

# W7500x

## Reference Manual

Version 1.1.2



<http://www.wiznet.io>

# 1 Table of Contents

<b>1</b>	<b>Table of Contents</b>	<b>2</b>
<b>2</b>	<b>List of table</b>	<b>19</b>
<b>3</b>	<b>List of figures</b>	<b>21</b>
<b>4</b>	<b>Documentation conventions</b>	<b>23</b>
4.1	Glossary	23
4.2	Register Bit Conventions	25
<b>5</b>	<b>System and memory overview</b>	<b>26</b>
5.1	System architecture	26
5.2	Memory organization	27
Introduction		27
Memory map		28
<b>6</b>	<b>System configuration controller (SYSCFG)</b>	<b>29</b>
<b>7</b>	<b>Interrupt and events</b>	<b>29</b>
7.1	Nested vectored interrupt controller (NVIC)	29
NVIC main features		29
SysTick calibration value register		29
Interrupt and exception vectors		29
7.2	Event	30
<b>8</b>	<b>Power supply</b>	<b>31</b>
8.1	Introduction	31
8.2	Voltage regulator	31
8.3	Low-power modes	31
Sleep mode		32
Peripheral clock gating		32
<b>9</b>	<b>System tick timer</b>	<b>33</b>
9.1	Introduction	33
9.2	Features	33
9.3	Functional description	33
9.4	Registers (Base : 0xE000_E000)	34
System Timer control and status register (SYST_CSR)		34
SysTick Reload Value Register (SYST_RVR)		34
SysTick Current Value Register (SYST_CVR)		35
SysTick Calibration Value Register (SYST_CALIB)		35
<b>10</b>	<b>Booting Sequence</b>	<b>36</b>
<b>11</b>	<b>Embedded Flash memory</b>	<b>37</b>
11.1	Flash main features	37
11.2	Flash memory organization	37

---

12 Clock Reset generator (CRG).....	39
12.1    Introduction .....	39
12.2    Features .....	39
Reset.....	39
Clock.....	40
12.3    Functional description .....	41
External Oscillator Clock .....	41
RC oscillator clock .....	42
PLL.....	42
Generated clock.....	42
12.4    Registers (Base address : 0x4100_1000).....	43
OSC power down register (OSC_PDR).....	43
PLL power down register (PLL_PDR).....	43
PLL frequency calculating register (PLL_FCR) .....	43
PLL output enable register (PLL_OER) .....	44
PLL bypass register (PLL_BPR) .....	44
PLL input clock source select register (PLL_IFSR).....	45
FCLK source select register (FCLK_SSR).....	45
FCLK prescale value select register (FCLK_PVSR) .....	46
SSPCLK source select register (SSPCLK_SSR) .....	46
SSPCLK prescale value select register (SSPCLK_PVSR) .....	47
ADCCLK source select register (ADCCLK_SSR) .....	47
ADCCLK prescale value select register (ADCCLK_PVSR).....	47
TIMER0CLK source select register (TIMER0CLK_SSR).....	48
TIMER0CLK prescale value select register (TIMER0CLK_PVSR) .....	48
TIMER1CLK source select register (TIMER1CLK_SSR).....	49
TIMER1CLK prescale value select register (TIMER1CLK_PVSR) .....	49
PWM0CLK source select register (PWM0CLK_SSR).....	50
PWM0CLK prescale value select register (PWM0CLK_PVSR) .....	50
PWM1CLK source select register (PWM1CLK_SSR).....	51
PWM1CLK prescale value select register (PWM1CLK_PVSR) .....	51
PWM2CLK source select register (PWM2CLK_SSR).....	52
PWM2CLK prescale value select register (PWM2CLK_PVSR) .....	52
PWM3CLK source select register (PWM3CLK_SSR) .....	53
PWM3CLK prescale value select register (PWM3CLK_PVSR) .....	54
PWM4CLK source select register (PWM4CLK_SSR) .....	54
PWM4CLK prescale value select register (PWM4CLK_PVSR) .....	55
PWM5CLK source select register (PWM5CLK_SSR) .....	55
PWM5CLK prescale value select register (PWM5CLK_PVSR) .....	56

---

---

PWM6CLK source select register (PWM6CLK_SSR) .....	56
PWM6CLK prescale value select register (PWM6CLK_PVSR) .....	57
PWM7CLK source select register (PWM7CLK_SSR) .....	57
PWM7CLK prescale value select register (PWM7CLK_PVSR) .....	58
RTC High Speed source select register (RTC_HS_SSR) .....	58
RTC High Speed prescale value select register (RTC_HS_PVSR) .....	59
RTC source select register (RTC_SSR) .....	59
WDOGCLK High Speed source select register (WDOGCLK_HS_SSR).....	60
WDOGCLK High Speed prescale value select register (WDOGCLK_HS_PVSR) .....	60
WDOGCLK clock source select register (WDOGCLK_SSR).....	61
UARTCLK source select register (UARTCLK_SSR) .....	61
UARTCLK prescale value select register (UARTCLK_PVSR).....	62
MIICLK enable control register (MIICLK_ECR) .....	62
Monitoring Clock source select register (MONCLK_SSR).....	63
<b>12.5 Register map .....</b>	<b>64</b>
<b>13 Tcp/ip core Offload Engine (TOE) .....</b>	<b>66</b>
13.1 Introduction .....	66
13.2 Features .....	66
13.3 Functional description .....	67
13.4 TOE Memory map.....	67
Common register map.....	69
Socket register map .....	69
Memory.....	70
13.5 Common register (Base : 0x4600_0000).....	72
VERSIONR (TOE Version Register).....	72
TCKCNTR (Ticker Counter Register) .....	72
IR (Interrupt Register).....	73
IMR (Interrupt Mask Register) .....	73
IRCR (Interrupt Clear Register) .....	74
SIR (Socket Interrupt Register) .....	75
SIMR (Socket Interrupt Mask Register).....	75
MR (Mode Register) .....	76
PTIMER (PPP Link Control Protocol Request Timer Register) .....	77
PMAGICR (PPP Link Control Protocol Magic number Register) .....	77
PHAR (Destination Hardware Address Register in PPPoE) .....	77
PSIDR (Session ID Register in PPPoE).....	78
PMRUR (Maximum Receive Unit Register in PPPoE) .....	79
SHAR (Source Hardware Address Register).....	79
GAR (Gateway Address) .....	80

---

---

SUBR ( Subnet Mask Register) .....	80
SIPR (Source IP address Register) .....	81
NCONFLR ( Network Configuration Lock Register) .....	81
RTR (Retry Time Register) .....	82
RCR (Retry Counter Register) .....	83
UIPR (Unreachable IP address Register).....	84
UPORTR (Unreachable Port Register).....	85
<b>13.6      Socket register (Base : 0x4601_0000 + 0x0004_000 x n)[n=0,...7, where n is socket number] .....</b>	<b>85</b>
Sn_MR (Socket n Mode Register).....	85
Sn_CR (Socket n Command Register) .....	88
Sn_IR (Socket n Interrupt Register) .....	90
Sn_IMR (Socket n Interrupt Mask Register) .....	90
Sn_ICR (Socket n Interrupt Clear Register) .....	91
Sn_SR (Socket n Status Register) .....	92
Sn_PNR (Socket n Protocol Number Register) .....	94
Sn_TOSR (Socket n IP Type of Service Register) .....	95
Sn_TTLR (Socket n TTL Register) .....	95
Sn_FRAGR (Socket n Fragment offset Register) .....	96
Sn_MSSR (Socket n Maximum Segment Register) .....	96
Sn_PORTR (Socket n Source Port Register) .....	97
Sn_DHAR (Socket n Destination Hardware address Register).....	97
Sn_DPORTR (Socket n Destination Port Number Register) .....	98
Sn_DIPR (Socket n Destination IP address Register).....	99
Sn_KATMR (Socket n Keep Alive Timer Register).....	100
Sn_RTR (Socket n Retry Time Register) .....	100
Sn_RCR (Socket n Retry Counter Register) .....	101
Sn_TXBUF_SIZE (Socket n TX Buffer Size Register).....	101
Sn_TX_FSR (Socket n TX Free Size Register) .....	102
Sn_TX_RD (Socket n TX Read Pointer Register) .....	103
Sn_TX_WR (Socket n TX Write Pointer Register).....	104
Sn_RXBUF_SIZE (Socket n RX Buffer Size Register).....	104
Sn_RX_RSR (Socket n RX Received Size Register) .....	105
Sn_RX_RD (Socket n RX Read Pointer Register) .....	106
Sn_RX_WR (Socket n RX Write Pointer Register).....	107
<b>14 Random number generator (RNG) .....</b>	<b>108</b>
14.1     Introduction .....	108
14.2     Features .....	108
14.3     Functional description .....	108

---

---

Operation RNG.....	109
14.4    Registers (Base address : 0x4000_7000).....	110
RNG run register (RNG_RUN).....	110
RNG SEED register (RNG_SEED) .....	110
RNG clock select register (RNG_CLKSEL) .....	110
RNG manual mode select register (RNG_MODE) .....	111
RNG random number value register (RNG_RN) .....	111
RNG polynomial register (RNG_POLY) .....	112
14.5    Register map .....	113
<b>15 Alternate Function Controller (AFC)</b> .....	<b>113</b>
15.1    Introduction .....	113
15.2    Features .....	113
15.3    Functional description .....	113
15.4    Registers (Base address : 0x4100_2000).....	116
PA_00 pad alternate function select register (PA_00_AFR) .....	116
PA_01 pad alternate function select register (PA_01_AFR) .....	116
PA_02 pad alternate function select register (PA_02_AFR) .....	117
PA_03 pad alternate function select register (PA_03_AFR) .....	117
PA_04 pad alternate function select register (PA_04_AFR) .....	118
PA_05 pad alternate function select register (PA_05_AFR) .....	118
PA_06 pad alternate function select register (PA_06_AFR) .....	119
PA_07 pad alternate function select register (PA_07_AFR) .....	119
PA_08 pad alternate function select register (PA_08_AFR) .....	120
PA_09 pad alternate function select register (PA_09_AFR) .....	120
PA_10 pad alternate function select register (PA_10_AFR) .....	121
PA_11 pad alternate function select register (PA_11_AFR) .....	121
PA_12 pad alternate function select register (PA_12_AFR) .....	122
PA_13 pad alternate function select register (PA_13_AFR) .....	122
PA_14 pad alternate function select register (PA_14_AFR) .....	123
PA_15 pad alternate function select register (PA_15_AFR) .....	123
PB_00 pad alternate function select register (PB_00_AFR) .....	124
PB_01 pad alternate function select register (PB_01_AFR) .....	124
PB_02 pad alternate function select register (PB_02_AFR) .....	125
PB_03 pad alternate function select register (PB_03_AFR) .....	125
PB_04 pad alternate function select register (PB_04_AFR) .....	126
PB_05 pad alternate function select register (PB_05_AFR) .....	126
PB_06 pad alternate function select register (PB_06_AFR) .....	127
PB_07 pad alternate function select register (PB_07_AFR) .....	127
PB_08 pad alternate function select register (PB_08_AFR) .....	128

---

---

PB_09 pad alternate function select register (PB_09_AFR) .....	128
PB_10 pad alternate function select register (PB_10_AFR) .....	129
PB_11 pad alternate function select register (PB_11_AFR) .....	129
PB_12 pad alternate function select register (PB_12_AFR) .....	130
PB_13 pad alternate function select register (PB_13_AFR) .....	130
PB_14 pad alternate function select register (PB_14_AFR) .....	131
PB_15 pad alternate function select register (PB_15_AFR) .....	131
PC_00 pad alternate function select register (PC_00_AFR) .....	132
PC_01 pad alternate function select register (PC_01_AFR) .....	132
PC_02 pad alternate function select register (PC_02_AFR) .....	133
PC_03 pad alternate function select register (PC_03_AFR) .....	133
PC_04 pad alternate function select register (PC_04_AFR) .....	134
PC_05 pad alternate function select register (PC_05_AFR) .....	134
PC_06 pad alternate function select register (PC_06_AFR) .....	135
PC_07 pad alternate function select register (PC_07_AFR) .....	135
PC_08 pad alternate function select register (PC_08_AFR) .....	136
PC_09 pad alternate function select register (PC_09_AFR) .....	136
PC_10 pad alternate function select register (PC_10_AFR) .....	137
PC_11 pad alternate function select register (PC_11_AFR) .....	137
PC_12 pad alternate function select register (PC_12_AFR) .....	138
PC_13 pad alternate function select register (PC_13_AFR) .....	138
PC_14 pad alternate function select register (PC_14_AFR) .....	139
PC_15 pad alternate function select register (PC_15_AFR) .....	139
PD_00 pad alternate function select register (PD_00_AFR) .....	140
PD_01 pad alternate function select register (PD_01_AFR) .....	140
PD_02 pad alternate function select register (PD_02_AFR) .....	141
PD_03 pad alternate function select register (PD_03_AFR) .....	141
PD_04 pad alternate function select register (PD_04_AFR) .....	142
15.5 Register map .....	143
<b>16 External Interrupt (EXTI) .....</b>	<b>146</b>
16.1 Introduction .....	146
16.2 Features .....	146
16.3 Functional description .....	146
16.4 Registers (Base address : 0x4100_2000).....	148
External interrupt enable register (Px_y EXTINT).....	148
16.5 Register map .....	148
<b>17 Pad Controller (PADCON) .....</b>	<b>149</b>
17.1 Introduction .....	149
17.2 Features .....	149

---

---

17.3	Functional description .....	149
17.4	Registers (Base address : 0x4100_3000).....	151
	PAD Control register (Px_y PCR)(x=A..D, y=0..15) .....	151
17.5	Register map .....	151
<b>18</b>	<b>General-purpose I/Os(GPIO)</b> .....	<b>152</b>
18.1	Introduction.....	152
18.2	Features .....	152
18.3	Functional description .....	152
	Masked access .....	153
18.4	GPIO Registers(Address Base: 0x4200_0000) .....	155
	GPIO Data Register(GPIOx_DATA) (x=A..D) .....	155
	GPIO Output Latch Register(GPIOx_DATAOUT) (x=A..D).....	155
	GPIO Enable Set Register(GPIOx_OUTENSET) (x=A..D) .....	155
	GPIO Enable Clear Register(GPIOx_OUTENCLR) (x=A..D) .....	156
	GPIO Interrupt Enable Set Register(GPIOx_INTENSET) (x=A..D) .....	156
	GPIO Interrupt Enable Clear Register(GPIOx_INTENCLR) (x=A..D).....	157
	GPIO Interrupt Type Set Register(GPIOx_INTPOLSET) (x=A..D) .....	157
	GPIO Interrupt Type Clear Register(GPIOx_INTPOLCLR) (x=A..D) .....	158
	GPIO Interrupt Polarity Set Register(GPIOx_INTPOLSET) (x=A..D).....	158
	GPIO Interrupt Polarity Clear Register(GPIOx_INTPOLCLR) (x=A..D) .....	159
	GPIO Interrupt Status/Clear Register(GPIO_INTPOLSTATUS/INTCLEAR) (x=A..D).....	160
	GPIO Lower Byte Masked Access Register(GPIOx_LB_MASKED) (x=A..D).....	160
	GPIO Upper Byte Masked Access Register(GPIOx_UB_MASKED) (x=A..D) .....	161
18.5	Register map .....	162
<b>19</b>	<b>Direct memory access controller (DMA)</b> .....	<b>163</b>
19.1	Introduction .....	163
19.2	Features .....	163
19.3	Functional description .....	163
	DMA request mapping .....	164
	DMA arbitration.....	164
	DMA cycle types .....	164
19.4	Registers (Base address : 0x4100_4000).....	167
	DMA status register (DMA_STATUS) .....	167
	DMA configuration register (DMA_CFG) .....	168
	DMA control data base pointer register (DMA_CTRL_BASE_PTR) .....	169
	DMA channel alternate control data base pointer register (DMA_ALT_CTRL_BASE_PTR) .....	169
	DMA channel wait on request status register (DMA_WAITONREQ_STATUS) .....	170
	DMA channel software request register (DMA_CHNL_SW_REQUEST) .....	170

---

---

DMA channel useburst set register (DMA_CHNL_USEBURST_SET) .....	171
DMA channel useburst clear register (DMA_CHNL_USEBURST_CLR) .....	171
DMA channel request mask set register (DMA_CHNL_REQ_MASK_SET) .....	172
DMA channel request mask clear register (DMA_CHNL_REQ_MASK_CLR).....	173
DMA channel enable set register (DMA_CHNL_ENABLE_SET) .....	173
DMA channel enable clear register (DMA_CHNL_ENABLE_CLR) .....	174
DMA channel primary-alternate set register (DMA_CHNL_PRI_ALT_SET) .....	174
DMA channel primary-alternate clear register (DMA_CHNL_PRI_ALT_CLR) .....	175
DMA channel priority set register (DMA_CHNL_PRIORITY_SET) .....	175
DMA channel priority clear register (DMA_CHNL_PRIORITY_CLR) .....	176
DMA bus error clear register (DMA_ERR_CLR) .....	176
19.5 Register map .....	178
<b>20 Analog-to-digital converter (ADC)</b> .....	<b>179</b>
20.1 Introduction .....	179
20.2 Features .....	179
20.3 Functional description .....	180
Operation ADC with non-interrupt .....	180
Operation ADC with interrupt.....	182
20.4 Registers (Base address : 0x4100_0000).....	182
ADC control register (ADC_CTR).....	182
ADC channel select register (ADC_CHSEL).....	183
ADC start register (ADC_START) .....	184
ADC conversion data register (ADC_DATA).....	184
ADC Interrupt register (ADC_INT) .....	184
ADC Interrupt Clear register (ADC_INTCLR) .....	185
20.5 Register map .....	186
<b>21 Pulse-Width Modulation (PWM)</b> .....	<b>187</b>
21.1 Introduction .....	187
21.2 Features .....	187
21.3 Functional description .....	188
Timer/Counter control .....	188
Timer/Counter.....	188
PWM mode .....	192
Interrupt.....	193
Dead zone generation .....	194
Capture event .....	195
How to set the PWM .....	197
21.4 PWM Channel-0 Registers (Base address : 0x4000_5000).....	197
Channel-0 interrupt register(PWMCH0IR) .....	197

---

---

Channel-0 interrupt enable register(PWMCH0IER) .....	198
Channel-0 interrupt clear register(PWMCH0ICR) .....	199
Channel-0 Timer/Counter Register (PWMCH0TCR) .....	199
Channel-0 Prescale Counter Register (PWMCH0PCR) .....	200
Channel-0 Prescale Register (PWMCH0PR).....	200
Channel-0 Match Register (PWMCH0MR) .....	200
Channel-0 Limit Register (PWMCH0LR).....	201
Channel-0 Up-Down Mode Register (PWMCH0UDMR).....	201
Channel-0 Timer/Counter Mode Register (PWMCH0TCMR).....	202
Channel-0 PWM output Enable and External input Enable Register (PWMCH0PEEER)	
202	
Channel-0 Capture Mode Register (PWMCH0CMR).....	203
Channel-0 Capture Register (PWMCH0CR) .....	203
Channel-0 Periodic Mode Register (PWMCH0PDMR) .....	203
Channel-0 Dead Zone Enable Register (PWMCH0DZER).....	204
Channel-0 Dead Zone Counter Register (PWMCH0DZCR) .....	204
21.5     Register map .....	205
21.6     PWM Channel-1 Registers (Base address : 0x4000_5100).....	207
Channel-1 interrupt register(PWMCH1IR) .....	207
Channel-1 interrupt enable register(PWMCH1IER) .....	207
Channel-1 interrupt clear register(PWMCH1ICR) .....	208
Channel-1 Timer/Counter Register (PWMCH1TCR) .....	208
Channel-1 Prescale Counter Register (PWMCH1PCR) .....	209
Channel-1 Prescale Register (PWMCH1PR).....	209
Channel-1 Match Register (PWMCH1MR) .....	210
Channel-1 Limit Register (PWMCH1LR).....	210
Channel-1 Up-Down Mode Register (PWMCH1UDMR).....	211
Channel-1 Timer/Counter Mode Register (PWMCH1TCMR).....	211
Channel-1 PWM output Enable and External input Enable Register (PWMCH1PEEER)	
212	
Channel-1 Capture Mode Register (PWMCH1CMR).....	212
Channel-1 Capture Register (PWMCH1CR) .....	212
Channel-1 Periodic Mode Register (PWMCH1PDMR) .....	213
Channel-1 Dead Zone Enable Register (PWMCH1DZER).....	213
Channel-1 Dead Zone Counter Register (PWMCH1DZCR) .....	214
21.7     Register map .....	215
21.8     PWM Channel-2 Registers (Base address : 0x4000_5200).....	216
Channel-2 interrupt register(PWMCH2IR) .....	216
Channel-2 interrupt enable register(PWMCH2IER) .....	216

---

Channel-2 interrupt clear register(PWMCH2ICR) .....	217
Channel-2 Timer/Counter Register (PWMCH2TCR) .....	217
Channel-2 Prescale Counter Register (PWMCH2PCR) .....	218
Channel-2 Prescale Register (PWMCH2PR).....	218
Channel-2 Match Register (PWMCH2MR) .....	219
Channel-2 Limit Register (PWMCH2LR).....	219
Channel-2 Up-Down Mode Register (PWMCH2UDMR).....	219
Channel-2 Timer/Counter Mode Register (PWMCH2TCMR).....	220
Channel-2 PWM output Enable and External input Enable Register (PWMCH2PEEER)	
220	
Channel-2 Capture Mode Register (PWMCH2CMR).....	221
Channel-2 Capture Register (PWMCH2CR) .....	221
Channel-2 Periodic Mode Register (PWMCH2PDMR) .....	222
Channel-2 Dead Zone Enable Register (PWMCH2DZER).....	222
Channel-2 Dead Zone Counter Register (PWMCH2DZCR).....	223
21.9     Register map .....	224
21.10    PWM Channel-3 Registers (Base address : 0x4000_5300).....	225
Channel-3 interrupt register(PWMCH3IR) .....	225
Channel-3 interrupt enable register(PWMCH3IER) .....	225
Channel-3 interrupt clear register(PWMCH3ICR) .....	226
Channel-3 Timer/Counter Register (PWMCH3TCR) .....	226
Channel-3 Prescale Counter Register (PWMCH3PCR) .....	227
Channel-3 Prescale Register (PWMCH3PR).....	227
Channel-3 Match Register (PWMCH3MR) .....	228
Channel-3 Limit Register (PWMCH3LR).....	228
Channel-3 Up-Down Mode Register (PWMCH3UDMR).....	229
Channel-3 Timer/Counter Mode Register (PWMCH3TCMR).....	229
Channel-3 PWM output Enable and External input Enable Register (PWMCH3PEEER)	
230	
Channel-3 Capture Mode Register (PWMCH3CMR).....	230
Channel-3 Capture Register (PWMCH3CR) .....	230
Channel-3 Periodic Mode Register (PWMCH3PDMR) .....	231
Channel-3 Dead Zone Enable Register (PWMCH3DZER).....	231
Channel-3 Dead Zone Counter Register (PWMCH3DZCR) .....	232
21.11    Register map .....	233
21.12    PWM Channel-4 Registers (Base address : 0x4000_5400).....	234
Channel-4 interrupt register(PWMCH4IR) .....	234
Channel-4 interrupt enable register(PWMCH4IER) .....	234
Channel-4 interrupt clear register(PWMCH4ICR) .....	235

Channel-4 Timer/Counter Register (PWMCH4TCR) .....	235
Channel-4 Prescale Counter Register (PWMCH4PCR) .....	236
Channel-4 Prescale Register (PWMCH4PR) .....	236
Channel-4 Match Register (PWMCH4MR) .....	237
Channel-4 Limit Register (PWMCH4LR) .....	237
Channel-4 Up-Down Mode Register (PWMCH4UDMR).....	237
Channel-4 Timer/Counter Mode Register (PWMCH4TCMR).....	238
Channel-4 PWM output Enable and External input Enable Register (PWMCH4PEEER)	
238	
Channel-4 Capture Mode Register (PWMCH4CMR).....	239
Channel-4 Capture Register (PWMCH4CR) .....	239
Channel-4 Periodic Mode Register (PWMCH4PDMR) .....	240
Channel-4 Dead Zone Enable Register (PWMCH4DZER).....	240
Channel-4 Dead Zone Counter Register (PWMCH4DZCR) .....	241
21.13    Register map .....	242
21.14    PWM Channel-5 Registers (Base address : 0x4000_5500).....	243
Channel-5 interrupt register(PWMCH5IR) .....	243
Channel-5 interrupt enable register(PWMCH5IER) .....	243
Channel-5 interrupt clear register(PWMCH5ICR) .....	244
Channel-5 Timer/Counter Register (PWMCH5TCR) .....	244
Channel-5 Prescale Counter Register (PWMCH5PCR) .....	245
Channel-5 Prescale Register (PWMCH5PR) .....	245
Channel-5 Match Register (PWMCH5MR) .....	246
Channel-5 Limit Register (PWMCH5LR) .....	246
Channel-5 Up-Down Mode Register (PWMCH5UDMR).....	246
Channel-5 Timer/Counter Mode Register (PWMCH5TCMR).....	247
Channel-5 PWM output Enable and External input Enable Register (PWMCH5PEEER)	
247	
Channel-5 Capture Mode Register (PWMCH5CMR).....	248
Channel-5 Capture Register (PWMCH5CR) .....	248
Channel-5 Periodic Mode Register (PWMCH5PDMR) .....	249
Channel-5 Dead Zone Enable Register (PWMCH5DZER).....	249
Channel-5 Dead Zone Counter Register (PWMCH5DZCR) .....	250
21.15    Register map .....	251
21.16    PWM Channel-6 Registers (Base address : 0x4000_5600).....	252
Channel-6 interrupt register(PWMCH6IR) .....	252
Channel-6 interrupt enable register(PWMCH6IER) .....	252
Channel-6 interrupt clear register(PWMCH6ICR) .....	253
Channel-6 Timer/Counter Register (PWMCH6TCR) .....	253

Channel-6 Prescale Counter Register (PWMCH6PCR) .....	254
Channel-6 Prescale Register (PWMCH6PR).....	254
Channel-6 Match Register (PWMCH6MR) .....	255
Channel-6 Limit Register (PWMCH6LR).....	255
Channel-6 Up-Down Mode Register (PWMCH6UDMR).....	256
Channel-6 Timer/Counter Mode Register (PWMCH6TCMR).....	256
Channel-6 PWM output Enable and External input Enable Register (PWMCH6PEEER)	
257	
Channel-6 Capture Mode Register (PWMCH6CMR).....	257
Channel-6 Capture Register (PWMCH6CR) .....	257
Channel-6 Periodic Mode Register (PWMCH6PDMR) .....	258
Channel-6 Dead Zone Enable Register (PWMCH6DZER).....	258
Channel-6 Dead Zone Counter Register (PWMCH6DZCR) .....	259
21.17     Register map .....	260
21.18     PWM Channel-7 Registers (Base address : 0x4000_5700).....	261
Channel-7 interrupt register(PWMCH7IR) .....	261
Channel-7 interrupt enable register(PWMCH7IER) .....	261
Channel-7 interrupt clear register(PWMCH7ICR) .....	262
Channel-7 Timer/Counter Register (PWMCH7TCR) .....	262
Channel-7 Prescale Counter Register (PWMCH7PCR) .....	263
Channel-7 Prescale Register (PWMCH7PR).....	263
Channel-7 Match Register (PWMCH7MR) .....	264
Channel-7 Limit Register (PWMCH7LR).....	264
Channel-7 Up-Down Mode Register (PWMCH7UDMR).....	265
Channel-7 Timer/Counter Mode Register (PWMCH7TCMR).....	265
Channel-7 PWM output Enable and External input Enable Register (PWMCH7PEEER)	
266	
Channel-7 Capture Mode Register (PWMCH7CMR).....	266
Channel-7 Capture Register (PWMCH7CR) .....	266
Channel-7 Periodic Mode Register (PWMCH7PDMR) .....	267
Channel-7 Dead Zone Enable Register (PWMCH7DZER).....	267
Channel-7 Dead Zone Counter Register (PWMCH7DZCR) .....	268
21.19     Register map .....	269
21.20     PWM Common Registers (Base address : 0x4000_5800) .....	270
Interrupt Enable Register (IER) .....	270
Start/Stop Register (SSR) .....	271
Pause Register (PSR) .....	272
21.21     Register map .....	273
<b>22 Dual timers .....</b>	<b>274</b>

---

22.1	Introduction .....	274
22.2	Features .....	274
22.3	Functional description .....	275
	Clock and clock enable .....	275
	Timer size.....	275
	Prescaler.....	275
	Repetition mode.....	275
	Interrupt.....	276
	Operation.....	276
	How to set the dual timers .....	277
22.4	Dual timer0_0 Registers (Base address : 0x4000_1000) .....	278
	Timer0_0 Load Register(DUALTIMER0_0TimerLoad).....	278
	Timer0_0 Value Register(DUALTIMER0_0TimerValue) .....	278
	Timer0_0 Control Register(DUALTIMER0_0TimerControl) .....	278
	Timer0_0 Interrupt Clear Register (DUALTIMER0_0TimerIntClr) .....	279
	Timer0_0 Raw Interrupt Status Register (DUALTIMER0_0TimerRIS) .....	279
	Timer0_0 Masked Interrupt Status Register (DUALTIMER0_0TimerMIS) .....	280
	Timer0_0 Background Load Register (DUALTIMER0_0TimerBGLoad) .....	280
22.5	Register map .....	282
22.6	Dual timer0_1 Registers (Base address : 0x4000_1020) .....	283
	Timer0_1 Load Register(DUALTIMER0_1TimerLoad).....	283
	Timer0_1 Value Register(DUALTIMER0_1TimerValue) .....	283
	Timer0_1 Control Register(DUALTIMER0_1TimerControl) .....	283
	Timer0_1 Interrupt Clear Register (DUALTIMER0_1TimerIntClr) .....	284
	Timer0_1 Raw Interrupt Status Register (DUALTIMER0_1TimerRIS) .....	284
	Timer0_1 Masked Interrupt Status Register (DUALTIMER0_1TimerMIS) .....	285
	Timer0_1 Background Load Register (DUALTIMER0_1TimerBGLoad) .....	285
22.7	Register map .....	287
22.8	Dual Timer 0 Clock Enable Register (Base address : 0x4000_1080) .....	288
	Timer0_0 Clock Enable Register (TIMCLKEN0_0) .....	288
	Timer0_1 Clock Enable Register (TIMCLKEN0_1) .....	288
22.9	Register map .....	289
22.10	Dual timer1_0 Registers (Base address : 0x4000_2000) .....	290
	Timer1_0 Load Register(DUALTIMER1_0TimerLoad).....	290
	Timer1_0 Value Register(DUALTIMER1_0TimerValue) .....	290
	Timer1_0 Control Register(DUALTIMER1_0TimerControl) .....	290
	Timer1_0 Interrupt Clear Register (DUALTIMER1_0TimerIntClr) .....	291
	Timer1_0 Raw Interrupt Status Register (DUALTIMER1_0TimerRIS) .....	291
	Timer1_0 Masked Interrupt Status Register (DUALTIMER1_0TimerMIS) .....	292

---

---

Timer1_0 Background Load Register (DUALTIMER1_0TimerBGLoad) .....	292
22.11    Register map .....	294
22.12    Dual timer1_1 Registers (Base address : 0x4000_2020) .....	295
Timer1_1 Load Register(DUALTIMER1_1TimerLoad).....	295
Timer1_1 Value Register(DUALTIMER1_1TimerValue) .....	295
Timer1_1 Control Register(DUALTIMER1_1TimerControl) .....	295
Timer1_1 Interrupt Clear Register (DUALTIMER1_1TimerIntClr) .....	296
Timer1_1 Raw Interrupt Status Register (DUALTIMER1_1TimerRIS) .....	296
Timer1_1 Masked Interrupt Status Register (DUALTIMER1_1TimerMIS) .....	297
Timer1_1 Background Load Register (DUALTIMER1_1TimerBGLoad) .....	297
22.13    Register map .....	299
22.14    Dual Timer 1 Clock Enable Register (Base address : 0x4000_2080) .....	300
Timer1_0 Clock Enable Register (TIMCLKEN1_0) .....	300
Timer1_1 Clock Enable Register (TIMCLKEN1_1) .....	300
22.15    Register map .....	301
<b>23 Watchdog timer .....</b>	<b>301</b>
23.1    Introduction .....	301
23.2    Features .....	301
23.3    Functional description .....	301
Clock.....	301
Interrupt and reset request .....	302
23.4    Watchdog timer Registers (Base address : 0x4000_0000) .....	302
Watchdog timer Load Register(WDTLoad) .....	302
Watchdog timer Value Register(WDTValue) .....	303
Watchdog timer Control Register(WDTControl).....	303
Watchdog timer Interrupt Clear Register (WDTIntClr).....	303
Watchdog timer Raw Interrupt Status Register (WDTRIS).....	304
Watchdog timer Masked Interrupt Status Register (WDTMIS) .....	304
Watchdog timer Lock Register(WDTLock).....	305
23.5    Register map .....	306
<b>24 Real-time Clock(RTC) .....</b>	<b>306</b>
24.1    Introduction .....	306
24.2    Features .....	306
24.3    Functional description .....	306
RTC clock.....	307
RTC interrupt .....	307
RTC Counter and Calendar .....	307
RTC Setting flow .....	308
24.4    RTC Registers (Base Address : 0x4000_E000) .....	309

---

---

RTC control register (RTCCON) .....	309
RTC Interrupt Mask register (RTCINTE).....	310
RTC Interrupt Pending register (RTCINTP).....	312
RTC Alarm Mask register (RTCAMR) .....	313
RTC BCD Second register (BCDSEC) .....	314
RTC BCD Minute register (BCDMIN).....	314
RTC BCD Hour register (BCDHOUR) .....	314
RTC BCD Day register (BCDDAY) .....	315
RTC BCD Date register (BCDDATE) .....	315
RTC BCD Month register (BCDMON) .....	315
RTC BCD Year register (BCDYEAR).....	316
RTC Predetermining Second register (PRESEC) .....	316
RTC Predetermining Minute register (PREMIN).....	316
RTC Predetermining Hour register (PREHOUR).....	317
RTC Predetermining Day register (PREDAY) .....	317
RTC Predetermining Date register (PREDATE).....	317
RTC Predetermining Month register (PREMON) .....	318
RTC Predetermining Year register (PREYEAR) .....	318
RTC Consolidated Time0 register (RTCTIME0).....	318
RTC Consolidated Time1 register (RTCTIME1).....	319
24.5     Register map .....	320
<b>25 UART(Universal Asynchronous Receive Transmit).....</b>	<b>321</b>
25.1     Introduction .....	321
25.2     Features .....	321
25.3     Functional description .....	322
Baud rate calculation .....	324
Data transmission .....	325
Data receive .....	325
Hardware flow control .....	326
25.4     UART0 Registers(Base address: 0x4000_C000) .....	327
UART0DR (UART0 Data Register) .....	327
UART0RSR/ECR (UART0 Receive Status Register/Error Clear Register).....	328
UART0FR (UART0 Flag Register) .....	329
UART0ILPR (UART0 IrDA Low-Power Counter Register) .....	330
UART0IBRD (UART0 Integer Baud Rate Register) .....	331
UART0FBRD (UART0 Fractional Baud Rate Register) .....	331
UART0LCR_H (UART0 Line Control Register) .....	332
UART0CR (UART0 Control register) .....	333
UART0IFLS (UART0 Interrupt FIFO Level Select Register) .....	335

---

---

UART0IMSC (UART0 Interrupt Mask Set/Clear Register) .....	335
UART0RIS (UART0 Raw Interrupt Status Register) .....	337
UART0MIS (UART0 Masked Interrupt Status Register).....	338
UART0ICR (UART0 Interrupt Clear Register) .....	339
25.5     Register map .....	340
25.6     UART1 Registers(Base address: 0x4000_D000) .....	341
UART1DR (UART1 Data Register) .....	341
UART1RSR/ECR (UART1 Receive Status Register/Error Clear Register).....	341
UART1FR (UART1 Flag Register) .....	342
UART1ILPR (UART1 IrDA Low-Power Counter Register) .....	343
UART1IBRD (UART1 Integer Baud Rate Register) .....	344
UART1FBRD (UART1 Fractional Baud Rate Register) .....	344
UART1LCR_H (UART1 Line Control Register) .....	345
UART1CR (UART1 Control register) .....	346
UART1IFLS (UART1 Interrupt FIFO Level Select Register).....	348
UART1IMSC (UART1 Interrupt Mask Set/Clear Register) .....	348
UART1RIS (UART1 Raw Interrupt Status Register) .....	350
UART1MIS (UART1 Masked Interrupt Status Register).....	351
UART1ICR (UART1 Interrupt Clear Register) .....	352
25.7     Register map .....	353
<b>26 Universal Asynchronous Receive Transmit(UART2) .....</b>	<b>354</b>
26.1     Introduction .....	354
26.2     Feature.....	354
26.3     Functional description .....	354
Baud rate calculation .....	354
26.4     UART2 Registers(Base address: 0x4000_6000) .....	356
UART2DR (UART2 Data Register) .....	356
UART2SR (UART2 Status Register) .....	356
UART2CR (UART2 Control Register) .....	357
UART2ISR/ICR (UART2 Interrupt Status/Interrupt Clear Register) .....	357
UART2BDR (UART2 Baud Rate Divider Register) .....	358
26.5     Register map .....	359
<b>27 Synchronous Serial Port (SSP) .....</b>	<b>360</b>
27.1     Introduction .....	360
27.2     Features .....	360
27.3     Functional description .....	361
Clock prescaler .....	361
Transmit FIFO .....	361
Receive FIFO .....	362

---

---

Interrupt generation logic .....	362
DMA interface .....	362
Interface reset .....	364
Configuring the SSP .....	364
Enable PrimeCell SSP operation .....	365
Clock ratios .....	365
Programming the SSPCR0 Control Register .....	366
Programming the SSPCR1 Control Register .....	366
Frame format .....	367
Texas Instruments synchronous serial frame format .....	368
Motorola SPI frame format .....	369
National Semiconductor Microwire frame format .....	375
Master and Slave configurations .....	377
SSP Flow chart .....	378
27.4     SSP0 Registers (Base Address : 0x4000_A000) .....	379
SSP0 Control register 0 (SSP0CR0) .....	379
SSP0 Control register 1 (SSP0CR1) .....	381
SSP0 Data register (SSP0DR) .....	381
SSP0 Status register (SSP0SR) .....	382
SSP0 Clock prescale register (SSP0CPSR) .....	383
SSP0 Interrupt mask set or clear register (SSP0IMSC) .....	383
SSP0 Raw interrupt status register (SSP0RIS) .....	384
SSP0 Masked interrupt status register, (SSP0MIS) .....	384
SSP0 Interrupt clear register (SSP0ICR) .....	385
SSP0 DMA control register, (SSP0DMACR) .....	385
27.5     Register map .....	387
27.6     SSP1 Registers (Base Address : 0x4000_B000) .....	388
SSP1 Control register 0 (SSP1CR0) .....	388
SSP1 Control register 1 (SSP1CR1) .....	389
SSP1 Data register (SSP1DR) .....	391
SSP1 Status register (SSP1SR) .....	391
SSP1 Clock prescale register (SSP1CPSR) .....	392
SSP1 Interrupt mask set or clear register (SSP1IMSC) .....	392
SSP1 Raw interrupt status register (SSP1RIS) .....	393
SSP1 Masked interrupt status register, (SSP1MIS) .....	393
SSP1 Interrupt clear register (SSP1ICR) .....	394
SSP1 DMA control register, (SSP1DMACR) .....	394
27.7     Register map .....	396
Document History Information .....	397

---

## 2 List of table

Table 1 W7500x interrupt vector table .....	29
Table 2 W7500x sleep mode summary .....	32
Table 3 operation of mode selection .....	36
Table 4 description of Flash memory.....	37
Table 5 CRG register map and reset values .....	66
Table 6. Offset Address for Common Register .....	69
Table 7. Offset Address in Socket n Register Block .....	70
Table 8 RNG register map and reset values .....	113
Table 9 functional description table .....	113
Table 10 AFC register map and reset values .....	143
Table 11 EXTINT register map and reset values .....	148
Table 12 PAD controller register map and reset values .....	151
Table 13 GPIO register map and reset values.....	162
Table 14 Summary of the DMA requests for each channel.....	164
Table 15 DMA register map and reset values.....	178
Table 16 ADC register map and reset values .....	186
Table 17 PWM channel 0 register map and reset values .....	206
Table 18 PWM channel 1 register map and reset values .....	215
Table 19 PWM channel 2 register map and reset values .....	224
Table 20 PWM channel 3 register map and reset values .....	233
Table 21 PWM channel 4 register map and reset values .....	242
Table 22 PWM channel 5 register map and reset values .....	251
Table 23 PWM channel 6 register map and reset values .....	260
Table 24 PWM channel 7 register map and reset values .....	269
Table 25 PWM common register map and reset values.....	273
Table 26 Dual timer 0_0 register map and reset values .....	282
Table 27 Dual timer 0_1 register map and reset values .....	287
Table 28 Dual timer 0 clock enable register map and reset values .....	289
Table 29 Dual timer 1_0 register map and reset values .....	294
Table 30 Dual timer 1_1 register map and reset values .....	299
Table 31 Dual timer 1 clock enable register map and reset values .....	301
Table 32 Watchdog Timer register map and reset values.....	306
Table 33 RTC register map and reset values .....	320
Table 34 UART0 register map and reset values .....	340
Table 35 UART1 register map and reset values.....	353
Table 36 UART2 register map and reset values .....	359
Table 37 DMA trigger points for the transmit and receive FIFOs. ....	363

---

Table 38 SSP0 register map and reset values .....	387
Table 39 SSP1 register map and reset values .....	396

### 3 List of figures

Figure 1. W7500x System Architecture .....	26
Figure 2 W7500x memory map .....	28
Figure 3. operation of boot code .....	36
Figure 4 CRG block diagram .....	41
Figure 5 Typical application with an 8 MHz crystal .....	41
Figure 6 TOE block diagram .....	67
Figure 7. Register & Memory Organization .....	68
Figure 8. Random Number Generator block diagram.....	108
Figure 9. Flow chart of RNG operation .....	109
Figure 10. External Interrupt diagram .....	147
Figure 11. function schematic of digital I/O pad .....	149
Figure 12. function schematic of digital/analog mux IO pad .....	149
Figure 13. GPIO block diagram .....	152
Figure 17. GPIO Flow chart .....	153
Figure 18. MASK LOWBYTE access .....	154
Figure 19 MASK HIGHBYTE access.....	154
Figure 17. DMA Block diagram .....	163
Figure 18. DMA ping pong cycle .....	167
Figure 19. ADC block diagram .....	180
Figure 20. The ADC operation flowchart with non-interrupt.....	181
Figure 21. The ADC operation flowchart with interrupt .....	182
Figure 22 PWM block diagram .....	188
Figure 23 Periodic mode .....	189
Figure 24 one-shot mode .....	189
Figure 25 Up-count mode .....	189
Figure 26 Down-count mode .....	189
Figure 27 Counter mode with rising edge .....	190
Figure 28 Counter mode with falling edge .....	190
Figure 29 Counter mode with rising and falling edge .....	191
Figure 30 Timer/Counter timing diagram with match interrupt .....	191
Figure 31 Timer/Counter timing diagram with overflow interrupt .....	192
Figure 32 The PWM output up to match register.....	193
Figure 33 The PWM output up to limit register.....	193
Figure 34 PWM waveform with dead zone time .....	194
Figure 35 PWM waveform with dead zone counter .....	195
Figure 36 Capture event with no interrupt clear .....	195
Figure 37 Capture event with interrupt clear .....	196

---

Figure 38 The PWM setting flow .....	197
Figure 39 Block diagram of Dualtimer .....	274
Figure 40 The Dual timer setting flow .....	277
Figure 41 Watchdog timer operation flow diagram .....	302
Figure 42. RTC block diagram .....	307
Figure 43. RTC setting flow for Counter Function .....	308
Figure 44. RTC setting flow for Alarm function .....	309
Figure 45. UART0,1 Block diagram.....	322
Figure 46. UART character frame .....	323
Figure 47. UART divider flow chart .....	324
Figure 48. UART Initial setting flow chart .....	324
Figure 49. Transmit and Receive data flow chart .....	325
Figure 50. Hardware flow control description.....	326
Figure 51. CTS Functional Timing .....	326
Figure 52. Algorithm for setting CTS/RTS flowchart .....	327
Figure 53. UART2 divider flow chart .....	354
Figure 54. UART2 Initial setting flow chart .....	355
Figure 55. SSP block diagram .....	361
Figure 56. DMA transfer waveforms .....	364
Figure 57. Texas Instruments synchronous serial, single transfer .....	368
Figure 58. Texas Instruments synchronous serial, continuous transfers .....	369
Figure 59 Motorola SPI, single transfer, SPO=0 and SPH=0 .....	370
Figure 60 Motorola SPI, continuous transfers, SPO=0 and SPH=0 .....	370
Figure 61 Motorola SPI, single & continuous transfers, SPO=0 and SPH=1 .....	371
Figure 62 Motorola SPI, single transfer, with SPO=1 and SPH=0 .....	372
Figure 63. Motorola SPI, continuous transfers, SPO=1 and SPH=0.....	373
Figure 64. Motorola SPI, single & continuous transfers, SPO=1 and SPH=1.....	374
Figure 65. National Semiconductor Microwire, single transfer .....	375
Figure 66. National Semiconductor Microwire, continuous transfers.....	377
Figure 67. PrimeCell SSP master coupled to an SPI slave.....	377
Figure 68. SPI master coupled to a PrimeCell SSP slave .....	378
Figure 69. how to setting TI or Microwire mode flow chart.....	378
Figure 70. how to setting SPI mode flow chart.....	379

---

## 4 Documentation conventions

### 4.1 Glossary

ARP	Address Resolution Protocol
AHB	Advanced High-performance Bus
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
AFC	Alternate Function Controller
ADC	Analog-to-Digital Converter
BOD	BrownOut Detection
CPU	Central Processing Unit
CRG	Clock Reset generator
DMA	Direct Memory Access
EOP	End Of Packet
EXTINT	External Interrupt
GPIO	General Purpose Input/Output
IrDA	Infrared Data Association
I/O	Input/Output
ICMP	Internet Control Message Protocol
IGMP	Internet Group Management Protocol
IPv4	Internet Protocol version 4
IRQ	interrupt request
NMI	NonMaskable Interrupt
PADCON	Pad Controller
PLL	Phase-Locked Loop
PHY	Physical Layer
PPPoE	Point-to-Point Protocol over Ethernet
POR	Power Of Reset
PWM	Pulse Width Modulator
RAM	Random Access Memory
RNG	Random number generator
SR	Status Register
SSP	Synchronous Serial Port

---

SYSCFG	System configuration controller
TOE	TCP/IP Core Offload Engine
TTL	Transistor-Transistor Logic
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
UDP	User Datagram Protocol
WOL	Wake On Lan
WDT	Watchdog Timer

---

## 4.2 Register Bit Conventions

Each register is shown with a key indicating the accessibility of the each individual bit, and the initial condition:

Key	Bit Accessibility
rw	Read/Write
r	Read Only
r0	Read as 0
r1	Read as 1
W	Write Only

## 5 System and memory overview

### 5.1 System architecture

Main system consists of :

- Three masters :
  - Cortex-M0 core
  - TCP/IP Offload Engine
  - uDMAC (PL230, 6channel)
- Ten slaves :
  - Internal BOOT ROM
  - Internal SRAM
  - Internal Flash memory
  - Two AHB2APB bridge which connects all APB peripherals
  - Four AHB dedicated to 16bit GPIOs
  - TCPIP Hardware core
- Ethernet :
  - IP101G

System architecture and AHB-Lite bus architecture shown in Figure 1.

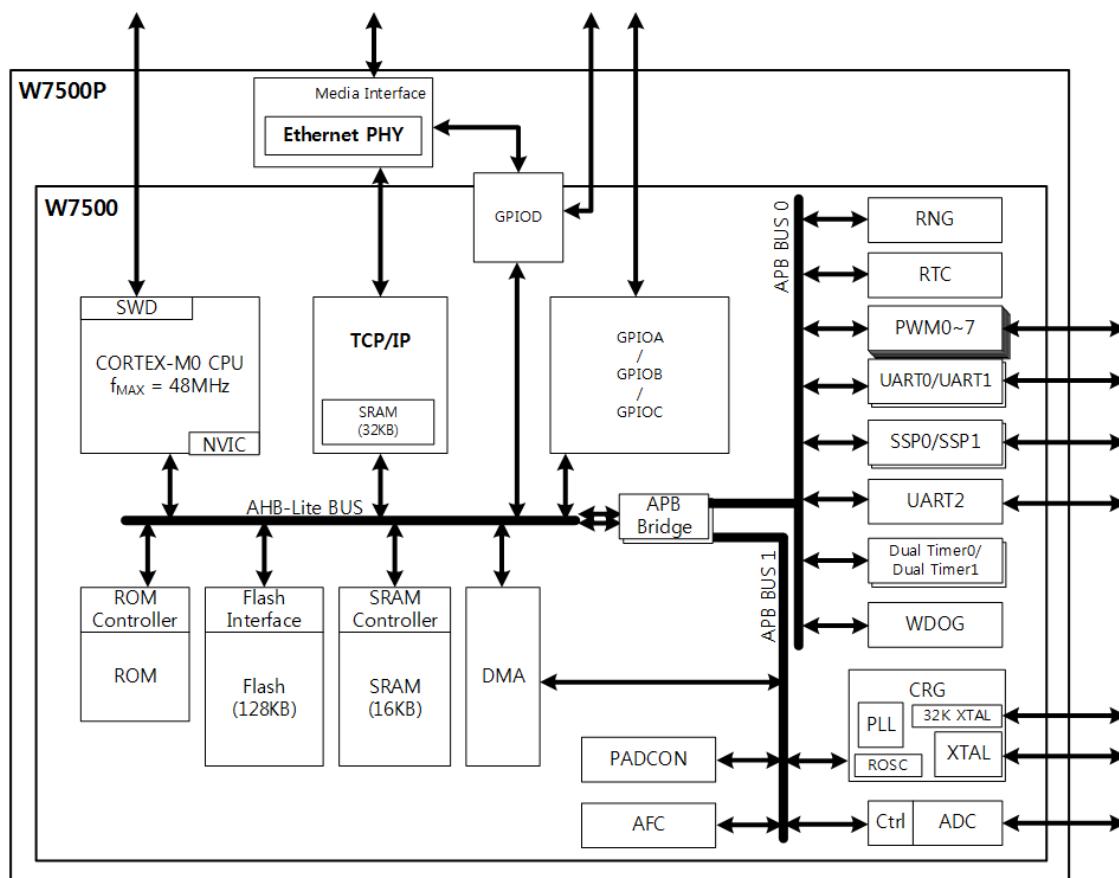


Figure 1. W7500x System Architecture

#### AHB-Lite BUS

- This bus connects the Three masters (Cortex-M0 and uDMAC and TCP/IP Offload Engine) and ten AHB slaves.

#### Two APB BUSs

- These buses connect Seventeen APB peripherals (Watchdog, two Dual timers, PWM, two UARTs, simple UART, two I2Cs, two SSPs, Random Number Generator, Real Time Clock, 12bits Analog Digital Converter, Clock Controller, IO Configuration, PAD MUX controller)

## 5.2 Memory organization

### Introduction

Program memory, data memory, registers and I/O ports are organized within the same linear 4-Gbyte address space.

The bytes are coded in memory in Little Endian format. The lowest numbered byte in a word is considered the word's least significant byte and the highest numbered byte the most significant.

## Memory map

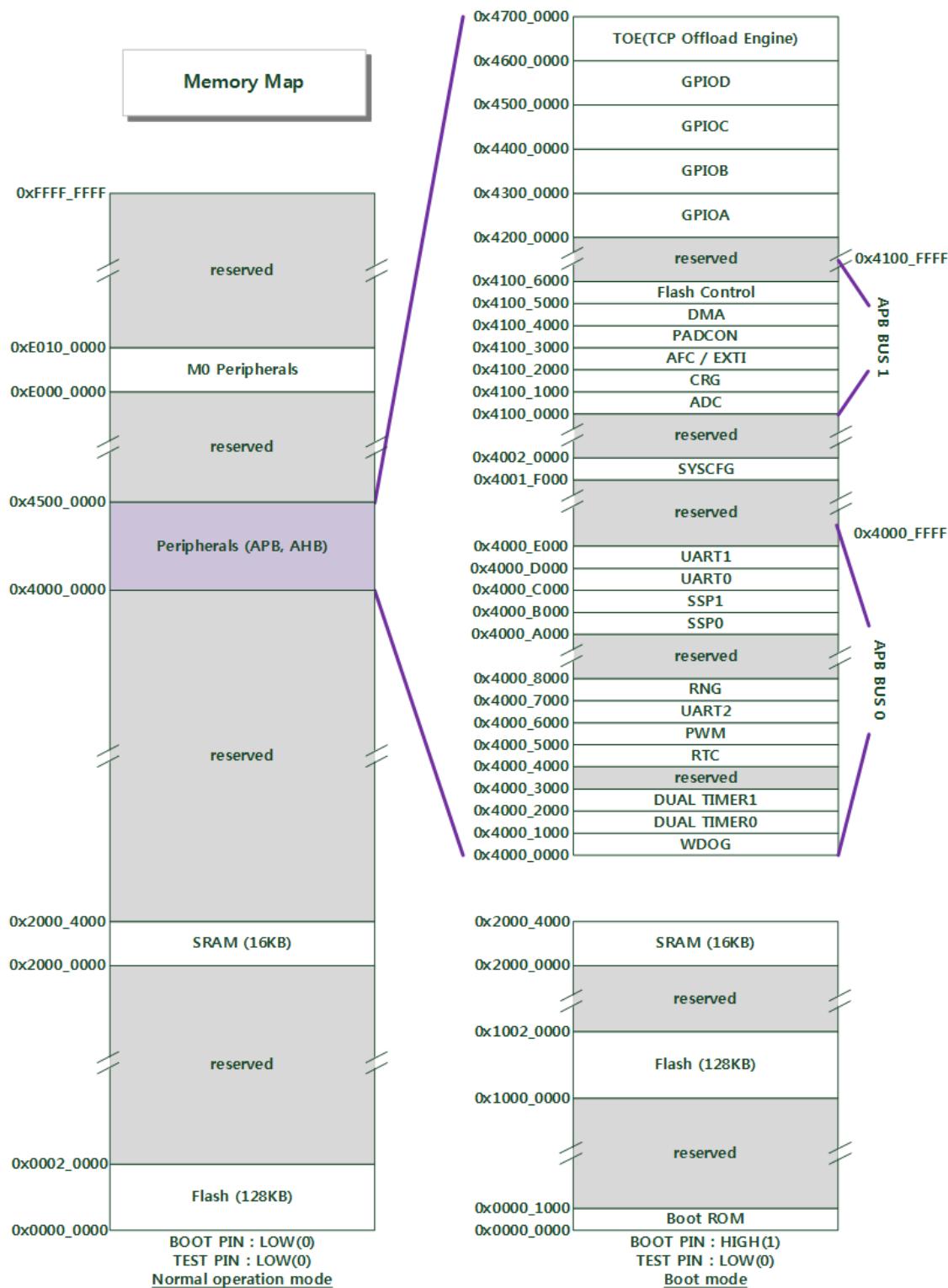


Figure 2 W7500x memory map

## 6 System configuration controller (SYSCFG)

Main purposes of the system configuration controller are the following

- Control of the memory remap feature
- The ability to enable an automatic reset if the system locks up
- Information about the cause of the last reset

## 7 Interrupt and events

### 7.1 Nested vectored interrupt controller (NVIC)

#### NVIC main features

- 32 maskable interrupt channels (not including the sixteen Cortex®-M0 interrupt lines)
- 4 programmable priority levels (2 bits of interrupt priority are used)
- Low-latency exception and interrupt handling
- Power management control
- Implementation of System Control Registers

The NVIC and the processor core interface are closely coupled, which enables low latency interrupt processing and efficient processing of late arriving interrupts. All interrupts including the core exceptions are managed by the NVIC.

#### SysTick calibration value register

The SysTick calibration value is set to 6000, which gives a reference time base of 1 ms with the SysTick clock set to 6 MHz (max  $f_{HCLK}/8$ ).

#### Interrupt and exception vectors

Table 1 describes the W7500x interrupt vector table.

Table 1 W7500x interrupt vector table

Priority	Type of priority	Exception type	Device	Description	Address
-3	fixed	Reset	C0M0	Reset	0x0000_0004
-2	fixed	NMI	Watchdog	Watchdog interrupt	0x0000_0008
-1	Fixed	HardFault	C0M0	All class of fault	0x0000_000C
3	Settable	SVCall	C0M0	System service call via SWI instruction	0x0000_002C
5	settable	PendSV	C0M0	Pendable request for system service	0x0000_0038
6	settable	SysTick	C0M0	System tick timer	0x0000_003C

7	settable	IRQ[0]	SSP0	SSP0 global interrupt	0x0000_0040
8	Settable	IRQ[1]	SSP1	SSP1 global interrupt	0x0000_0044
9	Settable	IRQ[2]	UART0	UART0 global interrupt	0x0000_0048
10	Settable	IRQ[3]	UART1	UART1 global interrupt	0x0000_004C
11	Settable	IRQ[4]	UART2	UART2 global interrupt	0x0000_0050
12	Settable	IRQ[5]	reserved		0x0000_0054
13	Settable	IRQ[6]	reserved		0x0000_0058
14	Settable	IRQ[7]	GPIO0	GPIOA global interrupt	0x0000_005C
15	Settable	IRQ[8]	GPIO1	GPIOB global interrupt	0x0000_0060
16	Settable	IRQ[9]	GPIO2	GPIOC global interrupt	0x0000_0064
17	Settable	IRQ[10]	GPIO3	GPIOD global interrupt	0x0000_0068
18	Settable	IRQ[11]	DMA	DMA interrupt	0x0000_006C
19	Settable	IRQ[12]	Dualtimer0	Dualtimer0 global interrupt	0x0000_0070
20	Settable	IRQ[13]	Dualtimer1	Dualtimer1 global interrupt	0x0000_0074
21	Settable	IRQ[14]	PWM0	PWM0 global interrupt	0x0000_0078
22	Settable	IRQ[15]	PWM1	PWM1 global interrupt	0x0000_007C
23	Settable	IRQ[16]	PWM2	PWM2 global interrupt	0x0000_0080
24	Settable	IRQ[17]	PWM3	PWM3 global interrupt	0x0000_0084
25	Settable	IRQ[18]	PWM4	PWM4 global interrupt	0x0000_0088
26	Settable	IRQ[19]	PWM5	PWM5 global interrupt	0x0000_008C
27	Settable	IRQ[20]	PWM6	PWM6 global interrupt	0x0000_0090
28	Settable	IRQ[21]	PWM7	PWM7 global interrupt	0x0000_0094
29	Settable	IRQ[22]	RTC	RTC interrupt	0x0000_0098
30	Settable	IRQ[23]	ADC	ADC acquisition end interrupt	0x0000_009C
31	Settable	IRQ[24]	TCPIP	TCPIP global interrupt	0x0000_00A0
32	Settable	IRQ[25]	EXT_INT	External pin interrupt	0x0000_00A4
33	Settable	IRQ[26]	reserved		0x0000_00A8
34	Settable	IRQ[27]	reserved		0x0000_00AC
35	Settable	IRQ[28]	reserved		0x0000_00B0
36	Settable	IRQ[29]	reserved		0x0000_00B4
37	Settable	IRQ[30]	reserved		0x0000_00B8
38	Settable	IRQ[31]	reserved		0x0000_00BC

## 7.2 Event

W7500x is able to handle internal events in order to wake up the core(WFE). The wakeup event can be generated by

- When after DMA process finished (DMA\_DONE)

---

## 8 Power supply

### 8.1 Introduction

W7500x embeds a voltage regulator in order to supply the internal 1.5V digital power domain.

- Require a 2.7V ~ 3.6V operating supply voltage (VDD)
- ADC ref voltage is same as VDD

### 8.2 Voltage regulator

The voltage regulator is always enabled after Reset and works in only one mode.

- In Run mode, the regulator supplies full power to the 1.5V domain.
- There is no power down or sleep mode

### 8.3 Low-power modes

W7500x is in RUN mode after a system or power reset. There are two low power modes to save power when the CPU does not need to be kept running. These modes are useful for instances like when the CPU is waiting for an external interrupt. Please note that there is no power-off mode for W7500x.

The device features two low-power modes:

- Sleep mode
- Deep Sleep mode

Additionally, the power consumption can be reducing by following method:

- User can slow down the system clocks
- User can gate the clocks to the peripherals when they are unused.

## Sleep mode

W7500x has two kinds of sleep modes. One is Sleep mode and the other is Deep sleep mode. Two of them are almost the same except the clock gated peripherals kinds. Table 2 shows the Sleep mode summary.

Table 2 W7500x sleep mode summary

Mode	Entry	Wakeup	Effect on clocks
Sleep mode	DEEPSLEEP = 0 Enable WFI	Any interrupt	CPU clock OFF APB Bus Clock ON AHB Bus clock ON Memory clocks ON
	DEEPSLEEP = 0 Enable WFE	Wakeup event	
Deep Sleep mode	DEEPSLEEP = 1 Enable WFI	Any interrupt	CPU clock OFF APB Bus Clock OFF AHB Bus clock OFF Memory clocks OFF
	DEEPSLEEP = 1 Enable WFE	Wakeup event	

## Peripheral clock gating

In Run mode, individual clocks can be stopped at any time to reduce power.

Peripheral clock gating is controlled by the CRG block.

Below is the list of clocks which can be gating in CRG block.

- ADC clock (ADCCLK)
- SSP0, SSP1 clock (SSPCLK)
- UART0, UART1 clock (UARTCLK)
- Two Timer clocks (TIMCLK0, TIMCLK1)
- 8ea PWM clocks (PWMCLK0 ~ PWMCLK7)
- WDOG clock (WDOGCLK)
- Random number generator clock (RNGCLK)

---

## 9 System tick timer

### 9.1 Introduction

System tick timer(SysTick) is part of the ARM Cortex-M0 core

### 9.2 Features

Simple 24bit timer. Clocked internally by the system clock.

### 9.3 Functional description

The SysTick timer is an integral part of Cortex-M0. The SysTick timer is intended to generate a fixed 10 millisecond interrupt for use by an operating system or other system management software.

Since the SysTick timer is a part of the Cortex-M0, it facilitates porting of software by providing a standard timer that is available on Cortex-M0 based devices. The SysTick timer can be used for :

- An RTOS tick timer which fires at a programmable rate (for example 100 Hz) and invokes a SysTick routine.
- A high-speed alarm timer using the core clock.
- A simple counter. Software can use this to measure time to completion and time used.
- An internal clock source control based on missing/meeting durations. The COUNTFLAG bit-field in the control and status register can be used to determine if an action completed within a set duration, as part of a dynamic clock management control loop.

## 9.4 Registers (Base : 0xE000\_E000)

### System Timer control and status register (SYST\_CSR)

Address Offset : 0x010

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	CNTFLAG														
															R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TICKINT	ENABLE													
														R/W	R/W

[0] ENABLE - Enables the counter

0 : Counter disabled

1 : Counter enabled

[1] TICKINT - Enables SysTick exception request

0 : Counting down to zero does not assert the SysTick exception request

1 : Counting down to zero asserts the SysTick exception request

[16] COUNTFLAG - Returns 1 if timer counted to 0 since the last read of this register.

### SysTick Reload Value Register (SYST\_RVR)

Address Offset : 0x014

Reset value : 0x0000\_0000

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

[23:0] RELOAD - Value to load into the SYST\_CVR when the counter is enabled and when it reaches 0

- The RELOAD value can be any value in the range 0x0000\_0001 - 0x00FFFFFF. You can program a value of 0, but this has no effect because the SysTick exception request and COUNTFLAG are activated when count from 1 to 0.

