

SN32F700 Series

USER'S MANUAL

SN32F707
SN32F706
SN32F705
SN32F7051
SN32F704
SN32F703
SN32F7031

SONiX 32-Bit Cortex-M0 Micro-Controller

SONiX reserves the right to make change without further notice to any products herein to improve reliability, function or design. SONiX does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. SONiX products are not designed, intended, or authorized for use as components in systems intended, for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SONiX product could create a situation where personal injury or death may occur. Should Buyer purchase or use SONiX products for any such unintended or unauthorized application. Buyer shall indemnify and hold SONiX and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that SONiX was negligent regarding the design or manufacture of the part.

AMENDMENT HISTORY

| Version | Date | Description |
|---------|------------|---|
| 1.0 | 2012/04/13 | First version released. |
| 1.1 | 2012/04/30 | 1. Fix typing errors. |
| 1.2 | 2012/06/27 | 1. Add ADC PCLK constraints. 2. Update typical value of I _{SLEEP} . 3. Fix typing errors. |
| 1.3 | 2012/08/31 | 1. Update Flash chapter. 2. Update Timer operation section of CT16Bn and CT32Bn. 3. Modify description of NVIC_IUSER and NVIC_ICER. |
| 1.4 | 2012/11/06 | 1. Remove prescaler related descriptions of all timers. |
| 1.5 | 2013/02/27 | 1. Add notifications of GPIO settings in low-power mode for low-pin count package. 2. Update Chap 17. Development Tool. 3. Modify SysTick register names refer to core_cm0.h provided by ARM. 4. Remove Flash checksum. 5. Modify IOL in Electrical Characteristic chapter. |
| 1.6 | 2013/05/06 | 1. Update SysTick Timer block diagram. 2. Add SYS0_ANTIEFT register. 3. Update supply current. 4. Add Operation Mode Comparison Table. |
| 2.0 | 2013/08/16 | 1. Fix typing errors. 2. Update V _{IH} and V _{IL} in Electrical Characteristics chapter. 3. Update Code Security table and flow diagram in Code Security section. 4. Add Note for I/O open-drain function. 5. Add descriptions of I2C events which trigger I2C interrupt for I2Cn_STAT register. |
| 2.1 | 2014/02/14 | 1. Add SN32F705/715/725 2. Fix typing errors in 11.4.1 and 11.4.2 sections. 3. Update Flash Organization table. 4. Update SN-LINK description in Chap 17. Development Tool . 5. Add Notice for BOOT pin in Chap 16. SWD . 6. Add WAKEUP sections. 7. Fix typing errors. 8. Update Code Security table and flow diagram in Code Security section. |
| 2.2 | 2014/06/04 | 1. Update 21.3 Marking Example section. 2. Update SN-LINK-V2 photos. 3. Fix typing errors. |
| 2.3 | 2015/05/29 | 1. Update QFN46 package information. 2. Update QFN33 4x4 package information. 3. Update UART baud rate sample. |
| 2.4 | 2015/08/21 | 1. Add Note for P2.n GPIO setting if ADC function is used. 2. Update PLL recommend input/output frequency setting table . |
| 2.5 | 2016/07/22 | 1. Modify PWM/EM related registers refer to the PWM/EM sets. - Modify CT16B0_EM, CT16B1_EM. - Modify CT16B0_PWMCTRL, CT16B1_PWMCTRL. 2. Add Notice: HCLK MUST be equal or less than 24MHz during Flash program and erase operations. 3. Add SSPn Data Fetch (SSPn_DF) register. 4. Update LQFP48 package information. |
| 2.6 | 2016/12/02 | 1. Update Features Selection Table. 2. Fix SPI transfer timing figure. 3. Remove SN32F71X & SN32F72X. 4. Add SN32F7051/SN32F704/SN32F703/SN32F7031 5. Remove SYSTICKPRE[1:0] |
| 2.7 | 2017/06/05 | 1. Remove SYS0_ANTIEFT register. 2. Fix Deep Sleep mode wake up time. 3. Update SN-LINK-V3.0 information. |
| 2.8 | 2017/12/15 | 1. Update WDTPRE[2:0] bits description in SYS1_APBPCP1 register. 2. Update TO[15:0] bits description in I2Cn_TOCTRL register. |

| | | |
|-----|------------|--|
| | | 3. Add Note for setting the pins which are not pin-out. |
| 2.9 | 2018/10/05 | 1. Add Checksum related registers. 2. Add Boot pin description. 3. Fix maximum F_{ADCLK} to 16MHz in Electrical Characteristics. |
| 3.0 | 2019/03/07 | 1. Add Unique number descriptions. |
| 3.1 | 2019/09/20 | 1. Update 20 PACKAGE INFORMATION with new format. |
| 3.2 | 2019/12/25 | 1. Update 6.2 ADC CONVERTING TIME 2. Complement Chap 12 I2C. 3. Add notifications of BOOT pin in 1.4 PIN ASSIGNMENT. 4. Add introduction of 2.6 UNIQUE NUMBER . |

Table of Content

| | |
|---|-----------|
| AMENDMENT HISTORY..... | 2 |
| 1 PRODUCT OVERVIEW..... | 14 |
| 1.1 FEATURES..... | 14 |
| 1.2 SYSTEM BLOCK DIAGRAM..... | 16 |
| 1.3 CLOCK GENERATION BLOCK DIAGRAM..... | 17 |
| 1.4 PIN ASSIGNMENT..... | 18 |
| 1.5 PIN DESCRIPTIONS..... | 24 |
| 1.6 PIN CIRCUIT DIAGRAMS..... | 27 |
| 2 CENTRAL PROCESSOR UNIT (CPU)..... | 29 |
| 2.1 MEMORY MAP..... | 29 |
| 2.2 SYSTEM TICK TIMER..... | 30 |
| 2.2.1 OPERATION..... | 30 |
| 2.2.2 SYSTICK USAGE HINTS AND TIPS..... | 31 |
| 2.2.3 SYSTICK REGISTERS..... | 31 |
| 2.2.3.1 System Tick Timer Control and Status register (SYSTICK_CTRL)..... | 31 |
| 2.2.3.2 System Tick Timer Reload value register (SYSTICK_LOAD)..... | 31 |
| 2.2.3.3 System Tick Timer Current Value register (SYSTICK_VAL)..... | 32 |
| 2.2.3.4 System Tick Timer Calibration Value register (SYSTICK_CALIB)..... | 32 |
| 2.3 NESTED VECTORED INTERRUPT CONTROLLER (NVIC)..... | 33 |
| 2.3.1 INTERRUPT AND EXCEPTION VECTORS..... | 33 |
| 2.3.2 NVIC REGISTERS..... | 34 |
| 2.3.2.1 IRQ0~31 Interrupt Set-Enable Register (NVIC_ISER)..... | 34 |
| 2.3.2.2 IRQ0~31 Interrupt Clear-Enable Register (NVIC_ICER)..... | 34 |
| 2.3.2.3 IRQ0~31 Interrupt Set-Pending Register (NVIC_ISPR)..... | 34 |
| 2.3.2.4 IRQ0~31 Interrupt Clear-Pending Register (NVIC_ICPR)..... | 34 |
| 2.3.2.5 IRQ0~31 Interrupt Priority Register (NVIC_IPRn) (n=0~7)..... | 35 |
| 2.4 APPLICATION INTERRUPT AND RESET CONTROL (AIRC)..... | 35 |
| 2.5 CODE OPTION TABLE..... | 36 |
| 2.6 UNIQUE NUMBER..... | 36 |
| 2.7 CORE REGISTER OVERVIEW..... | 37 |
| 3 SYSTEM CONTROL..... | 38 |
| 3.1 RESET..... | 38 |
| 3.1.1 POWER-ON RESET (POR)..... | 38 |
| 3.1.2 WATCHDOG RESET (WDT RESET)..... | 39 |

| | | |
|---------|---|----|
| 3.1.3 | <i>BROWN-OUT RESET</i> | 39 |
| 3.1.3.1 | BROWN OUT DESCRIPTION | 39 |
| 3.1.3.2 | THE SYSTEM OPERATING VOLTAGE DECSRIPTION | 40 |
| 3.1.3.3 | BROWN-OUT RESET IMPROVEMENT | 40 |
| 3.1.4 | <i>EXTERNAL RESET</i> | 41 |
| 3.1.4.1 | SIMPLY RC RESET CIRCUIT | 42 |
| 3.1.4.2 | DIODE & RC RESET CIRCUIT | 42 |
| 3.1.4.3 | ZENER DIODE RESET CIRCUIT | 43 |
| 3.1.4.4 | VOLTAGE BIAS RESET CIRCUIT | 43 |
| 3.1.4.5 | EXTERNAL RESET IC | 44 |
| 3.1.5 | <i>SOFTWARE RESET</i> | 44 |
| 3.2 | SYSTEM CLOCK | 45 |
| 3.2.1 | <i>INTERNAL RC CLOCK SOURCE</i> | 45 |
| 3.2.1.1 | Internal High-speed RC Oscillator (IHRC) | 45 |
| 3.2.1.2 | Internal Low-speed RC Oscillator (ILRC) | 45 |
| 3.2.2 | <i>PLL</i> | 46 |
| 3.2.2.1 | PLL Frequency selection | 46 |
| 3.2.3 | <i>EXTERNAL CLOCK SOURCE</i> | 47 |
| 3.2.3.1 | External High-speed (EHS) Clock | 47 |
| 3.2.3.2 | CRYSTAL/CERAMIC..... | 47 |
| 3.2.3.3 | External Low-speed (ELS) Clock..... | 48 |
| 3.2.3.4 | CRYSTAL..... | 48 |
| 3.2.3.5 | Bypass Mode | 48 |
| 3.2.4 | <i>SYSTEM CLOCK (SYSCLK) SELECTION</i> | 49 |
| 3.2.5 | <i>CLOCK-OUT CAPABILITY</i> | 49 |
| 3.3 | SYSTEM CONTROL REGISTERS 0 | 50 |
| 3.3.1 | <i>Analog Block Control register (SYS0_ANBCTRL)</i> | 50 |
| 3.3.2 | <i>PLL control register (SYS0_PLLCTRL)</i> | 50 |
| 3.3.2.1 | RECOMMEND FREQUENCY SETTING..... | 51 |
| 3.3.3 | <i>Clock Source Status register (SYS0_CSST)</i> | 51 |
| 3.3.4 | <i>System Clock Configuration register (SYS0_CLKCFG)</i> | 52 |
| 3.3.5 | <i>AHB Clock Prescale register (SYS0_AHBCP)</i> | 52 |
| 3.3.6 | <i>System Reset Status register (SYS0_RSTST)</i> | 52 |
| 3.3.7 | <i>LVD Control register (SYS0_LVDCTRL)</i> | 53 |
| 3.3.8 | <i>External RESET Pin Control register (SYS0_EXRSTCTRL)</i> | 54 |
| 3.3.9 | <i>SWD Pin Control register (SYS0_SWDCtrl)</i> | 54 |
| 3.4 | SYSTEM CONTROL REGISTERS 1 | 55 |
| 3.4.1 | <i>AHB Clock Enable register (SYS1_AHBCLKEN)</i> | 55 |
| 3.4.2 | <i>APB Clock Prescale register 0 (SYS1_APB0)</i> | 56 |
| 3.4.3 | <i>APB Clock Prescale register 1 (SYS1_APB1)</i> | 57 |

| | | |
|----------|--|-----------|
| 3.4.4 | Peripheral Reset register (SYSI_PRST)..... | 58 |
| 4 | SYSTEM OPERATION MODE | 60 |
| 4.1 | OVERVIEW | 60 |
| 4.2 | NORMAL MODE | 60 |
| 4.3 | LOW-POWER MODES | 60 |
| 4.3.1 | SLEEP MODE | 60 |
| 4.3.2 | DEEP-SLEEP MODE..... | 61 |
| 4.3.3 | DEEP POWER-DOWN (DPD) MODE..... | 61 |
| 4.3.3.1 | Entering Deep power-down mode..... | 62 |
| 4.3.3.2 | Exiting Deep power-down mode..... | 62 |
| 4.4 | WAKEUP INTERRUPT | 62 |
| 4.5 | WAKEUP | 62 |
| 4.5.1 | OVERVIEW | 62 |
| 4.5.2 | WAKEUP TIME..... | 62 |
| 4.6 | STATE MACHINE OF PMU | 64 |
| 4.7 | OPERATION MODE COMPARSION TABLE | 65 |
| 4.8 | PMU REGISTERS | 66 |
| 4.8.1 | Backup registers 0 to 15 (PMU_BKP0~15)..... | 66 |
| 4.8.2 | Power control register (PMU_CTRL)..... | 66 |
| 5 | GENERAL PURPOSE I/O PORT (GPIO) | 67 |
| 5.1 | OVERVIEW | 67 |
| 5.2 | GPIO MODE | 67 |
| 5.3 | GPIO REGISTERS..... | 68 |
| 5.3.1 | GPIO Port n Data register (GPIO _n _DATA) (n=0,1,2,3)..... | 68 |
| 5.3.2 | GPIO Port n Mode register (GPIO _n _MODE) (n=0,1,2,3)..... | 68 |
| 5.3.3 | GPIO Port n Configuration register (GPIO _n _CFG) (n=0,1,2,3)..... | 68 |
| 5.3.4 | GPIO Port n Interrupt Sense register (GPIO _n _IS) (n=0,1,2,3)..... | 69 |
| 5.3.5 | GPIO Port n Interrupt Both-edge Sense register (GPIO _n _IBS) (n=0,1,2,3)..... | 69 |
| 5.3.6 | GPIO Port n Interrupt Event register (GPIO _n _IEV) (n=0,1,2,3)..... | 70 |
| 5.3.7 | GPIO Port n Interrupt Enable register (GPIO _n _IE) (n=0,1,2,3)..... | 70 |
| 5.3.8 | GPIO Port n Raw Interrupt Status register (GPIO _n _RIS) (n=0,1,2,3) | 70 |
| 5.3.9 | GPIO Port n Interrupt Clear register (GPIO _n _IC) (n=0,1,2,3)..... | 70 |
| 5.3.10 | GPIO Port n Bits Set Operation register (GPIO _n _BSET) (n=0,1,2,3)..... | 71 |
| 5.3.11 | GPIO Port n Bits Clear Operation register (GPIO _n _BCLR) (n=0,1,2,3)..... | 71 |
| 5.3.12 | GPIO Port n Open-Drain Control register (GPIO _n _ODCTRL) (n=0,1,2,3) | 71 |
| 6 | 10 CHANNEL ANALOG TO DIGITAL CONVERTOR (ADC) | 73 |
| 6.1 | OVERVIEW | 73 |
| 6.2 | ADC CONVERTING TIME | 74 |

| | | |
|----------|--|-----------|
| 6.3 | ADC CONTROL NOTICE..... | 75 |
| 6.3.1 | ADC SIGNAL..... | 75 |
| 6.3.2 | ADC PROGRAM..... | 75 |
| 6.4 | ADC CIRCUIT..... | 75 |
| 6.5 | ADC REGISTERS..... | 76 |
| 6.5.1 | ADC Management register (ADC_ADM)..... | 76 |
| 6.5.2 | ADC Data register (ADC_ADB)..... | 77 |
| 6.5.3 | Port 2 Control register (ADC_P2CON)..... | 77 |
| 6.5.4 | ADC Interrupt Enable register (ADC_IE)..... | 78 |
| 6.5.5 | ADC Raw Interrupt Status register (ADC_RIS)..... | 78 |
| 7 | 16-BIT TIMER WITH CAPTURE FUNCTION | 79 |
| 7.1 | OVERVIEW..... | 79 |
| 7.2 | FEATURES..... | 79 |
| 7.3 | PIN DESCRIPTION..... | 79 |
| 7.4 | BLOCK DIAGRAM..... | 80 |
| 7.5 | TIMER OPERATION..... | 81 |
| 7.6 | PWM..... | 82 |
| 7.7 | CT16Bn REGISTERS..... | 83 |
| 7.7.1 | CT16Bn Timer Control register (CT16Bn_TMRCTRL) (n=0,1)..... | 83 |
| 7.7.2 | CT16Bn Timer Counter register (CT16Bn_TC) (n=0,1)..... | 83 |
| 7.7.3 | CT16Bn Count Control register (CT16Bn_CNTCTRL) (n=0,1)..... | 83 |
| 7.7.4 | CT16Bn Match Control register (CT16Bn_MCTRL) (n=0,1)..... | 84 |
| 7.7.5 | CT16Bn Match register 0~3 (CT16Bn_MR0~3) (n=0,1)..... | 85 |
| 7.7.6 | CT16Bn Capture Control register (CT16Bn_CAPCTRL) (n=0,1)..... | 85 |
| 7.7.7 | CT16Bn Capture 0 register (CT16Bn_CAP0) (n=0,1)..... | 85 |
| 7.7.8 | CT16Bn External Match register (CT16Bn_EM) (n=0)..... | 86 |
| 7.7.9 | CT16Bn External Match register (CT16Bn_EM) (n=1)..... | 86 |
| 7.7.10 | CT16Bn PWM Control register (CT16Bn_PWMCTRL) (n=0)..... | 87 |
| 7.7.11 | CT16Bn PWM Control register (CT16Bn_PWMCTRL) (n=1)..... | 87 |
| 7.7.12 | CT16Bn Timer Raw Interrupt Status register (CT16Bn_RIS) (n=0,1)..... | 88 |
| 7.7.13 | CT16Bn Timer Interrupt Clear register (CT16Bn_IC) (n=0,1)..... | 88 |
| 8 | 32-BIT TIMER WITH CAPTURE FUNCTION | 89 |
| 8.1 | OVERVIEW..... | 89 |
| 8.2 | FEATURES..... | 89 |
| 8.3 | PIN DESCRIPTION..... | 89 |
| 8.4 | BLOCK DIAGRAM..... | 90 |
| 8.5 | TIMER OPERATION..... | 91 |
| 8.6 | PWM..... | 92 |

| | | |
|-----------|---|------------|
| 8.7 | CT32Bn REGISTERS | 93 |
| 8.7.1 | CT32Bn Timer Control register (CT32Bn_TMRCTRL) (n=0,1) | 93 |
| 8.7.2 | CT32Bn Timer Counter register (CT32Bn_TC) (n=0,1) | 93 |
| 8.7.3 | CT32Bn Count Control register (CT16Bn_CNTCTRL) (n=0,1)..... | 93 |
| 8.7.4 | CT32Bn Match Control register (CT32Bn_MCTRL) (n=0,1) | 94 |
| 8.7.5 | CT32Bn Match register 0~3 (CT32Bn_MR0~3) (n=0,1)..... | 95 |
| 8.7.6 | CT32Bn Capture Control register (CT32Bn_CAPCTRL) (n=0,1) | 95 |
| 8.7.7 | CT32Bn Capture 0 register (CT32Bn_CAP0) (n=0,1) | 95 |
| 8.7.8 | CT32Bn External Match register (CT32Bn_EM) (n=0,1) | 95 |
| 8.7.9 | CT32Bn PWM Control register (CT32Bn_PWMCTRL) (n=0,1)..... | 96 |
| 8.7.10 | CT32Bn Timer Raw Interrupt Status register (CT32Bn_RIS) (n=0,1) | 97 |
| 8.7.11 | CT32Bn Timer Interrupt Clear register (CT32Bn_IC) (n=0,1)..... | 97 |
| 9 | WATCHDOG TIMER (WDT)..... | 98 |
| 9.1 | OVERVIEW | 98 |
| 9.2 | BLOCK DIAGRAM..... | 99 |
| 9.3 | WDT REGISTERS | 100 |
| 9.3.1 | Watchdog Configuration register (WDT_CFG)..... | 100 |
| 9.3.2 | Watchdog Clock Source register (WDT_CLKSOURCE)..... | 100 |
| 9.3.3 | Watchdog Timer Constant register (WDT_TC)..... | 100 |
| 9.3.4 | Watchdog Feed register (WDT_FEED)..... | 101 |
| 10 | REAL-TIME CLOCK (RTC) | 102 |
| 10.1 | OVERVIEW | 102 |
| 10.2 | FEATURES | 102 |
| 10.3 | FUNCTIONAL DESCRIPTION | 102 |
| 10.3.1 | INTRODUCTION | 102 |
| 10.3.2 | RESET RTC REGISTERS..... | 102 |
| 10.3.3 | RTC FLAG ASSERTION | 102 |
| 10.3.4 | RTC OPERATION | 103 |
| 10.4 | BLOCK DIAGRAM..... | 104 |
| 10.5 | RTC REGISTERS | 105 |
| 10.5.1 | RTC Control register (RTC_CTRL) | 105 |
| 10.5.2 | RTC Clock Source Select register (RTC_CLKS)..... | 105 |
| 10.5.3 | RTC Interrupt Enable register (RTC_IE)..... | 105 |
| 10.5.4 | RTC Raw Interrupt Status register (RTC_RIS) | 105 |
| 10.5.5 | RTC Interrupt Clear register (RTC_IC)..... | 106 |
| 10.5.6 | RTC Second Counter Reload Value register (RTC_SECCNTV)..... | 106 |
| 10.5.7 | RTC Second Count register (RTC_SECCNT) | 106 |
| 10.5.8 | RTC Alarm Counter Reload Value register (RTC_ALMCNTV)..... | 106 |

| | | |
|-----------|---|------------|
| 10.5.9 | RTC Alarm Count register (<i>RTC_ALMCNT</i>)..... | 108 |
| 11 | SPI/SSP..... | 109 |
| 11.1 | OVERVIEW..... | 109 |
| 11.2 | FEATURES..... | 109 |
| 11.3 | PIN DESCRIPTION..... | 110 |
| 11.4 | INTERFACE DESCRIPTION..... | 111 |
| 11.4.1 | SPI..... | 111 |
| 11.4.2 | SSI..... | 112 |
| 11.4.3 | COMMUNICATION FLOW..... | 112 |
| 11.4.3.1 | SINGLE-FRAME..... | 112 |
| 11.4.3.2 | MULTI-FRAME..... | 113 |
| 11.5 | SSP REGISTERS..... | 114 |
| 11.5.1 | SSP <i>n</i> Control register 0 (<i>SSPn_CTRL0</i>) (<i>n</i> =0, 1)..... | 114 |
| 11.5.2 | SSP <i>n</i> Control register 1 (<i>SSPn_CTRL1</i>) (<i>n</i> =0, 1)..... | 114 |
| 11.5.3 | SSP <i>n</i> Clock Divider register (<i>SSPn_CLKDIV</i>) (<i>n</i> =0, 1)..... | 115 |
| 11.5.4 | SSP <i>n</i> Status register (<i>SSPn_STAT</i>) (<i>n</i> =0, 1)..... | 115 |
| 11.5.5 | SSP <i>n</i> Interrupt Enable register (<i>SSPn_IE</i>) (<i>n</i> =0, 1)..... | 115 |
| 11.5.6 | SSP <i>n</i> Raw Interrupt Status register (<i>SSPn_RIS</i>) (<i>n</i> =0, 1)..... | 116 |
| 11.5.7 | SSP <i>n</i> Interrupt Clear register (<i>SSPn_IC</i>) (<i>n</i> =0, 1)..... | 116 |
| 11.5.8 | SSP <i>n</i> Data register (<i>SSPn_DATA</i>) (<i>n</i> =0, 1)..... | 116 |
| 11.5.9 | SSP <i>n</i> Data Fetch register (<i>SSPn_DF</i>) (<i>n</i> =0, 1)..... | 117 |
| 12 | I2C..... | 118 |
| 12.1 | OVERVIEW..... | 118 |
| 12.2 | FEATURES..... | 118 |
| 12.3 | PIN DESCRIPTION..... | 118 |
| 12.4 | I2C PROPOCOL..... | 119 |
| 12.4.1 | 7-BIT ADDRESSING MODES..... | 119 |
| 12.4.1.1 | MASTER TRANSMITTER MODE..... | 119 |
| 12.4.1.2 | MASTER RECEIVER MODE..... | 120 |
| 12.4.1.3 | SLAVE TRANSMITTER MODE..... | 120 |
| 12.4.1.4 | SLAVE RECEIVER MODE..... | 120 |
| 12.4.2 | 10-BIT ADDRESSING MODES..... | 120 |
| 12.4.2.1 | MASTER TRANSMITTER MODE..... | 120 |
| 12.4.2.2 | MASTER RECEIVER MODE..... | 120 |
| 12.5 | ARBITRATION..... | 121 |
| 12.6 | CLOCK STRETCHING..... | 121 |
| 12.7 | GENERAL CALL ADDRESS..... | 121 |
| 12.8 | TIMING CHARACTERISTICS..... | 122 |

| | | |
|-----------|--|------------|
| 12.8.1 | MASTER TRANSMITTER MODE | 122 |
| 12.8.2 | SLAVE TRANSMITTER MODE | 122 |
| 12.9 | MONITOR MODE | 123 |
| 12.9.1 | INTERRUPT | 123 |
| 12.9.2 | LOSS of ARBITRATION | 123 |
| 12.10 | I2C REGISTERS | 124 |
| 12.10.1 | I2C n Control register (I2Cn_CTRL) (n=0,1) | 124 |
| 12.10.2 | I2C n Status register (I2Cn_STAT) (n=0,1) | 125 |
| 12.10.3 | I2C n TX Data register (I2Cn_TXDATA) (n=0,1) | 126 |
| 12.10.4 | I2C n RX Data register (I2Cn_RXDATA) (n=0,1) | 126 |
| 12.10.5 | I2C n Slave Address 0 register (I2Cn_SLVADDR0) (n=0,1) | 126 |
| 12.10.6 | I2C n Slave Address 1~3 register (I2Cn_SLVADDR1~3) (n=0,1) | 126 |
| 12.10.7 | I2C n SCL High Time register (I2Cn_SCLHT) (n=0,1) | 127 |
| 12.10.8 | I2C n SCL Low Time register (I2Cn_SCLLT) (n=0,1) | 127 |
| 12.10.9 | I2C n Timeout Control register (I2Cn_TOCTRL) (n=0,1) | 127 |
| 12.10.10 | I2C n Monitor Mode Control register (I2Cn_MMCTRL) (n=0,1) | 127 |
| 13 | UNIVERSAL SYNCHRONOUS AND ASYNCHRONOUS RECEIVER AND TRANSMITTER | |
| | (USART) | 129 |
| 13.1 | OVERVIEW | 129 |
| 13.2 | FEATURES | 129 |
| 13.3 | PIN DESCRIPTION | 129 |
| 13.4 | BLOCK DIAGRAM | 131 |
| 13.5 | EIA-485/RS-485 MODES | 132 |
| 13.5.1 | RS-485/EIA-485 NORMAL MULTIDROP MODE (NMM) | 132 |
| 13.5.2 | RS-485/EIA-485 AUTO ADDRESS DETECTION (AAD) MODE | 132 |
| 13.5.3 | RS-485/EIA-485 AUTO DIRECTION CONTROL (ADC) | 132 |
| 13.5.4 | RS485/EIA-485 DRIVER DELAY TIME | 133 |
| 13.5.5 | RS485/EIA-485 OUTPUT INVERSION | 133 |
| 13.5.6 | RS485/EIA-485 FRAME STRUCTURE | 133 |
| 13.6 | BAUD RATE CALCULATION | 133 |
| 13.7 | MODEM CONTROL (MC) | 135 |
| 13.7.1 | AUTO-RTS | 135 |
| 13.7.2 | AUTO-CTS | 136 |
| 13.8 | AUTO-BAUD FLOW | 136 |
| 13.8.1 | AUTO-BAUD | 136 |
| 13.8.2 | AUTO-BAUD MODES | 137 |
| 13.9 | SERIAL IRDA (SIR) MODE | 138 |
| 13.10 | SMART CARD MODE | 140 |
| 13.10.1 | SMART CARD SETUP PROCEDURE | 140 |

| | | |
|-----------|---|------------|
| 13.11 | SYNCHRONOUS MODE..... | 141 |
| 13.12 | USART REGISTERS | 142 |
| 13.12.1 | USART n Receiver Buffer register (USARTn_RB) (n=0, 1)..... | 142 |
| 13.12.2 | USART n Transmitter Holding register (USARTn_TH) (n=0, 1)..... | 142 |
| 13.12.3 | USART n Divisor Latch LSB registers (USARTn_DLL) (n =0, 1)..... | 142 |
| 13.12.4 | USART n Divisor Latch MSB register (USARTn_DLM) (n=0,1) | 142 |
| 13.12.5 | USART n Interrupt Enable register (USARTn_IE) (n=0, 1) | 143 |
| 13.12.6 | USART n Interrupt Identification register (USARTn_II) (n=0,1) | 143 |
| 13.12.7 | USART n FIFO Control register (USARTn_FIFOCTRL) (n=0,1)..... | 145 |
| 13.12.8 | USART n Line Control register (USARTn_LC) (n=0,1) | 145 |
| 13.12.9 | USART n Modem Control register (USARTn_MC) (n=0,1) | 146 |
| 13.12.10 | USART n Line Status register (USARTn_LS) (n=0,1)..... | 147 |
| 13.12.11 | USART n Modem Status register (USARTn_MS) (n=0,1)..... | 148 |
| 13.12.12 | USART n Scratch Pad register (USARTn_SP) (n=0, 1)..... | 149 |
| 13.12.13 | USART n Auto-baud Control register (USARTn_ABCTRL) (n=0, 1)..... | 149 |
| 13.12.14 | USART n IrDA Control register (USARTn_IRDACTRL) (n=0, 1) | 149 |
| 13.12.15 | USART n Fractional Divider register (USARTn_FD) (n=0, 1)..... | 150 |
| 13.12.16 | USART n Control register (USARTn_CTRL) (n=0, 1)..... | 150 |
| 13.12.17 | USART n Half-duplex Enable register (USARTn_HDEN) (n=0, 1) | 151 |
| 13.12.18 | USART n Smart card Interface Control register (USARTn_SCICTRL) (n=0, 1) | 151 |
| 13.12.19 | USART n RS485 Control register (USARTn_RS485CTRL) (n=0, 1)..... | 152 |
| 13.12.20 | USART n RS485 Address Match register (USARTn_RS485ADRMATCH) (n=0, 1) | 152 |
| 13.12.21 | USART n RS485 Delay Value register (USARTn_RS485DLYV) (n=0, 1)..... | 153 |
| 13.12.22 | USART n Synchronous Mode Control Register (USARTn_SYNCCTRL) (n=0,1) | 153 |
| 14 | I2S | 154 |
| 14.1 | OVERVIEW | 154 |
| 14.2 | FEATURES | 154 |
| 14.3 | PIN DESCRIPTION | 154 |
| 14.4 | BLOCK DIAGRAM..... | 155 |
| 14.4.1 | I2S CLCOK CONTROL..... | 155 |
| 14.4.2 | I2S BLOCK DIAGRAM | 155 |
| 14.5 | FUNCTIONAL DESCRIPTION | 156 |
| 14.5.1 | I2S OPERATION | 156 |
| 14.5.2 | I2S FIFO OPERAION | 158 |
| 14.5.2.1 | MONO | 158 |
| 14.5.2.2 | STEREO | 158 |
| 14.6 | I2S REGISTERS..... | 159 |
| 14.6.1 | I2S Control register (I2S_CTRL) | 159 |
| 14.6.2 | I2S Clock register (I2S_CLK)..... | 160 |

| | | |
|-----------|--|------------|
| 14.6.3 | <i>I2S Status register (I2S_STATUS)</i> | 160 |
| 14.6.4 | <i>I2S Interrupt Enable register (I2S_IE)</i> | 161 |
| 14.6.5 | <i>I2S Raw Interrupt Status register (I2S_RIS)</i> | 161 |
| 14.6.6 | <i>I2S Interrupt Clear register (I2S_IC)</i> | 161 |
| 14.6.7 | <i>I2S FIFO register (I2S_FIFO)</i> | 161 |
| 15 | FLASH | 162 |
| 15.1 | OVERVIEW | 162 |
| 15.2 | EMBEDDED FLASH MEMORY | 162 |
| 15.3 | FEATURES | 162 |
| 15.4 | ORGANIZATION..... | 163 |
| 15.5 | READ | 163 |
| 15.6 | PROGRAM/ERASE..... | 163 |
| 15.7 | EMBEDDED BOOT LOADER | 163 |
| 15.8 | FLASH MEMORY CONTROLLER (FMC)..... | 164 |
| 15.8.1 | <i>CODE SECURITY (CS)</i> | 164 |
| 15.8.2 | <i>PROGRAM FLASH MEMORY</i> | 166 |
| 15.8.3 | <i>ERASE</i> | 166 |
| 15.8.3.1 | <i>PAGE ERASE</i> | 166 |
| 15.8.3.2 | <i>MASS ERASE</i> | 166 |
| 15.9 | READ PROTECTION..... | 166 |
| 15.10 | FMC REGISTERS..... | 167 |
| 15.10.1 | <i>Flash Status register (FLASH_STATUS)</i> | 167 |
| 15.10.2 | <i>Flash Control register (FLASH_CTRL)</i> | 167 |
| 15.10.3 | <i>Flash Data register (FLASH_DATA)</i> | 167 |
| 15.10.4 | <i>Flash Address register (FLASH_ADDR)</i> | 169 |
| 15.10.5 | <i>Flash Checksum register (FLASH_CHKSUM)</i> | 169 |
| 16 | SERIAL-WIRE DEBUG (SWD) | 170 |
| 16.1 | OVERVIEW | 170 |
| 16.2 | FEATURES | 170 |
| 16.3 | PIN DESCRIPTION | 170 |
| 16.4 | DEBUG NOTE..... | 170 |
| 16.4.1 | <i>LIMITATIONS</i> | 170 |
| 16.4.2 | <i>DEBUG RECOVERY</i> | 170 |
| 16.4.3 | <i>INTERNAL PULL-UP/DOWN RESISTORS on SWD PINS</i> | 171 |
| 17 | DEVELOPMENT TOOL | 172 |
| 17.1 | SN-LINK-V3 | 173 |
| 17.2 | SN32F707 STARTER-KIT | 174 |

| | | |
|-----------|--|------------|
| 18 | ELECTRICAL CHARACTERISTIC | 175 |
| 18.1 | ABSOLUTE MAXIMUM RATING | 175 |
| 18.2 | ELECTRICAL CHARACTERISTIC | 175 |
| 18.3 | CHARACTERISTIC GRAPHS | 177 |
| 19 | FLASH ROM PROGRAMMING PIN | 178 |
| 20 | PACKAGE INFORMATION | 180 |
| 20.1 | LQFP 48 PIN | 180 |
| 20.2 | QFN 46 PIN | 181 |
| 20.3 | QFN 33 PIN 4x4 | 182 |
| 20.4 | QFN 33 PIN 5x5 | 183 |
| 20.5 | SK-DIP 28 PIN | 184 |
| 20.6 | SOP 28 PIN | 185 |
| 20.7 | SSOP 28 PIN | 186 |
| 20.8 | SK-DIP 24 PIN | 187 |
| 20.9 | SOP 24 PIN | 188 |
| 20.10 | SSOPP 24 PIN | 189 |
| 20.11 | QFN 24 PIN | 190 |
| 21 | MARKING DEFINITION | 191 |
| 21.1 | INTRODUCTION | 191 |
| 21.2 | MARKING INDETIFICATION SYSTEM | 191 |
| 21.3 | MARKING EXAMPLE | 192 |
| 21.4 | DATECODE SYSTEM | 193 |

1 PRODUCT OVERVIEW

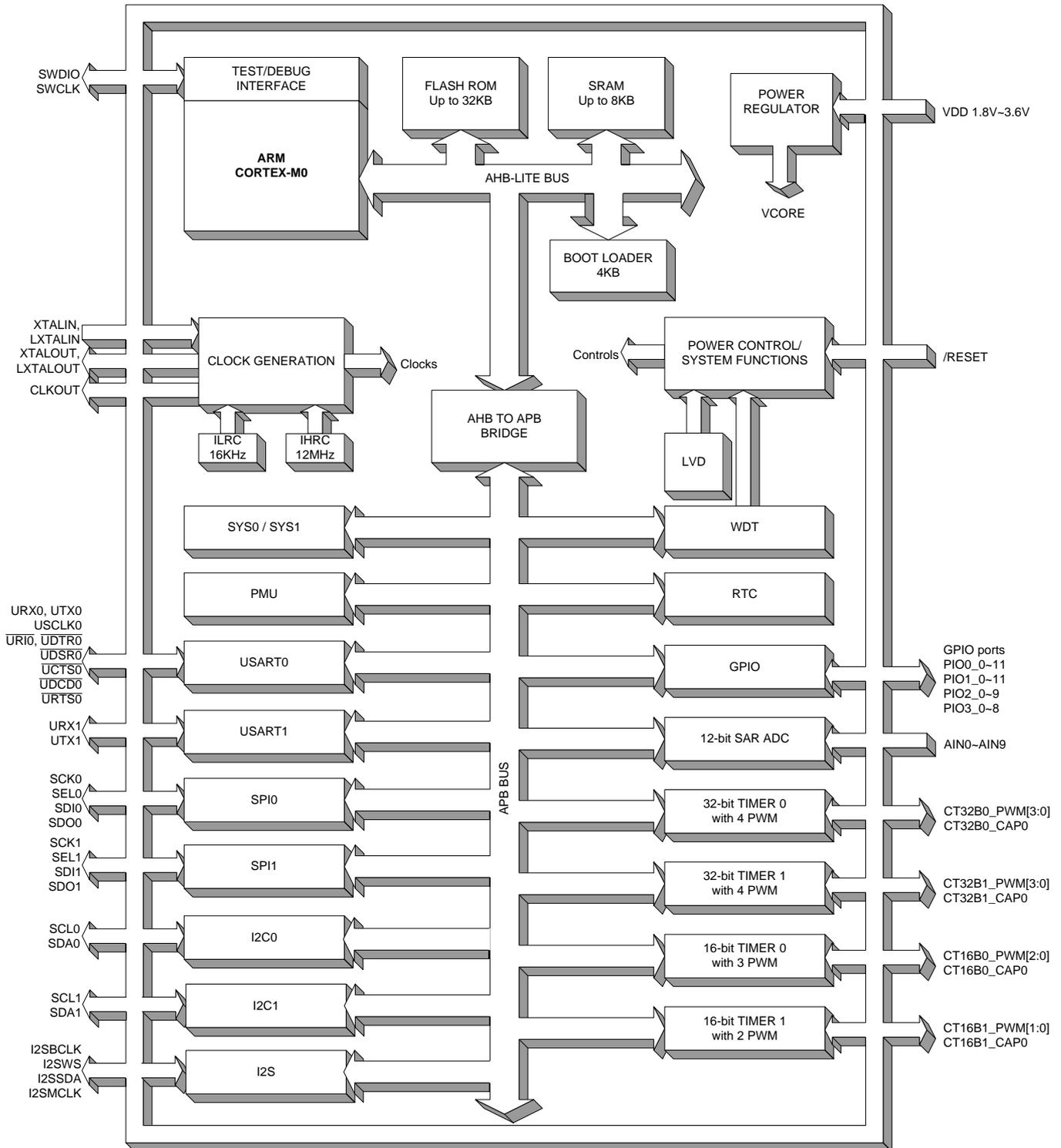
1.1 FEATURES

- ◆ **Memory configuration**
32KB on-chip Flash programming memory.
8KB SRAM.
4KB Boot ROM
- ◆ **Operation Frequency up to 50MHz**
- ◆ **Interrupt sources**
ARM Cortex-M0 built-in Nested Vectored Interrupt Controller (NVIC).
- ◆ **I/O pin configuration**
Up to 43 General Purpose I/O (GPIO) pins with configurable pull-up/pull-down resistors.
GPIO pins can be used as edge and level sensitive interrupt sources.
High-current source driver (20 mA)
- ◆ **Programmable Watchdog Timer (WDT)**
Programmable watchdog frequency with watchdog clock source and divider.
- ◆ **System tick timer**
24-bit timer.
The system tick timer clock is fixed to the frequency of the system clock.
The SysTick timer is intended to generate a fixed 10-ms interrupt.
- ◆ **Real-Time Clock (RTC)**
- ◆ **LVD with separate thresholds**
Reset: 1.65V for V_{CORE} 1.8V, 2.0/2.4/2.7V for VDD
Interrupt: 2.0/2.7/3.0V for VDD
- ◆ **F_{CPU} (Instruction cycle)**
 $F_{CPU} = F_{HCLK} = F_{SYSCLK}/1, F_{SYSCLK}/2, F_{SYSCLK}/4, \dots, F_{SYSCLK}/512.$
- ◆ **Operating modes**
Normal
Sleep
Deep-sleep
Deep power-down
- ◆ **Timer**
Two 16-bit and two 32-bit general purpose timers with a total of four capture inputs and 13PWMs.
- ◆ **Working voltage 1.8V ~ 3.6V**
- ◆ **ADC**
10-channel 12-bit SAR ADC.
- ◆ **Interface**
-Two I2C controllers supporting I2C-bus specification with multiple address recognition and monitor mode.
-Two USART controllers with fractional baud rate generation, and EIA-485 support.
-Two SPI controllers with SSP features and multi-protocol capabilities.
-I2S Function with mono and stereo audio data supported, MSB justified data format supported, and can operate as either master or slave.
- ◆ **System clocks**
-External high clock: Crystal type 10MHz~25MHz
-External low clock: Crystal type 32.768 KHz
-Internal high clock: RC type 12 MHz
-Internal low clock: RC type 16 KHz
-PLL allows CPU operation up to the maximum CPU rate without the need for a high-frequency crystal. May be run from the external high clock or the internal high RC oscillator.
-Clock output function which can reflect the internal high/low RC oscillator, HCLK, PLL output, and external high/low clock.
- ◆ **Serial Wire Debug (SWD)**
- ◆ **In-System Programming (ISP) supported**
- ◆ **Package (Chip form support)**
LQFP 48/32 pin
QFN 46/33/24 pin
SK-DIP 28/24 pin
SOP 28/24 pin
SSOP 28/24 pin

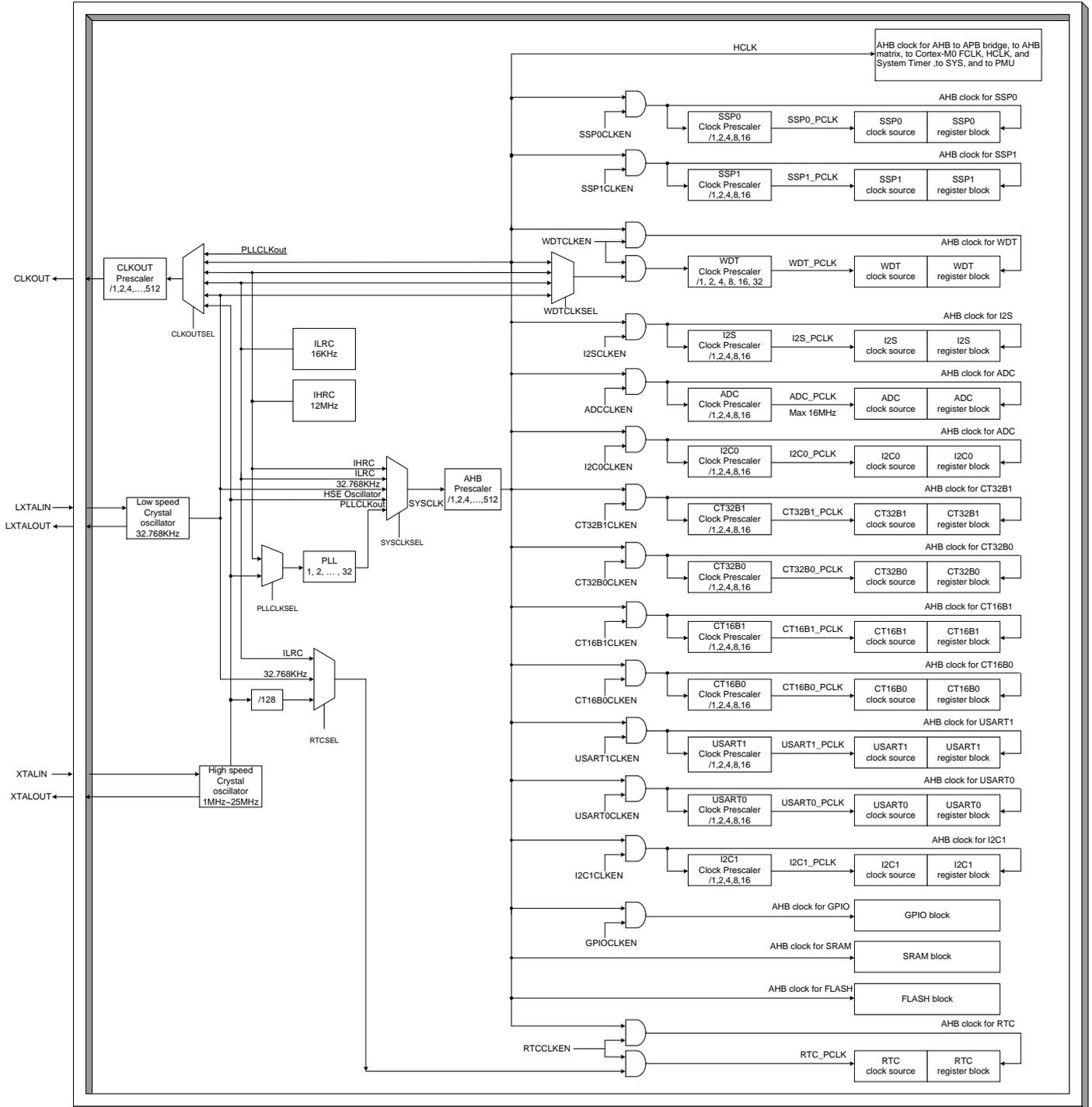
Features Selection Table

| Chip | ROM | RAM | Fcpu. (Max MHz) | USART | TIMER | SPI | I2C | I2S | PWM | 12-bit ADC | GPIO pin | Wakeup pin | Package |
|------------|------|-----|-----------------------|-------------------|----------------------|-----|-----|-----|-----|---------------|-------------|---------------|--------------|
| SN32F707F | 32KB | 8KB | 50 | USARTx1 UARTx1 | 16-bitx2 32-bitx2 | 2 | 2 | 1 | 13 | 10CH | 43 | 13 | LQFP48 |
| SN32F706J | 32KB | 8KB | 50 | USARTx1 | 16-bitx2 32-bitx2 | 2 | 2 | 1 | 13 | 10CH | 41 | 11 | QFN46 |
| SN32F705J | 32KB | 8KB | 50 | UARTx1 | 16-bitx2 32-bitx2 | 1 | 2 | 1 | 11 | 5CH | 27 | 11 | QFN33 4x4 |
| SN32F7051J | 32KB | 8KB | 50 | UARTx1 | 16-bitx1 32-bitx2 | 1 | 2 | 1 | 11 | 5CH | 27 | 11 | QFN33 5x5 |
| SN32F704K | 32KB | 8KB | 50 | UARTx1 | 16-bitx1 32-bitx2 | 1 | 1 | 1 | 10 | 2CH | 23 | 10 | SK-DIP28 |
| SN32F704S | 32KB | 8KB | 50 | UARTx1 | 16-bitx1 32-bitx2 | 1 | 1 | 1 | 10 | 2CH | 23 | 10 | SOP28 |
| SN32F704X | 32KB | 8KB | 50 | UARTx1 | 16-bitx1 32-bitx2 | 1 | 1 | 1 | 10 | 2CH | 23 | 10 | SSOP28 |
| SN32F703K | 32KB | 8KB | 50 | UARTx1 | 16-bitx1 32-bitx2 | 1 | 1 | 1 | 11 | 2CH | 19 | 8 | SK-DIP24 |
| SN32F703S | 32KB | 8KB | 50 | UARTx1 | 16-bitx1 32-bitx2 | 1 | 1 | 1 | 11 | 2CH | 19 | 8 | SOP24 |
| SN32F703X | 32KB | 8KB | 50 | UARTx1 | 16-bitx1 32-bitx2 | 1 | 1 | 1 | 11 | 2CH | 19 | 8 | SSOP24 |
| SN32F7031J | 32KB | 8KB | 50 | UARTx1 | 16-bitx1 32-bitx2 | 1 | 1 | 1 | 11 | 2CH | 19 | 8 | QFN24 |

1.2 SYSTEM BLOCK DIAGRAM

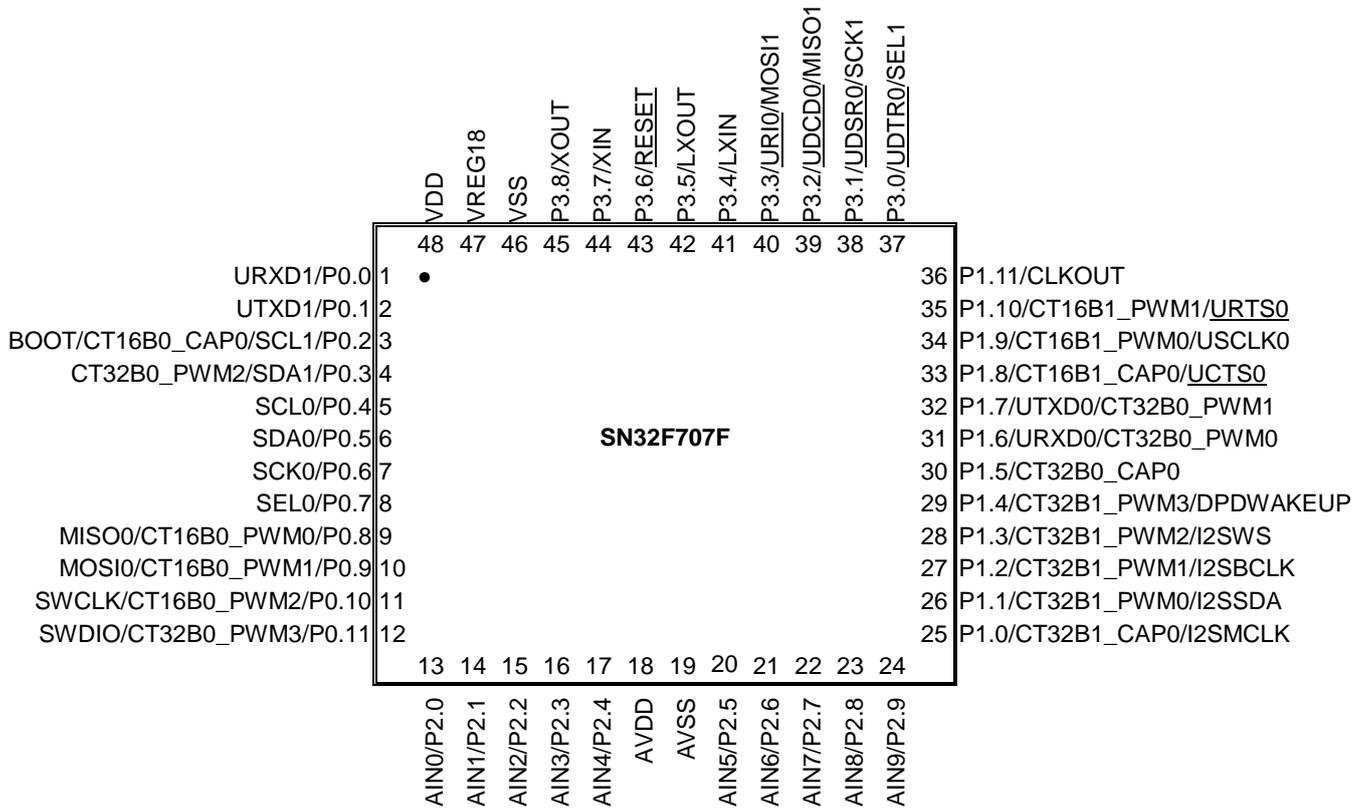


1.3 CLOCK GENERATION BLOCK DIAGRAM



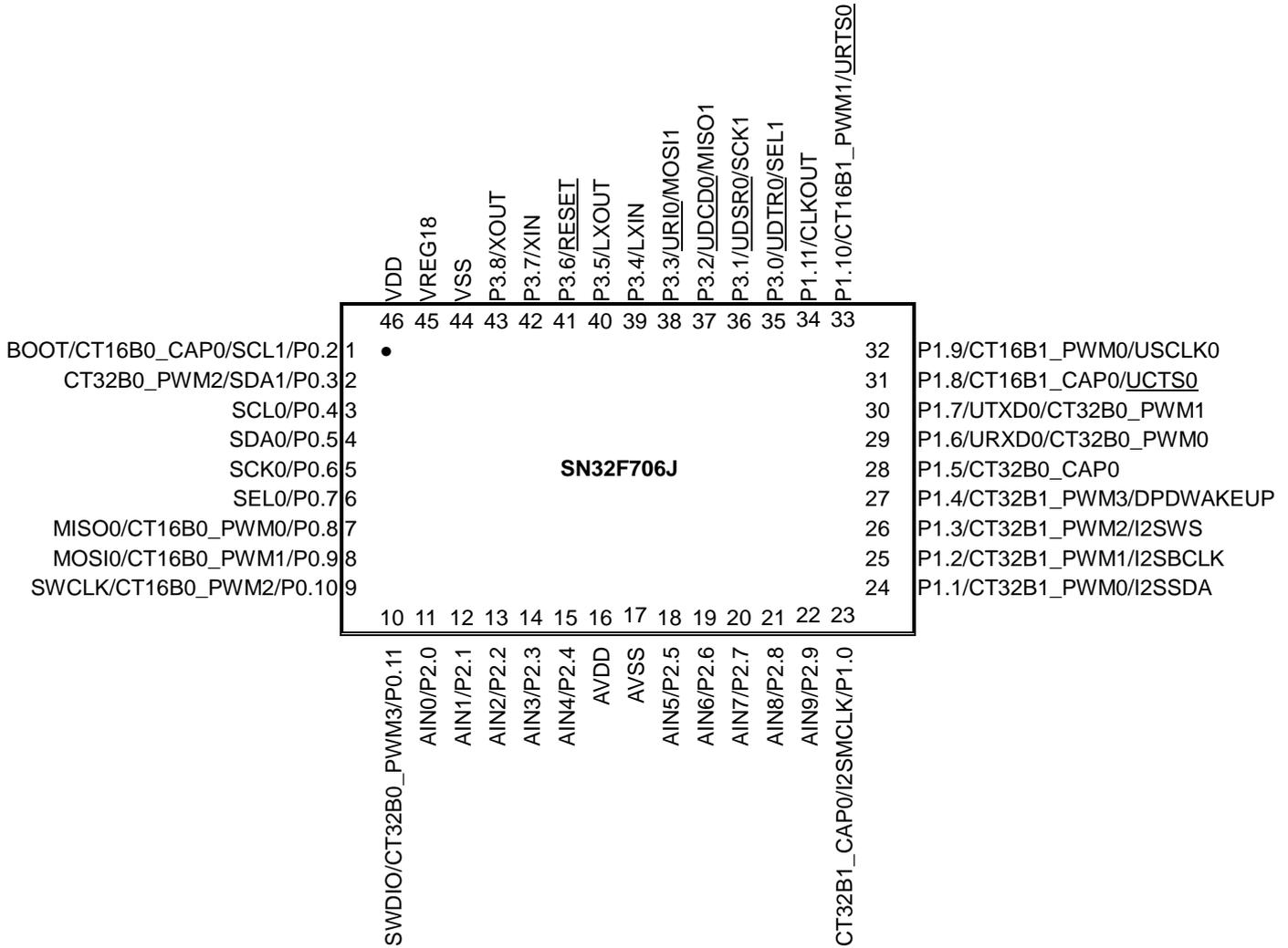
1.4 PIN ASSIGNMENT

SN32F707F (LQFP 48 pins)



* **Note:** SONiX provide Boot loader to check the status of P0.2 (BOOT pin) during boot procedure. If BOOT pin is Low during Boot procedure, MCU will execute code in Boot loader instead of User code. We strongly recommended NOT using BOOT pin as output pin to drive the LED, otherwise, the BOOT pin status may be low during boot procedure.

SN32F706J (QFN 46 pins)



* **Note:**

1. The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.
2. SONiX provide Boot loader to check the status of P0.2 (BOOT pin) during boot procedure. If BOOT pin is Low during Boot procedure, MCU will execute code in Boot loader instead of User code. We strongly recommended NOT using BOOT pin as output pin to drive the LED, otherwise, the BOOT pin status may be low during boot procedure.

SN32F704K (SK-DIP 28 pins)
SN32F704S (SOP 28 pins)
SN32F704X (SSOP 28 pins)

| | | | | |
|----------------------------|----|---|----|----------------------------|
| MISO0/CT16B0_PWM0/P0.8 | 1 | U | 28 | P0.7/SEL0 |
| MOSI0/CT16B0_PWM1/P0.9 | 2 | | 27 | P0.6/SCK0 |
| SWCLK/CT16B0_PWM2/P0.10 | 3 | | 26 | P0.5/SDA0 |
| SWDIO/CT32B0_PWM3/P0.11 | 4 | | 25 | P0.4/SCL0 |
| AIN0/P2.0 | 5 | | 24 | P0.2/CT16B0_CAP0/SCL1/BOOT |
| AIN1/P2.1 | 6 | | 23 | VDD |
| AVDD | 7 | | 22 | VREG18 |
| AVSS | 8 | | 21 | VSS |
| CT32B1_CAP0/I2SMCLK/P1.0 | 9 | | 20 | P3.8/XOUT |
| CT32B1_PWM0/I2SSDA/P1.1 | 10 | | 19 | P3.7/XIN |
| CT32B1_PWM1/I2SBCLK/P1.2 | 11 | | 18 | P3.6/RESET |
| CT32B1_PWM2/I2SWS/P1.3 | 12 | | 17 | P3.5/LXOUT |
| CT32B1_PWM3/DPDWAKEUP/P1.4 | 13 | | 16 | P3.4/LXIN |
| URXD0/CT32B0_PWM0/P1.6 | 14 | | 15 | P1.7/UTXD0/CT32B0_PWM1 |

SN32F704K
SN32F704S
SN32F704X

- * **Note:**
1. The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.
 2. SONiX provide Boot loader to check the status of P0.2 (BOOT pin) during boot procedure. If BOOT pin is Low during Boot procedure, MCU will execute code in Boot loader instead of User code. We strongly recommended NOT using BOOT pin as output pin to drive the LED, otherwise, the BOOT pin status may be low during boot procedure.

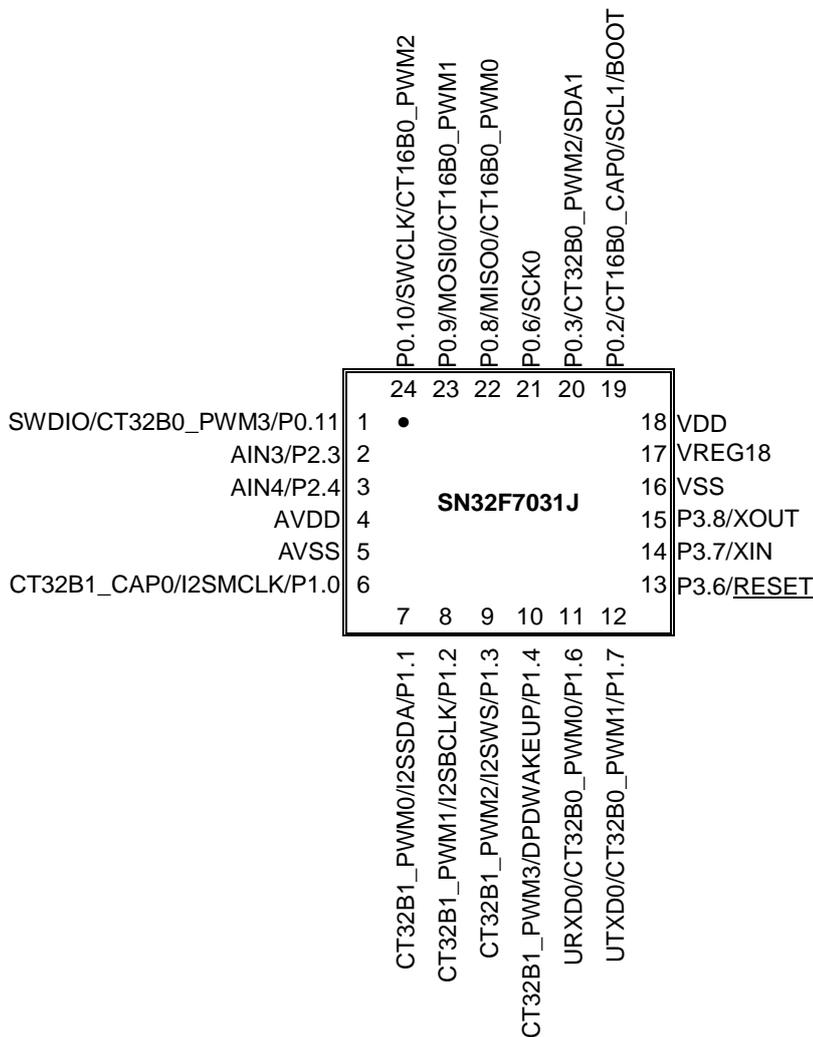
SN32F703K (SK-DIP 24 pins)
SN32F703S (SOP 24 pins)
SN32F703X (SSOP 24 pins)

| | | | | |
|----------------------------|----|---|----|----------------------------|
| MOSI0/CT16B0_PWM1/P0.9 | 1 | U | 24 | P0.8/MISO0/CT16B0_PWM0 |
| SWCLK/CT16B0_PWM2/P0.10 | 2 | | 23 | P0.6/SCK0 |
| SWDIO/CT32B0_PWM3/P0.11 | 3 | | 22 | P0.3/CT32B0_PWM2/SDA1 |
| AIN3/P2.3 | 4 | | 21 | P0.2/CT16B0_CAP0/SCL1/BOOT |
| AIN4/P2.4 | 5 | | 20 | VDD |
| AVDD | 6 | | 19 | VREG18 |
| AVSS | 7 | | 18 | VSS |
| CT32B1_CAP0/I2SMCLK/P1.0 | 8 | | 17 | P3.8/XOUT |
| CT32B1_PWM0/I2SSDA/P1.1 | 9 | | 16 | P3.7/XIN |
| CT32B1_PWM1/I2SBCLK/P1.2 | 10 | | 15 | P3.6/RESET |
| CT32B1_PWM2/I2SWS/P1.3 | 11 | | 14 | P1.7/UTXD0/CT32B0_PWM1 |
| CT32B1_PWM3/DPDWAKEUP/P1.4 | 12 | | 13 | P1.6/URXD0/CT32B0_PWM0 |

SN32F703K
SN32F703S
SN32F703X

- * **Note:**
1. The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.
 2. SONiX provide Boot loader to check the status of P0.2 (BOOT pin) during boot procedure. If BOOT pin is Low during Boot procedure, MCU will execute code in Boot loader instead of User code. We strongly recommended NOT using BOOT pin as output pin to drive the LED, otherwise, the BOOT pin status may be low during boot procedure.

SN32F7031J (QFN 24 pins)



* **Note:**

1. The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.
2. SONiX provide Boot loader to check the status of P0.2 (BOOT pin) during boot procedure. If BOOT pin is Low during Boot procedure, MCU will execute code in Boot loader instead of User code. We strongly recommended NOT using BOOT pin as output pin to drive the LED, otherwise, the BOOT pin status may be low during boot procedure.

