

# **SN32F100 Series**

## **USER'S MANUAL**

**SN32F107**  
**SN32F108**  
**SN32F109**

# **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     | 2013/03/18 | First version released.   |
| 1.1     | 2013/03/29 | <ol style="list-style-type: none"> <li>1. Update Codec Spec.</li> <li>2. Update DAC Setting 3 Register.</li> <li>3. Update Sigma-delta DAC Power-Up Sequence.</li> </ol>  |
| 1.2     | 2013/04/02 | <ol style="list-style-type: none"> <li>1. Update Codec Spec.</li> <li>2. Update ADC Setting 23 Register.</li> <li>3. Update DAC Setting 1 Register and DAC Setting 2 Register.</li> </ol>   |
| 1.3     | 2013/06/04 | <ol style="list-style-type: none"> <li>1. Add SN32F100 Start Kit V1.1 description.</li> <li>2. Add Comparator Output Debounce Time.</li> <li>3. Update supply current.</li> <li>4. Add Operation Mode Comparison Table.</li> <li>5. Update System Block Diagram.</li> <li>6. Update System Tick Timer description.</li> <li>7. Update LQFP 64 Pin Package Information.</li> <li>8. Update Comparator description.</li> <li>9. Modify ADC's SEL_MIC definition.</li> </ol> |
| 1.4     | 2013/07/16 | <ol style="list-style-type: none"> <li>1. Update I2S's Status register default value.</li> <li>2. Update ADC's SEL_MIC register default value.</li> <li>3. Update Code Security diagram.</li> <li>4. Update Code Option Table.</li> <li>5. Update High-level and Low-level input voltage Spec.</li> <li>6. Update P0.14/DPDWAKEUP pin description.</li> </ol>   |

# Table of Content

|   |           |
|---|-----------|
| AMENDMENT HISTORY.....  | 2         |
| <b>1 PRODUCT OVERVIEW.....</b>  | <b>13</b> |
| 1.1 FEATURES.....   | 13        |
| 1.2 SYSTEM BLOCK DIAGRAM.....   | 15        |
| 1.3 CLOCK GENERATION BLOCK DIAGRAM.....                                   | 16        |
| 1.4 PIN ASSIGNMENT.....   | 17        |
| 1.5 PIN DESCRIPTIONS.....   | 20        |
| 1.6 PIN CIRCUIT DIAGRAMS.....   | 25        |
| <b>2 CENTRAL PROCESSOR UNIT (CPU).....</b>                                | <b>27</b> |
| 2.1 MEMORY MAP.....   | 27        |
| 2.2 SYSTEM TICK TIMER.....  | 27        |
| 2.2.1 OPERATION.....  | 28        |
| 2.2.2 SYSTICK USAGE HINTS AND TIPS.....                                   | 28        |
| 2.2.3 SYSTICK REGISTERS.....  | 29        |
| 2.2.3.1 System Tick Timer Control and Status register (SYSTICK_CTRL)..... | 29        |
| 2.2.3.2 System Tick Timer Reload value register (SYSTICK_LOAD).....       | 29        |
| 2.2.3.3 System Tick Timer Current Value register (SYSTICK_VAL).....       | 29        |
| 2.2.3.4 System Tick Timer Calibration Value register (SYST_CALIB).....    | 30        |
| 2.3 NESTED VECTORED INTERRUPT CONTROLLER (NVIC).....                      | 31        |
| 2.3.1 INTERRUPT AND EXCEPTION VECTORS.....                                | 31        |
| 2.3.2 NVIC REGISTERS.....   | 31        |
| 2.3.2.1 IRQ0~31 Interrupt Set-Enable Register (NVIC_ISER).....            | 32        |
| 2.3.2.2 IRQ0~31 Interrupt Clear-Enable Register (NVIC_ICER).....          | 32        |
| 2.3.2.3 IRQ0~31 Interrupt Set-Pending Register (NVIC_ISPR).....           | 32        |
| 2.3.2.4 IRQ0~31 Interrupt Clear-Pending Register (NVIC_ICPR).....         | 32        |
| 2.3.2.5 IRQ0~31 Interrupt Priority Register (NVIC_IPRn) (n=0~7).....      | 33        |
| 2.4 APPLICATION INTERRUPT AND RESET CONTROL (AIRC).....                   | 33        |
| 2.5 CODE OPTION TABLE.....  | 35        |
| 2.6 CORE REGISTER OVERVIEW.....   | 36        |
| <b>3 SYSTEM CONTROL.....</b>  | <b>37</b> |
| 3.1 RESET.....  | 37        |
| 3.1.1 POWER-ON RESET (POR).....   | 37        |
| 3.1.2 WATCHDOG RESET (WDT RESET).....                                     | 38        |
| 3.1.3 BROWN-OUT RESET.....  | 38        |

|         |   |    |
|---------|---|----|
| 3.1.3.1 | BROWN OUT DESCRIPTION .....                                       | 38 |
| 3.1.3.2 | THE SYSTEM OPERATING VOLTAGE DECSRIPTION .....                    | 39 |
| 3.1.3.3 | BROWN-OUT RESET IMPROVEMENT .....                                 | 39 |
| 3.1.4   | <i>EXTERNAL RESET</i> .....                                       | 40 |
| 3.1.4.1 | SIMPLY RC RESET CIRCUIT .....                                     | 41 |
| 3.1.4.2 | DIODE & RC RESET CIRCUIT .....                                    | 41 |
| 3.1.4.3 | ZENER DIODE RESET CIRCUIT .....                                   | 42 |
| 3.1.4.4 | VOLTAGE BIAS RESET CIRCUIT .....                                  | 42 |
| 3.1.4.5 | EXTERNAL RESET IC .....   | 43 |
| 3.1.5   | <i>SOFTWARE RESET</i> .....                                       | 43 |
| 3.2     | SYSTEM CLOCK .....  | 44 |
| 3.2.1   | <i>INTERNAL RC CLOCK SOURCE</i> .....                             | 44 |
| 3.2.1.1 | Internal High-speed RC Oscillator (IHRC) .....                    | 44 |
| 3.2.1.2 | Internal Low-speed RC Oscillator (ILRC) .....                     | 44 |
| 3.2.2   | <i>PLL</i> .....  | 45 |
| 3.2.2.1 | PLL Frequency selection .....                                     | 45 |
| 3.2.3   | <i>EXTERNAL CLOCK SOURCE</i> .....                                | 46 |
| 3.2.3.1 | External High-speed (EHS) Clock .....                             | 46 |
| 3.2.3.2 | CRYSTAL/CERAMIC .....   | 46 |
| 3.2.3.3 | Audio External High-speed (AUEHS) Clock .....                     | 47 |
| 3.2.3.4 | External Low-speed (ELS) Clock .....                              | 47 |
| 3.2.3.5 | CRYSTAL .....   | 47 |
| 3.2.3.6 | 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> .....             | 52 |
| 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> .....            | 53 |
| 3.3.7   | <i>LVD Control register (SYS0_LVDCTRL)</i> .....                  | 53 |
| 3.3.8   | <i>External RESET Pin Control register (SYS0_EXRSTCTRL)</i> ..... | 55 |
| 3.3.9   | <i>SWD Pin Control register (SYS0_SWDCtrl)</i> .....              | 55 |
| 3.3.10  | <i>Anti-EFT Ability Control register (SYS0_ANTIEFT)</i> .....     | 55 |
| 3.4     | SYSTEM CONTROL REGISTERS 1 .....                                  | 56 |
| 3.4.1   | <i>AHB Clock Enable register (SYS1_AHBCLKEN)</i> .....            | 56 |
| 3.4.2   | <i>APB Clock Prescale register 0 (SYS1_APBPCP0)</i> .....         | 57 |

|          |  |           |
|----------|--|-----------|
| 3.4.3    | APB Clock Prescale register 1 (SYSI_APBPCP1).....  | 58        |
| 3.4.4    | Peripheral Reset register (SYSI_PRST).....   | 59        |
| <b>4</b> | <b>SYSTEM OPERATION MODE .....</b>   | <b>61</b> |
| 4.1      | OVERVIEW .....   | 61        |
| 4.2      | NORMAL MODE .....  | 61        |
| 4.3      | LOW-POWER MODES .....  | 61        |
| 4.3.1    | SLEEP MODE .....   | 61        |
| 4.3.2    | DEEP-SLEEP MODE.....   | 62        |
| 4.3.3    | DEEP POWER-DOWN (DPD) MODE.....  | 62        |
| 4.3.3.1  | Entering Deep power-down mode.....   | 63        |
| 4.3.3.2  | Exiting Deep power-down mode.....  | 63        |
| 4.4      | WAKEUP INTERRUPT .....   | 63        |
| 4.5      | STATE MACHINE OF PMU .....   | 63        |
| 4.6      | OPERATION MODE COMPARSION TABLE .....  | 64        |
| 4.7      | PMU REGISTERS .....  | 65        |
| 4.7.1    | Backup registers 0 to 15 (PMU_BKP0~15).....  | 65        |
| 4.7.2    | Power control register (PMU_CTRL).....   | 65        |
| <b>5</b> | <b>GENERAL PURPOSE I/O PORT (GPIO) .....</b>   | <b>66</b> |
| 5.1      | OVERVIEW .....   | 66        |
| 5.2      | GPIO MODE .....  | 66        |
| 5.3      | GPIO REGISTERS.....  | 67        |
| 5.3.1    | GPIO Port n Data register (GPIO <sub>n</sub> _DATA) (n=0,1,2,3).....                     | 67        |
| 5.3.2    | GPIO Port n Mode register (GPIO <sub>n</sub> _MODE) (n=0,1,2,3).....                     | 67        |
| 5.3.3    | GPIO Port n Configuration register (GPIO <sub>n</sub> _CFG) (n=0,1,2,3).....             | 67        |
| 5.3.4    | GPIO Port n Interrupt Sense register (GPIO <sub>n</sub> _IS) (n=0,1,2,3).....            | 69        |
| 5.3.5    | GPIO Port n Interrupt Both-edge Sense register (GPIO <sub>n</sub> _IBS) (n=0,1,2,3)..... | 69        |
| 5.3.6    | GPIO Port n Interrupt Event register (GPIO <sub>n</sub> _IEV) (n=0,1,2,3).....           | 69        |
| 5.3.7    | GPIO Port n Interrupt Enable register (GPIO <sub>n</sub> _IE) (n=0,1,2,3).....           | 69        |
| 5.3.8    | GPIO Port n Raw Interrupt Status register (GPIO <sub>n</sub> _RIS) (n=0,1,2,3) .....     | 70        |
| 5.3.9    | GPIO Port n Interrupt Clear register (GPIO <sub>n</sub> _IC) (n=0,1,2,3).....            | 70        |
| 5.3.10   | GPIO Port n Bits Set Operation register (GPIO <sub>n</sub> _BSET) (n=0,1,2,3).....       | 70        |
| 5.3.11   | GPIO Port n Bits Clear Operation register (GPIO <sub>n</sub> _BCLR) (n=0,1,2,3).....     | 70        |
| 5.3.12   | GPIO Port n Open-Drain Control register (GPIO <sub>n</sub> _ODCTRL) (n=0,1,2,3) .....    | 70        |
| <b>6</b> | <b>16-BIT TIMER WITH CAPTURE FUNCTION .....</b>  | <b>72</b> |
| 6.1      | OVERVIEW .....   | 72        |
| 6.2      | FEATURES .....   | 72        |
| 6.3      | PIN DESCRIPTION .....  | 72        |
| 6.4      | BLOCK DIAGRAM.....   | 73        |

|          |   |           |
|----------|---|-----------|
| 6.5      | TIMER OPERATION .....   | 74        |
| 6.6      | PWM.....  | 75        |
| 6.7      | CT16BN REGISTERS .....  | 76        |
| 6.7.1    | CT16Bn Timer Control register (CT16Bn_TMRCTRL) (n=0,1) .....          | 76        |
| 6.7.2    | CT16Bn Timer Counter register (CT16Bn_TC) (n=0,1) .....               | 76        |
| 6.7.3    | CT16Bn Prescale register (CT16Bn_PRE) (n=0,1).....                    | 76        |
| 6.7.4    | CT16Bn Prescale Counter register (CT16Bn_PC) (n=0,1).....             | 76        |
| 6.7.5    | CT16Bn Count Control register (CT16Bn_CNTCTRL) (n=0,1).....           | 77        |
| 6.7.6    | CT16Bn Match Control register (CT16Bn_MCTRL) (n=0,1) .....            | 77        |
| 6.7.7    | CT16Bn Match register 0~3 (CT16Bn_MR0~3) (n=0,1).....                 | 78        |
| 6.7.8    | CT16Bn Capture Control register (CT16Bn_CAPCTRL) (n=0,1) .....        | 78        |
| 6.7.9    | CT16Bn Capture 0 register (CT16Bn_CAP0) (n=0,1) .....                 | 79        |
| 6.7.10   | CT16Bn External Match register (CT16Bn_EM) (n=0,1) .....              | 79        |
| 6.7.11   | CT16Bn PWM Control register (CT16Bn_PWMCTRL) (n=0,1).....             | 79        |
| 6.7.12   | CT16Bn Timer Raw Interrupt Status register (CT16Bn_RIS) (n=0,1) ..... | 80        |
| 6.7.13   | CT16Bn Timer Interrupt Clear register (CT16Bn_IC) (n=0,1).....        | 80        |
| <b>7</b> | <b>32-BIT TIMER WITH CAPTURE FUNCTION .....</b>                       | <b>81</b> |
| 7.1      | OVERVIEW .....  | 81        |
| 7.2      | FEATURES .....  | 81        |
| 7.3      | PIN DESCRIPTION .....   | 81        |
| 7.4      | BLOCK DIAGRAM.....  | 82        |
| 7.5      | TIMER OPERATION .....   | 83        |
| 7.6      | PWM.....  | 84        |
| 7.7      | CT32BN REGISTERS .....  | 85        |
| 7.7.1    | CT32Bn Timer Control register (CT32Bn_TMRCTRL) (n=0,1) .....          | 85        |
| 7.7.2    | CT32Bn Timer Counter register (CT32Bn_TC) (n=0,1) .....               | 85        |
| 7.7.3    | CT32Bn Prescale register (CT32Bn_PRE) (n=0,1).....                    | 85        |
| 7.7.4    | CT32Bn Prescale Counter register (CT32Bn_PC) (n=0,1).....             | 85        |
| 7.7.5    | CT32Bn Count Control register (CT32Bn_CNTCTRL) (n=0,1).....           | 85        |
| 7.7.6    | CT32Bn Match Control register (CT32Bn_MCTRL) (n=0,1) .....            | 86        |
| 7.7.7    | CT32Bn Match register 0~3 (CT32Bn_MR0~3) (n=0,1).....                 | 87        |
| 7.7.8    | CT32Bn Capture Control register (CT32Bn_CAPCTRL) (n=0,1) .....        | 87        |
| 7.7.9    | CT32Bn Capture 0 register (CT32Bn_CAP0) (n=0,1) .....                 | 88        |
| 7.7.10   | CT32Bn External Match register (CT32Bn_EM) (n=0,1) .....              | 88        |
| 7.7.11   | CT32Bn PWM Control register (CT32Bn_PWMCTRL) (n=0,1).....             | 88        |
| 7.7.12   | CT32Bn Timer Raw Interrupt Status register (CT32Bn_RIS) (n=0,1) ..... | 89        |
| 7.7.13   | CT32Bn Timer Interrupt Clear register (CT32Bn_IC) (n=0,1).....        | 89        |
| <b>8</b> | <b>WATCHDOG TIMER (WDT).....</b>                                      | <b>90</b> |

|           |   |            |
|-----------|---|------------|
| 8.1       | OVERVIEW .....  | 90         |
| 8.2       | BLOCK DIAGRAM.....  | 91         |
| 8.3       | WDT REGISTERS .....   | 92         |
| 8.3.1     | <i>Watchdog Configuration register (WDT_CFG)</i> .....              | 92         |
| 8.3.2     | <i>Watchdog Clock Source register (WDT_CLKSOURCE)</i> .....         | 92         |
| 8.3.3     | <i>Watchdog Timer Constant register (WDT_TC)</i> .....              | 92         |
| 8.3.4     | <i>Watchdog Feed register (WDT_FEED)</i> .....                      | 93         |
| <b>9</b>  | <b>REAL-TIME CLOCK (RTC) .....</b>                                  | <b>94</b>  |
| 9.1       | OVERVIEW .....  | 94         |
| 9.2       | FEATURES .....  | 94         |
| 9.3       | FUNCTIONAL DESCRIPTION .....  | 94         |
| 9.3.1     | <i>INTRODUCTION</i> .....   | 94         |
| 9.3.2     | <i>RESET RTC REGISTERS</i> .....                                    | 94         |
| 9.3.3     | <i>RTC FLAG ASSERTION</i> .....                                     | 94         |
| 9.3.4     | <i>RTC OPERATION</i> .....  | 95         |
| 9.4       | BLOCK DIAGRAM.....  | 96         |
| 9.5       | RTC REGISTERS .....   | 97         |
| 9.5.1     | <i>RTC Control register (RTC_CTRL)</i> .....                        | 97         |
| 9.5.2     | <i>RTC Clock Source Select register (RTC_CLKS)</i> .....            | 97         |
| 9.5.3     | <i>RTC Interrupt Enable register (RTC_IE)</i> .....                 | 97         |
| 9.5.4     | <i>RTC Raw Interrupt Status register (RTC_RIS)</i> .....            | 97         |
| 9.5.5     | <i>RTC Interrupt Clear register (RTC_IC)</i> .....                  | 98         |
| 9.5.6     | <i>RTC Second Counter Reload Value register (RTC_SECCNTV)</i> ..... | 98         |
| 9.5.7     | <i>RTC Second Count register (RTC_SECCNT)</i> .....                 | 98         |
| 9.5.8     | <i>RTC Alarm Counter Reload Value register (RTC_ALMCNTV)</i> .....  | 98         |
| 9.5.9     | <i>RTC Alarm Count register (RTC_ALMCNT)</i> .....                  | 99         |
| <b>10</b> | <b>SPI/SSP .....</b>  | <b>100</b> |
| 10.1      | OVERVIEW .....  | 100        |
| 10.2      | FEATURES .....  | 100        |
| 10.3      | PIN DESCRIPTION .....   | 101        |
| 10.4      | INTERFACE DESCRIPTION .....   | 102        |
| 10.4.1    | <i>SPI</i> .....  | 102        |
| 10.4.2    | <i>SSI</i> .....  | 103        |
| 10.4.3    | <i>COMMUNICATION FLOW</i> .....                                     | 103        |
| 10.4.3.1  | <i>SINGLE-FRAME</i> .....   | 103        |
| 10.4.3.2  | <i>MULTI-FRAME</i> .....  | 104        |
| 10.5      | AUTO-SEL (AUTO-CS) .....  | 104        |
| 10.6      | SSP REGISTERS .....   | 105        |

|           |   |            |
|-----------|---|------------|
| 10.6.1    | SSP n Control register 0 (SSPn_CTRL0) (n=0, 1).....                       | 105        |
| 10.6.2    | SSP n Control register 1 (SSPn_CTRL1) (n=0, 1).....                       | 106        |
| 10.6.3    | SSP n Clock Divider register (SSPn_CLKDIV) (n=0, 1).....                  | 106        |
| 10.6.4    | SSP n Status register (SSPn_STAT) (n=0, 1).....                           | 106        |
| 10.6.5    | SSP n Interrupt Enable register (SSPn_IE) (n=0, 1).....                   | 107        |
| 10.6.6    | SSP n Raw Interrupt Status register (SSPn_RIS) (n=0, 1).....              | 107        |
| 10.6.7    | SSP n Interrupt Clear register (SSPn_IC) (n=0, 1).....                    | 107        |
| 10.6.8    | SSP n Data register (SSPn_DATA) (n=0, 1).....                             | 108        |
| <b>11</b> | <b>I2C.....</b>   | <b>109</b> |
| 11.1      | OVERVIEW.....   | 109        |
| 11.2      | FEATURES.....   | 109        |
| 11.3      | PIN DESCRIPTION.....  | 110        |
| 11.4      | WAVE CHARACTERISTICS.....   | 110        |
| 11.5      | I2C MASTER MODES.....   | 111        |
| 11.5.1    | MASTER TRANSMITTER MODE.....  | 111        |
| 11.5.2    | MASTER RECEIVER MODE.....   | 111        |
| 11.5.3    | ARBITRATION.....  | 111        |
| 11.6      | I2C SLAVE MODES.....  | 112        |
| 11.6.1    | SLAVE TRANSMITTER MODE.....   | 112        |
| 11.6.2    | SLAVE RECEIVER MODE.....  | 112        |
| 11.7      | MONITOR MODE.....   | 113        |
| 11.7.1    | INTERRUPT.....  | 113        |
| 11.7.2    | LOSS of ARBITRATION.....  | 113        |
| 11.8      | I2C REGISTERS.....  | 114        |
| 11.8.1    | I2C n Control register (I2Cn_CTRL) (n=0,1).....                           | 114        |
| 11.8.2    | I2C n Status register (I2Cn_STAT) (n=0,1).....                            | 115        |
| 11.8.3    | I2C n TX Data register (I2Cn_TXDATA) (n=0,1).....                         | 115        |
| 11.8.4    | I2C n RX Data register (I2Cn_RXDATA) (n=0,1).....                         | 116        |
| 11.8.5    | I2C n Slave Address 0 register (I2Cn_SLVADDR0) (n=0,1).....               | 116        |
| 11.8.6    | I2C n Slave Address 1~3 register (I2Cn_SLVADDR1~3) (n=0,1).....           | 116        |
| 11.8.7    | I2C n SCL High Time register (I2Cn_SCLHT) (n=0,1).....                    | 116        |
| 11.8.8    | I2C n SCL Low Time register (I2Cn_SCLLT) (n=0,1).....                     | 117        |
| 11.8.9    | I2C n Timeout Control register (I2Cn_TOCTRL) (n=0,1).....                 | 117        |
| 11.8.10   | I2C n Monitor Mode Control register (I2Cn_MMCTRL) (n=0,1).....            | 117        |
| <b>12</b> | <b>UNIVERSAL ASYNCHRONOUS SERIAL RECEIVER AND TRANSMITTER (UART) ....</b> | <b>118</b> |
| 12.1      | OVERVIEW.....   | 118        |
| 12.2      | FEATURES.....   | 118        |
| 12.3      | PIN DESCRIPTION.....  | 118        |



|           |   |            |
|-----------|---|------------|
| 12.4      | BLOCK DIAGRAM.....  | 119        |
| 12.5      | BAUD RATE CALCULATION .....                                       | 120        |
| 12.6      | AUTO-BAUD FLOW .....  | 121        |
| 12.6.1    | AUTO-BAUD.....  | 121        |
| 12.6.2    | AUTO-BAUD MODES.....  | 122        |
| 12.7      | UART REGISTERS .....  | 124        |
| 12.7.1    | UART n Receiver Buffer register (UARTn_RB) (n=0, 1).....          | 124        |
| 12.7.2    | UART n Transmitter Holding register (UARTn_TH) (n=0, 1).....      | 124        |
| 12.7.3    | UART n Divisor Latch LSB registers (UARTn_DLL) (n =0, 1).....     | 124        |
| 12.7.4    | UART n Divisor Latch MSB register (UARTn_DLM) (n=0,1) .....       | 124        |
| 12.7.5    | UART n Interrupt Enable register (UARTn_IE) (n=0, 1) .....        | 125        |
| 12.7.6    | UART n Interrupt Identification register (UARTn_II) (n=0,1) ..... | 125        |
| 12.7.7    | UART n FIFO Control register (UARTn_FIFOCTRL) (n=0,1).....        | 127        |
| 12.7.8    | UART n Line Control register (UARTn_LC) (n=0,1) .....             | 127        |
| 12.7.9    | UART n Line Status register (UARTn_LS) (n=0,1).....               | 127        |
| 12.7.10   | UART n Scratch Pad register (UARTn_SP) (n=0, 1).....              | 129        |
| 12.7.11   | UART n Auto-baud Control register (UARTn_ABCTRL) (n=0, 1).....    | 129        |
| 12.7.12   | UART n Fractional Divider register (UARTn_FD) (n=0, 1).....       | 129        |
| 12.7.13   | UART n Control register (UARTn_CTRL) (n=0, 1).....                | 130        |
| 12.7.14   | UART n Half-duplex Enable register (UARTn_HDEN) (n=0, 1) .....    | 130        |
| <b>13</b> | <b>AUDIO (I2S/CODEC).....</b>                                     | <b>132</b> |
| 13.1      | OVERVIEW .....  | 132        |
| 13.1.1    | I2S Description.....  | 132        |
| 13.1.2    | Codec Description .....   | 132        |
| 13.2      | FEATURES .....  | 132        |
| 13.2.1    | I2S Features.....   | 132        |
| 13.2.2    | Codec Features.....   | 132        |
| 13.3      | PIN DESCRIPTION .....   | 133        |
| 13.3.1    | I2S Pin Description .....   | 133        |
| 13.3.2    | Codec Pin Description.....  | 133        |
| 13.3.3    | Audio Clock Pin Description.....                                  | 133        |
| 13.4      | BLOCK DIAGRAM.....  | 134        |
| 13.4.1    | I2S CLCOK CONTROL.....  | 134        |
| 13.4.2    | I2S BLOCK DIAGRAM .....   | 134        |
| 13.4.3    | 16-Bit Sigma-Delta ADC BLOCK DIAGRAM .....                        | 135        |
| 13.4.4    | 16-Bit Sigma-Delta DAC BLOCK DIAGRAM .....                        | 136        |
| 13.5      | FUNCTIONAL DESCRIPTION .....                                      | 137        |
| 13.5.1    | I2S OPERATION .....   | 137        |
| 13.5.2    | I2S FIFO OPERAION .....   | 139        |

|          |  |     |
|----------|--|-----|
| 13.5.2.1 | MONO .....   | 139 |
| 13.5.2.2 | STEREO .....   | 139 |
| 13.6     | I2S REGISTERS.....                                       | 140 |
| 13.6.1   | <i>I2S Control register (I2S_CTRL)</i> .....             | 140 |
| 13.6.2   | <i>I2S Clock register (I2S_CLK)</i> .....                | 141 |
| 13.6.3   | <i>I2S Status register (I2S_STATUS)</i> .....            | 141 |
| 13.6.4   | <i>I2S Interrupt Enable register (I2S_IE)</i> .....      | 142 |
| 13.6.5   | <i>I2S Raw Interrupt Status register (I2S_RIS)</i> ..... | 142 |
| 13.6.6   | <i>I2S Interrupt Clear register (I2S_IC)</i> .....       | 143 |
| 13.6.7   | <i>I2S RX FIFO register (I2S_RXFIFO)</i> .....           | 143 |
| 13.6.8   | <i>I2S TX FIFO register (I2S_TXFIFO)</i> .....           | 143 |
| 13.7     | CODEC ADC REGISTERS .....                                | 143 |
| 13.7.1   | <i>ADC Setting 1 register (ADC_SET1)</i> .....           | 143 |
| 13.7.2   | <i>ADC Setting 2 register (ADC_SET2)</i> .....           | 144 |
| 13.7.3   | <i>ADC Setting 3 register (ADC_SET3)</i> .....           | 144 |
| 13.7.4   | <i>ADC Setting 4 register (ADC_SET4)</i> .....           | 144 |
| 13.7.5   | <i>ADC Setting 5 register (ADC_SET5)</i> .....           | 144 |
| 13.7.6   | <i>ADC Setting 6 register (ADC_SET6)</i> .....           | 144 |
| 13.7.7   | <i>ADC Setting 7 register (ADC_SET7)</i> .....           | 144 |
| 13.7.8   | <i>ADC Setting 8 register (ADC_SET8)</i> .....           | 145 |
| 13.7.9   | <i>ADC Setting 9 register (ADC_SET9)</i> .....           | 145 |
| 13.7.10  | <i>ADC Setting 10 register (ADC_SET10)</i> .....         | 145 |
| 13.7.11  | <i>ADC Setting 11 register (ADC_SET11)</i> .....         | 145 |
| 13.7.12  | <i>ADC Setting 12 register (ADC_SET12)</i> .....         | 145 |
| 13.7.13  | <i>ADC Setting 13 register (ADC_SET13)</i> .....         | 146 |
| 13.7.14  | <i>ADC Setting 14 register (ADC_SET14)</i> .....         | 146 |
| 13.7.15  | <i>ADC Setting 15 register (ADC_SET15)</i> .....         | 146 |
| 13.7.16  | <i>ADC Setting 16 register (ADC_SET16)</i> .....         | 146 |
| 13.7.17  | <i>ADC Setting 18 register (ADC_SET18)</i> .....         | 147 |
| 13.7.18  | <i>ADC Setting 19 register (ADC_SET19)</i> .....         | 147 |
| 13.7.19  | <i>ADC Setting 20 register (ADC_SET20)</i> .....         | 148 |
| 13.7.20  | <i>ADC Setting 21 register (ADC_SET21)</i> .....         | 148 |
| 13.7.21  | <i>ADC Setting 22 register (ADC_SET22)</i> .....         | 148 |
| 13.7.22  | <i>ADC Setting 23 register (ADC_SET23)</i> .....         | 148 |
| 13.7.23  | <i>ADC Setting 24 register (ADC_SET24)</i> .....         | 149 |
| 13.8     | CODEC DAC REGISTERS .....                                | 149 |
| 13.8.1   | <i>DAC Setting 1 register (DAC_SET1)</i> .....           | 149 |
| 13.8.2   | <i>DAC Setting 2 register (DAC_SET2)</i> .....           | 149 |
| 13.8.3   | <i>DAC Setting 3 register (DAC_SET3)</i> .....           | 149 |
| 13.8.4   | <i>DAC Setting 4 register (DAC_SET4)</i> .....           | 150 |

|           |   |            |
|-----------|---|------------|
| 13.8.5    | DAC Status register (DAC_STATUS).....               | 150        |
| 13.9      | SIGMA-DELTA ADC CONTROL FLOW .....                  | 150        |
| 13.9.1    | Sigma-delta ADC Power-up Sequence .....             | 150        |
| 13.9.2    | Sigma-delta ADC Power-down Sequence .....           | 150        |
| 13.9.3    | Sigma-delta ADC Enable Sequence .....               | 151        |
| 13.10     | SIGMA-DELTA DAC CONTROL FLOW .....                  | 151        |
| 13.10.1   | Sigma-delta DAC Power-up Sequence .....             | 151        |
| 13.10.2   | Sigma-delta DAC Power-down Sequence .....           | 151        |
| 13.10.3   | Sigma-delta DAC Enable Sequence .....               | 151        |
| <b>14</b> | <b>24-CHANNEL COMPARATOR .....</b>                  | <b>152</b> |
| 14.1      | OVERVIEW .....                                      | 152        |
| 14.2      | COMPARATOR OPERATION .....                          | 153        |
| 14.3      | COMPARATOR APPLICATION NOTICE .....                 | 154        |
| 14.4      | COMPARATOR CONTROL REGISTERS.....                   | 154        |
| 14.4.1    | Comparator Control register (CMPM).....             | 154        |
| 14.4.2    | Comparator Interrupt Enable register (CMP_IE) ..... | 155        |
| 14.4.3    | Comparator Interrupt Status register (CMP_RIS)..... | 156        |
| 14.4.4    | Comparator Interrupt Clear register (CMP_IC) .....  | 156        |
| <b>15</b> | <b>FLASH.....</b>                                   | <b>157</b> |
| 15.1      | OVERVIEW .....                                      | 157        |
| 15.2      | EMBEDDED FLASH MEMORY .....                         | 157        |
| 15.3      | FEATURES .....                                      | 157        |
| 15.4      | ORGANIZATION.....                                   | 158        |
| 15.5      | READ .....  | 158        |
| 15.6      | PROGRAM/ERASE.....                                  | 158        |
| 15.7      | EMBEDDED BOOT LOADER .....                          | 158        |
| 15.8      | FLASH MEMORY CONTROLLER (FMC).....                  | 159        |
| 15.8.1    | CODE SECURITY (CS).....                             | 159        |
| 15.8.2    | PROGRAM FLASH MEMORY.....                           | 160        |
| 15.8.3    | ERASE .....   | 160        |
| 15.8.3.1  | PAGE ERASE .....                                    | 160        |
| 15.8.3.2  | MASS ERASE.....                                     | 160        |
| 15.9      | READ PROTECTION.....                                | 160        |
| 15.10     | FMC REGISTERS.....                                  | 161        |
| 15.10.1   | Flash Status register (FLASH_STATUS) .....          | 161        |
| 15.10.2   | Flash Control register (FLASH_CTRL).....            | 161        |
| 15.10.3   | Flash Data register (FLASH_DATA).....               | 161        |
| 15.10.4   | Flash Address register (FLASH_ADDR) .....           | 162        |

|           |  |            |
|-----------|--|------------|
| <b>16</b> | <b>SERIAL-WIRE DEBUG (SWD)</b> .....                     | <b>163</b> |
| 16.1      | OVERVIEW .....   | 163        |
| 16.2      | FEATURES .....   | 163        |
| 16.3      | PIN DESCRIPTION .....                                    | 163        |
| 16.4      | DEBUG NOTE .....   | 163        |
| 16.4.1    | <i>LIMITATIONS</i> .....                                 | 163        |
| 16.4.2    | <i>DEBUG RECOVERY</i> .....                              | 163        |
| 16.4.3    | <i>INTERNAL PULL-UP/DOWN RESISTORS on SWD PINS</i> ..... | 163        |
| <b>17</b> | <b>DEVELOPMENT TOOL</b> .....                            | <b>164</b> |
| 17.1      | SN-LINK .....  | 164        |
| 17.2      | SN32F100 STARTER-KIT .....                               | 165        |
| 17.2.1    | <i>SN32F100 Start Kit V1.0</i> .....                     | 165        |
| 17.2.2    | <i>SN32F100 Start Kit V1.1/V1.2</i> .....                | 167        |
| <b>18</b> | <b>ELECTRICAL CHARACTERISTIC</b> .....                   | <b>169</b> |
| 18.1      | ABSOLUTE MAXIMUM RATING .....                            | 169        |
| 18.2      | ELECTRICAL CHARACTERISTIC .....                          | 169        |
| 18.3      | CHARACTERISTIC GRAPHS .....                              | 171        |
| <b>19</b> | <b>FLASH ROM PROGRAMMING PIN</b> .....                   | <b>172</b> |
| <b>20</b> | <b>PACKAGE INFORMATION</b> .....                         | <b>173</b> |
| 20.1      | LQFP 48 PIN .....  | 173        |
| 20.2      | LQFP 64 PIN .....  | 174        |
| 20.3      | LQFP 80 PIN .....  | 175        |
| <b>21</b> | <b>MARKING DEFINITION</b> .....                          | <b>176</b> |
| 21.1      | INTRODUCTION .....                                       | 176        |
| 21.2      | MARKING IDENTIFICATION SYSTEM .....                      | 176        |
| 21.3      | MARKING EXAMPLE .....                                    | 177        |
| 21.4      | DATECODE SYSTEM .....                                    | 177        |

# 1 PRODUCT OVERVIEW

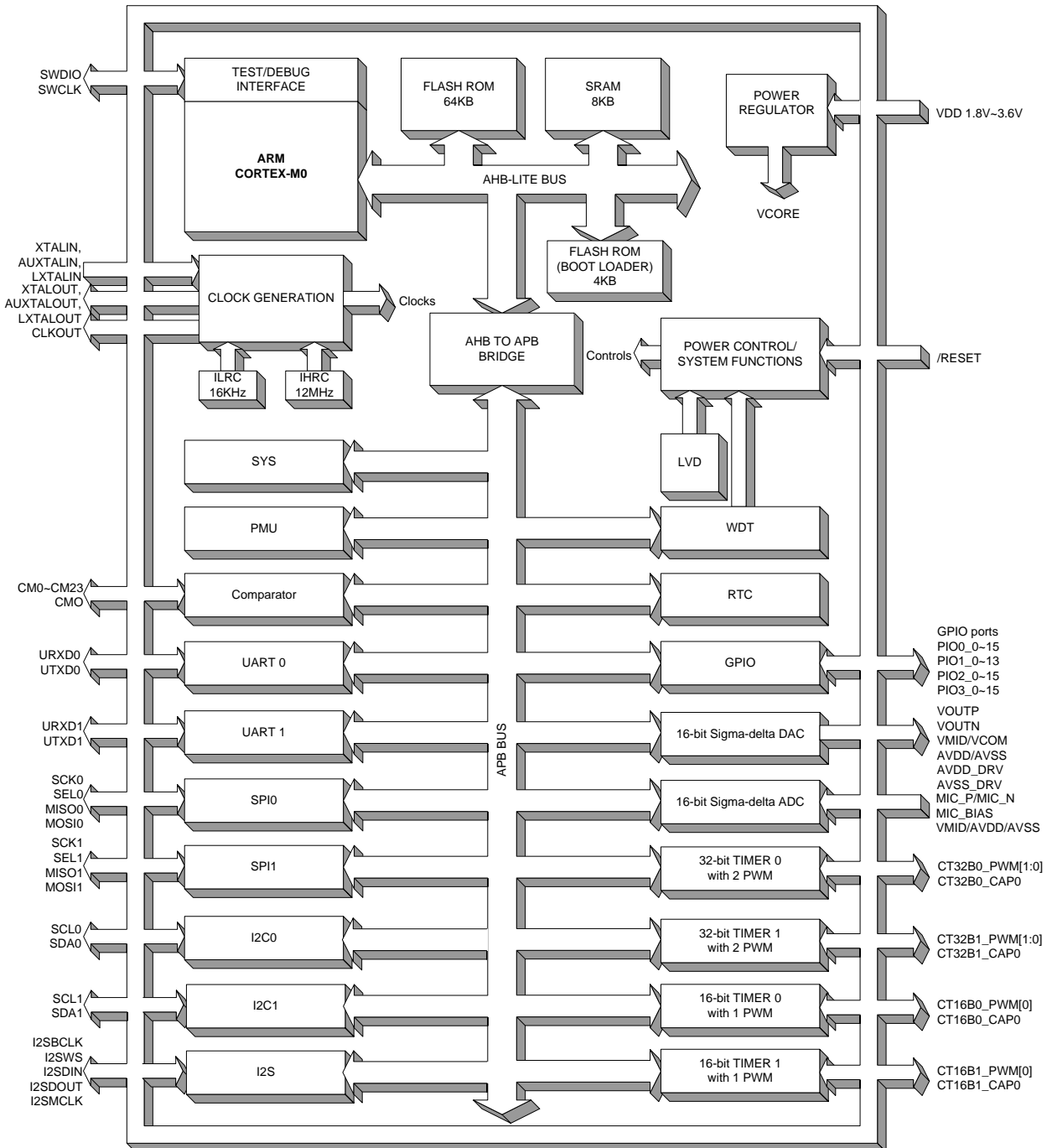
## 1.1 FEATURES

- ◆ **Memory configuration**  
64KB 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 62 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.  
Comparator input pin : CM0~CM23.  
Comparator output pin : CMO.
- ◆ **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<sub>CORE</sub> 1.8V, 2.0/2.4/2.7V for VDD  
Interrupt: 2.0/2.4/2.7/3.0V for VDD
- ◆ **Fcpu (Instruction cycle)**  
 $F_{CPU} = F_{HCLK} = F_{SYSCLK}/1, F_{SYSCLK}/2, F_{SYSCLK}/4, \dots, F_{SYSCLK}/512.$
- ◆ **Working voltage 1.8V ~ 3.6V**
- ◆ **Operating modes**  
Normal, Sleep, Deep-sleep, and Deep power-down
- ◆ **Serial Wire Debug (SWD)**
- ◆ **In-System Programming (ISP) supported**
- ◆ **Timer**  
Two 16-bit and two 32-bit general purpose timers with a total of four capture inputs, 6PWMS
- ◆ **DAC**  
16-bit Sigma-delta DAC for Audio.  
Can drive the L/R Channel Earphone.  
SNR 90dB.  
THD+N -75dB.
- ◆ **ADC**  
16-bit Sigma-delta ADC for Audio.  
AGC function.  
Differential Microphone input.  
Build-in Microphone Bias Voltage support.  
SNR 94dB.  
THD+N -80dB.
- ◆ **24-channel Comparator.**
- ◆ **Interface**  
-Two I2C controllers supporting I2C-bus specification with multiple address recognition and monitor mode.  
-Two UART controllers with fractional baud rate generation.  
-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 Audio high clock: Crystal type 16.384MHz  
-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.
- ◆ **Package (Chip form support)**  
LQFP 80 pin  
LQFP 64 pin  
LQFP 48 pin

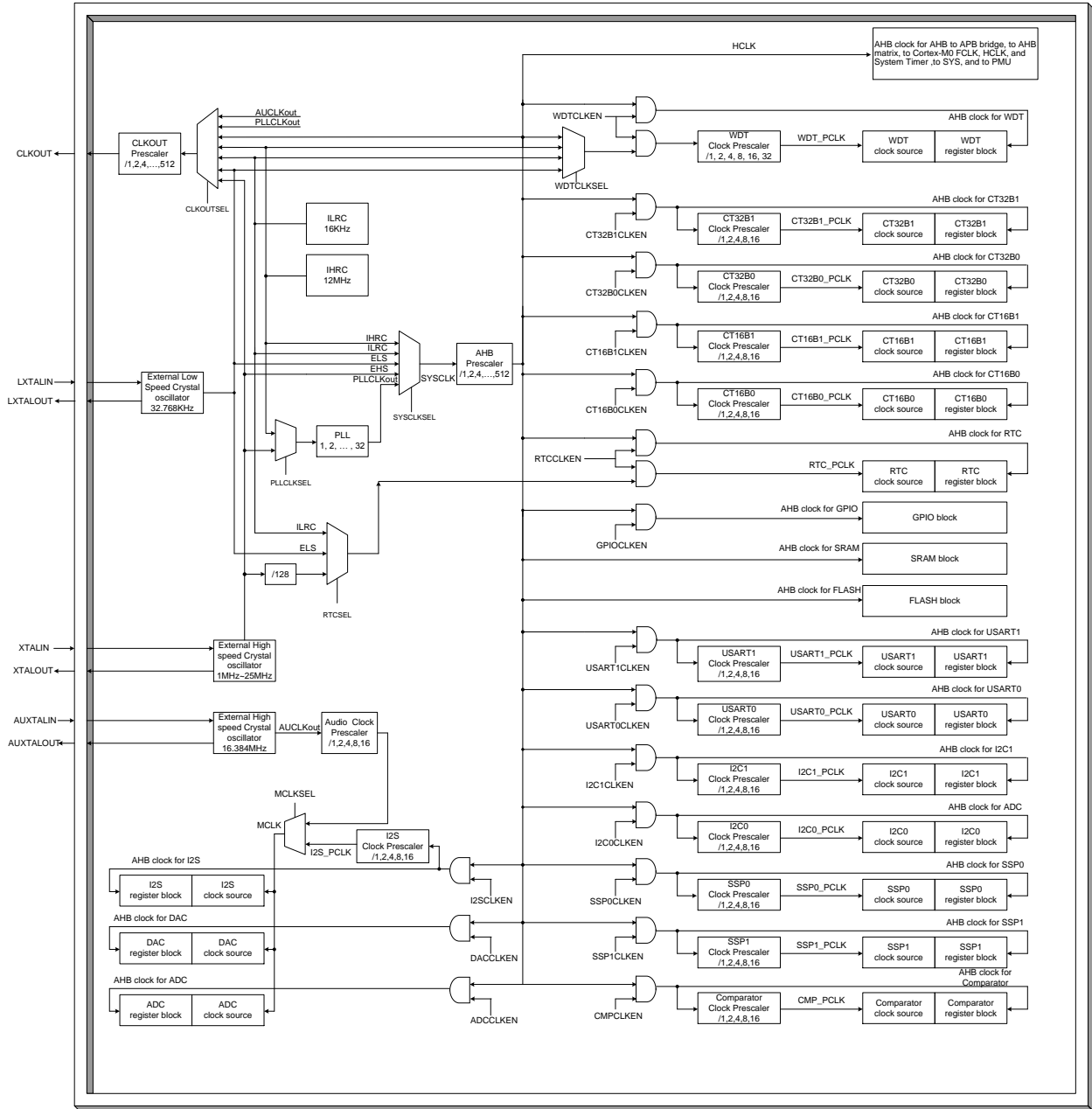
**Features Selection Table**

| Chip      | ROM  | RAM | Boot Loader | Operating Freq. (Max) | UART | SPI | I2C | I2S | TIMER                    | PWM | 16-bit Sigma-delta ADC | 16-bit Sigma-delta DAC | CMP | GPIO with Wakeup | Package |
|-----------|------|-----|-------------|-----------------------|------|-----|-----|-----|--------------------------|-----|------------------------|------------------------|-----|------------------|---------|
| SN32F107F | 64KB | 8KB | 4KB         | 50 MHz                | 1    | 1   | 2   | -   | 16-bit x 2<br>32-bit x 2 | 4   | 1                      | 1                      | 8   | 32               | LQFP48  |
| SN32F108F | 64KB | 8KB | 4KB         | 50 MHz                | 2    | 1   | 2   | -   | 16-bit x 2<br>32-bit x 2 | 6   | 1                      | 1                      | 17  | 46               | LQFP64  |
| SN32F109F | 64KB | 8KB | 4KB         | 50 MHz                | 2    | 2   | 2   | 1   | 16-bit x 2<br>32-bit x 2 | 6   | 1                      | 1                      | 24  | 62               | LQFP80  |

