ARM Cortex®-M0
32-bit Microcontroller

NuVoice™ Family
N572F072/P072 Multi-Algorithm Voice Processor
Technical Reference Manual

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

www.nuvoton.com

1 GENERAL DESCRIPTION

The N572F072/P072 is the enhanced N572 series from N572F065, the first Cortex™-M0 based processor for voice applications. It runs up to 48MHz and equips with 72KB flash and 8KB SRAM for high performance process of audio and voice algorithms. Integrating rich analog peripherals, like pre-amplifier, ADC, DAC, hardware mixer, and PA, this chip saves a lot of system design effort and cost.

To unfold the high performance M0 and high density of SRAM, advanced algorithms are designed, optimized, and tested in N572 chip. These algorithms include voice changer, low-bit rate compression, beat detection, pitch in and pitch out, and more in developing. In additional to algorithms developed by Nuvoton, we also seek third parties for more interesting software to enrich the applications on N572.

With the NVIC in M0, the latency of interrupt and response time to external events is very short. Multiple algorithms can be run together smoothly and naturally.

The development tools are based on Keil™ MDK using C/C++ programming language. This is a robust and easy to use environment for development and debug. Features in Keil™ MDK are compiler, debugger, and profiler. With the Nu-Link™ and evaluation board, the overall system, including HW and SW, can work seamlessly in your system testing and verification. In this full-features development environment and tools, you can design and build application software in an efficient way and get optimized code that can best realize your idea on N572.

Following is a brief table of all Part No. of N572 Series:

Part No.	N572F072	N572P072	N572F065
Program ROM	72KB Flash	72KB OTP or 64KB OTP + 8KB Flash	64KB Flash
SRAM	8KB		
CPU freq	48MHz		
SPI Interface	Master/Slave mode 12MHz, 1 set Master mode 24MHz, 1 set		Master mode 12MHz, 2 sets
USB	N/A	N/A	FS/12Mbps

2 FEATURES

- Core
  - ARM® Cortex™-M0 core runs up to 50 MHz
  - One 24-bit System timer
  - Supports low power sleep mode.
  - Single-cycle 32-bit hardware multiplier.
  - NVIC (Nested Vector Interrupt Controller) for 16 interrupt inputs, each with 4-levels of priority.
  - Serial Wire Debug (SWD) supports with 2 watchpoints/4 breakpoints.
- Power Management
  - Wide operating voltage range from 2.4V to 5.5V
- Flash EPROM Memory
  - 72K bytes Flash EPROM for program code and data storage.
  - Support In-system program (ISP) for Flash update
  - 512 bytes page erase for flash
  - Support 2 wire In-circuit Programming (ICP) update from SWD ICE interface
- SRAM Memory
  - 8K bytes embedded SRAM.
- Clock Control
  - Flexible selection for different applications.
  - Support PLL, up to 48MHz, for high performance system operation.
  - External 32 KHz crystal input for RTC function and system clock.
  - Internal 48 MHz RC oscillator
- GPIO
  - Four I/O modes:
    - Quasi bi-direction
    - Push-Pull output
    - Open-Drain output
    - Input only with high impendence
  - TTL/Schmitt trigger input selectable.
  - I/O pin can be configured as interrupt source with edge/level setting.
  - High driver and high sink IO mode support.
- Timers
  - 3 sets of the timer with 8-bit pre-scaler and 16-bit counter.
  - Counter auto reload.
  - IR carrier generator
  - One fixed frequency timer
- Watch Dog Timer
  - Default ON/OFF by configuration setting
  - Multiple clock sources
  - 8 selectable time out period from 6 ms ~ 3.0 sec (depending on clock source)
  - WDT can wake up power down/sleep.
  - Interrupt or reset selectable on watchdog time-out.
- RTC
  - Support time out interrupt
  - Support wake up function
- PWM/Capture/Compare Timer
  - One 16-bit timer and four 16-bit comparators

- Five clock selectors
- One 8-bit pre-scaler and one clock dividers
- Two Dead-Zone generators
- Programmable duty control of output waveform (PWM)
- Auto reload mode or one-shot pulse mode
- Capture and compare function

- SPI
  - Two sets of SPI device.
  - Master mode up to 24MHz (36MHz at 3.3V) serial clock (SPI0 only)
  - Support master/slave mode (SPI0 only)
  - Full duplex synchronous serial data transfer
  - Variable length of transfer data from 1 to 32 bits
  - MSB or LSB first data transfer
  - Rx and Tx on both rising or falling edge of serial clock independently
  - 2 slave/device select pins
  - Two 32-bit buffers.
- Audio Analog to Digital converter
  - 8-ch 12-bit with 320Ksps.
  - Single scan/single cycle scan/continuous scan
  - 8 channels share 8 result registers
  - Programmable channel scan sequence
  - Threshold voltage detection
  - Conversion start by S/W or external pin
  - Programmable gain control for sound record
  - Internal microphone bias
- APU
  - 13-bit DAC
  - H/W mixer with 2 channel PCM input
  - Embedded power amplifier
  - 7-level volume control.
- Low Voltage Detector
  - With 2 levels: 3.0V/2.7V
- Low Dropout Voltage Regulator (LDO)
  - Built-in 1.8V LDO
- 3V Regulated Power
  - Built-in 3V regulator power supply Voutx for driving external SPI-flash.
- Low Voltage Reset: 1.8V
- Operating Temperature: -20°C~85°C
- Package:
  - All Green Package (RoHS)
  - LQFP 64-pin

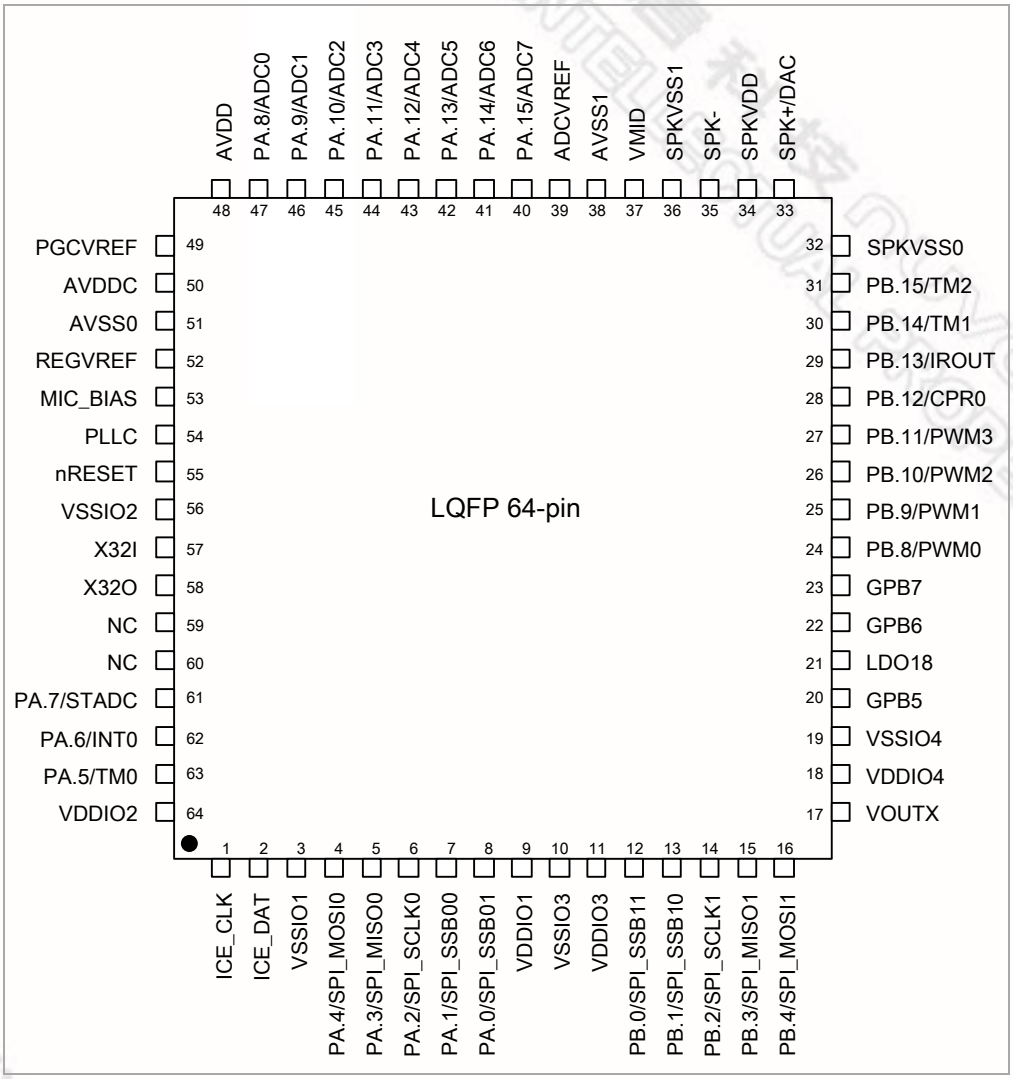
3 PART INFORMATIONAND PIN CONFIGURATION

3.1 Product Selection Guide

PART NUMBER	PACKAGE	SPECIAL FEATURE	PB FREE + HALOGEN FREE (GREEN)	RELEASE DATE
N572F072	NA (Die Form)	MTP	Yes	Available
N572F072G	LQFP 64pin 7mmx7mm	MTP	Yes	Available
N572P072	NA (Die Form)	OTP + MTP	Yes	Available
N572P072G	LQFP 64pin 7mmx7mm	OTP + MTP	Yes	Available

3.2 Pin Configuration

3.2.1 Pin Assignment (LQFP64)



3.2.2 Alternate Function List of GPIO

All General Purpose Input/Output (GPIO) pins except PB.5, PB.6 and PB.7 can be configured to alternate functions as described in the table below.

GPIO	Power	Alternate	I/O Of Alternate	Function Description
PA.0	VDDIO1	SPI_SSB01	O	SPI0 2nd chip select pin
PA.1	VDDIO1	SPI_SSB00	O	SPI0 1st chip select pin
PA.2	VDDIO1	SPI_SCLK0	O	SPI0 serial clock output
PA.3	VDDIO1	SPI_MISO0	I	SPI0 master data input
PA.4	VDDIO1	SPI_MOSI0	O	SPI0 master data output
PA.5	VDDIO2	TM0	I	Timer0 counter external input
PA.6	VDDIO2	INT0	I	External interrupt input pin
PA.7	VDDIO2	STADC	I	ADC external trigger input
PA.8	AVDD	ADC0	A	ADC analog input 0
PA.9	AVDD	ADC1	A	ADC analog input 1
PA.10	AVDD	ADC2	A	ADC analog input 2
PA.11	AVDD	ADC3	A	ADC analog input 3
PA.12	AVDD	ADC4	A	ADC analog input 4
PA.13	AVDD	ADC5	A	ADC analog input 5
PA.14	AVDD	ADC6	A	ADC analog input 6
PA.15	AVDD	ADC7	A	ADC analog input 7
PB.0	VDDIO3	SPI_SSB11	O	SPI1 2nd chip select output pin
PB.1	VDDIO3	SPI_SSB10	I/O	SPI1 1st chip select output/input pin
PB.2	VDDIO3	SPI_SCLK1	I/O	SPI1 serial clock output/input
PB.3	VDDIO3	SPI_MISO1	I/O	SPI1 master data input, slave data output
PB.4	VDDIO3	SPI_MOSI1	I/O	SPI1 master data output, slave data input
PB.5	VDDIO4	-		
PB.6	VDDIO4	-		
PB.7	VDDIO4	-		
PB.8	VDDIO4	PWM0	O	PWM output pin 0
PB.9	VDDIO4	PWM1	O	PWM output pin 1
PB.10	VDDIO4	PWM2	O	PWM output pin 2
PB.11	VDDIO4	PWM3	O	PWM output pin 3
PB.12	VDDIO4	CPR0	I	Capture input

PB.13	VDDIO4	IROUT	O	IR carrier output
PB.14	VDDIO4	TM1	I	Timer1 counter external clock input
PB.15	VDDIO4	TM2	I	Timer2 counter external clock input

I: Input, O: Output, A: Analog input

3.2.3 Pad Description

Name	Type	Power	Description
1. GPIO			
PA.0 ~ PA.15	I/O	See 3.2	Bidirectional general purpose I/O ports. All of these pins have alternate function; refer to section 3.2.2 for detail.
PB.0 ~ PB.15	I/O	See 3.2	Bidirectional general purpose I/O ports. Most of these pins have alternate function; refer to section 3.2.2 for detail.
2. Oscillator			
X32I	I	LDO18	32KHz crystal input
X32O	O	LDO18	32KHz crystal output
PLLC	A	-	Capacitor connection for built-in PLL1
3. PGC and ADC			
AVDD	P	-	Analog power supply
AVSS0	P	-	Analog ground
AVSS1	P	-	Analog ground
ADCVREF	A	-	Reference voltage input for ADC
MIC_BIAS	A	-	Microphone bias output
AVDDC	A	-	Regulator output pin for ADC and PGC, its output voltage is 0.85xAVDD. Connect an external 4.7uF capacitor to AVSS.
PGCVREF	A	-	AVDDC/2 for PGC. A 4.7uF (or higher) capacitor for low pass filter to filter power noise is needed.
REGVREF	A	-	AVDD/2 for microphone output. A 4.7uF (or higher) capacitor for low pass filter to filter power noise is needed.
4. Speaker Driver			
SPK+/DAC	O	SPVDD	Speaker positive output pin, or current type DAC output.
SPK-	O	SPVDD	Speaker negative output pin.
SPKVDD	P	-	Analog power supply
SPKVSS1	P	-	Analog ground.
SPKVSS0	P	-	Analog ground.
VMID	A	-	Connect a capacitor to SPKVSS1.
5. Power			
VDDIO4	P	-	Power supply for I/O port, LVR, BOD and source of LDO.
VSSIO4	P	-	Ground pin, connect to 0V.
VDDIO2	P	-	Power supply for I/O port.
VSSIO2	P	-	Ground pin, connect to 0V.
VDDIO1	P	-	Power supply for I/O port.

VSSIO1	P	-	Ground pin, connect to 0V.
VDDIO3	P	-	Power supply for I/O port.
VSSIO3	P	-	Ground pin, connect to 0V.
LDO18	P	-	1.8V LDO output
VOUTX	P	-	3.0V regulated power for driving out
VPP	P	-	Power supply for flash programming, for test use only. This pad is not bounded in LQFP64.
VNN	P	-	Power supply for flash programming, for test use only. This pad is not bounded in LQFP64.
6. SWD			
ICE_CLK	I	VDDIO2	Serial Wired Debugger Clock pin
ICE_DAT	I/O	VDDIO2	Serial Wired Debugger Data pin
7. Other			
nRESET	I	VDDIO2	Reset input pin, low active. Internal pull-high.

Total: 64 pads

4 BLOCK DIAGRAM

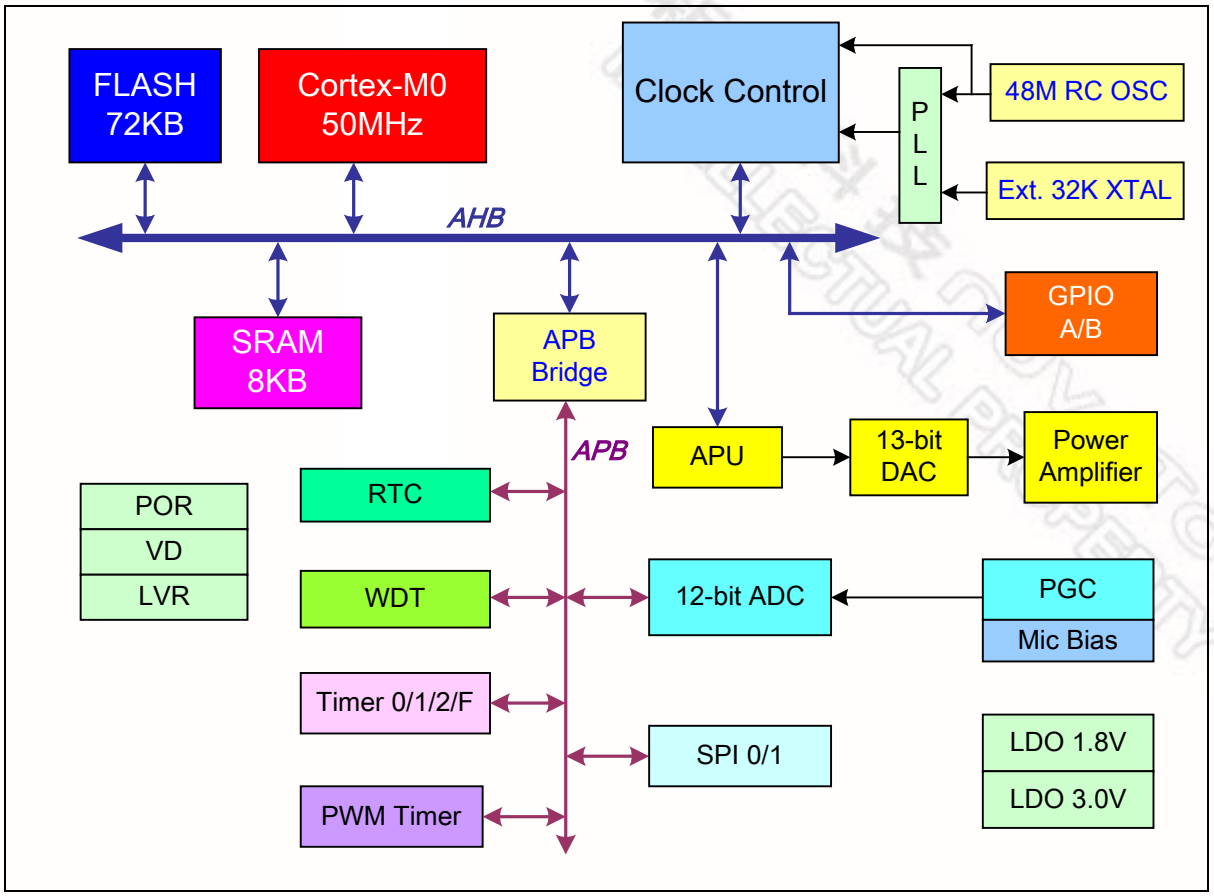


Figure 4-1 Functional Block Diagram

5 FUNCTIONAL DESCRIPTION

5.1 ARM® Cortex™-M0 core

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.

Figure 5-1 shows the functional blocks of processor.

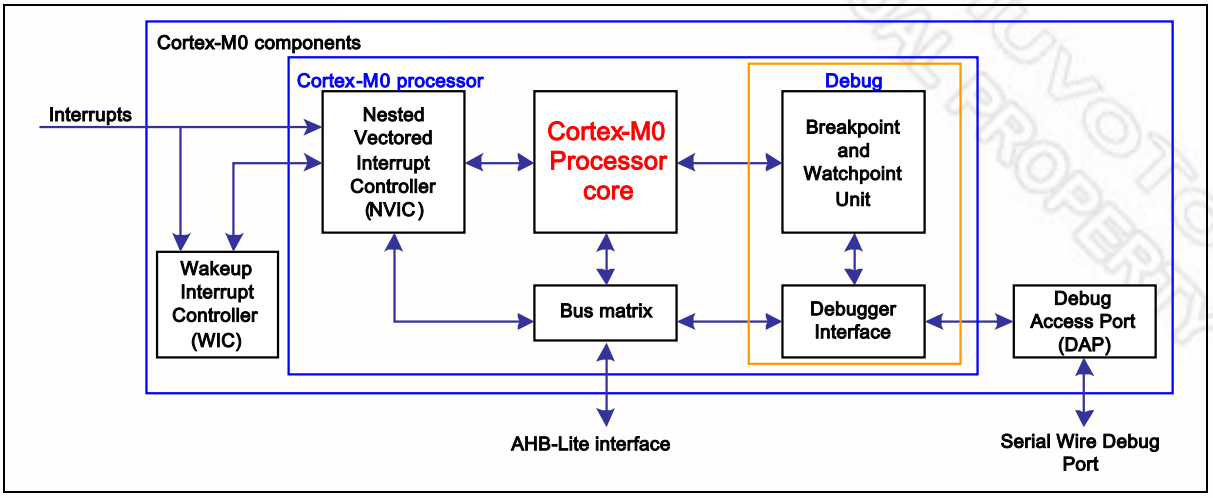


Figure 5-1 Functional Block Diagram

The implemented device provides:

- A low gate count processor that features:
  - The ARMv6-M Thumb® instruction set.
  - Thumb-2 technology.
  - ARMv6-M compliant 24-bit SysTick timer.
  - A 32-bit hardware multiplier.
  - The system interface supports little-endian data accesses.
  - The ability to have deterministic, fixed-latency, interrupt handling.
  - Load/store-multiples and multicycle-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 Even(WFE) instructions, or the return from interrupt sleep-on-exit feature.
- NVIC features
  - 16 external interrupt inputs, each with four levels of priority.
  - Dedicated non-Maskable Interrupt (NMI) input.

- Support 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 watchpoints.
  - 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 that provides simple integration to all system peripherals and memory.
  - Single 32-bit slave port that supports the DAP.

5.2 System Manager

5.2.1 Overview

The following functions are included in system manager section

- System Memory Map
- System management registers for chip and module functional reset and multi-function pin control
- Chip miscellaneous Control Register
- Combined peripheral interrupt source identify

5.2.2 System Memory Map

N572F072/P072 provides a 4G-byte address space for programmers. The memory locations assigned to each on-chip modules are shown in Table 5-1. The detailed register and memory addressing and programming will be described in the following sections for individual on-chip modules. N572F072/P072 series only supports little-endian data format.

Table 5-1 Address Space Assignments for On-Chip Modules

Address Space	Token	Modules	Reference
Flash & SRAM Memory Space			
0x0000_0000 – 0x0001_1FFF	FLASH_BA	FLASH Memory Space (72KB)	6.3
0x0030_0000 _ 0x0030_0003	CFG_BA	User Configuration Memory (4B)	6.4
0x2000_0000 – 0x2000_1FFF	SRAM_BA	SRAM Memory Space (8KB)	
AHB Modules Space (0x5000_0000 – 0x501F_FFFF)			
0x5000_0000 – 0x5000_01FF	SYS_BA	System Global Control Registers	5.2.3
0x5000_0200 – 0x5000_02FF	CLK_BA	Clock Control Registers	5.3.5
0x5000_0300 – 0x5000_03FF	INT_BA	Interrupt Multiplexer Control Registers	5.2.5.5
0x5000_4000 – 0x5000_7FFF	GPIO_BA	GPIO Control Registers	5.5.2
0x5000_8000 – 0x5000_BFFF	APU_BA	APU Controller Registers	5.11.1
0x5000_C000 – 0x5000_FFFF	FMC_BA	Flash Memory Control Registers	6.6
APB Modules Space (0x4000_0000 ~ 0x400F_FFFF)			
0x4000_4000 – 0x4000_7FFF	WDT_BA	Watch-Dog Timer Control Registers	5.10.1
0x4000_8000 – 0x4000_BFFF	RTC_BA	Real Time Clock (RTC) Control Register	5.7.2
0x4001_0000 – 0x4001_3FFF	TMR_BA	Timer0/Timer1 Control Registers	5.9.4
0x4003_0000 – 0x4003_3FFF	SPI0_BA	SPI0 Serial Interface Control Registers	5.8.6
0x4003_4000 – 0x4003_7FFF	SPI1_BA	SPI1 Serial Interface Control Registers	5.8.6
0x4004_0000 – 0x4004_3FFF	PWM_BA	PWM Control Registers	5.6.11
0x400E_0000 – 0x400E_FFFF	ADC_BA	Analog-Digital-Converter (ADC) Registers	5.12.5
System Control Space (0xE000_E000 ~ 0xE000_EFFF)			
0xE000_E010 – 0xE000_E01B	SCS_BA	System Timer Control Registers	5.2.4.1
0xE000_E100 – 0xE000_E40F	SCS_BA	NVIC Control Registers	5.2.5.4

5.2.3 System Manager Control Registers

Register	Offset	R/W	Description	Reset Value
SYS Base Address: **SYS_BA = 0x5000_0000**				
SYS_PDID	SYS_BA+0x00	R	Product Identifier Register	0xXXXX_XXXX
SYS_RSTSTS	SYS_BA+0x04	R/W	System Reset Source Register	0x0000_00XX
SYS_IPRST0	SYS_BA+0x08	R/W	IP Reset Control Resister0	0x0000_0000
SYS_IPRST1	SYS_BA+0x0C	R/W	IP Reset Control Resister1	0x0000_0000
SYS_BODCTL	SYS_BA+0x18	R/W	Brown-Out Detector Control Register	0x0000_0080
SYS_PORCTL	SYS_BA+0x1C	R/W	Power-On-Reset Controller Register	0x0000_0000
SYS_GPA_MFP	SYS_BA+0x30	R/W	GPIO PA Multiple Alternate Functions and Input Type Control Register	0x0000_0000
SYS_GPB_MFP	SYS_BA+0x34	R/W	GPIO PB Multiple Alternate Functions and Input Type Control Register	0x0000_0000
SYS_GPA_HS	SYS_BA+0x38	R/W	PA.4 ~ PA.0 High Speed Transition Control Register	0x0000_0000
SYS_REGLCTL	SYS_BA+0x100	R/W	Register Lock Control Register	0x0000_0000

Product Identifier Register (SYS_PDID)

This register provides specific read-only information for software to identify this chip.

Register	Offset	R/W	Description	Reset Value
SYS_PDID	SYS_BA+0x00	R	Product Identifier Register	0xXXXX_XXXX

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

Bits	Description	
[31:0]	PDID	**Product Identifier** Chip identifier (part number) for N572F072/N572P072 series.

System Reset Source Register (SYS_RSTSTS)

This register provides specific information for software to identify this chip's reset source from last operation.

Register	Offset	R/W	Description	Reset Value
SYS_RSTSTS	SYS_BA+0x04	R/W	System Reset Source Register	0x0000_00XX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**PMURSTF**	**Reserved**		**LVRF**	**WDTRF**	**PINRF**	**PORF**

Bits	Description	
[31:7]	**Reserved**	Reserved.
[6]	**PMURSTF**	**Reset Source From PMU** The PMURSTF flag is set by the reset signal from the PMU module to indicate the previous reset source. 0= No reset from PMU. 1= The PMU has issued the reset signal to reset the system. **Note:** Write 1 to clear this bit to 0.
[5:4]	**Reserved**	Reserved.
[3]	**LVRF**	**LVR Reset Flag** The LVR reset flag is set by the "Reset Signal" from the Low Voltage Reset Controller to indicate the previous reset source. 0 = No reset from LVR. 1 = LVR controller had issued the reset signal to reset the system. **Note:** Write 1 to clear this bit to 0.
[2]	**WDTRF**	**Reset Source From WDG** The WDTRF flag is set if pervious reset source originates from the Watch-Dog module. 0= No reset from Watch-Dog. 1= The Watch-Dog module issued the reset signal to reset the system. **Note:** Write 1 to clear this bit to 0.

[1]	**PINRF**	**nRESET Pin Reset Flag** The nRESET pin reset flag is set by the “Reset Signal” from the nRESET Pin to indicate the previous reset source. 0 = No reset from nRESET pin. 1 = Pin nRESET had issued the reset signal to reset the system. **Note:** Write 1 to clear this bit to 0.
[0]	**PORF**	**POR Reset Flag** The POR reset flag is set by the “Reset Signal” from the Power-on Reset (POR) Controller to indicate the previous reset source. 0 = No reset from POR. 1 = Power-on Reset (POR) Controller had issued the reset signal to reset the system. **Note:** Write 1 to clear this bit to 0.

IP Reset Control Register0(SYS_IPRST0)

Register	Offset	R/W	Description	Reset Value
SYS_IPRST0	SYS_BA+0x08	R/W	IP Reset Control Resister0	0x0000_0000

To program these bits needs an open lock sequence, write “59h”, “16h”, “88h” to register SYS_REGLCTL to un-lock these bits. Refer to the register SYS_REGLCTL at address SYS_BA+0x100.

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

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**RAMWS**	**Wait State Control For CPU Access RAM** 0 = 1 HCLK clock wait-state. 1 = zero wait-state.
[2]	**CPUWS**	**CPU Wait-State Control For Flash Memory Access** 0 = 1 HCLK clock wait-state. 1 = zero wait-state. **Note:** that CPUWS cannot be set as “1” when CPU runs the program to do Flash ISP operation.
[1]	**CPURST**	**CPU Kernel One Shot Reset** Setting this bit will reset the CPU kernel and Flash Memory Controller(FMC), this bit will automatically return to “0” after the 2 clock cycles 0 = Normal. 1 = Reset CPU.
[0]	**CHIPRST**	**CHIP One Shot Reset** Set this bit will reset the whole chip, this bit will automatically return to “0” after 2 clock cycles. CHIPRST has same behavior as POR reset, all the chip modules are reset and the chip configuration settings from flash are reloaded. 0 = Normal. 1 = Reset CHIP.

IP Reset Control Register1 (SYS_IPRST1)

Setting these bits “1” will generate an asynchronous reset signal to the corresponding peripheral block. The user needs to set bit to “0” to release block from the reset state.

Register	Offset	R/W	Description	Reset Value
SYS_IPRST1	SYS_BA+0x0C	R/W	IP Reset Control Resister1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			ADCRST	Reserved			
23	22	21	20	19	18	17	16
Reserved			PWMRST	Reserved			
15	14	13	12	11	10	9	8
Reserved		SPI1RST	SPI0RST	Reserved			
7	6	5	4	3	2	1	0
Reserved	TMRFRST	APURST	TMR2RST	TMR1RST	TMR0RST	GPIORST	Reserved

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28]	**ADCRST**	**ADC Controller Reset** 0 = Normal Operation. 1 = Reset.
[27:21]	**Reserved**	Reserved.
[20]	**PWMRST**	**PWM Controller Reset** 0 = Normal Operation. 1 = Reset.
[19:14]	**Reserved**	Reserved.
[13]	**SPI1RST**	**SPI1 Controller Reset** 0 = Normal Operation. 1 = Reset.
[12]	**SPI0RST**	**SPI0 Controller Reset** 0 = Normal Operation. 1 = Reset.
[11:7]	**Reserved**	Reserved.
[6]	**TMRFRST**	**TimerF Controller Reset** 0 = Normal operation. 1 = Reset.

[5]	**APURST**	**APU Controller Reset** 0 = Normal operation. 1 = Reset.
[4]	**TMR2RST**	**Timer2 Controller Reset** 0 = Normal operation. 1 = Reset.
[3]	**TMR1RST**	**Timer1 Controller Reset** 0 = Normal Operation. 1 = Reset.
[2]	**TMR0RST**	**Timer0 Controller Reset** 0 = Normal Operation. 1 = Reset.
[1]	**GPIORST**	**GPIO Controller Reset** 0 = Normal operation. 1 = Reset.
[0]	**Reserved**	Reserved.

Brown-Out Detector Control Register (SYS_BODCTL)

Register	Offset	R/W	Description	Reset Value
SYS_BODCTL	SYS_BA+0x18	R/W	Brown-Out Detector Control Register	0x0000_0080

Partial of the SYS_BODCTL control register values are initiated by the flash configuration. After the power on initialization, these bits are protected by the lock circuit. To program these bits needs an open lock sequence, write “59h”, “16h”, “88h” to address 0x5000_0100 to un-lock these bits. Refer to the register SYS_REGLCTL at address SYS_BA+0x100.