1.5 PIN DESCRIPTIONS

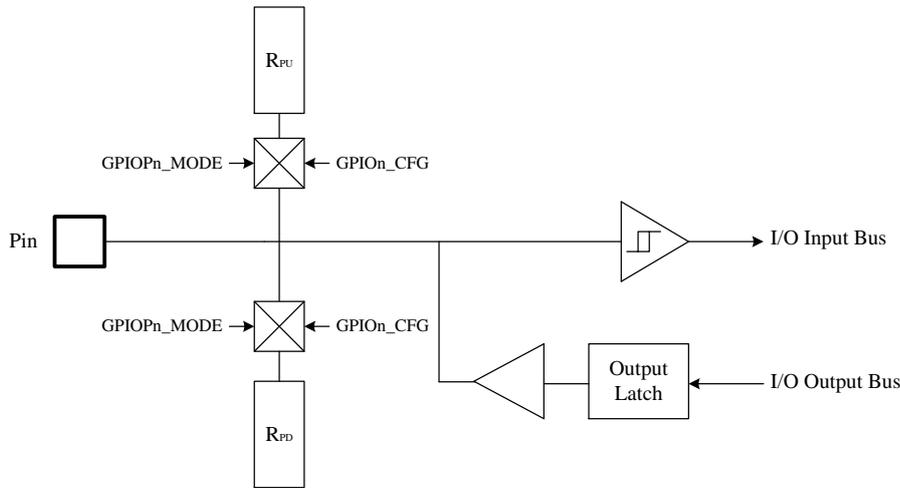
| PIN NAME | TYPE | DESCRIPTION |
|--------------------------------|------|--|
| VDD, VSS | P | Power supply input pins for digital circuit. |
| AVDD, AVSS | P | Power supply input pins for analog circuit. |
| VREG18 | P | 1.8V power pin. Please connect 1uF capacitor to GND. |
| P0.0/URXD1 | I/O | P0.0 — Port 0.0 bi-direction pin with high-current sink driver. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. Built-in wakeup function. URXD1 — Receiver data input pin for USART1. |
| P0.1/UTXD1 | I/O | P0.1 — Port 0.1 bi-direction pin with high-current sink driver. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. Built-in wakeup function. UTXD1 —USART1 Transmitter data output pin. |
| P0.2/CT16B0_CAP0/ SCL1/BOOT | I/O | P0.2 — Port 0.2 bi-direction pin with high-current sink driver. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. Built-in wakeup function. SCL1 — I2C1 clock pin. CT16B0_CAP0 — CT16B0 Capture input 0. BOOT — Internal pull-up in Boot loader, tie LOW to keep in Boot loader or left HIGH to exit Boot loader and execute User program at boot time. |
| P0.3/SDA1/ CT32B0_PWM2 | I/O | P0.3 — Port 0.3 bi-direction pin with high-current sink driver. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. Built-in wakeup function. SDA1 — I2C1 data pin. CT32B0_PWM2 —CT32B0 PWM output 2. |
| P0.4/SCL0 | I/O | P0.4 — Port 0.4 bi-direction pin with high-current sink driver. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. Built-in wakeup function. SCL0 — I2C0 clock pin. |
| P0.5/SDA0 | I/O | P0.5 — Port 0.5 bi-direction pin with high-current sink driver. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. Built-in wakeup function. SDA0 — I2C0 data pin |
| P0.6/SCK0 | I/O | P0.6 — Port 0.6 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. Built-in wakeup function. SCK0 — SSP0 Serial clock pin. |
| P0.7/SEL0 | I/O | P0.7 — Port 0.7 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. Built-in wakeup function. SEL0 —SSP0 Select pin. |
| P0.8/MISO0/ CT16B0_PWM0 | I/O | P0.8 — Port 0.8 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. Built-in wakeup function. MISO0 —SSP0 Master In Slave Out pin. CT16B0_PWM0 —CT16B0 PWM output 0. |
| P0.9/MOSI0/ CT16B0_PWM1 | I/O | P0.9 — Port 0.9 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. Built-in wakeup function. MOSI0 — SSP0 Master Out Slave In pin. CT16B0_PWM1 —CT16B0 PWM output 1. |
| P0.10/SWCLK/ CT16B0_PWM2 | I/O | P0.10 — Port 0.10 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. Built-in wakeup function. SWCLK — Serial Wire Clock pin. CT16B0_PWM2 —CT16B0 PWM output 2. |

| | | |
|--------------------------------|-----|---|
| P0.11/SWDIO/ CT32B0_PWM3 | I/O | P0.11 — Port 0.11 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. Built-in wakeup function. SWDIO — Serial Wire Debug input/output pin. CT32B0_PWM3 — CT32B0 PWM output 3. |
| P1.0/CT32B1_CAP0/ I2SMCLK | I/O | P1.0 — Port 1.0 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. CT32B1_CAP0 — CT32B1 Capture input 0. I2SMCLK — I2S Main Clock pin. |
| P1.1/CT32B1_PWM0/ I2SSDA | I/O | P1.1 — Port 1.1 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. CT32B1_PWM0 — PWM output 0 for CT32B1. I2SSDA — I2S Serial data pin. |
| P1.2/CT32B1_PWM1/ I2SBCLK | I/O | P1.2 — Port 1.2 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. CT32B1_PWM1 — PWM output 1 for CT32B1. I2SBCLK — I2S Bit Clock pin. |
| P1.3/CT32B1_PWM2/ I2SWS | I/O | P1.3 — Port 1.3 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. CT32B1_PWM2 — PWM output 2 for CT32B1. I2SWS — I2S Word Select pin. |
| P1.4/CT32B1_PWM3/ DPDWAKEUP | I/O | P1.4 — Port 1.4 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. Built-in wakeup function. CT32B1_PWM3 — CT32B1 PWM output 3. DPDWAKEUP — Deep power-down mode wake-up pin. |
| P1.5/CT32B0_CAP0 | I/O | P1.5 — Port 1.5 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. CT32B0_CAP0 — CT32B0 Capture input 0. |
| P1.6/URXD0/ CT32B0_PWM0 | I/O | P1.6 — Port 1.6 bi-direction pin with high-current sink driver. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. URXD0 —USART0 Receiver input pin. CT32B0_PWM0 — CT32B0 PWM output 0. |
| P1.7/UTXD0/ CT32B0_PWM1 | I/O | P1.7 — Port 1.7 bi-direction pin with high-current sink driver. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. UTXD0 — USART0 Transmitter output pin. CT32B0_PWM1 — CT32B0 PWM output 1. |
| P1.8/CT16B1_CAP0/ UCTS0 | I/O | P1.8 — Port 1.8 bi-direction pin with high-current sink driver. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. CT16B1_CAP0 — CT16B1 Capture input 0. UCTS0 — USART0.Clear To Send input pin. |
| P1.9/CT16B1_PWM0/ USCLK0 | I/O | P1.9 — Port 1.9 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. CT16B1_PWM0 — CT16B1 PWM output 0. USCLK0 — USART0 Clock pin. |
| P1.10/URTS0/ CT16B1_PWM1/ | I/O | P1.10 — Port 1.10 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode.. CT16B1_PWM1 — PWM output 1 for CT16B1. URTS0 — USART0 Request To Send output pin. |
| P1.11/CLKOUT | I/O | P1.11 — Port 1.11 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. CLKOUT — Clockout pin. |
| P2.0~P2.9/AIN0~9 | I/O | P2.0~P2.9 — Port 2 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. AIN0~AIN9 — ADC channel input 0~9 pins. |
| P3.0/UDTR0/SEL1/ | I/O | P3.0 — Port 3.0 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. UDTR0 — USART0 Data Terminal Ready output pin. SEL1 — SSP1 Slave Select pin. |

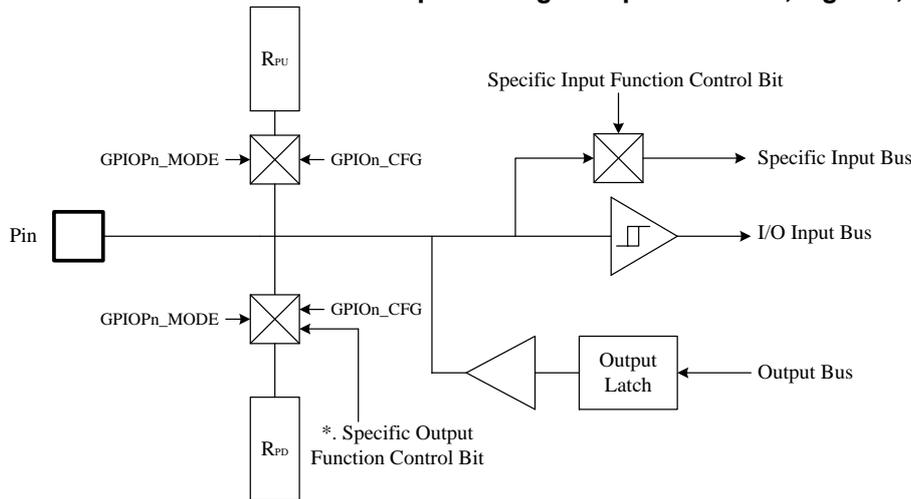
| | | |
|---------------------------|-----|--|
| P3.1/ <u>UDSR0</u> /SCK1 | I/O | P3.1 — Port 3.1 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. UDSR0 — USART0 Data Set Ready input pin. SCK1 — Serial clock pin for SSP1. |
| P3.2/ <u>UDCD0</u> /MISO1 | I/O | P3.2 — Port 3.2 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. UDCD0 — USART0 Data Carrier Detect input. MISO1 — SSP1 Master In Slave Out pin. |
| P3.3/ <u>URI0</u> /MOSI1 | I/O | P3.3 — Port 3.3 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. URI0 — USART0 Ring Indicator input pin. MOSI1 — Master Out Slave In for SSP1. |
| P3.4/LXIN | I/O | P3.4 — Port 3.4 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. LXIN — External low-speed X'tal input pin. |
| P3.5/LXOUT | I/O | P3.5 — Port 3.5 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. LXOUT — External low-speed X'tal output pin. |
| P3.6/ <u>RESET</u> | I/O | P3.6 — Port 3.6 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. RESET — External Reset input. Schmitt trigger structure, active "Low", normally stay "High". |
| P3.7/XIN | I/O | P3.7 — Port 3.7 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. XIN — External high-speed X'tal input pin. |
| P3.8/XOUT | I/O | P3.8 — Port 3.8 bi-direction pin. Schmitt trigger structure and built-in pull-up/pull-down resistors as input mode. XOUT — External high-speed X'tal output pin. |

1.6 PIN CIRCUIT DIAGRAMS

- Normal Bi-direction I/O Pin.

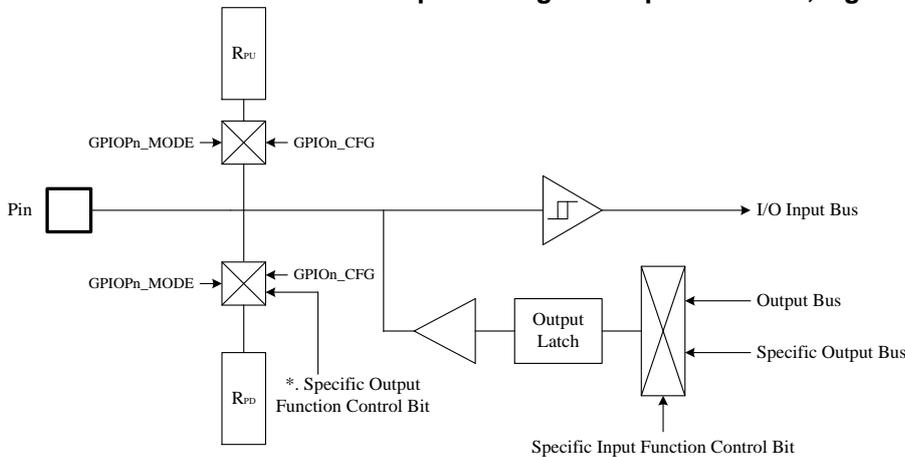


- Bi-direction I/O Pin Shared with Specific Digital Input Function, e.g. SPI, I2C...



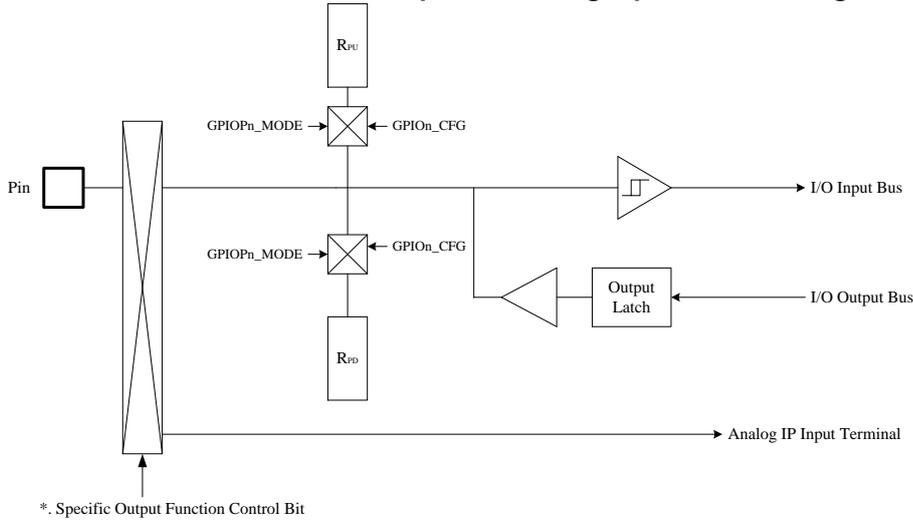
*. Some specific functions switch I/O direction directly, not through GPIO_n_MODE register.

- Bi-direction I/O Pin Shared with Specific Digital Output Function, e.g. SPI, I2C...



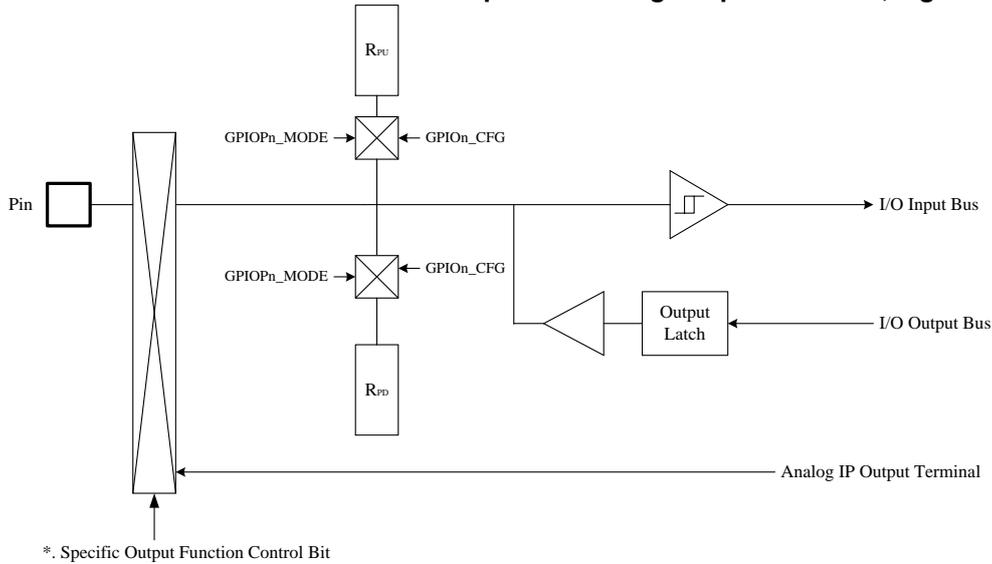
*. Some specific functions switch I/O direction directly, not through GPIO_n_MODE register.

● **Bi-direction I/O Pin Shared with Specific Analog Input Function, e.g. XIN, ADC...**



*. Some specific functions switch I/O direction directly, not through GPIO_n_MODE register.

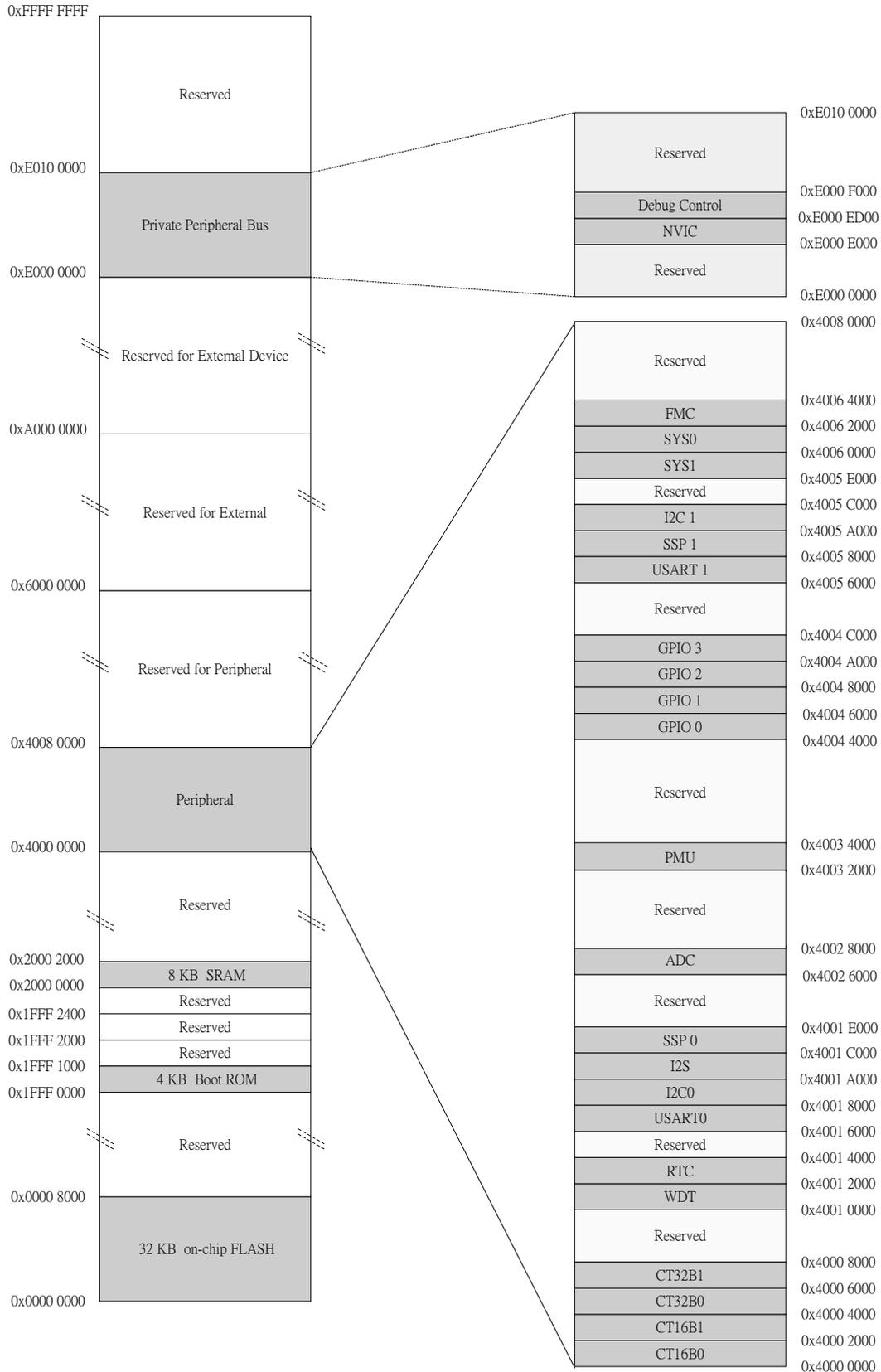
● **Bi-direction I/O Pin Shared with Specific Analog Output Function, e.g. XOUT...**



*. Some specific functions switch I/O direction directly, not through GPIO_n_MODE register.

2 CENTRAL PROCESSOR UNIT (CPU)

2.1 MEMORY MAP



2.2 SYSTEM TICK TIMER

The SysTick timer is an integral part of the Cortex-M0. The SysTick timer is intended to generate a fixed 10-ms 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.

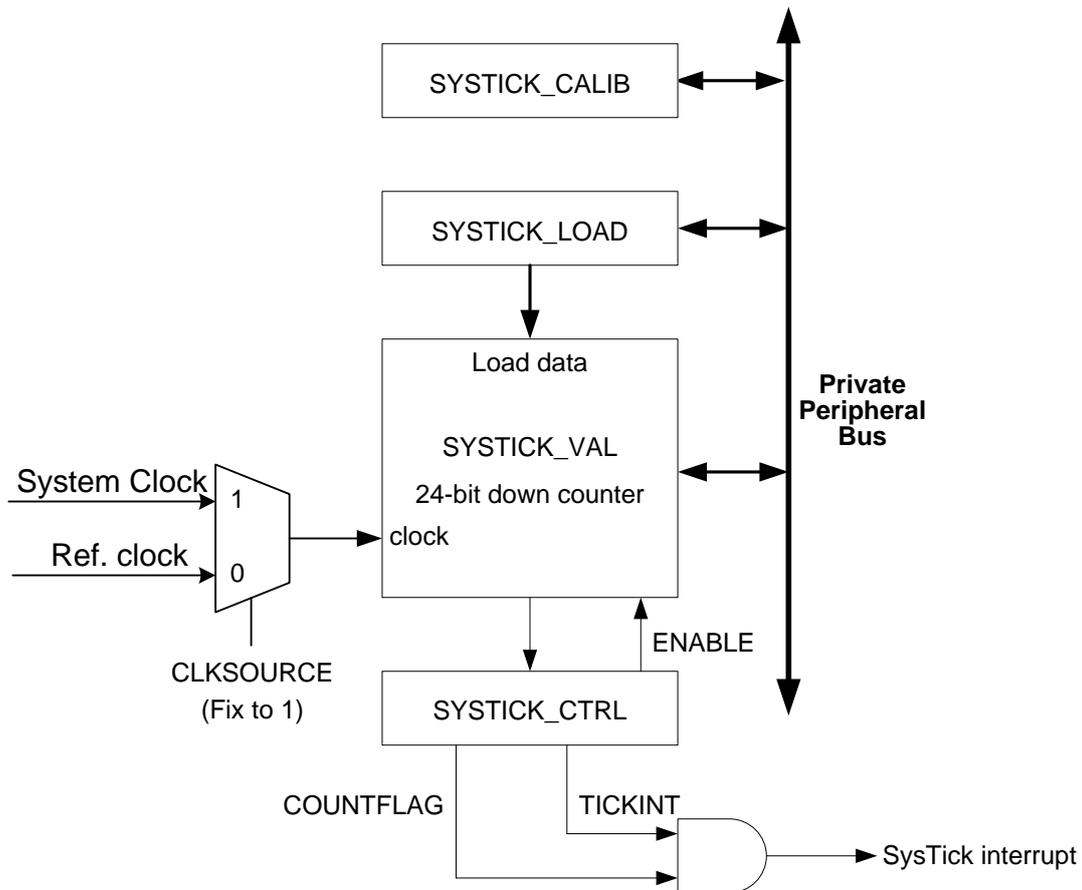
Refer to the *Cortex-M0 User Guide* for details.

2.2.1 OPERATION

The SysTick timer is a 24-bit timer that counts down to zero and generates an interrupt.

The intent is to provide a fixed 10-ms time interval between interrupts. The system tick timer is enabled through the SysTick control register. The system tick timer clock is fixed to the frequency of the system clock.

The block diagram of the SysTick timer:



When SysTick timer is enabled, the timer counts down from the current value (SYSTICK_VAL) to zero, reloads to the value in the SysTick Reload Value Register (SYSTICK_LOAD) on the next clock edge, then decrements on subsequent clocks. When the counter transitions to zero, the COUNTFLAG status bit is set to 1. The COUNTFLAG bit clears on reads.

*** Note: When the processor is halted for debugging the counter does not decrease.**

2.2.2 SYSTICK USAGE HINTS AND TIPS

The interrupt controller clock updates the SysTick counter. Some implementations stop this clock signal for low power mode. If this happens, the SysTick counter stops.

Ensure SW uses word accesses to access the SysTick registers.

The SysTick counter reload and current value are not initialized by HW. This means the correct initialization sequence for the SysTick counter is:

1. Program the reload value in SYSTICK_LOAD register.
2. Clear the current value by writing any value to SYSTICK_VAL register.
3. Program the Control and Status (SYSTICK_CTRL) register.

2.2.3 SYSTICK REGISTERS

2.2.3.1 System Tick Timer Control and Status register (SYSTICK_CTRL)

Address: 0xE000 E010 (Refer to Cortex-M0 Spec)

| Bit | Name | Description | Attribute | Reset |
|-------|-----------|---|-----------|-------|
| 31:17 | Reserved | | R | 0 |
| 16 | COUNTFLAG | This flag is set when the System Tick counter counts down to 0, and is cleared by reading this register. | R/W | 0 |
| 15:3 | Reserved | | R | 0 |
| 2 | CLKSOURCE | Selects the SysTick timer clock source. 0: reference clock. 1: system clock. (Fixed) | R | 1 |
| 1 | TICKINT | System Tick interrupt enable. 0: Disable the System Tick interrupt 1: Enable the System Tick interrupt, the interrupt is generated when the System Tick counter counts down to 0. | R/W | 0 |
| 0 | ENABLE | System Tick counter enable. 0: Disable 1: Enable | R/W | 0 |

2.2.3.2 System Tick Timer Reload value register (SYSTICK_LOAD)

Address: 0xE000 E014 (Refer to Cortex-M0 Spec)

The RELOAD register is set to the value that will be loaded into the SysTick timer whenever it counts down to zero. This register is set by software as part of timer initialization. The SYSTICK_CALIB register may be read and used as the value for RELOAD if the CPU or external clock is running at the frequency intended for use with the SYSTICK_CALIB value.

The following example illustrates selecting the SysTick timer reload value to obtain a 10 ms time interval with the system clock set to 50 MHz.

The SysTick clock = system clock = 50 MHz

$$\text{RELOAD} = (\text{system tick clock frequency} \times 10 \text{ ms}) - 1 = (50 \text{ MHz} \times 10 \text{ ms}) - 1 = 0x0007A11F.$$

| Bit | Name | Description | Attribute | Reset |
|-------|----------|-------------|-----------|-------|
| 31:24 | Reserved | | R | 0 |

| | | | | |
|-------------|--------|---|-----|----------|
| 23:0 | RELOAD | Value to load into the SYSTICK_VAL when the counter is enabled and when it reaches 0. | R/W | 0x5F7F9B |
|-------------|--------|---|-----|----------|

2.2.3.3 System Tick Timer Current Value register (SYSTICK_VAL)

Address: 0xE000 E018 (Refer to Cortex-M0 Spec)

| Bit | Name | Description | Attribute | Reset |
|--------------|----------|---|-----------|----------|
| 31:24 | Reserved | | R | 0 |
| 23:0 | CURRENT | Reading this register returns the current value of the System Tick counter. Writing any value clears the System Tick counter and the COUNTFLAG bit in SYSTICK_CTRL. | R/W | 0x7E7F35 |

2.2.3.4 System Tick Timer Calibration Value register (SYSTICK_CALIB)

Address: 0xE000 E01C (Refer to Cortex-M0 Spec)

| Bit | Name | Description | Attribute | Reset |
|--------------|----------|--|-----------|---------|
| 31 | NOREF | Indicates the reference clock to M0 is provided or not. 1: No reference clock provided. | R | 1 |
| 30 | SKEW | Indicates whether the TENMS value is exact, an inexact TENMS value can affect the suitability of SysTick as a software real time clock. 0: TENMS value is exact 1: TENMS value is inexact, or not given. | R | 0 |
| 29:24 | Reserved | | R | 0 |
| 23:0 | TENMS | Reload value for 10ms timing, subject to system clock skew errors. If the value reads as zero, the calibration value is not known. | R/W | 0xA71FF |

2.3 NESTED VECTORED INTERRUPT CONTROLLER (NVIC)

All interrupts including the core exceptions are managed by the NVIC. NVIC has the following Features:

- The NVIC supports 32 vectored interrupts.
- 4 programmable interrupt priority levels with hardware priority level masking.
- Low-latency exception and interrupt handling.
- Efficient processing of late arriving interrupts.
- Implementation of System Control Registers
- Software interrupt generation.

2.3.1 INTERRUPT AND EXCEPTION VECTORS

| Execution No. | Priority | Function | Description | Address Offset |
|---------------|----------|-------------------|---------------------------------|----------------|
| 0 | - | - | Reserved | 0x0000 0000 |
| 1 | -3 | Reset | Reset | 0x0000 0004 |
| 2 | -2 | NMI_Handler | Non maskable interrupt. | 0x0000 0008 |
| 3 | -1 | HardFault_Handler | All class of fault | 0x0000 000C |
| 4~10 | Reserved | Reserved | Reserved | - |
| 11 | Settable | SVCCall | | 0x0000 002C |
| 12~13 | Reserved | Reserved | Reserved | - |
| 14 | Settable | PendSV | | 0x0000 0038 |
| 15 | Settable | SysTick | | 0x0000 003C |
| 16 | Settable | IRQ0/WAKEIRQ | Wakeup interrupt | 0x0000 0040 |
| 17 | Settable | IRQ1/ | | 0x0000 0044 |
| 18 | Settable | IRQ2/ | | 0x0000 0048 |
| 19 | Settable | IRQ3/ | | 0x0000 004C |
| 20 | Settable | IRQ4/ | | 0x0000 0050 |
| 21 | Settable | IRQ5/ | | 0x0000 0054 |
| 22 | Settable | IRQ6/ | | 0x0000 0058 |
| 23 | Settable | IRQ7/ | | 0x0000 005C |
| 24 | Settable | IRQ8/ | | 0x0000 0060 |
| 25 | Settable | IRQ9/ | | 0x0000 0064 |
| 26 | Settable | IRQ10/ | | 0x0000 0068 |
| 27 | Settable | IRQ11/ | | 0x0000 006C |
| 28 | Settable | IRQ12/ | | 0x0000 0070 |
| 29 | Settable | IRQ13/SSP0IRQ | SSP0 | 0x0000 0074 |
| 30 | Settable | IRQ14/SSP1IRQ | SSP1 | 0x0000 0078 |
| 31 | Settable | IRQ15/I2C0IRQ | I2C0 | 0x0000 007C |
| 32 | Settable | IRQ16/CT16B0IRQ | CT16B0 | 0x0000 0080 |
| 33 | Settable | IRQ17/CT16B1IRQ | CT16B1 | 0x0000 0084 |
| 34 | Settable | IRQ18/CT32B0IRQ | CT32B0 | 0x0000 0088 |
| 35 | Settable | IRQ19/CT32B1IRQ | CT32B1 | 0x0000 008C |
| 36 | Settable | IRQ20/I2SIRQ | I2S | 0x0000 0090 |
| 37 | Settable | IRQ21/USART0IRQ | USART0 | 0x0000 0094 |
| 38 | Settable | IRQ22/USART1IRQ | USART1 | 0x0000 0098 |
| 39 | Settable | IRQ23/I2C1IRQ | I2C1 | 0x0000 009C |
| 40 | Settable | IRQ24/ADCIRQ | ADC | 0x0000 00A0 |
| 41 | Settable | IRQ25/WDTIRQ | WDT | 0x0000 00A4 |
| 42 | Settable | IRQ26/LVDIRQ | LVD | 0x0000 00A8 |
| 43 | Settable | IRQ27/RTCIRQ | RTC | 0x0000 00AC |
| 44 | Settable | IRQ28/P3IRQ | GPIO interrupt status of port 3 | 0x0000 00B0 |
| 45 | Settable | IRQ29/P2IRQ | GPIO interrupt status of port 2 | 0x0000 00B4 |
| 46 | Settable | IRQ30/P1IRQ | GPIO interrupt status of port 1 | 0x0000 00B8 |
| 47 | Settable | IRQ31/P0IRQ | GPIO interrupt status of port 0 | 0x0000 00BC |

2.3.2 NVIC REGISTERS

2.3.2.1 IRQ0~31 Interrupt Set-Enable Register (NVIC_ISER)

Address: 0xE000 E100 (Refer to Cortex-M0 Spec.)

The ISER enables interrupts, and shows the interrupts that are enabled.

| Bit | Name | Description | Attribute | Reset |
|------|--------------|---|-----------|-------|
| 31:0 | SETENA[31:0] | Interrupt set-enable bits. Write→ 0: No effect 1: Enable interrupt. Read→ 0: Interrupt disabled 1: Interrupt enabled. | R/W | 0 |

2.3.2.2 IRQ0~31 Interrupt Clear-Enable Register (NVIC_ICER)

Address: 0xE000 E180 (Refer to Cortex-M0 Spec.)

The ICER disables interrupts, and shows the interrupts that are enabled.

| Bit | Name | Description | Attribute | Reset |
|------|--------------|--|-----------|-------|
| 31:0 | CLRENA[31:0] | Interrupt clear-enable bits. Write→ 0: No effect 1: Disable interrupt. Read→ 0: Interrupt disabled 1: Interrupt enabled. | R/W | 0 |

2.3.2.3 IRQ0~31 Interrupt Set-Pending Register (NVIC_ISPR)

Address: 0xE000 E200 (Refer to Cortex-M0 Spec.)

The ISPR forces interrupts into the pending state, and shows the interrupts that are pending.

Note: Writing 1 to the ISPR bit corresponding to

- an interrupt that is pending has no effect
- a disabled interrupt sets the state of that interrupt to pending.

| Bit | Name | Description | Attribute | Reset |
|------|---------------|--|-----------|-------|
| 31:0 | SETPEND[31:0] | Interrupt set-pending bits. Write→ 0: No effect 1: Change interrupt state to pending Read→ 0: Interrupt is not pending 1: Interrupt is pending | R/W | 0 |

2.3.2.4 IRQ0~31 Interrupt Clear-Pending Register (NVIC_ICPR)

Address: 0xE000 E280 (Refer to Cortex-M0 Spec.)

The ICPR removes the pending state from interrupts, and shows the interrupts that are pending.

Note: Writing 1 to an ICPR bit does not affect the active state of the corresponding interrupt.

| Bit | Name | Description | Attribute | Reset |
|------|---------------|--|-----------|-------|
| 31:0 | CLRPEND[31:0] | Interrupt clear-pending bits. Write→ 0: No effect 1: Removes pending state of an interrupt Read→ 0: Interrupt is not pending 1: Interrupt is pending | R/W | 0 |

2.3.2.5 IRQ0~31 Interrupt Priority Register (NVIC_IPRn) (n=0~7)

Address: 0xE000 E400 + 0x4 * n (Refer to Cortex-M0 Spec.)

The interrupt priority registers provide an 8-bit priority field for each interrupt, and each register holds four priority fields. This means the number of registers is implementation-defined, and corresponds to the number of implemented interrupts.

| Bit | Name | Description | Attribute | Reset |
|-------|-------------|---|-----------|-------|
| 31:24 | PRI_(4*n+3) | Each priority field holds a priority value, 0-192. The lower the value, the greater the priority of the corresponding interrupt. The processor implements only bits[31:30] of each field, bits [29:24] read as zero and ignore writes. This means writing 255 to a priority register saves value 192 to the register. | R/W | 0 |
| 23:16 | PRI_(4*n+2) | Each priority field holds a priority value, 0-192. The lower the value, the greater the priority of the corresponding interrupt. The processor implements only bits[23:22] of each field, bits [21:16] read as zero and ignore writes. This means writing 255 to a priority register saves value 192 to the register. | R/W | 0 |
| 15:8 | PRI_(4*n+1) | Each priority field holds a priority value, 0-192. The lower the value, the greater the priority of the corresponding interrupt. The processor implements only bits[15:14] of each field, bits [13:8] read as zero and ignore writes. This means writing 255 to a priority register saves value 192 to the register. | R/W | 0 |
| 7:0 | PRI_4*n | Each priority field holds a priority value, 0-192. The lower the value, the greater the priority of the corresponding interrupt. The processor implements only bits[7:6] of each field, bits [5:0] read as zero and ignore writes. This means writing 255 to a priority register saves value 192 to the register. | R/W | 0 |

2.4 APPLICATION INTERRUPT AND RESET CONTROL (AIRC)

Address: 0xE000 ED0C (Refer to Cortex-M0 Spec)

The entire MCU, including the core, can be reset by SW by setting the SYSRESREQ bit in the AIRC register in Cortex-M0 spec.

*** Note: To write to this register, user must write 0x05FA to the VECTKEY field at the same time, otherwise the processor ignores the write.**

| Bit | Name | Description | Attribute | Reset |
|-------|---------------|---|-----------|-------|
| 31:16 | VECTKEY | Register key. Read as unknown. Write 0x05FA to VECTKEY, otherwise the write is ignored. | R/W | 0 |
| 15 | ENDIANESS | Data endianness implemented 0: Little-endian 1: Big-endian | R | 0 |
| 14:3 | Reserved | | R | 0 |
| 2 | SYSRESETREQ | System reset request. This bit read as 0. 0: No effect 1: Requests a system level reset. | W | 0 |
| 1 | VECTCLRACTIVE | Reserved for debug use. This bit read as 0. When writing to the register you must write 0 to this bit, otherwise behavior is Unpredictable. | W | 0 |
| 0 | Reserved | | R | 0 |

2.5 CODE OPTION TABLE

Address: 0x1FFF 2000

| Bit | Name | Description | Attribute | Reset |
|-------|---------------------|--|-----------|--------|
| 31:16 | Code Security[15:0] | Code Security 0xFFFF: CS0 0x5A5A: CS1 0xA5A5: CS2 | R/W | 0xFFFF |
| 15:1 | Reserved | | R | 0 |
| 0 | Reserved | | R/W | 1 |

2.6 UNIQUE NUMBER

The unique number is a 8-byte unique device serial number of each IC. In other words, the unique number is different and discontinuous for each IC. Users can use the unique number to pair in RF application, or use as USB string serial number.

Address: 0x1FFF 221C

| Bit | Name | Description | Attribute | Reset |
|------|--------------|--------------------------------|-----------|--------|
| 31:0 | L4BYTE[31:0] | Lower 4 bytes of Unique number | R | By Die |

Address: 0x1FFF 2220

| Bit | Name | Description | Attribute | Reset |
|------|--------------|-------------------------------|-----------|--------|
| 31:0 | H4BYTE[31:0] | High 4 bytes of Unique number | R | By Die |

3 SYSTEM CONTROL

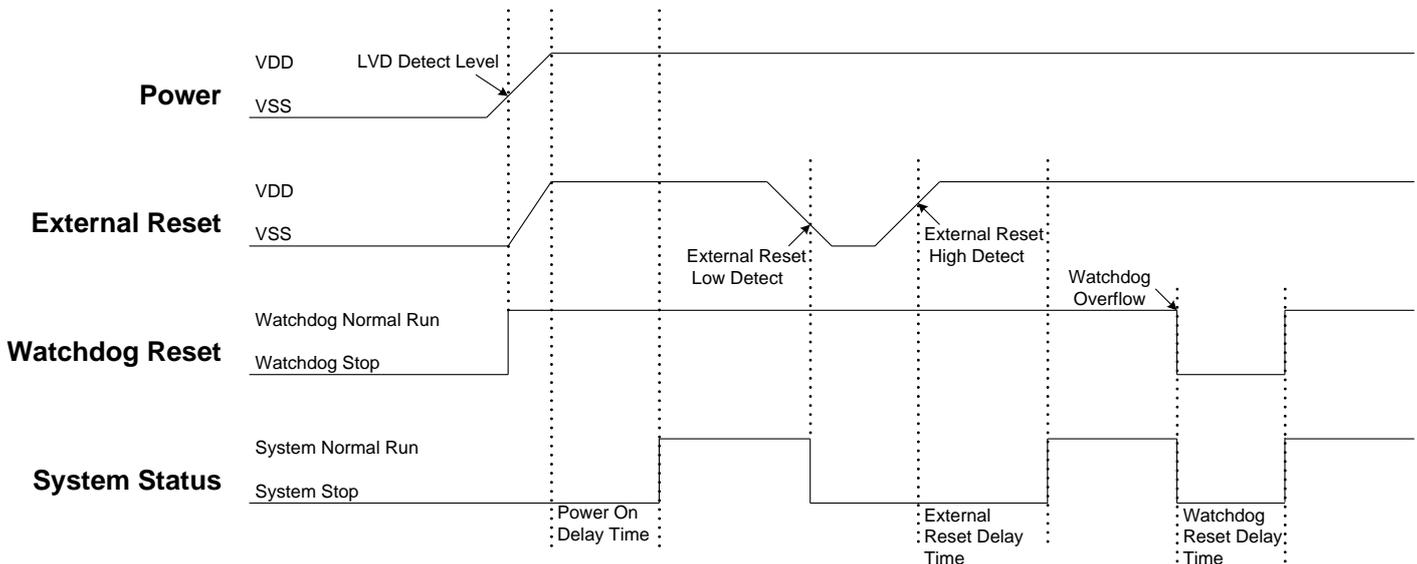
3.1 RESET

A system reset is generated when one of the following events occurs:

1. A low level on the RST pin (external reset).
2. Power-on reset (POR reset)
3. LVD reset
4. Watchdog Timer reset (WDT reset)
5. Software reset (SW reset)
6. DPDWAKEUP reset when exiting Deep power-down mode by DPDWAKEUP pin

The reset source can be identified by checking the reset flags in [System Reset Status register \(SYS0_RSTST\)](#). These sources act on the RST pin and it is always kept low during the delay phase. The RESET service routine vector is fixed at address 0x00000004 in the memory map. For more details, refer to [Interrupt and Exception Vectors](#).

Finishing any reset sequence needs some time. The system provides complete procedures to make the power on reset successful. For different oscillator types, the reset time is different. That causes the VDD rise rate and start-up time of different oscillator is not fixed. RC type oscillator's start-up time is very short, but the crystal type is longer. Under client terminal application, users have to take care of the power on reset time for the master terminal requirement. The reset timing diagram is as following.



3.1.1 POWER-ON RESET (POR)

The power on reset depends on LVD operation for most power-up situations. The power supplying to system is a rising curve and needs some time to achieve the normal voltage. Power on reset sequence is as following:

- **Power-up:** System detects the power voltage up and waits for power stable.
- **External reset (only external reset pin enable):** System checks external reset pin status. If external reset pin is not high level, the system keeps reset status and waits external reset pin released.
- **System initialization:** All system registers is set as initial conditions and system is ready.
- **Oscillator warm up:** Oscillator operation is successfully and supply to system clock.
- **Program executing:** Power on sequence is finished and program executes from Boot loader.

3.1.2 WATCHDOG RESET (WDT RESET)

Watchdog reset is a system protection. In normal condition, system works well and clears watchdog timer by program. Under error condition, system is in unknown situation and watchdog can't be clear by program before watchdog timer overflow. Watchdog timer overflow occurs and the system is reset. After watchdog reset, the system restarts and returns normal mode. Watchdog reset sequence is as following.

- **Watchdog timer status:** System checks watchdog timer overflow status. If watchdog timer overflow occurs, the system is reset.
- **System initialization:** All system registers is set as initial conditions and system is ready.
- **Oscillator warm up:** Oscillator operation is successfully and supply to system clock.
- **Program executing:** Power on sequence is finished and program executes from 0x0.

Watchdog timer application note is as following.

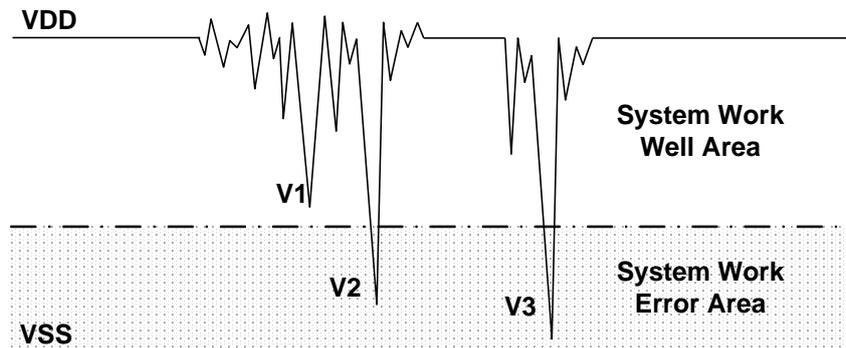
- Before clearing watchdog timer, check I/O status and check RAM contents can improve system error.
- Don't clear watchdog timer in interrupt vector and interrupt service routine. That can improve main routine fail.
- Clearing watchdog timer program is only at one part of the program. This way is the best structure to enhance the watchdog timer function.

* **Note:** Please refer to the "WATCHDOG TIMER" about watchdog timer detail information.

3.1.3 BROWN-OUT RESET

3.1.3.1 BROWN OUT DESCRIPTION

The brown-out reset is a power dropping condition. The power drops from normal voltage to low voltage by external factors (e.g. EFT interference or external loading changed). The brown out reset would make the system not work well or executing program error.



Brown-Out Reset Diagram

The power dropping might through the voltage range that's the system dead-band. The dead-band means the power range can't offer the system minimum operation power requirement. The above diagram is a typical brown out reset diagram. There is a serious noise under the VDD, and VDD voltage drops very deep. There is a dotted line to separate the system working area. The above area is the system work well area. The below area is the system work error area called dead-band. V1 doesn't touch the below area and not affect the system operation. But the V2 and V3 is under the below area and may induce the system error occurrence. Let system under dead-band includes some conditions.

DC application:

The power source of DC application is usually using battery. When low battery condition and MCU drive any loading, the power drops and keeps in dead-band. Under the situation, the power won't drop deeper and not touch the system reset voltage. That makes the system under dead-band.

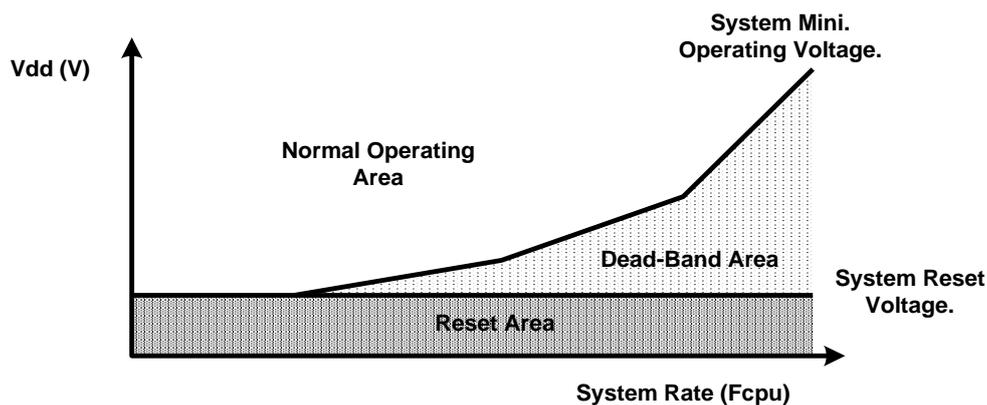
AC application:

In AC power application, the DC power is regulated from AC power source. This kind of power usually couples with AC noise that makes the DC power dirty. Or the external loading is very heavy, e.g. driving motor. The loading operating induces noise and overlaps with the DC power. VDD drops by the noise, and the system works under unstable power situation.

The power on duration and power down duration are longer in AC application. The system power on sequence protects the power on successful, but the power down situation is like DC low battery condition. When turn off the AC power, the VDD drops slowly and through the dead-band for a while.

3.1.3.2 THE SYSTEM OPERATING VOLTAGE DECSRIPTION

To improve the brown out reset needs to know the system minimum operating voltage which is depend on the system executing rate and power level. Different system executing rates have different system minimum operating voltage. The electrical characteristic section shows the system voltage to executing rate relationship.



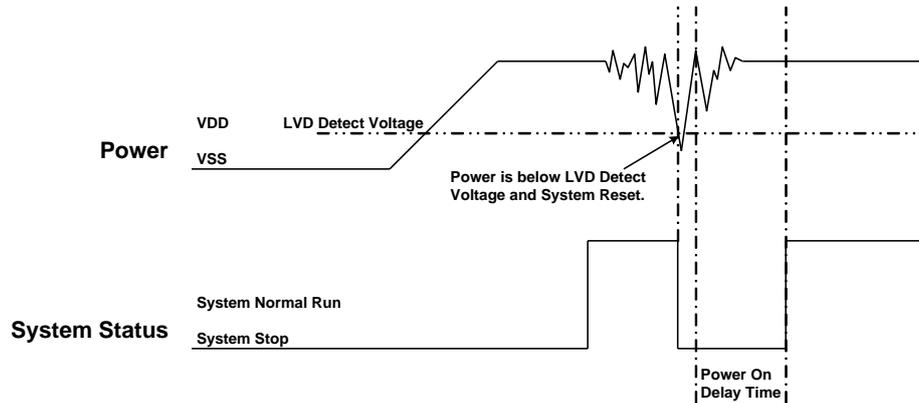
Normally the system operation voltage area is higher than the system reset voltage to VDD, and the reset voltage is decided by LVD detect level. The system minimum operating voltage rises when the system executing rate upper even higher than system reset voltage. The dead-band definition is the system minimum operating voltage above the system reset voltage.

3.1.3.3 BROWN-OUT RESET IMPROVEMENT

How to improve the brown reset condition? There are some methods to improve brown out reset as following.

- LVD reset
- Watchdog reset
- Reduce the system executing rate
- External reset circuit. (Zener diode reset circuit, Voltage bias reset circuit, External reset IC)

* **Note:** The “Zener diode reset circuit”, “Voltage bias reset circuit” and “External reset IC” can completely improve the brown out reset, DC low battery and AC slow power down conditions.

LVD reset:

The LVD (low voltage detector) is built-in SONiX 32-bit MCU to be brown out reset protection. When the VDD drops and is below LVD detect voltage, the LVD asserts an interrupt signal to the NVIC. This signal can be enabled for interrupt in the Interrupt Enable Register in the NVIC in order to cause a CPU interrupt; if not, SW can monitor the signal by reading a dedicated status register. An additional threshold level can be selected to cause a forced reset of the chip. The LVD detect level is different by each MCU. The LVD voltage level is a point of voltage and not easy to cover all dead-band range. Using LVD to improve brown out reset is dependent on application requirement and environment. If the power variation is very deep, violent and trigger the LVD, the LVD can be the protection. If the power variation can touch the LVD detect level and make system work error, the LVD can't be the protection and need to other reset methods. More detail LVD information is in the electrical characteristic section.

Watchdog reset:

The watchdog timer is a protection to make sure the system executes well. Normally the watchdog timer would be clear at one point of program. Don't clear the watchdog timer in several addresses. The system executes normally and the watchdog won't reset system. When the system is under dead-band and the execution error, the watchdog timer can't be clear by program. The watchdog is continuously counting until overflow occurrence. The overflow signal of watchdog timer triggers the system to reset and return to normal mode after reset sequence. This method also can improve brown out reset condition and make sure the system to return normal mode.

If the system reset by watchdog and the power is still in dead-band, the system reset sequence won't be successful and the system stays in reset status until the power return to normal range.

Reduce the system executing rate:

If the system rate is fast and the dead-band exists, to reduce the system executing rate can improve the dead-band. The lower system rate is with lower minimum operating voltage. Select the power voltage that's no dead-band issue and find out the mapping system rate. Adjust the system rate to the value and the system exits the dead-band issue. This way needs to modify whole program timing to fit the application requirement.

External reset circuit:

The external reset methods also can improve brown out reset and is the complete solution. There are three external reset circuits to improve brown out reset including "Zener diode reset circuit", "Voltage bias reset circuit" and "External reset IC". These three reset structures use external reset signal and control to make sure the MCU be reset under power dropping and under dead-band. The external reset information is described in the next section.

3.1.4 EXTERNAL RESET

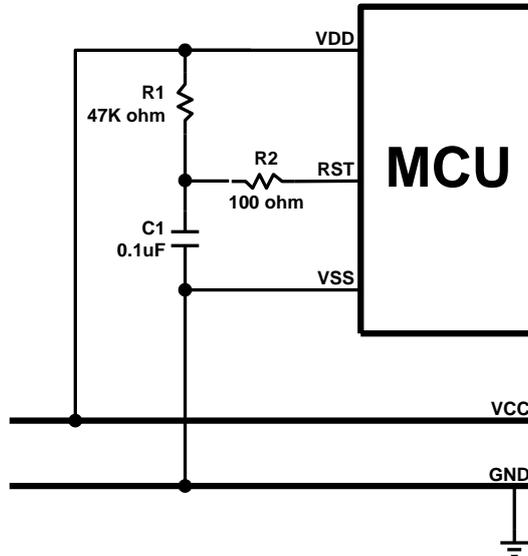
External reset function is controlled by [External RESET pin control \(SYS0_EXRSTCTRL\)](#) register. Default value is 1, which means external reset function is enabled. External reset pin is Schmitt Trigger structure and low level active. The system is running when reset pin is high level voltage input. The reset pin receives the low voltage and the system is reset. The external reset operation activates in power on and normal running mode. During system power-up, the external reset pin must be high level input, or the system keeps in reset status. External reset sequence is as following.

- **External reset (only external reset pin enable):** System checks external reset pin status. If external reset pin is not high level, the system keeps reset status and waits external reset pin released.
- **System initialization:** All system registers is set as initial conditions and system is ready.

- **Oscillator warm up:** Oscillator operation is successfully and supply to system clock.
- **Program executing:** Power on sequence is finished and program executes from Boot loader.

The external reset can reset the system during power on duration, and good external reset circuit can protect the system to avoid working at unusual power condition, e.g. brown out reset in AC power application.

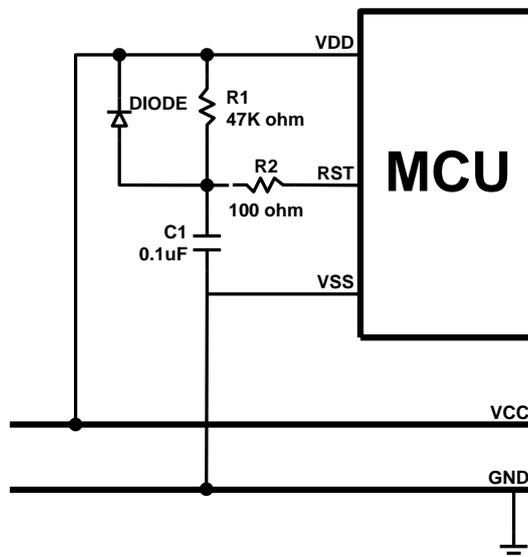
3.1.4.1 SIMPLY RC RESET CIRCUIT



This is the basic reset circuit, and only includes R1 and C1. The RC circuit operation makes a slow rising signal into reset pin as power up. The reset signal is slower than VDD power up timing, and system occurs a power on signal from the timing difference.

* **Note:** The reset circuit is no any protection against unusual power or brown out reset.

3.1.4.2 DIODE & RC RESET CIRCUIT

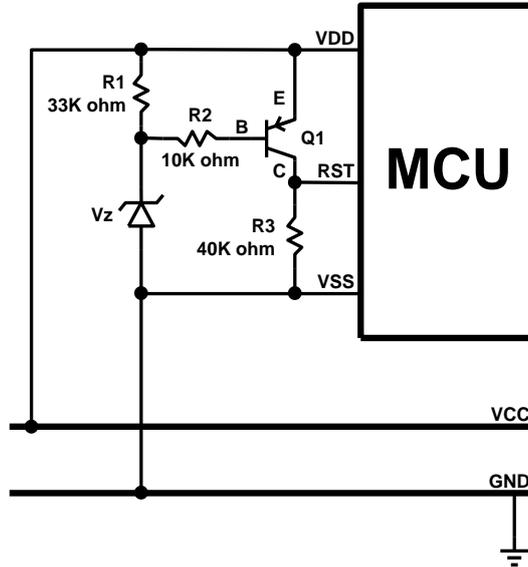


This is the better reset circuit. The R1 and C1 circuit operation is like the simply reset circuit to make a power on signal. The reset circuit has a simply protection against unusual power. The diode offers a power positive path to conduct higher power to VDD. It is can make reset pin voltage level to synchronize with VDD voltage. The structure can

improve slight brown out reset condition.

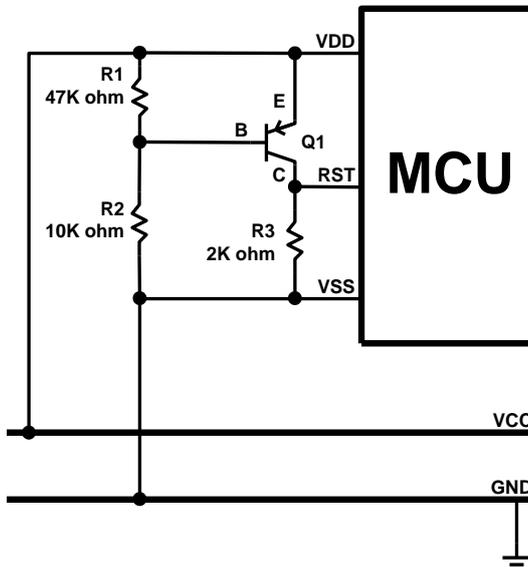
* **Note:** The R2 100 ohm resistor of “Simply reset circuit” and “Diode & RC reset circuit” is necessary to limit any current flowing into reset pin from external capacitor C in the event of reset pin breakdown due to Electrostatic Discharge (ESD) or Electrical Over-stress (EOS).

3.1.4.3 ZENER DIODE RESET CIRCUIT



The Zener diode reset circuit is a simple low voltage detector and can **improve brown out reset condition completely**. Use Zener voltage to be the active level. When VDD voltage level is above “ $V_z + 0.7V$ ”, the C terminal of the PNP transistor outputs high voltage and MCU operates normally. When VDD is below “ $V_z + 0.7V$ ”, the C terminal of the PNP transistor outputs low voltage and MCU is in reset mode. Decide the reset detect voltage by Zener specification. Select the right Zener voltage to conform the application.

3.1.4.4 VOLTAGE BIAS RESET CIRCUIT



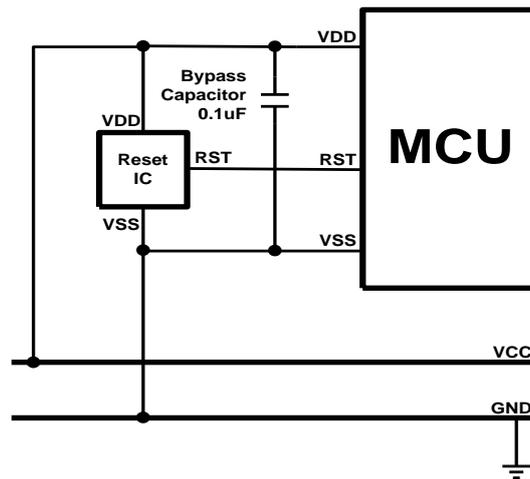
The voltage bias reset circuit is a low cost voltage detector and can **improve brown out reset condition completely**. The operating voltage is not accurate as Zener diode reset circuit. Use R1, R2 bias voltage to be the active level. When VDD voltage level is above or equal to “ $0.7V \times (R1 + R2) / R1$ ”, the C terminal of the PNP transistor outputs high

voltage and MCU operates normally. When VDD is below $0.7V \times (R1 + R2) / R1$, the C terminal of the PNP transistor outputs low voltage and MCU is in reset mode.

Decide the reset detect voltage by R1, R2 resistances. Select the right R1, R2 value to conform the application. In the circuit diagram condition, the MCU's reset pin level varies with VDD voltage variation, and the differential voltage is 0.7V. If the VDD drops and the voltage lower than reset pin detect level, the system would be reset. If want to make the reset active earlier, set the $R2 > R1$ and the cap between VDD and C terminal voltage is larger than 0.7V. The external reset circuit is with a stable current through R1 and R2. For power consumption issue application, e.g. DC power system, the current must be considered to whole system power consumption.

* **Note:** Under unstable power condition as brown out reset, “Zener diode reset circuit” and “Voltage bias reset circuit” can protect system no any error occurrence as power dropping. When power drops below the reset detect voltage, the system reset would be triggered, and then system executes reset sequence. That makes sure the system work well under unstable power situation.

3.1.4.5 EXTERNAL RESET IC



The external reset circuit also uses external reset IC to enhance MCU reset performance. This is a high cost and good effect solution. By different application and system requirement to select suitable reset IC. The reset circuit can improve all power variation.

3.1.5 SOFTWARE RESET

The entire MCU, including the core, can be reset by software by setting the SYSRESREQ bit in the [AIRC \(Application Interrupt and Reset Control\)](#) register in Cortex-M0 spec.

The software-initiated system reset sequence is as follows:

1. A software reset is initiated by setting the SYSRESREQ bit.
2. An internal reset is asserted.
3. The internal reset is deasserted and the MCU loads from memory the initial stack pointer, the initial program counter, and the first instruction designated by the program counter, and then begins execution.

3.2 SYSTEM CLOCK

Different clock sources can be used to drive the system clock (SYSCLK):

- 12 MHz internal high speed RC (IHRC)
- 16 KHz internal low speed RC (ILRC)
- PLL clock
- High speed external (EHS) crystal clock
- Low speed external (ELS) 32.768 KHz crystal

Each clock source can be switched on or off independently when it is not used, to optimize power consumption.

The micro-controller is a dual clock system. There are high-speed clock and low-speed clock. The high-speed clock is generated from the external oscillator & on-chip PLL circuit. The low-speed clock is generated from on-chip low-speed RC oscillator circuit (ILRC 16 KHz).

3.2.1 INTERNAL RC CLOCK SOURCE

3.2.1.1 Internal High-speed RC Oscillator (IHRC)

The internal high-speed oscillator is 12MHz RC type. The accuracy is $\pm 2\%$ under commercial condition. The IHRC can be switched on and off using the IHRcen bit in [Analog Block Control register \(SYS0_ANBCTRL\)](#).

3.2.1.2 Internal Low-speed RC Oscillator (ILRC)

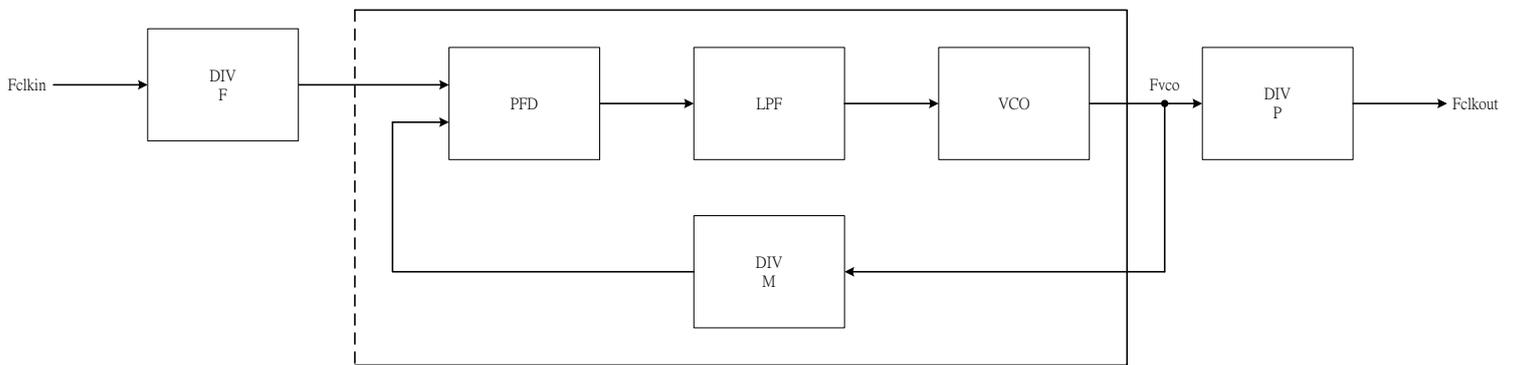
The system low clock source is the internal low-speed oscillator built in the micro-controller. The low-speed oscillator uses RC type oscillator circuit. The frequency is affected by the voltage and temperature of the system. In common condition, the frequency of the RC oscillator is about 16 KHz.

*** Note: The ILRC can ONLY be switched on and off by HW.**

3.2.2 PLL

SONiX 32-bit MCU uses the PLL to create the clocks for the core and peripherals. The input frequency range is 10MHz to 25MHz. The input clock is divided down and fed to the Phase-Frequency Detector (PFD). This block compares the phase and frequency of its inputs, and generates a control signal when phase and/or frequency do not match. The loop filter filters these control signals and drives the voltage controlled oscillator (VCO), which generates the main clock and optionally two additional phases. The VCO frequency range is 156MHz to 320MHz. These clocks are divided by P by the programmable post divider to create the output clock(s). The VCO output clock is then divided by M by the programmable feedback divider to generate the feedback clock. The output signal of the phase-frequency detector is also monitored by the lock detector, to signal when the PLL has locked on to the input clock.

The PLL settling time is 100 μ s.



3.2.2.1 PLL Frequency selection

The PLL frequency equations:

$$F_{VCO} = F_{CLKIN} / F * M$$

$$F_{CLKOUT} = F_{VCO} / P$$

The PLL frequency is determined by the following parameters:

- F_{CLKIN} : Frequency from the PLLCLKSEL multiplexer.
- F_{VCO} : Frequency of the Voltage Controlled Oscillator (VCO); 156 to 320 MHz.
- F_{CLKOUT} : Frequency of PLL output.
- P: System PLL post divider ratio, controlled by PSEL bits in [PLL control register \(SYS0_PLLCTRL\)](#).
- F: System PLL front divider ratio, controlled by FSEL bits in [PLL control register \(SYS0_PLLCTRL\)](#).
- M: System PLL feedback divider ratio, controlled by MSEL bits in [PLL control register \(SYS0_PLLCTRL\)](#).

