cleared by software.	0:level change has not been detected on SC_DETECT 1:level change has been detected on SC_DETECT
ACTIVIS	11	Activation Sequencer Interrupt Status Flag. This bit is cleared to zero by software when an activation sequence is initiated. Once the activation sequence is complete, the bit is set to 1 by hardware.	0: Activation sequence has been initiated or is on-going. 1: Activation or deactivation sequence is completed.
RFU	31:12	Reserved	N/A



23.8.10 Smart Card Transmit Register (SC_TXR, Offset 0x0024)

Table 23-12. SC_TXR (Offset 0x0024)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	7	6	5	4	3	2	1	0
Field	DATA[7:0]							
Reset	0000 0000							
Access	W							

Name	Bits	Description	Settings
DATA	7:0	Transmit Data. Writes to this register will load the 8 bit input data into the FIFO.	
RFU	31:8	Reserved	N/A



23.8.11 Smart Card Receive Register (SC_RXR, Offset 0x0028)

This register provides a read interface to the receive FIFO. Each read of this register contains both the received data and associated parity error detect bit.

Table 23-13. SC_RXR (Offset 0x0028)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Position	15	14	13	12	11	10	9	8
Field	RFU[6:0]							
Reset	0000 0000							
Access	R							

Position	7	6	5	4	3	2	1	0
Field	DATA[7:0]							
Reset	0000 0000							
Access	R							

Name	Bits	Description	Settings
DATA	7:0	Receive Data. Reads from this register will unload 8 bit receive data from the FIFO.	
PARER	8	Parity Error Detect Bit. Parity error bit associated with lower bits 7:0.	0: No parity error. 1: Parity error detected.
RFU	31:9	Reserved	N/A



23.8.12 Smart Card Clock Counter Register (SC_CCR, Offset 0x002C)

Table 23-14. SC_CCR (Offset 0x002C)

Position	31	30	29	28	27	26	25	24
Field	MAN			RFU[6:0]				
Reset	0			000 0000				
Access	RW			R				

Position	23	22	21	20	19	18	17	16
Field				CCYC[23:16]				
Reset				0000 0000				
Access				RW				

Position	15	14	13	12	11	10	9	8
Field				CCYC[15:8]				
Reset				0000 0000				
Access				RW				

Position	7	6	5	4	3	2	1	0
Field				CCYC[7:0]				
Reset				0000 0000				
Access				RW				

Name	Bits	Description	Settings
CCYC	23:0	Number of Clock Cycles to Count. NOTE: Writes to this register may be delayed up to two ETU cycles.	
RFU	30:24	Reserved	N/A
MAN	31	Manual Mode. The counter starts counting-down on if this bit is set instead of an AUTOMATIC hardware mechanism	0: UART controls this counter (counter is reloaded at each character transmission). 1: Counter is loaded when this register is written and then stars to decrement at each clock cycle.



24.0 TFT Controller

The Color LCD Controller is provided to make **seamless** connection to an LCD display such as:

- Active Matrix TFT panels with up to 24-bit bus interface
- Color STN panels, 8-bit bus interface

The panel resolution is programmable from 320 x 200 to 4096 x 4096. The LCD Controller supports a number of color modes depending of the supplied LCD panel:

- **TFT Panels**
 - 1 bpp, palletized, 2 colors from available colors
 - 2 bpp, palletized, 4 colors from available colors
 - 4 bpp, palletized, 16 colors
 - 8 bpp, palletized, 256 colors
 - 16 bpp, direct 5:5:6 BGR with one bit (R0) not normally used and can be used as an intensity bit.
 - 16 bpp, direct 5:6:5 RGB
 - 24 bpp, direct 8:8:8 RGB
- **Color STN Panels**
 - 1 bpp, palletized, 2 colors
 - 2 bpp, palletized, 4 colors
 - 4 bpp, palletized, 16 colors
 - 8 bpp, palletized, 256 colors
 - 16 bpp, direct 5:5:5 with one bit not being used

Furthermore, the controller provides the following functionality:

- Horizontal front and back porch
- Horizontal synchronization pulse width
- Number of pixels per line
- Vertical front and back porch
- Vertical synchronization pulse width
- **Number of lines per panel**
- Signal polarity, Active High or Low (LCDCLK, VSYNC, HSYNC, VDEN, LEND)
- Panel Clock frequency
- Bits per pixel
- Display Type STN color or TFT
- Little Endian, Big Endian or WinCE mode
- Interrupt Generation

24.1 Block Diagram

Figure 24-1 shows the simplified block diagram depicts the flow of data for the color LCD panel. The color LCD controller performs translation of pixel-coded data into the required formats and timings to drive a variety of color LCDs. The data processing procedures are described as the following:



- Packets of pixel coded data are fetched by the AMBA AHB master interface and are stored into a 32-bit wide DMA FIFO which acts as an input data flow buffer. The buffered pixel coded data is then converted to a stream of single pixel data by a pixel process engine.
- For TFT displays with color resolution of 16/24bpp, pixel process engine bypasses the palette RAM and directly streams the FIFO output pixel data onto the panel interface lines. The gray scaler is not used for TFT displays.
- For STN displays, either a value obtained from the addressed palette location or the value bypassed from the pixel process engine is sent to the gray scaling generator. The gray scaled data is then formatted to satisfy the data pattern of LCD panel and drives the display output. In the case of TFT display, either an addressed palette value or the serialized data is sent directly to the output display drivers, bypassing the gray scaling logic.

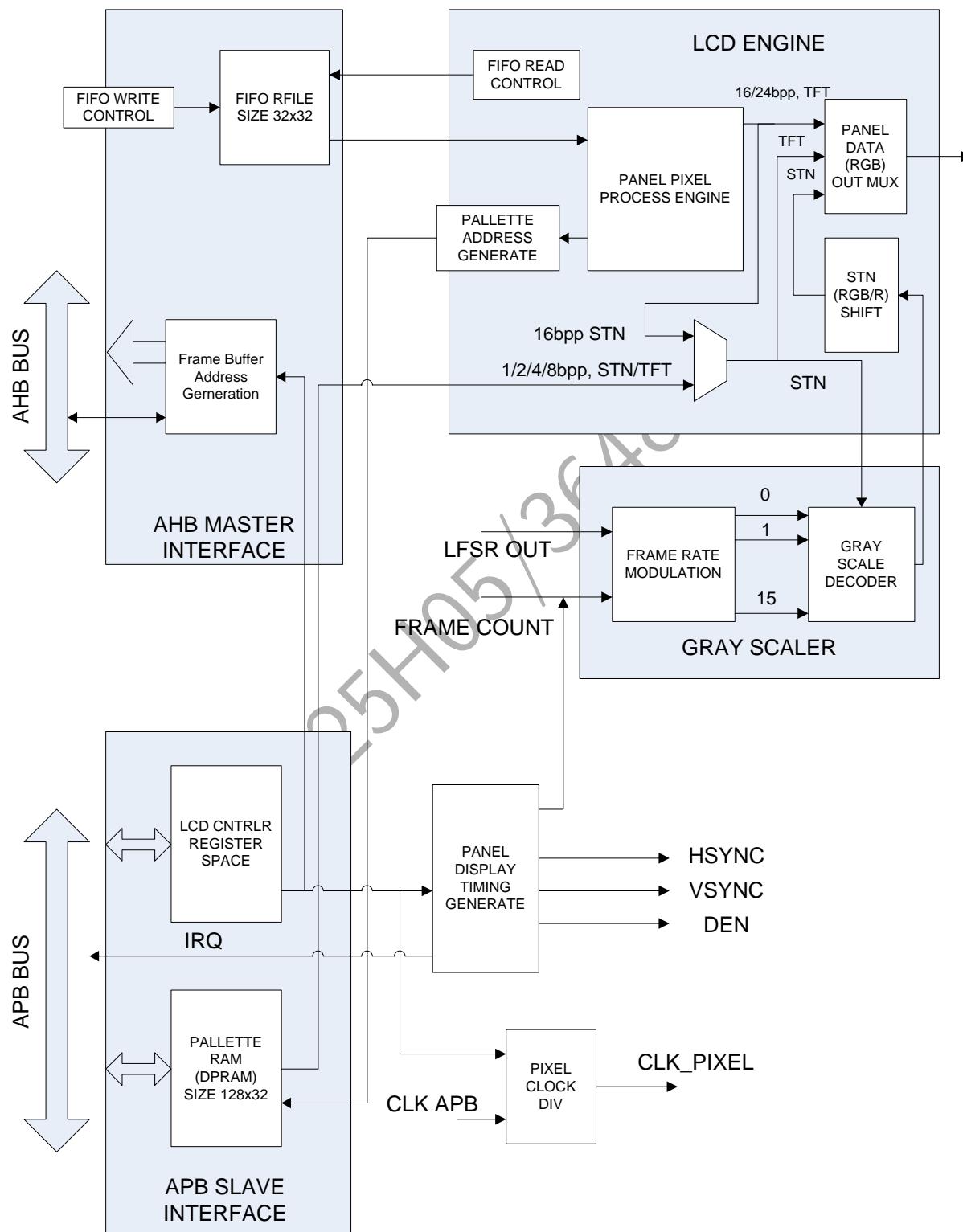
24.2 AHB Master Interface and FIFO Operation

The AMBA AHB Master interface drives FRAME buffer address into AHB bus and initiates a burst transfer of display frame data from the external memory onto the FIFO.

The AMBA master interface performs following functions

- AHB Bus interface handshaking and monitor bus status.
 - Monitors AHB bus status and asserts an interrupt if an error occurs during a data transfer.
- Frame Buffer Memory Address Generation.
 - Loads the upper panel base address into the AMBA AHB address incrementor and generates the addresses in burst for data transfer.
 - If the AHB master state machine detects the situation that the FIFO has enough locations (4, 8 or 16, depending on the burst length setting) to save the data of a burst, it initiates a new transfer, requesting for more data.
- FIFO full, empty conditions, write address generation
 - The AHB master interface based on 'hready' from the AHB bus writes the bus read data into the FIFO in bursts 4, 8 and 16.
 - The input ports of the FIFO are connected to the AMBA AHB interface, clocked at AHB bus clock and the output ports are connected to the pixel processor, clocked at the panel clock pulse.
 - The DMA FIFO is organized as 32 words deep by 32 bits wide dual port RFILE.
 - FIFO empty status is monitored and an interrupt signal indicating an underflow condition is asserted if an attempt is made to read the FIFO when it is empty.

The AHB master interface operates on the AHB bus clock.

**Figure 24-1. Block Diagram of the TFT controller**

24.3 Pixel Process Engine

The pixel process engine unpacks the 32-bit wide LCD data from output ports of the DMA FIFO and streams 1, 2, 4, 8, 16, or 24 bpp data. Depending upon the display mode, the extracted data may be used to access the palette RAM (when bpp is 1, 2, 4, or 8), point to the gray scaler (in the case of STN mode), or may actually be a true color value that can be directly applied to an LCD panel output (in the case of 16/24 bit TFT mode). The following tables show the structure of the data in each DMA FIFO word corresponding to the endianness and bpp combinations. The color LCD controller supports three types of endianness as shown below:

- Little Endian Byte, Little Endian Pixel (LBLP)
- Big Endian Byte, Big Endian Pixel (BBBP)
- Little Endian Byte, Big Endian Pixel (LBBP) (this is the WinCE format)

Data Format LBLP

bpp	DMA FIFO OUPUT BITS																																			
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
1	p31	p30	p29	p28	p27	p26	p25	p24	p23	p22	p21	p20	p19	p18	p17	p16	p15	p14	p13	p12	p11	p10	p9	p8	p7	p6	p5	p4	p3	p2	p1	p0				
2	p15	p14	p13	p12	p11	p10	p9	p8	p7	p6	p5	p4	p3	p2	p1	p0																				
4	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0		
8	p7	p6	p5	p4	p3	p2	p1	p0																												
16	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0				
24	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0				
compact	1 1/3 pixel data																																			



Data Format BBBP

bpp	DMA FIFO OUTPUT BITS																																		
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
1	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15	p16	p17	p18	p19	p20	p21	p22	p23	p24	p25	p26	p27	p28	p29	p30	p31			
2	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12	p13	p14	p15	p16	p17	p18	p19	p20	p21	p22	p23	p24	p25	p13	p14	p15						
4	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	
8	p0	p1	p2	p3	p4	p5	p6	p7																											
16	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1
24	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5
compact	1 1/3 pixel data																																		

Data Format LBBP

bpp	DMA FIFO OUTPUT BITS																																		
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
1	p24	p25	p26	p27	p28	p29	p30	p31	p16	p17	p18	p19	p20	p21	p22	p23	p8	p9	p10	p11	p12	p13	p14	p15	p0	p1	p2	p3	p4	p5	p6	p7			
2	p12	p13	p14	p15	p8	p9	p10	p11	p4	p5	p6	p7	p0	p1	p2	p3																			
4	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	
8	p6	p7	p4	p5	p2	p3	p1	p0																											
16	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1	0	3	2	1
24	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5
compact	1 1/3 pixel data																																		



24.4 AMBA APB Slave Interface and Palette RAM

The AMBA APB slave interface enables CPU to configure the LCD controller registers and the palette RAM. Refer to section 24.10 TFT Controller Registers on page 340 for more information on accessible registers.

The palette RAM is a dual port RAM. One port is used as a READ/WRITE port and is connected to the AMBA APB slave interface. The other port is used as read-only port and is connected to the data formatter and gray scale logic. The palette RAM is 768 bytes in size and structured as a 256 x 24 bits. Each 24-bit contains one color of the palette. This allows two entries to be written into the palette from a single word write access. Palette data is read out of the RAM in a 24-bit word format. The palette RAM is in little endian format. Table 24-1 shows the bit representation of each 24-bit data output.

Table 24-1. Palette RAM Data Format

Bits	Name	Description
[7 : 0]	R [7 : 0]	Red Palette Data
[15 : 8]	G [7 : 0]	Green Palette Data
[23 :16]	B [7 : 0]	Blue Palette Data

The palette data feeds the gray scaler in STN mode or the panel data interface mux in TFT mode when bpp is 1, 2, 4, or 8. The palette RAM is not enabled for color resolutions of 16/24 bpp for both display types STN, TFT. For the case that bpp is 16 or 24, the palette RAM is bypassed and the output of the pixel stream engine is used as the panel data for TFT. For the case that bpp is 16 the display type is STN, palette RAM is bypassed and the output of the pixel stream engine is fed to the gray scale logic. In STN color mode with color resolution 8 or lower bpp the green and blue [4:1] are also used.

24.5 Gray Scale Generation on the STN Panel

STN panels have inherent limitation that they are only capable of displaying on or off state. In order to display gray shades, a gray scale algorithm is implemented. By controlling pixel on and off sequences over several frame periods, with sufficient frame refreshing frequency, human eyes average out the darkness of a pixel so that the individual pixel shows gray. In the case of STN color displays, the three color components (red, green and blue) are gray scaled simultaneously which results in 4096 (16 x 16 x 16) available colors. The gray scaler transforms each 4-bit gray value into a pattern of on and off activities (dither pattern) over several frames. Each gray value corresponds to a temporal dithered pattern with the desired duty cycle (i.e. modulation rate).

The gray scaler has following blocks

24.5.1 Frame Rate Modulator

This block performs the temporal dither pattern. The dither logic contains 4 counters, with modulo 7, 5, 4, and 3. The 4 modulo counters are based on the frame pulse and their outputs are applied to create the dithering patterns over the frames. Furthermore, to randomize the frame based pixel ON/OFF sequence 2bit pseudo-random bits are applied to the frame rate modulation block. Randomization of dither sequence is done to avoid simultaneous switching of pixels leading to flickering, moving lines on the display.



24.5.2 Gray Scale Decoder

The dither values of the sixteen gray levels are shown in Table 24-2.

Table 24-2. Temporal Dither Sequence

GRAY LEVEL	MODULATION RATE (DUTY RATIO)	Time Color Mixture Sequence (without Randomization)
0000	0	0000
0001	1/7	0000001
0010	1/5	00001
0011	1/4	0010
0100	1/3	010
0101	2/5	00101
0110	3/7	0010101
0111	2/4	0101
1000	4/7	1101010
1001	3/5	11010
1010	2/3	101
1011	5/7	1101101
1100	3/4	1101
1101	4/5	1110
1110	6/7	111110
1111	1	1111

24.6 STN Panel Data Output Formatter

In TFT mode, the pixel process engine bypasses the data formatter and passes directly the output obtained from the palette RAM or the output data of the pixel serializer to the external data buses *RED/GREEN/BLUE*.

In STN mode, the LCD controller supports 8-bit output interface. That is, at each pixel-clock, there are 8-bit pixel data available at the output. In color STN mode, only the 8-bit interface is supported. Red, green and blue pixel data bits from the gray scaler are shifted into the respective registers. When enough data is collected, an 8-bit word is constructed by multiplexing the registered data to the correct bit position to satisfy the RGB data pattern of LCD panel. Table 24-3 shows the output 8-bit data output shift register for color STN. R_n (G_n or B_n) denotes the RED (GREEN or BLUE) data of pixel number n .

**Table 24-3. STN Data Output Format**

CLK	Output Data Bits							
	p0	p1	p2	p3	p4	p5	p6	p7
Cycle1	R1	G1	B1	R2	G2	B2	R3	G3
Cycle2	B3	R4	G4	B4	R5	G5	B5	R6
Cycle3	G6	B6	R7	G7	B7	R8	G8	B8
Cycle4							

In STN panel mode, output pins *RED[7:0]* are used for the 8-bit interface. Table 24-4 shows which output pins are used to supply the pixel data for supported operation modes. *DU* and *DL* denotes data for the upper and lower panel respectively.

