ESMT FE8116x Technical Reference Manual

Revision: V0.1 Date: October 15, 2022

TABLE OF CONTENTS

1 GENERAL DESCRIPTION

ESMT FE81 series is a general purpose 8051 platform that targets on various consumer applications. The MCU platform is integrated with an 8051 core running up to 12MHz, up to 8K-byte eMTP flash memory, and up to 1K-byte data SRAM (X-RAM) with 256-byte internal-SRAM (I-RAM). The MCU platform is also integrated with rich peripherals support, including real-time clock (RTC), watchdog timer controller (WDT), timer controller (TMR), UART interface controller (UART), I2C serial interface controller (I2C), serial peripheral interface controller (SPI), 24-bit high resolution delta-sigma ADC (SDM), analog controller (ANAC), STN LCD driver (LCM), and multi-function ALU (MFA).

ESMT FE81 series supports 2.2V ~ 3.6V voltage range. The operation temperature is from -40℃ ~ 125 ℃.

Part	RTC	WDT	TMR	UART	I2C	SPI	SDM	ANAC	LCM	MFA	CAP
FE81XX	1	1	3	1	1	1	1	1	1	1	1

Table 1-1 Peripheral Table

2 FEATURES

2.1 FE81 Series Features

- 8-bit Micro-controller
  - 8051 core runs up to 12MHz@2.9V~3.6V
  - 8051 core runs up to 3MHz@1.8V~2.9V
  - 1-T architecture with extensive clock-gating
  - Compatible 8051 instruction format
  - Single-cycle 8-bit hardware multiplier
  - 2-wire ICE debugger interface ESMT-SWD
    - 8 hardware breakpoints support
    - Memory subsystems and all peripherals access
  - Synchronous high-performance bus (HPB) for memory subsystems
  - Synchronous peripheral bus (PFB) for all peripherals
- Embedded RAM
  - 1 x 256-byte Internal I-RAM for application data storage
  - 1 x 1K-byte single port SRAM (X-RAM) for application data storage
- Flash Memory
  - 8K-byte eMTP flash for application code
  - 32-byte information eMTP flash for system configuration parameters
  - 32-byte information eMTP flash for customer data storage
- Advanced Power Management
  - Various low power mode including slow, wait, slow wait, halt, and shutdown modes
  - External interrupt, reset, and peripheral wake-up mechanism
- System Manager Controller (SMU)
  - Part number identification
  - Reset control mechanism
  - General purpose I/O control mechanism
  - Timer additional control mechanism
  - SRAM parameter configuration control
- Clock Controller (CLKC)
  - High-performance bus (HPB) clock control mechanism
  - Peripheral bus (PFB) clock control mechanism
  - Clock selection control mechanism
  - Clock division control mechanism
  - Power down control mechanism

- External Vector interrupt controller (EVIC)
  - 16 hardware vector interrupt signals
  - Fixed priority level triggered interrupt mechanism
  - Global interrupt enable for nested interrupt selection
  - Edge triggered type of external interrupt
- Flash Memory Controller (FMC)
  - Application code eMTP flash memory access
  - System configuration information eMTP flash memory access
  - Customer data information eMTP flash memory access
- Real-Time Clock (RTC)
  - Second, minute, hour, day for periodic interrupts
  - Programmable alarm support
- Watchdog Timer Controller (WDT)
  - Provide combinations of interrupt and reset when the watchdog timer expires
  - Provide a write protection mechanism for the control or restart registers
  - Programmable source of timer clock
  - Configurable magic numbers for write protection of registers and restart of the timer
  - Externally Watchdog timer be paused by FMC controller
- Timer Controller (TMR)
  - 3 timer controllers
  - Each Timer
    - Each timer has 2-channel
    - Each channel supports 6 multi-function timers
      - 32-bit Timer / 2 16-bit Timers / 4 8-bit Timers / PWM / PWM+16-bit Timer / PWM+8-bit Timers
  - Programmable source of timer clock
  - Timers can be externally paused by SMU controller
- UART Interface Controller (UART)
  - 1 UART interface controller
  - Over-sampling frequency is programmable
  - Support 5 to 8 bits per character
  - Support 1, 1.5, 2 STOP bits
  - Support even, odd and stick parity bits
  - Support programmable baud rate
  - 16-byte hardware transmit or receive FIFOs
- I2C Serial Interface Controller (I2C)
  - 1 I2C serial interface controllers
  - Support standard-mode (100Kb/s), fast-mode (400Kb/s), and fast-mode plus (1Mb/s)

protocols

- Programmable master or slave mode
- Support 7-bit and 10-bit addressing mode
- Support general call address
- Auto clock stretching
- Programmable clock or data timing
- 8-byte hardware transmit or receive FIFOs

