



# **XR871 User Manual**

---

**Revision 1.02**

**Aug 25, 2017**

## Declaration

THIS DOCUMENTATION IS THE ORIGINAL WORK AND COPYRIGHTED PROPERTY OF XRADO TECHNOLOGY ("XRADO"). REPRODUCTION IN WHOLE OR IN PART MUST OBTAIN THE WRITTEN APPROVAL OF XRADO AND GIVE CLEAR ACKNOWLEDGEMENT TO THE COPYRIGHT OWNER.

THE INFORMATION FURNISHED BY XRADO IS BELIEVED TO BE ACCURATE AND RELIABLE. XRADO RESERVES THE RIGHT TO MAKE CHANGES IN CIRCUIT DESIGN AND/OR SPECIFICATIONS AT ANY TIME WITHOUT NOTICE. XRADO DOES NOT ASSUME ANY RESPONSIBILITY AND LIABILITY FOR ITS USE. NOR FOR ANY INFRINGEMENTS OF PATENTS OR OTHER RIGHTS OF THE THIRD PARTIES WHICH MAY RESULT FROM ITS USE. NO LICENSE IS GRANTED BY IMPLICATION OR OTHERWISE UNDER ANY PATENT OR PATENT RIGHTS OF XRADO. THIS DATASHEET NEITHER STATES NOR IMPLIES WARRANTY OF ANY KIND, INCLUDING FITNESS FOR ANY PARTICULAR APPLICATION.

THIRD PARTY LICENCES MAY BE REQUIRED TO IMPLEMENT THE SOLUTION/PRODUCT. CUSTOMERS SHALL BE SOLELY RESPONSIBLE TO OBTAIN ALL APPROPRIATELY REQUIRED THIRD PARTY LICENCES. XRADO SHALL NOT BE LIABLE FOR ANY LICENCE FEE OR ROYALTY DUE IN RESPECT OF ANY REQUIRED THIRD PARTY LICENCE. XRADO SHALL HAVE NO WARRANTY, INDEMNITY OR OTHER OBLIGATIONS WITH RESPECT TO MATTERS COVERED UNDER ANY REQUIRED THIRD PARTY LICENCE.

## Revision History

Revision	Date	Summary of Changes
1.0	2017-5-9	Initial Version
1.01	2017-8-25	XR871 to XR871GT/ET (2.2 Features and 3.2 Clock) Remove Chapter 8 and Chapter 9, which are included in Datasheet Remove useless content Change Logo
1.02	2017-10-25	Update CSI Register, remove some undefined field

**Table 1-1 Revision History**

## Contents

Declaration.....	2
Revision History .....	3
Contents.....	4
Tables .....	9
Figures .....	22
Chapter 1     About This Documentation.....	24
1.1     Documentation Overview.....	24
1.2     Acronyms and Abbreviations.....	24
Chapter 2     Overview.....	26
2.1     General Description .....	26
2.2     Features .....	26
2.3     Application .....	27
2.4     Block Diagram .....	28
Chapter 3     System.....	29
3.1     Power Management .....	29
3.2     Clock.....	29
3.3     Memory Mapping .....	30
3.4     CPU System .....	31
3.4.1     Register List.....	31
3.4.2     Interrupt Source Description .....	31
3.5     Global Power, Reset and Clock Manager (GPRCM).....	33
3.5.1     Overview.....	33
3.5.2     Register List.....	33
3.5.3     Register Description .....	34
3.6     Clock Control Module (CCM) .....	50
3.6.1     Overview.....	50
3.6.2     Register List.....	50

3.6.3	Register Description .....	50
3.7	Direct Memory Access Controller (DMA).....	64
3.7.1	Overview.....	64
3.7.2	Register List.....	64
3.7.3	Register Description .....	65
3.8	Real Time Clock (RTC) .....	72
3.8.1	Overview .....	72
3.8.2	Register List .....	72
3.9	Timer Controller (Timer) .....	79
3.9.1	Overview .....	79
3.9.2	Register List .....	80
3.9.3	Register Description .....	80
3.10	Security ID (SID) .....	87
3.10.1	Overview .....	87
3.10.2	Register List .....	87
3.10.3	Register Description .....	88
3.11	Message Box (MSGBOX) .....	90
3.11.1	Overview .....	90
3.11.2	Typical Applications .....	91
3.11.3	Functional Block Diagram .....	93
3.11.4	Operation Principle .....	93
3.11.5	Register List .....	94
3.11.6	Register Description .....	95
3.12	Spinlock.....	102
3.12.1	Overview .....	102
3.12.2	Functionalities Description .....	103
3.12.3	Operation Principle .....	104
3.12.4	Register List .....	104
3.12.5	Register Description .....	104

3.13	Secure System (SS)/Crypto Engine (CE) .....	106
3.13.1	Overview .....	106
3.13.2	Register List .....	107
3.13.3	Register Description .....	108
Chapter 4	Peripherals .....	115
4.1	Universal Asynchronous Receivers/Transmitters (UART).....	115
4.1.1	Overview .....	115
4.1.2	Register List .....	116
4.1.3	Register Description .....	117
4.2	Serial Peripheral Interface (SPI) .....	136
4.2.1	Overview .....	136
4.2.2	Register List .....	137
4.2.3	Register Description .....	138
4.3	Two Wire Interface Controller (TWI).....	151
4.3.1	Overview .....	151
4.3.2	Register List .....	152
4.3.3	Register Description .....	153
4.4	SDMMC/SDIO Controller (SDC).....	159
4.4.1	Overview .....	159
4.4.2	Register List .....	160
4.4.3	Register Description .....	161
4.4.4	SD/MMC DMA Controller Description .....	186
4.5	Pulse Width Modulation/Event Capture Controller (PWM/ECT).....	189
4.5.1	Overview .....	189
4.5.2	Operation and Functional Description .....	190
4.5.3	Register List .....	197
4.5.4	Register Description .....	198
4.6	CIR Receiver Controller (IRRX) .....	216
4.6.1	Overview .....	216

4.6.2	Register List .....	217
4.6.3	Register Description .....	217
4.7	CIR Transmit Controller (IRTX).....	222
4.7.1	Overview .....	222
4.7.2	Register List .....	222
4.7.3	Register Description .....	223
4.8	General Purpose ADC (GPADC) .....	229
4.8.1	Overview .....	229
4.8.2	GPADC Block Diagram .....	230
4.8.3	Operations and Functional Descriptions.....	230
4.8.4	Register List .....	231
4.8.5	Register Description .....	231
Chapter 5	Audio Interface .....	238
5.1	Digital Audio Controller (DAUDIO).....	238
5.1.1	Overview .....	238
5.1.2	Interface Block Diagram.....	238
5.1.3	Operations and Functional Descriptions.....	239
5.1.4	Digital Audio Interface Transmit Format.....	240
5.1.5	Operation Modes .....	242
5.1.6	Register List .....	243
5.1.7	Register Description .....	244
5.2	Digital Microphone Controller (DMIC).....	259
5.2.1	Overview .....	259
5.2.2	DMIC Block Diagram .....	259
5.2.3	DMIC Clock Source .....	259
5.2.4	Register List .....	260
5.2.5	Register Description .....	260
Chapter 6	Camera .....	270
6.1	Camera Serial Interface (CSI) .....	270

---

6.1.1	Overview.....	270
6.1.2	Register List.....	271
6.1.3	Register Description.....	272
Chapter 7	GPIO .....	279
7.1.1	Register List.....	279
7.1.2	Register Description.....	280
7.1.3	Pin Multiplexing .....	285

## Tables

<b>Table 1-1 Revision History .....</b>	<b>3</b>
<b>Table 3-1 NVIC Register List .....</b>	<b>31</b>
<b>Table 3-2 Interrupt Source.....</b>	<b>33</b>
<b>Table 3-3 GPRCM Register List .....</b>	<b>34</b>
<b>Table 3-4 System DCDC Control Register .....</b>	<b>35</b>
<b>Table 3-5 System Power Control 1 Register .....</b>	<b>36</b>
<b>Table 3-6 System LFCLK Control Register.....</b>	<b>37</b>
<b>Table 3-7 System HOSC Type Register .....</b>	<b>37</b>
<b>Table 3-8 System RCOSC Calibration Control Register.....</b>	<b>38</b>
<b>Table 3-9 System PLL Control Register .....</b>	<b>38</b>
<b>Table 3-10 System Clock 1 Control Register.....</b>	<b>39</b>
<b>Table 3-11 System Clock 2 Control Register.....</b>	<b>39</b>
<b>Table 3-12 Audio Clock Control Register .....</b>	<b>40</b>
<b>Table 3-13 Device Clock Control Register .....</b>	<b>41</b>
<b>Table 3-14 Audio PLL Bias Register .....</b>	<b>41</b>
<b>Table 3-15 Audio PLL Pattern Control Register .....</b>	<b>42</b>
<b>Table 3-16 DCXO Control 0 Register .....</b>	<b>43</b>
<b>Table 3-17 System 1 Control Register.....</b>	<b>43</b>
<b>Table 3-18 System 1 Status Register.....</b>	<b>44</b>
<b>Table 3-19 System 2 Control Register.....</b>	<b>44</b>
<b>Table 3-20 System 2 Status Register.....</b>	<b>45</b>
<b>Table 3-21 DCXO Stable Reference Time Register .....</b>	<b>45</b>
<b>Table 3-22 AR400A Boot Flag Register .....</b>	<b>46</b>
<b>Table 3-23 AR400A Boot Address Register .....</b>	<b>46</b>
<b>Table 3-24 AR400A Boot Argument Register .....</b>	<b>47</b>
<b>Table 3-25 AR400A Private Register .....</b>	<b>47</b>
<b>Table 3-26 AR400A Wakeup Timer Counter Register .....</b>	<b>47</b>

---

Table 3-27 AR400A Wakeup Timer Value Register .....	48
Table 3-28 AR400A IO Wakeup Enable Register.....	48
Table 3-29 AR400A IO Wakeup Mode Register.....	48
Table 3-30 AR400A IO Wakeup Status Register .....	49
Table 3-31 AR400A IO Hold Control Register .....	49
Table 3-32 AR400A IO Wakeup Global Enable Register.....	49
Table 3-33 CCM Register List .....	50
Table 3-34 CPU Clock Control Register .....	51
Table 3-35 BUS Clock Gating Control Register .....	53
Table 3-36 BUS Module Reset Control Register .....	55
Table 3-37 SPIO Clock Control Register.....	56
Table 3-38 SPI1 Clock Control Register.....	57
Table 3-39 SDC Clock Control Register .....	57
Table 3-40 SS Clock Control Register .....	58
Table 3-41 DAUDIO Clock Control Register .....	59
Table 3-42 IRRX Clock Control Register .....	59
Table 3-43 IRTX Clock Control Register.....	60
Table 3-44 System Tick Reference Clock Control Register .....	61
Table 3-45 System Tick Clock Calibration Register .....	62
Table 3-46 DMIC Clock Control Register.....	62
Table 3-47 IRTX Clock Control Register.....	63
Table 3-48 CSI Device Clock Control Register.....	63
Table 3-49 SPIO Clock Control Register.....	64
Table 3-50 DMA Register List.....	65
Table 3-51 DMA IRQ Enable Register .....	67
Table 3-52 DMA IRQ Pending Status Register.....	69
Table 3-53 Normal DMA Configuration Register.....	71
Table 3-54 Normal DMA Source Address Register .....	71
Table 3-55 Normal DMA Destination Address Register.....	71

Table 3-56 Normal DMA Byte Counter Register.....	72
Table 3-57 RTC Register List.....	72
Table 3-58 RTC Status Control Register .....	73
Table 3-59 RTC YYMMDD Register .....	74
Table 3-60 RTC HHMMSS Register .....	75
Table 3-61 Alarm 0 Counter Register.....	75
Table 3-62 Alarm 0 Current Value Register .....	75
Table 3-63 Alarm 0 Enable Register .....	75
Table 3-64 Alarm 0 IRQ Enable Register .....	76
Table 3-65 Alarm 0 IRQ Status Register.....	76
Table 3-66 Alarm 1 Week HH-MM-SS Register .....	76
Table 3-67 Alarm 1 Enable Register .....	78
Table 3-68 Alarm 1 IRQ Enable Register .....	78
Table 3-69 Alarm 1 IRQ Enable Register .....	79
Table 3-70 Free Running Counter Low Register .....	79
Table 3-71 Free Running Counter High Register .....	79
Table 3-72 Timer Register List.....	80
Table 3-73 Timer IRQ Enable Register .....	81
Table 3-74 Timer IRQ Status Register .....	81
Table 3-75 Timer 0 Control Register.....	82
Table 3-76 Timer 0 Interval Value Register .....	82
Table 3-77 Timer 0 Current Value Register .....	83
Table 3-78 Timer 1 Control Register.....	84
Table 3-79 Timer 1 Interval Value Register .....	84
Table 3-80 Timer 1 Current Value Register .....	84
Table 3-81 Watchdog IRQ Enable Register.....	85
Table 3-82 Watchdog IRQ Enable Register.....	85
Table 3-83 Watchdog Control Register .....	85
Table 3-84 Watchdog Configuration Register .....	86

Table 3-85 Watchdog Mode Control Register .....	87
Table 3-86 Watchdog Output Control Register .....	87
Table 3-87 Security ID Register List .....	88
Table 3-88 SID Program/Read Control Register .....	89
Table 3-89 SID Program Key Value Register .....	89
Table 3-90 SID Program Key Value Register .....	89
Table 3-91 SID Burn Timing Register .....	90
Table 3-92 SID Debug Register .....	90
Table 3-93 SID Register .....	90
Table 3-94 MSGBOX Register List .....	95
Table 3-95 MSGBOX Control Register 0 .....	96
Table 3-96 MSGBOX Control Register 1 .....	97
Table 3-97 MSGBOX IRQ Enable Register u(u=0,1).....	99
Table 3-98 MSGBOX IRQ Status Register u .....	100
Table 3-99 MSGBOX FIFO Status Register m .....	101
Table 3-100 MSGBOX Message Status Register m .....	101
Table 3-101 MSGBOX Message Queue Register m .....	101
Table 3-102 MSGBOX Debug Register .....	102
Table 3-103 Spinlock Register List .....	104
Table 3-104 SPINLOCK System Status Register .....	105
Table 3-105 SPINLOCK Status Register .....	105
Table 3-106 SPINLOCK Register N (N=0 to 31) .....	105
Table 3-107 CE/SS Register List .....	107
Table 3-108 CE Control Register .....	110
Table 3-109 CE Key Register .....	110
Table 3-110 CE IV Register .....	110
Table 3-111 CE Counter Register .....	110
Table 3-112 CE FIFO Control/ Status Register .....	111
Table 3-113 CE Interrupt Control/ Status Register .....	112

Table 3-114 CE Message Digest Register .....	112
Table 3-115 CE CTS Length Register .....	113
Table 3-116 CE CRC Poly Register.....	113
Table 3-117 CE Result Register.....	113
Table 3-118 CE SHA256 Message Digest Register.....	113
Table 3-119 CE RX FIFO Register .....	113
Table 3-120 CE TX FIFO Register.....	114
Table 4-1 UART Register List .....	117
Table 4-2 UART Receiver Buffer Register.....	117
Table 4-3 UART Transmit Holding Register .....	118
Table 4-4 UART Divisor Latch Low Register .....	118
Table 4-5 UART Divisor Latch High Register.....	119
Table 4-6 UART Interrupt Enable Register .....	120
Table 4-7 UART Interrupt Identity Register .....	120
Table 4-8 UART Interrupt Identity Priority .....	121
Table 4-9 UART FIFO Control Register .....	123
Table 4-10 UART Line Control Register.....	125
Table 4-11 UART Modem Control Register .....	126
Table 4-12 UART Line Status Register.....	128
Table 4-13 UART Modem Status Register.....	131
Table 4-14 UART Scratch Register .....	131
Table 4-15 UART Status Register .....	132
Table 4-16 UART Transmit FIFO Level Register.....	132
Table 4-17 UART Receive FIFO Level Register .....	132
Table 4-18 UART Halt TX Register .....	134
Table 4-19 UART TX Delay .....	134
Table 4-20 UART Baudrate Detection Control Register .....	135
Table 4-21 UART Baudrate Detection Counter Low Register .....	135
Table 4-22 UART Baudrate Detection Counter High Register.....	135

---

Table 4-23 SPI Modes with Clock Polarity and Phase.....	136
Table 4-24 SPI Register List.....	137
Table 4-25 SPI Version Number Register .....	138
Table 4-26 SPI Global Control Register.....	139
Table 4-27 SPI Transfer Control Register .....	141
Table 4-28 SPI Interrupt Control Register .....	143
Table 4-29 SPI Interrupt Status Register.....	145
Table 4-30 SPI FIFO Control Register .....	146
Table 4-31 SPI FIFO Status Register.....	147
Table 4-32 SPI Wait Clock Register.....	148
Table 4-33 SPI Clock Control Register.....	148
Table 4-34 SPI Burst Counter Register .....	149
Table 4-35 SPI Transmit Counter Register.....	149
Table 4-36 SPI Master Burst Control Register .....	150
Table 4-37 SPI Normal DMA Mode Register .....	150
Table 4-38 SPI TX Data Register .....	151
Table 4-39 SPI RX Data Register .....	151
Table 4-40 TWI Register List .....	153
Table 4-41 TWI Slave Address Register .....	154
Table 4-42 TWI Extend Address Register .....	154
Table 4-43 TWI Data Register .....	154
Table 4-44 TWI Control Register .....	156
Table 4-45 TWI Status Register .....	157
Table 4-46 TWI Clock Register .....	158
Table 4-47 TWI Soft Reset Register .....	158
Table 4-48 TWI Enhance Feature Register .....	158
Table 4-49 TWI Line Control Register .....	159
Table 4-50 SD Register List.....	161
Table 4-51 SD Global Control Register.....	162

---

Table 4-52 SD Clock Control Register .....	163
Table 4-53 SD Timeout Register .....	163
Table 4-54 SD Card Width Register .....	164
Table 4-55 SD Clock Control Register .....	164
Table 4-56 SD Clock Control Register .....	164
Table 4-57 SD Command Register .....	167
Table 4-58 SD Command Argument Register.....	167
Table 4-59 SD Response 0 Register .....	167
Table 4-60 SD Response 1 Register .....	168
Table 4-61 SD Response 2 Register .....	168
Table 4-62 SD Response 0 Register .....	168
Table 4-63 SD Interrupt Mask Register.....	169
Table 4-64 SD Masked Interrupt Register.....	170
Table 4-65 SD Raw Interrupt Register .....	170
Table 4-66 SD Status Register .....	172
Table 4-67 SD Water Level Register.....	173
Table 4-68 SD Function Select Register .....	175
Table 4-69 SD Transferred CIU Card Byte Counter Register .....	175
Table 4-70 SD Transferred Host to BUI-FIFO Byte Counter Register .....	175
Table 4-71 SD Debug Control Register.....	176
Table 4-72 SD Auto CMD12 Argument Register .....	176
Table 4-73 SD Auto CMD12 Argument Register.....	177
Table 4-74 SD Hardware Reset Register .....	178
Table 4-75 SD DMAC Control Register.....	179
Table 4-76 SD Descriptor List Base Address Register .....	179
Table 4-77 SD DMA Status Register.....	181
Table 4-78 SD DMA Interrupt Enable Register .....	182
Table 4-79 SD Current Host Descriptor Address Register .....	183
Table 4-80 SD Current Buffer Descriptor Address Register.....	183

---

Table 4-81 SD Card Threshold Control Register .....	183
Table 4-82 eMMC DDR Start Bit Detection Register.....	184
Table 4-83 SD Response CRC Register .....	184
Table 4-84 SD DATA7 CRC Register .....	184
Table 4-85 SD DATA6 CRC Register .....	184
Table 4-86 SD DATA5 CRC Register .....	185
Table 4-87 SD DATA4 CRC Register .....	185
Table 4-88 SD DATA3 CRC Register .....	185
Table 4-89 SD DATA2 CRC Register .....	185
Table 4-90 SD DATA1 CRC Register .....	185
Table 4-91 SD DATA0 CRC Register .....	186
Table 4-92 SD CRC Status Register .....	186
Table 4-93 SD FIFO Register.....	186
Table 4-94 SDC Descriptor 0 Register .....	188
Table 4-95 SDC Descriptor 1 Register .....	188
Table 4-96 SDC Descriptor 2 Register .....	189
Table 4-97 SDC Descriptor 3 Register .....	189
Table 4-98 PWM/ECT Register List .....	198
Table 4-99 PWM IRQ Enable Register .....	199
Table 4-100 PWM IRQ Status Register .....	200
Table 4-101 Capture IRQ Enable Register .....	202
Table 4-102 PWM01 Clock Configuration Register.....	207
Table 4-103 PWM23 Clock Configuration Register.....	208
Table 4-104 PWM45 Clock Configuration Register.....	209
Table 4-105 PWM67 Clock Configuration Register.....	210
Table 4-106 PWM01 Dead Zone Control Register .....	210
Table 4-107 PWM23 Dead Zone Control Register .....	210
Table 4-108 PWM45 Dead Zone Control Register .....	211
Table 4-109 PWM67 Dead Zone Control Register .....	211

Table 4-110 PWM Enable Register .....	212
Table 4-111 Capture Enable Register .....	213
Table 4-112 PWM Control Register .....	214
Table 4-113 PWM Period Control Register .....	215
Table 4-114 PWM Counter Register .....	215
Table 4-115 Capture Control Register .....	215
Table 4-116 Capture Rise Lock Register.....	216
Table 4-117 Capture Fall Lock Register.....	216
Table 4-118 CIR Receiver Register List.....	217
Table 4-119 CIR Active Pulse Accept Mode Control Register .....	217
Table 4-120 CIR Receiver Control Register.....	218
Table 4-121 CIR Receiver Configure Register .....	218
Table 4-122 CIR Receiver FIFO Register .....	218
Table 4-123 CIR Receiver Interrupt Control Register .....	219
Table 4-124 CIR Receiver Status Register .....	220
Table 4-125 CIR Receiver Controller Configure Register.....	221
Table 4-126 CIR Transmitter Register List .....	223
Table 4-127 CIR Transmit Global Register.....	224
Table 4-128 CIR Transmit Modulation Control Register .....	224
Table 4-129 CIR Transmit Control Register .....	225
Table 4-130 CIR Transmit Idle Duration Counter Register .....	225
Table 4-131 CIR Transmit Idle Duration Counter Register .....	225
Table 4-132 CIR Transmit Idle Counter Register.....	226
Table 4-133 CIR Transmit Idle Counter Register.....	226
Table 4-134 CIR Transmit FIFO Empty Register .....	226
Table 4-135 CIR Transmit Interrupt Control Register.....	227
Table 4-136 CIR Transmit FIFO Available Counter Register.....	227
Table 4-137 CIR Transmit Status Register .....	228
Table 4-138 CIR Transmit Threshold Register .....	228

---

Table 4-139 CIR DMA Control Register .....	229
Table 4-140 CIR Transmit FIFO Data Register .....	229
Table 4-141 ADC Signals .....	230
Table 4-142 Timer Register List .....	231
Table 4-143 GPADC Sample Rate Configure Register .....	231
Table 4-144 GPADC Control Register .....	232
Table 4-145 GPADC Compare and Select Enable Register .....	233
Table 4-146 GPADC FIFO Interrupt Control Register .....	233
Table 4-147 GPADC FIFO Interrupt Status Register .....	234
Table 4-148 GPADC FIFO Data Register .....	234
Table 4-149 GPADC Calibration Data Register .....	235
Table 4-150 GPADC Low Interrupt Configure Register .....	235
Table 4-151 GPADC HIGH Interrupt Configure Register .....	235
Table 4-152 GPADC DATA Interrupt Configure Register .....	235
Table 4-153 GPADC Low Interrupt Status Register .....	236
Table 4-154 GPADC High Interrupt Status Register .....	236
Table 4-155 GPADC Data Interrupt Status Register .....	236
Table 4-156 GPADC CHn Compare Data Register .....	237
Table 4-157 GPADC CHn Data Register .....	237
Table 5-1 Digital Audio Interface External Signals .....	239
Table 5-2 Digital Audio Interface Clock Sources .....	240
Table 5-3 Digital Audio Register List .....	244
Table 5-4 Digital Audio Control Register .....	245
Table 5-5 Digital Audio Format Register 0 .....	246
Table 5-6 Digital Audio Format Register 1 .....	247
Table 5-7 Digital Audio Interrupt Status Register .....	248
Table 5-8 Digital Audio RX FIFO register .....	248
Table 5-9 Digital Audio FIFO Control Register .....	249
Table 5-10 Digital Audio FIFO Status Register .....	250

---

Table 5-11 Digital Audio DMA & Interrupt Control Register .....	251
Table 5-12 Digital Audio TX FIFO register .....	251
Table 5-13 Digital Audio Clock Divide Register .....	252
Table 5-14 Digital Audio TX Counter register.....	253
Table 5-15 Digital Audio RX Counter register .....	253
Table 5-16 Digital Audio Channel Configuration register .....	254
Table 5-17 Digital Audio TXn Channel Select register .....	254
Table 5-18 Digital Audio TXn Channel Mapping Register .....	256
Table 5-19 Digital Audio RX Channel Select register .....	256
Table 5-20 Digital Audio RX Channel Mapping Register .....	257
Table 5-21 Digital Audio DBG register .....	258
Table 5-22 DAI Timing Constants at 3.3V in Master Mode .....	258
Table 5-23 DAI Timing Constants in Slave Mode.....	259
Table 5-24 DMIC Clock Source .....	260
Table 5-25 DMIC Register List .....	260
Table 5-26 DMIC Enable Control Register .....	261
Table 5-27 DMIC Sample Rate Register .....	262
Table 5-28 DMIC Control Register .....	263
Table 5-29 DMIC DATA Register .....	263
Table 5-30 DMIC Interrupt Control Register .....	263
Table 5-31 DMIC Interrupt Status Register .....	264
Table 5-32 DMIC FIFO Control Register .....	264
Table 5-33 DMIC FIFO Status Register .....	265
Table 5-34 DMIC Channel Numbers Register .....	265
Table 5-35 DMIC Channel Mapping Register .....	265
Table 5-36 DMIC Counter Register .....	266
Table 5-37 DATA0 and DATA1 Volume Control Register.....	267
Table 5-38 DATA2 and DATA3 Volume Control Register.....	268
Table 5-39 High Pass Filter Enable Control Register .....	269

---

Table 5-40 High Pass Filter Coef Register.....	269
Table 5-41 High Pass Filter Gain Register .....	269
Table 6-1 CSI DATA Port .....	270
Table 6-2 CSI Register List.....	272
Table 6-3 CSI Enable Register.....	272
Table 6-4 CSI Configuration Register .....	273
Table 6-5 CSI Capture Register .....	273
Table 6-6 CSI Horizontal Scale Register .....	274
Table 6-7 CSI CH0 FIFO 0 Output Buffer-A Address Register.....	274
Table 6-8 CSI CH0 FIFO 0 Output Buffer-B Address Register.....	274
Table 6-9 CSI CH0 Output Buffer Control Register .....	275
Table 6-10 CSI CH0 Status Register.....	275
Table 6-11 CSI CH0 Interrupt Enable Register .....	276
Table 6-12 CSI CH0 Interrupt Status Register.....	277
Table 6-13 CSI CH0 Horizontal Size Register .....	277
Table 6-14 CSI CH0 Vertical Size Register.....	278
Table 6-15 CSI CH0 Buffer Length Register .....	278
Table 6-16 CSI FIFO SRAM Data Ready Int Enable Register.....	278
Table 6-17 CSI JPEG Mode Select Register .....	278
Table 7-1 GPIO Register List.....	279
Table 7-2 Pn Configure Register 0 .....	280
Table 7-3 Pn Configure Register 1 .....	280
Table 7-4 Pn Configure Register 2 .....	280
Table 7-5 Pn Configure Register 3 .....	281
Table 7-6 Pn Data Register.....	281
Table 7-7 Pn Multi-Driving Register 0.....	281
Table 7-8 Pn Multi-Driving Register 1.....	281
Table 7-9 Pn Pull Register 0 .....	282
Table 7-10 Pn Pull Register 1 .....	282

---

Table 7-11 Pn External Interrupt Configure 0 Register .....	282
Table 7-12 Pn External Interrupt Configure 1 Register .....	283
Table 7-13 Pn External Interrupt Configure 2 Register .....	283
Table 7-14 Pn External Interrupt Configure 3 Register .....	284
Table 7-15 Pn External Interrupt Control Register .....	284
Table 7-16 Pn External Interrupt Status Register .....	284
Table 7-17 Pn External Interrupt Debounce Register .....	285
Table 7-18 Pin Multiplexing.....	285

## Figures

Figure 2-1 XR871 Functional Block Diagram .....	28
Figure 3-1 XR871GT Clock Control .....	30
Figure 3-2 XR871 Memory Mapping .....	30
Figure 3-3 Message Box Typical Application Chart .....	92
Figure 3-4 Message Box Functional Block Diagram.....	93
Figure 3-5 Spinlock Typical Application Flow Chart .....	103
Figure 3-6 Spinlock Lock Register State Diagram .....	103
Figure 3-7 Crypto Engine Block Diagram .....	106
Figure 4-1 UART Serial Data Timing .....	116
Figure 4-2 UART IrDA Timing .....	116
Figure 4-3 SPI Phase 0 Transfer Format .....	136
Figure 4-4 SPI Phase 1 Transfer Format .....	137
Figure 4-5 TWI Timing Diagram .....	152
Figure 4-6 SDC IDMAC Chain Descriptor.....	187
Figure 4-7 SDC IDMAC Fix Step Descriptor .....	187
Figure 4-8 PWM Block Diagram .....	190
Figure 4-9 PWM Clock Controller Diagram.....	191
Figure 4-10 PWM Output Logic Module Diagram .....	191
Figure 4-11 PWM0 通道 active state 为 high level 的周期、占空比 .....	192
Figure 4-12 PWM0 通道 active state 为 low level 的周期、占空比 .....	192
Figure 4-13 PWM01 对的 PWM0 通道缓存载入、周期模式、脉冲模式和 active state .....	193
Figure 4-14 PWM01 对互补对输出 .....	194
Figure 4-15 插入死区时间的 PWM01 波形对输出 .....	195
Figure 4-16 PWM01 Capture Logic Module Diagram .....	196
Figure 4-17 PWM01 对的 PWM0 通道捕捉时序 .....	197
Figure 4-18 CIR Transmit Modulation .....	222

---

Figure 4-19 CIR Transmit Protocol Example.....	222
Figure 4-20 GPADC Block Diagram .....	230
Figure 5-1 DAUDIO Function Block Diagram.....	239
Figure 5-2 Timing diagram for standard I2S/TDM-I2S mode .....	240
Figure 5-3 Timing diagram for Left-justified/TDM-Left mode .....	241
Figure 5-4 Timing diagram for Right-justified/TDM-Right mode .....	241
Figure 5-5 Timing diagram for PCM/TDM-PCM mode .....	242
Figure 5-6 DAI Timing in Master Mode .....	258
Figure 5-7 DAI Timing in Slave Mode .....	258
Figure 5-8 DMIC Block Diagram .....	259
Figure 6-1 CSI Block Diagram.....	270

# Chapter 1 About This Documentation

## 1.1 Documentation Overview

This documentation provides an overall description of the Xradio XR871 wireless application processor, which will provide instructions to programmers from several sections, including application system, memory, network and interfaces

## 1.2 Acronyms and Abbreviations.

The table below contains acronyms and abbreviations used in this document.

A		
AES	Advanced Encryption Standard	A specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001
AGC	Automatic Gain Control	An adaptive system found in electronic devices that automatically controls the gain of a signal: the average output signal level is fed back to adjust the gain to an appropriate level for a range of input signal levels
AHB	AMBA High-speed Bus	A bus protocol introduced in Advanced Microcontroller Bus Architecture version 2 published by ARM Ltd company
APB	Advanced Peripheral Bus	APB is designed for low bandwidth control accesses, which has an address and data phase similar to AHB, but a much reduced, low complexity signal list (for example no bursts)
C		
CIR	Consumer IR	The CIR (Consumer IR) interface is used for remote control through infra-red light
CRC	Cyclic Redundancy Check	A type of hash function used to produce a checksum in order to detect errors in data storage or transmission
CSI	CMOS Sensor Interface	The hardware block that interfaces with different image sensor interfaces and provides a standard output that can be used for subsequent image processing
D		
DES	Data Encryption Standard	A previously predominant algorithm for the encryption of electronic data
DLL	Delay-Locked Loop	A digital circuit similar to a phase-locked loop (PLL), with the main

		difference being the absence of an internal voltage-controlled oscillator, replaced by a delay line
DVFS	Dynamic Voltage and Frequency Scaling	Dynamic voltage scaling is a power management technique where the voltage used in a component is increased or decreased, depending on circumstances. Dynamic frequency scaling is a technique whereby the frequency of a microprocessor can be automatically adjusted on the fly so that the power consumption or heat generated by the chip can be reduced. These two are often used together to save power in mobile devices.
<b>E</b>		
eMMC	Embedded Multi-Media Card	An architecture consisting of an embedded storage solution with MMC interface, flash memory and controller, all in a small BGA package.
<b>I</b>		
I2S	IIS	An electrical serial bus interface standard used for connecting digital audio devices together
<b>L</b>		
LSB	Least Significant Bit	The bit position in a binary integer giving the units value, that is, determining whether the number is even or odd. It is sometimes referred to as the right-most bit, due to the convention in positional notation of writing less significant digits further to the right.
<b>M</b>		
MAC	Media Access Control	A sublayer of the data link layer, which provides addressing and channel access control mechanisms that make it possible for several terminals or network nodes to communicate within a multiple access network that incorporates a shared medium, e.g. Ethernet.
MSB	Most Significant Bit	The bit position in a binary number having the greatest value, which is sometimes referred to as the left-most bit due to the convention in positional notation of writing more significant digits further to the left
<b>P</b>		
PCM	Pulse Code Modulation	A method used to digitally represent sampled analog signals
<b>S</b>		
SPI	Synchronous Peripheral Interface	A synchronous serial data link standard that operates in full duplex mode. Devices communicate in master/slave mode where the master device initiates the data frame.

# Chapter 2 Overview

## 2.1 General Description

XR871 is a highly integrated low-power WLAN Microcontroller System-on-Chip (SOC) solution designed for Internet of Things (IoT), Wearable equipment, Machine-to-Machine (M2M), Home automation, Cloud Connectivity and Smart Energy applications.

The XR871 application subsystem is powered by an ARM Cortex-M4F CPU that operates up to 192MHz. It supports an integrated 448KB SRAM and 64KB ROM, and a QSPI interface to external Flash. An integrated Flash Cache enables eXecute In Place (XIP) support for firmware from flash. It also includes many peripherals, including UART, TWI, SPI, I2S, DMIC, PWM, IrDA (T/R), CSI, SDIO and auxiliary ADC.

The WLAN subsystem contains the 802.11b/g/n radio, baseband and MAC that designed to meet both the low power and high throughput network application.

The SoC is designed for low-power operation and there is a separate power management unit for each subsystem. Multiple power domains and clocks can be individually shut down and the application and WLAN subsystems can be placed into low-power states, independently, to support a variety of application cases.

## 2.2 Features

- Package
  - 6 x 6mm 52-pin QFN package
- Power Management and Clock Source
  - Integrate high efficiency power management unit with single 2.7-5.5V power supply input.
  - Integrated DC-DC and LDOs for internal power supply
  - Separate power switches for CPU, RAM and peripherals
  - 24/26/40/52MHz source crystal clock support
  - **32KHz OSC and RC clock support (XR871GT)**
  - **32KHz RC clock support (XR871ET)**
- Application Microcontroller Subsystem
  - ARM Cortex-M4F, up to 192MHz
  - Embedded 448KB SRAM and 64KB ROM
  - Supports external SPI flash with QSPI mode and eXecute In Place (XIP) on flash
  - Flash cache for XIP mode
  - Supports Secure Boot

- Hardware Crypto Engine for Advanced Security, Including AES, DES/3DES, SHA2/MD5, CRC
  - 8-channels General Direct Memory Access(DMA) channels
  - 2 Universal Asynchronous Receivers and Transmitters (UART)
  - 2 Serial Peripheral Interface (SPI, boot & Application) with multi Chip-Select
  - 2 General Timers, 2 alarm Timers, 1 RTC and 1 watch dog
  - 8 PWM and Event Capture Controllers
  - 8 channels 12-bit accuracy ADC
  - 1 Camera Serial Interface (CSI)
  - 1 Digital Audio Controller supports PCM and IIS protocol
  - 1 Digital Microphone Controller
  - 1 SD/MMC/SDIO Controller for external storage
  - 2 Two Wire Interface Controllers for Camera module and some other sensors control
  - 1 IR receiver and 1 IR transmitter
- 
- WLAN Subsystem
    - 802.11b/g/n Radio, Baseband, Medium Access Control(MAC)
    - Embedded TCP/IP Stack
    - Station, AP Modes
    - SmartConfig Technology for Autonomous and Fast WIFI Connections
    - Security support for WEP, WPA/WPA2 personal, WPS2.0
    - Industry-Standard BSD Socket Application Programming Interfaces (APIs)
  
  - Miscellaneous
    - Integrates 2Kbit eFuse to store device specific information and RF calibration data

## 2.3 Application

- Home Automation
- Home Appliances
- Access Control
- Security Systems
- Smart Energy
- Internet Gateway
- Cloud Connectivity
- Industrial Control
- Wearable Equipment
- Wireless Audio
- IP Network Sensor Nodes

## 2.4 Block Diagram

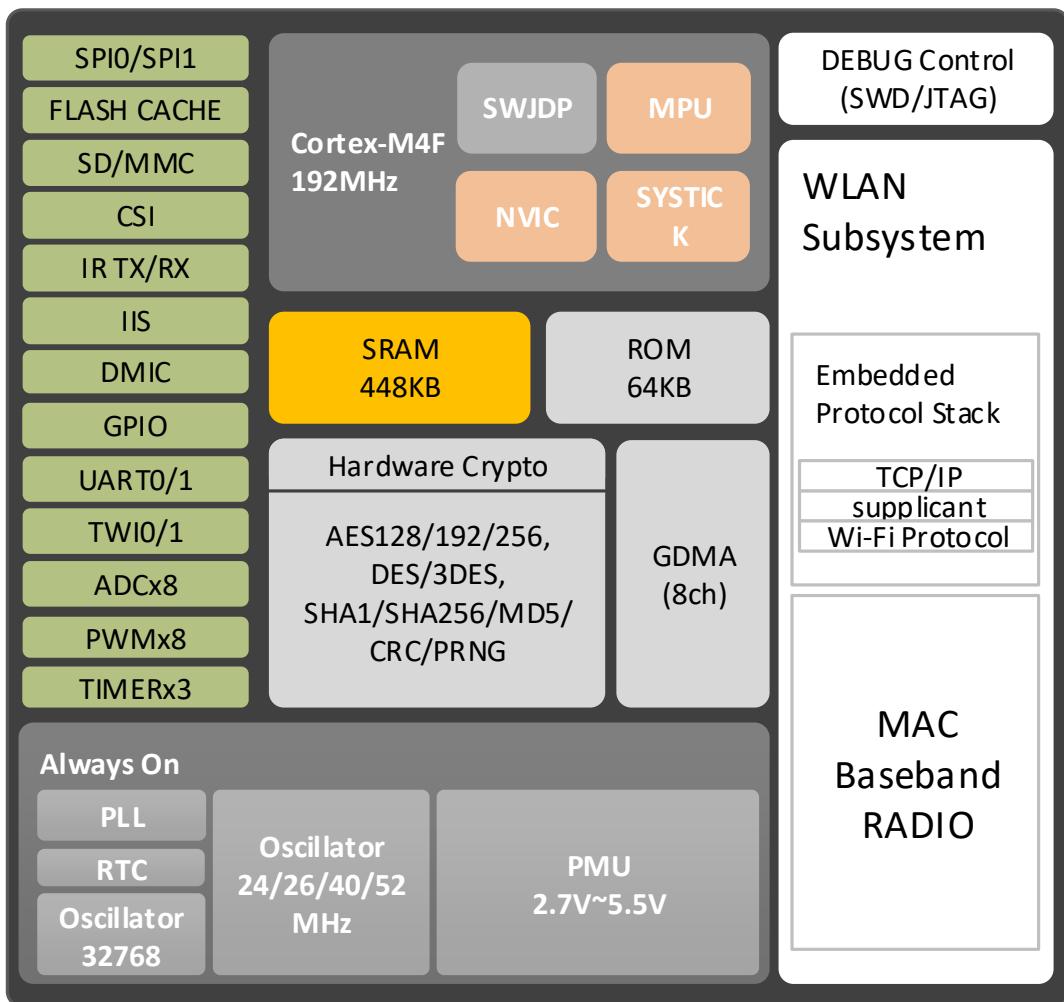


Figure 2-1 XR871 Functional Block Diagram

# Chapter 3 System

## 3.1 Power Management

A single 2.7 – 5.5V power supply is required for the XR871. It could be from an AC-DC converter to convert the AC voltage supply to 5V or a DC-DC converter to convert higher voltage supply to 3.3V. It could be from a battery directly too.

The Power Management Unit (PMU) contains a DC-DC, several Low Drop-out Regulators (LDOs), a highly efficient buck converter and a reference band-gap circuit. The circuits are optimized for low quiescent current, low drop-out voltage, efficient line/load regulation, high ripple rejection, and low output noise.

The PMU integrates several LDOs for different circuits: DLDO stands for digital core LDO and the ALDO stands for Analog and RF system LDO. PLDO stands for clock generate system LDO and RLDO stands for the RTC and SLEEP system LDO. In Deep-Sleep mode, the DLDO, PLDO, RLDO can be shut down and only the RLDO is working.

There are three power domains in the system: RTC domain, SRAM domain, Digital Core domain and WIFI domain.

## 3.2 Clock

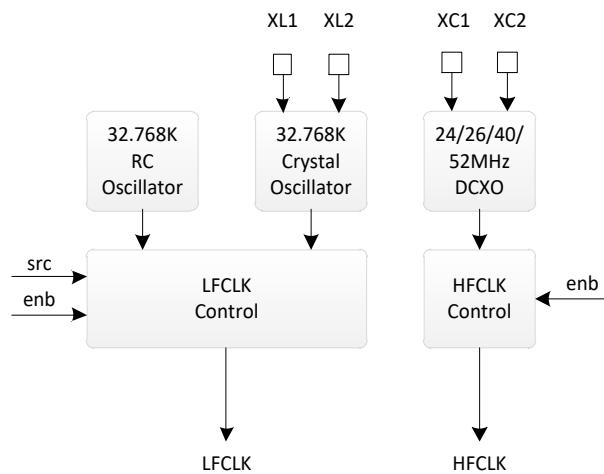
The clock management system can source the system clocks from a range of internal or external high and low frequency oscillators and distribute them to modules based up a module's individual requirements. The system depends on, and generates two different clocks: a high frequency clock *HFCLK* and a low frequency clock *LFCLK*.

The system supports two LFCLK clock sources, the 32.768 KHz crystal oscillator (**Only for XR871GT**) and the 32.768 KHz RC oscillator. The 32.768 KHz crystal oscillator requires an external AT-cut quartz crystal to be connected to the XL1 and XL2 pins. The LFCLK clock and all of the available LFCLK sources are switched off by default when the system is powered up. The LFCLK clock can be started by selecting the preferred clock source in PRCM register. The LFCLK is used for each subsystem to achieve lower current consumption for different running mode. In addition, the LFCLK is also used in RTC circuit to achieve accuracy timing.

There is only one clock source for HFCLK, the 24MHz, 26Mhz, 40MHz or 52MHz crystal oscillator. The HFCLK is enabled automatically when the system is powered up and can be switched off when all subsystems won't use it anymore in some low power modes.

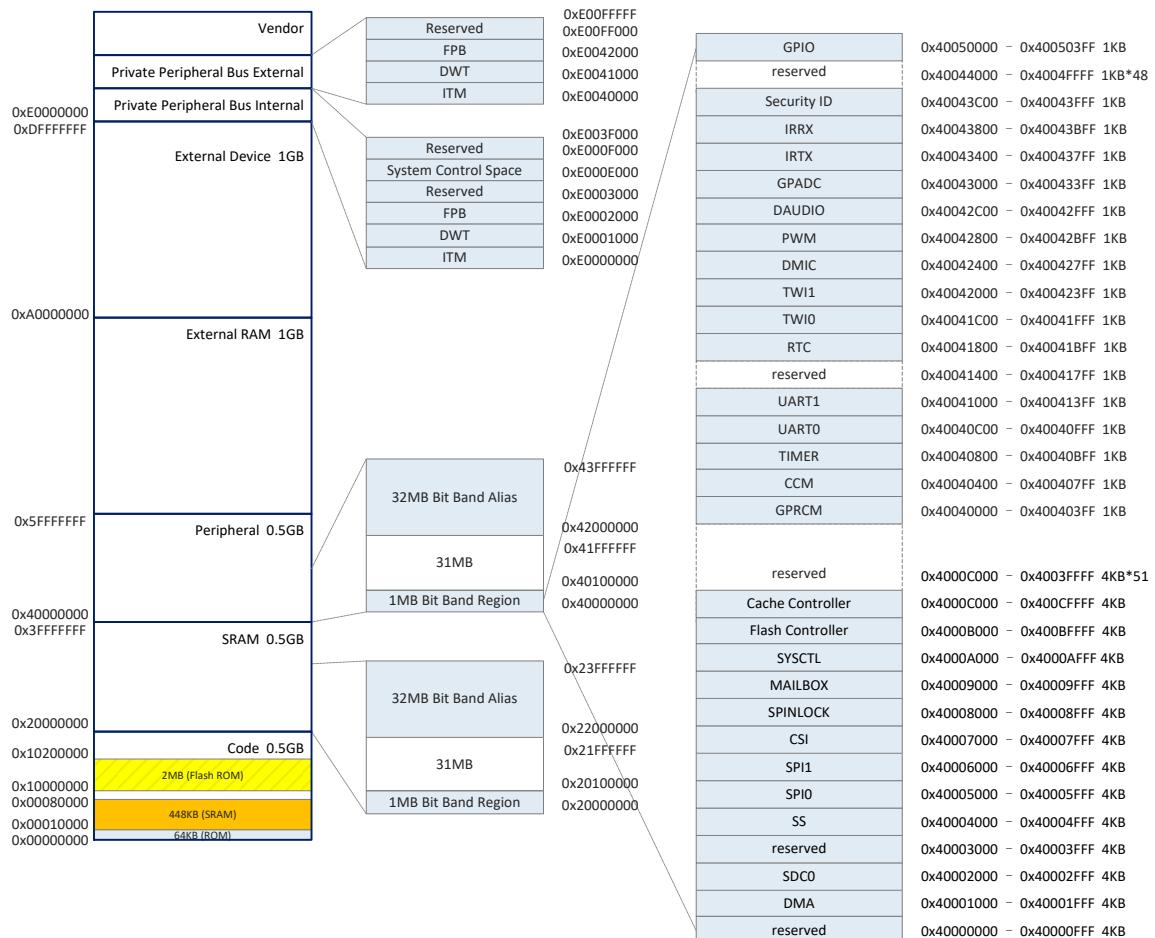
The HFCLK is used to generate the clock source for Digital PLL, which is used to generate the clock sources for Cortex-M4 core, WLAN and peripherals. There is also an Audio PLL used to generate the clock source for I2S (for external audio CODEC).

The following figure shows the clock control block diagram.



**Figure 3-1 XR871GT Clock Control**

### 3.3 Memory Mapping



**Figure 3-2 XR871 Memory Mapping**

## 3.4 CPU System

XR871 features an ARM Cortex-M4 processor, which is the most energy efficient ARM processor available. It supports the clock rates from 32KHz up to 192MHz. The processor provides a low-cost platform that meets the needs of minimal memory implementation, reduced pin count, and low power consumption.

The ARM Cortex-M4 core has low-latency interrupt processing with the following features:

- Thump-2 instruction set for optimal performance and code size
- Handler and thread modes
- Memory Protection Unit (MPU) for memory protection features
- Floating Point Unit (FPU) to support DSP related function
- Nested Vectored Interrupt Controller (NVIC) to achieve low latency interrupt processing
- Three Advanced High-Performance bus AHB-Lite interfaces: ICode, DCode and system bus
- Bit-band support for memory and select peripheral that include atomic bit-band write and read operations
- Wake-up Interrupt Controller (WIC) providing ultra-low power sleep mode support

*Note: Detail information of Cortex-M4, NVIC and MPU please refer to 'ARMv7-M Architecture Reference Manual' and 'ARM Cortex-M4 Processor Technology Reference Manual'.*

### 3.4.1 Register List

Module Name	Base Address	
Nested Vectored Interrupt Controller	0xE000E000	
Register Name	Offset Address	Description
	0x004	
SYSTICK_CTRL	0x010	SysTick Control and Status Register
SYSTICK_RELOAD	0x014	SysTick Reload Value Register
SYSTICK_CURVAL	0x018	SysTick Current Value Register
NVIC_ISERO – NVIC_ISER7	0x100 – 0x11C	Interrupt Set-Enable Registers
NVIC_ICERO – NVIC_ICER7	0x180 – 0x19C	Interrupt Clear-Enable Registers
NVIC_ISPRO – NVIC_ISPR7	0x200 – 0x21C	Interrupt Set-Pending Registers
NVIC_ICPRO – NVIC_ICPR7	0x280 – 0x29C	Interrupt Clear-Pending Registers
NVIC_IABRO – NVIC_IABR7	0x300 – 0x31C	Interrupt Active Bit Registers
NVIC_IPRO – NVIC_IPR59	0x400 – 0x4CF	Interrupt Priority Registers
SCB	0xD08 – 0xD3f	System Control Block

Table 3-1 NVIC Register List

### 3.4.2 Interrupt Source Description

IRQ Number	Vector	Interrupt Source	Priority	Description
------------	--------	------------------	----------	-------------

0	0x00			Stack top is loaded from first entry of vector table on reset.
1	0x04		-3	Reset
2	0x08		-2	NMI
3	0x0C		-1	Hardware Fault
4	0x10		configurable	Memory Management
5	0x14		configurable	Bus Fault
6	0x18		configurable	Usage Fault
7	0x1C			-
8	0x20			-
9	0x24			-
10	0x28			-
11	0x2C		configurable	SVCall
12	0x30		configurable	Debug Monitor
13	0x34			-
14	0x38		configurable	PendSV
15	0x3C		configurable	System Tick
16	0x40	DMA	configurable	Direct Memory Access Controller
17	0x44	GPIOA	configurable	GPIO Controller Port A
18	0x48	SDC	configurable	SDMMC/SDIO Controller
19	0x4C	MSGBOX	configurable	Message Box(IRQ0)
20	0x50	UART0	configurable	UART Controller 0
21	0x54	UART1	configurable	UART Controller 1
22	0x58	SPI0	configurable	SPI Controller 0
23	0x5C	SPI1	configurable	SPI Controller 1
24	0x60	TWI0	configurable	TWI Controller 0
25	0x64	TWI1	configurable	TWI Controller 1
26	0x68	WDT	configurable	Watchdog
27	0x6C	TIMERO	configurable	Timer 0
28	0x70	TIMER1	configurable	Timer 1
29	0x74	ALARM0	configurable	Alarm 0
30	0x78	ALARM1	configurable	Alarm 1
31	0x7C	CSI	configurable	CSI Controller
32	0x80	DAUDIO	configurable	Digital Audio Controller
33	0x84	PWM/ECT	configurable	PWM/ECT Controller
34	0x88	SS	configurable	Secure System Controller
35	0x8C	GPADC	configurable	GPADC Controller
36	0x90	GPIOB	configurable	GPIO Controller Port B
37	0x94	DMIC	configurable	Digital Microphone
38	0x98	IRRX	configurable	IR Receiver
39	0x9C	IRTX	configurable	IR Transmitter
40	0xA0	N-MBOX	configurable	N-Message Box(IRQ1)

41	0XA4	A-WUP	configurable	APP Wake Up(Timer & IO)
42	0xA8	FLASHC	configurable	FLASH Controller
43	0XAC	NUART	configurable	N-UART Controller
...				

Table 3-2 Interrupt Source

## 3.5 Global Power, Reset and Clock Manager (GPRCM)

### 3.5.1 Overview

The GPRCM module manages the power, reset and input clocks for this system. The GPRCM is placed in always on power domain.

The GPRCM module includes the following features:

- Manage the power of this system
- Manage the reset of each system
- Manage the OSC clock
- Manage the wakeup source

### 3.5.2 Register List

Module Name	Base Address	
Power, Reset and Clock Manager	0x40040000/0xa0040000	
Register Name	Offset Address	Description
SYS_DCDC_CTRL	0x0000	System DCDC Control
SYS_LDO_SW_CTRL	0x0004	System LDO & Switch Control
SYS_LFCLK_CTRL	0x0008	System Low Frequency Clock Control
SYS_HOSC_TYPE	0x000C	System HOSC Type
SYS_RCOSC_CAL_CTRL	0x0010	System RC-OSC Calibration Control
SYS_PLL_CTRL	0x0020	System PLL Control
SYS_CLK1_CTRL	0x0024	System Clock 1 Control
SYS_CLK2_CTRL	0x0028	System Clock 2 Control
AUDIO_PLL_CTRL	0x0030	Audio PLL Control
DEV_CLK_CTRL	0x0034	Device Clock Control
AUDIO_PLL_BIAS	0x0044	Audio PLL BIAS
AUDIO_PATTERN	0x0054	Audio Pattern Control
DCXO_CTRL	0x0058	
SYS1_CTRL	0x080	System 1 Control
SYS1_STATUS	0x084	System 1 Status
SYS2_CTRL	0x088	System 2 Control

SYS2_STATUS	0x08C	System 2 Status
SRAM_VOL_CTRL	0x0B8	SRAM Voltage Control
AR400A_BOOT_FLAG	0x0100	AR400A boot flag
AR400A_BOOT_ADDR	0x0104	AR400A boot address
AR400A_BOOT_ARG	0x0108	AR400A boot argument
AR400A_PRIV_REG	0x0118	AR400A Private Register
AR400A_WAKE_TIMER_CNT	0x0120	AR400A wakeup timer counter
AR400A_WAKE_TIMER_CTRL	0x0124	AR400A wakeup timer control
AR400A_IO_WAKE_EN	0x0130	AR400A IO Wakeup Enable
AR400A_IO_WAKE_MOD	0x0134	AR400A IO Wakeup Mode
AR400A_IO_WAKE_ST	0x0138	AR400A IO Wakeup Status
AR400A_IO_HOLD_CTRL	0x013c	AR400A IO Hold Control
AR400A_IO_WUP_GEN	0x0140	AR400A IO Wakeup Global Enable

Table 3-3 GPRCM Register List

### 3.5.3 Register Description

#### 3.5.3.1 System DCDC Control Register

<b>Address : 0x0000</b>			<b>Name:</b> SYS_DCDC_CTRL0 <b>Default:</b> 0x0008_0001
Field	Type	Default	Description
31:20	/	/	/
19:16	R/W	0x08	<p>DCDC_VOLTAGE_SEL (1.4V – 1.8V)</p> <p>N: 0~15</p> <p>other: reserved</p> <p><i>Voltage = 1800mV - N*26.7mV, default value = 1.586V</i></p> <p><b>Note:</b> This bit will only be reset to the default value in power-on-reset operation. When the system is waked up from the deep sleep mode by some external events, this bit will keep the value which is set by the last write operation.</p>
15:1	/	/	/
0	R	1	<p>SYS_DCDC_EN</p> <p>1: Write 1 to enable the system DCDC.</p> <p>0: write 0 to disable the system DCDC.</p>

			<b>Note:</b> This bit will only be reset to the default value in power-on-reset operation. When the system is waked up from the deep sleep mode by some external events, this bit should be set to 1 automatically to make sure that the CPU can start up correctly.
--	--	--	--

Table 3-4 System DCDC Control Register

### 3.5.3.2 System LDO & Switch Control Register

<b>Address :</b> 0x0004			<b>Name:</b> SYS_LDO_SW_CTRL <b>Default:</b> 0x0000_0707
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:16	/	/	/
15	R	0	SR_SW3_EN  1: enable 0: disable
14	R	0	SW5_EN  1: enable 0: disable
13	R	0	SW4_EN  1: enable 0: disable
12	R/W	0	SW3_EN Network System Power Switch Enable  1: enable 0: disable
11	R/W	0	SR_SW2_EN Network System SRAM Power Switch Enable  1: enable 0: disable
10	R	1	SR_SW1_EN Application System SRAM Power Switch Enable  1: enable 0: disable <i>Note: this bit will only be set by POWERCTRL module.</i>

9	R	1	<b>SW2_EN</b> Application System Power Switch Enable  1: enable 0: disable <i>Note: this bit will only be set by POWERCTRL module.</i>
8	R	1	<b>SW1_EN</b> AR400A Power Switch Enable  1: enable 0: disable <i>Note: this bit will only be set by POWERCTRL module.</i>
7	/	/	/
6:4	R/W	0	<b>LDO1_VOL_SEL</b> 0 ~ 4: N others: reserved  <i>Note: <math>V_{ldo1} = 1100mV - N * 50mV</math>, the output range is from 900mV to 1100mV                      the default is 1100mV</i>
3	/	/	/
2	R	1	<b>SRAM_LDO_EN</b>  1: Write 1 to enable the SRAM LDO. 0: write 0 to disable the SRAM LDO. <i>Note: this bit will only be set by POWERCTRL module.</i>
1	R/W	1	<b>PLL_LDO_EN</b>  1: Write 1 to enable the PLL_LDO_EN. 0: write 0 to disable the PLL_LDO_EN.
0	R	1	<b>LDO1_EN</b>  1: Write 1 to enable the LDO1. 0: write 0 to disable the LDO1. <i>Note: this bit will only be set by POWERCTRL module.</i>

Table 3-5 System Power Control 1 Register

### 3.5.3.3 System LFCLK Control Register

<b>Address :</b> 0x0008		<b>Name:</b> SYS_LFCLK_CTRL <b>Default:</b> 0xc000_000f	
Field	Type	Default	Description

31	R/W	1	OSC32K_EN  1: enable the 32768 Oscillator 0: disable the 32768 Oscillator
30	R/W	1	RC32K_EN  1: enable the 32768 RC Oscillator 0: disable the 32768 RC Oscillator
29:25	/	/	/
24	R/W	0	LFCLK_SRC_SEL  Low frequency clock source select 0: source the LFCLK from the internal 32768 RC oscillator. 1: source the LFCLK from the external 32768 crystal oscillator
23:0	/	/	/

**Table 3-6 System LFCLK Control Register**

### 3.5.3.4 System HOSC Type Register

<b>Address :</b> 0x000C			<b>Name:</b> SYS_HOSC_TYPE <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:2	/	/	/
1:0	R/W	0	HOSC_TYPE  0: System uses 26MHz HOSC 1: System uses 40MHz HOSC 2: System uses 24MHz HOSC 3: System uses 52MHz HOSC  <i>Note: This bit is set by application software and must be consistent with the real situation. A wrong configuration will cause the RCOSC calibration fail.</i>

**Table 3-7 System HOSC Type Register**

### 3.5.3.5 System RCOSC Calibration Control Register

<b>Address :</b> 0x0010			<b>Name:</b> SYS_RCOSC_CAL_CTR_L <b>Default:</b> 0x0000_0001
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:28	/	/	/
27:8	R	0xXXX	RC-OSC Frequency V=FREQ/10

7:1	/	/	/
0	R/W	1	RC-OSC Calibration Enable

Table 3-8 System RCOSC Calibration Control Register

### 3.5.3.6 System PLL Control Register

<b>Address :</b> 0x0020			<b>Name:</b> SYS_PLL_CTRL <b>Default:</b> 0x0ec4_f121@26MHz OSC, 0x800010c1@40MHz OSC
Field	Type	Default	Description
31	R/W	0	<b>PLL_ENABLE</b>  Enable or disable the system PLL. 0: Disable the system PLL 1: Enable the system PLL  <i>Note:</i> The PLL frequency $F_{output} = (F_{hosc} * 2 * N.f) / M$ , the output clock must be in range of 480~960MHz. The default value is set for 960MHz with 26MHz OSC. When using 40MHz OSC, the value can be set to 0x000010c1 to get the 960MHz clock.
30:29	RW	0	DPLL_DITHER_DISABLE
28:13	RW	0x7627	<b>DPLL_FRAC</b> <i>Note :</i> $V_{FRAC} = f * 2^{16}$
12	RW	1	DPLL_FRAC_CTRL
11:4	RW	18	DPLL_NDIV
3:0	RW	1	<b>FACTOR_M</b>  PLL factor M (1 ~ 8). M = Factor <i>Note:</i> when the factor is set to 0, M will be set to 1

Table 3-9 System PLL Control Register

### 3.5.3.7 System Clock 1 Control Register

<b>Address :</b> 0x0024			<b>Name:</b> SYS_CLK1_CTRL <b>Default:</b> 0x0000_000b
Field	Type	Default	Description
31	RW	0	<b>SYS_CLK1_EN</b>  1: enable the SYSCLK1 0: disable the SYSCLK1  <i>Note:</i> $F_{sysclk1} = F_{pll} / N$ , the default value is 80MHz.