Table 24-4. LCD Panel Signals

Output Pins	STN Panel	TFT		
		24 bpp	1/2/4/8/16 bpp	16 bit (5:6:5 mode)
RED[7]	DU[0]	Red[7]	Red[4]	Red[4]
RED[6]	DU[1]	Red[6]	Red[3]	Red[3]
RED[5]	DU[2]	Red[5]	Red[2]	Red[2]
RED[4]	DU[3]	Red[4]	Red[1]	Red[1]
RED[3]	DU[4]	Red[3]	Red[0]	Red[0]
RED[2]	DU[5]	Red[2]		
RED[1]	DU[6]	Red[1]		
RED[0]	DU[7]	Red[0]		
GREEN[7]		Green[7]	Green[4]	Green[5]
GREEN[6]		Green[6]	Green[3]	Green[4]
GREEN[5]		Green[5]	Green[2]	Green[3]
GREEN[4]		Green[4]	Green[1]	Green[2]
GREEN[3]		Green[3]	Green[0]	Green[1]
GREEN[2]		Green[2]		Green[0]
GREEN[1]		Green[1]		
GREEN[0]		Green[0]		
BLUE[7]		Blue[7]	Blue[4]	Blue[4]
BLUE[6]		Blue[6]	Blue[3]	Blue[3]
BLUE[5]		Blue[5]	Blue[2]	Blue[2]
BLUE[4]		Blue[4]	Blue[1]	Blue[1]
BLUE[3]		Blue[3]	Blue[0]	Blue[0]
BLUE[2]		Blue[2]	Intensity	
BLUE[1]		Blue[1]		
BLUE[0]		Blue[0]		



24.7 Panel/Pixel Clock Generation

The output of the clock generator block is the panel clock. This is a divided down version of the input APB clock. The clock divide value can be programmed and is stored in the LCD clock control register. Since the color STN displays address 8 segments per pixel or 8/3 pixels are addressed per pixel clock. The pixel clock needs be 1/8 of the system clock and in that case every 3rd system clock pixel data is driven. For calculation of the required pixel clock refer to section Pixel/Panel Clock Frequency Calculation.

Table 24-5. SYSCLK/PIXEL Clock Divider Ratio

DISPLAY TYPE	pixels per clock	Divide Ratio
TFT	1	1(min)
STN(COLOR) 8-bit interface	2 + 2/3	8

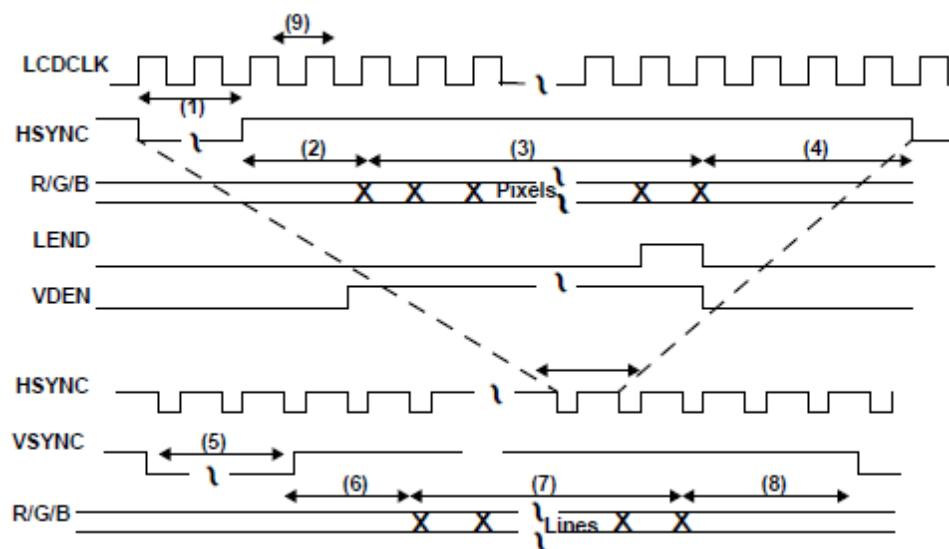
All the logic of the color LCD controller is synchronized with PCLK, Except for the AMBA AHB interface, which is synchronized with HCLK.

24.8 LCD Panel Timing Generation and Settings

The primary function of the LCD timing block is to control the frame buffer reading sequence, synchronize the pixel process engine with the output timing requirement, and generate the horizontal and vertical timing panel signals (HSYNC, VSYNC, LEND, and VDEN).

24.8.1 TFT Mode

Figure 24-2. TFT Mode



- Timing settings:

- | | | |
|-----|-------------|-----------------|
| (1) | HSYNCWIDTH | CLCDHTIM[7:0] |
| (2) | HBACKPORTCH | CLCDHTIM[31:24] |
| (3) | HSIZE | CLCDHTIM[23:16] |
| (4) | HFRONTPORCH | CLCDHTIM[15:8] |



(5)	VSYNCWIDTH	CLCD_VTIM_1[7:0]
(6)	VBACKPORTCH	CLCD_VTIM_0[23:16]
(7)	VLINES	CLCD_VTIM_0[11:0]
(8)	VFRONTPORCH	CLCD_VTIM_1[23:16]
(9)	CLKDIV	CLCD_CLK[7:0]

- Polarity settings:

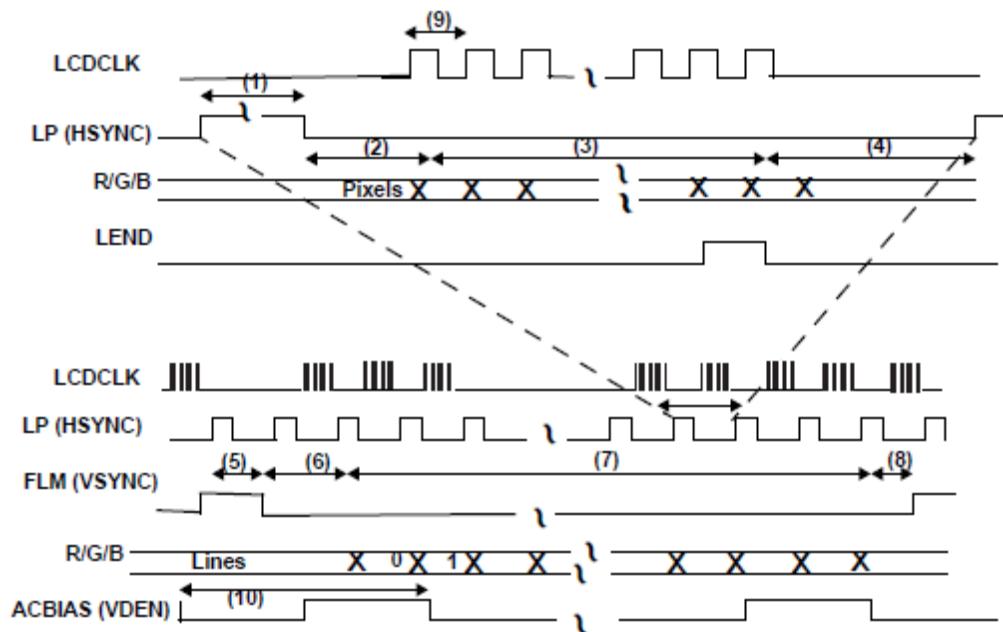
Name	Register	Description
EDGE	CLCD_CLK[19]	LCDCLK edge of latching data. Above diagram shows the case EDGE = '1'
HPOL	CLCD_CLK[18]	HSYNC polarity. Above diagram shows the case HPOL = '1'
VPOL	CLCD_CLK[17]	VSYNC polarity. Above diagram shows the case VPOL = '1'
DPOL (optional)	CLCD_CLK[16]	VDEN polarity. Above diagram shows the case DPOL = '0'
LPOL (optional)	CLCD_CTRL[21]	LEND polarity. Above diagram shows the case LPOL = '0'

- Other TFT-mode settings:

Name	Register	Description
EMODE	CLCD_CTRL[13:12]	Endian Mode. Refer to 23.1
C24 (for BPP=24)	CLCD_CTRL[15]	Compact mode. Refer to 23.1
BPP	CLCD_CTRL[10:8]	BPP count per pixel. Refer to 23.1
DISPTYPE	CLCD_CTRL[7:4]	For TFT, DISPTYPE = 4'b1000
LCDEN	CLCD_CTRL[0]	LCD Controller enabled
FRBUF	CLCD_FRBUF_0[31:0]	Starting address of frame data
IMASK (optional)	CLCD_IMASK[3:0]	Optional. Interrupt enables.
VISEL (optional)	CLCD_CTRL[2:1]	Optional
BRST4(optional)	CLCD_CTRL[20:19]	Optional.
PEN (optional)	CLCD_CTRL[22]	Optional

24.8.2 STN Mode

Figure 24-3. STN Mode



- Timing settings:

(1)	HsyncWidth	CLCD_HTIM[7:0]
(2)	Hbackportch	CLCD_HTIM[31:24]
(3)	Hsize	CLCD_HTIM[23:16]
(4)	Hfrontporch	CLCD_HTIM[15:8]
(5)	VsyncWidth	CLCD_VTIM_1[7:0]
(6)	Vbackportch	CLCD_VTIM_0[23:16]
(7)	Vlines	CLCD_VTIM_0[11:0]
(8)	Vfrontporch	CLCD_VTIM_1[23:16]
(9)	ClkDiv	CLCD_CLK[7:0]
(10)	ACB	CLCD_CLK[15:8]

- Polarity settings:

Name	Register	Description
EDGE	CLCD_CLK[19]	LCDCLK edge. Usually '1' for STN (refer to above diagram)
HPOL	CLCD_CLK[18]	Hsync polarity. Usually '0' for STN (refer to above diagram)
VPOL	CLCD_CLK[17]	Vsync polarity. Usually '0' for STN (refer to above diagram)
LPOL (optional)	CLCD_CTRL[21]	LEND polarity. Above diagram shows the case LPOL = '0'

- Other STN-mode settings:

Name	Register	Description
PASCLK	CLCD_CLK[20]	Set to '1' for STN
EMODE	CLCD_CTRL[13:12]	Endian Mode. Refer to 23.1
BPP	CLCD_CTRL[10:8]	BPP count per pixel. For STN, only BPP1, 2, 4, 8, 16 are valid. Refer to 23.1
DISPTYPE	CLCD_CTRL[7:4]	For STN, DISPTYPE is 4'b0000~4'b0101
LCDEN	CLCD_CTRL[0]	LCD Controller enabled



FRBUF	CLCD_FRBUF_0[31:0]	Starting address of frame data (Upper and Lower panels)
IMASK (optional)	CLCD_IMASK[3:0]	Optional. Interrupt enables.
VISEL (optional)	CLCD_CTRL[2:1]	Optional
BRST4(optional)	CLCD_CTRL[20:19]	Optional.

24.9 Interrupt Operation

The LCD controller provides four individually maskable interrupts:

- DMA FIFO underflow: an underflow interrupt is asserted if an attempt is made to read the frame buffer FIFO when it is empty
- Address ready signification: an interrupt is set when the current base address registers have been loaded to the AMBA AHB master state machine, signifying that a new next address can be loaded to the registers.
- Vertical status: an interrupt is asserted when the specified vertical state is reached. The vertical state is selected via the LCD control register.
- Bus error: an interrupt is asserted if an error occurs during an AHB data transfer

24.10 TFT Controller Registers

Address assignments for the LCD Interface registers are outlined in . Reserved register bits should only be written as 0. Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.

Table 24-6. LCD Interface Register Address (Base ADDR = 0x4003_1000)

Offset	Access	Register	Description
0x0000	RW	CLCD_CLK	LCD Clock Register
0x0004	RW	CLCD_VTIM_0	LCD Vertical Timing 0 Register
0x0008	RW	CLCD_VTIM_1	LCD Vertical Timing 1 Register
0x000C	RW	CLCD_HTIM	LCD Horizontal Timing Register
0x0010	RW	CLCD_CTRL	LCD Control Register
0x0018	RW	CLCD_FRBUF_0	LCD Frame Buffer 0 Register
0x001C	RW	CLCD_FRBUF_1	LCD Frame Buffer 1 Register
0x0020	RW	CLCD_INT_EN	LCD Interrupt Enable Register
0x0024	RW	CLCD_STAT	LCD Status Register
0x0030	RW	CLCD_HV_PHASE	LCD PHASE (between HSYNC and VSYNC) Register
0x0400 –0x07FC	RW	CLCD_PALETTE_RAM	LCD Palette RAM



24.10.1 LCD Clock Register (CLCD_CLK, Offset 0x0000)

Table 24-7. CLCD_CLK (Offset 0x0000)

Position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	RFU[31:21]															PAS CLK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ACB															CLKDIV
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	RW												

Name	Bits	Description	Settings
CLKDIV	7:0	LCD Clock Divide Value. Pixel clock frequency is PClk divided by (CLKDIV +1)	0: divide by 1 1: divide by 2 2: divide by 3 ... 255: divide by 256
ACB	15:8	AC Bias frequency. AC bias pin toggles when the specified ACB +1 number of line clocks are applied (this field is ignored in TFT mode)	0: 1 Line 1: 2 Lines 2: 3 Lines ... 256: 256Lines
DPOL	16	Video Enable polarity	0 High for data output enable 1 Low for data output enable
VPOL	17	Vertical Sync Polarity	0 High when vertical period 1 Low when vertical period (data period)
HPOL	18	Horizontal Sync Polarity	0 Low when horizontal period (data period) 1 High when horizontal period
EDGE	19	Clock Edge Select	0 Positive edge 1 Negative edge
PAS CLK	20	Passive Clock (for STN, set to 0 for TFT)	0: LCD clock always active 1: LCD clock only active during data



Name	Bits	Description	Settings
RFU	31:21	Reserved	N/A

E/UG25H05/36488764



24.10.2LCD Vertical Timing Register 0 (CLCD_VTIM_0, Offset 0x0004)

Table 24-8. CLCD_VTIM_0 (Offset 0x0004)

Position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	RFU[31:24]														VBACKPORCH	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	RW	RW						

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RFU[15:12]														VLINES	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	RW	RW										

Name	Bits	Description	Settings
VLINES	11:0	Vertical Lines	0: 1 line 1: 2 lines 2: 3 lines 4095: 4096 lines
RFU	15:12	Reserved	N/A
VBACKPORCH	23:16	Vertical back porch lines	0: 0 lines 1: 1 line 2: 2 lines 255: 255 lines
RFU	31:24	Reserved	N/A



24.10.3LCD Vertical Timing Register 1 (CLCD_VTIM_1, Offset 0x0008)

Table 24-9. CLCD_VTIM_1 (Offset 0x0008)

Position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	RFU[31:24]															VFRONTPORCH
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	R	R	R

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RFU[15:8]															VSYNCWIDTH
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	RW							

Name	Bits	Description	Settings
VSYNCWIDTH	7:0	Vertical Sync Width	0: 1 Line 1: 2 Lines 2: 3 Lines 255: 256 Lines
RFU	15:8	Reserved	N/A
VFRONTPORCH	23:16	Vertical Front Porch Lines	0: 0 lines 1: 1 lines 2: 2 lines 255: 255 lines
RFU	31:24	Reserved	N/A



24.10.4 LCD Horizontal Timing Register (CLCD_HTIM, Offset 0x000C)

Table 24-10. CLCD_HTIM (Offset 0x000C)

Position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	HBACKPORCH										HSIZE					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	HFRONTPORCH										HSYNCWIDTH					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
HSYNCWIDTH	7:0	Horizontal Sync Width	0: 1 Panel clock 1: 2 Panel clock 2: 3 Panel clocks ... 255: 256 Panel clocks
HFRONTPORCH	15:8	Horizontal front porch in LCD clocks	0: 1 clock 1: 2 clocks 2: 3 clocks ... 255: 256 clocks
HSIZE	23:16	Horizontal Size (Pixels)	0: 16 pixels 1: 32 pixels 2: 48 pixels ... 255: 4096 pixels
*HBACKPORCH	31:24	Horizontal back porch in LCD clocks	0: 1 clock 1: 2 clocks 2: 3 clocks ... 255: 256 clocks

Note: There is a restriction on HBACKPORCH. Some time is needed to propagate frame data from the buffers to the LCD interface. The minimum required time length of horizontal back porch is 4 PClk cycles. Therefore, the minimum allowed HBACKPORCH value also depends on CLKDIV (refer to LCD_CLK). If LCD clock frequency equals PClk (CLKDIV = 0), HBACKPORCH should be larger than or equal to 3. Otherwise, the minimum HBACKPORCH is the smallest integer that is larger than $4/(CLKDIV+1)-1$.



24.10.5LCD Control Register (CLCD_CTRL, Offset 0x0010)

Table 24-11. CLCD_CTRL (Offset 0x0010)

Position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	RFU[31:23]							PEN	LPOL	BURST		RFU[18:16]				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	RW	RW	RW	R	R	R	R

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	C24	RFU	EMODE		MODE 565	BPP			DISPTYPE			RFU	VISEL		LCDEN	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	RW	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	RW	RW	RW

Name	Bits	Description	Settings
LCDEN	0	LCD Enable	0: LCD is disabled 1: LCD is enabled
VISEL	2:1	Vertical Compare Interrupt select	00: VCI on start of vertical sync 01: VCI on start of vertical back porch 10: VCI on start of active video 11: VCI on start of vertical front porch
RFU	3	Reserved	N/A
DISPTYPE	7:4	Display Type	0000: Reserved 0001: Reserved 0010: Reserved 0011: Reserved 0100: STN Color 8 Bit 0101: Reserved 1000: TFT 1001-1111: Reserved
BPP	10	Bits per pixel	000: 1 bpp 001: 2 bpp 010: 4 bpp 011: 8 bpp 100: 16 bpp 101: 24 bpp
MODE565	11	5:6:5 Mode Select	0: BGR 5:5:6 Mode Select 1: RGB 5:6:5 Mode Select



Name	Bits	Description	Settings
EMODE	13:12	Endian Mode	00: LLBP 01: BBBP 10: LBBP 11: Reserved
RFU	14	Reserved	N/A
C24	15	Compact 24 bit. This bit changes the TFT 24 bit color mode to compact display data storage. When set to one, each word contains 1 1/3 pixels of information. When 0 only the three bytes of the word are used for LCD data.	0: 1 pixel per word 1: 1 1/3 pixels per word
RFU	18:16	Reserved	N/A
BURST	20:19	Threshold and Burst Size Select	FIFO burst size 00: 4 words 01: 8 words 1x:16 words
LPOL	21	LEND (Line End) Polarity	0: Active high 1: Active low
PEN	22	Power Enable to the panel	0: PWREN pin = 0 1: PWREN pin = 1
RFU	31:23	Reserved	N/A

Note: To use the 5:6:5 mode, TFT and 16 bpp should be selected in DISPTYPE and BPP fields respectively.