- Serial Peripheral Interface Controller (SPI)
  - 1 serial peripheral interface controllers
  - Support MSB or LSB first transfer
  - Support programmable SPI SCLK
  - Programmable master or slave mode
    - SPI Master maximum@6MHz with PFB clock 12MHz
    - SPI Slave maximum@3MHz with PFB clock 12MHz
  - Support 8- or 16- or 24-bit address bus
  - 8-byte hardware transmit or receive FIFOs
- Delta Sigma ADC (SDM)
  - 1K samples per second
  - 24-bit resolution
  - External sensor calibration
  - Temperature calibration
- Analog Controller (ANAC)
  - Current source generation
  - Comparator function
  - Low voltage detection control
  - Low voltage reset control
  - Internal OPAMP
- General Purpose I/O (GPIO)
  - 32 GPIOs, PA0 ~ PA7, PB0 ~ PB7, PC0 ~ PC7, PD0 ~ PD7
  - Five GPIO modes
    - Input mode
    - Push-pull mode
    - Open-drain mode
    - Internal pull-up mode
    - High-impendence mode
  - PA2, PA3, PA5, PB3 can be configured as external interrupt sources with hardware de-bounce support
  - Peripheral IOs are pin-shared with GPIOs

- Multi-Function ALU (MFA)
  - 16-bit signed- or unsigned- multiplier
  - IEEE754 floating point multiplier
  - IEEE754 Log2 unit
- Capture Controller (CAP)
  - 4 channels for input capture function
  - 4 capture modes
    - Edge, rising to falling, falling to rising, falling to falling
  - Capture input filter function
- Clocks
  - Build-in 12MHz high speed RC oscillator
  - Build-in 32KHz low speed RC oscillator
  - External 12MHz or 32KHz crystal input
- Packages
  - RoHS
  - LQFP 48-pin
  - QFN 48-pin

3 PARTS INFORMATION LIST

3.1 ESMT FE81 Series Selection Guide

Part No.	Flash	SRAM	IO	Timer PWM	UART	SPI	I2C	MFA	I2S	Comp.	ADC	ANAC	RTC	CAP	OCD ISP
FE8116x	8KB	1KB	32	3	1	1	1	1	—	1	1	1	1	1	Y