## 1.2 SYSTEM BLOCK DIAGRAM



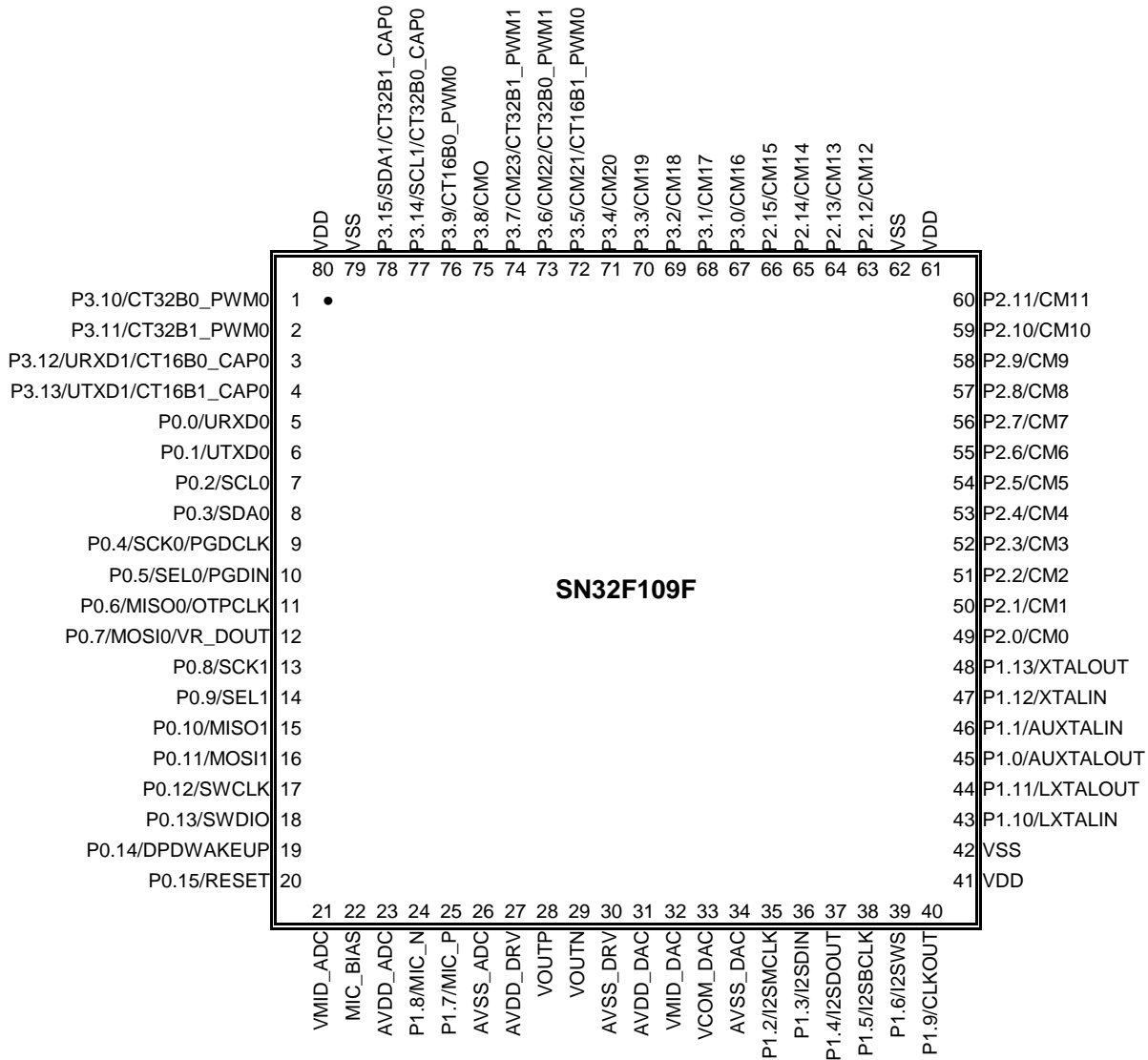
# 1.3 CLOCK GENERATION BLOCK DIAGRAM



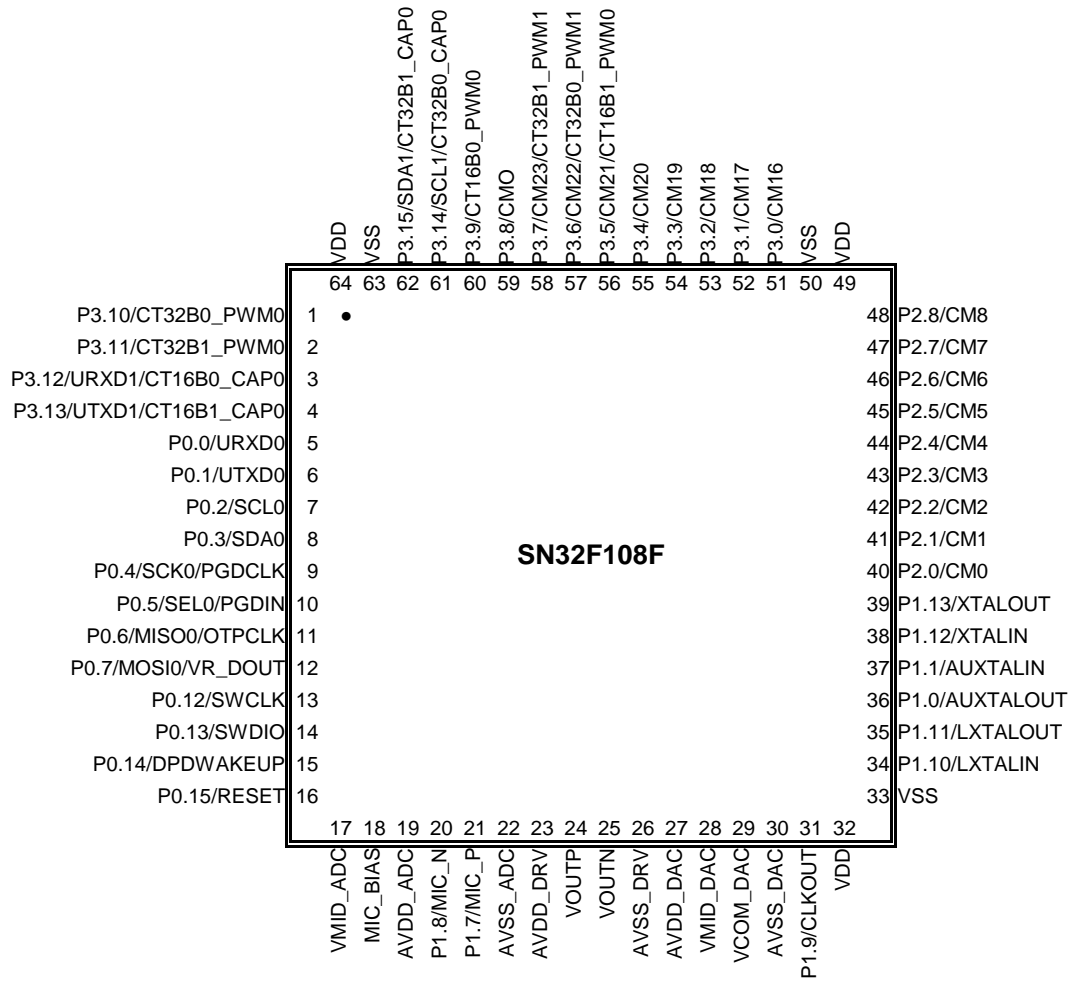


## 1.4 PIN ASSIGNMENT

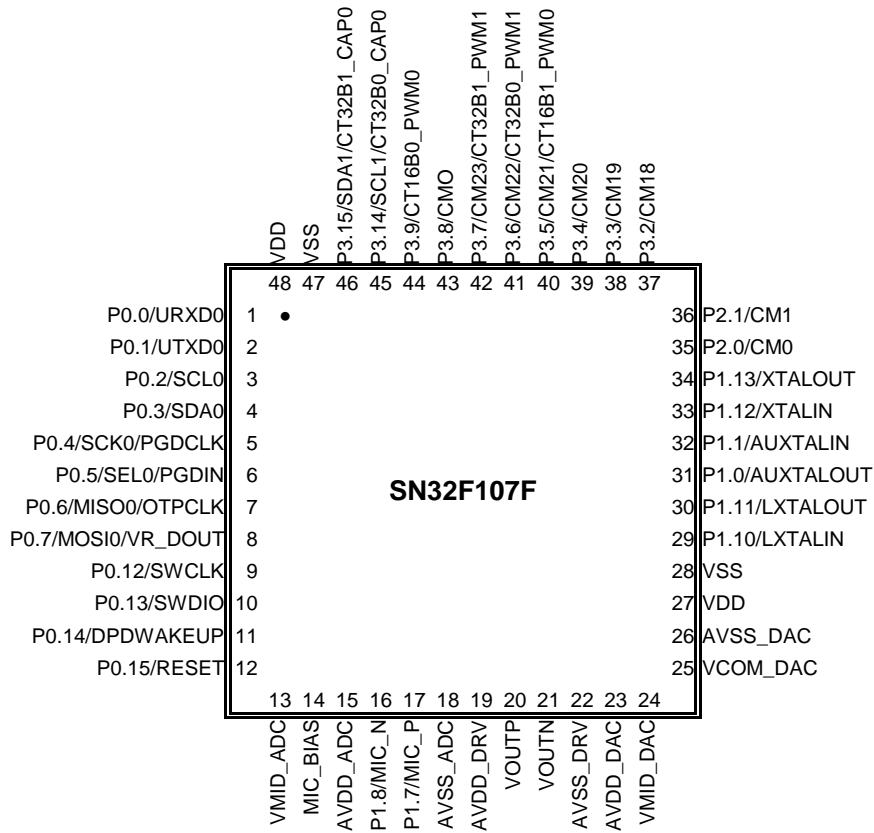
### SN32F109F (LQFP 80 pins)



**SN32F108F (LQFP 64 pins)**



**SN32F107F (LQFP 48 pins)**



## 1.5 PIN DESCRIPTIONS

| PIN NAME           | TYPE | DESCRIPTION   |
|--------------------|------|---|
| VDD, VSS           | P    | Power supply input pins for digital circuit.  |
| AVDD_DAC, AVSS_DAC | P    | Power supply input pins for Sigma-delta DAC.  |
| AVDD_ADC, AVSS_ADC | P    | Power supply input pins for Sigma-delta ADC.  |
| AVDD_DRV, AVSS_DRV | P    | Power supply input pins for Sigma-delta DAC Driver.                                     |
| VCOM_DAC           | P    | Sigma-delta DAC Common mode output.   |
| VMID_DAC           | P    | Sigma-delta DAC VMID output.  |
| VMID_ADC           | P    | Sigma-delta ADC VMID output.  |
| MIC_BIAS           | P    | Sigma-delta ADC Microphone Bias Voltage output.   |
| P0.0/URXD0         | I/O  | <b>P0.0</b> — General purpose digital input/output pin.                                 |
|                    |      | <b>URXD0</b> — Receiver input for UART0.  |
| P0.1/UTXD0         | I/O  | <b>P0.1</b> — General purpose digital input/output pin.                                 |
|                    |      | <b>UTXD0</b> — Transmitter output for UART0.  |
| P0.2/SCL0          | I/O  | <b>P0.2</b> — General purpose digital input/output pin.                                 |
|                    |      | <b>SCL0</b> — I2C clock input/output. High-current sink only during I2C Fast-mode Plus. |
| P0.3/SDA0          | I/O  | <b>P0.3</b> — General purpose digital input/output pin.                                 |
|                    |      | <b>SDA0</b> — I2C data input/output. High-current sink only during I2C Fast-mode Plus.  |
| P0.4/SCK0/PGDCLK   | I/O  | <b>P0.4</b> — General purpose digital input/output pin.                                 |
|                    |      | <b>SCK0</b> — Serial clock for SSP0.  |
|                    |      | <b>PGDCLK</b> — Flash clock pin in programming mode.                                    |
| P0.5/SEL0/PGDIN    | I/O  | <b>P0.5</b> — General purpose digital input/output pin.                                 |
|                    |      | <b>SEL0</b> — Slave Select for SSP0.  |
|                    |      | <b>PGDIN</b> — Serial data pin in programming mode.                                     |
| P0.6/MISO0/OTPCLK  | I/O  | <b>P0.6</b> — General purpose digital input/output pin.                                 |
|                    |      | <b>MISO0</b> — Master In Slave Out for SSP0.  |
|                    |      | <b>OTPCLK</b> — Serial clock input in programming mode.                                 |
| P0.7/MOSI0/VR_DOUT | I/O  | <b>P0.7</b> — General purpose digital input/output pin with high-current sink driver.   |
|                    |      | <b>MOSI0</b> — Master Out Slave In for SSP0.  |

|                     |     |   |
|---------------------|-----|---|
|                     |     | <b>VR_DOUT</b> — Recognition code/verify data/checksum/ID output in programming mode. |
| P0.8/SCK1           | I/O | <b>P0.8</b> — General purpose digital input/output pin.                               |
|                     |     | <b>SCK1</b> — Serial clock for SSP1.  |
| P0.9/SEL1           | I/O | <b>P0.9</b> — General purpose digital input/output pin.                               |
|                     |     | <b>SEL1</b> — Slave Select for SSP1.  |
| P0.10/MISO1         | I/O | <b>P0.10</b> — General purpose digital input/output pin.                              |
|                     |     | <b>MISO1</b> — Master In Slave Out for SSP1.  |
| P0.11/MOSI1         | I/O | <b>P0.11</b> — General purpose digital input/output pin.                              |
|                     |     | <b>MOSI1</b> — Master Out Slave In for SSP1.  |
| P0.12/SWCLK         | I/O | <b>P0.12</b> — General purpose digital input/output pin.                              |
|                     |     | <b>SWCLK</b> — Serial wire clock.   |
| P0.13/SWDIO         | I/O | <b>P0.13</b> — General purpose digital input/output pin.                              |
|                     |     | <b>SWDIO</b> — Serial wire debug input/output.  |
| P0.14/DPDWAKEUP     | I   | <b>P0.14</b> — General purpose digital input/output pin.                              |
|                     |     | <b>DPDWAKEUP</b> — Deep power-down mode wake-up pin.                                  |
| P0.15/ <u>RESET</u> | I/O | <b>P0.15</b> — General purpose digital input/output pin.                              |
|                     |     | <b>RESET</b> — external Reset input.  |
| P1.0/AUXTALOUT      | I/O | <b>P1.0</b> — General purpose digital input/output pin.                               |
|                     |     | <b>AUXTALOUT</b> — External high-speed X'tal output pin for audio.                    |
| P1.1/AUXTALIN       | I/O | <b>P1.1</b> — General purpose digital input/output pin.                               |
|                     |     | <b>AUXTALIN</b> — External high-speed X'tal input pin for audio.                      |
| P1.2/I2SMCLK        | I/O | <b>P1.2</b> — General purpose digital input/output pin.                               |
|                     |     | <b>I2SMCLK</b> — MCLK for I2S.  |
| P1.3/I2SDIN         | I/O | <b>P1.3</b> — General purpose digital input/output pin.                               |
|                     |     | <b>I2SDIN</b> — Serial data in for I2S.   |
| P1.4/I2SDOUT        | I/O | <b>P1.4</b> — General purpose digital input/output pin.                               |
|                     |     | <b>I2SDOUT</b> — Serial data out for I2S.   |
| P1.5/I2SBCLK        | I/O | <b>P1.5</b> — General purpose digital input/output pin.                               |
|                     |     | <b>I2SBCLK</b> — BCLK for I2S.  |
| P1.6/I2SWS          | I/O | <b>P1.6</b> — General purpose digital input/output pin.                               |
|                     |     | <b>I2SWS</b> — WS for I2S.  |

|                |     |  |
|----------------|-----|--|
| P1.7/MIC_P     | I/O | <b>P1.7</b> — General purpose digital input/output pin.  |
|                |     | <b>MIC_P</b> — Sigma-delta ADC MIC difference input (+). |
| P1.8/MIC_N     | I/O | <b>P1.8</b> — General purpose digital input/output pin.  |
|                |     | <b>MIC_N</b> — Sigma-delta ADC MIC difference input (-). |
| VOUTP          | O   | <b>VOUTP</b> — Sigma-delta DAC output (+).               |
| VOUTN          | O   | <b>VOUTN</b> — Sigma-delta DAC output (-).               |
| P1.9/CLKOUT    | I/O | <b>P1.9</b> — General purpose digital input/output pin.  |
|                |     | <b>CLKOUT</b> — Clockout pin.                            |
| P1.10/LXTALIN  | I/O | <b>P1.10</b> — General purpose digital input/output pin. |
|                |     | <b>LXTALIN</b> — External low-speed X'tal input pin.     |
| P1.11/LXTALOUT | I/O | <b>P1.11</b> — General purpose digital input/output pin. |
|                |     | <b>LXTALOUT</b> — External low-speed X'tal output pin.   |
| P1.12/XTALIN   | I/O | <b>P1.12</b> — General purpose digital input/output pin. |
|                |     | <b>XTALIN</b> — External high-speed X'tal input pin.     |
| P1.13/XTALOUT  | I/O | <b>P1.13</b> — General purpose digital input/output pin. |
|                |     | <b>XTALOUT</b> — External high-speed X'tal output pin.   |
| P2.0/CMP       | I/O | <b>P2.0</b> — General purpose digital input/output pin.  |
|                |     | <b>CMP</b> — Comparator channel 0.                       |
| P2.1/CM1       | I/O | <b>P2.1</b> — General purpose digital input/output pin.  |
|                |     | <b>CM1</b> — Comparator channel 1.                       |
| P2.2/CM2       | I/O | <b>P2.2</b> — General purpose digital input/output pin.  |
|                |     | <b>CM2</b> — Comparator channel 2.                       |
| P2.3/CM3       | I/O | <b>P2.3</b> — General purpose digital input/output pin.  |
|                |     | <b>CM3</b> — Comparator channel 3.                       |
| P2.4/CM4       | I/O | <b>P2.4</b> — General purpose digital input/output pin.  |
|                |     | <b>CM4</b> — Comparator channel 4.                       |
| P2.5/CM5       | I/O | <b>P2.5</b> — General purpose digital input/output pin.  |
|                |     | <b>CM5</b> — Comparator channel 5.                       |
| P2.6/CM6       | I/O | <b>P2.6</b> — General purpose digital input/output pin.  |
|                |     | <b>CM6</b> — Comparator channel 6.                       |
| P2.7/CM7       | I/O | <b>P2.7</b> — General purpose digital input/output pin.  |
|                |     | <b>CM7</b> — Comparator channel 7.                       |

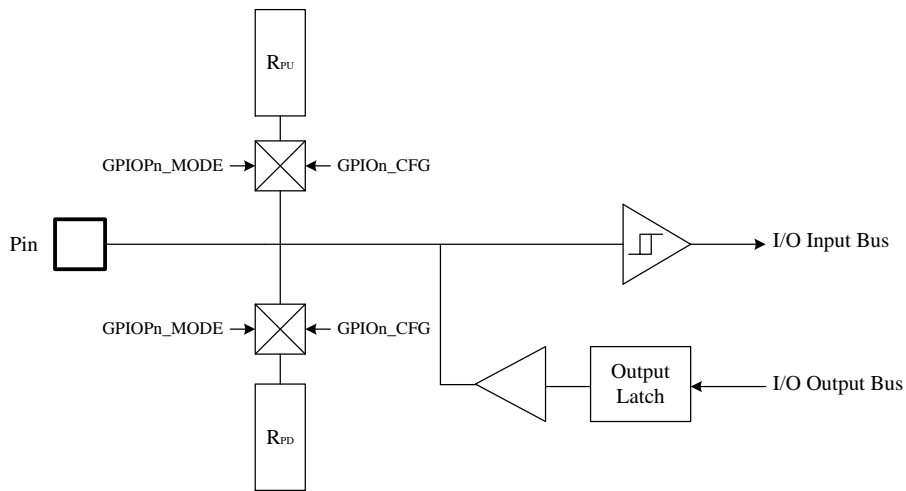
|                        |     |  |
|------------------------|-----|--|
| P2.8/CM8               | I/O | <b>P2.8</b> — General purpose digital input/output pin.  |
|                        |     | <b>CM8</b> — Comparator channel 8.                       |
| P2.9/CM9               | I/O | <b>P2.9</b> — General purpose digital input/output pin.  |
|                        |     | <b>CM9</b> — Comparator channel 9.                       |
| P2.10/CM10             | I/O | <b>P2.10</b> — General purpose digital input/output pin. |
|                        |     | <b>CM10</b> — Comparator channel 10.                     |
| P2.11/CM11             | I/O | <b>P2.11</b> — General purpose digital input/output pin. |
|                        |     | <b>CM11</b> — Comparator channel 11.                     |
| P2.12/CM12             | I/O | <b>P2.12</b> — General purpose digital input/output pin. |
|                        |     | <b>CM12</b> — Comparator channel 12.                     |
| P2.13/CM13             | I/O | <b>P2.13</b> — General purpose digital input/output pin. |
|                        |     | <b>CM13</b> — Comparator channel 13.                     |
| P2.14/CM14             | I/O | <b>P2.14</b> — General purpose digital input/output pin. |
|                        |     | <b>CM14</b> — Comparator channel 14.                     |
| P2.15/CM15             | I/O | <b>P2.15</b> — General purpose digital input/output pin. |
|                        |     | <b>CM15</b> — Comparator channel 15.                     |
| P3.0/CM16              | I/O | <b>P3.0</b> — General purpose digital input/output pin.  |
|                        |     | <b>CM16</b> — Comparator channel 16.                     |
| P3.1/CM17              | I/O | <b>P3.1</b> — General purpose digital input/output pin.  |
|                        |     | <b>CM17</b> — Comparator channel 17.                     |
| P3.2/CM18              | I/O | <b>P3.2</b> — General purpose digital input/output pin.  |
|                        |     | <b>CM18</b> — Comparator channel 18.                     |
| P3.3/CM19              | I/O | <b>P3.3</b> — General purpose digital input/output pin.  |
|                        |     | <b>CM19</b> — Comparator channel 19.                     |
| P3.4/CM20              | I/O | <b>P3.4</b> — General purpose digital input/output pin.  |
|                        |     | <b>CM20</b> — Comparator channel 20.                     |
| P3.5/CM21/CT16B1_ PWM0 | I/O | <b>P3.5</b> — General purpose digital input/output pin.  |
|                        |     | <b>CM21</b> — Comparator channel 21.                     |
|                        |     | <b>CT16B1_PWM0</b> — PWM output 0 for CT16B1.            |
| P3.6/CM22/CT32B0_ PWM1 | I/O | <b>P3.6</b> — General purpose digital input/output pin.  |
|                        |     | <b>CM22</b> — Comparator channel 22.                     |
|                        |     | <b>CT32B0_PWM1</b> — PWM output 1 for CT32B0.            |

|                         |     |   |
|-------------------------|-----|---|
| P3.7/CM23/CT32B1_PWM1   | I/O | <b>P3.7</b> — General purpose digital input/output pin.                                 |
|                         |     | <b>CM23</b> — Comparator channel 23.  |
|                         |     | <b>CT32B1_PWM1</b> — PWM output 1 for CT32B1.   |
| P3.8/CMO                | I/O | <b>P3.8</b> — General purpose digital input/output pin.                                 |
|                         |     | <b>CMO</b> — Comparator output pin.   |
| P3.9/CT16B0_PWM0        | I/O | <b>P3.9</b> — General purpose digital input/output pin.                                 |
|                         |     | <b>CT16B0_PWM0</b> — PWM output 0 for CT16B0.   |
| P3.10/CT32B0_PWM0       | I/O | <b>P3.10</b> — General purpose digital input/output pin.                                |
|                         |     | <b>CT32B0_PWM0</b> — PWM output 0 for CT32B0.   |
| P3.11/CT32B1_PWM0       | I/O | <b>P3.11</b> — General purpose digital input/output pin.                                |
|                         |     | <b>CT32B1_PWM0</b> — PWM output 0 for CT32B1.   |
| P3.12/URXD1/CT16B0_CAP0 | I/O | <b>P3.12</b> — General purpose digital input/output pin.                                |
|                         |     | <b>URXD1</b> — Receiver data input for UART1.   |
|                         |     | <b>CT16B0_CAP0</b> — Capture input 0 for CT16B0.  |
| P3.13/UTXD1/CT16B1_CAP0 | I/O | <b>P3.13</b> — General purpose digital input/output pin.                                |
|                         |     | <b>UTXD1</b> — Transmitter data output for UART1.                                       |
|                         |     | <b>CT16B1_CAP0</b> — Capture input 0 for CT16B1.  |
| P3.14/SCL1/CT32B0_CAP0  | I/O | <b>P3.14</b> — General purpose digital input/output pin.                                |
|                         |     | <b>SCL1</b> — I2C clock input/output. High-current sink only during I2C Fast-mode Plus. |
|                         |     | <b>CT32B0_CAP0</b> — Capture input 0 for CT32B0.  |
| P3.15/SDA1/CT32B1_CAP0  | I/O | <b>P3.15</b> — General purpose digital input/output pin.                                |
|                         |     | <b>SDA1</b> — I2C data input/output. High-current sink only during I2C Fast-mode Plus.  |
|                         |     | <b>CT32B1_CAP0</b> — Capture input 0 for CT32B1.  |

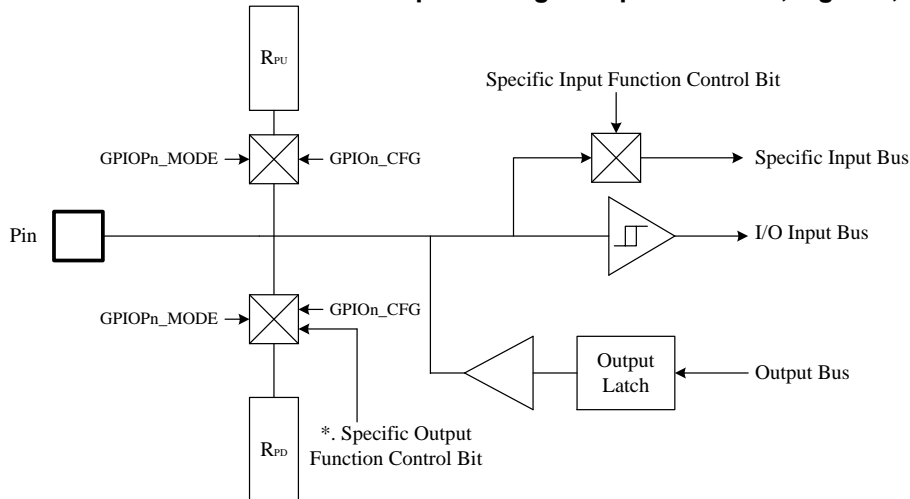


## 1.6 PIN CIRCUIT DIAGRAMS

- Normal Bi-direction I/O Pin.

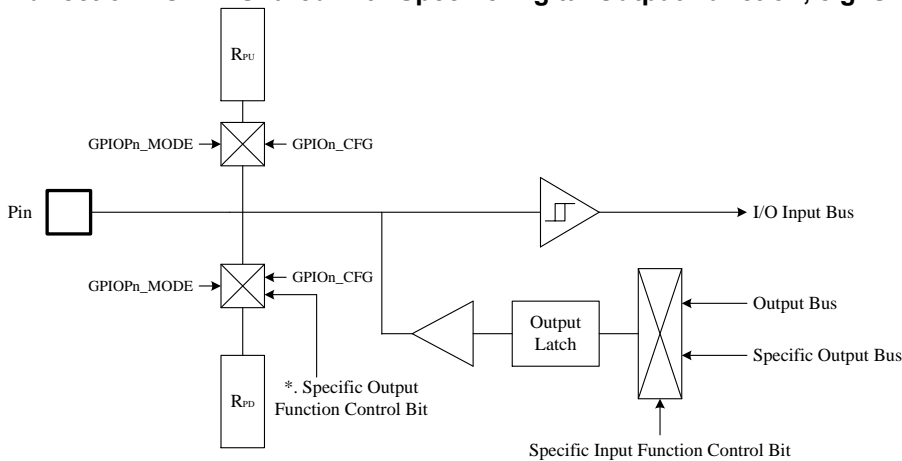


- Bi-direction I/O Pin Shared with Specific Digital Input Function, e.g. SPI, I2C...



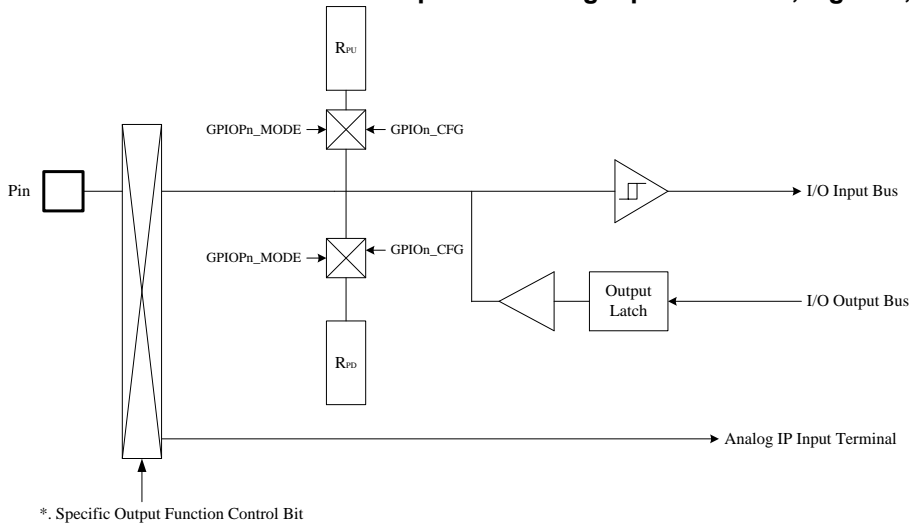
\*. Some specific functions switch I/O direction directly, not through `GPIO_n_MODE` register.

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

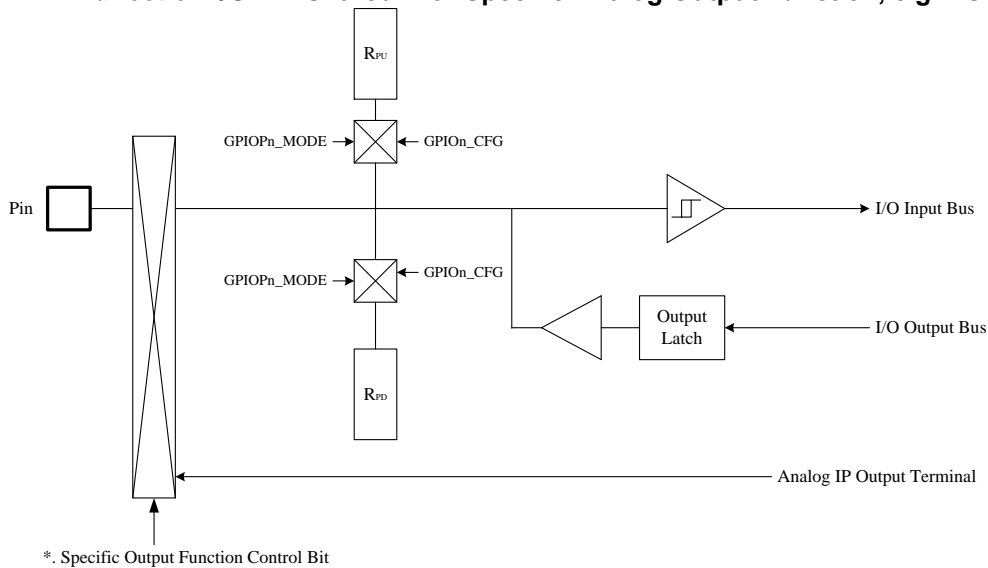


\*. Some specific functions switch I/O direction directly, not through `GPIO_n_MODE` register.

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

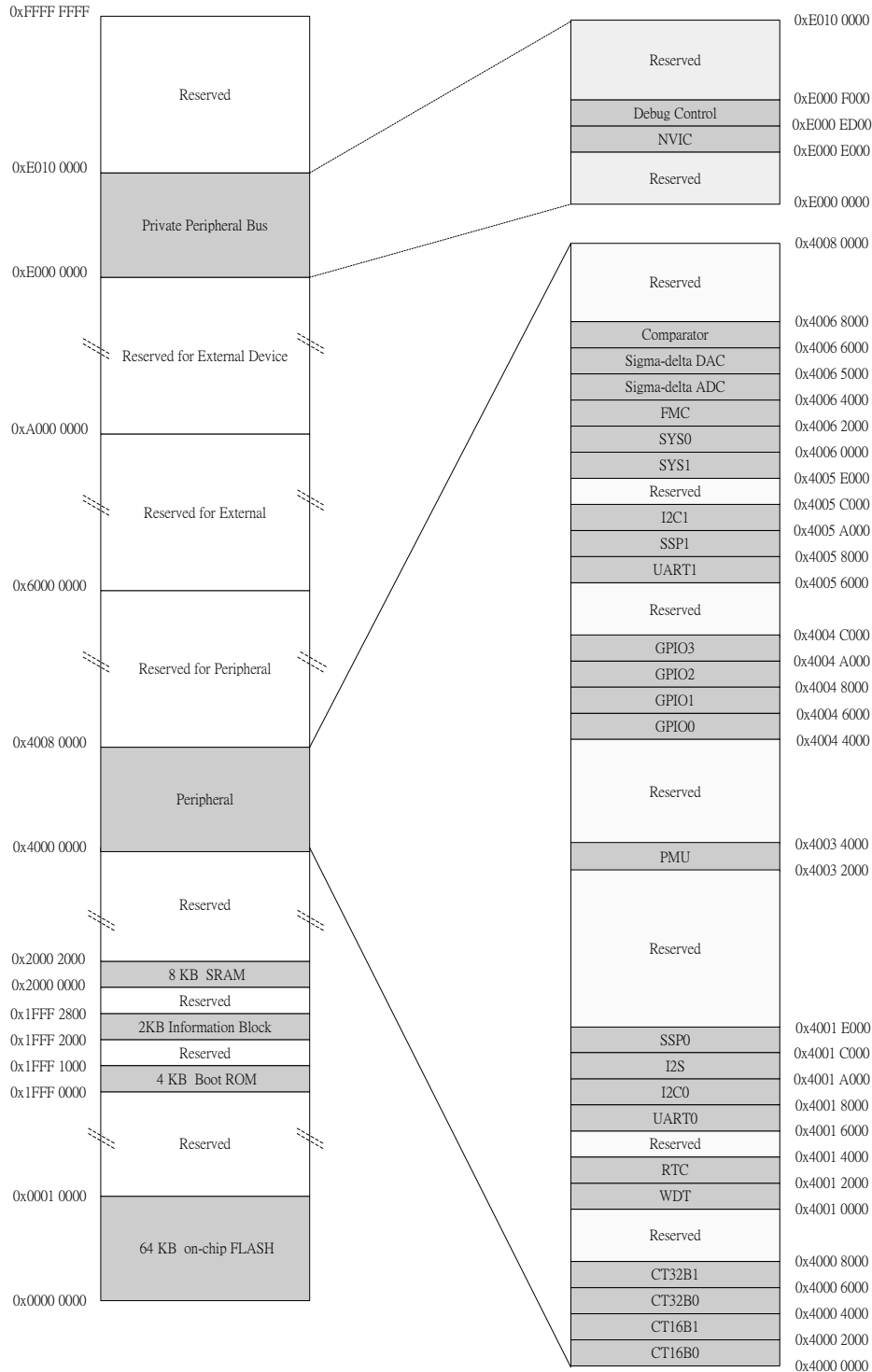


● **Bi-direction I/O Pin Shared with Specific Analog Output Function, e.g. XOUT...**



# 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.

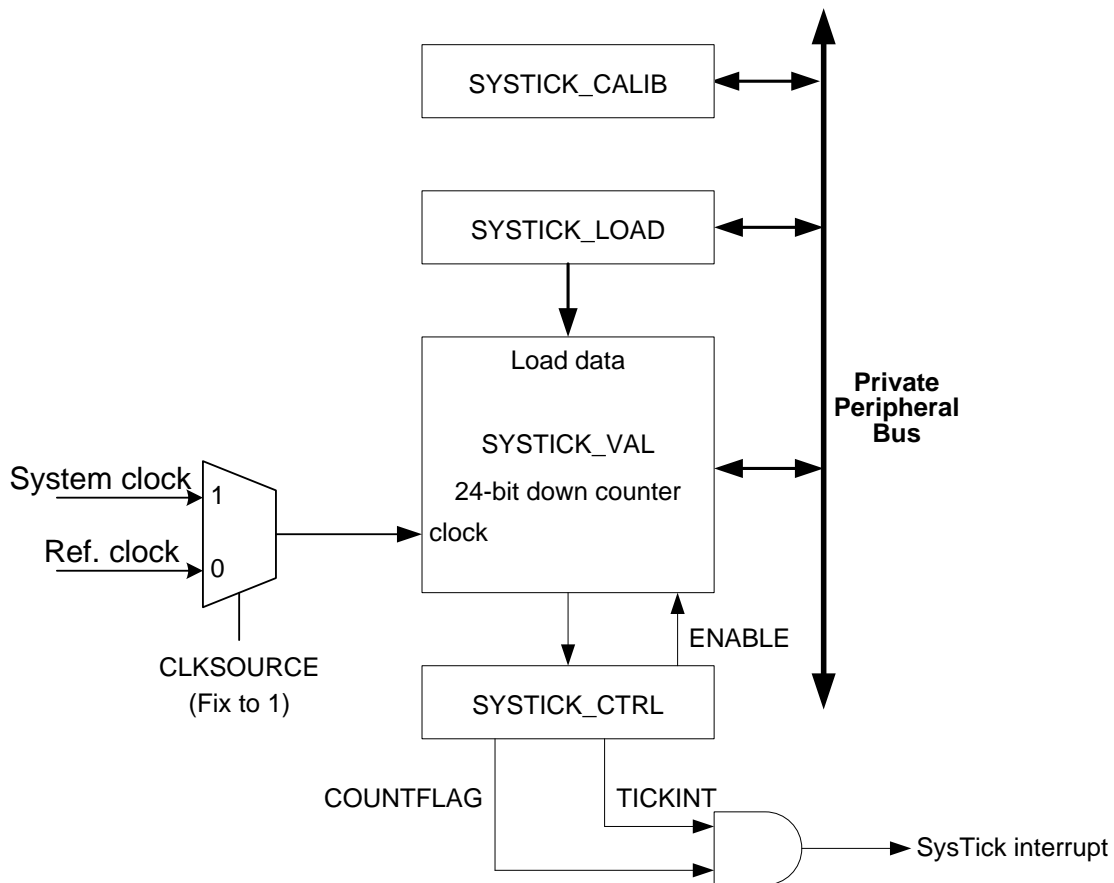
Refer to the *Cortex-M0 User Guide* for details.

## 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 (SYST\_VAL) to zero, reloads to the value in the SysTick Reload Value Register (SYST\_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.<br>0: reference clock.<br>1: system clock. (Fixed)  | R         | 1     |
| 1     | TICKINT   | System Tick interrupt enable.<br>0: Disable the System Tick interrupt<br>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.<br>0: Disable<br>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 SYST\_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 SYST\_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

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

| Bit   | Name     | Description  | Attribute | Reset    |
|-------|----------|--|-----------|----------|
| 31:24 | Reserved |  | R         | 0        |
| 23:0  | RELOAD   | Value to load into the SYST_CVR 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     |

|             |                |   |     |          |
|-------------|----------------|---|-----|----------|
| <b>23:0</b> | <b>CURRENT</b> | 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 SYST_CSR. | R/W | 0x7E7F35 |
|-------------|----------------|---|-----|----------|

#### 2.2.3.4 System Tick Timer Calibration Value register (SYST\_CALIB)

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

| Bit          | Name            | Description  | Attribute | Reset   |
|--------------|-----------------|--|-----------|---------|
| <b>31</b>    | <b>NOREF</b>    | Indicates the reference clock to M0 is provided or not.<br>1: No reference clock provided.   | R         | 1       |
| <b>30</b>    | <b>SKEW</b>     | Indicates whether the TENMS value is exact, an inexact TENMS value can affect the suitability of SysTick as a software real time clock.<br>0: TENMS value is exact<br>1: TENMS value is inexact, or not given. | R         | 0       |
| <b>29:24</b> | <b>Reserved</b> |  | R         | 0       |
| <b>23:0</b>  | <b>TENMS</b>    | 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

### 2.3.2 NVIC REGISTERS

| 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/P0IRQ        | GPIO interrupt status of port 0 | 0x0000 0040    |
| 17            | Settable | IRQ1/P1IRQ        | GPIO interrupt status of port 1 | 0x0000 0044    |
| 18            | Settable | IRQ2/P2IRQ        | GPIO interrupt status of port 2 | 0x0000 0048    |
| 19            | Settable | IRQ3/P3IRQ        | GPIO interrupt status of port 3 | 0x0000 004C    |
| 20~32         | Reserved | Reserved          | Reserved                        | -              |
| 33            | Settable | IRQ17/CMPIRQ      | CMP                             | 0x0000 0084    |
| 34            | Settable | IRQ18/CT16B0IRQ   | CT16B0                          | 0x0000 0088    |
| 35            | Settable | IRQ19/CT16B1IRQ   | CT16B1                          | 0x0000 008C    |
| 36            | Settable | IRQ20/CT32B0IRQ   | CT32B0                          | 0x0000 0090    |
| 37            | Settable | IRQ21/CT32B1IRQ   | CT32B1                          | 0x0000 0094    |
| 38            | Settable | IRQ22/I2SIRQ      | I2S                             | 0x0000 0098    |
| 39            | Settable | IRQ23/SSP0IRQ     | SSP0                            | 0x0000 009C    |
| 40            | Settable | IRQ24/SSP1IRQ     | SSP1                            | 0x0000 00A0    |
| 41            | Settable | IRQ25/UART0IRQ    | UART0                           | 0x0000 00A4    |
| 42            | Settable | IRQ26/UART1IRQ    | UART1                           | 0x0000 00A8    |

|    |          |               |      |             |
|----|----------|---------------|------|-------------|
| 43 | Settable | IRQ27/I2C0IRQ | I2C0 | 0x0000 00AC |
| 44 | Settable | IRQ28/I2C1IRQ | I2C1 | 0x0000 00B0 |
| 45 | Settable | IRQ29/WDTIRQ  | WDT  | 0x0000 00B4 |
| 46 | Settable | IRQ30/LVDIRQ  | LVD  | 0x0000 00B8 |
| 47 | Settable | IRQ31/RTCIRQ  | RTC  | 0x0000 00BC |

### 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.<br>Write→ 0: No effect<br>1: Enable interrupt.<br>Read→ 0: Interrupt disabled<br>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.<br>Write→ 0: No effect<br>1: Disable interrupt.<br>Read→ 0: Interrupt disabled<br>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.<br>Write→ 0: No effect<br>1: Change interrupt state to pending<br>Read→ 0: Interrupt is not pending<br>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.<br>Write → 0: No effect<br>1: Removes pending state of an interrupt<br>Read → 0: Interrupt is not pending<br>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.<br>Read as unknown. Write 0x05FA to VECTKEY, otherwise the write is ignored.  | R/W       | 0     |
| 15    | ENDIANESS     | Data endianness implemented<br>0: Little-endian<br>1: Big-endian  | R         | 0     |
| 14:3  | Reserved      |   | R         | 0     |
| 2     | SYSRESETREQ   | System reset request. This bit read as 0.<br>0: No effect<br>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     |

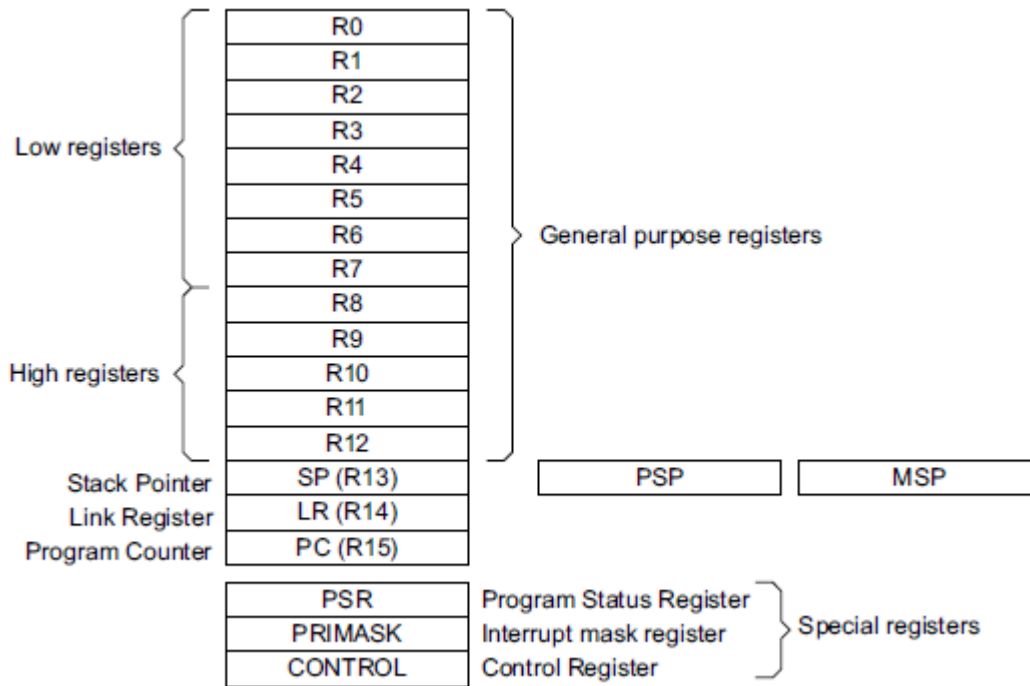
|   |          |  |   |   |
|---|----------|--|---|---|
| 0 | Reserved |  | R | 0 |
|---|----------|--|---|---|

## 2.5 CODE OPTION TABLE

Address: 0x1FFF 2000

| Bit   | Name                | Description   | Attribute | Reset  |
|-------|---------------------|---|-----------|--------|
| 31:16 | Code Security[15:0] | Code Security<br>0xFFFF: CS0<br>0x5A5A: CS1<br>0xA5A5: CS2<br>0x55AA: CS3 | R/W       | 0xFFFF |
| 15:0  | Reserved            |   | R         | 1      |

## 2.6 CORE REGISTER OVERVIEW



| Register | Description (Refer to Cortex-M0 Spec)  |    |    |    |          |          |    |    |    |    |  |  |   |                  |   |   |  |   |      |   |   |   |   |          |  |  |  |  |  |  |  |  |  |      |          |  |  |  |  |  |  |  |  |  |  |  |  |                  |  |  |      |          |  |  |  |   |          |  |  |  |  |  |  |  |  |  |
|----------|--|----|----|----|----------|----------|----|----|----|----|--|--|---|------------------|---|---|--|---|------|---|---|---|---|----------|--|--|--|--|--|--|--|--|--|------|----------|--|--|--|--|--|--|--|--|--|--|--|--|------------------|--|--|------|----------|--|--|--|---|----------|--|--|--|--|--|--|--|--|--|
| R0~R12   | General-purpose registers for data operations.   |    |    |    |          |          |    |    |    |    |  |  |   |                  |   |   |  |   |      |   |   |   |   |          |  |  |  |  |  |  |  |  |  |      |          |  |  |  |  |  |  |  |  |  |  |  |  |                  |  |  |      |          |  |  |  |   |          |  |  |  |  |  |  |  |  |  |
| SP (R13) | The Stack Pointer (SP). In Thread mode, the CONTROL register indicates the stack pointer to use, Main Stack Pointer (MSP) or Process Stack Pointer (PSP)<br>On reset, the processor loads the MSP with the value from address 0x00000000.  |    |    |    |          |          |    |    |    |    |  |  |   |                  |   |   |  |   |      |   |   |   |   |          |  |  |  |  |  |  |  |  |  |      |          |  |  |  |  |  |  |  |  |  |  |  |  |                  |  |  |      |          |  |  |  |   |          |  |  |  |  |  |  |  |  |  |
| LR (R14) | The Link Register (LR). It stores the return information for subroutines, function calls, and exceptions.  |    |    |    |          |          |    |    |    |    |  |  |   |                  |   |   |  |   |      |   |   |   |   |          |  |  |  |  |  |  |  |  |  |      |          |  |  |  |  |  |  |  |  |  |  |  |  |                  |  |  |      |          |  |  |  |   |          |  |  |  |  |  |  |  |  |  |
| PC (R15) | The Program Counter (PC). It contains the current program address.<br>On reset, the processor loads the PC with the value of the reset vector, at address 0x00000004.  |    |    |    |          |          |    |    |    |    |  |  |   |                  |   |   |  |   |      |   |   |   |   |          |  |  |  |  |  |  |  |  |  |      |          |  |  |  |  |  |  |  |  |  |  |  |  |                  |  |  |      |          |  |  |  |   |          |  |  |  |  |  |  |  |  |  |
| PSR      | <p>The Program Status Register (PSR) combines:</p> <ul style="list-style-type: none"> <li>• Application Program Status Register (APSR)</li> <li>• Interrupt Program Status Register (IPSR)</li> <li>• Execution Program Status Register (EPSR).</li> </ul> <p>These registers are mutually exclusive bit fields in the 32-bit PSR.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%;"></td> <td style="width: 5%; text-align: center;">31</td> <td style="width: 5%; text-align: center;">30</td> <td style="width: 5%; text-align: center;">29</td> <td style="width: 5%; text-align: center;">28</td> <td style="width: 5%; text-align: center;">27</td> <td style="width: 5%; text-align: center;">25</td> <td style="width: 5%; text-align: center;">24</td> <td style="width: 5%; text-align: center;">23</td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: center;">6</td> <td style="width: 5%; text-align: center;">5</td> <td style="width: 5%;"></td> <td style="width: 5%; text-align: center;">0</td> </tr> <tr> <td>APSR</td> <td style="text-align: center;">N</td> <td style="text-align: center;">Z</td> <td style="text-align: center;">C</td> <td style="text-align: center;">V</td> <td colspan="10" style="text-align: center;">Reserved</td> </tr> <tr> <td>IPSR</td> <td colspan="13" style="text-align: center;">Reserved</td> <td colspan="3" style="text-align: center;">Exception number</td> </tr> <tr> <td>EPSR</td> <td colspan="4" style="text-align: center;">Reserved</td> <td style="text-align: center;">T</td> <td colspan="10" style="text-align: center;">Reserved</td> </tr> </table> |    | 31 | 30 | 29       | 28       | 27 | 25 | 24 | 23 |  |  |   |                  | 6 | 5 |  | 0 | APSR | N | Z | C | V | Reserved |  |  |  |  |  |  |  |  |  | IPSR | Reserved |  |  |  |  |  |  |  |  |  |  |  |  | Exception number |  |  | EPSR | Reserved |  |  |  | T | Reserved |  |  |  |  |  |  |  |  |  |
|          | 31   | 30 | 29 | 28 | 27       | 25       | 24 | 23 |    |    |  |  | 6 | 5                |   | 0 |  |   |      |   |   |   |   |          |  |  |  |  |  |  |  |  |  |      |          |  |  |  |  |  |  |  |  |  |  |  |  |                  |  |  |      |          |  |  |  |   |          |  |  |  |  |  |  |  |  |  |
| APSR     | N  | Z  | C  | V  | Reserved |          |    |    |    |    |  |  |   |                  |   |   |  |   |      |   |   |   |   |          |  |  |  |  |  |  |  |  |  |      |          |  |  |  |  |  |  |  |  |  |  |  |  |                  |  |  |      |          |  |  |  |   |          |  |  |  |  |  |  |  |  |  |
| IPSR     | Reserved   |    |    |    |          |          |    |    |    |    |  |  |   | Exception number |   |   |  |   |      |   |   |   |   |          |  |  |  |  |  |  |  |  |  |      |          |  |  |  |  |  |  |  |  |  |  |  |  |                  |  |  |      |          |  |  |  |   |          |  |  |  |  |  |  |  |  |  |
| EPSR     | Reserved   |    |    |    | T        | Reserved |    |    |    |    |  |  |   |                  |   |   |  |   |      |   |   |   |   |          |  |  |  |  |  |  |  |  |  |      |          |  |  |  |  |  |  |  |  |  |  |  |  |                  |  |  |      |          |  |  |  |   |          |  |  |  |  |  |  |  |  |  |
| PRIMASK  | The PRIMASK register prevents activation of all exceptions with configurable priority.   |    |    |    |          |          |    |    |    |    |  |  |   |                  |   |   |  |   |      |   |   |   |   |          |  |  |  |  |  |  |  |  |  |      |          |  |  |  |  |  |  |  |  |  |  |  |  |                  |  |  |      |          |  |  |  |   |          |  |  |  |  |  |  |  |  |  |
| CONTROL  | The CONTROL register controls the stack used when the processor is in Thread mode.   |    |    |    |          |          |    |    |    |    |  |  |   |                  |   |   |  |   |      |   |   |   |   |          |  |  |  |  |  |  |  |  |  |      |          |  |  |  |  |  |  |  |  |  |  |  |  |                  |  |  |      |          |  |  |  |   |          |  |  |  |  |  |  |  |  |  |

# 3 SYSTEM CONTROL

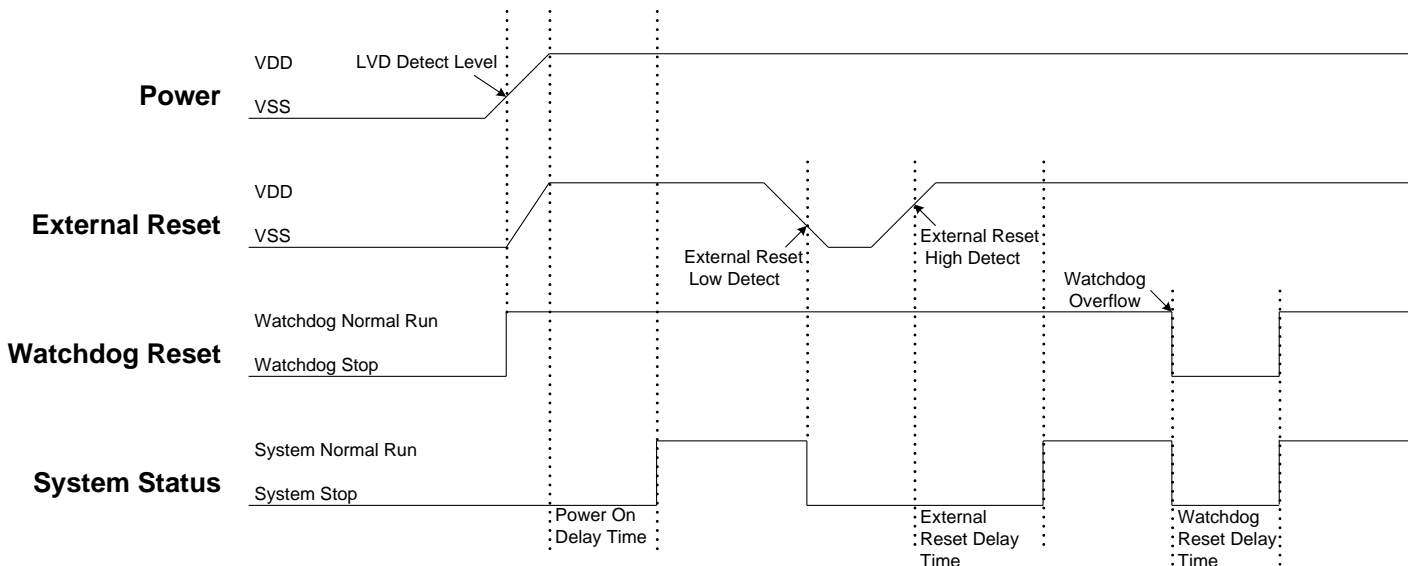
## 3.1 RESET

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 if BLEN bit =1, or from 0x0 if BLEN bit =0.

