# NuMicro™ Family
# Nano100 Series
# Datasheet

*The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.*

*Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.*

*All data and specifications are subject to change without notice.*

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

## Table of Contents

*LIST OF FIGURES* ..... 7
*LIST OF TABLES* ..... 12
1 GENERAL DESCRIPTION ..... 13
2 FEATURES ..... 15
2.1 Nano100 Features – Base Line ..... 15
2.2 Nano110 Features – LCD Line ..... 21
2.3 Nano120 Features – USB Connectivity Line ..... 27
2.4 Nano130 Features – Advanced Line ..... 33
3 PARTS INFORMATION LIST AND PIN CONFIGURATION ..... 39
3.1 NuMicro™ Nano100 Series Selection Code ..... 39
3.2 NuMicro™ Nano100 Products Selection Guide ..... 40
3.2.1 NuMicro™ Nano100 Base Line Selection Guide ..... 40
3.2.2 NuMicro™ Nano110 LCD Line Selection Guide ..... 40
3.2.3 NuMicro™ Nano120 USB Connectivity Line Selection Guide ..... 40
3.2.4 NuMicro™ Nano130 Advanced Line Selection Guide ..... 41
3.3 Pin Configuration ..... 42
3.3.1 NuMicro™ Nano100 Pin Diagrams ..... 42
3.3.2 NuMicro™ Nano110 Pin Diagrams ..... 45
3.3.3 NuMicro™ Nano120 Pin Diagrams ..... 47
3.3.4 NuMicro™ Nano130 Pin Diagrams ..... 50
3.4 Pin Description ..... 52
3.4.1 NuMicro™ Nano100 Pin Description ..... 52
3.4.2 NuMicro™ Nano110 Pin Description ..... 63
3.4.3 NuMicro™ Nano120 Pin Description ..... 77
3.4.4 NuMicro™ Nano130 Pin Description ..... 88
4 BLOCK DIAGRAM ..... 102
4.1 Nano100 Block Diagram ..... 102
4.2 Nano110 Block Diagram ..... 103
4.3 Nano120 Block Diagram ..... 104
4.4 Nano130 Block Diagram ..... 105
5 FUNCTIONAL DESCRIPTION ..... 106
5.1 Memory Organization ..... 106
5.1.1 Overview ..... 106
5.1.2 Memory Map ..... 106
5.2 Nested Vectored Interrupt Controller (NVIC) ..... 107
5.2.1 Overview ..... 107
5.2.2 Features ..... 107
5.2.3 Exception Model and System Interrupt Map ..... 108
5.2.4 Vector Table ..... 110
5.2.5 Operation Description ..... 110
5.3 System Manager ..... 110

5.3.1 Overview ..... 110
5.3.2 Features ..... 110
5.3.3 Functional Description ..... 111
5.4 Clock Controller ..... 114
5.4.1 Overview ..... 114
5.4.2 Features ..... 114
5.4.3 Block Diagram ..... 115
5.4.4 Functional Description ..... 115
5.5 Analog to Digital Converter (ADC) ..... 118
5.5.1 Overview ..... 118
5.5.2 Features ..... 118
5.5.3 Block Diagram ..... 119
5.5.4 Functional Description ..... 120
5.6 Digital to Analog Converter (DAC) ..... 130
5.6.1 Overview ..... 130
5.6.2 Features ..... 130
5.6.3 Block Diagram ..... 131
5.6.4 Functional Description ..... 131
5.7 DMA Controller ..... 136
5.7.1 Overview ..... 136
5.7.2 Features ..... 136
5.7.3 Block Diagram ..... 138
5.7.4 Functional Description ..... 139
5.8 External Bus Interface ..... 142
5.8.1 Overview ..... 142
5.8.2 Features ..... 142
5.8.3 Block Diagram ..... 142
5.8.4 Functional Description ..... 143
5.9 FLASH Memory Controller (FMC) ..... 148
5.9.1 Overview ..... 148
5.9.2 Features ..... 148
5.9.3 Block Diagram ..... 148
5.9.4 Functional Description ..... 149
5.10 General Purpose I/O Controller ..... 160
5.10.1 Overview ..... 160
5.10.2 Features ..... 160
5.10.3 Block Diagram ..... 160
5.10.4 Functional Description ..... 160
5.11 $I^2C$ ..... 162
5.11.1 Overview ..... 162
5.11.2 Features ..... 163
5.11.3 Functional Description ..... 163
5.12 $I^2S$ ..... 174
5.12.1 Overview ..... 174
5.12.2 Features ..... 174
5.12.3 Block Diagram ..... 174

5.12.4 Functional Description ..... 175
5.13 LCD Display Driver ..... 177
5.13.1 Overview ..... 177
5.13.2 Features ..... 177
5.13.3 Block Diagram ..... 178
5.13.4 Functional Description ..... 179
5.13.5 Application Circuit ..... 182
5.14 Pulse Width Modulation (PWM) ..... 186
5.14.1 Overview ..... 186
5.14.2 Features ..... 187
5.14.3 Block Diagram ..... 188
5.14.4 Functional Description ..... 190
5.15 RTC ..... 197
5.15.1 Overview ..... 197
5.15.2 Features ..... 197
5.15.3 Block Diagram ..... 198
5.15.4 Functional Description ..... 199
5.16 Smart Card Host Interface (SC) ..... 202
5.16.1 Overview ..... 202
5.16.2 Features ..... 202
5.16.3 Block Diagram ..... 203
5.16.4 Functional Description ..... 205
5.17 SPI ..... 211
5.17.1 Overview ..... 211
5.17.2 Features ..... 211
5.17.3 Block Diagram ..... 212
5.17.4 Functional Description ..... 212
5.18 Timer Controller ..... 226
5.18.1 Overview ..... 226
5.18.2 Features ..... 226
5.18.3 Block Diagram ..... 226
5.18.4 Functional Description ..... 227
5.19 UART Controller ..... 232
5.19.1 Overview ..... 232
5.19.2 Features ..... 234
5.19.3 Block Diagram ..... 234
5.19.4 Functional Description ..... 237
5.20 USB ..... 245
5.20.1 Overview ..... 245
5.20.2 Features ..... 245
5.20.3 Block Diagram ..... 246
5.20.4 Functional Description ..... 246
5.21 Watchdog Timer Controller ..... 250
5.21.1 Overview ..... 250
5.21.2 Features ..... 250
5.21.3 Block Diagram ..... 250

5.21.4 Functional Description ..... 251
5.22 Window Watchdog Timer Controller ..... 253
5.22.1 Overview ..... 253
5.22.2 Features ..... 253
5.22.3 Block Diagram ..... 253
5.22.4 Functional Description ..... 253
6 ARM® CORTEX™-M0 CORE ..... 256
6.1 Overview ..... 256
6.2 Features ..... 256
6.3 System Timer (SysTick) ..... 257
7 APPLICATION CIRCUIT ..... 258
7.1 LCD Charge Pump ..... 258
7.1.1 C-type 1/3 Bias ..... 258
7.1.2 C-type 1/2 Bias ..... 258
7.1.3 Internal R-type ..... 258
7.1.4 External R-type ..... 259
7.2 ADC Application Circuit ..... 260
7.2.1 Voltage Reference Source ..... 260
7.3 DAC Application Circuit ..... 262
7.3.1 Voltage Reference Source ..... 262
7.4 Whole Chip Application Circuit ..... 264
8 POWER COMSUMPTION ..... 265
9 ELECTRICAL CHARACTERISTIC ..... 266
9.1 Absolute Maximum Ratings ..... 266
9.2 Nano100/Nano110/Nano120/Nano130 DC Electrical Characteristics ..... 267
9.3 AC Electrical Characteristics ..... 273
9.3.1 External Input Clock ..... 273
9.3.2 External 4~24 MHz XTAL Oscillator ..... 273
9.3.3 External 32.768 kHz Crystal ..... 274
9.3.4 Internal 12 MHz Oscillator ..... 274
9.3.5 Internal 10 kHz Oscillator ..... 274
9.4 Analog Characteristics ..... 275
9.4.1 12-bit ADC ..... 275
9.4.2 Brown-out Detector ..... 275
9.4.3 Power-on Reset ..... 276
9.4.4 Temperature Sensor ..... 276
9.4.5 12-bit DAC ..... 276
9.4.6 LCD ..... 277
9.4.7 Internal Voltage Reference ..... 277
9.4.8 USB PHY Specifications ..... 278
10 PACKAGE DIMENSIONS ..... 280
10.1 LQFP128 (14x14x1.4 mm footprint 2.0 mm) ..... 280
10.2 LQFP64 (10x10x1.4 mm footprint 2.0 mm) ..... 281
10.3 LQFP64 (7x7x1.4 mm footprint 2.0 mm) ..... 282

10.4 LQFP48 (7x7x1.4 mm footprint 2.0 mm).....284
10.5 QFN48 (7x7x0.85 mm).....285
11 REVISION HISTORY .....286

## LIST OF FIGURES

Figure 3-1 NuMicro™ Nano100 Series Selection Code ..... 39
Figure 3-2 NuMicro™ Nano100 LQFP 128-pin Diagram ..... 42
Figure 3-3 NuMicro™ Nano100 LQFP 64-pin Diagram ..... 43
Figure 3-4 NuMicro™ Nano100 LQFP 48-pin Diagram ..... 44
Figure 3-5 NuMicro™ Nano110 LQFP 128-pin Diagram ..... 45
Figure 3-6 NuMicro™ Nano110 LQFP 64-pin Diagram ..... 46
Figure 3-7 NuMicro™ Nano120 LQFP 128-pin Diagram ..... 47
Figure 3-8 NuMicro™ Nano120 LQFP 64-pin Diagram ..... 48
Figure 3-9 NuMicro™ Nano120 LQFP 48-pin Diagram ..... 49
Figure 3-10 NuMicro™ Nano130 LQFP 128-pin Diagram ..... 50
Figure 3-11 NuMicro™ Nano130 LQFP 64-pin Diagram ..... 51
Figure 4-1 NuMicro™ Nano100 Block Diagram ..... 102
Figure 4-2 NuMicro™ Nano110 Block Diagram ..... 103
Figure 4-3 NuMicro™ Nano120 Block Diagram ..... 104
Figure 4-4 NuMicro™ Nano130 Block Diagram ..... 105
Figure 5-1 Power Modes ..... 112
Figure 5-2 Clock Controller Block Diagram ..... 115
Figure 5-3 Clock Sources of Frequency Divider ..... 117
Figure 5-4 Frequency Divider Block Diagram ..... 117
Figure 5-5 ADC and DAC Block Diagram ..... 119
Figure 5-6 ADC Controller Block Diagram ..... 120
Figure 5-7 ADC Clock Control ..... 120
Figure 5-8 ADC Single Mode Conversion Timing Diagram ..... 121
Figure 5-9 ADC Single-cycle Scan on Enabled Channels Timing Diagram ..... 122
Figure 5-10 ADC Continuous Scan on Enabled Channels Timing Diagram ..... 123
Figure 5-11 ADC Conversion Result Monitor Logic Diagram ..... 124
Figure 5-12 ADC Controller Interrupt ..... 124
Figure 5-13 ADC Start Conversion Conditions ..... 125
Figure 5-14 Model of the sampling network ..... 126
Figure 5-15 Increased Sampling Time Waveform ..... 126
Figure 5-16 Additional Sample and Hold Clock Cycles (n) as a Function of the Signal Source Output Resistance Rin (kΩ) ..... 127

Figure 5-17 Additional Sample and Hold Clock Cycles (n) as a Function of the Signal Source Output Resistance Rin (kΩ)....127
Figure 5-18 DAC0 and DAC1 Block Diagram....131
Figure 5-19 DAC0 and DAC1 Ungroup Update Example....133
Figure 5-20 DAC0 and DAC1 Group Update Example....134
Figure 5-21 DAC Interrupt....135
Figure 5-22 DMA Controller Block Diagram....138
Figure 5-23 DMA Clock Control Diagram....138
Figure 5-24 CRC Generator Block Diagram....139
Figure 5-25 VDMA Block Transfer....140
Figure 5-26 EBI Block Diagram....143
Figure 5-27 Connection of 16-bit EBI Data Width 16-bit Device....143
Figure 5-28 Connection of 8-bit EBI Data Width with 8-bit Device....144
Figure 5-29 Timing Control Waveform for 16-bit Data Width....145
Figure 5-30 Timing Control Waveform for 8-bit Data Width....146
Figure 5-31 Timing Control Waveform for Insert Idle Cycle....147
Figure 5-32 Block Diagram of Flash Memory Controller....148
Figure 5-33 Flash Memory Organization....150
Figure 5-34 Flash Memory Mapping of CBS in CONFIG0....151
Figure 5-35 32/64/123KB Flash Memory Structure....152
Figure 5-36 CPU Halt during ISP Operation....156
Figure 5-37 ISP Operation Flow....157
Figure 5-38 GPIO Block Diagram....160
Figure 5-39 Push-Pull Output....161
Figure 5-40 Open-Drain Output....161
Figure 5-41 $I^2C$ Protocol....163
Figure 5-42 $I^2C$ Master Transmits Data to Slave....164
Figure 5-43 $I^2C$ Master Reads Data from Slave....164
Figure 5-44 $I^2C$ START and STOP Conditions....164
Figure 5-45 Bit Transfer on $I^2C$ Bus....165
Figure 5-46 Acknowledge on $I^2C$ Bus....165
Figure 5-47 $I^2C$ Time-out Block Diagram....166
Figure 5-48 $I^2C$ Normal Master Transmitter State....167
Figure 5-49 $I^2C$ Master Transmitter Mode....168
Figure 5-50 $I^2C$ Master Receiver Mode....169

Figure 5-51 I²C Slave Receiver Mode ....170
Figure 5-52 I²C Slave Transmitter Mode ....171
Figure 5-53 I²C General Call Mode ....172
Figure 5-54 I²C Bus Error Timing ....173
Figure 5-55 I²S Controller Block Diagram ....174
Figure 5-56 I²S bus timing diagram (Format = 0) ....175
Figure 5-57 MSB Justified Timing Diagram (Format = 1) ....175
Figure 5-58 FIFO Contents for Various I²S Modes ....176
Figure 5-59 LCD Driver Block Diagram ....178
Figure 5-60 LCD Memory Map ....179
Figure 5-61 COM Signal Waveform ....180
Figure 5-62 SEG Signal Waveform ....180
Figure 5-63 COM-SEG Signal Waveform by 1/6 Duty with 1/3 Bias ....181
Figure 5-64 1/3 Bias (External Resistor Ladder) ....182
Figure 5-65 1/2 Bias (External Resistor Ladder) ....183
Figure 5-66 1/3 Bias (Resistor Ladder with Capacitor) ....184
Figure 5-67 1/2 Bias (Resistor Ladder with Capacitor) ....184
Figure 5-68 1/3 Bias (Charge Pump) ....185
Figure 5-69 1/2 Bias (Charge Pump) ....185
Figure 5-70 PWM0 Clock ....188
Figure 5-71 PWM0 Generator for Channel 0, 1 ....188
Figure 5-72 PWM1 Clock ....188
Figure 5-73 PWM Generator for Channel 2, 3 ....189
Figure 5-74 Legend of Internal Comparator Output of PWM-Timer ....190
Figure 5-75 PWM-Timer Operation Timing for Channel 0 ....190
Figure 5-76 PWM Double Buffer Illustration ....191
Figure 5-77 PWM Controller Output Duty Ratio ....191
Figure 5-78 Paired PWM Output with Dead Zone Generation Operation ....192
Figure 5-79 PWM Capture Operation Timing ....193
Figure 5-80 PWM-Timer Interrupt ....194
Figure 5-81 PWM-Timer Stop Method 1 ....195
Figure 5-82 RTC Block Diagram ....198
Figure 5-83 SC Clock Control Diagram (4-bit Pre-Scale Counter in Clock Controller) ....203
Figure 5-84 SC Controller Block Diagram ....204
Figure 5-85 SC Data Character ....205

Figure 5-86 SC Activation Sequence .....206
Figure 5-87 SC Warm Reset Sequence .....207
Figure 5-88 SC Deactivation Sequence .....208
Figure 5-89 Initial Character TS .....208
Figure 5-90 SC Error Signal .....209
Figure 5-91 SPI Block Diagram .....212
Figure 5-92 SPI Master Mode Application Block Diagram .....213
Figure 5-93 SPI Slave Mode Application Block Diagram .....213
Figure 5-94 SPI Variable Clock Frequency .....215
Figure 5-95 SPI 32-bit in One Transaction .....215
Figure 5-96 SPI Byte Reorder .....216
Figure 5-97 SPI Byte Suspend Mode .....217
Figure 5-98 SPI 2-bit Transfer Mode .....218
Figure 5-99 SPI 2-bit Transfer Mode Timing Diagram .....218
Figure 5-100 SPI DUAL-I/O output Sequence .....219
Figure 5-101 SPI FIFO Mode Control Timing .....221
Figure 5-102 SPI Timing in Master Mode .....222
Figure 5-103 SPI Timing in Master Mode (Alternate Phase of SPICLK & LSB = 1) .....222
Figure 5-104 SPI Timing in Master Mode (Alternate Phase of SPICLK & LSB = 0) .....223
Figure 5-105 SPI Timing in Slave Mode .....223
Figure 5-106 SPI Timing in Slave Mode (Alternate Phase of SPICLK) .....224
Figure 5-107 Timer Controller Block Diagram .....227
Figure 5-108 Timer Clock Controller Diagram .....227
Figure 5-109 Timer Clock Controller Diagram .....229
Figure 5-110 Inter-Timer Trigger Mode .....231
Figure 5-111 UART Clock Control Diagram .....235
Figure 5-112 UART Block Diagram .....235
Figure 5-113 UART Auto-Flow Control Block Diagram .....237
Figure 5-114 UART Auto-Baud Rate Block Diagram .....237
Figure 5-115 UART CTSn Wake-Up Case 1 .....238
Figure 5-116 UART CTSn Wake-Up Case 2 .....238
Figure 5-117 UART DATA Wake-Up .....238
Figure 5-118 IrDA Block Diagram .....239
Figure 5-119 IrDA TX/RX Timing Diagram .....240

Figure 5-120 Structure of RS-485 Frame ..... 242
Figure 5-121 Structure of LIN Frame ..... 242
Figure 5-122 USB Block Diagram ..... 246
Figure 5-123 Endpoint allocation in SRAM ..... 247
Figure 5-124 USB Wake-up Interrupt Operation Flow ..... 248
Figure 5-125 USB Setup Transaction Followed by Data IN Transaction ..... 249
Figure 5-126 USB Data OUT Transaction ..... 249
Figure 5-127 Watchdog Controller Block Diagram ..... 250
Figure 5-128 Watchdog Timer Clock Control Diagram ..... 250
Figure 5-129 Watchdog Timing of Interrupt and Reset Signal ..... 252
Figure 5-130 Window Watchdog Controller Block Diagram ..... 253
Figure 5-131 Watchdog Controller Block Diagram ..... 253
Figure 6-1 M0 Functional Block ..... 256
Figure 9-1 Typical Crystal Application Circuit ..... 273

# LIST OF TABLES

Table 1-1 Connectivity Support Table ..... 14
Table 3-1 Nano100 Base Line Selection Table ..... 40
Table 3-2 Nano110 LCD Line Selection Table ..... 40
Table 3-3 Nano120 USB Connectivity Line Selection Table ..... 40
Table 3-4 Nano130 Advanced Line Selection Table ..... 41
Table 5-1 Exception Model ..... 108
Table 5-2 System Interrupt Map ..... 109
Table 5-3 Vector Table Format ..... 110
Table 5-4 Condition of Entering Power-down Mode Again Table ..... 111
Table 5-5 IP clock ON/OFF in Power Modes ..... 113
Table 5-6 Peripheral Engine Clocks ..... 116
Table 5-8 Memory Access Map ..... 149
Table 5-9 Flash Size ..... 149
Table 5-10 Boot Selection ..... 151
Table 5-11 Boot Selection and Supports Function ..... 151
Table 5-12 ISP Operation Command ..... 159
Table 5-13 UART Baud Rate Equation ..... 232
Table 5-14 UART Baud Rate Setting ..... 233
Table 5-16 Watchdog Time-out Interval Selection ..... 251
Table 5-17 Window Watchdog Prescaler Value Selection ..... 254

# 1 GENERAL DESCRIPTION

The Nano100 series ultra-low power 32-bit microcontroller is embedded with ARM® Cortex™-M0 core operated at a wide voltage range from 1.8V to 3.6V and runs up to 42 MHz frequency with 32K/64K/128K bytes embedded Flash and 8K/16K-byte embedded SRAM. Integrating LCD 4x40 or 6x38 (COM/Segment), USB 2.0 full-speed function, RTC, 12-bit SAR ADC, 12-bit DAC and provides high performance connectivity peripheral interfaces such as UART, SPI, $I^2C$, $I^2S$, GPIOs, EBI (External Bus Interface) for external memory-mapped device access and ISO-7816-3 for Smart card, the Nano100 series supports Brown-out Detector, Power-down mode with RAM retention and fast wake-up via many peripheral interfaces.

The Nano100 series provides low power voltage, low power consumption, low standby current, high integration peripherals, high-efficiency operation, fast wake-up function and the lowest cost 32-bit microcontrollers. The Nano100 series is suitable for a wide range of battery device applications such as:

- Portable Data Collector
- Portable Medical Monitor
- Portable RFID Reader
- Portable Barcode Scanner
- Security Alarm System
- System Supervisors
- Power Metering
- USB Accessories
- Smart Card Reader
- Wireless Game Control Device
- IPTV Remote Smart Keyboard
- Wireless Sensors Node Device (WSN)
- Wireless RF4CE Remote Control
- Wireless Audio
- Wireless Automatic Meter Reader (AMR)
- Electronic Toll Collection (ETC)

The Nano100 Base line, an ultra-low power 32-bit microcontroller with the embedded ARM® Cortex™-M0 core, operates at wide voltage range from 1.8V to 3.6V and runs up to 42 MHz frequency with 32K/64K/128K bytes embedded flash and 8K/16K bytes embedded SRAM. It integrates RTC, 12- channels 12-bit SAR ADC, 2-channels 12-bit DAC and provides high performance connectivity peripheral interfaces such as 2xUART, 3xSPI, 2x$I^2C$, $I^2S$, GPIOs, EBI (External Bus Interface) for external memory-mapped device access and 3xISO-7816-3 for Smart card. The Nano100 Base line supports Brown-out Detector, Power-down mode with RAM retention and fast wake-up via many peripheral interfaces.

The Nano110 LCD line, an ultra-low power 32-bit microcontroller with the embedded ARM® Cortex™-M0 core, operates at wide voltage range from 1.8V to 3.6V and runs up to 42 MHz frequency with 32K/64K/128K bytes embedded flash and 8K/16K bytes embedded SRAM. It integrates LCD 4x40 or 6x38 (COM/Segment). RTC, 12-channels 12-bit SAR ADC, 2-channels 12-bit DAC and provides high performance connectivity peripheral interfaces such as 2xUART, 2xSPI, 2x$I^2C$, $I^2S$, GPIOs, EBI (External Bus Interface) for external memory-mapped device access and 3xISO-7816-3 for Smart card. The Nano110 LCD line supports Brown-out Detector, Power-down mode with RAM retention and fast wake-up via many peripheral interfaces.

The Nano120 USB Connectivity line, an ultra-low power 32-bit microcontroller with the embedded ARM® Cortex™-M0 core, operates at wide voltage range from 1.8V to 3.6V and runs up to 42 MHz frequency with 32K/64K/128K bytes embedded flash and 8K/16K bytes embedded SRAM. It integrates USB 2.0 full-speed device function, RTC, 12-channels12-bit SAR ADC, 2-channels 12-bit DAC and provides high performance connectivity peripheral interfaces such as 2xUART, 3xSPI, 2xI2C, I2S, GPIOs, EBI (External Bus Interface) for external memory-mapped device access and 3xISO-7816-3 for Smart card. The Nano120 USB Connectivity line supports Brown-out Detector, Power-down mode with RAM retention and fast wake-up via many peripheral interfaces.

The Nano130 Advanced line, an ultra-low power 32-bit microcontroller with the embedded ARM® Cortex™-M0 core, operates at wide voltage range from 1.8V to 3.6V and runs up to 42 MHz frequency with 32K/64K/128K bytes embedded flash and 8K/16K bytes embedded SRAM. It integrated LCD 4x40 or 6x38 (COM/Segment), USB 2.0 full-speed device function, RTC, 8-channels 12-bit SAR ADC, 2-channels 12-bit DAC and provides high performance connectivity peripheral interfaces such as 2xUART, 2xSPI, 2x$I^2C$, $I^2S$, GPIOs, EBI (External Bus Interface) for external memory-mapped device access and 3xISO-7816-3 for Smart card. The Nano130 Advanced line supports Brown-out Detector, Power-down mode with RAM retention and fast wake-up via many peripheral interfaces.

| Product Line | UART | SPI | $I^2C$ | $I^2S$ | USB | LCD | ADC | DAC | RTC | EBI | SC | Timer |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Nano100 | ● | ● | ● | ● | | | ● | ● | ● | ● | ● | ● |
| Nano110 | ● | ● | ● | ● | | ● | ● | ● | ● | ● | ● | ● |
| Nano120 | ● | ● | ● | ● | ● | | ● | ● | ● | ● | ● | ● |
| Nano130 | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● | ● |

Table 1-1 Connectivity Support Table

# 2 FEATURES

The equipped features are dependent on the product line and their sub products.

## 2.1 Nano100 Features – Base Line

- Core
  - ARM® Cortex™-M0 core running up to 42 MHz
  - One 24-bit system timer
  - Supports Low Power Sleep mode
  - Single-cycle 32-bit hardware multiplier
  - NVIC for the 32 interrupt inputs, each with 4-levels of priority
  - Serial Wire Debug supports with 2 watchpoints/4 breakpoints
- Brown-out
  - Built-in 2.5V/2.0V/1.7V BOD for wide operating voltage range operation
- Flash EPROM Memory
  - Runs up to 42 MHz with zero wait state for discontinuous address read access
  - 64K/32K/123K bytes application program memory (APROM)
  - 4 KB in system programming (ISP) loader program memory (LDROM)
  - Programmable data flash start address and memory size with 512 bytes page erase unit
  - In System Program (ISP)/In Application Program (IAP) to update on-chip Flash EPROM
- SRAM Memory
  - 16K/8K bytes embedded SRAM
  - Supports DMA mode
- DMA: Supports 8 channels: one VDMA channel, 6 PDMA channels and one CRC channel
  - VDMA
    - Memory-to-memory transfer
    - Supports block transfer with stride
    - Supports word/half-word/byte boundary address
    - Supports address direction: increment and decrement
  - PDMA
    - Peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfer
    - Supports word boundary address
    - Supports word alignment transfer length in memory-to-memory mode
    - Supports word/half-word/byte alignment transfer length in peripheral-to-memory and memory-to-peripheral mode
    - Supports word/half-word/byte transfer data width from/to peripheral

- Supports address direction: increment, fixed, and wrap around

- CRC
  - Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32
    - CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
    - CRC-8: $X^8 + X^2 + X + 1$
    - CRC-16: $X^{16} + X^{15} + X^2 + 1$
    - CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

- Clock Control
  - Flexible selection for different applications
  - Built-in 12 MHz OSC, can be trimmed to 0.25% deviation within all temperature range when turning on auto-trim function (system must have external 32.768 kHz crystal input) otherwise 12 MHz OSC has 2 % deviation within all temperarure range.
  - Low power 10 kHz OSC for watchdog and low power system operation
  - Supports one PLL, up to 120 MHz, for high performance system operation and USB application (48 MHz).
  - External 4~24 MHz crystal input for precise timing operation
  - External 32.768 kHz crystal input for RTC function and low power system operation

- GPIO
  - Three I/O modes:
    - Push-Pull output
    - Open-Drain output
    - Input only with high impendence
  - All inputs with Schmitt trigger
  - I/O pin configured as interrupt source with edge/level setting
  - Supports High Driver and High Sink I/O mode
  - Supports input 5V tolerance, except PA.0 ~ PA.7, PD.0 ~ PD.1 and PC.6 ~ PC.7

- Timer
  - Supports 4 sets of 32-bit timers, each with 24-bit up-counting timer and one 8-bit pre-scale counter
  - Independent Clock Source for each timer
  - Provides one-shot,periodic, output toggle and continuous operation modes
  - Internal trigger event to ADC, DAC and PDMA
  - Supports PDMA mode
  - Wake system up from Power-down mode

- Watchdog Timer
  - Clock Source from LIRC (Internal 10 kHz Low Speed Oscillator Clock)

- Selectable time-out period from 1.6 ms ~ 26 sec (depending on clock source)
- Interrupt or reset selectable when watchdog time-out
- Wake system up from Power-down mode

- Window Watchdog Timer(WWDT)
  - 6-bit down counter and 6-bit compare value to make the window period flexible
  - Selectable WWDT clock pre-scale counter to make WWDT time-out interval variable.
- RTC
  - Supports software compensation by setting frequency compensate register (FCR)
  - Supports RTC counter (second, minute, hour) and calendar counter (day, month, year)
  - Supports Alarm registers (second, minute, hour, day, month, year)
  - Selectable 12-hour or 24-hour mode
  - Automatic leap year recognition
  - Supports periodic time tick interrupt with 8 periodic options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
  - Wake system up from Power-down mode
  - Supports 80 bytes spare registers and a snoop pin to clear the content of these spare registers
- PWM/Capture
  - Supports 2 PWM modules, each has two 16-bit PWM generators
  - Provides eight PWM outputs or four complementary paired PWM outputs
  - Each PWM generator equipped with one clock divider, one 8-bit prescaler, two clock selectors, and one Dead-zone generator for complementary paired PWM
  - (Shared with PWM timers) with eight 16-bit digital capture timers provides eight rising/ falling/both capture inputs.
  - Supports One-shot and Continuous mode
  - Supports Capture interrupt
- UART
  - Up to two 16-byte FIFO UART controllers
  - UART ports with flow control (TX, RX, CTSn and RTSn)
  - Supports IrDA (SIR) function
  - Supports LIN function
  - Supports RS-485 9 bit mode and direction control.
  - Programmable baud rate generator
  - Supports PDMA mode
  - Wake system up from Power-down mode
- SPI
  - Up to three sets of SPI controller

- Master up to 32 MHz, and Slave up to 16 MHz
- Supports SPI/MICROWIRE Master/Slave mode
- Full duplex synchronous serial data transfer
- Variable length of transfer data from 4 to 32 bits
- MSB or LSB first data transfer
- RX and TX on both rising or falling edge of serial clock independently
- Two slave/device select lines when SPI controller is used as the master, and 1 slave/device select line when SPI controller is used as the slave
- Supports byte suspend mode in 32-bit transmission
- Supports two channel PDMA requests, one for transmit and another for receive
- Supports three wire mode, no slave select signal, bi-direction interface
- Wake system up from Power-down mode

- $I^2C$
  - Up to two sets of $I^2C$ device
  - Master/Slave up to 1 Mbit/s
  - Bi-directional data transfer between masters and slaves
  - Multi-master bus (no central master)
  - Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
  - Serial clock synchronization allows devices with different bit rates to communicate via one serial bus
  - Serial clock synchronization used as a handshake mechanism to suspend and resume serial transfer
  - Built-in 14-bit time-out counter requesting the $I^2C$ interrupt if the $I^2C$ bus hangs up and timer-out counter overflows
  - Programmable clocks allowing for versatile rate control
  - Supports 7-bit addressing mode
  - Supports multiple address recognition (four slave addresses with mask option)
- $I^2S$
  - Interface with external audio CODEC
  - Operated as either Master or Slave mode
  - Capable of handling 8, 16, 24 and 32 bit word sizes
  - Supports Mono and stereo audio data
  - Supports $I^2S$ and MSB justified data format
  - Provides two 8 word FIFO data buffers: one for transmitting and the other for receiving
  - Generates interrupt requests when buffer levels cross a programmable boundary
  - Supports two PDMA requests: one for transmitting and the other for receiving

- ADC
  - 12-bit SAR ADC up to 2Msps conversion rate
  - Up to 12-ch single-ended input from external pin (PA.0 ~ PA.7 and PD.0 ~ PD.3)
  - Six internal channels from DAC0, DAC1, internal reference voltage (Int_VREF), Temperature sensor, AVDD, and AVSS.
  - Supports three reference voltage sources from VREF pin, internal reference voltage (Int_VREF), and AVDD.
  - Supports Single Scan, Single Cycle Scan, and Continuous Scan mode
  - Each channel with individual result register
  - Only scan on enabled channels
  - Threshold voltage detection (comparator function)
  - Conversion started by software programming or external input
  - Supports PDMA mode
  - Supports up to four timer time-out events (TMR0, TMR1, TMR2 and TMR3) to enable ADC
- DAC
  - 12-bit monotonic output with 400K conversion rate
  - Supports three reference voltage sources from VREF pin, internal reference voltage (Int_VREF), and AVDD.
  - Synchronized update capability for two DACs (group function)
  - Supports up to four timer time-out events (TMR0, TMR1, TMR2 and TMR3), software or PDMA to trigger DAC to conversion
- SmartCard (SC)
  - Compliant to ISO-7816-3 T=0, T=1
  - Supports up to three ISO-7816-3 ports
  - Separates receive/transmit 4 bytes entry FIFO for data payloads
  - Programmable transmission clock frequency
  - Programmable receiver buffer trigger level
  - Programmable guard time selection (11 ETU ~ 266 ETU)
  - A 24-bit and two 8-bit time-out counters for Answer to Reset (ATR) and waiting times processing
  - Supports auto inverse convention function
  - Supports stop clock level and clock stop (clock keep) function
  - Supports transmitter and receiver error retry and error limit function
  - Supports hardware activation sequence process
  - Supports hardware warm reset sequence process
  - Supports hardware deactivation sequence process
  - Supports hardware auto deactivation sequence when detect the card is removal
  - Supports UART mode (Half Duplex)
- EBI (External bus interface) support

- Accessible space: 64 KB in 8-bit mode or 128 KB in 16-bit mode
- Supports 8bit/16bit data width
- Supports byte write in 16-bit Data Width mode

- One built-in temperature sensor with 1℃ resolution
- 96-bit unique ID
- 128-bit unique customer ID
- Operating Temperature: -40℃~85℃
- Packages:
  - All Green package (RoHS)
  - LQFP 128-pin(14x14) / 64-pin(7x7) / 48-pin(7x7) / QFN 48-pin(7x7)

## 2.2 Nano110 Features – LCD Line

- Core
  - ARM® Cortex™-M0 core running up to 42 MHz
  - One 24-bit system timer
  - Supports Low Power Sleep mode
  - Single-cycle 32-bit hardware multiplier
  - NVIC for the 32 interrupt inputs, each with 4-levels of priority
  - Serial Wire Debug supports with 2 watchpoints/4 breakpoints
- Brown-out
  - Built-in 2.5V/2.0V/1.7V BOD for wide operating voltage range operation
- Flash EPROM Memory
  - Runs up to 42 MHz with zero wait state for discontinuous address read access.
  - 64K/32K/123K bytes application program memory (APROM)
  - 4 KB In System Programming (ISP) loader program memory (LDROM)
  - Programmable data flash start address and memory size with 512 bytes page erase unit
  - In System Program (ISP)/In Application Program (IAP) to update on chip Flash EPROM
- SRAM Memory
  - 16K/8K bytes embedded SRAM
  - Supports DMA mode
- DMA : Supports 8 channels: one VDMA channel,6 PDMA channels, and one CRC channel
  - VDMA
    - Memory-to-memory transfer
    - Supports block transfer with stride
    - Supports word/half-word/byte boundary address
    - Supports address direction: increment and decrement
  - PDMA
    - Peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfer
    - Supports word boundary address
    - Supports word alignment transfer length in memory-to-memory mode
    - Supports word/half-word/byte alignment transfer length in peripheral-to-memory and memory-to-peripheral mode
    - Supports word/half-word/byte transfer data width from/to peripheral
    - Supports address direction: increment, fixed, and wrap around
  - CRC
    - Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and

CRC-32

- CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
- CRC-8: $X^8 + X^2 + X + 1$
- CRC-16: $X^{16} + X^{15} + X^2 + 1$
- CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

- Clock Control
  - Flexible selection for different applications
  - Built-in 12 MHz OSC, can be trimmed to 0.25% deviation within all temperature range when turning on auto-trim function (system must have external 32.768 kHz crystal input) otherwise 12 MHz OSC has 2 % deviation within all temperarure range.
  - Low power 10 kHz OSC for watchdog and low power system operation
  - Supports one PLL, up to 120 MHz, for high performance system operation and USB application (48 MHz).
  - External 4~24 MHz crystal input for precise timing operation
  - External 32.768 kHz crystal input for RTC function and low power system operation
- GPIO
  - Three I/O modes:
    - Push-Pull output
    - Open-Drain output
    - Input only with high impendence
  - All inputs with Schmitt trigger
  - I/O pin configured as interrupt source with edge/level setting
  - Supports High Driver and High Sink I/O mode
  - Supports input 5V tolerance, except PA.0 ~ PA.7, PD.0 ~ PD.1 and PC.6 ~ PC.7)
- Timer
  - Supports 4 sets of 32-bit timers, each with 24-bit up-timer and one 8-bit pre-scale counter
  - Independent Clock Source for each timer
  - Provides one-shot,periodic, output toggle and continuous operation modes
  - Internal trigger event to ADC, DAC and PDMA module
  - Supports PDMA mode
  - Wake system up from Power-down mode
- Watchdog Timer
  - Clock Source from LIRC (Internal 10 kHz Low Speed Oscillator Clock)
  - Selectable time-out period from 1.6 ms ~ 26 sec (depending on clock source)
  - Interrupt or reset selectable when watchdog time-out

- Wake system up from Power-down mode

- Window Watchdog Timer(WWDT)
  - 6-bit down counter and 6-bit compare value to make the window period flexible
  - Selectable WWDT clock pre-scale counter to make WWDT time-out interval variable.

- RTC
  - Supports software compensation by setting frequency compensate register (FCR)
  - Supports RTC counter (second, minute, hour) and calendar counter (day, month, year)
  - Supports Alarm registers (second, minute, hour, day, month, year)
  - Selectable 12-hour or 24-hour mode
  - Automatic leap year recognition
  - Supports periodic time tick interrupt with 8 periodic options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
  - Wake system up from Power-down mode
  - Supports 80 bytes spare registers and a snoop pin to clear the content of these spare registers

- PWM/Capture
  - Supports 2 PWM modules, each has two 16-bit PWM generators
  - Provides eight PWM outputs or four complementary paired PWM outputs
  - Each PWM generator equipped with one clock divider, one 8-bit prescaler, two clock selectors, and one Dead-zone generator for complementary paired PWM
  - (Shared with PWM timers) with eight 16-bit digital capture timers provides eight rising/ falling/both capture inputs.
  - Supports Capture interrupt

- UART
  - Up to two 16-byte FIFO UART controllers
  - UART ports with flow control (TX, RX, CTSn and RTSn)
  - Supports IrDA (SIR) function
  - Supports LIN function
  - Supports RS-485 9 bit mode and direction control (Low Density Only)
  - Programmable baud rate generator
  - Supports PDMA mode
  - Wake system up from Power-down mode

- SPI
  - Up to three sets of SPI controller
  - Master up to 32 MHz, and Slave up to 16 MHz
  - Supports SPI/MICROWIRE Master/Slave mode
  - Full duplex synchronous serial data transfer

- Variable length of transfer data from 4 to 32 bits
- MSB or LSB first data transfer
- RX and TX on both rising or falling edge of serial clock independently
- Two slave/device select lines when SPI controller is as the master, and 1 slave/device select line when SPI controller is as the slave
- Supports byte suspend mode in 32-bit transmission
- Supports two channel PDMA requests, one for transmit and another for receive
- Supports three wire mode, no slave select signal, bi-direction interface
- Wake system up from Power-down mode

- $I^2C$
  - Up to two sets of $I^2C$ device
  - Master/Slave up to 1Mbit/s
  - Bidirectional data transfer between masters and slaves
  - Multi-master bus (no central master)
  - Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
  - Serial clock synchronization allowing devices with different bit rates to communicate via one serial bus
  - Serial clock synchronization used as a handshake mechanism to suspend and resume serial transfer
  - Built-in 14-bit time-out counter requestING the $I^2C$ interrupt if the $I^2C$ bus hangs up and timer-out counter overflows
  - Programmable clocks allow versatile rate control
  - Supports 7-bit addressing mode
  - Supports multiple address recognition (four slave address with mask option)
- $I^2S$
  - Interface with external audio CODEC
  - Operated as either Master or Slave mode
  - Capable of handling 8, 16, 24 and 32 bit word sizes
  - Supports Mono and stereo audio data
  - Supports $I^2S$ and MSB justified data format
  - Provides two 8 word FIFO data buffers: one for transmitting and the other for receiving
  - Generates interrupt requests when buffer levels cross a programmable boundary
  - Supports two PDMA requests: one for transmitting and the other for receiving
- ADC
  - 12-bit SAR ADC up to 2Msps conversion rate
  - Up to 12-ch single-ended input from external pin (PA.0 ~ PA.7 and PD.0 ~ PD.3)

- Six internal channels from DAC0, DAC1, internal reference voltage (Int_VREF), Temperature sensor, AVDD, and AVSS
- Supports three reference voltage sources from VREF pin, internal reference voltage (Int_VREF), and AVDD.
- Single scan/single cycle scan/continuous scan
- Each channel with individual result register
- Only scan on enabled channels
- Threshold voltage detection (comparator function)
- Conversion start by software programming or external input
- Supports PDMA mode
- Supports up to four timer time-out events (TMR0, TMR1, TMR2, and TMR3) to enable ADC

- DAC
  - 12-bit monotonic output with 400K conversion rate
  - Supports three reference voltage sources from VREF pin, internal reference voltage (Int_VREF), and AVDD.
  - Synchronized update capability for two DACs (group function)
  - Supports up to four timer time-out events (TMR0, TMR1, TMR2 and TMR3), software or PDMA to trigger DAC to conversion
- SmartCard (SC)
  - Compliant to ISO-7816-3 T=0, T=1
  - Supports up to three ISO-7816-3 ports
  - Separates receive / transmit 4 bytes entry FIFO for data payloads
  - Programmable transmission clock frequency
  - Programmable receiver buffer trigger level
  - Programmable guard time selection (11 ETU ~ 266 ETU)
  - A 24-bit and two 8-bit time-out counter for Answer to Reset (ATR) and waiting times processing
  - Supports auto inverse convention function
  - Supports stop clock level and clock stop (clock keep) function
  - Supports transmitter and receiver error retry and error limit function
  - Supports hardware activation sequence process
  - Supports hardware warm reset sequence process
  - Supports hardware deactivation sequence process
  - Supports hardware auto deactivation sequence when detect the card is removal
  - Supports UART mode (Half Duplex)
- LCD
  - LCD driver for up to 4 COM x 40 SEG or 6 COM x 38 SEG
  - Supports Static,1/2 bias and 1/3 bias voltage
  - Four display modes; Static, 1/2 duty, 1/3 duty,1/4 duty, 1/5 duty and 1/6 duty.

- Selectable LCD frequency by frequency divider
- Configurable frame frequency
- Internal Charge pump, adjustable contrast adjustment
- Configurable Charge pump frequency
- Blinking capability
- Supports R-type/C-type method
- LCD frame interrupt

- One built-in temperature sensor with 1℃ resolution
- 96-bit unique ID
- 128-bit unique customer ID
- Operating Temperature: -40℃~85℃
- Packages:
  - All Green package (RoHS)
  - LQFP 128-pin(14x14) / 64-pin(10x10) / 64-pin(7x7)

## 2.3 Nano120 Features – USB Connectivity Line

- Core
  - ARM® Cortex™-M0 core running up to 42 MHz
  - One 24-bit system timer
  - Supports Low Power Sleep mode
  - Single-cycle 32-bit hardware multiplier
  - NVIC for the 32 interrupt inputs, each with 4-levels of priority
  - Serial Wire Debug supports with 2 watchpoints/4 breakpoints
- Brown-out
  - Built-in 2.5V/2.0V/1.7V BOD for wide operating voltage range operation
- Flash EPROM Memory
  - Runs up to 42 MHz with zero wait state for discontinuous address read access.
  - 64K/32K/123K bytes application program memory (APROM)
  - 4KB in system programming (ISP) loader program memory (LDROM)
  - Programmable data flash start address and memory size with 512 bytes page erase unit
  - In System Program (ISP)/In Application Program (IAP) to update on chip Flash EPROM
- SRAM Memory
  - 16K/8K bytes embedded SRAM
  - Supports PDMA mode
- DMA: Support 8 channels: one VDMA channel, 6 PDMA channels, and one CRC channel
  - VDMA
    - Memory-to-memory transfer
    - Supports block transfer with stride
    - Supports word/half-word/byte boundary address
    - Supports address direction: increment and decrement
  - PDMA
    - Peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfer
    - Supports word boundary address
    - Supports word alignment transfer length in memory-to-memory mode
    - Supports word/half-word/byte alignment transfer length in peripheral-to-memory and memory-to-peripheral mode
    - Supports word/half-word/byte transfer data width from/to peripheral
    - Supports address: increment, fixed, and wrap around
  - CRC
    - Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and

CRC-32

- CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
- CRC-8: $X^8 + X^2 + X + 1$
- CRC-16: $X^{16} + X^{15} + X^2 + 1$
- CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

- Clock Control
  - Flexible selection for different applications
  - Built-in 12MHz OSC, can be trimmed to 0.25% deviation within all temperature range when turning on auto-trim function (system must have external 32.768 kHz crystal input) otherwise 12 MHz OSC has 2 % deviation within all temperarure range
  - Low power 10 kHz OSC for watchdog and low power system operatin
  - Supports one PLL, up to 120 MHz, for high performance system operation and USB application (48 MHz).
  - External 4~24 MHz crystal input for precise timing operation
  - External 32.768 kHz crystal input for RTC function and low power system operation
- GPIO
  - Three I/O modes:
    - Push-Pull output
    - Open-Drain output
    - Input only with high impendence
  - All inputs with Schmitt trigger
  - I/O pin can be configured as interrupt source with edge/level setting
  - High driver and high sink IO mode support
  - Supports input 5V tolerance (except ADC and DAC shared pins)
- Timer
  - Supports 4 sets of 32-bit timers, each with 24-bit up-timer and one 8-bit pre-scale counter
  - Independent Clock Source for each timer
  - Provides one-shot,periodic, output toggle and continuous operation modes
  - Internal trigger event to ADC, DAC and PDMA module
  - Supports PDMA mode
  - Wake system up from Power-down mode
- Watchdog Timer
  - Clock Source from LIRC. (Internal 10 kHz Low Speed Oscillator Clock)
  - Selectable time-out period from 1.6 ms ~ 26 sec (depending on clock source)
  - Interrupt or reset selectable on watchdog time-out

- Wake system up from Power-down mode

- Window Watchdog Timer(WWDT)
  - 6-bit down counter and 6-bit compare value to make the window period flexible
  - Selectable WWDT clock pre-scale counter to make WWDT time-out interval variable.

- RTC
  - Supports software compensation by setting frequency compensate register (FCR)
  - Supports RTC counter (second, minute, hour) and calendar counter (day, month, year)
  - Supports Alarm registers (second, minute, hour, day, month, year)
  - Selectable 12-hour or 24-hour mode
  - Automatic leap year recognition
  - Supports periodic time tick interrupt with 8 periodic options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
  - Wake system up from Power-down or Idle mode
  - Support 80 bytes spare registers and a snoop pin to clear the content of these spare registers

- PWM/Capture
  - Supports 2 PWM module, each has two 16-bit PWM generators
  - Provide eight PWM outputs or four complementary paired PWM outputs
  - Each PWM generator equipped with one clock divider, one 8-bit prescaler, two clock selectors, and one Dead-Zone generator for complementary paired PWM
  - (Shared with PWM timers) with eight 16-bit digital capture timers provides eight rising/ falling/both capture inputs.
  - Supports one shot and continuous mode
  - Supports Capture interrupt

- UART
  - Up to two 16-byte FIFO UART controllers
  - UART ports with flow control (TX, RX, CTSn and RTSn)
  - Supports IrDA (SIR) function
  - Supports LIN function
  - Supports RS-485 9 bit mode and direction control. (Low Density Only)
  - Programmable baud rate generator
  - Supports PDMA mode
  - Wake system up from Power-down mode

- SPI
  - Up to three sets of SPI controller
  - Master up to 32 MHz, and Slave up to 16 MHz
  - Supports SPI/MICROWIRE Master/Slave mode

- Full duplex synchronous serial data transfer
- Variable length of transfer data from 4 to 32 bits
- MSB or LSB first data transfer
- RX and TX on both rising or falling edge of serial clock independently
- Two slave/device select lines when SPI controller is as the master, and 1 slave/device select line when SPI controller is as the slave
- Supports byte suspend mode in 32-bit transmission
- Supports two channel PDMA requests, one for transmit and another for receive
- Supports three wire, no slave select signal, bi-direction interface
- Wake system up from Power-down mode

- $I^2C$
  - Up to two sets of $I^2C$ device
  - Master/Slave up to 1Mbit/s
  - Bi-directional data transfer between masters and slaves
  - Multi-master bus (no central master)
  - Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
  - Serial clock synchronization allowing devices with different bit rates to communicate via one serial bus
  - Serial clock synchronization used as a handshake mechanism to suspend and resume serial transfer
  - Built-in 14-bit time-out counter requesting the $I^2C$ interrupt if the $I^2C$ bus hangs up and timer-out counter overflows
  - Programmable clocks allow versatile rate control
  - Supports 7-bit addressing mode
  - Supports multiple address recognition (four slave addresses with mask option)
- $I^2S$
  - Interface with external audio CODEC
  - Operated as either Master or Slave mode
  - Capable of handling 8, 16, 24 and 32 bit word sizes
  - Supports Mono and stereo audio data
  - Supports $I^2S$ and MSB justified data format
  - Provides two 8 word FIFO data buffers: one for transmitting and the other for receiving
  - Generates interrupt requests when buffer levels cross a programmable boundary
  - Supports two PDMA requests: one for transmitting and the other for receiving
- ADC
  - 12-bit SAR ADC up to 2Msps conversion rate

- Up to 12-ch single-ended input from external pin (PA.0 ~ PA.7 and PD.0 ~ PD.3).
- Six internal channels from DAC0, DAC1, internal reference voltage (Int_VREF), Temperature sensor, AVDD, and AVSS.
- Supports three reference voltage sources from VREF pin, internal reference voltage (Int_VREF), and AVDD
- Supports single scan, single cycle scan, and continuous scan modes
- Each channel with individual result register
- Only scan on enabled channels
- Threshold voltage detection (comparator function)
- Conversion start by software programming or external input
- Supports PDMA mode
- Supports up to four timer time-out events (TMR0, TMR1, TMR2 and TMR3) to enable ADC

- DAC
  - 12-bit monotonic output with 400K conversion rate
  - Supports three reference voltage sources from VREF pin, internal reference voltage (Int_VREF), and AVDD.
  - Synchronized update capability for two DACs (group function)
  - Supports up to four timer time-out event (TMR0, TMR1, TMR2 and TMR3), software or PDMA to trigger DAC to conversion
- SmartCard (SC)
  - Compliant to ISO-7816-3 T=0, T=1
  - Supports up to three ISO-7816-3 ports
  - Separates receive / transmit 4 bytes entry FIFO for data payloads
  - Programmable transmission clock frequency
  - Programmable receiver buffer trigger level
  - Programmable guard time selection (11 ETU ~ 266 ETU)
  - A 24-bit and two 8-bit time-out counter for Answer to Reset (ATR) and waiting times processing
  - Supports auto inverse convention function
  - Supports stop clock level and clock stop (clock keep) function
  - Supports transmitter and receiver error retry and error limit function
  - Supports hardware activation sequence process
  - Supports hardware warm reset sequence process
  - Supports hardware deactivation sequence process
  - Supports hardware auto deactivation sequence when detect the card is removal
  - Supports UART mode (Half Duplex)
- USB 2.0 Full-Speed Device
  - One set of USB 2.0 FS Device 12 Mbps

- On-chip USB Transceiver
- Provides 1 interrupt source with 4 interrupt events
- Supports Control, Bulk In/Out, Interrupt and Isochronous transfers
- Auto suspend function when no bus signaling for 3 ms
- Provides 8 programmable endpoints
- Includes 512 Bytes internal SRAM as USB buffer
- Provides remote wake-up capability

- EBI (External bus interface) support
  - Accessible space: 64 KB in 8-bit mode or 128 KB in 16-bit mode
  - Supports 8bit/16bit data width
  - Supports byte write in 16-bit Data Width mode
- One built-in temperature sensor with 1℃ resolution
- 96-bit unique ID
- 128-bit unique customer ID
- Operating Temperature: -40℃~85℃
- Packages:
  - All Green package (RoHS)
  - LQFP 128-pin(14x14) / 64-pin(7x7) / 48-pin(7x7)

## 2.4 Nano130 Features – Advanced Line

- Core
  - ARM® Cortex™-M0 core running up to 42 MHz
  - One 24-bit system timer
  - Supports Low Power Sleep mode
  - Single-cycle 32-bit hardware multiplier
  - NVIC for the 32 interrupt inputs, each with 4-levels of priority
  - Serial Wire Debug supports with 2 watchpoints/4 breakpoints
- Brown-out
  - Built-in 2.5V/2.0V/1.7V BOD for wide operating voltage range operation
- Flash EPROM Memory
  - Runs up to 42 MHz with zero wait state for discontinuous address read access.
  - 64K/32K/123K bytes application program memory (APROM)
  - 4KB in system programming (ISP) loader program memory (LDROM)
  - Programmable data flash start address and memory size with 512 bytes page erase unit
  - In System Program (ISP)/In Application Program (IAP) to update on chip Flash EPROM
- SRAM Memory
  - 16K/8K bytes embedded SRAM
  - Supports DMA mode
- DMA : Supports 8 channels: one VDMA channel,6 PDMA channels, and one CRC 33egiste
  - VDMA
    - Memory-to-memory transfer
    - Supports block transfer with stride
    - Supports word/half-word/byte boundary address
    - Supports address direction: increment and decrement
  - PDMA
    - Peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfer
    - Supports word boundary address
    - Supports word alignment transfer length in memory-to-memory mode
    - Supports word/half-word/byte alignment transfer length in peripheral-to-memory and memory-to-peripheral mode
    - Supports word/half-word/byte transfer data width from/to peripheral
    - Supports address direction: increment, fixed, and wrap around
  - CRC
    - Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and

CRC-32

- CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
- CRC-8: $X^8 + X^2 + X + 1$
- CRC-16: $X^{16} + X^{15} + X^2 + 1$
- CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

- Clock Control
  - Flexible selection for different applications
  - Built-in 12MHz OSC, can be trimmed to 0.25% deviation within all temperature range when turning on auto-trim function (system must have external 32.768 kHz crystal input) otherwise 12 MHz OSC has 2 % deviation within all temperarure range.
  - Low power 10 kHz OSC for watchdog and low power system operation
  - Supports one PLL, up to 120 MHz, for high performance system operation and USB application (48 MHz).
  - External 4~24 MHz crystal input for precise timing operation
  - External 32.768 kHz crystal input for RTC function and low power system operation
- GPIO
  - Three I/O modes:
    - Push-Pull output
    - Open-Drain output
    - Input only with high impendence
  - All inputs with Schmitt trigger
  - I/O pin configured as interrupt source with edge/level setting
  - Supports High Driver and High Sink I/O mode
  - Supports input 5V tolerance (except ADC and DAC shared pins)
- Timer
  - Supports 4 sets of 32-bit timers with 24-bit up-timer and one 8-bit pre-scale counter
  - Independent Clock Source for each timer
  - Provides one-shot,periodic, output toggle and continuous operation modes
  - Supports internal trigger event to ADC, DAC and PDMA module
  - Wake system up from Power-down mode
- Watchdog Timer
  - Clock Source is from LIRC. (Internal 10 kHz Low Speed Oscillator Clock)
  - Selectable time-out period from 1.6ms ~ 26sec (depends on clock source)
  - Interrupt or reset selectable on watchdog time-out
  - WDT can wake system up from Power-down mode

- Window Watchdog Timer(WWDT)
  - 6-bit down counter and 6-bit compare value to make the window period flexible
  - Selectable WWDT clock pre-scale counter to make WWDT time-out interval variable.
- RTC
  - Supports software compensation by setting frequency compensate register (FCR)
  - Supports RTC counter (second, minute, hour) and calendar counter (day, month, year)
  - Supports Alarm registers (second, minute, hour, day, month, year)
  - Selectable 12-hour or 24-hour mode
  - Automatic leap year recognition
  - Supports periodic time tick interrupt with 8 periodic options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
  - Wake system up from Power-down or Idle mode
  - Supports 80 bytes spare registers and a snoop pin to clear the content of these spare registers
- PWM/Capture
  - Supports 2 PWM module, each with two 16-bit PWM generators
  - Provides eight PWM outputs or four complementary paired PWM outputs
  - Each PWM generator equipped with one clock divider, one 8-bit prescaler, two clock selectors, and one Dead-Zone generator for complementary paired PWM
  - (Shared with PWM timers) with eight 16-bit digital capture timers provides eight rising/ falling/both capture inputs.
  - Supports Capture interrupt
- UART
  - Up to two 16-byte FIFO UART controllers
  - UART ports with flow control (TX, RX, CTSn and RTSn)
  - Supports IrDA (SIR) function
  - Supports LIN function
  - Supports RS-485 9 bit mode and direction control (Low Density Only)
  - Programmable baud rate generator
  - Supports PDMA mode
  - Wake system up from Power-down or Idle mode
- SPI
  - Up to 3 sets of SPI controller
  - Master up to 32 MHz, and Slave up to 16 MHz
  - Supports SPI/MICROWIRE Master/Slave mode
  - Full duplex synchronous serial data transfer
  - Variable length of transfer data from 4 to 32 bits

- MSB or LSB first data transfer
- RX and TX on both rising or falling edge of serial clock independently
- Two slave/device select lines when used as the master, and 1 slave/device select line when used as the slave
- Supports byte suspend mode in 32-bit transmission
- Supports two channel PDMA request, one for transmit and another for receive
- Supports three wire, no slave select signal, bi-direction interface
- Wake system up from Power-down or Idle mode

- $I^2C$
  - Up to two sets of $I^2C$ device
  - Master/Slave up to 1Mbit/s
  - Bi-directional data transfer between masters and slaves
  - Multi-master bus (no central master)
  - Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
  - Serial clock synchronization allowing devices with different bit rates to communicate via one serial bus
  - Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer
  - Built-in 14-bit time-out counter will request the $I^2C$ interrupt if the $I^2C$ bus hangs up and timer-out counter overflows
  - Programmable clocks allowing for versatile rate control
  - Supports 7-bit addressing mode
  - Supports multiple address recognition (four slave addresses with mask option)
- $I^2S$
  - Interface with external audio CODEC
  - Operate as either Master or Slave mode
  - Capable of handling 8, 16, 24 and 32 bit word sizes
  - Supports Mono and stereo audio data
  - Supports $I^2S$ and MSB justified data format
  - Provides two 8 word FIFO data buffers: one for transmitting and the other for receiving
  - Generates interrupt requests when buffer levels cross a programmable boundary
  - Supports two PDMA requests: one for transmitting and the other for receiving
- ADC
  - 12-bit SAR ADC up to 2Msps conversion rate
  - Up to 12-ch single-ended input from external pin (PA.0 ~ PA.7 and PD.0 ~ PD.3)
  - Six internal channels from DAC0, DAC1, internal reference voltage (Int_VREF), Temperature sensor, AVDD, and AVSS.

- Supports three reference voltage sources from VREF pin, internal reference voltage (Int_VREF), and AVDD
- Single scan/single cycle scan/continuous scan
- Each channel with individual result register
- Scan on enabled channels
- Threshold voltage detection (comparator function)
- Conversion start by software programming or external input
- Supports PDMA mode
- Supports up to four timer time-out events (TMR0, TMR1, TMR2 and TMR3) to enable ADC

- DAC
  - 12-bit monotonic output with 400K conversion rate
  - Supports three reference voltage sources from VREF pin, internal reference voltage (Int_VREF), and AVDD.
  - Synchronized update capability for two DACs (group function)
  - Supports up to four timer time-out events (TMR0, TMR1, TMR2 and TMR3), software or PDMA to trigger DAC to conversion
- SmartCard (SC)
  - Compliant to ISO-7816-3 T=0, T=1
  - Supports up to three ISO-7816-3 ports
  - Separates receive/transmit 4 bytes entry FIFO for data payloads
  - Programmable transmission clock frequency
  - Programmable receiver buffer trigger level
  - Programmable guard time selection (11 ETU ~ 266 ETU)
  - A 24-bit and two 8-bit time-out counter for Answer to Reset (ATR) and waiting times processing
  - Supports auto inverse convention function
  - Supports stop clock level and clock stop (clock keep) function
  - Supports transmitter and receiver error retry and error limit function
  - Supports hardware activation sequence process
  - Supports hardware warm reset sequence process
  - Supports hardware deactivation sequence process
  - Supports hardware auto deactivation sequence when detecting the card is removed
  - Support UART mode (Half Duplex)
- LCD
  - LCD driver for up to 4 COM x 40 SEG or 6 COM x 38 SEG
  - Supports Static,1/2 bias and 1/3 bias voltage
  - Four display modes: Static, 1/2 duty, 1/3 duty, 1/4 duty, 1/5 duty and 1/6 duty.
  - Selectable LCD frequency by frequency divider

- Configurable frame frequency
- Internal Charge pump, adjustable contrast adjustment
- Configurable Charge pump frequency
- Blinking capability
- Supports R-type/C-type method
- LCD frame interrupt

- USB 2.0 Full-speed Device
  - One set of USB 2.0 FS Device 12 Mbps
  - On-chip USB Transceiver
  - Provides 1 interrupt source with 4 interrupt events
  - Supports Control, Bulk In/Out, Interrupt and Isochronous transfers
  - Auto suspend function when no bus signaling for 3 ms
  - Provides 8 programmable endpoints
  - Includes 512 Bytes internal SRAM as USB buffer
  - Provides remote wake-up capability
- EBI (External bus interface)
  - Accessible space: 64 KB in 8-bit mode or 128 KB in 16-bit mode
  - Supports 8bit/16bit data width
  - Supports byte write in 16-bit data width mode
- One built-in temperature sensor with 1℃ resolution
- 96-bit unique ID
- 128-bit unique customer ID
- Operating Temperature: -40℃~85℃
- Packages:
  - All Green package (RoHS)
  - LQFP 128-pin(14x14) / 64-pin (7x7)

# 3 PARTS INFORMATION LIST AND PIN CONFIGURATION

## 3.1 NuMicro™ Nano100 Series Selection Code

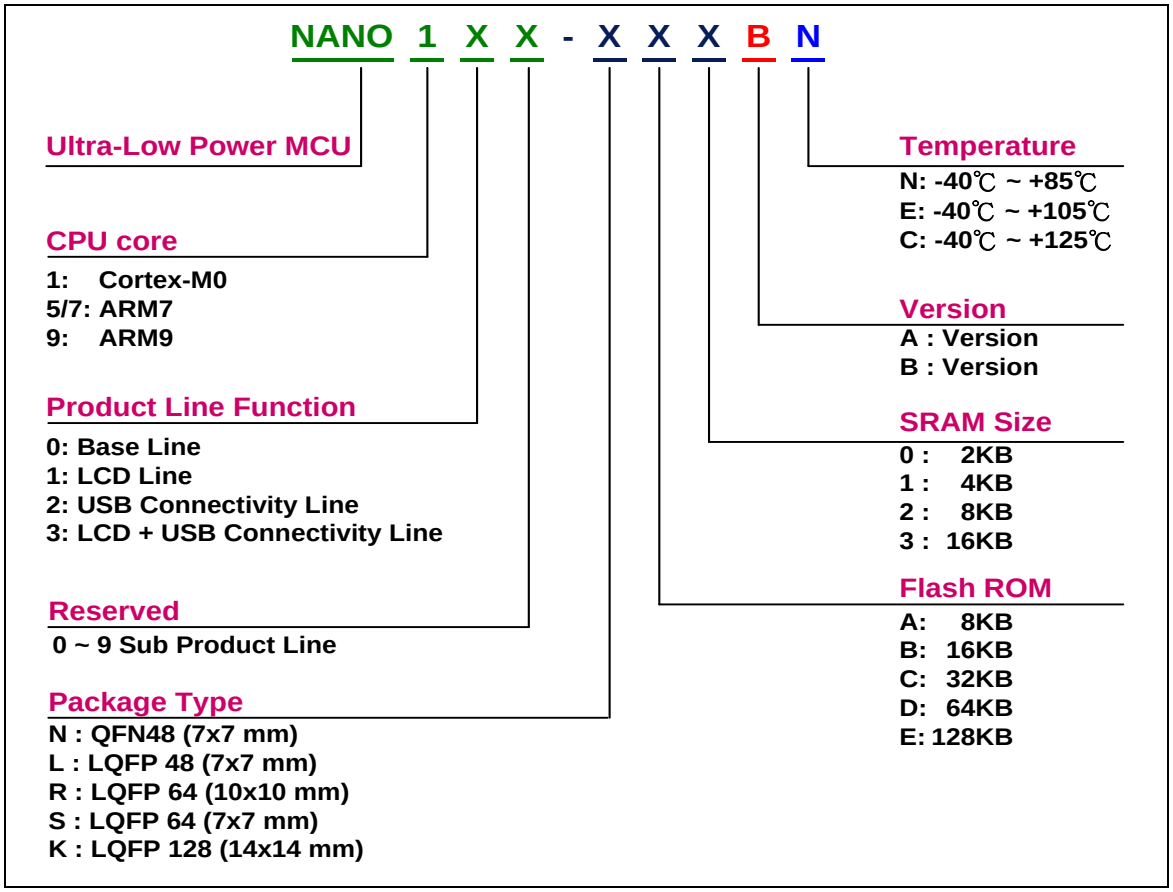



Figure 3-1 NuMicro™ Nano100 Series Selection Code

## 3.2 NuMicro™ Nano100 Products Selection Guide

### 3.2.1 NuMicro™ Nano100 Base Line Selection Guide

| Part No. | Flash (Kbytes) | SRAM (Kbytes) | Data Flash | ISP ROM (Kbytes) | I/O | Timer (32-bit) | Connectivity | | | | I2S | PWM (16-bit | ADC (12-bit) | RTC | EBI | IRC 10KHz 12MHz | PDMA | LCD | DAC (12-bit) | ISO-7816-3 | ISP ICP | Package |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | UART | SPI | I2C | USB | | | | | | | | | | | | |
| NANO100NC2BN | 32K | 8K | Configurable | 4K | up to 38 | 4x32-bit | 4 | 3 | 2 | - | 1 | 0 | 7 | V | - | V | 8 | - | 2 | 2 | V | QFN48* |
| NANO100ND2BN | 64K | 8K | Configurable | 4K | up to 38 | 4x32-bit | 4 | 3 | 2 | - | 1 | 6 | 7 | V | - | V | 8 | - | 2 | 2 | V | QFN48* |
| NANO100ND3BN | 64K | 16K | Configurable | 4K | up to 38 | 4x32-bit | 4 | 3 | 2 | - | 1 | 6 | 7 | V | - | V | 8 | - | 2 | 2 | V | QFN48* |
| NANO100NE3BN | 128K | 16K | Configurable | 4K | up to 38 | 4x32-bit | 4 | 3 | 2 | - | 1 | 6 | 7 | V | - | V | 8 | - | 2 | 2 | V | QFN48* |
| NANO100LC2BN | 32K | 8K | Configurable | 4K | up to 38 | 4x32-bit | 4 | 3 | 2 | - | 1 | 6 | 7 | V | - | V | 8 | - | 2 | 2 | V | LQFP48 |
| NANO100LD2BN | 64K | 8K | Configurable | 4K | up to 38 | 4x32-bit | 4 | 3 | 2 | - | 1 | 6 | 7 | V | - | V | 8 | - | 2 | 2 | V | LQFP48 |
| NANO100LD3BN | 64K | 16K | Configurable | 4K | up to 38 | 4x32-bit | 4 | 3 | 2 | - | 1 | 6 | 7 | V | - | V | 8 | - | 2 | 2 | V | LQFP48 |
| NANO100LE3BN | 128K | 16K | Configurable | 4K | up to 38 | 4x32-bit | 4 | 3 | 2 | - | 1 | 6 | 7 | V | - | V | 8 | - | 2 | 2 | V | LQFP48 |
| NANO100SC2BN | 32K | 8K | Configurable | 4K | up to 52 | 4x32-bit | 5 | 3 | 2 | - | 1 | 8 | 7 | V | - | V | 8 | - | 2 | 3 | V | LQFP64 |
| NANO100SD2BN | 64K | 8K | Configurable | 4K | up to 52 | 4x32-bit | 5 | 3 | 2 | - | 1 | 8 | 7 | V | - | V | 8 | - | 2 | 3 | V | LQFP64 |
| NANO100SD3BN | 64K | 16K | Configurable | 4K | up to 52 | 4x32-bit | 5 | 3 | 2 | - | 1 | 8 | 7 | V | - | V | 8 | - | 2 | 3 | V | LQFP64 |
| NANO100SE3BN | 128K | 16K | Configurable | 4K | up to 52 | 4x32-bit | 5 | 3 | 2 | - | 1 | 8 | 7 | V | - | V | 8 | - | 2 | 3 | V | LQFP64 |
| NANO100KD3BN | 64K | 16K | Configurable | 4K | up to 86 | 4x32-bit | 5 | 3 | 2 | - | 1 | 8 | 12 | V | V | V | 8 | - | 2 | 3 | V | LQFP128 |
| NANO100KE3BN | 128K | 16K | Configurable | 4K | up to 86 | 4x32-bit | 5 | 3 | 2 | - | 1 | 8 | 12 | V | V | V | 8 | - | 2 | 3 | V | LQFP128 |

QFN*48* : 7x7, pitch 0.5 mm ; LQFP48 : 7x7, pitch 0.5 mm ; LQFP64 : 7x7, pitch 0.4 mm ; LQFP128 : 14x14, pitch 0.4 mm

Table 3-1 Nano100 Base Line Selection Table

### 3.2.2 NuMicro™ Nano110 LCD Line Selection Guide

| Part No. | Flash (Kbytes) | SRAM (Kbytes) | Data Flash | ISP ROM (Kbytes) | I/O | Timer (32-bit) | Connectivity | | | | I2S | PWM (16-bit | ADC (12-bit) | RTC | EBI | IRC 10KHz 12MHz | PDMA | LCD | DAC (12-bit) | ISO-7816-3 | ISP ICP | Package |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | UART | SPI | I2C | USB | | | | | | | | | | | | |
| NANO110SC2BN | 32K | 8K | Configurable | 4K | up to 51 | 4x32-bit | 5 | 3 | 2 | - | 1 | 7 | 7 | V | - | V | 8 | 4x31, 6x29 | 2 | 3 | V | LQFP64 |
| NANO110SD2BN | 64K | 8K | Configurable | 4K | up to 51 | 4x32-bit | 5 | 3 | 2 | - | 1 | 7 | 7 | V | - | V | 8 | 4x31, 6x29 | 2 | 3 | V | LQFP64 |
| NANO110SD3BN | 64K | 16K | Configurable | 4K | up to 51 | 4x32-bit | 5 | 3 | 2 | - | 1 | 7 | 7 | V | - | V | 8 | 4x31, 6x29 | 2 | 3 | V | LQFP64 |
| NANO110SE3BN | 128K | 16K | Configurable | 4K | up to 51 | 4x32-bit | 5 | 3 | 2 | - | 1 | 7 | 7 | V | - | V | 8 | 4x31, 6x29 | 2 | 3 | V | LQFP64 |
| NANO110RC2BN | 32K | 8K | Configurable | 4K | up to 51 | 4x32-bit | 5 | 3 | 2 | - | 1 | 7 | 7 | V | - | V | 8 | 4x31, 6x29 | 2 | 3 | V | LQFP64* |
| NANO110RD2BN | 64K | 8K | Configurable | 4K | up to 51 | 4x32-bit | 5 | 3 | 2 | - | 1 | 7 | 7 | V | - | V | 8 | 4x31, 6x29 | 2 | 3 | V | LQFP64* |
| NANO110RD3BN | 64K | 16K | Configurable | 4K | up to 51 | 4x32-bit | 5 | 3 | 2 | - | 1 | 7 | 7 | V | - | V | 8 | 4x31, 6x29 | 2 | 3 | V | LQFP64* |
| NANO110RE3BN | 128K | 16K | Configurable | 4K | up to 51 | 4x32-bit | 5 | 3 | 2 | - | 1 | 7 | 7 | V | - | V | 8 | 4x31, 6x29 | 2 | 3 | V | LQFP64* |
| NANO110KC2BN | 32K | 8K | Configurable | 4K | up to 86 | 4x32-bit | 5 | 3 | 2 | - | 1 | 8 | 12 | V | V | V | 8 | 4x40, 6x38 | 2 | 3 | V | LQFP128 |
| NANO110KD2BN | 64K | 8K | Configurable | 4K | up to 86 | 4x32-bit | 5 | 3 | 2 | - | 1 | 8 | 12 | V | V | V | 8 | 4x40, 6x38 | 2 | 3 | V | LQFP128 |
| NANO110KD3BN | 64K | 16K | Configurable | 4K | up to 86 | 4x32-bit | 5 | 3 | 2 | - | 1 | 8 | 12 | V | V | V | 8 | 4x40, 6x38 | 2 | 3 | V | LQFP128 |
| NANO110KE3BN | 128K | 16K | Configurable | 4K | up to 86 | 4x32-bit | 5 | 3 | 2 | - | 1 | 8 | 12 | V | V | V | 8 | 4x40, 6x38 | 2 | 3 | V | LQFP128 |

LQFP64 : 7x7, pitch 0.4 mm ; LQFP64* : 10x10, pitch 0.5 mm ; LQFP128 : 14x14, pitch 0.4 mm

Table 3-2 Nano110 LCD Line Selection Table

### 3.2.3 NuMicro™ Nano120 USB Connectivity Line Selection Guide

| Part No. | Flash (Kbytes) | SRAM (Kbytes) | Data Flash | ISP ROM (Kbytes) | I/O | Timer (32-bit) | Connectivity | | | | I2S | PWM (16-bit | ADC (12-bit) | RTC | EBI | IRC 10KHz 12MHz | PDMA | LCD | DAC (12-bit) | ISO-7816-3 | ISP ICP | Package |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | UART | SPI | I2C | USB | | | | | | | | | | | | |
| NANO120LC2BN | 32K | 8K | Configurable | 4K | up to 34 | 4x32-bit | 4 | 3 | 2 | 1 | 1 | 4 | 7 | V | - | V | 8 | - | 2 | 2 | V | LQFP48 |
| NANO120LD2BN | 64K | 8K | Configurable | 4K | up to 34 | 4x32-bit | 4 | 3 | 2 | 1 | 1 | 4 | 7 | V | - | V | 8 | - | 2 | 2 | V | LQFP48 |
| NANO120LD3BN | 64K | 16K | Configurable | 4K | up to 34 | 4x32-bit | 4 | 3 | 2 | 1 | 1 | 4 | 7 | V | - | V | 8 | - | 2 | 2 | V | LQFP48 |
| NANO120LE3BN | 128K | 16K | Configurable | 4K | up to 34 | 4x32-bit | 4 | 3 | 2 | 1 | 1 | 4 | 7 | V | - | V | 8 | - | 2 | 2 | V | LQFP48 |
| NANO120SC2BN | 32K | 8K | Configurable | 4K | up to 48 | 4x32-bit | 5 | 3 | 2 | 1 | 1 | 8 | 7 | V | - | V | 8 | - | 2 | 3 | V | LQFP64 |
| NANO120SD2BN | 64K | 8K | Configurable | 4K | up to 48 | 4x32-bit | 5 | 3 | 2 | 1 | 1 | 8 | 7 | V | - | V | 8 | - | 2 | 3 | V | LQFP64 |
| NANO120SD3BN | 64K | 16K | Configurable | 4K | up to 48 | 4x32-bit | 5 | 3 | 2 | 1 | 1 | 8 | 7 | V | - | V | 8 | - | 2 | 3 | V | LQFP64 |
| NANO120SE3BN | 128K | 16K | Configurable | 4K | up to 48 | 4x32-bit | 5 | 3 | 2 | 1 | 1 | 8 | 7 | V | - | V | 8 | - | 2 | 3 | V | LQFP64 |
| NANO120KD3BN | 64K | 16K | Configurable | 4K | up to 86 | 4x32-bit | 5 | 3 | 2 | 1 | 1 | 8 | 8 | V | V | V | 8 | - | 2 | 3 | V | LQFP128 |
| NANO120KE3BN | 128K | 16K | Configurable | 4K | up to 86 | 4x32-bit | 5 | 3 | 2 | 1 | 1 | 8 | 8 | V | V | V | 8 | - | 2 | 3 | V | LQFP128 |

LQFP48 : 7x7, pitch 0.5 mm ; LQFP64 : 7x7, pitch 0.4 mm ; LQFP128 : 14x14, pitch 0.4 mm

Table 3-3 Nano120 USB Connectivity Line Selection Table

### 3.2.4 NuMicro™ Nano130 Advanced Line Selection Guide

| Part No. | Flash (Kbytes) | SRAM (Kbytes) | Data Flash | ISP ROM (Kbytes) | I/O | Timer (32-bit) | Connectivity | | | | I2S | PWM (16-bit | ADC (12-bit) | RTC | EBI | IRC 10KHz 12MHz | PDMA | LCD | DAC (12-bit) | ISO-7816-3 | ISP ICP | Package |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | UART | SPI | I2C | USB | | | | | | | | | | | | |
| NANO130SC2BN | 32K | 8K | Configurable | 4K | up to 47 | 4x32-bit | 5 | 3 | 2 | 1 | 1 | 7 | 7 | V | - | V | 8 | 4x31, 6x29 | 2 | 3 | V | LQFP64 |
| NANO130SD2BN | 64K | 8K | Configurable | 4K | up to 47 | 4x32-bit | 5 | 3 | 2 | 1 | 1 | 7 | 7 | V | - | V | 8 | 4x31, 6x29 | 2 | 3 | V | LQFP64 |
| NANO130SD3BN | 64K | 16K | Configurable | 4K | up to 47 | 4x32-bit | 5 | 3 | 2 | 1 | 1 | 7 | 7 | V | - | V | 8 | 4x31, 6x29 | 2 | 3 | V | LQFP64 |
| NANO130SE3BN | 128K | 16K | Configurable | 4K | up to 47 | 4x32-bit | 5 | 3 | 2 | 1 | 1 | 7 | 7 | V | - | V | 8 | 4x31, 6x29 | 2 | 3 | V | LQFP64 |
| NANO130KC2BN | 32K | 8K | Configurable | 4K | up to 86 | 4x32-bit | 5 | 3 | 2 | 1 | 1 | 8 | 8 | V | V | V | 8 | 4x40, 6x38 | 2 | 3 | V | LQFP128 |
| NANO130KD2BN | 64K | 8K | Configurable | 4K | up to 86 | 4x32-bit | 5 | 3 | 2 | 1 | 1 | 8 | 8 | V | V | V | 8 | 4x40, 6x38 | 2 | 3 | V | LQFP128 |
| NANO130KD3BN | 64K | 16K | Configurable | 4K | up to 86 | 4x32-bit | 5 | 3 | 2 | 1 | 1 | 8 | 8 | V | V | V | 8 | 4x40, 6x38 | 2 | 3 | V | LQFP128 |
| NANO130KE3BN | 128K | 16K | Configurable | 4K | up to 86 | 4x32-bit | 5 | 3 | 2 | 1 | 1 | 8 | 8 | V | V | V | 8 | 4x40, 6x38 | 2 | 3 | V | LQFP128 |