30:18	/	/	/
17:16	R/W	0x0	<p>CPU_CLK_SRC_SEL</p> <p>CPU clock source select</p> <p>00: HFCLK(default)</p> <p>01: LFCLK</p> <p>1x: SYSCLK1</p>
15:4	/	/	/
3:0	RW	0xB	<p>FACTOR_N</p> <p>PLL factor N (1 ~ 16).</p> <p>Factor = 0~15</p> <p>N = Factor + 1</p> <p><b>Note:</b> <math>F_{sysclk1} = F_{pll} / N</math>, the default value is 80MHz.</p>

Table 3-10 System Clock 1 Control Register

### 3.5.3.8 System Clock 2 Control Register

<b>Address :</b> 0x0028			<b>Name:</b> SYS_CLK2_CTRL <b>Default:</b> 0x0000_000b
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	R	0	<p>SYS_CLK2_EN</p> <p>1: enable the SYSCLK2</p> <p>0: disable the SYSCLK2</p> <p><b>Note:</b> <math>F_{sysclk2} = F_{pll} / N</math>, the default value is 80MHz.</p>
30:18	/	/	/
17:16	R	0x0	<p>CPU_CLK_SRC_SEL</p> <p>CPU clock source select</p> <p>00: HFCLK(default)</p> <p>01: LFCLK</p> <p>1x: SYSCLK2</p>
15:4	/	/	/
3:0	R	0xB	<p>FACTOR_N</p> <p>PLL factor N (1 ~ 16).</p> <p>Factor = 0~15</p> <p>N = Factor + 1</p> <p><b>Note:</b> <math>F_{sysclk2} = F_{pll} / N</math>, the default value is 80MHz.</p>

Table 3-11 System Clock 2 Control Register

### 3.5.3.9 Audio PLL Control Register

<b>Address :</b> 0x0030			<b>Name:</b> AUDIO_PLL_CTRL <b>Default:</b> 0x0103_0601@26MHz, 0x0103_0100@40MHz
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	RW	0	<p>PLL_EN</p> <p>1: enable the PLL 0: disable the PLL</p> <p><b>Note:</b></p> <p><i>This PLL is used for Audio application.</i></p> $\text{AUDIO\_PLL} = (\text{HFCLK} * N) / (M * P)$ $\text{AUDIO\_PLL\_8X} = (\text{HFCLK} * N * 2) / (M * P)$ $3 \leq N/M \leq 21$ <p><i>HFCLK * N / P must be in the range of 74MHz ~ 504MHz</i></p> <p><i>AUDIO_PLL default is 24.576MHz</i></p>
30:29	/	/	/
28	R	x	<p>LOCK</p> <p>0: Unlocked 1: Locked (means the pll is stable)</p>
27:25	/	/	/
24	RW	1	<p>PLL_SDM_EN</p> <p>0: Disable 1: Enable</p> <p>In this case, the PLL_FACTOR_N only low 4 bits are valid (1~16)</p>
23:20	/	/	/
19:16	RW	0x3	<p>PLL_POSTDIV_P</p> <p>Post-div factor (P = Factor + 1) the range if from 1 to 16</p>
15	/	/	/
14:8	RW	0x06	<p>PLL_FACTOR_N</p> <p>N = Factor + 1 the range if from 1 to 128</p>
7:5	/	/	/
4:0	RW	0x01	<p>PLL_PRETDIV_M</p> <p>Pre-div factor (M = Factor + 1) the range if from 1 to 32</p>

Table 3-12 Audio Clock Control Register

### 3.5.3.10 Device Clock Control Register

<b>Address :</b> 0x0034			<b>Name:</b> DEV_CLK_CTRL <b>Default:</b> 0x0000_0004
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:4	/	/	/
3:0	RW	4	<p><b>FACTOR_N</b></p> <p>PLL factor N (1 ~ 16).</p> <p>Factor = 0~15</p> <p>N = Factor + 1</p> <p><i>Note: this clock is used as a source clock for some devices like SDC, SPI, CSI, etc. The default value is 192MHz</i></p>

Table 3-13 Device Clock Control Register

### 3.5.3.11 Audio PLL Bias Register

<b>Address :</b> 0x044			<b>Name:</b> AUDIO_BIAS <b>Default:</b> 0x1010_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:29	/	/	/
28:24	RW	0x10	PLL_VCO_BIAS_CTRL PLL VCO Bias Control[4:0]
23:21	/	/	/
20:16	RW	0x10	PLL_CUR_BIAS_CTRL PLL Current Bias Control[4:0]
15:0	/	/	/

Table 3-14 Audio PLL Bias Register

### 3.5.3.12 Audio PLL Pattern Control Register

<b>Address :</b> 0x0054			<b>Name:</b> AUDIO_PLL_PATTERN_CTRL <b>Default:</b> 0xC001_1FAA@26MHz, 0xC000_EA4A@40MHz
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	RW	1	<p><b>DIG_DELT_PAT_EN</b></p> <p>Sigma-Delta Pattern Enable</p>
30:29	RW	0x2	<p><b>SPR_FREQ_MODE</b></p> <p>Spread Frequency Mode</p> <p>00: DC = 0</p> <p>01: DC = 1</p>

			1X: Triangular
28:20	RW	0	WAVE_STEP  Wave Step
19	/	/	/
18:17	RW	0	FREQ  00: 31.5KHz 01: 32KHz 10: 32.5 KHz 11: 33 KHz
16:0	RW	0x11faa	WAVE_BOT  Wave Bottom

**Table 3-15 Audio PLL Pattern Control Register**

### 3.5.3.13 DCXO Control Register

<b>Address : 0x058</b>			<b>Name:</b> DCXO_CTRL0 <b>Default:</b> 0x9409_0b00
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	RW	1	DCXO_EN  DCXO Enable  0: disable 1: enable
30	/	/	/
29:28	RW	1	CLK_MODE_SEL  Clock Mode Select (When AUTO_DET_EN is disable)  00: Internal Clock 01: Crystal 10: Analog 11: Digital
27:20	RW	0x40	FRE_TRI  Frequency Trimming
19	RW	1	ALC_EN  ALC Enable  0: disable 1: enable
18:17	/	/	/
16:12	RW	0x10	ICTRL  BIAS Current Control
11	RW	1	RF_DIG_REF_CLK_EN  0: disable 1: enable

10	/	/	/
9	RW	1	BUFFER_DPLL_EN  0: disable 1: enable
8:6	/	/	/
5:4	R	xx	DCXO_DET 00: Internal clock 01: Crystal 10: Analog 11: Digital
3:1	/	/	/
0	RW	0	AUTO_DET_EN Auto Detect Enable 0: disable 1: enable

Table 3-16 DCXO Control 0 Register

### 3.5.3.14 System 1 Control Register

<b>Address : 0x080</b>			<b>Name:</b> SYS1_CTRL <b>Default:</b> 0x0000_0003
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:2	/	/	/
1	R	1	AR400A_RST  CPU AR400A Reset 1: release the reset signal of AR400A 0: hold the global reset signal of AR400A
0	R	1	SYS1_RST  system 1 Reset 1: release the global reset signal of the system 1 0: hold the global reset signal of the system 1

Table 3-17 System 1 Control Register

### 3.5.3.15 System 1 Status Register

<b>Address : 0x0084</b>			<b>Name:</b> SYS1_STA <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:4	/	/	/
3			

2	R	0	CPU_SLEEP  1: the CPU of system 1 is in SLEEP mode 0: the CPU of system 1 is not in SLEEP mode
1	R	0	CPU_SLEEPDEEP  1: the CPU of system 1 is in SLEEPDEEP mode 0: the CPU of system 1 is not in SLEEPDEEP mode
0	R	0	SUBSYS1_ALIVE  1: system 1 is alive 0: system 1 is down

Table 3-18 System 1 Status Register

### 3.5.3.16 System 2 Control Register

<b>Address :</b> 0x0088			<b>Name:</b> SYS2_CTRL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:3	/	/	/
2	RW	0	<b>SYS2_ISOLATION_CTRL</b>  Isolation control for AR400N and system2 1: disable the isolation for system 2 0: enable the isolation for system2 <i>Note: set this bit to 0 to cut off the connection with the whole circuits of system2. So you must set this bit before disabling the SW3 and SR_SW2</i>
1	RW	0	<b>AR400N_RST</b>  CPU AR400N Reset 1: release the reset signal of AR400N 0: hold the global reset signal of AR400N
0	RW	0	<b>SYS2_RST</b>  System 2 Reset 1: release the global reset of the system 2 0: hold the global reset of the system 2

Table 3-19 System 2 Control Register

### 3.5.3.17 System 2 Status Register

<b>Address :</b> 0x008c	<b>Name:</b> SYS2_STA <b>Default:</b> 0x0000_0000
-------------------------	--

Field	Type	Default	Description
31:4	/	/	/
3			
2	R	0	CPU_SLEEP  1: the CPU of system 2 is in SLEEP mode 0: the CPU of system 2 is not in SLEEP mode
1	R	0	CPU_SLEEPDEEP  1: the CPU of system 2 is in SLEEPDEEP mode 0: the CPU of system 2 is not in SLEEPDEEP mode
0	R	0	SYS2_ALIVE  1: system 2 is alive 0: system 2 is down

**Table 3-20 System 2 Status Register**

### 3.5.3.18 SRAM Voltage Control Register

Address : 0x00B8			Name: SRAM_VOL_CTRL Default: 0x0000_0000
Field	Type	Default	Description
31:3	/	/	0
18:16	RW	0	SRAM Work Voltage Control  0: retention @1.10V 1: retention @1.00V 2: retention @0.90V 3: retention @0.80V 4: retention @0.75V 5: retention @0.70V 6: retention @0.65V 7: retention @0.60V
15:3	/	/	/
2:0	RW	0	SRAM Retention Voltage Control  0: retention @1.10V 1: retention @1.00V 2: retention @0.90V 3: retention @0.80V 4: retention @0.75V 5: retention @0.70V 6: retention @0.65V 7: retention @0.60V

**Table 3-21 DCXO Stable Reference Time Register**

### 3.5.3.19 AR400A Boot Flag Register

Address : 0x0100			Name: BOOT_FLAG  Default: 0x0000_0000
Field	Type	Default	Description
31:16	RW	0	<p>BOOT_FLAG_WRITE_LOCK</p> <p>0x429B: AR400A_BOOT_FLAG filed is writable other: AR400A_BOOT_FLAG filed is read only</p> <p><b>Note:</b> The field of AR400A_BOOT_FLAG only can be changed when this bit is set to <b>0x429B</b>. In one write operation, the GRPCM will check this field whether equals to 0x429B. If not, the value of AR400A_BOOT_FLAG will be ignored.</p> <p>This field is always read with value 0x0000.</p> <p>Examples:</p> <ol style="list-style-type: none"> <li>1. After write 0x429B0001, then read value = 0x00000001</li> <li>2. After write 0x11110000, then read value = 0x00000001</li> <li>3. After write 0x429B0000, then read value = 0x00000000</li> </ol>
15:4	/	/	/
3:0	RW	0	<p>AR400A_BOOT_FLAG</p> <p>0: boot from cold reset(default) 1: boot from deep sleep 2: boot for system update others: reserved</p>

Table 3-22 AR400A Boot Flag Register

### 3.5.3.20 AR400A Boot Address Register

Address : 0x0104			Name: BOOT_ADDR  Default: 0x0000_0000
Field	Type	Default	Description
31:0	RW	0	AR400A_BOOT_ADDR

Table 3-23 AR400A Boot Address Register

### 3.5.3.21 AR400A Boot Argument Register

Address : 0x0108	Name: BOOT_ARG
------------------	----------------

			<b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	RW	0	AR400A_BOOT_ARG

Table 3-24 AR400A Boot Argument Register

### 3.5.3.22 AR400A Private Register

<b>Address :</b> 0x0118			<b>Name:</b> AR400A_PRIV_REG  <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	RW	0	AR400A_PRIV_REG

Table 3-25 AR400A Private Register

### 3.5.3.23 AR400A Wakeup Timer Counter Register

<b>Address :</b> 0x0120			<b>Name:</b> AR400A_WAKE_TIMER_CNT  <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31	RW	0	AR400A_WAKE_TIMER_EN
30:0	R	0	AR400A_WAKE_TIMER_CNT  <i>Note: This is a free running counter and when the value of this counter equals to the value of AR400A_WAKE_TIMER_VAL, a wakeup signal will be issued to the Wakeup Interrupt Controller.</i> <i>If AR400A_WAKE_TIMER_EN is set to '0', this counter will be reset to 0 and stop counting.</i>

Table 3-26 AR400A Wakeup Timer Counter Register

### 3.5.3.24 AR400A Wakeup Timer Value Register

<b>Address :</b> 0x0124			<b>Name:</b> AR400A_WAKE_TIMER_VAL  <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31	RW	0	AR400A_WAKEUP_TIMER_IRQ_STATUS  0: the ar400a wakeup timer irq is not pending 1: the ar400a wakeup timer irq is pending

			<i>Note: this is a write-1-to-clear register. This is an asynchronous operation and this bit will not become to '0' immediately after writing '1' to clear the irq state. We should read the register until the value of this bit becomes '0', which means the irq pending state is actually cleared.</i>
30:0	RW	0	AR400A_WAKE_TIMER_VAL

Table 3-27 AR400A Wakeup Timer Value Register

### 3.5.3.25 AR400A IO Wakeup Enable Register

<b>Address :</b> 0x0130			<b>Name:</b> AR400A_WAKE_IO_EN  <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:10	/	/	/
9:0	RW	0	AR400A_WAKEUP_IOx_ENB  1: IOx wakeup detection enable  0: IOx wakeup detection disable  bit0- bit9: WUPIO0 – WUPIO9

Table 3-28 AR400A IO Wakeup Enable Register

### 3.5.3.26 AR400A IO Wakeup Mode Register

<b>Address :</b> 0x0134			<b>Name:</b> AR400A_WAKE_IO_MODE  <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:10	/	/	/
9:0	RW	0	AR400A_WAKEUP_IOx_MODE  1: Positive Edge  0: Negative Edge  bit0- bit9: WUPIO0 – WUPIO9

Table 3-29 AR400A IO Wakeup Mode Register

### 3.5.3.27 AR400A IO Wakeup Status Register

<b>Address :</b> 0x0138	<b>Name:</b> AR400A_WAKE_IO_STA  <b>Default:</b> 0x0000_0000
-------------------------	--

Field	Type	Default	Description
31:10	/	/	/
9:0	RW	0	<p>AR400A_WAKEUP_IOx_ST</p> <p>1: wakeup event is detected on IOx</p> <p>0: Nothing</p> <p>bit0- bit9: WUPIO0 – WUPIO9</p> <p><i>Note: Write 1 to clear the status</i></p>

Table 3-30 AR400A IO Wakeup Status Register

### 3.5.3.28 AR400A IO Hold Control Register

<b>Address :</b> 0x013C			<b>Name:</b> AR400A_IO_HODL_CTRL  <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:10	/	/	/
9:0	RW	0	<p>AR400A_WAKEUP_IOx_ST</p> <p>1: IO Configure Hold Enable</p> <p>0: IO Configure Hold Disable</p> <p>bit0- bit9: WUPIO0 – WUPIO9</p> <p><i>Note: when enable the IO hold function, the configuration (IEN, OEN, PULL, ODAT) of the relative IO will be hold to the current state even if the GPIO controller is powered down or the origin configuration is modified.</i></p>

Table 3-31 AR400A IO Hold Control Register

### 3.5.3.29 AR400A IO Wakeup Global Enable Register

<b>Address :</b> 0x0140			<b>Name:</b> AR400A_IO_WUP_GEN  <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:1	/	/	/
0	RW	0	<p>AR400A_IO_WUP_GEN</p> <p>1: IO Wakeup Enable</p> <p>0: IO Wakeup Disable</p>

Table 3-32 AR400A IO Wakeup Global Enable Register

## 3.6 Clock Control Module (CCM)

### 3.6.1 Overview

The CCMU module manages the reset and the *HCLK* for each module and provides the configurations for each module to adjust its module clock.

### 3.6.2 Register List

Module Name	Base Address	
Clock Control Module	0x40040400	
Register Name	Offset Address	Description
CPU_BUS_CLKCFG	0x0000	CPU BUS Clock Configure Register
BUS_CLK_GATING_CTRL	0x0004	Bus Clock Gating Control Register
BUS_DEV_RST_CTRL	0x0008	Bus Device Reset Control Register
SPI0_CLK_CTRL	0x0020	SPI0 Clock Control Register
SPI1_CLK_CTRL	0x0024	SPI1 Clock Control Register
SDC_CLK_CTRL	0x0028	SDC Clock Control Register
SS_CLK_CTRL	0x002C	SS Clock Control Register
DAUDIO_CLK_CTRL	0x0034	Digital Audio Clock Control Register
IRRX_CLK_CTRL	0x0038	IRRX Clock Control Register
IRTX_CLK_CTRL	0x003c	IRTX Clock Control Register
SYSTICK_REFCLK_CTRL	0x0040	System Tick Reference Clock Register
SYSTICK_CALIB_CTRL	0x0044	System Tick Clock Calibration Register
DMIC_CLK_CTRL	0x0048	DMIC Clock Control Register
GPADC_CLK_CTRL	0x004c	GPADC Clock Control Register
CSI_DCLK_CTRL	0x0050	CSI Device Clock Control Register
FLASHC_MCLK_CTRL	0x0054	FLASH Controller Module Clock Control Register

Table 3-33 CCM Register List

### 3.6.3 Register Description

#### 3.6.3.1 BUS Clock Configure Register

<b>Address :</b> 0x0000	<b>Name:</b> CPU_BUS_CLKCFG <b>Default:</b> 0x0000_0001
-------------------------	--

Field	Type	Default	Description
31:10	/	/	/
9:8	R/W	0x0	AHB2_CLK_DIV  00: /1 01: /2 10: /3 11: /4
7:6	/	/	/
5:4	R/W	0x0	APB_CLK_SRC_SEL  APB bus clock source select 00: HFCLK 01: LFCLK 1X: AHB2 Clock
3:2	/	/	/
1:0	R/W	1	APB_CLK_DIV  00: /1 01: /2 10: /4 11: /8

**Table 3-34 CPU Clock Control Register**

### 3.6.3.2 BUS Clock Gating Control Register

<b>Address :</b> 0x0004			<b>Name:</b> BUS_CLK_GATING_CTRL <b>Default:</b>
Field	Type	Default	Description
31:28	/	/	/
27	R/W	0	GPIO_CLK_GATING  0: the clock is gated off 1: the clock is running
26	R/W	0	DMIC_CLK_GATING  0: the clock is gated off 1: the clock is running
25	R/W	0	GPADC_CLK_GATING  0: the clock is gated off 1: the clock is running

24	R/W	0	IRRX_CLK_GATING  0: the clock is gated off 1: the clock is running
23	R/W	0	IRTX_CLK_GATING  0: the clock is gated off 1: the clock is running
22	R/W	0	DAUDIO_CLK_GATING  0: the clock is gated off 1: the clock is running
21	R/W	0	PWM_ECT_CLK_GATING  0: the clock is gated off 1: the clock is running
20	/	/	/
19	R/W	0	TWI1_CLK_GATING  0: the clock is gated off 1: the clock is running
18	R/W	0	TWI0_CLK_GATING  0: the clock is gated off 1: the clock is running
17	R/W	0	UART1_CLK_GATING  0: the clock is gated off 1: the clock is running
16	R/W	0	UART0_CLK_GATING  0: the clock is gated off 1: the clock is running
15:9	/	/	/
8	R/W	0	MSGBOX_CLK_GATING  0: the clock is gated off 1: the clock is running
7	R/W	0	SPINLOCK_CLK_GATING  0: the clock is gated off 1: the clock is running
6	R/W	0	DMA_CLK_GATING

			0: the clock is gated off 1: the clock is running
5	R/W	0	CSI_CLK_GATING  0: the clock is gated off 1: the clock is running
4	R/W	0	SDC0_CLK_GATING  0: the clock is gated off 1: the clock is running
3	R/W	0	FLASHC_CLK_GATING  0: the clock is gated off 1: the clock is running
2	R/W	0	SS_CLK_GATING  0: the clock is gated off 1: the clock is running
1	R/W	0	SPI1_CLK_GATING  0: the clock is gated off 1: the clock is running
0	R/W	0	SPI0_CLK_GATING  0: the clock is gated off 1: the clock is running

Table 3-35 BUS Clock Gating Control Register

### 3.6.3.3 BUS Module Reset Control Register

Address : 0x0008			Name: BUS_DEV_RST_CTRL Default:
Field	Type	Default	Description
31:28	/	/	/
27	/	/	/
26	R/W	0	DMIC_CLK_GATING  0: the device is in reset state 1: the clock is running
25	R/W	0	GPADC_CLK_GATING  0: the device is in reset state

			1: the clock is running
24	R/W	0	IRRX_RST 0: the device is in reset state 1: the clock is running
23	R/W	0	IRTX_RST 0: the clock is in reset state 1: the clock is running
22	R/W	0	DAUDIO_RST 0: the module is in reset state 1: the module is released to work
21	R/W	0	PWM_ECT_RST 0: the module is in reset state 1: the module is released to work
20	/	/	/
19	R/W	0	TWI1_RST 0: the module is in reset state 1: the module is released to work
18	R/W	0	TWI0_RST 0: the module is in reset state 1: the module is released to work
17	R/W	0	UART1_RST 0: the module is in reset state 1: the module is released to work
16	R/W	0	UART0_RST 0: the module is in reset state 1: the module is released to work
15:9	/	/	/
8	R/W	0	MSGBOX_RST 0: the module is in reset state 1: the module is released to work
7	R/W	0	SPINLOCK_RST 0: the module is in reset state 1: the module is released to work

6	R/W	0	DMA_RST  0: the module is in reset state 1: the module is released to work
5	R/W	0	CSI_RST  0: the module is in reset state 1: the module is released to work
4	R/W	0	SDCO_RST  0: the module is in reset state 1: the module is released to work
1	R/W	0	FLASHC_RST  0: the module is in reset state 1: the module is released to work
2	R/W	0	SS_RST  0: the module is in reset state 1: the module is released to work
1	R/W	0	SPI1_RST  0: the module is in reset state 1: the module is released to work
0	R/W	0	SPI0_RST  0: the module is in reset state 1: the module is released to work

**Table 3-36 BUS Module Reset Control Register**

### 3.6.3.4 SPI0 Clock Control Register

Address : 0x0020			Name: SPI0_CLK_CTRL Default:
Field	Type	Default	Description
31	R/W	0	MCLK_ENABLE  0: the module clock is gated off 1: the module clock is released to work $CLK_m = CLK_{src} / N / M$
30:26	/	/	/
25:24	R/W	0	MCLK_SRC_SEL

			Clock source selection 00: HFCLK 01: DEVCLK 1x: reserved
23:18	/	/	/
17:16	R/W	0	CLK_DIV_RATIO_N  clock pre-divide ratio N. 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3:0	R/W	0	CLK_DIV_RATIO_M  Clock divide ratio M. M = value + 1 (1~16)

**Table 3-37 SPI0 Clock Control Register**

### 3.6.3.5 SPI1 Clock Control Register

Address : 0x0024			Name: SPI1_CLK_CTRL Default:
Field	Type	Default	Description
31	R/W	0	MCLK_ENABLE  0: the module clock is gated off 1: the module clock is released to work $CLK_m = CLK_{src} / N / M$
30:26	/	/	/
25:24	R/W	0	MCLK_SRC_SEL  Clock source selection 00: HFCLK 01: DEVCLK 1x: reserved
23:18	/	/	/
17:16	R/W	0	CLK_DIV_RATIO_N  clock pre-divide ratio N. 00: N = 1 01: N = 2

			10: N = 4 11: N = 8
15:4	/	/	/
3:0	R/W	0	CLK_DIV_RATIO_M  Clock divide ratio M. M = value + 1 (1~16)

**Table 3-38 SPI1 Clock Control Register**

### 3.6.3.6 SDC Clock Control Register

Address : 0x0028			Name: SDC0_CLK_CTRL Default:
Field	Type	Default	Description
31	R/W	0	MCLK_ENABLE  0: the module clock is gated off 1: the module clock is released to work $CLK_m = CLK_{src} / N / M$
30:26	/	/	/
25:24	R/W	0	MCLK_SRC_SEL  Clock source selection 00: HFCLK 01: DEVCLK 1x: reserved
23:18	/	/	/
17:16	R/W	0	CLK_DIV_RATIO_N  clock pre-divide ratio N. 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3:0	R/W	0	CLK_DIV_RATIO_M  Clock divide ratio M. M = value + 1 (1~16)

**Table 3-39 SDC Clock Control Register**

### 3.6.3.7SS Clock Control Register

<b>Address :</b> 0x002c			<b>Name:</b> SS_CLK_CTRL <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	R/W	0	MCLK_ENABLE  0: the module clock is gated off 1: the module clock is released to work $CLK_m = CLK_{src} / N / M$
30:26	/	/	/
25:24	R/W	0	MCLK_SRC_SEL  Clock source selection 00: HFCLK 01: DEVCLK 1x: reserved
23:18	/	/	/
17:16	R/W	0	CLK_DIV_RATIO_N  Clock pre-divide ratio N. 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3:0	R/W	0	CLK_DIV_RATIO_M  Clock divide ratio M. $M = value + 1 (1\sim16)$

Table 3-40 SS Clock Control Register

### 3.6.3.8DAUDIO Clock Control Register

<b>Address :</b> 0x0034			<b>Name:</b> DAUDIO_CLK_CTRL <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	R/W	0	MCLK_ENABLE  0: the module clock is gated off 1: the module clock is released to work $CLK_m = CLK_{src} / N / M$

30:2	/	/	/
1:0	R/W	0	<p>MCLK_SRC_SEL</p> <p>Clock source selection</p> <p>00: AUDIO_CLK (8X)</p> <p>01: PLL_AUDIO (8X) / 2</p> <p>10: PLL_AUDIO (8X) / 4</p> <p>11: PLL_AUDIO</p>

Table 3-41 DAUDIO Clock Control Register

### 3.6.3.9 IRRX Clock Control Register

Address : 0x0038			Name: IRRX_CLK_CTRL Default:
Field	Type	Default	Description
31	R/W	0	<p>MCLK_ENABLE</p> <p>0: the module clock is gated off</p> <p>1: the module clock is released to work</p> <p><math>CLK_m = CLK_{src} / N / M</math></p>
30:26	/	/	/
25:24	R/W	0	<p>MCLK_SRC_SEL</p> <p>Clock source selection</p> <p>00: HFCLK</p> <p>01: LFCLK</p> <p>1x: reserved</p>
23:18	/	/	/
17:16	R/W	0	<p>CLK_DIV_RATIO_N</p> <p>clock pre-divide ratio N.</p> <p>00: N = 1</p> <p>01: N = 2</p> <p>10: N = 4</p> <p>11: N = 8</p>
15:4	/	/	/
3:0	R/W	0	<p>CLK_DIV_RATIO_M</p> <p>Clock divide ratio M.</p> <p><math>M = \text{value} + 1 (1 \sim 16)</math></p>

Table 3-42 IRRX Clock Control Register

### 3.6.3.10 IRTX Clock Control Register

<b>Address :</b> 0x003c			<b>Name:</b> IRRX_CLK_CTRL <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	R/W	0	MCLK_ENABLE  0: the module clock is gated off 1: the module clock is released to work $CLK_m = CLK_{src} / N / M$
30:26	/	/	/
25:24	R/W	0	MCLK_SRC_SEL  Clock source selection 00: HFCLK 01: LFCLK 1x: reserved
23:18	/	/	/
17:16	R/W	0	CLK_DIV_RATIO_N  clock pre-divide ratio N. 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3:0	R/W	0	CLK_DIV_RATIO_M  Clock divide ratio M. $M = \text{value} + 1 (1\sim16)$

Table 3-43 IRTX Clock Control Register

### 3.6.3.11 System Tick Reference Clock Control Register

<b>Address :</b> 0x0040			<b>Name:</b> SYSTICK_REFCLK_CTRL <b>Default:</b> 0x00000000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	R/W	0	MCLK_ENABLE  0: the module clock is gated off 1: the module clock is released to work $CLK_m = CLK_{src} / N / M$

30:26	/	/	/
25:24	R/W	0	<b>MCLK_SRC_SEL</b>  Clock source selection 00: HFCLK 01: LFCLK 1x: reserved
23:18	/	/	/
17:16	R/W	0	<b>CLK_DIV_RATIO_N</b>  clock pre-divide ratio N. 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3:0	R/W	0	<b>CLK_DIV_RATIO_M</b>  Clock divide ratio M. M = value + 1 (1~16)

Table 3-44 System Tick Reference Clock Control Register

### 3.6.3.12 System Tick Clock Calibration Register

Address : 0x0044			<b>Name:</b> SYSTICK_CALIB_CTRL <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:26	/	/	/
25	R/W	0	<b>ST_NOREF</b> 1: there is no reference clock 0: use the reference clock
24	R/W	0	<b>ST_SKEW</b>  <i>Note: set this bit to 0 if the system timer clock, the external reference clock, or FCLK as indicated by ST_NOREF, can guarantee an exact multiple of 10ms. Otherwise, set this bit to 1.</i>
23:0	R/W	0	<b>ST_10MS_COUNTER</b>  <i>Note: This field provides an integer value to compute a 10ms (100Hz) delay from either the reference clock, or FCLK if the reference clock is not implemented.</i> <i>For example:</i> 1. No reference clock, FCLK=50MHz, CNT = 0x7A11F(49999)

		2. REFCLK = 1MHz, CNT = 0x270F(9999)
--	--	--------------------------------------

**Table 3-45 System Tick Clock Calibration Register**

### 3.6.3.13 DMIC Clock Control Register

Address : 0x0048			Name: IRRX_CLK_CTRL Default:
Field	Type	Default	Description
31	R/W	0	MCLK_ENABLE  0: the module clock is gated off 1: the module clock is released to work $CLK_m = CLK_{src} / N / M$
30:0	/	/	/

**Table 3-46 DMIC Clock Control Register**

### 3.6.3.14 GPADC Clock Control Register

Address : 0x004c			Name: GPADC_CLK_CTRL Default:
Field	Type	Default	Description
31	R/W	0	MCLK_ENABLE  0: the module clock is gated off 1: the module clock is released to work $CLK_m = CLK_{src} / N / M$
30:26	/	/	/
25:24	R/W	0	MCLK_SRC_SEL  Clock source selection 00: HFCLK 01: LFCLK 1x: reserved
23:18	/	/	/
17:16	R/W	0	CLK_DIV_RATIO_N  clock pre-divide ratio N. 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3:0	R/W	0	CLK_DIV_RATIO_M

			Clock divide ratio M. M = value + 1 (1~16)
--	--	--	---

Table 3-47 IRTX Clock Control Register

### 3.6.3.15 CSI Device Clock Control Register

Address : 0x0050			Name: CSI_DCLK_CTRL Default:
Field	Type	Default	Description
31	R/W	0	CSI_DEVICE_CLK_ENABLE  0: the module clock is gated off 1: the module clock is released to work $CLK_m = CLK_{src} / N / M$
30:26	/	/	/
25:24	R/W	0	MCLK_SRC_SEL  Clock source selection 00: HFCLK 01: DEVCLK 1x: reserved
23:18	/	/	/
17:16	R/W	0	CLK_DIV_RATIO_N  clock pre-divide ratio N. 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3:0	R/W	0	CLK_DIV_RATIO_M  Clock divide ratio M. M = value + 1 (1~16)

Table 3-48 CSI Device Clock Control Register

### 3.6.3.16 Flash Controller Clock Control Register

Address : 0x0054			Name: FLASHC_CLK_CTRL Default:
Field	Type	Default	Description

31	R/W	0	MCLK_ENABLE  0: the module clock is gated off 1: the module clock is released to work $CLK_m = CLK_{src} / N / M$
30:26	/	/	/
25:24	R/W	0	MCLK_SRC_SEL  Clock source selection 00: HFCLK 01: DEVCLK 1x: reserved
23:18	/	/	/
17:16	R/W	0	CLK_DIV_RATIO_N  clock pre-divide ratio N. 00: N = 1 01: N = 2 10: N = 4 11: N = 8
15:4	/	/	/
3:0	R/W	0	CLK_DIV_RATIO_M  Clock divide ratio M. M = value + 1 (1~16)

Table 3-49 SPI0 Clock Control Register

## 3.7 Direct Memory Access Controller (DMA)

### 3.7.1 Overview

There are 8 AHB DMA channels for this DMA controller. Only one channel can be active and the sequence is according to the priority level.

The DMA controller can support 8-bit/16-bit/32-bit data width. The data width of Source and Destination can be different, but the address should be aligned. Although the increase mode of NDMA should be address aligned, but its byte counter should not be multiple. The DMA Source Address, Destination Address, Byte Counter Registers can be modified even if the DMA is started

### 3.7.2 Register List

Module Name	Base Address

DMA Controller	0x40001000	
<b>Register Name</b>	<b>Offset Address</b>	<b>Description</b>
DMA_INT_CTRL_REG	0x00	DMA Interrupt Control Register
DMA_INT_STA_REG	0x04	DMA Interrupt Status Register
DMA_PTY_CFG_REG	0x08	DMA Priority Configure Register
NDMA_CTRL_REG	0x100+N*0x20	Normal DMA Configuration (N=0,1,2,3,4,5,6,7)
NDMA_SRC_ADDR_REG	0x100+N*0x20+0x4	Normal DMA Source Address (N=0,1,2,3,4,5,6,7)
NDMA_DEST_ADDR_REG	0x100+N*0x20+0x8	Normal DMA Destination Address (N=0,1,2,3,4,5,6,7)
NDMA_BC_REG	0x100+N*0x20+0xC	Normal DMA Byte Counter (N=0,1,2,3,4,5,6,7)

Table 3-50 DMA Register List

### 3.7.3 Register Description

#### 3.7.3.1 DMA IRQ Enable Register

<b>Address : 0x0000</b>			<b>Name:</b> DMA_IRQ_EN_REG <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:16	/	/	/
15	R/W	0x0	NDMA7_END_IRQ_EN.  Normal DMA 7 End Transfer Interrupt Enable. 0: Disable, 1: Enable.
14	R/W	0x0	NDMA7_HF_IRQ_EN  Normal DMA 7 Half Transfer Interrupt Enable. 0: Disable, 1: Enable.
13	R/W	0x0	NDMA6_END_IRQ_EN  Normal DMA 6 End Transfer Interrupt Enable. 0: Disable, 1: Enable.
12	R/W	0x0	NDMA6_HF_IRQ_EN  Normal DMA 6 Half Transfer Interrupt Enable. 0: Disable, 1: Enable.

11	R/W	0x0	NDMA5_END_IRQ_EN  Normal DMA 5 End Transfer Interrupt Enable. 0: Disable, 1: Enable.
10	R/W	0x0	NDMA5_HF_IRQ_EN  Normal DMA 5 Half Transfer Interrupt Enable. 0: Disable, 1: Enable.
9	R/W	0x0	NDMA4_END_IRQ_EN  Normal DMA 4 End Transfer Interrupt Enable. 0: Disable, 1: Enable.
8	R/W	0x0	NDMA4_HF_IRQ_EN  Normal DMA 4 Half Transfer Interrupt Enable. 0: Disable, 1: Enable.
7	R/W	0x0	NDMA3_END_IRQ_EN  Normal DMA 3 End Transfer Interrupt Enable. 0: Disable, 1: Enable.
6	R/W	0x0	NDMA3_HF_IRQ_EN  Normal DMA 3 Half Transfer Interrupt Enable. 0: Disable, 1: Enable.
5	R/W	0x0	NDMA2_END_IRQ_EN  Normal DMA 2 End Transfer Interrupt Enable. 0: Disable, 1: Enable.
4	R/W	0x0	NDMA2_HF_IRQ_EN  Normal DMA 2 Half Transfer Interrupt Enable. 0: Disable, 1: Enable.
3	R/W	0x0	NDMA1_END_IRQ_EN  Normal DMA 1 End Transfer Interrupt Enable. 0: Disable, 1: Enable.
2	R/W	0x0	NDMA1_HF_IRQ_EN  Normal DMA 1 Half Transfer Interrupt Enable. 0: Disable, 1: Enable.
1	R/W	0x0	NDMA0_END_IRQ_EN  Normal DMA 0 End Transfer Interrupt Enable.

			0: Disable, 1: Enable.
0	R/W	0x0	<p>NDMA0_HF_IRQ_EN</p> <p>Normal DMA 0 Half Transfer Interrupt Enable.</p> <p>0: Disable, 1: Enable.</p>

Table 3-51 DMA IRQ Enable Register

### 3.7.3.2 DMA IRQ Pending Status Register

<b>Address :</b> 0x0004			<b>Name:</b> DMA_IRQ_PEND_STA_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:16	/	/	/
15	R/W	0x0	<p>NDMA7_END_IRQ_PEND.</p> <p>Normal DMA 7 End Transfer Interrupt Pending. Set 1 to the bit will clear it.</p> <p>0: No effect, 1: Pending.</p>
14	R/W	0x0	<p>NDMA7_HF_IRQ_PEND.</p> <p>Normal DMA 7 Half Transfer Interrupt Pending. Set 1 to the bit will clear it.</p> <p>0: No effect, 1: Pending.</p>
13	R/W	0x0	<p>NDMA6_END_IRQ_PEND.</p> <p>Normal DMA 6 End Transfer Interrupt Pending. Set 1 to the bit will clear it.</p> <p>0: No effect, 1: Pending.</p>
12	R/W	0x0	<p>NDMA6_HF_IRQ_PEND.</p> <p>Normal DMA 6 Half Transfer Interrupt Pending. Set 1 to the bit will clear it.</p> <p>0: No effect, 1: Pending.</p>
11	R/W	0x0	<p>NDMA5_END_IRQ_PEND.</p> <p>Normal DMA 5 End Transfer Interrupt Pending. Set 1 to the bit will clear it.</p> <p>0: No effect, 1: Pending.</p>
10	R/W	0x0	<p>NDMA5_HF_IRQ_PEND.</p> <p>Normal DMA 5 Half Transfer Interrupt Pending. Set 1 to the bit will clear it.</p> <p>0: No effect, 1: Pending.</p>
9	R/W	0x0	NDMA4_END_IRQ_PEND.

			Normal DMA 4 End Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending.
8	R/W	0x0	NDMA4_END_IRQ_PEND.  Normal DMA 4 Half Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending.
7	R/W	0x0	NDMA3_END_IRQ_PEND.  Normal DMA 3 End Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending.
6	R/W	0x0	NDMA3_HF_IRQ_PEND.  Normal DMA 3 Half Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending.
5	R/W	0x0	NDMA2_END_IRQ_PEND.  Normal DMA 2 End Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending.
4	R/W	0x0	NDMA2_HF_IRQ_PEND.  Normal DMA 2 Half Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending.
3	R/W	0x0	NDMA1_END_IRQ_PEND.  Normal DMA 1 End Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending.
2	R/W	0x0	NDMA1_HF_IRQ_PEND.  Normal DMA 1 Half Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending.
1	R/W	0x0	NDMA0_END_IRQ_PEND.  Normal DMA 0 End Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending.
0	R/W	0x0	NDMA0_HF_IRQ_PEND.  Normal DMA 0 Half Transfer Interrupt Pending. Set 1 to the bit will clear it. 0: No effect, 1: Pending.

**Table 3-52 DMA IRQ Pending Status Register**

### 3.7.3.3 Normal DMA Configuration Register

<b>Address :</b> 0x100+N*0x20 (N=0,1,2,3,4,5,6,7)			<b>Name:</b> NDMA_CTRL_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	R/W	0x0	<p>DMA_LOADING.</p> <p>DMA Loading.</p> <p>If set to 1, DMA will start and load the DMA registers to the shadow registers. The bit will hold on until the DMA finished. It will be cleared automatically.</p> <p>Set 0 to the bit will reset the corresponding DMA channel.</p>
30	RO	0x0	<p>DMA Busy Status.</p> <p>0: DMA idle, 1: DMA busy.</p>
29	R/W	0x0	<p>DMA_CONTI_MODE_EN.</p> <p>DMA Continuous Mode Enable.</p> <p>0: Disable, 1: Enable.</p>
28:26	R/W	0x0	<p>DMA Wait State. (n DMA clock cycles)</p> <p>0: 1 1: 2 2: 4 3: 8 4: 16 5: 32 6: 64 7: 128</p>
25:24	R/W	0x0	<p>NDMA_DEST_DATA_WIDTH.</p> <p>Normal DMA Destination Data Width.</p> <p>00: 8-bit 01: 16-bit 10: 32-bit 11: /</p>
23	R/W	0x0	<p>DMA_DEST_BST_LEN.</p> <p>DMA Destination Burst Length.</p> <p>00: 1, 01: 4</p>
22	/	/	/
21	R/W	0x0	NDMA_DEST_ADDR_TYPE.

			Normal DMA Destination Address Type. 0: Increment 1: No Change.
20:16	R/W	0x0	<p>NDMA_DEST_DRQ_TYPE.</p> <p>Normal DMA Destination DRQ Type.</p> <p>00000 : SRAM 00001 : SPIO TX 00010 : SPI1 TX 00011 : UART0 TX 00100 : UART1 TX 00101 : SS TX 00110 : DAUDIO TX 00111 : FLASHC TX 01000 : DMIC TX others : /</p>
15	R/W	0x0	<p>BC_MODE_SEL.</p> <p>BC mode select.</p> <p>0: normal mode (the value read back is equal to the value that is written) 1: remain mode (the value read back is equal to the left bytes to be transferred).</p>
14:10	/	/	/.
9:8	R/W	0x0	<p>NDMA_SRC_DATA_WIDTH.</p> <p>Normal DMA Source Data Width.</p> <p>00: 8-bit 01: 16-bit 10: 32-bit 11: /</p>
7	R/W	0x0	<p>DMA_SRC_BST_LEN.</p> <p>DMA Source Burst Length.</p> <p>00: 1 01: 4</p>
6	/	/	/
5	R/W	0x0	<p>NDMA_SRC_ADDR_TYPE.</p> <p>Normal DMA Source Address Type.</p> <p>0: Increment 1: No Change</p>
4:0	R/W	0x0	NDMA_SRC_DRQ_TYPE.

		Normal DMA Source DRQ Type. 00000 : SRAM 00001 : SPIO RX 00010 : SPI1 RX 00011 : UART0 RX 00100 : UART1 RX 00101 : SS RX 00110 : DAUDIO RX 00111 : FLASHC RX 01000 : DMIC RX others : /
--	--	---

**Table 3-53 Normal DMA Configuration Register**

### 3.7.3.4 Normal DMA Source Address Register

<b>Address</b> : 0x100+N*0x20+0x4 (N=0,1,2,3,4,5,6,7)			<b>Name:</b> NDMA_SRC_ADDR_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0x0	NDMA_SRC_ADDR.  Normal DMA Source Address.

**Table 3-54 Normal DMA Source Address Register**

### 3.7.3.5 Normal DMA Destination Address Register

<b>Address</b> : 0x100+N*0x20+0x8 (N=0,1,2,3,4,5,6,7)			<b>Name:</b> NDMA_DEST_ADDR_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0x0	NDMA_DEST_ADDR.  Normal DMA Destination Address.

**Table 3-55 Normal DMA Destination Address Register**

### 3.7.3.6 Normal DMA Byte Counter Register

<b>Address</b> : 0x100+N*0x20+0xC (N=0,1,2,3,4,5,6,7)			<b>Name:</b> NDMA_BC_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:18	/	/	/.
17:0	R/W	0x0	NDMA_BC.  Normal DMA Byte Counter.

			<b>Note:</b> If Byte Counter=0, DMA will transfer no byte. The maximum value is 128k.
--	--	--	---

**Table 3-56 Normal DMA Byte Counter Register**

## 3.8 Real Time Clock (RTC)

### 3.8.1 Overview

The real time clock (RTC) is for calendar usage. It is built around a 30-bit counter and used to count elapsed time in YY-MM-DD and HH-MM-SS. The unit can be operated by the backup battery while the system power is off. It has a built-in leap year generator.

The alarm generates an alarm signal at a specified time in the power-off mode or normal operation mode. In normal operation mode, both the alarm interrupt and the power management wakeup are activated. In power-off mode, the power management wakeup signal is activated. In this section, there are two kinds of alarm. Alarm 0 is a general alarm; its counter is based on second. Alarm 1 is a weekly alarm; its counter is based on the real time.

### 3.8.2 Register List

Module Name	Base Address	
Real Time Clock	0x40041800	
Register Name	Offset Address	Description
RTC_CTRL	0x00	RTC Control Register
RTC_YYMMDD	0x10	RTC Year Month Day Register
RTC_HHMMSS	0x14	RTC Hour Minute Second Register
ALM0_CNT	0x20	Alarm 0 Counter Register
ALM0_CURVAL	0x24	Alarm 0 Current Value Register
ALM0_EN	0x28	Alarm 0 Enable Register
ALM0_IRQEN	0x2c	Alarm 0 IRQ Enable Register
ALM0_IRQST	0x30	Alarm 0 IRQ Status Register
ALM1_WK_HHMMSS	0x40	Alarm 1 Week Hour-Minute-Second Register
ALM1_EN	0x44	Alarm 1 Enable Register
ALM1_IRQEN	0x48	Alarm 1 IRQ Enable Register
ALM1_IRQST	0x4c	Alarm 1 IRQ Status Register
FRUN_CNT_L	0x60	Free Running Counter bit[31:0]
FRUN_CNT_H	0x64	Free Running Counter bit[47:32]

**Table 3-57 RTC Register List**

### 3.8.2.1 RTC Control Register

<b>Address : 0x00</b>			<b>Name:</b> RTC_CTRL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	R/W	0x0	RTC_TEST_MODE_CTRL. RTC TEST Mode Control bit.
30	R/W	0x0	RTC_DEBUG. RTC Simulation Control bit 0: No effect. 1: simulation mode
29:3	/	/	/
2	R	0	ALM_DDHHMMSS_ACCE. ALARM DD-HH-MM-SS access. After writing the ALARM DD-HH-MM-SS register, this bit is set and it will be cleared until the real writing operation is finished
1	R	0	RTC_HHMMSS_ACCE.  RTC HH-MM-SS access. After writing the RTC HH-MM-SS register, this bit is set and it will be cleared until the real writing operation is finished. After writing the RTC YY-MM-DD register, the YY-MM-DD register will be refreshed for at most one second.
0	R	0	RTC_YYMMDD_ACCE.  RTC YY-MM-DD access. After writing the RTC YY-MM-DD register, this bit is set and it will be cleared until the real writing operation is finished. After writing the RTC YY-MM-DD register, the YY-MM-DD register will be refreshed for at most one second.

Table 3-58 RTC Status Control Register

### 3.8.2.2 RTC YYMMDD Register

<b>Address : 0x10</b>			<b>Name:</b> RTC_YYMMDD <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:25	/	/	/.
24	R/W	0x0	LEAP. Leap Year. 0: not 1: Leap year. This bit cannot be set by hardware. It should be set or cleared by software.

23:16	R/W	x	YEAR.  Year. Range from 0~256.
15:12	/	/	/
11:8	R/W	x	MONTH.  Month. Range from 1~12.
7:5	/	/	/.
4:0	R/W	x	DAY.  Day. Range from 1~31.  <i>Note: If the written value is not from 1 to 31, it turns into 31 automatically. The month field and the year field are similar.</i>

**Table 3-59 RTC YYMMDD Register**

### 3.8.2.3 RTC HHMMSS Register

Address : 0x14			Name: RTC_HHMMSS Default: 0x0000_0000
Field	Type	Default	Description
31:29	R/W	0x0	WEEK_NO.  Week number. 000: Monday 001: Tuesday 010: Wednesday 011: Thursday 100: Friday 101: Saturday 110: Sunday 111: /
28:21	/	/	/.
20:16	R/W	x	HOUR.  Range from 0~23
15:14	/	/	/.
13:8	R/W	x	MINUTE.

			Range from 0~59
7:6	/	/	/.
5:0	R/W	x	SECOND.  Range from 0~59

Table 3-60 RTC HHMMSS Register

### 3.8.2.4 Alarm 0 Counter Register

<b>Address : 0x20</b>			<b>Name:</b> ALM0_CNT <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0	ALM0_CNT Alarm 0 Counter is based on second.

Table 3-61 Alarm 0 Counter Register

### 3.8.2.5 Alarm 0 Current Value Register

<b>Address : 0x24</b>			<b>Name:</b> ALM0_CURVAL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R	x	ALM0_CURVAL

Table 3-62 Alarm 0 Current Value Register

### 3.8.2.6 Alarm 0 Enable Register

<b>Address : 0x28</b>			<b>Name:</b> ALM0_CURVAL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:1	/	/	/
0	R/W	0	ALM0_EN  If this bit is set to 1, the alarm 0 counter register's valid bits will down count to 0 and the alarm pending bit will be set to 1 0: disable 1: enable

Table 3-63 Alarm 0 Enable Register

### 3.8.2.7 Alarm 0 IRQ Enable Register

<b>Address : 0x2c</b>	<b>Name:</b> ALM0_IRQEN <b>Default:</b> 0x0000_0000
-----------------------	--

Field	Type	Default	Description
31:1	/	/	/
0	R/W	0	<b>ALM0_IRQEN</b>  0: disable 1: enable

Table 3-64 Alarm 0 IRQ Enable Register

### 3.8.2.8 Alarm 0 IRQ Status Register

<b>Address : 0x30</b>			<b>Name:</b> ALM0_IRQST <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:1	/	/	/
0	R/W	0	<b>ALM0_IRQST</b>  0: no effect 1: pending

Table 3-65 Alarm 0 IRQ Status Register

### 3.8.2.9 Alarm 1 Week HH-MM-SS Register

<b>Address : 0x40</b>			<b>Name:</b> Alarm 1_WK_HHMMSS <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:21	/	/	/
20:16	R/W	X	<b>HOUR</b> Range from 0-23
15:14	/	/	/
13:8	R/W	X	<b>MINUTE</b> Range from 0-59
7:6	/	/	/
5:0	R/W	X	<b>SECOND</b> Range from 0-59

Table 3-66 Alarm 1 Week HH-MM-SS Register

### 3.8.2.10 Alarm 1 Enable Register

<b>Address : 0x44</b>			<b>Name:</b> ALM1_EN <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:7	/	/	/

6	R/W	0	SUN_ALM1_EN  Sunday Alarm 1 Enable  0: disable  1: enable  If this bit is set to “1”, only when the Alarm 1 Week HH-MM-SS register valid bits is equal to RTC HH-MM-SS register and the register RTC HH-MM-SS bit [31:29] is 6, the alarm IRQ pending bit will be set to “1”.
5	R/W	0	SAT_ALM1_EN  Saturday Alarm 1 Enable  0: disable  1: enable  If this bit is set to “1”, only when the Alarm 1 Week HH-MM-SS register valid bits is equal to RTC HH-MM-SS register and the register RTC HH-MM-SS bit [31:29] is 5, the alarm IRQ pending bit will be set to “1”.
4	R/W	0	FRI_ALM1_EN  Friday Alarm 1 Enable  0: disable  1: enable  If this bit is set to “1”, only when the Alarm 1 Week HH-MM-SS register valid bits is equal to RTC HH-MM-SS register and the register RTC HH-MM-SS bit [31:29] is 4, the alarm IRQ pending bit will be set to “1”.
3	R/W	0	THU_ALM1_EN  Thursday Alarm 1 Enable  0: disable  1: enable  If this bit is set to “1”, only when the Alarm 1 Week HH-MM-SS register valid bits is equal to RTC HH-MM-SS register and the register RTC HH-MM-SS bit [31:29] is 3, the alarm IRQ pending bit will be set to “1”.
2	R/W	0	WED_ALM1_EN

			<p>Wednesday Alarm 1 Enable</p> <p>0: disable</p> <p>1: enable</p> <p>If this bit is set to “1”, only when the Alarm 1 Week HH-MM-SS register valid bits is equal to RTC HH-MM-SS register and the register RTC HH-MM-SS bit [31:29] is 2, the alarm IRQ pending bit will be set to “1”.</p>
1	R/W	0	<p>TUE_ALM1_EN</p> <p>Tuesday Alarm 1 Enable</p> <p>0: disable</p> <p>1: enable</p> <p>If this bit is set to “1”, only when the Alarm 1 Week HH-MM-SS register valid bits is equal to RTC HH-MM-SS register and the register RTC HH-MM-SS bit [31:29] is 1, the alarm IRQ pending bit will be set to “1”.</p>
0	R/W	0	<p>MON_ALM1_EN</p> <p>Monday Alarm 1 Enable</p> <p>0: disable</p> <p>1: enable</p> <p>If this bit is set to “1”, only when the Alarm 1 Week HH-MM-SS register valid bits is equal to RTC HH-MM-SS register and the register RTC HH-MM-SS bit [31:29] is 0, the alarm IRQ pending bit will be set to “1”.</p>

Table 3-67 Alarm 1 Enable Register

### 3.8.2.11 Alarm 1 IRQ Enable Register

<b>Address :</b> 0x48			<b>Name:</b> ALM1_IRQEN <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:1	/	/	/
0	R/W	0	<p>ALM1_IRQEN</p> <p>0: disable</p> <p>1: enable</p>

Table 3-68 Alarm 1 IRQ Enable Register

### 3.8.2.12 Alarm 1 IRQ Status Register

<b>Address :</b> 0x4c	<b>Name:</b> ALM1_IRQST <b>Default:</b> 0x0000_0000
-----------------------	--

Field	Type	Default	Description
31:1	/	/	/
0	R/W	0	ALM1_WEEK_IRQST Alarm 1 week(0-7) IRQ Pending 0: No effect 1: pending

Table 3-69 Alarm 1 IRQ Enable Register

### 3.8.2.13 Free Running Counter Low Register

<b>Address :</b> 0x60			<b>Name:</b> FREERUN_CNT_L <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R/W	0	Free Running Counter bit[31:0]  <i>Note: Free running counter is a 48-bit counter which is driven by LFCLK and starts to count as soon as the system reset is released and the LFCLK is ready. Write will reset the counter value.</i>

Table 3-70 Free Running Counter Low Register

### 3.8.2.14 Free Running Counter High Register

<b>Address :</b> 0x64			<b>Name:</b> FREERUN_CNT_H <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R/W	0	Free Running Counter bit[47:32]  <i>Note: Free running counter is a 48-bit counter which is driven by LFCLK and starts to count as soon as the system reset is released and the LFCLK is ready. Write will reset the counter value.</i>

Table 3-71 Free Running Counter High Register

## 3.9 Timer Controller (Timer)

### 3.9.1 Overview

Timer 0 and 1 can take their inputs from internal RC oscillator, external 32768Hz crystal or OSC. They provide the operating system's scheduler interrupt. It is designed to offer maximum accuracy and efficient management, even for systems with long or short response time. They provide 24-bit programmable overflow counter and work in auto-reload mode or no-reload mode.

The watch-dog is used to resume the controller operation when it had been disturbed by malfunctions such as noise and system errors. It features a down counter that allows a watchdog period of up to 16 seconds. It can generate a

general reset or an interrupt request.

### 3.9.2 Register List

Module Name	Base Address	
Timer	0x40040800	
Register Name	Offset Address	Description
TMR_IRQ_EN_REG	0x00	Timer IRQ Enable Register
TMR_IRQ_STA_REG	0x04	Timer Status Register
TMRO_CTRL_REG	0x10	Timer 0 Control Register
TMRO_INTV_VALUE_REG	0x14	Timer 0 Interval Value Register
TMRO_CUR_VALUE_REG	0x18	Timer 0 Current Value Register
TMR1_CTRL_REG	0x20	Timer 1 Control Register
TMR1_INTV_VALUE_REG	0x24	Timer 1 Interval Value Register
TMR1_CUR_VALUE_REG	0x28	Timer 1 Current Value Register
WDOG_IRQ_EN_REG	0xA0	Watchdog IRQ Enable Register
WDOG_IRQ_STA_REG	0xA4	Watchdog IRQ Status Register
WDOG_CTRL_REG	0xB0	Watchdog Control Register
WDOG_CFG_REG	0xB4	Watchdog Configuration Register
WDOG_MODE_REG	0xB8	Watchdog Mode Register
WDOG_OUTPUT_CTRL_REG	0xBC	Watchdog Output Control Register

Table 3-72 Timer Register List

### 3.9.3 Register Description

#### 3.9.3.1 Timer IRQ Enable Register

<b>Address : 0x0000</b>			<b>Name:</b> TMR_IRQ_EN_REG <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:2	/	/	/.
1	R/W	0x0	TMR1_IRQ_EN.  Timer 1 Interrupt Enable. 0: No effect 1: Timer 1 Interval Value reached interrupt enable.
0	R/W	0x0	TMRO_IRQ_EN.  Timer 0 Interrupt Enable.

