九齊科技股份有限公司

Nyquest Technology Co., Ltd.

USER MANUAL

NX12F / NX13F (EF Series)

32-bit Audio SoC for SBC / MIDI Playback & Voice Changer / Recording / Recognition

Version 1.0

Feb. 26, 2024

NYQUEST TECHNOLOGY CO. reserves the right to change this document without prior notice. Information provided by NYQUEST is believed to be accurate and reliable. However, NYQUEST makes no warranty for any errors which may appear in this document. Contact NYQUEST to obtain the latest version of device specifications before placing your orders. No responsibility is assumed by NYQUEST for any infringement of patent or other rights of third parties which may result from its use. In addition, NYQUEST products are not authorized for use as critical components in life support devices/systems or aviation devices/systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of NYQUEST.

Revision History

Version	Date	Description	Modified Page
1.0	2024/2/26	Formal release.	-

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 General Description

The NX12F / NX13F is a 32-bit MCU based high-quality speech/MIDI SoC, which is specially designed for various audio-related DSP applications. Its predecessor, the NX1 OTP series, has been well accepted in the market for numerous applications like long-duration speech/MIDI synthesis, voice changer, and stand-alone voice recognition. The NX1 Embedded Flash (EF) series follows the same architecture but uses EF to replace the OTP memory core and embeds OCD (On-Chip Debugger) at production chips to make the project development much easier than ever. As compared with mask ROM counterparts, the NX12F / NX13F got absolute advantage over MOQ and lead time.

The NX12F / NX13F got memory-mapped architecture, which can address up to 16MB space that includes memory (EF / RAM), register files, peripheral and SPI Flash (that supports Bank for expansion and Instruction / Data modes). The highly compressed SBC (Sub-Band Coding) is achieved with greatly enhanced quality & much less memory size compared against traditional ADPCM coding due to the incorporation of efficient DSP algorithms as well as the upgrade of H/W spec. By incorporating Instruction / Data local buses, 1-cycle multiplier and 7-cycle divider for the 32-bit N705-S core, it reaches outstanding performance of 50+ DMIPS when the system clock runs at 48MHz. The S/W-based MIDI synthesizer can reach more than 16-ch polyphonic channels, with all data including SBC / MIDI files, wavetable timbres, XIP (eXecutable In Place) program code and general user data stored in the embedded Flash and/or external SPI Flash memory.

The dual clock design with built-in 48MHz I_HRC and 32KHz I_LRC gives users the freedom to switch among Normal / Standby / Halt modes for the balance of power consumption and performance or to keep the time under low power condition. The NX12F / NX13F series got single-chip packages as well as MCP (Multi-Chip Package) that integrates SPI Flash inside the low-pin count packages. Set aside the 2Mb EF inside the NX12F / NX13F, users can select different SPI Flash density of MCP parts or choose external SPI Flash to expand the memory easily.

There are various useful features inside the NX12F / NX13F: Four sets of 16-bit Timers; up to 8-ch H/W PWM-IO pins; 8-channel, 12-bit SAR ADC that supports differential MIC input with 2-stage of pre-amplifiers & AGC / PGA, and various kinds of sensors; 14-bit DAC to connect with external power amplifier or built-in Σ-Δ PWM amplifier to drive speaker directly; independently configurable GPIO per pin with alternate functions; IR TX that supports 38KHz / 57KHz / 125KHz / 500KHz carrier for Infrared or QFID applications; SPI0 with embedded LDO to connect to external SPI Flash; SPI1 for connecting with 2.4GHz RF module or other SPI devices; I2C H/W interface; UART for ISP/IAP (In System/Application Programming), serial communication, or debugging via Hyper-Terminal; NFC tag for ISP/IAP to exchange data with iDevices.

The NX12F / NX13F supports C language programming that provides customers with more controllability over complicated projects. Besides, it also brings Q-Code (High-level script programming) to 32-bit MCU that provides customers with an easy-to-use, highly productive development environment for quick realization of product concepts. The NX12F / NX13F is embedded with On-Chip Debugger, which can provide ICE

functionality on the actual production chip. Moreover, it allows for ICP (In-Circuit Programming) to change code / content right on the target board and ISP/IAP (In-System/Application Programming) that makes possible on-board re-programming and code/content updates in the system.

Various package forms are available for the NX12F / NX13F: Dice, SOP-16, SSOP-28, LQFP-32, and MCP with SPI Flash to fit in diversified application needs.

1.2 Features

- Wide Operating Voltage: 2.0V ~ 5.5V
  - SPI Flash is powered by embedded 3.3V LDO.
  - Min. operating voltage is 2.0V for max. CPU clock @ 48MHz.
- 32-bit CPU core
  - Andes N705-S, like ARM Cortex-M0+.
  - Max. CPU clock @ 48MHz, up to 50+ DMIPS w/ 1 wait-state @ EF access.
  - 1-cycle fast multiplier, 7-cycle fast divider.
  - Instruction Local Memory & Data Local Memory employed.
- There are 12 single-die and MCP (Multi-Chip Package) parts for the NX12F / NX13F series.

P/N	VDD	RAM	EF	SPI Flash	Duration (Sec)		I/O	SPI 0	SPI 1	I^2C	UART	16-bit Timer	PWM-IO	12-bit SAR ADC	MIC	Cap Touch	DAC	Power Amp.	VR	Package
					16Kbps	24Kbps														
NX12FS51A	2.0V~5.5V	10KB	2Mb	-	98	66	25 10 22 25	v - v v	v v v v	v - v v	v	4	8 3 8 8	8 1 6 8	v	13 5 7 13	14-bit	Σ−Δ PWM	v	Dice SOP-16 SSOP-28 LQFP-32
NX12FS61A	2.0V~5.5V	12KB	2Mb	-	98	66	25 10 22 25	v - v v	v v v v	v - v v	v	4	8 3 8 8	8 1 6 8	v	13 5 7 13	14-bit	Σ−Δ PWM	v	Dice SOP-16 SSOP-28 LQFP-32
NX13FS51A	2.0V~5.5V	10KB	2Mb	-	98	66	25 10 22 25	v - v v	v v v v	v - v v	v	4	8 3 8 8	8 1 6 8	v	13 5 7 13	14-bit	Σ−Δ PWM	-	Dice SOP-16 SSOP-28 LQFP-32
NX13FS61A	2.0V~5.5V	12KB	2Mb	-	98	66	25 10 22 25	v - v v	v v v v	v - v v	v	4	8 3 8 8	8 1 6 8	v	13 5 7 13	14-bit	Σ−Δ PWM	-	Dice SOP-16 SSOP-28 LQFP-32
NX12FM52A	2.0V~5.5V	10KB	2Mb	4Mb	360	240	10 21	- -	v v	- v	v	4	3 4	1 8	v	5 11	14-bit	Σ−Δ PWM	v	SOP-16 SSOP-28
NX12FM54A	2.0V~5.5V	10KB	2Mb	16Mb	1,147	765	10 21	- -	v v	- v	v	4	3 4	1 8	v	5 11	14-bit	Σ−Δ PWM	v	SOP-16 SSOP-28
NX12FM62A	2.0V~5.5V	12KB	2Mb	4Mb	360	240	10 21	- -	v v	- v	v	4	3 4	1 8	v	5 11	14-bit	Σ−Δ PWM	v	SOP-16 SSOP-28
NX12FM64A	2.0V~5.5V	12KB	2Mb	16Mb	1,147	765	10 21	- -	v v	- v	v	4	3 4	1 8	v	5 11	14-bit	Σ−Δ PWM	v	SOP-16 SSOP-28
NX13FM52A	2.0V~5.5V	10KB	2Mb	4Mb	360	240	10 21	- -	v v	- v	v	4	3 4	1 8	v	5 11	14-bit	Σ−Δ PWM	-	SOP-16 SSOP-28
NX13FM54A	2.0V~5.5V	10KB	2Mb	16Mb	1,147	765	10 21	- -	v v	- v	v	4	3 4	1 8	v	5 11	14-bit	Σ−Δ PWM	-	SOP-16 SSOP-28
NX13FM62A	2.0V~5.5V	12KB	2Mb	4Mb	360	240	10 21	- -	v v	- v	v	4	3 4	1 8	v	5 11	14-bit	Σ−Δ PWM	-	SOP-16 SSOP-28
NX13FM64A	2.0V~5.5V	12KB	2Mb	16Mb	1,147	765	10 21	- -	v v	- v	v	4	3 4	1 8	v	5 11	14-bit	Σ−Δ PWM	-	SOP-16 SSOP-28

Table 1 Product Line-Up of NX12F / NX13F

- Built-in 2Mb Embedded Flash
  - Endurance @ 100K times of erase/programming cycles
  - Storage for program code, voice recording, and preset content
- Built-in 10KB / 12KB RAM for NX12F_13Fx5xA / NX12F_13Fx6xA
  - 2KB instruction RAM optional for program execution during EF erasure period
- Built-in oscillators for I_HRC (48MHz) and I_LRC (32,768Hz), accuracy trimmed to +/-1.0% for I_HRC and +/-3.0% for I_LRC.
- Power management to support 3 operating modes per system requirement: Normal / Standby / Halt modes. At Halt mode, the typical current consumption is 2uA ~ 4.5uA @ 5.5V.
- 8-level LVD (Low Voltage Detection): 3.6V, 3.4V, 3.2V, 3.0V, 2.8V, 2.4V, 2.2V, 2.0V. Reference voltage

3.4V trimmed to +/-3% accuracy before shipping.

- Built-in LVR (Low Voltage Reset) function.
- Four Timers (Timer0 / Timer1 / Timer2 / Timer3), each Timer consists of a 16-bit down-counter with various clock sources.
- Built-in 14-bit resolution DAC output for driving speaker via external power amplifier
- Built-in 13-bit Σ-Δ PWM power amplifier to drive speaker directly
- Up to 25 pins of GPIO
  - Bit configurable for every I/O pin by register control, except byte-defined strength of pull-up resistors
  - Multi-function pins via register control
- SPI Flash interface supported @ SPI0 with embedded 3.3V LDO
  - Master mode
  - Up to 24MHz clock speed
  - 32-bit mode supported to address beyond 128Mb
  - Support Data mode and XIP mode (eXecute In Place)
  - Support Single / Dual / Quad I/O modes of SPI Flash
- SPI master @ SPI1 supported with embedded 3.3V LDO shared with SPI0
  - Up to 24MHz clock speed
- IR TX supported (via any Timer, 0 ~ 3)
- RTC with optional 16.384KHz / 1.024KHz / 64Hz / 2Hz interrupts
- WDT (Watch-Dog Timer) supported with optional 188ms / 750ms reset periods
- Built-in NFC Tag function
  - ISO-14443A compliant NFC operation while powered to communicate in both directions with iDevices
- ISP / IAP (In-System/Application Programming) supported
  - Source: UART, NFC Tag
  - Destination: EF, SPI Flash (via program control)
  - ISP / IAP code protected from accidental EF erasure
- UART H/W supported
  - Full-duplex TX/RX with 4-level FIFO
  - Baud rates @ 115,200 / 230,400 / 1M bps
- I2C H/W supported
  - Master / Slave mode
  - 100KHz / 400KHz / 1MHz Clock
  - 4-byte FIFO
- Up to two PWM-IO Generators
  - Each generator with 4 PWM-IO pins (PWMA0 ~ PWMA3 / PWMB0 ~ PWMB3)
  - Each generator got one 16-bit timer, which could be used as a general timer
  - Independent 16-bit duty cycle register per PWM-IO pin

- CapTouch sensor supported
  - All pins of PA port are optional as CapTouch sensing pads
- ADC (Analog Digital Conversion)
  - 8-ch (with auto scan mode), 12-bit resolution SAR ADC
  - Trigger by underflow of Timer0 / 1 / 2, or by software
  - Ch0 with 4-level FIFO shared with MIC input for voice processing
  - Embedded with dedicated analog LDO: optional output at 3.3V @ 3-battery, or 2.3V @ 2-Battery.
- Built-in MIC control circuitry
  - Differential input MIC with 2-stage of pre-amplifiers and optional AGC or PGA for gain control
- Support EF Security Lock mechanism to prevent programmed data from being read out
- Easy-to-use Development Environment
  - High-level *Q-Code* programming supported
  - Advanced C-Module programming supported
  - 2-pin OCD (On-Chip Debugger) supported with source-level debugging for both Q-Code and C-Module
  - Multi-purpose NX_Programmer for ICE debugging and programming of EF / SPI Flash
  - Smart Writer used in MP for EF/SPI Flash pre-programming @ packages, or ICP @ target boards
  - 6-pin programming interface (VDD, VSS, PD1, PD0, PA12, RSTB/PA8) supported
- S/W-based Speech/MIDI Codec, Voice Recognition, & various algorithms supported
  - ADPCM Codec (Adaptive Differential PCM): 4 channels, 4-bit / 5-bit per sample
  - SBC Codec (Sub-Band Coding): 4.5K ~ 32Kbps, max. 16KHz bandwidth, up to 1-ch record & 2-ch playback
  - MIDI: up to 16-channel polyphonic melody @ 32KHz Output Sample Rate
  - Voice Effects: speed / pitch change, robotic sound, echo, real-time voice changer, etc.
  - Voice Recognition: phoneme-based for stand-alone VR applications (only for NX12F)
- Noise filter @ 4x/8x/16x/32x up-sampling supported
- Shipping Form
  - Dice
  - Package
    - Single-Die: SOP-16 / SSOP-28 / LQFP-32
    - MCP (with 4Mb / 16Mb SPI Flash): SOP-16 / SSOP-28

1.3 Block Diagram

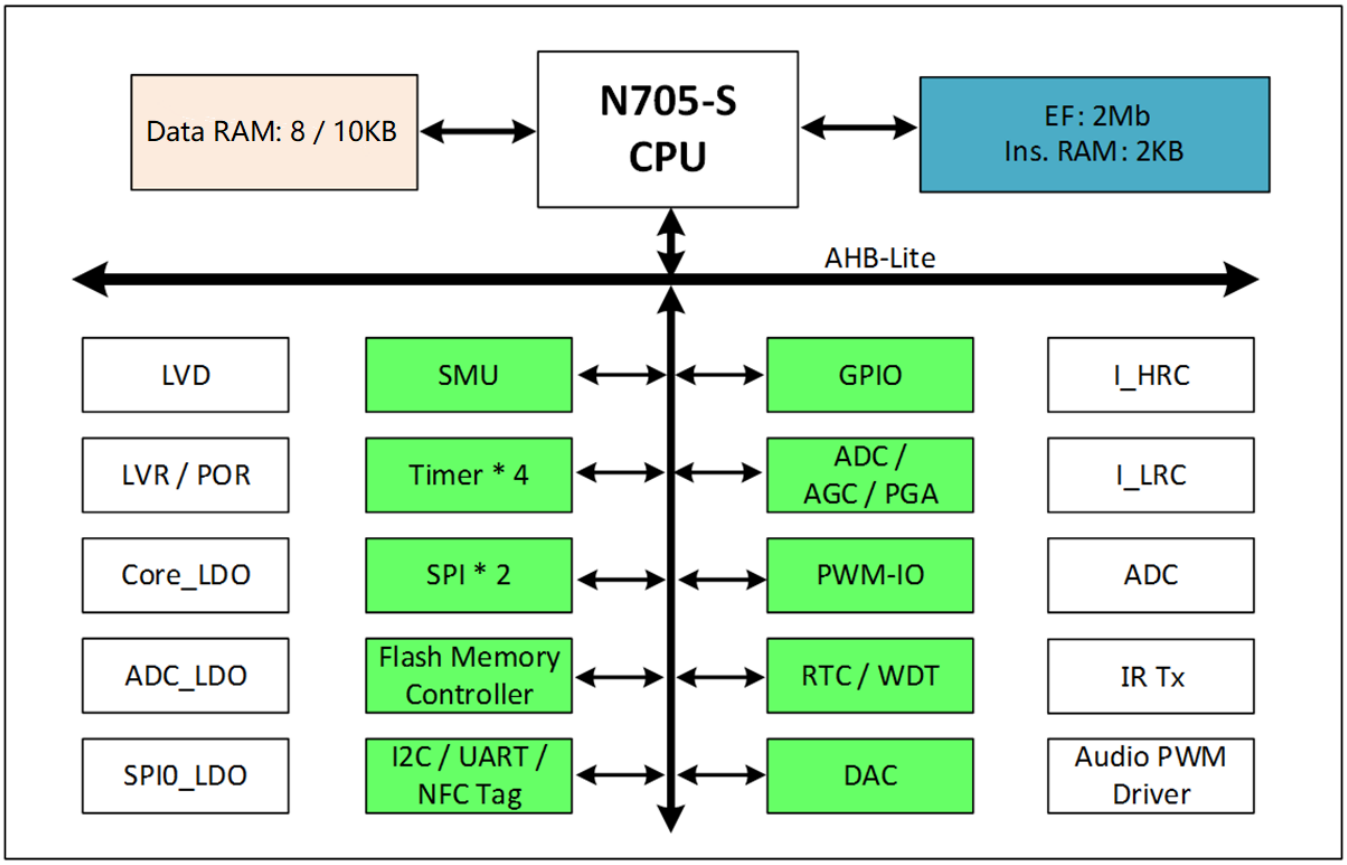


Figure 1 Block Diagram of NX12F / NX13F Series

1.4 Pad / Pin Description

Pad Name	Alt #	Type	SOP-16	SSOP-24	SSOP-24	LQFP-32	Dice	Pad Description
Part No.	-	-	NX12Fx5xAS16 NX13Fx5xAS16	NX12FS51AU24 NX13FS51AU24	NX12FM5xAU24 NX13FM5xAU24	NX12FS51AQ32 NX13FS51AQ32	NX12FS51A NX13FS51A	-
Power								
VDD	0	P	●	●	●	●	●	Power input
ADC_VDD	0	AP	●	●	●	●	●	Analog power output for connecting with 0.1uF cap.
VDD_PWM	0	P				●	●	Power for DAC and audio PWM driver.
SPI0_VDD	0	P	●	●	●	●	●	Power output for SPI Flash. Connect with 0.1uF cap.
VSS	0	P	●	●	●	●	●	Ground
VSS_PWM	0	P					●	Ground for DAC and audio PWM driver
VSS_ADC	0	AP					●	Ground for analog circuits
PWM / DAC								
PWM1 / DAC	0	O	●	●	●	●	●	Audio PWM driver output 1 or DAC output by option
PWM2	0	O	●	●	●	●	●	Audio PWM PA output 2
Port A								
PA0	0	I/O				●	●	GPIO pin PA0
AIN0	3	AI				●	●	Analog input 0
PA1	0	I/O	●	●	●	●	●	GPIO pin PA1
IR0	1	O	●	●	●	●	●	IR Tx output from Timer0
AIN1	3	AI	●	●	●	●	●	Analog input 1
PA2	0	I/O			●	●	●	GPIO pin PA2
IR1	1	O			●	●	●	IR Tx output from Timer1
INT0	2	I			●	●	●	External INT0
AIN2	3	AI			●	●	●	Analog input 2
PA3	0	I/O			●	●	●	GPIO pin PA3
TM0	1	I			●	●	●	External input for Timer0 / Timer1
INT1	2	I			●	●	●	External INT1
AIN3	3	AI			●	●	●	Analog input 3
PA4	0	I/O		●	●	●	●	GPIO pin PA4
SDA	1	I/O		●	●	●	●	SDA pin of I2C
AIN4	3	AI		●	●	●	●	Analog input 4
PA5	0	I/O		●	●	●	●	GPIO pin PA5
SCL	1	O		●	●	●	●	SCL pin of I2C
AIN5	3	AI		●	●	●	●	Analog input 5
PA6	0	I/O			●	●	●	GPIO pin PA6
TX	1	O			●	●	●	TX pin of UART
AIN6	3	AI			●	●	●	Analog input 6
PA7	0	I/O			●	●	●	GPIO pin PA7
RX	1	I			●	●	●	RX pin of UART

Pad Name	Alt #	Type	SOP-16	SSOP-24	SSOP-24	LQFP-32	Dice	Pad Description
Part No.	-	-	NX12Fx5xAS16 NX13Fx5xAS16	NX12FS51AU24 NX13FS51AU24	NX12FM5xAU24 NX13FM5xAU24	NX12FS51AQ32 NX13FS51AQ32	NX12FS51A NX13FS51A	-
AIN7	3	AI			●	●	●	Analog input 7
RSTB/PA8	0	I, I/O	●	●	●	●	●	RSTB (default) or GPIO pin by option. Since this pin is RSTB by default upon power up, so that it can't be set at low level when powered on.
CSB	3	I	●	●	●	●	●	Programming I/F
PA12	0	I/O	●	●	●	●	●	GPIO pin PA12
PWMA0	1	O	●	●	●	●	●	PWMA0 output pin
SPI1_CSB	2	O	●	●	●	●	●	CSB pin of SPI1
MISO	3	O	●	●	●	●	●	Programming I/F
PA13	0	I/O				●	●	GPIO pin PA13
PWMA1	1	O				●	●	PWMA1 output pin
SPI1_CLK	2	O				●	●	CLK pin of SPI1
PA14	0	I/O	●	●	●	●	●	GPIO pin PA14
PWMA2	1	O	●	●	●	●	●	PWMA2 output pin
SPI1_MOSI	2	O	●	●	●	●	●	MOSI pin of SPI1
PA15	0	I/O	●	●	●	●	●	GPIO pin PA15
PWMA3	1	O	●	●	●	●	●	PWMA3 output pin
SPI1_MISO	2	I	●	●	●	●	●	MISO pin of SPI1
Port B								
PB0	0	I/O		●		●	●	GPIO pin PB0
PWMB0	1	O		●		●	●	PWMB0 output pin
SPI0_IO1	2	O		●		●	●	IO1 / MISO pin of SPI0
PB1	0	I/O		●		●	●	GPIO pin PB1
PWMB1	1	O		●		●	●	PWMB1 output pin
SPI0_CSB	2	O		●		●	●	CSB pin of SPI0
PB2	0	I/O		●		●	●	GPIO pin PB2
PWMB2	1	O		●		●	●	PWMB2 output pin
SPI0_CLK	2	O		●		●	●	CLK pin of SPI0
PB3	0	I/O		●		●	●	GPIO pin PB3
PWMB3	1	O		●		●	●	PWMB3 output pin
SPI0_IO0	2	I/O		●		●	●	IO0 / MOSI pin of SPI0
PB4	0	I/O		●	●	●	●	GPIO pin PB4
NFC_LA	1	I		●	●	●	●	NFC Antenna terminal A
SPI0_IO2	2	I/O		●	●	●	●	IO2 pin of SPI0
TM1	3	I		●	●	●	●	External input for Timer2 / Timer3
PB5	0	I/O		●	●	●	●	GPIO pin PB5
NFC_LB	1	O		●	●	●	●	NFC Antenna terminal B
SPI0_IO3	2	I/O		●	●	●	●	IO3 pin of SPI0

Pad Name	Alt #	Type	SOP-16	SSOP-24	SSOP-24	LQFP-32	Dice	Pad Description
Part No.	-	-	NX12Fx5xAS16 NX13Fx5xAS16	NX12FS51AU24 NX13FS51AU24	NX12FM5xAU24 NX13FM5xAU24	NX12FS51AQ32 NX13FS51AQ32	NX12FS51A NX13FS51A	-
Port C								
PC0	0	I/O	●	●	●	●	●	GPIO pin PC0
IR2	1	O	●	●	●	●	●	IR Tx output from Timer2
MICP	3	AI	●	●	●	●	●	MIC+
PC1	0	I/O	●	●	●	●	●	GPIO pin PC1
IR3	1	O	●	●	●	●	●	IR Tx output from Timer3
MICN	3	AI	●	●	●	●	●	MIC-
PC2	0	I/O	●	●	●	●	●	GPIO pin PC2
VMIC	3	AO	●	●	●	●	●	MIC bias voltage output
PC3	0	I/O				●	●	GPIO pin PC1
OPO	3	AO				●	●	Output pin of 2nd stage Pre-Amp
Port D								
PD0	0	I/O	●	●	●	●	●	GPIO pin PI0
ICE_CLK	1	O	●	●	●	●	●	Clock pin of ICE port
TX	2	O	●	●	●	●	●	TX pin of UART
SCL	3	I	●	●	●	●	●	Programming I/F
PD1	0	I/O	●	●	●	●	●	GPIO pin PI1
ICE_DAT	1	I/O	●	●	●	●	●	Data pin of ICE port
RX	2	I	●	●	●	●	●	RX pin of UART
SDA	3	I/O	●	●	●	●	●	Programming I/F

Pad Type: P = Digital Power, I = Input, O = Output, I/O = Input / Output, AI = Analog Input, AO = Analog output, AP = Analog Power.