### 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.

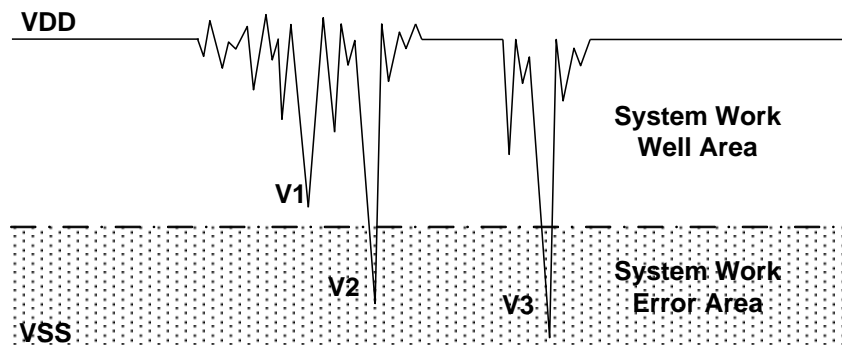
- Before clearing watchdog timer, check I/O status and check RAM contents can improve system error.
- Don't clear watchdog timer in interrupt vector and interrupt service routine. That can improve main routine fail.
- Clearing watchdog timer program is only at one part of the program. This way is the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

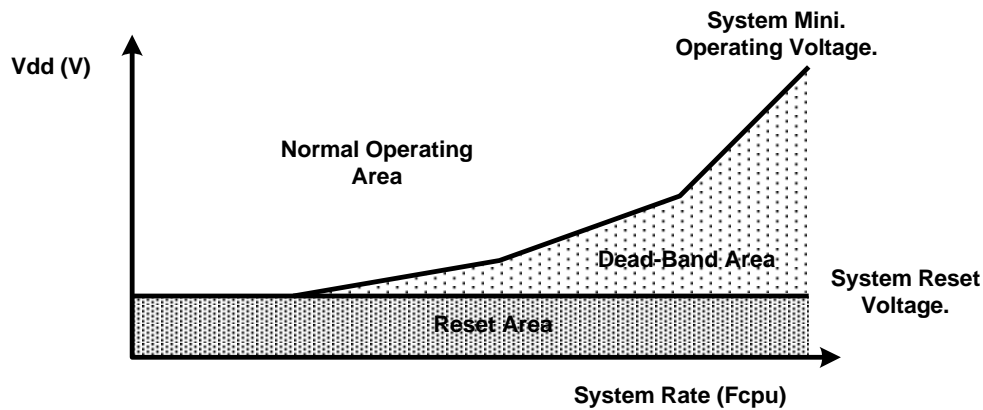
#### AC application:

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.



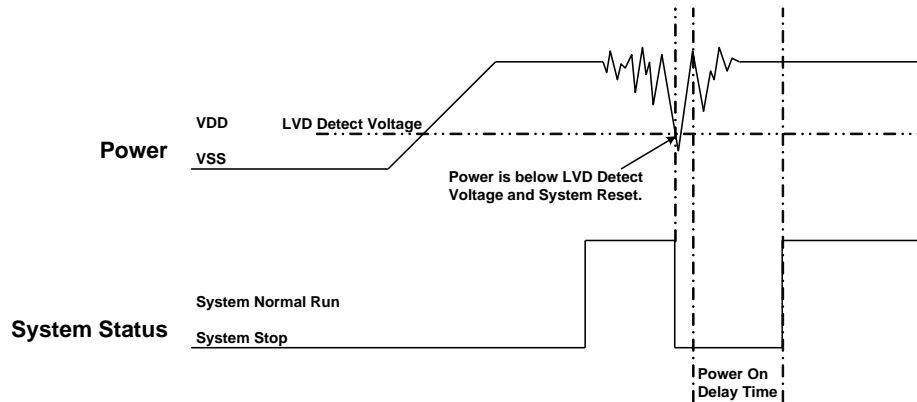
Normally the system operation voltage area is higher than the system reset voltage to VDD, and the reset voltage is decided by LVD detect level. The system minimum operating voltage rises when the system executing rate upper even higher than system reset voltage. The dead-band definition is the system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

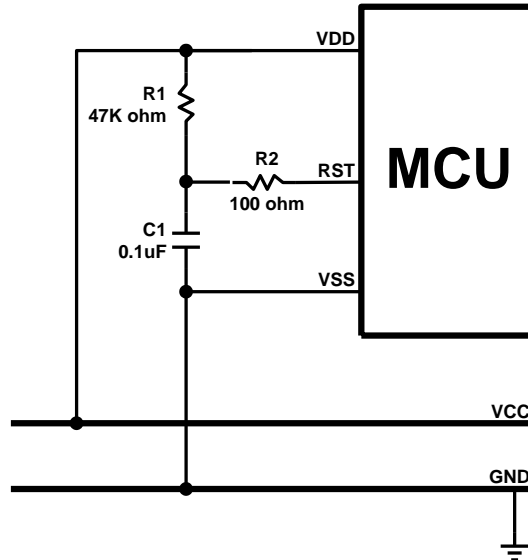
- **External reset (only external reset pin enable):** System checks external reset pin status. If external reset pin is not high level, the system keeps reset status and waits external reset pin released.
- **System initialization:** All system registers is set as initial conditions and system is ready.
- **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 if BLEN bit =1, or from 0x0 if BLEN bit =0.

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.

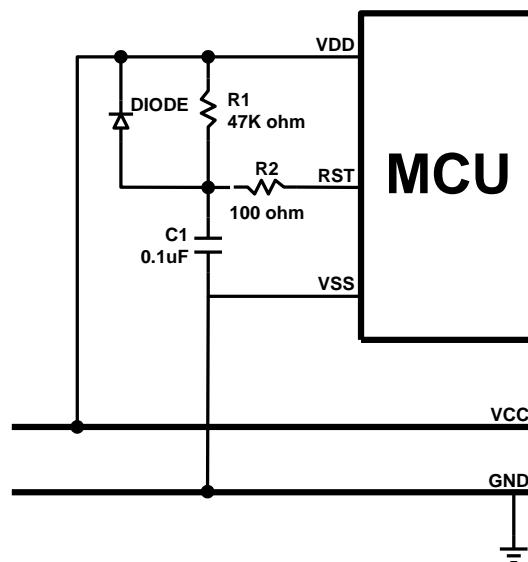
### 3.1.4.1 SIMPLY RC RESET CIRCUIT



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

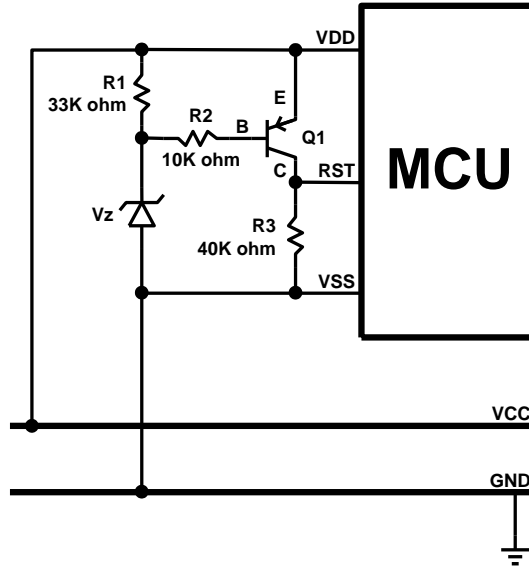


This is the better reset circuit. The R1 and C1 circuit operation is like the simply reset circuit to make a power on signal. The reset circuit has a simply protection against unusual power. The diode offers a power positive path to conduct higher power to VDD. It is can make reset pin voltage level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

improve slight brown out reset condition.

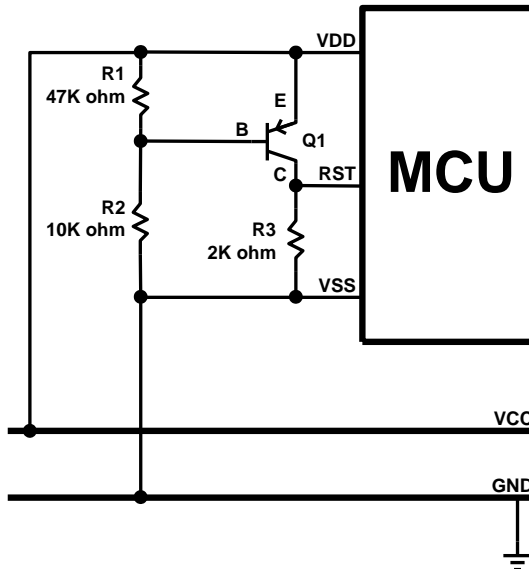
\* **Note:** The R2 100 ohm resistor of “Simply reset circuit” and “Diode & RC reset circuit” is necessary to limit any current flowing into reset pin from external capacitor C in the event of reset pin breakdown due to Electrostatic Discharge (ESD) or Electrical Over-stress (EOS).

### 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



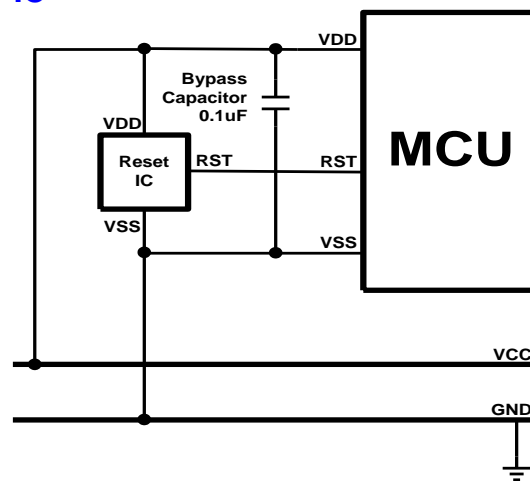
The voltage bias reset circuit is a low cost voltage detector and can **improve brown out reset condition completely**. The operating voltage is not accurate as Zener diode reset circuit. Use R1, R2 bias voltage to be the active level. When VDD voltage level is above or equal to “ $0.7V \times (R1 +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

One clock sources can be used to drive the audio (I2S/Codec) clock (I2SMCLK):

- Audio High speed external (AUEHS) crystal clock for audio

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 IHRGEN bit in [Analog Block Control register \(SYS0\\_ANBCTRL\)](#).

#### 3.2.1.2 Internal Low-speed RC Oscillator (ILRC)

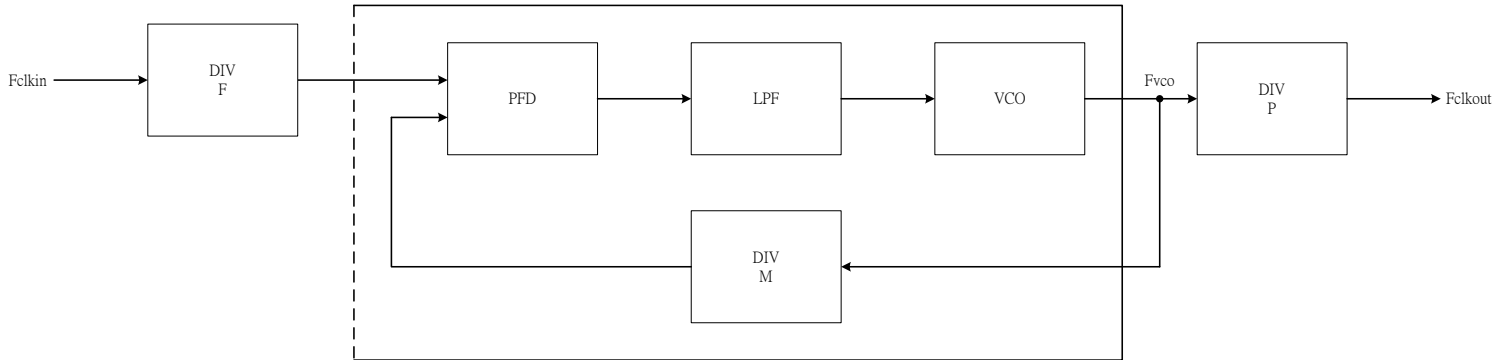
The system low clock source is the internal low-speed oscillator built in the micro-controller. The low-speed oscillator uses RC type oscillator circuit. The frequency is affected by the voltage and temperature of the system. In common condition, the frequency of the RC oscillator is about 16 KHz.

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

### 3.2.2 PLL

SN32F100 series 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  $\mu$ 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:

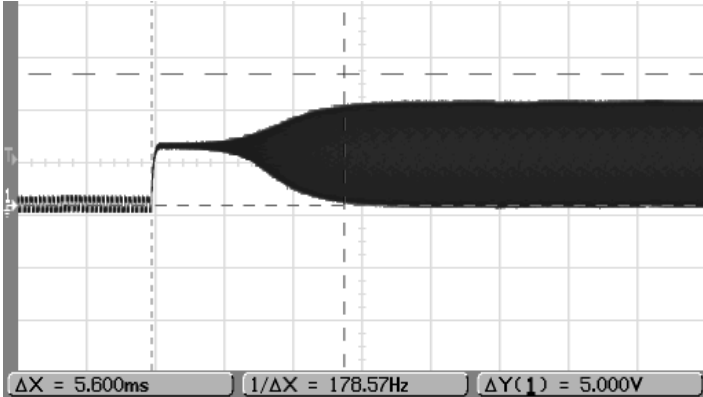
1.  $10\text{MHz} \leq F_{CLKIN} \leq 25\text{MHz}$
2.  $150\text{MHz} \leq F_{VCO} \leq 330\text{MHz}$
3.  $2 < M \leq 31$
4.  $F = 1$ , or 2
5.  $P = 6, 8, 10, 12$ , or 14 (duty 50% +/- 2.5%)
6.  $F_{CLKOUT} = 20\text{MHz}, 30\text{MHz}, 40\text{MHz}, 50\text{MHz}, 24\text{MHz}, 36\text{MHz}, 48\text{MHz},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

### 3.2.3 EXTERNAL CLOCK SOURCE

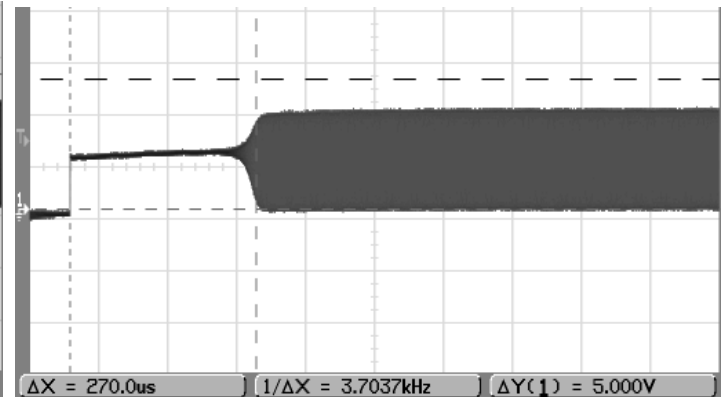
#### 3.2.3.1 External High-speed (EHS) Clock

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

4MHz Crystal

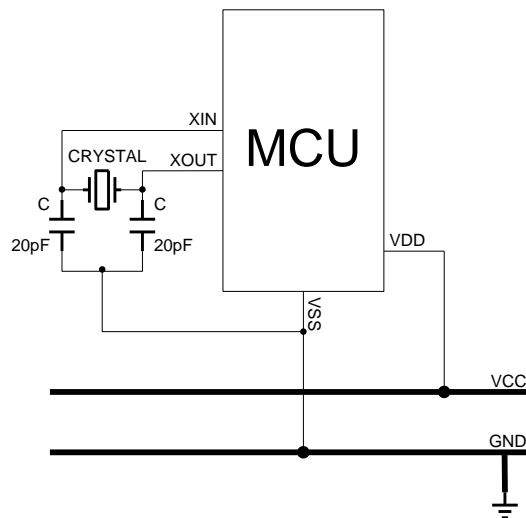


4MHz Ceramic



#### 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<sub>EHS</sub>
- **XIN/XOUT Shared Pin Selection:**

| Oscillator Mode | XTALIN pin      | XTALOUT 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 Audio External High-speed (AUEHS) Clock

\* **Note:** Connect the Crystal/Ceramic and C as near as possible to the AUCTALIN/AUCTALOUT/VSS pins of MCU.

- **Structure:** 1MHz~25MHz AUEHS external crystal/ceramic resonator
- **Main Purpose:** Audio high clock source.
- **Warm-up Time:**  $2048 \times F_{AUEHS}$
- **AUCTALIN/AUCTALOUT Shared Pin Selection:**

| Oscillator Mode | AUCTALIN pin    | AUCTALOUT pin   |
|-----------------|-----------------|-----------------|
| --              | GPIO            | GPIO            |
| AUEHS 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.

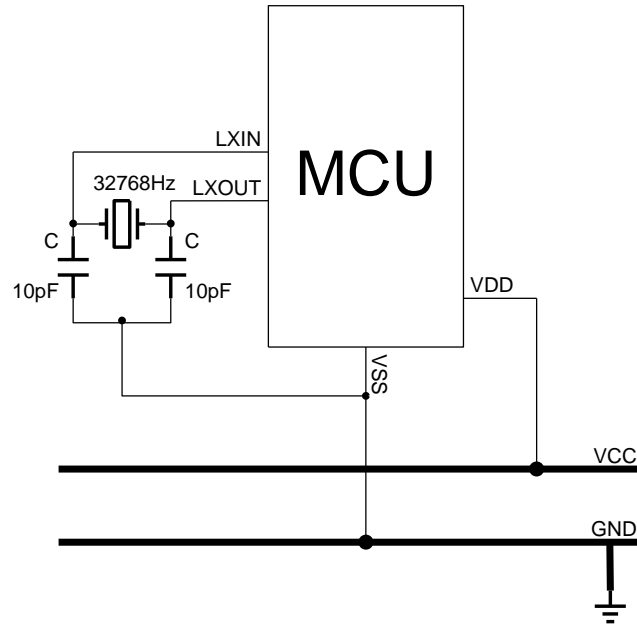
The AUEHS crystal is switched on and off using the AUEHSEN bit in [Analog Block Control register \(SYS0\\_ANBCTRL\)](#).

### 3.2.3.4 External Low-speed (ELS) Clock

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

### 3.2.3.5 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 ELSSEN 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.

### 3.2.3.6 Bypass Mode

| Clock Source                      | H/W Configuration | Description  |
|-----------------------------------|-------------------|--|
| External clock source<br>(Bypass) |                   | <p>In Bypass mode, the external clock signal (square, sinus or triangle) with ~50% duty cycle must be provided to drive the XTALIN/ LXTALIN pin while the XTALOUT/ LXTALOUT 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 EHSSEN bit in <a href="#">Analog Block Control register (SYS0_ANBCTRL)</a>.</p> <p>ELS X'TAL must have a frequency of 32.768 KHz. You select this mode by setting ELSESEN bit in <a href="#">Analog Block Control register (SYS0_ANBCTRL)</a>.</p> |
| External X'TAL<br>(EHS/ELS X'TAL) |                   | <p>The 1 to 25 MHz EHS X'TAL has the advantage of producing a very accurate rate on the main clock</p> <p>ELS X'TAL must have a frequency of 32.768 KHz.</p>   |



---

## 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
7. AUEHS 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 |
|-------|-----------|---|-----------|-------|
| 31:10 | Reserved  |   | R         | 0     |
| 9     | AUEHSFREQ | Frequency range of AUEHS X'TAL<br>0: <=12MHz<br>1: >12MHz   | R/W       | 0     |
| 8     | AUEHSEN   | Audio external high-speed clock enable<br>0: Disable AUEHS X'TAL.<br>1: Enable AUEHS X'TAL.                                   | R/W       | 0     |
| 7:6   | Reserved  |   | R         | 0     |
| 5     | EHSFREQ   | Frequency range (driving ability) of EHS X'TAL<br>0: <=12MHz<br>1: >12MHz   | R/W       | 0     |
| 4     | EHSEN     | External high-speed clock enable<br>0: Disable EHS X'TAL.<br>1: Enable EHS X'TAL.   | R/W       | 0     |
| 3     | Reserved  |   | R         | 0     |
| 2     | ELSEN     | External low-speed oscillator enable<br>0: Disable External 32.768 KHz oscillator<br>1: Enable External 32.768 KHz oscillator | R/W       | 0     |
| 1     | Reserved  |   | R         | 0     |
| 0     | IHRCEN    | Internal high-speed clock enable<br>0: Disable internal 12 MHz RC oscillator.<br>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<br>0: Disable<br>1: Enable | R/W       | 0     |
| 14    | Reserved |                                       | R         | 0     |

|              |                |  |     |      |
|--------------|----------------|--|-----|------|
| <b>13:12</b> | PLLCLKSEL[1:0] | System PLL clock source<br>00: IHRC 12 MHz oscillator<br>01: EHS X'TAL 10 MHz ~ 25 MHz<br>Other: Reserved                        | R/W | 0    |
| <b>11:9</b>  | Reserved       |  | R   | 0    |
| <b>8</b>     | FSEL           | Front divider value. The division value F is the programmed $2^{FSEL}$<br>0: F = 1<br>1: F = 2                                   | R/W | 0    |
| <b>7:5</b>   | PSEL[2:0]      | Post divider value. P= PSEL[2:0]*2<br>000~010: Reserved<br>011: P = 6<br>100: P = 8<br>101: P = 10<br>110: P = 12<br>111: P = 14 | R/W | 011b |
| <b>4:0</b>   | MSEL[4:0]      | Feedback divider value.<br>M: 3~31   | R/W | 0x3  |

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

1.  $10\text{MHz} \leq F_{CLKIN} \leq 25\text{MHz}$
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

|         | Fclkout | 10MHz | 12MHz | 16MHz | 20MHz | 22MHz | 24MHz | 25MHz | 30MHz | 32MHz | 36MHz | 40MHz | 44MHz | 48MHz | 50MHz |
|---------|---------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Fclkkin | 10MHz   |       |       |       | V     |       |       |       | V     |       |       | V     |       |       | V     |
|         | 12MHz   |       |       |       |       |       | V     |       |       |       | V     |       |       | V     |       |
|         | 16MHz   |       |       |       |       |       |       |       |       | V     |       |       |       | V     |       |
|         | 22MHz   |       |       |       |       |       |       |       |       |       |       |       | V     |       |       |
|         | 24MHz   |       |       |       |       |       |       |       |       |       |       |       |       | V     |       |
|         | 25MHz   |       |       |       |       |       |       |       |       |       |       |       |       |       | V     |

### 3.3.2.1 RECOMMEND FREQUENCY SETTING

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

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

| F <sub>CLKIN</sub> (MHz) | FSEL | F=2 <sup>FSEL</sup> | MSEL[4:0]=M | F <sub>VCO</sub> (MHz)<br>=F <sub>CLKIN</sub> / F * M | PSEL[2:0] | P= PSEL[2:0]*2 | F <sub>CLKOUT</sub> (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                        |
| 16                       | 0    | 1                   | 16          | 256   | 4         | 8              | 32                        |
| 16                       | 0    | 1                   | 18          | 288   | 3         | 6              | 48                        |
| 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 |
|------|----------|--|-----------|-------|
| 31:9 | Reserved |  | R         | 0     |
| 8    | AUEHSRDY | Audio external high-speed clock ready flag<br>0: AUEHS oscillator not ready<br>1: AUEHS oscillator ready | R         | 0     |
| 7    | Reserved |  | R         | 0     |
| 6    | PLLRDY   | PLL clock ready flag<br>0: PLL unlocked<br>1: PLL locked   | R         | 0     |
| 5    | Reserved |  | R         | 0     |
| 4    | EHSRDY   | External high-speed clock ready flag<br>0: EHS oscillator not ready<br>1: EHS oscillator ready           | R         | 0     |
| 3    | Reserved |  | R         | 0     |
| 2    | ELSRDY   | External low-speed clock ready flag<br>0: EHS oscillator not ready<br>1: EHS oscillator ready            | R         | 0     |
| 1    | Reserved |  | R         | 0     |
| 0    | IHRCDY   | IHRC ready flag<br>0: IHRC not ready<br>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<br>Set and cleared by HW to indicate which clock source is used as system clock.<br>000: IHRC is used as system clock<br>001: ILRC is used as system clock<br>010: EHS X'TAL is used as system clock<br>011: ELS X'TAL is used as system clock<br>100: PLL is used as system clock<br>Other: Reserved | R         | 0     |
| 3    | Reserved       |  | R         | 0     |
| 2:0  | SYSCLKSEL[2:0] | System clock switch<br>Set and cleared by SW.<br>000: IHRC<br>001: ILRC<br>010: EHS X'TAL<br>011: ELS X'TAL<br>100: PLL output<br>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     |

|            |             |   |     |   |
|------------|-------------|---|-----|---|
| <b>3:0</b> | AHBPRE[3:0] | AHB clock source prescale value<br>0000: SYSCLK / 1<br>0001: SYSCLK / 2<br>0010: SYSCLK / 4<br>0011: SYSCLK / 8<br>0100: SYSCLK / 16<br>0101: SYSCLK / 32<br>0110: SYSCLK / 64<br>0111: SYSCLK / 128<br>1000: SYSCLK / 256<br>1001: SYSCLK / 512<br>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 |
|-------------|----------|--|-----------|-------|
| <b>31:5</b> | Reserved |  | R         | 0     |
| <b>4</b>    | PORRSTF  | POR reset flag<br>Set by HW when a POR reset occurs.<br>0: Read→No POR reset occurred<br>Write→Clear this bit<br>1: POR reset occurred.  | R/W       | 1     |
| <b>3</b>    | EXTRSTF  | External reset flag<br>Set by HW when a reset from the <u>RESET</u> pin occurs.<br>0: Read→No reset from RESET pin occurred<br>Write→Clear this bit<br>1: Reset from RESET pin occurred. | R/W       | 0     |
| <b>2</b>    | LVDRSTF  | LVD reset flag<br>Set by HW when a LVD reset occurs.<br>0: Read→No LVD reset occurred<br>Write→Clear this bit<br>1: LVD reset occurred.  | R/W       | 0     |
| <b>1</b>    | WDRSTF   | WDT reset flag<br>Set by HW when a WDT reset occurs.<br>0: Read→No watchdog reset occurred<br>Write→Clear this bit<br>1: Watchdog reset occurred.  | R/W       | 0     |
| <b>0</b>    | SWRSTF   | Software reset flag<br>Set by HW when a software reset occurs.<br>0: Read→No software reset occurred<br>Write→Clear this bit<br>1: Software reset occurred.                              | R/W       | 0     |

### 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 |
|--------------|----------|---|-----------|-------|
| <b>31:16</b> | Reserved |   | R         | 0     |
| <b>15</b>    | LVDEN    | LVD enable<br>0: Disable<br>1: Enable       | R/W       | 0     |
| <b>14</b>    | LVDRSTEN | LVD Reset enable<br>0: Disable<br>1: Enable | R/W       | 0     |

|             |                |   |     |   |
|-------------|----------------|---|-----|---|
| <b>13:6</b> | Reserved       |   | R   | 0 |
| <b>5:4</b>  | LVDINTLVL[1:0] | LVD interrupt level<br>00: The interrupt assertion threshold voltage is 2.00V<br>01: The interrupt assertion threshold voltage is 2.40V<br>10: The interrupt assertion threshold voltage is 2.70V<br>11: The interrupt assertion threshold voltage is 3.00V | R/W | 0 |
| <b>3:2</b>  | Reserved       |   | R   | 0 |
| <b>1:0</b>  | LVDRSTLVL[1:0] | LVD reset level<br>00: The reset assertion threshold voltage is 2.00V<br>01: The reset assertion threshold voltage is 2.40V<br>10: The reset assertion threshold voltage is 2.70V<br>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.<br>0: Enable external <u>RESET</u> pin. (P0.15 acts as <u>RESET</u> pin)<br>1: Disable. (P0.15 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.<br>0: Enable SWD pin. (P0.13 acts as SWDIO pin, P0.12 acts as SWCLK pin)<br>1: Disable. (P0.13 and P0.12 act as GPIO pins) | R/W       | 0     |

### 3.3.10 Anti-EFT Ability Control register (SYS0\_ANTIEFT)

Address Offset: 0x30

This register decides the HW anti-EFT ability.

| Bit  | Name      | Description   | Attribute | Reset |
|------|-----------|---|-----------|-------|
| 31:3 | Reserved  |   | R         | 0     |
| 2:0  | AEFT[2:0] | HW anti-EFT ability.<br>000: No<br>010: Low<br>011: Medium<br>100: Strong | R/W       | 000   |

## 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<br>000: Disable<br>001: HCLK<br>010: PLL clock output<br>011: ILRC clock<br>100: IHRC clock<br>101: ELS clock<br>110: EHS clock<br>111: AUEHS clock | R/W       | 0     |
| 27:25 | Reserved       |   | R         | 0     |
| 24    | WDTCLKEN       | Enables clock for WDT.<br>0: Disable<br>1: Enable   | R/W       | 1     |
| 23    | RTCCLKEN       | Enables clock for RTC.<br>0: Disable<br>1: Enable   | R/W       | 0     |
| 22    | I2SCLKEN       | Enables clock for I2S.<br>0: Disable<br>1: Enable   | R/W       | 0     |
| 21    | I2C0CLKEN      | Enables clock for I2C0.<br>0: Disable<br>1: Enable  | R/W       | 0     |
| 20    | I2C1CLKEN      | Enables clock for I2C1.<br>0: Disable<br>1: Enable  | R/W       | 0     |
| 19:18 | Reserved       |   | R         | 0     |
| 17    | UART1CLKEN     | Enables clock for UART1.<br>0: Disable<br>1: Enable   | R/W       | 0     |
| 16    | UART0CLKEN     | Enables clock for UART0.<br>0: Disable<br>1: Enable   | R/W       | 0     |
| 15:14 | Reserved       |   | R         | 0     |
| 13    | SSP1CLKEN      | Enables clock for SSP1.<br>0: Disable<br>1: Enable  | R/W       | 0     |
| 12    | SSP0CLKEN      | Enables clock for SSP0.<br>0: Disable<br>1: Enable  | R/W       | 0     |
| 11    | CMPCLKEN       | Enables clock for Comparator.<br>0: Disable<br>1: Enable  | R/W       | 0     |



|     |             |  |     |   |
|-----|-------------|--|-----|---|
| 10  | Reserved    |  | R   | 0 |
| 9   | CT32B1CLKEN | Enables clock for CT32B1.<br>0: Disable<br>1: Enable | R/W | 0 |
| 8   | CT32B0CLKEN | Enables clock for CT32B0.<br>0: Disable<br>1: Enable | R/W | 0 |
| 7   | CT16B1CLKEN | Enables clock for CT16B1.<br>0: Disable<br>1: Enable | R/W | 0 |
| 6   | CT16B0CLKEN | Enables clock for CT16B0.<br>0: Disable<br>1: Enable | R/W | 0 |
| 5:4 | Reserved    |  | R   | 0 |
| 3   | GPIOCKEN    | Enables clock for GPIO.<br>0: Disable<br>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    | Reserved      |   | R         | 0     |
| 30:28 | AUEHSPRE[2:0] | Audio external high clock source prescale value<br>000: AUEHS / 1<br>001: AUEHS / 2<br>010: AUEHS / 4<br>011: AUEHS / 8<br>100: AUEHS / 16<br>Other: Reserved | R/W       | 0     |
| 27    | Reserved      |   | R         | 0     |
| 26:24 | SSP1PRE[2:0]  | SSP1 clock source prescale value<br>000: HCLK / 1<br>001: HCLK / 2<br>010: HCLK / 4<br>011: HCLK / 8<br>100: HCLK / 16<br>Other: Reserved                     | R/W       | 0     |
| 23    | Reserved      |   | R         | 0     |
| 22:20 | SSP0PRE[2:0]  | SSP0 clock source prescale value<br>000: HCLK / 1<br>001: HCLK / 2<br>010: HCLK / 4<br>011: HCLK / 8<br>100: HCLK / 16<br>Other: Reserved                     | R/W       | 0     |
| 19    | Reserved      |   | R         | 0     |
| 18:16 | CMPPRE[2:0]   | Comparator clock source prescale value<br>000: HCLK / 1<br>001: HCLK / 2<br>010: HCLK / 4<br>011: HCLK / 8<br>100: HCLK / 16<br>Other: Reserved               | R/W       | 0     |
| 15    | Reserved      |   | R         | 0     |

|       |                |  |     |   |
|-------|----------------|--|-----|---|
| 14:12 | CT32B1PRE[2:0] | CT32B1 clock source prescale value<br>000: HCLK / 1<br>001: HCLK / 2<br>010: HCLK / 4<br>011: HCLK / 8<br>100: HCLK / 16<br>Other: Reserved  | R/W | 0 |
| 11    | Reserved       |  | R   | 0 |
| 10:8  | CT32B0PRE[2:0] | CT32B0 clock source prescale value.<br>000: HCLK / 1<br>001: HCLK / 2<br>010: HCLK / 4<br>011: HCLK / 8<br>100: HCLK / 16<br>Other: Reserved | R/W | 0 |
| 7     | Reserved       |  | R   | 0 |
| 6:4   | CT16B1PRE[2:0] | CT16B1 clock source prescale value<br>000: HCLK / 1<br>001: HCLK / 2<br>010: HCLK / 4<br>011: HCLK / 8<br>100: HCLK / 16<br>Other: Reserved  | R/W | 0 |
| 3     | Reserved       |  | R   | 0 |
| 2:0   | CT16B0PRE[2:0] | CT16B0 clock source prescale value<br>000: HCLK / 1<br>001: HCLK / 2<br>010: HCLK / 4<br>011: HCLK / 8<br>100: HCLK / 16<br>Other: Reserved  | R/W | 0 |

### 3.4.3 APB Clock Prescale register 1 (SYS1\_APB1CP1)

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<br>0000: Clock-out source / 1<br>0001: Clock-out source / 2<br>0010: Clock-out source / 4<br>0011: Clock-out source / 8<br>0100: Clock-out source / 16<br>0101: Clock-out source / 32<br>0110: Clock-out source / 64<br>0111: Clock-out source / 128<br>1000: Clock-out source / 256<br>1001: Clock-out source / 512<br>Other: Reserved | R/W       | 0     |
| 27    | Reserved       |   | R         | 0     |
| 26:24 | I2C1PRE[2:0]   | I2C1 clock source prescale value<br>000: HCLK / 1<br>001: HCLK / 2<br>010: HCLK / 4<br>011: HCLK / 8<br>100: HCLK / 16<br>Other: Reserved   | R/W       | 0     |
| 23    | Reserved       |   | R         | 0     |

|              |                 |  |     |   |
|--------------|-----------------|--|-----|---|
| <b>22:20</b> | WDTPRE[2:0]     | WDT clock source prescale value<br>000: HCLK / 1<br>001: HCLK / 2<br>010: HCLK / 4<br>011: HCLK / 8<br>100: HCLK / 16<br>101: HCLK / 32<br>Other: Reserved | R/W | 0 |
| <b>19:18</b> | Reserved        |  | R   | 0 |
| <b>17:16</b> | SYSTICKPRE[1:0] | SysTick clock source prescale value<br>00: HCLK / 1<br>01: HCLK / 2<br>10: HCLK / 4<br>11: HCLK / 8  | R/W | 0 |
| <b>15</b>    | Reserved        |  | R   | 0 |
| <b>14:12</b> | I2SPRE[2:0]     | I2S clock source prescale value<br>000: HCLK / 1<br>001: HCLK / 2<br>010: HCLK / 4<br>011: HCLK / 8<br>100: HCLK / 16<br>Other: Reserved                   | R/W | 0 |
| <b>11</b>    | Reserved        |  | R   | 0 |
| <b>10:8</b>  | I2C0PRE[2:0]    | I2C0 clock source prescale value<br>000: HCLK / 1<br>001: HCLK / 2<br>010: HCLK / 4<br>011: HCLK / 8<br>100: HCLK / 16<br>Other: Reserved                  | R/W | 0 |
| <b>7</b>     | Reserved        |  | R   | 0 |
| <b>6:4</b>   | UART1PRE[2:0]   | UART1 clock source prescale value<br>000: HCLK / 1<br>001: HCLK / 2<br>010: HCLK / 4<br>011: HCLK / 8<br>100: HCLK / 16<br>Other: Reserved                 | R/W | 0 |
| <b>3</b>     | Reserved        |  | R   | 0 |
| <b>2:0</b>   | UART0PRE[2:0]   | UART0 clock source prescale value<br>000: HCLK / 1<br>001: HCLK / 2<br>010: HCLK / 4<br>011: HCLK / 8<br>100: HCLK / 16<br>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 |
|--------------|------------|---|-----------|-------|
| <b>31:27</b> | Reserved   |   | R         | 0     |
| <b>26</b>    | CODECADRST | Codec ADC reset<br>0: No effect<br>1: Reset Codec ADC | R/W       | 0     |
| <b>25</b>    | CODECDARST | Codec DAC reset<br>0: No effect<br>1: Reset Codec DAC | R/W       | 0     |
| <b>24</b>    | WDTRST     | WDT reset<br>0: No effect<br>1: Reset WDT             | R/W       | 0     |

|              |                  |   |     |   |
|--------------|------------------|---|-----|---|
| <b>23</b>    | <b>RTCST</b>     | RTC reset<br>0: No effect<br>1: Reset RTC                 | R/W | 0 |
| <b>22</b>    | <b>I2SRST</b>    | I2S reset<br>0: No effect<br>1: Reset I2S                 | R/W | 0 |
| <b>21</b>    | <b>I2C0RST</b>   | I2C0 reset<br>0: No effect<br>1: Reset I2C0               | R/W | 0 |
| <b>20</b>    | <b>I2C1RST</b>   | I2C1 reset<br>0: No effect<br>1: Reset I2C1               | R/W | 0 |
| <b>19:18</b> | Reserved         |   | R   | 0 |
| <b>17</b>    | <b>UART1RST</b>  | UART1 reset<br>0: No effect<br>1: Reset UART1             | R/W | 0 |
| <b>16</b>    | <b>UART0RST</b>  | UART0 reset<br>0: No effect<br>1: Reset UART0             | R/W | 0 |
| <b>15:14</b> | Reserved         |   | R   | 0 |
| <b>13</b>    | <b>SSP1RST</b>   | SSP1 reset<br>0: No effect<br>1: Reset SSP1               | R/W | 0 |
| <b>12</b>    | <b>SSP0RST</b>   | SSP0 reset<br>0: No effect<br>1: Reset SSP0               | R/W | 0 |
| <b>11</b>    | <b>CMRST</b>     | Comparator reset<br>0: No effect<br>1: Reset Comparator   | R/W | 0 |
| <b>10</b>    | Reserved         |   | R   | 0 |
| <b>9</b>     | <b>CT32B1RST</b> | CT32B1 reset<br>0: No effect<br>1: Reset CT32B1           | R/W | 0 |
| <b>8</b>     | <b>CT32B0RST</b> | CT32B0 reset<br>0: No effect<br>1: Reset CT32B0           | R/W | 0 |
| <b>7</b>     | <b>CT16B1RST</b> | CT16B1 reset<br>0: No effect<br>1: Reset CT16B1           | R/W | 0 |
| <b>6</b>     | <b>CT16B0RST</b> | CT16B0 reset<br>0: No effect<br>1: Reset CT16B0           | R/W | 0 |
| <b>5:4</b>   | Reserved         |   | R   | 0 |
| <b>3</b>     | <b>GPIOP3RST</b> | GPIO port 3 reset<br>0: No effect<br>1: Reset GPIO port 3 | R/W | 0 |
| <b>2</b>     | <b>GPIOP2RST</b> | GPIO port 2 reset<br>0: No effect<br>1: Reset GPIO port 2 | R/W | 0 |
| <b>1</b>     | <b>GPIOP1RST</b> | GPIO port 1 reset<br>0: No effect<br>1: Reset GPIO port 1 | R/W | 0 |
| <b>0</b>     | <b>GPIOP0RST</b> | GPIO port 0 reset<br>0: No effect<br>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, Codec, Comparator) 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, Codec, Comparator) 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 anyone of GPIO port pins (P0~P3) interrupt trigger 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 (Please confirm pull-up time to ensure that the DPDWAKEUP pin already in the pull-up state).
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

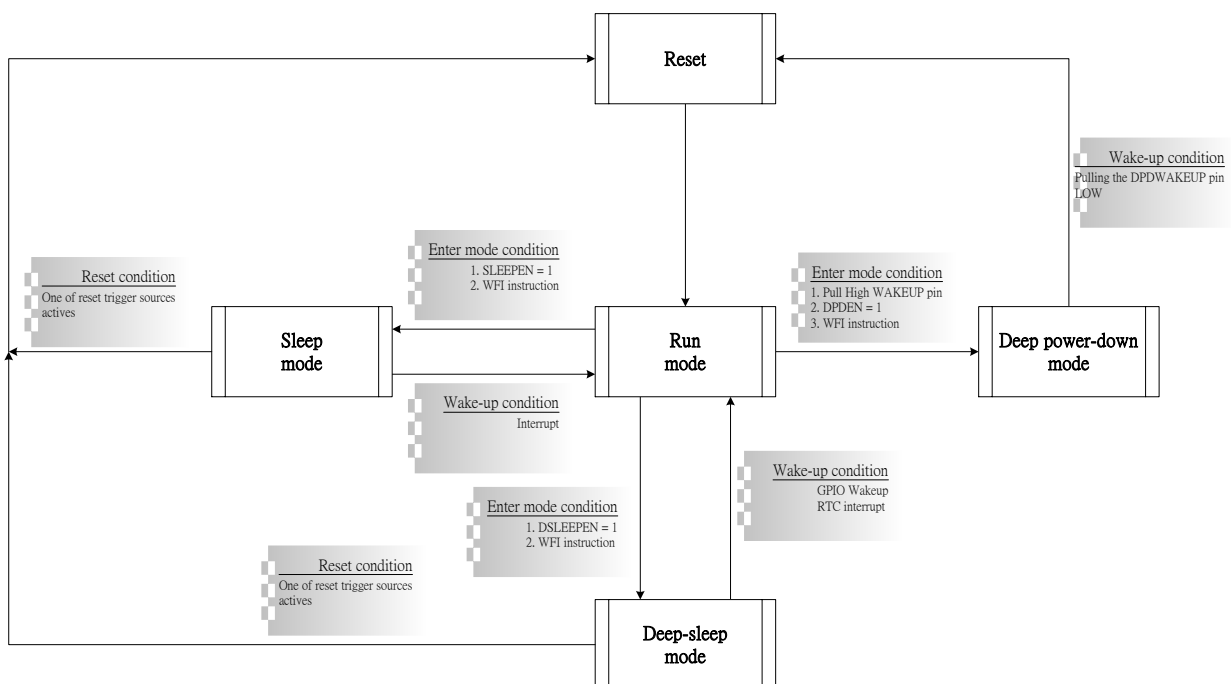
Follow these steps to wake up the chip from Deep power-down mode:

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 GPIO interrupt to the ARM core. The all GPIO port pins 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.