LQFP64 : 7x7, pitch 0.4 mm ; LQFP128 : 14x14, pitch 0.4 mm

Table 3-4 Nano130 Advanced Line Selection Table

## 3.3 Pin Configuration

### 3.3.1 NuMicro™ Nano100 Pin Diagrams

#### 3.3.1.1 NuMicro™ Nano100 LQFP 128-pin

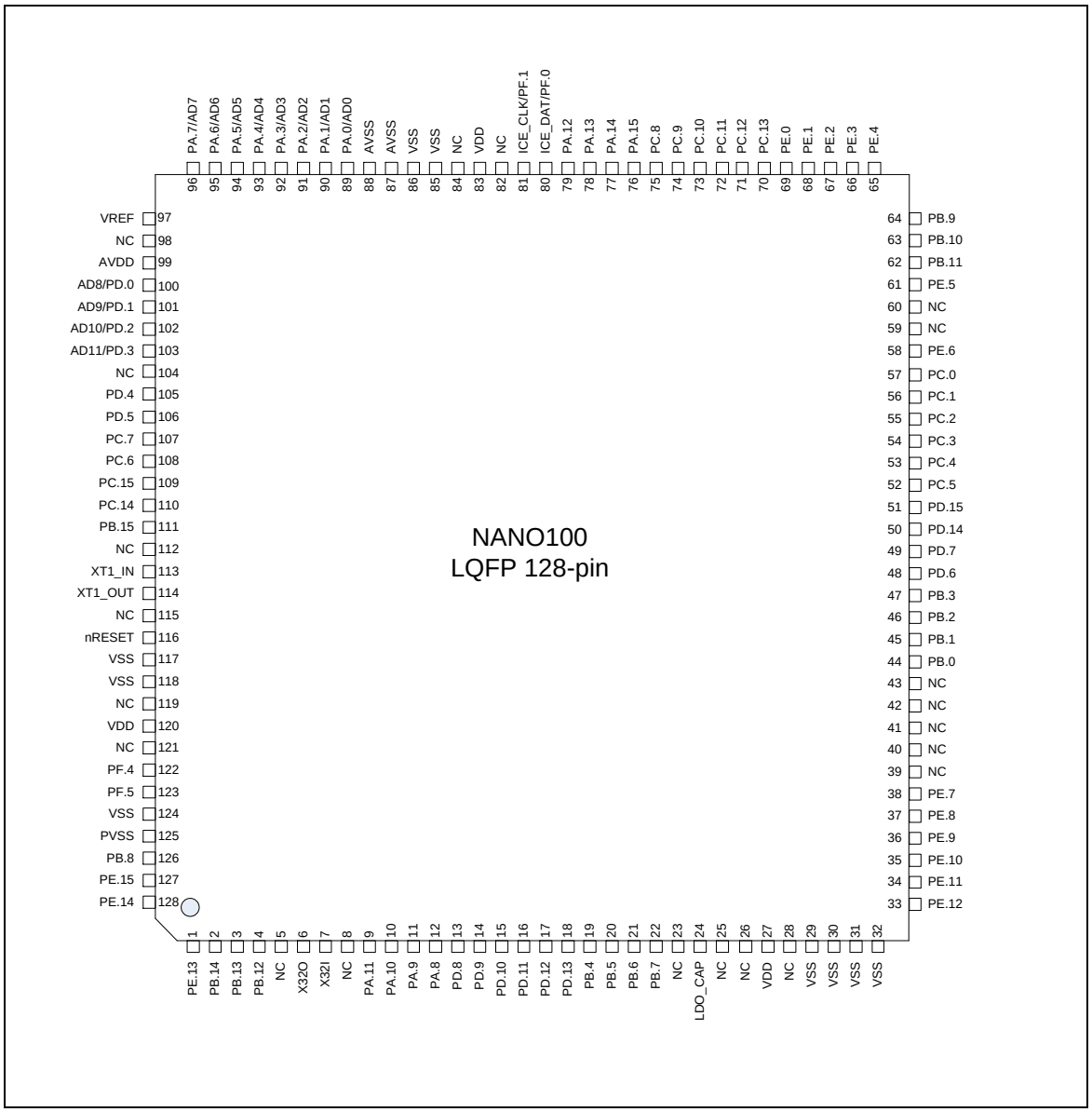



Figure 3-2 NuMicro™ Nano100 LQFP 128-pin Diagram

3.3.1.2 NuMicro™ Nano100 LQFP 64-pin

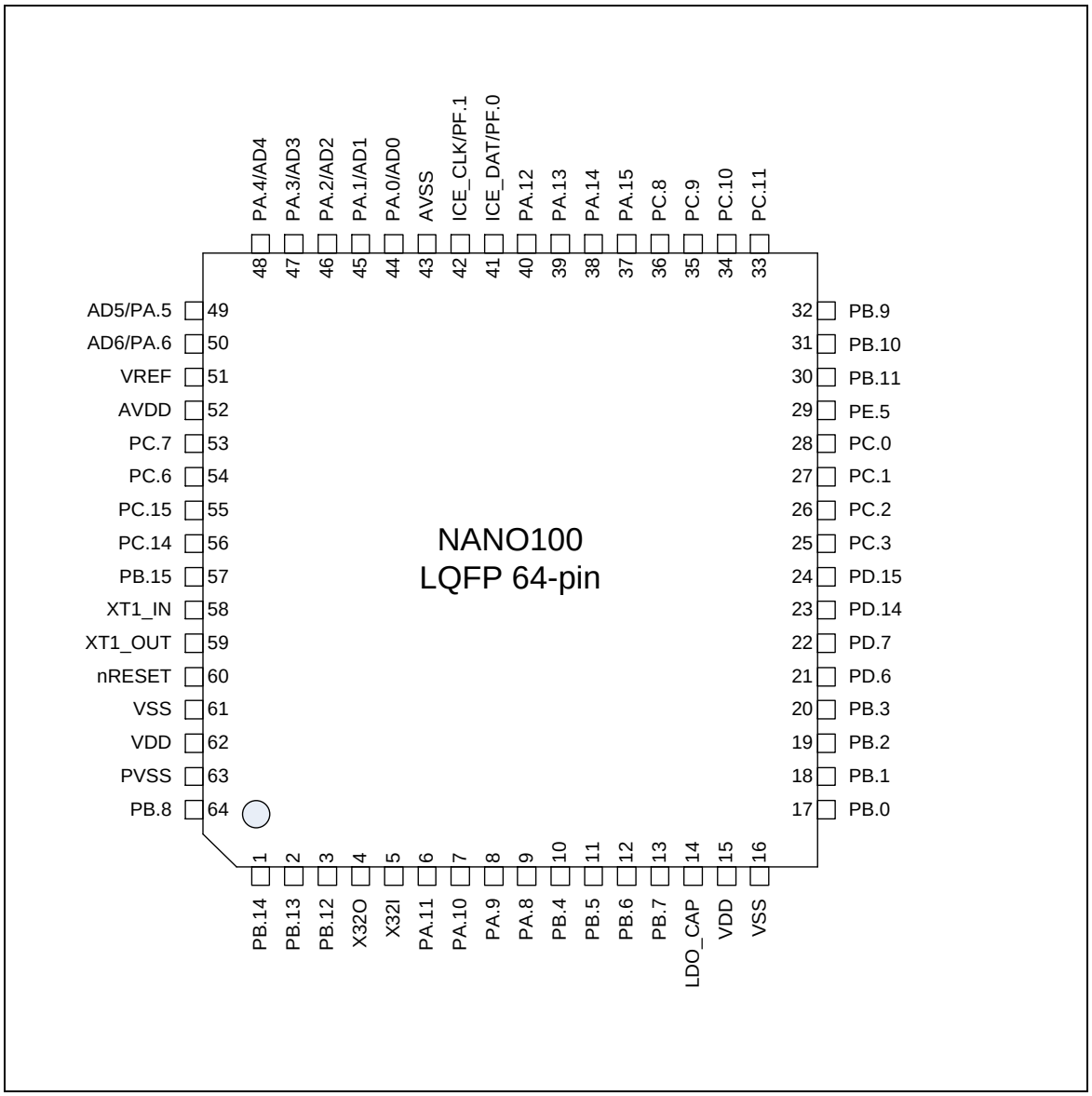



Figure 3-3 NuMicro™ Nano100 LQFP 64-pin Diagram

3.3.1.3 NuMicro™ Nano100 LQFP/QFN 48-pin

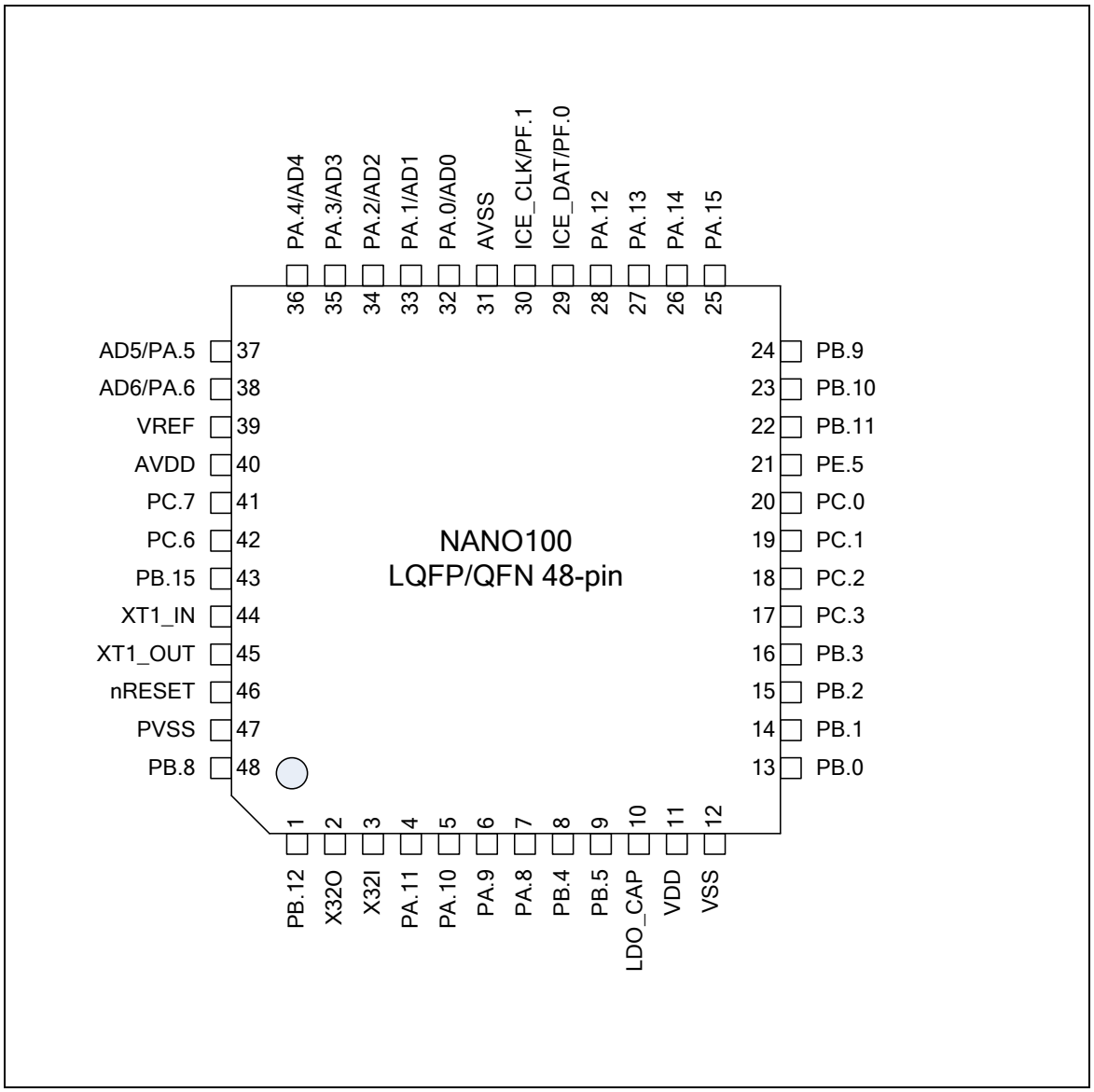



Figure 3-4 NuMicro™ Nano100 LQFP 48-pin Diagram

### 3.3.2 NuMicro™ Nano110 Pin Diagrams

#### 3.3.2.1 NuMicro™ Nano110 LQFP 128-pin

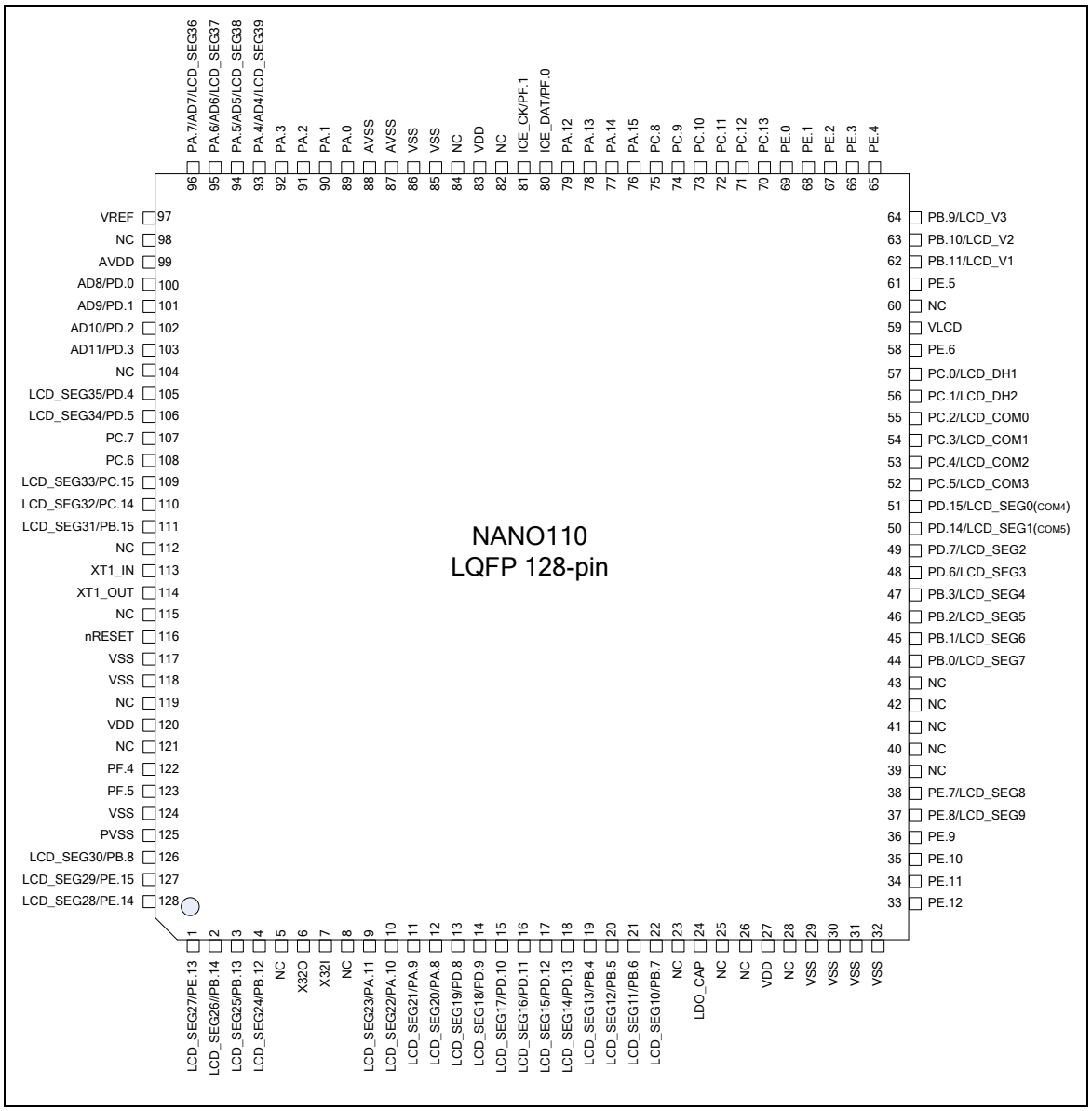



Figure 3-5 NuMicro™ Nano110 LQFP 128-pin Diagram

3.3.2.2 NuMicro™ Nano110 LQFP 64-pin

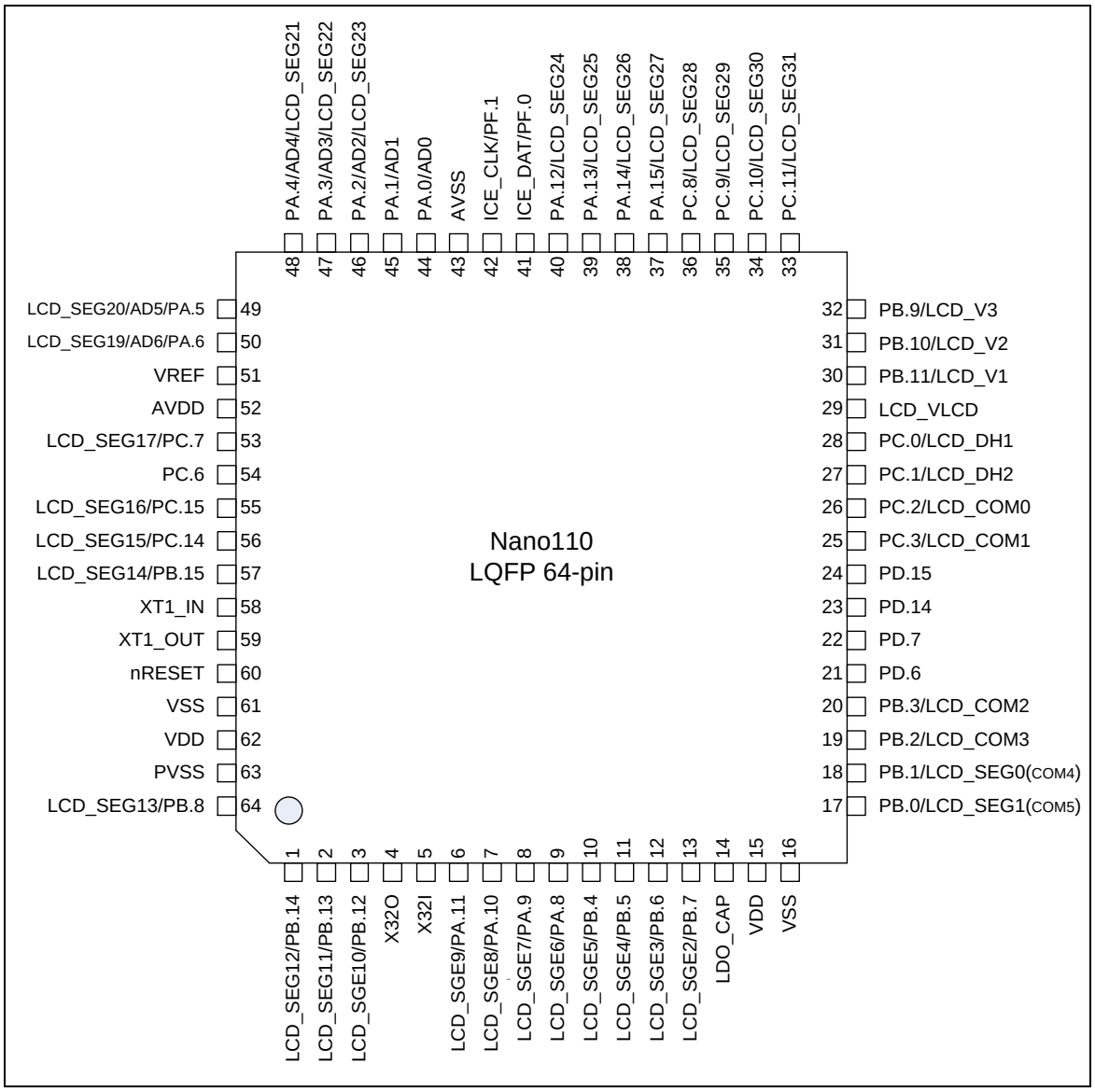



Figure 3-6 NuMicro™ Nano110 LQFP 64-pin Diagram

### 3.3.3 NuMicro™ Nano120 Pin Diagrams

#### 3.3.3.1 NuMicro™ Nano120 LQFP 128-pin

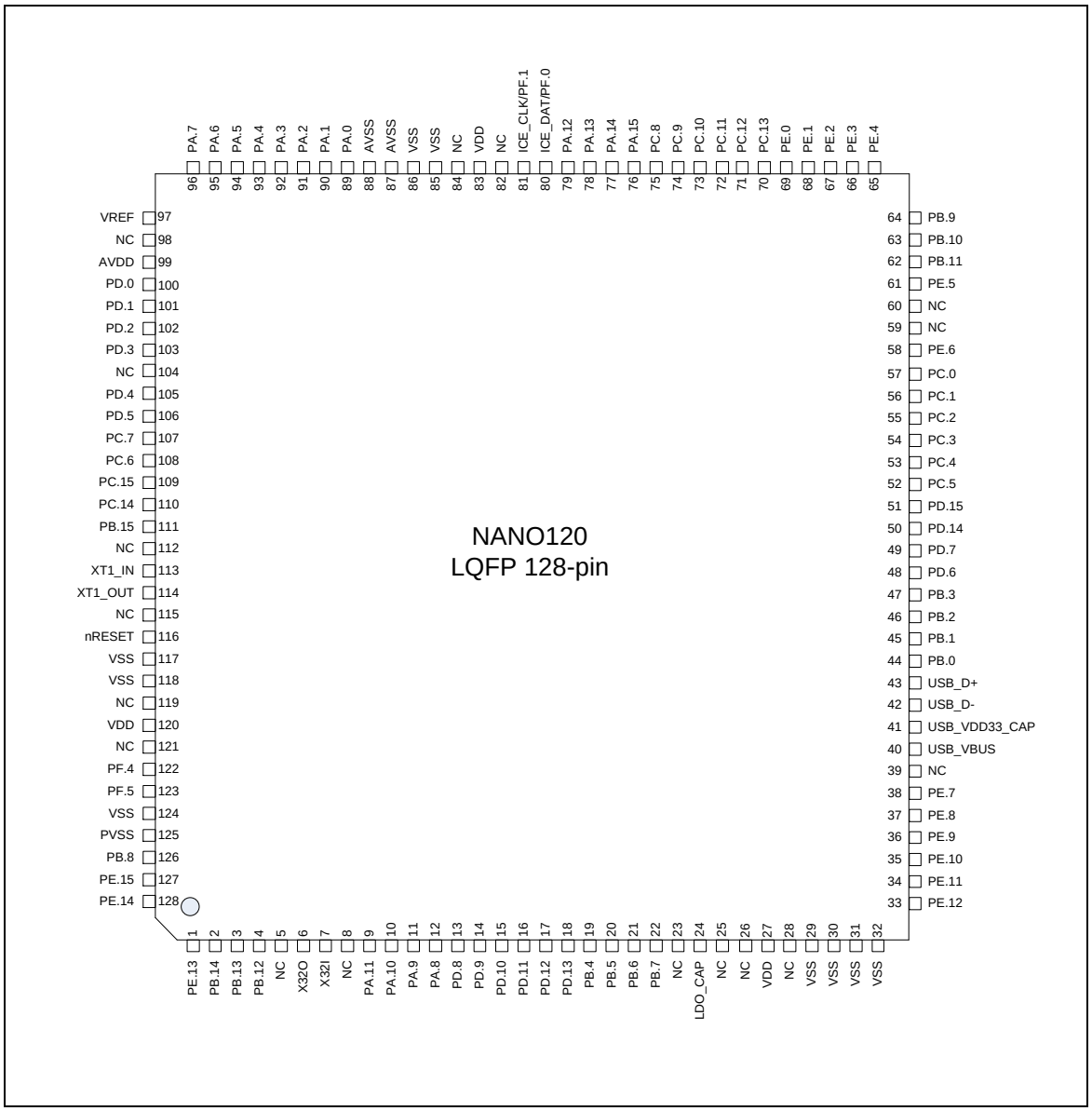



Figure 3-7 NuMicro™ Nano120 LQFP 128-pin Diagram

### 3.3.3.2 NuMicro™ Nano120 LQFP 64-pin

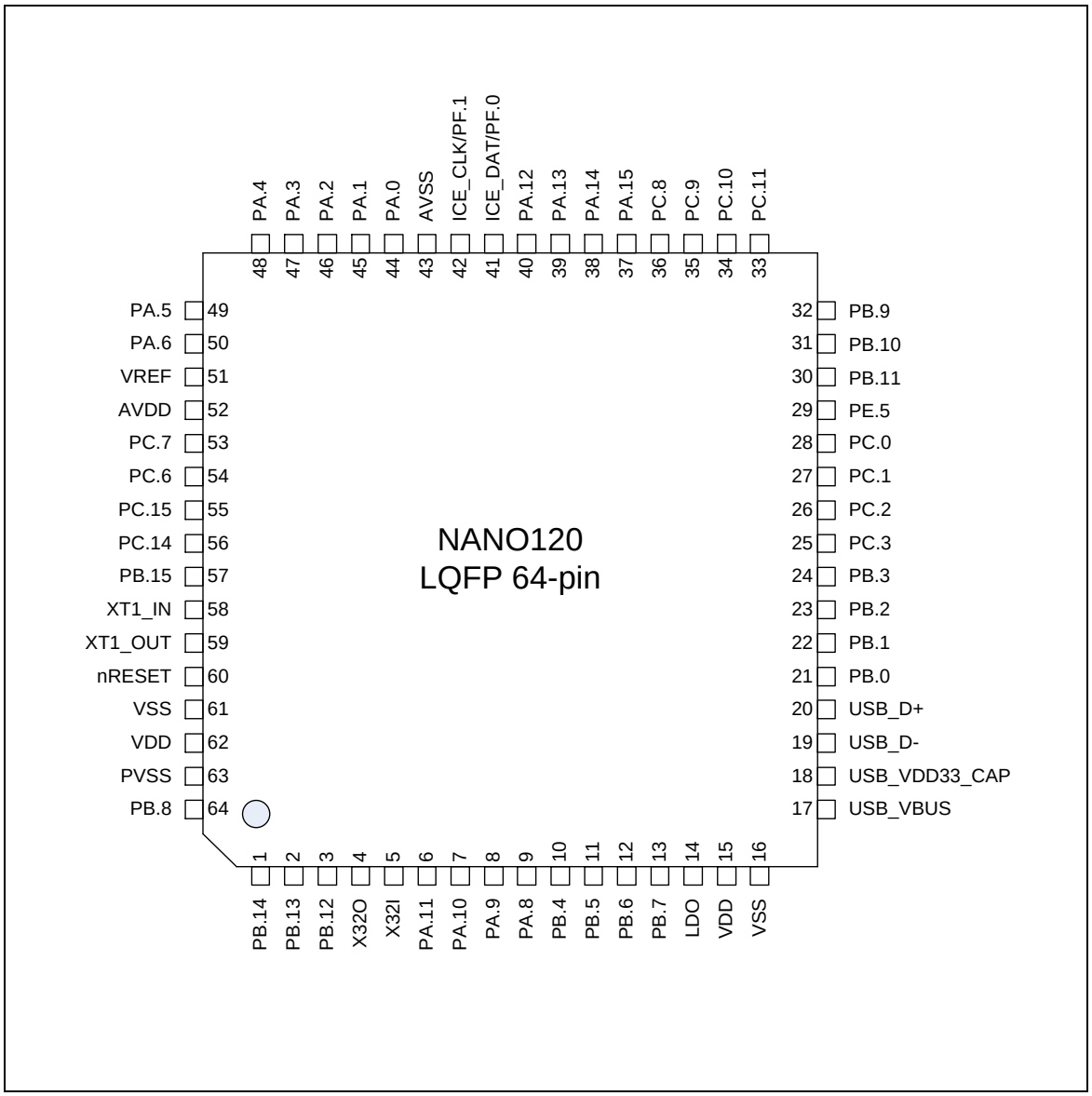



Figure 3-8 NuMicro™ Nano120 LQFP 64-pin Diagram

3.3.3.3 NuMicro™ Nano120 LQFP 48-pin

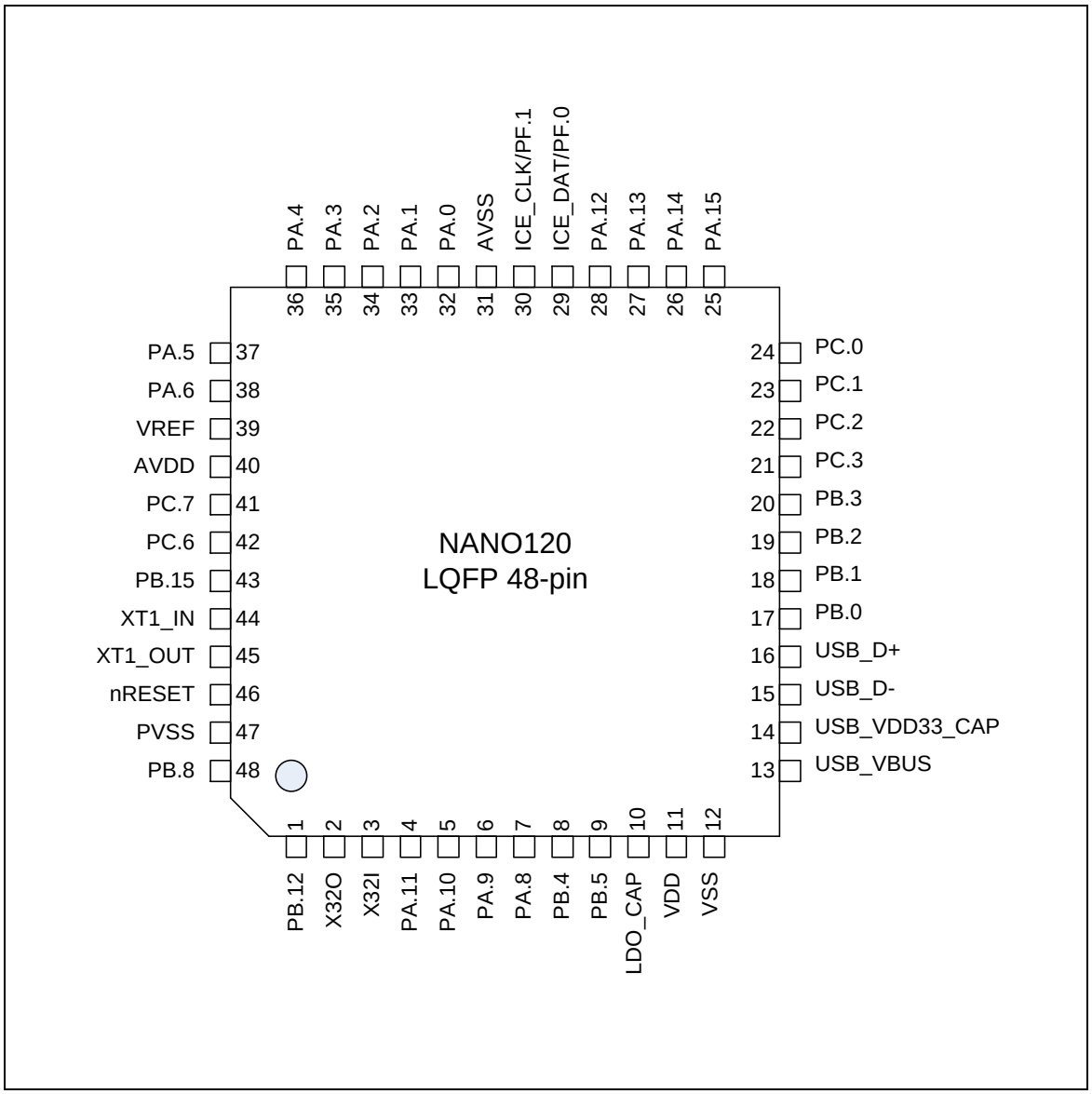



Figure 3-9 NuMicro™ Nano120 LQFP 48-pin Diagram

### 3.3.4 NuMicro™ Nano130 Pin Diagrams

#### 3.3.4.1 NuMicro™ Nano130 LQFP 128-pin

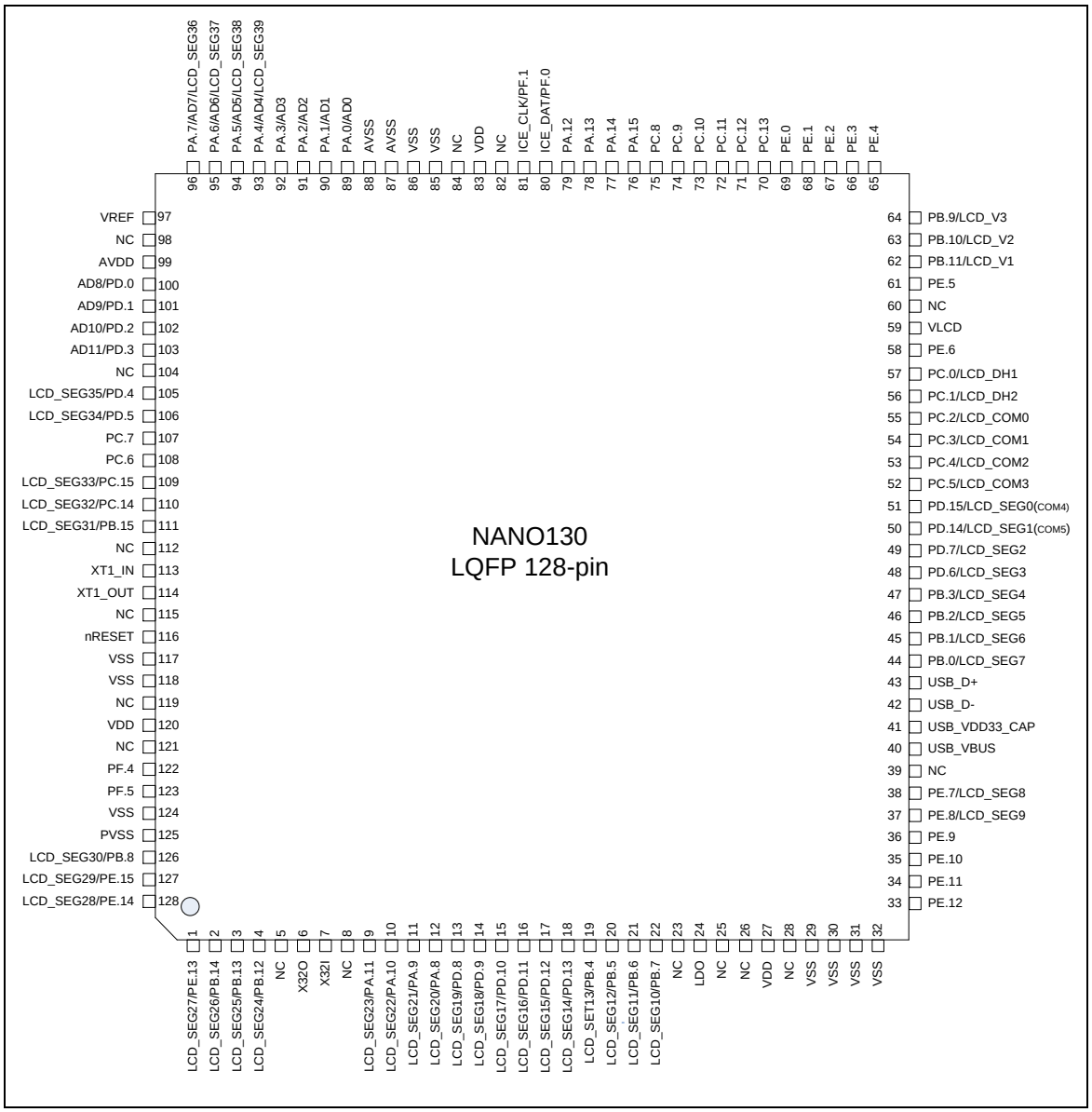



Figure 3-10 NuMicro™ Nano130 LQFP 128-pin Diagram

### 3.3.4.2 NuMicro™ Nano130 LQFP 64-pin

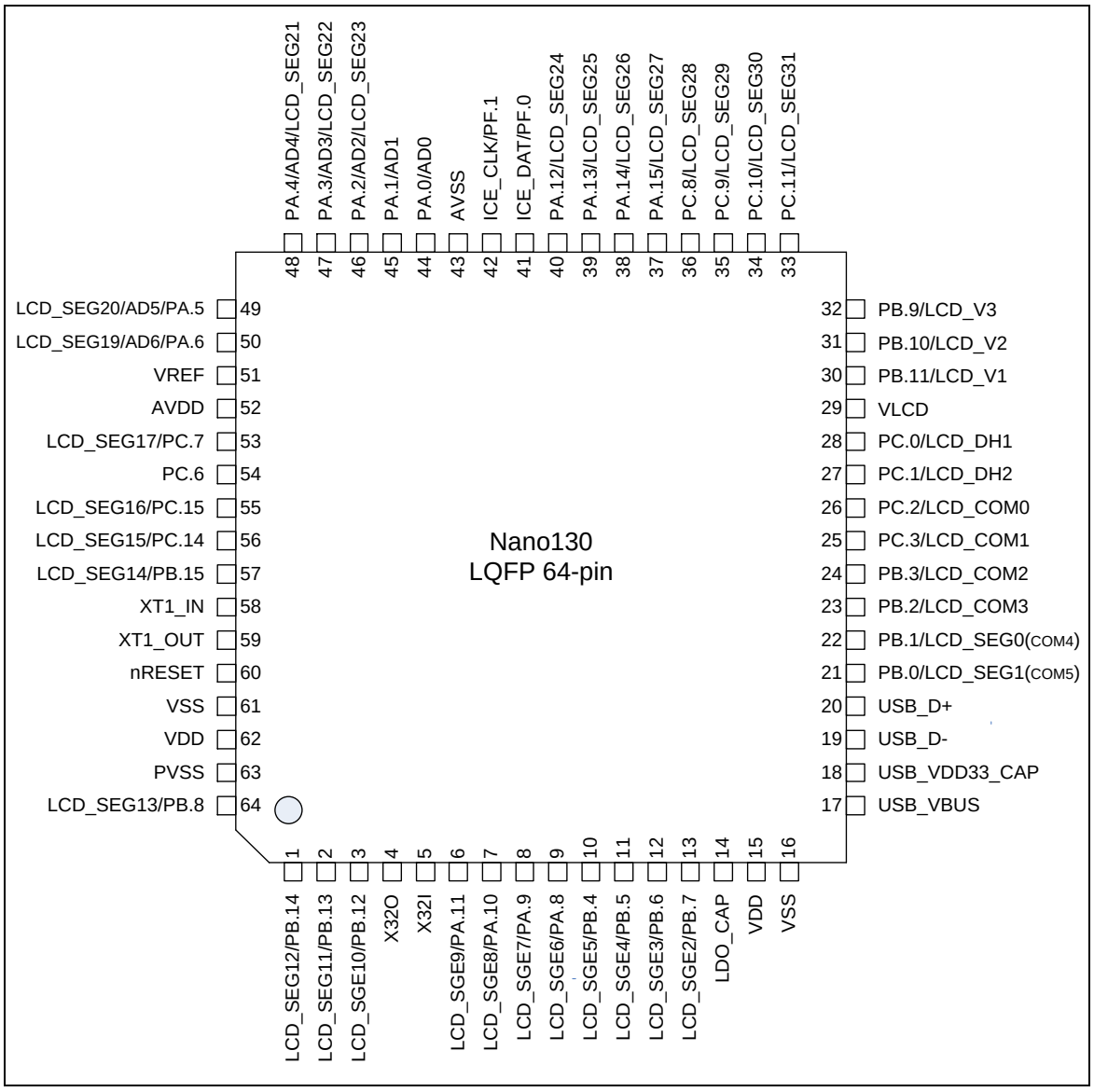



Figure 3-11 NuMicro™ Nano130 LQFP 64-pin Diagram

## 3.4 Pin Description

### 3.4.1 NuMicro™ Nano100 Pin Description

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP/QFN 48-pin | | | |
| 1 | | | PE.13 | **I/O** | General purpose digital I/O pin |
| 2 | 1 | | PB.14 | **I/O** | General purpose digital I/O pin |
| | | | INT0 | **I** | External interrupt0 input pin |
| | | | SC2_CD | **I** | SmartCard2 card detect pin |
| | | | SPI2_SS1 | **I/O** | SPI2 2nd slave select pin |
| 3 | 2 | | PB.13 | **I/O** | General purpose digital I/O pin |
| | | | EBI_AD1 | **I/O** | EBI Address/Data bus bit1 |
| 4 | 3 | 1 | PB.12 | **I/O** | General purpose digital I/O pin |
| | | | EBI_AD0 | **I/O** | EBI Address/Data bus bit0 |
| | | | FCLKO | **O** | Frequency Divider output pin |
| 5 | | | | | NC |
| 6 | 4 | 2 | X32O | **O** | External 32.768 kHz crystal output pin |
| 7 | 5 | 3 | X32I | **I** | External 32.768 kHz crystal input pin |
| 8 | | | | | NC |
| 9 | 6 | 4 | PA.11 | **I/O** | General purpose digital I/O pin |
| | | | I2C1_SCL | **I/O** | I2C1 clock pin |
| | | | EBI_nRD | **O** | EBI read enable output pin |
| | | | SC0_RST | **O** | SmartCard0 RST pin |
| | | | SPI2_MOSI0 | **I/O** | SPI2 1st MOSI (Master Out, Slave In) pin |
| 10 | 7 | 5 | PA.10 | **I/O** | General purpose digital I/O pin |
| | | | I2C1_SDA | **I/O** | I2C1 data I/O pin |
| | | | EBI_nWR | **O** | EBI write enable output pin |
| | | | SC0_PWR | **O** | SmartCard0 Power pin |
| | | | SPI2_MISO0 | **I/O** | SPI2 1st MISO (Master In, Slave Out) pin |
| 11 | 8 | 6 | PA.9 | **I/O** | General purpose digital I/O pin |
| | | | I2C0_SCL | **I/O** | I2C0 clock pin |
| | | | SC0_DAT | **I/O** | SmartCard0 DATA pin(SC0_UART_RXD) |
| | | | SPI2_CLK | **I/O** | SPI2 serial clock pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP/QFN 48-pin | | | |
| 12 | 9 | 7 | PA.8 | **I/O** | General purpose digital I/O pin |
| | | | I2C0_SDA | **I/O** | $I^2C0$ data I/O pin |
| | | | SC0_CLK | **O** | SmartCard0 clock pin(SC0_UART_TXD) |
| | | | SPI2_SS0 | **I/O** | SPI2 1st slave select pin |
| 13 | | | PD.8 | **I/O** | General purpose digital I/O pin |
| 14 | | | PD.9 | **I/O** | General purpose digital I/O pin |
| 15 | | | PD.10 | **I/O** | General purpose digital I/O pin |
| 16 | | | PD.11 | **I/O** | General purpose digital I/O pin |
| 17 | | | PD.12 | **I/O** | General purpose digital I/O pin |
| 18 | | | PD.13 | **I/O** | General purpose digital I/O pin |
| 19 | 10 | 8 | PB.4 | **I/O** | General purpose digital I/O pin |
| | | | UART1_RXD | **I** | UART1 Data receiver input pin |
| | | | SC0_CD | **I** | SmartCard0 card detect pin |
| | | | SPI2_SS0 | **I/O** | SPI2 1st slave select pin |
| 20 | 11 | 9 | PB.5 | **I/O** | General purpose digital I/O pin |
| | | | UART1_TXD | **O** | UART1 Data transmitter output pin |
| | | | SC0_RST | **O** | SmartCard0 RST pin |
| | | | SPI2_CLK | **I/O** | SPI2 serial clock pin |
| 21 | 12 | | PB.6 | **I/O** | General purpose digital I/O pin |
| | | | UART1_RTSn | **O** | UART1 Request to Send output pin |
| | | | EBI_ALE | **O** | EBI address latch enable output pin |
| | | | SPI2_MISO0 | **I/O** | SPI2 1st MISO (Master In, Slave Out) pin |
| 22 | 13 | | PB.7 | **I/O** | General purpose digital I/O pin |
| | | | UART1_CTSn | **I** | UART1 Clear to Send input pin |
| | | | EBI_nCS | **O** | EBI chip select enable output pin |
| | | | SPI2_MOSI0 | **I/O** | SPI2 1st MOSI (Master Out, Slave In) pin |
| 23 | | | | | NC |
| 24 | 14 | 10 | LDO_CAP | **P** | LDO output pin |
| 25 | | | | | NC |
| 26 | | | | | NC |
| 27 | 15 | 11 | VDD | **P** | Power supply for I/O ports and LDO source |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP/QFN 48-pin | | | |
| 28 | | | | | NC |
| 29 | 16 | 12 | VSS | **P** | Ground |
| 30 | | | VSS | **P** | Ground |
| 31 | | | VSS | **P** | Ground |
| 32 | | | VSS | **P** | Ground |
| 33 | | | PE.12 | **I/O** | General purpose digital I/O pin |
| 34 | | | PE.11 | **I/O** | General purpose digital I/O pin |
| 35 | | | PE.10 | **I/O** | General purpose digital I/O pin |
| 36 | | | PE.9 | **I/O** | General purpose digital I/O pin |
| 37 | | | PE.8 | **I/O** | General purpose digital I/O pin |
| 38 | | | PE.7 | **I/O** | General purpose digital I/O pin |
| 39 | | | | | NC |
| 40 | | | | | NC |
| 41 | | | | | NC |
| 42 | | | | | NC |
| 43 | | | | | NC |
| 44 | 17 | 13 | PB.0 | **I/O** | General purpose digital I/O pin |
| | | | UART0_RXD | **I** | UART0 Data receiver input pin |
| | | | SPI1_MOSI0 | **I/O** | SPI1 1st MOSI (Master Out, Slave In) pin |
| 45 | 18 | 14 | PB.1 | **I/O** | General purpose digital I/O pin |
| | | | UART0_TXD | **O** | UART0 Data transmitter output pin |
| | | | SPI1_MISO0 | **I/O** | SPI1 1st MISO (Master In, Slave Out) pin |
| 46 | 19 | 15 | PB.2 | **I/O** | General purpose digital I/O pin |
| | | | UART0_RTSn | **O** | UART0 Request to Send output pin |
| | | | EBI_nWRL | **O** | EBI low byte write enable output pin |
| | | | SPI1_CLK | **I/O** | SPI1 serial clock pin |
| 47 | 20 | 16 | PB.3 | **I/O** | General purpose digital I/O pin |
| | | | UART0_CTSn | **I** | UART0 Clear to Send input pin |
| | | | EBI_nWRH | **O** | EBI high byte write enable output pin |
| | | | SPI1_SS0 | **I/O** | SPI1 1st slave select pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP/QFN 48-pin | | | |
| 48 | 21 | | PD.6 | **I/O** | General purpose digital I/O pin |
| 49 | 22 | | PD.7 | **I/O** | General purpose digital I/O pin |
| 50 | 23 | | PD.14 | **I/O** | General purpose digital I/O pin |
| 51 | 24 | | PD.15 | **I/O** | General purpose digital I/O pin |
| 52 | | | PC.5 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_MOSI1 | **I/O** | SPI0 2nd MOSI (Master Out, Slave In) pin |
| 53 | | | PC.4 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_MISO1 | **I/O** | SPI0 2nd MISO (Master In, Slave Out) pin |
| 54 | 25 | 17 | PC.3 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_MOSI0 | **I/O** | SPI0 1st MOSI (Master Out, Slave In) pin |
| | | | I2S_DO | **O** | I2S data output |
| | | | SC1_RST | **O** | SmartCard1 RST pin |
| 55 | 26 | 18 | PC.2 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_MISO0 | **I/O** | SPI0 1st MISO (Master In, Slave Out) pin |
| | | | I2S_DI | **I** | I2S data input |
| | | | SC1_PWR | **O** | SmartCard1 PWR pin |
| 56 | 27 | 19 | PC.1 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_CLK | **I/O** | SPI0 serial clock pin |
| | | | I2S_BCLK | **I/O** | I2S bit clock pin |
| | | | SC1_DAT | **I/O** | SmartCard1 DATA pin(SC1_UART_RXD) |
| 57 | 28 | 20 | PC.0 / MCLKO | **I/O** | General purpose digital I/O pin / Module clock output pin |
| | | | SPI0_SS0 | **I/O** | SPI0 1st slave select pin |
| | | | I2S_LRCLK | **I/O** | I2S left right channel clock |
| | | | SC1_CLK | **O** | SmartCard1 clock pin(SC1_UART_TXD) |
| 58 | | | PE.6 | **I/O** | General purpose digital I/O pin |
| 59 | | | | | NC |
| 60 | | | | | NC |
| 61 | 29 | 21 | PE.5 | **I/O** | General purpose digital I/O pin |
| | | | PWM1_CH1 | **I/O** | PWM1 Channel1 output |
| 62 | 30 | 22 | PB.11 | **I/O** | General purpose digital I/O pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP/QFN 48-pin | | | |
| | | | PWM1_CH0 | **I/O** | PWM1 Channel0 output |
| | | | TM3 | **O** | Timer3 external counter input |
| | | | SC2_DAT | **I/O** | SmartCard2 DATA pin(SC2_UART_RXD) |
| | | | SPI0_MISO0 | **I/O** | SPI0 1st MISO (Master In, Slave Out) pin |
| 63 | 31 | 23 | PB.10 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_SS1 | **I/O** | SPI0 2nd slave select pin |
| | | | TM2 | **O** | Timer2 external counter input |
| | | | SC2_CLK | **O** | SmartCard2 clock pin(SC2_UART_TXD) |
| | | | SPI0_MOSI0 | **I/O** | SPI0 1st MOSI (Master Out, Slave In) pin |
| 64 | 32 | 24 | PB.9 | **I/O** | General purpose digital I/O pin |
| | | | SPI1_SS1 | **I/O** | SPI1 2nd slave select pin |
| | | | TM1 | **O** | Timer1 external counter input |
| | | | SC2_RST | **O** | SmartCard2 RST pin |
| | | | INT0 | **I** | External interrupt0 input pin |
| 65 | | | PE.4 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_MOSI0 | **I/O** | SPI0 1st MOSI (Master Out, Slave In) pin |
| 66 | | | PE.3 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_MISO0 | **I/O** | SPI0 1st MISO (Master In, Slave Out) pin |
| 67 | | | PE.2 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_CLK | **I/O** | SPI0 serial clock pin |
| 68 | | | PE.1 | **I/O** | General purpose digital I/O pin. |
| | | | PWM1_CH3 | **I/O** | PWM1 Channel3 output |
| | | | SPI0_SS0 | **I/O** | SPI0 1st slave select pin |
| 69 | | | PE.0 | **I/O** | General purpose digital I/O pin |
| | | | PWM1_CH2 | **I/O** | PWM1 Channel2 output |
| | | | I2S_MCLK | **O** | $I^2S$ master clock output pin |
| 70 | | | PC.13 | **I/O** | General purpose digital I/O pin |
| | | | SPI1_MOSI1 | **I/O** | SPI1 2nd MOSI (Master Out, Slave In) pin |
| | | | PWM1_CH1 | **O** | PWM1 Channel1 output |
| | | | SNOOPER | **I** | Snooper pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP/QFN 48-pin | | | |
| | | | INT1 | I | External interrupt 1 |
| | | | I2C0_SCL | O | $I^2C$0 clock pin |
| 71 | | | PC.12 | I/O | General purpose digital I/O pin |
| | | | SPI1_MISO1 | I/O | SPI1 2nd MISO (Master In, Slave Out) pin |
| | | | PWM1_CH0 | O | PWM1 Channel0 output |
| | | | INT0 | I | External interrupt0 input pin |
| | | | I2C0_SDA | I/O | $I^2C$0 data I/O pin |
| 72 | 33 | | PC.11 | I/O | General purpose digital I/O pin |
| | | | SPI1_MOSI0 | I/O | SPI1 1st MOSI (Master Out, Slave In) pin |
| | | | UART1_TXD | O | UART1 Data transmitter output pin |
| 73 | 34 | | PC.10 | I/O | General purpose digital I/O pin |
| | | | SPI1_MISO0 | I/O | SPI1 1st MISO (Master In, Slave Out) pin |
| | | | UART1_RXD | I | UART1 Data receiver input pin |
| 74 | 35 | | PC.9 | I/O | General purpose digital I/O pin |
| | | | SPI1_CLK | I/O | SPI1 serial clock pin |
| | | | I2C1_SCL | I/O | $I^2C$1 clock pin |
| 75 | 36 | | PC.8 | I/O | General purpose digital I/O pin |
| | | | SPI1_SS0 | I/O | SPI1 1st slave select pin |
| | | | EBI_MCLK | O | EBI external clock output pin |
| | | | I2C1_SDA | I/O | $I^2C$1 data I/O pin |
| 76 | 37 | 25 | PA.15 | I/O | General purpose digital I/O pin |
| | | | PWM0_CH3 | I/O | PWM0 Channel3 output |
| | | | I2S_MCLK | O | $I^2S$ master clock output pin |
| | | | TC3 | I | Timer3 capture input |
| | | | SC0_PWR | O | SmartCard0 Power pin |
| | | | UART0_TXD | O | UART0 Data transmitter output pin |
| 77 | 38 | 26 | PA.14 | I/O | General purpose digital I/O pin |
| | | | PWM0_CH2 | I/O | PWM0 Channel2 output |
| | | | EBI_AD15 | I/O | EBI Address/Data bus bit15 |
| | | | TC2 | I | Timer2 capture input |
| | | | UART0_RXD | I | UART0 Data receiver input pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP/QFN 48-pin | | | |
| 78 | 39 | 27 | PA.13 | **I/O** | General purpose digital I/O pin |
| | | | PWM0_CH1 | **I/O** | PWM0 Channel1 output |
| | | | EBI_AD14 | **I/O** | EBI Address/Data bus bit14 |
| | | | TC1 | **I** | Timer1 capture input |
| | | | I2C0_SCL | **I/O** | $I^2C0$ clock pin |
| 79 | 40 | 28 | PA.12 | **I/O** | General purpose digital I/O pin |
| | | | PWM0_CH0 | **I/O** | PWM0 Channel0 output |
| | | | EBI_AD13 | **I/O** | EBI Address/Data bus bit13 |
| | | | TC0 | **I** | Timer0 capture input |
| | | | I2C0_SDA | **I/O** | $I^2C0$ data I/O pin |
| 80 | 41 | 29 | ICE_DAT | **I/O** | Serial Wired Debugger Data pin |
| | | | PF.0 | **I/O** | General purpose digital I/O pin |
| | | | INT0 | **I** | External interrupt0 input pin |
| 81 | 42 | 30 | ICE_CLK | **I** | Serial Wired Debugger Clock pin |
| | | | PF.1 | **I/O** | General purpose digital I/O pin |
| | | | FCLKO | **O** | Frequency Divider output pin |
| | | | INT1 | **I** | External interrupt1 input pin |
| 82 | | | | | NC |
| 83 | | | VDD | **P** | Power supply for I/O ports and LDO source for internal PLL and digital circuit |
| 84 | | | | | NC |
| 85 | | | VSS | **P** | Ground |
| 86 | | | VSS | **P** | Ground |
| 87 | 43 | 31 | AVSS | **AP** | Ground Pin for analog circuit |
| 88 | | | AVSS | **AP** | Ground Pin for analog circuit |
| 89 | 44 | 32 | PA.0 | **I/O** | General purpose digital I/O pin |
| | | | AD0 | **AI** | ADC analog input0 |
| | | | SC2_CD | **I** | SmartCard2 card detect |
| 90 | 45 | 33 | PA.1 | **I/O** | General purpose digital I/O pin |
| | | | AD1 | **AI** | ADC analog input1 |
| | | | EBI_AD12 | **I/O** | EBI Address/Data bus bit12 |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP/QFN 48-pin | | | |
| 91 | 46 | 34 | PA.2 | **I/O** | General purpose digital I/O pin |
| | | | AD2 | **AI** | ADC analog input2 |
| | | | EBI_AD11 | **I/O** | EBI Address/Data bus bit11 |
| | | | UART1_RXD | **I** | UART1 Data receiver input pin |
| 92 | 47 | 35 | PA.3 | **I/O** | General purpose digital I/O pin |
| | | | AD3 | **AI** | ADC analog input3 |
| | | | EBI_AD10 | **I/O** | EBI Address/Data bus bit10 |
| | | | UART1_TXD | **O** | UART1 Data transmitter output pin |
| 93 | 48 | 36 | PA.4 | **I/O** | General purpose digital I/O pin |
| | | | AD4 | **AI** | ADC analog input4 |
| | | | EBI_AD9 | **I/O** | EBI Address/Data bus bit9 |
| | | | SC2_PWR | **O** | SmartCard2 Power pin |
| | | | I2C0_SDA | **I/O** | $I^2C0$ data I/O pin |
| 94 | 49 | 37 | PA.5 | **I/O** | General purpose digital I/O pin |
| | | | AD5 | **AI** | ADC analog input5 |
| | | | EBI_AD8 | **I/O** | EBI Address/Data bus bit8 |
| | | | SC2_RST | **O** | SmartCard2 RST pin |
| | | | I2C0_SCL | **I/O** | $I^2C0$ clock pin |
| 95 | 50 | 38 | PA.6 | **I/O** | General purpose digital I/O pin |
| | | | AD6 | **AI** | ADC analog input6 |
| | | | EBI_AD7 | **I/O** | EBI Address/Data bus bit7 |
| | | | TC3 | **I** | Timer3 capture input |
| | | | SC2_CLK | **O** | SmartCard2 clock pin(SC2_UART_TXD) |
| | | | PWM0_CH3 | **O** | PWM0 Channel3 output |
| 96 | | | PA.7 | **I/O** | General purpose digital I/O pin |
| | | | AD7 | **AI** | ADC analog input7 |
| | | | EBI_AD6 | **I/O** | EBI Address/Data bus bit6 |
| | | | TC2 | **I** | Timer2 capture input |
| | | | SC2_DAT | **I/O** | SmartCard2 DATA pin(SC2_UART_RXD) |
| | | | PWM0_CH2 | **O** | PWM0 Channel2 output |
| 97 | 51 | 39 | VREF | **AP** | Voltage reference input for ADC |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP/QFN 48-pin | | | |
| 98 | | | | | NC |
| 99 | 52 | 40 | AVDD | **AP** | Power supply for internal analog circuit |
| 100 | | | PD.0 | **I/O** | General purpose digital I/O pin |
| | | | UART1_RXD | **I** | UART1 Data receiver input pin |
| | | | SPI2_SS0 | **I/O** | SPI2 1st slave select pin |
| | | | SC1_CLK | **O** | SmartCard1 clock pin(SC1_UART_TXD) |
| | | | AD8 | **AI** | ADC analog input8 |
| 101 | | | PD.1 | **I/O** | General purpose digital I/O pin |
| | | | UART1_TXD | **O** | UART1 Data transmitter output pin |
| | | | SPI2_CLK | **I/O** | SPI2 serial clock pin |
| | | | SC1_DAT | **I/O** | SmartCard1 DATA pin(SC1_UART_RXD). |
| | | | AD9 | **AI** | ADC analog input9 |
| 102 | | | PD.2 | **I/O** | General purpose digital I/O pin |
| | | | UART1_RTSn | **O** | UART1 Request to Send output pin |
| | | | I2S_LRCLK | **I/O** | $I^2S$ left right channel clock |
| | | | SPI2_MISO0 | **I/O** | SPI2 1st MISO (Master In, Slave Out) pin |
| | | | SC1_PWR | **O** | SmartCard1 Power pin |
| | | | AD10 | **AI** | ADC analog input10 |
| 103 | | | PD.3 | **I/O** | General purpose digital I/O pin |
| | | | UART1_CTSn | **I** | UART1 Clear to Send input pin |
| | | | I2S_BCLK | **I/O** | $I^2S$ bit clock pin |
| | | | SPI2_MOSI0 | **I/O** | SPI2 1st MOSI (Master Out, Slave In) pin |
| | | | SC1_RST | **O** | SmartCard1 RST pin |
| | | | AD11 | **AI** | ADC analog input11 |
| 104 | | | | | NC |
| 105 | | | PD.4 | **I/O** | General purpose digital I/O pin |
| | | | I2S_DI | **I** | $I^2S$ data input |
| | | | SPI2_MISO1 | **I/O** | SPI2 2nd MISO (Master In, Slave Out) pin |
| | | | SC1_CD | **I** | SmartCard1 card detect |
| 106 | | | PD.5 | **I/O** | General purpose digital I/O pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP/QFN 48-pin | | | |
| | | | I2S_DO | **O** | $I^2S$ data output |
| | | | SPI2_MOSI1 | **I/O** | SPI2 2nd MOSI (Master Out, Slave In) pin |
| 107 | 53 | 41 | PC.7 | **I/O** | General purpose digital I/O pin |
| | | | DA1_OUT | **AO** | DAC 1 output |
| | | | EBI_AD5 | **I/O** | EBI Address/Data bus bit5 |
| | | | TC1 | **I** | Timer1 capture input |
| | | | PWM0_CH1 | **O** | PWM1 Channel1 output |
| 108 | 54 | 42 | PC.6 | **I/O** | General purpose digital I/O pin |
| | | | DA0_OUT | **I** | DAC0 output |
| | | | EBI_AD4 | **I/O** | EBI Address/Data bus bit4 |
| | | | TC0 | **I** | Timer0 capture input |
| | | | SC1_CD | **I** | SmartCard1 card detect pin |
| | | | PWM0_CH0 | **O** | PWM0 Channel0 output |
| 109 | 55 | | PC.15 | **I/O** | General purpose digital I/O pin |
| | | | EBI_AD3 | **I/O** | EBI Address/Data bus bit3 |
| | | | TC0 | **I** | Timer0 capture input |
| | | | PWM1_CH2 | **O** | PWM1 Channel1 output |
| 110 | 56 | | PC.14 | **I/O** | General purpose digital I/O pin |
| | | | EBI_AD2 | **I/O** | EBI Address/Data bus bit2 |
| | | | PWM1_CH3 | **I/O** | PWM1 Channel3 output |
| 111 | 57 | 43 | PB.15 | **I/O** | General purpose digital I/O pin |
| | | | INT1 | **I** | External interrupt1 input pin |
| | | | SNOOPER | **I** | Snooper pin |
| | | | SC1_CD | **I** | SmartCard1 card detect |
| 112 | | | | | NC |
| 113 | 58 | 44 | XT1_IN | **O** | External 4~24 MHz crystal output pin |
| | | | PF.3 | **I/O** | General purpose digital I/O pin |
| 114 | 59 | 45 | XT1_OUT | **I** | External 4~24 MHz crystal input pin |
| | | | PF.2 | **I/O** | General purpose digital I/O pin |
| 115 | | | | | NC |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP/QFN 48-pin | | | |
| 116 | 60 | 46 | nRESET | **I** | External reset input: Low active, set this pin low reset chip to initial state. With internal pull-up. |
| 117 | 61 | | VSS | **P** | Ground |
| 118 | | | VSS | **P** | Ground |
| 119 | | | | | NC |
| 120 | 62 | | VDD | **P** | Power supply for I/O ports and LDO source for internal PLL and digital circuit |
| 121 | | | | | NC |
| 122 | | | PF.4 | **I/O** | General purpose digital I/O pin |
| | | | I2C0_SDA | **I/O** | $I^2C0$ data I/O pin |
| 123 | | | PF.5 | **I/O** | General purpose digital I/O pin |
| | | | I2C0_SCL | **I/O** | $I^2C0$ clock pin |
| 124 | | | VSS | **P** | Ground |
| 125 | 63 | 47 | PVSS | **P** | PLL Ground |
| 126 | 64 | 48 | PB.8 | **I/O** | General purpose digital I/O pin |
| | | | STADC | **I** | ADC external trigger input. |
| | | | TM0 | **I** | Timer0 external counter input |
| | | | INT0 | **I** | External interrupt0 input pin |
| | | | SC2_PWR | **O** | SmartCard2 Power pin |
| 127 | | | PE.15 | **I/O** | General purpose digital I/O pin |
| 128 | | | PE.14 | **I/O** | General purpose digital I/O pin |

**Note:**

Pin Type: I = Digital Input, O = Digital Output; AI = Analog Input; AO = Analog Output; P = Power Pin; AP = Analog Power.

### 3.4.2 NuMicro™ Nano110 Pin Description

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| 1 | | | PE.13 | I/O | General purpose digital I/O pin |
| | | | LCD_SEG27 | O | LCD segment output 27 at LQFP128 |
| 2 | 1 | | PB.14 | I/O | General purpose digital I/O pin |
| | | | INT0 | I | External interrupt0 input pin |
| | | | SC2_CD | I | SmartCard2 card detect |
| | | | SPI2_SS1 | I/O | SPI2 2nd slave select pin |
| | | | LCD_SEG12 | O | LCD segment output 12 at LQFP64 |
| | | | LCD_SEG26 | O | LCD segment output 26 at LQFP128 |
| 3 | 2 | | PB.13 | I/O | General purpose digital I/O pin |
| | | | EBI_AD1 | I/O | EBI Address/Data bus bit1 |
| | | | LCD_SEG11 | O | LCD segment output 11 at LQFP64 |
| | | | LCD_SEG25 | O | LCD segment output 25 at LQFP128 |
| 4 | 3 | | PB.12 | I/O | General purpose digital I/O pin |
| | | | EBI_AD0 | I/O | EBI Address/Data bus bit0 |
| | | | FCLKO | O | Frequency Divider output pin |
| | | | LCD_SEG10 | O | LCD segment output 10 at LQFP64 |
| | | | LCD_SEG24 | O | LCD segment output 24 at LQFP128 |
| 5 | | | | | NC |
| 6 | 4 | | X32O | O | External 32.768 kHz crystal output pin |
| 7 | 5 | | X32I | I | External 32.768 kHz crystal input pin |
| 8 | | | | | NC |
| 9 | 6 | | PA.11 | I/O | General purpose digital I/O pin |
| | | | I2C1_SCL | I/O | I2C1 clock pin |
| | | | EBI_nRD | O | EBI read enable output pin |
| | | | SC0_RST | O | SmartCard0 RST pin |
| | | | SPI2_MOSI0 | I/O | SPI2 1st MOSI (Master Out, Slave In) pin |
| | | | LCD_SEG9 | O | LCD segment output 9 at LQFP64 |
| | | | LCD_SEG23 | O | LCD segment output 23 at LQFP128 |
| 10 | 7 | | PA.10 | I/O | General purpose digital I/O pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | I2C1_SDA | **I/O** | $I^2C1$ data I/O pin |
| | | | EBI_nWR | **O** | EBI write enable output pin |
| | | | SC0_PWR | **O** | SmartCard0 Power pin |
| | | | SPI2_MISO0 | **I/O** | SPI2 1st MISO (Master In, Slave Out) pin |
| | | | LCD_SEG8 | **O** | LCD segment output 8 at LQFP64 |
| | | | LCD_SEG22 | **O** | LCD segment output 22 at LQFP128 |
| 11 | 8 | | PA.9 | **I/O** | General purpose digital I/O pin |
| | | | I2C0_SCL | **I/O** | $I^2C0$ clock pin |
| | | | SC0_DAT | **I/O** | SmartCard0 DATA pin(SC0_UART_RXD) |
| | | | SPI2_CLK | **I/O** | SPI2 serial clock pin |
| | | | LCD_SEG7 | **O** | LCD segment output 7 at LQFP64 |
| | | | LCD_SEG21 | **O** | LCD segment output 21 at LQFP128 |
| 12 | 9 | | PA.8 | **I/O** | General purpose digital I/O pin |
| | | | I2C0_SDA | **I/O** | $I^2C0$ data I/O pin |
| | | | SC0_CLK | **O** | SmartCard0 clock pin(SC0_UART_TXD) |
| | | | SPI2_SS0 | **I/O** | SPI2 1st slave select pin |
| | | | LCD_SEG6 | **O** | LCD segment output 6 at LQFP64 |
| | | | LCD_SEG20 | **O** | LCD segment output 20 at LQFP128 |
| 13 | | | PD.8 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG19 | **O** | LCD segment output 19 at LQFP128 |
| 14 | | | PD.9 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG18 | **O** | LCD segment output 18 at LQFP128 |
| 15 | | | PD.10 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG17 | **O** | LCD segment output 17 at LQFP128 |
| 16 | | | PD.11 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG16 | **O** | LCD segment output 16 at LQFP128 |
| 17 | | | PD.12 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG15 | **O** | LCD segment output 15 at LQFP128 |
| 18 | | | PD.13 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG14 | **O** | LCD segment output 14 at LQFP128 |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| 19 | 10 | | PB.4 | I/O | General purpose digital I/O pin |
| | | | UART1_RXD | I | UART1 Data receiver input pin |
| | | | SC0_CD | I | SmartCard0 card detect pin |
| | | | SPI2_SS0 | I/O | SPI2 1st slave select pin |
| | | | LCD_SEG5 | O | LCD segment output 5 at LQFP64 |
| | | | LCD_SEG13 | O | LCD segment output 13 at LQFP128 |
| 20 | 11 | | PB.5 | I/O | General purpose digital I/O pin |
| | | | UART1_TXD | O | UART1 Data transmitter output pin |
| | | | SC0_RST | O | SmartCard0 RST pin |
| | | | SPI2_CLK | I/O | SPI2 serial clock pin |
| | | | LCD_SEG4 | O | LCD segment output 4 at LQFP64 |
| | | | LCD_SEG12 | O | LCD segment output 12 at LQFP128 |
| 21 | 12 | | PB.6 | I/O | General purpose digital I/O pin |
| | | | UART1_RTSn | O | UART1 Request to Send output pin |
| | | | EBI_ALE | O | EBI address latch enable output pin |
| | | | SPI2_MISO0 | I/O | SPI2 1st MISO (Master In, Slave Out) pin |
| | | | LCD_SEG3 | O | LCD segment output 3 at LQFP64 |
| | | | LCD_SEG11 | O | LCD segment output 11 at LQFP128 |
| 22 | 13 | | PB.7 | I/O | General purpose digital I/O pin |
| | | | UART1_CTSn | I | UART1 Clear to Send input pin |
| | | | EBI_nCS | O | EBI chip select enable output pin |
| | | | SPI2_MOSI0 | I/O | SPI2 1st MOSI (Master Out, Slave In) pin |
| | | | LCD_SEG2 | O | LCD segment output 2 at LQFP64 |
| | | | LCD_SEG10 | O | LCD segment output 10 at LQFP128 |
| 23 | | | | | NC |
| 24 | 14 | | LDO_CAP | P | LDO output pin |
| 25 | | | | | NC |
| 26 | | | | | NC |
| 27 | 15 | | VDD | P | Power supply for I/O ports and LDO source |
| 28 | | | | | NC |
| 29 | 16 | | VSS | P | Ground |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| 30 | | | VSS | **P** | Ground |
| 31 | | | VSS | **P** | Ground |
| 32 | | | VSS | **P** | Ground |
| 33 | | | PE.12 | **I/O** | General purpose digital I/O pin |
| | | | UART1_CTSn | **I** | UART1 Clear to Send input pin |
| 34 | | | PE.11 | **I/O** | General purpose digital I/O pin |
| | | | UART1_RTSn | **O** | UART1 Request to Send output pin |
| 35 | | | PE.10 | **I/O** | General purpose digital I/O pin |
| | | | UART1_TXD | **O** | UART1 Data transmitter output pin |
| 36 | | | PE.9 | **I/O** | General purpose digital I/O pin |
| | | | UART1_RXD | **I** | UART1 Data receiver input pin |
| 37 | | | PE.8 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG9 | **O** | LCD segment output 9 at LQFP128 |
| 38 | | | PE.7 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG8 | **O** | LCD segment output 8 at LQFP128 |
| 39 | | | | | NC |
| 40 | | | | | NC |
| 41 | | | | | NC |
| 42 | | | | | NC |
| 43 | | | | | NC |
| 44 | 17 | | PB.0 | **I/O** | General purpose digital I/O pin |
| | | | UART0_RXD | **I** | UART0 Data receiver input pin |
| | | | SPI1_MOSI0 | **I/O** | SPI1 1st MOSI (Master Out, Slave In) pin |
| | | | LCD_SEG1 | **O** | LCD segment output 1 at LQFP64 (or as LD_COM5) |
| | | | LCD_SEG7 | **O** | LCD segment output 7 at LQFP128 |
| 45 | 18 | | PB.1 | **I/O** | General purpose digital I/O pin |
| | | | UART0_TXD | **O** | UART0 Data transmitter output pin |
| | | | SPI1_MISO0 | **I/O** | SPI1 1st MISO (Master In, Slave Out) pin |
| | | | LCD_SEG0 | **O** | LCD segment output 0 at LQFP64 (or as LCD_COM4) |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | LCD_SEG6 | **O** | LCD segment output 6 at LQFP128 |
| 46 | 19 | | PB.2 | **I/O** | General purpose digital I/O pin |
| | | | UART0_RTSn | **O** | UART0 Request to Send output pin |
| | | | EBI_nWRL | **O** | EBI low byte write enable output pin |
| | | | SPI1_CLK | **I/O** | SPI1 serial clock pin |
| | | | LCD_COM3 | **O** | LCD common output 3 at LQFP64 |
| | | | LCD_SEG5 | **O** | LCD segment output 5 at LQFP128 |
| 47 | 20 | | PB.3 | **I/O** | General purpose digital I/O pin |
| | | | UART0_CTSn | **I** | UART0 Clear to Send input pin |
| | | | EBI_nWRH | **O** | EBI high byte write enable output pin |
| | | | SPI1_SS0 | **I/O** | SPI1 1st slave select pin |
| | | | LCD_COM2 | **O** | LCD common output 2 at LQFP64 |
| | | | LCD_SEG4 | **O** | LCD segment output 4 at LQFP128 |
| 48 | 21 | | PD.6 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG3 | **O** | LCD segment output 3 at LQFP128 |
| 49 | 22 | | PD.7 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG2 | **O** | LCD segment output 2 at LQFP128 |
| 50 | 23 | | PD.14 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG1 | **O** | LCD segment output 1 at LQFP128 (or as LCD_COM5) |
| 51 | 24 | | PD.15 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG0 | **O** | LCD segment output 0 at LQFP128 (or as LCD_COM4) |
| 52 | | | PC.5 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_MOSI1 | **I/O** | SPI0 2nd MOSI (Master Out, Slave In) pin |
| | | | LCD_COM3 | **O** | LCD common output 3 at LQFP128 |
| 53 | | | PC.4 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_MISO1 | **I/O** | SPI0 2nd MISO (Master In, Slave Out) pin |
| | | | LCD_COM2 | **O** | LCD common output 2 at LQFP128 |
| 54 | 25 | | PC.3 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_MOSI0 | **I/O** | SPI0 1st MOSI (Master Out, Slave In) pin |
| | | | I2S_DO | **O** | $I^2S$ data output |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | SC1_RST | **O** | SmartCard1 RST pin |
| | | | LCD_COM1 | **O** | LCD common output 1 at LQFP64 |
| | | | LCD_COM1 | **O** | LCD common output 1 at LQFP128 |
| 55 | 26 | | PC.2 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_MISO0 | **I/O** | SPI0 1st MISO (Master In, Slave Out) pin |
| | | | I2S_DI | **I** | $I^2S$ data input |
| | | | SC1_PWR | **O** | SmartCard1 PWR pin |
| | | | LCD_COM0 | **O** | LCD common output 0 at LQFP64 |
| | | | LCD_COM0 | **O** | LCD common output 0 at LQFP128 |
| 56 | 27 | | PC.1 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_CLK | **I/O** | SPI0 serial clock pin |
| | | | I2S_BCLK | **I/O** | $I^2S$ bit clock pin |
| | | | SC1_DAT | **I/O** | SmartCard1 DATA pin(SC1_UART_RXD) |
| | | | LCD_DH2 | **O** | LCD externl capacitor pin of charge pump circuit at LQFP64 |
| | | | LCD_DH2 | **O** | LCD externl capacitor pin of charge pump circuit at LQFP128 |
| 57 | 28 | | PC.0 / MCLKO | **I/O** | General purpose digital I/O pin / Module clock output pin |
| | | | SPI0_SS0 | **I/O** | SPI0 1st slave select pin |
| | | | I2S_LRCLK | **I/O** | $I^2S$ left right channel clock |
| | | | SC1_CLK | **O** | SmartCard1 clock pin(SC1_UART_TXD) |
| | | | LCD_DH1 | **O** | LCD externl capacitor pin of charge pump circuit at LQFP64 |
| | | | LCD_DH1 | **O** | LCD externl capacitor pin of charge pump circuit at LQFP128 |
| 58 | | | PE.6 | **I/O** | General purpose digital I/O pin |
| 59 | 29 | | LCD_VLCD | **AO** | LCD power supply pin |
| 60 | | | | | NC |
| 61 | | | PE.5 | **I/O** | General purpose digital I/O pin |
| 62 | 30 | | PB.11 | **I/O** | General purpose digital I/O pin |
| | | | PWM1_CH0 | **I/O** | PWM1 Channel0 output |
| | | | TM3 | **O** | Timer3 external counter input |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | SC2_DAT | I/O | SmartCard2 DATA pin(SC2_UART_RXD) |
| | | | SPI0_MISO0 | I/O | SPI0 1st MISO (Master In, Slave Out) pin |
| | | | LCD_V1 | O | Unit voltage for LCD charge pump circuit at LQFP64 |
| | | | LCD_V1 | O | LCD Unit voltage for LCD charge pump circuit at LQFP128 |
| 63 | 31 | | PB.10 | I/O | General purpose digital I/O pin |
| | | | SPI0_SS1 | I/O | SPI0 2nd slave select pin |
| | | | TM2 | O | Timer2 external counter input |
| | | | SC2_CLK | O | SmartCard2 clock pin(SC2_UART_TXD) |
| | | | SPI0_MOSI0 | I/O | SPI0 1st MOSI (Master Out, Slave In) pin |
| | | | LCD_V2 | O | LCD driver biasing voltage at LQFP64 |
| | | | LCD_V2 | O | LCD driver biasing voltage at LQFP128 |
| 64 | 32 | | PB.9 | I/O | General purpose digital I/O pin |
| | | | SPI1_SS1 | I/O | SPI1 2nd slave select pin |
| | | | TM1 | O | Timer1 external counter input |
| | | | SC2_RST | O | SmartCard2 RST pin |
| | | | INT0 | I | External interrupt0 input pin |
| | | | LCD_V3 | O | LCD driver biasing voltage at LQFP64 |
| | | | LCD_V3 | O | LCD driver biasing voltage at LQFP128 |
| 65 | | | PE.4 | I/O | General purpose digital I/O pin |
| | | | SPI0_MOSI0 | I/O | SPI0 1st MOSI (Master Out, Slave In) pin |
| 66 | | | PE.3 | I/O | General purpose digital I/O pin |
| | | | SPI0_MISO0 | I/O | SPI0 1st MISO (Master In, Slave Out) pin |
| 67 | | | PE.2 | I/O | General purpose digital I/O pin |
| | | | SPI0_CLK | I/O | SPI0 serial clock pin |
| 68 | | | PE.1 | I/O | General purpose digital I/O pin |
| | | | PWM1_CH3 | I/O | PWM1 Channel3 output |
| | | | SPI0_SS0 | I/O | SPI0 1st slave select pin |
| 69 | | | PE.0 | I/O | General purpose digital I/O pin |
| | | | PWM1_CH2 | I/O | PWM1 Channel2 output |
| | | | I2S_MCLK | O | I2S master clock output pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| 70 | | | PC.13 | **I/O** | General purpose digital I/O pin |
| | | | SPI1_MOSI1 | **I/O** | SPI1 2nd MOSI (Master Out, Slave In) pin |
| | | | PWM1_CH1 | **O** | PWM1 Channel1 output |
| | | | SNOOPER | **I** | Snooper pin |
| | | | INT1 | **I** | External interrupt 1 |
| | | | I2C0_SCL | **O** | I2C0 clock pin |
| 71 | | | PC.12 | **I/O** | General purpose digital I/O pin |
| | | | SPI1_MISO1 | **I/O** | SPI1 2nd MISO (Master In, Slave Out) pin |
| | | | PWM1_CH0 | **O** | PWM1 Channel0 output |
| | | | INT0 | **I** | External interrupt0 input pin |
| | | | I2C0_SDA | **I/O** | I2C0 data I/O pin |
| 72 | 33 | | PC.11 | **I/O** | General purpose digital I/O pin |
| | | | SPI1_MOSI0 | **I/O** | SPI1 1st MOSI (Master Out, Slave In) pin |
| | | | UART1_TXD | **O** | UART1 Data transmitter output pin |
| | | | LCD_SEG31 | **O** | LCD segment output 31 at LQFP64 |
| 73 | 34 | | PC.10 | **I/O** | General purpose digital I/O pin |
| | | | SPI1_MISO0 | **I/O** | SPI1 1st MISO (Master In, Slave Out) pin |
| | | | UART1_RXD | **I** | UART1 Data receiver input pin |
| | | | LCD_SEG30 | **O** | LCD segment output 30 at LQFP64 |
| 74 | 35 | | PC.9 | **I/O** | General purpose digital I/O pin |
| | | | SPI1_CLK | **I/O** | SPI1 serial clock pin |
| | | | I2C1_SCL | **I/O** | I2C1 clock pin |
| | | | LCD_SEG29 | **O** | LCD segment output 29 at LQFP64 |
| 75 | 36 | | PC.8 | **I/O** | General purpose digital I/O pin |
| | | | SPI1_SS0 | **I/O** | SPI1 1st slave select pin |
| | | | EBI_MCLK | **O** | EBI external clock output pin |
| | | | I2C1_SDA | **I/O** | I2C1 data I/O pin |
| | | | LCD_SEG28 | **O** | LCD segment output 28 at LQFP64 |
| 76 | 37 | | PA.15 | **I/O** | General purpose digital I/O pin |
| | | | PWM0_CH3 | **I/O** | PWM0 Channel3 output |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | I2S_MCLK | O | I²S master clock output pin |
| | | | TC3 | I | Timer3 capture input |
| | | | SC0_PWR | O | SmartCard0 Power pin |
| | | | UART0_TXD | O | UART0 Data transmitter output pin |
| | | | LCD_SEG27 | O | LCD segment output 27 at LQFP64 |
| 77 | 38 | | PA.14 | I/O | General purpose digital I/O pin |
| | | | PWM0_CH2 | I/O | PWM0 Channel2 output |
| | | | EBI_AD15 | I/O | EBI Address/Data bus bit15 |
| | | | TC2 | I | Timer2 capture input |
| | | | UART0_RXD | I | UART0 Data receiver input pin |
| | | | LCD_SEG26 | O | LCD segment output 26 at LQFP64 |
| 78 | 39 | | PA.13 | I/O | General purpose digital I/O pin |
| | | | PWM0_CH1 | I/O | PWM0 Channel1 output |
| | | | EBI_AD14 | I/O | EBI Address/Data bus bit14 |
| | | | TC1 | I | Timer1 capture input |
| | | | I2C0_SCL | I/O | I²C0 clock pin |
| | | | LCD_SEG25 | O | LCD segment output 25 at LQFP64 |
| 79 | 40 | | PA.12 | I/O | General purpose digital I/O pin |
| | | | PWM0_CH0 | I/O | PWM0 Channel0 output |
| | | | EBI_AD13 | I/O | EBI Address/Data bus bit13 |
| | | | TC0 | I | Timer0 capture input |
| | | | I2C0_SDA | I/O | I²C0 data I/O pin |
| | | | LCD_SEG24 | O | LCD segment output 24 at LQFP64 |
| 80 | 41 | | ICE_DAT | I/O | Serial Wired Debugger Data pin |
| | | | PF.0 | I/O | General purpose digital I/O pin |
| | | | INT0 | I | External interrupt0 input pin |
| 81 | 42 | | ICE_CLK | I | Serial Wired Debugger Clock pin |
| | | | PF.1 | I/O | General purpose digital I/O pin |
| | | | FCLKO | O | Frequency Divider output pin |
| | | | INT1 | I | External interrupt1 input pin |
| 82 | | | | | NC |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| 83 | | | VDD | **P** | Power supply for I/O ports and LDO source for internal PLL and digital circuit |
| 84 | | | | | NC |
| 85 | | | VSS | **P** | Ground |
| 86 | | | VSS | **P** | Ground |
| 87 | 43 | | AVSS | **AP** | Ground Pin for analog circuit |
| 88 | | | AVSS | **AP** | Ground Pin for analog circuit |
| 89 | 44 | | PA.0 | **I/O** | General purpose digital I/O pin |
| | | | AD0 | **AI** | ADC analog input0 |
| | | | SC2_CD | **I** | SmartCard2 card detect |
| 90 | 45 | | PA.1 | **I/O** | General purpose digital I/O pin |
| | | | AD1 | **AI** | ADC analog input1 |
| | | | EBI_AD12 | **I/O** | EBI Address/Data bus bit12 |
| 91 | 46 | | PA.2 | **I/O** | General purpose digital I/O pin |
| | | | AD2 | **AI** | ADC analog input2 |
| | | | EBI_AD11 | **I/O** | EBI Address/Data bus bit11 |
| | | | UART1_RXD | **I** | UART1 Data receiver input pin |
| | | | LCD_SEG23* | **AO** | LCD segment output 23 at LQFP64 |
| 92 | 47 | | PA.3 | **I/O** | General purpose digital I/O pin |
| | | | AD3 | **AI** | ADC analog input3 |
| | | | EBI_AD10 | **I/O** | EBI Address/Data bus bit10 |
| | | | UART1_TXD | **O** | UART1 Data transmitter output pin |
| | | | LCD_SEG22* | **AO** | LCD segment output 22 at LQFP64 |
| 93 | 48 | | PA.4 | **I/O** | General purpose digital I/O pin |
| | | | AD4 | **AI** | ADC analog input4 |
| | | | EBI_AD9 | **I/O** | EBI Address/Data bus bit9 |
| | | | SC2_PWR | **O** | SmartCard2 Power pin |
| | | | I2C0_SDA | **I/O** | $I^2C0$ data I/O pin |
| | | | LCD_SEG21* | **AO** | LCD segment output 21 at LQFP64 |
| | | | LCD_SEG39* | **AO** | LCD segment output 39 at LQFP128 |
| 94 | 49 | | PA.5 | **I/O** | General purpose digital I/O pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | AD5 | **AI** | ADC analog input5 |
| | | | EBI_AD8 | **I/O** | EBI Address/Data bus bit8 |
| | | | SC2_RST | **O** | SmartCard2 RST pin |
| | | | I2C0_SCL | **I/O** | $I^2C0$ clock pin |
| | | | LCD_SEG20* | **AO** | LCD segment output 19 at LQFP64 |
| | | | LCD_SEG38* | **AO** | LCD segment output 37 at LQFP128 |
| 95 | 50 | | PA.6 | **I/O** | General purpose digital I/O pin |
| | | | AD6 | **AI** | ADC analog input6 |
| | | | EBI_AD7 | **I/O** | EBI Address/Data bus bit7 |
| | | | TC3 | **I** | Timer3 capture input |
| | | | SC2_CLK | **O** | SmartCard2 clock pin(SC2_UART_TXD) |
| | | | PWM0_CH3 | **O** | PWM0 Channel3 output |
| | | | LCD_SEG19* | **AO** | LCD segment output 19 at LQFP64 |
| | | | LCD_SEG37* | **AO** | LCD segment output 37 at LQFP128 |
| 96 | | | PA.7 | **I/O** | General purpose digital I/O pin |
| | | | AD7 | **AI** | ADC analog input7 |
| | | | EBI_AD6 | **I/O** | EBI Address/Data bus bit6 |
| | | | TC2 | **I** | Timer2 capture input |
| | | | SC2_DAT | **I/O** | SmartCard2 DATA pin(SC2_UART_RXD) |
| | | | PWM0_CH2 | **O** | PWM0 Channel2 output |
| | | | LCD_SEG36* | **AO** | LCD segment output 36 output at LQFP128 |
| 97 | 51 | | VREF | **AP** | Voltage reference input for ADC |
| 98 | | | | | NC |
| 99 | 52 | | AVDD | **AP** | Power supply for internal analog circuit |
| 100 | | | PD.0 | **I/O** | General purpose digital I/O pin |
| | | | UART1_RXD | **I** | UART1 Data receiver input pin |
| | | | SPI2_SS0 | **I/O** | SPI2 1st slave select pin |
| | | | SC1_CLK | **O** | SmartCard1 clock pin(SC1_UART_TXD) |
| | | | AD8 | **AI** | ADC analog input8 |
| 101 | | | PD.1 | **I/O** | General purpose digital I/O pin |
| | | | UART1_TXD | **O** | UART1 Data transmitter output pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | SPI2_CLK | **I/O** | SPI2 serial clock pin |
| | | | SC1_DAT | **I/O** | SmartCard1 DATA pin(SC1_UART_RXD) |
| | | | AD9 | **AI** | ADC analog input9 |
| 102 | | | PD.2 | **I/O** | General purpose digital I/O pin |
| | | | UART1_RTSn | | UART1 Request to Send output pin |
| | | | I2S_LRCLK | **I/O** | $I^2S$ left right channel clock |
| | | | SPI2_MISO0 | **I/O** | SPI2 1st MISO (Master In, Slave Out) pin |
| | | | SC1_PWR | **O** | SmartCard1 Power pin |
| | | | AD10 | **AI** | ADC analog input10 |
| 103 | | | PD.3 | **I/O** | General purpose digital I/O pin |
| | | | UART1_CTSn | | UART1 Clear to Send input pin |
| | | | I2S_BCLK | **I/O** | $I^2S$ bit clock pin |
| | | | SPI2_MOSI0 | **I/O** | SPI2 1st MOSI (Master Out, Slave In) pin |
| | | | SC1_RST | **O** | SmartCard1 RST pin |
| | | | AD11 | **AI** | ADC analog input11 |
| 104 | | | | | NC |
| 105 | | | PD.4 | **I/O** | General purpose digital I/O pin |
| | | | I2S_DI | **I** | $I^2S$ data input |
| | | | SPI2_MISO1 | **I/O** | SPI2 2nd MISO (Master In, Slave Out) pin |
| | | | SC1_CD | **I** | SmartCard1 card detect |
| | | | LCD_SEG35 | **AO** | LCD segment output 35 at LQFP10 |
| 106 | | | PD.5 | **I/O** | General purpose digital I/O pin |
| | | | I2S_DO | **O** | $I^2S$ data output |
| | | | SPI2_MOSI1 | **I/O** | SPI2 2nd MOSI (Master Out, Slave In) pin |
| | | | LCD_SEG34 | **AO** | LCD segment output 34 at LQFP128 |
| 107 | 53 | | PC.7 | **I/O** | General purpose digital I/O pin |
| | | | DA1_OUT | **AO** | DAC 1 output |
| | | | EBI_AD5 | **I/O** | EBI Address/Data bus bit5 |
| | | | TC1 | **I** | Timer1 capture input |
| | | | PWM0_CH1 | **O** | PWM1 Channel1 output |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | LCD_SEG17* | **AO** | LCD segment output 17 at LQFP64 |
| 108 | 54 | | PC.6 | **I/O** | General purpose digital I/O pin |
| | | | DA0_OUT | **I** | DAC0 output |
| | | | EBI_AD4 | **I/O** | EBI Address/Data bus bit4 |
| | | | TC0 | **I** | Timer0 capture input |
| | | | SC1_CD | **I** | SmartCard1 card detect pin |
| | | | PWM0_CH0 | **O** | PWM0 Channel0 output |
| 109 | 55 | | PC.15 | **I/O** | General purpose digital I/O pin |
| | | | EBI_AD3 | **I/O** | EBI Address/Data bus bit3 |
| | | | TC0 | **I** | Timer0 capture input |
| | | | PWM1_CH2 | **O** | PWM1 Channel1 output |
| | | | LCD_SEG16 | **AO** | LCD segment output 16 at LQFP64 |
| | | | LCD_SEG33 | **AO** | LCD segment output 33 at LQFP128 |
| 110 | 56 | | PC.14 | **I/O** | General purpose digital I/O pin |
| | | | EBI_AD2 | **I/O** | EBI Address/Data bus bit2 |
| | | | PWM1_CH3 | **I/O** | PWM1 Channel3 output |
| | | | LCD_SEG15 | **AO** | LCD segment output 15 at LQFP64 |
| | | | LCD_SEG32 | **AO** | LCD segment output 32 at LQFP128 |
| 111 | 57 | | PB.15 | **I/O** | General purpose digital I/O pin |
| | | | INT1 | **I** | External interrupt1 input pin |
| | | | SNOOPER | **I** | Snooper pin |
| | | | LCD_SEG14 | **AO** | LCD segment output 14 at LQFP64 |
| | | | LCD_SEG31 | **AO** | LCD segment output 31 at LQFP128 |
| 112 | | | | | NC |
| 113 | 58 | | XT1_IN | **O** | External 4~24 MHz crystal output pin |
| | | | PF.3 | **I/O** | General purpose digital I/O pin |
| 114 | 59 | | XT1_OUT | **I** | External 4~24 MHz crystal input pin |
| | | | PF.2 | **I/O** | General purpose digital I/O pin |
| 115 | | | | | NC |
| 116 | 60 | | nRESET | **I** | External reset input: Low active, set this pin low reset chip to initial state. With internal pull-up. |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| 117 | 61 | | VSS | **P** | Ground |
| 118 | | | VSS | **P** | Ground |
| 119 | | | | | NC |
| 120 | 62 | | VDD | **P** | Power supply for I/O ports and LDO source for internal PLL and digital circuit |
| 121 | | | | | NC |
| 122 | | | PF.4 | **I/O** | General purpose digital I/O pin |
| | | | I2C0_SDA | **I/O** | $I^2C0$ data I/O pin |
| 123 | | | PF.5 | **I/O** | General purpose digital I/O pin |
| | | | I2C0_SCL | **I/O** | $I^2C0$ clock pin |
| 124 | | | VSS | **P** | Ground |
| 125 | 63 | | PVSS | **P** | PLL Ground |
| 126 | 64 | | PB.8 | **I/O** | General purpose digital I/O pin |
| | | | STADC | **I** | ADC external trigger input. |
| | | | TM0 | **I** | Timer0 external counter input |
| | | | INT0 | **I** | External interrupt0 input pin |
| | | | SC2_PWR | **O** | SmartCard2 Power pin |
| | | | LCD_SEG13 | **AO** | LCD segment output 13 at LQFP64 |
| | | | LCD_SEG30 | **AO** | LCD segment output 30 at LQFP128 |
| 127 | | | PE.15 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG29 | **O** | LCD segment output 29 at LQFP128 |
| 128 | | | PE.14 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG28 | **O** | LCD segment output 28 at LQFP128 |

**Note:**

1. Pin Type: I = Digital Input, O=Digital Output; AI=Analog Input; AO= Analog Output; P=Power Pin; AP=Analog Power;
2. * : Output voltage for ADC/LCD shared pins cannot be higher than VDD because these pins are without 5V tolerance.

### 3.4.3 NuMicro™ Nano120 Pin Description

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128 | LQFP 64 | LQFP 48 | | | |
| 1 | | | PE.13 | **I/O** | General purpose digital IO pin |
| 2 | 1 | | PB.14 | **I/O** | General purpose digital IO pin |
| | | | INT0 | **I** | External interrupt0 input pin |
| | | | SC2_CD | **I** | SmartCard2 card detect |
| | | | SPI2_SS1 | **I/O** | SPI2 2nd slave select pin |
| 3 | 2 | | PB.13 | **I/O** | General purpose digital IO pin |
| | | | EBI_AD1 | **I/O** | EBI Address/Data bus bit1 |
| 4 | 3 | 1 | PB.12 | **I/O** | General purpose digital IO pin |
| | | | EBI_AD0 | **I/O** | EBI Address/Data bus bit0 |
| | | | FCLKO | **O** | Frequency Divider output pin |
| 5 | | | | | NC |
| 6 | 4 | 2 | X32O | **O** | External 32.768 kHz crystal output pin |
| 7 | 5 | 3 | X32I | **I** | External 32.768 kHz crystal input pin |
| 8 | | | | | NC |
| 9 | 6 | 4 | PA.11 | **I/O** | General purpose digital IO pin |
| | | | I2C1_SCL | **I/O** | $I^2C$ 1 clock pin |
| | | | EBI_nRD | **O** | EBI read enable output pin |
| | | | SC0_RST | **O** | SmartCard0 RST pin |
| | | | SPI2_MOSI0 | **I/O** | SPI2 1st MOSI (Master Out, Slave In) pin |
| 10 | 7 | 5 | PA.10 | **I/O** | General purpose digital IO pin |
| | | | I2C1_SDA | **I/O** | $I^2C$ 1 data I/O pin |
| | | | EBI_nWR | **O** | EBI write enable output pin |
| | | | SC0_PWR | **O** | SmartCard0 Power pin |
| | | | SPI2_MISO0 | **I/O** | SPI2 1st MISO (Master In, Slave Out) pin |
| 11 | 8 | 6 | PA.9 | **I/O** | General purpose digital IO pin |
| | | | I2C0_SCL | **I/O** | $I^2C$ 0 clock pin |
| | | | SC0_DAT | **I/O** | SmartCard0 DATA pin(SC0_UART_RXD) |
| | | | SPI2_CLK | **I/O** | SPI2 serial clock pin |
| 12 | 9 | 7 | PA.8 | **I/O** | General purpose digital IO pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128 | LQFP 64 | LQFP 48 | | | |
| | | | I2C0_SDA | **I/O** | $I^2C$ 0 data I/O pin |
| | | | SC0_CLK | **O** | SmartCard0 clock pin(SC0_UART_TXD) |
| | | | SPI2_SS0 | **I/O** | SPI2 1st slave select pin |
| 13 | | | PD.8 | **I/O** | General purpose digital IO pin |
| 14 | | | PD.9 | **I/O** | General purpose digital IO pin |
| 15 | | | PD.10 | **I/O** | General purpose digital IO pin |
| 16 | | | PD.11 | **I/O** | General purpose digital IO pin |
| 17 | | | PD.12 | **I/O** | General purpose digital IO pin |
| 18 | | | PD.13 | **I/O** | General purpose digital IO pin |
| 19 | 10 | 8 | PB.4 | **I/O** | General purpose digital IO pin |
| | | | UART1_RXD | **I** | UART1 Data receiver input pin |
| | | | SC0_CD | **I** | SmartCard0 card detect pin |
| | | | SPI2_SS0 | **I/O** | SPI2 1st slave select pin |
| 20 | 11 | 9 | PB.5 | **I/O** | General purpose digital IO pin |
| | | | UART1_TXD | **O** | UART1 Data transmitter output pin |
| | | | SC0_RST | **O** | SmartCard0 RST pin |
| | | | SPI2_CLK | **I/O** | SPI2 serial clock pin |
| 21 | 12 | | PB.6 | **I/O** | General purpose digital IO pin |
| | | | UART1_nRTS | **O** | UART1 Request to Send output pin |
| | | | EBI_ALE | **O** | EBI address latch enable output pin |
| | | | SPI2_MISO0 | **I/O** | SPI2 1st MISO (Master In, Slave Out) pin |
| 22 | 13 | | PB.7 | **I/O** | General purpose digital IO pin |
| | | | UART1_nCTS | **I** | UART1 Clear to Send input pin |
| | | | EBI_nCS | **O** | EBI chip select enable output pin |
| | | | SPI2_MOSI0 | **I/O** | SPI2 1st MOSI (Master Out, Slave In) pin |
| 23 | | | | | NC |
| 24 | 14 | 10 | LDO_CAP | **P** | LDO output pin |
| 25 | | | | | NC |
| 26 | | | | | NC |
| 27 | 15 | 11 | VDD | **P** | Power supply for I/O ports and LDO source |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128 | LQFP 64 | LQFP 48 | | | |
| 28 | | | | | NC |
| 29 | 16 | 12 | VSS | **P** | Ground |
| 30 | | | VSS | **P** | Ground |
| 31 | | | VSS | **P** | Ground |
| 32 | | | VSS | **P** | Ground |
| 33 | | | PE.12 | **I/O** | General purpose digital IO pin |
| 34 | | | PE.11 | **I/O** | General purpose digital IO pin |
| 35 | | | PE.10 | **I/O** | General purpose digital IO pin |
| 36 | | | PE.9 | **I/O** | General purpose digital IO pin |
| 37 | | | PE.8 | **I/O** | General purpose digital IO pin |
| 38 | | | PE.7 | **I/O** | General purpose digital IO pin |
| 39 | | | | | NC |
| 40 | 17 | 13 | USB_VBUS | **USB** | POWER SUPPLY: From USB Host or HUB. |
| 41 | 18 | 14 | USB_VDD33_CAP | **USB** | Internal Power Regulator Output 3.3V Decoupling Pin |
| 42 | 19 | 15 | USB_D- | **USB** | USB Differential Signal D- |
| 43 | 20 | 16 | USB_D+ | **USB** | USB Differential Signal D+ |
| 44 | 21 | 17 | PB.0 | **I/O** | General purpose digital IO pin |
| | | | UART0_RXD | **I** | UART0 Data receiver input pin |
| | | | SPI1_MOSI0 | **I/O** | SPI1 1st MOSI (Master Out, Slave In) pin |
| 45 | 22 | 18 | PB.1 | **I/O** | General purpose digital IO pin |
| | | | UART0_TXD | **O** | UART0 Data transmitter output pin |
| | | | SPI1_MISO0 | **I/O** | SPI1 1st MISO (Master In, Slave Out) pin |
| 46 | 23 | 19 | PB.2 | **I/O** | General purpose digital IO pin |
| | | | UART0_nRTS | **O** | UART0 Request to Send output pin |
| | | | EBI_nWRL | **O** | EBI low byte write enable output pin |
| | | | SPI1_CLK | **I/O** | SPI1 serial clock pin |
| 47 | 24 | 20 | PB.3 | **I/O** | General purpose digital IO pin |
| | | | UART0_nCTS | **I** | UART0 Clear to Send input pin |
| | | | EBI_nWRH | **O** | EBI high byte write enable output pin |
| | | | SPI1_SS0 | **I/O** | SPI1 1st slave select pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128 | LQFP 64 | LQFP 48 | | | |
| 48 | | | PD.6 | I/O | General purpose digital IO pin |
| 49 | | | PD.7 | I/O | General purpose digital IO pin |
| 50 | | | PD.14 | I/O | General purpose digital IO pin |
| 51 | | | PD.15 | I/O | General purpose digital IO pin |
| 52 | | | PC.5 | I/O | General purpose digital IO pin |
| | | | SPI0_MOSI1 | I/O | SPI0 2nd MOSI (Master Out, Slave In) pin |
| 53 | | | PC.4 | I/O | General purpose digital IO pin |
| | | | SPI0_MISO1 | I/O | SPI0 2nd MISO (Master In, Slave Out) pin |
| 54 | 25 | 21 | PC.3 | I/O | General purpose digital IO pin |
| | | | SPI0_MOSI0 | I/O | SPI0 1st MOSI (Master Out, Slave In) pin |
| | | | I2S_DO | O | I2S data output |
| | | | SC1_RST | O | SmartCard1 RST pin |
| 55 | 26 | 22 | PC.2 | I/O | General purpose digital IO pin |
| | | | SPI0_MISO0 | I/O | SPI0 1st MISO (Master In, Slave Out) pin |
| | | | I2S_DI | I | I2S data input |
| | | | SC1_PWR | O | SmartCard1 PWR pin |
| 56 | 27 | 23 | PC.1 | I/O | General purpose digital IO pin |
| | | | SPI0_CLK | I/O | SPI0 serial clock pin |
| | | | I2S_BCLK | I/O | I2S bit clock pin |
| | | | SC1_DAT | I/O | SmartCard1 DATA pin(SC1_UART_RXD) |
| 57 | 28 | 24 | PC.0 / MCLKO | I/O | General purpose digital IO pin / Module clock output pin |
| | | | SPI0_SS0 | I/O | SPI0 1st slave select pin |
| | | | I2S_LRCLK | I/O | I2S left right channel clock |
| | | | SC1_CLK | O | SmartCard1 clock pin(SC1_UART_TXD) |
| 58 | | | PE.6 | I/O | General purpose digital IO pin |
| 59 | | | | | NC |
| 60 | | | | | NC |
| 61 | 29 | | PE.5 | I/O | General purpose digital IO pin |
| | | | PWM1_CH1 | I/O | PWM1 Channel1 output |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128 | LQFP 64 | LQFP 48 | | | |
| 62 | 30 | | PB.11 | I/O | General purpose digital IO pin |
| | | | PWM1_CH0 | I/O | PWM1 Channel0 output |
| | | | TM3 | O | Timer3 external counter input |
| | | | SC2_DAT | I/O | SmartCard2 DATA pin(SC2_UART_RXD) |
| | | | SPI0_MISO0 | I/O | SPI0 1st MISO (Master In, Slave Out) pin |
| 63 | 31 | | PB.10 | I/O | General purpose digital IO pin |
| | | | SPI0_SS1 | I/O | SPI0 2nd slave select pin |
| | | | TM2 | O | Timer2 external counter input |
| | | | SC2_CLK | O | SmartCard2 clock pin(SC2_UART_TXD) |
| | | | SPI0_MOSI0 | I/O | SPI0 1st MOSI (Master Out, Slave In) pin |
| 64 | 32 | | PB.9 | I/O | General purpose digital IO pin |
| | | | SPI1_SS1 | I/O | SPI1 2nd slave select pin |
| | | | TM1 | O | Timer1 external counter input |
| | | | SC2_RST | O | SmartCard2 RST pin |
| | | | INT0 | I | External interrupt0 input pin |
| 65 | | | PE.4 | I/O | General purpose digital IO pin |
| | | | SPI0_MOSI0 | I/O | SPI0 1st MOSI (Master Out, Slave In) pin |
| 66 | | | PE.3 | I/O | General purpose digital IO pin |
| | | | SPI0_MISO0 | I/O | SPI0 1st MISO (Master In, Slave Out) pin |
| 67 | | | PE.2 | I/O | General purpose digital IO pin |
| | | | SPI0_CLK | I/O | SPI0 serial clock pin |
| 68 | | | PE.1 | I/O | General purpose digital IO pin |
| | | | PWM1_CH3 | I/O | PWM1 Channel3 output |
| | | | SPI0_SS0 | I/O | SPI0 1st slave select pin |
| 69 | | | PE.0 | I/O | General purpose digital IO pin |
| | | | PWM1_CH2 | I/O | PWM1 Channel2 output |
| | | | I2S_MCLK | O | $I^2S$ master clock output pin |
| 70 | | | PC.13 | I/O | General purpose digital IO pin |
| | | | SPI1_MOSI1 | I/O | SPI1 2nd MOSI (Master Out, Slave In) pin |
| | | | PWM1_CH1 | O | PWM1 Channel1 output |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128 | LQFP 64 | LQFP 48 | | | |
| | | | SNOOPER | I | Snooper pin |
| | | | INT1 | I | External interrupt 1 input pin |
| | | | I2C0_SCL | O | $I^2C$ 0 clock pin |
| 71 | | | PC.12 | I/O | General purpose digital IO pin |
| | | | SPI1_MISO1 | I/O | SPI1 2nd MISO (Master In, Slave Out) pin |
| | | | PWM1_CH0 | O | PWM1 Channel 0 output |
| | | | INT0 | I | External interrupt 0 input pin |
| | | | I2C0_SDA | I/O | $I^2C$ 0 data I/O pin |
| 72 | 33 | | PC.11 | I/O | General purpose digital IO pin |
| | | | SPI1_MOSI0 | I/O | SPI1 1st MOSI (Master Out, Slave In) pin |
| | | | UART1_TXD | O | UART1 Data transmitter output pin |
| 73 | 34 | | PC.10 | I/O | General purpose digital IO pin |
| | | | SPI1_MISO0 | I/O | SPI1 1st MISO (Master In, Slave Out) pin |
| | | | UART1_RXD | I | UART1 Data receiver input pin |
| 74 | 35 | | PC.9 | I/O | General purpose digital IO pin |
| | | | SPI1_CLK | I/O | SPI1 serial clock pin |
| | | | I2C1_SCL | I/O | $I^2C$ 1 clock pin |
| 75 | 36 | | PC.8 | I/O | General purpose digital IO pin |
| | | | SPI1_SS0 | I/O | SPI1 1st slave select pin |
| | | | EBI_MCLK | O | EBI external clock output pin |
| | | | I2C1_SDA | I/O | $I^2C$ 1 data I/O pin |
| 76 | 37 | 25 | PA.15 | I/O | General purpose digital IO pin |
| | | | PWM0_CH3 | I/O | PWM0 Channel3 output |
| | | | I2S_MCLK | O | $I^2S$ master clock output pin |
| | | | TC3 | I | Timer3 capture input |
| | | | SC0_PWR | O | SmartCard0 Power pin |
| | | | UART0_TXD | O | UART0 Data transmitter output pin |
| 77 | 38 | 26 | PA.14 | I/O | General purpose digital IO pin |
| | | | PWM0_CH2 | I/O | PWM0 Channel2 output |
| | | | EBI_AD15 | I/O | EBI Address/Data bus bit15 |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128 | LQFP 64 | LQFP 48 | | | |
| | | | TC2 | **I** | Timer 2 capture input |
| | | | UART0_RXD | **I** | UART0 Data receiver input pin |
| 78 | 39 | 27 | PA.13 | **I/O** | General purpose digital IO pin |
| | | | PWM0_CH1 | **I/O** | PWM0 Channel1 output |
| | | | EBI_AD14 | **I/O** | EBI Address/Data bus bit14 |
| | | | TC1 | **I** | Timer1 capture input |
| | | | I2C0_SCL | **I/O** | $I^2C$ 0 clock pin |
| 79 | 40 | 28 | PA.12 | **I/O** | General purpose digital IO pin |
| | | | PWM0_CH0 | **I/O** | PWM0 Channel0 output |
| | | | EBI_AD13 | **I/O** | EBI Address/Data bus bit13 |
| | | | TC0 | **I** | Timer 0 capture input |
| | | | I2C0_SDA | **I/O** | $I^2C$ 0 data I/O pin |
| 80 | 41 | 29 | ICE_DAT | **I/O** | Serial Wired Debugger Data pin |
| | | | PF.0 | **I/O** | General purpose digital IO pin |
| | | | INT0 | **I** | External interrupt0 input pin |
| 81 | 42 | 30 | ICE_CLK | **I** | Serial Wired Debugger Clock pin |
| | | | PF.1 | **I/O** | General purpose digital IO pin |
| | | | FCLKO | **O** | Frequency Divider output pin |
| | | | INT1 | **I** | External interrupt1 input pin |
| 82 | | | | | NC |
| 83 | | | VDD | **P** | Power supply for I/O ports and LDO source for internal PLL and digital circuit |
| 84 | | | | | NC |
| 85 | | | VSS | **P** | Ground |
| 86 | | | VSS | **P** | Ground |
| 87 | 43 | 31 | AVSS | **AP** | Ground Pin for analog circuit |
| 88 | | | AVSS | **AP** | Ground Pin for analog circuit |
| 89 | 44 | 32 | PA.0 | **I/O** | General purpose digital IO pin |
| | | | AD0 | **AI** | ADC analog input0 |
| | | | SC2_CD | **I** | SmartCard2 card detect |
| 90 | 45 | 33 | PA.1 | **I/O** | General purpose digital IO pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128 | LQFP 64 | LQFP 48 | | | |
| | | | AD1 | **AI** | ADC analog input1 |
| | | | EBI_AD12 | **I/O** | EBI Address/Data bus bit12 |
| 91 | 46 | 34 | PA.2 | **I/O** | General purpose digital IO pin |
| | | | AD2 | **AI** | ADC analog input2 |
| | | | EBI_AD11 | **I/O** | EBI Address/Data bus bit11 |
| | | | UART1_RXD | **I** | UART1 Data receiver input pin |
| 92 | 47 | 35 | PA.3 | **I/O** | General purpose digital IO pin |
| | | | AD3 | **AI** | ADC analog input3 |
| | | | EBI_AD10 | **I/O** | EBI Address/Data bus bit10 |
| | | | UART1_TXD | **O** | UART1 Data transmitter output pin |
| 93 | 48 | 36 | PA.4 | **I/O** | Digital GPIO pin |
| | | | AD4 | **AI** | ADC analog input4 |
| | | | EBI_AD9 | **I/O** | EBI Address/Data bus bit9 |
| | | | SC2_PWR | **O** | SmartCard2 Power pin |
| | | | I2C0_SDA | **I/O** | $I^2C$ 0 data I/O pin |
| 94 | 49 | 37 | PA.5 | **I/O** | General purpose digital IO pin |
| | | | AD5 | **AI** | ADC analog input5 |
| | | | EBI_AD8 | **I/O** | EBI Address/Data bus bit8 |
| | | | SC2_RST | **O** | SmartCard2 RST pin |
| | | | I2C0_SCL | **I/O** | $I^2C$ 0 clock pin |
| 95 | 50 | 38 | PA.6 | **I/O** | General purpose digital IO pin |
| | | | AD6 | **AI** | ADC analog input6 |
| | | | EBI_AD7 | **I/O** | EBI Address/Data bus bit7 |
| | | | TC3 | **I** | Timer3 capture input |
| | | | SC2_CLK | **O** | SmartCard2 clock pin(SC2_UART_TXD) |
| | | | PWM0_CH3 | **O** | PWM0 Channel3 output |
| 96 | | | PA.7 | **I/O** | General purpose digital IO pin |
| | | | AD7 | **AI** | ADC analog input7 |
| | | | EBI_AD6 | **I/O** | EBI Address/Data bus bit6 |
| | | | TC2 | **I** | Timer2 capture input |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128 | LQFP 64 | LQFP 48 | | | |
| | | | SC2_DAT | **I/O** | SmartCard2 DATA pin(SC2_UART_RXD) |
| | | | PWM0_CH2 | **O** | PWM0 Channel2 output |
| 97 | 51 | 39 | VREF | **AP** | Voltage reference input for ADC |
| 98 | | | | | NC |
| 99 | 52 | 40 | AVDD | **AP** | Power supply for internal analog circuit |
| 100 | | | PD.0 | **I/O** | General purpose digital IO pin |
| | | | UART1_RXD | **I** | UART1 Data receiver input pin |
| | | | SPI2_SS0 | **I/O** | SPI2 1st slave select pin |
| | | | SC1_CLK | **O** | SmartCard1 clock pin(SC1_UART_TXD) |
| | | | AD8 | **AI** | ADC analog input8 |
| 101 | | | PD.1 | **I/O** | General purpose digital IO pin |
| | | | UART1_TXD | **O** | UART1 Data transmitter output pin |
| | | | SPI2_CLK | **I/O** | SPI2 serial clock pin |
| | | | SC1_DAT | **I/O** | SmartCard1 DATA pin(SC1_UART_RXD) |
| | | | AD9 | **AI** | ADC analog input9 |
| 102 | | | PD.2 | **I/O** | General purpose digital IO pin |
| | | | UART1_nRTS | **O** | UART1 Request to Send output pin |
| | | | I2S_LRCLK | **I/O** | I2S left right channel clock |
| | | | SPI2_MISO0 | **I/O** | SPI2 1st MISO (Master In, Slave Out) pin |
| | | | SC1_PWR | **O** | SmartCard1 Power pin |
| | | | AD10 | **AI** | ADC analog input10 |
| 103 | | | PD.3 | **I/O** | General purpose digital IO pin |
| | | | UART1_nCTS | **I** | UART1 Clear to Send input pin |
| | | | I2S_BCLK | **I/O** | I2S bit clock pin |
| | | | SPI2_MOSI0 | **I/O** | SPI2 1st MOSI (Master Out, Slave In) pin |
| | | | SC1_RST | **O** | SmartCard1 RST pin |
| | | | AD11 | **AI** | ADC analog input11 |
| 104 | | | | | NC |
| 105 | | | PD.4 | **I/O** | General purpose digital IO pin |
| | | | I2S_DI | **I** | I2S data input |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128 | LQFP 64 | LQFP 48 | | | |
| | | | SPI2_MISO1 | **I/O** | SPI2 2nd MISO (Master In, Slave Out) pin |
| | | | SC1_CD | **I** | SmartCard1 card detect |
| 106 | | | PD.5 | **I/O** | General purpose digital IO pin |
| | | | I2S_DO | **O** | I2S data output |
| | | | SPI2_MOSI1 | **I/O** | SPI2 2nd MOSI (Master Out, Slave In) pin |
| 107 | 53 | 41 | PC.7 | **I/O** | General purpose digital IO pin |
| | | | DA1+OUT | **AO** | DAC 1 output |
| | | | EBI_AD5 | **I/O** | EBI Address/Data bus bit5 |
| | | | TC1 | **I** | Timer1 capture input |
| | | | PWM0_CH1 | **O** | PWM1 Channel1 output |
| 108 | 54 | 42 | PC.6 | **I/O** | General purpose digital IO pin |
| | | | DA0_OUT | **I** | DAC0 output |
| | | | EBI_AD4 | **I/O** | EBI Address/Data bus bit4 |
| | | | TC0 | **I** | Timer 0 capture input |
| | | | SC1_CD | | SmartCard1 card detect pin |
| | | | PWM0_CH0 | **O** | PWM0 Channel0 output |
| 109 | 55 | | PC.15 | **I/O** | General purpose digital IO pin |
| | | | EBI_AD3 | **I/O** | EBI Address/Data bus bit3 |
| | | | TC0 | **I** | Timer0 capture input |
| | | | PWM1_CH2 | **O** | PWM1 Channel1 output |
| 110 | 56 | | PC.14 | **I/O** | General purpose digital IO pin |
| | | | EBI_AD2 | **I/O** | EBI Address/Data bus bit2 |
| | | | PWM1_CH3 | **I/O** | PWM1 Channel3 output |
| 111 | 57 | 43 | PB.15 | **I/O** | General purpose digital IO pin |
| | | | INT1 | **I** | External interrupt1 input pin |
| | | | SNOOPER | **I** | Snooper pin |
| | | | SC1_CD | **I** | SmartCard1 card detect |
| 112 | | | | | NC |
| 113 | 58 | 44 | XT1_IN | **O** | External 4~24 MHz crystal output pin |
| | | | PF.3 | **I/O** | General purpose digital I/O pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128 | LQFP 64 | LQFP 48 | | | |
| 114 | 59 | 45 | XT1_OUT | **I** | External 4~24 MHz crystal input pin |
| | | | PF.2 | **I/O** | General purpose digital I/O pin |
| 115 | | | | | NC |
| 116 | 60 | 46 | nRESET | **I** | External reset input: Low active, set this pin low reset chip to initial state. With internal pull-up. |
| 117 | 61 | | VSS | **P** | Ground |
| 118 | | | VSS | **P** | Ground |
| 119 | | | | | NC |
| 120 | 62 | | VDD | **P** | Power supply for I/O ports and LDO source for internal PLL and digital circuit |
| 121 | | | | | NC |
| 122 | | | PF.4 | **I/O** | General purpose digital IO pin |
| | | | I2C0_SDA | **I/O** | $I^2C$ 0 data I/O pin |
| 123 | | | PF.5 | **I/O** | General purpose digital IO pin |
| | | | I2C0_SCL | **I/O** | $I^2C$ 0 clock pin |
| 124 | | | VSS | **P** | Ground |
| 125 | 63 | 47 | PVSS | **P** | PLL Ground |
| 126 | 64 | 48 | PB.8 | **I/O** | General purpose digital IO pin |
| | | | STADC | **I** | ADC external trigger input. |
| | | | TM0 | **I** | Timer0 external counter input |
| | | | INT0 | **I** | External interrupt0 input pin |
| | | | SC2_PWR | **O** | SmartCard2 Power pin |
| 127 | | | PE.15 | **I/O** | General purpose digital IO pin |
| 128 | | | PE.14 | **I/O** | General purpose digital IO pin |