Chapter 2. Hardware Architecture

2.1 N705-S Core

The N705-S processor is a 2-stage, 32-bit RISC processor. It has an AMBA AHB-Lite interface and includes an Internal Vectored Interrupt Control (IVIC) component. It is built with a fast multiplier and 4-level programmable-priority interrupt control.

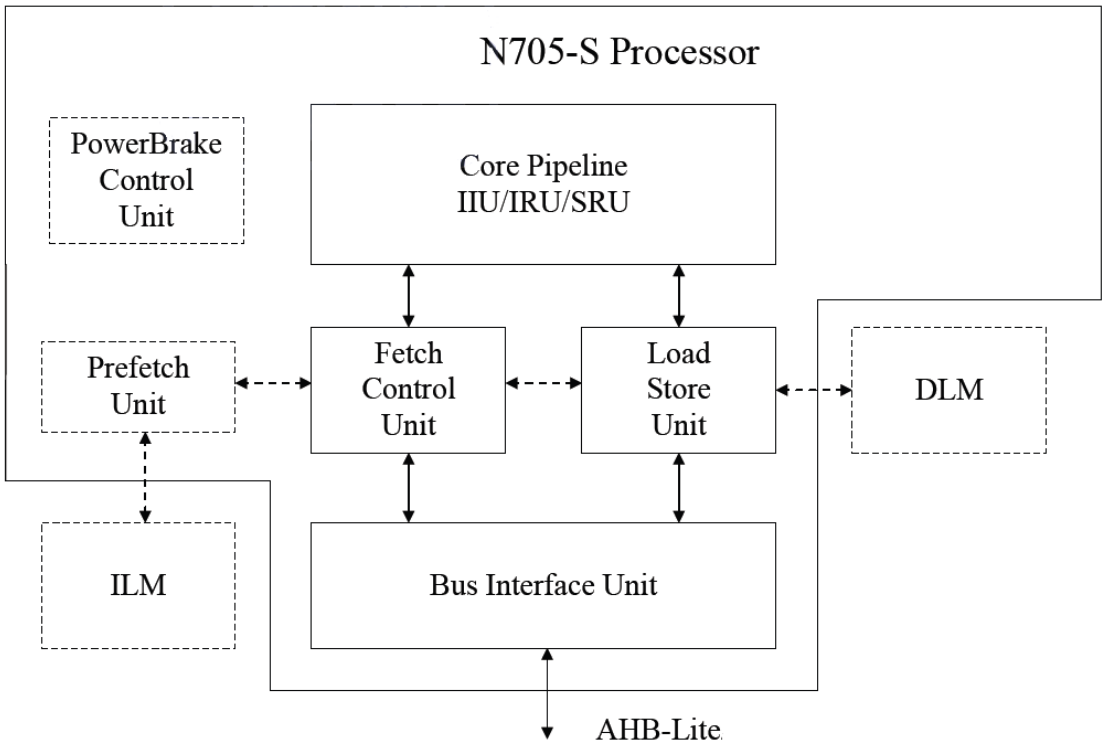


Figure 2 Block diagram of N705-S Core

The implemented device provides:

- 2-stage pipeline design
- 2R1W register file
- 16/32-bit mixed instructions with 16 GPRs (General-Purpose Registers)
- Hardware 1-cycle multiplier (32x32=32)
- Addressable space: 16MB (24-bit address)
- Interrupt vectored interrupt control: 4-level programmable-priority

2.2 Memory Map

The NX12F_NX13F series provides a 16M-byte address space for programmers. The memory allocation of all peripheral blocks is shown in Table 2. The detailed register, memory addressing and programming will be

described in the following sections for individual on-chip modules. The NX12F_NX13F series only supports little-endian data format.

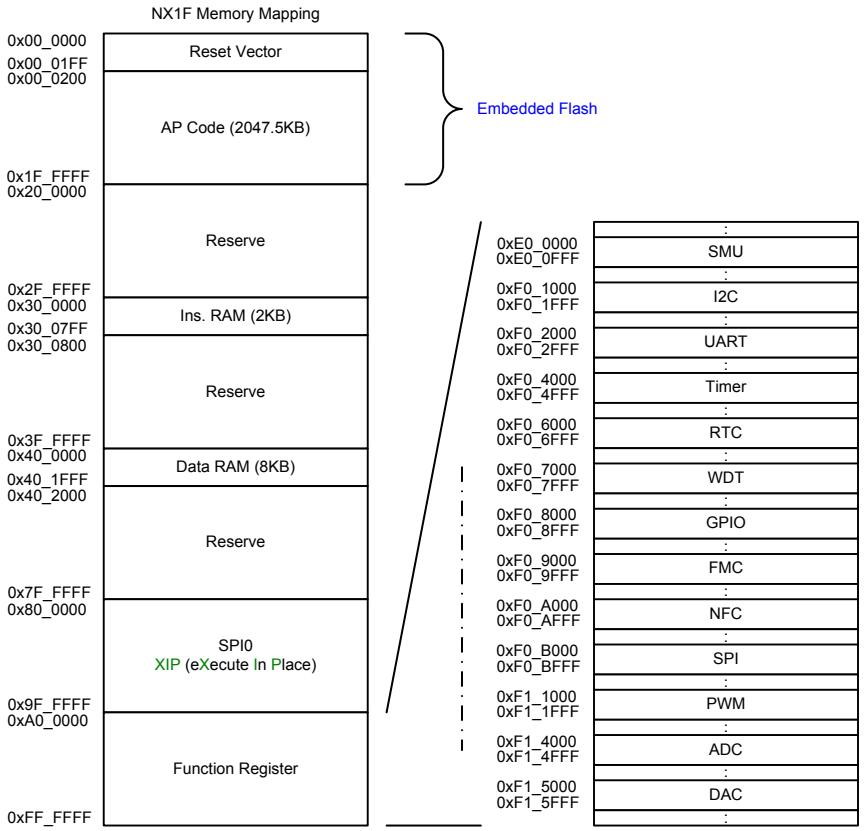


Table 1 Memory Map

2.3 Clock Generator

The clock generator consists of 2 clock sources:

- Built-in high clock (I_HRC): Output frequency can be 48MHz, or 40MHz by option.
- Built-in low clock (I_LRC): Output frequency is 32,768Hz.

The SYS_CLK's source is from I_HRC. The CPUCLK's source is SYS_CLK divided by 2^n (n = 0 ~ 7).

About the clock source for peripherals, please refer to the individual sections for further information.

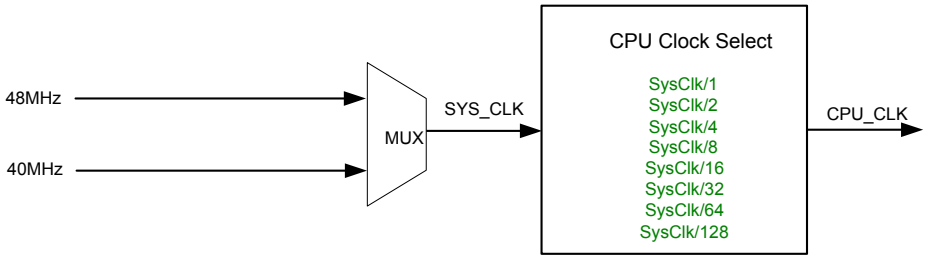


Figure 3 Clock Generator Block Diagram

2.4 System Reset

The system reset consists of 3 sources: Power-On Reset and Low Voltage Reset, External Reset pin (RSTb) and Watch-Dog Timer Reset.

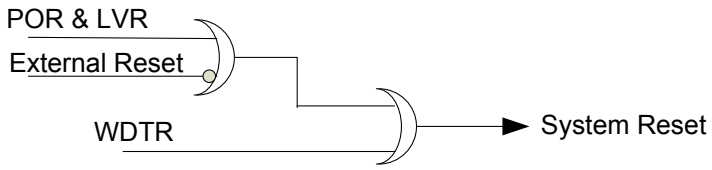


Figure 4 System Reset Block Diagram

2.4.1 POR & LVR (Power-On Reset & Low Voltage Reset)

Power on Reset (POR): When the NX1 is first powered up, the POR signal will reset the chip for a period then release it.

Low Voltage Reset (LVR): When the NX1 is running and supply power is lower than a specified voltage, the LVR signal is generated to reset the whole chip.

2.4.2 External Reset

The PA8 / Resetb pin is multiple function pin, which is an external reset pin, either as a GPIO pin. When the pin is selected to as Resetb pin, which is in low state, the whole chip enters reset state. When the Resetb changes from low to high, the NX12F_NX13F will release the reset state after two LO_CLK period.

2.4.3 WDTR (Watch-Dog Timer Reset)

There is an on-chip free-running oscillator and counter in the NX12F_NX13F series, which is used by WDT. The NX12F_NX13F provides two kinds of time settings for the WDT: 188ms or 750ms. When CPU executes STANDBY instruction, it will be cleared to zero until CPU released from standby state. Detailed information please refer to 2.17Watchdog Timer (WDT).

2.5 Operating Mode

The NX12F_NX13F series provides 3 kinds of operating modes to tailor for various kinds of applications while saving power consumption. These operating modes are normal mode, standby mode and halt mode.

Normal mode is designated for high-speed, high-performance operation, while standby mode, the NX12F_NX13F series will stop almost all operations, except peripheral blocks with clock source from LO_CLK, to wake-up periodically. At halt mode, the NX12F_NX13F series will stop all operations, waiting for external events to wake it up.

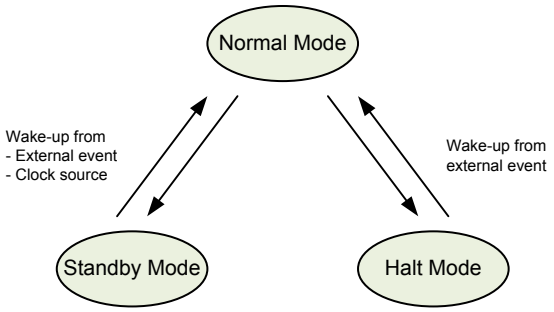


Figure 5 Three Operating Modes

2.5.1 Normal Mode

After any Reset Event occurred and chip released from reset state, the NX12F_NX13F will start executing program under normal mode. At normal mode, SYS_CLK's clock source comes from HI_CLK. It provides the highest performance, which in turn leads to the largest power consumption among four operating modes.

2.5.3 Standby Mode

The NX12F_NX13F will enter standby mode by setting P_SMU_PWR_Ctrl[1:0] (MODE) to 3 and executing “STANDBY 2” instruction. Some peripheral blocks are still active with clock source from LO_CLK. It can wake-up the NX12F_NX13F when certain peripheral blocks generate interrupt events.

2.5.4 Halt Mode

The NX12F_NX13F will enter halt mode by setting P_SMU_PWR_Ctrl[1:0] (MODE) to 1 and executing “STANDBY 2” instruction. At halt mode, HI_CLK and LO_CLK are automatically stopped. All peripheral blocks are disabled; instruction execution is stopped and only GPIO can wake-up the NX12F_NX13F. Therefore, halt mode is the most power-saving mode.

2.6 Interrupt

Interrupt signals are directly connected to the N705-S processor. The assignment of interrupt numbers is shown in Table 3. The interrupt priority is controlled by the processor. Each interrupt is assigned with 2 bits to represent 4 possible priority levels ranging from 0 (highest) to 3 (lowest). The hardware compares the priority level first: the smaller the priority level, the higher the priority. With the same priority, the lower the interrupt number, the higher the priority.

Interrupt Number	Interrupt Vector	System and Peripheral Interrupt	Interrupt Number	Interrupt Vector	System and Peripheral Interrupt
0	0	Reset/NMI	17	68	SPI0_INT
1	4	TLB fill	18	72	EXT1_INT
2	8	PTE not present	19	76	RTC_INT
3	12	TLB misc.	20	80	PWMTMR0_INT
4	16	TLB VLPT miss	21	84	PWMTMR1_INT
5	20	Machine error	22	88	CAP_INT
6	24	Debug related	23	92	TMR3_INT
7	28	General exception	24	96	NFC_INT
8	32	Syscall	25	100	Reserved
9	36	DAC0_INT	26	104	I2C_INT
10	40	DAC1_INT	27	108	UART_INT
11	44	ADC_INT	28	112	SPI1_INT
12	48	FMC_INT	29	116	Reserved
13	52	TMR0_INT	30	120	SW_INT
14	56	TMR1_INT	31	124	SWA_INT
15	60	TMR2_INT	32	128	Reserved
16	64	EXT0_INT			

Table 2 Interrupt Vector

2.7 System Management Register

2.7.1 Overview and Features

The following functions are included:

- Clock frequency and power consumption control
- Low voltage detection setting
- Operating mode setting
- Software interrupt
- Software reset control and reset flag register
- Peripheral block function enable control and function reset control

2.7.2 Register Map

Register	Offset	R/W	Description	Initial Value
SMU Base Address: SMU_BA = 0xE0_0000				
P_SMU_CLK_Ctrl	SMU_BA+0x00	R/W	Clock Control Register	0x0000_00E3
P_SMU_PWR_Ctrl	SMU_BA+0x04	R/W	Power Control Register	0x0000_0001
P_SMU_SW_INT	SMU_BA+0x10	R/W	Software Interrupt Control Register	0x0000_0000
SMU_INTV_BA	SMU_BA+0x18	R/W	Interrupt vector base address	0x0000_0000
P_SMU_RST_Flag	SMU_BA+0x20	R/W1C	Reset Flag Register	0x0000_0008
P_SMU_RST_SW_Ctrl	SMU_BA+0x24	W	Reset Control Register	0x0000_0000
P_SMU_FUNC_Ctrl	SMU_BA+0x30	R/W	Function Control Register	0x0000_0000
P_SMU_FUNC_RST	SMU_BA+0x34	W	Function Reset Register	0x0000_0000

2.7.3 Register Description

R: Read only, **W**: Write only, **W1C**: Write 1 to Clear, **R/W**: Read and Write

- **P_SMU_CLK_Ctrl (Clock Control Register)**

Register	Offset		Description	Initial Value
P_SMU_CLK_Ctrl	SMU_BA+0x00		Clock Control Register	0x0000_00E3
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:8]	Reserved			0x0
Bit[7:5]	CPUCLKDIV	R/W	111: SYS_CLK/1 110: SYS_CLK/2 101: SYS_CLK/4 100: SYS_CLK/8 011: SYS_CLK/16 010: SYS_CLK/32 001: SYS_CLK/64 000: SYS_CLK/128	0x7
Bit[4]	Reserved			0x0
Bit[3:2]	Reserved			
Bit[1]	Reserved			0x1
Bit[0]	Reserved			0x1

- **P_SMU_PWR_Ctrl (Power Control Register)**

Register	Offset		Description	Initial Value
P_SMU_PWR_Ctrl	SMU_BA+0x04		Power Control Register	0x0000_0001
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[14:12]	LVD_SEL	R/W	111: 3.6V 110: 3.4V 101: 3.2V 100: 3.0V 011: 2.8V 010: 2.4V 001: 2.2V 000: 2.0V	0x0
Bit[11:10]	Reserved			0x0
Bit[9]	LVD_FLAG	R	1: detect low voltage 0: no detect low voltage	0x0

Bit	Name	R/W	Descriptions	Initial Value
Bit[8]	LVD_EN	R/W	1: LVD enable 0: LVD disable	0x0
Bit[7]	Reserved			0x0
Bit[6]	Reserved			0x0
Bit[5]	Reserved			0x0
Bit[4]	LDOSPI0_EN	R/W	1: LDOSPI0 IP enabled 0: LDOSPI0 IP disabled	0x0
Bit[3]	Reserved			0x0
Bit[2]	ALDO_EN	R/W	1: ALDO IP enabled 0: ALDO IP disabled	0x0
Bit[1:0]	MODE	R/W	11: Standby mode 10: Reserved 01: Halt mode 00: Only CPU stops	0x1

- **P_SMU_SW_INT (Software Interrupt Control Register)**

Register	Offset		Description	Initial Value
P_SMU_SW_INT	SMU_BA+0x10		Software Interrupt Control Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	Protection Key	W		0x0
Bit[15:1]	Reserved			0x0
Bit[0]	SWA_INT	W/R	1: Software interrupt request, when Protection Key = 5AC3 0: Do nothing	0x0

- **SMU_INTV_BA (Interrupt Vector Base Address)**

Register	Offset		Description	Initial Value
SMU_INTV_BA	SMU_BA+0x18		Interrupt Vector Base Address	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31]	BA_EN	R/W	1: Enable Base Address 0: Disable Base Address	0x0
Bit[30:0]	Reserved			0x0

- **P_SMU_RST_Flag (Reset Flag Register)**

Register	Offset		Description	Initial Value
P_SMU_RST_Flag	SMU_BA+0x20		Reset Flag Register	0x0000_0008
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:6]	Reserved			0x0
Bit[5]	RSTF_ROMFAIL	R/W1C	1: Flag. reset by dl_opn rom fail 0: No reset	0x0
Bit[4]	RSTF_WDT	R/W1C	1: Flag. reset CHIP by WDT 0: No reset	0x0

Bit	Name	R/W	Descriptions	Initial Value
Bit[3]	RSTF_POR	R/W1C	1: Flag. reset CHIP by POR / LVR 0: No reset	0x1
Bit[2]	RSTF_ERST	R/W1C	1: Flag to indicate CHIP reset by hardware. (External Reset) 0: No reset	0x0
Bit[1]	RSTF_CHIP	R/W1C	1: Flag to indicate CHIP reset by software. (write register CHIP_RST) 0: No reset	0x0
Bit[0]	RSTF_CPU	R/W1C	1: Flag to indicate CPU reset by software. (write register CPU_RST) 0: No reset	0x0

- **P_SMU_RST_SW_Ctrl (Reset Control Register)**

Register	Offset		Description	Initial Value
P_SMU_RST_SW_Ctrl	SMU_BA+0x24		Reset Control Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	Protection Key	W		0x0
Bit[15:2]	Reserved			0x0
Bit[1]	CHIP_RST	W	When Protection Key=5AC3 & CHIP_RST=1, Whole CHIP reset	0x0
Bit[0]	CPU_RST	W	When Protection Key=5AC3 & CPU_RST=1, CPU reset (AMBA reset)	0x0