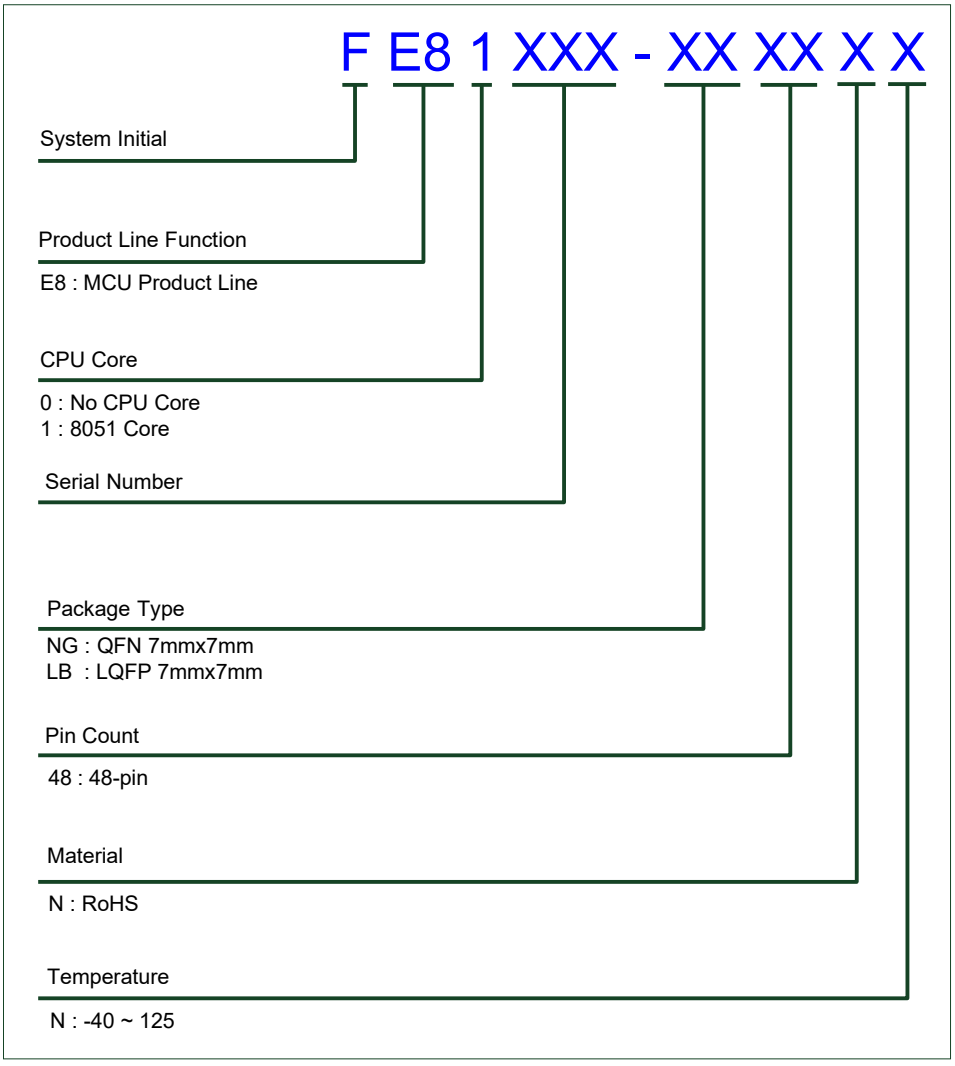


Figure 3.1-1 FE81 Series Selection Code

4 PIN CONFIGURATION

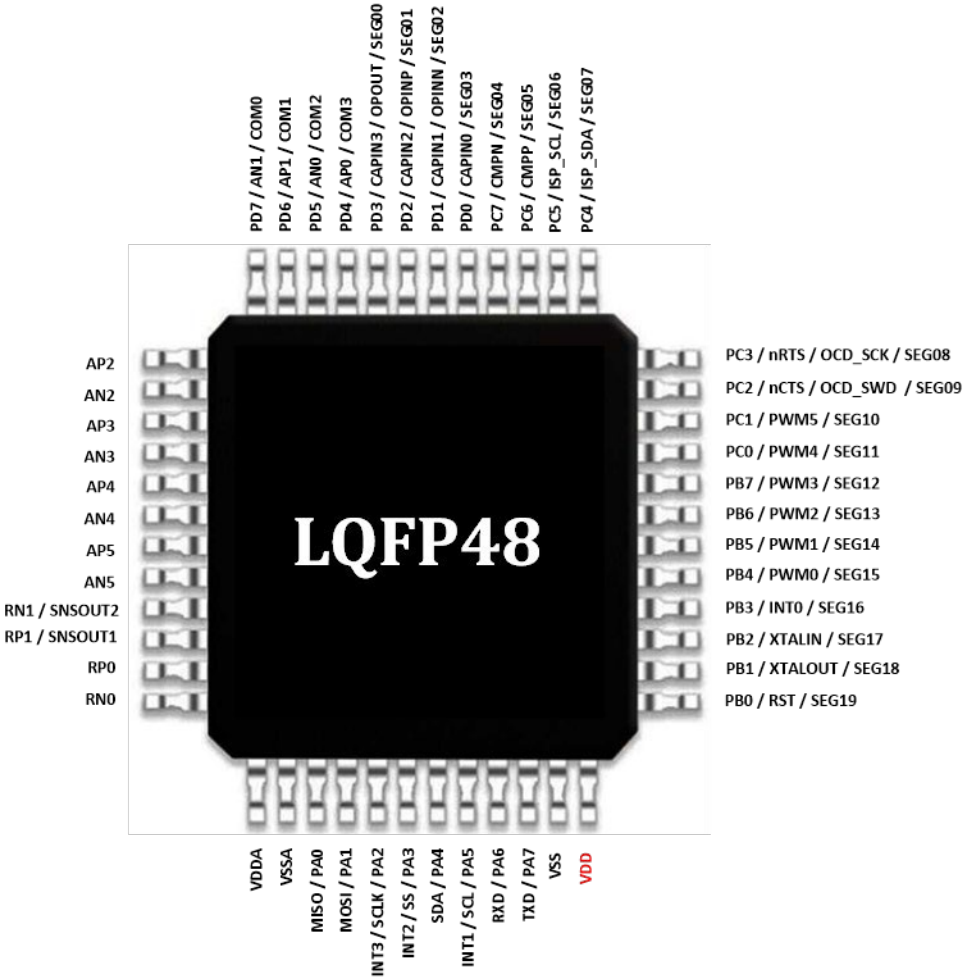


Figure 4-1 FE81 Series Pin Configuration

5 BLOCK DIAGRAM

5.1 ESMT FE81 Series Block Diagram

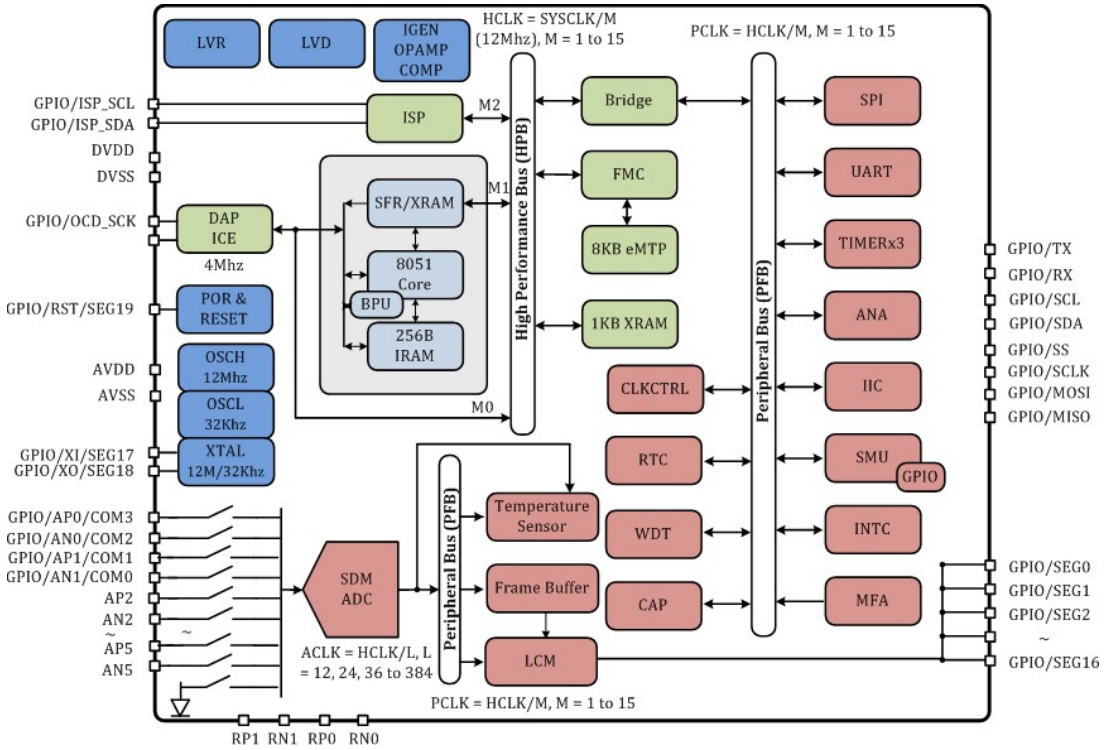


Figure 5.1-1 FE81 Series Block Diagram

The block diagram of FE81 series is illustrated in Figure 5.1-1. The 1-T 8051 core, different from conventional 8051 architecture, accesses memory subsystem and peripherals via the same master port (M1), which increases the benefit of limitation of amount of peripherals and eases the system hardware cost as well. The 8051 core runs up to 12MHz with internal 256-byte I-RAM runs at the same frequency.

The Debug Access Port (DAP), cooperated with Break Point Unit (BPU), is responsible for 8051 core control, break points management and access of memory subsystem and all peripherals via master port (M0), which implements on-chip debugging function with commercial IDE tool. The DAP runs up to 4MHz based on the frequency of external 2-wire ESMT-SWD (OCD_SCK, OCD_SWD) interface.

The In System Programming (ISP), used to program on-chip flash memory via external “I2C protocol”, runs at the frequency of external I2C interface (Up to 400Kb/s) and achieves the purpose of application code on-line update.

The FE81 series supports eMTP flash memory instead of conventional flash memory which has the benefit of hardware cost, although the endurance cycle of eMTP is 1000 cycles. The eMTP has an information block region which can be used for user data storage, whereas the program code is stored at the main block region of the eMTP.

The Flash Memory Controller (FMC), used to access the eMTP, together with eMTP and 1K-byte X-RAM located at high performance bus (HPB) runs at the same frequency domain with the 8051 core.

The FE81 series supports rich peripherals that can be used for signal-conditioning, system peripherals and system connectivity. All these peripherals are located at the peripheral bus (PFB) with the frequency be able to be decreased at lower clock rate (Up to 12MHz). Below lists the peripheral functionalities.