			0: No effect 1: Timer 0 Interval Value reached interrupt enable.
--	--	--	---

**Table 3-73 Timer IRQ Enable Register**

### 3.9.3.2 Timer IRQ Status Register

<b>Address :</b> 0x0004			<b>Name:</b> TMR_IRQ_STA_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:2	/	/	/
1	R/W	0x0	<p>TMR1_IRQ_PEND.</p> <p>Timer 1 IRQ Pending. Set 1 to the bit will clear it.</p> <p>0: No effect, 1: Pending, timer 1 interval value is reached.</p>
0	R/W	0x0	<p>TMRO_IRQ_PEND.</p> <p>Timer 0 IRQ Pending. Set 1 to the bit will clear it.</p> <p>0: No effect, 1: Pending, timer 0 interval value is reached.</p>

**Table 3-74 Timer IRQ Status Register**

### 3.9.3.3 Timer 0 Control Register

<b>Address :</b> 0x0010			<b>Name:</b> TMRO_CTRL_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/.
7	R/W	0x0	<p>TMRO_MODE.</p> <p>Timer 0 mode.</p> <p>0: Continuous mode. When interval value reached, the timer will not disable automatically.</p> <p>1: Single mode. When interval value reached, the timer will disable automatically.</p>
6:4	R/W	0x0	<p>TMRO_CLK_PRES.</p> <p>Select the pre-scale of timer 0 clock source.</p> <p>000: /1 001: /2 010: /4 011: /8</p>

			100: /16 101: /32 110: /64 111: /128
3:2	R/W	0x1	TMRO_CLK_SRC.  Timer 0 Clock Source. 00: LOSC, 01: HOSC. 10: / 11: /.
1	R/W	0x0	TMRO_RELOAD.  Timer 0 Reload. 0: No effect, 1: Reload timer 0 Interval value.
0	R/W	0x0	TMRO_EN.  Timer 0 Enable. 0: Stop/Pause 1: Start.  If the timer is started, it will reload the interval value to internal register, and the current counter will count from interval value to 0. If the current counter does not reach the zero, the timer enable bit is set to "0"; the current value counter will pause. At least wait for $2 \times T_{cycle}$ , the start bit can be set to 1. In timer pause state, the interval value register can be modified. If the timer is started again, and the Software hope the current value register to down-count from the new interval value, the reload bit and the enable bit should be set to 1 at the same time.

Table 3-75 Timer 0 Control Register

### 3.9.3.4 Timer 0 Interval Value Register

<b>Address :</b> 0x0014			<b>Name:</b> TMRO_INTV_VALUE_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	x	TMRO_INTV_VALUE.  Timer 0 Interval Value. <b>Note:</b> the value setting should consider the system clock and the timer clock source.

Table 3-76 Timer 0 Interval Value Register

### 3.9.3.5 Timer 0 Current Value Register

<b>Address :</b> 0x0018			<b>Name:</b> TMR0_CUR_VALUE_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0x0	TMR0_CUR_VALUE.  Timer 0 Current Value.

Table 3-77 Timer 0 Current Value Register

### 3.9.3.6 Timer 1 Control Register

<b>Address :</b> 0x0020			<b>Name:</b> TMR1_CTRL_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/.
7	R/W	0x0	TMR1_MODE.  Timer 1 mode. 0: Continuous mode. When interval value reached, the timer will not disable automatically. 1: Single mode. When interval value reached, the timer will disable automatically.
6:4	R/W	0x0	TMR1_CLK_PRES.  Select the pre-scale of timer 1 clock source. 000: /1 001: /2 010: /4 011: /8 100: /16 101: /32 110: /64 111: /128
3:2	R/W	0x1	TMR1_CLK_SRC.  Timer 1 Clock Source. 00: LOSC, 01: HOSC. 10: / 11: /.
1	R/W	0x0	TMR1_RELOAD.

			Timer 1 Reload. 0: No effect 1: Reload timer 1 Interval value.
0	R/W	0x0	<p>TMR1_EN.</p> <p>Timer 1 Enable. 0: Stop/Pause 1: Start.</p> <p>If the timer is started, it will reload the interval value to internal register, and the current counter will count from interval value to 0.</p> <p>If the current counter does not reach the zero, the timer enable bit is set to "0"; the current value counter will pause. At least wait for <math>2 \times T_{cycle}</math>, the start bit can be set to 1.</p> <p>In timer pause state, the interval value register can be modified. If the timer is started again, and the Software hope the current value register to down-count from the new interval value, the reload bit and the enable bit should be set to 1 at the same time.</p>

**Table 3-78 Timer 1 Control Register**

### 3.9.3.7 Timer 1 Interval Value Register

<b>Address :</b> 0x0024			<b>Name:</b> TMR1_INTV_VALUE_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	x	<p>TMR1_INTV_VALUE.</p> <p>Timer 1 Interval Value.</p> <p><b>Note:</b> the value setting should consider the system clock and the timer clock source.</p>

**Table 3-79 Timer 1 Interval Value Register**

### 3.9.3.8 Timer 1 Current Value Register

<b>Address :</b> 0x0028			<b>Name:</b> TMR1_CUR_VALUE_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0x0	<p>TMR1_CUR_VALUE.</p> <p>Timer 1 Current Value.</p>

**Table 3-80 Timer 1 Current Value Register**

### 3.9.3.9 Watchdog IRQ Enable Register

<b>Address :</b> 0x00A0			<b>Name:</b> WDOG_IRQ_EN_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:1	/	/	/
0	R/W	0	<p>WDOG_IRQ_EN</p> <p>0: No effect 1: Watchdog interrupt enable</p>

Table 3-81 Watchdog IRQ Enable Register

### 3.9.3.10 Watchdog IRQ Status Register

<b>Address :</b> 0x00A4			<b>Name:</b> WDOG_IRQ_EN_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:1	/	/	/
0	R/W	0	<p>WDOG_IRQ_STA</p> <p>Watchdog irq pending. Write 1 to clear it</p> <p>0: No effect 1: Pending, the interval value is reached.</p>

Table 3-82 Watchdog IRQ Enable Register

### 3.9.3.11 Watchdog Control Register

<b>Address :</b> 0x00B0			<b>Name:</b> WDOG_CTRL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:13	/	/	/
12:1	R/W	0	<p>WDOGO_KEY_FIELD.</p> <p>Watchdog 0 Key Field.</p> <p>Should be written at value 0xA57. Writing any other value in this field aborts the write operation.</p>
0	R/W	0	<p>WDOG_RELOAD</p> <p>Watchdog Reload</p> <p>0: no effect 1: reload the watchdog</p>

Table 3-83 Watchdog Control Register

### 3.9.3.12 Watchdog Configuration Register

<b>Address :</b> 0x00B4			<b>Name:</b> WDOG_CTRL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:2	/	/	/
8	R/W	0	CLK_SEL  0: 32K 1: 32.768K
7:2	/	/	/
1:0	R/W	1	WDOGO_CONFIG. 00: / 01: reset whole system 10: only interrupt 11: /

Table 3-84 Watchdog Configuration Register

### 3.9.3.13 Watchdog Mode Register

<b>Address :</b> 0x00B8			<b>Name:</b> WDOG_MODE <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:4	R/W	0	WDOG_INTV_VALUE  Watchdog Interval Value The watchdog clock source is HOSC/WDOG_PRESCALER. If the HOSC is turned off, the watchdog will not work.  0000: 16000 cycles (0.5s) 0001: 32000 cycles (1s) 0010: 64000 cycles (2s) 0011: 96000 cycles (3s) 0100: 128000 cycles (4s) 0101: 160000 cycles (5s) 0110: 192000 cycles (6s) 0111: 256000 cycles (8s) 1000: 320000 cycles (10s) 1001: 384000 cycles (12s) 1010: 448000 cycles (14s) 1011: 512000 cycles (16s)

			others: /
3:1	/	/	/
0	R/W	0	<p>WDOG_EN</p> <p>Watchdog Enable</p> <p>0: no effect</p> <p>1: Enable watchdog</p>

**Table 3-85 Watchdog Mode Control Register**

### 3.9.3.14 Watchdog Output Control Register

<b>Address :</b> 0x00BC			<b>Name:</b> WDOG_OUTPUT_CTRL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:5	/	/	/
4:0	R/W	0xA	<p>WDOG_OUTPUT_CONTROL</p> <p>0~31</p> <p>the number of cycles with which the watch dog holds the reset signal</p>

**Table 3-86 Watchdog Output Control Register**

## 3.10 Security ID (SID)

### 3.10.1 Overview

There is a 2048-bit on chip EFUSE, with size up to 128 bits for security chip ID. The other bits are reserved for system parameters. The work clock of Security ID is 26MHz, which clock is sourced from APB clock.

The Security ID includes the following features:

- 2048-bit on chip eFuse
- 128-bit electrical fuses for chip ID
- Other bits are used for system parameter.

### 3.10.2 Register List

<b>Module Name</b>	<b>Base Address</b>	
Security ID	0x40043C00	
<b>Register Name</b>	<b>Offset Address</b>	<b>Description</b>
SID_PRCTL	0x40	SID Program/Read Control Register
SID_PKEY	0x50	SID Program Key Value Register

SID_RKEY	0x60	SID Read Key Value Register
BURNED_TIMING	0x90	SID Burned Timing Control
SDI_DBG	0x94	SID Debug Register
SID_VALUE	0x100 – 0x1FF	SID Chip ID Register 0

**Table 3-87 Security ID Register List**

### 3.10.3 Register Description

#### 3.10.3.1 SID Program/Read Control Register

<b>Address :</b> 0x0040			<b>Name:</b> SID_PR_CTRL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:29	/	/	/
28	R/W	0	EFUSE_CLOCK_GATE_SW  OSC40M (eFuse_clock) clock gate enable. When read or program the effuse, this bit should be set to 1
27:24	/	/	/
23:16	R/W	0	PG_INDEX  Program Index  The index value of 8-bits electrical fuse hardware macrocell address offset, and the lowest two bits must be zero.
15:8	R/W	0	OP_LOCK  Operation Lock  The Read Start Bit (bit1) and Program Start bit (bit0) only can be written when these bits (bit[15:8]) set to 0xAC
7:3	/	/	/
2	R	x	HW_READ_STATUS  Hardware read status 0: No hardware operation 1: Hardware Reading
1	R/W	0	EFUSE_READ_START  Software Read Start  Write '1' to start software read and automatically clear to '0' after read.

0	R/W	0	EFUSE_PRG_START  Software Program Start  Write '1' to start software program and automatically clear to '0' after program.
---	-----	---	--

**Table 3-88 SID Program/Read Control Register**

### 3.10.3.2 SID Program Key Value Register

<b>Address :</b> 0x0050			<b>Name:</b> SID_PKEY <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0	SID_PGKEY_VALUE  Program key value

**Table 3-89 SID Program Key Value Register**

### 3.10.3.3 SID Read Key Value Register

<b>Address :</b> 0x0060			<b>Name:</b> SID_PKEY <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R	0	SID_RDKEY_VALUE  Program key value

**Table 3-90 SID Program Key Value Register**

### 3.10.3.4 SID Burn Timing Register

<b>Address :</b> 0x0090			<b>Name:</b> SID_BURN_TIMING <b>Default:</b> 0x630B_905C
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:28	R/W	0x6	Control $T_{aen}$ of the eFuse Reading cycle value * $T_{osc}$  $T_{aen} > T_{rd} + 35\text{ns}$
28:24	R/W	0x3	Control $T_{rd}$ of the eFuse Reading cycle value * $T_{osc}$  $T_{rd} > 40\text{ns}$
23:12	R/W	0X0B9	Control $T_{aen}$ of the eFuse program cycle

			value * $T_{osc}$  $T_{aen} > T_{pgm} + 2300\text{ns}$
			Control $T_{pgm}$ of the eFuse Program cycle value * $T_{osc}$
11:0	R/W	0X05C	$T_{pgm} \geq 2300\text{ns}$ ( $2600\text{ns} > T_{pgm} > 2000\text{ns}$ )

**Table 3-91 SID Burn Timing Register**

### 3.10.3.5 SID Debug Register

<b>Address :</b> 0x0094			<b>Name:</b> SID_DBG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:5	/	/	/
4	R/W	0	LDO-EFUSE software switch 0: soft switch off 1: soft switch on <i>Note: Switch on accumulative total time should less than 1 second!!!</i>
3:0	/	/	/

**Table 3-92 SID Debug Register**

### 3.10.3.6 SID Register

<b>Address :</b> 0x100 – 0x1FF			<b>Name:</b> SID_VALUE <b>Default:</b> 0XXXXXXXX
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R	x	SID [32n+31: 32n]

**Table 3-93 SID Register**

## 3.11 Message Box (MSGBOX)

### 3.11.1 Overview

Message Box provides an *MSGBOX*-interrupt mechanism for on-chip processors intercommunication. It allows a processor transmit messages to the other one or receive messages from the other through a series of Message Queues, each of which is a four 32-bits depth FIFO. An intercommunication channel could be established by configuring Message Box registers and it works under *MSGBOX*-interrupt mechanism.

The Message Box includes the following features:

- Two users for Message Box instance(User0 for CPUS and User1 for C0-CPUX/C1-CPUX)

- Eight Message Queues and each of Queues is a four 32-bits depth FIFO for establishing intercommunication channel
- Each of Queues could be configured as transmitter or receiver for user
- Message reception and queue-not-full notification interrupt mechanism

### 3.11.2 Typical Applications

Message Box is typically designed for making the on-chip processors interconnection be true. It could establish an interconnection channel between processors by configuring a set of Message Box registers. Each of Message Queues is bidirectional for users, which means, while a message queue is configured as a receiver for a user, it is a transmitter for the interconnect user beside. If a processor would like to interconnect with the other processor, it should configure one or more Message Queues firstly. Although Message Box provides two interrupt mechanism to notice user to transmit or receive messages, the way check out queue FIFO full status is usually adopted before transmitting a message and receiving a message is still depended on the reception notification. The Message Box is usually applied as the below flow chart:

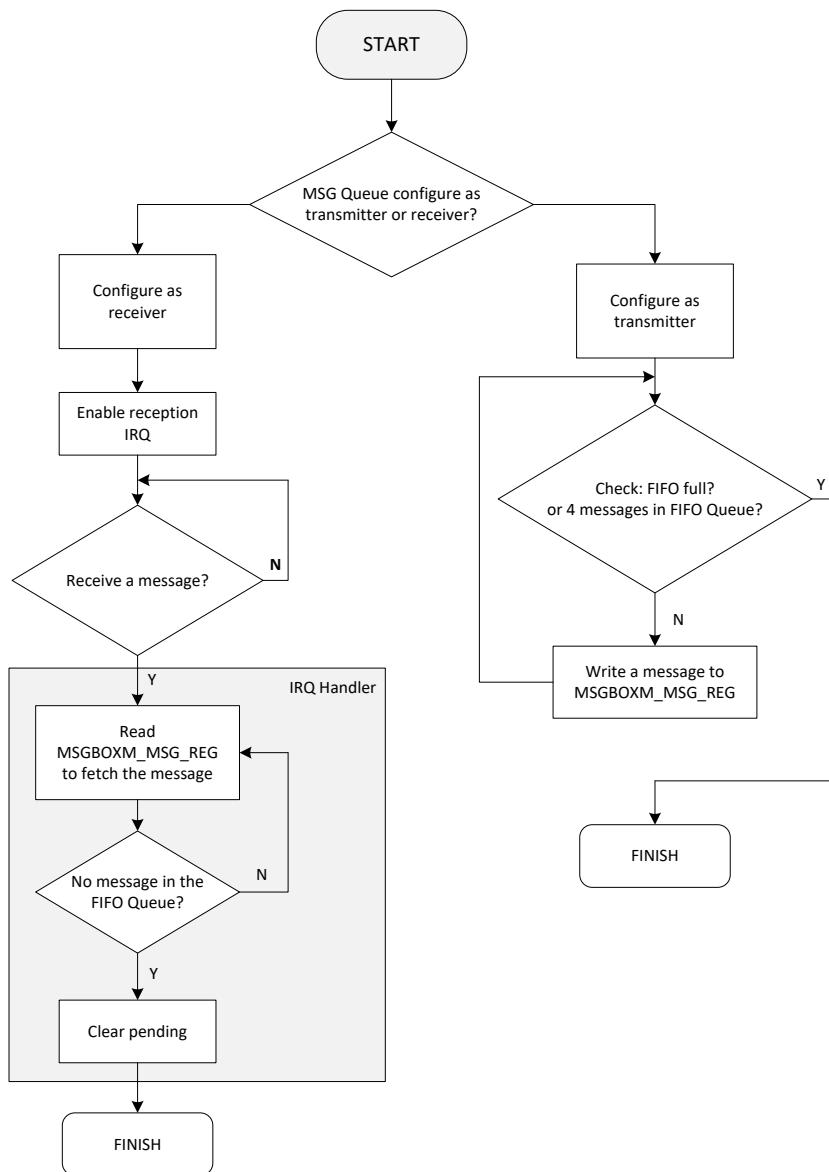


Figure 3-3 Message Box Typical Application Chart

### 3.11.3 Functional Block Diagram

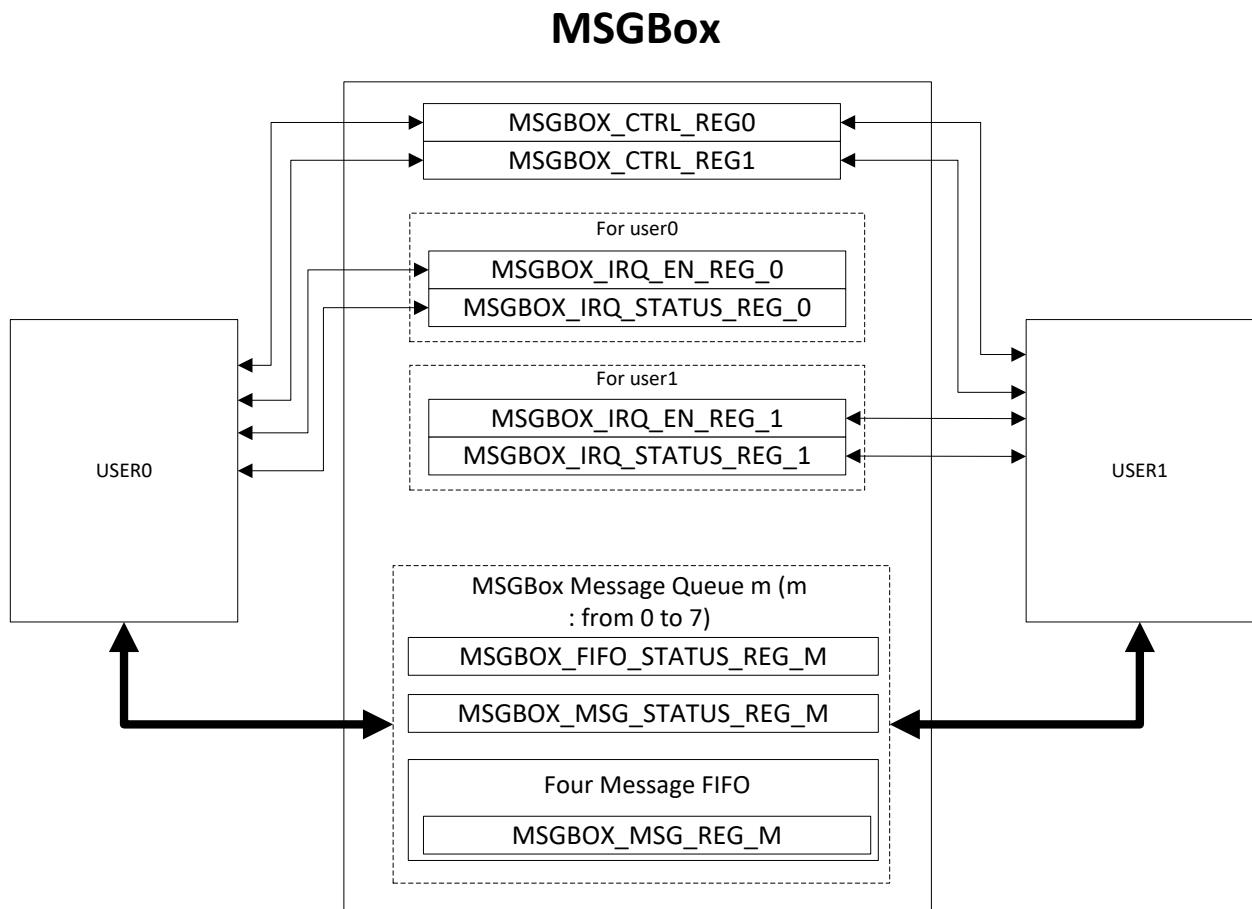


Figure 3-4 Message Box Functional Block Diagram

Message Box supports a set of registers for a processor to establish an interconnection channel with the others. The processor determines message queue numbers for interconnection and the used queues to be transmitter or receiver for itself and the interconnect one. Every queue has a *MSGBOX FIFO Status Register* for processor to check out queue FIFO full status and a *MSGBOX Message Status Register* for processor to check out message numbers in queue FIFO. Otherwise, every queue has a corresponding IRQ status bit and a corresponding IRQ enable bit, which used for requesting an interrupt.

### 3.11.4 Operation Principle

- Message Box clock gating and software reset

By default the Message Box clock gating is mask. When it is necessary to use Message Box, its clock gating should be open in *AHB Module Clock Gating Register* and then de-assert the software reset in *AHB Module Software Reset Register* on CCM module. If it is no need to use Message Box, both the gating bit and software reset bit should set 0.

- Message Queue Assignment

When a processor needs to transmit or receive a message from the other one, it should configure the Message Queue assignment for the other one and itself. *MSGBOX\_CTRL\_REG0* and *MSGBOX\_CTRL\_REG1* hold the eight Message Queues assignment. For an instance, *RECEPTION\_MQ0* bit is set to 0 and *TRANSMIT\_MQ0* bit is set to 1, which means, user1 transmits messages and user0 receives them. Or *RECEPTION\_MQ0* bit and *TRANSMIT\_MQ0* bit are both set to 0, which means user0 transmits messages to itself.

- Interrupt request

Message Box provides Message reception and queue-not-full notification interrupt mechanism. For a Message Queue configured as transmitter for a user, this queue transmit pending bit will always be set to 1 of this user if it is not full. For a Message Queue configured as receiver for a user, this queue reception pending bit will be set to 1 for this user only if it receives a new message. For example, Message Queue0 is configured as a transmitter for user0 and a receiver for user1. The thing Message Queue0 is not full always makes *TRANSMIT\_MQ0\_IRQ\_PEND* bit set to 1. If *TRANSMIT\_MQ0\_IRQ\_EN* bit is set to 1, user0 will request a queue-not-full interrupt. When Message Queue0 has received a new message, *RECEPTION\_MQ0\_IRQ\_PEND* bit would be set to 1 and user1 will request a new message reception interrupt if *RECEPTION\_MQ0\_IRQ\_EN* bit is set to 1. *MSGBOX IRQ Status Register u(u=0,1)* hold the IRQ status for user0 and user1. *MSGBOX IRQ Enable Register u(u=0,1)* determine whether the user could request the interrupt or not.

- Transmit and receive messages

Every Message Queue has a couple of private registers for query: *MSGBOX Message Status Register* and *MSGBOX FIFO Status Register* and a store register bridged to Message Queue FIFO: *MSGBOX Message Queue Register*. MSGBox Message Status Register records present message number in the Message Queue. *MSGBOX FIFO Status Register* indicates whether the Message Queue is full obviously. *MSGBOX Message Queue Register* stores the next to be read message of the message FIFO queue or the message to be wrote into the queue FIFO. The thing that queue is not full usually indicates that you could write messages into the queue FIFO and that there is one or more message in the queue FIFO indicates that you could read messages from the queue FIFO.

Writing a message into the queue FIFO realizes a transmission and reading a message makes a reception ture. You could transmit messages by writing messages to *MSGBOX Message Queue Register* continuously or receive messages by reading *MSGBOX Message Queue Register* continuously. The wiring or reading operation could be continuous means it's no need to make a delay between operations.

### 3.11.5 Register List

Module Name	Base Address	
Message Box	0x40009000	
Register Name	Offset Address	Description
<i>MSGBOX_CTRL_REG0</i>	0x0000	Message Queue attribute control register 0
<i>MSGBOX_CTRL_REG1</i>	0x0004	Message Queue attribute control register 1
<i>MSGBOXU_IRQ_EN_REG</i>	0x0040+n*0x20	IRQ enable for user n (n=0,1)
<i>MSGBOXU_IRQ_STATUS_REG</i>	0x0050+n*0x20	IRQ status for user n (n=0,1)

MSGBOXM_FIFO_STATUS_REG	0x0100+N*0x4	FIFO status for message queue N(N = 0~7)
MSGBOXM_MSG_STATUS_REG	0x0140+N*0x4	Message Status for message queue N(N=0~7)
MSGBOXM_MSG_REG	0x0180+N*0x4	Message Register for message queue N(N=0~7)
MSGBOX_DEBUG_REG	0x01C0	MSGBOX debug register

**Table 3-94 MSGBOX Register List**

### 3.11.6 Register Description

#### 3.11.6.1 MSGBOX Control Register 0

Address : 0x0000			Name: MSGBOX_CTRL_REG0 Default: 0x1010_1010
Field	Type	Default	Description
31:29	/	/	/
28	R/W	0x1	TRANSMIT_MQ3. Message Queue 3 is a Transmitter of user u. 0: user0 1: user1
27:25	/	/	/
24	R/W	0x0	RECEPTION_MQ3. Message Queue 3 is a Receiver of user u. 0: user0 1: user1
23:21	/	/	/
20	R/W	0x1	TRANSMIT_MQ2. Message Queue 2 is a Transmitter of user u. 0: user0 1: user1
19:17	/	/	/
16	R/W	0x0	RECEPTION_MQ2. Message Queue 2 is a Receiver of user u. 0: user0 1: user1
15:13	/	/	/
12	R/W	0x1	TRANSMIT_MQ1 Message Queue 1 is a Transmitter of user u. 0: user0 1: user1
11:9	/	/	/
8	R/W	0x0	RECEPTION_MQ1. Message Queue 1 is a Receiver of user u.

			0: user0 1: user1
7:5	/	/	/
4	R/W	0x1	TRANSMIT_MQ0. Message Queue 0 is a Transmitter of user u. 0: user0 1: user1
3:1	/	/	/
0	R/W	0x0	RECEPTION_MQ0. Message Queue 0 is a Receiver of user u. 0: user0 1: user1

**Table 3-95 MSGBOX Control Register 0**

### 3.11.6.2 MSGBOX Control Register 1

Address : 0x0004			Name: MSGBOX_CTRL_REG1 Default: 0x0000_0000
Field	Type	Default	Description
31:29	/	/	/
28	R/W	0x1	TRANSMIT_MQ7. Message Queue 7 is a Transmitter of user u. 0: user0 1: user1
27:25	/	/	/
24	R/W	0x0	RECEPTION_MQ7. Message Queue 7 is a Receiver of user u. 0: user0 1: user1
23:21	/	/	/
20	R/W	0x1	TRANSMIT_MQ6. Message Queue 6 is a Transmitter of user u. 0: user0 1: user1
19:17	/	/	/
16	R/W	0x0	RECEPTION_MQ6. Message Queue 6 is a Receiver of user u. 0: user0 1: user1
15:13	/	/	/
12	R/W	0x1	TRANSMIT_MQ5 Message Queue 5 is a Transmitter of user u.

			0: user0 1: user1
11:9	/	/	/
8	R/W	0x0	RECEPTION_MQ5. Message Queue 5 is a Receiver of user u. 0: user0 1: user1
7:5	/	/	/
4	R/W	0x1	TRANSMIT_MQ4. Message Queue 4 is a Transmitter of user u. 0: user0 1: user1
3:1	/	/	/
0	R/W	0x0	RECEPTION_MQ4. Message Queue 4 is a Receiver of user u. 0: user0 1: user1

**Table 3-96 MSGBOX Control Register 1**

### 3.11.6.3 MSGBOX IRQ Enable Register u(u=0,1)

Address : Offset:0x40+N*0x20 (N=0,1)			Name: MSGBOXU_IRQ_EN_REG Default: 0x0000_0000
Field	Type	Default	Description
31:16	/	/	/
15	R/W	0x0	TRANSMIT_MQ7_IRQ_EN. 0: Disable 1: Enable (It will Notify user u by interrupt when Message Queue 7 is not full.)
14	R/W	0x0	RECEPTION_MQ7_IRQ_EN. 0: Disable 1: Enable (It will notify user u by interrupt when Message Queue 7 has received a new message.)
13	R/W	0x0	TRANSMIT_MQ6_IRQ_EN. 0: Disable 1: Enable (It will Notify user u by interrupt when Message Queue 6 is not full.)
12	R/W	0x0	RECEPTION_MQ6_IRQ_EN. 0: Disable 1: Enable (It will notify user u by interrupt when Message Queue 6 has received a new message.)
11	R/W	0x0	TRANSMIT_MQ5_IRQ_EN.

			0: Disable 1: Enable (It will Notify user u by interrupt when Message Queue 5 is not full.)
10	R/W	0x0	RECEPTION_MQ5_IRQ_EN. 0: Disable 1: Enable (It will notify user u by interrupt when Message Queue 5 has received a new message.)
9	R/W	0x0	TRANSMIT_MQ4_IRQ_EN. 0: Disable 1: Enable (It will Notify user u by interrupt when Message Queue 4 is not full.)
8	R/W	0x0	RECEPTION_MQ4_IRQ_EN. 0: Disable 1: Enable (It will notify user u by interrupt when Message Queue 4 has received a new message.)
7	R/W	0x0	TRANSMIT_MQ3_IRQ_EN. 0: Disable 1: Enable (It will Notify user u by interrupt when Message Queue 3 is not full.)
6	R/W	0x0	RECEPTION_MQ3_IRQ_EN. 0: Disable 1: Enable (It will notify user u by interrupt when Message Queue 3 has received a new message.)
5	R/W	0x0	TRANSMIT_MQ2_IRQ_EN. 0: Disable 1: Enable (It will Notify user u by interrupt when Message Queue 2 is not full.)
4	R/W	0x0	RECEPTION_MQ2_IRQ_EN. 0: Disable 1: Enable (It will notify user u by interrupt when Message Queue 2 has received a new message.)
3	R/W	0x0	TRANSMIT_MQ1_IRQ_EN. 0: Disable 1: Enable (It will Notify user u by interrupt when Message Queue 1 is not full.)
2	R/W	0x0	RECEPTION_MQ1_IRQ_EN. 0: Disable 1: Enable (It will notify user u by interrupt when Message Queue 1 has received a new message.)
1	R/W	0x0	TRANSMIT_MQ0_IRQ_EN. 0: Disable 1: Enable (It will Notify user u by interrupt when Message Queue 0 is not full.)

0	R/W	0x0	RECEPTION_MQ0_IRQ_EN. 0: Disable 1: Enable (It will notify user u by interrupt when Message Queue 0 has received a new message.)
---	-----	-----	--

**Table 3-97 MSGBOX IRQ Enable Register u(u=0,1)**

### 3.11.6.4 MSGBOX IRQ Status Register u

Address : Offset:0x50+N*0x20 (N=0,1)			Name: MSGBOXU_IRQ_STATUS_REG Default: 0x0000_0000
Field	Type	Default	Description
31:16	/	/	/
15	R/W	0x1	TRANSMIT_MQ7_IRQ_PEND. 0: No effect, 1: Pending. This bit will be pending for user u when Message Queue 7 is not full. Set one to this bit will clear it.
14	R/W	0x0	RECEPTION_MQ7_IRQ_PEND. 0: No effect, 1: Pending. This bit will be pending for user u when Message Queue 7 has received a new message. Set one to this bit will clear it.
13	R/W	0x1	TRANSMIT_MQ6_IRQ_PEND. 0: No effect, 1: Pending. This bit will be pending for user u when Message Queue 6 is not full. Set one to this bit will clear it.
12	R/W	0x0	RECEPTION_MQ6_IRQ_PEND. 0: No effect, 1: Pending. This bit will be pending for user u when Message Queue 6 has received a new message. Set one to this bit will clear it.
11	R/W	0x1	TRANSMIT_MQ5_IRQ_PEND. 0: No effect, 1: Pending. This bit will be pending for user u when Message Queue 5 is not full. Set one to this bit will clear it.
10	R/W	0x0	RECEPTION_MQ5_IRQ_PEND. 0: No effect, 1: Pending. This bit will be pending for user u when Message Queue 5 has received a new message. Set one to this bit will clear it.
9	R/W	0x1	TRANSMIT_MQ4_IRQ_PEND. 0: No effect, 1: Pending. This bit will be pending for user u when Message Queue 4 is not full. Set one to this bit will clear it.
8	R/W	0x0	RECEPTION_MQ4_IRQ_PEND. 0: No effect,

			1: Pending. This bit will be pending for user u when Message Queue 4 has received a new message. Set one to this bit will clear it.
7	R/W	0x1	TRANSMIT_MQ3_IRQ_PEND. 0: No effect, 1: Pending. This bit will be pending for user u when Message Queue 3 is not full. Set one to this bit will clear it.
6	R/W	0x0	RECEPTION_MQ3_IRQ_PEND. 0: No effect, 1: Pending. This bit will be pending for user u when Message Queue 3 has received a new message. Set one to this bit will clear it.
5	R/W	0x1	TRANSMIT_MQ2_IRQ_PEND. 0: No effect, 1: Pending. This bit will be pending for user u when Message Queue 2 is not full. Set one to this bit will clear it.
4	R/W	0x0	RECEPTION_MQ2_IRQ_PEND. 0: No effect, 1: Pending. This bit will be pending for user u when Message Queue 2 has received a new message. Set one to this bit will clear it.
3	R/W	0x1	TRANSMIT_MQ1_IRQ_PEND. 0: No effect, 1: Pending. This bit will be pending for user u when Message Queue 1 is not full. Set one to this bit will clear it.
2	R/W	0x0	RECEPTION_MQ1_IRQ_PEND. 0: No effect, 1: Pending. This bit will be pending for user u when Message Queue 1 has received a new message. Set one to this bit will clear it.
1	R/W	01	TRANSMIT_MQ0_IRQ_PEND. 0: No effect, 1: Pending. This bit will be pending for user u when Message Queue 0 is not full. Set one to this bit will clear it.
0	R/W	0x0	RECEPTION_MQ0_IRQ_PEND. 0: No effect, 1: Pending. This bit will be pending for user u when Message Queue 0 has received a new message. Set one to this bit will clear it.

Table 3-98 MSGBOX IRQ Status Register u

### 3.11.6.5 MSGBOX FIFO Status Register m

<b>Address :</b> 0x100+N*0x4 (N=0~7)			<b>Name:</b> MSGBOXM_FIFO_STATUS_REG <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:1	/	/	/

0	R	0x0	FIFO_FULL_FLAG. 0: The Message FIFO queue is not full (space is available), 1: The Message FIFO queue is full. This FIFO status register has the status related to the message queue.
---	---	-----	--

**Table 3-99 MSGBOX FIFO Status Register m**

### 3.11.6.6 MSGBOX Message Status Register m

<b>Address</b> : 0x140+N*0x4 (N=0~7)			<b>Name:</b> MSGBOXM_MSG_STATUS_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:3	/	/	/
2:0	R	0x0	MSG_NUM. Number of unread messages in the message queue. Here, limited to four messages per message queue. 000: There is no message in the message FIFO queue. 001: There is 1 message in the message FIFO queue. 010: There are 2 messages in the message FIFO queue. 011: There are 3 messages in the message FIFO queue. 100: There are 4 messages in the message FIFO queue. 101~111:/

**Table 3-100 MSGBOX Message Status Register m**

### 3.11.6.7 MSGBOX Message Queue Register m

<b>Address</b> : 0x180+N*0x4 (N=0~7)			<b>Name:</b> MSGBOXM_MSG_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0x0	The message register stores the next to be read message of the message FIFO queue. Reads remove the message from the FIFO queue.

**Table 3-101 MSGBOX Message Queue Register m**

### 3.11.6.8 MSGBOX Debug Register

<b>Address</b> : 0x1C0			<b>Name:</b> MSGBOX_DEBUG_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:16	/	/	/
15:8	R/W	0x0	FIFO_CTRL. MQ[7:0] Control. In the debug mode, the corresponding FIFO channel will disable and only one register space valid for a message exchange. 0: Normal Mode.

			1: Disable the corresponding FIFO (Clear FIFO).
7:1	/	/	/
0	R/W	0x0	<p>DEBUG_MODE.</p> <p>In the Debug Mode, each user can transmit messages to itself through each Message Queue.</p> <p>0: Normal Mode</p> <p>1: Debug Mode.</p>

**Table 3-102 MSGBOX Debug Register**

## 3.12 Spinlock

### 3.12.1 Overview

Spinlock provides hardware assistance for synchronizing the processes running on multiple processors in the device. The spinlock module implements thirty-two 32-bit spinlocks (or hardware semaphores), which provide an efficient way to perform a lock operation of a device resource using a single read access, thus avoiding the need for a ‘read-modify-write’ bus transfer that not all the programmable cores are capable of.

Spinlocks are present to solve the need for synchronization and mutual exclusion between heterogeneous processors and those not operating under a single, shared operating system. There is no alternative mechanism to accomplish these operations between processors in separate subsystems. However, Spinlocks do not solve all system synchronization issues. They have limited applicability and should be used with care to implement higher level synchronization protocols.

A spinlock is appropriate for mutual exclusion for access to a shared data structure. It should be used only when:

1. The time to hold the lock is predictable and small (for example, a maximum hold time of less than 200 CPU cycles may be acceptable).
2. The locking task cannot be preempted, suspended, or interrupted while holding the lock (this would make the hold time large and unpredictable).
3. The lock is lightly contended, that is the chance of any other process (or processor) trying to acquire the lock while it is held is small.

If the conditions are not met, then a spinlock is not a good candidate. One alternative is to use a spinlock for critical section control (engineered to meet the conditions) to implement a higher level semaphore that can support preemption, notification, timeout or other higher level properties.

The Spinlock includes the following features:

- Spinlock module includes 32 spinlocks
- Two kinds of status of lock register: TAKEN and NOT TAKEN

### 3.12.2 Functionalities Description

- Typical Applications

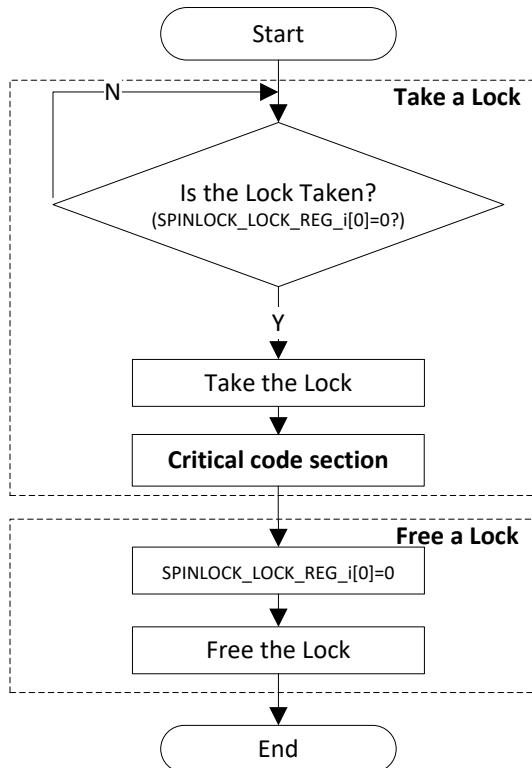


Figure 3-5 Spinlock Typical Application Flow Chart

- Functional Block Diagram

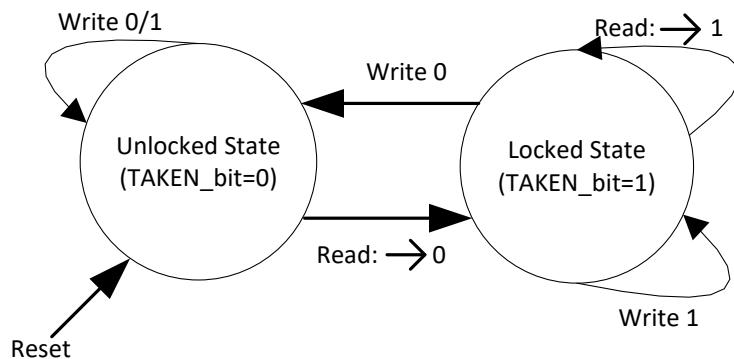


Figure 3-6 Spinlock Lock Register State Diagram

Every lock register has two kinds of states: TAKEN(locked) or NOT TAKEN(Unlocked). Only read-0-access and write-0-access could change lock register' state and the other accesses has no effect. Just 32-bit reads and writes are supported to access all lock registers.

### 3.12.3 Operation Principle

- Spinlock clock gating and software reset

Spinlock clock gating should be open before using it. Setting *AHB1 Module Clock Gating Register* bit[22] 1 could activate Spinlock and then de-asserting it's software reset. Setting *AHB1 Module Software Reset Register* bit[22] 1 could de-assert the software reset of Spinlock. If it is no need to use spinlock, both the gating bit and software reset bit should be set 0.

- Take and free a spinlock

Checking out *SpinLock Register Status* is necessary when a processor would like to take a spinlock. This register stores all 32 lock registers' status: TAKEN or NOT TAKEN(free).

In order to request to take a spinlock, a processor has to do a read-access to the corresponding lock register. If lock register returns 0, the processor takes this spinlock. And if lock register returns 1, the processor must retry.

Writing 0 to a lock register frees the corresponding spinlock. If the lock register is not taken, write-access has no effect. For a taken spinlock, every processor has the privilege to free this spinlock. But it is suggested that the processor which has taken the spinlock free it for strictness.

### 3.12.4 Register List

Module Name	Base Address	
Spinlock	0x40008000	
Register Name	Offset Address	Description
SPINLOCK_SYSTATUS_REG	0x0000	SpinLock System Status Register
SPINLOCK_STATUS_REG	0x0010	SpinLock Status Register
SPINLOCKN_LOCK_REG	0x100+N*0x4	SpinLock Register N (N=0~31)

Table 3-103 Spinlock Register List

### 3.12.5 Register Description

#### 3.12.5.1 SPINLOCK System Status Register

<b>Address :</b> 0x0000			<b>Name:</b> SPINLOCK_SYSTATUS_REG <b>Default:</b> 0x1000_0000
Field	Type	Default	Description
31:30	/	/	/
29:28	R	0x1	LOCKS_NUM. Number of lock registers implemented. 0x1: This instance has 32 lock registers. 0x2: This instance has 64 lock registers.

			0x3: This instance has 128 lock registers. 0x4: This instance has 256 lock registers.
27:9	/	/	/
8	R	0x0	IUO. In-Use flag0, covering lock register0-31. 0: All lock register 0-31 are in the Not Taken state. 1: At least one of the lock register 0-31 is in the Taken state.
7:0	/	/	/

**Table 3-104 SPINLOCK System Status Register**

### 3.12.5.2 SPINLOCK Status Register

<b>Address :</b> 0x10			<b>Name:</b> SPINLOCK_STATUS_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
[i] (i=0~31)	R	0x0	LOCK_REG_STATUS. SpinLock[i] status (i=0~31) 0: The Spinlock is free, 1: The Spinlock is taken.

**Table 3-105 SPINLOCK Status Register**

### 3.12.5.3 SPINLOCK Register N (N=0 to 31)

<b>Address :</b> 0x100+N*0x4 (N=0~31)			<b>Name:</b> SPINLOCKN_LOCK_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:1	/	/	/
0	R/W	0x0	TAKEN. Lock State. Read 0x0: The lock was previously Not Taken (free).The requester is granted the lock. Write 0x0: Set the lock to Not Taken (free). Read 0x1: The lock was previously Taken. The requester is not granted the lock and must retry. Write 0x1: No update to the lock value.

**Table 3-106 SPINLOCK Register N (N=0 to 31)**

## 3.13 Secure System (SS)/Crypto Engine (CE)

### 3.13.1 Overview

The Crypto Engine (CE) is one encrypt/decrypt algorithms accelerator. It is suitable for a variety of applications. Several modes are supported by the CE module. Both CPU mode and DMA method are supported for different applications.

It includes the following features:

- Supports AES, DES, 3DES, SHA-1, MD5, PRNG, CRC32/16, SHA256
- Supports ECB, CBC, CTR modes for AES/DES/3DES
- Supports CTS mode for AES
- Supports 128-bits, 192-bits and 256-bit key size for AES
- Supports 160-bits hardware PRNG with 192-bits seed
- 32-word RX FIFO and 32-word TX FIFO for high speed applications
- CPU mode and DMA mode are supported

The Crypto Engine block diagram is shown as below:

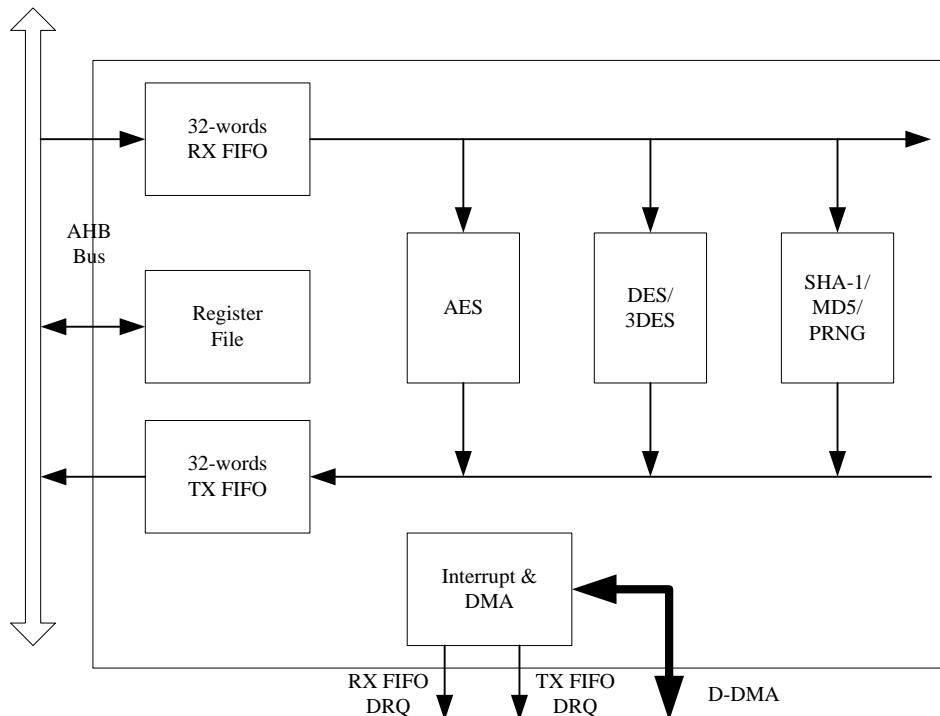


Figure 3-7 Crypto Engine Block Diagram

### 3.13.2 Register List

Module Name	Base Address	
Secure System Controller	0x40004000	
Register Name	Offset Address	Description
CE_CTL	0x00	Control Register
CE_KEY0	0x04	Input Key 0/ PRNG Seed 0
CE_KEY1	0x08	Input Key 1/ PRNG Seed 1
...	...	...
CE_KEY7	0x20	Input Key 7
CE_IV0	0x24	Initialization Vector 0
CE_IV1	0x28	Initialization Vector 1
CE_IV2	0x2C	Initialization Vector 2
CE_IV3	0x30	Initialization Vector 3
CE_CNT0	0x34	Counter 0/Initialization Vector 4
CE_CNT1	0x38	Counter 1/Initialization Vector 5
CE_CNT2	0x3C	Counter 2/Initialization Vector 6
CE_CNT3	0x40	Counter 3/Initialization Vector 7
CE_FCSR	0x44	FIFO Control/ Status Register
CE_ICSR	0x48	Interrupt Control/ Status Register
CE_MD0	0x4C	SHA1/MD5 Message Digest 0/PRNG Data0
CE_MD1	0x50	SHA1/MD5 Message Digest 1/PRNG Data1
CE_MD2	0x54	SHA1/MD5 Message Digest 2/PRNG Data2
CE_MD3	0x58	SHA1/MD5 Message Digest 3/PRNG Data3
CE_MD4	0x5C	SHA1/MD5 Message Digest 4/PRNG Data4
CE_CTS_LEN	0x60	AES-CTS/CRC text length
CE_CRC_POLY	0x70	CRC Poly Register
CRC_RESULT	0x74	CRC Result Register
CE_MD5	0xa0	SHA256 Message Digest Data 5
CE_MD6	0xa4	SHA256 Message Digest Data 6
CE_MD7	0xa8	SHA256 Message Digest Data 7
CE_RXFIFO	0x200	RX FIFO input port
CE_TXFIFO	0x204	TX FIFO output port

Table 3-107 CE/SS Register List

### 3.13.3 Register Description

#### 3.13.3.1 CE Control Register

<b>Address : 0x00</b>			<b>Name: CE_CTL</b> <b>Default: 0x0000_0000</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	R/W	0	CRC xor out value 0: CRC result as CRC out 1: CRC result xor 0xffffffff(CRC32) or 0xffff(CRC16) as CRC out
30	R/W	0	CRC ref out value 0: output word bit sequence not change 1: output word bit sequence reverse
29	R/W	0	CRC ref in value 0: input data bit sequence not change 1: input byte bit sequence reverse
28	R/W	0	CRC init value 0: init value is 0 1: init value is 0xffffffff for CRC32 or 0xffff for CRC16
27:24	R/W	0	KEY_SELECT AES/DES/3DES key select 0: Select input CE_KEYx (Normal Mode) 1: Select SID_RKEYx from SID 2: / 3-10: Select internal Key n (n from 0 to 7) Others: Reserved
18:16	R	x	DIE_ID Die Bonding ID
15	R/W	0	PRNG_MODE/CRC_CONT PRNG generator mode or CRC Package Continue 0: One-shot mode for PRNG or last package for CRC 1: Continue mode for PRNG or non-last package for CRC
14	R/W	0	IV_MODE IV mode for SHA-1/MD5 constants 0: use initial constants defined in FIPS-180 1: use input IV
13:12	R/W	0	CE_OP_MODE CE Operation Mode 00: Electronic Code Book (ECB) mode 01: Cipher Block Chaining (CBC) mode 10: Counter (CTR) mode 11: Cipher Text Stealing (CTS) mode

11:10	R/W	0	CTR_WIDTH Counter Width for CTR Mode 00: 16-bits Counter 01: 32-bits Counter 10: 64-bits Counter 11: 128-bits Counter
9:8	R/W	0	AES_KEY_SIZE Key Size for AES 00: 128-bits 01: 192-bits 10: 256-bits 11: Reserved
7	R/W	0	CE_OP_DIR CE Operation Direction 0: Encryption 1: Decryption
6:4	R/W	0	CE_METHOD CE Method 000: AES 001: DES 010: Triple DES (3DES) 011: SHA-1 100: MD5 101: PRNG 110: SHA256 111: CRC
3	R/W	0	CRC_WIDTH 0: width 16, use 16 bits generate nominal 1: width 32, use 32 bits generate nominal
2	R/W	0	SHA1_SHA256_MD5_CRC_END_BIT SHA-1/MD5/SHA256/CRC Data End bit Write '1' to tell SHA-1/MD5/SHA256/CRC engine that the text data ends. If RX FIFO is not empty, the engine would process the data in RX FIFO. After finishing message digest, this bit is clear to '0' by hardware and message digest can be read out from digest registers. Notes: It is only used for SHA-1/MD5/SHA256/CRC engine.
1	R/W	0	PRNG_START PRNG start bit In PRNG one-shot mode, write '1' to start PRNG. After generating one group random data (5 words), this bit is clear to '0' by hardware.
0	R/W	0	CE_ENABLE CE Enable bit 0: Disable: write '0' to tell engine ends

		1: Enable: after configuration, set this bit to '1' to start engine
--	--	---

**Table 3-108 CE Control Register**

### 3.13.3.2 CE Key Register

<b>Address</b> : 0x04 +4*n			<b>Name:</b> CE_KEY[n] <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R/W	0	CE_KEY Key[n] Input Value (n= 0~7)/ PRNG Seed[n] (n= 0~5)

**Table 3-109 CE Key Register**

### 3.13.3.3 CE IV Register

<b>Address</b> : 0x24 +4*n			<b>Name:</b> CE_IV[n] <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R/W	0	CE_IV_VALUE Initialization Vector (IV[n]) Input Value (n= 0~3)

**Table 3-110 CE IV Register**

### 3.13.3.4 CE Counter Register

<b>Address</b> : 0x34 +4*n			<b>Name:</b> CE_CTR[n] <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R/W	0	CE_CTR_VALUE/CE_IV_VALUE Counter mode preload Counter Input Value (n= 0~3) Initialization Vector (IV[4+n]) Input Value (n= 0~3)

**Table 3-111 CE Counter Register**

### 3.13.3.5 CE FIFO Control/ Status Register

<b>Address</b> : 0x44			<b>Name:</b> CE_FCSR <b>Default:</b> 0x6000_0F0F
Field	Type	Default	Description
31	/	/	/
30	R	0x1	RXFIFO_STATUS RX FIFO Empty 0: No room for new word in RX FIFO

			1: More than one room for new word in RX FIFO ( $\geq 1$ word)
29:24	R	0x20	RXFIFO_EMP_CNT RX FIFO Empty Space Word Counter
23	/	/	/
22	R	0	TXFIFO_STATUS TX FIFO Data Available Flag 0: No available data in TX FIFO 1: More than one data in TX FIFO ( $\geq 1$ word)
21:16	R	0	TXFIFO_AVA_CNT TX FIFO Available Word Counter
15:13	/	/	/
12:8	R/W	0xF	RXFIFO_INT_TRIG_LEVEL RX FIFO Empty Trigger Level Interrupt and DMA request trigger level for RXFIFO normal condition Trigger Level = RXTL + 1 Notes: RX FIFO is used for input the data.
7:5	/	/	/
4:0	R/W	0xF	TXFIFO_INT_TRIG_LEVEL TX FIFO Trigger Level Interrupt and DMA request trigger level for TXFIFO normal condition Trigger Level = TXTL + 1 Notes: TX FIFO is used for output the result data.

**Table 3-112 CE FIFO Control/ Status Register**

### 3.13.3.6 CE Interrupt Control/ Status Register

Address : 0x48			Name: CE_ICSR Default: 0x0000_0000
Field	Type	Default	Description
31:12	/	/	/
11	R/W	0	HASH_CRC_END_INT_PEND 0: No pending 1: HASH/CRC end pending
10	R/W	0	RXFIFO_EMP_PENDING_BIT RX FIFO Empty Pending bit 0: No pending 1: RX FIFO Empty pending Notes: Write '1' to clear or automatic clear if interrupt condition fails.
9	/	/	/
8	R/W	0	TXFIFO_AVA_PENDING_BIT TX FIFO Data Available Pending bit 0: No TX FIFO pending

			1: TX FIFO pending Notes: Write '1' to clear or automatic clear if interrupt condition fails.
7:5	/	/	/
4	R/W	0	DRQ_ENABLE DRQ Enable 0: Disable DRQ (CPU polling mode) 1: Enable DRQ (DMA mode)
3	R/W	0	HASH_CRC_END_INT_ENABLE 0: Disable 1: Enable
2	R/W	0	RXFIFO_EMP_INT_ENABLE RX FIFO Empty Interrupt Enable 0: Disable 1: Enable Notes: If it is set to '1', when the number of empty room is great or equal ( $\geq$ ) the preset threshold, the interrupt is trigger and the correspond flag is set.
1	/	/	/
0	R/W	0	TXFIFO_AVA_INT_ENABLE TX FIFO Data Available Interrupt Enable 0: Disable 1: Enable Notes: If it is set to '1', when available data number is great or equal ( $\geq$ ) the preset threshold, the interrupt is trigger and the correspond flag is set.

Table 3-113 CE Interrupt Control / Status Register

### 3.13.3.7 CE Message Digest Register

<b>Address :</b> 0x4C +4*n			<b>Name:</b> CE_MD[n] <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R	0	CE_MD SHA1/ MD5 Message digest for SHA1/MD5 (n= 0~4)

Table 3-114 CE Message Digest Register

### 3.13.3.8 CE CTS Length Register

<b>Address :</b> 0x60			<b>Name:</b> CE_CTS_LEN <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R/W	0	AES-CTS / CRC text length in byte unit The value of '0' means no data.

**Table 3-115 CE CTS Length Register**

### 3.13.3.9 CE CRC Poly Register

<b>Address</b> : 0x70			<b>Name:</b> CE_CRC_POLY <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0	CRC_POLY CRC poly nominal configuration

**Table 3-116 CE CRC Poly Register**

### 3.13.3.10 CE Result Register

<b>Address</b> : 0x74			<b>Name:</b> CE_RESULT <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0	CRC computation result

**Table 3-117 CE Result Register**

### 3.13.3.11 CE SHA256 Message Digest Register

<b>Address</b> : 0xA0 + 4*(n-5) n = 5~7			<b>Name:</b> CE_MD[n] <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R	0	CE_SHA256_MDn SHA256 Message Digest n (n = 5~7)

**Table 3-118 CE SHA256 Message Digest Register**

### 3.13.3.12 CE RX FIFO Register

<b>Address</b> : 0x200			<b>Name:</b> CE_RX <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	W	0	CE_RX_FIFO 32-bits RX FIFO for Input

**Table 3-119 CE RX FIFO Register**

### 3.13.3.13 CE TX FIFO Register

<b>Address</b> : 0x204	<b>Name:</b> CE_TX <b>Default:</b> 0x0000_0000
------------------------	---

Field	Type	Default	Description
31:0	R	0	CE_TX_FIFO 32-bits TX FIFO for Output

**Table 3-120 CE TX FIFO Register**

# Chapter 4 Peripherals

## 4.1 Universal Asynchronous Receivers/Transmitters (UART)

### 4.1.1 Overview

The UART is used for serial communication with a peripheral, modem (data carrier equipment, DCE) or data set. Data is written from a master (CPU) over the APB bus to the UART and it is converted to serial form and transmitted to the destination device. Serial data is also received by the UART and stored for the master (CPU) to read back.

The UART contains registers to control the character length, baud rate, parity generation/checking, and interrupt generation. Although there is only one interrupt output signal from the UART, there are several prioritized interrupt types that can be responsible for its assertion. Each of the interrupt types can be separately enabled /disabled with the control registers.

The UART has 16450 and 16550 modes of operation, which are compatible with a range of standard software drivers. In 16550 mode, transmit and receive operations are both buffered by FIFOs. In 16450 mode, these FIFOs are disabled.

The UART supports word lengths from five to eight bits, an optional parity bit and 1, 1 ½ or 2 stop bits, and is fully programmable by an AMBA APB CPU interface. A 16-bit programmable baud rate generator and an 8-bit scratch register are included, together with separate transmit and receive FIFOs. Eight modem control lines and a diagnostic loop-back mode are provided.

Interrupts can be generated for a range of TX Buffer/FIFO, RX Buffer/FIFO, Modem Status and Line Status conditions.

For integration in systems where Infrared SIR serial data format is required, the UART can be configured to have a software-programmable IrDA SIR Mode. If this mode is not selected, only the UART (RS232 standard) serial data format is available.

The UART controller includes the following features:

- Compatible with industry-standard 16550 UARTs
- 64-Bytes Transmit and receive data FIFOs
- Support DMA controller interface
- Support Software/ Hardware Flow Control
- Support IrDA 1.0 SIR
- Support RS-485/9-bit mode

The following figures show the UART timing in UART and IrDA modes.