**Note:**

1. Pin Type: I = Digital Input, O=Digital Output; AI=Analog Input; AO= Analog Output; P=Power Pin; AP=Analog Power;

### 3.4.4 NuMicro™ Nano130 Pin Description

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| 1 | | | PE.13 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG27 | **O** | LCD segment output 27 at LQFP128 |
| 2 | 1 | | PB.14 | **I/O** | General purpose digital I/O pin |
| | | | INT0 | **I** | External interrupt0 input pin |
| | | | SC2_CD | **I** | SmartCard2 card detect |
| | | | SPI2_SS1 | **I/O** | SPI2 2nd slave select pin |
| | | | LCD_SEG12 | **O** | LCD segment output 12 at LQFP64 |
| | | | LCD_SEG26 | **O** | LCD segment output 26 at LQFP128 |
| 3 | 2 | | PB.13 | **I/O** | General purpose digital I/O pin |
| | | | EBI_AD1 | **I/O** | EBI Address/Data bus bit1 |
| | | | LCD_SEG11 | **O** | LCD segment output 11 at LQFP64 |
| | | | LCD_SEG25 | **O** | LCD segment output 25 at LQFP128 |
| 4 | 3 | | PB.12 | **I/O** | General purpose digital I/O pin |
| | | | EBI_AD0 | **I/O** | EBI Address/Data bus bit0 |
| | | | FCLKO | **O** | Frequency Divider output pin |
| | | | LCD_SEG10 | **O** | LCD segment output 10 at LQFP64 |
| | | | LCD_SEG24 | **O** | LCD segment output 24 at LQFP128 |
| 5 | | | | | NC |
| 6 | 4 | | X32O | **O** | External 32.768 kHz crystal output pin |
| 7 | 5 | | X32I | **I** | External 32.768 kHz crystal input pin |
| 8 | | | | | NC |
| 9 | 6 | | PA.11 | **I/O** | General purpose digital I/O pin |
| | | | I2C1_SCL | **I/O** | $I^2C1$ clock pin |
| | | | EBI_nRD | **O** | EBI read enable output pin |
| | | | SC0_RST | **O** | SmartCard0 RST pin |
| | | | SPI2_MOSI0 | **I/O** | SPI2 1st MOSI (Master Out, Slave In) pin |
| | | | LCD_SEG9 | **O** | LCD segment output 9 at LQFP64 |
| | | | LCD_SEG23 | **O** | LCD segment output 23 at LQFP128 |
| 10 | 7 | | PA.10 | **I/O** | General purpose digital I/O pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | I2C1_SDA | **I/O** | $I^2C1$ data I/O pin |
| | | | EBI_nWR | **O** | EBI write enable output pin |
| | | | SC0_PWR | **O** | SmartCard0 Power pin |
| | | | SPI2_MISO0 | **I/O** | SPI2 1st MISO (Master In, Slave Out) pin |
| | | | LCD_SEG8 | **O** | LCD segment output 8 at LQFP64 |
| | | | LCD_SEG22 | **O** | LCD segment output 22 at LQFP128 |
| 11 | 8 | | PA.9 | **I/O** | General purpose digital I/O pin |
| | | | I2C0_SCL | **I/O** | $I^2C0$ clock pin |
| | | | SC0_DAT | **I/O** | SmartCard0 DATA pin(SC0_UART_RXD) |
| | | | SPI2_CLK | **I/O** | SPI2 serial clock pin |
| | | | LCD_SEG7 | **O** | LCD segment output 7 at LQFP64 |
| | | | LCD_SEG21 | **O** | LCD segment output 21 at LQFP128 |
| 12 | 9 | | PA.8 | **I/O** | General purpose digital I/O pin |
| | | | I2C0_SDA | **I/O** | $I^2C0$ data I/O pin |
| | | | SC0_CLK | **O** | SmartCard0 clock pin(SC0_UART_TXD) |
| | | | SPI2_SS0 | **I/O** | SPI2 1st slave select pin |
| | | | LCD_SEG6 | **O** | LCD segment output 6 at LQFP64 |
| | | | LCD_SEG20 | **O** | LCD segment output 20 at LQFP128 |
| 13 | | | PD.8 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG19 | **O** | LCD segment output 19 at LQFP128 |
| 14 | | | PD.9 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG18 | **O** | LCD segment output 18 at LQFP128 |
| 15 | | | PD.10 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG17 | **O** | LCD segment output 17 at LQFP128 |
| 16 | | | PD.11 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG16 | **O** | LCD segment output 16 at LQFP128 |
| 17 | | | PD.12 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG15 | **O** | LCD segment output 15 at LQFP128 |
| 18 | | | PD.13 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG14 | **O** | LCD segment output 14 at LQFP128 |
| 19 | 10 | | PB.4 | **I/O** | General purpose digital I/O pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | UART1_RXD | **I** | UART1 Data receiver input pin |
| | | | SC0_CD | **I** | SmartCard0 card detect pin |
| | | | SPI2_SS0 | **I/O** | SPI2 1st slave select pin |
| | | | LCD_SEG5 | **O** | LCD segment output 5 at LQFP64 |
| | | | LCD_SEG13 | **O** | LCD segment output 13 at LQFP128 |
| 20 | 11 | | PB.5 | **I/O** | General purpose digital I/O pin |
| | | | UART1_TXD | **O** | UART1 Data transmitter output pin |
| | | | SC0_RST | **O** | SmartCard0 RST pin |
| | | | SPI2_CLK | **I/O** | SPI2 serial clock pin |
| | | | LCD_SEG4 | **O** | LCD segment output 4 at LQFP64 |
| | | | LCD_SEG12 | **O** | LCD segment output 12 at LQFP128 |
| 21 | 12 | | PB.6 | **I/O** | General purpose digital I/O pin |
| | | | UART1_RTSn | **O** | UART1 Request to Send output pin |
| | | | EBI_ALE | **O** | EBI address latch enable output pin |
| | | | SPI2_MISO0 | **I/O** | SPI2 1st MISO (Master In, Slave Out) pin |
| | | | LCD_SEG3 | **O** | LCD segment output 3 at LQFP64 |
| | | | LCD_SEG11 | **O** | LCD segment output 11 at LQFP128 |
| 22 | 13 | | PB.7 | **I/O** | General purpose digital I/O pin |
| | | | UART1_CTSn | **I** | UART1 Clear to Send input pin |
| | | | EBI_nCS | **O** | EBI chip select enable output pin |
| | | | SPI2_MOSI0 | **I/O** | SPI2 1st MOSI (Master Out, Slave In) pin |
| | | | LCD_SEG2 | **O** | LCD segment output 2 at LQFP64 |
| | | | LCD_SEG10 | **O** | LCD segment output 10 at LQFP128 |
| 23 | | | | | NC |
| 24 | 14 | | LDO_CAP | **P** | LDO output pin |
| 25 | | | | | NC |
| 26 | | | | | NC |
| 27 | 15 | | VDD | **P** | Power supply for I/O ports and LDO source |
| 28 | | | | | NC |
| 29 | 16 | | VSS | **P** | Ground |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| 30 | | | VSS | **P** | Ground |
| 31 | | | VSS | **P** | Ground |
| 32 | | | VSS | **P** | Ground |
| 33 | | | PE.12 | **I/O** | General purpose digital I/O pin |
| 34 | | | PE.11 | **I/O** | General purpose digital I/O pin |
| 35 | | | PE.10 | **I/O** | General purpose digital I/O pin |
| 36 | | | PE.9 | **I/O** | General purpose digital I/O pin |
| 37 | | | PE.8 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG9 | **O** | LCD segment output 9 at LQFP128 |
| 38 | | | PE.7 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG8 | **O** | LCD segment output 8 at LQFP128 |
| 39 | | | | | NC |
| 40 | 17 | | USB_VBUS | **USB** | POWER SUPPLY: From USB Host or HUB. |
| 41 | 18 | | USB_VDD33_CAP | **USB** | Internal Power Regulator Output 3.3V Decoupling Pin |
| 42 | 19 | | USB_D- | **USB** | USB Differential Signal D- |
| 43 | 20 | | USB_D+ | **USB** | USB Differential Signal D+ |
| 44 | 21 | | PB.0 | **I/O** | General purpose digital I/O pin |
| | | | UART0_RXD | **I** | UART0 Data receiver input pin |
| | | | SPI1_MOSI0 | **I/O** | SPI1 1st MOSI (Master Out, Slave In) pin |
| | | | LCD_SEG1 | **O** | LCD segment output 1 at LQFP64 (or as LCD_COM5) |
| | | | LCD_SEG7 | **O** | LCD segment output 7 at LQFP128 |
| 45 | 22 | | PB.1 | **I/O** | General purpose digital I/O pin |
| | | | UART0_TXD | **O** | UART0 Data transmitter output pin |
| | | | SPI1_MISO0 | **I/O** | SPI1 1st MISO (Master In, Slave Out) pin |
| | | | LCD_SEG0 | **O** | LCD segment output 0 at LQFP64 (or as LCD_COM4) |
| | | | LCD_SEG6 | **O** | LCD segment output 6 at LQFP128 |
| 46 | 23 | | PB.2 | **I/O** | General purpose digital I/O pin |
| | | | UART0_RTSn | **O** | UART0 Request to Send output pin |
| | | | EBI_nWRL | **O** | EBI low byte write enable output pin |
| | | | SPI1_CLK | **I/O** | SPI1 serial clock pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | LCD_COM3 | **O** | LCD common output 3 at LQFP64 |
| | | | LCD_SEG5 | **O** | LCD segment output 5 at LQFP128 |
| 47 | 24 | | PB.3 | **I/O** | General purpose digital I/O pin |
| | | | UART0_CTSn | **I** | UART0 Clear to Send input pin |
| | | | EBI_nWRH | **O** | EBI high byte write enable output pin |
| | | | SPI1_SS0 | **I/O** | SPI1 1st slave select pin |
| | | | LCD_COM2 | **O** | LCD common output 2 at LQFP64 |
| | | | LCD_SEG4 | **O** | LCD segment output 4 at LQFP128 |
| 48 | | | PD.6 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG3 | **O** | LCD segment output 3 at LQFP128 |
| 49 | | | PD.7 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG2 | **O** | LCD segment output 2 at LQFP128 |
| 50 | | | PD.14 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG1 | **O** | LCD segment output 1 at LQFP128 (or as LCD_COM5) |
| 51 | | | PD.15 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG0 | **O** | LCD segment output 0 at LQFP128 (or as LCD_COM4) |
| 52 | | | PC.5 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_MOSI1 | **I/O** | SPI0 2nd MOSI (Master Out, Slave In) pin |
| | | | LCD_COM3 | **O** | LCD common output 3 at LQFP128 |
| 53 | | | PC.4 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_MISO1 | **I/O** | SPI0 2nd MISO (Master In, Slave Out) pin |
| | | | LCD_COM2 | **O** | LCD common output 2 at LQFP128 |
| 54 | 25 | | PC.3 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_MOSI0 | **I/O** | SPI0 1st MOSI (Master Out, Slave In) pin |
| | | | I2S_DO | **O** | $I^2S$ data output |
| | | | SC1_RST | **O** | SmartCard1 RST pin |
| | | | LCD_COM1 | **O** | LCD common output 1 at LQFP64 |
| | | | LCD_COM1 | **O** | LCD common output 1 at LQFP128 |
| 55 | 26 | | PC.2 | **I/O** | General purpose digital I/O pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | SPI0_MISO0 | **I/O** | SPI0 1st MISO (Master In, Slave Out) pin |
| | | | I2S_DI | **I** | I2S data input |
| | | | SC1_PWR | **O** | SmartCard1 PWR pin |
| | | | LCD_COM0 | **O** | LCD common output 0 at LQFP64 |
| | | | LCD_COM0 | **O** | LCD common output 0 at LQFP128 |
| | | | PC.1 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_CLK | **I/O** | SPI0 serial clock pin |
| | | | I2S_BCLK | **I/O** | I2S bit clock pin |
| 56 | 27 | | SC1_DAT | **I/O** | SmartCard1 DATA pin(SC1_UART_RXD) |
| | | | LCD_DH2 | **O** | LCD externl capacitor pin of charge pump circuit at LQFP64 |
| | | | LCD_DH2 | **O** | LCD externl capacitor pin of charge pump circuit at LQFP128 |
| | | | PC.0 / MCLKO | **I/O** | General purpose digital I/O pin / Module clock output pin |
| | | | SPI0_SS0 | **I/O** | SPI0 1st slave select pin |
| | | | I2S_LRCLK | **I/O** | I2S left right channel clock |
| 57 | 28 | | SC1_CLK | **O** | SmartCard1 clock pin(SC1_UART_TXD) |
| | | | LCD_DH1 | **O** | LCD externl capacitor pin of charge pump circuit at LQFP64 |
| | | | LCD_DH1 | **O** | LCD externl capacitor pin of charge pump circuit at LQFP128 |
| 58 | | | PE.6 | **I/O** | General purpose digital I/O pin |
| 59 | 29 | | LCD_VLCD | **AO** | LCD power supply pin |
| 60 | | | | | NC |
| 61 | | | PE.5 | | General purpose digital I/O pin |
| | | | PB.11 | **I/O** | General purpose digital I/O pin |
| | | | PWM1_CH0 | **I/O** | PWM1 Channel0 output |
| | | | TM3 | **O** | Timer3 external counter input |
| 62 | 30 | | SC2_DAT | **I/O** | SmartCard2 DATA pin(SC2_UART_RXD) |
| | | | SPI0_MISO0 | **I/O** | SPI0 1st MISO (Master In, Slave Out) pin |
| | | | LCD_V1 | **O** | LCD Unit voltage for LCD charge pump circuit at LQFP64 |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | LCD_V1 | **O** | LCD Unit voltage for LCD charge pump circuit at LQFP128 |
| 63 | 31 | | PB.10 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_SS1 | **I/O** | SPI0 2nd slave select pin |
| | | | TM2 | **O** | Timer2 external counter input |
| | | | SC2_CLK | **O** | SmartCard2 clock pin(SC2_UART_TXD) |
| | | | SPI0_MOSI0 | **I/O** | SPI0 1st MOSI (Master Out, Slave In) pin |
| | | | LCD_V2 | **O** | LCD driver biasing voltage at LQFP64 |
| | | | LCD_V2 | **O** | LCD driver biasing voltage at LQFP128 |
| 64 | 32 | | PB.9 | **I/O** | General purpose digital I/O pin |
| | | | SPI1_SS1 | **I/O** | SPI1 2nd slave select pin |
| | | | TM1 | **O** | Timer1 external counter input |
| | | | SC2_RST | **O** | SmartCard2 RST pin |
| | | | INT0 | **I** | External interrupt0 input pin |
| | | | LCD_V3 | **O** | LCD driver biasing voltage at LQFP64 |
| | | | LCD_V3 | **O** | LCD driver biasing voltage at LQFP128 |
| 65 | | | PE.4 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_MOSI0 | **I/O** | SPI0 1st MOSI (Master Out, Slave In) pin |
| 66 | | | PE.3 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_MISO0 | **I/O** | SPI0 1st MISO (Master In, Slave Out) pin |
| 67 | | | PE.2 | **I/O** | General purpose digital I/O pin |
| | | | SPI0_CLK | **I/O** | SPI0 serial clock pin |
| 68 | | | PE.1 | **I/O** | General purpose digital I/O pin |
| | | | PWM1_CH3 | **I/O** | PWM1 Channel3 output |
| | | | SPI0_SS0 | **I/O** | SPI0 1st slave select pin |
| 69 | | | PE.0 | **I/O** | General purpose digital I/O pin |
| | | | PWM1_CH2 | **I/O** | PWM1 Channel2 output |
| | | | I2S_MCLK | **O** | $I^2S$ master clock output pin |
| 70 | | | PC.13 | **I/O** | General purpose digital I/O pin |
| | | | SPI1_MOSI1 | **I/O** | SPI1 2nd MOSI (Master Out, Slave In) pin |
| | | | PWM1_CH1 | **O** | PWM1 Channel1 output |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | SNOOPER | I | Snooper pin |
| | | | INT1 | I | External interrupt 1 input pin |
| | | | I2C0_SCL | O | $I^2C$0 clock pin |
| 71 | | | PC.12 | I/O | General purpose digital I/O pin |
| | | | SPI1_MISO1 | I/O | SPI1 2nd MISO (Master In, Slave Out) pin |
| | | | PWM1_CH0 | O | PWM1 Channel0 output |
| | | | INT0 | I | External interrupt0 input pin |
| | | | I2C0_SDA | I/O | $I^2C$0 data I/O pin |
| 72 | 33 | | PC.11 | I/O | General purpose digital I/O pin |
| | | | SPI1_MOSI0 | I/O | SPI1 1st MOSI (Master Out, Slave In) pin |
| | | | UART1_TXD | O | UART1 Data transmitter output pin |
| | | | LCD_SEG31 | O | LCD segment output 31 at LQFP64 |
| 73 | 34 | | PC.10 | I/O | General purpose digital I/O pin |
| | | | SPI1_MISO0 | I/O | SPI1 1st MISO (Master In, Slave Out) pin |
| | | | UART1_RXD | I | UART1 Data receiver input pin |
| | | | LCD_SEG30 | O | LCD segment output 30 at LQFP64 |
| 74 | 35 | | PC.9 | I/O | General purpose digital I/O pin |
| | | | SPI1_CLK | I/O | SPI1 serial clock pin |
| | | | I2C1_SCL | I/O | $I^2C$1 clock pin |
| | | | LCD_SEG29 | O | LCD segment output 29 at LQFP64 |
| 75 | 36 | | PC.8 | I/O | General purpose digital I/O pin |
| | | | SPI1_SS0 | I/O | SPI1 1st slave select pin |
| | | | EBI_MCLK | O | EBI external clock output pin |
| | | | I2C1_SDA | I/O | $I^2C$1 data I/O pin |
| | | | LCD_SEG28 | O | LCD segment output 28 at LQFP64 |
| 76 | 37 | | PA.15 | I/O | General purpose digital I/O pin |
| | | | PWM0_CH3 | I/O | PWM0 Channel3 output |
| | | | I2S_MCLK | O | $I^2S$ master clock output pin |
| | | | TC3 | I | Timer3 capture input |
| | | | SC0_PWR | O | SmartCard0 Power pin |
| | | | UART0_TXD | O | UART0 Data transmitter output pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | LCD_SEG27 | **O** | LCD segment output 27 at LQFP64 |
| 77 | 38 | | PA.14 | **I/O** | General purpose digital I/O pin |
| | | | PWM0_CH2 | **I/O** | PWM0 Channel2 output |
| | | | EBI_AD15 | **I/O** | EBI Address/Data bus bit15 |
| | | | TC2 | **I** | Timer2 capture input |
| | | | UART0_RXD | **I** | UART0 Data receiver input pin |
| | | | LCD_SEG26 | **O** | LCD segment output 26 at LQFP64 |
| 78 | 39 | | PA.13 | **I/O** | General purpose digital I/O pin |
| | | | PWM0_CH1 | **I/O** | PWM0 Channel1 output |
| | | | EBI_AD14 | **I/O** | EBI Address/Data bus bit14 |
| | | | TC1 | **I** | Timer1 capture input |
| | | | I2C0_SCL | **I/O** | $I^2C0$ clock pin |
| | | | LCD_SEG25 | **O** | LCD segment output 25 at LQFP64 |
| 79 | 40 | | PA.12 | **I/O** | General purpose digital I/O pin |
| | | | PWM0_CH0 | **I/O** | PWM0 Channel0 output |
| | | | EBI_AD13 | **I/O** | EBI Address/Data bus bit13 |
| | | | TC0 | **I** | Timer0 capture input |
| | | | I2C0_SDA | **I/O** | $I^2C0$ data I/O pin |
| | | | LCD_SEG24 | **O** | LCD segment output 24 at LQFP64 |
| 80 | 41 | | ICE_DAT | **I/O** | Serial Wired Debugger Data pin |
| | | | PF.0 | **I/O** | General purpose digital I/O pin |
| | | | INT0 | **I** | External interrupt0 input pin |
| 81 | 42 | | ICE_CLK | **I** | Serial Wired Debugger Clock pin |
| | | | PF.1 | **I/O** | General purpose digital I/O pin |
| | | | FCLKO | **O** | Frequency Divider output pin |
| | | | INT1 | **I** | External interrupt1 input pin |
| 82 | | | | | NC |
| 83 | | | VDD | **P** | Power supply for I/O ports and LDO source for internal PLL and digital circuit |
| 84 | | | | | NC |
| 85 | | | VSS | **P** | Ground |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| 86 | | | VSS | **P** | Ground |
| 87 | 43 | | AVSS | **AP** | Ground Pin for analog circuit |
| 88 | | | AVSS | **AP** | Ground Pin for analog circuit |
| 89 | 44 | | PA.0 | **I/O** | General purpose digital I/O pin |
| | | | AD0 | **AI** | ADC analog input0 |
| | | | SC2_CD | **I** | SmartCard2 card detect |
| 90 | 45 | | PA.1 | **I/O** | General purpose digital I/O pin |
| | | | AD1 | **AI** | ADC analog input1 |
| | | | EBI_AD12 | **I/O** | EBI Address/Data bus bit12 |
| 91 | 46 | | PA.2 | **I/O** | General purpose digital I/O pin |
| | | | AD2 | **AI** | ADC analog input2 |
| | | | EBI_AD11 | **I/O** | EBI Address/Data bus bit11 |
| | | | UART1_RXD | **I** | UART1 Data receiver input pin |
| | | | LCD_SEG23* | **AO** | LCD segment output 23 at LQFP64 |
| 92 | 47 | | PA.3 | **I/O** | General purpose digital I/O pin |
| | | | AD3 | **AI** | ADC analog input3 |
| | | | EBI_AD10 | **I/O** | EBI Address/Data bus bit10 |
| | | | UART1_TXD | **O** | UART1 Data transmitter output pin |
| | | | LCD_SEG22* | **AO** | LCD segment output 22 at LQFP64 |
| 93 | 48 | | PA.4 | **I/O** | General purpose digital I/O pin |
| | | | AD4 | **AI** | ADC analog input4 |
| | | | EBI_AD9 | **I/O** | EBI Address/Data bus bit9 |
| | | | SC2_PWR | **O** | SmartCard2 Power pin |
| | | | I2C0_SDA | **I/O** | $I^2C0$ data I/O pin |
| | | | LCD_SEG21* | **AO** | LCD segment output 21 at LQFP64 |
| | | | LCD_SEG39* | **AO** | LCD segment output 39 at LQFP128 |
| 94 | 49 | | PA.5 | **I/O** | General purpose digital I/O pin |
| | | | AD5 | **AI** | ADC analog input5 |
| | | | EBI_AD8 | **I/O** | EBI Address/Data bus bit8 |
| | | | SC2_RST | **O** | SmartCard2 RST pin |
| | | | I2C0_SCL | **I/O** | $I^2C0$ clock pin |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | LCD_SEG20* | **AO** | LCD segment output 20 at LQFP64 |
| | | | LCD_SEG38* | **AO** | LCD segment output 38 at LQFP128 |
| 95 | 50 | | PA.6 | **I/O** | General purpose digital I/O pin |
| | | | AD6 | **AI** | ADC analog input6 |
| | | | EBI_AD7 | **I/O** | EBI Address/Data bus bit7 |
| | | | TC3 | **I** | Timer3 capture input |
| | | | SC2_CLK | **O** | SmartCard2 clock pin(SC2_UART_TXD) |
| | | | PWM0_CH3 | **O** | PWM0 Channel3 output |
| | | | LCD_SEG19* | **AO** | LCD segment output 19 at LQFP64 |
| | | | LCD_SEG37* | **AO** | LCD segment output 37 at LQFP128 |
| 96 | | | PA.7 | **I/O** | General purpose digital I/O pin |
| | | | AD7 | **AI** | ADC analog input7 |
| | | | EBI_AD6 | **I/O** | EBI Address/Data bus bit6 |
| | | | TC2 | **I** | Timer2 capture input |
| | | | SC2_DAT | **I/O** | SmartCard2 DATA pin(SC2_UART_RXD) |
| | | | PWM0_CH2 | **O** | PWM0 Channel2 output |
| | | | LCD_SEG36* | **AO** | LCD segment output 36 output at LQFP128 |
| 97 | 51 | | VREF | **AP** | Voltage reference input for ADC |
| 98 | | | | | NC |
| 99 | 52 | | AVDD | **AP** | Power supply for internal analog circuit |
| 100 | | | PD.0 | **I/O** | General purpose digital I/O pin |
| | | | UART1_RXD | **I** | UART1 Data receiver input pin |
| | | | SPI2_SS0 | **I/O** | SPI2 1$^{st}$ slave select pin |
| | | | SC1_CLK | **O** | SmartCard1 clock pin(SC1_UART_TXD) |
| | | | AD8 | **AI** | ADC analog input8 |
| 101 | | | PD.1 | **I/O** | General purpose digital I/O pin |
| | | | TX1 | **O** | UART1 Data transmitter output pin |
| | | | SPI2_CLK | **I/O** | SPI2 serial clock pin |
| | | | SC1_DAT | **I/O** | SmartCard1 DATA pin(SC1_UART_RXD) |
| | | | AD9 | **AI** | ADC analog input9 |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| 102 | | | PD.2 | **I/O** | General purpose digital I/O pin |
| | | | UART1_RTSn | **O** | UART1 Request to Send output pin |
| | | | I2S_LRCLK | **I/O** | $I^2S$ left right channel clock |
| | | | SPI2_MISO0 | **I/O** | SPI2 1st MISO (Master In, Slave Out) pin |
| | | | SC1_PWR | **O** | SmartCard1 Power pin |
| | | | AD10 | **AI** | ADC analog input10 |
| 103 | | | PD.3 | **I/O** | General purpose digital I/O pin |
| | | | UART1_CTSn | **I** | UART1 Clear to Send input pin |
| | | | I2S_BCLK | **I/O** | $I^2S$ bit clock pin |
| | | | SPI2_MOSI0 | **I/O** | SPI2 1st MOSI (Master Out, Slave In) pin |
| | | | SC1_RST | **O** | SmartCard1 RST pin |
| | | | AD11 | **AI** | ADC analog input11 |
| 104 | | | | | NC |
| 105 | | | PD.4 | **I/O** | General purpose digital I/O pin |
| | | | I2S_DI | **I** | $I^2S$ data input |
| | | | SPI2_MISO1 | **I/O** | SPI2 2nd MISO (Master In, Slave Out) pin |
| | | | SC1_CD | **I** | SmartCard1 card detect |
| | | | LCD_SEG35 | **AO** | LCD segment output 35 at LQFP128 |
| 106 | | | PD.5 | **I/O** | General purpose digital I/O pin |
| | | | I2S_DO | **O** | $I^2S$ data output |
| | | | SPI2_MOSI1 | **I/O** | SPI2 2nd MOSI (Master Out, Slave In) pin |
| | | | LCD_SEG34 | **AO** | LCD segment output 34 at LQFP128 |
| 107 | 53 | | PC.7 | **I/O** | General purpose digital I/O pin |
| | | | DA1_OUT | **AO** | DAC 1 output |
| | | | EBI_AD5 | **I/O** | EBI Address/Data bus bit5 |
| | | | TC1 | **I** | Timer1 capture input |
| | | | PWM0_CH1 | **O** | PWM1 Channel1 output |
| | | | LCD_SEG17* | **AO** | LCD segment output 17 at LQFP64 |
| 108 | 54 | | PC.6 | **I/O** | General purpose digital I/O pin |
| | | | DA0_OUT | **I** | DAC0 output |
| | | | EBI_AD4 | **I/O** | EBI Address/Data bus bit4 |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| | | | TC0 | I | Timer0 capture input |
| | | | SC1_CD | | SmartCard1 card detect pin |
| | | | PWM0_CH0 | O | PWM0 Channel0 output |
| 109 | 55 | | PC.15 | I/O | General purpose digital I/O pin |
| | | | EBI_AD3 | I/O | EBI Address/Data bus bit3 |
| | | | TC0 | I | Timer0 capture input |
| | | | PWM1_CH2 | O | PWM1 Channel1 output |
| | | | LCD_SEG16 | AO | LCD segment output 16 at LQFP64 |
| | | | LCD_SEG33 | AO | LCD segment output 33 at LQFP128 |
| 110 | 56 | | PC.14 | I/O | General purpose digital I/O pin |
| | | | EBI_AD2 | I/O | EBI Address/Data bus bit2 |
| | | | PWM1_CH3 | I/O | PWM1 Channel3 output |
| | | | LCD_SEG15 | AO | LCD segment output 15 at LQFP64 |
| | | | LCD_SEG32 | AO | LCD segment output 32 at LQFP128 |
| 111 | 57 | | PB.15 | I/O | General purpose digital I/O pin |
| | | | INT1 | I | External interrupt1 input pin |
| | | | SNOOPER | I | Snooper pin |
| | | | SC1_CD | I | SmartCard1 card detect |
| | | | LCD_SEG14 | AO | LCD segment output 14 at LQFP64 |
| | | | LCD_SEG31 | AO | LCD segment output 31 at LQFP128 |
| 112 | | | | | NC |
| 113 | 58 | | XT1_IN | O | External 4~24 MHz crystal output pin |
| | | | PF.3 | I/O | General purpose digital I/O pin |
| 114 | 59 | | XT1_OUT | I | External 4~24 MHz crystal input pin |
| | | | PF.2 | I/O | General purpose digital I/O pin |
| 115 | | | | | NC |
| 116 | 60 | | nRESET | I | External reset input: Low active, set this pin low reset chip to initial state. With internal pull-up. |
| 117 | 61 | | VSS | P | Ground |
| 118 | | | VSS | P | Ground |
| 119 | | | | | NC |

| Pin No. | | | Pin Name | Pin Type | Description |
|---|---|---|---|---|---|
| LQFP 128-pin | LQFP 64-pin | LQFP 48-pin | | | |
| 120 | 62 | | VDD | **P** | Power supply for I/O ports and LDO source for internal PLL and digital circuit |
| 121 | | | | | NC |
| 122 | | | PF.4 | **I/O** | General purpose digital I/O pin |
| | | | I2C0_SDA | **I/O** | $I^2C0$ data I/O pin |
| 123 | | | PF.5 | **I/O** | Digital GPI/O pin |
| | | | I2C0_SCL | **I/O** | $I^2C0$ clock pin |
| 124 | | | VSS | **P** | Ground |
| 125 | 63 | | PVSS | **I/O** | PLL Ground |
| 126 | 64 | | PB.8 | **I/O** | General purpose digital I/O pin |
| | | | STADC | **I** | ADC external trigger input. |
| | | | TM0 | **I** | Timer0 external counter input |
| | | | INT0 | **I** | External interrupt0 input pin |
| | | | SC2_PWR | **O** | SmartCard2 Power pin |
| | | | LCD_SEG13 | **AO** | LCD segment output 13 at LQFP64 |
| | | | LCD_SEG30 | **AO** | LCD segment output 30 at LQFP128 |
| 127 | | | PE.15 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG29 | **O** | LCD segment output 29 at LQFP128 |
| 128 | | | PE.14 | **I/O** | General purpose digital I/O pin |
| | | | LCD_SEG28 | **O** | LCD segment output 28 at LQFP128 |

**Note:**

1. Pin Type: I=Digital Input, O=Digital Output; AI=Analog Input; AO=Analog Output; P=Power Pin; AP=Analog Power
2. * : Output voltage for ADC/LCD shared pins cannot be higher than VDD because these pins are without 5V tolerance.

# 4 BLOCK DIAGRAM

## 4.1 Nano100 Block Diagram

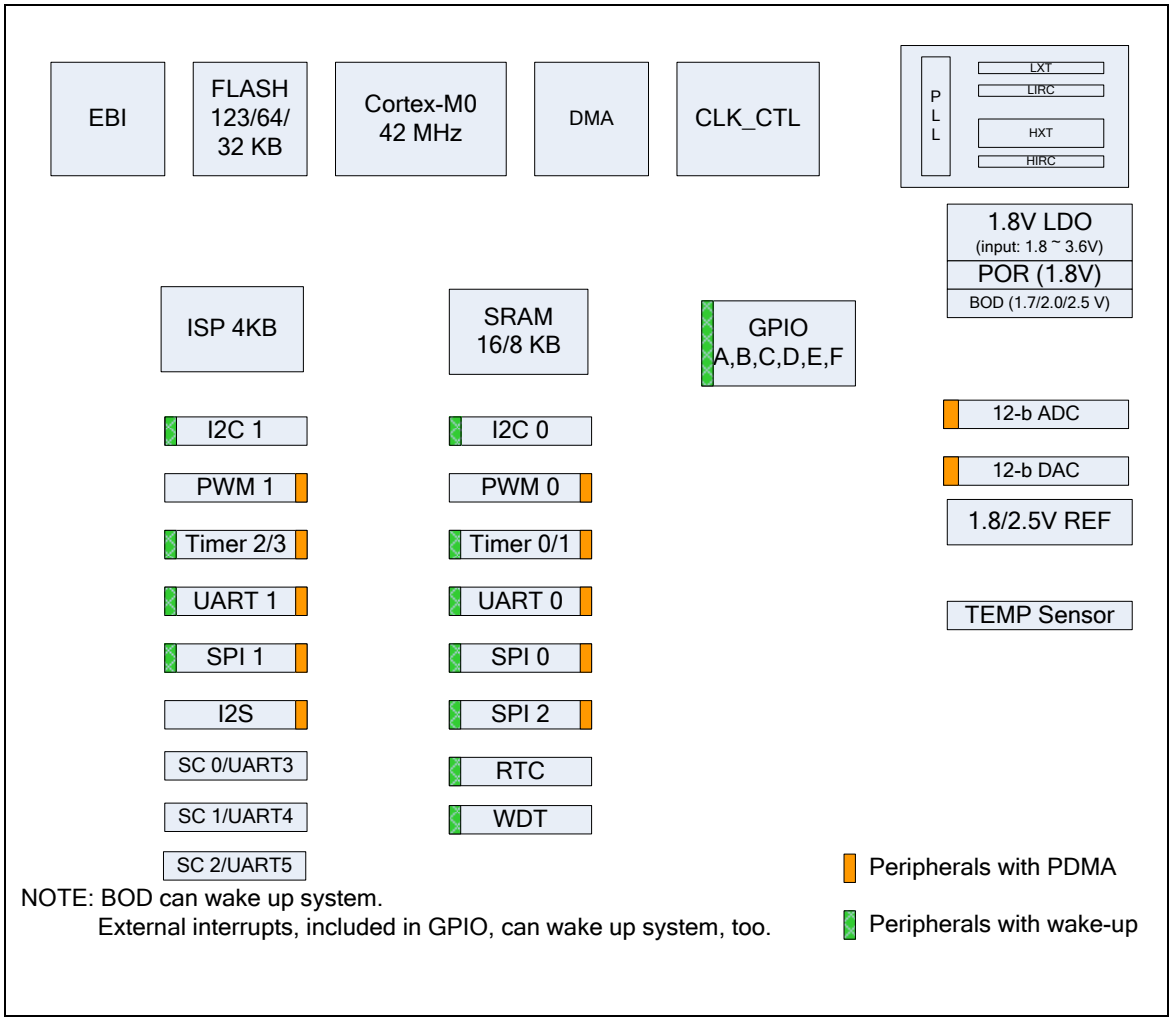



Figure 4-1 NuMicro™ Nano100 Block Diagram

## 4.2 Nano110 Block Diagram

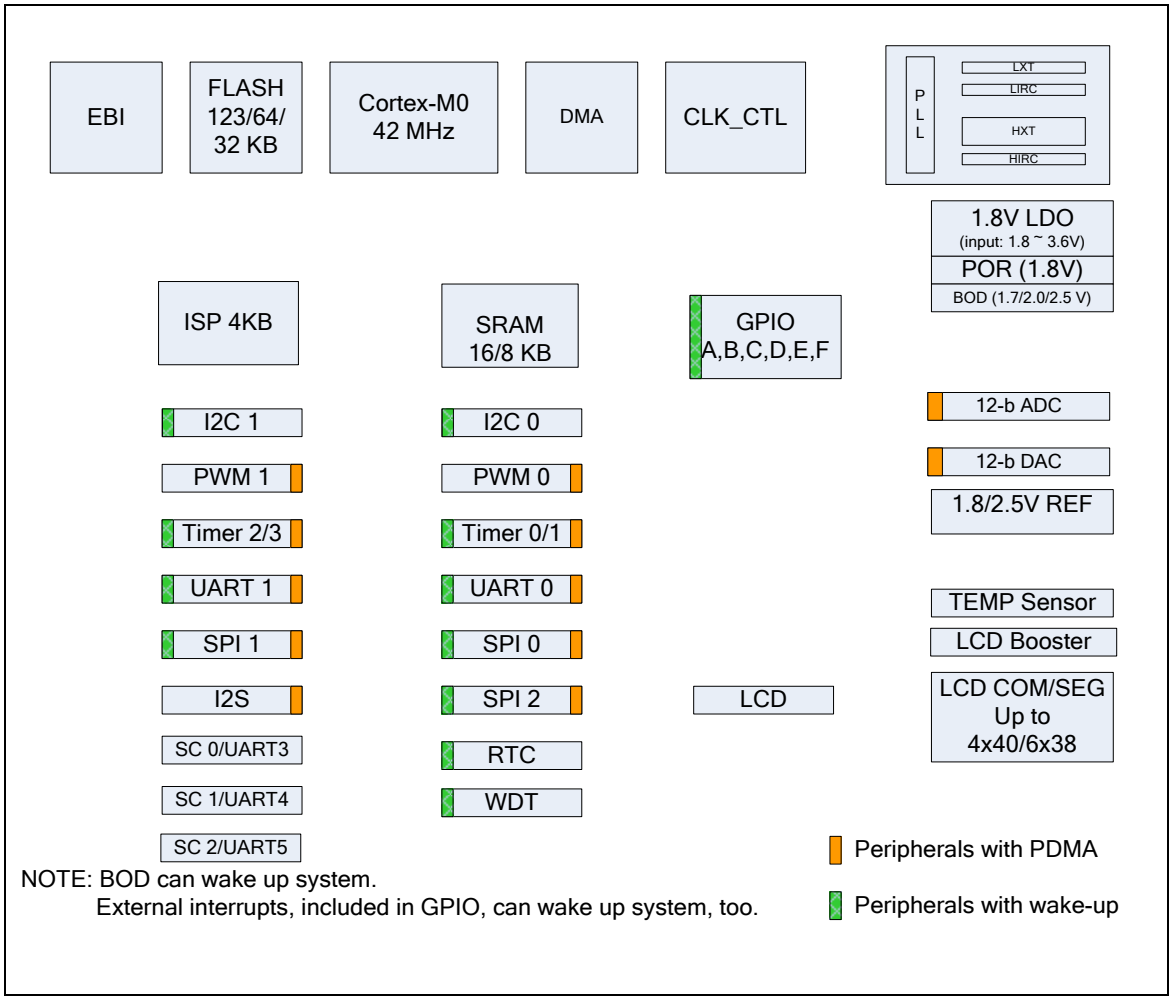



Figure 4-2 NuMicro™ Nano110 Block Diagram

## 4.3 Nano120 Block Diagram

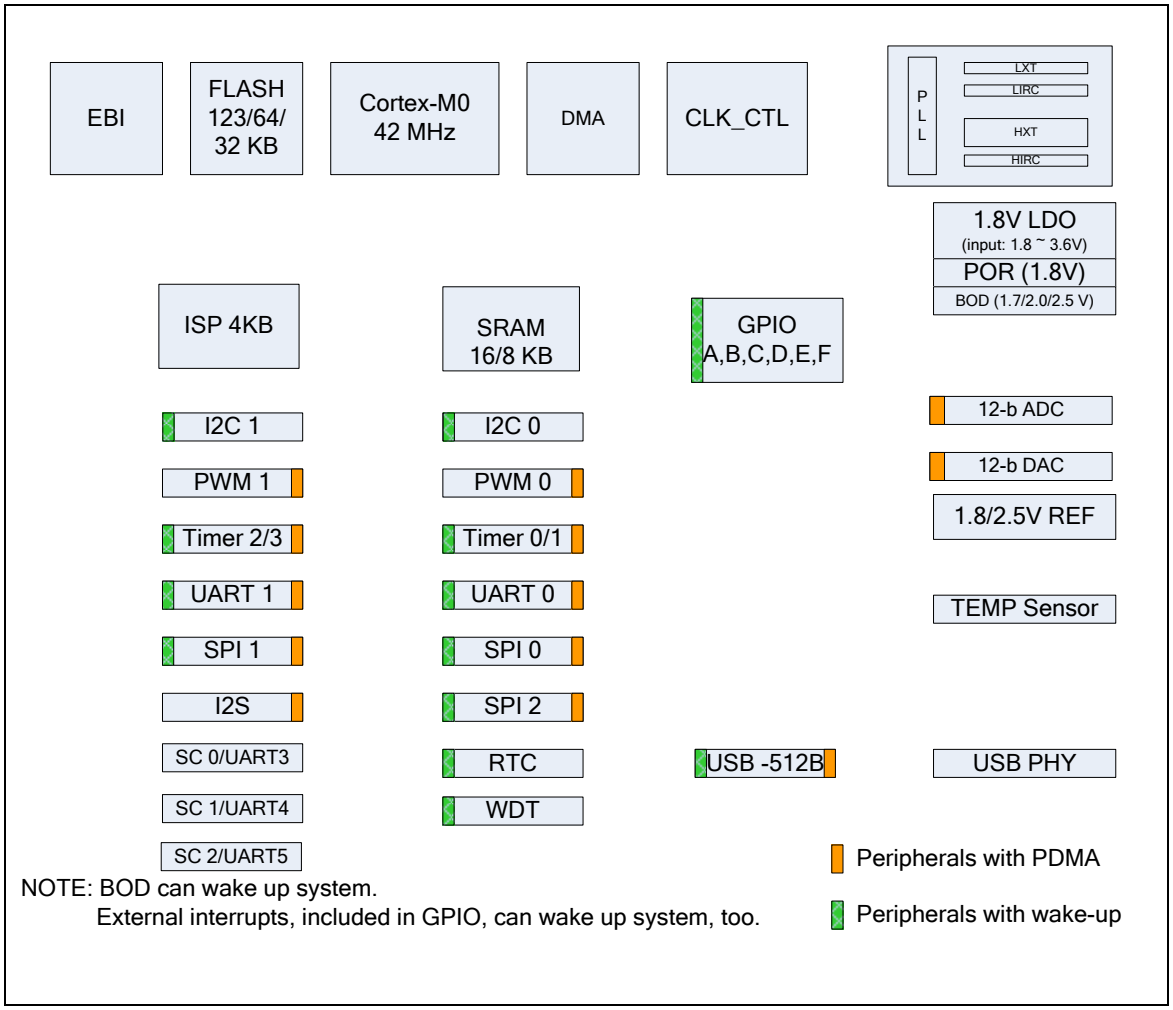



Figure 4-3 NuMicro™ Nano120 Block Diagram

## 4.4 Nano130 Block Diagram

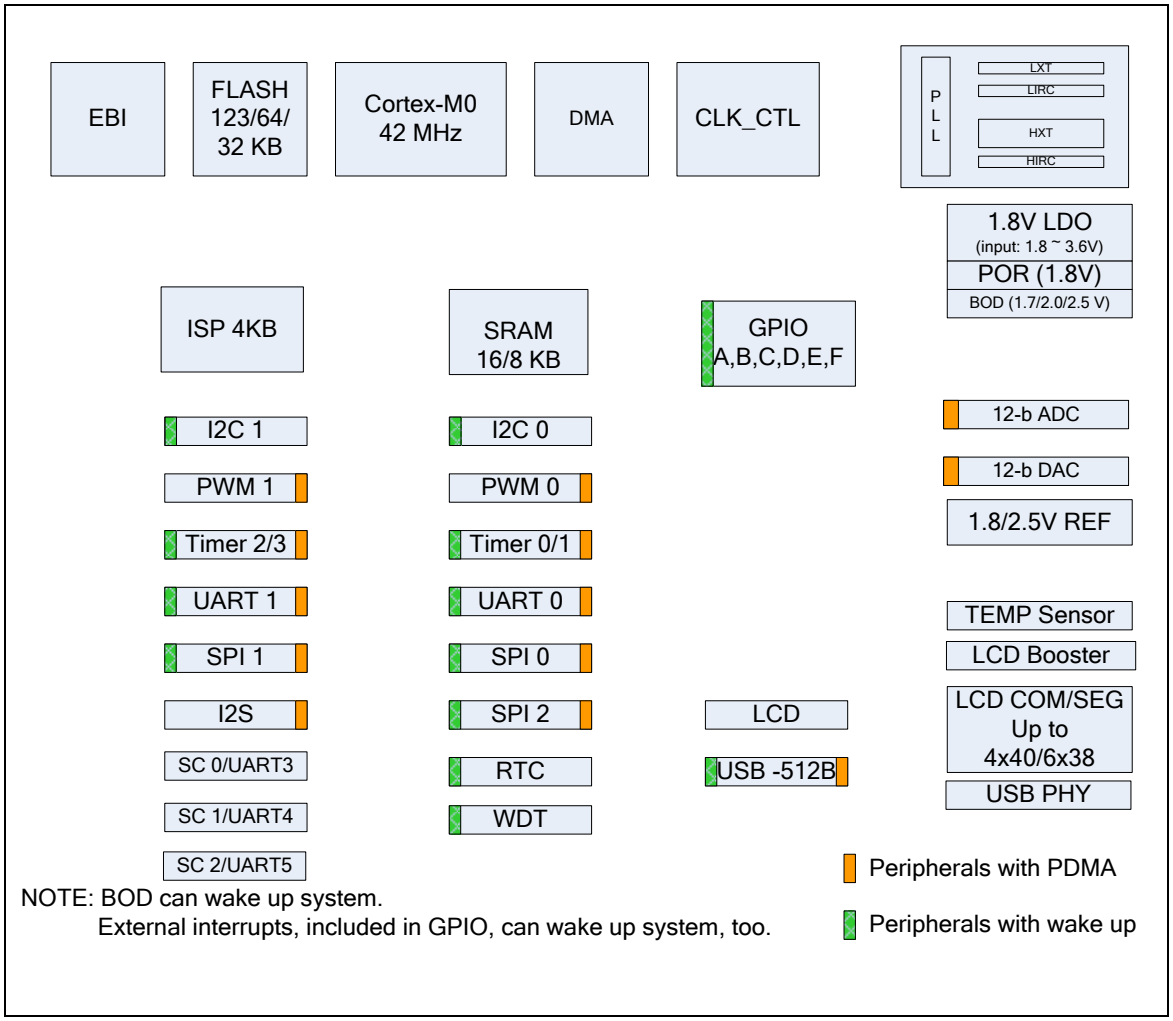



Figure 4-4 NuMicro™ Nano130 Block Diagram

# 5 FUNCTIONAL DESCRIPTION

## 5.1 Memory Organization

### 5.1.1 Overview

The Nano100 provides 4G-byte addressing space. The memory locations assigned to each on-chip modules are shown in following. The detailed register definition, memory space, and programming detailed will be described in the following sections for each on-chip module. The Nano100 series only supports little-endian data format.

### 5.1.2 Memory Map

The memory locations assigned to each on-chip controllers are shown in the following table.

| Address Space | Token | Modules |
|---|---|---|
| **Flash & SRAM Memory Space** | | |
| 0x0000_0000 – 0x0001_FFFF | FLASH_BA | FLASH Memory Space (128KB) |
| 0x2000_0000 – 0x2000_3FFF | SRAM_BA | SRAM Memory Space (16KB) |
| 0x6000_0000 --- 0x6001_FFFF | EXTMEM_BA | External Memory Space(128KB) |
| **AHB Modules Space (0x5000_0000 – 0x501F_FFFF)** | | |
| 0x5000_0000 – 0x5000_01FF | GCR_BA | System Management Control Registers |
| 0x5000_0200 – 0x5000_02FF | CLK_BA | Clock Control Registers |
| 0x5000_0300 – 0x5000_03FF | INT_BA | Interrupt Multiplexer Control Registers |
| 0x5000_4000 – 0x5000_7FFF | GPIO_BA | GPIO Control Registers |
| 0x5000_8000 – 0x5000_BFFF | DMA_BA | DMA Control Registers |
| 0x5000_C000 – 0x5000_FFFF | FMC_BA | Flash Memory Control Registers |
| 0x5001_0000 – 0x5001_03FF | EBI_BA | External Bus Interface Control Registers |
| **APB1 Modules Space (0x4000_0000 ~ 0x400F_FFFF)** | | |
| 0x4000_4000 – 0x4000_7FFF | WDT_BA | Watchdog Timer Control Registers |
| 0x4000_8000 – 0x4000_BFFF | RTC_BA | Real Time Clock (RTC) Control Register |
| 0x4001_0000 – 0x4001_3FFF | TMR01_BA | Timer0 and Timer1 Control Registers |
| 0x4002_0000 – 0x4002_3FFF | I2C0_BA | $I^2C0$ Interface Control Registers |
| 0x4003_0000 – 0x4003_3FFF | SPI0_BA | SPI0 with Master/Slave function Control Registers |
| 0x4004_0000 – 0x4004_3FFF | PWM0_BA | PWM0 Control Registers |
| 0x4005_0000 – 0x4005_3FFF | UART0_BA | UART0 Control Registers |
| 0x4006_0000 – 0x4006_3FFF | USBD_BA | USB FS device Controller Registers |
| 0x400A_0000 – 0x400A_3FFF | DAC_BA | Digital-Analog-Converter (DAC) Control Registers |
| 0x400B_0000 – 0x400B_3FFF | LCD_BA | LCD Control Registers |

| | | |
|---|---|---|
| 0x400D_0000 – 0x400D_3FFF | SPI2_BA | SPI2 with Master/Slave function Control Registers |
| 0x400E_0000 – 0x400E_3FFF | ADC12_BA | 12-bit Analog-Digital-Converter (ADC12) Control Registers |
| **APB2 Modules Space (0x4010_0000 ~ 0x401F_FFFF)** | | |
| 0x4011_0000 – 0x4011_3FFF | TMR23_BA | Timer2 and Timer3 Control Registers |
| 0x4012_0000 – 0x4012_3FFF | I2C1_BA | $I^2C1$ Interface Control Registers |
| 0x4013_0000 – 0x4013_3FFF | SPI1_BA | SPI1 with Master/Slave function Control Registers |
| 0x4014_0000 – 0x4014_3FFF | PWM1_BA | PWM1 Control Registers |
| 0x4015_0000 – 0x4015_3FFF | UART1_BA | UART1 Control Registers |
| 0x4019_0000 – 0x4019_3FFF | SC0_BA | SmartCard0 Control Registers |
| 0x401A_0000 – 0x401A_3FFF | I2S_BA | $I^2S$ Control Registers |
| 0x401B_0000 – 0x401B_3FFF | SC1_BA | SmartCard1 Control Registers |
| 0x401C_0000 – 0x401C_3FFF | SC2_BA | SmartCard2 Control Registers |
| **System Control Space (0xE000_E000 ~ 0xE000_EFFF)** | | |
| 0xE000_E010 – 0xE000_E0FF | SCS_BA | System Timer Control Registers |
| 0xE000_E100 – 0xE000_ECFF | SCS_BA | External Interrupt Controller Control Registers |
| 0xE000_ED00 – 0xE000_ED8F | SCS_BA | System Control Registers |

## 5.2 Nested Vectored Interrupt Controller (NVIC)

### 5.2.1 Overview

The Cortex-M0 provides an interrupt controller as an integral part of the exception mode, named as “Nested Vectored Interrupt Controller (NVIC)”. It is closely coupled to the processor kernel and provides following features:

### 5.2.2 Features

- Nested and Vectored interrupt support
- Automatic processor state saving and restoration
- Dynamic priority changing
- Reduced and deterministic interrupt latency

The NVIC prioritizes and handles all supported exceptions. All exceptions are handled in “Handler Mode”. This NVIC architecture supports 32 (IRQ[31:0]) discrete interrupts with 4 levels of priority. All of the interrupts and most of the system exceptions can be configured to different priority levels. When an interrupt occurs, the NVIC will compare the priority of the new interrupt to the current running one’s priority. If the priority of the new interrupt is higher than the current one, the new interrupt handler will override the current handler.

When any interrupts is accepted, the starting address of the interrupt service routine (ISR) is fetched from a vector table in memory. There is no need to determine which interrupt is accepted and branch to the starting address of the correlated ISR by software. While the starting address is fetched, NVIC will also automatically save processor state including the registers “PC, PSR, LR, R0~R3, R12” to the stack. At the end of the ISR, the NVIC will restore the mentioned registers

from stack and resume the normal execution. Thus it will take less and deterministic time to process the interrupt request.

The NVIC supports “Tail Chaining” which handles back-to-back interrupts efficiently without the overhead of states saving and restoration and therefore reduces delay time in switching to pending ISR at the end of current ISR. The NVIC also supports “Late Arrival” which improves the efficiency of concurrent ISRs. When a higher priority interrupt request occurs before the current ISR starts to execute (at the stage of state saving and starting address fetching), the NVIC will give priority to the higher one without delay penalty. Thus it advances the real-time capability.

For more detailed information, please refer to the “ARM® Cortex™-M0 Technical Reference Manual” and “ARM® v6-M Architecture Reference Manual”.

### 5.2.3 Exception Model and System Interrupt Map

The following table lists the exception model supported by Nano100 serials. Software can set four levels of priority on some of these exceptions as well as on all interrupts. The highest user-configurable priority is denoted as “0” and the lowest priority is denoted as “3”. The default priority of all the user-configurable interrupts is “0”. Note that priority “0” is treated as the fourth priority on the system, after three system exceptions “Reset”, “NMI” and “Hard Fault”.

| Exception Name | Vector Number | Priority |
|---|---|---|
| Reset | 1 | -3 |
| NMI | 2 | -2 |
| Hard Fault | 3 | -1 |
| Reserved | 4 ~ 10 | Reserved |
| SVCall | 11 | Configurable |
| Reserved | 12 ~ 13 | Reserved |
| PendSV | 14 | Configurable |
| SysTick | 15 | Configurable |
| Interrupt (IRQ0 ~ IRQ31) | 16 ~ 47 | Configurable |

Table 5-1 Exception Model

| Vector Number | Interrupt Number (Bit in Interrupt Registers) | Interrupt Name | Source IP | Interrupt Description |
|---|---|---|---|---|
| 0 ~ 15 | - | - | - | System exceptions |
| 16 | 0 | BOD_INT | Brown-out | Brown-out low voltage detected interrupt |
| 17 | 1 | WDT_INT | WDT | Watchdog Timer interrupt |
| 18 | 2 | EINT0 | GPIO | External signal interrupt from PB.14 pin |
| 19 | 3 | EINT1 | GPIO | External signal interrupt from PB.15 pin |
| 20 | 4 | GPABC_INT | GPIO | External signal interrupt |

| Vector Number | Interrupt Number (Bit in Interrupt Registers) | Interrupt Name | Source IP | Interrupt Description |
|---|---|---|---|---|
| | | | | from PA[15:0] / PB[13:0]/PC[15:0] |
| 21 | 5 | GPDEF_INT | GPIO | External interrupt from PD[15:0]/PE[15:0]/PF[5:0] |
| 22 | 6 | PWM0_INT | PWM0 | PWM0 interrupt |
| 23 | 7 | PWM1_INT | PWM1 | PWM1interrupt |
| 24 | 8 | TMR0_INT | TMR0 | Timer0 interrupt |
| 25 | 9 | TMR1_INT | TMR1 | Timer1 interrupt |
| 26 | 10 | TMR2_INT | TMR2 | Timer2 interrupt |
| 27 | 11 | TMR3_INT | TMR3 | Timer3 interrupt |
| 28 | 12 | UART0_INT | UART0 | UART0 interrupt |
| 29 | 13 | UART1_INT | UART1 | UART1 interrupt |
| 30 | 14 | SPI0_INT | SPI0 | SPI0 interrupt |
| 31 | 15 | SPI1_INT | SPI1 | SPI1 interrupt |
| 32 | 16 | SPI2_INT | SPI2 | SPI2 interrupt |
| 33 | 17 | IRC_INT | IRC | IRC TRIM interrupt |
| 34 | 18 | I2C0_INT | I2C0 | $I^2C$0 interrupt |
| 35 | 19 | I2C1_INT | I2C1 | $I^2C$1 interrupt |
| 36 | 20 | SC2_INT | SC2 | Smart Card2 interrupt |
| 37 | 21 | SC0_INT | SC0 | Smart Card0 interrupt |
| 38 | 22 | SC1_INT | SC1 | Smart Card1 interrupt |
| 39 | 23 | USB_INT | USBD | USB FS Device interrupt |
| 41 | 25 | LCD_INT | LCD | LCD interrupt |
| 42 | 26 | DMA_INT | DMA | DMA interrupt |
| 43 | 27 | I2S_INT | $I^2S$ | $I^2S$ interrupt |
| 44 | 28 | PD_WU_INT | CLKC | Clock controller interrupt for chip wake-up from power-down state |
| 45 | 29 | ADC_INT | ADC | ADC interrupt |
| 46 | 30 | DAC_INT | DAC | DAC interrupt |
| 47 | 31 | RTC_INT | RTC | Real time clock interrupt |

Table 5-2 System Interrupt Map

### 5.2.4 Vector Table

When any interrupts is accepted, the processor will automatically fetch the starting address of the interrupt service routine (ISR) from a vector table in memory. For ARMv6-M, the vector table base address is fixed at 0x00000000. The vector table contains the initialization value for the stack pointer on reset, and the entry point addresses for all exception handlers. The vector number on previous page defines the order of entries in the vector table associated with exception handler entry as illustrated in previous section.

| Vector Table Word Offset | Description |
|---|---|
| 0 | SP_main – The Main stack pointer |
| Vector Number | Exception Entry Pointer using that Vector Number |

Table 5-3 Vector Table Format

### 5.2.5 Operation Description

The NVIC interrupts can be enabled and disabled by writing to their corresponding Interrupt Set-Enable or Interrupt Clear-Enable register bit-field. The registers use a write-1-to-enable and write-1-to-clear policy, both registers reading back the current enabled state of the corresponding interrupts. When an interrupt is disabled, interrupt assertion will cause the interrupt to become Pending, however, the interrupt will not activate. If an interrupt is Active when it is disabled, it remains in its Active state until cleared by reset or an exception return. Clearing the enable bit prevents new activations of the associated interrupt.

The NVIC interrupts can be pended/un-pended using a complementary pair of registers to those used to enable/disable the interrupts, named the Set-Pending Register and Clear-Pending Register respectively. The registers use a write-1-to-enable and write-1-to-clear policy, both registers reading back the current pended state of the corresponding interrupts. The Clear-Pending Register has no effect on the execution status of an Active interrupt.

The NVIC interrupts are prioritized by updating an 8-bit field within a 32-bit register (each register supporting four interrupts).

The general registers associated with the NVIC are all accessible from a block of memory in the System Control Space and will be described in next section.

## 5.3 System Manager

### 5.3.1 Overview

System manager mainly controls the power modes, wake-up source, system resets and system memory map. It also provides information about product ID, chip reset, IP reset, and multi-function pin control.

### 5.3.2 Features

- Power modes and wake-up sources
- System resets
- System Memory Map
- System manager registers for :
  - Product ID
  - Chip and IP reset

- Multi-functional pin control

### 5.3.3 Functional Description

5.3.3.1 Power modes and Wake-up sources

There are several power modes in this chip, depending on the clock status (ON or OFF).

**Clocks:**

- External 32.768 kHz Low Speed Crystal (LXT)
- External 4~ 24 MHz High Speed Crystal (HXT)
- Internal RC 12 MHz High Speed Oscillator Clock (HIRC)
- Internal 10 kHz Low Speed Oscillator Clock (LIRC)

**Power Modes:**

- Normal mode: CPU runs normally and all clocks ON.
- Idle mode: CPU entered sleep mode. CPU clock stops and other clocks ON.
- Power-down mode: All clocks stop, except LXT and LIRC, and SRAM retention

After chip enters power down, the following wake-up sources can wake chip up to Normal mode and list the condition about how to enter pown-down mode again for each peripheral.

| Wake-up Source | Wake-up condition | System can enter Power-down mode: |
|---|---|---|
| External Interrupts | - | After software writes 1 to clear the GPIOx_ISRC bit. |
| UART | Data wake-up | Immediately after wake-up. |
| | CTSn wake-up | Requiring 2 UART_CLK after wake-up. |
| GPIO | - | After Software writes 1 to clear the GPIOx_ISRC bit. |
| RTC | - | Requiring 1 RTC_CLK (about 30 us) after wake-up. |
| USB | - | Immediately after wake up |
| SPI | - | After SPI slave clock goes from high to low. |
| Timer | TMRx_ISR[TCAP_IS] | After software sets timer TMRx_ISR[SW_RST] (software reset) or software writes 1 to clear TMRx_ISR[TCAP_IS]. |
| | TMRx_ISR[TMR_IS] | After software sets timer TMRx_ISR[SW_RST] (software reset) or software writes 1 to clear TMRx_ISR[TMR_IS]. |
| WDT | - | Immediately after wake-up. |
| BOD | - | After voltage is raised higher than target voltage or BOFx_INT_EN is set to low. |
| I2C | - | Immediately after wake-up. |

Table 5-4 Condition of Entering Power-down Mode Again Table

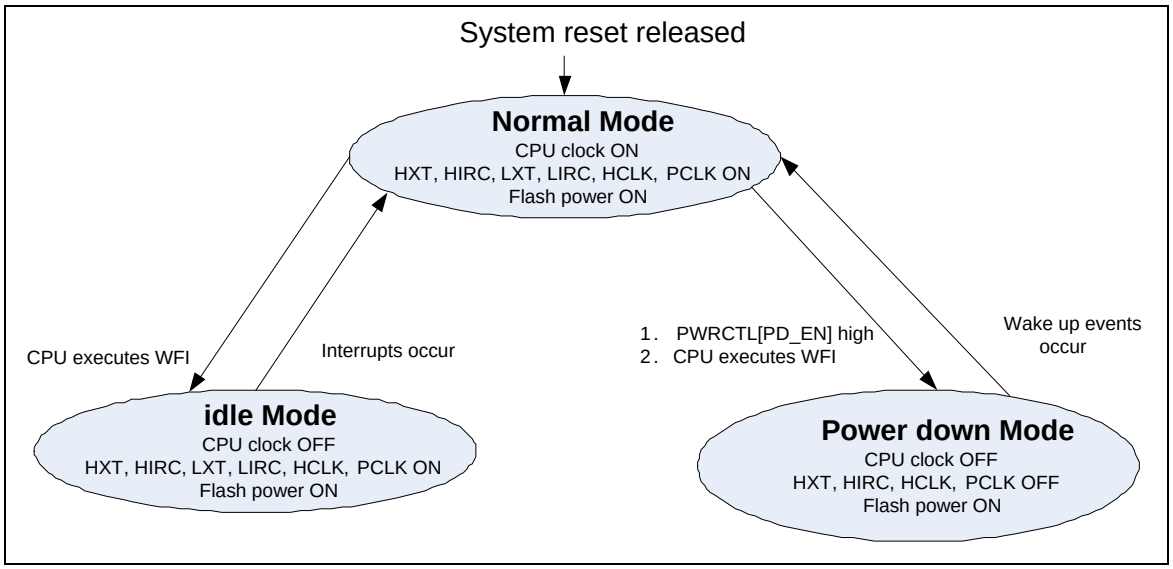



Figure 5-1 Power Modes

| Register | Normal Mode | Idle Mode | Power-down Mode |
|---|---|---|---|
| Wake-up time to Normal mode | - | - | 7us: from wake-up event to first CPU core valid clock<br>10us: from interrupt event to interrupt service routine first instruction. |
| HXT (4~24 MHz XTL) | ON | ON | - |
| HIRC (12 MHz OSC) | ON | ON | - |
| LXT (32 kHz XTL) | ON | ON | ON/OFF[1] |
| LIRC (10 kHz OSC) | ON | ON | ON/OFF[2] |
| PLL | ON | ON | - |
| LDO | ON | ON | ON |
| CPU | ON | - | - |
| HCLK/PCLK | ON | ON | - |
| SRAM retention | ON | ON | ON |
| FLASH[3] | ON | ON | ON |
| GPIO | ON | ON | - |
| DMA | ON | ON | - |
| I2C | ON | ON | - |
| PWM4 | ON | ON | -- |
| TIMER[5] | ON | ON | -- |

| | | | |
|---|---|---|---|
| UART | ON | ON | - |
| SPI | ON | ON | - |
| RTC | ON | ON | ON/OFF[4] |
| WDT | ON | ON | ON/OFF[5] |
| USB | ON | ON | - |
| LCD | ON | ON | ON/OFF[6] |
| I2S | ON | ON | - |
| ADC | ON | ON | - |
| DAC | ON | ON | - |

Table 5-5 IP clock ON/OFF in Power Modes

1. LXT (32 kHz XTL) ON or OFF is depended on SW setting in run mode
2. LIRC (10 kHz OSC) ON or OFF is depended on S/W setting in run mode
3. RTC can work if 32 kHz XTL is ON
4. WDT can work if 10 k Hz OSC is ON.
5. LCD can work if 32 kHz XTL is ON.

Note: If CPU clock is not from HIRC (12 MHz RC oscillator), users must enable HIRC before entering Power-down mode to avoid wake-up fail. The power-down circuit will automatically disable HIRC in Power-down mode to save power consumption. Users can disable HIRC after wake-up to Normal mode to reduce extra current consumption if HIRC is not needed anymore in Normal mode.

#### 5.3.3.2 System Resets

The system reset includes the events listed bellow. These reset events can be read by the RST_SRC register.

- Power-on Reset (POR)
- Brown-out Reset (BOD)
- Low level on the nRESET pin
- Watchdog Timer Time-out Reset
- Cortex-M0 MCU Reset

#### 5.3.3.3 Auto-trim

This chip supports auto-trim function: the HIRC trim (12 MHz RC oscillator), according to the accurate LXT (32.768 kHz crystal oscillator), automatically gets accurate HIRC output frequency, 0.25% deviation within all temperature ranges. For instance, PLL needs an accurate 12 MHz input clock to output 48 MHz clock in USB applications. In such case, if users do not want to use 12 MHz HXT as PLL input clock, they need to solder 32.768 kHz crystal in system, and set the HIRC target output clock to 12 MHz by setting HIRCTRIMCTL[TRIM_SEL] to “10”, and the auto-trim function will be enabled. It is recommended to set both TRIM_LOOP and TRIM_RETRY_CNT to “11” to get better results.

## 5.4 Clock Controller

### 5.4.1 Overview

The clock controller generates clocks for the whole chip, Iincluding system clocks (CPU clock, HCLKx, and PCLKx) and all peripheral engine clocks. HCLKx means AHB bus clock for peripherals on AHB bus. PCLKx means APB bus clock for peripherals on APB bus. The clock controller also implements the power control function with the individually clock ON/OFF control, clock source selection and a 4-bit clock divider. The chip will not enter power-down mode until CPU sets the power down enable bit (PD_EN) and CPU executes the WFI instruction. In the Power-down mode, clock controller turns off the external high frequency crystal, internal high frequency oscillator, and system clocks (CPU clock, HCLKx, and PCLKx) to reduce the power consumption to minimum.

### 5.4.2 Features

- Generates clocks for system clocks and all peripheral engine clocks.
- Each peripheral engine clock can be turned on/off.
- High frequency crystal, internal high frequency oscillator, and system clocks will be turned off when chip is in Power-down mode.

### 5.4.3 Block Diagram

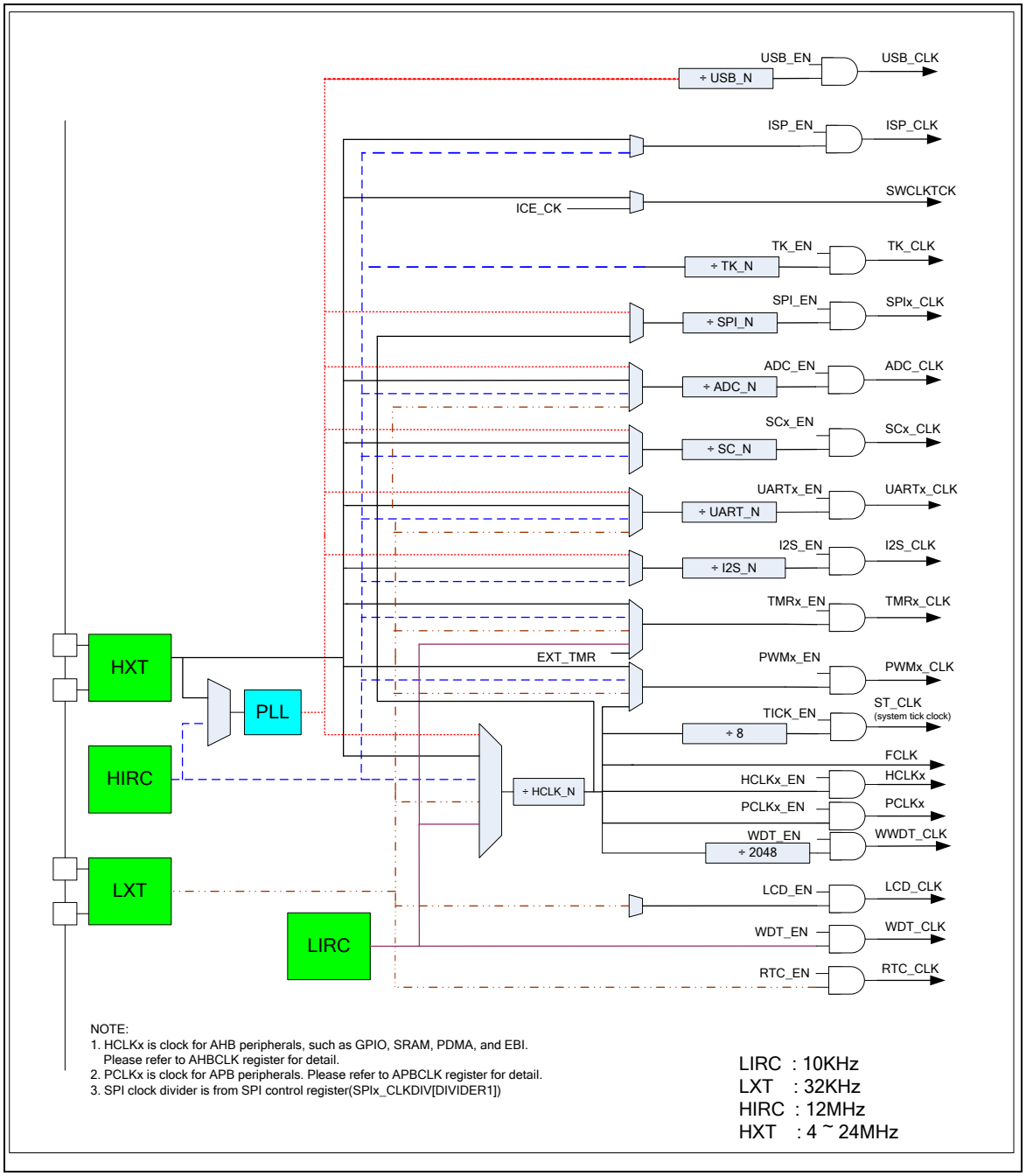



Figure 5-2 Clock Controller Block Diagram

### 5.4.4 Functional Description

5.4.4.1 The clock controller consists of 5 sources as listed below:

- 32768 low speed external crystal (LXT)
- 4~ 24 MHz high speed external crystal (HXT)

- One programmable PLL FOUT (PLL source consists of HXT and HIRC)
- 12 MHz high speed internal RC oscillator (HIRC)
- 10 kHz low speed internal RC oscillator (LIRC)

#### 5.4.4.2 Peripheral engine clocks

Each peripheral engine clock has different Clock Source switching setting. Please refer to the CLKSEL1 and CLKSEL2 description.

| | HXT | LXT | HIRC | LIRC | PCLK/HCLK | PLL | Ext. Pin |
|---|---|---|---|---|---|---|---|
| LCD | - | Yes | - | Yes | - | - | - |
| TM | Yes | Yes | Yes | Yes | - | - | Yes |
| PWM | Yes | Yes | Yes | - | - | Yes | - |
| ADC | Yes | Yes | Yes | - | - | Yes | - |
| UART | Yes | Yes | Yes | - | - | Yes | - |
| SC | Yes | - | Yes | - | - | Yes | - |
| $I^2S$ | Yes | - | Yes | - | - | Yes | - |
| SPI | - | - | Yes | - | - | Yes | - |
| $I^2C$ | - | - | - | - | Yes | - | - |
| USB | - | - | - | - | - | Yes | - |

Table 5-6 Peripheral Engine Clocks

#### 5.4.4.3 Clocks in Power-down Mode

When chip enters Power-down mode, system clocks (CPU clock, HCLKx, and PCLKx), HXT, HIRC will be disabled directly. LXT and LIRC could be still active in Power-down mode if CPU does not disable these clocks before entering Power-down mode. IP engine clock could be still active in Power-down mode if IP adopts LXT or LIRC and LXT or LIRC does not be disabled respectively.

#### 5.4.4.4 Frequency Divider Output

This device is equipped a power-of-2 frequency divider which is composed by16 chained divide-by-2 shift registers. One of the 16 shift register outputs selected by a sixteen to one multiplexer is reflected to GPIOB.12/GPIOF.1. Therefore there are 16 options of power-of-2 divided clocks with the frequency from Fin/2^1 to Fin/2^16 where Fin is input clock frequency to the clock divider.

The output formula is Fout = Fin/2^(N+1), where Fin is the input clock frequency, Fout is the clock divider output frequency and N is the 4-bit value in FSEL (FRQDIV[3:0]).

When FDIV_EN (FRQDIV[4]) is set to high, the rising transition of FDIV_EN will reset the chained counter and then chained counter starts to count. When FDIV_EN (FRQDIV[4]) is written with zero, the chained counter continuously runs till divided clock reaches low state and stay in low state.

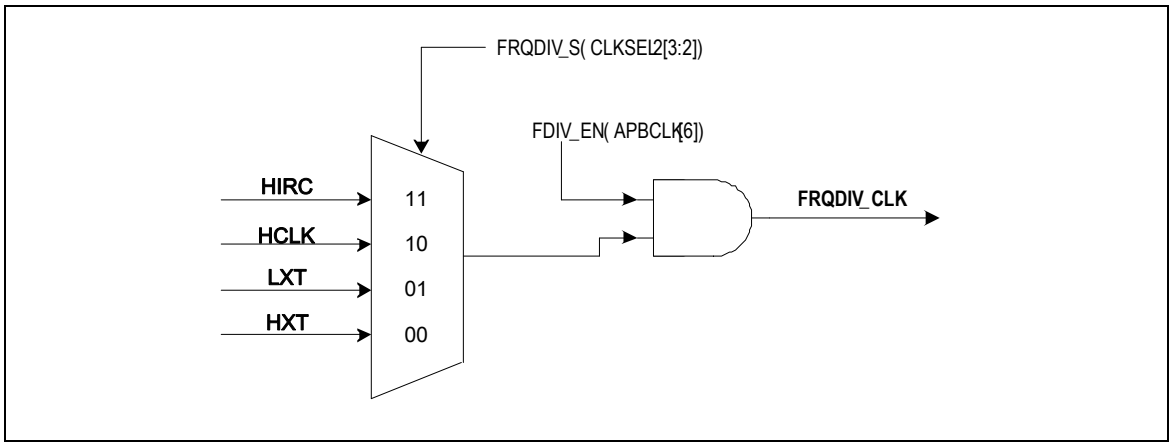



Figure 5-3 Clock Sources of Frequency Divider

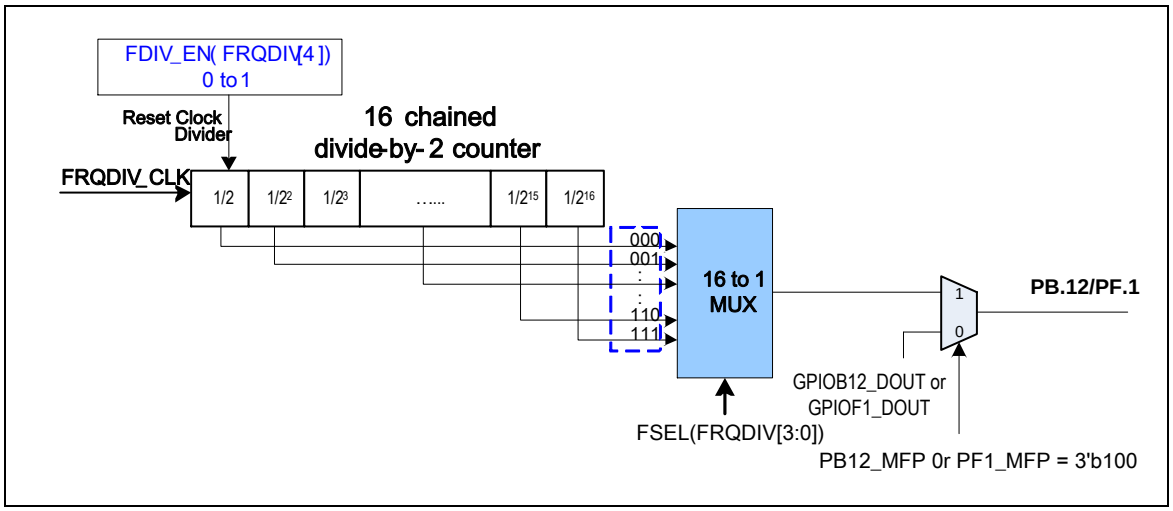



Figure 5-4 Frequency Divider Block Diagram

## 5.5 Analog to Digital Converter (ADC)

### 5.5.1 Overview

This chip contains one 12-bit successive approximation analog-to-digital converter (SAR A/D converter) with 12 external input channels and 6 internal channels. The A/D converter supports three operation modes: Single, Single-cycle Scan and Continuous Scan mode, and can be started by software and external STADC/PB.8 pin and timer event start.

Note that the I/O pins used as ADC analog input pins must be configured as input type and off digital function (GPIOA_OFFD) should be turned on before ADC function is enabled.

### 5.5.2 Features

- Analog input voltage range: 0~Vref (Max to 3.6V)
- Selectable 12-bits, 10-bits, 8-bits and 6-bits resolution
- Supports sampling time settings (in ADC_CLK unit) for channel 0~11 individually and channel 12~17 share the same one sampling time setting
- Supports two power-down modes:
  - Power-down mode
  - Standby mode
- Up to 12 external analog input channels (channel0 ~ channel11), and 6 internal channels (channel12~channel17) converting six voltage sources, including DAC0, DAC1, internal band-gap voltage, internal temperature sensor output, AVDD, and AVSS.
- Maximum ADC clock frequency is 42 MHz and each conversion is 19 clocks+ sampling time depending on the input resistance.
- Three operating modes
  - Single mode: A/D conversion is performed one time on a specified channel.
  - Single-cycle Scan mode: A/D conversion is performed one cycle on all specified channels with the sequence from the lowest numbered channel to the highest numbered channel.
  - Continuous Scan mode: A/D converter continuously performs Single-cycle scan mode until software stops A/D conversion.
- An A/D conversion can be started by:
  - Software write 1 to ADST bit
  - External pin STADC
  - Selects one from four timer events (TMR0, TMR1, TMR2 and TMR3) that enable ADC and transfer AD results by PDMA
- Conversion results held in data registers for each channel
- Conversion result can be compared with a specified value and user can select whether to generate an interrupt when conversion result is equal to the compare register setting.
- Supports Calibration and load Calibration words capability.

### 5.5.3 Block Diagram

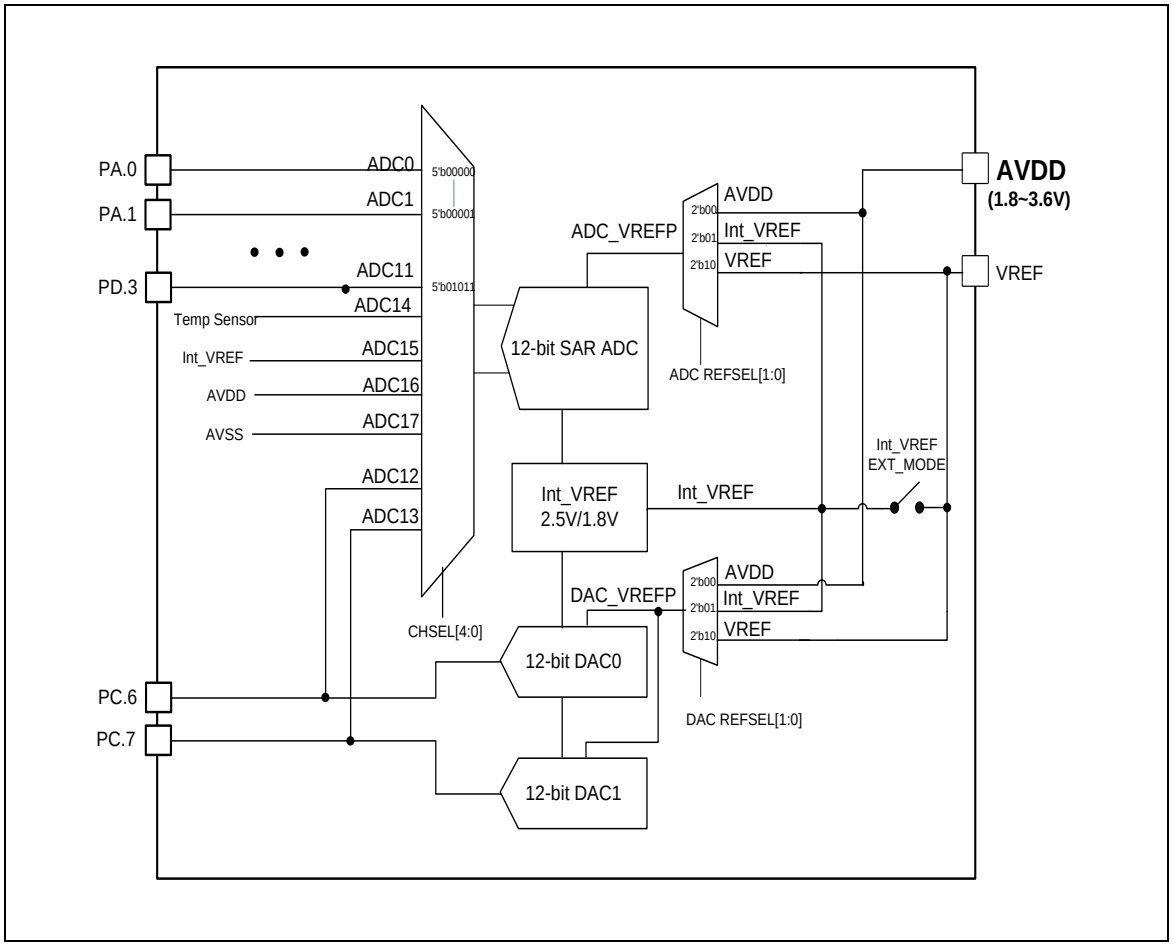



Figure 5-5 ADC and DAC Block Diagram

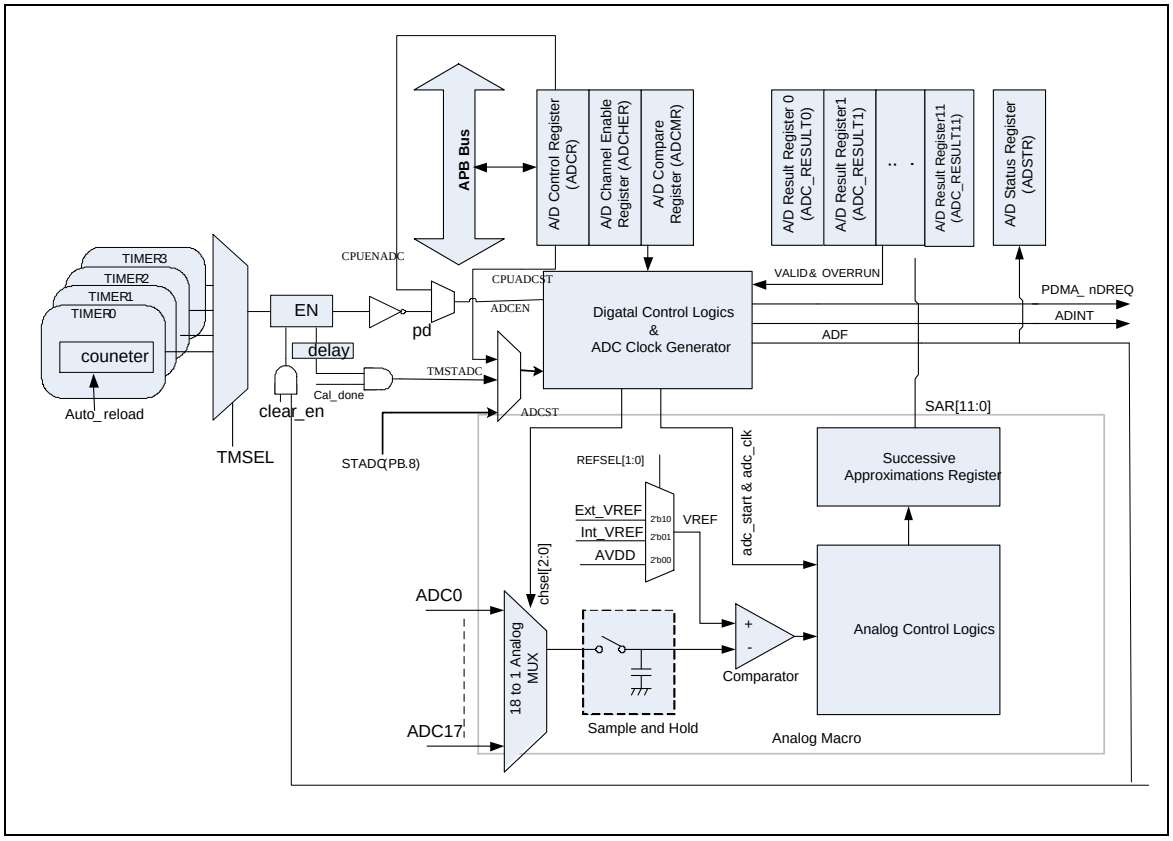



Figure 5-6 ADC Controller Block Diagram

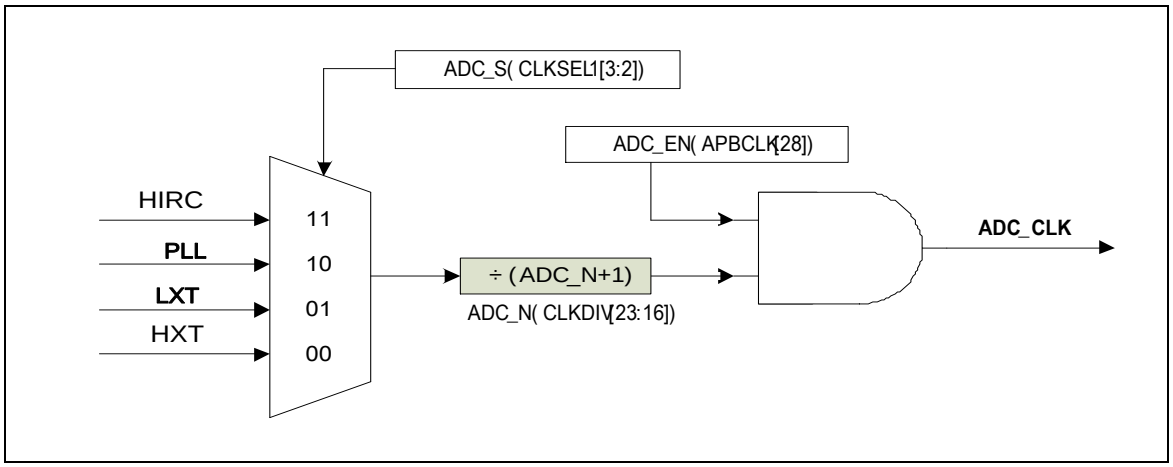



Figure 5-7 ADC Clock Control

### 5.5.4 Functional Description

The A/D converter is operated by successive approximation with 12-bit resolution. The ADC has three operation modes: Single mode, Single-cycle Scan mode and Continuous Scan mode. When changing the operating mode or analog input channel enabled, in order to avoid incorrect

operation, software must clear the ADST bit to 0 in ADCR register. After the operation, the A/D converter discards current conversion and enters idle state while ADST bit is cleared.

In some applications for saving power, ADC can be enabled by a time-out (TMRx Chy) signal and start A/D conversion after a delay time interval and enter power-down state after converting fixed amount of conversion data transferred to memory through PDMA, There are four time-out source(Timer0~3) to enable ADC by setting TMSEL[1:0] in ADCON register.

#### 5.5.4.1 Single Mode

In single mode, A/D conversion is performed only once on the specified single channel. The operations are as follows.

A/D conversion is started when the ADST bit in ADCR is set to 1 either by software or by external trigger input or by timer event selected by TMSEL[1:0] in ADCR register .

When A/D conversion is finished, the result is stored in the A/D data register corresponding to the channel.

On completion of conversion, the ADF bit in ADSR is set to 1 and ADC interrupt (ADINT) is requested if the ADIE bit is set to 1.

The ADST bit remains 1 during A/D conversion. When A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters in idle state. If the ADST bit is cleared to 0 by software during A/D conversion, A/D conversion will stop and enter in idle state.

After the previous conversion is complete, repeat method 1-4 to the next conversion PS: if the ADC clock is much lower than PCLK, wait for at least one ADC clock to start the next conversion

**Note:** If software enables more than one channel in single mode, the least channel is converted and other enabled channels will be ignored.

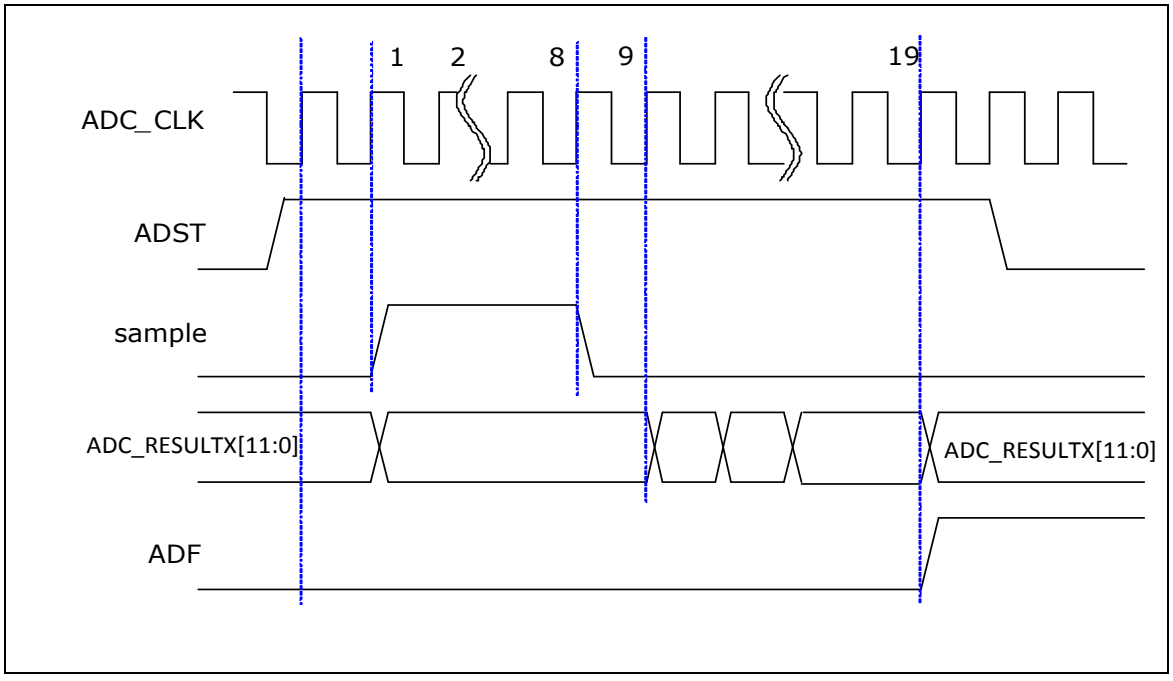



Figure 5-8 ADC Single Mode Conversion Timing Diagram

#### 5.5.4.2 Single-Cycle Scan Mode

In single-cycle scan mode, A/D conversion will sample and convert the specified channels once in the sequence from the least numbered channel to the highest numbered channel. Operations are described as follows.

When the ADST bit in ADCR is set to 1 by software or by an external trigger input or by timer event selected by TMSEL[1:0] in ADCR register, A/D conversion starts on the lowest numbered channel.

When A/D conversion for each enabled channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.

When conversions of all the enabled channels are completed, the ADF bit in ADSR is set to 1. If the ADIE bit is set to 1 at this time, an ADINT interrupt is set after A/D conversion ends.

After A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters in idle state. If the ADST bit is cleared to 0 by software during A/D conversion, A/D conversion will stop after current conversion complete and enter in idle state.

An example timing diagram for single-cycle scan is shown below:

(In this example, channel 0,2,3 and 7 are enabled.)

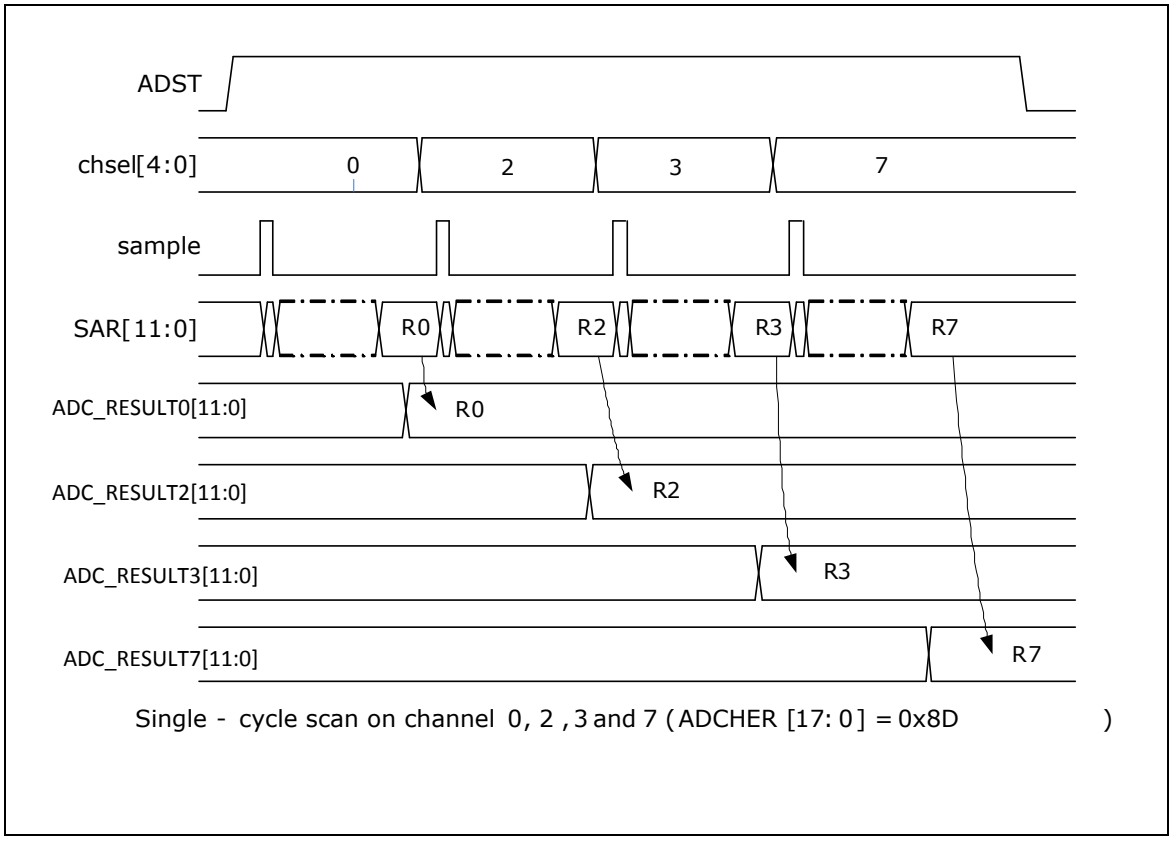



Figure 5-9 ADC Single-cycle Scan on Enabled Channels Timing Diagram

5.5.4.3 Continuous Scan Mode

In continuous scan mode, A/D conversion is performed sequentially on the specified channels that enabled by CHEN bits in ADCHER register (maximum 8 external channels and one internal channel for ADC and internal DAC0, DAC1). The operations are as follows.

1. When the ADST bit in ADCR is set to 1 by software or external trigger input or by timer event selected by TMSEL[1:0] in ADCR register, A/D conversion starts on the channel with the lowest number.
2. When A/D conversion for each enabled channel is completed, the result of each enabled channel is stored in the A/D data register corresponding to each enabled channel.

3. Once when all of the enabled channel sequentially completes A/D converting, the ADF bit (ADSR[0]) will be set to 1. If the ADIE bit is set to 1 at this time, an ADINT interrupt is set after A/D conversion ends.
4. Following the step 3, conversion of the first enabled channel starts again.
5. Steps 2 to 4 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops and the A/D converter enters the idle state.

An example timing diagram for continuous scan on enabled channels (0, 2, 3 and 7) is shown below:

(In this example, channel 0, 2, 3 and 7 are enabled.)

(This example is only appropriate for ADC.)

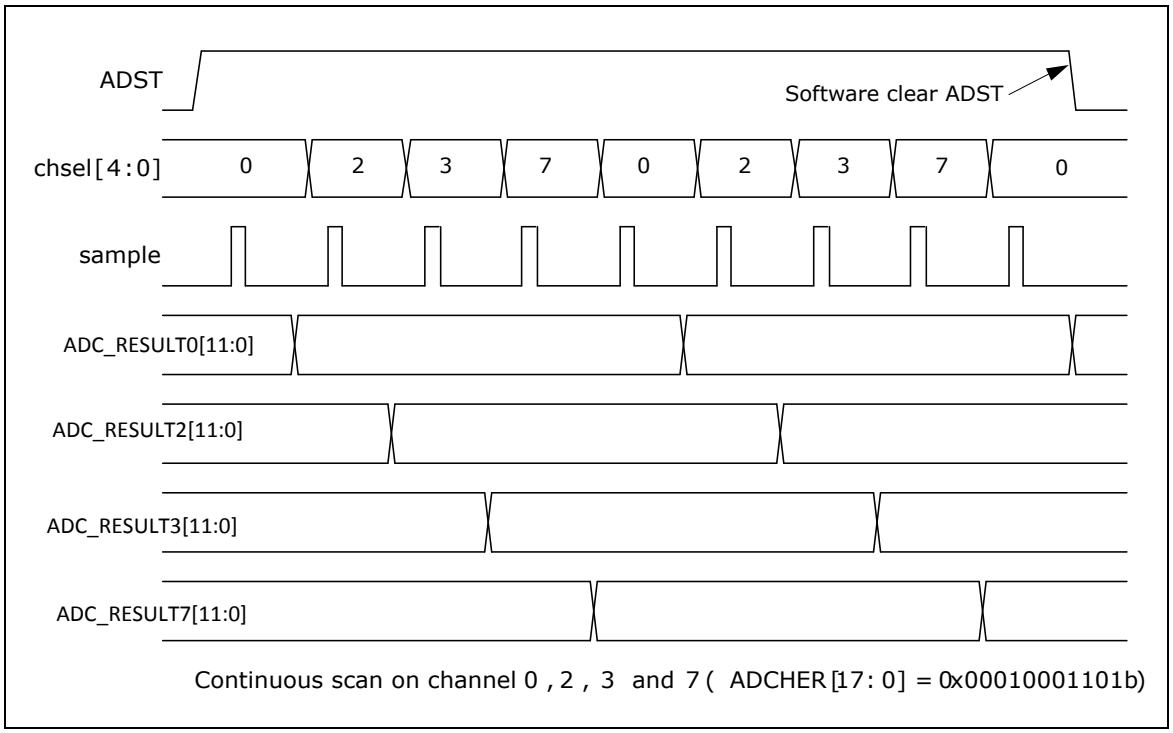



Figure 5-10 ADC Continuous Scan on Enabled Channels Timing Diagram

5.5.4.4 ADC Started by External Rriggering

A/D conversion can be triggered by external pin request. When the ADCR.TRGEN is set to high to enable ADC external trigger function, setting the TRGS[1:0] bits to 00b is to select external trigger input from the STADC pin. Software can set TRGCOND[1:0] to select trigger condition is falling/rising edge or low/high level. An 8-bit sampling counter is used to deglitch. If level trigger condition is selected, the STADC pin must be kept at defined state at least 8 PCLKs. The ADST bit will be set to 1 at the 9th PCLK and start to conversion. In level trigger mode conversion is continuous as long as the external trigger input is in asserted state if external trigger input is pull at low (or high state). It is stopped only when external condition trigger condition disappears. If edge trigger condition is selected, the high and low state must be kept at least 4 PLCKs. When a trigger signal with pulse width smaller than the specified width (4 PCLKs), conversion is not triggered.

5.5.4.5 Conversion Result Monitor by Compare Mode

The ADC controller provides two sets of compare register ADCMPR0 and ADCMPR1 to monitor at most two specified channel conversion results from A/D conversion module, refer to Figure 5-11. Software can select which channel to be monitored by set CMPCH(ADCMPRx[5:]0]) and

CMPCOND bit is used to check conversion result is either less than or greater than (equal to) the specified value in CMPD[11:0]. When the conversion of the channel specified by CMPCH is completed, the comparing action will be triggered one time automatically. When the compare result meets the setting, compare match counter will increase by 1, Once the counter value reaches the setting of (CMPMATCNT+1),CMPF bit will be set to 1, If CMPIE bit is set, an ADINT interrupt request is generated. Software can use this function to monitor whether an external analog input voltage traverse the specified threshold in scan mode without imposing a load on software. Detailed logics diagram is shown below.

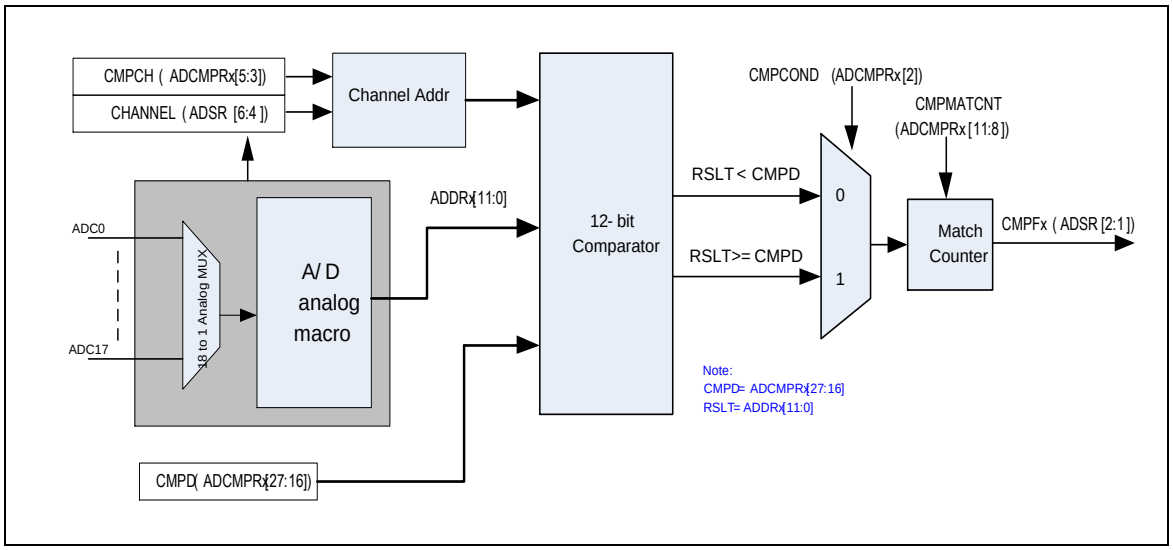



Figure 5-11 ADC Conversion Result Monitor Logic Diagram

5.5.4.6 Interrupt Sources

The A/D converter generates a conversion end flag, ADF in ADSR register at the ending moment of A/D conversion. If ADIE bit in ADCR is set, the conversion end interrupt is asserted via ADINT occurs. If CMPIE bit is enabled and A/D conversion result meets the setting in ADCMPR register, monitor interrupt occurs, and ADINT will be set also. CPU can clear CMPF and ADF to stop interrupt request.

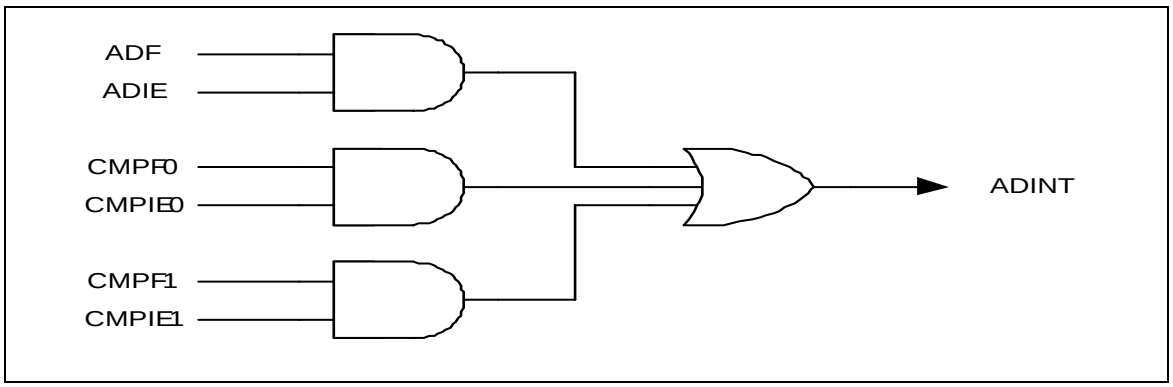



Figure 5-12 ADC Controller Interrupt

5.5.4.7 Peripheral DMA Request

When A/D conversion is finished, the converted result is loaded into AD_RESULTx(x=0~10) register and VALID bit is set to 1. If PTEN bit in ACDR is set, ADC controller will generate PDMA request to ask a data transfer. Having the converted result read by PDMA in response to PDMA

request enables continuous conversion to be achieved without CPU intervention.

#### 5.5.4.8 ADC enabled by timer event

Users can configure ADC to use timer trigger function by programming TMSEL,TMTRGMOD and TMPDMACNT in ADCR register. If AD is power down, timer event can enable ADC. TMSEL in ADCR register selects timer event source.

After ADC wakes up, it starts to transfer and pass the ADC_RESUT to memory by PDMA. User should configure PTEN in ADCR register to enable PDMA transfer and configure ADMD to run ADC in continuous, single or single cycle mode; and configure TMPDMACNT in ADCR register to specify the amount of ADC_RESULT that PDMA will deliver to memory each time the timer event occurs. After PDMA have delivered the amount of ADC_RESULT specified in TMPDMACNT, ADC will go to power down until the next timer event coming.

After the total amount of ADC_RESULT configured in PDMA byte count register have been delivered to memory, ADC will go to power down, this time the ADC will not be waken up by the following timer event any more.

All the configurations should be done before the system entering power down because CPU can't read and write register while the whole system is power down. In single-cycle and single mode, PDMA transfer count should be exact the same with enabled channel count.

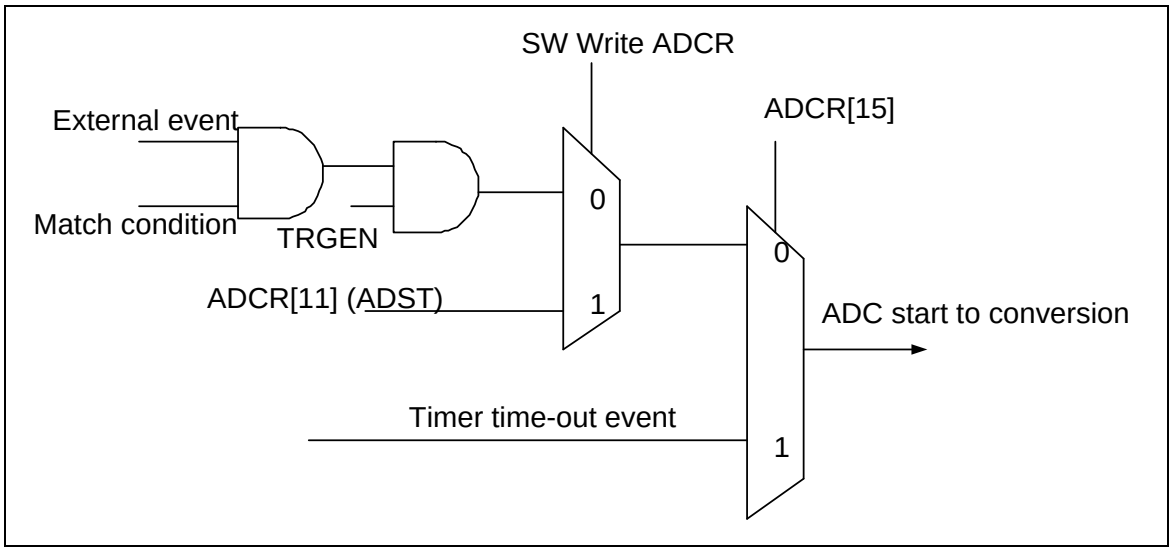



Figure 5-13 ADC Start Conversion Conditions

#### 5.5.4.9 ADC sampling time

The figure below shows the (simplified) equivalent circuit of the S/H (sample and hold) input network, where $C_S$ is the storage capacitor, $R_S$ is the resistance of the sampling switch and $R_I$ is the output impedance of the signal source $V_I$. The Figure 5-16 shows the situation where the conversion cycle j+1 starts immediately after conversion cycle n ends. In this case the duration of the sampling phase is, approximately, 1.5 x ADC_CLK. $C_S$ must be charged in that phase, and it must be ensured that the voltage at its terminals becomes sufficiently near $V_I$. To guarantee this, $R_I$ may not take arbitrarily large values

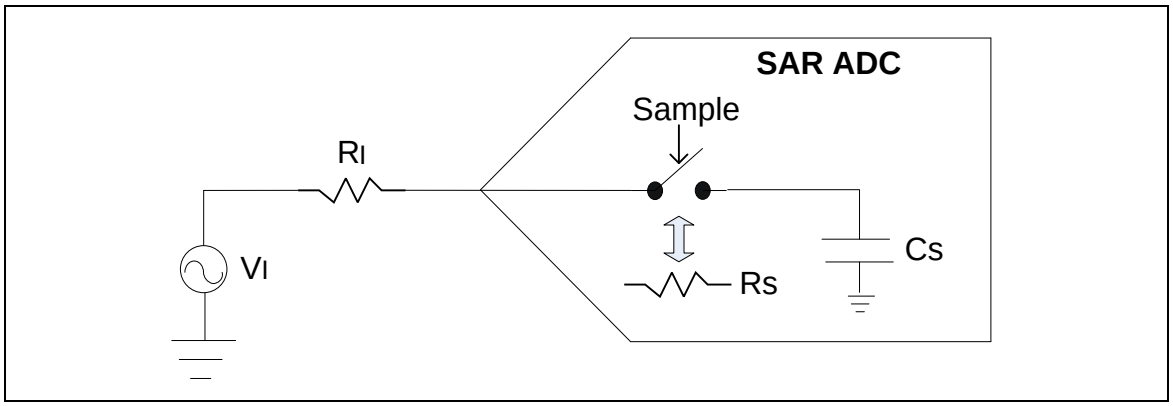



Figure 5-14 Model of the sampling network

Figure 5-15 shows how the sampling time can be increased, to allow the operation with signal sources having a low driving capability: the adc_start signal is delayed during the number of clock cycles necessary to guarantee the accurate input signal sampling. During this period the chsel must remain unchanged. Note that this operation reduces the effective sampling rate.

The ADC has two types of inputs: channel 0~5 are fast inputs and channel 6~17 are slow inputs. All inputs have the same functionality, but during the sampling period, channel 0~5 inputs are faster than channel 6~17

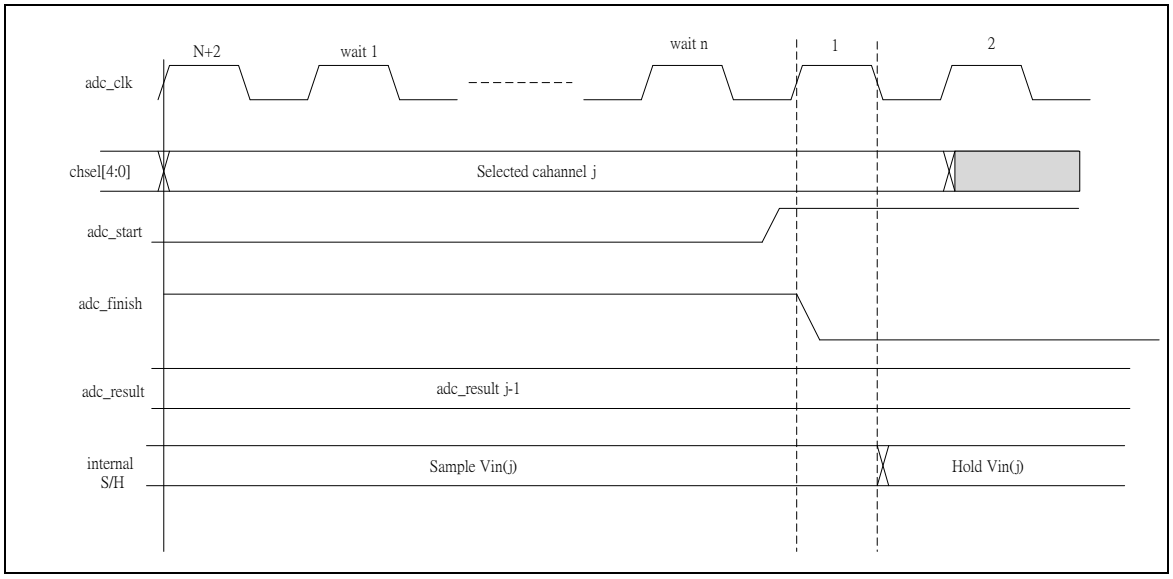



Figure 5-15 Increased Sampling Time Waveform

For both types of inputs, it is possible to achieve the maximum sampling frequency, but under certain conditions (depending either from the resolution mode (*RESSEL*) or from the output impedance of the signal source) the sampling period should be delayed during the necessary clock cycles to guarantee the sampling precision (above figure).

The following graphics indicates the number of additional sample and hold cycles (n), necessary for a wide range of Ri (signal source output resistance) values, for all ADC input channels depending on the resolution mode (*RESSEL*). Use sampling counter registers (ADCCHSAMP0 and ADCCHSAMP1) to add additional sample and hold cycle (n).

Please note that these graphics refer to the **additional** sampling and hold clock cycles (i.e. in the situations where n=0, the sampling period is 1.5 × *ADC_CLK*).

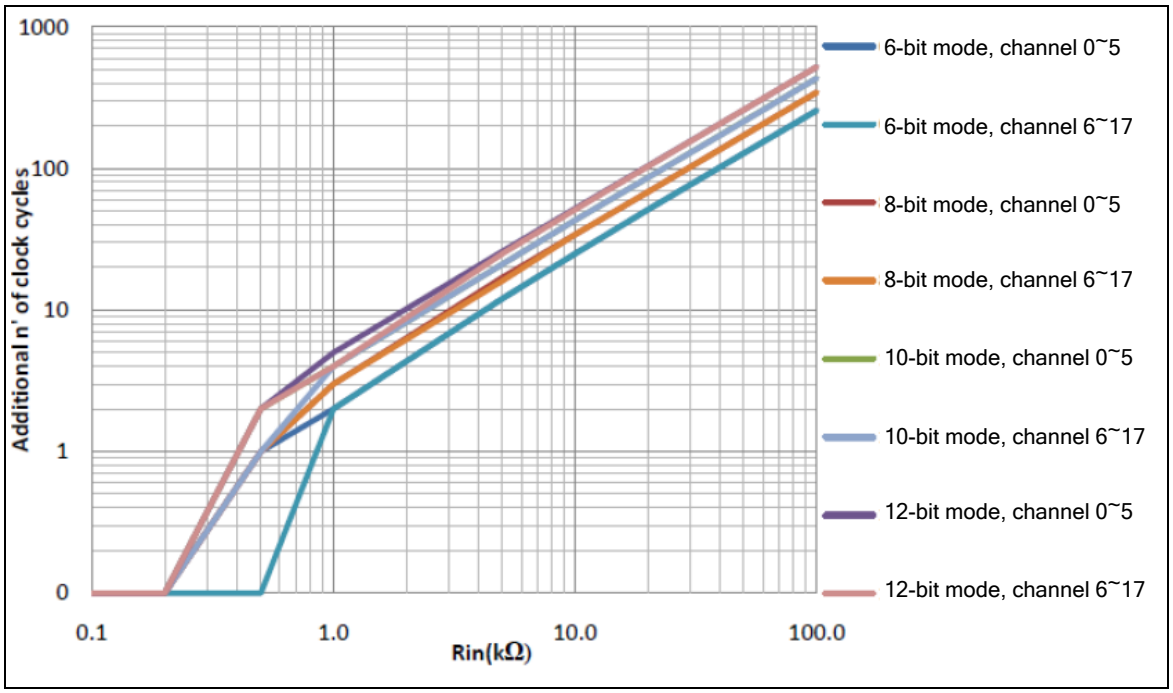



Figure 5-16 Additional Sample and Hold Clock Cycles (n) as a Function of the Signal Source Output Resistance Rin (kΩ)

The results presented in the graphic above were measured under normal conditions (typical process corner, $V_{DD}$ = $AV_{DD}$ = 3.3V, LDO output = 1.8V, Tjunction= 50°C, ADC_CLK=42 MHz, VREF= $AV_{DD}$).