Functionality	Peripheral
Signal Conditioning	Delta Sigma ADC
	External Sensor and Temperature
System Peripherals	Real Time Clock
	Watchdog Timer
	Timer
	STN LCD Controller
	Multi-function ALU
	Analog Controller
	Capture
System Connectivity	UART
	I2C
	SPI

Table 5.1-1 Peripheral Functionality

The FE81 series supports 3 clock sources for general purpose applications, which are on-chip 12 MHz high speed RC oscillator, on-chip 32KHz low speed RC oscillator, and external 12MHz/32KHz crystal.

The other analog (ANAC) features include Low Voltage Detect (LVD), Low Voltage Reset (LVR), current source generation (I-GEN) for off-chip sensor application, internal comparator with external differential input, internal OPAMP with external differential input and output.

6 BLOCK FUNCTIONAL DESCRIPTION

6.1 8051 Core and Bus Architecture

8051 is an 8-bit micro-controller designed for general purpose applications. Traditionally, 8051 uses three address buses to access memory subsystem and peripherals, where are I-RAM bus, SFR bus, and X-RAM bus. In ESMT FE81 series architecture, the I-RAM bus is used to access the internal 256-byte RAM as traditional 8051 does, however, the SFR bus and X-RAM bus are reduced to a single 16-bit high-performance bus and thus a larger memory addressing space is used for memory subsystem and peripherals without the limitation of the amounts of peripherals. That is, a more complicated bus matrix architecture is able to be used for more complicated MCU applications.

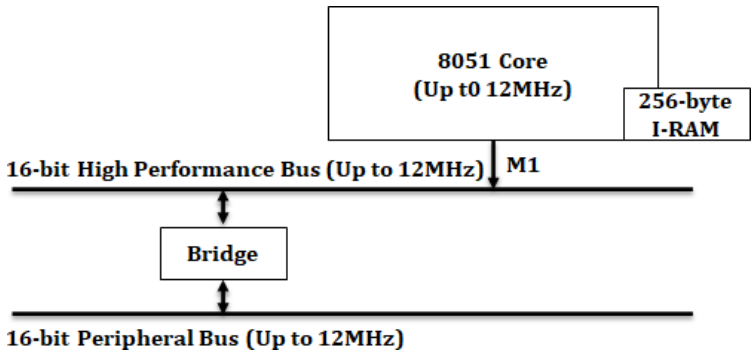


Figure 6.1-1 FE81 Series 8051 Core and Bus Architecture

Figure 6.1-1 depicts the 1T 8051 core and FE81 series bus architecture. This high-performance bus (HPB) runs at the same frequency with the 8051 core, and the frequency of the peripheral bus (PFB) can be decreased according to application requirements. For more detailed description, please refer to the clock architecture in section 6.5.