To select the appropriate values for M, P, and F, it is recommended to follow these constraints:

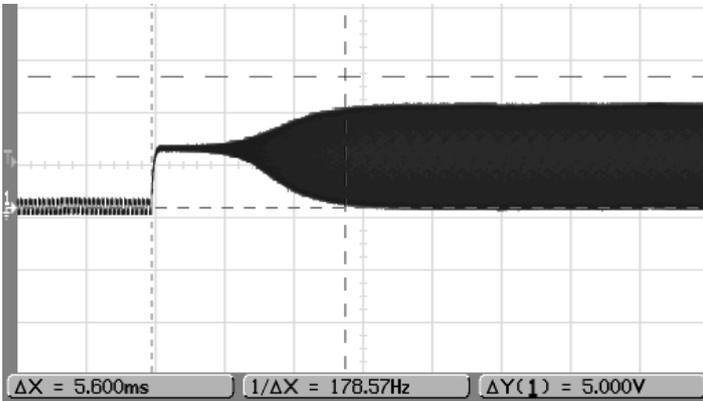
1. $10\text{MHz} \leq F_{CLKIN} \leq 25\text{MHz}$
2. $150\text{MHz} \leq F_{VCO} \leq 330\text{MHz}$
3. $2 < M \leq 31$
4. $F = 1$, or 2
5. $P = 6, 8, 10, 12$, or 14 (duty 50% +/- 2.5%)
6. $F_{CLKOUT} = 20\text{MHz}, 30\text{MHz}, 40\text{MHz}, 50\text{MHz}, 24\text{MHz}, 36\text{MHz}, 48\text{MHz}, 32\text{MHz}, 22\text{MHz}, 24\text{MHz}, 50\text{MHz}$
with jitter < ± 500 ps

3.2.3 EXTERNAL CLOCK SOURCE

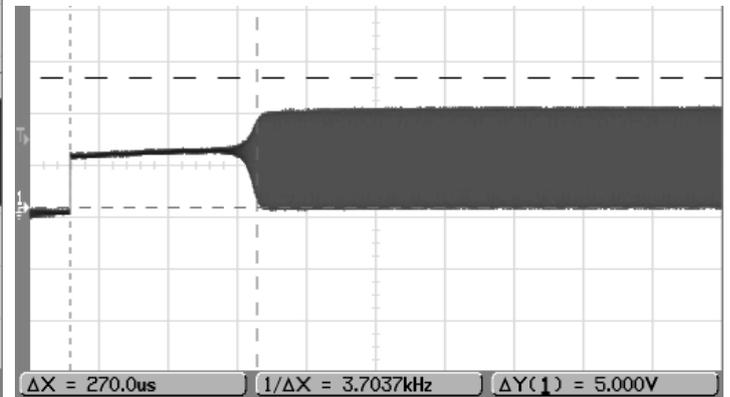
3.2.3.1 External High-speed (EHS) Clock

External high clock includes Crystal/Ceramic modules. The startup time of is longer. The oscillator start-up time decides reset time length.

4MHz Crystal

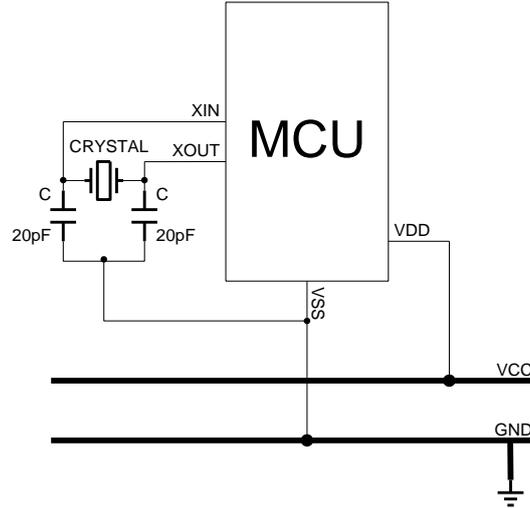


4MHz Ceramic



3.2.3.2 CRYSTAL/CERAMIC

Crystal/Ceramic devices are driven by XIN, XOUT pins. For high/normal/low frequency, the driving currents are different.



* **Note:** Connect the Crystal/Ceramic and C as near as possible to the XIN/XOUT/VSS pins of MCU.

- **Structure:** 1MHz~25MHz EHS external crystal/ceramic resonator
- **Main Purpose:** System high clock source, RTC clock source, and PLL clock source.
- **Warm-up Time:** 2048*F_{EHS}
- **XIN/XOUT Shared Pin Selection:**

| Oscillator Mode | XIN pin | XOUT pin |
|-----------------|-----------------|-----------------|
| IHRC | GPIO | GPIO |
| EHS X'TAL | Crystal/Ceramic | Crystal/Ceramic |

The resonator and the load capacitors have to be placed as close as possible to the oscillator pins in order to minimize output distortion and startup stabilization time. The loading capacitance values must be adjusted according to the selected oscillator.

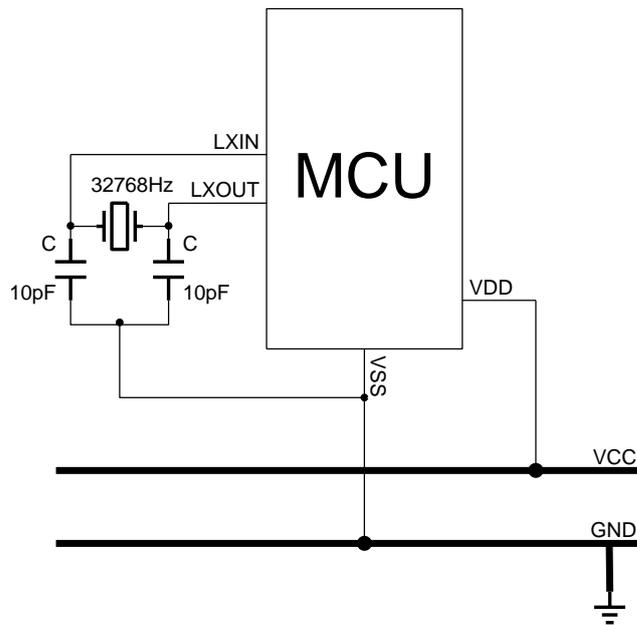
The EHS crystal is switched on and off using the EHSEN bit in [Analog Block Control register \(SYS0_ANBCTRL\)](#).

3.2.3.3 External Low-speed (ELS) Clock

The low-speed oscillator can use 32768 crystal oscillator circuit.

3.2.3.4 CRYSTAL

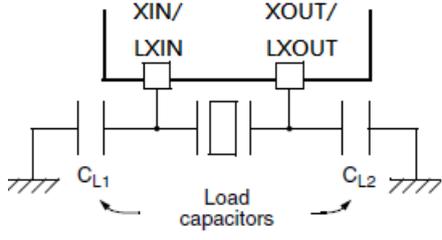
Crystal devices are driven by LXIN, LXOUT pins. The 32768 crystal and 10pF capacitor must be as near as possible to MCU. The ELS crystal is switched on and off using the ELSEN bit in [Analog Block Control register \(SYS0_ANBCTRL\)](#).



* **Note:** Connect the Crystal/Ceramic and C as near as possible to the LXIN/LXOUT/VSS pins of MCU. The capacitor between LXIN/LXOUT and VSS must be 10pF.

3.2.3.5 Bypass Mode

| Clock Source | H/W Configuration | Description |
|-----------------------------------|-------------------|--|
| External clock source (Bypass) | | <p>In Bypass mode, the external clock signal (square, sinus or triangle) with ~50% duty cycle must be provided to drive the XIN/LXIN pin while the XOUT/LXOUT pin should be the inverse of the input clock signal.</p> <p>EHS X'tal can have a frequency of up to 25 MHz. Select this mode by setting EHSEN bit in Analog Block Control register (SYS0_ANBCTRL).</p> <p>ELS X'TAL must have a frequency of 32.768 KHz.</p> |

| | | |
|-----------------------------------|---|---|
| | | You select this mode by setting ELSEN bit in Analog Block Control register (SYS0_ANBCTRL) . |
| External X'TAL (EHS/ELS X'TAL) |  | The 1 to 25 MHz EHS X'TAL has the advantage of producing a very accurate rate on the main clock ELS X'TAL must have a frequency of 32.768 KHz. |

3.2.4 SYSTEM CLOCK (SYSCLK) SELECTION

After a system reset, the IHRC is selected as system clock. When a clock source is used directly or through the PLL as system clock, it is not possible to stop it.

A switch from one clock source to another occurs only if the target clock source is ready (clock stable after startup delay or PLL locked). If a clock source which is not yet ready is selected, the switch will occur when the clock source is ready.

Ready bits in [SYS0_CSST](#) register indicate which clock(s) is (are) ready and SYSCLKST bits in [SYS0_CLKCFG](#) register indicate which clock is currently used as system clock.

3.2.5 CLOCK-OUT CAPABILITY

The MCU clock output (CLKOUT) capability allows the clock to be output onto the external CLKOUT pin. The configuration registers of the corresponding GPIO port must be programmed in alternate function mode.

One of 6 clock signals can be selected as clock output:

1. HCLK
2. IHRC
3. ILRC
4. PLL clock output
5. ELS X'TAL
6. EHS X'TAL

The selection is controlled by the CLKOUTSEL bits in [SYS1_AHBCLKEN](#) register.

3.3 SYSTEM CONTROL REGISTERS 0

Base Address: 0x4006 0000

3.3.1 Analog Block Control register (SYS0_ANBCTRL)

Address Offset: 0x00

Reset value: 0x0000 0001

*** Note: EHSEN / ELSEN / IHRCEN bit can NOT be cleared if the EHS X'tal / ELS X'tal / IHRC is selected as system clock or is selected to become the system clock.**

| Bit | Name | Description | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:6 | Reserved | | R | 0 |
| 5 | EHSFREQ | Frequency range (driving ability) of EHS X'TAL 0: <=12MHz 1: >12MHz | R/W | 0 |
| 4 | EHSEN | External high-speed clock enable 0: Disable EHS X'TAL. 1: Enable EHS X'TAL. | R/W | 0 |
| 3 | Reserved | | R | 0 |
| 2 | ELSEN | External low-speed oscillator enable 0: Disable External 32.768 KHz oscillator 1: Enable External 32.768 KHz oscillator | R/W | 0 |
| 1 | Reserved | | R | 0 |
| 0 | IHRCEN | Internal high-speed clock enable 0: Disable internal 12 MHz RC oscillator. 1: Enable internal 12 MHz RC oscillator. | R/W | 1 |

3.3.2 PLL control register (SYS0_PLLCTRL)

Address Offset: 0x04

*** Note: PLEN bit can NOT be cleared if the PLL is selected as system clock or is selected to become the system clock.**

| Bit | Name | Description | Attribute | Reset |
|-------|----------------|---|-----------|-------|
| 31:16 | Reserved | | R | 0 |
| 15 | PLEN | PLL enable 0: Disable 1: Enable | R/W | 0 |
| 14 | Reserved | | R | 0 |
| 13:12 | PLLCLKSEL[1:0] | System PLL clock source 00: IHRC 12 MHz oscillator 01: EHS X'TAL 10 MHz ~ 25 MHz Other: Reserved | R/W | 0 |
| 11:9 | Reserved | | R | 0 |
| 8 | FSEL | Front divider value. The division value F is the programmed 2^{FSEL} 0: F = 1 1: F = 2 | R/W | 0 |
| 7:5 | PSEL[2:0] | Post divider value. P= PSEL[2:0]*2 | R/W | 011b |

| | | | | |
|------------|-----------|--|-----|-----|
| | | 000~010: Reserved 011: P = 6 100: P = 8 101: P = 10 110: P = 12 111: P = 14 | | |
| 4:0 | MSEL[4:0] | Feedback divider value. M: 3~31 | R/W | 0x3 |

To select the appropriate values for M, P, and F, it is recommended to follow these constraints:

1. $10\text{MHz} \leq F_{\text{CLKIN}} \leq 25\text{MHz}$
2. $150\text{MHz} \leq F_{\text{VCO}} \leq 330\text{MHz}$
3. $2 < M \leq 31$
4. $F = 1$, or 2
5. $P = 6, 8, 10, 12$, or 14 (**duty 50% +/- 2.5%**)
6. $F_{\text{CLKOUT}} = 20\text{MHz}, 30\text{MHz}, 40\text{MHz}, 50\text{MHz}, 24\text{MHz}, 36\text{MHz}, 48\text{MHz}, 32\text{MHz}, 22\text{MHz}, 24\text{MHz}, 50\text{MHz}$
with jitter $< \pm 500$ ps

3.3.2.1 RECOMMEND FREQUENCY SETTING

$$F_{\text{VCO}} = F_{\text{CLKIN}} / F * M$$

$$F_{\text{CLKOUT}} = F_{\text{VCO}} / P$$

| F_{CLKIN} (MHz) | FSEL | $F=2^{\text{FSEL}}$ | MSEL[4:0]=M | F_{VCO} (MHz) $=F_{\text{CLKIN}} / F * M$ | PSEL[2:0] | $P = \text{PSEL}[2:0] * 2$ | F_{CLKOUT} (MHz) |
|--------------------------|------|---------------------|-------------|---|-----------|----------------------------|---------------------------|
| 10 | 0 | 1 | 20 | 200 | 5 | 10 | 20 |
| 10 | 0 | 1 | 22 | 220 | 5 | 10 | 22 |
| 10 | 0 | 1 | 18 | 180 | 3 | 6 | 30 |
| 10 | 0 | 1 | 24 | 240 | 3 | 6 | 40 |
| 10 | 0 | 1 | 30 | 300 | 3 | 6 | 50 |
| 12 | 0 | 1 | 16 | 192 | 4 | 8 | 24 |
| 12 | 0 | 1 | 18 | 216 | 3 | 6 | 36 |
| 12 | 0 | 1 | 24 | 288 | 3 | 6 | 48 |
| 12 | 0 | 1 | 25 | 300 | 3 | 6 | 50 |
| 16 | 0 | 1 | 16 | 256 | 4 | 8 | 32 |
| 16 | 0 | 1 | 18 | 288 | 3 | 6 | 48 |
| 20 | 1 | 2 | 30 | 300 | 3 | 6 | 50 |
| 22 | 0 | 1 | 12 | 264 | 3 | 6 | 44 |
| 24 | 0 | 1 | 12 | 288 | 3 | 6 | 48 |
| 25 | 0 | 1 | 12 | 300 | 3 | 6 | 50 |

3.3.3 Clock Source Status register (SYS0_CSST)

Address Offset: 0x08

| Bit | Name | Description | Attribute | Reset |
|-------------|----------|--|-----------|-------|
| 31:7 | Reserved | | R | 0 |
| 6 | PLLRDY | PLL clock ready flag 0: PLL unlocked 1: PLL locked | R | 0 |
| 5 | Reserved | | R | 0 |
| 4 | EHSRDY | External high-speed clock ready flag 0: EHS oscillator not ready 1: EHS oscillator ready | R | 0 |
| 3 | Reserved | | R | 0 |

| | | | | |
|----------|----------|---|---|---|
| 2 | ELSRDY | External low-speed clock ready flag 0: EHS oscillator not ready 1: EHS oscillator ready | R | 0 |
| 1 | Reserved | | R | 0 |
| 0 | IHRCDY | IHRC ready flag 0: IHRC not ready 1: IHRC ready | R | 1 |

3.3.4 System Clock Configuration register (SYS0_CLKCFG)

Address Offset: 0x0C

| Bit | Name | Description | Attribute | Reset |
|-------------|----------------|--|-----------|-------|
| 31:7 | Reserved | | R | 0 |
| 6:4 | SYSCLKST[2:0] | System clock switch status Set and cleared by HW to indicate which clock source is used as system clock. 000: IHRC is used as system clock 001: ILRC is used as system clock 010: EHS X'TAL is used as system clock 011: ELS X'TAL is used as system clock 100: PLL is used as system clock Other: Reserved | R | 0 |
| 3 | Reserved | | R | 0 |
| 2:0 | SYSCLKSEL[2:0] | System clock switch Set and cleared by SW. 000: IHRC 001: ILRC 010: EHS X'TAL 011: ELS X'TAL 100: PLL output Other: Reserved | R/W | 0 |

3.3.5 AHB Clock Prescale register (SYS0_AHBCP)

Address Offset: 0x10

| Bit | Name | Description | Attribute | Reset |
|-------------|--------------|---|-----------|-------|
| 31:4 | Reserved | | R | 0 |
| 3:0 | AHBPRES[3:0] | AHB clock source prescale value 0000: SYSCLK / 1 0001: SYSCLK / 2 0010: SYSCLK / 4 0011: SYSCLK / 8 0100: SYSCLK / 16 0101: SYSCLK / 32 0110: SYSCLK / 64 0111: SYSCLK / 128 1000: SYSCLK / 256 1001: SYSCLK / 512 Other: Reserved | R/W | 0 |

3.3.6 System Reset Status register (SYS0_RSTST)

Address Offset: 0x14

This register contains the reset source except DPDWAKEUP reset, since the LPFLAG bit in PMU_CTRL register had presented this case.

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:5 | Reserved | | R | 0 |
| 4 | PORRSTF | POR reset flag Set by HW when a POR reset occurs. 0: Read→No POR reset occurred Write→Clear this bit 1: POR reset occurred. | R/W | 1 |
| 3 | EXTRSTF | External reset flag Set by HW when a reset from the <u>RESET</u> pin occurs. 0: Read→No reset from RESET pin occurred Write→Clear this bit 1: Reset from RESET pin occurred. | R/W | 0 |
| 2 | LVDRSTF | LVD reset flag Set by HW when a LVD reset occurs. 0: Read→No LVD reset occurred Write→Clear this bit 1: LVD reset occurred. | R/W | 0 |
| 1 | WDTRSTF | WDT reset flag Set by HW when a WDT reset occurs. 0: Read→No watchdog reset occurred Write→Clear this bit 1: Watchdog reset occurred. | R/W | 0 |
| 0 | SWRSTF | Software reset flag Set by HW when a software reset occurs. 0: Read→No software reset occurred Write→Clear this bit 1: Software reset occurred. | R/W | 1 |

3.3.7 LVD Control register (SYS0_LVDCTRL)

Address Offset: 0x18

The LVD control register selects four separate threshold values for generating a LVD interrupt to the NVIC or LVD reset.

| Bit | Name | Description | Attribute | Reset |
|-------|-----------------|---|-----------|-------|
| 31:16 | Reserved | | R | 0 |
| 15 | LV DEN | LVD enable 0: Disable 1: Enable | R/W | 0 |
| 14 | LVDRSTEN | LVD Reset enable 0: Disable 1: Enable | R/W | 0 |
| 13:6 | Reserved | | R | 0 |
| 5:4 | LV DINTLVL[1:0] | LVD interrupt level 00: The interrupt assertion threshold voltage is 2.00V 01: The interrupt assertion threshold voltage is 2.70V 10: The interrupt assertion threshold voltage is 3.00V 11: Reserved | R/W | 0 |
| 3:2 | Reserved | | R | 0 |
| 1:0 | LVDRSTLVL[1:0] | LVD reset level 00: The reset assertion threshold voltage is 2.40V 01: The reset assertion threshold voltage is 2.70V 10: The reset assertion threshold voltage is 2.00V 11: Reserved | R/W | 0 |

3.3.8 External RESET Pin Control register (SYS0_EXRSTCTRL)

Address Offset: 0x1C

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:1 | Reserved | | R | 0 |
| 0 | RESETDIS | External RESET pin disable bit. 0: Enable external <u>RESET</u> pin. (P3.6 acts as <u>RESET</u> pin) 1: Disable. (P3.6 acts as GPIO pin) | R/W | 0 |

3.3.9 SWD Pin Control register (SYS0_SWDCtrl)

Address Offset: 0x20

| Bit | Name | Description | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:1 | Reserved | | R | 0 |
| 0 | SWDDIS | SWD pin disable bit. 0: Enable SWD pin. (P0.11 acts as SWDIO pin, P0.10 acts as SWCLK pin) 1: Disable. (P0.11 and P0.10 act as GPIO pins) | R/W | 0 |

3.4 SYSTEM CONTROL REGISTERS 1

Base Address: 0x4005 E000

3.4.1 AHB Clock Enable register (SYS1_AHBCLKEN)

Address Offset: 0x00

The SYS_AHBCLKEN register enables the AHB clock to individual system and peripheral blocks.

*** Note:**

- 1. When the clock is disabled, the peripheral register values may not be readable by SW and the value returned is always 0x0.
- 2. HW will replace GPIO with CLKOUT function directly if CLKOUTSEL is Not 0.

| Bit | Name | Description | Attribute | Reset |
|-------|----------------|---|-----------|-------|
| 31 | Reserved | | R | 0 |
| 30:28 | CLKOUTSEL[2:0] | Clock output source 000: Disable 001: ILRC clock 010: ELS clock 100: HCLK 101: IHRC clock 110: EHS clock 111: PLL clock output | R/W | 0 |
| 27:25 | Reserved | | R | 0 |
| 24 | WDTCLKEN | Enables clock for WDT. 0: Disable 1: Enable | R/W | 1 |
| 23 | RTCCLKEN | Enables clock for RTC. 0: Disable 1: Enable | R/W | 0 |
| 22 | I2SCLKEN | Enables clock for I2S. 0: Disable 1: Enable | R/W | 0 |
| 21 | I2C0CLKEN | Enables clock for I2C0. 0: Disable 1: Enable | R/W | 0 |
| 20 | I2C1CLKEN | Enables clock for I2C1. 0: Disable 1: Enable | R/W | 0 |
| 19:18 | Reserved | | R | 0 |
| 17 | USART1CLKEN | Enables clock for USART1. 0: Disable 1: Enable | R/W | 0 |
| 16 | USART0CLKEN | Enables clock for USART0. 0: Disable 1: Enable | R/W | 0 |
| 15:14 | Reserved | | R | 0 |
| 13 | SSP1CLKEN | Enables clock for SSP1. 0: Disable 1: Enable | R/W | 0 |
| 12 | SSP0CLKEN | Enables clock for SSP0. 0: Disable 1: Enable | R/W | 0 |
| 11 | ADCCLKEN | Enables clock for ADC. 0: Disable 1: Enable | R/W | 0 |

| | | | | |
|-----|-------------|--|-----|---|
| 10 | Reserved | | R | 0 |
| 9 | CT32B1CLKEN | Enables clock for CT32B1. 0: Disable 1: Enable | R/W | 0 |
| 8 | CT32B0CLKEN | Enables clock for CT32B0. 0: Disable 1: Enable | R/W | 0 |
| 7 | CT16B1CLKEN | Enables clock for CT16B1. 0: Disable 1: Enable | R/W | 0 |
| 6 | CT16B0CLKEN | Enables clock for CT16B0. 0: Disable 1: Enable | R/W | 0 |
| 5:4 | Reserved | | R | 0 |
| 3 | GPIOCKEN | Enables clock for GPIO. 0: Disable 1: Enable | R/W | 1 |
| 2:0 | Reserved | | R | 0 |

3.4.2 APB Clock Prescale register 0 (SYS1_APB0)

Address Offset: 0x04

* **Note:** Must reset the corresponding peripheral with SYS1_PRST register after changing the prescale value.

| Bit | Name | Description | Attribute | Reset |
|-------|----------------|---|-----------|-------|
| 31:27 | Reserved | | R | 0 |
| 26:24 | SSP1PRE[2:0] | SSP1 clock source prescale value 000: HCLK / 1 001: HCLK / 2 010: HCLK / 4 011: HCLK / 8 100: HCLK / 16 Other: Reserved | R/W | 0 |
| 23 | Reserved | | R | 0 |
| 22:20 | SSP0PRE[2:0] | SSP0 clock source prescale value 000: HCLK / 1 001: HCLK / 2 010: HCLK / 4 011: HCLK / 8 100: HCLK / 16 Other: Reserved | R/W | 0 |
| 19 | Reserved | | R | 0 |
| 18:16 | ADCPRE[2:0] | ADC clock source prescale value 000: HCLK / 1 001: HCLK / 2 010: HCLK / 4 011: HCLK / 8 100: HCLK / 16 Other: Reserved | R/W | 0 |
| 15 | Reserved | | R | 0 |
| 14:12 | CT32B1PRE[2:0] | CT32B1 clock source prescale value 000: HCLK / 1 001: HCLK / 2 010: HCLK / 4 011: HCLK / 8 100: HCLK / 16 Other: Reserved | R/W | 0 |

| | | | | |
|------|----------------|--|-----|---|
| 11 | Reserved | | R | 0 |
| 10:8 | CT32B0PRE[2:0] | CT32B0 clock source prescale value. 000: HCLK / 1 001: HCLK / 2 010: HCLK / 4 011: HCLK / 8 100: HCLK / 16 Other: Reserved | R/W | 0 |
| 7 | Reserved | | R | 0 |
| 6:4 | CT16B1PRE[2:0] | CT16B1 clock source prescale value 000: HCLK / 1 001: HCLK / 2 010: HCLK / 4 011: HCLK / 8 100: HCLK / 16 Other: Reserved | R/W | 0 |
| 3 | Reserved | | R | 0 |
| 2:0 | CT16B0PRE[2:0] | CT16B0 clock source prescale value 000: HCLK / 1 001: HCLK / 2 010: HCLK / 4 011: HCLK / 8 100: HCLK / 16 Other: Reserved | R/W | 0 |

3.4.3 APB Clock Prescale register 1 (SYS1_APBPCP1)

Address Offset: 0x08

* **Note:** Must reset the corresponding peripheral with SYS1_PRST register after changing the prescale value.

| Bit | Name | Description | Attribute | Reset |
|-------|----------------|---|-----------|-------|
| 31:28 | CLKOUTPRE[3:0] | Clock-out source prescale value 0000: Clock-out source / 1 0001: Clock-out source / 2 0010: Clock-out source / 4 0011: Clock-out source / 8 0100: Clock-out source / 16 0101: Clock-out source / 32 0110: Clock-out source / 64 0111: Clock-out source / 128 1000: Clock-out source / 256 1001: Clock-out source / 512 Other: Reserved | R/W | 0 |
| 27 | Reserved | | R | 0 |
| 26:24 | I2C1PRE[2:0] | I2C1 clock source prescale value 000: HCLK / 1 001: HCLK / 2 010: HCLK / 4 011: HCLK / 8 100: HCLK / 16 Other: Reserved | R/W | 0 |
| 23 | Reserved | | R | 0 |
| 22:20 | WDTPRE[2:0] | WDT clock source prescale value 000: WDT_PCLK = WDT clock source / 1 001: WDT_PCLK = WDT clock source / 2 010: WDT_PCLK = WDT clock source / 4 011: WDT_PCLK = WDT clock source / 8 100: WDT_PCLK = WDT clock source / 16 | R/W | 0 |

| | | | | |
|--------------|----------------|---|-----|---|
| | | 101: WDT_PCLK = WDT clock source / 32 Other: Reserved | | |
| 19:15 | Reserved | | R | 0 |
| 14:12 | I2SPRE[2:0] | I2S clock source prescale value 000: HCLK / 1 001: HCLK / 2 010: HCLK / 4 011: HCLK / 8 100: HCLK / 16 Other: Reserved | R/W | 0 |
| 11 | Reserved | | R | 0 |
| 10:8 | I2C0PRE[2:0] | I2C0 clock source prescale value 000: HCLK / 1 001: HCLK / 2 010: HCLK / 4 011: HCLK / 8 100: HCLK / 16 Other: Reserved | R/W | 0 |
| 7 | Reserved | | R | 0 |
| 6:4 | USART1PRE[2:0] | USART1 clock source prescale value 000: HCLK / 1 001: HCLK / 2 010: HCLK / 4 011: HCLK / 8 100: HCLK / 16 Other: Reserved | R/W | 0 |
| 3 | Reserved | | R | 0 |
| 2:0 | USART0PRE[2:0] | USART0 clock source prescale value 000: HCLK / 1 001: HCLK / 2 010: HCLK / 4 011: HCLK / 8 100: HCLK / 16 Other: Reserved | R/W | 0 |

3.4.4 Peripheral Reset register (SYS1_PRST)

Address Offset: 0x0C

All bits are cleared by HW automatically after setting as “1”.

| Bit | Name | Description | Attribute | Reset |
|--------------|-----------|---|-----------|-------|
| 31:25 | Reserved | | R | 0 |
| 24 | WDTRST | WDT reset 0: No effect 1: Reset WDT | R/W | 0 |
| 23 | RTCST | RTC reset 0: No effect 1: Reset RTC | R/W | 0 |
| 22 | I2SRST | I2S reset 0: No effect 1: Reset I2S | R/W | 0 |
| 21 | I2C0RST | I2C0 reset 0: No effect 1: Reset I2C0 | R/W | 0 |
| 20 | I2C1RST | I2C1 reset 0: No effect 1: Reset I2C1 | R/W | 0 |
| 19:18 | Reserved | | R | 0 |
| 17 | USART1RST | USART1 reset 0: No effect 1: Reset USART1 | R/W | 0 |

| | | | | |
|--------------|-----------|---|-----|---|
| 16 | USART0RST | USART0 reset 0: No effect 1: Reset USART0 | R/W | 0 |
| 15:14 | Reserved | | R | 0 |
| 13 | SSP1RST | SSP1 reset 0: No effect 1: Reset SSP1 | R/W | 0 |
| 12 | SSP0RST | SSP0 reset 0: No effect 1: Reset SSP0 | R/W | 0 |
| 11 | ADCRST | ADC reset 0: No effect 1: Reset ADC | R/W | 0 |
| 10 | Reserved | | R | 0 |
| 9 | CT32B1RST | CT32B1 reset 0: No effect 1: Reset CT32B1 | R/W | 0 |
| 8 | CT32B0RST | CT32B0 reset 0: No effect 1: Reset CT32B0 | R/W | 0 |
| 7 | CT16B1RST | CT16B1 reset 0: No effect 1: Reset CT16B1 | R/W | 0 |
| 6 | CT16B0RST | CT16B0 reset 0: No effect 1: Reset CT16B0 | R/W | 0 |
| 5:4 | Reserved | | R | 0 |
| 3 | GPIOP3RST | GPIO port 3 reset 0: No effect 1: Reset GPIO port 3 | R/W | 0 |
| 2 | GPIOP2RST | GPIO port 2 reset 0: No effect 1: Reset GPIO port 2 | R/W | 0 |
| 1 | GPIOP1RST | GPIO port 1 reset 0: No effect 1: Reset GPIO port 1 | R/W | 0 |
| 0 | GPIOP0RST | GPIO port 0 reset 0: No effect 1: Reset GPIO port 0 | R/W | 0 |

4 SYSTEM OPERATION MODE

4.1 OVERVIEW

The chip builds in four operating mode for difference clock rate and power saving reason. These modes control oscillators, op-code operation and analog peripheral devices' operation.

- Normal mode
- Sleep mode
- Deep sleep mode
- Deep Power-down mode

4.2 NORMAL MODE

In Normal mode, the ARM Cortex-M0 core, memories, and peripherals are clocked by the system clock. The [SYS1_AHBCLKEN](#) register controls which peripherals are running.

Selected peripherals have individual peripheral clocks with their own clock dividers in addition to the system clock. The peripheral clocks can be disabled respectively.

The power to various analog blocks (IHRC, EHS X'TAL, ELS X'TAL, PLL, Flash, LVD, ADC) can be controlled at any time individually through the enable bit of all blocks.

4.3 LOW-POWER MODES

There are three special modes of processor power reduction: Sleep mode, Deep-sleep mode, and Deep power-down mode. The [PMU_CTRL](#) register controls which mode is going to entered.

The CPU clock rate may also be controlled as needed by changing clock sources, re-configuring PLL values, and/or altering the system clock divider value. This allows a trade-off of power versus processing speed based on application requirements.

Run-time power control allows disable the clocks to individual on-chip peripherals, allowing fine tuning of power consumption by eliminating all dynamic power use in any peripherals that are not required for the application. Selected peripherals have their own clock divider for power control.

- * **Note: 1. The debug mode is not supported in Deep-sleep and Deep Power-down mode.**
- 2. The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.**

4.3.1 SLEEP MODE

In Sleep mode, the system clock to the ARM Cortex-M0 core is stopped and execution of instructions is suspended.

Peripheral functions, if selected to be clocked in [SYS1_AHBCLKEN](#) register, continue operation during Sleep mode and may generate interrupts to cause the processor to resume execution. Sleep mode eliminates dynamic power used by the processor itself, memory systems and related controllers, and internal buses.

The power state of the analog blocks (IHRC, EHS X'TAL, ELS X'TAL, PLL, Flash, LVD, ADC) is determined by the enable bit of all blocks.

The processor state and registers, peripheral registers, and internal SRAM values are maintained and the logic levels of the pins remain static.

Wake up the chip from Sleep mode by an interrupt occurs.

The RESET pin has keep functionality in Sleep mode.

The Sleep mode is entered by using the following steps:

1. Write 1 to SLEEPEN bit in [PMU_CTRL](#) register.
2. Execute ARM Cortex-M0 WFI instruction.

4.3.2 DEEP-SLEEP MODE

In Deep-sleep mode, the system clock to the ARM Cortex-M0 core is stopped, and execution of instructions is suspended.

The clock to the peripheral functions are stopped because the power state of oscillators are powered down, the clock source are stopped, except RTC low speed clock source (ELS X'TAL, ILRC) if used.

*** Note: User SHALL decide to power down RTC low speed clock source (ELS X'TAL, ILRC oscillator) or not if RTC is enabled.**

The processor state and registers, peripheral registers, and internal SRAM values are maintained and the logic levels of the pins remain static.

Wake up the chip from Deep-sleep mode by GPIO P0.0~P0.11 or RTC interrupt.

The RESET pin has keep functionality in Deep-sleep mode.

The Deep-sleep mode is entered by using the following steps:

1. Write 1 to DSLEEPEN bit in [PMU_CTRL](#) register.
2. Execute ARM WFI instruction.

The advantage of the Deep-sleep mode is that can power down clock generating blocks such as oscillators and PLL, thereby gaining far greater dynamic power savings over Sleep mode. In addition, the Flash can be powered down in Deep-sleep mode resulting in savings in static leakage power, however at the expense of longer wake-up times for the Flash memory.

4.3.3 DEEP POWER-DOWN (DPD) MODE

In Deep power-down mode, power (Turn off the on-chip voltage regulator) and clocks are shut off to the entire chip with the exception of the DPDWAKEUP pin. DPDWAKEUP pin must be pulled HIGH externally to enter Deep power-down mode and pulled LOW to exit Deep power-down mode.

The processor state and registers, peripheral registers, and internal SRAM values are not retained. However, the chip can retain data in four BACKUP registers.

Wakes up the chip from Deep power-down mode by pulling the DPDWAKEUP pin LOW (Turn on the on-chip voltage regulator. When the core voltage reaches the power-on-reset (POR) trip point, a system reset will be triggered and the chip re-boots).

The RESET pin has no functionality in Deep power-down mode.

4.3.3.1 Entering Deep power-down mode

Follow these steps to enter Deep power-down mode from Normal mode:

1. Pull the DPDWAKEUP pin externally HIGH (Strongly recommended to set output high first, and then set as input pull-up to reduce pull-up time).
2. (Optional) Save data to be retained during Deep power-down to the DATA bits in [Backup registers](#).
3. Write 1 to DPDEN bit in [PMU_CTRL](#) register to enable Deep power-down mode.
4. Time spent between step 1 and step 5 shall longer than 20 us.
5. Execute ARM Cortex-M0 WFI instruction.

After step 5, the PMU turns off the on-chip voltage regulator and waits for a wake-up signal from the DPDWAKEUP pin.

4.3.3.2 Exiting Deep power-down mode

Follow these steps to wake up the chip from Deep power-down mode:

1. DPDWAKEUP pin transition from HIGH to LOW.
 - The PMU will turn on the on-chip voltage regulator. When the core voltage reaches the power-on-reset (POR) Trigger point, a system reset will be triggered and the chip reboots.
 - All registers except the PMU_BKP0 to PMU_BKP 15 and PMU_CTRL will be reset.
2. Once the chip has rebooted, read DPDEN bit in [PMU_CTRL](#) register to verify that the reset was caused by a wake-up event from Deep power-down and was not a cold reset.
3. Clear the DPDEN bit in [PMU_CTRL](#) register.
4. (Optional) Read the stored data in the backup registers.
5. Setup the PMU for the next Deep power-down cycle.

4.4 WAKEUP INTERRUPT

System will exit Deep-sleep mode when GPIO indicates a WAKEUP interrupt to the ARM core. The port pins P0.0 to P0.11 are served as wakeup pins. The user must program the registers for each pin to set the appropriate edge polarity for the corresponding wakeup event. Only edge sensitive is supported to wakeup MCU. Furthermore, the interrupts corresponding to each input must be enabled in the NVIC. Interrupts 0 in the NVIC correspond to 12 GPIO pins.

4.5 WAKEUP

4.5.1 OVERVIEW

Under low power mode, program doesn't execute. The wakeup trigger can wake the system up to normal mode. The wakeup function builds in interrupt operation and trigger system executing interrupt service routine as system wakeup occurrence.

- * The wakeup trigger sources of the Sleep mode are all interrupts and the [RESET](#) pin.
- * The wakeup trigger sources of the Deep-sleep mode are the P0 WAKEUP interrupt, RTC interrupt, and the [RESET](#) pin.
- * The wakeup trigger source of the Deep Power-down mode is the DPDWAKEUP pin which is input pull-up before entering DPD mode.

4.5.2 WAKEUP TIME

When the system is in Sleep mode, the high clock is enabled or disabled by F/W. If the high clock stops and MCU is

waken up from Sleep mode, MCU waits for 2048 external high-speed oscillator clocks and 32 internal high-speed oscillator clocks as the wakeup time to stable the oscillator circuit. After the wakeup time, the system goes into the normal mode.

* **Note: Wakeup from Sleep mode spends NO wakeup time if the clock doesn't stop.**

When the system is in Deep-sleep mode, the high clock will stop. When MCU is waken up from Deep-sleep mode, MCU waits for $2T \cdot ILRC$ and IHRC/EHS warm up time ($32T \cdot IHRC / 2048T \cdot EHS$). After the wakeup time, the system goes into the normal mode.

The value of the external high clock oscillator wakeup time from Deep sleep mode is as the following.

The total Wakeup time of EHS X'tal = $2T \cdot ILRC + 1/F_{EHS} \cdot 2048$ (sec) + high clock start-up time

➤ Example: $F_{EHS} = 20\text{MHz}$, the wakeup time is as the following.

$$\begin{aligned} \text{The total Wakeup time} &= 2T \cdot ILRC + 1/F_{EHS} \cdot 2048 + \text{oscillator start-up time} \\ &= 125\mu\text{s} + 102.4 \mu\text{s} + \text{oscillator start-up time} \quad (F_{EHS} = 20\text{MHz}) \end{aligned}$$

The value of the IHRC wakeup time from Deep sleep mode is as the following.

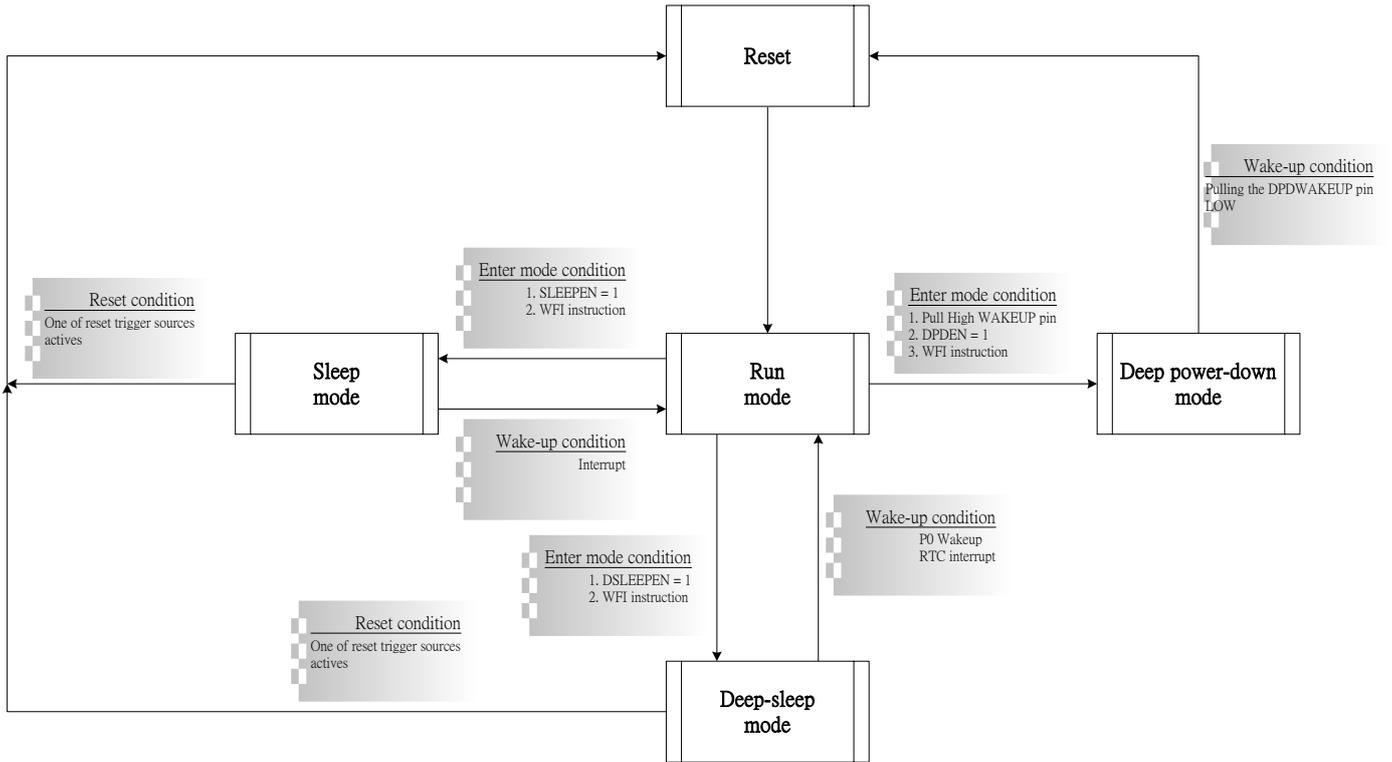
The total Wakeup time of IHRC = $2T \cdot ILRC + 1/F_{IHRC} \cdot 32$ (sec)

➤ Example: $F_{IHRC} = 12\text{MHz}$, the wakeup time is as the following.

$$\text{The total Wakeup time} = 2T \cdot ILRC + 1/F_{IHRC} \cdot 32 = 125\mu\text{s} + 2.67 \mu\text{s} \quad (F_{IHRC} = 12\text{MHz})$$

* **Note: The high clock start-up time is depended on the VDD and oscillator type of high clock.**

4.6 STATE MACHINE OF PMU



4.7 OPERATION MODE COMPARISION TABLE

| | Normal Mode | | Low-Power Mode | | |
|-----------------------|-----------------------------------|------------------------------|--|-----------------|----------------------|
| | | | Sleep Mode | Deep-Sleep Mode | Deep Power-down Mode |
| IHRC | By IHRCEN | | Disable | OFF | |
| ILRC | ON | | *** | OFF | |
| EHS X'TAL | By EHSEN | | Disable | OFF | |
| ELS X'TAL | By ELSEN | | *** | OFF | |
| PLL | By PLEN | | Disable | OFF | |
| Cortex-M0 core | Running | Stop | Stop | Stop | |
| Flash ROM | Enable | Disable | Disable | OFF | |
| RAM | Enable | Maintain | Maintain | OFF | |
| ADC | By ADENB | | Disable | Disable | |
| LVD | By LVDEN | | Disable | OFF | |
| RTC | By RTCEN | | By RTCEN | OFF | |
| Peripherals | By Enable bit of each peripherals | | Disable HCLK | OFF | |
| Wakeup Source | N/A | All interrupts, RESET pin | Wakeup interrupt, RTC interrupt, RESET pin | DPDWAKEUP pin | |

| RTCENB | RTC_CLKS | ILRC* | ELS* |
|--------|----------|-------|------|
| 0 | --- | X | X |
| 1 | 0 (ILRC) | O | X |
| | 1 (ELS) | X | O |

4.8 PMU REGISTERS

Base Address: 0x4003 2000

4.8.1 Backup registers 0 to 15 (PMU_BKP0~15)

Address Offset: 0x0, 0x04, 0x08, 0x0C, 0x10, 0x14, 0x18, 0x1C, 0x20, 0x24, 0x28, 0x2C, 0x30, 0x34, 0x38, 0x3C

The backup registers retain data through the Deep power-down mode when power is still applied to the VDD pin but the chip has entered Deep power-down mode.

*** Note: Backup registers will be reset only when all power has been completely removed from the chip.**

| Bit | Name | Description | Attribute | Reset |
|------|-----------------|---|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7:0 | BACKUPDATA[7:0] | BACKUPDATA Data retained during Deep power-down mode. | R/W | 0 |

4.8.2 Power control register (PMU_CTRL)

Address Offset: 0x40

The power control register selects whether one of the ARM Cortex-M0 controlled power-down modes (Sleep mode or Deep-sleep mode) or the Deep power-down mode is entered and provides the flags for Sleep or Deep-sleep modes and Deep power-down modes respectively.

*** Note: 1. The PMU_CTRL register retains data through the Deep power-down mode when power is still applied to the VDD pin, and will be reset only when all power has been completely removed from the chip.**
2. The pins which are not pin-out shall be set correctly to decrease power consumption in low-power modes. Strongly recommended to set these pins as input pull-up.

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:3 | Reserved | | R | 0 |
| 2 | SLEEPEN | Sleep mode enable 0: Disable. 1: Enable. WFI instruction will make MCU enter Sleep mode. | R/W | 0 |
| 1 | DSLEEPEN | Deep sleep mode enable 0: Disable. 1: Enable. WFI instruction will make MCU enter Deep-sleep mode. | R/W | 0 |
| 0 | DPDEN | Deep power-down mode enable 0: Disable. 1: Enable. WFI instruction will make MCU enter Deep power-down mode. | R/W | 0 |

5 GENERAL PURPOSE I/O PORT (GPIO)

5.1 OVERVIEW

Digital ports can be configured input/output by SW

- Each individual port pin can serve as external interrupt input pin.
- Interrupts can be configured on single falling or rising edges and on both edges.
- The I/O configuration registers control the electrical characteristics of the pads.
- Internal pull-up/pull-down resistor.
- Most of the I/O pins are mixed with analog pins and special function pins.

5.2 GPIO MODE

The MODE bits in the [GPIO_n CFG](#) (n=0,1,2,3) register allow the selection of on-chip pull-up or pull-down resistors for each pin or select the repeater mode.

The repeater mode enables the pull-up resistor if the pin is logic HIGH and enables the pull-down resistor if the pin is logic LOW. This causes the pin to retain its last known state if it is configured as an input and is not driven externally. The state retention is not applicable to the Deep power-down mode.

5.3 GPIO REGISTERS

Base Address: 0x4004 4000 (GPIO 0)
 0x4004 6000 (GPIO 1)
 0x4004 8000 (GPIO 2)
 0x4004 A000 (GPIO 3)

5.3.1 GPIO Port n Data register (GPIO_n_DATA) (n=0,1,2,3)

Address offset: 0x00

| Bit | Name | Description | Attribute | Reset |
|-------|------------|--|-----------|-------|
| 31:12 | Reserved | | R | 0 |
| 11:0 | DATA[11:0] | Input data (read) or output data (write) for Pn.0 to Pn.11 | R/W | 0 |

5.3.2 GPIO Port n Mode register (GPIO_n_MODE) (n=0,1,2,3)

Address offset: 0x04

* **Note:** HW will switch I/O Mode directly when Specific function (Peripheral, ADC) is enabled, not through GPIO_n_MODE register.

| Bit | Name | Description | Attribute | Reset |
|-------|------------|--|-----------|-------|
| 31:12 | Reserved | | R | 0 |
| 11:0 | MODE[11:0] | Selects pin x as input or output (x = 0 to 11) 0: Pn.x is configured as input 1: Pn.x is configured as output. | R/W | 0 |

5.3.3 GPIO Port n Configuration register (GPIO_n_CFG) (n=0,1,2,3)

Address offset: 0x08
 Reset value: 0x00AAAAAA

* **Note:** HW will switch I/O Mode directly when Specific function (Peripheral, ADC) is enabled, not through GPIO_n_MODE register.

| Bit | Name | Description | Attribute | Reset |
|-------|------------|---|-----------|-------|
| 31:24 | Reserved | | R | 0 |
| 23:22 | CFG11[1:0] | Configuration of Pn.11 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/pull-up resistor enabled). 11: Repeater mode. | R/W | 10b |
| 21:20 | CFG10[1:0] | Configuration of Pn.10 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/pull-up resistor enabled). 11: Repeater mode. | R/W | 10b |
| 19:18 | CFG9[1:0] | Configuration of Pn.9 00: Pull-up resistor enabled. | R/W | 10b |

| | | | | |
|--------------|-----------|--|-----|-----|
| | | 01: Pull-down resistor enabled. 10: Inactive (no pull-down/pull-up resistor enabled). 11: Repeater mode. | | |
| 17:16 | CFG8[1:0] | Configuration of Pn.8 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/pull-up resistor enabled). 11: Repeater mode. | R/W | 10b |
| 15:14 | CFG7[1:0] | Configuration of Pn.7 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/pull-up resistor enabled). 11: Repeater mode. | R/W | 10b |
| 13:12 | CFG6[1:0] | Configuration of Pn.6 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/pull-up resistor enabled). 11: Repeater mode. | R/W | 10b |
| 11:10 | CFG5[1:0] | Configuration of Pn.5 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/pull-up resistor enabled). 11: Repeater mode. | R/W | 10b |
| 9:8 | CFG4[1:0] | Configuration of Pn.4 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/pull-up resistor enabled). 11: Repeater mode. | R/W | 10b |
| 7:6 | CFG3[1:0] | Configuration of Pn.3 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/pull-up resistor enabled). 11: Repeater mode. | R/W | 10b |
| 5:4 | CFG2[1:0] | Configuration of Pn.2 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/pull-up resistor enabled). 11: Repeater mode. | R/W | 10b |
| 3:2 | CFG1[1:0] | Configuration of Pn.1 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/pull-up resistor enabled). 11: Repeater mode. | R/W | 10b |
| 1:0 | CFG0[1:0] | Configuration of Pn.0 00: Pull-up resistor enabled. 01: Pull-down resistor enabled. 10: Inactive (no pull-down/pull-up resistor enabled). 11: Repeater mode. | R/W | 10b |

5.3.4 GPIO Port n Interrupt Sense register (GPIO_n_IS) (n=0,1,2,3)

Address offset: 0x0C

| Bit | Name | Description | Attribute | Reset |
|--------------|----------|---|-----------|-------|
| 31:12 | Reserved | | R | 0 |
| 11:0 | IS[11:0] | Selects interrupt on pin x as level or edge sensitive (x = 0 to 11). 0: Interrupt on Pn.x is configured as edge sensitive. 1: Interrupt on Pn.x is configured as event sensitive. | R/W | 0 |

5.3.5 GPIO Port n Interrupt Both-edge Sense register (GPIO_n_IBS) (n=0,1,2,3)

Address offset: 0x10

| Bit | Name | Description | Attribute | Reset |
|-------|-----------|--|-----------|-------|
| 31:12 | Reserved | | R | 0 |
| 11:0 | IBS[11:0] | Selects interrupt on Pn.x to be triggered on both edges (x = 0 to 11). 0: Interrupt on Pn.x is controlled through register GPIO_n_IEV. 1: Both edges on Pn.x trigger an interrupt. | R/W | 0 |

5.3.6 GPIO Port n Interrupt Event register (GPIO_n_IEV) (n=0,1,2,3)

Address offset: 0x14

| Bit | Name | Description | Attribute | Reset |
|-------|-----------|---|-----------|-------|
| 31:12 | Reserved | | R | 0 |
| 11:0 | IEV[11:0] | Selects interrupt on pin x to be triggered rising or falling edges (x = 0 to 11). 0: Depending on setting in register GPIO_n_IS, Rising edges or HIGH level on Pn.x trigger an interrupt. 1: Depending on setting in register GPIO_n_IS, Falling edges or LOW level on Pn.x trigger an interrupt. | R/W | 0 |

5.3.7 GPIO Port n Interrupt Enable register (GPIO_n_IE) (n=0,1,2,3)

Address offset: 0x18

Bits set to HIGH in the GPIO_n_IE register allow the corresponding pins to trigger their individual interrupts. Clearing a bit disables interrupt triggering on that pin.

| Bit | Name | Description | Attribute | Reset |
|-------|----------|--|-----------|-------|
| 31:12 | Reserved | | R | 0 |
| 11:0 | IE[11:0] | Selects interrupt on pin x to be enabled (x = 0 to 11). 0: Disable Interrupt on Pn.x 1: Enable Interrupt on Pn.x | R/W | 0 |

5.3.8 GPIO Port n Raw Interrupt Status register (GPIO_n_RIS) (n=0,1,2,3)

Address offset: 0x1C

This register indicates the status for GPIO control raw interrupts. A GPIO interrupt is sent to the interrupt controller if the corresponding bit in GPIO_n_IE register is set.

| Bit | Name | Description | Attribute | Reset |
|-------|----------|---|-----------|-------|
| 31:12 | Reserved | | R | 0 |
| 11:0 | IF[11:0] | GPIO raw interrupt flag (x = 0 to 11). 0: No interrupt on Pn.x 1: Interrupt requirements met on Pn.x. | R | 0 |

5.3.9 GPIO Port n Interrupt Clear register (GPIO_n_IC) (n=0,1,2,3)

Address offset: 0x20

| Bit | Name | Description | Attribute | Reset |
|-------|----------|---|-----------|-------|
| 31:12 | Reserved | | R | 0 |
| 11:0 | IC[11:0] | Selects interrupt flag on pin x to be cleared (x = 0 to 11). 0: No effect 1: Clear interrupt flag on Pn.x | W | 0 |

5.3.10 GPIO Port n Bits Set Operation register (GPIO_n_BSET) (n=0,1,2,3)

Address offset: 0x24

In order for SW to set GPIO bits without affecting any other pins in a single write operation, the GPIO bit is set if the corresponding bit in the GPIO_n_BSET register is set.

| Bit | Name | Description | Attribute | Reset |
|-------|------------|--|-----------|-------|
| 31:12 | Reserved | | R | 0 |
| 11:0 | BSET[11:0] | Bit Set enable (x = 0 to 11) 0: No effect on Pn.x 1: Set Pn.x to "1" | W | 0 |

5.3.11 GPIO Port n Bits Clear Operation register (GPIO_n_BCLR) (n=0,1,2,3)

Address offset: 0x28

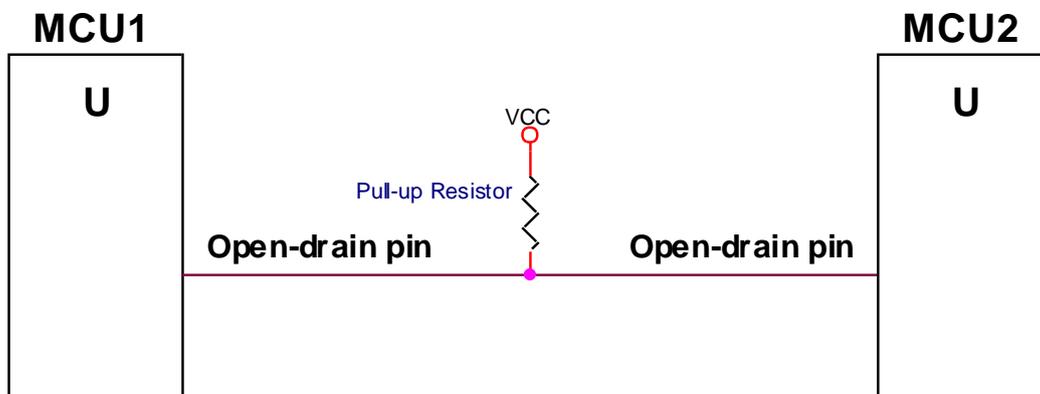
In order for SW to clear GPIO bits without affecting any other pins in a single write operation, the GPIO bit is cleared if the corresponding bit in this register is set.

| Bit | Name | Description | Attribute | Reset |
|-------|------------|--|-----------|-------|
| 31:12 | Reserved | | R | 0 |
| 11:0 | BCLR[11:0] | Bit clear enable (x = 0 to 11) 0: No effect on Pn.x 1: Clear Pn.x. | W | 0 |

5.3.12 GPIO Port n Open-Drain Control register (GPIO_n_ODCTRL) (n=0,1,2,3)

Address offset: 0x2C

Several I/Os have built-in open-drain function and must be set as output mode when enable open-drain function. Open-drain external circuit is as following.



The external pull-up resistor is necessary. The digital output function of I/O only supports sink current capability, so the open-drain output high is driven by pull-up resistor, and output low is sunken by MCU's pin.

* **Note:** VCC shall be less than or equal to VDD of MCU1 and MCU2.

| Bit | Name | Description | Attribute | Reset |
|-----|------|-------------|-----------|-------|
|-----|------|-------------|-----------|-------|

| | | | | |
|-------------|----------|--|-----|---|
| 31:8 | Reserved | | R | 0 |
| 7 | Pn7OC | <u>n = 1</u> P1.7 open-drain control bit. 0: Disable 1: Enable. HW set P1.7 as output mode automatically. | R/W | 0 |
| | | <u>n=0, 2~3</u> Reserved | R | |
| 6 | Pn6OC | <u>n = 1</u> P1.6 open-drain control bit. 0: Disable 1: Enable. HW set P1.6 as output mode automatically. | R/W | 0 |
| | | <u>n=0, 2~3</u> Reserved | R | |
| 5 | Pn5OC | <u>n = 0</u> P0.5 open-drain control bit. 0: Disable 1: Enable. HW set P0.5 as output mode automatically. | R/W | 0 |
| | | <u>n=1~3</u> Reserved | R | |
| 4 | Pn4OC | P0.4 open-drain control bit. 0: Disable 1: Enable. HW set P0.4 as output mode automatically. | R/W | 0 |
| | | <u>n=1~3</u> Reserved | R | |
| 3 | Pn3OC | <u>n = 0</u> P0.3 open-drain control bit. 0: Disable 1: Enable. HW set P0.3 as output mode automatically. | R/W | 0 |
| | | <u>n=1~3</u> Reserved | R | |
| 2 | Pn2OC | <u>n = 0</u> P0.2 open-drain control bit. 0: Disable 1: Enable. HW set P0.2 as output mode automatically. | R/W | 0 |
| | | <u>n=1~3</u> Reserved | R | |
| 1 | Pn1OC | <u>n = 0</u> P0.1 open-drain control bit. 0: Disable 1: Enable. HW set P0.1 as output mode automatically. | R/W | 0 |
| | | <u>n=1~3</u> Reserved | R | |
| 0 | Pn0OC | <u>n = 0</u> P0.0 open-drain control bit. 0: Disable 1: Enable. HW set P0.0 as output mode automatically. | R/W | 0 |
| | | <u>n=1~3</u> Reserved | R | |

6 10 CHANNEL ANALOG TO DIGITAL CONVERTOR (ADC)

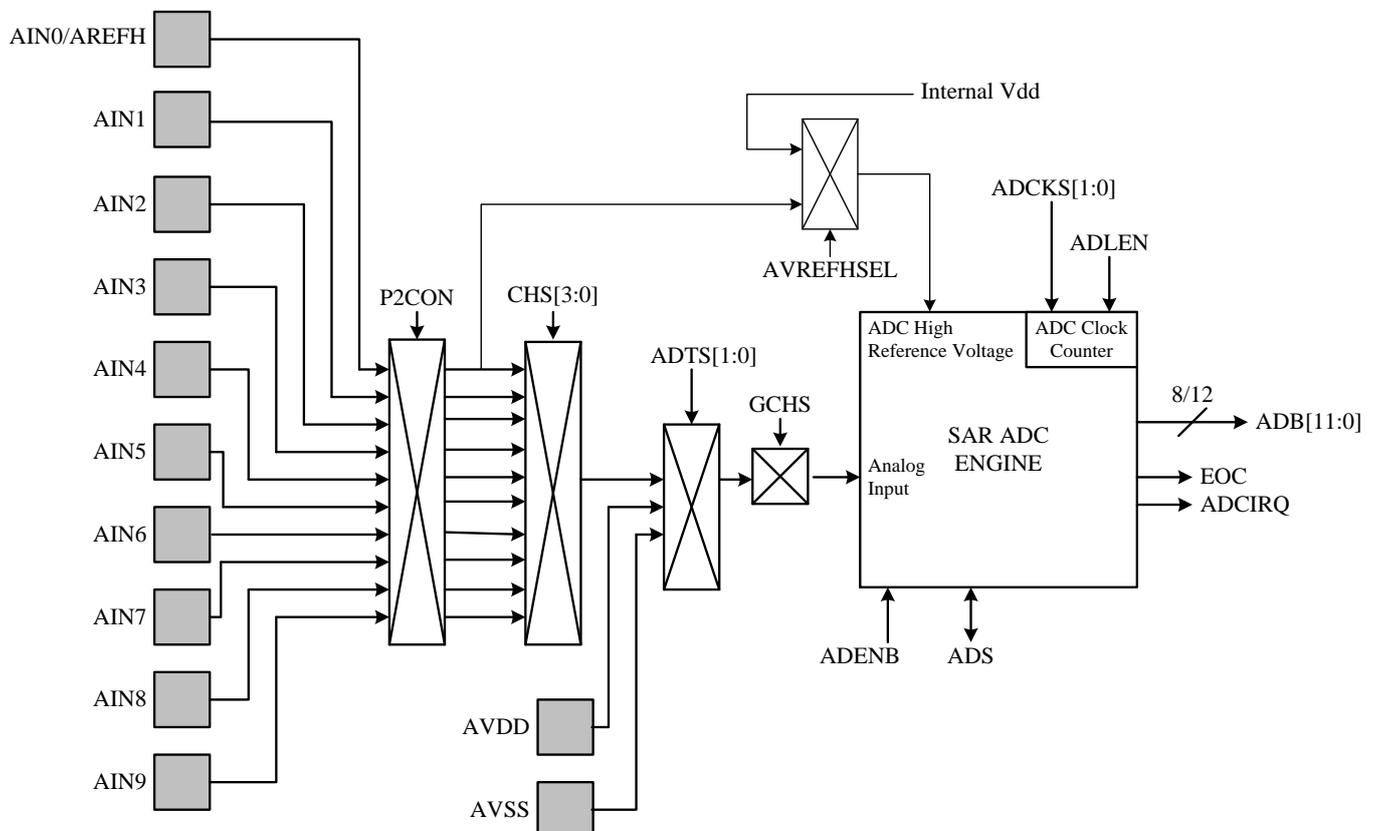
6.1 OVERVIEW

This analog to digital converter has 10-input sources with up to 4096-step resolution to transfer analog signal into 12-bits digital data. The sequence of ADC operation is to select input source (AIN0 ~ AIN9) at first, then set GCHS and ADS bit to "1" to start conversion. When the conversion is complete, the ADC circuit will set EOC bit to "1" and final value output in ADB register.

The ADC is 10-channel SAR structure and 12-bit resolution. Build in P2CON, to set pure analog input pin. It is necessary to set AIN pins as input mode without pull-up resistor by program. Use CHS[3:0] to select AIN pin and GCHS enables global ADC channel, the analog signal inputs to ADC engine.

The ADC reference high voltage includes two source, one is internal Vdd (AVREFHSEL=0), and the other one is external reference voltage input pin from P2.0 pin (AVREFHSEL=1).

The ADC resolution can be selected 8-bit or 12-bit through ADLEN bit in ADR register. The ADC converting rate can be selected by ADCKS[1:0] bits. The two parameters decide ADC converting time.



- * **Note:** For 8-bit resolution the conversion time is 12 steps.
For 12-bit resolution the conversion time is 16 steps
- * **Note:** ADC_PCLK shall be less than 16MHz.
- * **Note:** The analog input level must be between the AVREFH and AVREFL.
- * **Note:** The AVREFH level must be between the AVDD and AVREFL + 2.0V.

- * **Note: ADC programming notice:**
- 1. Set ADC input pin I/O direction as input mode
 - 2. Disable pull-up resistor of ADC input pin
 - 3. Disable ADC (set ADENB = "0") before enter low-power (Sleep/Deep-sleep/Deep power-down) mode to save power consumption.
 - 4. Set related bit of P2CON register to avoid extra power consumption in power down mode.
 - 5. Delay 100us after enable ADC (set ADENB = "1") to wait ADC circuit ready for conversion.

6.2 ADC CONVERTING TIME

The ADC converting time is from ADS=1 (Start to ADC convert) to EOC=1 (End of ADC convert). The converting time duration is depend on ADC resolution and ADC clock rate.

ADC clock source is controlled by ADCKS[2:0] bits. The ADC converting time affects ADC performance. If input high rate analog signal, it is necessary to select a high ADC converting rate. If the ADC converting time is slower than analog signal variation rate, the ADC result would be error. So to select a correct ADC clock rate and ADC resolution to decide a right ADC converting rate is very important.