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

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**LVR_EN**	**Low Voltage Reset (LVR) Enable (Protected Bit)** The LVR function resets the chip when the input power voltage is lower than LVR circuit setting. LVR function is enabled in default. 0 = Disable LVR function. 1 = Enable LVR function – After enable the bit, the LVR function will active with 100uS delay for LVR output stable.
[6]	**BOD_OUT**	**The Status For Brown-Out Detector Output It’s a read only bit.** 0 = The detected voltage is lower than BOD_VL setting. If the BOD_EN is “0”, this bit always responses “0”. 1 = The detected voltage is higher than BOD_VL setting.
[5:2]	**Reserved**	Reserved.
[1]	**BOD_VL**	**Brown-Out Detector Threshold Voltage Selection (Initiate & Protected Bit)** The default value is set by flash controller user configuration register CONFIG[21]. 0 = Threshold voltage is 2.7V. 1 = Threshold voltage is 3.0V.
[0]	**BOD_EN**	**Brown-Out Detector Enable (Initiated & Protected Bit)** The default value is set according to the flash controller User Configuration Register CONFIG[23], BOD_EN = ~CONFIG[23]. 0 = Brown-Out Detector function is disabled. 1 = Brown-Out Detector function is enabled,.

Power-On-Reset Controller Register (SYS_PORCTL)

Register	Offset	R/W	Description	Reset Value
SYS_PORCTL	SYS_BA+0x1C	R/W	Power-On-Reset Controller Register	0x0000_0000

This register is a protected register. To program this needs an open lock sequence, write “59h”, “16h”, “88h” to address 0x5000_0100 to un-lock this bit. Refer to the register SYS_REGLCTL at address SYS_BA+0x100.

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**POROFF**	**Power-On Reset Enable Bit (Write Protect)** When powered on, the POR circuit generates a reset signal to reset the whole chip function, but noise on the power may cause the POR active again. User can disable internal POR circuit to avoid unpredictable noise to cause chip reset by writing 0x5AA5 to this field. The POR function will be active again when this field is set to another value or chip is reset by other reset source, including: nRESET, Watchdog, LVR reset, ICE reset command and the software-chip reset function.

GPIO PA Multiple Alternate Function and Input Type Control Register (SYS_GPA_MFP)

Register	Offset	R/W	Description	Reset Value
SYS_GPA_MFP	SYS_BA+0x30	R/W	GPIO PA Multiple Alternate Functions and Input Type Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PA15TYPE	PA14TYPE	PA13TYPE	PA12TYPE	PA11TYPE	PA10TYPE	PA9TYPE	PA8TYPE
23	22	21	20	19	18	17	16
PA7TYPE	PA6TYPE	PA5TYPE	PA4TYPE	PA3TYPE	PA2TYPE	PA1TYPE	PA0TYPE
15	14	13	12	11	10	9	8
PA15MFP	PA14MFP	PA13MFP	PA12MFP	PA11MFP	PA10MFP	PA9MFP	PA8MFP
7	6	5	4	3	2	1	0
PA7MFP	PA6MFP	PA5MFP	PA4MFP	PA3MFP	PA2MFP	PA1MFP	PA0MFP

Bits	Description	
[31:16]	PAnTYPEn	0 = PA.n I/O cell input Schmitt Trigger function is disabled. 1 = PA.n I/O cell input Schmitt Trigger function is enabled.
[15]	PA15MFP	0 = The GPIOA-15 is selected to the pin PA.15. 1 = ADC input channel 7.
[14]	PA14MFP	0 = The GPIOA-14 is selected to the pin PA.14. 1 = ADC input channel 6.
[13]	PA13MFP	0 = The GPIOA-13 is selected to the pin PA.13. 1 = ADC input channel 5.
[12]	PA12MFP	0 = The GPIOA-12 is selected to the pin PA.12. 1 = ADC input channel 4.
[11]	PA11MFP	0 = The GPIOA-11 is selected to the pin PA.11. 1 = ADC input channel 3.
[10]	PA10MFP	0 = The GPIOA-10 is selected to the pin PA.10. 1 = ADC input channel 2.
[9]	PA9MFP	0 = The GPIOA-9 is selected to the pin PA.9. 1 = ADC input channel 1.
[8]	PA8MFP	0 = The GPIOA-8 is selected to the pin PA.8. 1 = ADC input channel 0.
[7]	PA7MFP	0 = The GPIOA-7 is selected to the pin PA.7. 1 = ADC input external trigger input.
[6]	PA6MFP	0 = The GPIOA-6 is selected to the pin PA.6. 1 = External interrupt input.

[5]	**PA5MFP**	0 = The GPIOA-5 is selected to the pin PA.5. 1 = Timer0 counter external input.
[4]	**PA4MFP**	0 = The GPIOA-4 is selected to the pin PA.4. 1 = SPI0 data output.
[3]	**PA3MFP**	0 = The GPIOA-3 is selected to the pin PA.3. 1 = SPI0 data input.
[2]	**PA2MFP**	0 = The GPIOA-2 is selected to the pin PA.2. 1 = SPI0 clock output.
[1]	**PA1MFP**	0 = The GPIOA-1 is selected to the pin PA.1. 1 = SPI0 1st chip select output.
[0]	**PA0MFP**	0 = The GPIOA-0 is selected to the pin PA.0. 1 = SPI0 2nd chip select output.

GPIO PB Multiple Alternate Function and Input Type Control Register (SYS_GPB_MFP)

Register	Offset	R/W	Description	Reset Value
SYS_GPB_MFP	SYS_BA+0x34	R/W	GPIO PB Multiple Alternate Functions and Input Type Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PB15TYPE	PB14TYPE	PB13TYPE	PB12TYPE	PB11TYPE	PB10TYPE	PB9TYPE	PB8TYPE
23	22	21	20	19	18	17	16
PB7TYPE	PB6TYPE	PB5TYPE	PB4TYPE	PB3TYPE	PB2TYPE	PB1TYPE	PB0TYPE
15	14	13	12	11	10	9	8
PB15MFP	PB14MFP	PB13MFP	PB12MFP	PB11MFP	PB10MFP	PB9MFP	PB8MFP
7	6	5	4	3	2	1	0
Reserved			PB4MFP	PB3MFP	PB2MFP	PB1MFP	PB0MFP

Bits	Description	
[31:16]	PBnTYPEn	0 = PB.n I/O cell input Schmitt Trigger function is disabled. 1 = PB.n I/O cell input Schmitt Trigger function is enabled.
[15]	PB15MFP	0 = The GPIOB-15 is selected to the pin PB.15. 1 = Timer2 counter external input.
[14]	PB14MFP	0 = The GPIOB-14 is selected to the pin PB.14. 1 = Timer1 counter external input.
[13]	PB13MFP	0 = The GPIOB-13 is selected to the pin PB.13. 1 = IR carrier output.
[12]	PB12MFP	0 = The GPIOB-12 is selected to the pin PB.12. 1 = PWM timer capture input.
[11]	PB11MFP	0 = The GPIOB-11 is selected to the pin PB.11. 1 = PWM output pin 3.
[10]	PB10MFP	0 = The GPIOB-10 is selected to the pin PB.10. 1 = PWM output pin 2.
[9]	PB9MFP	0 = The GPIOB-9 is selected to the pin PB.9. 1 = PWM output pin 1.
[8]	PB8MFP	0 = The GPIOB-8 is selected to the pin PB.8. 1 = PWM output pin 0.
[7:5]	Reserved	Reserved.
[4]	PB4MFP	0 = The GPIOB-4 is selected to the pin PB.4. 1 = SPI1 data output/input.
[3]	PB3MFP	0 = The GPIOB-3 is selected to the pin PB.3. 1 = SPI1 data input/output.

[2]	**PB2MFP**	0 = The GPIOB-2 is selected to the pin PB.2. 1 = SPI1 clock output/input.
[1]	**PB1MFP**	0 = The GPIOB-1 is selected to the pin PB.1. 1 = SPI1 1st chip select output or slave select input.
[0]	**PB0MFP**	0 = The GPIOB-0 is selected to the pin PB.0. 1 = SPI1 2nd chip select output.

GPIO PA.4~PA.0 High Speed Transition Control Register (SYS_GPA_HS)

Register	Offset	R/W	Description	Reset Value
SYS_GPA_HS	SYS_BA+0x38	R/W	PA.4 ~ PA.0 High Speed Transition Control Register	0x0000_0000

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

Bits	Description	
[31:21]	**Reserved**	Reserved.
[20:16]	**GPRB**	Five general purpose R/W register bits.
[17:5]	**Reserved**	Reserved.
[4:0]	**GPA_HS**	GPA_HS[n] =0: PA.n I/O is normal speed (less than 15MHz) transition. =1: PA.n I/O is high speed (up to 48MHz) transition. (n=0~4).

Register Lock Control Register (SYS_REGLCTL)

Certain critical system control registers are protected against inadvertent write operations which may disturb chip operation. These system control registers are locked after power on reset until the user specifically issues an unlock sequence to disable register protection. The unlock sequence is to write to SYS_REGLCTL the data 0x59, 0x16, 0x88 sequentially. Any different data value, different sequence or any other write to other address during these three data writing will abort the whole sequence.

User can check the lock status by reading SYS_REGLCTL bit0: “1” is unlocked, “0” is locked. Once unlocked, user can update the target protected register value. To lock registers again, write any data to the register SYS_REGLCTL to enable register protection.

This register is” write” accessible to disable/enable register protection and “read” accessible to know the lock/unlock status.

Register	Offset	R/W	Description	Reset Value
SYS_REGLCTL	SYS_BA+0x100	R/W	Register Lock Control Register	0x0000_0000

7	6	5	4	3	2	1	0
SYS_REGLCTL[7:1]							**SYS_REGLCTL[0]/ REGLCTL**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7:0]	**SYS_REGLCTL[7:0]**	**Register Lock Control Code (Write Only)** Some registers have write-protection function. Writing these registers have to disable the protected function by writing the sequence value “59h”, “16h”, “88h” to this field. After this sequence is completed, the REGLCTL bit will be set to 1 and write-protection registers can be normal write.

[0]	**REGLCTL**	**Protected Register Lock/Unlock Index (Read Only)** 0 = Protected registers are locked. Any write to the target register is ignored. 1 = Protected registers are unlocked. The protected registers are: **SYS_IPRST0** – address 0x5000_0008 **SYS_BODCTL** – address 0x5000_0018 **SYS_PORCTL** – address 0x5000_001C **CLK_PWRCTL** – address 0x5000_0200 (bit[6] is not protected for power wake-up interrupt clear) **CLK_APBCLK bit[0]** – address 0x5000_0208 (bit[0] is watch dog clock enable) **CLK_CLKSEL0** – address 0x5000_0210 (for HCLK clock source select) **CLK_CLKSEL1** -- address 0x5000_0214 (bit[1:0] is watch dog clock selection) **NMI_SEL[7]= IRQ_TM** – address 0x5000_0380. **SPI0_RCLK** – address 0x4003_0030 **FMC_ISPCTL** -- address 0x5000_C000 (Flash ISP Control register) **WDT_CTL** -- address 0x4000_4000

5.2.4 System Timer (SysTick)

The Cortex-M0 includes an integrated system timer, SysTick which provides a simple, 24-bit clear-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 time to completion and time used.
- An internal clock source control based on missing/meeting durations. The COUNTFLAG bit-field in the control and status register can be used to determine if an action completed within a set duration, as part of a dynamic clock management control loop.

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

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 documents “ARM®Cortex™-M0 Technical Reference Manual” and “ARM® v6-M Architecture Reference Manual”.

5.2.4.1 *System Timer Control Register Map*

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **SCS_BA = 0xE000_E000**				
SYST_CSR	SCS_BA + 0x10	R/W	SysTick Control and Status Register	0x0000_0000
SYST_RVR	SCS_BA + 0x14	R/W	SysTick Reload Value Register	0xXXXX_XXXX
SYST_CVR	SCS_BA + 0x18	R/W	SysTick Current Value Register	0xXXXX_XXXX

5.2.4.2 *System Timer Control Register Description*

SysTick Control and Status（SYST_CSR）

Register	Offset	R/W	Description	Reset Value
SYST_CSR	SCS_BA + 0x10	R/W	SysTick Control and Status Register	0x0000_0000

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

Bits	Description	
[31:17]	**Reserved**	Reserved.
[16]	**COUNTFLAG**	**System Tick Count Flag** Returns 1 if timer counted to 0 since last time this register was read. 0 = Cleared on read or by a write to the Current Value register. 1 = Set by a count transition from 1 to 0.
[15:3]	**Reserved**	Reserved.
[2]	**CLKSRC**	**System Tick Clock Source Selection** 0 = clock source is (optional) external reference clock. 1 = core clock used for SysTick. If no external clock provided, this bit will read as 1 and ignore writes.
[1]	**TICKINT**	**Enables SYST Exception Request** 0 = Counting down to 0 does not cause the SysTick exception to be pended. Software can use COUNTFLAG to determine if a count to zero has occurred. 1 = Counting down to 0 will cause SysTick exception to be pended. Clearing the SysTick current value register by a register write in software will not cause SysTick to be pended.
[0]	**ENABLE**	**System Tick Counter Enabled** 0 = The counter is disabled. 1 = The counter will operate in a multi-shot manner.

SysTick Reload Value Register（SYST_RVR）

Register	Offset	R/W	Description	Reset Value
SYST_RVR	SCS_BA + 0x14	R/W	SysTick Reload Value Register	0xXXXX_XXXX

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

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**RELOAD**	**System Tick Reload Value** Value to load into the Current Value register when the counter reaches 0. To generate a multi-shot timer with a period of N processor clock cycles, use a RELOAD value of N-1. For example, if the SysTick interrupt is required every 200 clock pulses, set RELOAD to 199.

SysTick Current Value Register（SYST_CVR）

Register	Offset	R/W	Description	Reset Value
SYST_CVR	SCS_BA + 0x18	R/W	SysTick Current Value Register	0xXXXX_XXXX

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

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:0]	**CURRENT**	**System Tick Current Counter Value** This is the value of the counter at the time it is sampled. The counter does not provide read-modify-write protection. The register is write-clear. A software write of any value will clear the register to 0.

5.2.5 Nested Vectored Interrupt Controller (NVIC)

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:

- 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 16 (IRQ[15:0]) 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 interrupt 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 documents "ARM® Cortex™-M0 Technical Reference Manual" and "ARM® v6-M Architecture Reference Manual".

5.2.5.1 *Exception Model and System Interrupt Map*

The following table lists the exception model supported by N572F072/P072. Software can set four levels of priority on certain 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".

Table 5-2 Exception Model

Exception Name	Vector Number	Priority
Reset	1	-3
NMI	2	-2
Hard Fault	3	-1
Reserved	4 ~ 10	N/A
SVCall	11	Configurable
Reserved	12 ~ 13	N/A
PendSV	14	Configurable
SysTick	15	Configurable
Interrupt (IRQ0 ~ IRQ15)	16 ~ 31	Configurable

Table 5-3 System Interrupt Map

Vector Number	Interrupt Number (Bit In Interrupt Registers)	Interrupt Name	Source IP	Interrupt Description
0 ~ 15	-	**-**	-	System exceptions
16	0	**WDT_INT**	WDT	Watch Dog Timer interrupt
17	1	**APU_INT**	APU	APU threshold interrupt
18	2	**ADC_INT**	ADC0~7	Interrupt from ADC
19	3	**EXINT**	GPIO	External interrupt
20	4	**Reserved**	-	-
21	5	**TMR0_INT**	Timer0	Timer0
22	6	**TMR1_INT**	Timer1	Timer1
23	7	**TMR2_INT**	Timer2	Timer2
24	8	**GPAB_INT**	GPIO A/B	Port interrupt from GPIO port
25	9	**SPI0_INT**	SPI0	Interrupt from SPI0
26	10	**PWM0_INT**	PWM Timer	PWM Timer interrupt
27	11	**SPI1_INT**	SPI1	Interrupt from SPI1
28	12	**TMRF_INT**	TimerF	Fixed frequency TimerF
29	13	**RTC_INT**	RTC	Real time clock interrupt
30	14	**PWRWU_INT**	CLKC	Interrupt form Clock controller for chip wake up from power-down state
31	15	**Reserved**	-	-

5.2.5.2 *Vector Table*

When an interrupt is accepted, the processor will automatically fetch the starting address of the interrupt service routine (ISR) from the vector table in memory. For ARMv6-M, the vector table base address is fixed in flash 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.

Table 5-4 Vector Table Format

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

5.2.5.3 *Operation Description*

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.

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.

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.

5.2.5.4 *NVIC Control Registers*

R: read only, **W**: write only, **R/W**: both read and write, **W&C**: Write 1 clear

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **SCS_BA = 0xE000_E000**				
NVIC_ISER	SCS_BA+0x100	R/W	IRQ0 ~ IRQ15 Set-Enable Control Register	0x0000_0000
NVIC_ICER	SCS_BA+0x180	R/W	IRQ0 ~ IRQ15 Clear-Enable Control Register	0x0000_0000
NVIC_ISPR	SCS_BA+0x200	R/W	IRQ0 ~ IRQ15 Set-Pending Control Register	0x0000_0000
NVIC_ICPR	SCS_BA+0x280	R/W	IRQ0 ~ IRQ15 Clear-Pending Control Register	0x0000_0000
NVIC_IPR0	SCS_BA+0x400	R/W	IRQ0 ~ IRQ3 Priority Control Register	0x0000_0000
NVIC_IPR1	SCS_BA+0x404	R/W	IRQ4 ~ IRQ7 Priority Control Register	0x0000_0000
NVIC_IPR2	SCS_BA+0x408	R/W	IRQ8 ~ IRQ11 Priority Control Register	0x0000_0000
NVIC_IPR3	SCS_BA+0x40C	R/W	IRQ12 ~ IRQ15 Priority Control Register	0x0000_0000

IRQ0 ~ IRQ15 Set-Enable Control Register （NVIC_ISER）

Register	Offset	R/W	Description	Reset Value
NVIC_ISER	SCS_BA+0x100	R/W	IRQ0 ~ IRQ15 Set-Enable Control Register	0x0000_0000

If a pending interrupt is enabled, the NVIC activates the interrupt based on its priority. If an interrupt is not enabled, asserting its interrupt signal changes the interrupt state to pending, but the NVIC never activates the interrupt, regardless of its priority.

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**SETENA**	**Interrupt Set-Enable Bit** The NVIC_ISER register enables interrupts, and shows what interrupts are enabled. Each bit represents an interrupt number from IRQ0 ~ IRQ15 (Vector number from 16 ~ 31). Write Operation: 0 = No effect. 1 = Interrupt Enabled. Read Operation: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ0 ~ IRQ15 Clear-Enable Control Register (NVIC_ICER）

Register	Offset	R/W	Description	Reset Value
NVIC_ICER	SCS_BA+0x180	R/W	IRQ0 ~ IRQ15 Clear-Enable Control Register	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CLRENA**	**Interrupt Clear-Enable Bit** The NVIC_ICER register disables interrupts, and shows what interrupts are enabled. Each bit represents an interrupt number from IRQ0 ~ IRQ15 (Vector number from 16 ~ 31). Write Operation: 0 = No effect. 1 = Interrupt Disabled. Read Operation: 0 = Interrupt Disabled. 1 = Interrupt Enabled.

IRQ0 ~ IRQ15 Set-Pending Control Register （NVIC_ISPR）

Register	Offset	R/W	Description	Reset Value
NVIC_ISPR	SCS_BA+0x200	R/W	IRQ0 ~ IRQ15 Set-Pending Control Register	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**SETPEND**	**Interrupt Set-Pending Bit** The NVIC_ISPR register forces interrupts into the pending state, and shows what interrupts are pending. Each bit represents an interrupt number from IRQ0 ~ IRQ15 (Vector number from 16 ~ 31). Write Operation: 0 = No effect. 1 = Changes interrupt state to pending. Read Operation: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ0 ~ IRQ15 Clear-Pending Control Register （NVIC_ICPR）

Register	Offset	R/W	Description	Reset Value
NVIC_ICPR	SCS_BA+0x280	R/W	IRQ0 ~ IRQ15 Clear-Pending Control Register	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CLRPEND**	**Interrupt Clear-Pending Bit** The NVIC_ICPR register removes the pending state of associated interrupts, and shows what interrupts are pending. Each bit represents an interrupt number from IRQ0 ~ IRQ15 (Vector number from 16 ~ 31). Write Operation: 0 = No effect. 1 = Removes pending state of an interrupt. Read Operation: 0 = Interrupt is not pending. 1 = Interrupt is pending.

IRQ0 ~ IRQ3 Interrupt Priority Register （NVIC_IPR0）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR0	SCS_BA+0x400	R/W	IRQ0 ~ IRQ3 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_3		Reserved					
23	22	21	20	19	18	17	16
PRI_2		Reserved					
15	14	13	12	11	10	9	8
PRI_1		Reserved					
7	6	5	4	3	2	1	0
PRI_0		Reserved					

Bits	Description	
[31:30]	PRI_3	**Priority Of IRQ3** “0” denotes the highest priority and “3” denotes lowest priority
[23:22]	PRI_2	**Priority Of IRQ2** “0” denotes the highest priority and “3” denotes lowest priority
[15:14]	PRI_1	**Priority Of IRQ1** “0” denotes the highest priority and “3” denotes lowest priority
[7:6]	PRI_0	**Priority Of IRQ0** “0” denotes the highest priority and “3” denotes lowest priority

IRQ4 ~ IRQ7 Interrupt Priority Register （NVIC_IPR1）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR1	SCS_BA+0x404	R/W	IRQ4 ~ IRQ7 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_7		Reserved					
23	22	21	20	19	18	17	16
PRI_6		Reserved					
15	14	13	12	11	10	9	8
PRI_5		Reserved					
7	6	5	4	3	2	1	0
PRI_4		Reserved					

Bits	Description	
[31:30]	**PRI_7**	**Priority Of IRQ7** “0” denotes the highest priority and “3” denotes lowest priority
[23:22]	**PRI_6**	**Priority Of IRQ6** “0” denotes the highest priority and “3” denotes lowest priority
[15:14]	**PRI_5**	**Priority Of IRQ5** “0” denotes the highest priority and “3” denotes lowest priority
[7:6]	**PRI_4**	**Priority Of IRQ4** “0” denotes the highest priority and “3” denotes lowest priority

IRQ8 ~ IRQ11 Interrupt Priority Register （NVIC_IPR2）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR2	SCS_BA+0x408	R/W	IRQ8 ~ IRQ11 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_11		Reserved					
23	22	21	20	19	18	17	16
PRI_10		Reserved					
15	14	13	12	11	10	9	8
PRI_9		Reserved					
7	6	5	4	3	2	1	0
PRI_8		Reserved					

Bits	Description	
[31:30]	**PRI_11**	**Priority Of IRQ11** “0” denotes the highest priority and “3” denotes lowest priority
[23:22]	**PRI_10**	**Priority Of IRQ10** “0” denotes the highest priority and “3” denotes lowest priority
[15:14]	**PRI_9**	**Priority Of IRQ9** “0” denotes the highest priority and “3” denotes lowest priority
[7:6]	**PRI_8**	**Priority Of IRQ8** “0” denotes the highest priority and “3” denotes lowest priority

IRQ12 ~ IRQ15 Interrupt Priority Register （NVIC_IPR3）

Register	Offset	R/W	Description	Reset Value
NVIC_IPR3	SCS_BA+0x40C	R/W	IRQ12 ~ IRQ15 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_15		RESERVED					
23	22	21	20	19	18	17	16
PRI_14		RESERVED					
15	14	13	12	11	10	9	8
PRI_13		RESERVED					
7	6	5	4	3	2	1	0
PRI_12		RESERVED					

Bits	Description	
[31:30]	PRI_15	**Priority Of IRQ15** “0” denotes the highest priority and “3” denotes lowest priority
[23:22]	PRI_14	**Priority Of IRQ14** “0” denotes the highest priority and “3” denotes lowest priority
[15:14]	PRI_13	**Priority Of IRQ13** “0” denotes the highest priority and “3” denotes lowest priority
[7:6]	PRI_12	**Priority Of IRQ12** “0” denotes the highest priority and “3” denotes lowest priority

5.2.5.5 *Interrupt Source Control Registers*

Along with the interrupt control registers associated with the NVIC, N572F072/P072 also implements some specific control registers to facilitate the interrupt functions, including “interrupt source identify”, ”NMI source selection” and “interrupt test mode”. They are described as below.

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
INT Base Address: **INT_BA = 0x5000_0300**				
IRQ0_SRC	INT_BA+0x00	R	IRQ0 (WDT) Interrupt Source Identity Register	0xXXXX_XXXX
IRQ1_SRC	INT_BA+0x04	R	IRQ1 (APU) Interrupt Source Identity Register	0xXXXX_XXXX
IRQ2_SRC	INT_BA+0x08	R	IRQ2 (ADC) Interrupt Source Identity Register	0xXXXX_XXXX
IRQ3_SRC	INT_BA+0x0C	R	IRQ3 (EXINT) Interrupt Source Identity Register	0xXXXX_XXXX
IRQ5_SRC	INT_BA+0x14	R	IRQ5 (Timer0) Interrupt Source Identity Register	0xXXXX_XXXX
IRQ6_SRC	INT_BA+0x18	R	IRQ6 (Timer1) Interrupt Source Identity Register	0xXXXX_XXXX
IRQ7_SRC	INT_BA+0x1C	R	IRQ7 (Timer2) Interrupt Source Identity Register	0xXXXX_XXXX
IRQ8_SRC	INT_BA+0x20	R	IRQ8 (GPA/B) Interrupt Source Identity Register	0xXXXX_XXXX
IRQ9_SRC	INT_BA+0x24	R	IRQ9 (SPI0) Interrupt Source Identity Register	0xXXXX_XXXX
IRQ10_SRC	INT_BA+0x28	R	IRQ10 (PWM) Interrupt Source Identity Register	0xXXXX_XXXX
IRQ11_SRC	INT_BA+0x2C	R	IRQ11 (SPI1) Interrupt Source Identity Register	0xXXXX_XXXX
IRQ12_SRC	INT_BA+0x30	R	IRQ12 (TimerF) Interrupt Source Identity Register	0xXXXX_XXXX
IRQ13_SRC	INT_BA+0x34	R	IRQ13 (RTC) Interrupt Source Identity Register	0xXXXX_XXXX
IRQ14_SRC	INT_BA+0x38	R	IRQ14 (PWRWU) Interrupt Source Identity Register	0xXXXX_XXXX
NMI_SEL	INT_BA+0x80	R/W	NMI Source Interrupt Select Control Register	0x0000_000F
MCU_IRQ	INT_BA+0x84	R/W	MCU IRQ Number Identify Register	0x0000_0000

IRQ0(BOD) Interrupt Source Identify Register (IRQ0_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ0_SRC	INT_BA+0x00	R	IRQ0 (WDT) Interrupt Source Identity Register	0xXXXX_XXXX

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**INT_SRC**	**Interrupt Source Identity** Bit2: 0 Bit1: 0 Bit0: WDT_INT

IRQ1(APU) Interrupt Source Identify Register (IRQ1_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ1_SRC	INT_BA+0x04	R	IRQ1 (APU) Interrupt Source Identity Register	0xXXXX_XXXX

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**INT_SRC**	**Interrupt Source Identity** Bit2: 0 Bit1: 0 Bit0: APU_INT

IRQ2(ADC) Interrupt Source Identify Register (IRQ2_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ2_SRC	INT_BA+0x08	R	IRQ2 (ADC) Interrupt Source Identity Register	0xXXXX_XXXX

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**INT_SRC**	**Interrupt Source Identity** Bit2: 0 Bit1: 0 Bit0: ADC_INT

IRQ3(EXINT) Interrupt Source Identify Register (IRQ3_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ3_SRC	INT_BA+0x0C	R	IRQ3 (EXINT) Interrupt Source Identity Register	0xXXXX_XXXX

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**INT_SRC**	**Interrupt Source Identity** Bit2: 0 Bit1: 0 Bit0: External interrupt

IRQ5(Timer0) Interrupt Source Identify Register (IRQ5_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ5_SRC	INT_BA+0x14	R	IRQ5 (Timer0) Interrupt Source Identity Register	0xXXXX_XXXX

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**INT_SRC**	**Interrupt Source Identity** Bit2: 0 Bit1: 0 Bit0: Timer0_INT

IRQ6(Timer1) Interrupt Source Identify Register (IRQ6_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ6_SRC	INT_BA+0x18	R	IRQ6 (Timer1) Interrupt Source Identity Register	0xXXXX_XXXX

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**INT_SRC**	**Interrupt Source Identity** Bit2: 0 Bit1: 0 Bit0: Timer1_INT

IRQ7(Timer2) Interrupt Source Identify Register (IRQ7_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ7_SRC	INT_BA+0x1C	R	IRQ7 (Timer2) Interrupt Source Identity Register	0xXXXX_XXXX

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**INT_SRC**	**Interrupt Source Identity** Bit2: 0 Bit1: 0 Bit0: Timer2_INT

IRQ8(GPA/B) Interrupt Source Identify Register (IRQ8_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ8_SRC	INT_BA+0x20	R	IRQ8 (GPA/B) Interrupt Source Identity Register	0xXXXX_XXXX

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**INT_SRC**	**Interrupt Source Identity** Bit2: 0 Bit1: GPB_INT Bit0: GPA_INT

IRQ9(SPI0) Interrupt Source Identify Register (IRQ9_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ9_SRC	INT_BA+0x24	R	IRQ9 (SPI0) Interrupt Source Identity Register	0xXXXX_XXXX

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**INT_SRC**	**Interrupt Source Identity** Bit2: 0 Bit1: 0 Bit0: SPI0_INT

IRQ10(PWM) Interrupt Source Identify Register (IRQ10_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ10_SRC	INT_BA+0x28	R	IRQ10 (PWM) Interrupt Source Identity Register	0xXXXX_XXXX

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**INT_SRC**	**Interrupt Source Identity** Bit2: 0 Bit1: 0 Bit0: PWM_INT

IRQ11(SPI1) Interrupt Source Identify Register (IRQ11_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ11_SRC	INT_BA+0x2C	R	IRQ11 (SPI1) Interrupt Source Identity Register	0xXXXX_XXXX

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**INT_SRC**	**Interrupt Source Identity** Bit2: 0 Bit1: 0 Bit0: SPI1_INT

IRQ12(TimerF) Interrupt Source Identify Register (IRQ12_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ12_SRC	INT_BA+0x30	R	IRQ12 (TimerF) Interrupt Source Identity Register	0xXXXX_XXXX

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**INT_SRC**	**Interrupt Source Identity** Bit2: 0 Bit1: 0 Bit0: TimerF_INT

IRQ13(RTC) Interrupt Source Identify Register (IRQ13_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ13_SRC	INT_BA+0x34	R	IRQ13 (RTC) Interrupt Source Identity Register	0xXXXX_XXXX

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**INT_SRC**	**Interrupt Source Identity** Bit2: 0 Bit1: 0 Bit0: RTC_INT

IRQ14(PWRWU) Interrupt Source Identify Register (IRQ14_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ14_SRC	INT_BA+0x38	R	IRQ14 (PWRWU) Interrupt Source Identity Register	0xXXXX_XXXX

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**INT_SRC**	**Interrupt Source Identity** Bit2: 0 Bit1: 0 Bit0: PWRWU_INT

NMI Interrupt Source Select Control Register (NMI_SEL)

Register	Offset	R/W	Description	Reset Value
NMI_SEL	INT_BA+0x80	R/W	NMI Source Interrupt Select Control Register	0x0000_000F

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

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**IRQ_TM**	**IRQ Test Mode** This bit is the protected bit. To program this bit needs an open lock sequence, write “59h”, “16h”, “88h” to register SYS_REGLCTL to un-lock this bit. Refer to the register SYS_REGLCTL at address SYS_BA+0x100. 0 = The interrupt register MCU_IRQ operates in normal mode. The MCU_IRQ collects all the interrupts from the peripheral and generates interrupt to MCU. 1 = All the interrupts from peripheral to MCU are blocked. The peripheral IRQ signals (0-15) are replaced by the value in the MCU_IRQ register.
[6:4]	**Reserved**	Reserved.
[3:0]	**NMI_SEL**	**NMI Source Interrupt Select** The NMI interrupt to Cortex-M0 can be selected from one of the interrupt[15:0]. The NMI_SEL bit is used to select the NMI interrupt source. **Note:** IRQ4 and IRQ15 are reserved in N572F072/P072.