- **P_SMU_FUNC_Ctrl (Function Control Register)**

Register	Offset		Description	Initial Value
P_SMU_FUNC_Ctrl	SMU_BA+0x30		Function Control Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:14]	Reserved			0x0
Bit[13]	NFC_EN	R/W	1: IP block's clock enable 0: IP block's clock disabled (default)	0x0
Bit[12]	Reserved			0x0
Bit[11]	FMCCLK_EN	R/W	1: IP block's clock enable 0: IP block's clock disabled (default)	0x0
Bit[10]	Reserved			0x0
Bit[9]	URCLK_EN	R/W	1: IP block's clock enable 0: IP block's clock disabled (default)	0x0
Bit[8]	Reserved			0x0
Bit[7]	DACCLK_EN	R/W	1: IP block's clock enable 0: IP block's clock disabled (default)	0x0
Bit[6]	ADCCLK_EN	R/W	1: IP block's clock enable 0: IP block's clock disabled (default)	0x0
Bit[5]	IICCLK_EN	R/W	1: IP block's clock enable 0: IP block's clock disabled (default)	0x0
Bit[4]	RTCCLK_EN	R/W	1: IP block's clock enable 0: IP block's clock disabled (default)	0x0
Bit[3]	SPICLK_EN	R/W	1: IP block's clock enable 0: IP block's clock disabled (default)	0x0

Bit	Name	R/W	Descriptions	Initial Value
Bit[2]	PWMCLK_EN	R/W	1: IP block's clock enable 0: IP block's clock disabled (default)	0x0
Bit[1]	TMRCLK_EN	R/W	1: IP block's clock enable 0: IP block's clock disabled (default)	0x0
Bit[0]	GPIOCLK_EN	R/W	1: IP block's clock enable 0: IP block's clock disabled (default)	0x0

- **P_SMU_FUNC_RST (Function Reset Register)**

Register	Offset		Description	Initial Value
P_SMU_FUNC_RST	SMU_BA+0x34		Function Reset Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:14]	Reserved			0x0
Bit[13]	NFCBK_RST	W	1: Gen Reset pulse 0: No Reset	0x0
Bit[12]	Reserved			0x0
Bit[11]	FMCBK_RST	W	1: Gen Reset pulse 0: No Reset	0x0
Bit[10]	Reserved			0x0
Bit[9]	URBK_RST	W	1: Gen Reset pulse 0: No Reset	0x0
Bit[8]	Reserved			0x0
Bit[7]	DACBK_RST	W	1: Gen Reset pulse 0: No Reset	0x0
Bit[6]	ADCBK_RST	W	1: Gen Reset pulse 0: No Reset	0x0
Bit[5]	IICBK_RST	W	1: Gen Reset pulse 0: No Reset	0x0
Bit[4]	Reserved			0x0
Bit[3]	SPIBK_RST	W	1: Gen Reset pulse 0: No Reset	0x0
Bit[2]	PWMBK_RST	W	1: Gen Reset pulse 0: No Reset	0x0
Bit[1]	TMRBK_RST	W	1: Gen Reset pulse 0: No Reset	0x0
Bit[0]	GPIOBK_RST	W	1: Gen Reset pulse 0: No Reset	0x0

2.8 IO Ports

2.8.1 Overview and Features

- Max. 25 Individually programmable input/output pins, default as input at reset
- Support multiple functions

2.8.2 Block Diagram

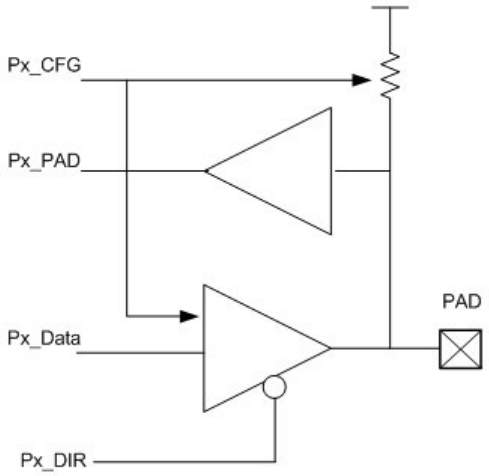


Figure 6 I/O Pad Structure

2.8.3 Function Description

Up to 25 pins are available. These are shared multiple function pins under control of the alternate multiple function registers. A total of 25 pins are arranged in 4 ports named with Port A (PA), Port B (PB), Port C (PC), and Port D (PD). The PA port has 13 pins, the PB port has 6 pins, the PC port has 4 pins, and PD port has 2 pins.

Each pin can be configured as input or output, weak / strong pull-high resistor and can generate interrupt signal to CPU upon wakeup. PA2 and PA3 also has external interrupt function.

2.8.4 Register Map

Register	Offset	R/W	Description	Initial Value
GPIO Base Address : GPIO_BA = 0xF0_8000				
P_PA_DIR	GPIO_BA+0x00	R/W	Port A input output control	0x0000_FFFF
P_PA_Data	GPIO_BA+0x04	R/W	Port A data output register	0x0000_FFFF
P_PA_PAD	GPIO_BA+0x08	R/W	Port A pad input register	0x0000_uuuu
P_PA_Wakeup_Mask	GPIO_BA+0x0C	R/W	Port A Config register0	0xFFFF_0000
P_PA_CFG	GPIO_BA+0x10	R/W	Port A Config register1	0xFFFF_FFFF
P_PA_MFP	GPIO_BA+0x14	R/W	Port A multi function register	0x0000_0000
P_PB_DIR	GPIO_BA+0x20	R/W	Port B input output control	0x0000_FFFF
P_PB_Data	GPIO_BA+0x24	R/W	Port B data output register	0x0000_FFFF
P_PB_PAD	GPIO_BA+0x28	R/W	Port B pad input register	0x0000_uuuu
P_PB_Wakeup_Mask	GPIO_BA+0x2C	R/W	Port B Config register0	0xFFFF_0000
P_PB_CFG	GPIO_BA+0x30	R/W	Port B Config register1	0xFFFF_FFFF
P_PB_MFP	GPIO_BA+0x34	R/W	Port B multi function register	0x0000_0000

Register	Offset	R/W	Description	Initial Value
P_PC_DIR	GPIO_BA+0x40	R/W	Port C input output control	0x0000_000F
P_PC_Data	GPIO_BA+0x44	R/W	Port C data output register	0x0000_000F
P_PC_PAD	GPIO_BA+0x48	R/W	Port C pad input register	0x0000_000u
P_PC_Wakeup_Mask	GPIO_BA+0x4C	R/W	Port C Config register0	0x000F_0000
P_PC_CFG	GPIO_BA+0x50	R/W	Port C Config register1	0x0000_000F
P_PC_MFP	GPIO_BA+0x54	R/W	Port C multi function register	0x0000_0000
P_EXT_INT_Trig	GPIO_BA+0x60	R/W	EXT INT TRIG Select	0x0000_0000
P_EXT_INT_Ctrl	GPIO_BA+0x64	R/W	EXT INT Interrupt Enable	0x0000_0000
P_EXT_INT_Flag	GPIO_BA+0x68	R/W1C	EXT INT Interrupt Flag	0x0000_0000
P_PULL_HIGH_CFG	GPIO_BA+0x70	R/W	EXT INT Interrupt Flag	0x0000_0000
P_PD_DIR	GPIO_BA+0x80	R/W	Port D input output control	0x0000_000F
P_PD_Data	GPIO_BA+0x84	R/W	PortD data output register	0x0000_000F
P_PD_PAD	GPIO_BA+0x88	R/W	PortD pad input register	0x0000_000u
P_PD_Wakeup_Mask	GPIO_BA+0x8C	R/W	Port D Config register0	0x000F_0000
P_PD_CFG	GPIO_BA+0x90	R/W	Port DConfig register1	0x000F_000F
P_PD_MFP	GPIO_BA+0x94	R/W	Port D multi function register	0x0000_0005

2.8.5 Register Description

R: Read only, **W**: Write only, **W1C**: Write 1 to clear, **R/W**: Read and Write

- **P_PA_DIR (Port A Input Output Control)**

Register	Offset		Description	Initial Value
P_PA_DIR	GPIO_BA+0x00		Port A input output control	0x0000_FFFF
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	Reserved			0x0
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[15:12]	PA_DIR	R/W	PA15 ~ PA12 I/O Direction 1: input 0: output	0xF
Bit[11:9]	Reserved			0x7
Bit[8:0]	PA_DIR	R/W	PA8 ~ PA0 I/O Direction 1: input 0: output	0x1FF

- **P_PA_Data (Port A Data Output Register)**

Register	Offset		Description	Initial Value
P_PA_Data	GPIO_BA+0x04		Port A data output register	0x0000_FFFF
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	Reserved			0x0
Bit[15:12]	PA_DATA	R/W	PA12 ~ PA15 Data output 1: data output 1 0: data output 0	0xF

Bit[11:9]	Reserved			0x7
Bit[8:0]	PA_DATA	R/W	PA8 ~ PA0 Data output 1: data output 1 0: data output 0	0x1FF

- **P_PA_PAD (Port A Pad Input Register)**

Register	Offset		Description	Initial Value
P_PA_PAD	GPIO_BA+0x08		Port A pad input register	0x0000_uuuu
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	Reserved			0x0
Bit[15:12]	PA_PAD	R/W	PA15 ~ PA12 Read pad data	0xu
Bit[11:9]	Reserved			0x0
Bit[8:0]	PA_PAD	R/W	PA8 ~PA0 Read pad data	0xuuu

- **P_PA_Wakeup_Mask (Port A Config Register0)**

Register	Offset		Description	Initial Value
P_PA_Wakeup_Mask	GPIO_BA+0x0C		Port A Config register0	0xFFFF_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:28]	PA_CCS1	R/W	PA15 ~ PA12 Sink current select 1: Normal Sink 0: Large Sink	0xF
Bit[27:25]	Reserved			0x7
Bit[24:16]	PA_CCS0	R/W	PA8 ~ PA0 Sink current select 1: Normal Sink 0: Large Sink	0x1FF
Bit[15:12]	PA_MASK	R/W	PA15 ~ PA12 Wakeup Function Mask 1: Have wakeup function 0: No Wake function	0x0
Bit[11:9]	Reserved			0x0
Bit[8:0]	PA_MASK	R/W	PA8 ~ PA0 Wakeup Function Mask 1: Have wakeup function 0: No Wake function	0x0

- **P_PA_CFG (Port A Config Register1)**

Register	Offset		Description	Initial Value
P_PA_CFG	GPIO_BA+0x10		Port A Config register1	0xFFFF_FFFF
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:24]	Reserved			0xFF
Bit[23:16]	Reserved			0xF
Bit[15:12]	PA_PUENB	R/W	PA15 ~ PA12 Pull-up Resistor select 1: Pull-up Resistor disabled 0: Pull-up Resistor enabled	0xF
Bit[11:9]	Reserved			0x7
Bit[8:0]	PA_PUENB	R/W	PA8 ~ PA0 Pull-up Resistor select 1: Pull-up Resistor disabled 0: Pull-up Resistor enabled	0x1FF

- **P_PA_MFP (Port A Multi-Function Register)**

Register	Offset		Description	Initial Value
P_PA_MFP	GPIO_BA+0x14		Port A multi function register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:30]	PA15_MFP	R/W	11 : reserved 10 : spi1_mosi 01 : pwma3 00 : gpio	0x0
Bit[29:28]	PA14_MFP	R/W	11 : reserved 10 : reserved 01 : pwma2 00 : gpio	0x0
Bit[27:26]	PA13_MFP	R/W	11 : reserved 10 : reserved 01 : pwma1 00 : gpio	0x0
Bit[25:24]	PA12_MFP	R/W	11 : reserved 10 : spi1_csb, spi1_sclk, spi1_mosi 01 : pwma0 00 : gpio	0x0
Bit[23:22]	Reserved			0x0
Bit[21:20]	Reserved			0x0
Bit[19:18]	Reserved			0x0
Bit[17:16]	PA8_MFP	R/W	11 : reserved 10 : reserved 01 : reserved 00 : gpio	0x0
Bit[15:14]	PA7_MFP	R/W	11 : adc ch7 10 : reserved 01 : rx 00 : gpio	0x0
Bit[13:12]	PA6_MFP	R/W	11 : adc ch6 10 : reserved 01 : tx 00 : gpio	0x0
Bit[11:10]	PA5_MFP	R/W	11 : adc ch5 10 : reserved 01 : scl 00 : gpio	0x0
Bit[9:8]	PA4_MFP	R/W	11 : adc ch4 10 : reserved 01 : sda 00 : gpio	0x0
Bit[7:6]	PA3_MFP	R/W	11 : adc ch3 10 : exint1 01 : tm0 00 : gpio	0x0

Bit[5:4]	PA2_MFP	R/W	11 : adc ch2 10 : exint0 01 : ir1 00 : gpio	0x0
Bit[3:2]	PA1_MFP	R/W	11 : adc ch1 10 : reserved 01 : ir0 00 : gpio	0x0
Bit[1:0]	PA0_MFP	R/W	11 : adc ch0 10 : reserved 01 : reserved 00 : gpio	0x0

- **P_PB_DIR (Port B Input Output Control)**

Register	Offset		Description	Initial Value
P_PB_DIR	GPIO_BA+0x20		Port B input output control	0x0000_FFFF
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	Reserved			0x0
Bit[15:6]	Reserved			0x3FF
Bit[5:0]	PB_DIR	R/W	PB5 ~ PB0 I/O Direction 1: Input 0: Output	0x3F

- **P_PB_Data (Port B Data Output Register)**

Register	Offset		Description	Initial Value
P_PB_Data	GPIO_BA+0x24		Port B data output register	0x0000_FFFF
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	Reserved			0x0
Bit[15:6]	Reserved			0x3FF
Bit[5:0]	PB_DATA	R/W	PB5 ~ PB0 Data output 1: Data output 1 0: Data output 0	0x3F

- **P_PB_PAD (Port B Pad Input Register)**

Register	Offset		Description	Initial Value
P_PB_PAD	GPIO_BA+0x28		Port B pad input register	0x0000_uuuu
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	Reserved			0x0
Bit[15:6]	Reserved			0x0
Bit[5:0]	PB_PAD	R/W	PB5 ~ PB0 Read pad data	0xuu

- **P_PB_Wakeup_Mask (Port B Config Register0)**

Register	Offset		Description	Initial Value
P_PB_Wakeup_Mask	GPIO_BA+0x2C		Port B Config register0	0xFFFF_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:22]	Reserved			0x3FF

Bit[21:16]	PB_CCS0	R/W	PB5 ~ PB0 Sink current select 1: Normal Sink 0: Large Sink	0x3F
Bit[15:6]	Reserved			0x0
Bit[5:0]	PB_MASK	R/W	PB5 ~ PB0 Wakeup Function Mask 1: Have wakeup function 0: No Wake function	0x0

- **P_PB_CFG (Port B Config Register1)**

Register	Offset		Description	Initial Value
P_PB_CFG	GPIO_BA+0x30		Port B Config register1	0xFFFF_FFFF
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:24]	Reserved			0xFF
Bit[23:6]	Reserved			0x3FFF
Bit[5:0]	PB_PUENB	R/W	PB5 ~ PB0 Pull-up Resistor select 1: Pull-up Resistor disabled 0: Pull-up Resistor enabled	0x3F

- **P_PB_MFP (Port B Multi-Function Register)**

Register	Offset		Description	Initial Value
P_PB_MFP	GPIO_BA+0x34		Port B multi function register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:30]	Reserved			0x0
Bit[29:28]	Reserved			0x0
Bit[27:26]	Reserved			0x0
Bit[25:24]	Reserved			0x0
Bit[23:22]	Reserved			0x0
Bit[21:20]	Reserved			0x0
Bit[19:18]	Reserved			0x0
Bit[17:16]	Reserved			0x0
Bit[15:14]	Reserved			0x0
Bit[13:12]	Reserved			0x0
Bit[11:10]	PB5_MFP	R/W	11 : reserved 10 : reserved 01 : nfc_sdo 00 : gpio	0x0
Bit[9:8]	PB4_MFP	R/W	11 : tm1 10 : spi0_sio2, spi0_io3 01 : nfc_sdi 00 : gpio	0x0
Bit[7:6]	PB3_MFP	R/W	11 : reserved 10 : reserved 01 : pwmb3 00 : gpio	0x0
Bit[5:4]	PB2_MFP	R/W	11 : reserved 10 : reserved 01 : pwmb2 00 : gpio	0x0

Bit[3:2]	PB1_MFP	R/W	11 : reserved 10 : spi0_csb(pb1),spi0_clk(pb2), spi0_mosi(pb3) 01 : pwmb1 00 : gpio	0x0
Bit[1:0]	PB0_MFP	R/W	11 : reserved 10 : spi0_miso(pb0) 01 : pwmb0 00 : gpio	0x0

- **P_PC_DIR (Port C Input Output Control)**

Register	Offset		Description	Initial Value
P_PC_DIR	GPIO_BA+0x40		Port C input output control	0x0000_000F
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:4]	Reserved			0x0
Bit[3:0]	PC_DIR	R/W	I/O Direction 1: Input 0: Output	0xF

- **P_PC_Data (Port C Data Output Register)**

Register	Offset		Description	Initial Value
P_PC_Data	GPIO_BA+0x44		Port C data output register	0x0000_000F
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:4]	Reserved			0x0
Bit[3:0]	PC_DATA	R/W	Data output 1: Data output 1 0: Data output 0	0xF

- **P_PC_PAD (Port C Pad Input Register)**

Register	Offset		Description	Initial Value
P_PC_PAD	GPIO_BA+0x48		Port C pad input register	0x0000_000u
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:4]	Reserved			0x0
Bit[3:0]	PC_PAD	R/W	Read pad data	0xu

- **P_PC_Wakeup_Mask (Port C Config Register0)**

Register	Offset		Description	Initial Value
P_PC_Wakeup_Mask	GPIO_BA+0x4C		Port C Config register0	0x000F_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:20]	Reserved			0x0
Bit[19:16]	PC_CCS0	R/W	PC3 ~ PC0 Sink current select 1: Normal Sink 0: Large Sink	0xF
Bit[15:4]	Reserved			0x0
Bit[3:0]	PC_MASK	R/W	PC3 ~ PC0 Wakeup Function Mask 1: Have wakeup function 0: No Wake function	0x0

- **P_PC_CFG (Port C Config Register1)**

Register	Offset		Description	Initial Value
P_PC_CFG	GPIO_BA+0x50		Port C Config register1	0x00FF_000F
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:24]	Reserved			0x0
Bit[23:16]	Reserved			0xFF
Bit[15:4]	Reserved			0x0
Bit[3:0]	PC_PUENB	R/W	PC3 ~ PC0 Pull-up Resistor select 1: Pull-up Resistor disabled 0: Pull-up Resistor enabled	0xF

- **P_PC_MFP (Port C Multi-Function Register)**

Register	Offset		Description	Initial Value
P_PC_MFP	GPIO_BA+0x54		Port C multi-function register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:8]	Reserved			0x0
Bit[7:6]	PC3_MFP	R/W	11: opo 10: reserved 01: reserved 00: gpio	0x0
Bit[5:4]	PC2_MFP	R/W	11: vmic 10: reserved 01: reserved 00: gpio	0x0
Bit[3:2]	PC1_MFP	R/W	11: micn 10: reserved 01: ir3 00: gpio	0x0
Bit[1:0]	PC0_MFP	R/W	11: micp 10: reserved 01: ir2 00: gpio	0x0

- **P_EXT_INT_Trig (EXT INT TRIG Select)**

Register	Offset		Description	Initial Value
P_EXT_INT_Trig	GPIO_BA+0x60		EXT INT TRIG Select	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:4]	Reserved			0x0
Bit[3:2]	TRIG_EXT1	R/W	1x: rising edge & falling edge 01: Falling edge 00: Rising edge	0x0
Bit[1:0]	TRIG_EXT0	R/W	1x: rising edge & falling edge 01: Falling edge 00: Rising edge	0x0

- **P_EXT_INT_Ctrl (EXT INT Interrupt Enable)**

Register	Offset		Description	Initial Value
P_EXT_INT_Ctrl	GPIO_BA+0x64		EXT INT Interrupt Enable	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:2]	Reserved			0x0
Bit[1]	EXT1_IEN	R/W	1: EXT1 interrupt enabled 0: EXT1 interrupt disabled (default)	0x0

Bit[1:0]	EXT0_IEN	R/W	1: EXT0 interrupt enabled 0: EXT0 interrupt disabled (default)	0x0

- **P_EXT_INT_Flag (EXT INT Interrupt Flag)**

Register	Offset		Description	Initial Value
P_EXT_INT_Flag	GPIO_BA+0x68		EXT INT Interrupt Flag	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:2]	Reserved			0x0
Bit[1]	EXT1_INTF	R/W1C	1: Have EXT1 interrupt flag 0: No EXT1 interrupt flag	0x0
Bit[1:0]	EXT0_INTF	R/W1C	1: Have EXT0 interrupt flag 0: No EXT0 interrupt flag	0x0

- **P_PULL_HIGH_CFG (Port pull-high Config register)**

Register	Offset		Description	Initial Value
P_PULL_HIGH_CFG	GPIO_BA+0x70		Port pull-high Config register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:6]	Reserved			0x0
Bit[5]	PD_L_WSB	R/W	1: PD0 ~ PD1 use Strong pull-up (100K) 0: PD0 ~ PD1 use Weak pull-up (1M)	0x0
Bit[4]	PC_L_WSB	R/W	1: PC0 ~ PC3 use Strong pull-up (100K) 0: PC0 ~ PC3 use Weak pull-up (1M)	0x0
Bit[3]	Reserved			0x0
Bit[2]	PB_L_WSB	R/W	1: PB0 ~ PB5 use Strong pull-up (100K) 0: PB0 ~ PB5 use Weak pull-up (1M)	0x0
Bit[1]	PA_H_WSB	R/W	1: PA8 ~ PA15 use Strong pull-up (100K) 0: PA8 ~ PA15 use Weak pull-up (1M)	0x0
Bit[0]	PA_L_WSB	R/W	1: PA0 ~ PA7 use Strong pull-up (100K) 0: PA0 ~ PA7 use Weak pull-up (1M)	0x0

- **P_PD_DIR (Port D Input Output Control)**

Register	Offset		Description	Initial Value
P_PD_DIR	GPIO_BA+0x80		Port D input output control	0x0000_0003
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:2]	Reserved			0x0
Bit[1:0]	PD_DIR	R/W	I/O Direction 1: Input 0: Output	0x3

- **P_PD_Data (Port D Data Output Register)**

Register	Offset		Description	Initial Value
P_PD_Data	GPIO_BA+0x84		Port D data output register	0x0000_0003
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:2]	Reserved			0x0
Bit[1:0]	PD_DATA	R/W	Data output 1: Data output 1 0: Data output 0	0x3

- **P_PD_PAD (Port D Pad Input Register)**

Register	Offset		Description	Initial Value
P_PD_PAD	GPIO_BA+0x88		Port D pad input register	0x0000_000u
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:2]	Reserved			0x0
Bit[1:0]	PD_PAD	R/W	Read pad data	0xu

- **P_PD_Wakeup_Mask (Port D Config Register0)**

Register	Offset		Description	Initial Value
P_PD_Wakeup_Mask	GPIO_BA+0x8C		Port D Config register0	0x0003_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:18]	Reserved			0x0
Bit[17:16]	PD_CCS0	R/W	PD1 ~ PD0 Sink current select 1: Normal Sink 0: Large Sink	0x3
Bit[15:2]	Reserved			0x0
Bit[1:0]	PD_MASK	R/W	PD1 ~ PD0 Wakeup Function Mask 1: Have wakeup function 0: No Wake function	0x0

- **P_PD_CFG (Port D Config Register1)**

Register	Offset		Description	Initial Value
P_PD_CFG	GPIO_BA+0x90		Port D Config register1	0x00FF_0003
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:24]	Reserved			0x0
Bit[23:16]	Reserved			0xFF
Bit[15:2]	Reserved			0x0
Bit[1:0]	PD_PUENB	R/W	PD1 ~ PC0 Pull-up Resistor select 1: Pull-up Resistor disabled 0: Pull-up Resistor enabled	0x3

- **P_PD_MFP (Port D Multi-Function Register)**

Register	Offset		Description	Initial Value
P_PD_MFP	GPIO_BA+0x94		Port D multi-function register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:4]	Reserved			0x0
Bit[3:2]	PD1_MFP	R/W	11: rx 10: reserved 01: reserved 00: gpio	0x0
Bit[1:0]	PD0_MFP	R/W	11: tx 10: reserved 01:reserved 00: gpio	0x0

2.9 Timer Controller

2.9.1 Overview and Features

The NX12F_NX13F has three 16-bit timers: TIMER0 / TIMER1 / TIMER2 / TIMER3, which can be used as a trigger source for DAC / PWM or as a function of time delay, clock generation, etc.