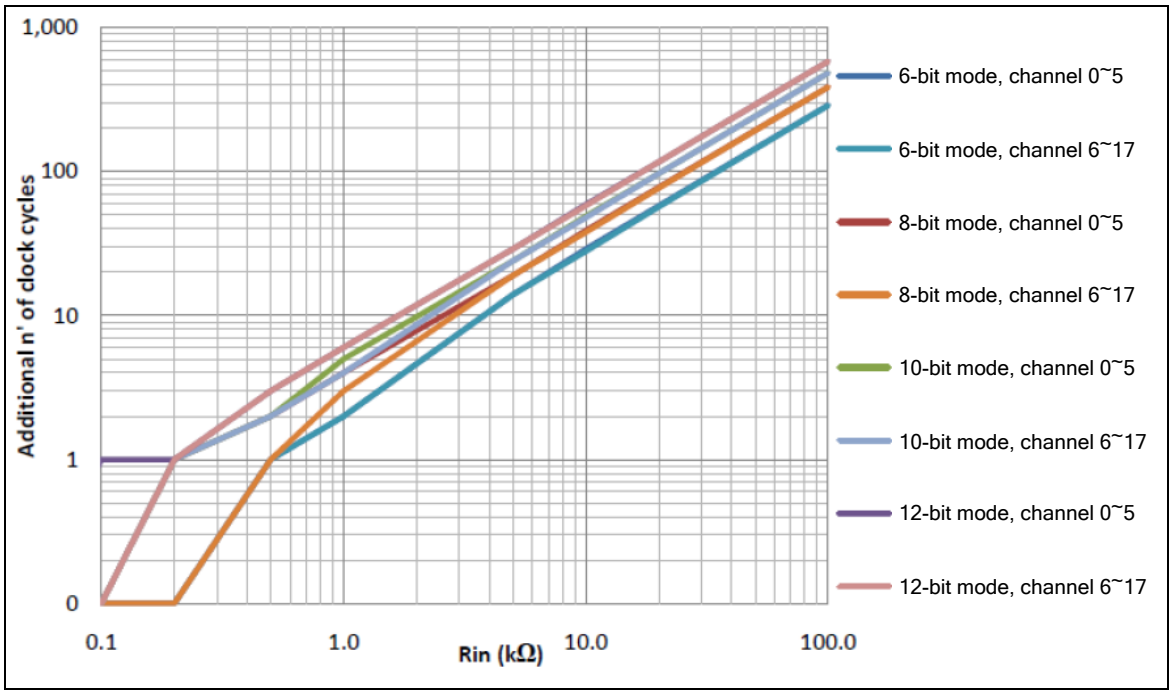



Figure 5-17 Additional Sample and Hold Clock Cycles (n) as a Function of the Signal Source Output Resistance Rin (kΩ)

The results presented in the graphic above were measured under the worst case conditions (slow process corner, $V_{DD}$ = $AV_{DD}$ = 1.8V, LDO output = 1.62V, Tjunction= -40°C, ADC_CLK=42 MHz, $V_{REF}$=

$AV_{DD}$).

#### 5.5.4.10 ADC Power-down mode

There are two Power-down modes user can select, including Power-down mode, and Standby mode. User can configure PWDMOD in ADCPWD register to determine what Power-down mode that user want to be before disabling ADEN in ADCR register.

In different Power-down mode (power down,standby ), the power up sequence are quite different, user should know currently Power-down mode and configure PWDMOD in ADCPWD register to determine what power up sequence that user want to be before enabling ADCEN in ADCR register, if the sequence was wrong, ADC would be mal-function .

The difference between those Power-down modes are power consumption and the stable time after resuming from each Power-down mode, the least power consumption is Power-down mode and then p standby mode,and stable time are in reversed order Before ADC entering power down, make sure that ADC is stop (by disabling ADST) and all conversion are completed (by polling ADF)

#### 5.5.4.11 ADC Offset calibration

To decrease the effect of electrical random noise, the ADC performs calibration to get average offset measurement. Afterwards, in normal operation, the digital block applies the calibrated word to the internal ADC capacitor array, so that the offset voltage is removed.

User can set CALEN to high and select CALSEL to 1(to do calibration) and write 1 to CALSTART, then waiting for CALDONE bit to high; when CALDONE is high, the calibration is complete and the calibration word is in ADC_CALWORD register.

User can also load the specified calibrated word to ADC_CALWORD register to save time to complete the calibration method. The configuration are the same except setting CALSEL to 0, and waiting for CALDONE bit to high; when CALDONE is high, the load calibration word is complete and the loaded calibration word is now applied to ADC.

#### 5.5.4.12 Slectable resolution

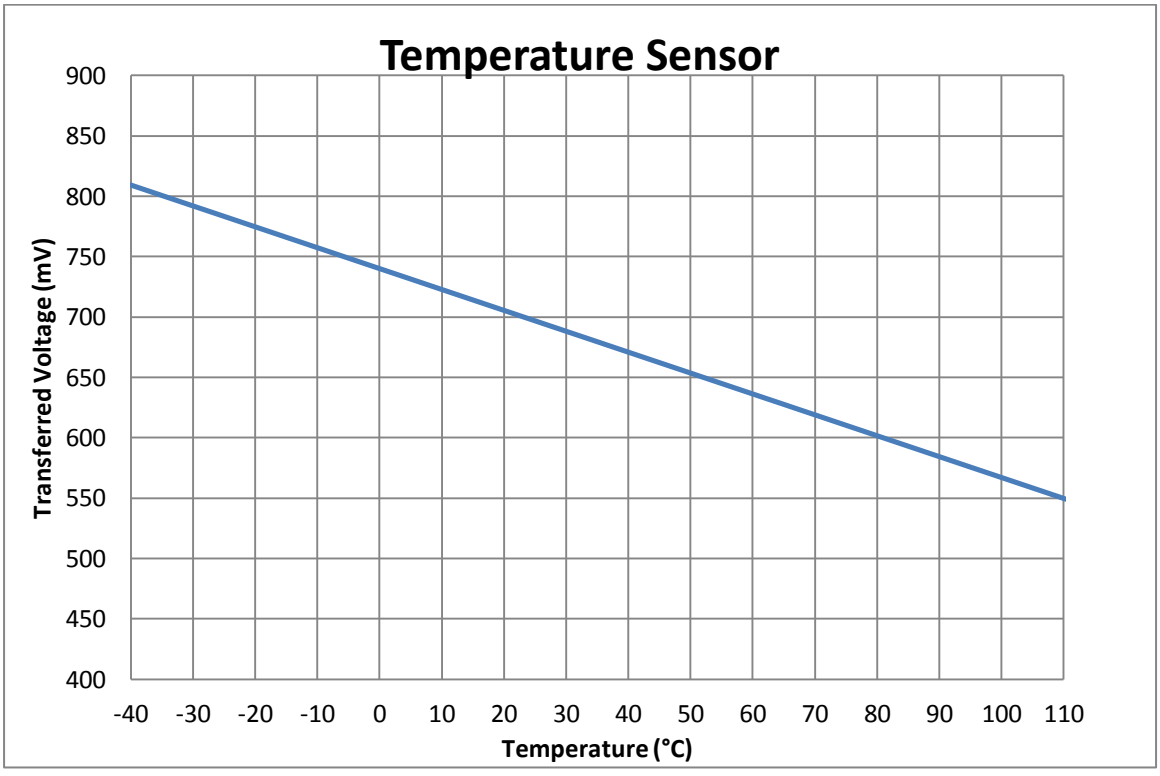
The ADC has the selectable resolution between 12, 10, 8 and 6 bit. User can choose the resolution by setting RESSEL in ADCR register.

The resolution selection can only be updated after the end of the conversion (ADF becomes high), different resolutions will result in the different conversion cycle. Take 12 bit resolution for example, it will take 19 ADC clock cycle to complete one channel conversion; and 17 ADC clock cycle,15 ADC clock cycle, 14 ADC clock cycle for the resolution 10 bits, 8 bits, and 6 bits.

#### 5.5.4.13 Temperature Sensor

The figure below shows the typical temperature sensor transfer function. The formula for the output voltage (Vtemp) is as below equation.

$V_{temp}$ (mV) = -1.73 (mV/℃) x Temperature (℃) + 740 (mV).


Temperature Sensor
Transferred Voltage (mV)
900
850
800
750
700
650
600
550
500
450
400
-40 -30 -20 -10 0 10 20 30 40 50 60 70 80 90 100 110
Temperature (°C)

## 5.6 Digital to Analog Converter (DAC)

### 5.6.1 Overview

DAC is a 12-bit voltage-output digital-to-analog converter. Two DACs are implemented in this chip.

### 5.6.2 Features

DAC is a 12-bit voltage-output DAC. DAC can use in conjunction with the PDMA controller. When two DACs are present, they may be grouped together for synchronous update operation.

Features:

- Int_VREF or VREF or AVDD reference voltage selection
- Synchronized update capability for two DACs
- DAC maximum conversion rate is 500 KSPS

### 5.6.3 Block Diagram

The block diagram of the two DACs is shown in the following figure.

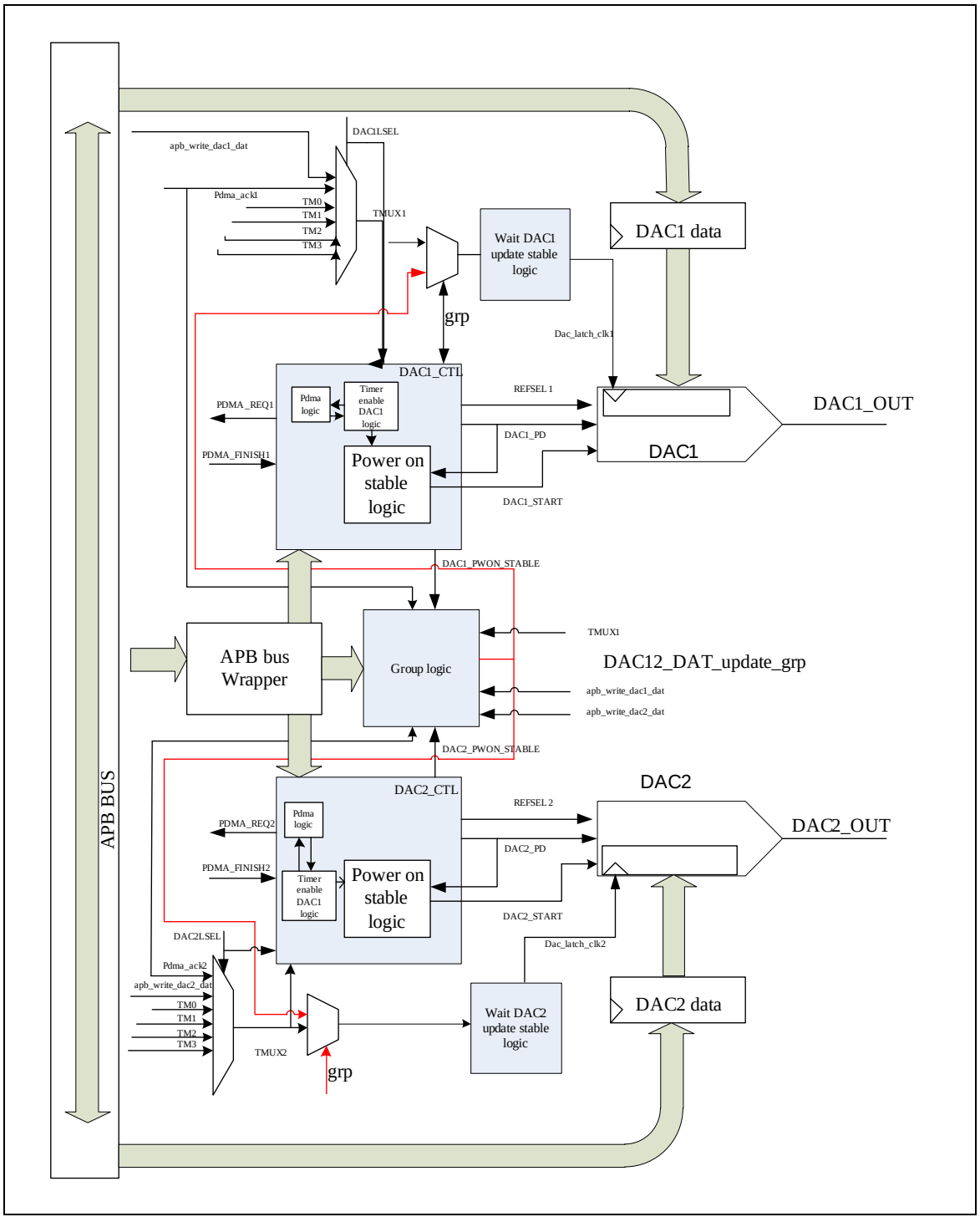



Figure 5-18 DAC0 and DAC1 Block Diagram

### 5.6.4 Functional Description

The DAC is configured by user software. The setup and operation of the DAC is discussed in the following sections.

#### 5.6.4.1 DAC Core

The conversion time between each DACx_DAT is 2 us and the power on stable time is 6 us, both are at the worse case.The maximum DAC0 and DAC1 output voltage (full scale) is limited to the selected reference voltage.

#### 5.6.4.2 DAC Reference voltage

The reference for the DAC is configured to use either an external reference voltage or the internal voltage reference generator or AVDD with three selectable voltage levels (Int_VREF \ VREF \ AVDD).

#### 5.6.4.3 DAC operation

DAC will update the output voltage level whenever user updates the DACx_DAT. There are several ways that user can update the DACx_DAT register by configuring the DACLSEL in DACx_CTL register.

When DACLSEL = 000, DAC will update the output voltage level when user write data to DACx_DAT register; When DACLSEL = 001, DAC will update the output voltage level when PDMA controller send acknowledge to DAC controller. Obviously, this configuration is for DAC's PDMA usage, user should also configure other PDMA setting together with this function; When DACLSEL = 010, 011, 100 or 101, DAC will update the output voltage level every time when the corresponding timer event coming and user should write DACx_DAT before each timer event coming. Owing to the limit of DAC's conversion time, the highest conversion rate is 500 kHz (1/2us). When user updates DACx_DAT register faster than 500 kHz, some data will lose and DAC will output the latest data in DACx_DAT. Because the operating frequency (PCLK) of DAC controller will be different case by case, user can adjust the value in WAITDACCONV of DAC01_COMCTL register to meet the limit of DAC conversion rate. Power on stable time is also adjustable by setting DACx_CTL[DACPWONSTBCNT] to meet the DAC 6us stable time.

**Note:** DAC has two timing requirements:

- Stable time: The time DAC ready to conversion after DAC power on from DAC power down state, user should wait 6 us to meet stable time requirement by setting DACx_CTL[DACPWONSTBCNT]
- Settle time: The time DAC converts digital data to analog data. User can set DAC01_COMCTL[WAITDACCONV] to meet 2us settle time requirement

#### 5.6.4.4 Grouping DAC0 and DAC1

Two DACx can be grouped together with the GRP bit to synchronize the update of each DAC output. Hardware ensures that those two DACs in a group update simultaneously independent of any interrupt.

The DAC0 and DAC1 are grouped by setting the GRP bit of DAC01_COMCTL.When DAC0 and DAC1 are grouped:

The DAC0's DACLSEL bits select the update trigger for both DACs

When DAC0's DACLSEL = 000, DACs will output each updated DAC_DAT simultaneously after user writing to DAC0_DAT and DAC1_DAT register, the order of writing both registers are not matter.

When DAC0's DACLSEL = 001, DACs will output each updated DAC_DAT simultaneously after PDMA sending both acknowledge to DAC controller

When DAC0's DACLSEL = 010, DACs will output each updated DAC_DAT simultaneously after Timer0's timer event coming

When DAC0's DACLSEL = 011, DACs will output each updated DAC_DAT simultaneously after

Timer1's timer event coming

When DAC0's DACLSEL = 100, DACs will output each updated DAC_DAT simultaneously after Timer2's timer event coming

When DAC0's DACLSEL = 101, DACs will output each updated DAC_DAT simultaneously after Timer3's timer event coming

**Note:**

When DAC0 and DAC1 are grouped and without PDMA operation ( DACLSEL≠001b ), both DACx_DAT registers must be written to before the output– update – even if data for one or both of the DACs is not changed (PDMA operation will prepare DAC_DAT automatically for both DAC)

Figures below show a latch-update timing example for grouped and ungrouped DAC0 and DAC1.

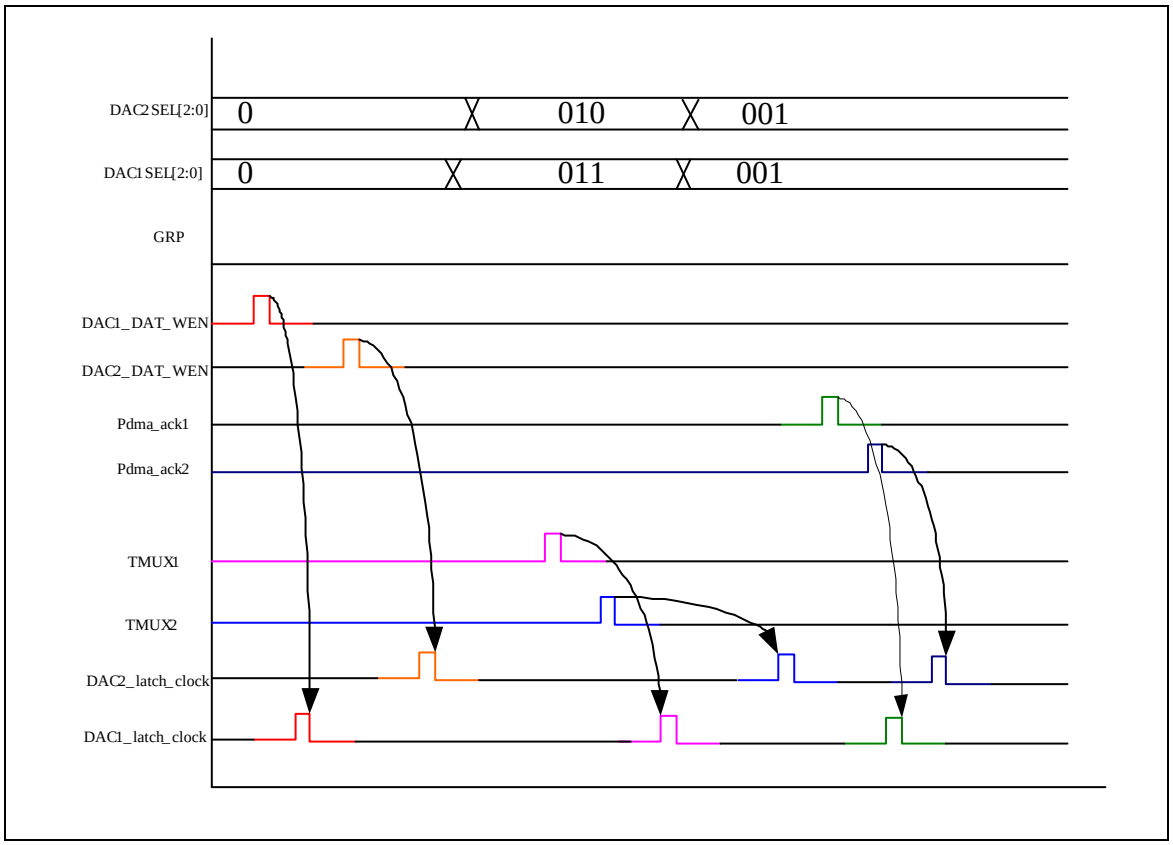



Figure 5-19 DAC0 and DAC1 Ungroup Update Example

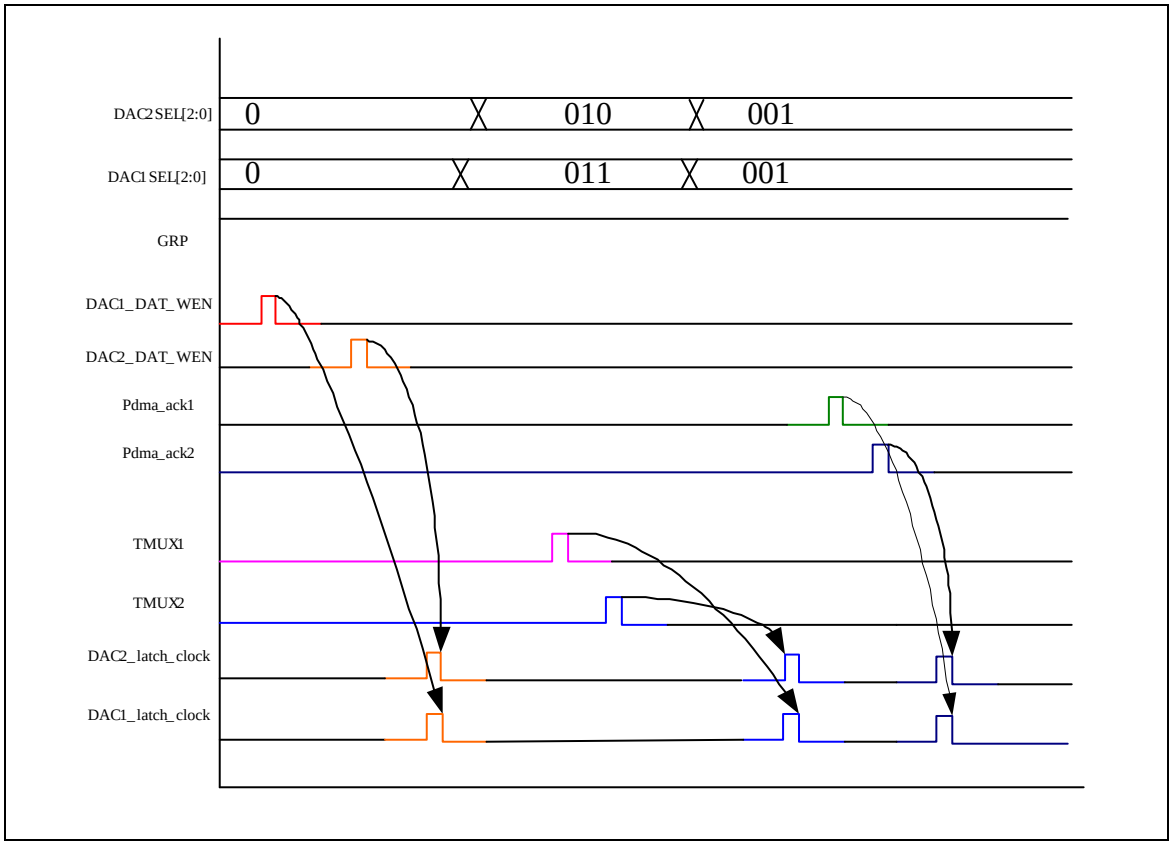



Figure 5-20 DAC0 and DAC1 Group Update Example

**Note:** DAC0,1 Settling Time:

The DMA controller is capable of transferring data to the DAC0,1 faster than the DAC0,1 output can settle. The user must assure the settling time is not violated when using the DMA controller. See the device-specific data sheet for parameters.

DAC0,1 Interrupts:

DACIFG bit is set when DACx_DATA is latched internally. DACIFG bit indicates that the DACx is ready for new data. If DACIE bit is set, the DACIFG generates an interrupt request. Users must write 1 to clear DACIFG

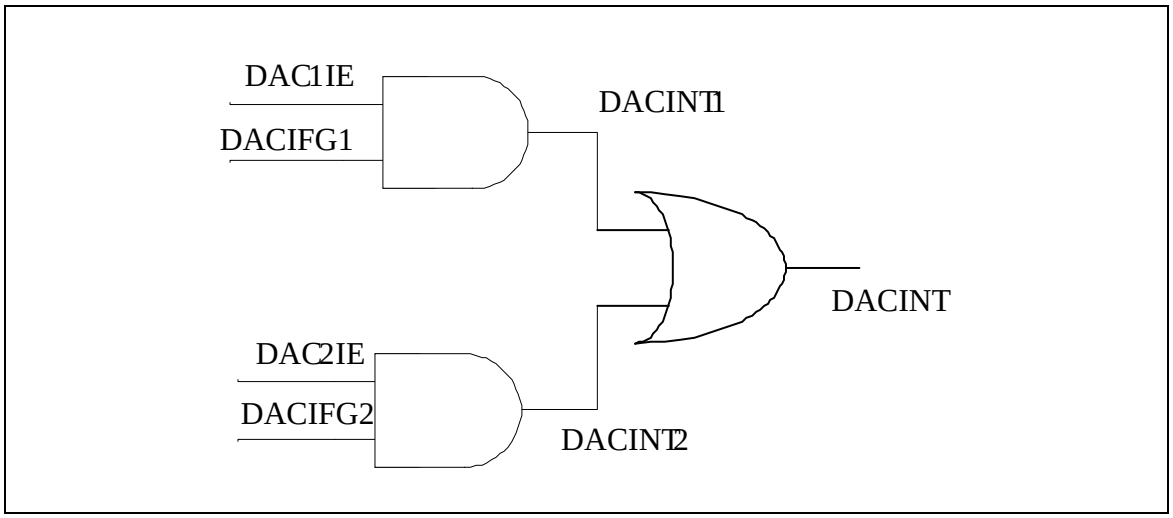



Figure 5-21 DAC Interrupt

## 5.7 DMA Controller

### 5.7.1 Overview

The DMA controller contains six channel peripheral direct memory access (PDMA) controllers, a video direct memory access (VDMA) controller and a cyclic redundancy check (CRC) generator. The PDMA controller can transfer data to and from memory or transfer data to and from APB devices. The DMA has eight channels of DMA including one channel VDMA (Memory-to-Memory) and six channels PDMA (Peripheral-to-Memory or Memory-to-Peripheral or Memory-to-Memory) and a CRC controller. For channel0 VDMA, it supports block transfer from memory to memory. For PDMA channel (DMA CH1~CH6), there is one-word buffer as transfer buffer between the Peripherals APB devices and Memory. And for channel 0 VDMA, there is a two-word buffer.

Software can stop the DMA operation by disable PDMA [PDMACEN]/VDMA [VDMACEN]. Software can recognize the completion of a DMA operation by software polling or when it receives an internal DMA interrupt. The DMA controller can increase source or destination address, fixed or wrap around them as well.

The DMA controller also contains a cyclic redundancy check (CRC) generator that can perform CRC calculation with programmable polynomial settings. The CRC engine support CPU PIO mode and DMA transfer mode.

### 5.7.2 Features

Seven DMA channels and a CRC generator: 1 VDMA channel and 6 PDMA channels. Each channel can support a unidirectional transfer.

AMBA AHB master/slave interface compatible, for data transfer and register read/write.

Hardware round robin priority scheme.

- VDMA
  - Memory-to-memory transfer
  - Supports block transfer with stride
  - Supports word/half-word/byte boundary address
  - Supports address direction: increment and decrement
- PDMA
  - Peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfer
  - Supports word boundary address
  - Supports word alignment transfer length in memory-to-memory mode
  - Supports word/half-word/byte alignment transfer length in peripheral-to-memory and memory-to-peripheral mode
  - Supports word/half-word/byte transfer data width from/to peripheral
  - Supports address direction: increment, fixed, and wrap around
- Cyclic Redundancy Check (CRC)
  - Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32
    - CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
    - CRC-8: $X^8 + X^2 + X + 1$
    - CRC-16: $X^{16} + X^{15} + X^2 + 1$
    - CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 +$

$X^2 + X + 1$

- Programmable seed value
- Supports programmable order reverse setting for input data and CRC checksum
- Supports programmable 1's complement setting for input data and CRC checksum
- Supports CPU PIO mode or DMA transfer mode
- Supports 8/16/32-bit of data width in CPU PIO mode
  - 8-bit write mode: 1-AHB clock cycle operation
  - 16-bit write mode: 2-AHB clock cycle operation
  - 32-bit write mode: 4-AHB clock cycle operation
- Supports byte alignment transfer length in CRC DMA mode

### 5.7.3 Block Diagram

The DMA clock control and block diagram are shown as follows.

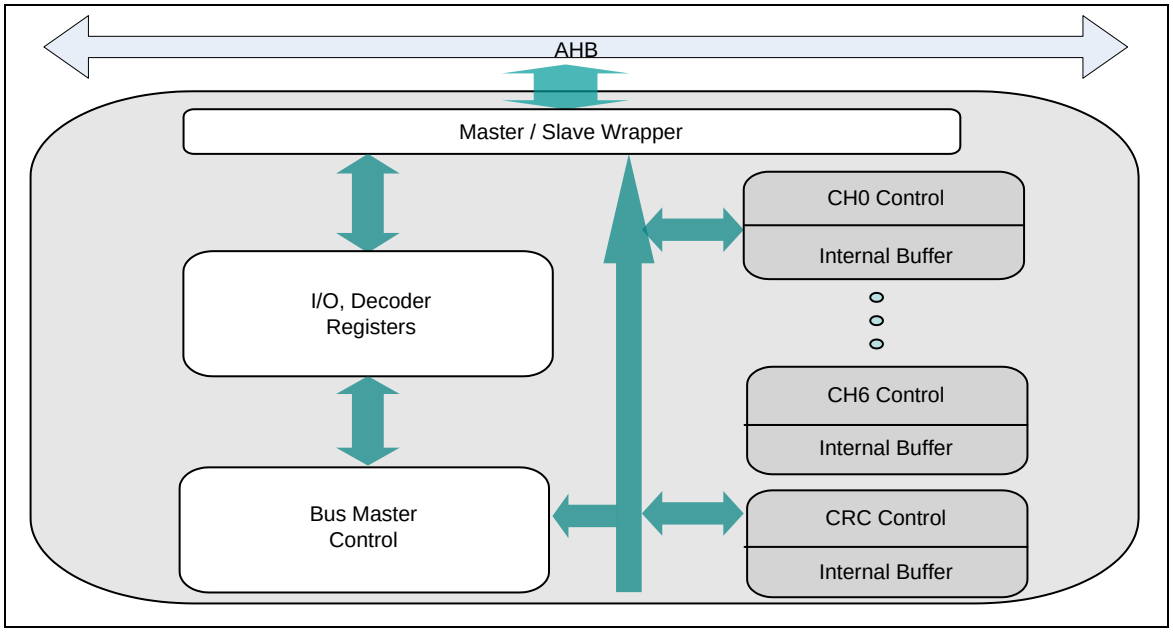



Figure 5-22 DMA Controller Block Diagram

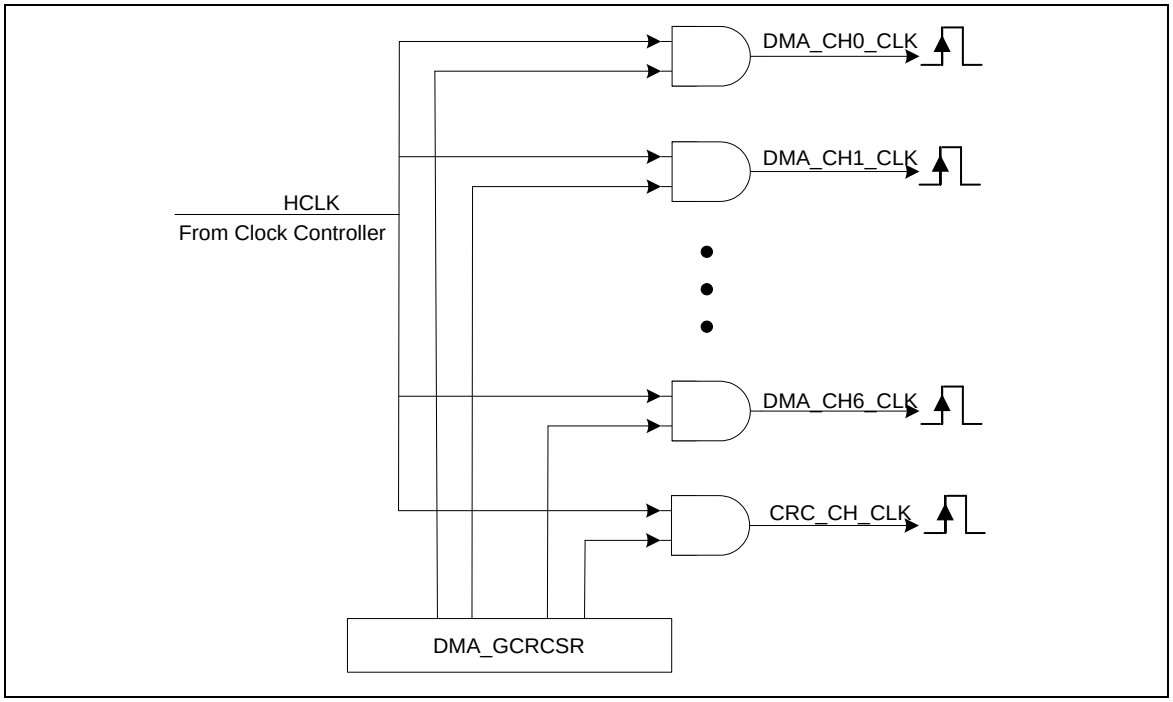



Figure 5-23 DMA Clock Control Diagram

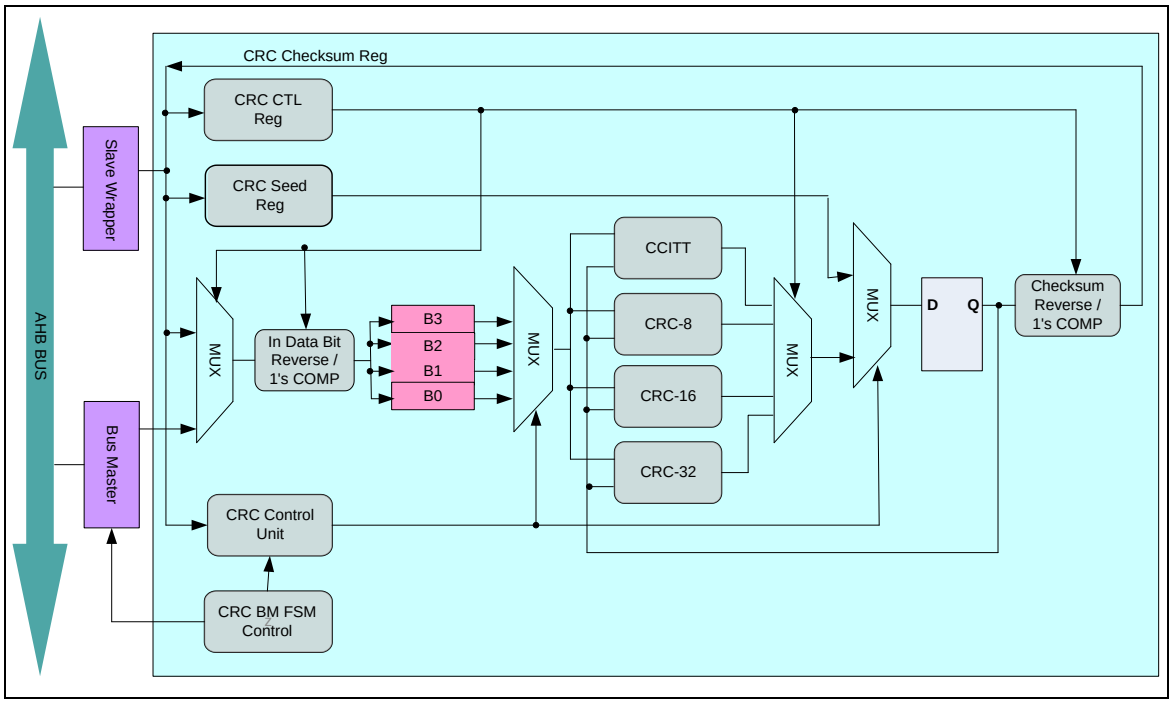



Figure 5-24 CRC Generator Block Diagram

### 5.7.4 Functional Description

The direct memory access (DMA) controller module transfers data from one address to another address, without CPU intervention. The DMA controller contains eight channels that including one channel VDMA (Memory-to-Memory) and six channels PDMA (Peripheral-to-Memory or Memory-to-Peripheral or Memory-to-Memory) and one CRC channel.

The CPU can recognize the completion of a DMA operation by software polling or when it receives an internal DMA interrupt. As to the source and destination address, the DMA controller has three different modes: increased, fixed and wrap around operation mode.

#### 5.7.4.1 VDMA

The DMA controller contains one channel VDMA (Memory-to-Memory). The VDMA module can transfers data from one address to another address and can support block transfer with stride. When operating in VDMA mode, the transfer address can incremented successively or decremented successively.

In Without Block Transfer mode (VDMA_CSR [STRIDE_EN] Disabled), software must enable DMA channel VDMA [VDMACEN] and then write a valid source address to the VDMA_SAR register, a destination address to the VDMA_DAR register, and a transfer count to the VDMA_BCR register. Next, trigger the VDMA_CSR [TRIG_EN]. The DMA will continue the transfer until VDMA_CBCR comes down to zero

In Block Transfer mode (VDMA_CSR [STRIDE_EN] Enabled), software must enable DMA channel VDMA [VDMACEN] and then write a valid source address to the VDMA_SAR register and a source address offset count to VDMA_SASOCR [SASTOBL] register, a destination address to the VDMA_DAR register and a destination address offset count to VDMA_DASOCR [DASTOBL], and a transfer count to the VDMA_BCR register and a block byte count to VDMA_SASOCR [STBC]. Next, trigger the VDMA_CSR [TRIG_EN]. The DMA will continue the transfer until VDMA_CBCR comes down to zero. The following figure shows the block transfer relationship between source memory and destination memory.

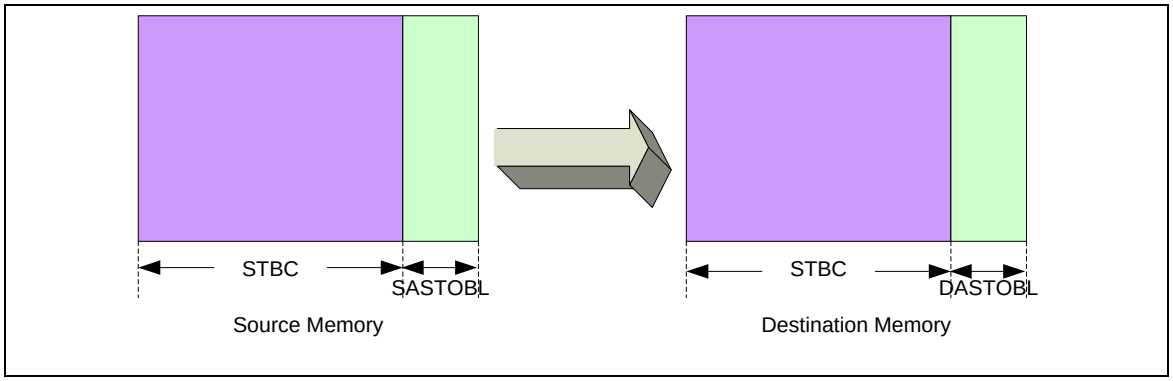



Figure 5-25 VDMA Block Transfer

If an error occurs during the VDMA operation, the channel stops unless software clears the error condition and sets the VDMA_CSR [SW_RST] to reset the VDMA channel and set VDMA_CSR [VDMACEN] and [TRIG_EN] bits field to start again.

#### 5.7.4.2 PDMA

The DMA controller contains six channels PDMA (Peripheral-to-Memory or Memory-to-Peripheral or Memory-to-Memory). As to the source and destination address, the DMA controller has three different modes: increased, fixed and wrap around operation mode.

Every PDMA channel behavior is not pre-defined, so users must configure the channel service settings of DMA_DSSR0 and DMA_DSSR1 before start the related PDMA channel.

Software must enable DMA channel PDMA [PDMACEN] and then write a valid source address to the PDMA_SARx register, a destination address to the PDMA_DARx register, and a transfer count to the PDMA_BCRx register. Next, trigger the DMA_CSRx [TRIG_EN]. If the source address and destination is not in wrap around mode, the PDMA will continue the transfer until PDMA_CBCRx comes down to zero (in wrap around mode, when PDMA_CBCRx is equal to zero, the PDMA will reload PDMA_CBCRx and work around until software disables PDMA_CSRx[DMACEN]), If an error occurs during the PDMA operation, the channel stops unless software clears the error condition and sets the PDMA_CSRx [SW_RST] to reset the PDMA channel and set PDMA_CSRx [PDMACEN] and [TRIG_EN] bits field to start again.

In PDMA (Peripheral-to-Memory or Memory-to-Peripheral) mode, DMA can transfer data between the Peripherals APB IP (ex: UART, SPI, ADC….) and Memory.

#### 5.7.4.3 CRC

The DMA controller contains a cyclic redundancy check (CRC) generator that can perform CRC calculation with programmable polynomial settings. The operation polynomial includes CRC-CCITT, CRC-8, CRC-16 and CRC-32; Software can choose the operation polynomial mode by setting CRC_MODE fields in CRC_CTL register.

The CRC engine support CPU PIO mode (CRC_CTL [CRCCEN] = 1, CRC_CTL [TRIG_EN] = 0) and DMA transfer mode (CRC_CTL [CRCCEN] = 1, CRC_CTL [TRIG_EN] = 1). The following sequence is a program sequence example.

**Procedure When operating in CPU PIO mode:**

Enable CRC engine by setting CRCCEN bit in CRC_CTL register.

Initial Setting. Setting the data format (WDATA_RVS, CHECKSUM_RVS, WDATA_COM and CHECKSUM_COM by setting CRC_CTL register), initial seed value (CRC_SEED) and select the data length by setting CRC_CTL [CPU_WDLEN] register.

Setting CRC reset to load the initial seed value to CRC circuit by setting CRC_RST bit in CRC_CTL register.

Write data to CRC_WDATA to perform CRC calculation.

Get the CRC checksum result by reading CRC_CHECKSUM register.

**Procedure When operating in CRC DMA mode:**

Enable CRC engine by setting CRCCEN bit in CRC_CTL register.

Initial Setting. Setting the data format (WDATA_RVS, CHECKSUM_RVS, WDATA_COM and CHECKSUM_COM by setting CRC_CTL register), initial seed value (CRC_SEED).

Give a valid source address and transfer count by setting CRC_DMASAR and CRC_DMABCR.

Enable CRC_CTL [TRIG_EN] and then hardware will reset the seed value and then read memory data to perform CRC calculation.

Wait CRC DMA transfer and CRC calculation done and then get the CRC checksum result by reading CRC_CHECKSUM register.

## 5.8 External Bus Interface

### 5.8.1 Overview

This chip is equipped with an external bus interface (EBI) to access external device. To save the connections between external device and this chip, EBI support address bus and data bus multiplex mode. Also, address latch enable (ALE) signal is used to differentiate the address and data cycle.

### 5.8.2 Features

- External devices with max. 64 Kbytes size (8-bit data width)/128 Kbytes (16-bit data width) supported
- Supports variable external bus base clock (MCLK)
- Supports 8-bit or 16-bit data width
- Supports variable data access time (tACC), address latch enable time (tALE) and address hold time (tAHD)
- Address bus and data bus multiplex mode supported to save the address pins
- Configurable idle cycle supported for different access condition: Write command finish (W2X), Read-to-Read (R2R), Read-to-Write (R2W)
- Supports PDMA and VDMA transfer

### 5.8.3 Block Diagram

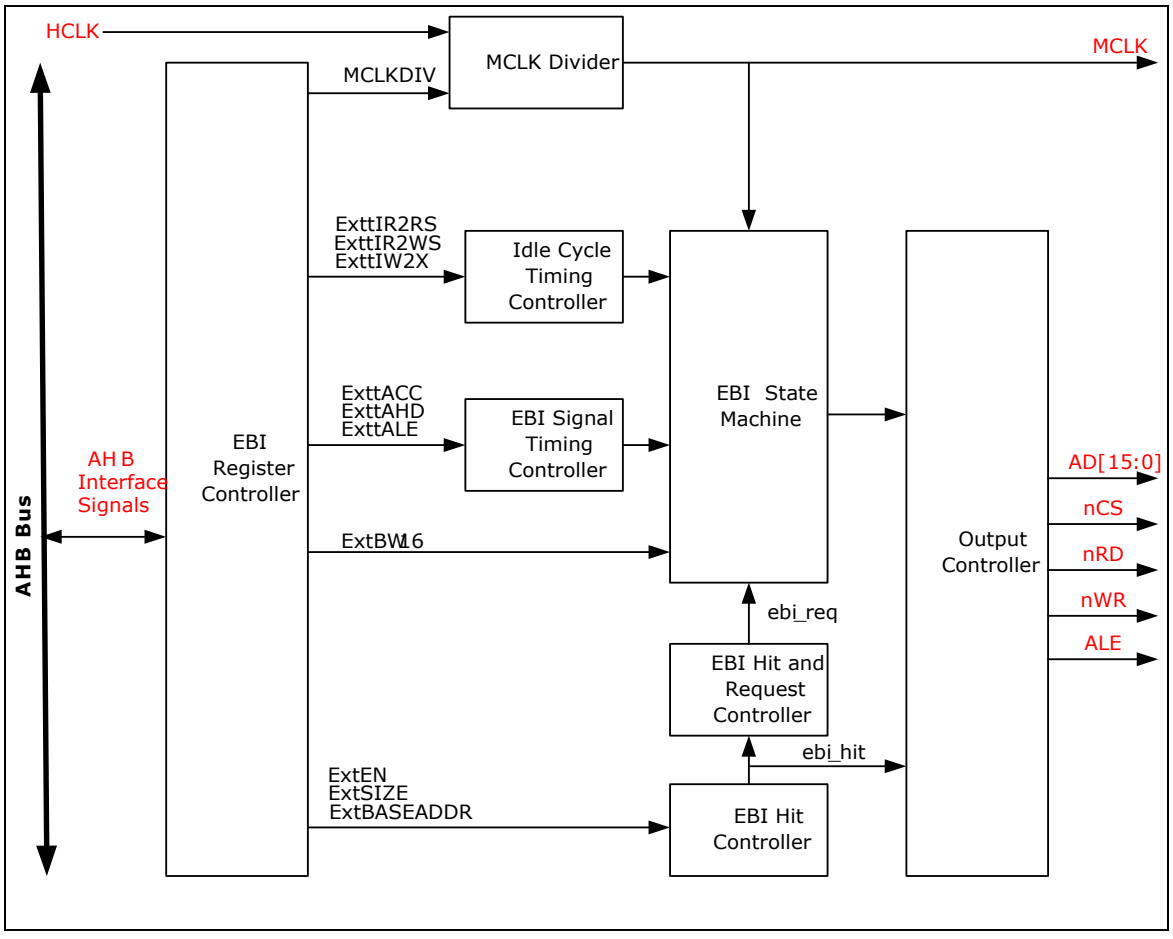


Figure 5-26 EBI Block Diagram

### 5.8.4 Functional Description

5.8.4.1 EBI Area and Address Hit

The EBI mapping address is located at 0x6000_0000 ~ 0x6001_FFFF and the total memory space is 128Kbyte. When system request address hit EBI's memory space, the corresponding EBI chip select signal is asserted and EBI state machine operates.

For an 8-bit device (64Kbyte), EBI mapped this 64Kbyte device to 0x6000_0000 ~ 0x6000_FFFF and 0x6001_0000 ~ 0x6001_FFFF simultaneously.

5.8.4.2 EBI Data Width Connection

The EBI supports devices whose address bus and data bus are multiplexed. For the external device with separated address and data bus, the connection to device needs additional logic to latch the address. In this case, pin ALE is connected to the latch device to latch the address value. Pin AD is the input of the latch device, and the output of the latch device is connected to the address of external device. For 16-bit device, the AD [15:0] shared by address and 16-bit data. For 8-bit device, only AD [7:0] shared by address and 8-bit data, AD [15:8] is dedicated for address and could be connected to 8-bit device directly.

For 8-bit data width, the system address bit [15:0] is used as the device's address [15:0]. For 16-bit data width, system address bit [16:1] is used as the device's address [15:0] and system address bit [0] is useless.

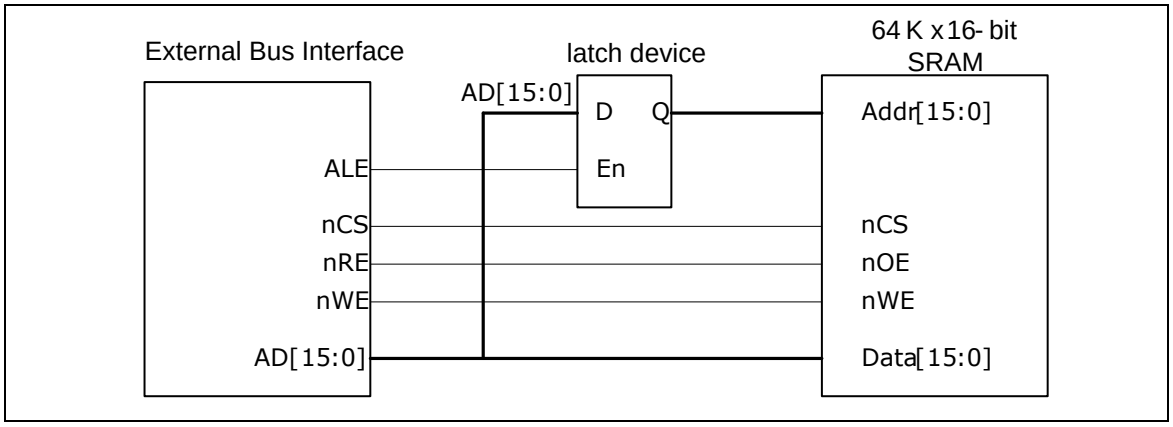



Figure 5-27 Connection of 16-bit EBI Data Width 16-bit Device

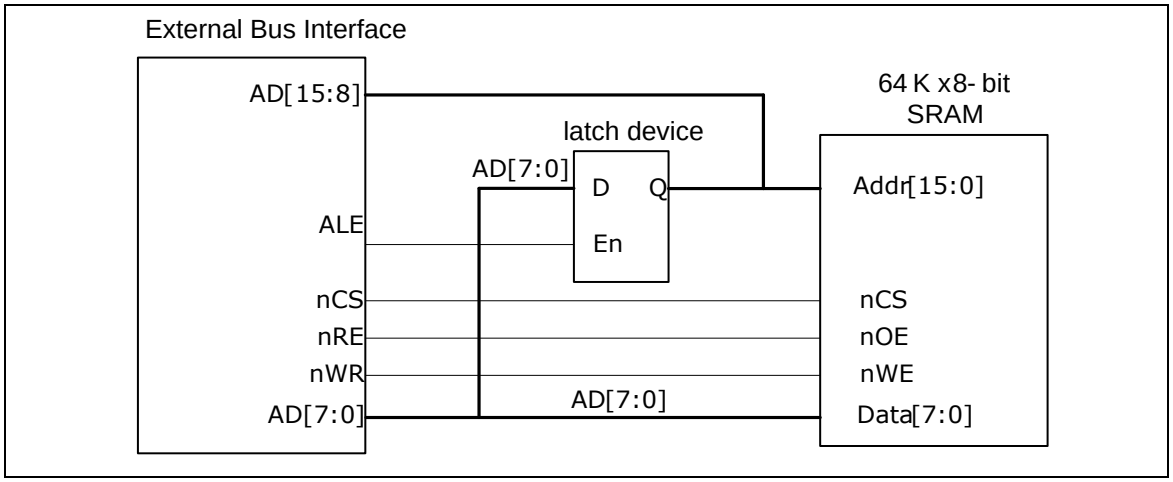


Figure 5-28 Connection of 8-bit EBI Data Width with 8-bit Device

When the system access data width is larger than EBI data width, EBI controller will finish a system access command by operating EBI access more than once. For example, if system requests a 32-bit data through EBI device, the EBI controller will operate accessing four times when setting EBI data width with 8-bit.

#### 5.8.4.3 EBI Operating Control

**MCLK Control**

In this chip, all EBI signals will be synchronized by MCLK when EBI is operating. When this chip connects to the external device with slower operating frequency, the MCLK can divide most to 32 by setting MCLKDIV of register EBICON. Therefore, the EBI is suitable for a wide frequency range of EBI device. If MCLK is set to HCLK/1, EBI signals are synchronized by positive edge of MCLK, else by negative edge of MCLK.

**Operation and Access Timing Control**

In the start of access, chip select (nCS) asserts to low and wait one MCLK for address setup time (tASU) for address stable. Then ALE asserts to high after address is stable and keeps for a period of time (tALE) for address latch. After latch address, ALE asserts to low and wait one MCLK for latch hold time (tLHD) and another one MCLK cycle (tA2D) that is inserted behind address hold time to be the bus turn-around time for address change to data. Then nRD asserts to low when read access or nWR asserts to low when write access. Then nRD or nWR asserts to high after keeps access time (tACC) for reading output stable or writing finish. After that, EBI signals keep for data access hold time (tAHD) and chip select asserts to high, address is released by current access control.

EBI provides a flexible timing control for different external device. In EBI timing control, tASU, tLHD and tA2D are fixed to 1 MCLK cycle, tAHD can modulate to 1~8 MCLK cycles by setting ExttAHD of register EXTIME, and tACC can modulate to 1~32 MCLK cycles by setting ExttACC of register EXTIME, and tALE can modulate to 1~8 MCLK cycles by setting tALE of register EBICON.

| Parameter | Value | Unit | Description |
|---|---|---|---|
| **tASU** | 1 | MCLK | Address Latch Setup Time. |
| **tALE** | 1~8 | MCLK | ALE High Period. Controlled by ExttALE of EBICON. |
| **tLHD** | 1 | MCLK | Address Latch Hold Time. |
| **tA2D** | 1 | MCLK | Address To Data Delay (Bus Turn-Around Time). |
| **tACC** | 1~32 | MCLK | Data Access Time. Controlled by ExttACC of EXTIME. |
| **tAHD** | 1~8 | MCLK | Data Access Hold Time. Controlled by ExttAHB of EXTIME. |
| **IDLE** | 1~16 | MCLK | Idle Cycle. Controlled by ExtIR2R, ExtIR2W and ExtIW2X of EXTIME. |

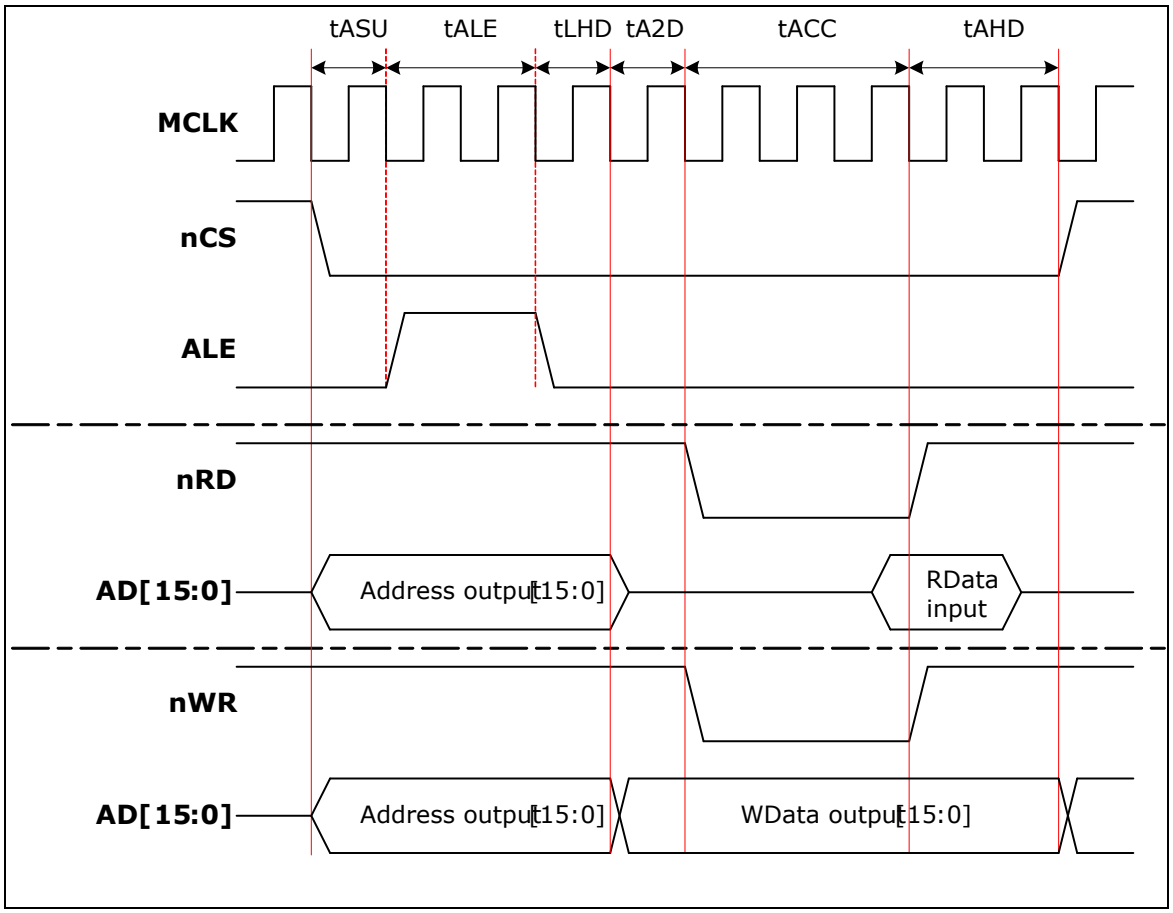



Figure 5-29 Timing Control Waveform for 16-bit Data Width

Above timing waveform is an example of setting 16-bit data width. In this example, AD bus is used for being address [15:0] and data[15:0]. When ALE assert to high, AD is address output. After address is latched, ALE asserts to low and the AD bus change to high impedance to wait device output data in read access operation, or it is used for being write data output.

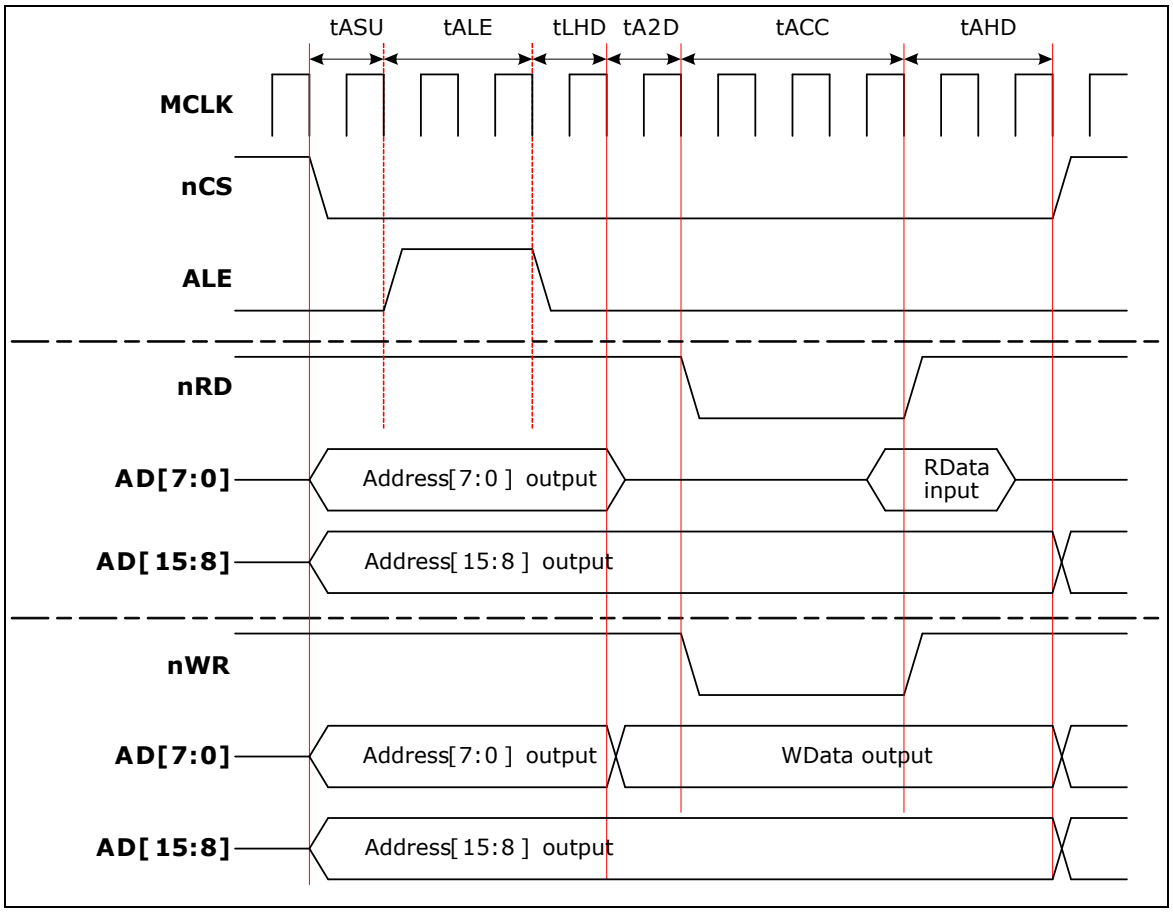



Figure 5-30 Timing Control Waveform for 8-bit Data Width

Above timing waveform is an example of setting 8-bit data width. The difference between 8-bit and 16-bit data width is AD[15:8]. In 8-bit data width setting, AD[15:8] is always Address[15:8] output so that external latch needs only 8-bit width.

**Insert Idle Cycle**

When EBI accessing continuously, there may occur bus conflict if the device access time is much longer compared with system clock frequency. EBI supply additional idle cycle to solve this problem. During idle cycle, all control signals of EBI are inactive. The following figure shows the idle cycle.

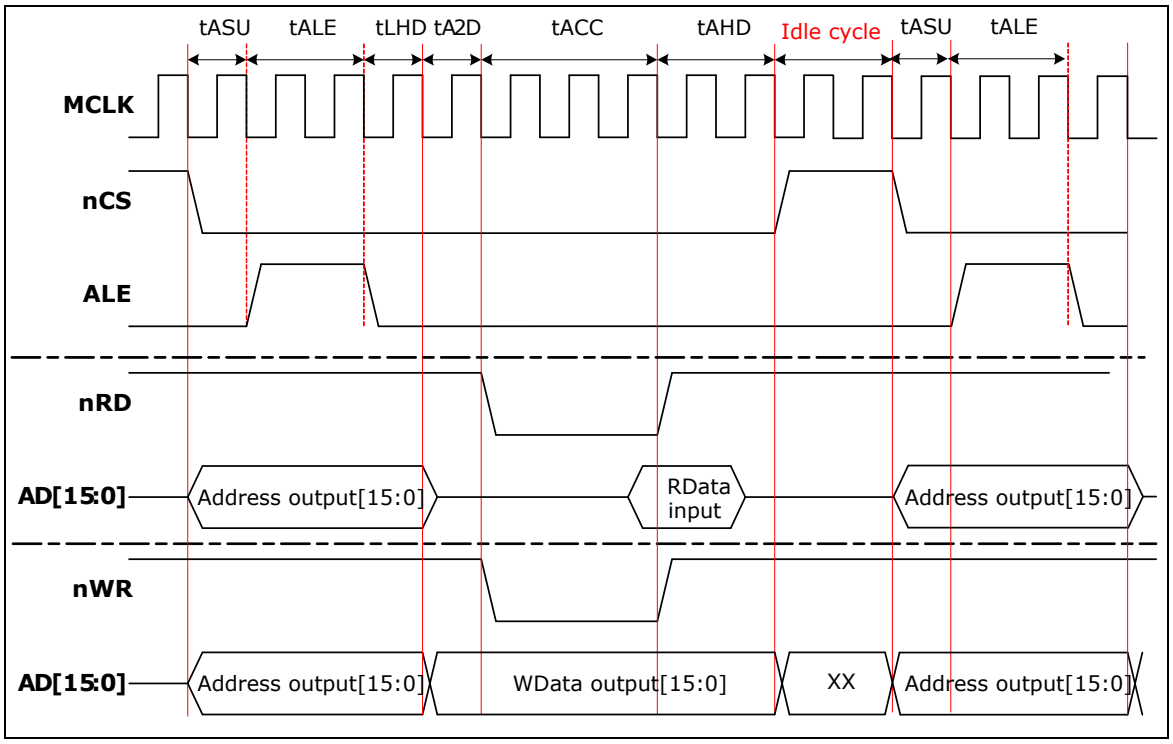



Figure 5-31 Timing Control Waveform for Insert Idle Cycle

There are three conditions that EBI can insert idle cycle by timing control:

- After write access
- After read access and before next read access
- After read access and before next write access

By setting ExtIW2X, ExtIR2R, and ExtIR2W of register EXTIME, the time of idle cycle can be specified from 0~15 MCLK.

## 5.9 FLASH Memory Controller (FMC)

### 5.9.1 Overview

This chip is equipped with 32K/64K/123K bytes on-chip embedded Flash EPROM for application program memory (APROM) that can be updated through ISP/IAP procedure. In System Programming (ISP) function enables user to update program memory when chip is soldered on PCB. After chip powered on Cortex-M0 CPU fetches code from APROM or LDROM decided by boot select (CBS) in Config0. By the way, this chip also provides DATA Flash Region, the data flash is shared with original program memory and its start address is configurable and defined by user in Config1. The data flash size is defined by user application request.

### 5.9.2 Features

- AHB interface compatible
- Run up to 42 MHz with zero wait state for discontinuous address read access
- 32/64/123KB application program memory (APROM)
- 4KB in system programming (ISP) loader program memory (LDROM)
- Programmable data flash start address and memory size with 512 bytes page erase unit
- In System Program (ISP)/In Application Program (IAP) to update on chip Flash EPROM

### 5.9.3 Block Diagram

The flash memory controller consists of AHB slave interface, ISP control logic, writer interface and flash macro interface timing control logic. The block diagram of flash memory controller is shown as follows.

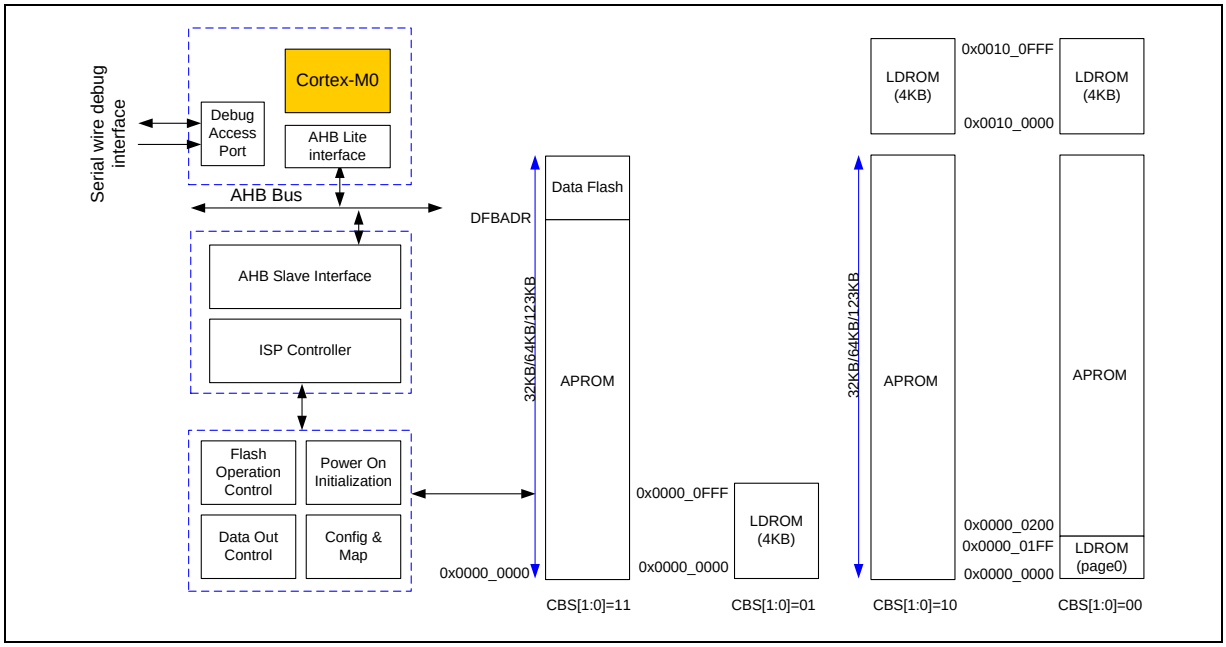



Figure 5-32 Block Diagram of Flash Memory Controller

### 5.9.4 Functional Description

#### 5.9.4.1 Flash Memory Organization

The flash memory consists of application program memory (32KB/64KB/123KB), data flash, ISP loader program memory, user configuration. User configuration block provides several bytes to control system logic, like flash security lock, boot select, Brown-out voltage level, data flash base address, and so on. It works like a fuse for power on setting. It is loaded from flash memory to its corresponding control registers during chip powered on. User can set these bits according to application request by writer before chip is mounted on PCB. The data flash start address and its size can defined by user application.

| Block Name | Size | Start Address | End Address |
|---|---|---|---|
| APROM | (32-0.5*N)KB /<br>(64-0.5*N)KB /<br>(123-0.5*N)KB | 0x0000_0000 | DFBADR-1 (if DFEN=0) |
| Data Flash | 0.5*N KB | DFBADR | 0x0000_7FFF /<br>0x0000_FFFF /<br>0x0001_EBFF |
| Reserved for future use | 901KB | 0x0001_EC00 | 0x000F_FFFF |
| LDROM | 4 KB | 0x0010_0000 | 0x0010_0FFF |
| User Configuration | 2 words | 0x0030_0000 | 0x0030_0004 |

Table 5-7 Memory Access Map

| Part Number | NANO1XX-XCXBN | NANO1XX-XDXBN | NANO1XX-XEXBN |
|---|---|---|---|
| | Flash ROM:32KB | Flash ROM:64KB | Flash ROM:128KB |
| APROM size | (32-0.5*N) KB | (64-0.5*N) KB | (123-0.5*N) KB |
| Data Flash size | (0.5*N) KB | (0.5*N) KB | (0.5*N) KB |
| LDROM | 4 KB | 4 KB | 4 KB |

Table 5-8 Flash Size

The Flash memory organization is shown below.

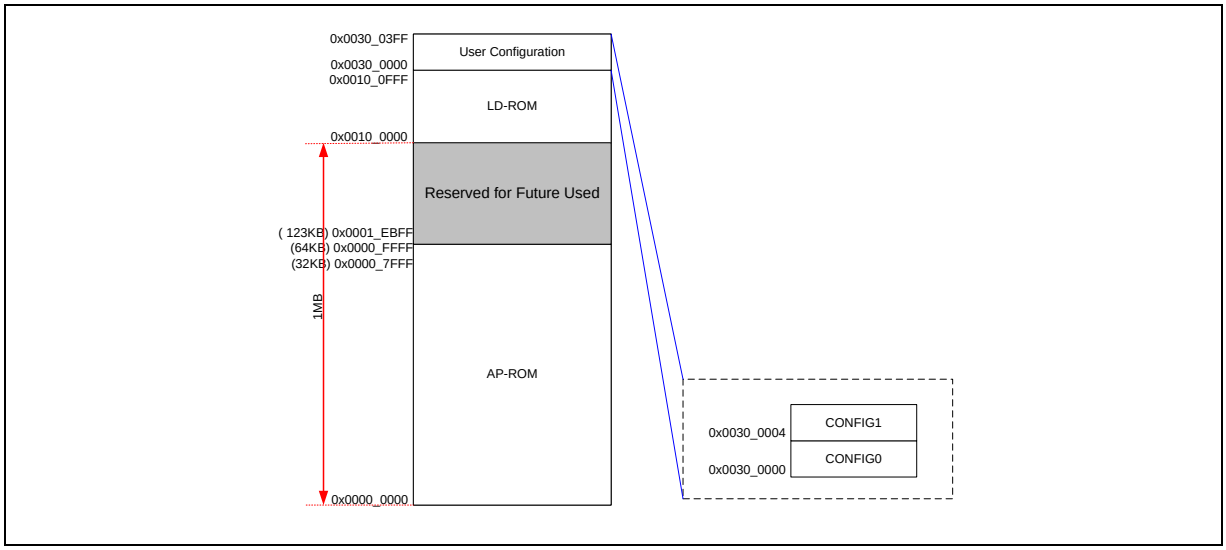



Figure 5-33 Flash Memory Organization

5.9.4.2 Boot Selection

This chip provides in system programming (ISP) feature for user to update program memory when chip is mounted on PCB. A dedicated 4KB program memory (LDROM) is used to store ISP firmware. Users can select to start program from APROM or LDROM by (CBS) in Config0.

| CBS[1:0] | Boot Selection |
|---|---|
| 11 | CPU booting from APROM, flash access range including APROM and Data Flash; LDROM can not be accessed directly, except by through ISP.<br>APROM is write-protected in this mode. |
| 01 | CPU booting from LDROM, flash access range only 4KB LDROM; APROM can not be accessed directly, except by through ISP.<br>APROM can be updated in this mode. |
| 10 | CPU booting from APROM, flash access range including LDROM and APROM<br>APROM can be updated in this mode.<br>LDROM address is mapping to 0x0010_0000~0x0010_0FFF<br>The address 0x0000_0000 ~ 0x0000_01FF mapping can be changed to LDROM by though ISP command. |
| 00 | CPU booting from LDROM, flash access range including LDROM and most of APROM (all except page0, because the address is mapping to LDROM)<br>APROM can be updated in this mode.<br>LDROM address is mapping to 0x0010_0000 ~ 0x0010_0FFF, and also the first 512 bytes of LDROM is mapping to the address 0x0000_0000 ~ 0x0000_01FF.<br>The address 0x0000_0000 ~ 0x0000_01FF mapping can be changed to APROM by though ISP command. |

Table 5-9 Boot Selection

| CBS[1:0] | Boot from | Vector Re-map | Run in LDROM Write to APROM | Run in APROM Write to LDROM | Run in LDROM Write to LDROM | Run in APROM Write to APROM |
|---|---|---|---|---|---|---|
| 11 | APROM | - | - | Yes | - | - |
| 01 | LDROM | - | Yes | - | - | - |
| 10 | APROM | Yes | - | Yes | - | Yes |
| 00 | LDROM | Yes | Yes | - | Yes | Yes |

Table 5-10 Boot Selection and Supports Function

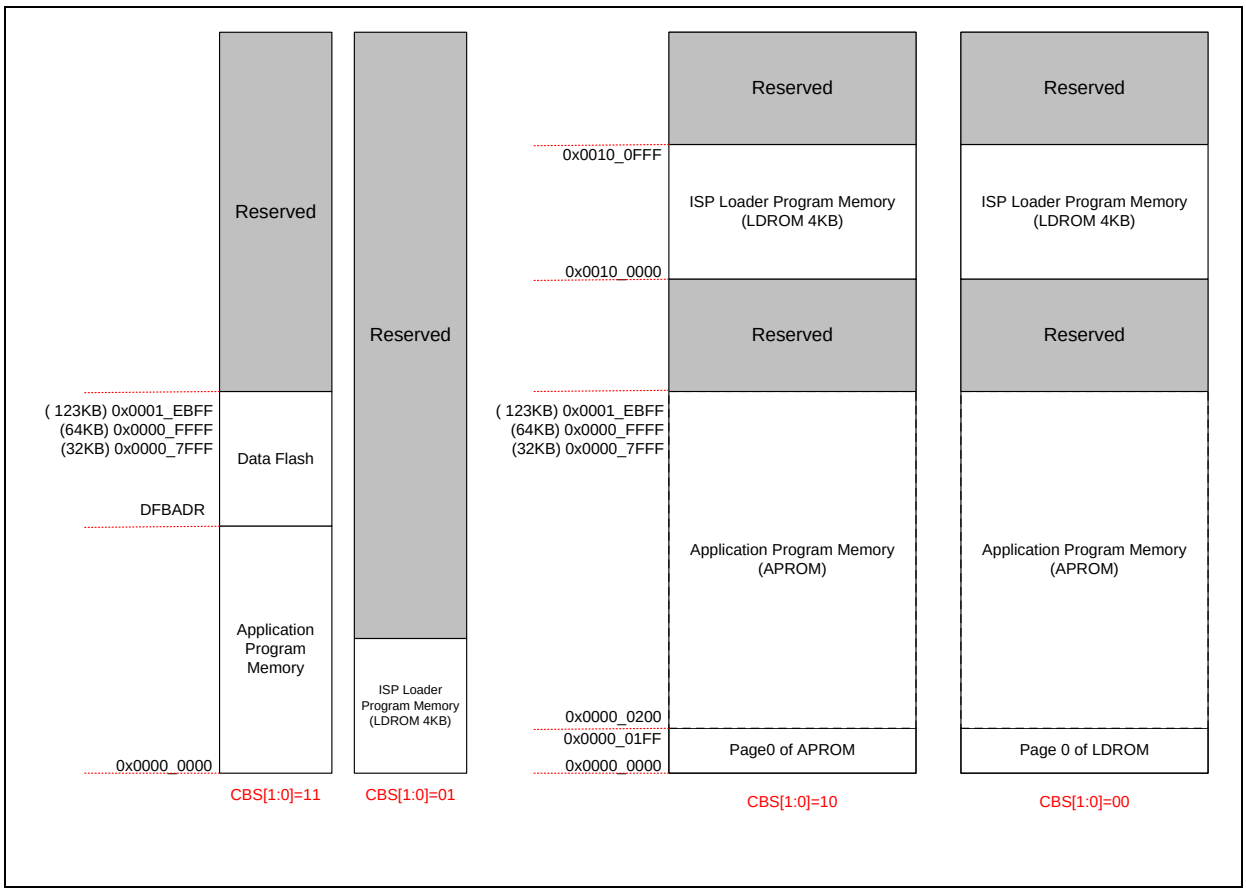



Figure 5-34 Flash Memory Mapping of CBS in CONFIG0

5.9.4.3 Data Flash

This chip provides data flash for user to store data. It is read/written through ISP procedure. The size of each erase unit is 512 bytes. When a word will be changed, all 128 words need to be copied to another page or SRAM in advance. The data flash base address is defined by DFBADR

if DFEN bit in Config0 is enabled and application program memory size is (64-0.5*N)KB for 64KB flash, (32-0.5*N)KB for 32KB flash and data flash size is 0.5*N KB.

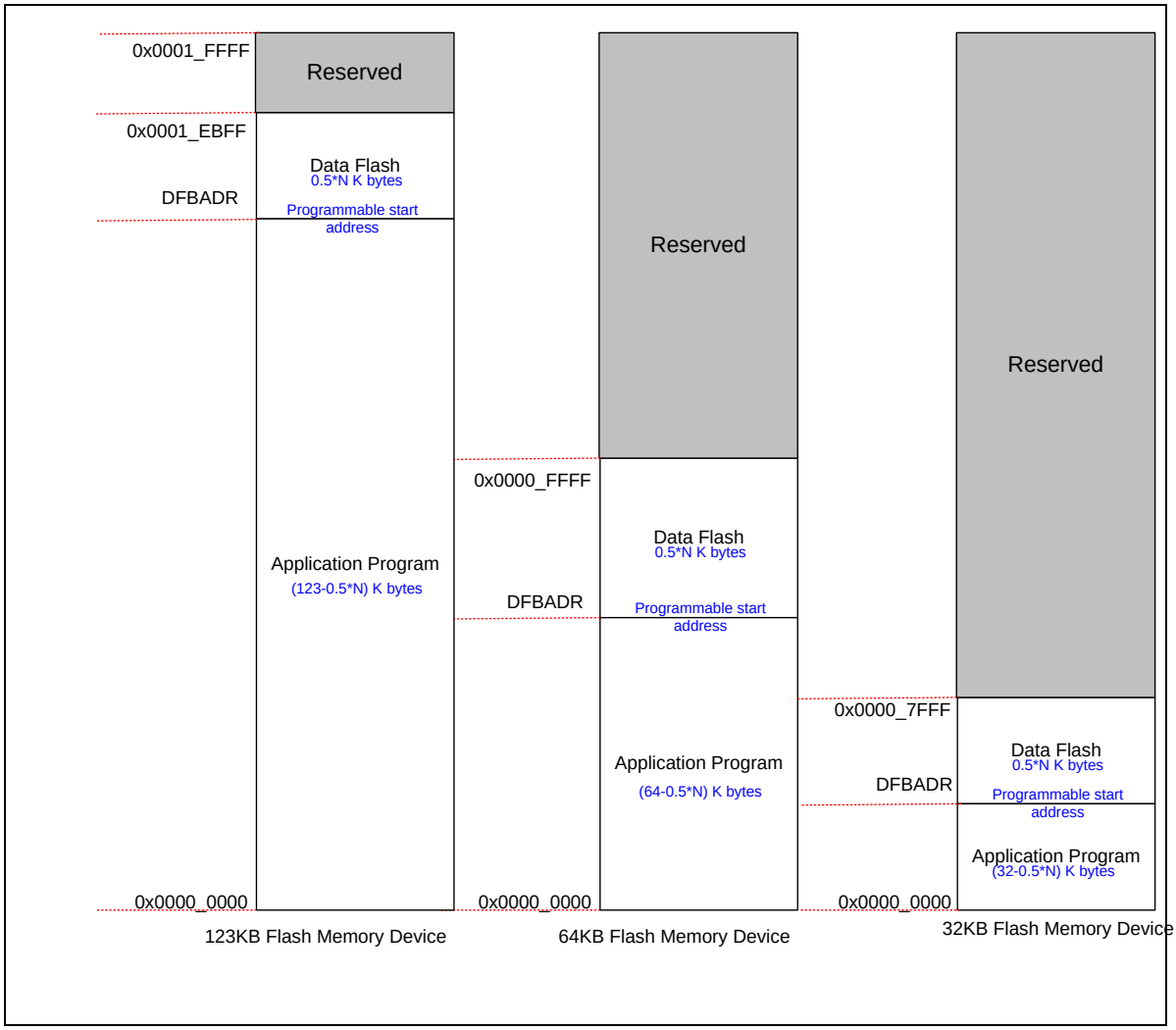



Figure 5-35 32/64/123KB Flash Memory Structure

### 5.9.4.4 User Configuration

**Config0 (Address = 0x0030_0000)**

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| CWDT_EN | - | | | | CFOSC | - | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| - | | | CBORST | | - | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| - | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| CBS | | - | | | | LOCK | DFEN |

| Config0 | Address = 0x0030_0000 | |
|---|---|---|
| **Bits** | **Description** | |
| [31] | **CWDT_EN** | **Force Watchdog Timer Clock On**<br>0 = Forcing the clock of Watchdog Timer to be enabled even if WDT_CTL[WTE] is set to 0.<br>1 = The clock of Watchdog Timer can be disabled by setting WDT_CTL[WTE] to 0. |
| [30:27] | **-** | **Reserved for Future Use** |
| [26] | **CFOSC** | **CPU Clock Source Selection After Reset**<br>**CFOSC** / **Clock Source**<br>0 / External 12 MHz crystal clock (HXT)<br>1 / Internal RC 12 MHz oscillator clock (HIRC)<br>The value of CFOSC will be load to CLKSEL0.HCLK_S[2] in system register after any reset occurs. |
| [25:21] | **-** | **Reserved for future use** |
| [20:19] | **CBORST** | **Brown-out Reset Enable Selection**<br>**CBORST[1:0]** / **Brown-out Reset Selection**<br>00 / BOD17 reset enable<br>01 / BOD20 reset enable<br>10 / BOD25 reset enable<br>11 / Disable all BOD function |
| [18:8] | **-** | **Reserved for Future Use** |

<table>
<tr><td>[7:6]</td><td>CBS</td><td>CONFIG Boot Selection
<table>
<tr><th>CBS[1:0]</th><th>Boot Selection</th></tr>
<tr><td>11</td><td>APROM(LDROM invisible)<br>CPU booting from APROM, flash access range including APROM and Data Flash; LDROM cannot be access directly, except by through ISP.<br>APROM is write-protected in this mode.</td></tr>
<tr><td>01</td><td>LDROM(APROM invisible)<br>CPU booting from LDROM, flash access range only LDROM 4KB; APROM cannot be access directly, except by through ISP.<br>APROM can be updated in this mode.</td></tr>
<tr><td>10</td><td>APROM<br>CPU booting from APROM, flash access range including LDROM and APROM<br>APROM can be updated in this mode.<br>LDROM address is mapping to 0x0010_0000~0x0010_0FFF<br>The address 0x0000_0000 ~ 0x0000_01FF mapping can be change to LDROM by though ISP command.</td></tr>
<tr><td>00</td><td>LDROM<br>CPU booting from LDROM, flash access range including LDROM and most of APROM (all but except page0, because the address is mapping to LDROM)<br>APROM can be updated in this mode.<br>LDROM address is mapping to 0x0010_0000 ~ 0x0010_0FFF, and also the first 512 bytes of LDROM is mapping to the address 0x0000_0000 ~ 0x0000_01FF.<br>The address 0x0000_0000 ~ 0x0000_01FF mapping can be change to APROM by though ISP command.</td></tr>
</table></td></tr>
<tr><td>[5:2]</td><td>-</td><td>Reserved for future use</td></tr>
<tr><td>[1]</td><td>LOCK</td><td>Security Lock<br>0 = Flash data is locked<br>1 = Flash data is not locked.<br>When flash data is locked, only Device ID, Config0 and Config1 can be read by writer and ICP through serial debug interface. Others data is locked as 0xFFFFFFFF. ISP can read data anywhere regardless of LOCK bit value.</td></tr>
<tr><td>[0]</td><td>DFEN</td><td>Data Flash Enable<br>0 = Data flash Enabled.<br>1 = Data flash Disabled.<br>This bit is valid when CBS[1:0] = 11 or 10.</td></tr>
</table>

**Config1 (Address = 0x0030_0004)**

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

| Config1 | Address = 0x0030_0004 | |
|---|---|---|
| **Bits** | **Description** | |
| [31:20] | Reserved | **Reserved**<br>It is mandatory to program 0x00 to these Reserved bits |
| [19:0] | DFBA | **Data Flash Base Address**<br>The data flash base address is defined by user. Since on chip flash erase unit is 512-byte, it is mandatory to keep bit 8-0 as 0. |

5.9.4.5 In System Program (ISP)

The application program memory and data flash supports both hardware programming mode and In System Programming (ISP) mode. Hardware programming mode uses gang-writers to reduce programming costs and time to market while the products enter into the mass production state. However, if the product is just under development or the end product needs firmware updating in the hand of an end user, the hardware programming mode will make repeated programming difficult and inconvenient. ISP method makes it easy and possible. This chip supports ISP mode allowing a device to be reprogrammed under software control. Furthermore, the capability to update the application firmware makes wide range of applications possible.

ISP is performed without removing the microcontroller from the system. Various interfaces enable LDROM firmware to get new program code easily. The most common method to perform ISP is via UART along with the firmware in LDROM. General speaking, PC transfers the new APROM code through serial port. Then LDROM firmware receives it and re-programs into APROM through ISP commands. Nuvoton provides ISP firmware and PC application program for this chip. It makes users quite easy to perform ISP through Nuvoton ISP tool.

**ISP Procedure**

This chip supports booting from APROM or LDROM initially defined by user configuration bit (CBS). If user wants to update application program in APROM, user can write BS=1 and starts software reset to make chip boot from LDROM. The first step to start ISP function is to write ISPEN bit to 1. S/W is required to write RegLockAddr register in Global Control Register (GCR, 0x5000_0100) with 0x59, 0x16 and 0x88 before writing ISPCON register. This procedure is used to protect flash memory from destroying owning to unintended write during power on/off duration.

Several error conditions are checked after software writes ISPGO bit. If error condition occurs, ISP operation is not been started and ISP fail flag will be set instead of. ISPFF flag is cleared by S/W but it will not be overwritten in next ISP operation. The next ISP procedure can be started even ISPFF bit keeps at 1. It is recommended that s/w to check ISPFF bit and clear it after each ISP operation if it is set to 1.

When ISPGO bit is set, CPU will wait for ISP operation finish, during this period; peripheral still keeps working as usual. If any interrupt request occur, CPU will not service it till ISP operation finish.

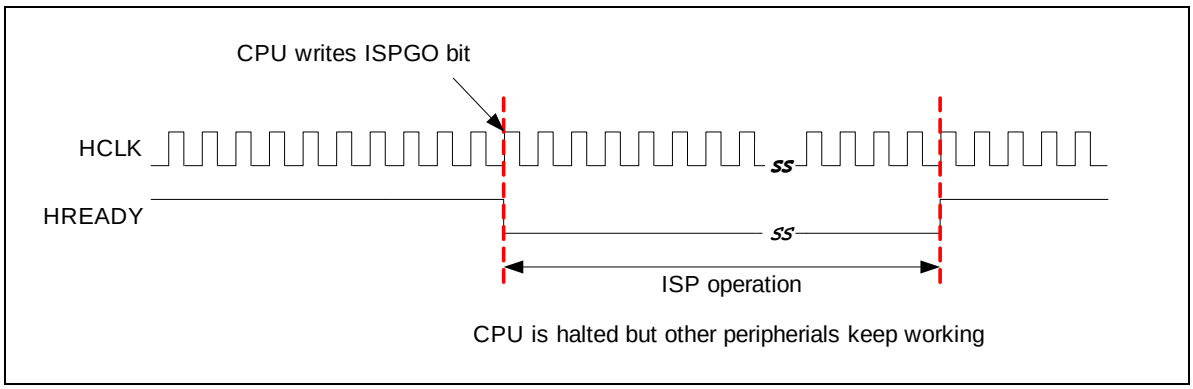



Figure 5-36 CPU Halt during ISP Operation

Note that this chip allows user to update CONFIG value by ISP, but for application program code security issue, s/w is required to erase APROM by page erase before erase CONFIG. Otherwise, erase CONFIG will not be allowed.

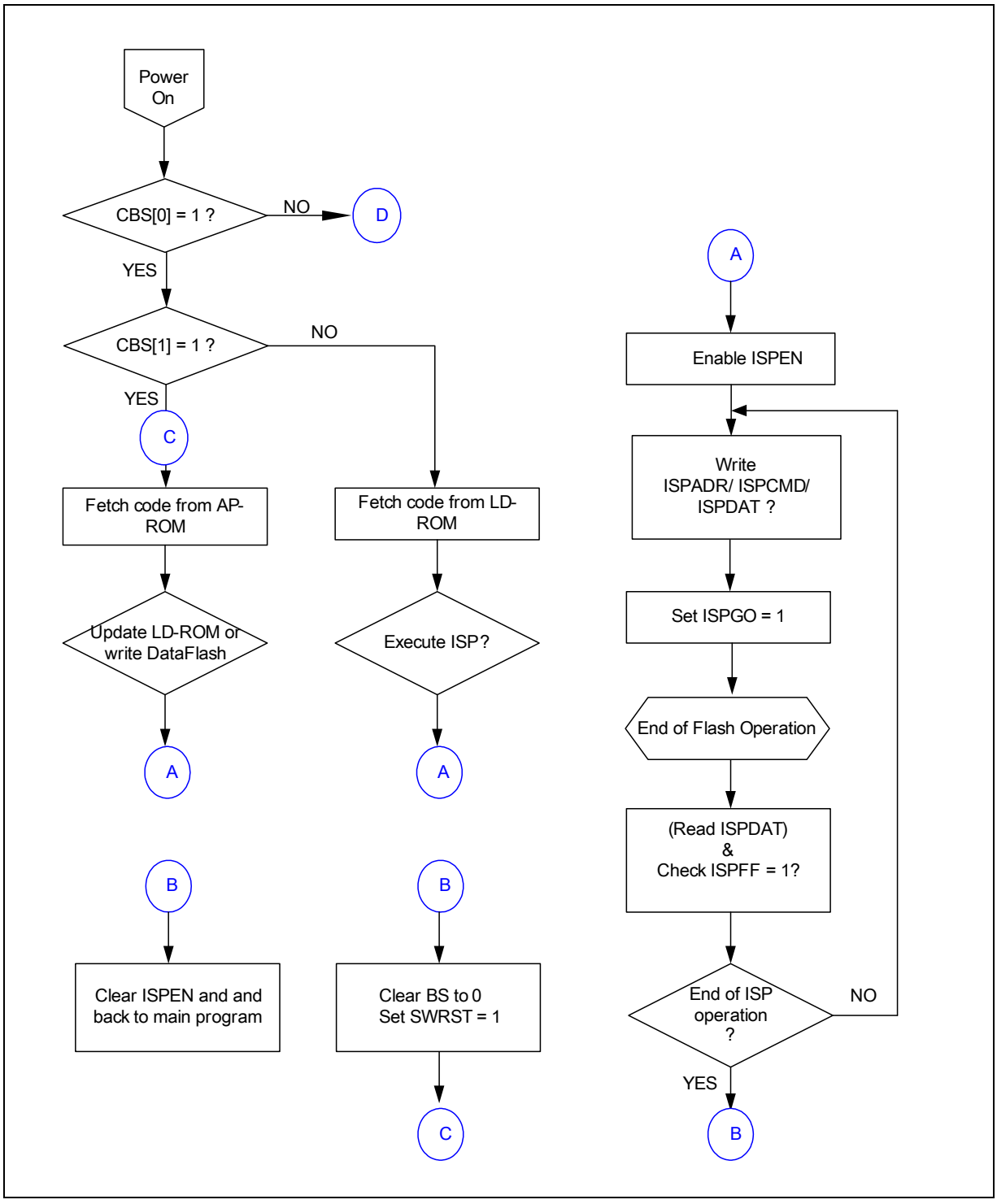



Figure 5-37 ISP Operation Flow

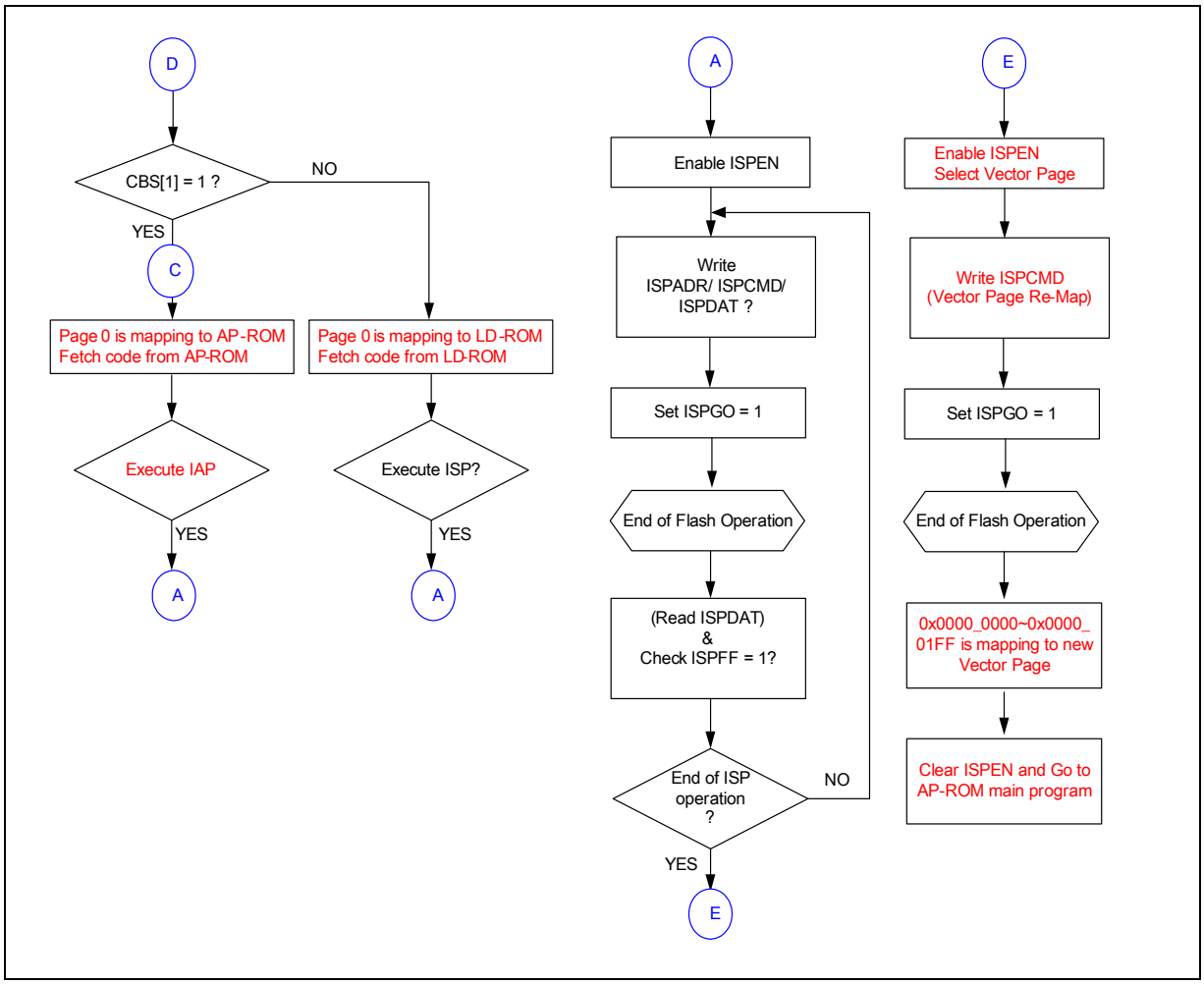



Figure 5-38 ISP Operation Flow (Continued)

| ISP Mode | ISPCMD | | | ISPADR | | | ISPDAT |
|---|---|---|---|---|---|---|---|
| | FOEN | FCEN | FCTRL[3:0] | A21 | A20 | A[19:0] | D[31:0] |
| Standby | 1 | 1 | X | x | x | x | x |
| Read Company ID | 0 | 0 | 1011 | x | x | x | Data out D[31:0] = 0x0000_00DA |
| Read Device ID | 0 | 0 | 1100 | x | x | Address in A[19:0] = 0x00000 | Data out D[31:0]= Device ID |
| Read Unique ID | 0 | 0 | 0100 | x | x | Address in A[19:0] = 0x00000 0x00004 0x00008 | Data out D[31:0]= Unique ID |
| *Read Unique Customer ID | 0 | 0 | 0100 | x | x | Address in A[19:0] = 0x00010 0x00014 0x00018 0x0001C | Data out D[31:0]= Unique Customer ID |
| Vector Page Re-Map | 1 | 0 | 1110 | 0 | A20 | Address in A[19:0] | x |
| FLASH Page Erase | 1 | 0 | 0010 | 0 | A20 | Address in A[19:0] | x |
| FLASH Program | 1 | 0 | 0001 | 0 | A20 | Address in A[19:0] | Data in D[31:0] |
| FLASH Read | 0 | 0 | 0000 | 0 | A20 | Address in A[19:0] | Data out D[31:0] |
| CONFIG Page Erase | 1 | 0 | 0010 | 1 | 1 | Address in A[19:0] | x |
| CONFIG Program | 1 | 0 | 0001 | 1 | 1 | Address in A[19:0] | Data in D[31:0] |
| CONFIG Read | 0 | 0 | 0000 | 1 | 1 | Address in A[19:0] | Data out D[31:0] |

Table 5-11 ISP Operation Command

***** The default value of “Unique Customer ID” is 0xFFFF which is from address 0x00010 to 0x0001C. “Unique Customer ID” only can be configured by Nuvoton, please contact Nuvoton or agent to deal with specific customer ID.

## 5.10 General Purpose I/O Controller

### 5.10.1 Overview

Up to 86 General Purpose I/O pins can be shared with other function pins; it depends on the chip configuration. These 86 pins are arranged in 6 ports named with GPIOA, GPIOB, GPIOC, GPIOD, GPIOE and GPIOF. Ports A ~ E have the maximum of 16 pins while port F have 6 pins. Each one of the 86 pins is independent and has the corresponding register bits to control the pin mode function and data.

The I/O type of each of I/O pins can be independently software configured as input, output, and open-drain mode. Each I/O pin has a very weak individual pull-up resistor which is about 110 KΩ~300 KΩ for VDD from 1.8 V to 3.6 V.

### 5.10.2 Features

- Up to 86 general purpose I/O pins
- Supports Input, Output, Open-drain Operation mode
- Programmable de-bounce timing
- Each I/O pin can be programmed as either edge-trigger or level-sensitive
- Each I/O pin can be programmed as either low-level active or high-level active
- Each I/O pin can be programmed as either falling-edge trigger or rising-edge trigger

### 5.10.3 Block Diagram

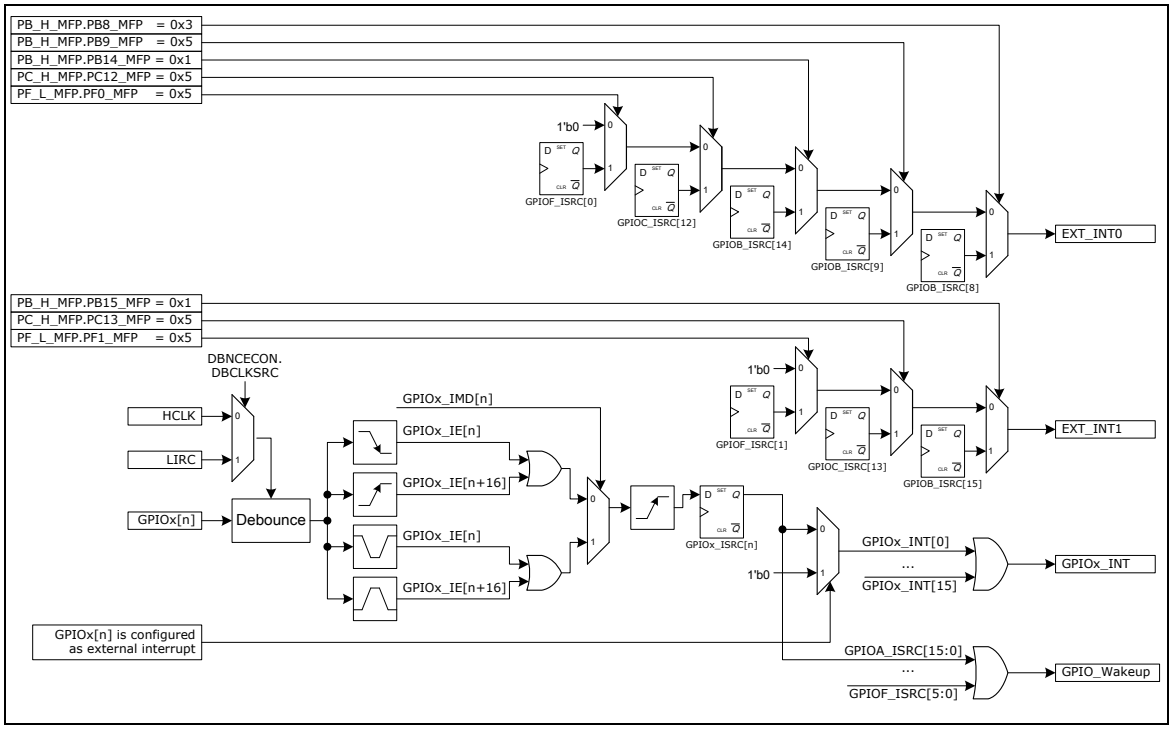



Figure 5-38 GPIO Block Diagram

### 5.10.4 Functional Description

5.10.4.1 Input Mode Explanation

Set GPIOx_PMD (PMDn [1:0]) to 00 the GPIOx port [n] pin is in Input mode and the I/O pin is in

tri-state (high impedance) without output drive capability. The GPIOx_PIN value reflects the status of the corresponding port pins.

#### 5.10.4.2 Output Mode Explanation

Set GPIOx_PMD (PMDn [1:0]) to 01 the GPIOx port [n] pin is in Output mode and the I/O pin supports digital output function with source/sink current capability. The bit value in the corresponding bit [n] of GPIO_DOUT is driven on the pin.

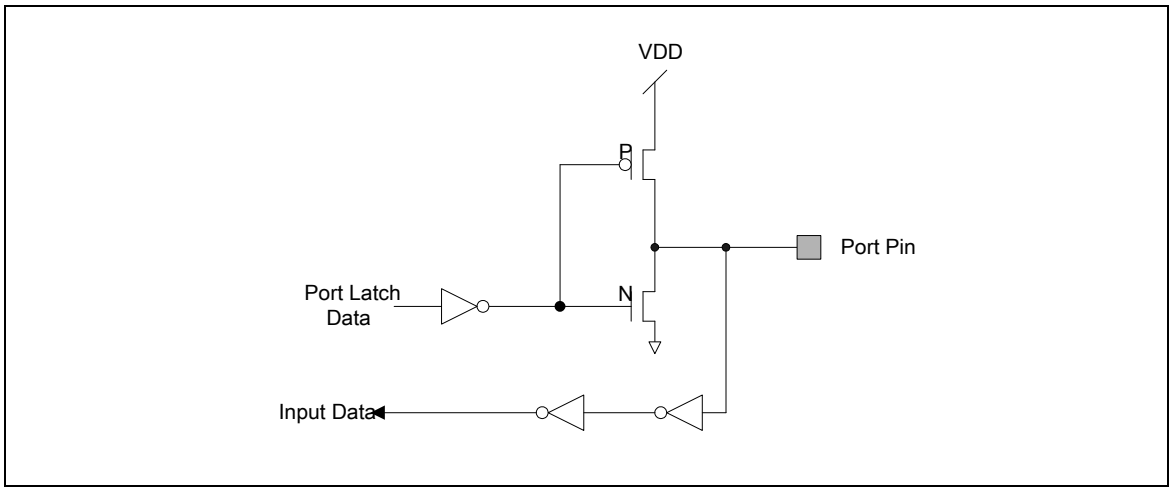



Figure 5-39 Push-Pull Output

#### 5.10.4.3 Open-Drain Mode Explanation

Set GPIOx_PMD (PMDn [1:0]) to 10 the GPIOx port [n] pin is in Open-Drain mode and the I/O pin supports digital output function but only with sink current capability, an additional pull-up resister is needed for driving high state. If the bit value in the corresponding bit [n] of GPIOx_DOUT is “0”, the pin drive a “low” output on the pin. If the bit value in the corresponding bit [n] of GPIOx_DOUT is “1”, the pin output drives high that is controlled by the internal pull-up resistor or the external pull high resistor.