## 4.5 STATE MACHINE OF PMU



## 4.6 OPERATION MODE COMPARSION TABLE

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

\*\*\*

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



## 4.7 PMU REGISTERS

Base Address: 0x4003 2000

### 4.7.1 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     |

### 4.7.2 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: 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.**

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

- \* **Note: HW will switch P1.7 and P1.8 to Microphone differential input if SEL\_MIC=1 in ADC\_SET23 register. Setting SEL\_MIC=0 before P1.7 and P1.8 as GPIO function.**
- \* **Note: P0.14 is the input pin only, please don't set it to the output function in GPIO0\_MODE register.**

## 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<sub>n</sub>\_DATA) (n=0,1,2,3)

Address offset: 0x00

| Bit   | Name       | Description  | Attribute | Reset |
|-------|------------|--|-----------|-------|
| 31:16 | Reserved   |  | R         | 0     |
| 15:0  | DATA[15:0] | Input data (read) or output data (write) for Pn.0 to Pn.15 | R/W       | 0     |

### 5.3.2 GPIO Port n Mode register (GPIO<sub>n</sub>\_MODE) (n=0,1,2,3)

Address offset: 0x04

\* **Note: HW will switch I/O Mode directly when Specific function (Peripheral) is enabled, not through GPIO<sub>n</sub>\_MODE register.**

| Bit   | Name       | Description  | Attribute | Reset |
|-------|------------|--|-----------|-------|
| 31:16 | Reserved   |  | R         | 0     |
| 15:0  | MODE[15:0] | Selects pin x as input or output (x = 0 to 15, <b>n = 0 and x = 14 is input only</b> )<br>0: Pn.x is configured as input<br>1: Pn.x is configured as output. | R/W       | 0     |

\* **Note: HW will switch P1.7 and P1.8 to Microphone differential input if SEL\_MIC=1 in ADC\_SET23 register. Setting SEL\_MIC=0 before P1.7 and P1.8 as GPIO function.**

\* **Note: P0.14 is the input pin only, please don't set it to the output function in GPIO0\_MODE register.**

### 5.3.3 GPIO Port n Configuration register (GPIO<sub>n</sub>\_CFG) (n=0,1,2,3)

Address offset: 0x08  
 Reset value: 0xAAAAAAAA

\* **Note: HW will switch I/O Mode directly when Specific function (Peripheral) is enabled, not through GPIO<sub>n</sub>\_MODE register.**

| Bit   | Name       | Description   | Attribute | Reset |
|-------|------------|---|-----------|-------|
| 31:30 | CFG15[1:0] | Configuration of Pn.15<br>00: Pull-up resistor enabled.<br>01: Pull-down resistor enabled.<br>10: Inactive (no pull-down/pull-up resistor enabled).<br>11: Repeater mode. | R/W       | 10b   |
| 29:28 | CFG14[1:0] | Configuration of Pn.14<br>00: Pull-up resistor enabled.<br>01: Pull-down resistor enabled.  | R/W       | 10b   |

|              |            |   |     |     |
|--------------|------------|---|-----|-----|
|              |            | 10: Inactive (no pull-down/pull-up resistor enabled).<br>11: Repeater mode.   |     |     |
| <b>27:26</b> | CFG13[1:0] | Configuration of Pn.13<br>00: Pull-up resistor enabled.<br>01: Pull-down resistor enabled.<br>10: Inactive (no pull-down/pull-up resistor enabled).<br>11: Repeater mode. | R/W | 10b |
| <b>25:24</b> | CFG12[1:0] | Configuration of Pn.12<br>00: Pull-up resistor enabled.<br>01: Pull-down resistor enabled.<br>10: Inactive (no pull-down/pull-up resistor enabled).<br>11: Repeater mode. | R/W | 10b |
| <b>23:22</b> | CFG11[1:0] | Configuration of Pn.11<br>00: Pull-up resistor enabled.<br>01: Pull-down resistor enabled.<br>10: Inactive (no pull-down/pull-up resistor enabled).<br>11: Repeater mode. | R/W | 10b |
| <b>21:20</b> | CFG10[1:0] | Configuration of Pn.10<br>00: Pull-up resistor enabled.<br>01: Pull-down resistor enabled.<br>10: Inactive (no pull-down/pull-up resistor enabled).<br>11: Repeater mode. | R/W | 10b |
| <b>19:18</b> | CFG9[1:0]  | Configuration of Pn.9<br>00: Pull-up resistor enabled.<br>01: Pull-down resistor enabled.<br>10: Inactive (no pull-down/pull-up resistor enabled).<br>11: Repeater mode.  | R/W | 10b |
| <b>17:16</b> | CFG8[1:0]  | Configuration of Pn.8<br>00: Pull-up resistor enabled.<br>01: Pull-down resistor enabled.<br>10: Inactive (no pull-down/pull-up resistor enabled).<br>11: Repeater mode.  | R/W | 10b |
| <b>15:14</b> | CFG7[1:0]  | Configuration of Pn.7<br>00: Pull-up resistor enabled.<br>01: Pull-down resistor enabled.<br>10: Inactive (no pull-down/pull-up resistor enabled).<br>11: Repeater mode.  | R/W | 10b |
| <b>13:12</b> | CFG6[1:0]  | Configuration of Pn.6<br>00: Pull-up resistor enabled.<br>01: Pull-down resistor enabled.<br>10: Inactive (no pull-down/pull-up resistor enabled).<br>11: Repeater mode.  | R/W | 10b |
| <b>11:10</b> | CFG5[1:0]  | Configuration of Pn.5<br>00: Pull-up resistor enabled.<br>01: Pull-down resistor enabled.<br>10: Inactive (no pull-down/pull-up resistor enabled).<br>11: Repeater mode.  | R/W | 10b |
| <b>9:8</b>   | CFG4[1:0]  | Configuration of Pn.4<br>00: Pull-up resistor enabled.<br>01: Pull-down resistor enabled.<br>10: Inactive (no pull-down/pull-up resistor enabled).<br>11: Repeater mode.  | R/W | 10b |
| <b>7:6</b>   | CFG3[1:0]  | Configuration of Pn.3<br>00: Pull-up resistor enabled.<br>01: Pull-down resistor enabled.<br>10: Inactive (no pull-down/pull-up resistor enabled).<br>11: Repeater mode.  | R/W | 10b |
| <b>5:4</b>   | CFG2[1:0]  | Configuration of Pn.2<br>00: Pull-up resistor enabled.<br>01: Pull-down resistor enabled.<br>10: Inactive (no pull-down/pull-up resistor enabled).<br>11: Repeater mode.  | R/W | 10b |
| <b>3:2</b>   | CFG1[1:0]  | Configuration of Pn.1<br>00: Pull-up resistor enabled.<br>01: Pull-down resistor enabled.<br>10: Inactive (no pull-down/pull-up resistor enabled).<br>11: Repeater mode.  | R/W | 10b |
| <b>1:0</b>   | CFG0[1:0]  | Configuration of Pn.0   | R/W | 10b |

|   |
|---|
| 00: Pull-up resistor enabled.                         |
| 01: Pull-down resistor enabled.                       |
| 10: Inactive (no pull-down/pull-up resistor enabled). |
| 11: Repeater mode.                                    |

- \* **Note: HW will switch P1.7 and P1.8 to Microphone differential input if SEL\_MIC=1 in ADC\_SET23 register. Setting SEL\_MIC=0 before P1.7 and P1.8 as GPIO function.**
- \* **Note: P0.14 is the input pin only, please don't set it to the output function in GPIO0\_MODE register.**

### 5.3.4 GPIO Port n Interrupt Sense register (GPIO<sub>n</sub>\_IS) (n=0,1,2,3)

Address offset: 0x0C

| Bit   | Name     | Description   | Attribute | Reset |
|-------|----------|---|-----------|-------|
| 31:16 | Reserved |   | R         | 0     |
| 15:0  | IS[15:0] | Selects interrupt on pin x as level or edge sensitive (x = 0 to 15).<br>0: Interrupt on Pn.x is configured as edge sensitive.<br>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<sub>n</sub>\_IBS) (n=0,1,2,3)

Address offset: 0x10

| Bit   | Name      | Description  | Attribute | Reset |
|-------|-----------|--|-----------|-------|
| 31:16 | Reserved  |  | R         | 0     |
| 15:0  | IBS[15:0] | Selects interrupt on Pn.x to be triggered on both edges (x = 0 to 15).<br>0: Interrupt on Pn.x is controlled through register GPIO <sub>n</sub> _IEV.<br>1: Both edges on Pn.x trigger an interrupt. | R/W       | 0     |

### 5.3.6 GPIO Port n Interrupt Event register (GPIO<sub>n</sub>\_IEV) (n=0,1,2,3)

Address offset: 0x14

| Bit   | Name      | Description   | Attribute | Reset |
|-------|-----------|---|-----------|-------|
| 31:16 | Reserved  |   | R         | 0     |
| 15:0  | IEV[15:0] | Selects interrupt on pin x to be triggered rising or falling edges (x = 0 to 15).<br>0: Depending on setting in register GPIO <sub>n</sub> _IS, Rising edges or HIGH level on Pn.x trigger an interrupt.<br>1: Depending on setting in register GPIO <sub>n</sub> _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<sub>n</sub>\_IE) (n=0,1,2,3)

Address offset: 0x18

Bits set to HIGH in the GPIO<sub>n</sub>\_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 |
|-------|----------|--|-----------|-------|
| 31:16 | Reserved |  | R         | 0     |
| 15:0  | IE[15:0] | Selects interrupt on pin x to be enabled (x = 0 to 15).<br>0: Disable Interrupt on Pn.x<br>1: Enable Interrupt on Pn.x | R/W       | 0     |

### 5.3.8 GPIO Port n Raw Interrupt Status register (GPIO<sub>n</sub>\_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<sub>n</sub>\_IE register is set.

| Bit   | Name     | Description   | Attribute | Reset |
|-------|----------|---|-----------|-------|
| 31:16 | Reserved |   | R         | 0     |
| 15:0  | IF[15:0] | GPIO raw interrupt flag (x = 0 to 15).<br>0: No interrupt on Pn.x<br>1: Interrupt requirements met on Pn.x. | R         | 0     |

### 5.3.9 GPIO Port n Interrupt Clear register (GPIO<sub>n</sub>\_IC) (n=0,1,2,3)

Address offset: 0x20

| Bit   | Name     | Description   | Attribute | Reset |
|-------|----------|---|-----------|-------|
| 31:16 | Reserved |   | R         | 0     |
| 15:0  | IC[15:0] | Selects interrupt flag on pin x to be cleared (x = 0 to 15).<br>0: No effect<br>1: Clear interrupt flag on Pn.x | W         | 0     |

### 5.3.10 GPIO Port n Bits Set Operation register (GPIO<sub>n</sub>\_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<sub>n</sub>\_BSET register is set.

| Bit   | Name       | Description  | Attribute | Reset |
|-------|------------|--|-----------|-------|
| 31:16 | Reserved   |  | R         | 0     |
| 15:0  | BSET[15:0] | Bit Set enable (x = 0 to 15)<br>0: No effect on Pn.x<br>1: Set Pn.x to "1" | W         | 0     |

### 5.3.11 GPIO Port n Bits Clear Operation register (GPIO<sub>n</sub>\_BCLR) (n=0,1,2,3)

Address offset: 0x28

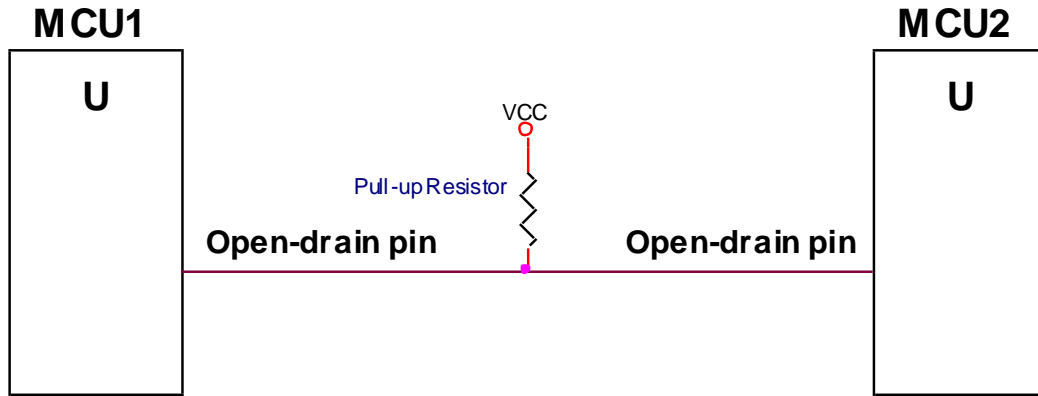
In order for SW to clear GPIO bits without affecting any other pins in a single write operation, the GPIO bit is cleared if the corresponding bit in this register is set.

| Bit   | Name       | Description  | Attribute | Reset |
|-------|------------|--|-----------|-------|
| 31:16 | Reserved   |  | R         | 0     |
| 15:0  | BCLR[15:0] | Bit clear enable (x = 0 to 15)<br>0: No effect on Pn.x<br>1: Clear Pn.x. | W         | 0     |

### 5.3.12 GPIO Port n Open-Drain Control register (GPIO<sub>n</sub>\_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.

| Bit   | Name     | Description   | Attribute | Reset |
|-------|----------|---|-----------|-------|
| 31:16 | Reserved |   | R         | 0     |
| 15    | Pn15OC   | $n = 3$<br>P3.15 open-drain control bit.<br>0: Disable<br>1: Enable. HW set P3.15 as output mode automatically. | R/W       | 0     |
|       |          | $n=0\sim 2$ Reserved  | R         |       |
| 14    | Pn14OC   | $n = 3$<br>P3.14 open-drain control bit.<br>0: Disable<br>1: Enable. HW set P3.14 as output mode automatically. | R/W       | 0     |
|       |          | $n=0\sim 2$ Reserved  | R         |       |
| 13    | Pn13OC   | $n = 3$<br>P3.13 open-drain control bit.<br>0: Disable<br>1: Enable. HW set P3.13 as output mode automatically. | R/W       | 0     |
|       |          | $n=0\sim 2$ Reserved  | R         |       |
| 12    | Pn12OC   | $n = 3$<br>P3.12 open-drain control bit.<br>0: Disable<br>1: Enable. HW set P3.12 as output mode automatically. | R/W       | 0     |
|       |          | $n=0\sim 2$ Reserved  | R         |       |
| 11:4  | Reserved |   | R         | 0     |
| 3     | Pn3OC    | $n = 0$<br>P0.3 open-drain control bit.<br>0: Disable<br>1: Enable. HW set P0.3 as output mode automatically.   | R/W       | 0     |
|       |          | $n=1\sim 3$ Reserved  | R         |       |
| 2     | Pn2OC    | $n = 0$<br>P0.2 open-drain control bit.<br>0: Disable<br>1: Enable. HW set P0.2 as output mode automatically.   | R/W       | 0     |
|       |          | $n=1\sim 3$ Reserved  | R         |       |
| 1     | Pn1OC    | $n = 0$<br>P0.1 open-drain control bit.<br>0: Disable<br>1: Enable. HW set P0.1 as output mode automatically.   | R/W       | 0     |
|       |          | $n=1\sim 3$ Reserved  | R         |       |
| 0     | Pn0OC    | $n = 0$<br>P0.0 open-drain control bit.<br>0: Disable<br>1: Enable. HW set P0.0 as output mode automatically.   | R/W       | 0     |
|       |          | $n=1\sim 3$ Reserved  | R         |       |

# 6 16-BIT TIMER WITH CAPTURE FUNCTION

## 6.1 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, one match register can be used to provide a single-edge controlled PWM output on the match output pins.

## 6.2 FEATURES

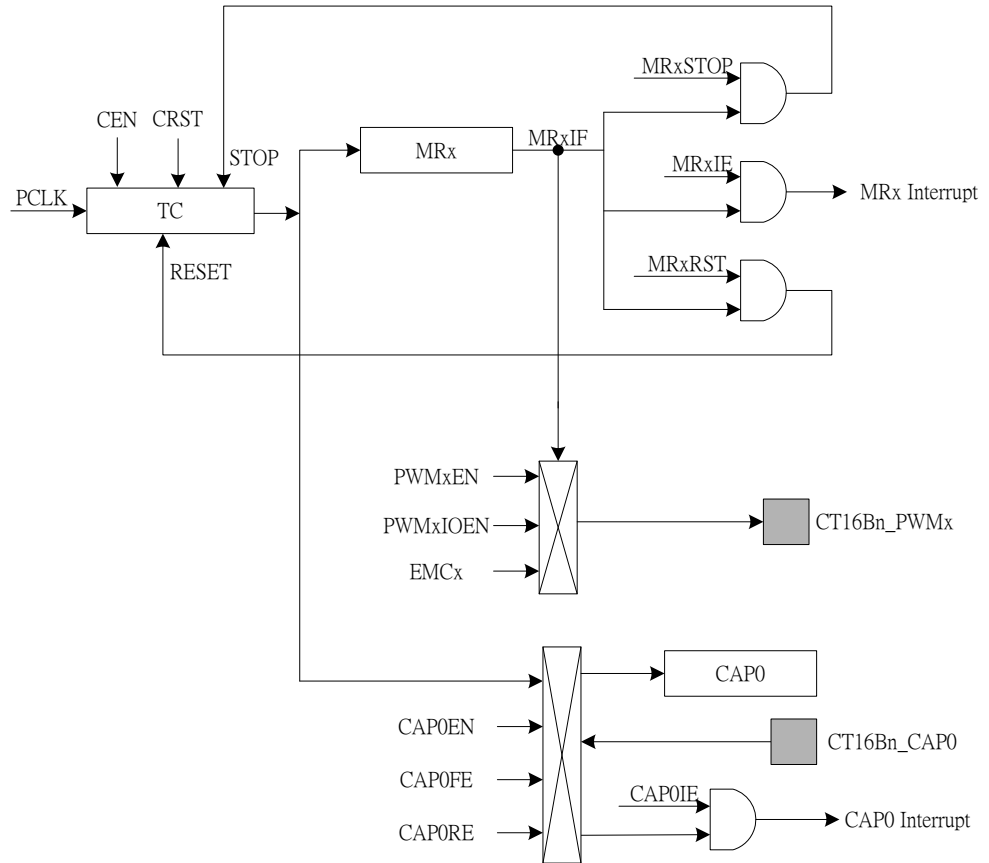
- Two 16-bit counter/timers.
- Counter or timer operation
- Two 16-bit capture channels that can take a snapshot of the timer value when an input signal transitions. A capture event may also optionally generate an interrupt.
- The timer value may be configured to be cleared on a designated capture event. 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.
- 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.
- One (CT16B0 or CT16B1) PWM output corresponding to match register with the following capabilities:
  - Set LOW on match.
  - Set HIGH on match.
  - Toggle on match.
  - Do nothing on match.
- For each timer, one match register (MR0) can be configured as PWM allowing to use one match output as single edge controlled PWM output.

## 6.3 PIN DESCRIPTION

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

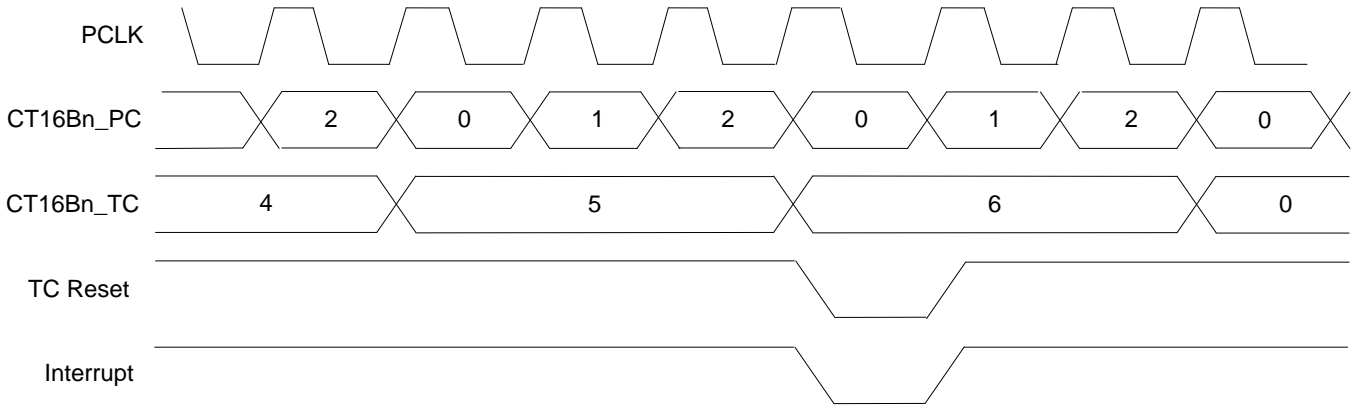


## 6.4 BLOCK DIAGRAM

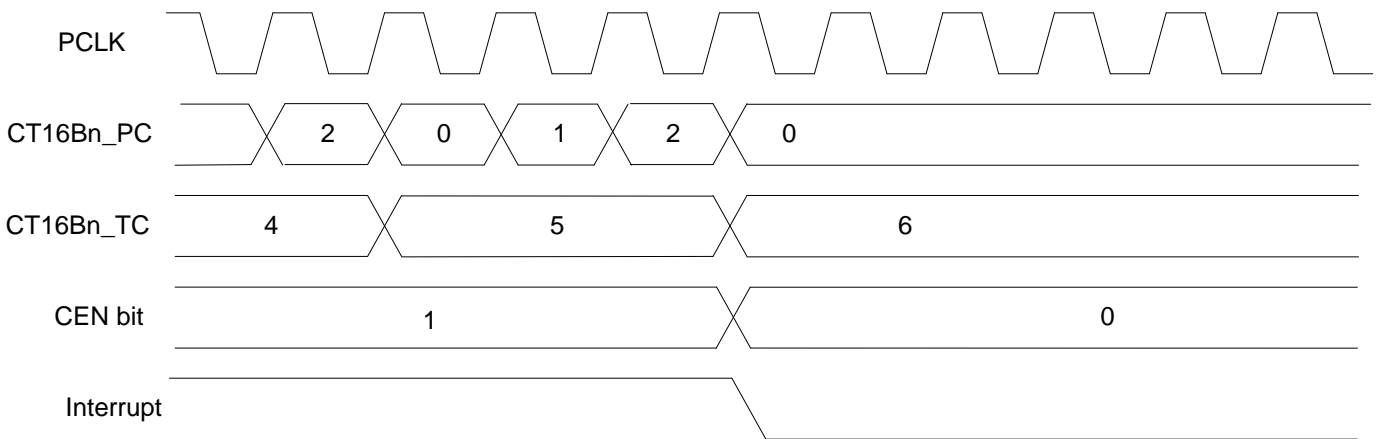


## 6.5 TIMER OPERATION

The following figure shows a timer configured to reset the count and generate an interrupt on match. The [CT16Bn\\_PRE](#) register is set to 2, and the [CT16Bn\\_MRx](#) register is set to 6. At the end of the timer cycle where the match occurs, the timer count is reset. The interrupt indicating that a match occurred is generated after the timer reached the match value.

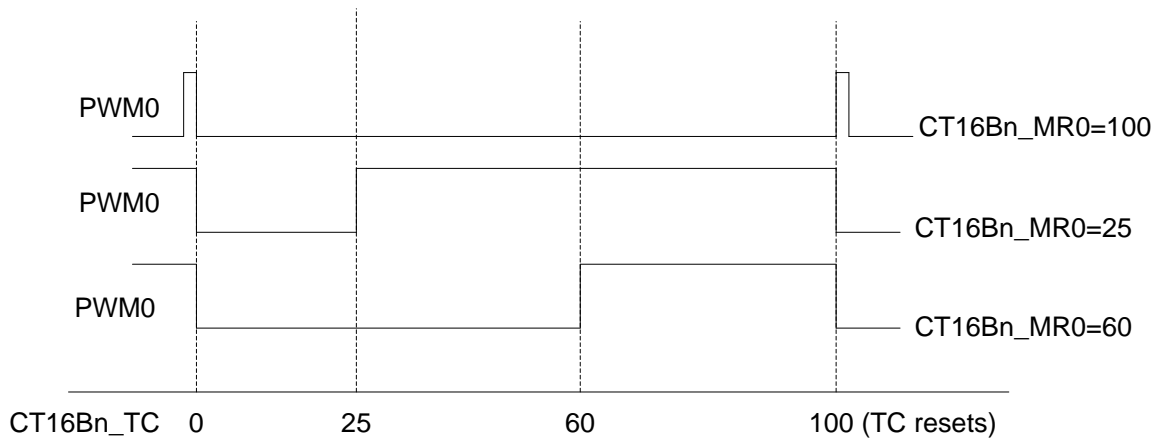


The following figure shows a timer configured to stop and generate an interrupt on match. The [CT16Bn\\_PRE](#) register is set to 2, and the [CT16Bn\\_MRx](#) register is set to 6. After the timer reaches the match value, the CEN bit in [CT16Bn\\_TMRCTRL](#) register is cleared, and the interrupt indicating that a match occurred is generated.



## 6.6 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.

## 6.7 CT16Bn REGISTERS

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

### 6.7.1 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.<br>0: Disable counter reset.<br>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<br>0: Disable Counter.<br>1: Enable Timer Counter for counting.   | R/W       | 0     |

### 6.7.2 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     |

### 6.7.3 CT16Bn Prescale register (CT16Bn\_PRE) (n=0,1)

Address Offset: 0x08

| Bit   | Name     | Description         | Attribute | Reset |
|-------|----------|---------------------|-----------|-------|
| 31:16 | Reserved |                     | R         | 0     |
| 15:0  | PR[15:0] | Prescale max value. | R/W       | 0     |

### 6.7.4 CT16Bn Prescale Counter register (CT16Bn\_PC) (n=0,1)

Address Offset: 0x0C

The 16-bit Prescale Counter controls division of PCLK by some constant value before it is applied to the Timer Counter. This allows control of the relationship between the resolution of the timer and the maximum time before the timer overflows. The Prescale Counter is incremented on every PCLK. When it reaches the value stored in the Prescale Register, the Timer Counter is incremented, and the Prescale Counter is reset on the next PCLK. This causes the TC to increment on every PCLK when PR = 0, every 2 PCLKs when PR = 1, etc.

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

### 6.7.5 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 can not exceed one half of the PCLK clock. Consequently, the duration of the HIGH/LOW levels on the same CAP input in this case can not 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.<br>In counter mode (when CTM[1:0] are not 00), these bits select which CAP0 pin is sampled for clocking.<br>00: CT16Bn_CAP0<br>Other: Reserved.  | R/W       | 0     |
| 1:0  | CTM[1:0] | Counter/Timer Mode.<br>This field selects which rising PCLK edges can clear PC and increment Timer Counter (TC).<br>00: Timer Mode: every rising PCLK edge<br>01: Counter Mode: TC is incremented on rising edges on the CAP0 input selected by CIS bits.<br>10: Counter Mode: TC is incremented on falling edges on the CAP0 input selected by CIS bits.<br>11: Counter Mode: TC is incremented on both edges on the CAP0 input selected by CIS bits. | R/W       | 0     |

### 6.7.6 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.<br>0: Disable<br>1: Enable | R/W       | 0     |
| 10    | MR3RST   | Enable reset TC when MR3 matches TC.<br>0: Disable<br>1: Enable                                  | R/W       | 0     |
| 9     | MR3IE    | Enable generating an interrupt when MR3 matches the value in the TC.<br>0: Disable<br>1: Enable  | R/W       | 0     |
| 8     | MR2STOP  | Stop MR2: TC will stop and CEN bit will be cleared if MR2 matches TC.<br>0: Disable<br>1: Enable | R/W       | 0     |

|   |         |  |     |   |
|---|---------|--|-----|---|
| 7 | MR2RST  | Enable reset TC when MR2 matches TC.<br>0: Disable<br>1: Enable                                  | R/W | 0 |
| 6 | MR2IE   | Enable generating an interrupt when MR2 matches the value in the TC.<br>0: Disable<br>1: Enable  | R/W | 0 |
| 5 | MR1STOP | Stop MR1: TC will stop and CEN bit will be cleared if MR1 matches TC.<br>0: Disable<br>1: Enable | R/W | 0 |
| 4 | MR1RST  | Enable reset TC when MR1 matches TC.<br>0: Disable<br>1: Enable                                  | R/W | 0 |
| 3 | MR1IE   | Enable generating an interrupt when MR1 matches the value in the TC.<br>0: Disable<br>1: Enable  | R/W | 0 |
| 2 | MR0STOP | Stop MR0: TC will stop and CEN bit will be cleared if MR0 matches TC.<br>0: Disable<br>1: Enable | R/W | 0 |
| 1 | MR0RST  | Enable reset TC when MR0 matches TC.<br>0: Disable<br>1: Enable                                  | R/W | 0 |
| 0 | MR0IE   | Enable generating an interrupt when MR0 matches the value in the TC.<br>0: Disable<br>1: Enable  | R/W | 0 |

### 6.7.7 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     |

### 6.7.8 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 |
|------|----------|---|-----------|-------|
| 31:7 | Reserved |   | R         | 0     |
| 6:5  | CAP0EN   | Capture 0 function enable bit<br>0: Disable<br>1: Enable Capture 0 function for external Capture pin.<br>2~3: Reserved.                     | R/W       | 0     |
| 4    | CAP0IE   | Interrupt on CT16Bn_CAP0 signal event: a CAP0 load due to a CT16Bn_CAP0 signal event will generate an interrupt.<br>0: Disable<br>1: Enable | R/W       | 0     |
| 3:2  | CAP0FE   | Capture/Reset on CT16Bn_CAP0 signal falling edge.<br>0: Disable<br>1: Enable a sequence of 1 then 0 on CT16Bn_CAP0 signal will cause        | R/W       | 0     |

|            |        |   |     |   |
|------------|--------|---|-----|---|
|            |        | CAP0 to be loaded with the contents of TC.<br>2: Enable a sequence of 1 then 0 on CT16Bn_CAP0 signal will reset the TC.<br>3: Reserved.   |     |   |
| <b>1:0</b> | CAP0RE | Capture/Reset on CT16Bn_CAP0 signal rising edge.<br>0: Disable<br>1: Enable a sequence of 0 then 1 on CT16Bn_CAP0 signal will cause CAP0 to be loaded with the contents of TC.<br>2: Enable a sequence of 0 then 1 on CT16Bn_CAP0 signal will reset the TC.<br>3: Reserved. | R/W | 0 |

### 6.7.9 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 |
|--------------|------------|-----------------------------|-----------|-------|
| <b>31:16</b> | Reserved   |                             | R         | 0     |
| <b>15:0</b>  | CAP0[15:0] | Timer counter capture value | R         | 0     |

### 6.7.10 CT16Bn External Match register (CT16Bn\_EM) (n=0,1)

Address Offset: 0x30

The External Match register provides both control and status of CT16Bn\_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 |
|-------------|-----------|---|-----------|-------|
| <b>31:6</b> | Reserved  |   | R         | 0     |
| <b>5:4</b>  | EMC0[1:0] | Determines the functionality of CT16Bn_PWM0.<br>00: Do Nothing.<br>01: CT16Bn_PWM0 pin is LOW<br>10: CT16Bn_PWM0 pin is HIGH<br>11: Toggle CT16Bn_PWM0. | R/W       | 0     |
| <b>3:1</b>  | Reserved  |   | R         | 0     |
| <b>0</b>    | 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     |

### 6.7.11 CT16Bn PWM Control register (CT16Bn\_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 in-dependently set to perform either as PWM output or as match output whose function is controlled by [CT16Bn\\_EM](#) register.

For each timer, a maximum of three single edge controlled PWM outputs can be selected on the CT16Bn\_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 |
|--------------|----------|--|-----------|-------|
| <b>31:21</b> | Reserved |  | R         | 0     |
| <b>20</b>    | PWM0IOEN | CT16Bn_PWM0/GPIO selection bit<br>0: CT16Bn_PWM0 pin act as GPIO<br>1: CT16Bn_PWM0 pin act as match output, and output signal depends on | R/W       | 0     |

|             |          |  |     |   |
|-------------|----------|--|-----|---|
|             |          | PWM0EN bit.  |     |   |
| <b>19:1</b> | Reserved |  | R   | 0 |
| <b>0</b>    | PWM0EN   | PWM0 enable<br>0: CT16Bn_PWM0 is controlled by EM0.<br>1: PWM mode is enabled for CT16Bn_PWM0. | R/W | 0 |

### 6.7.12 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 |
|-------------|----------|--|-----------|-------|
| <b>31:5</b> | Reserved |  | R         | 0     |
| <b>4</b>    | CAP0IF   | Interrupt flag for capture channel 0.<br>0: No interrupt on CAP0<br>1: Interrupt requirements met on CAP0.                     | R         | 0     |
| <b>3</b>    | MR3IF    | Interrupt flag for match channel 3.<br>0: No interrupt on match channel 3<br>1: Interrupt requirements met on match channel 3. | R         | 0     |
| <b>2</b>    | MR2IF    | Interrupt flag for match channel 2.<br>0: No interrupt on match channel 2<br>1: Interrupt requirements met on match channel 2. | R         | 0     |
| <b>1</b>    | MR1IF    | Interrupt flag for match channel 1.<br>0: No interrupt on match channel 1<br>1: Interrupt requirements met on match channel 1. | R         | 0     |
| <b>0</b>    | MR0IF    | Interrupt flag for match channel 0.<br>0: No interrupt on match channel 0<br>1: Interrupt requirements met on match channel 0. | R         | 0     |

### 6.7.13 CT16Bn Timer Interrupt Clear register (CT16Bn\_IC) (n=0,1)

Address Offset: 0x3C

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



# 7 32-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 two match registers can be used to provide a single-edge controlled PWM output on the match output pins.

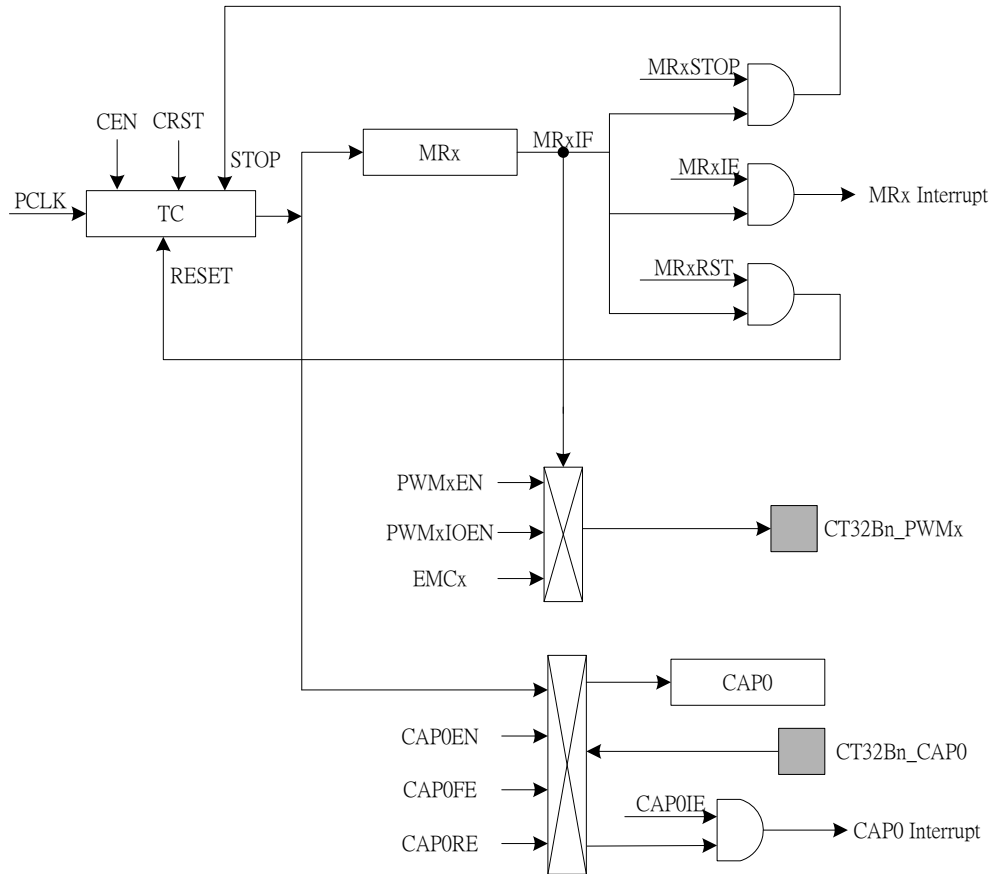
## 7.2 FEATURES

- 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.
- 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.
- Up to two (CT32B0 or CT32B1) 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.
- For each timer, up to two match registers (MR0~MR1) can be configured as PWM allowing to use up to two match outputs as single edge controlled PWM outputs.

## 7.3 PIN DESCRIPTION

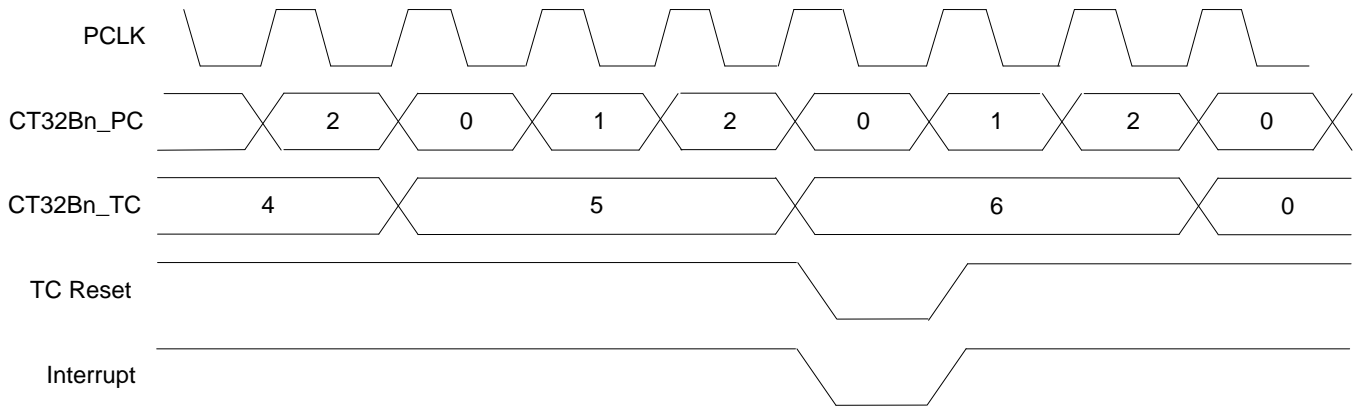
| Pin Name    | Type | Description                           | GPIO Configuration                |
|-------------|------|---------------------------------------|-----------------------------------|
| CT32Bn_CAP0 | I    | Capture channel input 0               | Depends on GPIO <sub>n</sub> _CFG |
| CT32Bn_PWMx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

## 7.4 BLOCK DIAGRAM

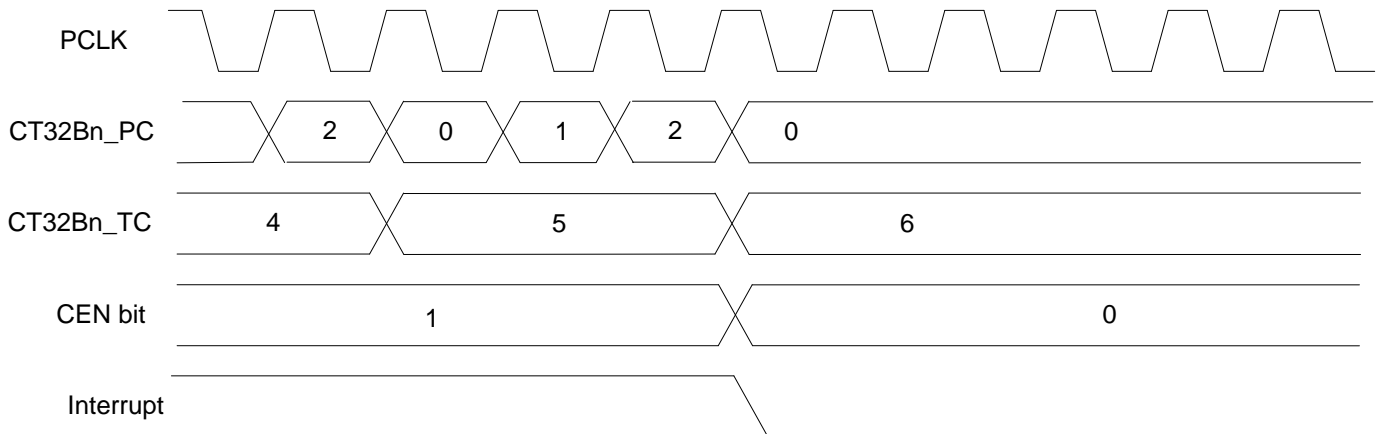


## 7.5 TIMER OPERATION

The following figure shows a timer configured to reset the count and generate an interrupt on match. The [CT32Bn\\_PRE](#) register is set to 2, and 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.

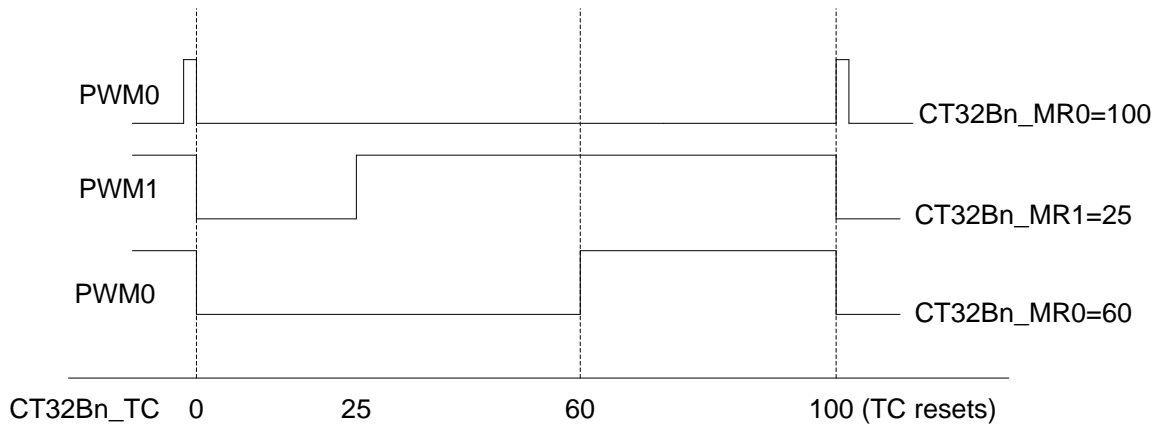


The following figure shows a timer configured to stop and generate an interrupt on match. The [CT32Bn\\_PRE](#) register is set to 2, and the [CT32Bn\\_MRx](#) register is set to 6. In the next clock after the timer reaches the match value, the CEN bit in [CT32Bn\\_TMRCTRL](#) register is cleared, and the interrupt indicating that a match occurred is generated.



## 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 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.

## 7.7 CT32Bn REGISTERS

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

### 7.7.1 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.<br>0: Disable counter reset.<br>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<br>0: Disable Counter.<br>1: Enable Timer Counter for counting.   | R/W       | 0     |

### 7.7.2 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     |

### 7.7.3 CT32Bn Prescale register (CT32Bn\_PRE) (n=0,1)

Address Offset: 0x08

| Bit  | Name     | Description         | Attribute | Reset |
|------|----------|---------------------|-----------|-------|
| 31:0 | PR[31:0] | Prescale max value. | R/W       | 0     |

### 7.7.4 CT32Bn Prescale Counter register (CT32Bn\_PC) (n=0,1)

Address Offset: 0x0C

The 32-bit Prescale Counter controls division of PCLK by some constant value before it is applied to the Timer Counter. This allows control of the relationship between the resolution of the timer and the maximum time before the timer overflows. The Prescale Counter is incremented on every PCLK. When it reaches the value stored in the Prescale Register, the Timer Counter is incremented, and the Prescale Counter is reset on the next PCLK. This causes the TC to increment on every PCLK when PR = 0, every 2 PCLKs when PR = 1, etc.

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

### 7.7.5 CT32Bn Count Control register (CT32Bn\_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 can not exceed one half of the PCLK clock. Consequently, the duration of the HIGH/LOW levels on the same CAP input in this case can not 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.<br>In counter mode (when CTM[1:0] are not 00), these bits select which CAP pin is sampled for clocking.<br>00: CT32Bn_CAP0<br>Other: Reserved.  | R/W       | 0     |
| 1:0  | CTM[1:0] | Counter/Timer Mode.<br>This field selects which rising PCLK edges can clear PC and increment Timer Counter (TC).<br>00: Timer Mode: every rising PCLK edge<br>01: Counter Mode: TC is incremented on rising edges on the CAP input selected by CIS bits.<br>10: Counter Mode: TC is incremented on falling edges on the CAP input selected by CIS bits.<br>11: Counter Mode: TC is incremented on both edges on the CAP input selected by CIS bits. | R/W       | 0     |

### 7.7.6 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.<br>0: Disable<br>1: Enable | R/W       | 0     |
| 10    | MR3RST   | Enable reset TC when MR3 matches TC.<br>0: Disable<br>1: Enable                                  | R/W       | 0     |
| 9     | MR3IE    | Enable generating an interrupt when MR3 matches the value in the TC.<br>0: Disable<br>1: Enable  | R/W       | 0     |
| 8     | MR2STOP  | Stop MR2: TC will stop and CEN bit will be cleared if MR2 matches TC.<br>0: Disable<br>1: Enable | R/W       | 0     |
| 7     | MR2RST   | Enable reset TC when MR2 matches TC.<br>0: Disable<br>1: Enable                                  | R/W       | 0     |
| 6     | MR2IE    | Enable generating an interrupt when MR2 matches the value in the TC.<br>0: Disable<br>1: Enable  | R/W       | 0     |
| 5     | MR1STOP  | Stop MR1: TC will stop and CEN bit will be cleared if MR1 matches TC.<br>0: Disable<br>1: Enable | R/W       | 0     |
| 4     | MR1RST   | Enable reset TC when MR1 matches TC.   | R/W       | 0     |

|          |         |  |     |   |
|----------|---------|--|-----|---|
|          |         | 0: Disable<br>1: Enable  |     |   |
| <b>3</b> | MR1IE   | Enable generating an interrupt when MR1 matches the value in the TC.<br>0: Disable<br>1: Enable  | R/W | 0 |
| <b>2</b> | MR0STOP | Stop MR0: TC will stop and CEN bit will be cleared if MR0 matches TC.<br>0: Disable<br>1: Enable | R/W | 0 |
| <b>1</b> | MR0RST  | Enable reset TC when MR0 matches TC.<br>0: Disable<br>1: Enable                                  | R/W | 0 |
| <b>0</b> | MR0IE   | Enable generating an interrupt when MR0 matches the value in the TC.<br>0: Disable<br>1: Enable  | R/W | 0 |

### 7.7.7 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 |
|-------------|----------|---------------------------|-----------|-------|
| <b>31:0</b> | MR[31:0] | Timer counter match value | R/W       | 0     |

### 7.7.8 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 |
|------|----------|--|-----------|-------|
| 31:7 | Reserved |  | R         | 0     |
| 6:5  | CAP0EN   | Capture 0 function enable bit<br>0: Disable<br>1: Enable Capture 0 function for external Capture pin.<br>2~3: Reserved.  | R/W       | 0     |
| 4    | CAP0IE   | Interrupt on CT32Bn_CAP0 signal event: a CAP0 load due to a CT32Bn_CAP0 signal event will generate an interrupt.<br>0: Disable<br>1: Enable  | R/W       | 0     |
| 3:2  | CAP0FE   | Capture/Reset on CT32Bn_CAP0 signal falling edge.<br>0: Disable<br>1: Enable a sequence of 1 then 0 on CT32Bn_CAP0 signal will cause CAP0 to be loaded with the contents of TC.<br>2: Enable a sequence of 1 then 0 on CT32Bn_CAP0 signal will reset the TC.<br>3: Reserved. | R/W       | 0     |
| 1:0  | CAP0RE   | Capture/Reset on CT32Bn_CAP0 signal rising edge.<br>0: Disable<br>1: Enable a sequence of 0 then 1 on CT32Bn_CAP0 signal will cause CAP0 to be loaded with the contents of TC.<br>2: Enable a sequence of 0 then 1 on CT32Bn_CAP0 signal will reset the TC.<br>3: Reserved.  | R/W       | 0     |

## 7.7.9 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     |

## 7.7.10 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 |
|------|-----------|--|-----------|-------|
| 31:8 | Reserved  |  | R         | 0     |
| 7:6  | EMC1[1:0] | Determines the functionality of CT32Bn_PWM1.<br>00: Do Nothing.<br>01: CT32Bn_PWM1 pin is LOW<br>10: CT32Bn_PWM1 pin is HIGH.<br>11: Toggle CT32Bn_PWM1. | R/W       | 0     |
| 5:4  | EMC0[1:0] | Determines the functionality of CT32Bn_PWM0.<br>00: Do Nothing.<br>01: CT32Bn_PWM0 pin is LOW<br>10: CT32Bn_PWM0 pin is HIGH<br>11: Toggle CT32Bn_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 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     |

## 7.7.11 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 |
|-------|----------|--|-----------|-------|
| 31:22 | Reserved |  | R         | 0     |
| 21    | PWM1IOEN | CT32Bn_PWM1/GPIO selection bit<br>0: CT32Bn_PWM1 pin act as GPIO<br>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<br>0: CT32Bn_PWM0 pin act as GPIO<br>1: CT32Bn_PWM0 pin act as match output, and output signal depends on PWM0EN bit. | R/W       | 0     |
| 19:2  | Reserved |  | R         | 0     |



|          |        |  |     |   |
|----------|--------|--|-----|---|
| <b>1</b> | PWM1EN | PWM1 enable<br>0: CT32Bn_PWM1 is controlled by EM1.<br>1: PWM mode is enabled for CT32Bn_PWM1. | R/W | 0 |
| <b>0</b> | PWM0EN | PWM0 enable<br>0: CT32Bn_PWM0 is controlled by EM0.<br>1: PWM mode is enabled for CT32Bn_PWM0. | R/W | 0 |

### 7.7.12 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 |
|-------------|----------|--|-----------|-------|
| <b>31:5</b> | Reserved |  | R         | 0     |
| <b>4</b>    | CAP0IF   | Interrupt flag for capture channel 0.<br>0: No interrupt on CAP0<br>1: Interrupt requirements met on CAP0.                     | R         | 0     |
| <b>3</b>    | MR3IF    | Interrupt flag for match channel 3.<br>0: No interrupt on match channel 3<br>1: Interrupt requirements met on match channel 3. | R         | 0     |
| <b>2</b>    | MR2IF    | Interrupt flag for match channel 2.<br>0: No interrupt on match channel 2<br>1: Interrupt requirements met on match channel 2. | R         | 0     |
| <b>1</b>    | MR1IF    | Interrupt flag for match channel 1.<br>0: No interrupt on match channel 1<br>1: Interrupt requirements met on match channel 1. | R         | 0     |
| <b>0</b>    | MR0IF    | Interrupt flag for match channel 0.<br>0: No interrupt on match channel 0<br>1: Interrupt requirements met on match channel 0. | R         | 0     |

### 7.7.13 CT32Bn Timer Interrupt Clear register (CT32Bn\_IC) (n=0,1)

Address Offset: 0x3C

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

# 8 WATCHDOG TIMER (WDT)

## 8.1 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 a 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 0 \(SYS1\\_APB0\)](#) 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.