12-bit ADC conversion time = 1/(ADC clock /4)*16 sec

| ADLEN | ADCKS [2:0] | ADC Clock | ADC_PCLK = 4 MHz | | ADC_PCLK = 12 MHz | | ADC_PCLK = 16 MHz | |
|-------|-------------|-------------|--------------------------|---------------------------|--------------------------|---------------------------|--------------------------|---------------------------|
| | | | ADC Conversion Time (us) | ADC Conversion Rate (KHz) | ADC Conversion Time (us) | ADC Conversion Rate (KHz) | ADC Conversion Time (us) | ADC Conversion Rate (KHz) |
| 1 | 000 | ADC_PCLK | 16 | 62.5 | 5.33 | 187.5 | 4 | 250 |
| | 001 | ADC_PCLK/2 | 32 | 31.25 | 10.67 | 93.75 | 8 | 125 |
| | 010 | ADC_PCLK/4 | 64 | 15.625 | 21.33 | 46.875 | 16 | 62.5 |
| | 011 | ADC_PCLK/8 | 128 | 7.813 | 42.67 | 23.437 | 32 | 31.25 |
| | 100 | ADC_PCLK/16 | 256 | 3.906 | 85.3 | 11.718 | 64 | 15.625 |
| | 101 | ADC_PCLK/32 | 512 | 1.953 | 170.67 | 5.859 | 128 | 7.813 |

8-bit ADC conversion time = 1/(ADC clock /4)*12 sec

| ADLEN | ADCKS [2:0] | ADC Clock | ADC_PCLK = 4 MHz | | ADC_PCLK = 12 MHz | | ADC_PCLK = 16 MHz | |
|-------|-------------|-------------|--------------------------|---------------------------|--------------------------|---------------------------|--------------------------|---------------------------|
| | | | ADC Conversion Time (us) | ADC Conversion Rate (KHz) | ADC Conversion Time (us) | ADC Conversion Rate (KHz) | ADC Conversion Time (us) | ADC Conversion Rate (KHz) |
| 1 | 000 | ADC_PCLK | 16 | 62.5 | 5.33 | 187.5 | 4 | 250 |
| | 001 | ADC_PCLK/2 | 32 | 31.25 | 10.67 | 93.75 | 8 | 125 |
| | 010 | ADC_PCLK/4 | 64 | 15.625 | 21.33 | 46.875 | 16 | 62.5 |
| | 011 | ADC_PCLK/8 | 128 | 7.813 | 42.67 | 23.437 | 32 | 31.25 |
| | 100 | ADC_PCLK/16 | 256 | 3.906 | 85.3 | 11.718 | 64 | 15.625 |
| | 101 | ADC_PCLK/32 | 512 | 1.953 | 170.67 | 5.859 | 128 | 7.813 |

6.3 ADC CONTROL NOTICE

6.3.1 ADC SIGNAL

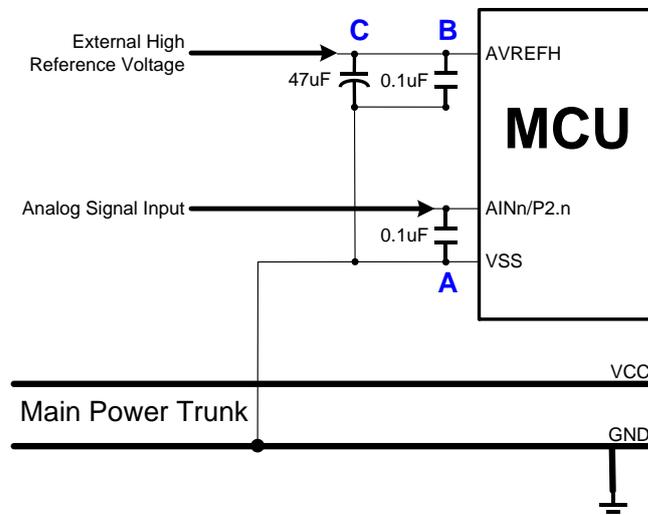
The ADC high reference voltage is internal Vdd or external voltage source. The ADC low reference voltage is ground. The ADC input signal voltage range must be from high reference voltage to low reference voltage. **The external high reference voltage from P2.0 must be higher than “Low reference voltage + 2V”. The low reference voltage is ground. So the external reference voltage range must be under 2V~Vdd.**

6.3.2 ADC PROGRAM

The first step of ADC execution is to setup ADC configuration. The ADC program setup sequence and notices are as following.

- **Step 1:** Enable ADC. ADENB is ADC control bit to control. ADENB = 1 is to enable ADC. ADENB = 0 is to disable ADC. **When ADENB is enabled, the system must be delay 100us to be the ADC warm-up time by program, and then set ADS to do ADC converting. The 100us delay time is necessary after ADENB setting (not ADS setting), or the ADC converting result would be error.** Normally, the ADENB is set one time when the system under normal run condition, and do the delay time only one time.
- **Step 2:** If the ADC high reference voltage is from external voltage source, set the AVREFHSEL = 1. The ADC external high reference voltage inputs from P2.0 pin. **It is necessary to set P2.0 as input mode without pull-up resistor.**
- **Step 3:** Select the ADC input pin by CHS[3:0], enable P2CON's related bit for the ADC input pin, and enable ADC global input. **When one AIN pin is selected to be analog signal input pin, it is necessary to setup the pin as input mode and disable the pull-up resistor by program. Also to set the P2CON, and the digital I/O function including pull-up is isolated.**
- **Step 4:** Start to execute ADC conversion by setting ADS = 1.
- **Step 5:** Wait the end of ADC converting through checking EOC = 1 or ADCIF = 1. If ADC interrupt function is enabled, the program executes ADC interrupt service when ADC interrupt occurrence. **ADS is cleared when the end of ADC converting automatically. EOC bit indicates ADC processing status immediately and is cleared when ADS = 1. Users needn't to clear it by program.**

6.4 ADC CIRCUIT



The analog signal is inputted to ADC input pin “AINn/P2.n”. The ADC input signal must be through a 0.1uF capacitor “A”. The 0.1uF capacitor is set between ADC input pin and VSS pin, and must be on the side of the ADC input pin as possible. Don't connect the capacitor's ground pin to ground plain directly, and must be through VSS pin. The capacitor can reduce the power noise effective coupled with the analog signal.

If the ADC high reference voltage is from external voltage source, the external high reference is connected to AVREFH pin (P2.0). The external high reference source must be through a 47uF “C” capacitor first, and then 0.1uF capacitor “B”.

These capacitors are set between AVREFH pin and VSS pin, and must be on the side of the AVREFH pin as possible. Don't connect the capacitor's ground pin to ground plain directly, and must be through VSS pin.

6.5 ADC REGISTERS

Base Address: 0x4002 6000

6.5.1 ADC Management register (ADC_ADM)

Address Offset: 0x00

| Bit | Name | Description | Attribute | Reset |
|-------|------------|---|-----------|-------|
| 31:13 | Reserved | | R | 0 |
| 12 | AVREFHSEL | ADC high reference voltage source select bit 0: Internal VDD. (P2.0 is GPIO or AIN0 pin) 1: Enable external reference voltage from P2.0 | R/W | 0 |
| 11 | ADENB | ADC Enable bit 0: Disable 1: Enable | R/W | 0 |
| 10:8 | ADCKS[2:0] | ADC Clock source divider 000: ADC_PCLK / 1 001: ADC_PCLK / 2 010: ADC_PCLK / 4 011: ADC_PCLK / 8 100: ADC_PCLK / 16 101: ADC_PCLK / 32 Other: Reversed | R/W | 0 |
| 7 | ADLEN | ADC resolution control bit. 0: 8-bit ADC. 1: 12-bit ADC. | R/W | 0 |
| 6 | ADS | ADC start control bit. 0: ADC converting stops. 1: Start to execute ADC converting. ADS is cleared when the end of ADC converting automatically. | R/W | 0 |
| 5 | EOC | ADC status bit indicates ADC processing status immediately and is cleared when ADS = 1. 0: ADC progressing. 1: End of converting and reset ADS bit. | R/W | 0 |
| 4 | GCHS | ADC global channel select bit. 0: Disable AIN channel 1: Enable AIN channel | R/W | 0 |
| 3:0 | CHS[3:0] | ADC input channels select bit. 0000: AIN0 0001: AIN1 0010: AIN2 0011: AIN3 0100: AIN4 0101: AIN5 0110: AIN6 0111: AIN7 1000: AIN8 1001: AIN9 Other: Reversed | R/W | 0 |

- * **Note: If ADENB = 1, users should set P2.n/AINn as input mode without pull-up. System doesn't set automatically. If P2CON[n] is set, the P2.n/AINn's digital I/O function including pull-up is isolated.**
- * **Note: If P2.0 is used as external reference voltage input pin, users should set P2.0 as input mode without pull-up, and set P2CON[0] bit to 1.**

6.5.2 ADC Data register (ADC_ADB)

Address Offset: 0x04

ADB is ADC data buffer to store AD converter result.

| Bit | Name | Description | Attribute | Reset |
|-------|-----------|---|-----------|-------|
| 31:12 | Reserved | | R | 0 |
| 11:0 | ADB[11:0] | ADB11~ADB4 bits for 8-bit ADC ADB11~ADB0 bits for 12-bit ADC | R | 0 |

The AIN's input voltage v.s. ADB's output data

| AIN n | ADB11 | ADB10 | ADB9 | ADB8 | ADB7 | ADB6 | ADB5 | ADB4 | ADB3 | ADB2 | ADB1 | ADB0 |
|-----------------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| 0/4096*VREFH | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1/4096*VREFH | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| . | . | . | . | . | . | . | . | . | . | . | . | . |
| . | . | . | . | . | . | . | . | . | . | . | . | . |
| 4094/4096*VREFH | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 4095/4096*VREFH | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

For different applications, users maybe need more than 8-bit resolution but less than 12-bit ADC converter. First, the AD resolution must be set 12-bit mode and then to execute ADC converter routine. Then delete the LSB of ADC data and get the new resolution result. The table is as following.

| | ADB11 | ADB10 | ADB9 | ADB8 | ADB7 | ADB6 | ADB5 | ADB4 | ADB3 | ADB2 | ADB1 | ADB0 |
|---------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| 8-bit | O | O | O | O | O | O | O | O | X | X | X | X |
| 9-bit | O | O | O | O | O | O | O | O | O | X | X | X |
| 10-bit | O | O | O | O | O | O | O | O | O | O | X | X |
| 11-bit | O | O | O | O | O | O | O | O | O | O | O | X |
| 12-bit. | O | O | O | O | O | O | O | O | O | O | O | O |

O = Selected, X = Delete

*** Note: The initial value of ADC buffer (ADB) after reset is unknown.**

6.5.3 Port 2 Control register (ADC_P2CON)

Address Offset: 0x08

The Port 2 is shared with ADC input function. Only one pin of port 2 can be configured as ADC input in the same time by ADM register. The other pins of port 2 are digital I/O pins.

Connect an analog signal to COMS digital input pin, especially, the analog signal level is about 1/2 VDD will cause extra current leakage. In the power down mode, the above leakage current will be a big problem. Unfortunately, if users connect more than one analog input signal to port 2 will encounter above current leakage situation.

P2CON is Port2 Configuration register. Write "1" into P2CON [9:0] will configure related port 2 pin as pure analog input pin to avoid current leakage.

| Bit | Name | Description | Attribute | Reset |
|-------|------------|---|-----------|-------|
| 31:10 | Reserved | | R | 0 |
| 9:0 | P2CON[9:0] | P2.x configuration control bits. (x=0 to 9) 0: P2.x can be an analog input (ADC input) or digital I/O pins. 1: P2.x is pure analog input, can't be a digital I/O pin. | R/W | 0 |

* **Note:** When Port 2.n is general I/O port not ADC channel, P2CON.n must set to "0" or the Port 2.n digital I/O signal would be isolated.

6.5.4 ADC Interrupt Enable register (ADC_IE)

Address offset: 0x0C

This register allows control over which A/D channels generate an interrupt when a conversion is complete. For example, it may be desirable to use some A/D channels to monitor sensors by continuously performing conversions on them. The most recent results are read by the application program whenever they are needed. In this case, an interrupt is not desirable at the end of each conversion for some A/D channels.

| Bit | Name | Description | Attribute | Reset |
|-------|----------|--|-----------|-------|
| 31:10 | Reserved | | R | 0 |
| 9:0 | IE[9:0] | These bits allow control over which A/D channels generate interrupts for conversion completion. When bit x is one, completion of a conversion on AIN x will generate an interrupt. | R/W | 0 |

6.5.5 ADC Raw Interrupt Status register (ADC_RIS)

Address offset: 0x10

| Bit | Name | Description | Attribute | Reset |
|-------|----------|---|-----------|-------|
| 31:10 | Reserved | | R | 0 |
| 9:0 | IF[9:0] | ADC raw interrupt flag. (x = 0 to 9). 0: Read→No interrupt on AINx Write→Write "0" to the corresponding bit will clear the bit and reset the Interrupt if the corresponding IE bit is set. 1: Interrupt requirements met on AINx ADC conversion. | R/W | 0 |

7 16-BIT TIMER WITH CAPTURE FUNCTION

7.1 OVERVIEW

Each Counter/timer is designed to count cycles of the peripheral clock (PCLK) or an externally supplied clock and can optionally generate interrupts or perform other actions at specified timer values based on four match registers. Each counter/timer also includes one capture input to trap the timer value when an input signal transitions, optionally generating an interrupt.

In PWM mode, up to three match registers can be used to provide a single-edge controlled PWM output on the match output pins.

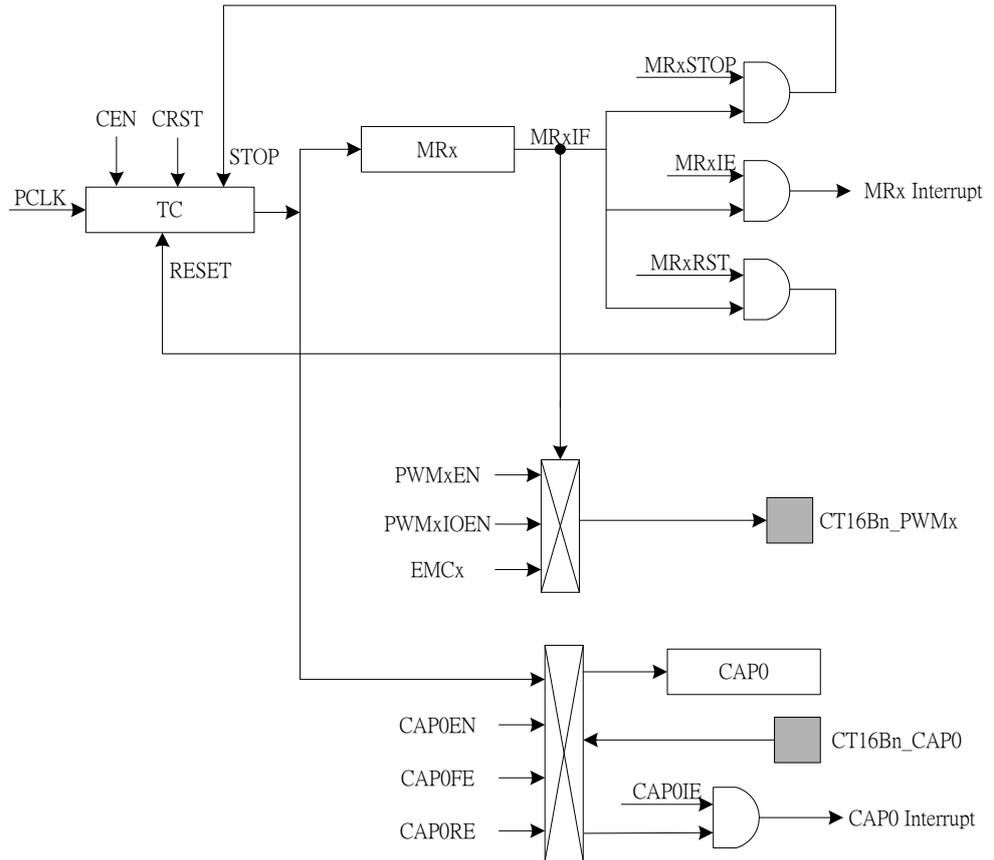
7.2 FEATURES

- Two 16-bit counter/timers.
- Counter or timer operation
- Two 16-bit capture channels that can take a snapshot of the timer value when an input signal transitions. A capture event may also optionally generate an interrupt.
- The timer value may be configured to be cleared on a designated capture event. This feature permits easy pulse-width measurement by clearing the timer on the leading edge of an input pulse and capturing the timer value on the trailing edge.
- For each timer, four 16-bit match registers that allow:
 - Continuous operation with optional interrupt generation on match.
 - Stop timer on match with optional interrupt generation.
 - Reset timer on match with optional interrupt generation.
 - Configured as PWM allowing using up to three match outputs as single edge controlled PWM outputs.
- For each timer, up to three PWM outputs corresponding to match registers with the following capabilities:
 - Set LOW on match.
 - Set HIGH on match.
 - Toggle on match.
 - Do nothing on match.

7.3 PIN DESCRIPTION

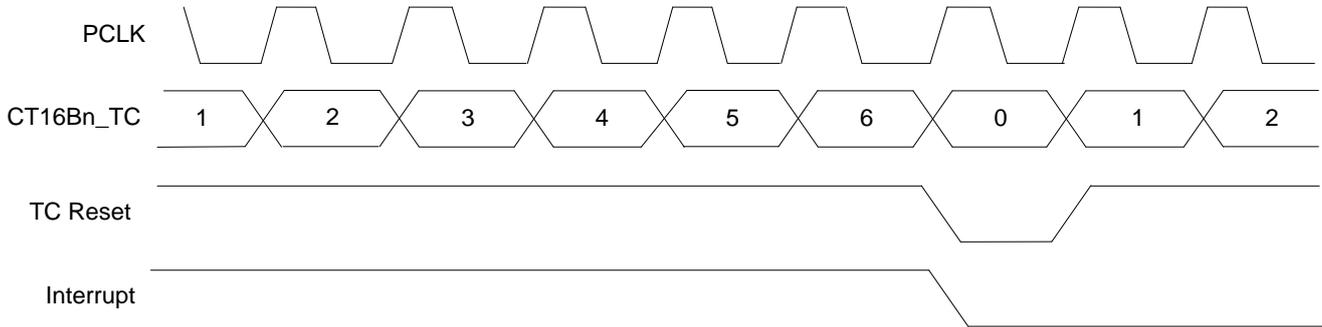
| Pin Name | Type | Description | GPIO Configuration |
|-------------|------|---------------------------------------|-----------------------------------|
| CT16Bn_CAP0 | I | Capture channel input 0 | Depends on GPIO _n _CFG |
| CT16Bn_PWMx | O | Output channel x of Match/PWM output. | |

7.4 BLOCK DIAGRAM

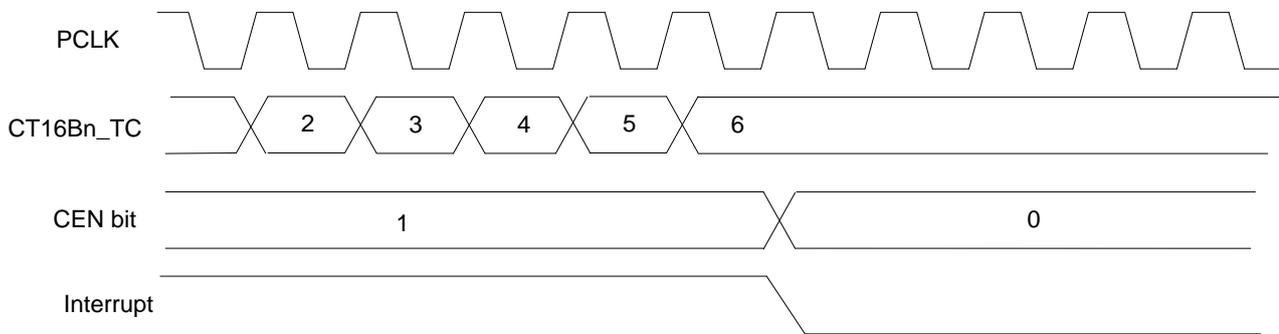


7.5 TIMER OPERATION

The following figure shows a timer configured to reset the count and generate an interrupt on match. The [CT16Bn_MRx](#) register is set to 6. At the end of the timer cycle where the match occurs, the timer count is reset. This gives a full length cycle to the match value. The interrupt indicating that a match occurred is generated in the next clock after the timer reached the match value.

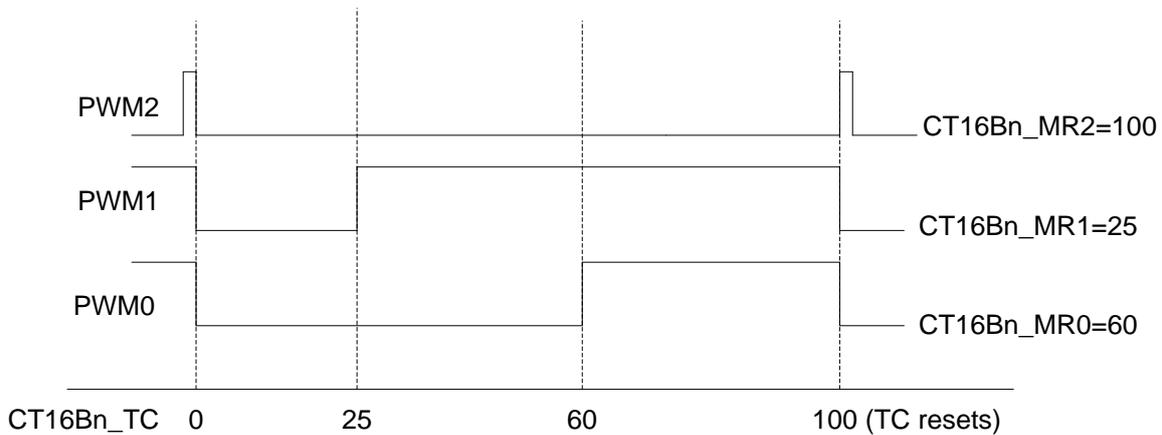


The following figure shows a timer configured to stop and generate an interrupt on match. The [CT16Bn_MRx](#) register is set to 6. In the next clock after the timer reaches the match value, the CEN bit in [CT16Bn_TMRCTRL](#) register is cleared, and the interrupt indicating that a match occurred is generated.



7.6 PWM

1. All single edge controlled PWM outputs go LOW at the beginning of each PWM cycle (timer is set to zero) unless their match value in CT16Bn_MR0~3 registers is equal to zero.
2. Each PWM output will go HIGH when its match value is reached. If no match occurs, the PWM output remains continuously LOW.
3. If a match value larger than the PWM cycle length is written to the CT16Bn_MR0~3 registers, and the PWM signal is HIGH already, then the PWM signal will be cleared on the next start of the next PWM cycle.
4. If a match register contains the same value as the timer reset value (the PWM cycle length), then the PWM output will be reset to LOW on the next clock tick. Therefore, the PWM output will always consist of a one clock tick wide positive pulse with a period determined by the PWM cycle length.
5. If a match register is set to zero, then the PWM output will go to HIGH the first time the timer goes back to zero and will stay HIGH continuously.



* **Note:** When the match outputs are selected to perform as PWM outputs, the timer reset (MRnRST) and timer stop (MRnSTOP) bits in [CT16Bn_MCTRL](#) register must be set to zero except for the match register setting the PWM cycle length. For this register, set the MRnR bit to one to enable the timer reset when the timer value matches the value of the corresponding match register.

7.7 CT16Bn REGISTERS

Base Address: 0x4000 0000 (CT16B0)
0x4000 2000 (CT16B1)

7.7.1 CT16Bn Timer Control register (CT16Bn_TMRCTRL) (n=0,1)

Address Offset: 0x00

*** Note: CEN bit shall be set at last!**

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:2 | Reserved | | R | 0 |
| 1 | CRST | Counter Reset. 0: Disable counter reset. 1: Timer Counter is synchronously reset on the next positive edge of PCLK. This is cleared by HW when the counter reset operation finishes. | R/W | 0 |
| 0 | CEN | Counter Enable 0: Disable Counter. 1: Enable Timer Counter for counting. | R/W | 0 |

7.7.2 CT16Bn Timer Counter register (CT16Bn_TC) (n=0,1)

Address Offset: 0x04

Unless it is reset before reaching its upper limit, the TC will count up to the value 0x0000FFFF and then wrap back to the value 0x00000000. This event does not cause an interrupt, but a Match register can be used to detect an overflow if needed.

| Bit | Name | Description | Attribute | Reset |
|-------|----------|---------------|-----------|-------|
| 31:16 | Reserved | | R | 0 |
| 15:0 | TC[15:0] | Timer Counter | R/W | 0 |

7.7.3 CT16Bn Count Control register (CT16Bn_CNTCTRL) (n=0,1)

Address Offset: 0x10

This register is used to select between Timer and Counter mode, and in Counter mode to select the pin and edges for counting.

When Counter Mode is chosen as a mode of operation, the CAP input (selected by the CIS bits) is sampled on every rising edge of the PCLK clock. After comparing two consecutive samples of this CAP input, one of the following four events is recognized: rising edge, falling edge, either of edges or no changes in the level of the selected CAP input. Only if the identified event occurs, and the event corresponds to the one selected by CTM bits in this register, will the Timer Counter register be incremented.

Effective processing of the externally supplied clock to the counter has some limitations. Since two successive rising edges of the PCLK clock are used to identify only one edge on the CAP selected input, the frequency of the CAP input cannot exceed one half of the PCLK clock. Consequently, the duration of the HIGH/LOW levels on the same CAP input in this case cannot be shorter than $1/(2 \times \text{PCLK})$.

*** Note: If Counter mode is selected in the CNTCTRL register, Capture Control (CAPCTRL) register must be**

programmed as 0x0.

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:4 | Reserved | | R | 0 |
| 3:2 | CIS[1:0] | Count Input Select. In counter mode (when CTM[1:0] are not 00), these bits select which CAP0 pin is sampled for clocking. 00: CT16Bn_CAP0 Other: Reserved. | R/W | 0 |
| 1:0 | CTM[1:0] | Counter/Timer Mode. This field selects which rising PCLK edges can clear PC and increment Timer Counter (TC). 00: Timer Mode: every rising PCLK edge 01: Counter Mode: TC is incremented on rising edges on the CAP0 input selected by CIS bits. 10: Counter Mode: TC is incremented on falling edges on the CAP0 input selected by CIS bits. 11: Counter Mode: TC is incremented on both edges on the CAP0 input selected by CIS bits. | R/W | 0 |

7.7.4 CT16Bn Match Control register (CT16Bn_MCTRL) (n=0,1)

Address Offset: 0x14

| Bit | Name | Description | Attribute | Reset |
|-------|----------|--|-----------|-------|
| 31:12 | Reserved | | R | 0 |
| 11 | MR3STOP | Stop MR3: TC will stop and CEN bit will be cleared if MR3 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 10 | MR3RST | Enable reset TC when MR3 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 9 | MR3IE | Enable generating an interrupt when MR3 matches the value in the TC. 0: Disable 1: Enable | R/W | 0 |
| 8 | MR2STOP | Stop MR2: TC will stop and CEN bit will be cleared if MR2 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 7 | MR2RST | Enable reset TC when MR2 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 6 | MR2IE | Enable generating an interrupt when MR2 matches the value in the TC. 0: Disable 1: Enable | R/W | 0 |
| 5 | MR1STOP | Stop MR1: TC will stop and CEN bit will be cleared if MR1 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 4 | MR1RST | Enable reset TC when MR1 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 3 | MR1IE | Enable generating an interrupt when MR1 matches the value in the TC. 0: Disable 1: Enable | R/W | 0 |
| 2 | MR0STOP | Stop MR0: TC will stop and CEN bit will be cleared if MR0 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 1 | MR0RST | Enable reset TC when MR0 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 0 | MR0IE | Enable generating an interrupt when MR0 matches the value in the TC. 0: Disable | R/W | 0 |

| | | | |
|--|-----------|--|--|
| | 1: Enable | | |
|--|-----------|--|--|

7.7.5 CT16Bn Match register 0~3 (CT16Bn_MR0~3) (n=0,1)

Address Offset: 0x18, 0x1C, 0x20, 0x24

The Match register values are continuously compared to the Timer Counter (TC) value. When the two values are equal, actions can be triggered automatically. The action possibilities are to generate an interrupt, reset the Timer Counter, or stop the timer. Actions are controlled by the settings in the CT16Bn_MCTRL register.

| Bit | Name | Description | Attribute | Reset |
|-------|----------|---------------------------|-----------|-------|
| 31:16 | Reserved | | R | 0 |
| 15:0 | MR[15:0] | Timer counter match value | R/W | 0 |

7.7.6 CT16Bn Capture Control register (CT16Bn_CAPCTRL) (n=0,1)

Address Offset: 0x28

The Capture Control register is used to control whether the Capture register is loaded with the value in the Counter/timer when the capture event occurs, and whether an interrupt is generated by the capture event. Setting both the rising and falling bits at the same time is a valid configuration, resulting in a capture event for both edges.

*** Note: HW will switch I/O Configuration directly when CAP0EN=1.**

| Bit | Name | Description | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:4 | Reserved | | R | 0 |
| 3 | CAP0EN | Capture 0 function enable bit 0: Disable 1: Enable. | R/W | 0 |
| 2 | CAP0IE | Interrupt on CT16Bn_CAP0 event: a CAP0 load due to a CT16Bn_CAP0 event will generate an interrupt. 0: Disable 1: Enable | R/W | 0 |
| 1 | CAP0FE | Capture on CT16Bn_CAP0 falling edge: a sequence of 1 then 0 on CT16Bn_CAP0 will cause CAP0 to be loaded with the contents of TC. 0: Disable 1: Enable | R/W | 0 |
| 0 | CAP0RE | Capture on CT16Bn_CAP0 rising edge: a sequence of 0 then 1 on CT16Bn_CAP0 will cause CAP0 to be loaded with the contents of TC. 0: Disable 1: Enable | R/W | 0 |

7.7.7 CT16Bn Capture 0 register (CT16Bn_CAP0) (n=0,1)

Address Offset: 0x2C

Each Capture register is associated with a device pin and may be loaded with the counter/timer value when a specified event occurs on that pin. The settings in the Capture Control register determine whether the capture function is enabled, and whether a capture event happens on the rising edge of the associated pin, the falling edge, or on both edges.

| Bit | Name | Description | Attribute | Reset |
|-------|------------|-----------------------------|-----------|-------|
| 31:16 | Reserved | | R | 0 |
| 15:0 | CAP0[15:0] | Timer counter capture value | R | 0 |

7.7.8 CT16Bn External Match register (CT16Bn_EM) (n=0)

Address Offset: 0x30

The External Match register provides both control and status of CT16B0_PWM[2:0]. If the match outputs are configured as PWM output, the function of the external match registers is determined by the [PWM rules](#).

| Bit | Name | Description | Attribute | Reset |
|-------|-----------|--|-----------|-------|
| 31:10 | Reserved | | R | 0 |
| 9:8 | EMC2[1:0] | Determines the functionality of CT16Bn_PWM2. 00: Do Nothing. 01: CT16Bn_PWM2 pin is LOW 10: CT16Bn_PWM2 pin is HIGH 11: Toggle CT16Bn_PWM2 pin | R/W | 0 |
| 7:6 | EMC1[1:0] | Determines the functionality of CT16Bn_PWM1. 00: Do Nothing. 01: CT16Bn_PWM1 pin is LOW 10: CT16Bn_PWM1 pin is HIGH 11: Toggle CT16Bn_PWM1 | R/W | 0 |
| 5:4 | EMC0[1:0] | Determines the functionality of CT16Bn_PWM0. 00: Do Nothing. 01: CT16Bn_PWM0 pin is LOW 10: CT16Bn_PWM0 pin is HIGH 11: Toggle CT16Bn_PWM0 | R/W | 0 |
| 3 | Reserved | | R | 0 |
| 2 | EM2 | When the TC and MR2 are equal, this bit will act according to EMC2 bits, and also drive the state of CT16Bn_PWM2 output. | R/W | 0 |
| 1 | EM1 | When the TC and MR1 are equal, this bit will act according to EMC1 bits, and also drive the state of CT16Bn_PWM1 output. | R/W | 0 |
| 0 | EM0 | When the TC and MR0 are equal, this bit will act according to EMC0 bits, and also drive the state of CT16Bn_PWM0 output. | R/W | 0 |

7.7.9 CT16Bn External Match register (CT16Bn_EM) (n=1)

Address Offset: 0x30

The External Match register provides both control and status of CT16B1_PWM[1:0]. If the match outputs are configured as PWM output, the function of the external match registers is determined by the [PWM rules](#).

| Bit | Name | Description | Attribute | Reset |
|------|-----------|--|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7:6 | EMC1[1:0] | Determines the functionality of CT16Bn_PWM1. 00: Do Nothing. 01: CT16Bn_PWM1 pin is LOW 10: CT16Bn_PWM1 pin is HIGH. 11: Toggle CT16Bn_PWM1. | R/W | 0 |
| 5:4 | EMC0[1:0] | Determines the functionality of CT16Bn_PWM0. 00: Do Nothing. 01: CT16Bn_PWM0 pin is LOW 10: CT16Bn_PWM0 pin is HIGH 11: Toggle CT16Bn_PWM0. | R/W | 0 |
| 3:2 | Reserved | | R | 0 |
| 1 | EM1 | When the TC and MR1 are equal, this bit will act according to EMC1 bits, and also drive the state of CT16Bn_PWM1 output. | R/W | 0 |
| 0 | EM0 | When the TC and MR0 are equal, this bit will act according to EMC0 bits, and also drive the state of CT16Bn_PWM0 output. | R/W | 0 |

7.7.10 CT16Bn PWM Control register (CT16Bn_PWMCTRL) (n=0)

Address Offset: 0x34

The PWM Control register is used to configure the match outputs as PWM outputs. Each match output can be in-dependently set to perform either as PWM output or as match output whose function is controlled by [CT16Bn_EM](#) register.

For CT16B0, a maximum of 3 single edge controlled PWM outputs can be selected on the CT16B0_PWMCTRL[2:0] outputs. One additional match register determines the PWM cycle length. When a match occurs in any of the other match registers, the PWM output is set to HIGH. The timer is reset by the match register that is configured to set the PWM cycle length. When the timer is reset to zero, all currently HIGH match outputs configured as PWM outputs are cleared.

| Bit | Name | Description | Attribute | Reset |
|-------|----------|--|-----------|-------|
| 31:23 | Reserved | | R | 0 |
| 22 | PWM2IOEN | CT16Bn_PWM2/GPIO selection bit 0: CT16Bn_PWM2 pin act as GPIO 1: CT16Bn_PWM2 pin act as match output, and output signal depends on PWM2EN bit. | R/W | 0 |
| 21 | PWM1IOEN | CT16Bn_PWM1/GPIO selection bit 0: CT16Bn_PWM1 pin act as GPIO 1: CT16Bn_PWM1 pin act as match output, and output signal depends on PWM1EN bit. | R/W | 0 |
| 20 | PWM0IOEN | CT16Bn_PWM0/GPIO selection bit 0: CT16Bn_PWM0 pin act as GPIO 1: CT16Bn_PWM0 pin act as match output, and output signal depends on PWM0EN bit. | R/W | 0 |
| 19:3 | Reserved | | R | 0 |
| 2 | PWM2EN | PWM2 enable 0: CT16Bn_PWM2 is controlled by EM2. 1: PWM mode is enabled for CT16Bn_PWM2. | R/W | 0 |
| 1 | PWM1EN | PWM1 enable 0: CT16Bn_PWM1 is controlled by EM1. 1: PWM mode is enabled for CT16Bn_PWM1. | R/W | 0 |
| 0 | PWM0EN | PWM0 enable 0: CT16Bn_PWM0 is controlled by EM0. 1: PWM mode is enabled for CT16Bn_PWM0. | R/W | 0 |

7.7.11 CT16Bn PWM Control register (CT16Bn_PWMCTRL) (n=1)

Address Offset: 0x34

The PWM Control register is used to configure the match outputs as PWM outputs. Each match output can be in-dependently set to perform either as PWM output or as match output whose function is controlled by [CT16Bn_EM](#) register.

For CT16B1, a maximum of 2 single edge controlled PWM outputs can be selected on the CT16B1_PWMCTRL[1:0] outputs. One additional match register determines the PWM cycle length. When a match occurs in any of the other match registers, the PWM output is set to HIGH. The timer is reset by the match register that is configured to set the PWM cycle length. When the timer is reset to zero, all currently HIGH match outputs configured as PWM outputs are cleared.

| Bit | Name | Description | Attribute | Reset |
|-------|----------|--|-----------|-------|
| 31:22 | Reserved | | R | 0 |
| 21 | PWM1IOEN | CT16Bn_PWM1/GPIO selection bit 0: CT16Bn_PWM1 pin act as GPIO 1: CT16Bn_PWM1 pin act as match output, and output signal depends on PWM1EN bit. | R/W | 0 |
| 20 | PWM0IOEN | CT16Bn_PWM0/GPIO selection bit 0: CT16Bn_PWM0 pin act as GPIO 1: CT16Bn_PWM0 pin act as match output, and output signal depends on PWM0EN bit. | R/W | 0 |

| | | | | |
|-------------|----------|--|-----|---|
| 19:2 | Reserved | | R | 0 |
| 1 | PWM1EN | PWM1 enable 0: CT16Bn_PWM1 is controlled by EM1. 1: PWM mode is enabled for CT16Bn_PWM1. | R/W | 0 |
| 0 | PWM0EN | PWM0 enable 0: CT16Bn_PWM0 is controlled by EM0. 1: PWM mode is enabled for CT16Bn_PWM0. | R/W | 0 |

7.7.12 CT16Bn Timer Raw Interrupt Status register (CT16Bn_RIS) (n=0,1)

Address Offset: 0x38

This register indicates the raw status for Timer/PWM interrupts. A Timer/PWM interrupt is sent to the interrupt controller if the corresponding bit in the CT16Bn_IE register is set.

| Bit | Name | Description | Attribute | Reset |
|-------------|----------|--|-----------|-------|
| 31:5 | Reserved | | R | 0 |
| 4 | CAP0IF | Interrupt flag for capture channel 0. 0: No interrupt on CAP0 1: Interrupt requirements met on CAP0. | R | 0 |
| 3 | MR3IF | Interrupt flag for match channel 3. 0: No interrupt on match channel 3 1: Interrupt requirements met on match channel 3. | R | 0 |
| 2 | MR2IF | Interrupt flag for match channel 2. 0: No interrupt on match channel 2 1: Interrupt requirements met on match channel 2. | R | 0 |
| 1 | MR1IF | Interrupt flag for match channel 1. 0: No interrupt on match channel 1 1: Interrupt requirements met on match channel 1. | R | 0 |
| 0 | MR0IF | Interrupt flag for match channel 0. 0: No interrupt on match channel 0 1: Interrupt requirements met on match channel 0. | R | 0 |

7.7.13 CT16Bn Timer Interrupt Clear register (CT16Bn_IC) (n=0,1)

Address Offset: 0x3C

| Bit | Name | Description | Attribute | Reset |
|-------------|----------|-------------------------------------|-----------|-------|
| 31:5 | Reserved | | R | 0 |
| 4 | CAP0IC | 0: No effect 1: Clear CAP0IF bit | W | 0 |
| 3 | MR3IC | 0: No effect 1: Clear MR3IF bit | W | 0 |
| 2 | MR2IC | 0: No effect 1: Clear MR2IF bit | W | 0 |
| 1 | MR1IC | 0: No effect 1: Clear MR1IF bit | W | 0 |
| 0 | MR0IC | 0: No effect 1: Clear MR0IF bit | W | 0 |

8 32-BIT TIMER WITH CAPTURE FUNCTION

8.1 OVERVIEW

Each Counter/timer is designed to count cycles of the peripheral clock (PCLK) or an externally supplied clock and can optionally generate interrupts or perform other actions at specified timer values based on four match registers. Each counter/timer also includes one capture input to trap the timer value when an input signal transitions, optionally generating an interrupt.

In PWM mode, up to four match registers can be used to provide a single-edge controlled PWM output on the match output pins.

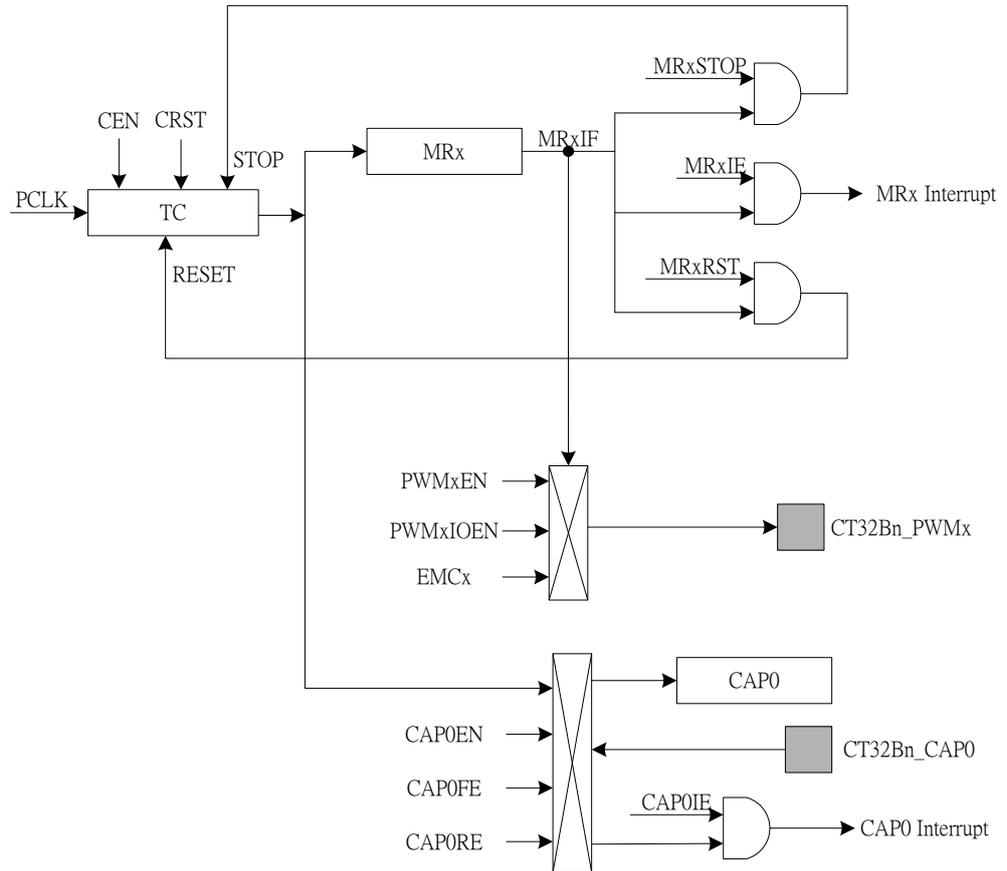
8.2 FEATURES

- Two 32-bit counter/timers
- Counter or timer operation
- Two 32-bit capture channels that can take a snapshot of the timer value when an input signal transitions. A capture event may also optionally generate an interrupt.
- The timer value may be configured to be cleared on a designated capture event. This feature permits easy pulse-width measurement by clearing the timer on the leading edge of an input pulse and capturing the timer value on the trailing edge.
- For each timer, four 32-bit match registers that allow:
 - Continuous operation with optional interrupt generation on match.
 - Stop timer on match with optional interrupt generation.
 - Reset timer on match with optional interrupt generation.
 - Configured as PWM allowing using up to four match outputs as single edge controlled PWM outputs.
- For each timer, up to four PWM outputs corresponding to match registers with the following capabilities:
 - Set LOW on match.
 - Set HIGH on match.
 - Toggle on match.
 - Do nothing on match.

8.3 PIN DESCRIPTION

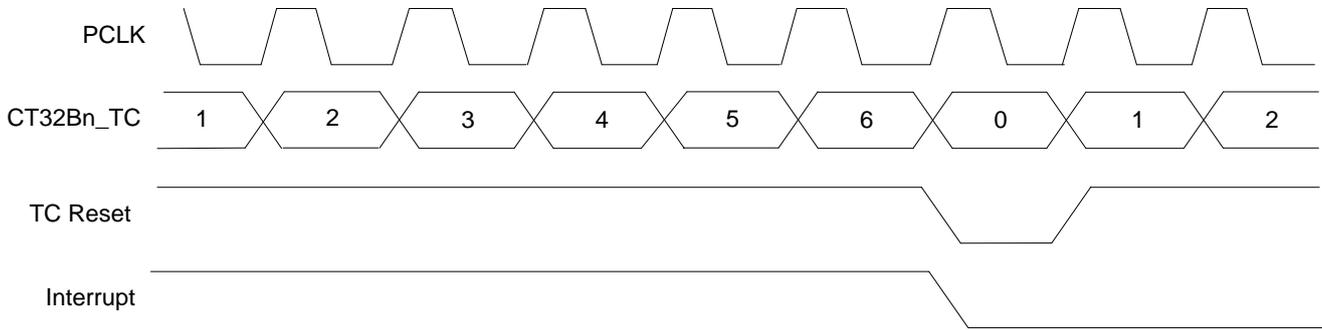
| Pin Name | Type | Description | GPIO Configuration |
|-------------|------|---------------------------------------|-----------------------------------|
| CT32Bn_CAP0 | I | Capture channel input 0 | Depends on GPIO _n _CFG |
| CT32Bn_PWMx | O | Output channel x of Match/PWM output. | |

8.4 BLOCK DIAGRAM

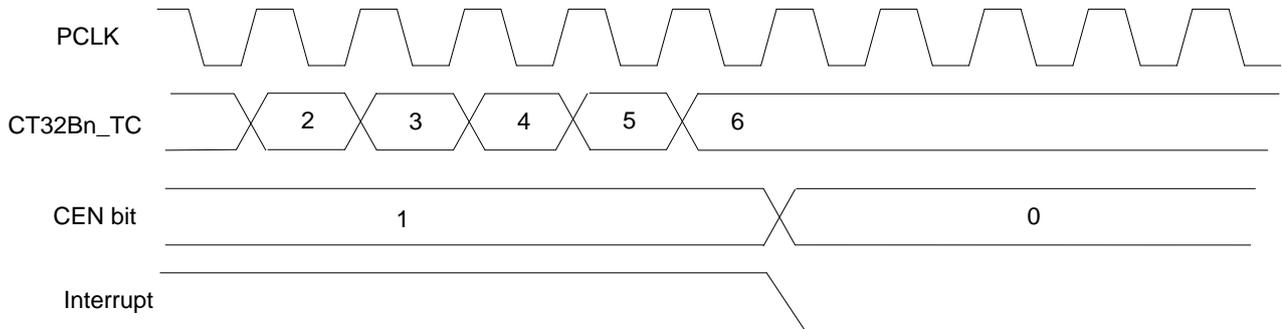


8.5 TIMER OPERATION

The following figure shows a timer configured to reset the count and generate an interrupt on match. The [CT32Bn_MRx](#) register is set to 6. At the end of the timer cycle where the match occurs, the timer count is reset. This gives a full length cycle to the match value. The interrupt indicating that a match occurred is generated in the next clock after the timer reached the match value.

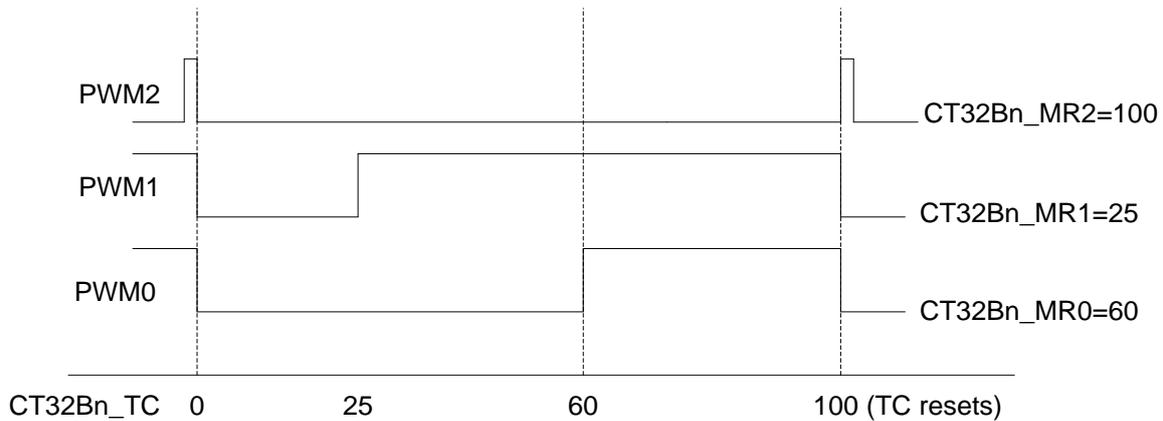


The following figure shows a timer configured to stop and generate an interrupt on match. The [CT32Bn_MRx](#) register is set to 6. In the next clock after the timer reaches the match value, the CEN bit in [CT32Bn_TMRCTRL](#) register is cleared, and the interrupt indicating that a match occurred is generated.



8.6 PWM

1. All single edge controlled PWM outputs go LOW at the beginning of each PWM cycle (timer is set to zero) unless their match value in CT32Bn_MR0~3 registers is equal to zero.
2. Each PWM output will go HIGH when its match value is reached. If no match occurs, the PWM output remains continuously LOW.
3. If a match value larger than the PWM cycle length is written to the CT32Bn_MR0~3 registers, and the PWM signal is HIGH already, then the PWM signal will be cleared on the next start of the next PWM cycle.
4. If a match register contains the same value as the timer reset value (the PWM cycle length), then the PWM output will be reset to LOW on the next clock tick. Therefore, the PWM output will always consist of a one clock tick wide positive pulse with a period determined by the PWM cycle length.
5. If a match register is set to zero, then the PWM output will go to HIGH the first time the timer goes back to zero and will stay HIGH continuously.



* **Note:** When the match outputs are selected to perform as PWM outputs, the timer reset (MRnRST) and timer stop (MRnSTOP) bits in [CT32Bn_MCTRL](#) register must be set to zero except for the match register setting the PWM cycle length. For this register, set the MRnR bit to one to enable the timer reset when the timer value matches the value of the corresponding match register.

8.7 CT32Bn REGISTERS

Base Address: 0x4000 4000 (CT32B0)
0x4000 6000 (CT32B1)

8.7.1 CT32Bn Timer Control register (CT32Bn_TMRCTRL) (n=0,1)

Address Offset: 0x00

* **Note:** *CEN bit shall be set at last!*

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:2 | Reserved | | R | 0 |
| 1 | CRST | Counter Reset. 0: Disable counter reset. 1: Timer Counter is synchronously reset on the next positive edge of PCLK. This is cleared by HW when the counter reset operation finishes. | R/W | 0 |
| 0 | CEN | Counter Enable 0: Disable Counter. 1: Enable Timer Counter for counting. | R/W | 0 |

8.7.2 CT32Bn Timer Counter register (CT32Bn_TC) (n=0,1)

Address Offset: 0x04

Unless it is reset before reaching its upper limit, the TC will count up through the value 0xFFFFFFFF and then wrap back to the value 0x00000000. This event does not cause an interrupt, but a Match register can be used to detect an overflow if needed.

| Bit | Name | Description | Attribute | Reset |
|------|----------|---------------|-----------|-------|
| 31:0 | TC[31:0] | Timer Counter | R/W | 0 |

8.7.3 CT32Bn Count Control register (CT16Bn_CNTCTRL) (n=0,1)

Address Offset: 0x10

This register is used to select between Timer and Counter mode, and in Counter mode to select the pin and edges for counting.

When Counter Mode is chosen as a mode of operation, the CAP input (selected by CIS bits) is sampled on every rising edge of the PCLK clock. After comparing two consecutive samples of this CAP input, one of the following four events is recognized: rising edge, falling edge, either of edges or no changes in the level of the selected CAP input. Only if the identified event occurs, and the event corresponds to the one selected by CTM bits in this register, will the Timer Counter register be incremented.

Effective processing of the externally supplied clock to the counter has some limitations. Since two successive rising edges of the PCLK clock are used to identify only one edge on the CAP selected input, the frequency of the CAP input cannot exceed one half of the PCLK clock. Consequently, the duration of the HIGH/LOW levels on the same CAP input in this case cannot be shorter than $1 / (2 \times \text{PCLK})$.

* **Note:** *If Counter mode is selected in the CNTCTRL register, Capture Control (CAPCTRL) register must be programmed as 0x0.*

| Bit | Name | Description | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:4 | Reserved | | R | 0 |
| 3:2 | CIS[1:0] | Count Input Select. In counter mode (when CTM[1:0] are not 00), these bits select which CAP pin is sampled for clocking. 00: CT32Bn_CAP0 Other: Reserved. | R/W | 0 |
| 1:0 | CTM[1:0] | Counter/Timer Mode. This field selects which rising PCLK edges can clear PC and increment Timer Counter (TC). 00: Timer Mode: every rising PCLK edge 01: Counter Mode: TC is incremented on rising edges on the CAP input selected by CIS bits. 10: Counter Mode: TC is incremented on falling edges on the CAP input selected by CIS bits. 11: Counter Mode: TC is incremented on both edges on the CAP input selected by CIS bits. | R/W | 0 |

8.7.4 CT32Bn Match Control register (CT32Bn_MCTRL) (n=0,1)

Address Offset: 0x14

| Bit | Name | Description | Attribute | Reset |
|-------|----------|--|-----------|-------|
| 31:12 | Reserved | | R | 0 |
| 11 | MR3STOP | Stop MR3: TC will stop and CEN bit will be cleared if MR3 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 10 | MR3RST | Enable reset TC when MR3 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 9 | MR3IE | Enable generating an interrupt when MR3 matches the value in the TC. 0: Disable 1: Enable | R/W | 0 |
| 8 | MR2STOP | Stop MR2: TC will stop and CEN bit will be cleared if MR2 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 7 | MR2RST | Enable reset TC when MR2 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 6 | MR2IE | Enable generating an interrupt when MR2 matches the value in the TC. 0: Disable 1: Enable | R/W | 0 |
| 5 | MR1STOP | Stop MR1: TC will stop and CEN bit will be cleared if MR1 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 4 | MR1RST | Enable reset TC when MR1 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 3 | MR1IE | Enable generating an interrupt when MR1 matches the value in the TC. 0: Disable 1: Enable | R/W | 0 |
| 2 | MR0STOP | Stop MR0: TC will stop and CEN bit will be cleared if MR0 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 1 | MR0RST | Enable reset TC when MR0 matches TC. 0: Disable 1: Enable | R/W | 0 |
| 0 | MR0IE | Enable generating an interrupt when MR0 matches the value in the TC. 0: Disable 1: Enable | R/W | 0 |

8.7.5 CT32Bn Match register 0~3 (CT32Bn_MR0~3) (n=0,1)

Address Offset: 0x18, 0x1C, 0x20, 0x24

The Match register values are continuously compared to the Timer Counter (TC) value. When the two values are equal, actions can be triggered automatically. The action possibilities are to generate an interrupt, reset the Timer Counter, or stop the timer. Actions are controlled by the settings in the CT32Bn_MCTRL register.

| Bit | Name | Description | Attribute | Reset |
|------|----------|---------------------------|-----------|-------|
| 31:0 | MR[31:0] | Timer counter match value | R/W | 0 |

8.7.6 CT32Bn Capture Control register (CT32Bn_CAPCTRL) (n=0,1)

Address Offset: 0x28

The Capture Control register is used to control whether the Capture register is loaded with the value in the Counter/timer when the capture event occurs, and whether an interrupt is generated by the capture event. Setting both the rising and falling bits at the same time is a valid configuration, resulting in a capture event for both edges.

* **Note:** HW will switch I/O Configuration directly when CAP0EN =1.

| Bit | Name | Description | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:4 | Reserved | | R | 0 |
| 3 | CAP0EN | Capture 0 function enable bit 0: Disable 1: Enable.. | R/W | 0 |
| 2 | CAP0IE | Interrupt on CT32Bn_CAP0 event: a CAP0 load due to a CT32Bn_CAP0 event will generate an interrupt. 0: Disable 1: Enable | R/W | 0 |
| 1 | CAP0FE | Capture on CT32Bn_CAP0 falling edge: a sequence of 1 then 0 on CT32Bn_CAP0 will cause CAP0 to be loaded with the contents of TC. 0: Disable 1: Enable | R/W | 0 |
| 0 | CAP0RE | Capture on CT32Bn_CAP0 rising edge: a sequence of 0 then 1 on CT32Bn_CAP0 will cause CAP0 to be loaded with the contents of TC. 0: Disable 1: Enable | R/W | 0 |

8.7.7 CT32Bn Capture 0 register (CT32Bn_CAP0) (n=0,1)

Address Offset: 0x2C

Each Capture register is associated with a device pin and may be loaded with the counter/timer value when a specified event occurs on that pin. The settings in the Capture Control register determine whether the capture function is enabled, and whether a capture event happens on the rising edge of the associated pin, the falling edge, or on both edges.

| Bit | Name | Description | Attribute | Reset |
|------|------------|-----------------------------|-----------|-------|
| 31:0 | CAP0[31:0] | Timer counter capture value | R | 0 |

8.7.8 CT32Bn External Match register (CT32Bn_EM) (n=0,1)

Address Offset: 0x30

The External Match register provides both control and status of the external match pins CT32Bn_PWMCTRL[3:0]. If the match outputs are configured as PWM output, the function of the external match registers is determined by the

PWM rules.

| Bit | Name | Description | Attribute | Reset |
|-------|-----------|---|-----------|-------|
| 31:12 | Reserved | | R | 0 |
| 11:10 | EMC3[1:0] | Determines the functionality of CT32Bn_PWM3. 00: Do Nothing. 01: CT32Bn_PWM3 pin is LOW 10: CT32Bn_PWM3 pin is HIGH 11: Toggle CT32Bn_PWM3 pin. | R/W | 0 |
| 9:8 | EMC2[1:0] | Determines the functionality of CT32Bn_PWM2. 00: Do Nothing. 01: CT32Bn_PWM2 pin is LOW 10: CT32Bn_PWM2 pin is HIGH 11: Toggle CT32Bn_PWM2 pin. | R/W | 0 |
| 7:6 | EMC1[1:0] | Determines the functionality of CT32Bn_PWM1. 00: Do Nothing. 01: CT32Bn_PWM1 pin is LOW 10: CT32Bn_PWM1 pin is HIGH. 11: Toggle CT32Bn_PWM1. | R/W | 0 |
| 5:4 | EMC0[1:0] | Determines the functionality of CT32Bn_PWM0. 00: Do Nothing. 01: CT32Bn_PWM0 pin is LOW 10: CT32Bn_PWM0 pin is HIGH 11: Toggle CT32Bn_PWM0. | R/W | 0 |
| 3 | EM3 | When the TC and MR3 are equal, this bit will act according to EMC3 bits, and also drive the state of CT32Bn_PWM3 output. | R/W | 0 |
| 2 | EM2 | When the TC and MR2 are equal, this bit will act according to EMC2 bits, and also drive the state of CT32Bn_PWM2 output. | R/W | 0 |
| 1 | EM1 | When the TC and MR1 are equal, this bit will act according to EMC1 bits, and also drive the state of CT32Bn_PWM1 output. | R/W | 0 |
| 0 | EM0 | When the TC and MR0 are equal, this bit will act according to EMC0 bits, and also drive the state of CT32Bn_PWM0 output. | R/W | 0 |

8.7.9 CT32Bn PWM Control register (CT32Bn_PWMCTRL) (n=0,1)

Address Offset: 0x34

The PWM Control register is used to configure the match outputs as PWM outputs. Each match output can be independently set to perform either as PWM output or as match output whose function is controlled by [CT32Bn_EM](#) register.

For each timer, a maximum of three single edge controlled PWM outputs can be selected on the CT32Bn_PWMCTRL[3:0] outputs. One additional match register determines the PWM cycle length. When a match occurs in any of the other match registers, the PWM output is set to HIGH. The timer is reset by the match register that is configured to set the PWM cycle length. When the timer is reset to zero, all currently HIGH match outputs configured as PWM outputs are cleared.

| Bit | Name | Description | Attribute | Reset |
|-------|----------|--|-----------|-------|
| 31:24 | Reserved | | R | 0 |
| 23 | PWM3IOEN | CT32Bn_PWM3/GPIO selection bit 0: CT32Bn_PWM3 pin act as GPIO 1: CT32Bn_PWM3 pin act as match output, and output signal depends on PWM3EN bit. | R/W | 0 |
| 22 | PWM2IOEN | CT32Bn_PWM2/GPIO selection bit 0: CT32Bn_PWM2 pin act as GPIO 1: CT32Bn_PWM2 pin act as match output, and output signal depends on PWM2EN bit. | R/W | 0 |
| 21 | PWM1IOEN | CT32Bn_PWM1/GPIO selection bit 0: CT32Bn_PWM1 pin act as GPIO 1: CT32Bn_PWM1 pin act as match output, and output signal depends on PWM1EN bit. | R/W | 0 |
| 20 | PWM0IOEN | CT32Bn_PWM0/GPIO selection bit 0: CT32Bn_PWM0 pin act as GPIO 1: CT32Bn_PWM0 pin act as match output, and output signal depends on | R/W | 0 |

| | | PWM0EN bit. | | |
|-------------|----------|--|-----|---|
| 19:4 | Reserved | | R | 0 |
| 3 | PWM3EN | PWM3 enable 0: CT32Bn_PWM3 is controlled by EM3. 1: PWM mode is enabled for CT32Bn_PWM3. | R/W | 0 |
| 2 | PWM2EN | PWM2 enable 0: CT32Bn_PWM2 is controlled by EM2. 1: PWM mode is enabled for CT32B0_MAT2. | R/W | 0 |
| 1 | PWM1EN | PWM1 enable 0: CT32Bn_PWM1 is controlled by EM1. 1: PWM mode is enabled for CT32Bn_PWM1. | R/W | 0 |
| 0 | PWM0EN | PWM0 enable 0: CT32Bn_PWM0 is controlled by EM0. 1: PWM mode is enabled for CT32Bn_PWM0. | R/W | 0 |

8.7.10 CT32Bn Timer Raw Interrupt Status register (CT32Bn_RIS) (n=0,1)

Address Offset: 0x38

This register indicates the raw status for Timer/PWM interrupts. A Timer/PWM interrupt is sent to the interrupt controller if the corresponding bit in the CT16Bn_IE register is set.

| Bit | Name | Description | Attribute | Reset |
|-------------|----------|--|-----------|-------|
| 31:5 | Reserved | | R | 0 |
| 4 | CAP0IF | Interrupt flag for capture channel 0. 0: No interrupt on CAP0 1: Interrupt requirements met on CAP0. | R | 0 |
| 3 | MR3IF | Interrupt flag for match channel 3. 0: No interrupt on match channel 3 1: Interrupt requirements met on match channel 3. | R | 0 |
| 2 | MR2IF | Interrupt flag for match channel 2. 0: No interrupt on match channel 2 1: Interrupt requirements met on match channel 2. | R | 0 |
| 1 | MR1IF | Interrupt flag for match channel 1. 0: No interrupt on match channel 1 1: Interrupt requirements met on match channel 1. | R | 0 |
| 0 | MR0IF | Interrupt flag for match channel 0. 0: No interrupt on match channel 0 1: Interrupt requirements met on match channel 0. | R | 0 |

8.7.11 CT32Bn Timer Interrupt Clear register (CT32Bn_IC) (n=0,1)

Address Offset: 0x3C

| Bit | Name | Description | Attribute | Reset |
|-------------|----------|-------------------------------------|-----------|-------|
| 31:5 | Reserved | | R | 0 |
| 4 | CAP0IC | 0: No effect 1: Clear CAP0IF bit | W | 0 |
| 3 | MR3IC | 0: No effect 1: Clear MR3IF bit | W | 0 |
| 2 | MR2IC | 0: No effect 1: Clear MR2IF bit | W | 0 |
| 1 | MR1IC | 0: No effect 1: Clear MR1IF bit | W | 0 |
| 0 | MR0IC | 0: No effect 1: Clear MR0IF bit | W | 0 |

9 WATCHDOG TIMER (WDT)

9.1 OVERVIEW

The purpose of the Watchdog is to reset the MCU within a reasonable amount of time if it enters an erroneous state. When enabled, the Watchdog will generate a system reset or interrupt if the user program fails to "feed" (or reload) the Watchdog within a predetermined amount of time.

The Watchdog consists of a divide by 128 fixed pre-scaler and an 8-bit counter. The clock is fed to the timer via a pre-scaler. The timer decrements when clocked. The minimum value from which the counter decrements is 0x01. Hence the minimum Watchdog interval is $(T_{WDT_PCLK} \times 128 \times 1)$ and the maximum Watchdog interval is $(T_{WDT_PCLK} \times 128 \times 256)$.

The Watchdog should be used in the following manner:

1. Select the clock source for the watchdog timer with WDTCLKSEL register.
2. Set the prescale value for the watchdog clock with WDTPRE bits in [_APB Clock Prescale register 1 \(SYS1_APBPCP1\)](#) register.
3. Set the Watchdog timer constant reload value in [WDT_TC](#) register.
4. Enable the Watchdog and setup the Watchdog timer operating mode in [WDT_CFG](#) register.
5. The Watchdog should be fed again by writing 0x55AA to [WDT_FEED](#) register before the Watchdog counter underflows to prevent reset or interrupt.