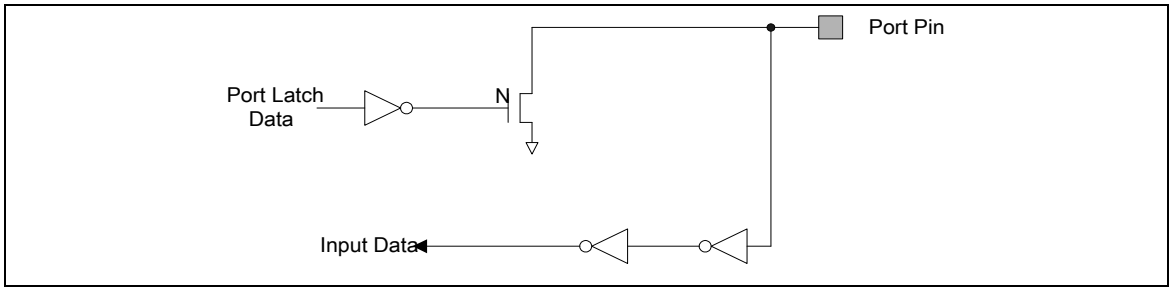



Figure 5-40 Open-Drain Output

## 5.11 I²C

### 5.11.1 Overview

I²C is a two-wire, bi-directional serial bus that provides a simple and efficient method of data exchange between devices. The I²C standard is a true multi-master bus including collision detection and arbitration that prevents data corruption if two or more masters attempt to control the bus simultaneously. Serial, 8-bit oriented bi-directional data transfers can be made up to 1.0 Mbps.

Data is transferred between a Master and a Slave synchronously to SCL on the SDA line on a byte-by-byte basis. Each data byte is 8-bit long. There is one SCL clock pulse for each data bit with the MSB being transmitted first. An acknowledge bit follows each transferred byte.

A transition on the SDA line while SCL is high is interpreted as a command (START or STOP). Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL.

The controller's on-chip I²C logic provides the serial interface that meets the I²C bus standard mode specification. The I²C controller handles byte transfers autonomously. Pull up resistor is needed for I²C operation as these are open drain pins.

The I²C controller is equipped with two slave address registers. The contents of the registers are irrelevant when I²C is in Master mode. In the Slave mode, the seven most significant bits must be loaded with the user's own slave address. The I²C hardware will react if the contents of I2CADDR are matched with the received slave address.

This controller supports the "General Call (GC)" function. If the GC bit is set this controller will respond to General Call address (00H). Clear GC bit to disable general call function. When GC bit is set and the I²C is in Slave mode, it can receive the general call address which is equal to 00H after master sends general call address to the I²C bus, then it will follow status of GC mode. If it is in Master mode, the ACK bit must be cleared when it sends general call address of 00H to the I²C bus.

The I²C-bus controller supports multiple address recognition with two address mask register. When the bit in the address mask register is set to one, it means the received corresponding address bit is don't-care. If the bit is set to zero, that means the received corresponding register bit should be exact the same as address register.

### 5.11.2 Features

- Acts as Master or Slave mode
- Bidirectional data transfer between masters and slaves
- Multi-master bus (no central master)
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus
- Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer
- One built-in 14-bit time-out counter requesting the $I^2C$ interrupt if the $I^2C$ bus hangs up and timer-out counter overflows.
- Programmable clock divider allows versatile rate control
- Supports 7-bit addressing mode
- Supports multiple address recognition ( Two slave addresses with mask option)
- Supports Power-down wake-up function

### 5.11.3 Functional Description

#### 5.11.3.1 $I^2C$ Protocol

Normally, a standard communication consists of four parts:

- START or Repeated START signal generation
- Slave address transfer
- Data transfer

#### 5.11.3.2 STOP signal generation

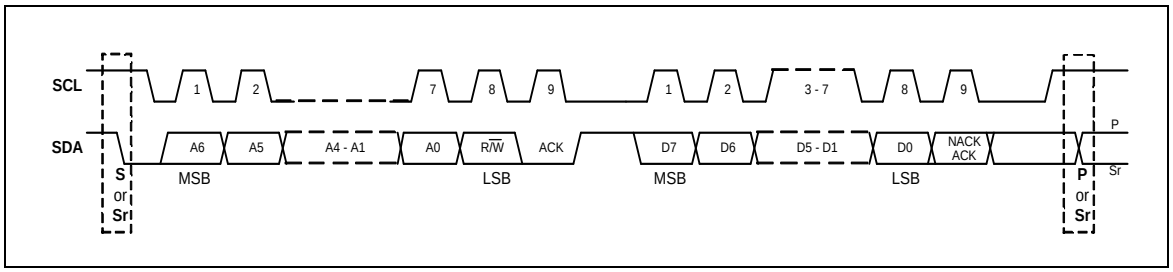



Figure 5-41 $I^2C$ Protocol

#### 5.11.3.3 Data transfer on the $I^2C$-bus

A master-transmitter addresses a slave receiver with a 7-bit address.

The transfer direction is not changed.

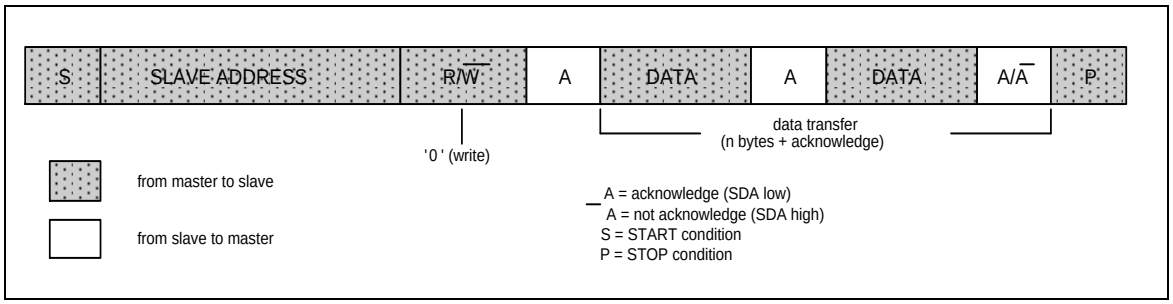



Figure 5-42 I²C Master Transmits Data to Slave

A master reads a slave immediately after the first byte (address)

The transfer direction is changed.

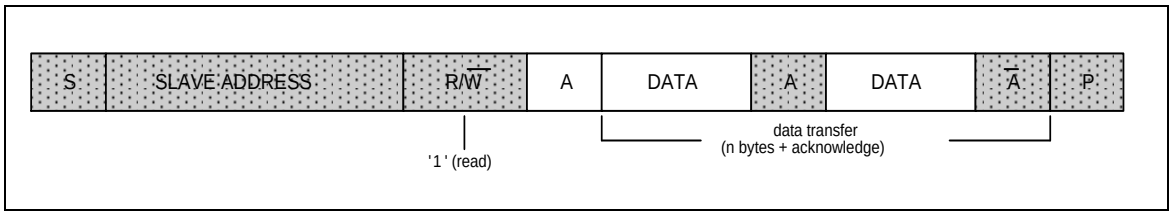



Figure 5-43 I²C Master Reads Data from Slave

5.11.3.4 START or Repeated START signal

When the bus is free/idle, meaning no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the S-bit, is defined as a HIGH to LOW transition on the SDA line while SCL is HIGH. The START signal denotes the beginning of a new data transfer.

A Repeated START (Sr) is a START signal without first generating a STOP signal. The master uses this method to communicate with another slave or the same slave in a different transfer direction (e.g. from writing to a device to reading from a device) without releasing the bus.

5.11.3.5 STOP signal

The master can terminate the communication by generating a STOP signal. A STOP signal, usually referred to as the P-bit, is defined as a LOW to HIGH transition on the SDA line while SCL is HIGH.

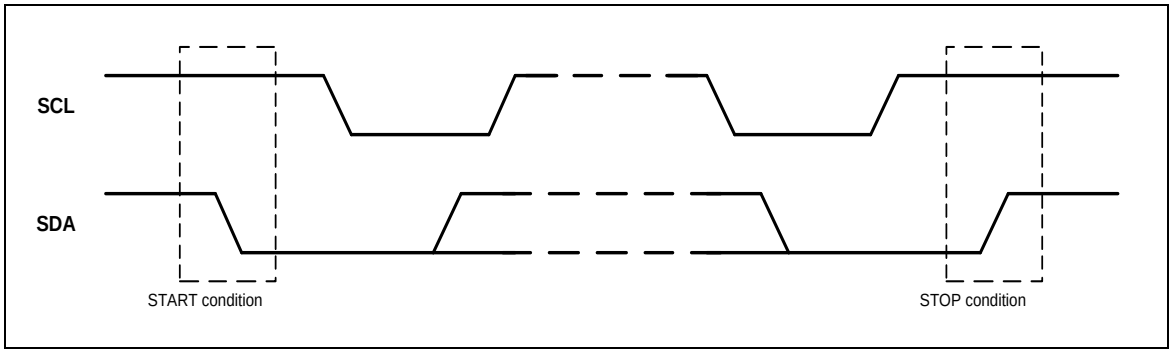



Figure 5-44 I²C START and STOP Conditions

5.11.3.6 Slave Address Transfer

The first byte of data transferred by the master immediately after the START signal is the slave address. This is a 7-bits calling address followed by the R/W bit. The R/W bit signals the slave the

data transfer direction. No two slaves in the system can have the same address. Only the slave with an address that matches the one transmitted by the master will respond by returning an acknowledge bit by pulling the SDA low at the 9th SCL clock cycle.

#### 5.11.3.7 Data Transfer

Once successful slave addressing has been achieved, the data transfer can proceed on a byte-by-byte basis in the direction specified by the RW bit sent by the master. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a Not Acknowledge (NACK), the master can generate a STOP signal to abort the data transfer or generate a Repeated START signal and start a new transfer cycle.

If the master, as the receiving device, does Not Acknowledge (NACK) to the slave, the slave releases the SDA line for the master to generate a STOP or Repeated START signal.

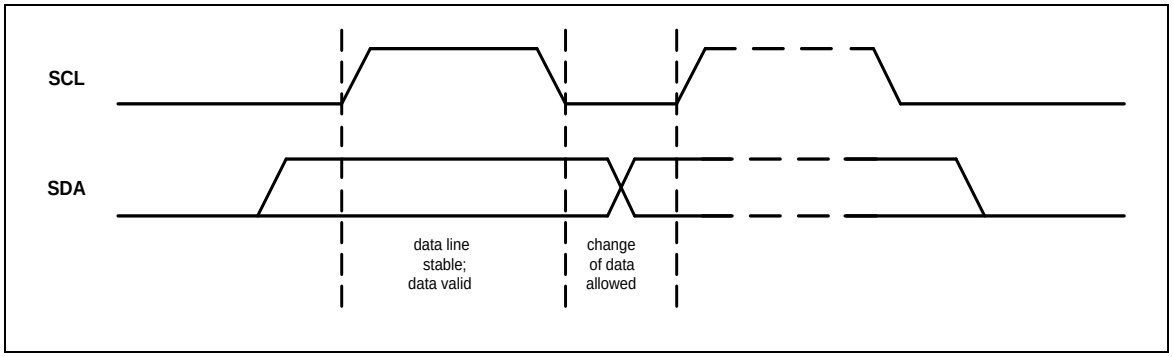



Figure 5-45 Bit Transfer on I2C Bus

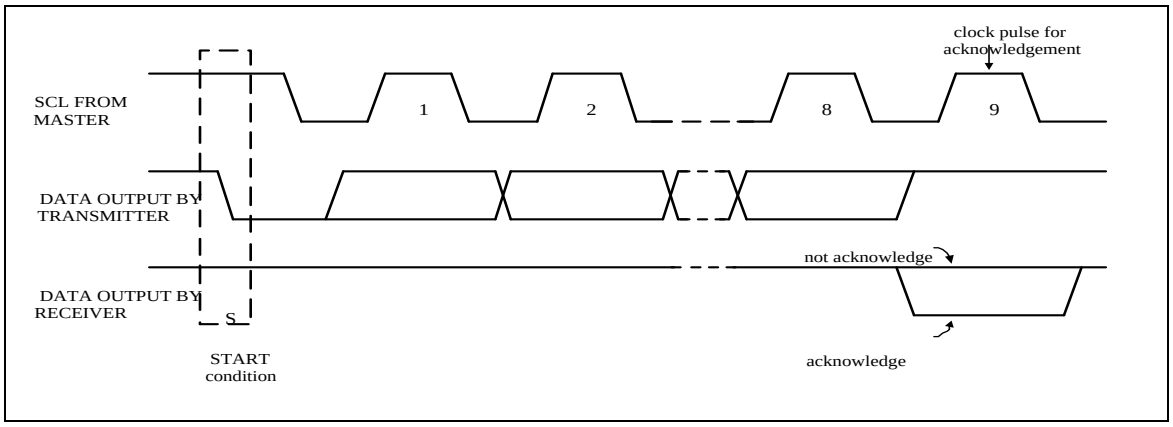



Figure 5-46 Acknowledge on I2C Bus

#### 5.11.3.8 Clock Baud Rate

The data baud rate of I2C is determined by CLK_DIV register when the controller is in a Master mode. It is not important when the controller is in Slave mode. In Slave mode, the controller will automatically synchronize to clock frequency from I2C master device.

The data baud rate of I2C setting is Data Baud Rate of I2C = PCLK /(4x(CLK_DIV+1)). If PCLK=16 MHz, the CLK_DIV = 40 (28H) so that the data baud rate of I2C = 16 MHz/(4X(40+1)) = 97.5 K bits/sec.

#### 5.11.3.9 Time-Out

There is a 14-bit time-out counter which can be used to deal with the I2C bus hang-up. If the time-out counter is enabled, when the bus start signal is detected, the counter starts up counting until counter overflows (TIF=1) and requests I2C interrupt to CPU or there is stop signal being detected. User can also stop counter counting by clearing TOUTEN to 0. When time-out counter

is enabled, setting flag I2C_STS and STAINTSTS to high and the falling edge of I²C bus clock will reset counter and re-start up counting after I2C_STS is cleared or after the falling edge of bus clock. If the I²C bus hangs up, it causes the STATUS and I2C_STS not to be updated for a period. The 14-bit time-out counter may overflow and acknowledge CPU the I²C interrupt. Refer to the following Figure for 14-bit time-out counter. User may clear TIF by write 1 to this bit.

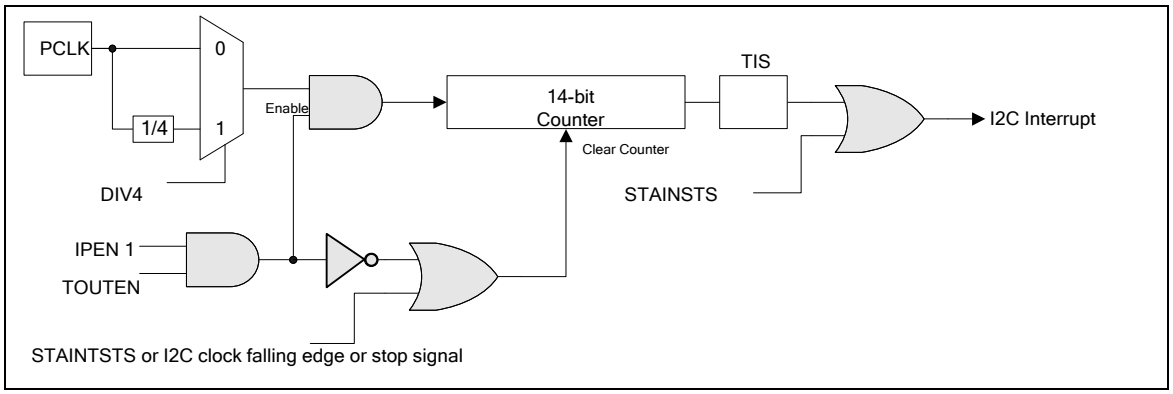



Figure 5-47 I²C Time-out Block Diagram

5.11.3.10 The I2C wake-up control Register (I2CWKUPCON)

When entering Power-down mode, other I²C master can wake up our chip by addressing our I²C device. User must set **I2CWKUPCON[**WKUPEN] before entering Power-down mode.

5.11.3.11 The I2C wake-up status Register (I2CWKUPSTS)

When system is waken up by other I²C master device, WKUPIF is set to indicate this event

5.11.3.12 Mode Operation

There are five operation modes in the I²C controller. They are Master transmitter, Master receiver, Slave transmitter, Slave receiver, and GC call. Bits START, STOP and ACK in I2CCON register will determine the next state of the controller after I2C_STS bit is cleared. Upon completion of the new action, a new status code will be updated and the I2C_STS and STAINTSTS bit will be set. If the I²C interrupt control bit INTEN is set, appropriate action or software branch of the new status code can be performed in the Interrupt service routine.

In a given application, I²C controller may operate as a master or as a slave. In Slave mode, it looks for its own slave address and the general call address. If one of these addresses is detected, and if the slave is willing to receive or transmit data from/to master (by setting the ACK bit), acknowledge pulse will be transmitted out $^{o}$n the 9$^{th}$ clock and an interrupt is requested if interrupt is enabled. When the controller wishes to become the bus master, the hardware will waits until the bus is free before entering Master mode so that a possible slave action is not interrupted. If bus arbitration is lost in Master mode, it switches to Slave mode immediately and can detect its own slave address in the same serial transfer.

5.11.3.13 Master Transmitter Mode

In master transmitter mode, the serial data output through SDA while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the data direction bit. In this case the data direction bit (R/W) will be logic 0, and it is represented by “W” in the flow diagrams. Thus the first byte transmitted is SLA+W. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an acknowledge bit is received. START and STOP conditions are output to indicate the beginning and the end of a serial transfer.

5.11.3.14 Master Receiver Mode

In the master receiver mode, the data direction bit (R/W) will be logic 1, and it is represented by "R" in the flow diagrams. Thus the first byte transmitted is SLA+R. Serial data is received via SDA while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are output to indicate the beginning and end of a serial transfer.

5.11.3.15 Slave Receiver Mode

In the salve receiver mode, the serial data and the serial clock are received through SDA and SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit.

5.11.3.16 Slave Transmitter Mode

In slave transmitter mode, the first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. Serial data is transmitted via SDA while the serial clock is input through SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer.

Data transfers in each mode are shown in the following figures.

**Note:**

- I2CON = I2CCON
- I2DAT = I2CDATA
- STA = START
- STO = STOP
- AA = ACK

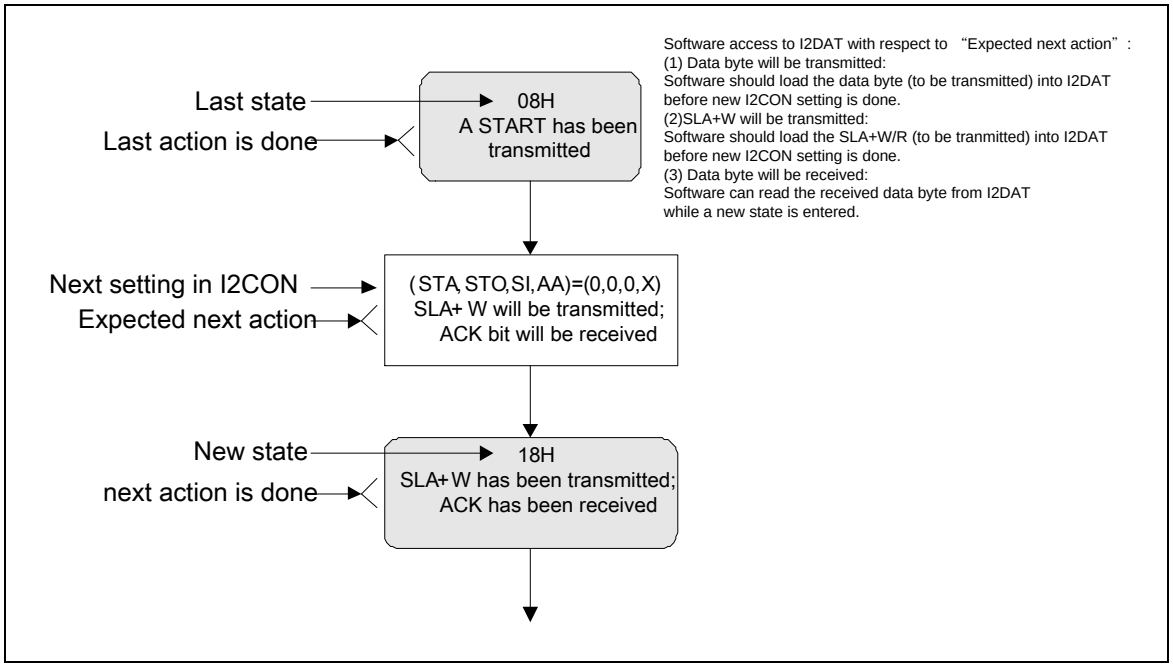



Figure 5-48 I²C Normal Master Transmitter State

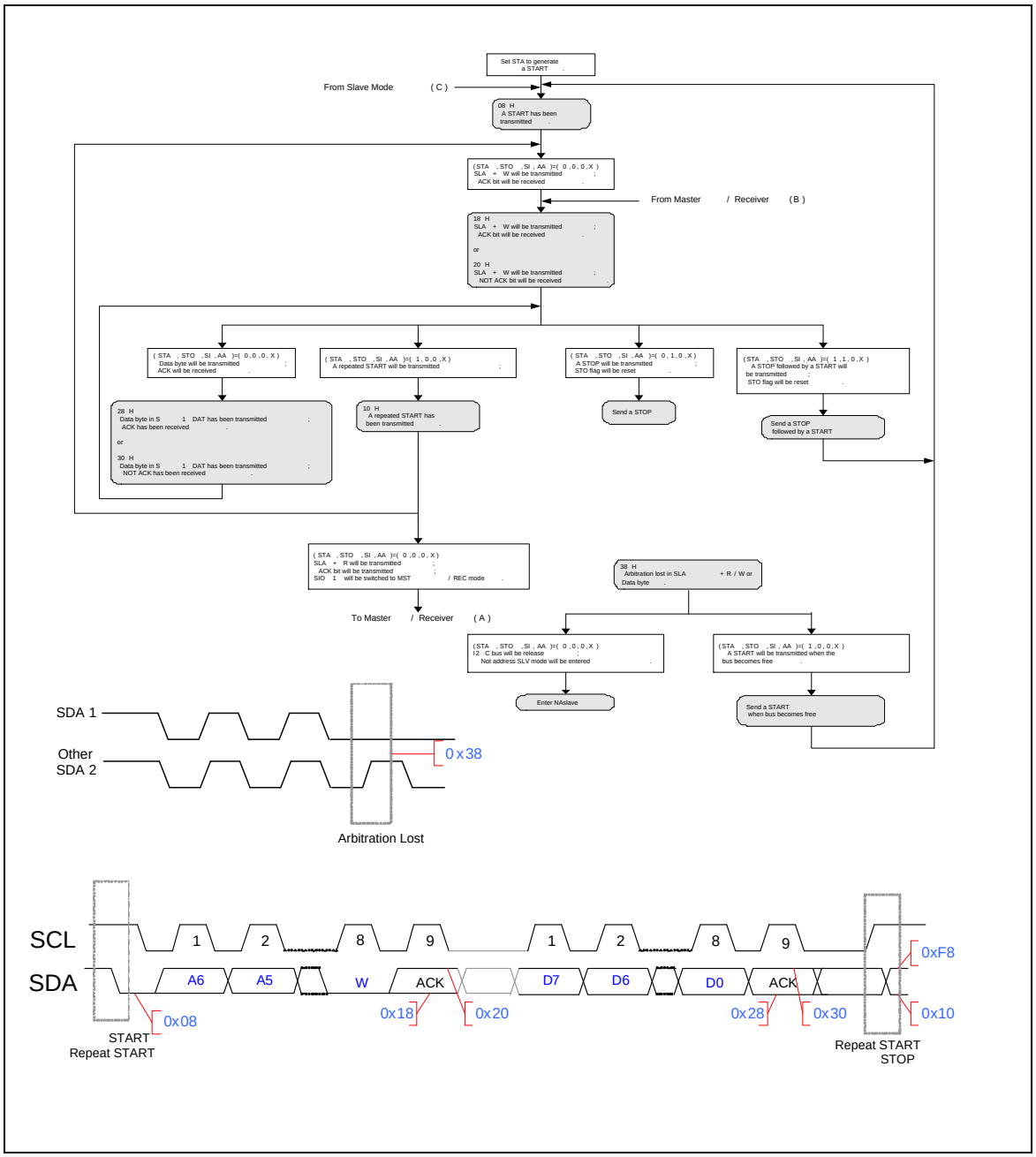



Figure 5-49 I²C Master Transmitter Mode

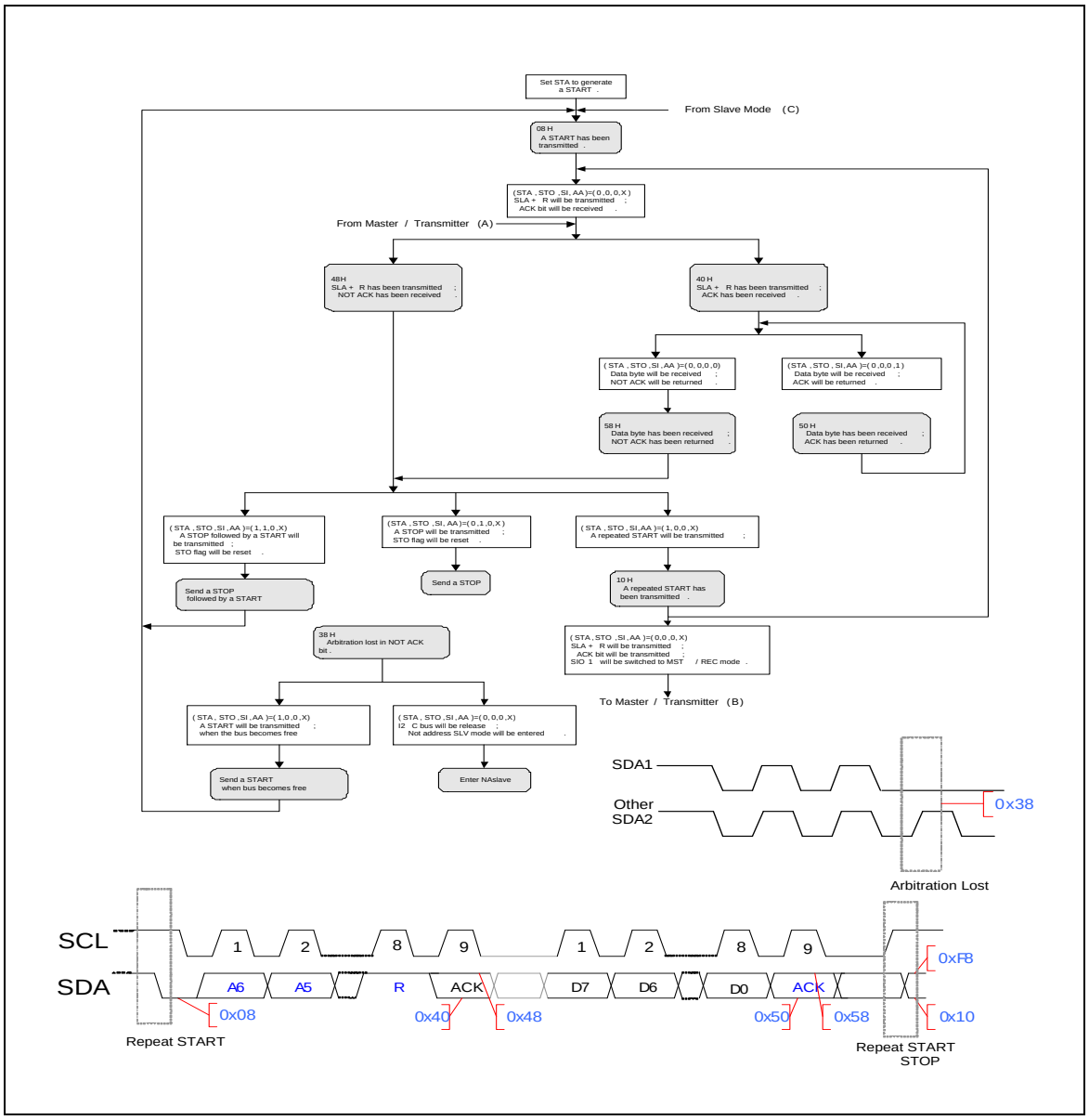



Figure 5-50 I²C Master Receiver Mode

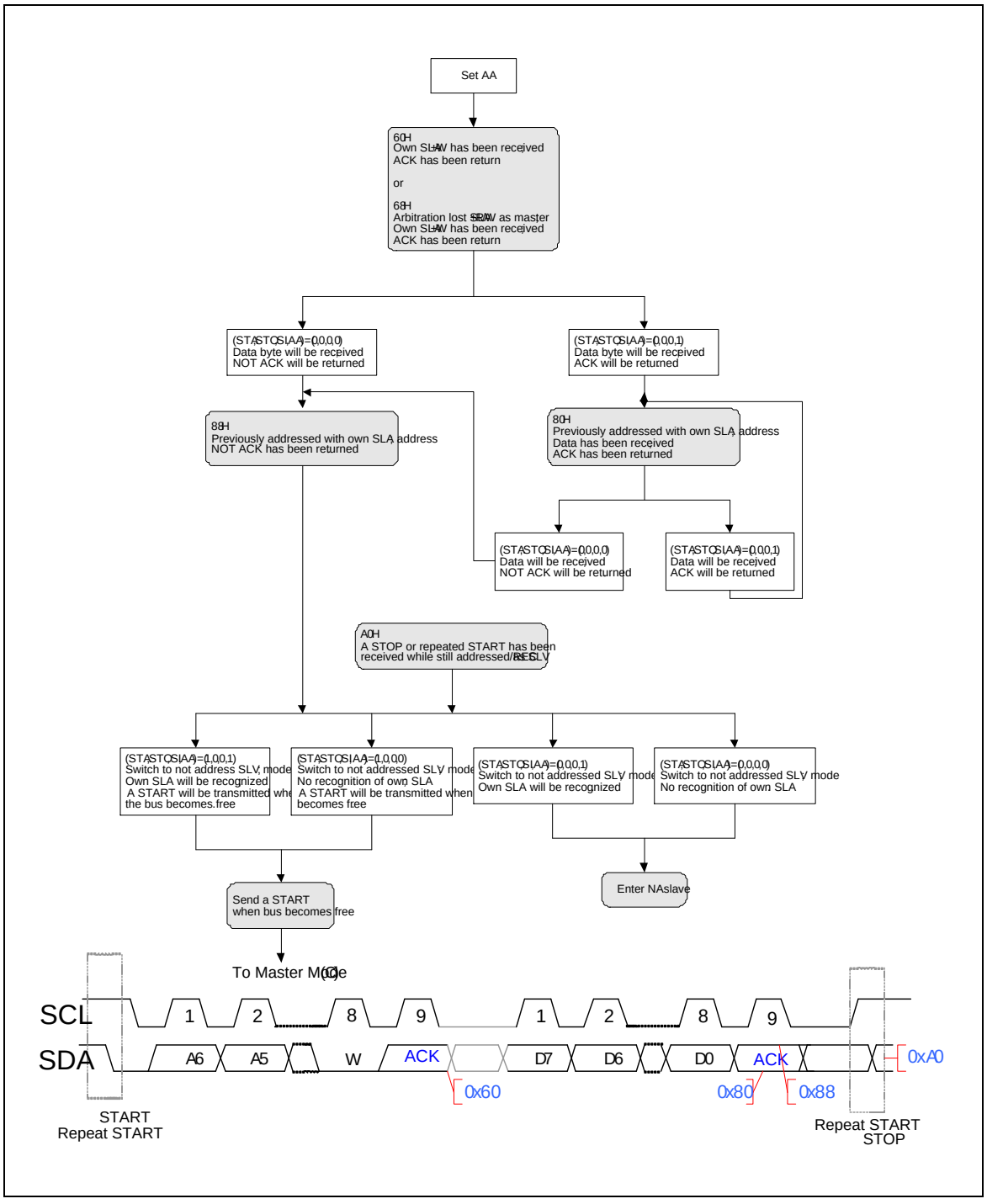



Figure 5-51 I²C Slave Receiver Mode

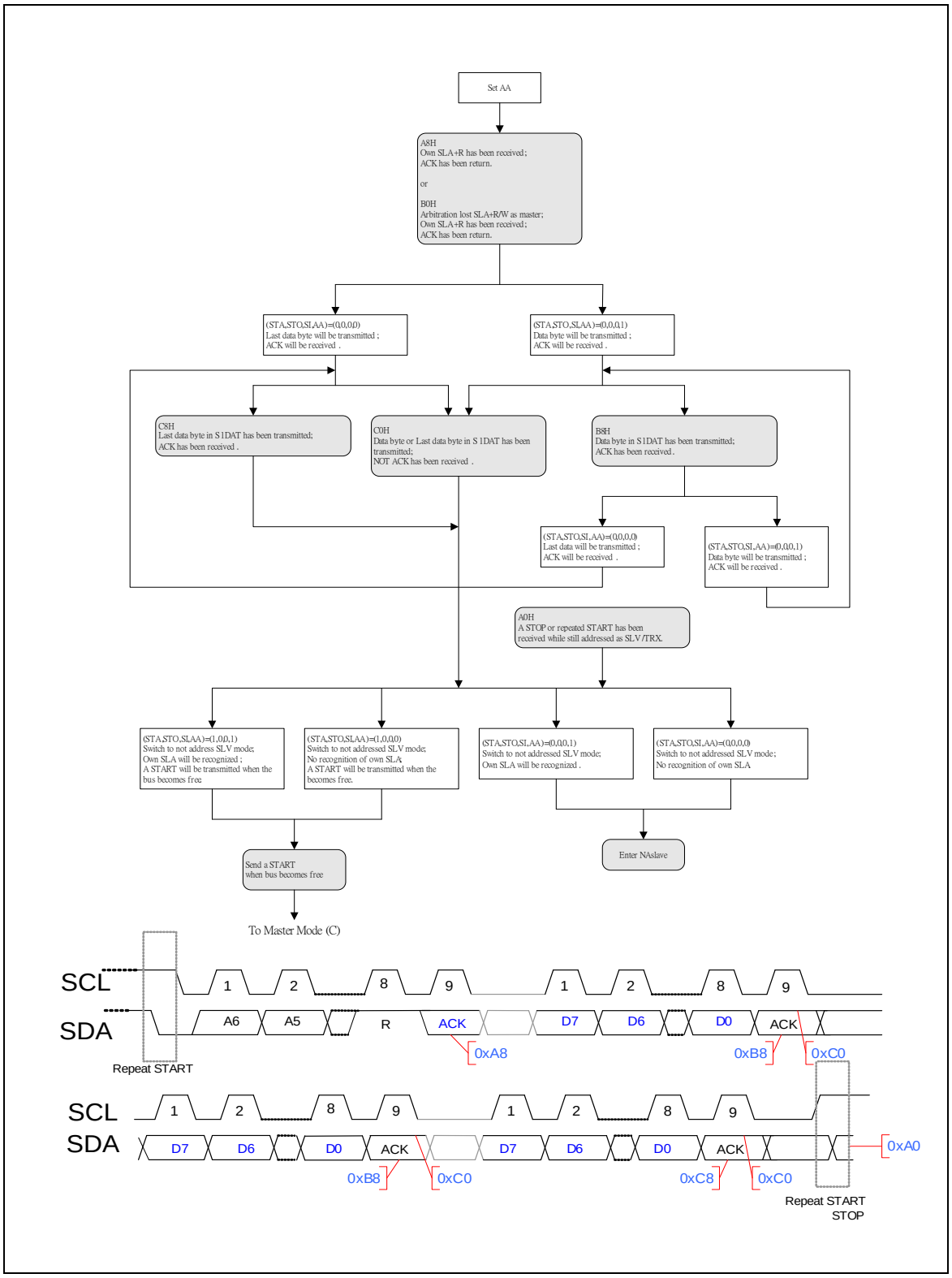



Figure 5-52 I²C Slave Transmitter Mode

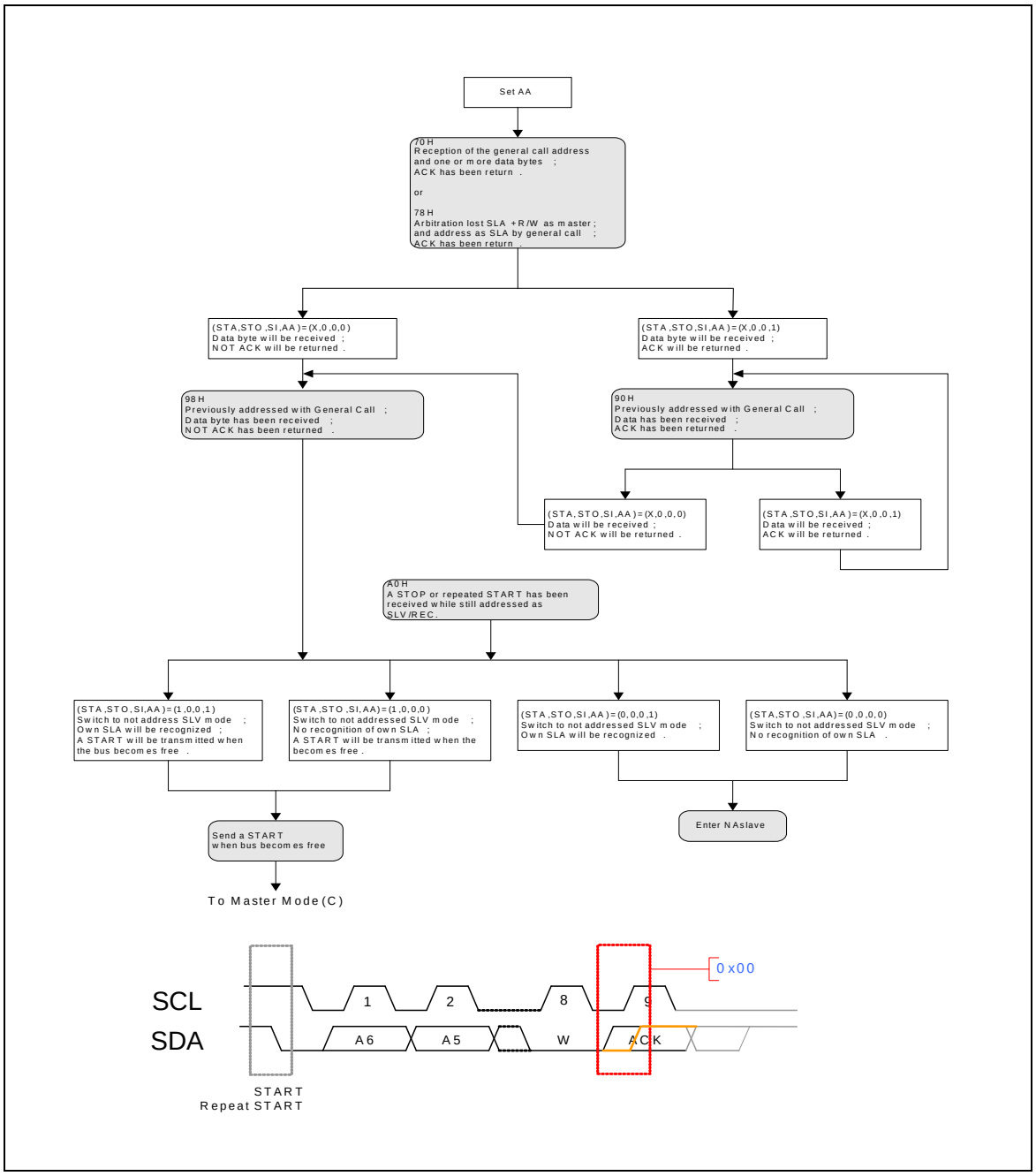



Figure 5-53 I²C General Call Mode

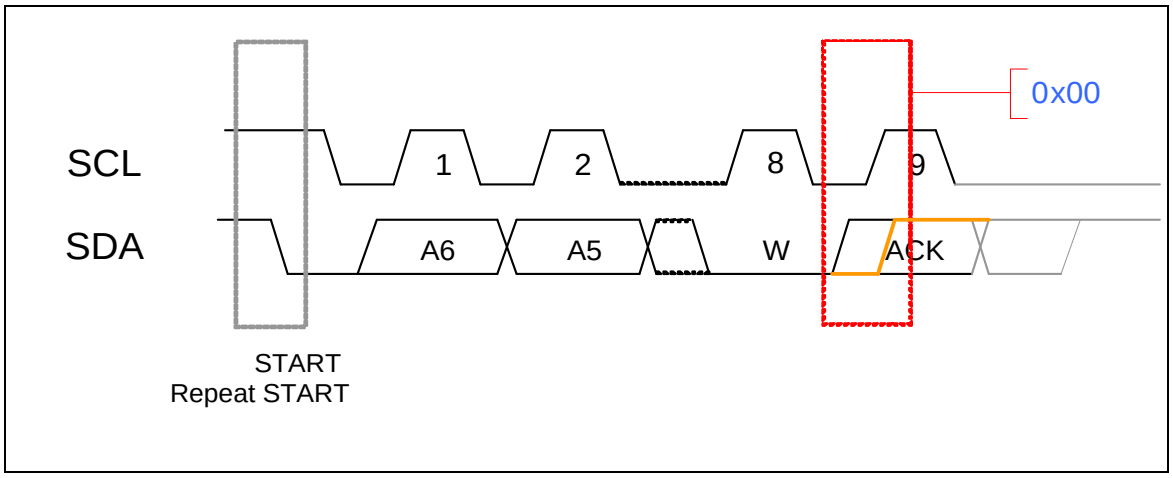



Figure 5-54 I²C Bus Error Timing

## 5.12 I²S

### 5.12.1 Overview

The audio controller consists of I²S protocol to interface with external audio CODEC. Two 8 word deep FIFO for receiving path and transmitting path respectively and is capable of handling 8 ~ 32 bit word sizes. PDMA controller handles the data movement between FIFO and memory.

### 5.12.2 Features

- I²S can operate as either master or Slave mode.
- Capable of handling 8, 16, 24 and 32 bits word sizes.
- Mono and stereo of audio data are supported.
- I²S and MSB justified data format are supported.
- Two FIFO data buffers (each 32 bits) are provided, one is for transmitting and the other is for receiving.
- Generate interrupt when buffer levels cross a programmable boundary.
- Two PDMA channels request, one is for transmitting and the other is for receiving.

### 5.12.3 Block Diagram

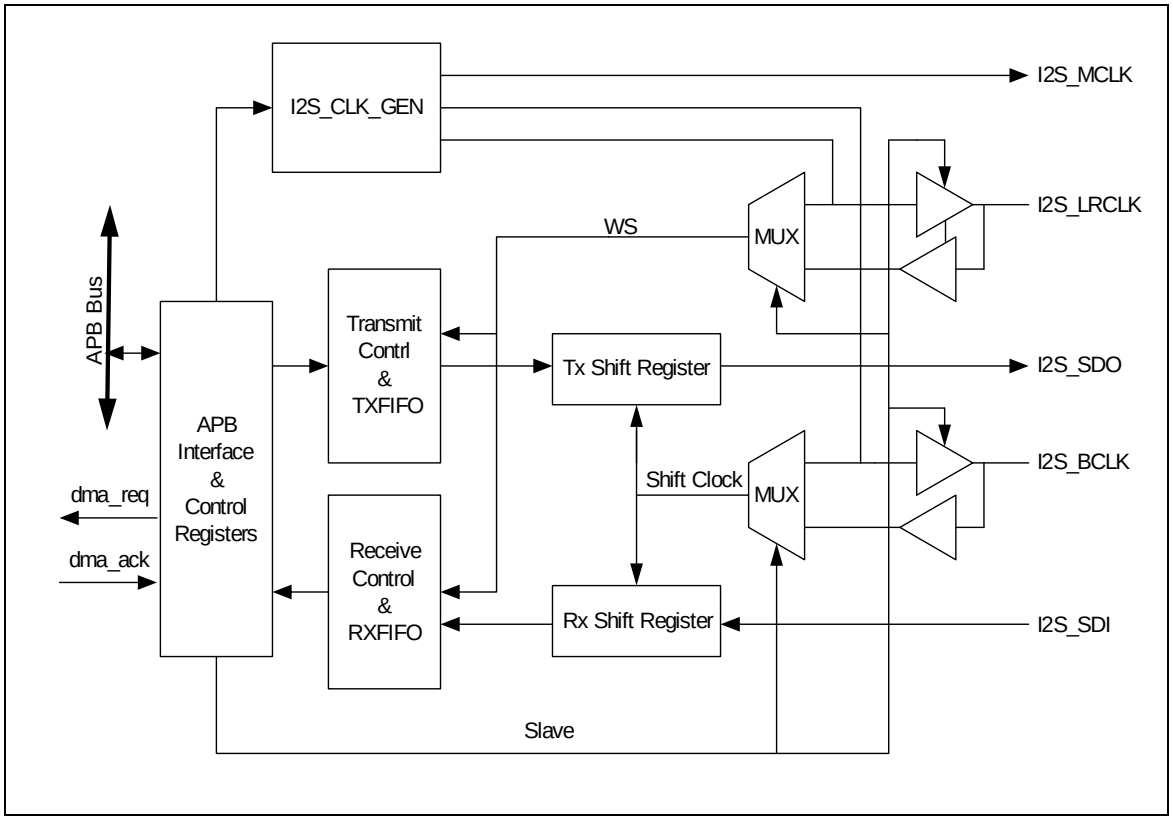



Figure 5-55 I²S Controller Block Diagram

### 5.12.4 Functional Description

5.12.4.1 I2S Operation

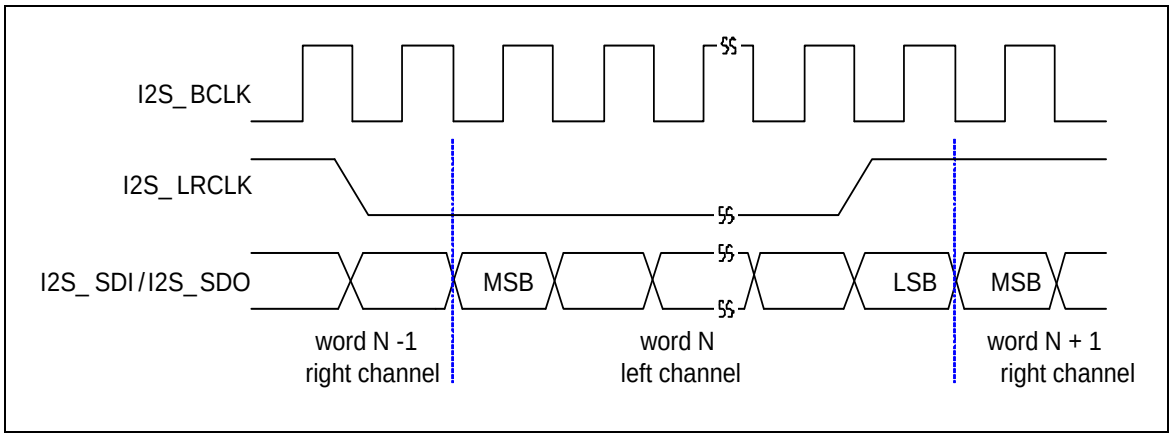



Figure 5-56 I²S bus timing diagram (Format = 0)

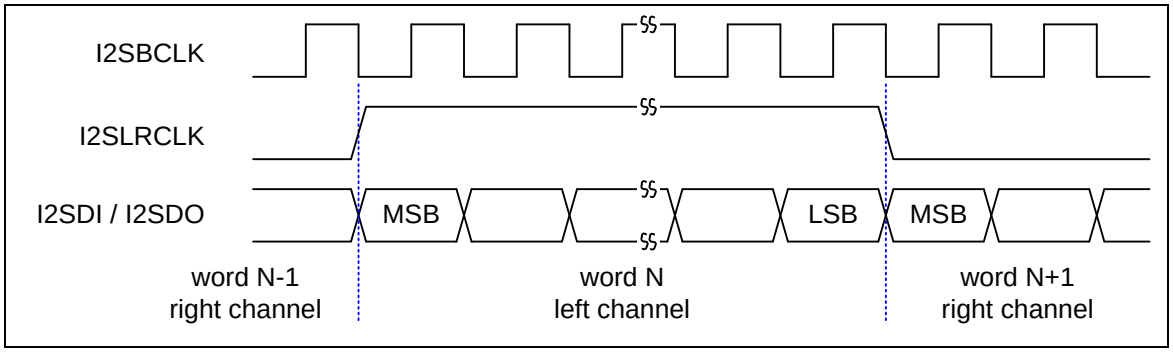



Figure 5-57 MSB Justified Timing Diagram (Format = 1)

5.12.4.2 I2S FIFO

The data structures of FIFO of 8/16/32 bits are described as follows.

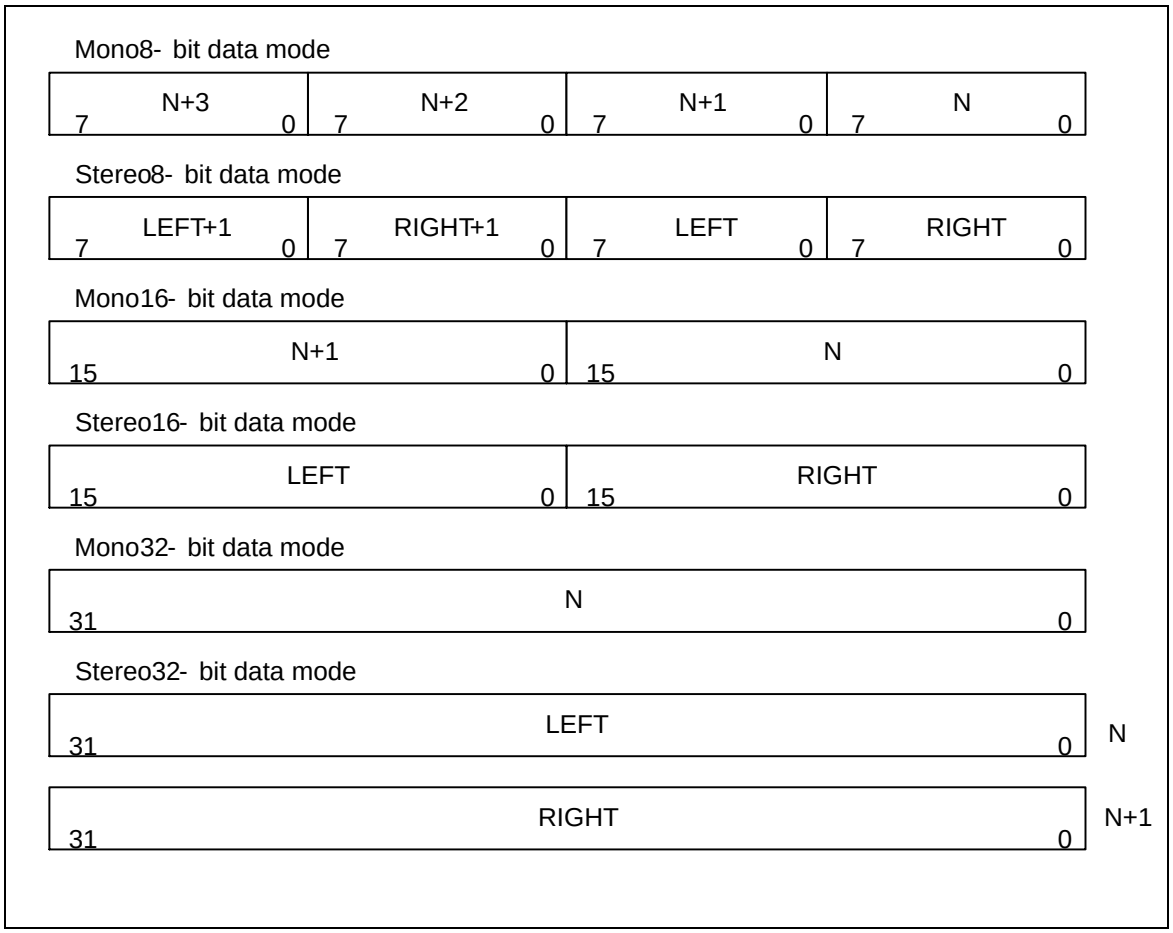



Figure 5-58 FIFO Contents for Various $I^2S$ Modes

## 5.13 LCD Display Driver

### 5.13.1 Overview

The LCD driver can directly drive a LCD glass by creating the ac segment and common voltage signals automatically. It can support static, 1/2 duty, 1/3 duty, 1/4 duty, 1/5 duty and 1/6 duty LCD glass with up to 38 segments with 6 COM (segment 0 is used as LCD_COM4 and segment 1 is used as LCD_COM5) or 40 segments with 4 COM (LCD_COM0 ~ LCD_COM3).

A built-in charge pump function can be enabled to provide the LCD glass with higher voltage than the system voltage. The LCD driver would generate voltage higher than the threshold voltage in older to darken a segment and a voltage lower than threshold to make a segment clear. However, the LCD display segment will degrade if the applied voltage has a DC-component. To avoid this, the generated waveform by LCD driver are arranged such that average voltage of each segment is zero and the RMS(root-mean-square) voltage applied on a LCD segment lower than the segment threshold making LCD clear and RMS voltage higher than the segment threshold making LCD dark.

Note : Output voltage for ADC/LCD shared pins cannot be higher than VDD because these pins are without 5V tolerance.

(LQFP64 : LCD_SEG17, LCD_SEG19, LCD_SEG20, LCD_SEG21, LCD_SEG22, LCD_SEG23)

(LQFP128 : LCD_SEG36, LCD_SEG37, LCD_SEG38, LCD_SEG39)

### 5.13.2 Features

- Supports up to 180 dots (6x30) or 128 dots (4x32) in LQFP64 package and 228 dots (6x38) or 160 dots (4x40) in LQFP100/LQFP128 package Segment/Com pins:
- Common 0-5 multiplexing functions with GPI/O pins
- Segment 0-39 multiplexing function with GPI/O pins
- Supports Static,1/2 bias and 1/3 bias voltage
- Six display modes: Static,1/2 duty, 1/3 duty, 1/4 duty, 1/5 duty or 1/6 duty Selectable LCD frequency by frequency divider
- Configurable frame frequency
- Internal Charge pump, adjustable contrast adjustment
- Embedded LCD bias reference ladder (R-Type, 200kΩ resisters)
- Configurable Charge pump frequency
- Blinking capability
- Supports R/C-type method
- LCD frame interrupt

### 5.13.3 Block Diagram

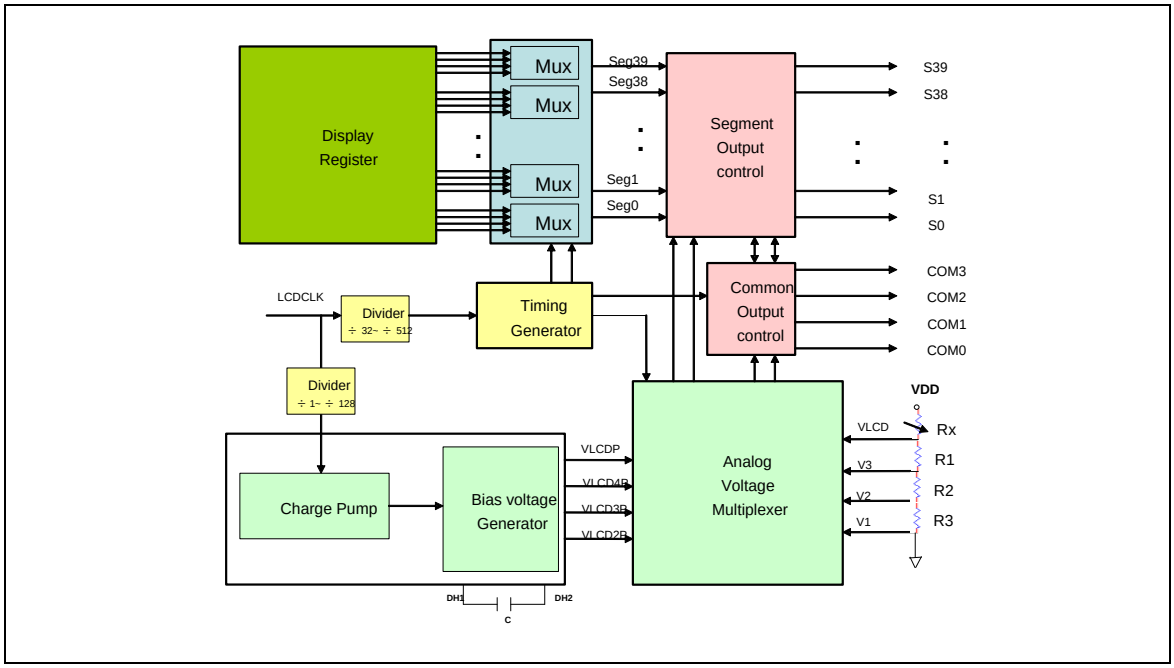



Figure 5-59 LCD Driver Block Diagram

### 5.13.4 Functional Description

The LCD driver consists of display memory register, segment output control, common output, timing generator, charge pump and analog voltage multiplexer blocks. The display memory register stores LCD segment darkened or cleared data. The data bit that is stored in display memory register with “1” makes the LCD segment be darkened and the data bit that is stored in display memory register with “0” makes the LCD segment be cleared. The display memory register is organized with LCD_MEM_0 ~ LCD_MEM_9 registers. Programming the data bits in LCD_MEM_0 ~ LCD_MEM_9 registers can make the corresponding LCD segment be darkened or cleared. The data stored in display memory register is multiplexed to segment output block sequentially with clock generated by timing generator block. The segment output block is in charged of producing SEG 0 ~ SEG 39 driving line and the common output block is in charged of producing COM0 ~ COM3 driving line. Charge pump block provides boosting voltage function for LCD glass. The charge pump input voltage range is from 1.8V to 3.6V. The multi-levels bias voltage can be programmed by CPUMP_VOL_SET bits of LCD_DISPCTL register and the multi-levels bias voltage from 2.6V to 3.3 V can be generated by charge pump block. The analog voltage multiplexer can generates static, 1/2 bias and 1/3 bias voltage output by setting BIAS_SEL bits of LCD_DISPCTL registers. User can program the BIAS_SEL bits to generate different bias voltage for COM and SEG driving line to drive LCD glass. Each common signal is selected sequentially according to the specified number of time slices of its frame period. For example, in 1/3 duty, COM0 to COM2 will output waveforms, COM3 will be tied to low. Whereas for 1/6 duty, COM0 to COM5 will output waveforms. COM signal waveform is shown in Figure 5-61.

#### 5.13.4.1 LCD Display Memory MAP

| DISPLAY Register | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| COM | X | X | COM5 | COM4 | COM3 | COM2 | OCM1 | COM0 | X | X | COM5 | COM4 | COM3 | COM2 | OCM1 | COM0 | X | X | COM5 | COM4 | COM3 | COM2 | OCM1 | COM0 | X | X | COM5 | COM4 | COM3 | COM2 | OCM1 | COM0 |
| LCD_MEM_9 | X | X | SEG39 | SEG39 | SEG39 | SEG39 | SEG39 | SEG39 | X | X | SEG38 | SEG38 | SEG38 | SEG38 | SEG38 | SEG38 | X | X | SEG37 | SEG37 | SEG37 | SEG37 | SEG37 | SEG37 | X | X | SEG36 | SEG36 | SEG36 | SEG36 | SEG36 | SEG36 |
| LCD_MEM_8 | X | X | SEG35 | SEG35 | SEG35 | SEG35 | SEG35 | SEG35 | X | X | SEG34 | SEG34 | SEG34 | SEG34 | SEG34 | SEG34 | X | X | SEG33 | SEG33 | SEG33 | SEG33 | SEG33 | SEG33 | X | X | SEG32 | SEG32 | SEG32 | SEG32 | SEG32 | SEG32 |
| LCD_MEM_7 | X | X | SEG31 | SEG31 | SEG31 | SEG31 | SEG31 | SEG31 | X | X | SEG30 | SEG30 | SEG30 | SEG30 | SEG30 | SEG30 | X | X | SEG29 | SEG29 | SEG29 | SEG29 | SEG29 | SEG29 | X | X | SEG28 | SEG28 | SEG28 | SEG28 | SEG28 | SEG28 |
| LCD_MEM_6 | X | X | SEG27 | SEG27 | SEG27 | SEG27 | SEG27 | SEG27 | X | X | SEG26 | SEG26 | SEG26 | SEG26 | SEG26 | SEG26 | X | X | SEG25 | SEG25 | SEG25 | SEG25 | SEG25 | SEG25 | X | X | SEG24 | SEG24 | SEG24 | SEG24 | SEG24 | SEG24 |
| LCD_MEM_5 | X | X | SEG23 | SEG23 | SEG23 | SEG23 | SEG23 | SEG23 | X | X | SEG22 | SEG22 | SEG22 | SEG22 | SEG22 | SEG22 | X | X | SEG21 | SEG21 | SEG21 | SEG21 | SEG21 | SEG21 | X | X | SEG20 | SEG20 | SEG20 | SEG20 | SEG20 | SEG20 |
| LCD_MEM_4 | X | X | SEG19 | SEG19 | SEG19 | SEG19 | SEG19 | SEG19 | X | X | SEG18 | SEG18 | SEG18 | SEG18 | SEG18 | SEG18 | X | X | SEG17 | SEG17 | SEG17 | SEG17 | SEG17 | SEG17 | X | X | SEG16 | SEG16 | SEG16 | SEG16 | SEG16 | SEG16 |
| LCD_MEM_3 | X | X | SEG15 | SEG15 | SEG15 | SEG15 | SEG15 | SEG15 | X | X | SEG14 | SEG14 | SEG14 | SEG14 | SEG14 | SEG14 | X | X | SEG13 | SEG13 | SEG13 | SEG13 | SEG13 | SEG13 | X | X | SEG12 | SEG12 | SEG12 | SEG12 | SEG12 | SEG12 |
| LCD_MEM_2 | X | X | SEG11 | SEG11 | SEG11 | SEG11 | SEG11 | SEG11 | X | X | SEG10 | SEG10 | SEG10 | SEG10 | SEG10 | SEG10 | X | X | SEG09 | SEG09 | SEG09 | SEG09 | SEG09 | SEG09 | X | X | SEG08 | SEG08 | SEG08 | SEG08 | SEG08 | SEG08 |
| LCD_MEM_1 | X | X | SEG07 | SEG07 | SEG07 | SEG07 | SEG07 | SEG07 | X | X | SEG06 | SEG06 | SEG06 | SEG06 | SEG06 | SEG06 | X | X | SEG05 | SEG05 | SEG05 | SEG05 | SEG05 | SEG05 | X | X | SEG04 | SEG04 | SEG04 | SEG04 | SEG04 | SEG04 |
| LCD_MEM_0 | X | X | SEG03 | SEG03 | SEG03 | SEG03 | SEG03 | SEG03 | X | X | SEG02 | SEG02 | SEG02 | SEG02 | SEG02 | SEG02 | X | X | SEG01 | SEG01 | SEG01 | SEG01 | SEG01 | SEG01 | X | X | SEG00 | SEG00 | SEG00 | SEG00 | SEG00 | SEG00 |

Figure 5-60 LCD Memory Map

#### 5.13.4.2 Frame Counter (FC) and Blinking Display

In 6-mux configuration, COM0, COM1, COM2, COM3, COM4 and COM5 organize one frame. In 5-mux configuration, COM0, COM1, COM2, COM3 and COM4 organize one frame. In 4-mux configuration, COM0, COM1, COM2 and COM3 organize one frame. In 3-mux configuration, COM0, COM1 and COM2 organize one frame. In 2-mux configuration, COM0 and COM1 organize one frame. In static configuration, COM0 organizes one frame. The frame counter can be pre-scaled by programming pre-scale counter (LCD_FCR[PRESCL]). The pre-scale counter can be divided by 1, 2, 4 and 8. The frame counter is counted down from FCV (LCD_FCR[9:4]) to zero. FCV is the top value of frame counter. If FCEN (LCD_FCR[0]) is set to 1 and FCINTEN (LCD_FCR[1]) is set to 1, once frame counter is counted down to zero, the frame counter overflow interrupt is generated. At the same time, the frame counter is reloaded with FCV automatically. The LCD blinking display is controlled with frame counter overflow time. In blinking configuration, the segments are turned on and turned off alternately by frame counter overflow time. The frame counter overflowing interrupt can be also used as synchronization for filling data to LCD display memory register.

#### 5.13.4.3 LCD Display Power Down

If the power-down request is triggered from system manager, LCD controller will execute the frame completely to avoid the DC component. When the frame is executed completely, the LCD power down

interrupt signal is generated to inform system manager the LCD controller is ready to enter power down state, if PDINT_EN (LCD_CTL[9]) is enabled.Otherwise, if PDINT_EN (LCD_CTL[9]) is disabled, the LCD power down interrupt signal is blocked and the interrupt is disabled. If the PDDSIP_EN (LCD_CTL[8]) is set to 1, the LCD display is operated in Power-down mode. Otherwise, if PDDSIP_EN (LCD_CTL[8]) is cleared to 0, the LCD display is off in Power-down mode.

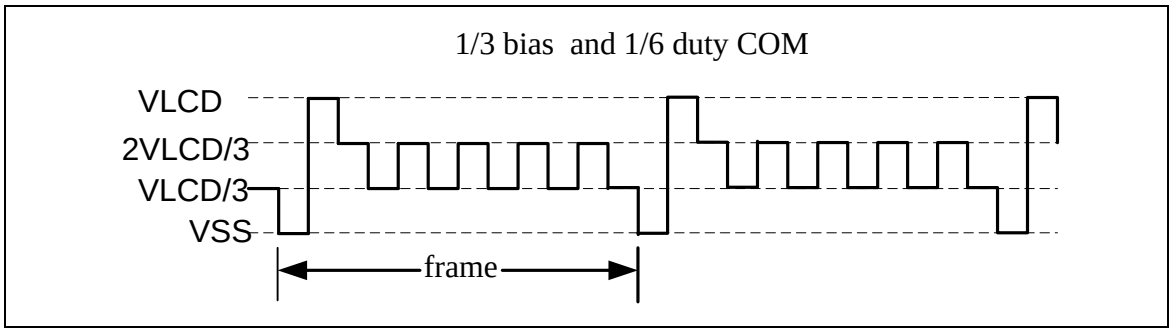



Figure 5-61 COM Signal Waveform

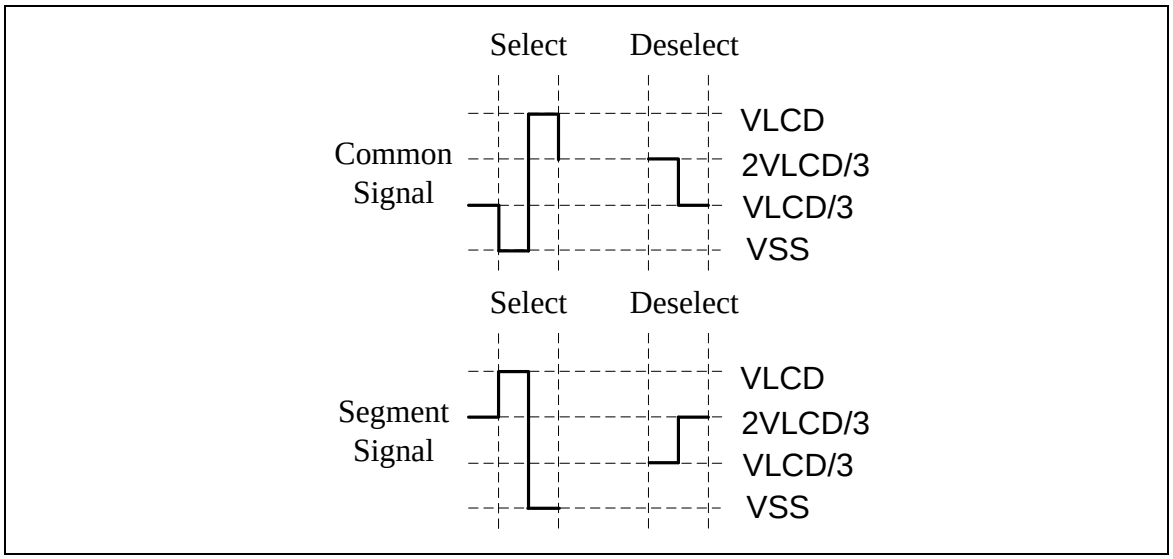



Figure 5-62 SEG Signal Waveform

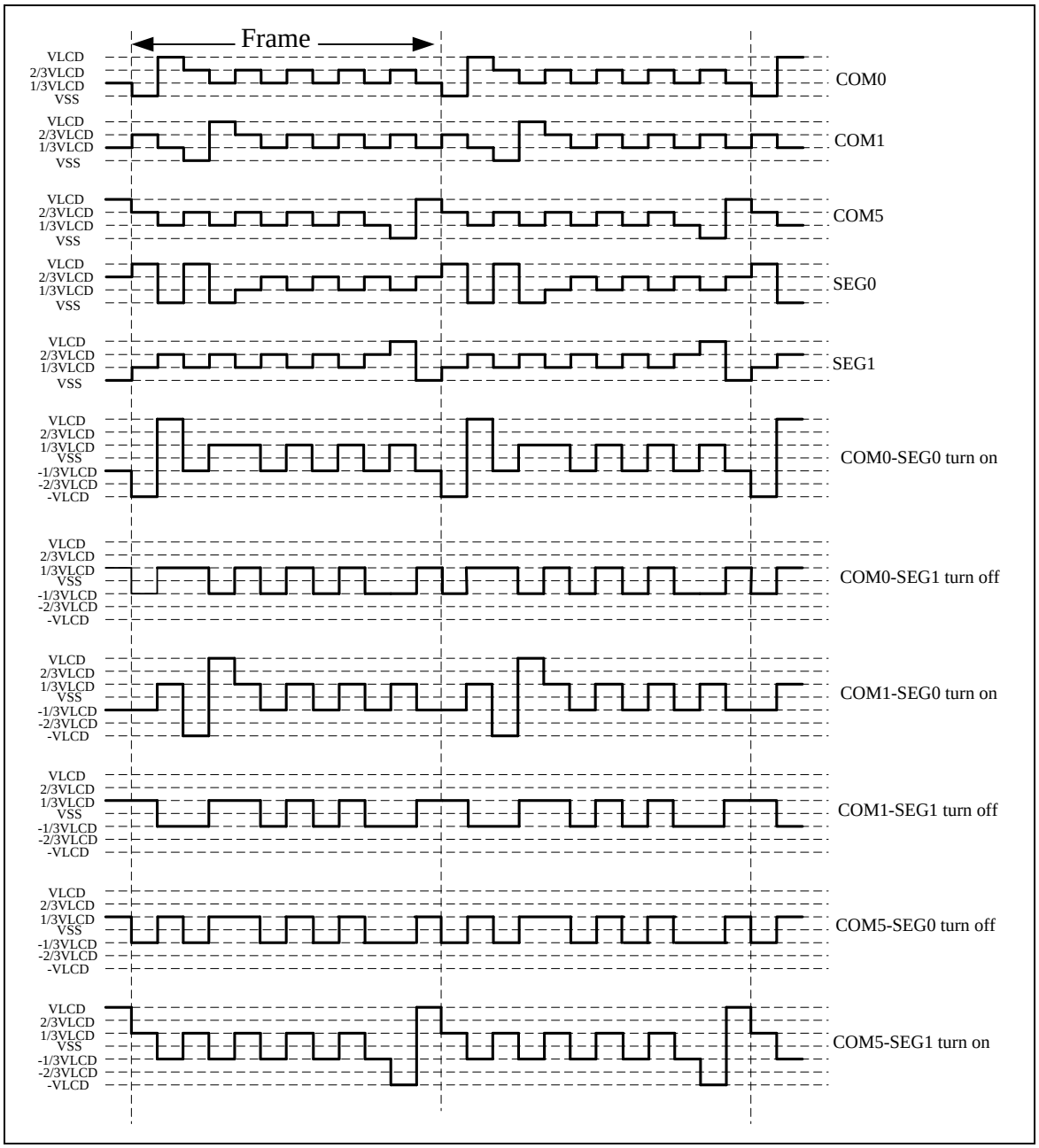



Figure 5-63 COM-SEG Signal Waveform by 1/6 Duty with 1/3 Bias

### 5.13.5 Application Circuit

**External Resister ladder**

1. Most commonly used for high VDD voltages.
2. Uses inexpensive resistors to create the multilevel LCD voltages. Regardless of the number of pixels that are energized the current remains constant. The voltage at point VLCD is typically tied to VDD, either internally or externally

3 The resister values are determined by two factors.

a. Display quality

b. Power consumption

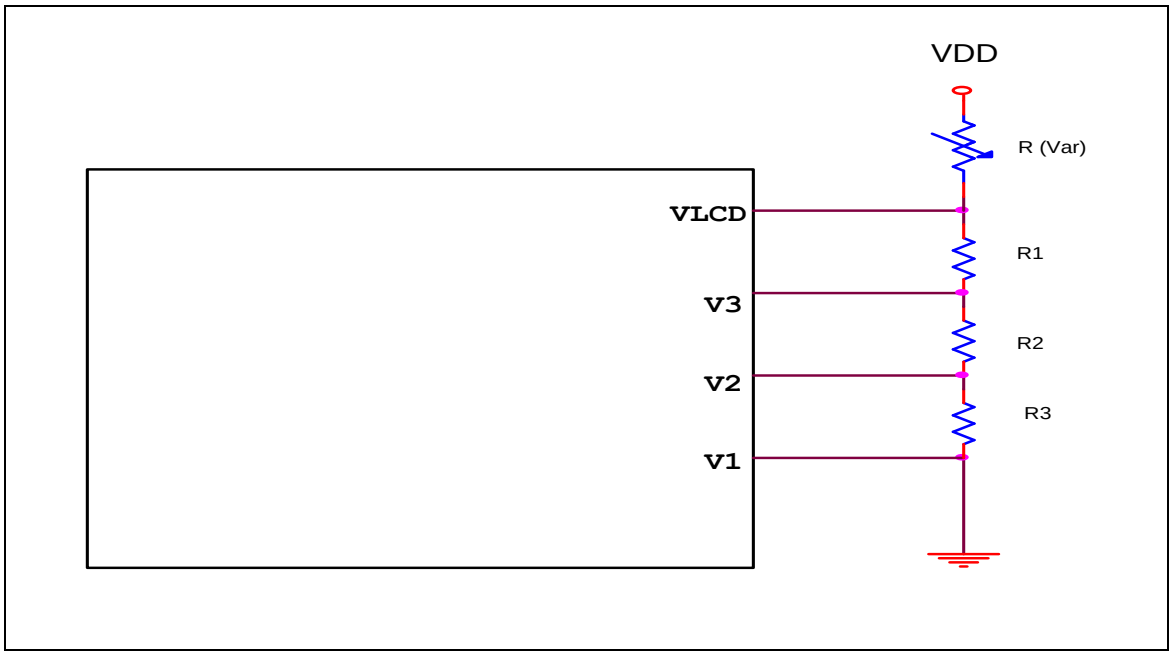



Figure 5-64 1/3 Bias (External Resistor Ladder)

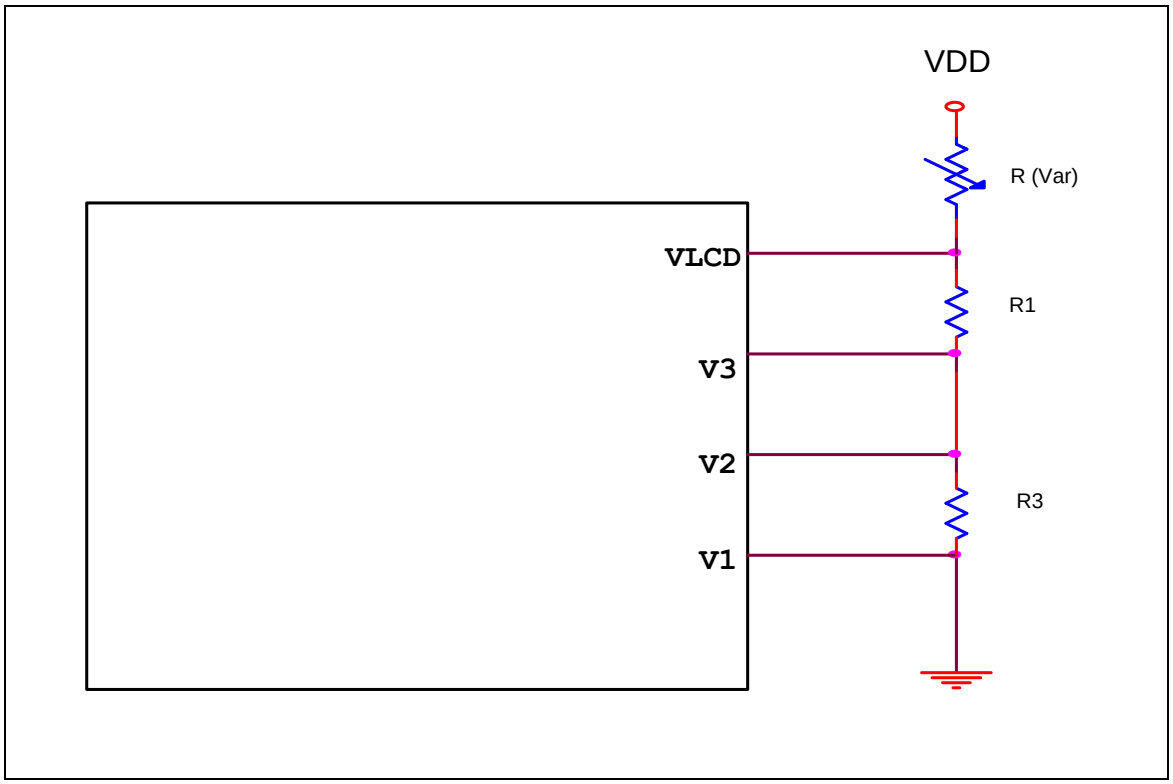



Figure 5-65 1/2 Bias (External Resistor Ladder)

**Resistor ladder with capacitors**

Sometimes the addition of parallel capacitors to the resistance can reduce the distortion caused by charging/discharging currents. This effect is limited since at some point a large resistor and large capacitor cause a voltage level shift which negatively impacts the display quality.

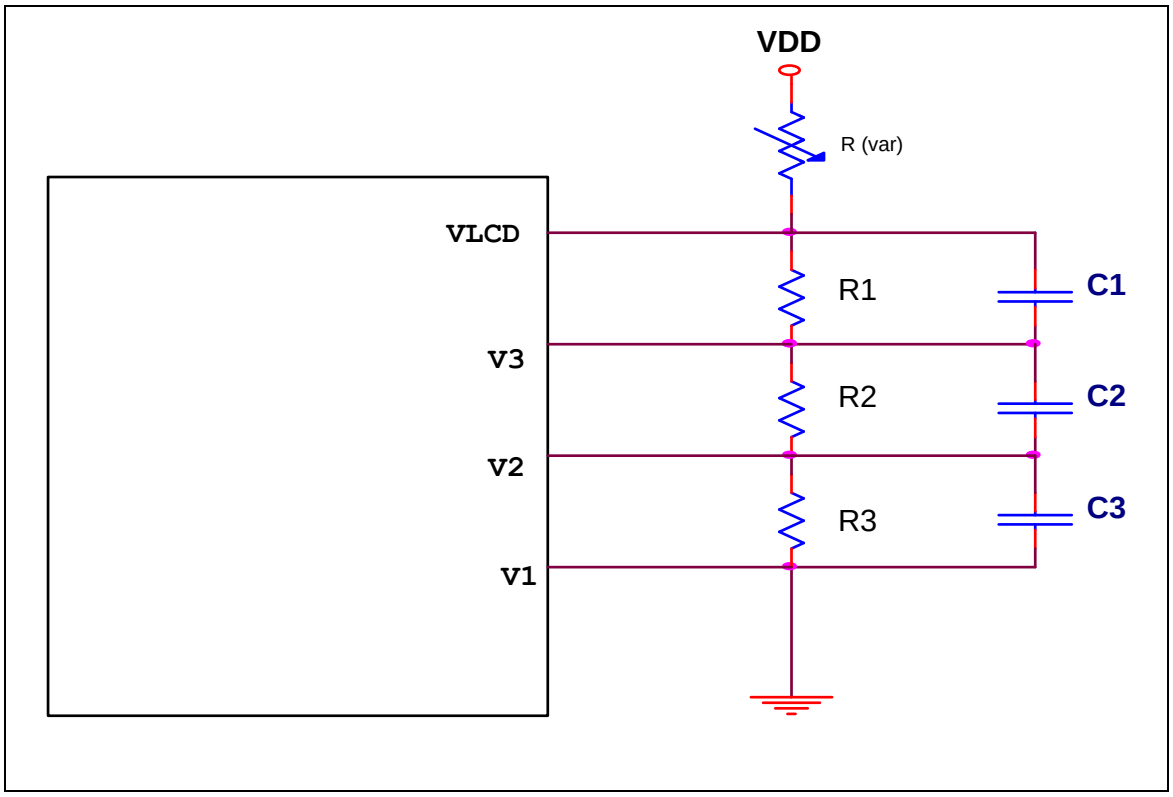



Figure 5-66 1/3 Bias (Resistor Ladder with Capacitor)

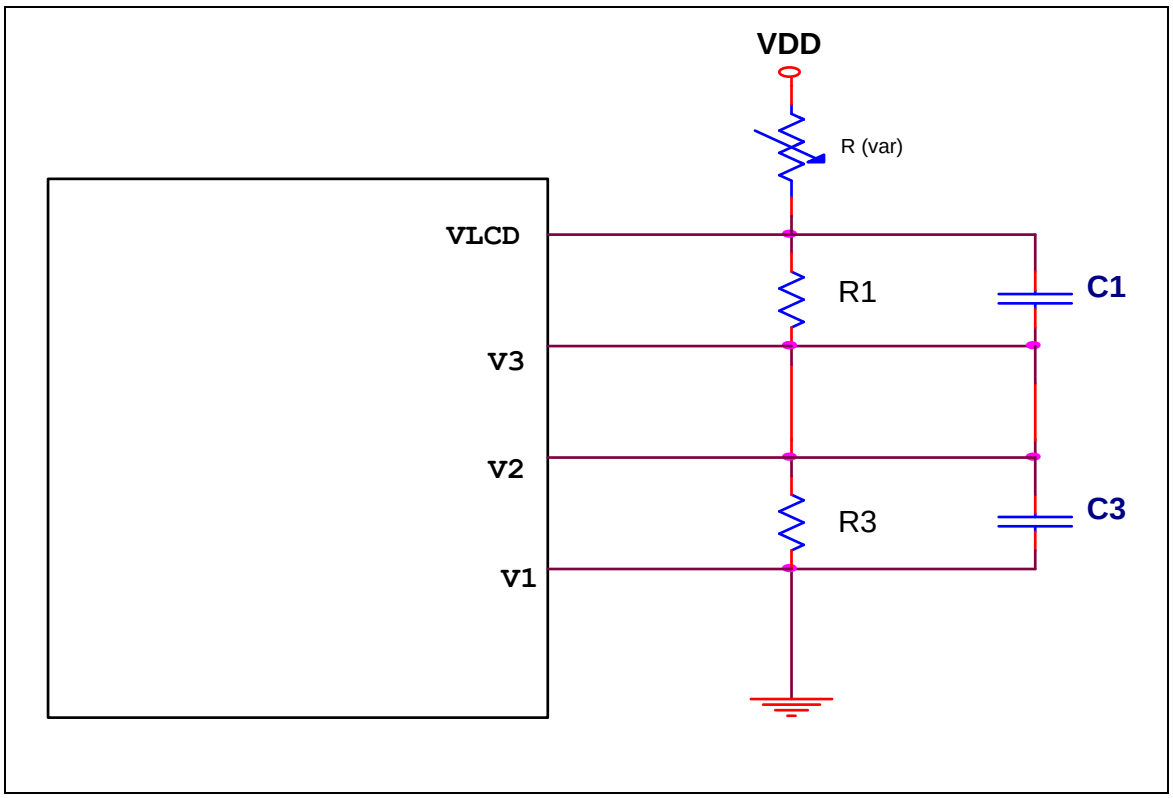



Figure 5-67 1/2 Bias (Resistor Ladder with Capacitor)

**Charge Pump**

1. Ideal for low voltage battery operation because the VDD voltage can be boosted up to drive the LCD panel.

2. The charge pump requires a charging capacitor and filter capacitor for each of the LCD voltages.

3. These capacitors are typically polyester, polypropylene, or polystyrene material.

4. Another feature that makes the charge pump ideal for battery applications is that the current consumption is proportional to the number of pixels that are energized.

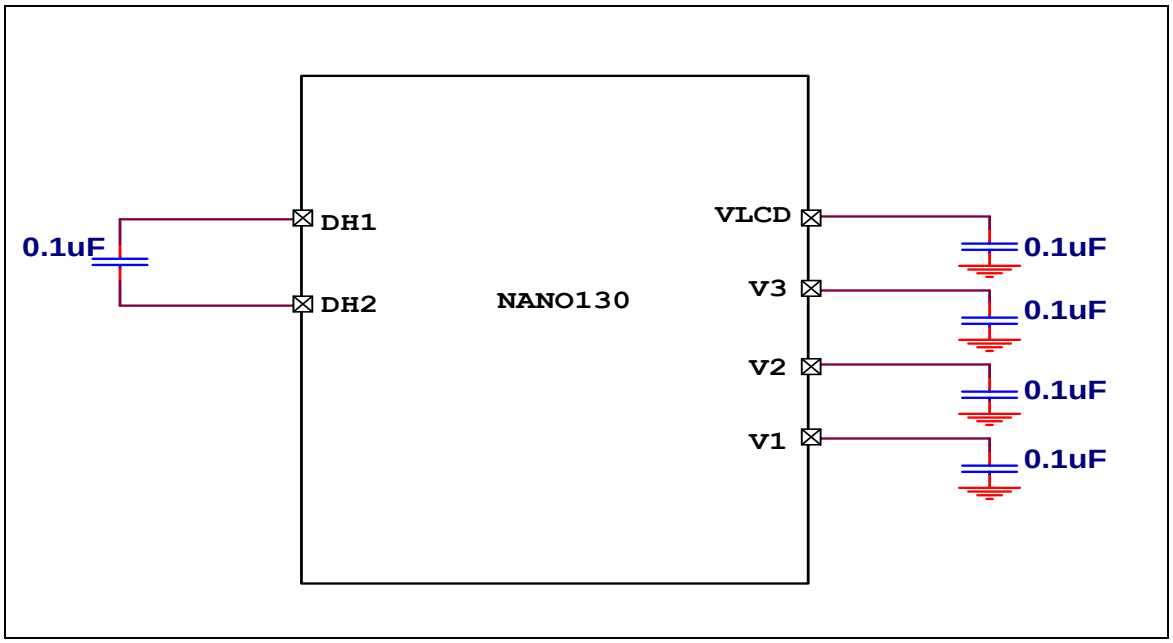



Figure 5-68 1/3 Bias (Charge Pump)

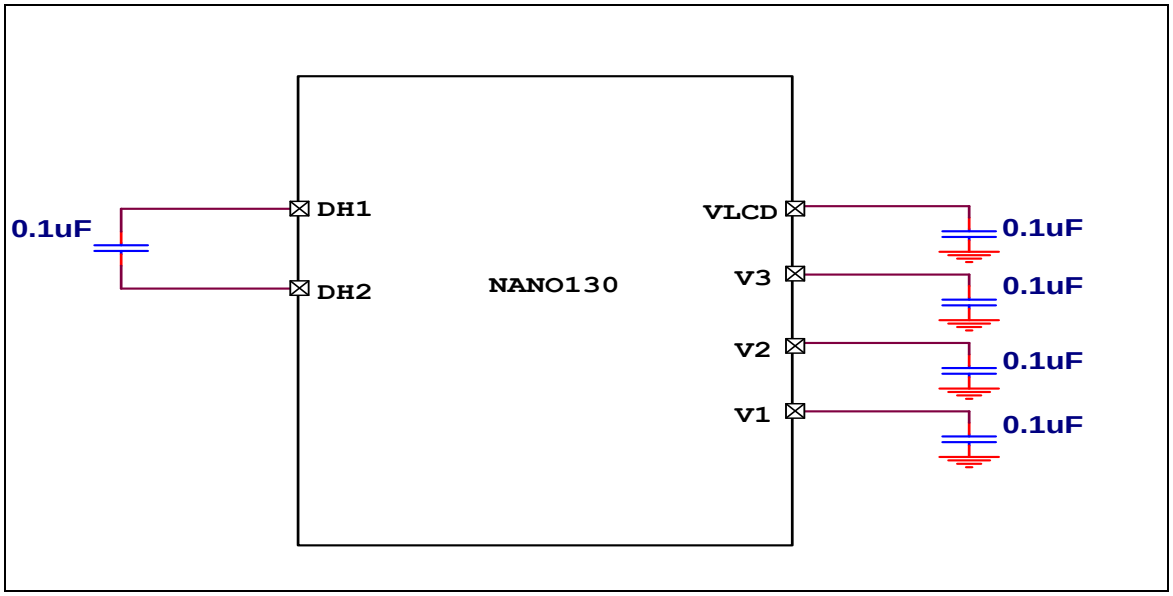



Figure 5-69 1/2 Bias (Charge Pump)

## 5.14 Pulse Width Modulation (PWM)

### 5.14.1 Overview

This chip has two PWM controllers, each controller has 4 independent PWM outputs, CH0~CH3, or as 2 complementary PWM pairs, (CH0, CH1), (CH2, CH3) with 2 programmable dead-zone generators.

Each two PWM outputs, (CH0, CH1), (CH2, CH3), share the same 8-bit prescaler, clock divider providing 5 divided frequencies (1, 1/2, 1/4, 1/8, 1/16). Each PWM output has independent 16-bit PWM down-count counter for PWM period control, and 16-bit comparators for PWM duty control. Each dead-zone generator has two outputs. The first dead-zone generator output is CH0 and CH1, and for the second dead-zone generator, the output is CH2 and CH3. The 2 sets of PWM controller total provide eight independent PWM interrupt flags which are set by hardware when the corresponding PWM period down counter reaches zero. PWM interrupt will be asserted when both PWM interrupt source and its corresponding enable bit are active. Each PWM output can be configured as one-shot mode to produce only one PWM cycle signal or continuous mode to output PWM waveform continuously.

When DZEN01 of PWMx_CTL is set, CH0 and CH1 perform complementary PWM paired function; the paired PWM timing, period, duty and dead-time are determined by PWM channel 0 timer and Dead-zone generator 0. Similarly, When DZEN23 of PWMx_CTL is set the complementary PWM pair of (CH2, CH3) is controlled by PWM channel 2.

To prevent PWM driving output pin with unsteady waveform, the 16-bit period down counter and 16-bit comparator are implemented with double buffer. When user writes data to counter/comparator buffer registers the updated value will be loaded into the 16-bit down counter/ comparator at the time down counter reaching zero. The double buffering feature avoids glitch at PWM outputs.

When the 16-bit period down counter reaches zero, the interrupt request is generated. If PWM output is set as continuous mode, when the down counter reaches zero, it is reloaded with CN of PWMx_DUTYy(y=0~3) Register automatically then start decreases, repeatedly. If the PWM output is set as one-shot mode, the down counter will stop and generate one interrupt request when it reaches zero.

The value of PWM counter comparator is used for pulse width modulation. The counter control logic changes the output level when down-counter value matches the value of compare register.

The alternate feature of the PWM is digital input capture function. If capture function is enabled the PWM output pin is switched as capture input pin. The capture channel 0 and PWM CH0 share one timer; and the capture channel 1 and PWM CH1 share one timer, and etc. Therefore user must setup the PWM timer before enabling capture feature. After capture feature is enabled, the capture always latches PWM timer to Capture Rising Latch Register (PWMx_CRLy) where y=0~3, when input channel has a rising transition and latches PWM timer to Capture Falling Latch Register (PWMx_CFLy) where y=0~3, when input channel has a falling transition. Capture channel 0 interrupt is programmable by setting PWMx_CAPINTEN. Whenever Capture event latched for channel 0/1/2/3, the PWM timer 0/1/2/3 will be reload at this moment if the corresponding reload enable bit specified in CAPCTL are set.

The maximum captured frequency that PWM can capture is dominated by the capture interrupt latency. When capture interrupt occurs, software will do at least three steps, they are: Read PWMINTSTS to tell it from interrupt source and Read PWMx_CRLy/PWMx_CFLy(y=0~3) to get capture value and finally write 1 to clear PWMx_INTSTS. If interrupt latency will take time T0 to finish, the capture signal mustn't transient during this interval. In this case, the maximum capture frequency will be 1/T0.

### 5.14.2 Features

5.14.2.1 PWM Function:

- Two PWM controllers, each controller having 4 independent PWM outputs, CH0~CH3, or as 2 complementary PWM pairs, (CH0, CH1), (CH2, CH3) with 2 programmable dead-zone generators
- Up to 8 PWM channels or 4 PWM paired channels
- Up to 16 bits PWM counter width
- PWM Interrupt request synchronous with PWM period
- Single-shot or Continuous mode
- Four Dead-Zone generators

5.14.2.2 Capture Function:

- Timing control logic shared with PWM timer.
- 8 Capture input channels shared with 8 PWM output channels.
- Each channel supports one rising latch register (PWMx_CRLy), one falling latch register (PWMx_CFLy) and Capture interrupt flag (CAPIFy) where x=0~1,y=0~3.
- Eight 16-bit counters for eight capture channels or four 32-bit counter for four capture channels when cascade is enabled: when CH01CASKEN is set, the original 16-bit counter of channel 1 will combine with channel 0's 16 bit counter for channel 0 input capture counting and so does CH23CASKEN for channel 2, 3
- Supports PDMA transfer function for PWMx channel 0, 2

### 5.14.3 Block Diagram

The following figures illustrate the architecture of PWM in groups. (Timer 0&1 are in one group and timer 2&3 are in another, and so on.)

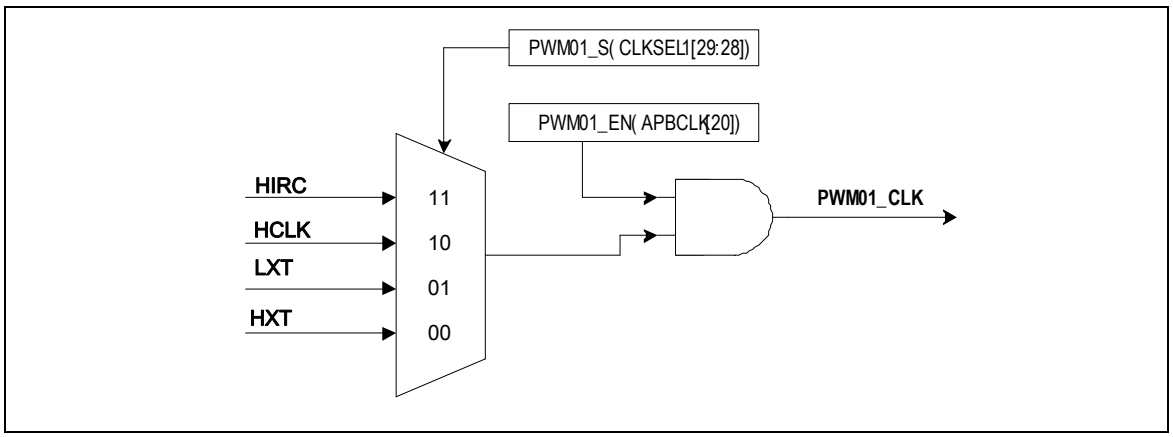



Figure 5-70 PWM0 Clock

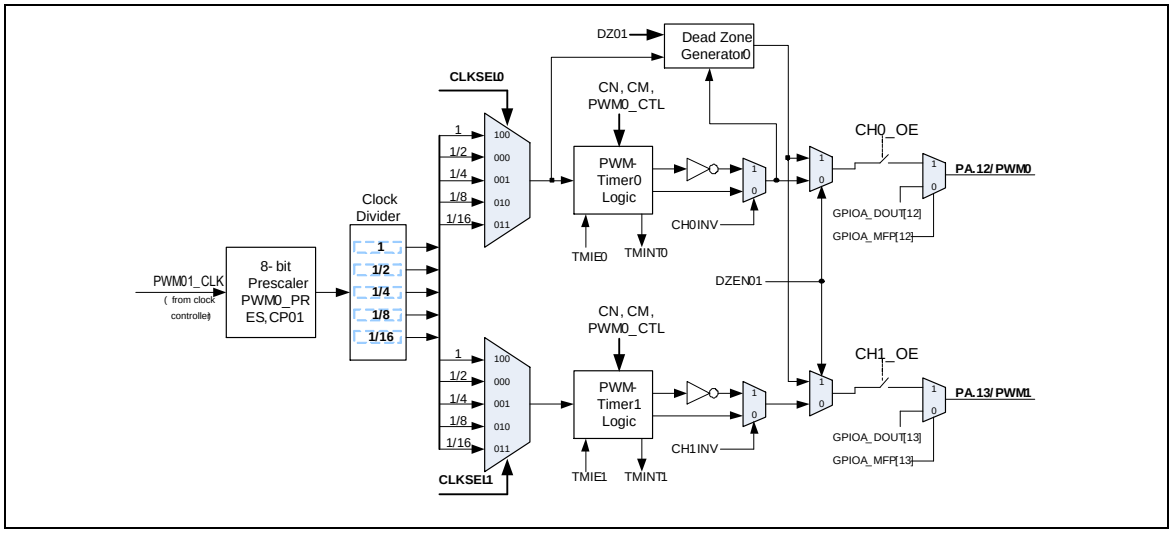



Figure 5-71 PWM0 Generator for Channel 0, 1

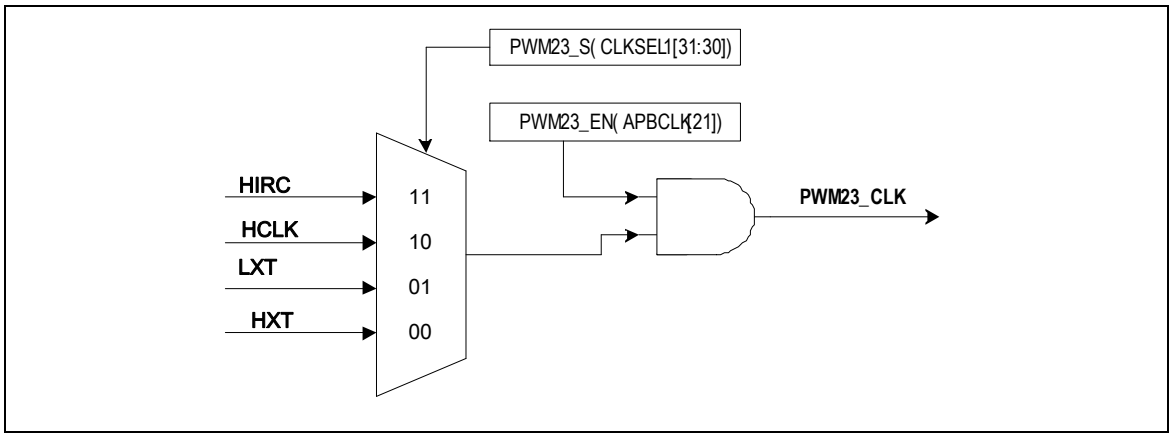



Figure 5-72 PWM1 Clock

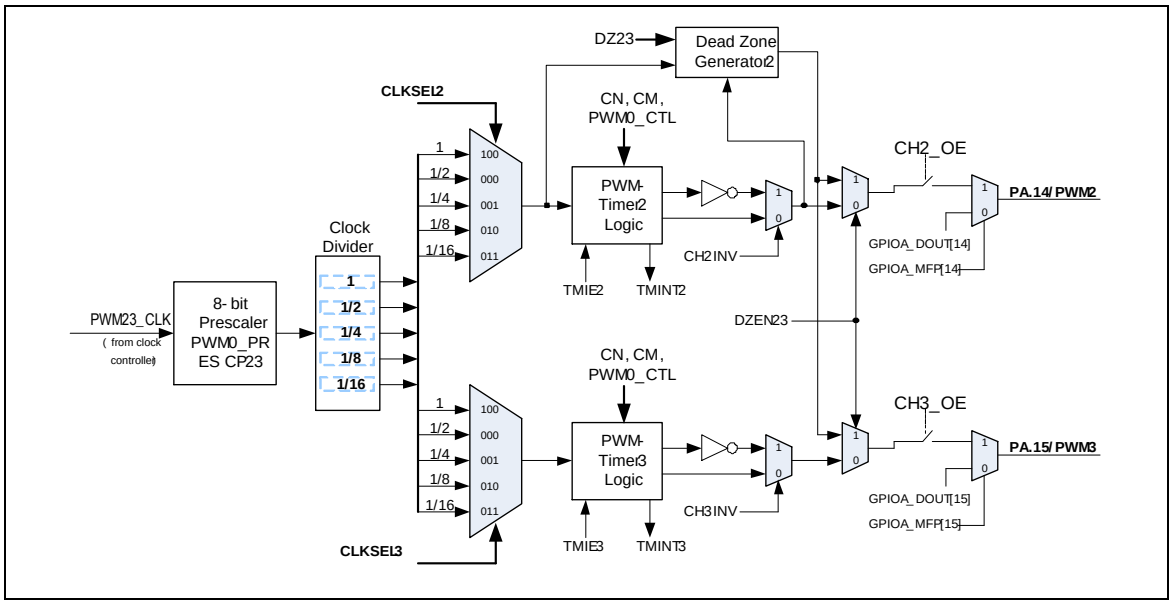



Figure 5-73 PWM Generator for Channel 2, 3

### 5.14.4 Functional Description

#### 5.14.4.1 PWM-Timer Operation

The PWM period and duty control are decided by PWMx_DUTYy (y =0~3) register CN (PWMx_DUTYy[15:0] ) and CM (PWMx_DUTYy[31:16] ). The PWM-timer timing operation is shown in Figure 5-75. The pulse width modulation follows the formula below and the legend of PWM-Timer Comparator is shown in Figure 5-74. Note that the corresponding I/O pins must be configured as output type before PWM function is enabled.

PWM frequency = PWMxy_CLK/(prescale+1)*(clock divider)/(CN+1); where xy, could be 01, 23, depending on selected PWM channel.

Duty ratio = (CM+1)/(CN+1).

CM >= CN: PWM output is always high.

CM < CN: PWM low width= (CN-CM) unit1; PWM high width = (CM+1) unit.

If CM = 0: PWM low width = (CN) unit; PWM high width = 1 unit

**Note:** 1. Unit = one PWM clock cycle.

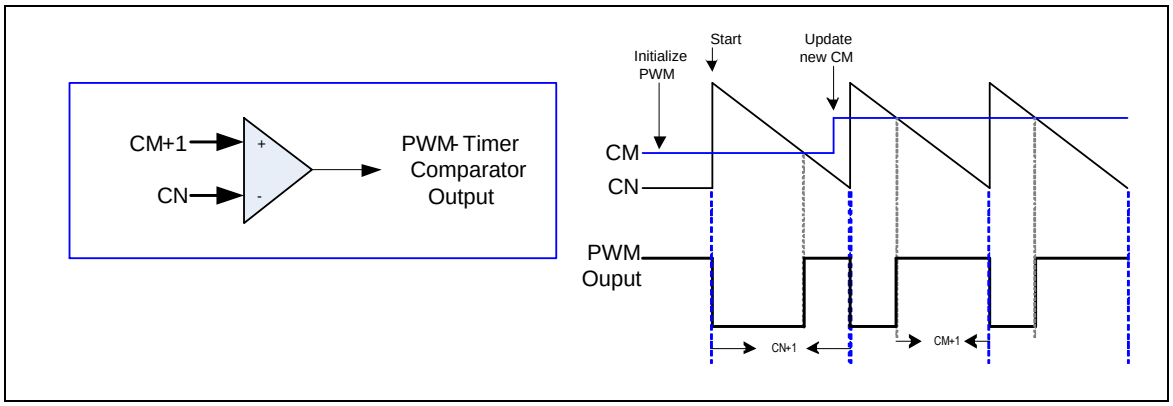



Figure 5-74 Legend of Internal Comparator Output of PWM-Timer

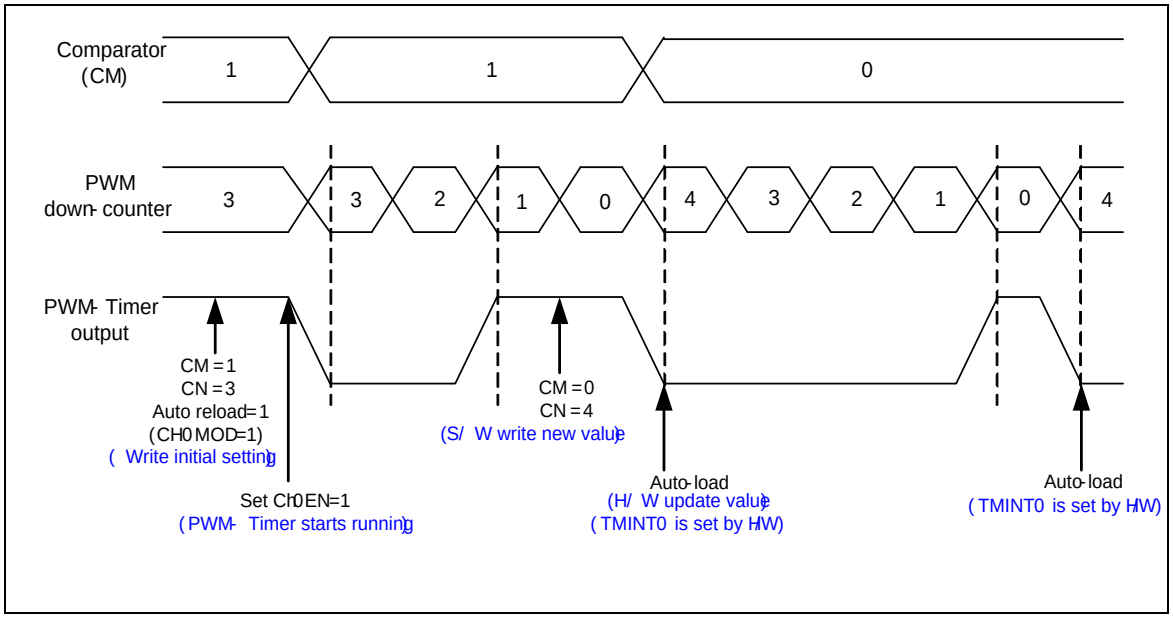



Figure 5-75 PWM-Timer Operation Timing for Channel 0

5.14.4.2 PWM Double Buffering, continuous and one-shot Operation

The PWM has double buffering function; the reload value is updated at the start of next period without affecting current timer operation. The PWM counter value can be written into bit [15:0] of PWMx_DUTY0~3.