**24.10.6 LCD Frame Buffer 0 Address Register (CLCD_FRBUF_0, Offset 0x0018)****Table 24-12. CLCD_FRBUF_0 (Offset 0x0018)**

Position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	ADDR0[31:16]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ADDR0[15:0]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
ADDR0	31:0	frame Buffer Address This address points to the start of frame Buffer 0. This address is always word aligned and bits 1:0 are ignored	Address of frame 0



24.10.7 LCD Frame Buffer 1 Address Register (CLCD_FRBUF_1, Offset 0x001C)

Table 24-13. CLCD_FRBUF_1 (Offset 0x001C)

Position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	ADDR1[31:16]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	ADDR1[15:0]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
ADDR1	31:0	frame Buffer Address This address points to the start of frame Buffer 1. This address is always word aligned and bits 1:0 are ignored	Address of frame 1



24.10.8LCD Interrupt Enable Register (CLCD_INT_EN, Offset 0x0020)

Table 24-14. CLCD_INT_EN (Offset 0x0020)

Position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	RFU[31:16]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RFU[15:4]												BERR	VCI	ADRRDY	UFLO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	RW	RW	RW	RW

Name	Bits	Description	Settings
UFLO	0	FIFO underflow interrupt enable	0: interrupt is disabled 1: interrupt is enabled
ADRRDY	1	Address ready interrupt enable	0: interrupt disabled 1: interrupt enabled
VCI	2	Vertical control interrupt enable	0: VCI interrupt disabled 1: VCI interrupt enabled
BERR	3	Bus Error interrupt enable	0: Bus Error interrupt disabled 1: Bus Error interrupt enabled
RFU	31:4	Reserved	N/A



24.10.9 LCD Interrupt Status Register (CLCD_STAT, Offset 0x0024)

Table 24-15. CLCD_STAT (Offset 0x0024)

Position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	RFU[31:16]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RFU[15:9]								LCD IDLE	RFU[7:4]				BERR	VCI	ADRRDY
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	RW1C	RW1C	RW1C	RW1C

Name	Bits	Description	Settings
UFLO	0	FIFO underflow interrupt	0: No FIFO underflow interrupt 1: FIFO underflow occurred
ADRRDY	1	Address ready interrupt	0: Base address register has not been read 1: Base address register has been read
VCI	2	Vertical control interrupt	0: Vertical state has not been reach 1: Specify vertical state has been reached
BERR	3	Bus Error interrupt	0: Bus Error not occurred 1: Bus Error occurred
RFU	7:4	Reserved	N/A
LCD IDLE	8	LCD Idle. This bit can be read after the LCD enable bit is cleared to determine when the current frame is complete.	0: LCD Running 1: LCD Idle
RFU	31:9	Reserved	N/A

24.10.10 LCD HSYNC-VSYNC Phase Difference Register (CLCD_HV_PHASE, Offset 0x0030)

Table 24-16. CLCD_HV_PHASE (Offset 0x0030)

Position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	RFU[31:16]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RFU[15:8]								THV[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	R	R	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
THV	7:0	THV: HSYNC-VSYNC Phase Difference in number of pixel clock	0: 0 pixel clock 1: 1 pixel clock 255: 255 pixel clock
RFU	31:8	Reserved	N/A



24.10.11 LCD Palette Register (CLCD_PALETTE_RAM, Offset 0x0400 to 0x07FC)

Table 24-17. CLCD_PALETTE_RAM (Offset 0x0400 to 0x07FC)

Position	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	RFU[31:24]														BLUE[7:0]	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	R	R	R	R	R	R	RW	RW								

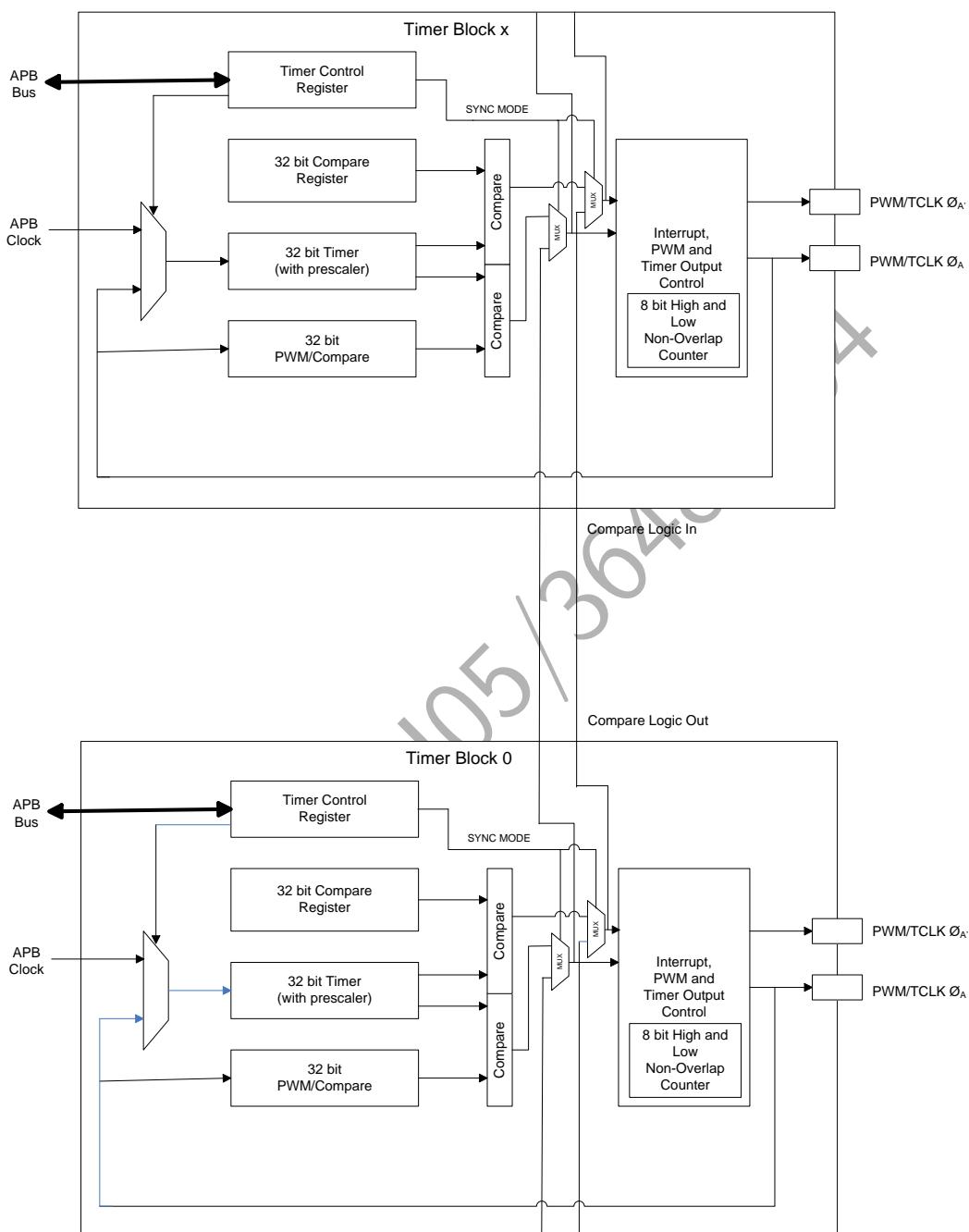
Position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	GREEN[7:0]								RED[7:0]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
RED	7:0	Red Data	
GREEN	15:8	Green Data	
BLUE	23:16	Blue Data	
RFU	31:24	Reserved	N/A

25.0 Timers

There are six 32-bit reloadable timers that can be used for timing and event counting, four of which may be used for generation of pulse-width modulated (PWM) signals. The timers' features include:

- 32-bit reload counter
- Programmable prescaler with prescale values from 1 to 4096
- Non-overlapping Hi-Drive (8mA) PWM output generation with configurable off-time
- Capture, compare and capture/compare capability
- External input pin for timer input, clock gating, or capture signal.
- Timer output pin
- Timer interrupt

**Figure 25-1. 32 bit Timer Block Diagram**



25.1 Basic Operation

The timers are 32-bit up-counter timers. Minimum time-out delay is set by:

- Loading the value 0x0000_0001 into the Timer Compare register
- Setting the prescale value to 1

Maximum time-out delay is set by:

- Loading the value 0x0000_0000 into the Timer Compare register
- Setting the prescale value to 4096.

If the Timer reaches 0xFFFF_FFFF, the timer rolls over to 0x0000_0000 and continues counting.

The current count value in the timers can be read while the timer is counting (TEN = '1'). This action has no effect on the timers operation.

As a general rule, the timer output is toggled every time the counter is reloaded.

25.2 Timer Operating Modes

The timers can be configured to operate in one of eight modes. Refer to for more information regarding the Timer Control register.

The Timer operating modes include the following:

- One-shot Mode
- Continuous Mode
- Counter Mode
- PWM Mode
- Capture Mode
- Compare Mode
- Gated Mode
- Capture/Compare Mode
- PWM Differential Mode

25.2.1 One Shot Mode

In One-Shot mode, the timer counts from the start count value stored in the Timer register up to the 32-bit Compare value stored in the Timer Compare register. The timer input is the PCLK. Upon reaching the Compare value, the timer generates an interrupt, and the count value in the Timer register is reset to 0x0000_0001. Then, the timer is automatically disabled and stops counting. Also, if the Timer Output function is enabled in the GPIO, the Timer output pin will change state for one clock cycle, then return to polarity value (TPOL bit in the Timer Control Register). A software workaround is to set the polarity to not (TPOL) during the counting phase.

The steps for configuring a timer for One-Shot mode and initiating the count are as follows:

1. Write to the Timer Control register to:
 - Disable the timer (TEN = "0").
 - Configure the timer for One-Shot mode (TMODE = "000").

- Set the prescale value (PRES).
 - If using the Timer Output function, set the initial output level (High or Low) via TPOL.
2. Write to the Timer register to set the starting count value.
 3. Write to the Timer Compare register to set the Compare value.
 4. If desired, enable the timer interrupt in the Interrupt Controller and set the timer interrupt priority by writing to the appropriate Configuration Table Register in the Interrupt Controller.
 5. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output function. See chapter covering GPIO.
 6. Write to the Timer Control register to enable the timer and initiate counting (TEN = “1”).
 7. Negate the TPOL bit during the counting phase.

In One-Shot mode, the PCLK always provides the timer input. The timer period is given by the following equation:

$$\text{Oneshot Timeout Period (s)} = \frac{\text{Reload} - \text{Start Value}}{\text{System clock Frequency (Hz)}} \times \text{Prescale}$$

25.2.2 Continuous Mode

In Continuous mode, the timer counts up to the 32-bit Compare value stored in the Timer Compare register. The timer input is the PCLK. Upon reaching the Compare value, the timer generates an interrupt; the count value in the Timer register is reset to 0x0000_0001 and counting resumes. Also, if the Timer Output function is enabled in the GPIO, the Timer Output pin changes state from Low to High or from High to Low upon Timer Compare.

The steps for configuring a timer for Continuous mode and initiating the count are as follows:

1. Write to the Timer Control register to:
 - Disable the timer (TEN = “0”).
 - Configure the timer for Continuous mode (TMODE = “001”).
 - Set the prescale value (PRES).
 - If using the Timer Output function, set the initial output level (High or Low) via TPOL.
2. Write to the Timer register to set the starting count value. This only affects the first pass in Continuous mode. After the first Timer Compare in Continuous mode, counting always begins at the reset value of 0x0000_0001.
3. Write to the Timer Compare register to set the Compare value.
4. If desired, enable the timer interrupt in the Interrupt Controller and set the timer interrupt priority by writing to the appropriate Configuration Table Register in the Interrupt Controller.
5. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output function. See chapter covering GPIO.
6. Write to the Timer Control register to enable the timer and initiate counting (TEN = “1”).

In Continuous mode, the PCLK always provides the timer input. The timer period is given by the following equation:

$$\text{Continuous Timeout Period (s)} = \frac{\text{Reload Value}}{\text{System clock Frequency (Hz)}} \times \text{Prescale}$$



If an initial starting value other than 0x0000_0001 is loaded into the Timer register, the One-Shot mode equation must be used to determine the first time-out period.

25.2.3 Counter Mode

In Counter mode, the timer counts input transitions from a GPIO port pin. The timer input is taken from the GPIO Port pin Timer Input function. The TPOL bit in the Timer Control Register selects whether the count occurs on the rising edge (TPOL = “0”) or the falling edge (TPOL = “1”) of the Timer Input signal. In Counter mode, the prescaler is disabled. Any value assigned to PRES in Counter Mode will have no effect. The input frequency of the Timer Input signal must not exceed one-fourth the PCLK frequency.

Upon reaching the Compare value stored in the Timer Compare register, the timer generates an interrupt, the count value in the Timer register is reset to 0x0000_0001 and counting resumes. Also, if the Timer Output function is enabled in the GPIO, the Timer Output pin changes state from Low to High or from High to Low at Timer Compare.

The steps for configuring a timer for Counter mode and initiating the count are as follows:

1. Write to the Timer Control register to:
 - Disable the timer (TEN = “0”).
 - Configure the timer for Counter mode (TMODE = “010”).
 - Select either the rising edge or falling edge of the Timer Input signal for the count (TPOL). This also sets the initial logic level (High or Low) for the Timer Output function. However, the Timer Output function does not have to be enabled (via GPIO).
2. Write to the Timer register to set the starting count value. This only affects the first pass in Counter mode. After the first Timer Compare in Counter mode, counting always begins at the reset value of 0x0000_0001. Generally, in Counter mode, the Timer register should be written with the value 0x0000_0001.
3. Write to the Timer Compare register to set the Compare value.
4. If desired, enable the timer interrupt (via the Interrupt Mask Register in the Interrupt Controller) and set the timer interrupt priority (by writing to the appropriate Configuration Table Register in the Interrupt Controller herein).
5. Configure the associated GPIO port pin for the Timer Input function.
6. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output function. See chapter covering GPIO.
7. Write to the Timer Control register to enable the timer (TEN = “1”).

In Counter mode, the number of Timer Input transitions since the timer start is given by the following equation:

$$\text{Counter Mode Timer Input Transition} = \text{Current Count value} - \text{Start Value}$$

25.2.4 PWM Mode

In PWM mode, the timer outputs a Pulse-Width Modulator (PWM) output signal through a GPIO Port pin. The timer input is the PCLK. The timer first counts up to the 32-bit PWM match value stored in the Timer PWM register. When the timer count value matches the PWM value, the Timer Output toggles. The timer continues counting until it reaches the Compare value stored in the Timer Compare register. Upon reaching the Compare value, the Timer Output signal toggles again, the timer generates an interrupt, the count value in the Timer register is reset to 0x0001 and counting resumes.

When TPOL = “0” (in TMRx_CTRL Register):

Timer Output signal begins as a Low (“0”) and then transitions to a High (“1”) when the timer value matches the PWM value. The Timer Output signal returns to a Low (“0”) after the timer reaches the Compare value, which is reset to 0x0000_0001.

When TPOL = “1” (in TMRx_CTRL Register):

Timer Output signal begins as a High (“1”) and then transitions to a Low (“0”) when the timer value matches the PWM value. The Timer Output signal returns to a High (“1”) after the timer reaches the Compare value, which is reset to 0x0000_0001.

The steps for configuring a timer for PWM mode and initiating the PWM operation are as follows:

1. Write to the Timer Control register to:
 - Disable the timer (TEN = “0”).
 - Configure the timer for PWM mode (TMODE = “011”).
 - Set the prescale value (PRES).
 - Set the initial logic level (High or Low) and PWM High/Low transition for the Timer Output function (TPOL).
2. Write to the Timer register to set the starting count value (typically 0x0000_0001). This only affects the first pass in PWM mode. After the first timer reset in PWM mode, counting always begins at the reset value of 0x0000_0001.
3. Write to the PWM register to set the Match value.
4. Write to the Timer Compare register to set the Compare value. The Compare value must be greater than the PWM value.
5. If desired, enable the timer interrupt (via the Interrupt Mask Register in the Interrupt Controller) and set the timer interrupt priority (by writing to the appropriate Configuration Table Register in the Interrupt Controller herein).
6. Configure the associated GPIO port pin for the Timer Output function. (See chapter covering GPIO herein.)
7. Write to the Timer Control register to enable the timer and initiate counting (TEN = “1”).

The PWM period is given by the following equation:

$$\text{PWM Period (s)} = \frac{\text{Reload Value}}{\text{System clock Frequency (Hz)}} \times \text{Prescale}$$

If an initial starting value other than 0x0000_0001 is loaded into the Timer register, the One-Shot mode equation must be used to determine the first PWM time-out period.

If TPOL is set to “0”, the ratio of the PWM output High time to the total period is given by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{Reload Value} - \text{PWM Value}}{\text{Reload Value}} \times 100$$

If TPOL is set to “1”, the ratio of the PWM output High time to the total period is given by:

$$\text{PWM Output High Time Ratio (\%)} = \frac{\text{PWM Value}}{\text{Reload Value}} \times 100$$

25.2.5 Capture Mode

In Capture mode, the current timer count value is recorded when the desired external Timer Input transition occurs. The Capture count value is written to the Timer PWM register. The timer input is the PCLK. The TPOL bit in the Timer Control register determines if the Capture occurs on a rising edge (TPOL = “0”) or a falling edge (TPOL =



“1”) of the Timer Input signal. When the Capture event occurs, an interrupt is generated, and the timer continues counting.

The timer continues counting up to the 32-bit Compare value stored in the Timer Compare register. Upon reaching the Compare value, the timer generates an interrupt, the count value in the Timer register is reset to 0x0000_0001, and counting resumes.

The steps for configuring a timer for Capture mode and initiating the count are as follows:

1. Write to the Timer Control register to:
 - Disable the timer (TEN = “0”).
 - Configure the timer for Capture mode (TMODE = “100”).
 - Set the prescale value (PRES).
 - Set the Capture edge (rising or falling) for the Timer Input (TPOL).
2. Write to the Timer register to set the starting count value (typically 0x0000_0001).
3. Write to the Timer Compare register to set the Compare value.
4. If desired, enable the timer interrupt (via the Interrupt Mask Register in the Interrupt Controller) and set the timer interrupt priority (by writing to the appropriate Configuration Table Register in the Interrupt Controller herein).
5. Configure the associated GPIO port pin for the Timer Input function.
6. Write to the Timer Control register to enable the timer and initiate counting (TEN = “1”).