- To generate a multi-short timer with a period of N processor clock cycles, use a RELOAD value of N-1. For example, if the SysTick interrupt is required every 100 clock pulses, set RELOAD to 99.

## SysTick Current Value Register (SYST\_CVR)

Address Offset : 0x018

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	CURRENT[23:16]														
								R/W							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRENT [15:0]								R/W							

[23:0] CURRENT - Reads return the current value of the SysTick counter.

A write of any value clears the field to 0, and also clears the SYST\_CSR.COUNTFLAG bit to 0.

## SysTick Calibration Value Register (SYST\_CALIB)

Address Offset : 0x01C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NORF	SKEW	res	res	res	res	res	res	TENMS[23:16]							
R	R							R							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TENMS [15:0]								R							

[23:0] TENMS - Reads as zero. Indicates calibration value is not known.

[30] SKEW - Reads as one. Calibration value for the 10ms inexact timing is not known because TENMS is not known. This can affect the suitability of SysTick as a software real time clock.

[31] NORF - Reads as one. Indicates that no separate reference clock is provided.

## 10 Booting Sequence

W7500x has three different boot modes that can be selected through the BOOT pin and TEST pin as shown in Table 3.

Table 3 operation of mode selection

Mode selection		Mode	Aliasing
TEST	BOOT		
0	0	APP	User code execute in Main Flash memory.
0	1	ISP	In this mode, W7500x can support ISP function in order to control flash using serial interface.

When W7500x is reset by hardware, it will be operated as below in embedded boot code.

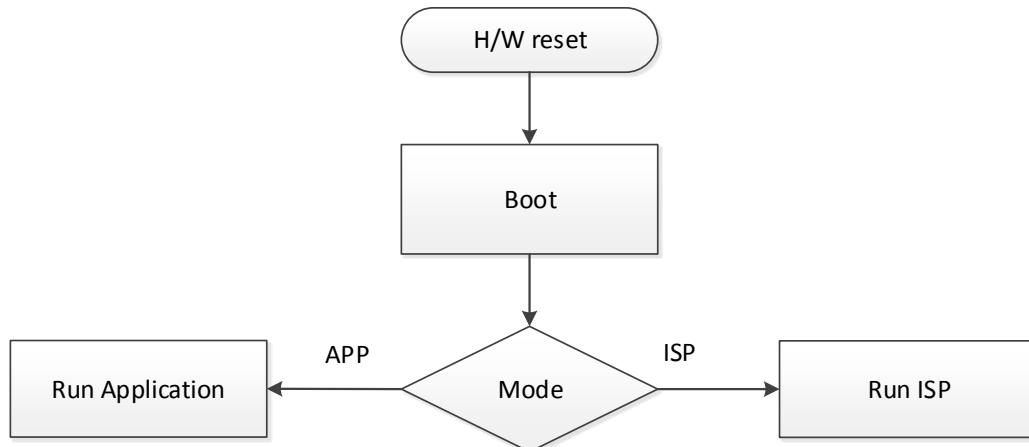


Figure 3. operation of boot code

## 11 Embedded Flash memory

### 11.1 Flash main features

- Up to 128Kbytes of Flash memory
- Memory organization:
  - Main Flash memory block:  
Up to 128Kbytes
  - Information block:  
Up to 512bytes  
Information block is read only
  - Data block:  
Up to 512bytes
- Flash memory interface features:
  - Read interface with prefetch buffer( 1 x 32-bit words )
  - Flash Program / Erase operation
- Flash memory lock function:
  - It can be controlled only by ISP tool provided by WIZnet

### 11.2 Flash memory organization

The Flash memory is organized of 32-bit wide memory cells that can be used for storing both code and data constants.

The memory organization is based on a main Flash memory block containing 512 sectors of 256byte or 32 blocks of 4Kbyte.

Table 4 description of Flash memory

Flash area	Flash memory address	Size (bytes)	Name	Description
Main Flash memory	0x0000 0000 ~ 0x0000 00FF	256	Sector 0	Block 0
	0x0000 0100 ~ 0x0000 01FF	256	Sector 1	
	0x0000 0200 ~ 0x0000 02FF	256	Sector 2	
	0x0000 0300 ~ 0x0000 03FF	256	Sector 3	
	:	:	:	Block 7
	0x0000 7000 ~ 0x0000 70FF	256	Sector112	
	0x0000 7100 ~ 0x0000 71FF	256	Sector113	
	0x0000 7200 ~ 0x0000 72FF	256	Sector114	
	0x0000 7300 ~ 0x0000 73FF	256	Sector115	

	⋮	⋮	⋮	⋮
	0x0001 FC00 ~ 0x0001 FCFF	256	Sector509	Block 32
	0x0001 FD00 ~ 0x0001 FDFF	256	Sector510	
	0x0001 FE00 ~ 0x0001 FEFF	256	Sector511	
	0x0001 FF00 ~ 0x0001 FFFF	256	Sector512	
Information block	0x0003 FC00 ~ 0x0003 FCFF	256		Lock info
	0x0003 FD00 ~ 0x0003 FDFF		Reserved	
Data block	0x0003 FE00 ~ 0x0003 FEFF	256		Data0
	0x0003 FF00 ~ 0x0003 FFFF	256		Data1

The W7500 embedded Flash memory can be programmed using in-application programming. IAP allows the user to re-program the Flash memory while the application is running. The program and erase operations can be performed over the whole product voltage range.

In

W7500x\_Library\_Examples([https://github.com/Wiznet/W7500/tree/master/W7500x\\_Library\\_Examples/Projects/Peripheral\\_Examples/Flash/IAP\\_Example](https://github.com/Wiznet/W7500/tree/master/W7500x_Library_Examples/Projects/Peripheral_Examples/Flash/IAP_Example)), there is the IAP Example Project and the below function is supported to use IAP.

```
void DO_IAP( uint32_t id, uint32_t dst_addr, uint8_t* src_addr, uint32_t size);
```

This function requests those parameters, id, dst\_addr, src\_addr and size. 'id' is already defined in 'main.c'. 'dst\_addr' is the flash memory address in the upper table. 'src\_addr' is the buffer pointer user want to program. 'size' is the flash size user chooses. Please refer flash address and size mentioned in the upper table.

```
// IAP 'id' parameter define

#define IAP_ENTRY          0x1FFF1001    // Because Thum code

#define IAP_ERAS           0x010

#define IAP_ERAS_DAT0      (IAP_ERAS + 0) // Erase Data 0 block

#define IAP_ERAS_DAT1      (IAP_ERAS + 1) // Erase Data 1 block

#define IAP_ERAS_SECT      (IAP_ERAS + 2) // Erase a Sector in Main Flash Memory

#define IAP_ERAS_BLCK      (IAP_ERAS + 3) // Erase a Block in Main Flash Memory
```

```
#define IAP_ERAS_CHIP      (IAP_ERAS + 4) // Erase all code  
  
#define IAP_ERAS_MASS      (IAP_ERAS + 5) // Erase all code & data  
  
#define IAP_PROG           0x022
```

This is how to Erase and Program flash memory. Especially, with IAP\_ERAS\_DAT0 and IAP\_ERAS\_DAT1, there is no need to put other parameters (there are default values).

```
// Step 1 DATA0 Erase, Read, Write Test  
  
DO_IAP(IAP_ERAS_DAT0,0,0,0);  
  
DO_IAP(IAP_PROG,DAT0_START_ADDR,buffer,SECT_SIZE);
```

Operating program can be deleted when user use 'IAP\_ERAS\_CHIP' or 'IAP\_ERAS\_MASS'.

```
// Using IAP_ERAS_CHIP or IAP_ERAS_MASS  
  
DO_IAP(IAP_ERAS_CHIP,0,0,0);  
  
DO_IAP(IAP_ERAS_MASS,0,0,0);
```

## 12 Clock Reset generator (CRG)

### 12.1 Introduction

CRG is clock reset generator block for W7500x System. It provides every clock/reset for all other block include CPU and peripherals. CRG includes PLL and POR.

### 12.2 Features

#### Reset

- Three types of reset - external reset, Power reset, system reset
- External reset is generated by low level on the RSTn pin (external reset)
- Power reset is generated by Power-on reset (POR)
- Power on reset is generated by POR

- System reset is generated when one of the following events occurs
  - Watchdog event
  - After remapping
  - Software reset (SYSRESETREQ bit in Cortex-M0. Refer to the Cortex-M0 technical reference manual for more detail)
- Power reset sets all registers to their reset values.
- System reset sets all registers to their reset values except the CRG block registers and remap register to protect remap value

## Clock

Two clock sources can be used to drive the system clock.

- External oscillator clock (8MHz ~ 24MHz) (OCLK)
- Internal 8MHz RC oscillator clock (RCLK)

One additional clock source

- 32.768KHz low speed external crystal which derives the real time clock.

There is a PLL

One PLL is integrated

- Input clock range is from 8MHz to 24MHz
- Frequency can be generated by M/N/OD registers. (refer register description)
- Bypass option enabled

There are many generated clocks for independent operating with system clock

- System clock (FCLK)
- ADC clock (ADCCLK)
- SSP0, SSP1 clock (SSPCLK)
- UART0, UART1 clock (UARTCLK)
- Two Timer clocks (TIMCLK0, TIMCLK1)
- 8ea PWM clocks (PWMCLK0 - PWMCLK7)
- Real time clock (RTCCLK)
- WDOG clock (WDOGCLK)
- Random number generator clock (RNGCLK)

RNGCLK have only one source (pll output) and no prescaler

Some of the generated clocks turn off automatically when CPU enters sleep mode.

- ADCCLK, RNGCLK

Generate two Hardware TCPIP Clocks (MII\_RXC, MII\_TXC) are from external PADs.

Hardware TCPIP Clocks can be gated by register control.

All clocks generated from CRG can be monitored.

## 12.3 Functional description

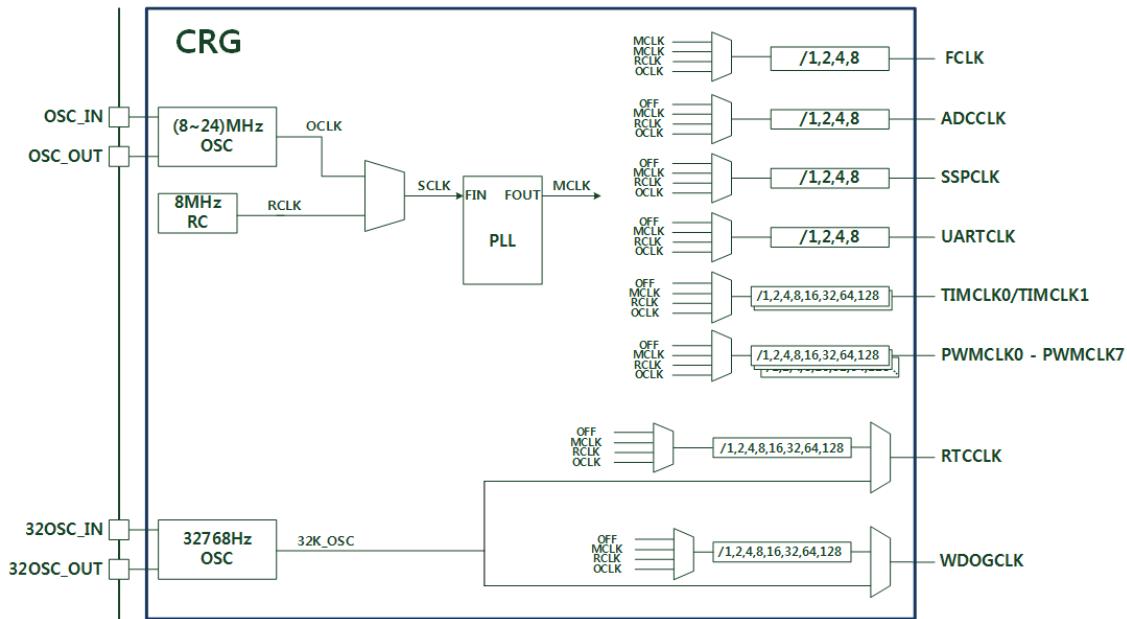


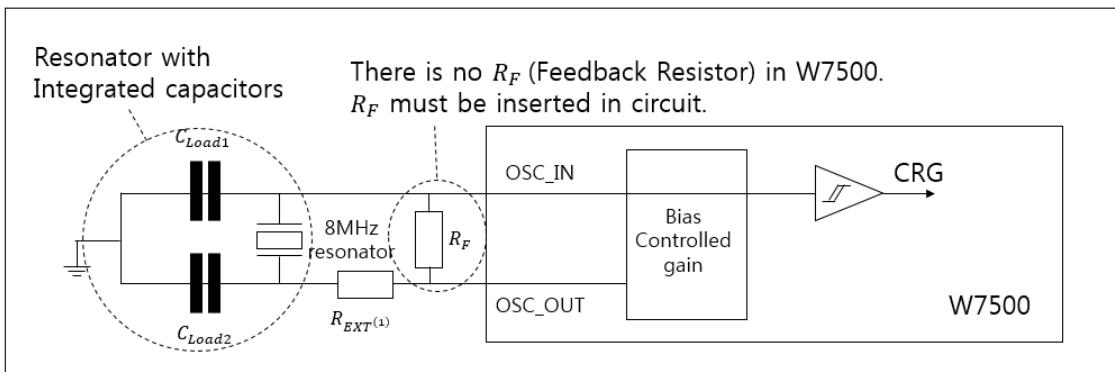
Figure 4 CRG block diagram

### External Oscillator Clock

The External oscillator clock (OCLK) can be supplied with a 8 to 24 MHz crystal/ceramic resonator oscillator. In the Typical application, Figure 5,  $R_F$  must be inserted in External oscillator clock circuit. In W7500x, there is no supported  $R_F$  for External oscillator clock (see Figure 5).

For  $C_{Load1}$  and  $C_{Load2}$ , it is recommended to use external ceramic capacitors in the 5 pF to 25 pF range(typ.) and are usually the same size, designed for application, and selected to match the requirements of the crystal or resonator (see Figure 5).

Refer to the crystal resonator manufacturer for more details on the resonator characteristics (frequency, package, accuracy).



<sup>1</sup>  $R_{EXT}$  value depends on the crystal characteristics

Figure 5 Typical application with an 8 MHz crystal

<sup>1</sup>  $R_{EXT}$  value depends on the crystal characteristics

## RC oscillator clock

RC oscillator clock (RCLK) signal is generated from an internal 8MHz RC oscillator.

RC oscillator has the advantage of providing a clock source at low cost (no external components). However the RC oscillator is less accurate than the external crystal or ceramic resonator.

- Accuracy : 1% at  $T_A = 25^\circ\text{C}$  (User don't need to calibration)

## PLL

The internal PLL can be used to multiply the External Oscillator Clock (OCLK) or RC Oscillator Clock (RCLK). PLL input can be selected by register.

PLL output clock can be generated by following the equations below.

- $F_{OUT} = F_{IN} \times M / N \times 1 / OD$
- Where:
- $M = M[5] \times 2^5 + M[4] \times 2^4 + M[3] \times 2^3 + M[2] \times 2^2 + M[1] \times 2 + M[0] \times 1$
- $N = N[5] \times 2^5 + N[4] \times 2^4 + N[3] \times 2^3 + N[2] \times 2^2 + N[1] \times 2 + N[0] \times 1$
- $OD = 2^{(2 \times OD[1])} \times 2^{(1 \times OD[0])}$

## Generated clock

Each generated clock source can be selected among 3 clock source as independent by each clock source select register.

- PLL output clock (MCLK)
- Internal 8MHz RC oscillator clock (RCLK)
- External oscillator clock (8MHz ~ 24MHz) (OCLK)

Each generated clock has own prescaler which can be selected individually by each prescale value register.

- FCLK, ADCCLK, SSPCLK, UARTCLK : 1/1, 1/2, 1/4, 1/8
- TIMCLK0, TIMCLK1, PWMCLK0 - PWMCLK7, RTCCLK, WDOGCLK : 1/1, 1/2, 1/4, 1/8, 1/16, 1/32, 1/64, 1/128

## 12.4 Registers (Base address : 0x4100\_1000)

### OSC power down register (OSC\_PDR)

Address offset : 0x000

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	OSCPD														

[0] OSCPDP - Internal 8MHz RC oscillator power down register

This bit written by S/W to RCOSC enter sleep mode or not

0 : normal operation

1 : power down (enter sleep mode)

### PLL power down register (PLL\_PDR)

Address offset : 0x010

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PLLPD														

[0] PLLPD - PLL power down register

This bit written by S/W to PLL power down or not

0 : power down

1 : normal operation

### PLL frequency calculating register (PLL\_FCR)

Address offset : 0x014

Reset value : 0x0005\_0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	M														

													R/W
--	--	--	--	--	--	--	--	--	--	--	--	--	-----

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res			N			res	res	res	res	res	res	OD		
				R/W											R/W

[1:0] OD

[13:8] N

[21:16] M

These bits are written by S/W to set frequency of PLL output.

PLL output frequency FOUT is calculated by the following equations:

$$FOUT = FIN \times M / N \times 1 / OD$$

Where:

$$M = M[5] \times 32 + M[4] \times 16 + M[3] \times 8 + M[2] \times 4 + M[1] \times 2 + M[0] \times 1 \quad (2 \sim 63)$$

$$N = N[5] \times 32 + N[4] \times 16 + N[3] \times 8 + N[2] \times 4 + N[1] \times 2 + N[0] \times 1 \quad (1 \sim 63)$$

$$OD = 2^{(2 \times OD[1])} \times 2^{(1 \times OD[0])}$$

## PLL output enable register (PLL\_OER)

Address offset : 0x018

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PLLOEN														
															R/W

[0] PLLOEN - output enable register of PLL

This bit written by S/W to control output enable of PLL

0 : Clock out is disable. VCO is working but FOUT is low only.

1 : Clock out is enable.

## PLL bypass register (PLL\_BPR)

Address offset : 0x01c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
res	PLLBP															
															R/W	

[0] PLLBP - bypass register of PLL

This bit written by S/W to control bypass or not of PLL

0 : bypass disable. Normal operation

1 : bypass enable. Clock out is clock input

## PLL input clock source select register (PLL\_IFSR)

Address offset : 0x020

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
res																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
res	PLLIS															
															R/W	

[0] PLLIS - select register of PLL input clock source

This bit written by S/W to select

0 : Internal 8MHz RC oscillator clock (RCLK)

1 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

## FCLK source select register (FCLK\_SSR)

Address offset : 0x030

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
res																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
res	FCKSRC															
															R/W	

[1:0] FCKSRC - select register of FCLK clock source

These bits are written by S/W to select

- 
- 00, 01 : output clock of PLL (MCLK)  
 10 : Internal 8MHz RC oscillator clock (RCLK)  
 11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

### FCLK prescale value select register (FCLK\_PVSR)

Address offset : 0x034

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	FCKPRE														
															R/W

[1:0] FCKPRE - select prescale value of FCLK clock

These bits are written by S/W to select

00 : 1/1 (bypass)

01 : 1/2

10 : 1/4

11 : 1/8

### SSPCLK source select register (SSPCLK\_SSR)

Address offset : 0x040

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	SSPCSS														
															R/W

[1:0] SSPCSS - SSPCLK clock source select register.

These bits are written by S/W to select clock source

00 : disable clock

01 : PLL output clock (MCLK)

10 : Internal 8MHz RC oscillator clock (RCLK)

11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

## SSPCLK prescale value select register (SSPCLK\_PVSR)

Address offset : 0x044

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	SSPCP														
															R/W

[1:0] SSPCP - select prescale value of SSPCLK clock

These bits are written by S/W to select

00 : 1/1 (bypass)

01 : 1/2

10 : 1/4

11 : 1/8

## ADCCLK source select register (ADCCLK\_SSR)

Address offset : 0x060

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	ADCSS														
															R/W

[1:0] ADCSS - ADCCLK clock source select register.

These bits are written by S/W to select clock source

00 : disable clock

01 : PLL output clock (MCLK)

10 : Internal 8MHz RC oscillator clock (RCLK)

11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

## ADCCLK prescale value select register (ADCCLK\_PVSR)

Address offset : 0x064

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	ADCCP														
														R/W	

[1:0] ADCCP - select prescale value of ADCCLK clock

These bits are written by S/W to select

00 : 1/1 (bypass)

01 : 1/2

10 : 1/4

11 : 1/8

### TIMER0CLK source select register (TIMER0CLK\_SSR)

Address offset : 0x070

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TOCSS														
														R/W	

[1:0] TOCSS - TIMCLK0 clock source select register.

These bits are written by S/W to select clock source

00 : disable clock

01 : PLL output clock (MCLK)

10 : Internal 8MHz RC oscillator clock (RCLK)

11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

### TIMER0CLK prescale value select register (TIMER0CLK\_PVSR)

Address offset : 0x074

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	T0CPS														
														R/W	

[2:0] T0CPS - select prescale value of TIM0CLK clock

These bits are written by S/W to select

000 : 1/1 (bypass)

001 : 1/2

010 : 1/4

011 : 1/8

100 : 1/16

101 : 1/32

110 : 1/64

111 : 1/128

### TIMER1CLK source select register (TIMER1CLK\_SSR)

Address offset : 0x080

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	T1CSS														
														R/W	

[1:0] T1CSS - TIMCLK1 clock source select register.

These bits are written by S/W to select clock source

00 : disable clock

01 : PLL output clock (MCLK)

10 : Internal 8MHz RC oscillator clock (RCLK)

11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

### TIMER1CLK prescale value select register (TIMER1CLK\_PVSR)

Address offset : 0x084

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	T1CPS														
														R/W	

[2:0] T1CPS - select prescale value of TIM1CLK clock

These bits are written by S/W to select

000 : 1/1 (bypass)

001 : 1/2

010 : 1/4

011 : 1/8

100 : 1/16

101 : 1/32

110 : 1/64

111 : 1/128

### PWM0CLK source select register (PWM0CLK\_SSR)

Address offset : 0x0b0

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P0CSS														
														R/W	

[1:0] P0CPS - PWMCLK0 clock source select register.

These bits are written by S/W to select clock source

00 : disable clock

01 : PLL output clock (MCLK)

10 : Internal 8MHz RC oscillator clock (RCLK)

11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

### PWM0CLK prescale value select register (PWM0CLK\_PVSR)

Address offset : 0x0b4

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

res																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P0CPS														
															R/W

[2:0] P0CPS - select prescale value of PWM0CLK clock

These bits are written by S/W to select

000 : 1/1 (bypass)

001 : 1/2

010 : 1/4

011 : 1/8

100 : 1/16

101 : 1/32

110 : 1/64

111 : 1/128

## PWM1CLK source select register (PWM1CLK\_SSR)

Address offset : 0x0c0

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P1CSS														
															R/W

[1:0] P1CSS - PWMCLK1 clock source select register.

These bits are written by S/W to select clock source

00 : disable clock

01 : PLL output clock (MCLK)

10 : Internal 8MHz RC oscillator clock (RCLK)

11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

## PWM1CLK prescale value select register (PWM1CLK\_PVSR)

Address offset : 0x0c4

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P1CPS														
														R/W	

[2:0] P1CPS - select prescale value of PWM1CLK clock

These bits are written by S/W to select

000 : 1/1 (bypass)

001 : 1/2

010 : 1/4

011 : 1/8

100 : 1/16

101 : 1/32

110 : 1/64

111 : 1/128

### PWM2CLK source select register (PWM2CLK\_SSR)

Address offset : 0x0d0

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P2CSS														
														R/W	

[1:0] P2CSS - PWMCLK2 clock source select register.

These bits are written by S/W to select clock source

00 : disable clock

01 : PLL output clock (MCLK)

10 : Internal 8MHz RC oscillator clock (RCLK)

11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

### PWM2CLK prescale value select register (PWM2CLK\_PVSR)

Address offset : 0x0d4

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P2CPS														
														R/W	

[2:0] PWM2CLK\_PRE - select prescale value of PWM2CLK clock

These bits are written by S/W to select

000 : 1/1 (bypass)

001 : 1/2

010 : 1/4

011 : 1/8

100 : 1/16

101 : 1/32

110 : 1/64

111 : 1/128

## PWM3CLK source select register (PWM3CLK\_SSR)

Address offset : 0x0e0

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P3CSS														
														R/W	

[1:0] P3CSS - PWMCLK3 clock source select register.

These bits are written by S/W to select clock source

00 : disable clock

01 : PLL output clock (MCLK)

10 : Internal 8MHz RC oscillator clock (RCLK)

11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

## PWM3CLK prescale value select register (PWM3CLK\_PVSR)

Address offset : 0x0e4

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P3CPS														
														R/W	

[2:0] P3CPS - select prescale value of PWM3CLK clock

These bits are written by S/W to select

000 : 1/1 (bypass)

001 : 1/2

010 : 1/4

011 : 1/8

100 : 1/16

101 : 1/32

110 : 1/64

111 : 1/128

## PWM4CLK source select register (PWM4CLK\_SSR)

Address offset : 0x0f0

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P4CSS														
														R/W	

[1:0] P4CSS - PWMCLK4 clock source select register.

These bits are written by S/W to select clock source

00 : disable clock

01 : PLL output clock (MCLK)

10 : Internal 8MHz RC oscillator clock (RCLK)

11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

## PWM4CLK prescale value select register (PWM4CLK\_PVSR)

Address offset : 0x0f4

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P4CPS														
															R/W

[2:0] P4CPS - select prescale value of PWM4CLK clock

These bits are written by S/W to select

000 : 1/1 (bypass)

001 : 1/2

010 : 1/4

011 : 1/8

100 : 1/16

101 : 1/32

110 : 1/64

111 : 1/128

## PWM5CLK source select register (PWM5CLK\_SSR)

Address offset : 0x100

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P5CSS														
															R/W

[1:0] P5CSS - PWMCLK5 clock source select register.

These bits are written by S/W to select clock source

00 : disable clock

01 : PLL output clock (MCLK)

10 : Internal 8MHz RC oscillator clock (RCLK)

11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

### PWM5CLK prescale value select register (PWM5CLK\_PVSR)

Address offset : 0x104

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P5CPS	R/W													

[2:0] P5CPS - select prescale value of PWM5CLK clock

These bits are written by S/W to select

000 : 1/1 (bypass)

001 : 1/2

010 : 1/4

011 : 1/8

100 : 1/16

101 : 1/32

110 : 1/64

111 : 1/128

### PWM6CLK source select register (PWM6CLK\_SSR)

Address offset : 0x110

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P6CSS	R/W													

[1:0] P6CSS - PWMCLK6 clock source select register.

These bits are written by S/W to select clock source

00 : disable clock

01 : PLL output clock (MCLK)

- 10 : Internal 8MHz RC oscillator clock (RCLK)  
 11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

### PWM6CLK prescale value select register (PWM6CLK\_PVSR)

Address offset : 0x114

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	res														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P6CPS														
														R/W	

[2:0] P6CPS - select prescale value of PWM6CLK clock

These bits are written by S/W to select

000 : 1/1 (bypass)

001 : 1/2

010 : 1/4

011 : 1/8

100 : 1/16

101 : 1/32

110 : 1/64

111 : 1/128

### PWM7CLK source select register (PWM7CLK\_SSR)

Address offset : 0x120

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	res														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P7CSS														
														R/W	

[1:0] P7CSS - PWMCLK7 clock source select register.

These bits are written by S/W to select clock source

00 : disable clock

- 
- 01 : PLL output clock (MCLK)  
 10 : Internal 8MHz RC oscillator clock (RCLK)  
 11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

## PWM7CLK prescale value select register (PWM7CLK\_PVSR)

Address offset : 0x124

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	P7CPS														
														R/W	

[2:0] P7CPS - select prescale value of PWM7CLK clock

These bits are written by S/W to select

000 : 1/1 (bypass)

001 : 1/2

010 : 1/4

011 : 1/8

100 : 1/16

101 : 1/32

110 : 1/64

111 : 1/128

## RTC High Speed source select register (RTC\_HS\_SSR)

Address offset : 0x130

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	RTCHS														
														R/W	

[1:0] RTCHS - RTCCLK\_hs clock source select register.

These bits are written by S/W to select clock source

- 
- 00 : disable clock  
 01 : PLL output clock (MCLK)  
 10 : Internal 8MHz RC oscillator clock (RCLK)  
 11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

## RTC High Speed prescale value select register (RTC\_HS\_PVSR)

Address offset : 0x134

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	R/W														

[2:0] RTCPRE - select prescale value of RTCCLK\_hs clock

These bits are written by S/W to select

- 000 : 1/1 (bypass)
- 001 : 1/2
- 010 : 1/4
- 011 : 1/8
- 100 : 1/16
- 101 : 1/32
- 110 : 1/64
- 111 : 1/128

## RTC source select register (RTC\_SSR)

Address offset : 0x13c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	R/W														

[0] RTCSEL - RTCCLK clock source select register.

These bits are written by S/W to select clock source

0 : RTCCLK\_hs

1 : 32K\_OSC\_CLK (Low speed external oscillator clock)

### WDOGCLK High Speed source select register (WDOGCLK\_HS\_SSR)

Address offset : 0x140

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	WDHS														
															R/W

[1:0] WDHS - WDOGCLK\_hs clock source select register.

These bits are written by S/W to select clock source

00 : disable clock

01 : PLL output clock (MCLK)

10 : Internal 8MHz RC oscillator clock (RCLK)

11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

### WDOGCLK High Speed prescale value select register (WDOGCLK\_HS\_PVSR)

Address offset : 0x144

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	WDPRE														
															R/W

[2:0] WDPRE - select prescale value of WDOGCLK\_hs clock

These bits are written by S/W to select

000 : 1/1 (bypass)

001 : 1/2

010 : 1/4

011 : 1/8

100 : 1/16

101 : 1/32

110 : 1/64

111 : 1/128

### WDOGCLK clock source select register (WDOGCLK\_SSR)

Address offset : 0x14c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	WDSEL														
															R/W

[0] WDSEL - WDOGCLK clock source select register.

These bits are written by S/W to select clock source

0 : WDOGCLK\_hs

1 : 32K\_OSC\_CLK (Low speed external oscillator clock)

### UARTCLK source select register (UARTCLK\_SSR)

Address offset : 0x150

Reset value : 0x0000\_0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	UCSS														
															R/W

[1:0] UCSS - UARTCLK clock source select register.

These bits are written by S/W to select clock source

00 : disable clock

01 : PLL output clock (MCLK)

10 : Internal 8MHz RC oscillator clock (RCLK)

11 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

## UARTCLK prescale value select register (UARTCLK\_PVSR)

Address offset : 0x154

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	UCP														
															R/W

[1:0] UCP - select prescale value of UARTCLK clock

These bits are written by S/W to select

00 : 1/1 (bypass)

01 : 1/2

10 : 1/4

11 : 1/8

## MIICLK enable control register (MIICLK\_ECR)

Address offset : 0x160

Reset value : 0x0000\_0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	MIITEN	MIIREN													
														R/W	R/W

[0] MIIREN - MII RX Clock source enable register

This bit is written by S/W to set enable or disable

0 : Disable MII\_RCK and MII\_RCK\_N

1 : Enable MII\_RCK and MII\_RCK\_N

[1] MIITEN - MII TX Clock source enable register

This bit is written by S/W to set enable or disable

0 : Disable MII\_TCK and MII\_TCK\_N

1 : Enable MII\_TCK and MII\_TCK\_N

## Monitoring Clock source select register (MONCLK\_SSR)

Address offset : 0x170

Reset value : 0x0000\_0000

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

[4:0] CLKMON\_SEL - Select clock source for monitoring (monitoring pin : PA\_02)

This bit is written by S/W to set enable or disable

00000 : PLL output clock (MCLK)

00001 : FCLK

00010 : Internal 8MHz RC oscillator clock (RCLK)

00011 : External oscillator clock (OCLK, 8MHz ~ 24MHz)

00100 : ADCCLK

00101 : SSPCLK

00110 : TIMCLK0

00111 : TIMCLK1

01000 : PWMCLK0

01001 : PWMCLK1

01010 : PWMCLK2

01011 : PWMCLK3

01100 : PWMCLK4

01101 : PWMCLK5

01110 : PWMCLK6

01111 : PWMCLK7

10000 : UARTCLK

10001 : MII\_RCK

10010 : MII\_TCK

10011 : RTCCLK

## 12.5 Register map

The following table summarizes the CRG registers.



Table 5 CRG register map and reset values

## 13 Tcp/ip core Offload Engine (TOE)

## 13.1 Introduction

The TCP/IP Core Offload Engine (TOE) is a Hardwired TCP/IP embedded Ethernet controller that provides easier Internet connection to embedded systems. TOE enables users to have Internet connectivity in their applications by using the TCP/IP stack.

WIZnet's Hardwired TCP/IP is the market-proven technology that supports TCP, UDP, IPv4, ICMP, ARP, IGMP, and PPPoE protocols. TOE embeds the 32Kbyte internal memory buffer for the Ethernet packet processing. Using TOE allows users to implement the Ethernet application by adding the simple socket program. It's faster and easier than using any other Embedded Ethernet solutions. 8 independent hardware sockets can be used simultaneously.

TOE also provides WOL (Wake on LAN) to reduce power consumption of the system.

## 13.2 Features

- Supports Hardwired TCP/IP Protocols : TCP, UDP, ICMP, IPv4, ARP, IGMP, PPPoE
  - Supports 8 independent sockets simultaneously
  - Supports Power down mode
  - Supports Wake on LAN over UDP
  - Internal 32Kbytes Memory for TX/RX Buffers
  - Not supports IP Fragmentation

### 13.3 Functional description

Figure 6 shows the TOE block diagram.

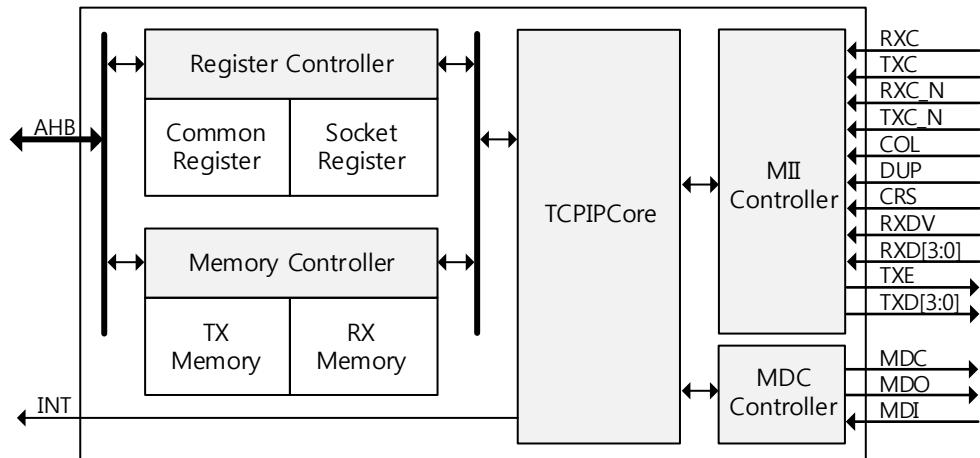


Figure 6 TOE block diagram

### 13.4 TOE Memory map

TOE has one Common Register Block, eight Socket Register Blocks, and TX/RX Buffer Blocks allocated to each Socket. Figure 7 shows the selected block by the base address and the available offset address range of Socket TX/RX Buffer Blocks. Each Socket's TX Buffer Block physically exists in one 16KB TX memory and is initially allocated with 2KB. Also, Each Socket's RX Buffer Block physically exists in one 16KB RX Memory and is initially allocated with 2KB. Regardless of the allocated size of each Socket TX/RX Buffer, it can be accessible within the 16 bits offset address range (From 0x0000 to 0xFFFF).

## Blocks

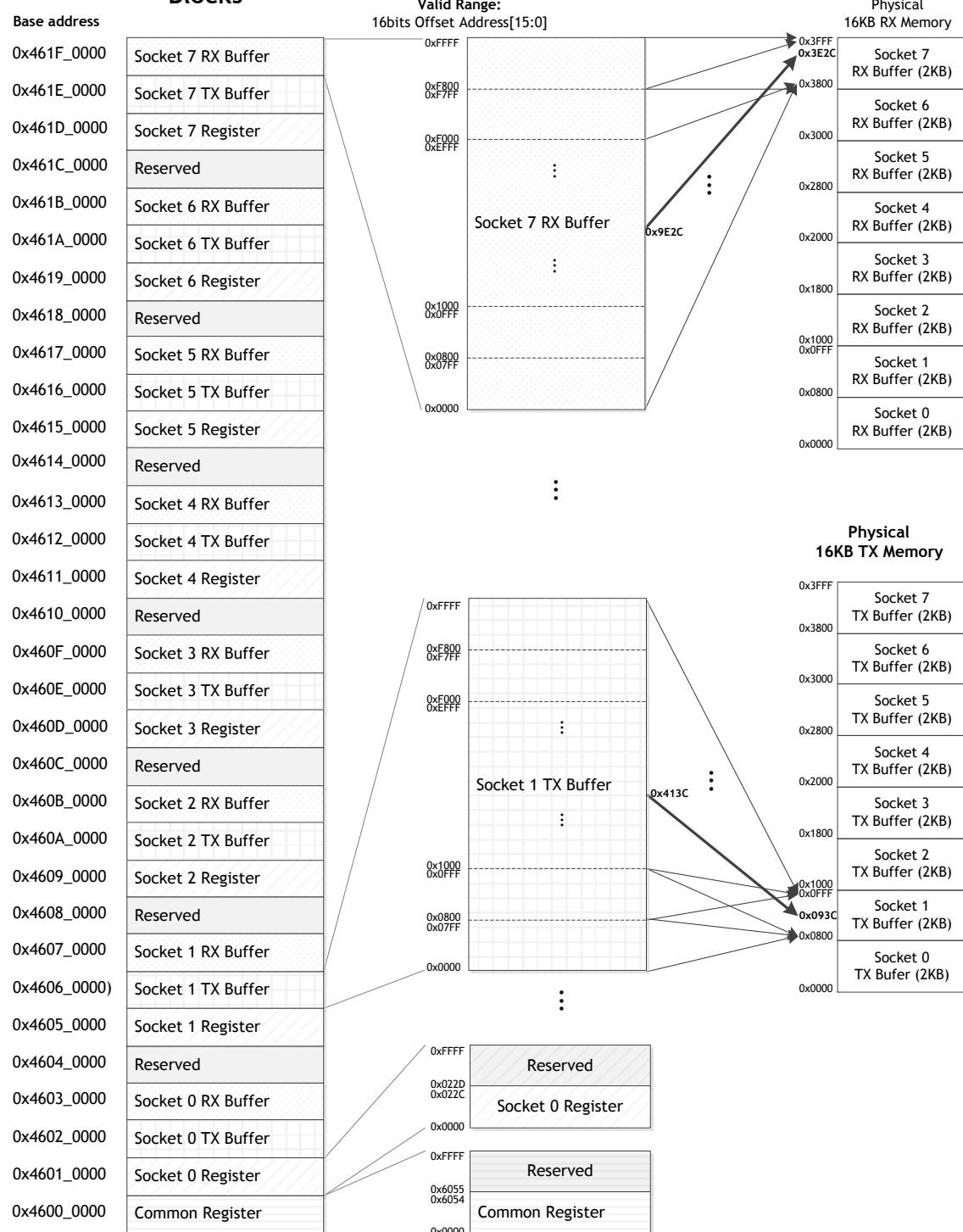


Figure 7. Register & Memory Organization

## Common register map

Common Register Block configures the general information of TOE such as IP and MAC address.

<Table 6> defines the offset address of registers in this block.

Table 6. Offset Address for Common Register

Address	Register
0x0000	TOE Version (VERSIONR)
0x2000	TICKCLOK (TCLKR)
0x2100	Interrupt (IR)
0x2104	Interrupt Mask (IMR)
0x2108	Interrupt Clear (IRCR)
0x2110	Socket Interrupt (SIR)
0x2114	Socket Mask (SIMR)
0x2300	Mode (MR)
0x2400	PPP Timer (PTIMER)
0x2404	PPP Magic (PMAGIC)
0x2408	PPP Destination MAC Address (PHAR1)
0x240C	PPP Destination MAC address (PHAR0)
0x2410	PPP Session Identification (PSIDR)
0x2414	PPP Maximum Segment Size (PMSS)
0x6000	Source Hardware Address (SHAR1)
0x6004	Source Hardware Address (SHAR0)
0x6008	Gateway Address (GA)
0x600C	Subnet Mask (SUB)
0x6010	Source IP Address (SIP)
0x6020	Network Configuration Lock (NCONFL)
0x6040	Retry Time (RTR)
0x6044	Retry Counter (RCR)
0x6050	Unreachable IP Address (UIP)
0x6054	Unreachable Port Address (UPORT)

## Socket register map

TOE supports 8 Sockets for communication channel. Each Socket is controlled by Socket n Register (n = 0,...,7 , where n is socket number). <Table 2> defines the 16bits Offset Address of registers in Socket n Register Block.

Table 7. Offset Address in Socket n Register Block (n = 0,...,7, where n is Socket number)

Offset	Register
0x0000	Socket Mode (Sn_MR)
0x0010	Socket Command (Sn_CR)
0x0020	Socket Interrupt (Sn_IR)
0x0024	Socket Interrupt Mask (Sn_IMR)
0x0028	Socket Interrupt Clear (Sn_ICR)
0x0030	Socket Status (Sn_SR)
0x0100	Socket Protocol Number (Sn_PNR)
0x0104	Socket IP Type of Service (Sn_TOS)
0x0108	Socket TTL (Sn_TTLR)
0x010C	Socket Fragment Offset (Sn_FRAG)
0x0110	Socket Maximum Segment (Sn_MSSR)
0x0114	Socket Port Number (Sn_PORTR)
0x0118	Socket Destination Hardware address0 (Sn_DHAR0)
0x011C	Socket Destination Hardware address1 (Sn_DHAR1)
0x0120	Socket Destination Port Number (Sn_DPORTR)
0x0124	Socket Destination IP Address (Sn_DIPR)
0x0180	Socket Keep Alive Timer (Sn_KATMR)
0x0184	Socket Retry Time (Sn_RTR)
0x0188	Socket Retry Counter (Sn_RCR)
0x0200	Socket TX Memory Size (Sn_TXBUF_SIZE)
0x0204	Socket TX Free Size (Sn_TX_FSR)
0x0208	Socket TX Read Pointer (Sn_TX_RD)
0x020C	Socket TX Write Pointer (Sn_TX_WR)
0x0220	Socket RX Memory Size (Sn_RXBUF_SIZE)
0x0224	Socket RX Received Size (Sn_RX_RSR)
0x0228	Socket RX Read Pointer (Sn_RX_RD)
0x022C	Socket RX Write Pointer (Sn_RX_WR)

## Memory

TOE has one 16KB TX memory for Socket n TX Buffer Blocks and one 16KB RX memory for Socket n RX buffer Blocks.

16KB TX memory is initially allocated in 2KB size for each Socket TX Buffer Block (2KB X 8 = 16KB). The initial allocated 2KB size of Socket n TX Buffer can be re-allocated by using ‘Socket n TX Buffer Size Register (Sn\_TXBUF\_SIZE)’.

---

Once all Sn\_TXBUF\_SIZE registers have been configured, Socket TX Buffer is allocated with the configured size of 16KB TX Memory and is assigned sequentially from Socket 0 to Socket 7. Its physical memory address is automatically determined in 16KB TX memory. Therefore, the total sum of Sn\_TXBUF\_SIZE should not exceed 16 in case of error in data transmission.

The 16KB RX memory allocation method is the same as the 16KB TX memory allocation method. 16KB RX memory is initially allocated into 2KB size for each Socket RX Buffer Block (2KB X 8 = 16KB). The initial allocated 2KB size of Socket n RX Buffer can be re-allocated by using ‘Socket n RX Buffer Size Register (Sn\_RXBUF\_SIZE)’.

When all Sn\_RXBUF\_SIZE registers have been configured, the Socket RX Buffer is allocated with the configured size in 16KB RX Memory and is assigned sequentially from Socket 0 to Socket 7. The physical memory address of the Socket RX Buffer is automatically determined in 16KB RX memory. Therefore, the total sum of Sn\_RXBUF\_SIZE should not exceed 16 or data reception error will occur.

For 16KB TX/RX memory allocation, refer to Sn\_TXBUF\_SIZE & Sn\_RXBUF\_SIZE in ‘Chapter 0’. The Socket n TX Buffer Block allocated in 16KB TX memory is buffer for saving data to be transmitted by host. The 16bits Offset Address of Socket n TX Buffer Block has 64KB address space ranged from 0x0000 to 0xFFFF, and is configured with reference to ‘Socket n TX Write Pointer Register (Sn\_TX\_WR)’ & ‘Socket n TX Read Pointer Register(Sn\_RX\_RD)’. However, the 16bits Offset Address automatically converts into the physical address to be accessible in 16KB TX memory such as Figure 7.

The Socket n RX Buffer Block allocated in 16KB RX memory is buffer for saving the received data through the Ethernet. The 16bits Offset Address of Socket n RX Buffer Block has 64KB address space ranged from 0x0000 to 0xFFFF, and is configured with reference to ‘Socket n RX RD Pointer Register (Sn\_RX\_RD)’ & ‘Socket n RX Write Pointer Register (Sn\_RX\_WR)’. However, the 16bits Offset Address automatically converts into the physical address to be accessible in 16KB RX memory such as Figure 7.

## 13.5 Common register (Base : 0x4600\_0000)

### VERSIONR (TOE Version Register)

Address Offset : 0x0000

Reset value : 0x0000\_0005

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

[7:0] VERSION - indicates the TOE version as 0x05.

### TCKCNT (Ticker Counter Register)

Address Offset : 0x2000

Reset value : 0x0000\_07D0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	res	Res	res	res	res	res	res	res	res	res	res	res	res	res	res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCKCNT[15:0]															
R/W															

[15:0] TCKCNT - Ticker counter register is used Tick counter of 100usec. for internal timer of TOE. The unit of tick is HCLK. RTR, Sn\_RTR, Sn\_KATMR operates on the value of this register.

$$\frac{1}{f} \times TCKCNT = \text{Reference time of TOE}$$

Ex) When the reference time of TOE is set to 100us, and when  $f = 20\text{MHz}$ ,

$$\frac{1}{f} \times TCKCNT = 100\text{us}$$

$$TCKCNT = 100\text{us} \times f$$

$$TCKCNT = 2000 = 0x07D0$$

## IR (Interrupt Register)

Address Offset : 0x2100

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	es	res													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	R	R	R	R	res	res	res								

IR indicates the interrupt status. If IR is not equal to ‘0x00’, INTn PIN is asserted low until it is ‘0x00’.

### [4] WOL - Magic Packet

When WOL mode is enabled and receives the magic packet over UDP, this bit is set.

### [5] PPPoE - PPPoE Close

When PPPoE is disconnected during PPPoE mode, this bit is set.

### [6] UNREACH - Destination unreachable

When receiving the ICMP (Destination port unreachable) packet, this bit is set as ‘1’.

When this bit is ‘1’, Destination Information such as IP address and Port number may be checked with the corresponding UIPR & UPORTR.

### [7] Conflict - IP Conflict

Bit is set as ‘1’ when own source IP address is same with the sender IP address in the received ARP request.

## IMR (Interrupt Mask Register)

Address Offset : 0x2104

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	R/W	res	res	res	res	res									

IMR is used to mask interrupts. Each bit of IMR corresponds to each bit of IR. When a bit of IMR is ‘1’ and the corresponding bit of IR is ‘1’, an interrupt will be issued. In other words, if a bit of IMR is ‘0’, an interrupt will not be issued even if the corresponding bit of IR is ‘1’.

#### [4] Magic Packet

0: Disable Magic Packet Interrupt

1: Enable Magic Packet Interrupt

#### [5] PPPoE Close Interrupt Mask

0: Disable PPPoE Close Interrupt

1: Enable PPPoE Close Interrupt

#### [6] Destination unreachable Interrupt Mask

0: Disable Destination unreachable Interrupt

1: Enable Destination unreachable Interrupt

#### [7] IP Conflict Interrupt Mask

0: Disable IP Conflict Interrupt

1: Enable IP Conflict Interrupt

## IRCR (Interrupt Clear Register)

Address Offset : 0x2108

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res		IRC[8:4]		res	res	res	res								
									*R/C_W1						

IRCR is used to clear interrupts. Each bit of IR can be cleared when the host writes ‘1’ value to each bit of IRCR corresponding to each bit of IR.

*\* ReadClearWrite1 (R/C\_W1) : Software can read as well as clear this bit by writing ‘1’. Writing ‘0’ has no effect on the bit value.*

#### [4] Magic Packet Interrupt Clear

#### [5] PPPoE Close Interrupt Clear

#### [6] Destination unreachable Interrupt Clear

#### [7] IP Conflict Interrupt Clear

## SIR (Socket Interrupt Register)

Address Offset : 0x2110

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	S7	S6	S5	S4	S3	S2	S1	S0							
								R/W							

SIR indicates the interrupt status of Socket. Each bit of SIR be still ‘1’ until Sn\_IR is cleared by the host. If Sn\_IR is not equal to ‘0x00’, the n-th bit of SIR is ‘1’ and INTn PIN is asserted until SIR is ‘0x00’

[7:0] SIR - When the interrupt of Socket n occurs, the n-th bit of SIR becomes ‘1’.

## SIMR (Socket Interrupt Mask Register)

Address Offset : 0x2114

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	S7	S6	S5	S4	S3	S2	S1	S0							
								R/W							

Each bit of SIMR corresponds to each bit of SIR. When a bit of SIMR is ‘1’ and the corresponding bit of SIR is ‘1’, Interrupt will be issued. In other words, if a bit of SIMR is ‘0’, an interrupt will be not issued even if the corresponding bit of SIR is ‘1’.

[7:0] SIR - Socket n Interrupt Mast

0: Disable Socket n Interrupt

1: Enable Socket n Interrupt

## MR (Mode Register)

Address Offset : 0x2300

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	res	res	NTR	res	RST	res	WOL	PB	res	NSC	FA	res
						R/W		R/W		R/W	R/W		R/W	R/W	

MR is used for S/W reset, ping block mode and PPPoE mode

### [2] NSC - MACRAW No Size Check

If this bit is ‘1’, it does not check packet size In MACRAW mode.

0 : Enable Check Packet Size

1 : Disable Check Packet Size

### [4] PB - Ping Block

If the bit is ‘1’, it blocks the response to a ping request.

0: Disable Ping block

1: Enable Ping block

### [5] WOL - Wake on Lan

If WOL mode is enabled and the received magic packet over UDP has been normally processed, the Interrupt PIN (INTn) asserts to low. When using WOL mode, the UDP Socket should be opened with any source port number. (Refer to Socket n Mode Register (Sn\_MR) for opening Socket.)

Notice: The magic packet over UDP supported by TOE consists of 6 bytes synchronization stream ('0xFFFFFFFFFFFF') and 16 times Target MAC address stream in UDP payload. The options such like password are ignored. You can use any UDP source port number for WOL mode.

0: Disable WOL mode

1: Enable WOL mode

### [7] RST - Software Reset

If this bit is ‘1’, All internal registers will be initialized. It will be automatically cleared as ‘0’ after S/W reset.

### [10] NTR - No Send TCP Reset Packet

If this bit is ‘1’, in TCP mode, Socket does not send Reset Packet.

0 : Send Reset Pakcet

1 : No Send Reset Packet

## PTIMER (PPP Link Control Protocol Request Timer Register)

Address Offset : 0x2400

Reset value : 0x0000\_0028

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res							PTIME[7:0]								
									R/W						

[7:0] PTIME configures the time for sending LCP echo request. The unit of time is 25ms

**Ex)** in case that PTIMER is 200,

$$200 * 25(\text{ms}) = 5000(\text{ms}) = 5 \text{ seconds}$$

## PMAGICR (PPP Link Control Protocol Magic number Register)

Address Offset : 0x2404

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res							PMAGIC[7:0]								
									R/W						

PMAGICR configures the 4bytes magic number to be used in LCP echo request.

[7:0] PMAGIC

**Ex)** PMAGIC = 0x01

0x001D

0x01

LCP Magic number = 0x01010101

## PHAR (Destination Hardware Address Register in PPPoE)

Address Offset : 0x2408

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PHAR[31:24]								PHAR1[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHAR[15:8]								PHAR[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address Offset : 0x240C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PHAR[31:24]								PHAR[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res								res							

PHAR should be written to the PPPoE server hardware address acquired in PPPoE connection process.

PHAR0 and PHAR1 - configures Destination hardware address

**Ex)** In case that destination hardware address is 00:08:DC:12:34:56

PHAR0[32:24]      PHAR0[23:16]      PHAR0[15:8]      PHAR0[32:24]      PHAR1[32:24]      PHAR1[23:16]

0x00	0x08	0xDC	0x12	0x34	0x56
------	------	------	------	------	------

## PSIDR (Session ID Register in PPPoE)

Address Offset : 0x2410

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSID[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[15:0] PSID - should be written to the PPPoE sever session ID acquired in PPPoE connection process.

## PMRUR (Maximum Receive Unit Register in PPPoE)

Address Offset : 0x2414

Reset value : 0x0000\_FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMSS[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[15:0] PMRUR configures the maximum receive unit of PPPoE.

Ex) in case that maximum receive unit in PPPoE is 0x1234

PMSS[15:8] PMSS[7:0]

0x12	0x34
------	------

## SHAR (Source Hardware Address Register)

Address Offset : 0x6000

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SHAR0[31:24]								SHAR0[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SHAR0[15:8]								SPHAR0[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address Offset : 0x6004

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SHA1[31:24]								SHAR1[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res								res							

SHAR configures the source hardware address.

**Ex) In case of “00.08.DC.12.34.56”**

SHAR0[32:24]      SHAR0[23:16]      SHAR0 [15:8]      SHAR0 [32:24]      SHAR1 [32:24]      SHAR2 [23:16]

0x00	0x08	0xDC	0x12	0x34	0x56
------	------	------	------	------	------

## GAR (Gateway Address)

Address Offset : 0x6008

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GA[31:24]								GA[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
GA[15:8]								GA[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
GA[15:8]								GA[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

GAR[31:0] - configures the default gateway address

**Ex) In case of “192.168.0.1”**

GA[31:24]      GA[23:16]      GA[15:8]      GA[7:0]

192 (0xC0)	168 (0xA8)	0 (0x00)	1 (0x01)
------------	------------	----------	----------

## SUBR ( Subnet Mask Register)

Address Offset : 0x600C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SUB[31:24]								SUB[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SUBR[15:8]								SUBR[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SUBR configures the subnet mask address.

**Ex)** In case of “255.255.255.0”

SUB[31:24]	SUB[23:16]	SUB[15:8]	SUB[7:0]
255 (0xFF)	255 (0xFF)	255 (0xFF)	0 (0x00)

## SIPR (Source IP address Register)

Address Offset : 0x6010

Reset value : 0x0000\_0000

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

SIPR configures the source IP address.

**Ex)** In case of “192.168.0.2”

SIPR[31:24]	SIPR[23:16]	SIPR[15:8]	SIPR[7:0]
192 (0xC0)	168 (0xA8)	0 (0x00)	2 (0x02)

## NCONFLR ( Network Configuration Lock Register)

Address Offset : 0x6020

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NCONFL[31:24]								NCONFL [23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NCONFL [15:8]								NCONFL [7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NCONFLR is used to unlock and lock the network configuration registers which are SIR, SUBR, GAR and SHAR. When LOCK is ‘ON’, the protected registers are not able to access. In this case a value of 0x01ACCE55 is written to NCONFLR. When LOCK is ‘OFF’, the protected registers are allowed to access. In this case any value except 0x01ACCE55 is written.

#### Ex) In case of ‘LOOK is ON’

NCONFL[31:24]	NCONFL [23:16]	NCONFL [15:8]	NCONFL [7:0]
---------------	----------------	---------------	--------------

1 (0x01)	172 (0xAC)	206 (0xCE)	85 (0x55)
----------	------------	------------	-----------

## RTR (Retry Time Register)

Address Offset : 0x6040

Reset value : 0x0000\_07D0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTR[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

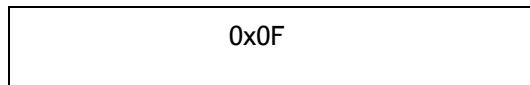
RTR configures the retransmission timeout period. The unit of timeout period is 100us(Refer to TCKCNTR) and the default of RTR is ‘0x07D0’ or ‘2000’. And so the default timeout period is 200ms(100us X 2000).

During the time configured by RTR, WZTOE waits for the peer response to the packet that is transmitted by Sn\_CR(CONNECT, DISCON, CLOSE, SEND, SEND\_MAC, SEND\_KEEP command). If the peer does not respond within the RTR time, WZTOE retransmits the packet or issues timeout.

When RTR is not ‘0x0000\_0000’, RTR is used to configure the timeout period of all socket. When RTR is ‘0x0000\_0000’, the timeout period of each socket could set by using Socket n Retry Time Register (Sn\_RTR).

**Ex) When timeout-period is set as 400ms, RTR = (400ms / 1ms) X 10 = 4000(0x0FA0)**

RTR[15:8]



## RCR (Retry Counter Register)

Address Offset : 0x6044

Reset value : 0x0000\_0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res								res							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res								RC[7:0]							
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

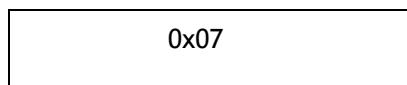
RCR configures the number of time of retransmission. When retransmission occurs as many as ‘RCR+1’, Timeout interrupt is issued (Sn\_IR[TIMEOUT] = ‘1’).

When RCR is not ‘0x0000\_0000’, RCR is used to configure the timeout period of all socket.

When RCR is ‘0x0000\_0000’, the timeout period of each socket could set by using Socket n Retry Counter Register (Sn\_RCR).

**Ex)** RCR = 0x0007

RC[7:0]



The timeout of WZTOE can be configurable with RTR and RCR. WZTOE has two kind timeout such as Address Resolution Protocol (ARP) and TCP retransmission.

At the ARP (Refer to RFC 826, <http://www.ietf.org/rfc.html>) retransmission timeout, WZTOE automatically sends ARP-request to the peer’s IP address in order to acquire MAC address information (used for communication of IP, UDP, or TCP). While waiting for ARP-response from the peer, if there is no response during the configured RTR time, a temporary timeout is occurred and ARP-request is retransmitted. It is repeated as many as ‘RCR + 1’ times. Even after the ARP-request retransmissions are repeated as ‘RCR+1’ and there is no response to the ARP-request, the final timeout is occurred and Sn\_IR(TIMEOUT) becomes ‘1’. The time of final timeout ( $ARP_{TO}$ ) of ARP-request is as below.

$$ARP_{TO} = (RTR \times 0.1ms) \times (RCR + 1)$$

At the TCP packet retransmission timeout, WZTOE transmits TCP packets (SYN, FIN, RST, DATA packets) and waits for the acknowledgement (ACK) during the configured RTR time and RCR. If there is no ACK from the peer, a temporary timeout occurs and the TCP packet is retransmitted. The retransmission is repeated as many as ‘RCR+1’. Even after TCP retransmission is repeated as ‘RCR+1’ and there is no response to the TCP retransmission, the final timeout is occurred and Sn\_IR(TIMEOUT) becomes ‘1’. The time of final timeout (TCP TO) of TCP retransmission is as below.

$$TCP_{TO} = \left( \sum_{N=0}^{M} (RTR \times 2^N) + ((RCR - M) \times RTR_{MAX}) \right) \times 0.1ms$$

N : Retransmission count,  $0 \leq N \leq M$

M : Minimum value when  $RTR \times 2^{(M+1)} > 65535$  and  $0 \leq M \leq RCR$

RTRMAX :  $RTR \times 2^M$

**Ex)** When RTR = 2000(0x07D0), RCR = 8(0x0008),

$$ARP_{TO} = 2000 \times 0.1ms \times 9 = 1800ms = 1.8s$$

$$\begin{aligned} TCP_{TO} &= (0x07D0+0x0FA0+0x1F40+0x3E80+0x7D00+0xFA00+0xFA00+0xFA00+0xFA00) \times 0.1ms \\ &= (2000 + 4000 + 8000 + 16000 + 32000 + ((8 - 4) \times 64000)) \times 0.1ms \\ &= 318000 \times 0.1ms = 31.8s \end{aligned}$$

## UIPR (Unreachable IP address Register)

Address Offset : 0x6050

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UIP[31:24]								UIPR[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UIPR[15:8]								UIPR[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TOE receives an ICMP packet(Destination port unreachable) when data is sent to a port number which socket is not open and UNREACH bit of IR becomes ‘1’ and UIPR indicates the destination IP address.

**Ex)** In case of “192.168.0.11”

UIP[31:24]	UIP[23:16]	UIP[15:8]	UIP[7:0]
192 (0xC0)	168 (0xA8)	0 (0x00)	11 (0x0E)

### UPORTR (Unreachable Port Register)

Address Offset : 0x6054

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res
UPORT[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TOE receives an ICMP packet(Destination port unreachable) when data is sent to a port number which socket is not open and UNREACH bit of IR becomes ‘1’ and UPORTR indicates the destination port number.

[15 :0] UPORT - Destination port number bits

**Ex)** In case of “0x1234”

UPORT[15:8]	UPORT[7:0]
18 (0x12)	52(0x34)

### 13.6 Socket register (Base : 0x4601\_0000 + 0x0004\_000 x n)[n=0,...7, where n is socket number]

#### Sn\_MR (Socket n Mode Register)

Address Offset : 0x0000

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

**Sn\_MR** configures the option or protocol type of Socket n.

[3:0] These bits configures the protocol mode of Socket n as follows

Sn_MR[3:0]				Meaning
0	0	0	0	Closed
0	0	0	1	TCP
0	0	1	0	UDP
0	1	0	0	MACRAW

- MACRAW mode should be only used in Socket 0.

## [4] UNICAST Blocking and IPv6 packet Blocking

## UNICAST Blocking in UDP mode

0 : disable Unicast Blocking

1 : enable Unicast Blocking

This bit blocks receiving the unicast packet during UDP mode( $P[3:0] = '0010'$ ) and MULTI = '1'.

## IPv6 packet Blocking in MACRAW mode

0 : disable IPv6 Blocking

1 : enable IPv6 Blocking

This bit is applied only during MACRAW mode ( $P[3:0] = '0100'$ ). It blocks to receiving the IPv6 packet.

## [5] Use No Delayed ACK, Multicast and Multicast Blocking mode

## Use No Delayed ACK

0 : Disable No Delayed ACK option

1 : Enable No Delayed ACK option

This bit is applied only during TCP mode ( $P[3:0] = '0001'$ ). When this bit is '1', It sends the ACK packet without delay as soon as a

---

### Multicast

0 : using IGMP version 2

1 : using IGMP version 1

This bit is applied only during UDP mode( $P[3:0] = '0010'$ ) and MULTI = '1'. It configures the version for IGMP messages (Join/Leave/Report).

### Multicast Blocking in MACRAW mode

0 : disable Multicast Blocking

1 : enable Multicast Blocking

This bit is applied only when MACRAW mode( $P[3:0] = '0100'$ ). It blocks to receive the packet with multicast MAC address.

### [6] Broadcast Blocking in MACRAW and UDP mode

0 : disable Broadcast Blocking

1 : enable Broadcast Blocking

This bit blocks to receive broadcasting packet during UDP mode( $P[3:0] = '0010'$ ). In addition, This bit does when MACRAW mode( $P[3:0] = '0100'$ ) Data packet is received from a peer. When this bit is '0', It sends the ACK packet after waiting for the timeout time configured by RTR.

---

### [7] Multicasting and MAC Filter Enable mode

#### Multicasting in UDP mode

0 : disable Multicasting

1 : enable Multicasting

This bit is applied only during UDP mode( $P[3:0] = '0010'$ ). To use multicasting, Sn\_DIPR & Sn\_DPORT should be respectively configured with the multicast group IP address & port number before Socket n is opened by OPEN command of Sn\_CR

#### MAC Filter Enable in MACRAW mode

0 : disable MAC Filtering

1 : enable MAC Filtering

This bit is applied only during MACRAW mode( $P[3:0] = '0100'$ ). When set as '1', WZTOE can only receive broadcasting packet or packet sent to itself. When this bit is '0', WZTOE can receive all packets on Ethernet. If user wants to implement Hybrid TCP/IP stack, it is recommended that this bit is set as '1' for reducing host overhead to process the all received packets.