The bit CH0MOD in PWM Control Register (PWMx_CTL) defines PWM operation in Continuous or One-shot mode If CH0MOD is set to one (continuous mode), the controller loads CN to PWM counter when PWM counter reaches zero. If CN is set to zero, PWM counter will be halt when PWM counter counts to zero.

In one-shot mode (CH0MOD=0), the corresponding channel will output only one duty waveform and counter will be stopped if no further corresponding duty register updated. When PWM counter is running, updating corresponding duty register will engage the next duty waveform.

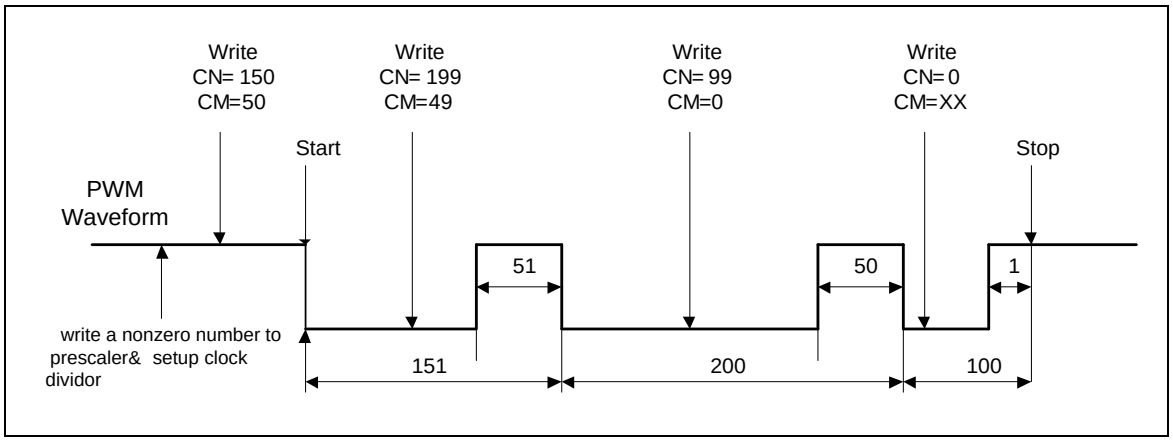



Figure 5-76 PWM Double Buffer Illustration

5.14.4.3 Modulate Duty Ratio

The double buffering function allows CM to be written at any point in current cycle. The loaded value will take effect from next cycle.

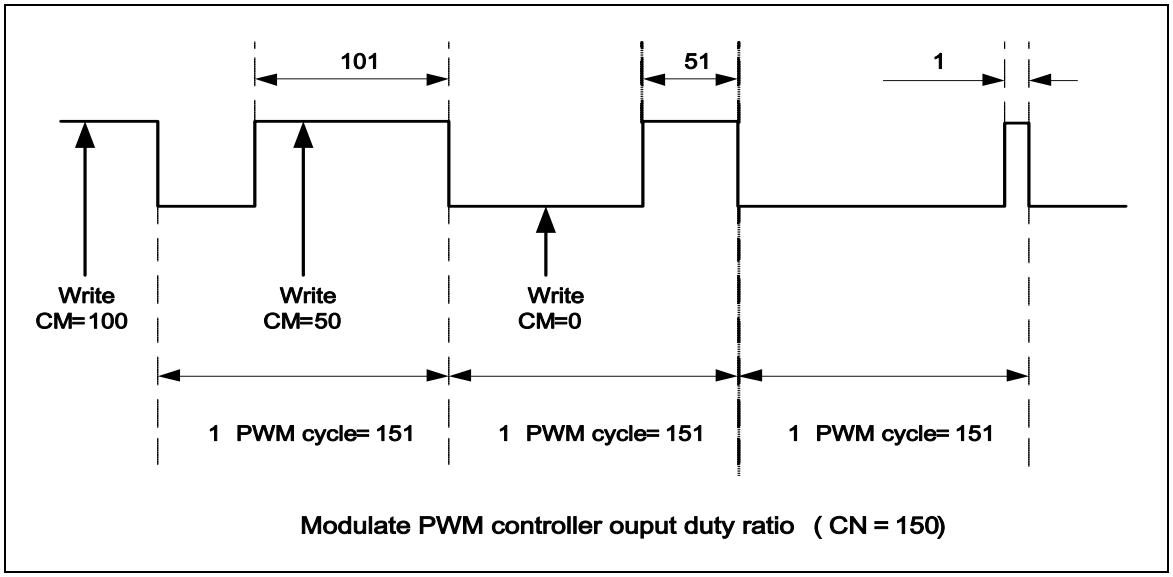



Figure 5-77 PWM Controller Output Duty Ratio

#### 5.14.4.4 Dead-Zone Generator

PWM implements Dead Zone generator. They are built for power device protection. This function generates a programmable time gap to delay PWM rising output. User can program Dead-Zone counter to determine the Dead Zone interval.

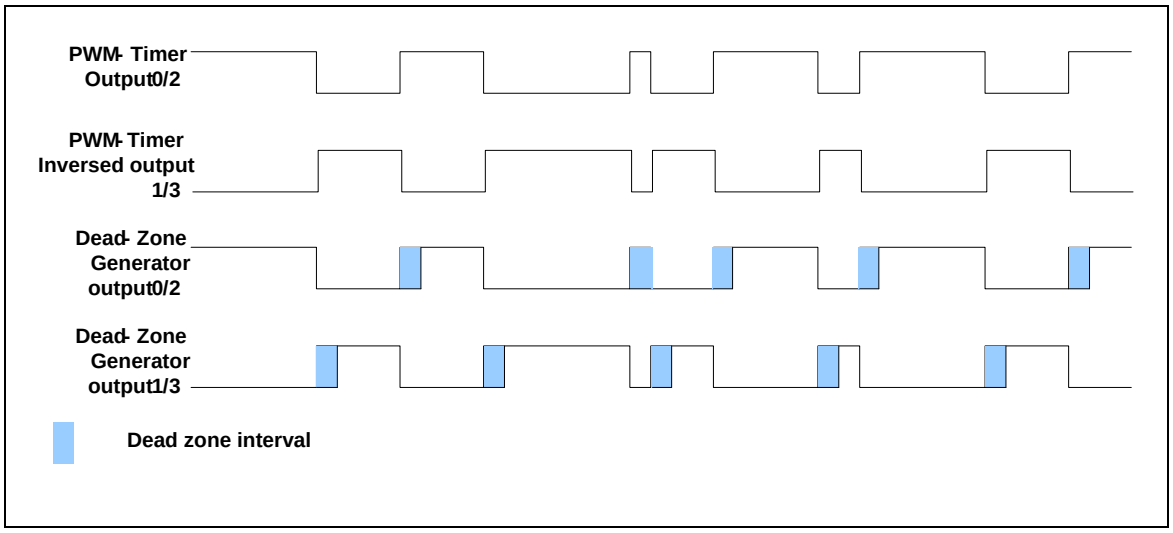



Figure 5-78 Paired PWM Output with Dead Zone Generation Operation

#### 5.14.4.5 Capture Operation

The Capture channel 0 and PWM channel 0 share one timer ; and the Capture channel1 and PWM channel 1 share another timer, and etc. The capture always latches PWM-timer to PWMx_CRL0 when input channel has a rising transition and latches PWM-timer to PWMx_CFL0 when input channel has a falling transition. Capture channel 0 interrupt is programmable by setting PWMx_CAPINTEN[0] (Rising latch Interrupt enable) and PWM_CAPINTEN[1] (Falling latch Interrupt enable) to decide the condition of interrupt occur. Whenever the Capture module issues a capturing flag (rising latched or falling latched) that are defined in CAPUNTSTS, and the reload enable bit(defined in CAPCTL) is also set the corresponding PWM timer will be reloaded with CN at this moment. Note that the corresponding I/O pins must be configured as input type before Capture function is enabled

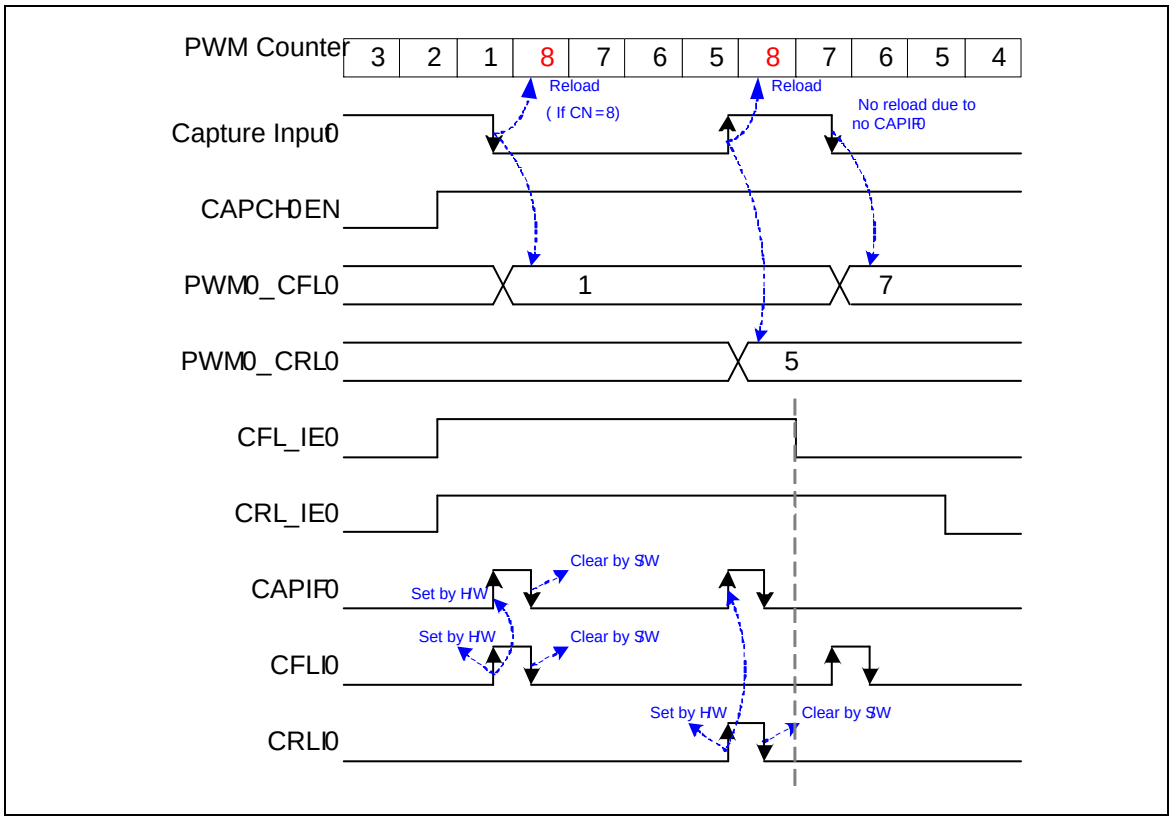



Figure 5-79 PWM Capture Operation Timing

At this example, the CN is 8:

The PWM timer will be reloaded with CN when a capture interrupt flag (CAPIF0) is set.

The channel low pulse widt– is (CN – CRL).

The channel high pulse widt– is (CN – CFL).

In some case that need wider counter, user can cascade two 16 bit counters to 32 bit counter for capturing

The cascade method is depicted below

When enabling CH01CASK in bit[13] of PWMx_CAPCTL, the internal cascade logic will combine CH0's 16 bit counter with CH1's 16 bit counter to become 32 bit counter for CH0 and the same for enabling CH23CASK in bit[29] of PWMx_CAPCTL for CH2 and CH3.CNR0 /CNR2 and CNR1/CNR3 are also cascaded. At this case, the capturing function for CH1 and CH3 are useless.

When capturing flag is setup (rising for CAPINTSTS[1],and falling for CAPINTSTS[2]), the capture data for rising latched is stored in CRLR0 and CRLR1 and CFLR1 and CFLR0 for falling latched. CRLR1 is located in the upper half word and CRLR0 is in lower half word and the same for CFLR1 and CFLR0.

User can also read CRLR0 or CFLR0 register directly to get 32 bit capturing data when cascade is enabled

**Note:** Cascade function is only for PWM capture function.

5.14.4.6 PWM PDMA Function

PWM support PDMA transfer function when operating in capture mode and is only for specified

channel (channel 0,2),when the corresponding PDMA enable bit(defined in CAPCTL register) is set, capture module will issue a request to PDMA controller when the preceding capture event happened. PDMA controller will issue ACK to capture module and read back PDMACH0 register to memory. By setting PDMACAPMOD0 and PDMACAPMOD2, PDMA can transfer rising latched data or falling latched data or both of them to memory. When using PDMA to transfer both falling and rising data, remember to set CHxRFORDER in PWMx_CAPCTL to decide the order of transferring data ( falling edge latched is first or rising edged latched first)

#### 5.14.4.7 PWM-Timer Interrupt Architecture

There are eight PWM, interrupts, PWM0CH0_INT ~ PWMCH3_INT, PWM1CH0_INT ~ PWM1CH3_INT which are OR into PWM0_INT and PWM1_INT. PWM CH0 and Capture channel 0 share one interrupt, PWM CH1 and Capture channel 1 share the same interrupt and so on. Therefore, PWM function and Capture function in the same channel cannot be used at the same time.

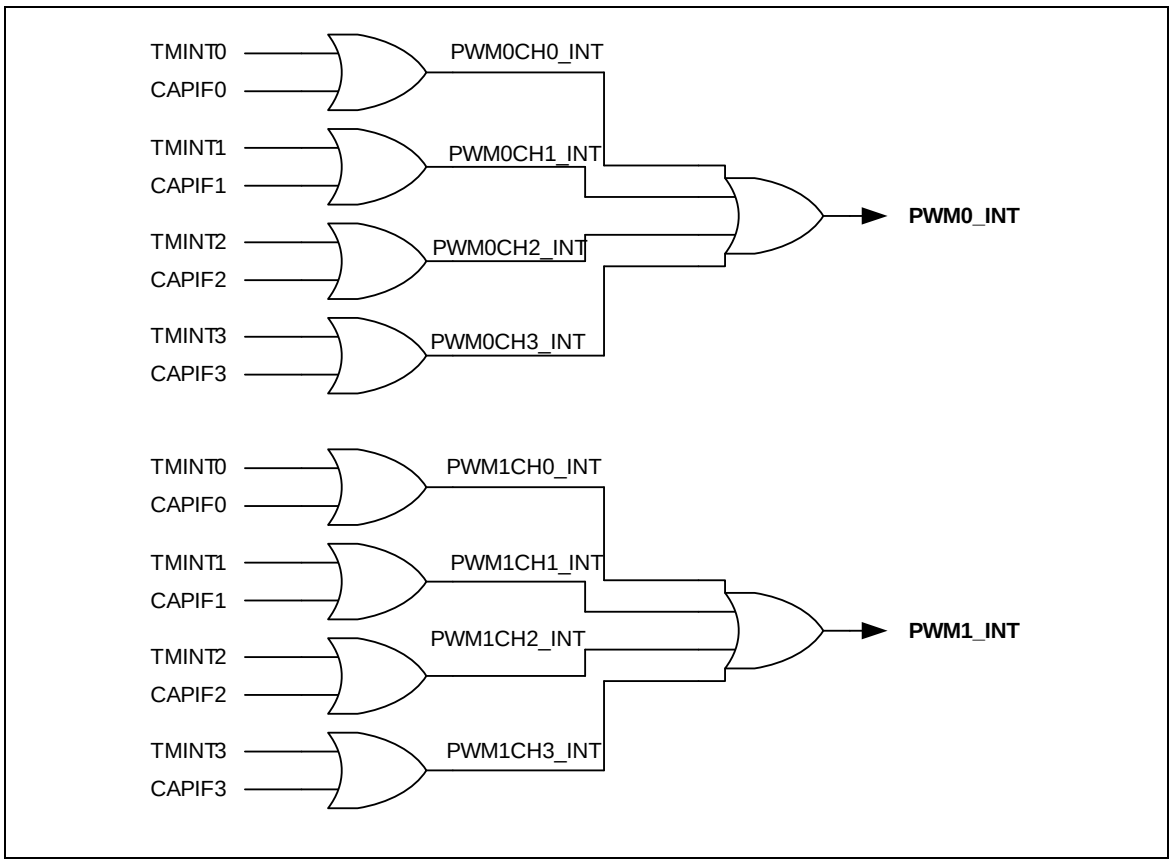



Figure 5-80 PWM-Timer Interrupt

#### 5.14.4.8 PWM-Timer Start Procedure

The following procedure is for starting a PWM drive.

- Setup clock selector (PWMx_CLKSEL),x=0~1
- Setup prescaler (PWMx_PRES),x=0~1
- Setup inverter on/off, dead zone generator on/off, auto-reload/one-shot mode and Stop PWM-timer (PWMx_CTL, x=0~1)
- Setup interrupt enable register (PWMx_INTEN),x=0~1
- Setup PWM output enable (PWMx_OE),x=0~1

- Setup the corresponding GPI/O pins to PWM function
- Setup the corresponding GPI/O pins to output type
- Enable PWM down-counter start running (Set ChxEN = 1 in PWMx_CTL, x=0~1)
- Setup CM and CN of PWMx_DUTYy register for setting PWM duty, x=0~1,y=0~3.(When cascade is enabled the CM is used for the upper half word of the 32 bit CN)
- The procedure 1~8 mentioned above may be set up not in the order and PWM Timer can still work fine

#### 5.14.4.9 PWM-Timer Stop Procedure

Take PWM0, Channel 0 for example:

Method 1:

Set 32 bit PWMx_DUTY0 register to 0, and wait for PWM timeout interrupt (if bit 0 of PWMx_IE is set) occurring or polling the corresponding time-out flag. When PWM timeout interrupt occurred or the flag is set, disable PWM-Timer (CH0EN in PWMx_CTL). (Recommended)

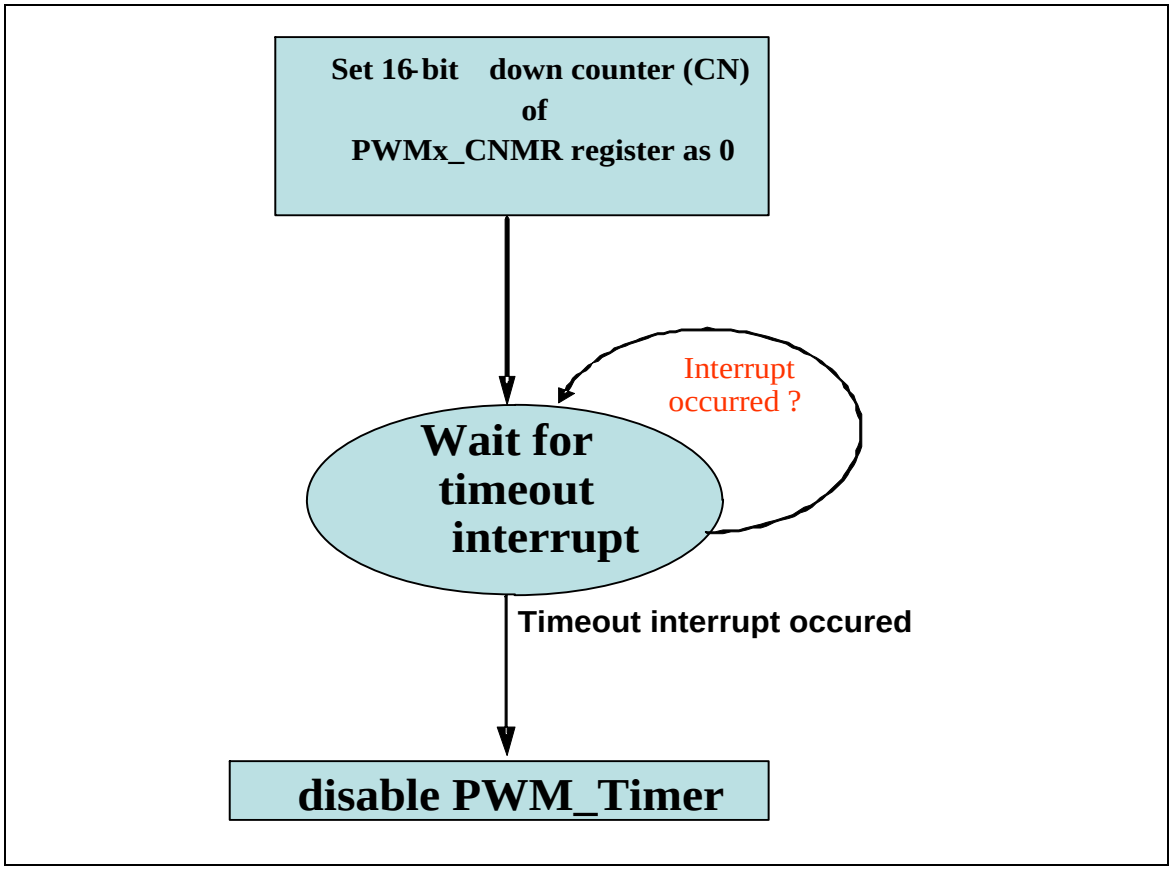



Figure 5-81 PWM-Timer Stop Method 1

Method 2:

Disable PWM-Timer directly ((ChxEN in PWMx_CTL). (Not recommended)

The reason is that disabling ChxEN will immediately stop PWM output signal and lead to change the duty of the PWM output, this may cause damage to the control circuit of motor.

5.14.4.10 Capture Start Procedure

- Setup clock selector (PWMx_CLKSEL),x=0~1
- Setup prescaler (PWMx_PRES),x=0~1
- Setup channel enabled, rising/falling interrupt enable and input signal inverter on/off (PWMx_CAPCTL,PWMx_CAPINTEN),x=0~1
- Setup PWM down-counter (CN) of PWMx_DUTYy, x= 0~1, y=0~3 register
- Set Capture Input Enable Register (PWMx_CAPCTL),x=0~1
- Setup the corresponding GPI/O pins to PWM function
- Setup the corresponding GPI/O pins to input type
- Enable PWM down-counter start running (Set ChyEN = 1 in PWMx_CTL), x=0~1,y=0~3

## 5.15 RTC

### 5.15.1 Overview

Real Time Clock (RTC) unit provides user the real time and calendar message. The Clock Source of RTC is from an external 32.768 kHz crystal connected at pins X32I and X32O (reference to pin Description) or from an external 32.768 kHz oscillator output fed at pin X32I. The RTC unit provides the time message (second, minute, hour) in Time Loading Register (TLR) as well as calendar message (day, month, year) in Calendar Loading Register (CLR). The data message is expressed in BCD format. This unit offers alarm function that user can preset the alarm time in Time Alarm Register (TAR) and alarm calendar in Calendar Alarm Register (CAR).

The RTC unit supports periodic Time Tick and Alarm Match interrupts. The periodic interrupt has 8 period options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second which are selected by TTR (TTR[2:0]). When RTC counter in TLR and CLR is equal to alarm setting time registers TAR and CAR, the alarm interrupt status (RIIR.AIS) is set and the alarm interrupt is requested if the alarm interrupt is enabled (RIER.AIER=1). The RTC Time Tick (if wake-up CPU function is enabled, RTC_TTR[TWKE] high) and Alarm Match can cause CPU wake-up from idle or Power-down mode.

### 5.15.2 Features

- One time counter (second, minute, hour) and calendar counter (day, month, year) for user to check the time
- Alarm register (second, minute, hour, day, month, year)
- 12-hour or 24-hour mode is selectable
- Leap year compensation automatically
- Day of week counter
- Frequency compensate register (FCR)
- All time and calendar message is expressed in BCD code
- Supports periodic time tick interrupt with 8 period options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
- Supports RTC Time Tick and Alarm Match interrupt
- Supports wake-up CPU from Power-down mode
- Supports 80 bytes spare registers and a snoop pin to clear the content of these spare registers

### 5.15.3 Block Diagram

The block diagram of Real Time Clock is depicted as follows.

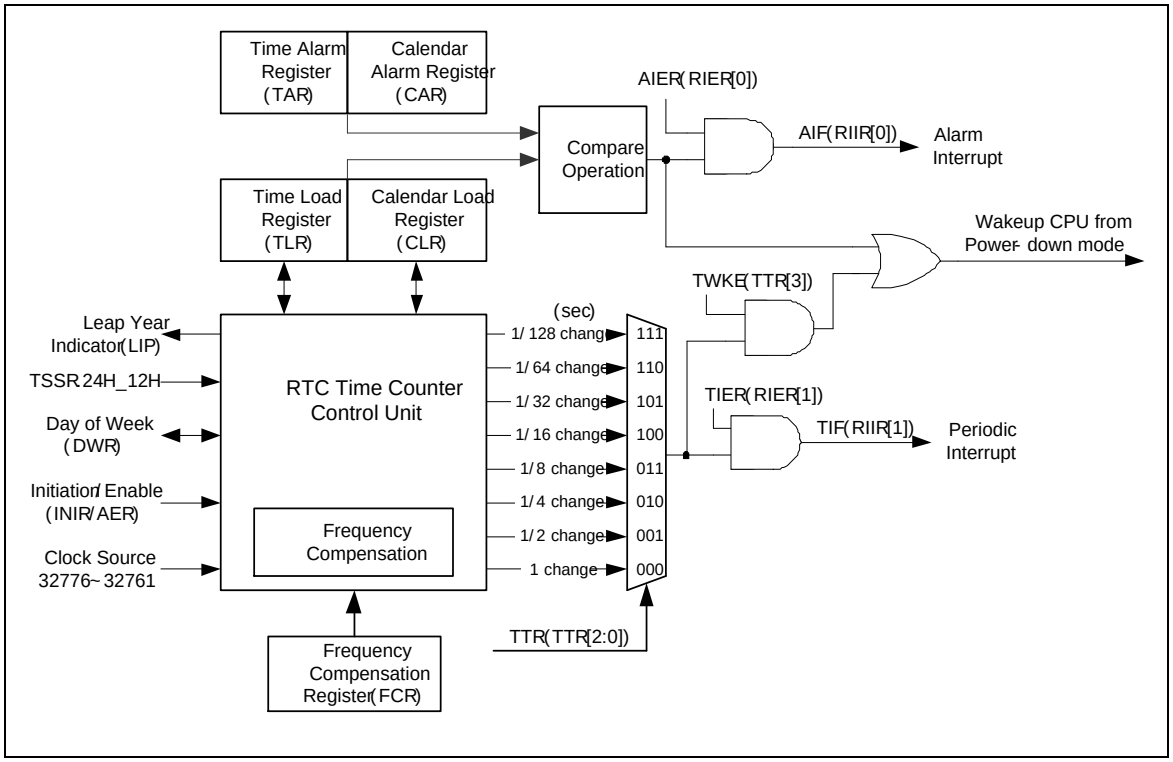



Figure 5-82 RTC Block Diagram

### 5.15.4 Functional Description

5.15.4.1 Access to RTC register

Due to clock difference between RTC clock and system clock, when user write new data to any one of the registers, the register will not be updated until 2 RTC clocks later (60us).

In addition, user must be aware that RTC block does not check whether loaded data is out of bounds or not. RTC does not check rationality between DWR and CLR either.

5.15.4.2 RTC Initiation

When RTC block is powered on, user has to write a number (0xa5eb1357) to INIR to reset all logic. INIR acts as hardware reset circuit. Once INIR has been set as 0xa5eb1357, there is no action for RTC if any value is programmed into INIR register.

5.15.4.3 RTC Read/Write Enable

Register AER bit 15~0 is served as RTC read/write password to protect RTC registers. AER bit 15~0 has to be set as 0xa965 to enable access restriction. Once it is set, it will take effect at least 512 RTC clocks (about 15ms). Programmer can read RTC enabled status flag in AER.ENF to check whether if RTC unit start operating or not.

5.15.4.4 Frequency Compensation

The RTC FCR allows software to make digital compensation to a clock input. The frequency of clock input must be in the range from 32776 Hz to 32761 Hz. The cycle of RTC frequency compensation is 60 seconds. It will compensate the input crystal via writing RTC_FCR register and the precision is ±0.4768 ppm (±0.0412 sec/day). Please follow the example and formula below to write the actual frequency of 32k crystal to RTC_FCR register. Following are the compensation examples for higher or lower than 32768 Hz.

Example 1:

Frequency counter measurement : 32773.65 Hz (> 32768 Hz)

Integer part: 32773

FCR.Integer = (32773 – 32768) + 7 = 12 = 0x0c

Fraction part: 0.65

FCR.Fraction = 0.65 x 64 = 41.6 => 0x29

Example 2:

Frequency counter measurement : 32763.25 Hz (< 32768 Hz)

Integer part: 32763

FCR.Integer = (32763 – 32768) + 7 = 2 = 0x02

Fraction part: 0.25

FCR.Fraction = 0.25 x 64 = 16 => 0x10

Note:

The value of RTC_FCR register will be as default value (0x0000_0700) while the compensation is not executed. User can utilize a frequency counter to measure RTC clock source via RTC time ticks interrupt event on one of GPI/O pin in manufacturing, and store the value in Flash memory for retrieval when the product is first power on. In the meanwhile, user can use RTC time ticks interrupt event to check the result of RTC frequency compensation.

#### 5.15.4.5 Time and Calendar counter

TLR and CLR are used to load the time and calendar. TAR and CAR are used for alarm. They are all represented by BCD.

#### 5.15.4.6 12/24 hour Time Scale Selection

The 12/24 hour time scale selection depends on TSSR bit 0.

#### 5.15.4.7 Day of the week counter

The RTC unit provides day of week in Day of the Week Register (DWR). The value is defined from 0 to 6 to represent Sunday to Saturday respectively.

#### 5.15.4.8 Periodic Time Tick Interrupt

The periodic interrupt has 8 period option 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second which are selected by TTR.TTR[2:0]. When periodic time tick interrupt is enabled by setting RIER.TIER to 1, the Periodic Time Tick Interrupt is requested periodically in the period selected by TTR register.

#### 5.15.4.9 Alarm Interrupt

When RTC counter in TLR and CLR is equal to alarm setting time TAR and CAR the alarm interrupt status (RIIR.AIS) is set and the alarm interrupt is requested if the alarm interrupt is enabled (RIER.AIER=1).

Application Note:

TAR, CAR, TLR and CLR registers are all BCD counter.

Programmer has to make sure that the loaded values are reasonable. For example, Load CLR as 201a (year), 13 (month), 00 (day), or CLR does not match with DWR, etc.

Reset state :

| Register | Reset State |
|---|---|
| AER | 0 |
| CLR | 05/1/1 (year/month/day) |
| TLR | 00:00:00 (hour : minute : second) |
| CAR | 00/00/00 (year/month/day) |
| TAR | 00:00:00 (hour : minute : second) |
| TSSR | 1 (24 hr mode) |
| DWR | 6 (Saturday) |
| RIER | 0 |
| RIIR | 0 |
| LIR | 0 |
| TTR | 0 |

In TLR and TAR, only 2 BCD digits are used to express “year”. It is assumed that 2 BCD digits of xY denote 20xY, but not 19xY or 21xY.

#### 5.15.4.10 Spare registers and snoop pin

The RTC is equipped with 80 bytes spare registers to store important user information, and also has a snoop function to detect the transition of snoop pin. Once the transition defined in register RTC_SPRCTL[1] is detected in snoop pin, the 80 bytes spare registers will be cleared by RTC automatically.

As these 80 bytes spare registers locates in LXT (32.768 kHz) clock domain (it's asynchronous with system clock domain), a synchronization latency is necessary when writing data to these 80 bytes spare registers. Once CPU writes one of 20 spare registers (RTC_SPR0 ~ RTC_SPR19), it's necessary to polling bit SPRRDY (RTC_SPRCTL[7]) to check if data is written into registers. CPU could only access (read or write) the spare registers again once SPRRDY is high. Any access (read or write) to spare registers while SPRRDY low is undefined.

## 5.16 Smart Card Host Interface (SC)

### 5.16.1 Overview

The Smart Card Interface controller (SC controller) is based on ISO/IEC 7816-3 standard and fully compliant with PC/SC Specifications. It also provides status of card insertion/removal.

### 5.16.2 Features

- ISO-7816-3 T = 0, T = 1 compliant
- EMV2000 compliant
- Supports up to three ISO-7816-3 ports
- Separates receive / transmit 4 byte entry buffer for data payloads
- Programmable transmission clock frequency
- Programmable receiver buffer trigger level
- Programmable guard time selection (11 ETU ~ 266 ETU)
- A 24-bit and two 8-bit counters for Answer to Reset (ATR) and waiting times processing
- Supports auto inverse convention function
- Supports stop clock level and clock stop (clock keep) function
- Supports transmitter and receiver error retry and error number limitation function
- Supports hardware activation sequence process
- Supports hardware warm reset sequence process
- Supports hardware deactivation sequence process
- Supports hardware auto deactivation sequence when detected the card removal.
- Support UART mode
  - Half duplex, asynchronous communications
  - Separate receiving / transmitting 4 bytes entry FIFO for data payloads
  - Support programmable baud rate generator for each channel
  - Support programmable receiver buffer trigger level
  - Programmable transmitting data delay time between the last stop bit leaving the TX-FIFO and the de-assertion by setting SCx_EGTR [EGT] register
  - Programmable even, odd or no parity bit generation and detection
  - Programmable stop bit, 1 or 2 stop bit generation

### 5.16.3 Block Diagram

The SC clock control and block diagram are shown as follows. The SC controller is completely asynchronous design with to clock domains, PCLK and engine clock, note that the PCLK should be higher than or equal to the frequency of engine clock.

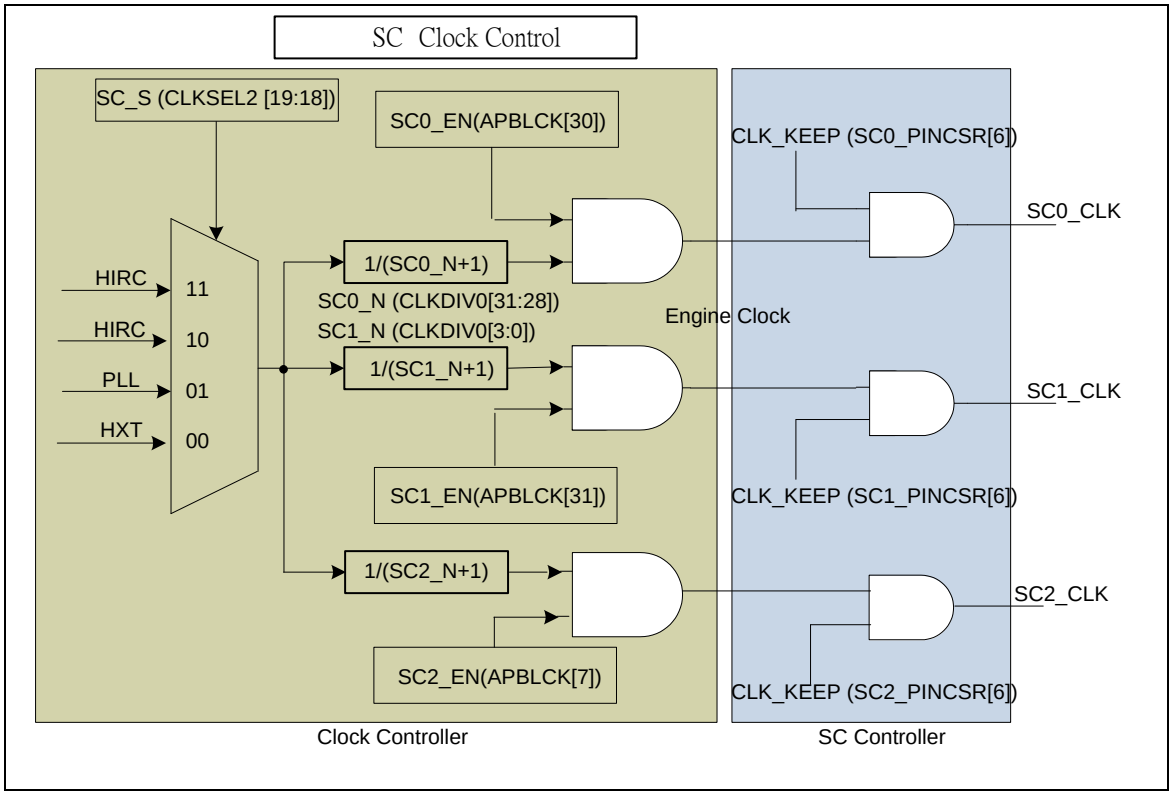



Figure 5-83 SC Clock Control Diagram (4-bit Pre-Scale Counter in Clock Controller)

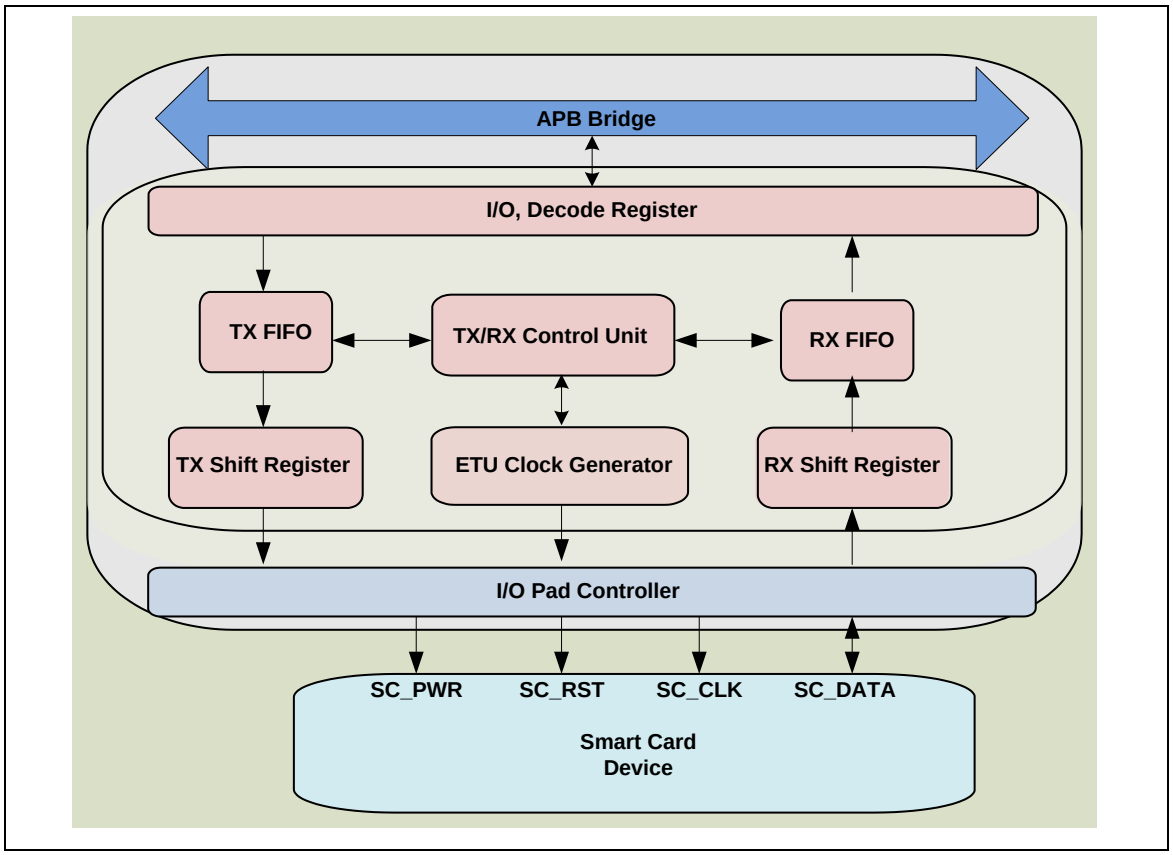



Figure 5-84 SC Controller Block Diagram

### 5.16.4 Functional Description

Basically, the smart card interface acts as a half-duplex asynchronous communication port and its data format is composed of ten consecutive bits which is shown as follows.

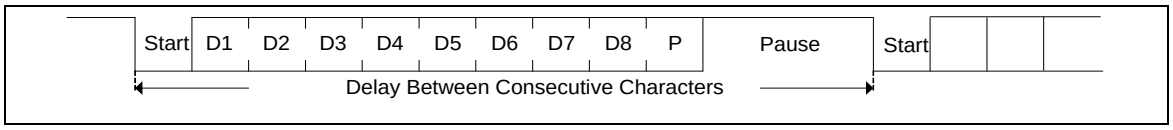



Figure 5-85 SC Data Character

5.16.4.1 Activation, Warm Reset and Deactivation Sequence

**Activation**

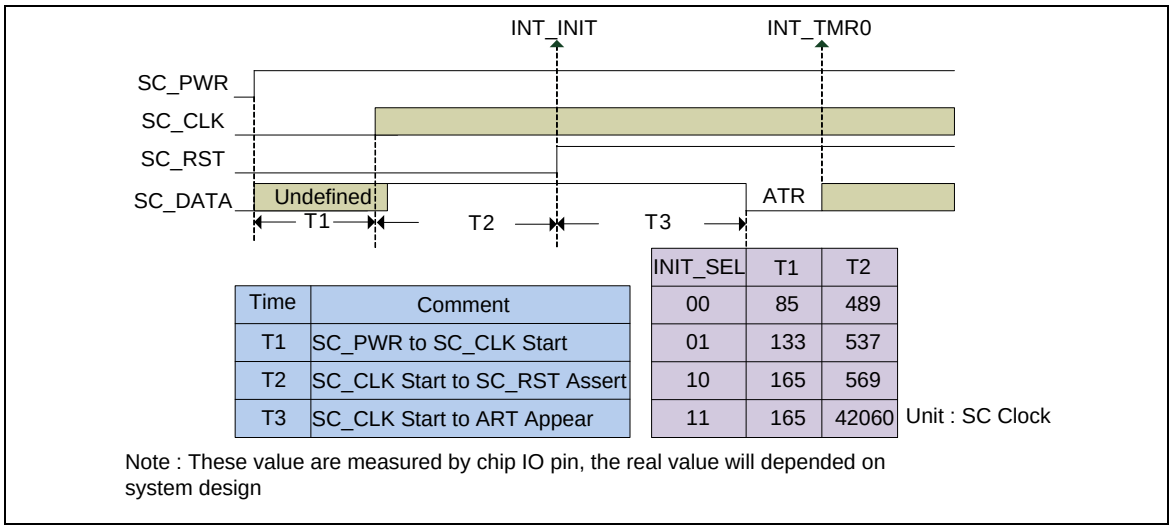
The Smart Card Interface controller supports hardware activation, warm reset and deactivation sequence. The activation sequence is shown as follows.

- Set SC_RST to low
- Set SC_PWR at high level and SC_DATA in reception mode.
- Enable SC_CLK clock
- De-assert SC_RST to high

The activation sequence can be controlled by software or hardware. If software wants to control it, software can control SC_PINCSR and SC_TMRx register to process the activation sequence or setting SC_ALTCTL [ACT_EN] register, and then the interface will perform the hardware activation sequence.

Following is activation control sequence in hardware activation mode:

- Set activation timing by setting SC_ALTCTL [INIT_SEL].
- TMR0 can be selected when SC_CTL [TMR_SEL] is 01, 10 or 11.
- Set operation mode SC_TMR0 [MODE] to 0011 and give an Answer to Reset value by setting SC_TMR0 [CNT] register.
- When hardware de-asserts SC_RST to high, hardware will generator an interrupt INT_INIT to CPU at the same time (SC_IER[INIT_IE] = “1”)
- If the TMR0 decreases the counter to “1” (start from SC_RST) and the card does not response ATR before that time, hardware will generate interrupt INT_TMR0 to CPU.




| Time | Comment |
|---|---|
| T1 | SC_PWR to SC_CLK Start |
| T2 | SC_CLK Start to SC_RST Assert |
| T3 | SC_CLK Start to ART Appear |

| INIT_SEL | T1 | T2 |
|---|---|---|
| 00 | 85 | 489 |
| 01 | 133 | 537 |
| 10 | 165 | 569 |
| 11 | 165 | 42060 |

Unit : SC Clock

Note : These value are measured by chip IO pin, the real value will depended on system design

Figure 5-86 SC Activation Sequence

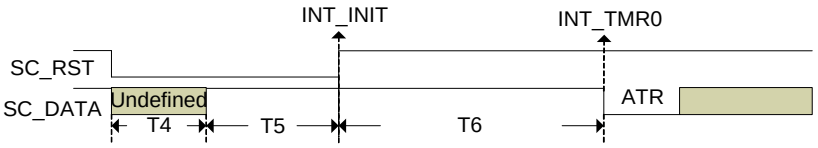
**Warm Reset**

The warm reset sequence is shown as follows.

The warm reset sequence can be controlled by software or hardware. If software wants to control it, software can control SC_PINCSR and SC_TMRx register to process the warm reset sequence or set SC_ALTCTL [WARST_EN] register, and then the interface will perform the hardware warm reset sequence.

Following is warm reset control sequence in hardware warm reset mode

- Set warm reset timing by setting SC_ALTCTL [INIT_SEL].
- Select TMR0 by setting SC_CTL [TMR_SEL] register (TMR_SEL can be 01, 10, or 11).
- Set operation mode SC_TMR0 [MODE]) to 011 and give an Answer to Reset value by setting SC_TMR0 [CNT] register.
- Set TMR0_SEN and WARST_EN to start counting by. SC_ALTCTL register.
- When hardware de-asserts SM_RST to high, hardware will generate an interrupt INT_INIT to CPU at the same time (SC_IER[INIT_IE] = “1”)
- If the TMR0 decrease the counter to “1” (start from SC_RST) and the card does not response ATR before that time, hardware will generate interrupt INT_TMR0 to CPU.




| Time | Comment |
|---|---|
| T4 | SC_RST to SC_DATA Reception Mode |
| T5 | SC_DATA Reception Mode to SC_RST Assert |
| T6 | SC_CLK Start to ATR Appear |

| INIT_SEL | T4 | T5 |
|---|---|---|
| 00 | 81 | 483 |
| 01 | 129 | 531 |
| 10 | 161 | 563 |
| 11 | 161 | 42106 |

Unit:SC Clock

Note: This value is measured by chip IO pin, the real value will depended on system design

Figure 5-87 SC Warm Reset Sequence

**Deactivation**

The deactivation sequence is shown as follows:

- Set SC_RST to low
- Stop SC_CLK
- Set SC_DATA to state low
- Deactivated SC_PWR

The deactivation sequence can be controlled by software or hardware. If software wants to control it, software can control SC_PINCSR and SC_TMR0 register to process the deactivation sequence or set SC_ALTCTL [DACT_EN] register, and then the interface will perform the hardware deactivation sequence.

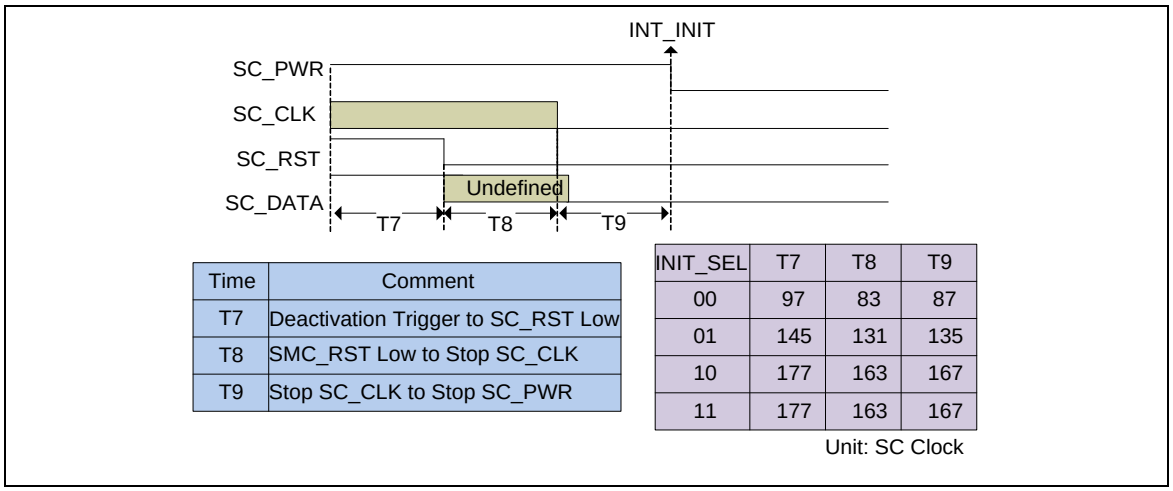
The SC controller also supports auto deactivation sequence when the card removal detection is set (SC_PINCSR [A)AC_CDEN)).

Following is deactivation control sequence in hardware deactivation mode:

Set deactivation timing by setting SC_ALTCTL [INIT_SEL].

Set DACT_EN to start counting by. SC_ALTCTL register.

When hardware de-asserts SC_PWR to low, controller will generate an interrupt INT_INIT to CPU at the same time (SC_IER[INIT_IE] = “1”).




| Time | Comment |
|---|---|
| T7 | Deactivation Trigger to SC_RST Low |
| T8 | SMC_RST Low to Stop SC_CLK |
| T9 | Stop SC_CLK to Stop SC_PWR |

| INIT_SEL | T7 | T8 | T9 |
|---|---|---|---|
| 00 | 97 | 83 | 87 |
| 01 | 145 | 131 | 135 |
| 10 | 177 | 163 | 167 |
| 11 | 177 | 163 | 167 |

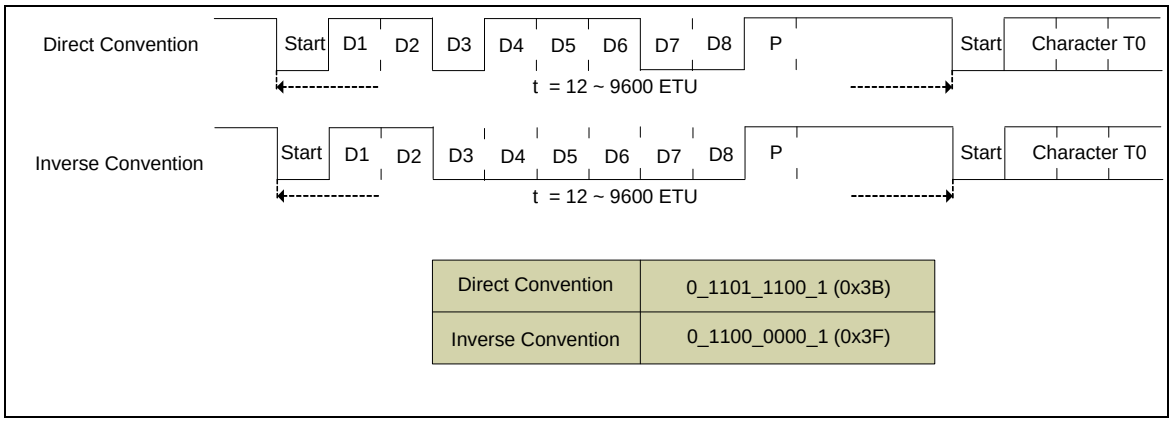
Unit: SC Clock

Figure 5-88 SC Deactivation Sequence

5.16.4.2 Initial Character TS

According to 7816-3, the initial character TS of answer to request (ATR) has two possible patterns (as shown in the following figure). If the TS pattern is 0_1100_0000_1, it is inverse convention. When decoded by inverse convention, the conveyed byte is equal to '3F'. If the TS pattern is 0_1101_1100_1, it is direct convention. When decoded by direct convention, the conveyed byte is equal to '3B'.Software can set SC_CTL [AUTO_CON_EN] and then the operating convention will be decided by hardware. Software can also set the SC_CTL [CON_SEL] register (set to 00 or 11) to change the operating convention after SC received TS of answer to request (ATR).

If software enables auto convention function by setting SC_CTL [AUTO_CON_EN] register, the setting step must be done before Answer to Reset state and the first data must be 0x3B or 0x3F. After hardware received first data and stored it at buffer, the hardware will decided the convention and change the SC_CTL [CON_SEL] register automatically. If the first data is neither 0x3B nor 0x3F, the hardware will generator an interrupt INT_ACON_ERR (if SC_IER [ACON_ERR_IE] = "1") to CPU.




| | |
|---|---|
| Direct Convention | 0_1101_1100_1 (0x3B) |
| Inverse Convention | 0_1100_0000_1 (0x3F) |

Figure 5-89 Initial Character TS

5.16.4.3 Error Signal and Character Repetition

According to 7816-3 T=0 mode description, as shown in following, if the receiver receives a wrong parity bit, it will pull the SC_DATA to low one to two bit period to inform the transmitter parity error. Then the transmitter will retransmit the character. The SC interface controller supports hardware error detection function in receiver and supports hardware re-transmit function in transmitter. Software can enable re-transmit function by setting SC_CTL [TX_ERETRY_EN]. Software can

also define the retry (re-transmit) number limitation in SC_CTL [TX_ERETRY] register. If the re-transmit number is greater than SC_CTL [TX_ERETRY], transmitter will transfer the next new data to device and generate an interrupt INT_TERR (if SC_IER [TERR_IE] = “1”) to CPU. If the number of received errors by receiver is greater than SC_CTL [RX_ERETRY], receiver will receive this error data to buffer and generate an interrupt INT_TERR(if SC_IER [TERR_IE] = “1”) to CPU.

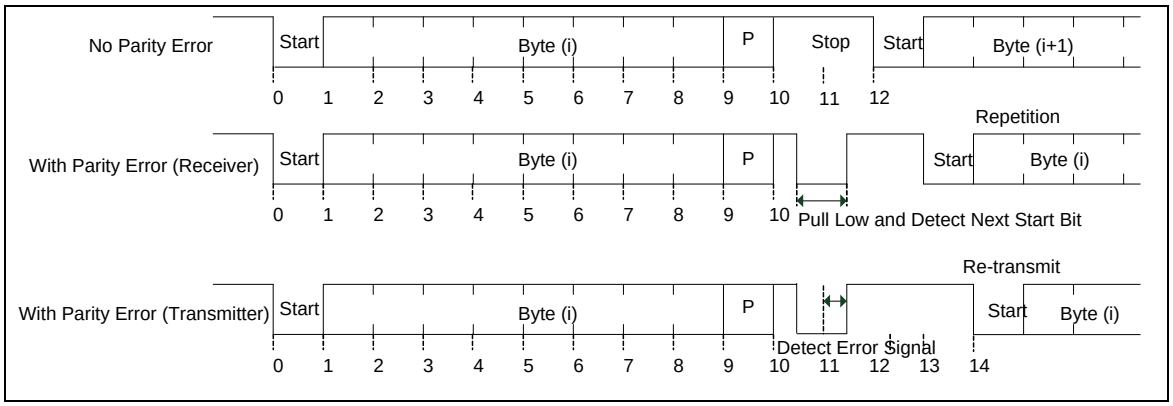



Figure 5-90 SC Error Signal

5.16.4.4 Internal Time-Out Counter

The smart card interface includes a 24-bit time-out counter and two 8-bit time-out counters. These counters help the controller in processing different real-time interval (ATR, WWT, BWT, etc.). Each counter can be set to start counting once the trigger enable bit has been written or a START bit has been detected.

The following is the programming flow:

- Enable / Disable counter by setting SC_CTL[TMR_SEL]
- Select operation mode ([MODE]) and give a count value ([CNT]) by setting SC_TMRx register.
- Set TMRx_SEN to start counting.

5.16.4.5 UART Mode

When the SCx_UACTL [UA_MODE_EN] bit is set, the Smart Card Interface controller can also be used as UART function without flow control. In UART mode, SC data (SCx_DATA) pin will be used as UART RXD and SC clock (SCx_CLK) pin will be used as UART TXD. The following is the programming example for UART mode.

Programming example:

1. Enter UART mode by setting the SCx_UACTL[UA_MODE_EN] bit .
2. Do software reset by setting the SCx_ALTCTL[RX_RST] and SCx_ALTCTL[TX_RST] bit to ensure that all state machines return to idle state.
3. Fill “0” to the SCx_CTL [CON_SEL] and SCx_CTL [AUTO_CON_EN] field. (in UART mode, those fields must be “0”)
4. Select the UART baud rate by setting SCx_ETUCR [ETU_RDIV] fields.
   - Baud rate = f / (ETU_RDIV+1), where f is SMC engine clock frequence (SCx_CLK), the effective ETU_RDIV is between 0x04 to 0xFFF. The value that less than 0x04 will be regarded as 0x04.
   - For example, if user wants to set the baud rate as 115200, and SC clock frequency is 12 MHz, ETU_RDIV should be set to 0x67 and the error rate is around 0.16%.

5. Select the data format including data length (by setting SCx_UA_CTL [DATA_LEN]), parity format (by setting SCx_UA_CTL [OPB] and SCx_UA_CTL [PBDIS] bit) and stop bit length (by setting SCx_CTL [SLEN] or SCx_EGTR [EGT]).
6. Select the receiver buffer trigger level by setting the SCx_CTL [RX_FTRI_LEV] field and select the receiver buffer time-out value by setting the SCx_RFTMR [RFTMR] field.
7. Write the SCx_THR (TX) register or read the SC_RBR (RX) register to perform UART function.

## 5.17 SPI

### 5.17.1 Overview

The Serial Peripheral Interface (SPI) is a synchronous serial data communication protocol. Devices communicate in Master/Slave mode with 4-wire bi-direction interface. It is used to perform a serial-to-parallel conversion on data received from a peripheral device, and a parallel-to-serial conversion on data transmitted to a peripheral device. The SPI controller can be configured as a master or a slave devicee.

The SPI controller supports wake-up function. When this chip stays in power-down mode, it can be waked up chip by off-chip device.

This controller supports variable serial clock function for special application and 2-bit transfer mode to connect 2 off-chip slave devices at the same time. The SPI controller also supports PDMA function to access the data buffer.

### 5.17.2 Features

- Supports Master (max. 32 MHz) or Slave (max. 16 MHz) mode operation
- Supports 1 bit and 2 bit transfer mode
- Support Dual IO transfer mode
- Configurable bit length of a transaction from 8 to 32-bit
- Supports MSB first or LSB first transfer sequence
- Two slave select lines supported in Master mode
- Configurable byte or word suspend mode
- Supports byte re-ordering function
- Supports variable serial clock in Master mode
- Provide separate 8-level depth transmit and receive FIFO buffer
- Supports wake-up function
- Supports PDMA transfer
- Supports three wires, no slave select signal, bi-direction interface

### 5.17.3 Block Diagram

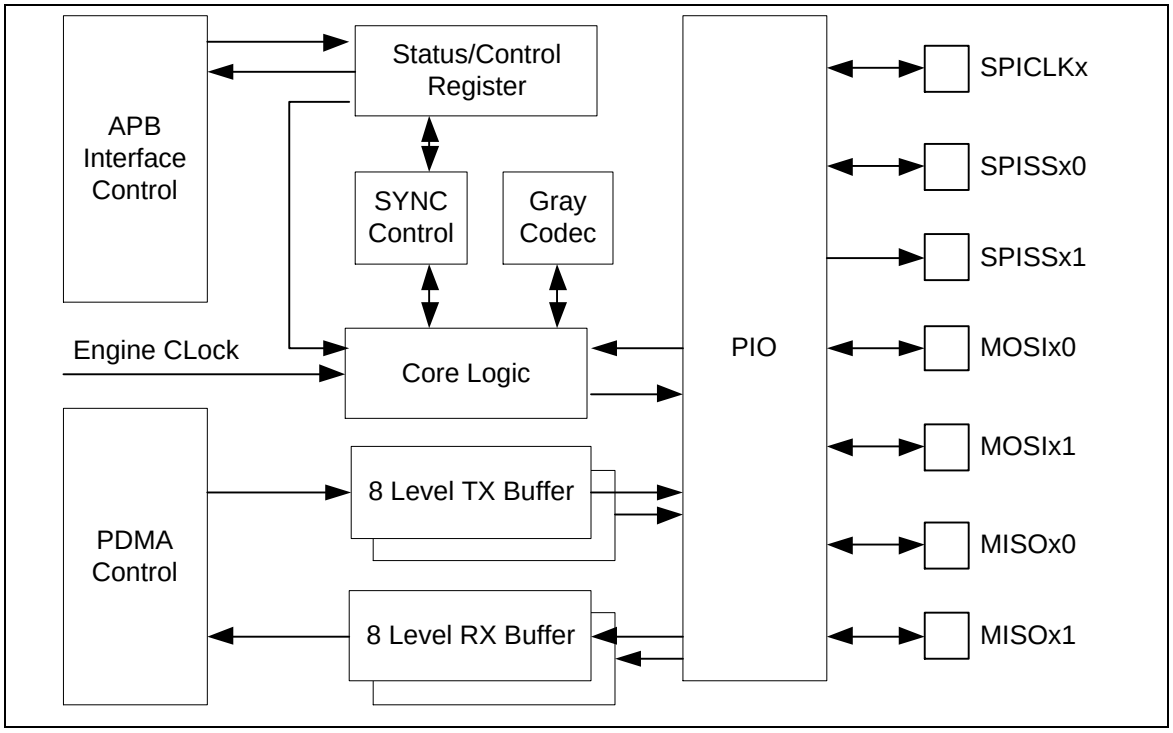



Figure 5-91 SPI Block Diagram

Where "x" indicates the number of SPI block. Ex: SPICLK0 is the 1st SPI block serial clock port.

### 5.17.4 Functional Description

#### 5.17.4.1 SPI Engine Clock and Serial Clock

SPI controller needs the SPI engine clock to drive the SPI logic unit to perform the data transfer. The SPI engine clock rate is determined by the settings of clock source and clock divisor. The SPIx_S bit of CLKSEL2 register determines the clock source of the SPI engine clock. The clock source can be HCLK or PLL output clock. The DIVIDER setting of SPI_DIVIDER register determines the divisor of the clock rate calculation.

In master mode, if the variable clock function is disabled, the output frequency of the serial clock output pin is equal to the SPI engine clock rate. In slave mode, the SPI serial clock is provided by an off-chip master device. The SPI engine clock rate of slave device must be faster than the serial clock rate of the master device connected together. The frequency of SPI engine clock cannot be faster than the APB clock rate regardless of master mode or slave mode.

#### 5.17.4.2 Master/Slave Mode

This SPI controller can be set as master or Slave mode by setting the SLAVE bit (SPI_CTL[18]) to communicate with the off-chip SPI slave or master device. The application block diagrams in Master and Slave mode are shown below.

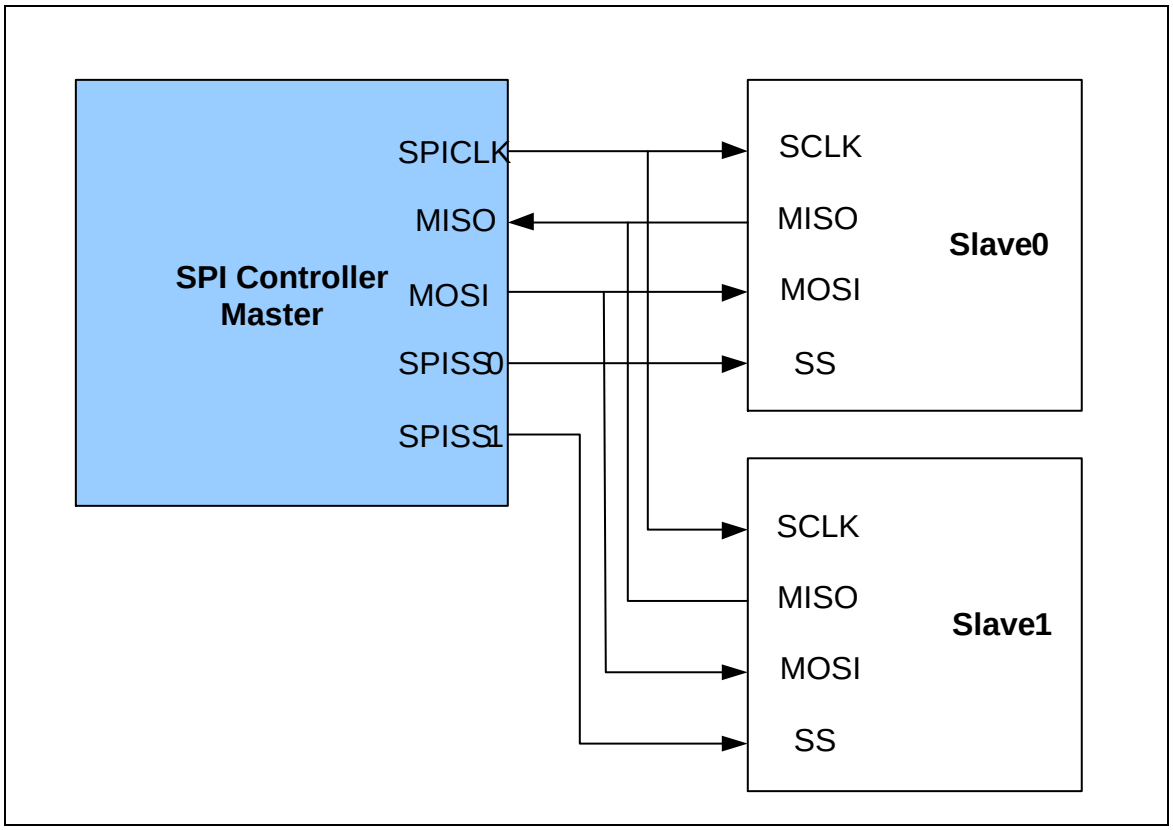



Figure 5-92 SPI Master Mode Application Block Diagram

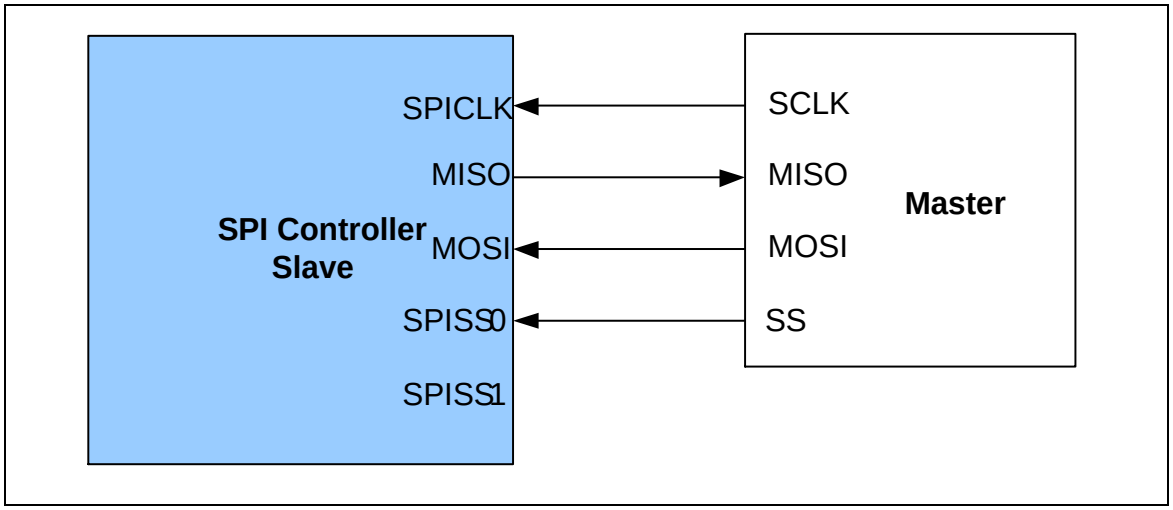



Figure 5-93 SPI Slave Mode Application Block Diagram

5.17.4.3 Slave Selection

In Master mode, this SPI controller can drive up to two off-chip slave devices through the slave select output signals SPIxSS0 and SPIxSS1, but it is a time-sharing operation and it can not operate with two slave devices simultaneously.

In Slave mode, the off-chip master device drives the slave select signal from the SPIxSS0 port to

this SPI controller. In Master/Slave mode, the active level of slave select signal can be programmed to low active or high active in SS_LVL bit (SPI_SSR[2]), and the SS_LTRIG bit (SPI_SSR[4]) define the slave select signal SPIxSS0/1 is level trigger or edge trigger. The selection of trigger condition depends on what type of peripheral slave/master device is connected.

In Slave mode, if the SS_LTRIG bit is configured as level trigger, the LTRIG_FLAG bit (SPI_SSR[4]) is used to indicate if the count of received bits among one transaction meets the requirement which define in TX_BIT_LEN.

#### 5.17.4.4 Level-trigger / Edge-trigger

In Slave mode, the slave select signal can be configured as level-trigger or edge-trigger. In edge-trigger, the data transfer starts from an active edge and ends on an inactive edge. If master does not send an inactive edge to slave, the transfer procedure will not be completed and the unit transfer interrupt flag of slave will not be set. In level-trigger, the following two conditions will terminate the transfer procedure and the unit transfer interrupt flag of slave will be set. The first condition is that if the number of transferred bits matches the settings of TX_BIT_LEN, the unit transfer interrupt flag of slave will be set. The second condition, if master set the slave select pin to inactive level during the transfer is in progress, it will force slave device to terminate the current transfer no matter how many bits have been transferred and the unit transfer interrupt flag will be set. User can read the status of LTRIG_FLAG bit to check if the data has been completely transferred.

#### 5.17.4.5 Automatic Slave Select

In Master mode, if the AUTOSS bit (SPI_SSR[3]) is set as 1, the slave select signals will be generated automatically and output to SPIxSS0 and SPIxSS1 ports according to SSR[0] (SPI_SSR[0]) and SSR[1] (SPI_SSR[1]) whether it is enabled or not. It means that the slave select signals, which is enabled in SPI_SSR[1:0] register is asserted by the SPI controller when the SPI data transfer is started by setting the GO_BUSY bit (SPI_CTL[0]) and is de-asserted after the data transfer is finished. If the AUTOSS bit is cleared to 0, the slave select output signals are asserted and de-asserted by manual setting and clearing the related bits in SPI_SSR[1:0] register. The active level of the slave select output signals is specified in SS_LVL bit (SPI_SSR[2]).

In master mode, if the value of SP_CYCLE[3:0] is less than 3 and the AUTOSS is set as 1, the slave select signal will keep at active state between two successive transactions.

In slave mode, to recognize the inactive state of the slave select signal, the inactive period of the slave select signal must be larger than or equal to 6 engine clock periods between two successive transactions.

#### 5.17.4.6 No Slave Select Mode (3-WIRE mode)

When the software sets the NOSLVSEL bit to enable the 3-wire mode, the SPI controller can work with no slave select signal in slave mode. The NOSLVSEL bit only takes effect in slave mode. It only needs three pins, SPICLK, SPI_MISO, and SPI_MOSI, to communicate with a SPI master. The SPISS pin can be configured as a GPIO. When the NOSLVSEL bit is set to 1, the SPI slave will be ready to transmit/receive data after the GO_BUSY bit is set to 1. In 3-wire mode, the SS_LTRIG, SPI_SSR[4], shall be set as 1.

In normal operation, the interrupt flag in SLV_START_INTSTS will be set when the transfer has start and there is also interrupt event when the received data meet the required bits which define in TX_BIT_LEN. If the received bits are less than the requirement and there is no more serial clock input over the time period which is defined by the user in Slave mode with no slave select, the user can set the SLV_ABORT bit to force the current transfer done and then the user can get a transfer done interrupt event.

#### 5.17.4.7 Variable Clock Function

In master mode, if the VARCLK_EN bit (SPI_CTL[23]) is set to 1, the output of serial clock can be programmed as variable frequency pattern. The serial clock period of each cycle depends on the setting of the SPI_VARCLK register. When the variable clock function is enabled, the TX_BIT_LEN setting must be set as 0x10 to configure the data transfer as 16-bit transfer mode. The VARCLK[31] determines the clock period of the first clock cycle. If VARCLK[31] is 0, the first clock cycle depends on the DIVIDER1 setting; if it is 1, the first clock cycle depends on the DIVIDER2 setting. Two successive bits in VARCLK[30:1] defines one clock cycle. The bit field VARCLK[30:29] defines the second clock cycle of SPI serial clock of a transaction, and the bit field VARCLK[28:27] defines the third clock cycle and so on. The VARCLK[0] is unmeaning. The following figure shows the timing relationship among the serial clock (SPICLK), the VARCLK, the DIVIDER1 and the DIVIDER2 registers.

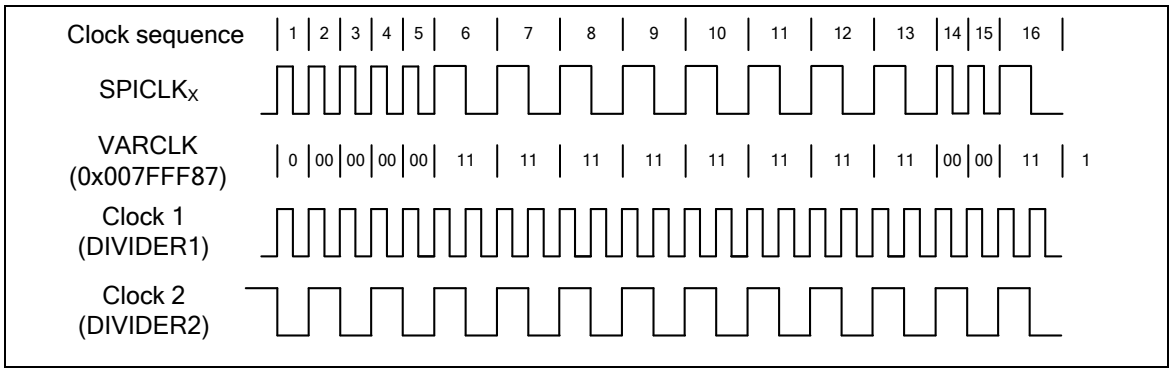



Figure 5-94 SPI Variable Clock Frequency

#### 5.17.4.8 Clock Polarity

The CLKP bit (SPI_CTL[11]) defines the serial clock idle state in Master mode. If CLKP = 1, the output SPICLK is idle at high state. If CLKP = 0, it is idle at low state. For variable serial clock, it works in CLKP = 0 only.

#### 5.17.4.9 Transmitting/Receiving Bit Length

The bit length of a transaction word is defined in TX_BIT_LEN bit (SPI_CTL[7:3]). It can be configured up to 32 bits in a transaction word for transmitting and receiving.

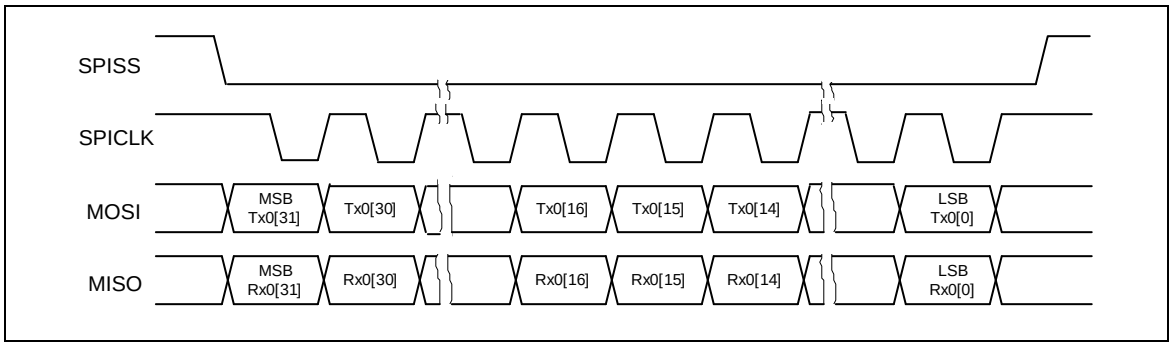



Figure 5-95 SPI 32-bit in One Transaction

#### 5.17.4.10 LSB First

The LSB bit (SPI_CTL[10]) defines the bit transfer sequence in a transaction. If set the LSB bit to 1, the transfer sequence is LSB first. The bit 0 will be transferred firstly. If clear the LSB bit to 0, the transfer sequence is MSB first (refer to the figure above).

The TX_NEG bit (SPI_CTL[2]) defines the data transmitted out either at negative edge or at positive edge of serial clock SPICLK.

#### 5.17.4.11 Receive Edge

The RX_NEG bit (SPI_CTL[1]) defines the data received in either at negative edge or at positive edge of serial clock SPICLK. Note that TX_NEG and RX_NEG must be exclusive.

#### 5.17.4.12 Word Suspend

These four bits field of SP_CYCLE (SPI_CTL[15:12]) provide a configurable clock suspend interval 0.5 ~ 15.5 serial clock periods between two successive transaction words in Master mode. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. The default value of SP_CYCLE is 0x3 (3.5 serial clock cycles). This SP_CYCLE setting will not take effect to the word suspend interval if the software disables the FIFO mode.

If both the VARCLK_EN, SPI_CNTRL[23], and the FIFO bit, SPI_CNTRL[21], are set as 1, the minimum word suspend period is (6.5 + SP_CYCLE)*SPI serial clock period.

#### 5.17.4.13 Byte Reorder

When the transfer is set as MSB first (LSB = 0), the REORDER is enabled, the data stored in the SPI_TX FIFO and SPI_RX FIFO will be rearranged in the order as [BYTE0, BYTE1, BYTE2, BYTE3] in 32 bit transfer mode (TX_BIT_LEN = 0). The sequence of transmitted/received data will be BYTE0, BYTE1, BYTE2, and then BYTE3. If the TX_BIT_LEN is set as 24-bits transfer mode, the data in TX buffer and RX buffer will be rearranged as [unknown byte, BYTE0, BYTE1, BYTE2]. The SPI controller will transmit/receive data with the sequence of BYTE0, BYTE1 and then BYTE2. Each byte will be transmitted/received with MSB first. The rule of 16-bits mode is the same as above. Byte reorder function is only available when TX_BIT_LEN is configured as 16, 24, and 32 bits.

**Note:** The byte reorder function is not supported when the variable serial clock function is enabled.

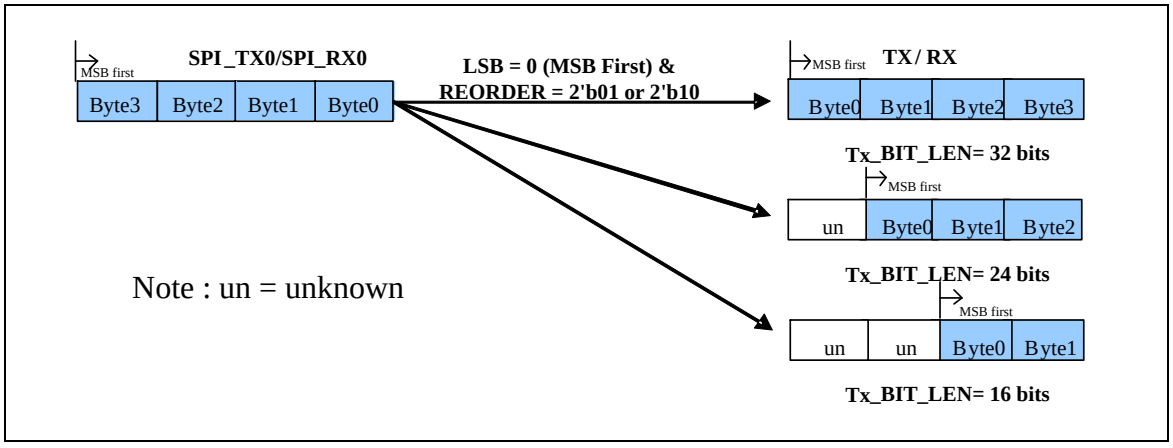



Figure 5-96 SPI Byte Reorder

#### 5.17.4.14 Byte Suspend

In Master mode, if SPI_CTL[19] is set to 1, the hardware will insert a suspend interval of 0.5~15.5 serial clock periods between two successive bytes in a transfer word. Both settings of byte suspend interval and word suspend interval are configured in SP_CYCLE.

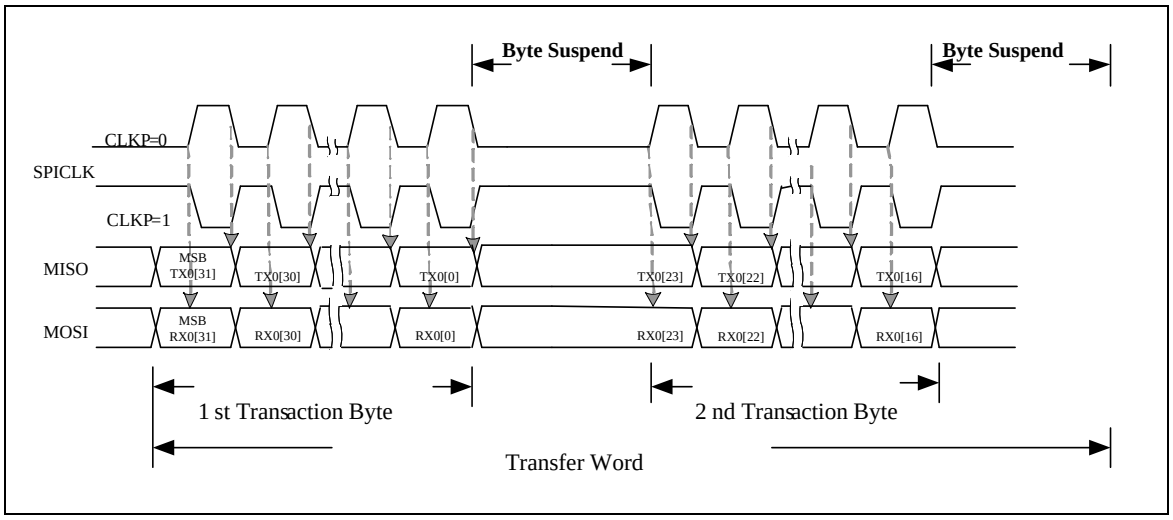



Figure 5-97 SPI Byte Suspend Mode

5.17.4.15 Interrupt

Each SPI controller can generates an individual interrupt when data transfer is finished or the FIFO reach the threshold level and the respective interrupt event flag INTSTS (SPI_STATUS[7], TX_INTSTS (SPI_STATUS[10]), RX_INTSTS (SPI_STATUS[8]) will be set. The interrupt event flag will generates an interrupt to CPU if the interrupt enable bit INTEN (SPI_CTL[17]) is set. The interrupt event flag INTSTS, **TX_INTSTS, and RX_INTSTS** can be cleared only by writing 1 to it.

As the SPI controller finishes a unit transfer, the unit transfer interrupt flag IF (SPI_STS[7]) will be set to 1. The unit transfer interrupt event will generate an interrupt to CPU if the unit transfer interrupt enable bit IE (SPI_CNTRL[17]) is set. The unit transfer interrupt flag can be cleared only by writing 1 to it.

In 3-wire mode, the slave 3-wire mode start interrupt flag, SLV_START_INTSTS, will be set to 1 when the slave senses the SPI clock signal. The SPI controller will issue an interrupt if the SSTA_INTEN is set to 1. If the count of the received bits is less than the setting of TX_BIT_LEN and there is no more serial clock input over the expected time period which is defined by the user, the user can set the SLV_ABORT bit to abort the current transfer. The unit transfer interrupt flag, IF, will be set to 1 if the software set the SLV_ABORT bit.

In FIFO mode, there is time-out function to inform user. If there is a received data in the FIFO and it does not be read by software over 64 SPI engine clock periods in master mode or over 576 SPI engine clock periods in slave mode, it will send a time-out interrupt to the system if the time-out interrupt enable bit, FIFO_CTL[7], is set to 1.

5.17.4.16 Two Bit Transfer Mode

This SPI controller also supports 2-bit transfer mode when enabling the TWOB bit, SPI_CTL[22]. When the TWOB bit is enabled, it can transmit and receive 2-bit serial transfer data simultaneousl[y.]

The 1st bit is through the MOSI0 and MISO0 port to transmit the data from SPI_TX0 register and receive the data into SPI_RX0 regist[er]. The 2nd bit is through the MOSI1and MISO1 port to transmit the data from SPI_TX1 register and receive the data into SPI_RX1 register. The system block and timing of 2-bit transfer mode are shown below.

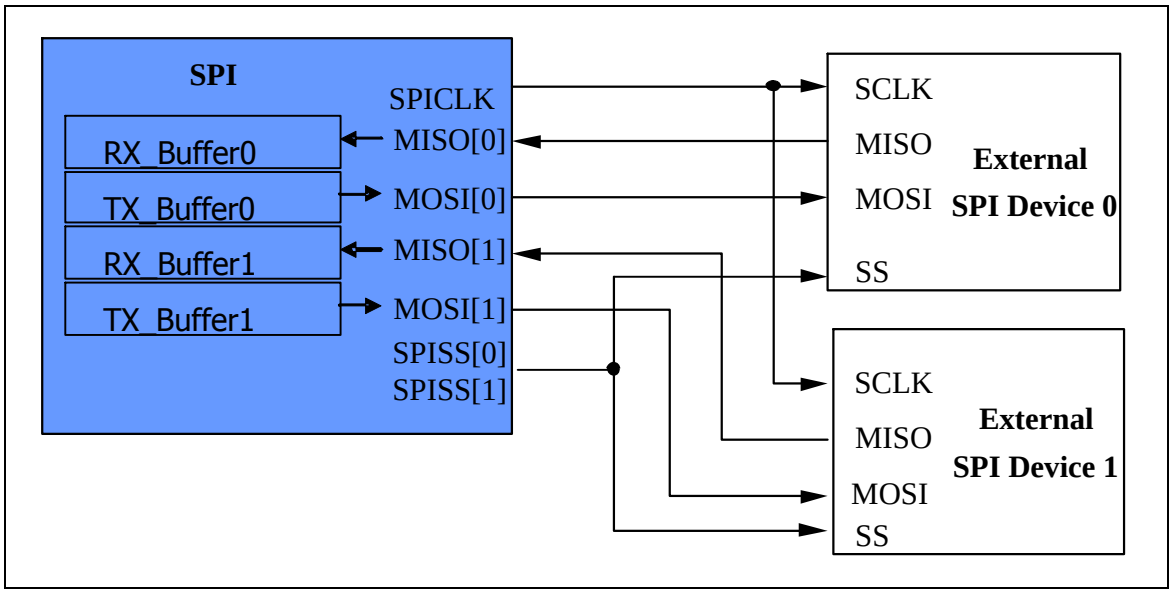



Figure 5-98 SPI 2-bit Transfer Mode

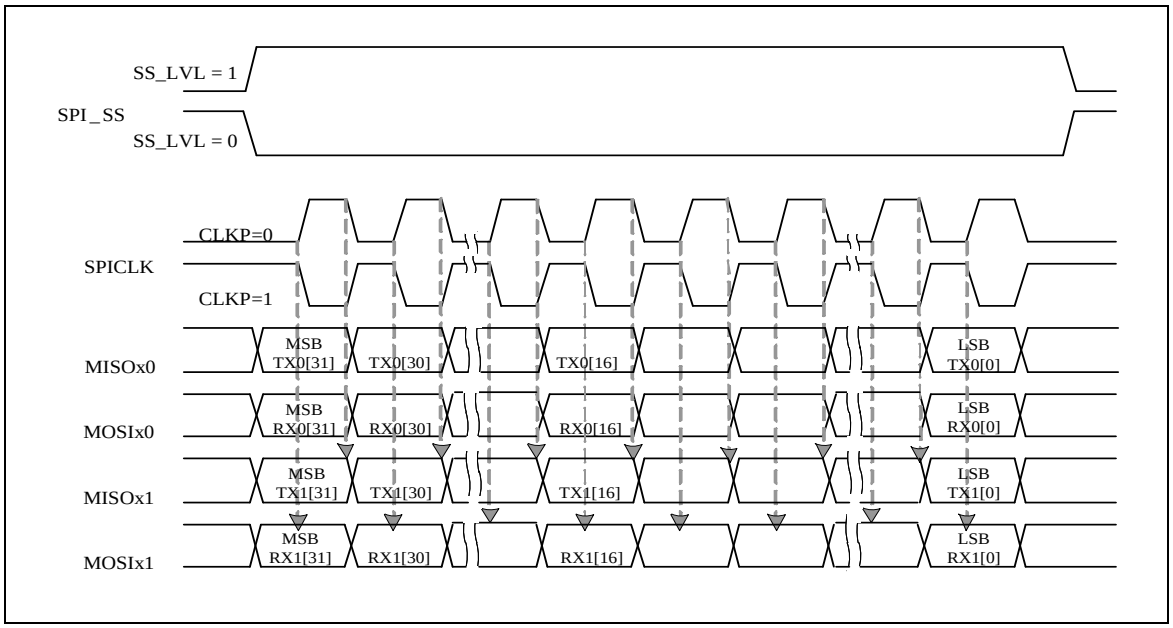



Figure 5-99 SPI 2-bit Transfer Mode Timing Diagram

5.17.4.17 Dual IO Mode

This SPI controller also supports dual IO transfer when set the DUAL_IO_EN bit (SPI_CTL[29]) to 1. The DUAL_IO_DIR bit (SP)_CTL2[28] is used to define the direction of the transfer data. When set the DUAL_IO_DIR bit to 1, the controller will send the data to external device. When the DUAL_IO_DIR bit set 0, the controller will read the data from the external device. This function supports 8, 16, 24, and 32-bits of bit length.

The dual IO mode is not supported when the 3-wire mode or the byte reorder function is enabled.

If both the DUAL_IO_EN and DUAL_IO_DIR bits are set as 1, the MOSI0 is the even bit data output and the MISO0 will be set as the odd bit data output. If the DUAL_IO_EN is set as 1 and DUAL_IO_DIR is set as 0, both the MISO0 and MOSI0 will be set as data input ports.

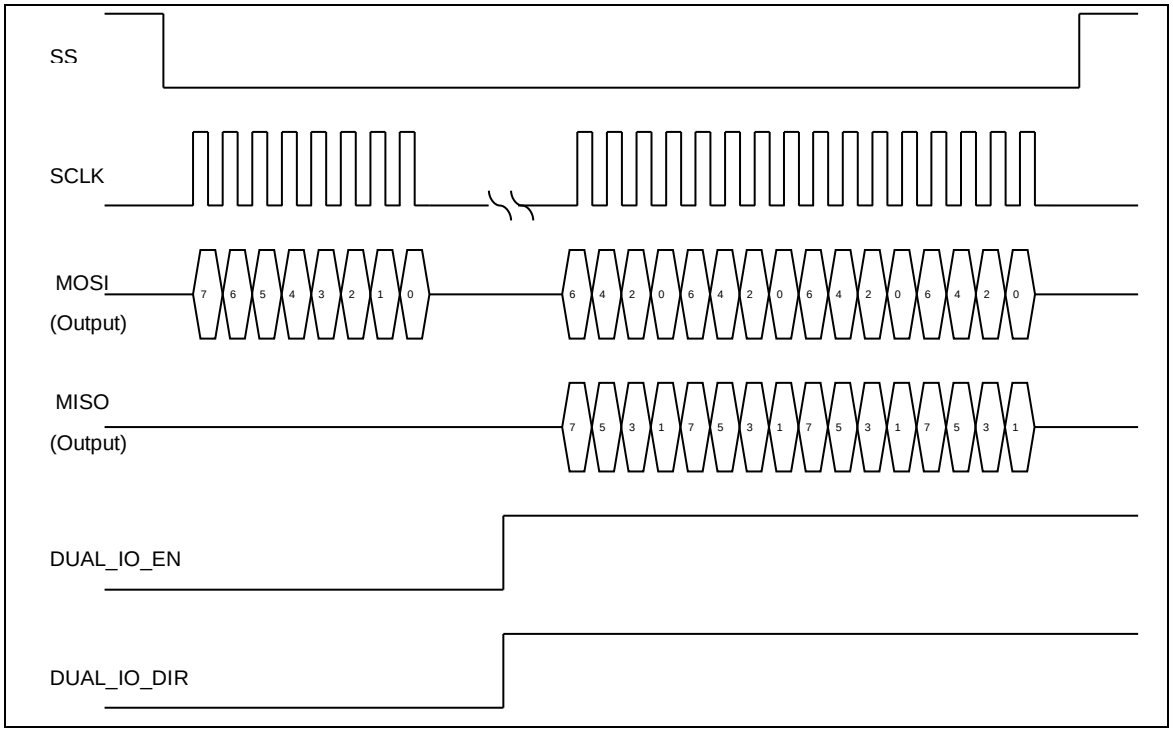



Figure 5-100 SPI DUAL-I/O output Sequence

**Note:** The byte reorder and no slave select functions are not support in DUAL I/O mode.

5.17.4.18 Time Out

In FIFO mode, there is time-out function to inform user. If there is a received data in the FIFO and it isn't read by user over 64 SPI engine clock in master mode or over 514 SPI engine clock in slave mode, it will send a time-out interrupt to the system if the time-out enable bit, FIFO_CTL[7], is set to 1

5.17.4.19 FIFO Mode

The SPI controller supports FIFO mode when the FIFO bit, SPI_CNTRL[21], is set as 1. The SPI controllers equip with 8 32-bit wide transmit and receive FIFO buffers.

The transmit FIFO buffer is an 8-level depth, 32-bit wide, first-in, first-out register buffer. The software can write data to the transmit FIFO buffer by writing the SPI_TX0 register. The data stored in the transmit FIFO buffer will be read and sent out by the transmission control logic. If the 8-level transmit FIFO buffer is full, the TX_FULL bit will be set to 1. When the SPI transmission logic unit draws out the last datum of the transmit FIFO buffer, so that the 8-level transmit FIFO buffer is empty, the TX_EMPTY bit will be set to 1. Notice that the TX_EMPTY flag is set to 1 while the last transaction is still in progress. In master mode, the software should check both the GO_BUSY bit and TX_EMPTY bit to make sure whether the SPI is in idle or not.

The received FIFO buffer is also an 8-level depth, 32-bit wide, first-in, first-out register buffer. The receive control logic will store the received data to this buffer. The software can read the FIFO buffer data from SPI_RX0 register. There are FIFO related status bits, like RX_EMPTY and RX_FULL, to indicate the current status of FIFO buffer.

In FIFO mode, the software can set the transmitting and receiving threshold by setting the TX_THRESHOLD and RX_THRESHOLD settings. When the count of valid data stored in transmit FIFO buffer is less than or equal to TX_THRESHOLD setting, the TX_INTSTS bit will be set to 1. When the count of valid data stored in receive FIFO buffer is larger than RX_THRESHOLD

setting, the RX_INTSTS bit will be set to 1.

In FIFO mode, the software can write 8 data to the SPI transmit FIFO buffer in advance. When the SPI controller operates in FIFO mode, the GO_BUSY bit of SPI_CNTRL register will be controlled by hardware, software should not modify the content of SPI_CNTRL register unless clearing the FIFO bit to disable the FIFO mode.

In Master mode transmission operation, the TX_EMPTY flag will be cleared to 0 when the FIFO bit is set to 1 and the software write the first datum to the SPI_TX0 register. The transmission starts immediately as long as the transmit FIFO buffer is not empty. User can write the next data into SPI_TX0 register immediately. The SPI controller will insert a suspend interval between two successive transactions in FIFO mode and the period of suspend interval is decided by the setting of SP_CYCLE (SPI_CNTRL [15:12]). User can write data into SPI_TX0 register as long as the TX_FULL flag is 0.

The subsequent transactions will be triggered automatically if the transmitted data are updated in time. If the SPI_TX0 register does not be updated after all data transfer are done, the transfer will stop.

In Master mode reception operation, the serial data are received from MISOx pin and stored to receive FIFO buffer. The RX_EMPTY flag will be cleared to 0 while the receive FIFO buffer contain unread data. The software can read the received data from SPI_RX0 register as long as the RX_EMPTY flag is 0. If the receive FIFO buffer contains 8 unread data, the RX_FULL flag will be set to 1. The SPI controller will stop receiving data until the software read the SPI_RX0 register.

In Slave mode, when the FIFO bit is set as 1, the GO_BUSY bit will be set as 1 by hardware automatically. If user wants to stop the slave mode SPI data transfer, both the FIFO bit and GO_BUSY bit must be cleared to 0 by software.

In Slave mode transmission operation, when the software writes data to SPI_TX0 register, the data will be loaded into transmit FIFO buffer and the TX_EMPTY flag will be set to 0. The transmission will start when the slave device receives clock signal from master. The software can write data to SPI_TX0 register as long as TX_FULL flag is 0. After all data have been drawn out by the SPI transmission logic unit and the software does not update the SPI_TX0 register, the TX_EMPTY flag will be set to 1.

In Slave mode reception operation, the serial data is received from MOSIx pin and stored to SPI_RX0 register. The reception mechanism is similar to master mode receiving operation.

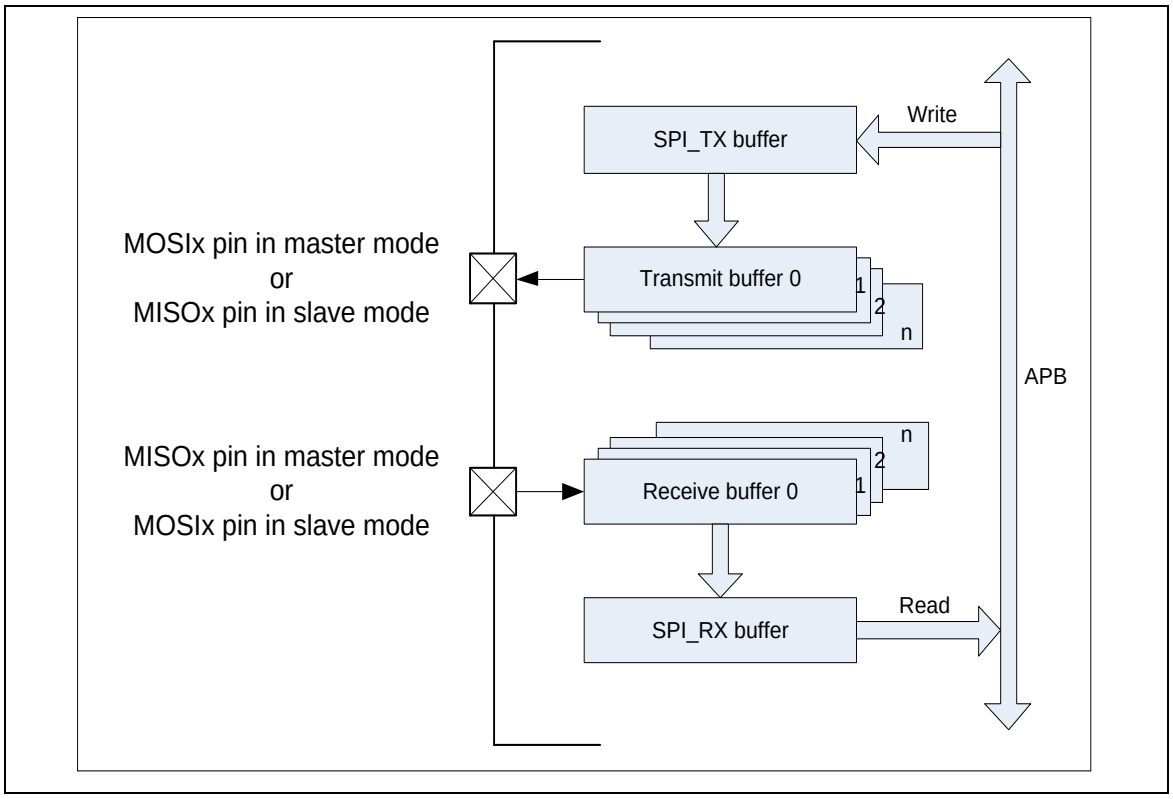



Figure 5-101 SPI FIFO Mode Control Timing

5.17.4.20 Wake-Up

There is wake-up function in the SPI controller. When the system enter power-down mode by user, it can be waked up by receive a toggle signal of SPICLK from the external device.

5.17.4.21 SPI Programming

In Master/Slave mode, the active level of device/slave select (SPI_SS) signal can be programmed to low active or high active in SS_LVL bit (SPI_SSR[2]), but the SPISS0/1 is level trigger or edge trigger which is defined in SS_LTRIG bit (SPI_SSR[4]). The serial clock (SPICLK) idle state can be configured as high state or low state by setting the CLKP bit (SPI_CTL[11]). It also provides the bit length of a transaction in TX_BIT_LEN (SPI_CTL[7:3]), and transmit/receive data from MSB or LSB first in LSB bit (SPI_CTL[10]). Users also can select which edge of serial clock to transmit/receive data in TX_NEG/RX_NEG (SPI_CTL[2:1]) . Four SPI timing diagrams for Master/Slave operations and the related settings are shown below.