MCU Interrupt Request Source Test Mode Register (MCU_IRQ)

Register	Offset	R/W	Description	Reset Value
MCU_IRQ	INT_BA+0x84	R/W	MCU IRQ Number Identify Register	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**MCU_IRQ**	**MCU IRQ Source Test Mode** The MCU_IRQ collects all the interrupts from the peripherals and generates the synchronous interrupt to MCU Cortex-M0. There are two modes to generate interrupt to MCU Cortex-M0, the normal mode and test mode. In Normal mode (NMI_SEL register bit [7] = 0) The MCU_IRQ collects all interrupts from each peripheral and synchronizes them to interrupt the Cortex-M0. In Test mode (NMI_SEL register bit [7] = 1), the interrupts from peripherals are blocked, and the interrupts are replaces by MCU_IRQ[15:0]. When MCU_IRQ[n] is “0” : Writing MCU_IRQ[n] “1” will generate an interrupt to Cortex_M0 IRQ[n]. When MCU_IRQ[n] is “1” (meaning an interrupt is asserted): Writing MCU_IRQ[n] “1” will clear the interrupt; writing MCU_IRQ[n] “0”: has no effect. **Note:** IRQ4 and IRQ15 are reserved in N572F072/P072.

5.3 Clock Controller

The clock controller generates the clock sources for the whole chip. It includes all AMBA interface modules and all peripheral clocks, ADC, and so on. The controller also implements the power control function, include the individually clock on or off control register, clock source select and the divided number from clock source. These functions minimize the extra power consumption and the chip run on the just clock condition. The chip will enter power-down mode after setting both the PD_WAIT_CPU and PWR_DOWN bits, asserting the signal SLEEPHOLDACKn and later the CPU Cortex-M0 executes the WFI or the WFE instruction. On the power down mode, the controller turns off the internal oscillator and system clock related circuit (except 32K oscillator) to reduce the power consumption to minimum.

5.3.1 Clock Generator

The clock generator consists of 4 sources which are listed below:

- One external 32KHz crystal
- One internal 48MHz RC oscillator.
- First PLL output, named as PLL1. PLL1 clock source is from 32KHz crystal, with fixed 6MHz clock output
- Second PLL output, named as PLL2. PLL2 clock source can be from 48MHz RC oscillator or PLL1 output, and its maximum frequency output is 48MHz

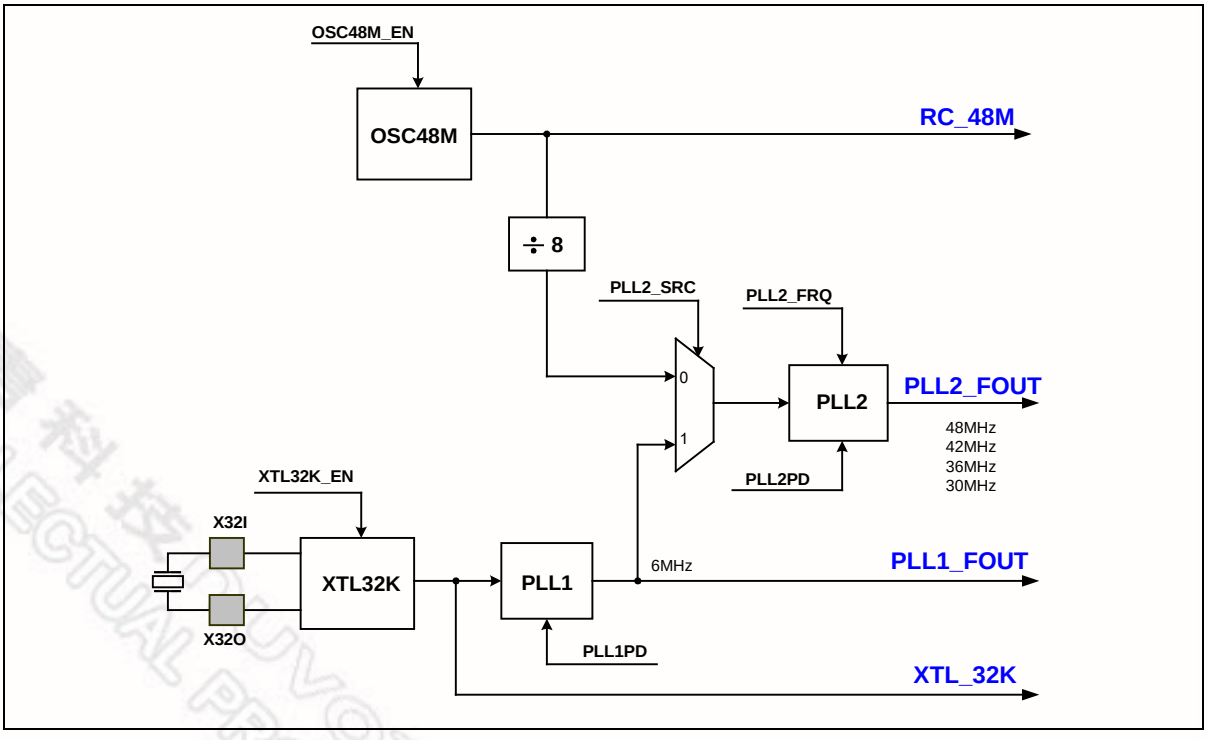


Figure 5-2 Clock generator block diagram

5.3.2 System Clock

The system clock has 4 clock sources which were generated from clock generator block. The clock source switch depends on the register HCLKSEL(CLK_CLKSEL0[2:0]). The block diagram is listed below.

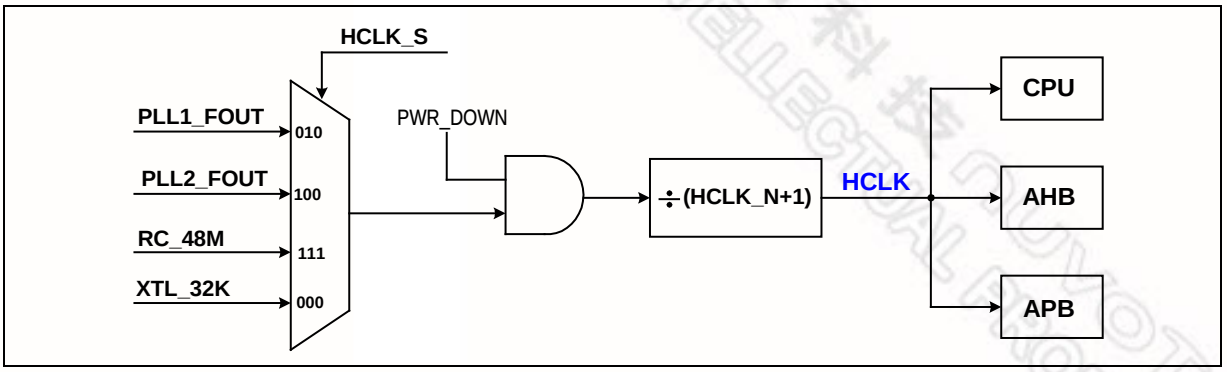


Figure 5-3 System Clock Block Diagram

5.3.3 Peripheral Clock

The peripheral clock has different clock source switch setting which depends on the different peripheral. Please refer to the descriptions in CLK_CLKSEL1 and CLK_APBCLK registers.

5.3.4 Power Down Mode Clock

When chip enters into power down mode, some clock sources and peripherals clock and system clock will be disabled. Some clock sources and peripherals clock can still be active in power down mode.

Those clocks which still can be kept active are listed below:

- Clock Generator
  - External 32K crystal clock
- Peripheral Clock (When these IP adopt 32KHz as clock source)
  - Watch Dog Clock
  - RTC Clock
  - Timer 0/1/2/F Clock
  - PWM Clock

5.3.5 Clock Control Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
CLK Base Address: **CLK_BA = 0x5000_0200**				
CLK_PWRCTL	CLK_BA + 0x00	R/W	System Power Control Register	0x0000_001C
CLK_AHBCLK	CLK_BA + 0x04	R/W	AHB Device Clock Enable Control Register	0x0000_0004
CLK_APBCLK	CLK_BA + 0x08	R/W	APB Device Clock Enable Control Register	0x0000_0000
CLK_CLKSEL0	CLK_BA + 0x10	R/W	Clock Source Select Control Register 0	0x0000_0007
CLK_CLKSEL1	CLK_BA + 0x14	R/W	Clock Source Select Control Register 1	0x0070_0004
CLK_CLKDIV	CLK_BA + 0x18	R/W	Clock Divider Number Register	0x0000_0003
CLK_PLLCON	CLK_BA + 0x20	R/W	PLL Control Register	0x0000_0003

5.3.6 Clock Control Register Description

System Power Control Register (CLK_PWRCTL)

Except the BIT[6], all the other bits are protected. To program these bits needs an open lock sequence, write “59h”, “16h”, “88h” to register SYS_REGLCTL to un-lock these bits. Refer to the register SYS_REGLCTL at address SYS_BA + 0x100.

Register	Offset	R/W	Description	Reset Value
CLK_PWRCTL	CLK_BA + 0x00	R/W	System Power Control Register	0x0000_001C

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							PD_WAIT_CPU
7	6	5	4	3	2	1	0
PWR_DOWN	INTSTS	WINT_EN	WU_DLY	VOUTX_PD	OSC48M_EN	XTL32K_EN	Reserved

Bits	Description	
[31:9]	Reserved	Reserved.
[8]	PD_WAIT_CPU	**This Bit Controls The Power Down Entry Condition** Please refer to PWR_DOWN bit for the usage of PD_WAIT_CPU bit. The following is a brief description of PD_WAIT_CPU bit. 0 = Chip is at normal mode. Note that PWR_DOWN cannot be set to 1 when PD_WAIT_CPU value remains at 0, otherwise the chip may not wake up normally. 1 = Chip waits to enter power-down mode.

<table>
<tr>
<td>[7]</td>
<td>PWR_DOWN</td>
<td>
System Power Down Active Or Enable Bit
To make the programming simple, In order to allow the chip enter power-down mode, both the PWR_DOWN bit and PD_WAIT_CPU bit must be set to 1. To reset both PWR_DOWN=0 and PD_WAIT_CPU=0 can make sure the chip operates at normal mode.
The below truth table summarizes the chip statuses caused by the setting of PWR_DOWN bit and PD_WAIT_CPU bit.
<table>
<tr><th>PD_WAIT_CPU</th><th>PWR_DOWN</th><th>Function</th></tr>
<tr><td>0</td><td>0</td><td>Chip is at normal operation.</td></tr>
<tr><td>0</td><td>1</td><td>This setting is prohibited.
The chip will shut down immediately but may not wake up normally.</td></tr>
<tr><td>1</td><td>0</td><td>Chip is at normal operation. This condition exists after the chip wakes up</td></tr>
<tr><td>1</td><td>1</td><td>This setting is a prerequisite for entering power-down mode. The chip still operates normally at this stage, but it is already standing by power-down entry condition.
The complete procedure to enter power-down mode is described in below paragraph.</td></tr>
</table>
When both the PD_WAIT_CPU and PWR_DOWN bit are set to “1”, the chip still operates normally. Until the Cortex_M0 enters sleep state and asserts the signal SLEEPHOLDACKn, then the chip enters the power-down mode.
When the chip enters the power-down mode and all the chip clocks are disabled, except IP from the 32KHz clock.
It is suggested to follow the below three steps in order to enter power-down mode. The following illustrates the pseudo code for these three steps,
Step 1. CLK_PWRCTL[8]= PD_WAIT_CPU=1 and CLK_PWRCTL[7]= PWR_DOWN=1.
Step 2. 0xE000ED10[2]=1.
Step 3. __wfi();

When chip wakes up from power down mode, this bit is auto cleared. User needs to set this bit again for next power down.
In power down mode, the internal 48MHz OSC, PLL and all of the AMBA clocks are disabled. The 32KHz oscillator is still active if it has been working before entering power-down. So the RTC and WDT functions whose clocks are from 32KHz oscillator are active.
If system wants to avoid interrupt by WDT waking up from power down mode, the WDT should be turned off before entering the power down mode.
0 = Chip operates at normal mode.
1 = Chip is standing by power-down entry condition.
</td>
</tr>
<tr>
<td>[6]</td>
<td>WINT_STS</td>
<td>
Chip Power Down Wake Up Status Flag
Set by “power down wake up”, it indicates that resume from power down mode.
The flag is set if the GPIO, WDT or RTC wakeup.
Note: Write 1 to clear the bit.
</td>
</tr>
<tr>
<td>[5]</td>
<td>WINT_EN</td>
<td>
Enable Interrupt When Wake Up From Power Down Mode
0 = Disable,.
1 = Enable. The interrupt will occur when MCU wakes up from power down mode
</td>
</tr>
</table>

[4]	**WU_DLY**	**Enable The Wake Up Delay Counter** When the chip wakes up from idle mode, the clock control will delay some clock cycles to wait the internal 48MHz oscillator clock stable. 0 = Disable the clock cycles delay. 1 = Enable the clock cycles delay, the delay is 512 clock cycles.
[3]	**VOUTX_PD**	**Driving Out 3.0V (Through VOUTX Pad) LDO Control** After reset, this bit is “1”. 0 = 3.0V LDO (VOUTX) is enabled. 1 = 3.0V LDO (VOUTX) is disabled.
[2]	**OSC48M_EN**	**Internal 48MHz RC Oscillator Control** After reset, this bit is “1”. 0 = 48MHz oscillation is disabled. 1 = 48MHz oscillation is enabled.
[1]	**XTL32K_EN**	**External 32.768KHz Crystal Control** After reset, this bit is “0”. 0 = 32.768KHz Crystal is disabled. 1 = 32.768KHz Crystal is enabled.
[0]	**Reserved**	Reserved.

AHB Device Clock Enable Control Register (CLK_AHBCLK)

These register bits are used to enable/disable the clock source for AMBA, AHB (Advanced High-Performance Bus) blocks and peripherals.

Register	Offset	R/W	Description	Reset Value
CLK_AHBCLK	CLK_BA + 0x04	R/W	AHB Device Clock Enable Control Register	0x0000_0004

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

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3]	**APUCKEN**	**APU Clock Enable Control** 0 = To disable the APU engine clock. 1 = To enable the APU engine clock.
[2]	**ISPCKEN**	**Flash ISP Engine Clock Enable Control** 0 = To disable the Flash ISP engine clock. 1 = To enable the Flash ISP engine clock.
[1:0]	**Reserved**	Reserved.

APB Device Clock Enable Control Register (CLK_APBCLK)

These register bits are used to enable/disable clocks for APB (Advanced Peripheral Bus) peripherals. To enable the clocks write '1' to the appropriate bit. To reduce power consumption and disable the peripheral, write '0' to the appropriate bit.

Register	Offset	R/W	Description	Reset Value
CLK_APBCLK	CLK_BA + 0x08	R/W	APB Device Clock Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			**ADC_EN**	**Reserved**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**PWM_EN**	**Reserved**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**SPI1_EN**	**SPI0_EN**	**Reserved**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**TMRF_EN**	**TMR2_EN**	**TMR1_EN**	**TMR0_EN**	**RTC_EN**	**WDT_EN**

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28]	**ADC_EN**	**Audio Analog-Digital-Converter (ADC) Clock Enable Control** 0=Disable. 1=Enable.
[27:21]	**Reserved**	Reserved.
[20]	**PWM_EN**	**PWM Block Clock Enable Control** 0=Disable. 1=Enable.
[19:14]	**Reserved**	Reserved.
[13]	**SPI1_EN**	**SPI1 Clock Enable Control** 0=Disable. 1=Enable.
[12]	**SPI0_EN**	**SPI0 Clock Enable Control** 0=Disable. 1=Enable.
[11:6]	**Reserved**	Reserved.
[5]	**TMRF_EN**	**TimerF Clock Enable Control** 0=Disable. 1=Enable.

[4]	**TMR2_EN**	**Timer2 Clock Enable Control** 0=Disable. 1=Enable.
[3]	**TMR1_EN**	**Timer1 Clock Enable Control** 0=Disable. 1=Enable.
[2]	**TMR0_EN**	**Timer0 Clock Enable Control** 0=Disable. 1=Enable.
[1]	**RTC_EN**	**Real-Time-Clock APB Interface Clock Control** This bit is used to control the RTC APB clock only. The RTC engine clock source is from the 32.768KHz crystal. 0=Disable. 1=Enable.
[0]	**WDT_EN**	**Watchdog Clock Enable Control** This bit is the protected bit. To program this bit needs an open lock sequence, write “59h”, “16h”, “88h” to register SYS_REGLCTL to un-lock this bit. Refer to the register SYS_REGLCTL at address SYS_BA+0x100. The default bit value is set according to the Flash Controller User Configuration Register CONFIG[31]. 0=Disable. 1=Enable.

Clock Source Select Control Register 0 (CLK_CLKSEL0)

These bits are protected bits. To program these bits needs an open lock sequence, write “59h”, “16h”, “88h” to register SYS_REGLCTL to un-lock these bits. Refer to the register SYS_REGLCTL at address SYS_BA+0x100.

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL0	CLK_BA + 0x10	R/W	Clock Source Select Control Register 0	0x0000_0007

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

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4:3]	**STCLKSEL**	**MCU Cortex_M0 SysTick Clock Source Select** 00 = Clock source from HCLK/2. 01 = Clock source from XTL_32K. 1x = Clock source from RC_48M/8.
[2:0]	**HCLKSEL**	**HCLK Clock Source Select** 000 = Clock source from XTL_32K. 010 = Clock source from PLL1_FOUT. 100 = Clock source from PLL2_FOUT. 111 = Clock source from RC_48M. Others = equivalent with “111”. **Note:** 1. When power on, 48MHz RC is selected as HCLK clock source. 2. Before clock switch, the related clock sources (pre-select and new-select) must be turned on.

Clock Source Select Control Register 1 (CLK_CLKSEL1)

Register	Offset	R/W	Description	Reset Value
CLK_CLKSEL1	CLK_BA + 0x14	R/W	Clock Source Select Control Register 1	0x0070_0004

31	30	29	28	27	26	25	24
Reserved		PWMSEL		Reserved			
23	22	21	20	19	18	17	16
Reserved	TMRFSEL			Reserved	TMR2SEL		
15	14	13	12	11	10	9	8
Reserved	TMR1SEL			Reserved	TMR0SEL		
7	6	5	4	3	2	1	0
Reserved		SPI0SEL		ADCSEL		WDTSEL	

Bits	Description	
[31:30]	**Reserved**	Reserved.
[29:28]	**PWMSEL**	**PWM Timer Clock Source Select** 00 = Clock source from HCLK. 01 = Clock source from XTL_32K. 10 = Equivalent with “00”. 11 = Clock source from RC_48M.
[27:23]	**Reserved**	Reserved.
[22:20]	**TMRFSEL**	**TimerF Clock Source Select** 000 = Clock source from external XTL_32K/32,. 001 = Clock source from external XTL_32K/(4x32),. 110 = Clock source from RC_48M/65536,. 111 = Clock source from RC_48M/(4x65536),. Others = Equivalent with “000”.
[19]	**Reserved**	Reserved.
[18:16]	**TMR2SEL**	**Timer2 Clock Source Select** 000 = Clock source from HCLK. 001 = Clock source from XTL_32K. 010 = Equivalent with “000”. 011 = Clock source from external trigger. 1xx = Clock source from RC_48M.
[15]	**Reserved**	Reserved.

[14:12]	**TMR1SEL**	**Timer1 Clock Source Select** 000 = Clock source from HCLK. 001 = Clock source from XTL_32K. 010 = Equivalent with “000”. 011 = Clock source from external trigger. 1xx = Clock source from RC_48M.
[11]	**Reserved**	Reserved.
[10:8]	**TMR0SEL**	**Timer0 Clock Source Select** 000 = Clock source from HCLK. 001 = Clock source from XTL_32K. 010 = Equivalent with “000”. 011 = Clock source from external trigger. 1xx = Clock source from RC_48M.
[7:6]	**Reserved**	Reserved.
[5:4]	**SPI0SEL**	**SPI0 Clock Source Select** 00 = Clock source from HCLK. 01 = Clock source from PLL2_FOUT. 1x = Clock source from RC_48M.
[3:2]	**ADCSEL**	**ADC Clock Source Select** 00 = Clock source from PLL2_FOUT,. 01 = Clock source from HCLK,. 1x = Clock source from RC_48M.
[1:0]	**WDTSEL**	**Watchdog Timer Clock Source Selection (Write Protect)** These bits are protected bits. To program these bits needs an open lock sequence, write “59h”, “16h”, “88h” to SYS_REGLCTL to un-lock these bits. Refer to the register SYS_REGLCTL at address SYS_BA+0x100.. 00 = Clock source from HCLK/2048,. 01 = Clock source from XTL_32K,. 10 = Clock source from PLL1_FOUT,. 11 = Clock source from RC_48M/2.

Clock Divider Register (CLK_CLKDIV)

Register	Offset	R/W	Description	Reset Value
CLK_CLKDIV	CLK_BA + 0x18	R/W	Clock Divider Number Register	0x0000_0003

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

Bits	Description	
[31:28]	**Reserved**	Reserved.
[23:16]	**ADCDIV**	**ADC Clock Divide Number From ADC Clock Source** The ADC clock frequency ADCLK = (ADC clock source frequency) / (ADCDIV + 1).
[15:4]	**Reserved**	Reserved.
[3:0]	**HCLKDIV**	**HCLK Clock Divide Number From HCLK Clock Source** The HCLK clock frequency = (HCLK clock source frequency) / (HCLKDIV + 1).

PLL Control Register (CLK_PLLCON)

Register	Offset	R/W	Description	Reset Value
CLK_PLLCON	CLK_BA + 0x20	R/W	PLL Control Register	0x0000_0003

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved			PLL2_FRQ		PLL2_SRC	PLL1PD	PLL2PD

Bits	Descriptions	
[31:5]	**Reserved**	Reserved
[4:3]	**PLL2_FRQ**	**PLL2 Clock Output Frequency Select.** 00 = 30MHz 01 = 36MHz 10 = 42MHz 11 = 48MHz
[2]	**PLL2_SRC**	**PLL2 Input Source Clock Select.** PLL2 input clock is 6MHz, its source can be one of followings. 0 = PLL2 source clock from RC_48M/8, 1 = PLL2 source clock from PLL1_FOUT.
[1]	**PLL1PD**	PLL1 Power Down Mode. 0 = PLL1 is in normal mode, 1 = PLL1 is in power-down mode.
[0]	**PLL2PD**	PLL2 Power Down Mode. 0 = PLL2 is in normal mode, 1 = PLL2 is in power-down mode.

5.4 SRAM

5.4.1 Overview

N572F072/P072 equips with 8KB SRAM for program code and data storage. It's an AHB slave interface. It supports byte, half, and word read, and write access. An arbiter design enables Cortex-M0 CPU and DMA to access SRAM as the same without any data conflict.

5.4.2 Block Diagram

The block diagram of SRAM Controller with rotation engine is depicted as following:

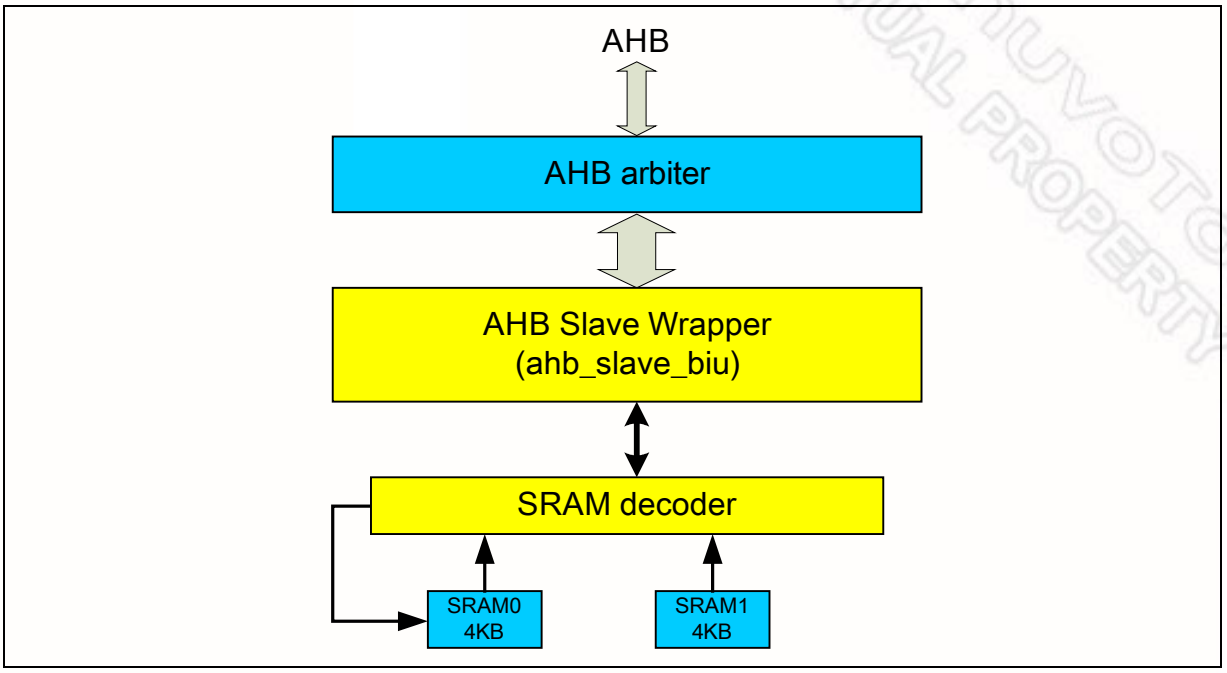


Figure 5-4 SRAM Controller Block Diagram

5.5 General Purpose I/O

5.5.1 Overview and Features

There are 32 pins of General Purpose I/O shared with special feature functions. These pins are arranged in 2 groups, PA and PB. Each pin of the 32 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, open-drain or quasi-bidirectional mode.

In input mode, the I/O pin is in tri-state (high impedance) without output drive capability.

In output mode, the I/O pin supports digital output function with source/sink current capability.

In open-drain mode, the I/O pin supports digital output function but only with sink current capability, an additional pull-up resistor is needed for driving high state.

In quasi-mode, the I/O pin supports digital output and input function at the same time but the source current is only up to hundreds μA. Each I/O pin equips an individual pull-up resistor which is dependent on pin status. The pull-up resistor is about 110KΩ~300KΩ for V_{DD} is from 5.0V to 2.5V.

5.5.1.1 *Open-Drain mode explanation*

For Px_MODE*n* = 10b the GPIO Px[*n*] pin is in Open-Drain mode.

If the bit value in the corresponding bit Px_DOUT[n] is “0”, the pin drive a “low” output on the pin.

If the bit value in the corresponding bit Px_DOUT[n] is “1”, the pin output drive is disabled and the pin status is controlled by external pull-high resistor.

.

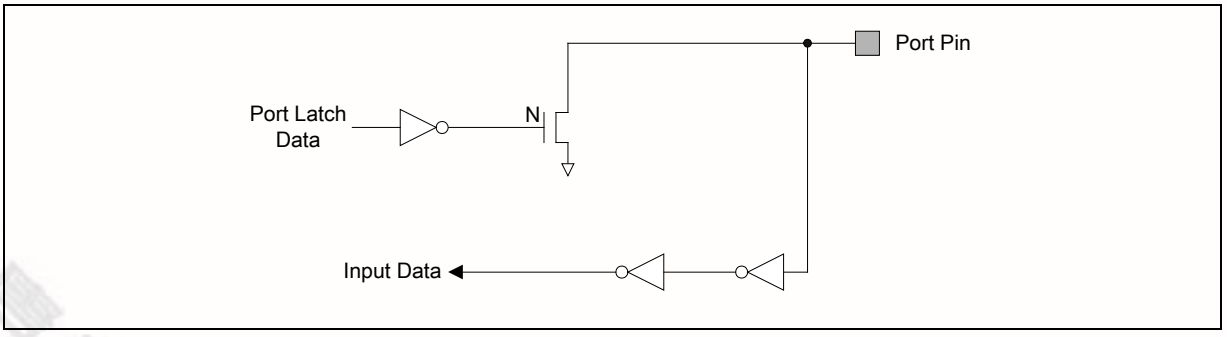


Figure 5-5 Open-Drain Output

5.5.1.2 *Quasi-bidirectional Mode Explanation*

For Px_MODE*n* = 11b the GPIO Px[*n*] pin is in Quasi-bidirectional mode.

If the bit value in the corresponding bit Px_DOUT[n] is “0”, the pin drive a “low” output on the pin.

If the bit value in the corresponding bit Px_DOUT[n] is “1”, the pin will check the pin value. If pin value is high, no action takes. If pin state is low, it will drive 2 clock cycles “high” and then disable the output drive and then the pin status is controlled by internal pull-high MOS.

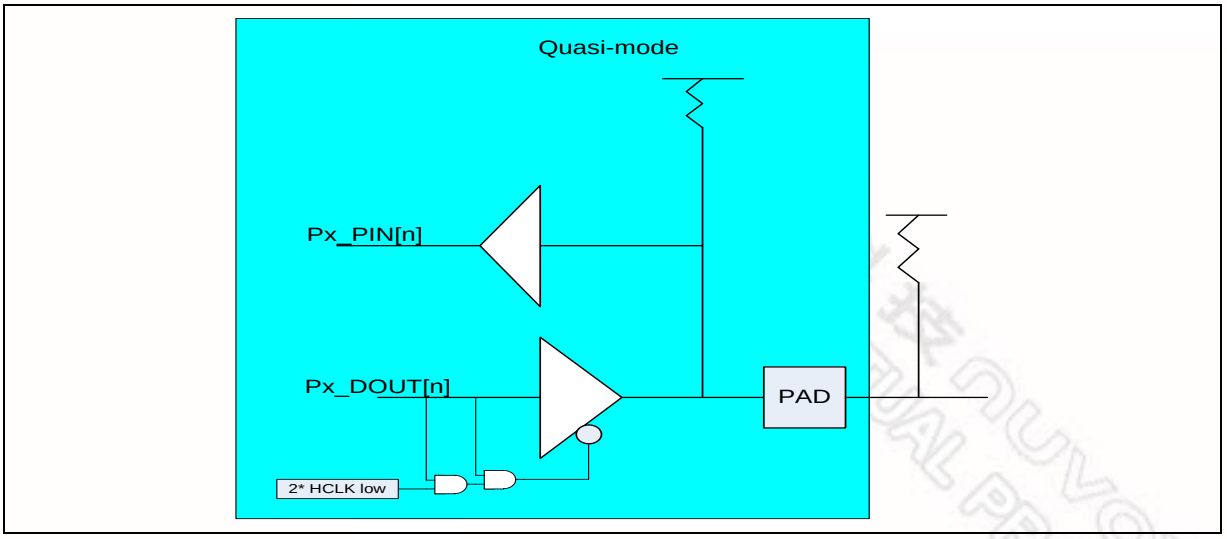


Figure 5-6 Quasi-bidirectional GPIO Mode

5.5.2 GPIO Control Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
GPIO Base Address: **GPIO_BA = 0x5000_4000**				
PA_MODE	GPIO_BA+0x000	R/W	GPIO PA Pin I/O Mode Control	0xFFFF_FFFF
PA_DINOFF	GPIO_BA+0x004	R/W	GPIO PA Digital Input Path Disable Control	0x0000_0000
PA_DOUT	GPIO_BA+0x008	R/W	GPIO PA Data Output Value	0x0000_FFFF
PA_DATMSK	GPIO_BA+0x00C	R/W	GPIO PA Data Output Write Mask	0x0000_0000
PA_PIN	GPIO_BA+0x010	R	GPIO PA Pin Value	0x0000_XXXX
PA_INTTYPE	GPIO_BA+0x018	R/W	GPIO PA Interrupt Trigger Type	0x0000_0000
PA_INTEN	GPIO_BA+0x01C	R/W	GPIO PA Interrupt Enable	0x0000_0000
PA_INTSRC	GPIO_BA+0x020	R/W	GPIO PA Interrupt Source Flag	0x0000_0000
PB_MODE	GPIO_BA+0x040	R/W	GPIO PB Pin I/O Mode Control	0xFFFF_FFFF
PB_DINOFF	GPIO_BA+0x044	R/W	GPIO PB Digital Input Path Disable Control	0x0000_0000
PB_DOUT	GPIO_BA+0x048	R/W	GPIO PB Data Output Value	0x0000_FFFF
PB_DATMSK	GPIO_BA+0x04C	R/W	GPIO PB Data Output Write Mask	0x0000_0000
PB_PIN	GPIO_BA+0x050	R	GPIO PB Pin Value	0x0000_XXXX
PB_INTTYPE	GPIO_BA+0x058	R/W	GPIO PB Interrupt Trigger Type	0x0000_0000
PB_INTEN	GPIO_BA+0x05C	R/W	GPIO PB Interrupt Enable	0x0000_0000
PB_INTSRC	GPIO_BA+0x060	R/W	GPIO PB Interrupt Source Flag	0x0000_0000

5.5.3 GPIO Control Register Description

GPIO Port [A/B] I/O Mode Control (Px_MODE)

Register	Offset	R/W	Description	Reset Value
PA_MODE	GPIO_BA+0x000	R/W	GPIO PA Pin I/O Mode Control	0xFFFF_FFFF
PB_MODE	GPIO_BA+0x040	R/W	GPIO PB Pin I/O Mode Control	0xFFFF_FFFF

31	30	29	28	27	26	25	24
MODE15		MODE14		MODE13		MODE12	
23	22	21	20	19	18	17	16
MODE11		MODE10		MODE9		MODE8	
15	14	13	12	11	10	9	8
MODE7		MODE6		MODE5		MODE4	
7	6	5	4	3	2	1	0
MODE3		MODE2		MODE1		MODE0	

Bits	Description	
[2n+1 :2n] n=0,1..15	**MODEn**	**Port [A/B] Pin[N] I/O Mode Control** Each GPIO Px pin has four modes: 00 = GPIO Px[n] pin is in INPUT mode. 01 = GPIO Px[n] pin is in OUTPUT mode. 10 = GPIO Px[n] pin is in Open-Drain mode. 11 = GPIO Px[n] pin is in Quasi-bidirectional mode.