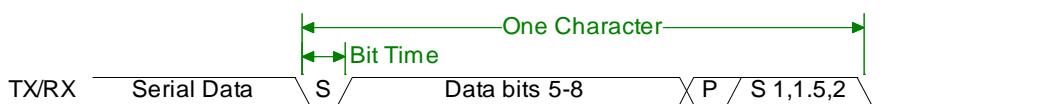


Figure 4-1 UART Serial Data Timing

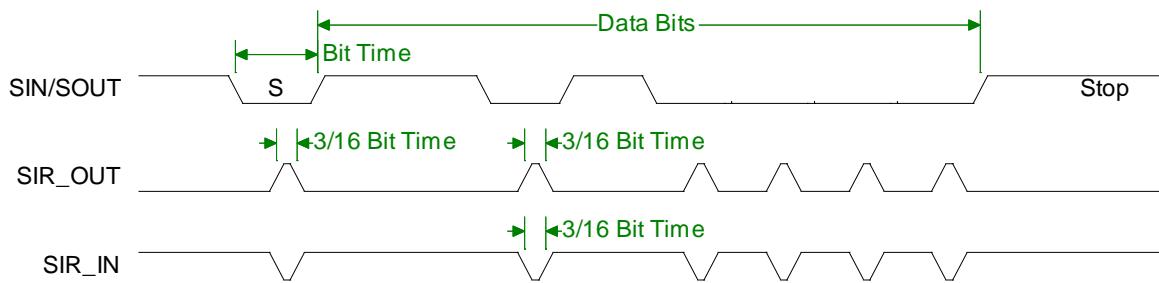


Figure 4-2 UART IrDA Timing

## 4.1.2 Register List

Module Name	Base Address	
UART Controller 0	0x40040c00	
UART Controller 1	0x40041000	
Register Name	Offset Address	Description
UART_RBR	0x00	UART Receive Buffer Register
UART_THR	0x00	UART Transmit Holding Register
UART_DLL	0x00	UART Divisor Latch Low Register
UART_DLH	0x04	UART Divisor Latch High Register
UART_IER	0x04	UART Interrupt Enable Register
UART_IIR	0x08	UART Interrupt Identity Register
UART_FCR	0x08	UART FIFO Control Register
UART_LCR	0x0C	UART Line Control Register
UART_MCR	0x10	UART Modem Control Register
UART_LSR	0x14	UART Line Status Register
UART_MSR	0x18	UART Modem Status Register
UART_SCH	0x1C	UART Scratch Register
UART_USR	0x7C	UART Status Register
UART_TFL	0x80	UART Transmit FIFO Level
UART_RFL	0x84	UART_RFL
UART_HALT	0xA4	UART Halt TX Register
UART_DBG_DLL	0xB0	UART Debug DLL Register
UART_DBG_DLH	0xB4	UART Debug DLH Register
UART_485_CTL	0xC0	UART RS485 Control and Status Register
RS485_ADDR_MATCH	0xC4	UART RS485 Address Match Register
BUS_IDLE_CHK	0xC8	UART RS485 Bus Idle Check Register
TX_DLY	0xCC	UART TX Delay Register
UART_BDCR	0xD4	UART Baudrate Detection Control Register
UART_BDCLR	0xD8	UART Baudrate Detection Counter Low Register

UART_BDCHR	0xDC	UART Baudrate Detection Counter High Register
------------	------	---

**Table 4-1 UART Register List**

## 4.1.3 Register Description

### 4.1.3.1 UART Receiver Buffer Register

<b>Address :</b> 0x0000			<b>Name:</b> UART_RBR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	R	0	<p><b>RBR</b></p> <p>Receiver Buffer Register</p> <p>Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LCR) is set.</p> <p>If in FIFO mode and FIFOs are enabled (FCR [0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an overrun error occurs.</p>

**Table 4-2 UART Receiver Buffer Register**

### 4.1.3.2 UART Transmit Holding Register

<b>Address :</b> 0x0000			<b>Name:</b> UART_THR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	W	0	<p><b>THR</b></p> <p>Transmit Holding Register</p> <p>Data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set.</p> <p>If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, 16 number of characters of data may be written to the THR before the FIFO is full. Any attempt to write data when the FIFO is full results in the write</p>

		data being lost.
--	--	------------------

**Table 4-3 UART Transmit Holding Register****4.1.3.3UART Divisor Latch Low Register**

<b>Address :</b> 0x0000			<b>Name:</b> UART_DLL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	R/W	0	<p>DLL</p> <p>Divisor Latch Low</p> <p>Lower 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor).</p> <p>Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLL is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.</p>

**Table 4-4 UART Divisor Latch Low Register****4.1.3.4UART Divisor Latch High Register**

<b>Address :</b> 0x0004			<b>Name:</b> UART_DLH <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	R/W	0	<p>DLH</p> <p>Divisor Latch High</p> <p>Upper 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (LCR[7]) is set and the UART is not busy (USR[0] is zero). The output baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor).</p> <p>Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest UART clock should be</p>

		allowed to pass before transmitting or receiving data.
--	--	--

**Table 4-5 UART Divisor Latch High Register**

#### 4.1.3.5UART Interrupt Enable Register

<b>Address :</b> 0x0004			<b>Name:</b> UART_IER <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7	R/W	0	<p><b>PTIME</b></p> <p>Programmable THRE Interrupt Mode Enable</p> <p>This is used to enable/disable the generation of THRE Interrupt.</p> <p>0: Disable 1: Enable</p>
6:5	/	/	/
4	R/W	0	<p><b>RS485_INT_EN</b></p> <p>RS485 Interrupt Enable</p> <p>0:Disable 1:Enable</p>
3	R/W	0	<p><b>EDSSI</b></p> <p>Enable Modem Status Interrupt</p> <p>This is used to enable/disable the generation of Modem Status Interrupt.</p> <p>This is the fourth highest priority interrupt.</p> <p>0: Disable 1: Enable</p>
2	R/W	0	<p><b>ELSI</b></p> <p>Enable Receiver Line Status Interrupt</p> <p>This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt.</p> <p>0: Disable 1: Enable</p>
1	R/W	0	<p><b>ETBEI</b></p> <p>Enable Transmit Holding Register Empty Interrupt</p> <p>This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt.</p> <p>0: Disable 1: Enable</p>

0	R/W	0	<b>ERBFI</b>  Enable Received Data Available Interrupt This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. 0: Disable 1: Enable
---	-----	---	--

**Table 4-6 UART Interrupt Enable Register**

#### 4.1.3.6UART Interrupt Identity Register

<b>Address :</b> 0x0008			<b>Name:</b> UART_IIR <b>Default:</b> 0x0000_0001
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:6	R	0	<b>FEFLAG</b>  FIFOs Enable Flag This is used to indicate whether the FIFOs are enabled or disabled. 00: Disable 11: Enable
5:4	/	/	/
3:0	R	0x1	<b>IID</b>  <b>Interrupt ID</b> This indicates the highest priority pending interrupt which can be one of the following types: 0000: modem status 0001: no interrupt pending 0010: THR empty 0011:RS485 Interrupt 0100: received data available 0110: receiver line status 0111: busy detect 1100: character timeout Bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt.

**Table 4-7 UART Interrupt Identity Register**

<b>Interrupt ID</b>	<b>Priority Level</b>	<b>Interrupt Type</b>	<b>Interrupt Source</b>	<b>Interrupt Reset</b>
0001	-	None	None	-

0110	Highest	Receiver line status	Overrun/parity/ framing errors or break interrupt	Reading the line status register
0011	Second	RS485 Interrupt	In RS485 mode, receives address data and match setting address	Writes 1 to addr flag to reset
0100	Third	Received data available	Receiver data available (non-FIFO mode or FIFOs disabled) or RCVR FIFO trigger level reached (FIFO mode and FIFOs enabled)	Reading the receiver buffer register (non-FIFO mode or FIFOs disabled) or the FIFO drops below the trigger level (FIFO mode and FIFOs enabled)
1100	Fourth	Character timeout indication	No characters in or out of the RCVR FIFO during the last 4 character times and there is at least 1character in it during This time	Reading the receiver buffer register
0010	Fifth	Transmit holding register empty	Transmitter holding register empty (Program THRE Mode disabled) or XMIT FIFO at or below threshold (Program THRE Mode enabled)	Reading the IIR register (if source of interrupt); or, writing into THR (FIFOs or THRE Mode not selected or disabled) or XMIT FIFO above threshold (FIFOs and THRE Mode selected and enabled).
0000	Sixth	Modem status	Clear to send or data set ready or ring indicator or data carrier detect. Note that if auto flow control mode is enabled, a change in CTS (that is, DCTS set) does not cause an interrupt.	Reading the Modem status Register
0111	Seventh	Busy detect indication	UART_16550_COMPATIBLE = NO and master has tried to write to the Line Control Register while the UART is busy (USR[0] is set to one).	Reading the UART status register

**Table 4-8 UART Interrupt Identity Priority**

#### 4.1.3.7UART FIFO Control Register

<b>Address :</b> 0x0008			<b>Name:</b> UART_FCR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:6	W	0	RT

			<p><b>RCVR Trigger</b></p> <p>This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation.</p> <p>00: 1 character in the FIFO 01: FIFO ¼ full 10: FIFO ½ full 11: FIFO-2 less than full</p>
5:4	W	0	<p><b>TFT</b></p> <p><b>TX Empty Trigger</b></p> <p>Writes have no effect when THRE_MODE_USER = Disabled. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation.</p> <p>00: FIFO empty 01: 2 characters in the FIFO 10: FIFO ¼ full 11: FIFO ½ full</p>
3	W	0	<p><b>DMAM</b></p> <p><b>DMA Mode</b></p> <p>0: Mode 0</p> <p>In this mode, if PTE is high and TX FIFO is enable, the TX DMA request will send when TFL is less than or equals to FIFO Trigger Level. If PTE is high and TX FIFO is disabled, the TX DMA request will send when THRE is empty. If PTE is low, the TX DMA request will send when the TX FIFO is empty.</p> <p>If dma_pte_rx is high and RX FIFO is enabled, the RX DRQ will send when RFL is equals to or more than FIFO Trigger Level.</p> <p>1: Mode 1</p> <p>In this mode, if TX FIFO is enabled and the PTE is high, the TX DMA request will send when TFL is less than or equals to FIFO Trigger Level. If PTE is low, the TX DMA request will send when TX FIFO is empty and the request stops only when TX FIFO is empty.</p> <p>If RFL is equals to or more than FIFO Trigger Level, the RX DRQ will be set 1,in otherwise, it will be set 0.</p>
2	W	0	<p><b>XFIFOR</b></p> <p><b>XMIT FIFO Reset</b></p> <p>This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request.</p>

			It is 'self-clearing'. It is not necessary to clear this bit.
1	W	0	<p>RFIFOR RCVR FIFO Reset</p> <p>This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request.</p> <p>It is 'self-clearing'. It is not necessary to clear this bit.</p>
0	W	0	<p>FIFOE</p> <p>Enable FIFOs</p> <p>This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset.</p>

**Table 4-9 UART FIFO Control Register**

#### 4.1.3.8UART Line Control Register

<b>Address :</b> 0x000C			<b>Name:</b> UART_LCR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7	R/W	0	<p>DLAB</p> <p>Divisor Latch Access Bit</p> <p>It is writeable only when UART is not busy (USR[0] is zero) and always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers.</p> <p>0: Select RX Buffer Register (RBR) / TX Holding Register(THR) and Interrupt Enable Register (IER)</p> <p>1: Select Divisor Latch LS Register (DLL) and Divisor Latch MS Register (DLM)</p>
6	R/W	0	<p>BC</p> <p>Break Control Bit</p> <p>This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE = Enabled and active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.</p>
5:4	R/W	0	EPS

			<p><b>Even Parity Select</b></p> <p>It is writeable only when UART is not busy (USR[0] is zero) and always writable readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). Setting the LCR[5] is used to reverse the LCR[4].</p> <p>00: Odd Parity 01: Even Parity 1X: Reverse LCR[4]</p> <p>In RS485 mode, it is the 9th bit--address bit.</p> <p>11:9th bit = 1, indicates that this is a address byte 10:9th bit = 0, indicates that this is a data byte</p> <p>Note: When use this function, PEN(LCR[3]) must set to 1.</p>
3	R/W	0	<p><b>PEN</b></p> <p><b>Parity Enable</b></p> <p>It is writeable only when UART is not busy (USR[0] is zero) and always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively.</p> <p>0: parity disabled 1: parity enabled</p>
2	R/W	0	<p><b>STOP</b></p> <p><b>Number of stop bits</b></p> <p>It is writeable only when UART is not busy (USR[0] is zero) and always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data. If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Other/Wise, two stop bits are transmitted. Note that regardless of the number of stop bits selected the receiver checks only the first stop bit.</p> <p>0: 1 stop bit 1: 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit</p>
1:0	R/W	0	<p><b>DLS</b></p> <p><b>Data Length Select</b></p> <p>It is writeable only when UART is not busy (USR[0] is zero) and always readable. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows:</p> <p>00: 5 bits 01: 6 bits 10: 7 bits</p>

		11: 8 bits
--	--	------------

**Table 4-10 UART Line Control Register**

#### 4.1.3.9UART Modem Control Register

<b>Address :</b> 0x0010			<b>Name:</b> UART_MCR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:6	R/W	0	<p>UART_FUNCTION</p> <p>Select IrDA or RS485</p> <p>0:Uart Mode</p> <p>1:IrDA SIR Mode</p> <p>2:RS485 Mode</p> <p>3:Reverse</p>
5	R/W	0	<p>AFCE</p> <p>Auto Flow Control Enable</p> <p>When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled.</p> <p>0: Auto Flow Control Mode disabled</p> <p>1: Auto Flow Control Mode enabled</p>
4	R/W	0	<p>LOOP</p> <p>Loop Back Mode</p> <p>0: Normal Mode</p> <p>1: Loop Back Mode</p> <p>This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE != Enabled or not active, MCR[6] set to zero), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE == Enabled AND active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.</p>
3:2	/	/	/
1	R/W	0	<p>RTS</p> <p>Request to Send</p>

			<p>This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low.</p> <p>0: rts_n de-asserted (logic 1)      1: rts_n asserted (logic 0)</p> <p>Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input.</p>
0	R/W	0	<p>DTR</p> <p>Data Terminal Ready</p> <p>This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n.</p> <p>0: dtr_n de-asserted (logic 1)      1: dtr_n asserted (logic 0)</p> <p>The Data Terminal Ready output is used to inform the modem or data set that the UART is ready to establish communications.</p> <p>Note that in Loopback mode (MCR[4] set to one), the dtr_n output is held inactive high while the value of this location is internally looped back to an input.</p>

**Table 4-11 UART Modem Control Register**

#### 4.1.3.10 UART Line Status Register

Address : 0x0014			Name: UART_LSR Default: 0x0000_0060
Field	Type	Default	Description
31:8	/	/	/
7	R	0	<p>FIFOERR</p> <p>RX Data Error in FIFO</p> <p>When FIFOs are disabled, this bit is always 0. When FIFOs are enabled, this bit is set to 1 when there is at least one PE, FE, or BI in the RX FIFO. It is cleared by a read from the LSR register provided there are no subsequent errors in the FIFO.</p>

6	R	1	<p><b>TEM</b></p> <p>Transmitter Empty</p> <p>If the FIFOs are disabled, this bit is set to "1" whenever the TX Holding Register and the TX Shift Register are empty. If the FIFOs are enabled, this bit is set whenever the TX FIFO and the TX Shift Register are empty. In both cases, this bit is cleared when a byte is written to the TX data channel.</p>
5	R	1	<p><b>THRE</b></p> <p>TX Holding Register Empty</p> <p>If the FIFOs are disabled, this bit is set to "1" whenever the TX Holding Register is empty and ready to accept new data and it is cleared when the CPU writes to the TX Holding Register.</p> <p>If the FIFOs are enabled, this bit is set to "1" whenever the TX FIFO is empty and it is cleared when at least one byte is written to the TX FIFO.</p>
4	R	0	<p><b>BI</b></p> <p>Break Interrupt</p> <p>This is used to indicate the detection of a break sequence on the serial input data.</p> <p>If in UART mode (<code>SIR_MODE == Disabled</code>), it is set whenever the serial input, <code>sir_in</code>, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits.</p> <p>If in infrared mode (<code>SIR_MODE == Enabled</code>), it is set whenever the serial input, <code>sir_in</code>, is continuously pulsed to logic '0' for longer than the sum of start time + data bits + parity + stop bits. A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART.</p> <p>In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read.</p>
3	R	0	<p><b>FE</b></p> <p>Framing Error</p> <p>This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data.</p> <p>In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start</p>

			bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0: no framing error 1:framing error Reading the LSR clears the FE bit.
2	R	0	PE  Parity Error This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0: no parity error 1: parity error Reading the LSR clears the PE bit.
1	R	0	OE  Overrun Error This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is oveR/Written. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. 0: no overrun error 1: overrun error Reading the LSR clears the OE bit.
0	R	0	DR  Data Ready This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0: no data ready 1: data ready This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode.

**Table 4-12 UART Line Status Register**

#### 4.1.3.11 UART Modem Status Register

<b>Address :</b> 0x0018			<b>Name:</b> UART_MSR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7	R	0	<p><b>DCD</b></p> <p>Line State of Data Carrier Detect</p> <p>This is used to indicate the current state of the modem control line dcd_n. This bit is the complement of dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set.</p> <p>0: dcd_n input is de-asserted (logic 1) 1: dcd_n input is asserted (logic 0)</p>
6	R	0	<p><b>RI</b></p> <p>Line State of Ring Indicator</p> <p>This is used to indicate the current state of the modem control line ri_n. This bit is the complement of ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by the modem or data set.</p> <p>0: ri_n input is de-asserted (logic 1) 1: ri_n input is asserted (logic 0)</p>
5	R	0	<p><b>DSR</b></p> <p>Line State of Data Set Ready</p> <p>This is used to indicate the current state of the modem control line dsr_n. This bit is the complement of dsr_n. When the Data Set Ready input (dsr_n) is asserted it is an indication that the modem or data set is ready to establish communications with UART.</p> <p>0: dsr_n input is de-asserted (logic 1) 1: dsr_n input is asserted (logic 0)</p> <p>In Loopback Mode (MCR[4] set to one), DSR is the same as MCR[0] (DTR).</p>
4	R	0	<p><b>CTS</b></p> <p>Line State of Clear To Send</p> <p>This is used to indicate the current state of the modem control line cts_n. This bit is the complement of cts_n. When the Clear to Send input (cts_n) is asserted it is an indication that the modem or data set is ready to exchange data with UART.</p> <p>0: cts_n input is de-asserted (logic 1)</p>

			1: cts_n input is asserted (logic 0) In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS).
3	R	0	<p>DDCD</p> <p>Delta Data Carrier Detect</p> <p>This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read.</p> <p>0: no change on dcd_n since last read of MSR 1: change on dcd_n since last read of MSR</p> <p>Reading the MSR clears the DDCD bit.</p> <p>Note: If the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs (software or other/Wise), then the DDCD bit is set when the reset is removed if the dcd_n signal remains asserted.</p>
2	R	0	<p>TERI</p> <p>Trailing Edge Ring Indicator</p> <p>This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read.</p> <p>0: no change on ri_n since last read of MSR 1: change on ri_n since last read of MSR</p> <p>Reading the MSR clears the TERI bit.</p>
1	R	0	<p>DDSR</p> <p>Delta Data Set Ready</p> <p>This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read.</p> <p>0: no change on dsr_n since last read of MSR 1: change on dsr_n since last read of MSR</p> <p>Reading the MSR clears the DDSR bit. In Loopback Mode (MCR[4] = 1), DDSR reflects changes on MCR[0] (DTR).</p> <p>Note: If the DDSR bit is not set and the dsr_n signal is asserted (low) and a reset occurs (software or other/Wise), then the DDSR bit is set when the reset is removed if the dsr_n signal remains asserted.</p>
0	R	0	<p>DCTS</p> <p>Delta Clear to Send</p> <p>This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read.</p> <p>0: no change on ctsdsr_n since last read of MSR 1: change on ctsdsr_n since last read of MSR</p> <p>Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] = 1), DCTS reflects changes on MCR[1] (RTS).</p>

		Note: If the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or other/Wise), then the DCTS bit is set when the reset is removed if the cts_n signal remains asserted.
--	--	---

**Table 4-13 UART Modem Status Register****4.1.3.12 UART Scratch Register**

<b>Address :</b> 0x001C			<b>Name:</b> UART_SCH <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	R/W	0	<p><b>SCRATCH_REG</b></p> <p>Scratch Register</p> <p>This register is for programmers to use as a temporary storage space. It has no defined purpose in the UART.</p>

**Table 4-14 UART Scratch Register****4.1.3.13 UART Status Register**

<b>Address :</b> 0x007C			<b>Name:</b> UART_USR <b>Default:</b> 0x0000_0006
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:5	/	/	/
4	R	0	<p><b>RFF</b></p> <p>Receive FIFO Full</p> <p>This is used to indicate that the receive FIFO is completely full.</p> <p>0: Receive FIFO not full</p> <p>1: Receive FIFO Full</p> <p>This bit is cleared when the RX FIFO is no longer full.</p>
3	R	0	<p><b>RFNE</b></p> <p>Receive FIFO Not Empty</p> <p>This is used to indicate that the receive FIFO contains one or more entries.</p> <p>0: Receive FIFO is empty</p> <p>1: Receive FIFO is not empty</p> <p>This bit is cleared when the RX FIFO is empty.</p>
2	R	1	<p><b>TFE</b></p> <p>Transmit FIFO Empty</p> <p>This is used to indicate that the transmit FIFO is completely empty.</p>

			0: Transmit FIFO is not empty 1: Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty.
1	R	1	TFNF  Transmit FIFO Not Full This is used to indicate that the transmit FIFO is not full. 0: Transmit FIFO is full 1: Transmit FIFO is not full This bit is cleared when the TX FIFO is full.
0	R	0	BUSY  UART Busy Bit 0: Idle or inactive 1: Busy

**Table 4-15 UART Status Register**

#### 4.1.3.14 UART Transmit FIFO Level Register

<b>Address :</b> 0x0080			<b>Name:</b> UART_TFL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:7	/	/	/
6:0	R	0	TFL  Transmit FIFO Level This indicates the number of data entries in the transmit FIFO.

**Table 4-16 UART Transmit FIFO Level Register**

#### 4.1.3.15 UART Receive FIFO Level Register

<b>Address :</b> 0x0084			<b>Name:</b> UART_RFL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:7	/	/	/
6:0	R	0	RFL  Receive FIFO Level This indicates the number of data entries in the receive FIFO.

**Table 4-17 UART Receive FIFO Level Register**

#### 4.1.3.16 UART Halt TX Register

<b>Address :</b> 0x00A4			<b>Name:</b> UART_HALT <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7	R/W	0	<p>PTE</p> <p>TX_REQ 的发送。</p> <p>对于 DMA1 模式 0(FIFO 开, FCR[3]=1),如果 PTE 信号为高, 当 TFL 小于等于 TRIG, 发送 DMA 请求。当 PTE 信号为低, 当 FIFO 为空时发送 DMA 请求。DMA 请求只会在 FIFO 满的时候掉下来。</p> <p>对于 DMA0 模式(FIFO 关或 FCR[3]=0),如果 PTE 信号为高且 FIFO 开, 当 TFL 小于等于 TRIG, 发送 DMA 请求,如果条件不满足则 DRQ 掉下来。如果 PTE 信号为高且 FIFO 关, THRE 空时, 发送 DMA 请求,如果条件不满足则 DRQ 掉下来。当 PTE 信号为低, 当 FIFO 为空时发送 DMA 请求。</p>
6	R/W	0	<p>DMA_PTE_RX</p> <p>RX_DRQ 的发送。</p> <p>对于 DMA1 模式,当 RFL 大于等于 TRIG 或者接收超时发送 DRQ, 否则 DRQ 为低。</p> <p>对于 DMA0 模式,当 DMA_PTE_RX = 1 且 FIFO 开时, 当 RFL 大于等于 TRIG 时发送 REQ, 否则不发送。否则只要存在 RX 有效数据就发送 DRQ。</p>
5	R/W	0	<p>SIR_RX_INVERT</p> <p>SIR Receiver Pulse Polarity Invert</p> <p>0: Not invert receiver signal</p> <p>1: Invert receiver signal</p>
4	R/W	0	<p>SIR_TX_INVERT</p> <p>SIR Transmit Pulse Polarity Invert</p> <p>0: Not invert transmit pulse</p> <p>1: Invert transmit pulse</p>
3	/	/	/
2	R/W AC	0	<p>CHANGE_UPDATE</p> <p>After the user using HALT[1] to change the baud rate or LCR configuration, write 1 to update the configuration and waiting this bit self-clear to 0 to finish update process. Write 0 to this bit has no effect.</p> <p>1: Update trigger, Self clear to 0 when finish update.</p>

1	R/W	0	<b>CHCFG_AT_BUSY</b>  This is an enable bit for the user to change LCR register configuration (except for the DLAB bit) and baud rate register (DLH and DLL) when the UART is busy (USB[0] is 1). 1: Enable change when busy
0	R/W	0	<b>HALT_TX</b>  Halt TX  This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 0 : Halt TX disabled 1 : Halt TX enabled  Note: If FIFOs are not enabled, the setting of the halt TX register has no effect on operation.

**Table 4-18 UART Halt TX Register**

#### 4.1.3.17 UART TX Delay

<b>Address :</b> 0x00CC			<b>Name:</b> UART_RXDLY <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	R/W	0	<b>DLY</b>  The delay time between the last stop bit and the next start bit. The unit is Tclk. It is use to control the space between two bytes in TX.

**Table 4-19 UART TX Delay**

#### 4.1.3.18 UART Baudrate Detection Control Register

<b>Address :</b> 0x0D4			<b>Name:</b> UART_BDC <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:2	/	/	/
1	R/W	0	<b>Auto Baudrate Detect Enable</b>  1: enable 0: disable  <i>Note: Set this bit to 1 to start the UART Baudrate Detection. The detection circuit will detect the start bit and count the cycles of PCLK in one bit</i>

			<i>duration. The number of the cycles will be written in the PCLK_NUM field. Set this bit to 0 to stop the UART Baudrate Detection.</i>
0	R/W	0	<p>Mode</p> <p>1: Normal mode 0: Compare mode</p> <p><i>Note:</i></p> <p><i>In normal mode, when the detection circuit is enabled, the number of the PCLK cycles will be latched in each one bit duration and be written into the PCLK_NUM field.</i></p> <p><i>In compare mode, the number of the PCLK cycles will be latched in each one bit duration, but only the value which equals to the value latched in last time will be written into the PCLK_NUM field.</i></p>

**Table 4-20 UART Baudrate Detection Control Register****4.1.3.19 UART Baudrate Detection Counter Low Register**

<b>Address :</b> 0x0D8			<b>Name:</b> UART_BDCL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	R/W	0	<p>PCLK_NUM_LOW</p> <p>The low 8 bits of the number of the cycles counted in one bit duration.</p>

**Table 4-21 UART Baudrate Detection Counter Low Register****4.1.3.20 UART Baudrate Detection Counter High Register**

<b>Address :</b> 0x0DC			<b>Name:</b> UART_ABCR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	R/W	0	<p>PCLK_NUM_HIGH</p> <p>The high 8 bits of the number of the cycles counted in one bit duration.</p>

**Table 4-22 UART Baudrate Detection Counter High Register**

## 4.2 Serial Peripheral Interface (SPI)

### 4.2.1 Overview

The SPI supports 4 different formats for data transfer. Software can select one of the four modes in which the SPI works by setting the bit1 (Polarity) and bit0 (Phase) of SPI Transfer Control Register. The SPI controller master uses the SPI\_SCLK signal to transfer data in and out of the shift register. Data is clocked using any one of four programmable clock phase and polarity combinations.

During Phase 0, Polarity 0 and Phase 1, Polarity 1 operations, output data changes on the falling clock edge and input data is shifted in on the rising edge.

During Phase 1, Polarity 0 and Phase 0, Polarity 1 operations, output data changes on the rising edges of the clock and is shifted in on falling edges.

The POL defines the signal polarity when SPI\_SCLK is in idle state. The SPI\_SCLK is high level when POL is '1' and it is low level when POL is '0'. The PHA decides whether the leading edge of SPI\_SCLK is used for setup or sample data. The leading edge is used for setup data when PHA is '1' and for sample data when PHA is '0'. The four kinds of modes are listed Table 1-3:

SPI Mode	POL	PHA	Leading Edge	Trailing Edge
0	0	0	Rising, Sample	Falling, Setup
1	0	1	Rising, Setup	Falling, Sample
2	1	0	Falling, Sample	Rising, Setup
3	1	1	Falling, Setup	Rising, Sample

Table 4-23 SPI Modes with Clock Polarity and Phase

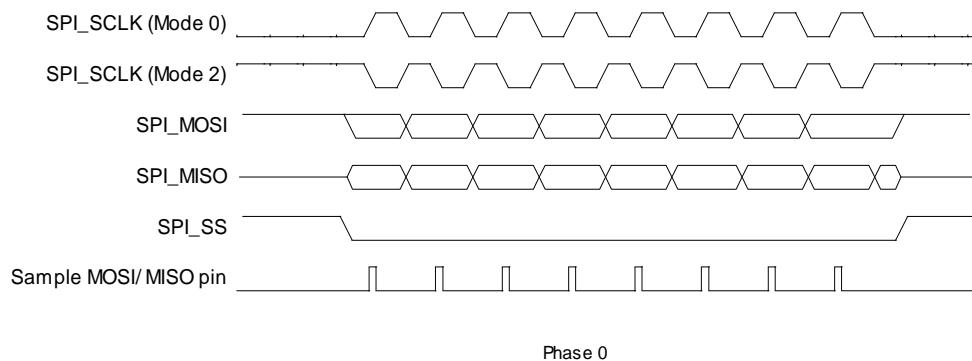
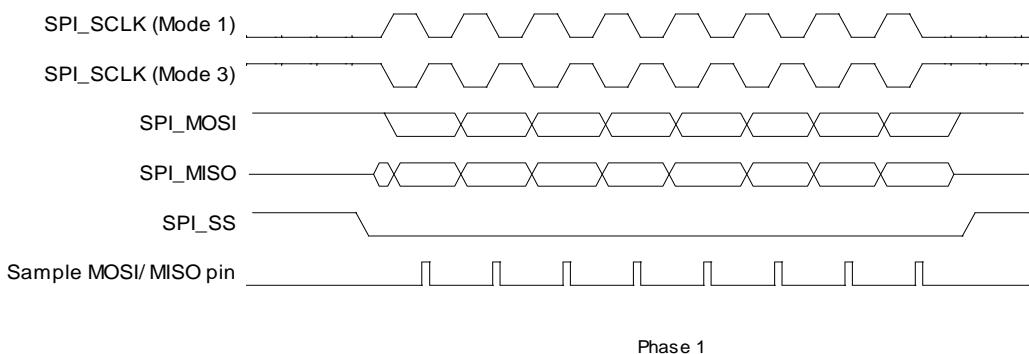


Figure 4-3 SPI Phase 0 Transfer Format

**Figure 4-4 SPI Phase 1 Transfer Format**

#### 4.2.2 Register List

<b>Module Name</b>	<b>Base Address</b>	
SPI Controller 0	0x40005000	
SPI Controller 1	0x40006000	
<b>Register Name</b>	<b>Offset Address</b>	<b>Description</b>
SPI_VER	0x0000	SPI Version Number Register
SPI_CTRL	0x0004	SPI Global Control Register
SPI_TCTRL	0x0008	SPI Transfer Control register
/	0x000c	reserved
SPI_IER	0x0010	SPI Interrupt Control register
SPI_STA	0x0014	SPI Interrupt Status register
SPI_FCTL	0x0018	SPI FIFO Control register
SPI_FST	0x001c	SPI FIFO Status register
SPI_WAIT	0x0020	SPI Wait Clock Counter register
SPI_CCTR	0x0024	SPI Clock Rate Control register
/	0x0028	reserved
/	0x002c	reserved
SPI_BC	0x0030	SPI Burst Counter register
SPI_TC	0x0034	SPI Transmit Counter Register
SPI_BCC	0x0038	SPI Burst Control register
SPI_NDMA_MODE_CTRL	0x0088	SPI Normal DMA Mode Control Register
SPI_TXD	0x0200	SPI TX Data register
SPI_RXD	0x0300	SPI RX Data register

**Table 4-24 SPI Register List**

## 4.2.3 Register Description

### 4.2.3.1 SPI Version Number Register

<b>Address :</b> 0x0000			<b>Name:</b> SPI_VER <b>Default:</b> 0x0009_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R	0x90000	SPI_VER

Table 4-25 SPI Version Number Register

### 4.2.3.2 SPI Global Control Register

<b>Address :</b> 0x0004			<b>Name:</b> SPI_CTRL <b>Default:</b> 0x0000_0080
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	R/WAC	0	<p><b>SPI_RST</b> SPI Software Reset Write 1 to this bit will clear the SPI controller, and auto clear to 0 when reset operation completes Write 0 has no effect.</p>
30:8	/	/	/
7	R/W	1	<p><b>TP_EN</b> Transmit Pause Enable In master mode, it is used to control transmit state machine to stop smart burst sending when RX FIFO is full. 1: stop transmit data when RXFIFO full 0: normal operation Note: cannot be written when XCH=1</p>
6:2	/	/	/
1	R/W	0	<p><b>SPI_MODE</b> Master Slave selection 0: slave mode 1: master mode Note: cannot be written when XCH=1</p>
0	R/W	0	<p><b>SPI_EN</b> SPI Module Enable Control 0: SPI Controller Disable</p>

			1: SPI Controller Enable
--	--	--	--------------------------

**Table 4-26 SPI Global Control Register**

#### 4.2.3.3 SPI Transfer Control Register

Address : 0x0008			Name: SPI_TCTRL Default: 0x0000_0087
Field	Type	Default	Description
31	R/W	0	<p>XCH</p> <p>Exchange Burst</p> <p>In master mode it is used to start SPI burst.</p> <p>0: Idle</p> <p>1: Initiates exchange</p> <p>Write 1 to this bit will start the SPI burst and this bit will auto clear after finishing the bursts transfer specified by BC. Write 1 to SPI_RST will also clear this bit.</p> <p>Write 0 to this bit has no effect.</p>
30:15	/	/	/
14	R/W	0	<p>SDDM</p> <p>Sending Data Delay Mode</p> <p>0: normal sending</p> <p>1: delay sending</p> <p>Set this bit to 1 to make the data that should be send with a delay of half cycle of SPI_CLK in dual io mode for SPI mode 0</p>
13	R/W	0	<p>SDM</p> <p>Master Sample Data Mode</p> <p>1: normal sample mode</p> <p>0: delay sample mode</p> <p>In normal sample mode, SPI master samples the data at the correct edge for each SPI mode;</p> <p>In delay sample mode, SPI master samples data at the edge that is half cycle delayed by the correct edge defined in respective SPI mode.</p>
12	R/W	0	<p>FBS</p> <p>First Transmit Bit Select</p> <p>0: MSB first</p> <p>1: LSB first</p>
11	R/W	0	SDC

			<p>Master Sample Data Control Set this bit to 1 to make the internal read sample point with</p>
10	R/W	0x0	<p>RPSM  Rapids mode select Select Rapids mode for high speed write. 0: normal write mode 1: rapids write mode Note: Can't be written when XCH=1.</p>
9	R/W	0x0	<p>DDB  Dummy Burst Type 0: The bit value of dummy SPI burst is zero 1: The bit value of dummy SPI burst is one Note: Can't be written when XCH=1.</p>
8	R/W	0x0	<p>DHB  Discard Hash Burst In master mode it controls whether discarding unused SPI bursts 0: Receiving all SPI bursts in BC period 1: Discard unused SPI bursts, only fetching the SPI bursts during dummy burst period. The bursts number is specified by TC. Note: Can't be written when XCH=1.</p>
7	R/W	0x1	<p>SS_LEVEL  When control SS signal manually (SPI_CTRL_REG.SS_CTRL==1), set this bit to '1' or '0' to control the level of SS signal. 0: set SS to low 1: set SS to high Note: Can't be written when XCH=1.</p>
6	R/W	0x0	<p>SS_OWNER  SS Output Owner Select Usually, controller sends SS signal automatically with data together. When this bit is set to 1, software must manually write SPI_CTL_REG.SS_LEVEL to 1 or 0 to control the level of SS signal. 0: SPI controller 1: Software Note: Can't be written when XCH=1.</p>
5:4	R/W	0x0	<p>SS_SEL  SPI Chip Select</p>

			Select one of four external SPI Master/Slave Devices 00: SPI_SS0 will be asserted 01: SPI_SS1 will be asserted 10: SPI_SS2 will be asserted 11: SPI_SS3 will be asserted Note: Can't be written when XCH=1.
3	R/W	0x0	SSCTL  In master mode, this bit selects the output wave form for the SPI_SSx signal. Only valid when SS_OWNER = 0. 0: SPI_SSx remains asserted between SPI bursts 1: Negate SPI_SSx between SPI bursts Note: Can't be written when XCH=1.
2	R/W	0x1	SPOL  SPI Chip Select Signal Polarity Control 0: Active high polarity (0 = Idle) 1: Active low polarity (1 = Idle) Note: Can't be written when XCH=1.
1	R/W	0x1	CPOL  SPI Clock Polarity Control 0: Active high polarity (0 = Idle) 1: Active low polarity (1 = Idle) Note: Can't be written when XCH=1.
0	R/W	0x1	CPHA  SPI Clock/Data Phase Control 0: Phase 0 (Leading edge for sample data) 1: Phase 1 (Leading edge for setup data) Note: Can't be written when XCH=1.

**Table 4-27 SPI Transfer Control Register**

#### 4.2.3.4 SPI Interrupt Control Register

<b>Address : 0x0010</b>			<b>Name:</b> SPI_IER <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:14	R	0x0	Reserved.
13	R/W	0x0	SS_INT_EN  SSI Interrupt Enable

			Chip Select Signal (SSx) from valid state to invalid state 0: Disable 1: Enable
12	R/W	0x0	TC_INT_EN  Transfer Completed Interrupt Enable 0: Disable 1: Enable
11	R/W	0x0	TF_UDR_INT_EN  TXFIFO under run Interrupt Enable 0: Disable 1: Enable
10	R/W	0x0	TF_OVF_INT_EN  TX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
9	R/W	0x0	RF_UDR_INT_EN  RXFIFO under run Interrupt Enable 0: Disable 1: Enable
8	R/W	0x0	RF_OVF_INT_EN  RX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
7	R	0x0	Reserved.
6	R/W	0x0	TF_FUL_INT_EN  TX FIFO Full Interrupt Enable 0: Disable 1: Enable
5	R/W	0x0	TX_EMP_INT_EN  TX FIFO Empty Interrupt Enable 0: Disable 1: Enable
4	R/W	0x0	TX_ERQ_INT_EN  TX FIFO Empty Request Interrupt Enable 0: Disable

			1: Enable
3	R	0x0	Reserved
2	R/W	0x0	<p>RF_FUL_INT_EN</p> <p>RX FIFO Full Interrupt Enable</p> <p>0: Disable 1: Enable</p>
1	R/W	0x0	<p>RX_EMP_INT_EN</p> <p>RX FIFO Empty Interrupt Enable</p> <p>0: Disable 1: Enable</p>
0	R/W	0x0	<p>RF_RDY_INT_EN</p> <p>RX FIFO Ready Request Interrupt Enable</p> <p>0: Disable 1: Enable</p>

**Table 4-28 SPI Interrupt Control Register**

#### 4.2.3.5 SPI Interrupt Status Register

<b>Address :</b> 0x0014			<b>Name:</b> SPI_STA <b>Default:</b> 0x0000_0032
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:14	/	0x0	/
13	R/W1C	0x0	<p>SSI</p> <p>SS Invalid Interrupt</p> <p>When SSI is 1, it indicates that SS has changed from valid state to invalid state. Writing 1 to this bit clears it.</p>
12	R/W1C	0x0	<p>TC</p> <p>Transfer Completed</p> <p>In master mode, it indicates that all bursts specified by BC has been exchanged. In other condition, When set, this bit indicates that all the data in TXFIFO has been loaded in the Shift register, and the Shift register has shifted out all the bits. Writing 1 to this bit clears it.</p> <p>0: Busy 1: Transfer Completed</p>
11	R/W1C	0x0	<p>TF_UDF</p> <p>TXFIFO under run</p>

			This bit is set when if the TXFIFO is under run. Writing 1 to this bit clears it. 0: TXFIFO is not under run 1: TXFIFO is under run
10	R/W1C	0x0	TF_OVF  TXFIFO Overflow This bit is set when if the TXFIFO is overflow. Writing 1 to this bit clears it. 0: TXFIFO is not overflow 1: TXFIFO is overflowed
9	R/W1C	0x0	RX_UDF  RXFIFO Under run When set, this bit indicates that RXFIFO has under run. Writing 1 to this bit clears it.
8	R/W1C	0x0	RX_OVF  RXFIFO Overflow When set, this bit indicates that RXFIFO has overflowed. Writing 1 to this bit clears it. 0: RXFIFO is available. 1: RXFIFO has overflowed.
7	/	/	/
6	R/W1C	0x0	TX_FULL  TXFIFO Full This bit is set when if the TXFIFO is full. Writing 1 to this bit clears it. 0: TXFIFO is not Full 1: TXFIFO is Full
5	R/W1C	0x1	TX_EMP  TXFIFO Empty This bit is set if the TXFIFO is empty. Writing 1 to this bit clears it. 0: TXFIFO contains one or more words. 1: TXFIFO is empty
4	R/W1C	0x1	TX_READY  TXFIFO Ready 0: TX_WL > TX_TRIG_LEVEL 1: TX_WL <= TX_TRIG_LEVEL This bit is set any time if TX_WL <= TX_TRIG_LEVEL. Writing "1" to this bit clears it. Where TX_WL is the water level of RXFIFO
3	/	/	reserved

2	R/W1C	0x0	<b>RX_FULL</b>  RXFIFO Full This bit is set when the RXFIFO is full. Writing 1 to this bit clears it. 0: Not Full 1: Full
1	R/W1C	0x1	<b>RX_EMP</b>  RXFIFO Empty This bit is set when the RXFIFO is empty. Writing 1 to this bit clears it. 0: Not empty 1: empty
0	R/W1C	0x0	<b>RX_RDY</b>  RXFIFO Ready 0: RX_WL < RX_TRIG_LEVEL 1: RX_WL >= RX_TRIG_LEVEL This bit is set any time if RX_WL >= RX_TRIG_LEVEL. Writing "1" to this bit clears it. Where RX_WL is the water level of RXFIFO.

**Table 4-29 SPI Interrupt Status Register**

#### 4.2.3.6 SPI FIFO Control Register

Address : 0x0018			Name: SPI_FCTL Default: 0x0040_0001
Field	Type	Default	Description
31	R/WAC	0x0	<b>TX_FIFO_RST</b>  TX FIFO Reset Write '1' to this bit will reset the control portion of the TX FIFO and auto clear to '0' when completing reset operation, write to '0' has no effect.
30	R/W	0x0	<b>TF_TEST_ENB</b>  TX Test Mode Enable 0: disable 1: enable Note: In normal mode, TX FIFO can only be read by SPI controller, write '1' to this bit will switch TX FIFO read and write function to AHB bus. This bit is used to test the TX FIFO, don't set in normal operation and don't set RF_TEST and TF_TEST at the same time.
29:25	/	/	/
24	R/W	0x0	<b>TF_DRQ_EN</b>

			TX FIFO DMA Request Enable 0: Disable 1: Enable
23:16	R/W	0x40	TX_TRIG_LEVEL  TX FIFO Empty Request Trigger Level
15	R/WAC	0x0	RF_RST  RXFIFO Reset Write '1' to this bit will reset the control portion of the receiver FIFO, and auto clear to '0' when completing reset operation, write '0' to this bit has no effect.
14	R/W	0x0	RF_TEST  RX Test Mode Enable 0: Disable 1: Enable Note: In normal mode, RX FIFO can only be written by SPI controller, write '1' to this bit will switch RX FIFO read and write function to AHB bus. This bit is used to test the RX FIFO, don't set in normal operation and don't set RF_TEST and TF_TEST at the same time.
13:9	/	/	/
8	R/W	0x0	RF_DRQ_EN  RX FIFO DMA Request Enable 0: Disable 1: Enable
7:0	R/W	0x1	RX_TRIG_LEVEL  RX FIFO Ready Request Trigger Level

**Table 4-30 SPI FIFO Control Register**

#### 4.2.3.7 SPI FIFO Status Register

<b>Address :</b> 0x001C			<b>Name:</b> SPI_FST <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	R	0x0	TB_WR  TX FIFO Write Buffer Write Enable
30:28	R	0x0	TB_CNT

			TX FIFO Write Buffer Counter These bits indicate the number of words in TX FIFO Write Buffer
27:24	R	0x0	Reserved
23:16	R	0x0	TF_CNT  TX FIFO Counter  These bits indicate the number of words in TX FIFO 0: 0 byte in TX FIFO 1: 1 byte in TX FIFO ... 64: 64 bytes in TX FIFO other: Reserved
15	R	0x0	RB_WR  RX FIFO Read Buffer Write Enable
14:12	R	0x0	RB_CNT  RX FIFO Read Buffer Counter These bits indicate the number of words in RX FIFO Read Buffer
11:8	R	0x0	Reserved
7:0	R	0x0	RF_CNT  RX FIFO Counter These bits indicate the number of words in RX FIFO 0: 0 byte in RX FIFO 1: 1 byte in RX FIFO ... 64: 64 bytes in RX FIFO other: Reserved

**Table 4-31 SPI FIFO Status Register**

#### 4.2.3.8 SPI Wait Clock Register

<b>Address :</b> 0x0020			<b>Name:</b> SPI_WAIT <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:20	/	/	/
19:16	R/W	0x0	SWC  Dual mode direction switch wait clock counter (for master mode only).

			<p>0: No wait states inserted n: n SPI_SCLK wait states inserted Note: These bits control the number of wait states to be inserted before start dual data transfer in dual SPI mode. The SPI module counts SPI_SCLK by SWC for delaying next word data transfer. Note: Can't be written when XCH=1.</p>
15:0	R/W	0x0	<p>WCC  Wait Clock Counter (In Master mode) These bits control the number of wait states to be inserted in data transfers. The SPI module counts SPI_SCLK by WCC for delaying next word data transfer. 0: No wait states inserted N: N SPI_SCLK wait states inserted</p>

**Table 4-32 SPI Wait Clock Register**

#### 4.2.3.9 SPI Clock Control Register

Address : 0x0024			Name: SPI_CCTR Default: 0x0000_0002
Field	Type	Default	Description
31:13	/	/	/
12	R/W	0x0	<p>DRS  Divide Rate Select (Master Mode Only) 0: Select Clock Divide Rate 1 1: Select Clock Divide Rate 2</p>
11:8	R/W	0x0	<p>CDR1  Clock Divide Rate 1 (Master Mode Only) The SPI_SCLK is determined according to the following equation: SPI_CLK = SRC_CLK / 2^n.</p>
7:0	R/W	0x2	<p>CDR2  Clock Divide Rate 2 (Master Mode Only) The SPI_SCLK is determined according to the following equation: SPI_CLK = SRC_CLK / (2^(n + 1)).</p>

**Table 4-33 SPI Clock Control Register**

#### 4.2.3.10 SPI Master Burst Counter Register

Address : 0x0030	Name: SPI_BC
------------------	--------------

			<b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:24	/	/	/
23:0	R/W	0x0	<p>MBC</p> <p>Master Burst Counter</p> <p>In master mode, this field specifies the total burst number .</p> <p>0: 0 burst</p> <p>1: 1 burst</p> <p>...</p> <p>N: N bursts</p>

**Table 4-34 SPI Burst Counter Register**

#### 4.2.3.11 SPI Transmit Counter Register

<b>Address :</b> 0x0034			<b>Name:</b> SPI_TC <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:24	/	/	/
23:0	R/W	0x0	<p>MWTC</p> <p>Master Write Transmit Counter</p> <p>In master mode, this field specifies the burst number that should be sent to TXFIFO before automatically sending dummy burst . For saving bus bandwidth, the dummy burst (all zero bits or all one bits) is sent by SPI Controller automatically.</p> <p>0: 0 burst</p> <p>1: 1 burst</p> <p>...</p> <p>N: N bursts</p>

**Table 4-35 SPI Transmit Counter Register**

#### 4.2.3.12 SPI Master Burst Control Register

<b>Address :</b> 0x0038			<b>Name:</b> SPI_BCC <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:29	R	0x0	Reserved
28	R/W	0x0	<p>DRM</p> <p>Master Dual Mode RX Enable</p> <p>0: RX use single-bit mode</p>

			1: RX use dual mode Note: Can't be written when XCH=1.
27:24	R/W	0x0	<p>DBC</p> <p>Master Dummy Burst Counter</p> <p>In master mode, this field specifies the burst number that should be sent before receive in dual SPI mode. The data is don't care by the device.</p> <p>0: 0 burst 1: 1 burst ... N: N bursts</p> <p>Note: Can't be written when XCH=1.</p>
23:0	R/W	0x0	<p>STC</p> <p>Master Single Mode Transmit Counter</p> <p>In master mode, this field specifies the burst number that should be sent in single mode before automatically sending dummy burst. This is the first transmit counter in all bursts.</p> <p>0: 0 burst 1: 1 burst ... N: N bursts</p> <p>Note: Can't be written when XCH=1.</p>

Table 4-36 SPI Master Burst Control Register

#### 4.2.3.13 SPI Normal DMA Mode Control

<b>Address :</b> 0x0088			<b>Name:</b> SPI_NDMA_MODE_CTRL <b>Default:</b> 0x0000_00a5
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	R/W	0xA5	<p>NDMA_MODE_CTL</p> <p>0xEA: NDMA handshake mode</p> <p>Note: NDMA wait mode doesn't care this value. 0xA5 can be used in handshake mode, but 0xEA is better.</p>

Table 4-37 SPI Normal DMA Mode Register

#### 4.2.3.14 SPI TX Data Register

<b>Address :</b> 0x0200	<b>Name:</b> SPI_TXD <b>Default:</b> 0x0000_0000
-------------------------	---

Field	Type	Default	Description
31:0	R/W	0x0	<p>TDATA Transmit Data</p> <p>This register can be accessed in byte, half-word or word unit by AHB. In byte accessing method, if there are rooms in TXFIFO, one burst data is written to TXFIFO and the depth is increased by 1. In half-word accessing method, two SPI burst data are written and the TXFIFO depth is increase by 2. In word accessing method, four SPI burst data are written and the TXFIFO depth is increased by 4.</p> <p>Note: This address is writing-only if TF_TEST is '0', and if TF_TEST is set to '1', this address is readable and writable to test the TX FIFO through the AHB bus.</p>

**Table 4-38 SPI TX Data Register**

#### 4.2.3.15 SPI RX Data Register

<b>Address : 0x0300</b>			<b>Name:</b> SPI_RXD <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R	0x0	<p>RDATA Receive Data</p> <p>This register can be accessed in byte, half-word or word unit by AHB. In byte accessing method, if there are data in RXFIFO, the top word is returned and the RXFIFO depth is decreased by 1. In half-word accessing method, two SPI bursts are returned and the RXFIFO depth is decrease by 2. In word accessing method, the four SPI bursts are returned and the RXFIFO depth is decreased by 4.</p> <p>Note: This address is read-only if RF_TEST is '0', and if RF_TEST is set to '1', this address is readable and writable to test the RX FIFO through the AHB bus.</p>

**Table 4-39 SPI RX Data Register**

### 4.3 Two Wire Interface Controller (TWI)

#### 4.3.1 Overview

This TWI Controller is designed to be used as an interface between CPU host and the serial TWI bus. It can support all the standard TWI transfer, including Slave and Master. The communication to the TWI bus is carried out on a byte-wise basis using interrupt or polled handshaking. This TWI Controller can be operated in standard mode (100K bps) or fast-mode, supporting data rate up to 400K bps. Multiple Masters and 10-bit addressing Mode are supported for this specified application. General Call Addressing is also supported in Slave mode.

The TWI controller includes the following features:

- Software-programmable for Slave or Master
- Supports Repeated START signal
- Multi-master systems supported
- Allows 10-bit addressing with TWI bus
- Performs arbitration and clock synchronization
- Own address and General Call address detection
- Interrupt on address detection
- Supports speeds up to 400Kbits/s ('fast mode')
- Allows operation from a wide range of input clock frequencies

Data transferred are always in a unit of 8-bit (byte), followed by an acknowledge bit. The number of bytes that can be transmitted per transfer is unrestricted. Data is transferred in serial with the MSB first. Between each byte of data transfer, a receiver device will hold the clock line SCL low to force the transmitter into a wait state while waiting the response from microprocessor.

Data transfer with acknowledgement is obligatory. The clock line is driven by the master all the time, including the acknowledgement-related clock cycle, except for the SCL holding between each bytes. After sending each byte, the transmitter releases the SDA line to allow the receiver to pull down the SDA line and send an acknowledgement signal (or leave it high to send a "not acknowledge") to the transmitter.

When a slave receiver doesn't acknowledge the slave address (unable to receive because of no resource available), the data line must be left high by the slave so that the master can then generate a STOP condition to abort the transfer. Slave receiver can also indicate not to want to send more data during a transfer by leave the acknowledgement signal high. And the master should generate the STOP condition to abort the transfer.

The figure provides an illustration the relation of SDA signal line and SCL signal line on the TWI serial bus.