- Programmable source of timer clock
- 16-bit counter for each timer
- Support IR TX function
- TIMER0 support software touch

2.9.2 Block Diagram

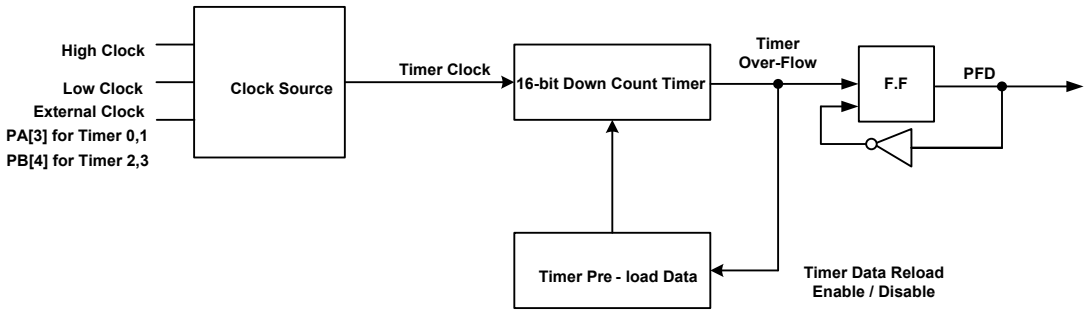


Figure 7 Timer Block Diagram

2.9.3 Function Description

The NX12F_NX13F provides up to four multi-function timers for users to implement a variety of applications. Beside clock sources selected from High and Low clocks, these timers also accept external clock sources from PA port as well. When external clocks are used as the source for the timer, users can further decide if to synchronize with SYS_CLK. It supports IR function, which is generated by timer.

2.9.4 Register Map

Register	Offset	R/W	Description	Initial Value
Timer Base Address: TMR_BA = 0xF0_4000				
P_TMR0_Ctrl	TMR_BA+0x00	R/W	Timer 0 Control Register	0x0000_0372
P_TMR0_Data	TMR_BA+0x04	R/W	Timer 0 Data Register	0xFFFF_FFFF
P_TMR0_INT_Ctrl	TMR_BA+0x08	R/W	Timer 0 Interrupt Control Register	0x0000_0000
P_TMR0_Flag	TMR_BA+0x0C	R/W1C	Timer 0 Interrupt FLAG	0x0000_0000
P_TMR1_Ctrl	TMR_BA+0x40	R/W	Timer 1 Control Register	0x0000_0372
P_TMR1_Data	TMR_BA+0x44	R/W	Timer 1 Data Register	0xFFFF_FFFF
P_TMR1_INT_Ctrl	TMR_BA+0x48	R/W	Timer 1 Interrupt Control Register	0x0000_0000
P_TMR1_Flag	TMR_BA+0x4C	R/W1C	Timer 1 Interrupt FLAG	0x0000_0000
P_TMR2_Ctrl	TMR_BA+0x80	R/W	Timer 2 Control Register	0x0000_0372
P_TMR2_Data	TMR_BA+0x84	R/W	Timer 2 Data Register	0xFFFF_FFFF

P_TMR2_INT_Ctrl	TMR_BA+0x88	R/W	Timer 2 Interrupt Control Register	0x0000_0000
P_TMR2_Flag	TMR_BA+0x8C	R/W1C	Timer 2 Interrupt FLAG	0x0000_0000
P_TMR3_Ctrl	TMR_BA+0xC0	R/W	Timer 3 Control Register	0x0000_0372
P_TMR3_Data	TMR_BA+0xC4	R/W	Timer 3 Data Register	0xFFFF_FFFF
P_TMR3_INT_Ctrl	TMR_BA+0xC8	R/W	Timer 3 Interrupt Control Register	0x0000_0000
P_TMR3_Flag	TMR_BA+0xCC	R/W1C	Timer 3 Interrupt FLAG	0x0000_0000

2.9.5 Register Description

R: Read only, **W**: Write only, **W1C**: Write 1 to clear, **R/W**: Read and Write

- **P_TMR0_Ctrl (Timer 0 Control Register)**

Register	Offset		Description	Initial Value
P_TMR0_Ctrl	TMR_BA+0x00		Timer 0 Control Register	0x0000_0372
Bit	**Name**	**R/W**	**Descriptions**	**Initial V**
Bit[31:26]	Reserved			0x0000_0000
Bit[25:24]	SWT_SOURCE	R/W	11: TMR3OV 10: TMR2OV 01: TMR1OV 00: PA	0x0
Bit[23:20]	SWT_SEL	R/W	0000 ~ 1111: PA0 ~ PA15	0x0
Bit[19]	Reserved			
Bit[18]	SWT_TRGS	R/W	1: 1->0 stop counter 0: 0->1 stop counter	
Bit[17]	SWT_TRIG	W	write 1 to trig CNT count	
Bit[16]	SWT_EN	R/W	1: Software touch 0: 16-bit timer	
Bit[15:10]	Reserved			
Bit[9:8]	CKS	R/W	11: TMCK's source from LOCLK (default) 10: TMCK's source from HICLK 01: TMCK's source from EXCLK (no sync w/ SYS_CLK) 00: TMCK's source from EXCLK (sync w/ SYS_CLK)	0x3
Bit[7]	Reserved			0x0
Bit[6:4]	Reserved			0x7
Bit[3]	IR_STP	R/W	1: IR stop at 1 0: IR stop at 0 (default)	0x0
Bit[2]	IR_EN	R/W	1: IR enable 0: IR disable (default)	0x0
Bit[1]	TMR_LD	R/W	1: TMR reload in overflow (TMR) (default) 0: TMR no reload in overflow (CNT)	0x1
Bit[0]	TMR_EN	R/W	1: TMR enable 0: TMR disabled (default)	0x0

- **P_TMR0_Data (Timer 0 Data Register)**

Register	Offset		Description	Initial Value
P_TMR0_Data	TMR_BA+0x04		Timer 0 Data Register	0xFFFF_FFFF
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	TMR_DAT	R	TMR's counter data	0xFFFF
Bit[15:0]	TMR_LDAT	R/W	TMR's preload Data	0xFFFF

- **P_TMR0_INT_Ctrl (Timer 0 Interrupt Control Register)**

Register	Offset		Description	Initial Value
P_TMR0_INT_Ctrl	TMR_BA+0x08		Timer 0 Interrupt Control Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:2]	Reserved			0x0
Bit[1]	SWT_EN	R/W	1: SWT's interrupt enable 0: SWT's interrupt disable (default)	0x0
Bit[0]	TMR_IEN	R/W	1: TMR's interrupt enabled 0: TMR's interrupt disabled (default)	0x0

- **P_TMR0_Flag (Timer 0 Interrupt FLAG)**

Register	Offset		Description	Initial Value
P_TMR0_Flag	TMR_BA+0x0C		Timer 0 Interrupt FLAG	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:2]	Reserved			0x0
Bit[1]	SWT_INTF	R/W1C	1: SWT's interrupt flag 0: SWT's interrupt no flag (default)	
Bit[0]	TMR_INTF	R/W1C	1: TMR's interrupt flag 0: TMR's interrupt no flag (default)	0x0

- **P_TMR1_Ctrl (Timer 1 Control Register)**

Register	Offset		Description	Initial Value
P_TMR1_Ctrl	TMR_BA+0x40		Timer 1 Control Register	0x0000_0302
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:10]	Reserved			0x0000_0302
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[9:8]	CKS	R/W	11: TMCK's source from LOCLK (default) 10: TMCK's source from HICLK 01: TMCK's source from EXCLK (no sync w/ SYS_CLK) 00: TMCK's source from EXCLK (sync w/ SYS_CLK)	0x3
Bit[7]	Reserved			0x0
Bit[6:4]	Reserved			0x7
Bit[3]	IR_STP	R/W	1: IR stop at 1 0: IR stop at 0 (default)	0x0
Bit[2]	IR_EN	R/W	1: IR enable 0: IR disable (default)	0x0
Bit[1]	TMR_LD	R/W	1: TMR reload in overflow (TMR) 0: TMR no reload in overflow (CNT)	0x1
Bit[0]	TMR_EN	R/W	1: TMR enable 0: TMR disabled (default)	0x0

- **P_TMR1_Data (Timer 1 Data Register)**

Register	Offset		Description	Initial Value
P_TMR1_Data	TMR_BA+0x44		Timer 1 Data Register	0xFFFF_FFFF
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	TMR_DAT	R	TMR's counter data	0xFFFF
Bit[15:0]	TMR_LDAT	R/W	TMR's preload Data	0xFFFF

- **P_TMR1_INT_Ctrl (Timer 1 Interrupt Control Register)**

Register	Offset		Description	Initial Value
P_TMR1_INT_Ctrl	TMR_BA+0x48		Timer 1 Interrupt Control Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:1]	Reserved			0x0
Bit[0]	TMR_IEN	R/W	1: TMR's interrupt enabled 0: TMR's interrupt disabled (default)	0x0

- **P_TMR1_Flag (Timer 1 Interrupt FLAG)**

Register	Offset		Description	Initial Value
P_TMR1_Flag	TMR_BA+0x4C		Timer 1 Interrupt FLAG	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:1]	Reserved			0x0
Bit[0]	TMR_INTF	R/W1C	1: TMR's interrupt flag 0: TMR's interrupt no flag (default)	0x0

- **P_TMR2_Ctrl (Timer 2 Control Register)**

Register	Offset		Description	Initial Value
P_TMR2_Ctrl	TMR_BA+0x80		Timer 2 Control Register	0x0000_0302
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:10]	Reserved			0x0000_0302
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[9:8]	CKS	R/W	11: TMCK's source from LOCLK (default) 10: TMCK's source from HICLK 01: TMCK's source from EXCLK (no sync w/ SYS_CLK) 00: TMCK's source from EXCLK (sync w/ SYS_CLK)	0x3
Bit[7]	Reserved			0x0
Bit[6:4]	Reserved			0x7
Bit[3]	IR_STP	R/W	1: IR stop at 1 0: IR stop at 0 (default)	0x0
Bit[2]	IR_EN	R/W	1: IR enable 0: IR disable (default)	0x0
Bit[1]	TMR_LD	R/W	1: TMR reload in overflow (TMR) 0: TMR no reload in overflow (CNT)	0x1
Bit[0]	TMR_EN	R/W	1: TMR enable 0: TMR disabled (default)	0x0

- **P_TMR2_Data (Timer 2 Data Register)**

Register	Offset		Description	Initial Value
P_TMR2_Data	TMR_BA+0x84		Timer 2 Data Register	0xFFFF_FFFF
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	TMR_DAT	R	TMR's counter data	0xFFFF
Bit[15:0]	TMR_LDAT	R/W	TMR's preload Data	0xFFFF

- **P_TMR2_INT_Ctrl (Timer 2 Interrupt Control Register)**

Register	Offset		Description	Initial Value
P_TMR2_INT_Ctrl	TMR_BA+0x88		Timer 2 Interrupt Control Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:1]	Reserved			0x0
Bit[0]	TMR_IEN	R/W	1: TMR's interrupt enabled 0: TMR's interrupt disabled (default)	0x0

- **P_TMR2_Flag (Timer 2 Interrupt FLAG)**

Register	Offset		Description	Initial Value
P_TMR2_Flag	TMR_BA+0x8C		Timer 2 Interrupt FLAG	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:1]	Reserved			0x0
Bit[0]	TMR_INTF	R/W1C	1: TMR's interrupt flag 0: TMR's interrupt no flag (default)	0x0

2.10 I²C Serial Interface Controller (Master/Slave)

2.10.1 Overview and Features

The controller is an I²C (Inter-Integrated Circuit) master/slave controller.

- Support Standard-mode (100 Kb/s), Fast-mode (400 Kb/s) and Fast-mode Plus (1 Mb/s) protocols
- Programmable Master/Slave mode
- Support 7-bit and 10-bit addressing mode
- Support general call address
- Auto clock stretching

2.10.2 Block Diagram

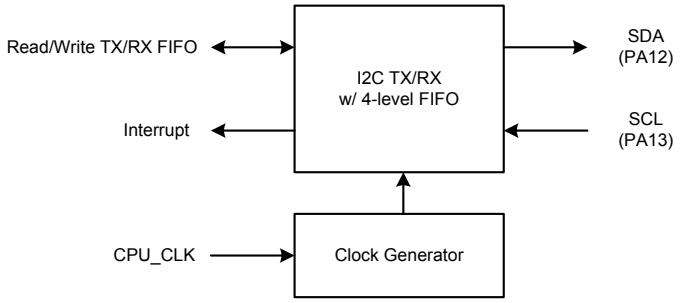


Figure 8 I²C Block Diagram

2.10.3 Function Description

The controller can act as either an I²C master device or an I²C slave device, depending on the control register settings.

I²C Master: As an I²C master, every transaction can be delineated by four phases: Start, Address, Data and Stop.

I²C Slave: As an I²C slave, the controller is addressed when the address byte of an I²C transaction matches the Address Register. An Address Hit interrupt can be generated for the software to prepare for the subsequent operations.

General Call Address: The controller at the slave mode will respond with an ACK to the general call address and set the GenCall field of the Status Register.

Auto Clock Stretch: When the software is not ready for the next byte of data or when the FIFO is full, it can automatically pause bus transactions by stretching clocks on the I²C-bus

Auto-ACK: With Auto-ACK, it automatically generates proper acknowledgements for each byte received except for the last byte, which should be responded with a NACK according to the I²C-bus protocol. By enabling the Byte Receive Interrupt, user can turned off Auto-ACK to determine each byte's acknowledgement status.

2.10.4 Register Map

Register	Offset	R/W	Description	Initial Value
I²C Base Address: IIC_BA = 0xF0_1000				
P_I2C_IDREV	I2C_BA+0x00	R	I²C ID and Revision Register	0x2021_0002
P_I2C_FIFO_Ctrl	I2C_BA+0x10	R	I²C FIFO Configuration Register	0x0000_0001
P_I2C_INT_Ctrl	I2C_BA+0x14	R/W	I²C Interrupt Enable Register	0x0000_0000
P_I2C_Status	I2C_BA+0x18	R/W1C	I²C Status Register	0xu000_0001
P_I2C_Addr	I2C_BA+0x1C	R/W	I²C Address Register	0xu000_0000
P_I2C_Data	I2C_BA+0x20	R/W	I²C Data Register	0xu000_0000
P_I2C_Mode	I2C_BA+0x24	R/W	I²C Control Register	0x0000_1E00
P_I2C_Cmd	I2C_BA+0x28	R/W	I²C Command Register	0x0000_0000
P_I2C_Ctrl	I2C_BA+0x2C	R/W	I²C Controller Setting Register	0x0525_2100

2.10.5 Register Description

R: Read only, **W**: Write only, **W1C**: Write 1 to clear, **R/W**: Read and Write

- **P_I2C_IDREV (I²C ID and Revision Register)**

Register	Offset		Description	Initial Value
P_I2C_IDREV	I2C_BA+0x00		I²C ID and Revision Register	0x2021_0002
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:12]	ID	R	ID number for ATCIIC100	0x2_0210
Bit[11:4]	RevMajor	R	Major revision number	0x0
Bit[3:0]	RevMinor	R	Minor revision number	0x2

- **P_I2C_FIFO_Ctrl (I²C FIFO Configuration Register)**

Register	Offset		Description	Initial Value
P_I2C_FIFO_Ctrl	I2C_BA+0x10		I²C FIFO Configuration Register	0x0000_0001
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:2]	Reserved			0x0
Bit[1:0]	FIFOSize	R	FIFO SIZE 0: 2 bytes 1: 4 bytes 2: 8 bytes 3: 16 bytes	0x1

- **P_I2C_INT_Ctrl (I²C Interrupt Enable Register)**

Register	Offset		Description	Initial Value
P_I2C_INT_Ctrl	I2C_BA+0x14		I²C Interrupt Enable Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:10]	Reserved			0x0
Bit[9]	Cmpl	R/W	Set to enable the Completion Interrupt Master: interrupts when a transaction is issued from this master and completed without losing the bus arbitration Slave: interrupts when a transaction addressing the controller is completed	0x0

Bit	Name	R/W	Descriptions	Initial Value
Bit{8]	ByteRecv	R/W	Set to enable the Byte Receive Interrupt Interrupts when a byte of data is received Auto-ACK will be disabled if this interrupt is enabled, that is, the software needs to ACK/NACK the received byte manually.	0x0
Bit[7]	ByteTrans	R/W	Set to enable the Byte Transmit Interrupt Interrupts when a byte of data is transmitted	0x0
Bit[6]	Start	R/W	Set to enable the START Condition Interrupt Interrupts when a START condition/repeated START condition is detected	0x0
Bit[5]	Stop	R/W	Set to enable the STOP Condition Interrupt Interrupts when a STOP condition is detected	0x0
Bit[4]	ArbLose	R/W	Set to enable the Arbitration Lose Interrupt Master: interrupts when the controller loses the bus arbitration Slave: not available in this mode	0x0
Bit[3]	AddrHit	R/W	Set to enable the Address Hit Interrupt Master: interrupts when the addressed slave returned an ACK Slave: interrupts when the controller is addressed	0x0
Bit[2]	FIFOHalf	R/W	Set to enable the FIFO Half Interrupt Receiver: Interrupts when the FIFO is half-full Transmitter: Interrupts when the FIFO is half-empty This interrupt depends on the transaction direction; do not enable this interrupt unless the transfer direction is determined, otherwise unintended interrupts may be triggered.	0x0
Bit1]	FIFOFull	R/W	Set to enable the FIFO Full Interrupt Interrupts when the FIFO is full	0x0
Bit[0]	FIFOEmpty	R/W	Set to enable the FIFO Empty Interrupt Interrupts when the FIFO is empty	0x0

- **P_I2C_Status (I^2C Status Register)**

Register	Offset		Description	Initial Value
P_I2C_Status	I2C_BA+0x18		I^2C Status Register	0xu000_0001
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:15]	Reserved			0x0
Bit[14]	LineSDA	R	Indicates the current status of the SDA line on the bus 1: High 0: Low	SDA line status
Bit[13]	LineSCL	R	Indicates the current status of the SCL line on the bus 1: High 0: Low	SCL line status
Bit[12]	GenCall	R	Indicates that the address of the current transaction is a general call address. This status is only valid in slave mode. 1: General call 0: Not general call	0x0
Bit[11]	BusBusy	R	Indicates that the bus is busy The bus is busy when a START condition is on bus and it ends when a STOP condition is seen on bus. 1: Busy 0: Not busy	0x0