In Capture mode, the elapsed time from timer start to Capture event can be calculated using the following equation:

$$\text{Capture Elapsed Time (s)} = \frac{\text{Capture Value} - \text{Start Value}}{\text{System Clock Frequency (Hz)}} \times \text{Prescale}$$

25.2.6 Compare Mode

In Compare mode, the timer counts up to the 32-bit maximum Compare value stored in the Timer Compare register. The timer input is the PCLK. Upon reaching the Compare value, the timer generates an interrupt and counting continues (the timer value is not reset to 0x0000_0001). Also, if the Timer Output function is enabled (in the GPIO), the Timer Output pin changes state (from Low to High or from High to Low) upon Compare.

When the Timer reaches 0xFFFF_FFFF, the timer rolls over to 0x0000_0000 and continues counting.

The steps for configuring a timer for Compare mode and initiating the count are as follows:

1. Write to the Timer Control register to:
 - Disable the timer (TEN = “0”).
 - Configure the timer for Compare mode (TMODE = “101”).
 - Set the prescale value (PRES).
 - If using the Timer Output function, set the initial logic level (High or Low) via TPOL.
2. Write to the Timer register to set the starting count value.
3. Write to the Timer Compare register to set the Compare value.



4. If desired, enable the timer interrupt (via the Interrupt Mask Register in the Interrupt Controller) and set the timer interrupt priority (by writing to the appropriate Configuration Table Register in the Interrupt Controller herein).
5. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output function. (See chapter covering GPIO herein.)
6. Write to the Timer Control register to enable the timer and initiate counting (TEN = “1”).

In Compare mode, the PCLK always provides the timer input. The Compare time is given by the following equation:

$$\text{Compare Mode Time (s)} = \frac{\text{Compare Value} - \text{Start Value}}{\text{System Clock Frequency (Hz)}} \times \text{Prescale}$$



25.2.7 Gated Mode

In Gated mode, the timer counts only when the Timer Input signal is in its active state (asserted), as determined by the TPOL bit in the Timer Control register. When the Timer Input signal is asserted, counting begins. A timer interrupt is generated when the Timer Input signal is de-asserted or a Timer Compare occurs.

The timer counts up to the 32-bit Compare value stored in the Timer Compare register. The timer input is the PCLK. When reaching the Compare value, the timer generates an interrupt, the count value in the Timer register is reset to 0x0000_0001 and counting resumes (assuming the Timer Input signal is still asserted). Also, if the Timer Output function is enabled (in the GPIO), the Timer Output pin changes state (from Low to High or from High to Low) at Timer Compare.

The steps for configuring a timer for Gated mode and initiating the count are as follows:

1. Write to the Timer Control register to:
 - Disable the timer (TEN = “0”).
 - Configure the timer for Gated mode (TMODE = “110”).
 - Set the prescale value (PRES).
2. Write to the Timer register to set the starting count value. This only affects the first pass in Gated mode. After the first timer reset in Gated mode, counting always begins at the reset value of 0x0000_0001.
3. Write to the Timer Compare register to set the Compare value.
4. If desired, enable the timer interrupt (via the Interrupt Mask Register in the Interrupt Controller) and set the timer interrupt priority (by writing to the appropriate Configuration Table Register in the Interrupt Controller herein).
5. Configure the associated GPIO port pin for the Timer Input function.
6. If using the Timer Output function, configure the associated GPIO port pin for the Timer Output function. (See chapter covering GPIO herein.)
7. Write to the Timer Control register to enable the timer (TEN = “1”).
8. Assert the Timer Input signal to initiate the counting (High if TPOL = “0”, Low if TPOL = “1”).

25.2.8 Capture/Compare Mode

In Capture/Compare mode, the timer begins counting after the first desired external Timer Input transition occurs. The desired transition (rising edge or falling edge) is set by the TPOL bit in the Timer Control Register. The timer input is the PCLK.

Every subsequent desired transition (after the first) of the Timer Input signal captures the current count value. The Captured timer value is written to the Timer PWM register. When the Capture event occurs, an interrupt is generated, the count value in the Timer register is reset to 0x0001, and counting resumes.

If no Capture event occurs, the timer counts up to the 32-bit Compare value stored in the Timer Compare register. Upon reaching the Compare value, the timer generates an interrupt, the count value in the Timer register is reset to 0x0000_0001, and counting resumes.

The steps for configuring a timer for Capture/Compare mode and initiating the count are as follows:

1. Write to the Timer Control register to:
 - Disable the timer (TEN = “0”).

- Configure the timer for Capture/Compare mode (TMODE = “111”).
 - Set the prescale value (PRES).
 - Set the Capture edge (TPOL = “0” for rising; TPOL = “1” for falling) for the Timer Input.
2. Write to the Timer register to set the starting count value (typically 0x0001).
 3. Write to the Timer Compare register to set the Compare value.
 4. If desired, enable the timer interrupt (via the Interrupt Mask Register in the Interrupt Controller) and set the timer interrupt priority (by writing to the appropriate Configuration Table Register in the Interrupt Controller herein).
 5. Configure the associated GPIO port pin for the Timer Input function.
 6. Write to the Timer Control register to enable the timer (TEN = “1”).
 7. Counting begins after the first desired transition of the Timer Input signal. No interrupt is generated by this first edge.

In Capture/Compare mode, the elapsed time from timer start to Capture event can be calculated using the following equation:

$$\text{Capture Elapsed Time (s)} = \frac{\text{Capture Value} - \text{Start Value}}{\text{System Clock Frequency (Hz)}} \times \text{Prescale}$$



25.2.9 PWM Differential Mode

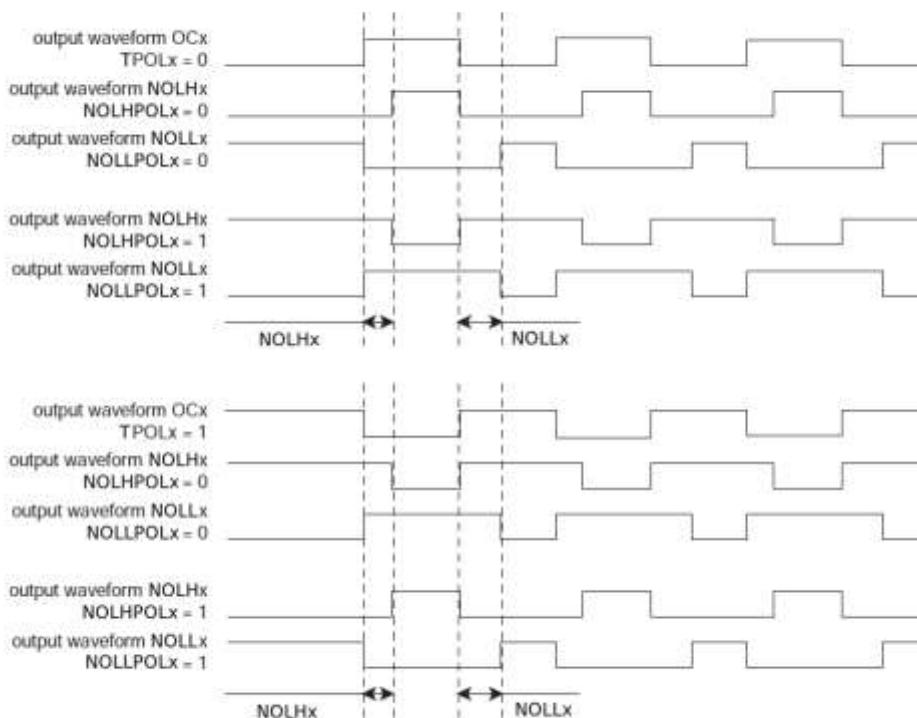
In Differential Mode, an additional PWM output $\emptyset_{A'}$ (phase A prime) is available. \emptyset_A functions as a differential or complementary output to the primary PWM output (\emptyset_A). By default, $\emptyset_{A'}$ will behave as an exact inverse. For non-overlapping control of power transistors and other complementary switching applications, an 8-bit Non-Overlapping High and Low counter is featured to configure the differential output phases. This counter adjusts the 2 two “dead time” phases from PWM output $\emptyset_{A'}$ falling edge to \emptyset_A rising edge using the Non-Overlapping High Compare register (NOLH CMP) and vice-versa for the Non-Overlapping Low Compare register (NOLL CMP). This scheme provides independent control of these two dead phases. In PWM mode, each output has an additional independent polarity control bit (TMRCTRL.10 or NOLHPOL and TMRCTRL.11 or NOLLPOL) for additional waveform configuration which can result in some redundancies with the primary polarity bit (TMRCTRL.6 or TPOL). Figure 25-2 Timer Output Complementary Waveforms shows the effect of TPOL, NOLHPOL and NOLLPOL bits on the output waveforms.

Finally, a timer PWM Synchronization Mode bit is available (TMRCTRL.9 or PWMSYNC) that multiplexes a primary timer’s main counter to other secondary (or slave) timers. This attribute will synchronize the timers’ periods, primary polarity polarities and duty-cycles to fix a base PWM waveform depicted as OCx in Figure 25-2 Timer Output Complementary Waveforms. However, the Non-Overlapping registers are independent and configurable for each of the synchronized timers, regardless of the PWMSYNC bit. This provides independent adjustments to several sets of complementary switches.

The device is set up to have two sets of complementary outputs.

- For Timer 0 configured in PWM Differential Mode with PWMSYNC=1, $\emptyset_A = \text{TCLK}0$, $\emptyset_{A'} = \text{TCLK}1$.
- For Timer 2 configured in PWM Differential Mode with PWMSYNC=1, $\emptyset_A = \text{TCLK}2$, $\emptyset_{A'} = \text{TCLK}3$.

Figure 25-2 Timer Output Complementary Waveforms



25.3 Timer Registers

Address assignments for Timer registers within each block are outlined in Table 2-2. Peripheral Bus Region. Reserved register bits should only be written as 0.

Each Timer instance is controlled by a block of registers assigned to that port; all blocks contain identical registers for control, interrupt etc. Address assignments for the Timer registers are outlined in Table 25-1. Timer Register Address Offset (For Each Instance). Reserved register bits should only be written as 0.

Register names for different instances can be defined by appending the instance number to the peripheral name, so for instance, the Timer Count Register for Timer 0 would be TMR0_CNT, while the Timer Count Register for Timer 1 would be TMR1_CNT, and so on.

Table 25-1. Timer Register Address Offset (For Each Instance)

Offset	Access	Register	Description
0x0000	RW	TMR_CNT	Timer Count Register
0x0004	RW	TMR_CMP	Timer Compare Register
0x0008	RW	TMR_PWM	Timer PWM Register
0x000C	RW	TMR_INT	Timer Interrupt Register
0x0010	RW	TMR_CN	Timer Control Register
0x0014	RW	TMR_NOLCMP	Timer Non-Overlapping Compare Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



25.3.1 Timer Count Register (TMR_CNT, Offset 0x0000)

Table 25-2. TMR_CNT (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	CNT[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	CNT[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	CNT[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	CNT[7:0]							
Reset	0000 0001							
Access	RW							

Name	Bits	Description	Settings
CNT	31:0	Count. This register stores the current timer count.	



25.3.2 Timer Compare Register (TMR_CMP, Offset 0x0004)

Table 25-3. TMR_CMP (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	CMP[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	CMP[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	CMP[15:8]							
Reset	1111 1111							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	CMP[7:0]							
Reset	1111 1111							
Access	RW							
Name	Bits	Description					Settings	
CMP	31:0	Compare. This register stores the compare value, which is used to set the maximum count value to initiate a reload of the timer to 0x0001.						



25.3.3 Timer PWM Register (TMR_PWM, Offset = 0x0008)

Table 25-4. TMR_PWM (Offset 0x0008)

Position	31	30	29	28	27	26	25	24
Field	PWM[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	PWM[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	PWM[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	PWM[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
PWM	31:0	PWM. This register stores the value that is compared to the current timer count.	



25.3.4 Timer Interrupt Register (TMR_INT, Offset 0x000C)

Table 25-5. TMR_INT (Offset 0x000C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[6:0]							
Reset	0000 000							
Access	R							

Name	Bits	Description	Settings
IRQ_CLR	0	Clear Interrupt. Writing a value (0 or 1) to a bit in this register clears the associated interrupt.	
RFU	31:1	Reserved	N/A



25.3.5 Timer Control Register (TMR_CN, Offset 0x0010)

Table 25-6. TMR_CN (Offset 0x0010)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[2:0]			PWMCKBD*	NOLHPOL	NOLHPOL	PWMSYNC	PRES.3
Reset	000			0	0	0	0	0
Access	R			RW	RW	RW	RW	RW
Position	7	6	5	4	3	2	1	0
Field	TEN	TPOL	PRES[2:0]			TMODE[2:0]		
Reset	0	0	000			000		
Access	RW	RW	RW			RW		

Name	Bits	Description	Settings
TMODE	2:0	Timer Mode.	000: One-Shot mode 001: Continuous mode 010: Counter mode 011: PWM mode 100: Capture mode 101: Compare mode 110: Gated mode 111: Capture/Compare mode
PRES	5:3	Prescaler. The prescaler bits are formed from PRES.3:PRES[2:0]	0000: /1 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 1001: /512 1010: /1024 1011: /2048 1100: /4096 1101: undefined
TPOL	6	Timer input/output polarity bit, refer to 25.2.4, PWM Mode section for details.	
TEN	7	Timer Enable.	0: Timer disabled. 1: Timer enabled.
PRES.3	8	MSB of prescaler value.	
PWMSYNC	9	Timer PWM Synchronization Mode Enable.	0: Synchronization mode disabled 1: Synchronization mode enabled
NOLHPOL	10	Timer PWM output \emptyset_A polarity bit.	0: Output \emptyset_A not inverted 1: Output \emptyset_A inverted



Name	Bits	Description	Settings
NOLLPOL	11	Timer PWM output $\emptyset_{A'}$ polarity bit.	0: Output $\emptyset_{A'}$ not inverted 1: Output $\emptyset_{A'}$ inverted
PWMCKBD*	12	Timer PWM output $\emptyset_{A'}$ disable.	0: Output $\emptyset_{A'}$ is enabled 1: Output $\emptyset_{A'}$ is disabled
RFU	31:13	Reserved	N/A

* From Silicon Revision B1. Does not exist on Revision A3 and prior.

E/UG25H05/36488764



25.3.6 Timer Non-Overlapping Compare Register (TMR_NOLCMP, Offset 0x0014)

Table 25-7. TMR_NOLCMP (Offset 0x0014)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	NOLHCM ^P [7:0]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	NOLLCMP [7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
NOLLCMP	7:0	Non-overlapping Low Compare. The 8-bit timer count value of non-overlapping time between falling edge of PWM output \emptyset_A and next rising edge of PWM output $\emptyset_{A'}$.	
NOLHCM ^P	15:8	Non-overlapping High Compare. The 8-bit timer count value of non-overlapping time between falling edge of PWM output $\emptyset_{A'}$ and next rising edge of PWM output \emptyset_A .	
RFU	31:16	Reserved	N/A



26.0 True Random Number Generator

26.1 Overview

The Random Number Generator (TRNG) has the following features:

- True 32-bit random number generation
- Optimized for 128-bit random number generation
- AES key generation

26.2 Reading the Random Number

The TRNG block is optimized to generate 128-bit random number. To read this value, the TRNG_DATA register needs to be read four times as long as the time between each access is within 32 clock cycles. The first access reads the Least Significant 32-bit Word value of the 128 bit random number. Subsequent accesses, read the next higher 32 bits of the number. After the fourth access (Most Significant 32-bit Word value), the system must wait for at least 32 clock cycles to read the next 128 bit random number from TRNG Data Register again.

26.3 AES Key Generation

By setting the AESKG bit in the TRNG Control Register, a 256-bits random number is automatically generated and sent to the battery backed memory.

During the whole process (256-bits random number generation and AES key transfer through the bus), reading the TRNG Data Register will return 0x0000 0000 in order to prevent the AES key monitoring through the TRNG Data Register. TRNG_CN.I4S and TRNG_CN.IS bits are also forced to zero and no interrupt is generated. The entire process is transparent from the user and software stand point.

26.4 Interrupt Management

The TRNG block can generate interrupt if it is enabled (TRNG_CN.RNG_IE bit is set). When an interrupt occurs, the software must first access the TRNG_DATA register and then clear the status bit (TRNG_CN.RGN_I4S) by writing a 1 to the clear bit (TRNG.RNG_ISC) in order to acknowledge the interrupt. After acknowledging the interrupt, software can access the TRNG Data Register.

The system can also work in poll mode. The software has to disable the interrupt (clear TRNG.RNG_IE) and polls the two status bits. (TRNG.RNG_I4S for four consecutive read accesses and TRNG.RNG_IS for at least one read access). When TRNG.RNG_I4S is set, software must acknowledge the status by setting TRNG.RNG_ISC bit and then access the TRNG Data Register.

In either case it is very important to clear the TRNG.RNG_I4S before accessing the TRNG Data Register if it is set.

26.5 TRNG Registers

Address assignments for the TRNG are outlined below.