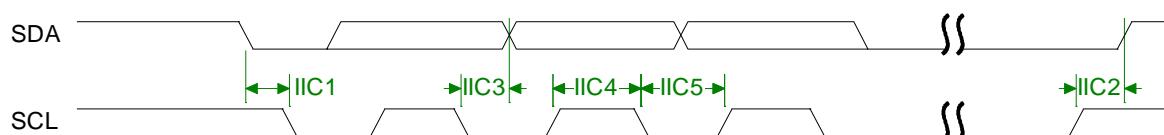


Figure 4-5 TWI Timing Diagram

### 4.3.2 Register List

Module Name	Base Address	
TWI Controller 0	0x40041c00	
TWI Controller 1	0x40042000	
Register Name	Offset Address	Description
TWI_ADDR	0x0000	TWI Slave address
TWI_XADDR	0x0004	TWI Extended slave address
TWI_DATA	0x0008	TWI Data byte

TWI_CNTR	0x000C	TWI Control register
TWI_STAT	0x0010	TWI Status register
TWI_CCR	0x0014	TWI Clock control register
TWI_SRST	0x0018	TWI Software reset
TWI_EFR	0x001C	TWI Enhance Feature register
TWI_LCR	0x0020	TWI Line Control register

**Table 4-40 TWI Register List**

### 4.3.3 Register Description

#### 4.3.3.1 TWI Slave Address Register

Address : 0x0000			Name: TWI_ADDR Default: 0x0000_0000
Field	Type	Default	Description
31:8	/	/	/
7:1	R/W	0	<p>SLA</p> <p>Slave address</p> <ul style="list-style-type: none"> <li>- 7-bit addressing</li> <li>SLA6, SLA5, SLA4, SLA3, SLA2, SLA1, SLA0</li> </ul> <p>- 10-bit addressing</p> <p>1, 1, 1, 1, 0, SLAX[9:8]</p> <p>For 7-bit addressing:</p> <p>SLA6 – SLA0 is the 7-bit address of the TWI when in slave mode. When the TWI receives this address after a START condition, it will generate an interrupt and enter slave mode. (SLA6 corresponds to the first bit received from the TWI bus.) If GCE is set to '1', the TWI will also recognize the general call address (00h).</p> <p>For 10-bit addressing:</p> <p>When the address received starts with 11110b, the TWI recognizes this as the first part of a 10-bit address and if the next two bits match ADDR[2:1] (i.e. SLAX9 and SLAX8 of the device's extended address), it sends an ACK. (The device does not generate an interrupt at this point.) If the next byte of the address matches the XADDR register (SLAX7 – SLAX0), the TWI generates an interrupt and goes into slave mode.</p>
0	R/W	0	GCE
			General call address enable

			0: Disable 1: Enable
--	--	--	-------------------------

**Table 4-41 TWI Slave Address Register**

#### 4.3.3.2 TWI Extend Address Register

<b>Address :</b> 0x0004			<b>Name:</b> TWI_XADDR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	R/W	0	SLAX Extend Slave Address SLAX[7:0]

**Table 4-42 TWI Extend Address Register**

#### 4.3.3.3 TWI Data Register

<b>Address :</b> 0x0008			<b>Name:</b> TWI_DATA <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	R/W	0	TWI_DATA Data byte for transmitting or received

**Table 4-43 TWI Data Register**

#### 4.3.3.4 TWI Control Register

<b>Address :</b> 0x000C			<b>Name:</b> TWI_CTR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7	R/W	0	INT_EN Interrupt Enable 1'b0: The interrupt line always low 1'b1: The interrupt line will go high when INT_FLAG is set.
6	R/W	0	BUS_EN TWI Bus Enable 1'b0: The TWI bus inputs ISDA/ISCL are ignored and the TWI Controller will

			<p>not respond to any address on the bus</p> <p>1'b1: The TWI will respond to calls to its slave address – and to the general call address if the GCE bit in the ADDR register is set.</p> <p>Notes: In master operation mode, this bit should be set to '1'</p>
5	R/W AC	0	<p>M_STA</p> <p><b>Master Mode Start</b></p> <p>When M_STA is set to '1', TWI Controller enters master mode and will transmit a START condition on the bus when the bus is free. If the M_STA bit is set to '1' when the TWI Controller is already in master mode and one or more bytes have been transmitted, then a repeated START condition will be sent. If the M_STA bit is set to '1' when the TWI is being accessed in slave mode, the TWI will complete the data transfer in slave mode then enter master mode when the bus has been released.</p> <p>The M_STA bit is cleared automatically after a START condition has been sent: writing a '0' to this bit has no effect.</p>
4	R/W1 C	0	<p>M_STP</p> <p><b>Master Mode Stop</b></p> <p>If M_STP is set to '1' in master mode, a STOP condition is transmitted on the TWI bus. If the M_STP bit is set to '1' in slave mode, the TWI will behave as if a STOP condition has been received, but no STOP condition will be transmitted on the TWI bus. If both M_STA and M_STP bits are set, the TWI will first transmit the STOP condition (if in master mode) then transmit the START condition.</p> <p>The M_STP bit is cleared automatically: writing a '0' to this bit has no effect.</p>
3	R/W1 C	0	<p>INT_FLAG</p> <p><b>Interrupt Flag</b></p> <p>INT_FLAG is automatically set to '1' when any of 28 (out of the possible 29) states is entered (see 'STAT Register' below). The only state that does not set INT_FLAG is state F8h. If the INT_EN bit is set, the interrupt line goes high when IFLG is set to '1'. If the TWI is operating in slave mode, data transfer is suspended when INT_FLAG is set and the low period of the TWI bus clock line (SCL) is stretched until '1' is written to INT_FLAG. The TWI clock line is then released and the interrupt line goes low.</p>
2	R/W	0	<p>A_ACK</p> <p><b>Assert Acknowledge</b></p> <p>When A_ACK is set to '1', an Acknowledge (low level on SDA) will be sent</p>

			<p>during the acknowledge clock pulse on the TWI bus if:</p> <ol style="list-style-type: none"> <li>1. Either the whole of a matching 7-bit slave address or the first or the second byte of a matching 10-bit slave address has been received.</li> <li>2. The general call address has been received and the GCE bit in the ADDR register is set to '1'.</li> <li>3. A data byte has been received in master or slave mode.</li> </ol> <p>When A_ACK is '0', a Not Acknowledge (high level on SDA) will be sent when a data byte is received in master or slave mode.</p> <p>If A_ACK is cleared to '0' in slave transmitter mode, the byte in the DATA register is assumed to be the 'last byte'. After this byte has been transmitted, the TWI will enter state C8h then return to the idle state (status code F8h) when INT_FLAG is cleared.</p> <p>The TWI will not respond as a slave unless A_ACK is set.</p>
1:0	/	/	/

**Table 4-44 TWI Control Register**

#### 4.3.3.5 TWI Status Register

Address : 0x0010			Name: TWI_STAT Default: 0x0000_00f8
Field	Type	Default	Description
31:8	/	/	/
7:0	R	0xF8	<p>STA</p> <p>Status Information Byte</p> <p>Code Status</p> <p>0x00: Bus error</p> <p>0x08: START condition transmitted</p> <p>0x10: Repeated START condition transmitted</p> <p>0x18: Address + Write bit transmitted, ACK received</p> <p>0x20: Address + Write bit transmitted, ACK not received</p> <p>0x28: Data byte transmitted in master mode, ACK received</p> <p>0x30: Data byte transmitted in master mode, ACK not received</p> <p>0x38: Arbitration lost in address or data byte</p> <p>0x40: Address + Read bit transmitted, ACK received</p> <p>0x48: Address + Read bit transmitted, ACK not received</p> <p>0x50: Data byte received in master mode, ACK transmitted</p> <p>0x58: Data byte received in master mode, not ACK transmitted</p> <p>0x60: Slave address + Write bit received, ACK transmitted</p> <p>0x68: Arbitration lost in address as master, slave address + Write bit</p>

		<p>received, ACK transmitted</p> <p>0x70: General Call address received, ACK transmitted</p> <p>0x78: Arbitration lost in address as master, General Call address received, ACK transmitted</p> <p>0x80: Data byte received after slave address received, ACK transmitted</p> <p>0x88: Data byte received after slave address received, not ACK transmitted</p> <p>0x90: Data byte received after General Call received, ACK transmitted</p> <p>0x98: Data byte received after General Call received, not ACK transmitted</p> <p>0xA0: STOP or repeated START condition received in slave mode</p> <p>0xA8: Slave address + Read bit received, ACK transmitted</p> <p>0xB0: Arbitration lost in address as master, slave address + Read bit received, ACK transmitted</p> <p>0xB8: Data byte transmitted in slave mode, ACK received</p> <p>0xC0: Data byte transmitted in slave mode, ACK not received</p> <p>0xC8: Last byte transmitted in slave mode, ACK received</p> <p>0xD0: Second Address byte + Write bit transmitted, ACK received</p> <p>0xD8: Second Address byte + Write bit transmitted, ACK not received</p> <p>0xF8: No relevant status information, INT_FLAG=0</p> <p>Others: Reserved</p>
--	--	--

**Table 4-45 TWI Status Register**

#### 4.3.3.6 TWI Clock Register

<b>Address : 0x0014</b>			<b>Name:</b> TWI_CCR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:7	/	/	/
6:3	R/W	0	CLK_M
2:0	R/W	0	<p>CLK_N</p> <p>The TWI bus is sampled by the TWI at the frequency defined by F0:  <math>F_{samp} = F_0 = Fin / 2^{CLK\_N}</math></p> <p>The TWI OSCL output frequency, in master mode, is <math>F_1 / 10</math>:  <math>F_1 = F_0 / (CLK\_M + 1)</math>  <math>F_{oscl} = F_1 / 10 = Fin / (2^{CLK\_N} * (CLK\_M + 1) * 10)</math></p> <p>For Example:  <math>Fin = 48MHz</math> (APB clock input)  For 400kHz full speed 2Wire, <math>CLK\_N = 2</math>, <math>CLK\_M = 2</math>  <math>F_0 = 48M / 2^2 = 12MHz</math>, <math>F_1 = F_0 / (10 * (2+1)) = 0.4MHz</math>  For 100Khz standard speed 2Wire, <math>CLK\_N = 2</math>, <math>CLK\_M = 11</math>  <math>F_0 = 48M / 2^2 = 12MHz</math>, <math>F_1 = F_0 / (10 * (11+1)) = 0.1MHz</math></p>

**Table 4-46 TWI Clock Register**

<b>Address :</b> 0x0018			<b>Name:</b> TWI_SRST <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:1	/	/	/
0	R/W AC	0	<p><b>SOFT_RST</b></p> <p>Soft Reset</p> <p>Write ‘1’ to this bit to reset the TWI and clear to ‘0’ when completing Soft Reset operation.</p>

**Table 4-47 TWI Soft Reset Register**

#### 4.3.3.7TWI Soft Reset Register

<b>Address :</b> 0x001C			<b>Name:</b> TWI_EFR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:2	/	/	/
0:1	R/W	0	<p><b>DBN</b></p> <p>Data Byte number follow Read Command Control</p> <p>No Data Byte to be wrote after read command</p> <p>Only 1 byte data to be wrote after read command</p> <p>2 bytes data can be wrote after read command</p> <p>3 bytes data can be wrote after read command</p>

**Table 4-48 TWI Enhance Feature Register**

#### 4.3.3.8TWI Enhance Feature Register

<b>Address :</b> 0x0020			<b>Name:</b> TWI_LCR <b>Default:</b> 0x0000_003a
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:6	/	/	/
5	R	1	<p><b>SCL_STATE</b></p> <p>Current state of TWI_SCL</p> <p>0 – low</p> <p>1 - high</p>
4	R	1	<b>SDA_STATE</b>

			Current state of TWI_SDA 0 – low 1 - high
3	R/W	1	SCL_CTL  TWI_SCL line state control bit When line control mode is enabled (bit[2] set), value of this bit decide the output level of TWI_SCL 0 – output low level 1 – output high level
2	R/W	0	SCL_CTL_EN  TWI_SCL line state control enable When this bit is set, the state of TWI_SCL is control by the value of bit[3]. 0-disable TWI_SCL line control mode 1-enable TWI_SCL line control mode
1	R/W	1	SDA_CTL  TWI_SDA line state control bit When line control mode is enabled (bit[0] set), value of this bit decide the output level of TWI_SDA 0 – output low level 1 – output high level
0	R/W	0	SDA_CTL_EN  TWI_SDA line state control enable When this bit is set, the state of TWI_SDA is control by the value of bit[1]. 0-disable TWI_SDA line control mode 1-enable TWI_SDA line control mode

**Table 4-49 TWI Line Control Register**

## 4.4 SDMMC/SDIO Controller (SDC)

### 4.4.1 Overview

The SD3.0 controller can be configured either as a Secure Digital Multimedia Card controller, which simultaneously supports Secure Digital memory (SD Memo), UHS-1 Card, Secure Digital I/O (SDIO), Multimedia Cards (MMC), eMMC Card and Consumer Electronics Advanced Transport Architecture (CE-ATA).

The SD3.0 controller includes the following features:

- Supports Secure Digital memory protocol commands (up to SD3.0)

- Supports Secure Digital I/O protocol commands
- Supports Multimedia Card protocol commands (up to MMC4.41)
- Supports CE-ATA digital protocol commands
- Supports eMMC boot operation and alternative boot operation
- Supports Command Completion signal and interrupt to host processor and Command Completion Signal disable feature
- Supports one SD (Version1.0 to 3.0) or MMC (Version3.3 to 4.41) or CE-ATA device
- Supports hardware CRC generation and error detection
- Supports programmable baud rate
- Supports host pull-up control
- Supports SDIO interrupts in 1-bit and 4-bit modes
- Supports SDIO suspend and resume operation
- Supports SDIO read wait
- Supports block size of 1 to 65535 bytes
- Supports descriptor-based internal DMA controller
- Internal 128 bytes FIFO for data transfer

#### 4.4.2 Register List

<b>Module Name</b>	<b>Base Address</b>	
SDMMC/SDIO Controller 0	0x40002000	
<b>Register Name</b>	<b>Offset Address</b>	<b>Description</b>
SD_GCTL	0x00	Control register
SD_CKCR	0x04	Clock Control register
SD_TMOR	0x08	Time out register
SD_BWDR	0x0C	Bus Width register
SD_BKSR	0x10	Block size register
SD_BYCR	0x14	Byte count register
SD_CMDR	0x18	Command register
SD_CAGR	0x1c	Command argument register
SD_RESP0	0x20	Response 0 register
SD_RESP1	0x24	Response 1 register
SD_RESP2	0x28	Response 2 register
SD_RESP3	0x2C	Response 3 register
SD_IMKR	0x30	Interrupt mask register
SD_MISR	0x34	Masked interrupt status register
SD_RISR	0x38	Raw interrupt status register
SD_STAR	0x3C	Status register
SD_FWLR	0x40	FIFO Water Level register
SD_FUNS	0x44	FIFO Function Select register
SD_CBCR	0x48	Transferred byte count between controller and card

SD_BBCR	0x4c	Transferred byte count between host memory and internal FIFO
SD_DBGC	0x50	Current Debug Control address
SD_A12A	0x58	Auto command 12 argument
SD_NTSR	0x5c	SD New Timing Set Register
SD_SDDBG	0x60	SD New Timing Set Debug Register
SD_HWRST	0x78	Hardware Reset Register
SD_DMAC	0x80	BUS Mode Control
SD_DLBA	0x84	Descriptor List Base Address
SD_IDST	0x88	DMAC Status
SD_IDIE	0x8c	DMAC Interrupt Enable
SD_CHDA	0x90	Current host descriptor address
SD_CBDA	0x94	Current buffer descriptor address
SD_THLDC	0x100	Card Threshold Control register
SD_DSBD	0x10c	eMMC4.5 DDR Start Bit Detection Control
SD_RESP_CRC	0x110	Response CRC
SD_D7_CRC	0x114	DATA7 CRC
SD_D6_CRC	0x118	DATA7 CRC
SD_D5_CRC	0x11c	DATA7 CRC
SD_D4_CRC	0x120	DATA7 CRC
SD_D3_CRC	0x124	DATA7 CRC
SD_D2_CRC	0x128	DATA7 CRC
SD_D1_CRC	0x12c	DATA7 CRC
SD_D0_CRC	0x130	DATA7 CRC
SD_WCRC_STA	0x134	Write DATA CRC Status
SD_DRV_DL	0x140	Drive delay control register
SD_SMAP_DL	0x144	sample delay control register
SD_DS_DL	0x148	Data strobe signal delay control register
SD_FIFO	0x200	Read/ Write FIFO

**Table 4-50 SD Register List**

## 4.4.3 Register Description

### 4.4.3.1 SD Global Control Register

<b>Address :</b> 0x0000		<b>Name:</b> SD_GCTRL <b>Default:</b> 0x0000_0300	
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	R/W	0	FIFO_ACCESS_MODE

			0: DMA bus 1: AHB bus
30:11	/	/	/
10	R/W	0	DDR_MODE  DDR Mode Select 0: SDR mode 1: DDR mode
9	/	/	/
8	R/W	1	CARD_DBC_ENB  Card Detect (D3 irq) De-bounce Enable 0: disable de-bounce 1: enable de-bounce
7:6	/	/	/
5	R/W	0	IDMA_ENB  Internal DMA Enable 0: disable IDMA to transfer data 1: enable IDMA to transfer data
4	R/W	0	INT_ENB  Global Interrupt Enable 0: disable interrupts 1: enable interrupts
3	/	/	/
2	R/WAC	0	IDMA_RST  0: no effect 1: Internal DMA Reset. This bit is auto-cleared after reset operation.
1	R/WAC	0	FIFO_RST  0: no effect 1: FIFO Reset. This bit is auto-cleared after reset operation.
0	R/WAC	0	SOFT_RST  0: no effect 1: Software Reset. This bit is auto-cleared after reset operation.

**Table 4-51 SD Global Control Register**

#### 4.4.3.2SD Clock Control Register

<b>Address :</b> 0x0004			<b>Name:</b> SD_CLK_CTRL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	R/W	0	MASK_DATA0  0: ignore the data0 status when update card clock 1: consider the data0 status when update card clock.
20:18	/	/	/
17	R/W	0	CCLK_CTRL  0: card clock always on 1: card clock is turned off when FSM is in IDLE state.
16	R/W	0	CCLK_ENB  0: card clock is off 1: card clock is on
15:8	/	/	/
7:0	R/W	0	CCLK_DIV  Card Clock Divider n: the card source clock is divided by 2 * n (n = 0 ~ 255)

Table 4-52 SD Clock Control Register

#### 4.4.3.3SD Timeout Register

<b>Address :</b> 0x0008			<b>Name:</b> SD_TMOUT <b>Default:</b> 0xffff_ff40
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	R/W	0xffffffff	DATA_TMO_LIMIT  Data Timeout Limit
7:0	R/W	0x40	RSP_TMO_LIMIT  Response Timeout Limit

Table 4-53 SD Timeout Register

#### 4.4.3.4SD Card Width Register

<b>Address :</b> 0x000C	<b>Name:</b> SD_WIDTH <b>Default:</b> 0x0000_0000
-------------------------	--

Field	Type	Default	Description
31:2	/	/	/
1:0	R/W	0	CARD_WIDTH  Card Width 0: 1-bit width 1: 4-bit width 2'b1x: 8-bit width

Table 4-54 SD Card Width Register

#### 4.4.3.5 SD Block Size Register

<b>Address :</b> 0x0010			<b>Name:</b> SD_BLK_SZ <b>Default:</b> 0x0000_0200
Field	Type	Default	Description
31:16	/	/	/
15:0	R/W	0x200	BLK_SIZE  Transfer Block Size

Table 4-55 SD Clock Control Register

#### 4.4.3.6 SD Byte Counter Register

<b>Address :</b> 0x0014			<b>Name:</b> SD_BYTE_CNT <b>Default:</b> 0x0000_0200
Field	Type	Default	Description
31:0	R/W	0x200	BYTE_CNT  Transfer Byte Counter The value is the number of bytes to be transferred and must be integer multiple of BLK_SIZE for block transfers.

Table 4-56 SD Clock Control Register

#### 4.4.3.7 SD Command Register

<b>Address :</b> 0x0018			<b>Name:</b> SD_CMD <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31	R/W	0	CMD_LOAD  Start Command. This bit is auto-cleared when current command is sent. If there is no any

			response error happened, a command complete interrupt bit, CMD_OVER, will be set in SD_INTR register. You should not write any other command before this bit is cleared. Or a command busy interrupt bit, CMD_BUSY, will be set in SD_INTR register.
30:29	/	/	/
28	R/W	0	<p>VOL_SWITCH</p> <p>Voltage Switch</p> <p>0: this command is a normal command</p> <p>1: this command is a voltage switch command, only set for CMD11.</p>
27	R/W	0	<p>BOOT_ABRT</p> <p>Boot Abort</p> <p>Setting this bit will terminate the boot operation.</p>
26	R/W	0	<p>EXP_BOOT_ACK</p> <p>Expect the Boot Acknowledge</p> <p>When software sets this bit along in mandatory boot operation, controller expects a boot acknowledge start pattern of 0-1-0 from the selected card.</p>
25:24	R/W	0	<p>BOOT_MODE</p> <p>2'b00: normal command</p> <p>2'b01: Mandatory Boot Operation</p> <p>2'b10: Alternate Boot Operation</p> <p>2'b11: reserved</p>
23	R/W	0	<p>CCS_EXP</p> <p>0 – Interrupts are not enabled in CE-ATA device (nIEN = 1 in ATA control register), or command does not expect CCS from device</p> <p>1 – Interrupts are enabled in CE-ATA device (nIEN = 0), and RW_BLK command expects command completion signal from CE-ATA device</p> <p>If the command expects Command Completion Signal, CCS, from CE-ATA device, the software should set this bit. SD/MMC sets DATA_OVER bit in SD_INTR register and generates interrupt to host if Data Transfer Over is not masked.</p>
22	R/W	0	<p>RD_CEATA_DEV</p> <p>0 – Host is not performing read access (RW_REG or RW_BLK) towards CE-ATA device</p> <p>1 – Host is performing read access (RW_REG or RW_BLK) towards CE-ATA device</p>

21	R/W	0	UPD_CLK  Update Clock  0: normal command  1: Update Card Clock  When this bit is set, controller will update clock domain and clock output state. No command will be sent in this situation.
20:16	/	/	/
15	R/W	0	SEND_INIT_SEQ  0: normal command  1: Send initialization sequence before sending this command.
14	R/W	0	STOP_ABT_CMD  0: normal command  1: Send stop or abort command to stop current data transfer in progress. CMD12, CMD52 for writing I/O Abort in SDIO CCCR
13	R/W	0	WAIR_PRE_OVER  Wait Data Transfer Over 0: send command at once, do not care of data transferring 1: wait for data transfer completion before sending current command
12	R/W	0	STOP_CMD_FLAG  Send Stop Command Automatically, CMD12 0: do not send CMD12 at the end of data transfer 1: send CMD12 automatically at the end of data transfer
11	R/W	0	TRANS_MODE  0: Block mode 1: Stream mode
10	R/W	0	TRANS_DIR  0: Read operation 1: Write operation
9	R/W	0	DATA_TRANS

			0: no data transfer 1: with data transfer
8	R/W	0	CHK_RESP_CRC  0: do not check the response CRC 1: check the response CRC
7	R/W	0	LONG_RESP  0: short response 1: long response
6	R/W	0	RESP_RCV  0: command without a response 1: command with a response
5:0	R/W	0	CMD_INDEX  Command Index

**Table 4-57 SD Command Register**

#### 4.4.3.8 SD Command Argument Register

<b>Address :</b> 0x001C			<b>Name:</b> SD_CMD_ARG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0	CMD_ARG  Command Argument

**Table 4-58 SD Command Argument Register**

#### 4.4.3.9 SD Response 0 Register

<b>Address :</b> 0x0020			<b>Name:</b> SD_RESPO <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R	0	CMD_RESPO  Command Response 0

**Table 4-59 SD Response 0 Register**

#### 4.4.3.10 SD Response 1 Register

<b>Address :</b> 0x0024	<b>Name:</b> SD_RESP1
-------------------------	-----------------------

			<b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R	0	CMD_RESP1  Command Response 1

**Table 4-60 SD Response 1 Register**

#### 4.4.3.11 SD Response 2 Register

<b>Address :</b> 0x0028			<b>Name:</b> SD_RESP2 <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R	0	CMD_RESP2  Command Response 2

**Table 4-61 SD Response 2 Register**

#### 4.4.3.12 SD Response 3 Register

<b>Address :</b> 0x002C			<b>Name:</b> SD_RESP3 <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R	0	CMD_RESP3  Command Response 3

**Table 4-62 SD Response 0 Register**

#### 4.4.3.13 SD Interrupt Mask Register

<b>Address :</b> 0x0030			<b>Name:</b> SD_INT_MASK <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R/W	0	INT_MASK  0: interrupt is disabled 1: interrupt is enabled  Bit field defined as follows: bit 31: card removed bit 30: card inserted bit 17~19: reserved bit 16: SDIO interrupt

		bit 15: Data End-bit Error bit 14: Auto Stop Command Done bit 13: Data Start Bit Error bit 12: Command Busy and illegal write bit 11: FIFO under run or overflow bit 10: Data Starvation Timeout/V1.8 Switch Done bit 9: Data Timeout/Boot Data Start bit 8: Response Timeout/Boot ACK Received bit 7: Data CRC Error bit 6: Response CRC Error bit 5: Data Receive Request bit 4: Data Transmit Request bit 3: Data Transfer complete bit 2: Command complete bit 1: Response Error, no response or CRC error bit 0: reserved
--	--	---

Table 4-63 SD Interrupt Mask Register

#### 4.4.3.14 SD Masked Interrupt Status Register

<b>Address :</b> 0x0034			<b>Name:</b> SD_MINTR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R	0	<p><b>MSKDISTA</b></p> <p>Interrupt status. Enabled only when corresponding bit in INT_MASK register is set.</p> <p>Bit field defined as follows:</p> <ul style="list-style-type: none"> <li>bit 31: card removed</li> <li>bit 30: card inserted</li> <li>bit 17~19: reserved</li> <li>bit 16: SDIO interrupt</li> <li>bit 15: Data End-bit Error</li> <li>bit 14: Auto Stop Command Done</li> <li>bit 13: Data Start Bit Error</li> <li>bit 12: Command Busy and illegal write</li> <li>bit 11: FIFO under run or overflow</li> <li>bit 10: Data Starvation Timeout/V1.8 Switch Done</li> <li>bit 9: Data Timeout/Boot Data Start</li> <li>bit 8: Response Timeout/Boot ACK Received</li> <li>bit 7: Data CRC Error</li> <li>bit 6: Response CRC Error</li> </ul>

			bit 5: Data Receive Request bit 4: Data Transmit Request bit 3: Data Transfer complete bit 2: Command complete bit 1: Response Error, no response or CRC error bit 0: reserved
--	--	--	---

**Table 4-64 SD Masked Interrupt Register****4.4.3.15 SD Raw Interrupt Status Register**

<b>Address :</b> 0x0038			<b>Name:</b> SD_RINTR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W1C	0	RAWISTA  Raw Interrupt status. These bits are write-1-to-clear bits. Bit field defined as follows: bit 31: card removed bit 30: card inserted bit 17~19: reserved bit 16: SDIO interrupt bit 15: Data End-bit Error bit 14: Auto Stop Command Done bit 13: Data Start Bit Error bit 12: Command Busy and illegal write bit 11: FIFO under run or overflow bit 10: Data Starvation Timeout/V1.8 Switch Done bit 9: Data Timeout/Boot Data Start bit 8: Response Timeout/Boot ACK Received bit 7: Data CRC Error bit 6: Response CRC Error bit 5: Data Receive Request bit 4: Data Transmit Request bit 3: Data Transfer complete bit 2: Command complete bit 1: Response Error, no response or CRC error bit 0: reserved

**Table 4-65 SD Raw Interrupt Register**

#### 4.4.3.16 SD Status Register

<b>Address :</b> 0x003c			<b>Name:</b> SD_STATUS <b>Default:</b> 0x0000_0006
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	R	0	DMA_REQ  DMA Request Signal State
30:23	/	/	/
22:17	R	0	FIFO_LEVEL  Number of filled locations in FIFO
16:11	R	0	RESP_IDX  Index of previous response, including the auto-stop command sent by controller.
10	R	0	FSM_BUSY  Data transmit or receive state machine is busy.
9	R	0	CARD_BUSY  Inverted version of DATA0 0: card is not busy 1: card is busy
8	R	0	CARD_PRESENT  Status of DATA3 0: card not present 1: card present
7:4	R	0	FSM_STA  Command FSM States: 0: Idle 1: Send init sequence 2: TX CMD start bit 3: TX CMD tx bit 4: TX CMD index/arg 5: TX CMD CRC7 6: TX CMD End bit 7:RX RESP start bit 8: RX RESP IRQ response 9: RX RESP tx bit

			10: RX RESP CMD Index 11: RX RESP data 12: RX RESP CRC7 13: RX RESP end bit 14: CMD path wait NCC 15: Wait, CMD-to-RESP turnaround
3	R	0	FIFO_FULL  0: FIFO not full 1: FIFO full
2	R	1	FIFO_EMPTY  0: FIFO not empty 1: FIFO empty
1	R	1	FIFO_TX_LEVEL  FIFO TX Water Level Flag 0: FIFO does not reach the transmit trigger level 1: FIFO reaches transmit trigger level
0	R	0	FIFO_RX_LEVEL  FIFO RX Water Level Flag 0: FIFO does not reach the receive trigger level 1: FIFO reaches receive trigger level

Table 4-66 SD Status Register

#### 4.4.3.17 SD Water Level Register

<b>Address :</b> 0x0040			<b>Name:</b> SD_FIFO_WL <b>Default:</b> 0x000F_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	/	/	/
30:28	R/W	0	BSIZE_OF_TRANS  Burst size of multiple transaction 000: 1 transfers      001: 4 010: 8      011: 16 100: 32      101: 64 110: 128      111: 256 Should be set same as DMA controller multiple transaction size. The units for transfers are the DWORD. A single transfer would be signaled based on this value. Value should be sub-multiple of RX_TL +1 and FIFO_DEPTH –

			TX_TL Recommended: MSize = 8, TX_TL = 16, RX_TL = 15
27:21	/	/	/
20:16	R/W	0xF	RX_TL  RX Trigger Level 0x0 ~ 0x1e: 0 ~ 30 0x1f: reserved  FIFO threshold when FIFO request host to receive data from FIFO. When FIFO data level is greater than this value, DMA request is raised if DMA enabled, or RX interrupt bit is set if interrupt enabled. At the end of packet, if the last transfer is less than this level, the value will be ignored and relative request will be raised as usual. Recommended: 15
15:5	/	/	/
4:0	R/W	0xF	TX_TL  TX Trigger Level 0x1 ~ 0x1f: 1 ~ 31 0x0: reserved  FIFO threshold when FIFO request host to transmit data from FIFO. When FIFO data level is less than or equal to this value, DMA RX request is raised if DMA enabled, or TX interrupt bit is set if interrupt enabled. At the end of packet, if the last transfer is less than this level, the value will be ignored and relative request will be raised as usual. Recommended: 16

Table 4-67 SD Water Level Register

#### 4.4.3.18 SD Function Select Register

Address : 0x0044			Name: SD_FUNC_SEL Default: 0x0000_0000
Field	Type	Default	Description
31:16	R/W	0	CEATA_EN  CEATA Support ON/OFF 0xceaa – CEATA support on. All hidden CEATA relative bits are accessible normally and these 16 bits return value of 0x1 when be read. Other value – CEATA support off. All hidden CEATA relative bits cannot be

			access and these 16 bits return value of 0 when be read.
15:11	-	0	/
10	R/W	0	<p>CEATAISTA</p> <p>CEATA device interrupt status</p> <p>0 – Interrupts not enabled in CE-ATA device (nIEN = 1 in ATA control register)</p> <p>1 – Interrupts are enabled in CE-ATA device (nIEN = 0 in ATA control register)</p> <p>Software should appropriately write to this bit after power-on reset or any other reset to CE-ATA device. After reset, usually CE-ATA device interrupt is disabled (nIEN = 1). If the host enables CE-ATA device interrupt, then software should set this bit.</p>
9	R/W	0	<p>SEND_AUTO_STOP_CCSD</p> <p>send auto stop ccsd</p> <p>0 – Clear bit if SD/MMC does not reset the bit.</p> <p>1 – Send internally generated STOP after sending CCSD to CE-ATA device. When set, SD/MMC automatically sends internally generated STOP command (CMD12) to CE-ATA device. After sending internally-generated STOP command, Auto Command Done (ACD) bit in RINTSTS is set and generates interrupt to host if Auto Command Done interrupt is not masked. After sending the CCSD, SD/MMC automatically clears send_auto_stop_ccsd bit.</p>
8	R/W	0	<p>SEND_CCSD</p> <p>send ccsd</p> <p>0 – Clear bit if SD/MMC does not reset the bit.</p> <p>1 – Send Command Completion Signal Disable (CCSD) to CE-ATA device. When set, SD/MMC sends CCSD to CE-ATA device. Software sets this bit only if current command is expecting CCS (that is, RW_BLK) and interrupts are enabled in CE-ATA device. Once the CCSD pattern is sent to device, SD/MMC automatically clears send_ccsd bit. It also sets Command Done (CD) bit in RINTSTS register and generates interrupt to host if Command Done interrupt is not masked.</p>
7:3	-	-	/
2	R/W	0	<p>ABT_RDATA</p> <p>Abort Read Data</p> <p>0 – Ignored</p> <p>1 – After suspend command is issued during read-transfer, software polls card to find when suspend happened. Once suspend occurs, software sets bit to reset data state-machine, which is waiting for next block of data.</p>

			Used in SDIO card suspends sequence. This bit is auto-cleared once controller reset to idle state.
1	R/W	0	<p>READ_WAIT</p> <p>Read Wait</p> <p>0 – Clear SDIO read wait 1 – Assert SDIO read wait</p>
0	R/W	0	<p>HOST_SEND_MMC_IRQRESQ</p> <p>Host Send MMC IRQ Response</p> <p>0 – Ignored 1 – Send auto IRQ response</p> <p>When host is waiting MMC card interrupt response, setting this bit will make controller cancel wait state and return to idle state, at which time, controller will receive IRQ response sent by itself.</p> <p>This bit is auto-cleared after response is sent.</p>

**Table 4-68 SD Function Select Register**

#### 4.4.3.19 SD Transferred CIU Card Byte Counter Register

<b>Address :</b> 0x0048			<b>Name:</b> SD_TCBCNT <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R	0	<p>TRANS_BYTE_CNT_TO_CARD</p> <p>Number of bytes transferred by CIU unit to card.</p> <p>The register should be accessed in full to avoid read-coherency problems and read only after data transfer completes.</p>

**Table 4-69 SD Transferred CIU Card Byte Counter Register**

<b>Address :</b> 0x004c			<b>Name:</b> SD_TBBCNT <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R	0	<p>TRANS_BYTE_CNT_ON_BUS</p> <p>Number of bytes transferred by Host/DMA memory and BIU FIFO.</p> <p>The register should be accessed in full to avoid read-coherency problems and read only after data transfer completes.</p>

**Table 4-70 SD Transferred Host to BUI-FIFO Byte Counter Register**

#### 4.4.3.21 SD Debug Control Register

<b>Address :</b> 0x0050			<b>Name:</b> SD_DBG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:12	/	/	/
11:0	R/W	0	<p><b>DBG_ON</b></p> <p>0xdeb – Debug on. ALL hidden register bits will be accessible and these 12 bits return 0x1 when be read.</p> <p>Other values – Debug off. All hidden register bits will not be accessed and these 12 bits return 0 when be read.</p>

Table 4-71 SD Debug Control Register

#### 4.4.3.22 SD Auto Command CMD12 Argument Register

<b>Address :</b> 0x0058			<b>Name:</b> SD_A12A <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:16	/	/	/
0:15	R/W	0xffff	<p><b>SD_A12A</b>.</p> <p>SD_A12A set the argument of command 12 automatically send by controller</p>

Table 4-72 SD Auto CMD12 Argument Register

#### 4.4.3.23 SD New Timing Set Register

<b>Address :</b> 0x005c			<b>Name:</b> SD_NTSR <b>Default:</b> 0x8000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	R/W	1	<p><b>MODE_SELEC</b></p> <p>0: Old mode of sample/output timing.</p> <p>1: New mode of sample/output timing</p>
30:28	/	/	/
27	R/W	0	<p><b>DATA0_BYPASS</b></p> <p>Select data0 input async or bypass sample logic, used to check card busy or not</p> <p>0: disable</p> <p>1: enable</p>
26:25	/	/	/

24	R/W	1	CMD_DAT_RX_PHASE_CLR During update clock, command and data rx phase clear 0: disable 1: enable
23	/	/	/
22	R/W	1	DAT_CRC_STATUS_RX_PHASE_CLR Before receiving CRC status, data rx phase clear 0: disable 1: enable
21	R/W	1	DAT_TRANS_RX_PHASE_CLR Before transferring data, data rx phase clear 0: disable 1: enable
20	R/W	1	DAT_RECV_RX_PHASE_CLR Before receiving data, data rx phase clear 0: disable 1: enable
19:17	/	/	/
16	R/W	1	CMD_SEND_RX_PHASE_CLR Before send command, command rx phase clear 0: disable 1: enable
15:6	/	/	/
5:4	R/W	0	SAMPLE_TIMING_PHASE 00: sample timing phase offset 90° 01: sample timing phase offset 180° 10: sample timing phase offset 270° 11: ignore
3:0	/	/	/

Table 4-73 SD Auto CMD12 Argument Register

#### 4.4.3.24 SD Hardware Reset Register

<b>Address :</b> 0x0078			<b>Name:</b> SD_HRST <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:1	/	/	/
0	R/W	1	HW_RST.  1 – Active mode 0 – Reset  These bits cause the cards to enter pre-idle state, which requires them to

		be re-initialized.
--	--	--------------------

**Table 4-74 SD Hardware Reset Register**

#### 4.4.3.25 SD DMAC Control Register

<b>Address :</b> 0x0080			<b>Name:</b> SD_DMA_CTRL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	W	0	<p><b>DES_LOAD_CTRL</b></p> <p>When DMAC fetches a descriptor, if the valid bit of a descriptor is not set, DMAC FSM will go to the suspend state. Setting this bit will make DMAC re-fetch descriptor again and do the transfer normally.</p>
30:11	/	/	/
10:8	R	0	<p><b>PRG_BURST_LEN</b></p> <p>Programmable Burst Length.</p> <p>These bits indicate the maximum number of beats to be performed in one IDMAC transaction. The IDMAC will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. The permissible values are 1, 4, 8, 16, 32, 64, 128 and 256. This value is the mirror of MSIZE of FIFOTH register. In order to change this value, write the required value to FIFOTH register. This is an encode value as follows.</p> <ul style="list-style-type: none"> <li>000 – 1 transfers</li> <li>001 – 4 transfers</li> <li>010 – 8 transfers</li> <li>011 – 16 transfers</li> <li>100 – 32 transfers</li> <li>101 – 64 transfers</li> <li>110 – 128 transfers</li> <li>111 – 256 transfers</li> </ul> <p>Transfer unit is 16, 32, or 64 bits, based on HDATA_WIDTH. PBL is a read-only value.</p>
7	R/W	0	<p><b>IDMAC_ENB</b></p> <p>IDMAC Enable.</p> <p>When set, the IDMAC is enabled. DE is read/write.</p>
6:2	R/W	0	<p><b>DES_SKIP_LEN</b></p> <p>Descriptor Skip Length.</p> <p>Specifies the number of Word to skip between two unchained descriptors.</p> <p>This is applicable only for dual buffer structure.</p>

			Default value is set to 4 DWORD.
1	R/W	0	<p>FIX_BUST_CTRL</p> <p>Fixed Burst.</p> <p>Controls whether the AHB Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 during start of normal burst transfers. When reset, the AHB will use SINGLE and INCR burst transfer operations.</p>
0	R/W	0	<p>IDMAC_RST</p> <p>DMA Reset.</p> <p>When set, the DMA Controller resets all its internal registers. SWR is read/write. It is automatically cleared after 1 clock cycle.</p>

**Table 4-75 SD DMAC Control Register**

#### 4.4.3.26 SD Descriptor List Base Address Register

<b>Address :</b> 0x0084			<b>Name:</b> SD_DLBA_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0	<p>DES_BASE_ADDR</p> <p>Start of Descriptor List.</p> <p>Contains the base address of the First Descriptor. The LSB bits [1:0] are ignored and taken as all-zero by the IDMAC internally. Hence these LSB bits are read-only.</p>

**Table 4-76 SD Descriptor List Base Address Register**

#### 4.4.3.27 SD DMAC Status Register

<b>Address :</b> 0x0088			<b>Name:</b> SD_DMASTER <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:17	/	/	/
16:13	R	0	<p>DMAC_FSM_STA</p> <p>DMAC FSM present state.</p> <p>0 – DMA_IDLE</p> <p>1 – DMA_SUSPEND</p> <p>2 – DESC_RD</p> <p>3 – DESC_CHK</p> <p>4 – DMA_RD_REQ_WAIT</p>

			<p>5 – DMA_WR_REQ_WAIT 6 – DMA_RD 7 – DMA_WR 8 – DESC_CLOSE This bit is read-only.</p>
12:10	R	0	<p>DMAC_ERR_STA</p> <p>Error Bits.</p> <p>Indicates the type of error that caused a Bus Error. Valid only with Fatal Bus Error bit (IDSTS[2]) set. This field does not generate an interrupt.</p> <p>3'b001 – Host Abort received during transmission 3'b010 – Host Abort received during reception Others: Reserved EB is read-only.</p>
9	R/W	0	<p>ABN_INT_SUM</p> <p>Abnormal Interrupt Summary. Logical OR of the following:</p> <p>IDSTS[2] – Fatal Bus Interrupt IDSTS[4] – DU bit Interrupt IDSTS[5] – Card Error Summary Interrupt Only unmasked bits affect this bit.</p> <p>This is a sticky bit and must be cleared each time a corresponding bit that causes AIS to be set is cleared. Writing a 1 clears this bit.</p>
8	R/W	0	<p>NOR_INT_SUM</p> <p>Normal Interrupt Summary. Logical OR of the following:</p> <p>IDSTS[0] – Transmit Interrupt IDSTS[1] – Receive Interrupt Only unmasked bits affect this bit.</p> <p>This is a sticky bit and must be cleared each time a corresponding bit that causes NIS to be set is cleared. Writing a 1 clears this bit.</p>
7:6	/	/	/
5	R/W	0	<p>ERR_FLAG_SUM</p> <p>Card Error Summary. Indicates the status of the transaction to/from the card; also present in RINTSTS. Indicates the logical OR of the following bits:</p> <p>EBE – End Bit Error RTO – Response Timeout/Boot Ack Timeout RCRC – Response CRC SBE – Start Bit Error DRTO – Data Read Timeout/BDS timeout</p>

			DCRC – Data CRC for Receive RE – Response Error Writing a 1 clears this bit.
4	R/W	0	DES_UNAVL_INT  Descriptor Unavailable Interrupt. This bit is set when the descriptor is unavailable due to OWN bit = 0 (DESO[31] =0). Writing a 1 clears this bit.
3	/	/	/
2	R/W	0	FATAL_BERR_INT  Fatal Bus Error Interrupt. Indicates that a Bus Error occurred (IDSTS[12:10]). When this bit is set, the DMA disables all its bus accesses. Writing a 1 clears this bit.
1	R/W	0	RX_INT  Receive Interrupt. Indicates the completion of data reception for a descriptor. Writing a 1 clears this bit.
0	R/W	0	TX_INT  Transmit Interrupt. Indicates that data transmission is finished for a descriptor. Writing a '1' clears this bit.

Table 4-77 SD DMA Status Register

#### 4.4.3.28 SD DMAC Interrupt Enable Register

<b>Address :</b> 0x008c			<b>Name:</b> SD_IDIE <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:10	/	/	/
9	R/W	0	ABN_INT_ENB  Abnormal Interrupt Summary Enable. When set, an abnormal interrupt is enabled. This bit enables the following bits: IDINTEN[2] – Fatal Bus Error Interrupt IDINTEN[4] – DU Interrupt IDINTEN[5] – Card Error Summary Interrupt
8	R/W	0	NOR_INT_ENB

			Normal Interrupt Summary Enable. When set, a normal interrupt is enabled. When reset, a normal interrupt is disabled. This bit enables the following bits: IDINTEN[0] – Transmit Interrupt IDINTEN[1] – Receive Interrupt
7:6	/	/	/
5	R/W	0	ERR_SUM_INT_ENB  Card Error summary Interrupt Enable. When set, it enables the Card Interrupt summary.
4	R/W	0	DES_UNAVL_INT_ENB  Descriptor Unavailable Interrupt. When set along with Abnormal Interrupt Summary Enable, the DU interrupt is enabled.
3	/	/	/
2	R/W	0	FERR_INT_ENB  Fatal Bus Error Enable. When set with Abnormal Interrupt Summary Enable, the Fatal Bus Error Interrupt is enabled. When reset, Fatal Bus Error Enable Interrupt is disabled.
1	R/W	0	RX_INT_ENB  Receive Interrupt Enable. When set with Normal Interrupt Summary Enable, Receive Interrupt is enabled. When reset, Receive Interrupt is disabled.
0	R/W	0	TX_INT_ENB  Transmit Interrupt Enable. When set with Normal Interrupt Summary Enable, Transmit Interrupt is enabled. When reset, Transmit Interrupt is disabled.

Table 4-78 SD DMA Interrupt Enable Register

#### 4.4.3.29 SD Current Host Descriptor Address Register

<b>Address :</b> 0x0090			<b>Name:</b> SD_CHDAR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R	0	CUR_DES_ADDR  Host Descriptor Address Pointer.

			Cleared on reset. Pointer updated by IDMAC during operation. This register points to the start address of the current descriptor read by the IDMAC.
--	--	--	---

**Table 4-79 SD Current Host Descriptor Address Register****4.4.3.30 SD Current Buffer Descriptor Address Register**

<b>Address :</b> 0x009c			<b>Name:</b> SD_CLK_CTRL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R	0	<p>CUR_BUFF_ADDR</p> <p>Host Buffer Address Pointer.</p> <p>Cleared on Reset. Pointer updated by IDMAC during operation. This register points to the current Data Buffer Address being accessed by the IDMAC.</p>

**Table 4-80 SD Current Buffer Descriptor Address Register****4.4.3.31 SD Card Threshold Control Register**

<b>Address :</b> 0x0100			<b>Name:</b> SD_THLD_CTRL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:28	/	/	/
27:16	R/W	0	<p>CARD_RD_THLD</p> <p>Card Read Threshold Size</p>
15:1	/	/	/
0	R/W	0	<p>CARD_RD_THLD_ENB</p> <p>Card Read Threshold Enable</p> <p>0 – Card Read Threshold Disable</p> <p>1 - Card Read Threshold Enable</p> <p>Host controller initiates Read Transfer only if CARD_RD_THLD amount of space is available in receive FIFO</p>

**Table 4-81 SD Card Threshold Control Register****4.4.3.32 eMMC4.5 DDR Start Bit Detection Control Register**

<b>Address :</b> 0x010c			<b>Name:</b> EMMC_DDR_SBIT_DET <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>

31:1	/	/	/
0	R/W	0	<p>HALF_START_BIT</p> <p>Control for start bit detection mechanism inside mstorage based on duration of start bit.</p> <p>For eMMC 4.5, start bit can be:</p> <ul style="list-style-type: none"> <li>0 - Full cycle</li> <li>1 - Less than one full cycle</li> </ul> <p>Set HALF_START_BIT=1 for eMMC 4.5 and above; set to 0 for SD applications.</p>

Table 4-82 eMMC DDR Start Bit Detection Register

#### 4.4.3.33 SD Response CRC Register

<b>Address :</b> 0x0110			<b>Name:</b> SD_RESP_CRC <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:7	/	/	/
6:0	R	0	RESP_CRC

Table 4-83 SD Response CRC Register

#### 4.4.3.34 SD DATA7 CRC Register

<b>Address :</b> 0x0114			<b>Name:</b> SD_DATA7_CRC <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R	0	DATA7_CRC

Table 4-84 SD DATA7 CRC Register

#### 4.4.3.35 SD DATA6 CRC Register

<b>Address :</b> 0x0118			<b>Name:</b> SD_DATA6_CRC <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R	0	DATA6_CRC

Table 4-85 SD DATA6 CRC Register

#### 4.4.3.36 SD DATA5 CRC Register

<b>Address :</b> 0x011C			<b>Name:</b> SD_DATA5_CRC <b>Default:</b> 0x0000_0000
-------------------------	--	--	--

Field	Type	Default	Description
31:0	R	0	DATA5_CRC

Table 4-86 SD DATA5 CRC Register

#### 4.4.3.37 SD DATA4 CRC Register

<b>Address</b> : 0x0120			<b>Name:</b> SD_DATA4_CRC <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R	0	DATA4_CRC

Table 4-87 SD DATA4 CRC Register

#### 4.4.3.38 SD DATA3 CRC Register

<b>Address</b> : 0x0124			<b>Name:</b> SD_DATA3_CRC <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R	0	DATA3_CRC

Table 4-88 SD DATA3 CRC Register

#### 4.4.3.39 SD DATA2 CRC Register

<b>Address</b> : 0x0128			<b>Name:</b> SD_DATA2_CRC <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R	0	DATA2_CRC

Table 4-89 SD DATA2 CRC Register

#### 4.4.3.40 SD DATA1 CRC Register

<b>Address</b> : 0x012C			<b>Name:</b> SD_DATA1_CRC <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R	0	DATA1_CRC

Table 4-90 SD DATA1 CRC Register

#### 4.4.3.41 SD DATA0 CRC Register

<b>Address</b> : 0x0130			<b>Name:</b> SD_DATA0_CRC <b>Default:</b> 0x0000_0000
Field	Type	Default	Description

31:0	R	0	DATA0_CRC
------	---	---	-----------

**Table 4-91 SD DATA0 CRC Register**

#### 4.4.3.42 SD CRC Status Register

<b>Address :</b> 0x0134			<b>Name:</b> SD_CRC_STA <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:3	/	/	/
2:0	R	0	CRC_STATUS

**Table 4-92 SD CRC Status Register**

#### 4.4.3.43 SD FIFO Register

<b>Address :</b> 0x0200			<b>Name:</b> SD_FIFO <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0	TX/RX_FIFO  Data FIFO

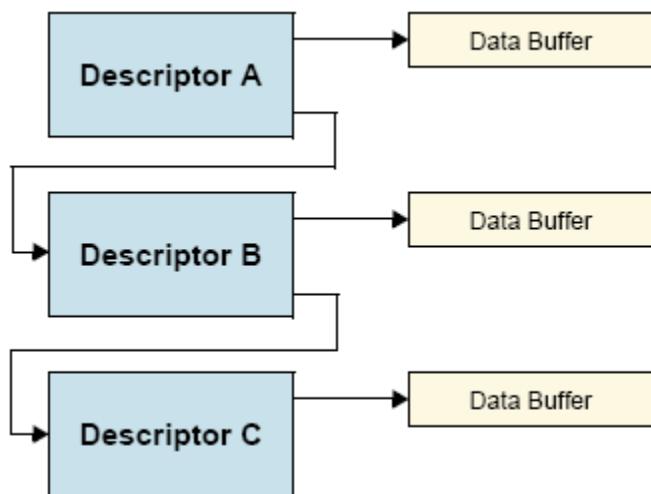
**Table 4-93 SD FIFO Register**

### 4.4.4 SD/MMC DMA Controller Description

SD3.0 controller has an internal DMA controller (IDMAC) to transfer data between host memory and SDMMC port. With a descriptor, IDMAC can efficiently move data from source to destination by automatically loading next DMA transfer arguments, which need less CPU intervention. Before transfer data in IDMAC, host driver should construct a descriptor list, configure arguments of every DMA transfer, then launch the descriptor and start the DMA. IDMAC has an interrupt controller, when enabled, it can interrupt the HOST CPU in situations such as data transmission completed or some errors happened.

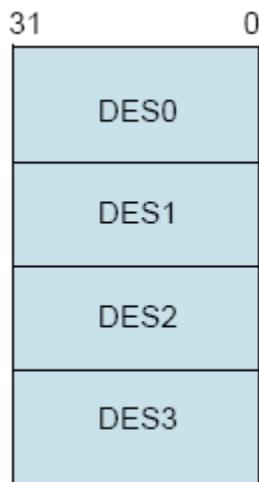
#### 4.4.4.1 IDMAC Descriptor Structure

The IDMAC uses a descriptor with a chain structure, and each descriptor points to a unique buffer and the next descriptor.



**Figure 4-6 SDC IDMAC Chain Descriptor**

This figure illustrates the internal formats of a descriptor. The descriptor addresses must be aligned to the bus width used for 32-bit buses. Each descriptor contains 16 bytes of control and status information.



**Figure 4-7 SDC IDMAC Fix Step Descriptor**

DES0 is a notation used to denote the [31:0] bits, DES1 to denote [63:32] bits, DES2 to denote [95:64] bits, and DES3 to denote [127:96] bits in a descriptor.

#### 4.4.4.2 DES0 definition

Bits	Name	Descriptor
31	HOLD	<b>DES_OWN_FLAG</b>  When set, this bit indicates that the descriptor is owned by the IDMAC. When this bit is reset, it indicates that the descriptor is owned by the host. This bit is cleared when transfer is over.

30	ERROR	ERR_FLAG  When some error happened in transfer, this bit will be set.
29:5	/	/
4	Chain Flag	CHAIM_MOD  When set, this bit indicates that the second address in descriptor is the next descriptor address. Must be set 1.
3	First DES Flag	FIRST_FLAG  When set, this bit indicates that this descriptor contains the first buffer of data. Must be set to 1 in first DES.
2	Last DES Flag	LAST_FLAG  When set, this bit indicates that the buffers pointed to by this descriptor are the last data buffer
1	Disable Interrupt on completion	CUR_TXRX_OVER_INT_DIS  When set, this bit will prevent the setting of the TX/RX interrupt bit of the IDMAC status register for data that ends in the buffer pointed to by this descriptor
0	/	/

**Table 4-94 SDC Descriptor 0 Register**

#### 4.4.4.3DES1 definition

Bits	Name	Descriptor
31:13	/	/
12:0	Buffer size	BUFF_SIZE  These bits indicate the data buffer byte size, which must be a multiple of 4 bytes. If this field is 0, the DMA ignores this buffer and proceeds to the next descriptor.

**Table 4-95 SDC Descriptor 1 Register**

#### 4.4.4.4DES2 definition

Bits	Name	Descriptor
31:0	Buffer address pointer	BUFF_ADDR  These bits indicate the physical address of data buffer. The IDMAC ignores DES2 [1:0], corresponding to the bus width of 32.

**Table 4-96 SDC Descriptor 2 Register**

#### 4.4.4.5 DES3 definition

Bits	Name	Descriptor
31:0	Next descriptor address	NEXT_DESP_ADDR These bits indicate the pointer to the physical memory where the next descriptor is present.

**Table 4-97 SDC Descriptor 3 Register**

### 4.5 Pulse Width Modulation/Event Capture Controller (PWM/ECT)

#### 4.5.1 Overview

PWM 模块带 8 个 PWM 通道，分 4 个 PWM 对： PWM01 对、 PWM23 对、 PWM45 对、 PWM67 对， PWM01 对由 PWM0 和 PWM1 通道构成， PWM23 对由 PWM2 和 PWM3 通道构成， PWM45 对由 PWM4 和 PWM5 通道构成， PWM67 对由 PWM6 和 PWM7 通道构成。 PWM 对主要由 3 部分组成： 1 个时钟控制器、 2 个定时器逻辑模块、 1 个可编程死区发生器。

每个 PWM 通道支持 PWM 输出和捕捉输入两大功能。 PWM 通道的时钟源有 24M\_PLL 和 APB1 (100M) 可选，经过时钟控制器处理后送至 PWM 通道的定时器逻辑模块。 PWM 通道配置为 PWM 输出功能，其定时器逻辑模块可输出单脉冲波形或周期波形，频率范围在 0Hz~24/100MHz 之间。

PWM 通道的定时器逻辑模块也可复用于捕捉输入功能， PWM 通道捕捉到外部上升沿时锁存 16-bit 加法计数器当前值至上升沿锁存寄存器 CRLR， PWM 通道捕捉到外部下降沿时锁存 16-bit 加法计数器当前值至下降沿锁存寄存器 CFLR，根据 2 个寄存器 CRLR 和 CFLR 的值可精确计算出外部时钟的频率。

PWM 对可配置执行互补输出或插入死区时间的 PWM 波形对输出。当 PWM 对里的 2 个通道的 prescale 相同、 period 寄存器值相同且 active state 相反时， PWM 对输出互补波形；当 PWM 对的可编程死区发生器被使能， PWM 对输出插入死区时间的波形对，波形对的死区插入时间可控。

PWM 通道配置为 PWM 输出功能或捕捉输入功能， PWM 通道都可配置产生中断。配置为 PWM 输出功能时，每当 16-bit 加法计数器达到 Entire cycle 值， PWM 通道可被使能产生中断；配置为捕捉输入功能时，每当 PWM 通道捕捉到外部上升沿， PWM 通道可被使能产生 1 个中断，当 PWM 通道捕捉到下降沿， PWM 通道可被使能产生 1 个中断，捕捉到上升沿或下降沿都可触发中断产生。

The PWM controller includes the following features:

- 8 个 PWM 通道，分 4 个 PWM 对
- 支持脉冲、周期和互补对输出
- 支持捕捉输入
- 带可编程死区发生器，死区时间可控
- 0-24/100M 输出频率范围， 0%-100% 占空比可调，最小分辨率 1/65536
- 支持 PWM 输出和捕捉输入中断

## 4.5.2 Operation and Functional Description

### 4.5.2.1 Typical Application

- Suitable for display device, such as LCD
- Suitable for electric motor control

### 4.5.2.2 Block Diagram

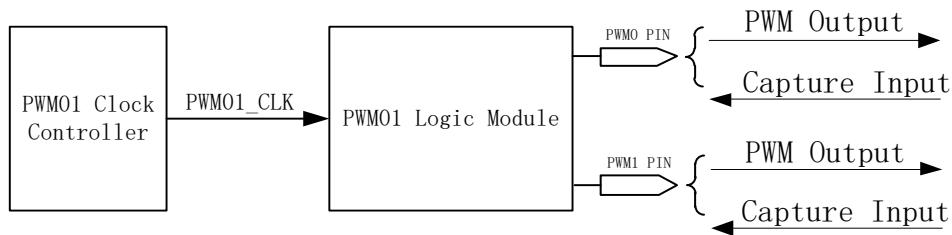
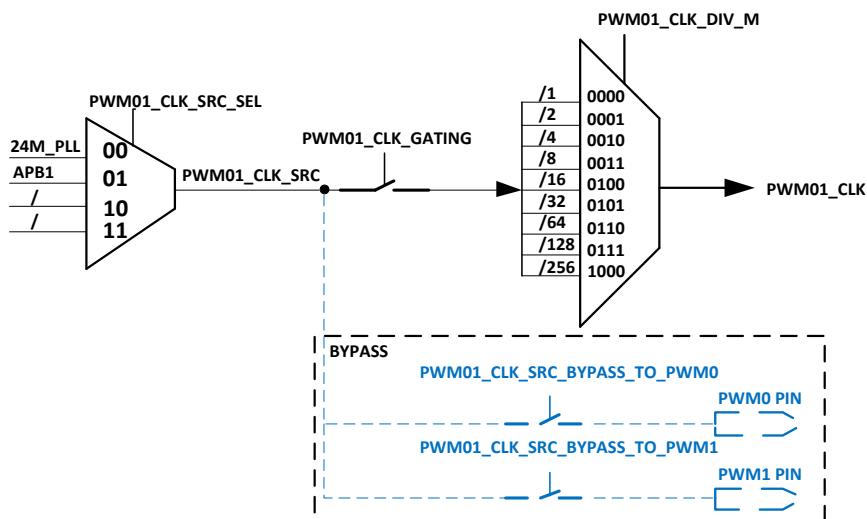


Figure 4-8 PWM Block Diagram

每个 PWM 对由 3 部分组成，如图 **Figure 4-8 PWM Block Diagram** 的 PWM01 对：1 个时钟控制器（PWM01 Clock Controller）、1 个 PWM01 逻辑模块（PWM01 Logic Module）、2 个 PWM 引脚（PIN），支持 PWM Output 和 Capture Input。每一个 PWM 通道都有 1 个引脚，使用 PWM Output 功能时需将 PWM PIN 配置为 PWM 输出，使用 Capture Input 功能时需将 PWM PIN 配置为捕捉输入。PWM23 对、PWM45 对、PWM67 对的组成与 PWM01 对的一致。

PWM01 对的 PWM01 逻辑模块包括 3 部分：2 个 8-bit 预分频器、2 个定时器逻辑模块（PWM TIMER LOGIC）、1 个可编程死区发生器（Dead Zone Generator 01）。时钟控制器包括时钟源选择、1~256 分频和时钟源选通。

### 4.5.2.3 Clock Controller 时钟控制器



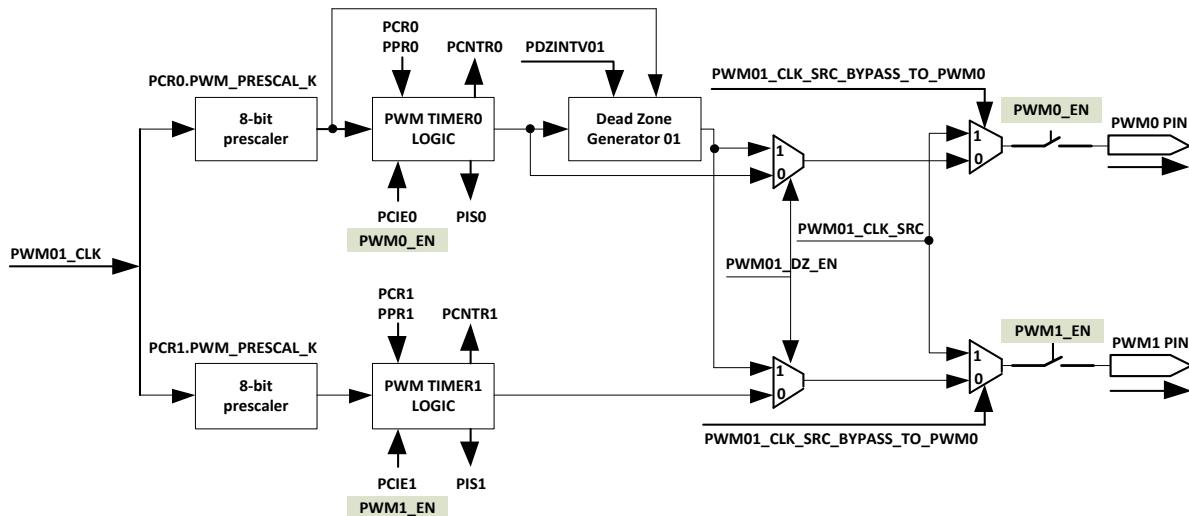
**Figure 4-9 PWM Clock Controller Diagram**

每个 PWM 对的时钟控制器如 PWM01 包括时钟源选择 (PWM01\_CLK\_SRC\_SEL)、1~256 分频 (PWM01\_CLK\_DIV\_M) 和时钟源选通 (CLK\_SRC\_BYPASS)。PWM 模块时钟源有 2 个：24M\_PLL 和 APB1，24M\_PLL 来源于 PLL 组，APB1 即 APB1 总线时钟(通常是 100MHz); 时钟源选通指直接将时钟源选通至 PWM 输出，PWM 通道输出的波形就是时钟源的波形，图中 BYPASS 虚框表示时钟源选通的功能，硬件上选通功能的实现请查看图 7-3。PWM01 对的时钟控制器输出的时钟 (PWM01\_CLK) 将送至 PWM01 逻辑模块使用。

#### 4.5.2.4 PWM Output

当使用 PWM 输出功能，PWM 对逻辑模块如 PWM01 逻辑模块的框图如下，其他 PWM23 对、PWM45 对和 PWM67 对的逻辑模块框图与 PWM01 对的一致。

PWM 通道的定时器逻辑模块 (PWM TIMER LOGIC) 主要由 1 个 16-bit 加法计数器和 1 个 16-bit 比较器组成，计数器和比较器均支持缓存载入，即 PWM 输出被使能后，计数器和比较器的寄存器值 (PPR.PWM\_ENTIRE\_CYCLE 和 PPR.PWM\_ACT\_CYCLE) 可以在任意时刻更改，更改的值会先缓存到缓存寄存器，当加法计数器的值达到 PPR.PWM\_ENTIRE\_CYCLE 值后，缓存寄存器里的值会被装载到计数器和比较器。缓存载入的作用是避免更新计数器和比较器的值时产生毛刺，导致 PWM 输出波形不稳定。加法计数器用于 PWM 周期控制，比较器用于占空比控制。PWM 支持周期模式和脉冲模式的波形输出，周期模式指加法计数器的值达到 PPR.PWM\_ENTIRE\_CYCLE 后，加法计数器的值会被自动重载为 0 并继续计数，因此输出一个连续的波形；脉冲模式指加法计数器的值达到 PPR.PWM\_ENTIRE\_CYCLE 后，加法计数器的值会被自动重载为 0 但停止计数，因此输出一个单脉冲的波形。