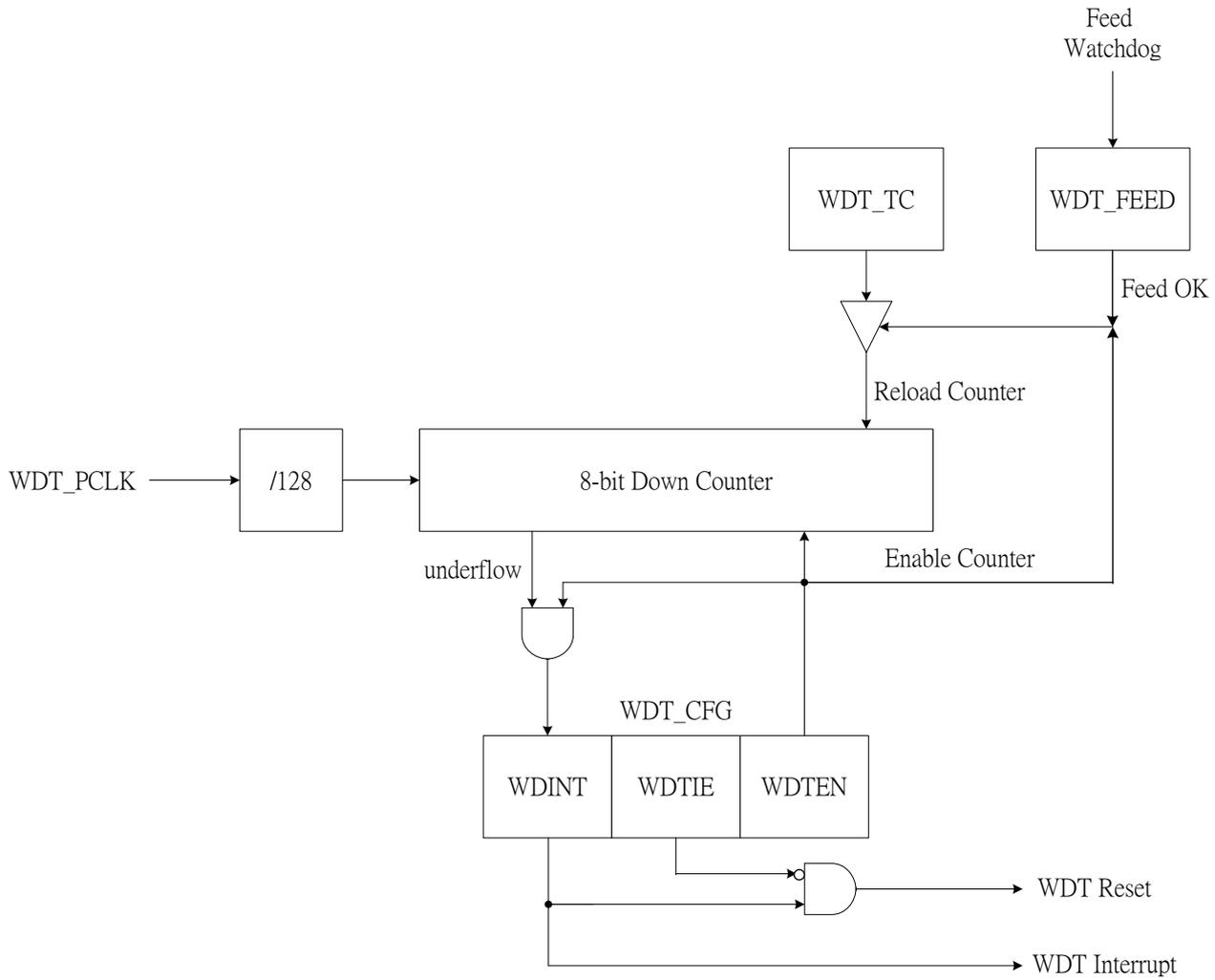
When the watchdog is started by setting the WDTEN in [WDT_CFG](#) register, the time constant value is loaded in the watchdog counter and the counter starts counting down. When the Watchdog is in the reset mode and the counter underflows, the CPU will be reset, loading the stack pointer and program counter from the vector table as in the case of external reset. Whenever the value 0x55AA is written in [WDT_FEED](#) register, the WDT_TC value is reloaded in the watchdog counter and the watchdog reset or interrupt is prevented.

The watchdog timer block uses two clocks: HCLK and WDT_PCLK. HCLK is used for the AHB accesses to the watchdog registers and is derived from the system clock. The WDT_PCLK is used for the watchdog timer counting. Several clocks can be used as a clock source for WDT_PCLK clock: IHRC, ILRC, ELS X'tal, and HCLK.

The clock to the watchdog register block can be disabled in [AHB Clock Enable register \(SYS1_AHBCLKEN\)](#) register for power savings.

Watchdog reset or interrupt will occur any time the watchdog is running and has an operating clock source.

9.2 BLOCK DIAGRAM



9.3 WDT REGISTERS

Base Address: 0x4001 0000

9.3.1 Watchdog Configuration register (WDT_CFG)

Address Offset: 0x00

The WDT_CFG register controls the operation of the Watchdog through the combination of WDTEN and WDTIE bits. This register indicates the raw status for Watchdog Timer interrupts. A WDT interrupt is sent to the interrupt controller if both the WDINT bit and the WDTIE bit are set.

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:3 | Reserved | | R | 0 |
| 2 | WDTINT | Watchdog interrupt flag <Read> 0: Watchdog does not cause an interrupt. 1: Watchdog timeout and causes an interrupt (Only when WDTIE =1). <Write> 0: Clear this flag. SW shall feed Watchdog before clearing. | R/W | 0 |
| 1 | WDTIE | Watchdog interrupt enable 0: Watchdog timeout will cause a chip reset. (Watchdog reset mode) Watchdog counter underflow will reset the MCU, and will clear the WDINT flag. 1: Watchdog timeout will cause an interrupt. (Watchdog interrupt mode) | R/W | 0 |
| 0 | WDTEN | Watchdog enable 0: Disable 1: Enable. When enable the watchdog, the WDT_TC value is loaded in the watchdog counter. | R/W | 0 |

9.3.2 Watchdog Clock Source register (WDT_CLKSOURCE)

Address Offset: 0x04

| Bit | Name | Description | Attribute | Reset |
|------|-------------|--|-----------|-------|
| 31:2 | Reserved | | R | 0 |
| 1:0 | CLKSEL[1:0] | Selected Watchdog clock source. 00: IHRC oscillator 01: HCLK 10: ILRC oscillator 11: ELS X'TAL | R/W | 0 |

9.3.3 Watchdog Timer Constant register (WDT_TC)

Address Offset: 0x08

The WDT_TC register determines the time-out value. Every time a feed sequence occurs the WDT_TC content is reloaded in to the Watchdog timer. It's an 8-bit counter. Thus the time-out interval is $T_{WDT_PCLK} \times 128 \times 1 \sim T_{WDT_PCLK} \times 128 \times 256$.

$$\begin{aligned} \text{Watchdog overflow time} &= (0.02\mu\text{s} \times 1) \times 128 \times 1 \sim (0.0625\text{ms} \times 32) \times 128 \times 256 \\ &= 2.56\mu\text{s} \sim 65536\text{ms} \end{aligned}$$

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7:0 | TC[7:0] | Watchdog timer constant reload value = TC[7:0]+1 0000 0000 : Timer constant = 1 0000 0001 : Timer constant = 2 | R/W | 0xFF |

| |
|----------------------------------|
| 1111 1110 : Timer constant = 255 |
| 1111 1111 : Timer constant = 256 |

9.3.4 Watchdog Feed register (WDT_FEED)

Address Offset: 0x0C

| Bit | Name | Description | Attribute | Reset |
|-------|----------|--|-----------|-------|
| 31:16 | Reserved | | R | 0 |
| 15:0 | FV[15:0] | Feed value (Read as 0x0) 0x55AA: The watchdog is fed, and the WDT_TC value is reloaded in the watchdog counter. | W | 0 |

10 REAL-TIME CLOCK (RTC)

10.1 OVERVIEW

The RTC is an independent timer. The RTC provides a set of continuously running counters which can be used to provide a clock-calendar function with suitable software.

The counter values can be written to set the current time/date of the system.

10.2 FEATURES

- Programmable prescale value: division factor up to 2^{20}
- 32-bit programmable counter for long-term measurement
- The RTC clock source could be any of the following:
 - EHS XTAL clock divided by 128
 - ELS X'TA
 - ILRC
- Reset sources of the RTC Core (Prescale value, Alarm, Counter and Divider):
 - “Cold” boot
 - DPDWAKEUP
- Three dedicated enabled interrupt lines:
 - Alarm interrupt: generating a software programmable alarm interrupt.
 - Seconds interrupt: generating a periodic interrupt signal with a programmable period length (up to 1 second).
 - Overflow interrupt: to detect when the internal programmable counter rolls over to zero.

10.3 FUNCTIONAL DESCRIPTION

10.3.1 INTRODUCTION

RTC core includes a 20-bit preload value (RTC_SECCNTV). Every TR_CLK period, the RTC generates an interrupt (Second Interrupt) if it is enabled in [RTC_IE](#) register. The second block is a 32-bit programmable counter that can be initialized to the current system time. The system time is incremented at the TR_CLK rate and compared with a programmable date (stored in the RTC_ALR register) in order to generate an alarm interrupt, if enabled in [RTC_IE](#) register.

10.3.2 RESET RTC REGISTERS

The RTC_SECCNTV, RTC_ALMCNTV, RTC_SECCNT, and RTC_ALMCNT registers are reset by “cold” boot or DPDWAKEUP reset.

10.3.3 RTC FLAG ASSERTION

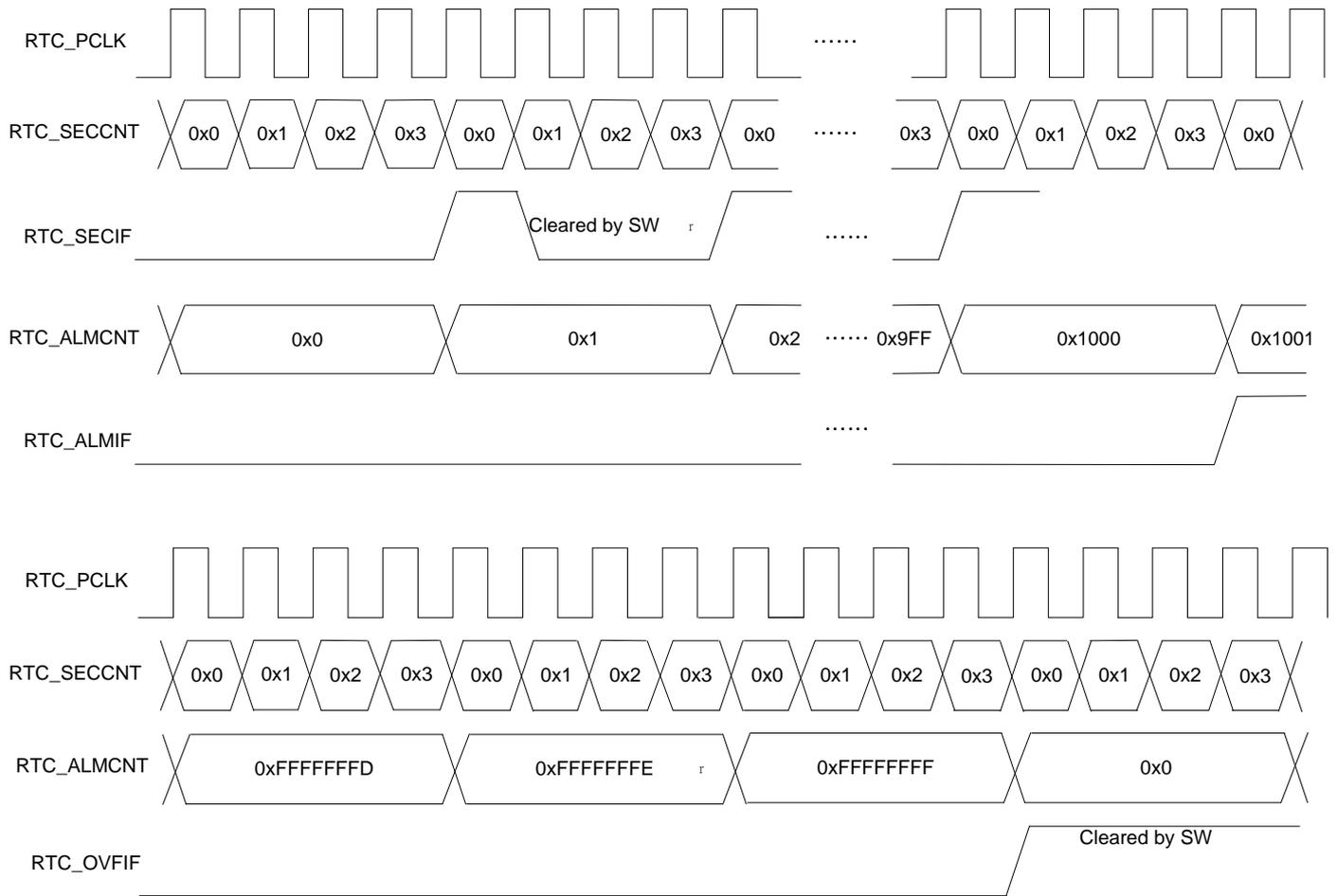
The RTC Second interrupt flag (SECIF) is asserted on each RTC Core clock cycle before the update of the RTC Counter.

The RTC Overflow interrupt flag (OVFIF) is asserted on the last RTC Core clock cycle before the counter reaches 0x0.

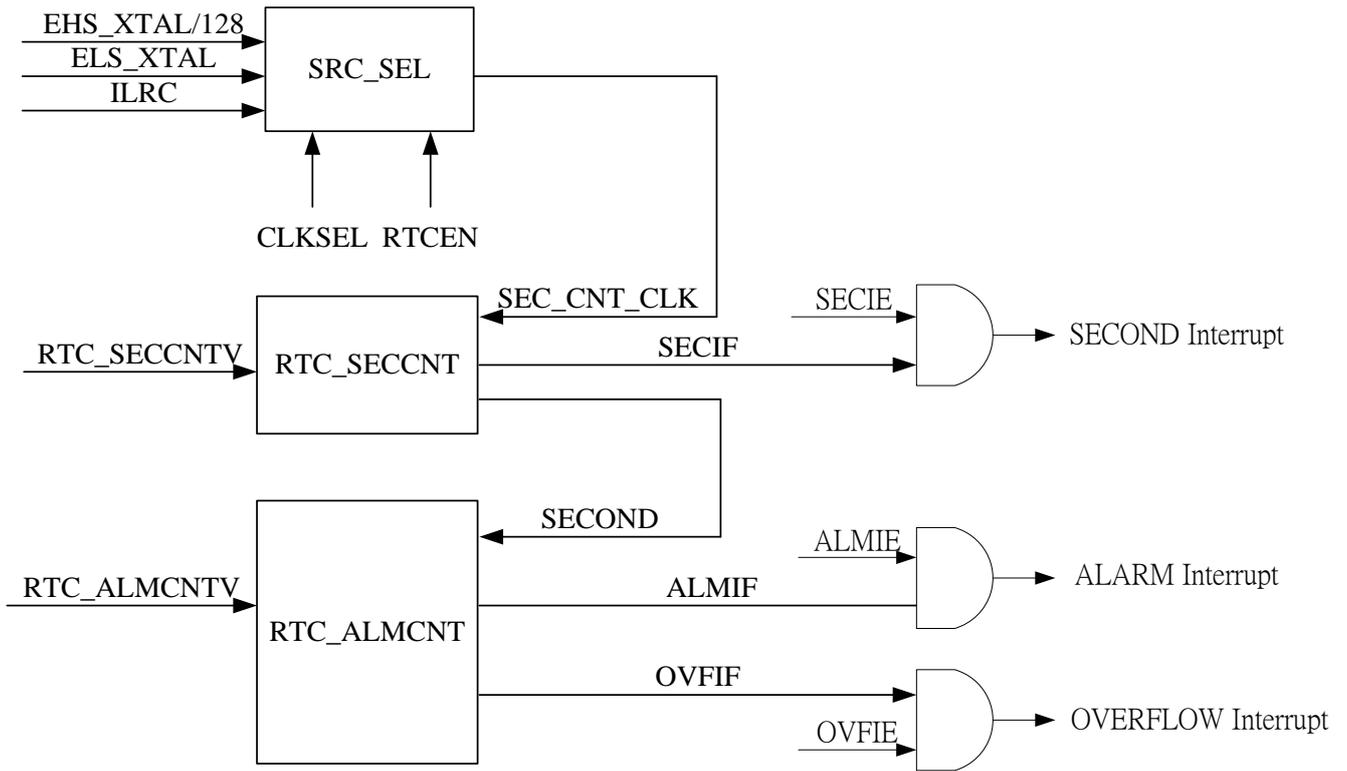
The RTC Alarm interrupt flag (ALMIF) are asserted on the last RTC Core clock cycle before the counter reaches the RTC Alarm counter reload value stored in the Alarm register.

10.3.4 RTC OPERATION

The following figure shows the RTC waveform when it is configured with RTC_SECCNTV=3, RTC_ALMCNTV=0x1000.



10.4 BLOCK DIAGRAM



10.5 RTC REGISTERS

Base Address: 0x4001 2000

10.5.1 RTC Control register (RTC_CTRL)

Address offset: 0x00

* **Note: RTCEN bit shall be set at last!**

| Bit | Name | Description | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:1 | Reserved | | R | 0 |
| 0 | RTCEN | RTC enable bit 0: Disable 1: Enable. Reset SEC_CNT and ALM_CNT. | R/W | 0 |

10.5.2 RTC Clock Source Select register (RTC_CLKS)

Address offset: 0x04

* **Note: SW shall disable RTC (RTCEN=0) when changing the value of this register.**

| Bit | Name | Description | Attribute | Reset |
|------|-------------|---|-----------|-------|
| 31:2 | Reserved | | R | 0 |
| 1:0 | CLKSEL[1:0] | RTC clock source selection. HW will reset SEC_CNT and ALM_CNT when changing the value. 00: ILRC 01: ELS X'TAL 10: Reserved 11: EHS X'TAL clock / 128 | R/W | 0 |

10.5.3 RTC Interrupt Enable register (RTC_IE)

Address offset: 0x08

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:3 | Reserved | | R | 0 |
| 2 | OVFIE | Overflow interrupt enable 0: Disable 1: Enable | R/W | 0 |
| 1 | ALMIE | Alarm interrupt enable 0: Disable 1: Enable | R/W | 0 |
| 0 | SECIE | Second interrupt enable 0: Disable 1: Enable | R/W | 0 |

10.5.4 RTC Raw Interrupt Status register (RTC_RIS)

Address offset: 0x0C

| Bit | Name | Description | Attribute | Reset |
|------|----------|-------------|-----------|-------|
| 31:3 | Reserved | | R | 0 |

| | | | | |
|----------|--------------|---|---|---|
| 2 | OVFIF | Overflow interrupt flag This bit is set by HW when ALM_CNT overflows (ALM_CNT counts from 0xFFFFFFFF to 0x0). An interrupt is generated if OVFIE=1. 0: Overflow not detected 1: 32-bit programmable counter overflow occurred. | R | 0 |
| 1 | ALMIF | Alarm interrupt flag This bit is set by HW when ALM_CNT=ALM_CNTV. An interrupt is generated if ALRIE=1. 0: Alarm not detected 1: Alarm detected. | R | 0 |
| 0 | SECIF | Second interrupt flag This bit is set by HW when SEC_CNT=SEC_CNTV. An interrupt is generated if SECIE=1. 0: Second flag condition not met. 1: Second flag condition met. | R | 0 |

10.5.5 RTC Interrupt Clear register (RTC_IC)

Address offset: 0x10

| Bit | Name | Description | Attribute | Reset |
|-------------|--------------|------------------------------------|-----------|-------|
| 31:3 | Reserved | | R | 0 |
| 2 | OVFIC | 0: No effect 1: Clear OVFIF bit | W | 0 |
| 1 | ALMIC | 0: No effect 1: Clear ALMIF bit | W | 0 |
| 0 | SECIC | 0: No effect 1: Clear SECIF bit | W | 0 |

10.5.6 RTC Second Counter Reload Value register (RTC_SECNTV)

Address offset: 0x14

Reset value: 0x8000

| Bit | Name | Description | Attribute | Reset |
|--------------|---------------------|---|-----------|--------|
| 31:20 | Reserved | | R | 0 |
| 19:0 | SECNTV[19:0] | RTC second counter reload value. Update this register will reset RTC_SECNT and RTC_ALMCNT registers. The zero value is not recommended, and will be replaced with default value (0x8000) by HW. | R/W | 0x8000 |

10.5.7 RTC Second Count register (RTC_SECNT)

Address offset: 0x18

The RTC core has one 32-bit programmable counter, and this register keeps the current counting value of this counter.

| Bit | Name | Description | Attribute | Reset |
|-------------|--------------------|---|-----------|-------|
| 31:0 | SECNT[31:0] | RTC second counter The current value of the RTC counter. | R | 0 |

10.5.8 RTC Alarm Counter Reload Value register (RTC_ALMCNTV)

Address offset: 0x1C

Reset value: 0xFFFFFFFF

| Bit | Name | Description | Attribute | Reset |
|-------------|----------------------|---------------------------------|-----------|------------|
| 31:0 | ALMCNTV[31:0] | RTC alarm counter reload value. | R/W | 0xFFFFFFFF |

| | | | | |
|--|--|---|--|--|
| | | <p>Update this register will reset ALMCNT. The zero value is not recommended, and will be replaced with default value (0xFFFFFFFF) by HW.</p> | | |
|--|--|---|--|--|

10.5.9 RTC Alarm Count register (RTC_ALMCNT)

Address offset: 0x20

| Bit | Name | Description | Attribute | Reset |
|------|--------------|--|-----------|-------|
| 31:0 | ALMCNT[31:0] | RTC alarm counter The current value of the RTC alarm counter. | R | 0x0 |

11 SPI/SSP

11.1 OVERVIEW

The SSP is a Synchronous Serial Port controller capable of operation on a SPI, and 4-wire SSI bus. It can interact with multiple masters and slaves on the bus. Only a single master and a single slave can communicate on the bus during a given data transfer. Data transfers are in principle full duplex, with frames of 4 to 16 bits of data flowing from the master to the slave and from the slave to the master. In practice it is often the case that only one of these data flows carries meaningful data.

11.2 FEATURES

- Compatible with Motorola SPI, and 4-wire TI SSI bus.
- Synchronous Serial Communication.
- Supports master or slave operation.
- 8-frame FIFO for both transmitter and receiver.
- 4-bit to 16-bit frame.
- Maximum SPI speed of 25 Mbps (master) or 6 Mbps (slave) in SSP mode.
- Data transfer format is from MSB or LSB controlled by register.
- The start phase of data sampling location selection is 1st-phase or 2nd-phase controlled register.

11.3 PIN DESCRIPTION

| Pin Name | Type | Description | GPIO Configuration |
|-------------------|------|--|-----------------------------------|
| SCKn | O | SSP Serial clock (Master) | |
| | I | SSP Serial clock (Slave) | Depends on GPIO _n _CFG |
| SELn | O | SPI Slave Select/SSI Frame Sync (Master) | |
| | I | SSP Slave Select (Slave) | Depends on GPIO _n _CFG |
| MISO _n | I | Master In Slave Out (Master) | Depends on GPIO _n _CFG |
| | O | Master In Slave Out (Slave) | |
| MOSIn | O | Master Out Slave In (Master) | |
| | I | Master Out Slave In (Slave) | Depends on GPIO _n _CFG |

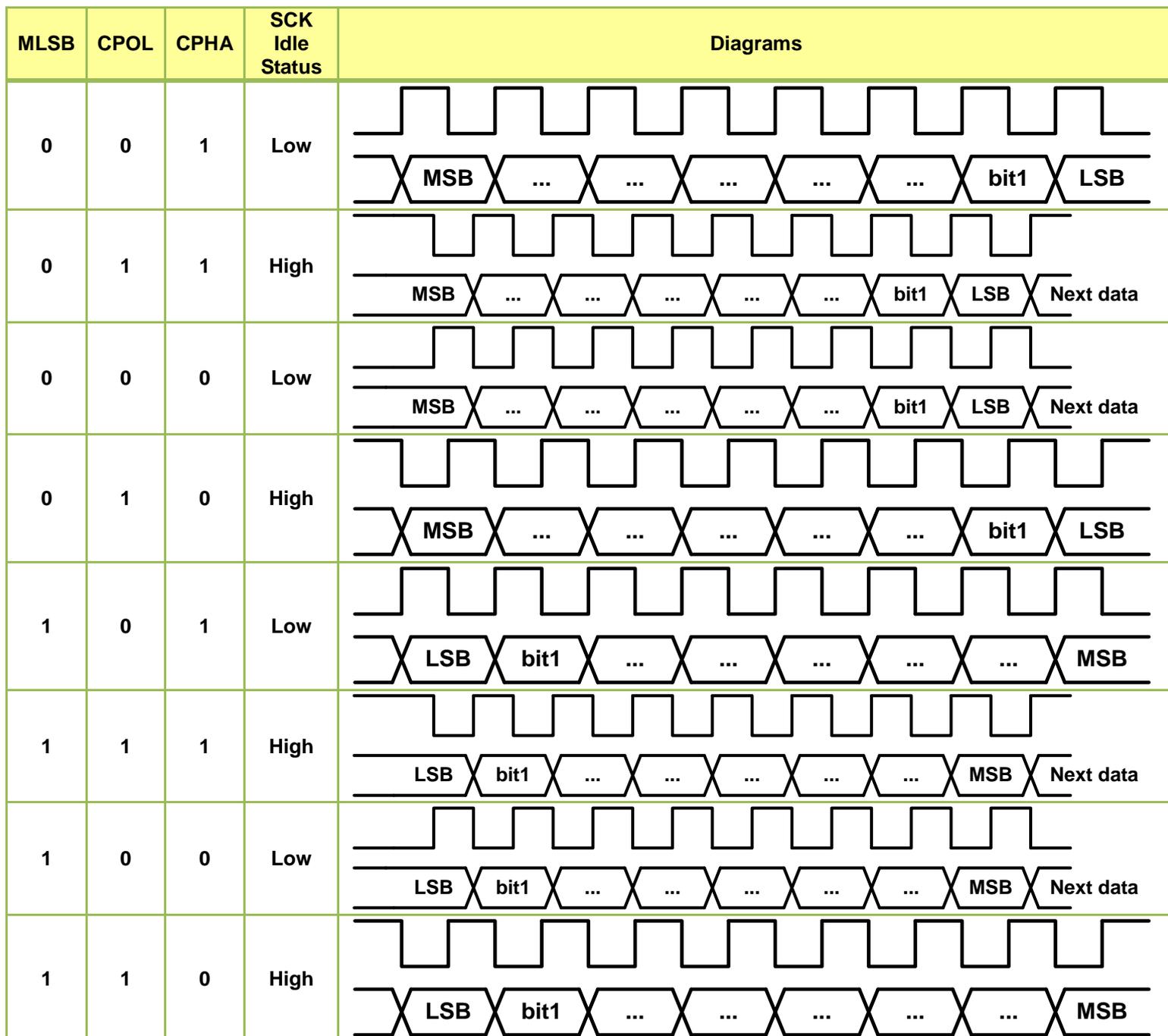
11.4 INTERFACE DESCRIPTION

11.4.1 SPI

The SPI interface is a 4-wire interface where the SEL signal behaves as a slave select. The main feature of the SPI format is that the inactive state and phase of the SCK signal are programmable through the CPOL and CPHA bits in [SSPn_CTRL1](#) register.

When the “CPOL” clock polarity control bit is LOW, it produces a steady state low value on the SCK pin. If the CPOL clock polarity control bit is HIGH, a steady state high value is placed on the CLK pin when data is not being transferred. The “CPHA” clock phase bit controls the phase of the clock on which data is sampled. When CPHA=1, the SCK first edge is for data transition, and receive and transmit data is at SCK 2nd edge. When CPHA=0, the 1st bit is fixed already, and the SCK first edge is to receive and transmit data.

The SPI data transfer timing as following figure:



11.4.2 SSI

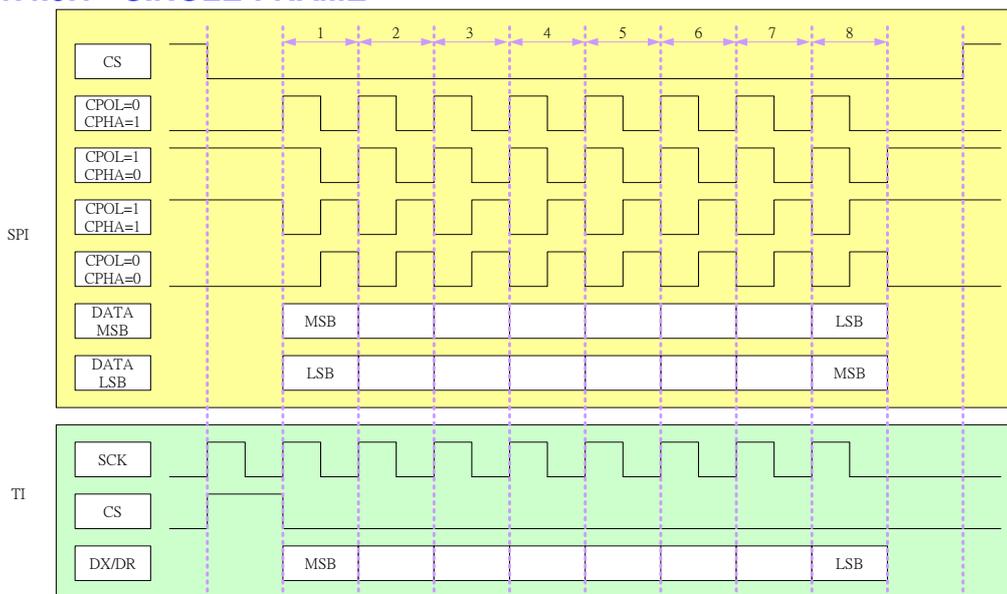
For device configured as a master in this mode, SCK and CS are forced LOW, and the transmit data line DX is in 3-state mode whenever the SSP hardware is idle.

Once the bottom entry of the transmit FIFO contains data, CS is pulsed HIGH for one SCK period. The value to be transmitted is also transferred from the transmit FIFO to the serial shift register of the shifted out on the DX pin. Likewise, the MSB of the received data is shifted onto the DR pin by the off-chip serial slave device.

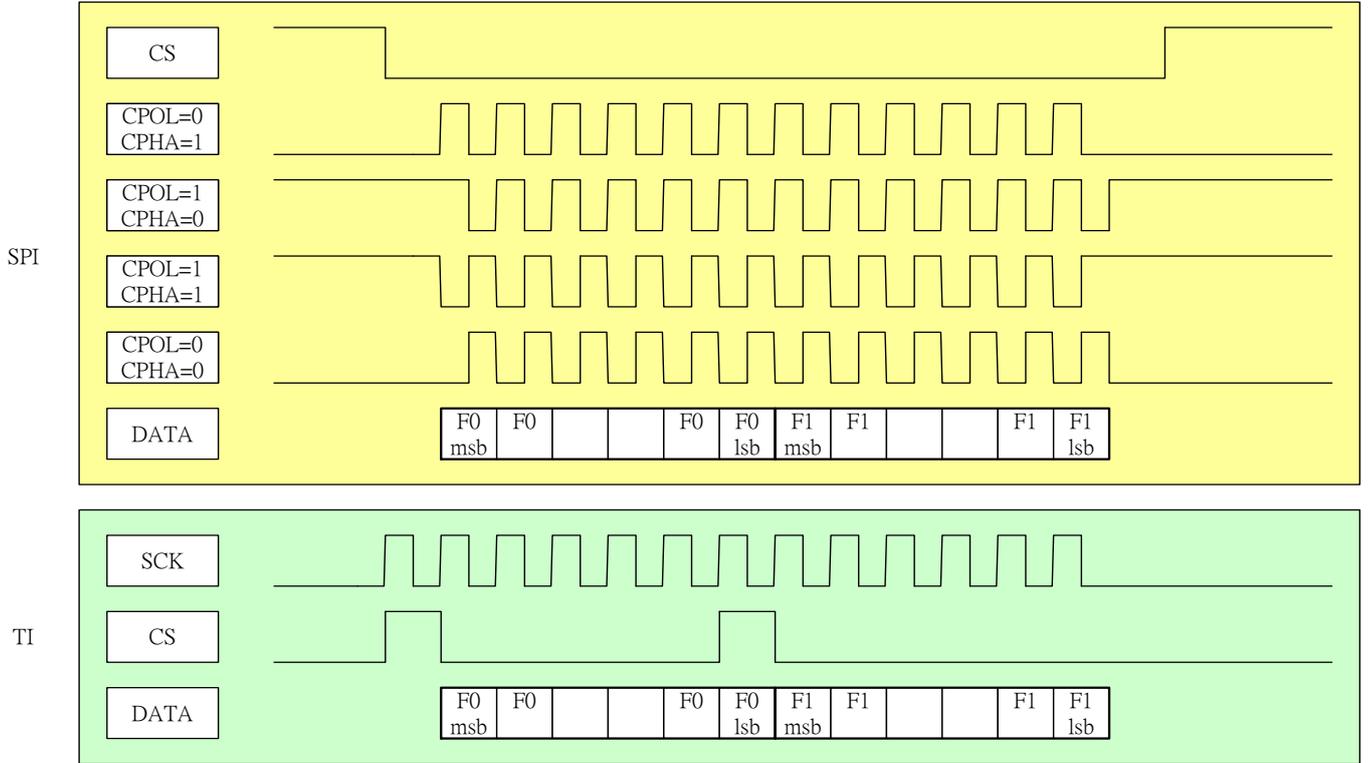
Both the SSP hardware and the off-chip serial slave device then clock each data bit into their serial shifter on the falling edge of each SCK. The received data is transferred from the serial shifter to the receive FIFO on the first rising edge of SCK after the LSB has been latched.

11.4.3 COMMUNICATION FLOW

11.4.3.1 SINGLE-FRAME



11.4.3.2 MULTI-FRAME



11.5 SSP REGISTERS

Base Address: 0x4001 C000 (SSP0)
0x4005 8000 (SSP1)

11.5.1 SSP n Control register 0 (SSPn_CTRL0) (n=0, 1)

Address Offset:0x00

- * Note:**
- 1. **Must reset SSP FSM with FRESET[1:0] after changing any configuration of SSP when SSPEN = 1.**
 - 2. **HW will switch I/O configurations refer to FORMAT bit directly when SSPEN = 1.**

| Bit | Name | Description | Attribute | Reset |
|-------|-------------|--|-----------|-------|
| 31:12 | Reserved | | R | 0 |
| 11:8 | DL[3:0] | Data length = DL[3:0] + 1 0000~0001: Reversed 0010: data length = 3 ... 1110: data length = 15 1111: data length = 16 | R/W | 1111b |
| 7:6 | FRESET[1:0] | SSP FSM and FIFO Reset bit 00: No effect 01: Reserved 10: Reserved 11: Reset finite state machine and FIFO. (BUF_BUSY = 0, data in shift BUF is cleared, TX_EMPTY = 1, TX_FULL = 0, RX_EMPTY = 1, RX_FULL = 0, and data in FIFO is cleared). This bit will be cleared by HW automatically. | W | 0 |
| 5 | Reserved | | R | 0 |
| 4 | FORMAT | Interface format. 0: SPI 1: SSI | R/W | 0 |
| 3 | MS | Master/Slave selection bit 0: Act as Master. 1: Act as Slave. | R/W | 0 |
| 2 | SDODIS | Slave data output disable bit (ONLY used in slave mode) 0: Enable slave data output. 1: Disable slave data output. (MISO=0) | R/W | 0 |
| 1 | LOOPBACK | Loop back mode enable 0: Disable 1: Data input from data output | R/W | 0 |
| 0 | SSPEN | SSP enable bit 0: Disable 1: Enable. | R/W | 0 |

11.5.2 SSP n Control register 1 (SSPn_CTRL1) (n=0, 1)

Address Offset: 0x04

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:3 | Reserved | | R | 0 |
| 2 | CPHA | Clock phase for edge sampling. 0: Data changes at clock falling edge, latches at clock rising edge when CPOL = 0; Data changes at clock rising edge, latches at clock falling edge when CPOL = 1. 1: Data changes at clock rising edge, latches at clock falling edge when | R/W | 0 |

| | | | | |
|---|------|---|-----|---|
| | | CPOL = 0; Data changes at clock falling edge, latches at clock rising edge when CPOL = 1. | | |
| 1 | CPOL | Clock polarity selection bit 0: SCK idles at Low level. 1: SCK idles at High level. | R/W | 0 |
| 0 | MLSB | MSB/LSB selection bit 0: MSB transmit first. 1: LSB transmit first. | R/W | 0 |

11.5.3 SSP n Clock Divider register (SSPn_CLKDIV) (n=0, 1)

Address Offset: 0x08

| Bit | Name | Description | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7:0 | DIV[7:0] | SSPn clock divider 0: SCK = SSPn_PCLK / 2 1: SCK = SSPn_PCLK / 4 2: SCK = SSPn_PCLK / 6 X: SCK = SSPn_PCLK / (2X+2) | R/W | 0 |

11.5.4 SSP n Status register (SSPn_STAT) (n=0, 1)

Address Offset: 0x0C

| Bit | Name | Description | Attribute | Reset |
|------|-------------|--|-----------|-------|
| 31:7 | Reserved | | R | 0 |
| 6 | RX_HF_FULL | RX FIFO half-full flag 0: Data in RX FIFO < 4 1: Data in RX FIFO ≥ 4 | R | 0 |
| 5 | TX_HF_EMPTY | TX FIFO half-empty flag 0: Data in TX FIFO > 4 1: Data in TX FIFO ≤ 4 | R | 1 |
| 4 | BUSY | Busy flag. 0: SSP controller is idle. 1: SSP controller is transferring. | R | 0 |
| 3 | RX_FULL | RX FIFO full flag. 0: RX FIFO is NOT full. 1: RX FIFO is full. | R | 0 |
| 2 | RX_EMPTY | RX FIFO empty flag 0: RX FIFO is NOT empty. 1: RX FIFO is empty. | R | 1 |
| 1 | TX_FULL | TX FIFO full flag. 0: TX FIFO is NOT full. 1: TX FIFO is full. | R | 0 |
| 0 | TX_EMPTY | TX FIFO empty flag 0: TX FIFO is NOT empty. In Master mode, the transmitter will begin to transmit automatically. 1: TX FIFO is empty. | R | 1 |

11.5.5 SSP n Interrupt Enable register (SSPn_IE) (n=0, 1)

Address Offset: 0x10

This register controls whether each of the four possible interrupt conditions in the SSP controller is enabled.

| Bit | Name | Description | Attribute | Reset |
|------|----------|--------------------------------|-----------|-------|
| 31:4 | Reserved | | R | 0 |
| 3 | TXHEIE | TX half-empty interrupt enable | R/W | 0 |

| | | | | |
|----------|---------|--|-----|---|
| | | 0: Disable 1: Enable | | |
| 2 | RXHFIE | RX half-full interrupt enable 0: Disable 1: Enable | R/W | 0 |
| 1 | RXTOIE | RX time-out interrupt enable 0: Disable 1: Enable | R/W | 0 |
| 0 | RXOVFIE | RX Overflow interrupt enable 0: Disable 1: Enable | R/W | 0 |

11.5.6 SSP n Raw Interrupt Status register (SSPn_RIS) (n=0, 1)

Address Offset: 0x14

This register contains the status for each interrupt condition, regardless of whether or not the interrupt is enabled in SSPn_IE register.

This register indicates the status for SSP control raw interrupts. An SSP interrupt is sent to the interrupt controller if the corresponding bit in the SSPn_IE register is set.

| Bit | Name | Description | Attribute | Reset |
|-------------|----------|--|-----------|-------|
| 31:4 | Reserved | | R | 0 |
| 3 | TXHEIF | TX FIFO is at least half empty or not. 0: TX FIFO isn't at least half empty. 1: TX FIFO is at least half empty. | R | 0 |
| 2 | RXHFIF | RX FIFO is at least half full or not. 0: RX FIFO isn't at least half full. 1: RX FIFO is at least half full. | R | 0 |
| 1 | RXTOIF | RX time-out interrupt flag RXTO occurs when the RX FIFO is not empty, and has not been read for a time-out period (32*SSPn_PCLK). The time-out period is the same for master and slave modes. 0: RXTO doesn't occur. 1: RXTO occurs. | R | 0 |
| 0 | RXOVFIF | RX Overflow interrupt flag RXOVF occurs when the RX FIFO is full and another frame is completely received. The ARM spec implies that the preceding frame data is overwritten by the new frame data when this occurs. 0: RXOVF doesn't occur. 1: RXOVF occurs. | R | 0 |

11.5.7 SSP n Interrupt Clear register (SSPn_IC) (n=0, 1)

Address Offset: 0x18

| Bit | Name | Description | Attribute | Reset |
|-------------|----------|---------------------------------------|-----------|-------|
| 31:4 | Reserved | | R | 0 |
| 3 | TXHEIC | 0: No effect 1: Clear TXHEIF bit. | W | 0 |
| 2 | RXHFIC | 0: No effect 1: Clear RXHFIF bit. | W | 0 |
| 1 | RXTOIC | 0: No effect 1: Clear RXTOIF bit. | W | 0 |
| 0 | RXOVFIC | 0: No effect 1: Clear RXOVFIF bit. | W | 0 |

11.5.8 SSP n Data register (SSPn_DATA) (n=0, 1)

Address Offset: 0x1C

| Bit | Name | Description | Attribute | Reset |
|-------|------------|--|-----------|-------|
| 31:16 | Reserved | | R | 0 |
| 15:0 | DATA[15:0] | <p><u>Write</u> SW can write data to be sent in a future frame to this register when TX_FULL = 0 in SSPn_STAT register (TX FIFO is not full). If the TX FIFO was previously empty and the SSP controller is not busy on the bus, transmission of the data will begin immediately. Otherwise the data written to this register will be sent as soon as all previous data has been sent (and received).</p> <p><u>Read</u> SW can read data from this register when RX_EMPTY=0 in SSPn_STAT register (Rx FIFO is not empty). When SW reads this register, the SSP controller returns data from the least recent frame in the RX FIFO. If the data length is less than 16 bit, the data is right-justified in this field with higher order bits filled with 0s.</p> | R/W | 0 |

11.5.9 SSP n Data Fetch register (SSPn_DF) (n=0, 1)

Address Offset: 0x20

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:1 | Reserved | | R | 0 |
| 0 | DF | <p>SSP data fetch control bit 0: Disable 1: Enable when SCKn frequency > 6MHz</p> | R/W | 0 |

12 I2C

12.1 OVERVIEW

The I2C bus is bidirectional for inter-IC control using only two wires: Serial Clock Line (SCL) and Serial Data line (SDA). Each device is recognized by a unique address and can operate as either a receiver-only device (e.g., an LCD driver) or a transmitter with the capability to both receive and send information (such as memory). Transmitters and/or receivers can operate in either master or slave mode, depending on whether the chip has to initiate a data transfer or is only addressed. The I2C is a multi-master bus and can be controlled by more than one bus master connected to it. It is also SMBus 2.0 compatible.

The I2C interface is byte oriented and has four operating modes:

- Master transmitter mode
- Master receiver mode
- Slave transmitter mode
- Slave receiver mode

12.2 FEATURES

The I2C interface complies with the entire I2C specification, supporting the ability to turn power off to the ARM Cortex-M0 without interfering with other devices on the same I2C-bus.

- Standard I2C-compliant bus interfaces may be configured as Master or Slave.
- I2C Master features:
 - 7-bit and 10-bit addressing mode
 - Clock generation
 - Start and Stop generation
- I2C Slave features:
 - Programmable I2C Address detection
 - Recognition of up to four distinct slave addresses
 - Stop bit detection
 - Clock stretching
- Supports different communication speeds:
 - Standard Speed (up to 100 Kbits/s)
 - Fast Speed (up to 400 Kbits/s)
- Arbitration is handled between simultaneously transmitting masters without corruption of serial data on the bus.
- Programmable clock allows adjustment of I2C transfer rates.
- Data transfer is bidirectional between masters and slaves.
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus.
- Serial clock synchronization is used as a handshake mechanism to suspend and resume serial transfer.
- Monitor mode allows observing all I2C-bus traffic, regardless of slave address.
- I2C-bus can be used for test and diagnostic purposes.
- Generation and detection of 7-bit/10-bit addressing and General Call.

12.3 PIN DESCRIPTION

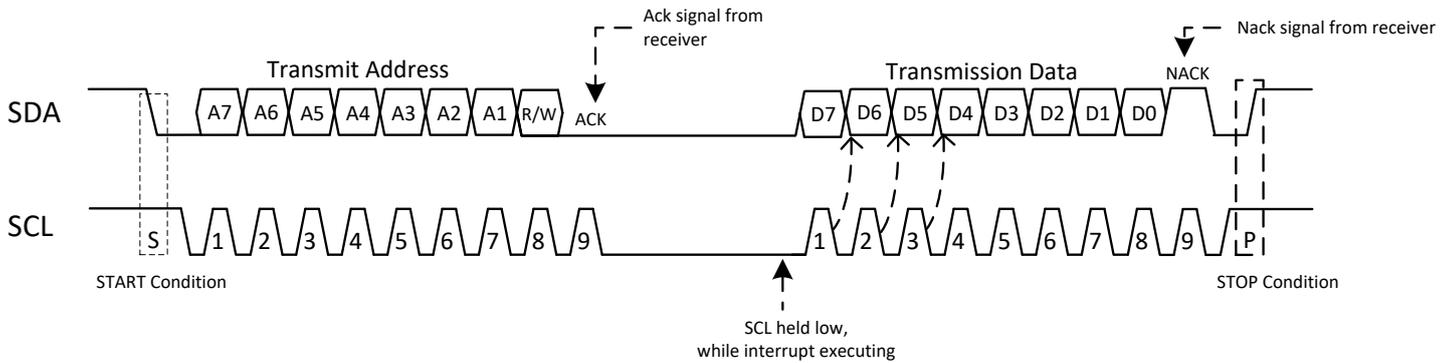
| Pin Name | Type | Description | GPIO Configuration |
|----------|------|------------------|---|
| SCLn | I/O | I2C Serial clock | Output with Open-drain Input depends on GPIO _n _CFG |
| SDAn | I/O | I2C Serial data | Output with Open-drain Input depends on GPIO _n _CFG |

12.4 I2C PROPOCOL

I2C transmission structure includes a START(S) condition, 8-bit address byte, one or more data byte and a STOP (P) condition. START condition is generated by master to initial any transmission.

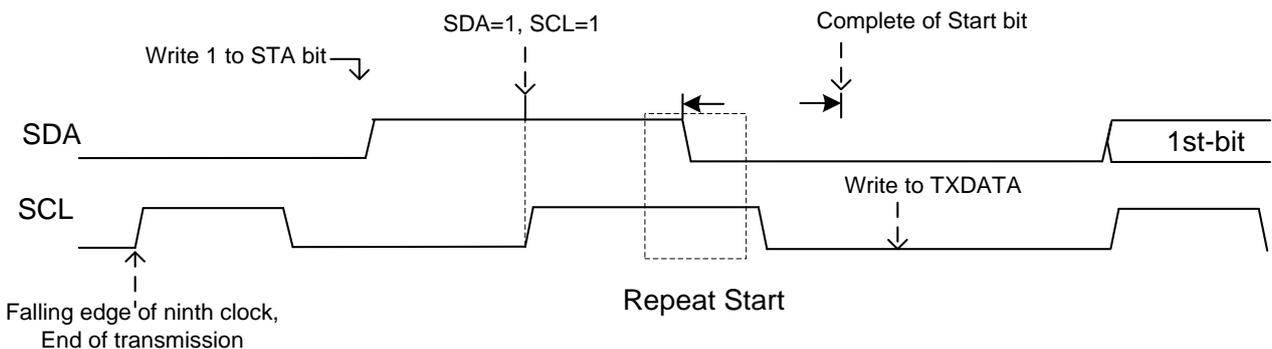
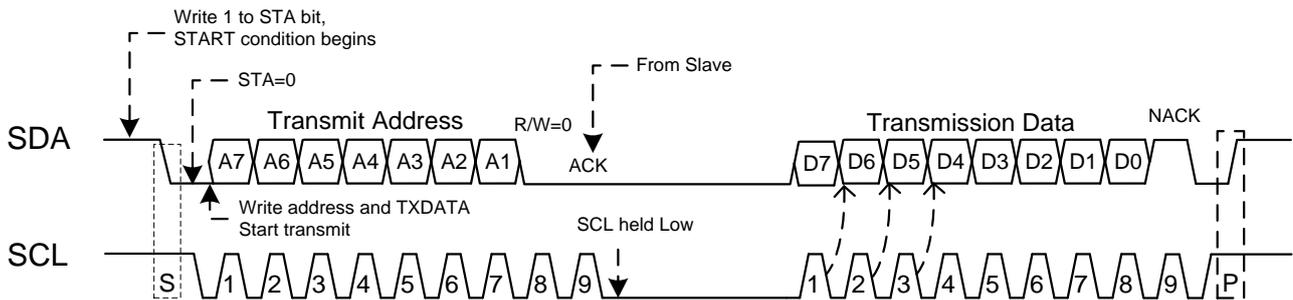
Data is transmitted with the Most Significant Bit (MSB) first. In address byte, the higher 7-bit is address bit and the lowest bit is data direction (R/W) bit. When R/W=0, it assigns a “WRITE” operation. When R/W=1, it assigns a “READ” operation.

After each byte is received, the receiver (a master or a slave) must send an acknowledge (ACK) bit. If transmitter can't receive an ACK, it will recognize a not acknowledge (NACK). In WRITE operation, the master will transmit data to the slave and then waits for ACK from slave. In READ operation, the slave will transmit data to the master and then waits for ACK from master. In the end, the master will generate a STOP condition to finish transmission.

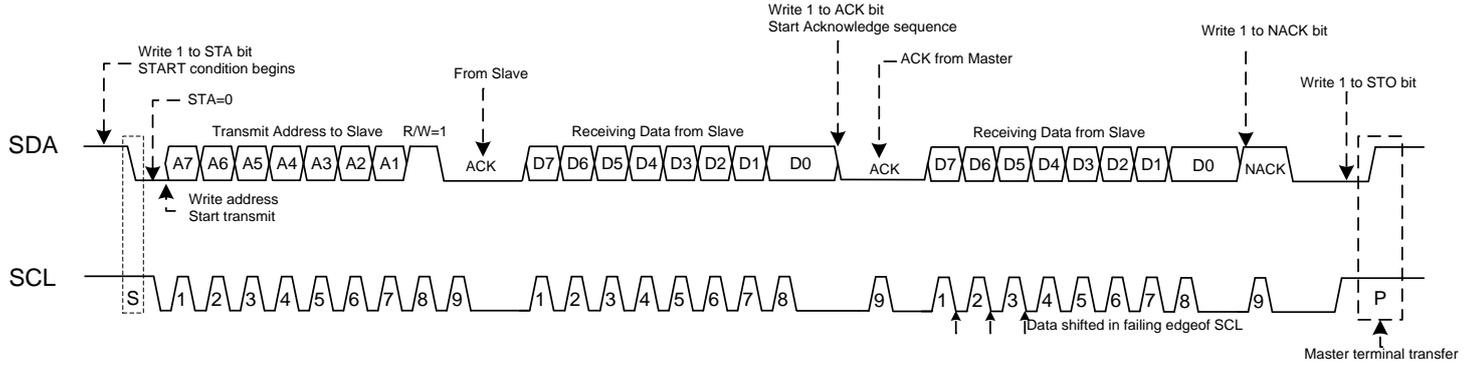


12.4.1 7-BIT ADDRESSING MODES

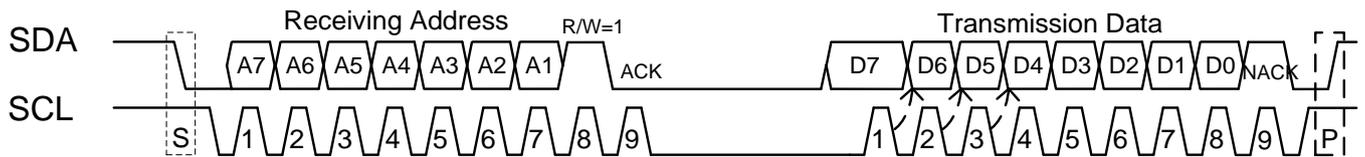
12.4.1.1 MASTER TRANSMITTER MODE



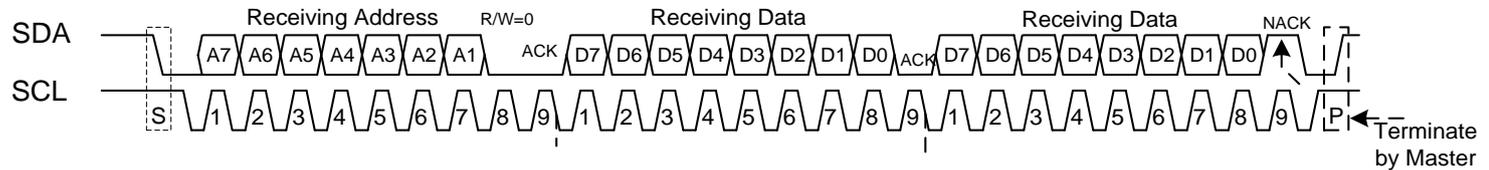
12.4.1.2 MASTER RECEIVER MODE



12.4.1.3 SLAVE TRANSMITTER MODE

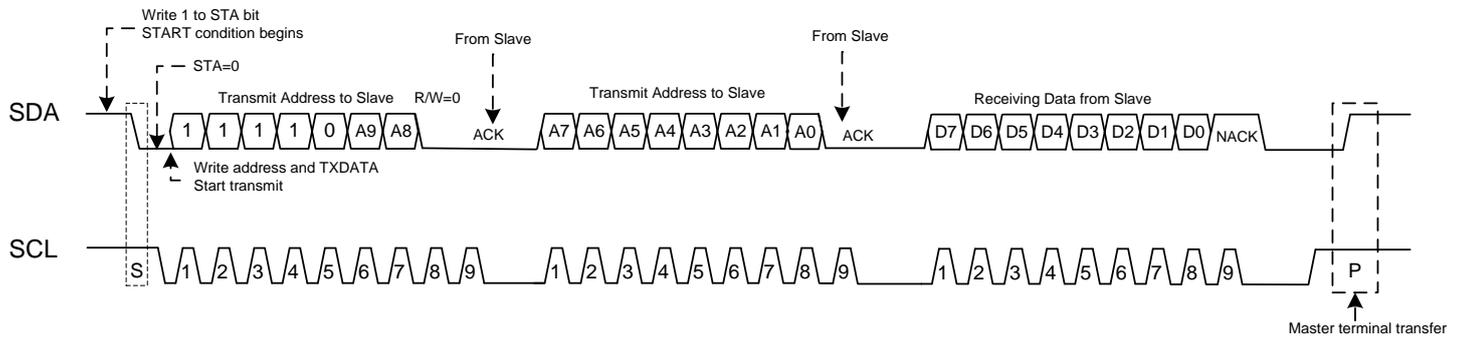


12.4.1.4 SLAVE RECEIVER MODE

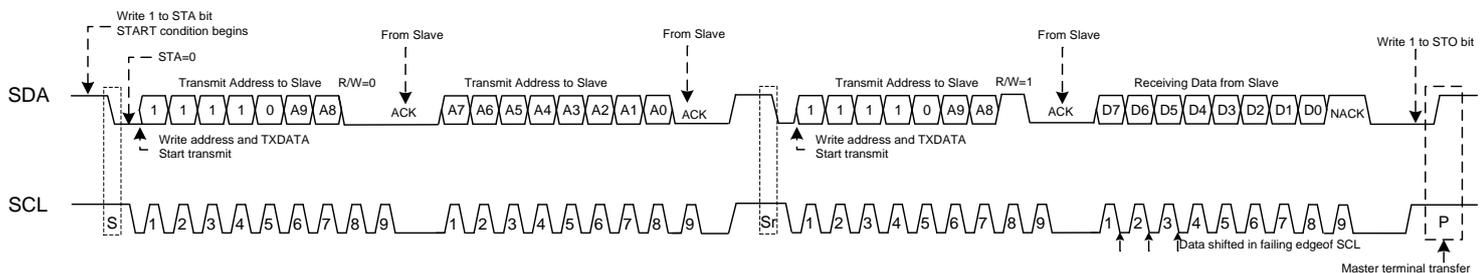


12.4.2 10-BIT ADDRESSING MODES

12.4.2.1 MASTER TRANSMITTER MODE



12.4.2.2 MASTER RECEIVER MODE



12.5 ARBITRATION

In multi-master condition, more than one master may transmit on bus in the same time. It must be decided which master has the control of bus and complete its transmission. Clock synchronization and arbitration are used to configure multi-master transmission.

Clock synchronization is executed by synchronizing the SCL signal with another devices. When two masters want to transmit data in the same time, the clock synchronization will start by the High to Low transition on the SCL. If master 1 pulls the SCL line LOW first, it holds the SCL in LOW status until the SCL line is released to HIGH status. However, if another master still pulls the SCL line LOW, the SCL Low to High transition of master 1 may not change SCL status (SCL line is still LOW). The SCL will transit from LOW to HIGH when the all masters release the SCL line. In the duration, the master1 will wait for SCL transition from LOW to HIGH, and then continue its transmission.

After clock synchronization, the clock of all devices is synchronized with the SCL clock. Arbitration is used to decide which master can complete its transmission by SDA signal. Two masters may send out a START condition and transmit data on bus in the same time, and may be influenced by each other. Arbitration will force one master to lose the control on bus. Data transmission will keep until two masters output different data signal. If one master transmits HIGH status and another master transmits LOW status, the SDA will be pulled low. The master which pulls the SDA line High will detect the different with SDA and loses the control on bus. The master which pulls the SDA line LOW status wins the bus control and continues its transmission. There is no data miss during arbitration.

Arbitration may also be lost in the master receiver mode. Loss of arbitration in this mode can only occur while the I2C block is returning a “not acknowledge” to the bus. Arbitration is lost when another device on the bus pulls this signal low. Since this can occur only at the end of a serial byte, the I2C block generates no further clock pulses.

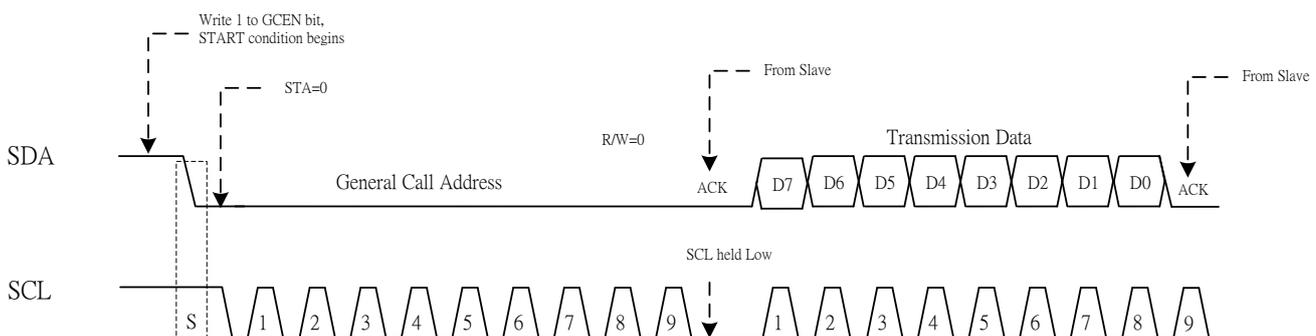
12.6 CLOCK STRETCHING

Clock stretching pauses a transaction by pulling the SCL line LOW. The transaction cannot continue until the line is released HIGH again. Clock stretching is optional.

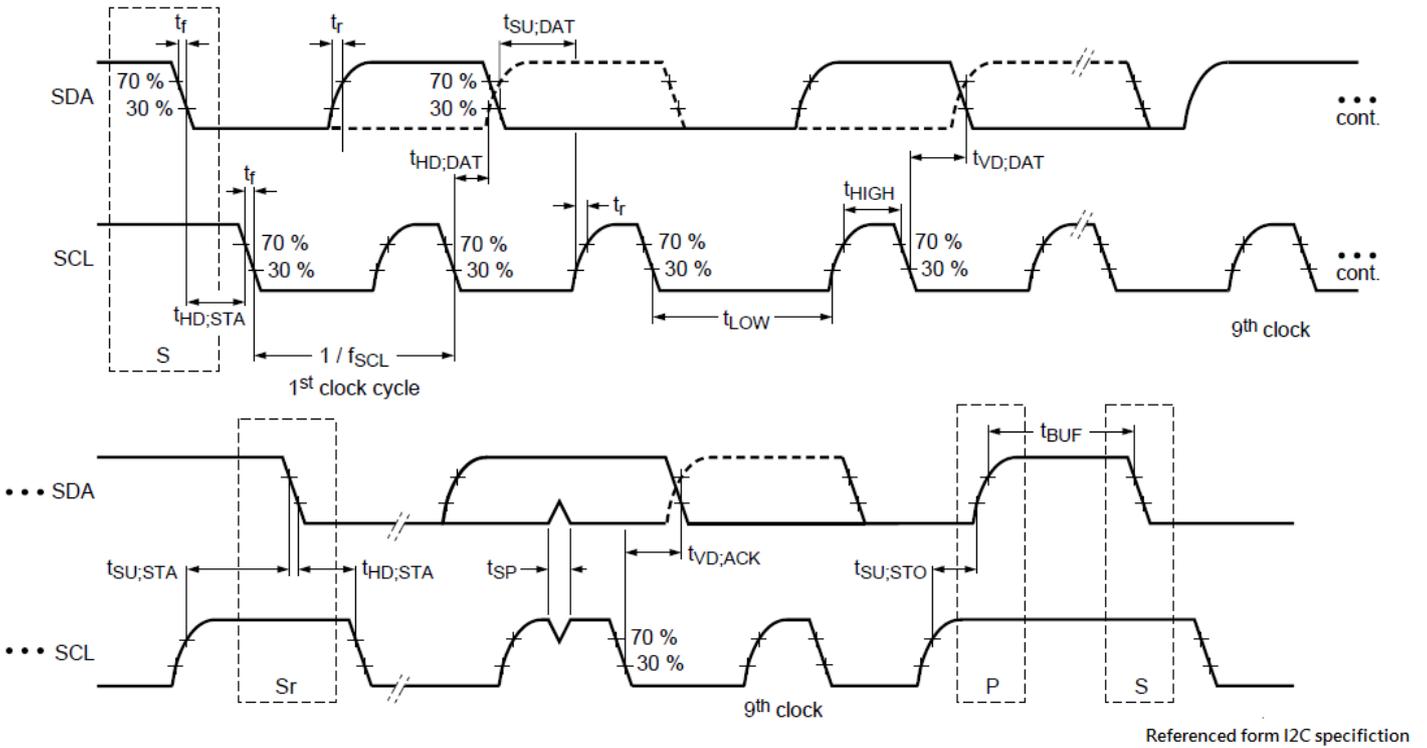
On the byte level, a device may be able to receive bytes of data at a fast rate, but needs more time to store a received byte or prepare another byte to be transmitted. Slaves can then pull the SCL line LOW after reception and acknowledgment of a byte to force the master into a wait state until the slave is ready for the next byte transfer in a type of handshake procedure.

12.7 GENERAL CALL ADDRESS

The general call address is a special address which is reserved as all “0” of 7-bit address and is for addressing every device connected to the I2C-bus at the same time. However, if a device does not need any of the data supplied within the general call structure, it can ignore this address by not issuing an acknowledgment (ACK). If a device does require data from a general call address, it acknowledges this address and behaves as a slave-receiver. The master does not actually know how many devices acknowledged if one or more devices respond. The second and following bytes are acknowledged by every slave-receiver capable of handling this data. A slave who cannot process one of these bytes must ignore it by not-acknowledging. If one or more slaves acknowledge, the not-acknowledge will not be seen by the master. The meaning of the general call address is specified in the second byte



12.8 TIMING CHARACTERISTICS



12.8.1 MASTER TRANSMITTER MODE

t_{HIGH} : $(SCLHT + 1) * I2C_PCLK$ cycle

t_{LOW} : $(SCLLT + 1) * I2C_PCLK$ cycle

$t_{HD;STA}$: $(SCLLT + 4) * I2C_PCLK$ cycle

$t_{HD;DAT}$: $2 * I2C_PCLK$ cycle ~ $3 * I2C_PCLK$ cycle

$t_{SU;STA}$: $(SCLL + 3) * I2C_PCLK$ cycle

$t_{SU;STO}$: $(SCLLT + 2) * I2C_PCLK$ cycle ~ $(SCLLT + 3) * I2C_PCLK$ cycle

12.8.2 SLAVE TRANSMITTER MODE

t_{HIGH} : controlled by Master

t_{LOW} : controlled by Master

$t_{HD;DAT}$: $2 * I2C_PCLK$ cycle ~ $3 * I2C_PCLK$ cycle

$t_{SU;DAT}$: $t_{LOW} - 2 * I2C_PCLK$ cycle ~ $t_{LOW} - 3 * I2C_PCLK$ cycle

12.9 MONITOR MODE

12.9.1 INTERRUPT

All interrupts will occur as normal when the module is in monitor mode. This means that the first interrupt will occur when an address-match is detected (any address received if the MATCH_ALL bit is set, otherwise an address matching one of the four address registers).