6.2 Memory Map Organization

6.2.1 FE81 Series Memory Map

The FE81 series provides 64KB (16-bit) memory addressing space. The memory map of FE81 series is illustrated in Figure 6.2-1.

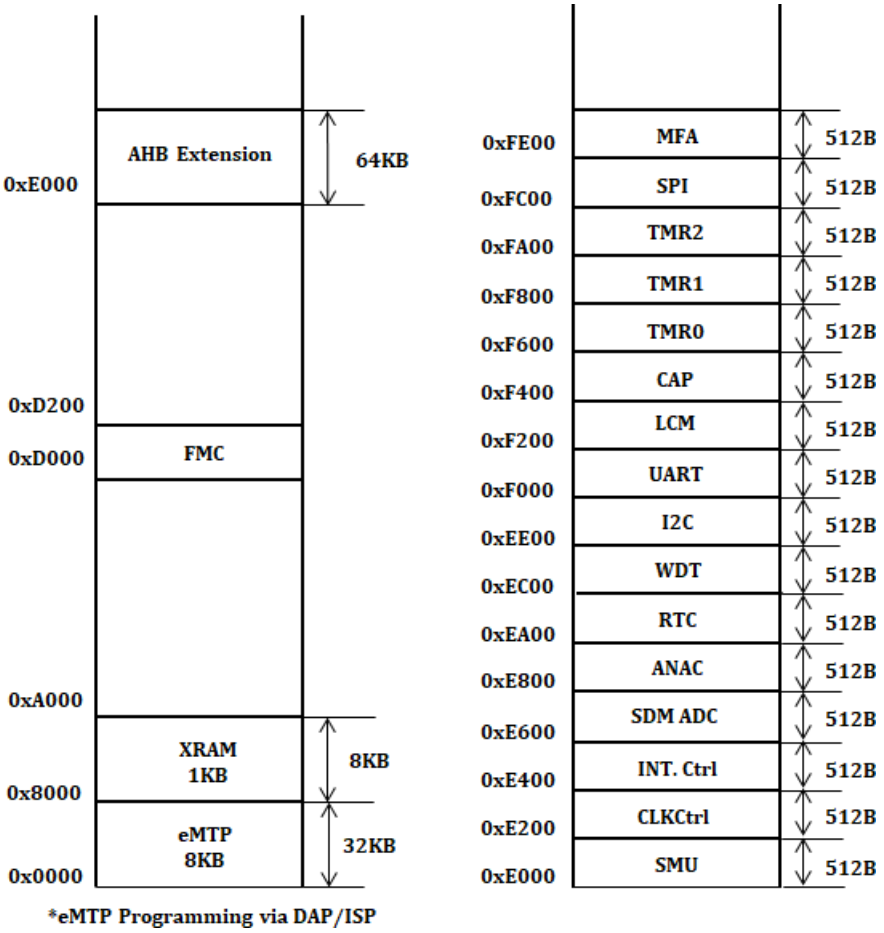


Figure 6.2-1 Memory Map of FE81 Series

The 0x0000 address is mapped to the 8K-byte eMTP for fetching application code by 8051 core after powering on. The 0x8000 address is mapped to the 1K-byte SRAM (X-RAM) for application data. The 0xD000 address is mapped to the flash memory controller (FMC) for accessing data from eMTP by 8051 core or DAP or ISP. The 0xE000 address is mapped to the peripherals for signal conditioning, system peripherals and system connectivity. Table 6.2-1 illustrates the detail information of FE81 series memory organization

Address Space	Token	Controllers
Flash and SRAMs Memory Space		
0x0000 ~ 0x7FFF	FLASH_BA	eMTP Flash Memory Space (32KB) Only 8KB is used.
0x8000 ~ 0xBFFF	XRAM_BA	XRAM Memory Space (8KB) Only 1KB is used.
High Performance Bus (HPB) Controllers Space		
0xD000 ~ 0xD1FF	FMC_BA	FMC Memory Space (512B)
Peripheral Bus (PFB) Controllers Space		
0xE000 ~ 0xE1FF	SMU_BA	System Management Unit Memory Space (512B)
0xE200 ~ 0xE3FF	CLKCTRL_BA	Clock Controller Memory Space (512B)
0xE400 ~ 0xE5FF	INTCTRL_BA	Interrupt Controller Memory Space (512B)
0xE600 ~ 0xE7FF	SDM_BA	Delta Sigma ADC Memory Space (512B)
0xE800 ~ 0xE9FF	ANAC_BA	Analog Controller Memory Space (512B)
0xEA00 ~ 0xEBFF	RTC_BA	Real-Time Clock Memory Space (512B)
0xEC00 ~ 0xEDFF	WDT_BA	Watchdog Memory Space (512B)
0xEE00 ~ 0xEFFF	I2C_BA	I2C Memory Space (512B)
0xF000 ~ 0xF1FF	UART_BA	UART Memory Space (512B)
0xF200 ~ 0xF3FF	LCM_BA	LCD Controller Memory Space (512B)
0xF400 ~ 0xF5FF	CAP_BA	Capture Controller Memory Space (512B)
0xF600 ~ 0xF7FF	TMR0_BA	Timer0 Memory Space (512B)
0xF800 ~ 0xF9FF	TMR1_BA	Timer1 Memory Space (512B)
0xFA00 ~ 0xFBFF	TMR2_BA	Timer2 Memory Space (512B)
0xFC00 ~ 0xFDFF	SPI_BA	SPI Memory Space (512B)
0xFE00 ~ 0xFFFF	MFA_BA	Multi-Function ALU Memory Space (512B)