**Figure 4-10 PWM Output Logic Module Diagram**

- 加法计数器和比较器操作

PWM 输出波形的周期、占空比和 active state 由加法计数器和比较器决定，比较器的比较规则如下：

$PCNTR \geq (PWM\_ENTIRE\_CYCLE - PWM\_ACT\_CYCLE)$ , output "active state"

$PCNTR < (PWM\_ENTIRE\_CYCLE - PWM\_ACT\_CYCLE)$ , output " $\sim$  (active state)"

Active state 在每个 PWM 通道的 PCR 可编程控制，如 PWM0 通道：

假如 active state 设置为 high level，即 PCRO. PWM\_ACT\_STA = 1，

当 PCNTR0 >= (PPR0. PWM\_ENTIRE\_CYCLE - PPR0.PWM\_ACT\_CYCLE)，PWM0 outputs high level，即 output 1；

当 PCNTR0 < (PPR0. PWM\_ENTIRE\_CYCLE - PPR0.PWM\_ACT\_CYCLE)，PWM0 outputs ~ (high level)，即 output ~(1) = output 0；

Active state 设置为 low level 时同理。

PWM 输出波形的周期、占空比公式如下：

当 active state = high level，如图 Figure 4-11 PWM0 通道 active state 为 high level 的周期、占空比，

**1 个周期时间长度** = (PWM01\_CLK / PWM0\_PRESCALE\_K)<sup>-1</sup> \* PPR0. PWM\_ENTIRE\_CYCLE

**高电平时间长度** = (PWM01\_CLK / PWM0\_PRESCALE\_K)<sup>-1</sup> \* PPR0.PWM\_ACT\_CYCLE

**低电平时间长度** = (PWM01\_CLK / PWM0\_PRESCALE\_K)<sup>-1</sup> \* (PPR0. PWM\_ENTIRE\_CYCLE - PPR0.PWM\_ACT\_CYCLE)

**占空比** = 高电平时间长度 / 1 个周期时间长度

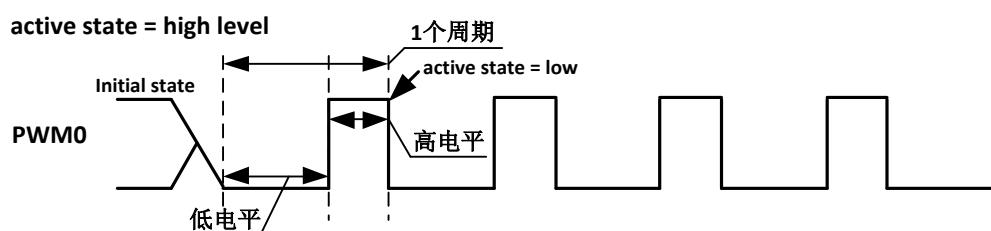


Figure 4-11 PWM0 通道 active state 为 high level 的周期、占空比

当 active state = low level，如图 Figure 4-12 PWM0 通道 active state 为 low level 的周期、占空比，

**1 个周期时间长度** = (PWM01\_CLK / PWM0\_PRESCALE\_K)<sup>-1</sup> \* PPR0. PWM\_ENTIRE\_CYCLE

**高电平时间长度** = (PWM01\_CLK / PWM0\_PRESCALE\_K)<sup>-1</sup> \* (PPR0. PWM\_ENTIRE\_CYCLE - PPR0.PWM\_ACT\_CYCLE)

**低电平时间长度** = (PWM01\_CLK / PWM0\_PRESCALE\_K)<sup>-1</sup> \* PPR0.PWM\_ACT\_CYCLE

**占空比** = 高电平时间长度 / 1 个周期时间长度

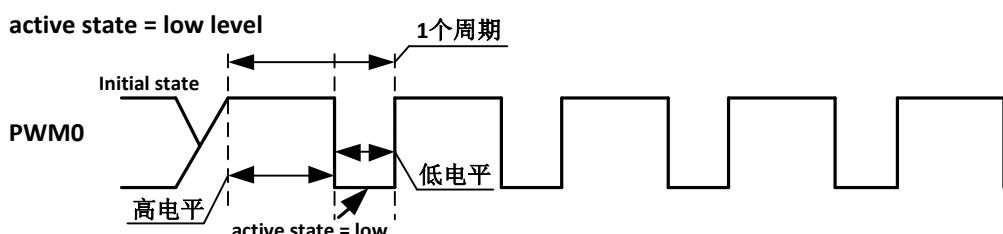
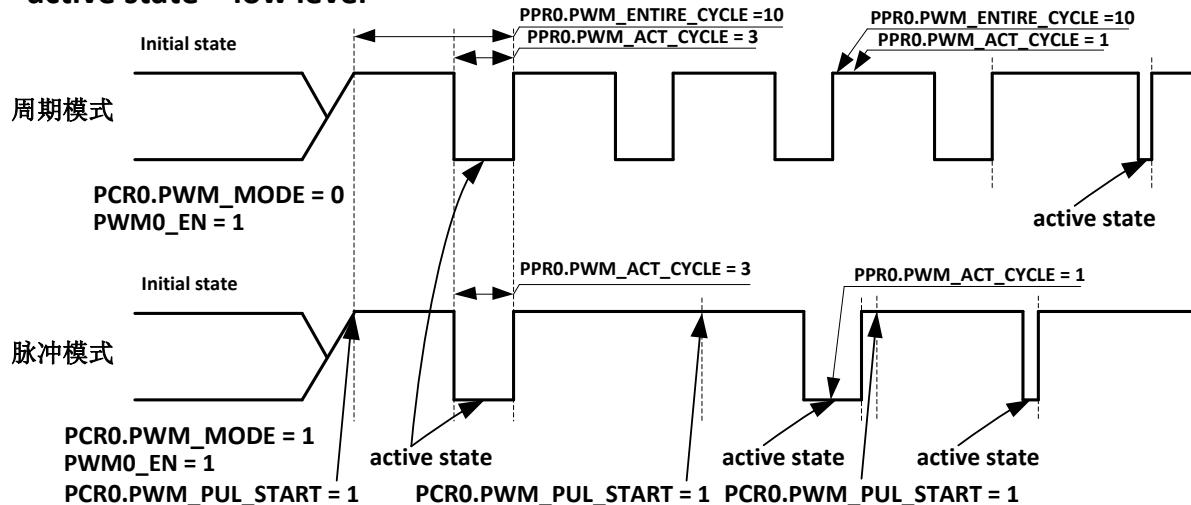


Figure 4-12 PWM0 通道 active state 为 low level 的周期、占空比

- 缓存载入、周期模式、脉冲模式和 active state

PWM 输出支持周期模式 (cycle mode) 和脉冲模式 (pulse mode)，周期模式下 PWM 输出连续的波形，脉冲模式下 PWM 输出 1 个单脉冲波形。如图 Figure 4-13 PWM01 对的 PWM0 通道缓存载入、周期模式、脉冲模式和 active state：

**active state = low level**



**active state = Hihg level**

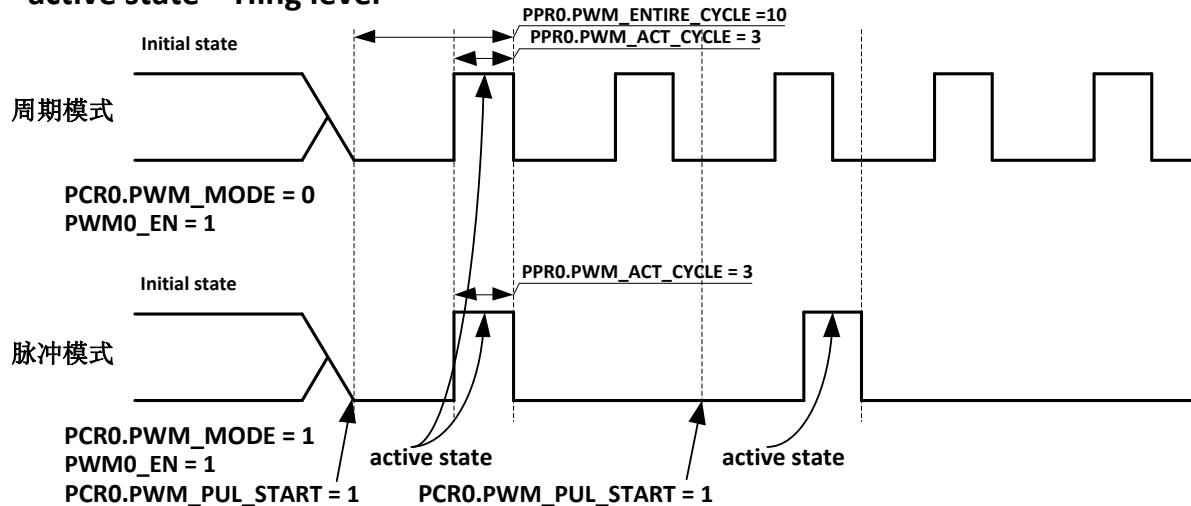


Figure 4-13 PWM01 对的 PWM0 通道缓存载入、周期模式、脉冲模式和 active state

PWM 模块的每一个通道都支持周期模式和脉冲模式的 PWM 输出，PWM 输出波形的 active state 可编程控制。

PCRO.PWM\_MODE = 0 时 PWM0 通道以周期模式输出，输出的连续波形的 1 个周期时间长度由 PPRO 的 PPRO.PWM\_ENTIRE\_CYCLE 值决定，输出的连续波形的 active state 时间长度由 PPRO 的 PPRO.PWM\_ACT\_CYCLE 值决定，计算公式如下：

$$1 \text{ 个周期时间长度} = (\text{PWM01\_CLK} / \text{PWM0\_PRESCALE\_K})^{-1} * \text{PPRO.PWM\_ENTIRE\_CYCLE}$$

$$\text{active state 时间长度} = (\text{PWM01\_CLK} / \text{PWM0\_PRESCALE\_K})^{-1} * \text{PPRO.PWM\_ACT\_CYCLE}$$

PCRO.PWM\_ACT\_STA = 0 时，输出连续波形的 active state 为低电平，

PCRO. PWM\_ACT\_STA = 1 时，输出连续波形的 active state 为高电平。

PCRO.PWM\_MODE = 1 时 PWM0 通道以脉冲模式输出，输出的单脉冲波形的宽度和电平状态由 PPRO 的 PPRO.PWM\_ACT\_CYCLE 值和 PCRO 寄存器的 PWM\_ACT\_STA 位决定，脉冲宽度计算公式如下：

$$\text{脉冲宽度} = \text{PWM01_CLK} / \text{PWM0_PRESCALE\_K} * \text{PPRO.PWM_ACT_CYCLE}$$

PCRO. PWM\_ACT\_STA = 0 时，脉冲电平是低电平，即 PWM0 通道输出低脉冲，

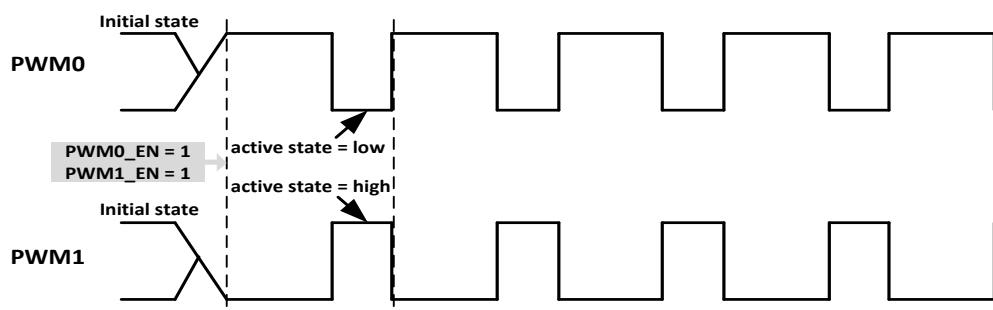
PCRO. PWM\_ACT\_STA = 1 时，脉冲电平是高电平，即 PWM0 通道输出高脉冲。

PWM0 通道使能后，PWM0 通道输出脉冲波形需将 PCRO. PWM\_PUL\_START 置 1，置 1 后 PWM0 通道输出 1 个单脉冲波形，输出完毕后 PCRO. PWM\_PUL\_START 位由硬件自动清零，PCRO. PWM\_PUL\_START 位自动清零后可进行下次置 1 操作输出 1 个脉冲波形。

PWM0 通道的加法计数器和比较器支持缓存载入，PWM0 通道使能后，不管是周期模式还是脉冲模式，PPRO 值被更改后会先被缓存到 PPRO 的缓存寄存器，当加法计数器的值达到 PPRO.PWM\_ENTIRE\_CYCLE 值后，缓存寄存器里的值会被装载到计数器和比较器，即加法计数器和比较器的值会在下个周期被重载。以图 7-6 active state = low level 的情况为例，周期模式下 PPRO 初始 PPRO.PWM\_ENTIRE\_CYCLE 值是 10，初始 PPRO.PWM\_ACT\_CYCLE 值是 3，在某一时刻 PPRO 的 PPRO.PWM\_ACT\_CYCLE 值被修改为 1，在当前周期内被修改的 PPRO 的值被缓存到 PPRO 缓存寄存器，下个周期开始时 PPRO 缓存寄存器值被载入到加法计数器和比较器，然后加法计数器开始工作。脉冲模式下，PPRO 的初始 PPRO.PWM\_ACT\_CYCLE 值是 3，在某个单脉冲波形产生过程中，PPRO.PWM\_ACT\_CYCLE 值被修改为 1，在当前周期内被修改的 PPRO 的值被缓存到 PPRO 缓存寄存器，加法计数器的值达到 PPRO.PWM\_ENTIRE\_CYCLE 值后即脉冲波形输出完毕，PPRO 缓存寄存器值被载入到加法计数器和比较器，在下次 PCRO. PWM\_PUL\_START 位置 1 后可看到 PPRO 值更改已生效。PPRO 缓存寄存器值被载入到加法计数器和比较器的时间非常短，可忽略，不影响 PWM 输出。

### ● 互补输出

每一个 PWM 对都支持 PWM 对波形输出，PWM 对波形输出包括互补对输出和插入死区时间的 PWM 对输出。互补对输出如图 **Figure 4-14 PWM01 对互补对输出**：



**Figure 4-14 PWM01 对互补对输出**

PWM01 对实现互补对输出需满足以下 3 个条件：

1. PWM0 与 PWM1 通道的频率、占空比一致
2. PWM0 与 PWM1 通道的 active state 相反

### 3. 同时使能 PWM0 与 PWM1 通道

- 死区发生器

每个 PWM 对都带一个可编程的死区发生器。

PWM01\_DZ\_EN = 1 后，PWM01 对的死区功能使能，PWM01 对输出一对插入死区时间的 PWM 波形，PWM01 对输出的波形由 PWM0 定时器逻辑模块和 Dead Zone Generator 01 决定，输出的波形如图 Figure 4-15 插入死区时间的 PWM01 波形对输出：

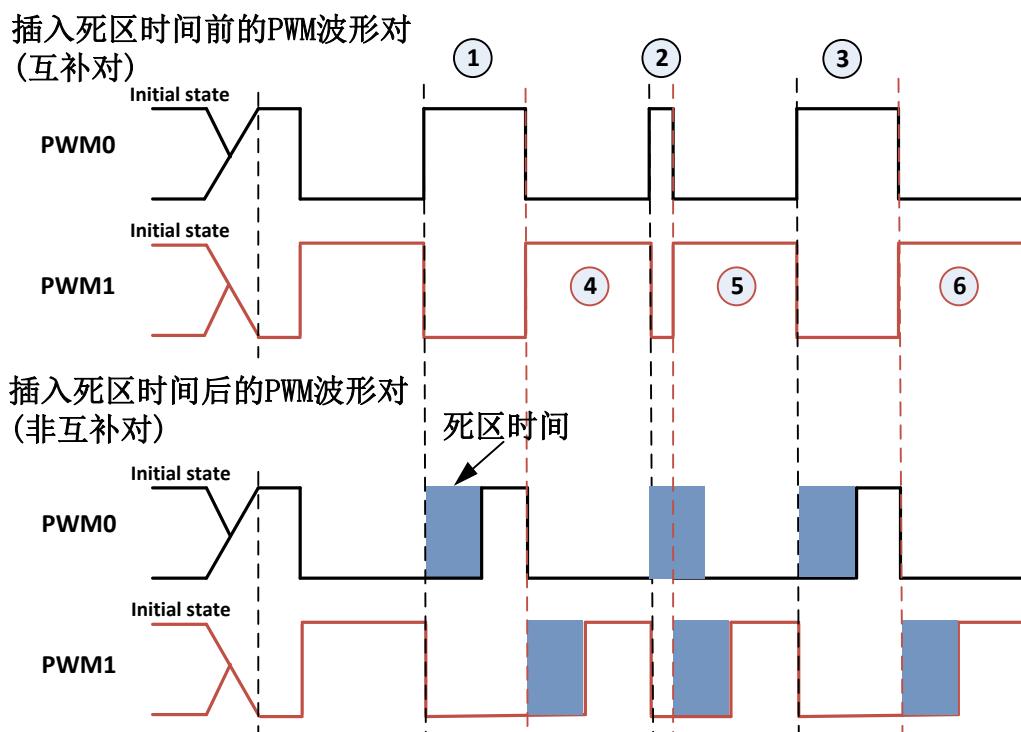
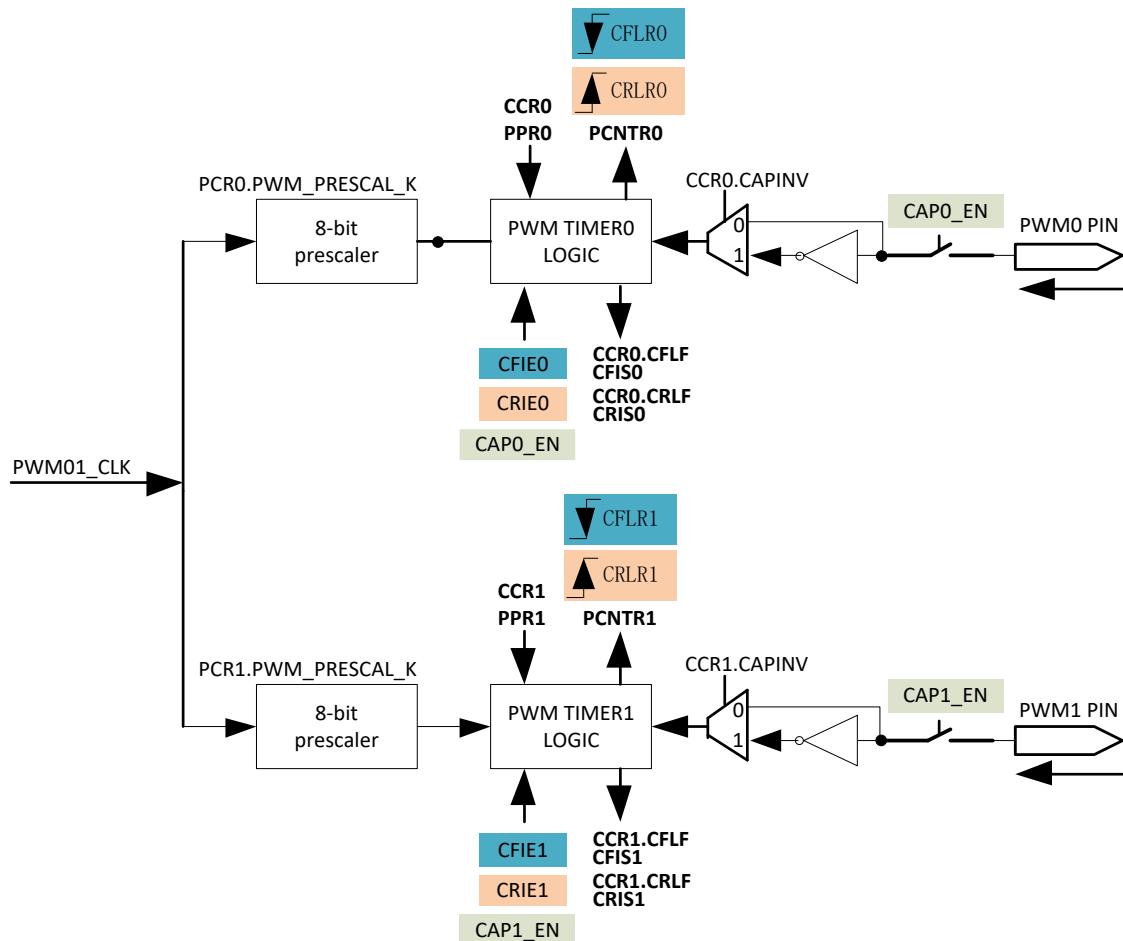


Figure 4-15 插入死区时间的 PWM01 波形对输出

插入死区时间前的 PWM 波形对指在 Dead Zone Generator 01 内部未插入死区时间的一对互补波形，插入死区时间后的 PWM 波形对指在 Dead Zone Generator 01 内部在互补对基础上插入死区时间后的一对非互补的 PWM 波形对，这对 PWM 波形对即是最终在 PWM0 PIN 和 PWM1 PIN 输出的波形。Dead Zone Generator 01 在处理互补对时，插入死区时间的原则是：凡遇到上升沿就插入一段死区时间。如果遇到图中②的高电平宽度比死区时间小，那么死区时间会覆盖这段高电平时间，因此死区时间的设置需考虑输出波形的周期和占空比。死区时间计算公式：

$$\text{死区时间} = (\text{PWM01\_CLK} / \text{PWM0\_PRESCALE\_K})^{-1} * \text{PDZINTV01}$$

#### 4.5.2.5 Capture Input



**Figure 4-16 PWM01 Capture Logic Module Diagram**

每一个 PWM 对里的 PWM 通道的定时器逻辑模块除了用于产生 PWM 输出，还被复用于捕捉外部时钟，定时器逻辑模块捕捉外部时钟的上升沿和下降沿。以 PWM0 通道为例，PWM0 通道有 1 个 CFLR0 和 1 个 CRLR0 用于锁存捕捉到下降沿时加法计数器的当前值和捕捉到上升沿时加法计数器的当前值，根据 CFLR0 和 CRLR0 的值可以计算出外部时钟的周期，计算公式如下：

$$\text{外部时钟高电平时间长度} = (\text{PWM01\_CLK} / \text{PWM0\_PRESCALE\_K})^{-1} * \text{CRLR0}$$

$$\text{外部时钟低电平时间长度} = (\text{PWM01\_CLK} / \text{PWM0\_PRESCALE\_K})^{-1} * \text{CFLR0}$$

$$\text{外部时钟周期} = \text{外部时钟高电平时间长度} + \text{外部时钟低电平时间长度}$$

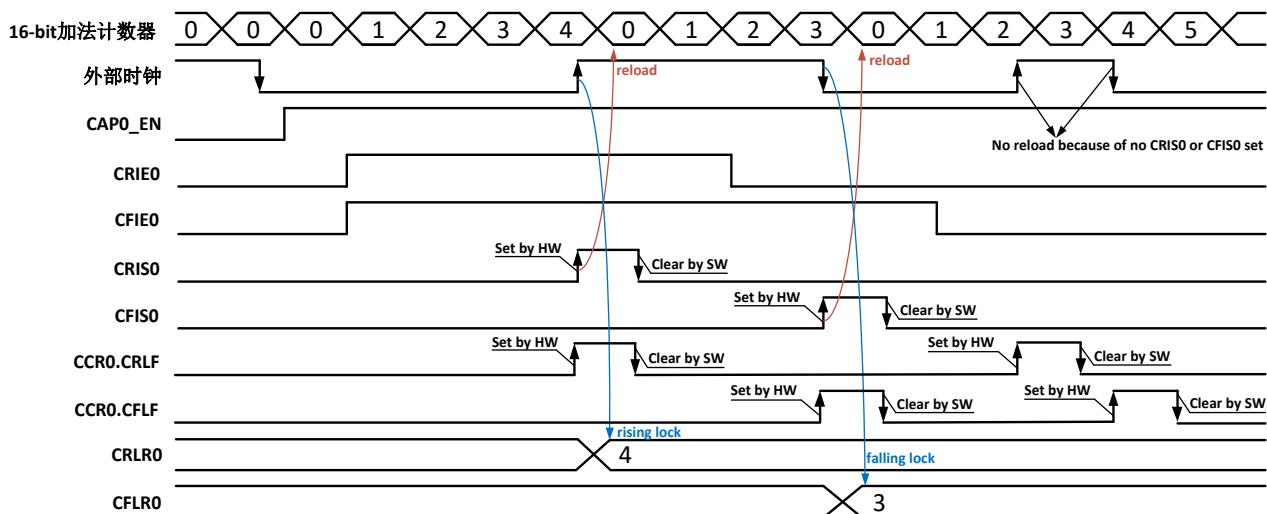


Figure 4-17 PWM01 对的 PWM0 通道捕捉时序

当 PWM 通道的捕捉输入功能被使能，如 PWM0 通道，PWM0 通道的加法计数器开始工作，PWM0 的定时器逻辑模块捕捉到 1 个上升沿，加法计数器当前值被锁存到 CRLR0，CCRO.CRLF 位被置 1，若 CRIEO = 1，CRISO 位被置 1，PWM0 通道发起中断请求，同时 CRISO 位被置 1 后加法计数器被重载为 0 并继续计数；若 CRIEO = 0，定时器逻辑模块捕捉到上升沿时 CRISO 不会被置 1，因此加法计数器不会被重载为 0。PWM0 的定时器逻辑模块捕捉到 1 个下降沿，加法计数器当前值被锁存到 CFLR0，CCRO.CFLF 位被置 1，若 CFIEO = 1，CFISO 位被置 1，PWM0 通道发起中断请求，同时 CFISO 位被置 1 后加法计数器被重载为 0 并继续计数；若 CFIEO = 0，定时器逻辑模块捕捉到下降沿时 CFISO 不会被置 1，因此加法计数器不会被重载为 0。

#### 4.5.2.6 Interrupt

PWM 通道被配置为 PWM 输出或捕捉输入时，都支持中断产生。对 PWM 输出功能，不管是周期模式还是单脉冲模式，只要加法计数器的值达到 PPR.PWM\_ENTIRE\_CYCLE 值，定时器逻辑模块都会自动将 PWM 通道的 PIS 位置 1，PWM 通道的 PIS 位由硬件自动置 1 但必须由软件清除；对捕捉输入功能，如 PWM0 通道被配置为捕捉功能，捕捉通道 0 的定时器逻辑模块捕捉到上升沿，若 CRIEO = 1，CRISO 置 1；捕捉通道 0 的定时器逻辑模块捕捉到下降沿，若 CFIEO = 1，CFISO 置 1；

#### 4.5.3 Register List

Module Name	Base Address	
PWM	0x40042800	
Register Name	Offset Address	Description
PIER	0x00	PWM IRQ Enable Register
PISR	0x04	PWM IRQ Status Register
CIER	0x10	Capture IRQ Enable Register
CISR	0x14	Capture IRQ Status Register
PCCR01	0x20	PWM01 Clock Configuration Register
PCCR23	0x24	PWM23 Clock Configuration Register

PCCR45	0x28	PWM45 Clock Configuration Register
PCCR67	0x2C	PWM67 Clock Configuration Register
PDZCR01	0x30	PWM01 Dead Zone Control Register
PDZCR23	0x34	PWM23 Dead Zone Control Register
PDZCR45	0x38	PWM45 Dead Zone Control Register
PDZCR67	0x3C	PWM67 Dead Zone Control Register
PER	0x40	PWM Enable Register
CER	0x44	Capture Enable Register
PCR	0x60+0x0+N*0x20(N= 0~7)	PWM Control Register
PPR	0x60+0x4+N*0x20(N= 0~7)	PWM Period Register
PCNTR	0x60+0x8+N*0x20(N= 0~7)	PWM Count Register
CCR	0x60+0xC+N*0x20(N= 0~7)	Capture Control Register
CRLR	0x60+0x10+N*0x20(N= 0~7)	Capture Rise Lock Register
CFLR	0x60+0x14+N*0x20(N= 0~7)	Capture Fall Lock Register

**Table 4-98 PWM/ECT Register List**

## 4.5.4 Register Description

### 4.5.4.1 PWM IRQ Enable Register

Address : 0x00			Name: PIER Default: 0x0000_0000
Field	Type	Default	Description
31:8	/	/	/
7	R/W	0x0	<b>PCIE7.</b> PWM channel 7 Interrupt Enable. 0: PWM channel 7 Interrupt Disable; 1: PWM channel 7 Interrupt Enable.
6	R/W	0x0	<b>PCIE6.</b> PWM channel 6 Interrupt Enable. 0: PWM channel 6 Interrupt Disable; 1: PWM channel 6 Interrupt Enable.
5	R/W	0x0	<b>PCIE5.</b> PWM channel 5 Interrupt Enable. 0: PWM channel 5 Interrupt Disable; 1: PWM channel 5 Interrupt Enable.
4	R/W	0x0	<b>PCIE4.</b> PWM channel 4 Interrupt Enable. 0: PWM channel 4 Interrupt Disable; 1: PWM channel 4 Interrupt Enable.
3	R/W	0x0	<b>PCIE3.</b>

			PWM channel 3 Interrupt Enable. 0: PWM channel 3 Interrupt Disable; 1: PWM channel 3 Interrupt Enable.
2	R/W	0x0	<b>PCIE2.</b> PWM channel 2 Interrupt Enable. 0: PWM channel 2 Interrupt Disable; 1: PWM channel 2 Interrupt Enable.
1	R/W	0x0	<b>PCIE1.</b> PWM channel 1 Interrupt Enable. 0: PWM channel 1 Interrupt Disable; 1: PWM channel 1 Interrupt Enable.
0	R/W	0x0	<b>PCIE0.</b> PWM channel 0 Interrupt Enable. 0: PWM channel 0 Interrupt Disable; 1: PWM channel 0 Interrupt Enable.

**Table 4-99 PWM IRQ Enable Register**

#### 4.5.4.2 PWM IRQ Status Register

Address : 0x04			Name: PISR  Default: 0x0000_0000
Field	Type	Default	Description
31:8	/	/	/
7	R/W1 C	0x0	PIS7. PWM channel 7 Interrupt Status. When PWM channel 7 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: PWM channel 7 interrupt is not pending. 1: PWM channel 7 interrupt is pending. Writes 0: no effect. 1: Clear PWM channel 7 interrupt status.
6	R/W1 C	0x0	PIS6. PWM channel 6 Interrupt Status. When PWM channel 6 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: PWM channel 6 interrupt is not pending. 1: PWM channel 6 interrupt is pending. Writes 0: no effect. 1: Clear PWM channel 6 interrupt status.
5	R/W1 C	0x0	PIS5. PWM channel 5 Interrupt Status. When PWM channel 5 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: PWM channel 5 interrupt is not pending. 1: PWM channel 5 interrupt is pending.

			Writes 0: no effect. 1: Clear PWM channel 5 interrupt status.
4	R/W1 C	0x0	PIS4. PWM channel 4 Interrupt Status. When PWM channel 4 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: PWM channel 4 interrupt is not pending. 1: PWM channel 4 interrupt is pending. Writes 0: no effect. 1: Clear PWM channel 4 interrupt status.
3	R/W1 C	0x0	PIS3. PWM channel 3 Interrupt Status. When PWM channel 3 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: PWM channel 3 interrupt is not pending. 1: PWM channel 3 interrupt is pending. Writes 0: no effect. 1: Clear PWM channel 3 interrupt status.
2	R/W1 C	0x0	PIS2. PWM channel 2 Interrupt Status. When PWM channel 2 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: PWM channel 2 interrupt is not pending. 1: PWM channel 2 interrupt is pending. Writes 0: no effect. 1: Clear PWM channel 2 interrupt status.
1	R/W1 C	0x0	PIS1. PWM channel 1 Interrupt Status. When PWM channel 1 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: PWM channel 1 interrupt is not pending. 1: PWM channel 1 interrupt is pending. Writes 0: no effect. 1: Clear PWM channel 1 interrupt status.
0	R/W1 C	0x0	PISO. PWM channel 0 Interrupt Status. When PWM channel 0 counter reaches Entire Cycle Value, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: PWM channel 0 interrupt is not pending. 1: PWM channel 0 interrupt is pending. Writes 0: no effect. 1: Clear PWM channel 0 interrupt status.

**Table 4-100 PWM IRQ Status Register**

#### 4.5.4.3 Capture IRQ Enable Register

<b>Address :</b> 0x10	<b>Name:</b> CIER
	<b>Default:</b> 0x0000_0000

Bit	R/W	Default/Hex	Description
31:16	/	/	/
15	R/W	0x0	<b>CFIE7.</b> If this enable bit is set 1, when capture channel 7 captures falling edge, it generates a capture channel 7 pending. 0: Capture channel 7 fall lock Interrupt disable; 1: Capture channel 7 fall lock Interrupt enable.
14	R/W	0x0	<b>CRIE7.</b> If this enable bit is set 1, when capture channel 7 captures rising edge, it generates a capture channel 7 pending. 0: Capture channel 7 rise lock Interrupt disable; 1: Capture channel 7 rise lock Interrupt enable.
13	R/W	0x0	<b>CFIE6.</b> If this enable bit is set 1, when capture channel 6 captures falling edge, it generates a capture channel 6 pending. 0: Capture channel 6 fall lock Interrupt disable; 1: Capture channel 6 fall lock Interrupt enable.
12	R/W	0x0	<b>CRIE6.</b> If this enable bit is set 1, when capture channel 6 captures rising edge, it generates a capture channel 6 pending. 0: Capture channel 6 rise lock Interrupt disable; 1: Capture channel 6 rise lock Interrupt enable.
11	R/W	0x0	<b>CFIE5.</b> If this enable bit is set 1, when capture channel 5 captures falling edge, it generates a capture channel 5 pending. 0: Capture channel 5 fall lock Interrupt disable; 1: Capture channel 5 fall lock Interrupt enable.
10	R/W	0x0	<b>CRIE5.</b> If this enable bit is set 1, when capture channel 5 captures rising edge, it generates a capture channel 5 pending. 0: Capture channel 5 rise lock Interrupt disable; 1: Capture channel 5 rise lock Interrupt enable.
9	R/W	0x0	<b>CFIE4.</b> If this enable bit is set 1, when capture channel 4 captures falling edge, it generates a capture channel 4 pending. 0: Capture channel 4 fall lock Interrupt disable; 1: Capture channel 4 fall lock Interrupt enable.
8	R/W	0x0	<b>CRIE4.</b> If this enable bit is set 1, when capture channel 4 captures rising edge, it generates a capture channel 4 pending. 0: Capture channel 4 rise lock Interrupt disable; 1: Capture channel 4 rise lock Interrupt enable.
7	R/W	0x0	<b>CFIE3.</b>

			If this enable bit is set 1, when capture channel 3 captures falling edge, it generates a capture channel 3 pending. 0: Capture channel 3 fall lock Interrupt disable; 1: Capture channel 3 fall lock Interrupt enable.
6	R/W	0x0	<b>CRIE3.</b> If this enable bit is set 1, when capture channel 3 captures rising edge, it generates a capture channel 3 pending. 0: Capture channel 3 rise lock Interrupt disable; 1: Capture channel 3 rise lock Interrupt enable.
5	R/W	0x0	<b>CFIE2.</b> If this enable bit is set 1, when capture channel 2 captures falling edge, it generates a capture channel 2 pending. 0: Capture channel 2 fall lock Interrupt disable; 1: Capture channel 2 fall lock Interrupt enable.
4	R/W	0x0	<b>CRIE2.</b> If this enable bit is set 1, when capture channel 2 captures rising edge, it generates a capture channel 2 pending. 0: Capture channel 2 rise lock Interrupt disable; 1: Capture channel 2 rise lock Interrupt enable.
3	R/W	0x0	<b>CFIE1.</b> If this enable bit is set 1, when capture channel 1 captures falling edge, it generates a capture channel 1 pending. 0: Capture channel 1 fall lock Interrupt disable; 1: Capture channel 1 fall lock Interrupt enable.
2	R/W	0x0	<b>CRIE1.</b> If this enable bit is set 1, when capture channel 1 captures rising edge, it generates a capture channel 1 pending. 0: Capture channel 1 rise lock Interrupt disable; 1: Capture channel 1 rise lock Interrupt enable.
1	R/W	0x0	<b>CFIE0.</b> If this enable bit is set 1, when capture channel 0 captures falling edge, it generates a capture channel 0 pending. 0: Capture channel 0 fall lock Interrupt disable; 1: Capture channel 0 fall lock Interrupt enable.
0	R/W	0x0	<b>CRIE0.</b> If this enable bit is set 1, when capture channel 0 captures rising edge, it generates a capture channel 0 pending. 0: Capture channel 0 rise lock Interrupt disable; 1: Capture channel 0 rise lock Interrupt enable.

**Table 4-101 Capture IRQ Enable Register**

#### 4.5.4.4 Capture IRQ Status Register

<b>Address:</b> 0x14			<b>Name:</b> CISR <b>Default:</b> 0x0000_0000
Bit	R/W	Default/Hex	Description
31:8	/	/	/
15	R/W1C	0x0	<p><b>CFIS7.</b></p> <p>Capture channel 7 falling lock interrupt status.</p> <p>When capture channel 7 captures falling edge, if capture channel 7 fall lock interrupt (CFIE7) is enabled, this bit is set 1 by hardware.</p> <p>Write 1 to clear this bit.</p> <p>Reads 0: Capture channel 7 interrupt is not pending.</p> <p>1: Capture channel 7 interrupt is pending.</p> <p>Writes 0: no effect.</p> <p>1: Clear capture channel 7 interrupt status.</p>
14	R/W1C	0x0	<p><b>CRIS7.</b></p> <p>Capture channel 7 rising lock interrupt status.</p> <p>When capture channel 7 captures rising edge, if capture channel 7 rise lock interrupt (CRIE7) is enabled, this bit is set 1 by hardware.</p> <p>Write 1 to clear this bit.</p> <p>Reads 0: Capture channel 7 interrupt is not pending.</p> <p>1: Capture channel 7 interrupt is pending.</p> <p>Writes 0: no effect.</p> <p>1: Clear capture channel 7 interrupt status.</p>
13	R/W1C	0x0	<p><b>CFIS6.</b></p> <p>Capture channel 6 falling lock interrupt status.</p> <p>When capture channel 6 captures falling edge, if capture channel 6 fall lock interrupt (CFIE6) is enabled, this bit is set 1 by hardware.</p> <p>Write 1 to clear this bit.</p> <p>Reads 0: Capture channel 6 interrupt is not pending.</p> <p>1: Capture channel 6 interrupt is pending.</p> <p>Writes 0: no effect.</p> <p>1: Clear capture channel 6 interrupt status.</p>
12	R/W1C	0x0	<p><b>CRIS6.</b></p> <p>Capture channel 6 rising lock interrupt status.</p> <p>When capture channel 6 captures rising edge, if capture channel 6 rise lock interrupt (CRIE6) is enabled, this bit is set 1 by hardware.</p> <p>Write 1 to clear this bit.</p> <p>Reads 0: Capture channel 6 interrupt is not pending.</p> <p>1: Capture channel 6 interrupt is pending.</p> <p>Writes 0: no effect.</p> <p>1: Clear capture channel 6 interrupt status.</p>

11	R/W1C	0x0	<b>CFIS5.</b> Capture channel 5 falling lock interrupt status. When capture channel 5 captures falling edge, if capture channel 5 fall lock interrupt (CFIE5) is enabled, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: Capture channel 5 interrupt is not pending. 1: Capture channel 5 interrupt is pending. Writes 0: no effect. 1: Clear capture channel 5 interrupt status.
10	R/W1C	0x0	<b>CRIS5.</b> Capture channel 5 rising lock interrupt status. When capture channel 5 captures rising edge, if capture channel 5 rise lock interrupt (CRIE5) is enabled, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: Capture channel 5 interrupt is not pending. 1: Capture channel 5 interrupt is pending. Writes 0: no effect. 1: Clear capture channel 5 interrupt status.
9	R/W1C	0x0	<b>CFIS4.</b> Capture channel 4 falling lock interrupt status. When capture channel 4 captures falling edge, if capture channel 4 fall lock interrupt (CFIE4) is enabled, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: Capture channel 4 interrupt is not pending. 1: Capture channel 4 interrupt is pending. Writes 0: no effect. 1: Clear capture channel 4 interrupt status.
8	R/W1C	0x0	<b>CRIS4.</b> Capture channel 4 rising lock interrupt status. When capture channel 4 captures rising edge, if capture channel 4 rise lock interrupt (CRIE4) is enabled, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: Capture channel 4 interrupt is not pending. 1: Capture channel 4 interrupt is pending. Writes 0: no effect. 1: Clear capture channel 4 interrupt status.
7	R/W1C	0x0	<b>CFIS3.</b> Capture channel 3 falling lock interrupt status. When capture channel 3 captures falling edge, if capture channel 3 fall lock interrupt (CFIE3) is enabled, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: Capture channel 3 interrupt is not pending. 1: Capture channel 3 interrupt is pending.

			Writes 0: no effect. 1: Clear capture channel 3 interrupt status.
6	R/W1C	0x0	<b>CRIS3.</b> Capture channel 3 rising lock interrupt status. When capture channel 3 captures rising edge, if capture channel 3 rise lock interrupt (CRIE3) is enabled, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: Capture channel 3 interrupt is not pending. 1: Capture channel 3 interrupt is pending. Writes 0: no effect. 1: Clear capture channel 3 interrupt status.
5	R/W1C	0x0	<b>CFIS2.</b> Capture channel 2 falling lock interrupt status. When capture channel 2 captures falling edge, if capture channel 2 fall lock interrupt (CFIE2) is enabled, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: Capture channel 2 interrupt is not pending. 1: Capture channel 2 interrupt is pending. Writes 0: no effect. 1: Clear capture channel 2 interrupt status.
4	R/W1C	0x0	<b>CRIS2.</b> Capture channel 2 rising lock interrupt status. When capture channel 2 captures rising edge, if capture channel 2 rise lock interrupt (CRIE2) is enabled, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: Capture channel 2 interrupt is not pending. 1: Capture channel 2 interrupt is pending. Writes 0: no effect. 1: Clear capture channel 2 interrupt status.
3	R/W1C	0x0	<b>CFIS1.</b> Capture channel 1 falling lock interrupt status. When capture channel 1 captures falling edge, if capture channel 1 fall lock interrupt (CFIE1) is enabled, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: Capture channel 1 interrupt is not pending. 1: Capture channel 1 interrupt is pending. Writes 0: no effect. 1: Clear capture channel 1 interrupt status.
2	R/W1C	0x0	<b>CRIS1.</b> Capture channel 1 rising lock interrupt status. When capture channel 1 captures rising edge, if capture channel 1 rise lock interrupt (CRIE1) is enabled, this bit is set 1 by hardware. Write 1 to clear this bit.

			Reads 0: Capture channel 1 interrupt is not pending. 1: Capture channel 1 interrupt is pending. Writes 0: no effect. 1: Clear capture channel 1 interrupt status.
1	R/W1C	0x0	<b>CFISO.</b> Capture channel 0 falling lock interrupt status. When capture channel 0 captures falling edge, if capture channel 0 fall lock interrupt (CFIE0) is enabled, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: Capture channel 0 interrupt is not pending. 1: Capture channel 0 interrupt is pending. Writes 0: no effect. 1: Clear capture channel 0 interrupt status.
0	R/W1C	0x0	<b>CRISO.</b> Capture channel 0 rising lock interrupt status. When capture channel 0 captures rising edge, if capture channel 0 rise lock interrupt (CRIE0) is enabled, this bit is set 1 by hardware. Write 1 to clear this bit. Reads 0: Capture channel 0 interrupt is not pending. 1: Capture channel 0 interrupt is pending. Writes 0: no effect. 1: Clear capture channel 0 interrupt status.

#### 4.5.4.5 PWM01 Clock Configuration Register

<b>Address:</b> 0x20			<b>Name:</b> PCCR01 <b>Default:</b> 0x0000_0000
Bit	R/W	Default/Hex	Description
31:9	/	/	/
8:7	R/W	0x00	<b>PWM01_CLK_SRC.</b> Select PWM01 clock source. 00: 24M_PLL 01: APB1 Others: /
6	R/W	0x0	<b>PWM01_CLK_SRC_BYPASS_TO_PWM1.</b> Bypass PWM01 clock source to PWM1 output. 0: not bypass 1: bypass
5	R/W	0x0	<b>PWM01_CLK_SRC_BYPASS_TO_PWM0.</b> Bypass PWM01 clock source to PWM0 output. 0: not bypass 1: bypass

4	R/W	0x0	<b>PWM01_CLK_GATING.</b> Gating clock for PWM01. 0: Mask 1: Pass
0:3	R/W	0x0000	<b>PWM01_CLK_DIV_M.</b> PWM01 clock divide M 0000: /1 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 others: /

Table 4-102 PWM01 Clock Configuration Register

#### 4.5.4.6 PWM23 Clock Configuration Register

<b>Address:</b> 0x24			<b>Name:</b> PCCR23 <b>Default:</b> 0x0000_0000
Bit	R/W	Default/Hex	Description
31:9	/	/	/
8:7	R/W	0x00	<b>PWM23_CLK_SRC_SEL.</b> Select PWM23 clock source. 00: 24M_PLL 01: APB1 Others: /
6	R/W	0x0	<b>PWM23_CLK_SRC_BYPASS_TO_PWM3.</b> Bypass PWM23 clock source to PWM3 output. 0: not bypass 1: bypass
5	R/W	0x0	<b>PWM23_CLK_SRC_BYPASS_TO_PWM2.</b> Bypass PWM23 clock source to PWM2 output. 0: not bypass 1: bypass
4	R/W	0x0	<b>PWM23_CLK_GATING</b> <b>Gating clock for PWM23.</b> 0: Mask 1: Pass
0:3	R/W	0x0000	<b>PWM23_CLK_DIV_M.</b>

			PWM23 clock divide M 0000: /1 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 others: /
--	--	--	--

Table 4-103 PWM23 Clock Configuration Register

#### 4.5.4.7 PWM45 Clock Configuration Register

Address: 0x28			Name: PCCR45 Default: 0x0000_0000
Bit	R/W	Default/Hex	Description
31:9	/	/	/
8:7	R/W	0x00	<b>PWM45_CLK_SRC_SEL.</b> Select PWM45 clock source. 00: 24M_PLL 01: APB1 Others: /
6	R/W	0x0	<b>PWM45_CLK_SRC_BYPASS_TO_PWM5.</b> Bypass PWM45 clock source to PWM5 output. 0: not bypass 1: bypass
5	R/W	0x0	<b>PWM45_CLK_SRC_BYPASS_TO_PWM4.</b> Bypass PWM45 clock source to PWM4 output. 0: not bypass 1: bypass
4	R/W	0x0	<b>PWM45_CLK_GATING.</b> Gating clock for PWM45. 0: Mask 1: Pass
0:3	R/W	0x0000	<b>PWM45_CLK_DIV_M.</b> PWM45 clock divide M 0000: /1 0001: /2 0010: /4 0011: /8

			0100: /16 0101: /32 0110: /64 0111: /128 1000: /256 others: /
--	--	--	--

**Table 4-104 PWM45 Clock Configuration Register**

#### 4.5.4.8 PWM67 Clock Configuration Register

<b>Address:</b> 0x2C			<b>Name:</b> PCCR67 <b>Default:</b> 0x0000_0000
Bit	R/W	Default/Hex	Description
31:9	/	/	/
8:7	R/W	0x00	<b>PWM67_CLK_SRC_SEL.</b> Select PWM67 clock source. 00: 24M_PLL 01: APB1 Others: /
6	R/W	0x0	<b>PWM67_CLK_SRC_BYPASS_TO_PWM7.</b> Bypass PWM67 clock source to PWM7 output. 0: not bypass 1: bypass
5	R/W	0x0	<b>PWM67_CLK_SRC_BYPASS_TO_PWM6.</b> Bypass PWM67 clock source to PWM6 output. 0: not bypass 1: bypass
4	R/W	0x0	<b>PWM67_CLK_GATING.</b> Gating clock for PWM67. 0: Mask 1: Pass
0:3	R/W	0x0000	<b>PWM67_CLK_DIV_M.</b> PWM67 clock divide M 0000: /1 0001: /2 0010: /4 0011: /8 0100: /16 0101: /32 0110: /64 0111: /128 1000: /256

			others: /
--	--	--	-----------

**Table 4-105 PWM67 Clock Configuration Register**

#### 4.5.4.9 PWM01 Dead Zone Control Register

<b>Address:</b> 0x30			<b>Name:</b> PDZCR01 <b>Default:</b> 0x0000_0000
Bit	R/W	Default/Hex	Description
15:8	R/W	0x0	<b>PDZINTV01.</b> PWM01 Dead Zone interval value.
7:1	/	/	/
0	R/W	0x0	<b>PWM01_DZ_EN.</b> PWM01 Dead Zone enable. 0: Dead Zone disable 1: Dead Zone enable.

**Table 4-106 PWM01 Dead Zone Control Register**

#### 4.5.4.10 PWM23 Dead Zone Control Register

<b>Address:</b> 0x34			<b>Name:</b> PDZCR23 <b>Default:</b> 0x0000_0000
Bit	R/W	Default/Hex	Description
15:8	R/W	0x0	<b>PWM23_DZ_INTV.</b> PWM23 Dead Zone interval value.
7:1	/	/	/
0	R/W	0x0	<b>PWM23_DZ_EN.</b> PWM23 Dead Zone enable. 0: Dead Zone disable 1: Dead Zone enable.

**Table 4-107 PWM23 Dead Zone Control Register**

#### 4.5.4.11 PWM45 Dead Zone Control Register

<b>Address:</b> 0x38			<b>Name:</b> PDZCR45 <b>Default:</b> 0x0000_0000
Bit	R/W	Default/Hex	Description
15:8	R/W	0x0	<b>PWM45_DZ_INTV.</b> PWM45 Dead Zone interval value.
7:1	/	/	/
0	R/W	0x0	<b>PWM45_DZ_EN.</b> PWM45 Dead Zone enable. 0: Dead Zone disable

		1: Dead Zone enable.
--	--	----------------------

**Table 4-108 PWM45 Dead Zone Control Register**

#### 4.5.4.12 PWM67 Dead Zone Control Register

<b>Address:</b> 0x3C			<b>Name:</b> PDZCR67 <b>Default:</b> 0x0000_0000
Bit	R/W	Default/Hex	Description
15:8	R/W	0x0	<b>PWM67_DZ_INTV.</b> PWM67 Dead Zone interval value.
7:1	/	/	/
0	R/W	0x0	<b>PWM67_DZ_EN.</b> PWM67 Dead Zone enable. 0: Dead Zone disable 1: Dead Zone enable.

**Table 4-109 PWM67 Dead Zone Control Register**

#### 4.5.4.13 PWM Enable Register

<b>Address:</b> 0x40			<b>Name:</b> PER <b>Default:</b> 0x0000_0000
Bit	R/W	Default/Hex	Description
31:1	/	/	/
7	R/W	0x0	<b>PWM7_EN.</b> When enable PWM, the 16-bit up-counter starts working and PWM channel7 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable.
6	R/W	0x0	<b>PWM6_EN.</b> When enable PWM, the 16-bit up-counter starts working and PWM channel6 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable.
5	R/W	0x0	<b>PWM5_EN.</b> When enable PWM, the 16-bit up-counter starts working and PWM channel5 is permitted to output PWM waveform. 0: PWM disable 1: PWM enable.
4	R/W	0x0	<b>PWM4_EN.</b> When enable PWM, the 16-bit up-counter starts working and PWM channel4 is permitted to output PWM waveform. 0: PWM disable

			1: PWM enable.
3	R/W	0x0	<p><b>PWM3_EN.</b></p> <p>When enable PWM, the 16-bit up-counter starts working and PWM channel3 is permitted to output PWM waveform.</p> <p>0: PWM disable 1: PWM enable.</p>
2	R/W	0x0	<p><b>PWM2_EN.</b></p> <p>When enable PWM, the 16-bit up-counter starts working and PWM channel2 is permitted to output PWM waveform.</p> <p>0: PWM disable 1: PWM enable.</p>
1	R/W	0x0	<p><b>PWM1_EN.</b></p> <p>When enable PWM, the 16-bit up-counter starts working and PWM channel1 is permitted to output PWM waveform.</p> <p>0: PWM disable 1: PWM enable.</p>
0	R/W	0x0	<p><b>PWM0_EN.</b></p> <p>When enable PWM, the 16-bit up-counter starts working and PWM channel0 is permitted to output PWM waveform.</p> <p>0: PWM disable 1: PWM enable.</p>

Table 4-110 PWM Enable Register

#### 4.5.4.14 Capture Enable Register

<b>Address:</b> 0x44			<b>Name:</b> CER <b>Default:</b> 0x0000_0000
Bit	R/W	Default/Hex	Description
31:8	/	/	/
7	R/W	0x0	<p><b>CAP7_EN.</b></p> <p>When enable capture function, the 16-bit up-counter starts working and capture channel7 is permitted to capture external falling edge or rising edge.</p> <p>0: Capture disable 1: Capture enable.</p>
6	R/W	0x0	<p><b>CAP6_EN.</b></p> <p>When enable capture function, the 16-bit up-counter starts working and capture channel6 is permitted to capture external falling edge or rising edge.</p> <p>0: Capture disable 1: Capture enable.</p>
5	R/W	0x0	<b>CAP5_EN.</b>

			When enable capture function, the 16-bit up-counter starts working and capture channel5 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable.
4	R/W	0x0	<b>CAP4_EN.</b> When enable capture function, the 16-bit up-counter starts working and capture channel4 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable.
3	R/W	0x0	<b>CAP3_EN.</b> When enable capture function, the 16-bit up-counter starts working and capture channel3 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable.
2	R/W	0x0	<b>CAP2_EN.</b> When enable capture function, the 16-bit up-counter starts working and capture channel2 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable.
1	R/W	0x0	<b>CAP1_EN.</b> When enable capture function, the 16-bit up-counter starts working and capture channel1 is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable.
0	R/W	0x0	<b>CAP0_EN.</b> When enable capture function, the 16-bit up-counter starts working and capture channel is permitted to capture external falling edge or rising edge. 0: Capture disable 1: Capture enable.

Table 4-111 Capture Enable Register

#### 4.5.4.15 PWM Control Register

<b>Address:</b> 0x60+0x0+N*0x20(N= 0~7)		<b>Name:</b> PCR	
<b>Default:</b> 0x0000_0000			
Bit	R/W	Default/Hex	Description

31:13	/	/	/.
11	R	0x0	<b>PWM_PERIOD_RDY.</b> PWM period register ready. 0: PWM period register is ready to write 1: PWM period register is busy.
10	R/W1S	0x0	<b>PWM_PUL_START.</b> PWM pulse output start. 0: no effect 1: output 1 pulse. After finishing configuring for outputting pulse, set this bit once and then PWM would output one pulse. After the pulse is finished, the bit will be cleared automatically.
9	R/W	0x0	<b>PWM_MODE.</b> PWM output mode select. 0: cycle mode 1: pulse mode.
8	R/W	0x0	<b>PWM_ACT_STA.</b> PWM active state. 0: Low Level 1: High Level.
7:0	R/W	0x0	<b>PWM_PRESCAL_K.</b> PWM pre-scale K, actual pre-scale is (K+1). K = 0, actual pre-scale: 1 K = 1, actual pre-scale: 2 K = 2, actual pre-scale: 3 K = 3, actual pre-scale: 4 ..... K = 255, actual pre-scale: 256.

Table 4-112 PWM Control Register

#### 4.5.4.16 PWM Period Register

<b>Address:</b> 0x60+0x4+N*0x20(N= 0~7)			<b>Name:</b> PPR <b>Default:</b> 0x0000_0000
Bit	R/W	Default/Hex	Description
31:16	R/W	x	<b>PWM_ENTIRE_CYCLE.</b> Number of the entire cycles in the PWM clock. 0 = 1 cycle 1 = 2 cycles ..... N = N+1 cycles If the register need to be modified dynamically, the PCLK should be

			faster than the PWM CLK .
15:0	R/W	x	<p><b>PWM_ACT_CYCLE.</b></p> <p>Number of the active cycles in the PWM clock.</p> <p>0 = 0 cycle</p> <p>1 = 1 cycles</p> <p>.....</p> <p>N = N cycles</p>

**Table 4-113 PWM Period Control Register****4.5.4.17 PWM Counter Register**

<b>Address:</b> 0x60+0x8+N*0x20(N= 0~7)			<b>Name:</b> PCNTR <b>Default:</b> 0x0000_0000
Bit	R/W	Default/Hex	Description
31:16	/	/	/.
15:0	R	x	On PWM output or capture input, reading this register could get the current value of the PWM 16bit up-counter.

**Table 4-114 PWM Counter Register****4.5.4.18 Capture Control Register**

<b>Address:</b> 0x60+0xC+N*0x20(N= 0~7)			<b>Name:</b> CCR <b>Default:</b> 0x0000_0000
Bit	R/W	Default/Hex	Description
31:3	/	/	/.
2	R/W	0x0	<p><b>CRLF.</b></p> <p>When capture channel captures rising edge, the 16-bit up-counter's current value is latched to CRLR and then this bit is set 1 by hardware.</p> <p>Write 1 to clear this bit.</p>
1	R/W	0x0	<p><b>CFLF.</b></p> <p>When capture channel captures falling edge, the 16-bit up-counter's current value is latched to CFLR and then this bit is set 1 by hardware.</p> <p>Write 1 to clear this bit.</p>
0	R/W	0x0	<p><b>CAPINV.</b></p> <p>Inversing the signal inputted from capture channel before capture channel's 16bit counter.</p> <p>0: not inverse</p> <p>1: inverse</p>

**Table 4-115 Capture Control Register**

#### 4.5.4.19 Capture Rise Lock Register

<b>Address:</b> 0x60+0x10+N*0x20(N= 0~7)			<b>Name:</b> CRLR <b>Default:</b> 0x0000_0000
Bit			Description
31:16	/	/	/
15:0	R	x	When capture channel captures rising edge, the 16-bit up-counter's current value is latched to this register.

Table 4-116 Capture Rise Lock Register

#### 4.5.4.20 Capture Fall Lock Register

<b>Address:</b> 0x60+0x14+N*0x20(N= 0~7)			<b>Name:</b> CFLR <b>Default:</b> 0x0000_0000
Bit			Description
31:16	/	/	/
15:0	R	x	When capture channel captures falling edge, the 16-bit up-counter's current value is latched to this register.

Table 4-117 Capture Fall Lock Register

### 4.6 CIR Receiver Controller (IRRX)

#### 4.6.1 Overview

The CIR includes the following features:

- Full physical layer implementation
- Support CIR for remote control
- 64x8bits FIFO for data buffer
- Programmable FIFO thresholds

For saving CPU resource, CIR receiver is implemented in hardware. The CIR receiver samples the input signal on the programmable frequency and records these samples into RX FIFO when one CIR signal is found on the air. The CIR receiver uses Run-Length Code (RLC) to encode pulse width. The encoded data is buffered in a 64 levels and 8-bit width RX FIFO; the MSB bit is used to record the polarity of the receiving CIR signal. The high level is represented as '1' and the low level is represented as '0'. The rest 7 bits are used for the length of RLC. The maximum length is 128. If the duration of one level (high or low level) is more than 128, another byte is used.