Bit	Name	R/W	Descriptions	Initial Value
Bit[10]	ACK	R	Indicates the type of the last received/transmitted acknowledgement bit 1: ACK 0: NACK	0x0
Bit[9]	Cmpl	R/W1C	Transaction Completion Master: Indicate that a transaction is issued has been issued from this master and completed without losing the bus arbitration. Slave: Indicate that a transaction addressing the controller has been completed. This status bit must be cleared to receive the next transaction; otherwise, the next incoming transaction will be blocked.	0x0
Bit{8]	ByteRecv	R/W1C	Indicates that a byte of data has been received	0x0
Bit[7]	ByteTrans	R/W1C	Indicates that a byte of data has been transmitted	0x0
Bit[6]	Start	R/W1C	Indicates that a START Condition or a repeated START condition has been transmitted/received	0x0
Bit[5]	Stop	R/W1C	Indicates that a STOP Condition has been transmitted/received	0x0
Bit[4]	ArbLose	R/W1C	Indicates that the controller has lost the bus arbitration (master mode only)	0x0
Bit[3]	AddrHit	R/W1C	Master: indicates that a slave has responded to the transaction. Slave: indicates that a transaction is targeting the controller (including the General Call).	0x0
Bit[2]	FIFOHalf	R/W	Transmitter: Indicates that the FIFO is half-full. Receiver: Indicates that the FIFO is half-empty	0x0
Bit1]	FIFOFull	R/W	Indicates that the FIFO is full	0x0
Bit[0]	FIFOEmpty	R/W	Indicates that the FIFO is empty	0x01

- **P_I2C_Addr (I²C Address Register)**

Register	Offset		Description	Initial Value
P_I2C_Addr	I2C_BA+0x1C		I²C Address Register	0xu000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:10]	Reserved			
Bit[9:0]	Addr	R/W	The slave address. For 7-bit addressing mode, the most significant 3 bits are ignored and only the least-significant 7 bits of Addr are valid	0x0

- **P_I2C_Data (I²C Data Register)**

Register	Offset		Description	Initial Value
P_I2C_Data	I2C_BA+0x20		I²C Data Register	0xu000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:8]	Reserved			0x0000_0000
Bit[7:0]	Data	R/W	Write this register to put one byte of data to the FIFO Read this register to get one byte of data from the FIFO	0x0

- **P_I2C_Mode (I²C Control Register)**

Register	Offset		Description	Initial Value
P_I2C_Mode	I2C_BA+0x24		I²C Control Register	0x0000_1E00
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:13]	Reserved			0x0

Bit[12]	Phase_start	R/W	Enable this bit to send a START condition at the beginning of transaction Master mode only	0x1
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[11]	Phase_addr	R/W	Enable this bit to send the address after START condition Master mode only	0x1
Bit[10]	Phase_dat	R/W	Enable this bit to send the data after Address phase Master mode only	0x1
Bit[9]	Phase_stop	R/W	Enable this bit to send a STOP condition at the end of a transaction Master mode only	0x1
Bit[8]	Dir	R/W	Transaction direction Master: Set this bit to determine the direction for the next transaction. 0: Transmitter 1: Receiver Slave: The direction of the last received transaction. 0: Receiver 1: Transmitter	0x0
Bit[7:0]	DataCnt	R/W	Data counts in bytes. Master: The number of bytes to transmit/receive. 0 means 256 bytes. DataCnt will be decreased by one for each byte transmitted/received. Slave: the meaning of DataCnt depends on the DMA mode: If DMA is not enabled, DataCnt is the number of bytes transmitted/received from the bus master. It is reset to 0 when the controller is addressed and then increased by one for each byte of data transmitted/received. If DMA is enabled, DataCnt is the number of bytes to transmit/receive. It will not be reset to 0 when the slave is addressed and it will be decreased by one for each byte of data transmitted/received.	0x0

- **P_I2C_Cmd (I^2C Command Register)**

Register	Offset		Description	Initial Value
P_I2C_Cmd	I2C_BA+0x28		I^2C Command Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:3]	Reserved			0x0
Bit[2:0]	CMD	R/W	Write this register with the following values to perform the corresponding actions: 0x0: no action 0x1: issue a data transaction (Master only) 0x2: respond with an ACK to the received byte 0x3: respond with a NACK to the received byte 0x4: clear the FIFO 0x5: reset the I^2C controller (abort current transaction, clear the Status Register and clear the FIFO) When issuing a data transaction by writing 0x1 to this register, the CMD field stays at 0x1 for the duration of the entire transaction, and it is only cleared to 0x0 after when the transaction has completed or when the controller loses the arbitration. ***Note: No transaction will be issued by the controller when all phases (Start, Address, Data and Stop) are disabled.***	0x0

- **P_I2C_Ctrl (I²C Controller Setting Register)**

Register	Offset		Description	Initial Value
P_I2C_Ctrl	I2C_BA+0x2C		I²C Controller Setting Register	0x0525_2100
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:29]	Reserved			0x0
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[28:24]	T_SUDAT	R/W	T_SUDAT defines the data setup time before releasing the SCL Setup time = (4 + T_SP + T_SUDAT) * tpclk tpclk = PCLK period	0x5
Bit[23:21]	T_SP	R/W	T_SP defines the pulse width of spikes that must be suppressed by the input filter Pulse width = T_SP * tpclk	0x1
Bit[20:16]	T_HDDAT	R/W	T_SUDAT defines the data hold time after SCL goes LOW Hold time = (4 + T_SP + T_HDDAT) * tpclk	0x5
Bit[15:14]	Reserved			0x0
Bit[13]	T_SCLRatio	R/W	The LOW period of the generated SCL clock is defined by the combination of T_SCLRatio and T_SCLHi values. When T_SCLRatio = 0, the LOW period is equal to HIGH period. When T_SCLRatio = 1, the LOW period is roughly two times of HIGH period. SCL LOW period = (4 + T_SP + T_SCLHi * ratio) * tpclk 1: ratio = 2 0: ratio = 1 This field is only valid when the controller is in the master mode.	0x1
Bit[12:4]	T_SCLHi	R/W	The HIGH period of generated SCL clock is defined by T_SCLHi. SCL HIGH period = (4 + T_SP + T_SCLHi) * tpclk The T_SCLHi value must be greater than T_SP and T_HDDAT values. This field is only valid when the controller is in the master mode.	0x10
Bit[3]	DMAEn	R/W	Enable the direct memory access mode data transfer 1: Enabled 0: Disabled	0x0
Bit[2]	Master	R/W	Configure this device as a master or a slave 1: Master mode 0: Slave mode	0x0
Bit[1]	Addressing	R/W	I²C addressing mode: 1: 10-bit addressing mode 0: 7-bit addressing mode	0x0
Bit[0]	IICEN	R/W	Enable the ATCIIC100 I²C controller 1: Enabled 0: Disabled	0x0

2.11 Serial Peripheral Interface (SPI) Controller

2.11.1 Overview and Features

- There are 2 SPI masters (SPI0 and SPI1). Each master can connect 1 slave device only.
- The SPI0 and SPI1 support Transmit and Receive mode and only the SPI0 supports XIP modes.
- There are 16 bytes and 8 bytes of receive, and transmit FIFO in SPI0 and SPI1, respectively.
- The SPI0 supports single/dual/quad I/O mode, and SPI1 only supports single/dual I/O mode.
- The maximum frequency of bit rate clock is SYS_CLK.

2.11.2 Block Diagram

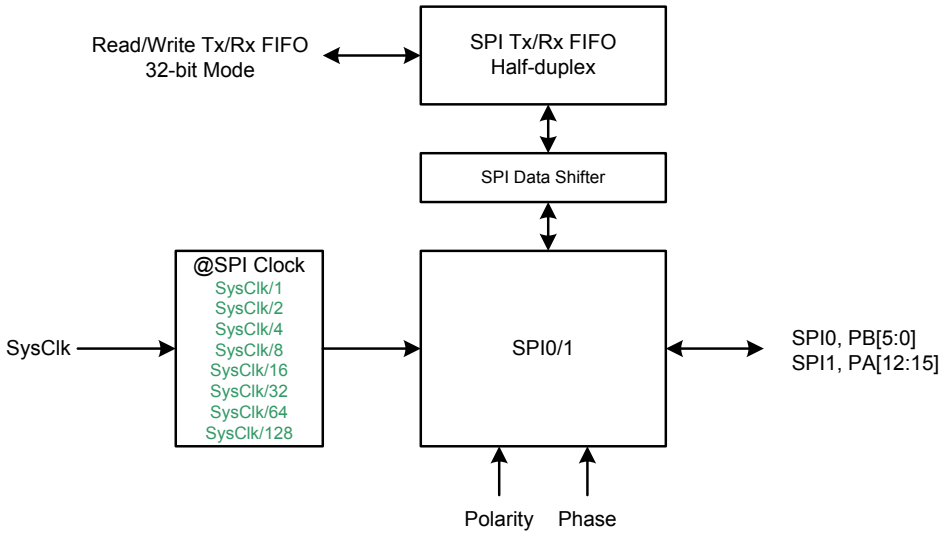


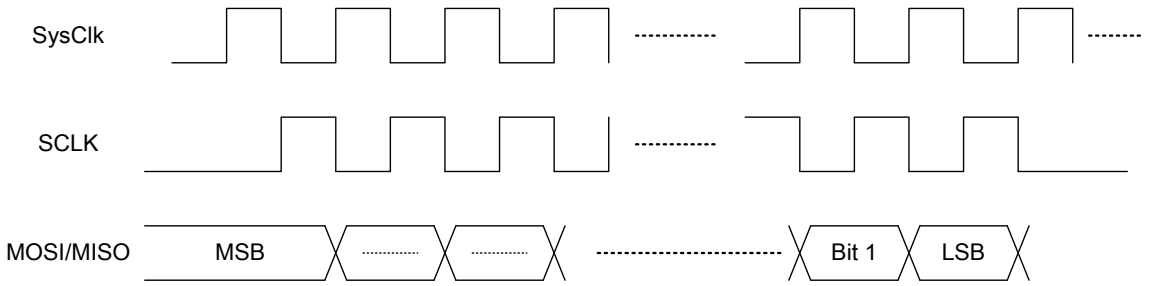
Figure 9 SPI Block Diagram

2.11.3 Function Description

There are 2 components in the SPI controller. One is a SPI0 single/dual/quad I/O master series interface protocol, the other is SPI1 single/dual I/O master series interface protocol. They can be selected for connecting to a series-slave peripheral device, respectively.

2.11.3.1 Clock Ratios

The SPI controller is selected, the maximum frequency of the bit-rate clock (SCLK) is the same the frequency of SYC_CLK. The minimum frequency of the bit-rate clock (SCLK) is SYS_CLK / 128. The P_SPIx_Ctrl[10:8] (x=0, or 1) (SPISEL) bits are selected to generate the bit-rate clock (SCLK).



2.11.3.2 Interrupt

The SPI controller supports individual interrupt request. The SPI interrupts are described below.

- While the transmission byte number is equivalent to send byte number.
- While the transmission byte number is equivalent to receive byte number.

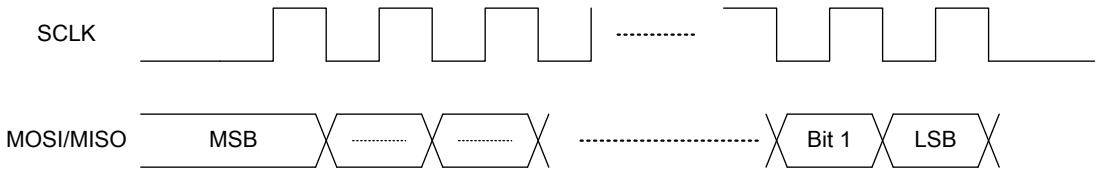
2.11.3.3 Transfer Mode

The SPI controller operates in the following two modes when transferring data on the serial bus:

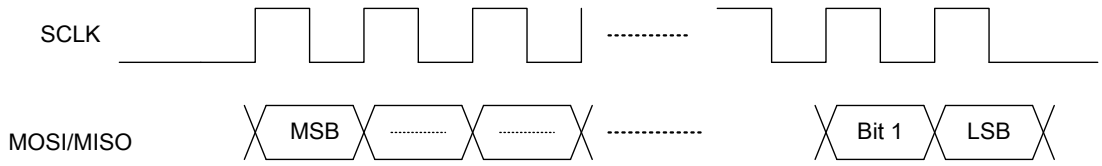
- Data mode

The P_SPIx_Ctrl[1] (TXD), or P_SPIx_Ctrl[0] (RXD) bit is set to start transmit data, or receive data. Transmit data are popped from FIFO and are sent through the MOSI line to the target device. The receive data from the target device through the MISO line is moved into FIFO. The P_SPIx_Ctrl[20-x:16] (x=0, or 1) (SPI_NUM) bits are expected transmit data bytes are equivalent to the transmit data bytes and the TXD bit is auto-clear, or expected receive data bytes are equivalent to the receive data bytes and the RXD bit is auto-clear. The P_SPI0_Ctrl[5:4] (SPI_DATS) bits are selected single, dual, or quad I/O mode to transmit/receive data.

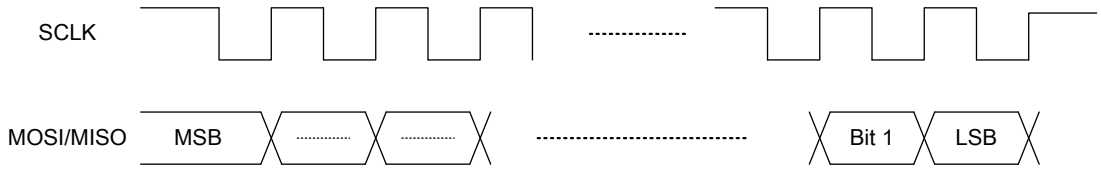
The clock polarity P_SPIx_Ctrl[7] (SPI_POL) configuration parameter determines whether the inactive state of the serial clock is high or low. To transmit data, both SPI peripherals must have identical serial clock phase P_SPIx_Ctrl[6] (SPI_PHA) and clock polarity P_SPIx_Ctrl[7] (SPI_POL) values. When the configuration parameter SPI_POL = 0, for SPI_PHA = 0, data are captured on the clock's rising edge and data is output on a falling edge, for SPI_PHA = 1, data are captured on the clock's falling and data is output on a rising edge. When the configuration parameter SPI_POL = 1, for SPI_PHA = 0, data are captured on the clock's falling edge and data is output on a rising edge, for SPI_PHA = 1, data are captured on the clock's rising and data is output on a falling edge.



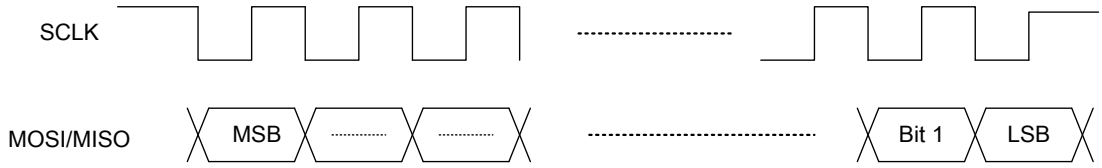
SPI_POL = 0, and SPI_PHA=0



SPI_POL = 0, and SPI_PHA=1



SPI_POL = 1, and SPI_PHA=0



SPI_POL = 1, and SPI_PHA=1

- XIP mode

The SPI0 supports XIP mode by using P_SPI0_Addr_Mode to control transfer I/O mode. The SPI0 in XIP mode is used to transmit a fixed command, an address and dummy cycles to the SPI flash device. Typically transmit data takes three data frames (8-bit command followed by 24/32-bit address and 0/8-bit dummy cycles). A transmission of the command and address is captured by a SPI flash device and the 32-bit data is shifted out by the SPI flash device. The P_SPI0_Addr_Mode[2:0] (SPI_FMT) bits are selected which command is sent to SPI flash device. The commands take Read Data(0x03), Fast Read Dual I/O(0xBB), Fast Read Quad I/O(0xEB), Fast Read Dual Output(0x3B), and Fast Read Quad Output(0x6B). The P_SPI0_Addr_Mode[3] (ADR_NUM) bit is selected 3/4-byte address mode. In XIP mode, the 2MB address range in SPI flash device is divided 2 parts, one is XIP_ROM1(1MB) and the other is XIP_ROM2(1MB). The address range of the XIP_ROM1 in the NX1 system is fixed at 0x800000 ~ 0x8FFFFF is mapped to 0x000000 ~ 0x0FFFFF in SPI flash device. The SPI0_INS[17:8] (SPI_BANK) bits are selected the address range of the XIP_ROM2 in the NX1 system is fixed at 0x900000 ~ 0x9FFFFF is mapped to 0xN00000 ~ 0xNFFFFF (N=SPI_BANK=0,1,2,...) in SPI flash device. For example: SPI_BANK = 2, the address range of the NX1 system is 0x900000 ~ 0x9FFFFF is mapped to 0x200000 ~ 0x2FFFFF in SPI flash device.

2.11.4 Register Map

Register	Offset	R/W	Description	Initial Value
SPI Base Address: SPI_BA = 0xF0_B000				
P_SPI0_Ctrl	SPI_BA+0x00	R/W	SPI0 Control Register	0x0000_0700
P_SPI0_Addr_Mode	SPI_BA+0x04	R/W	SPI0 Address Mode Setting Register	0x0000_0000
P_SPI0_FIFO_RST	SPI_BA+0x08	R/W	SPI0 FIFO Reset Register	0x0000_0004
P_SPI0_Data	SPI_BA+0x10	R/W	SPI0 Data Register	0x0000_0000
P_SPI0_INTEN	SPI_BA+0x14	R/W	SPI0 Interrupt Enable	0x0000_0000
P_SPI0_INTF	SPI_BA+0x18	R/W1C	SPI0 Interrupt Flag	0x0000_0000
P_SPI1_Ctrl	SPI_BA+0x40	R/W	SPI1 Control Register	0x0000_0700
P_SPI1_FIFO_RST	SPI_BA+0x48	R/W	SPI1 FIFO Reset Register	0x0000_0004
P_SPI1_Data	SPI_BA+0x50	R/W	SPI1 Data Register	0x0000_0000
P_SPI1_INTEN	SPI_BA+0x54	R/W	SPI1 Interrupt Enable	0x0000_0000
P_SPI1_INTF	SPI_BA+0x58	R/W1C	SPI1 Interrupt Flag	0x0000_0000

2.11.5 Register Description

R: Read only, **W**: Write only, **W1C**: Write 1 to clear, **R/W**: Read and Write

- **P_SPI0_Ctrl (SPI0 Control Register)**

Register	Offset		Description	Initial Value
P_SPI0_Ctrl	SPI_BA+0x00		SPI0 Control Register	0x0000_0700
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:20]	Reserved			0x0
Bit[19:16]	SPI_NUM	R/W	SPI Txd/Rcv 's Num byte	0x0
Bit[15:11]	Reserved			0x0
Bit[10:8]	SPISEL	R/W	111: SPICK0=SYSCLK/1 (default) 110: SPICK0=SYSCLK/2 101: SPICK0=SYSCLK/4 100: SPICK0=SYSCLK/8 011: SPICK0=SYSCLK/16 010: SPICK0=SYSCLK/32 001: SPICK0=SYSCLK/64 000: SPICK0=SYSCLK/128	0x7
Bit[7]	SPI_POL	R/W	1: SPICK stop at 1, when #CS = 1 0: SPICK stop at 0, When #CS = 1 (default)	0x0
Bit[6]	SPI_PHA	R/W	1: sampling on second clock edge 0: sampling on first clock edge (default)	0x0
Bit[5:4]	SPI_DATS	R/W	11: SPI use 4 IO 10: SPI use 2 IO 01: SPI use 1 IO (DI connect to DO) 00: SPI use 1 IO (default)	0x0
Bit[3:2]	Reserved			0x0
Bit[1]	TXD	R/W	1: start SEND DATA 0: do nothing (default)	0x0
Bit[0]	RXD	R/W	1: start Receive DATA 0: do nothing (default)	0x0

- **P_SPI0_Addr_Mode (SPI0 Address Mode Setting Register)**

Register	Offset	Description	Initial Value

P_SPI0_Addr_Mode	SPI_BA+0x04		SPI0 Address Mode Setting Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:18]	Reserved			0x0
Bit[17:8]	SPI_BANK	R/W	XIP mode Bank select	0x0
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[7:4]	Reserved			0x0
Bit[3]	ADR_NUM	R/W	1: 4 ADR for Instruction mode 0: 3 ADR for Instruction mode (default)	0x0
Bit[2:0]	SPI_FMT	R/W	111: Reserved 110: Reserved 101: command 6BH, 5W, 1-1-4 100: command 3BH, 5W, 1-1-2 011: command EBH, 5W, 1-4-4 010: command BBH, 5W, 1-2-2 001: command 03H, 5W, 1-1-1 (DI connect to DO) 000: command 03H, 5W, 1-1-1 (default)	0x0

- **P_SPI0_FIFO_RST (SPI0 FIFO Reset Register)**

Register	Offset		Description	Initial Value
P_SPI0_FIFO_RST	SPI_BA+0x08		SPI0 FIFO Reset Register	0x0000_0004
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31]	Reserved			0x0
Bit[30]	FIFO_RESET	R/W	1: FIFO reset 0: FIFO no reset (default)	0x0
Bit[29:0]	Reserved			0x4

- **P_SPI0_Data (SPI0 Data Register)**

Register	Offset		Description	Initial Value
P_SPI0_Data	SPI_BA+0x10		SPI0 Data Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:0]	FIFO_D	R/W	Write Data into FIFO Read Data from FIFO	0x0

- **P_SPI0_INTEN (SPI0 Interrupt enable)**

Register	Offset		Description	Initial Value
P_SPI0_INTEN	SPI_BA+0x14		SPI0 Interrupt enable	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:1]	Reserved			
Bit[0]	SPI0_INTEN	R/W	1: interrupt enable 0: interrupt disable (default)	0x0

- **P_SPI0_INTF (SPI0 Interrupt flag)**

Register	Offset		Description	Initial Value
P_SPI0_INTF	SPI_BA+0x18		SPI0 Interrupt flag	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:1]	Reserved			
Bit[0]	SPI0_INF	R/W1C	1: have interrupt 0: don't have interrupt (default)	0x0

- **P_SPI1_Ctrl (SPI1 Control Register)**

Register	Offset		Description	Initial Value
P_SPI1_Ctrl	SPI_BA+0x40		SPI1 Control Register	0x0000_0700
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:19]	Reserved			
Bit[18:16]	SPI_NUM	R/W	SPI Txd/Rcv 's Num byte	0x0
Bit[15:11]	Reserved			0x0
Bit[10:8]	SPISEL	R/W	111: SPICK1=SYSCLK/1 110: SPICK1=SYSCLK/2 101: SPICK1=SYSCLK/4 100: SPICK1=SYSCLK/8 011: SPICK1=SYSCLK/16 010: SPICK1=SYSCLK/32 001: SPICK1=SYSCLK/64 000: SPICK1=SYSCLK/128	0x7
Bit[7]	SPI_POL	R/W	1: SPICK stop at 1, when #CS = 1 0: SPICK stop at 0, When #CS = 1	0x0
Bit[6]	SPI_PHA	R/W	1: sampling on second clock edge 0: sampling on first clock edge	0x0
Bit[5:4]	SPI_DATS	R/W	10: SPI use 2 IO 01: SPI use 1 IO (DI connect to DO) 00: SPI use 1 IO	0x0
Bit[3:2]	Reserved			
Bit[1]	TXD	R/W	1: start SEND DATA 0: do nothing	0x0
Bit[0]	RXD	R/W	1: start Receive DATA 0: do nothing	0x0

- **P_SPI1_FIFO_RST (SPI1 FIFO Reset Register)**

Register	Offset		Description	Initial Value
P_SPI1_FIFO_RST	SPI_BA+0x48		SPI1 FIFO Reset Register	0x0000_0004
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31]	Reserved			0x0
Bit[30]	FIFO_RESET	R/W	1: FIFO reset 0: FIFO no reset	0x0
Bit[29:0]	Reserved			0x4

- **P_SPI1_Data (SPI1 Data Register)**

Register	Offset		Description	Initial Value
P_SPI1_Data	SPI_BA+0x50		SPI1 Data Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:0]	FIFO_D	R/W	Write Data into FIFO Read Data from FIFO	0x0

- **P_SPI1_INTEN (SPI1 Interrupt enable)**

Register	Offset		Description	Initial Value
P_SPI1_INTEN	SPI_BA+0x54		SPI1 Interrupt enable	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:1]	Reserved			
Bit[0]	SPI1_INTEN	R/W	1: interrupt enable 0: interrupt disable (default)	0x0

- **P_SPI1_INTF (SPI1 Interrupt flag)**

Register	Offset		Description	Initial Value
P_SPI1_INTF	SPI_BA+0x58		SPI1 Interrupt flag	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:1]	Reserved			
Bit[0]	SPI1_INF	R/W1C	1: have interrupt 0: don't have interrupt (default)	0x0

2.12 UART Interface Controller

2.12.1 Overview and Features

The NX1 provides one UART (Universal Asynchronous Receiver Transmitter) module, UART is a full-duplex, asynchronous communication channel that communicates with peripheral devices and personal computers through protocols such as RS-232.