## Sn\_CR (Socket n Command Register)

Address Offset : 0x0010

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res								Sn_CR[7:0]							
								R/W							

This is used to set the command for Socket n such as OPEN, CLOSE, CONNECT, LISTEN, SEND, and RECEIVE. After WZTOE accepts the command, the Sn\_CR register is automatically cleared to 0x00. Even though Sn\_CR is cleared to 0x00, the command is still being processed. To check whether the command is completed or not, please check the Sn\_IR or Sn\_SR.

Value	Symbol	Description										
0x01	OPEN	<p>Socket n is initialized and opened according to the protocol selected in Sn_MR (P3:P0). The table below shows the value of Sn_SR corresponding to Sn_MR.</p> <table border="1"> <thead> <tr> <th>Sn_MR (P[3:0])</th><th>Sn_SR</th></tr> </thead> <tbody> <tr> <td>Sn_MR_CLOSE ('0000')</td><td>-</td></tr> <tr> <td>Sn_MR_TCP ('0001')</td><td>SOCK_INIT (0x13)</td></tr> <tr> <td>Sn_MR_UDP ('0010')</td><td>SOCK_UDP (0x22)</td></tr> <tr> <td>SO_MR_MACRAW ('0100')</td><td>SOCK_MACRAW (0x02)</td></tr> </tbody> </table>	Sn_MR (P[3:0])	Sn_SR	Sn_MR_CLOSE ('0000')	-	Sn_MR_TCP ('0001')	SOCK_INIT (0x13)	Sn_MR_UDP ('0010')	SOCK_UDP (0x22)	SO_MR_MACRAW ('0100')	SOCK_MACRAW (0x02)
Sn_MR (P[3:0])	Sn_SR											
Sn_MR_CLOSE ('0000')	-											
Sn_MR_TCP ('0001')	SOCK_INIT (0x13)											
Sn_MR_UDP ('0010')	SOCK_UDP (0x22)											
SO_MR_MACRAW ('0100')	SOCK_MACRAW (0x02)											
0x02	LISTEN	<p>This is valid only in TCP mode (Sn_MR(P3:P0) = Sn_MR_TCP). In this mode, Socket n operates as a ‘TCP server’ and waits for connection-request (SYN packet) from any ‘TCP client’.</p> <p>The Sn_SR changes the state from SOCK_INIT to SOCKET_LISTEN. When a ‘TCP client’ connection request is successfully established, the Sn_SR changes from SOCKET_LISTEN to SOCK_ESTABLISHED and the Sn_IR(0) becomes ‘1’. But when a ‘TCP client’ connection request is failed, Sn_IR(3) becomes ‘1’ and the status of Sn_SR changes to SOCK_CLOSED.</p>										
0x04	CONNECT	<p>This is valid only in TCP mode and operates when Socket n acts as ‘TCP client’. To connect, a connect-request (SYN packet) is sent</p>										

		<p>to ‘TCP server’ configured by Sn_DIPR &amp; Sn_DPORT(destination address &amp; port). If the connect-request is successful, the Sn_SR is changed to SOCK_ESTABLISHED and the Sn_IR(0) becomes ‘1’.</p> <p>The connect-request fails in the following three cases.</p> <ol style="list-style-type: none"> <li>1. When a ARP<sub>TO</sub> occurs (Sn_IR(3)=‘1’) because the destination hardware address is not acquired through the ARP-process.</li> <li>2. When a SYN/ACK packet is not received and TCP<sub>TO</sub> (Sn_IR(3) = ‘1’)</li> <li>3. When a RST packet is received instead of a SYN/ACK packet.</li> </ol> <p>In these cases, Sn_SR is changed to SOCK_CLOSED.</p>
0x08	DISCON	<p>Valid only in TCP mode.</p> <p>Regardless of ‘TCP server’ or ‘TCP client’, the DISCON command processes the disconnect-process (‘Active close’ or ‘Passive close’).</p> <p><b>Active close:</b> it transmits disconnect-request(FIN packet) to the connected peer</p> <p><b>Passive close:</b> When FIN packet is received from peer, a FIN packet is replied back to the peer.</p> <p>When the disconnect-process is successful (that is, FIN/ACK packet is received successfully), Sn_SR is changed to SOCK_CLOSED. Otherwise, TCP<sub>TO</sub> occurs (Sn_IR(3)=‘1’= and then Sn_SR is changed to SOCK_CLOSED.</p> <p>cf&gt; If CLOSE is used instead of DISCON, only Sn_SR is changed to SOCK_CLOSED without disconnect-process.</p> <p>If a RST packet is received from a peer during communication, Sn_SR is unconditionally changed to SOCK_CLOSED.</p>
0x10	CLOSE	<p>Close Socket n.</p> <p>Sn_SR is changed to SOCK_CLOSED.</p>
0x20	SEND	<p>SEND transmits all the data in the Socket n TX buffer. For more details, please refer to Socket n TX Free Size Register (Sn_TX_FSR), Socket n, TX Write Pointer Register (Sn_TX_WR), and Socket n TX Read Pointer Register(Sn_TX_RD).</p>
0x21	SEND_MAC	<p>Valid only in UDP mode.</p> <p>The basic operation is same as SEND. Normally SEND transmits data after destination hardware address is acquired by the</p>

---

		automatic ARP-process(Address Resolution Protocol). But SEND_MAC transmits data without the automatic ARP-process. In this case, the destination hardware address is acquired from Sn_DHAR configured by host, instead of APR-process.
0x22	SEND_KEEP	Valid only in TCP mode. It checks the connection status by sending 1byte keep-alive packet. If the peer cannot respond to the keep-alive packet during timeout time, the connection is terminated and the timeout interrupt will occur.
0x40	RECV	RECV completes the processing of the received data in Socket n RX Buffer by using a RX read pointer register (Sn_RX_RD). For more details, refer to Socket n RX Received Size Register (Sn_RX_RSR), Socket n RX Write Pointer Register (Sn_RX_WR), and Socket n RX Read Pointer Register (Sn_RX_RD).

---

## Sn\_IR (Socket n Interrupt Register)

Address Offset : 0x0020

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															

Sn\_IR indicates the status of Socket Interrupt such as establishment, termination, receiving data, timeout). When an interrupt occurs and the corresponding bit of Sn\_IMR is ‘1’, the corresponding bit of Sn\_IR becomes ‘1’.

- [0] **CONNECT Interrupt** - This is issued one time when the connection with peer is successful and then Sn\_SR is changed to SOCK\_ESTABLISHED
- [1] **DISCONNECT Interrupt** - This is issued when FIN or FIN/ACK packet is received.
- [2] **RECV Interrupt** - This is issued whenever data is received from a peer.
- [3] **TIMEOUT Interrupt** - This is issued when ARP<sub>TO</sub> or TCP<sub>TO</sub> occurs.
- [4] **SENDOK Interrupt** - This is issued when SEND command is completed

## Sn\_IMR (Socket n Interrupt Mask Register)

Address Offset : 0x0024

Reset value: 0x0000\_00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res				Sn_IMR[4:0]											
												R/W	R/W	R/W	R/W

Sn\_IMR is used to mask interrupts. Each bit of Sn\_IMR corresponds to each bit of Sn\_IR. When a bit of Sn\_IMR is ‘1’ and the corresponding bit of Sn\_IR is ‘1’, an interrupt will be issued. In other words, if a bit of Sn\_IMR is ‘0’, an interrupt will not be issued even if the corresponding bit of Sn\_IR is ‘1’.

#### [0] CONNECT Interrupt Mask

0: Disable CONNECT Interrupt

1: Enable CONNECT Interrupt

#### [1] DISCONNECT Interrupt Mask

0: Disable DISCONNECT Interrupt

1: Enable DISCONNECT Interrupt

#### [2] RECV Interrupt Mask

0: Disable RECV Interrupt

1: Enable RECV Interrupt

#### [3] TIMEOUT Interrupt Mask

0: Disable TIMEOUT Interrupt

1: Enable TIMEOUT Interrupt

#### [4] SENDOK Interrupt Mask

0: Disable SENDOK Interrupt

1: Enable SENDOK Interrupt

## Sn\_ICR (Socket n Interrupt Clear Register)

Address Offset : 0x0028

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res								Sn_ICR[4:0]							

									R/W						
--	--	--	--	--	--	--	--	--	-----	-----	-----	-----	-----	-----	-----

Sn\_ICR is used to clear interrupts. Each bit of Sn\_IR can be cleared when the host writes ‘1’ value to each bit of Sn\_ICR corresponding to each bit of Sn\_IR.

- [0] CONNECT Interrupt Clear
- [1] DISCONNECT Interrupt Clear
- [2] RECV Interrupt Mask
- [3] TIMEOUT Interrupt Mask
- [4] SENDOK Interrupt Mask

\* *ReadClearWrite1 (R/C\_W1) : Software can read as well as clear this bit by writing*

1. *Writing ‘0’ has no effect on the bit value.*

## Sn\_SR (Socket n Status Register)

Address Offset : 0x0030

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	Res	Res	res	res	res	res	res	res	res	res	res	res	res	res	res
<hr/>															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	Res	Res	res	res	res	res	res		R/W						
Sn_SR[7:0]															

Sn\_SR indicates the status of Socket n. The status of Socket n is changed by Sn\_CR or some special control packet as SYN, FIN packet in TCP.

Value	Symbol	Description
0x00	SOCK_CLOSED	This indicates that Socket n is released. When DISCON, CLOSE command is ordered, or when a timeout occurs, it is changed to SOCK_CLOSED regardless of previous status.
0x13	SOCK_INIT	This indicates Socket n is opened with TCP mode. It is changed to SOCK_INIT when Sn_MR (P[3:0]) = ‘0001’ and OPEN command is ordered. After SOCK_INIT, user can use LISTEN /CONNECT command.

0x14	SOCK_LISTEN	<p>This indicates Socket n is operating as ‘TCP server’ mode and waiting for connection-request (SYN packet) from a peer (‘TCP client’).</p> <p>It will change to SOCK_ESTABLISHED when the connection-request is successfully accepted.</p> <p>Otherwise it will change to SOCK_CLOSED after TCPTO occurred (Sn_IR(TIMEOUT) = ‘1’).</p>
0x17	SOCK_ESTABLISHED	<p>This indicates the status of the connection of Socket n.</p> <p>It changes to SOCK_ESTABLISHED when the ‘TCP SERVER’ processed the SYN packet from the ‘TCP CLIENT’ during SOCK_LISTEN, or when the CONNECT command is successful.</p> <p>During SOCK_ESTABLISHED, DATA packet can be transferred using SEND or RECV command.</p>
0x1C	SOCK_CLOSE_WAIT	<p>This indicates Socket n received the disconnect-request (FIN packet) from the connected peer. This is half-closing status, and data can be transferred. For full-closing, DISCON command is used. But For just-closing, CLOSE command is used.</p>
0x22	SOCK_UDP	<p>This indicates Socket n is opened in UDP mode(Sn_MR(P[3:0]) = ‘0010’).</p> <p>It changes to SOCK_UDP when Sn_MR(P[3:0]) = ‘0010’ and OPEN command is ordered.</p> <p>Unlike TCP mode, data can be transferred without the connection-process.</p>
0x42	SOCK_MACRAW	<p>This indicates Socket 0 is opened in MACRAW mode (S0_MR(P[3:0]) = ‘0100’) and is valid only in Socket 0.</p> <p>It changes to SOCK_MACRAW when S0_MR(P[3:0]) = ‘0100’ and OPEN command is ordered.</p> <p>Like UDP mode socket, MACRAW mode Socket 0 can transfer a MAC packet (Ethernet frame) without the connection-process.</p>

The following table shows a temporary status indicated during changing the status of Socket n.

Value	Symbol	Description
0x15	SOCK_SYNSENT	This indicates Socket n sent the connect-request packet (SYN packet) to a peer.

		<p>It is temporarily shown when Sn_SR is changed from SOCK_INIT to SOCK_ESTABLISHED by CONNECT command.</p> <p>If connect-accept(SYN/ACK packet) is received from the peer at SOCK_SYNSENT, it changes to SOCK_ESTABLISHED.</p> <p>Otherwise, it changes to SOCK_CLOSED after TCPTO (Sn_IR[TIMEOUT] = '1') is occurred.</p>
0x16	SOCK_SYNRECV	<p>It indicates Socket n successfully received the connect-request packet (SYN packet) from a peer.</p> <p>If socket n sends the response (SYN/ACK packet) to the peer successfully, it changes to SOCK_ESTABLISHED. If not, it changes to SOCK_CLOSED after timeout occurs (Sn_IR[TIMEOUT] = '1').</p>
0x18	SOCK_FIN_WAIT	<p>These indicate Socket n is closing.</p>
0x1A	SOCK_CLOSING	<p>These are shown in disconnect-process such as active-close and passive-close.</p>
0X1B	SOCK_TIME_WAIT	<p>When Disconnect-process is successfully completed, or when timeout occurs, these change to SOCK_CLOSED.</p>
0X1D	SOCK_LAST_ACK	<p>This indicates Socket n is waiting for the response (FIN/ACK packet) to the disconnect-request (FIN packet) by passive-close.</p> <p>It changes to SOCK_CLOSED when Socket n received the response successfully, or when timeout occurs (Sn_IR[TIMEOUT] = '1').</p>

## Sn\_PNR (Socket n Protocol Number Register)

Address Offset : 0x0100

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res		R/W													

This IP Protocol Register is user to set up the Protocol Field of IP Header at the IP layer RAW mode. There are several protocol number defined in advance by registering to IANA. For the

overall list of upper level protocol identification number that IP is using, refer to online documents of IANA (<http://www.iana.org/assignments/protocol-numbers>).

Ex) Internet Control Message Protocol (ICMP) = 0x01, Internet Group Management Protocol = 0x02

### Sn\_TOSR (Socket n IP Type of Service Register)

Address Offset : 0x0104

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res								Sn_ROS[7:0]							

Sn\_TOSR configures the TOS(Type Of Service field in IP Header) of Socket n. It is set before OPEN command.

For more the details, refer to <http://www.iana.org/assignments/ip-parameters>.

### Sn\_TTLR (Socket n TTL Register)

Address Offset : 0x0108

Reset value : 0x0000\_0080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res								Sn_TTL[7:0]							

Sn\_TTL configures the TTL(Time To Live field in IP Header) of Socket n. It is set before OPEN command.

For more the details, refer to <http://www.iana.org/assignments/ip-parameters>.

## Sn\_FRAGR (Socket n Fragment offset Register)

Address Offset : 0x010C

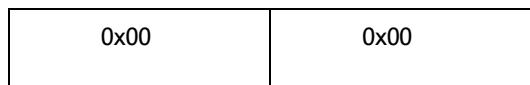
Reset value : 0x0000\_4000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	Res	Res	res	res	res	res	res	res	res	res	res	res	res	res	res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sn_FRAG[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[15:0] Sn\_FRAG configures the FRAG(Fragment field in IP header)

Ex) Sn\_FRAGR = 0x0000 (Don't Fragment)

Sn\_FRAG[15:8]                    Sn\_FRAG[7:0]



## Sn\_MSSR (Socket n Maximum Segment Register)

Address Offset : 0x0110

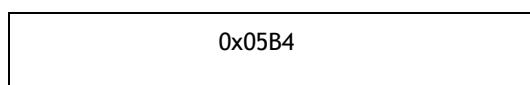
Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	Res	Res	res	res	res	res	res	res	Res	res	res	res	res	res	res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sn_MSS[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

This register is used for MSS (Maximum Segment Size) of TCP, and the register displays MSS set by the other party when TCP is activated in Passive Mode.

Ex) In case of Socket 0 MSS = 1460 (0x05B4), configure as below,

0x4101\_0110



## Sn\_PORTR (Socket n Source Port Register)

Address Offset : 0x0114

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	Res	Res	res	res	res	res	res	res	res	res	res	res	res	res	res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sn_PORT[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sn\_PORTR configures the source port number of Socket n. It is valid when Socket n is used in TCP/UDP mode. It should be set before OPEN command is ordered.

Ex) In case of Socket 0 Port = 5000(0x1388), configure as below,

0x4101\_0114

0x1388

## Sn\_DHAR (Socket n Destination Hardware address Register)

Address Offset : 0x0118

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Sn_DHAR0[31:24]								Sn_DHAR0 [23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sn_DHAR0 [15:8]								Sn_DHAR0 [7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address Offset : 0x011E

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
	Sn_DHAR1 [31:24]								Sn_DHAR1 [23:16]							
R	R	R	R	R		R	R	R	R	R	R	R	R	R	R	

Address Offset : 0x011C

Reset value : 0x0000\_0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sn_DHAR1 [15:8]								Sn_DHAR1 [7:0]							
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Sn\_DHAR configures the destination hardware address of Socket n when using SEND\_MAC command in UDP mode or it indicates that it is acquired in ARP-process by CONNECT/SEND command.

**Ex) In case of writing MAC address “00.08.DC.12.34.56”**

Sn_DHAR0 [31:24]	Sn_DHAR0 [23:16]	Sn_DHAR0 [15:8]	Sn_DHAR0 [7:0]	Sn_DHAR1 [15:8]	Sn_DHAR1 [7:0]
0x00	0x08	0xDC	0x12	0x34	0x56

**Ex) In case of reading MAC address “00.08.DC.12.34.56”**

Sn_DHAR0 [31:24]	Sn_DHAR0 [23:16]	Sn_DHAR0 [15:8]	Sn_DHAR0 [7:0]	Sn_DHAR1 [31:24]	Sn_DHAR1 [23:16]
0x00	0x08	0xDC	0x12	0x34	0x56

## Sn\_DPORTR (Socket n Destination Port Number Register)

Address Offset : 0x0120

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sn_DPROT[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sn\_DPORT configures or indicates the destination port number of Socket n. It is valid when Socket n is used in TCP/UDP mode.

In TCP client mode, it configures the listen port number of ‘TCP server’ before CONNECT command.

In TCP server mode, it indicates the port number of ‘TCP client’ after successfully establishing connection.

In UDP mode, it configures the port number of peer to be transmitted the UDP packet by SEND/SEND\_MAC command.

**Ex)** In case of Socket 0 Destination Port = 5000(0x1388), configure as below,

0x4101\_0120

0x1388
--------

### Sn\_DIPR (Socket n Destination IP address Register)

Address Offset : 0x0124

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Sn_DIPR[31:24]								Sn_DIPR [23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sn_DIPR [15:8]								Sn_DIPR [7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sn\_DIPR configures or indicates the destination IP address of Socket n. It is valid when Socket n is used in TCP/UDP mode.

In TCP client mode, it configures an IP address of ‘TCP server’ before CONNECT command.

In TCP server mode, it indicates an IP address of ‘TCP client’ after successfully establishing connection.

In UDP mode, it configures an IP address of peer to be received the UDP packet by SEND or SEND\_MAC command.

**Ex)** In case of “192.168.0.2”

Sn_DIPR[31:24]	Sn_DIPR [23:16]	Sn_DIPR [15:8]	Sn_DIPR [7:0]
----------------	-----------------	----------------	---------------

192 (0xC0)	168 (0xA8)	0 (0x00)	2 (0x02)
------------	------------	----------	----------

## Sn\_KATMR (Socket n Keep Alive Timer Register)

Address Offset : 0x0180

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															
									R/W						

Sn\_KPALVTR configures the transmitting timer of 'KEEP ALIVE(KA)' packet of SOCKETn. It is valid only in TCP mode, and ignored in other modes. The time unit is 5s(Refer to TCKCNTR).

KA packet is transmittable after Sn\_SR is changed to SOCK\_ESTABLISHED and after the data is transmitted to a peer at least once. In case of 'Sn\_KPALVTR > 0', WZTOE automatically transmits KA packet after time-period for checking the TCP connection (Auto-keepalive-process). In case of 'Sn\_KPALVTR = 0', Auto-keep-alive-process will not operate, and KA packet can be transmitted by SEND\_KEEP command by the host (Manual-keep-alive-process). Manual-keep-alive-process is ignored in case of 'Sn\_KPALVTR > 0'.

**Ex) Sn\_KPALVTR = 10 (Keep Alive packet will be transmitted every 50 seconds.)**

Sn\_KATM[7:0]

0x0A
------

## Sn\_RTR (Socket n Retry Time Register)

Address Offset : 0x0184

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

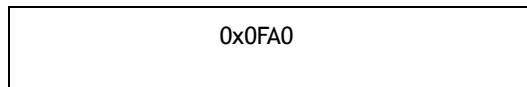
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sn_RTR[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sn\_RTR configures the retransmission timeout period of Socket n. When Retry Counter Register(RTR) is zero, Sn\_RTR is valid. The unit of timeout period is 100us(Refer to TCKCNTR).

During the time configured by Sn\_RTR, WZTOE waits for the peer response to the packet that is transmitted by Sn\_CR(CONNECT, DISCON, CLOSE, SEND, SEND\_MAC, SEND\_KEEP command). If the peer does not respond within the RTR time, WZTOE retransmits the packet or issues timeout.

**Ex)** When timeout-period is set as 400ms, RTR = (400ms / 1ms) X 10 = 4000(0x0FA0)

RTR[15:0]



### Sn\_RCR (Socket n Retry Counter Register)

Address Offset : 0x0188

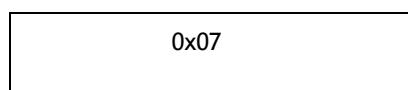
Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res								Res							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res								Sn_RC[7:0]							
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sn\_RCR configures the number of time of retransmission of Socket n. When Retry Counter Register(RCR) is zero, Sn\_RCR is valid. When retransmission occurs as many as ‘Sn\_RCR+1’, Timeout interrupt is issued (Sn\_IR[TIMEOUT] = ‘1’).

**Ex)** Sn\_RCR = 0x0007

Sn\_RC[7:0]



### Sn\_TXBUF\_SIZE (Socket n TX Buffer Size Register)

Address Offset : 0x0200

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	Sn_TMS[7:0]														
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sn\_TXBUF\_SIZE configures the TX Buffer size of Socket n. Socket n TX Buffer size can be configured with 0,1,2,4,8 and 16 Kbytes. If a different size is configured, the data can't be normally transmitted to a peer.

Although Socket n TX Buffer size is initially configured to 2Kbytes, user can be re-configure its size using Sn\_TXBUF\_SIZE. The total sum of Sn\_TXBUF\_SIZE cannot be exceed 16Kbytes. When exceeded, the data transmission error is occurred.

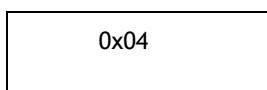
When all Sn\_TXBUF\_SIZE have been configured, Socket n TX Buffer is allocated with the configured size in 16KB TX Buffer and is assigned sequentially from Socket 0 to Socket 7.

Socket n TX Buffer can be accessible with 16bits Offset Address ranged from 0x0000 to 0xFFFF regardless of the configured size. (Refer to Sn\_TX\_WR & Sn\_TX\_RD).

Value (dec)	0	1	2	4	8	16
Buffer size	0KB	1KB	2KB	4KB	8KB	16KB

**Ex) Socket 0 TX Buffer Size = 4KB**

0x4600\_0200



### Sn\_TX\_FSR (Socket n TX Free Size Register)

Address Offset : 0x0204

Reset value : 0x0000\_0800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res		R/W													

Sn\_TX\_FSR indicates the free size of Socket n TX Buffer. It is initialized to the configured size by Sn\_TX\_SIZE. Data bigger than Sn\_TX\_SIZE should not be saved in the Socket n TX Buffer

because the bigger data overwrites the previous saved data not yet sent. Therefore, check before saving the data to the Socket n TX Buffer, and if data is equal or smaller than its checked size, transmit the data with SEND/SEND\_MAC command after saving the data in Socket n TX Buffer. But, if data is bigger than its checked size, transmit the data after dividing into the checked size and saving in the Socket n TX Buffer.

If Sn\_MR(P[3:0]) is not TCP mode('0001'), it is automatically calculated as the difference between 'Socket n TX Write Pointer (Sn\_TX\_WR)' and 'Socket n TX Read Pointer (Sn\_TX\_RD)'. If Sn\_MR(P[3:0]) is TCP mode('0001'), it is automatically calculated as the difference between Sn\_TX\_WR and the internal ACK pointer which indicates the point of data is received already by the connected peer.

**Ex)** In case of 2048(0x0800) in S0\_TX\_FSR,

0x4600\_0204

0x0800

**Note)** Because this register for representing the size information is 16 bits, it is impossible to read all bytes at the same time. Before 16 bit-read operation is not completed, the value may be changed.

Therefore, it is recommended that you read all 16-bits twice or more until getting the same value.

### Sn\_TX\_RD (Socket n TX Read Pointer Register)

Address Offset : 0x0208

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	res	res	res	res	res	res	res	res	res	res	res	res	res
Sn_TXRP[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sn\_TX\_RD is initialized by OPEN command. However, if Sn\_MR(P[3:0]) is TCP mode('0001'), it is re-initialized while connecting with TCP.

After its initialization, it is auto-increased by SEND command. SEND command transmits the saved data from the current Sn\_TX\_RD to the Sn\_TX\_WR in the Socket n TX Memory. After transmitting the saved data, the SEND command increases the Sn\_TX\_RD as same as the Sn\_TX\_WR. If its increment value exceeds the maximum value 0xFFFF, (greater than 0x10000 and the carry bit occurs), then the carry bit is ignored and will automatically update with the lower 16bits value.

### Sn\_TX\_WR (Socket n TX Write Pointer Register)

Address Offset : 0x020C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	res	res	res	res	res	res	res	res	res	res	res	res	res
Sn_TXWP[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sn\_TX\_WR is initialized by OPEN command. However, if Sn\_MR(P[3:0]) is TCP mode('0001'), it is re-initialized while connecting with TCP.

It should be read or to be updated like as follows.

1. Read the starting address for saving the transmitting data.
2. Save the transmitting data from the starting address of Socket n TX buffer.
3. After saving the transmitting data, update Sn\_TX\_WR to the increased value as many as transmitting data size. If the increment value exceeds the maximum value 0xFFFF(greater than 0x10000 and the carry bit occurs), then the carry bit is ignored and will automatically update with the lower 16bits value.
4. Transmit the saved data in Socket n TX Buffer by using SEND/SEND command

### Sn\_RXBUF\_SIZE (Socket n RX Buffer Size Register)

Address Offset : 0x0220

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	res	res	res	res	res	res	res	res	res	res	res	res	res
Sn_RXBUFSIZE[15:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Res	Sn_RXMS[15:0]														
								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sn\_RXBUF\_SIZE configures the RX Buffer size of Socket n. Socket n RX Buffer size can be configured with 1,2,4,8, and 16 Kbytes. If a different size is configured, the data cannot be normally received from a peer.

Although Socket n RX Buffer size is initially configured to 2Kbytes, user can re-configure its size using Sn\_RXBUF\_SIZE. The total sum of Sn\_RX\_SIZE cannot be exceed 16Kbytes. When exceeded, the data reception error is occurred.

When all Sn\_RXBUF\_SIZE have been configured, Socket n RX Memory is allocated with the configured size in 16KB RX Memory and is assigned sequentially from Socket 0 to Socket 7.

Socket n RX Buffer Block can be accessible with the 16bits Offset Address ranged from 0x0000 to 0xFFFF regardless of the configured size. (Refer to Sn\_RX\_RD & Sn\_RX\_WR).

Value (dec)	0	1	2	4	8	16
Buffer size	0KB	1KB	2KB	4KB	8KB	16KB

**Ex) Socket 0 RX Buffer Size = 8KB**

0x4101\_0220

0x08

## Sn\_RX\_RSR (Socket n RX Received Size Register)

Address Offset : 0x0224

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	res	res	res	res	res	res	res	res	res	res	res	res	res
Sn_RX_RSR[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sn\_RX\_RSR indicates the data size received and saved in Socket n RX Buffer. Sn\_RX\_RSR does not exceed the Sn\_RXBUF\_SIZE and is calculated as the difference between ‘Socket n RX Write Pointer (Sn\_RX\_WR)’ and ‘Socket n RX Read Pointer (Sn\_RX\_PD)’.

**Ex)** In case of 2048(0x0800) in S0\_RXSR,

0x4101\_0224

0x0800
--------

**Note)** Because this register for representing the size information is 16 bits, it is impossible to read all bytes at the same time. Before 16 bit-read operation is not completed, the value may be changed.

Therefore, it is recommended that you read all 16-bits twice or more until getting the same value.

### Sn\_RX\_RD (Socket n RX Read Pointer Register)

Address Offset : 0x0228

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	res	res	res	res	res	res	res	res	res	res	res	res	res
Sn_RXRP[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sn\_RX\_RD is initialized by OPEN command. Make sure to be read or updated as follows.

1. Read the starting save address of the received data
2. Read data from the starting address of Socket n RX Buffer.
3. After reading the received data, Update Sn\_RX\_RD to the increased value as many as the reading size. If the increment value exceeds the maximum value 0xFFFF, that is, is greater than 0x10000 and the carry bit occurs, update with the lower 16bits value ignored the carry bit.
4. Order RECV command is for notifying the updated Sn\_RX\_RD to TOE.

**Ex)** In case of 2048(0x0800) in S0\_RX\_RD,

0x4101\_0228

0x08
------

## Sn\_RX\_WR (Socket n RX Write Pointer Register)

Address Offset : 0x022C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Sn_RX_WR[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Sn\_RX\_WR is initialized by OPEN command and it is auto-increased by the data reception.

If the increased value exceeds the maximum value 0xFFFF, (greater than 0x10000 and the carry bit occurs), then the carry bit is ignored and will automatically update with the lower 16bits value.

**Ex)** In case of 2048(0x0800) in S0\_RX\_WR,

0x4101\_022C

0x0800
--------

## 14 Random number generator (RNG)

### 14.1 Introduction

RNG is a 32bit random number generator. RNG generates power on random number when power on reset. RNG can run/stop by software. RNG seed value and polynomial of RNG can be modified by software.

### 14.2 Features

- 32bit pseudo random number generator
- Formula of pseudo random number generator (polynomial) can be modified.
- Seed value of random generator can be modified.
- Support power on reset random value.
- Random value can be obtained by control start/stop by software.

### 14.3 Functional description

Figure 8 shows the RNG block diagram.

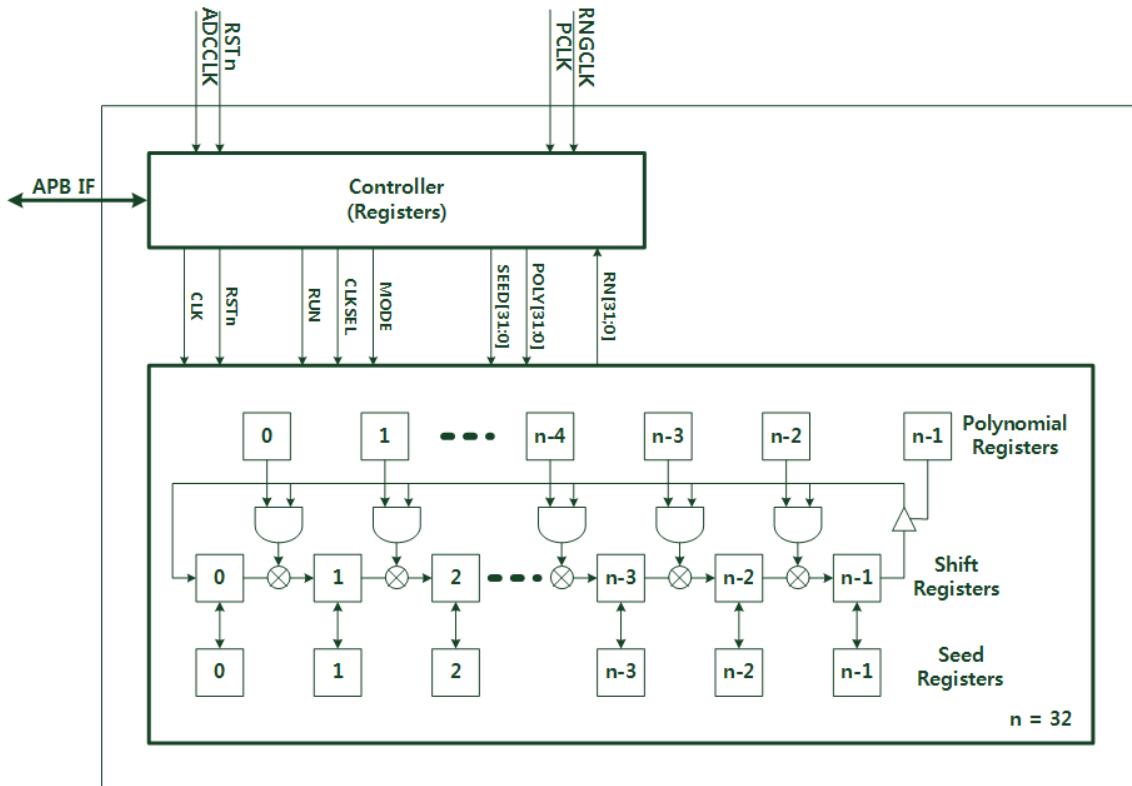


Figure 8. Random Number Generator block diagram

## Operation RNG

Figure 9 shows the flowchart of RNG operation.

A random number is automatically generated after powering on reset,

Follow the procedure below to manually generate a random number.

1. Change MODE to start/stop by register.
2. Change clock source / seed value / polynomial value if need.
3. Run and Stop the RNG.
4. Read Random value.

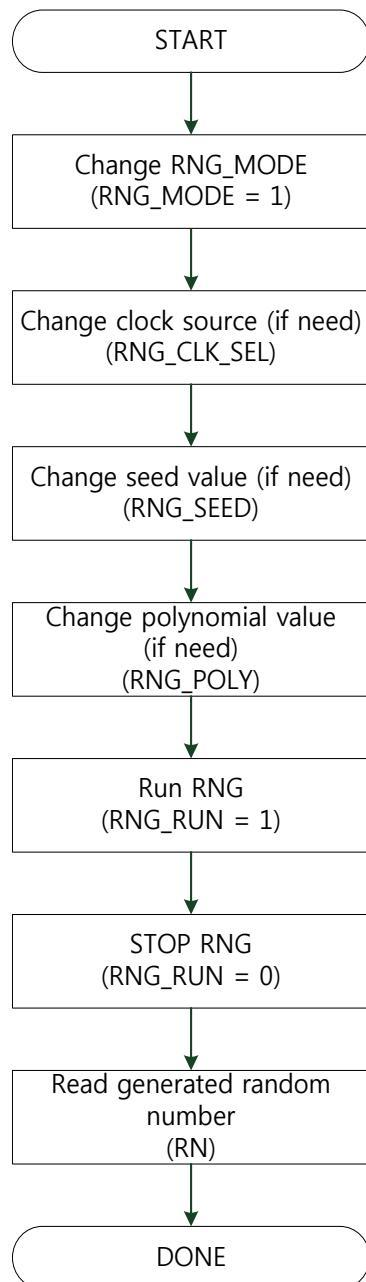


Figure 9. Flow chart of RNG operation

## 14.4 Registers (Base address : 0x4000\_7000)

### RNG run register (RNG\_RUN)

Address offset : 0x000

Reset value : 0x0000\_0000

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

[0] RUN - run RNG shift register

This bit written by S/W to run or stop RNG

0 : stop random number generator shift register

1 : run random number generator shift register

### RNG SEED register (RNG\_SEED)

Address offset : 0x004

Reset value : 0x0000\_0000

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

[31:0] SEED - seed value of random number generator shift register

These bits written by S/W to set seed value of RNG before start(run) RNG shift register

### RNG clock select register (RNG\_CLKSEL)

Address offset : 0x008

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

res																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
res	CLKSEL	R/W														

[0] CLKSEL - select clock source register of RNG shift register

This bit written by S/W to select clock source of RNG shift register

0 : RNG clock (refer to clock generator block)

1 : PCLK

### RNG manual mode select register (RNG\_MODE)

Address offset : 0x00c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
res																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
res	MODE	R/W															

[0] MODE - RNG run mode select register

This bit written by S/W to select which mode

0 : run/stop by PLL\_LOCK signal (which is for power on random number)

1 : run/stop by RNG\_RUN register (refer 1.4.1)R

### RNG random number value register (RNG\_RN)

Address offset : 0x010

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
RN[31:16]																
R																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RN[15:0]															
R															

[31:0] RN - random number of RNG shift register

These bits are read only registers.

### RNG polynomial register (RNG\_POLY)

Address offset : 0x014

Reset value : 0xE000\_0202

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
POLY[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLY[15:0]															
R/W															

[31:0] POLY - 32bit polynomial of random number generator

These bits are written by S/W to modify the formula of random number generator

Default polynomial:  $F(x) = x^{31} + x^{30} + x^{29} + x^9 + x$

## 14.5 Register map

The following Table 8 summarizes the RNG registers.

Table 8 RNG register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	RUN	0
0x000	RNG_RUN	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	0			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x004	RNG_SEED	SEED																																	
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x008	RNG_CLKSEL	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	0			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x00C	RNG_MODE	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	0			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x010	RNG_RN	RN																																	
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x014	RNG_POLY	POLY																																	
	reset value	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

## 15 Alternate Function Controller (AFC)

### 15.1 Introduction

Each functional PADS have several functions.

Users can select a function in Alternate Function Controller block.

### 15.2 Features

Each functional pad has 2 ~ 4 functions.

Pads can be selected by each registers individually.

Each pad can be used as an external interrupt source.

### 15.3 Functional description

Table 9 shows the function table of each functional pad.

Table 9 functional description table

		function selection register value				
		00 (reset value)	01	10	11	
PAD	Chip	PIN	Normal Function	2nd Function	3rd Function	4th Function

PA_00	W7500	29	GPIOA_0		PWM6/CAP6	
	W7500P	27				
PA_01	W7500	30	GPIOA_1		PWM7/CAP7	
	W7500P	28				
PA_02	W7500	31	GPIOA_2		CLKOUT	
	W7500P	29				
PA_03	W7500	49	SWCLK			PWM0/CAP0
	W7500P	30				
PA_04	W7500	50	SWDIO			PWM1/CAP1
	W7500P	31				
PA_05	W7500	33	SSEL0		SCL1	PWM2/CAP2
	W7500P	19				
PA_06	W7500	34	SCLK0		SDA1	PWM3/CAP3
	W7500P	20				
PA_07	W7500	35	MISO0		U_CTS1	PWM4/CAP4
	W7500P	21				
PA_08	W7500	36	MOSI0		U_RTS1	PWM5/CAP5
	W7500P	22				
PA_09	W7500	37	SCL0		U_TXD1	PWM6/CAP6
	W7500P	23				
PA_10	W7500	38	SDA0		U_RXD1	PWM7/CAP7
	W7500P	24				
PA_11	W7500	40	U_CTS0		SSEL1	
	W7500P	41				
PA_12	W7500	41	U_RTS0		SCLK1	
	W7500P	42				
PA_13	W7500	42	U_TXD0		MISO1	
	W7500P	43				
PA_14	W7500	43	U_RXD0		MOSI1	
	W7500P	44				
PA_15	W7500	44	GPIOA_15	GPIOA_15		
PB_00	W7500	45	SSEL1	GPIOB_0	U_CTS0	
	W7500P	45				
PB_01	W7500	46	SCLK1	GPIOB_1	U_RTS0	
	W7500P	46				
PB_02	W7500	47	MISO1	GPIOB_2	U_TXD0	
	W7500P	47				
	W7500	48				

PB_03	W7500P	48	MOSI1	GPIOB_3	U_RXD0	
PB_04	W7500	24	TXEN	GPIOB_4		
PB_05	W7500	25	COL	GPIOB_5		
PB_06	W7500	16	RXD3	GPIOB_6		
	W7500P	15	DUP			
PB_07	W7500	17	RXCLK	GPIOB_7		
PB_08	W7500	18	DUP	GPIOB_8		
PB_09	W7500	19	TXCLK	GPIOB_9		
PB_10	W7500	20	TXD0	GPIOB_10		
PB_11	W7500	21	TXD1	GPIOB_11		
PB_12	W7500	22	TXD2	GPIOB_12		
PB_13	W7500	23	TXD3	GPIOB_13		
PB_14	W7500	26	GPIOB_14	GPIOB_14		
PB_15	W7500	27	GPIOB_15	GPIOB_15		
PC_00	W7500	53	U_CTS1	GPIOC_0	PWM0/CAP0	
	W7500P					
PC_01	W7500	54	U_RTS1	GPIOC_1	PWM1/CAP1	
	W7500P					
PC_02	W7500	55	U_TXD1	GPIOC_2	PWM2/CAP2	
	W7500P					
PC_03	W7500	56	U_RXD1	GPIOC_3	PWM3/CAP3	
	W7500P					
PC_04	W7500	57	SCL1	GPIOC_4	PWM4/CAP4	
	W7500P					
PC_05	W7500	58	SDA1	GPIOC_5	PWM5/CAP5	
	W7500P					
PC_06	W7500	51	GPIOC_6	GPIOC_6	U_TXD2	
	W7500P	11				
PC_07	W7500	52	GPIOC_7	GPIOC_7	U_RXD2	
PC_08	W7500	1	PWM0/CAP0	GPIOC_8	SCL0	AIN7
	W7500P					
PC_09	W7500	2	PWM1/CAP1	GPIOC_9	SDA0	AIN6
	W7500P					
PC_10	W7500	3	U_TXD2	GPIOC_10	PWM2/CAP2	AIN5
	W7500P					
PC_11	W7500	4	U_RXD2	GPIOC_11	PWM3/CAP3	AIN4
	W7500P					
	W7500					

PC_12	W7500P	5	AIN3	GPIOC_12	SSEL0	AIN3
PC_13	W7500	6	AIN2	GPIOC_13	SCLK0	AIN2
	W7500P					
PC_14	W7500	7	AIN1	GPIOC_14	MISO0	AIN1
	W7500P					
PC_15	W7500	8	AIN0	GPIOC_15	MOSI0	AIN0
	W7500P					
PD_00	W7500	11	CRS	GPIOD_0		
PD_01	W7500	12	RXDV	GPIOD_1		
PD_02	W7500	13	RXD0	GPIOD_2		
PD_03	W7500	14	RXD1	GPIOD_3		
PD_04	W7500	15	RXD2	GPIOD_4		

## 15.4 Registers (Base address : 0x4100\_2000)

### PA\_00 pad alternate function select register (PA\_00\_AFR)

Address offset : 0x000

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA00AF														
															R/W

[1:0] PA00AF - PA\_00 Pad alternate function selection register.

These bits are written by S/W.

00 : GPIOA\_0

01 : GPIOA\_0

10 : PWM6/CAP6

11 : None

### PA\_01 pad alternate function select register (PA\_01\_AFR)

Address offset : 0x004

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA01AF														
															R/W

[1:0] PA01AF - PA\_01 Pad function selection register.

These bits are written by S/W.

00 : GPIOA\_1

01 : GPIOA\_1

10 : PWM7/CAP7

11 : None

## PA\_02 pad alternate function select register (PA\_02\_AFR)

Address offset : 0x008

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA02AF														
															R/W

[1:0] PA02AF - PA\_02 Pad function selection register.

These bits are written by S/W.

00 : GPIOA\_2

01 : GPIOA\_2

10 : CLKOUT

11 : None

## PA\_03 pad alternate function select register (PA\_03\_AFR)

Address offset : 0x00c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA03AF														
															R/W

[1:0] PA03AF - PA\_03 Pad function selection register.

These bits are written by S/W.

00 : SWCLK

01 : GPIOA\_3

10 : None

11 : PWM0/CAP0

## PA\_04 pad alternate function select register (PA\_04\_AFR)

Address offset : 0x010

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA04AF														
															R/W

[1:0] PA04AF - PA\_04 Pad function selection register.

These bits are written by S/W.

00 : SWDIO

01 : GPIOA\_4

10 : None

11 : PWM1/CAP1

## PA\_05 pad alternate function select register (PA\_05\_AFR)

Address offset : 0x014

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA05AF														
															R/W

[1:0] PA05AF - PA\_05 Pad function selection register.

These bits are written by S/W.

00 : SSEL0

01 : GPIOA\_5

10 : SCL1

11 : PWM2/CAP2

### PA\_06 pad alternate function select register (PA\_06\_AFR)

Address offset : 0x018

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA06AF														
															R/W

[1:0] PA06AF - PA\_06 Pad function selection register.

These bits are written by S/W.

00 : SCLK0

01 : GPIOA\_6

10 : SDA1

11 : PWM3/CAP3

### PA\_07 pad alternate function select register (PA\_07\_AFR)

Address offset : 0x01c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA07AF														
															R/W

[1:0] PA07AF - PA\_07 Pad function selection register.

These bits are written by S/W.

00 : MIS00

01 : GPIOA\_7

10 : CTS1

11 : PWM4/CAP4

### PA\_08 pad alternate function select register (PA\_08\_AFR)

Address offset : 0x020

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA08AF														
															R/W

[1:0] PA08AF - PA\_08 Pad function selection register.

These bits are written by S/W.

00 : MOSI0

01 : GPIOA\_8

10 : RTS1

11 : PWM5/CAP5

### PA\_09 pad alternate function select register (PA\_09\_AFR)

Address offset : 0x024

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA09AF														
															R/W

[1:0] PA09AF - PA\_09 Pad function selection register.

These bits are written by S/W.

00 : SCL0

01 : GPIOA\_9

10 : TXD1

11 : PWM6/CAP6

### PA\_10 pad alternate function select register (PA\_10\_AFR)

Address offset : 0x028

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA10AF														
															R/W

[1:0] PA10AF - PA\_10 Pad function selection register.

These bits are written by S/W.

00 : SDA0

01 : GPIOA\_10

10 : RXD1

11 : PWM7/CAP7

### PA\_11 pad alternate function select register (PA\_11\_AFR)

Address offset : 0x02c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA11AF														
															R/W

[1:0] PA11AF - PA\_11 Pad function selection register.

These bits are written by S/W.

00 : CTS0

01 : GPIOA\_11

10 : SSEL1

11 : None

### PA\_12 pad alternate function select register (PA\_12\_AFR)

Address offset : 0x030

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA12AF														
															R/W

[1:0] PA12AF - PA\_12 Pad function selection register.

These bits are written by S/W.

00 : RTS0

01 : GPIOA\_12

10 : SCLK1

11 : None

### PA\_13 pad alternate function select register (PA\_13\_AFR)

Address offset : 0x034

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA13AF														
															R/W

[1:0] PA13AF - PA\_13 Pad function selection register.

These bits are written by S/W.

00 : TXD0

01 : GPIOA\_13

10 : MISO1

11 : None

### PA\_14 pad alternate function select register (PA\_14\_AFR)

Address offset : 0x038

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA14AF														
															R/W

[1:0] PA14AF - PA\_14 Pad function selection register.

These bits are written by S/W.

00 : RXD0

01 : GPIOA\_14

10 : MOSI1

11 : None

### PA\_15 pad alternate function select register (PA\_15\_AFR)

Address offset : 0x03c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PA15AF														
														R/W	

[1:0] PA15AF - PA\_15 Pad function selection register.

These bits are written by S/W.

00 : GPIOA\_15

01 : GPIOA\_15

10 : None

11 : None

### PB\_00 pad alternate function select register (PB\_00\_AFR)

Address offset : 0x040

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15    14    13    12    11    10    9    8    7    6    5    4    3    2    1    0

res															

[1:0] PB00AF - PB\_00 Pad function selection register.

These bits are written by S/W.

00 : SSEL1

01 : GPIOB\_0

10 : CTS0

11 : None

### PB\_01 pad alternate function select register (PB\_01\_AFR)

Address offset : 0x044

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15    14    13    12    11    10    9    8    7    6    5    4    3    2    1    0

res	PB01AF														
															R/W

[1:0] PB01AF - PB\_01 Pad function selection register.

These bits are written by S/W.

00 : SCLK1

01 : GPIOB\_1

10 : RTS0

11 : None

### PB\_02 pad alternate function select register (PB\_02\_AFR)

Address offset : 0x048

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PB02AF														
															R/W

[1:0] PB02AF - PB\_02 Pad function selection register.

These bits are written by S/W.

00 : MISO1

01 : GPIOB\_2

10 : TXD0

11 : None

### PB\_03 pad alternate function select register (PB\_03\_AFR)

Address offset : 0x04c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PB03AF														

														R/W
--	--	--	--	--	--	--	--	--	--	--	--	--	--	-----

[1:0] PB03AF - PB\_03 Pad function selection register.

These bits are written by S/W.

00 : MOSI1

01 : GPIOB\_3

10 : RXD0

11 : None

### PB\_04 pad alternate function select register (PB\_04\_AFR)

Address offset : 0x050

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PB04AF														
															R/W

[1:0] PB04AF - PB\_04 Pad function selection register.

These bits are written by S/W.

00 : TXEN

01 : GPIOB\_4

10 : None

11 : None

### PB\_05 pad alternate function select register (PB\_05\_AFR)

Address offset : 0x054

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PB05AF														
															R/W

[1:0] PB05AF - PB\_05 Pad function selection register.

These bits are written by S/W.

00 : COL

01 : GPIOB\_5

10 : None

11 : None

### PB\_06 pad alternate function select register (PB\_06\_AFR)

Address offset : 0x058

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PB06AF														
															R/W

[1:0] PB06AF - PB\_06 Pad function selection register.

These bits are written by S/W.

00 : RXD3

01 : GPIOB\_6

10 : None

11 : None

### PB\_07 pad alternate function select register (PB\_07\_AFR)

Address offset : 0x05c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PB07AF														
															R/W

[1:0] PB07AF - PB\_07 Pad function selection register.

These bits are written by S/W.

00 : RXCLK

01 : GPIOB\_7

10 : None

11 : None

## PB\_08 pad alternate function select register (PB\_08\_AFR)

Address offset : 0x060

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PB08AF														
															R/W

[1:0] PB08AF - PB\_08 Pad function selection register.

These bits are written by S/W.

00 : DUP

01 : GPIOB\_8

10 : None

11 : None

## PB\_09 pad alternate function select register (PB\_09\_AFR)

Address offset : 0x064

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PB09AF														
															R/W

[1:0] PB09AF - PB\_09 Pad function selection register.

These bits are written by S/W.

00 : TXCLK

01 : GPIOB\_9

10 : None

11 : None

### PB\_10 pad alternate function select register (PB\_10\_AFR)

Address offset : 0x068

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PB10AF														
															R/W

[1:0] PB10AF - PB\_10 Pad function selection register.

These bits are written by S/W.

00 : TXDO

01 : GPIOB\_10

10 : None

11 : None

### PB\_11 pad alternate function select register (PB\_11\_AFR)

Address offset : 0x06c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PB11AF														
															R/W

[1:0] PB11AF - PB\_11 Pad function selection register.

These bits are written by S/W.

00 : TxD1

01 : GPIOB\_11

10 : None

11 : None

## PB\_12 pad alternate function select register (PB\_12\_AFR)

Address offset : 0x070

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PB12AF														
															R/W

[1:0] PB12AF - PB\_12 Pad function selection register.

These bits are written by S/W.

00 : TxD2

01 : GPIOB\_12

10 : None

11 : None

## PB\_13 pad alternate function select register (PB\_13\_AFR)

Address offset : 0x074

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PB13AF														
															R/W

[1:0] PB13AF - PB\_13 Pad function selection register.

These bits are written by S/W.

00 : TXD3

01 : GPIOB\_13

10 : None

11 : None

### PB\_14 pad alternate function select register (PB\_14\_AFR)

Address offset : 0x078

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PB14AF														
															R/W

[1:0] PB14AF - PB\_14 Pad function selection register.

These bits are written by S/W.

00 : MDIO

01 : GPIOB\_14

10 : None

11 : None

### PB\_15 pad alternate function select register (PB\_15\_AFR)

Address offset : 0x07c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PB15AF														
															R/W

[1:0] PB15AF - PB\_15 Pad function selection register.

These bits are written by S/W.

- 
- 00 : MDC  
 01 : GPIOB\_15  
 10 : None  
 11 : None

### PC\_00 pad alternate function select register (PC\_00\_AFR)

Address offset : 0x080

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC00AF														
															R/W

[1:0] PC00AF - PC\_00 Pad function selection register.

These bits are written by S/W.

- 00 : CTS1  
 01 : GPIOC\_0  
 10 : PWM0/CAP0  
 11 : None

### PC\_01 pad alternate function select register (PC\_01\_AFR)

Address offset : 0x084

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC01AF														
															R/W

[1:0] PC01AF - PC\_01 Pad function selection register.

These bits are written by S/W.

- 00 : RTS1

01 : GPIOC\_1  
 10 : PWM1/CAP1  
 11 : None

## PC\_02 pad alternate function select register (PC\_02\_AFR)

Address offset : 0x088

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC02AF														
															R/W

[1:0] PC02AF - PC\_02 Pad function selection register.

These bits are written by S/W.

00 : TxD1  
 01 : GPIOC\_2  
 10 : PWM2/CAP2  
 11 : None

## PC\_03 pad alternate function select register (PC\_03\_AFR)

Address offset : 0x08c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC03AF														
															R/W

[1:0] PC03AF - PC\_03 Pad function selection register.

These bits are written by S/W.

- 
- 00 : RXD1  
 01 : GPIOC\_3  
 10 : PWM3/CAP3  
 11 : None

### PC\_04 pad alternate function select register (PC\_04\_AFR)

Address offset : 0x090

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC04AF														
															R/W

[1:0] PC04AF - PC\_04 Pad function selection register.

These bits are written by S/W.

- 00 : SCL1  
 01 : GPIOC\_4  
 10 : PWM4/CAP4  
 11 : None

### PC\_05 pad alternate function select register (PC\_05\_AFR)

Address offset : 0x094

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC05AF														
															R/W

[1:0] PC05AF - PC\_05 Pad function selection register.

These bits are written by S/W.

- 00 : SDA1

- 
- 01 : GPIOC\_5  
 10 : PWM5/CAP5  
 11 : None

### PC\_06 pad alternate function select register (PC\_06\_AFR)

Address offset : 0x098

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC06AF														
															R/W

[1:0] PC06AF - PC\_06 Pad function selection register.

These bits are written by S/W.

00 : GPIOC\_6

01 : GPIOC\_6

10 : TxD2

11 : None

### PC\_07 pad alternate function select register (PC\_07\_AFR)

Address offset : 0x09c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC07AF														
															R/W

[1:0] PC07AF - PC\_07 Pad function selection register.

These bits are written by S/W.

00 : GPIOC\_7

01 : GPIOC\_7

10 : RXD2

11 : None

### PC\_08 pad alternate function select register (PC\_08\_AFR)

Address offset : 0x0a0

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC08AF														
															R/W

[1:0] PC08AF - PC\_08 Pad function selection register.

These bits are written by S/W.

00 : PWM0/CAP0

01 : GPIOC\_8

10 : SCL0

11 : ADC\_IN7

### PC\_09 pad alternate function select register (PC\_09\_AFR)

Address offset : 0x0a4

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC09AF														
															R/W

[1:0] PC09AF - PC\_09 Pad function selection register.

These bits are written by S/W.

00 : PWM1/CAP1

01 : GPIOC\_9

10 : SDA0

---

11 : ADC\_IN6

### PC\_10 pad alternate function select register (PC\_10\_AFR)

Address offset : 0x0a8

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC10AF														
															R/W

[1:0] PC10AF - PC\_10 Pad function selection register.

These bits are written by S/W.

00 : TXD2

01 : GPIOC\_10

10 : PWM2/CAP2

11 : ADC\_IN5

### PC\_11 pad alternate function select register (PC\_11\_AFR)

Address offset : 0x0ac

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC11AF														
															R/W

[1:0] PC11AF - PC\_11 Pad function selection register.

These bits are written by S/W.

00 : RXD2

01 : GPIOC\_11

10 : PWM3/CAP3

11 : ADC\_IN4

## PC\_12 pad alternate function select register (PC\_12\_AFR)

Address offset : 0x0b0

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC12AF														
															R/W

[1:0] PC12AF - PC\_12 Pad function selection register.

These bits are written by S/W.

00 : ADC\_IN3

01 : GPIOC\_12

10 : SSELO

11 : ADC\_IN3

## PC\_13 pad alternate function select register (PC\_13\_AFR)

Address offset : 0x0b4

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC13AF														
															R/W

[1:0] PC13AF - PC\_13 Pad function selection register.

These bits are written by S/W.

00 : ADC\_IN2

01 : GPIOC\_13

10 : SCLK0

11 : ADC\_IN2

## PC\_14 pad alternate function select register (PC\_14\_AFR)

Address offset : 0x0b8

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC14AF														
															R/W

[1:0] PC14AF - PC\_14 Pad function selection register.

These bits are written by S/W.

00 : ADC\_IN1

01 : GPIOC\_14

10 : MISO0

11 : ADC\_IN1

## PC\_15 pad alternate function select register (PC\_15\_AFR)

Address offset : 0x0bc

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PC15AF														
															R/W

[1:0] PC15AF - PC\_15 Pad function selection register.

These bits are written by S/W.

00 : ADC\_IN0

01 : GPIOC\_15

10 : MOSI0

11 : ADC\_IN0

## PD\_00 pad alternate function select register (PD\_00\_AFR)

Address offset : 0x0c0

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PD00AF														
															R/W

[1:0] PD00AF - PD\_00 Pad function selection register.

These bits are written by S/W.

00 : CRS

01 : GPIOD\_0

10 : None

11 : None

## PD\_01 pad alternate function select register (PD\_01\_AFR)

Address offset : 0x0c4

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PD01AF														
															R/W

[1:0] PD01AF - PD\_01 Pad function selection register.

These bits are written by S/W.

00 : RXDV

01 : GPIOD\_1

10 : None

11 : None

## PD\_02 pad alternate function select register (PD\_02\_AFR)

Address offset : 0x0c8

Reset value : 0x0000\_0000

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

res	PD02AF														
															R/W

[1:0] PD02AF - PD\_02 Pad function selection register.

These bits are written by S/W.

00 : RXD0

01 : GPIOD\_2

10 : None

11 : None

## PD\_03 pad alternate function select register (PD\_03\_AFR)

Address offset : 0x0cc

Reset value : 0x0000\_0000

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

res	PD03AF														
															R/W

[1:0] PD03AF - PD\_03 Pad function selection register.

These bits are written by S/W.

00 : RXD1

01 : GPIOD\_3

10 : None

11 : None

## PD\_04 pad alternate function select register (PD\_04\_AFR)

Address offset : 0x0d0

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PD04AF														
															R/W

[1:0] PD04AF - PD\_04 Pad function selection register.

These bits are written by S/W.

00 : RXD2

01 : GPIOD\_4

10 : None

11 : None

## 15.5 Register map

The following Table 10 summarizes the AFC registers.

Table 10 AFC register map and reset values





---

## 16 External Interrupt (EXTI)

### 16.1 Introduction

Each functional pads are connected to the external interrupt(EXTINT) source.

### 16.2 Features

- All functional pads can be used as an external interrupt source regardless of any set of pad function.
- External Interrupt controller has the following functions and can be controlled by registers.
  - Interrupt mask (enable or disable, default : disable)
  - Interrupt polarity (rising or falling, default : rising)

### 16.3 Functional description

All pads are connected to the control register individually. (External interrupt mask register and External Interrupt polarity register)

External interrupt working as following expression:

- Each pad interrupt = Interrupt mask & (Interrupt polarity ^ Pad input)
- EXTINT = any Each pad interrupt

Figure 10 shows the External Interrupt diagram.

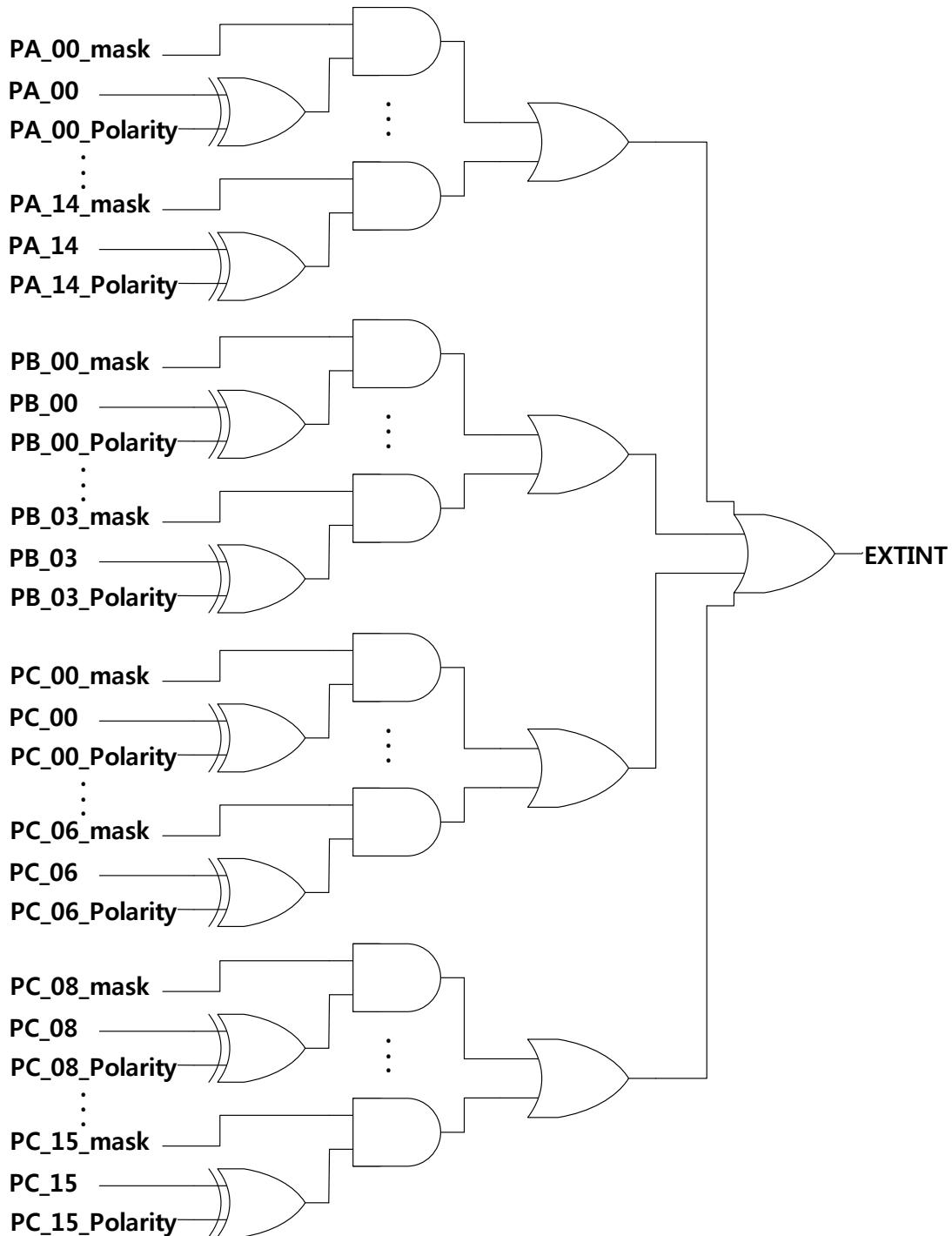


Figure 10. External Interrupt diagram

## 16.4 Registers (Base address : 0x4100\_2000)

### External interrupt enable register (Px\_y EXTINT)

Address offset : 0x200

Reset value : 0x0000\_0000

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

[0] POL - External interrupt polarity selection register

These bits are written by S/W.

0 : interrupt occurs when pad detect HIGH level signal

1 : interrupt occurs when pad detect LOW level signal

[1] PA00IEN - External interrupt enable register

These bits are written by S/W.

0 : external interrupt disable

1 : external interrupt enable

## 16.5 Register map

The following Table 11 summarizes the EXTINT registers.

Table 11 EXTINT register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	Px_y EXTINT (where x = A..D where y = 0..15)	res	IEN	1	POL	0																											
	reset value																																

## 17 Pad Controller (PADCON)

### 17.1 Introduction

Pads of W7500x are controllable. User can control pad's characteristic.