GPIO Port [A/B] Digital Input Path Disable (Px_DINOFF)

Register	Offset	R/W	Description	Reset Value
PA_DINOFF	GPIO_BA+0x004	R/W	GPIO PA Digital Input Path Disable Control	0x0000_0000
PB_DINOFF	GPIO_BA+0x044	R/W	GPIO PB Digital Input Path Disable Control	0x0000_0000

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

Bits	Description	
[n+16] n=0,1..15	**DINOFF[n]**	**Port [A/B] Pin[N] Digital Input Path Disable Control** 0 = Px.n Digital input path Enable (Default). 1 = Px.n Digital input path Disable (digital input tied to low).
[15:0]	**Reserved**	Reserved.

GPIO Port [A/B] Data Output Value (Px_DOUT)

Register	Offset	R/W	Description	Reset Value
PA_DOUT	GPIO_BA+0x008	R/W	GPIO PA Data Output Value	0x0000_FFFF
PB_DOUT	GPIO_BA+0x048	R/W	GPIO PB Data Output Value	0x0000_FFFF

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

Bits	Description	
[31:16]	**Reserved**	Reserved
[15:0]	**DOUT**	**Port [A/B] Pin[N] Output Value** Each of these bits controls the status of a GPIO pin when the GPIO pin is configured as output, open-drain or quasi-bidirectional mode. 0 = GPIO port [A/B] Pin[n] will drive Low if the corresponding output mode bit is set. 1 = GPIO port [A/B] Pin[n] will drive High if the corresponding output mode bit is set.

GPIO Port [A/B] Data Output Write Mask (Px_DATMSK)

Register	Offset	R/W	Description	Reset Value
PA_DATMSK	GPIO_BA+0x00C	R/W	GPIO PA Data Output Write Mask	0x0000_0000
PB_DATMSK	GPIO_BA+0x04C	R/W	GPIO PB Data Output Write Mask	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved
[15:0]	**DATMSK[n]**	**Port [A/B] Pin[N] Data Output Write Mask** These bits are used to protect the corresponding register of Px_DOUT[n]. When set the DATMSK[n] to “1”, the corresponding Px_DOUT[n] bit is writing protected. 0 = The corresponding Px_DOUT[n] bit can be updated. 1 = The corresponding Px_DOUT[n] bit is read only.

GPIO Port [A/B] Pin Value (Px_PIN)

Register	Offset	R/W	Description	Reset Value
PA_PIN	GPIO_BA+0x010	R	GPIO PA Pin Value	0x0000_XXXX
PB_PIN	GPIO_BA+0x050	R	GPIO PB Pin Value	0x0000_XXXX

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

Bits	Description	
[31:16]	**Reserved**	Reserved
[15:0]	**PIN[n]**	**Port [A/B] Pin[N] Pin Values** Each bit of the register reflects the actual status of the respective Px.n pin. If the bit is 1, it indicates the corresponding pin status is high; else the pin status is low.

GPIO Port [A/B] Interrupt Mode Control (Px_INTTYPE)

Register	Offset	R/W	Description	Reset Value
PA_INTTYPE	GPIO_BA+0x018	R/W	GPIO PA Interrupt Trigger Type	0x0000_0000
PB_INTTYPE	GPIO_BA+0x058	R/W	GPIO PB Interrupt Trigger Type	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved
[15:0]	**TYPE[n]**	**Port [A/B] Pin[N] Edge Or Level Detection Interrupt Trigger Type Control** TYPE[n] is used to control whether the interrupt mode is level triggered or edge triggered. If the interrupt mode is level triggered, the input source is sampled by one HCLK clock to generate the interrupt 0 = Edge triggered interrupt. 1 = Level triggered interrupt. **Note:** If level triggered interrupt is selected, then only one level can be selected in the Px_INTEN register. If both levels are set, the setting is ignored and no interrupt will occur

GPIO Port [A/B] Interrupt Enable Control (Px _INTEN)

Register	Offset	R/W	Description	Reset Value
PA_INTEN	GPIO_BA+0x01C	R/W	GPIO PA Interrupt Enable	0x0000_0000
PB_INTEN	GPIO_BA+0x05C	R/W	GPIO PB Interrupt Enable	0x0000_0000

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

Bits	Description	
[n+16] n=0,1..15	**RHIEN[n]**	**Port [A/B] Interrupt Enable By Input Rising Edge Or Input Level High** RHIEN[n] is used to enable the rising/high-level interrupt for each of the corresponding input Px.n pin. To set “1” also enables the pin wake-up function. When setting the RHIEN[n] (Px_INTEN[n+16]) bit to 1 : If the interrupt is configured as level trigger mode (TYPE[n] is set to 1), one interrupt will occur while the input Px.n state is at high level. If the interrupt is configured as edge trigger mode (TYPE[n] is set to 0), one interrupt will occur while he input Px.n state changes from low to high. 0 = Disable Px.n for low-to-high or level-high interrupt. 1 = Enable Px.n for low-to-high or level-high interrupt.
[n] n=0,1..15	**FLIEN[n]**	**Port [A/B] Interrupt Enable By Input Falling Edge Or Input Level Low** FLIEN[n] is used to enable the falling/low-level interrupt for each of the corresponding input Px.n pin. To set “1” also enables the pin wake-up function. When setting the FLIEN[n] (Px_INTEN[n]) bit to 1 : If the interrupt is configured as level trigger mode (TYPE[n] is set to 1), one interrupt will occur while the input Px.n state is at low level. If the interrupt is configured as edge trigger mode (TYPE[n] is set to 0), one interrupt will occur while he input Px.n state changes from high to low. 0 = Disable Px.n for low-level or high-to-low interrupt. 1 = Enable Px.n for low-level or high-to-low interrupt.

GPIO Port [A/B] Interrupt Source Flag(Px_INTSRC)

Register	Offset	R/W	Description	Reset Value
PA_INTSRC	GPIO_BA+0x020	R/W	GPIO PA Interrupt Source Flag	0x0000_0000
PB_INTSRC	GPIO_BA+0x060	R/W	GPIO PB Interrupt Source Flag	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved
[15:0]	**INTSRC[n]**	**Port [A/B] Interrupt Source Flag** Read operation: 0 = No interrupt from Px.n. 1 = Px.n generated an interrupt. Write operation: 0 = No action. 1 = Clear the corresponding pending interrupt.

5.6 PWM Generator and Capture Timer

5.6.1 Introduction

The N572F072/P072 has 1 PWM timer. The PWM timer has 1 prescaler, 1 clock divider, 5 clock selectors, one 16-bit counter, four 16-bit comparators, and two Dead-Zone generators.

Clock divider provides PWM timer with 5 clock sources (1, 1/2, 1/4, 1/8, 1/16). PWM timer receives its own clock signal from clock divider which receives clock from 8-bit prescaler. The 16-bit counter receives clock signal from clock selector and can be used to handle one PWM period. The 16-bit comparator compares number in counter with threshold number in register loaded previously to generate PWM duty cycle.

The clock signal from clock divider is called PWM clock. Dead-Zone generator utilize PWM clock as clock source. Once Dead-Zone generator is enabled, output of two PWM comparators is blocked. Two output pin are used as Dead-Zone generator output signal to control off-chip power device. Dead-Zone generator 0 is used to control outputs of PWM0 & PWM1, and Dead-Zone generator 1 is used to control outputs of PWM2 & PWM3.

When 16-bit down counter reaches zero, the interrupt request is generated to inform CPU that time is up. When counter reaches zero, if counter is set as toggle mode, it is reloaded automatically and starts to generate next cycle. User can set counter as one-shot mode instead of toggle mode. If counter is set as one-shot mode, counter will stop and generate one interrupt request when it reaches zero.

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

The PWM timer includes a capture channel. The Capture input and PWM output share the PWM timer. Therefore user must setup the PWM timer before turn on Capture feature. After enabling Capture feature, the Capture always latches PWM counter to CRLR register when capture input has a rising transition and latches PWM counter to CFLR register when capture input has a falling transition. Capture input interrupt is programmable by setting PWM_CAPCTL[1] (Rising latch Interrupt enable) and PWM_CAPCTL[2] (Falling latch Interrupt enable) to decide the condition of interrupt event. Whenever Capture issues an interrupt, the PWM counter will be reloaded at this moment.

There is only an interrupt from PWM to interrupt controller. PWM Timer and Capture input share the same interrupt. Therefore, PWM function and Capture function cannot be used at the same time.

5.6.2 Features

- One 8-bit prescaler and clock divider
- Five clock input selector
- One 16-bit counters and four 16-bit comparators
- Two Dead-Zone generator
- Four PWM output, one capture function

5.6.3 PWM Generator Architecture

The following figures illustrate the architecture of the PWM generator.

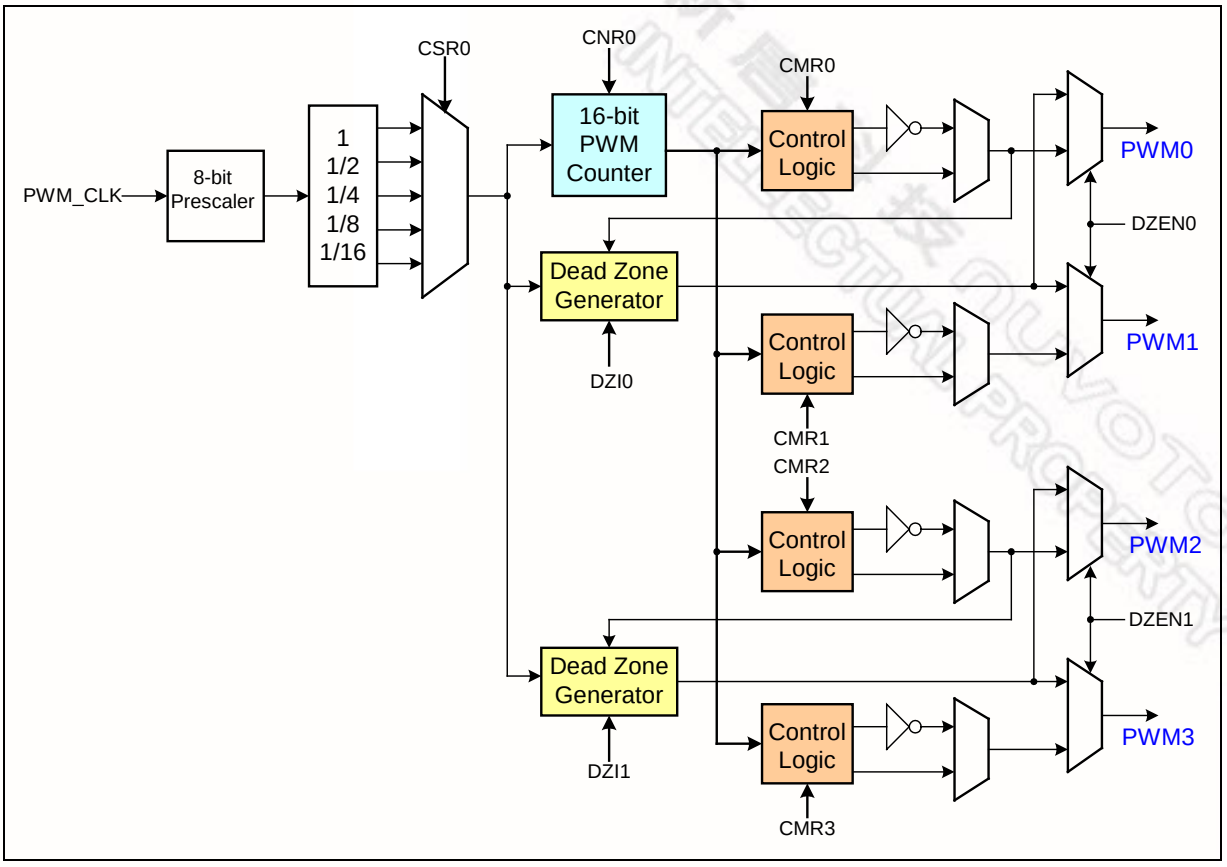


Figure 5-7 PWM Generator Architecture Diagram

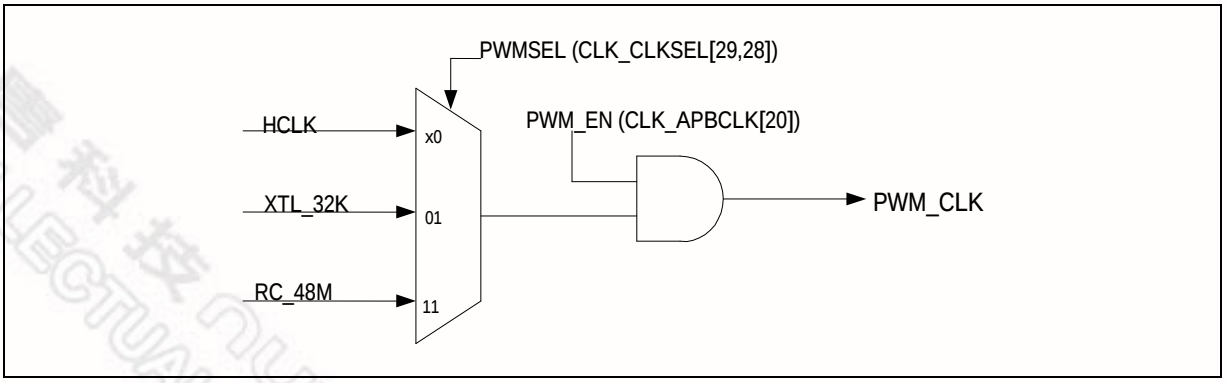


Figure 5-8 PWM Generator Clock Source Control

5.6.4 PWM-Timer Operation

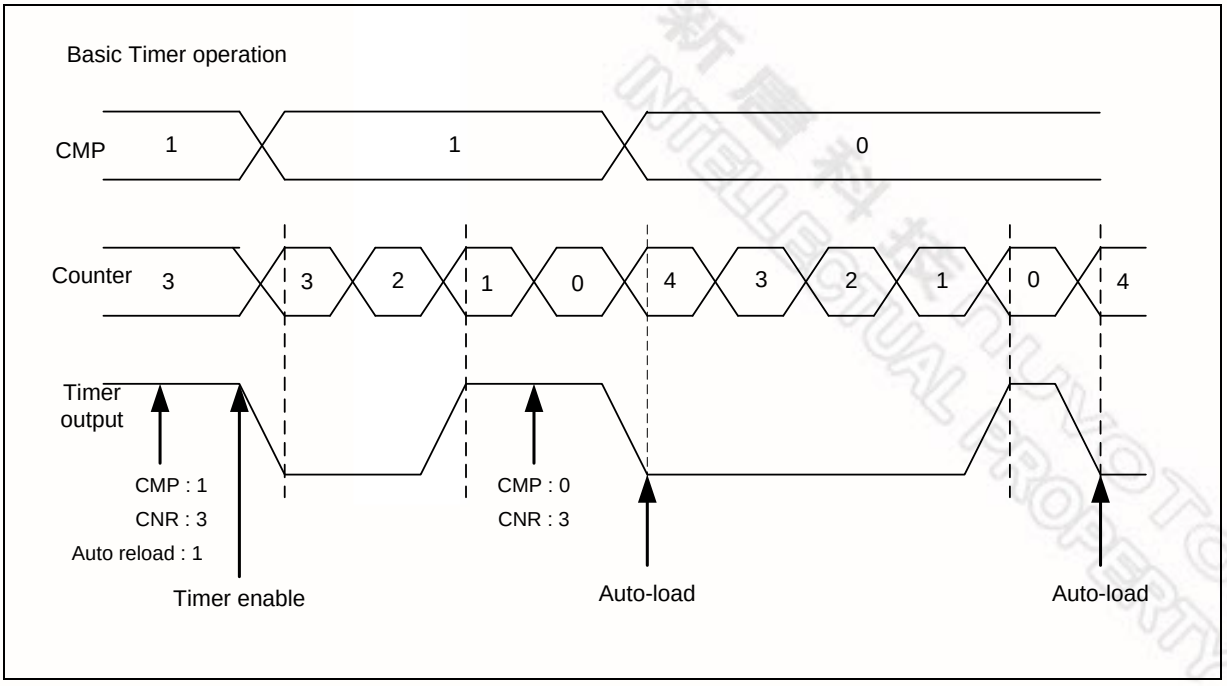


Figure 5-9 PWM Timer Operation Timing

5.6.5 PWM Auto-reload

The counter value can be written into PWM_PERIOD and current counter value can be read from PWM_CNT.

The auto-reload operation copies from PWM_PERIOD to down-counter when down-counter reaches zero. If PWM_PERIOD is set as zero, counter will be halt when counter counts to zero. If auto-reload bit is set as zero, counter will be stopped immediately.

.

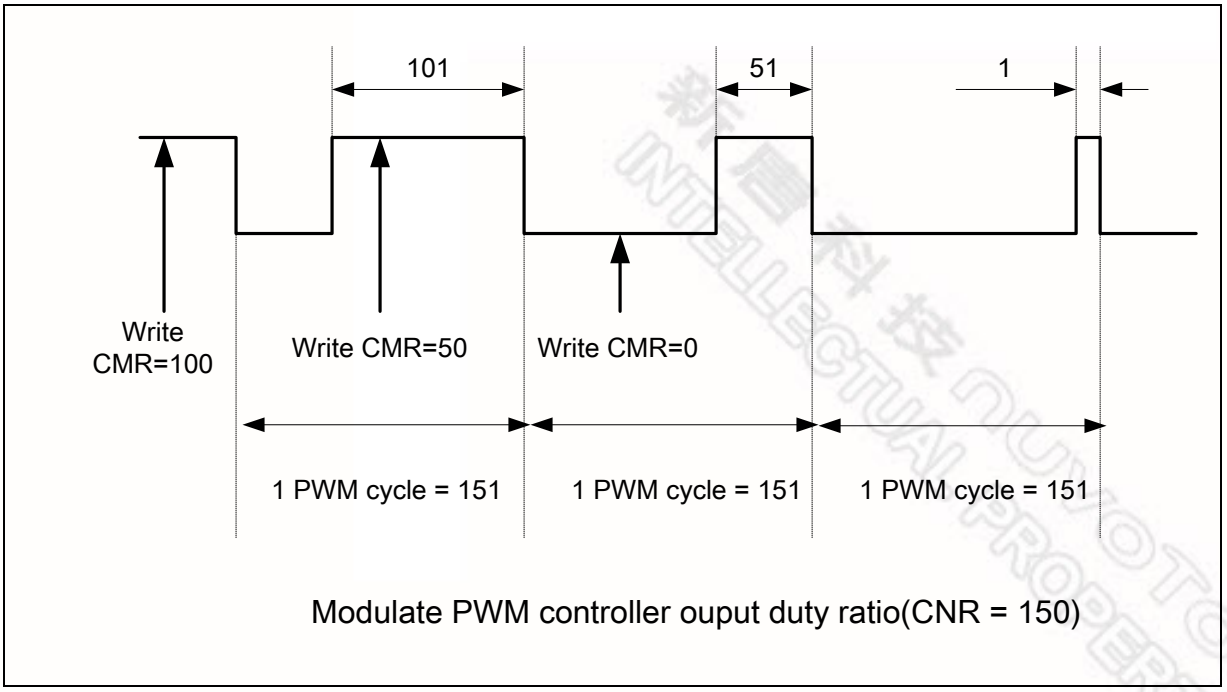


Figure 5-10 PWM Controller Output Duty Ratio.

5.6.6 Dead-Zone Generator

N572F072/P072 PWM is implemented with Dead Zone generator. They are built for power device protection. This function enables generation of a programmable time gap at the rising of PWM output waveform. User can program PWM_CLKPSC[31:24] and PWM_CLKPSC[23:16] to determine the two Dead Zone interval respectively.

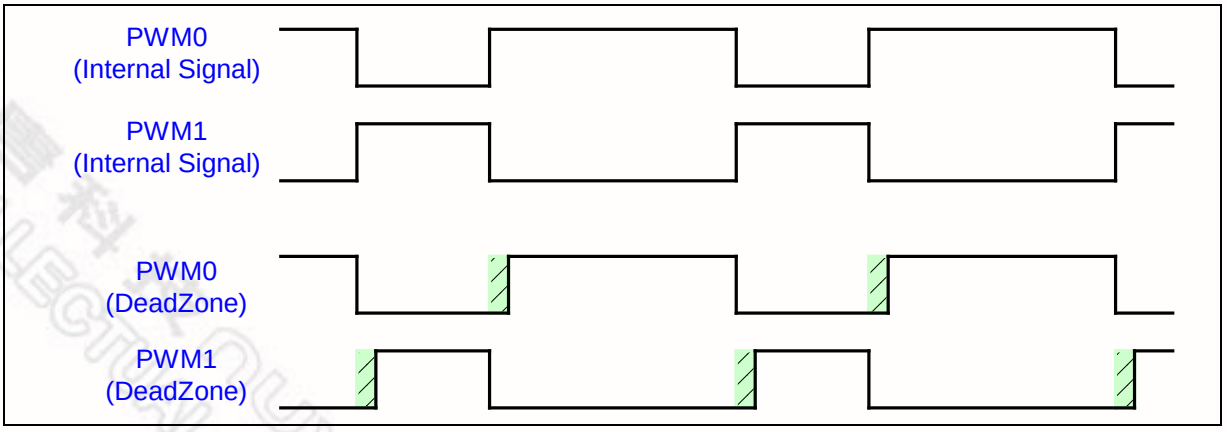


Figure 5-11 Dead Zone Generation Operation

5.6.7 PWM-Timer Start Procedure

The following procedure is recommended for starting a PWM generator.

1. Setup clock selector (PWM_CLKDIV)
2. Setup prescaler and dead zone interval (PWM_CLKPSC)
3. Setup inverter on/off, dead zone generator on/off, auto-reload/one-shot mode and PWM timer off (PWM_CTL)
4. Setup comparator register (PWM_CMPDATn) to set PWM duty cycle.
5. Setup PWM down-counter register (PWM_PERIOD) to set PWM period.
6. Setup interrupt enable register (PWM_INTEN)
7. Setup PWM output enable (PWM_PCEN)
8. Setup the corresponding GPIO pins to PWM function (SYS_GPB_MFP)
9. Enable PWM timer start running (PWM_CTL)

5.6.8 PWM-Timer Stop Procedure

Method 1:

Set 16-bit down counter (PWM_PERIOD) as 0, and monitor PWM_CNT (current value of 16-bit down-counter). When PWM_CNT reaches to 0, disable PWM-Timer (PWM_CTL). ***(Recommended)***

Method 2:

Set 16-bit down counter (PWM_PERIOD) as 0. When interrupt request occurs, disable PWM-Timer (PWM_CTL). ***(Recommended)***

Method 3:

Disable PWM-Timer directly (PWM_CTL). ***(Not recommended)***

5.6.9 Capture Start Procedure

1. Setup clock selector (PWM_CLKDIV)
2. Setup prescaler and dead zone interval (PWM_CLKPSC)
3. Setup inverter on/off, dead zone generator on/off, auto-reload/one-shot mode and PWM timer off (PWM_CTL)
4. Setup PWM down-counter register (PWM_PERIOD)
5. Setup capture register (PWM_CAPCTL)
6. Enable PWM timer start running (PWM_CTL)

5.6.10 Capture Timer Operation

The Capture and PWM output function share the same PWM timer. The capture timer latches PWM-counter to PWM_RCAPDAT when input channel has a rising transition and latches PWM-counter to PWM_FCAPDAT when input channel has a falling transition. Capture interrupt is programmable by setting PWM_CAPCTL[1] (Rising latch Interrupt enable) and PWM_CAPCTL[2] (Falling latch Interrupt enable) to decide the condition of interrupt occurrence. Whenever the Capture module issues a capture interrupt, the corresponding PWM counter will be reloaded with PERIOD at this moment. Note that the corresponding GPIO pin must be configured as their alternate function before Capture function is enabled.

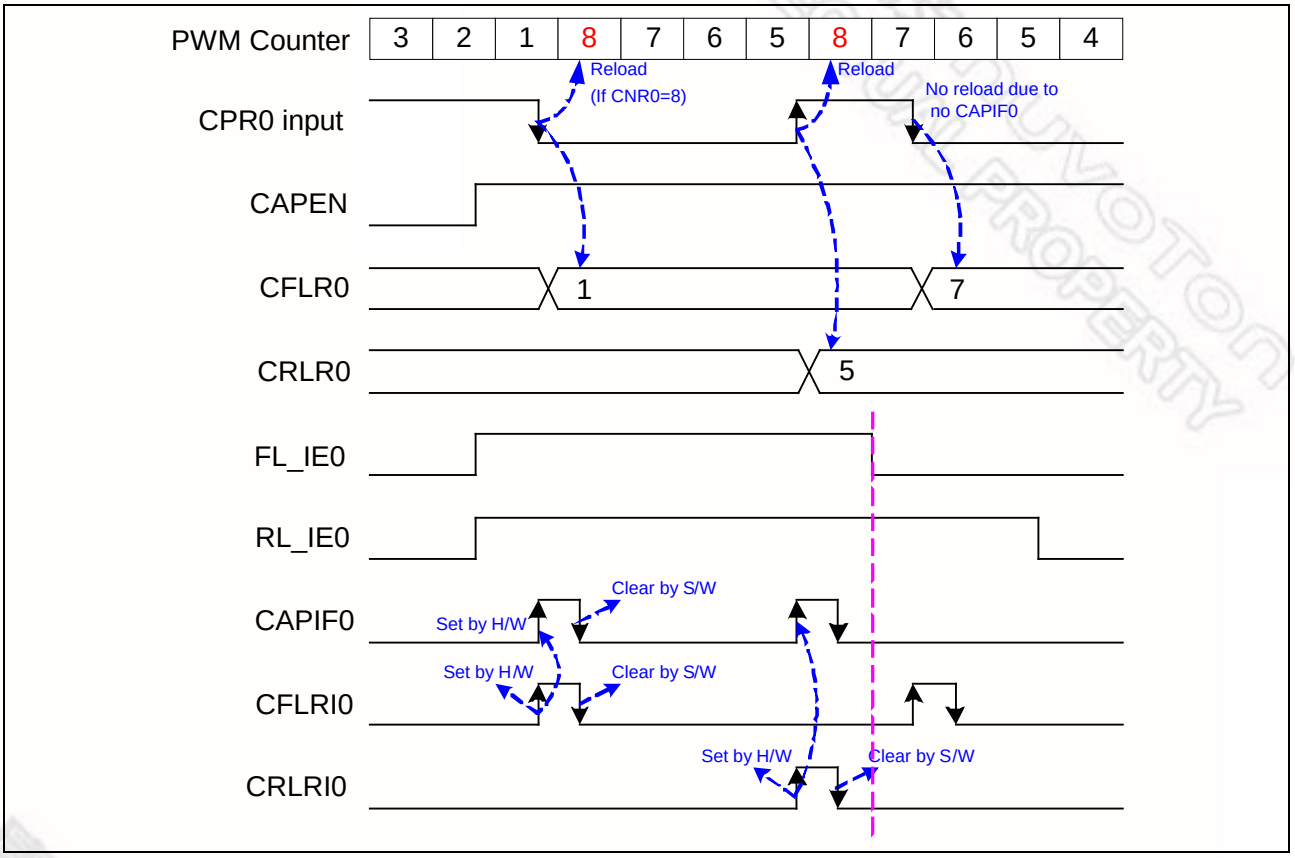


Figure 5-12 Capture Operation Timing

Figure 5-12 demonstrates the case where PERIOD = 8:

1. The PWM counter will be reloaded with PERIOD=8 at the time of interrupt occurrence when both falling and rising interrupt enabled.
2. The channel low pulse width is given by (PERIOD - RCAPDAT).
3. The channel high pulse width is given by (PERIOD - FCAPDAT).

5.6.11 Register Map

R: read only, **W**: write only, **R/W**: both read and write, **C**: Only value 0 can be written

Register	Offset	R/W	Description	Reset Value
PWM Base Address: **PWM_BA = 0x4004_0000**				
PWM_CLKPSC	PWM_BA+0x000	R/W	PWM Prescaler Register	0x0000_0000
PWM_CLKDIV	PWM_BA+0x004	R/W	PWM Clock Select Register	0x0000_0000
PWM_CTL	PWM_BA+0x008	R/W	PWM Control Register	0x0000_0000
PWM_PERIOD	PWM_BA+0x00C	R/W	PWM Period Register	0x0000_0000
PWM_CMPDAT0	PWM_BA+0x010	R/W	PWM Comparator Register 0	0x0000_0000
PWM_CNT	PWM_BA+0x014	R	PWM Counter Register	0x0000_0000
PWM_CMPDAT1	PWM_BA+0x01C	R/W	PWM Comparator Register 1	0x0000_0000
PWM_CMPDAT2	PWM_BA+0x028	R/W	PWM Comparator Register 2	0x0000_0000
PWM_CMPDAT3	PWM_BA+0x034	R/W	PWM Comparator Register 3	0x0000_0000
PWM_INTEN	PWM_BA+0x040	R/W	PWM Interrupt Enable Register	0x0000_0000
PWM_INTSTS	PWM_BA+0x044	R/W	PWM Interrupt Flag Register	0x0000_0000
PWM_CAPCTL	PWM_BA+0x050	R/W	Capture Control Register	0x0000_0000
PWM_RCAPDAT	PWM_BA+0x058	R	Capture Rising Latch Register	0x0000_0000
PWM_FCAPDAT	PWM_BA+0x05C	R	Capture Falling Latch Register	0x0000_0000
PWM_PCEN	PWM_BA+0x07C	R/W	PWM Output and Capture Input Enable Register	0x0000_0000

5.6.12 Register Description

PWM Pre-Scale Register (PWM_CLKPSC)

Register	Offset	R/W	Description	Reset Value
PWM_CLKPSC	PWM_BA+0x000	R/W	PWM Prescaler Register	0x0000_0000

31	30	29	28	27	26	25	24
DZI1							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DZI0							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CLKPSC							

Bits	Description	
[31:24]	**DZI1**	**Dead Zone Interval Register 1** These 8 bits determine dead zone length. The unit time of dead zone length is that from clock selector.
[23:16]	**DZI0**	**Dead Zone Interval Register 0** These 8 bits determine dead zone length. The unit time of dead zone length is that from clock selector.
[15:8]	**Reserved**	Reserved.
[7:0]	**CLKPSC**	**Clock Prescaler For PWM Timer** Clock input is divided by (CLKPSC + 1) If CLKPSC = 0, then the prescaler output clock will be stopped. This implies PWM counter will also be stopped.