Table 26-1. TRNG Register Address (Base ADDR = 0x400B_5000)

Offset	Access	Register	Description
0x0000	RW	TRNG_CN	TRNG Control Register
0x0004	R	TRNG_DATA	TRNG Data Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



26.5.1 TRNG Control Register (TRNG_CN, Offset 0x0000)

Table 26-2. TRNG CN (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU	AESKG	RNG_IS	RNG_I4S	RNG_ISC	RNG_IE	RFU	RFU
Reset	0	0	0	0	0	0	1	1
Access	R	RW	R	R	W	RW	R	R

Name	Bits	Description	Settings
RFU	0	Reserved	Always read 1
RFU	1	Reserved	Always read 1
RNG_IE	2	Random Number Interrupt Enable. This bit enables an interrupt to be generated when a new, 128 bit, random number is ready to be read.	0: Interrupt disabled 1: Interrupt enabled
RNG_ISC	3	Random Number Interrupt Status Clear. Setting this bit to 1 clears the RNG_I4S bit and acknowledges the interrupt, if enabled. This bit is a write only bit and always reads as zero.	0: No effect. 1: Clear the status bit.
RNG_I4S	4	Random Number 4 Word Status. This bit is set when a new 128 bit random number is ready to be read (using 4 consecutive reads of the TRNG Data Register). When set, an interrupt will be generated if the RNG_IE bit is also set. This bit is cleared by setting the RNG_ISC bit.	0: TRNG Data Register is not ready. 1: TRNG Data Register is ready to be read four consecutive times.
RNG_IS	5	Random Number Word Status. This bit is set when at least one 32 bit random number is ready to be read. This bit is cleared by hardware if all the random words have been read. It is needed to poll this bit before reading the TRNG Data Register	0: A 32-bit random word is not ready. 1: A 32-bit random word is ready to be read at least once.
AESKG	6	AES Key Generate. When enabled, the key for securing NVSRAM is generated and transferred to the secure key register automatically without user visibility or intervention. This bit is cleared by hardware once the key has been transferred to the secure key register.	0: No effect. 1: Initiate AES Key Generation
RFU	31:7	Reserved	N/A



26.5.2 TRNG Data Register (TRNG_DATA, Offset 0x0004)

Table 26-3. TRNG DATA (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	DATA[31:24]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	DATA[23:16]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	DATA[15:8]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	DATA[7:0]							
Reset	0000 0000							
Access	R							

Name	Bits	Description	Settings
DATA	31:0	Data. The content of this register is valid only when RNG_IS = 1. When TRNG is disabled, read returns 0x0000 0000. Refer to section 26.2 for further information.	

27.0 UART (Serial Port)

Two UARTs are provided which can be used to communicate with external devices requiring an asynchronous serial protocol.

Features of the UARTs on the device are listed below.

- Flexible baud rate generation based on the peripheral clock nominal frequency (typically half of the system clock frequency)
- Two DMA channels can be connected (one to read FIFO, one to write FIFO)
- Programmable word size (5 bits to 8 bits), stop bits and parity settings
- Automatic parity and framing error detection
- Automatic flow control can be enabled for RTS, CTS lines
- 8 byte FIFO in both directions (separate read/RX and write/TX FIFOs)
- Interrupts available for frame error, parity error, CTS, RX FIFO overrun and FIFO full/partially full conditions

27.1 Detailed Description

The following sections provide more details on configuring UART operating modes.

27.1.1 Format and Baud Rate Selection

The overall baud rate for the UART is based on the frequency of PLL0 (typically 108 MHz). If the peripheral clock is divided down under system control, this will not change the baudrate frequency basis used by the UART. To set the baud rate value, write the appropriate divider values to the registers UART_BAUD0 and UART_BAUD1.

Note that the PLL0 output must be selected as the system clock source for proper UART operation. Using the crypto ring or the nano ring as the clock source will not provide sufficient clock accuracy to generate a stable baud rate, and using the HFXIN oscillator clock source will not allow the baud rate to be calculated correctly given the formulas provided in this section.

The transfer size (5 bits to 8 bits long), number of stop bits used and parity calculation mode are all set by writing to the appropriate bits and bit fields in the UART_CTRL register.

27.1.2 Transferring and Receiving Data

Data to be transferred must be written to the TX FIFO by writing to the UART_DATA register (either directly or using a DMA channel). This data will be transferred out by the hardware automatically a character at a time, in the order that it was received. The status flags and associated UART interrupts can be used to monitor the FIFO status and determine when the transfer cycle or cycles have completed.

As data is received, it is loaded into the RX FIFO, and it can then be unloaded by reading from the UART_DATA register. If 8 characters are received and are not unloaded by the CPU or DMA controller, and another character arrives on RX, an overrun error will occur and the new character will be lost.

27.1.3 UART Interrupts

Interrupts can be generated by each UART instance for one or more of the following conditions.

- A framing error occurs on a received character (RX transitions out of sync to the baud clock which is synchronized to the initial RX falling edge).
- The received character has an invalid parity bit according to the chosen parity settings.
- The level on the CTS line changes (when hardware flow control is disabled).
- The number of bytes in the FIFO RX buffer reaches a configurable threshold level.

- An overrun error occurs on the FIFO RX buffer.
- The FIFO TX buffer reaches a half-empty level.
- The FIFO TX buffer has only one byte remaining.

27.1.4 DMA Transfers

The UARTs support DMA transfers in both the transmit and receive directions; separate DMA channels can be connected to the RX and TX FIFO buffers. UART interrupts are not rerouted to the DMA controller and must still be handled by the CPU in the usual manner.

27.1.5 Hardware Flow Control

When hardware flow control is enabled, the CTS and RTS lines are controlled and sampled directly by hardware. With hardware flow control disabled, the bits in the UART Pin Control register can be used to control CTS and RTS manually.

The CTS (Clear To Send) line is an input to the UART that is driven by an external device. It is used by the external device to notify the UART whether or not the external device is ready to receive additional data. With hardware flow control enabled, the CTS input will automatically pause transmission from the TX FIFO when CTS is driven high (active low input). The UART samples CTS just before a new character is transmitted from the TX FIFO. If CTS is low, then the character transmission continues. If CTS is high, then the UART pauses and waits for CTS to return to a low level before continuing the transmission.

The RTS (Ready To Send) line is an output from the UART that notifies an external device whether or not the UART is ready to receive more data. When hardware flow control is enabled, the RTS output is automatically driven low (active state) whenever the RX FIFO on the UART is not full. When the RX FIFO becomes full, the RTS output will be driven high to indicate to the external device that it should pause its data transmission until the UART has more buffer space available to receive data.

27.2 UART Instance Register Blocks

Each UART instance is controlled by a block of registers assigned to that port; all blocks contain identical control, interrupt, status and FIFO read/write registers.

The addresses for each register block are listed in Table 2-2. Peripheral Bus Region.

27.3 UART Registers

Address assignments for UART registers within each block are outlined in Table 27-1. Reserved register bits should only be written as 0.

Register names for different instances can be defined by appending the instance number to the peripheral name, so for instance, the Control Register for UART 0 would be UART0_CTRL, while the Control Register for UART 1 would be UART1_CTRL, and so on.

Table 27-1. UART Register Addresses (for each instance)

Offset	Access	Register	Description
0x0000	RW	UART_CTRL	UART Control Register
0x0004	R	UART_STAT	UART Status Register
0x0008	RW	UART_INT_EN	UART Interrupt Enable Register
0x000C	RC	UART_INT_STAT	UART Interrupt Status Register
0x0010	RW	UART_BAUD0	UART Baud Rate Register 0
0x0014	RW	UART_BAUD1	UART Baud Rate Register 1
0x001C	RW	UART_PIN	UART Pin Control Register
0x0020	RW	UART_DATA	UART Data Buffer Register
0x0030	RW	UART_DMA	UART DMA Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



27.3.1 UART Control Register (UART_CTRL, Offset 0x0000)

Table 27-2. UART_CTRL (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU	RFU	RTSCTSF	STOP	SIZE1	SIZE0	RXFLUSH	TXFLUSH
Reset	0	0	0	0	0	0	0	0
Access	R	R	RW	RW	RW	RW	RW	RW
Position	7	6	5	4	3	2	1	0
Field	RFU	PARMD	PAREO	PAREN	RXTHD[3:0]			
Reset	0	0	0	0	0000			
Access	R	RW	RW	RW	RW			

Name	Bits	Description	Settings
RXTHD[3:0]	3:0	Number of bytes that must be loaded into the receive FIFO before the FFRXIS interrupt will be triggered.	0: Ignored 1-8: Interrupt will be triggered when the number of bytes in the RX FIFO reaches this level or higher 9+: Ignored
PAREN	4	Enables/disables generation and testing of parity bit.	0: Parity is disabled. 1: Parity is enabled.
PAREO	5	Selects odd or even parity (when PAREN=1).	0: Even parity selected. 1: Odd parity selected.
PARMD	6	Selects parity based on 1s or 0s count (when PAREN=1).	0: Parity calculation is based on number of 1s in frame. 1: Parity calculation is based on number of 0s in frame.
RFU	7	Reserved.	N/A
TXFLUSH	8	Flushes the TX FIFO buffer.	Write to 1 to flush the FIFO.
RXFLUSH	9	Flushes the RX FIFO buffer.	Write to 1 to flush the FIFO.
SIZE[1:0]	11:10	Selects UART character size.	00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits
STOP	12	Selects the number of stop bits that will be generated.	0: 1 stop bit 1: 1.5 stop bits (for 5 bit mode) or 2 stop bits (for 6/7/8 bit mode)
RTSCTSF	13	Enables/disables hardware flow control.	0: Hardware flow control disabled. 1: Hardware flow control enabled.
RFU	31:14	Reserved.	N/A



27.3.2 UART Status Register (UART_STAT, Offset 0x0004)

Table 27-3. UART_STAT (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	TXELT				RXELT			
Reset	0000				0000			
Access	R				R			
Position	7	6	5	4	3	2	1	0
Field	TXFULL	TXEMPTY	RXFULL	RXEMPTY	RFU	RFU	RXBUSY	TXBUSY
Reset	0	1	0	1	0	0	0	0
Access	R	R	R	R	R	R	R	R

Name	Bits	Description	Settings
TXBUSY	0	Read-only flag indicating the UART transmit status.	0: UART is not currently transmitting data. 1: UART is currently transmitting data.
RXBUSY	1	Read-only flag indicating the UART receiver status.	0: UART is not currently receiving a character. 1: UART is currently receiving a character.
RFU	3:2	Reserved.	N/A
RXEMPTY	4	Read-only flag indicating the RX FIFO state.	0: RX FIFO is not empty. 1: RX FIFO is empty (RXELT=0).
RXFULL	5	Read-only flag indicating the RX FIFO state.	0: RX FIFO is not full. 1: RX FIFO is full (RXELT=8).
TXEMPTY	6	Read-only flag indicating the TX FIFO state.	0: TX FIFO is not empty. 1: TX FIFO is empty (TXELT=0).
TXFULL	7	Read-only flag indicating the TX FIFO state.	0: TX FIFO is not full. 1: TX FIFO is full (TXELT=8).
RXELT	11:8	Read-only field.	Indicates the number of bytes currently in the RX FIFO (0-8)
TXELT	15:12	Read-only field.	Indicates the number of bytes currently in the TX FIFO (0-8)
RFU	31:16	Reserved.	N/A



27.3.3 UART Interrupt Enable Register (UART_INT_EN, Offset 0x0008)

Table 27-4. UART_INT_EN (Offset 0x0008)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU	FFTXHIE	FFTXOIE	FFRXIE	OVERIE	SIGNALIE	PARITYIE	FRAMIE
Reset	0	0	0	0	0	0	0	0
Access	R	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
FRAMIE	0	Enable for RX Frame Error interrupt.	0: Interrupt disabled. 1: Interrupt enabled.
PARITYIE	1	Enable for RX Parity Error interrupt.	0: Interrupt disabled. 1: Interrupt enabled.
SIGNALIE	2	Enable for CTS signal change interrupt (hardware flow control disabled).	0: Interrupt disabled. 1: Interrupt enabled.
OVERIE	3	Enable for RX FIFO Overrun interrupt.	0: Interrupt disabled. 1: Interrupt enabled.
FFRXIE	4	Enable for interrupt when RX FIFO reaches the number of bytes configured by the RXTHD field.	0: Interrupt disabled. 1: Interrupt enabled.
FFTXOIE	5	Enable for interrupt when TX FIFO has only one byte remaining.	0: Interrupt disabled. 1: Interrupt enabled.
FFTXHIE	6	Enable for interrupt when TX FIFO is half empty or less.	0: Interrupt disabled. 1: Interrupt enabled.
RFU	31:7	Reserved.	N/A



27.3.4 UART Interrupt Status Register (UART_INT_STAT, Offset 0x000C)

Table 27-5. UART_INT_STAT (Offset 0x000C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU	FFTXHIS	FFTXOIS	FFRXIS	OVERIS	SIGNALIS	PARITYIS	FRAMIS
Reset	0	0	0	0	0	0	0	0
Access	R	RC	RC	RC	RC	RC	RC	RC

Name	Bits	Description	Settings
FRAMIS	0	Flag for RX Frame Error Interrupt.	Set to 1 by hardware when interrupt occurs; software must write to 0 to clear. Writes to 1 by software are ignored.
PARITYIS	1	Flag for RX Parity Error interrupt.	Set to 1 by hardware when interrupt occurs; software must write to 0 to clear. Writes to 1 by software are ignored.
SIGNALIS	2	Flag for CTS signal change interrupt (hardware flow control disabled).	Set to 1 by hardware when interrupt occurs; software must write to 0 to clear. Writes to 1 by software are ignored.
OVERIS	3	Flag for RX FIFO Overrun interrupt.	Set to 1 by hardware when interrupt occurs; software must write to 0 to clear. Writes to 1 by software are ignored.
FFRXIS	4	Flag for interrupt when RX FIFO reaches the number of bytes configured by the RXTHD field.	Set to 1 by hardware when interrupt occurs; software must write to 0 to clear. Writes to 1 by software are ignored.
FFTXOIS	5	Flag for interrupt when TX FIFO has only one byte remaining.	Set to 1 by hardware when interrupt occurs; software must write to 0 to clear. Writes to 1 by software are ignored.
FFTXHIS	6	Flag for interrupt when TX FIFO is half empty or less.	Set to 1 by hardware when interrupt occurs; software must write to 0 to clear. Writes to 1 by software are ignored.
RFU	31:7	Reserved.	N/A



27.3.5 UART Baud Rate Register 0 (UART_BAUD0, Offset 0x0010)

Table 27-6. UART_BAUD0 (Offset 0x0010)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU				IDIV[11:8]			
Reset	0000				0000			
Access	R				RW			
Position	7	6	5	4	3	2	1	0
Field	IDIV[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
IDIV[11:0]	11:0	Integer portion of baud rate divisor value.	Use the following formula to obtain the proper value for this field: $\text{IDIV} = (\text{integer portion of 'Nominal Frequency'}) / (128 \times \text{Baud Rate Frequency})$ where the 'Nominal Frequency' = $2 \times f_{\text{PLL}}$ The integer portion is rounded value (i.e. 5.7 becomes 5)
RFU	31:12	Reserved.	N/A



27.3.6 UART Baud Rate Register 1 (UART_BAUD1, Offset 0x0014)

Table 27-7. UART_BAUD1 (Offset 0x0014)

Position	31	30	29	28	27	26	25	24	
Field	RFU[7:0]								
Reset	0000 0000								
Access	R								
Position	23	22	21	20	19	18	17	16	
Field	RFU[7:0]								
Reset	0000 0000								
Access	R								
Position	15	14	13	12	11	10	9	8	
Field	RFU[7:0]								
Reset	0000 0000								
Access	R								
Position	7	6	5	4	3	2	1	0	
Field	RFU	DDIV[6:0]							
Reset	0	00 0000							
Access	R	RW							

Name	Bits	Description	Settings
DDIV	6:0	Decimal portion of baud rate divisor value.	Use the following formula to obtain the proper value for this field after obtaining IDIV: $DIV = \text{Nominal Frequency} / (128 \times \text{Baud Rate Frequency})$ $DDIV = (DIV - IDIV) \times 128$ where the 'Nominal Frequency' = $2 \times f_{\text{PLL}}$
RFU	31:7	Reserved.	N/A



27.3.7 UART Pin Control Register (UART_PIN, Offset 0x001C)

Table 27-8. UART_PIN (Offset 0x001C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[5:0]						RTS	CTS
Reset	00 0000						1	1
Access	R						RW	R

Name	Bits	Description	Settings
CTS	0	Current state of CTS I/O pin.	0: CTS is currently low. 1: CTS is currently high.
RTS	1	If hardware flow control is disabled, this bit is used to control the RTS I/O pin output state.	0: RTS will be driven low. 1: RTS will be driven high. This bit has no effect when RTSCTSF=1.
RFU	31:2	Reserved.	N/A



27.3.8 UART Data Buffer Register (UART_DATA, Offset 0x0020)

Table 27-9. UART DATA (Offset 0x0020)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[6:0]							
Reset	000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	DATA[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
DATA[7:0]	7:0	Load/unload location for TX and RX FIFO buffers.	Writing to this field loads the next character into the TX FIFO (if space is available). Reading from this field returns the next character available at the output of the RX FIFO (if one is available, otherwise 00h is returned).
PARITY	8	Parity error flag for next byte to be read from FIFO.	If a parity error occurred during the reception of the character at the output end of the RX FIFO (that would be returned by reading the DATA field), this bit will read 1, otherwise it will read 0.
RFU	31:9	Reserved.	N/A



27.3.9 UART DMA Register (UART_DMA, Offset 0x0030)

Table 27-10. UART DMA (Offset 0x0030)

Position	31	30	29	28	27	26	25	24	
Field	RFU[7:0]								
Reset	0000 0000								
Access	R								
Position	23	22	21	20	19	18	17	16	
Field	RFU[7:0]								
Reset	0000 0000								
Access	R								
Position	15	14	13	12	11	10	9	8	
Field	RFU[5:0]								
Reset	00 0000								
Access	R								
Position	7	6	5	4	3	2	1	0	
Field	RXCNT[2:0]				TXEN	TXCNT[3:0]			
Reset	000				0	0000			
Access	RW				RW	RW			

Name	Bits	Description	Settings
TXCNT	3:0	TX threshold for DMA transmission	
TXEN	4	TX DMA channel enable	0: Disabled 1: Enabled
RXCNT	8:5	RX threshold for DMA transmission	
RXEN	9	RX DMA channel enable	0: Disabled 1: Enabled
RFU	31:10	Reserved.	N/A

28.0 USB Controller

The MAX32550 Universal Serial Bus (USB) Peripheral supports a dedicated USB device.

The USB controller defined in this User Guide is compliant with USB Specification Revision 2.0.

The USB can be enabled by setting the USBEN bit to 1 in the corresponding USB Control Registers described in Table 28-4. on page 395. To ensure proper operation of the USB controller, the minimum AHB bus frequency has to be set to 24MHz.