- Supports 5 to 8 bits per character
- Supports 1, 1.5 and 2 STOP bits
- Supports even, odd and stick parity bits
- Supports programmable baud rate
- Supports parity errors, framing errors and data overrun detection

2.12.2 Block Diagram

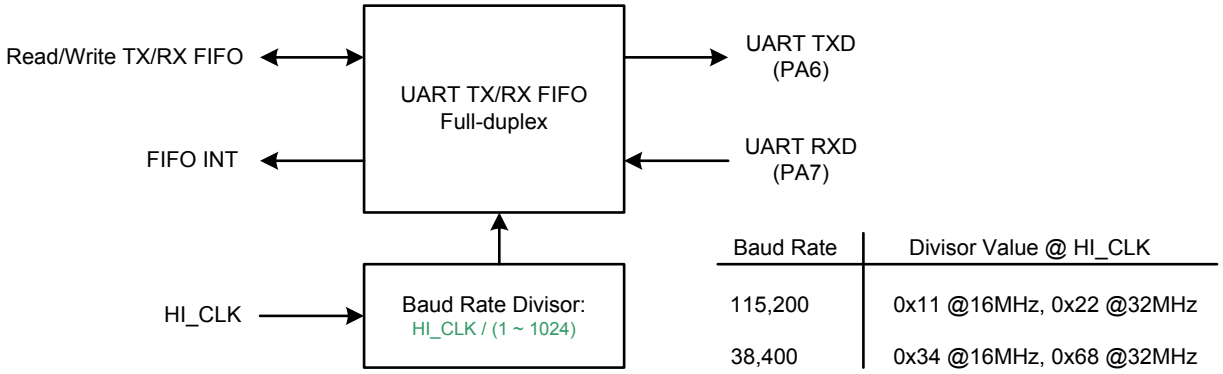


Figure 10 UART Block Diagram

2.12.3 Function Description

User need to enable UART's block by write 1 to P_SMU_FUNC_Ctrl [9] (URCLK_EN) before using UART function.

Baud Rate Generator: Program the Baud Rate registers (P_UART_BaudRate) to select the desired communication baud rate for the UART mode. The baud rate is given by:

$$\text{Baud rate} = \frac{sys\_clk}{(P\_UART\_BaudRate+1)*8}$$

Transmitter: Writes data to TX FIFO via data register (P_UART_Data), IP will start transmitter and stop until FIFO is empty.

Receiver: UART receive data and saves it to FIFO, user can read out the data via data register (P_UART_Data).

UART support six interrupt methods check P_UART0_Ctrl/ P_UART_Ctrl [5:0] for further info.

2.12.4 Register Map

Register	Offset	R/W	Description	Initial Value
UART Base Address: UART_BA = 0xF0_2000, offset = 0x40, n=set number (0,1)				
P_UART_Data	UART_BA+0x00	R/W	UART Data Register	0x0000_0000
P_UART_Ctrl	UART_BA+0x04	R/W	UART Control Register	0x000A_1300
P_UART_Flag	UART_BA+0x08	R/W1C	UART Status Register	0x0000_0022
P_UART_BaudRate	UART_BA+0x0C	R/W	UART Baud Rate Divider Register	0x0000_0022

2.12.5 Register Description

R: Read only, **W**: Write only, **W1C**: Write 1 to clear, **R/W**: Read and Write

- **P_UART_Data (UART Data Register)**

Register	Offset		Description	Initial Value
P_UART_Data	UART_BA+0x00		UART Data Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:0]	Reserved			
Bit[7:0]	UDR	R/W	write to TXD FIFO in WR (THR) read from RXD FIFO in RD (RBR)	0x0

- **P_UART_Ctrl (UART Control Register)**

Register	Offset		Description	Initial Value
P_UART_Ctrl	UART_BA+0x04		UART Control Register	0x000a_1300
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:21]	Reserved			
Bit[19:18]	TX_FIFO_LEVEL	R/W	TX's FIFO level small than level have interrupt	0x2
Bit[17:16]	RX_FIFO_LEVEL	R/W	RX's FIFO level big than level have interrupt	0x2
Bit[15]	T_FIFO_R	W	1: Reset TXD's FIFO 0: no reset	0x0
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[14]	R_FIFO_R	W	1: Reset RXD's FIFO 0: no reset	0x0
Bit[13]	Reserved			0x0
Bit[12]	EP	R/W	1: even parity 0: odd parity	0x1

Bit[11]	PE	R/W	1: one parity bit 0: no parity bits	0x0
Bit[10]	STPS	R/W	1: if WLS=0, 1.5 stop, else 2 stop 0: 1 stop	0x0
Bit[9:8]	WLS	R/W	11: 8 bits 10: 7 bits 01: 6 bits 00: 5 bits	0x3
Bit[7]	TX_DONE_IEN	R/W	1: TX_DONE interrupt disable 0: TX_DONE interrupt enable	0x0
Bit[6]	LSR_IEN	R/W	1: LSR_INT enable 0: LSR_INT disable	0x0
Bit[5]	TX_FIFO_FLAG_IEN	R/W	1: TX_FIFO_FLAG_INT enable 0: TX_FIFO_FLAG_INT disable	0x0
Bit[4]	RX_FIFO_FLAG_IEN	R/W	1: RX_FIFO_FLAG_INT enable 0: RX_FIFO_FLAG_INT disable	0x0
Bit[3]	T_FIFO_FULL_IEN	R/W	1: T_FIFO_FULL_INT enable 0: T_FIFO_FULL_INT disable	0x0
Bit[2]	R_FIFO_FULL_IEN	R/W	1: R_FIFO_FULL_INT enable 0: R_FIFO_FULL_INT disable	0x0
Bit[1]	TX_DONE_IEN	R/W	1: T_DONE_INT enable 0: T_DONE_INT disable	0x0
Bit[0]	R_FIFO_NOEMPTY_IEN	R/W	1: R_FIFO_NOEMPTY_INT enable 0: R_FIFO_NOEMPTY_INT disable	0x0

- **P_UART_Flag (UART Status Register)**

Register	Offset		Description	Initial Value
P_UART_Flag	UART_BA+0x08		UART Status Register	0x0000_0022
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:12]	Reserved			0x0
Bit[11]	FE	R/W1C	1: have framing error (stop not high) 0: no framing error	0x0
Bit[10]	PE	R/W1C	1: have parity error (parity not match) 0: no parity error	0x0
Bit[9]	OE	R/W1C	1: have overrun error (FIFO overrun) 0: no overrun error	0x0
Bit[8]	TX_BUSY	R	1: tx busy (tx fifo no empty | tx send) 0: tx idle	0x0
Bit[7]	TX_DONE_FLAG	R/W1C	1: tx done (tx busy falling & tx fifo empty) 0: don't have tx done	0x0
Bit[6]	Reserved			0x0
Bit[5]	TX_FIFO_FLAG	R	1: TX's FIFO flag is set 0: TX's FIFO flag is not set	0x1
Bit[4]	RX_FIFO_FLAG	R	1: RX's FIFO flag is set 0: RX's FIFO flag is not set	0x0
Bit[3]	TX_FIFO_FULL	R	1: TX FIFO is FULL 0: TX FIFO not FULL (default)	0x0
Bit[2]	RX_FIFO_FULL	R	1: RX FIFO is FULL 0: RX FIFO not FULL (default)	0x0
Bit[1]	TX_FIFO_EMPTY	R/W1C	1: TX FIFO is EMPTY (default) 0: TX FIFO not EMPTY	0x1
Bit[0]	R_FIFO_NOEMPTY	R	1: RX FIFO not EMPTY 0: RX FIFO is EMPTY (default)	0x0

- **P_UART_BaudRate (UART BaudRate Divider Register)**

Register	Offset		Description	Initial Value
P_UART_BaudRate	UART_BA+0x0C		UART Baud Rate Divider Register	0x0000_0022
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:10]	Reserved			
Bit[9:0]	BDIV	R/W	Divisor's Data	0x22

2.13 NFC Controller

2.13.1 Overview and Features

- Allowing the NFC device to read and/or write passive NFC tags and stickers
- Allowing the NFC device to exchange data with other NFC peers; this operation mode is used by Android Beam
- Allowing the NFC device itself to act as an NFC card.

2.13.2 Block Diagram

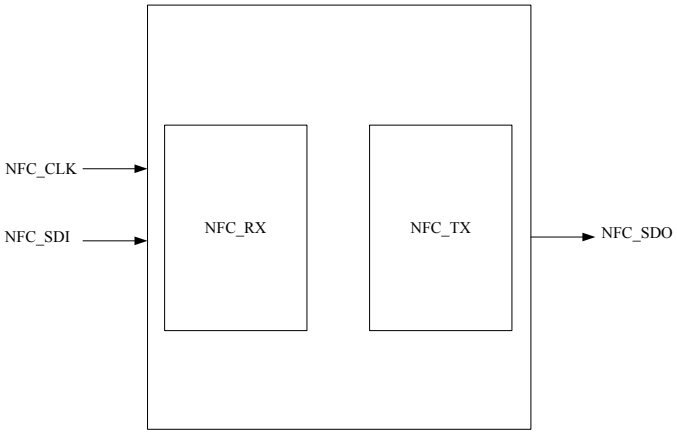


Figure 11 NFC Block Diagram

2.13.3 Function Description

Near Field Communication (NFC) is a set of short-range wireless technologies, typically requiring a distance of 4cm or less to initiate a connection. NFC allows you to share small payloads of data between an NFC tag and a NFC device, or between two NFC devices.

2.13.4 Register Map

Register	Offset	R/W	Description	Initial Value
NFC Base Address: NFC_BA = 0xF0_A000				
P_NFC_Ctrl	NFC_BA+0x00	R/W	NFC Control Register	0x0000_0000
P_NFC_DAT	NFC_BA+0x04	R/W	NFC Data Register	0x0000_0000
P_NFC_INTEN	NFC_BA+0x08	R/W	NFC Interrupt Control Register	0x0000_0000

P_NFC_INTF	NFC_BA+0x0C	R/W	NFC Interrupt FLAG	0x0000_0000
P_NFC_TM	NFC_BA+0x10	R/W	NFC Time Register	0x0000_0000

2.13.5 Register Description

R: Read only, **W**: Write only, **W1C**: Write 1 to clear, **R/W**: Read and Write

- **P_NFC_Ctrl (NFC Control Register)**

Register	Offset		Description	Initial Value
P_NFC_Ctrl	NFC_BA+0x00		NFC Control Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:8]	Reserver			0x0
Bit[7]	T4_EN	R/W	1 : T4 timer enable 0 : T4 timer disable	0x0
Bit[8]	Parity	R	1 : High 0 : Low	
Bit[5]	TX_8BITS	R/W	TX send bit select 1 : 8bits [7:0] 0 : 4bits [3:0]	0x0
Bit[4]	TX_STR	R/W	NFC TX Response Start/Stop 1 : Start to sent TX data 0 : Stop to send TX data after next data sent	0x0
Bit[3]	MODE	R/W	1 : TX mode 0 : RX mode	0x0
Bit[2]	Reserved			0x0
Bit[1]	RX_BUSY	R	NFC RX command Start/Stop 1 : busy 0 : finish	0x0
Bit[0]	NFC_EN	R/W	1 : enable 0 : disable	0x0

- **P_NFC_DAT (NFC Data Register)**

Register	Offset		Description	Initial Value
P_NFC_DATA	NFC_BA+0x04		NFC DATA Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:7]	Reserved			0x0
Bit[7:0]	DATA	R/W	RX / TX Data	0x0

- **P_NFC_INTEN (NFC Interrupt Control Register)**

Register	Offset		Description	Initial Value
P_NFC_INTEN	NFC_BA+0x08		NFC Interrupt Enable Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:3]	Reserved			0x0
Bit[2]	NFC_IDLEINTEN	R/W	1 : interrupt enable 0 : interrupt disable	0x0

Bit[1]	NFC_RXINTEN	R/W	1 : interrupt enable 0 : interrupt disable	0x0
Bit[0]	NFC_TXINTEN	R/W	1 : interrupt enable 0 : interrupt disable	0x0

- **P_NFC_INTF (NFC Interrupt FLAG)**

Register	Offset		Description	Initial Value
P_NFC_INTF	NFC_BA+0x0C		NFC Interrupt flag Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:3]	Reserved			0x0
Bit[2]	NFC_RXTO	R/W1C	1 : have rx timeout flag 0 : no interrupt flag	0x0
Bit[1]	NFC_RXINTF	R/W1C	1 : have interrupt flag 0 : no interrupt flag	0x0
Bit[0]	NFC_TXINTF	R/W1C	1 : have interrupt flag 0 : no interrupt flag	0x0

- **P_NFC_TM (NFC Time Register)**

Register	Offset		Description	Initial Value
P_NFC_Ctrl	NFC_BA+0x10		NFC Control Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:27]	TX_T4S	R/W	TX's T4 timer select , nfc_clk=32/40/48MHz TX T4 Timer = (n+1)@nfc_clk , n=0~31	0x0
Bit[26:24]	Reserved			
Bit[23:16]	TX_T3S	R/W	TX's T3 timer select , nfc_clk=32/40/48MHz TX T3 Timer = (n+1)@nfc_clk , n=0~255	0x0
Bit[15:14]	Reserved			
Bit[13:8]	TX_T2S	R/W	TX's T2 timer select , nfc_clk=32/40/48MHz TX T2 Timer = (n+1)@nfc_clk , n=0~63	
Bit[7:6]	Reserved			
Bit[5:0]	TX_T1S	R/W	TX's T1 timer select , nfc_clk=32/40/48MHz TX T1 Timer = (n+1)@nfc_clk , n=0~63	0x0

2.14 PWM-IO Generator

2.14.1 Overview and Features

The NX1 has 2 sets of PWM-IO groups - PWMA / PWMB, each with four PWM outputs (PWMA0 ~ PWMA3, PWMB0 ~ PWMB3). Each four PWM outputs share a PWM timer, every PWM output has its own duty and output port. Complementary PWM pairs also provided for PWMA0 / PWMA1 and PWMB0 / PWMB1.

- Four PWM outputs share a timer
- 16-bit counter for each timer
- 16-bit PWM duty

2.14.2 Block Diagram

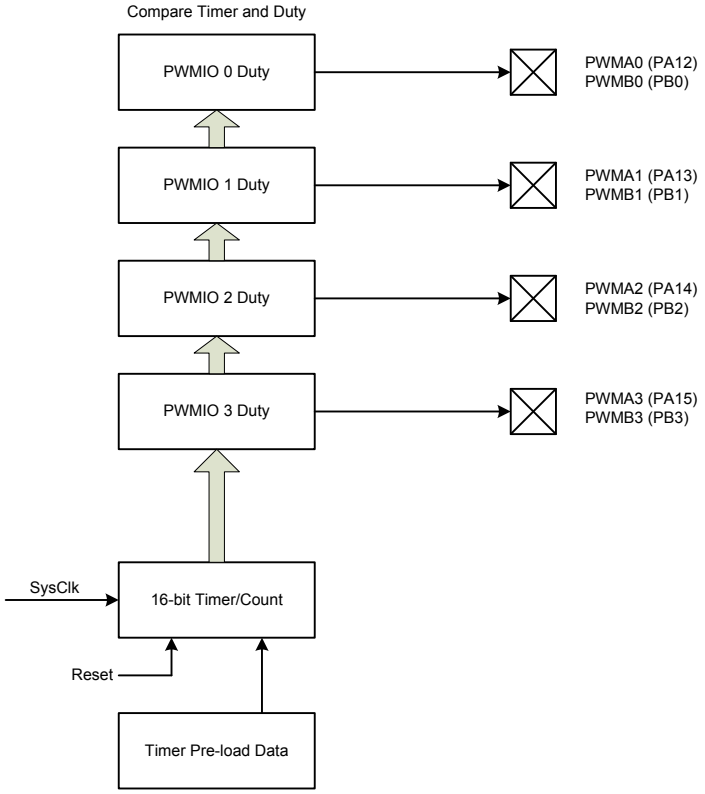


Figure 12 PWM-IO Block Diagram

2.14.3 Function Description

Each PWM-IO generator has its own duty register and four PWM output pins share one timer.