### 17.2 Features

- W7500x has digital I/O pads and digital/analog mux I/O pads
- Controllable characteristics of pads are pull-up, pull-down, driving strength, input enable, and CMOS/Schmitt trigger input buffer
- Each pad can be controlled individually by register.

### 17.3 Functional description

Figure 11 shows the function schematic of digital I/O pad of W7500x.

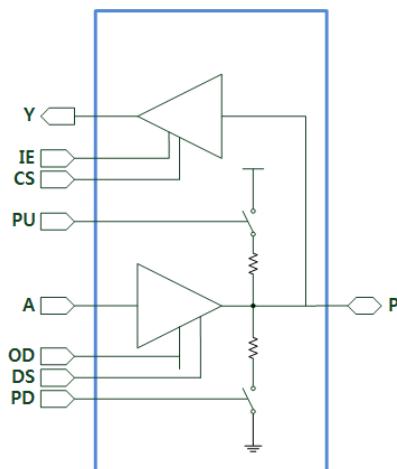


Figure 11. function schematic of digital I/O pad

Figure 12 shows the function schematic of digital/analog mux IO pad of W7500x

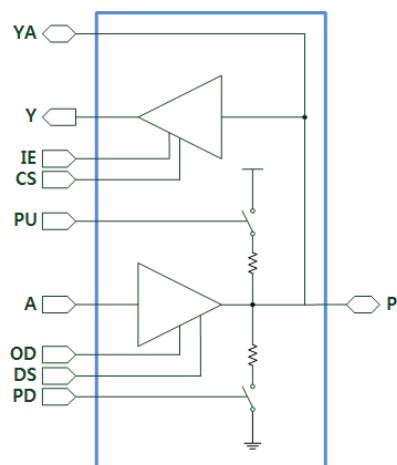


Figure 12. function schematic of digital/analog mux IO pad

Initials of Pad diagram is same as below.

P - PAD

YA - Analog Input (connect to ADC input)

Y - Digital Input

IE - Input buffer enable

Condition		A	Y	P
Input buffer enable (IE = 1)	Output mode	OUT	OUT	OUT
	Input mode	No use	IN	IN
Input buffer disable (IE = 0)	Output mode	OUT	Low (0)	OUT
	Input mode	No use	IN	IN

CS - CMOS/Schmitt trigger input buffer select

PU - Pull-up enable

A - Digital Output

OD - Open Drain

DS - Driving strength select

Condition		Rise/Fall Time (nSec)		Propagation Delay (nSec)	
Driving Strength	Capacitance loading	Min	Max	Min	Max
Low (DS = 1)	25pF	4	18	7	27
	100pF	11	53	11	44
High (DS = 0)	25pF	1	8	4	16
	100pF	4	23	7	24

PD - Pull-down enable

User can set pad condition with IE, CS, PU/PD, OD, DS by register.

And pads are can be controlled individually.

## 17.4 Registers (Base address : 0x4100\_3000)

### PAD Control register (Px\_y PCR)(x=A..D, y=0..15)

Address offset : 0x000

Reset value : 0x0000\_0060

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	res														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CS	IE	res	OD	DS	PUPD									
									R/W	R/W		R/W	R/W	R/W	

[1:0] PUPD - Pull-up, Pull-down selection register

These bits are written by S/W.

00 : Neither

01 : pull-down

10 : pull-up

11 : Neither

[2] DS - Driving strength selection register

0 : High driving strength

1 : Low driving strength

[3] OD - Open Drain output mode register

0 : Open Drain output mode disable

1 : Open Drain output mode enable

[5] IE : Input buffer enable selection register

0 : Input buffer disable

1 : Input buffer enable

[6] CS - CMOS input or Summit trigger input selection register

0 : Summit trigger input buffer

1 : CMOS input buffer

## 17.5 Register map

The following Table 12 summarizes the PAD controller registers.

Table 12 PAD controller register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	Px_y PCR (x = A..D y = 0..15)	res	CS	IE	res	OD	DS	PUPD																									
	reset value																								1	1	0	0	0	0	0	0	

## 18 General-purpose I/Os(GPIO)

### 18.1 Introduction

The GPIO(General-Purpose I/O Port) is composed of three physical GPIO blocks, each corresponding to an individual GPIO port(PORT A, PORT B and PORT C). The GPIO supports up to 34 programmable input/output pins, depending on the peripherals being used.

### 18.2 Features

- The GPIO peripheral consists the following features.
  - GPIO\_DATAOUT can SET/CLEAR by the SET register and CLEAR register. (1 for set and 0 for clear)
  - Mask registers allow treating sets of port bits as a group leaving other bits unchanged.
  - Up to 34 GPIOs depending on configuration
  - Programmable control for GPIO interrupts
    - Interrupt generation masking
    - Edge-triggered on rising, falling, or both

### 18.3 Functional description

Figure 13 shows the GPIO block diagram.

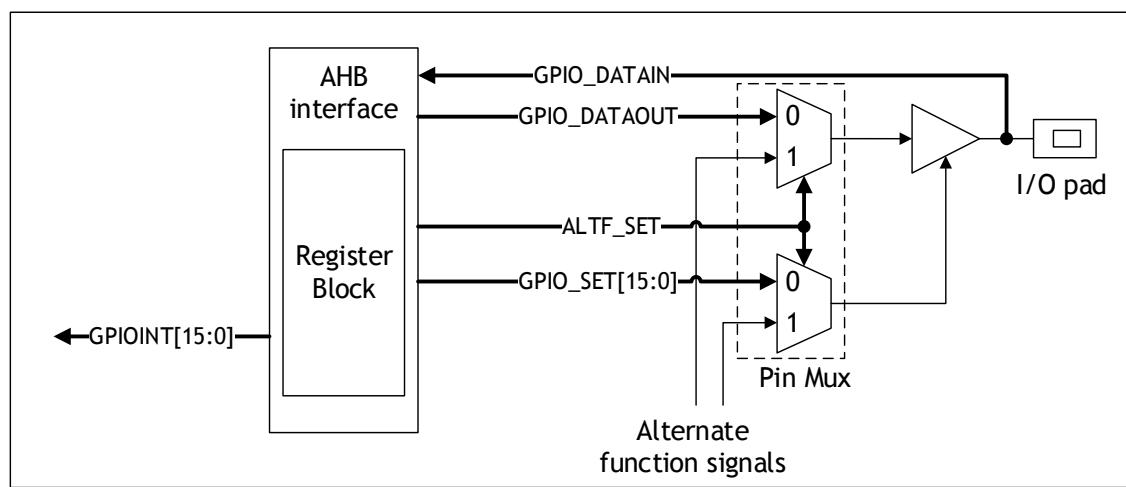


Figure 13. GPIO block diagram

Figure 14 shows the operation sequences available for the GPIO.

The pad alternate function is using the pad alternate function select register.

Refer to ‘Alternate Function Controller(AFC)’ for more details about each register.

The pad control supports pull-down, pull-up, input buffer, and summit trigger input buffer. Refer to ‘Pad Controller (PADCON)’ for more details about each register.

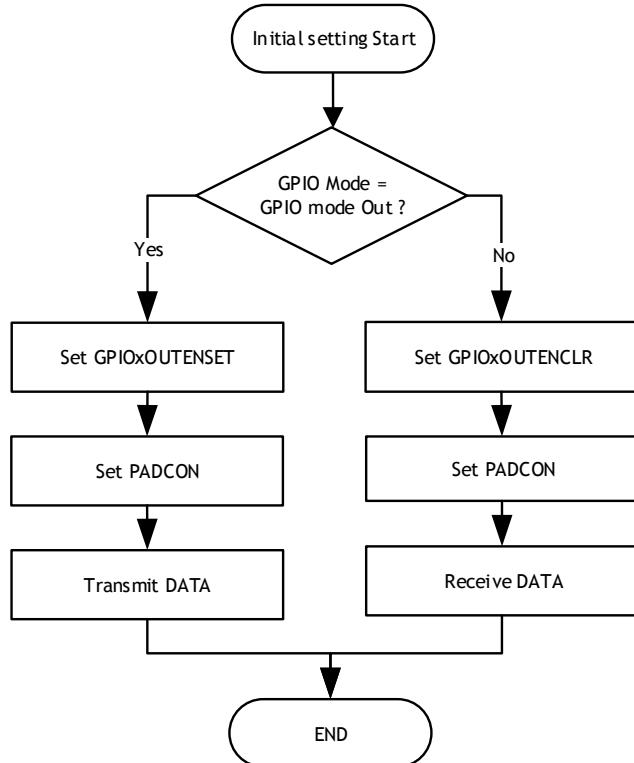


Figure 14. GPIO Flow chart

## Masked access

The masked access feature permits individual bits or multiple bits to be read from or written to in a single transfer. This avoids software-based read-modify-write operations that are not thread safe. With the masked access operations, the 16-bit I/O is divided into two halves, lower byte and upper byte. The bit mask address spaces are defined as two arrays each containing 256 words.

For example, to set bits[1:0] to 1 and clear bits[7:6] in a single operation, users can carry out the write to the lower byte mask access address space. The required bit mask is 0xC3, and users can write the operation as MASKLOWBYTE[0xC3] = 0x03. Refer to Figure 15 below.

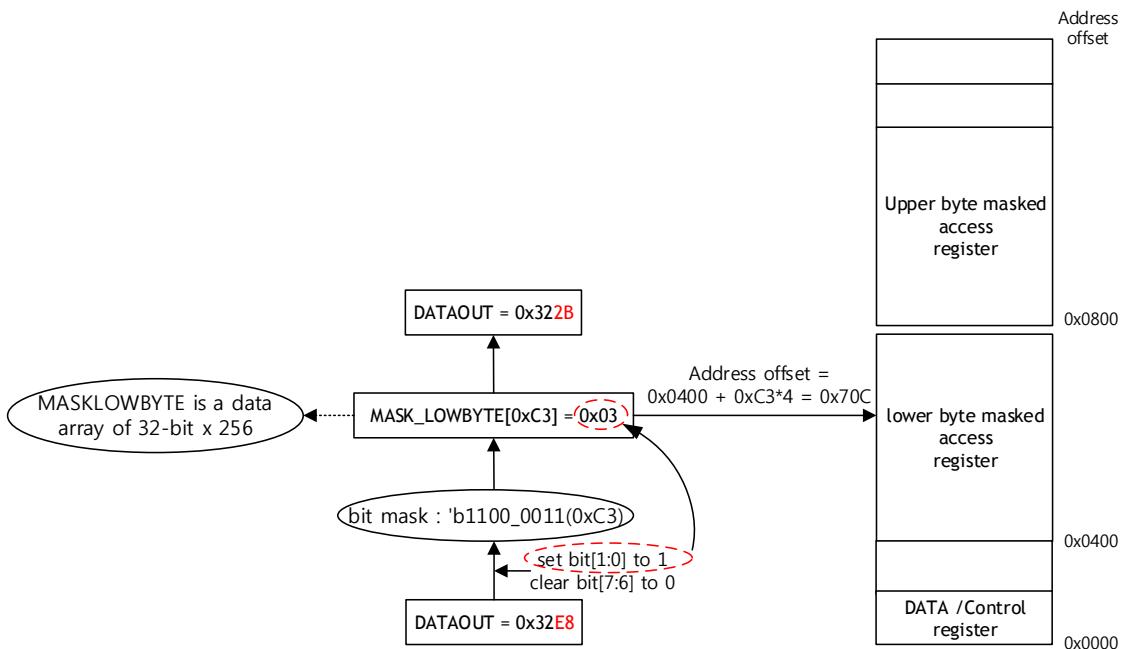


Figure 15. MASK LOWBYTE access

To update some of the bits in the upper eight bits of the GPIO port, users can use the **MASKHIGHBYTE** array as Figure 16 below.

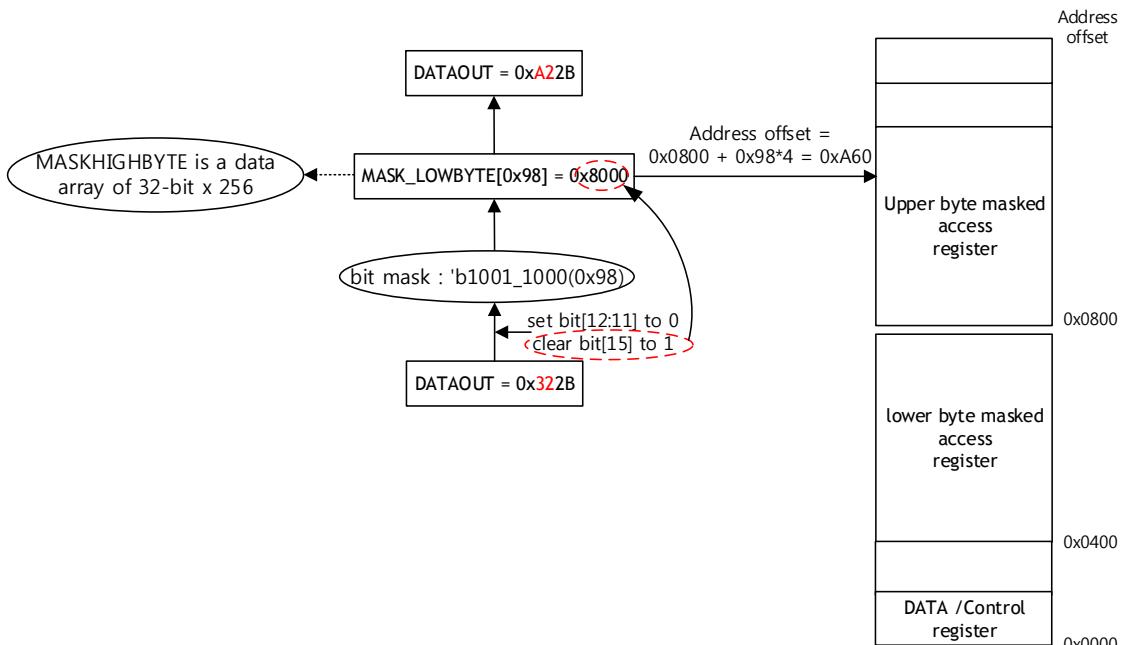


Figure 16 MASK HIGHBYTE access

## 18.4 GPIO Registers(Address Base: 0x4200\_0000)

### GPIO Data Register(GPIOx\_DATA) (x=A..D)

Address offset: 0x0000

Reset value: 0x----

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAT15	DAT14	DAT13	DAT12	DAT11	DAT10	DAT9	DAT8	DAT7	DAT6	DAT5	DAT4	DAT3	DAT2	DAT1	DAT0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

[15:0] DATy(y = 0..15)

READ as : Port input data bit

### GPIO Output Latch Register(GPIOx\_DATAOUT) (x=A..D)

Address offset: 0x0004

Reset value: 0x----

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAO15	DAO14	DAO13	DAO12	DAO11	DAO10	DAO9	DAO8	DAO7	DAO6	DAO5	DAO4	DAO3	DAO2	DAO1	DAO0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[15:0] DAOy(y = 0..15)

READ as : Port out data bit

WRITE as : WRITE to GPIOA\_DATAOUT register

### GPIO Enable Set Register(GPIOx\_OUTENSET) (x=A..D)

Address offset: 0x0010

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES15	ES14	ES13	ES12	ES11	ES10	ES9	ES8	ES7	ES6	ES5	ES4	ES3	ES2	ES1	ES0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[15:0] ESy(y = 0..15)

WRITE as :

'0' is no effect

'1' is sets the corresponding output enable bit

READ as :

'0' is indicates the signal direction as input

'1' is indicates the signal direction as output

### GPIO Enable Clear Register(GPIOx\_OUTENCLR) (x=A..D)

Address offset: 0x0014

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EC15	EC14	EC13	EC12	EC11	EC10	EC9	EC8	EC7	EC6	EC5	EC4	EC3	EC2	EC1	EC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[15:0] ECy(y = 0..15)

WRITE as :

'0' is no effect

'1' is clears the output enable bit

READ as :

'0' is indicates the signal direction as input

'1' is indicates the signal direction as output

### GPIO Interrupt Enable Set Register(GPIOx\_INTENSET) (x=A..D)

Address offset: 0x0020

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IES15	IES14	IES13	IES12	IES11	IES10	IES9	IES8	IES7	IES6	IES5	IES4	IES3	IES2	IES1	IES0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[15:0] IESy(y = 0..15)

WRITE as :

'0' is no effect

'1' is sets the interrupt enable bit

READ as :

'0' is indicates the interrupt disable

'1' is indicates the interrupt enable

### GPIO Interrupt Enable Clear Register(GPIOx\_INTENCLR) (x=A..D)

Address offset: 0x0024

Reset value: 0x----

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEC15	IEC14	IEC13	IEC12	IEC11	IEC10	IEC9	IEC8	IEC7	IEC6	IEC5	IEC4	IEC3	IEC2	IEC1	IEC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[15:0] IECy(y = 0..15)

WRITE as :

'0' is no effect

'1' is clears the interrupt enable bit

READ as :

'0' is indicates the interrupt disable

'1' is indicates the interrupt enable

### GPIO Interrupt Type Set Register(GPIOx\_INTTYPESET) (x=A..D)

Address offset: 0x0028

Reset value: 0x----

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITS15	ITS14	ITS13	ITS12	ITS11	ITS10	ITS9	ITS8	ITS7	ITS6	ITS5	ITS4	ITS3	ITS2	ITS1	ITS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[15:0] ITS<sub>y</sub>(y = 0..15)

WRITE as :

'0' is no effect

'1' is sets the interrupt type bit

READ as :

'0' is indicates for LOW or HIGH level

'1' is indicates for falling edge or rising edge

### GPIO Interrupt Type Clear Register(GPIOx\_INTERRUPTCLR) (x=A..D)

Address offset: 0x002c

Reset value: 0x----

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ITC15	ITC14	ITC13	ITC12	ITC11	ITC10	ITC9	ITC8	ITC7	ITC6	ITC5	ITC4	ITC3	ITC2	ITC1	ITC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[15:0] ITC<sub>y</sub>(y = 0..15)

WRITE as :

'0' is no effect

'1' is clears the interrupt type bit

READ as :

'0' is indicates for LOW or HIGH level

'1' is indicates for falling edge or rising edge

### GPIO Interrupt Polarity Set Register(GPIOx\_INTERRUPTPOLSET) (x=A..D)

Address offset: 0x0030

Reset value: 0x----

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPS15	IPS14	IPS13	IPS12	IPS11	IPS10	IPS9	IPS8	IPS7	IPS6	IPS5	IPS4	IPS3	IPS2	IPS1	IPSO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[15:0] IPSy(y = 0..15)

WRITE as :

'0' is no effect

'1' is sets the interrupt polarity bit

READ as :

'0' is indicates for LOW level or falling edge

'1' is indicates for HIGH level or rising edge

### GPIO Interrupt Polarity Clear Register(GPIOx\_INTPOLCLR) (x=A..D)

Address offset: 0x0034

Reset value: 0x----

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPC15	IPC14	IPC13	IPC12	IPC11	IPC10	IPC9	IPC8	IPC7	IPC6	IPC5	IPC4	IPC3	IPC2	IPC1	IPCO
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[15:0] IPCy(y = 0..15)

WRITE as :

'0' is no effect

'1' is clears the interrupt polarity bit

READ as :

'0' is indicates for LOW level or falling edge

'1' is indicates for HIGH level or rising edge

## GPIO Interrupt Status/Interrupt Clear Register(GPIO\_ INTSTATUS/INTCLEAR) (x=A..D)

Address offset: 0x0038

Reset value: 0x----

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISC15	ISC14	ISC13	ISC12	ISC11	ISC10	ISC9	ISC8	ISC7	ISC6	ISC5	ISC4	ISC3	ISC2	ISC1	ISC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

[15:0] ISC<sub>y</sub>(y = 0..15)

WRITE as :

'0' is no effect

'1' is request to clear the interrupt

READ as : IRQ status Register

## GPIO Lower Byte Masked Access Register(GPIOx\_LB\_MASKED) (x=A..D)

Address offset: 0x0400 - 0x07FC

Reset value: 0x----

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	LBM														
R/W															

**Lower eight bits masked access**

[15:8] Not used

[7:0] Data for lower byte access, with bits[9:2] of address value used as enable bit mask for each bit

## GPIO Upper Byte Masked Access Register(GPIOx\_ UB\_MASKED) (x=A..D)

Address offset: 0x0800-0x0FC

Reset value: 0x----

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HBM								res							
R/W															

### Higher eight bits masked access

[15:8] Data for higher byte access, with bits[9:2] of address value used as enable bit mask  
for each bit

[7:0] Not used

## 18.5 Register map

The following Table 13 summarizes the GPIO registers.

Table 13 GPIO register map and reset values

## 19 Direct memory access controller (DMA)

### 19.1 Introduction

Direct memory access (DMA) is used in order to provide high-speed data transfer between peripherals and memory as well as memory to memory. Data can be quickly moved by DMA without any CPU actions. This keeps CPU resources free for other operations.

The DMA controller has up to 6 channels in total, each dedicated to managing memory access requests from one or more peripherals. It has an arbiter for handling the priority between DMA requests. For more details, refer to “PrimeCell® µDMA Controller (PL230)” from the Technical Reference Manual

### 19.2 Features

- 6 channels
- Each channel is connected to dedicated hardware DMA requests and software trigger is also supported on each channel.
- Priorities between requests from the DMA channels are software programmable (2 levels consisting of high, default)
- Memory-to-memory transfer (software request only)
- TCP/IP-to-memory transfer (software request only)
- SPI/UART-to-memory transfer (hardware request and software request)
- Access to Flash, SRAM, APB and AHB peripherals as source and destination

### 19.3 Functional description

Figure 17 shows the DMA block diagram.

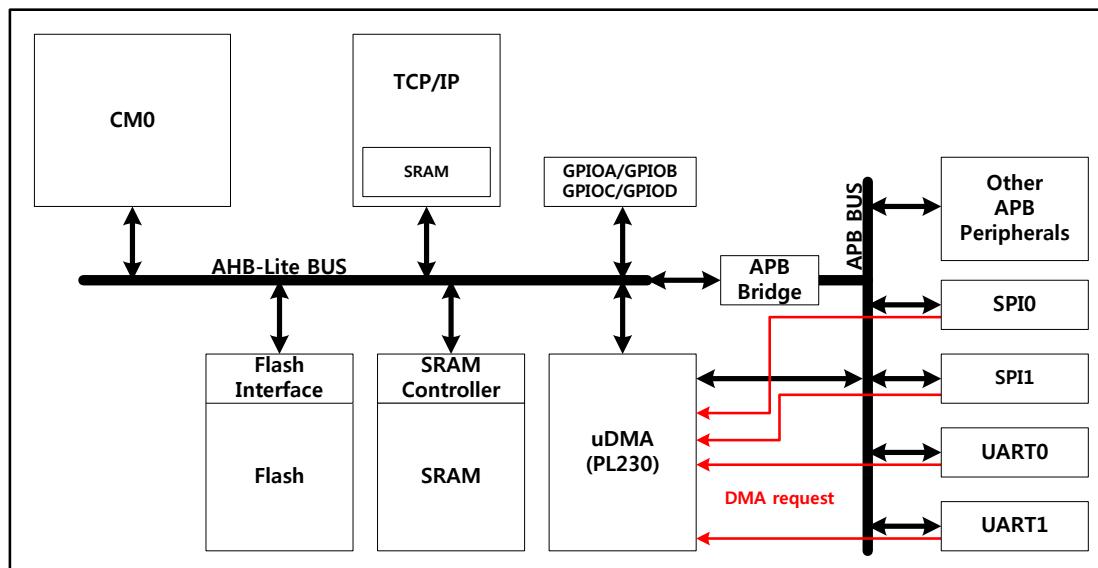


Figure 17. DMA Block diagram

## DMA request mapping

The hardware requests from the peripherals (UART0, UART1, SSP0, SSP1) are simply connected to the DMA. Refer to Table 17 which lists the DMA requests for each channel.

Table 14 Summary of the DMA requests for each channel

	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5	Channel 6
Hardware Request	SSP0_TX SSP0_RX	SSP1_TX SSP1_RX	UART0_TX UART0_RX	UART1_TX UART1_RX	NONE	NONE
Software Request <sup>(1)</sup>	Support	Support	Support	Support	Support	Support

1. Software request is the only way to use DMA for memory-to-memory or TCP/IP-to-memory.

## DMA arbitration

The controller can be configured to perform arbitration during a DMA cycle before and after a programmable number of transfers. This reduces the latency for servicing a higher priority channel.

The controller uses four bits in the channel control data structure that configures how many AHB bus transfers occur before the controller re-arbitrates. These bits are known as the R\_power bits because the value R is raised to the power of two and this determines the arbitration rate. For example, if R = 4, then the arbitration rate is  $2^4$ , which means the controller arbitrates every 16 DMA transfers.

**Remark:** Do not assign a low-priority channel with a large R\_power value because this prevents the controller from servicing high-priority requests until it re-arbitrates.

When  $N > 2^R$  and is not an integer multiple of  $2^R$ , then the controller always performs sequences of  $2^R$  transfers until  $N < 2^R$  remain to be transferred. The controller performs the remaining N transfers at the end of the DMA cycle.

## DMA cycle types

The cycle\_ctrl bits in the channel control data structure controls how the DMA controller performs a cycle.

The controller uses four cycle types described in this manual:

- Invalid
- Basic
- Auto-request

---

- Ping-pong

See ARM micro DMA (PL230) documentation for additional cycle types.

For all cycle types, the controller arbitrates after  $2^R$  DMA transfers. If a low-priority channel is set to a large  $2^R$  value then it prevents all other channels from performing a DMA transfer until the low-priority DMA transfer completes. Therefore, the user must take care when setting the R\_power bit in the channel\_cfg data structure, that the latency for high-priority channels is not significantly increased.

### Invalid cycle

After the controller completes a DMA cycle, it sets the cycle type to invalid to prevent it from repeating the same DMA cycle.

### Basic cycle

In this mode, the controller can be configured to use either the primary or the alternate channel control data structure. After the channel is enabled and the controller receives a request for this channel, the flow for basic cycle is as below:

1. The controller performs  $2^R$  transfers.  
If the number of transfers remaining is zero the flow continues at step 3.
2. The controller arbitrates:
  - If a higher-priority channel is requesting service, then the controller services that channel.
  - If the peripheral or software signals a request to the controller, then it continues at step 1.
3. The controller sets dma\_done[c] signal for this channel HIGH for one system clock cycle.  
This indicates to the host processor that the DMA cycle is complete.

### Auto-request cycle

When the controller operates in this mode, it is only necessary to receive a single request to enable the controller to complete the entire DMA cycle. This enables a large data transfer to occur, without significantly increasing the latency for servicing higher priority requests or requiring multiple requests from the processor or peripheral.

The auto-request cycle is typically used for memory-to-memory requests. In this case, software generates the starting request for the  $2^R$  transfers after setting up the DMA control data structure.

---

In this mode, the controller can be configured to use either the primary or the alternate channel control data structure. After the channel is enabled and the controller receives a request for this channel, the flow for the auto-request cycle is as below:

1. The controller performs  $2^R$  transfers. If the number of transfers remaining is zero the flow continues at step 3.
2. The controller arbitrates if there are any transfers remaining after  $2^R$  transfers. If the current channel c has the highest priority, the cycle continues at step 1.

The controller sets `dma_done[c]` signal for this channel HIGH for one system clock cycle. This indicates to the host processor that the DMA cycle is complete.

### Ping-pong cycle

In this mode, the controller performs a DMA cycle using one of the data structures and then performs a DMA cycle using the other data structure. The controller continues to switch between primary and alternate structures until it reads a data structure that is invalid, until the user reprograms the `cycle_type` to basic, or until the host processor disables the channel.

In ping-pong mode, the user can program or reprogram one of the two channel data structures (primary or alternate) while using the other channel data structure for the active transfer. When a transfer is done, the next transfer can be started immediately using the prepared channel data structure - provided that a higher priority channel does not require servicing. If the user does not reprogram the channel control data structure not in use for a transfer, the cycle type remains invalid (which is the value at the end of the last transfer using that structure), and the ping-pong cycle completes.

The ping-pong cycle can be used for transfers to or from peripherals or for memory- to-memory transfers.

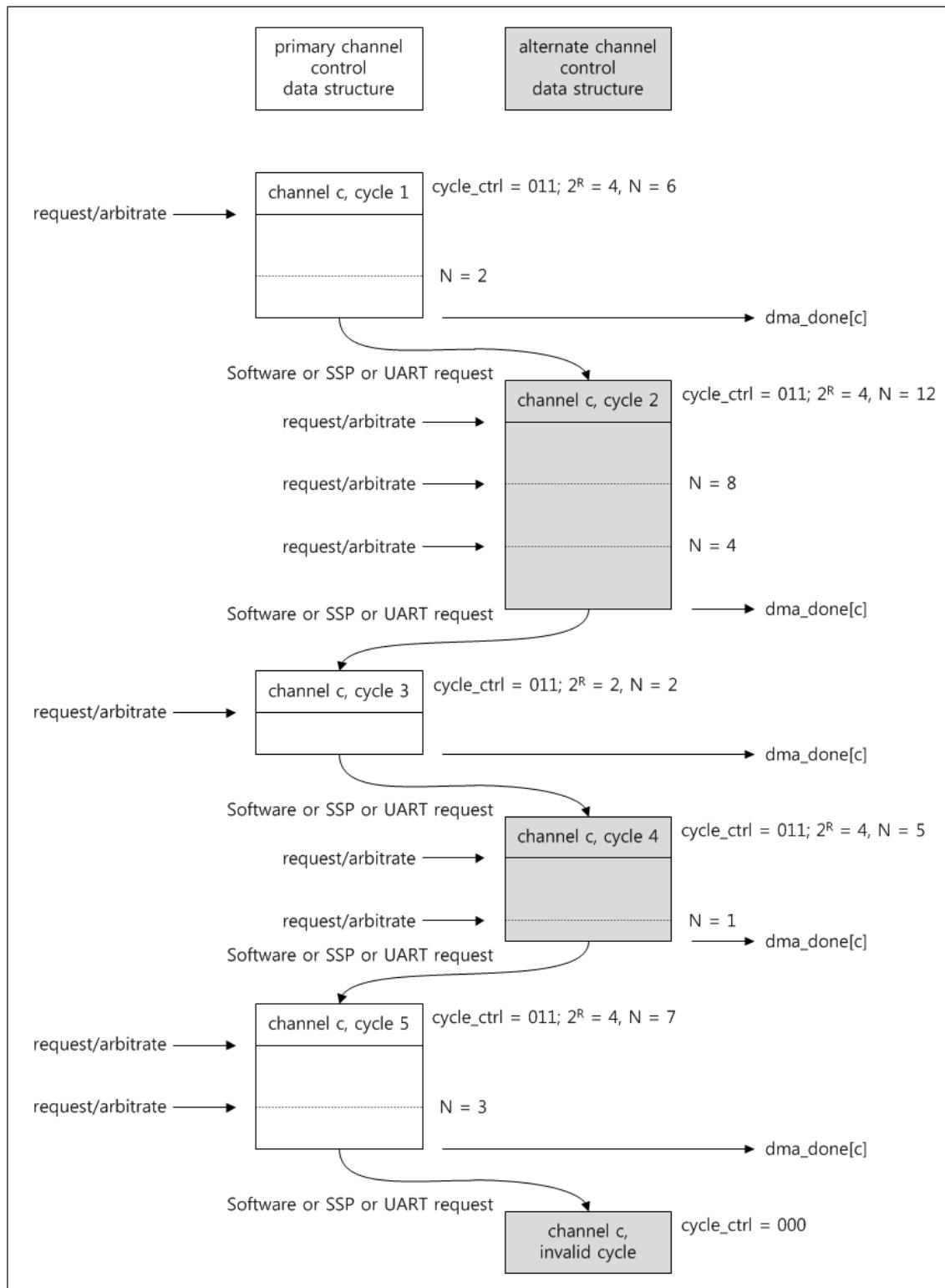


Figure 18. DMA ping pong cycle

## 19.4 Registers (Base address : 0x4100\_4000)

### DMA status register (DMA\_STATUS)

Address offset : 0x000

Reset value : 0x0005\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res			STATE		res	res	res	ENABLE							
										RO					RO

#### [0] ENABLE - Enable status of the controller

This bit is read only register to check enable status of DMA controller

0 : controller is disabled

1 : controller is enabled

#### [7:4] STATE - Current state of the control state machine.

These bits are read only register to check current state of controller. State can be one of the following

0000 : idle

0001 : reading channel controller data

0010 : reading source data end pointer

0011 : reading destination data end pointer

0100 : reading source data

0101 : writing destination data

0110 : waiting channel controller data

1000 : stalled

1001 : done

1010 : peripheral scatter-gather transition

1011 - 1111 : undefined.

## DMA configuration register (DMA\_CFG)

Address offset : 0x004

Reset value : -

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res		PROT_CTRL		res	res	res	res	ENABLE							
									WO						WO

[0] ENABLE - Enable for the controller

This bit is write only register to enable of DMA controller

0 : disable the controller

1 : enable the controller

[7:5] PROT\_CTRL - Set the AHB-Lite protection by controlling the HPROT[3:1] signal levels

These bits are write only register to set HPROT[3:1] signal as follows

[7] : controls HPROT[3] to indicate if a cacheable access is occurring.

[6] : controls HPROT[2] to indicate if a bufferable access is occurring.

[5] : controls HPROT[1] to indicate if a privileged access is occurring.

## DMA control data base pointer register (DMA\_CTRL\_BASE\_PTR)

Address offset : 0x008

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTRL_BASE_PTR[31:16]															
R/W															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTRL_BASE_PTR[15:8]								res							
R/W															

[31:8] CTRL\_BASE\_PTR - Pointer to the base address of the primary data structure

These bits are read/write register. User must configure this register so that the base pointer points to a location in system memory.

## DMA channel alternate control data base pointer register (DMA\_ALT\_CTRL\_BASE\_PTR)

Address offset : 0x00c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALT_CTRL_BASE_PTR[31:16]															
RO															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALT_CTRL_BASE_PTR[15:0]															

RO
----

[31:0] ALT\_CTRL\_BASE\_PTR : Base address of the alternate data structure

This read only register returns the base address of the alternate data structure.

### DMA channel wait on request status register (DMA\_WAITONREQ\_STATUS)

Address offset : 0x010

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															

[Channel-1] DMA\_WAITONREQ - Channel wait on request status

This read-only register returns the status of dma\_waitonreq[Channel-1].

0 : dma\_waitonreq is low

1 : dma\_waitonreq is high

### DMA channel software request register (DMA\_CHNL\_SW\_REQUEST)

Address offset : 0x014

Reset value : -

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															

[Channel-1] CHNL\_SW\_REQUEST - Set the appropriate bit to generate a software DMA request on the corresponding DMA channel

This read-only register enables to generate a software DMA request.

0 : dose not create a DMA request for [Channel-1]

1 : creates a DMA request for [Channel -1]

### DMA channel useburst set register (DMA\_CHNL\_USEBURST\_SET)

Address offset : 0x018

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															

[Channel-1] CHNL\_USEBURST\_SET - Returns the useburst status, or disable dma\_sreq[Channel-1] form generating DMA requests.

This read/write register disables the single request dma\_sreq[Channel-1] input from generating requests, and therefore only the request, dma\_req[Channel-1], generates requests. Reading the register returns the useburst status.

Read as :

0 : DMA [Channel-1] responds to requests that it receives on dma\_req[Channel-1] or dma\_sreq[Channel-1]

1 : DMA [Channel-1] does not responds to requests that it receives on dma\_sreq[Channel-1]. The controller only responds to dma\_req[Channel-1] requests

Write as :

0 : No effect. Use the CHNL\_USEBURST\_CLR register to set bit [Channel-1] to 0

1 : Disables dma\_sreq[Channel-1] form generating DMA requests.

### DMA channel useburst clear register (DMA\_CHNL\_USEBURST\_CLR)

Address offset : 0x01c

Reset value : -

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CHNL_USEBURST_CLR[5:0]														
															WO

[Channel-1] CHNL\_USEBURST\_CLR - Set the appropriate bit to enable dma\_sreq[Channel-1] to generate requests.

This Write register enables dma\_sreq[Channel-1] to generate requests.

0 - No effect. User the CHNL\_USEBURST\_SET register to disable dma\_sreq[Channel-1] from generating requests.

1 - Enable dma\_sreq[Channel-1] to generate DMA requests.

### DMA channel request mask set register (DMA\_CHNL\_REQ\_MASK\_SET)

Address offset : 0x020

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CHNL_REQ_MASK_SET[5:0]														
															R/W

[Channel-1] CHNL\_REQ\_MASK\_SET - Returns the request mask status of dma\_req[Channel-1] and dma\_sreq[Channel-1], or disables the corresponding channel from generating DMA requests.

This read/write register disables a HIGH on dma\_req[Channel-1], for dma\_sreq[Channel-1], from generating a request. Reading the register returns the request mask status for dma\_req[Channel-1] and dma\_sreq[Channel-1].

Read as :

0 - External requests are enabled for channel [Channel-1]

1 - External requests are disabled for channel [Channel-1]

Write as :

0 - No effect. Use the CHNL\_REQ\_MASK\_CLR register to enable DMA requests.

1 - Disables dma\_req[Channel-1] and dma\_sreq[Channel-1] from generating DMA requests.

## DMA channel request mask clear register (DMA\_CHNL\_REQ\_MASK\_CLR)

Address offset : 0x024

Reset value : -

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															
															WO

[Channel-1] CHNL\_REQ\_MASK\_CLR - Set the appropriate bit to enable DMA requests for the channel corresponding to dma\_req[Channel-1] and dma\_sreq[Channel-1]

This write only register enables a HIGH on dma\_req[Channel-1], or dma\_sreq[Channel-1].

0 : No effect. Use the CHNL\_REQ\_MASK\_SET register to disable dma\_req[Channel-1] and dma\_sreq[Channel-1] from generating requests.

1 : Enables dma\_req[Channel-1] or dma\_sreq[Channel-1] to generate DMA requests.

## DMA channel enable set register (DMA\_CHNL\_ENABLE\_SET)

Address offset : 0x028

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															
															R/W

[Channel-1] CHNL\_ENABLE\_SET - Returns the enable status of channels or enables the corresponding channels.

This read/write register enables a DMA channel. Reading the register returns the enable status of the channels.

Read as :

0 - Channel [Channel-1] is disabled

1 - Channel [Channel-1] is enabled

Write as :

0 - No effect. Use the CHNL\_ENABLE\_CLR register to disable a channel.

1 - Enables channels [Channel-1]

### DMA channel enable clear register (DMA\_CHNL\_ENABLE\_CLR)

Address offset : 0x02c

Reset value : -

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															
															WO

[Channel-1] CHNL\_ENABLE\_CLR - Set the appropriate bit to disable the corresponding DMA channel.

This write only register disable a DMA channel.

0 : No effect. Use the CHNL\_ENABLE\_SET register to enable DMA channel.

1 : Disable channel [Channel-1]

### DMA channel primary-alternate set register (DMA\_CHNL\_PRI\_ALT\_SET)

Address offset : 0x030

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															
															R/W

[Channel-1] CHNL\_PRI\_ALT\_SET - Returns the channel control data structure status, or selects the alternate data structure for the corresponding DMA channels.

This read/write register configures a DMA channel to use the alternate data structure. Reading the register returns the status of which data structure is in use for the corresponding DMA channel.

Read as :

0 - DMA Channel [Channel-1] is using the primary data structure.

1 - DMA Channel [Channel-1] is using the alternate data structure.

Write as :

0 - No effect. Use the CHNL\_PRI\_ALT\_CLR register to set bit [Channel-1] to 0

1 - Selects the alternate data structure for channel [Channel-1]

### DMA channel primary-alternate clear register (DMA\_CHNL\_PRI\_ALT\_CLR)

Address offset : 0x034

Reset value : -

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															
															CHNL_PRI_ALT_CLR[5:0]
															WO

[Channel-1] CHNL\_PRI\_ALT\_CLR - Set the appropriate bit to select the primary data structure for the corresponding DMA channel.

This write only register configures a DMA channels to use the primary data structure.

0 : No effect. Use the CHNL\_PRI\_ALT\_SET register to select the alternate data structure.

1 : Selects the primary data structure for channel [Channel-1]

### DMA channel priority set register (DMA\_CHNL\_PRIORITY\_SET)

Address offset : 0x038

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

res	CHNL_PRIORITY_SET [5:0]									
										R/W

[Channel-1] CHNL\_PRIORITY\_SET - Returns the channel priority mask status, or set the channel priority to high.

This read/write register configures a DMA channel to use the high priority level. Reading the register returns the status of channel priority mask

Read as :

0 - DMA Channel [Channel-1] is using the default priority level.

1 - DMA Channel [Channel-1] is using a high priority level.

Write as :

0 - No effect. Use the CHNL\_PRIORITY\_CLR register to set bit [Channel-1] to default priority level

1 - Channel [Channel-1] uses the high priority level.

### DMA channel priority clear register (DMA\_CHNL\_PRIORITY\_CLR)

Address offset : 0x03C

Reset value : -

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															
															CHNL_PRIORITY_CLR[5:0]
															WO

[Channel-1] CHNL\_PRIORITY\_CLR - Set the appropriate bit to select the default priority level for the specified DMA channel.

This write only register configures a DMA channels to use the default priority level.

0 : No effect. Use the CHNL\_PRIORITY\_SET register to set channel [Channel-1] to the high priority level.

1 : channel [Channel-1] uses the default priority level.

### DMA bus error clear register (DMA\_ERR\_CLR)

Address offset : 0x04C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

res																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
res	ERR_CLR															
															R/W	

[0] ERR\_CLR - Returns the status of DMA\_ERR, or set the signal LOW.

This read/write register returns the status of DMA\_ERR, and enables set DMA\_ERR LOW.

Read as :

0 : DMA\_ERR is LOW

1 : DMA\_ERR is HIGH

Write as :

0 : No effect, status of DMA\_ERR is unchanged.

1 : Sets DMA\_ERR LOW.

## 19.5 Register map

The following Table 15 summarizes the DMA registers.

Table 15 DMA register map and reset values

## 20 Analog-to-digital converter (ADC)

### 20.1 Introduction

ADC is a 12bit analog-to-digital converter. It has up to 9 multiplexed channels allowing it to measure signals from 8 externals and 1 internal source.

ADC of various channels can be performed in single mode. The result of the ADC is stored in 12 bit register.

### 20.2 Features

- 12bit configuration resolution
- Conversion time : Max 10MHz (Sampling time can be programmable)
  - 8 channel for external analog inputs
    - CH0 : PC\_15
    - CH1 : PC\_14
    - CH2 : PC\_13
    - CH3 : PC\_12
    - CH4 : PC\_11
    - CH5 : PC\_10
    - CH6 : PC\_09
    - CH7 : PC\_08
  - 1 channel for internal LDO(1.5v) voltage.
    - CH15 : Internal voltage
- Start of conversion can be initiated by software.
- Convert selected inputs once per trigger.
- Interrupt generation at the end of conversion.

## 20.3 Functional description

Figure 19 shows the ADC block diagram.

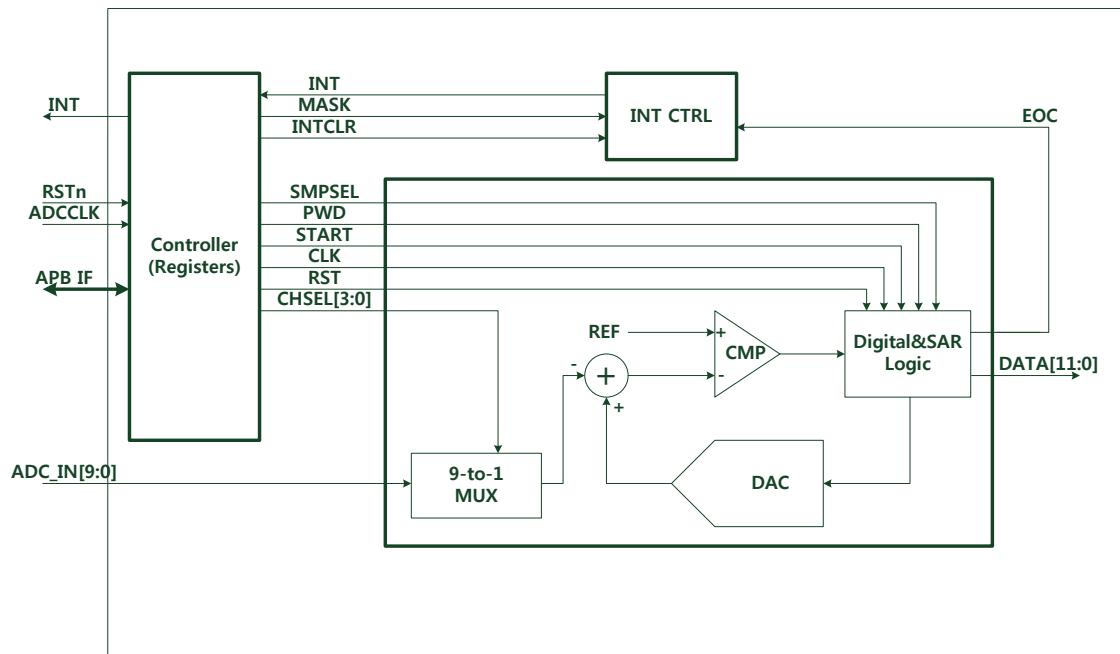


Figure 19. ADC block diagram

### Operation ADC with non-interrupt

Figure 20 shows the flowchart of ADC operation with non-interrupt.

ADC can be used as below:

1. ADC needs to be initialized before operation.  
To initialize the ADC, clear the PWD bit first.
2. Select the ADC channel from 0 to 7 and 15 (initial core voltage).
3. Run start ADC conversion by set ADC\_SRT bit.
4. Check INT bit to know finish of conversion.
5. If INT bit is high (1), read ADC conversion data.
6. Finally, ADC operation is finished by setting the PWD bit.

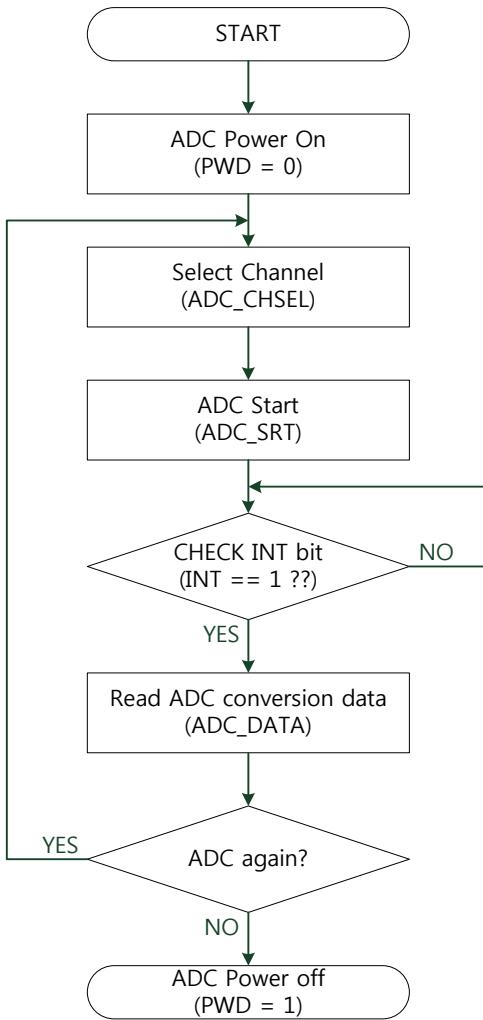


Figure 20. The ADC operation flowchart with non-interrupt

## Operation ADC with interrupt

Figure 21 shows the flowchart of ADC operation with interrupt.

Operation is almost the same as the non-interrupt mode except checking INT register bit to know when enabling interrupt mask bit and conversion is completed.

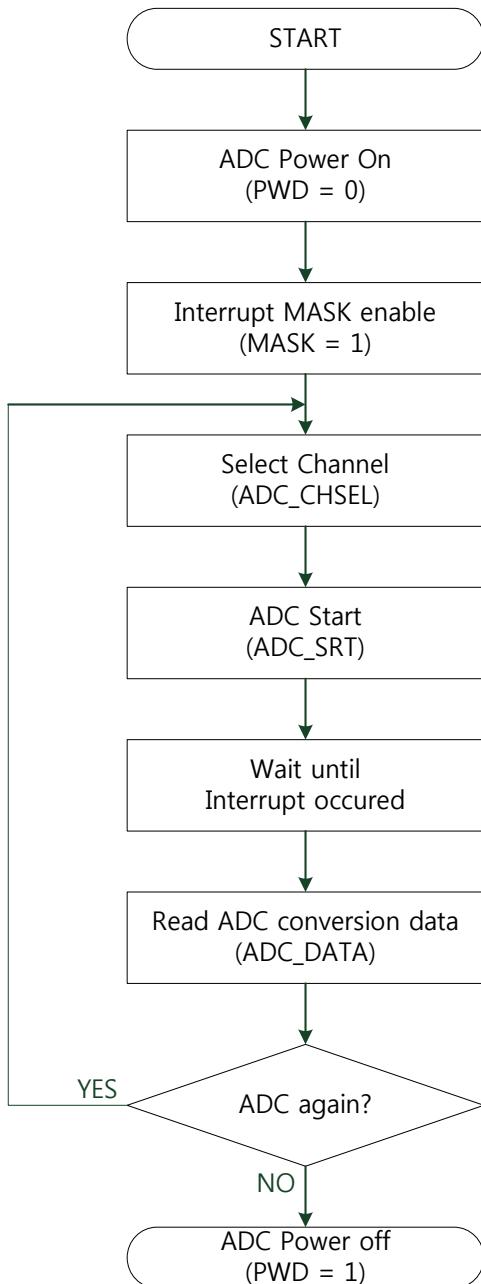


Figure 21. The ADC operation flowchart with interrupt

## 20.4 Registers (Base address : 0x4100\_0000)

### ADC control register (ADC\_CTR)

Address offset : 0x000

Reset value : 0x0000\_0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PWD	SMPSEL													
														R/W	R/W

[0] SMPSEL - Sampling mode select

This bit written by S/W to select sampling mode

0 : Abnormal mode

1 : Normal mode

[1] PWD - Power down

This bit set and cleared by S/W to enable/disable power down mode

0 : Active

1 : Power down

## ADC channel select register (ADC\_CHSEL)

Address offset : 0x004

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CHSEL														
														R/W	

[3:0] CHSEL - ADC Channel Select

These bits are written by S/W to define which channel is selected to be converted.

0000 : Channel 0 select

0001 : Channel 1 select

0010 : Channel 2 select

0011 : Channel 3 select

0100 : Channel 4 select

0101 : Channel 5 select

0110 : Channel 6 select

0111 : Channel 7 select

1000 ~ 1000 : no select  
 1111 : LDO output(1.5V) select

### ADC start register (ADC\_START)

Address offset : 0x008

Reset value : 0x0000\_0000

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

[0] ADC\_SRT - ADC Start bit

This bit set by S/W to start ADC for conversion. This bit is write-only.

0 : ready to start

1 : start ADC for conversion (This bit clear automatically after conversion)

### ADC conversion data register (ADC\_DATA)

Address offset : 0x00C

Reset value : 0x0000\_0000

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

[11:0] ADC\_DATA - ADC conversion result data

It contains ADC conversion result data of last converted channel. These bits are read-only.

### ADC Interrupt register (ADC\_INT)

Address offset : 0x010

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	MASK	INT													
														R/W	R

[0] DONE - Interrupt bit

This bit indicates that conversion is done or not. This bit is set after conversion is done and this bit is cleared by set of Interrupt clear bit. This bit is read-only.

[1] MASK - Interrupt mask signal.

This bit is interrupt mask bit of ADC. This bit can be set and cleared by S/W to enable/disable interrupt mask.

0 : Interrupt disable

1 : Interrupt enable

### ADC Interrupt Clear register (ADC\_INTCLR)

Address offset : 0x01c

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	INTCLR														
															w

[0] INTCLR - Interrupt Clear bit.

This bit set by S/W to clear interrupt signal to CM0. This bit is write-only.

0 - nothing

1 - Clear interrupt signal (This bit clear automatically after clear interrupt)

## 20.5 Register map

The following Table 16 summarizes the ADC registers.

Table 16 ADC register map and reset values

## 21 Pulse-Width Modulation (PWM)

### 21.1 Introduction

The PWM consists a 8-channel 32-bit Timer/Counter driven by a programmable prescaler. The function of the PWM is based on the basic Timer. Each timer and counter runs independently.

The PWM can be used to control the width of the pulse, formally the pulse duration, to generate output waveform or to count the counter triggered by external input.

### 21.2 Features

- Counter or Timer operation can use the peripheral clock, external clock source, or one of the capture inputs as the clock source.
- Eight independent 32-bit Timer/Counter driven by a programmable 6 bits prescaler runs as the PWM or standard timer if the PWM mode is not enabled.
- Eight PWM output waveforms.
- Each of Timer/Counter can have different or same clock source.
- Counter or timer operation.
- Eight capture registers that can take the timer value when an external input signal. A capture event can generate an interrupt signal optionally.
- 32-bit match register and limit register.

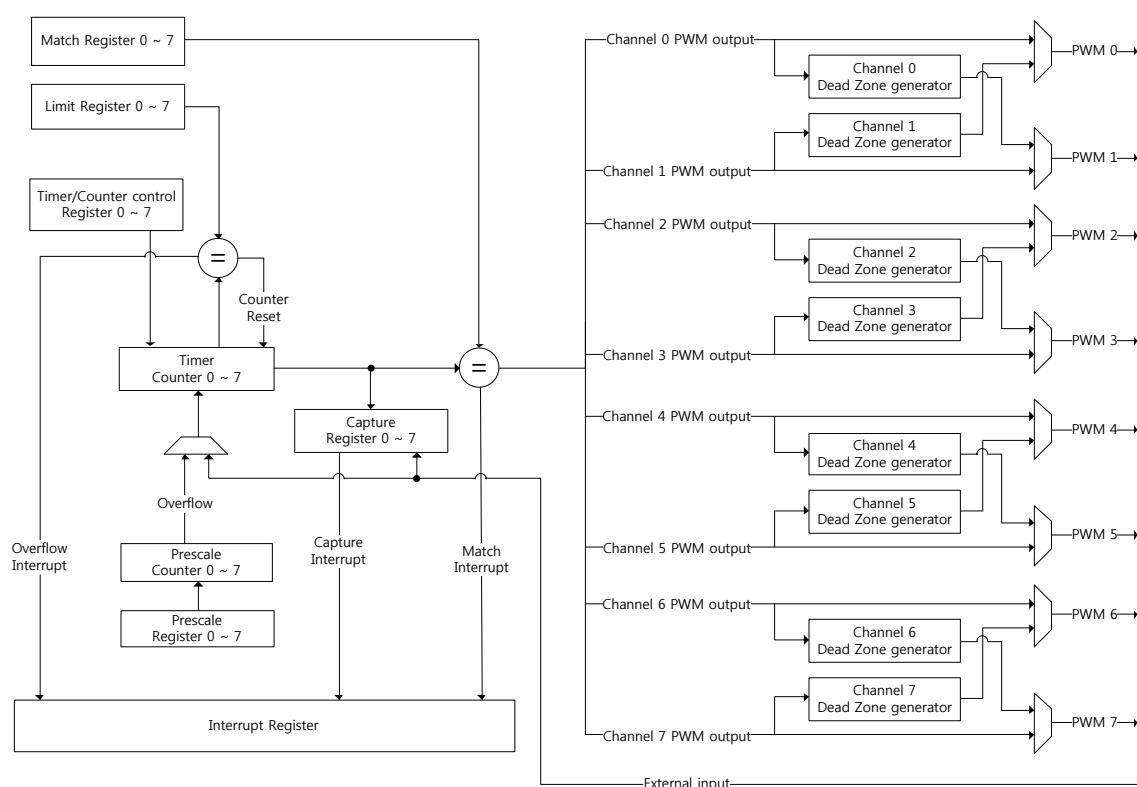


Figure 22 PWM block diagram

## 21.3 Functional description

### Timer/Counter control

The PWM has Start/Stop register. It controls start or stop of the Timer/Counter. If you set this register, the Timer/Counter starts to run. If you reset this register, the Timer/Counter stops immediately. Also there is a pause register. The pause register is used to stop temporarily after one period. Although you set this register while the Timer/Counter is running, the Timer/Counter will stop when the period ends.

The registers of PWM can be updated when it stops or pauses. Users cannot update the registers while PWM is running.

### Timer/Counter

The PWM has 8 Timer/Counter clocks, which can be divided by a prescaler. Each Timer/Counter runs independently. The Timer/Counter is designed to count cycles of the clocks or external input signal and generate interrupts when specified timer values are occurred based on match register and limit register. The Timer/Counter can count up or down.

The PWM has match registers and limit registers. The match registers control the duty cycle of PWM output waveform. The limit registers control the period of the PWM output waveform. The Timer/Counter becomes 0 when it reaches value of the limit register. If PDMR(Periodic Mode Register) is set, the Timer/Counter counts stops counting and if PDMR is reset, the Timer/Counter counts repeatedly.

Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

### Repetition mode

The Timer/Counter has two repetition mode: periodic and one-shot mode. In periodic mode, the Timer/Counter recycles and then restarts when the Timer/Counter reaches the value of limit register. Figure 23 shows periodic mode timing diagram.

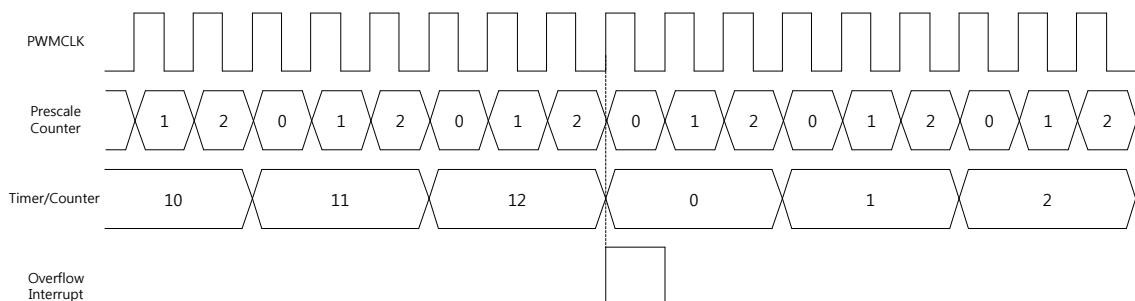


Figure 23 Periodic mode

In one-shot mode, the Timer /Counter reset to the initial value and then stops when the Timer/Counter reaches the value of limit register. Figure 24 shows one-shot mode timing diagram.

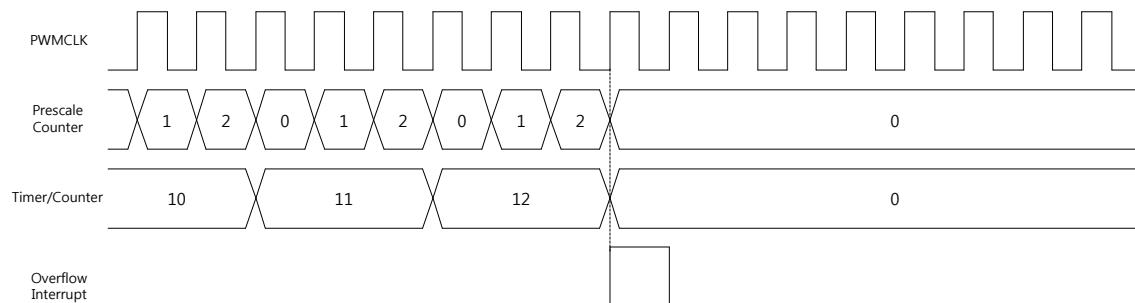


Figure 24 one-shot mode

## Counting mode

The Timer/Counter has two counting mode: Up-count and Down-count mode. In up-count mode, the Timer/Counter counts up from 0 to the limit register value, then recycles. If repetition mode is periodic, the Timer/Counter restarts, if repetition mode is one-shot mode, the Timer/Counter stops. Figure 25 shows up-count mode timing diagram.

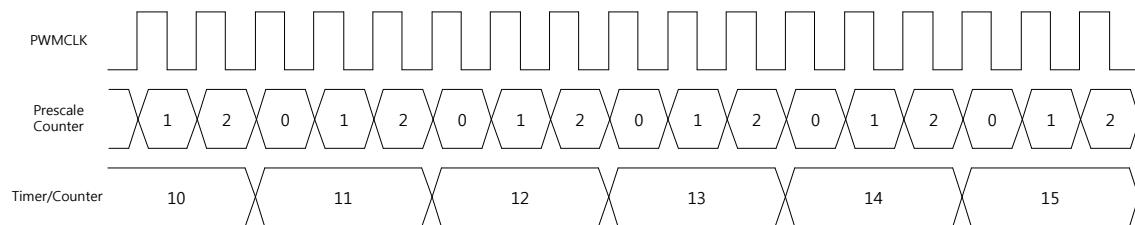


Figure 25 Up-count mode

In Down-count mode, the Timer/Counter counts from 0xFFFF\_FFFF, then recycles. If repetition mode is periodic, the Timer/Counter restarts, if repetition mode is one-shot mode, the Timer/Counter stops. Figure 26 shows down-count mode timing diagram.

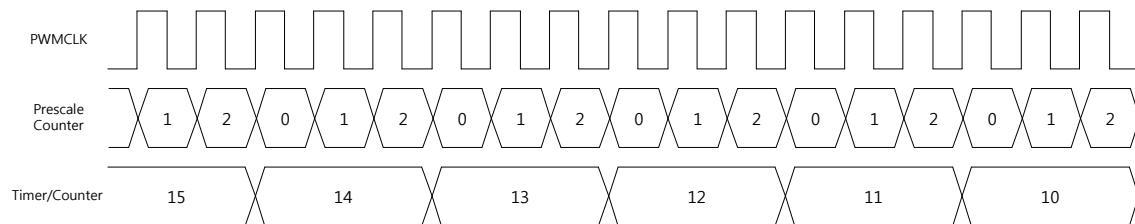


Figure 26 Down-count mode

## Timer and Counter mode

The Timer/Counter can run in timer mode or counter mode. In timer mode, the Timer/Counter is counted by PWMCLK after Prescale counter is overflowed. If prescale is set by 0, the Timer/Counter counts every PWMCLK period. In counter mode, the Timer/Counter is counted by external input signal. There are three counting method: rising edge, falling edge, and both edge. The counter mode has up-count or down-count mode and also has periodic or one-shot mode. The external input pin and PWM output pin are the same, so PWM output is disabled in counter mode.

Figure 27 is counter mode example with rising edge mode,

Figure 28 is with falling edge mode and

Figure 29 is with both rising and falling edge mode.