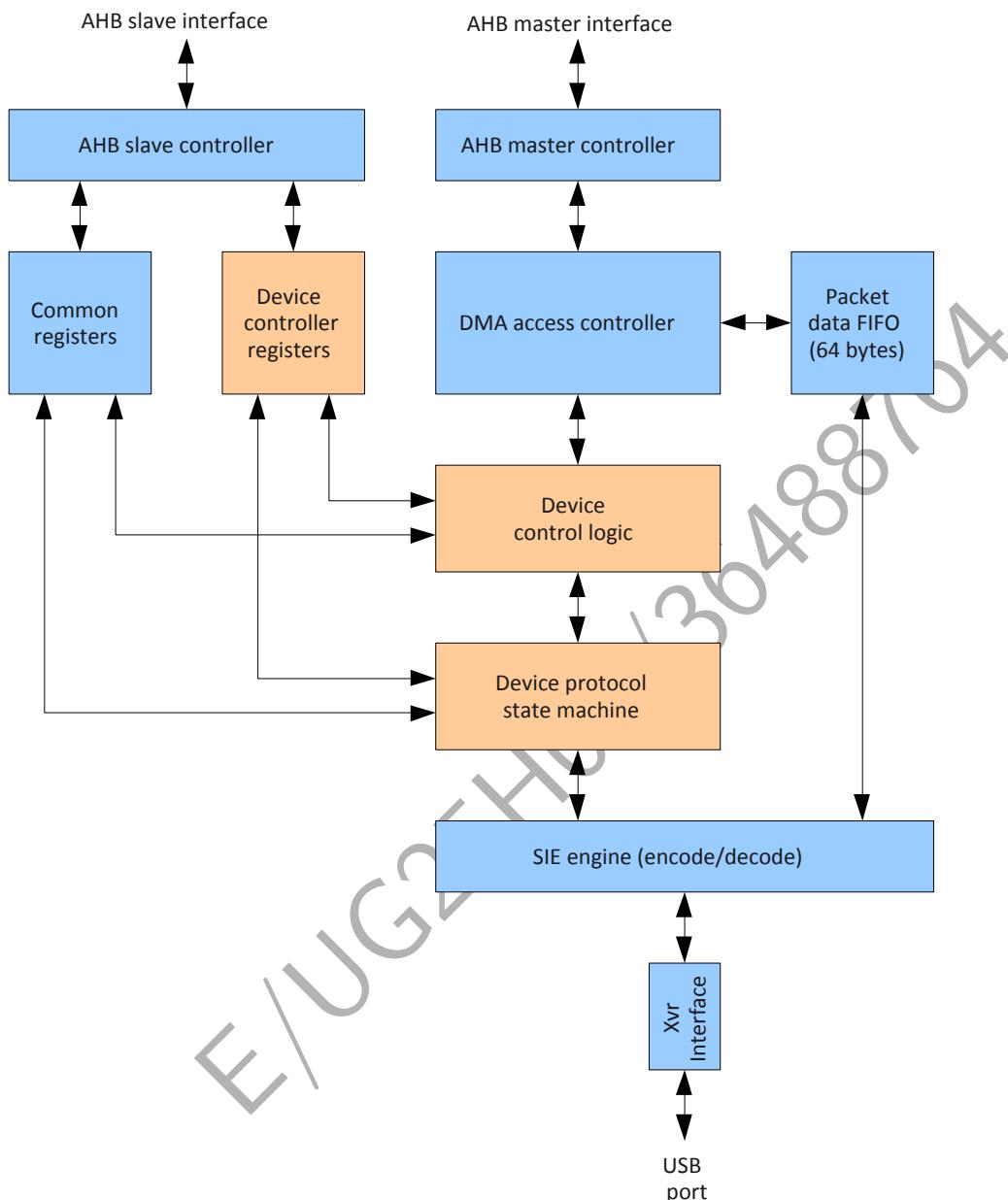
28.1 Device Operation

The USB device controller can support up to a total of 16 endpoints with programmable configuration.

The USB device is reset under the following conditions:

- USB Bus Reset: Host issues a bus reset
- USB Device Reset: Software sets USB_DEV_CN.URST bit to 1.
- System Reset: System undergoing a reset.

While describing device operation, a reset condition without any qualifier refers to system reset. The term “USB reset” refers to either a USB bus reset or a USB device reset.

Figure 28-1. USB Device Block Diagram

28.2 USB Endpoint

Each of the USB endpoints can be individually configured and supports:

- Single / Double buffer
- Programmable buffer starting address
- Programmable interrupt generation
- Ability to STALL a packet
- Bulk and Interrupt transfer
- Control transfer (endpoint 0 only)

- Configurable response to Status Stage of Control transfer

Each endpoint is characterized by:

- Endpoint Control Register
- Endpoint Buffer Descriptor
- Endpoint Buffer Address

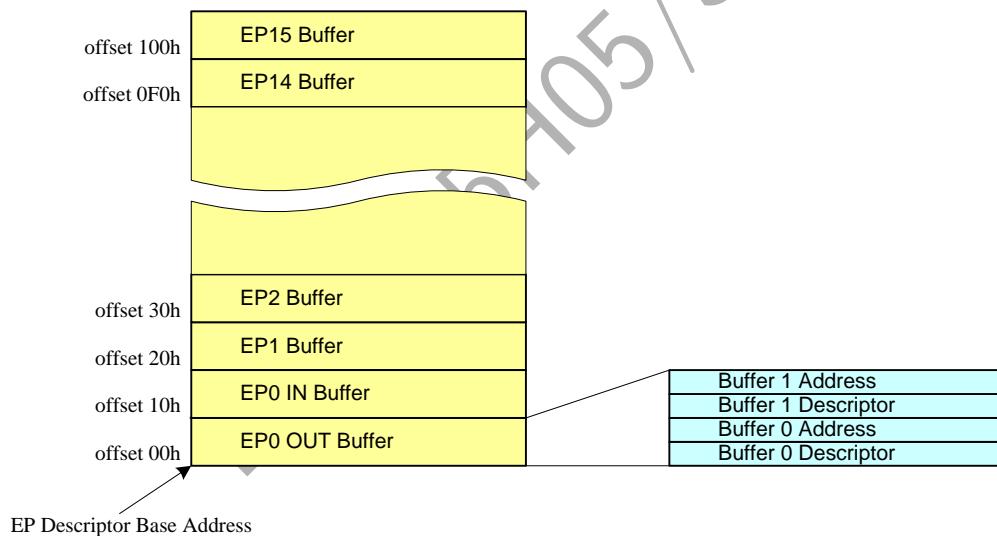
The Endpoint Control Register (USB_EPx) is used to define the endpoint general characteristics such as direction of packet transfer, number of buffers, and interrupt generation.

28.2.1 Endpoint Buffer Descriptor

The Endpoint Buffer Descriptor is used as a communication port between the software and SIE in system memory. The Endpoint Buffer Descriptor starting address is specified by the USB Endpoint Descriptor Base Address Register and is described in Table 28-9 on page 404.

The endpoint data toggle value and the buffer are maintained internally by the SIE. The data toggle value is initialized to DATA0 on USB reset. The buffer will be reset to buffer 0 on USB reset. When an endpoint is double-buffered, the SIE will use the two buffers in ping pong fashion.

Figure 28-2. Endpoint Buffer Descriptor Memory



Each endpoint can have two buffers per direction of transfer: Buffer 0 and Buffer 1. Each buffer has two entries: the endpoint Buffer Descriptor and the Buffer Starting Address.

The first entry – Endpoint Buffer Descriptor is used to determine how and when the SIE should respond to a packet targeted to this buffer location.

**Table 28-1. Endpoint Buffer Descriptor**

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[5:0] BC[9:8]							
Reset	0000 00 00							
Access	R RW							
Position	7	6	5	4	3	2	1	0
Field	BC[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description	Settings					
BC	9:0	Buffer Byte Count.	Software writes the byte count value to indicate buffer length. SIE updates the byte count value with the number of bytes received for an OUT/SETUP transfer.					
RFU	31:10	Reserved	N/A					



28.2.2 Endpoint Buffer Address

The Endpoint Buffer Address specifies the starting address for the endpoint buffer in system memory.

Table 28-2. USB Endpoint Address

Position	31	30	29	28	27	26	25	24
Field	EP_ADDR[31:24]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	EP_ADDR[23:16]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	EP_ADDR[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	EP_ADDR[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
EP_ADDR	31:0	Endpoint Buffer Address	This register defines the starting address of the buffer for this endpoint descriptor in system memory. This field has to be aligned to 512 byte boundary.



28.3 USB Device Registers

Unless otherwise stated, the USB device registers are not affected by a USB bus reset. Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.

Table 28-3. USB Device Registers (Base ADDR= 0x400B_0000)

Offset	Register Name	Description	Reset Value
0x0000	USB_CN	USB Control Register	0x00000 0000
0x0200	USB_DEV_ADDR	USB Device Address Register	0x00000 0000
0x0204	USB_DEV_CN	USB Device Control Register	0x00000 0000
0x0208	USB_DEV_INT	USB Device Interrupt Register	0x00000 0000
0x020C	USB_DEV_INT_EN	USB Device Interrupt Enable Register	0x00000 0000
0x0210 - 0x021C		Reserved	0x00000 0000
0x0220	USB_EP_BASE	USB Endpoint Descriptor Base Address Register	0x00000 0000
0x0224	USB_CUR_BUF	USB Current Buffer Register	0x00000 0000
0x0228	USB_IN_OWNER	USB IN Buffer Owner Register	0x00000 0000
0x022C	USB_OUT_OWNER	USB OUT Buffer Owner Register	0x00000 0000
0x0230	USB_IN_INT	USB IN Interrupt Register	0x00000 0000
0x0234	USB_OUT_INT	USB OUT Interrupt Register	0x00000 0000
0x0238	USB_NAK_INT	USB NAK Interrupt Register	0x00000 0000
0x023C	USB_DMA_ERR_INT	USB DMA Error Interrupt Register	0x00000 0000
0x0240	USB_BUF_OVR_INT	USB Buffer Overflow Interrupt Register	0x00000 0000
0x0244 - 0x025C		Reserved	0x00000 0000
0x0260	USB_SETUP0	USB SETUP Register 0	0x00000 0000
0x0264	USB_SETUP1	USB SETUP Register 1	0x00000 0000
0x0268 - 0x027C		Reserved	0x00000 0000
0x0280	USB_EP0	USB Endpoint0 Control Register	0x00000 0000
0x0284	USB_EP1	USB Endpoint1 Control Register	0x00000 0000
0x0288	USB_EP2	USB Endpoint2 Control Register	0x00000 0000
0x028C	USB_EP3	USB Endpoint3 Control Register	0x00000 0000
0x0290	USB_EP4	USB Endpoint4 Control Register	0x00000 0000
0x0294	USB_EP5	USB Endpoint5 Control Register	0x00000 0000
0x0298	USB_EP6	USB Endpoint6 Control Register	0x00000 0000
0x029C	USB_EP7	USB Endpoint7 Control Register	0x00000 0000
0x02A0	USB_EP8	USB Endpoint8 Control Register	0x00000 0000
0x02A4	USB_EP9	USB Endpoint9 Control Register	0x00000 0000
0x02A8	USB_EP10	USB Endpoint10 Control Register	0x00000 0000
0x02AC	USB_EP11	USB Endpoint11 Control Register	0x00000 0000
0x02B0	USB_EP12	USB Endpoint12 Control Register	0x00000 0000
0x02B4	USB_EP13	USB Endpoint13 Control Register	0x00000 0000
0x02B8	USB_EP14	USB Endpoint14 Control Register	0x00000 0000
0x02BC	USB_EP15	USB Endpoint15 Control Register	0x00000 0000
0x02C0 - 0x03FF		Reserved	0x00000 0000



28.3.1 USB Control Register (USB_CN, Offset 0x0000)

The USB Control Register is used for enabling the USB device and determining mode of operation.

This register will contain the value of 0h after a reset.

Table 28-4. USB_CN (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU[5:0]					RFU	USB_EN	
Reset	0000 00					0	0	
Access	R					R	RW	

Name	Bits	Description	Settings
USB_EN	0	USB Enable	0: USB disabled 1: USB enabled
RFU	1	Reserved	Always read zero
RFU	31:2	Reserved	N/A



28.3.2 USB Device Address Register (USB_DEV_ADDR, Offset 0x0200)

The USB Device Address Registers contains the address of the USB Peripheral. The address is used by the host to communicate to the USB device.

The SetAddress() request directed to endpoint 0 is the only USB request handled entirely by the USB controller. For all other requests or SetAddress directed to other control endpoints, the USB controller stores the SETUP bytes in control endpoint buffer, and then asserts an interrupt to the core.

After reset, the device address defaults to a value of zero and must be programmed by the host during the enumeration process. Device address zero is reserved as the default address and may not be assigned to any other use. A USB device maintains its assigned address while suspended, and resets it to zero on reset or USB bus reset.

After a reset recovery interval, if a device receives a SetAddress() request, the device must be able to complete processing of the request and be able to successfully complete the Status stage of the request within 50ms. After successful completion of the Status stage, the device is allowed a SetAddress() recovery interval of 2ms. At the end of this interval, the device must be able to accept Setup packets addressed to the new address.

The USB device address is stored in USB_DEV_ADDR register. Internal the USB controller clears the address register during USB bus reset. This register is read only.

This register will contain the value of 00h after a reset or a USB reset.

Table 28-5. USB_DEV_ADDR (Offset 0x0200)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	RFU	DEV_ADDR[6:0]						
Reset	0	000 0000						
Access	R	R						

Name	Bits	Description	Settings
DEV_ADDR	6:0	USB Device Address.	These bits contain the USB device address
RFU	31:17	Reserved	N/A



28.3.3 USB Device Control Register (USB_DEV_CN, Offset 0x0204)

The USB Device Control Register defines the USB device operation.

This register will contain the value of 0B0h after a reset or a USB device reset.

Table 28-6. USB_DEV_CN (Offset 0x0204)

Position	31	30	29	28	27	26	25	24		
Field	RFU[7:0]									
Reset	0000 0000									
Access	R									
Position	23	22	21	20	19	18	17	16		
	RFU[7:0]									
Reset	0000 0000									
Access	R									
Position	15	14	13	12	11	10	9	8		
Field	RFU[5:0]						FIFO_MODE	BACT_OE		
Reset	00 0000						0	0		
Access	R						R	RW		
Position	7	6	5	4	3	2	1	0		
	OSCEN	VBGATE	URST	ULPM	CONNECT	SIGRWU	RFU[1:0]			
Reset	1	0	1	1	0	0	00			
Access	RW	RW	RW	RW	RW	RW	R			
Name	Bits	Description	Settings							
RFU	1:0	Reserved	N/A							
SIGRWU	2	USB Signal Remote Wakeup.	0: Do not send remote wakeup signal. 1: Send remote wakeup signal to host.							
CONNECT	3	Connect to USB	0: Disconnect pull up resistor between DPLUS and VBUS. 1: Connect pull up resistor between DPLUS and VBUS							

Name	Bits	Description	Settings
ULPM	4	<p>USB Low Power Mode. On receiving a USB suspend interrupt, software can put the USB transceiver into a low power state by setting this bit to 1. To return the USB transceiver to normal operation, software can clear this bit to 0.</p> <p>The USB is capable of causing the device to exit low power mode on detecting</p> <ol style="list-style-type: none"> 1) DPLUS activity when DPLUS interrupt is enabled 2) VBUS detect when VBUS interrupt is enabled. <p>NOTE: When the above cases occur, the USB controller will generate a wake up signal to the core. The core defines system wake up sources which are configurable by software.</p>	<p>0: USB transceiver in normal operation. 1: USB transceiver will enter a low power state.</p>
URST	5	<p>USB Device Controller Reset. NOTE: This bit applies only to device controller registers. It does not apply to</p> <ol style="list-style-type: none"> 1) USB_CN register 2) Host controller or any of its registers. 3) Endpoint buffer descriptor residing in system memory <p>When URST is set to 1, the USB controller shall disconnect the device from the USB bus and then reset its states.</p>	<p>0: Release the USB device controller from reset 1: Reset USB device controller internal states and all USB device registers to their default reset values. Data toggle field is set to DATA0 and buffer is set to buffer 0.</p>
VBGATE	6	VBUS Gate	<p>0: CONNECT operation independent of VBUS status 1: CONNECT operation conditional on VBUS present</p>
OSCEN	7	Oscillator Enable. The oscillator is assumed to be ready when the USB controller is enabled.	<p>0: Maintain oscillator current state. 1: Enable the oscillator after a Suspend when USB bus activity is detected.</p>
BACT_OE	8	Bus Activity Output Enable	<p>0: Bus activity is not outputted 1: Output bus state onto the BUSACT signal. The output signal goes high when a SYNC field is detected and low after 32 consecutive J-States. This signal may be connected externally to a port pin to give the user an indication of bus activity. Also, by timing the duty cycle of this output, the bus bandwidth utilization may be measured.</p>

Name	Bits	Description	Settings
FIFO_MODE	9	FIFO Mode. This setting reduces the likelihood of the device returning NAK due to a busy system bus. In the event of a data underflow (FIFO empty when more data is needed), the device controller will force a bit stuff error on the bus and terminate the packet.	0: The device controller will return NAK to an IN request until the entire packet has been read from system memory into its internal FIFO. 1: The device controller will respond to an IN request as soon as data becomes available in the FIFO.
RFU	31:10	Reserved	N/A

E/UG25H05/36488764



28.3.4 USB Device Interrupt Register (USB_DEV_INT, Offset 0x0208)

The USB Interrupt Register contains the interrupt bits for each of the interrupt sources within the USB interface. All of the interrupts are set by hardware. Each of these bits is qualified with their respective interrupt enable bits. Once an interrupt bit has been set, it may only be cleared by writing a one to the respective interrupt bit.

This register will contain the value of 00h after a reset or USB device reset.