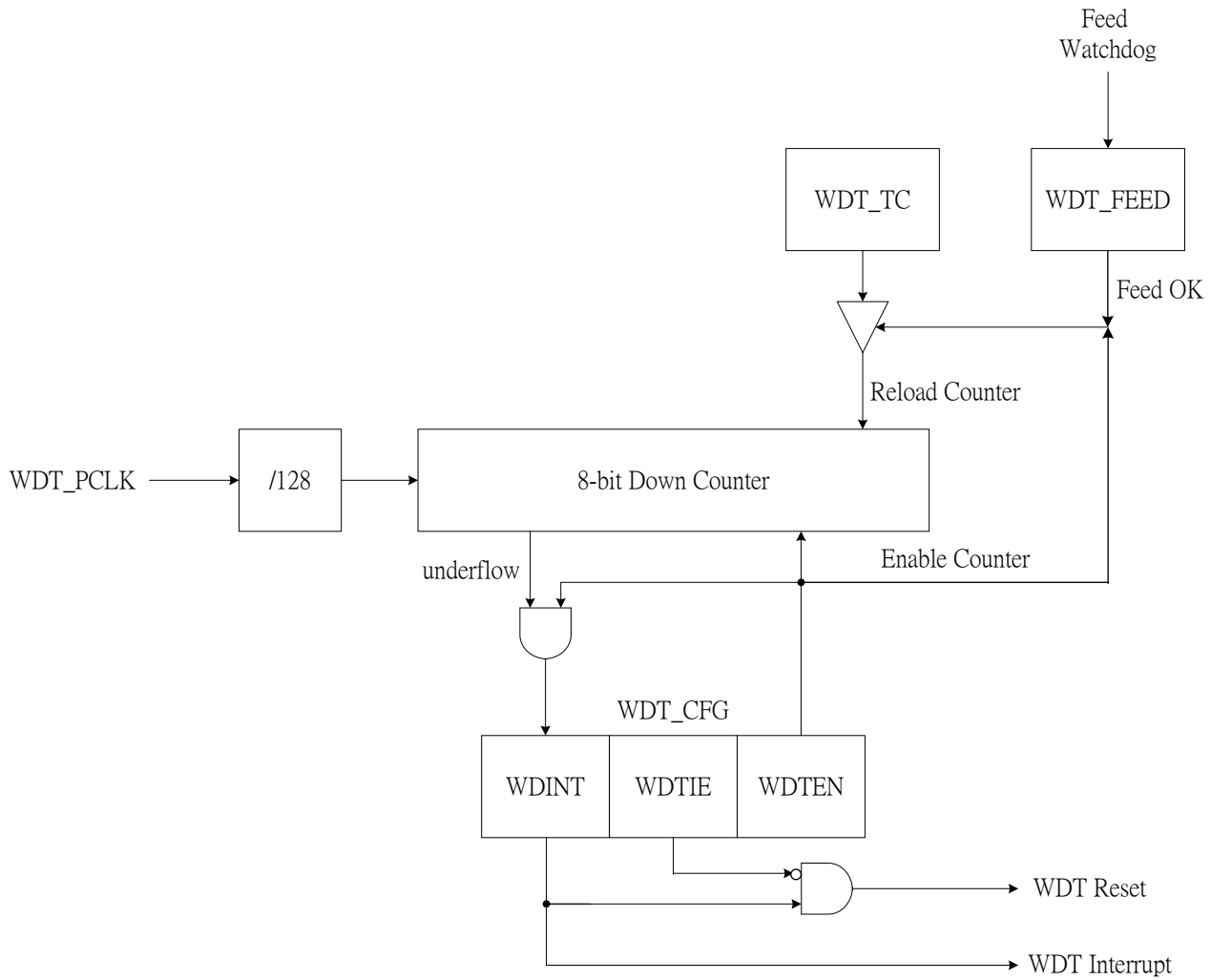
When the watchdog is started by setting the WDTEN in [WDT\\_CFG](#) register, the time constant value is loaded in the watchdog counter and the counter starts counting down. When the Watchdog is in the reset mode and the counter underflows, the CPU will be reset, loading the stack pointer and program 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.

## 8.2 BLOCK DIAGRAM



## 8.3 WDT REGISTERS

Base Address: 0x4001 0000

### 8.3.1 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 |
|-------|----------|--|-----------|-------|
| 31:16 | WDKEY    | Watchdog register key.<br>Read as 0. When writing to the register you must write 0x5AFA to WDKEY, otherwise behaviour of writing to the register is ignored.   | W         | 0     |
| 15:3  | Reserved |  | R         | 0     |
| 2     | WDTINT   | Watchdog interrupt flag<br><Read><br>0: Watchdog does not cause an interrupt.<br>1: Watchdog timeout and causes an interrupt (Only when WDTIE =1).<br><Write><br>0: Clear this flag. SW shall feed Watchdog before clearing.                               | R/W       | 0     |
| 1     | WDTIE    | Watchdog interrupt enable<br>0: Watchdog timeout will cause a chip reset. (Watchdog reset mode)<br>Watchdog counter underflow will reset the MCU, and will clear the WDINT flag.<br>1: Watchdog timeout will cause an interrupt. (Watchdog interrupt mode) | R/W       | 0     |
| 0     | WDTEN    | Watchdog enable<br>0: Disable<br>1: Enable. When enable the watchdog, the WDT_TC value is loaded in the watchdog counter.  | R/W       | 0     |

### 8.3.2 Watchdog Clock Source register (WDT\_CLKSOURCE)

Address Offset: 0x04

| Bit   | Name        | Description  | Attribute | Reset |
|-------|-------------|--|-----------|-------|
| 31:16 | WDKEY       | Watchdog register key.<br>Read as 0. When writing to the register you must write 0x5AFA to WDKEY, otherwise behaviour of writing to the register is ignored. | W         | 0     |
| 15:2  | Reserved    |  | R         | 0     |
| 1:0   | CLKSEL[1:0] | Selected Watchdog clock source.<br>00: IHRC oscillator<br>01: HCLK<br>10: ILRC oscillator<br>11: ELS X'TAL   | R/W       | 0     |

### 8.3.3 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  $T_{WDT\_PCLK} \times 128 \times 1 \sim T_{WDT\_PCLK} \times 128 \times 256$ .

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

| Bit   | Name  | Description  | Attribute | Reset |
|-------|-------|--|-----------|-------|
| 31:16 | WDKEY | Watchdog register key.<br>Read as 0. When writing to the register you must write 0x5AFA to WDKEY, otherwise behaviour of writing to the register is ignored. | W         | 0     |

|             |          |  |     |      |
|-------------|----------|--|-----|------|
| <b>15:8</b> | Reserved |  | R   | 0    |
| <b>7:0</b>  | TC[7:0]  | Watchdog timer constant reload value = TC[7:0]+1<br>0000 0000 : Timer constant = 1<br>0000 0001 : Timer constant = 2<br>.....<br>.....<br>1111 1110 : Timer constant = 255<br>1111 1111 : Timer constant = 256 | R/W | 0xFF |

### 8.3.4 Watchdog Feed register (WDT\_FEED)

Address Offset: 0x0C

| Bit          | Name     | Description  | Attribute | Reset |
|--------------|----------|--|-----------|-------|
| <b>31:16</b> | WDKEY    | Watchdog register key.<br>Read as 0. When writing to the register you must write 0x5AFA to WDKEY, otherwise behaviour of writing to the register is ignored. | W         | 0     |
| <b>15:0</b>  | FV[15:0] | Feed value (Read as 0x0)<br>0x55AA: The watchdog is fed, and the WDT_TC value is reloaded in the watchdog counter.   | W         | 0     |

# 9 REAL-TIME CLOCK (RTC)

## 9.1 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.

## 9.2 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.

## 9.3 FUNCTIONAL DESCRIPTION

### 9.3.1 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.

### 9.3.2 RESET RTC REGISTERS

The RTC\_SECCNTV, RTC\_ALMCNTV, RTC\_SECCNT, and RTC\_ALMCNT registers are reset by “cold” boot or DPDWAKEUP reset.

### 9.3.3 RTC FLAG ASSERTION

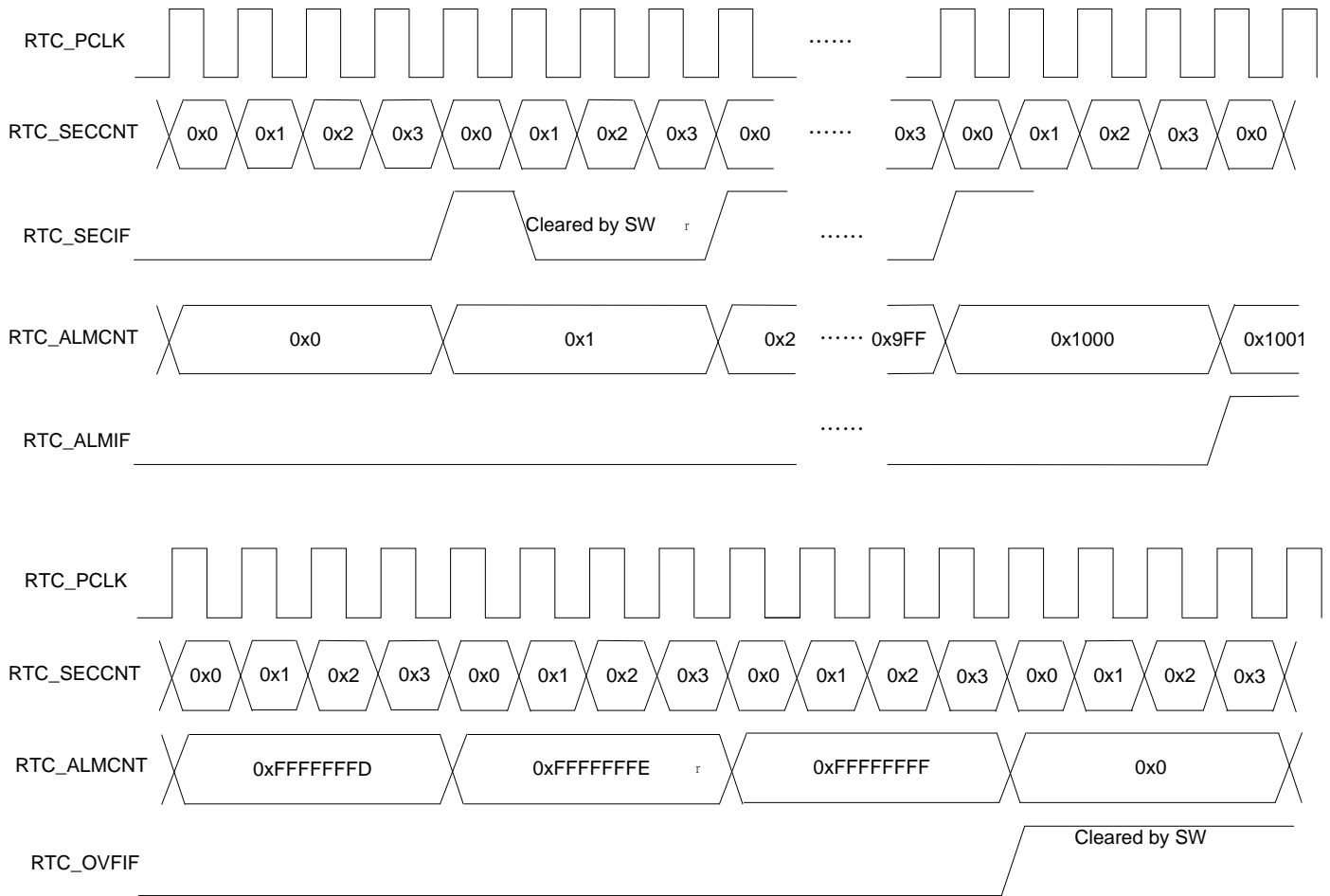
The RTC Second interrupt flag (SECIF) is asserted on each RTC Core clock cycle before the update of the RTC Counter.

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

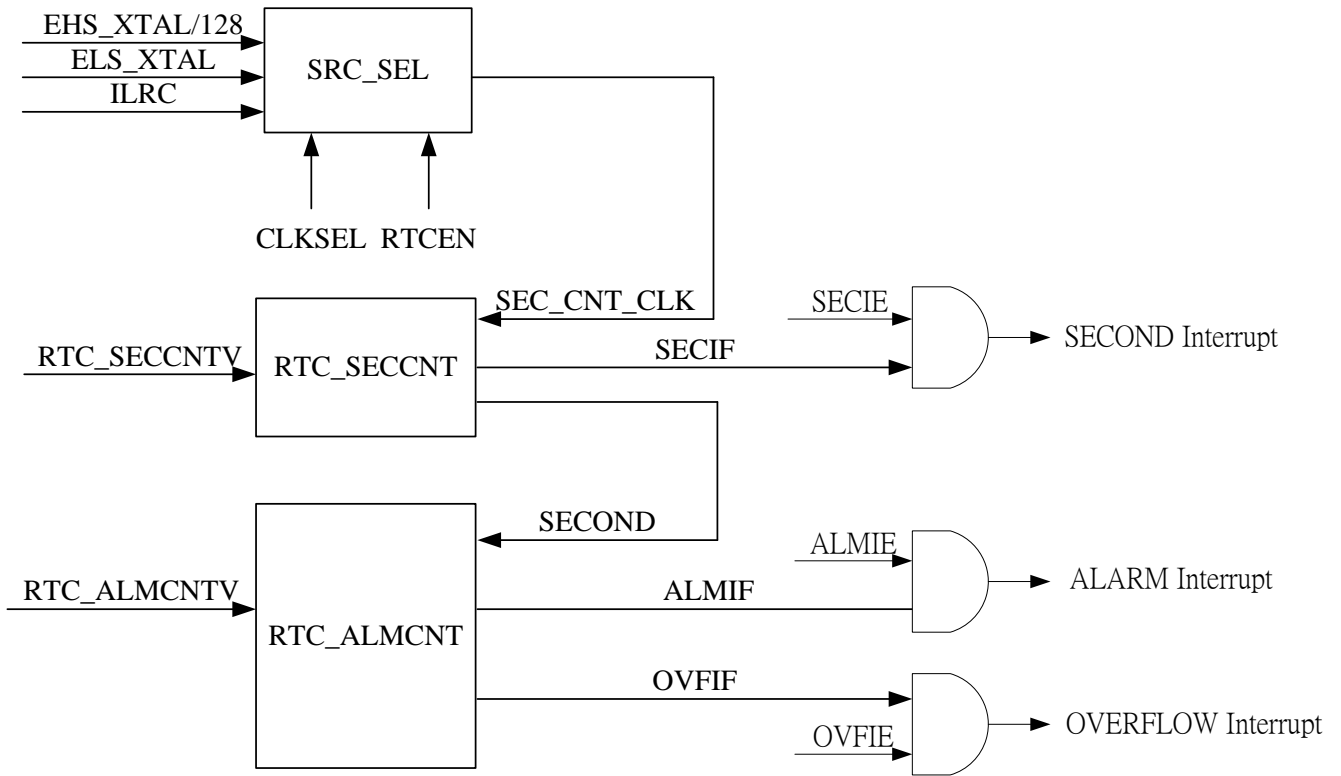
The RTC Alarm interrupt flag (ALMIF) are asserted on the last RTC Core clock cycle before the counter reaches the RTC Alarm counter reload value stored in the Alarm register.

### 9.3.4 RTC OPERATION

The following figure shows the RTC waveform when it is configured with RTC\_SECCNTV=3, RTC\_ALMCNTV=0x1000.



## 9.4 BLOCK DIAGRAM





## 9.5 RTC REGISTERS

Base Address: 0x4001 2000

### 9.5.1 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<br>0: Disable<br>1: Enable. Reset SEC_CNT and ALM_CNT. | R/W       | 0     |

### 9.5.2 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.<br>HW will reset SEC_CNT and ALM_CNT when changing the value.<br>00: ILRC<br>01: ELS X'TAL<br>10: Reserved<br>11: EHS X'TAL clock / 128 | R/W       | 0     |

### 9.5.3 RTC Interrupt Enable register (RTC\_IE)

Address offset: 0x08

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

### 9.5.4 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<br>This bit is set by HW when ALM_CNT overflows (ALM_CNT counts from 0xFFFFFFFF to 0x0). An interrupt is generated if OVFIE=1.<br>0: Overflow not detected<br>1: 32-bit programmable counter overflow occurred. | R         | 0     |

|          |       |   |   |   |
|----------|-------|---|---|---|
| <b>1</b> | ALMIF | Alarm interrupt flag<br>This bit is set by HW when ALM_CNT=ALM_CNTV. An interrupt is generated if ALRIE=1.<br>0: Alarm not detected<br>1: Alarm detected.                         | R | 0 |
| <b>0</b> | SECIF | Second interrupt flag<br>This bit is set by HW when SEC_CNT=SEC_CNTV. An interrupt is generated if SECIE=1.<br>0: Second flag condition not met.<br>1: Second flag condition met. | R | 0 |

### 9.5.5 RTC Interrupt Clear register (RTC\_IC)

Address offset: 0x10

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

### 9.5.6 RTC Second Counter Reload Value register (RTC\_SECNTV)

Address offset: 0x14

Reset value: 0x8000

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

### 9.5.7 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 |
|-------------|-------------|---|-----------|-------|
| <b>31:0</b> | SECNT[31:0] | RTC second counter<br>The current value of the RTC counter. | R         | 0     |

### 9.5.8 RTC Alarm Counter Reload Value register (RTC\_ALMCNTV)

Address offset: 0x1C

Reset value: 0xFFFFFFFF

| Bit         | Name          | Description  | Attribute | Reset      |
|-------------|---------------|--|-----------|------------|
| <b>31:0</b> | ALMCNTV[31:0] | RTC alarm counter reload value.<br>Update this register will reset ALMCNT.<br>The zero value is not recommended, and will be replaced with default value (0xFFFFFFFF) by HW. | R/W       | 0xFFFFFFFF |

## 9.5.9 RTC Alarm Count register (RTC\_ALMCNT)

Address offset: 0x20

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

# 10 SPI/SSP

## 10.1 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.

## 10.2 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 1<sup>st</sup>-phase or 2<sup>nd</sup>-phase controlled register.

## 10.3 PIN DESCRIPTION

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

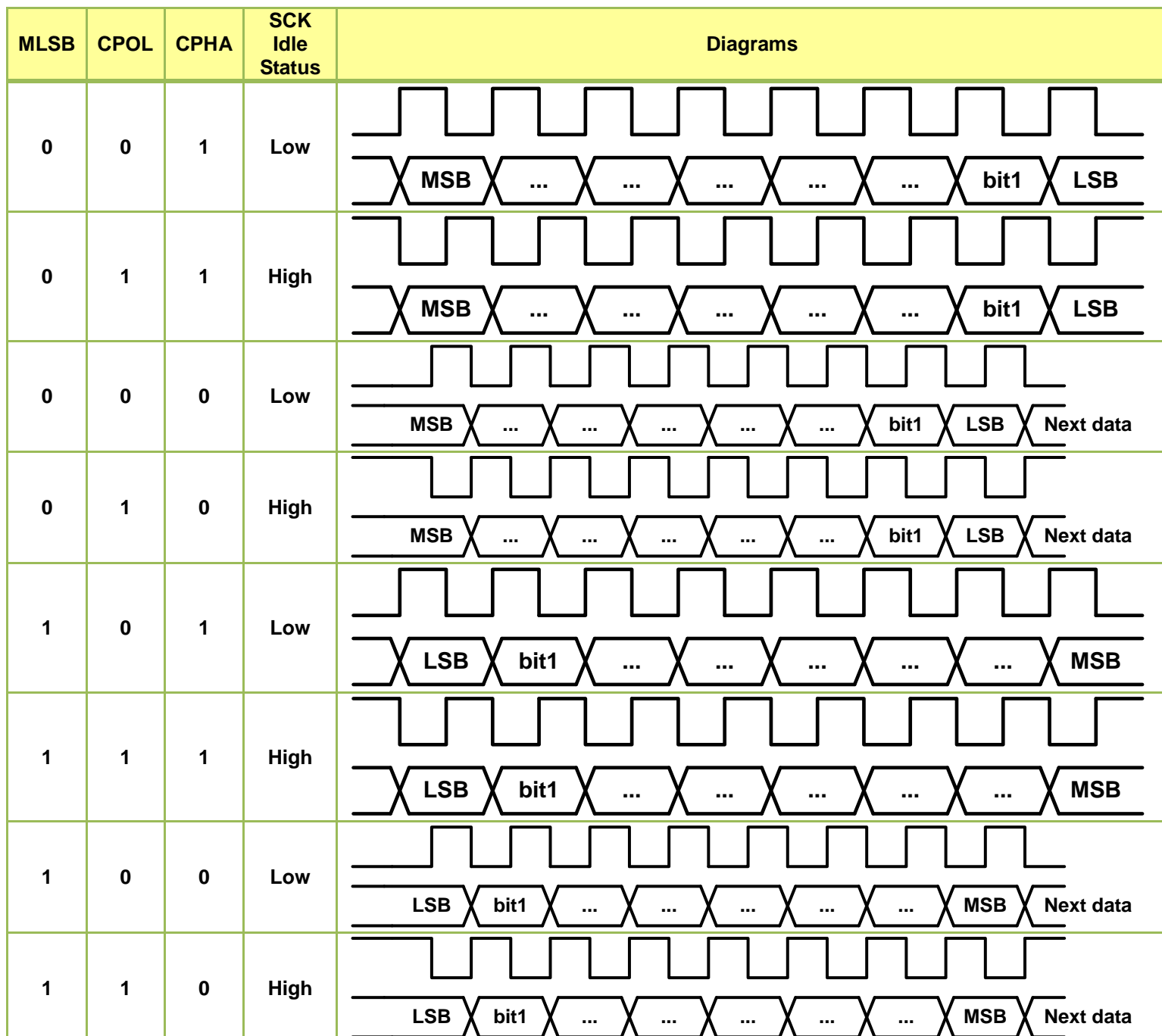
## 10.4 INTERFACE DESCRIPTION

### 10.4.1 SPI

The SPI interface is a 4-wire interface where the SSEL 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 2<sup>nd</sup> edge. When CPHA=0, the 1<sup>st</sup> bit is fixed already, and the SCK first edge is to receive and transmit data.

The SIO data transfer timing as following figure:



## 10.4.2 SSI

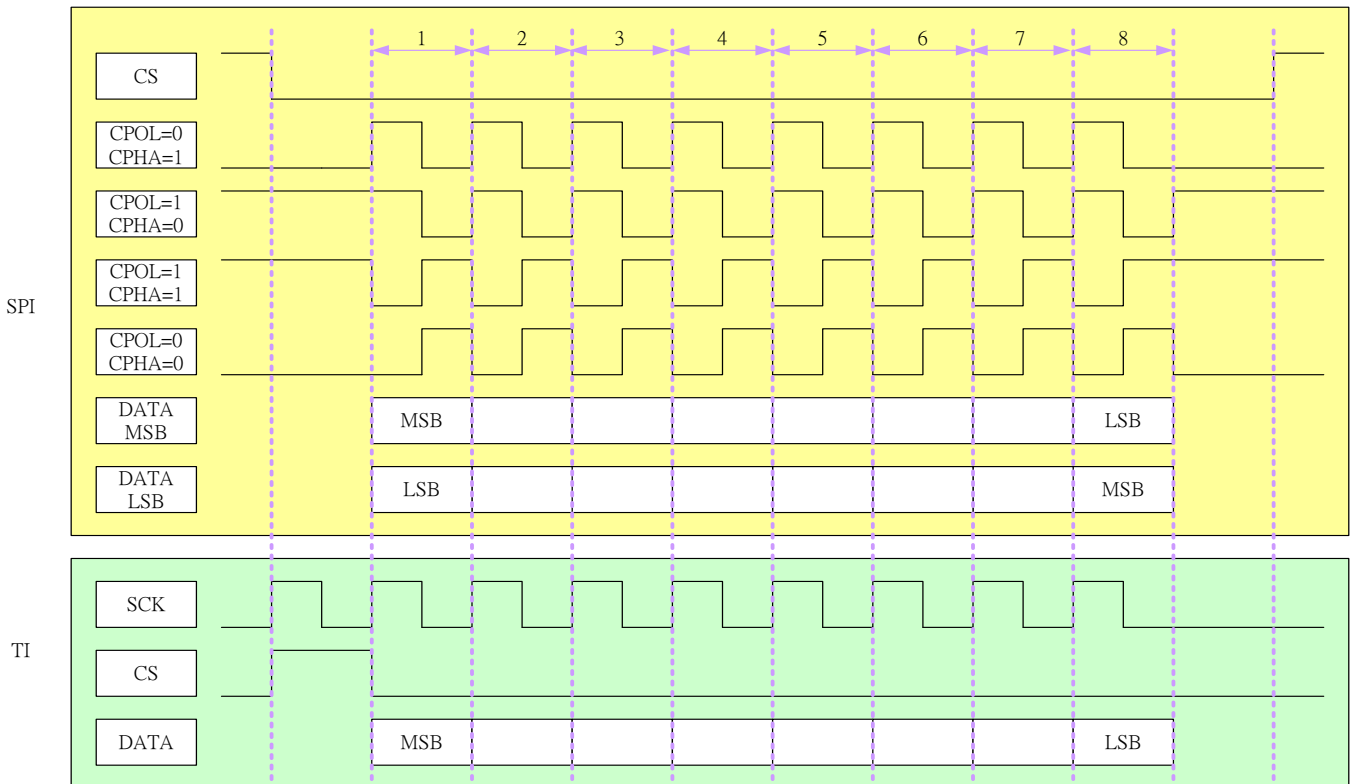
For device configured as a master in this mode, SCK and CS are forced LOW, and the transmit data line DATA is in 3-state mode whenever the SSP is idle.

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 DATA pin. Likewise, the MSB of the received data is shifted onto the DATA pin by the off-chip serial slave device.

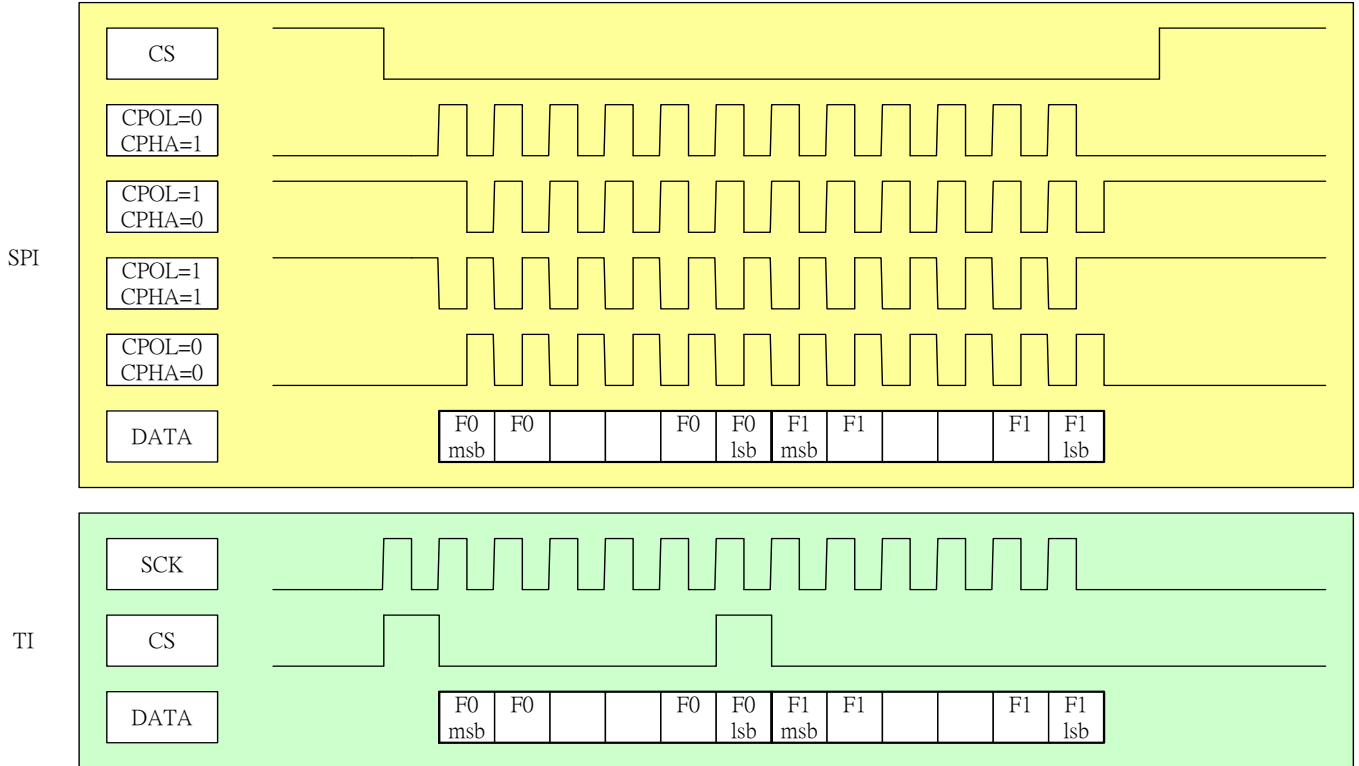
Both the SSP and the off-chip serial slave device then clock each data bit into their serial shifter on the falling edge of each SCK. The received data is transferred from the serial shifter to the receive FIFO on the first rising edge of SCK after the LSB has been latched.

## 10.4.3 COMMUNICATION FLOW

### 10.4.3.1 SINGLE-FRAME



### 10.4.3.2 MULTI-FRAME



## 10.5 Auto-SEL (Auto-CS)

The Auto-SEL function is enabled by default, and Auto-SEL data flow is controlled by hardware. If Auto-SEL function is enabled, the SPI's hardware controls the SEL output of the SPI.

If Auto-SEL function is disabled by setting the SELDIS bit, the SELCTRL bit controls the SEL output of the SPI. If Auto-SEL function is enabled, hardware controls the SEL output, and the actual value of SEL will be copied in the SELCTRL Control bit of the SPI. As long as Auto-SEL is enabled, the value of the SELCTRL Control bit is read-only for software.



## 10.6 SSP REGISTERS

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

### 10.6.1 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 |
|-------|---------------|--|-----------|-------|
| 31:20 | Reserved      |  | R         | 0     |
| 19    | SELCTRL       | Source for SEL pin. For SPI mode only.<br>0: SEL pin is low level.<br>1: SEL pin is high level.  | R/W       | 1     |
| 18    | SELDIS        | Auto-SEL disable bit. For SPI mode only.<br>0: Enable Auto-SEL flow control.<br>1: Disable Auto-SEL flow control.  | R/W       | 0     |
| 17:15 | RXFIFOTH[2:0] | RX FIFO Threshold level<br>0: RX FIFO threshold level = 0<br>1: RX FIFO threshold level = 1<br>...<br>...<br>n: RX FIFO threshold level = n  | R/W       | 000b  |
| 14:12 | TXFIFOTH[2:0] | TX FIFO Threshold level<br>0: TX FIFO threshold level = 0<br>1: TX FIFO threshold level = 1<br>...<br>...<br>n: TX FIFO threshold level = n  | R/W       | 000b  |
| 11:8  | DL[3:0]       | Data length = DL[3:0] + 1<br>0000~0001: Reversed<br>0010: data length = 3<br>...<br>...<br>1110: data length = 15<br>1111: data length = 16  | R/W       | 1111b |
| 7:6   | FRESET[1:0]   | SSP FSM and FIFO Reset bit<br>00: No effect<br>01: Reserved<br>10: Reserved<br>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.<br>0: SPI<br>1: SSI  | R/W       | 0     |
| 3     | MS            | Master/Slave selection bit<br>0: Act as Master.<br>1: Act as Slave.  | R/W       | 0     |
| 2     | SDODIS        | Slave data output disable bit (ONLY used in slave mode)<br>0: Enable slave data output.<br>1: Disable slave data output. (MISO=0)  | R/W       | 0     |
| 1     | LOOPBACK      | Loop back mode enable<br>0: Disable<br>1: Data input from data output  | R/W       | 0     |

|          |       |  |     |   |
|----------|-------|--|-----|---|
| <b>0</b> | SSPEN | SSP enable bit<br>0: Disable<br>1: Enable. | R/W | 0 |
|----------|-------|--|-----|---|

### 10.6.2 SSP n Control register 1 (SSPn\_CTRL1) (n=0, 1)

Address Offset: 0x04

| Bit         | Name     | Description  | Attribute | Reset |
|-------------|----------|--|-----------|-------|
| <b>31:3</b> | Reserved |  | R         | 0     |
| <b>2</b>    | CPHA     | Clock phase for edge sampling.<br>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.<br>1: Data changes at clock rising edge, latches at clock falling edge when CPOL = 0; Data changes at clock falling edge, latches at clock rising edge when CPOL = 1. | R/W       | 0     |
| <b>1</b>    | CPOL     | Clock polarity selection bit<br>0: SCK idles at Low level.<br>1: SCK idles at High level.  | R/W       | 0     |
| <b>0</b>    | MLSB     | MSB/LSB selection bit<br>0: MSB transmit first.<br>1: LSB transmit first.  | R/W       | 0     |

### 10.6.3 SSP n Clock Divider register (SSPn\_CLKDIV) (n=0, 1)

Address Offset: 0x08

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

### 10.6.4 SSP n Status register (SSPn\_STAT) (n=0, 1)

Address Offset: 0x0C

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

### 10.6.5 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    | TXFIFOTHIE | TX FIFO threshold interrupt enable<br>0: Disable<br>1: Enable | R/W       | 0     |
| 2    | RXFIFOTHIE | RX FIFO threshold interrupt enable<br>0: Disable<br>1: Enable | R/W       | 0     |
| 1    | RXTOIE     | RX time-out interrupt enable<br>0: Disable<br>1: Enable       | R/W       | 0     |
| 0    | RXOVFIE    | RX Overflow interrupt enable<br>0: Disable<br>1: Enable       | R/W       | 0     |

### 10.6.6 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    | TXFIFOTHIF | TX FIFO threshold interrupt flag<br>0: No TX FIFO threshold interrupt<br>1: TX FIFO threshold triggered.   | R         | 0     |
| 2    | RXFIFOTHIF | RX FIFO threshold interrupt flag<br>0: No RX FIFO threshold interrupt<br>1: RX FIFO threshold triggered.   | R         | 0     |
| 1    | RXTOIF     | RX time-out interrupt flag<br>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.<br>0: RXTO doesn't occur.<br>1: RXTO occurs.                          | R         | 0     |
| 0    | RXOVFIF    | RX Overflow interrupt flag<br>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.<br>0: RXOVF doesn't occur.<br>1: RXOVF occurs. | R         | 0     |

### 10.6.7 SSP n Interrupt Clear register (SSPn\_IC) (n=0, 1)

Address Offset: 0x18

| Bit  | Name       | Description                              | Attribute | Reset |
|------|------------|--|-----------|-------|
| 31:4 | Reserved   |  | R         | 0     |
| 3    | TXFIFOTHIC | 0: No effect<br>1: Clear TXFIFOTHIF bit. | W         | 0     |
| 2    | RXFIFOTHIC | 0: No effect<br>1: Clear RXFIFOTHIF bit. | W         | 0     |

|          |         |                                       |   |   |
|----------|---------|---------------------------------------|---|---|
| <b>1</b> | RXTOIC  | 0: No effect<br>1: Clear RXTOIF bit.  | W | 0 |
| <b>0</b> | RXOVFIC | 0: No effect<br>1: Clear RXOVFIF bit. | W | 0 |

### 10.6.8 SSP n Data register (SSPn\_DATA) (n=0, 1)

Address Offset: 0x1C

| Bit          | Name       | Description  | Attribute | Reset |
|--------------|------------|--|-----------|-------|
| <b>31:16</b> | Reserved   |  | R         | 0     |
| <b>15:0</b>  | DATA[15:0] | <p><u>Write</u><br/>SW can write data to be sent in a future frame to this register when TX_FULL = 0 in <a href="#">SSPn_STAT</a> 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><br/>SW can read data from this register when RX_EMPTY=0 in <a href="#">SSPn_STAT</a> 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 I2C

## 11.1 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.

Depending on the state of the direction bit (R/W), two types of data transfers are possible on the I2C bus:

- Data transfer from a master transmitter to a slave receiver.  
The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte.
- Data transfer from a slave transmitter to a master receiver.  
The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. Next follows the data bytes transmitted by the slave to the master. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a “not acknowledge” is returned. The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a Repeated START condition. Since a Repeated START condition is also the beginning of the next serial transfer, the I2C bus will not be released.

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

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

## 11.2 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.

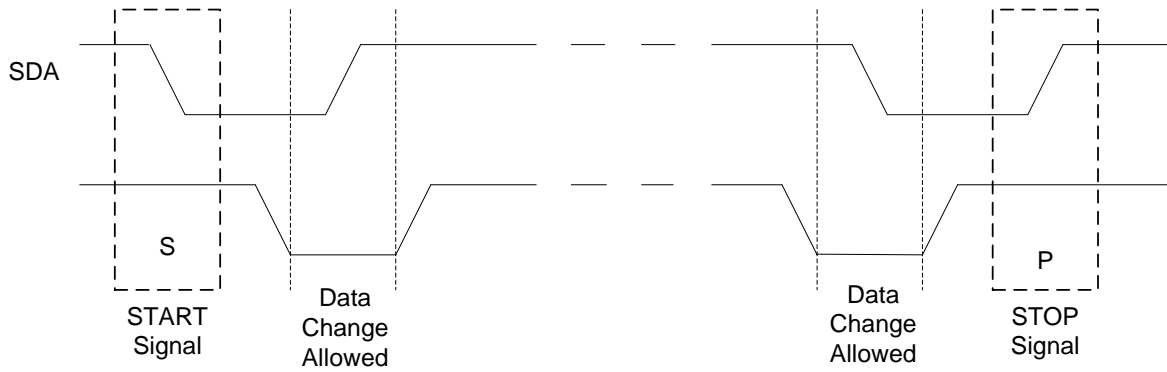
- Standard I2C-compliant bus interfaces may be configured as Master or Slave.
- I2C Master features:
  - Clock generation
  - Start and Stop generation
- I2C Slave features:
  - Programmable I2C Address detection
  - Optional recognition of up to four distinct slave addresses
  - Stop bit detection
- Supports different communication speeds:
  - Standard Speed (up to 100KHz)
  - Fast Speed (up to 400 KHz)
- 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.

### 11.3 PIN DESCRIPTION

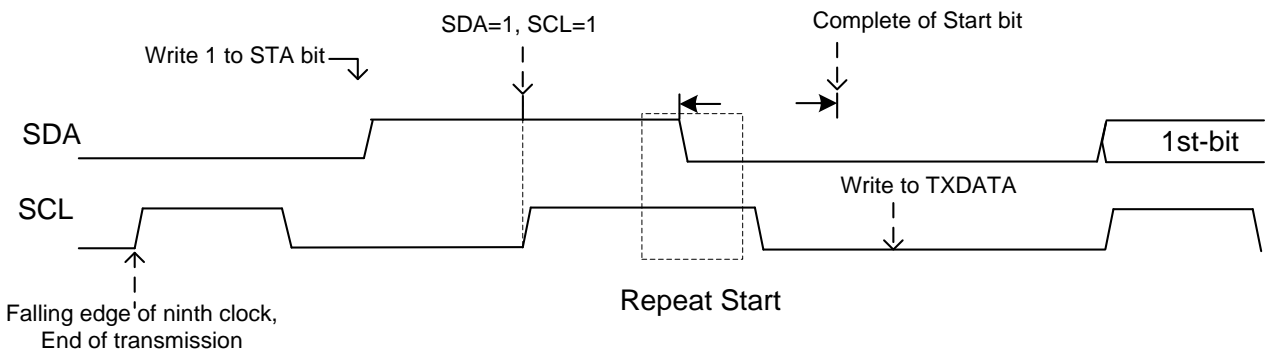
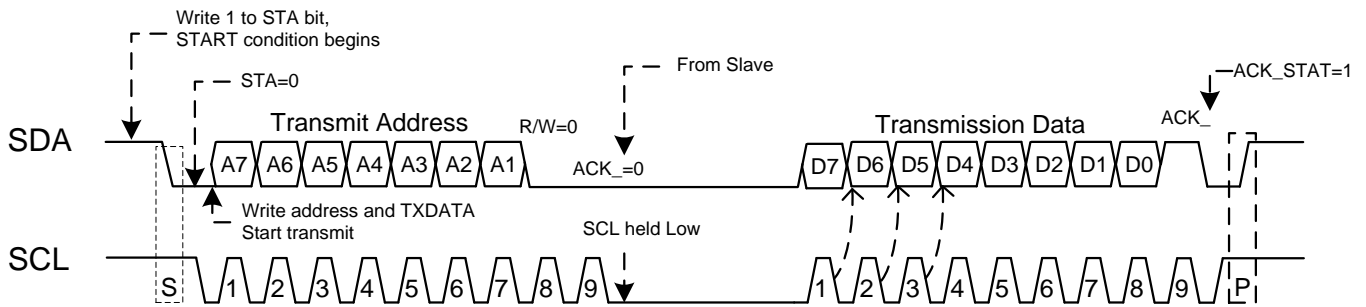
| Pin Name | Type | Description      | GPIO Configuration  |
|----------|------|------------------|---|
| SCLn     | I/O  | I2C Serial clock | Output with Open-drain<br>Input depends on GPIO <sub>n</sub> _CFG |
| SDAn     | I/O  | I2C Serial data  | Output with Open-drain<br>Input depends on GPIO <sub>n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

### 11.4 WAVE CHARACTERISTICS

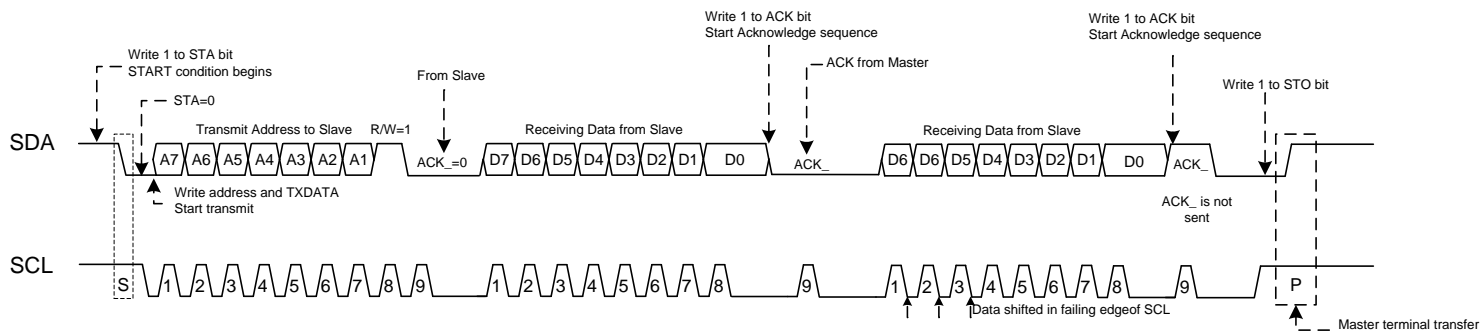


## 11.5 I2C MASTER MODES

### 11.5.1 MASTER TRANSMITTER MODE



### 11.5.2 MASTER RECEIVER MODE



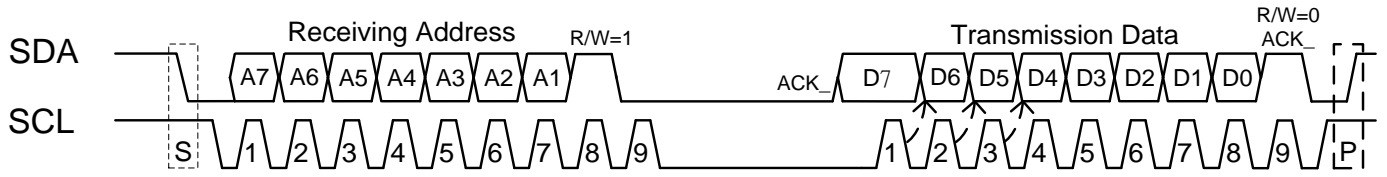
### 11.5.3 ARBITRATION

In the master transmitter mode, the arbitration logic checks that every transmitted logic 1 actually appears as a logic 1 on the I2C bus. If another device on the bus overrules a logic 1 and pulls the SDA line low, arbitration is lost, and the I2C block immediately changes from master transmitter to slave receiver. The I2C block will continue to output clock pulses (on SCL) until transmission of the current serial byte is complete.

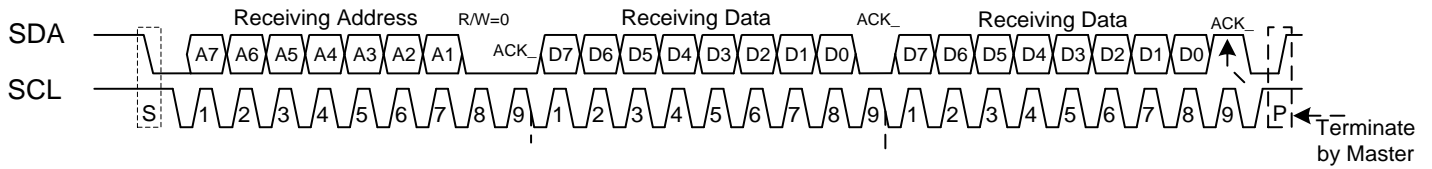
Arbitration may also be lost in the master receiver mode. Loss of arbitration in this mode can only occur while the I2C block is returning a “not acknowledge” to the bus. Arbitration is lost when another device on the bus pulls this signal low. Since this can occur only at the end of a serial byte, the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

## 11.6 I2C SLAVE MODES

### 11.6.1 SLAVE TRANSMITTER MODE



### 11.6.2 SLAVE RECEIVER MODE





## **11.7 MONITOR MODE**

### **11.7.1 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.

### **11.7.2 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.

## 11.8 I2C REGISTERS

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

### 11.8.1 I2C n Control register (I2Cn\_CTRL) (n=0,1)

Address Offset: 0x00

The I2Cn\_CTRL registers control setting of bits that 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, and HW will assign SCL0/SCL1 and SDA0/SDA1 pins as 20mA high-sinking current if I2CMODE =1.
- 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<br>0: Disable. The STO bit is forced to “0”.<br>1: Enable.<br>I2EN 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.<br>0: No START condition or Repeated START condition will be generated.<br>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<br>0: Stop condition idle.<br>1: Cause the I2C interface to transmit a STOP condition in master mode, or recover from an error condition in slave mode.<br>Automatically cleared by HW.  | R/W       | 0     |
| 3    | Reserved |  | R         | 0     |
| 2    | ACK      | Assert ACK (Low level to SDA) flag.<br>0: Master mode→ No function<br>Slave mode→Return a NACK after receiving address or data.<br>1: An ACK will be returned during the acknowledge clock pulse on SCLn when <ul style="list-style-type: none"> <li>➤ The address in the Slave Address register has been received.</li> <li>➤ The General Call address has been received while the General Call bit (GC) in the ADR register is set.</li> </ul> | R/W       | 0     |

|          |          |  |     |   |
|----------|----------|--|-----|---|
|          |          | <ul style="list-style-type: none"> <li>➤ A data byte has been received while the I2C is in the master receiver mode.</li> <li>➤ A data byte has been received while the I2C is in the addressed slave receiver mode.</li> </ul> HW will clear after issuing ACK automatically.   |     |   |
| <b>1</b> | NACK     | Assert NACK (HIGH level to SDA) flag.<br>0: No function<br>1: An NACK will be returned during the acknowledge clock pulse on SCLn when <ul style="list-style-type: none"> <li>➤ A data byte has been received while the I2C is in the master receiver mode.</li> </ul> HW will clear after issuing NACK automatically. | R/W | 0 |
| <b>0</b> | Reserved |  | R   | 0 |

### 11.8.2 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.

While I2CIF =1, the low period of the serial clock on the SCL line is stretched, and the serial transfer is suspended. When SCL is HIGH, it is unaffected by the state of I2CIF.