In the air, there is always some noise. One threshold can be set to filter the noise to reduce system loading and improve the system stability.

## 4.6.2 Register List

Module Name	Base Address	
CIR Receiver	0x40043800	
Register Name	Offset Address	Description
CIR_ACTL	0x00	CIR Active Control Register
CIR_CTL	0x04	CIR Control Register
CIR_RXCTL	0x10	CIR Receiver Configure Register
CIR_RXFIFO	0x20	CIR Receiver FIFO Register
CIR_RXINT	0x2C	CIR Receiver Interrupt Control Register
CIR_RXSTA	0x30	CIR Receiver Status Register
CIR_CONFIG	0x34	CIR Configure Register

Table 4-118 CIR Receiver Register List

## 4.6.3 Register Description

### 4.6.3.1 CIR Active Pulse Accept Mode Control Register

<b>Address : 0x0000</b>			<b>Name: CIR_ACTL</b> <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:8	/	/	/
			Active Pulse Accept Mode 00: Start to accept data after detect de-bounce pulse edge 01: Start to accept data after detect de-bounce pulse edge 10: Start to accept data after detect low voltage de-bounce pulse 11: Start to accept data after detect high voltage de-bounce pulse
7:6	R/W	R/W	
5:0	/	/	/.

Table 4-119 CIR Active Pulse Accept Mode Control Register

### 4.6.3.2 CIR Receiver Control Register

<b>Address : 0x0004</b>			<b>Name: CIR_CTL</b> <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:9	/	/	/
8	R/W	0x0	CGPO General Program Output (GPO) Control in CIR mode for TX Pin 0: Low level 1: High level

7:6	/	/	/
5:4	R/W	0x0	CIR ENABLE 00~10: Reserved 11: CIR mode enable
3:2	/	/	/.
1	R/W	0x0	RXEN Receiver Block Enable 0: Disable 1: Enable
0	R/W	0x0	GEN Global Enable A disable on this bit overrides any other block or channel enables and flushes all FIFOs. 0: Disable 1: Enable

**Table 4-120 CIR Receiver Control Register**

#### 4.6.3.3 CIR Receiver Configure Register

<b>Address :</b> 0x0010			<b>Name:</b> IR_RXCTL <b>Default:</b> 0x0000_0004
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:3	/	/	/
2	R/W	0x1	RPPI Receiver Pulse Polarity Invert 0: Not invert receiver signal 1: Invert receiver signal
1:0	/	/	/

**Table 4-121 CIR Receiver Configure Register**

#### 4.6.3.4 CIR Receiver FIFO Register

<b>Address :</b> 0x0020			<b>Name:</b> IR_RXFIFO <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	R	0x0	Receiver Byte FIFO

**Table 4-122 CIR Receiver FIFO Register**

#### 4.6.3.5 CIR Receiver Interrupt Control Register

<b>Address :</b> 0x002C	<b>Name:</b> IR_RXINT
-------------------------	-----------------------

			<b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:14	/	/	/
13:8	R/W	0x0	<p><b>RAL</b>            RX FIFO Available Received Byte Level for interrupt and DMA request            TRIGGER_LEVEL = RAL + 1</p>
5	R/W	0x0	<p><b>DRQ_EN</b>            RX FIFO DMA Enable            0: Disable            1: Enable            When set to '1', the Receiver FIFO DRQ is asserted if reaching RAL. The DRQ is de-asserted when condition fails.</p>
4	R/W	0x0	<p><b>RAI_EN</b>            RX FIFO Available Interrupt Enable            0: Disable            1: Enable            When set to '1', the Receiver FIFO IRQ is asserted if reaching RAL. The IRQ is de-asserted when condition fails.</p>
3:2	/	/	/
1	R/W	0x0	<p><b>RPEI_EN</b>            Receiver Packet End Interrupt Enable            0: Disable            1: Enable</p>
0	R/W	0x0	<p><b>ROI_EN</b>            Receiver FIFO Overrun Interrupt Enable            0: Disable            1: Enable</p>

**Table 4-123 CIR Receiver Interrupt Control Register**

#### 4.6.3.6 CIR Receiver Status Register

<b>Address :</b> 0x0030			<b>Name:</b> IR_RXSTA
			<b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:15	/	/	/
14:8	R	0x0	<p><b>RAC</b>            RX FIFO Available Counter            0: No available data in RX FIFO            1: 1 byte available data in RX FIFO            2: 2 byte available data in RX FIFO            ...            64: 64 byte available data in RX FIFO</p>

7	R	0x0	STAT Status of CIR 0x0 – Idle 0x1 – busy
6:5	/	/	/
4	R/W	0x0	RA RX FIFO Available 0: RX FIFO not available according its level 1: RX FIFO available according its level This bit is cleared by writing a '1'.
3:2	/	/	/
1	R/W	0x0	RPE Receiver Packet End Flag 0: STO was not detected. In CIR mode, one CIR symbol is receiving or not detected. 1: STO field or packet abort symbol (7'b0000,000 and 8'b0000,0000 for MIR and FIR) is detected. In CIR mode, one CIR symbol is received. This bit is cleared by writing a '1'.
0	R/W	0x0	ROI Receiver FIFO Overrun 0: Receiver FIFO not overrun 1: Receiver FIFO overrun This bit is cleared by writing a '1'.

**Table 4-124 CIR Receiver Status Register**

#### 4.6.3.7CIR Receiver Controller Configure Register

Address : 0x0004			Name: IR_CIR Default: 0x0000_1828
Field	Type	Default	Description
31:25	/	/	/
24	R/W	0x0	SCS2 Bit2 of Sample Clock Select for CIR This bit is defined by SCS bits below.
23	R/W	0x0	ATHC Active Threshold Control for CIR 0x0 –ATHR in Unit of (Sample Clock) 0x1 –ATHR in Unit of (128*Sample Clocks)
22:16	R/W	0x0	ATHR Active Threshold for CIR These bits control the duration of CIR from Idle to Active State. The duration can be calculated by ((ATHR + 1)*(ATHC? Sample Clock:

			128*Sample Clock)).																																				
15:8	R/W	0x18	<p>ITHR Idle Threshold for CIR The Receiver uses it to decide whether the CIR command has been received. If there is no CIR signal on the air, the receiver is staying in IDLE status. One active pulse will bring the receiver from IDLE status to Receiving status. After the CIR is end, the inputting signal will keep the specified level (high or low level) for a long time. The receiver can use this idle signal duration to decide that it has received the CIR command. The corresponding flag is asserted. If the corresponding interrupt is enable, the interrupt line is asserted to CPU.</p> <p>When the duration of signal keeps one status (high or low level) for the specified duration ( (ITHR + 1)*128 sample_clk ), this means that the previous CIR command has been finished.</p>																																				
7:2	R/W	0xa	<p>NTHR Noise Threshold for CIR When the duration of signal pulse (high or low level) is less than NTHR, the pulse is taken as noise and should be discarded by hardware.</p> <p>0: all samples are recorded into RX FIFO</p> <p>1: If the signal is only one sample duration, it is taken as noise and discarded.</p> <p>2: If the signal is less than (<math>\leq</math>) two sample duration, it is taken as noise and discarded.</p> <p>...</p> <p>61: if the signal is less than (<math>\leq</math>) sixty-one sample duration, it is taken as noise and discarded.</p>																																				
1:0	R/W	0x0	<p>SCS Sample Clock Select for CIR</p> <table border="1"> <thead> <tr> <th>SCS2</th> <th>SCS[1]</th> <th>SCS[0]</th> <th>Sample Clock</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>ir_clk/64</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>ir_clk/128</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>ir_clk/256</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>ir_clk/512</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>ir_clk</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>Reserved</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>Reserved</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>Reserved</td> </tr> </tbody> </table>	SCS2	SCS[1]	SCS[0]	Sample Clock	0	0	0	ir_clk/64	0	0	1	ir_clk/128	0	1	0	ir_clk/256	0	1	1	ir_clk/512	1	0	0	ir_clk	1	0	1	Reserved	1	1	0	Reserved	1	1	1	Reserved
SCS2	SCS[1]	SCS[0]	Sample Clock																																				
0	0	0	ir_clk/64																																				
0	0	1	ir_clk/128																																				
0	1	0	ir_clk/256																																				
0	1	1	ir_clk/512																																				
1	0	0	ir_clk																																				
1	0	1	Reserved																																				
1	1	0	Reserved																																				
1	1	1	Reserved																																				

**Table 4-125 CIR Receiver Controller Configure Register**

## 4.7 CIR Transmit Controller (IRTX)

### 4.7.1 Overview

The CIR includes the following features:

- Supports industry-standard AMBA Peripheral Bus (APB) and it is fully compliant with the AMBA Specification, Revision 2.0
- Full physical layer implementation
- 128 bytes FIFO for data buffer
- Configurable carrier frequency
- Interrupt and DMA Support
- Support DMA shake and wait mode

The CIR mode uses a variable pulse-width modulation technique to encompass the various formats of infrared encoding for remote-control applications.

Modulation (example: NEC protocol): the logic “1” takes 2.25ms to transmit, while a logical “0” is only 1.12ms.

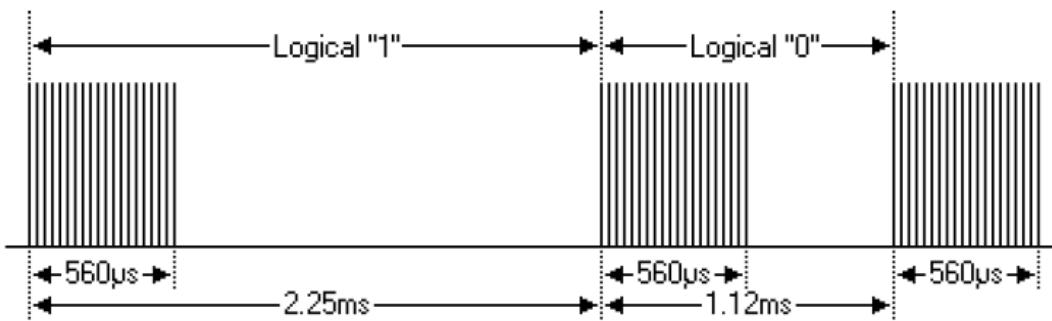


Figure 4-18 CIR Transmit Modulation

Protocol (example: NEC protocol):

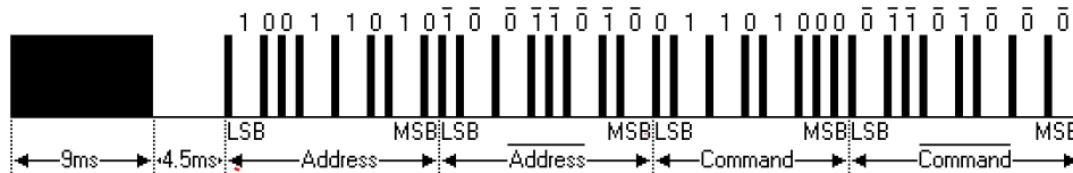


Figure 4-19 CIR Transmit Protocol Example

### 4.7.2 Register List

Module Name	Base Address	
CIR Transmitter	0x40043400	
Register Name	Offset Address	Description
CIR_TGLR	0x00	CIR Transmit Global Register

CIR_TMCR	0x04	CIR Transmit Modulation Control Register
CIR_TCR	0x08	CIR Transmit Control Register
CIR_IDC_H	0x0C	CIR Transmit Idle Duration threshold Register
CIR_IDC_L	0x10	CIR Transmit Idle Duration threshold Register
CIR_TICR_H	0x14	CIR Transmit Idle Counter Register
CIR_TICR_L	0x18	CIR Transmit Idle Counter Register
CIR_TEL	0x20	CIR TX FIFO empty Level Register
CIR_TXINT	0x24	CIR Transmit Interrupt Control Register
CIR_TAC	0x28	CIR Transmit FIFO Available Counter Register
CIR_TXSTA	0x2C	CIR Transmit Status Register
CIR_TXT	0x30	CIR Transmit Threshold Register
CIR_DMA	0x34	CIR DMA Control Register
CIR_TXFIFO	0x80	CIR Transmit FIFO Data Register

**Table 4-126 CIR Transmitter Register List**

## 4.7.3 Register Description

### 4.7.3.1 CIR Transmit Global Register

Address : 0x0000			Name: CIR_TGLR Default: 0x0000_0000
Field	Type	Default	Description
31:8	/	/	/
7	R/W	0	IMS Internal Modulation Select 0: the transmitting signal is not modulated 1: the transmitting signal is modulated internally
6:5	R/W	0	DRMC duty ratio of modulated carrier is high level /low level. <b>0: low level is the one time of high level</b> <b>1: low level is the two times of high level</b> <b>2: low level is the three times of high level</b> 3: reserved
4:3	/	/	/
2	R/W	0	TPPI Transmit Pulse Polarity Invert 0: Not invert transmit pulse 1: Invert transmit pulse
1	R/W	0	TR Transmit Reset When this bit is set, the transmitting is reset. The FIFO will be flush, the

			TIC filed and CSS field will be clean during Transmit Reset. This field will automatically clean when the Transmit Reset is finished, and the CIR transmitter will state Idle .
0	R/W	0	<b>TXEN</b> Transmit Block Enable 0: Disable the CIR Transmitter 1: Enable the CIR Transmitter

**Table 4-127 CIR Transmit Global Register**

#### 4.7.3.2CIR Transmit Modulation Control Register

<b>Address : 0x0004</b>			<b>Name: CIR_TMCR</b> <b>Default: 0x0000_009e</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	R/W	0x9E	<b>RFMC</b> Reference Frequency of modulated carrier. Reference Frequency of modulated carrier based on a division of a fixed functional clock(FCLK).The range of the modulated carrier is usually 30KHZ to 60KHz.The most consumer electronics is 38Khz. The default modulated carrier is 38Khz when FCLK is 6MHz. RFMC= FCLK/N.

**Table 4-128 CIR Transmit Modulation Control Register**

#### 4.7.3.3CIR Transmit Control Register

<b>Address : 0x0008</b>			<b>Name: CIR_TCR</b> <b>Default: 0x0000_0002</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7	R/W	0	<b>CSS</b> Cyclical Pulse Start/Stop Control Start to transmit when set to '1', 0: Stop when cleared to '0'. From start to stop, all data in FIFO must be transmitted. 1: Start.
6:4	/	/	/
3:1	R/W	1	<b>RCS</b> Reference Clock Select for CIR Transmit( $\geq 1$ ) 001: CIR Transmit reference clock is ir_clk/2 010: CIR Transmit reference clock is ir_clk/4

			011: CIR Transmit reference clock is ir_clk/8 100: CIR Transmit reference clock is ir_clk/64 101: CIR Transmit reference clock is ir_clk/128 110: CIR Transmit reference clock is ir_clk/256 111: CIR Transmit reference clock is ir_clk/512
0	R/W	0	TTS Type of the transmission signal 0: The transmitting wave is single non-cyclical pulse. 1: The transmitting wave is cyclical short-pulse.

**Table 4-129 CIR Transmit Control Register**

#### 4.7.3.4CIR Transmit Idle Duration Counter Register

<b>Address :</b> 0x000c			<b>Name:</b> CIR_IDC_H <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/

			IDC_H Idle Duration Counter threshold(High 4 bit)
3:0	R/W	0	Idle Duration = 128*IDC*Ts (IDC = 0~4095)

**Table 4-130 CIR Transmit Idle Duration Counter Register**

#### 4.7.3.5CIR Transmit Idle Duration Counter Register

<b>Address :</b> 0x0010			<b>Name:</b> CIR_IDC_L <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/

			IDC_L Idle Duration Counter threshold(Low 8 bit)
7:0	R/W	0	Idle Duration = 128*IDC*Ts (IDC = 0~4095)

**Table 4-131 CIR Transmit Idle Duration Counter Register**

#### 4.7.3.6CIR Transmit Idle Counter Register

<b>Address :</b> 0x0014			<b>Name:</b> CIR_TICR_H <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/

			TIC_H Transmit Idle Counter_H(High 8 bit)
7:0	R/W	0	Count in 128*Ts (Sample Duration, 1/Fs) when transmitter is idle, and it

			should be reset when the transmitter active. When this counter reaches the maximum value (0xFFFF), it will stop automatically, and should not be cleared to zero.
--	--	--	--

**Table 4-132 CIR Transmit Idle Counter Register**

#### 4.7.3.7CIR Transmit Idle Counter Register

<b>Address :</b> 0x0018			<b>Name:</b> CIR_TICR_L <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	R/W	0	TIC_L Transmit Idle Counter_L(Low 8 bit) Count in 128*Ts (Sample Duration, 1/Fs) when transmitter is idle, and it should be reset when the transmitter active. When this counter reach the maximum value (0xFFFF), it will stop automatically, and should not be cleared to zero.

**Table 4-133 CIR Transmit Idle Counter Register**

#### 4.7.3.8CIR Transmit FIFO Empty Register

<b>Address :</b> 0x0020			<b>Name:</b> CIR_TEL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	R/W	0	TEL TX FIFO empty Level for DRQ and IRQ. TRIGGER_LEVEL = TEL + 1

**Table 4-134 CIR Transmit FIFO Empty Register**

#### 4.7.3.9CIR Transmit Interrupt Control Register

<b>Address :</b> 0x0024			<b>Name:</b> CIR_TXINT <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:3	/	/	/
2	R/W	0	DRQ_EN TX FIFO DMA Enable 0: Disable 1: Enable When set to '1', the Tx FIFO DRQ is asserted if the number of the transmitting data in the FIFO is less than the RAL. The DRQ is de-asserted

			when condition fails.
1	R/W	0	TAI_EN TX FIFO Available Interrupt Enable 0:Disable 1:Enable
0	R/W	0	TPEI_EN Transmit Packet End Interrupt Enable for Cyclical Pulse 0:Disable 1:Enable  TUI_EN Transmitter FIFO under run Interrupt Enable for Non-cyclical Pulse 0: Disable 1: Enable

**Table 4-135 CIR Transmit Interrupt Control Register**

#### 4.7.3.10 CIR Transmit FIFO Available Counter Register

<b>Address :</b> 0x0028			<b>Name:</b> CIR_TAC <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:6	/	/	/
7:0	R	128	TAC TX FIFO Available Space Counter 0: No available space in TX FIFO 1: 1 byte available space in TX FIFO 2: 2 byte available space in TX FIFO ... 128: 128 byte available space in TX FIFO

**Table 4-136 CIR Transmit FIFO Available Counter Register**

#### 4.7.3.11 CIR Transmit Status Register

<b>Address :</b> 0x002c			<b>Name:</b> CIR_TXSTA <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:4	/	/	/
3	R	0	STCT Status of CIR Transmitter 0x0:Idle 0x1:Active

			This bit will automatically set when the controller begins transmit the data in the FIFO. The “1” will last when the data in the FIFO. It will automatically be cleaned to “0” when all data in the FIFO is transmitted. The bit is for debug.
2	R	0	DRQ DMA Request Flag When set to ‘1’, the Tx FIFO DRQ is asserted if the number of the transmitting data in the FIFO is less than the RAL. The DRQ is de-asserted when condition fails. This bit is for debug.
1	R/W	0	TAI TX FIFO Available Interrupt Flag 0: TX FIFO not available by its level 1: TX FIFO available by its level This bit is cleared by writing a ‘1’.
0	R/W	0	TPE Transmitter Packet End Flag for Cyclical Pulse 0: Transmissions of address, control and data fields not completed 1: Transmissions of address, control and data fields completed  TUR Transmitter FIFO Under Run Flag for Non-cyclical Pulse 0: No transmitter FIFO under run 1: Transmitter FIFO under run  This bit is cleared by writing a ‘1’.

**Table 4-137 CIR Transmit Status Register**

#### 4.7.3.12 CIR Transmit Threshold Register

<b>Address : 0x0030</b>			<b>Name: CIR_TXT</b> <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	R/W	0	<b>NCTT</b> <b>Non-cyclical Pulse Transmit Threshold</b> The controller will trigger transmitting the data in the FIFO when the data byte number has reach the Transmit Threshold set in this field.

**Table 4-138 CIR Transmit Threshold Register**

#### 4.7.3.13 CIR DMA Control Register

<b>Address :</b> 0x0034			<b>Name:</b> CIR_DMA_CTL <b>Default:</b> 0x0000_00A5
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:6	R/W	2	00:dma_active is low 01:dma_active is high 10:dma_req 11:active is control by controller
5	R/W	1	0: active fall do not care ack 1: active fall must after detect ack is high
4:0	R/W	5	dma_active high after last signal is high

**Table 4-139 CIR DMA Control Register**

#### 4.7.3.14 CIR Transmit FIFO Data Register

<b>Address :</b> 0x0080			<b>Name:</b> CIR_TXFIFO <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
7:0	W	0	Transmit Byte FIFO When the transmitting is trigger, the data in the FIFO will be transmitted until the data number has been transmitted finished.

**Table 4-140 CIR Transmit FIFO Data Register**

### 4.8 General Purpose ADC (GPADC)

#### 4.8.1 Overview

The GPADC is a 12-bit sampling analog to digital converter with 9 channels multiplexer. This ADC is a type of successive approximation register (SAR) converter.

- 12-bit Resolution and 10-bit effective SAR type A/D converter
- 9-channel multiplexer and the 9<sup>th</sup> channel is used for VBAT voltage detection
- 64 FIFO depth of data register
- Power Supply Voltage: 2.5V, Analog Input Range: 0 to 2.5V
- Maximum Sampling frequency: 1 MHz
- Support self-calibration
- Support data compare and interrupt
- Support four operation mode
  - Single conversion mode

- Single-cycle conversion mode
- Continuous conversion mode
- Outbreak conversion mode

## 4.8.2 GPADC Block Diagram

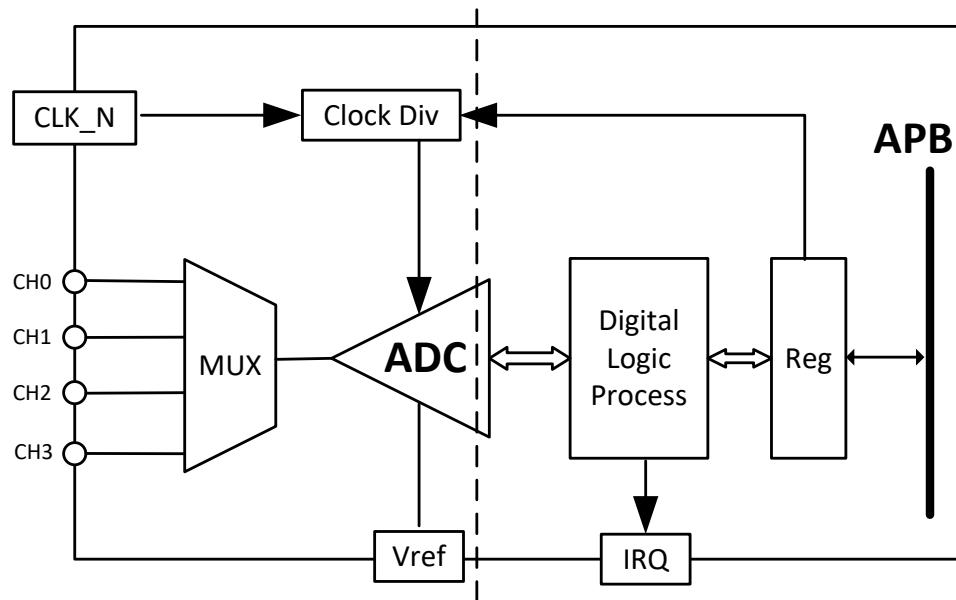


Figure 4-20 GPADC Block Diagram

## 4.8.3 Operations and Functional Descriptions

### 4.8.3.1 External Signals

Signal	Description	Pin Number	Type
GPADC0	Analog Input	IO	Analog
GPADC1	Analog Input	IO	Analog
GPADC2	Analog Input	IO	Analog
GPADC3	Analog Input	IO	Analog
GPADC4	Analog Input	IO	Analog
GPADC5	Analog Input	IO	Analog
GPADC6	Analog Input	IO	Analog
GPADC7	Analog Input	IO	Analog
GPADC8	VBAT Input	VBAT	Analog

Table 4-141 ADC Signals

## 4.8.4 Register List

Module Name	Base Address	
GPADC	0x40043000	
Register Name	Offset Address	Description
GP_SR_CON	0x00	GPADC Sample Rate Configure Register
GP_CTRL	0x04	GPADC Control Register
GP_CS_EN	0x08	GPADC Compare and Select Enable Register
GP_FIFO_INTC	0x0C	GPADC FIFO Interrupt Control Register
GP_FIFO_INTS	0x10	GPADC FIFO Interrupt Status Register
GP_FIFO_DATA	0x14	GPADC FIFO Data Register
GP_CDATA	0x18	GPADC Calibration Data Register
/	/	/
GP_DATAL_INTC	0x20	GPADC Data Low Interrupt Configure Register
GP_DATAH_INTC	0x24	GPADC Data High Interrupt Configure Register
GP_DATA_INTC	0x28	GPADC Data Interrupt Configure Register
GP_DATAL_INTS	0x30	GPADC Data Low Interrupt Status Register
GP_DATAH_INTS	0x34	GPADC Data High Interrupt Status Register
GP_DATA_INTS	0x38	GPADC Data Interrupt Status Register
/	/	/
GP_CHn_CMP_DATA	0x40 + 4 * n	GPADC CH n Compare Data Register
GP_CHn_DATA	0x80 + 4 * n	GPADC CH n Data Register

**Table 4-142 Timer Register List**

## 4.8.5 Register Description

### 4.8.5.1 GPADC Sample Rate Configure Register

<b>Address : 0x0000</b>			<b>Name: GP_SR_CON</b> <b>Default: 0x01DF_002F</b>
Field	Type	Default	Description
31: 16	R/W	0x207(50K)	FS_DIV. ADC Sample Frequency Divider CLK_IN/(n+1) <i>Note: HOSC = 26MHz, n = 0x207; HOSC = 40MHz, n = 0x320</i>
15:0	R/W	0x34(2uS)	TACQ. ADC acquire time CLK_IN/(n+1) <i>Note: HOSC = 26MHz, n = 0x34; HOSC = 40MHz, n = 0x50</i>

**Table 4-143 GPADC Sample Rate Configure Register**

#### 4.8.5.2 GPADC Control Register

<b>Address : 0x0004</b>			<b>Name: GP_CTRL</b> <b>Default:</b> 0x00800000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:24	R/W	0x0	ADC_FIRST_DLY. ADC First Convert Delay setting, ADC conversion of each channel is delayed by N samples
23:22	/	/	/
21:20	R/W	0x0	ADC_OP_BIAS.(Adjust the bandwidth of the ADC amplifier) ADC OP Bias
19:18	R/W	0x0	GPADC Work Mode 00: Single conversion mode 01: Single-cycle conversion mode 10: Continuous conversion mode 11: Burst conversion mode
17	R/W	0x0	ADC_CALI_EN. ADC Calibration 1: start Calibration, it is clear to 0 after calibration
16	R/W	0x0	ADC_EN. ADC Function Enable 0: Disable 1: Enable <i>Note: the work mode and the channel number must be set before the ADC_EN bit being set.</i>
15:5	/	/	/
4	R/W	0x0	VBAT_DET_EN 0: disable 1: enable <i>Note: the 8<sup>th</sup> channel is designed for VBAT Voltage Detection and this bit should be set to 1 before enable the CH8</i>
3:1	/	/	/
0	R/W	0x0	ADC_LDO_EN 0: disable 1: enable <i>Note: before enabling the ADC function, you must enable the LDO-ADC firstly.</i>

Table 4-144 GPADC Control Register

#### 4.8.5.3 GPADC Compare and Select Enable Register

<b>Address : 0x0008</b>	<b>Name: GP_CS_EN</b>
-------------------------	-----------------------

			<b>Default:</b> 0x00000000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:24	/	/	/
[n+16] (n = 0-8)	R/W	0x0	ADC_CHn_CMP_EN Channel n Compare Enable 0: Disable 1: Enable
15:8	/	/	/
[n] (n = 0-8)	R/W	0x0	ADC_CHn_SELECT. Analog input channel n Select 0: Disable 1: Enable

**Table 4-145 GPADC Compare and Select Enable Register**

#### 4.8.5.4 GPADC FIFO Interrupt Control Register

<b>Address :</b> 0x000c			<b>Name:</b> GP_FIFO_INTC
			<b>Default:</b> 0x0000_1F00
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:19	/	/	/
18	R/W	0x0	FIFO_DATA_DRQ_EN ADC FIFO Data DRQ Enable 0:Disable 1:Enable
17	R/W	0x0	FIFO_OVERRUN_IRQ_EN. ADC FIFO Over Run IRQ Enable 0: Disable 1: Enable
16	R/W	0x0	FIFO_DATA_IRQ_EN. ADC FIFO Data Available IRQ Enable 0: Disable 1: Enable
15:14	/	/	/
13:8	R/W	0x1F	FIFO_TRIGGER_LEVEL. Interrupt and DMA request trigger level for ADC Trigger Level = TXTL + 1
7:5	/	/	/
4	R/W1 C	0x0	FIFO_FLUSH. ADC FIFO Flush Write '1' to flush TX FIFO, self clear to '0'
3:0	/	/	/

**Table 4-146 GPADC FIFO Interrupt Control Register**

#### 4.8.5.5GPADC FIFO Interrupt Status Register

<b>Address :</b> 0x0010			<b>Name:</b> GP_FIFO_INTS <b>Default:</b> 0x00000000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:18	/	/	/
17	R/W1 C	0x0	FIFO_OVERRUN_PENDING. ADC FIFO Over Run IRQ pending 0: No Pending IRQ 1: FIFO Overrun Pending IRQ Write '1' to clear this interrupt or automatic clear if interrupt condition fails
16	R/W1 C	0x0	FIFO_DATA_PENDING. ADC FIFO Data Available pending Bit 0: NO Pending IRQ 1: FIFO Available Pending IRQ Write '1' to clear this interrupt or automatic clear if interrupt condition fails
15:14	/	/	/
13:8	R	0x0	<b>RXA_CNT.</b> ADC FIFO available Sample Word Counter
7:0	/	/	/

Table 4-147 GPADC FIFO Interrupt Status Register

#### 4.8.5.6GPADC FIFO Data Register

<b>Address :</b> 0x0014			<b>Name:</b> GP_FIFO_DATA <b>Default:</b> 0x00000000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:12	/	/	/
11:0	R/W	0xX	GP_FIFO_DATA GPADC Data in FIFO

Table 4-148 GPADC FIFO Data Register

#### 4.8.5.7GPADC Calibration Data Register

<b>Address :</b> 0x0018			<b>Name:</b> GP_CDATA <b>Default:</b> 0x00000000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
11:0	R/W	0x000	GP_CDATA GPADC Calibration Data

**Table 4-149 GPADC Calibration Data Register**

#### 4.8.5.8 GPADC Low Interrupt Configure Register

<b>Address</b> : 0x0020			<b>Name:</b> GP_DATAL_INTC <b>Default:</b> 0x00000000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
[n] (n = 0-8)	RW	0	CHn_LOW_IRQ_EN 0: Disable 1: Enable

**Table 4-150 GPADC Low Interrupt Configure Register**

#### 4.8.5.9 GPADC HIGH Interrupt Configure Register

<b>Address</b> : 0x0024			<b>Name:</b> GP_DATAH_INTC <b>Default:</b> 0x00000000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
[n] (n = 0-8)	RW	0	CHn_HIGH_IRQ_EN 0: Disable 1: Enable

**Table 4-151 GPADC HIGH Interrupt Configure Register**

#### 4.8.5.10 GPADC DATA Interrupt Configure Register

<b>Address</b> : 0x0028			<b>Name:</b> GP_DATA_INTC <b>Default:</b> 0x00000000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
[n] (n = 0-8)	RW	0	CHn_DATA_IRQ_EN 0: Disable 1: Enable

**Table 4-152 GPADC DATA Interrupt Configure Register**

#### 4.8.5.11 GPADC Low Interrupt Status Register

<b>Address</b> : 0x0030			<b>Name:</b> GP_DATAL_INTS <b>Default:</b> 0x00000000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:8	/	/	/
[n] (n = 0-8)	RW	0	CHn_LOW_PENGDING

			1: Channel n Voltage Low Available Pending IRQ Write '1' to clear this interrupt or automatic clear if interrupt condition fails
--	--	--	---

**Table 4-153 GPADC Low Interrupt Status Register**

#### 4.8.5.12 GPADC High Interrupt Status Register

Address : 0x0034			Name: GP_DATAH_INTS Default: 0x00000000
Field	Type	Default	Description
31:8	/	/	/
[n] (n = 0-8)	RW	0	CHn_HIGH_PENGDING 1: Channel n Voltage High Available Pending IRQ Write '1' to clear this interrupt or automatic clear if interrupt condition fails

**Table 4-154 GPADC High Interrupt Status Register**

#### 4.8.5.13 GPADC Data Interrupt Status Register

Address : 0x0038			Name: Default: 0x00000000
Field	Type	Default	Description
31:8	/	/	/
[n] (n = 0-8)	RW	0	CHn_DATA_PENGDING 0: NO Pending IRQ 1: Channel n Data Available Pending IRQ Write '1' to clear this interrupt or automatic clear if interrupt condition fails

**Table 4-155 GPADC Data Interrupt Status Register**

#### 4.8.5.14 GPADC CHn Compare Data Register

Address : 0x0040 + 4 * n (n = 0-8)			Name: GP_CHn_CMP_DATA Default: 0xBFF_0400
Field	Type	Default	Description
31:28	/	/	/
27:16	R/W	0xBFF	CHn_CMP_HIG_DATA Channel n Voltage High Value
15:12	/	/	/
11:0	R/W	0x400	CHn_CMP_LOW_DATA Channel n Voltage Low Value

**Table 4-156 GPADC CHn Compare Data Register****4.8.5.15 GPADC CHn Data Register**

<b>Address :</b> 0x0080 + 4 * n (n = 0-8)			<b>Name:</b> GP_CHn_DATA <b>Default:</b> 0x00000000
Field	Type	Default	Description
31:12	/	/	/
11:0	R	0x000	GP_CHn_DATA Channel n Data

**Table 4-157 GPADC CHn Data Register**

# Chapter 5 Audio Interface

## 5.1 Digital Audio Controller (DAUDIO)

### 5.1.1 Overview

The Digital Audio Interface Controller has been designed to transfer streaming audio-data between the system memory and the codec chip. The controller supports standard I2S format, Left-justified Mode format, Right-justified Mode format, PCM Mode format and TDM Mode format.

Features:

- Compliant with standard Philips Inter-IC sound (I2S) bus specification
- Compliant with Left-justified, Right-justified, PCM mode, and TDM (Time Division Multiplexing) format
- Support different sample period width in each interface when using LRCK and LRCKR at the same time
- Support full-duplex synchronous work mode
- Support Master / Slave mode
- Support adjustable interface voltage
- Support clock up to 100MHz
- Support adjustable audio sample resolution from 8-bit to 32-bit.
- Support up to 8 slots which has adjustable width from 8-bit to 32-bit.
- Support sample rate from 8KHz to 192KHz
- Support up to 4 data output pin
- Support 8-bits u-law and 8-bits A-law companded sample
- One 128 depth x 32-bit width FIFO for data transmit, one 64 depth x 32-bit width FIFO for data receive
- Support programmable PCM frame width: 1 BCLK width (short frame) and 2 BCLKs width (long frame)

### 5.1.2 Interface Block Diagram

#### 5.1.2.1 Typical Applications

The Digital Audio Interface is the extended of I2S and pcm which provides a serial bus interface for stereo and multichannel audio data. This interface is most commonly used by consumer audio market, including compact disc,

digital audio tape, digital sound processors, and digital TV-sound.

### 5.1.2.2 Functional Block Diagram

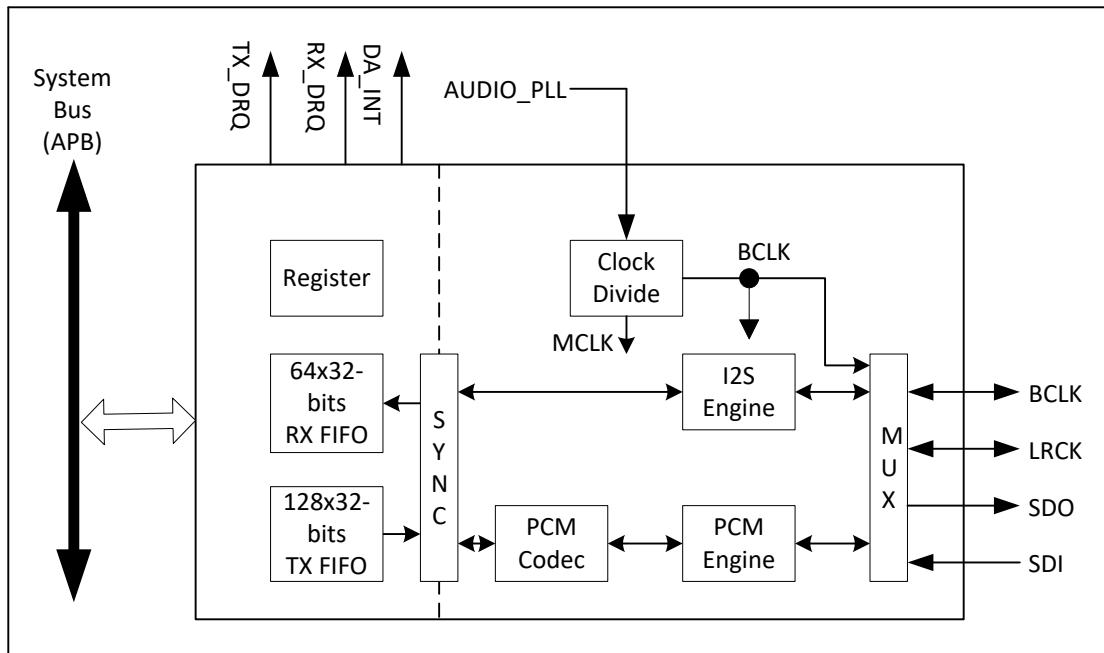


Figure 5-1 DAUDIO Function Block Diagram

### 5.1.3 Operations and Functional Descriptions

#### 5.1.3.1 External Signals

The following table describes the external signals of DAI. BCLK and LRCK are bidirectional I/O, when DAI is configured as Master device, BCLK and LRCK is output pin; when DAI is configured as slave device, BCLK and LRCK is input pin. MCLK is a output pin for external device. SDO is always the serial data output pin, and SDI is the serial data input. For information about General Purpose I/O port, see Port Controller.

Signal	Description	Type
MCLK	Digital Audio Interface MCLK Output	O
LRCK	Digital Audio Interface Serial Clock	I/O
BCLK	Digital Audio Interface Sample Rate Clock/Sync	I/O
SDO	Digital Audio Interface Serial Data Output	O
SDI	Digital Audio Interface Serial Data Input	I

Table 5-1 Digital Audio Interface External Signals

### 5.1.3.2 Clock Sources

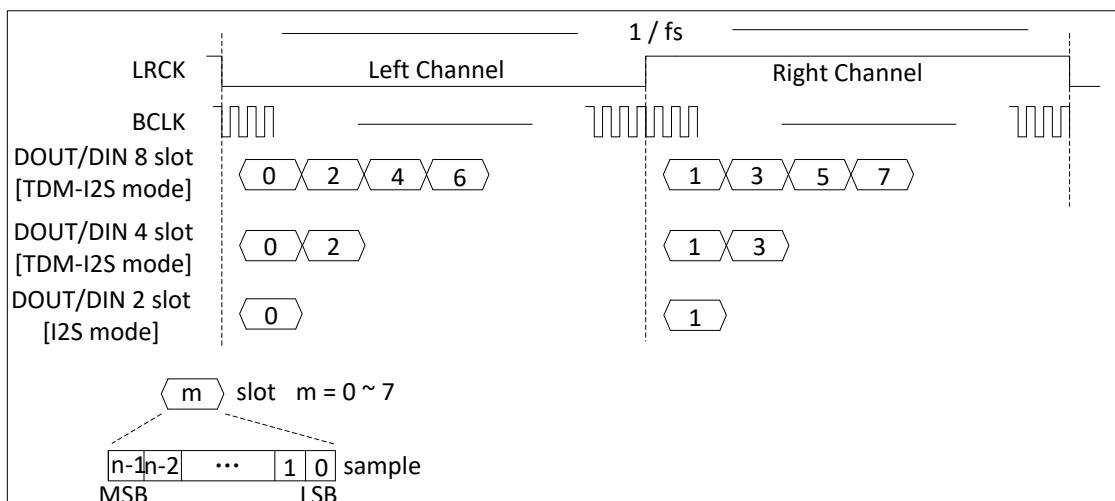
Each DAI controller get three different clocks. Users can select one of them to make DAI Clock Source. The table describes the clock sources for DAI. Users can see Clock Controller Unit for clock setting, configuration and gating information.

**Table 5-2 Digital Audio Interface Clock Sources**

Clock Sources	Description
Audio_PLL	24.576Mhz or 22.5792Mhz generated by AUDIO-PLL to produce 48KHz or 44.1KHz serial frequency.

### 5.1.4 Digital Audio Interface Transmit Format

The DAI support standard I2S mode, Left-justified I2S mode, Right-justified I2S mode, PCM mode and TDM mode. Software can select one of them in which the DAI works by setting the DAI Control Register.



**Figure 5-2 Timing diagram for standard I2S/TDM-I2S mode**

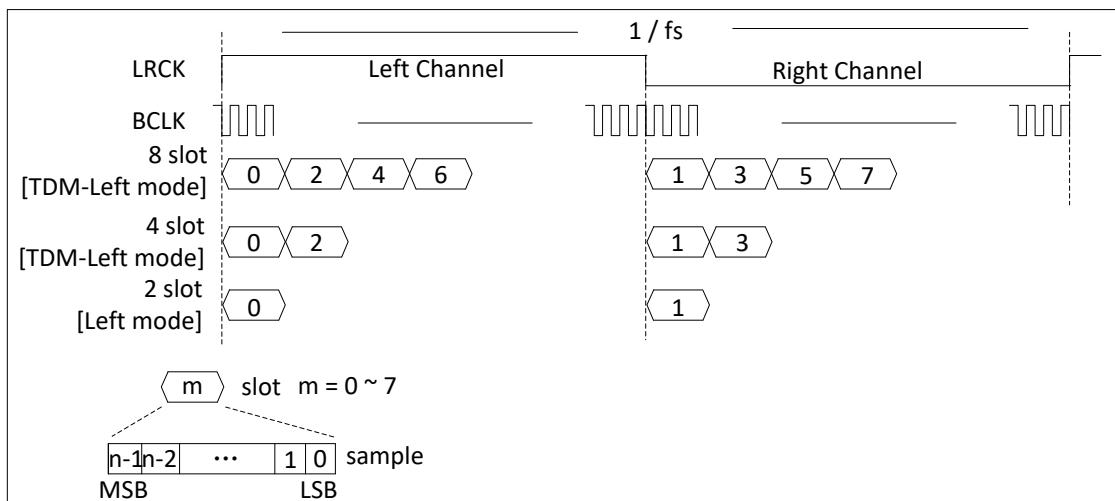


Figure 5-3 Timing diagram for Left-justified/TDM-Left mode

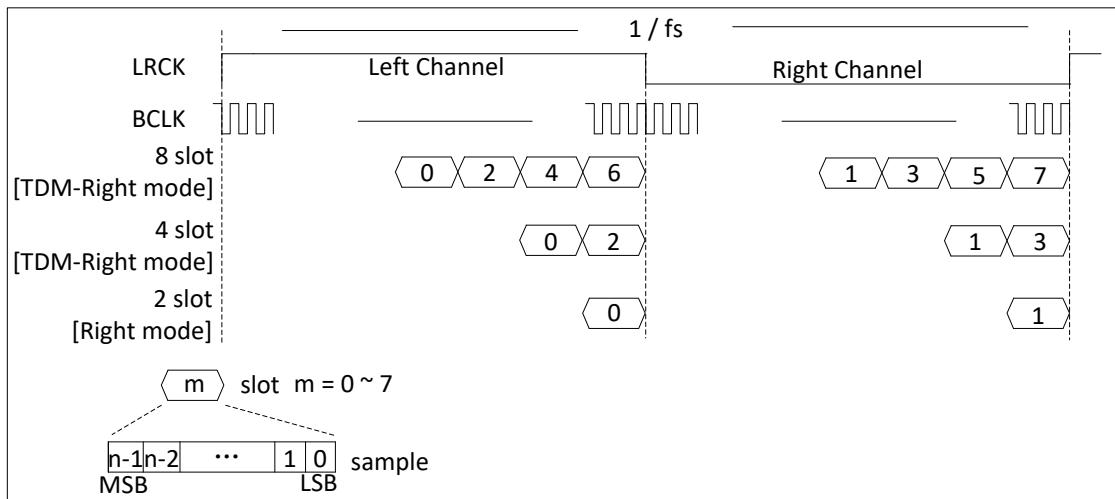


Figure 5-4 Timing diagram for Right-justified/TDM-Right mode

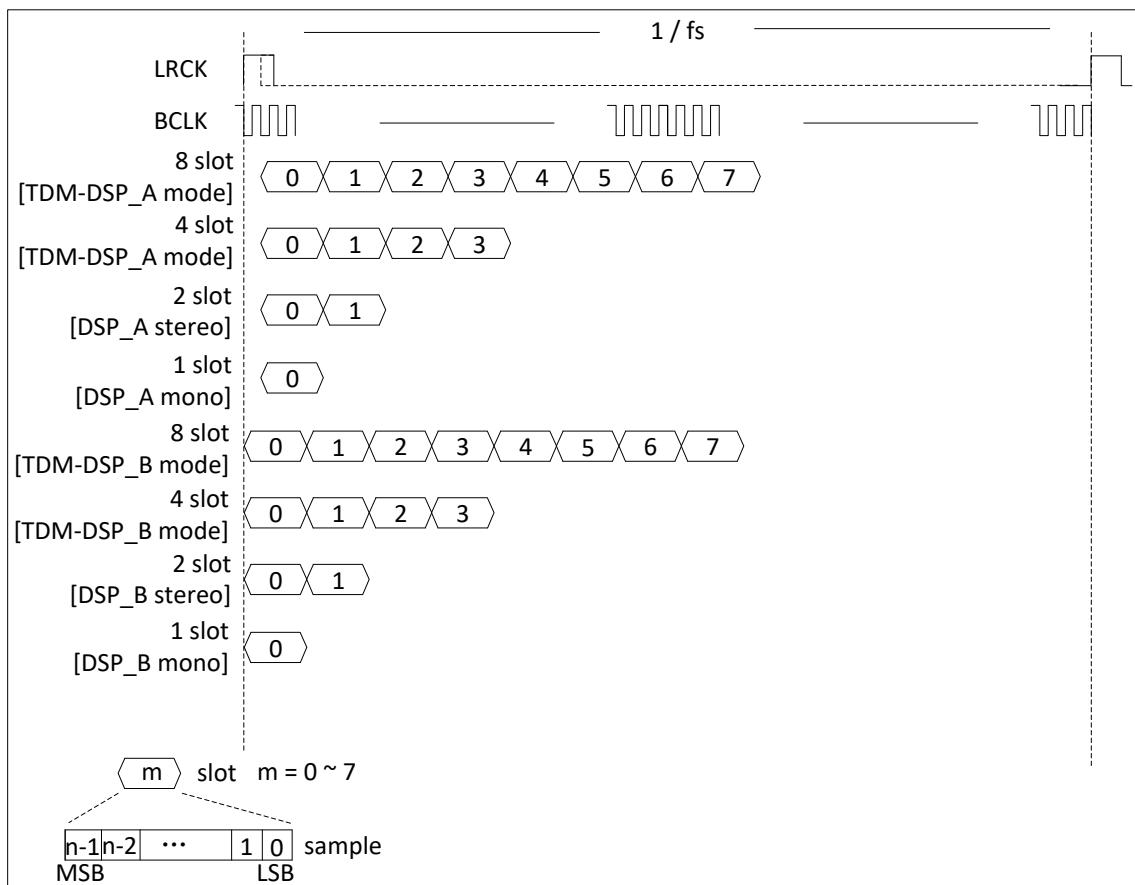


Figure 5-5 Timing diagram for PCM/TDM-PCM mode

## 5.1.5 Operation Modes

The software operation of the DAI is divided into five steps: system setup, DAI initialization, the channel setup, DMA setup and Enable/Disable module. These five setup are described in detail in the following sections.

### 5.1.5.1 System setup and DAI initialization

The first step in the system setup is properly programming the GPIO. Because the DAI port is a multiplex pin. You can find the function in the pin multiplex specification. The clock source for the DAI should be followed. At first you must reset the audio PLL through the PLL\_ENABLE bit of PLL\_AUDIO\_CTRL\_REG in the CCMU. The second step, you must setup the frequency of the audio pll in the PLL\_AUDIO\_CTRL\_REG. The configuration of audio pll can be found in the chapter 1.3.2. After that, you must open the DAI gating through the DAIO\_CLK\_REG/DAI1\_CLK\_REG when you checkout that the LOCK bit of PLL\_AUDIO\_CTRL\_REG become 1. At last, you must reset the DAI the BUS\_RST\_REG3's bit[13:12] and open the DAI bus gating in the BUS\_GATING\_REG2's bit[13:12].

After the system setup, the register of DAI can be setup. At first, you should initialize the DAI. You should close the global enable bit(DAI\_CTL[0]), TX enable bit(DAI\_CTL[2]) and RX enable bit(DAI\_CTL[1]) by write 0 to it. After

that, you must clear the TX/RX FIFO by write 0 to register DAI\_FCTL[25:24]. At last, you can clear the TX/RX FIFO counter by write 0 to DAI\_TXCNT/DAI\_RXCNT.

### 5.1.5.2 The channel setup and DMA setup

Before the usage and control of DAI, you must configure the TWI. The configuration of TWI will not described in this chapter. But you can only configure DAI master and slave though the TWI. In the following, you can setup the DAI of mater and slave. The configuration can be referred to the the protocol of DAI. Then, you can set the translation mode, the sample precision, the wide of slot, the frame mode and the trigger level. The register set can be found in the spec.

The DAI support three method to transfer the data. The most common way is DMA, the set of DMA can be found in the DMA spec. In this module, you just to enable the DRQ.

### 5.1.5.3 Enable and disable the DAI

To enable the function, you can enable TX/RX by write the DAI\_CTL[2:1]. After that, you must enable DAI by write the Globe Enable bit to 1 in the DAI\_CTL. The disable process is write the Globe Enable to 0.

## 5.1.6 Register List

Module Name	Base Address	
Digital Audio	0x40042C00	
Register Name	Offset Address	Description
DA_CTL	0x00	Digital Audio Control Register
DA_FMT0	0x04	Digital Audio Format Register 0
DA_FMT1	0x08	Digital Audio Format Register 1
DAISTA	0x0C	Digital Audio Interrupt Status Register
DA_RXFIFO	0x10	Digital Audio RX FIFO Register
DA_FCTL	0x14	Digital Audio FIFO Control Register
DA_FSTA	0x18	Digital Audio FIFO Status Register
DA_INT	0x1C	Digital Audio DMA & Interrupt Control Register
DA_TXFIFO	0x20	Digital Audio TX FIFO Register
DA_CLKD	0x24	Digital Audio Clock Divide Register
DA_TXCNT	0x28	Digital Audio TX Sample Counter Register
DA_RXCNT	0x2C	Digital Audio RX Sample Counter Register
DA_CHCFG	0x30	Digital Audio Channel Configuration register
DA_TX0CHCFG	0x34	Digital Audio TX0 Channel Configuration register
DA_TX1CHSEL	0x38	Digital Audio TX1 Channel Select Register
DA_TX2CHSEL	0x3C	Digital Audio TX2 Channel Select Register
DA_TX3CHSEL	0x40	Digital Audio TX3 Channel Select Register
DA_TX0CHMAP	0x44	Digital Audio TX0 Channel Mapping Register

DA_TX1CHMAP	0x48	Digital Audio TX1 Channel Mapping Register
DA_TX2CHMAP	0x4C	Digital Audio TX2 Channel Mapping Register
DA_TX3CHMAP	0x50	Digital Audio TX3 Channel Mapping Register
DA_RXCHSEL	0x54	Digital Audio RX Channel Select register
DA_RXCHMAP	0x58	Digital Audio RX Channel Mapping Register

Table 5-3 Digital Audio Register List

## 5.1.7 Register Description

### 5.1.7.1 Digital Audio Control Register

Address : 0x0000			Name: DA_CTL Default: 0x0006_0000
Field	Type	Default	Description
31:19	/	/	/
18	R/W	1	BCLK_OUT 0: input 1: output
17	R/W	1	LRCK_OUT 0: input 1: output
16:9	/	/	/
8	R/W	0	SDO0_EN 0: Disable, Hi-Z state 1: Enable
7	/	/	/
6	R/W	0	OUT Mute 0: normal transfer 1: force DOUT to output 0
5:4	R/W	0	MODE_SEL Mode Selection 0: PCM mode (offset 0: DSP_B; offset 1: DSP_A) 1: Left mode (offset 0: LJ mode; offset 1: I2S mode) 2: Right-Justified mode 3: Reserved
3	R/W	0	LOOP Loop back test 0: Normal mode 1: Loop back test When set ‘1’ , connecting the SDO0 with the SDI
2	R/W	0	TXEN

			Transmitter Block Enable 0: Disable 1: Enable
1	R/W	0	RXEN Receiver Block Enable 0: Disable 1: Enable
0	R/W	0	GEN Globe Enable A disable on this bit overrides any other block or channel enables. 0: Disable 1: Enable

**Table 5-4 Digital Audio Control Register**

### 5.1.7.2Digital Audio Format Register 0

Address : 0x0004			Name: DA_FMT0 Default: 0x0000_0033
Field	Type	Default	Description
31	/	/	/
30	R/W	0	LRCK_WIDTH (only apply in PCM mode ) LRCK width 0: LRCK = 1 BCLK width (short frame) 1: LRCK = 2 BCLK width (long frame)
29:20	/	/	/
19	R/W	0	LRCK_POLARITY/LRCKR_POLARITY When apply in I2S / Left-Justified / Right-Justified mode: 0: Left channel when LRCK is low 1: Left channel when LRCK is high When apply in PCM mode: 0: PCM LRCK/LRCKR asserted at the negative edge 1: PCM LRCK/LRCKR asserted at the positive edge
18	/	/	/
17:8	R/W	0	LRCK_PERIOD It is used to program the number of BCLKs per channel of sample frame. This value is interpreted as follow: PCM mode: Number of BCLKs within (Left + Right) channel width I2S / Left-Justified / Right-Justified mode: Number of BCLKs within each individual channel width (Left or Right) N+1 For example: n = 7: 8 BCLK width

			... n = 1023: 1024 BCLKs width
7	R/W	0	BCLK_POLARITY 0: normal mode, negative edge drive and positive edge sample 1: invert mode, positive edge drive and negative edge sample
6:4	R/W	3	SR Sample Resolution 0: Reserved 1: 8-bit 2: 12-bit 3: 16-bit 4: 20-bit 5: 24-bit 6: 28-bit 7: 32-bit
3	R/W	0	EDGE_TRANSFER 0: SDO drive data and SDI sample data at the different BCLK edge 1: SDO drive data and SDI sample data at the same BCLK edge BCLK_POLARITY = 0, use negative edge BCLK_POLARITY = 1, use positive edge
2:0	R/W	0x3	SW Slot Width Select 0: Reserved 1: 8-bit 2: 12-bit 3: 16-bit 4: 20-bit 5: 24-bit 6: 28-bit 7: 32-bit

**Table 5-5 Digital Audio Format Register 0**

### 5.1.7.3Digital Audio Format Register 1

Address : 0x0008			Name: DA_FMT1 Default: 0x0000_0030
Field	Type	Default	Description
31:8	/	/	
7	R/W	0	RX MLS MSB / LSB First Select 0: MSB First 1: LSB First

6	R/W	0	TX MLS MSB / LSB First Select 0: MSB First 1: LSB First
5:4	R/W	3	SEXT Sign Extend in slot [sample resolution < slot width] 0: Zeros or audio gain padding at LSB position 1: Sign extension at MSB position 2: Reserved 3: Transfer 0 after each sample in each slot
3:2	R/W	0	RX_PDM PCM Data Mode 0: Linear PCM 1: reserved 2: 8-bits u-law 3: 8-bits A-law
1:0	R/W	0	TX_PDM PCM Data Mode 0: Linear PCM 1: reserved 2: 8-bits u-law 3: 8-bits A-law

**Table 5-6 Digital Audio Format Register 1**

### 5.1.7.4 Digital Audio Interrupt Status Register

Address : 0x000C			Name: DAISTA Default: 0x0000_0010
Field	Type	Default	Description
31:7	/	/	/
6	R/W	0	TXU_INT TX FIFO Under run Pending Interrupt 0: No Pending Interrupt 1: FIFO Under run Pending Interrupt Write 1 to clear this interrupt
5	R/W	0	TXO_INT TX FIFO Overrun Pending Interrupt 0: No Pending Interrupt 1: FIFO Overrun Pending Interrupt Write ‘1’ to clear this interrupt
4	R/W	1	TXE_INT TX FIFO Empty Pending Interrupt

			0: No Pending IRQ 1: FIFO Empty Pending Interrupt when data in TX FIFO are less than TX trigger level Write ‘1’ to clear this interrupt or automatic clear if interrupt condition fails.
3	/	/	/
2	R/W	0	RXU_INT RX FIFO Under run Pending Interrupt 0: No Pending Interrupt 1:FIFO Under run Pending Interrupt Write 1 to clear this interrupt
1	R/W	0	RXO_INT RX FIFO Overrun Pending Interrupt 0: No Pending IRQ 1: FIFO Overrun Pending IRQ Write ‘1’ to clear this interrupt
0	R/W	0	RXA_INT RX FIFO Data Available Pending Interrupt 0: No Pending IRQ 1: Data Available Pending IRQ when data in RX FIFO are more than RX trigger level Write ‘1’ to clear this interrupt or automatic clear if interrupt condition fails.

**Table 5-7 Digital Audio Interrupt Status Register**

### 5.1.7.5Digital Audio RX FIFO register

Address : 0x0010			Name: DA_RXFIFO Default: 0x0000_0000
Field	Type	Default	Description
31:0	R	0	RX_DATA RX Sample Host can get one sample by reading this register. The left channel sample data is first and then the right channel sample.