Table 6.2-1 FE81 Series Memory Map Table

Table 6-2 illustrates all peripheral related register fields started from address 0xE000 to 0xFFFF. Please refer to each section of peripheral for detailed descriptions.

Symbol	Description	Ofset	Bit Addre ss & Symbol								Reset
			Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
Flash Memory Controller Base 0xD000											
FMCCON	FMC Control	0x00	FMCEN	INFOEN	ISAVB	—	—	—	—	WDTOF	0x20
FMCCMD	FMC Command	0x01	CMD								0x00
FMCADR0	FMC Address0	0x02	FMADR0								0x00
FMCADR1	FMC Address1	0x03	FMADR1								0x00
FMCDATA0	FMC Data 0	0x04	FMDAT0								0x00
FMCDATA1	FMC Data 1	0x05	FMDAT1								0x00
FMCSTAT	FMC Status	0x06	BUSY	ECC	—	—	—	—	—	—	0x00
FMCCKSUM0	FMC Checksum 0	0x07	CKSUM0								0x00
FMCCKSUM1	FMC Checksum 1	0x08	CKSUM1								0x00
System Management Unit Base 0xE000											
PNID0	Part Number 0	0x00	PNID0								0x00
PNID1	Part Number 1	0x01	PNID1								0x00
PNID2	Part Number 2	0x02	PNID2								0x00
PNID3	Part Number 3	0x03	PNID3								0x00
RSTSRC	Reset Source	0x04	—	LVR	—	WDT	CPU	CHIP	EXT	POR	0x01
SYSRSTC	System Reset	0x05	—	—	—	—	EXTR	CPURH	CPUR	CHIPR	0x00
PERRSTC0	Peripheral Reset 0	0x06	UARTR	I2CR	—	RTCR	—	LCMR	ANAR	SDMR	0x00
PERRSTC1	Peripheral Reset 1	0x07	—	—	CAPR	MFAR	SPIR	TMR2R	TMR1R	TMR0R	0x00
PACTRL0	PA Control 0	0x08	PA7CTL0	PA6CTL0	PA5CTL0	PA4CTL0	PA3CTL0	PA2CTL0	PA1CTL0	PA0CTL0	0x0F
PACTRL1	PA Control 1	0x09	PA7CTL1	PA6CTL1	PA5CTL1	PA4CTL1	PA3CTL1	PA2CTL1	PA1CTL1	PA0CTL1	0x0F
PAMS0	PA Mode Select 0	0x0C	PA7MOD0	PA6MOD0	PA5MOD0	PA4MOD0	PA3MOD0	PA2MOD0	PA1MOD0	PA0MOD0	0x00
PAMS1	PA Mode Select 1	0x0D	PA7MOD1	PA6MOD1	PA5MOD1	PA4MOD1	PA3MOD1	PA2MOD1	PA1MOD1	PA0MOD1	0x0F
PAMS2	PA Mode Select 2	0x0E	PA7MOD2	PA6MOD2	PA5MOD2	PA4MOD2	PA3MOD2	PA2MOD2	PA1MOD2	PA0MOD2	0x00
PADO0	PA0 DOUT	0x0F	—	—	—	—	—	—	—	PA0DOUT	0x00
PADO1	PA1 DOUT	0x10	—	—	—	—	—	—	—	PA1DOUT	0x00
PADO2	PA2 DOUT	0x11	—	—	—	—	—	—	—	PA2DOUT	0x00
PADO3	PA3 DOUT	0x12	—	—	—	—	—	—	—	PA3DOUT	0x00
PADO4	PA4 DOUT	0x13	—	—	—	—	—	—	—	PA4DOUT	0x00
PADO5	PA5 DOUT	0x14	—	—	—	—	—	—	—	PA5DOUT	0x00
PADO6	PA6 DOUT	0x15	—	—	—	—	—	—	—	PA6DOUT	0x00