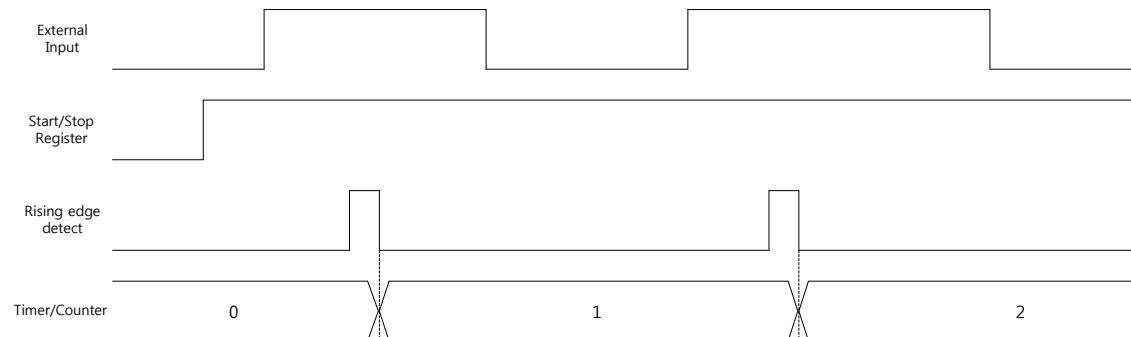


Figure 27 Counter mode with rising edge

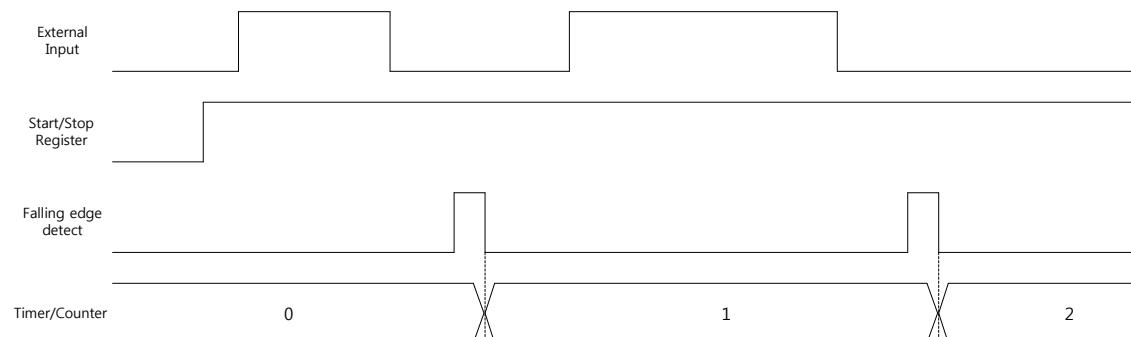


Figure 28 Counter mode with falling edge

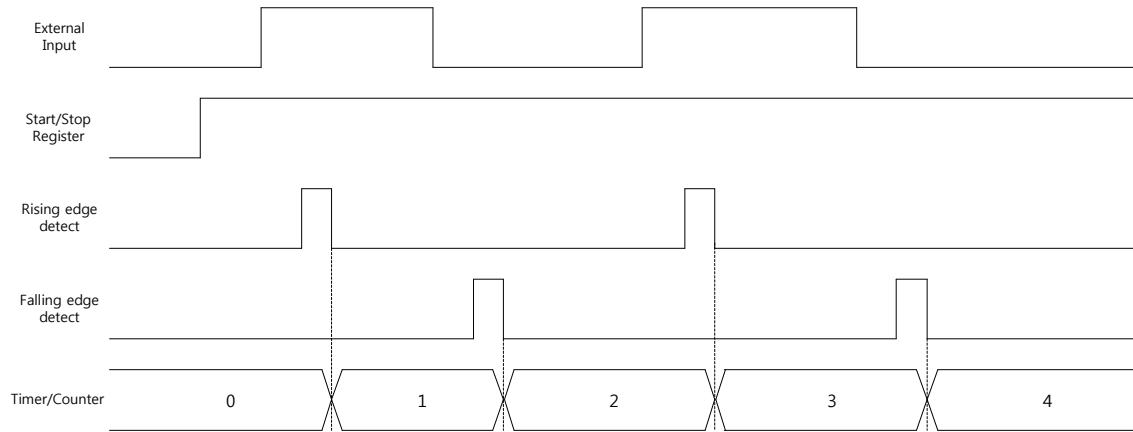


Figure 29 Counter mode with rising and falling edge

## Prescaler description

The PWM has 6-bit prescale counter(PC) and the prescaler can divide the Timer/Counter clock frequency. Users can control it by Prescale Register(PR).

Figure 30 and Figure 31 shows some examples of the Timer/Counter timing with prescale register is 2, match register is 2, limit register is 12, timer mode, periodic mode, up-count mode, and no interrupt clear.

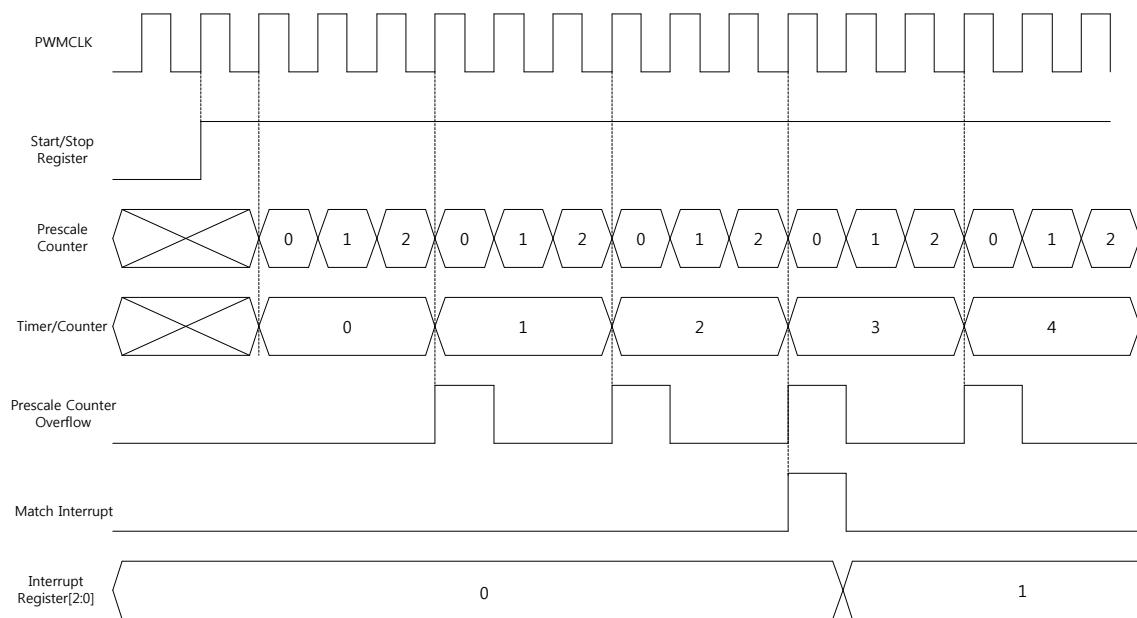


Figure 30 Timer/Counter timing diagram with match interrupt

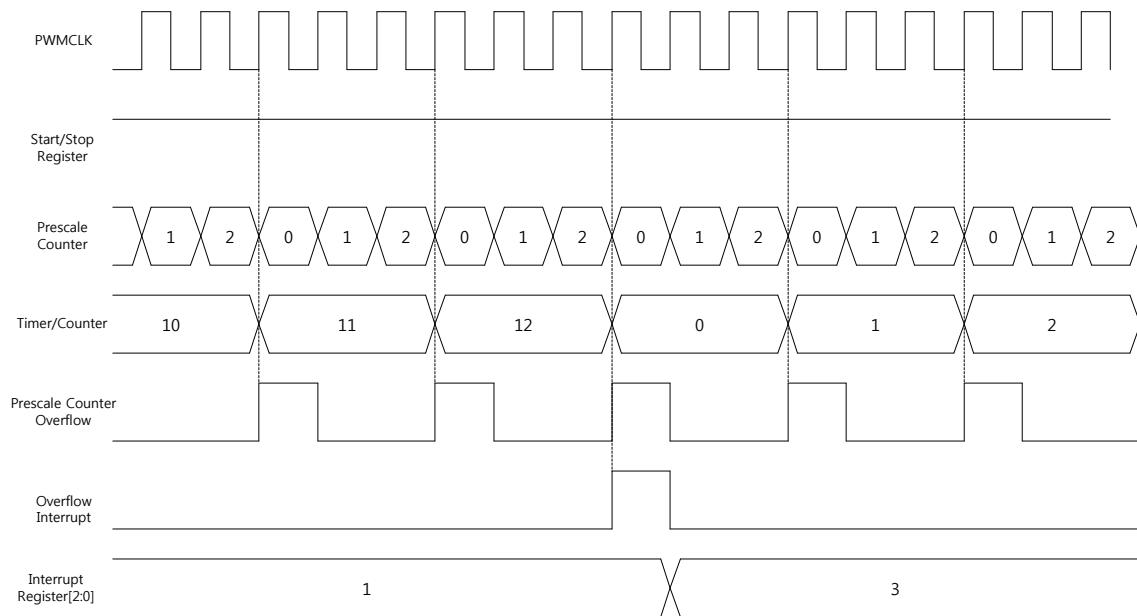


Figure 31 Timer/Counter timing diagram with overflow interrupt

## PWM mode

Pulse Width Modulation mode generates a waveform with a period determined by the value of limit register and a duty cycle determined by the value of the match register.

The PWM output becomes always 1 when the Timer/Counter starts to count. Then the PWM output becomes 0 when the Timer/Counter reaches the value of match register. If the Timer/Counter is in periodic mode, the PWM output becomes 1 again when the Timer/Counter reaches the value of limit register. In one-shot mode, the PWM output does not change to 1 but stays 0 and the Timer/Counter stops.

The PWM mode can be selected independently on each channel(0~7) by PWM output enable and external input enable register. The external input pin and PWM output pin are the same, so external input is disabled in PWM mode.

Figure 32 is an example of the PWM output waveform when the Timer/Counter is reached to the value of match register.

Figure 33 is example of the PWM output waveform when to the Timer/Counter is reached to the value of limit register.

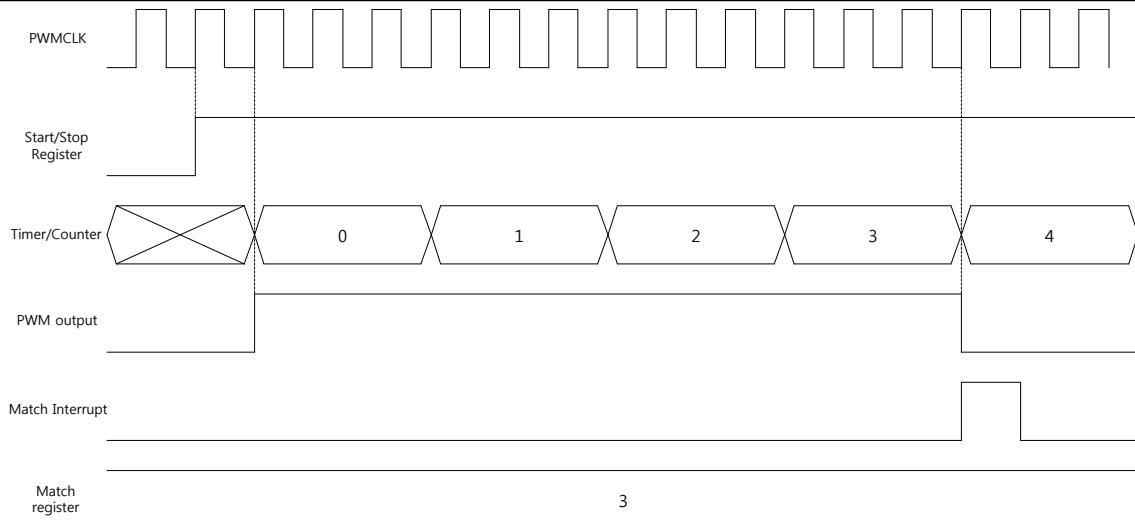


Figure 32 The PWM output up to match register

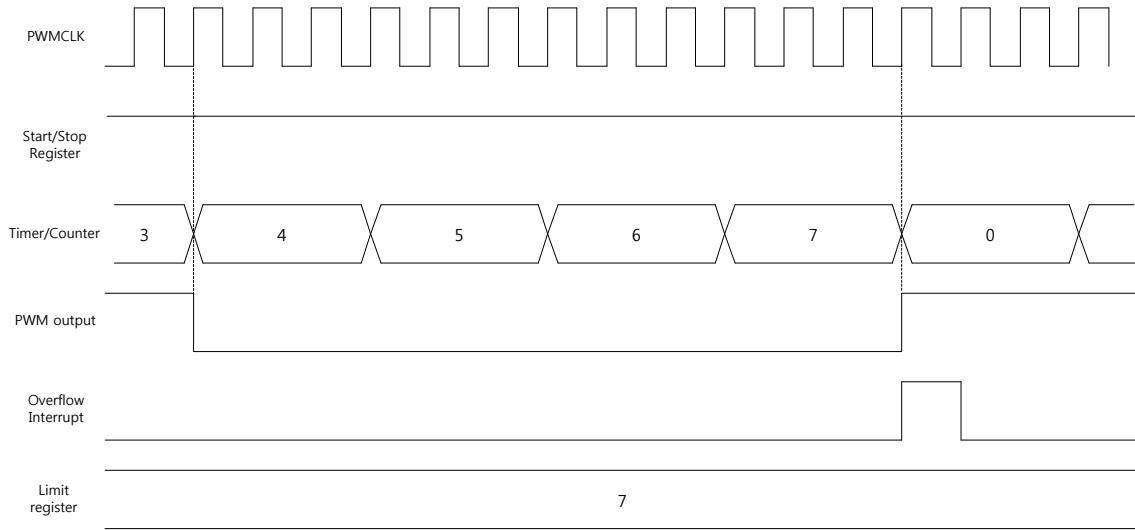


Figure 33 The PWM output up to limit register

If match register is set as 0, the PWM output will be 1 while the Timer/Counter is 0.

If the match register is bigger than the limit register, the PWM output is always 1.

## Interrupt

The PWM has 8-bit interrupt enable register(IER) and each bit of IER corresponds to each interrupt of channel. Each PWM channel has Channel-n Interrupt Enable register(CHn\_IER). The CHn\_IER includes three types of interrupt: match, overflow, and capture. The match interrupt occurs when the Timer/Counter is reached to value of match register. The overflow interrupt occurs when the Timer/Counter is reached to value of limit register. The capture interrupt occurs when external input is entered for capture.

If interrupt occurs, corresponded bit of Channel-x interrupt register(CHn\_IR) bit is set and PWM channel-n interrupt signal is generated. All CHn\_IR is cleared by channel-n interrupt clear register(CHn\_ICR) and then PWM channel-n interrupt signal is cleared.

## Dead zone generation

Each PWM channel can output two complementary signals with dead zone time and it can be enabled by Channel-n Dead Zone Enable Register(CHn\_DZER). Only 4 channels can be enabled because there are 8 PWM output pins. Channel 0 and 1 are a pair, channel 2 and 3 are a pair, channel 4 and 5 are a pair, and channel 6 and 7 are a pair. If users want to use channel-0 dead zone generation, channel-1 should be disabled. If channel 0 and 1 dead zone generation are enabled both, all outputs are 0. In that case, users should choose 1 channel.

Dead zone time are generated by the value of Channel-n Dead Zone Counter Register(DZCR). The dead zone counter counts up to value of DZCR. During the dead zone time, both complementary signals are both 0. Users have to adjust the signal depending on the devices that are connected to the outputs and their characteristics. If DZCR is bigger than the limit register, main output signal is toggled 0 to 1 and then 1 to 0 while 1 PWMCLK and inverted output signal is always 0.

Figure 34 shows two complementary PWM outputs with dead zone time. During dead zone time, both outputs are 0. Figure 35 shows a more detailed timing with dead zone counter. The dead zone counter and the Timer/Counter starts to count together and PWM output is 0 until dead zone counter is reached to value of dead zone counter register. The PWM output becomes 1 and 0 when the Timer/Counter is reached to value of match register. The inverted PWM output is also 0 until dead zone counter is reached to value of dead zone counter register. Then inverted PWM output becomes 1 after dead zone counter is reached to the value of dead zone counter register.

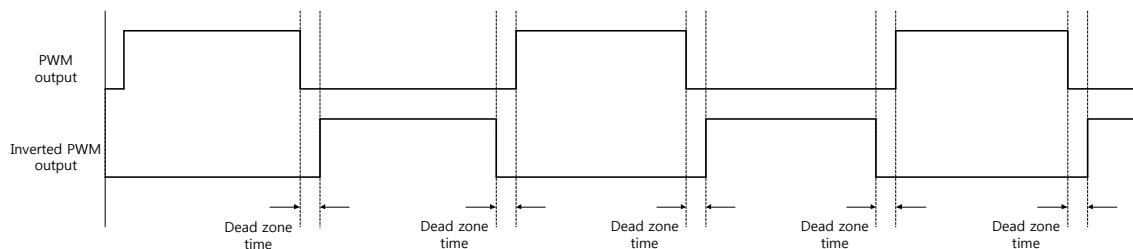


Figure 34 PWM waveform with dead zone time

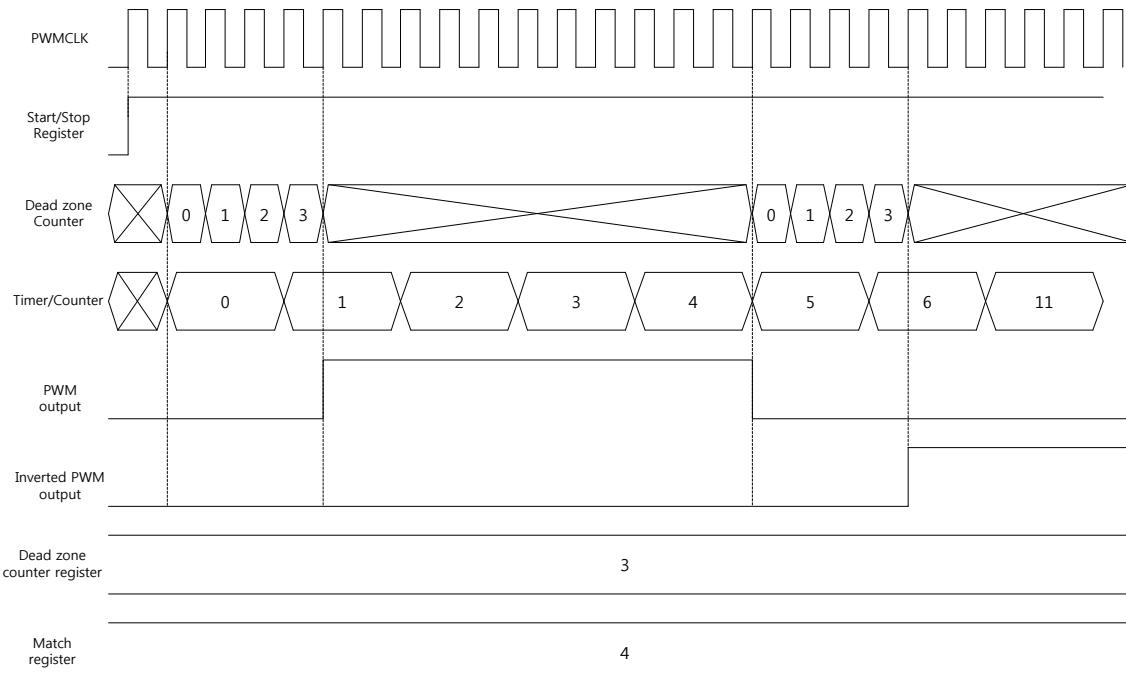


Figure 35 PWM waveform with dead zone counter

## Capture event

Each PWM channel can capture its Timer/Counter value when an external input signal changes. Any channel could use any method of rising or falling edges. If capture interrupt is enabled, capture interrupt occurs when the external input signal is toggled. The Timer/Counter value is saved in Channel-n Capture Register(CHn\_CR) and the capture register is not overwritten until capture interrupt is cleared. Figure 36 shows the capture event timing diagram. There is no interrupt clear, so second capture does not save during second rising edge detection.

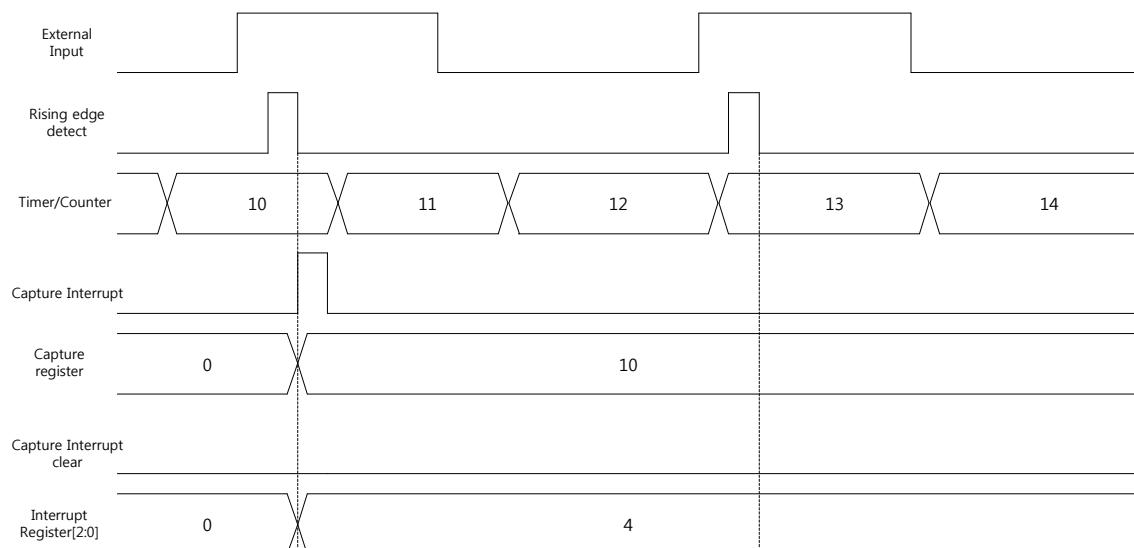


Figure 36 Capture event with no interrupt clear

Figure 37 shows, also, capture event timing diagram with interrupt clear. The second capture is saved at the second rising edge detection because there is interrupt clear.

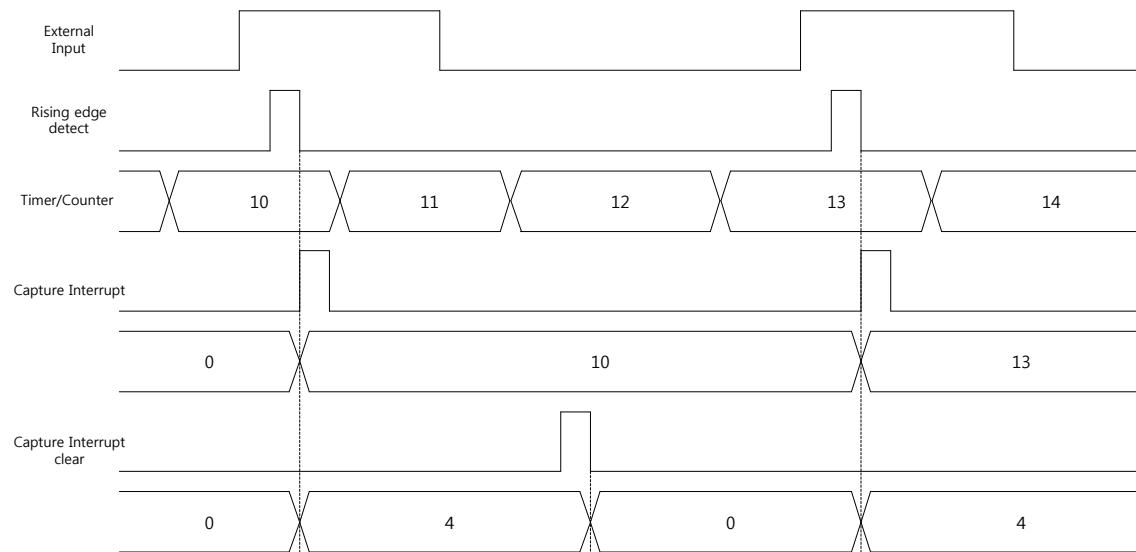


Figure 37 Capture event with interrupt clear

## How to set the PWM

Figure 38 shows the PWM setting flow step by step.

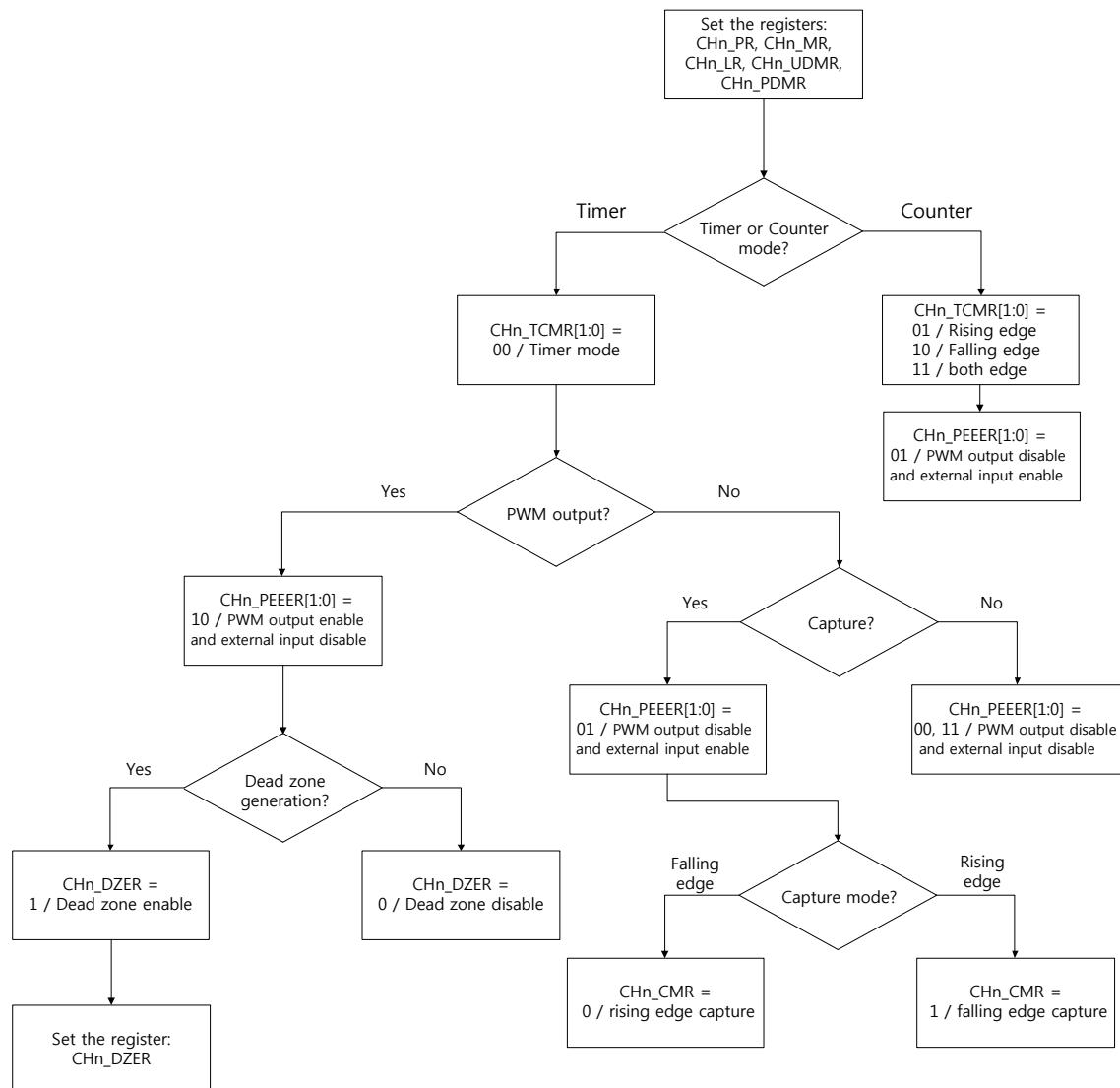


Figure 38 The PWM setting flow

## 21.4 PWM Channel-0 Registers (Base address : 0x4000\_5000)

### Channel-0 interrupt register(PWMCH0IR)

Base address : 0x4000\_5000

Address offset : 0x000

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CI	OI	MI												
													R	R	R

#### [0] MI - Match Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Match interrupt does not occur.

1 : Match interrupt occurs.

#### [1] OI - Overflow Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Overflow interrupt does not occur.

1 : Overflow interrupt occurs.

#### [2] CI - Capture Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Capture interrupt does not occur.

1 : Capture interrupt occurs.

## Channel-0 interrupt enable register(PWMCH0IER)

Base address : 0x4000\_5000

Address offset : 0x04

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIE	OIE	MIE												
													R/W	R/W	R/W

#### [0] MIE - Match Interrupt Enabled.

0 : Match interrupt is not enabled.

1 : Match interrupt is enabled.

#### [1] OIE - Overflow Interrupt Enable.

0 : Overflow interrupt is not enabled.

1 : Overflow interrupt is enabled.

#### [2] CIE - Capture Interrupt Enable.

0 : Capture interrupt is not enabled.

1 : Capture interrupt is enabled.

## Channel-0 interrupt clear register(PWMCH0ICR)

Base address : 0x4000\_5000

Address offset : 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIC	OIC	MIC												
													W	W	W

This bit is set by software, cleared by hardware when a capture interrupt becomes 0.

[0] MIC - Match Interrupt

0 : No action.

1 : Match interrupt is cleared.

[1] OIC - Overflow Interrupt

0 : No action.

1 : Overflow Interrupt is cleared.

[2] CIC - Capture Interrupt Clear.

0 : No action.

1 : Capture Interrupt is cleared.

## Channel-0 Timer/Counter Register (PWMCH0TCR)

Base address : 0x4000\_5000

Address offset : 0x0C

Reset value : 0x0000\_0000

31	0
TCR	
R	

[31:0] TCR - Timer/Counter register

Timer/Counter register. These register hold the current values of the Timer/Counter(TC). The TC is incremented every PR cycles. When the TC is reached to value of match register, the match interrupt is occurred and PWM output waveform becomes 0. When the TC is reached to the value of limit register, the overflow interrupt is occurred, the TC is reset as 0.

## Channel-0 Prescale Counter Register (PWMCHOPCR)

Base address : 0x4000\_5000

Address offset : 0x10

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PCR														
															R

### [5:0] PC - Prescale Counter register

Prescale Counter register. These registers hold the current values of prescale counter(PC). The PC is incremented to the value stored in PR. When the PC is reached to PR, the TC is incremented and the PC is reset as 0.

## Channel-0 Prescale Register (PWMCH0PR)

Base address : 0x4000\_5000

Address offset : 0x14

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PR														
															R/W

### [5 - 0] PR - Prescale Register

Prescale register. The PC is incremented when the PC is reached to the PR.

## Channel-0 Match Register (PWMCH0MR)

Base address : 0x4000\_5000

Address offset : 0x18

Reset value : 0x0000\_0000

31

0

MR

R/W

### [31:0] MR - Match Register

Match register. The MR can generate a match interrupt and PWM output waveform becomes 0 when the TC is reached to the MR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

## Channel-0 Limit Register (PWMCH0LR)

Base address : 0x4000\_5000

Address offset : 0x1C

Reset value : 0x0000\_0000

31

0

LR

R/W

### [31:0] LR - Limit Register

Limit Register. The LR can generate an overflow interrupt and PWM output waveform becomes 1 when the TC is reached to the LR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

## Channel-0 Up-Down Mode Register (PWMCH0UDMR)

Base address : 0x4000\_5000

Address offset : 0x20

Reset value : 0x0000\_0000

31

30

29

28

27

26

25

24

23

22

21

20

19

18

17

16

res

15

14

13

12

11

10

9

8

7

6

5

4

3

2

1

0

res

UDM

R/W

### [0] UDM - Up-Down mode

0 : TC runs up count.

1 : TC runs down count.

### Channel-0 Timer/Counter Mode Register (PWMCH0TCMR)

Base address : 0x4000\_5000

Address offset : 0x24

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TCM														
															R/W

[1:0] TCM - Timer/Counter mode

00 : Timer mode.

01 : Counter mode with counting driven by rising edge external input .

10 : Counter mode with counting driven by falling edge external input.

11 : Counter mode with counting driven by rising and falling edge external input.

### Channel-0 PWM output Enable and External input Enable Register (PWMCH0PEEEER)

Base address : 0x4000\_5000

Address offset : 0x28

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PEEE														
															R/W

[1:0] PEEE - PWM output Enable and External input Enable

00 : PWM output is disable and external input is disable.

01 : PWM output is disable and external input is enable.

10 : PWM output is enable and external input is disable.

## Channel-0 Capture Mode Register (PWMCH0CMR)

Base address : 0x4000\_5000

Address offset : 0x2C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CM														
															R/W

[0] CM - Capture mode

0 : Timer/Counter is captured when external input signal is rising edge.

1 : Timer/Counter is captured when external input signal is falling edge.

## Channel-0 Capture Register (PWMCH0CR)

Base address : 0x4000\_5000

Address offset : 0x30

Reset value : 0x0000\_0000

31	0
CR	
R	

[31:0] CR - Capture Register

Capture register. The CR is loaded with the value of the TC when external input signal is triggered.

## Channel-0 Periodic Mode Register (PWMCH0PDMR)

Base address : 0x4000\_5000

Address offset : 0x34

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PDM														
															R/W

#### [0] PDM - Periodic Mode

0 : Periodic mode. When the TC is reached to the LR, the TC returns to 0 and then continues counting periodically.

1 : One-shot mode. When the TC is reached to the LR, the TC returns to 0 and then stops counting.

### Channel-0 Dead Zone Enable Register (PWMCH0DZER)

Base address : 0x4000\_5000

Address offset : 0x38

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	DZE														
															R/W

#### [0] DZE - Dead Zone Enable

0 : Dead zone generation is disabled.

1 : Dead zone generation is enabled.

### Channel-0 Dead Zone Counter Register (PWMCH0DZCR)

Base address : 0x4000\_5000

Address offset : 0x3C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	res	res										DZC

						R/W
--	--	--	--	--	--	-----

#### [9:0] DZC - Dead Zone Counter value

Dead zone generation counter value register. If the DZE bit in DZER is 1, dead zone counter counts to this value and during this time, the two PWM output waveforms are all 0.

## 21.5 Register map

The following Table 17 summarizes the PWM Channel-0 registers.

Table 17 PWM channel 0 register map and reset values

## 21.6 PWM Channel-1 Registers (Base address : 0x4000\_5100)

### Channel-1 interrupt register(PWMCH1IR)

Base address : 0x4000\_5100

Address offset : 0x00

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CI	OI													
													R	R	R

#### [0] MI - Match Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Match interrupt does not occur.

1 : Match interrupt occurs.

#### [1] OI - Overflow Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Overflow interrupt does not occur.

1 : Overflow interrupt occurs.

#### [2] CI - Capture Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Capture interrupt does not occur.

1 : Capture interrupt occurs.

### Channel-1 interrupt enable register(PWMCH1IER)

Base address : 0x4000\_5100

Address offset : 0x04

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIE	OIE	MIE												
													R/W	R/W	R/W

[0] MIE - Match Interrupt Enabled.

0 : Match interrupt is not enabled.

1 : Match interrupt is enabled.

[1] OIE - Overflow Interrupt Enable.

0 : Overflow interrupt is not enabled.

1 : Overflow interrupt is enabled.

[2] CIE - Capture Interrupt Enable.

0 : Capture interrupt is not enabled.

1 : Capture interrupt is enabled.

## Channel-1 interrupt clear register(PWMCH1ICR)

Base address : 0x4000\_5100

Address offset : 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIC	OIC	MIC												
													W	W	W

This bit is set by software, cleared by hardware when a capture interrupt becomes 0.

[0] MIC - Match Interrupt

0 : No action.

1 : Match interrupt is cleared.

[1] OIC - Overflow Interrupt

0 : No action.

1 : Overflow Interrupt is cleared.

[2] CIC - Capture Interrupt Clear.

0 : No action.

1 : Capture Interrupt is cleared.

## Channel-1 Timer/Counter Register (PWMCH1TCR)

Base address : 0x4000\_5100

Address offset : 0x0C

Reset value : 0x0000\_0000

31

0

TCR

R

### [31:0] TCR - Timer/Counter register

Timer/Counter register. These register hold the current values of the Timer/Counter(TC). The TC is incremented every PR cycles. When the TC is reached to value of match register, the match interrupt is occurred and PWM output waveform becomes 0. When the TC is reached to the value of limit register, the overflow interrupt is occurred, the TC is reset as 0.

## Channel-1 Prescale Counter Register (PWMCH1PCR)

Base address : 0x4000\_5100

Address offset : 0x10

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res					PCR										
															R

### [5:0] PC - Prescale Counter register

Prescale Counter register. These registers hold the current values of prescale counter(PC). The PC is incremented to the value stored in PR. When the PC is reached to PR, the TC is incremented and the PC is reset as 0.

## Channel-1 Prescale Register (PWMCH1PR)

Base address : 0x4000\_5100

Address offset : 0x14

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res		PR													
												R/W			

#### [5: 0] PR - Prescale Register

Prescale register. The PC is incremented when the PC is reached to the PR.

### Channel-1 Match Register (PWMCH1MR)

Base address : 0x4000\_5100

Address offset : 0x18

Reset value : 0x0000\_0000

31	0
MR	
R/W	

#### [31:0] MR - Match Register

Match register. The MR can generate a match interrupt and PWM output waveform becomes 0 when the TC is reached to the MR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

### Channel-1 Limit Register (PWMCH1LR)

Base address : 0x4000\_5100

Address offset : 0x1C

Reset value : 0x0000\_0000

31	0
LR	
R/W	

#### [31:0] LR - Limit Register

Limit Register. The LR can generate an overflow interrupt and PWM output waveform becomes 1 when the TC is reached to the LR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

## Channel-1 Up-Down Mode Register (PWMCH1UDMR)

Base address : 0x4000\_5100

Address offset : 0x20

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	UDM														
															R/W

[0] UDM - Up-Down mode

0 : TC runs up count.

1 : TC runs down count.

## Channel-1 Timer/Counter Mode Register (PWMCH1TCMR)

Base address : 0x4000\_5100

Address offset : 0x24

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TCM														
															R/W

[1:0] TCM - Timer/Counter mode

00 : Timer mode.

01 : Counter mode with counting driven by rising edge external input .

10 : Counter mode with counting driven by falling edge external input.

11 : Counter mode with counting driven by rising and falling edge external input.

## Channel-1 PWM output Enable and External input Enable Register (PWMCH1PEEER)

Base address : 0x4000\_5100

Address offset : 0x28

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PEEE														
															R/W

[1:0] PEEE - PWM output Enable and External input Enable

00 : PWM output is disable and external input is disable.

01 : PWM output is disable and external input is enable.

10 : PWM output is enable and external input is disable.

## Channel-1 Capture Mode Register (PWMCH1CMR)

Base address : 0x4000\_5100

Address offset : 0x2C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CM														
															R/W

[0] CM - Capture mode

0 : Timer/Counter is captured when external input signal is rising edge.

1 : Timer/Counter is captured when external input signal is falling edge.

## Channel-1 Capture Register (PWMCH1CR)

Base address : 0x4000\_5100

Address offset : 0x30

Reset value : 0x0000\_0000

31	0
CR	
R	

#### [31:0] CR - Capture Register

Capture register. The CR is loaded with the value of the TC when external input signal is triggered.

### Channel-1 Periodic Mode Register (PWMCH1PDMR)

Base address : 0x4000\_5100

Address offset : 0x34

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PDM														
															R/W

#### [0] PDM - Periodic Mode

0 : Periodic mode. When the TC is reached to the LR, the TC returns to 0 and then continues counting periodically.

1 : One-shot mode. When the TC is reached to the LR, the TC returns to 0 and then stops counting.

### Channel-1 Dead Zone Enable Register (PWMCH1DZER)

Base address : 0x4000\_5100

Address offset : 0x38

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	DZE														
															R/W

---

**[0] DZE - Dead Zone Enable**

0 : Dead zone generation is disabled.

1 : Dead zone generation is enabled.

## Channel-1 Dead Zone Counter Register (PWMCH1DZCR)

Base address : 0x4000\_5100

Address offset : 0x3C

Reset value : 0x0000\_0000

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

**[9:0] DZC - Dead Zone Counter value**

Dead zone generation counter value register. If the DZE bit in DZER is 1, dead zone counter counts to this value and during this time, the two PWM output waveforms are all 0.

## 21.7 Register map

The following Table 18 summarizes the PWM Channel-1 registers.

Table 18 PWM channel 1 register map and reset values

## 21.8 PWM Channel-2 Registers (Base address : 0x4000\_5200)

### Channel-2 interrupt register(PWMCH2IR)

Base address : 0x4000\_5200

Address offset : 0x00

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CI	OI													
													R	R	R

#### [0] MI - Match Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Match interrupt does not occur.

1 : Match interrupt occurs.

#### [1] OI - Overflow Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Overflow interrupt does not occur.

1 : Overflow interrupt occurs.

#### [2] CI - Capture Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Capture interrupt does not occur.

1 : Capture interrupt occurs.

### Channel-2 interrupt enable register(PWMCH2IER)

Base address : 0x4000\_5200

Address offset : 0x04

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIE	OIE	MIE												
													R/W	R/W	R/W

[0] MIE - Match Interrupt Enabled.

0 : Match interrupt is not enabled.

1 : Match interrupt is enabled.

[1] OIE - Overflow Interrupt Enable.

0 : Overflow interrupt is not enabled.

1 : Overflow interrupt is enabled.

[2] CIE - Capture Interrupt Enable.

0 : Capture interrupt is not enabled.

1 : Capture interrupt is enabled.

## Channel-2 interrupt clear register(PWMCH2ICR)

Base address : 0x4000\_5200

Address offset : 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIC	OIC	MIC												
													W	W	W

This bit is set by software, cleared by hardware when a capture interrupt becomes 0.

[0] MIC - Match Interrupt

0 : No action.

1 : Match interrupt is cleared.

[1] OIC - Overflow Interrupt

0 : No action.

1 : Overflow Interrupt is cleared.

[2] CIC - Capture Interrupt Clear.

0 : No action.

1 : Capture Interrupt is cleared.

## Channel-2 Timer/Counter Register (PWMCH2TCR)

Base address : 0x4000\_5200

Address offset : 0x0C

Reset value : 0x0000\_0000

31

0

TCR

R

### [31:0] TCR - Timer/Counter register

Timer/Counter register. These register hold the current values of the Timer/Counter(TC). The TC is incremented every PR cycles. When the TC is reached to value of match register, the match interrupt is occurred and PWM output waveform becomes 0. When the TC is reached to the value of limit register, the overflow interrupt is occurred, the TC is reset as 0.

## Channel-2 Prescale Counter Register (PWMCH2PCR)

Base address : 0x4000\_5200

Address offset : 0x10

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res					PCR										
															R

### [5:0] PC - Prescale Counter register

Prescale Counter register. These registers hold the current values of prescale counter(PC). The PC is incremented to the value stored in PR. When the PC is reached to PR, the TC is incremented and the PC is reset as 0.

## Channel-2 Prescale Register (PWMCH2PR)

Base address : 0x4000\_5200

Address offset : 0x14

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res		PR													
												R/W			

#### [5: 0] PR - Prescale Register

Prescale register. The PC is incremented when the PC is reached to the PR.

### Channel-2 Match Register (PWMCH2MR)

Base address : 0x4000\_5200

Address offset : 0x18

Reset value : 0x0000\_0000

31	0
MR	
R/W	

#### [31:0] MR - Match Register

Match register. The MR can generate a match interrupt and PWM output waveform becomes 0 when the TC is reached to the MR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

### Channel-2 Limit Register (PWMCH2LR)

Base address : 0x4000\_5200

Address offset : 0x1C

Reset value : 0x0000\_0000

31	0
LR	
R/W	

#### [31:0] LR - Limit Register

Limit Register. The LR can generate an overflow interrupt and PWM output waveform becomes 1 when the TC is reached to the LR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

### Channel-2 Up-Down Mode Register (PWMCH2UDMR)

Base address : 0x4000\_5200

Address offset : 0x20

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	UDM														
															R/W

[0] UDM - Up-Down mode

0 : TC runs up count.

1 : TC runs down count.

## Channel-2 Timer/Counter Mode Register (PWMCH2TCMR)

Base address : 0x4000\_5200

Address offset : 0x24

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TCM														
															R/W

[1:0] TCM - Timer/Counter mode

00 : Timer mode.

01 : Counter mode with counting driven by rising edge external input .

10 : Counter mode with counting driven by falling edge external input.

11 : Counter mode with counting driven by rising and falling edge external input.

## Channel-2 PWM output Enable and External input Enable Register (PWMCH2PEER)

Base address : 0x4000\_5200

Address offset : 0x28

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PEEE														
														R/W	

[1:0] PEEE - PWM output Enable and External input Enable

00 : PWM output is disable and external input is disable.

01 : PWM output is disable and external input is enable.

10 : PWM output is enable and external input is disable.

## Channel-2 Capture Mode Register (PWMCH2CMR)

Base address : 0x4000\_5200

Address offset : 0x2C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CM														
															R/W

[0] CM - Capture mode

0 : Timer/Counter is captured when external input signal is rising edge.

1 : Timer/Counter is captured when external input signal is falling edge.

## Channel-2 Capture Register (PWMCH2CR)

Base address : 0x4000\_5200

Address offset : 0x30

Reset value : 0x0000\_0000

31	0
CR	
R	

[31:0] CR - Capture Register

Capture register. The CR is loaded with the value of the TC when external input signal is triggered.

## Channel-2 Periodic Mode Register (PWMCH2PDMR)

Base address : 0x4000\_5200

Address offset : 0x34

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PDM														
															R/W

### [0] PDM - Periodic Mode

0 : Periodic mode. When the TC is reached to the LR, the TC returns to 0 and then continues counting periodically.

1 : One-shot mode. When the TC is reached to the LR, the TC returns to 0 and then stops counting.

## Channel-2 Dead Zone Enable Register (PWMCH2DZER)

Base address : 0x4000\_5200

Address offset : 0x38

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	DZE														
															R/W

### [0] DZE - Dead Zone Enable

0 : Dead zone generation is disabled.

1 : Dead zone generation is enabled.

## Channel-2 Dead Zone Counter Register (PWMCH2DZCR)

Base address : 0x4000\_5200

Address offset : 0x3C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	res	res										DZC
															R/W

### [9:0] DZC - Dead Zone Counter value

Dead zone generation counter value register. If the DZE bit in DZER is 1, dead zone counter counts to this value and during this time, the two PWM output waveforms are all 0.

## 21.9 Register map

The following Table 19 summarizes the PWM Channel-2 registers.

Table 19 PWM channel 2 register map and reset values

## 21.10 PWM Channel-3 Registers (Base address : 0x4000\_5300)

### Channel-3 interrupt register(PWMCH3IR)

Base address : 0x4000\_5300

Address offset : 0x00

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CI	OI													
													R	R	R

#### [0] MI - Match Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Match interrupt does not occur.

1 : Match interrupt occurs.

#### [1] OI - Overflow Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Overflow interrupt does not occur.

1 : Overflow interrupt occurs.

#### [2] CI - Capture Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Capture interrupt does not occur.

1 : Capture interrupt occurs.

### Channel-3 interrupt enable register(PWMCH3IER)

Base address : 0x4000\_5300

Address offset : 0x04

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIE	OIE	MIE												

													R/W	R/W	R/W
--	--	--	--	--	--	--	--	--	--	--	--	--	-----	-----	-----

[0] MIE - Match Interrupt Enabled.

0 : Match interrupt is not enabled.

1 : Match interrupt is enabled.

[1] OIE - Overflow Interrupt Enable.

0 : Overflow interrupt is not enabled.

1 : Overflow interrupt is enabled.

[2] CIE - Capture Interrupt Enable.

0 : Capture interrupt is not enabled.

1 : Capture interrupt is enabled.

### Channel-3 interrupt clear register(PWMCH3ICR)

Base address : 0x4000\_5300

Address offset : 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIC	OIC	MIC												

This bit is set by software, cleared by hardware when a capture interrupt becomes 0.

[0] MIC - Match Interrupt

0 : No action.

1 : Match interrupt is cleared.

[1] OIC - Overflow Interrupt

0 : No action.

1 : Overflow Interrupt is cleared.

[2] CIC - Capture Interrupt Clear.

0 : No action.

1 : Capture Interrupt is cleared.

### Channel-3 Timer/Counter Register (PWMCH3TCR)

Base address : 0x4000\_5300

Address offset : 0x0C

Reset value : 0x0000\_0000

31	0
TCR	
R	

#### [31:0] TCR - Timer/Counter register

Timer/Counter register. These register hold the current values of the Timer/Counter(TC). The TC is incremented every PR cycles. When the TC is reached to value of match register, the match interrupt is occurred and PWM output waveform becomes 0. When the TC is reached to the value of limit register, the overflow interrupt is occurred, the TC is reset as 0.

### Channel-3 Prescale Counter Register (PWMCH3PCR)

Base address : 0x4000\_5300

Address offset : 0x10

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															

#### [5:0] PC - Prescale Counter register

Prescale Counter register. These registers hold the current values of prescale counter(PC). The PC is incremented to the value stored in PR. When the PC is reached to PR, the TC is incremented and the PC is reset as 0.

### Channel-3 Prescale Register (PWMCH3PR)

Base address : 0x4000\_5300

Address offset : 0x14

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PR														
														R/W	

#### [5: 0] PR - Prescale Register

Prescale register. The PC is incremented when the PC is reached to the PR.

### Channel-3 Match Register (PWMCH3MR)

Base address : 0x4000\_5300

Address offset : 0x18

Reset value : 0x0000\_0000

31	0
MR	
R/W	

#### [31:0] MR - Match Register

Match register. The MR can generate a match interrupt and PWM output waveform becomes 0 when the TC is reached to the MR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

### Channel-3 Limit Register (PWMCH3LR)

Base address : 0x4000\_5300

Address offset : 0x1C

Reset value : 0x0000\_0000

31	0
LR	
R/W	

#### [31:0] LR - Limit Register

Limit Register. The LR can generate an overflow interrupt and PWM output waveform becomes 1 when the TC is reached to the LR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

## Channel-3 Up-Down Mode Register (PWMCH3UDMR)

Base address : 0x4000\_5300

Address offset : 0x20

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	UDM														
															R/W

[0] UDM - Up-Down mode

0 : TC runs up count.

1 : TC runs down count.

## Channel-3 Timer/Counter Mode Register (PWMCH3TCMR)

Base address : 0x4000\_5300

Address offset : 0x24

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TCM														
															R/W

[1:0] TCM - Timer/Counter mode

00 : Timer mode.

01 : Counter mode with counting driven by rising edge external input .

10 : Counter mode with counting driven by falling edge external input.

11 : Counter mode with counting driven by rising and falling edge external input.

## Channel-3 PWM output Enable and External input Enable Register (PWMCH3PEEER)

Base address : 0x4000\_5300

Address offset : 0x28

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PEEE														
															R/W

[1:0] PEEE - PWM output Enable and External input Enable

00 : PWM output is disable and external input is disable.

01 : PWM output is disable and external input is enable.

10 : PWM output is enable and external input is disable.

## Channel-3 Capture Mode Register (PWMCH3CMR)

Base address : 0x4000\_5300

Address offset : 0x2C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CM														
															R/W

[0] CM - Capture mode

0 : Timer/Counter is captured when external input signal is rising edge.

1 : Timer/Counter is captured when external input signal is falling edge.

## Channel-3 Capture Register (PWMCH3CR)

Base address : 0x4000\_5300

Address offset : 0x30

Reset value : 0x0000\_0000

31	0
CR	
R	

#### [31:0] CR - Capture Register

Capture register. The CR is loaded with the value of the TC when external input signal is triggered.

### Channel-3 Periodic Mode Register (PWMCH3PDMR)

Base address : 0x4000\_5300

Address offset : 0x34

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PDM														
															R/W

#### [0] PDM - Periodic Mode

0 : Periodic mode. When the TC is reached to the LR, the TC returns to 0 and then continues counting periodically.

1 : One-shot mode. When the TC is reached to the LR, the TC returns to 0 and then stops counting.

### Channel-3 Dead Zone Enable Register (PWMCH3DZER)

Base address : 0x4000\_5300

Address offset : 0x38

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	DZE														

															R/W
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-----

**[0] DZE - Dead Zone Enable**

0 : Dead zone generation is disabled.

1 : Dead zone generation is enabled.

### Channel-3 Dead Zone Counter Register (PWMCH3DZCR)

Base address : 0x4000\_5300

Address offset : 0x3C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	res	res							DZC			

**[9:0] DZC - Dead Zone Counter value**

Dead zone generation counter value register. If the DZE bit in DZER is 1, dead zone counter counts to this value and during this time, the two PWM output waveforms are all 0.

## 21.11 Register map

The following Table 20 summarizes the PWM Channel-3 registers.

Table 20 PWM channel 3 register map and reset values

## 21.12 PWM Channel-4 Registers (Base address : 0x4000\_5400)

### Channel-4 interrupt register(PWMCH4IR)

Base address : 0x4000\_5400

Address offset : 0x00

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CI	OI													
													R	R	R

#### [0] MI - Match Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Match interrupt does not occur.

1 : Match interrupt occurs.

#### [1] OI - Overflow Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Overflow interrupt does not occur.

1 : Overflow interrupt occurs.

#### [2] CI - Capture Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Capture interrupt does not occur.

1 : Capture interrupt occurs.

### Channel-4 interrupt enable register(PWMCH4IER)

Base address : 0x4000\_5400

Address offset : 0x04

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIE	OIE	MIE												

													R/W	R/W	R/W
--	--	--	--	--	--	--	--	--	--	--	--	--	-----	-----	-----

[0] MIE - Match Interrupt Enabled.

0 : Match interrupt is not enabled.

1 : Match interrupt is enabled.

[1] OIE - Overflow Interrupt Enable.

0 : Overflow interrupt is not enabled.

1 : Overflow interrupt is enabled.

[2] CIE - Capture Interrupt Enable.

0 : Capture interrupt is not enabled.

1 : Capture interrupt is enabled.

## Channel-4 interrupt clear register(PWMCH4ICR)

Base address : 0x4000\_5400

Address offset : 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIC	OIC	MIC												

This bit is set by software, cleared by hardware when a capture interrupt becomes 0.

[0] MIC - Match Interrupt

0 : No action.

1 : Match interrupt is cleared.

[1] OIC - Overflow Interrupt

0 : No action.

1 : Overflow Interrupt is cleared.

[2] CIC - Capture Interrupt Clear.

0 : No action.

1 : Capture Interrupt is cleared.

## Channel-4 Timer/Counter Register (PWMCH4TCR)

Base address : 0x4000\_5400

Address offset : 0x0C

Reset value : 0x0000\_0000

31

0

TCR

R

### [31:0] TCR - Timer/Counter register

Timer/Counter register. These register hold the current values of the Timer/Counter(TC). The TC is incremented every PR cycles. When the TC is reached to value of match register, the match interrupt is occurred and PWM output waveform becomes 0. When the TC is reached to the value of limit register, the overflow interrupt is occurred, the TC is reset as 0.

## Channel-4 Prescale Counter Register (PWMCH4PCR)

Base address : 0x4000\_5400

Address offset : 0x10

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															

### [5:0] PCR - Prescale Counter register

Prescale Counter register. These registers hold the current values of prescale counter(PC). The PC is incremented to the value stored in PR. When the PC is reached to PR, the TC is incremented and the PC is reset as 0.

## Channel-4 Prescale Register (PWMCH4PR)

Base address : 0x4000\_5400

Address offset : 0x14

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

res	PR									
										R/W

#### [5: 0] PR - Prescale Register

Prescale register. The PC is incremented when the PC is reached to the PR.

### Channel-4 Match Register (PWMCH4MR)

Base address : 0x4000\_5400

Address offset : 0x18

Reset value : 0x0000\_0000

31	0
MR	
R/W	

#### [31:0] MR - Match Register

Match register. The MR can generate a match interrupt and PWM output waveform becomes 0 when the TC is reached to the MR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

### Channel-4 Limit Register (PWMCH4LR)

Base address : 0x4000\_5400

Address offset : 0x1C

Reset value : 0x0000\_0000

31	0
LR	
R/W	

#### [31:0] LR - Limit Register

Limit Register. The LR can generate an overflow interrupt and PWM output waveform becomes 1 when the TC is reached to the LR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

### Channel-4 Up-Down Mode Register (PWMCH4UDMR)

Base address : 0x4000\_5400

Address offset : 0x20

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	UDM														
															R/W

[0] UDM - Up-Down mode

0 : TC runs up count.

1 : TC runs down count.

### Channel-4 Timer/Counter Mode Register (PWMCH4TCMR)

Base address : 0x4000\_5400

Address offset : 0x24

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TCM														
															R/W

[1:0] TCM - Timer/Counter mode

00 : Timer mode.

01 : Counter mode with counting driven by rising edge external input .

10 : Counter mode with counting driven by falling edge external input.

11 : Counter mode with counting driven by rising and falling edge external input.

### Channel-4 PWM output Enable and External input Enable Register (PWMCH4PEEER)

Base address : 0x4000\_5400

Address offset : 0x28

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PEEE														
														R/W	

[1:0] PEEE - PWM output Enable and External input Enable

00 : PWM output is disable and external input is disable.

01 : PWM output is disable and external input is enable.

10 : PWM output is enable and external input is disable.

### Channel-4 Capture Mode Register (PWMCH4CMR)

Base address : 0x4000\_5400

Address offset : 0x2C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CM														
															R/W

[0] CM - Capture mode

0 : Timer/Counter is captured when external input signal is rising edge.

1 : Timer/Counter is captured when external input signal is falling edge.

### Channel-4 Capture Register (PWMCH4CR)

Base address : 0x4000\_5400

Address offset : 0x30

Reset value : 0x0000\_0000

31	0
CR	
R	

[31:0] CR - Capture Register

Capture register. The CR is loaded with the value of the TC when external input signal is triggered.

### Channel-4 Periodic Mode Register (PWMCH4PDMR)

Base address : 0x4000\_5400

Address offset : 0x34

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PDM														
															R/W

[0] PDM - Periodic Mode

0 : Periodic mode. When the TC is reached to the LR, the TC returns to 0 and then continues counting periodically.

1 : One-shot mode. When the TC is reached to the LR, the TC returns to 0 and then stops counting.

### Channel-4 Dead Zone Enable Register (PWMCH4DZER)

Base address : 0x4000\_5400

Address offset : 0x38

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	DZE														
															R/W

[0] DZE - Dead Zone Enable

0 : Dead zone generation is disabled.

1 : Dead zone generation is enabled.

## Channel-4 Dead Zone Counter Register (PWMCH4DZCR)

Base address : 0x4000\_5400

Address offset : 0x3C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	res	res										DZC
															R/W

### [9:0] DZC - Dead Zone Counter value

Dead zone generation counter value register. If the DZE bit in DZER is 1, dead zone counter counts to this value and during this time, the two PWM output waveforms are all 0.

## 21.13 Register map

The following Table 21 summarizes the PWM Channel-4 registers.

Table 21 PWM channel 4 register map and reset values

## 21.14 PWM Channel-5 Registers (Base address : 0x4000\_5500)

### Channel-5 interrupt register(PWMCH5IR)

Base address : 0x4000\_5500

Address offset : 0x00

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CI	OI													
													R	R	R

#### [0] MI - Match Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Match interrupt does not occur.

1 : Match interrupt occurs.

#### [1] OI - Overflow Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Overflow interrupt does not occur.

1 : Overflow interrupt occurs.

#### [2] CI - Capture Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Capture interrupt does not occur.

1 : Capture interrupt occurs.

### Channel-5 interrupt enable register(PWMCH5IER)

Base address : 0x4000\_5500

Address offset : 0x04

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIE	OIE	MIE												

													R/W	R/W	R/W
--	--	--	--	--	--	--	--	--	--	--	--	--	-----	-----	-----

[0] MIE - Match Interrupt Enabled.

0 : Match interrupt is not enabled.

1 : Match interrupt is enabled.

[1] OIE - Overflow Interrupt Enable.

0 : Overflow interrupt is not enabled.

1 : Overflow interrupt is enabled.

[2] CIE - Capture Interrupt Enable.

0 : Capture interrupt is not enabled.

1 : Capture interrupt is enabled.

## Channel-5 interrupt clear register(PWMCH5ICR)

Base address : 0x4000\_5500

Address offset : 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIC	OIC	MIC												

This bit is set by software, cleared by hardware when a capture interrupt becomes 0.

[0] MIC - Match Interrupt

0 : No action.

1 : Match interrupt is cleared.

[1] OIC - Overflow Interrupt

0 : No action.

1 : Overflow Interrupt is cleared.

[2] CIC - Capture Interrupt Clear.

0 : No action.

1 : Capture Interrupt is cleared.

## Channel-5 Timer/Counter Register (PWMCH5TCR)

Base address : 0x4000\_5500

Address offset : 0x0C

Reset value : 0x0000\_0000

31

0

TCR

R

### [31:0] TCR - Timer/Counter register

Timer/Counter register. These register hold the current values of the Timer/Counter(TC). The TC is incremented every PR cycles. When the TC is reached to value of match register, the match interrupt is occurred and PWM output waveform becomes 0. When the TC is reached to the value of limit register, the overflow interrupt is occurred, the TC is reset as 0.

## Channel-5 Prescale Counter Register (PWMCH5PCR)

Base address : 0x4000\_5500

Address offset : 0x10

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res					PCR										
															R

### [5:0] PC - Prescale Counter register

Prescale Counter register. These registers hold the current values of prescale counter(PC). The PC is incremented to the value stored in PR. When the PC is reached to PR, the TC is incremented and the PC is reset as 0.

## Channel-5 Prescale Register (PWMCH5PR)

Base address : 0x4000\_5500

Address offset : 0x14

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res		PR													
												R/W			

#### [5: 0] PR - Prescale Register

Prescale register. The PC is incremented when the PC is reached to the PR.

### Channel-5 Match Register (PWMCH5MR)

Base address : 0x4000\_5500

Address offset : 0x18

Reset value : 0x0000\_0000

31	0
MR	
R/W	

#### [31:0] MR - Match Register

Match register. The MR can generate a match interrupt and PWM output waveform becomes 0 when the TC is reached to the MR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

### Channel-5 Limit Register (PWMCH5LR)

Base address : 0x4000\_5500

Address offset : 0x1C

Reset value : 0x0000\_0000

31	0
LR	
R/W	

#### [31:0] LR - Limit Register

Limit Register. The LR can generate an overflow interrupt and PWM output waveform becomes 1 when the TC is reached to the LR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

### Channel-5 Up-Down Mode Register (PWMCH5UDMR)

Base address : 0x4000\_5500

Address offset : 0x20

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	UDM														
															R/W

[0] UDM - Up-Down mode

0 : TC runs up count.

1 : TC runs down count.

### Channel-5 Timer/Counter Mode Register (PWMCH5TCMR)

Base address : 0x4000\_5500

Address offset : 0x24

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TCM														
															R/W

[1:0] TCM - Timer/Counter mode

00 : Timer mode.

01 : Counter mode with counting driven by rising edge external input .

10 : Counter mode with counting driven by falling edge external input.

11 : Counter mode with counting driven by rising and falling edge external input.

### Channel-5 PWM output Enable and External input Enable Register (PWMCH5PEEER)

Base address : 0x4000\_5500

Address offset : 0x28

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PEEE														
														R/W	

[1:0] PEEE - PWM output Enable and External input Enable

00 : PWM output is disable and external input is disable.

01 : PWM output is disable and external input is enable.

10 : PWM output is enable and external input is disable.

### Channel-5 Capture Mode Register (PWMCH5CMR)

Base address : 0x4000\_5500

Address offset : 0x2C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CM														
															R/W

[0] CM - Capture mode

0 : Timer/Counter is captured when external input signal is rising edge.

1 : Timer/Counter is captured when external input signal is falling edge.

### Channel-5 Capture Register (PWMCH5CR)

Base address : 0x4000\_5500

Address offset : 0x30

Reset value : 0x0000\_0000

31	0
CR	
R	

### [31:0] CR - Capture Register

Capture register. The CR is loaded with the value of the TC when external input signal is triggered.

## Channel-5 Periodic Mode Register (PWMCH5PDMR)

Base address : 0x4000\_5500

Address offset : 0x34

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PDM														
															R/W

### [0] PDM - Periodic Mode

0 : Periodic mode. When the TC is reached to the LR, the TC returns to 0 and then continues counting periodically.

1 : One-shot mode. When the TC is reached to the LR, the TC returns to 0 and then stops counting.

## Channel-5 Dead Zone Enable Register (PWMCH5DZER)

Base address : 0x4000\_5500

Address offset : 0x38

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	DZE														
															R/W

### [0] DZE - Dead Zone Enable

0 : Dead zone generation is disabled.

1 : Dead zone generation is enabled.

## Channel-5 Dead Zone Counter Register (PWMCH5DZCR)

Base address : 0x4000\_5500

Address offset : 0x3C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	res	res										DZC
															R/W

### [9:0] DZC - Dead Zone Counter value

Dead zone generation counter value register. If the DZE bit in DZER is 1, dead zone counter counts to this value and during this time, the two PWM output waveforms are all 0.