**Table 5-8 Digital Audio RX FIFO register**

### 5.1.7.6Digital Audio FIFO Control Register

Address : 0x0014			Name: DA_FCTL Default: 0x0004_00F0
Field	Type	Default	Description
31	R/W	0	HUB_EN

			Audio Hub Enable 0 : Disable 1 : Enable
30:26	/	/	/
25	R/W	0	FTX Write ‘1’ to flush TX FIFO, self clear to ‘0’ .
24	R/W	0	FRX Write ‘1’ to flush RX FIFO, self clear to ‘0’ .
23:19	/	/	/
18:12	R/W	0x40	TXTL TX FIFO Empty Trigger Level Interrupt and DMA request trigger level for TXFIFO normal condition Trigger Level = TXTL
11:10	/	/	/
9:4	R/W	0xF	RXTL RX FIFO Trigger Level Interrupt and DMA request trigger level for RXFIFO normal condition Trigger Level = RXTL + 1
3	/	/	/
2	R/W	0	TXIM TX FIFO Input Mode (Mode 0, 1) 0: Valid data at the MSB of TXFIFO register 1: Valid data at the LSB of TXFIFO register Example for 20-bits transmitted audio sample: Mode 0: FIFO_I[31:0] = {APB_WDATA[31:12], 12' h0} Mode 1: FIFO_I[31:0] = {APB_WDATA[19:0], 12' h0}
1:0	R/W	0	RXOM RX FIFO Output Mode (Mode 0, 1, 2, 3) 00: Expanding ‘0’ at LSB of DA_RXFIFO register. 01: Expanding received sample sign bit at MSB of DA_RXFIFO register. 10: Truncating received samples at high half-word of DA_RXFIFO register and low half-word of DA_RXFIFO register is filled by ‘0’ . 11: Truncating received samples at low half-word of DA_RXFIFO register and high half-word of DA_RXFIFO register is expanded by its sign bit. Example for 20-bits received audio sample: Mode 0: APB_RDATA[31:0] = {FIFO_O[31:12], 12' h0} Mode 1: APB_RDATA [31:0] = {12{FIFO_O[31]}, FIFO_O[31:12]} Mode 2: APB_RDATA [31:0] = {FIFO_O[31:16], 16' h0} Mode 3: APB_RDATA [31:0] = {16{FIFO_O[31]}, FIFO_O[31:16]}

**Table 5-9 Digital Audio FIFO Control Register**

### 5.1.7.7 Digital Audio FIFO Status Register

Address : 0x0018			Name: DA_FSTA Default: 0x1080_0000
Field	Type	Default	Description
31:29	/	/	/
28	R	1	TXE TX FIFO Empty 0: No room for new sample in TX FIFO 1: More than one room for new sample in TX FIFO (>= 1 word)
27:24	/	/	/
23:16	R	0x80	TXE_CNT TX FIFO Empty Space Word Counter
15:9	/	/	/
8	R	0	RXA RX FIFO Available 0: No available data in RX FIFO 1: More than one sample in RX FIFO (>= 1 word)
7	/	/	/
6:0	R	0	RXA_CNT RX FIFO Available Sample Word Counter

Table 5-10 Digital Audio FIFO Status Register

### 5.1.7.8 Digital Audio DMA & Interrupt Control Register

Address : 0x001C			Name: DA_INT Default: 0x0000_0000
Field	Type	Default	Description
31:8	/	/	/
7	R/W	0	TX_DRQ TX FIFO Empty DRQ Enable 0: Disable 1: Enable
6	R/W	0	TXUI_EN TX FIFO Under run Interrupt Enable 0: Disable 1: Enable
5	R/W	0	TXOI_EN TX FIFO Overrun Interrupt Enable 0: Disable 1: Enable When set to ‘1’ , an interrupt happens when writing new audio data if

			TX FIFO is full.
4	R/W	0	TXEI_EN TX FIFO Empty Interrupt Enable 0: Disable 1: Enable
3	R/W	0	RX_DRQ RX FIFO Data Available DRQ Enable 0: Disable 1: Enable When set to ‘1’ , RXFIFO DMA Request line is asserted if Data is available in RX FIFO.
2	R/W	0	RXUI_EN RX FIFO Under run Interrupt Enable 0: Disable 1: Enable
1	R/W	0	RXOI_EN RX FIFO Overrun Interrupt Enable 0: Disable 1: Enable
0	R/W	0	RXAI_EN RX FIFO Data Available Interrupt Enable 0: Disable 1: Enable

**Table 5-11 Digital Audio DMA & Interrupt Control Register**

### 5.1.7.9 Digital Audio TX FIFO register

Address : 0x0020			Name: DA_TXFIFO Default: 0x0000_0000
Field	Type	Default	Description
31:0	W	0	TX_DATA TX Sample Transmitting left, right channel sample data should be written this register one by one. The left channel sample data is first and then the right channel sample.

**Table 5-12 Digital Audio TX FIFO register**

### 5.1.7.10 Digital Audio Clock Divide Register

Address : 0x0024			Name: DA_CLKD Default: 0x0000_0000
Field	Type	Default	Description

31:9	/	/	/
8	R/W	0	<p>MCLKO_EN            0: Disable MCLK Output            1: Enable MCLK Output            Notes: Whether in Slave or Master mode, when this bit is set to 1, MCLK should be output.</p>
7:4	R/W	0	<p>BCLKDIV            BCLK Divide Ratio from PLL2            0: reserved            1: Divide by 1            2: Divide by 2            3: Divide by 4            4: Divide by 6            5: Divide by 8            6: Divide by 12            7: Divide by 16            8: Divide by 24            9: Divide by 32            10: Divide by 48            11: Divide by 64            12: Divide by 96            13: Divide by 128            14: Divide by 176            15: Divide by 192</p>
3:0	R/W	0	<p>MCLKDIV            MCLK Divide Ratio from PLL2 Output            0: reserved            1: Divide by 1            2: Divide by 2            3: Divide by 4            4: Divide by 6            5: Divide by 8            6: Divide by 12            7: Divide by 16            8: Divide by 24            9: Divide by 32            10: Divide by 48            11: Divide by 64            12: Divide by 96            13: Divide by 128            14: Divide by 176            15: Divide by 192</p>

**Table 5-13 Digital Audio Clock Divide Register**

### 5.1.7.11 Digital Audio TX Counter register

Address : 0x0028			Name: DA_TXCNT Default: 0x0000_0000
Field	Type	Default	Description
31:0	R/W	0	<p>TX_CNT</p> <p>TX Sample Counter</p> <p>The audio sample number of sending into TXFIFO. When one sample is put into TXFIFO by DMA or by host IO, the TX sample counter register increases by one. The TX sample counter register can be set to any initial value at any time. After been updated by the initial value, the counter register should count on base of this initial value.</p>

Table 5-14 Digital Audio TX Counter register

### 5.1.7.12 Digital Audio RX Counter register

Address : 0x002C			Name: DA_RXCNT Default: 0x0000_0000
Field	Type	Default	Description
31:0	R/W	0	<p>RX_CNT</p> <p>RX Sample Counter</p> <p>The audio sample number of writing into RXFIFO. When one sample is written by Digital Audio Engine, the RX sample counter register increases by one. The RX sample counter register can be set to any initial value at any time. After been updated by the initial value, the counter register should count on base of this initial value.</p>

Table 5-15 Digital Audio RX Counter register

### 5.1.7.13 Digital Audio Channel Configuration register

Address : 0x0030			Name: DA_CHCFG Default: 0x0000_0000
Field	Type	Default	Description
31:10	/	/	/
9	R/W	0	<p>TX_SLOT_HIZ</p> <p>0: normal mode for the last half cycle of BCLK in the slot</p> <p>1: turn to hi-z state for the last half cycle of BCLK in the slot</p>
8	R/W	0	<p>TXn_STATE</p> <p>0: transfer level 0 when not transferring slot</p> <p>1: turn to hi-z state (TDM) when not transferring slot</p>
7	/	/	/

6:4	R/W	0	RX_SLOT_NUM RX Channel/Slot Number which between CPU/DMA and FIFO 0: 1 channel or slot ... 7: 8 channels or slots
3	/	/	/
2:0	R/W	0	TX_SLOT_NUM TX Channel/Slot Number which between CPU/DMA and FIFO 0: 1 channel or slot ... 7: 8 channels or slots

**Table 5-16 Digital Audio Channel Configuration register**

#### 5.1.7.14 Digital Audio TXn Channel Select register

Address : 0x34 + N*4 (N = 0~3)			Name: DA_TXnCHSEL Default: 0x0000_0000
Field	Type	Default	Description
31:13	/	/	/
12	R/W	0	TXn_OFFSET TXn offset tune, TXn data offset to LRCK 0: no offset 1: data is offset by 1 BCLKs to LRCK
11:4	R/W	0	TXn_CHEN TXn Channel (slot) enable, bit[11:4] refer to slot [7:0]. When one or more slot(s) is(are) disabled, the affected slot(s) is(are) set to disable state 0: disable 1: enable
3	/	/	/
2:0	R/W	0	TXn_CHSEL TXn Channel (slot) number Select for each output 0: 1 channel / slot ... 7: 8 channels / slots

**Table 5-17 Digital Audio TXn Channel Select register**

#### 5.1.7.15 Digital Audio TXn Channel Mapping Register

Address : 0x44 + n*4 (n = 0, 1, 2, 3)			Name: DA_TXnCHMAP Default: 0x0000_0000
Field	Type	Default	Description
31	/	/	/

30:28	R/W	0	TXn_CH7_MAP TXn Channel7 Mapping 0: 1st sample ... 7: 8th sample
27	/	/	/
26:24	R/W	0	TXn_CH6_MAP TXn Channel6 Mapping 0: 1st sample ... 7: 8th sample
23	/	/	/
22:20	R/W	0	TXn_CH5_MAP TXn Channel5 Mapping 0: 1st sample ... 7: 8th sample
19	/	/	/
18:16	R/W	0	TXn_CH4_MAP TXn Channel4 Mapping 0: 1st sample ... 7: 8th sample
15	/	/	/
14:12	R/W	0	TXn_CH3_MAP TXn Channel3 Mapping 0: 1st sample ... 7: 8th sample
11	/	/	/
10:8	R/W	0	TXn_CH2_MAP TXn Channel2 Mapping 0: 1st sample ... 7: 8th sample
7	/	/	/
6:4	R/W	0	TXn_CH1_MAP TXn Channel1 Mapping 0: 1st sample ... 7: 8th sample
3	/	/	/
2:0	R/W	0	TXn_CH0_MAP

			TXn Channel0 Mapping 0: 1st sample ... 7: 8th sample
--	--	--	---

**Table 5-18 Digital Audio TXn Channel Mapping Register**

### 5.1.7.16 Digital Audio RX Channel Select register

Address : 0x0054			Name: DA_RXCHSEL Default: 0x0000_0000
Field	Type	Default	Description
31:13	/	/	/
12	R/W	0	RX_OFFSET RX offset tune, RX data offset to LRCK 0: no offset 1: data is offset by 1 BCLKs to LRCK
11:3	/	/	
			RX_CHSEL RX Channel (slot) number Select for input 0: 1 channel / slot ... 7: 8 channels / slots
2:0	R/W	0	

**Table 5-19 Digital Audio RX Channel Select register**

### 5.1.7.17 Digital Audio RX Channel Mapping Register

Address : 0x0058			Name: DA_RXCHMAP Default: 0x0000_0000
Field	Type	Default	Description
31	/	/	/
30:28	R/W	0	RX_CH7_MAP RX Channel7 Mapping 0: 1st sample ... 7: 8th sample
27	/	/	/
26:24	R/W	0	RX_CH6_MAP RX Channel6 Mapping 0: 1st sample ... 7: 8th sample
23	/	/	/

22:20	R/W	0	RX_CH5_MAP RX Channel5 Mapping 0: 1st sample ... 7: 8th sample
19	/	/	/
18:16	R/W	0	RX_CH4_MAP RX Channel4 Mapping 0: 1st sample ... 7: 8th sample
15	/	/	/
14:12	R/W	0	RX_CH3_MAP RX Channel3 Mapping 0: 1st sample ... 7: 8th sample
11	/	/	/
10:8	R/W	0	RX_CH2_MAP RX Channel2 Mapping 0: 1st sample ... 7: 8th sample
7	/	/	/
6:4	R/W	0	RX_CH1_MAP TX Channel1 Mapping 0: 1st sample ... 7: 8th sample
3	/	/	/
2:0	R/W	0	RX_CH0_MAP RX Channel0 Mapping 0: 1st sample ... 7: 8th sample

**Table 5-20 Digital Audio RX Channel Mapping Register**

### 5.1.7.18 Digital Audio DBG register

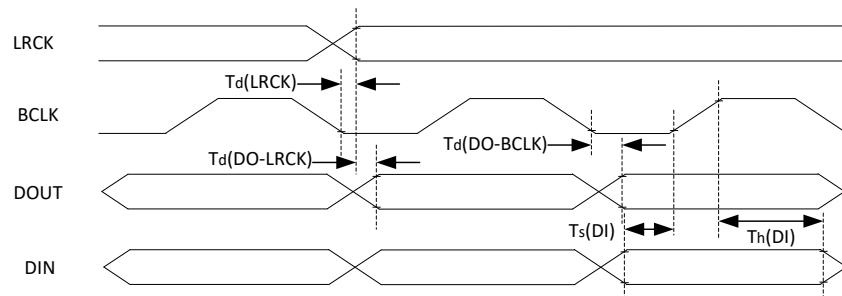
Address : 0x005C		Name: DA_DBG Default: 0x0000_0000	
Field	Type	Default	Description

31:0	/	/	/
------	---	---	---

**Table 5-21 Digital Audio DBG register**

### 5.1.7.19 Digital Audio Interface Electrical Characteristics

The timing of DAI in Master mode is showed in the following Figure.


**Figure 5-6 DAI Timing in Master Mode**

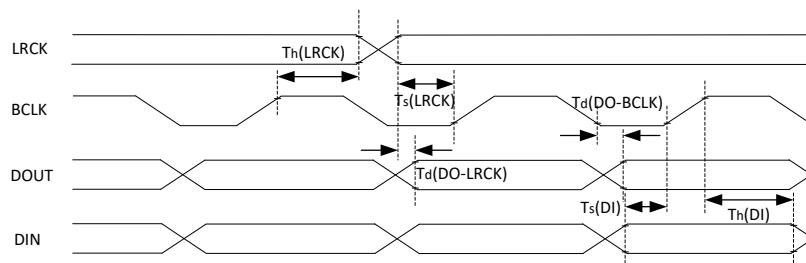
The timing parameters of DAI in Master mode are showed in the following Table.

PARAMETER		MIN	MAX	UNITS
$T_d(LRCK)$	LRCK delay		10	ns
$T_d(DO-LRCK)$	LRCK to DOUT delay(For L/JF)		10	ns
$T_d(DO-BCLK)$	BCLK to DOUT delay		10	ns
$T_s(DI)$	DIN setup	4		ns
$T_h(DI)$	DIN hold	4		ns
$T_r$	BCLK Rise time		8*	ns
$T_f$	BCLK Fall time		8	ns

Note \*容性负载的大小有关

**Table 5-22 DAI Timing Constants at 3.3V in Master Mode**

The timing of DAI in Slave mode is showed in the following Figure


**Figure 5-7 DAI Timing in Slave Mode**

The timing parameters of DAI in Slave mode are showed in the following Table

PARAMETER	MIN	MAX	UNITS
-----------	-----	-----	-------

$T_s(LRCK)$	LRCK setup	4		ns
$T_h(LRCK)$	LRCK hold	4		ns
$T_d(DO-LRCK)$	LRCK to DOUT delay(For LJF)		10	ns
$T_d(DO-BCLK)$	BCLK to DOUT delay		10	ns
$T_s(DI)$	DIN setup	4		ns
$T_h(DI)$	DIN hold	4		ns
$T_r$	BCLK Rise time		4	ns
$T_f$	BCLK Fall time		4	ns

Table 5-23 DAI Timing Constants in Slave Mode

## 5.2 Digital Microphone Controller (DMIC)

### 5.2.1 Overview

DMIC Controller supports a 2-channels digital microphone interface, the DMIC controller can output 128fs or 64fs (fs= ADC sample rate).

The DMIC includes the following features:

- Support up to 2 channels
- Support sample rate from 8KHz to 48KHz

### 5.2.2 DMIC Block Diagram

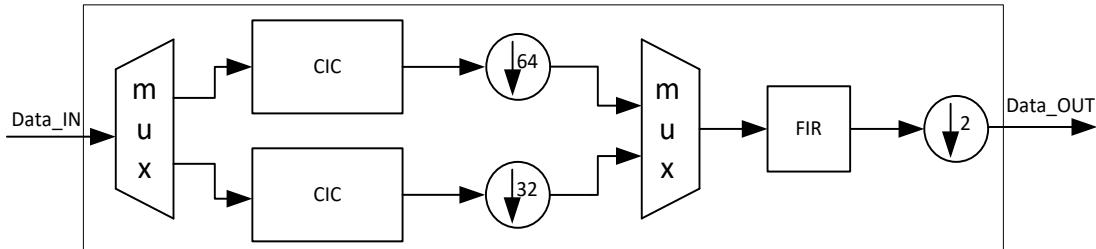


Figure 5-8 DMIC Block Diagram

### 5.2.3 DMIC Clock Source

DMIC get three different clocks. The following table describes the clock source for DMIC. Users can see **Clock Controller Unit (CCU)** for clock setting, configuration and gating information.

Clock Sources	Description
PLL_AUDIO	Audio Clock, default value is 24.571MHz
EXTERNAL_CLK0	Clock from External OSC0 for Audio

EXTERNAL_CLK1	Clock from External OSC1 for Audio
---------------	------------------------------------

**Table 5-24 DMIC Clock Source**

## 5.2.4 Register List

Module Name	Base Address	
Digital Microphone Controller	0x40042400	
Register Name	Offset Address	Description
DMIC_EN	0x00	DMIC Enable Control Register
DMIC_SR	0x04	DMIC Sample Rate Register
DMIC_CTR	0x08	DMIC Control Register
DMIC_DATA	0x10	DMIC DATA Register
DMIC_INTC	0x14	DMIC Interrupt Control Register
DMIC_INTS	0x18	DMIC Interrupt Status Register
DMIC_FIFO_CTR	0x1C	DMIC FIFO Control Register
DMIC_FIFO_STA	0x20	DMIC FIFO Status Register
DMIC_CH_NUM	0x24	DMIC Channel Numbers Register
DMIC_CH_MAP	0x28	DMIC Channel Mapping Register
DMIC_CNT	0x2C	DMIC Counter Register
DATA0_DATA1_VOL_CTR	0x30	DATA0 And DATA1 Volume Control Register
DATA2_DATA3_VOL_CTR	0x34	DATA2 And DATA3 Volume Control Register
HPF_EN_CTR	0x38	High Pass Filter Enable Control Register
HPF_COEF_REG	0x3C	High Pass Filter Coef Register
HPF_GAIN_REG	0x40	High Pass Filter Gain Register

**Table 5-25 DMIC Register List**

## 5.2.5 Register Description

### 5.2.5.1 DMIC Enable Control Register

<b>Address : 0x0000</b>			<b>Name:</b> DMIC_EN <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:9	/	/	/
8	R/W	0x0	GLOBE_EN DMIC Globe Enable 0: Disable 1: Enable
7	R/W	0x0	DATA3_CHR_EN DATA3 Right Channel 1 Enable

			0: Disable 1: Enable
6	R/W	0x0	DATA3_CHL_EN DATA3 Left Channel Enable 0: Disable 1: Enable
5	R/W	0x0	DATA2_CHR_EN DATA2 Right Channel 1 Enable 0: Disable 1: Enable
4	R/W	0x0	DATA2_CHL_EN DATA2 Left Channel Enable 0: Disable 1: Enable
3	R/W	0x0	DATA1_CHR_EN DATA1 Right Channel 1 Enable 0: Disable 1: Enable
2	R/W	0x0	DATA1_CHL_EN DATA1 Left Channel Enable 0: Disable 1: Enable
1	R/W	0x0	DATA0_CHR_EN DATA0 Right Channel 1 Enable 0: Disable 1: Enable
0	R/W	0x0	DATA0_CHL_EN DATA0 Left Channel Enable 0: Disable 1: Enable

**Table 5-26 DMIC Enable Control Register**

### 5.2.5.2DMIC Sample Rate Register

<b>Address : 0x0004</b>			<b>Name:</b> DMIC_SR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:3	/	/	/
2:0	R/W	0x0	DMIC_SR Sample Rate of DMIC 000: 48KHz 010: 24KHz

			100: 12KHz 110: Reserved 001: 32KHz 011: 16KHz 101: 8KHz 111: Reserved 44.1KHz/22.05KHz/11.025KHz can be supported by Audio PLL Configure Bit
--	--	--	---

**Table 5-27 DMIC Sample Rate Register****5.2.5.3 DMIC Control Register**

<b>Address:</b> 0x0008			<b>Name:</b> DMIC_CTR <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:11	/	/	/
10:9	R/W	0X0	<b>DMICFDT</b> DMIC FIFO Delay Time for writing Data after GLOBE_EN 00:5ms 01:10ms 10:200ms 11:30ms
8	R/W	0X0	<b>DMICDFEN</b> DMIC FIFO Delay Function for writing Data after GLOBE_EN 0: Disable 1: Enable
7	R/W	0X0	<b>DATA3 Left Data and Right Data Sweep Enable</b> 0: Disable 1: Enable
6	R/W	0X0	<b>DATA2 Left Data and Right Data Sweep Enable</b> 0: Disable 1: Enable
5	R/W	0X0	<b>DATA1 Left Data and Right Data Sweep Enable</b> 0: Disable 1: Enable
4	R/W	0X0	<b>DATA0 Left Data and Right Data Sweep Enable</b> 0: Disable 1: Enable
3:1	/	/	/
0	R/W	0X0	<b>DMIC Oversample Rate</b> 1: 64 (Support 16KHz~48KHz) 0: 128 (Support 8KHz~24KHz)

**Table 5-28 DMIC Control Register**

### 5.2.5.4 DMIC DATA Register

<b>Address:</b> 0x0010			<b>Name:</b> DMIC_DATA <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R	0X0	DMIC_DATA

**Table 5-29 DMIC DATA Register**

### 5.2.5.5 DMIC Interrupt Control Register

<b>Address:</b> 0x0014			<b>Name:</b> DMIC_INTC <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:3	/	/	/
2	R/W	0X0	FIFO_DRQ_EN DMIC FIFO Data Available DRQ Enable 0: Disable 1: Enable
1	R/W	0X0	FIFO_OVERRUN_IRQ_EN DMIC FIFO Over Run IRQ Enable 0: Disable 1: Enable
0	R/W	0X0	DATA_IRQ_EN DMIC FIFO Data Available IRQ Enable 0: Disable 1: Enable

**Table 5-30 DMIC Interrupt Control Register**

### 5.2.5.6 DMIC Interrupt Status Register

<b>Address:</b> 0x0018			<b>Name:</b> DMIC_INTS <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:2	/	/	/
1	R/W1	0X0	FIFO_OVERRUN_IRQ_PENDING DMIC FIFO Over Run Pending Interrupt 0: No Pending IRQ 1: FIFO Overrun Pending IRQ Write '1' to clear this interrupt or automatic clear if interrupt condition fails

0	R/ W1C	0X0	FIFO_DATA_IRQ_PENDING DMIC FIFO Data Available Pending Interrupt 0: No Pending IRQ 1: Data Available Pending IRQ Write '1' to clear this interrupt or automatic clear if interrupt condition fails
---	-----------	-----	--

**Table 5-31 DMIC Interrupt Status Register**

### 5.2.5.7 DMIC FIFO Control Register

<b>Address:</b> 0x001c			<b>Name:</b> DMIC_FIFO_CTR <b>Default:</b> 0x0000_0040
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31	R/W1 C	0X0	DMIC_FIFO_FLUSH DMIC FIFO Flush Write '1' to flush TX FIFO, self clear to '0'
30:10	/	/	/
9	R/W	0X0	FIFO_MODE RX FIFO Output Mode (Mode 0, 1) 0: Expanding '0' at LSB of TX FIFO register 1: Expanding received sample sign bit at MSB of TX FIFO register For 24-bits received audio sample: Mode 0: RXDATA[31:0] = {FIFO_O[23:0], 8'h0} Mode 1: Reserved For 16-bits received audio sample: Mode 0: RXDATA[31:0] = {FIFO_O[23:8], 16'h0} Mode 1: RXDATA[31:0] = {16{FIFO_O[23]}, FIFO_O[23:8]}
8	R/W	0x0	Sample Resolution 0 : 16 Bit 1 : 24 Bit
7:0	R/W	0X40	FIFO_TRG_LEVEL FIFO Trigger Level (TRLV[7:0]) Interrupt and DMA request trigger level for DMIC FIFO normal condition IRQ/DRQ Generated when WLEVEL > TRLV[7:0]) Notes: WLEVEL represents the number of valid samples in the DMIC FIFO

**Table 5-32 DMIC FIFO Control Register**

### 5.2.5.8 DMIC FIFO Status Register

<b>Address:</b> 0x0020	<b>Name:</b> DMIC_FIFO_STA <b>Default:</b> 0x0000_0000
------------------------	---

Field	Type	Default	Description
31:8	/	/	/
7:0	R/W	0X0	DMIC_DATA_CNT DMIC FIFO Available Sample Word Counter

**Table 5-33 DMIC FIFO Status Register**

### 5.2.5.9 DMIC Channel Numbers Register

<b>Address:</b> 0x0024			<b>Name:</b> DMIC_CH_NUM <b>Default:</b> 0x0000_0001
Field	Type	Default	Description
31:3	/	/	/
2:0	R/W	0X1	DMIC_CH_NUM DMIC Enable Channel Numbers are (N+1)

**Table 5-34 DMIC Channel Numbers Register**

### 5.2.5.10 DMIC Channel Mapping Register

<b>Address:</b> 0x0028			<b>Name:</b> DMIC_CH_NUM <b>Default:</b> 0x0
Field	Type	Default	Description
31:4	/	/	/
3:0	R/W	0x0	DMIC_CHO_MAP DMIC Channel0 Mapping 0: DATA0 Left Channel 1: DATA0 Right Channel 2: DATA1 Left Channel 3: DATA1 Right Channel 4: DATA2 Left Channel 5: DATA2 Right Channel 6: DATA3 Left Channel 7: DATA3 Right Channel

**Table 5-35 DMIC Channel Mapping Register**

### 5.2.5.11 DMIC Counter Register

<b>Address:</b> 0x002c			<b>Name:</b> DMIC_CNT <b>Default:</b> 0x0000_0000
Field	Type	Default	Description
31:0	R/W	0X0	DMIC_CUNT RX Sample Counter The audio sample number of writing into RXFIFO. When one sample is

			written by Digital Audio Engine, the RX sample counter register increases by one. The RX sample counter register can be set to any initial value at any time. After been updated by the initial value, the counter register should count on base of this initial value  Notes: It is used for Audio/ Video Synchronization
--	--	--	--

Table 5-36 DMIC Counter Register

### 5.2.5.12 DATA0 and DATA1 Volume Control Register

<b>Address:</b> 0x0030			<b>Name:</b> DATA0_DATA1_VOL_CTR <b>Default:</b> 0xa0a0_a0a0
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:24	R/W	0XA0	DATA1L_VOL (-119.25dB To 71.25dB, 0.75dB/Step) 0x00: Mute 0x01: -119.25Db ..... 0x9F = -0.75dB 0xA0 = 0dB 0xA1 = 0.75dB ..... 0xFF = 71.25dB
23:16	R/W	0XA0	DATA1R_VOL (-119.25dB to 71.25dB, 0.75dB/Step) 0x00: Mute 0x01: -119.25Db ..... 0x9F = -0.75dB 0xA0 = 0dB 0xA1 = 0.75dB ..... 0xFF = 71.25dB
15:8	R/W	0XA0	DATA0L_VOL (-119.25dB to 71.25dB, 0.75dB/Step) 0x00: Mute 0x01: -119.25Db ..... 0x9F = -0.75dB 0xA0 = 0dB 0xA1 = 0.75dB ..... 0xFF = 71.25dB
7:0	R/W	0XA0	DATA0R_VOL

			(-119.25dB to 71.25dB, 0.75dB/Step) 0x00: Mute 0x01: -119.25Db ..... 0x9F = -0.75dB 0xA0 = 0dB 0xA1 = 0.75dB ..... 0xFF = 71.25dB
--	--	--	---

**Table 5-37 DATA0 and DATA1 Volume Control Register****5.2.5.13 DATA2 and DATA3 Volume Control Register**

<b>Address:</b> 0x0034			<b>Name:</b> DATA2_DATA3_VOL_CTR <b>Default:</b> 0xA0A0_A0A0
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:24	R/W	0XA0	DATA3L_VOL (-119.25dB to 71.25dB, 0.75dB/Step) 0x00: Mute 0x01: -119.25Db ..... 0x9F = -0.75dB 0xA0 = 0dB 0xA1 = 0.75dB ..... 0xFF = 71.25dB
23:16	R/W	0XA0	DATA3R_VOL (-119.25dB to 71.25dB, 0.75dB/Step) 0x00: Mute 0x01: -119.25Db ..... 0x9F = -0.75dB 0xA0 = 0dB 0xA1 = 0.75dB ..... 0xFF = 71.25dB
15:8	R/W	0XA0	DATA2L_VOL (-119.25dB to 71.25dB, 0.75dB/Step) 0x00: Mute 0x01: -119.25Db ..... 0x9F = -0.75dB 0xA0 = 0dB

			0xA1 = 0.75dB ..... 0xFF = 71.25dB
7:0	R/W	0XA0	DATA2R_VOL (-119.25dB to 71.25dB, 0.75dB/Step) 0x00: Mute 0x01: -119.25Db ..... 0x9F = -0.75dB 0xA0 = 0dB 0xA1 = 0.75dB ..... 0xFF = 71.25dB

**Table 5-38 DATA2 and DATA3 Volume Control Register**

#### 5.2.5.14 High Pass Filter Enable Control Register

Address: 0x0038			Name: HPF_EN_CTR Default: 0x0000_0000
Field	Type	Default	Description
31:8	/	/	/
7	R/W	0x0	HPF_DATA3_CHR_EN DATA3 Right Channel 1 Enable 0: Disable 1: Enable
6	R/W	0x0	HPF_DATA3_CHL_EN DATA3 Left Channel Enable 0: Disable 1: Enable
5	R/W	0x0	HPF_DATA2_CHR_EN DATA2 Right Channel 1 Enable 0: Disable 1: Enable
4	R/W	0x0	HPF_DATA2_CHL_EN DATA2 Left Channel Enable 0: Disable 1: Enable
3	R/W	0x0	HPF_DATA1_CHR_EN DATA1 Right Channel 1 Enable 0: Disable 1: Enable
2	R/W	0x0	HPF_DATA1_CHL_EN

			DATA1 Left Channel Enable 0: Disable 1: Enable
1	R/W	0x0	HPF_DATA0_CHR_EN DATA0 Right Channel 1 Enable 0: Disable 1: Enable
0	R/W	0x0	HPF_DATA0_CHL_EN DATA0 Left Channel Enable 0: Disable 1: Enable

**Table 5-39 High Pass Filter Enable Control Register**

### 5.2.5.15 High Pass Filter Coef Register

<b>Address:</b> 0x003C			<b>Name:</b> HPF_COEF_REG <b>Default:</b> 0x00FF_AA45
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0x00FFAA45	

**Table 5-40 High Pass Filter Coef Register**

### 5.2.5.16 High Pass Filter Gain Register

<b>Address:</b> 0x0040			<b>Name:</b> HPF_GAIN_REG <b>Default:</b> 0x00FF_D522
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0x00FFD522	

**Table 5-41 High Pass Filter Gain Register**

# Chapter 6 Camera

## 6.1 Camera Serial Interface (CSI)

### 6.1.1 Overview

The Camera Serial Interface (CSI) is a parallel image input interface. It includes the following features:

- 8 bits input data
- support CCIR656 protocol for NTSC and PAL
- pass data direct to Ping-Pang buffer
- pass raw data direct to memory

#### 6.1.1.1 CSI Block Diagram

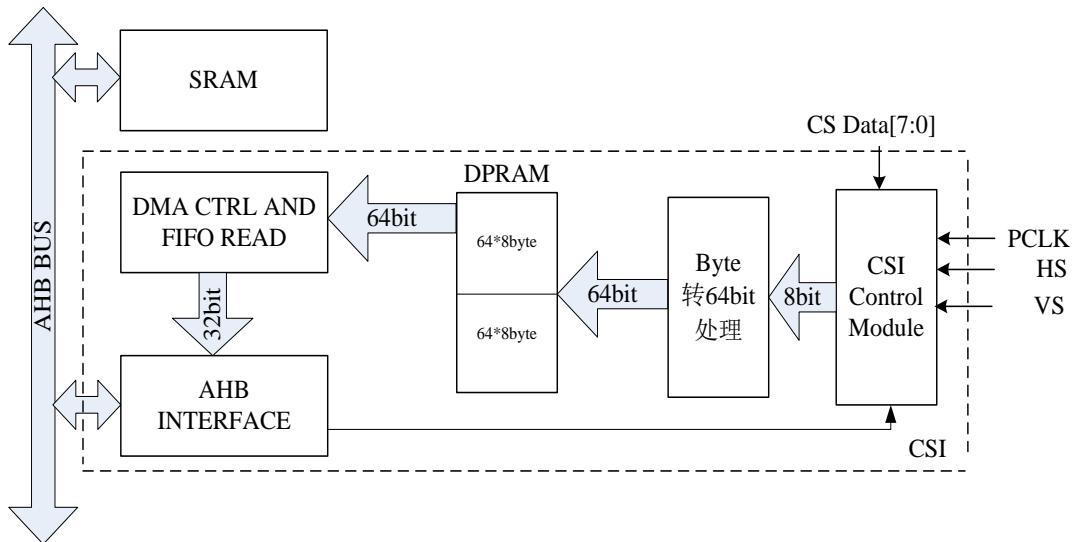


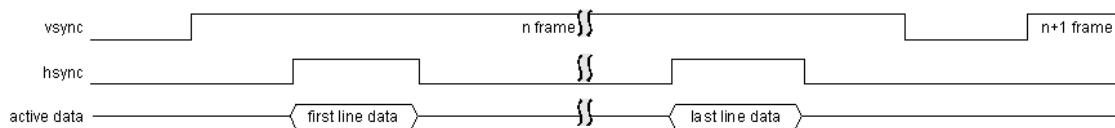
Figure 6-1 CSI Block Diagram

#### 6.1.1.2 CSI DATA Port

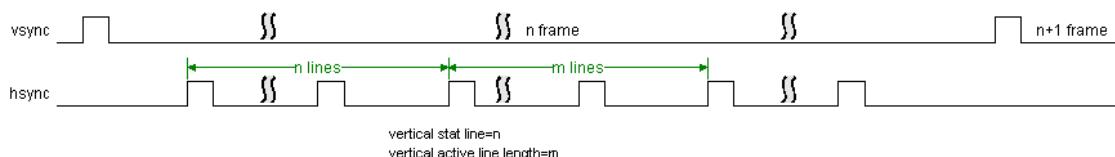
	Bayer	YCbCr (YUV)	Interlaced	Pass-through
FIFO0	RGB pixel data	YUV pixel data		All pixel data

Table 6-1 CSI DATA Port

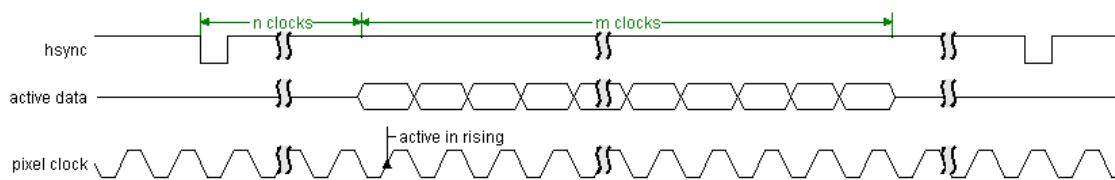
### 6.1.1.3 CSI Timing



Vref= positive; Href= positive



Vertical size setting



horizontal start clock = n  
horizontal active clock's length = m

Horizontal size setting and pixel clock timing (Href= positive)

### 6.1.2 Register List

Module Name	Base Address	
CSI Controller	0x40007000	
Register Name	Offset Address	Description
CSI_EN_REG	0X000	CSI enable register
CSI_CFG_REG	0X004	CSI configuration register
CSI_CAP_REG	0X008	CSI capture control register
CSI_SCALE_REG	0X00C	CSI scale register
CSI_F0_BUFA_REG	0X800	CSI FIFO 0 output buffer-A address register
CSI_F0_BUFB_REG	0XA00	CSI FIFO 0 output buffer-B address register
CSI_BUF_CTL_REG	0X028	CSI output buffer control register
CSI_BUF_STA_REG	0X02C	CSI status register
CSI_INT_EN_REG	0X030	CSI interrupt enable register
CSI_INT_STA_REG	0X034	CSI interrupt status register
CSI_HSIZE_REG	0X040	CSI horizontal size register
CSI_VSIZE_REG	0X044	CSI vertical size register

CSI_BUF_LEN_REG	0X048	CSI line buffer length register
-----------------	-------	---------------------------------

**Table 6-2 CSI Register List**

## 6.1.3 Register Description

### 6.1.3.1 CSI Enable Register

<b>Address :</b> 0x0000			<b>Name:</b> CSIO_EN_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:1	/	/	/
0	R/W	0	CSI_EN  Enable 0: Reset and disable the CSI module 1: Enable the CSI module

**Table 6-3 CSI Enable Register**

### 6.1.3.2 CSI Configuration Register

<b>Address :</b> 0x0004			<b>Name:</b> CSIO_CFG_REG <b>Default:</b> 0x0030_0205
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:23	/	/	/
22:20	R/W	3	INPUT_FMT  Input data format 000: RAW stream others: reserved
19:12	R/W		
11:10	R/W	0	FIELD_SEL  Field selection. Applies to CCIR656 interface only. 00: start capturing with field 1. 01: start capturing with field 2. 10: start capturing with either field. 11: reserved
09:03	/	/	/
02	R/W	1	VREF_POL  Vref polarity 0: negative 1: positive  This register is not apply to CCIR656 interface.

01	R/W	0	HERF_POL Href polarity 0: negative 1: positive This register is not apply to CCIR656 interface.
00	R/W	1	CLK_POL Data clock type 0: active in falling edge 1: active in rising edge

**Table 6-4 CSI Configuration Register**

### 6.1.3.3 CSI Capture Register

Address : 0x0008			Name: CSIO_CAP_REG Default: 0x0000_0000
Field	Type	Default	Description
31:1	/	/	/
1	R/W	0	VCAP_ON Video capture control: Capture the video image data stream. 0: Disable video capture If video capture is in progress, the CSI stops capturing image data at the end of the current frame, and all of the current frame data is wrote to output FIFO. 1: Enable video capture The CSI starts capturing image data at the start of the next frame.
0	R/W	0	SCAP_ON Still capture control: Capture a single still image frame. 0: Disable still capture. 1: Enable still capture The CSI module starts capturing image data at the start of the next frame. The CSI module captures only one frame of image data. This bit is self clearing and always reads as a 0.

**Table 6-5 CSI Capture Register**

### 6.1.3.4 CSI Horizontal Scale Register

Address : 0x000C			Name: CSIO_SYNC_CNT_REG Default: 0x0f00_ffff
Field	Type	Default	Description
31:28	/	/	/
27:24	R/W	F	VER_MASK Vertical (line) mask. Every 4-line is a mask group. Bit 24 mask the first line,

			bit 25 mask the second line, and so on. Mask bit = 0 means discarding this line data.
23:16	/	/	/
15:0	R/W	FFFF	HOR_MASK Horizontal (datastream) mask. Every 16-byte is a mask group. Bit 0 mask the first byte, bit 1 mask the second byte, and so on. Mask bit = 0 means discarding this byte from the datastream.

**Table 6-6 CSI Horizontal Scale Register**

### 6.1.3.5CSI CH0 FIFO 0 Output Buffer-A Address Register

<b>Address :</b> 0x0800			<b>Name:</b> CSIO_C0_F0_BUFA_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0	COFO_BUFA  FIFO 0 output buffer-A address

**Table 6-7 CSI CH0 FIFO 0 Output Buffer-A Address Register**

### 6.1.3.6CSI CH0 FIFO 0 Output Buffer-B Address Register

<b>Address :</b> 0x0a00			<b>Name:</b> CSIO_C0_F0_BUFB_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0	COFO_BUFB  FIFO 0 output buffer-B address

**Table 6-8 CSI CH0 FIFO 0 Output Buffer-B Address Register**

### 6.1.3.7CSI CH0 Output Buffer Control Register

<b>Address :</b> 0x0028			<b>Name:</b> CSI_BUF_CTL_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:03	/	/	/
02	R/W	0	DBN  Buffer selected at next storing for CSI 0: Next buffer selection is buffer-A 1: Next buffer selection is buffer-B
01	R	0	DBS  output buffer selected status 0: Selected output buffer-A

			1: Selected output buffer-B
00	R/W	0	<p>DBE Double buffer mode enable 0: disable 1: enable If the double buffer mode is disabled, the buffer-A will be always selected by CSI module.</p>

**Table 6-9 CSI CH0 Output Buffer Control Register**

### 6.1.3.8CSI CH0 Status Register

<b>Address : 0x002C</b>			<b>Name:</b> CSIO_CO_CAP_STA_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:08	R	0	<p>LUM_STATIS luminance statistical value When frame done interrupt flag come, value is ready and will last until next frame done. For raw data, value = (G&gt;&gt;1+R+G)&gt;&gt;8</p>
07:02	/	/	/
1	R	0	<p>VCAP_STA Video capture in progress Indicates the CSI is capturing video image data (multiple frames). The bit is set at the start of the first frame after enabling video capture. When software disables video capture, it clears itself after the last pixel of the current frame is captured.</p>
0	R	0	<p>SCAP_STA Still capture in progress Indicates the CSI is capturing still image data (single frame). The bit is set at the start of the first frame after enabling still frame capture. It clears itself after the last pixel of the first frame is captured. For CCIR656 interface, if the output format is frame planar YCbCr 420 mode, the frame end means the field2 end, the other frame end means filed end.</p>

**Table 6-10 CSI CH0 Status Register**

### 6.1.3.9CSI CH0 Interrupt Enable Register

<b>Address : 0x0030</b>			<b>Name:</b> CSIO_CO_INT_EN_REG <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:10	/	/	/

09	R/W	0	second FIFO SRAM data ready int enable
08	R/W	0	first FIFO SRAM data ready int enable
07	R/W	0	VS_INT_EN vsync flag The bit is set when vsync come. And at this time load the buffer address for the coming frame. So after this irq come, change the buffer address could only effect next frame
06	R/W	0	HB_OF_INT_EN Hblank FIFO overflow The bit is set when 3 FIFOs still overflow after the hblank.
05	R/W	0	PRTC_ERR_INT_EN
04	R/W	0	/
03	R/W	0	/
02	R/W	0	FIFO0_OF_INT_EN FIFO 0 overflow The bit is set when the FIFO 0 become overflow.
01	R/W	0	FD_INT_EN Frame done Indicates the CSI has finished capturing an image frame. Applies to video capture mode. The bit is set after each completed frame capturing data is wrote to buffer as long as video capture remains enabled.
00	R/W	0	CD_INT_EN Capture done Indicates the CSI has completed capturing the image data. For still capture, the bit is set when one frame data has been wrote to buffer. For video capture, the bit is set when the last frame has been wrote to buffer after video capture has been disabled. For CCIR656 interface, if the output format is frame planar YCbCr 420 mode, the frame end means the field2 end, the other frame end means field end.

**Table 6-11 CSI CH0 Interrupt Enable Register**

### 6.1.3.10 CSI CH0 Interrupt Status Register

Address : 0x0034			Name: CSIO_C0_INT_STA_REG Default: 0x0000_0000
Field	Type	Default	Description
31:10	/	/	/
09	R/W	0	second FIFO SRAM data ready int pending
08	R/W	0	first FIFO SRAM data ready int pending
07	R/W	0	VS_PD

			vsync flag
06	R/W	0	HB_OF_PD Hblank FIFO overflow
05	R/W	0	PRTC_ERR_PD
04	R/W	0	/
03	R/W	0	/
02	R/W	0	FIFO0_OF_PD FIFO 0 overflow
01	R/W	0	FD_PD Frame done
00	R/W	0	CD_PD Capture done

**Table 6-12 CSI CH0 Interrupt Status Register**

### 6.1.3.11 CSI CH0 Horizontal Size Register

<b>Address :</b> 0x0040			<b>Name:</b> CSIO_CO_HSIZE_REG <b>Default:</b> 0x0500_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:29	/	/	/
28:16	R/W	500	HOR_LEN  Horizontal pixel unit length. Valid pixel of a line.
15:13	/	/	/
12:00	R/W	0	HOR_START  Horizontal pixel unit start. Pixel is valid from this pixel.

**Table 6-13 CSI CH0 Horizontal Size Register**

### 6.1.3.12 CSI CH0 Vertical Size Register

<b>Address :</b> 0x0044			<b>Name:</b> CSIO_CO_VSIZE_REG <b>Default:</b> 0x01E0_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:29	/	/	/
28:16	R/W	1E0	VER_LEN  Vertical line length. Valid line number of a frame.
15:13	/	/	/
12:0	R/W	0	VER_START  Vertical line start. data is valid from this line.

**Table 6-14 CSI CH0 Vertical Size Register**

### 6.1.3.13 CSI CH0 Buffer Length Register

<b>Address</b> : 0x0048			<b>Name:</b> CSIO_C0_BUF_LEN_REG <b>Default:</b> 0x0140_0280
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:13	/	/	/
12:00	R/W	280	<p><b>BUF_LEN</b></p> <p>Buffer length of luminance Y in a line. Unit is byte.</p>

**Table 6-15 CSI CH0 Buffer Length Register**

### 6.1.3.14 CSI True Data Number of FIFO SRAM Register

<b>Address</b> : 0x004C			<b>Name:</b> TRUE_DATA_NUM <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:26	/	/	/
25:16	R/W	0	the true data number of first FIFO SRAM
9:0	R/W	0	the true data number of second FIFO SRAM

**Table 6-16 CSI FIFO SRAM Data Ready Int Enable Register**

### 6.1.3.15 CSI JPEG Mode Select Register

<b>Address</b> : 0x0050			<b>Name:</b> JPEG_MOD_SEL <b>Default:</b> 0x0000_0000
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:1	/	/	/
0	R/W	0	1: Bypass JPEG Mode 0: JPEG Mode

**Table 6-17 CSI JPEG Mode Select Register**

# Chapter 7 GPIO

The XR871 GPIO unit provides as many as 31 GPIO (General Purpose IO) pins. All ports are brought out of the device using alternate function multiplexing. The GPIO function can be multiplexed on a multi-function I/O pin by selecting the GPIO alternate function in the GPIO Controller registers.

There are two types of GPIO designs in XR871: GPIO and AGPIO. Each GPIO can be configured with the following options:

- Input / Output / Floating(Hi-Z) mode
- Input mode: Pull-up or Pull-down
- Output mode: Active driving
- Pull-up/down control: the pull-up and pull-down resistance is  $90\text{K}\Omega$  with  $\pm 30\%$  variation over PVT condition
- External Interrupt IO with 5 trigger modes: high-level, low-level, rising edge, falling edge, double edge

The digital IO AGPIO function is equivalent to GPIO as shown above. A dedicated internal control signal is used to select between the digital and analog functions. These IOs are multiplexed with 8 channels ADC.

## 7.1.1 Register List

Module Name	Base Address	
GPIO Controller	0x40050000	
Register Name	Offset Address	Description
Pn_CFG0	$n*0x24 + 0x00$	Port n Configure Register 0 ( $n = 0,1$ )
Pn_CFG1	$n*0x24 + 0x04$	Port n Configure Register 1 ( $n = 0,1$ )
Pn_CFG2	$n*0x24 + 0x08$	Port n Configure Register 2 ( $n = 0,1$ )
Pn_CFG3	$n*0x24 + 0x0C$	Port n Configure Register 3 ( $n = 0,1$ )
Pn_DATA	$n*0x24 + 0x10$	Port n Data Register ( $n = 0,1$ )
Pn_DRVO	$n*0x24 + 0x14$	Port n Multi-Driving Register 0 ( $n = 0,1$ )
Pn_DRV1	$n*0x24 + 0x18$	Port n Multi-Driving Register 1 ( $n = 0,1$ )
Pn_PULL0	$n*0x24 + 0x1C$	Port n Pull Register 0 ( $n = 0,1$ )
Pn_PULL1	$n*0x24 + 0x20$	Port n Pull Register 1 ( $n = 0,1$ )
Pn_INT_CFG0	$0x200 + n*0x20 + 0x00$	PIO Interrupt Configure Register 0
Pn_INT_CFG1	$0x200 + n*0x20 + 0x04$	PIO Interrupt Configure Register 1
Pn_INT_CFG2	$0x200 + n*0x20 + 0x08$	PIO Interrupt Configure Register 2
Pn_INT_CFG3	$0x200 + n*0x20 + 0x0C$	PIO Interrupt Configure Register 3
Pn_INT_CTL	$0x200 + n*0x20 + 0x10$	PIO Interrupt Control Register
Pn_INT_STA	$0x200 + n*0x20 + 0x14$	PIO Interrupt Status Register
Pn_INT_DBC	$0x200 + n*0x20 + 0x18$	PIO Interrupt Debounce Register

Table 7-1 GPIO Register List

## 7.1.2 Register Description

### 7.1.2.1 Pn Configure Register 0

<b>Address :</b> n*0x24 + 0x00			<b>Name:</b> Pn_CFG0 <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
[4*m+3:4*m] (m = 0~7)	R/W	0x7	GPIO Group n IO m Config

Table 7-2 Pn Configure Register 0

### 7.1.2.2 Pn Configure Register 1

<b>Address :</b> n*0x24 + 0x04			<b>Name:</b> Pn_CFG1 <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
[4*(m-8)+3:4*(m-8)] (m = 8~15)	R/W	0x7	GPIO Group n IO m Config  <i>Note: The function numbers please refer to the AW1735 Pin Multiplexing. If this IO is not exist, this field is reserved.</i>

Table 7-3 Pn Configure Register 1

### 7.1.2.3 Pn Configure Register 2

<b>Address :</b> n*0x24 + 0x08			<b>Name:</b> Pn_CFG2 <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
[4*(m-16)+3:4*(m-16)] (m = 16~23)	R/W	0x7	GPIO Group n IO m Config  <i>Note: The function numbers please refer to the AW1735 Pin Multiplexing. If this IO is not exist, this field is reserved.</i>

Table 7-4 Pn Configure Register 2

### 7.1.2.4 Pn Configure Register 3

<b>Address :</b> n*0x24 + 0x0C			<b>Name:</b> Pn_CFG3 <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
[4*(m-24)+3:4*(m-24)]	R/W	0x7	GPIO Group n IO m Config  <i>Note: The function numbers please refer to the AW1735 Pin Multiplexing.</i>

(m = 24~31)		If this IO is not exist, this field is reserved.
-------------	--	--

**Table 7-5 Pn Configure Register 3**

### 7.1.2.5 Pn Data Register

<b>Address</b> : n*0x24 + 0x10			<b>Name:</b> Pn_DATA <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:0	R/W	0	DATA If the port is configured as input, the corresponding bit is the pin state. If the port is configured as output, the pin state is the same as the corresponding bit. The read bit value is the value setup by software. If the port is configured as functional pin, the undefined value will be read.

**Table 7-6 Pn Data Register**

### 7.1.2.6 Pn Multi-Driving Register 0

<b>Address</b> : n*0x24 + 0x14			<b>Name:</b> Pn_DRV0 <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
[2*m+1:2*m] (m = 15~0)	R/W	0x1	DRV_STRENGTH 00: level 0                  01: level 1 10: level 2                  11: level 3  <i>Note: This field is only valid when the related IO is exist.</i>

**Table 7-7 Pn Multi-Driving Register 0**

### 7.1.2.7 Pn Multi-Driving Register 1

<b>Address</b> : n*0x24 + 0x18			<b>Name:</b> Pn_DRV1 <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
[2*(m-16)+1:2*(m-16)] (m = 31~16)	R/W	0x1	DRV_STRENGTH 00: level 0                  01: level 1 10: level 2                  11: level 3  <i>Note: This field is only valid when the related IO is exist.</i>

**Table 7-8 Pn Multi-Driving Register 1**

### 7.1.2.8 Pn Pull Register 0

<b>Address :</b> n*0x24 + 0x1C			<b>Name:</b> Pn_PULL0 <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
[2*m+1:2*m] (m = 15~0)	R/W	0x0	PULL 00: NO PULL                    01: Pull-Up 10: Pull-Down                11: /
<i>Note: This field is only valid when the related IO is exist.</i>			

Table 7-9 Pn Pull Register 0

### 7.1.2.9 Pn Pull Register 1

<b>Address :</b> n*0x24 + 0x20			<b>Name:</b> Pn_PULL0 <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
[2*(m-16)+1:2*(m-16)] (m = 31~16)	R/W	0x0	PULL 00: NO PULL                    01: Pull-Up 10: Pull-Down                11: /
<i>Note: This field is only valid when the related IO is exist.</i>			

Table 7-10 Pn Pull Register 1

### 7.1.2.10 Pn External Interrupt Configure 0 Register

<b>Address :</b> 0x200 + 0x20*n + 0x00			<b>Name:</b> Pn_EINT_CFG0 <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
[4*m+3:4*m] (m = 0~7)	R/W	0x0	EINT_MODE 0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
<i>Note: This field is only valid when the related IO is exist.</i>			

Table 7-11 Pn External Interrupt Configure 0 Register

### 7.1.2.11 Pn External Interrupt Configure 1 Register

<b>Address</b> : 0x200 + 0x20*n + 0x04			<b>Name:</b> Pn_EINT_CFG1 <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
[4*(m-8)+3:4*(m-8)] (m = 8~15)	R/W	0x0	<p>EINT_MODE</p> <p>0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /</p> <p><i>Note: This field is only valid when the related IO is exist.</i></p>

Table 7-12 Pn External Interrupt Configure 1 Register

### 7.1.2.12 Pn External Interrupt Configure 2 Register

<b>Address</b> : 0x200 + 0x20*n + 0x08			<b>Name:</b> Pn_EINT_CFG2 <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
[4*(m-16)+3:4*(m-16)] (m = 16~23)	R/W	0x0	<p>EINT_MODE</p> <p>0: Positive Edge 1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /</p> <p><i>Note: This field is only valid when the related IO is exist.</i></p>

Table 7-13 Pn External Interrupt Configure 2 Register

### 7.1.2.13 Pn External Interrupt Configure 3 Register

<b>Address</b> : 0x200 + 0x20*n + 0x0C			<b>Name:</b> Pn_EINT_CFG3 <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
[4*(m-24)+3:4*(m-24)]	R/W	0x0	<p>EINT_MODE</p> <p>0: Positive Edge</p>

(m = 24~31)			1: Negative Edge 2: High Level 3: Low Level 4: Double Edge others: /
<i>Note: This field is only valid when the related IO is exist.</i>			

**Table 7-14 Pn External Interrupt Configure 3 Register****7.1.2.14 Pn External Interrupt Control Register**

<b>Address</b> : 0x200 + 0x20*n + 0x10			<b>Name:</b> Pn_EINT_CTRL <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
m (m = 31~0)	R/W	0x0	EINT_IRQEN 0: disable 1: enable <i>Note: This field is only valid when the related IO is exist.</i>

**Table 7-15 Pn External Interrupt Control Register****7.1.2.15 Pn External Interrupt Status Register**

<b>Address</b> : 0x200 + 0x20*n + 0x14			<b>Name:</b> Pn_EINT_ST <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
[m] (m = 31~0)	R/W	0x0	EINT IRQST 0: nothing 1: irq pending write 1 to clear <i>Note: This field is only valid when the related IO is exist.</i>

**Table 7-16 Pn External Interrupt Status Register****7.1.2.16 Pn External Interrupt Debounce Register**

<b>Address</b> : 0x200 + 0x20*n + 0x18			<b>Name:</b> Pn_DBC <b>Default:</b>
<b>Field</b>	<b>Type</b>	<b>Default</b>	<b>Description</b>
31:7	/	/	/
6:4	R/W	0	Debounce Clock Pre-Scale n The selected clock is prescaled by 2^n
3:1	/	/	/

0	R/W	0x0	PIO_INT_CLK_SEL 0: LFCLK 1: HFCLK	
---	-----	-----	---	--

**Table 7-17 Pn External Interrupt Debounce Register**

### 7.1.3 Pin Multiplexing

GPIO	FUNC1	FUNC2	FUNC3	FUNC4	FUNC5
PA00	SPI1_MOSI	SD_CMD	UART0_TX	CSI_D0	EINTA0
PA01	SPI1_MISO	SD_DATA0	UART0_RX	CSI_D1	EINTA1
PA02	SPI1_CLK	SD_CLK	TWI1_SCL	CSI_D2	EINTA2
PA03	SPI1_CS0	SD_DATA1	TWI1_SDA	CSI_D3	EINTA3
PA04	UART1_CTS	SD_DATA2	TWI0_SCL	CSI_D4	EINTA4
PA05	UART1_RTS	SD_DATA3	TWI0_SDA	CSI_D5	EINTA5
PA06	UART1_TX	SPI1_CS1	TWI0_SCL	CSI_D6	EINTA6
PA07	UART1_RX	SPI1_CS2	TWI0_SDA	CSI_D7	EINTA7
PA08	ADC_CH0	PWM0/ECT0	TWI1_SCL	CSI_PCLK	EINTA8
PA09	ADC_CH1	PWM1/ECT1	TWI1_SDA	CSI_MCLK	EINTA9
PA10	ADC_CH2	PWM2/ECT2	DMIC_CLK	CSI_HSYNC	EINTA10
PA11	ADC_CH3	PWM3/ECT3	DMIC_DATA	CSI_VSYNC	EINTA11
PA12	ADC_CH4	PWM4/ECT4	I2S_MCLK	IR_TX	EINTA12
PA13	ADC_CH5	PWM5/ECT5	I2S_BCLK	32KOSCO	EINTA13
PA14	ADC_CH6	PWM6/ECT6	I2S_DI	IR_RX	EINTA14
PA15	ADC_CH7	PWM7/ECT7	I2S_DO	UART1_CTS	EINTA15
PA16	IR_TX	IR_RX	I2S_LRCLK	UART1_RTS	EINTA16
PA17	TWI0_SCL	IR_RX	TWI1_SCL	UART1_TX	EINTA17
PA18	TWI0_SDA	IR_TX	TWI1_SDA	UART1_RX	EINTA18
PA19	NUART_CTS		PWM0/ECT0	SPI1_MOSI	EINTA19
PA20	NUART_RTS		PWM1/ECT1	SPI1_MISO	EINTA20
PA21	NUART_TX	DMIC_CLK	PWM2/ECT2	SPI1_CLK	EINTA21
PA22	NUART_RX	DMIC_DATA	PWM3/ECT3	SPI1_CS0	EINTA22
PB00	UART0_TX		PWM4/ECT4		EINTB0
PB01	UART0_RX		PWM5/ECT5		EINTB1
PB02	SWD_TMS		PWM6/ECT6		EINTB2
PB03	SWD_TCK		PWM7/ECT7		EINTB3
PB04	SPI0_MOSI				EINTB4
PB05	SPI0_MISO				EINTB5
PB06	SPI0_CS0				EINTB6
PB07	SPI0_CLK				EINTB7

**Table 7-18 Pin Multiplexing**