**Note:** Tx_NEG = TX_NEG; Rx_NEG = RX_NEG and Tx_BIT_LEN = TX_BIT_LEN

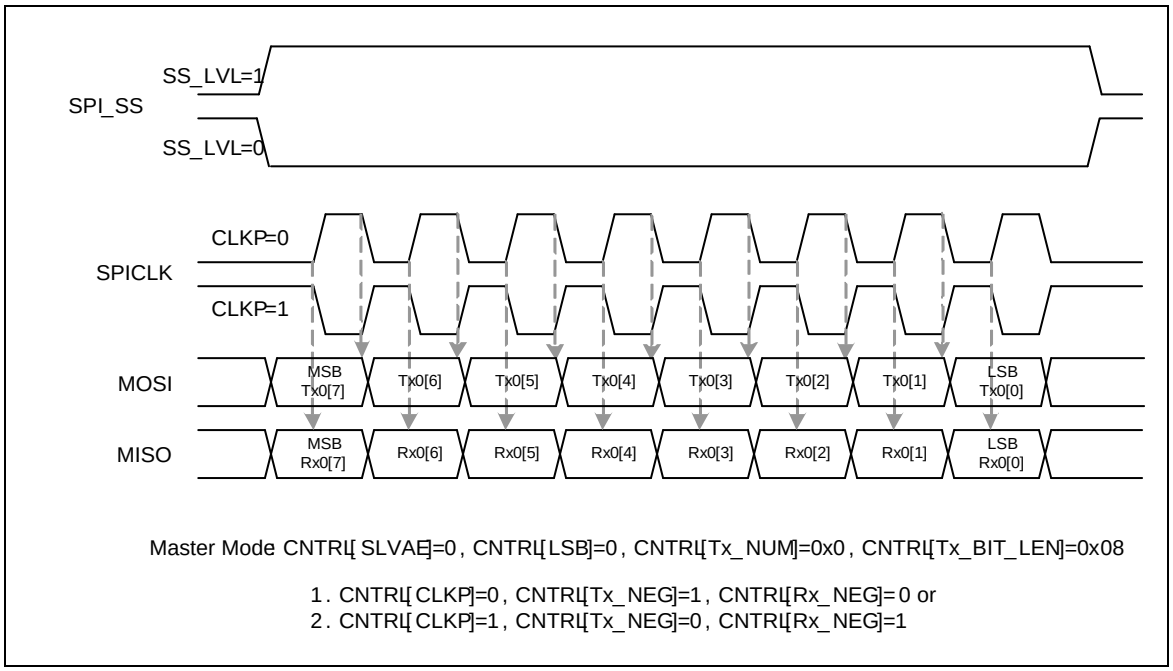



Figure 5-102 SPI Timing in Master Mode

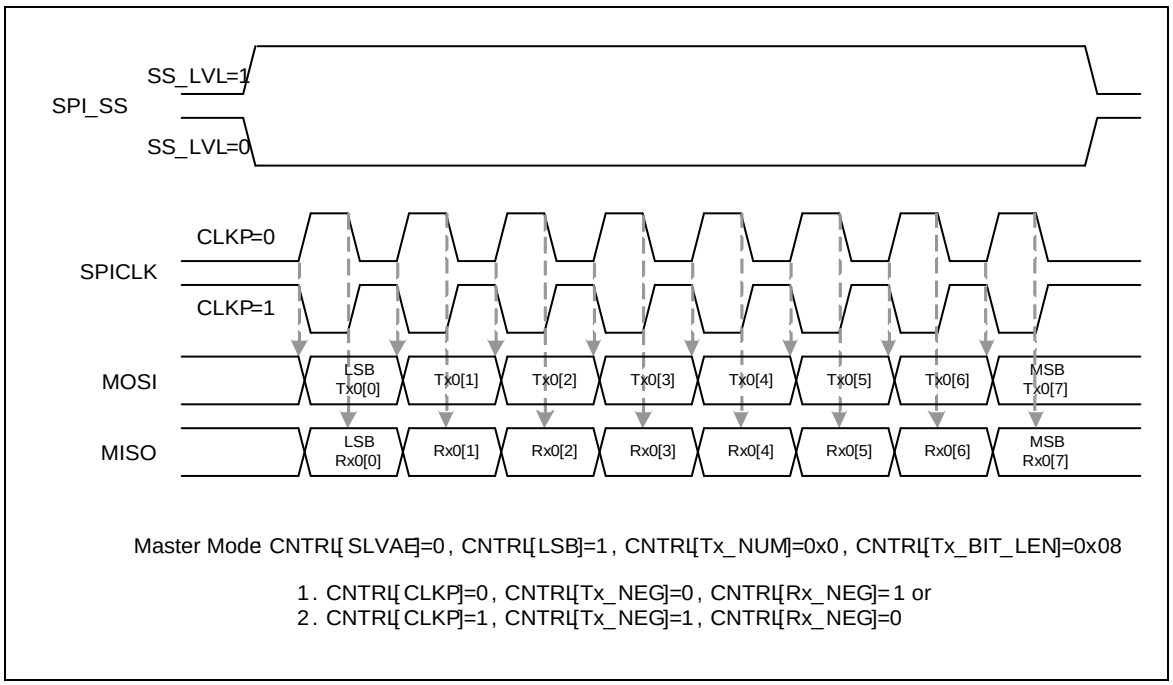



Figure 5-103 SPI Timing in Master Mode (Alternate Phase of SPICLK & LSB = 1)

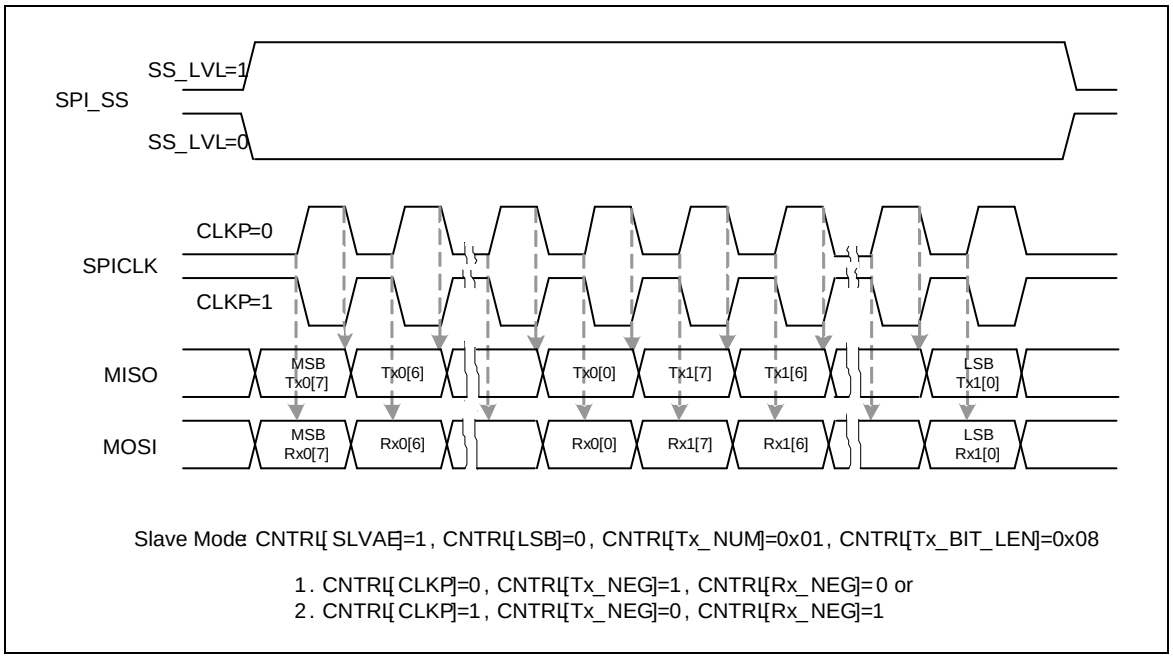



Figure 5-104 SPI Timing in Master Mode (Alternate Phase of SPICLK & LSB = 0)

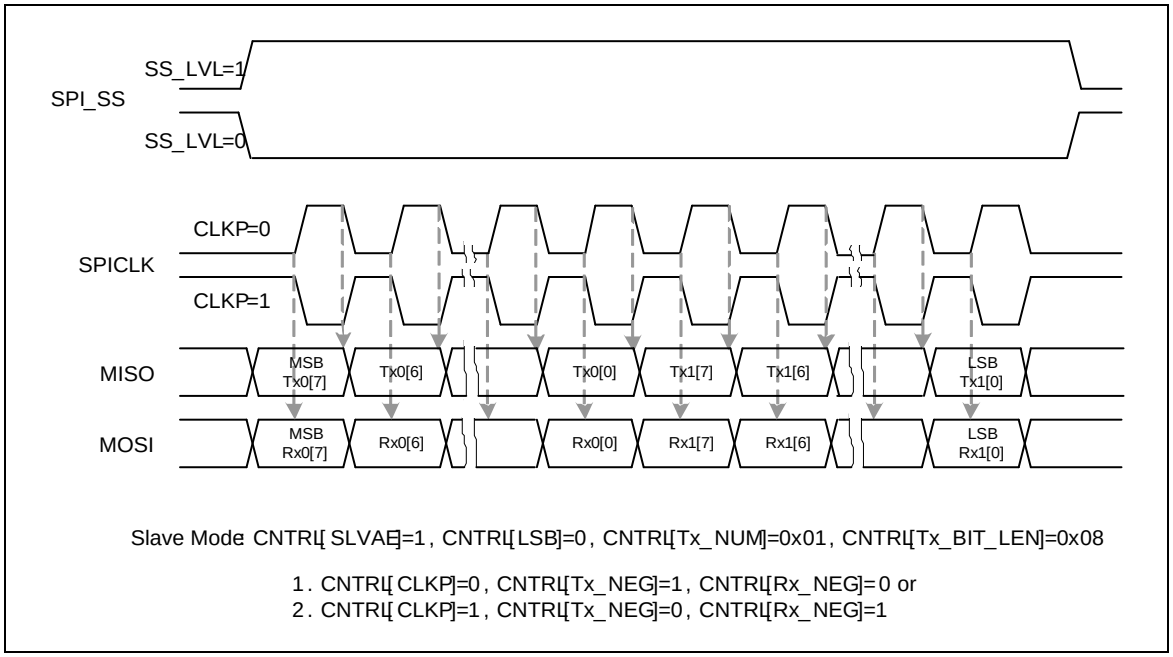



Figure 5-105 SPI Timing in Slave Mode

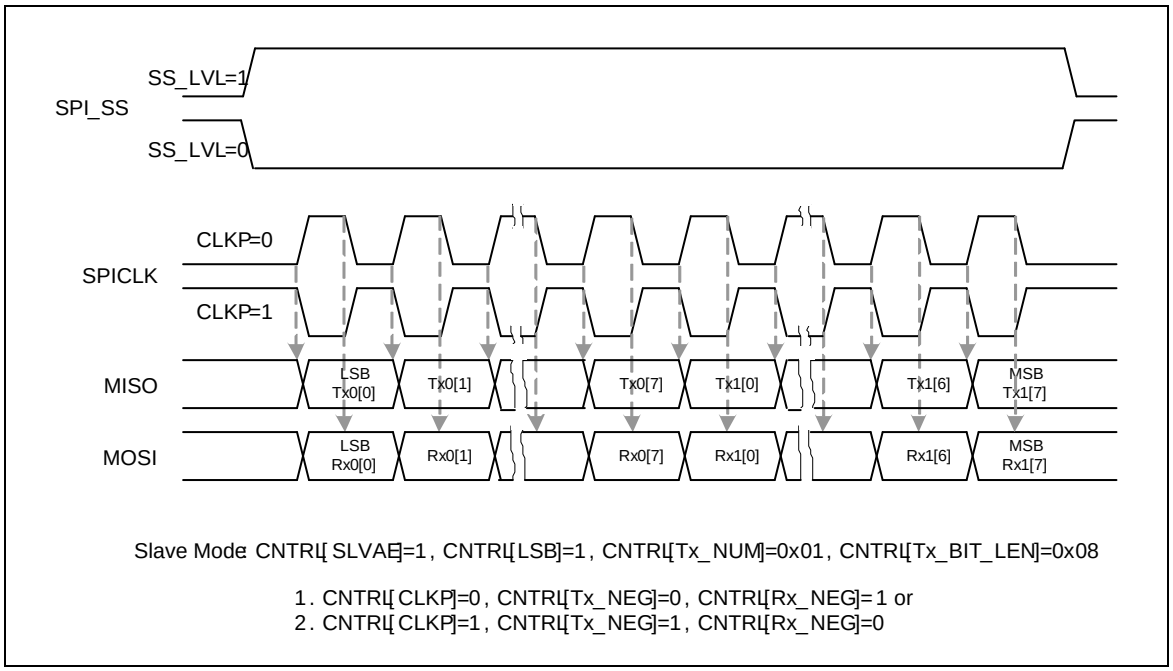



Figure 5-106 SPI Timing in Slave Mode (Alternate Phase of SPICLK)

5.17.4.22 SPI Programming Example

Example 1, SPI controller is set as a master to access an off-chip slave device with following specifications:

- Data bit is received on positive edge of serial clock
- Data bit is transmitted on negative edge of serial clock
- Data bit is transferred from MSB first
- SPICLK is idle at low state
- Only one byte of data to be transmitted/received in a transfer
- Slave select signal is active low

Basically, the specification of the connected off-chip slave device should be referred in details before the following steps:

1. Set the DIVIDER1 (SPI_CLKDIV[7:0]) register to determine the output frequency of serial clock.
2. Write the SPI_SSR register a proper value for the related settings of Master mode:

   (a). to disable the Automatic Slave Select bit AUTOSS (SPI_SSR[3] = 0), (b). Select low level trigger output of slave select signal in the Slave Select Active Level bit SS_LVL (SPI_SSR[2] = 0) and Slave Select Level Trigger bit SS_LTRIG. (c). Select slave select signal to be output at the I/O pin by setting the respective Slave Select Register bits SSR[0] or SSR[1] (SPI_SSR[1:0]) to active the off-chip slave devices.
3. Write the related settings into the SPI_CTL register (1). To control this SPI controller as master device in SLAVE bit (SPI_CTL[18] = 0). (2). Force the serial clock idle state at low in CLKP bit (SPI_CTL[11] = 0), (3). Select data transmitted at negative edge of serial clock in TX_NEG bit (SPI_CTL[2] = 1). (4). Select data received at positive edge of serial clock in RX_NEG bit (SPI_CTL[1] = 0). (5). Set the bit length of transaction as 8 in TX_BIT_LEN bit field (SPI_CTL[7:3] = 0x08). (6). Set LSB transfer first in LSB bit (SPI_CTL[10] = 1).

4. If this SPI master attempts to transmit (write) one byte data to the off-chip slave device, write the byte data that will be transmitted into the SPI_TX0 register.
5. If this SPI master just only attempts to receive (read) one byte data from the off-chip slave device and does not care what data will be transmitted, the software does not need to update the SPI_TX0 register.
6. Enable the GO_BUSY bit (SPI_CTL[0] = 1) to start the data transfer at the SPI interface.
7. Waiting for SPI interrupt (if the Interrupt Enable INTEN bit is set) or just polling the GO_BUSY bit till it be cleared to 0 by hardware automatically.
8. Read out the received one byte data from RDATA0[7:0) (SPI_RX0[7:0]) register.
9. Go to 4) to continue another data transfer or set SSR[0] or SSR[1] to 0 to inactivate the off-chip slave devices.

Example 2, SPI controller is set as a slave device that is controlled by an off-chip master device, and supposes the off-chip master device to access the on-chip SPI slave controller through the SPI interface with the following specifications:

- Data bit is received on positive edge of serial clock
- Data bit is transmitted on negative edge of serial clock
- Data bit is transferred from LSB first
- SPICLK is idle at high state
- Only one byte of data to be transmitted/received in a transfer
- Slave select signal is high level trigger

Basically, the specification of the connected off-chip master device should be referred in details before the following steps

1. Select high level and level trigger for the input of slave select signal in the Slave Select Active Level bit SS_LVL (SPI_SSR[2] = 1) and the Slave Select Level Trigger bit SS_LTRIG (SPI_SSR[4] = 1).
2. Write the related settings into the SPI_CTL register to control this SPI slave actions. Set this SPI controller as slave device in SLAVE bit (SPI_CTL[18] = 1). Select the serial clock idle state at high in CLKP bit (SPI_CTL[11] = 1). Select data transmitted at negative edge of serial clock in TX_NEG bit (SPI_CTL[2] = 1), Select data received at positive edge of serial clock in RX_NEG bit (SPI_CTL[1] = 0). Set the bit length of transaction as 8 bits in TX_BIT_LEN bit field (SPI_CTL[7:3] = 0x08). Set LSB transfer first in LSB bit (SPI_CTL[10] = 1), and don't care the SP_CYCLE bit field (SPI_CTL[15:12]) due to not burst mode in this case.
3. If this SPI slave attempts to transmit (be read) one byte data to the off-chip master device, write the byte data that will be transmitted into the SPI_TX0 register.
4. If this SPI slave just only attempts to receive (be written) one byte data from the off-chip master device and does not care what data will be transmitted, the software does not need to update the SPI_TX0 register.
5. Enable the GO_BUSY bit (SPI_CTL[0] = 1) to wait for the slave select trigger input and serial clock input from the off-chip master device to start the data transfer at the SPI interface.
6. Waiting for SPI interrupt (if the Interrupt Enable INTEN bit is set) or just polling the GO_BUSY bit till it be cleared to 0 by hardware automatically
7. Read out the received one byte data from (SPI_RX0[7:0])
8. Go to 3) to continue another data transfer or stop data transfer.

## 5.18 Timer Controller

### 5.18.1 Overview

This chip is equipped with four timer modules including TIMER0, TIMER1, TIMER2 and TIMER3 (TIMER0/1 is at APB1 and TIMER2/3 is at APB2), which allow user to easily implement a counting scheme or timing control for applications. The timer can perform functions like frequency measurement, event counting, interval measurement, clock generation, delay timing, and so on. The timer can generate an interrupt signal upon timeout, or provide the current value of count during operation.

### 5.18.2 Features

- Independent Clock Source for each Timer (TMRx_CLK, x= 0, 1,2,3)
- Time-out period = (Period of timer clock input) * (8-bit pre-scale counter + 1) * (24-bit TCMP)
- Counting cycle time = (1 / TMRx_CLK) * (2^8) * (2^24)
- Internal 8-bit pre-scale counter
- Internal 24-bit up counter is readable through TDR (Timer Data Register)
- Supports One-shot, Periodic,Output Toggle and Countinuous Counting Operation mode
- Supports external pin capture for interval measurement
- Supports external pin capture for timer counter reset
- Supports Inter-Timer trigger
- Supports Internal trigger event to ADC, DAC and PDMA

### 5.18.3 Block Diagram

Each timer is equipped with an 8-bit pre-scale counter, a 24-bit up-counter, a 24-bit compare register and an interrupt request signal. Refer to Figure 5-107 for the timer controller block diagram. There are five options of clock sources for each timer, Figure 5-108 illustrate the Clock Source control function.

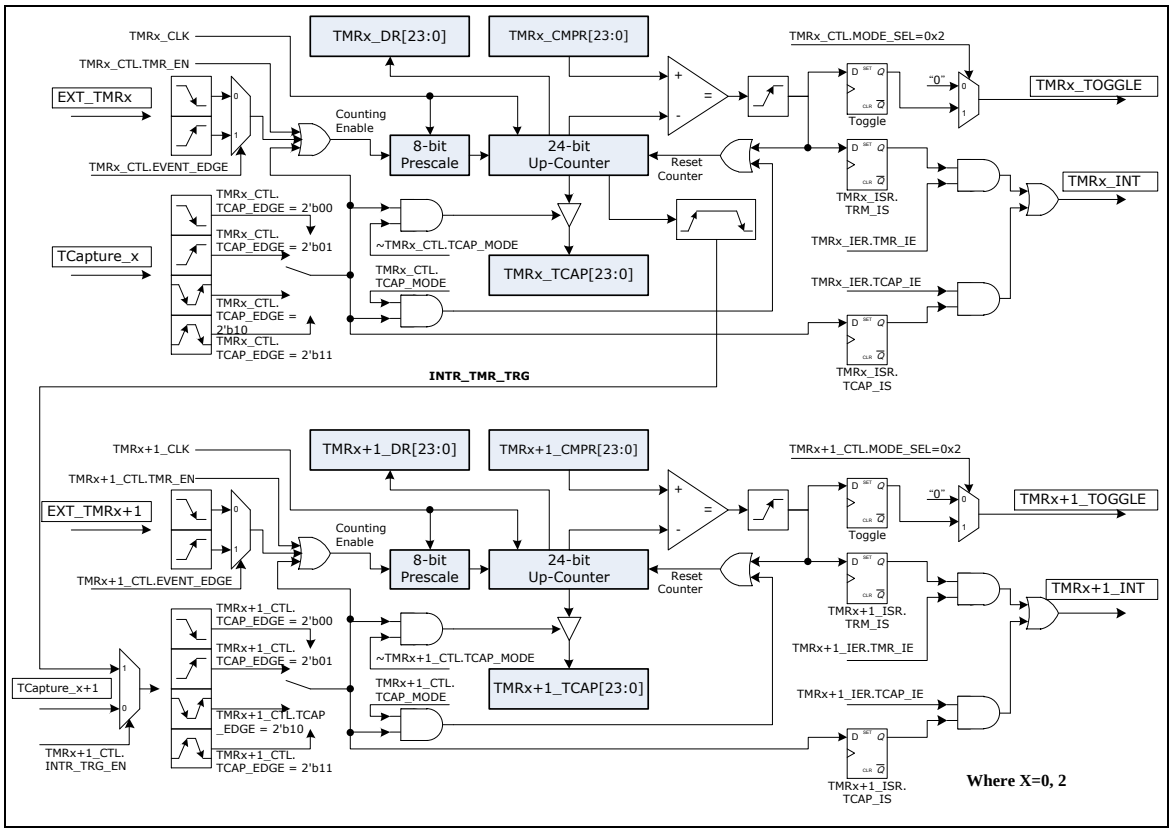



Figure 5-107 Timer Controller Block Diagram

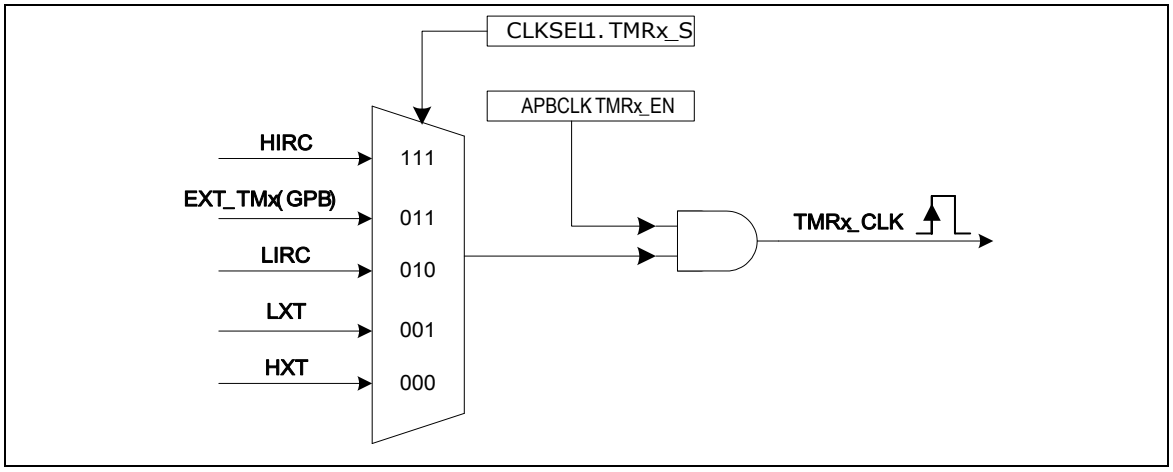



Figure 5-108 Timer Clock Controller Diagram

### 5.18.4 Functional Description

Timer controller provides One-shot, Period, Toggle and Continuous Counting operation modes. The event counting function is also provided to count the events/counts from external pin and external pin capture function for interval measurement or reset timer counter. In addition, timer controller provides the Inter-Timer Trigger Mode to measure input frequency precisely. Each operating function mode is shown as follows:

#### 5.18.4.1 One-Shot Mode

If the timer is operated in One-shot mode (MODE_SEL[1:0] is 0x0) and TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1, the timer counter starts up counting. Once the timer counter

value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, the TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 1, and TMR_IS (TMRx_ISR[0] timer interrupt status) is 1 then the interrupt signal is generated and sent to NVIC to inform CPU for indicating that the timer counting overflow happens. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 0, no interrupt signal is generated.

In this operating mode, once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1, timer counting operation stops and the timer counter value (TMRx_DR value) goes back to counting initial value then TMR_EN (TMRx_CTL[0] timer counter enable bit) is cleared to 0 by timer controller automatically. That is to say, timer operates timer counting and compares with TMRx_CMPR value function only one time after programming the timer compare register (TMRx_CMPR) value and TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1. So, this operating mode is called One-Shot mode.

#### 5.18.4.2 Periodic Mode

If the timer is operated in Period mode (MODE_SEL[1:0] is 0x1) and TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1, the timer counter starts up counting. Once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, the TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 1, and TMR_IS (TMRx_ISR[0] timer interrupt status) is 1 then the interrupt signal is generated and sent to NVIC to inform CPU for indicating that the timer counting overflow happens. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 0, no interrupt signal is generated.

In this operating mode, once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1, the timer counter value (TMRx_DR value) goes back to counting initial value and TMR_EN (TMRx_CTL[0] timer counter enable bit) is kept at 1 (counting enable continuously) and timer counter operates up counting again. If TMR_IS (TMRx_ISR[0] timer interrupt status) is cleared by software, once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value again, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 also. That is to say, timer operates timer counting and compares with TMRx_CMPR value function periodically. The timer counting operation does not stop until the TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 0. The interrupt signal is also generated periodically. So, this operating mode is called Periodic mode.

#### 5.18.4.3 Toggle Mode

If the timer is operated in Toggle mode (MODE_SEL[1:0] is 0x2) and TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1, the timer counter starts up counting. Once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, the TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 1, and TMR_IS (TMRx_ISR[0] timer interrupt status) is 1 then the interrupt signal is generated and sent to NVIC to inform CPU for indicating that the timer counting overflow happens. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 0, no interrupt signal is generated.

In this operating mode, once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1, toggle out signal is set to 1, the timer counter value (TMRx_DR value) goes back to counting initial value and TMR_EN (TMRx_CTL[0] timer counter enable bit) is kept at 1 (counting enable continuously) and timer counter operates up counting again. If TMR_IS (TMRx_ISR[0] timer interrupt status) is cleared by software, once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value again, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 also and toggle out signal is set to 0. The timer counting operation does not stop until the TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 0. Thus, the toggle output signal is

changing back and forth with 50% duty cycle. So, this operating mode is called Toggle mode.

#### 5.18.4.4 Continuous Counting Mode

If the timer is operated in Continuous Counting mode (MODE_SEL[1:0] is 0x3) and TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1, the timer counter starts up counting. Once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, the TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1. If TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1, and TMR_IS (TMRx_ISR[0] timer interrupt status) is 1 then the interrupt signal is generated and sent to NVIC to inform CPU for indicating that the timer counting overflow happens. If TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 0, no interrupt signal is generated.

In this operating mode, once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 and TMR_EN (TMRx_CTL[0] timer counter enable bit) is kept at 1 (counting enable continuously) and timer counter continuous counting without reload the timer counter value (TMRx_DR value) to counting initial value. User can change different timer compare register (TMRx_CMPR) value immediately without disabling timer counter and restarting timer counter counting.

For example, the timer compare register (TMRx_CMPR) value is set as 80, first. (The timer compare register (TMRx_CMPR) should be less than 2^24 and be greater than 1). Once the timer counter value (TMRx_DR value) reaches to 80, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 and TMR_EN (TMRx_CTL[0] timer counter enable bit) is kept at 1 (counting enable continuously) and timer counter value (TMRx_DR value) will not goes back to 0, it continues counting to 81, 82, 83,⋯ to (2^24 -1) then 0, 1, 2, 3, ⋯ to 2^24 -1 again and again. Next, if user programs timer compare register (TMRx_CMPR) value as 200 and the TMR_IS (TMRx_ISR[0] timer interrupt status) is cleared to 0, then TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 again when timer counter value (TMRx_DR value) reaches to 200. At last, user programs timer compare register (TMRx_CMPR) value as 500 and clears TMR_IS (TMRx_ISR[0] timer interrupt status) to 0, then TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 again when timer counter value (TMRx_DR value) reaches to 500. In this mode, the timer counter value (TMRx_DR value) is keeping up counting always even if TMR_IS (TMRx_ISR[0] timer interrupt status) is 1. So, this operation mode is called as Continuous Counting mode.

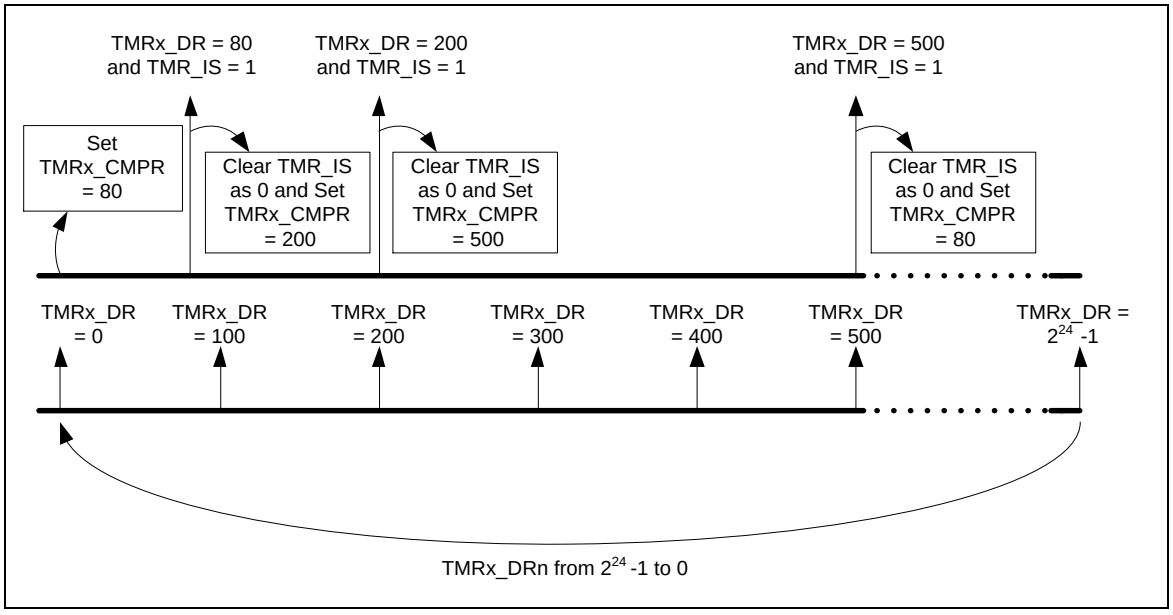



Figure 5-109 Timer Clock Controller Diagram

#### 5.18.4.5 Event Counting Mode

An application which can count the events/counts from external event input pin is called as event

counting function. In this mode, most of the timer control registers are the same with the timer operating function mode except EVENT_EN (TMRx_CTL[12] event counting mode enable) has to set to 1. When status transition on external event input pin, the event counter value (TMRx_DR value) will be counted according to EVENT_EDGE (TMRx_CTL[13] event counting mode edge selection) setting. EVNT_DEB_EN (TMRx_CTL[14] external event de-bounce enable) bit is for enabled or disabled edge detection de-bounce circuit of external event input pin. The max frequency of event counting source on external event input pin should be less than 1/4 TRMx_CLK if EVNT_DEB_EN (TMRx_CTL[14] external event de-bounce enable) is 0 or less than 1/10 TRMx_CLK if EVNT_DEB_EN (TMRx_CTL[14] external event de-bounce enable) is 1. Otherwise, the event counter value (TMRx_DR value) will not be counted normally.

#### 5.18.4.6 Timer Counter Capture/Reset Function

In this mode, Timer will monitor the transition of external pin to save the 24-bit counter value or reset the 24-bit counter.

If TCAP_MODE is 0, the transition on external pin is used as timer counter capture function. In this mode, if CAP_CNT_MOD is 0, the free-counting mode, 24-bit up-counting timer will keep counting continuously. And when the transition of external pin matches the TCAP_EDGE setting, the value of 24-bit up-counting timer will be saved into register TMRx_TCAP. If CAP_CNT_MOD is 1, the trigger-counting mode, 24-bit up-counting timer will keep its value at zero. Once the transition of external pin matches the 1st transition of TCAP_EDGE setting, the 24-bit up-counting timer will start counting. And then if the transition of external pin matches the 2nd transition of TCAP_EDGE setting, the 24-bit up-counting timer will stop counting. And its value will be saved into register TMRx_TCAP.

If TCAP_MODE is 1, the transition on external pin is used as timer counter reset function. In this mode, once the transition of external pin matches the TCAP_EDGE setting, the 24-bit up-counting timer will be reset.

To detect the transition of external pin, the timer circuit implements the de-bounce for external pin. Based on the result of de-bounce circuit and external pin, the rising-edge or falling-edge could be detected. The reset value of de-bounce circuit is “0” and the de-bounce would only active when both TCAP_DEB_EN and TCAP_EN are enabled. So, if the external pin level is “1” and TCAP_EDGE is set to detect rising-edge of external pin, then after de-bounce circuit active (TCAP_DEB_EN is “1” and TCAP_EN is “1”), a false rising-edge would be detected. This would result in the incorrect capture data (TMRx_TCAP) while 1st time the TCAP_IS is set. To avoid this incorrect capture data to affect the capture application, discard this 1st capture data is necessary and recommended.

#### 5.18.4.7 Inter-Timer Trigger Mode

In this mode, the TMRx (where x=0 or 2), will be forced in counter mode, counting with external event, and will generate a signal (INTR_TMR_TRG) to trigger TMRx+1 (where x=0 or 2). The TMRx+1 will be forced in trigger-counting mode of capture function.

While INTR_TRG_EN is set, the TMRx will make a rising-edge transition of trigger signal (INTR_TMR_TRG) to TMRx+1 while 24-bit counter is counting from 0x0 to 0x1. And when 24-bit counter reaches the 24-bit TCMPR value, the TMRx will make a falling-edge transition of trigger signal (INTR_TMR_TRG) to TMRx+1.

When INTR_TMR_TRG transited from low to high (rising-edge), the 24-bit counter of TMRx+1 will start to count. Also, when INTR_TMR_TRG transited from high to low (falling-edge), the 24-bit counter of TMRx+1 will stop counting. At the same time, the value of 24-bit counter will be saved into TMRx+1_TCAP.

The example listed below is to show how inter-timer trigger mode work. When inter-timer trigger mode is enabled (TMRx_CTL.INTR_TRG_EN = 1), the TMRx_Counter starts to up-counting in each external event (TMRx) detected. When TRMx_Counter counted from 0x0 to 0x1, the INTR_TMR_TRG is set high, and TMRx+1_Counter is then start counting in each TMRx+1_CLK.

When TMRx_Counter reaches the value of TMRx_CMPR, it stops counting and ITNR_TMR_TRG is set low. And then, the TMRx+1_Counter stops counting, too. In the same time, the value of TMRx+1_Counter is sampled into TMRx+1_TCAP and interrupt status TMRx_ISR.TMR_IS is also set. In addition, the TRMx_CTL.INTR_TRG_EN is also automatically cleared by H/W.

By using Inter-timer Trigger mode, the period of external event (TMRx) could be measured more precisely.

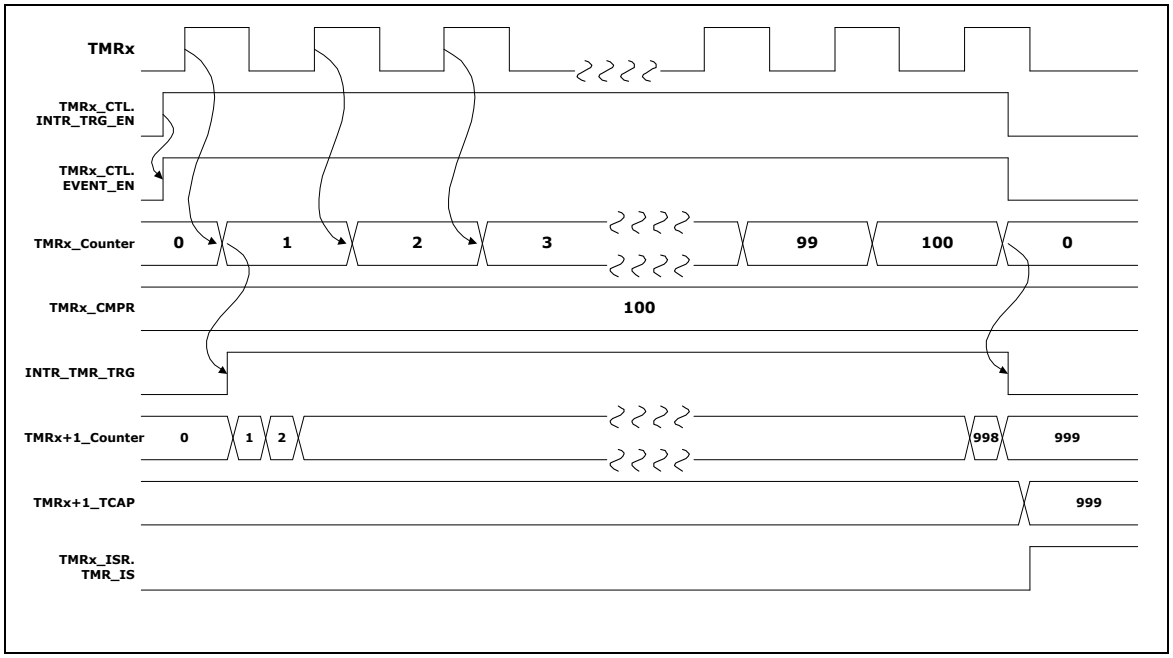



Figure 5-110 Inter-Timer Trigger Mode

## 5.19 UART Controller

### 5.19.1 Overview

The UART controllers provides up to two channels of Universal Asynchronous Receiver/Transmitter (UART) modules that are UART0 and UART1. (UART0 is at APB1 and UART1 is at APB2).

The Universal Asynchronous Receiver/Transmitter (UART) performs a serial-to-parallel conversion on data received from the peripheral, and a parallel-to-serial conversion on data transmitted from the CPU. The UART controller also supports IrDA (SIR) function mode, LIN Master/Slave function mode and RS-485 function mode. Each UART channel supports nine types of interrupts including receiver threshold level reaching interrupt (INT_RDA), transmitter FIFO empty interrupt (INT_THRE), line status interrupt (break error, parity error, framing error or RS-485 interrupt) (INT_RLS), time-out interrupt (INT_TOUT), MODEM status interrupt (INT_MODEM), Buffer error interrupt (INT_BUF_ERR), wake-up interrupt (INT_WAKE), auto-baud rate detect or auto-baud rate counter overflow flag (INT_ABAUD) and LIN function interrupt (INT_LIN).

The UART0 and UART1 are built-in with a 16-byte transmitter FIFO (TX_FIFO) and a 16-byte receiver FIFO (RX_FIFO) that reduces the number of interrupts presented to the CPU. The CPU can read the status of the UART at any time during the operation. The reported status information includes the type and condition of the transfer operations being performed by the UART, as well as 3 error conditions (parity error, framing error or break interrupt) occur while receiving data. The UART controller supports auto-baud rate detection. The auto-baud rate detection controls the process of measuring the incoming clock/data rate for the baud rate generation and can be read and written at user discretion. The UART controller also support incoming data or CTSn wake-up function. When the system is in power-down mode, an incoming data or CTSn signal will wake-up CPU from power-down mode. The UART includes a programmable baud rate generator that is capable of dividing crystal clock input by divisors to produce the clock that transmitter and receiver need. The baud rate equation is Baud Rate = UART_CLK / [BRD + 1], where BRD are defined in UART Baud Rate Divider Register (UARTx_BAUD). Below table lists the equations in the various conditions and the UART baud rate setting table.

| DIV_16_EN | BRD | Baud Rate Equation |
|---|---|---|
| Disable (Mode 0) | A | UART_CLK / (A+1), A must >8 |
| Enable (Mode 1) | A | UART_CLK / [16 * (A+1)] |

Table 5-12 UART Baud Rate Equation

| System clock =12 MHz | | |
|---|---|---|
| Baud rate | Mode 0 | Mode 1 |
| 921600 | A=12 | Not Supported |
| 460800 | A=25 | Not Supported |
| 230400 | A=51 | A=2 |
| 115200 | A=103 | A=6 |
| 57600 | A=207 | A=12 |
| 38400 | A=311 | A=19 |

| | | |
|---|---|---|
| 19200 | A=624 | A=38 |
| 9600 | A=1249 | A=77 |
| 4800 | A=2499 | A=155 |

Table 5-13 UART Baud Rate Setting

#### 5.19.1.1 Auto-Flow Control

The UART0 and UART1 controllers support auto-flow control function that uses two low-level signals, CTSn (clear-to-send) and RTSn (request-to-send) to control the flow of data transfer between the UART and external devices (ex: Modem). When auto-flow is enabled, the UART is not allowed to receive data until the UART asserts RTSn (RTSn high) to external device. When the number of bytes in the RX-FIFO equals the value of UART_TLCTL [RTS_TRI_LEV], the RTSn is de-asserted. The UART sends data out when UART controller detects CTSn is asserted (CTSn high) from external device. If a valid asserted CTSn is not detected the UART controller will not send data out.

#### 5.19.1.2 Auto-Baud Rate Detection

The UART0 and UART1 controllers support auto-baud rate detection. The auto-baud rate function can be used to measure the receiver incoming data baud rate. If enabled the auto-baud feature, UART controller will measure the bit time of the received data stream and set the divisor latch registers UART_BARD. Auto-baud rate detection is started by setting the UART_CTL [ABAUD_EN].

#### 5.19.1.3 UART Wake-Up Function

The UART0 and UART1 controllers support wake-up system function. The wake-up function includes CTSn wake-up function (UART_CTL [WAKE_CTS_EN]) and data wake-up function (UART_CTL [WAKE_DATA_EN]). When the system is operation in power-down mode, the UART can wake-up system by CTSn pin or by incoming data.

#### 5.19.1.4 IrDA Function Mode

The UART controllers also provides Serial IrDA (SIR, Serial Infrared) function (User must set UART_FUN_SEL to select IrDA function). The SIR specification defines a short-range infrared asynchronous serial transmission mode with one start bit, 8 data bits, and 1 stop bit. The maximum data rate is 115.2 Kbps (half duplex). The IrDA SIR block contains an IrDA SIR Protocol encoder/decoder. The IrDA SIR protocol is half-duplex only. So it cannot transmit and receive data at the same time. The IrDA SIR physical layer specifies a minimum 10 ms transfer delay between transmission and reception, and in IrDA Operation mode the UART_BAUD setting must be mode1 (UART_BAUD [DIV_16_EN] = “1”).

#### 5.19.1.5 RS-485 Function Mode

Another alternate function of UART controllers is RS-485 9 bit mode function whose direction control can be controlled by RTSn pin or GPIO. The RS-485 function mode is selected by setting the UART_FUN_SEL register to select RS-485 function. The RS-485 driver control is implemented by using the RTSn control signal from an asynchronous serial port to enable the RS-485 driver. In RS-485 mode, many characteristics of the RX and TX are same as UART.

#### 5.19.1.6 LIN Function Mode

The LIN mode is selected by setting the LIN_EN bit in UART_FUN_SEL register. In LIN mode, one start bit and 8-bit data format with 1-bit stop bit are required in accordance with the LIN standard.

### 5.19.2 Features

- Full duplex, asynchronous communications.
- Separate receiving / transmitting 16 bytes entry FIFO for data payloads.
- Supports hardware auto-flow control/flow control function (CTSn, RTSn) and programmable (CTSn, RTSn) flow control trigger level.
- Supports programmable baud rate generator for each channel.
- Supports auto-baud rate detect function.
- Supports programmable receiver buffer trigger level.
- Supports incoming data or CTSn to wake-up function.
- Supports 9 bit receiver buffer time-out detection function.
- All UART channels can be served by the PDMA controller.
- Programmable transmitting data delay time between the last stop bit leaving the TX-FIFO and the de-assertion by setting UART_TMCTL [DLY] register.
- Supports break error, frame error, parity error and receiving / transmitting buffer overflow detect function.
- Fully programmable serial-interface characteristics:
  - Programmable number of data bit, 5, 6, 7, 8 character.
  - Programmable parity bit, even, odd, no parity or stick parity bit generation and detection.
  - Programmable stop bit, 1, 1.5, or 2 stop bit generation.
- Supports IrDA SIR function mode
  - Supports 3/16 bit period modulation.
- Supports LIN function mode.
  - Supports LIN Master/Slave mode
  - Supports programmable break generation function for transmitter.
  - Supports break detect function for receiver.
- Supports RS-485 function mode.
  - Supports RS-485 9bit mode.
  - Supports hardware or software controls RTSn or software control GPIO to control transfer direction.

### 5.19.3 Block Diagram

The UART clock control and block diagram are shown as follows. The UART controller is completely asynchronous design with two clock domains, PCLK and engine clock. Note that the PCLK should be higher than or equal to the frequency of engine clock.

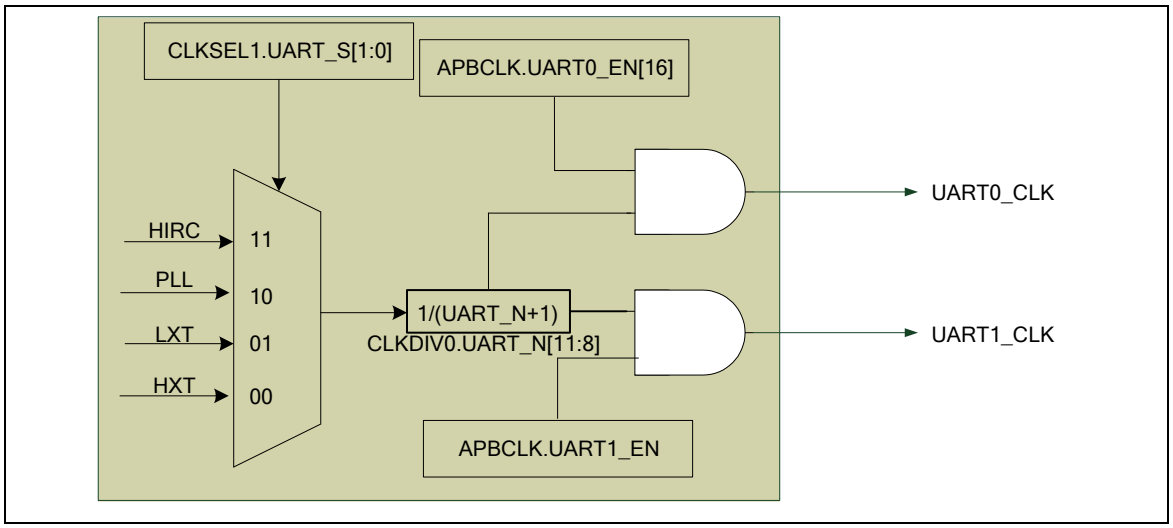



Figure 5-111 UART Clock Control Diagram

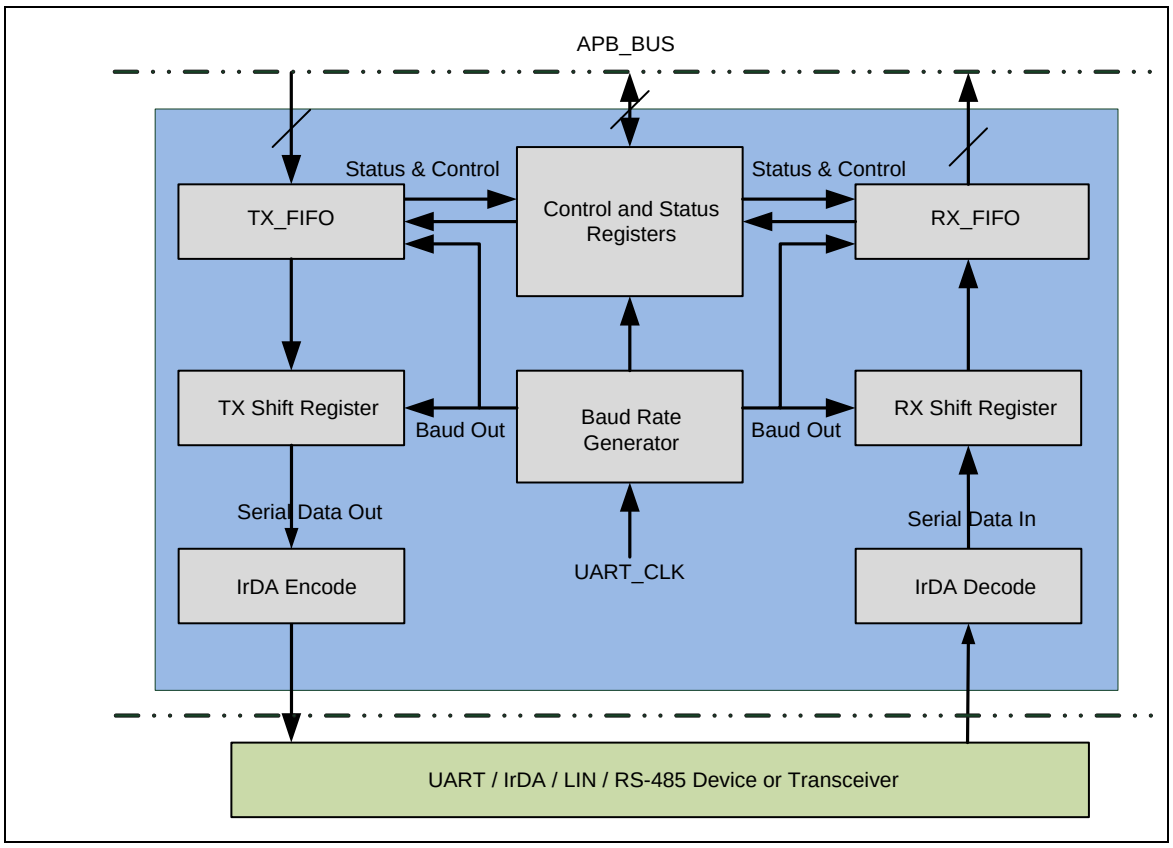



Figure 5-112 UART Block Diagram

**TX_FIFO**

The transmitter buffered is a 16 byte FIFO to reduce the number of interrupts presented to the CPU.

**RX_FIFO**

The receiver buffered is a 16 byte FIFO (plus three error bits per byte) to reduce the number of interrupts presented to the CPU.

**TX Shift Register**

The block shifts the transmitting data out serially control block.

**RX Shift Register**

The block shifts the receiving data in serially control block.

**Baud Rate Generator**

Divide the external clock or internal clock by the divisor to get the desired baud rate clock. Refer to for baud rate equation.

**IrDA Encode**

This block is the IrDA encode control block.

**IrDA Decode**

This block is the IrDA decode control block.

**Control and Status Register**

This is a register set, including the transfer line control registers (UART_TLCTL), transfer status registers (UART_TRSR), and control register (UART_CTL) for transmitter and receiver. The time-out control register (UART_TMCTL) identifies the condition of time-out interrupt. This register set also includes the interrupt enable register (UART_IER) and interrupt status register (UART_ISR) to enable or disable the responding interrupt and to identify the occurrence of the responding interrupt. There are nine types of interrupts including receiver threshold level reaching interrupt (INT_RDA), transmitter FIFO empty interrupt (INT_THRE), line status interrupt (break error, parity error, framing error or RS-485 interrupt) (INT_RLS), time-out interrupt (INT_TOUT), MODEM status interrupt (INT_MODEM), Buffer error interrupt (INT_BUF_ERR), wake-up interrupt (INT_WAKE), auto-baud rate detect or auto-baud rate counter overflow flag (INT_ABAUD) of LIN function interrupt (INT_LIN).

### 5.19.4 Functional Description

#### 5.19.4.1 Auto-Flow Control

The following diagram demonstrates the auto-flow control block diagram.

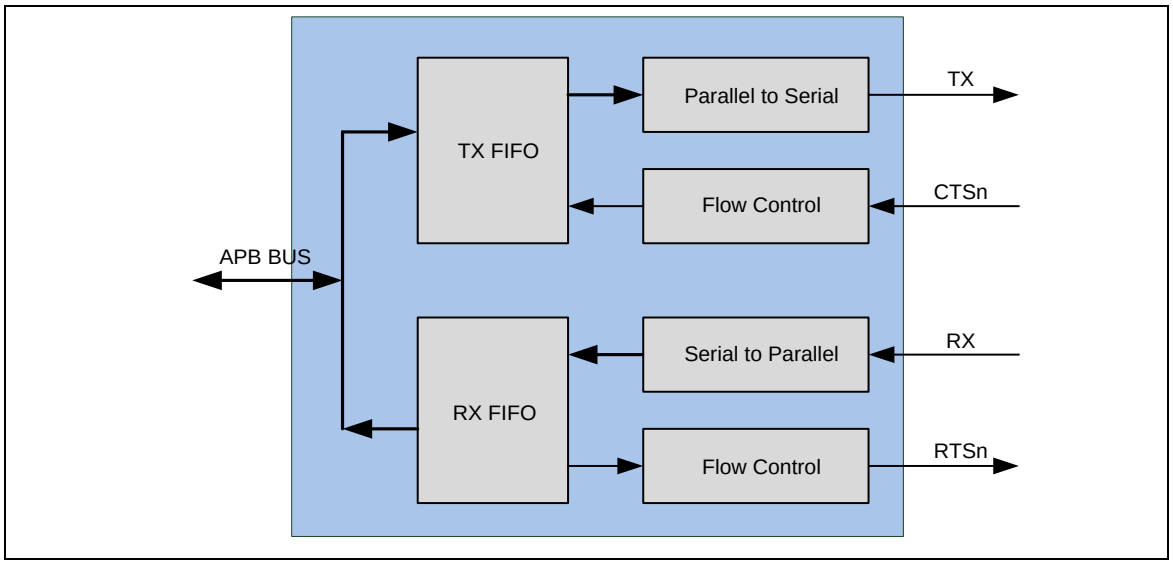



Figure 5-113 UART Auto-Flow Control Block Diagram

#### 5.19.4.2 Auto-baud rate detect

The UART supports auto-baud rate detection. If auto-baud feature enabled, controller will measure the bit time of the received data stream (LSB must be “1”) and set the divisor latch registers UART_BARD. Auto-baud rate detection is started by setting the UART_CTL [ABAUD_EN]. When the auto-baud rate detection flow finishes, the ABAUD_EN bit will be cleared automatically, and the UART_ISR [ABAUD_IS] and UART_TRSR [ABAUD_F] will be setting. If have time-out occurs (baud rate counter overflow), the UART_ISR [ABAUD_IS] and UART_TRSR [ABAUD_TOUT_F] will be setting. The following diagram demonstrates the auto-baud rate detection function.

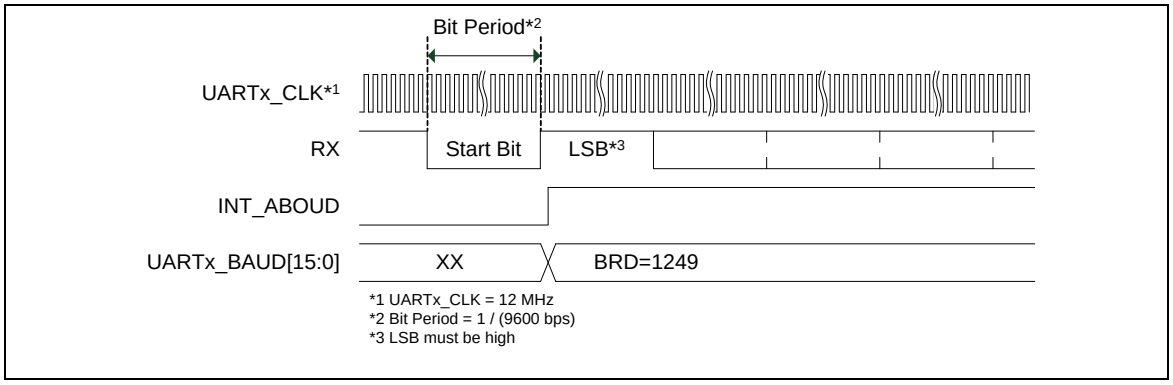



Figure 5-114 UART Auto-Baud Rate Block Diagram

#### 5.19.4.3 Wake-Up Function

The UART0 and UART1 controllers support wake-up system function. The wake-up function includes CTSn wake-up function (UART_CTL [WAKE_CTS_EN]) and data wake-up function (UART_CTL [WAKE_DATA_EN]). When the system is in power-down, the UART can wake-up system by CTSn pin or by incoming data. When incoming data wakes system up, the incoming

data will be received and stored in FIFO, and controller will clear the UART_CTL [WAKE_DATA_EN] register automatically. The following diagram demonstrates the wake-up function.

**CTSn Wake-Up Case 1**

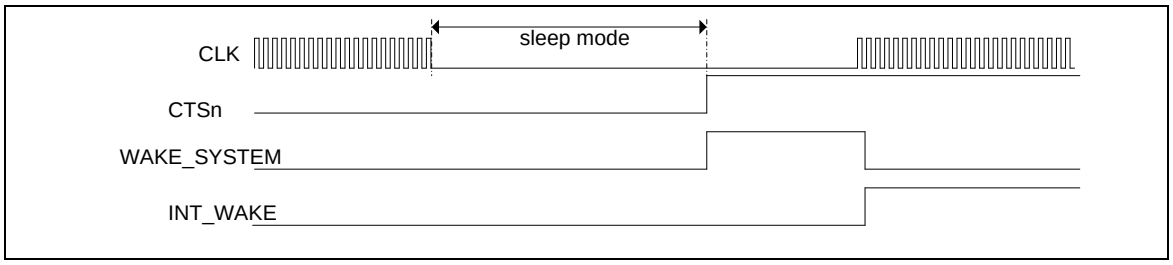



Figure 5-115 UART CTSn Wake-Up Case 1

**CTSn Wake-Up Case 2**

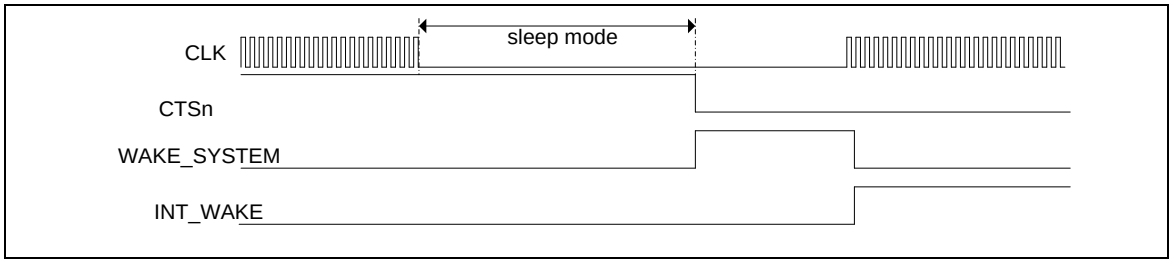



Figure 5-116 UART CTSn Wake-Up Case 2

**Data Wake-Up**

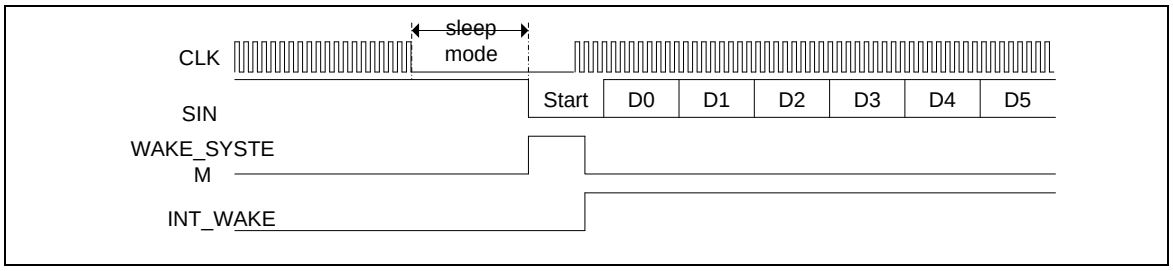



Figure 5-117 UART DATA Wake-Up

5.19.4.4 IrDA Function Mode

The UART support IrDA SIR (Serial Infrared) Transmit Encoder and Receive Decoder. IrDA mode is selected by setting the FUN_SEL bit in UART_FUN_SEL register to select IrDA mode and when operating in IrDA mode, the receive FIFO trigger level must be “1” by setting UART_TLCTL [RFITL] = “0”.

The UART_BAUD [DIV_16_EN] bit must be enabled in IrDA mode operation.

Baud Rate = Clock / (16 * (BRD + 1)), where BRD is Baud Rate Divider in UART_BAUD register.

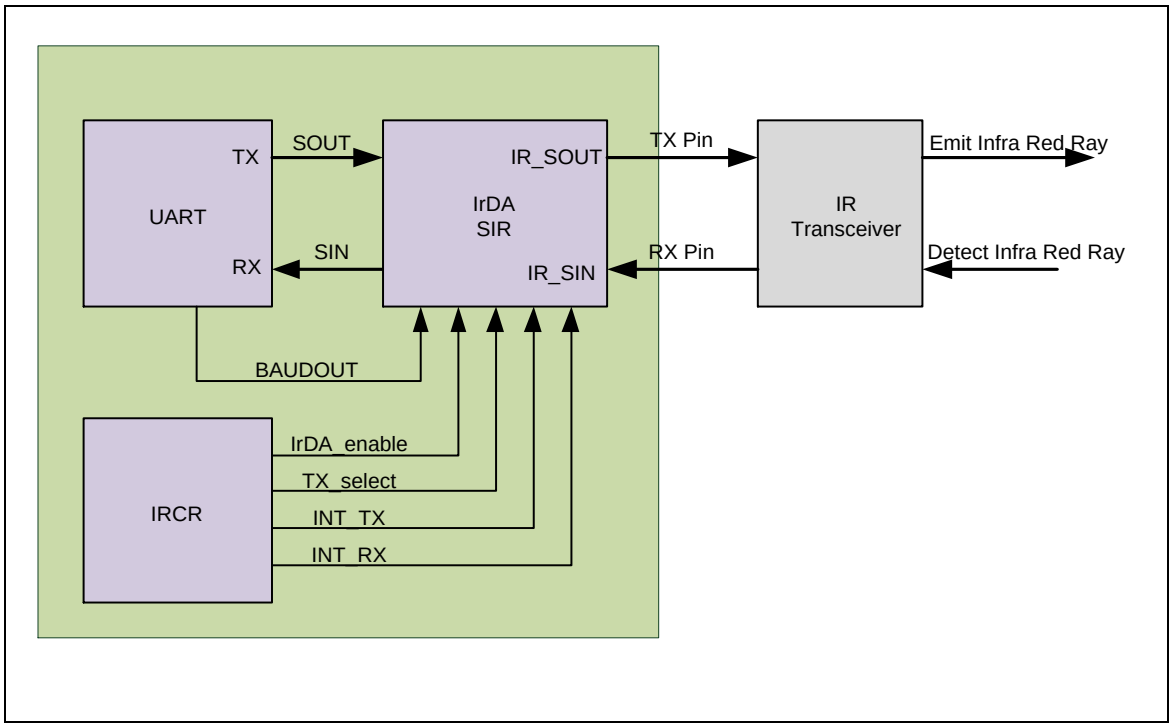



Figure 5-118 IrDA Block Diagram

### IrDA SIR Transmit Encoder

The IrDA SIR Transmit Encoder modulates Non-Return-to Zero (NRZ) transmit bit stream output from UART. The IrDA SIR physical layer specifies the usage of Return-to-Zero, Inverted (RZI) mlation scheme which represent logic 0 as an infra light pulse. The modulated output pulse stream is transmitted to an external output driver and infrared Light Emitting Diode.

The transmitted pulse width is specified as 3/16 period of baud rate.

### IrDA SIR Receive Decoder

The IrDA SIR Receive Decoder demodulates the return-to-zero bit stream from the input detector and outputs the NRZ serial bits stream to the UART received data input. The decoder input is normally high in the idle state. (Because of this, IRCR bit 6 should be set as “1” by default)

A start bit is detected when the decoder input is LOW

### IrDA SIR Operation

The IrDA SIR Encoder/decoder provides functionality which converts between UART data stream and half duplex serial SIR interface. The following diagram is IrDA encoder/decoder waveform:

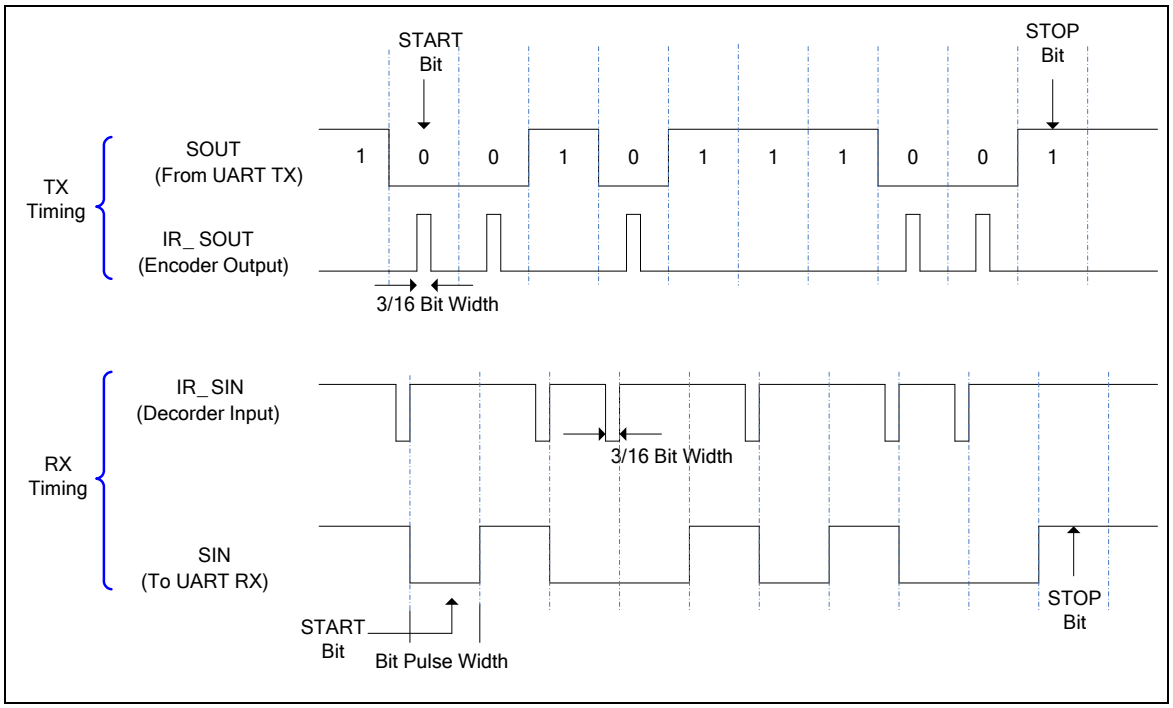



Figure 5-119 IrDA TX/RX Timing Diagram

5.19.4.5 RS-485 Function Mode

The UART supports RS-485 9-bit mode function. The RS-485 mode is selected by setting the UART_FUN_SEL register to select RS-485 function and when operating in RS-485 mode, the receive FIFO trigger level must be “1” by setting UART_TLCTL [RFITL] = “0”. The RS-485 driver control is implemented by the RTSn control signal from an asynchronous serial port to enable the RS-485 driver. In RS-485 mode, many characteristics of the RX and TX are same as UART.

In RS-485 function mode, the bit 9 will be configured as address bit. For data characters, the bit 9 is set to “0”. Software can program UART_TLCTL register to control the 9-th bit (When the PBE, EPE and SPE are set, the 9-th bit is transmitted as “0” and when PBE and SPE are set and EPE is cleared, the 9-th bit is transmitted as “1”). The Controller supports three operation modes that are RS-485 Normal Multi-drop Operation Mode (RS-485 NMM Mode), RS-485 Auto Address Detection Operation Mode (RS-485 AAD Mode) and RS-485 Auto Direction Control Operation Mode (RS-485 AUD Mode). One of the three operation modes can be selected by programming UART_ALT_CTL register, and software can driving the transfer delay time between the last stop bit leaving the TX-FIFO and the de-assertion by setting UART_TMCTL [DLY] register.

**RS-485 Normal Multi-drop Operation Mode (RS-485 NMM Mode)**

In RS-485 Normal Multi-drop operation mode, software can decide whether receiver will ignore data before an address byte is detected (bit 9 = “1”). When an address byte be detected (bit 9 = “1”) by hardware, the address byte data will be stored in the RX-FIFO. Software can decide whether to enable or disable receiver to accept the following data byte by setting UART_CTL [RX_DIS]. If the receiver is be enabled (UART_CTL[RX_DIS] is low), all received byte data will be accepted and stored in the RX-FIFO, and if the receiver is disabled(UART_CTL[RX_DIS] is high), all received byte data will be ignore until the next address byte be detected. If software disable receiver by setting UART_CTL [RX_DIS] register high, when a next address byte is detected, the controller will clear the UART_CTL [RX_DIS] bit and the address byte data will be stored in the RX-FIFO.

**Program Sequence Example:**

1. Program FUN_SEL in UART_FUN_SEL to select RS-485 function.

2. Program the RX_DIS bit in UART_CTL register to determine whether to store the received data before an address byte is detected (bit 9 = “1”).

3. Program the RS-485_NMM by setting UART_ALT_CTL register.

4. When an address byte is detected (bit 9 = “1”), hardware will set UART_ISR [RLS_IS] and UART_TRSR [RS-485_ADDET_F] flag.

5. Software can decide whether to accept the following data byte by setting UART_CTL [RX_DIS].

6. Repeat step 4 and step 5.

**RS-485 Auto Address Detection Operation Mode (RS-485 AAD Mode)**

In RS-485 Auto Address Detection Operation Mode, the receiver will ignore any data until an address byte is detected (bit 9 = “1”) and the address byte data match the UART_ALT_CTL [ADDR_MATCH] value. The address byte data will be stored in the RX-FIFO. The following all data will be accepted and stored in the RX-FIFO until an address byte not match the UART_ALT_CTL [ADDR_MATCH] value. In RS-485 AAD mode, don’t fill any value to UART_CTL [RX_DIS] bit.

Program Sequence example:

1. Program FUN_SEL in UART_FUN_SEL to select RS-485 function.

2. Program the RS-485_AAD by setting UART_ALT_CTL register.

3. When an address byte is detected (bit9 = “1”), hardware will compare the address byte and the UART_ALT_CTL [ADDR_MATCH] value.

4. If the address byte matches the UART_ALT_CTL [ADDR_MATCH] value, hardware will set UART_ISR [RLS_IS] and UART_TRSR [RS-485_ADDET_F] flag. And the receiver will sorted address byte to FIFO and accept the following data transfer and stored data in FIFO until next address byte be detected.

However if the address byte does not match the UART_ALT_CTL [ADDR_MATCH] value, hardware will ignored the address byte data and ignored the following data nsfer

2012. 5. Respect step 3 and step 4.

**RS-485 Auto Direction Mode (RS-485 AUD Mode)**

Another option function of RS-485 controllers is RS-485 auto direction control function. The RS-485 driver control is implemented by using the RTSn control signal from an asynchronous serial port to enable the RS-485 driver. The RTSn line is connected to the RS-485 driver enable such that setting the RTSn line to high (logic “1”) will enable the RS-485 driver. Setting the RTSn line to low (logic “0”) will put the driver into the tri-state condition. User can setting LEV_RTS in UART_MCSR register to change the RTSn driving level.

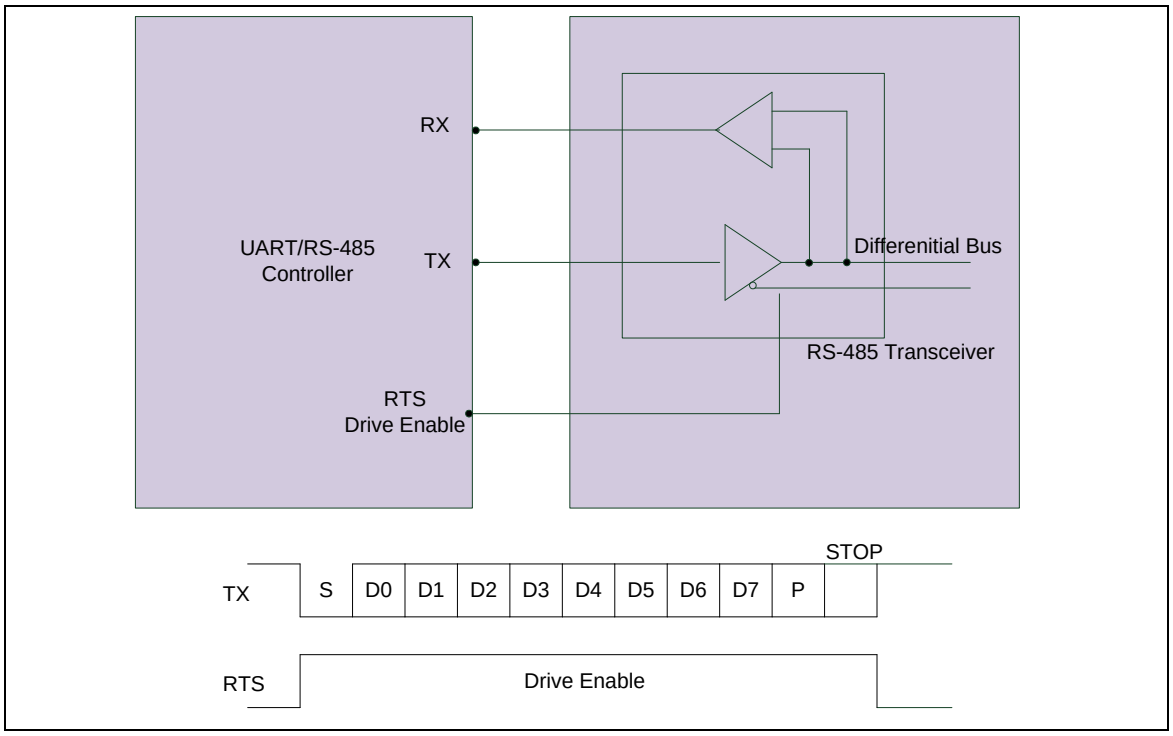



Figure 5-120 Structure of RS-485 Frame

5.19.4.6 LIN (Local Interconnection Network) Function Mode

The UART support LIN function, and LIN mode is selected by setting the UART_FUN_SEL register to select LIN function. In LIN function mode, each byte field is initialed by a start bit with value 0 (dominant), followed by 8 data bits (LSB is first) and ended by 1 stop bit with value one (recessive) in accordance with the LIN standard.

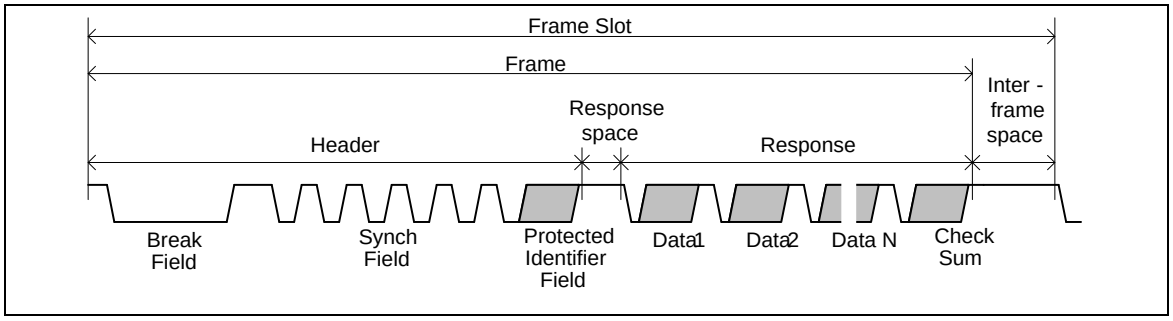



Figure 5-121 Structure of LIN Frame

**Program Flow of LIN Bus Transmit Transfer (TX)**

The program flow of LIN Bus Transmit transfer (TX) is shown as follows

**Case 1: The Header Field Select as "Break"**

1. Set LIN function mode by setting UART_FUN_SEL register.

2. Choose the data header to "break field" by setting UART_ALT_CTL [LIN_HEAD_SEL]).

3. Enable UART_ALT_CTL [BIT_ERR_EN] bit, and when the SIN pin is not equal to SOUT pin that the hardware will generator an interrupt to CPU). If user wants to receive data at the same time, user must enable UART_ALT_CTL [LIN_RX_EN] bit.

4. Fill LIN_TX_BCNT to choose break field length. (The break field length is LIN_TX_BCNT + 8).

5. Set the LIN_TX_EN bit in UART_ALT_CTL register to start transmission break field, and when break filed operation is finished, hardware will set UART_ISR [LIN_IS] and UART_TRSR [LIN_TX_F] flag and UART_ALT_CTL[LIN_TX_EN will be cleared automatically.

6. Fill 0x55 to UART_THR to request synch field transmission.

7. Fill the protected identifier value (PID) in the UART_THR

8. Fill N bytes data and Checksum to UART_THR then repeat step 4 ~ step 8 to transmit the data.

**Case 2: The Header Field Select as "Break + Sync"**

1. Set LIN function mode by setting UART_FUN_SEL register.

2. Choose the data header to "break + sync" field by setting UART_ALT_CTL [LIN_HEAD_SEL]).

3. Enable UART_ALT_CTL [BIT_ERR_EN] bit, and when the SIN pin is not equal to SOUT pin that the hardware will generator an interrupt to CPU). If user wants to receive data at the same time, user must enable UART_ALT_CTL [LIN_RX_EN] bit.

4. Fill LIN_TX_BCNT to choose break field length. (The break field length is LIN_TX_BCNT + 8).

5. Set the LIN_TX_EN bit in UART_ALT_CTL register to start transmission break and sync field, and when break and sync filed operation is finished, hardware will set UART_ISR [LIN_IS] and UART_TRSR [LIN_TX_F] flag and UART_ALT_CTL[LIN_TX_EN will be cleared automatically.

6. Fill the protected identifier value (PID) in the UART_THR

7. Fill N bytes data and Checksum to UART_THR then repeat step 4 ~ step 7 to transmit the data.

**Case 3: The Header Field Select as "Break + Sync + PID"**

1. Set LIN function mode by setting UART_FUN_SEL register.

2. Choose the data header to "break + sync + PID field" by setting UART_ALT_CTL [LIN_HEAD_SEL]).

3. Enable UART_ALT_CTL [BIT_ERR_EN] bit, and when the SIN pin is not equal to SOUT pin that the hardware will generator an interrupt to CPU). If user wants to receive data at the same time, user must enable UART_ALT_CTL [LIN_RX_EN] bit.

4. Fill LIN_TX_BCNT to choose break field length. (The break field length is LIN_TX_BCNT + 8).

5. Set the LIN_TX_EN bit in UART_ALT_CTL register to start transmission break, sync and PID field, and when break, sync and PID filed operation is finished, hardware will set UART_ISR [LIN_IS] and UART_TRSR [LIN_TX_F] flag and UART_ALT_CTL[LIN_TX_EN will be cleared automatically.

6. Fill N bytes data and Checksum to UART_THR then repeat step 4 ~ step 6 to transmit the data.

**Program Flow of LIN Bus Receiver transfer (RX)**

The program flow of LIN Bus Receiver transfer (RX) is show as follows.

**Case 1: The header Field Select as "Break"**

1. Set LIN function mode by setting UART_FUN_SEL register.

2. Choose the data header to "break field" by setting UART_ALT_CTL [LIN_HEAD_SEL]).

3. Set the LIN_RX_EN bit in UART_ALT_CTL register to enable LIN RX mode.

4. Wait for the flag LIN_RX_F in UART_TRSR to check RX received break field or not. (The break field will not be stored in FIFO)

5. Wait for the flag RDA_IF in UART_ISR and read back the UART_RBR register.

**Case 2: The Header Field Select as "Break + Sync"**

1. Set LIN function mode by setting UART_FUN_SEL register.

2. Choose the data header to "break + sync field" by setting UART_ALT_CTL [LIN_HEAD_SEL]).

3. Set the LIN_RX_EN bit in UART_ALT_CTL register to enable LIN RX mode.

4. Wait for the flag LIN_RX_F in UART_TRSR to check RX received break field and sync field. If the break and sync field is received, hardware will set UART_TRSR [LIN_RX_F] flag If the break be received but the sync field not equal 0x55, hardware will set UART_TRSR [LIN_RX_F] and UART_TRSR [LIN_RX_SYNC_ERR_F] flag. The break and sync data (equal 0x55 or not) will not be stored in FIFO.

5. Wait for the flag RDA_IF in UART_ISR and read back the UART_RBR register.

**Case 3: The Header Field Select as "Break + Sync + PID"**

1. Set LIN function mode by setting UART_FUN_SEL register.

2. Choose the data header to "break + sync + PID field" by setting UART_ALT_CTL [LIN_HEAD_SEL]).

3. Set the LIN_RX_EN bit in UART_ALT_CTL register to enable LIN RX mode.

4. In this operation mode, hardware will control data automatically. Hardware will ignore any data until received break + sync (0x55) + PID value match the UART_ALT_CTL [ADDR_MATCH] value (break + sync + PID will not be stored in FIFO). When received break + sync (0x55) + PID value match the UART_ALT_CTL [ADDR_MATCH] value, hardware will set UART_TRSR [LIN_RX_F] and the following all data will be accepted and stored in the RX-FIFO until detect next break field. If the receiver received break + wrong sync (not equal 0x55) + PID value, that hardware will set UART_TRSR [LIN_RX_F] and UART_TRSR [LIN_RX_SYNC_ERR_F] flag and the receiver will be disabled. If the receiver received break + sync (0x55) + wrong PID value, that hardware will set UART_TRSR [LIN_RX_F] flag and the receiver will be disabled.

5. Wait for the flag RDA_IF in UART_ISR and read back the UART_RBR register.

## 5.20 USB

### 5.20.1 Overview

The USB controller is a USB 2.0 full-speed device controller. It is compliant with USB 2.0 full speed device specification and supports control/bulk/interrupt/isochronous transfer types.

In this device controller, there are two main interfaces: the APB bus and USB bus which comes from the USB PHY transceiver. For the APB bus, the CPU can program control registers through it. There is an internal 512-byte SRAM as data buffer in this controller. For IN token or OUT token transfer, it is necessary to write data to SRAM or read data from SRAM through the APB interface. Users need to allocate the effective starting address of SRAM for each endpoint buffer through “buffer segmentation register (BUFSEG)”.

This device controller contains 8 configurable endpoints. Each endpoint can be configured as IN or OUT endpoint. The function address of the device and endpoint number in each endpoint shall be configured properly in advance for receiving or transmitting a data packet correctly. The transmitting/receiving length in each endpoint is defined in maximum payload register (MXPLD) and the handshakes between Host and Device are also handled by it.

There are four different interrupt events in this controller. They are the wake-up function, device plug-in or plug-out event, USB events, like IN ACK, OUT ACK etc, and BUS events, like suspend and resume, etc. Any event will cause an interrupt, and users just need to check the related event flags in interrupt event status register (USB_INTSTS) to acknowledge what kind of events occurring, and then check the related USB Endpoint Status Register (USB_EPSTS) to acknowledge what kind of event occurring in this endpoint.

A software-disable function is also supported for this USB controller. It is used to simulate the disconnection of this device from the host. If user enables the DRVSE0 bit (USB_CTL[4]), the USB controller will force USB_DP and USB_DM to level low and USB device function is disabled (disconnected). After disable the DRVSE0 bit, USB_DP will be pulled high by internal pull-high circuit then host will enumerate the USB device connection again.

Reference: Universal Serial Bus Specification Revision 2.0

### 5.20.2 Features

This Universal Serial Bus (USB) performs a serial interface with a single connector type for attaching all USB peripherals to the host system. Following is the feature listing of this USB.

- Compliant with USB 2.0 Full-Speed specification.
- Provide 1 interrupt vector with 4 different interrupt events (WAKEUP, FLDET, USB and BUS).
- Supports Control/Bulk/Interrupt/Isochronous transfer type.
- Supports suspend function when no bus activity existing for 3 ms.
- Provide 8 endpoints for configurable Control/Bulk/Interrupt/Isochronous transfer types
- 512-byte SRAM buffer inside
- Provide remote wake-up capability.

### 5.20.3 Block Diagram

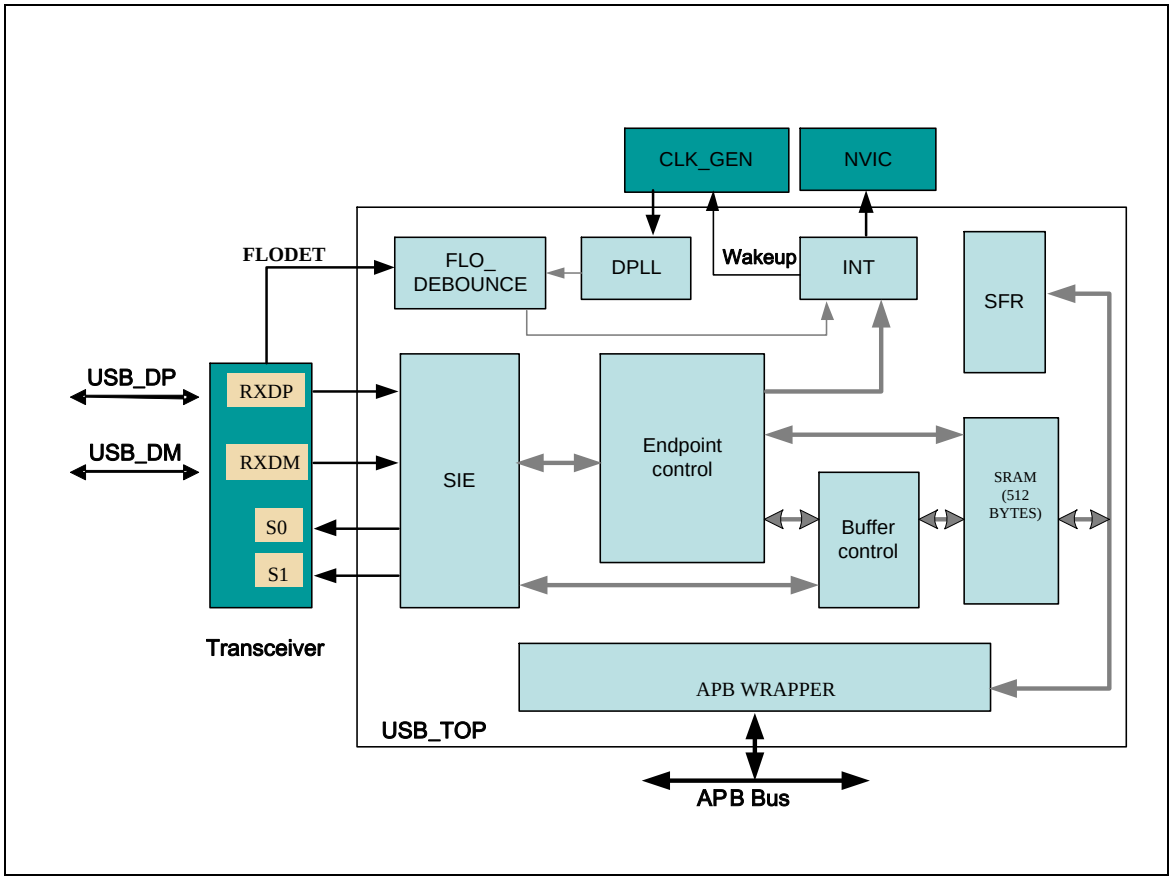



Figure 5-122 USB Block Diagram

### 5.20.4 Functional Description

5.20.4.1 SIE (Serial Interface Engine)

The SIE is the front-end of the device controller and handles most of the USB packet protocol. The SIE typically comprehends signaling up to the transaction level. The functions that it handles could include:

- Packet recognition, transaction sequencing
- SOP, EOP, RESET, RESUME signal detection/generation
- Clock/Data separation
- NRZI Data encoding/decoding and bit-stuffing
- CRC generation and checking (for Token and Data)
- Packet ID (PID) generation and checking/ decoding
- Serial-Parallel/ Parallel-Serial conversion

5.20.4.2 Endpoint Control

There are 8 endpoints in this controller. Each of the endpoint can be configured as IN or OUT endpoint. All the operations including Control, Bulk, Interrupt and Isochronous transfer are

implemented in this block. The block of ENDPOINT CONTROL is also used to manage the data sequential synchronization, endpoint states, current start address, transaction status, and data buffer status for each endpoint.

#### 5.20.4.3 Buffer Control

There is 512-byte SRAM in the controller and the 8 endpoints share this buffer. The user shall configure each endpoint's effective starting address in the buffer segmentation register before the function active. The BUFFER CONTROL block is used to control each endpoint's effective starting address and its SRAM size is defined in the MXPLD register.

The following figure describes the starting address for each endpoint according to the content of BUFSEG and MXPLD registers. If the BUFSEG0 is programmed as 0x08h and MXPLD0 is set as 0x40h, the SRAM size of endpoint 0 is start from USB_BASE + 0x108h and end in USB_BASE + 0x148h because the USB SRAM base is USB_BASE + 0x100h.

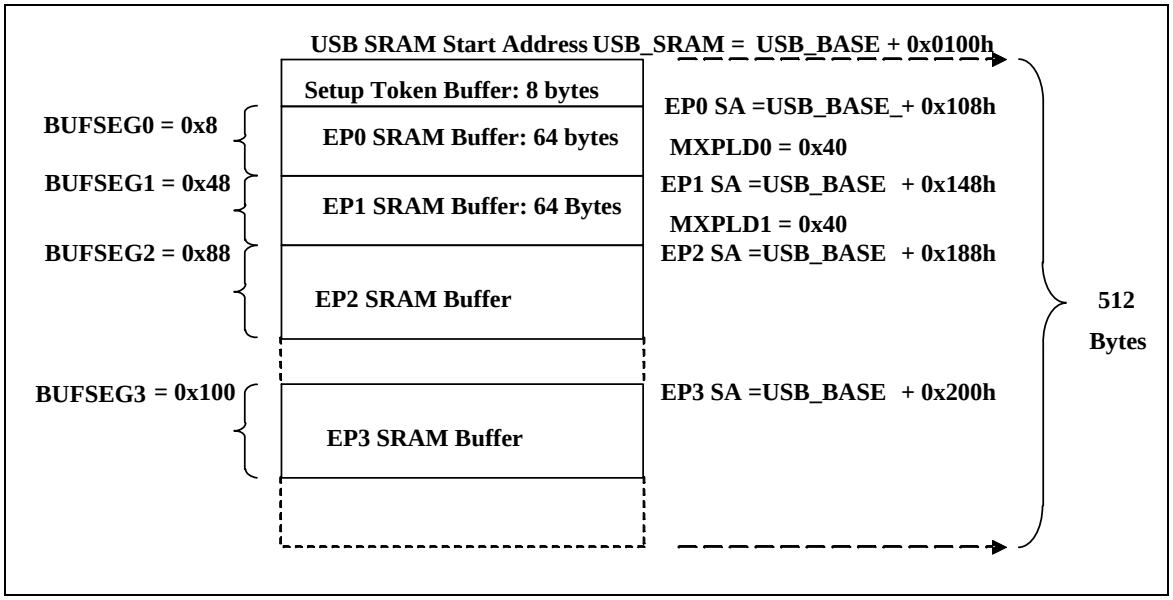



Figure 5-123 Endpoint allocation in SRAM

#### 5.20.4.4 DPLL (Digital Phase Lock Loop)

The bit rate of USB data is 12 MHz. The DPLL uses the 48 MHz which comes from the clock controller to lock the input data RXDP and RXDM. The 12 MHz bit rate clock is also converted from DPLL.

#### 5.20.4.5 Floating De-bounce

A USB device may be plug-in or plug-out from the USB. In order to monitor the state of a USB device when it is detached from the USB, the device controller provides hardware de-bounce for USB floating detect interrupt to avoid bounce problems on USB plug in and unplug. Floating detect interrupt appears about 10 ms later than USB plug-in and plug-out. A user can acknowledge USB plug-in/plug-out by reading "FLDET" in USB_BUSSTS register. The flag in "FLDET" bit represents the current state on the bus without de-bounce. If the FLDET is 1, it means the controller has plug-in the USB bus. If polling this bit to check USB state, users must add software de-bounce if necessary.

#### 5.20.4.6 Interrupt & Wake-up

This USB provides wake-up function and 1 interrupt vector with 4 interrupt events (WAKEUP, FLDET, USB and BUS). The WAKEUP event is used to wake-up the system clock when the power-down mode is enabled. (The power-down mode function is defined in system power control register, PWRCTL). The FLDET event is used for USB plug-in or plug-out. The USB event notifies

users of some USB requests, like IN ACK, OUT ACK etc., and the BUS event notifies users of some bus events, like suspend, resume, etc. User must set related bits in both NVIC and "interrupt enable register (USB_INTEN) of USB Device Controller to enable USB interrupts.

Wake-up interrupt is only present when the system entered power-down mode and then wake-up event had happened. When the system enters power-down mode, any change on USB_DP, USB_DM and device floating detect pin can wake-up this chip (provided that USB wake-up function is enabled). If this change is not intentionally, for example, a noise on floating detect pin, no interrupt but wake-up interrupt will occur. After USB wake-ups, this interrupt will occur when no other USB interrupt events are present for more than 20ms. The following is the control flow of wake-up interrupt.

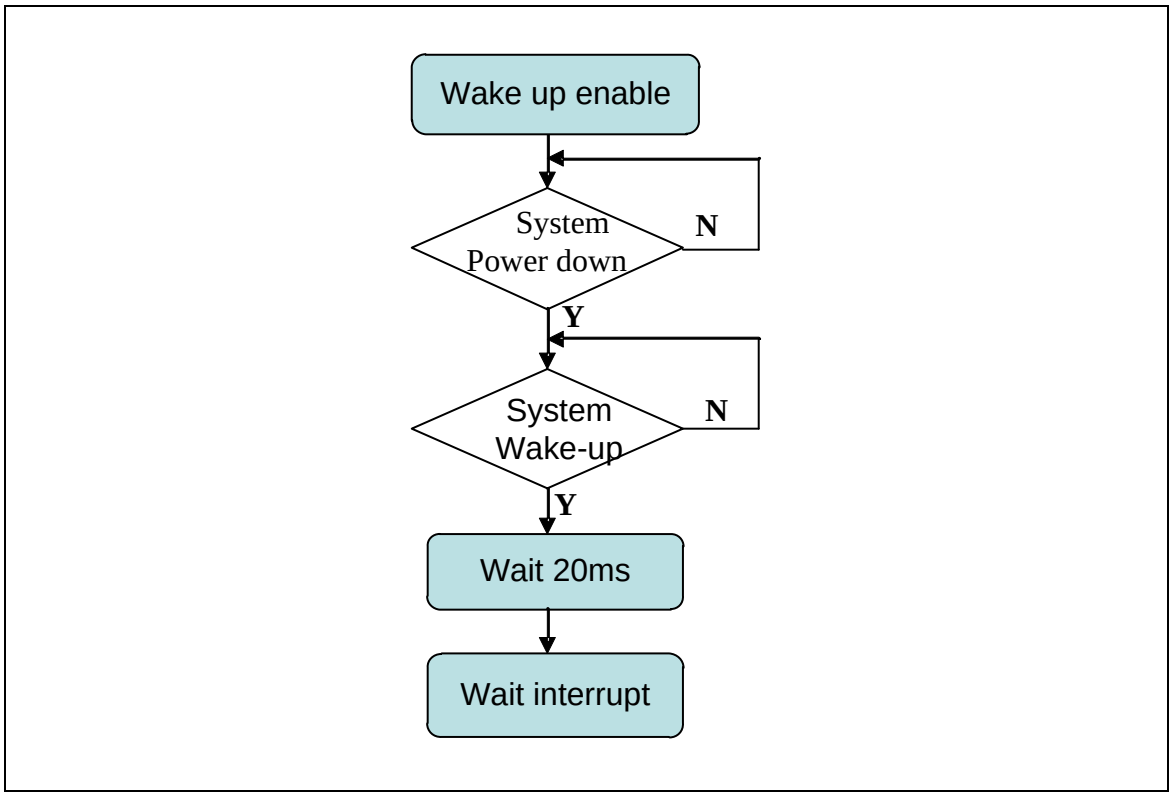



Figure 5-124 USB Wake-up Interrupt Operation Flow

USB interrupt is used to notify users of any USB event on the bus, and a user can read USB_EPSTS and USB_INTSTS to know what kind of request is to which endpoint and take necessary responses.

Same as USB interrupt, BUS interrupt notifies users of some bus events, like USB reset, suspend, time-out, and resume. A user can read USB_BUSSTS register to acknowledge bus events.

#### 5.20.4.7 Power Saving

USB turns off PHY transceiver automatically to save power while this chip enters power-down mode. Furthermore, the users can write 0 into the PHY_EN bit (USB_CTL[1]), to turn off the PHY under special circumstances like suspend to save power.

Handling Transactions with USB device Peripheral

User can use interrupt or polling USB_INTSTS to monitor the USB Transactions. When a transaction occurs, the USB_INTSTS will be set by hardware and send an interrupt request to CPU (if related interrupt was enabled). Users can poll USB_INTSTS register to get these events

without interrupt. The following is the control flow with interrupt enable bits active.

When USB host has requested data from device controller, users need to prepare related data into the specified endpoint buffer. After the required data is in buffer, users need to write the actual data length in the specified MAXPLD register. Once this register is written, the internal signal “In_Rdy” will be asserted and the data in buffer will be transmitted immediately after receiving associated IN token from Host. Note that after transferring the specified data, the signal “In_Rdy” will de-assert automatically by hardware.

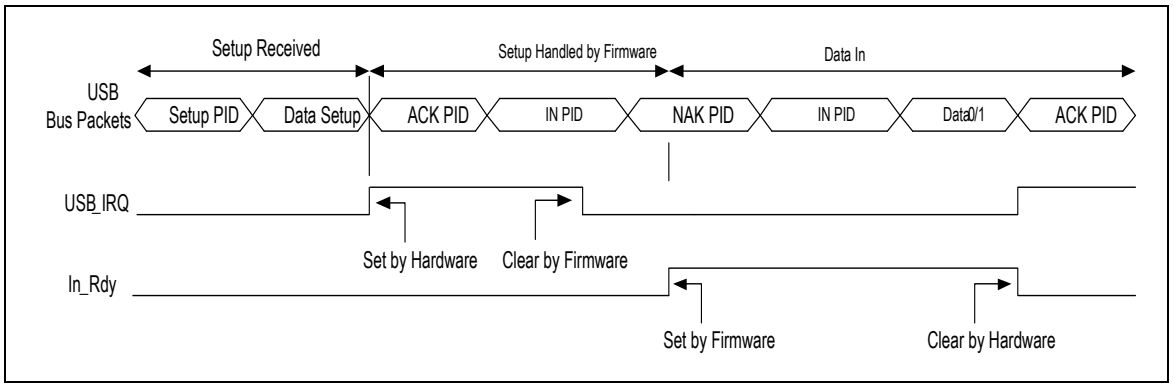



Figure 5-125 USB Setup Transaction Followed by Data IN Transaction

Alternatively, when USB host wants to transmit data to the OUT endpoint in the device controller, hardware will buffer these data to the specified endpoint buffer. After this transaction is completed, hardware will record the data length in related MAXPLD register and de-assert the signal “Out_Rdy”. This will avoid hardware accepting next transaction until users move out current data in the related endpoint buffer. Once users have processed this transaction, the related register “MAXPLD” needs to be written by firmware to assert the signal “Out_Rdy” again to receive next transaction.

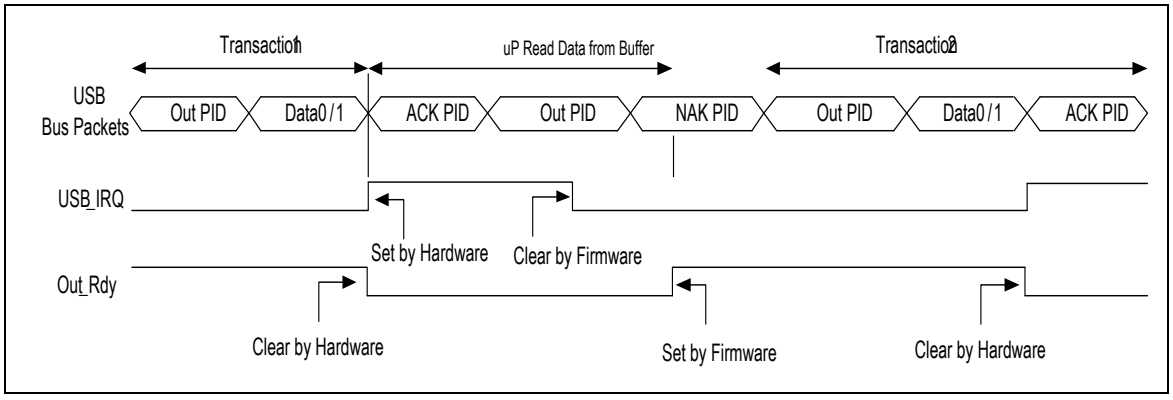



Figure 5-126 USB Data OUT Transaction

## 5.21 Watchdog Timer Controller

### 5.21.1 Overview

The purpose of Watchdog Timer is to perform a system reset after the software running into a problem. This prevents system from hanging for an infinite period of time. Besides, this Watchdog Timer supports the function to wake-up CPU from power-down mode. The watchdog timer includes an 18-bit free running counter with programmable time-out intervals.

### 5.21.2 Features

- 18-bit free running WDT counter for Watchdog timer time-out interval.
- Selectable time-out interval (2^4 ~ 2^18) and the time-out interval is 104 ms ~ 26.316 s (if WDT_CLK = 10 kHz).
- Reset period = (1 / 10 kHz) * 63, if WDT_CLK = 10 kHz.

### 5.21.3 Block Diagram

The Watchdog Timer clock control and block diagram are shown as follows.

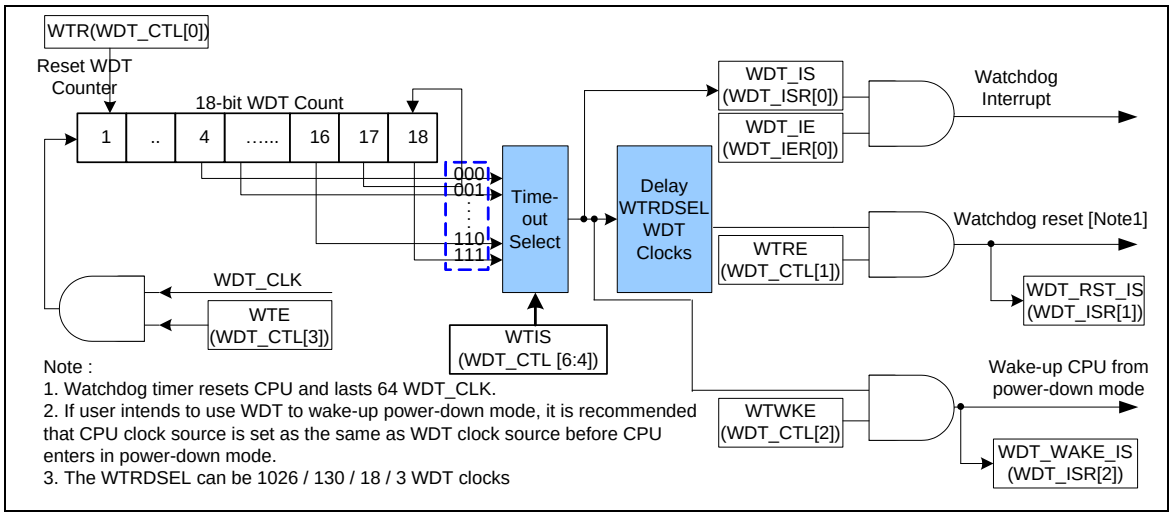



Figure 5-127 Watchdog Controller Block Diagram

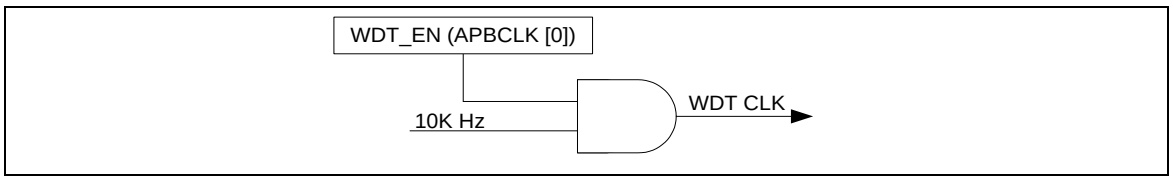



Figure 5-128 Watchdog Timer Clock Control Diagram

### 5.21.4 Functional Description

The purpose of Watchdog Timer is to perform a system reset after the software running into a problem. This prevents system from hanging for an infinite period of time. Besides, this Watchdog Timer supports the function to wake-up the chip from Power-down mode. Moreover, the Watchdog counter will be automatically reset when the chip is entering Power-down mode. The watchdog timer includes an 18-bit free running counter with programmable time-out intervals. The following figure show the watchdog time-out interval selection and the next figure show the timing of watchdog interrupt signal and reset signal.

Setting WTE (WDT_CTL [3]) enables the watchdog timer and the WDT counter starts counting up. When the counter reaches the selected time-out interval, Watchdog timer interrupt flag WDT_IS will be set immediately to request a WDT interrupt if the watchdog timer interrupt enable bit WDT_IE is set, in the meanwhile, a specified delay time (WTRDSEL * $T^{WDT}$) follows the time-out event. User must set WTR (WDT_CTL [0]) (Watchdog timer reset) high to reset the 18-bit WDT counter to avoid CPU from Watchdog timer reset before the delay time expires. WTR bit is auto cleared by hardware after WDT counter is reset. There are eight time-out intervals with specific delay time which are selected by Watchdog timer interval select bits WTIS (WDTCR [6:4]). If the WDT counter has not been cleared after the specific delay time expires, the watchdog timer will set Watchdog Timer Reset Flag (WDT_RST_IS) high and reset CPU. This reset will last

63 * WDT clocks ($T^{RST}$) then CPU restarts executing program from reset vector (0x0000 0000). WDT_RST_IS will not be cleared by Watchdog reset. User may poll WDT_RST_IS by software to recognize the reset source.

| WTIS | WTR Time-out Interval | Interrupt Period $T_{INT}$ | Time-out Interval WDT_CLK = 10 KHz $T_{TIS}$ | Reset Internal WDT_CLK = 10 KHz $T_{WTR}$ |
|---|---|---|---|---|
| 000 | $2^4 * T_{WDT}$ | $1024 * T_{WDT}$ | 1.6 ms | 104 ms |
| 001 | $2^6 * T_{WDT}$ | $1024 * T_{WDT}$ | 6.4 ms | 108.8 ms |
| 010 | $2^8 * T_{WDT}$ | $1024 * T_{WDT}$ | 25.6 ms | 128 ms |
| 011 | $2^{10} * T_{WDT}$ | $1024 * T_{WDT}$ | 102.4 ms | 204.8 ms |
| 100 | $2^{12} * T_{WDT}$ | $1024 * T_{WDT}$ | 407 ms | 512 ms |
| 101 | $2^{14} * T_{WDT}$ | $1024 * T_{WDT}$ | 1.638 s | 1.741 s |
| 110 | $2^{16} * T_{WDT}$ | $1024 * T_{WDT}$ | 6.553 s | 6.656 s |
| 111 | $2^{18} * T_{WDT}$ | $1024 * T_{WDT}$ | 26.214 s | 26.316 s |

Table 5-14 Watchdog Time-out Interval Selection

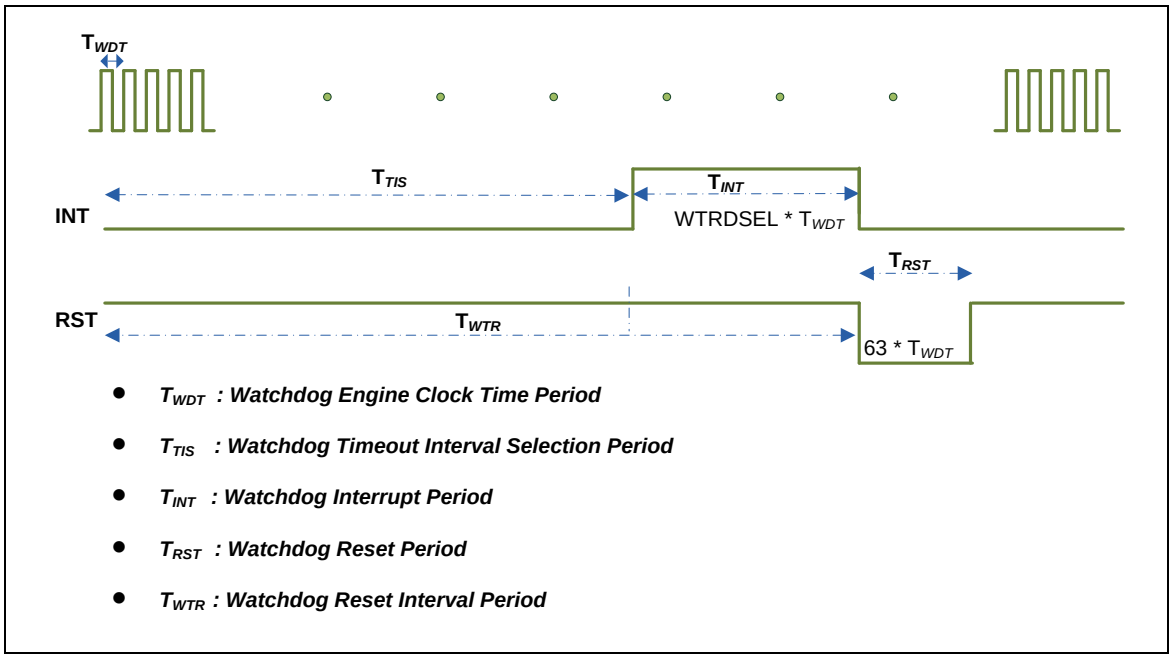



Figure 5-129 Watchdog Timing of Interrupt and Reset Signal

## 5.22 Window Watchdog Timer Controller

### 5.22.1 Overview

The purpose of Window Watchdog Timer is to perform a system reset within a specified window period to prevent software run to uncontrollable status by any unpredictable condition.