21.15 Register map

The following Table 22 summarizes the PWM Channel-5 registers.

Table 22 PWM channel 5 register map and reset values

## 21.16 PWM Channel-6 Registers (Base address : 0x4000\_5600)

### Channel-6 interrupt register(PWMCH6IR)

Base address : 0x4000\_5600

Address offset : 0x00

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CI	OI	MI												
													R	R	R

#### [0] MI - Match Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Match interrupt does not occur.

1 : Match interrupt occurs.

#### [1] OI - Overflow Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Overflow interrupt does not occur.

1 : Overflow interrupt occurs.

#### [2] CI - Capture Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Capture interrupt does not occur.

1 : Capture interrupt occurs.

### Channel-6 interrupt enable register(PWMCH6IER)

Base address : 0x4000\_5600

Address offset : 0x04

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIE	OIE	MIE												

													R/W	R/W	R/W
--	--	--	--	--	--	--	--	--	--	--	--	--	-----	-----	-----

[0] MIE - Match Interrupt Enabled.

0 : Match interrupt is not enabled.

1 : Match interrupt is enabled.

[1] OIE - Overflow Interrupt Enable.

0 : Overflow interrupt is not enabled.

1 : Overflow interrupt is enabled.

[2] CIE - Capture Interrupt Enable.

0 : Capture interrupt is not enabled.

1 : Capture interrupt is enabled.

## Channel-6 interrupt clear register(PWMCH6ICR)

Base address : 0x4000\_5600

Address offset : 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIC	OIC	MIC												

This bit is set by software, cleared by hardware when a capture interrupt becomes 0.

[0] MIC - Match Interrupt

0 : No action.

1 : Match interrupt is cleared.

[1] OIC - Overflow Interrupt

0 : No action.

1 : Overflow Interrupt is cleared.

[2] CIC - Capture Interrupt Clear.

0 : No action.

1 : Capture Interrupt is cleared.

## Channel-6 Timer/Counter Register (PWMCH6TCR)

Base address : 0x4000\_5600

Address offset : 0x0C

Reset value : 0x0000\_0000

31

0

TCR

R

### [31:0] TCR - Timer/Counter register

Timer/Counter register. These register hold the current values of the Timer/Counter(TC). The TC is incremented every PR cycles. When the TC is reached to value of match register, the match interrupt is occurred and PWM output waveform becomes 0. When the TC is reached to the value of limit register, the overflow interrupt is occurred, the TC is reset as 0.

## Channel-6 Prescale Counter Register (PWMCH6PCR)

Base address : 0x4000\_5600

Address offset : 0x10

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															

### [5:0] PC - Prescale Counter register

Prescale Counter register. These registers hold the current values of prescale counter(PC). The PC is incremented to the value stored in PR. When the PC is reached to PR, the TC is incremented and the PC is reset as 0.

## Channel-6 Prescale Register (PWMCH6PR)

Base address : 0x4000\_5600

Address offset : 0x14

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PR														
														R/W	

#### [5:0] PR - Prescale Register

Prescale register. The PC is incremented when the PC is reached to the PR.

### Channel-6 Match Register (PWMCH6MR)

Base address : 0x4000\_5600

Address offset : 0x18

Reset value : 0x0000\_0000

31	0
MR	
R/W	

#### [31:0] MR - Match Register

Match register. The MR can generate a match interrupt and PWM output waveform becomes 0 when the TC is reached to the MR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

### Channel-6 Limit Register (PWMCH6LR)

Base address : 0x4000\_5600

Address offset : 0x1C

Reset value : 0x0000\_0000

31	0
LR	
R/W	

#### [31:0] LR - Limit Register

Limit Register. The LR can generate an overflow interrupt and PWM output waveform becomes 1 when the TC is reached to the LR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

## Channel-6 Up-Down Mode Register (PWMCH6UDMR)

Base address : 0x4000\_5600

Address offset : 0x20

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	UDM														
															R/W

[0] UDM - Up-Down mode

0 : TC runs up count.

1 : TC runs down count.

## Channel-6 Timer/Counter Mode Register (PWMCH6TCMR)

Base address : 0x4000\_5600

Address offset : 0x24

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TCM														
															R/W

[1:0] TCM - Timer/Counter mode

00 : Timer mode.

01 : Counter mode with counting driven by rising edge external input .

10 : Counter mode with counting driven by falling edge external input.

11 : Counter mode with counting driven by rising and falling edge external input.

## Channel-6 PWM output Enable and External input Enable Register (PWMCH6PEER)

Base address : 0x4000\_5600

Address offset : 0x28

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PEEE														
															R/W

[1:0] PEEE - PWM output Enable and External input Enable

00 : PWM output is disable and external input is disable.

01 : PWM output is disable and external input is enable.

10 : PWM output is enable and external input is disable.

## Channel-6 Capture Mode Register (PWMCH6CMR)

Base address : 0x4000\_5600

Address offset : 0x2C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CM														
															R/W

[0] CM - Capture mode

0 : Timer/Counter is captured when external input signal is rising edge.

1 : Timer/Counter is captured when external input signal is falling edge.

## Channel-6 Capture Register (PWMCH6CR)

Base address : 0x4000\_5600

Address offset : 0x30

Reset value : 0x0000\_0000

31	0
CR	
R	

#### [31:0] CR - Capture Register

Capture register. The CR is loaded with the value of the TC when external input signal is triggered.

### Channel-6 Periodic Mode Register (PWMCH6PDMR)

Base address : 0x4000\_5600

Address offset : 0x34

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PDM														
															R/W

#### [0] PDM - Periodic Mode

0 : Periodic mode. When the TC is reached to the LR, the TC returns to 0 and then continues counting periodically.

1 : One-shot mode. When the TC is reached to the LR, the TC returns to 0 and then stops counting.

### Channel-6 Dead Zone Enable Register (PWMCH6DZER)

Base address : 0x4000\_5600

Address offset : 0x38

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	DZE														

															R/W
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-----

[0] DZE - Dead Zone Enable

0 : Dead zone generation is disabled.

1 : Dead zone generation is enabled.

### Channel-6 Dead Zone Counter Register (PWMCH6DZCR)

Base address : 0x4000\_5600

Address offset : 0x3C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	res	res							DZC			
												R/W			

[9:0] DZC - Dead Zone Counter value

Dead zone generation counter value register. If the DZE bit in DZER is 1, dead zone counter counts to this value and during this time, the two PWM output waveforms are all 0.

21.17 Register map

The following Table 23 summarizes the PWM Channel-5 registers.

Table 23 PWM channel 6 register map and reset values

## 21.18 PWM Channel-7 Registers (Base address : 0x4000\_5700)

### Channel-7 interrupt register(PWMCH7IR)

Base address : 0x4000\_5700

Address offset : 0x00

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CI	OI													
													R	R	R

#### [0] MI - Match Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Match interrupt does not occur.

1 : Match interrupt occurs.

#### [1] OI - Overflow Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Overflow interrupt does not occur.

1 : Overflow interrupt occurs.

#### [2] CI - Capture Interrupt

This bit is set by hardware and cleared by interrupt clear register.

0 : Capture interrupt does not occur.

1 : Capture interrupt occurs.

### Channel-7 interrupt enable register(PWMCH7IER)

Base address : 0x4000\_5700

Address offset : 0x04

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIE	OIE	MIE												

													R/W	R/W	R/W
--	--	--	--	--	--	--	--	--	--	--	--	--	-----	-----	-----

[0] MIE - Match Interrupt Enabled.

0 : Match interrupt is not enabled.

1 : Match interrupt is enabled.

[1] OIE - Overflow Interrupt Enable.

0 : Overflow interrupt is not enabled.

1 : Overflow interrupt is enabled.

[2] CIE - Capture Interrupt Enable.

0 : Capture interrupt is not enabled.

1 : Capture interrupt is enabled.

## Channel-7 interrupt clear register(PWMCH7ICR)

Base address : 0x4000\_5700

Address offset : 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CIC	OIC	MIC												

This bit is set by software, cleared by hardware when a capture interrupt becomes 0.

[0] MIC - Match Interrupt

0 : No action.

1 : Match interrupt is cleared.

[1] OIC - Overflow Interrupt

0 : No action.

1 : Overflow Interrupt is cleared.

[2] CIC - Capture Interrupt Clear.

0 : No action.

1 : Capture Interrupt is cleared.

## Channel-7 Timer/Counter Register (PWMCH7TCR)

Base address : 0x4000\_5700

Address offset : 0x0C

Reset value : 0x0000\_0000

31

0

TCR

R

### [31:0] TCR - Timer/Counter register

Timer/Counter register. These register hold the current values of the Timer/Counter(TC). The TC is incremented every PR cycles. When the TC is reached to value of match register, the match interrupt is occurred and PWM output waveform becomes 0. When the TC is reached to the value of limit register, the overflow interrupt is occurred, the TC is reset as 0.

## Channel-7 Prescale Counter Register (PWMCH7PCR)

Base address : 0x4000\_5700

Address offset : 0x10

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															

### [5:0] PC - Prescale Counter register

Prescale Counter register. These registers hold the current values of prescale counter(PC). The PC is incremented to the value stored in PR. When the PC is reached to PR, the TC is incremented and the PC is reset as 0.

## Channel-7 Prescale Register (PWMCH7PR)

Base address : 0x4000\_5700

Address offset : 0x14

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PR														
														R/W	

#### [5:0] PR - Prescale Register

Prescale register. The PC is incremented when the PC is reached to the PR.

### Channel-7 Match Register (PWMCH7MR)

Base address : 0x4000\_5700

Address offset : 0x18

Reset value : 0x0000\_0000

31	0
MR	
R/W	

#### [31:0] MR - Match Register

Match register. The MR can generate a match interrupt and PWM output waveform becomes 0 when the TC is reached to the MR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

### Channel-7 Limit Register (PWMCH7LR)

Base address : 0x4000\_5700

Address offset : 0x1C

Reset value : 0x0000\_0000

31	0
LR	
R/W	

#### [31:0] LR - Limit Register

Limit Register. The LR can generate an overflow interrupt and PWM output waveform becomes 1 when the TC is reached to the LR. Match register should be smaller than limit register(LR). If not, match interrupt is not occurred and PWM output waveform is always 1.

## Channel-7 Up-Down Mode Register (PWMCH7UDMR)

Base address : 0x4000\_5700

Address offset : 0x20

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	UDM														
															R/W

[0] UDM - Up-Down mode

0 : TC runs up count.

1 : TC runs down count.

## Channel-7 Timer/Counter Mode Register (PWMCH7TCMR)

Base address : 0x4000\_5700

Address offset : 0x24

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TCM														
															R/W

[1:0] TCM - Timer/Counter mode

00 : Timer mode.

01 : Counter mode with counting driven by rising edge external input .

10 : Counter mode with counting driven by falling edge external input.

11 : Counter mode with counting driven by rising and falling edge external input.

## Channel-7 PWM output Enable and External input Enable Register (PWMCH7PEEER)

Base address : 0x4000\_5700

Address offset : 0x28

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PEEE														
															R/W

[1:0] PEEE - PWM output Enable and External input Enable

00 : PWM output is disable and external input is disable.

01 : PWM output is disable and external input is enable.

10 : PWM output is enable and external input is disable.

## Channel-7 Capture Mode Register (PWMCH7CMR)

Base address : 0x4000\_5700

Address offset : 0x2C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CM														
															R/W

[0] CM - Capture mode

0 : Timer/Counter is captured when external input signal is rising edge.

1 : Timer/Counter is captured when external input signal is falling edge.

## Channel-7 Capture Register (PWMCH7CR)

Base address : 0x4000\_5700

Address offset : 0x30

Reset value : 0x0000\_0000

31	0
CR	
R	

#### [31:0] CR - Capture Register

Capture register. The CR is loaded with the value of the TC when external input signal is triggered.

### Channel-7 Periodic Mode Register (PWMCH7PDMR)

Base address : 0x4000\_5700

Address offset : 0x34

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PDM														
															R/W

#### [0] PDM - Periodic Mode

0 : Periodic mode. When the TC is reached to the LR, the TC returns to 0 and then continues counting periodically.

1 : One-shot mode. When the TC is reached to the LR, the TC returns to 0 and then stops counting.

### Channel-7 Dead Zone Enable Register (PWMCH7DZER)

Base address : 0x4000\_5700

Address offset : 0x38

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	DZE														

															R/W
--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	-----

[0] DZE - Dead Zone Enable

0 : Dead zone generation is disabled.

1 : Dead zone generation is enabled.

### Channel-7 Dead Zone Counter Register (PWMCH7DZCR)

Base address : 0x4000\_5700

Address offset : 0x3C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	res	res							DZC			

[9:0] DZC - Dead Zone Counter value

Dead zone generation counter value register. If the DZE bit in DZER is 1, dead zone counter counts to this value and during this time, the two PWM output waveforms are all 0.

21.19 Register map

The following Table 24 summarizes the PWM Channel-5 registers.

Table 24 PWM channel 7 register map and reset values

---

## 21.20 PWM Common Registers (Base address : 0x4000\_5800)

### Interrupt Enable Register (IER)

Base address : 0x4000\_5800

Address offset : 0x00

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0							
								R/W							

#### [0] IE0 - Channel 0 Interrupt Enable

0 : Channel 0 interrupt is disabled.

1 : Channel 0 interrupt is enabled.

#### [1] IE1 - Channel 1 Interrupt Enable

0 : Channel 1 interrupt is disabled.

1 : Channel 1 interrupt is enabled.

#### [2] IE2 - Channel 2 Interrupt Enable

0 : Channel 2 interrupt is disabled.

1 : Channel 2 interrupt is enabled.

#### [3] IE3 - Channel 3 Interrupt Enable

0 : Channel 3 interrupt is disabled.

1 : Channel 3 interrupt is enabled.

#### [4] IE4 - Channel 4 Interrupt Enable

0 : Channel 4 interrupt is disabled.

1 : Channel 4 interrupt is enabled.

#### [5] IE5 - Channel 5 Interrupt Enable

0 : Channel 5 interrupt is disabled.

1 : Channel 5 interrupt is enabled.

#### [6] IE6 - Channel 6 Interrupt Enable

0 : Channel 6 interrupt is disabled.

1 : Channel 6 interrupt is enabled.

#### [7] IE7 - Channel 7 Interrupt Enable

0 : Channel 7 interrupt is disabled.

1 : Channel 7 interrupt is enabled.

## Start/Stop Register (SSR)

Base address : 0x4000\_5800

Address offset : 0x04

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	SS7	SS6	SS5	SS4	SS3	SS2	SS1	SS0							
								R/W							

[0] SS0 - Channel 0 Timer/Counter Start or Stop.

0 : Timer/Counter stop.

1 : Timer/Counter start.

[1] SS1 - Channel 1 Timer/Counter Start or Stop.

0 : Timer/Counter stop.

1 : Timer/Counter start.

[2] SS2 - Channel 2 Timer/Counter Start or Stop.

0 : Timer/Counter stop.

1 : Timer/Counter start.

[3] SS3 - Channel 3 Timer/Counter Start or Stop.

0 : Timer/Counter stop.

1 : Timer/Counter start.

[4] SS4 - Channel 4 Timer/Counter Start or Stop.

0 : Timer/Counter stop.

1 : Timer/Counter start.

[5] SS5 - Channel 5 Timer/Counter Start or Stop.

0 : Timer/Counter stop.

1 : Timer/Counter start.

[6] SS6 - Channel 6 Timer/Counter Start or Stop.

0 : Timer/Counter stop.

1 : Timer/Counter start.

[7] SS7 - Channel 7 Timer/Counter Start or Stop.

0 : Timer/Counter stop.

1 : Timer/Counter start.

## Pause Register (PSR)

Base address : 0x4000\_5800

Address offset : 0x04

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	PS7	PS6	PS5	PS4	PS3	PS2	PS1	PS0							
								R/W							

The Timer/Counter is paused after TC is reached to value of limit register.

[0] PS0 - Channel 0 Timer/Counter Pause.

0 : Timer/Counter is not paused.

1 : Timer/Counter is paused.

[1] PS1 - Channel 1 Timer/Counter Pause.

0 : Timer/Counter is not paused.

1 : Timer/Counter is paused.

[2] PS2 - Channel 2 Timer/Counter Pause.

0 : Timer/Counter is not paused.

1 : Timer/Counter is paused.

[3] PS3 - Channel 3 Timer/Counter Pause.

0 : Timer/Counter is not paused.

1 : Timer/Counter is paused.

[4] PS0 - Channel 4 Timer/Counter Pause.

0 : Timer/Counter is not paused.

1 : Timer/Counter is paused.

[5] PS0 - Channel 5 Timer/Counter Pause.

0 : Timer/Counter is not paused.

1 : Timer/Counter is paused.

[6] PS0 - Channel 6 Timer/Counter Pause.

0 : Timer/Counter is not paused.

1 : Timer/Counter is paused.

[7] PS0 - Channel 7 Timer/Counter Pause.

0 : Timer/Counter is not paused.

1 : Timer/Counter is paused.

## 21.21 Register map

The following Table 25 summarizes the PWM Common registers.

Table 25 PWM common register map and reset values

## 22 Dual timers

### 22.1 Introduction

The dual timer consists two programmable 32-bit or 16-bit Free-running counters(FRCs) that can generate interrupts when they reach 0. There are two dual timers and 4 FRCs. One dual timer has one interrupt handler, resulting in two interrupts of timers. Also one dual timer has one clock but two clock enable signals. Users can select one repetition modes one-shot or wrapping mode, and wrapping mode consists free-running and periodic mode. Two FRCs are one set so two FRCs has one clock, reset, and interrupt but each FRC has an individual clock enable.

### 22.2 Features

- One dual timer has two Free-Running Counters(FRCs).
- One dual timer has one interrupt handler and one clock.
- One dual timer has two clock enable signals.
- There are 2 dual timers.
- A 32-bit or a 16-bit down counter.
- One of the following repetition modes: one-shot and wrapping mode.
- One of the following wrapping modes: Free-running and periodic mode.
- There is a prescaler that can divide down the clock rate by 1, 16, or 256.

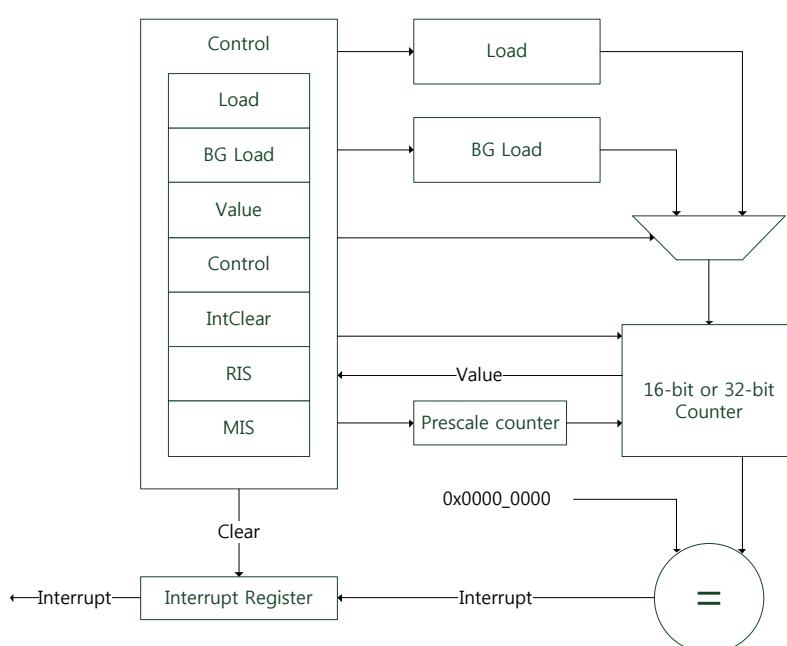


Figure 39 Block diagram of Dualtimer

## 22.3 Functional description

### Clock and clock enable

The dual timers contain PCLK and TIMERCLK clock inputs. PCLK is the main APB system clock and is used by the register interface. TIMERCLK is the input to the prescale units and the decrementing counters. PCLK and TIMERCLK are synchronous.

The dual timers consist two programmable 32-bit Free-Running Counters(FRC) which operate independently. The two timers operate from one TIMERCLK but Each FRC is controlled independently by individual clock enable.

### Timer size

Users can select FRC as 16-bit or 32-bit using the control register.

### Prescaler

The timer has a prescaler that can divide down the enabled clock rate by 1, 16 or 256.

### Repetition mode

There are two repetition mode: one-shot and wrapping mode. Wrapping mode has two modes: free-running and periodic mode.

#### One-shot mode

The counter generates an interrupt once. When the counter reaches 0, it halts until users reprogram it. Users can do this as below:

- Clear the one-shot count bit in the control register, in which case the count proceeds according to the selection of wrapping mode(free-running or periodic mode).
- Write a new value to the Load Value register.

#### Wrapping mode

##### Free-running mode

The counter wraps after reaching its zero value, and continues to count down from the maximum value. This is the default mode.

---

### Periodic mode

The counter generates an interrupt at a constant interval, reloading the original value after wrapping past zero.

## Interrupt

An interrupt is generated when the counter reaches 0 and is only cleared when the interrupt clear register is accessed.

The register holds the value until the interrupt is cleared.

Users can mask interrupts by writing 0 to the Interrupt Enable bit in the control register. Users can read the following from status registers:

- Raw interrupt status before masking.
- Final interrupt status after masking.

The interrupts from the individual timers after masking are logically ORed into a combined interrupt.

## Operation

The operation of each timer is identical. The timer is loaded by writing to the load register and counts down to 0 if enabled. When a counter is already running, writing to the load register causes the counter to immediately restart at the new value. Writing to the background load value has no effect on the current count. In periodic mode, the counter continues to decrease to 0 and restart from the new load value.

An interrupt is generated when 0 is reached. Users can clear the interrupt by writing to the clear register. If users select one-shot mode, the counter halts when it reaches 0 until users deselect one-shot mode or write a new load value.

Otherwise, after reaching a zero count, if the timer is operating in free-running mode, it continues to decrease from its maximum value. If users select periodic mode, the timer reloads the count value from the load register and continues to decrease. In this mode, the counter effectively generates a periodic interrupt.

## How to set the dual timers

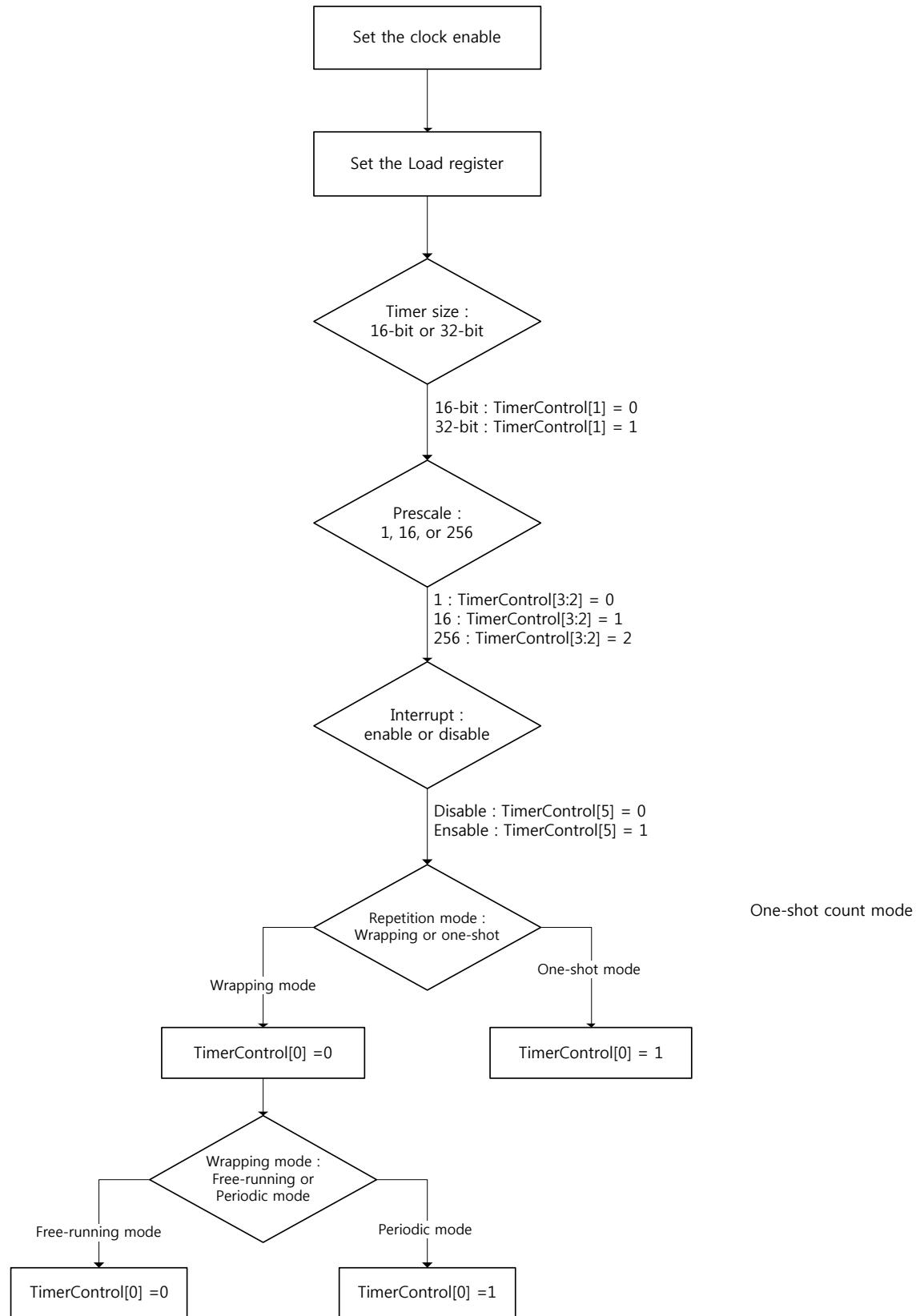


Figure 40 The Dual timer setting flow

## 22.4 Dual timer0\_0 Registers (Base address : 0x4000\_1000)

### Timer0\_0 Load Register(DUALTIMER0\_0TimerLoad)

Base address : 0x4000\_1000

Address offset : 0x00

Reset value : 0x0000\_0000

31	0
TLR	
R/W	

#### [31:0] TLR - Timer Load Register

This register contains the value from which the counter is to decrement. This is the value used to reload the counter when Periodic mode is enabled, and the current count reaches 0.

### Timer0\_0 Value Register(DUALTIMER0\_0TimerValue)

Base address : 0x4000\_1000

Address offset : 0x04

Reset value : 0xFFFF\_FFFF

31	0
TVR	
R	

#### [31:0] TVR - Timer Value Register

This register provides the current value of the decrementing counter.

### Timer0\_0 Control Register(DUALTIMER0\_0TimerControl)

Base address : 0x4000\_1000

Address offset : 0x08

Reset value : 0x0000\_0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TE	TM	IE	res	TP	TS	OC								

[0] OC - One-shot Count

0 : Wrapping mode, default.

1 : One-shot mode.

[1] TS - Timer Size

0 : 16-bit counter, default.

1 : 32-bit counter.

[3:2] TP - Timer Prescale.

00 : 0 stages of prescale, clock is divided by 1, default.

01 : 4 stages of prescale, clock is divided by 16.

10 : 8 stages of prescale, clock is divided by 256.

11 : Undefined, do not use.

[5] IE - Interrupt Enable.

0 : Timer Interrupt disable.

1 : Timer Interrupt enabled, default.

[6] TM - Timer Mode.

0 : Timer is in free-running mode, default.

1 : Timer is in periodic mode.

[7] TE - Timer Enable.

0 : Timer disabled, default.

1 : Timer enabled.

## Timer0\_0 Interrupt Clear Register (DUALTIMERO\_0TimerIntClr)

Base address : 0x4000\_1000

Address offset : 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TIC														

[0] TIC - Interrupt Clear

Set to this register clears the interrupt output from the counter.

## Timer0\_0 Raw Interrupt Status Register (DUALTIMERO\_0TimerRIS)

Base address : 0x4000\_1000

Address offset : 0x10

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	RIS														
															R

#### [0] RIS - Raw Interrupt Status Register

This register indicates the raw interrupt status from the counter. This value is ANDed with the timer interrupt enable bit from the Timer Control Register to create the masked interrupt, that is passed to the interrupt output pin.

### Timer0\_0 Masked Interrupt Status Register (DUALTIMERO\_0TimerMIS)

Base address : 0x4000\_1000

Address offset : 0x14

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	MIS														
															R

#### [0] MIS - Masked Interrupt Status Register

This register indicates the masked interrupt status from the counter. This value is the logical AND of the raw interrupt status with the timer interrupt enable bit from the Timer Control Register, and is the same value that is passed to the interrupt output pin.

### Timer0\_0 Background Load Register (DUALTIMERO\_0TimerBGLoad)

Base address : 0x4000\_1000

Address offset : 0x18

Reset value : 0x0000\_0000

BGL
R/W

#### [31:0] BGL - Background Load Register

This register contains the value from which the counter is to decrement. This is the value used to reload the counter when Periodic mode is enabled, and the current count reaches 0.

This register provides an alternative method of accessing the TimerLoad Register. The difference is that writes to TimerBGLoad Register do not cause the counter to immediately restart from the new value.

Reading from this register returns the same value returned from TimerLoad Register.

## 22.5 Register map

The following Table 26 summarizes the Dual timer 0\_0 registers.

Table 26 Dual timer 0\_0 register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	비고				
0x00	DUALTIMER0_0TimerLoad	res	1	1	Timer0_0 Load Register																																	
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x04	DUALTIMER0_0TimerValue	res	1	1	res	1	1	Timer0_0 Value Register																														
	reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
0x08	DUALTIMER0_0TimerControl	res	res	res	OC	Timer0_0 Control Register																																
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	DUALTIMER0_0TimerIntClr	res	res	res	TIC	Timer0_0 Interrupt Clear Register																																
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Write only register
0x10	DUALTIMER0_0TimerRIS	res	res	res	res	0	Timer0_0 Raw Interrupt Status Register																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	DUALTIMER0_0TimerMIS	res	res	res	res	0	Timer0_0 Masked Interrupt Status Register																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	DUALTIMER0_0TimerBGLoad	res	res	res	res	res	0	Timer0_0 Background Load Register																														
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

## 22.6 Dual timer0\_1 Registers (Base address : 0x4000\_1020)

### Timer0\_1 Load Register(DUALTIMER0\_1TimerLoad)

Base address : 0x4000\_1020

Address offset : 0x00

Reset value : 0x0000\_0000

31	0
TLR	
R/W	

[31:0] TLR - Timer Load Register

This register contains the value from which the counter is to decrement. This is the value used to reload the counter when Periodic mode is enabled, and the current count reaches 0.

### Timer0\_1 Value Register(DUALTIMER0\_1TimerValue)

Base address : 0x4000\_1020

Address offset : 0x04

Reset value : 0xFFFF\_FFFF

31	0
TVR	
R	

[31:0] TVR - Timer Value Register

This register provides the current value of the decrementing counter.

### Timer0\_1 Control Register(DUALTIMER0\_1TimerControl)

Base address : 0x4000\_1020

Address offset : 0x08

Reset value : 0x0000\_0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TE	TM	IE	res	TP	TS	OC								
								R/W	R/W	R/W		R/W	R/W	R/W	

[0] OC - One-shot Count

0 : Wrapping mode, default.

1 : One-shot mode.

[1] TS - Timer Size

0 : 16-bit counter, default.

1 : 32-bit counter.

[3:2] TP - Timer Prescale.

00 : 0 stages of prescale, clock is divided by 1, default.

01 : 4 stages of prescale, clock is divided by 16.

10 : 8 stages of prescale, clock is divided by 256.

11 : Undefined, do not use.

[5] IE - Interrupt Enable.

0 : Timer Interrupt disable.

1 : Timer Interrupt enabled, default.

[6] TM - Timer Mode.

0 : Timer is in free-running mode, default.

1 : Timer is in periodic mode.

[7] TE - Timer Enable.

0 : Timer disabled, default.

1 : Timer enabled.

## Timer0\_1 Interrupt Clear Register (DUALTIMERO\_1TimerIntClr)

Base address : 0x4000\_1020

Address offset : 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	IC														
															W

[0] IC - Interrupt Clear

Set to the this register clears the interrupt output from the counter.

## Timer0\_1 Raw Interrupt Status Register (DUALTIMERO\_1TimerRIS)

Base address : 0x4000\_1020

Address offset : 0x10

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	RIS														
															R

[0] RIS - Raw Interrupt Status Register

This register indicates the raw interrupt status from the counter. This value is ANDed with the timer interrupt enable bit from the Timer Control Register to create the masked interrupt, that is passed to the interrupt output pin.

### Timer0\_1 Masked Interrupt Status Register (DUALTIMERO\_1TimerMIS)

Base address : 0x4000\_1020

Address offset : 0x14

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	MIS														
															R

[0] MIS - Masked Interrupt Status Register

This register indicates the masked interrupt status from the counter. This value is the logical AND of the raw interrupt status with the timer interrupt enable bit from the Timer Control Register, and is the same value that is passed to the interrupt output pin.

### Timer0\_1 Background Load Register (DUALTIMERO\_1TimerBGLoad)

Base address : 0x4000\_1020

Address offset : 0x18

Reset value : 0x0000\_0000

31	0
BGL	

R/W

## [31:0] BGL - Background Load Register

This register contains the value from which the counter is to decrement. This is the value used to reload the counter when Periodic mode is enabled, and the current count reaches 0.

This register provides an alternative method of accessing the TimerLoad Register. The difference is that writes to TimerBGLoad Register do not cause the counter to immediately restart from the new value.

Reading from this register returns the same value returned from TimerLoad Register.

## 22.7 Register map

The following Table 27 summarizes the Dual timer 0\_1 registers.

Table 27 Dual timer 0\_1 register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hi	Lo
0x00	DUALTIMER0_1TimerLoad	TLR																															Timer0_1 Load Register		
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x04	DUALTIMER0_1TimerValue	TVR																														Timer0_1 Value Register			
	reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
0x08	DUALTIMER0_1TimerControl	TCR																														Timer0_1 Control Register			
	reset value	RES	0																																
0x0C	DUALTIMER0_1TimerIntClr	TIC																														Timer0_1 Interrupt Clear Register			
	reset value	RES	Write only register																																
0x10	DUALTIMER0_1TimerRIS	RIS																														Timer0_1 Raw Interrupt Status Register			
	reset value	RES	0																																
0x14	DUALTIMER0_1TimerMIS	MIS																														Timer0_1 Masked Interrupt Status Register			
	reset value	RES	0																																
0x18	DUALTIMER0_1TimerBGLoad	BGL																														Timer0_1 Background Load Register			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

## 22.8 Dual Timer 0 Clock Enable Register (Base address : 0x4000\_1080)

### Timer0\_0 Clock Enable Register (TIMCLKEN0\_0)

Base address : 0x4000\_1080

Address offset : 0x00

Reset value : 0x0000\_0000

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

[0] CE - Clock Enable Register

0 : Clock disable

1 : Clock enable

### Timer0\_1 Clock Enable Register (TIMCLKEN0\_1)

Base address : 0x4000\_1080

Address offset : 0x20

Reset value : 0x0000\_0000

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

[0] CE - Clock Enable Register

0 : Clock disable

1 : Clock enable

## 22.9 Register map

The following Table 28 summarizes the Dual timer 0 registers.

Table 28 Dual timer 0 clock enable register map and reset values

## 22.10 Dual timer1\_0 Registers (Base address : 0x4000\_2000)

### Timer1\_0 Load Register(DUALTIMER1\_0TimerLoad)

Base address : 0x4000\_2000

Address offset : 0x00

Reset value : 0x0000\_0000

31	0
TLR	
R/W	

[31:0] TLR - Timer Load Register

This register contains the value from which the counter is to decrement. This is the value used to reload the counter when Periodic mode is enabled, and the current count reaches 0.

### Timer1\_0 Value Register(DUALTIMER1\_0TimerValue)

Base address : 0x4000\_2000

Address offset : 0x04

Reset value : 0xFFFF\_FFFF

31	0
TVR	
R	

[31:0] TVR - Timer Value Register

This register provides the current value of the decrementing counter.

### Timer1\_0 Control Register(DUALTIMER1\_0TimerControl)

Base address : 0x4000\_2000

Address offset : 0x08

Reset value : 0x0000\_0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TE	TM	IE	res	TP	TS	OC								

[0] OC - One-shot Count

0 : Wrapping mode, default.

1 : One-shot mode.

#### [1] TS - Timer Size

0 : 16-bit counter, default.

1 : 32-bit counter.

#### [3:2] TP - Timer Prescale.

00 : 0 stages of prescale, clock is divided by 1, default.

01 : 4 stages of prescale, clock is divided by 16.

10 : 8 stages of prescale, clock is divided by 256.

11 : Undefined, do not use.

#### [5] IE - Interrupt Enable.

0 : Timer Interrupt disable.

1 : Timer Interrupt enabled, default.

#### [6] TM - Timer Mode.

0 : Timer is in free-running mode, default.

1 : Timer is in periodic mode.

#### [7] TE - Timer Enable.

0 : Timer disabled, default.

1 : Timer enabled.

## Timer1\_0 Interrupt Clear Register (DUALTIMER1\_0TimerIntClr)

Base address : 0x4000\_2000

Address offset : 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	IC														

#### [0] IC - Interrupt Clear

Set to the this register clears the interrupt output from the counter.

## Timer1\_0 Raw Interrupt Status Register (DUALTIMER1\_0TimerRIS)

Base address : 0x4000\_2000

Address offset : 0x10

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	RIS														
															R

#### [0] RIS - Raw Interrupt Status Register

This register indicates the raw interrupt status from the counter. This value is ANDed with the timer interrupt enable bit from the Timer Control Register to create the masked interrupt, that is passed to the interrupt output pin.

### Timer1\_0 Masked Interrupt Status Register (DUALTIMER1\_0TimerMIS)

Base address : 0x4000\_2000

Address offset : 0x14

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	MIS														
															R

#### [0] MIS - Masked Interrupt Status Register

This register indicates the masked interrupt status from the counter. This value is the logical AND of the raw interrupt status with the timer interrupt enable bit from the Timer Control Register, and is the same value that is passed to the interrupt output pin.

### Timer1\_0 Background Load Register (DUALTIMER1\_0TimerBGLoad)

Base address : 0x4000\_2000

Address offset : 0x18

Reset value : 0x0000\_0000

31															0
BGL															
R/W															

#### [31:0] BGL - Background Load Register

---

This register contains the value from which the counter is to decrement. This is the value used to reload the counter when Periodic mode is enabled, and the current count reaches 0.

This register provides an alternative method of accessing the TimerLoad Register. The difference is that writes to TimerBGLoad Register do not cause the counter to immediately restart from the new value.

Reading from this register returns the same value returned from TimerLoad Register.

## 22.11 Register map

The following Table 29 summarizes the Dual timer 1\_0 registers.

Table 29 Dual timer 1\_0 register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DUALTIMER1_0TimerLoad	RES																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x04	DUALTIMER1_0TimerValue	RES																															
	reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
0x08	DUALTIMER1_0TimerControl	RES																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x0C	DUALTIMER1_0TimerIntClr	RES																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x10	DUALTIMER1_0TimerRIS	RES																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x14	DUALTIMER1_0TimerMIS	RES																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x18	DUALTIMER1_0TimerBGLoad	RES																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

## 22.12 Dual timer1\_1 Registers (Base address : 0x4000\_2020)

### Timer1\_1 Load Register(DUALTIMER1\_1TimerLoad)

Base address : 0x4000\_2020

Address offset : 0x00

Reset value : 0x0000\_0000

31	0
TLR	
R/W	

[31:0] TLR - Timer Load Register

This register contains the value from which the counter is to decrement. This is the value used to reload the counter when Periodic mode is enabled, and the current count reaches 0.

### Timer1\_1 Value Register(DUALTIMER1\_1TimerValue)

Base address : 0x4000\_2020

Address offset : 0x04

Reset value : 0xFFFF\_FFFF

31	0
TVR	
R	

[31:0] TVR - Timer Value Register

This register provides the current value of the decrementing counter.

### Timer1\_1 Control Register(DUALTIMER1\_1TimerControl)

Base address : 0x4000\_2020

Address offset : 0x08

Reset value : 0x0000\_0020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TE	TM	IE	res	TP	TS	OC								

[0] OC - One-shot Count

0 : Wrapping mode, default.

1 : One-shot mode.

#### [1] TS - Timer Size

0 : 16-bit counter, default.

1 : 32-bit counter.

#### [3:2] TP - Timer Prescale.

00 : 0 stages of prescale, clock is divided by 1, default.

01 : 4 stages of prescale, clock is divided by 16.

10 : 8 stages of prescale, clock is divided by 256.

11 : Undefined, do not use.

#### [5] IE - Interrupt Enable.

0 : Timer Interrupt disable.

1 : Timer Interrupt enabled, default.

#### [6] TM - Timer Mode.

0 : Timer is in free-running mode, default.

1 : Timer is in periodic mode.

#### [7] TE - Timer Enable.

0 : Timer disabled, default.

1 : Timer enabled.

## Timer1\_1 Interrupt Clear Register (DUALTIMER1\_1TimerIntClr)

Base address : 0x4000\_2020

Address offset : 0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	IC														

#### [0] IC - Interrupt Clear

Set to the this register clears the interrupt output from the counter.

## Timer1\_1 Raw Interrupt Status Register (DUALTIMER1\_1TimerRIS)

Base address : 0x4000\_2020

Address offset : 0x10

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	RIS														
															R

#### [0] RIS - Raw Interrupt Status Register

This register indicates the raw interrupt status from the counter. This value is ANDed with the timer interrupt enable bit from the Timer Control Register to create the masked interrupt, that is passed to the interrupt output pin.

### Timer1\_1 Masked Interrupt Status Register (DUALTIMER1\_1TimerMIS)

Base address : 0x4000\_2020

Address offset : 0x14

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	MIS														
															R

#### [0] MIS - Masked Interrupt Status Register

This register indicates the masked interrupt status from the counter. This value is the logical AND of the raw interrupt status with the timer interrupt enable bit from the Timer Control Register, and is the same value that is passed to the interrupt output pin.

### Timer1\_1 Background Load Register (DUALTIMER1\_1TimerBGLoad)

Base address : 0x4000\_2020

Address offset : 0x18

Reset value : 0x0000\_0000

31	0
----	---

BGL
R/W

#### [31:0] BGL - Background Load Register

This register contains the value from which the counter is to decrement. This is the value used to reload the counter when Periodic mode is enabled, and the current count reaches 0.

This register provides an alternative method of accessing the TimerLoad Register. The difference is that writes to TimerBGLoad Register do not cause the counter to immediately restart from the new value.

Reading from this register returns the same value returned from TimerLoad Register.

## 22.13 Register map

The following Table 30 summarizes the Dual timer 1\_1 registers.

Table 30 Dual timer 1\_1 register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DUALTIMER1_1TimerLoad	RES																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x04	DUALTIMER1_1TimerValue	RES																															
	reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x08	DUALTIMER1_1TimerControl	RES																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0C	DUALTIMER1_1TimerIntClr	RES																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x10	DUALTIMER1_1TimerRIS	RES																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x14	DUALTIMER1_1TimerMIS	RES																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x18	DUALTIMER1_1TimerBGLoad	RES																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

## 22.14 Dual Timer 1 Clock Enable Register (Base address : 0x4000\_2080)

### Timer1\_0 Clock Enable Register (TIMCLKEN1\_0)

Base address : 0x4000\_2080

Address offset : 0x00

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CE														
															R/W

[0] CE - Clock Enable Register

0 : Clock disable

1 : Clock enable

### Timer1\_1 Clock Enable Register (TIMCLKEN1\_1)

Base address : 0x4000\_2080

Address offset : 0x20

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CE														
															R/W

[0] CE - Clock Enable Register

0 : Clock disable

1 : Clock enable

## 22.15 Register map

The following Table 31 summarizes the Dual timer 1 registers.

Table 31 Dual timer 1 clock enable register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	CE
0x00	TIMCLKEN1_0	res	0	Timer1_0 Clock Enable Register																														
	reset value																														0			
0x04	TIMCLKEN1_1	res	0	Timer1_1 Clock Enable Register																														
	reset value																														0			

## 23 Watchdog timer

### 23.1 Introduction

The watchdog is based on a 32-bit down-counter that is initialized from the Reload Register, WDTLoad. The watchdog generates a regular interrupt depending on a programmed value. The counter decreases by one on each positive clock edge of watchdog clock.

The watchdog monitors the interrupt and asserts a reset request signal when the counter reaches 0 and the counter is stopped. On the next enabled watchdog clock edge, the counter is reloaded from the WDTLoad Register and the countdown sequence continues. The watchdog reasserts the reset signal if the interrupt is not cleared by the time the counter next reaches 0.

The watchdog applies a reset to a system in the event of a software failure to provide a way to recover from software crashes. Users can enable or disable the watchdog unit as required.

### 23.2 Features

- 32-bit down counter.
- Internally resets chip if not periodically reloaded.
- The watchdog timer has lock register to prevent rogue software from disabling the watchdog timer functionality.
- The watchdog timer clock(WDTCLK) and system clock(PCLK) are synchronous.

### 23.3 Functional description

#### Clock

The watchdog timer contains PCLK and WDTCLK clock inputs.

PCLK is the main APB system clock and is used by the register interface.

## Interrupt and reset request

An interrupt is generated when the counter reaches 0 and is only cleared when the interrupt clear register is accessed.

The register holds the value until the interrupt is cleared.

Reset request is asserted when the counter reaches 0 repeatedly and is not reprogrammed.

Users can mask interrupts by writing 0 to the Interrupt Enable bit in the control register.

Users can read the following from status registers:

- Raw interrupt status, before masking.
- Final interrupt status, after masking.

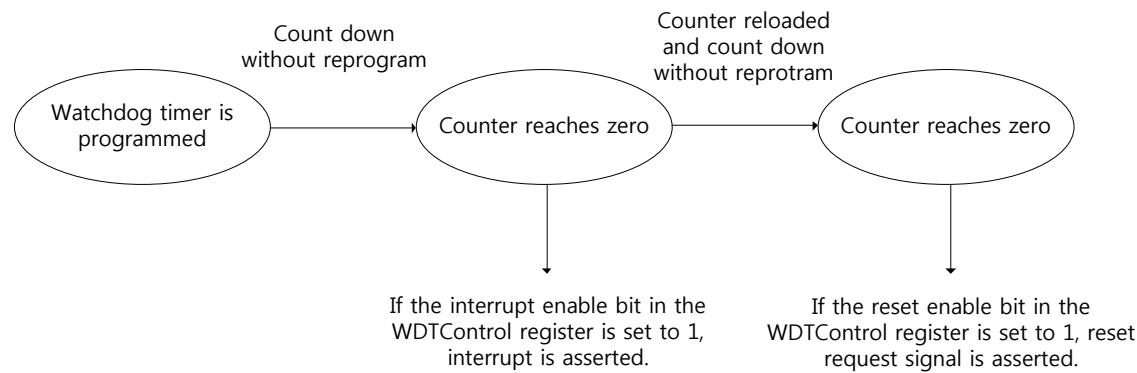


Figure 41 Watchdog timer operation flow diagram

## 23.4 Watchdog timer Registers (Base address : 0x4000\_0000)

### Watchdog timer Load Register(WDTLoad)

Address offset : 0x000

Reset value : 0xFFFF\_FFFF

31		0
	WLR	
	R/W	

[31:0] WLR - Watchdog timer Load Register.

This register contains the value from which the counter is to decrement.

When this register is written to, the count is immediately restarted from the new value. The minimum valid value for WDTLoad is 1.

## Watchdog timer Value Register(WDTValue)

Address offset : 0x004

Reset value : 0xFFFF\_FFFF

31	0
WVR	
R	

[31:0] WVR - Watchdog timer Value Register.

This register gives the current value of the decrementing counter.

## Watchdog timer Control Register(WDTControl)

Address offset : 0x008

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	REN	IEN													

[0] IEN - Interrupt Enable.

0 : Disable the counter and the interrupt.

1 : Enable the counter and the interrupt. Reloads the counter from the value in WDTLoad, after previously being disabled.

[1] REN - Reset Request Enable.

0 : Disable watchdog reset output.

1 : Enable watchdog reset output.

## Watchdog timer Interrupt Clear Register (WDTIntClr)

Address offset : 0x00C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	WIC														

---

### [0] WIC - Watchdog timer Interrupt Clear

A write of 1 to this register clears the watchdog interrupt, and reloads the counter from the value in WDTLoad.

## Watchdog timer Raw Interrupt Status Register (WDTRIS)

Address offset : 0x010

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	RIS														
															R

### [0] RIS - Watchdog timer Raw Interrupt Status.

This register indicates the raw interrupt status from the counter. This value is ANDed with the interrupt enable bit from the control register to create the masked interrupt, that is passed to the interrupt output pin.

## Watchdog timer Masked Interrupt Status Register (WDTMIS)

Address offset : 0x014

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	MIS														
															R

### [0] MIS - Watchdog timer Masked Interrupt Status.

This register indicates the masked interrupt status from the counter. This value is the logical AND of the raw interrupt status with the interrupt enable(IEN) bit from the control register, and is the same value that is passed to the interrupt output pin.

---

## Watchdog timer Lock Register(WDTLock)

Address offset : 0xC00

Reset value : 0x0000\_0000

31	1	0
ERW		WES
W		R

This register disables write accesses to all other registers. This is to prevent rogue software from disabling the watchdog functionality. Writing a value of 0x1ACCE551 enables write access to all other registers. Writing any other value disables write accesses. A read from this register returns only the bottom bit.

[0] WES - Register Write Enable status.

0 : Indicates that write access is enabled, not locked. Default.

1 : Indicates that write access is disabled, locked.

[31:1] ERW - Enable Register Writes

Enable write access to all other registers by writing 0x1ACCE551. Disable write access by writing any other value.

## 23.5 Register map

The following Table 32 summarizes the Watchdog timer registers.

Table 32 Watchdog Timer register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	WDTLoad	res																															
	reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x004	WDTValue	res																															
	reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0x008	WDTControl	res																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x00C	WDTIntClr	res																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x010	WDTRIS	res																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x014	WDTMIS	res																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0xC00	WDTLock	res																															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

## 24 Real-time Clock(RTC)

### 24.1 Introduction

The real-timer clock (RTC) is an independent BCD timer/counter. The RTC provides a timer-of-day clock/calendar with programmable alarm interrupt.

### 24.2 Features

Calendar with year, month, day, weekday, hours, minutes, and seconds

Programmable alarm with interrupt function

### 24.3 Functional description

Figure 46 shows the RTC block diagram.

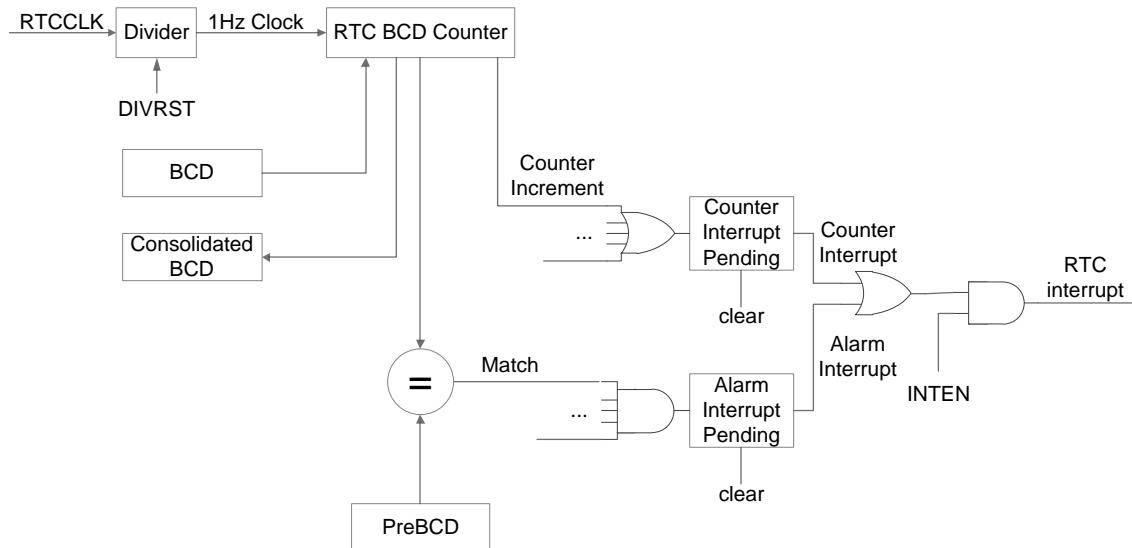


Figure 42. RTC block diagram

## RTC clock

RTC Clock (RTCCLK) can be selected among several clocks (32768Hz oscillator, MCLK, RCLK, OCLK). Please refer to the Clock Reset Generator chapter for configuring the clock. If the 32768Hz oscillator clock is used, the divider generates 1 Hz clock internally. If the DIVRST (Bit[1] of RTC Control Register ) value is high, the RTC Divider is cleared. If the DIVRST value is low, the divider operates.

## RTC interrupt

RTC has two kinds of interrupt source, Counter Interrupt and Alarm Interrupt. Interrupts are enabled in the NVIC using the appropriate Interrupt Set Enable register. There are following Counter Interrupts: Second, Minute, Hour, Day (Day of Week), Date (Day of Month) and Year. Each bit of RTCINTM (Interrupt Mask Register) can disable or enable interrupt for each Counter Interrupt. Alarm Interrupt can be generated when the Alarm matches with Counter.

## RTC Counter and Calendar

When user writes 1 to bit 0 of control register, then the RTC counter runs according to the RTC Clock.

Timer value should be written as the BCD format.

The RTC counter operates for Calendar function including Day, Month, Year and Leap Year calculation.

## RTC Setting flow

RTC setting sequences are shown in flow chart.

Figure 47 shows RTC setting flow for Counter Function.

Figure 48 shows RTC setting flow for Alarm function.

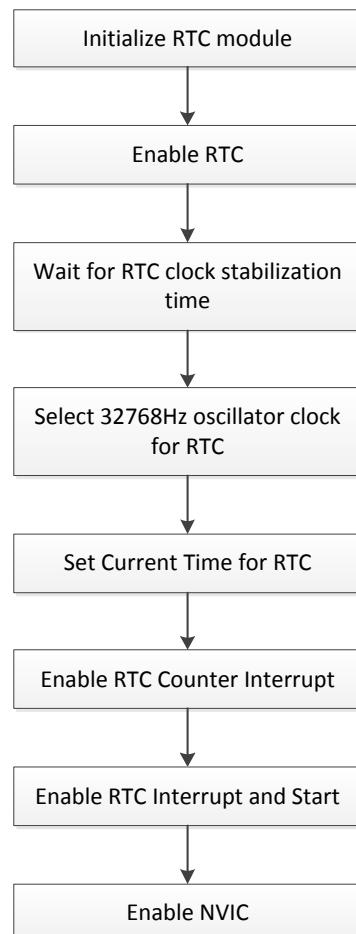


Figure 43. RTC setting flow for Counter Function

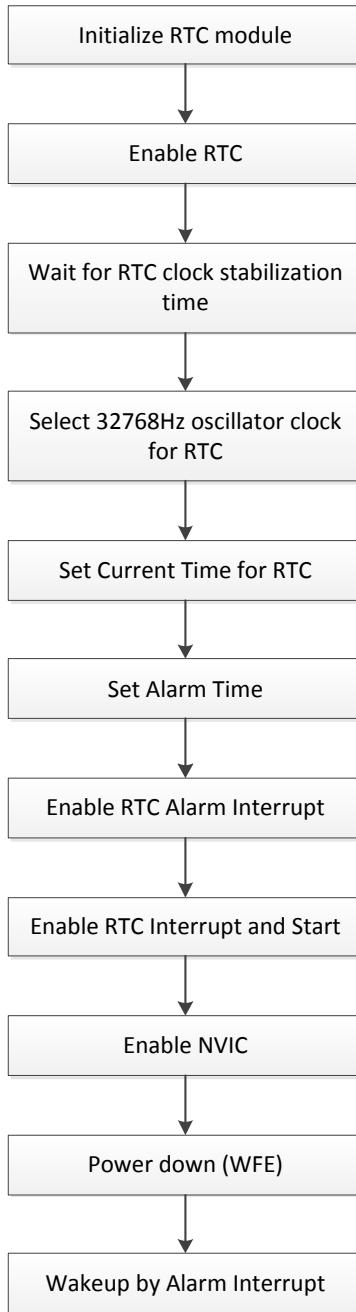


Figure 44. RTC setting flow for Alarm function

## 24.4 RTC Registers (Base Address : 0x4000\_E000)

In these descriptions, for most of the registers the Reset Value shows "NC", meaning that these registers are Not Changed by a Reset. Software must initialize these registers between power-on and setting the RTC into operation.

### RTC control register (RTCCON)

Address offset: 0x0000

Reset value: 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	INTEN	res	res	res	DIVRST	CLKEN									
										R/W				R/W	R/W

[0] CLKEN - Clock Enable

This bit written by S/W to enable or disable clock

0 : The time counters are disabled (clock stop)

1 : The time counters are enabled (clock enable)

[1] DIVRST - RTC Divider Reset

This bit set and cleared by S/W to reset RTC divider.

0 : No effect

1 : When 1, divider reset and remain reset until RTCCON[1] is changed to 0.

[5] INTEN - RTC Interrupt Enable

This bit set and cleared by S/W to enable or disable RTC interrupt.

0 : RTC Interrupt is disabled

1 : RTC Interrupt is enabled

## RTC Interrupt Mask register (RTCCON)

Address offset: 0x0004

Reset value: 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	AIN T	IMM ON	IMD AY	IMD ATE	IMH OUR	IMMI N	IMSE C								
									R/W	R/W	R/W	R/W	R/W	R/W	R/W

[0] IMSEC - RTC Second Interrupt Enable

This bit set and cleared by S/W to enable or disable RTC Second interrupt.

0 : No effect

1 : an increment of the Second value generates an interrupt

[1] IMMIN - RTC Minute Interrupt Enable

---

This bit set and cleared by S/W to enable or disable RTC Minute interrupt.

0 : No effect

1 : an increment of the Minute value generates an interrupt

[2] IMHOUR - RTC Hour Interrupt Enable

This bit set and cleared by S/W to enable or disable RTC Hour interrupt.

0 : No effect

1 : an increment of the Hour value generates an interrupt

[3] IMDATE - RTC Date Interrupt Enable

This bit set and cleared by S/W to enable or disable RTC Date interrupt.

0 : No effect

1 : an increment of the Date value generates an interrupt

[4] IMDAY - RTC Day Interrupt Enable

This bit set and cleared by S/W to enable or disable RTC Day interrupt.

0 : No effect

1 : an increment of the Day value generates an interrupt

[5] IMMON - RTC Month Interrupt Enable

This bit set and cleared by S/W to enable or disable RTC Month interrupt.

0 : No effect

1 : an increment of the Month value generates an interrupt

### [6] AINT - RTC Alarm Interrupt Enable

This bit set and cleared by S/W to enable or disable RTC Alarm interrupt.

0 : No effect

1 : Alarm interrupt enabled

## RTC Interrupt Pending register (RTCINTP)

Address offset: 0x0008

Reset value: 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	RTC ALF	RTC CIF													
														R/W	R/W

### Bit [0] RTCCIF - RTC Counter Interrupt pending flag

When one, the Counter Increment Interrupt block generated an interrupt. Writing a one to this bit clears the counter increment interrupt.

Read 0 : No effect

Read 1 : an increment of the Second value generates an interrupt

Write 0 : No effect

Write 1 : clear interrupt

### Bit [1] RTCALF - RTC Alarm interrupt pending flag

When one, the alarm registers generated an interrupt. Writing a one to this bit clears the alarm interrupt.

Read 0 : No effect

Read 1 : an increment of the Minute value generates an interrupt

Write 0 : No effect

Write 1 : clear interrupt

## RTC Alarm Mask register (RTCAMR)

Address offset: 0x000C

Reset value: 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	AMR YEA R	AMR MON	AMR DAT E	AMR DAY	AMR HOU R	AMR MIN	AMR SEC								
									R/W	R/W	R/W	R/W	R/W	R/W	R/W

### [0] AMRSEC - RTC Alarm Mask for Second

This bit set and cleared by S/W to enable or disable comparison for Second

0 : No effect

1 : the Second value is compared to the Predetermining Second for the alarm

### [1] AMRMIN - RTC Alarm Mask for Minute

This bit set and cleared by S/W to enable or disable comparison for Minute

0 : No effect

1 : the Minute value is compared to the Predetermining Minute for the alarm

### [2] AMRHOUR - RTC Alarm Mask for Hour

This bit set and cleared by S/W to enable or disable comparison for Minute

0 : No effect

1 : the Hour value is compared to the Predetermining Hour for the alarm

### [3] AMRDAY - RTC Alarm Mask for Day

This bit set and cleared by S/W to enable or disable comparison for Day of Week

0 : No effect

1 : the Day value is compared to the Predetermining Day for the alarm

### [4] AMRDATE - RTC Alarm Mask for Date

This bit set and cleared by S/W to enable or disable comparison for Date of Month

0 : No effect

1 : the Date value is compared to the Predetermining Date for the alarm

### [5] AMRMON - RTC Alarm Mask for Month

This bit set and cleared by S/W to enable or disable comparison for Date of Month

0 : No effect

1 : the Month value is compared to the Predetermining Month for the alarm

### [6] AMRYEAR - RTC Alarm Mask for Year

This bit set and cleared by S/W to enable or disable comparison for Date of Month

0 : No effect

1 : the Date value is compared to the Predetermining Date for the alarm

## RTC BCD Second register (BCDSEC)

Address offset: 0x0010

Reset value: 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res							BCDSEC								
									R/W						

Bit [6:0] BCDSEC - RTC Seconds value (0 to 59)

## RTC BCD Minute register (BCDMIN)

Address offset: 0x0014

Reset value: 0x0000\_0000

31	30	29	28	27	26	25	24		23	22	21	20	19	18	17	16
res		res														

15	14	13	12	11	10	9	8		7	6	5	4	3	2	1	0
res		res							BCDMIN							
									R/W							

Bit [6:0] BCDMIN - RTC Minute value (0 to 59)

## RTC BCD Hour register (BCDHOUR)

Address offset: 0x0018

Reset value: 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res						BCDHOUR									
									R/W						

Bit [5:0] BCDHOUR - RTC Hour value (0 to 23)

### RTC BCD Day register (BCDDAY)

Address offset: 0x001C

Reset value: 0x0000\_0000

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

Bit [3:0] BCDDAY - RTC Day of Week value (1 to 7)

### RTC BCD Date register (BCDDATE)

Address offset: 0x0020

Reset value: 0x0000\_0000

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

Bit [5:0] BCDDATE - RTC Day of Month value (1 to 28, 29, 30, or 31)

### RTC BCD Month register (BCDMON)

Address offset: 0x0024

Reset value: 0x0000\_0000

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

Bit [4:0] BCDMON - RTC Month value (1 to 12)

### RTC BCD Year register (BCDYEAR)

Address offset: 0x0028

Reset value: 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	res	res	res	res		res									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BCDYEAR														
R/W	R/W	R/W	R/W	R/W		R/W									

Bit [15:0] BCDYEAR - RTC Year value (0 to 4095)

### RTC Predetermining Second register (PRESEC)

Address offset: 0x002C

Reset value : 0xFFFF\_FFFF, NC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
res		PRESEC																						
										R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit [6:0] PRESEC - RTC PREDETERMINING Seconds value (0 to 59)

### RTC Predetermining Minute register (PREMIN)

Address offset: 0x0030

Reset value : 0xFFFF\_FFFF, NC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0									
res		PREMIN																						
										R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit [6:0] PREMIN - RTC Predetermining Minute value (0 to 59)

## RTC Predetermining Hour register (PREHOUR)

Address offset: 0x0034

Reset value : 0xFFFF\_FFFF, NC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															

Bit [5:0] PREHOUR - RTC Predetermining Hour value (0 to 23)

## RTC Predetermining Day register (PREDAY)

Address offset: 0x0038

Reset value : 0xFFFF\_FFFF, NC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															

Bit [3:0] PREDAY - RTC Predetermining Day of Week value (1 to 7)

## RTC Predetermining Date register (PREDATE)

Address offset: 0x003C

Reset value : 0xFFFF\_FFFF, NC

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res															

Bit [5:0] PREDATE - RTC Predetermining Day of Month value (1 to 28, 29, 30, or 31)

## RTC Predetermining Month register (PREMON)

Address offset: 0x0040

Reset value : 0xFFFF\_FFFF, NC

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

Bit [4:0] PREMON - RTC Predetermining Month value (1 to 12)

## RTC Predetermining Year register (PREYEAR)

Address offset: 0x0044

Reset value : 0xFFFF\_FFFF, NC

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

Bit [15:0] PREYEAR - RTC Predetermining Year value (0 to 4095)

## RTC Consolidated Time0 register (RTCTIME0)

Address offset: 0x0048

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	res	res	res	CBCDDAY				res	res	CBCDHOUR					
				R	R	R	R			R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	CBCDMIN							res	CBCDSEC						
	R	R	R	R	R	R	R		R	R	R	R	R	R	R

Bit [6:0] CBCDSEC - RTC Consolidated Second value (0 to 59)

Bit [14:8] CBCDMIN - RTC Consolidated Minute value (0 to 59)

Bit [21:16] CBCDHOUR - RTC Consolidated Hour value (0 to 23)

Bit [27:24] CBCDDAY - RTC Consolidated Day of Week value (1 to 7)

## RTC Consolidated Time1 register (RTCTIME1)

Address offset: 0x004C

Reset value : 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CBCDYEAR															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	CBCDMON					res	res	CBCDDATE					
			R	R	R	R	R			R	R	R	R	R	R

Bit [5:0] CBCDDATE - RTC Consolidated Day of Month value (1 to 28, 29, 30, or 31)

Bit [12:8] CBCDMON - RTC Consolidated Month value (1 to 12)

Bit [31:16] CBCDYEAR - RTC Consolidated Year value (0 to 4095)

## 24.5 Register map

Table 33 RTC register map and reset values

## 25 UART(Universal Asynchronous Receive Transmit)

## 25.1 Introduction

The UART supports synchronous one-way communication, half-duplex single wire communication, and multiprocessor communications(CTS/RTS).