Table 28-7. USB_DEV_INT (Offset 0x0208)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[6:0]							
Reset	000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[1:0]		BUF_OVR	DMA_ERR	EP_NAK	EP_OUT	EP_IN	SETUP
Reset	00		0	0	0	0	0	0
Access	R		RW	RW	RW	RW	RW	RW
Position	7	6	5	4	3	2	1	0
Field	BRST_DN	VBUS	NO_VBUS	SUSP	BRST	BACT	RWU_DN	DPACT
Reset	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
DPACT	0	DPLUS Activity. DPLUS activity is an interrupt source that will bring the core out of low power mode.	0: No Activity detected on DPLUS pin. 1: Activity detected on the DPLUS pin.
RWU_DN	1	Remote Wakeup Signaling Done. This bit is cleared by a USB bus reset.	0: Remote Wakeup signaling not done. 1: Remote Wakeup signaling done.
BACT	2	USB Bus Active. This bit is cleared by a USB bus reset.	0: No USB bus activity has been detected 1: The USB controller has received a SYNC field
BRST	3	USB Bus Reset. This bit is self-cleared by hardware at the end of the USB reset condition.	0: No USB bus reset has been detected. 1: USB bus reset has been detected. When the bus reset condition is detected, the USB controller will reset its internal registers to their default values.
SUSP	4	Suspend. This bit is cleared by a USB bus reset.	0: No USB suspend has been detected 1: The SIE detects 3 ms of idle state on the bus
NO_VBUS	5	No VBUS. This bit is cleared by a USB bus reset. This bit can be an interrupt source that will bring the core out of low power mode	0: No change on VBUS 1: A 1 to 0 transition (VBUS not present) is detected
VBUS	6	VBUS Detect. This bit is cleared by a USB bus reset. This bit can be an interrupt source that will bring the core out of low power mode	0: No change on VBUS 1: A 0 to 1 transition (VBUS present) is detected

Name	Bits	Description	Settings
BRST_DN	7	Bus Reset Done	0: No USB bus reset done has been detected 1: USB bus reset is done. The BRSTDN bit indicates the end of USB bus reset condition.
SETUP	8	Setup Interrupt	0: No SETUP packet interrupt pending. 1: A SETUP packet is received at endpoint 0. Software should read USB_SETUPx to retrieve the SETUP packet.
EP_IN	9	Endpoint IN Interrupt	0: No endpoint IN interrupt pending 1: An endpoint IN interrupt is pending. Software should read USB_IN_INT register to determine the endpoint causing this interrupt.
EP_OUT	10	Endpoint OUT Interrupt	0: No endpoint OUT interrupt pending 1: An endpoint OUT interrupt is pending. Software should read USB_OUT_INT register to determine the endpoint causing this interrupt.
EP_NAK	11	Endpoint NAK Interrupt	0: No endpoint NAK interrupt pending 1: An endpoint NAK interrupt is pending. Software should read USB_NAK_INT register to determine the endpoint causing this interrupt.
DMA_ERR	12	DMA Error	0: No DMA error interrupt pending. 1: An DMA error interrupt is pending. Software should read USB_DMA_ERR_INT register to determine the endpoint causing this interrupt.
BUF_OVR	13	Buffer Overflow	0: No Buffer Overflow error has been detected. 1: An BUF_OVR error interrupt is pending. Software should read USB_BUF_OVR_INT register to determine the endpoint causing this interrupt.
RFU	15:14	Reserved	N/A
VBUS_ST	16	VBUS Status	0: VBUS is at level low 1: VBUS is at level high
RFU	31:17	Reserved	N/A



28.3.5 USB Device Interrupt Enable Register (USB_DEV_INT_EN, Offset 0x020C)

The USB Interrupt Enable Register contains the enable bits for each of the interrupt sources within the USB interface. Setting any of these bits will enable the respective interrupt source in USB_DEV_INT register.

This register will contain the value of 00h after a reset or USB device reset.

Table 28-8. USB_DEV_INT_EN (Offset 0x020C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[1:0]	BUF_OVR	DMA_ERR	EP_NAK	EP_OUT	EP_IN	SETUP	
Reset	00	0	0	0	0	0	0	
Access	R	RW	RW	RW	RW	RW	RW	
Position	7	6	5	4	3	2	1	0
Field	BRST_DN	VBUS	NO_VBUS	SUSP	BRST	BACT	RWU_DN	DPACT
Reset	0	0	0	0	0	0	0	0
Access	RW	RW	RW	RW	RW	RW	RW	RW

Name	Bits	Description	Settings
DPACT	0	DPLUS Activity Interrupt Enable	0: DPLUS activity interrupt disabled 1: DPLUS activity interrupt enabled
RWU_DN	1	Remote Wakeup Signaling Done Interrupt Enable	0: Remote wakeup signaling done interrupt disabled 1: Remote wakeup signaling done interrupt enabled
BACT	2	USB Bus Active Interrupt Enable	0: USB bus activity interrupt disabled 1: USB bus activity interrupt enabled
BRST	3	USB Bus Reset Interrupt Enable	0: USB bus reset interrupt disabled 1: USB bus reset interrupt enabled
SUSP	4	Suspend Interrupt Enable	0: USB suspend interrupt disabled 1: USB suspend interrupt enabled
NO_VBUS	5	No VBUS Interrupt Enable	0: No VBUS interrupt disabled 1: No VBUS interrupt enabled
VBUS	6	VBUS Detect Interrupt Enable	0: VBUS interrupt disabled 1: VBUS interrupt enabled
BRST_DN	7	Bus Reset Done Interrupt Enable	0: USB bus reset done interrupt disabled 1: USB bus reset done interrupt enabled
SETUP	8	Endpoint SETUP Interrupt Enable	0: Endpoint SETUP interrupt disabled 1: Endpoint SETUP interrupt enabled
EP_IN	9	Endpoint IN Interrupt Enable	0: Endpoint IN interrupt disabled 1: Endpoint IN interrupt enabled
EP_OUT	10	Endpoint OUT Interrupt Enable	0: Endpoint OUT interrupt disabled 1: Endpoint OUT interrupt enabled
EP_NAK	11	Endpoint NAK Interrupt Enable	0: Endpoint NAK interrupt disabled 1: Endpoint NAK interrupt enabled



Name	Bits	Description	Settings
DMA_ERR	12	DMA Error Interrupt Enable	0: DMA error interrupt disabled 1: DMA error interrupt enabled
BUF_OVR	13	Buffer Overflow	0: Buffer overflow interrupt disabled 1: Buffer overflow interrupt enabled
RFU	31:14	Reserved	N/A

E/UG25H05/36488764



28.3.6 USB Endpoint Descriptor Base Address Register (USB_EP_BASE, Offset 0x0220)

The USB Endpoint Descriptor Base Address Register contains the base address for the endpoint buffer descriptor in system memory and is aligned to 512 byte boundary.

This register will contain the value of 00h after a reset or USB device reset.

Table 28-9. USB_EP_BASE (Offset 0x0220)

Position	31	30	29	28	27	26	25	24
Field	EP_BASE[22:15]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	EP_BASE[14:7]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	EP_BASE[6:0]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							

Name	Bits	Description	Settings
RFU	8:0	Reserved	N/A
EP_BASE	31:9	USB_Endpoint Descriptor Base Address	<p>This register defines the starting address of the endpoint descriptor in system memory.</p> <p>The descriptor address of each buffer in the endpoint is formed by:</p> <p>EP_BASE[31:9] : EP_DESCRIPTOR_OFFSET.</p> <p>See Figure 28-2. Endpoint Buffer Descriptor Memory for the offset of each endpoint descriptor.</p>



28.3.7 USB Current Buffer Register (USB_CUR_BUF, Offset 0x0224)

The USB Current Buffer is used to indicate to software the buffer the USB controller will use when it receives a new transfer request. For an IN transfer, this indicates the buffer from which the USB controller will transmit from. For an OUT transfer, this number indicates the buffer into which the USB controller will receive data to. If the endpoint is single buffered (EP_BUF2), this bit corresponding to the transfer direction will remain 0 all the time.

Endpoint Current Buffer Register is cleared to 00h on reset or USB device reset. This register is read only.

Table 28-10. USB CUR BUF (Offset 0x0224)

Position	31	30	29	28	27	26	25	24
Field	IN_BUF[15:8]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	IN_BUF[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	OUT_BUF[15:8]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	OUT_BUF[7:0]							
Reset	0000 0000							
Access	R							

Name	Bits	Description	Settings
OUT_BUF	15:0	OUT Transfer Current Buffer. Each bit represents the OUT transfer buffer of the corresponding endpoint.	0: Current buffer is buffer 0 1: Current buffer is buffer 1
IN_BUF	31:16	IN Transfer Current Buffer. Each bit represents the IN transfer buffer of the corresponding endpoint	0: Current buffer is buffer 0 1: Current buffer is buffer 1



28.3.8 USB IN Buffer Owner Register (USB_IN_OWNER, Offset 0x0228)

This register indicates whether the software or the hardware has control over a specific IN buffer. The ownership bit is set to 1 by software and cleared to 0 by the USB controller upon finishing an transfer associated with this buffer.

Software should set ownership bit as the last step in setting up an endpoint IN buffer. Once the ownership bit is set to 1, software should not access any of the system memory associated with this endpoint IN buffer.

This register is cleared to 00h on reset or USB device reset. In addition, setting the Data Toggle bit of an endpoint will clear the corresponding endpoint IN Buffer Owner bits to 0.

Table 28-11. USB IN OWNER (Offset 0x0228)

Position	31	30	29	28	27	26	25	24
Field	INBUF1_OWNER[15:8]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	INBUF1_OWNER[7:0]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	INBUFO_OWNER[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	INBUFO_OWNER[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
INBUFO_OWNER	15:0	IN Buffer 0 Owner. Each bit represents buffer 0 ownership of the corresponding endpoint.	0: This endpoint IN buffer 0 is owned by the software 1: This endpoint IN buffer 0 is owned by the USB controller
INBUF1_OWNER	31:16	IN Buffer 1 Owner. Each bit represents buffer 1 ownership of the corresponding endpoint.	0: This endpoint IN buffer 1 is owned by the software 1: This endpoint IN buffer 1 is owned by the USB controller



28.3.9 USB OUT Buffer Owner Register (USB_OUT_OWNER, Offset 0x022C)

This register indicates whether the software or the hardware has control over a specific OUT buffer. The ownership bit is set to 1 by software and cleared to 0 by the USB controller upon finishing an transfer associated with this buffer.

Software should set ownership bit as the last step in setting up an endpoint OUT buffer. Once the ownership bit is set to 1, software should not access any of the system memory associated with this endpoint OUT buffer.

This register is cleared to 00h on reset or USB device reset. In addition, setting the Data Toggle bit of an endpoint will clear the corresponding endpoint OUT Buffer Owner bits to 0.

Table 28-12. USB OUT OWNER (Offset 0x022C)

Position	31	30	29	28	27	26	25	24
Field	OUTBUF1_OWNER[15:8]							
Reset	0000 0000							
Access	RW							
Position	23	22	21	20	19	18	17	16
Field	OUTBUF1_OWNER[7:0]							
Reset	0000 0000							
Access	RW							
Position	15	14	13	12	11	10	9	8
Field	OUTBUF0_OWNER[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	OUTBUF0_OWNER[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
OUTBUF0_OWNER	15:0	OUT Buffer 0 Owner. Each bit represents buffer 0 ownership of the corresponding endpoint.	0: This endpoint OUT buffer 0 is owned by the software 1: This endpoint OUT buffer 0 is owned by the USB controller
OUTBUF1_OWNER	31:16	OUT Buffer 1 Owner. Each bit represents buffer 0 ownership of the corresponding endpoint.	0: This endpoint OUT buffer 1 is owned by the software 1: This endpoint OUT buffer 1 is owned by the USB controller



28.3.10 USB IN Interrupt Register (USB_IN_INT, Offset 0x0230)

The USB IN Interrupt Register contains the interrupt bits of an IN interrupt transfer. Each of the INBAV bits represents an interrupt from the corresponding endpoint IN buffer. All of the interrupts are set by hardware. Once an interrupt bit has been set, it may only be cleared by writing a one to the respective interrupt bit.

For an IN transfer, the SIE update this register after it has successfully transmit an IN packet to the host and receives the ACK handshake from the host. This indicates that the endpoint buffer is available for software writing.

This register will contain the value of 00h after a reset or USB device reset.

Table 28-13. USB IN INT (Offset 0x0230)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	INBAV[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	INBAV[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
INBAV	15:0	IN Buffer Available Interrupt.	0: No interrupt pending at this endpoint 1: An interrupt pending at this endpoint IN buffer
RFU	31:16	Reserved	N/A



28.3.11 USB OUT Interrupt Register (USB_OUT_INT, Offset 0x0234)

The USB OUT Interrupt Register contains the interrupt bits of an OUT interrupt transfer. Each of the OUTDAV bits represents an interrupt from the corresponding endpoint OUT buffer. All of the interrupts are set by hardware. Once an interrupt bit has been set, it may only be cleared by writing a one to the respective interrupt bit.

For an OUT endpoint, the SIE update this register with the endpoint index when it has successfully received an OUT packet from the host. This indicates that the endpoint buffer is ready for software reading.

This register will contain the value of 00h after a reset or USB device reset.

Table 28-14. USB_OUT_INT (Offset 0x0234)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	OUTDAV[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	OUTDAV[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
OUTDAV	15:0	OUT Buffer Data Available Interrupt.	0: No interrupt pending at this endpoint 1: An interrupt pending at this endpoint OUT buffer
RFU	31:16	Reserved	N/A



28.3.12 USB NAK Interrupt Register (USB_NAK_INT, Offset 0x0238)

The USB NAK Interrupt Register contains the interrupt bits of an NAK response. Each of the NAK bits represents an interrupt from the corresponding endpoint IN buffer. All of the interrupts are set by hardware. Once an interrupt bit has been set, it may only be cleared by writing a one to the respective interrupt bit.

The SIE update this register after an NAK handshake is sent to the host in response to an IN request. This indicates that the endpoint buffer has no data to be transmitted to the host (EP_IN_BUFX_OWNER.x=0).

This register will contain the value of 00h after a reset or USB device reset.

Table 28-15. USB_NAK_INT (Offset 0x0238)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	NAK[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	NAK[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
NAK	15:0	Endpoint NAK Interrupt	0: No interrupt pending at this endpoint 1: An interrupt pending at this endpoint IN buffer
RFU	31:16	Reserved	N/A

28.3.13 USB DMA Error Interrupt Register (USB_DMA_ERR_INT, Offset 0x023C)

The USB DMA Error Interrupt Register contains the interrupt bits of a DMA error. Each of the DMA_ERR bits represents an interrupt from the corresponding endpoint. All of the interrupts are set by hardware. Once an interrupt bit has been set, it may only be cleared by writing a one to the respective interrupt bit.

These bits are set when a USB interface has requested a DMA access to the system memory, but has not been given the bus in a timely manner. If processing an IN transfer, this would cause a transmit data underflow condition. Or if processing an OUT transfer, this would cause a receive data overflow condition.

This register will contain the value of 00h after a reset or USB device reset.

Table 28-16. USB DMA ERR INT (Offset 0x023C)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	DMA_ERR[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	DMA_ERR[7:0]							
Reset	0000 0000							
Access	RW							
Name	Bits	Description	Settings					
DMA_ERR	15:0	DMA Error Interrupt	0: No DMA error has been detected at this endpoint 1: A DMA interrupt pending at the endpoint with index corresponding to the bit index					
RFU	31:16	Reserved	N/A					



28.3.14 USB Buffer Overflow Interrupt Register (USB_BUF_OVR_INT, Offset 0x0240)

The USB Buffer Overflow Interrupt Register contains the interrupt bits of a buffer overflow error. Each of the BUF_OVR bits represents an interrupt from the corresponding endpoint. All of the interrupts are set by hardware. Once an interrupt bit has been set, it may only be cleared by writing a one to the respective interrupt bit.

The SIE sets the interrupt flag if a data packet to or from the host is larger than the buffer size that is allocated in the endpoint descriptor. In this case the data packet is truncated as it is put into buffer memory.

This register will contain the value of 00h after a reset or USB device reset.

Table 28-17. USB_BUF_OVR_INT (Offset 0x0240)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	BUF_OVR[15:8]							
Reset	0000 0000							
Access	RW							
Position	7	6	5	4	3	2	1	0
Field	BUF_OVR[7:0]							
Reset	0000 0000							
Access	RW							

Name	Bits	Description	Settings
BUF_OVR	15:0	Buffer Overflow Interrupt	0: No Buffer Overflow has been detected at this endpoint 1: A Buffer Overflow interrupt pending at the endpoint with index corresponding to the bit index
RFU	31:16	Reserved	N/A



28.3.15 USB SETUP Register 0 (USB_SETUP0, Offset 0x0260)

The USB SETUP Register0 contains the first 4 bytes of a SETUP packet.

The SIE updates this register with a SETUP packet directed to endpoint 0. Once a SETUP packet has been received successfully, the SIE will set the SETUP interrupt flag and generate an interrupt if enabled. The SIE will overwrite the SETUP registers content whenever it receives a new SETUP packets.

This register will contain the value of 00h after a reset or USB device reset.

Table 28-18. USB_SETUP0 (Offset 0x0260)

Position	31	30	29	28	27	26	25	24
Field	SETUP_BYT3[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	SETUP_BYT2[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	SETUP_BYT1[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	SETUP_BYT0[7:0]							
Reset	0000 0000							
Access	R							

Name	Bits	Description	Settings
SETUP_BYT0	7:0	SETUP Byte 0.	This is byte 0 of the SETUP packet.
SETUP_BYT1	15:8	SETUP Byte 1.	This is byte 1 of the SETUP packet.
SETUP_BYT2	23:16	SETUP Byte 2.	This is byte 2 of the SETUP packet.
SETUP_BYT3	31:24	SETUP Byte 3.	This is byte 3 of the SETUP packet.



28.3.16 USB SETUP Register 1 (USB_SETUP1, Offset 0x0264)

The USB SETUP Register1 contains the last 4 bytes of a SETUP packet.

The SIE updates this register with a SETUP packet directed to endpoint 0. Once a SETUP packet has been received successfully, the SIE will set the SETUP interrupt flag and generate an interrupt if enabled. The SIE will overwrite the SETUP registers content whenever it receives a new SETUP packets.

This register will contain the value of 00h after a reset or USB device reset.

Table 28-19. USB_SETUP1 (Offset 0x0264)

Position	31	30	29	28	27	26	25	24
Field	SETUP_BYTEx[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	SETUP_BYTEx[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	SETUP_BYTEx[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	SETUP_BYTEx[7:0]							
Reset	0000 0000							
Access	R							

Name	Bits	Description	Settings
SETUP_BYTEx4	7:0	SETUP Byte 4.	This is byte 4 of the SETUP packet.
SETUP_BYTEx5	15:8	SETUP Byte 5.	This is byte 5 of the SETUP packet.
SETUP_BYTEx6	23:16	SETUP Byte 6.	This is byte 6 of the SETUP packet.
SETUP_BYTEx7	31:24	SETUP Byte 7.	This is byte 7 of the SETUP packet.