PADO7	PA7 DOUT	0x16	–	–	–	–	–	–	–	PA7DOUT	0x00
PAMSK	PA Mask	0x17	PA7MSK	PA6MSK	PA5MSK	PA4MSK	PA3MSK	PA2MSK	PA1MSK	PA0MSK	0x00
PADOUT	PA DOUT	0x18	PA7DOUT	PA6DOUT	PA5DOUT	PA4DOUT	PA3DOUT	PA2DOUT	PA1DOUT	PA0DOUT	0x00
PAPIN	PA Pin	0x19	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	0x00
PBCTRL0	PB Control 0	0x1A	PB7CTL0	PB6CTL0	PB5CTL0	PB4CTL0	PB3CTL0	PB2CTL0	PB1CTL0	PB0CTL0	0x07
PBCTRL1	PB Control 1	0x1B	PB7CTL1	PB6CTL1	PB5CTL1	PB4CTL1	PB3CTL1	PB2CTL1	PB1CTL1	PB0CTL1	0x00
PBMS0	PB Mode Select 0	0x1E	PB7MOD0	PB6MOD0	PB5MOD0	PB4MOD0	PB3MOD0	PB2MOD0	PB1MOD0	PB0MOD0	0x00
PBMS1	PB Mode Select 1	0x1F	PB7MOD1	PB6MOD1	PB5MOD1	PB4MOD1	PB3MOD1	PB2MOD1	PB1MOD1	PB0MOD1	0x00
PBMS2	PB Mode Select 2	0x20	PB7MOD2	PB6MOD2	PB5MOD2	PB4MOD2	PB3MOD2	PB2MOD2	PB1MOD2	PB0MOD2	0x00
PBDO0	PB0 DOUT	0x21	–	–	–	–	–	–	–	PB0DOUT	0x00
PBDO1	PB1 DOUT	0x22	–	–	–	–	–	–	–	PB1DOUT	0x00
PBDO2	PB2 DOUT	0x23	–	–	–	–	–	–	–	PB2DOUT	0x00
PBDO3	PB3 DOUT	0x24	–	–	–	–	–	–	–	PB3DOUT	0x00
PBDO4	PB4 DOUT	0x25	–	–	–	–	–	–	–	PB4DOUT	0x00
PBDO5	PB5 DOUT	0x26	–	–	–	–	–	–	–	PB5DOUT	0x00
PBDO6	PB6 DOUT	0x27	–	–	–	–	–	–	–	PB6DOUT	0x00
PBDO7	PB7 DOUT	0x28	–	–	–	–	–	–	–	PB7DOUT	0x00
PBMSK	PB Mask	0x29	PB7MSK	PB6MSK	PB5MSK	PB4MSK	PB3MSK	PB2MSK	PB1MSK	PB0MSK	0x00
PBDOUT	PB DOUT	0x2A	PB7DOUT	PB6DOUT	PB5DOUT	PB4DOUT	PB3DOUT	PB2DOUT	PB1DOUT	PB0DOUT	0x00
PBPIN	PB Pin	0x2B	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	0x00
PCCTRL0	PC Control 0	0x2C	PC7CTL0	PC6CTL0	PC5CTL0	PC4CTL0	PC3CTL0	PC2CTL0	PC1CTL0	PC0CTL0	0x00
PCCTRL1	PC Control 1	0x2D	PC7CTL1	PC6CTL1	PC5CTL1	PC4CTL1	PC3CTL1	PC2CTL1	PC1CTL1	PC0CTL1	0x00
PCMS0	PC Mode Select 0	0x30	PC7MOD0	PC6MOD0	PC5MOD0	PC4MOD0	PC3MOD0	PC2MOD0	PC1MOD0	PC0MOD0	0x00
PCMS1	PC Mode Select 1	0x31	PC7MOD1	PC6MOD1	PC5MOD1	PC4MOD1	PC3MOD1	PC2MOD1	PC1MOD1	PC0MOD1	0x00
PCMS2	PC Mode Select 2	0x32	PC7MOD2	PC6MOD2	PC5MOD2	PC4MOD2	PC3MOD2	PC2MOD2	PC1MOD2	PC0MOD2	0x00
PCDO0	PC0 DOUT	0x33	–	–	–	–	–	–	–	PC0DOUT	0x00
PCDO1	PC1 DOUT	0x34	–	–	–	–	–	–	–	PC1DOUT	0x00
PCDO2	PC2 DOUT	0x35	–	–	–	–	–	–	–	PC2DOUT	0x00
PCDO3	PC3 DOUT	0x36	–	–	–	–	–	–	–	PC3DOUT	0x00
PCDO4	PC4 DOUT	0x37	–	–	–	–	–	–	–	PC4DOUT	0x00
PCDO5	PC5 DOUT	0x38	–	–	–	–	–	–	–	PC5DOUT	0x00
PCDO6	PC6 DOUT	0x39	–	–	–	–	–	–	–	PC6DOUT	0x00
PCDO7	PC7 DOUT	0x3A	–	–	–	–	–	–	–	PC7DOUT	0x00
PCMSK	PC Mask	0x3B	PC7MSK	PC6MSK	PC5MSK	PC4MSK	PC3MSK	PC2MSK	PC1MSK	PC0MSK	0x00
PCDOUT	PC DOUT	0x3C	PC7DOUT	PC6DOUT	PC5DOUT	PC4DOUT	PC3DOUT	PC2DOUT	PC1DOUT	PC0DOUT	0x00
PCPIN	PC Pin	0x3D	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	0x00