Subsequent to an address-match detection, interrupts will be generated after each data byte is received for a slave-write transfer, or after each byte that the module “thinks” it has transmitted for a slave-read transfer. In this second case, the data register will actually contain data transmitted by some other slave on the bus which was actually addressed by the master.

Following all of these interrupts, the processor may read the data register to see what was actually transmitted on the bus.

12.9.2 LOSS of ARBITRATION

In monitor mode, the I2C module will not be able to respond to a request for information by the bus master or issue an ACK). Some other slave on the bus will respond instead. This will most probably result in a lost-arbitration state as far as our module is concerned. Software should be aware of the fact that the module is in monitor mode and should not respond to any loss of arbitration state that is detected. In addition, hardware may be designed into the module to block some/all loss of arbitration states from occurring if those state would either prevent a desired interrupt from occurring or cause an unwanted interrupt to occur. Whether any such hardware will be added is still to be determined.

12.10 I2C REGISTERS

Base Address: 0x4001 8000 (I2C0)
0x4005 A000 (I2C1)

12.10.1 I2C n Control register (I2Cn_CTRL) (n=0,1)

Address Offset: 0x00

Setting of the bits in this register controls operation of the I2C interface.

When STA =1 and the I2C interface is not already in master mode, it enters master mode, checks the bus and generates a START condition if the bus is free. If the bus is not free, it waits for a STOP condition (which will free the bus) and generates a START condition after a delay of a half clock period of the internal clock generator. If the I2C interface is already in master mode and data has been transmitted or received, it transmits a Repeated START condition. STA may be set at any time, including when the I2C interface is in an addressed slave mode.

When STO = 1 in master mode, a STOP condition is transmitted on the I2C bus. When the bus detects the STOP condition, STO is cleared automatically. In slave mode, setting STO bit can recover from an error condition. In this case, no STOP condition is transmitted to the bus. The HW behaves as if a STOP condition has been received and it switches to “not addressed” slave receiver mode.

If STA and STO are both set, then a STOP condition is transmitted on the I2C bus if it the interface is in master mode, and transmits a START condition thereafter. If the I2C interface is in slave mode, an internal STOP condition is generated, but is not transmitted on the bus.

*** Note:**

1. **I2CEN shall be set at last.**
2. **HW will assign SCL0/SCL1 and SDA0/SDA1 pins as output pins with open-drain function instead of GPIO automatically.**
3. **ACK and NACK bits can't both be “1” when receiving data.**
4. **User has to write 1 to ACK or NACK bit in Master mode to continue next RX process.**

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:9 | Reserved | | R | 0 |
| 8 | I2CEN | I2C Interface enable bit 0: Disable. The STO bit is forced to “0”. 1: Enable. I2CEN shall not be used to temporarily release the I2C bus since the bus status is lost when I2CEN resets. The ACK flag should be used instead. | R/W | 0 |
| 7:6 | Reserved | | R | 0 |
| 5 | STA | START bit. 0: No START condition or Repeated START condition will be generated. 1: Cause the I2C interface to enter master mode and transmit a START or a Repeated START condition. Automatically cleared by HW. | R/W | 0 |
| 4 | STO | STOP flag 0: Stop condition idle. 1: Cause the I2C interface to transmit a STOP condition in master mode, or recover from an error condition in slave mode. Automatically cleared by HW. | R/W | 0 |
| 3 | Reserved | | R | 0 |
| 2 | ACK | Assert ACK flag. 0: Master mode→ No function Slave mode→Return a NACK after receiving address or data. 1: An ACK will be returned during the acknowledge clock pulse on SCLn when <ul style="list-style-type: none"> ➢ The address in the Slave Address register has been received. ➢ The General Call address has been received while the General Call enable bit (GCEN) in SLVADDR0 register is set. ➢ A data byte has been received while the I2C is in the master receiver | R/W | 0 |

| | | | | |
|----------|-------------|--|-----|---|
| | | mode. ➤ A data byte has been received while the I2C is in the addressed slave receiver mode. HW will clear after issuing ACK automatically. | | |
| 1 | NACK | Assert NACK flag. 0: No function 1: An NACK will be returned during the acknowledge clock pulse on SCLn when ➤ A data byte has been received while the I2C is in the master receiver mode. HW will clear after issuing NACK automatically. | R/W | 0 |
| 0 | Reserved | | R | 0 |

12.10.2 I2C n Status register (I2Cn_STAT) (n=0,1)

Address Offset: 0x04

Check this register when I2C interrupt occurs, and all status will be cleared automatically by writing I2Cn_CTRL or I2Cn_TXDATA register.

Following events will trigger I2C interrupt if I2C interrupt is enabled in NVIC interrupt controller.

- **START/Repeat START condition**
- **STOP condition**
- **Timeout**
- **Data byte transmitted or received**
- **ACK Transmit or received**
- **NACK Transmit or received**

| Bit | Name | Description | Attribute | Reset |
|--------------|------------|---|-----------|-------|
| 31:16 | Reserved | | R | 0 |
| 15 | I2CIF | I2C Interrupt flag. 0: I2C status doesn't change. 1: Read→I2C status changes. Write→Clear this flag. | R/W | 0 |
| 14:10 | Reserved | | R | 0 |
| 9 | TIMEOUT | Time-out status 0: No Timeout 1: Timeout | R | 0 |
| 8 | LOST_ARB | Lost arbitration 0: Not lost arbitration 1: Lost arbitration | R | 0 |
| 7 | SLV_TX_HIT | 0: No matched slave address. 1: Slave address hit, and is called for TX in slave mode. | R | 0 |
| 6 | SLV_RX_HIT | 0: No matched slave address. 1: Slave address hit, and is called for RX in slave mode. | R | 0 |
| 5 | MST | Master/Slave status 0: I2C is in Slave state. 1: I2C is in Master state. | R | 0 |
| 4 | START_DN | Start done status 0: No START bit. 1: MASTER mode→ a START bit was issued. SLAVE mode→a START bit was received. | R | 0 |
| 3 | STOP_DN | Stop done status 0: No STOP bit. 1: MASTER mode→a STOP condition was issued. SLAVE mode→a STOP condition was received. | R | 0 |
| 2 | NACK_STAT | NACK done status 0 : Not received a NACK 1 : Received a NACK | R | 0 |
| 1 | ACK_STAT | ACK done status 0 : Not received an ACK 1 : Received an ACK | R | 0 |
| 0 | RX_DN | RX done status | R | 0 |

| | | | |
|--|---|--|--|
| | 0: No RX with ACK/NACK transfer. 1: 8-bit RX with ACK/NACK transfer is done. | | |
|--|---|--|--|

12.10.3 I2C n TX Data register (I2Cn_TXDATA) (n=0,1)

Address Offset: 0x08

This register contains the data to be transmitted.

In Master TX mode, CPU writes this register will trigger a TX function. In Slave TX mode, CPU has to write this register before next TX procedure.

| Bit | Name | Description | Attribute | Reset |
|------|-----------|-------------------------|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7:0 | DATA[7:0] | Data to be transmitted. | R/W | 0x00 |

12.10.4 I2C n RX Data register (I2Cn_RXDATA) (n=0,1)

Address Offset: 0x0C

| Bit | Name | Description | Attribute | Reset |
|------|-----------|--|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7:0 | DATA[7:0] | Contains the data received. Read this register when RX_DN = 1. | R | 0x00 |

12.10.5 I2C n Slave Address 0 register (I2Cn_SLVADDR0) (n=0,1)

Address Offset: 0x10

Only used in slave mode. In master mode, this register has no effect.

If this register contains 0x00, the I2C will not acknowledge any address on the bus. Register ADR0 to ADR3 will be cleared to this disabled state on reset.

| Bit | Name | Description | Attribute | Reset |
|-------|-----------|--|-----------|-------|
| 31 | ADD_MODE | Slave address mode. 0 : 7-bit address mode 1: 10-bit address mode | RW | 0 |
| 30 | GCEN | General call address enable bit. 0: Disable 1: Enable general call address (0x0) | RW | 0 |
| 29:10 | Reserved | | R | 0 |
| 9:0 | ADDR[9:0] | The I2C slave address. ADD[9:0] is valid when ADD_MODE = 1 ADD[7:1] is valid when ADD_MODE = 0 | R/W | 0 |

12.10.6 I2C n Slave Address 1~3 register (I2Cn_SLVADDR1~3) (n=0,1)

Address Offset: 0x14, 0x18, 0x1C

| Bit | Name | Description | Attribute | Reset |
|-------|-----------|--|-----------|-------|
| 31:10 | Reserved | | R | 0 |
| 9:0 | ADDR[9:0] | The I2C slave address. ADD[9:0] is valid when ADD_MODE = 1 ADD[7:1] is valid when ADD_MODE = 0 | R/W | 0 |

12.10.7 I2C n SCL High Time register (I2Cn_SCLHT) (n=0,1)

Address Offset: 0x20

* **Note:** $I2C \text{ Bit Frequency} = I2Cn_PCLK / (I2Cn_SCLHT + I2Cn_SCLLT)$

| Bit | Name | Description | Attribute | Reset |
|------|-----------|---|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7:0 | SCLH[7:0] | Count for SCL High Period time SCL High Period Time = (SCLH+1) * I2C0_PCLK cycle | R/W | 0x04 |

12.10.8 I2C n SCL Low Time register (I2Cn_SCLLT) (n=0,1)

Address Offset: 0x24

| Bit | Name | Description | Attribute | Reset |
|------|-----------|---|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7:0 | SCLL[7:0] | Count for SCL Low Period time SCL Low Period Time = (SCLL+1) * I2C0_PCLK cycle | R/W | 0x04 |

12.10.9 I2C n Timeout Control register (I2Cn_TOCTRL) (n=0,1)

Address Offset: 0x2C

Timeout happens when Master/Slave SCL remained LOW for:

$$TO * 32 * I2C0_PCLK \text{ cycle.}$$

When I2C timeout occurs, the I2C transfer will return to "IDLE" state, and issue a TO interrupt to inform user. That means SCL/SDA will be released by HW after timeout. User can issue a STOP after timeout interrupt occurred in Master mode.

Time-out status will be cleared automatically by writing I2Cn_CTRL or I2Cn_TXDATA register.

| Bit | Name | Description | Attribute | Reset |
|-------|----------|--|-----------|-------|
| 31:16 | Reserved | | R | 0 |
| 15:0 | TO[15:0] | Count for checking Timeout. 0: Disable Timeout checking N: Timeout period time = $N * 32 * I2Cn_PCLK \text{ cycle}$ | R/W | 0x0 |

12.10.10 I2C n Monitor Mode Control register (I2Cn_MMCTRL) (n=0,1)

Address Offset: 0x30

This register controls the Monitor mode which allows the I2C module to monitor traffic on the I2C bus without actually participating in traffic or interfering with the I2C bus.

In Monitor mode, SDA output will be forced high to prevent the I2C module from outputting data of any kind (including ACK) onto the I2C data bus. Depending on the state of the SCLOEN bit, the SCL output may be also forced high to prevent the module from having control over the I2C clock line.

* **Note:** *The SCLOEN and MATCH_ALL bits have no effect if MMEN bit is '0' (i.e. if the module is NOT in monitor mode).*

| Bit | Name | Description | Attribute | Reset |
|-------------|-----------|--|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 2 | MATCH_ALL | Match address selection 0: Interrupt will only be generated when the address matches one of the values in I2Cn_SLVADDR0~3 register. 1: If I2C is in monitor mode, an interrupt will be generated on ANY address received. This will enable the part to monitor all traffic on the bus. | R/W | 0 |
| 1 | SCLOEN | SCL output enable bit. 0: SCL output will be forced high. 1: I2C module may act as a slave peripheral just like in normal operation, the I2C holds the clock line low until it has had time to respond to an I2C interrupt. | R/W | 0 |
| 0 | MMEN | Monitor mode enable bit. 0: Disable 1: Enable | R/W | 0 |

13 UNIVERSAL SYNCHRONOUS AND ASYNCHRONOUS RECEIVER AND TRANSMITTER (USART)

13.1 OVERVIEW

The USART offers a flexible means of full-duplex data exchange with external equipment requiring an industry standard NRZ asynchronous serial data format. The serial interface is applied to low speed data transfer and communicate with low speed peripheral devices.

The USART offers a very wide range of baud rates using a fractional baud rate generator. It supports both synchronous one-way communication and single wire communication. It also supports the LIN (local interconnection network), Smart card Protocol and IrDA (infrared data association) SIR ENDEC specifications, and modem operations (CTS/RTS).

13.2 FEATURES

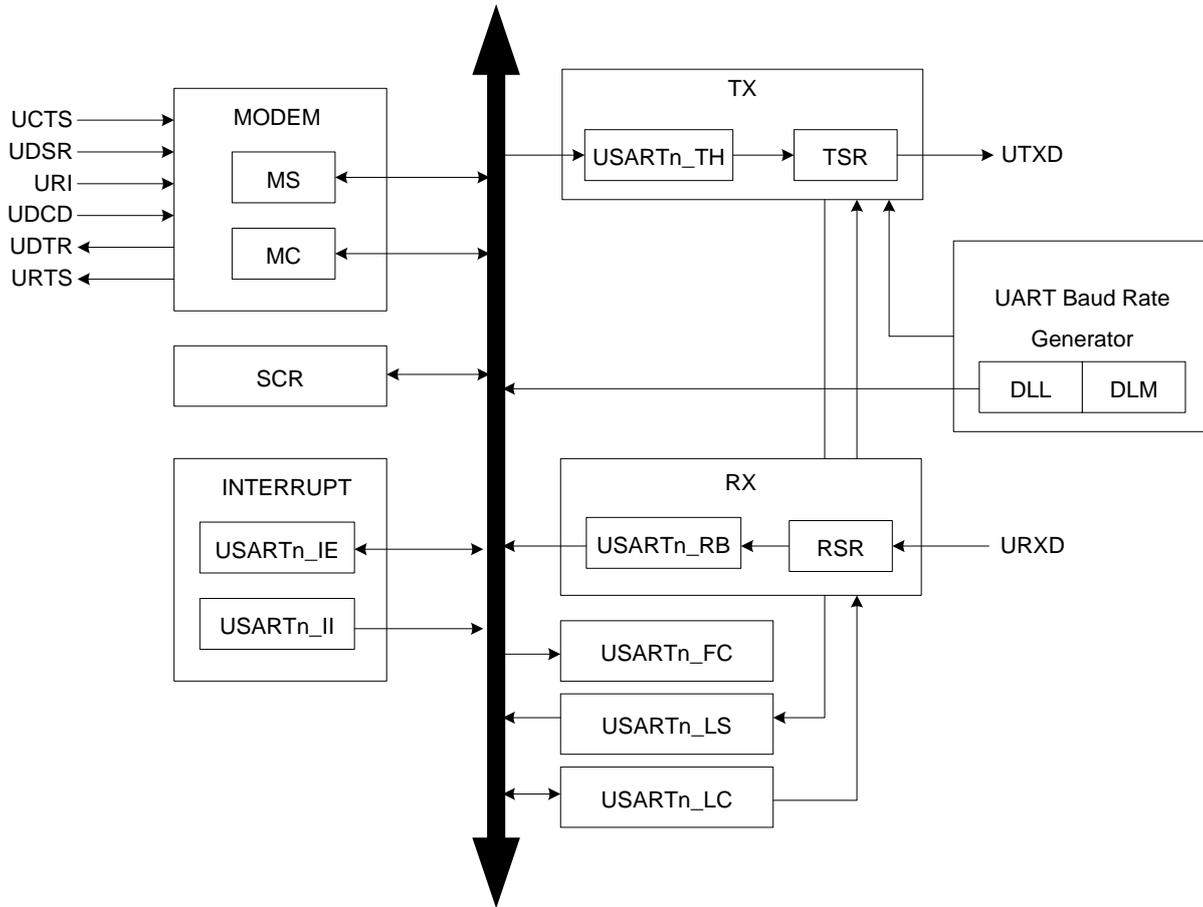
- Full-duplex, 2-wire asynchronous data transfer.
- Single-wire half-duplex communication
- Transmitter clock output for synchronous transmission
- 16-byte receive and transmit FIFOs
- Register locations conform to 16550 industry standard.
- Receiver FIFO trigger points at 1, 4, 8, and 14 bytes.
- Built-in baud rate generator.
- Software or hardware flow control.
- EIA-485 9-bit mode support with output enable.
- Modem control signals (CTS/RTS).
- ISO 7816-3 compliant Smart card interface.
- IrDA support.

13.3 PIN DESCRIPTION

| Pin Name | Type | Description | GPIO Configuration |
|--------------|------|---|-----------------------------------|
| UTXDn | O | Serial Transmit data. | |
| URXDn | I | Serial Receive data. | Depends on GPIO _n _CFG |
| USCKn | O | Serial clock of Synchronous mode (Master) | |
| <u>UCTSn</u> | I | Clear to Send. When low, this indicates that the MODEM or data set is ready to exchange data. The <u>CTS</u> signal is a MODEM status input whose conditions can be tested by reading bit 4 (CTS) of USART _n _MS register. | Depends on GPIO _n _CFG |
| <u>URTSn</u> | O | Request to Send. RS-485 direction control pin. When low, this informs the MODEM or data set that the UART is ready to exchange data. The <u>RTS</u> output signal can be set to an active low by programming bit 1 (RTS) of USART _n _MC register. Loop mode operation holds this signal | |

| | | | |
|--------------|---|--|-----------------------------------|
| | | in its inactive state. | |
| <u>UDTRn</u> | O | Data Terminal Ready. When low, this informs the MODEM or data set that the UART is ready to establish a communications link. The <u>DTR</u> output signal can be set to an active low by programming bit 0 (DTR) of <u>USARTn_MC</u> register to a high level. Loop mode operation holds this signal in its inactive state. | |
| <u>UDSRn</u> | I | Data Set Ready. When low, this indicates that the MODEM or data set is ready to establish the communications link with the UART. The <u>DSR</u> signal is a MODEM status input whose condition can be tested by reading bit 5 (DSR) of <u>USARTn_MS</u> register. | Depends on GPIO _n _CFG |
| <u>UDCDn</u> | I | Data Carrier Detect. When low, indicates that the data carrier has been detected by the MODEM or data set. The <u>DCD</u> signal is a MODEM status input whose condition can be tested by reading bit 7 (DCD) of <u>USARTn_MS</u> register. | Depends on GPIO _n _CFG |
| <u>URIn</u> | I | Ring Indicator When low, this indicates that a telephone ringing signal has been received by the MODEM or data set. The <u>RI</u> signal is a MODEM status input whose condition can be tested by reading bit 6 (RI) of <u>USARTn_MS</u> register. | Depends on GPIO _n _CFG |

13.4 BLOCK DIAGRAM



13.5 EIA-485/RS-485 MODES

The RS-485/EIA-485 feature allows the USART to be configured as an addressable slave receiver. The addressable slave receiver is one of multiple slaves receivers controlled by a single master.

The USART master transmitter will identify an address character by setting the parity (9th) bit to '1'. For data characters, the parity bit is set to '0'.

Each USART slave receiver can be assigned a unique address. The slave can be programmed to either manually or automatically reject data following an address which is not theirs.

In RS-485 mode, PS bits in [USARTn_LC](#) register shall be selected as forced 1 stick parity (Address), or forced stick 0 parity (Data) by SW. In addition, the word length shall be 8 bits by setting WLS bits in [USARTn_LC](#) register to 11b by SW.

13.5.1 RS-485/EIA-485 NORMAL MULTIDROP MODE (NMM)

Setting the NMMEN bit in [USARTn_RS485CTRL](#) register enables this mode. In this mode, an address is detected when a received byte causes the USART to set the parity error and generate an interrupt.

If the receiver is disabled (RXEN = 0 in [USARTn_RS485CTRL](#) register), any received data bytes will be ignored and will not be stored in the RXFIFO. When an address byte is detected (parity bit = '1') it will be placed into the RXFIFO and a parity error (PE) Interrupt will be generated. The processor can then read the address byte and decide whether or not to enable the receiver to accept the following data.

While the receiver is enabled (RXEN = 1 in [USARTn_RS485CTRL](#) register), all received bytes will be accepted and stored in the RXFIFO regardless of whether they are data or address. When an address character is received a parity error interrupt will be generated and the processor can decide whether or not to disable the receiver.

13.5.2 RS-485/EIA-485 AUTO ADDRESS DETECTION (AAD) MODE

When both NMMEN (9-bit mode enable) bit and AADEN (AAD mode enable) bit in [USARTn_RS485CTRL](#) register are set, the USART is in auto address detect mode.

In this mode, the receiver will compare any address byte received (parity = '1') to the 8-bit value programmed into the [USARTn_RS485ADRMATCH](#) register.

If the receiver is disabled (RXEN = 0 in [USARTn_RS485CTRL](#) register), any received byte will be discarded if it is either a data byte or an address byte which is different from the value in [USARTn_RS485ADRMATCH](#) register.

When a matching address character is detected it will be pushed onto the RXFIFO along with the parity bit, and the receiver will be automatically enabled (RXEN bit will be set by HW). The receiver will also generate an RX Data Available (RDA) Interrupt.

While the receiver is enabled (RXEN = 1 in [USARTn_RS485CTRL](#) register), all bytes received will be accepted and stored in the RXFIFO until an address byte which is different from the MATCH value is received. When this occurs, the receiver will be automatically disabled by HW (RXEN bit will be cleared by HW), the received non-matching address character will not be stored in the RXFIFO.

13.5.3 RS-485/EIA-485 AUTO DIRECTION CONTROL (ADC)

RS485/EIA-485 mode includes the option of allowing the transmitter to automatically control the state of the DIR pin as a direction control output signal. Set ADCEN bit in [USARTn_RS485CTRL](#) register to enable this feature.

The ADCEN bit takes precedence over all other mechanisms controlling the direction control pin with the exception of loopback mode.

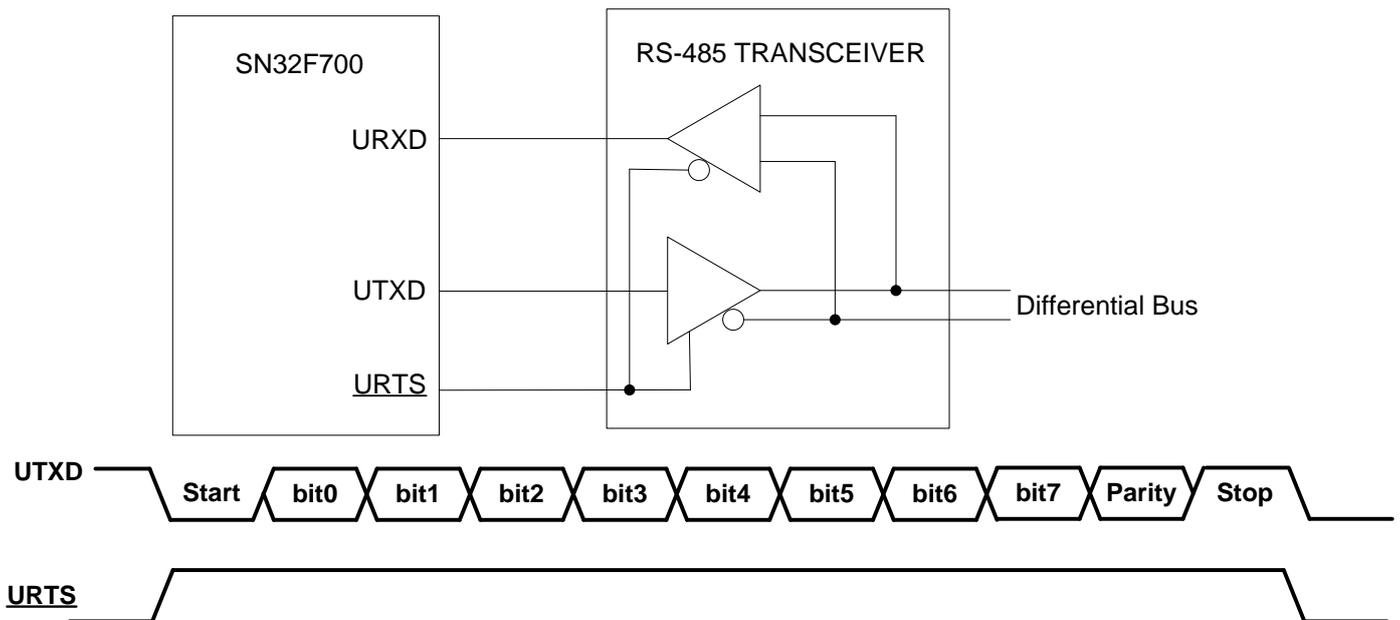
13.5.4 RS485/EIA-485 DRIVER DELAY TIME

The driver delay time is the delay between the last stop bit leaving the TXFIFO and the de-assertion of URTS. This delay time can be programmed in the 8-bit USARTn_RS485DLYV register. The delay time is in periods of the baud clock. Any delay time from 0 to 255 bit times may be used.

13.5.5 RS485/EIA-485 OUTPUT INVERSION

The polarity of the direction control signal on the URTS pin can be reversed by programming OINV bit in USARTn_RS485CTRL register. When OINV bit is set, the direction control pin will be driven to logic 1 (driven LOW) when the transmitter has data waiting to be sent. The direction control pin will be driven to logic 0 (driven High) once the last bit of data has been transmitted.

13.5.6 RS485/EIA-485 FRAME STRUCTURE



13.6 BAUD RATE CALCULATION

The USART baud rate is calculated as:

$$\text{UART}_{\text{BAUDRATE}} = \frac{\text{USARTn_PCLK}}{\text{Oversampling} \times (256 \times \text{DLM} + \text{DLL}) \times (1 + \text{DIVADDVAL} / \text{MULVAL})}$$

Where USARTn_PCLK is the peripheral clock, USARTn_DLM and USARTn_DLL are the standard UART baud rate divider registers, and DIVADDVAL and MULVAL are USART fractional baud rate generator specific parameters in USARTn_FD register.

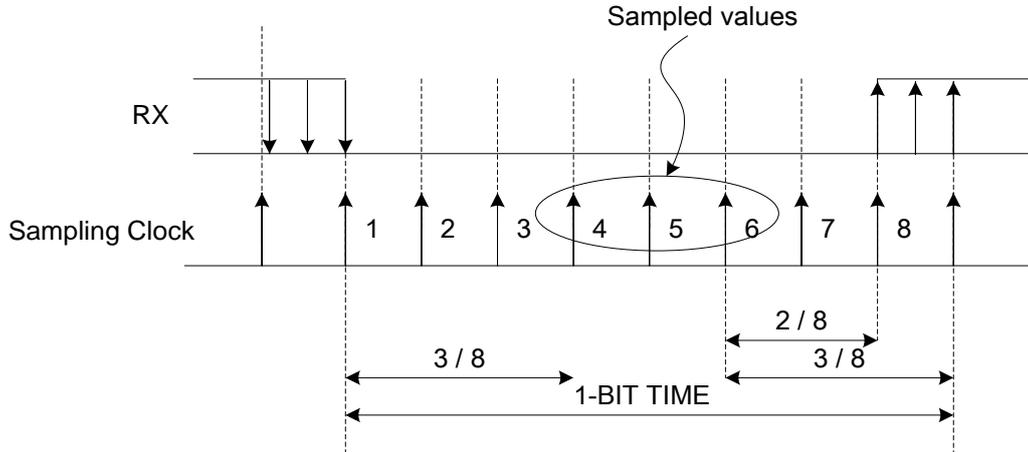
The value of MULVAL and DIVADDVAL should comply with the following conditions:

1. $1 \leq \text{MULVAL} \leq 15$
2. $0 \leq \text{DIVADDVAL} \leq 14$
3. $\text{DIVADDVAL} < \text{MULVAL}$
4. Oversampling is 8 or 16

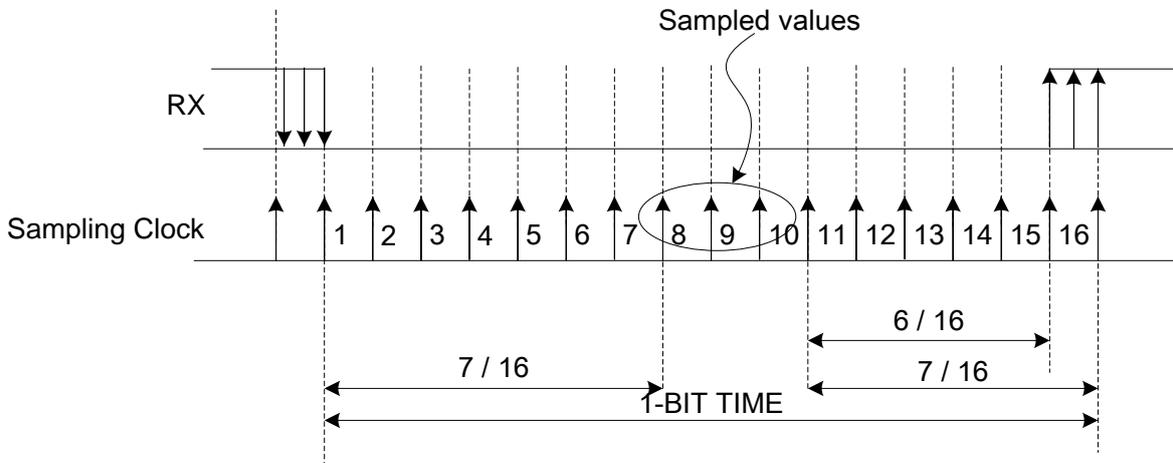
The value of the USARTn_FD register should not be modified while transmitting/receiving data or data may be lost or corrupted.

The oversampling method can be selected by programming the OVER8 bit in [USARTn_FD](#) register and can be either 16 or 8 times the baud rate clock.

- OVER8=1: Oversampling by 8 to achieve higher speed (up to $USARTn_PCLK/8$). In this case the maximum receiver tolerance to clock deviation is reduced.



- OVER8=0: Oversampling by 16 to increase the tolerance of the receiver to clock deviations. In this case, the maximum speed is limited to maximum $USARTn_PCLK/16$



If the [USARTn_FD](#) register value does not comply with these two requests, then the fractional divider output is undefined. If DIVADDVAL is zero then the fractional divider is disabled, and the clock will not be divided.

USART can operate with or without using the Fractional Divider. The desired baud rate can be achieved using several different Fractional Divider settings. The following algorithm illustrates one way of finding a set of DLM, DLL, MULVAL, and DIVADDVAL values. Such set of parameters yields a baud rate with a relative error of less than 1.1% from the desired one.

The following example illustrates selecting the DIVADDVAL, MULVAL, DLM, and DLL to generate $BR = 115200$ when $USARTn_PCLK = 12\text{ MHz}$, and Oversampling = 16.

$$UART_{BAUDRATE} = \frac{USARTn_PCLK}{Oversampling \times (256 \times DLM + DLL) \times (1 + DIVADDVAL / MULVAL)}$$

$$115200 = \frac{12000000}{16 \times (256 \times DLM + DLL) \times (1 + DIVADDVAL / MULVAL)}$$

$$(256 \times DLM + DLL) \times (1 + DIVADDVAL / MULVAL) = 6.51$$

Since the value of MULVAL and DIVADDVAL should comply with the following conditions:

1. $1 \leq \text{MULVAL} \leq 15$
2. $0 \leq \text{DIVADDVAL} \leq 14$
3. $\text{DIVADDVAL} < \text{MULVAL}$

Thus, the suggested UART settings would be: DLM = 0, DLL = 4, DIVADDVAL = 5, and MULVAL = 8 (fill in 7 in the MULVAL bits). The baud rate generated is 115384, and has a relative error of 0.16% from the originally specified 115200.

13.7 MODEM CONTROL (MC)

If Auto-RTS mode is enabled, the USART's receiver FIFO hardware controls the URTS output of the USART. If the auto-CTS mode is enabled, the USART's transmitter will only start sending if the UCTS pin is low.

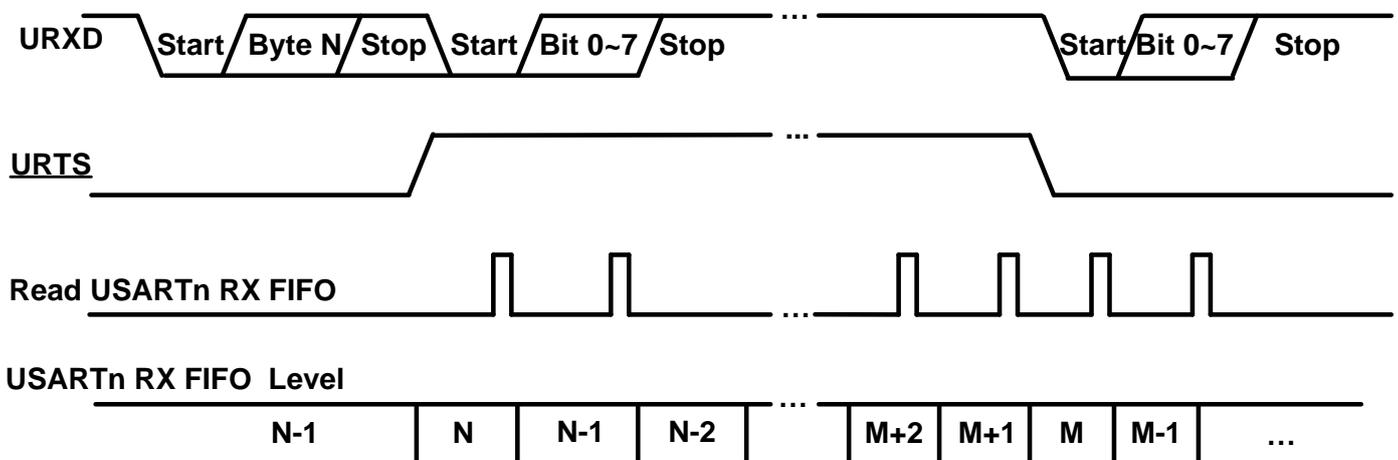
13.7.1 AUTO-RTS

The Auto-RTS function is enabled by setting the RTSEN bit. Auto-RTS data flow control originates in the USARTn_RB module and is linked to the programmed receiver FIFO trigger level. If auto-RTS is enabled, the data-flow is controlled as follows:

When the receiver FIFO level reaches the programmed trigger level, URTS is deasserted (to a high value). It is possible that the sending USART sends an additional byte after the trigger level is reached (assuming the sending USART has another byte to send) because it might not recognize the deassertion of URTS until after it has begun sending the additional byte. URTS is automatically reasserted (to a low value) once the receiver FIFO has reached the previous trigger level. The reassertion of URTS signals the sending USART to continue transmitting data.

If Auto-RTS mode is disabled, the RTSEN bit controls the URTS output of the USART. If Auto-RTS mode is enabled, hardware controls the RTS output, and the actual value of URTS will be copied in the URTS Control bit of the USART. As long as Auto-RTS is enabled, the value of the RTS Control bit is read-only for software.

Example: Suppose the USART operating in type 16550 mode has the trigger level in USARTn_FIFOCtrl register set to 0x2, then, if Auto-RTS is enabled, the USART will deassert the URTS output as soon as the receive FIFO contains 8 bytes. The URTS output will be reasserted as soon as the receive FIFO hits the previous trigger level: 4 bytes.

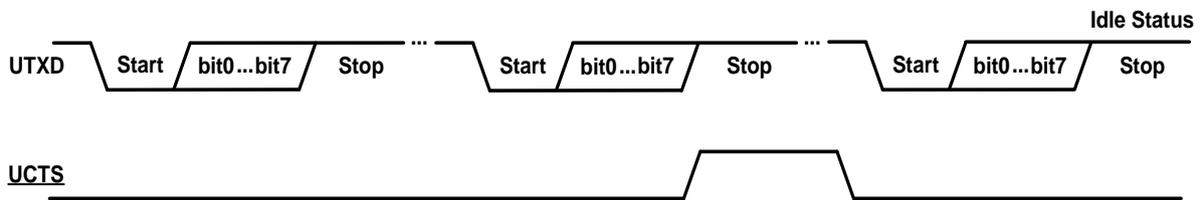


13.7.2 AUTO-CTS

The Auto-CTS function is enabled when CTSEN=1. If Auto-CTS is enabled, the transmitter circuitry checks the UCTS input before sending the next data byte. When UCTS is active (low), the transmitter sends the next byte. To stop the transmitter from sending the following byte, UCTS must be released before the middle of the last stop bit that is currently being sent. In Auto-CTS mode, a change of the UCTS signal does not trigger a modem status interrupt unless the CTS Interrupt Enable bit is set, but the DCTS bit in the USARTn_MS register will be set.

| MSIE | CTSEN | Delta CTS (DCTS) | Delta DCD (DDCD) or Trailing edge RI (TERI) or Delta DSR (DDSR) | MODEM status interrupt |
|------|-------|------------------|---|------------------------|
| 0 | X | X | X | X |
| 1 | 0 | 0 | 0 | X |
| 1 | 0 | 1 | X | ○ |
| 1 | 0 | X | 1 | ○ |
| 1 | 1 | X | 0 | X |
| 1 | 1 | X | 1 | ○ |
| 1 | 1 | 0 | 0 | X |
| 1 | 1 | 1 | X | ○ |
| 1 | 1 | X | 1 | ○ |

The Auto-CTS function typically eliminates the need for CTS interrupts. When flow control is enabled, a UCTS state change does not trigger host interrupts because the device automatically controls its own transmitter. Without Auto-CTS, the transmitter sends any data present in the transmit FIFO and a receiver overrun error can result.



During transmission of the second character the UCTS signal is negated. The third character is not sent thereafter. The USART maintains 1 on UTXD as long as UCTS is negated (high). As soon as UCTS is asserted, transmission resumes and a start bit is sent followed by the data bits of the next character.

13.8 AUTO-BAUD FLOW

13.8.1 AUTO-BAUD

The USART auto-baud function can be used to measure the incoming baud rate based on the "AT" protocol (Hayes command). If enabled the auto-baud feature will measure the bit time of the receive data stream and set the divisor latch registers USARTn_DLM and USARTn_DLL accordingly.

Auto-baud function is started by setting the START bit in USARTn_ABCTRL register, and can be stopped by clearing the START bit. The START bit will clear once auto-baud has finished and reading the bit will return the status of auto-baud (pending/finished). When auto-baud function is started, FIFO will be cleared, not available to write the TX FIFO, and the transmitter will stop transmitting until auto-baud function finishes or be stopped.

Two auto-baud measuring modes are available which can be selected by the MODE bit in USARTn_ABCTRL register. In Mode 0 the baud rate is measured on two subsequent falling edges of the USART RX pin (the falling edge of the start bit and the falling edge of the least significant bit). In Mode 1 the baud rate is measured between the falling edge and the subsequent rising edge of the USART RX pin (the length of the start bit).

The AUTORESTART bit in USARTn_ABCTRL register can be used to automatically restart baud rate measurement if a timeout occurs (the rate measurement counter overflows). If this bit is set, the rate measurement will restart at the next falling edge of the URXD pin.

The auto-baud function can generate two interrupts.

- The ABTOINT interrupt in [USARTn_I1](#) register will get set if the interrupt is enabled (ABTOIE bit in [USARTn_IE](#) register is set and the auto-baud rate measurement counter overflows).
- The ABEOINT interrupt in [USARTn_I1](#) register will get set if the interrupt is enabled (ABTOIE bit in [USARTn_IE](#) register is set and the auto-baud has completed successfully).

The auto-baud interrupts have to be cleared by setting the corresponding ABTOINTCLR and ABEOIE bits in [USARTn_IE](#) register.

The fractional baud rate generator must be disabled (DIVADDDVAL = 0) during auto-baud. Also, when auto-baud is used, any write to [USARTn_DLM](#) and [USARTn_DLL](#) registers should be done before [USARTn_ABCCTRL](#) register write. The minimum and the maximum baud rates supported by USART are a function of USARTn_PCLK and the number of data bits, stop bits and parity bits.

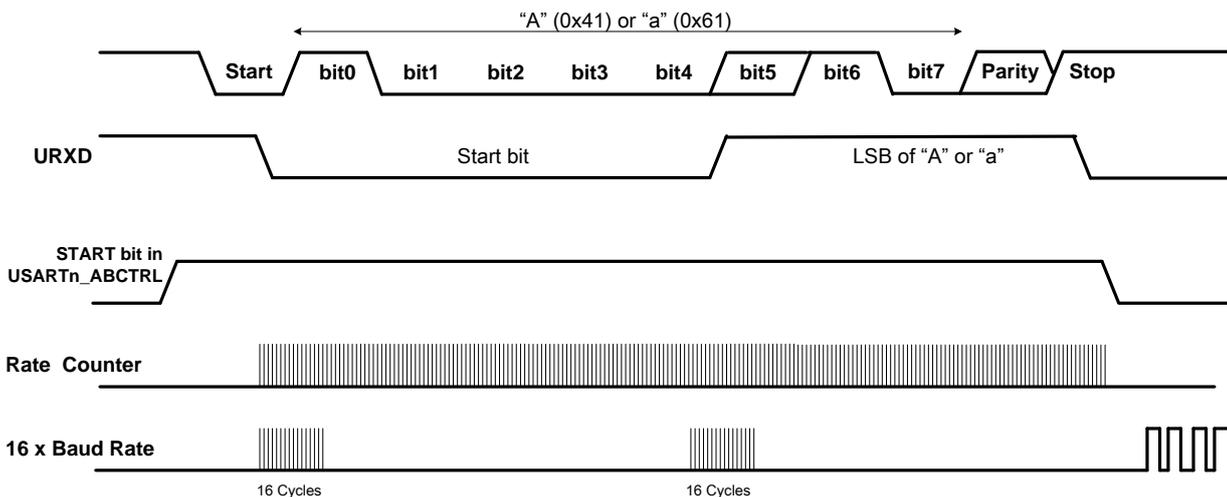
$$ratemin = \frac{2 \times PCLK}{16 \times 2^{15}} \leq \text{USART}_{baudrate} \leq \frac{PCLK}{16 \times (2 + databits + paritybits + stopbits)} = ratemax$$

13.8.2 AUTO-BAUD MODES

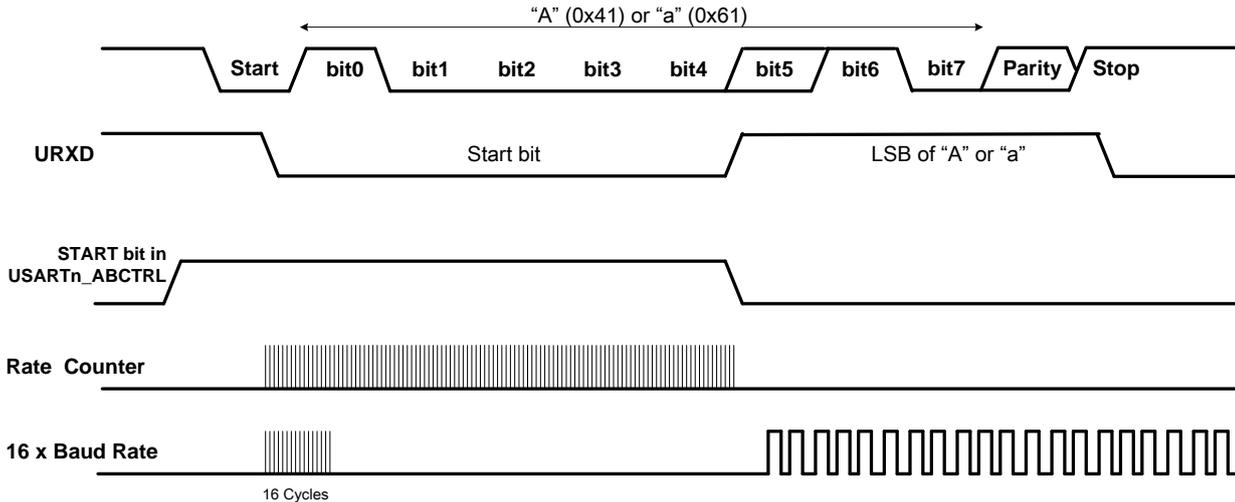
When the SW is expecting an “A” command, it configures the USART with the expected character format and sets the ACR Start bit. The initial values in the divisor latches DLM and DLL don’t care. Because of the “A” or “a” ASCII coding (“A” = 0x41, “a” = 0x61), the USART Rx pin sensed start bit and the LSB of the expected character are delimited by two falling edges. When the ACR Start bit is set, the auto-baud protocol will execute the following phases:

1. On START bit setting, the baud rate measurement counter is reset and the RSR is reset. The RSR baud rate is switched to the highest rate.
2. A falling edge on URXD pin triggers the beginning of the start bit. The rate measuring counter will start counting USARTn_PCLK cycles.
3. During the receipt of the start bit, 16 pulses are generated on the RSR baud input with the frequency of the USART input clock, guaranteeing the start bit is stored in the RSR.
4. During the receipt of the start bit (and the character LSB for MODE = 0 in [USARTn_ABCTRL](#) register), the rate counter will continue incrementing with the pre-scaled USART input clock (USARTn_PCLK).
5. If MODE = 0, the rate counter will stop on next falling edge of the USART RX pin. If MODE = 1, the rate counter will stop on the next rising edge of the URXD pin.
6. The rate counter is loaded into [USARTn_DLM/USARTn_DLL](#) and the baud rate will be switched to normal operation. After setting the DLM/DLL, the end of auto-baud interrupt ABEOINT in [USARTn_I1](#) register will be set, if enabled. The RSR will now continue receiving the remaining bits of the character.

➤ **AUTO-BAUD RATE MODE 0 Waveform**



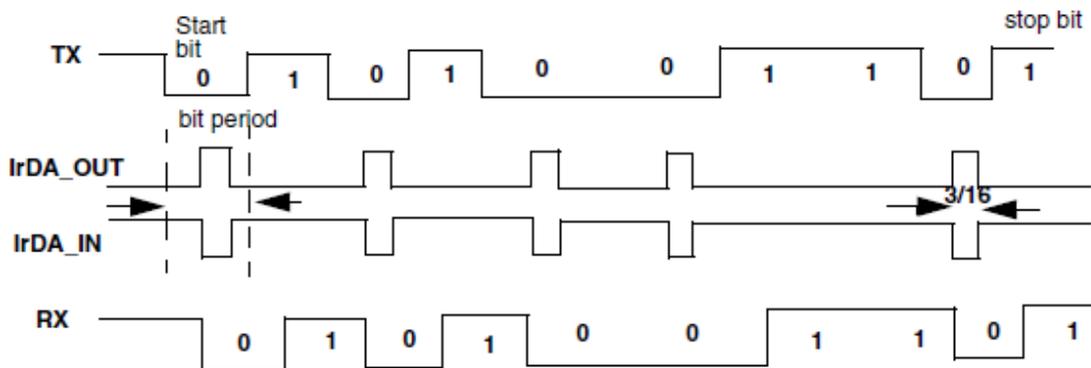
➤ **AUTO-BAUD RATE MODE 1 Waveform**



13.9 SERIAL IRDA (SIR) MODE

The IrDA mode is enabled by setting the USARTEN bit to 1 and MODE[2:0] = 010b in [USARTn_CTRL](#) register. The SIR Transmit encoder modulates the Non Return to Zero (NRZ) transmit bit stream output from USART. The output pulse stream is transmitted to an external output driver and infrared LED. USART supports only bit rates up to 115.2Kbps for the SIR ENDEC. In normal mode the transmitted pulse width is specified as 3/16 of a bit period. The SIR receive decoder demodulates the Return-to-Zero bit stream from the infrared detector and outputs the received NRZ serial bit stream to USART. The decoder input is normally HIGH (marking state) in the Idle state. The transmit encoder output has the opposite polarity to the decoder input. A start bit is detected when the decoder input is low.

- IrDA is a half-duplex communication protocol. If the Transmitter is busy (i.e. the USART is sending data to the IrDA encoder), any data on the IrDA receive line will be ignored by the IrDA decoder and if the Receiver is busy (USART is receiving decoded data from the USART), data on the TX from the USART to IrDA will not be encoded by IrDA. While receiving data, transmission should be avoided as the data to be transmitted could be corrupted.
- A '0' is transmitted as a high pulse and a '1' is transmitted as a '0'. The width of the pulse is specified as 3/16th of the selected bit period in normal mode.



- The SIR decoder converts the IrDA compliant receive signal into a bit stream for USART.
- The SIR receive logic interprets a high state as 1 and low pulses as 0.
- The transmit encoder output has the opposite polarity to the decoder input. The SIR output is 0 when Idle.
- In IrDA mode, the STOP bits must be configured to "1 stop bit".
- The IrDA specification requires the acceptance of pulses greater than 1.41 us. The acceptable pulse width is programmable. Glitch detection logic on the receiver filters out pulses of width less than 2 x low-power baud rate. Pulses of width greater than 2 low-power baud rate will be accepted as a pulse.
- The receiver can communicate with a low-power transmitter. In low-power mode the pulse width is not maintained at 3/16 of the bit period. Instead, the width of the pulse is 3 times the low-power baud rate which can be a minimum of 1.42 MHz. Generally the low-power baud rate is 1.8432 MHz (1.42 MHz < low-power baud rate < 2.12 MHz).

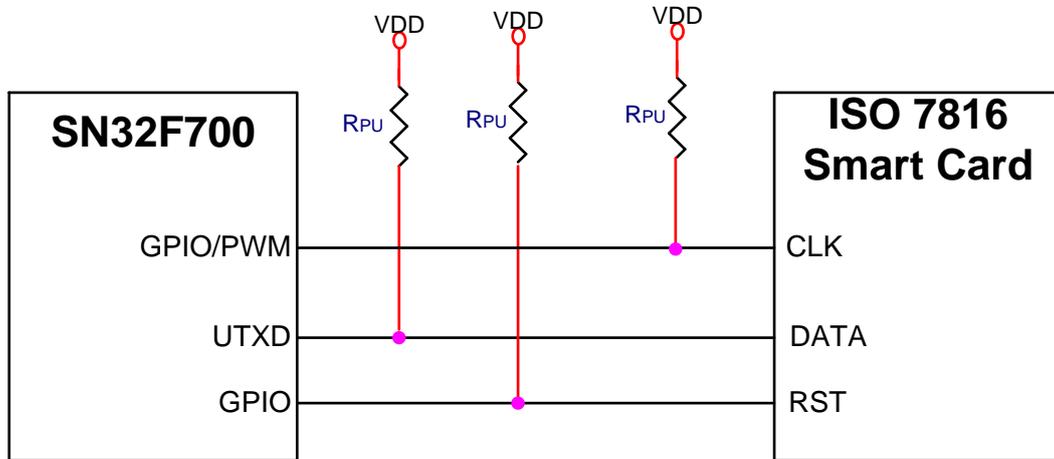
- If FIXPULSEEN= 0 in USARTn_FD register, the low level pulse width shall $> (2/16 \cdot \text{baud cycle})$ for receiver to be accepted as a low pulse; if FIXPULSEEN = 1, the low level pulse width shall $> (1/2 \cdot \text{IrDA Transmitter Pulse Width})$ for receiver to be accepted as a low pulse.
- The PULSEDIV bits are used to select the pulse width when the fixed pulse width mode is used in IrDA mode when FIXPULSEEN = 1. The value of these bits should be set so that the resulting pulse width is at least 1.63 μs .
- OVER8 bit in USARTn_FD register must be 0 in IrDA mode.

| IrDA Pulse Width | | | |
|------------------|---------------|-------|-----------------------------------|
| FIXPULSEEN | PULSEDIV[2:0] | OVER8 | IrDA Transmitter Pulse Width (us) |
| 0 | X | 0 | 3 / (16 x Baud rate) |
| 1 | 0 | 0 | 2 x T_{PCLK} |
| 1 | 1 | 0 | 4 x T_{PCLK} |
| 1 | 2 | 0 | 8 x T_{PCLK} |
| 1 | 3 | 0 | 16 x T_{PCLK} |
| 1 | 4 | 0 | 32 x T_{PCLK} |
| 1 | 5 | 0 | 64 x T_{PCLK} |
| 1 | 6 | 0 | 128 x T_{PCLK} |
| 1 | 7 | 0 | 256 x T_{PCLK} |

13.10 SMART CARD MODE

The Smart card mode is enabled by setting the USARTEN bit to 1 and MODE[2:0] = 011b in [USARTn_CTRL](#) register, the USART provides bidirectional serial data on the open-drain UTXD pin. No URXD pin is used in this mode. If a clock source is needed as an oscillator source into the Smart Card, a timer match or PWM output can be used in cases when a higher frequency clock is needed that is not synchronous with the data bit rate.

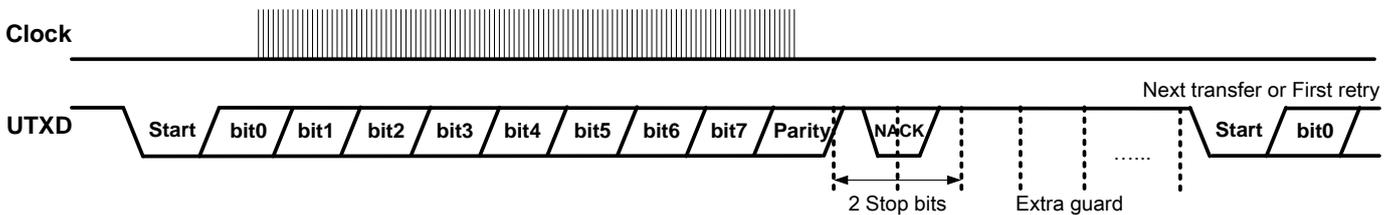
The USCLK pin may not be adequate for most asynchronous cards since it will output synchronously with the data and the data bit rate. SW must use timers to implement character and block waiting times instead.



13.10.1 SMART CARD SETUP PROCEDURE

A T = 0 protocol transfer consists of 8-bits of data, an even parity bit, and two stop bits that allow for the receiver of the particular transfer to flag parity errors through the NACK response. Extra guard bits may be added according to card requirements.

If no NACK is sent, the next byte may be transmitted immediately after the last guard bit. If the NACK is sent, the transmitter will retry sending the byte until successfully received or until the SCICTRL retry limit has been met.



The smart card must be set up with the following considerations:

- Program [SYS1_PRST](#) register so that the USART is not continuously reset.
- Program USARTnPRE bits in [SYS1_APBPC1](#) register for an initial USART frequency of 3.58 MHz.
- If necessary, program the USARTn_DLM and USARTn_DLL to 00 and 01 respectively, to pass the USART clock through without division.
- Program the [USARTn_LC](#) register for 8-bit characters, parity enabled, even parity.
- Program [USARTn_SCICTRL](#) register to enable the smart card feature with the desired options, and HW enables a USART TXD function automatically.
- Set up one or more timer(s) to provide timing as needed for ISO-7816 startup.
- Program USARTnCLKEN bit in [SYS1_AHBCLKEN](#) register to enable the USART clock.

Thereafter, SW should monitor card insertion, handle activation, wait for answer to reset as described in ISO7816-3.

13.11 SYNCHRONOUS MODE

The synchronous mode is selected by writing the MODE bits to 100b in [USARTn_CTRL](#) register.

The USART allows the user to control a bidirectional synchronous serial communications in master mode. The SCLK pin is the output of the USART transmitter clock. No clock pulses are sent to the SCLK pin during start bit and stop bit.

The CPOL bit in [USARTn_CTRL](#) register allows the user to select the clock polarity, and the CPHA bit allows the user to select the phase of the clock.

During the Idle state, preamble and send break, the external SCLK clock is not activated. In synchronous mode the USART transmitter works exactly like in asynchronous mode. But as SCLK is synchronized with TX (according to CPOL and CPHA), the data on TX is synchronous.

In synchronous mode, the USART receiver works in a different manner compared to the asynchronous mode. If Receiver is enabled (RXEN=1), the data is sampled on SCLK (depending on CPOL and CPHA) without any oversampling. A setup and a hold time must be respected (which depends on the baud rate: 1/16 bit time).

*** Note:**

- 1. The SCLK pin works in conjunction with the UTXD pin, so the clock is provided only if TXEN=1 in [USARTn_CTRL](#) register, and a data is being transmitted (the data register USART_DR has been written). This means that it is not possible to receive a synchronous data without transmitting data.
- 2. The CPOL and CPHA bits in [USARTn_SYNCCTRL](#) register have to be selected when both the transmitter and the receiver are disabled (TXEN=0 and RXEN=0) to ensure that the clock pulses function correctly. These bits should not be changed while the transmitter or the receiver is enabled (TXEN=1 and RXEN=1).
- 3. The Synchronous mode supports master mode only, it can NOT receive or send data related to an input clock (SCLK is always an output).

| CPOL | CPHA | SCLK Idle Status | Diagrams |
|------|------|------------------|----------|
| 0 | 1 | Low | |
| 1 | 1 | High | |
| 0 | 0 | Low | |
| 1 | 0 | High | |

13.12 USART REGISTERS

Base Address: 0x4001 6000 (USART0)
0x4005 6000 (USART1)

13.12.1 USART n Receiver Buffer register (USARTn_RB) (n=0, 1)

Address Offset: 0x00

This register is the top byte of the USART RX FIFO, and contains the oldest character received and can be read via the bus interface. The LSB (bit 0) contains the first-received data bit. If the character received is less than 8 bits, the unused MSBs are padded with zeros.

The Divisor Latch Access Bit (DLAB) in the [USARTn_LC](#) register must be zero in order to access this register.

Since PE, FE and BI bits correspond to the byte on the top of the USART RX FIFO (i.e. the one that will be read in the next read from this register), the right approach for fetching the valid pair of received byte and its status bits is first to read the content of the [USARTn_LS](#) register, and then to read a byte from this register.

| Bit | Name | Description | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7:0 | RB[7:0] | Contains the oldest received byte in the USART RX FIFO. | R | 0 |

13.12.2 USART n Transmitter Holding register (USARTn_TH) (n=0, 1)

Address Offset: 0x00

This register is the top byte of the USART TX FIFO. The top byte is the newest character in the TX FIFO and can be written via the bus interface. The LSB represents the first bit to transmit.

The Divisor Latch Access Bit (DLAB) in [USARTn_LC](#) register must be zero in order to access this register.

| Bit | Name | Description | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7:0 | TH[7:0] | The byte will be sent when it is the oldest byte in TX FIFO and the transmitter is available. | W | 0 |

13.12.3 USART n Divisor Latch LSB registers (USARTn_DLL) (n =0, 1)

Address Offset: 0x00

The USART Divisor Latch is part of the USART Baud Rate Generator and holds the value used (optionally with the Fractional Divider) to divide the USARTn_PCLK clock in order to produce the baud rate clock, which must be the multiple of the desired baud rate that is specified by the Oversampling Register (typically 16X).

The USARTn_DLL and USARTn_DLM registers together form a 16-bit divisor, and DLAB bit in [USARTn_LC](#) register must be one in order to access these registers.

DLL contains the lower 8 bits of the divisor and DLM contains the higher 8 bits. A zero value is treated like 0x0001.

| Bit | Name | Description | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7:0 | DLL[7:0] | The USART Divisor Latch LSB Register, along with the DLM register, determines the baud rate of the USART. | R/W | 0 |

13.12.4 USART n Divisor Latch MSB register (USARTn_DLM) (n=0,1)

Address Offset: 0x04

| Bit | Name | Description | Attribute | Reset |
|-----|------|-------------|-----------|-------|
|-----|------|-------------|-----------|-------|

| | | | | |
|-------------|----------|---|-----|---|
| 31:8 | Reserved | | R | 0 |
| 7:0 | DLM[7:0] | The USART Divisor Latch MSB Register, along with the DLL register, determines the baud rate of the USART. | R/W | 0 |

13.12.5 USART n Interrupt Enable register (USARTn_IE) (n=0, 1)

Address Offset: 0x04

The DLAB bit in [USARTn_LC](#) register must be zero in order to access this register.

| Bit | Name | Description | Attribute | Reset |
|--------------|----------|--|-----------|-------|
| 31:11 | Reserved | | R | 0 |
| 10 | TXERRIE | TXERR interrupt enable bit The status of this interrupt can be read from TXERR bit in USARTn_LS register. 0: Disable 1: Enable | R/W | 0 |
| 9 | ABTOIE | Enables the auto-baud time-out interrupt enable bit. 0: Disable 1: Enable | R/W | 0 |
| 8 | ABEOIE | End of auto-baud interrupt enable bit. 0: Disable 1: Enable | R/W | 0 |
| 7:5 | Reserved | | R | 0 |
| 4 | TEMTIE | TEMT interrupt enable bit. The status of this interrupt can be read from TEMT bit in USARTn_LS register. 0: Disable 1: Enable | R/W | 0 |
| 3 | MSIE | Modem Status interrupt enable bit. The components of this interrupt can be read from USARTn_MS register. 0: Disable 1: Enable | R/W | 0 |
| 2 | RLSIE | Receive Line Status (RLS) interrupt enable bit. The status of this interrupt can be read from USARTn_LS [4:1]. 0: Disable 1: Enable | R/W | 0 |
| 1 | THREIE | THRE interrupt enable bit. The status of this interrupt can be read from THRE bit in USARTn_LS register. 0: Disable 1: Enable | R/W | 0 |
| 0 | RDAIE | RDA interrupt enable bit. Enables the Receive Data Available interrupt. It also controls the Character Receive Time-out interrupt. 0: Disable 1: Enable | R/W | 0 |

13.12.6 USART n Interrupt Identification register (USARTn_II) (n=0,1)

Address Offset: 0x08

This register provides a status code that denotes the priority and source of a pending interrupt. The interrupts are frozen during a USARTn_II register access. If an interrupt occurs during a USARTn_II register access, the interrupt is recorded for the next USARTn_II register access.

| Bit | Name | Description | Attribute | Reset |
|--------------|----------|----------------------|-----------|-------|
| 31:11 | Reserved | | R | 0 |
| 10 | TXERRIF | TXERR interrupt flag | R | 0 |

| | | | | |
|-----|------------|---|---|---|
| | | 0: TXERR has not occurred. 1: TXERR has occurred and interrupt is enabled. | | |
| 9 | ABTOIF | Auto-baud time-out interrupt flag. 0: Auto-baud has not timed-out 1: Auto-baud has timed out and interrupt is enabled. | R | 0 |
| 8 | ABEOIF | End of auto-baud interrupt flag 0: Auto-baud has not finished. 1: Auto-baud has finished successfully and interrupt is enabled. | R | 0 |
| 7:6 | FIFOEN | Equivalent to FIFOEN bit in USARTn_FIFOCTRL register. | R | 1 |
| 5:4 | Reserved | | R | 0 |
| 3:1 | INTID[2:0] | Interrupt identification which identifies an interrupt corresponding to the USARTn RX FIFO. 0x3: 1 - Receive Line Status (RLS). 0x2: 2a - Receive Data Available (RDA). 0x6: 2b - Character Time-out Indicator (CTI). 0x1: 3a - THRE Interrupt. 0x0: 4 - Modem status 0x7: 3b – TEMT Interrupt Other: Reserved | R | 0 |
| 0 | INTSTATUS | Interrupt status. The pending interrupt can be determined by evaluating USARTn_II[3:1]. 0: At least one interrupt is pending. 1: No interrupt is pending. | R | 1 |

Bits USARTn_II[9:8] are set by the auto-baud function and signal a time-out or end of auto-baud condition. The auto-baud interrupt conditions are cleared by setting the corresponding Clear bits in the Auto-baud Control Register.