PWM Clock Select Register (PWM_CLKDIV)

Register	Offset	R/W	Description	Reset Value
PWM_CLKDIV	PWM_BA+0x004	R/W	PWM Clock Select Register	0x0000_0000

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**CLKDIV**	**PWM Timer Clock Source Selection** Value : Input clock divided by 000 : 2 001 : 4 010 : 8 011 : 16 1xx : 1

PWM Control Register (PWM_CTL)

Register	Offset	R/W	Description	Reset Value
PWM_CTL	PWM_BA+0x008	R/W	PWM Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**DTEN1**	**DTEN0**	**CNTMODE**	**PINV**	**Reserved**	**CNTEN**

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**DTEN1**	**Dead-Zone 1 Generator Enable/Disable** 0 = Disable. 1 = Enable.
[4]	**DTEN0**	**Dead-Zone 0 Generator Enable/Disable** 0 = Disable. 1 = Enable.
[3]	**CNTMODE**	**PWM-Timer Auto-Reload/One-Shot Mode** 0 = One-Shot Mode. 1 = Auto-reload Mode.
[2]	**PINV**	**PWM-Timer Output Inverter ON/OFF** 0 = Inverter OFF. 1 = Inverter ON.
[0]	**CNTEN**	**PWM-Timer Enable** 0 = Stop PWM-Timer Running. 1 = Enable PWM-Timer.

PWM Period Register (PWM_PERIOD)

Register	Offset	R/W	Description	Reset Value
PWM_PERIOD	PWM_BA+0x00C	R/W	PWM Period Register	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**PERIOD**	**PWM Counter/Timer Reload Value** PERIOD determines the PWM period. **Note:** One PWM cycle width = (PERIOD + 1).

PWM Comparator Register 3-0 (PWM_CMPDAT3-0)

Register	Offset	R/W	Description	Reset Value
PWM_CMPDAT0	PWM_BA+0x010	R/W	PWM Comparator Register 0	0x0000_0000
PWM_CMPDAT1	PWM_BA+0x01C	R/W	PWM Comparator Register 1	0x0000_0000
PWM_CMPDAT2	PWM_BA+0x028	R/W	PWM Comparator Register 2	0x0000_0000
PWM_CMPDAT3	PWM_BA+0x034	R/W	PWM Comparator Register 3	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CMP**	**PWM Comparator Register** CMP determines the PWM duty ratio. Assumption: PWM output initial is high • CMP > = PERIOD: PWM output is always high. • CMP < PERIOD: PWM low width = (PERIOD - CMP) unit; PWM high width = (CMP+1) unit. • CMP = 0: PWM low width = (PERIOD) unit; PWM high width = 1 unit. **Note1:** Unit = one PWM clock cycle. **Note2:** Any write to CMP will take effect in next PWM cycle.

PWM Counter Register (PWM_CNT)

Register	Offset	R/W	Description	Reset Value
PWM_CNT	PWM_BA+0x014	R	PWM Counter Register	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CNT**	**PWM Counter Register** Reports the current value of the 16-bit down counter.

PWM Interrupt Enable Register (PWM_INTEN)

Register	Offset	R/W	Description	Reset Value
PWM_INTEN	PWM_BA+0x040	R/W	PWM Interrupt Enable Register	0x0000_0000

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

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**PIEN**	**PWM Timer Interrupt Enable** 0 = Disable. 1 = Enable.

PWM Interrupt Flag Register (PWM_INTSTS)

Register	Offset	R/W	Description	Reset Value
PWM_INTSTS	PWM_BA+0x044	R/W	PWM Interrupt Flag Register	0x0000_0000

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

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**PIF**	**PWM Timer Interrupt Flag** Flag is set by hardware when PWM down counter reaches zero, software can clear this bit by writing ‘1’ to it.

Capture Control Register (PWM_CAPCTL)

Register	Offset	R/W	Description	Reset Value
PWM_CAPCTL	PWM_BA+0x050	R/W	Capture Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CFLIF	**CRLIF**	**Reserved**	**CAPIF**	**CAPEN**	**CFLIEN**	**CRLIEN**	**CAPINV**

Bits	Description	
[31:8]	**Reserved**	Reserved.
[7]	**CFLIF**	**PWM_FCAPDAT Latched Indicator Bit** When input channel has a **falling** transition, PWM_FCAPDAT was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing a zero to it.
[6]	**CRLIF**	**PWM_RCAPDAT Latched Indicator Bit** When input channel has a **rising** transition, PWM_RCAPDAT was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing a zero to it.
[5]	**Reserved**	Reserved.
[4]	**CAPIF**	**Capture Interrupt Indication Flag** When capture input has a falling/rising transition and falling/rising latch interrupt is enabled (CFLIEN = 1/CRLIEN = 1), CAPIF0 is set 1 by hardware. Software can clear this bit by writing a one to it. **Note:** If this bit is “1”, PWM counter will not be reloaded when next capture interrupt occurs.
[3]	**CAPEN**	**Capture Channel Input Transition Enable/Disable** 0 = Disable capture function. 1 = Enable capture function. When enabled, Capture function latches the PMW-counter to RCAPDAT (Rising latch) and FCAPDAT (Falling latch) registers on input edge transition. When disabled, Capture function is inactive as is interrupt.
[2]	**CFLIEN**	**Falling Latch Interrupt Enable ON/OFF** 0 = Disable falling latch interrupt. 1 = Enable falling latch interrupt. When enabled, capture block generates an interrupt on falling edge of input.

[1]	**CRLIEN**	**Rising Latch Interrupt Enable ON/OFF** 0 = Disable rising latch interrupt. 1 = Enable rising latch interrupt. When enabled, capture block generates an interrupt on rising edge of input.
[0]	**CAPINV**	**Inverter ON/OFF** 0 = Inverter OFF. 1 = Inverter ON. Reverse the input signal from GPIO before fed to Capture timer

Capture Rising Latch Register (PWM_RCAPDAT)

Register	Offset	R/W	Description	Reset Value
PWM_RCAPDAT	PWM_BA+0x058	R	Capture Rising Latch Register	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**RCAPDAT**	**Capture Rising Latch Register** In Capture mode, this register is latched with the value of the PWM counter on a rising edge of the input signal.

Capture Falling Latch Register (PWM_FCAPDAT)

Register	Offset	R/W	Description	Reset Value
PWM_FCAPDAT	PWM_BA+0x05C	R	Capture Falling Latch Register	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**FCAPDAT**	**Capture Falling Latch Register** In Capture mode, this register is latched with the value of the PWM counter on a falling edge of the input signal.

PWM Output and Capture Input Enable Register (PWM_PCEN)

Register	Offset	R/W	Description	Reset Value
PWM_PCEN	PWM_BA+0x07C	R/W	PWM Output and Capture Input Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**CAPINEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**POEN3**	**POEN2**	**POEN1**	**POEN0**

Bits	Description	
[31:9]	**Reserved**	Reserved.
[8]	**CAPINEN**	**Capture Input Enable Register** 0 = OFF (PB.12 pin input disconnected from Capture block). 1 = ON (PB.12 pin, if in PWM alternative function, will be configured as an input and fed to capture function).
[7:4]	**Reserved**	Reserved.
[3]	**POEN3**	**PWM3 Output Enable Register** 0 = Disable PWM3 output to pin. 1 = Enable PWM3 output to pin. **Note:** The corresponding GPIO pin also must be switched to PWM function (refer to SYS_GPB_MFP Table 5-8)
[2]	**POEN2**	**PWM2 Output Enable Register** 0 = Disable PWM2 output to pin. 1 = Enable PWM2 output to pin. **Note:** The corresponding GPIO pin also must be switched to PWM function (refer to SYS_GPA_MFP Table 5-8)
[1]	**POEN1**	**PWM1 Output Enable Register** 0 = Disable PWM1 output to pin. 1 = Enable PWM1 output to pin. **Note:** The corresponding GPIO pin also must be switched to PWM function (refer to SYS_GPA_MFP Table 5-8)
[0]	**POEN0**	**PWM0 Output Enable Register** 0 = Disable PWM0 output to pin. 1 = Enable PWM0 output to pin. **Note:** The corresponding GPIO pin also must be switched to PWM function (refer to SYS_GPA_MFP Table 5-8)

5.7 Real Time Clock (RTC)

5.7.1 Overview

The 32 KHz crystal oscillator is clock source of RTC. The RTC time-out frequency is programmable. During normal operation, the RTC time-out can generate an interrupt at a pre-programmed frequency. During power down mode, the RTC time-out could be one of wake-up source.

5.7.2 Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
RTC Base Address: **RTC_BA = 0x4000_8000**				
RTC_CTL	RTC_BA+0x000	R/W	RTC Control Register	0x0000_0000

5.7.3 Register Description

RTC Control Register (RTC_CTL)

Register	Offset	R/W	Description	Reset Value
RTC_CTL	RTC_BA+0x000	R/W	RTC Control Register	0x0000_0000

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

Bits	Description	
[31:5]	**Reserved**	Reserved.
[4:3]	**RTIS**	**RTC Timer Interval Select** These two bits select the timeout interval for the RTC. 00 = Time-out frequency is 0.25Hz,. 01 = Time-out frequency is 2Hz,. 10 = Time-out frequency is 8Hz,. 11 = Time-out frequency is 32Hz.

[2]	**RTCE**	**RTC Enable** 0 = Disable RTC function. 1 = Enable RTC function.
[1]	**RTIE**	**RTC Interrupt Enable** 0 = Disable the RTC interrupt. 1 = Enable the RTC interrupt.
[0]	**RTIF**	**RTC Interrupt Flag** If the RTC interrupt is enabled, then the hardware will set this bit to indicate that the RTC interrupt has occurred. If the RTC interrupt is not enabled, then this bit indicates that a timeout period has elapsed. 0 = RTC interrupt does not occur. 1 = RTC interrupt occurs. **Note:** This bit is cleared by writing 1 to this bit.

5.8 Serial Peripheral Interface (SPI) Controller

5.8.1 Overview

The SPI synchronous serial interface controller performs a serial-to-parallel conversion on data received from the peripheral, and a parallel-to-serial conversion on data received by CPU. This interface can drive up to 2 external peripherals in time-shared operation. It can generate an interrupt signal when data transfer is finished and can be cleared by writing 1 to the interrupt flag. The active level of device/slave select signal can be chosen to low active or high active on SS_LVL bit, which depends on the peripheral it's connected. Writing a divisor into DIVIDER register can program the frequency of serial clock output.

This SPI core contains two 32-bit transmit/receive buffers, and can provide burst mode operation. It supports variable length transfer and the maximum transmitted/received length can be up to 32 bits.

Two sets SPI are designed in N572F072/P072 and they are named as SPI0 and SPI1. There is little different between those SPI. SPI0 only supports master mode, SPI1 supports both master and slave modes. In SPI1 slave mode, SPI_SSB10 is the slave select input pin.

5.8.2 Features

- AMBA APB interface compatible
- Variable length of transfer word up to 32 bits
- Provide burst mode operation, transmit/receive can be executed up to two times in one transfer
- MSB or LSB first data transfer
- Up to 2 slave/device select pins in master mode, but 1 device/slave select pin in slave mode
- Fully static synchronous design with one clock domain

5.8.3 SPI Block Diagram

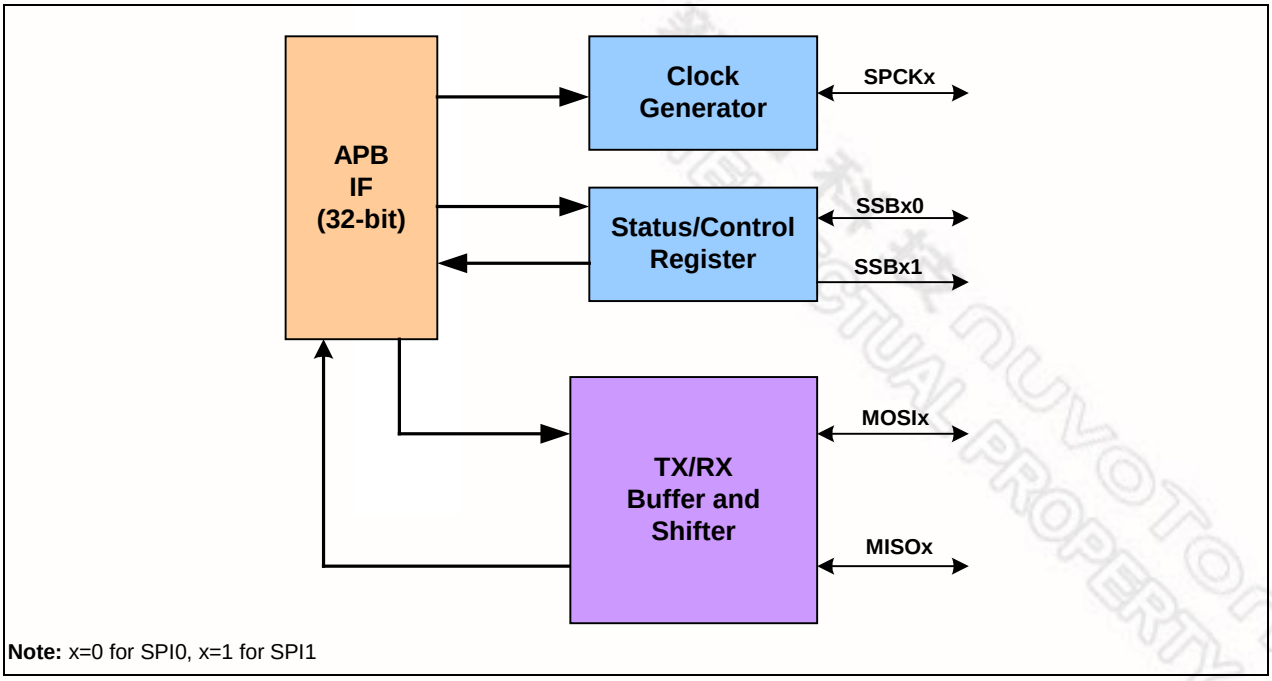


Figure 5-13 SPI Block Diagram

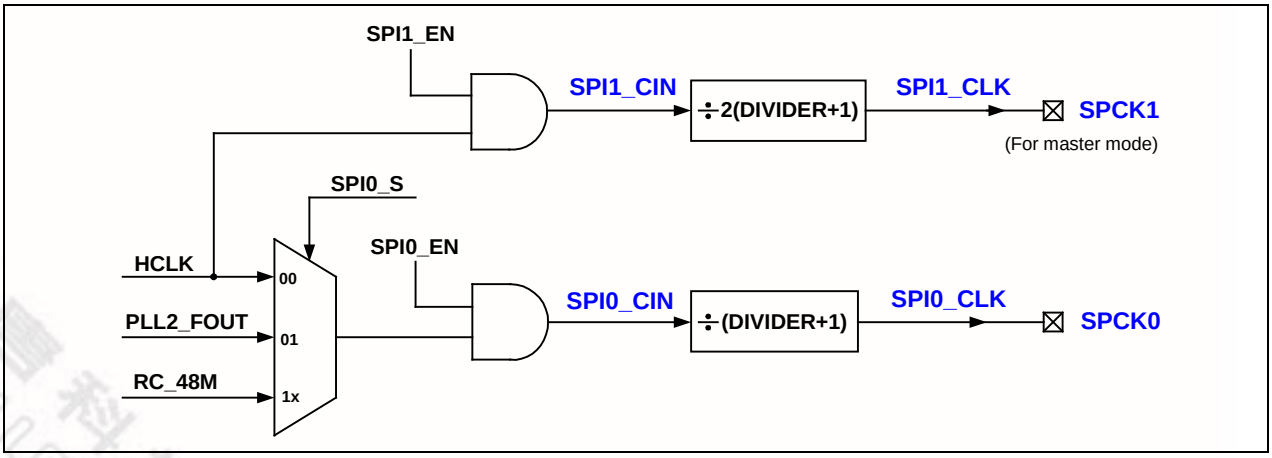


Figure 5-14 SPI Clock Source

5.8.4 SPI Applications

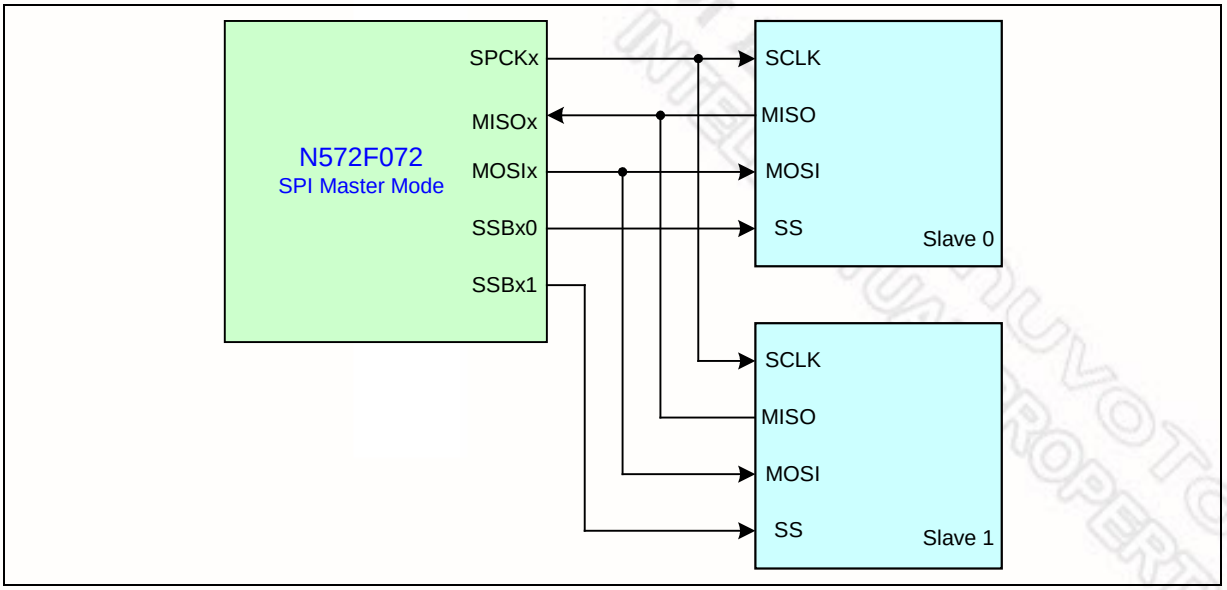


Figure 5-15 SPI Master Mode Application Diagram

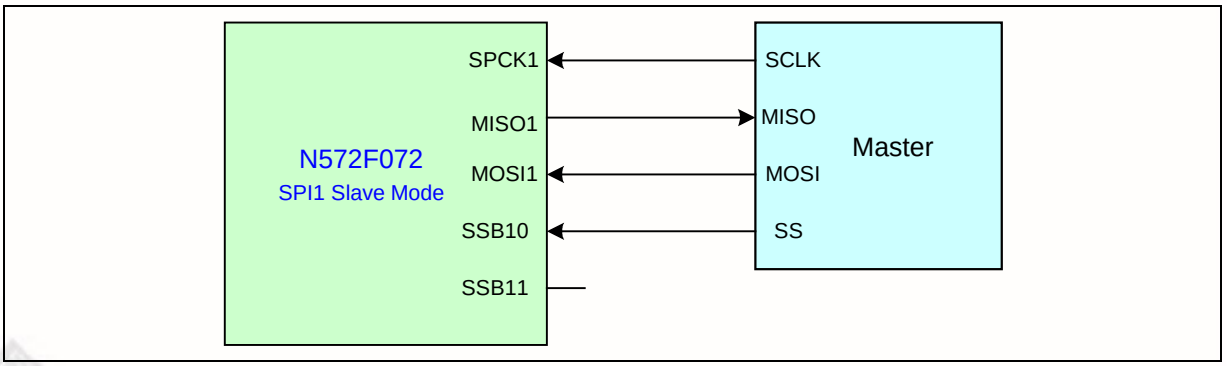


Figure 5-16 SPI Slave Mode Application Diagram

5.8.5 SPI Configuration Examples

When using this SPI controller as a master to access a slave device with following specifications:

- Data bit latches on positive edge of serial clock
- Data bit drives on negative edge of serial clock
- Data is transferred with the MSB first
- SPCK idle low.
- Only one byte transmit/receive in a transfer
- Device/Slave select signal is active low

Basically, the following actions should be done (also, the specification of the connected slave device should be referred to when consider the following steps in detail):

1) Write a divisor into DIVIDER to determine the frequency of serial clock.
2) Setup SPI_SSR. Set AUTOSS=0, SS_LVL=0 and SS[0] or SS[1] to 1 to activate the device to be accessed.

When transmit (write) data to device:

3) Write the data to be transmitted into SPI_TX0[7:0].

When receive (read) data from device:

4) Write 0xFFFFFFFF into SPI_TX0.
5) Setup SPI_CTL. Set SLAVE=0, CLKP=0, RX_NEG=0, TX_NEG=1, DWIDTH=0x08, TX_NUM=0x0, LSB=0, SUSPITV=0x0, and GO_BUSY=1 to start the transfer.

--- Wait for interrupt (if UNIT_INTEN=1) or polling the GO_BUSY bit until it turns to 0 ---

6) Read out the received data from SPI_RX0.
7) Go to 3) to continue another data transfer, or set SS[0] or SS[1] to 0 to inactivate the device.

5.8.6 SPI Control Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
SPI Base Address: **SPIn_BA = 0x4003_0000 + (0x4000 * n)** **n= 0,1**				
SPI_CTL	SPIn_BA+0x00	R/W	Control and Status Register	0x0000_0004
SPI_CLKDIV	SPIn_BA+0x04	R/W	Clock Divider Register (Master Only)	0x0000_0000
SPI_SSCTL	SPIn_BA+0x08	R/W	Slave Select Register	0x0000_0000
SPI_RX0	SPIn_BA+0x10	R	Data Receive Register 0	0x0000_0000
SPI_RX1	SPIn_BA+0x14	R	Data Receive Register 1	0x0000_0000
SPI_TX0	SPIn_BA+0x20	W	Data Transmit Register 0	0x0000_0000
SPI_TX1	SPIn_BA+0x24	W	Data Transmit Register 1	0x0000_0000
SPI0_RCLK	SPI0_BA+0x30	R/W	SPI0 Receive Timing Control Register	0x0000_000F

5.8.7 SPI Control Register Description

SPI Control and Status Register (SPI_CTL)

Register	Offset	R/W	Description	Reset Value
SPI_CTL	SPIn_BA+0x00	R/W	Control and Status Register	0x0000_0004

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved			REORDER	Reserved	SLAVE	UNIT_INTEN	UNIT_INTSTS
15	14	13	12	11	10	9	8
SUSPITV				CLKP	LSB	TX_NUM	
7	6	5	4	3	2	1	0
DWIDTH					TX_NEG	RX_NEG	GO_BUSY

Bits	Description	
[31:21]	**Reserved**	Reserved.
[20]	**REORDER**	**BYTE ENDIAN** 0 = Disable the BYTE ENDIAN. 1 = Enable the BYTE ENDIAN. Only the 16, 24, and 32 bits which are defined in DWIDTH are supported. When the transition is set as MSB first and REORDER=1, the data stored in the SPI_TXn buffer will be arranged in the order as [BYTE0, BYTE1, BYTE2, BYTE3] in DWIDTH=32-bit mode, and the sequence of transmitted data will be BYTE0, BYTE1, BYTE2, and BYTE3. If the DWIDTH is set as 24-bit mode, the data in SPI_TXn buffer will be arranged as [unknown_byte, BYTE0, BYTE1, BYTE2] and the BYTE0, BYTE1, and BYTE2 will be transmitted step by step in MSB first. The rule of 16-bit mode is the same as above.
[19]	**Reserved**	Reserved.
[18]	**SLAVE**	**Master/Slave Mode Select** 0 = Master mode. 1 = Slave mode. This bit exists in SPI1 only. SPI0 only supports master mode and it does not have this bit.
[17]	**UNIT_INTEN**	**Unit Transfer Interrupt Enable** 0 = Disable SPI Unit Transfer Interrupt. 1 = Enable SPI Unit Transfer Interrupt to CPU.
[16]	**UNIT_INTSTS**	**Unit Transfer Interrupt Status** 0 = No transaction has been finished since this bit was cleared to 0. 1 = SPI controller has finished one unit transfer. **Note:** This bit is read only, but can be cleared by writing 1 to this bit.

[15:12]	**SUSPITV**	**Suspend Interval (Master Mode Only)** The four bits provide configurable suspend interval between two successive transmit/receive transaction in a transfer. The definition of the suspend interval is the interval between the last falling edge of SPI_SCLKn of the preceding transaction word and the first rising edge of SPI_SCLKn of the following transaction word. The default value is 0x0. When TX_NUM=00, setting this field has no effect on transfer. The period of the suspend interval is obtained according to the following equation. (SUSPITV+2)* SPI_SCLKn clock cycles **Note:** SUSPITV cannot be "0" for SPI0.
[11]	**CLKP**	**Clock Polarity** 0 = SPI_SCLKn idle low. 1 = SPI_SCLKn idle high.
[10]	**LSB**	**Send LSB First** 0 = The MSB is transmitted/received first (which bit in SPI_TXn/SPI_RXn is MSB is dependent on DWIDTH). 1 = The LSB (SPI_TXn[0]) is sent first to SPI_MOSIn, and the first bit received from SPI_MISOn will be put in the LSB (SPI_RXn[0]).
[9:8]	**TX_NUM**	**Transmit/Receive Numbers** This field specifies how many transmit/receive numbers should be executed in one transfer. 00 = Only one transmit/receive will be executed in one transfer. 01 = Two successive transmit/receive will be executed in one transfer. 1x = Reserved.
[7:3]	**DWIDTH**	**Transmit Bit Length** This field specifies how many bits are transmitted in one transmit/receive. Up to 32 bits can be transmitted. DWIDTH = 0x01 … 1 bit DWIDTH = 0x02 … 2 bits …… DWIDTH = 0x1F … 31 bits DWIDTH = 0x00 … 32 bits.
[2]	**TX_NEG**	**Transmit On Negative Edge** 0 = The output on SPI_MOSIn is changed on the rising edge of SPI_SCLKn. 1 = The output on SPI_MOSIn is changed on the falling edge of SPI_SCLKn.
[1]	**RX_NEG**	**Receive On Negative Edge** 0 = The input on SPI_MISOn is latched on the rising edge of SPI_SCLKn. 1 = The input on SPI_MISOn is latched on the falling edge of SPI_SCLKn.
[0]	**GO_BUSY**	**Go And Busy Status** 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit starts the transfer. This bit remains "1" during the transfer and is automatically cleared after the transfer is finished. **NOTE:** All registers should be set readily before writing 1 to the GO_BUSY bit. When a transfer is in progress, writing to any register of the SPI core has no effect.

SPI Divider Register (SPI_CLKDIV)

Register	Offset	R/W	Description	Reset Value
SPI_CLKDIV	SPIn_BA+0x04	R/W	Clock Divider Register (Master Only)	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**DIVIDER**	**Clock Divider Register (Master Mode Only)** **SPI0:** SPI0 clock pin SPI_SCLK0 output clock frequency is SPI0_CIN/(DIVIDER+1). DIVIDER can be from 0 to 65535. **SPI1:** SPI1 clock pin SPI_SCLK1 output clock frequency is SPI1_CIN/2*(DIVIDER+1). Suggest DIVIDER should be at least 1 in master mode. In slave mode, the period of SPI_SCLK1 clock input shall be equal or over 5 times HCLK at least.

SPI Slave Select Register (SPI_SSCTL)

Register	Offset	R/W	Description	Reset Value
SPI_SSCTL	SPIn_BA+0x08	R/W	Slave Select Register	0x0000_0000

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

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5]	**LTRIG_FLAG**	**Level Trigger Flag (Slave Mode Only)** When the **SS_LTRIG** bit is set in slave mode, this bit can be read to indicate the received bit number meets the requirement or not. 0 = One of the received number and the received bit length doesn't meet the requirement in one transfer. 1 = The received number and received bits meet the requirement which is defined in **TX_NUM** and **DWIDTH** among one transfer. **Note 1**: This bit is READ only. **Note 2**: This bit exists in SPI1 only. SPI0 only supports master mode and it does not have this bit.
[4]	**SS_LTRIG**	**Slave Select Level Trigger (Slave Mode Only)** 0 = The input slave select signal is edge-trigger. This is the default value. 1 = The slave select signal will be level-trigger. It depends on **SS_LVL** to decide the signal is active low or active high. **Note 1**: This bit exists in SPI1 only. SPI0 only supports master mode and it does not have this bit.
[3]	**AUTOSS**	**Automatic Slave Select (Master Mode Only)** 0 = Slave select signal (SPI_SSBx0/SPI_SSBx1) is asserted and de-asserted by setting and clearing related bit SS. 1 = Slave select signal (SPI_SSBx0/SPI_SSBx1) is generated automatically. It means that slave select signal, which is set in bits SS, is asserted by the SPI controller when transmit/receive is started by setting GO_BUSY, and is de-asserted after the transfer is finished.

[2]	**SS_LVL**	**Slave Select Active Level** It defines the active level of device/slave select signal. 0 = The SPI_SSBx0/SPI_SSBx1 slave select signal is active Low. 1 = The SPI_SSBx0/SPI_SSBx1 slave select signal is active High.
[1:0]	**SS**	**Slave Select Pin Control** If AUTOSS bit is 0, SPIn_SSBx0 and SPIn_SSBx1 output are determined by SS[0] and SS[1] respectively. 0 = Any bit location of this field forces the pin to inactive state. 1 = Any bit location of this field forces the proper SPIn_SSBx0/SPIn_SSBx1 pin to an active state If AUTOSS bit is 1, 0 = Any bit location of this field will select appropriate SPIn_SSBx0/SPIn_SSBx1 pin to be driven to inactive state. 1 = Any bit location of this field will select appropriate SPIn_SSBx0/SPIn_SSBx1 pin to be automatically driven to active state for the duration of the transmit/receive, and to be driven to inactive state for the rest of the time. The active state of SPIn_SSBx0/SPIn_SSBx1 is specified in SS_LVL bit (SPI_SSCTL[2]). **Note 1:** This interface can only drive one device/slave at a given time. Therefore, the slaves select of the selected device must be set to its active level before starting any read or write transfer. **Note 2:** SPIn_SSB10 is also defined as device/slave select input signal in **slave mode**. And that the slave select input signal must be driven by edge active trigger which level depend on the SS_LVL setting, otherwise the SPI slave core will go into dead path until the edge active trigger again or reset the SPI core by software.