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

### 11.8.3 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  |

### 11.8.4 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  |

### 11.8.5 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.<br>0 : 7-bit address mode<br>1: 10-bit address mode                              | RW        | 0     |
| 30    | GCEN      | General call address enable bit.<br>0: Disable<br>1: Enable general call address (0x0)               | RW        | 0     |
| 29:10 | Reserved  |  | R         | 0     |
| 9:0   | ADDR[9:0] | The I2C slave address.<br>ADD[9:0] is valid when ADD_MODE = 1<br>ADD[7:1] is valid when ADD_MODE = 0 | R/W       | 0     |

### 11.8.6 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.<br>ADD[9:0] is valid when ADD_MODE = 1<br>ADD[7:1] is valid when ADD_MODE = 0 | R/W       | 0     |

### 11.8.7 I2C n SCL High Time register (I2Cn\_SCLHT) (n=0,1)

Address Offset: 0x20

\* **Note:** I2C 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<br>SCL High Period Time = (SCLH+1) * I2C0_PCLK cycle | R/W       | 0x04  |

### 11.8.8 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<br>SCL Low Period Time = (SCLL+1) * I2C0_PCLK cycle | R/W       | 0x04  |

### 11.8.9 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 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.<br>0: Disable Timeout checking<br>N: Timeout period time = N*I2Cn_PCLK cycle | R/W       | 0x0   |

### 11.8.10 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<br>0: Interrupt will only be generated when the address matches one of the values in I2Cn_SLVADDR0~3 register.<br>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.<br>0: SCL output will be forced high.<br>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.<br>0: Disable<br>1: Enable.   | R/W       | 0     |

# 12 UNIVERSAL ASYNCHRONOUS SERIAL RECEIVER AND TRANSMITTER (UART)

## 12.1 OVERVIEW

The UART 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 UART offers a very wide range of baud rates using a fractional baud rate generator.

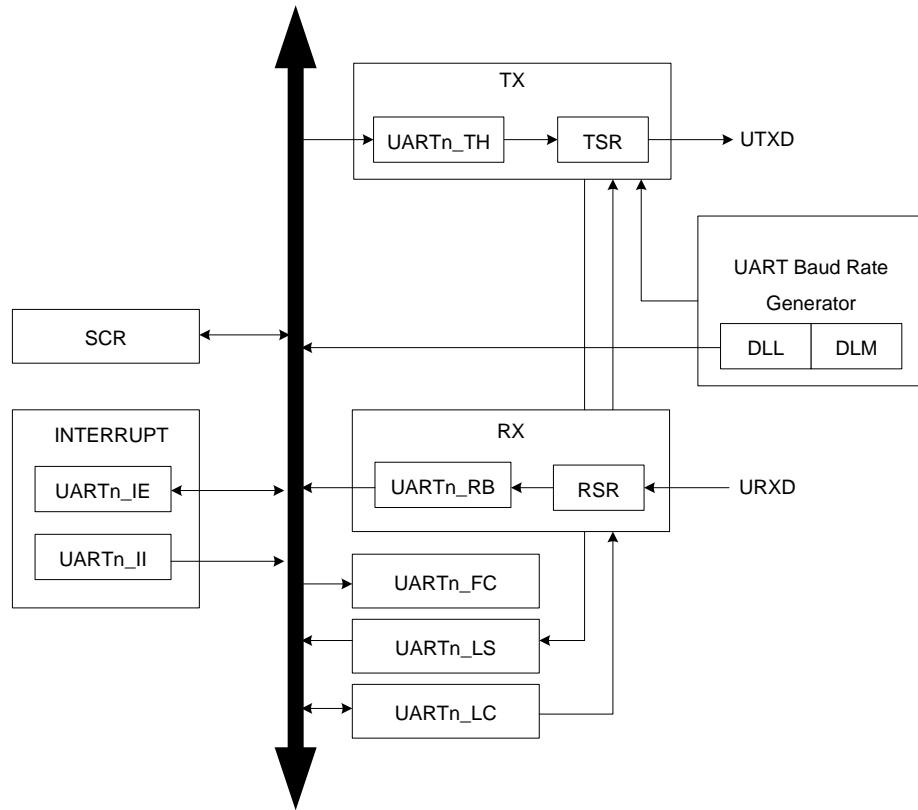
## 12.2 FEATURES

- Full-duplex, 2-wire asynchronous data transfer.
- Single-wire half-duplex communication
- 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.

## 12.3 PIN DESCRIPTION

| Pin Name | Type | Description           | GPIO Configuration                |
|----------|------|-----------------------|-----------------------------------|
| UTXDn    | O    | Serial Transmit data. |                                   |
| URXDn    | I    | Serial Receive data.  | Depends on GPIO <sub>n</sub> _CFG |

**12.4 BLOCK DIAGRAM**



## 12.5 BAUD RATE CALCULATION

The UART baud rate is calculated as:

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

Where UARTn\_PCLK is the peripheral clock, [UARTn\\_DLM](#) and [UARTn\\_DLL](#) are the standard UART baud rate divider registers, and DIVADDDVAL and MULVAL are UART fractional baud rate generator specific parameters in [UARTn\\_FD](#) register.

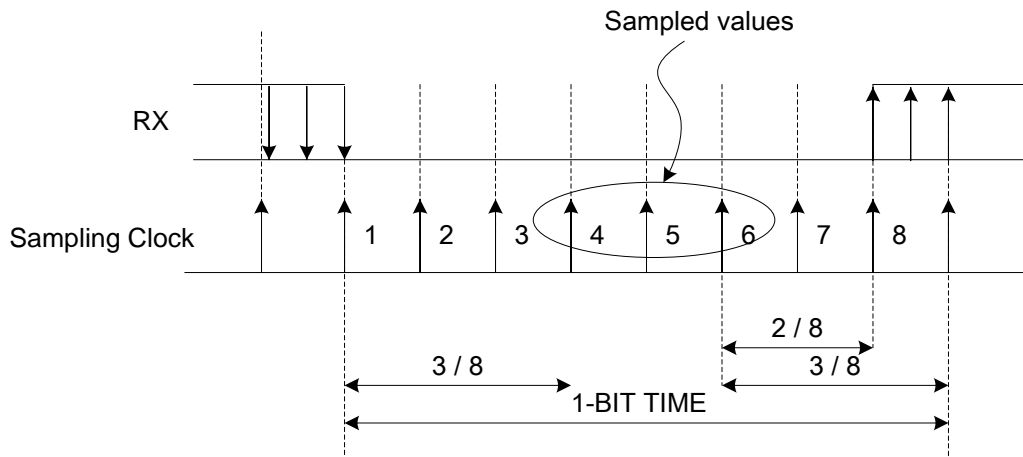
The value of MULVAL and DIVADDDVAL should comply to the following conditions:

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

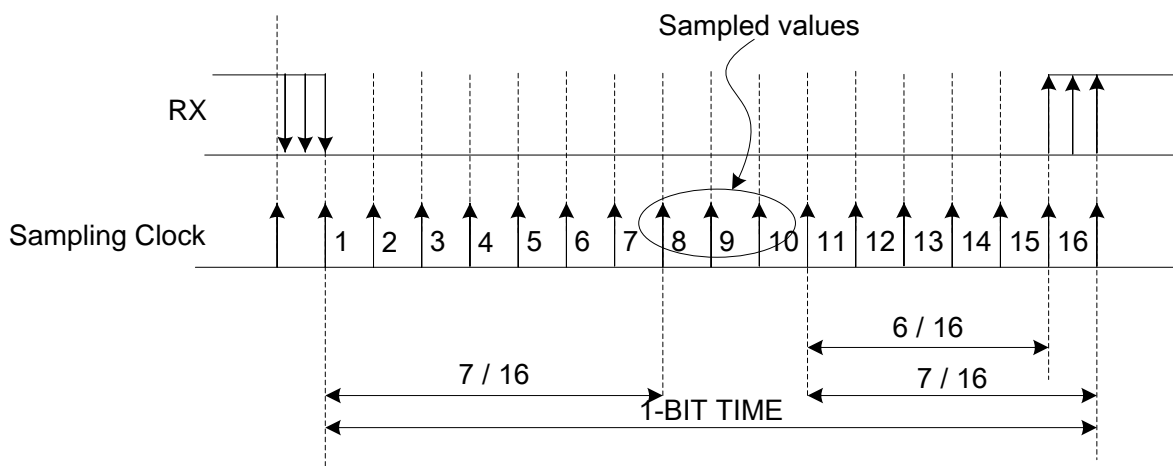
The value of the [UARTn\\_FD](#) register should not be modified while transmitting/receiving data or data may be lost or corrupted.

The oversampling method can be selected by programming the OVER8 bit in [UARTn\\_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  $\text{UARTn\_PCLK}/8$ ). 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  $\text{UARTn\_PCLK}/16$





If the [UARTn\\_FD](#) register value does not comply to 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.

UART 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 UARTn\_PCLK = 12 MHz, and Oversampling = 16.

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

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

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

Since the value of MULVAL and DIVADDVAL should comply to 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. The baud rate generated is 115384, and has a relative error of 0.16% from the originally specified 115200.

## 12.6 AUTO-BAUD FLOW

### 12.6.1 AUTO-BAUD

The UART 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 [UARTn\\_DLM](#) and [UARTn\\_DLL](#) accordingly.

Auto-baud function is started by setting the START bit in [UARTn\\_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 [UARTn\\_ABCTRL](#) register. In Mode 0 the baud rate is measured on two subsequent falling edges of the UART 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 UART RX pin (the length of the start bit).

The AUTORESTART bit in [UARTn\\_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 [UARTn\\_II](#) register will get set if the interrupt is enabled (ABTOIE bit in [UARTn\\_IE](#) register is set and the auto-baud rate measurement counter overflows).
- The ABEOINT interrupt in [UARTn\\_II](#) register will get set if the interrupt is enabled (ABTOIE bit in [UARTn\\_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 [UARTn\\_IE](#) register.

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

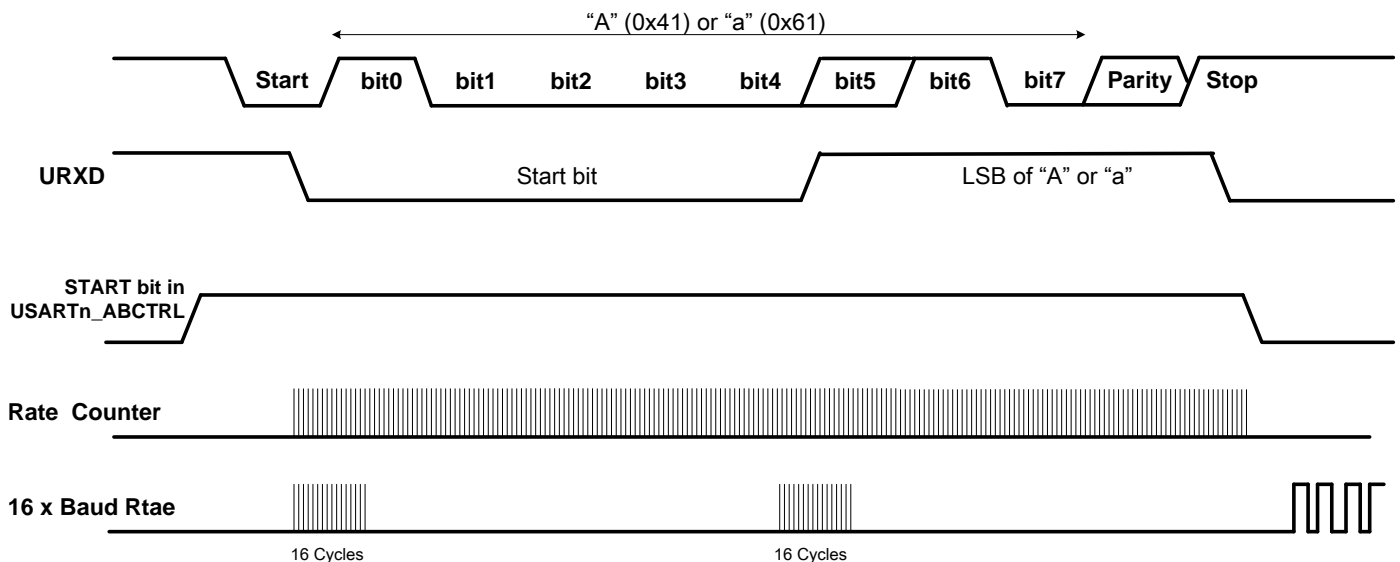
$$ratemin = \frac{2 \times PCLK}{16 \times 2^{15}} \leq \text{UART baudrate} \leq \frac{PCLK}{16 \times (2 + databits + paritybits + stopbits)} = ratemax$$

## 12.6.2 AUTO-BAUD MODES

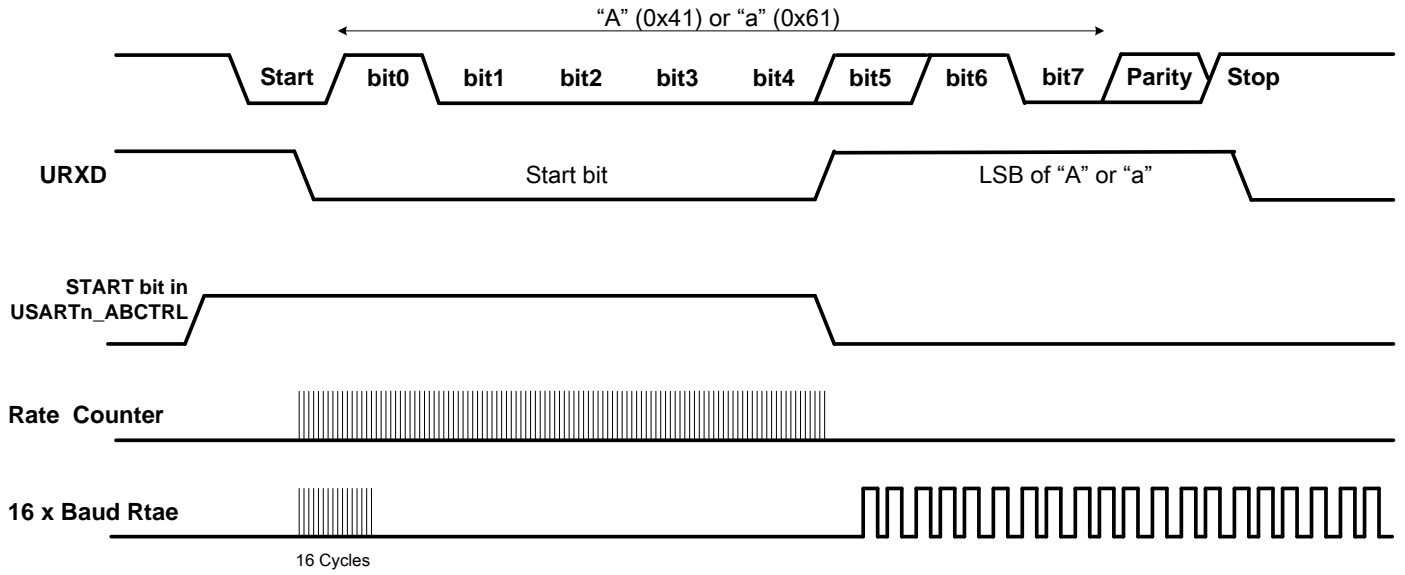
When the SW is expecting an “AT” command, it configures the UART with the expected character format and sets the ACR Start bit. The initial values in the divisor latches DLM and DLL don’t care. Because of the “A” or “a” ASCII coding (“A” = 0x41, “a” = 0x61), the UART Rx pin sensed start bit and the LSB 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 UARTn\_PCLK cycles.
3. During the receipt of the start bit, 16 pulses are generated on the RSR baud input with the frequency of the UART 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 [UARTn\\_ABCTRL](#) register), the rate counter will continue incrementing with the pre-scaled UART input clock (UARTn\_PCLK).
5. If MODE = 0, the rate counter will stop on next falling edge of the UART 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 [UARTn\\_DLM/UARTn\\_DLL](#) and the baud rate will be switched to normal operation. After setting the DLM/DLL, the end of auto-baud interrupt ABEOINT in [UARTn\\_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



## 12.7 UART REGISTERS

Base Address: 0x4001 6000 (UART0)  
0x4005 6000 (UART1)

### 12.7.1 UART n Receiver Buffer register (UARTn\_RB) (n=0, 1)

Address Offset: 0x00

This register is the top byte of the UART 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 [UARTn\\_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 UART 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 [UARTn\\_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 UART RX FIFO. | R         | 0     |

### 12.7.2 UART n Transmitter Holding register (UARTn\_TH) (n=0, 1)

Address Offset: 0x00

This register is the top byte of the UART 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 [UARTn\\_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     |

### 12.7.3 UART n Divisor Latch LSB registers (UARTn\_DLL) (n = 0, 1)

Address Offset: 0x00

The UART Divisor Latch is part of the UART Baud Rate Generator and holds the value used (optionally with the Fractional Divider) to divide the UARTn\_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 UARTn\_DLL and UARTn\_DLM registers together form a 16-bit divisor, and DLAB bit in [UARTn\\_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 UART Divisor Latch LSB Register, along with the DLM register, determines the baud rate of the UART. | R/W       | 0     |

### 12.7.4 UART n Divisor Latch MSB register (UARTn\_DLM) (n=0,1)

Address Offset: 0x04

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

### 12.7.5 UART n Interrupt Enable register (UARTn\_IE) (n=0, 1)

Address Offset: 0x04

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

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

### 12.7.6 UART n Interrupt Identification register (UARTn\_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 [UARTn\\_II](#) register access. If an interrupt occurs during a [UARTn\\_II](#) register access, the interrupt is recorded for the next [UARTn\\_II](#) register access.

| Bit   | Name       | Description  | Attribute | Reset |
|-------|------------|--|-----------|-------|
| 31:10 | Reserved   |  | R         | 0     |
| 9     | ABTOIF     | Auto-baud time-out interrupt flag.<br>0: Auto-baud has not timed-out<br>1: Auto-baud has timed out and interrupt is enabled.                       | R         | 0     |
| 8     | ABEOIF     | End of auto-baud interrupt flag<br>0: Auto-baud has not finished.<br>1: Auto-baud has finished successfully and interrupt is enabled.              | R         | 0     |
| 7:6   | FIFOEN     | Equivalent to FIFOEN bit in <a href="#">UARTn_FIFOCTRL</a> register.   | R         | 1     |
| 5:4   | Reserved   |  | R         | 0     |
| 3:1   | INTID[2:0] | Interrupt identification which identifies an interrupt corresponding to the <a href="#">UARTn_RX_FIFO</a> .<br>0x3: 1 - Receive Line Status (RLS). | R         | 0     |

|          |                  |  |          |          |
|----------|------------------|--|----------|----------|
|          |                  | 0x2: 2a - Receive Data Available (RDA).<br>0x6: 2b - Character Time-out Indicator (CTI).<br>0x1: 3a - THRE Interrupt.<br>0x7: 3b – TEMT Interrupt<br>Other: Reserved |          |          |
| <b>0</b> | <b>INTSTATUS</b> | Interrupt status. The pending interrupt can be determined by evaluating UARTn_I[3:1].<br>0: At least one interrupt is pending.<br>1: No interrupt is pending.        | <b>R</b> | <b>1</b> |

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

| Interrupt | UARTn_I [3:0] | Priority        | Interrupt Source   | Interrupt Reset  |
|-----------|---------------|-----------------|--|--|
| RLS       | 0110          | Highest         | Overrun error (OE), Parity error (PE), Framing error (FE) or Break interrupt (BI)  | Read UARTn_LS register   |
| RDA       | 0100          | 2 <sup>nd</sup> | RX data in FIFO reached trigger level (FCR0=1)   | Read UARTn_RB register or UART FIFO drops below trigger level        |
| CTI       | 1100          | 2 <sup>nd</sup> | 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 UARTn_RB register   |
| THRE      | 0010          | 3 <sup>rd</sup> | THRE   | Read UARTn_I register (if source of interrupt) or Write THR register |
| TEMT      | 1110          | 3 <sup>rd</sup> | TEMT   | Read UARTn_I register (if source of interrupt) or Write THR register |

### 12.7.7 UART n FIFO Control register (UARTn\_FIFCTRL) (n=0,1)

Address Offset: 0x08

This register controls the operation of the UART 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 UART FIFO characters must be written before an interrupt is activated.<br>00: Trigger level 0 (1 character)<br>01: Trigger level 1 (4 characters)<br>10: Trigger level 2 (8 characters)<br>11: Trigger level 3 (14 characters) | W         | 0     |
| 5:3  | Reserved  |   | R         | 0     |
| 2    | TXFIFORST | TX FIFO Reset bit.<br>0: No impact on either of UART FIFOs.<br>1: Writing a logic 1 to reset the pointer logic in UART TX FIFO. HW shall clear this bit automatically.  | W         | 0     |
| 1    | RXFIFORST | RX FIFO Reset bit.<br>0: No impact on either of UART FIFOs.<br>1: Writing a logic 1 to reset the pointer logic in UART RX FIFO. HW shall clear this bit automatically.  | W         | 0     |
| 0    | FIFOEN    | FIFO enable<br>0: No effect<br>1: Enable for both UART Rx and TX FIFOs and UARTn_FIFCTRL [7:1] access. This bit must be set for proper UART operation.  | W         | 1     |

### 12.7.8 UART n Line Control register (UARTn\_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<br>0: Disable access to Divisor Latches.<br>1: Enable access to Divisor Latches.  | R/W       | 0     |
| 6    | BC       | Break Control bit<br>0: Disable break transmission.<br>1: Enable break transmission. Output pin UART TXD is forced to logic 0.   | R/W       | 0     |
| 5:4  | PS[1:0]  | Parity Select bits<br>00: Odd parity. Number of 1s in the transmitted character and the attached parity bit will be odd.<br>01: Even Parity. Number of 1s in the transmitted character and the attached parity bit will be even.<br>10: Forced 1 stick parity.<br>11: Forced 0 stick parity. | R/W       | 0     |
| 3    | PE       | Parity Enable bit<br>0: Disable parity generation and checking.<br>1: Enable parity generation and checking.   | R/W       | 0     |
| 2    | SBS      | Stop Bit Select bit<br>0: 1 stop bit.<br>1: 2 stop bits (1.5 if WLS bits=00).  | R/W       | 0     |
| 1:0  | WLS[1:0] | Word Length Select bits<br>00: 5-bit character length.<br>01: 6-bit character length.<br>10: 7-bit character length.<br>11: 8-bit character length.  | R/W       | 0     |

### 12.7.9 UART n Line Status register (UARTn\_LS) (n=0,1)

Address Offset: 0x14

**\* Note:**

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

| Bit  | Name     | Description   | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:8 | Reserved |   | R         | 0     |
| 7    | RXFE     | Error in RX FIFO flag.<br>RXFE =1 when a character with a RX error such as framing error, parity error, or break interrupt, is loaded into the UARTn_RB register. This bit is cleared when the UARTn_LS register is read and there are no subsequent errors in the UART FIFO.<br>0: UARTn_RB register contains no UART RX errors or FIFOEN=0<br>1: UARTn_RB register contains at least one UART RX error.   | R         | 0     |
| 6    | TEMT     | Transmitter Empty flag<br>TEMT=1 when both THR and TSR are empty; TEMT is cleared when either the TSR or the THR contain valid data.<br>0: THR and/or TSR contains valid data.<br>1: THR and TSR are empty.   | R         | 1     |
| 5    | THRE     | Transmitter Holding Register Empty flag<br>THRE indicates that the UART is ready to accept a new character for transmission. In addition, this bit causes the UART 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.<br>0: THR contains valid data.<br>1: THR (TX FIFO) is empty.   | R         | 1     |
| 4    | BI       | Break Interrupt flag.<br>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 UARTn_LS register read clears BI bit. The time of break detection is dependent on FIFOEN bit in UARTn_FIFOCtrl register.<br>0: Break interrupt status is inactive.<br>1: Break interrupt status is active.   | R         | 0     |
| 3    | FE       | Framing Error flag.<br>When the stop bit of a received character is a logic 0, a framing error occurs. A UARTn_LS register read clears FE bit. The time of the framing error detection is dependent on FIFOEN bit in UARTn_FIFOCtrl register.<br>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.<br>0: Framing error status is inactive.<br>1: Framing error status is active. | R         | 0     |
| 2    | PE       | Parity Error flag.<br>When the parity bit of a received character is in the wrong state, a parity error occurs. A UARTn_LS register read clears PE bit. Time of parity error detection is dependent on FIFOEN bit in UARTn_FIFOCtrl register.<br>0: Parity error status is inactive.<br>1: Parity error status is active.   | R         | 0     |
| 1    | OE       | Overrun Error flag.<br>The overrun error condition is set as soon as it occurs. A UARTn_LS register read clears OE bit. OE=1 when UART RSR has a new character assembled and the UARTn_RB FIFO is full. In this case, the UARTn_RB FIFO will not be overwritten and the character in the UARTn_RS register will be lost.<br>0: Overrun error status is inactive.<br>1: Overrun error status is active.  | R         | 0     |
| 0    | RDR      | Receiver Data Ready flag<br>RDR=1 when the UARTn_RB FIFO holds an unread character and is   | R         | 0     |



|  |   |  |  |
|--|---|--|--|
|  | cleared when the UARTn_RB FIFO is empty.<br>0: UARTn_RB FIFO is empty.<br>1: UARTn_RB FIFO contains valid data. |  |  |
|--|---|--|--|

### 12.7.10 UART n Scratch Pad register (UARTn\_SP) (n=0, 1)

Address Offset: 0x1C

This register has no effect on the UART 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     |

### 12.7.11 UART n Auto-baud Control register (UARTn\_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 UART disabled making sure that UART is fully SW and HW compatible with UARTs 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<br>0: No effect<br>1: Clear ABTOIF bit. This bit is automatically cleared by HW.  | W         | 0     |
| 8     | ABEOIFC     | End of auto-baud interrupt flag clear bit<br>0: No effect.<br>1: Clear ABEOIF bit. This bit is automatically cleared by HW.   | W         | 0     |
| 7:3   | Reserved    |   | R         | 0     |
| 2     | AUTORESTART | Restart mode<br>0: No restart<br>1: Restart in case of timeout (counter restarts at next UART RX falling edge)  | R/W       | 0     |
| 1     | MODE        | Auto-baud mode select bit.<br>0: Mode 0.<br>1: Mode 1.  | R/W       | 0     |
| 0     | START       | This bit is automatically cleared after auto-baud completion.<br>0: Auto-baud stop (auto-baud is not running).<br>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     |

### 12.7.12 UART n Fractional Divider register (UARTn\_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 UART 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  $UARTn\_DLM = 0$ , the value of the  $UARTn\_DLL$  register must  $\geq 3$ .

| Bit  | Name           | Description   | Attribute | Reset |
|------|----------------|---|-----------|-------|
| 31:9 | Reserved       |   | R         | 0     |
| 8    | OVER8          | Oversampling value<br>0: Oversampling by 16<br>1: Oversampling by 8   | R/W       | 0     |
| 7:4  | MULVAL[3:0]    | Baud rate pre-scaler multiplier value = MULVAL[3:0] +1<br>0000: Baud rate pre-scaler multiplier value is 1 for HW<br>0001: Baud rate pre-scaler multiplier value is 2 for HW<br>...<br>...<br>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 UART baud rate  | R/W       | 0     |

### 12.7.13 UART n Control register (UARTn\_CTRL) (n=0, 1)

Address Offset: 0x30

In addition to HW flow control, this register enables implementation of SW flow control.

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

It is strongly suggested to let the UART 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 UARTn_TH register is output on the TXD pin as soon as any preceding data has been sent.<br>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.<br>In other words, a 0 in this bit blocks the transfer of characters from the UARTn_TH register or TX FIFO into the transmit shift register. | R/W       | 1     |
| 6    | RXEN     | 0: Disable RX related function<br>1: Enable RX  | R/W       | 1     |
| 5:1  | Reserved |   | R         | 0     |
| 0    | UARTEN   | UART enable<br>0: Disable . All UART shared pins act as GPIO.<br>1: Enable. HW switches GPIO to UART pin automatically.   | R/W       | 0     |

### 12.7.14 UART n Half-duplex Enable register (UARTn\_HDEN) (n=0, 1)

Address Offset: 0x34

After reset the UART will be in full-duplex mode, meaning that both TX and RX work independently. After setting the HDEN bit, the UART will be in half-duplex mode. In this mode, the UART 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 UART 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.

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

# 13 AUDIO (I2S/CODEC)

## 13.1 OVERVIEW

### 13.1.1 I2S Description

The I2S bus specification defines a 5-wire serial bus, having one data in, one data out, one MCLK clock, one BCLK clock, and one word select signal. The basic I2S connection has one master, which is always the master, and one slave.

### 13.1.2 Codec Description

The SN32F100 mono Codec offers fundamental features suitable for system applications, which contains 16-Bit Sigma-delta ADC and DAC for audio in and audio out respectively.

The ADC supports a set of differential MIC input and the microphone input path includes a programmable gain amplifier (PGA) and MIC boost to adjust the analog volume. Also, there is a digital volume attenuation control after ADC, which can adjust the volume of digital output by bit shift. Besides, Audio Gain Controller (AGC) is programmable to monitor the input and further adjust the volume properly.

The DAC includes a power supply input for DAC driver, a power supply input pins for DAC, a DAC VMID output, and a DAC Common mode output.

The Codec circuit requires a 2.7~3.6 V operating voltage supply.

## 13.2 FEATURES

### 13.2.1 I2S 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.

### 13.2.2 Codec Features

- Mono Codec.
- Audio sample rates: 8K, 16K, 32K, and 48KHz.
- ADC.
  - ◇ SNR: 94dB (-120dBr, A-W).
  - ◇ DR: 94dB (-60dBr, A-W).
  - ◇ THD+N: -80dB (+0dBr)
- DAC.
  - ◇ SNR: 90dB (-120dBr, A-W).
  - ◇ DR: 90dB (-60dBr, A-W).
  - ◇ THD+N: -75dB (+0dBr).
- Differential microphone interface.
  - ◇ Programmable gain amplifier (PGA) -12dB~+33dB.
  - ◇ MIC boost gain 0,+12,+20,+30dB.
  - ◇ Auto Gain Control (AGC).
- Common mode output interface.
  - ◇ Mute on/off.

\* **Note:** The Codec circuit requires a 2.7V~3.6V operating voltage supply.

## 13.3 PIN DESCRIPTION

### 13.3.1 I2S 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 |
| I2SDIN   | I    | I2S Received Serial data         | Depends on GPIOOn_CFG |
| I2SDOUT  | O    | I2S Transmitted Serial data      | Depends on GPIOOn_CFG |
| I2SMCLK  | O    | I2S Master clock output          |                       |
|          | I    | I2S Master clock input from GPIO | Depends on GPIOOn_CFG |

### 13.3.2 Codec Pin Description

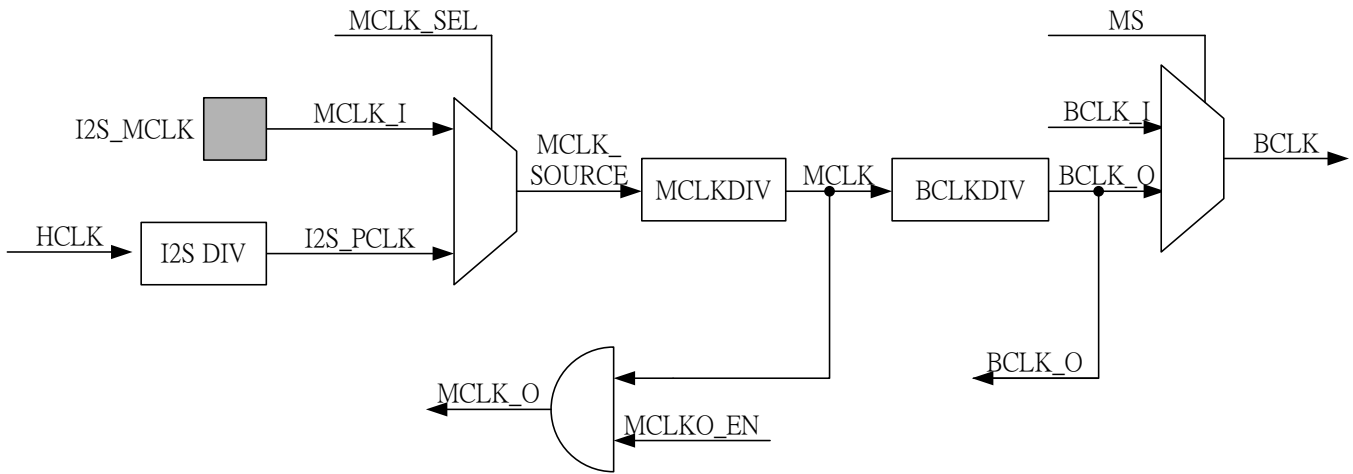
| Pin Name              | Type | Description   | GPIO Configuration    |
|-----------------------|------|---|-----------------------|
| AVDD_ADC,<br>AVSS_ADC | P    | Power supply input pins for Sigma-delta ADC.        |                       |
| VMID_ADC              | P    | Sigma-delta ADC VMID output.                        |                       |
| MIC_BIAS              | P    | Sigma-delta ADC Microphone Bias Voltage output.     |                       |
| MIC_P                 | I/O  | Sigma-delta ADC MIC difference input (+).           | Depends on GPIOOn_CFG |
| MIC_N                 | I/O  | Sigma-delta ADC MIC difference input (-).           | Depends on GPIOOn_CFG |
| AVDD_DAC,<br>AVSS_DAC | P    | Power supply input pins for Sigma-delta DAC.        |                       |
| AVDD_DRV,<br>AVSS_DRV | P    | Power supply input pins for Sigma-delta DAC Driver. |                       |
| VCOM_DAC              | P    | Sigma-delta DAC Common mode output.                 |                       |
| VMID_DAC              | P    | Sigma-delta DAC VMID output.                        |                       |
| VOUTP                 | I/O  | Sigma-delta DAC output (+).                         |                       |
| VOUTN                 | I/O  | Sigma-delta DAC output (-).                         |                       |

### 13.3.3 Audio Clock Pin Description

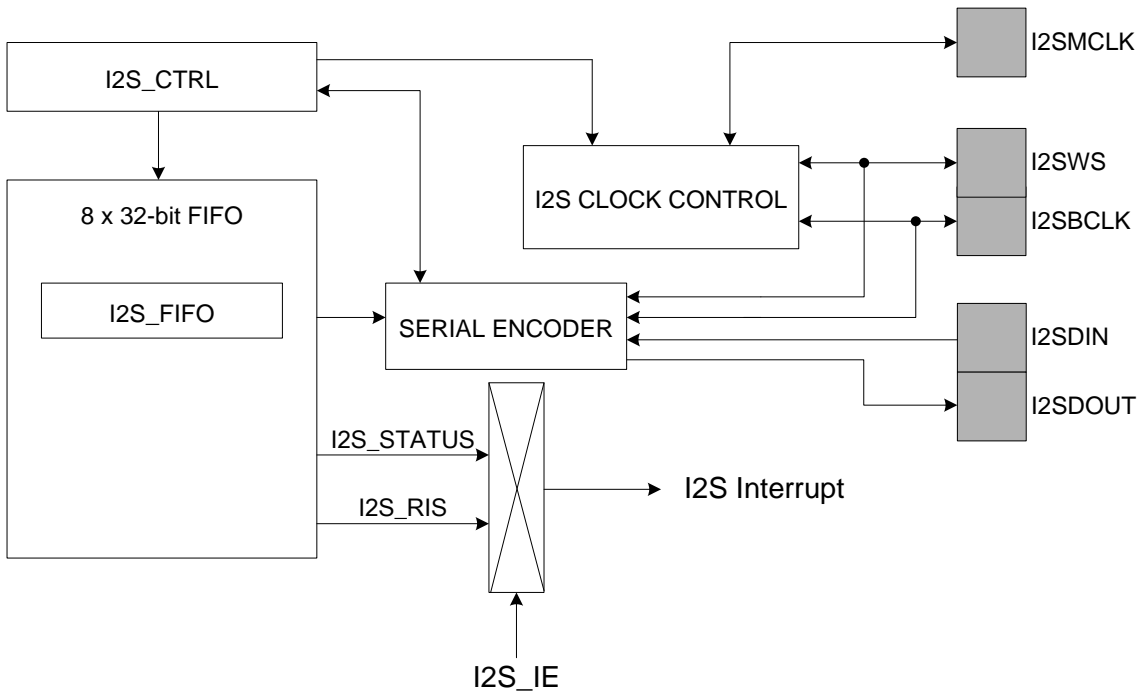
| Pin Name  | Type | Description                                     | GPIO Configuration    |
|-----------|------|---|-----------------------|
| AUXTALOUT | I/O  | External high-speed X'tal output pin for audio. | Depends on GPIOOn_CFG |
| AUXTALIN  | I/O  | External high-speed X'tal input pin for audio.  | Depends on GPIOOn_CFG |

## 13.4 BLOCK DIAGRAM

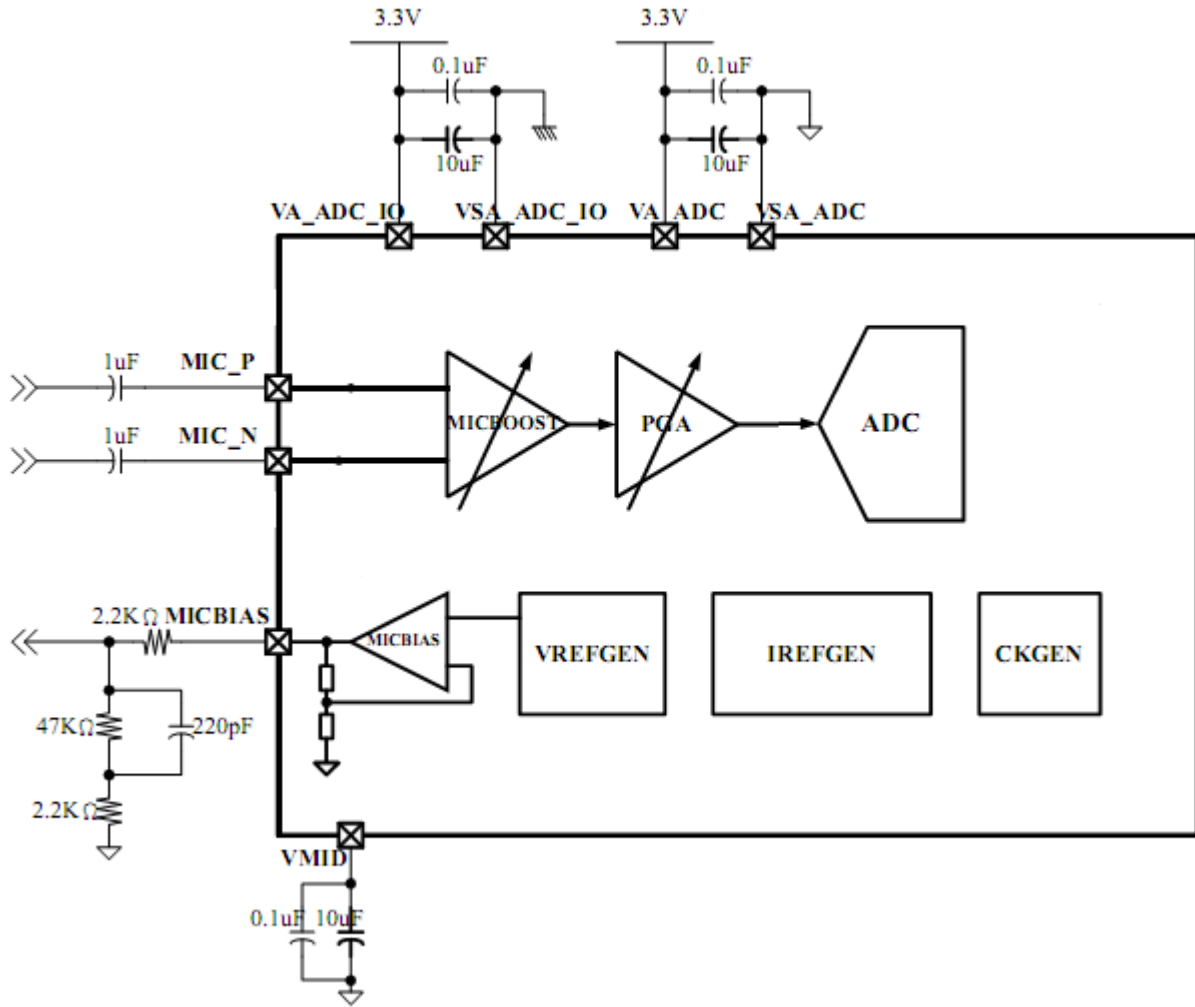
### 13.4.1 I2S CLCOK CONTROL



### 13.4.2 I2S BLOCK DIAGRAM



**13.4.3 16-Bit Sigma-Delta ADC BLOCK DIAGRAM**





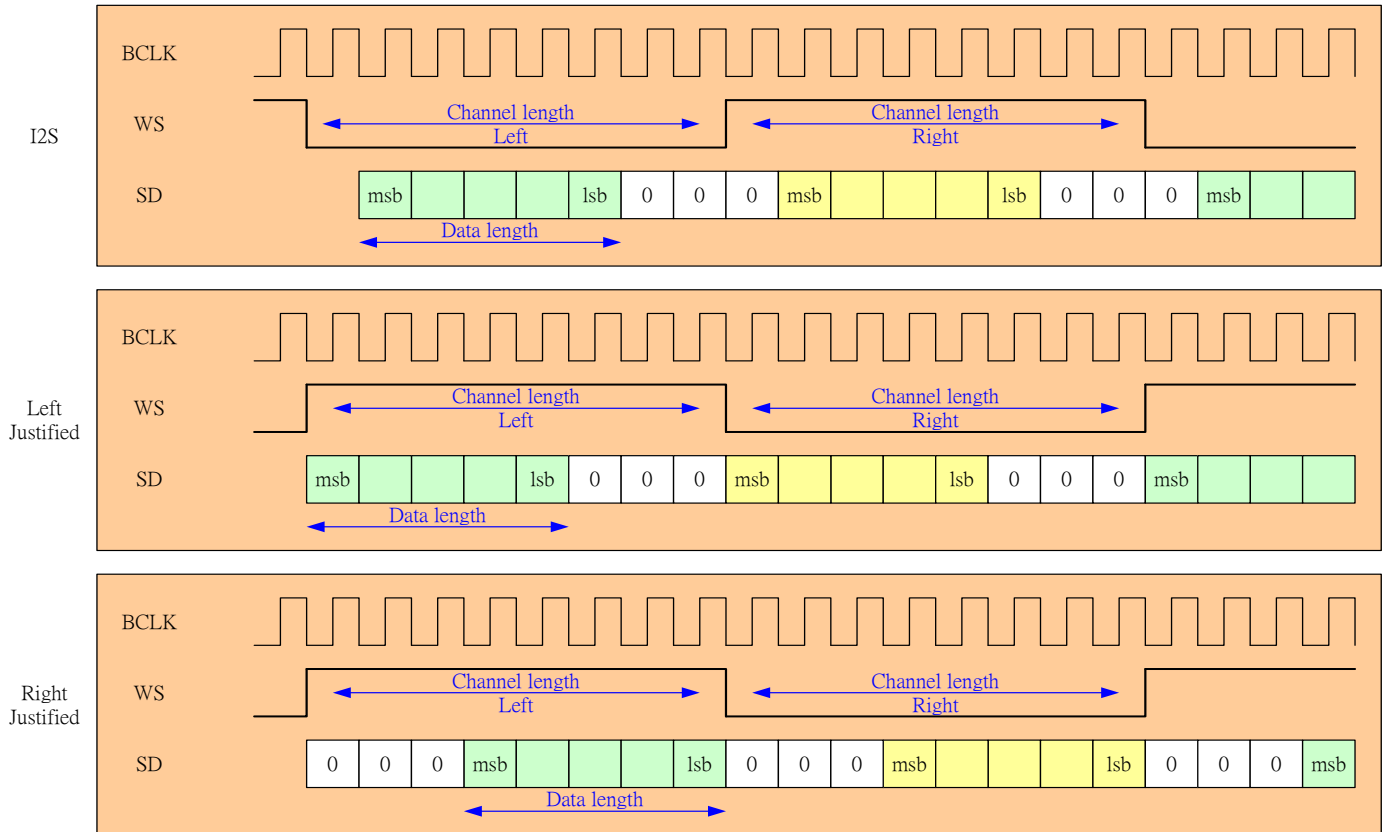


## 13.5 FUNCTIONAL DESCRIPTION

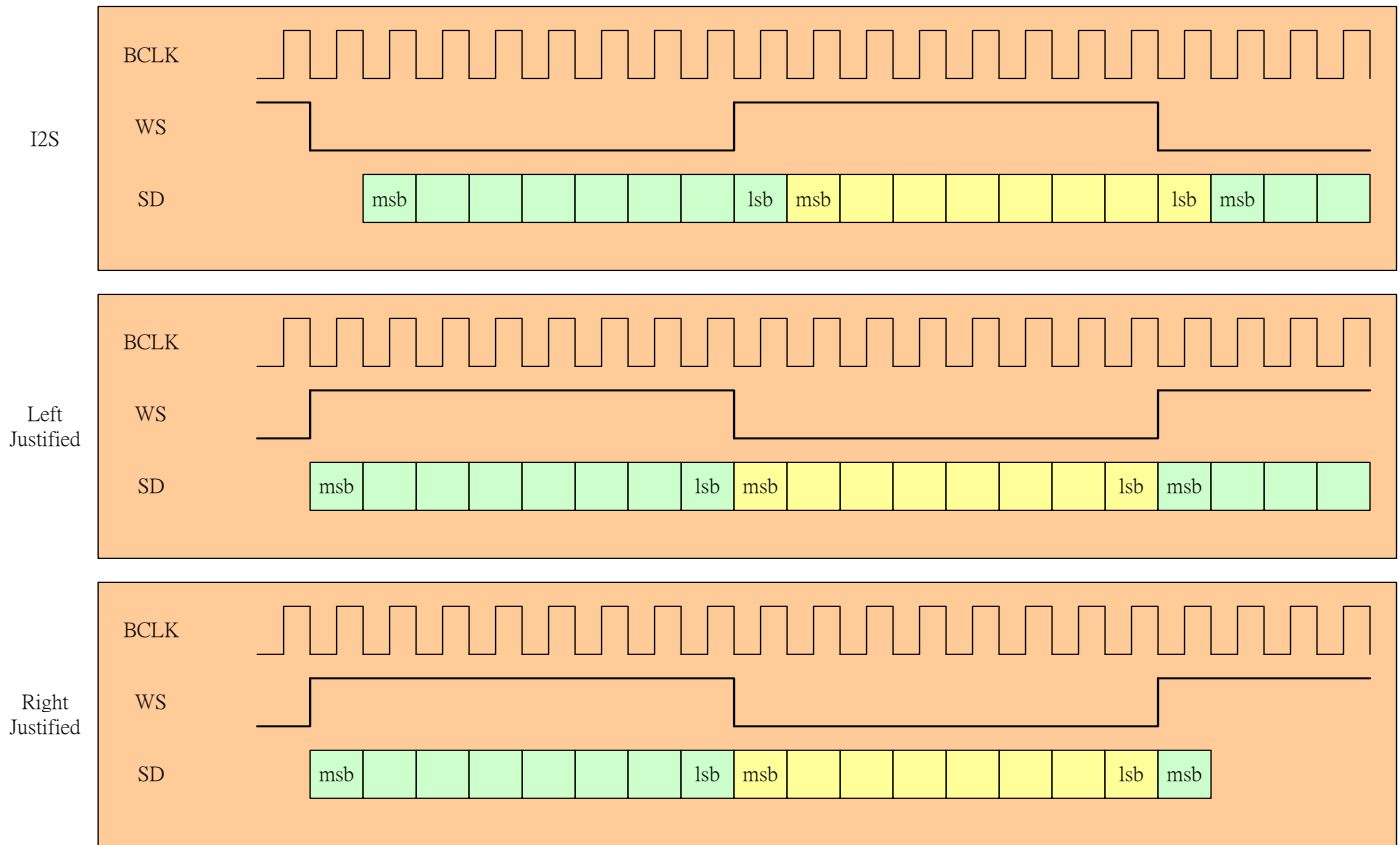
### 13.5.1 I2S OPERATION

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

Channel Length > Data Length:



Channel Length = Data Length



## 13.5.2 I2S FIFO OPERAION

### 13.5.2.1 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 |

### 13.5.2.2 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 |

## 13.6 I2S REGISTERS

Base Address: 0x4001 A000

### 13.6.1 I2S Control register (I2S\_CTRL)

Address Offset: 0x00

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

| Bit   | Name          | Description  | Attribute | Reset |
|-------|---------------|--|-----------|-------|
| 31    | I2SEN         | I2S enable bit<br>0: Disable<br>1: Enable I2S.   | R/W       | 0     |
| 30    | I2SMOD        | I2S mode select bit<br>0: I2S mode for external I2S interface. (If I2SEN=1 and I2SMOD=0, HW will switch GPIO to DIN, DOUT, MCLK, BCLK, and WS.)<br>1: Codec mode for internal I2S interface connected to ADC and DAC. (If I2SEN=1 and I2SMOD=1, HW will switch channel length=32bits, BCLK=MCLK/4, Standard I2S format, I2S master, and I2S mono mode) | R/W       | 0     |
| 29:25 | Reserved      |  | R         | 0     |
| 24:20 | CHLENGTH[4:0] | Bit number of single channel = CHLENGTH[4:0]+1.<br>0-6: Reserved<br>7: 8 bits<br>8: 9 bits<br>...<br>...<br>31: 32bits (Max)<br><br>(If I2SEN=1, and I2SMOD=1, HW will switch channel length=32bits)   | R/W       | 0x1F  |
| 19    | Reserved      |  | R         | 0     |
| 18:16 | RXFIFOTH[2:0] | RX FIFO Threshold level<br>0: RX FIFO threshold level = 0<br>1: RX FIFO threshold level = 1<br>...<br>...<br>n: RX FIFO threshold level = n  | R/W       | 0x3   |
| 15    | Reserved      |  | R         | 0     |
| 14:12 | TXFIFOTH[2:0] | TX FIFO Threshold level<br>0: TX FIFO threshold level = 0<br>1: TX FIFO threshold level = 1<br>...<br>...<br>n: TX FIFO threshold level = n  | R/W       | 0x3   |
| 11:10 | DL[1:0]       | Data Length<br>00: 8 bits<br>01: 16 bits<br>10: 24 bits<br>11: 32 bits   | R/W       | 0x1   |
| 9     | CLRRXFIFO     | Clear I2S RX FIFO<br>0: No effect.<br>1: Reset RX FIFO(RXFIFOLV bit becomes 0, RXFIFOEMPTY bit becomes 1, Data in RX FIFO will be cleared). This bit returns "0" automatically   | W         | 0     |
| 8     | CLRTXFIFO     | Clear I2S TX FIFO<br>0: No effect.<br>1: Reset TX FIFO(TXFIFOLV bit becomes 0, TXFIFOEMPTY bit becomes 1, Data in TX FIFO will be cleared). This bit returns "0" automatically   | W         | 0     |
| 7     | RXEN          | Receiver enable bit<br>0: Disable<br>1: Enable   | R/W       | 0     |
| 6     | TXEN          | Transmit enable bit<br>0: Disable  | R/W       | 0     |

|     |             |  |     |   |
|-----|-------------|--|-----|---|
| 5:4 | FORMAT[1:0] | 1: Enable<br>I2S operation format.<br>00: Standard I2S format<br>01: Left-justified format<br>10: Right(MSB)-justified format<br>11: Reserved<br><br>(If I2SEN=1 and I2SMOD=1, HW will switch Standard I2S format)   | R/W | 0 |
| 3   | MS          | Master/Slave selection bit<br>0: Act as Master using internally generated BCLK and WS signals.<br>1: Act as Slave using externally BCLK and WS signals.<br><br>(If I2SEN=1 and I2SMOD=1, HW will switch Master mode) | R/W | 0 |
| 2   | MONO        | Mono/Stereo selection bit<br>0: Stereo<br>1: Mono<br><br>(If I2SEN=1 and I2SMOD=1, HW will switch Mono mode)   | R/W | 0 |
| 1   | MUTE        | Mute enable bit<br>0: Disable Mute<br>1: Enable. I2SSDA Output = 0   | R/W | 0 |
| 0   | START       | Start Transmit/Receive bit.<br>0: Stop Transmit/Receive<br>1: Start Transmit/Receive   | R/W | 0 |

### 13.6.2 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<br>0: BCLK = MCLK / 2<br>1: BCLK = MCLK / 4<br>2: BCLK = MCLK / 6<br>3: BCLK = MCLK / 8<br>...<br>...<br>n: BCLK = MCLK / (2*n + 2)<br><br>(If I2SEN=1, and I2SMOD=1, HW will switch BCLK=MCLK/4)  | R/W       | 1     |
| 7:5   | Reserved     |   | R         | 0     |
| 4     | MCLKSEL      | MCLK source selection bit<br>0: MCLK source of master is from I2S_PCLK<br>1: MCLK source of master is from External high-speed X'tal for audio(AUXTALOUT/AUXTALIN) and Audio Clock Prescaler out(AUEHSPRE[2:0]) | R/W       | 0     |
| 3     | MCKOEN       | MCLK output enable bit<br>0: Disable<br>1: Enable   | R/W       | 0     |
| 2:0   | MCLKDIV[2:0] | MCLK divider<br>0: MCLK = MCLK source<br>1: MCLK = MCLK source / 2<br>2: MCLK = MCLK source / 4<br>...<br>...<br>n: MCLK = MCLK source / (2*n), n>0   | R/W       | 0     |

### 13.6.3 I2S Status register (I2S\_STATUS)

Address Offset: 0x08

| Bit   | Name          | Description   | Attribute | Reset |
|-------|---------------|---|-----------|-------|
| 31:21 | Reserved      |   | R         | 0     |
| 20:17 | RXFIFOLV[3:0] | RX FIFO used level<br>0000: 0/8 RX FIFO is used (Empty) | R         | 0     |

|              |               |  |   |   |
|--------------|---------------|--|---|---|
|              |               | 0001: 1/8 RX FIFO is used<br>0010: 2/8 RX FIFO is used<br>...<br>...<br>1000: 8/8 RX FIFO is used (Full)<br>Other: Reserved  |   |   |
| <b>16</b>    | Reserved      |  | R | 0 |
| <b>15:12</b> | TXFIFOLV[3:0] | TX FIFO used level<br>0000: 0/8 TX FIFO is used (Empty)<br>0001: 1/8 TX FIFO is used<br>0010: 2/8 TX FIFO is used<br>...<br>...<br>1000: 8/8 TX FIFO is used (Full)<br>Other: Reserved | R | 0 |
| <b>11</b>    | RXFIFOEMPTY   | RX FIFO empty flag<br>0: RX FIFO is not empty.<br>1: RX FIFO is empty. Data read from RX FIFO will be zero.  | R | 1 |
| <b>10</b>    | TXFIFOEMPTY   | TX FIFO empty flag<br>0: TX FIFO is not empty.<br>1: TX FIFO is empty.   | R | 1 |
| <b>9</b>     | RXFIFOFULL    | RX FIFO full flag<br>0: RX FIFO is not full.<br>1: RX FIFO is full.  | R | 0 |
| <b>8</b>     | TXFIFOFULL    | TX FIFO full flag<br>0: TX FIFO is not full.<br>1: TX FIFO is full. Write operation to TX FIFO will be ignored.  | R | 0 |
| <b>7</b>     | RXFIFOTHF     | RX FIFO threshold flag<br>0: RXFIFOLV $\leq$ RXFIFOTH<br>1: RXFIFOLV > RXFIFOTH  | R | 0 |
| <b>6</b>     | TXFIFOTHF     | TX FIFO threshold flag<br>0: TXFIFOLV $\geq$ TXFIFOTH<br>1: TXFIFOLV < TXFIFOTH  | R | 1 |
| <b>5:2</b>   | Reserved      |  | R | 0 |
| <b>1</b>     | RIGHTCH       | Current channel status<br>0: Current channel is Left channel<br>1: Current channel is Right channel  | R | 1 |
| <b>0</b>     | I2SINT        | I2S interrupt flag<br>0: No I2S interrupt<br>1: I2S interrupt occurs.  | R | 0 |

### 13.6.4 I2S Interrupt Enable register (I2S\_IE)

Address Offset: 0x0C

| Bit         | Name        | Description   | Attribute | Reset |
|-------------|-------------|---|-----------|-------|
| <b>31:8</b> | Reserved    |   | R         | 0     |
| <b>7</b>    | RXFIFOTHIE  | RX FIFO threshold interrupt enable bit<br>0: Disable<br>1: Enable | R/W       | 0     |
| <b>6</b>    | TXFIFOTHIE  | TX FIFO threshold interrupt enable bit<br>0: Disable<br>1: Enable | R/W       | 0     |
| <b>5</b>    | RXFIFOUDFIE | RX FIFO underflow interrupt enable bit<br>0: Disable<br>1: Enable | R/W       | 0     |
| <b>4</b>    | TXFIFOOVFIE | TX FIFO overflow interrupt enable bit<br>0: Disable<br>1: Enable  | R/W       | 0     |
| <b>3:0</b>  | Reserved    |   | R         | 0     |

### 13.6.5 I2S Raw Interrupt Status register (I2S\_RIS)

Address Offset: 0x10

| Bit  | Name       | Description  | Attribute | Reset |
|------|------------|--|-----------|-------|
| 31:8 | Reserved   |  | R         | 0     |
| 7    | RXFIFOTHIF | RX FIFO threshold interrupt flag<br>0: No RX FIFO threshold interrupt<br>1: RX FIFO threshold triggered.                     | R         | 0     |
| 6    | TXFIFOTHIF | TX FIFO threshold interrupt flag<br>0: No TX FIFO threshold interrupt<br>1: TX FIFO threshold triggered.                     | R         | 0     |
| 5    | RXFIFOUDIF | RX FIFO underflow interrupt flag<br>0: No RX FIFO underflow<br>1: RX FIFO underflow (RX FIFO is empty and still being read). | R         | 0     |
| 4    | TXFIFOOVIF | TX FIFO overflow interrupt flag<br>0: No TX FIFO overflow<br>1: TX FIFO overflow (TX FIFO is full and still being written).  | R         | 0     |
| 3:0  | Reserved   |  | R         | 0     |

### 13.6.6 I2S Interrupt Clear register (I2S\_IC)

Address Offset: 0x14

| Bit  | Name       | Description                             | Attribute | Reset |
|------|------------|---|-----------|-------|
| 31:8 | Reserved   |   | R         | 0     |
| 7    | RXFIFOTHIC | 0: No effect<br>1: Clear RXFIFOTHIF bit | W         | 0     |
| 6    | TXFIFOTHIC | 0: No effect<br>1: Clear TXFIFOTHIF bit | W         | 0     |
| 5    | RXFIFOUDIC | 0: No effect<br>1: Clear RXFIFOUDIF bit | W         | 0     |
| 4    | TXFIFOOVIC | 0: No effect<br>1: Clear TXFIFOOVIF bit | W         | 0     |
| 3:0  | Reserved   |   | R         | 0     |

### 13.6.7 I2S RX FIFO register (I2S\_RXFIFO)

Address Offset: 0x18

| Bit  | Name         | Description        | Attribute | Reset |
|------|--------------|--------------------|-----------|-------|
| 31:0 | RXFIFO[31:0] | 8 x 32-bit RX FIFO | R         | 0     |

### 13.6.8 I2S TX FIFO register (I2S\_TXFIFO)

Address Offset: 0x1C

| Bit  | Name         | Description        | Attribute | Reset |
|------|--------------|--------------------|-----------|-------|
| 31:0 | TXFIFO[31:0] | 8 x 32-bit TX FIFO | W         | 0     |

## 13.7 CODEC ADC REGISTERS

Base Address: 0x4006 4000

\* **Note:** Codec ADC Registers are available only when codec mode is selected by I2SMOD=1.

### 13.7.1 ADC Setting 1 register (ADC\_SET1)

Address Offset: 0x540

| Bit  | Name     | Description | Attribute | Reset |
|------|----------|-------------|-----------|-------|
| 31:8 | Reserved |             | R         | 0     |

|            |      |  |     |      |
|------------|------|--|-----|------|
| <b>7:0</b> | LB_H | AGC Control.<br>Low bound setting for output amplitude of ADC: High byte | R/W | 0x40 |
|------------|------|--|-----|------|

### 13.7.2 ADC Setting 2 register (ADC\_SET2)

Address Offset: 0x550

| Bit         | Name     | Description   | Attribute | Reset |
|-------------|----------|---|-----------|-------|
| <b>31:8</b> | Reserved |   | R         | 0     |
| <b>7:0</b>  | LB_L     | AGC Control.<br>Low bound setting for output amplitude of ADC: Low byte | R/W       | 0x00  |

### 13.7.3 ADC Setting 3 register (ADC\_SET3)

Address Offset: 0x560

| Bit         | Name     | Description   | Attribute | Reset |
|-------------|----------|---|-----------|-------|
| <b>31:8</b> | Reserved |   | R         | 0     |
| <b>7:0</b>  | HB_H     | AGC Control.<br>High bound setting for output amplitude of ADC: High byte | R/W       | 0x60  |

### 13.7.4 ADC Setting 4 register (ADC\_SET4)

Address Offset: 0x570

| Bit         | Name     | Description  | Attribute | Reset |
|-------------|----------|--|-----------|-------|
| <b>31:8</b> | Reserved |  | R         | 0     |
| <b>7:0</b>  | HB_L     | AGC Control.<br>High bound setting for output amplitude of ADC: Low byte | R/W       | 0x00  |

### 13.7.5 ADC Setting 5 register (ADC\_SET5)

Address Offset: 0x580

| Bit         | Name     | Description  | Attribute | Reset |
|-------------|----------|--|-----------|-------|
| <b>31:4</b> | Reserved |  | R         | 0     |
| <b>3:0</b>  | NOR_POD  | AGC Control.<br>The period of gain update at normal mode when AGC is enabled.<br>Fs : Audio sampling rate<br>0000: 1/Fs x 2 <sup>(0)</sup><br>0001: 1/Fs x 2 <sup>(1)</sup><br>.....<br>1110: 1/Fs x 2 <sup>(14)</sup><br>1111: 1/Fs x 2 <sup>(15)</sup> | R/W       | 0x05  |

### 13.7.6 ADC Setting 6 register (ADC\_SET6)

Address Offset: 0x590

| Bit         | Name     | Description  | Attribute | Reset |
|-------------|----------|--|-----------|-------|
| <b>31:4</b> | Reserved |  | R         | 0     |
| <b>3:0</b>  | MUTE_POD | AGC Control.<br>The period of gain update at mute mode when AGC is enabled.<br>Fs : Audio sampling rate<br>0000: 1/Fs x 2 <sup>(0)</sup><br>0001: 1/Fs x 2 <sup>(1)</sup><br>.....<br>1110: 1/Fs x 2 <sup>(14)</sup><br>1111: 1/Fs x 2 <sup>(15)</sup> | R/W       | 0x0B  |

### 13.7.7 ADC Setting 7 register (ADC\_SET7)

Address Offset: 0x5A0

| Bit         | Name     | Description | Attribute | Reset |
|-------------|----------|-------------|-----------|-------|
| <b>31:8</b> | Reserved |             | R         | 0     |



|            |             |   |     |      |
|------------|-------------|---|-----|------|
| <b>7:0</b> | SEARCH_TH_H | AGC Control.<br>Threshold for activating AGC. High byte.<br>Once ADC output is larger than the threshold, AGC will update internal digital gain until the output amplitude is between high bound and low bound. | R/W | 0x03 |
|------------|-------------|---|-----|------|

### 13.7.8 ADC Setting 8 register (ADC\_SET8)

Address Offset: 0x5B0

| Bit         | Name        | Description   | Attribute | Reset |
|-------------|-------------|---|-----------|-------|
| <b>31:8</b> | Reserved    |   | R         | 0     |
| <b>7:0</b>  | SEARCH_TH_L | AGC Control.<br>Threshold for activating AGC. Low byte. | R/W       | 0x00  |

### 13.7.9 ADC Setting 9 register (ADC\_SET9)

Address Offset: 0x5C0

| Bit         | Name      | Description   | Attribute | Reset |
|-------------|-----------|---|-----------|-------|
| <b>31:8</b> | Reserved  |   | R         | 0     |
| <b>7:0</b>  | MUTE_TH_H | AGC Control.<br>Threshold for inactivating AGC. High byte<br>Once ADC output is lower than the threshold ( <i>the check period setting is MUTE_CAL_POD in ADC_SET11</i> ), AGC will update internal digital gain until the PGA and Boost setting are equal to the setting in ADC_SET19. Then the internal digital gain is adjusted to default value. Finally, the AGC is entering mute mode until the ADC output is over search threshold in ADC_SET7 & ADC_SET8. | R/W       | 0x10  |

### 13.7.10 ADC Setting 10 register (ADC\_SET10)

Address Offset: 0x5D0

| Bit         | Name      | Description  | Attribute | Reset |
|-------------|-----------|--|-----------|-------|
| <b>31:8</b> | Reserved  |  | R         | 0     |
| <b>7:0</b>  | MUTE_TH_L | AGC Control.<br>Threshold for inactivating AGC. Low byte | R/W       | 0x00  |

### 13.7.11 ADC Setting 11 register (ADC\_SET11)

Address Offset: 0x5E0

| Bit         | Name         | Description   | Attribute | Reset |
|-------------|--------------|---|-----------|-------|
| <b>31:4</b> | Reserved     |   | R         | 0     |
| <b>3:0</b>  | MUTE_CAL_POD | AGC Control.<br>The calculating period for inactivating AGC.<br>Fs : Audio sampling rate<br>0000: 256/Fs x 2 <sup>(0)</sup><br>0001: 256/Fs x 2 <sup>(1)</sup><br>.....<br>1110: 256/Fs x 2 <sup>(14)</sup><br>1111: 256/Fs x 2 <sup>(15)</sup><br>AGC will check whether entering mute mode or not according to mute threshold per (MUTE_CAL_POD) seconds. | R/W       | 0x09  |

### 13.7.12 ADC Setting 12 register (ADC\_SET12)

Address Offset: 0x5F0

| Bit         | Name     | Description   | Attribute | Reset |
|-------------|----------|---|-----------|-------|
| <b>31:4</b> | Reserved |   | R         | 0     |
| <b>3:0</b>  | SAT_TH   | AGC Control.<br>Threshold for ADC saturation condition.<br>The saturation condition is for sigma-delta ADC, if there are more than (SAT_TH) bit streams are saturating, then the internal gain will be adjusted rapidly to avoid the ADC output saturation. | R/W       | 0x03  |

### 13.7.13 ADC Setting 13 register (ADC\_SET13)

Address Offset: 0x600

| Bit  | Name     | Description   | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:4 | Reserved |   | R         | 0     |
| 3:0  | SAT_POD  | AGC Control.<br>The detection period for ADC saturation condition.<br>Fs : Sampling rate<br>0000: 1/Fs x 2 <sup>(0)</sup><br>0001: 1/Fs x 2 <sup>(1)</sup><br>.....<br>1110: 1/Fs x 2 <sup>(14)</sup><br>1111: 1/Fs x 2 <sup>(15)</sup> | R/W       | 0x0A  |

### 13.7.14 ADC Setting 14 register (ADC\_SET14)

Address Offset: 0x610

| Bit  | Name          | Description   | Attribute | Reset |
|------|---------------|---|-----------|-------|
| 31:8 | Reserved      |   | R         | 0     |
| 7    | AGC_OFF       | AGC Control<br>AGC function<br>0: Enable<br>1: Disable  | R/W       | 0x01  |
| 6:5  | BOOST_SET_VAL | AGC Control<br>Boost setting value at normal mode when AGC is on.<br>00: +0dB<br>01: +12dB<br>10: +20dB<br>11: +30dB  | R/W       | 0x03  |
| 4:0  | PGA_SET_VAL   | AGC Control.<br>PGA setting value at normal mode when AGC is on (1.5dB/step).<br>00000: Mute<br>00001: -12dB<br>01001: 0dB<br>.....<br>11110: +31.5dB<br>11111: +33dB | R/W       | 0x10  |

### 13.7.15 ADC Setting 15 register (ADC\_SET15)

Address Offset: 0x620

| Bit  | Name     | Description  | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:3 | Reserved |  | R         | 0     |
| 2    | ACTIVE   | Digital Audio Interface Control.<br>0: Disable<br>1: Enable                              | R/W       | 0x01  |
| 1:0  | IWL      | Word length of DA interface.<br>00: 16-bits<br>01: 18-bits<br>10: 20-bits<br>11: 24-bits | R/W       | 0x03  |

### 13.7.16 ADC Setting 16 register (ADC\_SET16)

Address Offset: 0x630

| Bit  | Name     | Description  | Attribute | Reset |
|------|----------|--|-----------|-------|
| 31:7 | Reserved |  | R         | 0     |
| 6:5  | BOOST    | Boost setting value when AGC is off<br>00: +0dB<br>01: +12dB<br>10: +20dB<br>11: +30dB | R/W       | 0x00  |

|            |     |  |     |      |
|------------|-----|--|-----|------|
| <b>4:0</b> | PGA | PGA setting value when AGC is off<br>00000: Mute<br>00001: -12dB<br>.....<br>01001: 0dB<br>.....<br>11110: +31.5dB<br>11111: +33dB | R/W | 0x09 |
|------------|-----|--|-----|------|

### 13.7.17 ADC Setting 18 register (ADC\_SET18)

Address Offset: 0x650

| Bit         | Name      | Description  | Attribute | Reset |
|-------------|-----------|--|-----------|-------|
| <b>31:8</b> | Reserved  |  | R         | 0     |
| <b>7:4</b>  | VOL_CTRL  | Digital Volume attenuation control. At the normal mode when AGC is on or off.<br>0000: 0dB<br>0001: -3dB<br>0010: -6dB<br>0011: -9dB<br>0100: -12dB<br>0101: -15dB<br>0110: -18dB<br>0111: -21dB<br>1000: -24dB<br>1001: -27dB<br>1010: -30dB<br>1011: -36dB<br>1100: -42dB<br>1101: -48dB<br>1110: -54dB<br>1111: -78dB | R/W       | 0x00  |
| <b>3:0</b>  | MUTE_CTRL | Digital Volume attenuation control. At the mute mode when AGC is on.<br>0000: 0dB<br>0001: -3dB<br>0010: -6dB<br>0011: -9dB<br>0100: -12dB<br>0101: -15dB<br>0110: -18dB<br>0111: -21dB<br>1000: -24dB<br>1001: -27dB<br>1010: -30dB<br>1011: -36dB<br>1100: -42dB<br>1101: -48dB<br>1110: -54dB<br>1111: -78dB          | R/W       | 0x00  |

### 13.7.18 ADC Setting 19 register (ADC\_SET19)

Address Offset: 0x660

| Bit         | Name           | Description  | Attribute | Reset |
|-------------|----------------|--|-----------|-------|
| <b>31:7</b> | Reserved       |  | R         | 0     |
| <b>6:5</b>  | BOOST_MUTE_VAL | AGC Control.<br>Boost setting value at mute mode when AGC is enabled.<br>00: +0dB<br>01: +12dB<br>10: +20dB<br>11: +30dB | R/W       | 0x03  |
| <b>4:0</b>  | PGA_MUTE_VAL   | AGC Control.<br>PGA setting value at mute mode when AGC is enabled.<br>00000: Mute<br>00001: -12dB<br>.....              | R/W       | 0x10  |

|  |  |   |  |  |
|--|--|---|--|--|
|  |  | 01001: 0dB<br>.....<br>11110: +31.5dB<br>11111: +33dB |  |  |
|--|--|---|--|--|

### 13.7.19 ADC Setting 20 register (ADC\_SET20)

Address Offset: 0x670

| Bit  | Name         | Description   | Attribute | Reset |
|------|--------------|---|-----------|-------|
| 31:8 | Reserved     |   | R         | 0     |
| 7:0  | VOL_MUTE_POD | The updating period for Digital Volume attenuation at the normal/mute mode when AGC is on.<br>Bit[7:4]: for normal mode.<br>Bit[3:0]: for mute mode.<br>Fs : Sampling rate<br>0000: 1/Fs x 2 <sup>(0)</sup><br>0001: 1/Fs x 2 <sup>(1)</sup><br>.....<br>1110: 1/Fs x 2 <sup>(14)</sup><br>1111: 1/Fs x 2 <sup>(15)</sup><br>The volume update when mode transition. This is mainly for avoiding large variation of analog gain setting during the transition of normal mode and mute mode. | R/W       | 0x12  |

### 13.7.20 ADC Setting 21 register (ADC\_SET21)

Address Offset: 0x6B0

| Bit  | Name     | Description                           | Attribute | Reset |
|------|----------|---------------------------------------|-----------|-------|
| 31:8 | Reserved |                                       | R         | 0     |
| 7    | ADC_EN   | ADC power-on enable, active High      | R/W       | 0     |
| 6    | Reserved |                                       | R         | 0     |
| 5    | MICBT_EN | MICBOOST power-on enable, active High | R/W       | 0     |
| 4    | PGA_EN   | PGA power-on enable, active High      | R/W       | 0     |
| 3:0  | Reserved |                                       | R         | 0     |

### 13.7.21 ADC Setting 22 register (ADC\_SET22)

Address Offset: 0x6C0

| Bit  | Name     | Description                         | Attribute | Reset |
|------|----------|-------------------------------------|-----------|-------|
| 31:4 | Reserved |                                     | R         | 0     |
| 3    | IREF_EN  | IREF circuit enable, active High    | R/W       | 0     |
| 2    | VREF_EN  | VREF circuit enable, active High    | R/W       | 0     |
| 1    | MICB_EN  | Microphone bias enable, active High | R/W       | 0     |
| 0    | CK_EN    | CKGEN enable, active High           | R/W       | 0     |

### 13.7.22 ADC Setting 23 register (ADC\_SET23)

Address Offset: 0x6D0

| Bit  | Name        | Description   | Attribute | Reset |
|------|-------------|---|-----------|-------|
| 31:5 | Reserved    |   | R         | 0     |
| 4    | SEL_MICB    | Microphone Bias Output select<br>0: 0.8*VA<br>1: 0.9*VA   | R/W       | 0     |
| 3    | Reserved    |   | R         | 0     |
| 2    | SEL_MIC     | P1.7/MIC_P and P1.8/MIC_N function selection<br>0: General purpose IO<br>1: Microphone Differential input when ADC is enabled | R/W       | 0     |
| 1    | Reserved    |   | R         | 0     |
| 0    | SEL_MIX_MIC | Microphone input path to mixer enable<br>0: Disable<br>1: Enable  | R/W       | 1     |

### 13.7.23 ADC Setting 24 register (ADC\_SET24)

Address Offset: 0x6E0

| Bit  | Name      | Description                         | Attribute | Reset |
|------|-----------|-------------------------------------|-----------|-------|
| 31:7 | Reserved  |                                     | R         | 0     |
| 6:5  | BOOST_AGC | Boost setting value when AGC is on. | R         | -     |
| 4:0  | PGA_AGC   | PGA setting value when AGC is on.   | R         | -     |

## 13.8 CODEC DAC REGISTERS

Base Address: 0x4006 5000

\* *Note: Codec DAC Registers are available only when codec mode is selected by I2SMOD=1.*

### 13.8.1 DAC Setting 1 register (DAC\_SET1)

Address Offset: 0x000

| Bit  | Name     | Description   | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:8 | Reserved |   | R         | 0     |
| 7    | PD_DAC   | DAC Power-down, active High   | R/W       | 1     |
| 6    | PD_CLK   | CKGEN Power-down, active High   | R/W       | 1     |
| 5    | PD_IREF  | IREF Circuit Power-down, active High  | R/W       | 1     |
| 4:3  | Reserved |   | R         | 0     |
| 2    | PD_VREF  | VREF Circuit Power-down, active High  | R/W       | 1     |
| 1    | VMIDSEL  | Normal mode/Fast Start-up select<br>0 : Normal mode<br>1 : Fast Start-up mode | R/W       | 0     |
| 0    | Reserved |   | R         | 0     |

### 13.8.2 DAC Setting 2 register (DAC\_SET2)

Address Offset: 0x010

| Bit  | Name      | Description  | Attribute | Reset |
|------|-----------|--|-----------|-------|
| 31:8 | Reserved  |  | R         | 0     |
| 7:6  | RMP[1:0]  | Attenuation ramp rate<br>T= LRCK clock<br>RMP = 00 1T<br>RMP = 01 2T<br>RMP = 10 4T<br>RMP = 11 8T<br>If VOL[7:0] in DAC_SET3 is changed, then after (RMP) seconds, the DAC output will start updating the volume. | R/W       | 0     |
| 5:3  | Reserved  |  | R         | 0     |
| 2    | MUTX      | Mute ON/OFF<br>1: Mute on<br>0 : Mute off  | R/W       | 1     |
| 1    | DAC_EN_IN | DAC Enable<br>1: Enable<br>0: Disable  | R/W       | 0     |
| 0    | SOFT_RSTN | Software reset digital circuit . low reset . one MCLK pulse trigger  | R/W       | 1     |

### 13.8.3 DAC Setting 3 register (DAC\_SET3)

Address Offset: 0x020

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

|             |          |   |     |   |
|-------------|----------|---|-----|---|
| <b>31:8</b> | Reserved |   | R   | 0 |
| <b>7:0</b>  | VOL[7:0] | Digital volume attenuation, -0.5dB/step<br>0x00: 0dB<br>0x01: -0.5dB<br>0x02: -1dB<br>.....<br>0x7E: -63dB<br>0x7F: -63.5dB | R/W | 0 |

### 13.8.4 DAC Setting 4 register (DAC\_SET4)

Address Offset: 0x030

| Bit         | Name       | Description   | Attribute | Reset |
|-------------|------------|---|-----------|-------|
| <b>31:8</b> | Reserved   |   | R         | 0     |
| <b>7</b>    | PD_DRV     | DAC driver power down   | R/W       | 1     |
| <b>6:3</b>  | Reserved   |   | R         | 0     |
| <b>2:1</b>  | DEMS[1:0]  | Select the DAC de-emphasis response curve<br>0: Reserved<br>1: De-emphasis for 48 kHz<br>2: De-emphasis for 44.1 kHz<br>3: De-emphasis for 32 kHz | R/W       | 0     |
| <b>0</b>    | INI_RAM_EN | Initialize DAC RAM,<br>set 1, until Ini_RAM_Ready =1, then clear this bit.  | R/W       | 0     |

### 13.8.5 DAC Status register (DAC\_STATUS)

Address Offset: 0x040

| Bit         | Name          | Description   | Attribute | Reset |
|-------------|---------------|---|-----------|-------|
| <b>31:1</b> | Reserved      |   | R         | 0     |
| <b>0</b>    | Ini_RAM_Ready | Status for checking whether the ram initialization is ready or not. | R         | 0     |

## 13.9 Sigma-delta ADC Control Flow

### 13.9.1 Sigma-delta ADC Power-up Sequence

Step1: IREF\_EN = 1

Step2: VREF\_EN = 1

Step3: MICBOOST\_EN = 1

Step4: PGA\_EN = 1

Step5: ADC\_EN = 1

Step6: CK\_EN = 1

Step7: MICB\_EN = 1

### 13.9.2 Sigma-delta ADC Power-down Sequence

Step1: ADC\_EN = 0

Step2: PGA\_EN = 0

Step3: MICBOOST\_EN = 0

Step4: MICB\_EN = 0

Step5: VREF\_EN = 0

Step6: IREF\_EN = 0

Step7: CK\_EN = 0

### **13.9.3 Sigma-delta ADC Enable Sequence**

Step1: ADC Analog Enable (Sigma-delta ADC Power-Up Sequence)

Step2: ADC Digital Enable (ACTIVE = 1)

Step3: Delay 21us for ADC setup time

Step4: MCLK Output Enable (MCLKOEN = 1 and I2S Enable)

## **13.10 Sigma-delta DAC Control Flow**

### **13.10.1 Sigma-delta DAC Power-up Sequence**

Step1: PD\_IREF = 0

Step2: VMIDSEL = 1

Step3: PD\_VREF = 0

Step4: VMIDSEL = 0

Step5: PD\_DAC = 0

Step6: PD\_CLK = 0

Step7: After PD\_CLK, delay 6.3us

Step8: After I2S data ready, delay 125ms

Step9: PD\_DRV = 0

### **13.10.2 Sigma-delta DAC Power-down Sequence**

Step1: PD\_DRV = 1

Step2: PD\_DAC = 1

Step3: VMIDSEL = 0

Step4: PD\_VREF = 1

Step5: PD\_IREF = 1

Step6: PD\_CLK = 1

### **13.10.3 Sigma-delta DAC Enable Sequence**

Step1: DAC Digital Enable (DAC\_EN\_IN = 1)

Step2: MCLK Output Enable (MCLKOEN = 1 and I2S Enable)

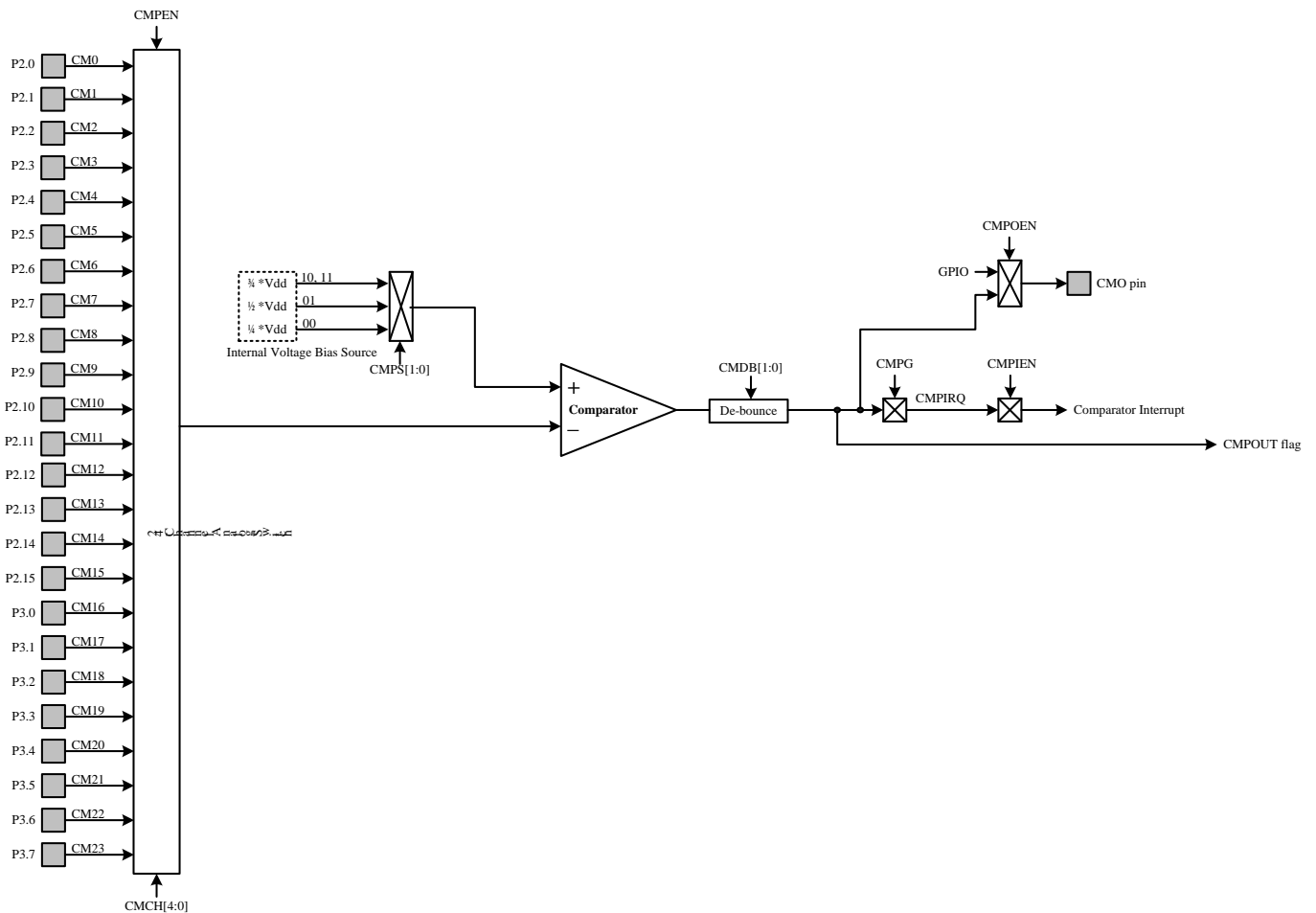
Step3: DAC Analog Enable (Sigma-delta DAC Power-Up Sequence)

# 14 24-CHANNEL COMPARATOR

## 14.1 OVERVIEW

The analog comparator compares negative input voltage, and then output the result to comparator output terminal. The comparator has multi-input selection for different applications. The comparator negative input terminal is up to 24-channel controlled by CMCH[4:0]. The comparator positive input terminal has three selections controlled by CMPS[1:0] bits. The comparator output terminal connects to external pin CMO and connects to internal path. There is a programmable direction function to decide comparator trigger edge for indicator function. The comparator has flag indicator, interrupt function and sleep mode weak-up function for different application.

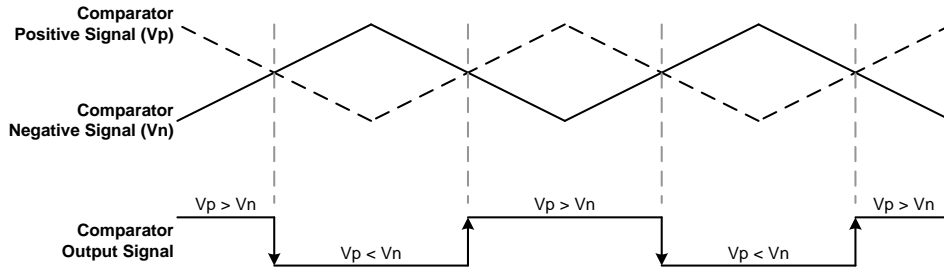
- 24-channel negative input selection.
- Comparator output function.
- Programmable internal reference voltage connected to comparator's positive terminal.
- Comparator unit with programmable output de-bounce
- Programmable trigger direction.
- Interrupt function.
- Sleep mode wake-up function.





## 14.2 COMPARATOR OPERATION

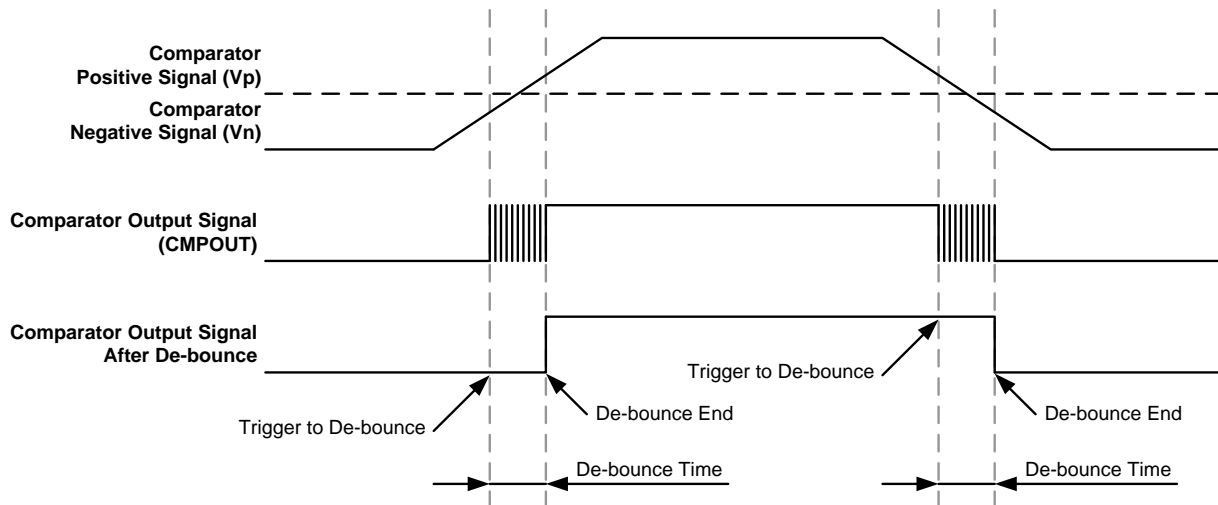
The comparator operation is to compare the voltage between comparator positive input and negative input terminals. When the positive input voltage is greater than the negative input voltage, the comparator output is high status. When the positive input voltage is smaller than the negative input voltage, the comparator output is low status.



The comparator builds in interrupt function. The interrupt function trigger edge is selected by CMPG. The trigger edge supports rising edge (CMPG=0), falling edge (CMPG=1). If the trigger edge condition is found, the CMPIRQ is set as "1". If the comparator interrupt function enables, the system will execute interrupt routine. The CMPIRQ must be cleared by program.

The comparator builds in sleep mode wake-up function. The comparator sleep mode wake-up trigger edge is bi-direction. The comparator's wake-up function only supports sleep mode, not deep-sleep mode and deep-power down mode. If the trigger edge condition (comparator output status exchanging) is found, the system will be wake-up from sleep mode. If the trigger edge direction is interrupt trigger condition, the CMPIRQ is set as "1". Of course the interrupt routine is executed if the interrupt function enabled. When the wake-up trigger edge direction is equal to interrupt trigger condition, the system will execute interrupt operation after sleep mode wake-up immediately.

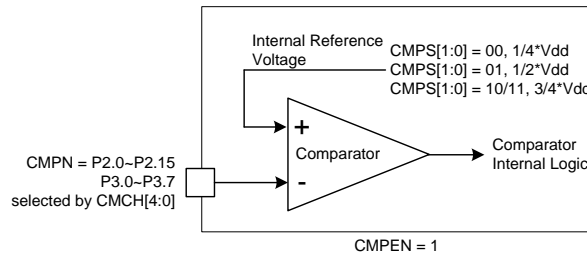
The critical condition is comparator positive voltage equal to comparator negative voltage, and the voltage range is decided comparator offset parameter of input common mode. In the voltage range, the comparator output signal is unstable and keeps oscillating until the differential voltage exits the range. In the condition, the comparator flag (CMPIRQ) latches the first exchanging and issue the status, but the status is a transient, not a stable condition. So the comparator builds in a filter to de-bounce the transient condition. The comparator output signal is through a de-bounce circuit to filter comparator transient status. The de-bounce time is controlled by CMDB[1:0] bits that means the comparator minimum response time is zero,  $1 \times \text{CMP\_PCLK}$ ,  $2 \times \text{CMP\_PCLK}$  or  $3 \times \text{CMP\_PCLK}$ . The de-bounce time depends on the signal slew rate and selected by program.



The comparator positive input terminal includes internal reference voltage source. The internal reference voltage source supports three levels which are  $1/4 \cdot V_{dd}$  (CMPS[1:0] = 00),  $1/2 \cdot V_{dd}$  (CMPS[1:0] = 01) and  $3/4 \cdot V_{dd}$  (CMPS[1:0] = 10,11).

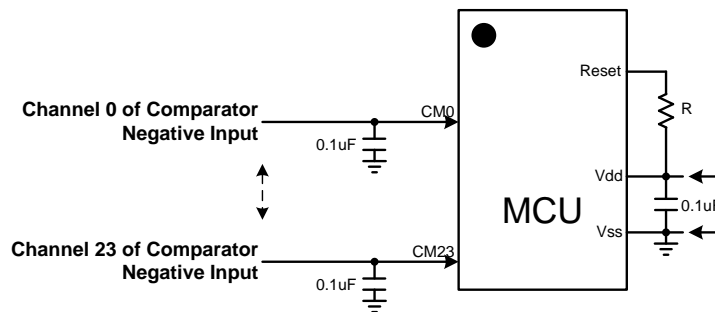
The comparator negative input terminal supports maximum 24-channel controlled by CMCH [4:0]. 00000 = P2.0. 00001 = P2.1. 00010 = P2.2. 00011 = P2.3. 00100 = P2.4. 00101 = P2.5. 00110 = P2.6. 00111 = P2.7. 01000 = P2.8. 01001 = P2.9. 01010 = P2.10. 01011 = P2.11. 01100 = P2.12. 01101 = P2.13. 01110 = P2.14. 01111 = P2.15. 10000 = P3.0. 10001 = P3.1. 10010 = P3.2. 10011 = P3.3. 10100 = P3.4. 10101 = P3.5. 10110 = P3.6. 10111 = P3.7. These channels selected is only when the comparator enables (CMPEN=1), the 24-channel analog switch works, or not workable. If one pin is selected to be comparator negative input pin, the pin is switched to input mode and connected to comparator negative input terminal. When the system selects to other pin or comparator disables, the original channel will returns to last GPIO mode automatically.

The comparator output status can output to CMO pin controlled by CMPOEN bit. When CMPOEN=0, the comparator output pin is GPIO mode. If CMPOEN=1, CMO pin outputs comparator output status and isolates GPIO mode. The comparator output terminal connects to internal path. The CMPOUT flag is the CMPOUT shows the comparator result immediately, but the CMPIRQ only indicates the event of the comparator result. The comparator output terminal through de-bounce circuit generates the comparator trigger edge controlled by CMPG. The even condition is controlled by register and includes rising edge (CMPOUT changes from low to high), falling edge (CMPOUT changes from high to low) controlled by CMPG bit. The CMPIRQ = 1 condition makes the comparator interrupt service executed when CMPGIE (comparator interrupt control bit) set.



### 14.3 COMPARATOR APPLICATION NOTICE

The comparator is to compares the positive voltage and negative voltage to output result. The positive used internal reference and negative sources are analog signal. In hardware application circuit, the comparator input pins must be connected a 0.1uF capacitance to reduce power noise and make the input signal more stable. The application circuit is as following.



### 14.4 COMPARATOR CONTROL REGISTERS

Base Address: 0x4006 6000

#### 14.4.1 Comparator Control register (CMPM)

Address Offset: 0x00

| Bit | Name  | Description             | Attribute | Reset |
|-----|-------|-------------------------|-----------|-------|
| 31  | CMPEN | Comparator control bit. | R/W       | 0     |

|              |           |   |     |   |
|--------------|-----------|---|-----|---|
|              |           | 0 : Disable. P2[15:0], P3[7:0] are GPIO mode.<br>1 : Enable. Comparator negative input pins are controlled by CMCH[4:0] bits.   |     |   |
| <b>30:15</b> | Reserved  |   | R   | 0 |
| <b>14</b>    | CMPOUT    | Comparator output flag bit. <b>The comparator output status is “1” as comparator disabled.</b><br>0 : Comparator internal reference voltage is less than CMPN voltage.<br>1 : Comparator internal reference voltage is larger than CMPN voltage.  | R   | 1 |
| <b>13:12</b> | Reserved  |   | R   | 0 |
| <b>11:10</b> | CMDB[1:0] | Comparator output debounce time select bit.<br>00 : 1* CMP_PCLK.<br>01 : 2* CMP_PCLK.<br>10 : 3* CMP_PCLK.<br>11 : No de-bounce.  | R/W | 0 |
| <b>9</b>     | CMPOEN    | Comparator output pin control bit.<br>0 : Disable. CMO pin is GPIO mode.<br>1 : Enable comparator output pin. P3.8 pin exchanges to comparator output pin (CMO pin), and GPIO function is isolated.   | R/W | 0 |
| <b>8</b>     | CMPG      | Comparator interrupt trigger direction control bit.<br>0 : Rising edge trigger. CMPP > CMPN or comparator internal reference voltage.<br>1 : Falling edge trigger. CMPP < CMPN or comparator internal reference voltage.  | R/W | 0 |
| <b>7</b>     | Reserved  |   | R   | 0 |
| <b>6:5</b>   | CMPS[1:0] | Comparator positive input voltage control bit. <b>If TCHEN=1, CMPS[1:0] bits are useless, and the internal reference switches automatically.</b><br>00 : Internal 1/4*Vdd and enable internal reference voltage generator.<br>01 : Internal 1/2*Vdd and enable internal reference voltage generator.<br>10,11 : Internal 3/4*Vdd and enable internal reference voltage generator.   | R/W | 0 |
| <b>4:0</b>   | CMCH[4:0] | Comparator negative input pin control bit. <b>CMPEN must be “1”.</b><br>00000 : Comparator negative input pin is P2.0.<br>00001 : Comparator negative input pin is P2.1.<br>00010 : Comparator negative input pin is P2.2.<br>00011 : Comparator negative input pin is P2.3.<br>00100 : Comparator negative input pin is P2.4.<br>00101 : Comparator negative input pin is P2.5.<br>00110 : Comparator negative input pin is P2.6.<br>00111 : Comparator negative input pin is P2.7.<br>01000 : Comparator negative input pin is P2.8.<br>01001 : Comparator negative input pin is P2.9.<br>01010 : Comparator negative input pin is P2.10.<br>01011 : Comparator negative input pin is P2.11.<br>01100 : Comparator negative input pin is P2.12.<br>01101 : Comparator negative input pin is P2.13.<br>01110 : Comparator negative input pin is P2.14.<br>01111 : Comparator negative input pin is P2.15.<br>10000 : Comparator negative input pin is P3.0.<br>10001 : Comparator negative input pin is P3.1.<br>10010 : Comparator negative input pin is P3.2.<br>10011 : Comparator negative input pin is P3.3.<br>10100 : Comparator negative input pin is P3.4.<br>10101 : Comparator negative input pin is P3.5.<br>10110 : Comparator negative input pin is P3.6.<br>10111 : Comparator negative input pin is P3.7.<br>11000~11111 : Reserved. | R/W | 0 |

### 14.4.2 Comparator Interrupt Enable register (CMP\_IE)

Address Offset: 0x10

This register controls whether the interrupt condition in the Comparator controller is enabled.

| Bit         | Name     | Description   | Attribute | Reset |
|-------------|----------|---|-----------|-------|
| <b>31:1</b> | Reserved |   | R         | 0     |
| <b>0</b>    | CMPGIE   | Comparator edge trigger interrupt enable. (Comparator interrupt trigger direction refer to CMPG)<br>0: Disable<br>1: Enable | R/W       | 0     |

### 14.4.3 Comparator Interrupt Status register (CMP\_RIS)

Address Offset: 0x14

This register contains the status for interrupt condition, regardless of whether or not the interrupt is enabled in CMP\_IE register.