2.14.4 Register Map

Register	Offset	R/W	Description	Initial Value
PWM Base Address: PWM_BA = 0xF1_1000				
P_PWMA_TMR_Ctrl	PWM_BA+0x00	R/W	PWMA Timer Control Register	0x0000_0000
P_PWMA_TMR_Data	PWM_BA+0x04	R/W	PWMA Timer Data Register	0xFFFF_FFFF
P_PWMA_TMR_INT_Ctrl	PWM_BA+0x08	R/W	PWMA Timer Interrupt Enable	0x0000_0000
P_PWMA_TMR_Flag	PWM_BA+0x0C	R/W1C	PWMA Timer Interrupt Flag	0x0000_0000
P_PWMA0_Duty	PWM_BA+0x10	R/W	PWMA0 duty Register	0x0000_0000
P_PWMA1_Duty	PWM_BA+0x14	R/W	PWMA1 duty Register	0x0000_0000
P_PWMA2_Duty	PWM_BA+0x18	R/W	PWMA2 duty Register	0x0000_0000
P_PWMA3_Duty	PWM_BA+0x1C	R/W	PWMA3 duty Register	0x0000_0000
P_PWMB_TMR_Ctrl	PWM_BA+0x40	R/W	PWMB Timer Control Register	0x0000_0000
P_PWMB_TMR_Data	PWM_BA+0x44	R/W	PWMB Timer Data Register	0xFFFF_FFFF
P_PWMB_TMR_INT_Ctrl	PWM_BA+0x48	R/W	PWMB Timer Interrupt Enable	0x0000_0000
P_PWMB_TMR_Flag	PWM_BA+0x4C	R/W1C	PWMB Timer Interrupt Flag	0x0000_0000

Register	Offset	R/W	Description	Initial Value
P_PWMB0_Duty	PWM_BA+0x50	R/W	PWMB0 duty Register	0x0000_0000
P_PWMB1_Duty	PWM_BA+0x54	R/W	PWMB1duty Register	0x0000_0000
P_PWMB2_Duty	PWM_BA+0x58	R/W	PWMB2 duty Register	0x0000_0000
P_PWMB3_Duty	PWM_BA+0x5C	R/W	PWMB3 duty Register	0x0000_0000

2.14.5 Register Description

R: Read only, **W**: Write only, **W1C**: Write 1 to clear, **R/W**: Read and Write

- **P_PWMA_TMR_Ctrl (PWMA Timer Control Register)**

Register	Offset		Description	Initial Value
P_PWMA_TMR_Ctrl	PWM_BA+0x00		PWMA Timer Control Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:20]	Reserved			
Bit[19:16]	Reserved			0x0
Bit[15:8]	Reserved			0x0
Bit[7]	Reserved			
Bit[6:4]	Reserved			0x0
Bit[3]	Reserved			0x0
Bit[2]	Reserved			0x0
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[1]	PWMTMR_STR	W	1: reload data 0: do nothing	0x0
Bit[0]	PWMTMR_EN	R/W	1: PWMTMR enabled 0: PWMTMR disabled	0x0

- **P_PWMA_TMR_Data (PWMA Timer Data Register)**

Register	Offset		Description	Initial Value
P_PWMA_TMR_Data	PWM_BA+0x04		PWMA Timer Control Register	0xFFFF_FFFF
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	PWMTMR_DAT	R	PWMTMR's counter data	0xFFFF
Bit[15:0]	PWMTMR_LDAT	R/W	PWMTMR's preload Data	0xFFFF

- **P_PWMA_TMR_INT_Ctrl (PWMA Timer Interrupt Enable)**

Register	Offset		Description	Initial Value
P_PWMA_TMR_INT_Ctrl	PWM_BA+0x08		PWMA Timer Interrupt Enable	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:1]	Reserved			0x0
Bit[0]	PWMTMR_IEN	R/W	1: TMR's interrupt enabled 0: TMR's interrupt disabled	0x0

- **P_PWMA_TMR_Flag (PWMA Timer Interrupt Flag)**

Register	Offset		Description	Initial Value
P_PWMA_TMR_Flag	PWM_BA+0x0C		PWMA Timer Interrupt Flag	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:1]	Reserved			0x0
Bit[0]	TMR_INTF	R/W1C	1: TMR0's interrupt flag 0: TMR0's interrupt no flag	0x0

- **P_PWMA0_Duty (PWMA0 Duty Register)**

Register	Offset		Description	Initial Value
P_PWMA0_Duty	PWM_BA+0x10		PWMA IO0's duty Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	PWMA0_DUTY	R/W	PWMA0 duty data	0x0
Bit[15:2]	Reserved			0x0
Bit[1]	PWMA0_DSB	R/W	1: Sink (PWM start 1) 0: Drive (PWM start 0)	0x0
Bit[0]	PWMA0_EN	R/W	1: PWMA0 enabled 0: PWMA0 disabled	0x0

- **P_PWMA1_Duty (PWMA IO1's duty Register)**

Register	Offset		Description	Initial Value
P_PWMA1_Duty	PWM_BA+0x14		PWMA1 duty Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	PWMA1_DUTY	R/W	PWMA1 duty data	0x0
Bit[15:2]	Reserved			0x0
Bit[1]	PWMA1_DSB	R/W	1: Sink (PWM start 1) 0: Drive (PWM start 0)	0x0
Bit[0]	PWMA1_EN	R/W	1: PWMA1 enabled 0: PWMA1 disabled	0x0

- **P_PWMA2_Duty (PWMA2 duty Register)**

Register	Offset		Description	Initial Value
P_PWMA2_Duty	PWM_BA+0x18		PWMA2 duty Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	PWMA2_DUTY	R/W	PWMA2 duty data	0x0
Bit[15:2]	Reserved			0x0
Bit[1]	PWMA2_DSB	R/W	1: Sink (PWM start 1) 0: Drive (PWM start 0)	0x0
Bit[0]	PWMA2_EN	R/W	1: PWMA2 enabled 0: PWMA2 disabled	0x0

- **P_PWMA3_Duty (PWMA3 duty Register)**

Register	Offset		Description	Initial Value
P_PWMA3_Duty	PWM_BA+0x1C		PWMA3 duty Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	PWMA3_DUTY	R/W	PWMA3 duty data	0x0
Bit[15:2]	Reserved			0x0
Bit[1]	PWMA3_DSB	R/W	1: Sink (PWM start 1) 0: Drive (PWM start 0)	0x0

Bit[0]	PWMA3_EN	R/W	1: PWMA3 enabled 0: PWMA3 disabled	0x0

- **P_PWMB_TMR_Ctrl (PWMB Timer Control Register)**

Register	Offset		Description	Initial Value
P_PWMB_TMR_Ctrl	PWM_BA+0x40		PWMB Timer Control Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:20]	Reserved			
Bit[19:16]	Reserved			0x0
Bit[15:8]	Reserved			0x0
Bit[7]	Reserved			
Bit[6:4]	Reserved			0x0
Bit[3]	Reserved			0x0
Bit[2]	Reserved			0x0
Bit[1]	PWMTMR_STR	W	1: reload data 0: do nothing	0x0
Bit[0]	PWMTMR_EN	R/W	1: PWMTMR enabled 0: PWMTMR disabled	0x0

- **P_PWMB_TMR_Data (PWMB Timer Data Register)**

Register	Offset		Description	Initial Value
P_PWMB_TMR_Data	PWM_BA+0x44		PWMB Timer Control Register	0xFFFF_FFFF
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	PWMTMR_DAT	R	PWMTMR's counter data	0xFFFF
Bit[15:0]	PWMTMR_LDAT	R/W	PWMTMR's preload Data	0xFFFF

- **P_PWMB_TMR_INT_Ctrl (PWMB Timer Interrupt Enable)**

Register	Offset		Description	Initial Value
P_PWM1_TMR_INT_Ctrl	PWM_BA+0x48		PWM1 Timer Interrupt Enable	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:1]	Reserved			0x0
Bit[0]	PWMTMR_IEN	R/W	1: TMR's interrupt enable 0: TMR's interrupt disable	0x0

- **P_PWMB_TMR_Flag (PWMB Timer Interrupt Flag)**

Register	Offset		Description	Initial Value
P_PWMB_TMR_Flag	PWM_BA+0x4C		PWMB Timer Interrupt Flag	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:1]	Reserved			0x0
Bit[0]	TMR_INTF	R/W1C	1: TMR1's interrupt flag 0: TMR1's interrupt no flag	0x0

- **P_PWMB0_Duty (PWMB0 duty Register)**

Register	Offset	Description	Initial Value
P_PWMB0_Duty	PWM_BA+0x50	PWMB0 duty Register	0x0000_0000

Bit	Name	R/W	Descriptions	Initial Value
Bit[31:16]	PWMB0_DUTY	R/W	PWMB0 duty data	0x0
Bit[15:2]	Reserved			0x0
Bit[1]	PWMB0_DSB	R/W	1: Sink (PWM start 1) 0: Drive (PWM start 0)	0x0
Bit[0]	PWMB0_EN	R/W	1: PWMB0 enabled 0: PWMB0 disabled	0x0

- **P_PWMB1_Duty (PWMB1 duty Register)**

Register	Offset		Description	Initial Value
P_PWMB1_Duty	PWM_BA+0x54		PWMB1 duty Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	PWMB1_DUTY	R/W	PWMB1 duty data	0x0
Bit[15:2]	Reserved			0x0
Bit[1]	PWMB1_DSB	R/W	1: Sink (PWM start 1) 0: Drive (PWM start 0)	0x0
Bit[0]	PWMB1_EN	R/W	1: PWMB1 enabled 0: PWMB1 disabled	0x0

- **P_PWMB2_Duty (PWMB2 duty Register)**

Register	Offset		Description	Initial Value
P_PWMB2_Duty	PWM_BA+0x58		PWMB2 duty Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	PWMB2_DUTY	R/W	PWMB2 duty data	0x0
Bit[15:2]	Reserved			0x0
Bit[1]	PWMB2_DSB	R/W	1: Sink (PWM start 1) 0: Drive (PWM start 0)	0x0
Bit[0]	PWMB2_EN	R/W	1: PWMB2 enabled 0: PWMB2 disabled	0x0

- **P_PWMB3_Duty (PWMB3 duty Register)**

Register	Offset		Description	Initial Value
P_PWMB3_Duty	PWM_BA+0x5C		PWMB3 duty Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	PWMB3_DUTY	R/W	PWMB3 duty data	0x0
Bit[15:2]	Reserved			0x0
Bit[1]	PWMB3_DSB	R/W	1: Sink (PWM start 1) 0: Drive (PWM start 0)	0x0
Bit[0]	PWMB3_EN	R/W	1: PWMB3 enabled 0: PWMB3 disabled	0x0

2.15 Flash Memory Control (FMC)

2.15.1 Features

- Support to program embedded flash, while running program in embedded flash
- Support to sector erase embedded flash, while running program in embedded flash

2.15.2 Block Diagram

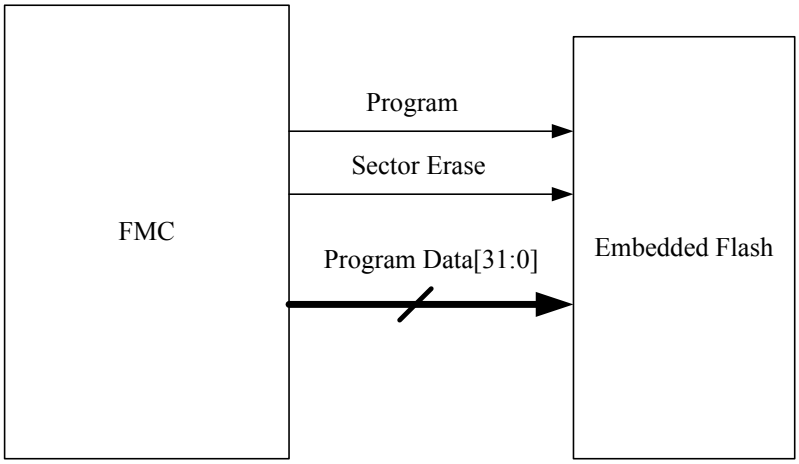


Figure 13 FMC Block Diagram

2.15.3 Function Description

It can program/sector erase to the address in embedded flash. And generate interrupt flag to show this command is finish, or error.

2.15.4 Register Map

Register	Offset	R/W	Description	Initial Value
FMC Base Address : FMC_BA = 0xF0_9000				
P_FMC_CTL	FMC_BA+0x00	R/W	Flash control register	0x00000000
P_FMC_ADDR	FMC_BA+0x04	R/W	Flash control address register	0x00000000
P_FMC_DATA	FMC_BA+0x08	R/W	Flash control data register	0x00000000
P_FMC_INTEN	FMC_BA+0x10	R/W	Flash interrupt enable triger	0x00000000
P_FMC_INTF	FMC_BA+0x14	R/W	Flash interrupt flag triger	0x00000000
P_FMC_TIME0	FMC_BA+0x18	R/W	Adjust Flash Program/ERASE Time	0x000A071D
P_FMC_TIME1	FMC_BA+0x1C	R/W	Adjust Recovery Time & Program Setup Time	0x00002960

2.15.5 Register Description

- **P_FMC_CTL**

Register	Offset		Description	Initial Value
P_FMC_CTL	FMC_BA+0x00		Flash control register	
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**

Bit[31:8]	Reserved			0x0
Bit[7:0]	FCTL	R/W	flash control mode 0x00 : reserved 0x01 : main flash word program 0x02 : main flash sector erase	0xFF

- **P_FMC_ADDR**

Register	Offset		Description	Initial Value
P_FMC_ADDR	FMC_BA+0x04		Flash control address register	
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31]	INC	R/W	address inc after trigger finish	0x0
Bit[30:21]	Reserved			0x0
Bit[20:2]	ADDR	R/W	flash address	0xuuu_uuuu
Bit[1:0]	Reserved			0x0

- **P_FMC_DATA**

Register	Offset		Description	Initial Value
P_FMC_DATA(Flash data)	FMC_BA+0x08		Flash control data register	
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:0]	DATA	R/W	Flash data	0xuuu_uuuu

- **P_FMC_INTEN**

Register	Offset		Description	Initial Value
P_FMC_INTEN	FMC_BA+0x10		Flash interrupt enable trigger	
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:1]	Reserved			
Bit[0]	INTEN	R/W	1: interrupt enable 0 : interrupt disable	0x0

- **P_FMC_INTF**

Register	Offset		Description	Initial Value
P_FMC_INTF	FMC_BA+0x14		Flash interrupt flag triger	
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:2]	Reserved			
Bit[1]	ERRF	R/W1C	1 : have error 0 : no error	0x0
Bit[0]	INTF	R/W1C	1 : have finish flag 0 : no transfer flag	0x0

- **P_FMC_TIME0**

Register	Offset		Description	Initial Value
P_FMC_TIME0	FMC_BA+0x18			
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:21]	Reserved			
Bit[20:16]	PROGRAM	R	Adjust Program Time	0xA
Bit[15:13]	Reserved			
Bit[12:8]	SECTOR_ERASE	R	Adjust Sector Erase Time	0x07
Bit[7:4]	Reserved			
Bit[5:0]	CHIP_ERASE	R	Adjust Chip Erase Time	0x1D

- **P_FMC_TIME1**

Register	Offset		Description	Initial Value
P_FMC_TIME1	FMC_BA+0x1C			
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	Reserved			
Bit[15:14]	Reserved			
Bit[13:8]	RCV	R	Adjust Recovery Time	0x29
Bit[7]	Reserved			
Bit[6:0]	PGS	R	Adjust Program Setup Time	0x60

2.16 Real Time Clock (RTC)

2.16.1 Overview and Features

The RTC support periodic time tick interrupts with 4 options: 16.384KHz, 1.024KHz, 64Hz, and 2Hz.

2.16.2 Block Diagram

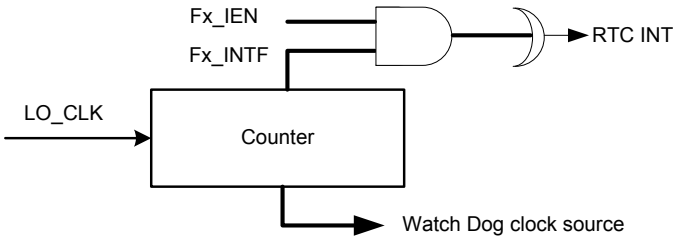


Figure 14 RTC Block Diagram

2.16.3 Function Description

Real Time Clock (RTC) provides fixed / regular interrupt signals for time-keeping applications. The clock source is the internal built-in I_LRC. The RTC support periodic time tick interrupts with 4 options: 16.384KHz, 1.024KHz, 64Hz, and 2Hz.

2.16.4 Register Map

R: Read only, **W**: Write only, **W1C**: Write 1 to clear, **R/W**: Read and Write

Register	Offset	R/W	Description	Initial Value
RTC Base Address: RTC_BA = 0xF0_6000				
P_RTC_INT_Ctrl	RTC_BA+0x00	R/W	RTC Interrupt Enable Register	0x0000_0000
P_RTC_Flag	RTC_BA+0x04	R/W1C	RTC Interrupt Flag Register	0x0000_0000
P_RTC_CLR	RTC_BA+0x08	W	when CLR_RTC ==16'3CA5, clear RTC & WDT counter	0x0000_0000

2.16.5 Register Description

R: Read only, **W**: Write only, **W1C**: Write 1 to clear, **R/W**: Read and Write

- **P_RTC_INT_Ctrl (RTC Interrupt Enable Register)**

Register	Offset		Description	Initial Value
P_RTC_INT_Ctrl	RTC_BA+0x00		RTC Interrupt Enable Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:4]	Reserved			0x0
Bit[3]	F16KHZ_IEN	R/W	1: interrupt enable 0: interrupt disable	0x0
Bit[2]	F1KHZ_IEN	R/W	1: interrupt enable 0: interrupt disable	0x0
Bit[1]	F64HZ_IEN	R/W	1: interrupt enable 0: interrupt disable	0x0

Bit[0]	F2HZ_IEN	R/W	1: interrupt enable 0: interrupt disable	0x0

- **P_RTC_Flag (RTC Interrupt Flag Register)**

Register	Offset		Description	Initial Value
P_RTC_Flag	RTC_BA+0x04		RTC Interrupt Flag Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:4]	Reserved			0x0
Bit[3]	F16KHZ_INTF	R/W1C	1: have interrupt flag 0: no interrupt flag	0x0
Bit[2]	F1KHZ_INTF	R/W1C	1: have interrupt flag 0: no interrupt flag	0x0
Bit[1]	F64HZ_INTF	R/W1C	1: have interrupt flag 0: no interrupt flag	0x0
Bit[0]	F2HZ_INTF	R/W1C	1: have interrupt flag 0: no interrupt flag	0x0

- **P_RTC_CLR (RTC Clear Register)**

Register	Offset		Description	Initial Value
P_RTC_CLR	RTC_BA+0x08		RTC & WDT Clear Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	Reserved			0x0
Bit[15:0]	CLR_RTC_WDT	W	when CLR_RTC ==16'3CA5, clear RTC & WDT counter	0x0

2.17 Watchdog Timer (WDT)

2.17.1 Overview and Features

- Counter
- Programmable period register

2.17.2 Block Diagram

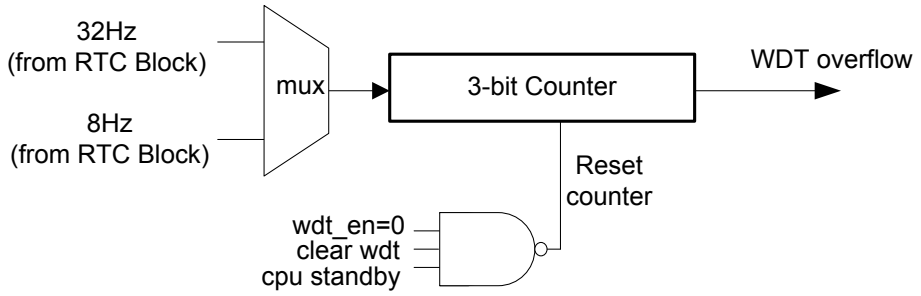


Figure 15 WDT Block Diagram

2.17.3 Function Description

The Watchdog Timer (WDT) is used to perform a system reset when the system is not responding as expected. It would prevent the system from hanging-up for an infinite period.

2.17.4 Register Map

R: Read only, **W**: Write only, **R/W**: Read and Write

Register	Offset	R/W	Description	Initial Value
WDT Base Address: WDT_BA = 0xF0_7000				
P_WDT_Ctrl	RTC_BA+0x00	R/W	WDT Control Register	0x0000_0000
P_WDT_CLR	RTC_BA+0x04	W	WDT Clear Register	0x0000_0000

2.17.5 Register Description

R: Read only, **W**: Write only, **W1C**: Write 1 to clear, **R/W**: Read and Write

- **P_WDT_Ctrl (WDT Control Register)**

Register	Offset		Description	Initial Value
P_WDT_Ctrl	RTC_BA+0x00		WDT Control Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:1]	Reserved			
Bit[0]	WDT_TIME	R/W	1: 188ms ~ 219ms (= 1/32Hz * 6 ~ 7) 0: 750ms ~ 875ms (= 1/8Hz * 6 ~ 7)	0x0

- **P_WDT_CLR (WDT Clear Register)**

Register	Offset		Description	Initial Value
P_WDT_CLR	RTC_BA+0x04		WDT Clear Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	Reserved			0x0
Bit[15:0]	CLR_WDT	W	When CLR_WDT = 0x3CA5, clear WDT counter	0x0

2.18 Analog-to-Digital Converter

2.18.1 Overview and Features

The NX12F_NX13F provides one 12-bit Analog-to-Digital converter with eight input channels. The A/D converter supports single and continuous scan mode. It can be started by software or TIMER trigger.

- Provide 4 level FIFO or data registers for each channel.
- Auto scan mode.

2.18.2 Block Diagram

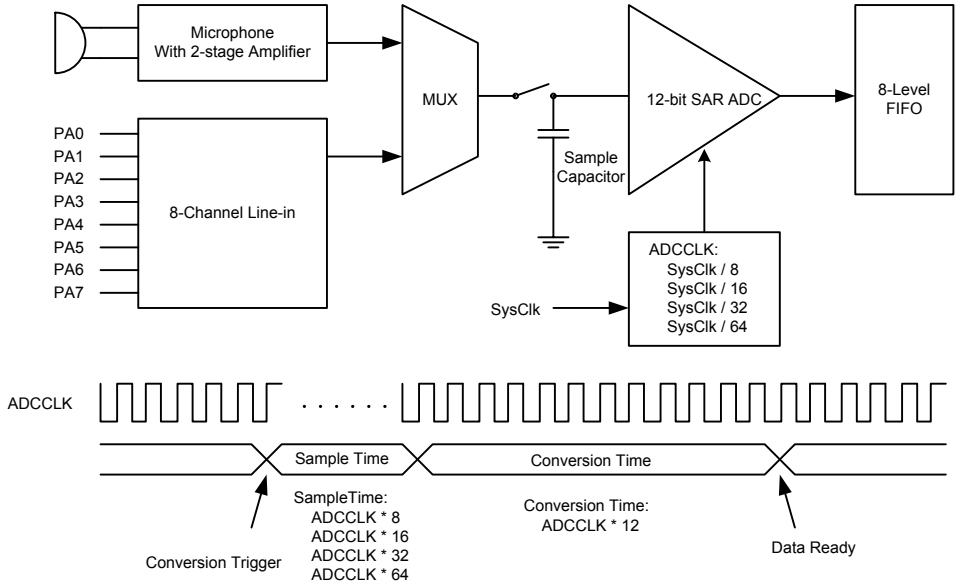


Figure 16 ADC Block Diagram

2.18.3 Function Description

The NX12F_NX13F provides 4 trigger sources and 9 analog input signals as shown in Figure 16 ADC Block Diagram.

Data register: the NX12F_NX13F provides 2 ways to store the data, one is to save all channels' data to FIFO; the other way is to save data to individual addresses.

Scan mode: the NX12F_NX13F supports scan modes that can auto get 4/3/2 channel's data by just one trigger.