SPI Data Receive Register (SPI_RXn)

Register	Offset	R/W	Description	Reset Value
SPI_RX0	SPIn_BA+0x10	R	Data Receive Register 0	0x0000_0000
SPI_RX1	SPIn_BA+0x14	R	Data Receive Register 1	0x0000_0000

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

Bits	Description	
[31:0]	**RX**	**Data Receive Register** The Data Receive Registers hold the value of received data of the last executed transfer. Valid bits depend on the transmit bit length field in the SPI_CTL register. For example, if DWIDTH=0x08 and TX_NUM=0x0, bits SPI_RX0[7:0] hold the received data. **Note:** The Data Receive Registers are **read only** registers. A Write to these registers will actually modify the Data Transmit Registers because those registers share the same flip-flops.

SPI Data Transmit Register (SPI_TXn)

Register	Offset	R/W	Description	Reset Value
SPI_TX0	SPIn_BA+0x20	W	Data Transmit Register 0	0x0000_0000
SPI_TX1	SPIn_BA+0x24	W	Data Transmit Register 1	0x0000_0000

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

Bits	Description	
[31:0]	**TX**	**Data Transmit Register** The Data Transmit Registers hold the data to be transmitted in the next transfer. Valid bit depends on the transmit bit length field in the SPI_CTL register. For example, if DWIDTH=0x08 and the TX_NUM=0x0, the bits SPI_TX0[7:0] will be transmitted in next transfer. If DWIDTH=0x00 and TX_NUM=0x1, H/W will perform two 32-bit transmit/receive successively with the data order {SPI_TX0[31:0], SPI_TX1[31:0]}. **Note:** The SPI_RXn and SPI_TXn registers share the same flip-flops, which mean that what is received in one transfer will be transmitted in the next transfer if there is not any write to the SPI_TXn register between the two transfers.

SPI0 Receive Timing Control Register (SPI0_RCLK)

These bits are protected bits. To program these bits needs an open lock sequence, write “59h”, “16h”, “88h” to register SYS_REGLCTL to un-lock these bits. Refer to the register SYS_REGLCTL at address SYS_BA + 0x100.

Note: Only SPI0 provides this register.

Register	Offset	R/W	Description	Reset Value
SPI0_RCLK	SPI0_BA+0x30	R/W	SPI0 Receive Timing Control Register	0x0000_000F

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**SPI0_CTIM**		**SPI0_FTIM**	

Bits	Description	
[31:4]	**Reserved**	Reserved.
[3:2]	**SPI0_CTIM**	**Coarse Timing Control For SPI0 Data Receiving** Setting these bits can adjust receiving clock for latching serial-in data correctly in high speed transmission mode. 00 = Receiving data clock of SPI0 is same as the SPI0_CLK. 01 = Receiving data clock of SPI0 is delayed 2 half SPI0_CLK clock cycle,. 10 = Receiving data clock of SPI0 is delayed 3 half SPI0_CLK clock cycle,. 11 = Receiving data clock of SPI0 is delayed 1 half SPI0_CLK clock cycle ,. The default value are set according to the flash controller User Configuration Register CONFIG, SPI0_CTIM=CONFIG[7:6].
[1:0]	**SPI0_FTIM**	**Fine Timing Control For SPI0 Data Receiving** The delay timing selected by SPI0_CTIM can be further tuned finely by SPI0_FTIM. 00 = Receiving data clock of SPI0 has extra 7.5nS delay. 01 = Receiving data clock of SPI0 has extra 5.0nS delay,. 10 = Receiving data clock of SPI0 has extra 2.5nS delay,. 11 = Receiving data clock of SPI0 has no extra delay. **Note:** The extra delay is implemented by delay chains. The accuracy of delay time would base on process deviation. The default value are set according to the flash controller User Configuration Register CONFIG, SPI0_FTIM=CONFIG[5:4].

5.9 Timer Controller

5.9.1 General Timer Controller

The timer module includes three channels, Timer0, Timer1 and Timer2 which allow user to easily implement a counting scheme for use. The timer can perform functions like frequency measurement, event counting, interval measurement, clock generation, delay timing, and so on. The timer possesses features such as adjustable resolution, programmable counting period, and detailed information. The timer can generate an interrupt signal upon timeout, or provide the current value of count during operation.

Besides the general timer function, Timer0 and Timer1 provide another function. Timer0 time out rate is used as APU sample output rate when APU is enabled. Timer1 can be used to generate IR carrier output.

A simple fixed frequency timer TimerF is provided. Its frequency can be one of the followings.

32KHz crystal / 32,
32KHz crystal / (4x32),
48MHz RC oscillator / 65536,
48MHz RC oscillator / (4x65536).

5.9.2 Features

The general timer (Timer0/1/2) controller includes the following features

- AMBA APB interface compatible
- Each channel with an 8-bit pre-scale counter, a 16-bit up-counter and an interrupt request signal.
- Independent clock source for each channel.
- Time out period = (Period of timer clock input) * (Prescale + 1) * (TCMP)
- Maximum counting cycle time = (1/24M) * (2^8) * (2^16), if TMRx_CLK=24MHz.
- Internal 16-bit up-counter is readable on the fly.

5.9.3 Timer Controller Block Diagram

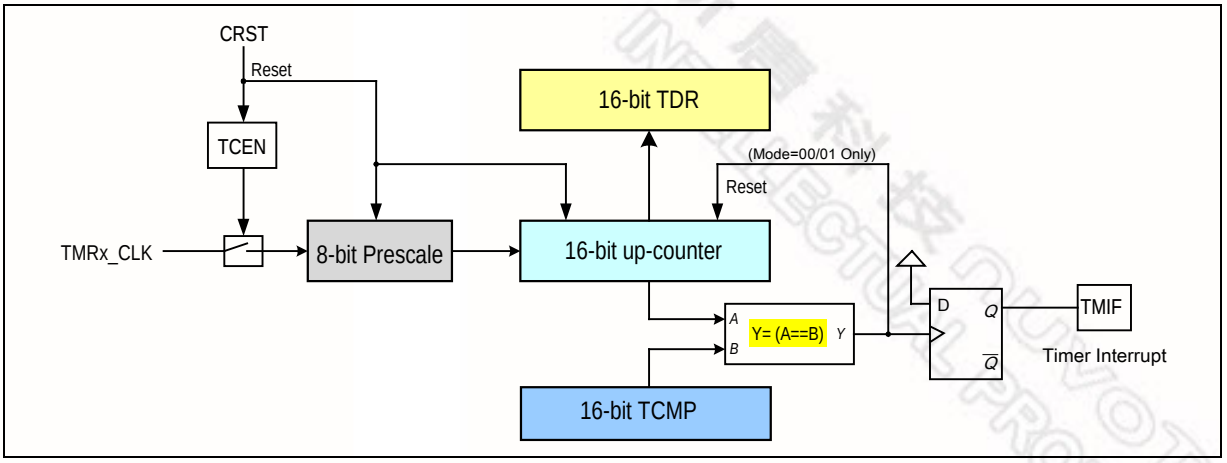


Figure 5-17 Timer0/1/2 Block Diagram

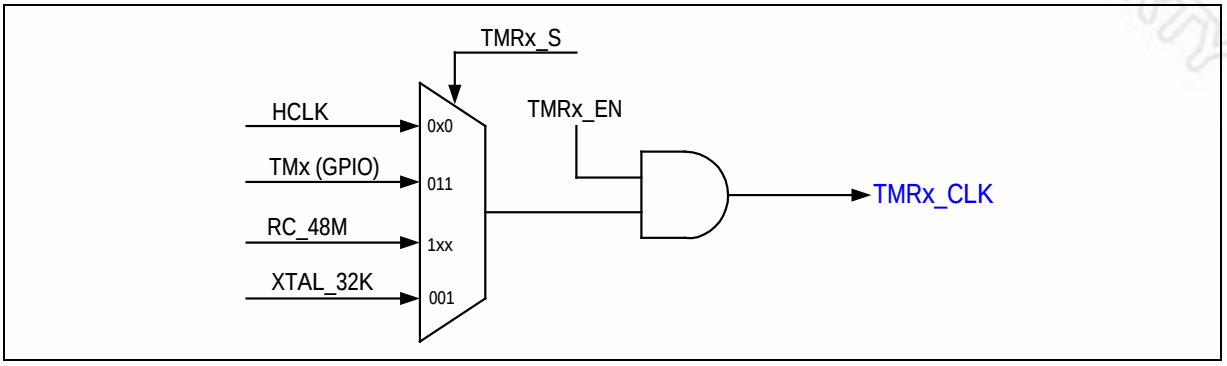


Figure 5-18 Clock Source of Timer0/1/2

5.9.4 Timer Controller Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
TMR Base Address: **TMR_BA=0x4001_0000**				
TIMER0_CTL	TMR_BA+0x00	R/W	Timer0 Control and Status Register	0x0000_0005
TIMER0_CMP	TMR_BA+0x04	R/W	Timer0 Compare Register	0x0000_0000
TIMER0_INTSTS	TMR_BA+0x08	R/W	Timer0 Interrupt Status Register	0x0000_0000
TIMER0_CNT	TMR_BA+0x0C	R	Timer0 Data Register	0x0000_0000
TIMER1_CTL	TMR_BA+0x20	R/W	Timer1 Control and Status Register	0x0000_0005
TIMER1_CMP	TMR_BA+0x24	R/W	Timer1 Compare Register	0x0000_0000
TIMER1_INTSTS	TMR_BA+0x28	R/W	Timer1 Interrupt Status Register	0x0000_0000
TIMER1_CNT	TMR_BA+0x2C	R	Timer1 Data Register	0x0000_0000
TIMERF_INTSTS	TMR_BA+0x30	R/W	TimerF Interrupt Status Register	0x0000_0000
IR_CTL	TMR_BA+0x34	R/W	IR Carrier Output Control Register	0x0000_0000
TIMER2_CTL	TMR_BA+0x40	R/W	Timer2 Control and Status Register	0x0000_0005
TIMER2_CMP	TMR_BA+0x44	R/W	Timer2 Compare Register	0x0000_0000
TIMER2_INTSTS	TMR_BA+0x48	R/W	Timer2 Interrupt Status Register	0x0000_0000
TIMER2_CNT	TMR_BA+0x4C	R	Timer2 Data Register	0x0000_0000

5.9.5 Timer Controller Register Description

Timer Control Register (TIMERx_CTL)

Register	Offset	R/W	Description	Reset Value
TIMER0_CTL	TMR_BA+0x00	R/W	Timer0 Control and Status Register	0x0000_0005
TIMER1_CTL	TMR_BA+0x20	R/W	Timer1 Control and Status Register	0x0000_0005
TIMER2_CTL	TMR_BA+0x40	R/W	Timer2 Control and Status Register	0x0000_0005

31	30	29	28	27	26	25	24
Reserved	CNTEN	INTEN	OPMODE[1:0]		RSTCNT	ACTSTS	Reserved
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
PSC[7:0]							

Bits	Description	
[31]	**Reserved.**	Reserved.
[30]	**CNTEN**	**Counter Enable Bit** 0 = Stop/Suspend counting. 1 = Start counting. **Note:** This bit is auto-cleared by hardware in one-shot mode (OPMODE = 00b) when the associated timer interrupt is generated (INTEN = 1).
[29]	**INTEN**	**Interrupt Enable Bit** 0 = Disable TIMER Interrupt. 1 = Enable TIMER Interrupt. If timer interrupt is enabled, the timer asserts its interrupt signal when the associated count is equal to TIMERx_CMP.
[28:27]	**OPMODE**	**Timer Operating Mode** 00 = The Timer is operating in the one-shot mode. The associated interrupt signal is generated once (if INTEN is 1) and CNTEN is automatically cleared by hardware. 01 = The Timer is operating in the periodic mode. The associated interrupt signal is generated periodically (if INTEN is 1). 10 = Reserved. 11 = The Timer is operating in continuous counting mode. The associated interrupt signal is generated when TIMERx_CNT = TIMERx_CMP (if INTEN is 1); however, the 16-bit up-counter counts continuously without reset.

[26]	**RSTCNT**	**Counter Reset Bit** Set this bit will reset the Timer counter, pre-scale and also force CNTEN to 0. 0 = No effect. 1 = Reset Timer's pre-scale counter, internal 16-bit up-counter and CNTEN bit.
[25]	**ACTSTS**	**Timer Active Status Bit (Read Only)** This bit indicates the counter status of Timer. 0 = Timer is not active. 1 = Timer is active.
[24:8]	**Reserved**	Reserved.
[7:0]	**PSC**	**Timer Clock Prescaler** Clock input is divided by (PSC+1) before it is fed to the counter. If PSC = 0, then there is no scaling.

Timer Compare Register (TIMERx_CMP)

Register	Offset	R/W	Description	Reset Value
TIMER0_CMP	TMR_BA+0x04	R/W	Timer0 Compare Register	0x0000_0000
TIMER1_CMP	TMR_BA+0x24	R/W	Timer1 Compare Register	0x0000_0000
TIMER2_CMP	TMR_BA+0x44	R/W	Timer2 Compare Register	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CMPDAT**	**Timer Comparison Value** CMPDAT is a 16-bit comparison register. When the 16-bit up-counter is enabled and its value is equal to CMPDAT value, a Timer Interrupt is requested if the timer interrupt is enabled with TIMERx_CTL.INTEN = 1. **Note 1:** Never set CMPDAT to 0x000 or 0x001. Timer will not function correctly. **Note 2:** No matter CNTEN is 0 or 1, whenever software writes a new value into this register, TIMER will restart counting by using this new value and abort previous count.

Timer Interrupt Status Register (TIMERx_INTSTS)

Register	Offset	R/W	Description	Reset Value
TIMER0_INTSTS	TMR_BA+0x08	R/W	Timer0 Interrupt Status Register	0x0000_0000
TIMER1_INTSTS	TMR_BA+0x28	R/W	Timer1 Interrupt Status Register	0x0000_0000
TIMER2_INTSTS	TMR_BA+0x48	R/W	Timer2 Interrupt Status Register	0x0000_0000

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

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**TIF**	**Timer Interrupt Flag (Read Only)** This bit indicates the interrupt status of Timer. TIF bit is set by hardware when the 16-bit counter matches the timer comparison value (CMPDAT). It is cleared by writing 1 to itself 0 = No effect. 1 = CNT (TIMERx_CNT[15:0]) value matches the CMPDAT (TIMERx_CMP[15:0]) value.

Timer Data Register (TIMERx_CNT)

Register	Offset	R/W	Description	Reset Value
TIMER0_CNT	TMR_BA+0x0C	R	Timer0 Data Register	0x0000_0000
TIMER1_CNT	TMR_BA+0x2C	R	Timer1 Data Register	0x0000_0000
TIMER2_CNT	TMR_BA+0x4C	R	Timer2 Data Register	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**CNT**	**Timer Data Register** User can read this register for the current up-counter value while TIMERx_CTL.CNTEN is set to 1,

TimerF Interrupt Status Register (TIMERF_INTSTS)

Register	Offset	R/W	Description	Reset Value
TIMERF_INTSTS	TMR_BA+0x30	R/W	TimerF Interrupt Status Register	0x0000_0000

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

Bits	Description	
[31:2]	**Reserved**	Reserved.
[1]	**TFIE**	**TimerF Interrupt Enable** 0 = Disable TimerF Interrupt. 1 = Enable TimerF Interrupt.
[0]	**TFIF**	**TimerF Interrupt Flag** This bit indicates the interrupt status of TimerF. TFIF bit is set by hardware when TimerF time out. It is cleared by writing 1 to this bit. 0 = It indicates that TimerF does not time out yet. 1 = It indicates that TimerF time out. The interrupt flag is set if TimerF interrupt was enabled.

IR Carrier Output Control Register (IR_CTL)

Register	Offset	R/W	Description	Reset Value
IR_CTL	TMR_BA+0x34	R/W	IR Carrier Output Control Register	0x0000_0000

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

Timer1 time-out signal is used to toggle IROUT output. Before IR carrier output is enabled, user needs to setup Timer1 according to output frequency of IR carrier.

Bits	Descriptions	
[31:2]	**Reserved**	Reserved
[1]	**IRCEN**	**IR carrier output enable** 0 = Disable IR carrier output, 1 = Enable IR carrier output. Timer1 time out will toggle the output state on IROUT pin.
[0]	**NONCS**	**Non-carrier state** 0 = IROUT keeps low when IRCEN is 0, 1 = IROUT keeps high when IRCEN is 0.

5.10 Watchdog Timer

The purpose of Watchdog Timer (WDT) is to perform a system reset if system runs into an unknown state. This prevents system from hanging for an indefinite period of time. The watchdog timer includes a 19-bit free running counter with programmable time-out intervals.

Setting WDT_CTL[7](WDTEN) 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_CTL[3](IF) will be set immediately to request a WDT interrupt if the watchdog timer interrupt enable bit WDT_CTL[6](INTEN) is set, in the meantime, a specified delay time follows the time-out event. User must set WDT_CTL[0](RSTCNT) (Watchdog timer reset) high to reset the 19-bit WDT counter to prevent Watchdog timer reset before the delay time expires. WDT_CTL[0](RSTCNT) 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 WDT_CTL[10:8](TOUTSEL). If the WDT counter has not been cleared after the specific delay time expires, the watchdog timer will set Watchdog Timer Reset Flag WDT_CTL[2](RSTF) high and reset CPU. This reset will last 64 WDT clocks then CPU restarts executing program from reset vector (0x0000 0000). WDT_CTL[2](RSTF) will not be cleared by Watchdog reset. User may poll WDT_CTL[2](RSTF) by software to recognize the reset source.

Table 5-5 Watchdog Timeout Interval Selection

TOUTSEL	WDT Interrupt Timeout	WDT Reset Timeout	WDT Reset Timeout Interval (WDT_CLK=24 MHz)	WDT Reset Timeout Interval (WDT_CLK=32 KHz)
000	2^4 WDT_CLK	(2^4 + 1024) WDT_CLK	43.33 μs	32.5 ms
001	2^6 WDT_CLK	(2^6 + 1024) WDT_CLK	45.33 μs	34 ms
010	2^8 WDT_CLK	(2^8 + 1024) WDT_CLK	53.33 μs	40 ms
011	2^{10} WDT_CLK	(2^{10} + 1024) WDT_CLK	85.33 μs	64 ms
100	2^{12} WDT_CLK	(2^{12} + 1024) WDT_CLK	213.33 μs	160 ms
101	2^{14} WDT_CLK	(2^{14} + 1024) WDT_CLK	0.72 ms	544 ms
110	2^{16} WDT_CLK	(2^{16} + 1024) WDT_CLK	2.78 ms	2080 ms
111	2^{18} WDT_CLK	(2^{18} + 1024) WDT_CLK	10.97 ms	8224 ms

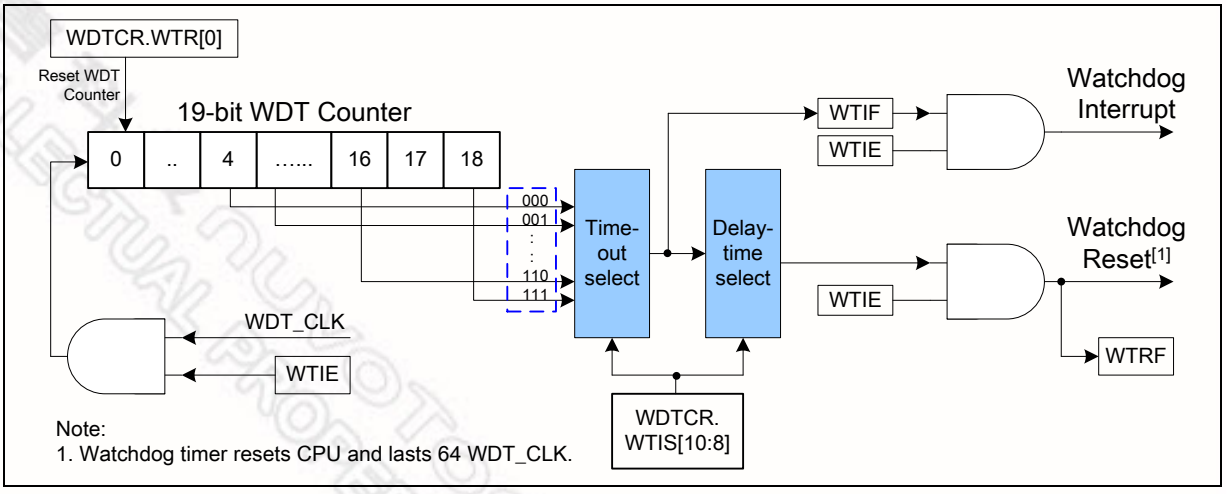


Figure 5-19 Watchdog Timer Block Diagram

5.10.1 Watchdog Timer Control Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
WDT Base Address: **WDT_BA = 0x4000_4000**				
WDT_CTL	WDT_BA+0x00	R/W	Watchdog Timer Control Register	0x0000_0700

5.10.2 Watchdog Timer Control Register Description

Watchdog Timer Control Register (WDT_CTL)

This is a protected register, to write to register, first issue the unlock sequence (see Protected Register Lock Key Register (SYS_REGLCTL)). Only flag bits IF and RSTF are unprotected and can be write-cleared at any time.

Register	Offset	R/W	Description	Reset Value
WDT_CTL	WDT_BA+0x00	R/W	Watchdog Timer Control Register	0x0000_0700

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**TOUTSEL**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WDTEN	**INTEN**	**WKF**	**WKEN**	**IF**	**RSTF**	**RSTEN**	**RSTCNT**

Bits	Description	
[31:11]	**Reserved**	Reserved.

[10:8]	**TOUTSEL**	**Watchdog Timer Interval Select** These three bits select the timeout interval for the Watchdog timer, a watchdog reset will occur 1024 clock cycles later if Watchdog timer is not reset. The WDT interrupt timeout is given by: 000 = 2^4 * WDT_CLK. 001 = 2^6 * WDT_CLK. 010 = 2^8 * WDT_CLK. 011 = 2^{10} * WDT_CLK. 100 = 2^{12} * WDT_CLK. 101 = 2^{14} * WDT_CLK. 110 = 2^{16} * WDT_CLK. 111 = 2^{18} * WDT_CLK. WDT reset timeout = (WDT interrupt timeout +1024) * WDT_CLK. Where WDT_CLK is the period of the Watchdog Timer clock source.
[7]	**WDTEN**	**Watchdog Timer Enable** 0 = Disable the WDT(Watchdog timer) (This action will reset the internal counter). 1 = Enable the WDT(Watchdog timer).
[6]	**INTEN**	**Watchdog Time-Out Interrupt Enable** 0 = Disable the WDT time-out interrupt. 1 = Enable the WDT time-out interrupt.
[5]	**WKF**	**WDT Time-Out Wake-Up Flag** If WDT causes CPU wake up from sleep or power-down mode, this bit will be set to high. 0 = WDT does not cause CPU wake-up. 1 = CPU wakes up from sleep or power-down mode by WDT time-out interrupt. **Note:** This bit is cleared by writing 1 to it.
[4]	**WKEN**	**WDT Time-Out Wake-Up Function Control** If this bit is set to 1, while WDT time-out interrupt flag IF (WDT_CTL[3]) is generated to 1 and interrupt enable bit INTEN (WDT_CTL[6]) is enabled, the WDT time-out interrupt signal will generate a wake-up trigger event to chip. 0 = Enable the Wakeup function that WDT timeout can wake up CPU from power-down mode. 1 = Disable WDT Wakeup CPU function.
[3]	**IF**	**Watchdog Timer Interrupt Flag** If the Watchdog timer interrupt is enabled, then the hardware will set this bit to indicate that the Watchdog timer interrupt has occurred. If the Watchdog timer interrupt is not enabled, then this bit indicates that a timeout period has elapsed. 0 = Watchdog timer interrupt has not occurred. 1 = Watchdog timer interrupt has occurred. **Note:** This bit is cleared by writing 1 to this bit.
[2]	**RSTF**	**Watchdog Timer Reset Flag** When the Watchdog timer initiates a reset, the hardware will set this bit. This flag can be read by software to determine the source of reset. Software is responsible to clear it manually by writing 1 to it. If RSTEN is disabled, then the Watchdog timer has no effect on this bit. 0 = Watchdog timer reset has not occurred. 1= Watchdog timer reset has occurred. **Note:** This bit is cleared by writing 1 to this bit.

[1]	**RSTEN**	**Watchdog Timer Reset Enable** Setting this bit will enable the Watchdog timer reset function. 0 = Disable Watchdog timer reset function. 1= Enable Watchdog timer reset function. **Note:** This function cannot work with XTL32-based clock source.
[0]	**RSTCNT**	**Clear Watchdog Timer** Set this bit will clear the Watchdog timer. 0 = Writing 0 to this bit has no effect. 1 = Reset the contents of the Watchdog timer. **Note:** This bit will auto clear after few clock cycles

5.11 Audio Processing Unit (APU)

The main purpose of Audio Processing Unit (APU) is used to playback the audio data (PCM format) which are decoded and stored in local buffer by CPU. The APU builds in a H/W mixer and a digital to analog convertor (DAC) with 13-bit resolution, and a Power Amplifier (PA) with 7-level volume adjustment.

There are two channels input for the H/W mixer. H/W Mixer mixes inputs and clips the mixed output to 13 bits.

If DAC is enabled, Timer0 is used as the channel 0 sample rate control timer automatically. User needs to program the time-out rate of Timer0 as the audio data sample rate.

There is an 8-level buffer for channel 0 audio data buffer. Whenever Timer0 time-out, one sample is read out from buffer sequentially to DAC. Once the data in the defined threshold of buffer is read out, a threshold interrupt is generated if its corresponding interrupt is enabled.

There is only one level buffer for channel 1 audio data buffer. No Timer H/W is associated with channel 1.

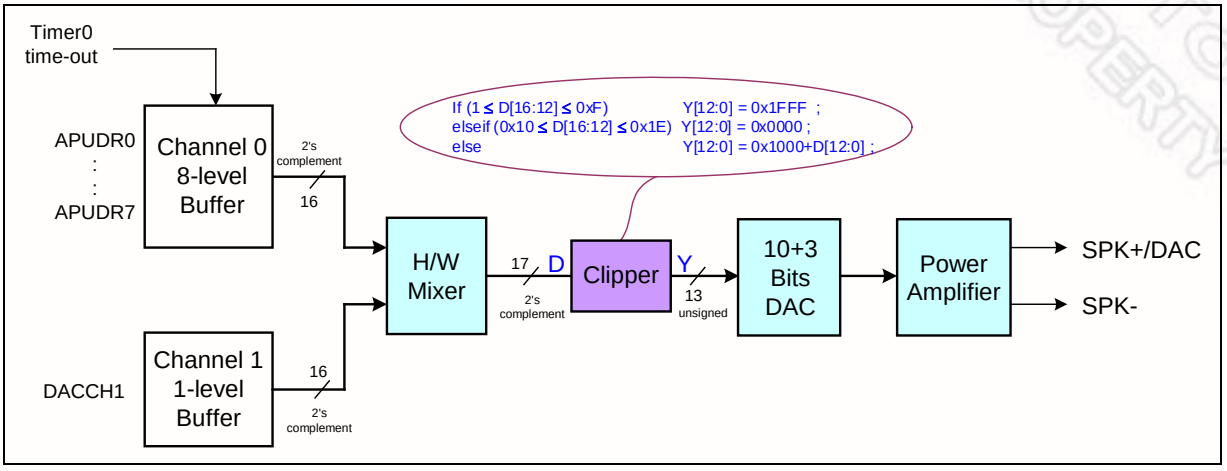


Figure 5-20 APU Block Diagram

5.11.1 APU Control Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
APU Base Address: **APU_BA = 0x5000_8000**				
APU_CTL	APU_BA+0x00	R/W	APU Control Register	0x0000_0000
APU_VM	APU_BA+0x04	R/W	APU Volume Control Register	0x0000_0000
APU_CH1DAT0	APU_BA+0x0C	R/W	APU Channel 1 Data Buffer Register	0x0000_0000
APU_CH0DAT0	APU_BA+0x10	R/W	APU Channel 0 Data Buffer Register 0	0x0000_0000
APU_CH0DAT1	APU_BA+0x14	R/W	APU Channel 0 Data Buffer Register 1	0x0000_0000
APU_CH0DAT2	APU_BA+0x18	R/W	APU Channel 0 Data Buffer Register 2	0x0000_0000
APU_CH0DAT3	APU_BA+0x1C	R/W	APU Channel 0 Data Buffer Register 3	0x0000_0000
APU_CH0DAT4	APU_BA+0x20	R/W	APU Channel 0 Data Buffer Register 4	0x0000_0000
APU_CH0DAT5	APU_BA+0x24	R/W	APU Channel 0 Data Buffer Register 5	0x0000_0000
APU_CH0DAT6	APU_BA+0x28	R/W	APU Channel 0 Data Buffer Register 6	0x0000_0000
APU_CH0DAT7	APU_BA+0x2C	R/W	APU Channel 0 Data Buffer Register 7	0x0000_0000

5.11.2 APU Control Register Description

APU Control Register (APU_CTL)

Register	Offset	R/W	Description	Reset Value
APU_CTL	APU_BA+0x00	R/W	APU Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved		DACGN	Reserved			BPPAM	DACE
7	6	5	4	3	2	1	0
PAMPE	APUIE	APUIS	Reserved		TSHD		

Bits	Description	
[31:14]	**Reserved**	Reserved.
[13]	**DACGN**	**DAC Output Current Control** 0 = 3mA. 1 = 5mA. This bit is effective only when BPPAM is “1”.
[12:10]	**Reserved**	Reserved.
[9]	**BPPAM**	**Bypass Power Amplifier, DAC Output To Pin** 0 = SPK+/DAC pin is one of power amplifier output,. 1 = SPK+/DAC pin is current DAC output, no output on SPK-.
[8]	**DACE**	**DAC Enable** 0 = Disable DAC function. 1 = Enable DAC function.
[7]	**PAMPE**	**Power Amplifier Enable** 0 = Disable PA function. 1 = Enable PA function.
[6]	**APUIE**	**APU Interrupt Enable** 0 = Disable the APU threshold interrupt. 1 = Enable the APU threshold interrupt.

[5]	**APUIS**	**APU Interrupt Status** 0 = APU threshold interrupt does not occur. 1 = APU threshold interrupt occur. This flag is set by hardware when APU threshold is met. Software can clear this bit by writing a zero to it.
[4:3]	**Reserved**	Reserved.
[2:0]	**TSHD**	**APU Interrupt Threshold** 000 = Buffer 0 is read out by APU. 001 = Buffer 1 is read out by APU. 010 = Buffer 2 is read out by APU. 011 = Buffer 3 is read out by APU. 100 = Buffer 4 is read out by APU. 101 = Buffer 5 is read out by APU. 110 = Buffer 6 is read out by APU. 111 = Buffer 7 is read out by APU.

APU Volume Control Register (APU_VM)

Register	Offset	R/W	Description	Reset Value
APU_VM	APU_BA+0x04	R/W	APU Volume Control Register	0x0000_0000

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

Bits	Description	
[31:3]	**Reserved**	Reserved.
[2:0]	**VOLUM**	**APU Volume Adjustment** 000 = 0 dB. 001 = -3 dB. 010 = -6 dB. 011 = -9 dB. 100 = -12 dB. 101 = -15 dB. 110 = -18 dB. 111 = Reserved.

APU Channel 1 Data Register (APU_CH1DAT0)

Register	Offset	R/W	Description	Reset Value
APU_CH1DAT0	APU_BA+0x0C	R/W	APU Channel 1 Data Buffer Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PCM1[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PCM1[7:0]							

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**PCM1**	**PCM Data Of Channel 1** This field contains 16-bit PCM data that is one of the Mixer input. User needs to take care of the effective bit of PCM because the H/W mixer output is clipped to 13 bits automatically. The data format of PCM is 2'complement.