28.3.17 USB Endpoint *n* Control Registers (USB_EP*n*, Offset 0x0280+4*n*)

The USB Endpoint Control Registers contains the control and configuration bits for each of the 16 endpoints. These four bits define all of the control necessary for any one endpoint. The formats for these registers are shown in the table on the following page.

Endpoint Control Registers for endpoint 1 to 15 are cleared to 00h on reset or USB device reset.

Endpoint 0 operates as the control pipe by default. It is set to 03h on reset or USB device reset.

Table 28-20. USB_EP*n* (Offset 0x0280+4*n*)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[4:0]					EP_ST_ACK	EP_ST_STALL	EP_STALL
Reset	00 0000					0	0	0
Access	R					RW	RW	RW
Position	7	6	5	4	3	2	1	0
Field	RFU	EP_DT	EP_NAK_EN	EP_INT_EN	EP_BUF2	RFU	EP_DIR[1:0]	
Reset	0	0	0	0	0	0	00	
Access	R	RW	RW	RW	RW	R	RW	

Name	Bits	Description	Settings
EP_DIR	1:0	<p>Endpoint Direction. EP_DIR for endpoint 0 are set to 11b and are read only.</p> <p>Only endpoint 0 can be configured as SETUP. All the other endpoints can only be configured as either IN or OUT.</p> <p>When an endpoint is disabled, it will not respond to any host request. In addition, the endpoint will reset its internal state such as clearing the data toggle.</p>	<p>00: Endpoint disable 01: OUT 10: IN 11: CONTROL. This Configure the endpoint to be a CONTROL pipe accepting IN, OUT and SETUP packets.</p>
RFU	2	Reserved	N/A



Name	Bits	Description	Settings
EP_BUF2	3	Double Buffer Enable. For a single buffered endpoint, the SIE will only use buffer 0 associated with the endpoint transfer direction. For a doubled buffered endpoint, the SIE will ping pong between the two buffers. This allows the SIE to transfer data to/from system memory while still communicating data to/from the host.	0: The endpoint is single buffered 1: The endpoint is double buffered
EP_INT_EN	4	Endpoint Interrupt Enable. The SETUP interrupt generation is controlled by USB_DEV_INT_EN.SETUP.	0: Endpoint interrupt is disabled 1: The SIE generates an interrupt upon a successful endpoint transfer (IN or OUT transfer) and the corresponding qualifier (USB_DEV_INT_EN.EP_IN or USB_DEV_INT_EN.EP_OUT) is enabled.
EP_NAK_EN	5	Endpoint NAK Interrupt Enable	0: No interrupt is generated when this endpoint return a NAK handshake 1: The SIE generates an interrupt when this endpoint return a NAK handshake and the corresponding qualifier (USB_DEV_INT_EN.EP_NAK) is enabled.
EP_DT	6	Endpoint Data Toggle Clear	The endpoint Data Toggle Clear is used to reset the data toggle field to DATA0 and reset the current buffer to buffer 0. In addition, the IN and OUT Buffer Owner bits associated with this endpoint will also be cleared to 0. Writing a '0' to this bit is ignored. This bit is set to 1 by software and cleared to 0 by SIE upon resetting internal DATA0/1 toggle field and buffer index. This bit is also cleared to 0 by USB reset.
RFU	7	Reserved	N/A
EP_STALL	8	Endpoint Stalled. Any access to a stalled endpoint will cause the USB interface to return a STALL handshake. For CONTROL endpoint, this bit is automatically cleared to 0 upon receiving a SETUP packet.	0: The endpoint is not stalled. 1: The endpoint is stalled



Name	Bits	Description	Settings
EP_ST_STALL	9	<p>Stall Status Stage of Control Transfer. This bit determines whether a STALL will be returned to the Status Stage of a control transfer. The Stall Status Stage bit, together with the EP_ST_ACK bit, indicates to the SIE how to respond in the status stage of a Control transfer. The EP_ST_STALL has priority over EP_ST_ACK. Until software either acknowledges (EP_ST_ACK=1) or stalls the transfer (EP_ST_STALL=1), the SIE answers the status stage of a CONTROL transfer with the NAK handshake.</p> <p>Regardless of the setting of this bit, a CONTROL endpoint must receive a SETUP packet and respond with ACK if the packet is received successfully. Upon receiving a SETUP packet, this bit is automatically cleared to 0.</p>	0: Do not send STALL 1: Send STALL to the host
EP_ST_ACK	10	<p>Acknowledge Status Stage of Control Transfer. This bit determines whether an ACK will be returned to the Status Stage of a control transfer. Until software either acknowledges (EP_ST_ACK=1) or stalls the transfer (EP_ST_STALL=1), the SIE answers the status stage of a CONTROL transfer with the NAK handshake.</p> <p>Regardless of the setting of this bit, a CONTROL endpoint must receive a SETUP packet and respond with ACK if the packet is received successfully. Upon receiving a SETUP packet, this bit is automatically cleared to 0.</p>	0: Do not send ACK. 1: Send ACK to the host
RFU	31:11	Reserved	N/A

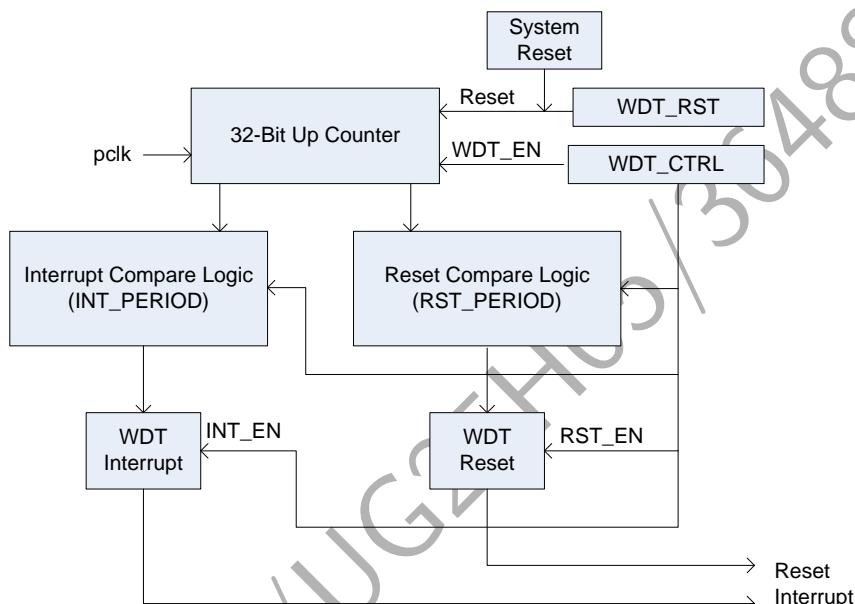
29.0 Watchdog Timer

The watchdog timer protects against corrupt or unreliable software, power faults, and other system-level problems, which may place the microcontroller into unsuitable operating states. When the application is working correctly, application software will **periodically** reset the watchdog counter. If the watchdog timer interrupt is enabled and the software does not reset the counter within the interrupt time period (INT_PERIOD), the watchdog timer will generate a watchdog timer interrupt. If the watchdog timer reset is enabled and the software does not reset the counter within the reset time period (RST_PERIOD), the watchdog timer will generate a system reset.

The watchdog timer features sixteen programmable time delay periods, 2^{16} through 2^{31} clock cycles for both the interrupt and reset periods.

The watchdog timer counter, which is not directly accessible, is reset on all forms of reset.

Figure 29-1. Watchdog Timer Block Diagram



29.1 Enabling and Disabling the Watchdog Timer

The application software must set the WDT_EN bit to enable the watchdog timer. Because the watchdog timer is free-running, the following procedure must be followed when enabling the watchdog timer to prevent an unintended reset during the enable process.

1. Write 0x0000_00A5 to WDT_RST
2. Write 0x0000_005A to WDT_RST
3. Set WDT_EN bit

The watchdog timer can be disabled in two ways:

- The application software can clear WDT_EN bit to “0” or,
- By a POR, which will clear WDT_EN to “0”.

The interrupt and reset signals from the watchdog timer can also be enabled or disabled independently through the INT_EN and RST_EN bits of the WDT_CTRL register.



29.2 Time Delay Period Selection

Two independent time delay periods may be specified in the watchdog timer:

- INT_PERIOD—Specify the time delay before the watchdog timer sends an interrupt to the CPU.
- RST_PERIOD—Specify the time between the reset of the time delay before the watchdog timer sends a system reset request. In a correctly operating system application software will reset the watchdog timer in the watchdog timer interrupt routine before the watchdog counter exceeds the reset period, so the reset would never occur.

INT_PERIOD must be less than **RST_PERIOD** or the interrupt will never occur. The interrupt routine must reset the watchdog counter by writing 0x0000_00A5 then 0x0000_005A to the WDT_RST register before the RST_PERIOD or the watchdog timer will generate a system interrupt, if enabled.

There are sixteen choices of time delay periods for the watchdog timer: 2^{16} through 2^{31} PCLK cycles. The time delay for a specific clock source may be calculated as follows:

$$\begin{aligned}\text{Interrupt Period} &= \text{Clock Source Period} \times \text{WDT_CTRL.INT_PERIOD} \\ \text{Reset Period} &= \text{Clock Source Period} \times \text{WDT_CTRL.RST_PERIOD}\end{aligned}$$

Examples of approximate interrupt period for different watchdog timer INT_PERIOD are listed in Table 29-1 below.

Table 29-1. Watchdog Timer Interrupt Period Example

Clock Source	INT_PERIOD	Interrupt Period
54MHz PCLK	1111 (2^{16})	1,213µs
54MHz PCLK	1011 (2^{20})	19ms
54MHz PCLK	0110 (2^{25})	621ms
54MHz PCLK	0000 (2^{31})	40 s

29.3 Watchdog Timer Operation

Utilizing the watchdog timer in the application software is straightforward. As early as possible in the application software enable the watchdog timer interrupt and watchdog timer reset. Periodically the application software must write to the WDT_RST register to reset the watchdog counter. If program execution has become lost, the watchdog timer interrupt will occur, giving the system a “last chance” to recover from whatever circumstance caused the improper code execution. The interrupt routine can either attempt to repair the situation or allow the watchdog timer reset to occur. In the event of a system software failure, the interrupt will not be executed and the watchdog system reset will recover operation.

At the earliest possible point the reset routine in the application software, the RST_FLAG bit should be interrogated to see if the reset was caused by the watchdog timer. If so, application software should assume that there was a program execution error and take whatever steps necessary to guard against a software corruption issue.

29.4 Watchdog Timer Registers

Address assignments for registers are outlined below.

Table 29-2. Watchdog Timer Register Addresses

Offset Address	Access	Register	Register Set
0x0000	RW	WDT_CTRL	Watchdog Timer Control Register
0x0004	RW	WDT_RST	Watchdog Timer Reset Register

Refer to section 3.0 Digital Peripherals and System Control Registers for details on reset values and various access modes.



29.4.1 Watchdog Timer Control Register (WDT_CTRL, Offset 0x0000)

This register is reset only by a POR, and is unaffected by all other reset.

Table 29-3. WDT_CTRL (Offset 0x0000)

Position	31	30	29	28	27	26	25	24
Field	RST_FLAG				RFU[6:0]			
POR	0				000 0000			
Reset	S				000 0000			
Access	RW				R			
Position	23	22	21	20	19	18	17	16
Field				RFU[7:0]				
POR				0000 0000				
Reset				0000 0000				
Access				R				
Position	15	14	13	12	11	10	9	8
Field			RFU[3:0]		RST_EN	INT_EN	INT_FLAG	WDT_EN
POR			0000		0	0	0	0
Reset			0000		N/A	N/A	N/A	N/A
Access			R		RW	RW	RW	RW
Position	7	6	5	4	3	2	1	0
Field			RST_PERIOD[3:0]			INT_PERIOD[3:0]		
POR			0000			0000		
Reset			N/A			N/A		
Access			RW			RW		

Name	Bits	Description	Settings	
			INT_PERIOD	Clock Cycles
INT_PERIOD	3:0	Watchdog Interrupt Period. The watchdog timer will assert an interrupt, if enabled, if the CPU does not write the watchdog reset sequence to the WDT_RST register before the watchdog timer has counted this time period since the last timer reset.	1111 1110 1101 1100 1011 1010 1001 1000 0111 0110 0101 0100 0011 0010 0001 0000	2^{16} 2^{17} 2^{18} 2^{19} 2^{20} 2^{21} 2^{22} 2^{23} 2^{24} 2^{25} 2^{26} 2^{27} 2^{28} 2^{29} 2^{30} 2^{31}

Name	Bits	Description	Settings
RST_PERIOD	7:4	Watchdog Reset Period. The watchdog timer will assert a reset, if enabled, if the CPU does not write the watchdog reset sequence to the WDT_RST register before the watchdog timer has counted this time period since the last timer reset.	See watchdog interrupt period setting
WDT_EN	8	Watchdog Timer Enable.	0: Watchdog timer is disabled 1: Watchdog timer is enabled
INT_FLAG	9	Watchdog Timer Interrupt Flag.	0: Watchdog timer interrupt is not pending. 1: Watchdog timer time-out has occurred. An interrupt is pending.
INT_EN	10	Watchdog Timer Interrupt Enable.	0: Watchdog timer interrupt disabled 1: Watchdog timer will cause an interrupt if INT_FLAG=1
RST_EN	11	Watchdog Timer Reset Enable.	0: Watchdog timer reset disabled 1: Watchdog timer will cause a reset if enabled.
RFU	30:12	Reserved	
RST_FLAG	31	Watchdog Timer Reset Flag.	0: Previous reset was not caused by Watchdog timer 1: Previous reset was caused by Watchdog timer



29.4.2 Watchdog Timer Reset Register (WDT_RST, Offset 0x0004)

Table 29-4. WDT_RST (Offset 0x0004)

Position	31	30	29	28	27	26	25	24
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	23	22	21	20	19	18	17	16
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	15	14	13	12	11	10	9	8
Field	RFU[7:0]							
Reset	0000 0000							
Access	R							
Position	7	6	5	4	3	2	1	0
Field	WDT_RST[7:0]							
Reset	0000 0000							
Access	W							

Name	Bits	Description	Settings
WDT_RST	7:0	Watchdog Timer Reset Register. Writing the watchdog counter 'reset sequence' to this register resets the watchdog counter. If the watchdog count exceeds INT_PERIOD then a watchdog interrupt will occur, if enabled. If the watchdog count exceeds RST_PERIOD then a watchdog reset will occur, if enabled.	The reset sequence is 0x0000_00A5 followed by 0x0000_005A. Always reads zero.
RFU	31:8	Reserved.	

**Revision History**

02/27/2014	Rev. A Release Change SC_PN.VCCSEL bits from R to RW. Add a note to Smart Card SC_SR.CCOV to clarify its behavior. Add a note to clarify the interrupt vector table alignment. Change the MEMZCN.NVSRAM bit to RFU. Update Figure 24-1. Clarify the AESKT bit behavior during a DRS. Change the number of SPI instances to three. Update SECALM Register access field. Update KBD_ISR Register access field. Update PRPOL bit definition. Remove I2C extended mode. Fix memory map typos. Update DSPEN bit description. Fix typo in external oscillator frequency. Clarify GPIO2 port reset states and directions. Clarify that PLL1 is the source clock of DAC.
05/30/2014	Rev. B Release 6/24/2014
	Rev. C Release Add the OSCPD description to PM register. Fix typo in SECDIAG.EXTSTAT and SECALM.EXTSTAT bit definitions. Update features of MAX32551. Update RTC chapter to clarify that the square wave output is not available on revision A silicon. Update the RTC_CN.ADE and RTC_CN.RDY access field. Fix typos in address offsets of CACHE_CTRL and INVALIDATE registers. Clarify port number on the JTAG related GPIO registers. Fix the typo in MAA memory address. Add ACNTL Access Control Register to the Flash Controller Chapter. Add the flash page numbers and size of each page. Fix the flash erase address range. Add DAC data in register. Clarify the reset value and access to the BSTAPEN bit. Remove the UART1 bit from RSTR register. Correct number of interrupts under chapter 6. Change the reset value of TMR_CMP register from 0xFFFF_FFFF to 0x0000_FFFF. Remove the DMA access to flash. Add system bus interconnect. Change the GPIO Tertiary function to Secondary function and Secondary function to Primary. Update the ADC conversion formula.
10/30/2014	5/5/2015
	Rev. D Release Fix typo in Watchdog WDT_CTRL register. Fix typo in USB INBUF1_OWNER register. Fix typo in USB Device Registers. Update the GPIO states on POR. Update the SC_ETUR register reset value. Remove DAC data in register (DAC_REG). Add DAC_FIFO and update DAC chapter. Remove I2C slave mode. Update MSR chapter with changes on B1 silicon. Change the reset value of BATLO to zero. Add BORF flag to SECDIAG bit 0. (only on B1 silicon forward) Remove MAX32551 from the document. Remove DMA support for I2C on revision A silicon. Add VBATTMR to RTC_TRIM register. Add Smart card Dual Mode and Sequencer support for revision B of silicon.

10/30/2015

- Add Cryptographic Accelerator for revision B of silicon.
- Add PWMCKBD bit to TMR_CN register for revision B silicon.
- Recalculated the Watchdog Timer Interrupt intervals based on 54MHz.
- Change ADC reference voltage to external reference instead of AVDD.
- Remove FLSH_PREBOOT from FLSH_CN register.
- Clarify the usecase of FLASH_PAGE_FLIP in SCON register.

01/07/2016

- Rev. E Release
- Rename the 0x10100000 of the memory map to Maxim OTP.
- Fix the maximum SPI FIFO Counts.
- Fix the AESKT definition. AESKT can only be erased by a BOR.
- Fix the RTC output frequency when FT is 1x.
- Add a note on the USB suspend conditions under section StandBy Mode.
- Change the 5:5:5 mode of the TFT to 5:5:6. Clarified the mode.