Given the status of USARTn_II[3:0], an interrupt handler routine can determine the cause of the interrupt and how to clear the active interrupt. The USARTn_II register must be read in order to clear the interrupt prior to exiting the Interrupt service routine.

| Interrupt | USARTn_II [3:0] | Priority | Interrupt Source | Interrupt Reset |
|-----------|-----------------|-----------------|--|--|
| RLS | 0110 | Highest | Overrun error (OE), Parity error (PE), Framing error (FE) or Break interrupt (BI) | Read USARTn_LS register |
| RDA | 0100 | 2 nd | RX data in FIFO reached trigger level (FCR0=1) | Read USARTn_RB register or USART FIFO drops below trigger level |
| CTI | 1100 | 2 nd | Minimum of one character in the RX FIFO and no character input or removed during a time period depending on how many characters are in FIFO and what the trigger level is set at 3.5 to 4.5 character times. | Read USARTn_RB register |
| THRE | 0010 | 3 rd | THRE | Read USARTn_II register (if source of interrupt) or Write THR register |
| MS | 0000 | Lowest | CTS, DSR, RI, or DCD. | MSR Read |
| TEMT | 1110 | 3 rd | TEMT | Read USARTn_II register (if source of interrupt) or Write THR register |

13.12.7 USART n FIFO Control register (USARTn_FIFOCtrl) (n=0,1)

Address Offset: 0x08

This register controls the operation of the USART RX and TX FIFOs.

| Bit | Name | Description | Attribute | Reset |
|------|-----------|--|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7:6 | RXTL[1:0] | RX Trigger Level. These two bits determine how many receiver USART FIFO characters must be written before an interrupt is activated. 00: Trigger level 0 (1 character) 01: Trigger level 1 (4 characters) 10: Trigger level 2 (8 characters) 11: Trigger level 3 (14 characters) | W | 0 |
| 5:3 | Reserved | | R | 0 |
| 2 | TXFIFORST | TX FIFO Reset bit. 0: No impact on either of USART FIFOs. 1: Writing a logic 1 to reset the pointer logic in USART TX FIFO. HW shall clear this bit automatically. | W | 0 |
| 1 | RXFIFORST | RX FIFO Reset bit. 0: No impact on either of USART FIFOs. 1: Writing a logic 1 to reset the pointer logic in USART RX FIFO. HW shall clear this bit automatically. | W | 0 |
| 0 | FIFOEN | FIFO enable 0: No effect 1: Enable for both USART Rx and TX FIFOs and USARTn_FIFOCtrl [7:1] access. This bit must be set for proper USART operation. | W | 1 |

13.12.8 USART n Line Control register (USARTn_LC) (n=0,1)

Address Offset: 0x0C

This register determines the format of the data character that is to be transmitted or received.

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7 | DLAB | Divisor Latch Access bit 0: Disable access to Divisor Latches. 1: Enable access to Divisor Latches. | R/W | 0 |
| 6 | BC | Break Control bit 0: Disable break transmission. 1: Enable break transmission. Output pin USART TXD is forced to logic 0. | R/W | 0 |
| 5:4 | PS[1:0] | Parity Select bits 00: Odd parity. Number of 1s in the transmitted character and the attached parity bit will be odd. 01: Even Parity. Number of 1s in the transmitted character and the attached parity bit will be even. 10: Forced 1 stick parity. 11: Forced 0 stick parity. | R/W | 0 |
| 3 | PE | Parity Enable bit 0: Disable parity generation and checking. 1: Enable parity generation and checking. | R/W | 0 |
| 2 | SBS | Stop Bit Select bit 0: 1 stop bit. 1: 2 stop bits (1.5 if WLS bits=00). Must be 1 in Smart card mode. | R/W | 0 |
| 1:0 | WLS[1:0] | Word Length Select bits 00: 5-bit character length. 01: 6-bit character length. 10: 7-bit character length. 11: 8-bit character length. | R/W | 0 |

13.12.9 USART n Modem Control register (USARTn_MC) (n=0,1)

Address Offset: 0x10

This register enables the modem loopback mode and controls the modem output signals.

| Bit | Name | Description | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7 | CTSEN | CTS enable bit 0: Disable Auto-CTS flow control. 1: Enable Auto-CTS flow control. | R/W | 0 |
| 6 | RTSEN | RTS enable bit 0: Disable Auto-RTS flow control. 1: Enable Auto-RTS flow control. | R/W | 0 |
| 5 | Reserved | | R | 0 |
| 4 | LMS | Modem Loopback Mode Select bit. 0: Disable 1: Enable The modem loopback mode provides a mechanism to perform diagnostic loopback testing. Serial data from the transmitter is connected internally to serial input of the receiver. RXD has no effect on loopback and TXD is held in marking state. The <u>DSR</u> , <u>CTS</u> , <u>DCD</u> , and <u>RI</u> pins are ignored. Externally, <u>DTR</u> and <u>RTS</u> are set inactive. Internally, the bit 4 and bit 5 of <u>USARTn_MS</u> register are driven by bit 0 and bit 1 of <u>USARTn_MC</u> register. This permits modem status interrupts to be generated in Loopback mode by writing bit0 and bit 1 of USARTn_MC register. This bit provides a local loopback feature for diagnostic testing of the USART. When this bit is set to 1, the following occur: the transmitter TXD is set to the Marking (logic 1) state; the RXD is disconnected; the internal TXD and RXD pin are connected; the 4 MODEM Control inputs (<u>DSR</u> , <u>CTS</u> , <u>RI</u> , and <u>DCD</u>) are disconnected; and the 4 MODEM Control outputs (<u>DTR</u> , <u>RTS</u> , <u>OUT1</u> , and <u>OUT2</u>) are internally connected to the 4 MODEM Control inputs, and the MODEM Control output pins are forced to their inactive state (high). In the loopback mode, data that is transmitted is immediately received. This feature allows the processor to verify the transmitter and receiver data paths of the USART. In the loopback mode, the receiver and transmitter interrupts are fully operational. Their sources are external to the part. The MODEM Control Interrupts are also operational, but the interrupts' sources are now the lower four bits of the MODEM Control Register instead of the four MODEM Control inputs. The interrupts are still controlled by the Interrupt Enable Register. | R/W | 0 |
| 3 | OUT2 | This bit controls the OUT2 internal signal in loopback mode only, and OUT2 is internally connected to <u>DCD</u> . | R/W | 0 |
| 2 | OUT1 | This bit controls the OUT1 internal signal in loopback mode only, and OUT2 is internally connected to <u>RI</u> . | R/W | 0 |
| 1 | RTSCTRL | Source for modem output pin <u>RTS</u> . <u>RTS</u> pin is always forced to inactive state (high) in modem loopback mode. | R/W | 0 |
| 0 | DTRCTRL | Source for modem output pin <u>DTR</u> . <u>DTR</u> pin is always forced to inactive state (high) in modem loopback mode. | R/W | 0 |

13.12.10 USART n Line Status register (USARTn_LS) (n=0,1)

Address Offset: 0x14

*** Note:**

- 1. The break interrupt (BI) is associated with the character at the top of the USARTn_RB FIFO.
- 2. The framing error (FE) is associated with the character at the top of the USARTn_RB FIFO.
- 3. The parity error (PE) is associated with the character at the top of the USARTn_RB FIFO.

| Bit | Name | Description | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:9 | Reserved | | R | 0 |
| 8 | TXERR | TX Error flag. (Only available in smart card T=0 operation) 0: No TX Error. 1: Smart card has NACKed a transmitted character, one more than the number of times indicated by the TXRETRY field. | R | 0 |
| 7 | RXFE | Error in RX FIFO flag. RXFE =1 when a character with a RX error such as framing error, parity error, or break interrupt, is loaded into the USARTn_RB register. This bit is cleared when the USARTn_LS register is read and there are no subsequent errors in the USART FIFO. 0: USARTn_RB register contains no USART RX errors or FIFOEN=0 1: USARTn_RB register contains at least one USART RX error. | R | 0 |
| 6 | TEMT | Transmitter Empty flag TEMT=1 when both THR and TSR are empty; TEMT is cleared when either the TSR or the THR contain valid data. 0: THR and/or TSR contains valid data. 1: THR and TSR are empty. | R | 1 |
| 5 | THRE | Transmitter Holding Register Empty flag THRE indicates that the USART is ready to accept a new character for transmission. In addition, this bit causes the USART to issue THRE interrupt to if THREIE=1. THRE=1 when a character is transferred from the THR into the TSR. The bit is reset to logic 0 concurrently with the loading of the Transmitter Holding Register by the CPU. 0: THR contains valid data. 1: THR (TX FIFO) is empty. | R | 1 |
| 4 | BI | Break Interrupt flag. When RXD1 is held in the spacing state (all zeros) for one full character transmission (start, data, parity, stop), a break interrupt occurs. Once the break condition has been detected, the receiver goes idle until RXD1 goes to marking state (all ones). A USARTn_LS register read clears BI bit. The time of break detection is dependent on FIFOEN bit in USARTn_FIFOCTRL register. 0: Break interrupt status is inactive. 1: Break interrupt status is active. | R | 0 |
| 3 | FE | Framing Error flag. When the stop bit of a received character is a logic 0, a framing error occurs. A USARTn_LS register read clears FE bit. The time of the framing error detection is dependent on FIFOEN bit in USARTn_FIFOCTRL register. Upon detection of a framing error, the RX will attempt to re-synchronize to the data and assume that the bad stop bit is actually an early start bit. However, it cannot be assumed that the next received byte will be correct even if there is no Framing Error. 0: Framing error status is inactive. 1: Framing error status is active. | R | 0 |
| 2 | PE | Parity Error flag. When the parity bit of a received character is in the wrong state, a parity error occurs. A USARTn_LS register read clears PE bit. Time of parity error detection is dependent on FIFOEN bit in USARTn_FIFOCTRL register. 0: Parity error status is inactive. 1: Parity error status is active. | R | 0 |

| | | | | |
|---|-----|---|---|---|
| 1 | OE | <p>Overrun Error flag.</p> <p>The overrun error condition is set as soon as it occurs. A USARTn_LS register read clears OE bit. OE=1 when USART RSR has a new character assembled and the USARTn_RB FIFO is full. In this case, the USARTn_RB FIFO will not be overwritten and the character in the USARTn_RS register will be lost.</p> <p>0: Overrun error status is inactive. 1: Overrun error status is active.</p> | R | 0 |
| 0 | RDR | <p>Receiver Data Ready flag</p> <p>RDR=1 when the USARTn_RB FIFO holds an unread character and is cleared when the USARTn_RB FIFO is empty.</p> <p>0: USARTn_RB FIFO is empty. 1: USARTn_RB FIFO contains valid data.</p> | R | 0 |

13.12.11 USART n Modem Status register (USARTn_MS) (n=0,1)

Address Offset: 0x18

This register is a read-only register that provides status information on USART input signals.

| | | | | |
|---|--|--|--|--|
| <p>* Note:</p> <ul style="list-style-type: none"> ➤ 1. Whenever the DCD bit changes state, an interrupt is generated if the MODEM Status Interrupt is enabled. ➤ 2. Whenever the RI bit changes from a high to a low state, an interrupt is generated if the MODEM Status Interrupt is enabled.. ➤ 3. Whenever the DSR bit changes state, an interrupt is generated if the MODEM Status Interrupt is enabled. ➤ 4. Whenever the CTS bit changes state, an interrupt is generated if the MODEM Status Interrupt is enabled. | | | | |
|---|--|--|--|--|

| Bit | Name | Description | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7 | DCD | Data Carrier Detect State. Complement of input <u>DCD</u> . This bit is connected to USARTn_MC[3] in modem loopback mode. | R | 0 |
| 6 | RI | Ring Indicator State. Complement of input <u>RI</u> . This bit is connected to USARTn_MC[2] in modem loopback mode. | R | 1 |
| 5 | DSR | Data Set Ready State. Complement of input signal <u>DSR</u> . This bit is connected to USARTn_MC[0] in modem loopback mode. | R | 1 |
| 4 | CTS | Clear To Send State. Complement of input signal <u>CTS</u> . This bit is connected to USARTn_MC[1] in modem loopback mode. | R | 0 |
| 3 | DDCD | Delta DCD. Set upon state change of input <u>DCD</u> . Cleared after reading this register. 0: No change detected on modem input <u>DCD</u> . 1: State change detected on modem input <u>DCD</u> . | R | 0 |
| 2 | TERI | Trailing Edge RI. Set upon low to high transition of input <u>RI</u> . Cleared after reading this register. 0: No change detected on modem input <u>RI</u> . 1: Low-to-high transition detected on <u>RI</u> . | R | 0 |
| 1 | DDSR | Delta DSR. Set upon state change of input <u>DSR</u> . Cleared after reading this register. 0: No change detected on modem input <u>DSR</u> . 1: State change detected on modem input <u>DSR</u> . | R | 0 |
| 0 | DCTS | Delta CTS. Set upon state change of input <u>CTS</u> . Cleared after reading this register. 0: No change detected on modem input <u>CTS</u> . 1: State change detected on modem input <u>CTS</u> . | R | 0 |

13.12.12 USART n Scratch Pad register (USARTn_SP) (n=0, 1)

Address Offset: 0x1C

This register has no effect on the USART operation. This register can be written and/or read at user's discretion. There is no provision in the interrupt interface that would indicate to the host that a read or write of this register has occurred.

| Bit | Name | Description | Attribute | Reset |
|------|----------|----------------------------|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7:0 | PAD[7:0] | A readable, writable byte. | R/W | 0 |

13.12.13 USART n Auto-baud Control register (USARTn_ABCTRL) (n=0, 1)

Address Offset: 0x20

This register controls the process of measuring the incoming clock/data rate for the baud rate generation and can be read and written at user's discretion. Besides, it also controls the clock pre-scaler for the baud rate generation. The reset value of the register keeps the fractional capabilities of USART disabled making sure that USART is fully SW and HW compatible with USARTs not equipped with this feature.

| Bit | Name | Description | Attribute | Reset |
|-------|-------------|---|-----------|-------|
| 31:10 | Reserved | | R | 0 |
| 9 | ABTOIFC | Auto-baud time-out interrupt flag clear bit 0: No effect 1: Clear ABTOIF bit. This bit is automatically cleared by HW. | W | 0 |
| 8 | ABEOIFC | End of auto-baud interrupt flag clear bit 0: No effect. 1: Clear ABEOIF bit. This bit is automatically cleared by HW. | W | 0 |
| 7:3 | Reserved | | R | 0 |
| 2 | AUTORESTART | Restart mode 0: No restart 1: Restart in case of timeout (counter restarts at next USART RX falling edge) | R/W | 0 |
| 1 | MODE | Auto-baud mode select bit. 0: Mode 0. 1: Mode 1. | R/W | 0 |
| 0 | START | This bit is automatically cleared after auto-baud completion. 0: Auto-baud stop (auto-baud is not running). 1: Auto-baud start (auto-baud is running). Auto-baud run bit. This bit is automatically cleared by HW after auto-baud completion. | R/W | 0 |

13.12.14 USART n IrDA Control register (USARTn_IRDACTRL) (n=0, 1)

Address Offset: 0x24

This register enables and configures the IrDA mode. The value of this register should not be changed while transmitting or receiving data, or data loss or corruption may occur.

| Bit | Name | Description | Attribute | Reset |
|------|---------------|--|-----------|-------|
| 31:6 | Reserved | | R | 0 |
| 5:3 | PULSEDIV[2:0] | Configures the pulse width when FIXPULSEEN = 1. 000: 2 x TPCLK 001: 4 x TPCLK 010: 8 x TPCLK 011: 16 x TPCLK 100: 32 x TPCLK 101: 64 x TPCLK 110: 128 x TPCLK 111: 256 x TPCLK | R/W | 0 |
| 2 | FIXPULSEEN | IrDA fixed pulse width mode enable. | R/W | 0 |

| | | | | |
|----------|----------|--|-----|---|
| | | 0: Disable. Pulse width = 3 / (Oversampling x baud rate) 1: Enable. Pulse width is set by PULSEDIV bits. | | |
| 1 | IRDAINV | Serial input inverter 0: The serial input is not inverted. 1: The serial input is inverted. This has no effect on the serial output. | R/W | 0 |
| 0 | Reserved | | R | 0 |

13.12.15 USART n Fractional Divider register (USARTn_FD) (n=0, 1)

Address Offset: 0x28

This register controls the clock prescaler for the baud rate generation and can be read and written at the user's discretion. This prescaler takes the APB clock and generates an output clock according to the specified fractional requirements.

In most applications, the USART samples received data 16 times in each nominal bit time, and sends bits that are 16 input clocks wide. OVER8 bit allows software to control the ratio between the input clock and bit clock. This is required for smart card mode, and provides an alternative to fractional division for other modes.

*** Note: If the fractional divider is active (DIVADDVAL>0) and USARTn_DLM=0, the value of the USARTn_DLL register must ≥ 3.**

| Bit | Name | Description | Attribute | Reset |
|-------------|----------------|---|-----------|-------|
| 31:9 | Reserved | | R | 0 |
| 8 | OVER8 | Oversampling value 0: Oversampling by 16 1: Oversampling by 8 (Not supported for IrDA mode) | R/W | 0 |
| 7:4 | MULVAL[3:0] | Baud rate pre-scaler multiplier value = MULVAL[3:0] +1 0000: Baud rate pre-scaler multiplier value is 1 for HW 0001: Baud rate pre-scaler multiplier value is 2 for HW 1111: Baud rate pre-scaler multiplier value is 16 for HW. | R/W | 0 |
| 3:0 | DIVADDVAL[3:0] | Baud rate generation pre-scaler divisor value. If this field is 0, fractional baud rate generator will not impact the USART baud rate | R/W | 0 |

13.12.16 USART n Control register (USARTn_CTRL) (n=0, 1)

Address Offset: 0x30

In addition to HW flow control (Auto-CTS and Auto-RTS mechanisms), this register enables implementation of SW flow control.

When TXEN = 1, the USART transmitter will keep sending data as long as they are available. As soon as TXEN bit becomes 0, USART transmission will stop.

It is strongly suggested to let the USART HW implemented auto flow control features take care of limit the scope of TXEN to SW flow control.

*** Note: It is advised that TXEN and RXEN are set in the same instruction if needed in order to minimize the setup and the hold time of the receiver.**

| Bit | Name | Description | Attribute | Reset |
|-------------|----------|---|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7 | TXEN | When this bit is 1, data written to the USARTn_TH register is output on | R/W | 1 |

| | | | | |
|-----|-----------|--|-----|---|
| | | the TXD pin as soon as any preceding data has been sent. If this bit is cleared to 0 while a character is being sent, the transmission of that character is completed, but no further characters are sent until this bit is set again. In other words, a 0 in this bit blocks the transfer of characters from the USARTn_TH register or TX FIFO into the transmit shift register. SW can clear this bit when it detects that the HW-handshaking TX-permit signal (<u>CTS</u>) has gone false, or with SW handshaking, when it receives an XOFF character (DC3). SW can set this bit again when it detects that the TX-permit signal has gone true, or when it receives an XON (DC1) character. | | |
| 6 | RXEN | 0: Disable RX related function 1: Enable RX | R/W | 1 |
| 5:4 | Reserved | | R | 0 |
| 3:1 | MODE[2:0] | USARTn Mode 000: UART mode. HW will switch GPIO to UTXDn and URXDn. 001: Modem control mode. HW will switch GPIO to UTXDn, URXDn, <u>UDSRn</u> , <u>UCTSn</u> , <u>UDCDn</u> , <u>URIn</u> , <u>UDTRn</u> and <u>URTSn</u> . 010: IRDA mode. HW switches GPIO to UTXDn and URXDn . 011: Smart Card mode. HW will switch GPIO to UTXDn, and enable UTXDn pin with open-drain. 100: Synchronous mode. HW will switch GPIO to UTXDn, URXDn , and USCLK pin. 101:RS-485 mode. HW will switch GPIO to UTXDn, URXDn pin. | R/W | 0 |
| 0 | USARTEN | USART enable 0: Disable . All USART shared pins act as GPIO. 1: Enable. HW switches GPIO to USART pin according to MODE bits automatically. | R/W | 0 |

13.12.17 USART n Half-duplex Enable register (USARTn_HDEN) (n=0, 1)

Address Offset: 0x34

After reset the USART will be in full-duplex mode, meaning that both TX and RX work independently. After setting the HDEN bit, the USART will be in half-duplex mode. In this mode, the USART ensures that the receiver is locked when idle, or will enter a locked state after having received a complete ongoing character reception. Line conflicts must be handled in SW.

The behavior of the USART is unpredictable when data is presented for reception while data is being transmitted. For this reason, the value of the HDEN register should not be modified while sending or receiving data, or data may be lost or corrupted.

*** Note: This register should be disabled when in smart card mode or IrDA mode (Smart card and IrDA by default run in half-duplex mode).**

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:1 | Reserved | | R | 0 |
| 0 | HDEN | Half-duplex mode enable bit 0: Disable 1: Enable | R/W | 0 |

13.12.18 USART n Smart card Interface Control register (USARTn_SCICTRL) (n=0, 1)

Address Offset: 0x38

| Bit | Name | Description | Attribute | Reset |
|-------|----------|-------------|-----------|-------|
| 31:24 | Reserved | | R | 0 |

| | | | | |
|--------------|--------------|--|-----|-----|
| 23:16 | TC[7:0] | Count for SCLK clock cycle when SCLKEN=1. SCLK will toggle every (TC[7:0]+1) * USARTn_PCLK cycle | R/W | 0x0 |
| 15:8 | XTRAGUARD | When the protocol selection T= 0, this field indicates the number of bit times (ETUs) by which the guard time after a character transmitted by the USART should exceed the nominal 2 bit times. 0xFF in this field may indicate that there is just a single bit after a character and 11 bit times/character | R/W | N/A |
| 7:5 | TXRETRY[2:0] | When the protocol selection T = 0, the field controls the maximum number of retransmissions that the USART will attempt if the remote device signals NACK. When NACK has occurred this number of times plus one, the TX Error (TXERR) bit in USARTn_LS register is set, an interrupt is requested if enabled, and the USART is locked until the FIFO is cleared. | R/W | N/A |
| 4 | Reserved | | R | 0 |
| 3 | SCLKEN | SCLK enable Enable if the smart card to be communicated with requires a clock. 0: Disable 1: Enable. HW will switch GPIO to UnSCLK pin. | R/W | 0 |
| 2 | PROTSEL | Protocol selection as defined in the ISO7816-3 standard. 0: T = 0 1: T = 1 | R/W | 0 |
| 1 | NACKDIS | NACK response disable bit. Only applicable in T=0. 0: A NACK response is enabled. 1: A NACK response is inhibited. | R/W | 0 |
| 0 | Reserved | | R | 0 |

13.12.19 USART n RS485 Control register (USARTn_RS485CTRL) (n=0, 1)

Address Offset: 0x3C

| Bit | Name | Description | Attribute | Reset |
|-------------|----------|--|-----------|-------|
| 31:6 | Reserved | | R | 0 |
| 5 | OINV | Polarity control. This bit reverses the polarity of the direction control signal on the RTS pin. 0: The direction control pin will be driven to logic 0 when the transmitter has data to be sent. It will be driven to logic 1 after the last bit of data has been transmitted. 1: The direction control pin will be driven to logic 1 when the transmitter has data to be sent. It will be driven to logic 0 after the last bit of data has been transmitted. | R/W | 0 |
| 4 | ADCEN | Auto Direction control enable bit. 0: Disable 1: Enable. <u>RTS</u> pin is used for direction control. HW will switch GPIO to <u>URTSn</u> pin automatically. | R/W | 0 |
| 3 | Reserved | | R | 0 |
| 2 | AADEN | Auto Address Detect (AAD) enable bit. 0: Disable 1: Enable | R/W | 0 |
| 1 | RXEN | RS-485/EIA-485 Receiver enable bit. (Only work when NMMEN = 1) 0: Disable 1: Enable | R/W | 0 |
| 0 | NMMEN | RS-485/EIA-485 Normal Multidrop Mode (NMM) enable bit. 0: Disable 1: Enable. In this mode, an address is detected when a received byte causes the USART to set the parity error and generate an interrupt. | R/W | 0 |

13.12.20 USART n RS485 Address Match register (USARTn_RS485ADRMATCH) (n=0, 1)

Address Offset: 0x40

This register contains the address match value for RS-485/EIA-485 mode.

| Bit | Name | Description | Attribute | Reset |
|------|------------|----------------------------------|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7:0 | MATCH[7:0] | The address value to be matched. | R/W | 0 |

13.12.21 USART n RS485 Delay Value register (USARTn_RS485DLYV) (n=0, 1)

Address Offset: 0x44

The user may program this register with a delay between the last stop bit leaving the TXFIFO and the de-assertion of RTS. This delay time is in periods of the baud clock. Any delay time from 0 to 255 bit times may be programmed.

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7:0 | DLY[7:0] | The direction control (RTS) delay value. This register works in conjunction with an 8-bit counter. | R/W | 0 |

13.12.22 USART n Synchronous Mode Control Register (USARTn_SYNCCTRL) (n=0,1)

Address Offset: 0x48

This register controls the synchronous mode. When this mode is in effect, the USART generates or receives a bit clock on the SCLK pin and applies it to transmit and receive shift registers. Synchronous mode should not be used with smart card mode.

| Bit | Name | Description | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:3 | Reserved | | R | 0 |
| 2 | CPHA | Clock phase for edge sampling. 0: Sample on the rising edge of SCLK 1: Sample on the falling edge of SCLK | R/W | 0 |
| 1 | CPOL | Clock polarity selection bit 0: SCLK idles at Low level. 1: SCLK idles at High level. | R/W | 0 |
| 0 | Reserved | | R | 0 |

14 I2S

14.1 OVERVIEW

The I2S bus specification defines a 3-wire serial bus, having one data, one clock, and one word select signal. The basic I2S connection has one master, which is always the master, and one slave.

14.2 FEATURES

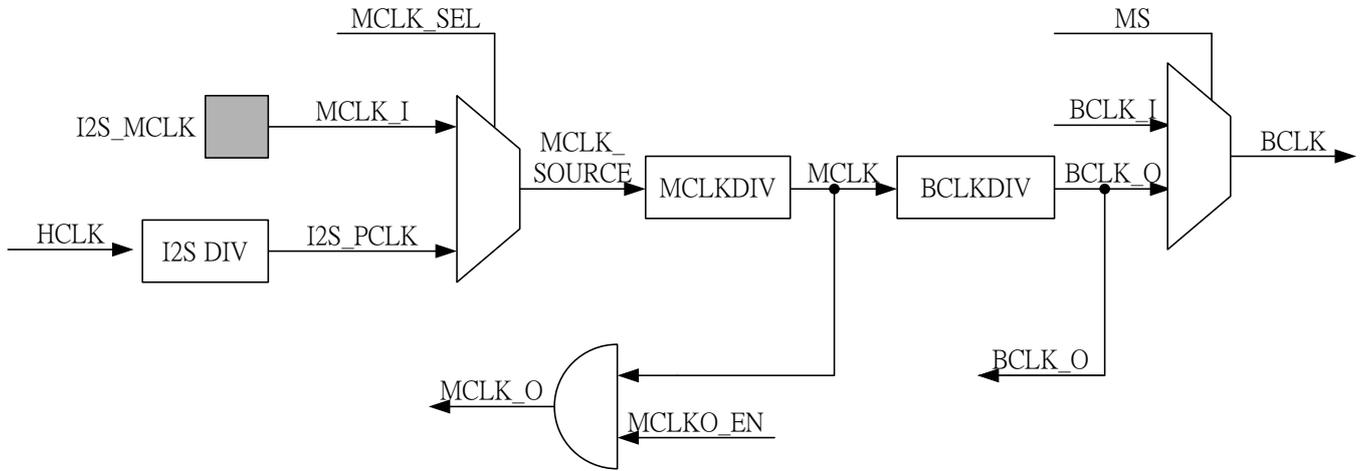
- I2S can operate as either master or slave.
- Capable of handling 8/16/24/32-bit data length.
- Mono and stereo audio data supported.
- I2S and MSB justified data format supported.
- 8 word (32-bit) FIFO data buffers are provided.
- Generate interrupt requests when buffer levels cross a programmable boundary.
- Controls include reset, stop and mute options separately for I2S input and I2S output.

14.3 PIN DESCRIPTION

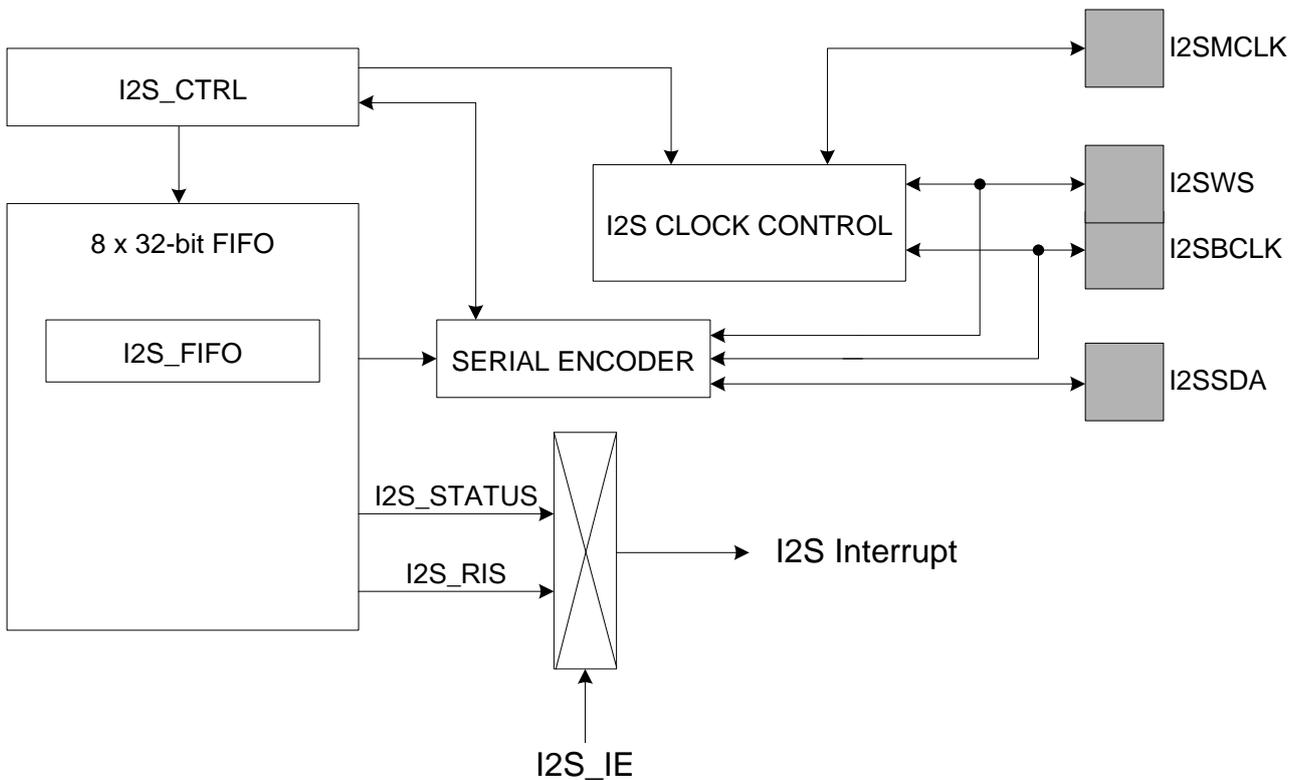
| Pin Name | Type | Description | GPIO Configuration |
|----------|------|----------------------------------|-----------------------|
| I2SBCLK | O | I2S Bit clock (Master) | |
| | I | I2S Bit clock (Slave) | Depends on GPIOOn_CFG |
| I2SWS | O | I2S Word Select (Master) | |
| | I | I2S Word Select (Slave) | Depends on GPIOOn_CFG |
| I2SSDA | O | I2S Transmitted Serial data | |
| | I | I2S Received Serial data | Depends on GPIOOn_CFG |
| I2SMCLK | O | I2S Master clock output | |
| | I | I2S Master clock input from GPIO | Depends on GPIOOn_CFG |

14.4 BLOCK DIAGRAM

14.4.1 I2S CLCOK CONTROL



14.4.2 I2S BLOCK DIAGRAM

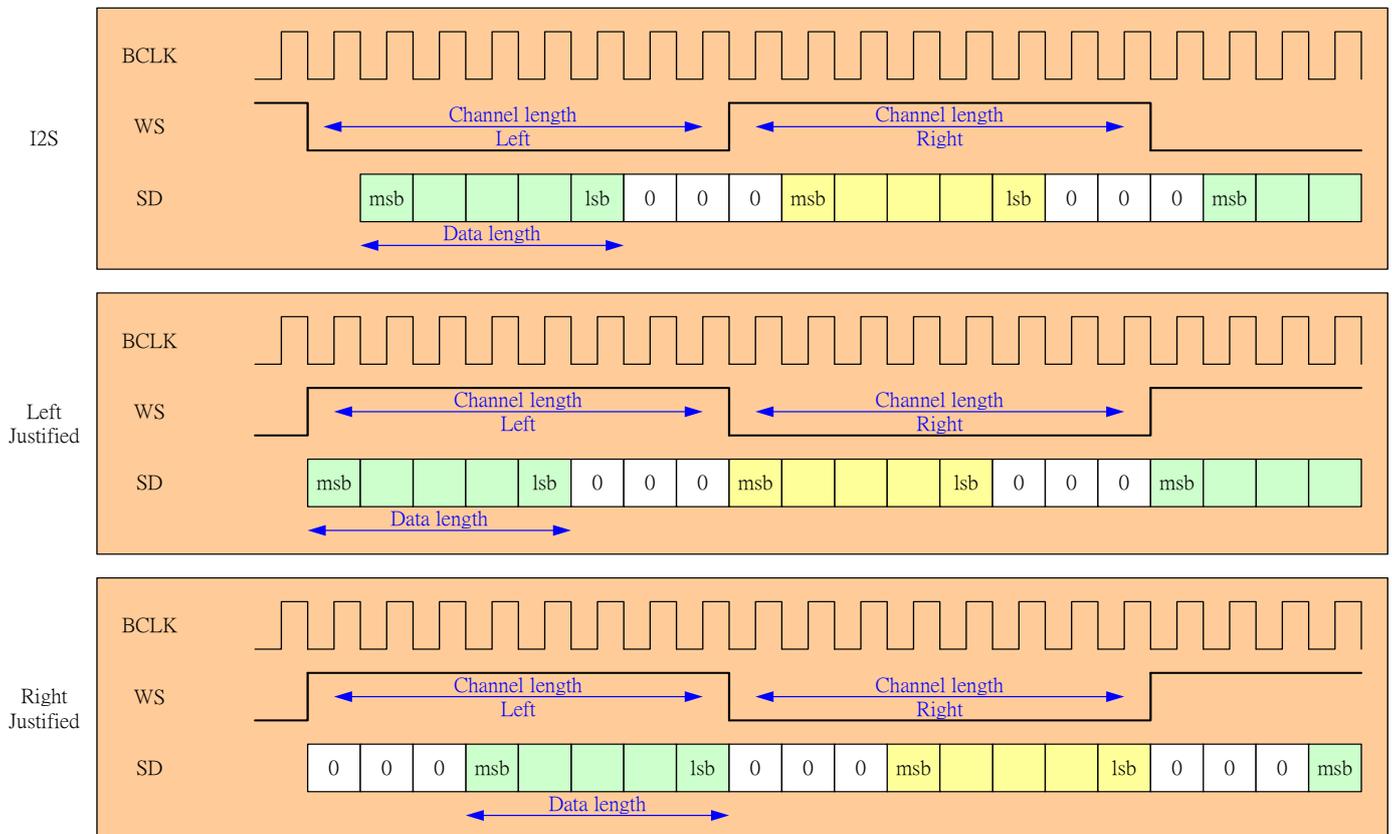


14.5 FUNCTIONAL DESCRIPTION

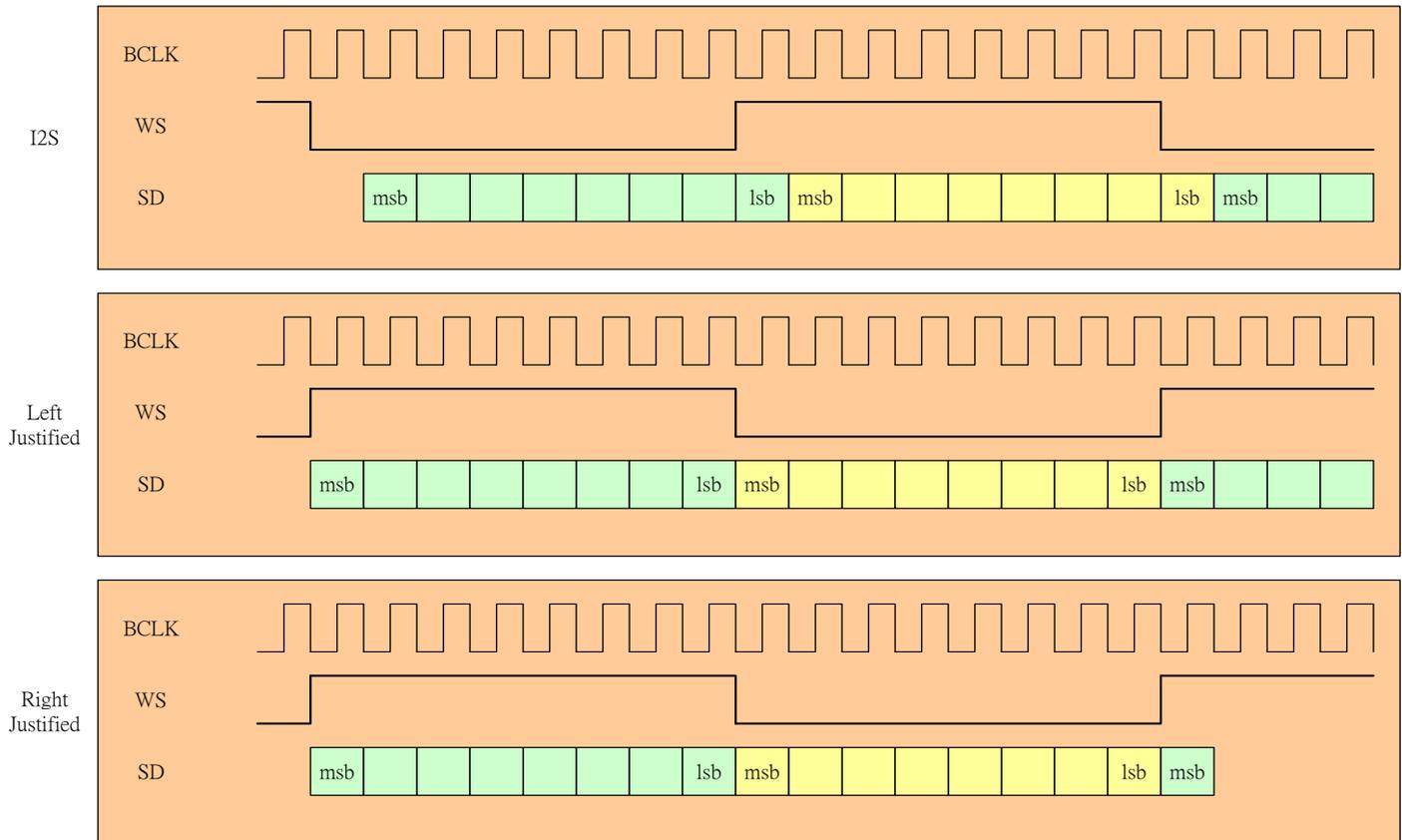
14.5.1 I2S OPERATION

- Standard I2S
- Right-justified Data Format
- MSB (Left)-justified Data Format

Channel Length > Data Length:



Channel Length = Data Length



14.5.2 I2S FIFO OPERAION

14.5.2.1 MONO

8bit

| | | | |
|-----|-----|-----|-----|
| N+3 | N+2 | N+1 | N |
| N+7 | N+6 | N+5 | N+4 |

16bit

| | |
|-----|-----|
| N+1 | N |
| N+3 | N+2 |

24 bit

| | |
|--|-----|
| | N |
| | N+1 |

32 bit

| |
|-----|
| N |
| N+1 |

14.5.2.2 STEREO

8bit

| | | | |
|----------|---------|----------|---------|
| RIGHT +1 | LEFT +1 | RIGHT | LEFT |
| RIGHT +3 | LEFT +3 | RIGHT +2 | LEFT +2 |

16bit

| | |
|----------|--------|
| RIGHT | LEFT |
| RIGHT +1 | LEFT+1 |

24 bit

| | |
|--|-------|
| | LEFT |
| | RIGHT |

32 bit

| |
|-------|
| LEFT |
| RIGHT |

14.6 I2S REGISTERS

Base Address: 0x4001 A000

14.6.1 I2S Control register (I2S_CTRL)

Address Offset: 0x00

*** Note: START bit shall be set at last.**

| Bit | Name | Description | Attribute | Reset |
|-------|---------------|--|-----------|-------|
| 31:21 | Reserved | | R | 0 |
| 20:16 | CHLENGTH[4:0] | Bit number of single channel = CHLENGTH[4:0]+1. 0~6: Reserved 7: 8 bits 8: 9 bits 31: 32bits (Max) | R/W | 0x1F |
| 15 | I2SEN | I2S enable bit 0: Disable 1: Enable I2S. HW will switch GPIO to MCLK, BCLK, and WS. | R/W | 0 |
| 14:12 | FIFOTH[2:0] | FIFO Threshold level 0: FIFO threshold level = 0 1: FIFO threshold level = 1 n: FIFO threshold level = n | R/W | 0x3 |
| 11:10 | DL[1:0] | Data Length 00: 8 bits 01: 16 bits 10: 24 bits 11: 32 bits | R/W | 0x1 |
| 9:8 | Reserved | | R | 0 |
| 7 | CLRFIFO | Clear I2S FIFO 0: No effect. 1: Reset FIFO (FIFOLV bit becomes 0, FIFOEMPTY bit becomes 1, Data in FIFO will be cleared). This bit returns "0" automatically | W | 0 |
| 6:5 | FORMAT[1:0] | I2S operation format. 00: Standard I2S format 01: Left-justified format 10: Right(MSB)-justified format 11: Reserved | R/W | 0 |
| 4 | MS | Master/Slave selection bit 0: Act as Master using internally generated BCLK and WS signals. 1: Act as Slave using externally BCLK and WS signals. | R/W | 0 |
| 3 | TRS | Transmit/Receiver selection bit 0: Transmitter 1: Receiver | R/W | 0 |
| 2 | MONO | Mono/Stereo selection bit 0: Stereo 1: Mono | R/W | 0 |
| 1 | MUTE | Mute enable bit 0: Disable Mute 1: Enable. I2SSDA Output = 0 | R/W | 0 |
| 0 | START | Start Transmit/Receive bit. 0: Stop Transmit/Receive 1: Start Transmit/Receive | R/W | 0 |

14.6.2 I2S Clock register (I2S_CLK)

Address Offset: 0x04

| Bit | Name | Description | Attribute | Reset |
|-------|--------------|---|-----------|-------|
| 31:16 | Reserved | | R | 0 |
| 15:8 | BCLKDIV[7:0] | BCLK divider 0: BCLK = MCLK / 2 1: BCLK = MCLK / 4 2: BCLK = MCLK / 6 3: BCLK = MCLK / 8 n: BCLK = MCLK / (2*n + 2) | R/W | 1 |
| 7:5 | Reserved | | R | 0 |
| 4 | MCLKSEL | MCLK source selection bit 0: MCLK source of master is from I2S_PCLK 1: MCLK source of master is from GPIO | R/W | 0 |
| 3 | MCLKOEN | MCLK output enable bit 0: Disable 1: Enable | R/W | 0 |
| 2:0 | MCLKDIV[2:0] | MCLK divider 0: MCLK = MCLK source 1: MCLK = MCLK source / 2 2: MCLK = MCLK source / 4 n: MCLK = MCLK source / (2*n), n>0 | R/W | 0 |

14.6.3 I2S Status register (I2S_STATUS)

Address Offset: 0x08

| Bit | Name | Description | Attribute | Reset |
|-------|-------------|---|-----------|-------|
| 31:16 | Reserved | | R | 0 |
| 15:12 | FIFOLV[3:0] | FIFO used level 0000: 0/8 FIFO is used (Empty) 0001: 1/8 FIFO is used 0010: 2/8 FIFO is used 1000: 8/8 FIFO is used (Full) Other: Reserved | R | 0 |
| 11 | FIFOEMPTY | FIFO empty flag 0: FIFO is not empty. 1: FIFO is empty. Data read from FIFO will be zero. | R | 0 |
| 10 | FIFOFULL | FIFO full flag 0: FIFO is not full. 1: FIFO is full. Write operation to FIFO will be ignored. | R | 0 |
| 9:7 | Reserved | | R | 0 |
| 6 | FIFOTHF | FIFO threshold flag 0: FIFOLV \geq FIFOTH if act as Transmitter; FIFOLV \leq FIFOTH if act as Receiver 1: FIFOLV < FIFOTH if act as Transmitter; FIFOLV > FIFOTH if act as Receiver | R | 0 |
| 5:2 | Reserved | | R | 0 |
| 1 | RIGHTCH | Current channel status 0: Current channel is Left channel 1: Current channel is Right channel | R | 1 |
| 0 | I2SINT | I2S interrupt flag 0: No I2S interrupt 1: I2S interrupt occurs. | R | 0 |

14.6.4 I2S Interrupt Enable register (I2S_IE)

Address Offset: 0x0C

| Bit | Name | Description | Attribute | Reset |
|------|------------|--|-----------|-------|
| 31:7 | Reserved | | R | 0 |
| 6 | FIFOTHIE | FIFO threshold interrupt enable bit 0: Disable 1: Enable | R/W | 0 |
| 5 | FIFOOVFIEN | FIFO overflow interrupt enable bit 0: Disable 1: Enable | R/W | 0 |
| 4 | FIFOUDFIEN | FIFO underflow interrupt enable bit 0: Disable 1: Enable | R/W | 0 |
| 3:0 | Reserved | | R | 0 |

14.6.5 I2S Raw Interrupt Status register (I2S_RIS)

Address Offset: 0x10

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:7 | Reserved | | R | 0 |
| 6 | FIFOTHIF | FIFO threshold interrupt flag 0: No FIFO threshold interrupt 1: FIFO threshold triggered. | R | 0 |
| 5 | FIFOOVIF | FIFO overflow interrupt flag 0: No FIFO overflow 1: FIFO overflow (FIFO is full and still being written). | R | 0 |
| 4 | FIFOUDIF | FIFO underflow interrupt flag 0: No FIFO underflow 1: FIFO underflow (FIFO is empty and still being read). | R | 0 |
| 3:0 | Reserved | | R | 0 |

14.6.6 I2S Interrupt Clear register (I2S_IC)

Address Offset: 0x14

| Bit | Name | Description | Attribute | Reset |
|------|----------|---------------------------------------|-----------|-------|
| 31:7 | Reserved | | R | 0 |
| 6 | FIFOTHIC | 0: No effect 1: Clear FIFOTHIF bit | W | 0 |
| 5 | FIFOOVIC | 0: No effect 1: Clear FIFOOVIF bit | W | 0 |
| 4 | FIFOUDIC | 0: No effect 1: Clear FIFOUDIF bit | W | 0 |
| 3:0 | Reserved | | R | 0 |

14.6.7 I2S FIFO register (I2S_FIFO)

Address Offset: 0x18

| Bit | Name | Description | Attribute | Reset |
|------|------------|--|-----------|-------|
| 31:0 | FIFO[31:0] | 8 x 32-bit FIFO Write Only if act as Transmitter Read Only if act as Receiver. | R/W | 0 |

15 FLASH

15.1 OVERVIEW

SONiX 32-bit MCU integrated device feature in-system programmable (ISP) FLASH memory for convenient, upgradeable code storage. The FLASH memory may be programmed via the SONiX 32-bit MCU programming interface or by application code for maximum flexibility. SONiX 32-bit MCU provides security options at the disposal of the designer to prevent unauthorized access to information stored in FLASH memory.

- The MCU is stalled during Flash program and erase operations, although peripherals (Timers, WDT, I/O, PWM, etc.) remain active.
- Watchdog timer should be cleared if enabled before the Flash write or erase operation.
- The erase operation sets all the bits in the Flash page to logic 1.
- HW will hold system clock and automatically move out data from RAM and do programming, after programming finished, HW will release system clock and let MCU execute the next instruction.

*** Note: HCLK MUST be equal or less than 24MHz during Flash program and erase operations.**

15.2 EMBEDDED FLASH MEMORY

The Flash memory is organized as 32-bit wide memory cells that can be used for storing both code and data constants, and is located at a specific base address in the memory map of chip.

The high-performance Flash memory module in chip has the following key features:

- Memory organization: the Flash memory is organized as a User ROM, Boot ROM.

| | |
|----------|---|
| User ROM | 8K × 32 bits divided into 64 pages of 512 Bytes |
| Boot ROM | 1K × 32 bits divided into 8 pages of 512 Bytes |

The Flash interface implements instruction access and data access based on the AHB protocol. It implements the logic necessary to carry out Flash memory operations (Program/Erase). Program/Erase operations can be performed over the whole product voltage range.

15.3 FEATURES

- Read interface (32-bit)
- Flash Program / Erase operation
- Code Option includes Code Security (CS)

Write operations to the main memory block and the code options are managed by an embedded Flash Memory Controller (FMC). The high voltage needed for Program/Erase operations is internally generated. The main Flash memory can be read/write protected against different levels of Code Security (CS).

During a write operation to the Flash memory, any attempt to read the Flash memory will stall the bus. The read operation will proceed correctly once the write operation has completed. This means that code or data fetches cannot be made while a write/erase operation is ongoing.

For write and erase operations on the Flash memory, the IHRC will be turn ON by FMC. The Flash memory can be programmed and erased using ICP and ISP.

15.4 ORGANIZATION

| Block | Name | Base Address | Size (Byte) |
|-------------|-------------------------|-------------------------|-------------|
| User ROM | Page 0 | 0x00000000 ~ 0x000001FF | 512 |
| | Page 1 | 0x00000200 ~ 0x000003FF | 512 |
| | . | . | |
| | . | . | |
| | . | . | |
| | Page 15 | 0x00001E00 ~ 0x00001FFF | 512 |
| | . | . | |
| | . | . | |
| | . | . | |
| Boot Loader | Page 31 | 0x00003E00 ~ 0x00003FFF | 512 |
| | . | . | |
| | . | . | |
| | . | . | |
| | Page 63 | 0x00007E00 ~ 0x00007FFF | 512 |
| | Page 0 | 0x1FFF0000 ~ 0x1FFF01FF | 512 |
| | Page 1 | 0x1FFF0200 ~ 0x1FFF03FF | 512 |
| | . | . | |
| | . | . | |
| . | . | | |
| Page 7 | 0x1FFF0E00 ~ 0x1FFF0FFF | 512 | |

15.5 READ

The embedded Flash module can be addressed directly, as a common memory space. Any data read operation accesses the content of the Flash module through dedicated read senses and provides the requested data.

The read interface consists of a read controller on one side to access the Flash memory, and an AHB interface on the other side to interface with the CPU. The main task of the read interface is to generate the control signals to read from the Flash memory as required by the CPU.

15.6 PROGRAM/ERASE

The Flash memory erase operation can be performed at page level.

To ensure that there is no over-programming, the Flash programming and erase controller blocks are clocked by IHRC.

15.7 EMBEDDED BOOT LOADER

The embedded boot loader is used to reprogram the Flash memory using the USART0 serial interface. This program is located in the Boot ROM and is programmed by SONiX during production.

15.8 FLASH MEMORY CONTROLLER (FMC)

The FMC handles the program and erase operations of the Flash memory.

15.8.1 CODE SECURITY (CS)

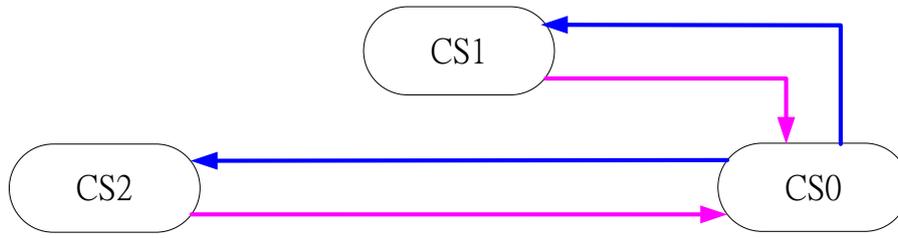
Code Security is a mechanism that allows the user to enable different levels of security in the system so that access to the on-chip Flash and use of the ISP can be restricted.

* **Note: Any Code Security change becomes effective only after the MCU has been Reboot.**

| User ROM | | CS0 | CS1 | CS2 | |
|----------------------------------|---------|-----|------|------|---|
| WRITER | Read | O | X | X | |
| | Erase | O | O(*) | O(*) | (*) WRITER will change the CS level to CS0. |
| | Program | O | O | O | |
| FW (EEPROM emulation) | Read | O | O | O | |
| | Erase | O | O | X | |
| | Program | O | O | X | |
| SWD | Read | O | X | X | |
| | Erase | O | X | X | |
| | Program | O | X | X | |

* **Note: User may try to change security level from CS2 to CS0, or from CS1 to CS0. HW shall:**

1. Mass erase the User ROM first. User shall NOT execute this operation in debug mode, since the SWD communication may fail during the mass erase procedure.
2. Update security level.



— includes:
- New option byte programming

— includes:
- Option byte erase
- Mass Erase

15.8.2 PROGRAM FLASH MEMORY

The Flash memory can be programmed 32 bits at a time. CPU can program the main Flash memory by performing standard word write operations. The PG bit in the FLASH_CTRL register must be set. FMC preliminarily reads the value at the addressed main Flash memory location and checks that it has been erased. If not, the program operation is skipped and a warning is issued by the PGERR bit in FLASH_STATUS register. The end of the program operation is indicated by the EOP bit in the FLASH_STATUS register.

The main Flash memory programming sequence in standard mode is as follows:

- ◆ Set the PG bit in the FLASH_CTRL register.
- ◆ Perform the data write at the desired address.
- ◆ Wait for the BUSY bit to be reset.
- ◆ Read the programmed value and verify.

15.8.3 ERASE

The Flash memory can be erased page by page or completely (Mass Erase).

15.8.3.1 PAGE ERASE

A page of the Flash memory can be erased using the Page Erase feature of the FMC. To erase a page, the procedure below should be followed:

- ◆ Set the PER bit in the FLASH_CTRL register
- ◆ Program the FLASH_ADDR register to select a page to erase
- ◆ Set the STARTE bit in the FLASH_CTRL register
- ◆ Wait for the BUSY bit to be reset
- ◆ Read the erased page and verify

15.8.3.2 MASS ERASE

When the Flash memory read protection is changed from protected to unprotected, a Mass Erase of the User ROM is performed by HW before reprogramming the read protection option.

15.9 READ PROTECTION

The read protection is activated by setting the Code Security bytes in Code option.

When the Flash memory read protection is changed from protected to unprotected, a Mass Erase of the User ROM is performed by HW before reprogramming the read protection option.

15.10 FMC REGISTERS

Base Address: 0x4006 2000

15.10.1 Flash Status register (FLASH_STATUS)

Address offset: 0x04

Reset value: 0x0000 0000

| Bit | Name | Description | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:6 | Reserved | | R | 0 |
| 5 | EOP | End of operation flag 0: Flash operation (programming/erase) is not completed. 1: Set by HW when a Flash operation (programming/erase) is completed, and is cleared on the beginning of a Flash operation. | R | 1 |
| 4:3 | Reserved | | R | 0 |
| 2 | PGERR | Programming error flag 0: Read→No error. Write→Clear this flag. 1: Set by HW when the address to be programmed contains a value different from 0xFFFFFFFF before programming. | R/W | 0 |
| 1 | Reserved | | R | 0 |
| 0 | BUSY | Busy flag 0: Flash operation is not busy. 1: Flash operation is in progress. This is set on the beginning of a Flash operation (clear EOP bit at the same time) and reset when the operation finishes or when an error occurs by HW. | R | 0 |

15.10.2 Flash Control register (FLASH_CTRL)

Address offset: 0x08

*** Note: HCLK MUST be equal or less than 24MHz during Flash program and erase operations.**

| Bit | Name | Description | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:8 | Reserved | | R | 0 |
| 7 | CHK | Checksum calculation chosen 1: Triggers Checksum calculation when set. This bit is set only by SW and reset when the BUSY bit resets. | R/W | 0 |
| 6 | STARTE | Start Erase operation 1: Triggers an ERASE operation when set. This bit is set only by SW and resets when the BUSY bit resets. PER bit shall also be 1 when setting this bit. | R/W | 0 |
| 5:2 | Reserved | | R | 0 |
| 1 | PER | Page Erase chosen. This bit is set only by SW and reset when the BUSY bit resets. | R/W | 0 |
| 0 | PG | Flash Programming chosen. This bit is set only by SW and reset when the BUSY bit resets. | R/W | 0 |

15.10.3 Flash Data register (FLASH_DATA)

Address offset: 0x0C

For Page Program operations, this should be updated by SW to indicate the data to be programmed.

| Bit | Name | Description | Attribute | Reset |
|------|------------|------------------------|-----------|-------|
| 31:0 | DATA[31:0] | Data to be programmed. | R/W | 0 |

15.10.4 Flash Address register (FLASH_ADDR)

Address offset: 0x10

The Flash address to be erased or programmed should be updated by SW, and the PG bit or PER bit shall be set before filling in the Flash address.

*** Note: Write access to this register is blocked when the BUSY bit in the FLASH_STATUS register is set.**

| Bit | Name | Description | Attribute | Reset |
|------|-----------|--|-----------|-------|
| 31:0 | FAR[31:0] | Flash Address Choose the Flash address to erase when Page Erase is selected, or to program when Page Program is selected. | R/W | 0 |

15.10.5 Flash Checksum register (FLASH_CHKSUM)

Address offset: 0x14

| Bit | Name | Description | Attribute | Reset |
|-------|--------------|-----------------------|-----------|-------|
| 31:16 | Reserved | | R | 0 |
| 15:0 | CHKSUM[15:0] | Checksum of User ROM. | R | 0 |

16 SERIAL-WIRE DEBUG (SWD)

16.1 OVERVIEW

SWD functions are integrated into the ARM Cortex-M0. The ARM Cortex-M0 is configured to support up to four breakpoints and two watch points.

16.2 FEATURES

- Supports ARM Serial Wire Debug (SWD) mode.
- Direct debug access to all memories, registers, and peripherals.
- No target resources are required for the debugging session.
- Up to four breakpoints.
- Up to two data watch points that can also be used as triggers.

16.3 PIN DESCRIPTION

| Pin Name | Type | Description | GPIO Configuration |
|----------|------|--|--------------------|
| SWCLK | I | Serial Wire Clock pin in SWD mode. | |
| SWDIO | I/O | Serial Wire Data Input/Output pin in SWD mode. | |

16.4 DEBUG NOTE

16.4.1 LIMITATIONS

Debug mode changes the way in which reduced power modes work internal to the ARM Cortex-M0 CPU, and this ripples through the entire system. These differences mean that power measurements should not be made while debugging, the results will be higher than during normal operation in an application.

During a debugging session, the SysTick Timer is automatically stopped whenever the CPU is stopped. Other peripherals are not affected.

16.4.2 DEBUG RECOVERY

User code may disable SWD function in order to use P0.10 and P0.11 as GPIO, and may not debug by SWD function to debug or download FW any more.

SONiX provide Boot loader to check the status of P0.2 (BOOT pin) during boot procedure if BLEN=1. If P0.2 is Low during Boot procedure, MCU will execute code in Boot loader instead of User code, so SWD function is not disabled.

Exit Boot loader, user code can still configure P0.2 as other functions such as GPIO.

* **Note: We strongly recommended NOT using BOOT pin as output pin to drive the LED, otherwise, the BOOT pin status may be low during boot procedure.**

16.4.3 INTERNAL PULL-UP/DOWN RESISTORS on SWD PINS

To avoid any uncontrolled IO levels, the device embeds internal pull-up and pull-down resistor on the SWD input pins:

- NJTRST: Internal pull-up
- SWDIO/JTMS: Internal pull-up
- SWCLK/JTCK: Internal pull-down

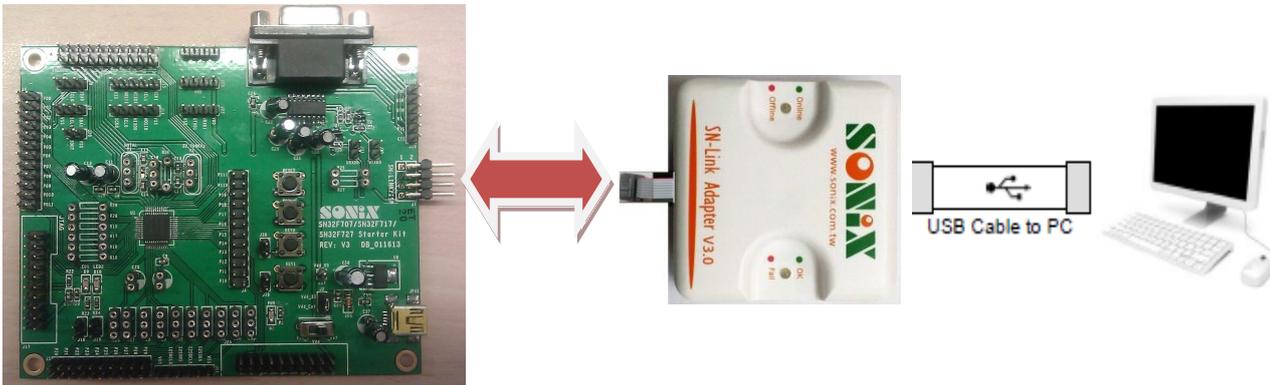
Once a SWD function is disabled by SW, the GPIO controller takes control again.

17 DEVELOPMENT TOOL

SONiX provides an Embedded ICE emulator system to offer 32-bit series MCU firmware development.

SONiX 32-bit series Embedded ICE Emulator System includes:

- SONiX 32-bit MCU Starter-Kit.
- SN-LINK-V3
- USB cable to provide communications between the SN-LINK-V3 and PC.
- IDE Tools (KEIL RVMDK)



SONiX 32-bit MCU Starter-Kit.

SN-LINK-V3

IDE Tools

SONiX 32-bit series Embedded ICE Emulator Feature:

- Target's Operating Voltage: 1.8V~3.6V.
- Up to 4 hardware break points.
- System clock rate up to 50MHz.
- Oscillator supports IHRC, ILRC, EHS/ELS X'tal.

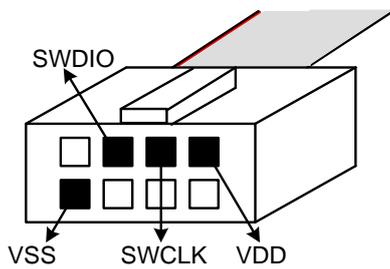
SONiX 32-bit series Embedded ICE Emulator Limitation:

- SWCLK and SWDIO pins are shared with GPIO pins. In embedded ICE mode, the shared GPIO function can't work.

17.1 SN-LINK-V3

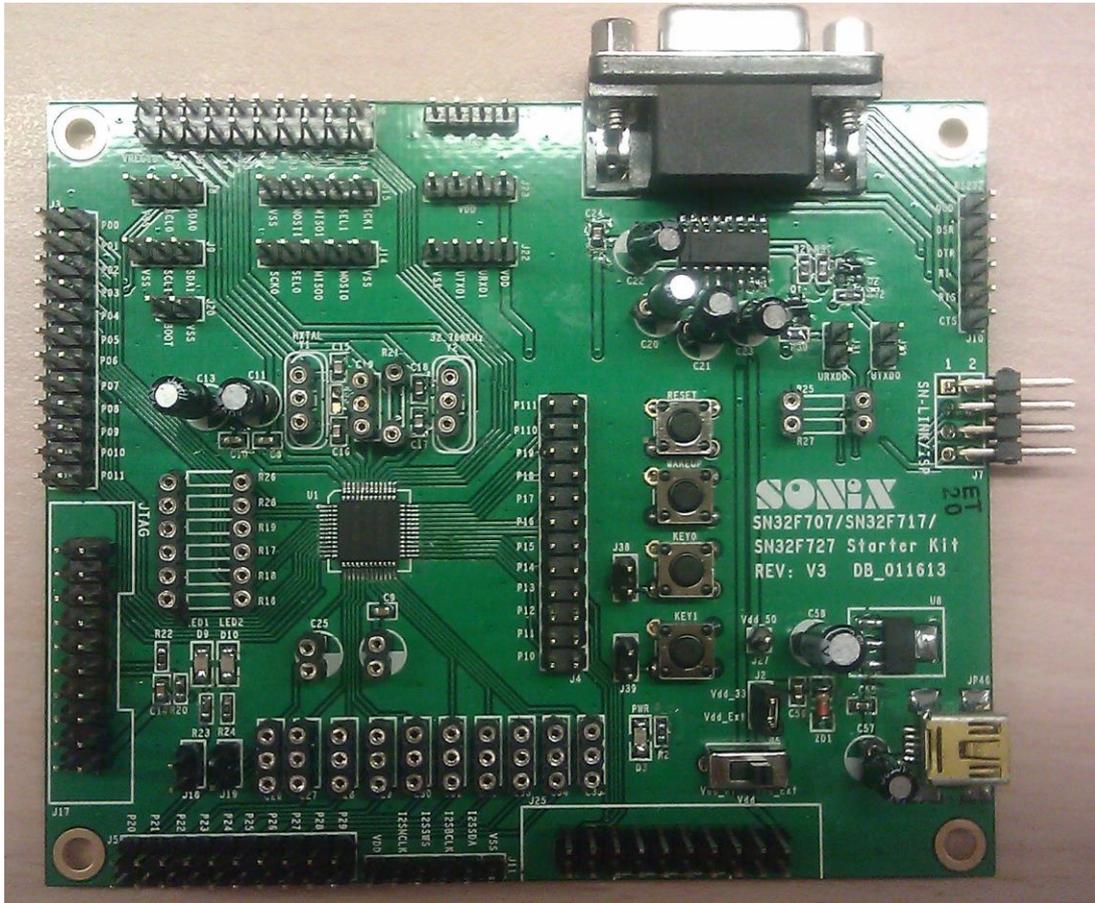
SN-LINK-V3 is a high speed emulator for SONiX 32-bit MCU. It debugs and programs based on SWD protocol. In addition to debugger functions, the SN-LINK-V3 also may be used as a programmer to load firmware from PC to MCU for engineering production, even mass production.