APU Channel 0 Data Register (APU_CH0DATn)

Register	Offset	R/W	Description	Reset Value
APU_CH0DAT0	APU_BA+0x10	R/W	APU Channel 0 Data Buffer Register 0	0x0000_0000
APU_CH0DAT1	APU_BA+0x14	R/W	APU Channel 0 Data Buffer Register 1	0x0000_0000
APU_CH0DAT2	APU_BA+0x18	R/W	APU Channel 0 Data Buffer Register 2	0x0000_0000
APU_CH0DAT3	APU_BA+0x1C	R/W	APU Channel 0 Data Buffer Register 3	0x0000_0000
APU_CH0DAT4	APU_BA+0x20	R/W	APU Channel 0 Data Buffer Register 4	0x0000_0000
APU_CH0DAT5	APU_BA+0x24	R/W	APU Channel 0 Data Buffer Register 5	0x0000_0000
APU_CH0DAT6	APU_BA+0x28	R/W	APU Channel 0 Data Buffer Register 6	0x0000_0000
APU_CH0DAT7	APU_BA+0x2C	R/W	APU Channel 0 Data Buffer Register 7	0x0000_0000

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

Bits	Description	
[31:16]	**Reserved**	Reserved.
[15:0]	**PCM**	**PCM Data Of Channel 0** This field contains 16-bit PCM data that will be sent to mixer H/W. User needs to take care of the effective bit of PCM because the H/W Mixer output is clipped to 13 bits automatically. The data format of PCM is 2'complement.

5.12 12-bit Analog-to-Digital Converter (ADC) and Pre-amplifier

N572F072/P072 contains one 12-bit successive approximation analog-to-digital converters (ADC). ADC can be programmed operation independently. ADC has 8-channel inputs. The A/D converter supports three operation modes: single, single-cycle scan and continuous scan mode. There are two kinds of scan mode: continuous mode and single cycle mode. Each of A/D converters can be started by software and external STADC pin.

Note: The PCLK is equivalent to HCLK thereafter.

5.12.1 Features

- Analog input voltage range: 0~5.0V
- 12-bits resolution and 10-bits accuracy is guaranteed
- Total 8 analog inputs: ADC has 8-channel inputs
- Maximum ADC clock frequency is 8MHz
- Up to 320K SPS conversion rate
- Three operating modes
  - Single mode: Single channel A/D conversion
  - Single-cycle scan mode: Conversion on all enabled channels once
  - Continuous scan mode: Repetitive conversion on enabled channels
- An A/D conversion can be started by
  - Software write SWTRG bit
  - External pin STADC
- Conversion results are held in data registers for each channel with valid and overrun indicators
- Conversion result can be compared with specified value and user can select whether to generate an interrupt when conversion result is equal to the compare register setting.
- Support single-end

5.12.2 ADC Block Diagram

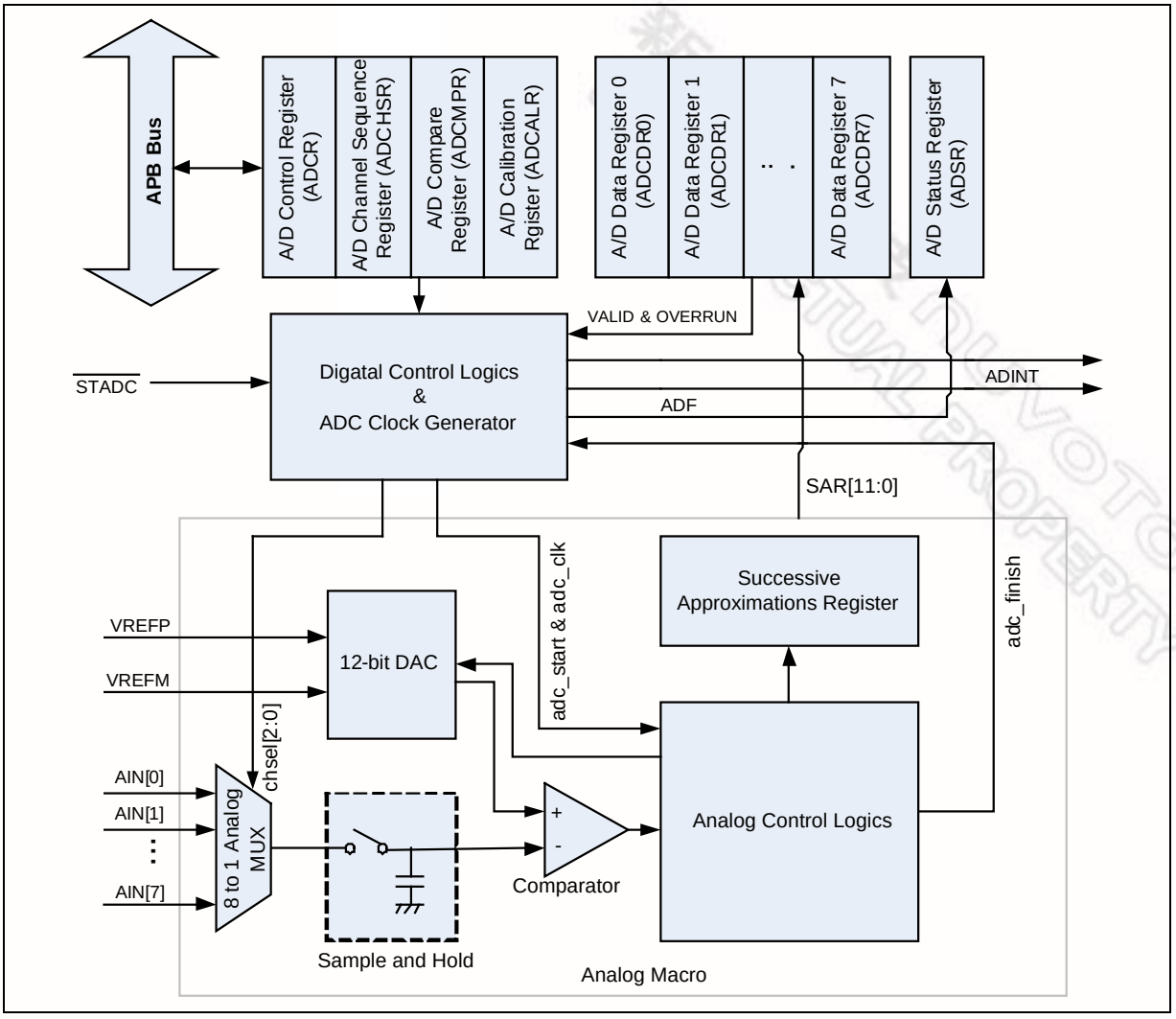


Figure 5-21 ADC Controller Block Diagram

5.12.3 ADC Operation Procedure

The A/D converter operates by successive approximation with 12-bit resolution. The ADCs have two operation modes: single mode and scan mode. There are two kinds of scan mode: continuous mode and single-cycle mode. When changing the operating mode or analog input channel enable, in order to prevent incorrect operation, software must clear SWTRG bit to 0 in ADC_CTL register. The A/D converter discards current conversion immediately and enters idle state while SWTRG bit is cleared

5.12.3.1 *ADC Clock Generator*

The maximum sampling rate is up to 320 KHz and the conversion time is less 3.125μS. It needs 25 ADC clocks to complete an A/D conversion. The maximum clock rate to A/D conversion module should be less than 25 x sampling rate. The ADC clock source can be from PLL, HCLK or 48MHz RC oscillator. The selected clock source is divided by (ADC_N+1) to produce the clock to A/D converter which should be less than or equal to 8MHz.

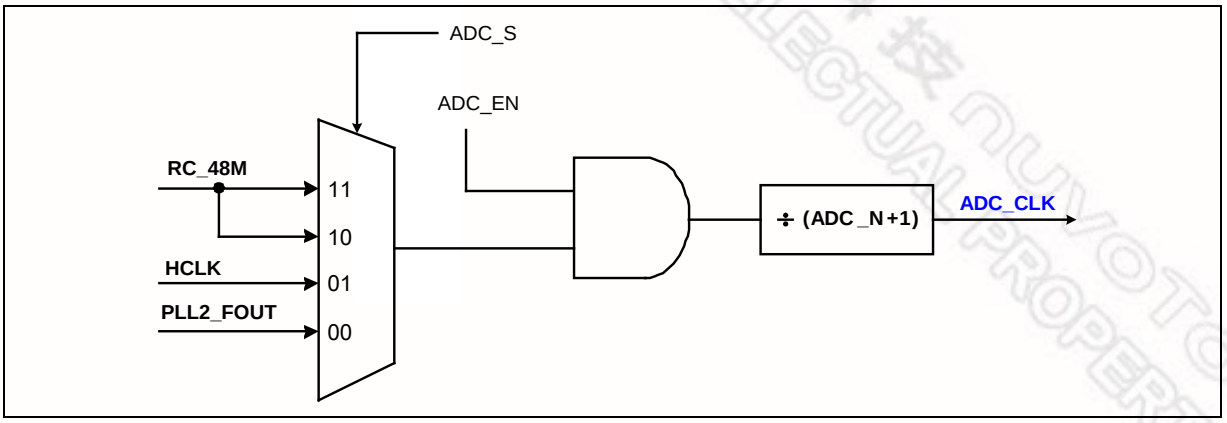


Figure 5-22 ADC Clock Source

5.12.3.2 *Single Mode*

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

1. The channel defined in CHSEQ0 is the channel to be converted.
2. A/D conversion is started when the SWTRG bit in ADC_CTL is set to 1 by software or external trigger input.
3. When A/D conversion is finished, the result is transferred to the A/D data register ADC_DATn corresponding to the channel.
4. On completion of conversion, the ADIF bit in ADC_STATUS is set to 1. If the ADCIE bit is 1, an ADINT interrupt request is generated.
5. The SWTRG bit remains “1” during A/D conversion. When A/D conversion ends, the SWTRG bit is automatically cleared to 0 and the A/D converter enters the idle state.
6. When the SWTRG bit is cleared to 0 during A/D conversion, A/D conversion stops and the A/D converter enters the idle state.

5.12.3.3 *Continuous Scan Mode*

In continuous scan mode, A/D conversion is to be performed sequentially on the specified channels that are defined by CHSEQx bits in ADC_CHSEQ register (8 channels maximum). The operations are as follows:

1. When the SWTRG bit in ADC_CTL is set to 1 by software or external trigger input, A/D conversion starts on the channel selected by CHSEQ0.
2. When A/D conversion for each channel selected by channel sequence register (ADC_CHSEQ) is completed, the converted result is sequentially transferred to the A/D data register corresponding to channel sequence.
3. When the conversion for all the selected channels is completed, the ADIF bit in ADC_STATUS is set to 1. If the ADCIE bit is 1, an ADINT interrupt is requested after A/D conversion ends. Conversion of the channel defined in CHSEQ0 starts again.
4. Steps 2 to 3 are repeated as long as the SWTRG bit remains to 1. When the SWTRG 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 mode on 3 channels (CHSEQ0=0, CHSEQ1=3, CHSEQ2=0, CHSEQ3=6 and CHSEQ4=F) is shown as below:

(This example is only appropriate for ADC)

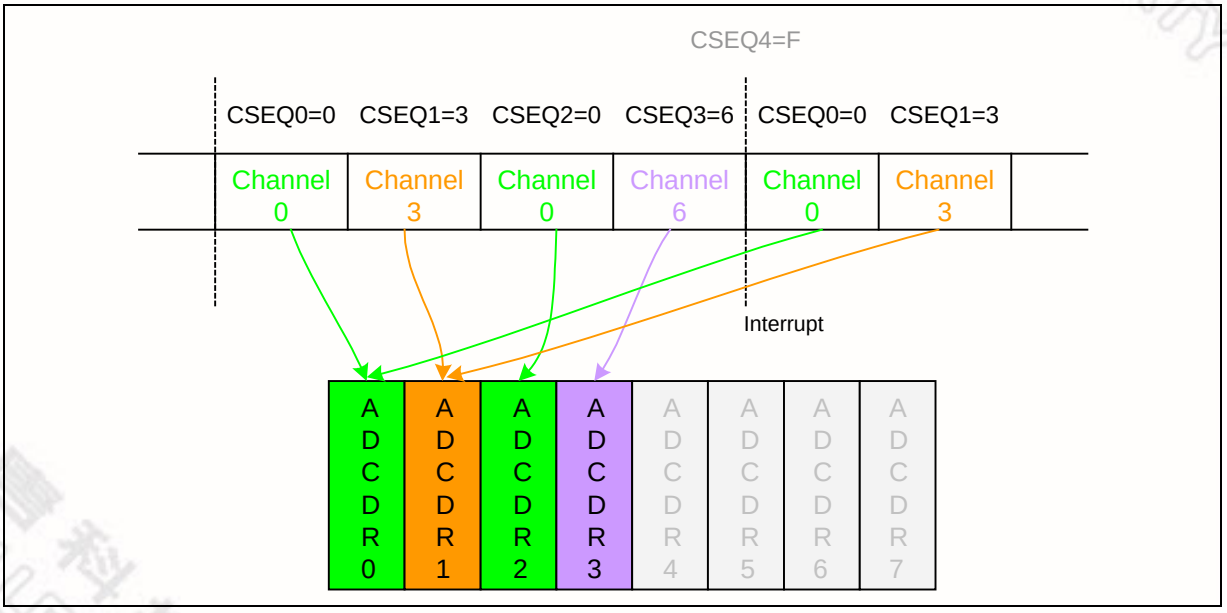


Figure 5-23 Continuous Scan on Selected Channels

5.12.3.4 *Single-Cycle Scan Mode*

In single-cycle scan mode, A/D conversion is to be performed once on the specified channels that are defined by CHSEQx bits in ADC_CHSEQ register (8 channels maximum for ADC). Operations are as follows:

1. When the SWTRG bit in ADC_CTL is set to 1 by a software, or external trigger input, A/D conversion starts on the channel selected by CHSEQ0.
2. When A/D conversion for channel selected by channel sequence registers (CHSEQx) is completed, the result is sequentially transferred to the A/D data register corresponding to channel sequence.
3. When the conversion for all the selected channels is completed, the ADIF bit in ADC_STATUS is set to 1. If the ADCIE bit is 1, an ADINT interrupt is requested after A/D conversion ends.
4. After A/D conversion ends, the SWTRG bit is automatically cleared to 0 and the A/D converter enters the idle state. When the SWTRG bit is cleared to 0 during A/D conversion, A/D conversion stops and the A/D converter enters the idle state.

An example timing diagram for single-cycle scan on 4 channels (CHSEQ0=0, CHSEQ1=2, CHSEQ2=3, CHSEQ3=7 and CHSEQ4=F) is shown as below:

(This example is only appropriate for ADC)

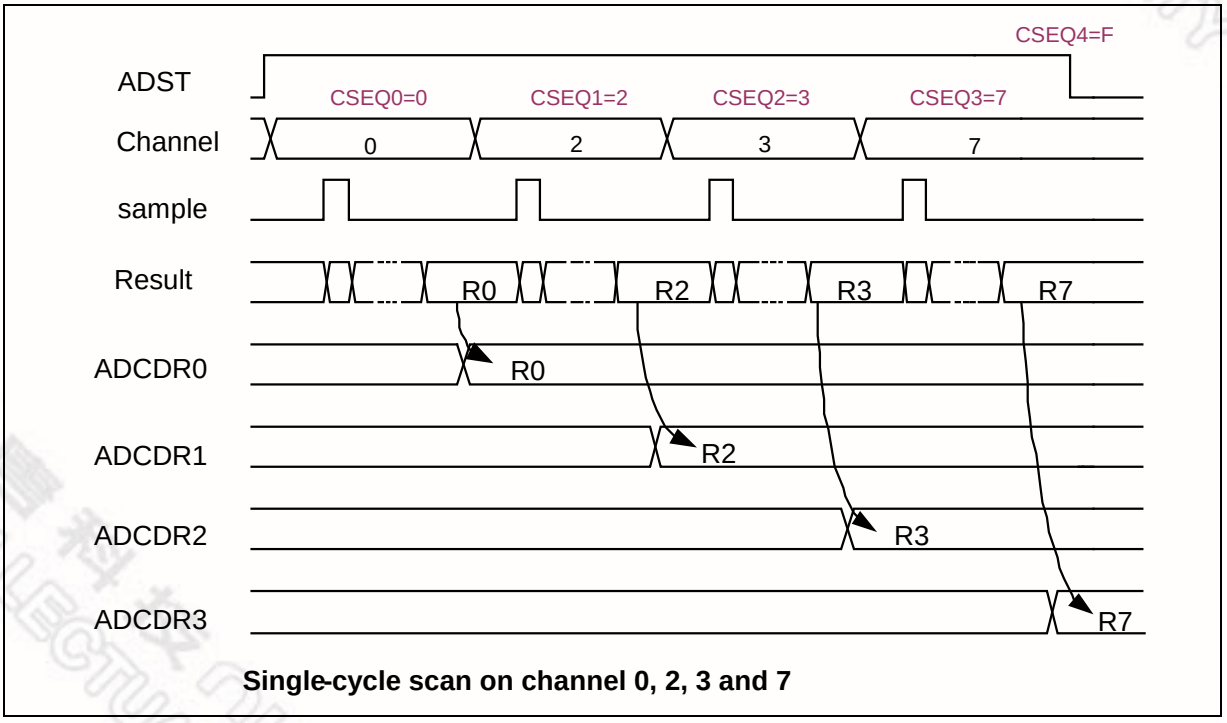


Figure 5-24 Single-Cycle Scan on selected Channels

5.12.3.5 *Input Sampling and A/D Conversion Time*

The A/D converter has a built-in sample-and-hold circuit for each module, the A/D samples the analog input when A/D conversion start delay time (Td) has passed after SWTRG bit in ADC_CTL is set to 1, then start conversion. Due to ADC clock is generated by PCLK divided by (N+1), the maximum delay time from APB write to A/D start sampling analog input time is 2N PCLKs. The start delay time is shown as below:

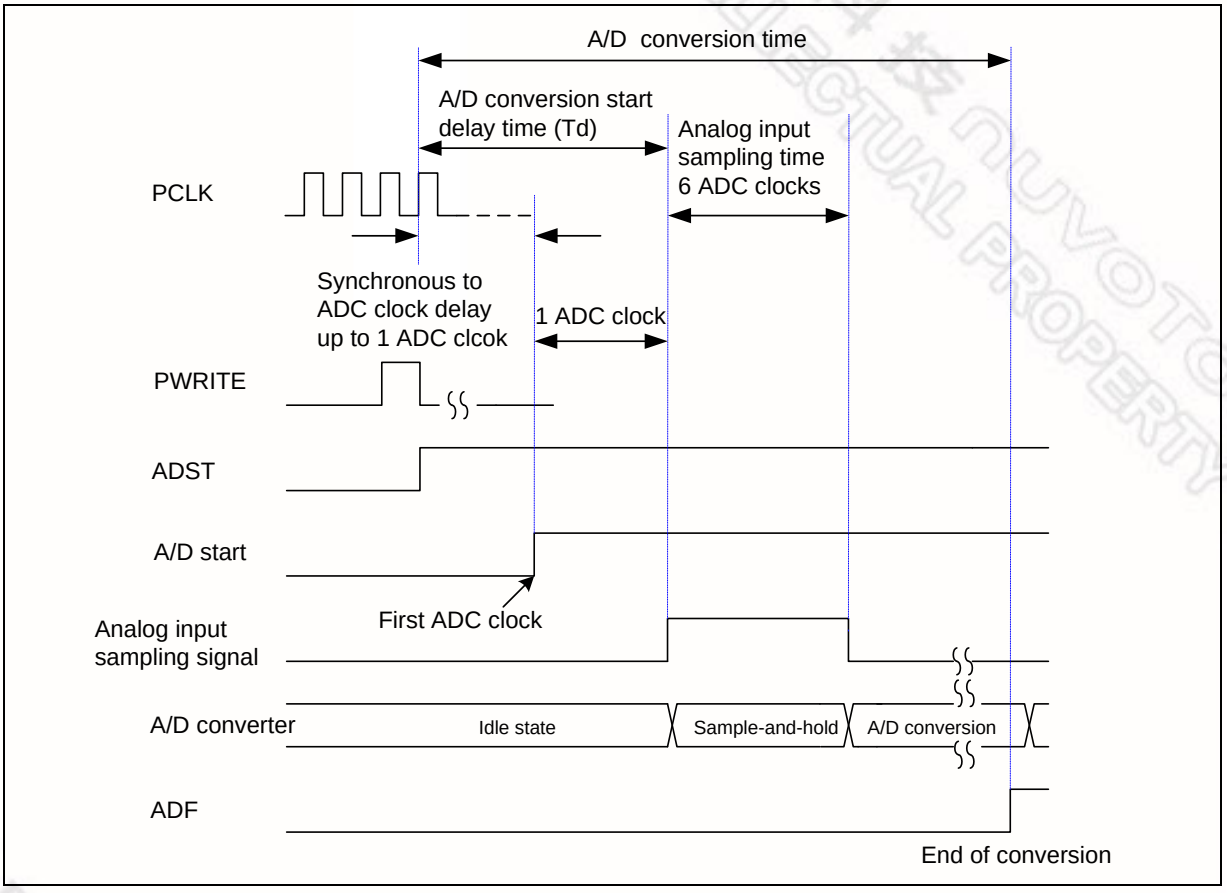


Figure 5-25 Conversion Start Delay Timing Diagram

A/D conversion can be triggered by external pin request. External trigger input is enabled at the STADC pin. Software can set HWTRGCOND 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 SWTRG bit will be set to 1 at the 9th PCLK and start to conversion. Conversion is continuous if external trigger input is pull at low (or high state) in level trigger mode. It is stopped only when external trigger condition disappears. If edge trigger condition is selected, the high and low state must be kept at least 4 PLCKs. Pulse that is shorter than this specification will be ignored. Other operations, in both single and scan modes, are the same as when the SWTRG bit has been set to 1.

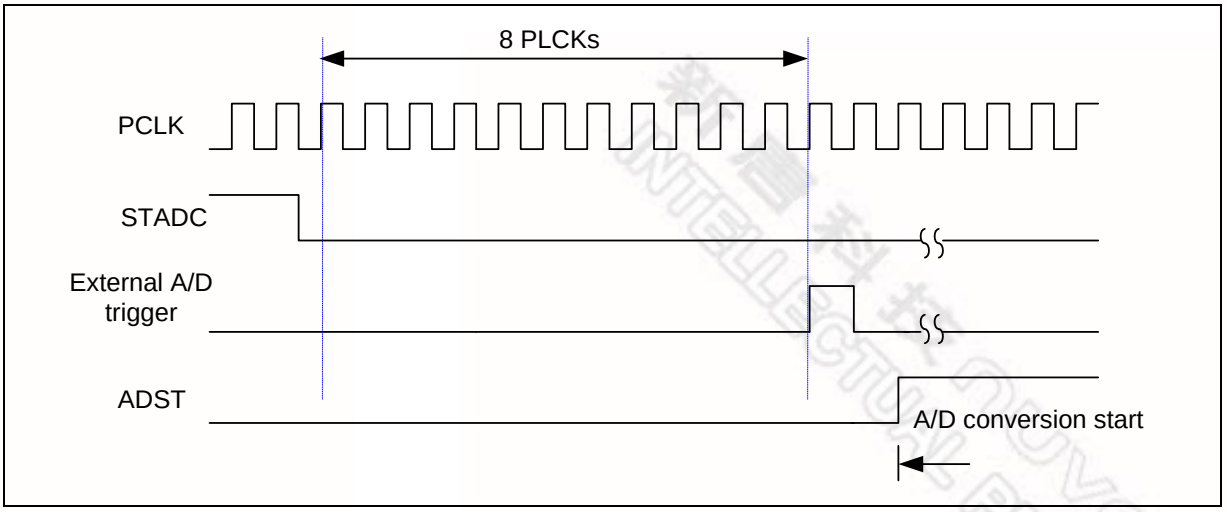


Figure 5-26 External A/D Conversion Trigger Timing Diagram

5.12.3.6 *Conversion Result Monitor*

N572F072/P072 ADC controller provides two compare registers ADC_CMP0/1 to monitor specified channel conversion result from A/D conversion module (see figure below). Software can select which channel to be monitored by set CMPCH and CMPCOND bit is used to check conversion result is less than or greater than (equal to) specified value in CMPDAT. When the compared result meets the setting, compare match counter will increase 1. When counter value reaches the setting of CMPMCNT then ADCMPF bit will be set to 1. If ADCMPIE bit is set, then an ADINT interrupt request is generated. The block diagram is shown as below:

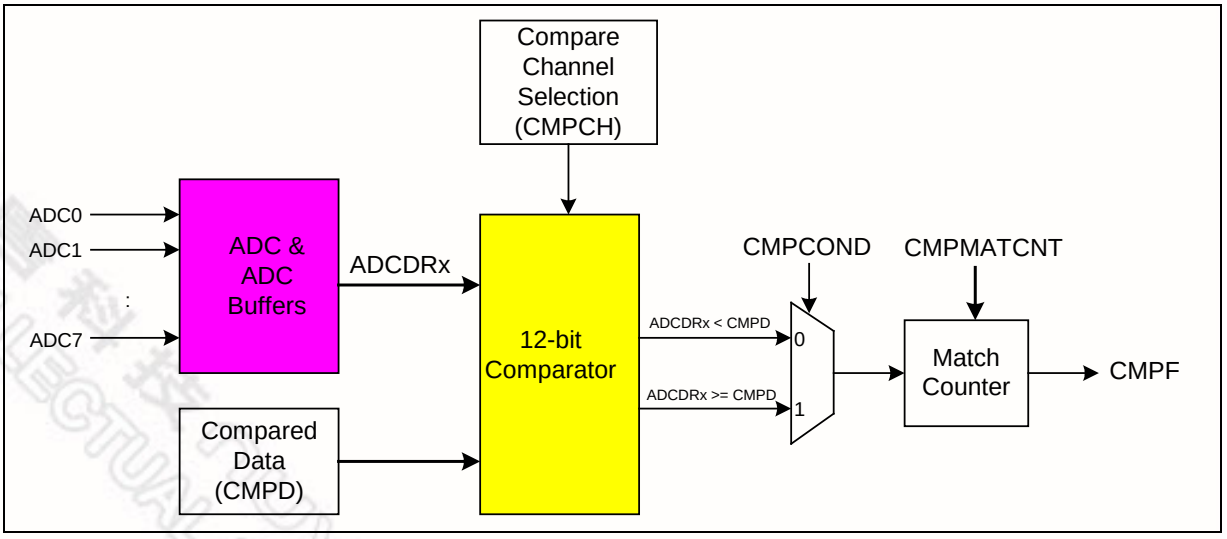


Figure 5-27 A/D Conversion Result Comparison

5.12.3.7 *Interrupt Sources*

The A/D converter generates a conversion end ADIF in ADC_STATUS register upon the end of A/D conversion. If ADCIE bit in ADC_CTL is set then conversion end interrupt request ADINT is generated. If ADCMPIE bit is enabled, when A/D conversion result meets setting in ADC_CMPn register, monitor interrupt is generated, ADINT will be set also. CPU can clear ADCMPF and ADIF to stop interrupt request.

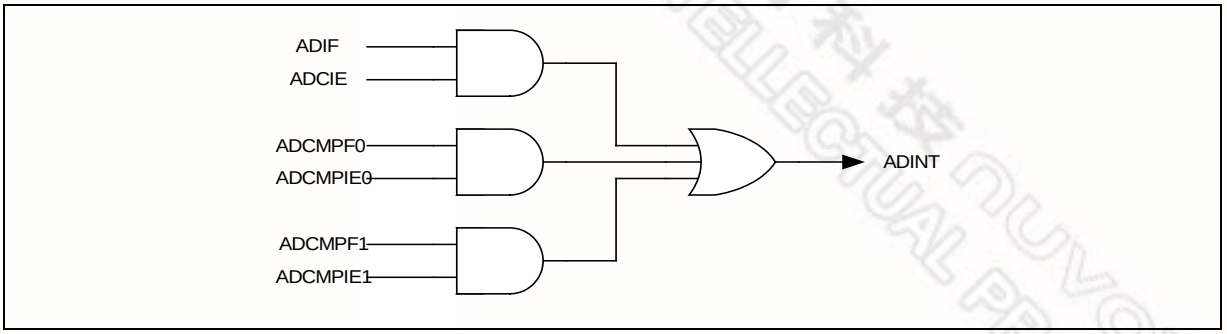


Figure 5-28 A/D Controller Interrupt

5.12.4 Pre-amplifier Block Diagram

For Voice recording application, N572F072/P072 equips with microphone bias output, and programmable gain control (PGC). There is a dedicated regulator for the whole PGC and ADC block with output voltage 0.85 time the voltage of AVDD.

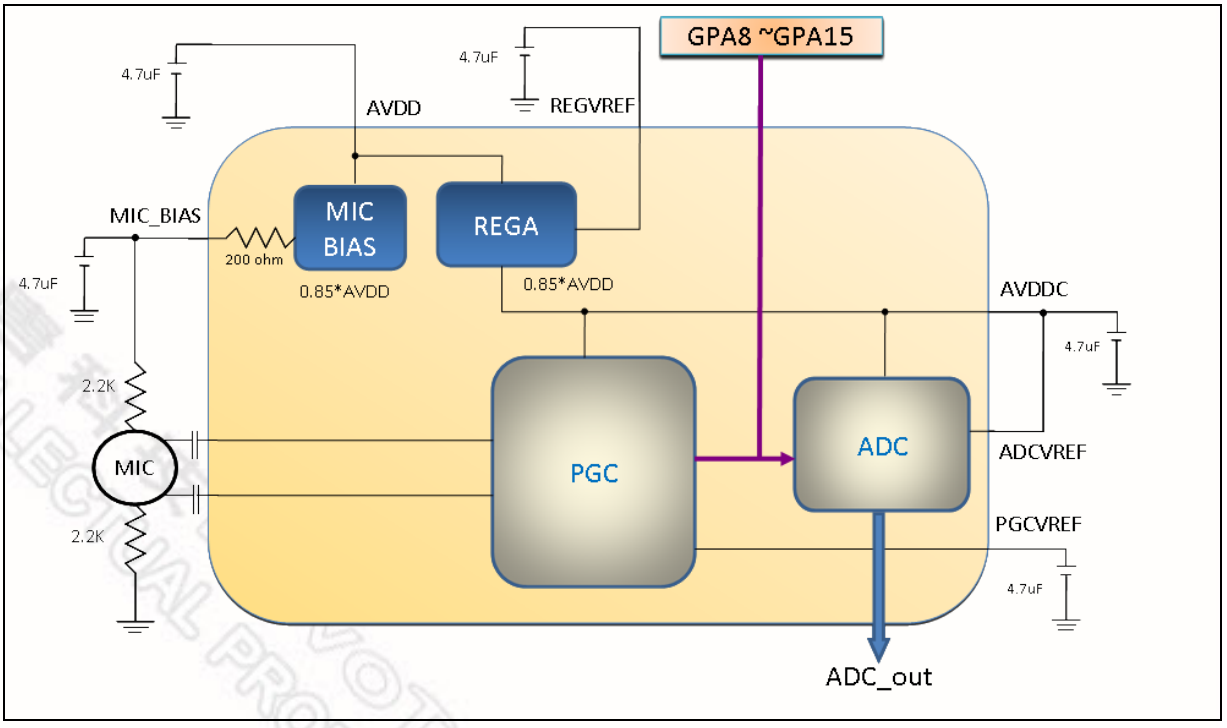


Figure 5-29 Mic Bias and Pre-amplifier

5.12.5 ADC and Pre-amplifier Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
ADC Base Address: **ADC_BA = 0x400E_0000**				
ADC_DAT0	ADC_BA+0x00	R	A/D Data Register for the channel defined in CHSEQ0	0x0000_0000
ADC_DAT1	ADC_BA+0x04	R	A/D Data Register for the channel defined in CHSEQ1	0x0000_0000
ADC_DAT2	ADC_BA+0x08	R	A/D Data Register for the channel defined in CHSEQ2	0x0000_0000
ADC_DAT3	ADC_BA+0x0C	R	A/D Data Register for the channel defined in CHSEQ3	0x0000_0000
ADC_DAT4	ADC_BA+0x10	R	A/D Data Register for the channel defined in CHSEQ4	0x0000_0000
ADC_DAT5	ADC_BA+0x14	R	A/D Data Register for the channel defined in CHSEQ5	0x0000_0000
ADC_DAT6	ADC_BA+0x18	R	A/D Data Register for the channel defined in CHSEQ6	0x0000_0000
ADC_DAT7	ADC_BA+0x1C	R	A/D Data Register for the channel defined in CHSEQ7	0x0000_0000
ADC_CTL	ADC_BA+0x20	R/W	A/D Control Register	0x0000_0000
ADC_CHSEQ	ADC_BA+0x24	R/W	A/D Channel Sequence Register	0xFFFF_FFFF
ADC_CMP0	ADC_BA+0x28	R/W	A/D Compare Register 0	0x0000_0000
ADC_CMP1	ADC_BA+0x2C	R/W	A/D Compare Register 1	0x0000_0000
ADC_STATUS	ADC_BA+0x30	R/W	A/D Status Register	0x0000_0070
ADC_PGCTL	ADC_BA+0x3C	R/W	ADC Pre-amplifier Gain Control Register	0x0001_0000
ADC_MIBSCTL	ADC_BA+0x40	R/W	MIC bias and PGC Control Register	0x0000_0033

5.12.6 ADC and Pre-amplifier Register Description

A/D Data Registers (ADC_DATn)

Register	Offset	R/W	Description	Reset Value
ADC_DAT0	ADC_BA+0x00	R	A/D Data Register for the channel defined in CHSEQ0	0x0000_0000
ADC_DAT1	ADC_BA+0x04	R	A/D Data Register for the channel defined in CHSEQ1	0x0000_0000
ADC_DAT2	ADC_BA+0x08	R	A/D Data Register for the channel defined in CHSEQ2	0x0000_0000
ADC_DAT3	ADC_BA+0x0C	R	A/D Data Register for the channel defined in CHSEQ3	0x0000_0000
ADC_DAT4	ADC_BA+0x10	R	A/D Data Register for the channel defined in CHSEQ4	0x0000_0000
ADC_DAT5	ADC_BA+0x14	R	A/D Data Register for the channel defined in CHSEQ5	0x0000_0000
ADC_DAT6	ADC_BA+0x18	R	A/D Data Register for the channel defined in CHSEQ6	0x0000_0000
ADC_DAT7	ADC_BA+0x1C	R	A/D Data Register for the channel defined in CHSEQ7	0x0000_0000

The A/D converted results are saved in these ADC_DATn registers sequentially. The channel scan sequence is defined in ADC_HSR register.