2.18.4 Register Map

Register	Offset	R/W	Description	Initial Value
ADC Base Address: ADC_BA = 0xF1_4000				
P_ADC_Ctrl	ADC_BA+0x00	R/W	ADC Control Register	0x0000_E000
P_ADC_AGC_Ctrl	ADC_BA+0x04	R/W	AGC Control Register	0x0000_00C0
P_ADC_FIFO_Ctrl	ADC_BA+0x08	R/W	ADC FIFO Control Register	0x0000_0040
P_ADC_Flag	ADC_BA+0x0C	R/W	ADC FIFO Status Register	0x0000_0000
P_ADC_Data_CH0	ADC_BA+0x10	R	ADC Channel 0 Register	0x0000_0000
P_ADC_Data_CH1	ADC_BA+0x14	R	ADC Channel 1 Register	0x0000_0000
P_ADC_Data_CH2	ADC_BA+0x18	R	ADC Channel 2 Register	0x0000_0000
P_ADC_Data_CH3	ADC_BA+0x1C	R	ADC Channel 3 Register	0x0000_0000
P_ADC_Data_CH4	ADC_BA+0x20	R	ADC Channel 4 Register	0x0000_0000
P_ADC_Data_CH5	ADC_BA+0x24	R	ADC Channel 5 Register	0x0000_0000
P_ADC_Data_CH6	ADC_BA+0x28	R	ADC Channel 6 Register	0x0000_0000
P_ADC_Data_CH7	ADC_BA+0x2C	R	ADC Channel 7 Register	0x0000_0000

2.18.5 Register Description

R: Read only, **W**: Write only, **W1C**: Write 1 to clear, **R/W**: Read and Write

- **P_ADC_Ctrl (ADC Control Register)**

Register	Offset		Description	Initial Value
P_ADC_Ctrl	ADC_BA+0x00		ADC Control Register	0x0000_E000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:15]	Reserved			0x0
Bit[15:14]	EN_4CLK	R/W	11: sample 8 * ANCK 10: sample 16 * ANCK 01: sample 32 * ANCK 00: sample 64 * ANCK	0x3
Bit[13]	CHG_EN	R/W	1: enable channel pre-charged to ½*VDD 0: disable channel pre-charged to ½*VDD	0x1
Bit[12]	ADC_EN	R/W	1: ADC enabled 0: ADC disabled	0x0
Bit[11]	Reserved			0x0
Bit[10]	SOFTWARE_TRIG	W		0x0
Bit[9:8]	ANSEL	R/W	11: ANCK=SYSCLK/(2*4) 10: ANCK=SYSCLK/(4*4) 01: ANCK=SYSCLK/(8*4) 00: ANCK=SYSCLK/(16*4)	0x0
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[7:6]	SCAN_MODE	R/W	11: CH0/MIC -> CH1 -> CH0/MIC -> CH2 -> CH0/MIC -> CH3 loop 10: CH0/MIC -> CH1 -> CH0/MIC -> CH2 loop 01: CH0/MIC -> CH1 loop 00: SCAN MODE disabled	0x0
Bit[5:4]	ADC_TRIG_SEL	R/W	11: Timer2 Triggered 10: Timer1 Triggered 01: Timer0 Triggered 00: Software Triggered	0x0
Bit[3]	MIC_SEL	R/W	1: MIC channel selected 0: CH0 channel selected	0x0
Bit[2:0]	ADC_CHS	R/W	111: CH7 110: CH6 101: CH5 100: CH4 011: CH3 010: CH2 001: CH1 000: CH0 / MIC	0x0

- **P_ADC_AGC_Ctrl (AGC Control Register)**

Register	Offset		Description	Initial Value
P_ADC_AGC_Ctrl	ADC_BA+0x04		AGC Control Register	0x0000_00C0
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:15]	Reserved			0x0
Bit[14:12]	AGC_CKSEL	R/W	N = 0 ~ 6: AGCCLK = LOCLK/(2*(N+1)) N = 7: AGCCLK = LOCLK	0x0
Bit[11:9]	Reserved			0x0
Bit[8:4]	PGA_GAIN	R/W	PGA gain selection	0x11
Bit[3]	AGC_SEL	R/W	1: AGC enabled 0: PGA enabled	0x0

Bit[2]	MAX_GAIN	R/W	1: mic gain 30 times 0: mic gain 15 times	0x0
Bit[1]	Reserved			0x0
Bit[0]	MIC_EN	R/W	1: V_MIC power supply enabled 0: V_MIC power supply disabled	0x0

- **P_ADC_FIFO_Ctrl (ADC FIFO Control Register)**

Register	Offset		Description	Initial Value
P_ADC_FIFO_Ctrl	ADC_BA+0x08		ADC FIFO Control Register	0x0000_0040
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:5]	Reserved			0x0
Bit[9]	Reserved			0x0
Bit[8]	AFIFO_RESET	W	1: FIFO reset 0: FIFO no reset	0x0
Bit[7]	ADLOOP_IEN	R/W	1: ADC scan loop finish interrupt enable 0: ADC scan loop finish interrupt disable	0x0
Bit[6]	Reserved			0x0
Bit[5:4]	AFIFO_LEVEL	R/W	When FIFO number is big than level, set FIFO_FLAG	0x2
Bit[3]	ADIF_FLAG_IEN	R/W	1: ADC convert finish INT enabled 0: ADC convert finish INT disabled	0x0
Bit[2]	AFIFO_FLAG_IEN	R/W	1: AFIFO_FLAG_INT enabled 0: AFIFO_FLAG_INT disabled	0x0
Bit[1]	AFIFO_EMPTY_IEN	R/W	1: AFIFO_EMPTY_INT enabled 0: AFIFO_EMPTY_INT disabled	0x0
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[0]	AFIFO_FULL_IEN	R/W	1: AFIFO_FULL_INT enabled 0: AFIFO_FULL_INT disabled	0x0

- **P_ADC_Flag (ADC FIFO Status Register)**

Register	Offset		Description	Initial Value
P_ADC_Flag	ADC_BA+0x0C		ADC FIFO Status Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:4]	Reserved			0x0
Bit[7]	ADLOOP_FLAG	R/W	1: ADC scan loop finish flag 0: No ADC scan loop finish flag (default)	0x0
Bit[6]	Reserved			
Bit[5]	ADC_ECO / CMPO	R	1: ADC transfer finish 0: ADC transfer not finish	0x0
Bit[4]	ADC_BUSY	R	1: ADC is BUSY 0: ADC is IDLE (default)	0x0
Bit[3]	ADIF_FLAG	R/W	1: ADC convert finish flag 0: no ADC convert finish flag (default)	0x0
Bit[2]	AFIFO_FLAG	R/W1C	1: FIFO flag set 0: no FIFO flag	0x0
Bit[1]	AFIFO_EMPTY	R/W1C	1: FIFO empty flag set 0: no FIFO empty flag	0x0
Bit[0]	AFIFO_FULL	R/W1C	1: FIFO full flag set 0: no FIFO full flag	0x0

- **P_ADC_Data_CH0 (ADC Channel 0 Register)**

Register	Offset	Description	Initial Value
P_ADC_Data_CH0	ADC_BA+0x10	ADC Channel 0 Register	0x0000_0000

Bit	Name	R/W	Descriptions	Initial Value
Bit[31:12]	Reserved			0x0
Bit[15:4]	CH0_DAT	R	ADC's Channel 0 Register	0x0
Bit[3:0]	Reserved			0x0

- **P_ADC_Data_CH1 (ADC Channel 1 Register)**

Register	Offset		Description	Initial Value
P_ADC_Data_CH1	ADC_BA+0x14		ADC Channel 1 Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:12]	Reserved			0x0
Bit[15:4]	CH1_DAT	R	ADC's Channel 1 Register	0x0
Bit[3:0]	Reserved			0x0

- **P_ADC_Data_CH2 (ADC Channel 2 Register)**

Register	Offset		Description	Initial Value
P_ADC_Data_CH2	ADC_BA+0x18		ADC Channel 2 Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:12]	Reserved			0x0
Bit[15:4]	CH2_DAT	R	ADC's Channel 2 Register	0x0
Bit[3:0]	Reserved			0x0

- **P_ADC_Data_CH3 (ADC Channel 3 Register)**

Register	Offset		Description	Initial Value
P_ADC_Data_CH3	ADC_BA+0x1C		ADC Channel 3 Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:12]	Reserved			0x0
Bit[15:4]	CH3_DAT	R	ADC's Channel 3 Register	0x0
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[3:0]	Reserved			0x0

- **P_ADC_Data_CH4 (ADC Channel 4 Register)**

Register	Offset		Description	Initial Value
P_ADC_Data_CH4	ADC_BA+0x20		ADC Channel 4 Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:12]	Reserved			0x0
Bit[15:4]	CH4_DAT	R	ADC's Channel 4 Register	0x0
Bit[3:0]	Reserved			0x0

- **P_ADC_Data_CH5 (ADC Channel 5 Register)**

Register	Offset		Description	Initial Value
P_ADC_Data_CH5	ADC_BA+0x24		ADC Channel 5 Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:12]	Reserved			0x0
Bit[15:4]	CH5_DAT	R	ADC's Channel 5 Register	0x0
Bit[3:0]	Reserved			0x0

- **P_ADC_Data_CH6 (ADC Channel 6 Register)**

Register	Offset		Description	Initial Value
P_ADC_Data_CH6	ADC_BA+0x28		ADC Channel 6 Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:12]	Reserved			0x0
Bit[15:4]	CH6_DAT	R	ADC's Channel 6 Register	0x0
Bit[3:0]	Reserved			0x0

- **P_ADC_Data_CH7 (ADC Channel 7 Register)**

Register	Offset		Description	Initial Value
P_ADC_Data_CH7	ADC_BA+0x2C		ADC Channel 7 Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:12]	Reserved			0x0
Bit[15:4]	CH7_DAT	R	ADC's Channel 7 Register	0x0
Bit[3:0]	Reserved			0x0

2.19 Digital-to-Analog Converter and Delta-Sigma PWM

2.19.1 Overview and Features

The NX12F_NX13F provides two DAC channels with interpolation function. It can be started by software or TIMER trigger.

- Provide 4-level FIFO
- Provide hardware interpolation function

2.19.2 Block Diagram

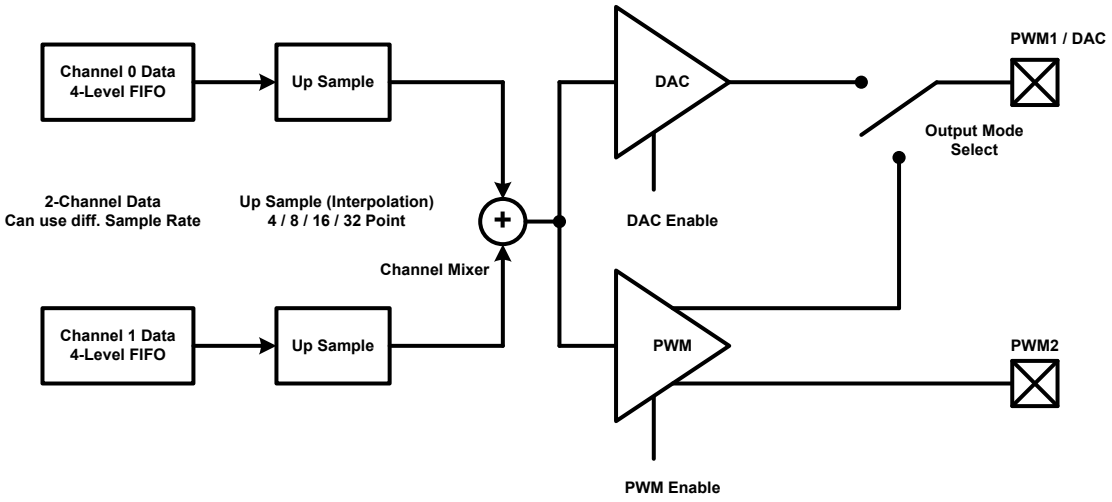


Figure 17 Block Diagram of DAC and PWM PA

2.19.3 Function Description

The NX12F_NX13F supports 4 trigger sources for DAC.

ITP (InTerPolation): Enable ITP also requires the speed-up of timer at the same time (by N times). The hardware ITP will generate another N-1 data points between 2 consecutive FIFO data. N=4/8/16/32

2.19.4 Register Map

Register	Offset	R/W	Description	Initial Value
DAC Base Address: DAC_BA = 0xF1_5000				
P_DAC_CH0_Ctrl	DAC_BA+0x00	R/W	DAC0 Control Register	0x0000_0800
P_DAC_CH0_Data	DAC_BA+0x04	W	DAC0 FIFO Data Register	0x0000_0000
P_DAC_CH0_FIFO	DAC_BA+0x08	R/W	DAC0 FIFO Control Register	0x0000_0040
P_DAC_CH0_Flag	DAC_BA+0x0C	R/W1C	DAC0 FIFO Status Register	0x0000_0006
P_DAC_CH1_Ctrl	DAC_BA+0x40	R/W	DAC1 Control Register	0x0000_0000
P_DAC_CH1_Data	DAC_BA+0x44	W	DAC1 FIFO Data Register	0x0000_0000
P_DAC_CH1_FIFO	DAC_BA+0x48	R/W	DAC1 FIFO Control Register	0x0000_0040
P_DAC_CH1_Flag	DAC_BA+0x4C	R/W1C	DAC1 FIFO Status Register	0x0000_0006
P_DAC_PWM_CTL	DAC_BA+0x50	R/W	PWM control Register	0x0000_0002

2.19.5 Register Description

R: Read only, **W**: Write only, **W1C**: Write 1 to clear, **R/W**: Read and Write

- **P_DAC_CH0_Ctrl (DAC0 Control Register)**

Register	Offset		Description	Initial Value
P_DAC_CH0_Ctrl	DAC_BA+0x00		DAC0 Control Register	0x0000_0800
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:14]	Reserved			0x0
Bit[13:12]	ITP	R/W	11: 32 point 10: 16 point 01: 8 point 00: 4 point	
Bit[11:8]	Reserved			0x0
Bit[7]	Reserved			0x0
Bit[6]	TRIG_ALL	R/W	1: DAC0's data trig by manually or DAC_TRIG_SEL 0: DAC0's data trig only by manually	0x0
Bit[5:4]	DAC_TRIG_SEL	R/W	11: Latch DATA to DAC by Timer2 10: Latch DATA to DAC by Timer1 01: Latch DATA to DAC by Timer0 00: Manual Latch	0x0
Bit[3]	ITP_EN	R/W	1: ITP enabled 0: ITP disabled	0x0
Bit[2]	Reserved			0x0
Bit[1]	Reserved			0x0

Bit[0]	DAC_EN	R/W	1: Function enabled 0: Function disabled	0x0

- **P_DAC_CH0_Data (DAC0 FIFO Data Register)**

Register	Offset		Description	Initial Value
P_DAC_CH0_Data	DAC_BA+0x04		DAC0 FIFO Data Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:16]	Reserved			0x0
Bit[15:2]	DAC_DATA		DAC's DATA	0x0
Bit[1:0]	Reserved			0x0

- **P_DAC_CH0_FIFO (DAC0 FIFO Control Register)**

Register	Offset		Description	Initial Value
P_DAC_CH0_FIFO	DAC_BA+0x08		DAC0 FIFO Control Register	0x0000_0040
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:9]	Reserved			0x0
Bit[8]	DFIFO_RESET	W	1: FIFO reset 0: FIFO no reset (default)	0x0
Bit[7]	Reserved			0x0
Bit[6:4]	DFIFO_LEVEL	R/W	When FIFO number is smaller than level, set FIFO_FLAG (default=4)	0x4
Bit[3]	Reserved			0x0
Bit[2]	DFIFO_FLAG_IEN	R/W	1: DFIFO_FLAG_INT enabled 0: DFIFO_FLAG_INT disabled	0x0
Bit[1]	DFIFO_EMPTY_IEN	R/W	1: DFIFO_EMPTY_INT enabled 0: DFIFO_EMPTY_INT disabled	0x0
Bit[0]	DFIFO_FULL_IEN	R/W	1: DFIFO_FULL_INT enabled 0: DFIFO_FULL_INT disabled	0x0

- **P_DAC_CH0_Flag (DAC0 FIFO Status Register)**

Register	Offset		Description	Initial Value
P_DAC_CH0_Flag	DAC_BA+0x0C		DAC0 FIFO Status Register	0x0000_0006
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:3]	Reserved			0x0
Bit[2]	DFIFO_FLAG	R/W1C	1: FIFO flag set 0: no FIFO flag	0x1
Bit[1]	DFIFO_EMPTY	R/W1C	1: FIFO empty flag set 0: no FIFO empty flag	0x1
Bit[0]	DFIFO_FULL	R/W1C	1: FIFO full flag set 0: no FIFO full flag	0x1

- **P_DAC_CH1_Ctrl (DAC1 Control Register)**

Register	Offset		Description	Initial Value
P_DAC_CH1_Ctrl	DAC_BA+0x40		DAC1 Control Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:14]	Reserved			0x0
Bit[13:12]	ITP	R/W	11: 32 point 10: 16 point 01: 8 point 00: 4 point	0x0
Bit[11:8]	Reserved			

Bit[7]	MIX_EN	R/W	1: DAC0 data and DAC1 data mix 0: DAC0 data and DAC1 data no mix	0x0
Bit[6]	TRIG_ALL	R/W	1: DAC0's data trig by manually or DAC_TRIG_SEL 0: DAC0's data trig only by manually	0x0
Bit[5:4]	DAC_TRIG_SEL	R/W	11: Latch DATA to DAC by Timer2 10: Latch DATA to DAC by Timer1 01: Latch DATA to DAC by Timer0 00: Manual Latch	0x0
Bit[3]	ITP_EN	R/W	1: ITP enabled 0: ITP disabled	0x0
Bit[2:1]	Reserved			0x0
Bit[0]	DAC_EN	R/W	1: Function enabled 0: Function disabled	0x0

- **P_DAC_CH1_Data (DAC1 FIFO Data Register)**

Register	Offset		Description	Initial Value
P_DAC_CH1_Data	DAC_BA+0x44		DAC1 FIFO Data Register	0x0000_0000
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:18]	Reserved			0x0
Bit[17:16]	Reserved			0x0
Bit[15:2]	DAC_DATA	W	DAC's DATA	0x0
Bit[1:0]	Reserved			0x0

- **P_DAC_CH1_FIFO (DAC1 FIFO Control Register)**

Register	Offset		Description	Initial Value
P_DAC_CH1_FIFO	DAC_BA+0x48		DAC1 FIFO Control Register	0x0000_0040
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:9]	Reserved			0x0
Bit[8]	DFIFO_RESET	W	1: FIFO reset 0: FIFO no reset (default)	0x0
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[7]	Reserved			0x0
Bit[6:4]	DFIFO_LEVEL	R/W	When FIFO number is smaller than level, set FIFO_FLAG (default=4)	0x4
Bit[3]	Reserved			0x0
Bit[2]	DFIFO_FLAG_IEN	R/W	1: DFIFO_FLAG_INT enabled 0: DFIFO_FLAG_INT disabled	0x0
Bit[1]	DFIFO_EMPTY_IEN	R/W	1: DFIFO_EMPTY_INT enabled 0: DFIFO_EMPTY_INT disabled	0x0
Bit[0]	DFIFO_FULL_IEN	R/W	1: DFIFO_FULL_INT enabled 0: DFIFO_FULL_INT disabled	0x0

- **P_DAC_CH1_Flag (DAC1 FIFO Status Register)**

Register	Offset		Description	Initial Value
P_DAC_CH1_Flag	DAC_BA+0x4C		DAC1 FIFO Status Register	0x0000_0006
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:3]	Reserved			0x0

Bit[2]	DFIFO_FLAG	R/W1C	1: FIFO flag set 0: no FIFO flag	0x1
Bit[1]	DFIFO_EMPTY	R/W1C	1: FIFO empty flag set 0: no FIFO empty flag	0x1
Bit[0]	DFIFO_FULL	R/W1C	1: FIFO full flag set 0: no FIFO full flag	0x0

- **P_DAC_PWM_CTL (PWM control Register)**

Register	Offset		Description	Initial Value
P_DAC_PWM_CTL	DAC_BA+0x50		PWM control Register	0x0000_0002
Bit	**Name**	**R/W**	**Descriptions**	**Initial Value**
Bit[31:1]	Reserved			0x0
Bit[0]	PWM_EN	R/W	1: PWM enable 0: PWM disable	0x0

2.20 Low Voltage Detector

Low Voltage Detector (LVD) builds in precise band-gap reference circuit for accurately detecting VDD level. If P_SMU_PWR_Ctrl[8] (LVD_EN) = 1 and VDD voltage value falls below LVD voltage which is selected by P_SMU_PWR_Ctrl[14;12] (LVD_SEL), as shown in Table4 LVD voltage select, the LVD output P_SMU_PWR_Ctrl[9] (LVD_FLAG) will become high.

LVD_SEL[2:0]	Voltage
111	3.6V
110	3.4V
101	3.2V
100	3.0V
011	2.8V
010	2.4V
001	2.2V
000	2.0V

Table 3 LVD voltage select

2.21 Configuration Options

Users may select different options depending on the application requirement. There are several options that users may select for the NX12F / NX13F series, as shown in Table 5 User Options.

Item	Name	Options
1	Reset Pin (PA8)	1.Reset Pin (default) 2.GPIO
2	LVR (Halt Mode)	1.LVR Enable at Halt Mode 2.LVR Disable at Halt Mode (default)
3	VDD Voltage	1.4.5V (ADC_VDD @ 3.3V, default) 2.3.0V (ADC_VDD @ 2.3V)
4	WDT	1.WDT Enable (default) 2.WDT Disable
5	Ins. RAM (IRAM)	1.IRAM Enable 2.IRAM Disable (default)
6	Input Voltage (V_{IH}/V_{IL})	1.0.7VDD/0.3VDD (default) 2.0.5VDD/0.2VDD
7	SPI0 Functions	1.GPIO (default) 2.SPI
8	SPI0 IO Mode	1.Single/Dual (default) 2.Quad
9	SPI1 Functions	1.GPIO (default) 2.SPI
10	Voice Output	1.PWM (default) 2.DAC
11	PWM Current	1.Normal (default) 2.Large 3.Ultra

Table 4 User Options