This register indicates the status for Comparator control raw interrupts. A Comparator interrupt is sent to the interrupt controller if the corresponding bit in the CMP\_IE register is set.

| Bit  | Name     | Description   | Attribute | Reset |
|------|----------|---|-----------|-------|
| 31:1 | Reserved |   | R         | 0     |
| 0    | CMPGIF   | Comparator edge trigger interrupt flag<br>0: Comparator edge trigger doesn't occur.<br>1: Comparator edge trigger occurs. | R         | 0     |

### 14.4.4 Comparator Interrupt Clear register (CMP\_IC)

Address Offset: 0x18

| Bit  | Name     | Description                          | Attribute | Reset |
|------|----------|--------------------------------------|-----------|-------|
| 31:1 | Reserved |                                      | R         | 0     |
| 0    | CMPGIC   | 0: No effect<br>1: Clear CMPGIF bit. | W         | 0     |

# 15 FLASH

## 15.1 OVERVIEW

The SN32F100 series 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. The SN32F100 series 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.

## 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 | Up to 16K × 32 bits divided into 64 pages of 1024 Bytes |
| Boot ROM | Up to 1K × 32 bits divided into 4 pages of 1024 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 ~ 0x000003FF | 1024        |
|             | Page 1  | 0x00000400 ~ 0x000007FF | 1024        |
|             | .       | .                       | .           |
|             | .       | .                       | .           |
| Boot Loader | Page 63 | 0x0000FC00 ~ 0x0000FFFF | 1024        |
|             | Page 0  | 0x1FFF0000 ~ 0x1FFF03FF | 1024        |
|             | Page 1  | 0x1FFF0400 ~ 0x1FFF07FF | 1024        |
|             | Page 2  | 0x1FFF0800 ~ 0x1FFF0BFF | 1024        |
|             | Page 3  | 0x1FFF0C00 ~ 0x1FFF0FFF | 1024        |

## 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 UART0 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)

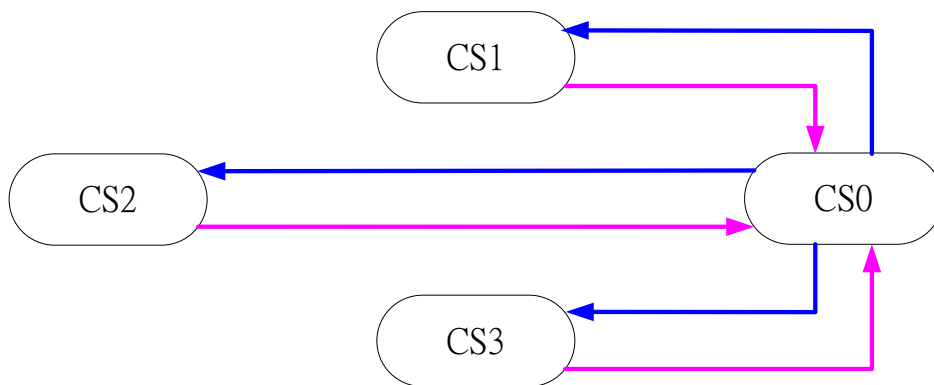
Code Security is a mechanism that allows the user to enable different levels of security in the system so that access to the on-chip Flash and use of the ISP can be restricted.

Important: Any Code Security change becomes effective only after the device has gone through a power cycle.

| CS Level | Pattern | Read | Erase | Program | SWD | Description  |
|----------|---------|------|-------|---------|-----|--|
| CS0      | 0xFFFF  | O    | O     | O       | O   | Writer can Read/Erase/Program User ROM.<br>SWD can Read/Erase/Program User ROM.<br>FW can Read/Erase/Program User ROM (EEPROM emulation)             |
| CS1      | 0x5A5A  | X    | O     | O       | X   | Writer can Erase/Program User ROM,<br>SWD can NOT Read/Erase/Program User ROM.<br>FW can Read/Erase/Program User ROM (EEPROM emulation)              |
| CS2      | 0xA5A5  | X    | O/X   | O/X     | X   | Writer can NOT Read/Erase/Program User ROM.<br>SWD can NOT Read/Erase/Program User ROM.<br>FW can Read/Erase/Program User ROM (EEPROM emulation)     |
| CS3      | 0x55AA  | X    | X     | X       | X   | Writer can NOT Read/Erase/Program User ROM.<br>SWD can NOT Read/Erase/Program User ROM.<br>FW can NOT Read/Erase/Program User ROM (EEPROM emulation) |

\* **Note:** User may try to change security level from CS3 to CS0, 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 STRT 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<br>0: Flash operation (programming/erase) is not completed.<br>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<br>0: Read→No error.<br>Write→Clear this flag.<br>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<br>0: Flash operation is not busy.<br>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

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

| Bit  | Name      | Description  | Attribute | Reset |
|------|-----------|--|-----------|-------|
| 31:0 | FAR[31:0] | Flash Address<br>Choose the Flash address to erase when Page Erase is selected, or to program when Page Program is selected.<br><i>Note: Write access to this register is blocked when the BUSY bit in the FLASH_STATUS register is set.</i> | R/W       | 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 P0.12 and P0.13 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.

### 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 SN32F100 series MCU firmware development.

SN32F100 Embedded ICE Emulator System includes:

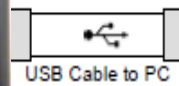
- SN32F100 Starter-Kit.
- SN-LINK
- USB cable to provide communications between the SN-Link and PC.
- IDE Tools (KEIL RVMDK)



SN32F100 Starter-Kit.



SN-LINK



IDE Tools

## 17.1 SN-LINK

SN-LINK is a high speed emulator for SONiX 32-bit series MCU. It debugs and programs based on SWD protocol. In addition to debugger functions, the SN-LINK also may be used as a programmer to load firmware from PC to MCU for engineering production, even mass production.

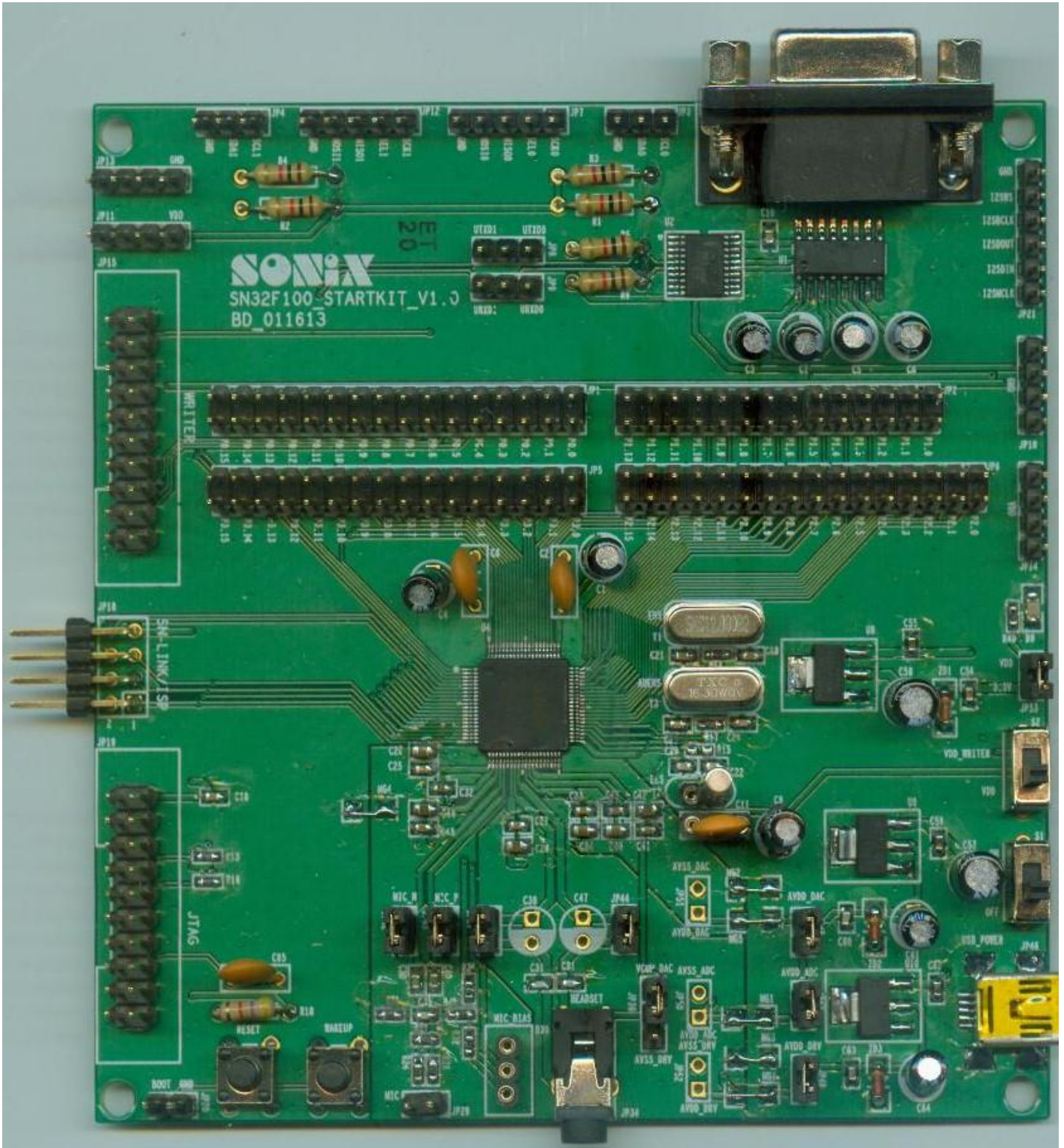




## 17.2 SN32F100 STARTER-KIT

SN32F100 Starter-kit is an easy-development platform. It includes SN32F109 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 SN32F100 series MCU integrates SWD debugger circuitry.

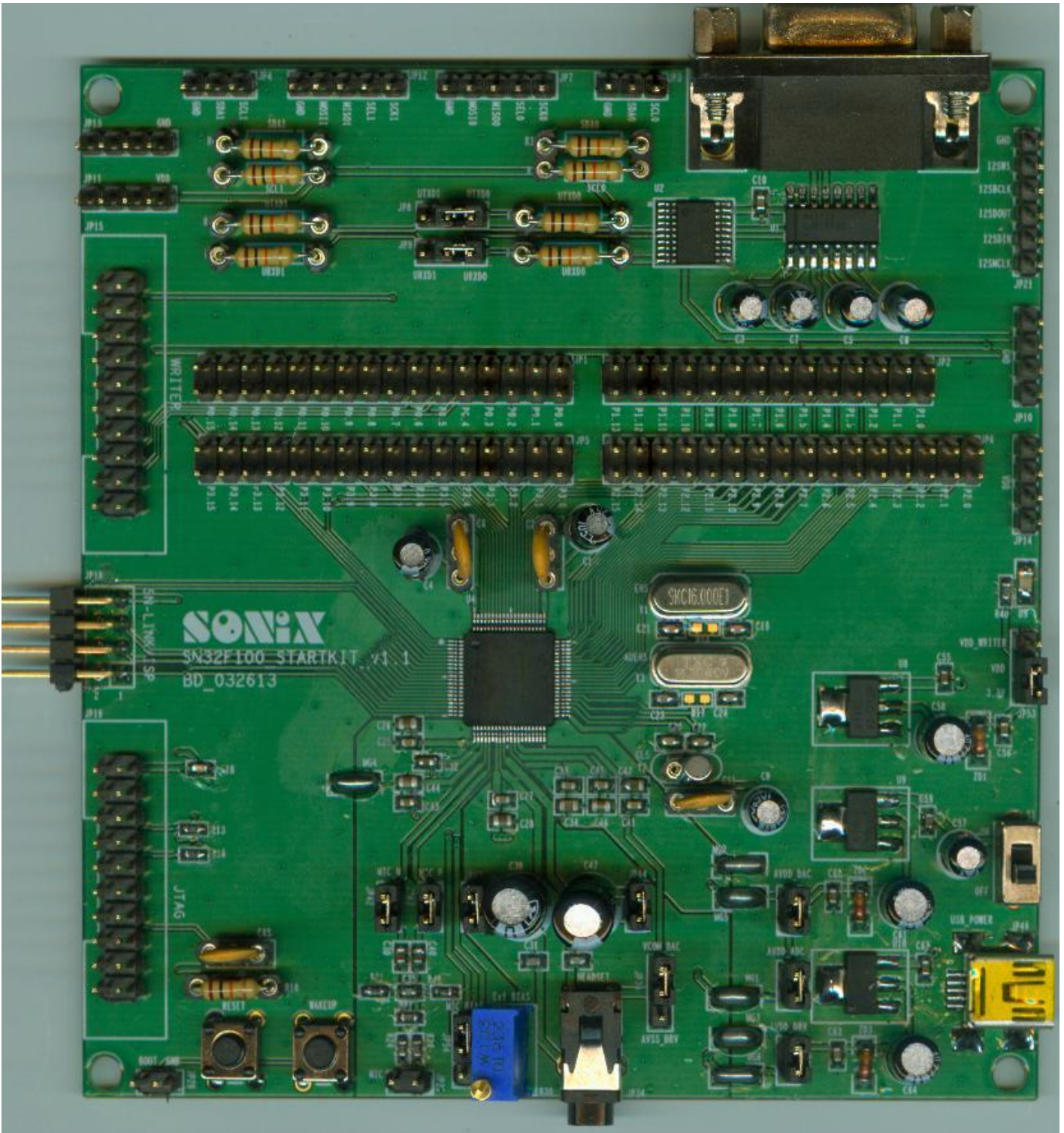
### 17.2.1 SN32F100 Start Kit V1.0



- JP46: Mini USB connector for power supply.
- S1: USB power on/off.
- S2: MCU power source is VDD or Writer.
- JP53: VDD power source is 3.3V from board or external power. Do not short if External power source is used.
- U4: SN32F109F real chip.
- D9: Power LED.
- RESET button: External reset trigger source.
- WAKEUP button: Trigger source to wake up from deep power-down mode.
- Y1: External high-speed X'tal
- Y2: External low-speed 32.768KHz X'tal
- Y3: External high-speed X'tal for Audio.
- JP18: SN-LINK connector
- JP15: Writer connector
- JP20: Short to force MCU stay in Boot loader.
- JP21: I2S connector.
- JP3/JP4: I2C0/I2C1 connector.
- JP7/JP12: SPI0/SPI1 connector.
- JP8/JP9: UART0/UART1 connector.
- R1: SCL0 pull-up resistor.
- R3: SDA0 pull-up resistor.
- R2: SCL1 pull-up resistor.
- R4: SDA1 pull-up resistor.
- R5: UTXD0 or UTXD1 pull-up resistor.
- R6: URXD0 or URXD1 pull-up resistor.
- JP47: Codec ADC power connector.
- JP48: Codec DAC power connector.
- JP49: Codec Driver power connector.
- JP34: Headset connector.
- JP29: Microphone connector.
- JP42: MIC\_N connector.
- JP43: MIC\_P connector.
- R30: MIC\_BIAS from external bias adjust.



17.2.2 SN32F100 Start Kit V1.1/V1.2



- JP46: Mini USB connector for power supply.
- S1: USB power on/off.
- JP53: VDD power source is 3.3V from board, Writer, or external power. Open if External power source is used.
- U4: SN32F109F real chip.
- D9: Power LED.
- RESET button: External reset trigger source.
- WAKEUP button: Trigger source to wake up from deep power-down mode.
- Y1: External high-speed X'tal

- Y2: External low-speed 32.768KHz X'tal
- Y3: External high-speed X'tal for Audio.
- JP18: SN-LINK connector
- JP15: Writer connector
- JP20: Short to force MCU stay in Boot loader.
- JP21: I2S connector.
- JP3/JP4: I2C0/I2C1 connector.
- JP7/JP12: SPI0/SPI1 connector.
- JP8/JP9: UART0/UART1 connector.
- R1: SCL0 pull-up resistor.
- R3: SDA0 pull-up resistor.
- R2: SCL1 pull-up resistor.
- R4: SDA1 pull-up resistor.
- R5: UTXD0 pull-up resistor.
- R6: URXD0 pull-up resistor.
- R41: UTXD1 pull-up resistor.
- R42: URXD1 pull-up resistor.
- JP47: Codec ADC power connector.
- JP48: Codec DAC power connector.
- JP49: Codec Driver power connector.
- JP34: Headset connector.
- JP29: Microphone connector.
- JP42: MIC\_N connector.
- JP43: MIC\_P connector.
- JP54: MIC\_BIAS from chip supply or external bias.
- R30: MIC\_BIAS from external bias adjust.



# 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)      |                         |
| SN32F107, SN32F108, SN32F109 .....        | -40°C ~ + 85°C          |
| Storage ambient temperature (Tstor) ..... | -40°C ~ + 125°C         |

## 18.2 ELECTRICAL CHARACTERISTIC

| Standard Operating Conditions (Typical temperature Ta = 25°C) |                      |   |   |                      |                    |      |    |
|---|----------------------|---|---|----------------------|--------------------|------|----|
| Operating Temperature   |                      | -40°C ≤ Ta ≤ +85°C for Industrial Class                           |   |                      |                    |      |    |
| The below data covers process corner range (SS-TT~FF).        |                      |   |   |                      |                    |      |    |
| 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 <sub>POR</sub>     | VDD rise rate to ensure internal power-on reset                   | 0.05                                      | -                    | -                  | V/ms |    |
| Power Consumption   |                      |   |   |                      |                    |      |    |
| Supply Current  | I <sub>dd1</sub>     | Normal mode   | System clock = 12MHz                      | -                    | 7                  | -    | mA |
|   |                      |   | [1][2][3]                                 | System clock = 50MHz | -                  | 30   | -  |
|   | I <sub>dd2</sub>     | Sleep Mode  | System clock = 12MHz                      | -                    | 2.6                | -    | mA |
|   |                      |   | [1][2][3][6]                              | System clock = 16KHz | -                  | 2.2  | -  |
|   | I <sub>dd3</sub>     | Deep-sleep Mode   | V <sub>dd</sub> =3.3V                     | -                    | 7                  | -    | uA |
|   | I <sub>dd4</sub>     | Deep power-down Mode  | V <sub>dd</sub> =3.3V                     | -                    | 220                | -    | nA |
|   |                      | [7]   |   |                      |                    |      |    |
| Port Pins, RESET pin  |                      |   |   |                      |                    |      |    |
| High-level input voltage                                      | V <sub>IH</sub>      |   | 0.8V <sub>dd</sub>                        | -                    | V <sub>dd</sub>    | V    |    |
| Low-level input voltage                                       | V <sub>IL</sub>      |   | V <sub>ss</sub>                           | -                    | 0.2V <sub>dd</sub> | V    |    |
| Input voltage   | V <sub>i</sub>       |   | 0   | -                    | V <sub>dd</sub>    | V    |    |
| Output voltage  | V <sub>o</sub>       |   | 0   | -                    | V <sub>dd</sub>    | V    |    |
| I/O port pull-up resistor                                     | R <sub>PU</sub>      | Vin = V <sub>ss</sub> , V <sub>dd</sub> = 3.3V                    | 40  | 60                   | 80                 | KΩ   |    |
| I/O port pull-down resistor                                   | R <sub>PD</sub>      | Vin = 3.3V  | 40  | 60                   | 80                 | KΩ   |    |
| I/O port input leakage current                                | I <sub>leakg</sub>   | Pull-up resistor disable, Vin = V <sub>dd</sub>                   | -   | -                    | 2                  | uA   |    |
|   |                      | I2C-bus pins (P0.2, P0.3, P3.14 and P3.15), Vin = V <sub>dd</sub> | -   | 2                    | 4                  | uA   |    |
| I/O High-level output source current                          | I <sub>OH</sub>      | Standard port and RESET pins                                      | V <sub>OP</sub> = V <sub>dd</sub> – 0.5V; | 5                    | 10                 | -    | mA |
|   |                      | High-drive output pin (P0.0~P0.3, P3.12~P3.15)                    | V <sub>OP</sub> = V <sub>dd</sub> – 0.5V  | 12                   | 20                 | -    | mA |
| I/O Low-level output sink current                             | I <sub>OL</sub>      | Standard port and RESET pins                                      | V <sub>OP</sub> = V <sub>ss</sub> + 0.5V  | 5                    | 10                 | -    | mA |
| Codec   |                      |   |   |                      |                    |      |    |
| ADC Analog Power  | AVDD <sub>_ADC</sub> |   | 2.7                                       | 3.3                  | 3.6                | V    |    |
| Microphone Bias   | MIC <sub>_BIAS</sub> | Microphone Bias Voltage<br>SEL_MICB=0                             | -5%                                       | 2.64                 | +5%                | V    |    |

|                                |                   |   |         |       |       |       |   |
|--------------------------------|-------------------|---|---------|-------|-------|-------|---|
|                                |                   | Microphone Bias Voltage<br>SEL_MICB=1   | -5%     | 2.97  | +5%   | V     |   |
| ADC Analog Reference Levels    | VMID_A<br>DC      | Middle reference voltage<br>AVDD_ADC=3.3V, VDD=1.8V<br>AVSS_ADC=VSS=0V<br>CL=10uF   | -5%     | 1.65  | +5%   | V     |   |
| Analog Input to ADC Output     | SNR               | Signal-to-Noise Ratio<br>1KHz input, -120dB, A-weighted   |         | 94    |       | dB    |   |
|                                | THD+N             | Total Harmonic Distortion<br>1KHz input, -6dB   |         | -80   |       | dB    |   |
| DAC Analog Power               | AVDD_<br>DAC      |   | 2.7     | 3.3   | 3.6   | V     |   |
| Headphone Driver Power         | AVDD_<br>DRV      |   | 2.7     | 3.3   | 3.6   | V     |   |
| DAC Analog Reference Levels    | VMID_D<br>AC      | Middle reference voltage<br>AVDD_DAC=AVDD_DRV=3.3V, VDD=1.8V<br>AVSS_DAC=AVSS_DRV=VSS=0V<br>CL=4.7uF                            | -5%     | 1.65  | +5%   | V     |   |
|                                | VCOM_<br>DAC      | Common-mode voltage<br>AVDD_DAC=AVDD_DRV=3.3V, VDD=1.8V<br>AVSS_DAC=AVSS_DRV=VSS=0V<br>CL=4.7uF                                 | -5%     | 1.65  | +5%   | V     |   |
| Headphone Driver Analog Output | SNR               | Signal-to-Noise Ratio<br>AVDD_DAC=AVDD_DRV=3.3V, VDD=1.8V<br>AVSS_DAC=AVSS_DRV=VSS=0V<br>1KHz input, RL=16Ω, -120dB, A-weighted |         | 90    |       | dB    |   |
|                                | THD+N             | Total Harmonic Distortion<br>AVDD_DAC=AVDD_DRV=3.3V, VDD=1.8V<br>AVSS_DAC=AVSS_DRV=VSS=0V<br>1KHz input, RL=16Ω, Po=20mW, -6dB  |         | -75   |       | dB    |   |
| <b>FLASH</b>                   |                   |   |         |       |       |       |   |
| Supply Voltage                 | Vdd1              |   | 1.8     |       | Vdd   | V     |   |
| Endurance time                 | T <sub>EN</sub>   | Erase + Program   | 20K     | *100K | -     | Cycle |   |
| Page Erase current             | I <sub>PER</sub>  | Vdd1=2.5V   | -       | 2.5   | 5     | mA    |   |
| Program current                | I <sub>PG</sub>   | Vdd1=2.5V   | -       | 3.5   | 7     | mA    |   |
| Page erase time                | T <sub>PE</sub>   | Vdd = 2.5V, 1-Page (1024 bytes).  | -       | 25    | 30    | ms    |   |
| 1-Word Programming time        | T <sub>PG</sub>   | Vdd = 2.5V, 1-Word (32 bits).   | -       | 60    | 70    | us    |   |
| <b>MISC</b>                    |                   |   |         |       |       |       |   |
| Low Voltage Detector           | LVD               | Interrupt   | 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 |
|                                | 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 <sub>IHRC</sub> | T=25°C, Vdd=1.8V~3.6V   | 11.76   | 12    | 12.24 | MHz   |   |
|                                |                   | T=-40°C~85°C, Vdd=1.8V~3.6V   | 11.4    | 12    | 12.6  | MHz   |   |

\* These parameters are for design reference, not tested.

[1] I<sub>DD</sub> measurements were performed with all pins configured as GPIO outputs driven LOW and pull-up resistors disabled and VDD=3.3V

[2] LVD and all peripherals are disabled.

[3] IHRC and ILRC are enabled, external X'tal are disabled, and PLL is disabled.

[4] IHRC is disabled, external high X'tal is enabled, and PLL is enabled.

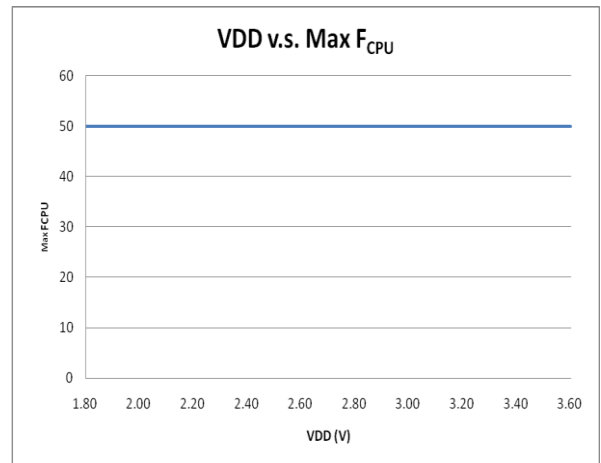
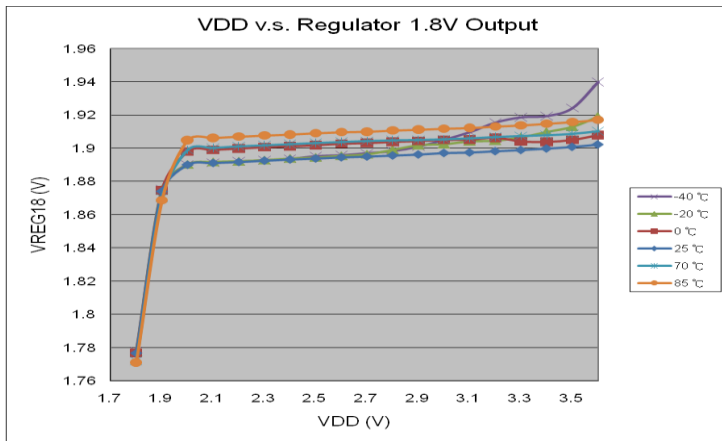
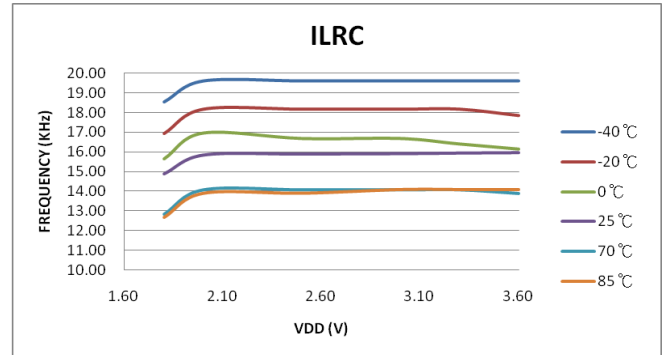
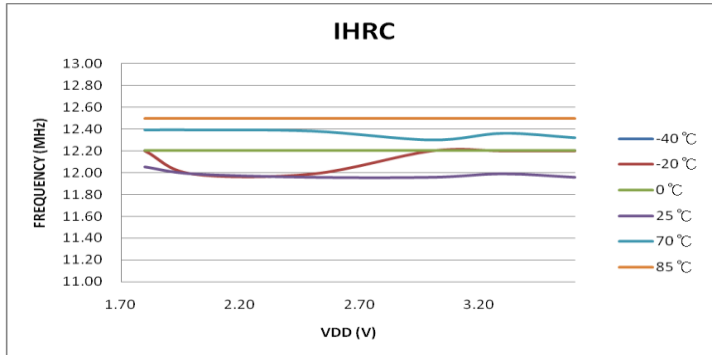
[5] ILRC is enabled, IHRC and external X'tal are disabled, and PLL is disabled.

[6] All oscillators and analog blocks are turned off.

[7] DPDWAKEUP pin is pulled HIGH internally.

## 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.

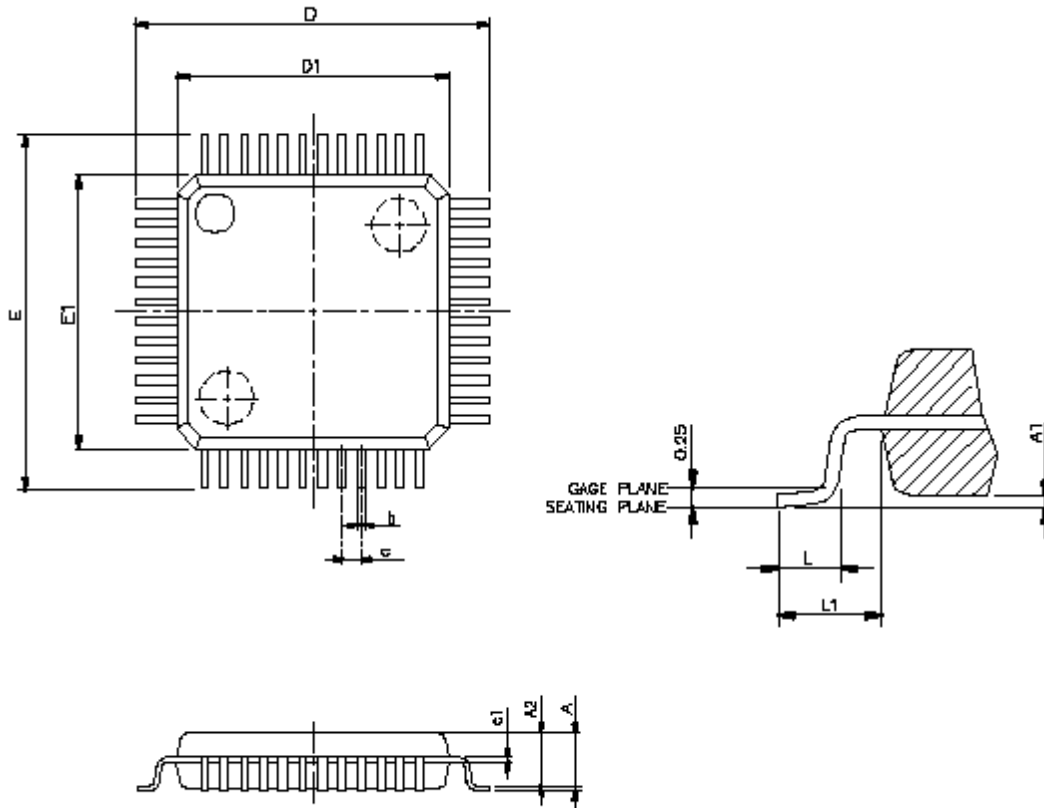


# 19 FLASH ROM PROGRAMMING PIN

| Programming Information of SN32F100 Series |                    |                               |           |                |      |          |      |        |     |        |     |
|--|--------------------|-------------------------------|-----------|----------------|------|----------|------|--------|-----|--------|-----|
| Chip Name                                  |                    | SN32F109F                     | SN32F108F | SN32F107F      |      |          |      |        |     |        |     |
| MP PRO Writer Connector                    |                    | Flash IC / JP3 Pin Assignment |           |                |      |          |      |        |     |        |     |
| Number                                     | Name               | Number                        | Pin       | Number         | Pin  | Number   | Pin  | Number | Pin | Number | Pin |
| 1  | VDD                | 41<br>61<br>80                | VDD       | 32<br>49<br>64 | VDD  | 27<br>48 | VDD  |        |     |        |     |
| 2  | GND                | 42<br>62<br>79                | VSS       | 33<br>50<br>63 | VSS  | 28<br>47 | VSS  |        |     |        |     |
| 3  | PGDCLK (CLK)       | 9                             | P0.4      | 9              | P0.4 | 5        | P0.4 |        |     |        |     |
| 4  | CE                 |                               |           |                |      |          |      |        |     |        |     |
| 5  | OTPCLK (PGM)       | 11                            | P0.6      | 11             | P0.6 | 7        | P0.6 |        |     |        |     |
| 6  | PGDIN (OE)         | 10                            | P0.5      | 10             | P0.5 | 6        | P0.5 |        |     |        |     |
| 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   | VR_DOUT (ALSB/PDB) | 12                            | P0.7      | 12             | P0.7 | 8        | P0.7 |        |     |        |     |

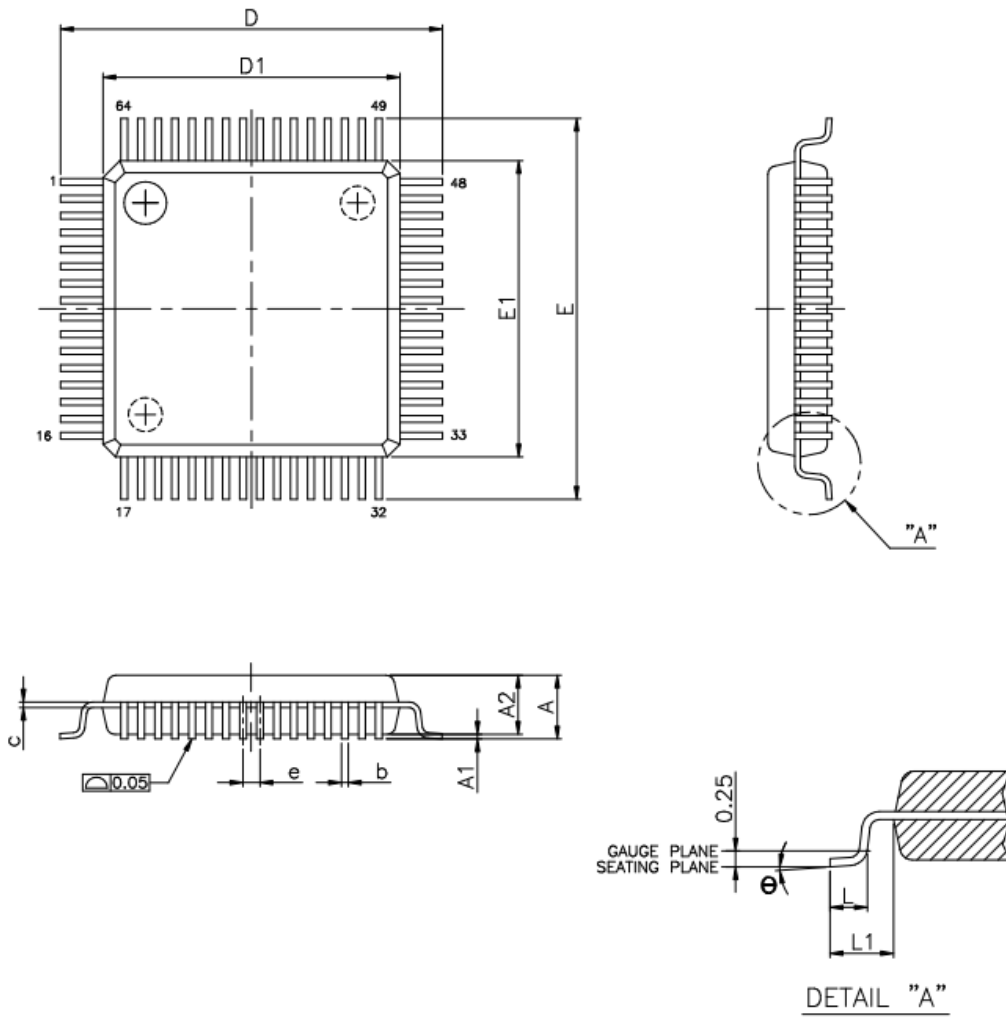
# 20 PACKAGE INFORMATION

## 20.1 LQFP 48 PIN



| SYMBOLS | MIN      | NOR | MAX  |
|---------|----------|-----|------|
|         | (mm)     |     |      |
| A       | -        | -   | 1.6  |
| A1      | 0.05     | -   | 0.15 |
| A2      | 1.35     | -   | 1.45 |
| c1      | 0.09     | -   | 0.16 |
| D       | 9.00 BSC |     |      |
| D1      | 7.00 BSC |     |      |
| E       | 9.00 BSC |     |      |
| E1      | 7.00 BSC |     |      |
| e       | 0.5 BSC  |     |      |
| B       | 0.17     | -   | 0.27 |
| L       | 0.45     | -   | 0.75 |
| L1      | 1 REF    |     |      |

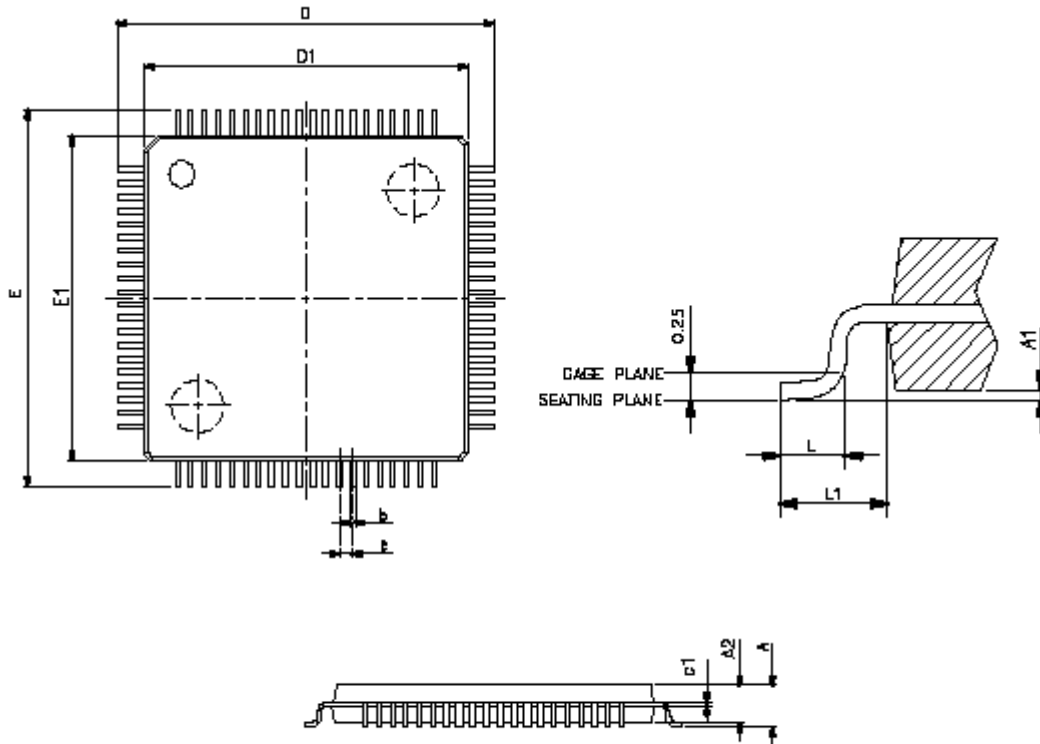
## 20.2 LQFP 64 PIN



VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

| SYMBOLS  | MIN.     | NOM. | MAX. |
|----------|----------|------|------|
| A        | —        | —    | 1.60 |
| A1       | 0.05     | —    | 0.15 |
| A2       | 1.35     | 1.40 | 1.45 |
| b        | 0.13     | 0.18 | 0.23 |
| c        | 0.09     | —    | 0.20 |
| D        | 9.00 BSC |      |      |
| D1       | 7.00 BSC |      |      |
| e        | 0.40 BSC |      |      |
| E        | 9.00 BSC |      |      |
| E1       | 7.00 BSC |      |      |
| L        | 0.45     | 0.60 | 0.75 |
| L1       | 1.00 REF |      |      |
| $\theta$ | 0°       | 3.5° | 7°   |

## 20.3 LQFP 80 PIN



VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

| SYMBOLS | MIN.    | MAX. |
|---------|---------|------|
| A       | --      | 1.6  |
| A1      | 0.05    | 0.15 |
| A2      | 1.35    | 1.45 |
| a1      | 0.09    | 0.16 |
| D       | 12 BSC  |      |
| D1      | 10 BSC  |      |
| E       | 12 BSC  |      |
| E1      | 10 BSC  |      |
| e       | 0.4 BSC |      |
| b       | 0.17    | 0.27 |
| L       | 0.45    | 0.75 |
| L1      | 1 REF   |      |

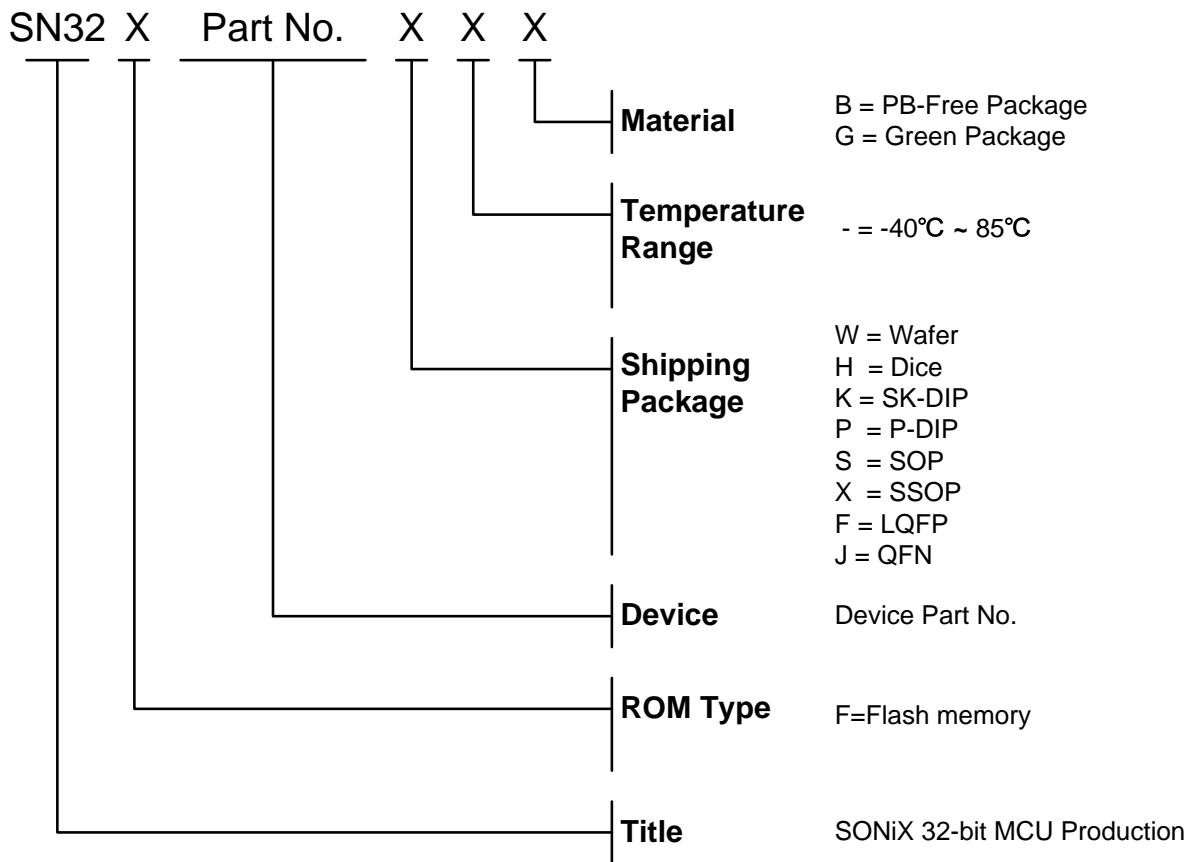
# 21 MARKING DEFINITION

## 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

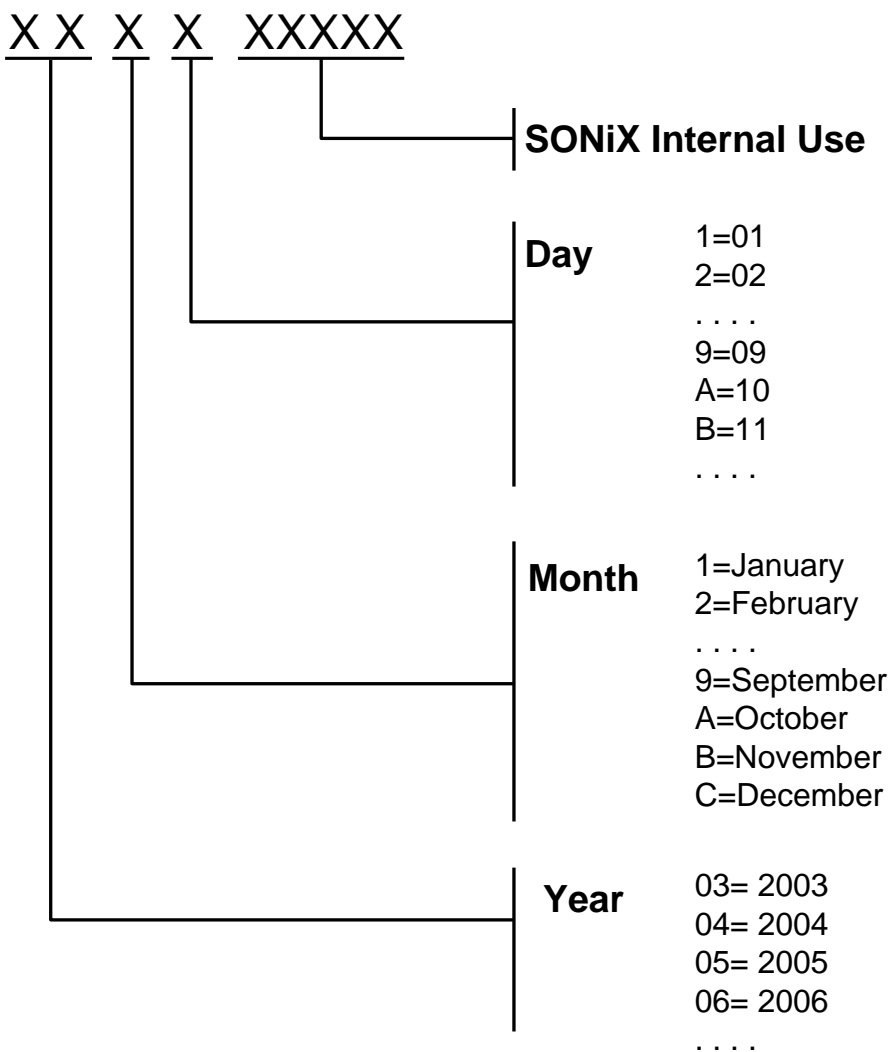




### 21.3 MARKING EXAMPLE

| Name       | ROM Type     | Device | Package | Temperature | Material      |
|------------|--------------|--------|---------|-------------|---------------|
| SN32F109FG | Flash memory | 109    | LQFP    | -40°C ~85°C | Green Package |
| SN32F109W  | Flash memory | 109    | Wafer   | -40°C ~85°C | -             |
| SN32F109H  | Flash memory | 109    | Dice    | -40°C ~85°C | -             |
| SN32F108FG | Flash memory | 109    | LQFP    | -40°C ~85°C | Green Package |
| SN32F107FG | Flash memory | 109    | LQFP    | -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