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**VALID**	**OV**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
EXTS				**RESULT[11:8]**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RESULT[7:0]							

Bits	Description	
[31:18]	**Reserved**	Reserved.
[17]	**VALID**	**Valid Flag** 0 = Data in RESULT are not valid. 1 = Data in RESULT are valid. This bit is set to 1 when corresponding channel analog input conversion is completed and cleared by hardware after ADC_DAT register is read.

[16]	**OV**	**Over Run Flag** 0 = Data in RESULT are recent conversion result. 1 = Data in RESULT are overwritten. If converted data in RESULT[11:0] have not been read before new conversion result is loaded to this register, OV is set to 1. It is cleared by hardware after ADC_DAT register is read.
[15:12]	**EXTS**	**Extension Bits Of RESULT for Different Data Format** If ADCFM is “0”, EXTS all are read as “0”. If ADCFM is “1”, EXTS all are read as bit RESULT[11].
[11:0]	**RESULT**	**A/D Conversion Result** This field contains the 12-bit conversion result. Its data format is defined by ADCFM bit.

A/D Control Register (ADC_CTL)

Register	Offset	R/W	Description	Reset Value
ADC_CTL	ADC_BA+0x20	R/W	A/D Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**ADCFM**	**SWTRG**	**Reserved**		**HWTRGEN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
HWTRGCOND		**Reserved**		**OPMODE**		**ADCIE**	**ADCEN**

Bits	Description	
[31:13]	**Reserved**	Reserved.
[12]	**ADCFM**	**Data Format Of ADC Conversion Result** 0 = Unsigned 1 = 2'Complemet
[11]	**SWTRG**	**A/D Conversion Start** 0 = Conversion is stopped and A/D converter enters idle state. 1 = Start conversion. **Note**: SWTRG bit can be set to 1 from three sources: software write and external pin STADC. SWTRG is cleared to 0 by hardware automatically at the end of single mode and single-cycle scan mode on specified channel. In continuous scan mode, A/D conversion is continuously performed sequentially until software writes 0 to this bit or chip resets.
[10:9]	**Reserved**	Reserved.
[8]	**HWTRGEN**	**Trigger Enable** Enable or disable triggering of A/D conversion by external STADC pin. 0 = Disable. 1 = Enable.
[7:6]	**HWTRGCOND**	**Trigger Condition** These two bits decide external pin STADC trigger event is level or edge. The signal must be kept at stable state at least 8 PCLKs for level trigger and 4 PCLKs at high and low state. 00 = Low level 01 = High level 10 = Falling edge 11 = Positive edge
[5:4]	**Reserved**	Reserved.

[3:2]	**OPMODE**	**A/D Converter Operation Mode** 00 = Single conversion 01 = Reserved 10 = Single-cycle scan 11 = Continuous scan When changing the operation mode, software should disable SWTRG bit firstly.
[1]	**ADCIE**	**A/D Interrupt Enable** 0 = Disable A/D interrupt function 1 = Enable A/D interrupt function A/D conversion end interrupt request is generated if ADCIE bit is set to 1.
[0]	**ADCEN**	**A/D Converter Enable** 0 = Disable 1 = Enable Before starting A/D conversion function, this bit should be set to 1. Clear it to 0 to disable A/D converter analog circuit power consumption.

A/D Channel Sequence Register (ADC_CHSEQ)

Register	Offset	R/W	Description	Reset Value
ADC_CHSEQ	ADC_BA+0x24	R/W	A/D Channel Sequence Register	0xFFFF_FFFF

31	30	29	28	27	26	25	24
CHSEQ7				CHSEQ6			
23	22	21	20	19	18	17	16
CHSEQ5				CHSEQ4			
15	14	13	12	11	10	9	8
CHSEQ3				CHSEQ2			
7	6	5	4	3	2	1	0
CHSEQ1				CHSEQ0			

Bits	Description	
[31:28]	**CHSEQ7**	**Select Channel N As The 8th Conversion In Scan Sequence** The definition of channel selection is the same as CHSEQ0.
[27:24]	**CHSEQ6**	**Select Channel N As The 7th Conversion In Scan Sequence** The definition of channel selection is the same as CHSEQ0.
[23:20]	**CHSEQ5**	**Select Channel N As The 6th Conversion In Scan Sequence** The definition of channel selection is the same as CHSEQ0.
[19:16]	**CHSEQ4**	**Select Channel N As The 5th Conversion In Scan Sequence** The definition of channel selection is the same as CHSEQ0.
[15:12]	**CHSEQ3**	**Select Channel N As The 4th Conversion In Scan Sequence** The definition of channel selection is the same as CHSEQ0.
[11:8]	**CHSEQ2**	**Select Channel N As The 3rd Conversion In Scan Sequence** The definition of channel selection is the same as CHSEQ0.
[7:4]	**CHSEQ1**	**Select Channel N As The 2nd Conversion In Scan Sequence** The definition of channel selection is the same as CHSEQ0.

<table>
<tr>
<td>[3:0]</td>
<td>CHSEQ0</td>
<td>
Select Channel N As The 1st Conversion In Scan Sequence
If CHSEQ0[3]=0, one of the following channel is selected according to CHSEQ0[2:0].
<table>
<tr><td>CHSEQ0</td><td>Selected channel to ADC</td></tr>
<tr><td>0000</td><td>Channel 0</td></tr>
<tr><td>0001</td><td>Channel 1</td></tr>
<tr><td>0010</td><td>Channel 2</td></tr>
<tr><td>0011</td><td>Channel 3</td></tr>
<tr><td>0100</td><td>Channel 4</td></tr>
<tr><td>0101</td><td>Channel 5</td></tr>
<tr><td>0110</td><td>Channel 6</td></tr>
<tr><td>0111</td><td>Channel 7</td></tr>
</table>
If CHSEQ0[3]=1,.
For CHSEQ0[0].=0,.
Output of pre-amplifier with differential input pair is selected as ADC input. The input pair of pre-amplifier is defined in CHSEQ0[2:1].
<table>
<tr><td>CHSEQ0</td><td>Channel pair to pre-amplifier input</td></tr>
<tr><td>1000</td><td>Channel 0 and 1</td></tr>
<tr><td>1010</td><td>Channel 2 and 3</td></tr>
<tr><td>1100</td><td>Channel 4 and 5</td></tr>
<tr><td>1110</td><td>None</td></tr>
</table>
For CHSEQ0[0].=1,
1xx1 = no channel is selected, scan sequence end.
</td>
</tr>
</table>

A/D Compare Register 0/1 (ADC_CMPn)

Register	Offset	R/W	Description	Reset Value
ADC_CMP0	ADC_BA+0x28	R/W	A/D Compare Register 0	0x0000_0000
ADC_CMP1	ADC_BA+0x2C	R/W	A/D Compare Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				CMPDAT[11:8]			
23	22	21	20	19	18	17	16
CMPDAT[7:0]							
15	14	13	12	11	10	9	8
Reserved				CMPMCNT			
7	6	5	4	3	2	1	0
Reserved		CMPCH			CMPCOND	ADCMPIE	ADCMPEN

Bits	Description	
[31:28]	**Reserved**	Reserved.
[27:16]	**CMPDAT**	**Compare Data** The 12 bits data are used to compare with conversion result of specified channel. Software can use it to monitor the external analog input pin voltage transition in scan mode without imposing a load on software. The data format should be consistent with the setting of ADCFM bit.
[15:12]	**Reserved**	Reserved.
[11:8]	**CMPMCNT**	**Compare Match Count** When the specified A/D channel analog conversion result matches the comparing condition, the internal match counter will increase 1. When the internal counter achieves the setting, (CMPMCNT+1) hardware will set the ADCMPF bit.
[7:6]	**Reserved**	Reserved.
[5:3]	**CMPCH**	**Compare Channel Selection** 000 = Channel 0 conversion result is selected to be compared. 001 = Channel 1 conversion result is selected to be compared. 010 = Channel 2 conversion result is selected to be compared. 011 = Channel 3 conversion result is selected to be compared. 100 = Channel 4 conversion result is selected to be compared. 101 = Channel 5 conversion result is selected to be compared. 110 = Channel 6 conversion result is selected to be compared. 111 = Channel 7 conversion result is selected to be compared.
[2]	**CMPCOND**	**Compare Condition** 0 = ADCMPFx bit is set if conversion result is less than CMPDAT[11:0]. 1 = ADCMPFx bit is set if conversion result is greater or equal to CMPDAT[11:0],

[1]	**ADCMPIE**	**Compare Interrupt Enable** 0 = Disable 1 = Enable When converted data in RESULT is less (or greater) than the compare data CMPDAT[11:0], ADCMPF bit is asserted. If ADCMPIE is set to 1, a compare interrupt request is generated.
[0]	**ADCMPEN**	**Compare Enable** 0 = Disable compare. 1 = Enable compare. Set this bit to 1 to enable the comparison CMPDAT[11:0] with specified channel conversion result when converted data is loaded into ADC_DAT register.

A/D Status Register (ADC_STATUS)

Register	Offset	R/W	Description	Reset Value
ADC_STATUS	ADC_BA+0x30	R/W	A/D Status Register	0x0000_0070

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
OV							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VALID							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**CHANNEL**			**BUSY**	**ADCMPF1**	**ADCMPF0**	**ADIF**

Bits	Description	
[31:24]	**Reserved**	Reserved.
[23:16]	**OV**	**Over Run Flag** It is a mirror to OV bit in ADC_DATn.
[15:8]	**VALID**	**Data Valid Flag** It is a mirror of VALID bit in ADC_DATn.
[7]	**Reserved**	Reserved.
[6:4]	**CHANNEL**	**Current Conversion Channel** This filed reflects current conversion channel when BUSY=1. When BUSY=0, it shows the next channel will be converted. It is read only.
[3]	**BUSY**	**BUSY/IDLE** 0 = A/D converter is in idle state. 1 = A/D converter is busy at conversion. This bit is mirror of SWTRG bit in ADC_CTL. It is read only.
[2]	**ADCMPF1**	**Compare Flag** When the selected channel A/D conversion result meets setting conditions in ADC_CMP1, then this bit is set to 1. And it is cleared by write 1. 0 = Converted result RESULT in ADC_DAT does not meet ADC_CMP1 setting. 1 = Converted result RESULT in ADC_DAT meets ADC_CMP1 setting,
[1]	**ADCMPF0**	**Compare Flag** When the selected channel A/D conversion result meets setting conditions in ADC_CMP0, then this bit is set to 1. And it is cleared by write 1. 0 = Converted result RESULT in ADC_DAT does not meet ADC_CMP0 setting. 1 = Converted result RESULT in ADC_DAT meets ADC_CMP0 setting,

[0]	**ADIF**	**A/D Conversion End Flag** A status flag that indicates the end of A/D conversion. ADIF is set to 1 under the following two conditions: 1. When A/D conversion ends in single mode, 2. When A/D conversion ends on all channels specified by channel sequence register in scan mode. And it is cleared when 1 is written.

Pre-amplifier Gain Control Register (ADC_PGCTL)

Register	Offset	R/W	Description	Reset Value
ADC_PGCTL	ADC_BA+0x3C	R/W	ADC Pre-amplifier Gain Control Register	0x0001_0000

31	30	29	28	27	26	25	24
Reserved			**PAG_I**				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
APPS	**MICE**	**Reserved**					**OS[8]**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
OS[7:0]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**PAG_II**		**OPMUTE**

Bits	Description	
[31:29]	**Reserved**	Reserved.
[28:24]	**PAG_I**	**Gain Setting Bits for the First Stage Of Pre-Amp** 00000 = 20 dB, 00001 = 21 dB, 00010 = 22 dB, 00011 = 23 dB, : 10100 = 40 dB, Others = Equivalent with “00000”.
[23]	**APPS**	**ADC and Pre-Amplifier Power Source Selection** 0 = Internal regulator is disabled, and AVDD is selected as power source. 1 = Internal regulator is enabled, and regulator output is selected as power source. The input of internal regulator is from AVDD pin, output of regulator is 0.85 time the voltage of AVDD.
[22]	**MICE**	**MIC_BIAS Output Enable** 0 = MIC_BIAS output is disabled (tri-state). 1 = MIC_BIAS output is enabled, its output is 0.85 time the voltage of AVDD.
[21:17]	**Reserved**	Reserved.
[16:8]	**OS**	**Configuration for Pre-Amp OP Offset Bias Compensation Voltage** There are 512 levels and 0.25mV per level @ 5V condition. The compensation is available only when FWU of ADC_MIBSCTL register is 2’b11.
[7:3]	**Reserved**	Reserved.

[2:1]	**PAG_II**	**Gain Setting Bits For The Second Stage Of Pre-Amp** 00 = 0 dB, 01 = 10 dB, 10 = 20 dB, 11 = 30 dB.
[0]	**OPMUTE**	**Mute Control of First Stage Pre-Amp for Offset Bias Calibration** When this bit set is as “1”, two input end of first stage pre-amp will be shorted, and feedback resistor of this stage will be shorted. 0 = open 1 = short

MIC Bias and PGC Control Register (ADC_MIBSCTL)

Register	Offset	R/W	Description	Reset Value
ADC_MIBSCTL	ADC_BA+0x40	R/W	MIC bias and PGC Control Register	0x0000_0033

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

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:4]	**CTRS**	**AVDD/2 Accelerating** 00 = R~10K 01 = R~40K 10 = R~200K 11 = R~600K
[3:2]	**Reserved**	Reserved.
[1:0]	**FWU**	**Fast Wake Up For RC** 00 = 8K (Twu<100ms). set 00 at initial, then 11 after 80ms 01 = 40K 10 = 200K 11 = 400K (Normal path)

6 FLASH MEMORY CONTROLLER (FMC)

6.1 Overview

N572F072/P072 equips with 72K bytes on chip embedded flash for application program memory (APROM) that can be updated thru ISP (In System Programming) procedure. ISP function could enable user to update program memory when chip is soldered on PCB.

6.2 Features

- AHB interface compatible
- Run up to 25MHz with zero wait state for discontinuous address read access
- 72K bytes APROM
- In System Programming (ISP) to update on chip Flash EPROM

6.3 Flash Memory Organization

The N572F072/P072 flash memory consists of 72K bytes flash and user configuration.

User configuration block provides several bytes to control system logic, like Flash security lock, voltage detector voltage level, 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 power on. User can set these bits according to application request.

Table 6-1 Memory Address Map

Block Name	Size	Start Address	End Address
Flash	72KB	0x0000_0000	0x0001_1FFF
User Configuration	1 word	0x0030_0000	0x0030_0003

6.4 User Configuration

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
CONFIGURE Base Address: **CFG_BA=0x0030_0000**				
CONFIG	CFG_BA+0x000	R/W	User Configuration Memory	0x0000_0000

User Configuration Memory (CONFIG)

Register	Offset	R/W	Description	Reset Value
CONFIG	CFG_BA+0x000	R/W	User Configuration Memory	0x0000_0000

31	30	29	28	27	26	25	24
CWDTEN	-	-	-	-	-	-	-
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CVDEN	-	**CVDTV**	-	-	-	-	-
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
-	-	-	-	-	-	-	-
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CSPI0_CT		**CSPI0_FT**		-	**PRTB**	**LOCK**	-

Bits	Description	
[31]	**CWDTEN**	**Watchdog Enable** 0 = Watchdog is disabled after power on. 1 = Watchdog is enabled after power on.
[30:24]	**Reserved**	Reserved.
[23]	**CVDEN**	**Voltage Detector Enable** 0 = Enable Voltage Detector after power on. 1 = Disable Voltage Detector after power on.
[22]	**Reserved**	Reserved.
[21]	**CVDTV**	**Voltage Detector Threshold Voltage Selection** CVDTV: Threshold voltage 0: 2.7V 1: 3.0V
[20:8]	**Reserved**	Reserved.

[7:6]	**CSPI0_CT**	**Coarse Timing Control for SPI0 Data Receiving After Power On** These bits are used to adjust receiving clock for latching serial-in data correctly in high speed transmission mode. 11 = Receiving data clock is delayed 1 half SPI0_CLK clock cycle ,. 10 = Receiving data clock is delayed 3 half SPI0_CLK clock cycle,. 01 = Receiving data clock is delayed 2 half SPI0_CLK clock cycle,. 00 = Receiving data clock is same as the SPI0_CLK.
[5:4]	**CSPI0_FT**	**Fine Timing Control for SPI0 Data Receiving After Power On** The delay timing selected by CSPI0_CT can be further tuned finely by CSPI0_FT. 11 = Receiving data clock has no extra delay,. 10 = Receiving data clock has extra 2.5nS delay,. 01 = Receiving data clock has extra 5.0nS delay,. 00 = Receiving data clock has extra 7.5nS delay. **Note:** The extra delay is implemented by delay chains. The accuracy of delay time would base on process deviation.
[3]	**Reserved**	Reserved.
[2]	**PRTB**	**Protection On 8K Bytes Flash** This bit is effective only for the part of 72KB flash. 0 = ISP function only can operate on the 64KB Flash (address: 0x0000~0xFFFF) and CONFIG, and cannot operate on the 8KB Flash (address: 0x10000~0x11FFF). 1 = ISP function can operate on the whole 72KB Flash (address: 0x00000~0x11FFF) and CONFIG.
[1]	**LOCK**	**Security Lock** 0 = Flash data are locked. 1 = Flash data are not locked. When flash data is locked, (1) only device ID, CONFIG can be read by Writer and ICP thru serial debug interface. Other data are locked as 0xFFFFFFFF. (2) ISP can read data in legal address regardless of LOCK bit value. (3) SWD interface cannot access internal RAM and Flash
[0]	**Reserved**	Reserved.

6.5 In-System Programming (ISP)

The program and data flash memory support both in hardware In-Circuit Programming (ICP) and firmware based In-System programming (ISP). Hardware ICP programming mode uses the Serial-Wire Debug (SWD) port to program chip. Dedicated ICE Debug hardware (Nu-Link) or ICP gang-writers are available to reduce programming and manufacturing costs. For firmware updates in the field, the N572F072/P072 provides an ISP mode allowing a device to be reprogrammed under software control.

ISP is performed without removing the device from the system. The most common method to perform ISP is via SPI Flash along with the firmware stored in APROM. The approach example is shown below.

> The ISP F/W is stored in APROM initially. CPU copies the ISP F/W to internal RAM and then jumps from APROM to ISP F/W entrance in RAM to execute ISP procedure. The data to be programmed to APROM come from external SPI Flash.

6.5.1 ISP Procedure

To enable ISP functionality software must first ensure the ISP clock (CLK_AHBCLK.ISPCKEN) is present then set the FMC_ISPCTL.ISPEN bit. Software is required to write register SYS_REGLCTL (address 0x5000_0100) with 0x59, 0x16 and 0x88 sequentially before writing FMC_ISPCTL register. This procedure is used to protect flash memory from being destroyed owning to unintended write during power on/off duration.

Several error conditions are checked after software writes the FMC_ISPTRG[0](ISPGO) bit. If an error condition occurs, ISP operation is not been started and the ISP fail flag (FMC_ISPCTL.ISPFF) will be set instead. The ISPFF flag will remain set until it is cleared by software. Subsequent ISP procedure can be started even if ISPFF is set. It is recommended that software checks ISPFF bit and clear it after each ISP operation if it is set to “1”.

When FMC_ISPTRG[0](ISPGO) bit is set, the CoretxM0 CPU will wait for ISP operation to finish, during this period; peripherals operate as usual. If any interrupt request occurs, CPU will not service it until ISP operation finishes.

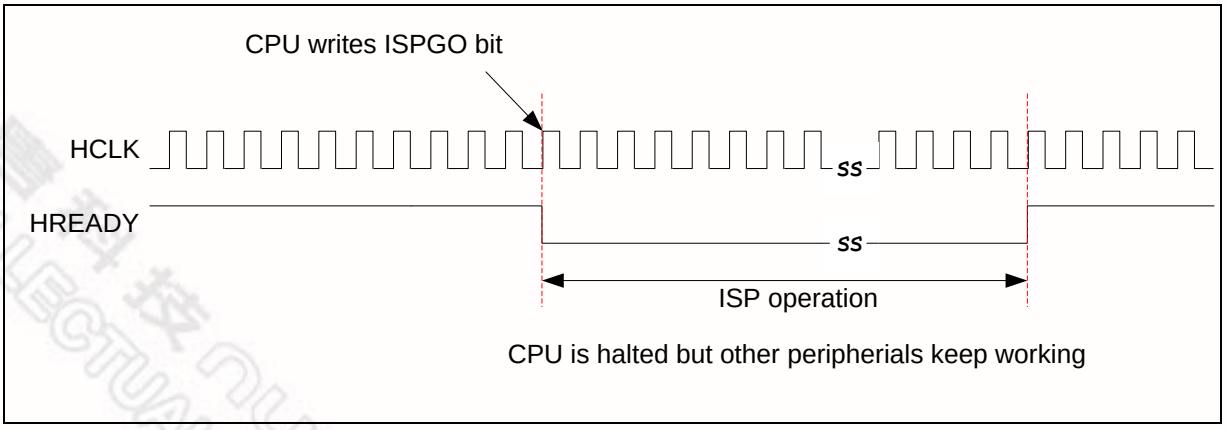


Figure 6-1 ISP Operation Timing

Table 6-2 ISP Command Set

ISP Mode	FMC_ISPCMD	FMC_ISPADDR			FMC_ISPDAT
	CMD[5:0]	A21	A20	A[19:0]	D[31:0]
Standby	0x3X	x	x	x	x
Read Company ID	0x0B	x	x	x	Returns “0x0000_00DA“
Read Device ID	0x0C	x	x	0x00000	Data output “Device ID”
FLASH Page Erase	0x22	0	A[20]	A[19:0]	x
FLASH Program	0x21	0	A[20]	A[19:0]	Data input
FLASH Read	0x00	0	A[20]	A[19:0]	Data output
CONFIG Page Erase	0x22	1	1	A[19:0]	x
CONFIG Program	0x21	1	1	A[19:0]	Data input
CONFIG Read	0x00	1	1	A[19:0]	Data output

6.6 Flash Control Register Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
FMC Base Address: **FMC_BA=0x5000_C000**				
FMC_ISPCTL	FMC_BA+0x00	R/W	ISP Control Register	0x0000_0000
FMC_ISPADDR	FMC_BA+0x04	R/W	ISP Address Register	0x0000_0000
FMC_ISPDAT	FMC_BA+0x08	R/W	ISP Data Register	0x0000_0000
FMC_ISPCMD	FMC_BA+0x0C	R/W	ISP Command Register	0x0000_0000
FMC_ISPTRG	FMC_BA+0x10	R/W	ISP Trigger Control Register	0x0000_0000

6.7 Flash Control Register Description

ISP Control Register (FMC_ISPCTL)

This register is a protected register. To program this register needs an unlock sequence, writing “59h”, “16h”, “88h” sequentially to SYS_REGLCTL, to gain access. Refer to Protected Register Lock Key Register (SYS_REGLCTL).

Register	Offset	R/W	Description	Reset Value
FMC_ISPCTL	FMC_BA+0x00	R/W	ISP Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**ET**			**Reserved**	**PT**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**ISPFF**	**Reserved**	**CFGUEN**	**Reserved**		**EWEN**	**ISPEN**

Bits	Description	
[31:15]	**Reserved**	Reserved.
[14:12]	**ET**	**Flash Page Erase Time** 000 = 2.0 ms (default). 001 = 2.5 ms. 010 = 3.0 ms. 011 = 1.5 ms. 100 = 1.0 ms. 101 = 0.75 ms. 110 = 4.0 ms. 111 = 5.0 ms.
[11]	**Reserved**	Reserved.
[10:8]	**PT**	**Flash Program Time** 000 = 20 μs (default). 001 = 25 μs. 010 = 30 μs. 011 = 35 μs. 100 = 40 μs. 101 = 45 μs. 110 = 10 μs. 111 = 15 μs.

[7]	**Reserved**	Reserved.
[6]	**ISPFF**	**ISP Fail Flag** This bit is set by hardware when a triggered ISP meets any of the following conditions: (1) MCU writes (or erase) to Flash when EWEN is “0”. (2) Destination address is illegal, such as over an available range. (3) Destination address is within 0x10000~0x11FFF if PRTB bit of CONFIG is “0”. (4) Destination address is within 0x00000~0x0FFFF if MOTPB bit of MAP0 is “0”. (5) Destination address is within 0x10000~0x11FFF if both MOTPB and SOTPB bits of MAP0 are “0”. Write 1 to clear.
[5]	**Reserved**	Reserved.
[4]	**CFGUEN**	**CONFIG Update Enable** 0 = Disable. 1 = Enable. When enabled, ISP functions can access the CONFIG address space and modify device configuration area.
[3:2]	**Reserved**	Reserved.
[1]	**EWEN**	**Enable Erase/Write Of ISP Function** 0 = Disable erase/write. 1 = Enable erase/write.
[0]	**ISPEN**	**ISP Enable** 0 = Disable ISP function. 1 = Enable ISP function.

ISP Address Register (FMC_ISPADDR)

Register	Offset	R/W	Description	Reset Value
FMC_ISPADDR	FMC_BA+0x04	R/W	ISP Address Register	0x0000_0000

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

Bits	Description	
[31:0]	**ISPADDR**	**ISP Address Register** This is the memory address register that a subsequent ISP command will access. ISP operation are carried out on 32-bit word only, consequently ISPADDR [1:0] must be 00b for correct ISP operation. N572F072/P072 equips with an 18Kx32 bits embedded flash.

ISP Data Register (FMC_ISPDAT)

Register	Offset	R/W	Description	Reset Value
FMC_ISPDAT	FMC_BA+0x08	R/W	ISP Data Register	0x0000_0000

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

Bits	Description	
[31:0]	**ISPDAT**	**ISP Data Register** Write data to this register before an ISP program operation. Read data from this register after an ISP read operation

ISP Command (FMC_ISPCMD)

Register	Offset	R/W	Description	Reset Value
FMC_ISPCMD	FMC_BA+0x0C	R/W	ISP Command Register	0x0000_0000

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

Bits	Description	
[31:6]	**Reserved**	Reserved.
[5:0]	**CMD**	**ISP Command** Operation Mode : CMD ---------- ------- Standby : 0x3X Read CID : 0x0B Read DID : 0x0C Flash Page Erase : 0x22 Flash Program : 0x21 Flash Read : 0x00

ISP Trigger Control Register (FMC_ISPTRG)

Register	Offset	R/W	Description	Reset Value
FMC_ISPTRG	FMC_BA+0x10	R/W	ISP Trigger Control Register	0x0000_0000

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

Bits	Description	
[31:1]	**Reserved**	Reserved.
[0]	**ISPGO**	**ISP Start Trigger** Write 1 to start ISP operation. This bit will be cleared to 0 by hardware automatically when ISP operation is finished. 0 = ISP operation is finished. 1 = ISP is ongoing.

7 FUNCTION REGISTER COMPARISON

Register	N572F072/P072	N572F065
APU base	APU_BA=0x5000_8000	APU_BA=0x400C_0000
CLK_PWRCTL	• Remove bit 0. • Default value of CLK_PWRCTL [3] is “1”.	Default value of CLK_PWRCTL[3] is “0”.
CLK_AHBCLK	APUCKEN is moved to APUCKEN=CLK_AHBCLK[3].	APUCKEN=CLK_APBCLK[30]
SYS_GPA_HS	New added register	No
CLK_CLKSEL0	Add SysTick clock source selection, STCLKSEL = CLK_CLKSEL0[4:3].	
CLK_CLKSEL1	• Modify clock source selection because 12MHz XTAL is removed, 24MHz RC is changed to 48MHz RC. • Add SPI0 clock source selection. SPI0SEL = CLK_CLKSEL1[5:4].	
SYST_CSR **SYST_RVR** **SYST_CVR**	New added SysTick registers	No
CLK_PLLCON	Modify PLL2 clock source and output frequency	
SPI_CTL (SPI1 only)	Add master/slave mode selection. SLAVE=SPI_CTL[18].	
SPI_ SSCTL (SPI1 only)	Add slave mode related bits. LTRIG_FLAG= SPI_SSCTL[5], SS_LTRIG=SPI_SSCTL[4].	
SPI0_RCLK	New added register	No
APU_CH0DAT0~ **APU_CH0DAT7** **APU_CH1DAT0**	16-bit registers	13-bit registers
ADCALR	No	Yes
ADC_PGCTL	• Definitions of control bits are changed. OS=ADC_PGCTL[16:8], PAG_I=ADC_PGCTL[28:24], PAG_II=ADC_PGCTL[2:1]. • New added APPS, MICE bits.	OS=ADC_PGCTL[10:6], PAG_I=ADC_PGCTL[12:11], PAG_II=ADC_PGCTL[5:1].
APU_VM	VOLUM=111: reserved	VOLUM=111: reserved and prohibited
ADC_MIBSCTL	New added register	No
FATCON	No	Yes

8 REVISION HISTORY

VERSION	DATE	PAGE/ CHAP.	DESCRIPTION
V1.00	Aug, 2014	-	Initial release.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.