SN-LINK-V3 communicates with SONiX 32-bit MCU through SWD interface. The pin definition of the Modular cable is as following:



17.2 SN32F707 STARTER-KIT

SONiX 32-bit MCU Starter-kit is an easy-development platform. It includes real chip and I/O connectors to input signal or drive extra device of user's application. It is a simple platform to develop application as target board not ready. The starter-kit can be replaced by target board because of integrated SWD debugger circuitry.



18 ELECTRICAL CHARACTERISTIC

18.1 ABSOLUTE MAXIMUM RATING

| | |
|---|-------------------------|
| Supply voltage (Vdd)..... | - 0.3V ~ 3.6V |
| Input in voltage (Vin)..... | Vss – 0.2V ~ Vdd + 0.2V |
| Operating ambient temperature (Topr)..... | -40°C ~ + 85°C |
| Storage ambient temperature (Tstor) | -40°C ~ + 125°C |

18.2 ELECTRICAL CHARACTERISTIC

| Standard Operating Conditions (Typical temperature Ta = 25°C) | | | | | | | |
|---|--------------------|---|--------------------------------------|------|--------|----------------------|----|
| PARAMETER | SYM. | DESCRIPTION | MIN. | TYP. | MAX. | UNIT | |
| Operating Voltage | Vdd | Supply voltage for core and external rail | 1.8 | 3.3 | 3.6 | V | |
| VDD rise rate | V _{POR} | VDD rise rate to ensure internal power-on reset | 0.05 | - | - | V/ms | |
| Power Consumption | | | | | | | |
| Supply Current | I _{dd1} | Normal mode | System clock = 12MHz [1][2][3] | - | 7 | - | mA |
| | | | System clock = 50MHz [1][3][4] | - | 20 | - | mA |
| | I _{dd2} | Sleep Mode | System clock = 12MHz [1][2][3][5] | - | 500 | - | µA |
| | I _{dd3} | Sleep Mode | System clock = 16KHz [1][3][5][7] | - | 100 | - | µA |
| | I _{dd4} | Deep-sleep Mode | Vdd=3.3V [1][3][5] | - | 6 | - | µA |
| | I _{dd5} | Deep power-down Mode | Vdd=3.3V [6] | - | 150 | - | nA |
| Port Pins, RESET pin | | | | | | | |
| High-level input voltage | V _{IH} | | 0.8Vdd | - | Vdd | V | |
| Low-level input voltage | V _{IL} | | Vss | - | 0.2Vdd | V | |
| Input voltage | V _i | | 0 | - | Vdd | V | |
| Output voltage | V _o | | 0 | - | Vdd | V | |
| I/O port pull-up resistor | R _{PU} | Vin = Vss , Vdd = 3.3V | 40 | 60 | 80 | KΩ | |
| I/O port pull-down resistor | R _{PD} | Vin = 3.3V | 40 | 60 | 80 | KΩ | |
| I/O port input leakage current | I _{lekg} | Pull-up resistor disable, Vin = Vdd | - | - | 2 | µA | |
| | | I2C-bus pins (P0.2, P0.3, P0.4 and P0.5), Vin = Vdd | - | 2 | 4 | µA | |
| Output Voltage | V _{OH} | | 0 | - | Vdd | V | |
| I/O High-level output source current | I _{OH} | Standard port and RESET pins | V _{OP} = Vdd – 0.5V; | 5 | 10 | - | mA |
| | | High-drive output pin (P0.0~P0.5, P1.6~P1.8) | V _{OP} = Vdd – 0.5V | 12 | 20 | - | mA |
| I/O Low-level output sink current | I _{OL} | Standard port and RESET pins | V _{OP} = Vss + 0.5V | 5 | 10 | - | mA |
| ADC | | | | | | | |
| ADC Operating Voltage | V _{ADC} | | 2.5 | | 3.6 | V | |
| AIN0 ~ AIN9 input voltage | V _{ani} | Vdd=3.3V | 0 | - | Avrefh | V | |
| ADC reference Voltage | V _{ref} | | 2.5 | - | - | V | |
| *ADC enable time | T _{ast} | Ready to start convert after set ADENB = "1" | 100 | - | - | us | |
| *ADC current consumption | I _{ADC} | Vdd=3.3V, ADS=0 | - | 220 | - | µA | |
| ADC Clock Frequency | F _{ADCLK} | Vdd=3.3V | - | - | 16 | MHz | |
| ADC Conversion Cycle Time | F _{ADCYL} | VDD=2.5V~3.6V | 64 | - | - | 1/F _{ADCLK} | |
| ADC Sampling Rate | F _{ADSMP} | Vdd=3.3V | | - | 125 | KHz | |
| Differential Nonlinearity | DNL | Vdd=3.6V , AVREFH=2.4V | -1 | - | +1 | LSB | |

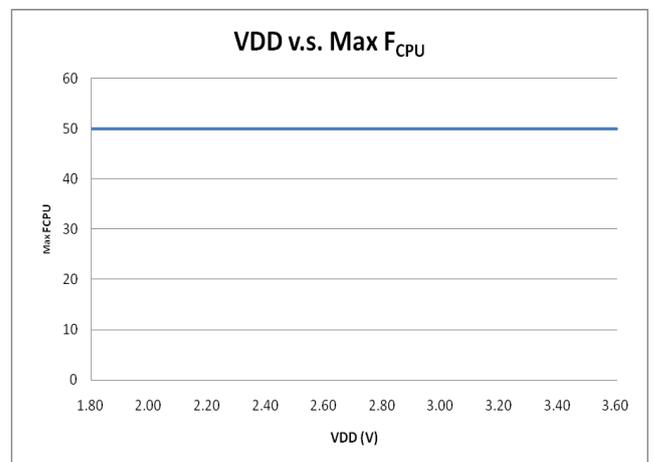
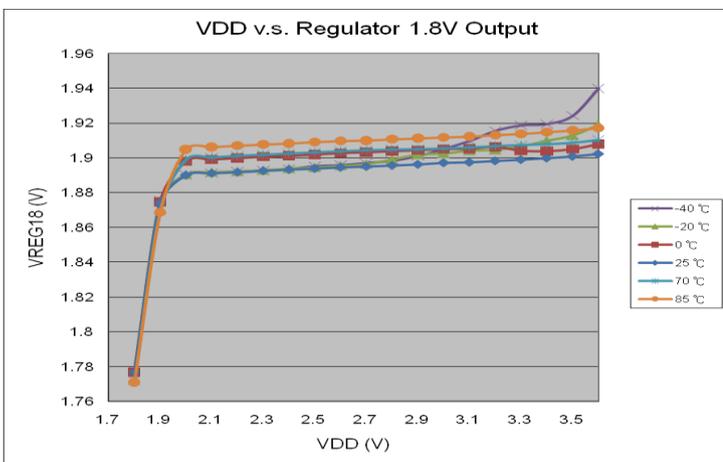
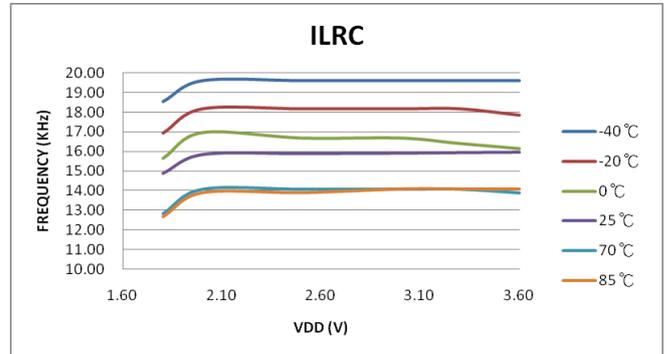
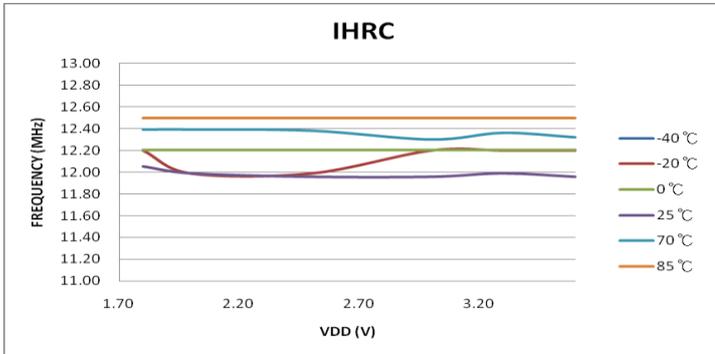
| | | | | | | | |
|-------------------------|------------------------|---|---------|-------|-----------------|-------|---|
| Integral Nonlinearity | INL | V _{dd} =3.6V , AVREFH=2.4V | -1 | - | +1 | LSB | |
| No Missing Code | NMC | V _{dd} =3.6V , AVREFH=2.4V | 10 | - | 12 | Bits | |
| ADC offset Voltage | V _{ADCOffset} | | -5 | | +5 | mV | |
| FLASH | | | | | | | |
| Supply Voltage | V _{dd1} | | 1.8 | | V _{dd} | V | |
| Endurance time | T _{EN} | Erase + Program | 20K | *100K | - | Cycle | |
| Page erase time | T _{PE} | 1-Page (512 bytes). | - | 25 | 30 | ms | |
| 1-Word Programming time | T _{PG} | 1-Word (32 bits). | - | 60 | 70 | us | |
| MISC | | | | | | | |
| Low Voltage Detector | LVD | Interrupt | Level 0 | 1.90 | 2.00 | 2.10 | V |
| | | | Level 1 | 2.60 | 2.70 | 2.80 | V |
| | | | Level 2 | 2.90 | 3.00 | 3.10 | V |
| | | Reset | Level 0 | 1.90 | 2.00 | 2.10 | V |
| | | | Level 1 | 2.30 | 2.40 | 2.50 | V |
| | | | Level 2 | 2.60 | 2.70 | 2.80 | V |
| IHRC Freq. | F _{IHRC} | T=25°C, V _{dd} =1.8V~ 3.6V | 11.76 | 12 | 12.24 | MHz | |
| | | T=-40°C ~85°C, V _{dd} =1.8V~3.6V | 11.40 | 12 | 12.60 | MHz | |

* **These parameters are for design reference, not tested.**

- [1] I_{DD} measurements were performed with all pins configured as GPIO outputs driven LOW and pull-up resistors disabled and V_{DD}=3.3V
- [2] IHRC and ILRC are enabled, external X'tal are disabled, and PLL is disabled.
- [3] LVD and all peripherals are disabled.
- [4] IHRC is disabled, external high X'tal is enabled, and PLL is enabled.
- [5] All oscillators and analog blocks are turned off.
- [6] DPDWAKEUP pin is pulled HIGH internally.
- [7] ILRC is enabled, IHRC and external X'tal are disabled, and PLL is disabled.

18.3 CHARACTERISTIC GRAPHS

The Graphs in this section are for design guidance, not tested or guaranteed. In some graphs, the data presented are outside specified operating range. This is for information only and devices are guaranteed to operate properly only within the specified range.



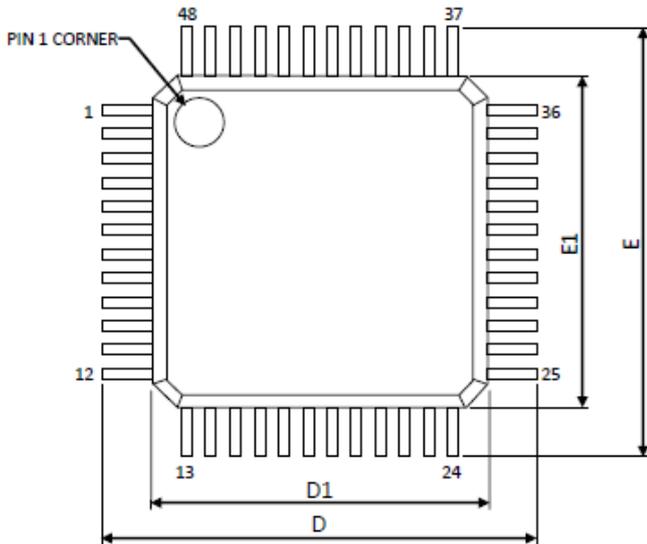
19 FLASH ROM PROGRAMMING PIN

| Programming Information of SN32F700 Series | | | | | | | | | | | |
|--|----------|-------------------------------|-----------|-----------|------------|---------------|------|---------------|------|---------|------|
| Chip Name | | SN32F707F | SN32F706J | SN32F705J | SN32F7051J | SN32F704K/S/X | | | | | |
| MP PRO Writer Connector | | Flash IC / JP3 Pin Assignment | | | | | | | | | |
| Number | Name | Number | Pin | Number | Pin | Number | Pin | Number | Pin | Number | Pin |
| 1 | VDD | 18 48 | VDD | 16 46 | VDD | 8 24 | VDD | 8 24 | VDD | 7 23 | VDD |
| 2 | GND | 19 46 | VSS | 17 44 | VSS | 9 22 33 | VSS | 9 22 33 | VSS | 8 21 | VSS |
| 3 | CLK | 25 | P1.0 | 23 | P1.0 | 10 | P1.0 | 10 | P1.0 | 9 | P1.0 |
| 4 | CE | | | | | | | | | | |
| 5 | PGM | 27 | P1.2 | 25 | P1.2 | 12 | P1.2 | 12 | P1.2 | 11 | P1.2 |
| 6 | OE | 26 | P1.1 | 24 | P1.1 | 11 | P1.1 | 11 | P1.1 | 10 | P1.1 |
| 7 | D1 | | | | | | | | | | |
| 8 | D0 | | | | | | | | | | |
| 9 | D3 | | | | | | | | | | |
| 10 | D2 | | | | | | | | | | |
| 11 | D5 | | | | | | | | | | |
| 12 | D4 | | | | | | | | | | |
| 13 | D7 | | | | | | | | | | |
| 14 | D6 | | | | | | | | | | |
| 15 | VDD | | | | | | | | | | |
| 16 | - | | | | | | | | | | |
| 17 | HLS | | | | | | | | | | |
| 18 | RST | | | | | | | | | | |
| 19 | - | | | | | | | | | | |
| 20 | ALSB/PDB | 28 | P1.3 | 26 | P1.3 | 13 | P1.3 | 13 | P1.3 | 12 | P1.3 |

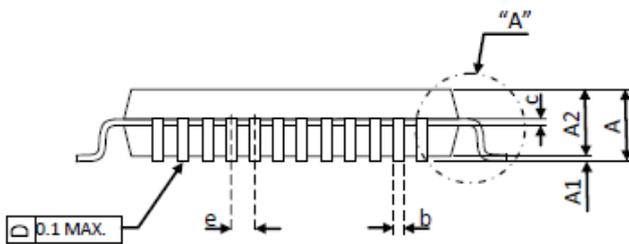
| Programming Information of SN32F700 Series | | | | | | | | | | | |
|--|----------|-------------------------------|------------|---------|------|--------|-----|--------|-----|--------|-----|
| Chip Name | | SN32F703K/S/X | SN32F7031J | | | | | | | | |
| MP PRO Writer Connector | | Flash IC / JP3 Pin Assignment | | | | | | | | | |
| Number | Name | Number | Pin | Number | Pin | Number | Pin | Number | Pin | Number | Pin |
| 1 | VDD | 6 20 | VDD | 4 18 | VDD | | | | | | |
| 2 | GND | 7 18 | VSS | 5 16 | VSS | | | | | | |
| 3 | CLK | 8 | P1.0 | 6 | P1.0 | | | | | | |
| 4 | CE | | | | | | | | | | |
| 5 | PGM | 10 | P1.2 | 8 | P1.2 | | | | | | |
| 6 | OE | 9 | P1.1 | 7 | P1.1 | | | | | | |
| 7 | D1 | | | | | | | | | | |
| 8 | D0 | | | | | | | | | | |
| 9 | D3 | | | | | | | | | | |
| 10 | D2 | | | | | | | | | | |
| 11 | D5 | | | | | | | | | | |
| 12 | D4 | | | | | | | | | | |
| 13 | D7 | | | | | | | | | | |
| 14 | D6 | | | | | | | | | | |
| 15 | VDD | | | | | | | | | | |
| 16 | - | | | | | | | | | | |
| 17 | HLS | | | | | | | | | | |
| 18 | RST | | | | | | | | | | |
| 19 | - | | | | | | | | | | |
| 20 | ALSB/PDB | 11 | P1.3 | 9 | P1.3 | | | | | | |

20 PACKAGE INFORMATION

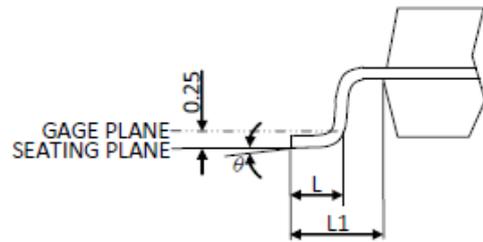
20.1 LQFP 48 PIN



TOP VIEW



SIDE VIEW



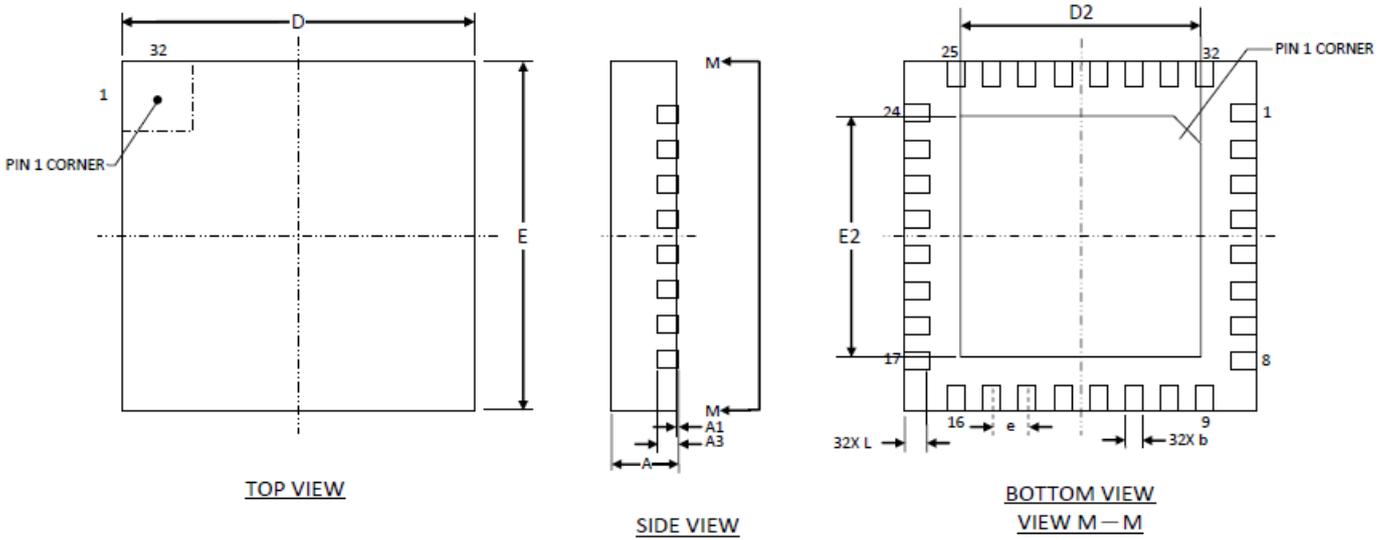
DETAIL "A"

| SYMBOLS | Dimension in mm | | | Dimension in inch | | |
|----------|-----------------|------|------|-------------------|-------|-------|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | -- | -- | 1.60 | -- | -- | 0.063 |
| A1 | 0.05 | -- | 0.15 | 0.002 | -- | 0.006 |
| A2 | 1.35 | 1.40 | 1.45 | 0.053 | 0.055 | 0.057 |
| b | 0.17 | 0.22 | 0.27 | 0.007 | 0.009 | 0.011 |
| c | 0.09 | -- | 0.20 | 0.004 | -- | 0.008 |
| D | 9.00 BSC | | | 0.354 BSC | | |
| D1 | 7.00 BSC | | | 0.276 BSC | | |
| E | 9.00 BSC | | | 0.354 BSC | | |
| E1 | 7.00 BSC | | | 0.276 BSC | | |
| e | 0.50 BSC | | | 0.020 BSC | | |
| L | 0.40 | 0.60 | 0.80 | 0.016 | 0.024 | 0.031 |
| L1 | 1.00 REF | | | 0.039 REF | | |
| θ | 0° | 3.5° | 7° | 0° | 3.5° | 7° |

Notes :

1. CONTROLLING DIMENSION : MILLIMETER (mm)
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

20.3 QFN 33 PIN 4x4

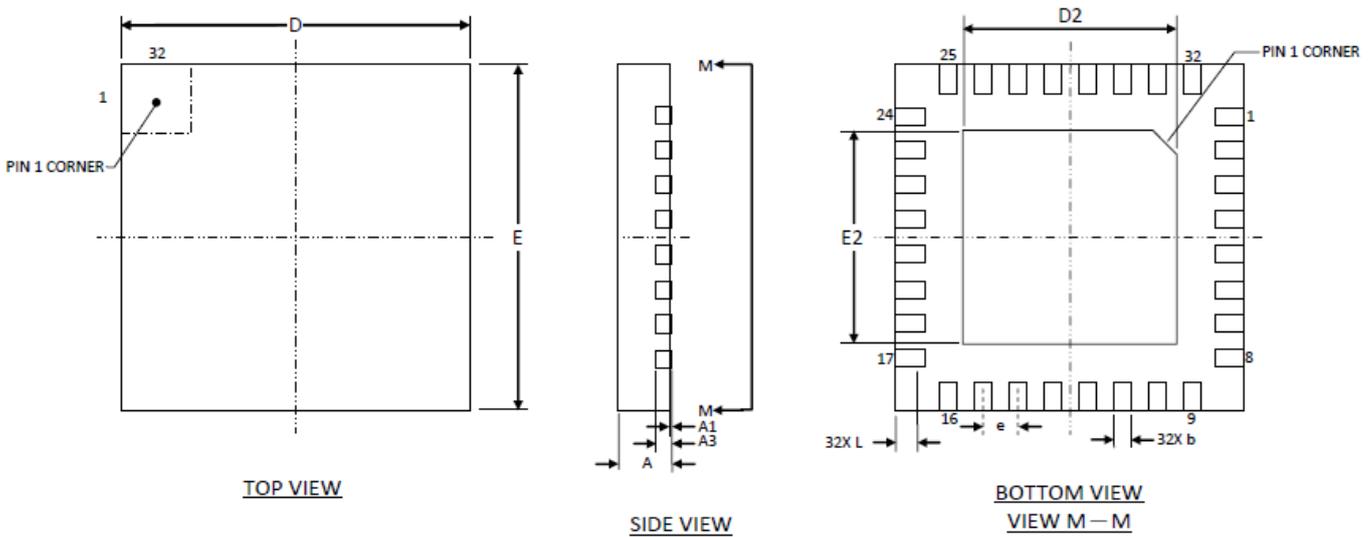


| SYMBOLS | Dimension in mm | | | Dimension in inch | | |
|---------|-----------------|------|------|-------------------|-------|-------|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | 0.70 | 0.80 | 0.90 | 0.028 | 0.031 | 0.035 |
| A1 | 0.00 | 0.02 | 0.05 | 0.000 | 0.000 | 0.002 |
| A3 | 0.20 REF | | | 0.008 REF | | |
| b | 0.15 | 0.20 | 0.25 | 0.006 | 0.008 | 0.010 |
| D | 4.00 BSC | | | 0.157 BSC | | |
| E | 4.00 BSC | | | 0.157 BSC | | |
| e | 0.40 BSC | | | 0.016 BSC | | |
| D2 | 2.00 | 2.45 | 2.9 | 0.080 | 0.096 | 0.114 |
| E2 | 2.00 | 2.45 | 2.9 | 0.080 | 0.096 | 0.114 |
| L | 0.25 | 0.35 | 0.45 | 0.010 | 0.013 | 0.017 |

Notes :

1. CONTROLLING DIMENSION : MILLIMETER (mm)

20.4 QFN 33 PIN 5x5

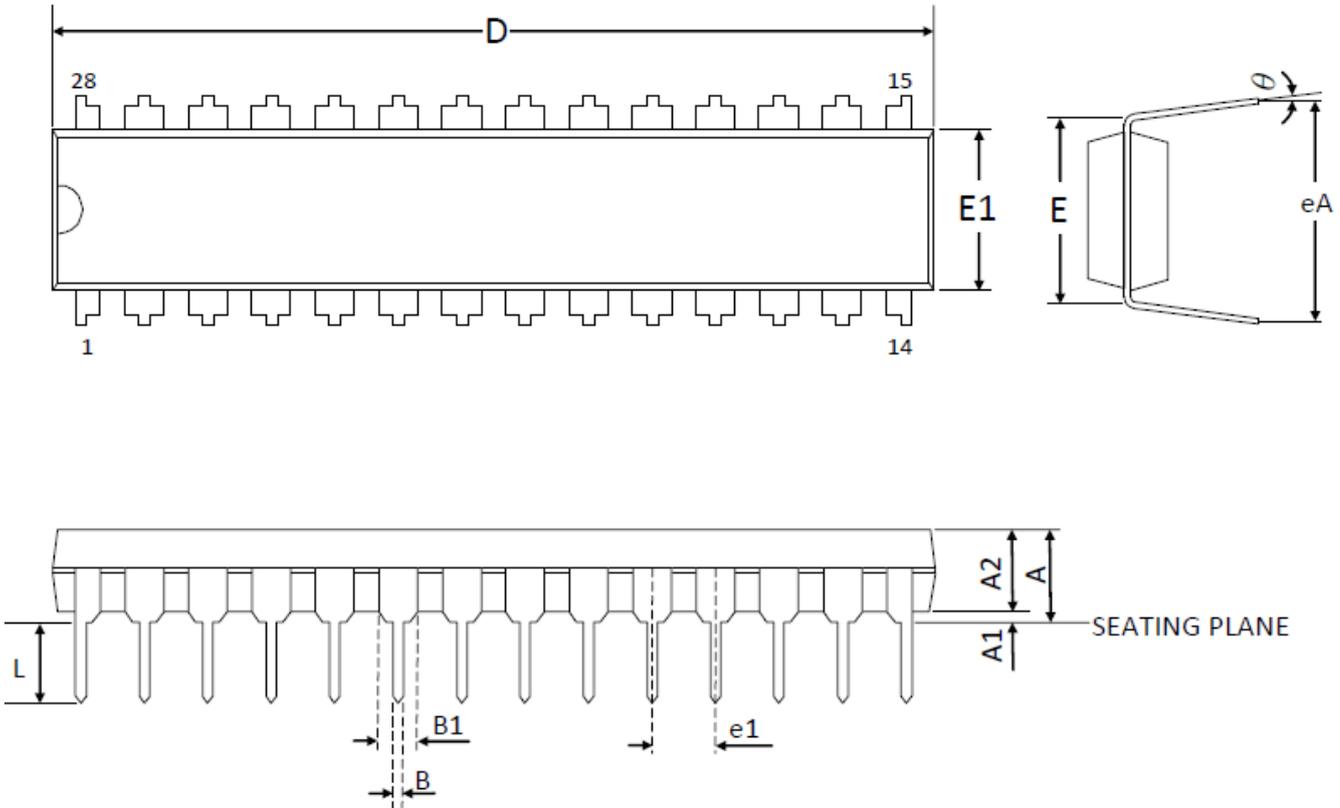


| SYMBOLS | Dimension in mm | | | Dimension in inch | | |
|---------|-----------------|------|------|-------------------|-------|-------|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | 0.70 | 0.80 | 0.90 | 0.028 | 0.031 | 0.035 |
| A1 | 0.00 | 0.02 | 0.05 | 0.000 | 0.001 | 0.002 |
| A3 | 0.203 REF | | | 0.008 REF | | |
| b | 0.18 | 0.25 | 0.30 | 0.007 | 0.010 | 0.012 |
| D | 5.00 BSC | | | 0.197 BSC | | |
| E | 5.00 BSC | | | 0.197 BSC | | |
| e | 0.50 BSC | | | 0.020 BSC | | |
| D2 | 2.60 | 3.10 | 3.60 | 0.102 | 0.122 | 0.142 |
| E2 | 2.60 | 3.10 | 3.60 | 0.102 | 0.122 | 0.142 |
| L | 0.30 | 0.40 | 0.50 | 0.012 | 0.016 | 0.020 |

Notes :

1. CONTROLLING DIMENSION : MILLIMETER (mm)

20.5 SK-DIP 28 PIN

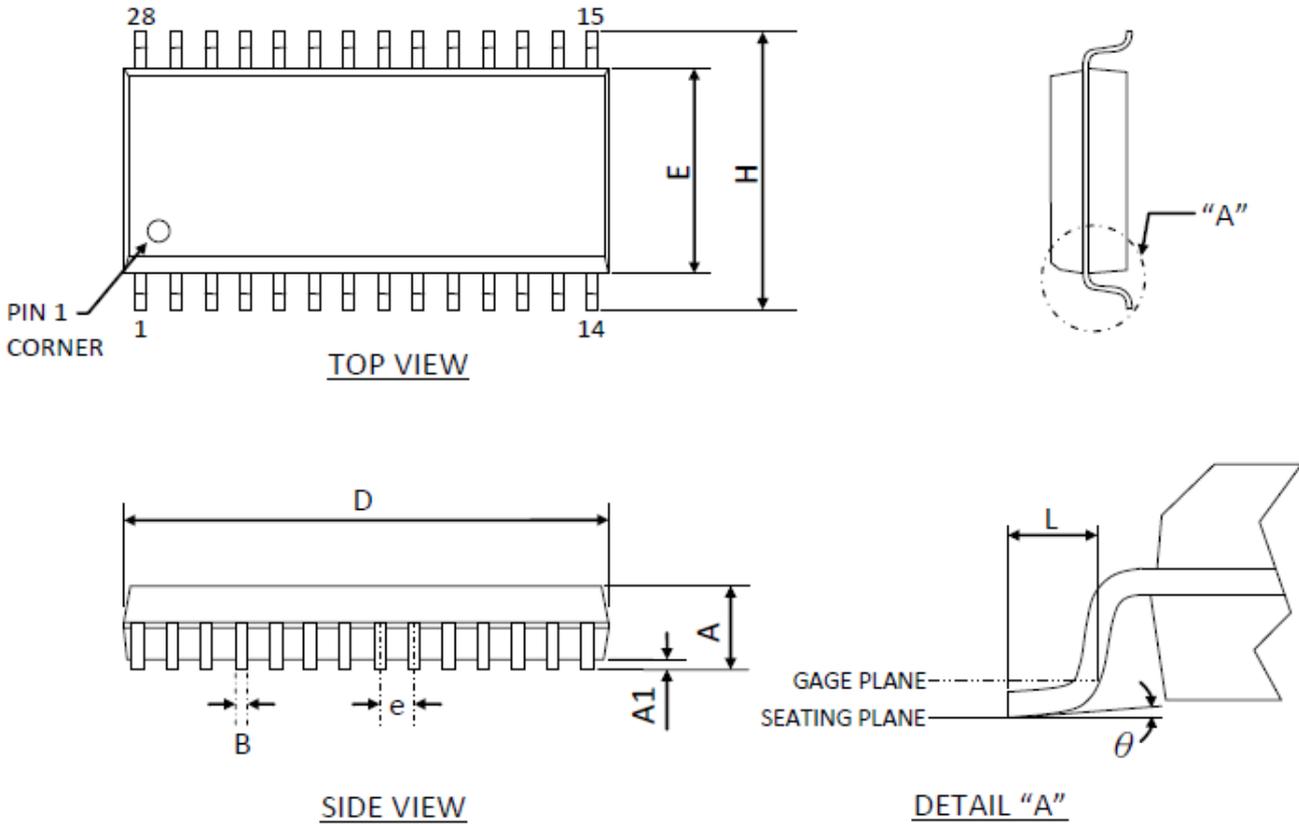


| SYMBOLS | Dimension in mm | | | Dimension in inch | | |
|----------|-----------------|-------|-------|-------------------|-------|-------|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | -- | -- | 4.45 | -- | -- | 0.175 |
| A1 | 0.38 | -- | -- | 0.015 | -- | -- |
| A2 | 3.18 | 3.30 | 3.43 | 0.125 | 0.130 | 0.135 |
| B | 0.46 typ. | | | 0.018 typ. | | |
| B1 | 1.52 typ. | | | 0.060 typ. | | |
| D | 35.18 | 35.31 | 35.56 | 1.385 | 1.390 | 1.400 |
| E | 7.87 BSC | | | 0.310 BSC | | |
| E1 | 7.19 | 7.32 | 7.44 | 0.283 | 0.288 | 0.293 |
| e1 | 2.54 typ. | | | 0.100 typ. | | |
| L | 3.05 | 3.30 | 3.56 | 0.120 | 0.130 | 0.140 |
| eA | 8.38 | 8.89 | 9.40 | 0.330 | 0.350 | 0.370 |
| θ | 0° | 7° | 15° | 0° | 7° | 15° |

Notes :

1. JEDEC OUTLINE : MS-015 AH
2. CONTROLLING DIMENSION : inch

20.6 SOP 28 PIN

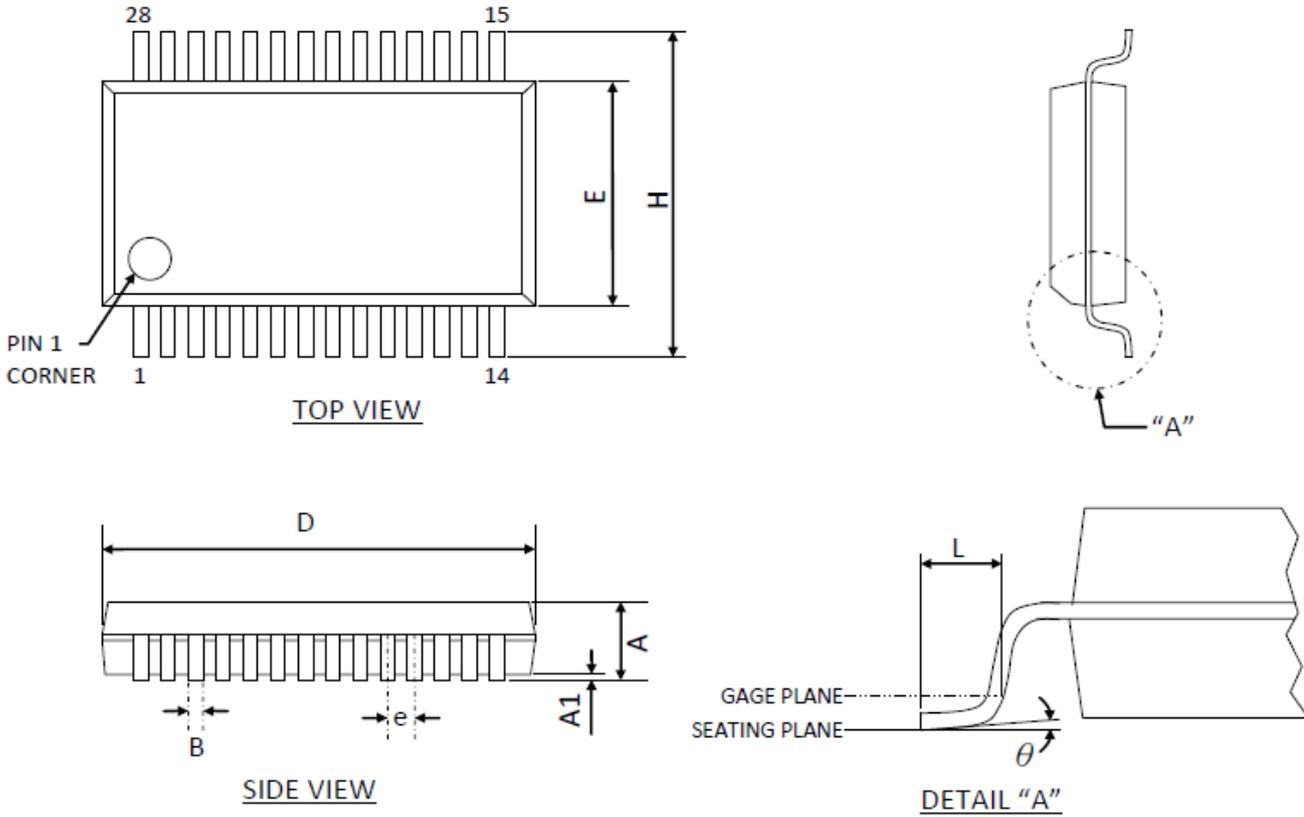


| SYMBOLS | Dimension in mm | | | Dimension in inch | | |
|----------|-----------------|-------|-------|-------------------|-------|-------|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | -- | -- | 2.65 | -- | -- | 0.104 |
| A1 | 0.10 | -- | 0.30 | 0.004 | -- | 0.011 |
| B | 0.31 | 0.41 | 0.51 | 0.012 | 0.016 | 0.020 |
| D | 17.70 | 18.20 | 18.70 | 0.697 | 0.716 | 0.736 |
| E | 7.50 BSC | | | 0.295 BSC | | |
| e | 1.27 BSC | | | 0.050 BSC | | |
| H | 10.30 BSC | | | 405 BSC | | |
| L | 0.40 | -- | 1.27 | 0.016 | -- | 0.050 |
| θ | 0° | 4° | 8° | 0° | 4° | 8° |

Notes :

1. CONTROLLING DIMENSION : mm
2. JEDEC OUTLINE : MO-119 AB

20.7 SSOP 28 PIN

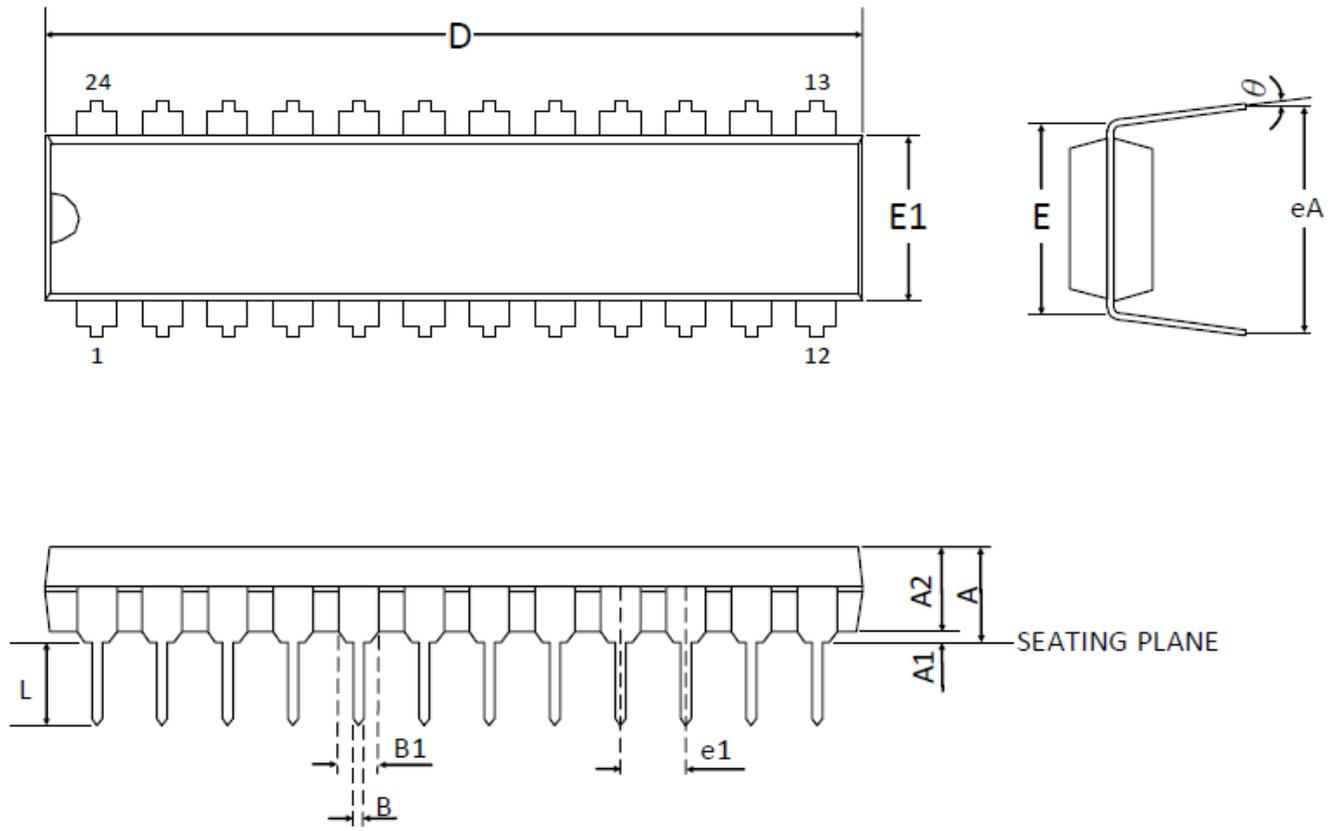


| SYMBOLS | Dimension in mm | | | Dimension in inch | | |
|----------|-----------------|-------|-------|-------------------|-------|-------|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | -- | -- | 2.0 | -- | -- | 0.079 |
| A1 | 0.05 | -- | -- | 0.002 | -- | -- |
| B | 0.22 | -- | 0.38 | 0.009 | -- | 0.015 |
| D | 10.05 | 10.20 | 10.50 | 0.396 | 0.402 | 0.413 |
| E | 5.00 | 5.30 | 5.60 | 0.197 | 0.209 | 0.220 |
| e | 0.65 BSC. | | | 0.026 BSC. | | |
| H | 7.65 | 7.80 | 7.90 | 0.301 | 0.307 | 0.311 |
| L | 0.55 | 0.80 | 1.05 | 0.022 | 0.031 | 0.041 |
| θ | 0° | 4° | 8° | 0° | 4° | 8° |

Notes :

1. CONTROLLING DIMENSION : mm
2. JEDEC OUTLINE : MO-105 AH

20.8 SK-DIP 24 PIN

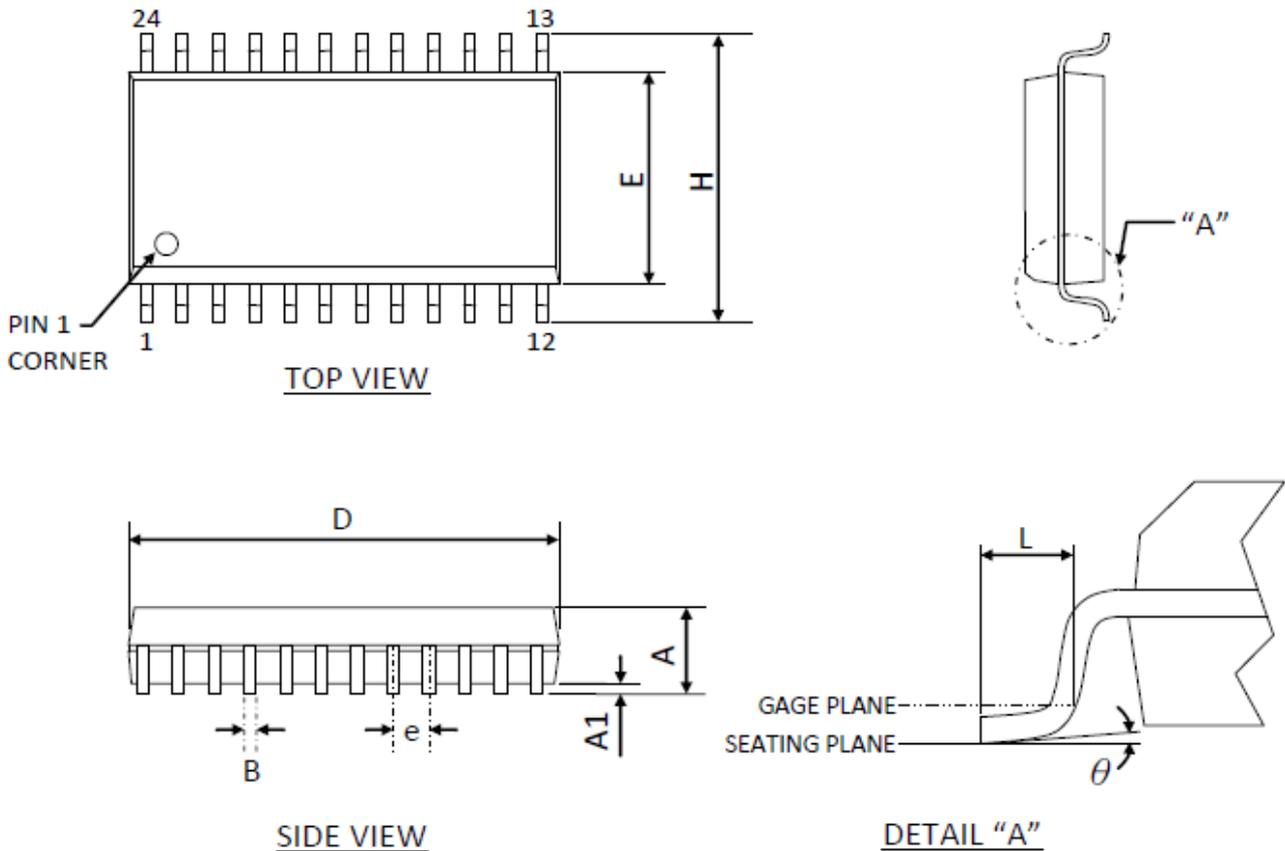


| SYMBOLS | Dimension in mm | | | Dimension in inch | | |
|----------|-----------------|-------|-------|-------------------|-------|-------|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | -- | -- | 5.33 | -- | -- | 0.210 |
| A1 | 0.35 | -- | -- | 0.015 | -- | -- |
| A2 | 3.18 | 3.30 | 3.43 | 0.125 | 0.130 | 0.135 |
| B | 0.46 typ. | | | 0.018 typ. | | |
| B1 | 1.52 typ. | | | 0.060 typ. | | |
| D | 31.24 | 31.75 | 32.51 | 1.230 | 1.250 | 1.280 |
| E | 7.62 BSC. | | | 0.300 BSC. | | |
| E1 | 6.43 | 6.55 | 6.68 | 0.253 | 0.258 | 0.263 |
| e1 | 2.54 typ. | | | 0.100 typ. | | |
| L | 2.92 | 3.30 | 3.81 | 0.115 | 0.130 | 0.150 |
| eA | 8.51 | 9.02 | 9.53 | 0.335 | 0.355 | 0.375 |
| θ | 0° | 7° | 15° | 0° | 7° | 15° |

Notes :

1. JEDEC OUTLINE : MS-001 AF
2. CONTROLLING DIMENSION : inch

20.9 SOP 24 PIN

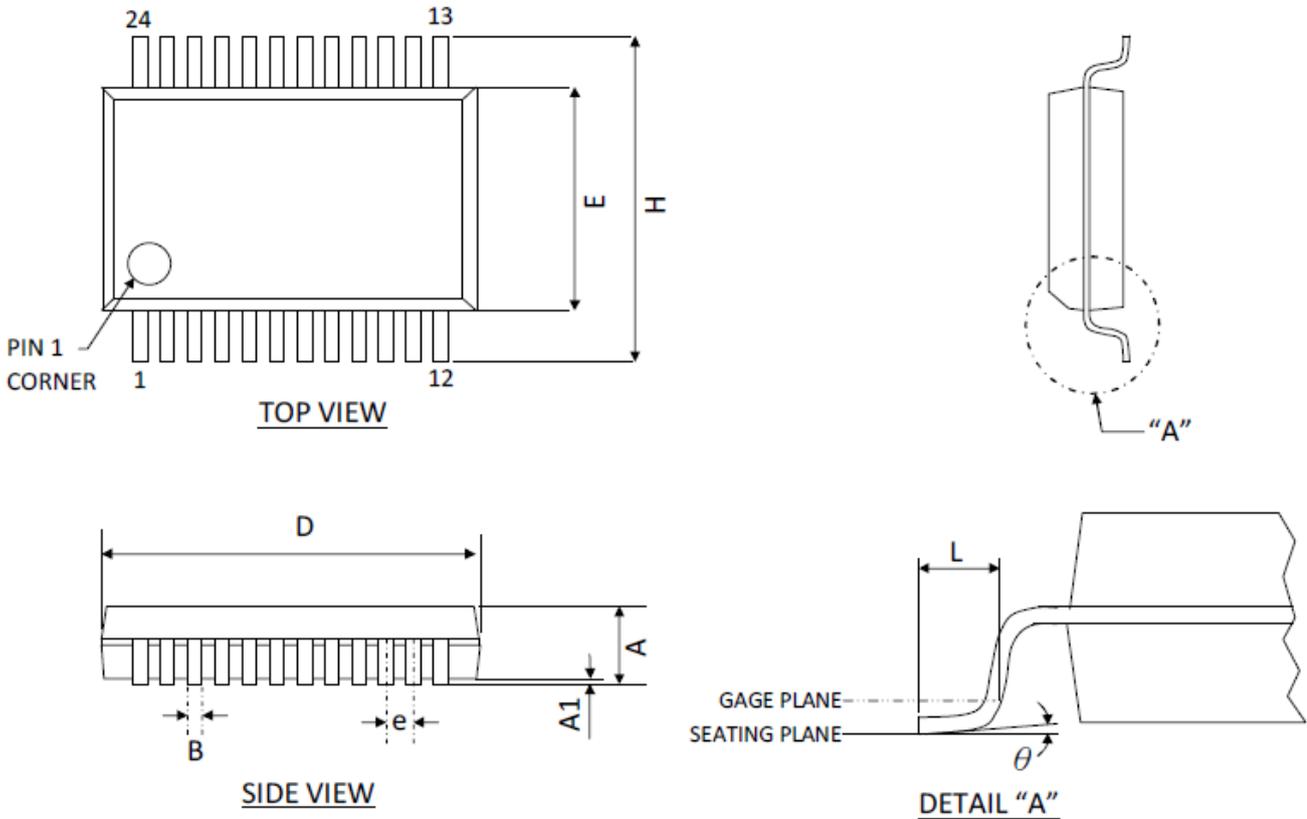


| SYMBOLS | Dimension in mm | | | Dimension in inch | | |
|----------|-----------------|-------|-------|-------------------|-------|-------|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | -- | -- | 2.65 | -- | -- | 0.104 |
| A1 | 0.10 | -- | 0.30 | 0.004 | -- | 0.011 |
| B | 0.31 | 0.41 | 0.51 | 0.012 | 0.016 | 0.020 |
| D | 15.30 | 15.50 | 15.70 | 0.602 | 0.618 | 0.618 |
| E | 7.50 BSC | | | 0.295 BSC | | |
| e | 1.27 BSC | | | 0.050 BSC | | |
| H | 10.30 BSC | | | 0.405 BSC | | |
| L | 0.4 | -- | 1.27 | 0.015 | -- | 0.05 |
| θ | 0° | 4° | 8° | 0° | 4° | 8° |

Notes :

1. CONTROLLING DIMENSION : mm
2. JEDEC OUTLINE : MO-119 AA

20.10 SSOPP 24 PIN

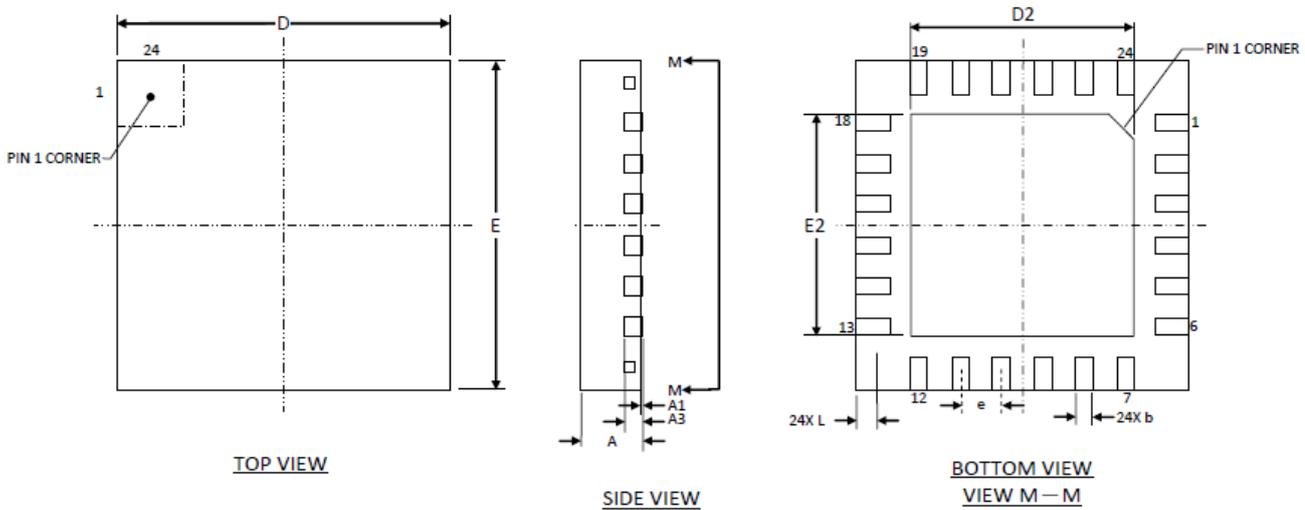


| SYMBOLS | Dimension in mm | | | Dimension in inch | | |
|----------|-----------------|------|------|-------------------|-------|-------|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | -- | -- | 2.0 | -- | -- | 0.079 |
| A1 | 0.05 | -- | -- | 0.002 | -- | -- |
| B | 0.22 | -- | 0.38 | 0.009 | -- | 0.015 |
| D | 7.90 | 8.20 | 8.50 | 0.311 | 0.322 | 0.334 |
| E | 5.00 | 5.30 | 5.60 | 0.197 | 0.209 | 0.220 |
| e | 0.65 BSC. | | | 0.026 BSC. | | |
| H | 7.40 | 7.80 | 8.20 | 0.291 | 0.307 | 0.322 |
| L | 0.55 | 0.80 | 1.05 | 0.022 | 0.031 | 0.041 |
| θ | 0° | 4° | 8° | 0° | 4° | 8° |

Notes :

1. CONTROLLING DIMENSION : mm
2. JEDEC OUTLINE : MO-105 AH

20.11 QFN 24 PIN



| SYMBOLS | Dimension in mm | | | Dimension in inch | | |
|---------|-----------------|------|------|-------------------|-------|-------|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | 0.70 | 0.80 | 0.90 | 0.028 | 0.031 | 0.035 |
| A1 | 0.00 | 0.02 | 0.05 | 0.000 | 0.001 | 0.002 |
| A3 | 0.203 REF | | | 0.008 REF | | |
| b | 0.15 | 0.25 | 0.30 | 0.007 | 0.010 | 0.012 |
| D | 4.00 BSC | | | 0.157 BSC | | |
| E | 4.00 BSC | | | 0.157 BSC | | |
| e | 0.50 BSC | | | 0.020 BSC | | |
| D2 | 1.90 | 2.35 | 2.80 | 0.075 | 0.093 | 0.110 |
| E2 | 1.90 | 2.35 | 2.80 | 0.075 | 0.093 | 0.110 |
| L | 0.30 | 0.40 | 0.50 | 0.012 | 0.016 | 0.020 |

Notes :

1. CONTROLLING DIMENSION : MILLIMETER (mm)

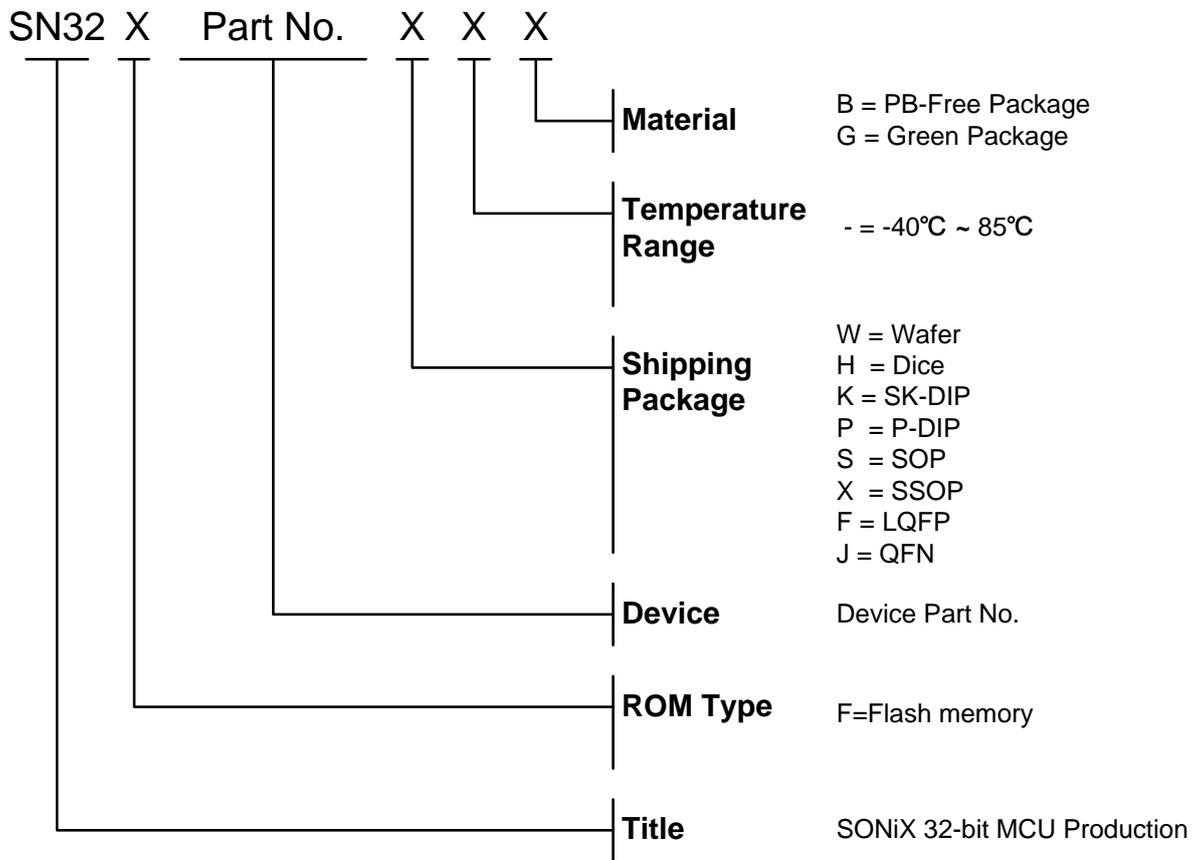
21 MARKING DEFINITION

21.1 INTRODUCTION

There are many different types in SONiX 32-bit MCU production line.

This note lists the marking definitions of all 32-bit MCU for order or obtaining information.

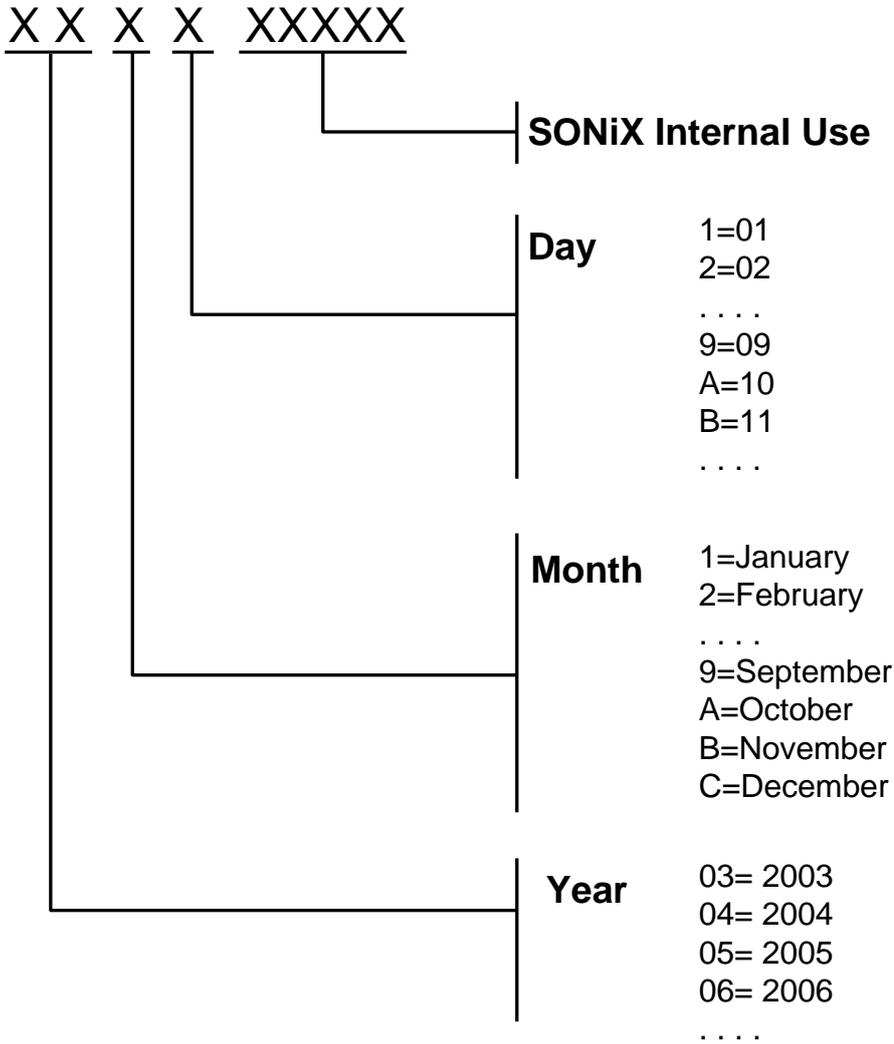
21.2 MARKING INDETIFICATION SYSTEM



21.3 MARKING EXAMPLE

| Name | ROM Type | Device | Package | Temperature | Material |
|-------------|--------------|--------|---------|-------------|---------------|
| SN32F707FG | Flash memory | 707 | LQFP | -40°C~85°C | Green Package |
| SN32F707W | Flash memory | 707 | Wafer | -40°C~85°C | - |
| SN32F707H | Flash memory | 707 | Dice | -40°C~85°C | - |
| SN32F706JG | Flash memory | 707 | QFN | -40°C~85°C | Green Package |
| SN32F705JG | Flash memory | 707 | QFN | -40°C~85°C | Green Package |
| SN32F7051JG | Flash memory | 707 | QFN | -40°C~85°C | Green Package |
| SN32F704KG | Flash memory | 707 | SK-DIP | -40°C~85°C | Green Package |
| SN32F704SG | Flash memory | 707 | SOP | -40°C~85°C | Green Package |
| SN32F704XG | Flash memory | 707 | SSOP | -40°C~85°C | Green Package |
| SN32F703KG | Flash memory | 707 | SK-DIP | -40°C~85°C | Green Package |
| SN32F703SG | Flash memory | 707 | SOP | -40°C~85°C | Green Package |
| SN32F703XG | Flash memory | 707 | SSOP | -40°C~85°C | Green Package |
| SN32F7031JG | Flash memory | 707 | QFN | -40°C~85°C | Green Package |

21.4 DATECODE SYSTEM



SONIX reserves the right to make change without further notice to any products herein to improve reliability, function or design. SONIX does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. SONIX products are not designed, intended, or authorized for use as components in systems intended, for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the SONIX product could create a situation where personal injury or death may occur. Should Buyer purchase or use SONIX products for any such unintended or unauthorized application. Buyer shall indemnify and hold SONIX and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that SONIX was negligent regarding the design or manufacture of the part.

Main Office:

Address: 10F-1, NO. 36, Taiyuan Street., Chupei City, Hsinchu, Taiwan R.O.C.

Tel: 886-3-5600 888

Fax: 886-3-5600 889

Taipei Office:

Address: 15F-2, NO. 171, Song Ted Road, Taipei, Taiwan R.O.C.

Tel: 886-2-2759 1980

Fax: 886-2-2759 8180

Hong Kong Office:

Unit No.705,Level 7 Tower 1,Grand Central Plaza 138 Shatin Rural Committee Road, Shatin, New Territories, Hong Kong.

Tel: 852-2723-8086

Fax: 852-2723-9179

Technical Support by Email:

Sn8fae@sonix.com.tw