## 25.2 Features

- Serial-to-parallel conversion on data received from a peripheral device
  - Parallel-to-serial conversion on data transmitted to the peripheral device
  - Data size of 5,6,7 and 8 bits
  - One or two stop bits
  - Even, odd, stick, or no-parity bit generation and detection
  - Support of hardware flow control
  - Programmable FIFO disabling for 1-byte depth.
  - Programmable use of UART or IrDA SIR input/output
  - False start bit detection

## 25.3 Functional description

UART bidirectional communication requires a minimum of two pins: RX, TX

The frame are comprised of:

- An Idle Line prior to transmission or reception
- A start bit
- A data word (8 or 9 bits) least significant bit first
- 1, 1.5, 2 Stop bits indicating that the frame is complete
- The USART interface uses a baud rate generator
- A status register (UART1\_RISR)
- data registers (UART1DR)
- A baud rate register (UART1\_IBRD, UART1\_FBRD)

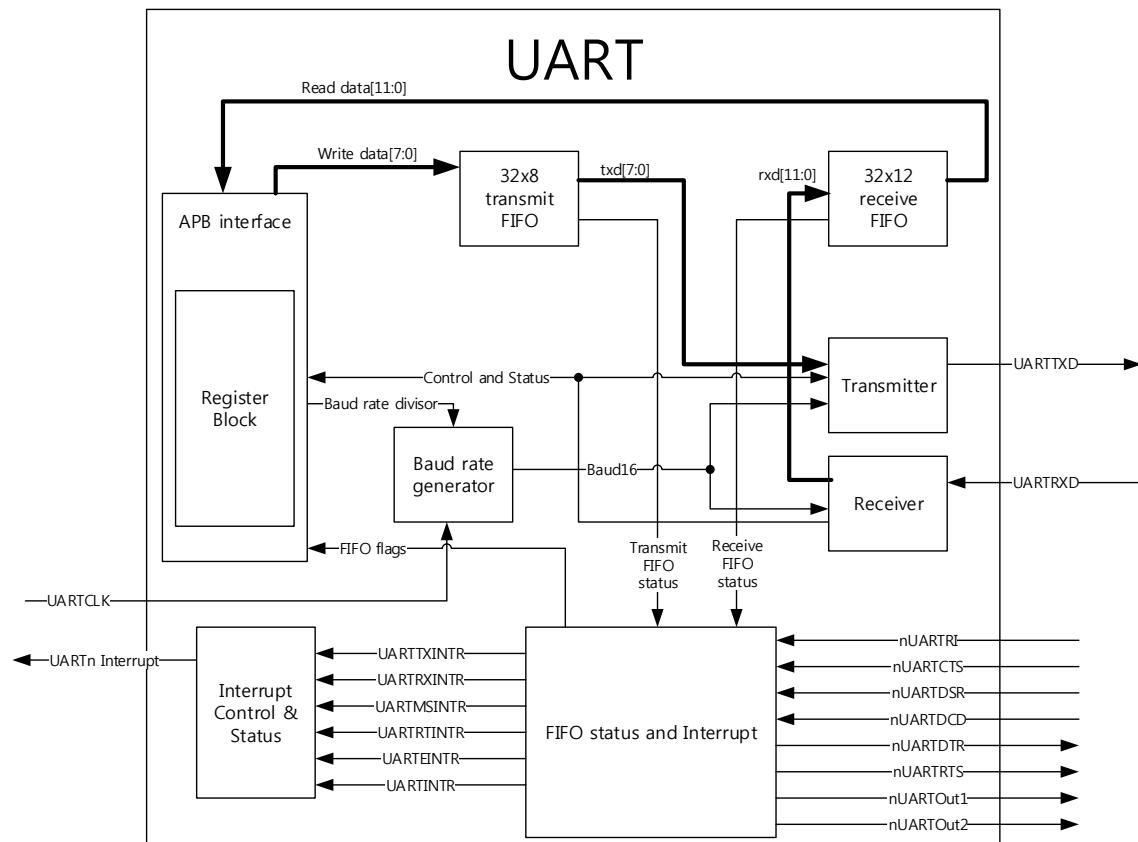
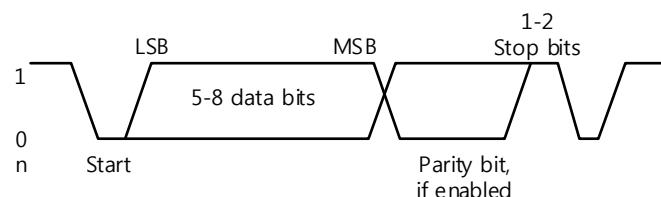


Figure 45. UART0,1 Block diagram

Figure 46 shows the UART character frame



---

Figure 46. UART character frame

## Baud rate calculation

UARTx can operate with or without using the Fractional Divider. The baud rate divisor is a 22-bit number consisting the UARTxIBRD(16-bit integer) and the UARTxFBRD(6-bit fractional).

This is used by the baud rate generator to determine the bit period.

$$- \text{ Baud Rate Divisor} = \frac{\text{UARTCLK}}{(16 * \text{baud rate})} = BRD_I + BRD_F$$

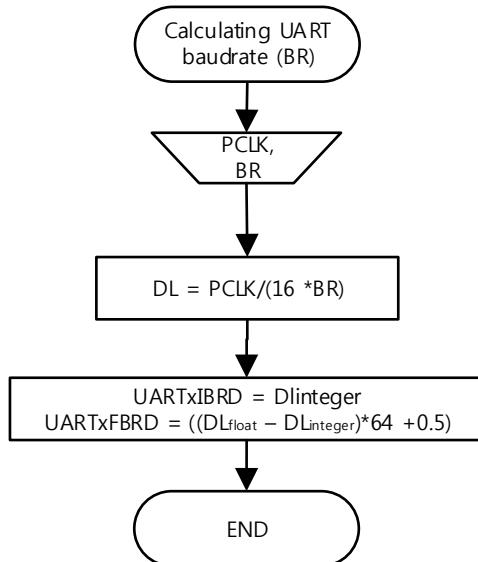


Figure 47. UART divider flow chart

Figure 48 show how to set the UART Initial value.

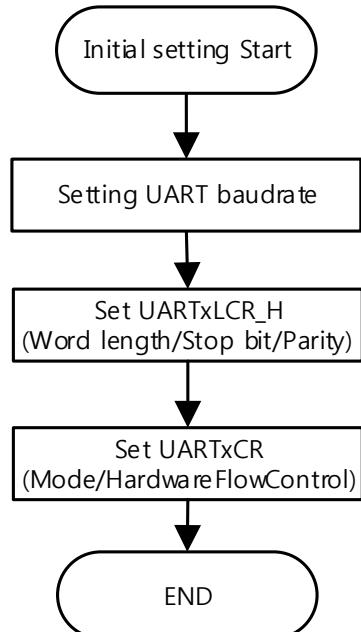


Figure 48. UART Initial setting flow chart

## Data transmission

Data transmitted is stored in a 32-byte FIFOs. Transmit data is written into the transmit FIFO for transmission. If UART is enabled, it causes a data frame to start transmitting with parameters indicated in the `UARTxLCR_H`.

Data continues to transmit until there is no data left in the transmit FIFO. The `BUSY` bit of `UARTxFR` is ‘1’ as soon as data is written to the transmit FIFO, which means the FIFO is not empty, and remains as ‘1’ while data is being transmitted.

## Data receive

Received data is stored in the 32-byte FIFOs. When a start bit has been received, it begins running and data is sampled on the eighth cycle of that counter in UART mode. A valid stop bit is confirmed if `UARTRXD` is ‘1’. When a full word is received, the data is stored in the receive FIFO. Error bit is stored in bit[10:8] of `UARTxCR` and overrun is stored in bit[11] of `UARTxCR`.

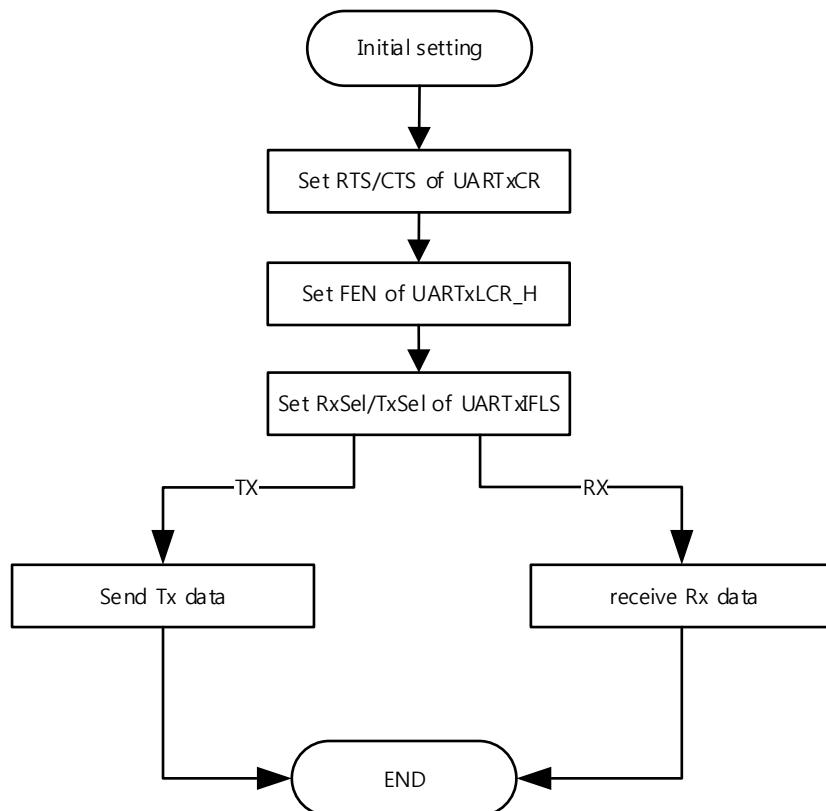


Figure 49. Transmit and Receive data flow chart

## Hardware flow control

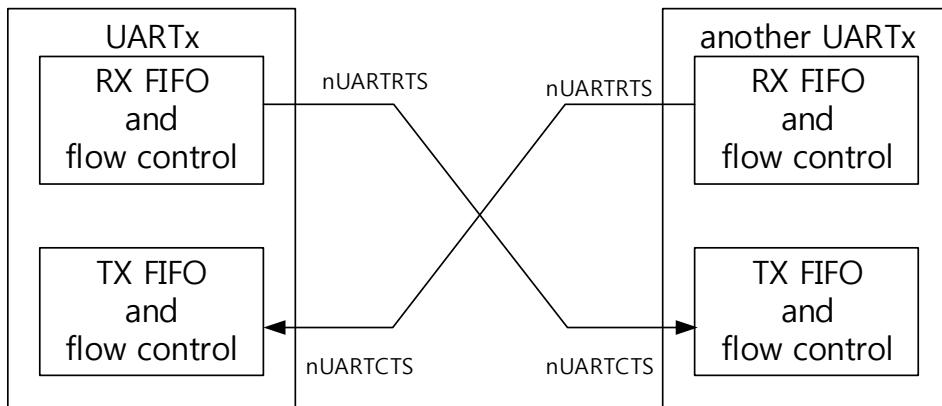


Figure 50. Hardware flow control description

The RTS flow control is enabled by setting the RTSen of UARTxCR. If RTS is enabled, the data flow is controlled as follows.

When the receiver FIFO level reaches the programmed trigger level, nUARTRTS(pin) is asserted(to a low value). nUARTRTS is reasserted(to a low level) once the receiver FIFO has reached the previous trigger level. The reasserted of nUARTRTS signals to the sending UART to continue transmitting data.

The CTS flow control is enabled, the transmitter can only transmit data when nUARTCTS is asserted. When nUARTCTR is re-asserted(to a low) the transmitter sends the next byte. To stop the transmitter from sending the following byte, nUARTCTS must be released before the middle of the last stop bit that is currently being sent.

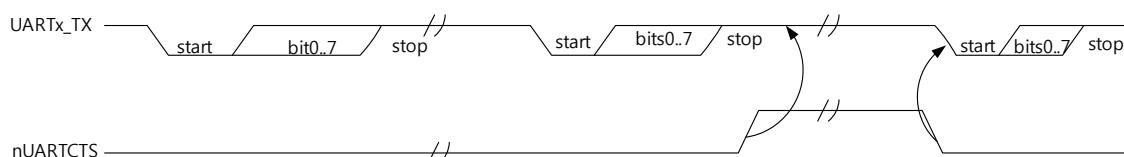


Figure 51. CTS Functional Timing

Figure 52 shows how software should use the RTS/CTR.

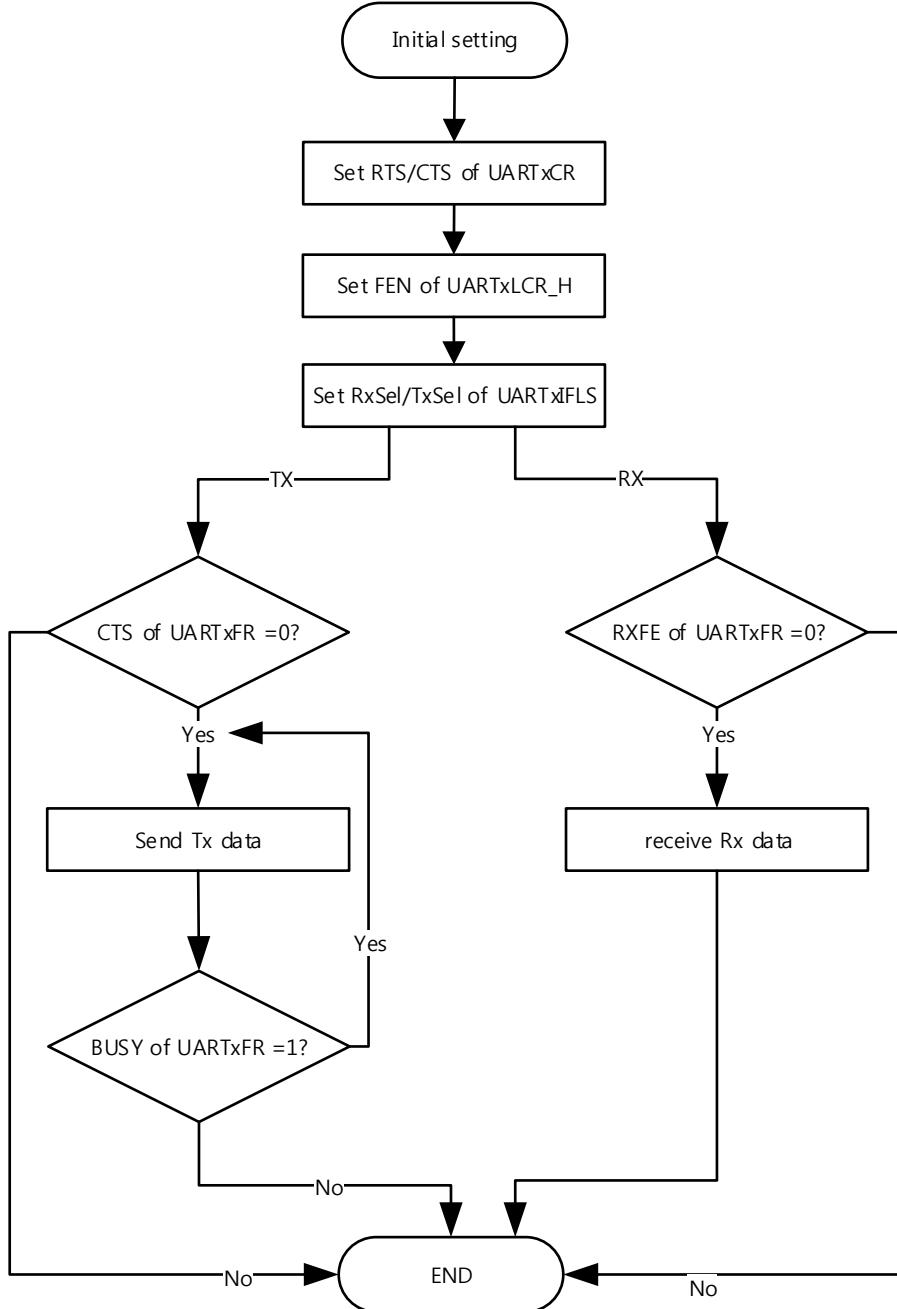


Figure 52. Algorithm for setting CTS/RTS flowchart

## 25.4 UART0 Registers(Base address: 0x4000\_C000)

### UART0DR (UART0 Data Register)

Address offset: 0x000

Reset value: 0x0000\_0000

The UART0DR is the data register.

The write operation initiates transmission from the UART. The data is prefixed with a start bit, appended with the appropriate parity bit(if parity is enabled), and a stop bit. The resultant word is then transmitted.

The received data byte is read by performing reads from the UARTDR register along with the corresponding status information.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	OE	BE	PE	FE								DATA
				R	R	R	R								R/W

#### [11] OE - Overrun error

0: data is empty

1: data is received and the receive FIFO is already full.

#### [10] BE - Break error

1: if a break condition was detected, indicating that the received data input was held LOW of longer than a full-word transmission time(defined as start, data, parity and stop bits)

#### [9] PE - Parity error

1: it indicates that the parity of the received, it indicates that the parity of the received data character does not match the parity that the EPS and SPS bits in the line control register, UARTLCR\_H

#### [8] FE - Framing error

1: it indicates that the received

#### [7:0] DATA - Receive (READ)/Transmit (WRITE) data

## UART0RSR/ECR (UART0 Receive Status Register/Error Clear Register)

Address offset: 0x004

Reset value: 0x0000\_0000

The UART0RSR/ECR is the receive status register/error clear register.

Receive status can also be read from the UART0RSR register.

A write to the UART0ECR register clears the framing, parity, break, and overrun errors.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	OE	BE	PE	FE											

### [3] OE - Overrun error

This bit is set to 1 if data is received and the FIFO is already full.

This bit is cleared to 0 by a write to UART0ECR

### [2] BE - Break error

This bit is cleared to 0 by a write to UART0ECR

### [1] PE - Parity error

When set to 1, it indicates that the parity of the received data character does not match

the parity that the EPS and SPS bits in the line control register, UARTLCR\_H select

This bit is cleared to 0 by a write to UART0ECR

### [0] FE - Framing error

When set to 1, it indicates that the received character didn't have a valid stop bit

This bit is cleared to 0 by a write to UART0ECR

## UART0FR (UART0 Flag Register)

Address offset: 0x0018

Reset value: 0bx11000xxx

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	RI	TXFE	RXFF	TXFF	RXFE	BUSY	DCD	DSR	CTS						
							R	R	R	R	R	R	R	R	R

### [8] RI - Ring indicator

This bit is the complement of the UART ring indicator, UART0RI.

1: When nUART0RI is LOW

### [7] TXFE - Transmit FIFO empty

This bit depends on the state of the FEN bit in the line control register, UARTLCR\_H.

0: The bit is set when transmit holding register is empty.

1: The bit is set when transmit FIFO is empty

---

[6] RXFF - Receive FIFO full

This bit depends on the state of the FEN bit in the line control register, UARTLCR\_H.

0: The bit is set when the receive holding register is full

1: The bit is set when the receive FIFO is full

[5] TXFF - Transmit FIFO full

This bit depends on the state of the FEN bit in the line control register, UARTLCR\_H.

0: The bit is set when transmit holding register is full.

1: The bit is set when transmit FIFO is full.

[4] RXFE - Receive FIFO empty

This bit depends on the state of the FEN bit in the line control register, UARTLCR\_H.

0: The bit is set when the receive holding register is empty.

1: The bit is set when the receive FIFO is empty.

[3] BUSY - UART busy

1: the UART is busy transmitting data

[2] DCD - Data carrier detect

This bit is the complement of the UART data carrier detect, UART0DCD, status input

1: The bit is the complement of the UART data carrier detect

[1] DSR - Data set ready

This bit is the complement of the UART data set ready, UART0DSR, status input

1: The bit is the complement of the UART data set ready

[0] CTS - Clear to send

This bit is the complement of the UART clear to send, UART0CTS, status input

1: The bit is the complement of the UART clear to send

## UART0ILPR (UART0 IrDA Low-Power Counter Register)

Address offset: 0x0020

Reset value: 0x00

The UARTILPR Register is the IrDA low-power counter register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res								ILPDVSR							

[7:0] ILPDVSR - 8-bit low-power divisor value

These bits are cleared to 0 at reset.

$$\text{ILPDVSR} = (F_{\text{UARTCLK}} / F_{\text{IrLPBaud16}})$$

Where,  $F_{\text{IrLPBaud16}}$  is nominally 1.8432MHz

The divisor is  $1.42\text{MHz} < F_{\text{IrLPBaud16}} < 2.12\text{MHz}$ , results in a low-power pulse duration of 1.41 - 2.11us.

## UART0IBRD (UART0 Integer Baud Rate Register)

Address offset: 0x0024

Reset value: 0x00

The UART0IBRD Register is the integer part of the baud rate divisor value.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res
BAUD DIVINT															
w															

[15:0] BAUD DIVINT - The integer baud rate divisor.

These bits are cleared to 0 on reset

## UART0FBRD (UART0 Fractional Baud Rate Register)

Address offset: 0x0028

Reset value: 0x00

The UART0FBRD register is the fractional part of the baud rate divisor value.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res
BAUD DIVFRAC															
w															

[5:0] BAUD DIVFRAC - The fractional baud rate divisor.

These bits are cleared to 0 on reset

The baud rate divisor is calculated as follows:

$$\text{Baud rate divisor BAUDDIV} = (F_{\text{UARTCLK}} / (16 \times \text{Baud rate}))$$

Where,  $F_{\text{UARTCLK}}$  is the UART reference clock frequency.

The BAUDDIV is comprised of the integer value (BAUD DIVINT) and the fractional value (BAUD DIVFRAC).

### Example 1

If the required baud rate is 115200 and UARTCLK = 8MHz then:

$$\text{Baud rate divisor} = (8 \times 10^6) / (16 \times 115200) = 4.340278$$

This means  $\text{BRD}_I = 4$  and  $\text{BRD}_F = 0.340278$

(Therefore,  $\text{UART0IBRD} = 4$ )

Therefore,  $\text{UART0FBRD} = \text{inerger}[(0.340278 \times 64) + 0.5] = 22$

Generated baud rate divider =  $\text{UART0IBRD} + (\text{UART0FBRD}/64) = 4.34375$

Generated baud rate =  $(8 \times 10^6) / (16 \times 4.34375) = 115107.914$

Error =  $(115108 - 115200) / (115200) \times 100 = -0.07861\%$

When UartCLK = 8MHz

Integer divisor	Fractional divisor	Required bit rate(bps)	Generated bit rate(bps)	Error%
0x2	0x0B	230400	2.171875	-0.07994
0x4	0x16	115200	4.34375	-0.07994
0x6	0x21	76800	6.515625	-0.07994
0x8	0x2C	57600	8.6875	-0.07994
0x22	0x2E	14400	34.71875	0.010001

### UART0LCR\_H (UART0 Line Control Register)

Address offset: 0x002C

Reset value: 0x00

The UART0LCR\_H register is the line control register. This register accesses bits 29 to 22 of the UART line control register, UART0LCR.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	SPS	WLEN	FEN	STP2	EPS	PEN	BRK								
								R/W							

[7] SPS - Stick parity select

0: stick parity is disable

1: either:

The parity bit is transmitted and checked as a 1 when EPS bit set '0'

The parity bit is transmitted and checked as a 0 when EPS bit set '1'

[6:5] WLEN - Word length

00	01	10	11
5 bits	6 bits	7 bits	8 bits

[4] FEN - Enable FIFO

0: The FIFO become 1-byte-deep holding register.

1: The transmit and receive FIFO buffers are enable (FIFO mode)

[3] STP2 - Two stop bit select

1: Two stop bits are transmitted at the end of the frame

[2] EPS - Even parity select

0: odd parity.

1: even parity

[1] PEN - Parity enable

0: parity is disabled and no parity bit added to the data frame

1: parity checking and generations is enabled

[0] BRK - Send break

0: For normal use, the bit must be cleared to 0

1: The low-level is continually output on the UARTRXD output

PEN	EPS	SPS	Parity bit(Transmitted or checked)
0	X	X	Not transmitted or checked
1	1	0	Even parity
1	0	0	Odd parity
1	0	1	1
1	1	1	0

## UART0CR (UART0 Control register)

Address offset: 0x0030

Reset value: 0x0300

The UART0CR register is the control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTSEn	RTSEN	Out2	Out1	RTS	DTR	RXE	TXE	res	res	res	res	SIRLP	SIREN	UARTEN	

---

[15] CTSEn - CTS hardware flow control enable.

1: CTS hardware flow control is enable. Data is only transmitted when the UART0CTS signal is asserted.

[14] RTSEn - RTS hardware flow control enable

1: RTS hardware flow control is enable. Data is only requested when there is space in the receive FIFO for it to be received.

[13] Out2

This bit is the complement of the UART Out2 (nUARTOut2) modem status output.

That is, when the bit is programmed to 1, the output is 0.

For DTE this can be used as “Ring Indicator” (RI).

[12] Out1

This bit is the complement of the UART Out1 (nUARTOut1) modem status output.

That is, when the bit is programmed to 1 the output is 0.

For DTE this can be used as “Data Carrier Detect”(DCD).

[11] RTS - Request to send

This bit is the complement of the UART request to send, UART0RTS, modem status output.

That is, when the bit is programmed to 1 then UART0RTS is LOW.

[10] DTS - Data transmit ready

This bit is the complement of the UART data transmit ready, UART0DTR, modem status output.

That is, when the bit is programmed to 1 then UART0DTR is LOW.

[9] RXE - Receive enable

If this bit is set to 1, the receive section of the UART is enabled.

Data reception occurs for either UART signals or SIR signals depending on the setting of the SIREN bit. When the UART is disabled in the middle of reception, it completes the current character before stopping.

[8] TXE - Transmit enable

1: The transmit section of the UART is enabled,

[7:3] Reserved

[2] SIRLP - SIR low-power IrDA mode

0: low-level bits are transmitted as an active high pulse with a width of 3/16th of the bit period.

1: low-level bits are transmitted with a pulse width that is 3 times the period of the IrLPBaud16 input signal, regardless of the selected bit rate.

[1] SIREN - SIR enable

0: IrDA SIR ENDEC is disable

1: IrDA SIR ENDEC is enable

---

[0] UARTEN -UART enable

0: UART is disabled

1: UART is enabled

Program the control registers as follows:

1. Disable the UART.
2. Wait for the end of transmission or reception of the current character.
3. Flush the transmit FIFO by setting the FEN bit to 0 in the Line Control Register, **UARTLCR\_H**.
4. Reprogram the **UARTCR** Register.
5. Enable the UART.

## UART0IFLS (UART0 Interrupt FIFO Level Select Register)

Address offset: 0x0034

Reset value: 0x12

The **UARTIFLS** register is the interrupt FIFO level select register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	RXIFLSEL		TXIFLSEL												
											W		W		

### [5:3] RXIFLSEL - Receive interrupt FIFO level select

7	6	5	4	3	2	1	0
Reserved		7/8 full		3/4 full		1/2 full	

### [2:0] TXIFLSEL - Transmit interrupt FIFO level select

7	6	5	4	3	2	1	0
Reserved		7/8 full		3/4 full		1/2 full	

## UART0IMSC (UART0 Interrupt Mask Set/Clear Register)

Address offset: 0x0038

Reset value: 0x00

The **UART0IMSC** register is the interrupt mask set/clear register. When a bit of **UART0IMSC** is ‘1’ and the corresponding bit of interrupt register is ‘1’, an interrupt will be issued.

In other words, if a bit of **UART0IMSC** is ‘0’, an interrupt will not be issued even if the corresponding bit of interrupt register is ‘1’.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	Res	res	res	res	OEIM	BEIM	PEIM	FEIM	RTIM	TXIM	RXIM	DSR MIM	DCD MIM	CTS MIM	RIMI M
					R/W	R/W	R/W	R/W							

[10] **OEIM** - Overrun error interrupt mask

0: Disable **UART0OEINTR**

1: Enable **UART0OEINTR**

[9] **BEIM** - Break error interrupt mask

0: Disable **UART0BEINTR**

1: Enable **UART0BEINTR**

[8] **PEIM** - Parity error interrupt mask

0: Disable **UART0EINTR**

1: Enable **UART0EINTR**

[7] **FEIM** - Framing error interrupt mask

0: Disable **UART0FEINTR**

1: Enable **UART0FEINTR**

[6] **RTIM** - Receive timeout interrupt mask

0: Disable **UART0RTINTR**

1: Enable **UART0RTINTR**

[5] **TXIM** - Transmit interrupt mask

0: Disable **UART0TXINTR**

1: Enable **UART0TXINTR**

[4] **RXIM** - Receive interrupt mask

0: Disable **UART0RXINTR**

1: Enable **UART0RXINTR**

[3] **DSRMIM** - nUART0DSR modem interrupt mask

0: Disable **UART0DSRINTR**

1: Enable **UART0DSRINTR**

[2] **DCDMIM** - nUART0DCD modem interrupt mask

0: Disable **UART0DCDINTR**

1: Enable **UART0DCDINTR**

[1] **CTSMIM** - nUART0CTS modem interrupt mask

- 0: Disable UART0CRSINTR  
 1: Enable UART0CRSINTR  
 [0] RIMIM - nUART0RI modem interrupt mask  
 0: Disable UART0RIINTR  
 1: Enable UART0RIINTR

## UART0RIS (UART0 Raw Interrupt Status Register)

Address offset: 0x003C

Reset value: 0x00-

The UART0RIS register indicates the raw interrupt status register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	res	OE RIS	BE RIS	PE RIS	FE RIS	RT RIS	TX RIS	RX RIS	DRR RMIS	DCD RMIS	CTSR MIS	RI RMIS
					R	R	R	R	R	R	R	R	R	R	R

[10] OERIS - Overrun error interrupt status

It indicates state of the UART0OEINTR interrupt.

[9] BERIS - Break error interrupt status

It indicates state of the UART0BEINTR interrupt.

[8] PERIS - Parity error interrupt status

It indicates state of the UART0PEINTR interrupt.

[7] FERIS - Framing error interrupt status

It indicates state of the UART0FEINTR interrupt.

[6] RTRIS - Receive timeout interrupt status

It indicates state of the UART0RTINTR interrupt.

[5] TXRIS - Transmit interrupt status

It indicates state of the UART0TXINTR interrupt.

[4] RXRIS - Receive interrupt status

It indicates state of the UART0RXINTR interrupt.

[3] DSRRMIS - nUART0DSR modem interrupt status

It indicates state of the UART0DSRINTR interrupt.

[2] DCDRMIS - nUART0DCD modem interrupt status

It indicates state of the UART0DCDINTR interrupt.

[1] CTSRMIS - nUART0CTS modem interrupt status

It indicates state of the UART0CTSINTR interrupt.

[0] RIRMIS - nUART0RI modem interrupt status

It indicates state of the UART0RIINTR interrupt.

## UART0MIS (UART0 Masked Interrupt Status Register)

Address offset: 0x0040

Reset value: 0x00-

The UART0MIS register is the masked interrupt status register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	res	OE MIS	BE MIS	PE MIS	FE MIS	RT MIS	TX MIS	RX MIS	DRR MMIS	DCD MMIS	CTS MMIS	RI MMIS
					R	R	R	R	R	R	R	R	R	R	R

[10] OEMIS - Overrun error masked interrupt status

It indicates state of the UART0OEINTR interrupt.

[9] BEMIS - Break error masked interrupt status

It indicates state of the UART0BEINTR interrupt.

[8] PEMIS - Parity error masked interrupt status

It indicates state of the UART0PEINTR interrupt.

[7] FEMIS - Framing error masked interrupt status

It indicates state of the UART0FEINTR interrupt.

[6] RTMIS - Receive timeout masked interrupt status

It indicates state of the UART0RTINTR interrupt.

[5] TXMIS - Transmit masked interrupt status

It indicates state of the UART0TXINTR interrupt.

[4] RXMIS - Receive masked interrupt status

It indicates state of the UART0RXINTR interrupt.

[3] DSRMMIS - nUART0DSR modem masked interrupt status

It indicates state of the UART0DSRINTR interrupt.

[2] DCDMMIS - nUART0DCD modem masked interrupt status

It indicates state of the UART0DCDINTR interrupt.

[1] CTSMMIS - nUART0CTS modem masked interrupt status

It indicates state of the UART0CTSINTR interrupt.

[0] RIMMIS - nUART0RI modem masked interrupt status

It indicates state of the UART0RIINTR interrupt.

## UART0ICR (UART0 Interrupt Clear Register)

Address offset: 0x0044

Reset value: -

The UART0ICR register is the interrupt clear register and is write-only.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	res	OEIC	BEIC	PEIC	FEIC	RTIC	TXIC	RXIC	DSR MIC	DCD MIC	CTS MIC	RI MIC
					R	R	R	R	R	R	R	R	R	R	R

[10] OEIC - Overrun error interrupt clear

Clear the UART0OEINTR interrupt.

[9] BEIC - Break error interrupt clear

Clear the UART0BEINTR interrupt.

[8] PEIC - Parity error interrupt clear

Clear the UART0PEINTR interrupt.

[7] FEIC - Framing error interrupt clear

Clear the UART0FEINTR interrupt.

[6] RTIC - Receive timeout interrupt clear

Clear the UART0RTINTR interrupt.

[5] TXIC - Transmit interrupt clear

Clear the UART0TXINTR interrupt.

[4] RXIC - Receive interrupt clear

Clear the UART0RXINTR interrupt.

[3] DSRMIC - nUART0DSR modem interrupt clear

Clear the UART0DSRINTR interrupt.

[2] DCDCMIC - nUART0DCD modem interrupt clear

Clear the UART0DCDINTR interrupt.

[1] CTSMIC - nUART0CTS modem interrupt clear

Clear the UART0CTSINTR interrupt.

[0] RIMIC - nUART0RI modem interrupt clear

Clear the UART0RIINTR interrupt.

## 25.5 Register map

The following Table 34 summarizes the UART0 registers.

Table 34 UART0 register map and reset values

## 25.6 UART1 Registers(Base address: 0x4000\_D000)

### UART1DR (UART1 Data Register)

Address offset: 0x000

Reset value: 0x0000\_0000

The UART1DR is the data register.

The write operation initiates transmission from the UART. The data is prefixed with a start bit, appended with the appropriate parity bit(if parity is enabled), and a stop bit. The resultant word is then transmitted.

The received data byte is read by performing reads from the UARTDR register along with the corresponding status information.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	OE	BE	PE	FE								DATA
				R	R	R	R								R/W

[11] OE - Overrun error

0: data is empty

1: data is received and the receive FIFO is already full.

[10] BE - Break error

1: if a break condition was detected, indicating that the received data input was held LOW of longer than a full-word transmission time(defined as start, data, parity and stop bits)

[9] PE - Parity error

1: it indicates that the parity of the received, it indicates that the parity of the received data character does not match the parity that the EPS and SPS bits in the line control register, UARLTCR\_H

[8] FE - Framing error

1: it indicates that the received

[7:0] DATA - Receive (READ)/Transmit (WRITE) data

### UART1RSR/ECR (UART1 Receive Status Register/Error Clear Register)

Address offset: 0x004

Reset value: 0x0000\_0000

The UART1RSR/ECR is the receive status register/error clear register.

Receive status can also be read from the UART1RSR register.

A write to the UART1ECR register clears the framing, parity, break, and overrun errors.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	OE	BE	PE	FE											

### [3] OE - Overrun error

This bit is set to 1 if data is received and the FIFO is already full.

This bit is cleared to 0 by a write to UART1ECR

### [2] BE - Break error

This bit is cleared to 0 by a write to UART1ECR

### [1] PE - Parity error

When set to 1, it indicates that the parity of the received data character does not match

the parity that the EPS and SPS bits in the line control register, UARTLCR\_H select

This bit is cleared to 0 by a write to UART1ECR

### [0] FE - Framing error

When set to 1, it indicates that the received character didn't have a valid stop bit

This bit is cleared to 0 by a write to UART1ECR

## UART1FR (UART1 Flag Register)

Address offset: 0x0018

Reset value: 0bx11000xxx

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	RI	TXFE	RXFF	TXFF	RXFE	BUSY	DCD	DSR	CTS						
							R	R	R	R	R	R	R	R	R

### [8] RI - Ring indicator

This bit is the complement of the UART ring indicator, UART1RI.

1: When nUART1RI is LOW

---

[7] TXFE - Transmit FIFO empty

This bit depends on the state of the FEN bit in the line control register, UARTLCR\_H.

0: The bit is set when transmit holding register is empty.

1: The bit is set when transmit FIFO is empty

[6] RXFF - Receive FIFO full

This bit depends on the state of the FEN bit in the line control register, UARTLCR\_H.

0: The bit is set when the receive holding register is full

1: The bit is set when the receive FIFO is full

[5] TXFF - Transmit FIFO full

This bit depends on the state of the FEN bit in the line control register, UARTLCR\_H.

0: The bit is set when transmit holding register is full.

1: The bit is set when transmit FIFO is full.

[4] RXFE - Receive FIFO empty

This bit depends on the state of the FEN bit in the line control register, UARTLCR\_H.

0: The bit is set when the receive holding register is empty.

1: The bit is set when the receive FIFO is empty.

[3] BUSY - UART busy

1: the UART is busy transmitting data

[2] DCD - Data carrier detect

This bit is the complement of the UART data carrier detect, UART1DCD, status input

1: The bit is the complement of the UART data carrier detect

[1] DSR - Data set ready

This bit is the complement of the UART data set ready, UART1DSR, status input

1: The bit is the complement of the UART data set ready

[0] CTS - Clear to send

This bit is the complement of the UART clear to send, UART1CTS, status input

1: The bit is the complement of the UART clear to send

## UART1ILPR (UART1 IrDA Low-Power Counter Register)

Address offset: 0x0020

Reset value: 0x00

The UARTILPR Register is the IrDA low-power counter register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res							ILPDVSR								

								R/W
--	--	--	--	--	--	--	--	-----

[7:0] ILPDVSR - 8-bit low-power divisor value

These bits are cleared to 0 at reset

$$\text{ILPDVSR} = (F_{\text{UARTCLK}} / F_{\text{IrLPBaud16}})$$

Where,  $F_{\text{IrLPBaud16}}$  is nominally 1.8432MHz

The divisor is  $1.42\text{MHz} < F_{\text{IrLPBaud16}} < 2.12\text{MHz}$ , results in a low-power pulse duration of 1.41 - 2.11us.

## UART1IBRD (UART1 Integer Baud Rate Register)

Address offset: 0x0024

Reset value: 0x00

The UART1IBRD Register is the integer part of the baud rate divisor value.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res
BAUD DIVINT															
w															

[15:0] BAUD DIVINT - The integer baud rate divisor.

These bits are cleared to 0 on reset

## UART1FBRD (UART1 Fractional Baud Rate Register)

Address offset: 0x0028

Reset value: 0x00

The UART1FBRD register is the fractional part of the baud rate divisor value.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res
BAUD DIVFRAC															
w															

[5:0] BAUD DIVFRAC - The fractional baud rate divisor.

These bits are cleared to 0 on reset

The baud rate divisor is calculated as follows:

$$\text{Baud rate divisor BAUDDIV} = (F_{\text{UARTCLK}} / (16 \times \text{Baud rate}))$$

Where,  $F_{\text{UARTCLK}}$  is the UART reference clock frequency.

The BAUDDIV is comprised of the integer value (BAUD DIVINT) and the fractional value (BAUD DIVFRAC).

### Example 1

If the required baud rate is 115200 and  $\text{UARTCLK} = 8\text{MHz}$  then:

$$\text{Baud rate divisor} = (8 \times 10^6) / (16 \times 115200) = 4.340278$$

This means  $\text{BRD}_I = 4$  and  $\text{BRD}_F = 0.340278$

(Therefore,  $\text{UART1IBRD} = 4$ )

Therefore,  $\text{UART1FBRD} = \text{inerger}[(0.340278 \times 64) + 0.5] = 22$

Generated baud rate divider =  $\text{UART1IBRD} + (\text{UART1FBRD}/64) = 4.34375$

Generated baud rate =  $(8 \times 10^6) / (16 \times 4.34375) = 115107.914$

Error =  $(115108 - 115200) / (115200) \times 100 = -0.07861\%$

When UartCLK = 8MHz

Integer divisor	Fractional divisor	Required bit rate(bps)	Generated bit rate(bps)	Error%
0x2	0x0B	230400	2.171875	-0.07994
0x4	0x16	115200	4.34375	-0.07994
0x6	0x21	76800	6.515625	-0.07994
0x8	0x2C	57600	8.6875	-0.07994
0x22	0x2E	14400	34.71875	0.010001

### UART1LCR\_H (UART1 Line Control Register)

Address offset: 0x002C

Reset value: 0x00

The UART1LCR\_H register is the line control register. This register accesses bits 29 to 22 of the UART line control register, UART1LCR.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	SPS	WLEN	FEN	STP2	EPS	PEN	BRK								
								R/W							

[7] SPS - Stick parity select

0: stick parity is disable

1: either:

The parity bit is transmitted and checked as a 1 when EPS bit set '0'

The parity bit is transmitted and checked as a 0 when EPS bit set '1'

[6:5] WLEN - Word length

00	01	10	11
5 bits	6 bits	7 bits	8 bits

[4] FEN - Enable FIFO

0: The FIFO become 1-byte-deep holding register.

1: The transmit and receive FIFO buffers are enable (FIFO mode)

[3] STP2 - Two stop bit select

1: Two stop bits are transmitted at the end of the frame

[2] EPS - Even parity select

0: odd parity.

1: even parity

[1] PEN - Parity enable

0: parity is disabled and no parity bit added to the data frame

1: parity checking and generations is enabled

[0] BRK - Send break

0: For normal use, the bit must be cleared to 0

1: The low-level is continually output on the UARTRXD output

PEN	EPS	SPS	Parity bit(Transmitted or checked)
0	X	X	Not transmitted or checked
1	1	0	Even parity
1	0	0	Odd parity
1	0	1	1
1	1	1	0

## UART1CR (UART1 Control register)

Address offset: 0x0030

Reset value: 0x0300

The UART1CR register is the control register

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

CTSEn	RTSEn	Out2	Out1	RTS	DTR	RXE	TXE	res	SIRLP	SIREN	UARTEN

[15] CTSEn - CTS hardware flow control enable.

1: CTS hardware flow control is enable. Data is only transmitted when the UART1CTS signal is asserted.

[14] RTSEn - RTS hardware flow control enable

1: RTS hardware flow control is enable. Data is only requested when there is space in the receive FIFO for it to be received.

[13] Out2

This bit is the complement of the UART Out2 (nUARTOut2) modem status output.

That is, when the bit is programmed to 1, the output is 0.

For DTE this can be used as “Ring Indicator” (RI).

[12] Out1

This bit is the complement of the UART Out1 (nUARTOut1) modem status output.

That is, when the bit is programmed to 1 the output is 0.

For DTE this can be used as “Data Carrier Detect”(DCD).

[11] RTS - Request to send

This bit is the complement of the UART request to send, UART1RTS, modem status output.

That is, when the bit is programmed to 1 then UART1RTS is LOW.

[10] DTS - Data transmit ready

This bit is the complement of the UART data transmit ready, UART1DTR, modem status output.

That is, when the bit is programmed to 1 then UART1DTR is LOW.

[9] RXE - Receive enable

If this bit is set to 1, the receive section of the UART is enabled.

Data reception occurs for either UART signals or SIR signals depending on the setting of the SIREN bit. When the UART is disabled in the middle of reception, it completes the current character before stopping.

[8] TXE - Transmit enable

1: The transmit section of the UART is enabled,

[7:3] Reserved

[2] SIRLP - SIR low-power IrDA mode

0: low-level bits are transmitted as an active high pulse with a width of 3/16th of the bit period.

1: low-level bits are transmitted with a pulse width that is 3 times the period of the IrLPBaud16 input signal, regardless of the selected bit rate.

[1] SIREN - SIR enable

0: IrDA SIR ENDEC is disable

1: IrDA SIR ENDEC is enable

[0] UARTEN - UART enable

0: UART is disabled

1: UART is enabled

Program the control registers as follows:

1. Disable the UART.
2. Wait for the end of transmission or reception of the current character.
3. Flush the transmit FIFO by setting the FEN bit to 0 in the Line Control Register, [UARTLCR\\_H](#) on page 3-12.
4. Reprogram the [UARTCR](#) Register.
5. Enable the UART.

## UART1IFLS (UART1 Interrupt FIFO Level Select Register)

Address offset: 0x0034

Reset value: 0x12

The [UARTIFLS](#) register is the interrupt FIFO level select register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	RXIFLSEL	TXIFLSEL	W	W											

[5:3] RXIFLSEL - Receive interrupt FIFO level select

7	6	5	4	3	2	1	0
Reserved			7/8 full	3/4 full	1/2 full	1/4 full	1/8 full

[2:0] TXIFLSEL - Transmit interrupt FIFO level select

7	6	5	4	3	2	1	0
Reserved			7/8 full	3/4 full	1/2 full	1/4 full	1/8 full

## UART1IMSC (UART1 Interrupt Mask Set/Clear Register)

Address offset: 0x0038

Reset value: 0x00

The UART1IMSC register is the interrupt mask set/clear interrupts. When a bit of UART1IMSC is ‘1’ and the corresponding bit of interrupt register is ‘1’, an interrupt will be issued.

In other words, if a bit of UART1IMSC is ‘0’, an interrupt will not be issued even if the corresponding bit of interrupt register is ‘1’.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	Res	res	res	res	OEIM	BEIM	PEIM	FEIM	RTIM	TXIM	RXIM	DSR MIM	DCD MIM	CTS MIM	RIMI M
					R/W	R/W	R/W	R/W							

[10] OEIM - Overrun error interrupt mask

0: Disable UART1OEINTR

1: Enable UART1OEINTR

[9] BEIM - Break error interrupt mask

0: Disable UART1BEINTR

1: Enable UART1BEINTR

[8] PEIM - Parity error interrupt mask

0: Disable UART1EINTR

1: Enable UART1EINTR

[7] FEIM - Framing error interrupt mask

0: Disable UART1FEINTR

1: Enable UART1FEINTR

[6] RTIM - Receive timeout interrupt mask

0: Disable UART1RTINTR

1: Enable UART1RTINTR

[5] TXIM - Transmit interrupt mask

0: Disable UART1TXINTR

1: Enable UART1TXINTR

[4] RXIM - Receive interrupt mask

0: Disable UART1RXINTR

1: Enable UART1RXINTR

[3] DSRMIM - nUART1DSR modem interrupt mask

0: Disable UART1DSRINTR

1: Enable UART1DSRINTR

[2] DCDMIM - nUART1DCD modem interrupt mask

0: Disable UART1DCDINTR

1: Enable UART1DCDINTR

[1] CTSMIM - nUART1CTS modem interrupt mask

0: Disable UART1CRSINTR

1: Enable UART1CRSINTR

[0] RIMIM - nUART1RI modem interrupt mask

0: Disable UART1RIINTR

1: Enable UART1RIINTR

## UART1RIS (UART1 Raw Interrupt Status Register)

Address offset: 0x003C

Reset value: 0x00-

The UART1RIS register indicates the raw interrupt status register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	res	OE RIS	BE RIS	PE RIS	FE RIS	RT RIS	TX RIS	RX RIS	DRR RMIS	DCD RMIS	CTSR MIS	RI RMIS
					R	R	R	R	R	R	R	R	R	R	R

[10] OERIS - Overrun error interrupt status

It indicates state of the UART1OEINTR interrupt.

[9] BERIS - Break error interrupt status

It indicates state of the UART1BEINTR interrupt.

[8] PERIS - Parity error interrupt status

It indicates state of the UART1PEINTR interrupt.

[7] FERIS - Framing error interrupt status

It indicates state of the UART1FEINTR interrupt.

[6] RTRIS - Receive timeout interrupt status

It indicates state of the UART1RTINTR interrupt.

[5] TXRIS - Transmit interrupt status

It indicates state of the UART1TXINTR interrupt.

[4] RXRIS - Receive interrupt status

It indicates state of the UART1RXINTR interrupt.

[3] DSRRMIS - nUART1DSR modem interrupt status

It indicates state of the UART1DSRINTR interrupt.

[2] DCDRMIS - nUART1DCD modem interrupt status

It indicates state of the UART1DCDINTR interrupt.

[1] CTSRMIS - nUART1CTS modem interrupt status

It indicates state of the UART1CTSINTR interrupt.

[0] RIRMIS - nUART1RI modem interrupt status

It indicates state of the UART1RIINTR interrupt.

## UART1MIS (UART1 Masked Interrupt Status Register)

Address offset: 0x0040

Reset value: 0x00-

The UART1MIS register is the masked interrupt status register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	res	OE MIS	BE MIS	PE MIS	FE MIS	RT MIS	TX MIS	RX MMIS	DRR MMIS	DCD MMIS	CTS MMIS	RI MMIS
					R	R	R	R	R	R	R	R	R	R	R

[10] OEMIS - Overrun error masked interrupt status

It indicates state of the UART1OEINTR interrupt.

[9] BEMIS - Break error masked interrupt status

It indicates state of the UART1BEINTR interrupt.

[8] PEMIS - Parity error masked interrupt status

It indicates state of the UART1PEINTR interrupt.

[7] FEMIS - Framing error masked interrupt status

It indicates state of the UART1FEINTR interrupt.

[6] RTMIS - Receive timeout masked interrupt status

It indicates state of the UART1RTINTR interrupt.

[5] TXMIS - Transmit masked interrupt status

It indicates state of the UART1TXINTR interrupt.

[4] RXMIS - Receive masked interrupt status

It indicates state of the UART1RXINTR interrupt.

[3] DSRMMIS - nUART1DSR modem masked interrupt status

It indicates state of the UART1DSRINTR interrupt.

[2] DCDMMIS - nUART1DCD modem masked interrupt status

It indicates state of the UART1DCDINTR interrupt.

[1] CTSMMIS - nUART1CTS modem masked interrupt status

It indicates state of the UART1CTSINTR interrupt.

[0] RIMMIS - nUART1RI modem masked interrupt status

It indicates state of the UART1RIINTR interrupt.

## UART1ICR (UART1 Interrupt Clear Register)

Address offset: 0x0044

Reset value: -

The UART1ICR register is the interrupt clear register and is write-only.

31	30	29	28	27	26	25	24	23		22	21	20	19	18	17	16
res		res														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	res	res	res	res	OEIC	BEIC	PEIC	FEIC	RTIC	TXIC	RXIC	DSRM IC	DCDM IC	CTSM IC	RI MIC
					R	R	R	R	R	R	R	R	R	R	R

[10] OEIC - Overrun error interrupt clear

Clear the UART1OEINTR interrupt.

[9] BEIC - Break error interrupt clear

Clear the UART1BEINTR interrupt.

[8] PEIC - Parity error interrupt clear

Clear the UART1PEINTR interrupt.

[7] FEIC - Framing error interrupt clear

Clear the UART1FEINTR interrupt.

[6] RTIC - Receive timeout interrupt clear

Clear the UART1RTINTR interrupt.

[5] TXIC - Transmit interrupt clear

Clear the UART1TXINTR interrupt.

[4] RXIC - Receive interrupt clear

Clear the UART1RXINTR interrupt.

[3] DSRMIC - nUART1DSR modem interrupt clear

Clear the UART1DSRINTR interrupt.

[2] DCDMIC - nUART1DCD modem interrupt clear

Clear the UART1DCDINTR interrupt.

[1] CTSMIC - nUART1CTS modem interrupt clear

Clear the UART1CTSINTR interrupt.

[0] RIMIC - nUART1RI modem interrupt clear

Clear the UART1RIINTR interrupt.

## 25.7 Register map

The following Table 35 summarizes the UART1 registers.

Table 35 UART1 register map and reset values

## 26 Universal Asynchronous Receive Transmit(UART2)

### 26.1 Introduction

The UART2 supports Asynchronous one-way communication, half-duplex single wire communication, and without CTS/RTS. UART2 is called the simple UART.

### 26.2 Feature

- Serial-to-parallel conversion on data received from a peripheral device
- Parallel-to-serial conversion on data transmitted to the peripheral device

### 26.3 Functional description

UART bidirectional communication requires a minimum of two pins: RX, TX

The frame are comprised of:

- An Idle Line prior to transmission or reception
- The UART interface uses a baud rate generator
- A status register (UART2\_SR)
- data registers (UART2DR)
- A baud rate register (UART2\_B,UART1\_FBRD)

#### Baud rate calculation

UART2 can operate with or without using the Fractional Divider. The baud rate divisor is a 20-bit number consisting the UART2BDR.

This is used by the baud rate generator to determine the bit period.

$$- \text{ Baud Rate Divisor} = \frac{\text{UARTCLK}}{(\text{baud rate})}$$

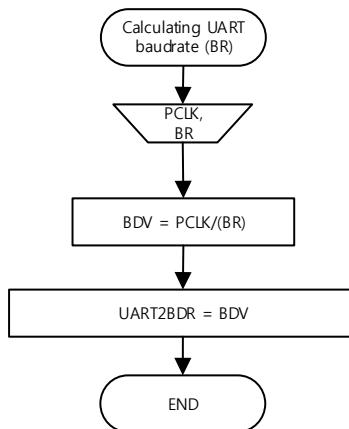


Figure 53. UART2 divider flow chart

---

Figure 54 show how to set the UART Initial value.

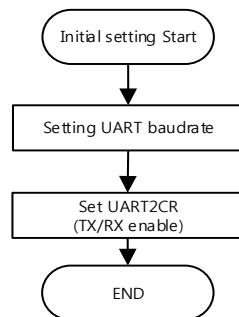


Figure 54. UART2 Initial setting flow chart

## 26.4 UART2 Registers(Base address: 0x4000\_6000)

### UART2DR (UART2 Data Register)

Address offset: 0x000

Reset value: 0x0000\_0000

The UART2DR is the data register.

The write operation initiates transmission from the UART2. The received data byte is read by performing reads from the UART2DR register along with the corresponding status information.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res							DATA								
															R/W

[7:0] DATA - Receive (READ)/Transmit (WRITE) data

### UART2SR (UART2 Status Register)

Address offset: 0x004

Reset value: 0x0000\_0000

The UART2SR is the status register. RXBF and TXBF is read only.

A write to the UART2SR register clears the TX/RX overrun errors.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	RXO	TXO	RXBF												
													R/W	R/W	R

[3] RXO -Receive Overrun

[2] TXO - Transmit Overrun

[1] RXBF - Receive buffer full

Read Only

0: The bit is set when the receive holding register is full

1: The bit is set when the receive buffer is full

[0] TXBF - Transmit buffer full

### Read Only

0: The bit is set when transmit holding register is full.

1: The bit is set when transmit buffer is full.

## UART2CR (UART2 Control Register)

Address offset: 0x008

Reset value: 0x0000\_0000

The UART2CR is the control register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	ROIE	TOIE	RXIE	TXIE	RXE	TXE									
										R/W	R/W	R/W	R/W	R/W	R/W

[5] ROIE - Receive Overrun Interrupt Enable

[4] TOIE - Transmit Overrun Interrupt Enable

[3] RXIE - Receive Interrupt Enable

[2] TXIE - Transmit Interrupt Enable

[1] RXE - Receive enable

If this bit is set to 1, the receive section of the UART2 is enabled.

When the UART2 is disabled in the middle of reception, it completes the current character before stopping.

[0] TXE - Transmit enable

If this bit is set to 1, the transmit section of the UART2 is enabled

## UART2ISR/ICR (UART2 Interrupt Status/Interrupt Clear Register)

Address offset: 0x00C

Reset value: 0x0000\_0000

The UART2ISR is the interrupt status register.

A write to the UART2ISR register clear to interrupts.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	RXOI	TXOI	RXI	TXI											
												R/W	R/W	R/W	R/W

[3] RXOI -Receive Overrun Interrupt

This bit depends on the state of the ROIE bit in the control register, UART2CR

[2] TXOI - Transmit Overrun Interrupt

This bit depends on the state of the TOIE bit in the control register, UART2CR

[1] RXI - Receive Interrupt

This bit depends on the state of the RXIE bit in the control register, UART2CR

[0] TXI - Transmit Interrupt

This bit depends on the state of the TXIE bit in the control register, UART2CR

## UART2BDR (UART2 Baud Rate Divider Register)

Address offset: 0x010

Reset value: 0x0000\_0000

The UART2BDR r is the integer part of the baud rate divisor value.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	R/W														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BAUDDIV															
R/W															

[19:0] Baud rate divider

The minimum number is 16. These bits are cleared to 0 on reset

### Example 1

If the required baud rate is 115200 and UARTCLK = 8MHz then:

$$\text{Baud rate divisor BAUDDIV} = (F_{\text{UARTCLK}} / \text{Baud rate})$$

$$\text{Baud rate divisor} = (8 \times 10^6) / (115200) = 69.4$$

This means BRD = 69

## 26.5 Register map

The following Table 36 summarizes the UART2 registers.

Table 36 UART2 register map and reset values

---

## 27 Synchronous Serial Port (SSP)

### 27.1 Introduction

The SSP block is an IP provided by ARM (PL022 “PrimeCell® Synchronous Serial Port”).

Additional details about its functional blocks may be found in “ARM PrimeCell® Synchronous Serial Port (PL022) Technical Reference Manual”.

### 27.2 Features

- The SSP is a master or slave interface that enables synchronous serial communication with slave or master peripherals having one of the following:
  - A MOTOROLA SPI-compatible interface
  - A TEXAS INSTRUMENTS synchronous serial interface
  - A National Semiconductor MICROWIRE® interface.
- The SPI interface operates as a master or slave interface. It supports bit rates up to 2 MHz and higher in both master and slave configurations. The SPI has the following features:
  - Parallel-to-serial conversion on data written to an internal 16-bit wide, 8-location deep transmit FIFO
  - Serial-to-parallel conversion on received data, buffering it in a 16-bit wide, 8-location deep receive FIFO
  - Programmable data frame size from 4 to 16 bits
  - Programmable clock bit rate and prescaler. The input clock may be divided by a factor of 2 to 254 in steps of two to provide the serial output clock
  - Programmable clock phase and polarity.

## 27.3 Functional description

Figure 55 shows the SSP block diagram.

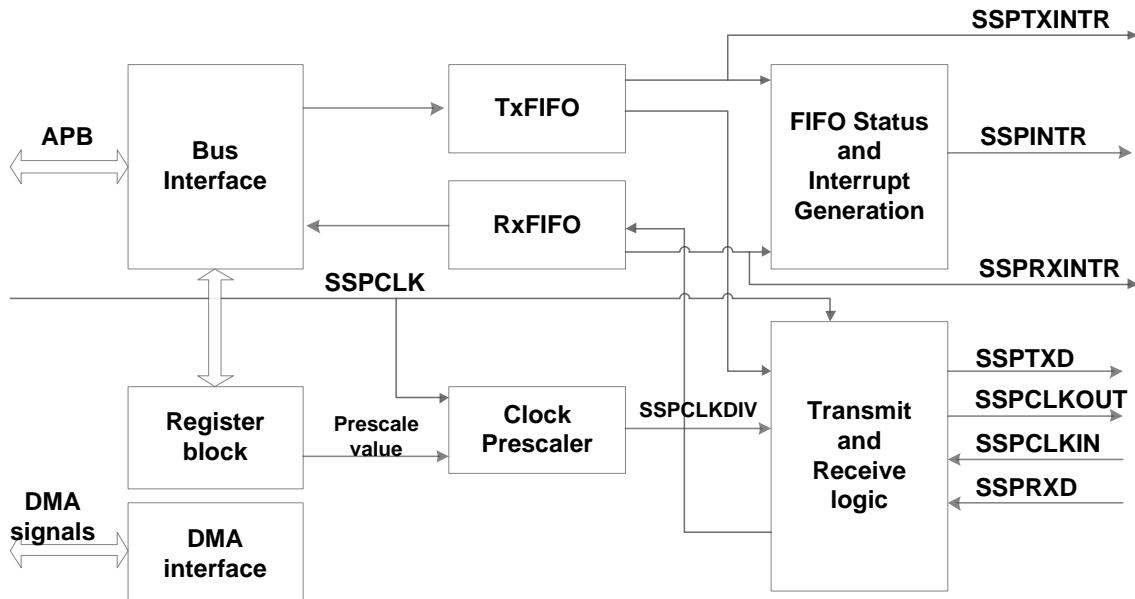


Figure 55. SSP block diagram

### Clock prescaler

When configured as a master, an internal prescaler is used to provide the serial output clock. The prescaler may be programmed through the SSPCPSR register to divide the SSPCLK by a factor of 2 to 254 in two steps. As the least significant bit of the SSPCPSR register is not used, division by an odd number is impossible and this ensures a symmetrical (equal mark space ratio) clock is generated.

The output of this prescaler is further divided by a factor 1 to 256 through the programming of the SSPCR0 control register, to give a final master output clock.

### Transmit FIFO

The common transmit FIFO is a 16-bit wide, 8-locations deep, First-In, First-Out (FIFO) memory buffer. CPU data written across the AMBA APB interface are stored in the buffer until it is read out by the transmit logic.

When configured as a master or a slave, parallel data is written into the transmit FIFO prior to serial conversion and is transmitted to the attached slave or master through the SSPTXD pin.

---

## Receive FIFO

The common receive FIFO is a 16-bit wide, 8-locations deep, first-in, first-out memory buffer. Received data from the serial interface are stored in the buffer until it is read out by the CPU across the AMBA APB interface.

When configured as a master or slave, serial data received through the SSPRXD pin is registered prior to parallel loading into the attached slave or master receive FIFO.

### Interrupt generation logic

The PrimeCell SSP generates four individual maskable, active-HIGH interrupts. A combined interrupt output is also generated as an OR function of the individual interrupt requests.

Users can use the single combined interrupt with a system interrupt controller that provides another level of masking on a per-peripheral basis. This enables use of modular device drivers that always know where to find the interrupt source control register bits.