### 5.22.2 Features

- 6-bit down counter and 6-bit compare value to make the window period flexible
- Selectable WWDT clock pre-scale counter to make WWDT time-out interval variable

### 5.22.3 Block Diagram

The Window Watchdog Timer block diagram is shown as follows.

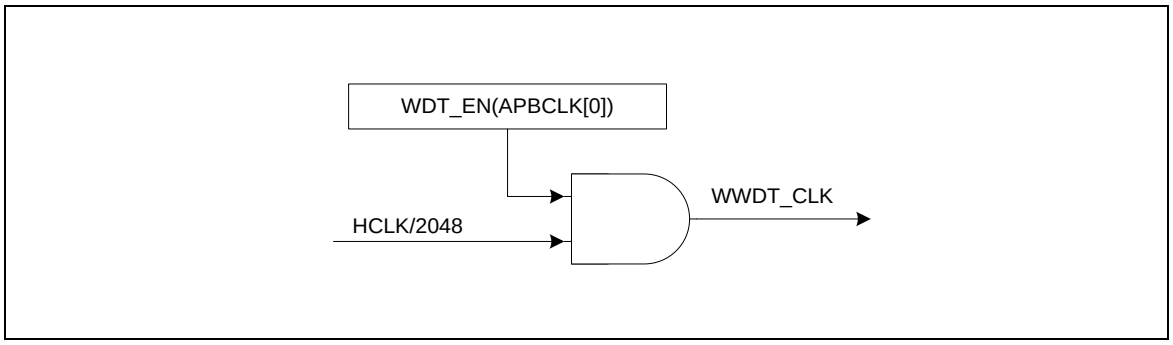



Figure 5-130 Window Watchdog Controller Block Diagram

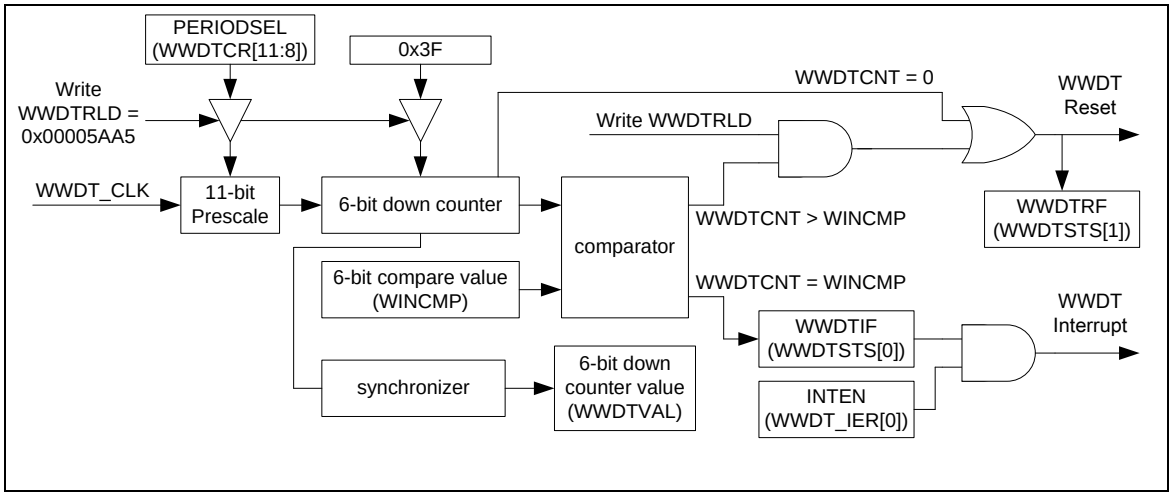



Figure 5-131 Watchdog Controller Block Diagram

### 5.22.4 Functional Description

The window watchdog timer includes a 6-bit down counter with programmable prescaler to define different time-out intervals.

The clock source of 6-bit window watchdog timer is based on system clock divide 2048 with a programmable 11-bit prescaler. The programmable 11-bit prescaler is controlled by register PERIODSEL (WWDTCR[11:8]) and the correlate of PERIODSEL and prescaler value is list in following table

| PERIODSEL | Prescaler Value | Timeout Period | Timeout Interval<br>12 M/2048 = 5.859 kHz<br>WWDT_CLK=5.859 kHz |
|---|---|---|---|
| 0000 | 1 | 1 * 64 * $T_{WWDT}$ | 10.9 ms |
| 0001 | 2 | 2 * 64 * $T_{WWDT}$ | 21.8 ms |
| 0010 | 4 | 4 * 64 * $T_{WWDT}$ | 43.7 ms |
| 0011 | 8 | 8 * 64 * $T_{WWDT}$ | 87.4 ms |
| 0100 | 16 | 16 * 64 * $T_{WWDT}$ | 174.7 ms |
| 0101 | 32 | 32 * 64 * $T_{WWDT}$ | 349.5 ms |
| 0110 | 64 | 64 * 64 * $T_{WWDT}$ | 699.1 ms |
| 0111 | 128 | 128 * 64 * $T_{WWDT}$ | 1.3981 s |
| 1000 | 192 | 192 * 64 * $T_{WWDT}$ | 2.0971 s |
| 1001 | 256 | 256 * 64 * $T_{WWDT}$ | 2.7962 s |
| 1010 | 384 | 384 * 64 * $T_{WWDT}$ | 4.1943 s |
| 1011 | 512 | 512 * 64 * $T_{WWDT}$ | 5.5924 s |
| 1100 | 768 | 768 * 64 * $T_{WWDT}$ | 8.3886 s |
| 1101 | 1024 | 1024 * 64 * $T_{WWDT}$ | 11.1848 s |
| 1110 | 1536 | 1536 * 64 * $T_{WWDT}$ | 16.7772 s |
| 1111 | 2048 | 2048 * 64 * $T_{WWDT}$ | 22.3696 s |

Table 5-15 Window Watchdog Prescaler Value Selection

The window watchdog timer can be enabled by software setting WWDTEN (WWDTCR[0]) to 1. As window watchdog timer is enabled, the down counter will start counting from 0x3F and cannot be stopped by software.

During WWDT down counting, the WWDT interrupt will happen if the counter value is equal to window watchdog timer compare value WINCMP (WWDTCR[21:16)) and INTEN(WWDT_IER[0]) is set to 1. The WWDT reset will happen if the WWDT counter value reaches to 0. Before WWDT counter down to 0, software can write certain value (0x00005AA5) to register WWDTRLD to reload 0x3F to WWDT counter to prevent WWDT reset happen and this reload action only active when WWDT counter value is equal or smaller than WINCMP. If software writes WWDTRLD during the period that WWDT counter larger than WINCMP, additional WWDT reset will happen to cause chip be reset.

When software writes certain value (0x00005AA5) to register WWDTRLD to reload WWDT counter, it need 3 window watchdog clocks to sync reload command to actually perform reload action. It means if software set window watchdog clock prescaler as divide 1, the compare value WINCMP (WWDTCR[21:16]) should larger than 2 or software will not able to reload WWDT counter before WWDT reset happened.

To prevent program run to unexpected code to disable window watchdog, the control register WWDTCR and WWDT_IER can only be write 1 time after chip power on or reset. Software can not to disable window watchdog, change pre-scale period or change window compare value as window watchdog is enabled by software unless chip is reset. And when CPU in sleep mode (WFI

or WFE), the CPU will be disabled but system clock will keep, so the window watchdog timer will still counting; but when system in Power-down mode, the system clock will also be disabled, so the window watchdog timer will be stopped.

# 6 ARM® CORTEX™-M0 CORE

## 6.1 Overview

The Cortex™-M0 processor is a configurable, multistage, 32-bit RISC processor. It has an AMBA AHB-Lite interface and includes an NVIC component. It also has optional hardware debug functionality. The processor can execute Thumb code and is compatible with other Cortex-M profile processor. The profile supports two modes – Thread mode and Handler mode. Handler mode is entered as a result of an exception. An exception return can only be issued in Handler mode. Thread mode is entered on Reset, and can be entered as a result of an exception return. The following figure shows the functional controller of processor.

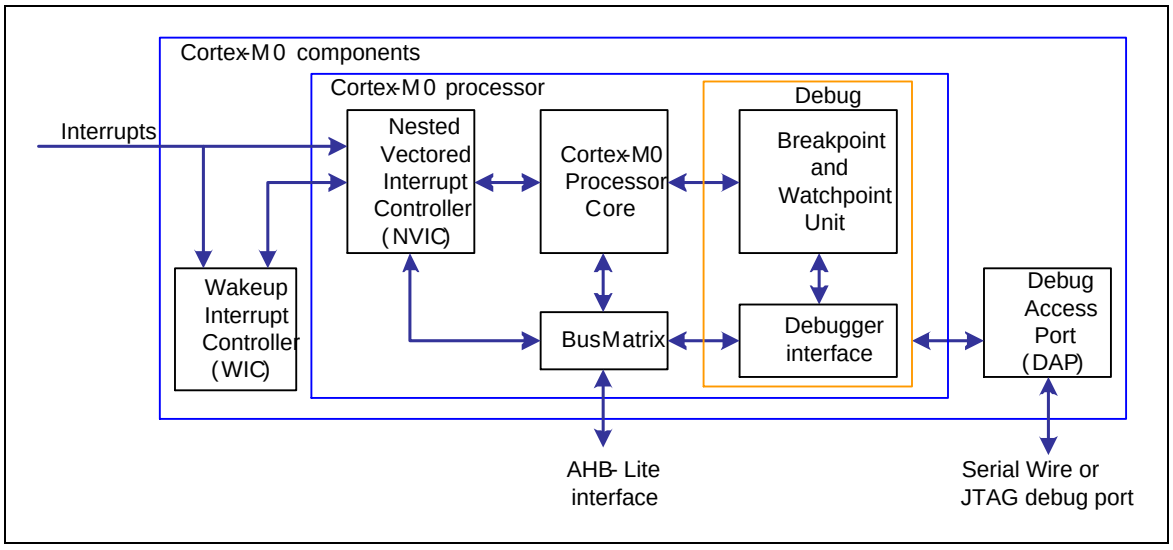



Figure 6-1 M0 Functional Block

## 6.2 Features

- A low gate count processor:
  - ARMv6-M Thumb® instruction set
  - Thumb-2 technology
  - ARMv6-M compliant 24-bit SysTick timer
  - A 32-bit hardware multiplier
  - Supports little-endian data accesses
  - Capable of deterministic, fixed-latency, interrupt handling
  - Load/store-multiples and multi-cycle-multiplies that can be abandoned and restarted to facilitate rapid interrupt handling
  - C Application Binary Interface compliant exception model. This is the ARMv6-M, C Application Binary Interface (C-ABI) compliant exception model that enables the use of pure C functions as interrupt handlers
  - Low Power Sleep mode entry using Wait For Interrupt (WFI), Wait For Event (WFE) instructions, or return from interrupt sleep-on-exit feature
- NVIC:
  - 32 external interrupt inputs, each with four levels of priority

- Dedicated Non-Maskable Interrupt (NMI) input
- Supports for both level-sensitive and pulse-sensitive interrupt lines
- Wake-up Interrupt Controller (WIC), providing Ultra-low Power Sleep mode support

- Debug support:
  - Four hardware breakpoints
  - Two watch points
  - Program Counter Sampling Register (PCSR) for non-intrusive code profiling
  - Single step and vector catch capabilities
- Bus interfaces:
  - Single 32-bit AMBA-3 AHB-Lite system interface providing simple integration to all system peripherals and memory
  - Single 32-bit slave port that supports the DAP (Debug Access Port)

## 6.3 System Timer (SysTick)

The Cortex-M0 includes an integrated system timer, SysTick. SysTick provides a simple, 24-bit

cleared-on-write, decrementing, wrap-on-zero counter with a flexible control mechanism. The counter can be used in several different ways, for example:

An RTOS tick timer which fires at a programmable rate (for example 100Hz) and invokes a SysTick routine.

A high speed alarm timer using Core clock.

A variable rate alarm or signal timer – the duration range dependent on the reference clock used and the dynamic range of the counter.

A simple counter. Software can use this to measure task completion time.

An internal Clock Source control based on missing/meeting durations. The COUNTFLAG bit-field in the control and status register can be used to determine if an action completed within a set duration, as part of a dynamic clock management control loop.

When enabled, the timer will count down from the value in the SysTick Current Value Register (SYST_CVR) to zero, and reload (wrap) to the value in the SysTick Reload Value Register (SYST_RVR) on the next clock edge, and then decrement on subsequent clocks. When the counter transitions to zero, the COUNTFLAG status bit is set. The COUNTFLAG bit clears on read.

The SYST_CVR value is UNKNOWN on reset. Software should write to the register to clear it to zero before enabling the feature. This ensures the timer will count from the SYST_RVR value rather than an arbitrary value when it is enabled.

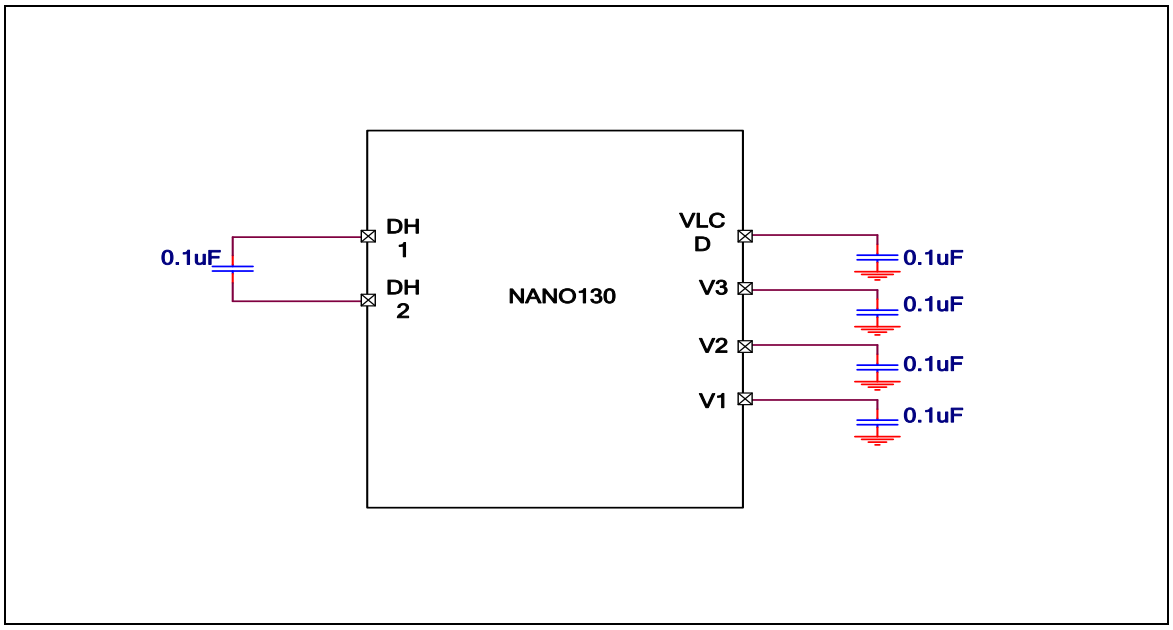
If the SYST_RVR is zero, the timer will be maintained with a current value of zero after it is reloaded with this value. This mechanism can be used to disable the feature independently from the timer enable bit.

For more detailed information, please refer to the “ARM® Cortex™-M0 Technical Reference Manual” and “ARM® v6-M Architecture Reference Manual”.

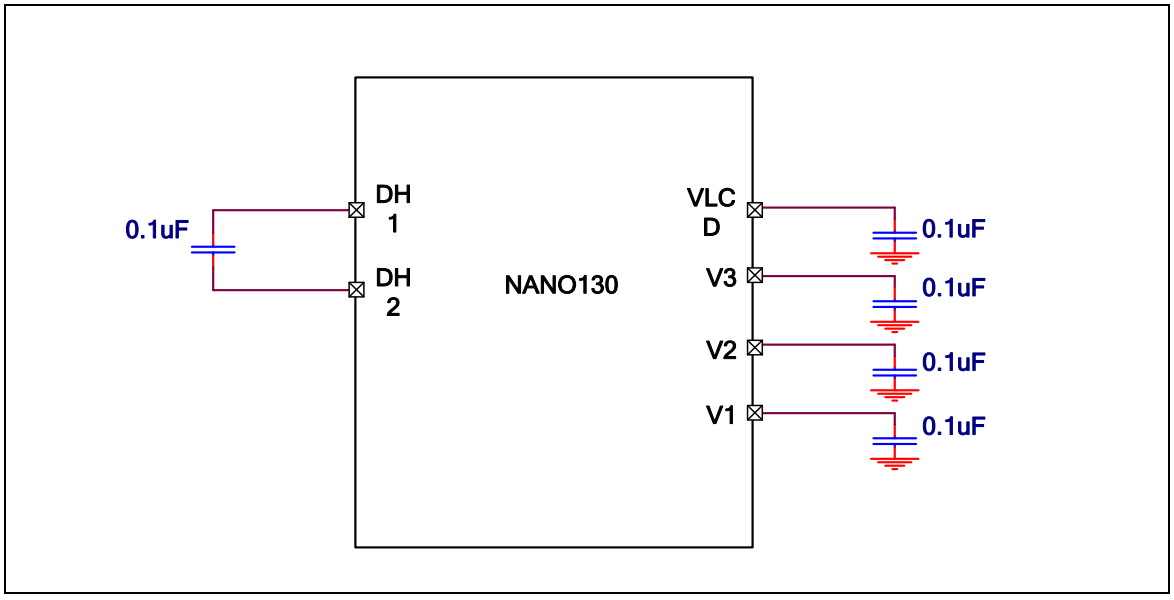
# 7 APPLICATION CIRCUIT

## 7.1 LCD Charge Pump

### 7.1.1 C-type 1/3 Bias

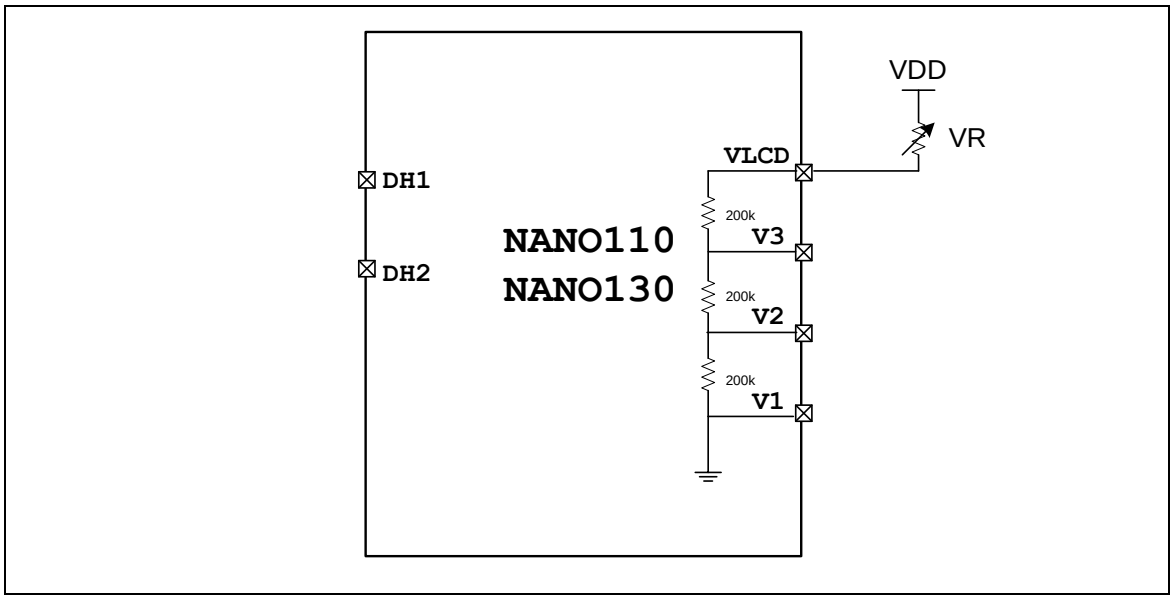



### 7.1.2 C-type 1/2 Bias



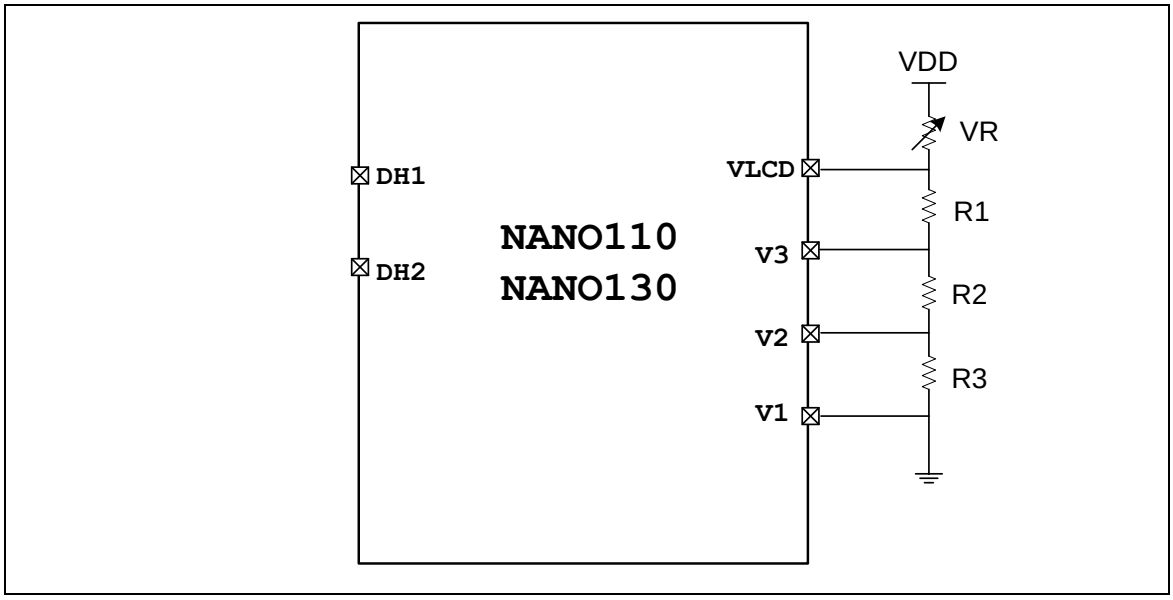

### 7.1.3 Internal R-type

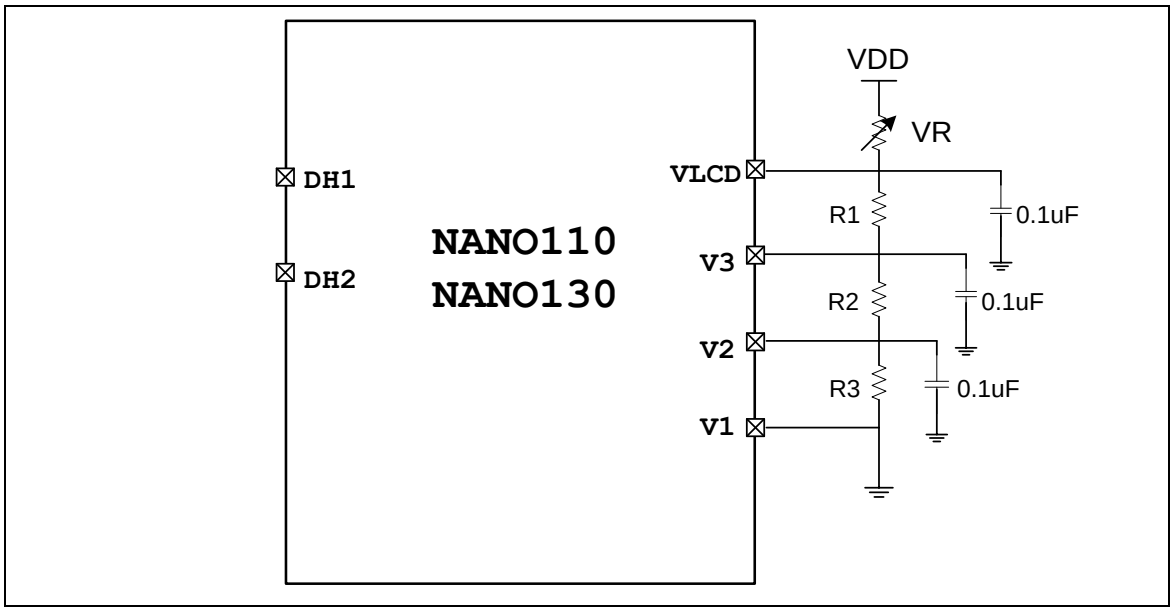
Nano110/130 series MCUs also support external R-type mode (bypass internal R) to reduce current consumption. For external R-type application, VLCD is normally connected to system VDD, or it can be connected to VDD through an external variable resistor (VR) which is used for adjusting LCD contrast.




### 7.1.4 External R-type

To reduce the current, the resistor ladder value can be increased. At some point, when the resistor ladder value is increased, the contrast will become affected and the waveform shape will be altered. Therefore, capacitors around 0.1uF should be chosen and place closed to resistor ladder based on the contrast and size of the pixels on the glass.

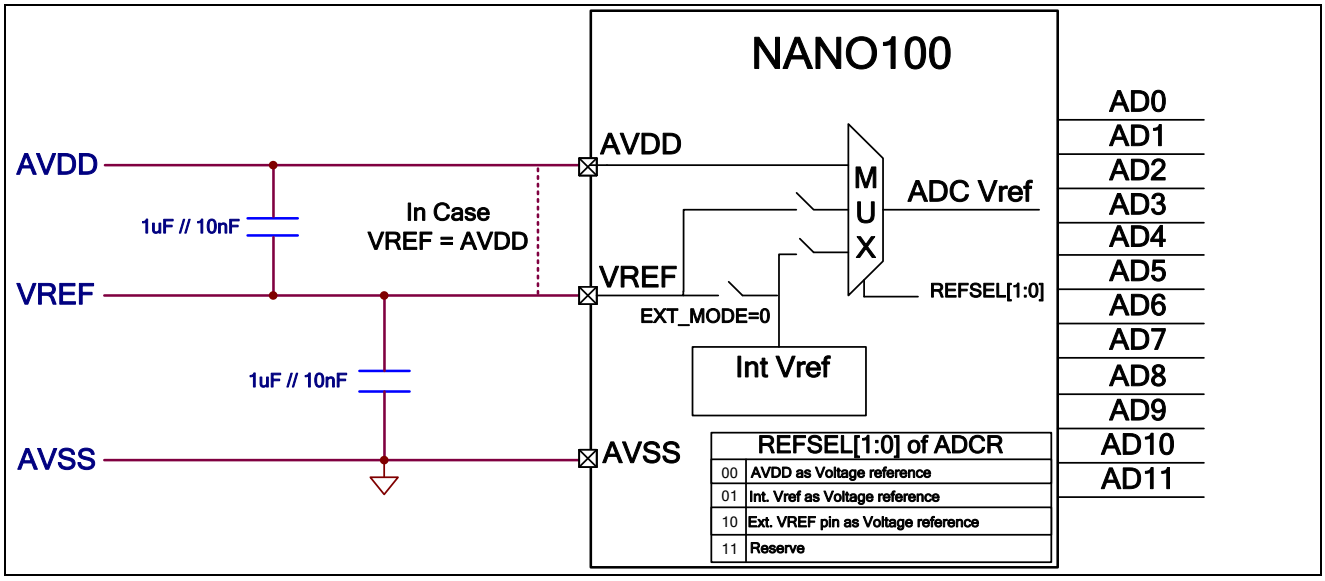




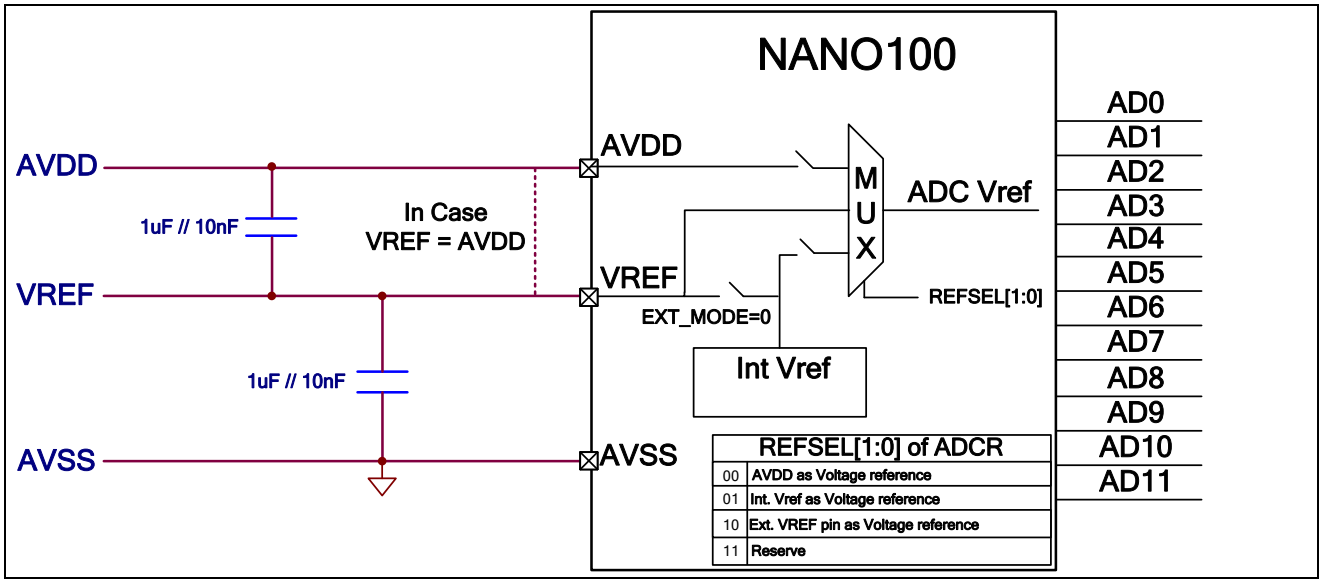

## 7.2 ADC Application Circuit

### 7.2.1 Voltage Reference Source

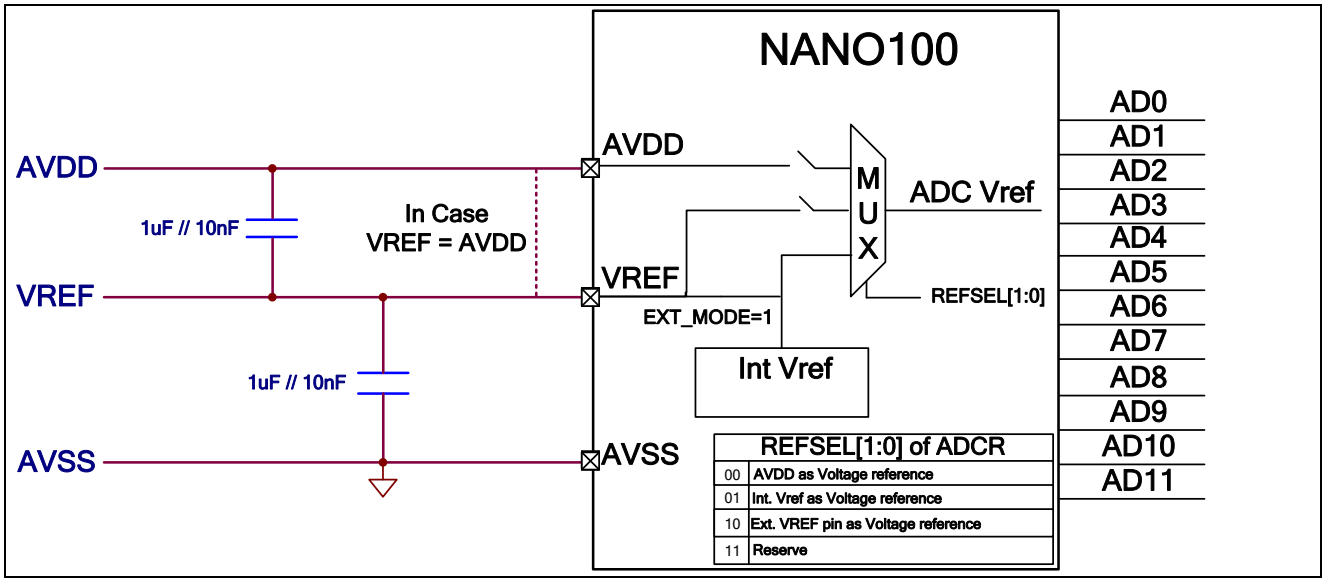
7.2.1.1 AVDD



7.2.1.2 Vref Pin



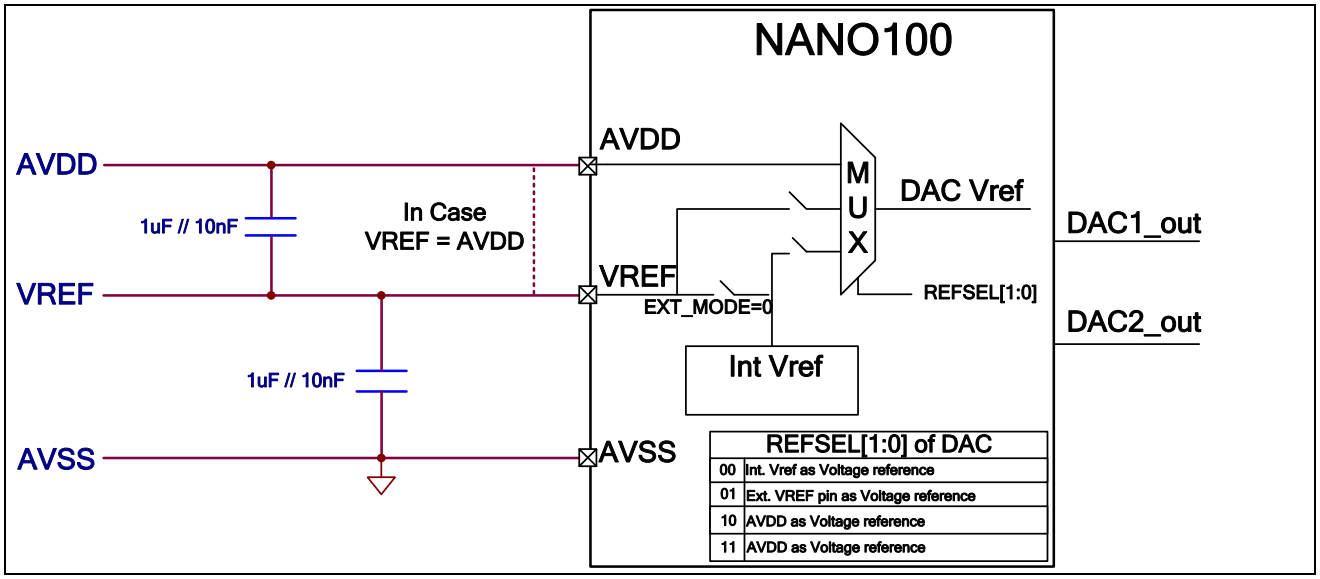

7.2.1.3 Int Vref



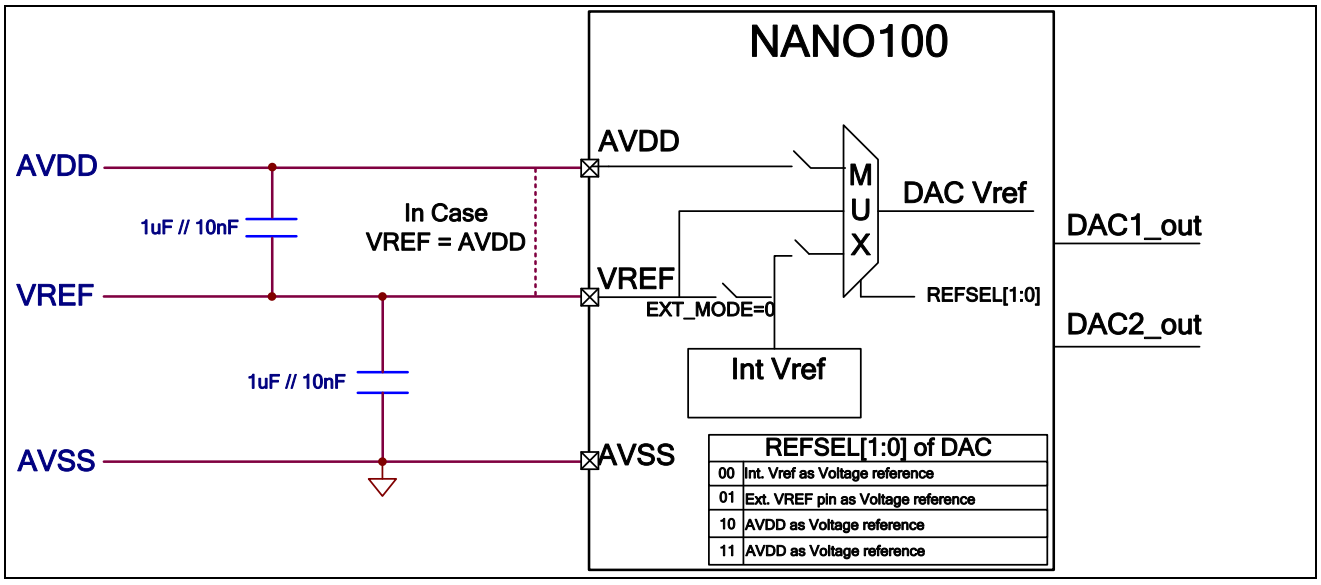
## 7.3 DAC Application Circuit

### 7.3.1 Voltage Reference Source

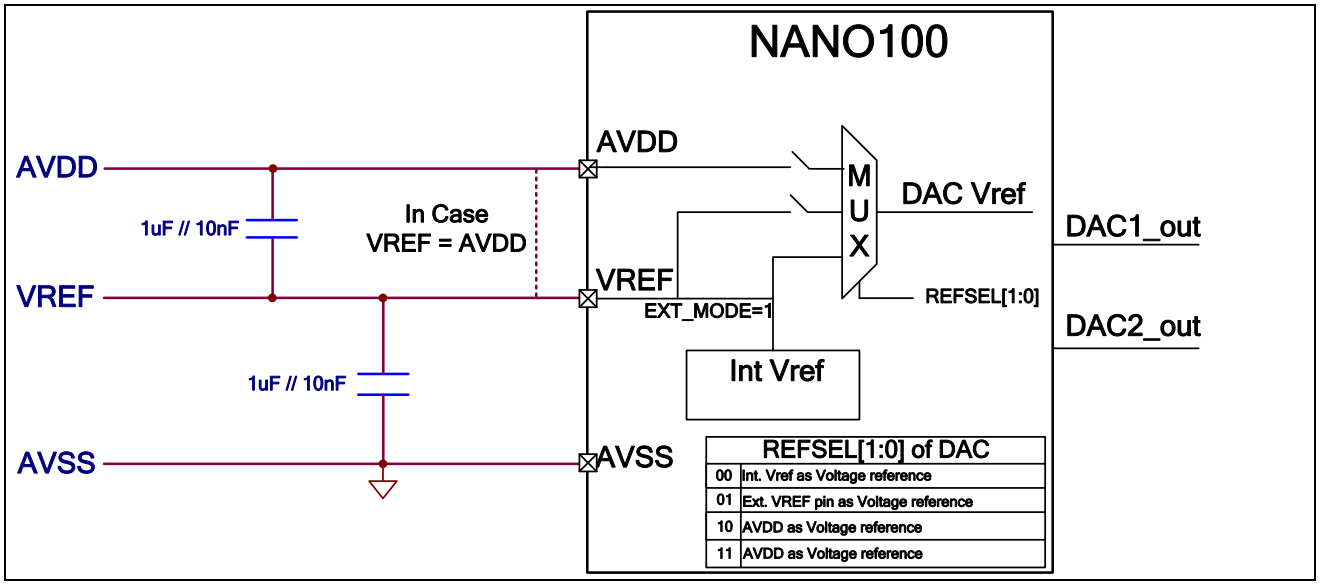
7.3.1.1 AVDD



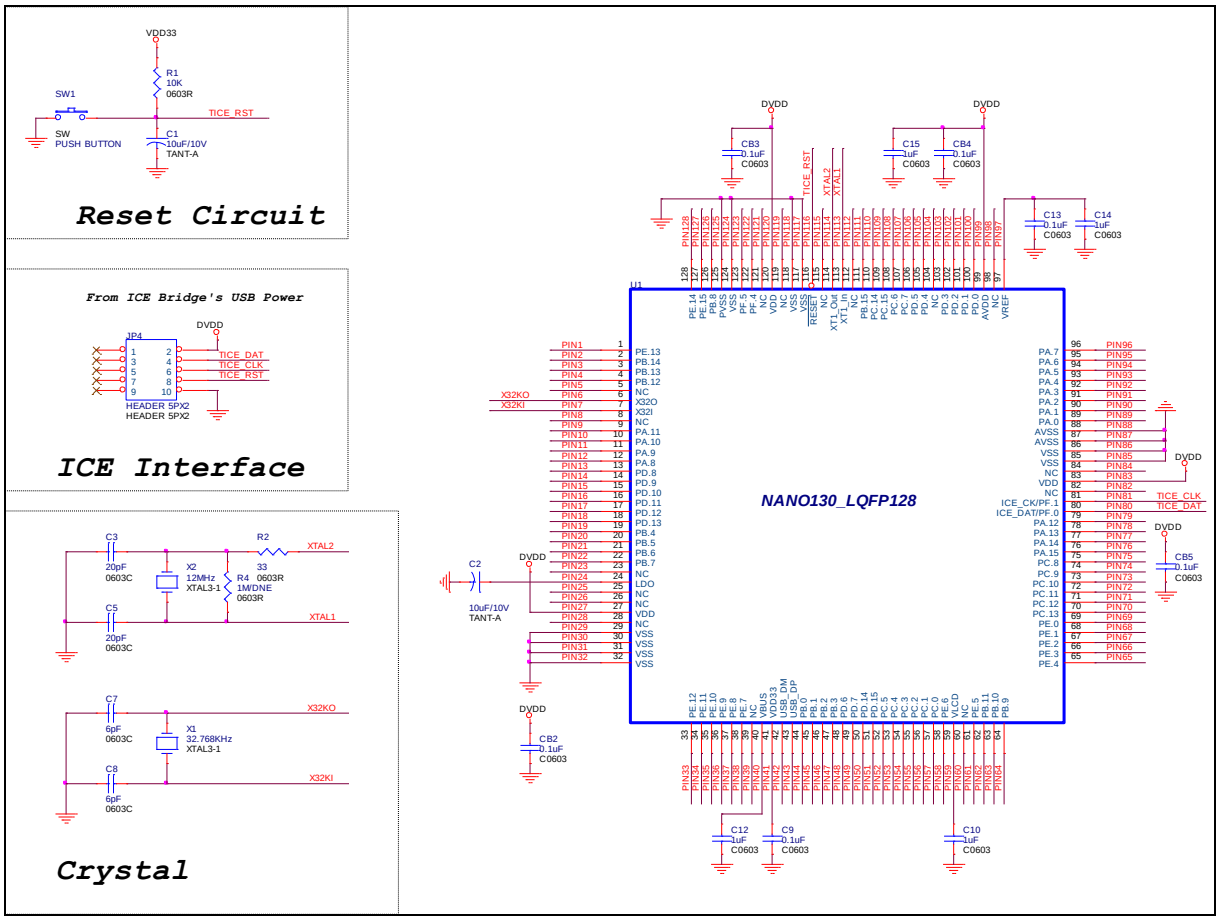

7.3.1.2 Vref Pin



7.3.1.3 Int Vref



## 7.4 Whole Chip Application Circuit



# 8 POWER COMSUMPTION

| Part No | Test Condition | | VDD | CPU Clock | Current |
|---|---|---|---|---|---|
| Nano100 (B) series<br>128 KB Flash<br>16 KB RAM | **Operating Mode:**<br>CPU run while(1) in FLASH ROM<br>Clock = 12 MHz Crystal Oscillator<br>Disable all peripherial | | 3.3V | 12 MHz | 2.41mA<br>200uA/MHz |
| | | | 1.8V | 12 MHz | N/A |
| | **Idle Mode:**<br>CPU stop<br>Clock = 12 MHz Crystal Oscillator<br>Disable all peripherial | | 3.3V | 12 MHz | 900uA<br>75uA/MHz |
| | | | 1.8V | 12 MHz | N/A |
| | **RTC + LCD Mode: (RAM retention)**<br>(Power down with 32K and LCD enabled)<br>CPU stop<br>Clock = 32.768 kHz Crystal Oscillator<br>Disable all peripherial except RTC and LCD circuit<br>Without panel loading | C-type | 3.3V | - | 10uA |
| | | Internal R-type<br>( With 200kΩ Resistor ladder ) | | | 8.5uA |
| | | External R-type<br>( With 1MΩ Resistor ladder ) | | | 4.5uA |
| | | C-type/R-type | 1.8V | - | N/A |
| | **RTC Mode: (RAM retention)**<br>(Power down with 32K enabled)<br>CPU stop<br>Clock = 32.768 kHz Crystal Oscillator<br>Disable all peripherial except RTC circuit | | 3.3V | - | 2.5uA |
| | | | 1.8V | - | 2.0uA |
| | **Power-down Mode: (RAM retention)**<br>CPU and all clocks stop | | 3.3V | - | 1uA |
| | | | 1.8V | - | 0.8uA |
| | **Wake-Up from Power-down Mode** | | 3.3V | 7us | N/A |

**Note:** Wake-up time: 7us from wake-up event to first CPU core valid clock; 10us from interrupt event to interrupt service routine first instruction.

# 9 ELECTRICAL CHARACTERISTIC

## 9.1 Absolute Maximum Ratings

| SYMBOL | PARAMETER | MIN | MAX | UNIT |
|---|---|---|---|---|
| DC Power Supply | $V_{DD}$–$V_{SS}$ | -0.3 | +4.0 | V |
| Input Voltage on 5V Tolerance Pin | $V_{IN}$ | $V_{SS}$ -0.3 | $V_{DD}$ +3.7 | V |
| Input Voltage on Any Other Pin without 5V Tolerance Pin | $V_{IN}$ | $V_{SS}$ -0.3 | $V_{DD}$ +0.3 | V |
| Oscillator Frequency | $1/t_{CLCL}$ | 4 | 24 | MHz |
| Operating Temperature | $T_A$ | -40 | +85 | °C |
| Storage Temperature | $T_{ST}$ | -55 | +150 | °C |
| Maximum Current into VDD | | - | 150 | mA |
| Maximum Current out of VSS | | - | 150 | mA |
| Maximum Current sunk by a I/O Pin | | - | 25 | mA |
| Maximum Current Sourced by a I/O Pin | | - | 25 | mA |
| Maximum Current Sunk by Total I/O Pins | | - | 100 | mA |
| Maximum Current Sourced by Total I/O Pins | | - | 100 | mA |

**Note :** Output voltage for ADC/LCD shared pins cannot be higher than VDD because these pins are without 5V tolerance.

(LQFP64 : LCD_SEG17, LCD_SEG19, LCD_SEG20, LCD_SEG21, LCD_SEG22, LCD_SEG23)

(LQFP128 : LCD_SEG36, LCD_SEG37, LCD_SEG38, LCD_SEG39)

## 9.2 Nano100/Nano110/Nano120/Nano130 DC Electrical Characteristics

(VDD-VSS=3.3V, TA = 25°C, FOSC = 32 MHz unless otherwise specified.)

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITIONS |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Operation voltage | $V_{DD}$ | 1.8 | - | 3.6 | V | $V_{DD}$ =1.8V up to 42 MHz |
| Power Ground | $V_{SS}$<br>$AV_{SS}$ | -0.3 | - | | V | |
| LDO Output Voltage | $V_{LDO1}$ | 1.62 | 1.8 | 1.98 | V | MCU operating in Run or Idle mode |
| | $V_{LDO2}$ | 1.49 | 1.66 | 1.83 | V | MCU operating in Power-down mode |
| Analog Operating Voltage | $AV_{DD}$ | | $V_{DD}$ | | V | |
| Reference Voltage | Vref | 1.8 | | $AV_{DD}$ | V | |
| Run Mode<br>at IRC = 12 MHz<br>Crystal Oscillator<br>Disable all peripherial | | | | | | CPU run while(1) in FLASH ROM<br>Clock = 12 MHz Crystal Oscillator<br>Disable all peripherial |
| Operating Current<br>Run Mode<br>at XTAL 12 MHz,<br>HCLK = 42 MHz | $I_{DD1}$ | | 20.5 | | mA | $V_{DD}$ = 3.6V at 42 MHz,<br>all IP and PLL enabled [*5] |
| | $I_{DD2}$ | | 10.6 | | mA | $V_{DD}$ = 3.6V at 42 MHz<br>all IP disabled and PLL enabled |
| | $I_{DD3}$ | | 19.1 | | mA | $V_{DD}$ = 1.8V at 42 MHz<br>all IP and PLL enabled [*5] |
| | $I_{DD4}$ | | 10.3 | | mA | $V_{DD}$ = 1.8V at 42 MHz<br>all IP disabled and PLL enabled |
| Operating Current<br>Run Mode<br>at XTAL 12 MHz,<br>HCLK = 32 MHz | $I_{DD5}$ | | 16.2 | | mA | $V_{DD}$ = 3.6V at 32 MHz,<br>all IP and PLL enabled [*5] |
| | $I_{DD6}$ | | 8.3 | | mA | $V_{DD}$ = 3.6V at 32 MHz<br>all IP disabled and PLL enabled |
| | $I_{DD7}$ | | 15.3 | | mA | $V_{DD}$ = 1.8V at 32 MHz<br>all IP and PLL enabled [*5] |

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITIONS |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| | $I_{DD8}$ | | 8.0 | | mA | $V_{DD}$ = 1.8V at 32 MHz<br>all IP disabled and PLL enabled |
| Operating Current<br>Run Mode<br>at XTAL 12 MHz,<br>HCLK = 12 MHz | $I_{DD9}$ | | 6.4 | | mA | $V_{DD}$ = 3.6V at 12 MHz,<br>all IP enabled and PLL disabled |
| | $I_{DD10}$ | | 2.8 | | mA | $V_{DD}$ = 3.6V at 12 MHz,<br>all IP and PLL disabled |
| | $I_{DD11}$ | | 6.3 | | mA | $V_{DD}$ = 1.8V at 12 MHz,<br>all IP enabled and PLL disabled |
| | $I_{DD12}$ | | 2.8 | | mA | $V_{DD}$ = 1.8V at 12 MHz,<br>all IP and PLL disabled |
| Operating Current<br>Run Mode<br>at IRC 12 MHz,<br>HCLK = 12 MHz | $I_{DD13}$ | | 6.7 | | mA | $V_{DD}$ = 3.6V at 12 MHz,<br>all IP enabled and PLL disabled |
| | $I_{DD14}$ | | 3.0 | | mA | $V_{DD}$ = 3.6V at 12 MHz,<br>all IP and PLL disabled |
| | $I_{DD15}$ | | 6.6 | | mA | $V_{DD}$ = 1.8V at 12 MHz,<br>all IP enabled and PLL disabled |
| | $I_{DD16}$ | | 3.0 | | mA | $V_{DD}$ = 1.8V at 12 MHz,<br>all IP and PLL disabled |
| Operating Current<br>Run Mode<br>at XTAL 4 MHz,<br>HCLK = 4 MHz | $I_{DD17}$ | | 3.3 | | mA | $V_{DD}$ = 3.6V at 4 MHz,<br>all IP enabled and PLL disabled |
| | $I_{DD18}$ | | 1.3 | | mA | $V_{DD}$ = 3.6V at 4 MHz,<br>all IP and PLL disabled |
| | $I_{DD19}$ | | 3.2 | | mA | $V_{DD}$ = 1.8V at 4 MHz,<br>all IP enabled and PLL disabled |
| | $I_{DD20}$ | | 1.3 | | mA | $V_{DD}$ = 1.8V at 4 MHz,<br>all IP and PLL disabled |
| Operating Current<br>Run Mode<br>at XTAL 32.768 kHz,<br>HCLK = 32.768 kHz | $I_{DD21}$ | | 82 | | uA | $V_{DD}$ = 3.6V at 32.768 kHz<br>all IP enabled and PLL disabled, |
| | $I_{DD22}$ | | 74 | | uA | $V_{DD}$ = 3.6V at 32.768 kHz<br>all IP and PLL disabled |
| | $I_{DD23}$ | | 77 | | uA | $V_{DD}$ = 1.8V at 32.768 kHz<br>all IP enabled and PLL disabled |

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITIONS |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| | $I_{DD24}$ | | 68 | | uA | $V_{DD}$ = 1.8V at 32.768 kHz<br>all IP and PLL disabled |
| Operating Current<br>Run Mode<br>at IRC 10 kHz,<br>HCLK = 10 kHz | $I_{DD25}$ | | 70 | | uA | $V_{DD}$ = 3.6V at 10 kHz<br>all IP enabled and PLL disabled |
| | $I_{DD26}$ | | 68 | | uA | $V_{DD}$ = 3.6V at 10 kHz<br>all IP and PLL disabled |
| | $I_{DD27}$ | | 65 | | uA | $V_{DD}$ = 1.8V at 10 kHz<br>all IP enabled and PLL disabled |
| | $I_{DD28}$ | | 62 | | uA | $V_{DD}$ = 1.8V at 10 kHz<br>all IP and PLL disabled |
| Operating Current<br>Idle Mode<br>at XTAL 12 MHz,<br>HCLK = 42 MHz | $I_{IDLE1}$ | | 14.5 | | mA | $V_{DD}$= 3.6V at 42 MHz<br>all IP and PLL enabled [*5] |
| | $I_{IDLE2}$ | | 4.6 | | mA | $V_{DD}$=3.6V at 42 MHz<br>all IP disabled and PLL enabled |
| | $I_{IDLE3}$ | | 13.8 | | mA | $V_{DD}$ = 1.8V at 42MHz<br>all IP and PLL enabled [*5] |
| | $I_{IDLE4}$ | | 4.5 | | mA | $V_{DD}$ = 1.8V at 42 MHz<br>all IP disabled and PLL enabled |
| Operating Current<br>Idle Mode<br>at XTAL 12 MHz,<br>HCLK = 32 MHz | $I_{IDLE5}$ | | 11.6 | | mA | $V_{DD}$= 3.6V at 32 MHz<br>all IP and PLL enabled [*5] |
| | $I_{IDLE6}$ | | 3.6 | | mA | $V_{DD}$=3.6V at 32 MHz<br>all IP disabled and PLL enabled |
| | $I_{IDLE7}$ | | 11.1 | | mA | $V_{DD}$ = 1.8V at 32MHz<br>all IP and PLL enabled [*5] |
| | $I_{IDLE8}$ | | 3.6 | | mA | $V_{DD}$ = 1.8V at 32 MHz<br>all IP disabled and PLL enabled |
| Operating Current<br>Idle Mode<br>at XTAL 12 MHz,<br>HCLK = 12 MHz | $I_{IDLE9}$ | | 4.7 | | mA | $V_{DD}$ = 3.6V at 12 MHz,<br>all IP enabled and PLL disabled |
| | $I_{IDLE10}$ | | 0.99 | | mA | $V_{DD}$ = 3.6V at 12 MHz,<br>all IP and PLL disabled |
| | $I_{IDLE11}$ | | 4.6 | | mA | $V_{DD}$ = 1.8V at 12 MHz,<br>all IP enabled and PLL disabled |
| | $I_{IDLE12}$ | | 0.94 | | mA | $V_{DD}$ = 1.8V at 12 MHz,<br>all IP and PLL disabled |

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITIONS |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Operating Current Idle Mode at IRC 12 MHz, HCLK = 12 MHz | $I_{IDLE13}$ | | 5.9 | | mA | $V_{DD}$ = 3.6V at 12 MHz, all IP enabled and PLL disabled |
| | $I_{IDLE14}$ | | 1.3 | | mA | $V_{DD}$ = 3.6V at 12 MHz, all IP and PLL disabled |
| | $I_{IDLE15}$ | | 4.9 | | mA | $V_{DD}$ = 1.8V at 12 MHz, all IP enabled and PLL disabled |
| | $I_{IDLE16}$ | | 1.3 | | mA | $V_{DD}$ = 1.8V at 12 MHz, all IP and PLL disabled |
| Operating Current Idle Mode at XTAL 4 MHz, HCLK = 4 MHz | $I_{IDLE17}$ | | 2.7 | | mA | $V_{DD}$ = 3.6V at 4 MHz, all IP enabled and PLL disabled |
| | $I_{IDLE18}$ | | 0.66 | | mA | $V_{DD}$ = 3.6V at 4 MHz, all IP and PLL disabled |
| | $I_{IDLE19}$ | | 2.7 | | mA | $V_{DD}$ = 1.8V at 4 MHz, all IP enabled and PLL disabled |
| | $I_{IDLE20}$ | | 0.64 | | mA | $V_{DD}$ = 1.8V at 4 MHz, all IP and PLL disabled |
| Operating Current Idle Mode at XTAL 32.768 kHz, HCLK = 32.768 kHz | $I_{IDLE21}$ | | 78 | | uA | $V_{DD}$ = 3.6V at 32.768 kHz all IP enabled and PLL disabled |
| | $I_{IDLE22}$ | | 69 | | uA | $V_{DD}$ = 3.6V at 32.768 kHz all IP and PLL disabled |
| | $I_{IDLE23}$ | | 72 | | uA | $V_{DD}$ = 1.8V at 32.768 kHz all IP enabled and PLL disabled |
| | $I_{IDLE24}$ | | 63 | | uA | $V_{DD}$ = 1.8V at 32.768 kHz all IP and PLL disabled |
| Operating Current Idle Mode at IRC 10 kHz, HCLK = 10 kHz | $I_{IDLE25}$ | | 69 | | uA | $V_{DD}$ = 3.6V at 10 kHz all IP enabled and PLL disabled |
| | $I_{IDLE26}$ | | 66 | | uA | $V_{DD}$ = 3.6V at 10 kHz all IP and PLL disabled |
| | $I_{IDLE27}$ | | 63 | | uA | $V_{DD}$ = 1.8V at 10 kHz all IP enabled and PLL disabled |
| | $I_{IDLE28}$ | | 61 | | uA | $V_{DD}$ = 1.8V at 10 kHz all IP and PLL disabled |

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITIONS |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Standby Current<br>Power-down Mode | $I_{PWD1}$ | | 1.2 | | μA | $V_{DD}$ = 3.6V, RTC OFF, all clock stop<br>With RAM Retenstion, IO no loading |
| | $I_{PWD2}$ | | 0.8 | | μA | $V_{DD}$ = 1.8V, RTC OFF, all clock stop<br>With RAM Retenstion, IO no loading |
| | $I_{PWD3}$ | | 2.8 | | μA | $V_{DD}$ = 3.6V, RTC ON, all clock stop except 32.768 kHz<br>With RAM Retenstion, IO no loading |
| | $I_{PWD4}$ | | 2.0 | | μA | $V_{DD}$ = 1.8V, RTC ON, all clock stop except 32.768 kHz<br>With RAM Retenstion, IO no loading |
| Input Pull Up Resistor PA, PB, PC, PD, PE, PF | $R_{IN}$ | | 40 | | KΩ | $V_{DD}$ = 3.3V |
| | | | 98 | | KΩ | $V_{DD}$ = 1.8V |
| Input Leakage Current PA, PB, PC, PD, PE, PF | $I_{LK}$ | -0.1 | - | +0.1 | μA | $V_{DD}$ = 3.3V, 0<$V_{IN}$<$V_{DD}$ |
| Input Low Voltage PA, PB, PC, PD, PE, PF<br>(Schmitt input) | $V_{IL1}$ | | - | $0.4V_{DD}$ | V | |
| Input High Voltage PA, PB, PC, PD, PE, PF<br>(Schmitt input) | $V_{IH1}$ | $0.6V_{DD}$ | | 5.5 | V | ADC and DAC shared pins without Input 5V tolerance. |
| Hysteresis voltage of PA~PF (Schmitt input) | $V_{HY}$ | | $0.2V_{DD}$ | | V | |
| Input Low Voltage XT1_IN / XT1_OUT [*2] | $V_{IL2}$ | 0 | - | 0.4 | | $V_{DD}$ = 1.8V |
| Input High Voltage XT1_IN / XT1_OUT [*2] | $V_{IH2}$ | 1.5 | - | $V_{DD}$ +0.2 | V | $V_{DD}$ = 1.8V |
| Input Low Voltage X32I / X32O [*2] | $V_{IL4}$ | 0 | - | 0.3 | V | |
| Input High Voltage X32I / X32O [*2] | $V_{IH4}$ | 1.5 | - | 1.98 | V | |
| Negative going threshold<br>(Schmitt input), /RESET | $V_{ILS}$ | 1.28 | 1.33 | 1.37 | V | $V_{DD}$ = 3.3V |
| Positive going threshold<br>(SchmittIput), /RESET | $V_{IHS}$ | 1.75 | 1.98 | 2.25 | V | $V_{DD}$ = 3.3V |

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITIONS |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Source Current PA, PB, PC, PD, PE, PF (Push-pull Mode) | $I_{SR21}$ | -10 | -14 | - | mA | $V_{DD}$ = 3.3V, $V_S$ = Vdd-0.7V |
| | $I_{SR22}$ | -3 | -5 | - | mA | $V_{DD}$ = 1.8V, $V_S$ = Vdd-0.45V |
| Sink Current PA, PB, PC, PD, PE, PF (Push-pull Mode) | $I_{SK21}$ | 10 | 15 | - | mA | $V_{DD}$ = 3.3V, $V_S$ = 0.7V |
| | $I_{SK22}$ | 3 | 6 | - | mA | $V_{DD}$ = 1.8V, $V_S$ = 0.45V |

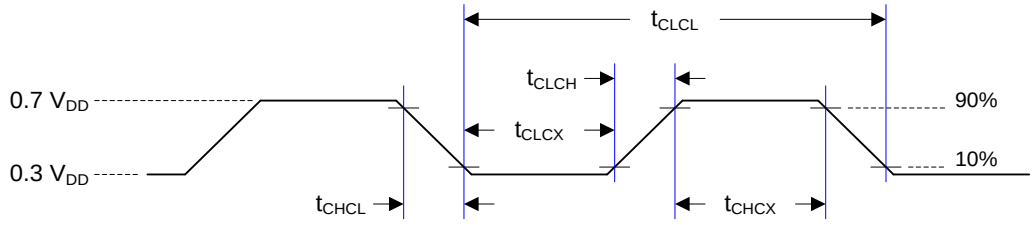
**Note:**

1. /RESET pin is a Schmitt trigger input.
2. Crystal Input is a CMOS input.
3. It is recommended that a 10uF or higher capacitor and a 100nF bypass capacitor are connected between VDD and the closest VSS pin of the device.
4. For ensuring power stability, a 4.7uF or higher capacitor must be connected between LDO pin and the closest VSS pin of the device. Also a 100nF bypass capacitor between LDO and VSS help suppressing output noise.
5. All peripherals' clock source is from HXT (12 MHz), except SPI from HCLK.

## 9.3 AC Electrical Characteristics

### 9.3.1 External Input Clock

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITION |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Clock High Time | $t_{CHCX}$ | 10 | - | | nS | |
| Clock Low Time | $t_{CLCX}$ | 10 | - | | nS | |
| Clock Rise Time | $t_{CLCH}$ | 2 | - | 15 | nS | |
| Clock Fall Time | $t_{CHCL}$ | 2 | - | 15 | nS | |




**Note:** Duty cycle is 50%.

### 9.3.2 External 4~24 MHz XTAL Oscillator

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITION |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Oscillator frequency | $f_{HXT}$ | 4 | 12 | 24 | MHz | VDD = 1.8V ~ 3.6V |
| Temperature | $T_{HXT}$ | -40 | - | +85 | $^{o}C$ | |
| Operating current | $I_{HXT}$ | | 0.3 | | mA | VDD = 3.0V |

#### 9.3.2.1 Typical Crystal Application Circuits

| CRYSTAL | C1 | C2 | R |
|---|---|---|---|
| 4MHz ~ 24 MHz | Optional(Depend on crystal specification) | | without |

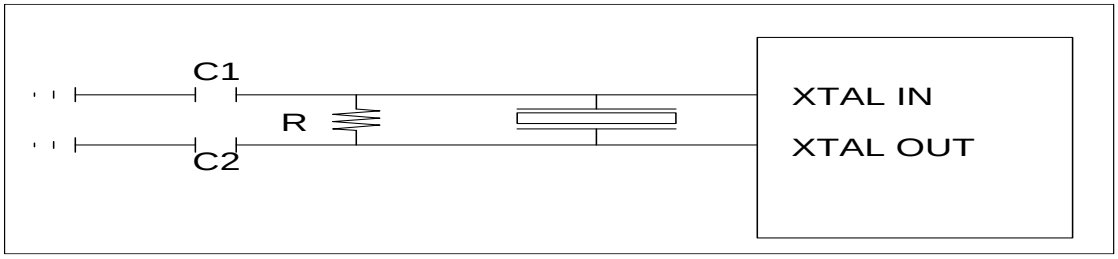



Figure 9-1 Typical Crystal Application Circuit

### 9.3.3 External 32.768 kHz Crystal

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITION |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Oscillator frequency | $f_{LXT}$ | | 32.768 | | kHz | VDD = 1.8V ~ 3.6V |
| Temperature | $T_{LXT}$ | -40 | - | +85 | $^{o}$C | |
| Operating current | $I_{LXT}$ | | 1.2 | | μA | VDD = 3.0V |

### 9.3.4 Internal 12 MHz Oscillator

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITION |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Supply voltage[1] | $V_{HRC}$ | | 1.8 | | V | |
| Calibrated Internal Oscillator Frequency | $F_{HRC}$ | 11.88 | 12 | 12.12 | MHz | 25$^{o}$C, $V_{DD}$ = 3V |
| | | 11.76 | 12 | 12.24 | MHz | -40$^{o}$C~+85 $^{o}$C, $V_{DD}$ = 1.8V~3.6V |
| | | 11.97 | 12 | 12.03 | MHz | -40$^{o}$C~+85 $^{o}$C, $V_{DD}$ = 1.8V~3.6V Enable 32.768K crystal oscillator and set TRIM_SEL[1:0]=”10” |
| Operating current | $I_{HRC}$ | | 450 | | μA | |

**Note:** Internal oscillator operation voltage comes from LDO.

### 9.3.5 Internal 10 kHz Oscillator

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITION |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Supply voltage[1] | $V_{LRC}$ | | 1.8 | | V | |
| Center Frequency | $F_{LRC}$ | 7 | 10 | 13 | kHz | 25$^{o}$C, $V_{DD}$ = 3V |
| | | 5 | 10 | 15 | kμHz | -40$^{o}$C~+85 $^{o}$C, $V_{DD}$ = 1.8V~3.6V |
| Operating current | $I_{LRC}$ | | 0.7 | | μA | $V_{DD}$ = 3V |

**Note:** Internal oscillator operation voltage comes from LDO.

## 9.4 Analog Characteristics

### 9.4.1 12-bit ADC

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITION |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Operating voltage | $AV_{DD}$ | 1.8 | | 3.6 | V | $AV_{DD} = V_{DD}$ |
| Operating current | $I_{ADC42}$ | | 147 | | μA | $AV_{DD} = V_{DD} = 3.0V$<br>$ADC_VREF = AV_{DD}$<br>ADC Clock Rate = 42 MHz |
| | $I_{ADC12}$ | | 50 | | μA | $AV_{DD} = V_{DD} = 3.0V$<br>$ADC_VREF = AV_{DD}$<br>ADC Clock Rate = 12 MHz |
| Resolution | $R_{ADC}$ | | | 12 | Bit | |
| Reference voltage | $V_{REF}$ | 1.8 | | $A_{VDD}$ | V | |
| Reference input current (Avg.) | $I_{REF}$ | | | 10 | μA | |
| ADC input voltage | $V_{IN}$ | 0 | | $V_{REF}$ | V | |
| Conversion time | $T_{CONV}$ | 0.5 | | | μS | |
| Sampling Rate | $F_{SPS}$ | | | 2M | Hz | $V_{DD}$ = 3V |
| Integral Non-Linearity Error | INL | | ±1 | ±2 | LSB | $V_{REF}$ is external Vref pin |
| Differential Non-Linearity | DNL | | ±0.8 | -1~+1.5 | LSB | $V_{REF}$ is external Vref pin |
| Gain error | $E_G$ | | - | ±2 | LSB | $V_{REF}$ is external Vref pin |
| Offset error | $E_{OFFSET}$ | | - | ±3 | LSB | $V_{REF}$ is external Vref pin |
| Absolute error | $E_{ABS}$ | | - | ±6 | LSB | $V_{REF}$ is external Vref pin |
| ADC Clock frequency | $F_{ADC}$ | 0.25 | | 42 | MHz | |
| Clock cycle | $AD_{CYC}$ | 20 | | | Cycle | |
| Internal Capacitance | $C_{IN}$ | - | 5 | - | pF | |
| Monotonic | - | Guaranteed | | | - | |

### 9.4.2 Brown-out Detector

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITION |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Operating voltage | $V_{BOD}$ | 1.8 | | 3.6 | V | |
| BOD17 Quiescent current | $I_{BOD17}$ | | 1 | | μA | $AV_{DD}$ = 3.0V, BOD17 enabled |
| BOD20 Quiescent current | $I_{BOD20}$ | | 1 | | μA | $AV_{DD}$ = 3.0V, BOD20 enabled |

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITION |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| BOD25 Quiescent current | $I_{BOD25}$ | | 1 | | μA | $AV_{DD}$ = 3.0V, BOD25 enabled |
| BOD17 detection level | $V_{B17dt}$ | 1.6 | 1.7 | 1.8 | V | 25°C |
| BOD20 detection level | $V_{B20dt}$ | 1.9 | 2.0 | 2.1 | V | 25°C |
| BOD25 detection level | $V_{B25dt}$ | 2.4 | 2.5 | 2.6 | V | 25°C |

### 9.4.3 Power-on Reset

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITION |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Reset voltage | $V_{POR}$ | - | 1.6 | - | V | |
| Quiescent current | $I_{POR}$ | - | 1 | - | nA | LDO output > Reset voltage |

### 9.4.4 Temperature Sensor

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITION (SUPPLY VOLTAGE = 3.36V) |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Detection Temperature | $T_{DET}$ | -40 | | +110 | °C | |
| Operating current | $I_{TEMP}$ | - | 5 | - | μA | |
| Gain | $V_{TG}$ | -1.80 | -1.73 | -1.65 | mV/°C | |
| Offset | $V_{TO}$ | 730 | 740 | 750 | mV | Tempeature at 0 °C |

**Note:** Internal operation voltage comes form LDO.

### 9.4.5 12-bit DAC

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITION |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Operating voltage | $AV_{DD}$ | 2.0 | | 3.6 | V | $AV_{DD}$ = $V_{DD}$ |
| Operating current | $I_{DAC}$ | | 2.20 | | mA | $AV_{DD}$ = $V_{DD}$ = 3.0V,<br>DAC_VREF = $AV_{DD}$<br>DAC conversion rate 500kHz |
| Resolution | $R_{ADC}$ | | | 12 | Bit | |
| Reference voltage | $V_{REF}$ | 1.8 | | $A_{VDD}$ | V | |

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITION |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Reference input current (Avg.) | $I_{REF}$ | | 0.85 | | mA | $AV_{DD}$ = $V_{DD}$ = 3.0V<br>DAC_VREF=Ext_Vref<br>DAC conversion rate 500kHz |
| DAC output swin range | $V_{OUT}$ | 0.1 x $V_{REF}$ | - | 0.9 x $V_{REF}$ | V | |
| Conversion Rate (code to adjacent code) | $F_{SPS}$ | | | 500 | kHz | $V_{DD}$ = 3V |
| Integral Non-Linearity Error | INL | | ±4 | ±5 | LSB | $V_{REF}$ is external Vref pin<br>Not include offset and gain error |
| Differential Non-Linearity | DNL | | ±1 | ±2 | LSB | $V_{REF}$ is external Vref pin<br>Not include offset and gain error |
| Gain error | $E_G$ | | 290 | | LSB | |
| Offset error | $E_{OFFSET}$ | | 150 | | LSB | |

### 9.4.6 LCD

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITION |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Operating voltage | $V_{DD}$ | 1.8 | - | 3.6 | V | |
| VLCD voltage | $V_{LCD34}$ | - | 3.4 | - | V | CPUMP_VOL_SET=111, no loading |
| VLCD voltage | $V_{LCD33}$ | - | 3.3 | - | V | CPUMP_VOL_SET=110, no loading |
| VLCD voltage | $V_{LCD32}$ | - | 3.2 | - | V | CPUMP_VOL_SET=101, no loading |
| VLCD voltage | $V_{LCD31}$ | - | 3.1 | - | V | CPUMP_VOL_SET=100, no loading |
| VLCD voltage | $V_{LCD30}$ | - | 3.0 | - | V | CPUMP_VOL_SET=011, no loading |
| VLCD voltage | $V_{LCD29}$ | - | 2.9 | - | V | CPUMP_VOL_SET=010, no loading |
| VLCD voltage | $V_{LCD28}$ | - | 2.8 | - | V | CPUMP_VOL_SET=001, no loading |
| VLCD voltage | $V_{LCD27}$ | - | 2.7 | - | V | CPUMP_VOL_SET=000, no loading |
| Operating current | $I_{LCD}$ | - | 10 | - | μA | $V_{DD}$ = 3V, frame rate = 32Hz<br>Without loading |

### 9.4.7 Internal Voltage Reference

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITION |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| Operating voltage | $AV_{DD}$ | 1.8 | - | 3.6 | V | |
| 1.8V voltage reference | $V_{REF1}$ | 1.69 | 1.8 | 1.87 | V | $AV_{DD}$ ≥ 2.0V (-40°C ~85°C) |

| PARAMETER | SYM. | SPECIFICATIONS | | | | TEST CONDITION |
|---|---|---|---|---|---|---|
| | | MIN. | TYP. | MAX. | UNIT | |
| 2.5V voltage reference | $V_{REF2}$ | 2.35 | 2.5 | 2.60 | V | $AV_{DD}$ ≥ 2.8V (-40°C ~85°C) |
| Stable Time | $T_{REFTAB}$ | - | 1 | - | ms | |
| Operating current | $I_{VREF}$ | - | 30 | - | μA | $AV_{DD}$ = 3V |

### 9.4.8 USB PHY Specifications

#### 9.4.8.1 USB PHY DC Electrical Characteristics

| SYMBOL | PARAMETER | CONDITION | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| $V_{IH}$ | Input high (driven) | | 2.0 | - | | V |
| $V_{IL}$ | Input low | | | - | 0.8 | V |
| $V_{DI}$ | Differential input sensitivity | \|PADP-PADM\| | 0.2 | - | | V |
| $V_{CM}$ | Differential common-mode range | Includes $V_{DI}$ range | 0.8 | - | 2.5 | V |
| $V_{SE}$ | Single-ended receiver threshold | | 0.8 | - | 2.0 | V |
| | Receiver hysteresis | | | 200 | | mV |
| $V_{OL}$ | Output low (driven) | | 0 | - | 0.3 | V |
| $V_{OH}$ | Output high (driven) | | 2.8 | - | 3.6 | V |
| $V_{CRS}$ | Output signal cross voltage | | 1.3 | - | 2.0 | V |
| $R_{PU}$ | Pull-up resistor | | 1.425 | - | 1.575 | kΩ |
| $R_{PD}$ | Pull-down resistor | | 14.25 | - | 15.75 | kΩ |
| $V_{TRM}$ | Termination Voltage for upstream port pull up (RPU) | | 3.0 | - | 3.6 | V |
| $Z_{DRV}$ | Driver output resistance | Steady state drive* | | 10 | | Ω |
| $C_{IN}$ | Transceiver capacitance | Pin to GND | | - | 20 | pF |

*Driver output resistance doesn't include series resistor resistance.

#### 9.4.8.2 USB PHY Full-Speed Driver Elevtrical Characteristics

| SYMBOL | PARAMETER | CONDITION | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| $T_{FR}$ | Rise Time | $C_L$=50p | 4 | - | 20 | ns |
| $T_{FF}$ | Fall Time | $C_L$=50p | 4 | - | 20 | ns |
| $T_{FRFF}$ | Rise and fall time matching | $T_{FRFF}=T_{FR}/T_{FF}$ | 90 | - | 111.11 | % |

#### 9.4.8.3 USB PHY Power Dissipation

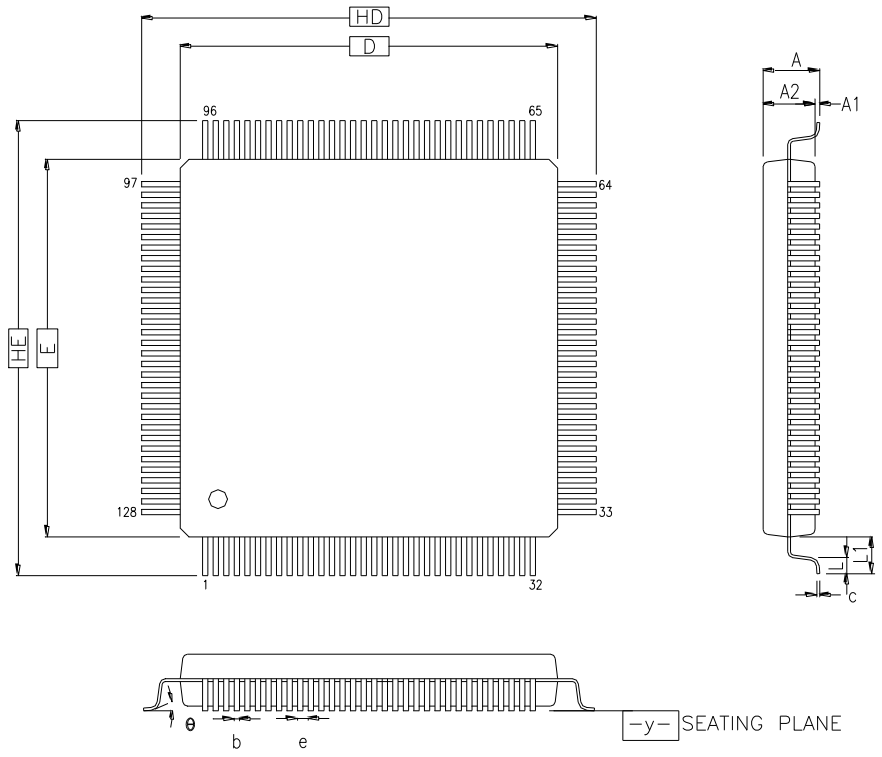
| SYMBOL | PARAMETER | CONDITION | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| $I_{VDDREG}$ (Full Speed) | VDDD and VDDREG Supply Current (Steady State) | Standby | | 50 | | uA |

#### 9.4.8.4 USB LDO DC Electrical Characteristics

| SYMBOL | PARAMETER | CONDITION | MIN. | TYP. | MAX. | UNIT |
|---|---|---|---|---|---|---|
| VBUS | | | | 5 | | V |
| V33 | Output voltage | VBUS = 5V, 25°C | 2.97 | 3.3 | 3.63 | V |
| Iop | Operation Current | | | 100 | | uA |

# 10 PACKAGE DIMENSIONS

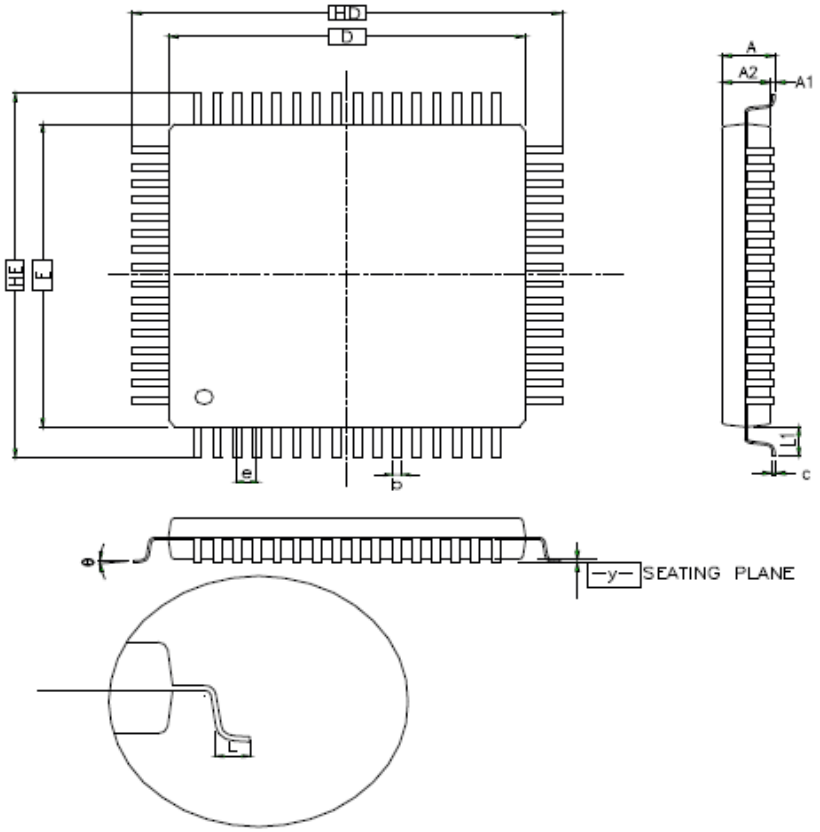
## 10.1 LQFP128 (14x14x1.4 mm footprint 2.0 mm)




COTROL DIMENSIONS ARE IN MILLIMETERS.

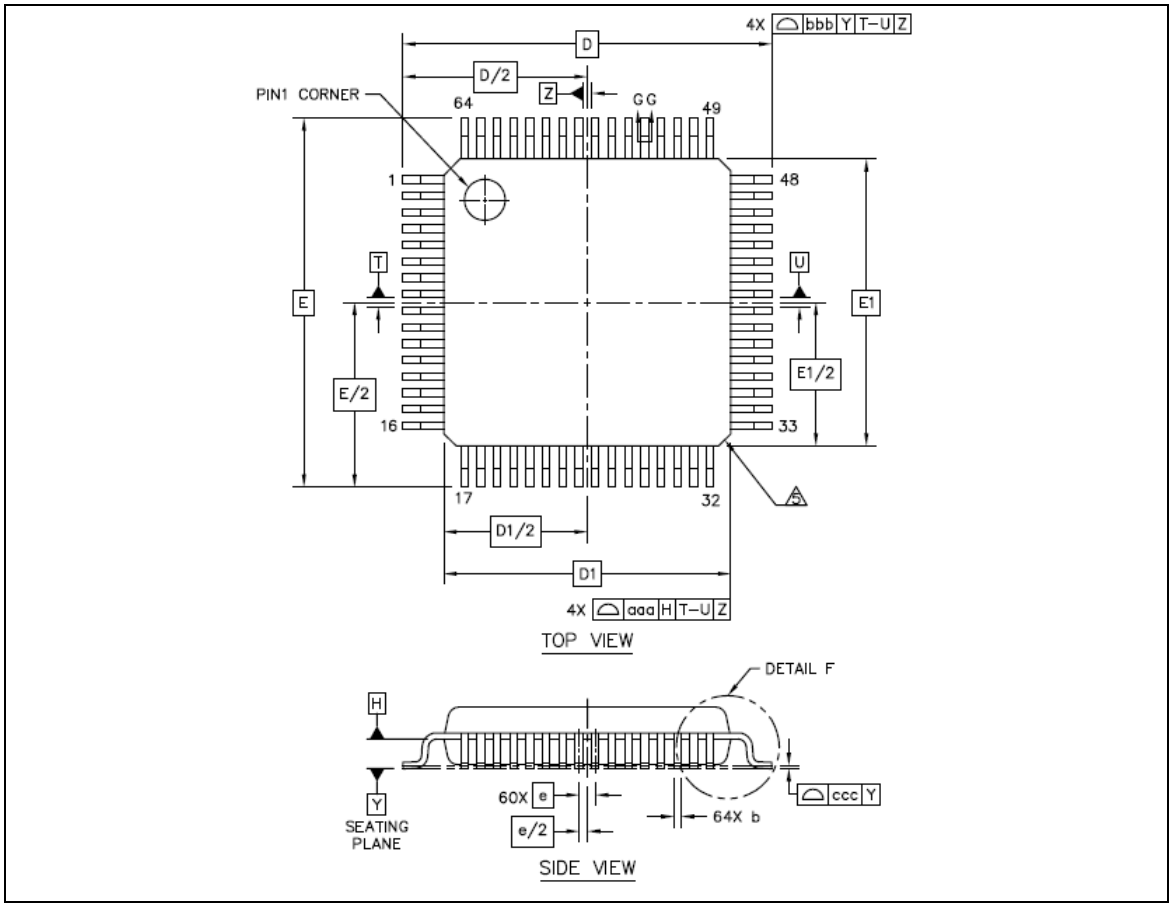
| SYMBOL | MILLIMETER | | | INCH | | |
|---|---|---|---|---|---|---|
| | MIN. | NOM. | MAX. | MIN. | NOM. | MAX. |
| A | — | — | 1.60 | — | — | 0.063 |
| A1 | 0.05 | — | 0.15 | 0.002 | — | 0.006 |
| A2 | 1.35 | 1.40 | 1.45 | 0.053 | 0.055 | 0.057 |
| HD | 16.00 BSC. | | | 0.630 BSC. | | |
| D | 14.00 BSC. | | | 0.551 BSC. | | |
| HE | 16.00 BSC. | | | 0.630 BSC. | | |
| E | 14.00 BSC. | | | 0.551 BSC. | | |
| b | 0.13 | 0.16 | 0.23 | 0.005 | 0.006 | 0.009 |
| e | 0.40 BSC. | | | 0.016 BSC. | | |
| θ | 0° | 3.5° | 7° | 0° | 3.5° | 7° |
| c | 0.09 | — | 0.20 | 0.004 | — | 0.008 |
| L | 0.45 | 0.60 | 0.75 | 0.018 | 0.024 | 0.030 |
| $L_1$ | 1.00 REF | | | 0.039 REF | | |
| y | – | – | 0.1 | – | – | 0.004 |

## 10.2 LQFP64 (10x10x1.4 mm footprint 2.0 mm)

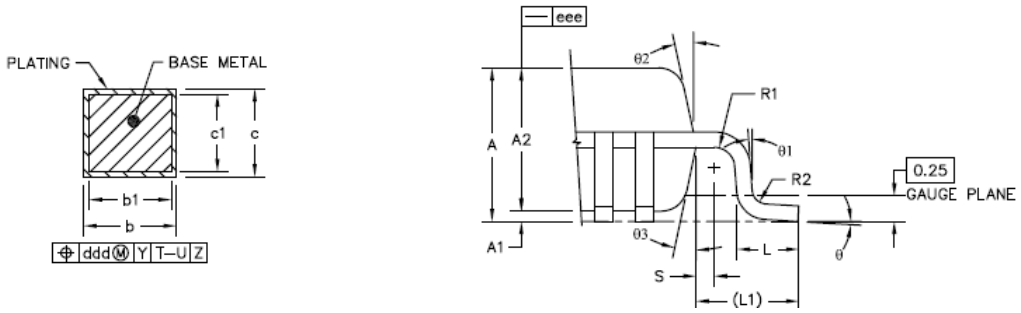



| Symbol | Dimension in inch | | | Dimension in mm | | |
|---|---|---|---|---|---|---|
| | Min | Nom | Max | Min | Nom | Max |
| A | — | — | 0.063 | — | — | 1.60 |
| $A_1$ | 0.002 | — | 0.006 | 0.05 | — | 0.15 |
| $A_2$ | 0.053 | 0.055 | 0.057 | 1.35 | 1.40 | 1.45 |
| b | 0.007 | 0.008 | 0.011 | 0.17 | 0.20 | 0.27 |
| c | 0.004 | — | 0.008 | 0.09 | — | 0.20 |
| D | — | 0.393 | — | — | 10.00 | — |
| E | — | 0.393 | — | — | 10.00 | — |
| e | — | 0.020 | — | — | 0.50 | — |
| $H_D$ | — | 0.472 | — | — | 12.00 | — |
| $H_E$ | — | 0.472 | — | — | 12.00 | — |
| L | 0.018 | 0.024 | 0.030 | 0.45 | 0.60 | 0.75 |
| $L_1$ | — | 0.039 | — | — | 1.00 | — |
| y | — | 0.004 | — | — | 0.10 | — |
| θ | 0° | 3.5° | 7° | 0° | 3.5° | 7° |

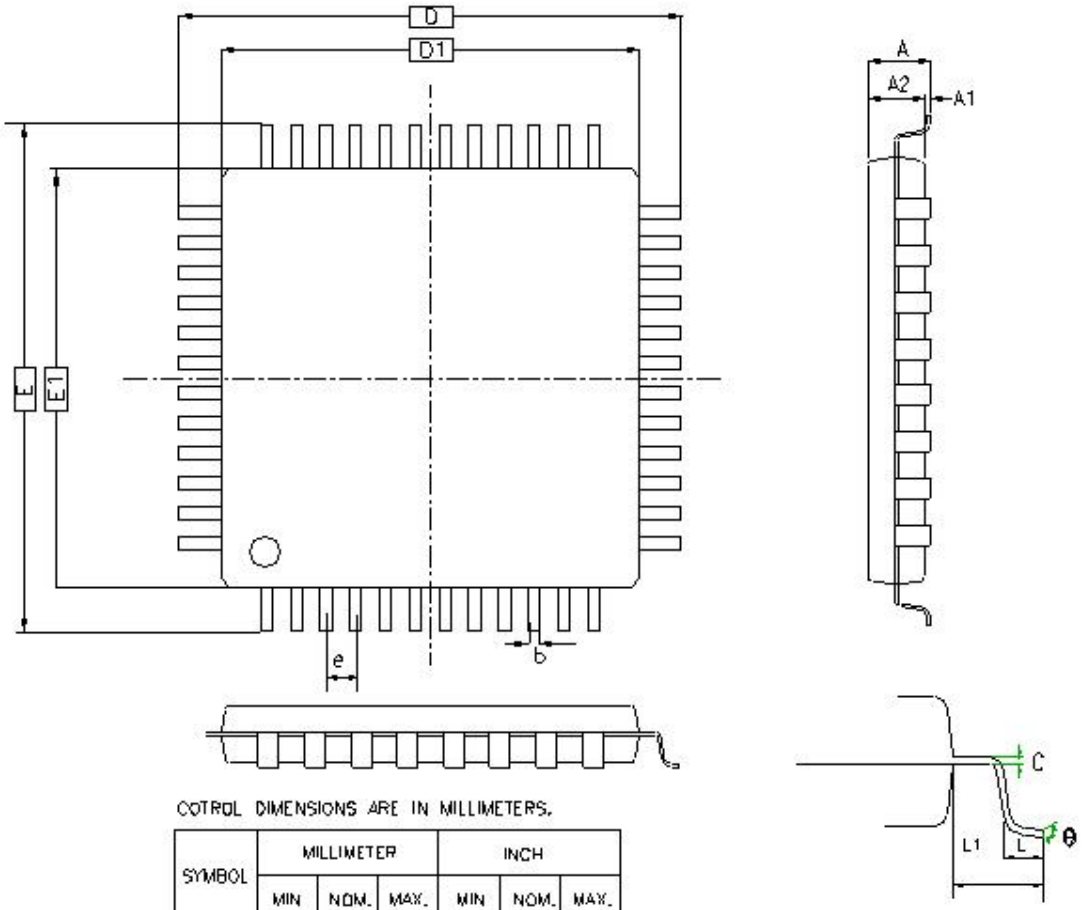
### 10.3 LQFP64 (7x7x1.4 mm footprint 2.0 mm)



| | | SYMBOL | MIN | NOM | MAX |
|---|---|---|---|---|---|
| TOTAL THICKNESS | | A | --- | --- | 1.6 |
| STAND OFF | | A1 | 0.05 | --- | 0.15 |
| MOLD THICKNESS | | A2 | 1.35 | 1.4 | 1.45 |
| LEAD WIDTH(PLATING) | | b | 0.13 | 0.18 | 0.23 |
| LEAD WIDTH | | b1 | 0.13 | 0.16 | 0.19 |
| L/F THICKNESS(PLATING) | | c | 0.09 | --- | 0.2 |
| L/F THICKNESS | | c1 | 0.09 | --- | 0.16 |
| BODY SIZE | X | D | 9 BSC | | |
| | Y | E | 9 BSC | | |
| | X | D1 | 7 BSC | | |
| | Y | E1 | 7 BSC | | |
| LEAD PITCH | | e | 0.4 BSC | | |
| | | L | 0.45 | 0.6 | 0.75 |
| FOOTPRINT | | L1 | 1 REF | | |
| | | θ | 0° | 3.5° | 7° |
| | | θ1 | 0° | --- | --- |
| | | θ2 | 11° | 12° | 13° |
| | | θ3 | 11° | 12° | 13° |
| | | R1 | 0.08 | --- | --- |
| | | R2 | 0.08 | --- | 0.2 |
| | | S | 0.2 | --- | --- |
| PACKAGE EDGE TOLERANCE | | aaa | 0.2 | | |
| LEAD EDGE TOLERANCE | | bbb | 0.2 | | |
| COPLANARITY | | ccc | 0.08 | | |
| LEAD OFFSET | | ddd | 0.07 | | |
| MOLD FLATNESS | | eee | 0.05 | | |



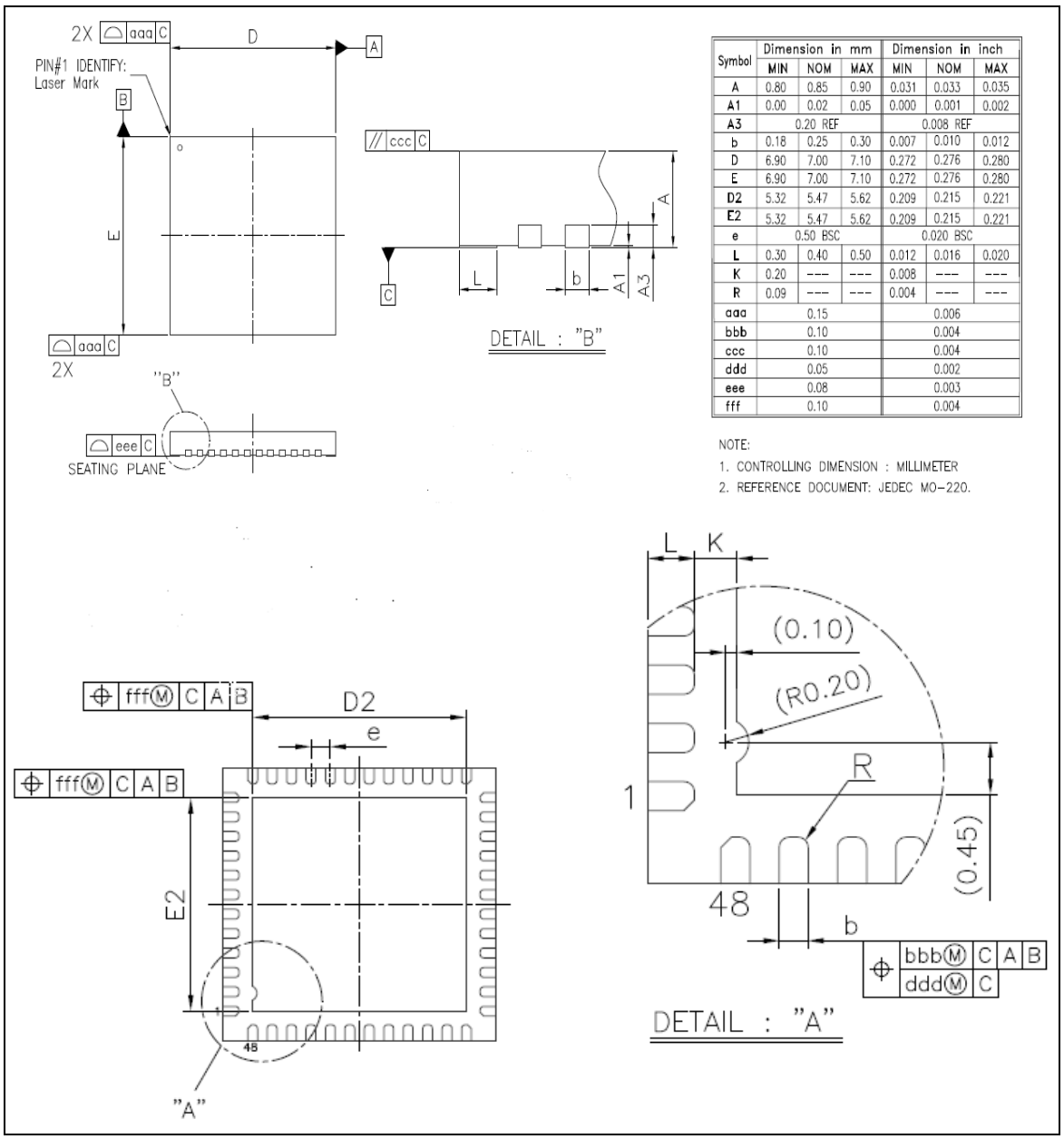
## 10.4 LQFP48 (7x7x1.4 mm footprint 2.0 mm)




CONTROL DIMENSIONS ARE IN MILLIMETERS.

| SYMBOL | MILLIMETER | | | INCH | | |
|---|---|---|---|---|---|---|
| | MIN | NOM. | MAX. | MIN | NOM. | MAX. |
| A | — | — | 1.60 | — | — | 0.063 |
| A1 | 0.05 | 0.10 | 0.15 | 0.002 | 0.004 | 0.006 |
| A2 | 1.35 | 1.40 | 1.45 | 0.053 | 0.055 | 0.057 |
| D1 | 6.90 | 7.00 | 7.10 | 0.272 | 0.276 | 0.280 |
| E1 | 6.90 | 7.00 | 7.10 | 0.272 | 0.276 | 0.280 |
| e | 0.35 | 0.50 | 0.65 | 0.014 | 0.020 | 0.260 |
| D | 8.9 | 9.00 | 9.10 | 0.350 | 0.354 | 0.358 |
| E | 8.9 | 9.00 | 9.10 | 0.350 | 0.354 | 0.358 |
| L | 0.45 | 0.60 | 0.75 | 0.018 | 0.024 | 0.030 |
| L1 | — | 1.00 | — | — | 0.039 | — |
| C | 0.09 | — | 0.20 | 0.0035 | — | 0.0079 |
| θ | 0° | — | 7° | 0° | — | 7° |
| b | 0.17 | 0.22 | 0.27 | 0.007 | 0.0087 | 0.011 |

## 10.5 QFN48 (7x7x0.85 mm)




| Symbol | Dimension in mm | | | Dimension in inch | | |
|---|---|---|---|---|---|---|
| | MIN | NOM | MAX | MIN | NOM | MAX |
| A | 0.80 | 0.85 | 0.90 | 0.031 | 0.033 | 0.035 |
| A1 | 0.00 | 0.02 | 0.05 | 0.000 | 0.001 | 0.002 |
| A3 | 0.20 REF | | | 0.008 REF | | |
| b | 0.18 | 0.25 | 0.30 | 0.007 | 0.010 | 0.012 |
| D | 6.90 | 7.00 | 7.10 | 0.272 | 0.276 | 0.280 |
| E | 6.90 | 7.00 | 7.10 | 0.272 | 0.276 | 0.280 |
| D2 | 5.32 | 5.47 | 5.62 | 0.209 | 0.215 | 0.221 |
| E2 | 5.32 | 5.47 | 5.62 | 0.209 | 0.215 | 0.221 |
| e | 0.50 BSC | | | 0.020 BSC | | |
| L | 0.30 | 0.40 | 0.50 | 0.012 | 0.016 | 0.020 |
| K | 0.20 | --- | --- | 0.008 | --- | --- |
| R | 0.09 | --- | --- | 0.004 | --- | --- |
| aaa | 0.15 | | | 0.006 | | |
| bbb | 0.10 | | | 0.004 | | |
| ccc | 0.10 | | | 0.004 | | |
| ddd | 0.05 | | | 0.002 | | |
| eee | 0.08 | | | 0.003 | | |
| fff | 0.10 | | | 0.004 | | |

NOTE:
1. CONTROLLING DIMENSION : MILLIMETER
2. REFERENCE DOCUMENT: JEDEC MO-220.

# 11 REVISION HISTORY

| Date | Revision | Description |
|---|---|---|
| 2012.10.11 | 1.00 | Initial release |
| 2012.12.11 | 1.01 | 1. Added SmartCard UART mode description in Pin Description.<br>2. Unified the abbreviation (TMR) in the Timer Controller section.<br>3. Modified the specifications of external input clock.<br>4. Added LCD COM4 and COM5 description for each pin description and diagram.<br>5. Updated the ADC enabled by timer event description in the ADC section.<br>6. Changed Timer0/1 Ch0/1 to Timer x (x=0, 1, 2, 3) in the Timer Controller section.<br>7. Added register description of Countinuous Counting mode in Timer Controller section. |
| 2012.12.17 | 1.02 | 1. Added description of reading UCID in ISP mode. |
| 2012.12.28 | 1.03 | 1. Added a table about entering Power-down mode again in section 5.3.3.<br>2. Added R-type related description in LCD section.<br>3. Updated the operating current data of Run mode and Idle mode at each frequency and added related data at 42 MHz in section 9.2. |
| 2013.01.02 | 1.04 | 1. Updated the table in Power Consumption section. |
| 2013.03.05 | 1.05 | 1. Updated the descriptions in sections 5.5.4.8 and 5.20.4.5.<br>2. Updated the display modes from four to six in section 5.13.2.<br>3. Corrected the pin descriptions in section 3.4.<br>4. Updated bit 31 (CWDT_EN) description for Config0 in section 5.9.4.4.<br>5. Updated measuring condition in Fig.5-16 and 5-17.<br>6. Added temperature sensor related description in section 5.5.4.13.<br>7. Updated clock control block diagram in section 5.4.3.<br>8. Updated temperature sensor of analog characteristic in section 9.4.4.<br>9. Corrected the description and the example of RTC frequency compensation in section 5.15.4.4.<br>10. Corrected the channel of fast and slow input in section 5.5.4.9.<br>11. Corrected UART Clock Control Diagram in section 5.19.3.<br>12. Corrected IrDA TX/RX Timing Diagram in section 5.19.4.4.<br>13. Corrected Smart Card's feature to be half duplex in UART mode in section 5.16.2.<br>14. Added the description, "The Watchdog counter will be automatically reset when the chip is entering Power-down mode." in section |

| | | |
|---|---|---|
| | | 5.21.4. |
| 2013.05.28 | 1.06 | 1. Updated the Nano110 LQFP128-pin diagram in section 3.3.2.<br>2. Updated “12 MHz OSC has 2 % deviation within all temperarure range” in sections 2.1 to 2.4.<br>3. Updated DAC analog characteristics in section 9.4.5.<br>4. Added Nano110RC2BN to the Nano110 LCD Line Selection Guide.<br>5. Corrected the typo of PA2_MFP and PA3_MFP register description in section 5.3.5. |
| 2013.12.04 | 1.07 | 1. Updated Nano100 series selection code in section 3.1.<br>2. Added the Nano100 QFN48 package in section 3.2 and QFN48 package dimensions in chapter 10.<br>3. Fixed the typo of LCD characteristic in section 9.4.7.<br>4. Added a note that “Output voltage for ADC/LCD shared pins cannot be higher than VDD because these pins are without 5V tolerance.” for pin description in section 3.4, LCD overview in section 5.13.1 and Absolute Maximum Ratings in section 9.1.<br>5. Modified the schematic for ADC and DAC application circuit in section 7.2 and 7.3. |

## Important Notice

**Nuvoton products are not designed, intended, authorized or warranted for use as components in systems or equipment intended for surgical implantation, atomic energy control instruments, airplane or spaceship instruments, transportation instruments, traffic signal instruments, combustion control instruments, or for other applications intended to support or sustain life. Further more, Nuvoton products are not intended for applications wherein failure of Nuvoton products could result or lead to a situation wherein personal injury, death or severe property or environmental damage could occur.**

**Nuvoton customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Nuvoton for any damages resulting from such improper use or sales.**

---

**Please Note that all data and specifications are subject to change without notice. All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.**