Users can also use the individual interrupt requests with a system interrupt controller that provides masking for the outputs of each peripheral. In this way, a global interrupt controller service routine can read the entire set of sources from one wide register in the system interrupt controller. This is attractive when the time to read from the peripheral registers is significant compared to the CPU clock speed in a real-time system.

The peripheral supports both methods above.

The transmit and receive dynamic data-flow interrupts, SSPTXINTR and SSPRXINTR, are separated from the status interrupts so that data can be read or written in response to the FIFO trigger levels.

### DMA interface

The PrimeCell SSP provides an interface to connect to the DMA controller. The PrimeCell SSP DMA control register, SSPDMACR controls the DMA operation of the PrimeCell SSP.

Receive - The DMA interface includes the following signals for receive:

- SSPRXDMASREQ
  - Single-character DMA transfer request asserted by the SSP. This signal is asserted when the receive FIFO contains at least one character.
- SSPRXDMABREQ

- Burst DMA transfer request, asserted by the SSP. This signal is asserted when the receive FIFO contains four or more characters.
- SSPrXDMACLR
  - DMA request clear asserted by the DMA controller to clear the receive request signals. If DMA burst transfer is requested, the clear signal is asserted during the transfer of the last data in the burst.

Transmit - The DMA interface includes the following signals for transmit:

- SSPTXDMASREQ
  - Single-character DMA transfer request asserted by the SSP. This signal is asserted when there is at least one empty location in the transmit FIFO.
- SSPTXDMABREQ
  - Burst DMA transfer request asserted by the SSP. This signal is asserted when the transmit FIFO contains four characters or fewer.
- SSPTXDMACLR
  - DMA request clear asserted by the DMA controller to clear the transmit request signals. If a DMA burst transfer is requested, the clear signal is asserted during the transfer of the last data in the burst.

The burst transfer and single transfer request signals are not mutually exclusive. They can both be asserted at the same time. For example, when there is more data than the watermark level of four in the receive FIFO, the burst transfer request and the single transfer request are asserted.

When the amount of data left in the receive FIFO is less than the watermark level, the single request only is asserted. This is useful for situations when the number of characters left to be received in the stream is less than a burst.

For example, if 19 characters must be received, the DMA controller then transfers four bursts of four characters and three single transfers to complete the stream.

The PrimeCell SSP does not assert the burst request for the remaining three characters.

Each request signal remains asserted until the relevant DMA clear signal is asserted. After the request clear signal is de-asserted, a request signal can become active again depending on the conditions that previous sections describe. All request signals are de-asserted if the PrimeCell SSP is disabled or the DMA enable signal is cleared.

Table 37 shows the trigger points for DMABREQ of both the transmit and receive FIFOs.

Table 37 DMA trigger points for the transmit and receive FIFOs.

---

Burst length

Watermark level	Transmit, number of empty locations	Receive, number of filled locations
1/2	4	4

Figure 56 shows the timing diagram for both a single transfer request, and a burst transfer request, with the appropriate DMA clear signal. The signals are all synchronous to PCLK.

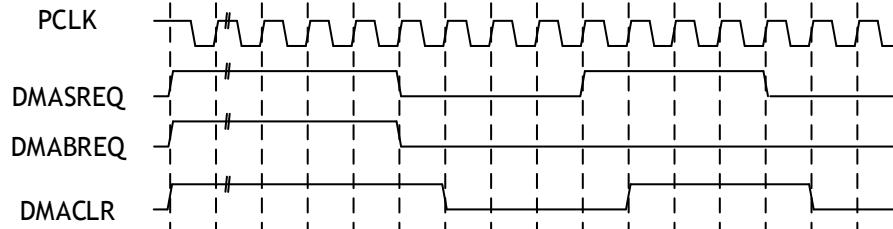


Figure 56. DMA transfer waveforms

## Interface reset

The PrimeCell SSP is reset by the global reset signal, PRESETn, and a block-specific reset signal, nSSPRST. An external reset controller must use PRESETn to assert nSSPRST asynchronously and negate it synchronously to SSPCLK. PRESETn must be asserted LOW for a period long enough to reset the slowest block in the on-chip system, and then taken HIGH again. The PrimeCell SSP requires PRESETn to be asserted LOW for at least one period of PCLK.

## Configuring the SSP

The Following reset, the PrimeCell SSP logic is disabled and must be configured when in this state.

It is necessary to program control registers SSPCR0 and SSPCR1 to configure the peripheral as a master or slave operating under one of the following protocols:

- Motorola SPI
- Texas Instruments SSI
- National Semiconductor.

The bit rate, derived from the external SSPCLK, requires the programming of the clock prescale register SSPCPSR.

## Enable PrimeCell SSP operation

You can either prime the transmit FIFO, by writing up to eight 16-bit values when the PrimeCell SSP is disabled, or permit the transmit FIFO service request to interrupt the CPU. Once enabled, transmission or reception of data begins on the transmit, SSPTXD, and receive, SSPRXD, pins.

## Clock ratios

There is a constraint on the ratio of the frequencies of PCLK to SSPCLK. The frequency of SSPCLK must be less or equal to that of PCLK. This ensures that control signals from the SSPCLK domain to the PCLK domain are guaranteed to get synchronized before one frame duration:  
 $F_{SSPCLK} \leq F_{PCLK}$ .

In the slave mode of operation, the SSPCLKIN signal from the external master is double-synchronized and then delayed to detect an edge. It takes three SSPCLKs to detect an edge on SSPCLKIN. SSPTXD has less setup time to the falling edge of SSPCLKIN on which the master is sampling the line.

The setup and hold times on SSPRXD, with reference to SSPCLKIN, must be more conservative to ensure that it is at the right value when the actual sampling occurs within the SSPMS. To ensure correct device operation, SSPCLK must be at least 12 times faster than the maximum expected frequency of SSPCLKIN.

The frequency selected for SSPCLK must accommodate the desired range of bit clock rates. The ratio of minimum SSPCLK frequency to SSPCLKOUT maximum frequency in the case of the slave mode is 12, and for the master mode, it is two.

To generate a maximum bit rate of 1.8432Mbps in the master mode, the frequency of SSPCLK must be at least 3.6864MHz. With an SSPCLK frequency of 3.6864MHz, the SSPCPSR register must be programmed with a value of 2, and the SCR[7:0] field in the SSPCR0 register must be programmed with a value of 0.

To work with a maximum bit rate of 1.8432Mbps in the slave mode, the frequency of SSPCLK must be at least 22.12MHz. With an SSPCLK frequency of 22.12MHz, the SSPCPSR register can be programmed with a value of 12 and the SCR[7:0] field in the SSPCR0 register can be programmed with a value of 0. Similarly, the ratio of SSPCLK maximum frequency to SSPCLKOUT minimum frequency is 254 x 256.

The minimum frequency of SSPCLK is calculated by the following equations, both of which must be satisfied:

$$F_{SSPCLK(min)} \Rightarrow 2 \times F_{SSPCLKOUT(max)}, \text{ for master mode}$$

---

FSSPCLK(min) => 12 x FSSPCLKIN(max), for slave mode.

The maximum frequency of SSPCLK is calculated by the following equations, both of which must be satisfied:

FSSPCLK(max) <= 254 x 256 x FSSPCLKOUT(min), for master mode

FSSPCLK(max) <= 254 x 256 x FSSPCLKIN(min), for slave mode.

## Programming the SSPCR0 Control Register

The SSPCR0 register is used to:

- program the serial clock rate
- select one of the three protocols
- select the data word size, where applicable.

The Serial Clock Rate (SCR) value in conjunction with the SSPCPSR clock prescale divisor value, CPSDVSR, is used to derive the PrimeCell SSP transmit and receive bit rate from the external SSPCLK.

The frame format is programmed through the FRF bits and the data word size through the DSS bits.

Bit phase and polarity applicable to Motorola SPI format only are programmed through the SPH and SPO bits.

## Programming the SSPCR1 Control Register

The SSPCR1 register is used to:

- select master or slave mode
- enable a loop back test feature
- enable the PrimeCell SSP peripheral.

To configure the PrimeCell SSP as a master, clear the SSPCR1 register master or slave selection bit, MS, to 0. This is the default value on reset.

Setting the SSPCR1 register MS bit to 1 configures the PrimeCell SSP as a slave. When configured as a slave, enabling or disabling of the PrimeCell SSP SSPTXD signal is provided through the SSPCR1 slave mode SSPTXD output disable bit, SOD. You can use this in some multi-slave environments where masters might parallel broadcast.

---

Set the Synchronous Serial Port Enable (SSE) bit to 1 to enable the operation of the PrimeCell SSP.

### Bit rate generation

The serial bit rate is derived by dividing down the input clock SSPCLK. The clock is first divided by an even prescale value CPSDVSR in the range of 2-254, and is programmed in SSPCPSR. The clock is divided again by a value in the range of 1-256, that is 1 + SCR, where SCR is the value programmed in SSPCR0.

The following equation defines the frequency of the output signal bit clock, SSPCLKOUT:

$$F_{SSPCLKOUT} = \frac{F_{SSPCLK}}{CPSDVSR \times (1 + SCR)}$$

For example, if SSPCLK is 3.6864MHz, and CPSDVSR = 2, then SSPCLKOUT has a frequency range of 7.2kHz-1.8432MHz.

### Recommendation

The SSPCLKOUT can be 24MHz as Max frequency when SSPCLK is 48MHz. But 20MHz is recommended as SSPCLKOUT Max frequency for stable CLK output.

## Frame format

Each data frame is between 4-16 bits long depending on the size of data programmed and is transmitted starting with the MSB. Users can select the following basic frame types:

- Texas Instruments synchronous serial
- Motorola SPI
- National Semiconductor Microwire.

For all formats, the serial clock SSPCLKOUT is held inactive while the PrimeCell SSP is idle and transitions at the programmed frequency only during active transmission or reception of data. The idle state of SSPCLKOUT is utilized to provide a receive timeout indication that occurs when the receive FIFO still contains data after a timeout period.

For Motorola SPI and National Semiconductor Microwire frame formats, the serial frame SSPFSSOUT pin is active-LOW and is asserted and pulled-down during the entire transmission of the frame.

For Texas Instruments synchronous serial frame format, the SSPFSSOUT pin is pulsed for one serial clock period starting at its rising edge prior to the transmission of each frame. For this frame format, both the PrimeCell SSP and the off-chip slave device drive their output data on the rising edge of SSPCLKOUT and latch data from the other device on the falling edge.

Unlike the full-duplex transmission of the other two frame formats, the National Semiconductor Microwire format uses a special master-slave messaging technique which operates at half-duplex. In this mode, an 8-bit control message is transmitted to the off-chip slave when a frame begins. During this transmit, the SSP receives no incoming data. After the message has been sent, the off-chip slave decodes it and responds with the requested data after waiting one serial clock after the last bit of the 8-bit control message has been sent. The returned data can be 4-16 bits in length making the total frame length in the range of 13-25 bits.

## Texas Instruments synchronous serial frame format

Figure 57 shows the Texas Instruments synchronous serial frame format for a single transmitted frame.

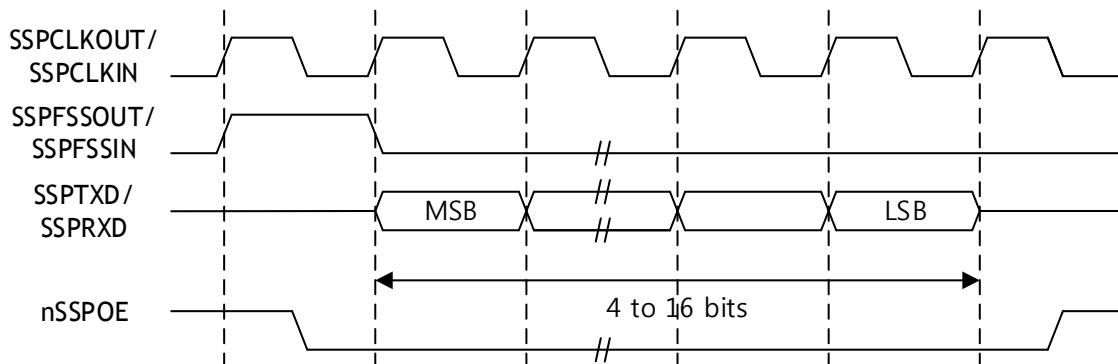


Figure 57. Texas Instruments synchronous serial frame format, single transfer

In this mode, SSPCLKOUT and SSPFSSOUT are forced LOW and the transmit data line SSPTXD is tristated whenever the PrimeCell SSP is idle. When the bottom entry of the transmit FIFO contains data, SSPFSSOUT is pulsed HIGH for one SSPCLKOUT period. The value to be transmitted is also transferred from the transmit FIFO to the serial shift register of the transmit logic. On the next rising edge of SSPCLKOUT, the MSB of the 4-bit to 16-bit data frame is shifted out on the SSPTXD pin. In a similar way, the MSB of the received data is shifted onto the SPRXD pin by the off-chip serial slave device.

Both the PrimeCell SSP and the off-chip serial slave device then clock each data bit into their serial shifter on the falling edge of each SSPCLKOUT. The received data is transferred from the serial shifter to the receive FIFO on the first rising edge of PCLK after the LSB has been latched.

Figure 58 shows the Texas Instruments synchronous serial frame format when back-to-back frames are transmitted.

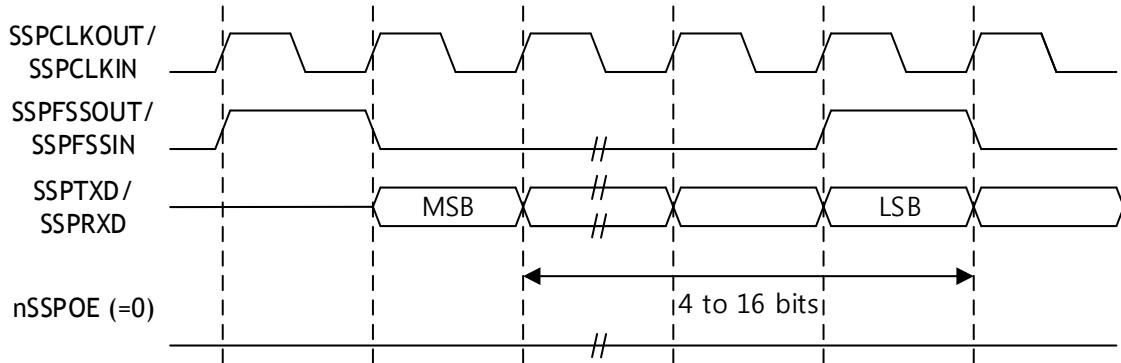


Figure 58. Texas Instruments synchronous serial frame format, continuous transfers

## Motorola SPI frame format

The Motorola SPI interface is a four-wire interface where the SSPFSSOUT signal behaves as a slave select. The main feature of the Motorola SPI format is that you can program the inactive state and phase of the SSPCLKOUT signal using the SPO and SPH bits of the SSPSCR0 control register.

### SPO, clock polarity

When the SPO clock polarity control bit is LOW, it produces a steady state LOW value on the SSPCLKOUT pin. If the SPO clock polarity control bit is HIGH, a steady state HIGH value is placed on the SSPCLKOUT pin when data is not being transferred.

### SPH, clock phase

The SPH control bit selects the clock edge that captures data and enables it to change state. It has the most impact on the first bit transmitted by either permitting or not permitting a clock transition before the first data capture edge.

When the SPH phase control bit is LOW, data is captured on the first clock edge transition.

When the SPH clock phase control bit is HIGH, data is captured on the second clock edge transition.

Figure 59 and Figure 60 show single and continuous transmission signal sequences for Motorola SPI format with SPO=0, SPH=0.

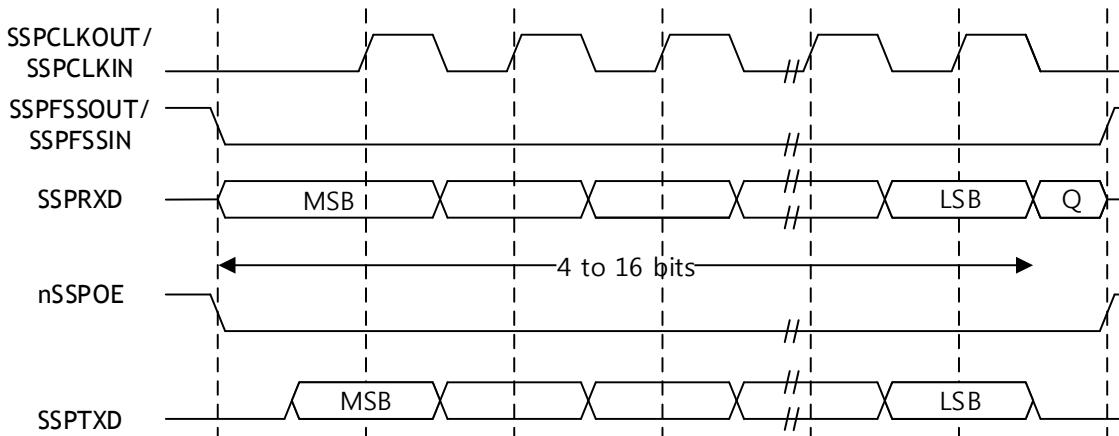


Figure 59 Motorola SPI frame format, single transfer, with SPO=0 and SPH=0

Figure 60 shows a continuous transmission signal sequence for Motorola SPI frame format with SPO=0, SPH=0.

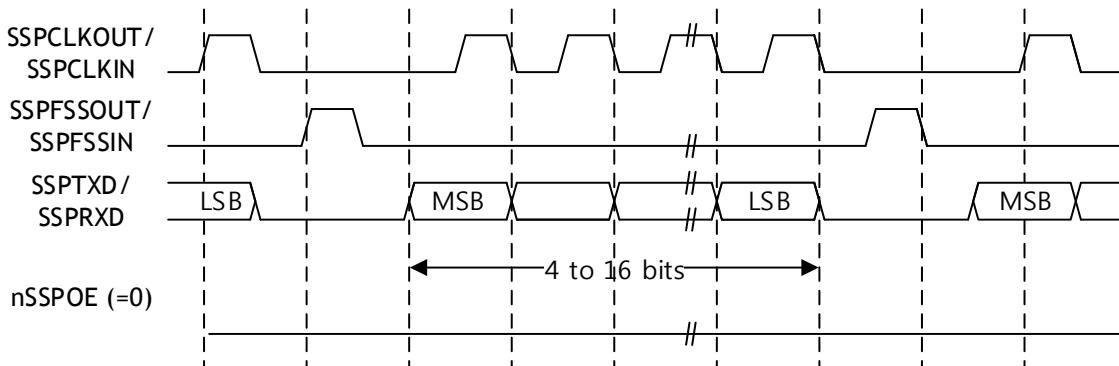


Figure 60 Motorola SPI frame format, continuous transfers, with SPO=0 and SPH=0

In this configuration, during idle periods:

- the SSPCLKOUT signal is forced LOW
- the SSPFSSOUT signal is forced HIGH
- the transmit data line SSPTXD is arbitrarily forced LOW
- the nSSPOE pad enable signal is forced HIGH, making the transmit pad high impedance
- when the PrimeCell SSP is configured as a master, the nSSPCTLOE line is driven LOW, enabling the SSPCLKOUT pad, active-LOW enable
- when the PrimeCell SSP is configured as a slave, the nSSPCTLOE line is driven HIGH, disabling the SSPCLKOUT pad, active-LOW enable.

If the PrimeCell SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW. This causes the slave data to be enabled onto the SSPRXD input line of the master. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad.

One half SSPCLKOUT period later, valid master data is transferred to the SSPTXD pin. Now that both the master and slave data have been set, the SSPCLKOUT master clock pin goes HIGH after one additional half SSPCLKOUT period.

The data is now captured on the rising and propagated on the falling edges of the SSPCLKOUT signal.

In the case of a single word transmission after all bits of the data word have been transferred, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured.

However, in the case of continuous back-to-back transmissions, the SSPFSSOUT signal must be pulsed HIGH between each data word transfer. This is because the slave select pin freezes the data in its serial peripheral register and does not permit it to be altered if the SPH bit is logic zero. Therefore, the master device must raise the SSPFSSIN pin of the slave device between each data transfer to enable the serial peripheral data write. On completion of the continuous transfer, the SSPFSSOUT pin is returned to its idle state one SSPCLKOUT period after the last bit has been captured.

Figure 61 shows the transfer signal sequence for Motorola SPI format with SPO=0, SPH=1, and it covers both single and continuous transfers.

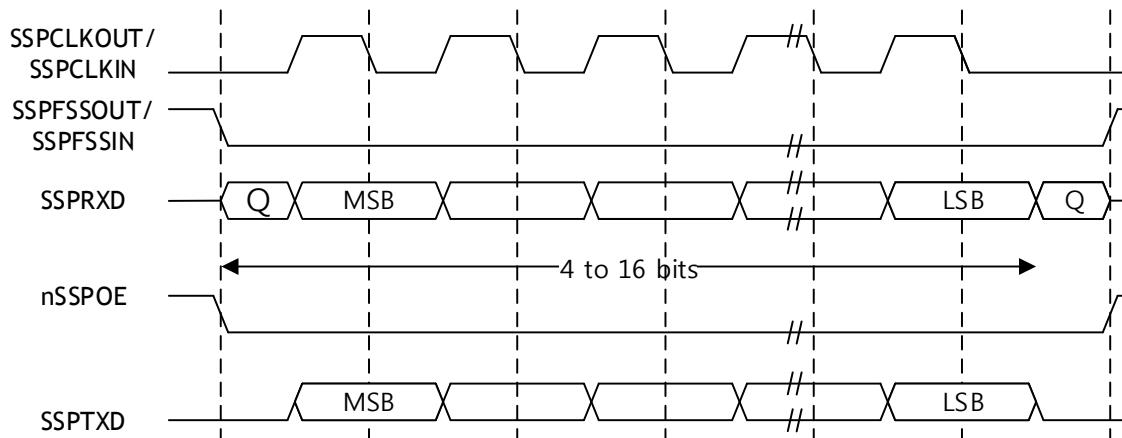


Figure 61 Motorola SPI frame format, single and continuous transfers, with SPO=0 and SPH=1

In this configuration, during idle periods:

- the SSPCLKOUT signal is forced LOW
- The SSPFSSOUT signal is forced HIGH
- the transmit data line SSPTXD is arbitrarily forced LOW

- the nSSPOE pad enable signal is forced HIGH, making the transmit pad high impedance
- when the PrimeCell SSP is configured as a master, the nSSPCTLOE line is driven LOW, enabling the SSPCLKOUT pad, active-LOW enable
- when the PrimeCell SSP is configured as a slave, the nSSPCTLOE line is driven HIGH, disabling the SSPCLKOUT pad, active-LOW enable.

If the PrimeCell SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad. After an additional one half SSPCLKOUT period, both master and slave valid data is enabled onto their respective transmission lines. At the same time, the SSPCLKOUT is enabled with a rising edge transition.

Data is then captured on the falling edges and propagated on the rising edges of the SSPCLKOUT signal.

In the case of a single word transfer after all bits have been transferred, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured.

For continuous back-to-back transfers, the SSPFSSOUT pin is held LOW between successive data words and termination is the same as that of the single word transfer.

Figure 62 shows a single transmission signal sequence for Motorola SPI format with SPO=1, SPH=0.

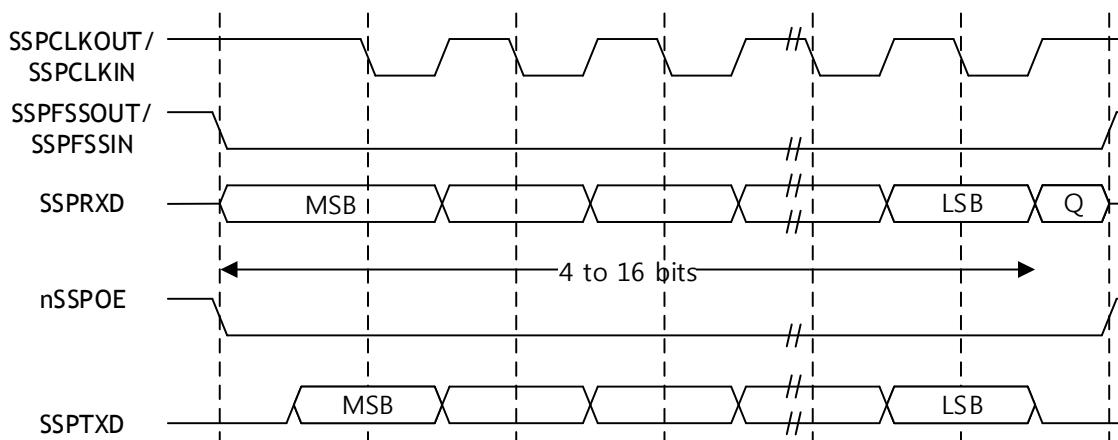


Figure 62 Motorola SPI frame format, single transfer, with SPO=1 and SPH=0

Figure 63 shows a continuous transmission signal sequence for Motorola SPI format with SPO=1, SPH=0. In Figure 9, Q is an undefined signal.

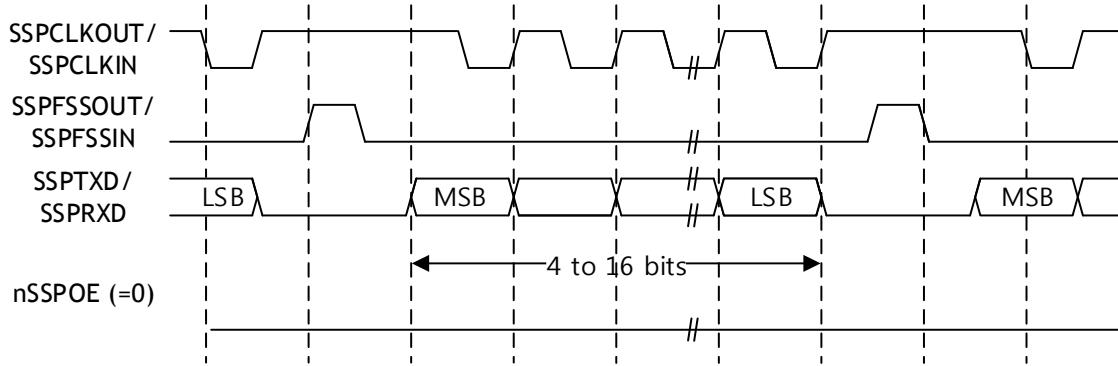


Figure 63. Motorola SPI frame format, continuous transfers, with SPO=1 and SPH=0

In this configuration, during idle periods:

- the SSPCLKOUT signal is forced HIGH
- the SSPFSSOUT signal is forced HIGH
- the transmit data line SSPTXD is arbitrarily forced LOW
- the nSSPOE pad enable signal is forced HIGH, making the transmit pad high impedance
- when the PrimeCell SSP is configured as a master, the nSSPCTLOE line is driven LOW, enabling the SSPCLKOUT pad, active-LOW enable
- when the PrimeCell SSP is configured as a slave, the nSSPCTLOE line is driven HIGH, disabling the SSPCLKOUT pad, active-LOW enable.

If the PrimeCell SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW, and this causes slave data to be immediately transferred onto the SSPRXD line of the master. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad.

One half period later, valid master data is transferred to the SSPTXD line. Now that both the master and slave data have been set, the SSPCLKOUT master clock pin becomes LOW after one additional half SSPCLKOUT period. This means that data is captured on the falling edges and be propagated on the rising edges of the SSPCLKOUT signal.

In the case of a single word transmission after all bits of the data word are transferred, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured.

However, in the case of continuous back-to-back transmissions, the SSPFSSOUT signal must be pulsed HIGH between each data word transfer. This is because the slave select pin freezes the data in its serial peripheral register and does not permit it to be altered if the SPH bit is logic zero. Therefore, the master device must raise the SSPFSSIN pin of the slave device between each data transfer to enable the serial peripheral data write. On completion of the continuous

transfer, the SSPFSSOUT pin is returned to its idle state one SSPCLKOUT period after the last bit has been captured.

Figure 64 shows the transfer signal sequence for Motorola SPI format with SPO=1, SPH=1, and it covers both single and continuous transfers. In

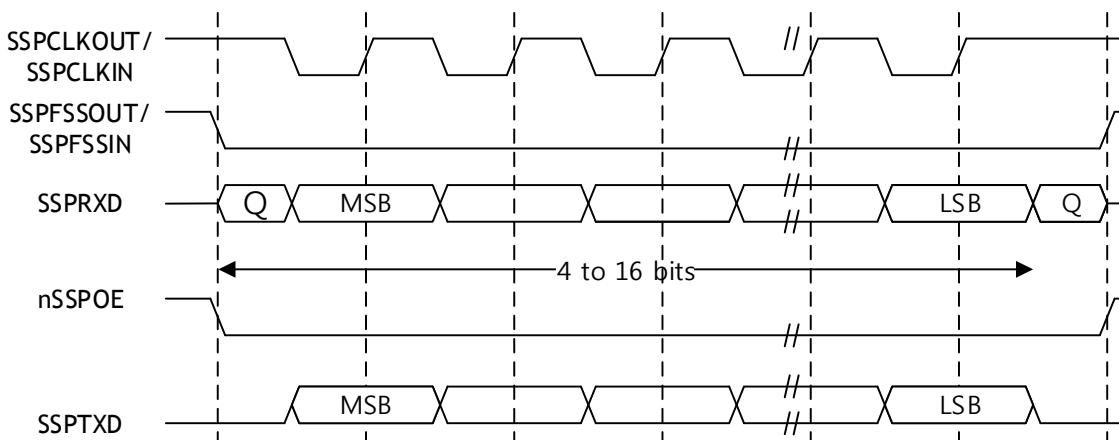


Figure 64. Motorola SPI frame format, single and continuous transfers, with SPO=1 and SPH=1, Q is an undefined signal.

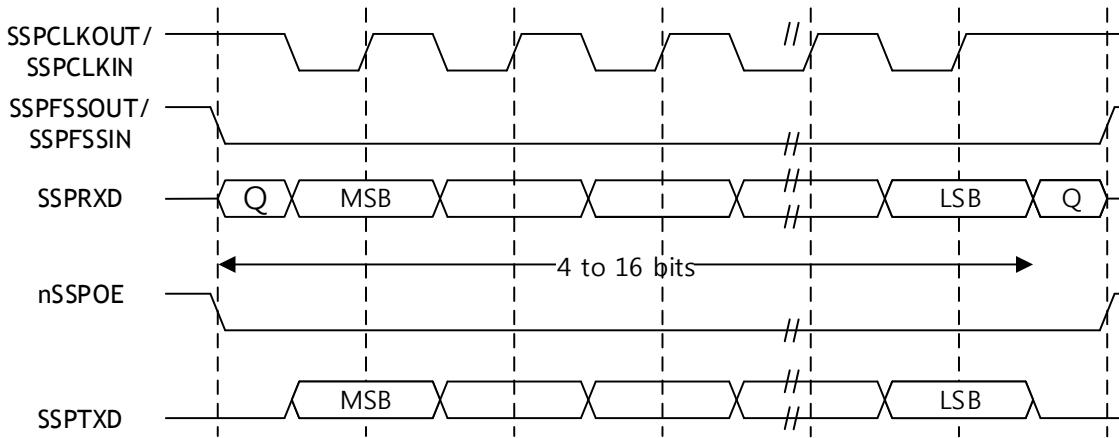


Figure 64. Motorola SPI frame format, single and continuous transfers, with SPO=1 and SPH=1

In this configuration, during idle periods:

- the SSPCLKOUT signal is forced HIGH
- the SSPFSSOUT signal is forced HIGH
- the transmit data line SSPTXD is arbitrarily forced LOW
- the nSSPOE pad enable signal is forced HIGH, making the transmit pad high impedance
- when the PrimeCell SSP is configured as a master, the nSSPCTLOE line is driven LOW, enabling the SSPCLKOUT pad, active-LOW enable
- when the PrimeCell SSP is configured as a slave, the nSSPCTLOE line is driven HIGH, disabling the SSPCLKOUT pad, active-LOW enable.

If the PrimeCell SSP is enabled and there is valid data within the transmit FIFO, the start of transmission is signified by the SSPFSSOUT master signal being driven LOW. The nSSPOE line is driven LOW, enabling the master SSPTXD output pad. After an additional one half SSPCLKOUT period, both master and slave data are enabled onto their respective transmission lines. At the same time, the SSPCLKOUT is enabled with a falling edge transition. Data is then captured on the rising edges and propagated on the falling edges of the SSPCLKOUT signal.

After all bits have been transferred in the case of a single word transmission, the SSPFSSOUT line is returned to its idle HIGH state one SSPCLKOUT period after the last bit has been captured.

For continuous back-to-back transmissions, the SSPFSSOUT pin remains in its active-LOW state, until the final bit of the last word has been captured, and then returns to its idle state as the previous section describes.

For continuous back-to-back transfers, the SSPFSSOUT pin is held LOW between successive data words and termination is the same as that of the single word transfer.

## National Semiconductor Microwire frame format

Figure 65 shows the National Semiconductor Microwire frame format for a single frame.

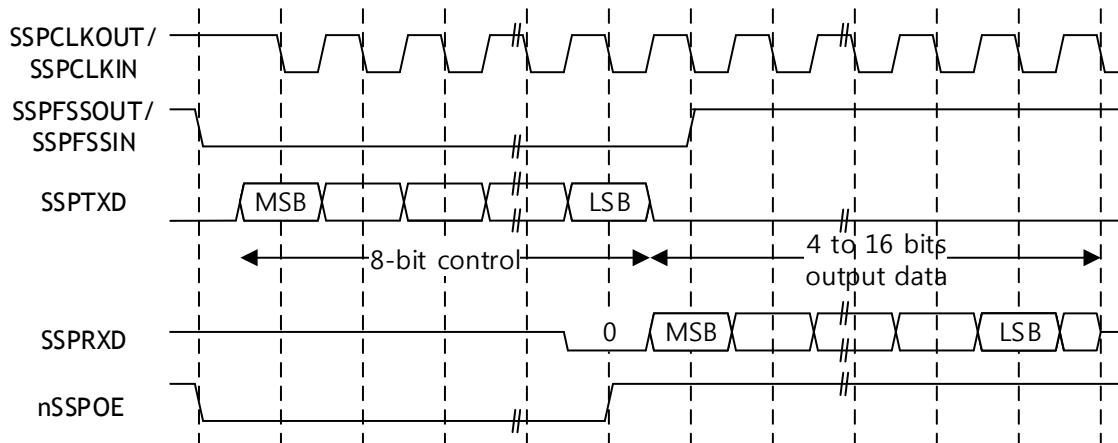


Figure 65. National Semiconductor Microwire frame format, single transfer

Microwire format is very similar to the SPI format except that transmission is half-duplex instead of full-duplex using a master-slave message passing technique. Each serial transmission begins with an 8-bit control word that is transmitted from the PrimeCell SSP to the off-chip slave device. During this transmission, the PrimeCell SSP receives no incoming data. After the message has been sent, the off-chip slave decodes it and responds with the required data after waiting one serial clock after the last bit of the 8-bit control message has been sent. The

---

returned data is 4 to 16 bits in length, making the total frame length in the range of 13-25 bits.

In this configuration, during idle periods:

- SSPCLKOUT is forced LOW
- SSPFSSOUT is forced HIGH
- the transmit data line, SSPTXD, is arbitrarily forced LOW
- the nSSPOE pad enable signal is forced HIGH, making the transmit pad high impedance.

A transmission is triggered by writing a control byte to the transmit FIFO. The falling edge of SSPFSSOUT causes the value contained in the bottom entry of the transmit FIFO to be transferred to the serial shift register of the transmit logic and the MSB of the 8-bit control frame to be shifted out onto the SSPTXD pin. SSPFSSOUT remains LOW for the duration of the frame transmission. The SSPRXD pin remains tristated during this transmission.

The off-chip serial slave device latches each control bit into its serial shifter on the rising edge of each SSPCLKOUT. After the last bit is latched by the slave device, the control byte is decoded during a one clock wait-state and the slave responds by transmitting data back to the PrimeCell SSP. Each bit is driven onto the SSPRXD line on the falling edge of SSPCLKOUT. The PrimeCell SSP in turn latches each bit on the rising edge of SSPCLKOUT. At the end of the frame for single transfers, the SSPFSSOUT signal is pulled HIGH one clock period after the last bit has been latched in the receive serial shifter, which causes the data to be transferred to the receive FIFO.

The off-chip slave device can tristate the receive line either on the falling edge of SSPCLKOUT after the LSB has been latched by the receive shifter or when the SSPFSSOUT pin goes HIGH.

For continuous transfers, data transmission begins and ends in the same manner as a single transfer. However, the SSPFSSOUT line is continuously asserted, held LOW, and transmission of data occurs back-to-back. The control byte of the next frame follows directly after the LSB of the received data from the current frame. Each of the received values is transferred from the receive shifter on the falling edge SSPCLKOUT, after the LSB of the frame has been latched into the PrimeCell SSP.

Figure 66 shows the National Semiconductor Microwire frame format when back-to-back frames are transmitted.

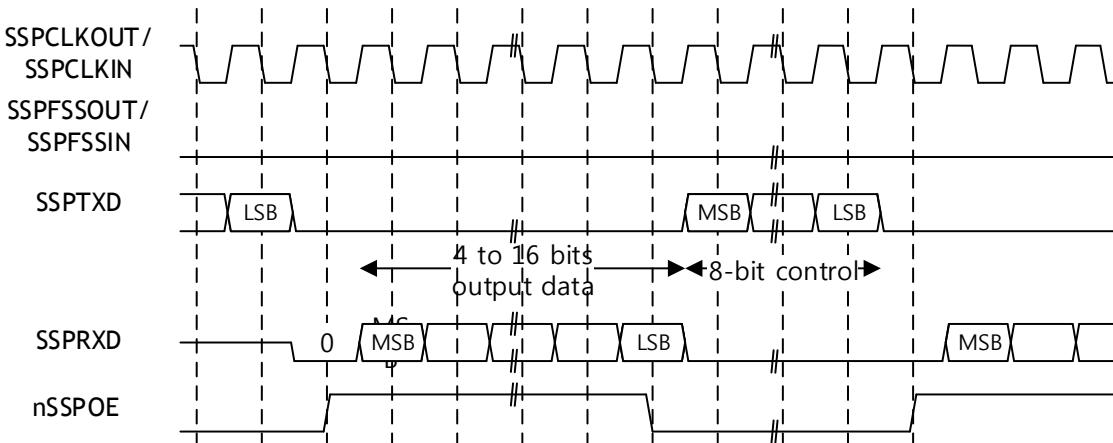


Figure 66. National Semiconductor Microwire frame format, continuous transfers

## Master and Slave configurations

Figure 67 shows how a PrimeCell SSP (PL022) configured as master, interfaces to a Motorola SPI slave. The SPI Slave Select (SS) signal is permanently tied LOW and configures it as a slave. Similar to the above operation, the master can broadcast to the slave through the master PrimeCell SSP SSPTXD line. In response, the slave drives its SPI MISO port onto the SSPRXD line of the master.

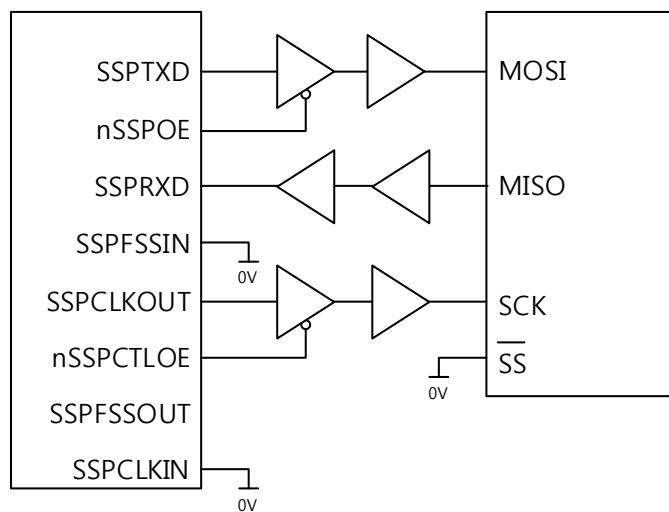


Figure 67. PrimeCell SSP master coupled to an SPI slave

Figure 68 shows a Motorola SPI configured as a master and interfaced to an instance of a PrimeCell SSP (PL022) configured as a slave. In this case, the slave Select Signal (SS) is permanently tied HIGH to configure it as a master. The master can broadcast to the slave through the master SPI MOSI line. In response, the slave drives its nSSPOE signal LOW. This enables its SSPTXD data onto the MISO line of the master.

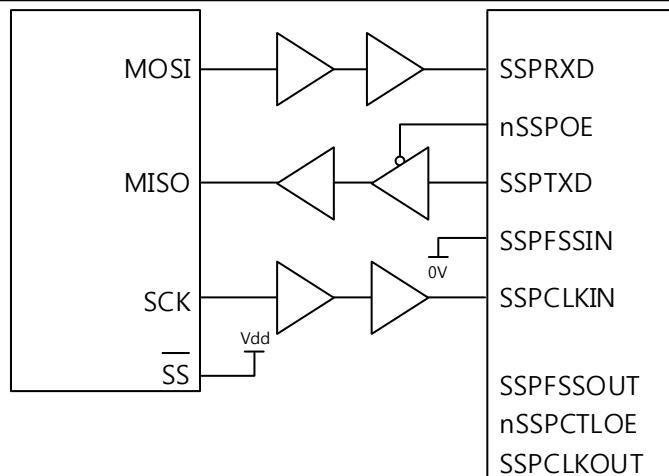


Figure 68. SPI master coupled to a PrimeCell SSP slave

## SSP Flow chart

Figure 69 shows how to setting TI or Microwire mode.

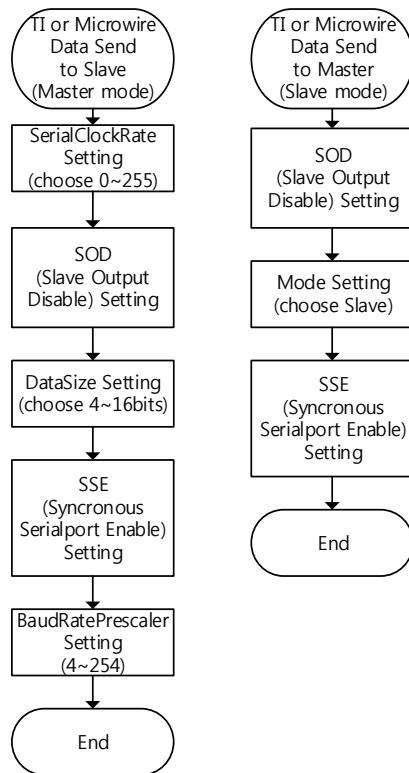


Figure 69. how to setting TI or Microwire mode flow chart

Figure 70 shows how to setting SPI mode.

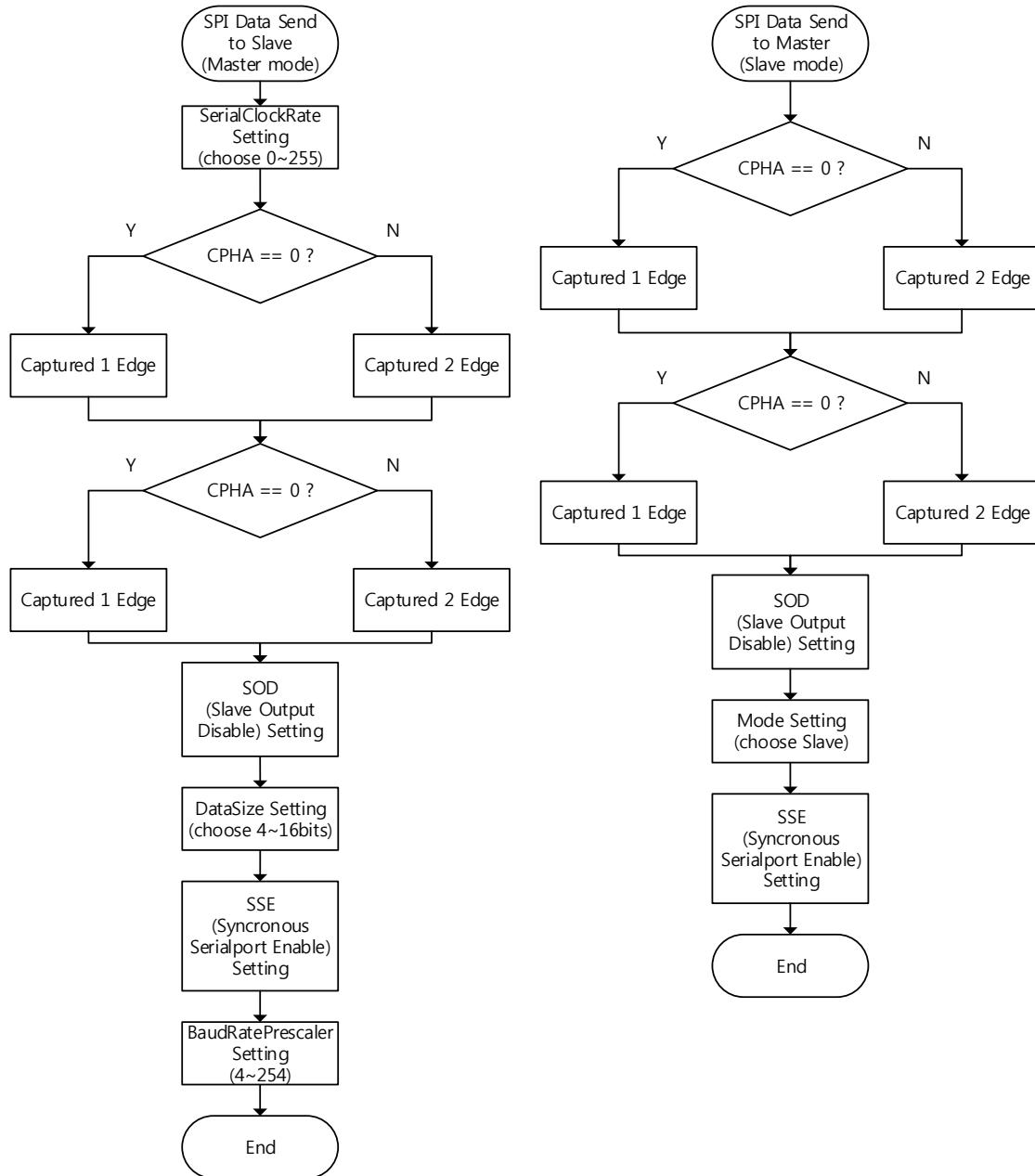


Figure 70. how to setting SPI mode flow chart

## 27.4 SSP0 Registers (Base Address : 0x4000\_A000)

This section describes the SSP0 registers.

### SSP0 Control register 0 (SSP0CR0)

Address offset: 0x0000

Reset value: 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

res																

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR								SPH	SPO	FRF		DSS			
R/W								R/W	R/W	R/W		R/W			

[3:0] DSS - Data size select:

- 0000 : reserved, undefined operation
- 0001 : reserved, undefined operation
- 0010 : reserved, undefined operation
- 0011 : 4-bit data
- 0100 : 5-bit data
- 0101 : 6-bit data
- 0110 : 7-bit data
- 0111 : 8-bit data
- 1000 : 9-bit data
- 1001 : 10-bit data
- 1010 : 11-bit data
- 1011 : 12-bit data
- 1100 : 13-bit data
- 1101 : 14-bit data
- 1111 : 15-bit data

[5:4] FRF - Frame Format

- 00 : Motorola SPI frame format
- 01 : TI synchronous serial frame format
- 10 : National Microwire frame format
- 11 : Reserved, undefined operation

[6] SPO - SSPCLKOUT polarity

This is applicable to Motorola SPI frame format only.

[7] SPH - SSPCLKOUT phase

This is applicable to Motorola SPI frame format only.

[15:8] SCR - Serial clock rate

The value SCR is used to generate the transmit and receive bit rate of the SSP. The bit rate is:

$$f_{SSPCLK} / (CPSDVSR * (1 + SCR))$$

where

CPSDVSR is an even value from 2-254, programmed through the SSPCPSR register and SCR is a value from 0-255.

## SSP0 Control register 1 (SSP0CR1)

Address offset: 0x0004

Reset value: 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	SOD	MS	SSE	LBM											
												R/W	R/W	R/W	R/W

[0] LBM - Loop back mode:

0 : normal serial port operation enabled

1 : output of transmit serial shifter is connected to input of receive serial shifter internally

[1] SSE - Synchronous serial port enable:

0 : SSP0 operation disabled.

1 : SSP0 operation enabled.

[2] MS - Master or Slave mode select:

0 : device configured as master, default.

1 : device configured as slave.

[3] SOD - Slave-mode output disable.

This bit is relevant only in the slave mode, MS = 1. In multiple-slave systems, it is possible for a SSP0 master to broadcast a message to all slaves in the system while ensuring that only one slave drives data onto its serial output line. In such systems the RXD lines from multiple slaves could be tied together.

To operate in such systems, the SOD bit can be set if the SSP0 slave is not supposed to drive the SSPTXD line:

0 : SSP0 can drive the SSPTXD output in slave mode.

1 : SSP0 must not drive the SSPTXD output in slave mode.

## SSP0 Data register (SSP0DR)

Address offset: 0x0008

Reset value: 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data															
R/W															

[15:0] DATA - Transmit/Receive FIFO:

Read: Read: receive FIFO.

Write: transmit FIFO.

You must right-justify data when the SSP0 is programmed for a data size that is less than 16 bits. Unused bits at the top are ignored by transmit logic. The receive logic automatically right-justifies.

## SSP0 Status register (SSP0SR)

Address offset: 0x000C

Reset value: 0x0000\_0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	BSY	RFF	RNE	TNF	TFE										
											R/W	R/W	R/W	R/W	R/W

[0] TFE - Transmit FIFO empty, RO

0 : Transmit FIFO is not empty.

1 : Transmit FIFO is empty.

[1] TNF - Transmit FIFO not full, RO:

0 : Transmit FIFO is full.

1 : Transmit FIFO is not full.

[2] RNE - Receive FIFO not empty, RO:

0 : Receive FIFO is empty.

1 : Receive FIFO is not empty.

[3] RFF - Receive FIFO full, RO:

0 : Receive FIFO is not full.

1 : Receive FIFO is full.

[4] BSY - SSP busy flag, RO:

0 : SSP is idle.No effect

1 : SSP is currently transmitting and/or receiving a frame or the transmit FIFO is not empty.

## SSP0 Clock prescale register (SSP0CPSR)

Address offset: 0x0010

Reset value: 0x0000\_00000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res								CPSDVS							
								R/W							

[7:0] CPSDVS - Clock prescale divisor

This must be an even number from 2-254, depending on the frequency of SSPCLK. The least significant bit always returns zero on reads.

## SSP0 Interrupt mask set or clear register (SSP0IMSC)

Address offset: 0x0014

Reset value: 0x0000\_00000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TXI	RXI	RTI	ROR											
												M	M	M	IM
												R/W	R/W	R/W	R/W

[0] RORIM - Receive overrun interrupt mask:

0 : Receive FIFO written to while full condition interrupt is masked.

1 : Receive FIFO written to while full condition interrupt is not masked.

[1] RTIM - Receive timeout interrupt mask:

0 : Receive FIFO not empty and no read prior to timeout period interrupt is masked.

1 : Receive FIFO not empty and no read prior to timeout period interrupt is not masked.

[2] RXIM - Receive FIFO interrupt mask:

0 : Receive FIFO half full or less condition interrupt is masked.

1 : Receive FIFO half full or less condition interrupt is not masked.

[3] TXIM - Transmit FIFO interrupt mask:

0 : Transmit FIFO half empty or less condition interrupt is masked.

1 : Transmit FIFO half empty or less condition interrupt is not masked.

## SSP0 Raw interrupt status register (SSPORIS)

Address offset: 0x0018

Reset value: 0x0000\_00004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TXRI S	RXRI S	RTRI S	ROR RIS											
												R/W	R/W	R/W	R/W

[0] RORRIS - Gives the raw interrupt state, prior to masking, of the SSPRORINTR interrupt

[1] RTRIS - Gives the raw interrupt state, prior to masking, of the SSPRTINTR interrupt

[2] RXRIS - Gives the raw interrupt state, prior to masking, of the SSPRXINTR interrupt

[3] TXRIS - Gives the raw interrupt state, prior to masking, of the SSPTXINTR interrupt

## SSP0 Masked interrupt status register, (SSP0MIS)

Address offset: 0x001C

Reset value: 0x0000\_00000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

res	TXM IS	RXM IS	RTM IS	ROR MIS											
												R/W	R/W	R/W	R/W

[0] RORMIS - Gives the receive over run masked interrupt status, after masking, of the SSPRORINTR interrupt

[1] RTMIS - Gives the receive timeout masked interrupt state, after masking, of the SSPRTINTR interrupt

[2] RXMIS - Gives the receive FIFO masked interrupt state, after masking, of the SSPRXINTR interrupt

[3] TXMIS - Gives the transmit FIFO masked interrupt state, after masking, of the SSPTXINTR interrupt

### SSP0 Interrupt clear register (SSP0ICR)

Address offset: 0x0020

Reset value: 0x0000\_00000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	RTIC	ROR IC													

[0] RORICS - Clears the SSPRORINTR interrupt

[1] RTIC - Clears the SSPRTINTR interrupt

### SSP0 DMA control register, (SSP0DMACR)

Address offset: 0x0024

Reset value: 0x0000\_00000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TXD MAE	RXD MAE													
														R/W	R/W

[0] RXDMAE - Receive DMA Enable:

0 : DMA for the receive FIFO is disabled.

1 : DMA for the receive FIFO is enabled.

[1] TXDMAE - Transmit DMA Enable:

0 : DMA for the transmit FIFO is disabled.

1 : DMA for the transmit FIFO is enabled.

## 27.5 Register map

The following Table 38 summarizes the SSP0 registers.

Table 38 SSP0 register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
0x00	SSPCR0	res																																						
	reset value																																							
0x04	SSPCR1	res																																						
	reset value																																							
0x08	SSPDR	res																																						
	reset value																																							
0x0C	SSPSR	res																																						
	reset value																																							
0x10	SSPCPSR	res																																						
	reset value																																							
0x14	SSPIMSC	res																																						
	reset value																																							
0x18	SSPRIS	res																																						
	reset value																																							
0x1C	SSPMIS	res																																						
	reset value																																							
0x20	SSPICR	res																																						
	reset value																																							
0x24	SSPDMACR	res																																						
	reset value																																							

## 27.6 SSP1 Registers (Base Address : 0x4000\_B000)

This section describes the SSP0 registers.

### SSP1 Control register 0 (SSP1CR0)

Address offset: 0x0000

Reset value: 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCR								SPH	SPO	FRF		DSS			
R/W								R/W	R/W	R/W		R/W			

[3:0] DSS - Data size select:

0000 : reserved, undefined operation

0001 : reserved, undefined operation

0010 : reserved, undefined operation

0011 : 4-bit data

0100 : 5-bit data

0101 : 6-bit data

0110 : 7-bit data

0111 : 8-bit data

1000 : 9-bit data

1001 : 10-bit data

1010 : 11-bit data

1011 : 12-bit data

1100 : 13-bit data

1101 : 14-bit data

1111 : 15-bit data

[5:4] FRF - Frame Format

00 : Motorola SPI frame format

01 : TI synchronous serial frame format

10 : National Microwire frame format

11 : Reserved, undefined operation

[6] SPO - SSPCLKOUT polarity

This is applicable to Motorola SPI frame format only.

[7] SPH - SSPCLKOUT phase

This is applicable to Motorola SPI frame format only.

[15:8] SCR - Serial clock rate

The value SCR is used to generate the transmit and receive bit rate of the SSP. The bit rate is:

$$f_{SSPCLK} / (CPSDVSR * (1 + SCR))$$

where

CPSDVSR is an even value from 2-254, programmed through the SSPCPSR register and SCR is a value from 0-255.

## SSP1 Control register 1 (SSP1CR1)

Address offset: 0x0004

Reset value: 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	SOD	MS	SSE	LBM											
												R/W	R/W	R/W	R/W

[0] LBM - Loop back mode:

0 : normal serial port operation enabled

1 : output of transmit serial shifter is connected to input of receive serial shifter internally

[1] SSE - Synchronous serial port enable:

0 : SSP1 operation disabled.

1 : SSP1 operation enabled.

[2] MS - Master or Slave mode select:

0 : device configured as master, default.

1 : device configured as slave.

[3] SOD - Slave-mode output disable.

This bit is relevant only in the slave mode, MS = 1. In multiple-slave systems, it is possible for a SSP1 master to broadcast a message to all slaves in the system while ensuring that only one slave drives data onto its serial output line. In such systems the RXD lines from multiple slaves could be tied together.

To operate in such systems, the SOD bit can be set if the SSP1 slave is not supposed to drive the SSPTXD line:

0 : SSP1 can drive the SSPTXD output in slave mode.

---

1 : SSP1 must not drive the SSPTXD output in slave mode.

## SSP1 Data register (SSP1DR)

Address offset: 0x0008

Reset value: 0x0000\_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	res	res	res	res	res	res	res	res	res	res	res	res	res	res	res
Data															
R/W															

[15:0] DATA - Transmit/Receive FIFO:

Read: Read: receive FIFO.

Write: transmit FIFO.

You must right-justify data when the SSP1 is programmed for a data size that is less than 16 bits. Unused bits at the top are ignored by transmit logic. The receive logic automatically right-justifies.

## SSP1 Status register (SSP1SR)

Address offset: 0x000C

Reset value: 0x0000\_0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															
res															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	BSY	RFF	RNE	TNF	TFE										
											R/W	R/W	R/W	R/W	R/W

[0] TFE - Transmit FIFO empty, RO

0 : Transmit FIFO is not empty.

1 : Transmit FIFO is empty.

[1] TNF - Transmit FIFO not full, RO:

0 : Transmit FIFO is full.

1 : Transmit FIFO is not full.

[2] RNE - Receive FIFO not empty, RO:

0 : Receive FIFO is empty.

1 : Receive FIFO is not empty.

[3] RFF - Receive FIFO full, RO:

0 : Receive FIFO is not full.

1 : Receive FIFO is full.

[4] BSY - SSP busy flag, RO:

0 : SSP is idle. No effect

1 : SSP is currently transmitting and/or receiving a frame or the transmit FIFO is not empty.

## SSP1 Clock prescale register (SSP1CPSR)

Address offset: 0x0010

Reset value: 0x0000\_00000

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

[7:0] CPSDVSR - Clock prescale divisor

This must be an even number from 2-254, depending on the frequency of SSPCLK. The least significant bit always returns zero on reads.

## SSP1 Interrupt mask set or clear register (SSP1IMSC)

Address offset: 0x0014

Reset value: 0x0000\_00000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res	res	res	res												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TXI M	RXI M	RTI M	ROR IM											
												R/W	R/W	R/W	R/W

[0] RORIM - Receive overrun interrupt mask:

0 : Receive FIFO written to while full condition interrupt is masked.

1 : Receive FIFO written to while full condition interrupt is not masked.

[1] RTIM - Receive timeout interrupt mask:

0 : Receive FIFO not empty and no read prior to timeout period interrupt is masked.

1 : Receive FIFO not empty and no read prior to timeout period interrupt is not masked.

[2] RXIM - Receive FIFO interrupt mask:

0 : Receive FIFO half full or less condition interrupt is masked.

1 : Receive FIFO half full or less condition interrupt is not masked.

[3] TXIM - Transmit FIFO interrupt mask:

0 : Transmit FIFO half empty or less condition interrupt is masked.

1 : Transmit FIFO half empty or less condition interrupt is not masked.

## SSP1 Raw interrupt status register (SSP1RIS)

Address offset: 0x0018

Reset value: 0x0000\_00004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TXRI S	RXRI S	RTRI S	ROR RIS											
												R/W	R/W	R/W	R/W

[0] RORRIS - Gives the raw interrupt state, prior to masking, of the SSPRORINTR interrupt

[1] RTRIS - Gives the raw interrupt state, prior to masking, of the SSPRTINTR interrupt

[2] RXRIS - Gives the raw interrupt state, prior to masking, of the SSPRXINTR interrupt

[3] TXRIS - Gives the raw interrupt state, prior to masking, of the SSPTXINTR interrupt

## SSP1 Masked interrupt status register, (SSP1MIS)

Address offset: 0x001C

Reset value: 0x0000\_00000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	TXM IS	RXM IS	RTM IS	ROR MIS											
												R/W	R/W	R/W	R/W

[0] RORMIS - Gives the receive over run masked interrupt status, after masking, of the SSPRORINTR interrupt

[1] RTMIS - Gives the receive timeout masked interrupt state, after masking, of the SSPRTINTR interrupt

[2] RXMIS - Gives the receive FIFO masked interrupt state, after masking, of the SSPRXINTR interrupt

[3] TXMIS - Gives the transmit FIFO masked interrupt state, after masking, of the SSPTXINTR interrupt

### SSP1 Interrupt clear register (SSP1ICR)

Address offset: 0x0020

Reset value: 0x0000\_00000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
res	RTIC	ROR IC													
														R/W	R/W

[0] RORICS - Clears the SSPRORINTR interrupt

[1] RTIC - Clears the SSPRTINTR interrupt

### SSP1 DMA control register, (SSP1DMACR)

Address offset: 0x0024

Reset value: 0x0000\_00000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
res															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

res	TXD MAE	RXD MAE													
														R/W	R/W

[0] RXDMAE - Receive DMA Enable:

0 : DMA for the receive FIFO is disabled.

1 : DMA for the receive FIFO is enabled.

[1] TXDMAE - Transmit DMA Enable:

0 : DMA for the transmit FIFO is disabled.

1 : DMA for the transmit FIFO is enabled.

## 27.7 Register map

The following Table 39 summarizes the SSP1 registers.

Table 39 SSP1 register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0x00	SSPCR0	res																																			
	reset value																																				
0x04	SSPCR1	res																																			
	reset value																																				
0x08	SSPDR	res																																			
	reset value																																				
0x0C	SSPSR	res																																			
	reset value																																				
0x10	SSPCPSR	res																																			
	reset value																																				
0x14	SSPIMSC	res																																			
	reset value																																				
0x18	SSPRIS	res																																			
	reset value																																				
0x1C	SSPMIS	res																																			
	reset value																																				
0x20	SSPICR	res																																			
	reset value																																				
0x24	SSPDMACR	res																																			
	reset value																																				

## Document History Information

Version	Date	Descriptions
Ver. 1.0.0	18SEP2017	Initial Release
Ver. 1.0.1	12OCT2017	Edit Flash Chapter.
Ver. 1.0.2	26OCT2017	Edit GPIO,PADCON Register and added Open Drain function
Ver. 1.0.3	27OCT2017	TCKCNTR, RTR, Sn_RTR, Sn_KATMR Additional description added.
Ver. 1.0.4	09NOV2017	Edit PADCON register description CS,DS inverted value.
Ver. 1.0.5	17NOV2017	Edit UARTCR register description about UARTEN.
Ver. 1.0.6	29NOV2017	Edit WZTOE Sn_KATMR description about timer trigger.
Ver. 1.0.7	26JAN2018	Edit typo about PWM preodic mode
Ver. 1.0.8	12FAB2018	Edit Base Address about RTC (4000_4000 -> 4000_E000)
Ver. 1.0.9	05MAR2018	Edit UART2 description & register map.
Ver. 1.1.0	11APR2018	Edit Systick description & PADCON register table fix
		Edit Register Table about SSP0, SSP1 (Add FRF)
		Edit DSS about SSP0, SSP1
		Edit Register map about SSP0, SSP1 (Add FRF)
Ver. 1.1.1	27JUL2018	Fixed figure 4. CRG Diagram (Input Information of WDOGCLK, TIMCLK0, TIMCLK1).
		Flash memory lock function Additional description added (11.1)

## Copyright Notice

Copyright 2018 WIZnet Co., Ltd. All Rights Reserved.

Technical Support: <https://forum.wizwiki.io/>

Sales & Distribution: [sales@wiznet.io](mailto:sales@wiznet.io)

For more information, visit our website at <http://www.wiznet.io/>