PDCTRL0	PD Control 0	0x3E	PD7CTL0	PD6CTL0	PD5CTL0	PD4CTL0	PD3CTL0	PD2CTL0	PD1CTL0	PD0CTL0	0x00
PDCTRL1	PD Control 1	0x3F	PD7CTL1	PD6CTL1	PD5CTL1	PD4CTL1	PD3CTL1	PD2CTL1	PD1CTL1	PD0CTL1	0x00
PDMS0	PD Mode Select 0	0x42	PD7MOD0	PD6MOD0	PD5MOD0	PD4MOD0	PD3MOD0	PD2MOD0	PD1MOD0	PD0MOD0	0x00
PDMS1	PD Mode Select 1	0x43	PD7MOD1	PD6MOD1	PD5MOD1	PD4MOD1	PD3MOD1	PD2MOD1	PD1MOD1	PD0MOD1	0x00
PDMS2	PD Mode Select 2	0x44	PD7MOD2	PD6MOD2	PD5MOD2	PD4MOD2	PD3MOD2	PD2MOD2	PD1MOD2	PD0MOD2	0x00
PDDO0	PD0 DOUT	0x45	–	–	–	–	–	–	–	PD0DOUT	0x00
PDDO1	PD1 DOUT	0x46	–	–	–	–	–	–	–	PD1DOUT	0x00
PDDO2	PD2 DOUT	0x47	–	–	–	–	–	–	–	PD2DOUT	0x00
PDDO3	PD3 DOUT	0x48	–	–	–	–	–	–	–	PD3DOUT	0x00
PDDO4	PD4 DOUT	0x49	–	–	–	–	–	–	–	PD4DOUT	0x00
PDDO5	PD5 DOUT	0x4A	–	–	–	–	–	–	–	PD5DOUT	0x00
PDDO6	PD6 DOUT	0x4B	–	–	–	–	–	–	–	PD6DOUT	0x00
PDDO7	PD7 DOUT	0x4C	–	–	–	–	–	–	–	PD7DOUT	0x00
PDMSK	PD Mask	0x4D	PD7MSK	PD6MSK	PD5MSK	PD4MSK	PD3MSK	PD2MSK	PD1MSK	PD0MSK	0x00
PDDOUT	PD DOUT	0x4E	PD7DOUT	PD6DOUT	PD5DOUT	PD4DOUT	PD3DOUT	PD2DOUT	PD1DOUT	PD0DOUT	0x00
PDPIN	PD Pin	0x4F	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	0x00
TMRSTAL	Timer Stall	0x50	–	–	–	–	–	TMR2ST	TMR1ST	TMR0ST	0x00
RAMCONF	RAM Config.	0x51	MS				–	MSE	DVS		0x02
Clock Controller Base 0xE200											
CLKEN	CK Enable	0x00	–	–	–	XTALSEL	–	LSRCEN	HSRCEN	XTALEN	0x07
CLKSTB	CK Stable	0x01	–	–	–	–	–	LSRCSTB	HSRCSTB	XTALSTB	0x00
CLKSEL	CK Selection	0x02	TMR2SRC	TMR1SRC	TMR0SRC	EINTSRC	RTCSRC	PMUSRC	SYSSRC		0x00
HPBCLKC	HP. Bus CK Contr.	0x03	–	–	–	–	FMCEN	PFBEN	HPBEN	MCUEN	0x07
PFBCLKC0	PF. Bus CK Contr. 0	0x04	UARTEN	I2CEN	WDTEN	RTCEN	–	LCMEN	ANAEN	SDMEN	0x00
PFBCLKC1	PF. Bus CK Contr. 1	0x05	PDEN	PCEN	PBEN	PAEN	SPIEN	TMR2EN	TMR1EN	TMR0EN	0x00
PFBCLKC2	PF. Bus CK Contr. 2	0x06	–	–	–	–	–	CAPEN	MFAEN	EINTEN	0x00
SYSDIV	System CK Div.	0x07	PFB_N				MCU_N				0x00
PFBDIV0	PF. Bus CK Div. 0	0x08	TMR0_N		–	–	–	SDM_N			0x01
PFBDIV1	PF. Bus CK Div. 1	0x09	–	–	–	–	TMR2_N		TMR1_N		0x00
PMCON	Power Mode	0x0A	–	–	–	–	–	–	PMCON		0x00
CLKTRIM	CK Trim	0x10	HSRCH	HSRCL							0x00
Interrupt Controller Base 0xE400											
GINTMSK	Global INT. Mask	0x00	–	–	–	–	–	–	–	GINTEN	0x00
EINTCTRL	EINT. Control	0x01	EINT3CTL		EINT2CTL		EINT1CTL		EINT0CTL		0x00
EINTEN	EINT. Enable	0x02	–	–	–	–	EINT3EN	EINT2EN	EINT1EN	EINT0EN	0x00
DBCTRL	EINT Deb. Control	0x03	–	–	–	–	DBSAMPC				0x00
EINTSTAT	EINT Status	0x04	–	–	–	–	EINT3F	EINT2F	EINT1F	EINT0F	0x00
INTSTAT0	Peri. INT. STAT 0	0x05	WDTINTF	RTCINTF	SDMINTF	–	EINT3F	EINT2F	EINT1F	EINT0F	0x00
INTSTAT1	Peri. INT. STAT 1	0x06	–	SPIINTF	UARTINTF	I2CINTF	–	TMR2INTF	TMR1INTF	TMR0INTF	0x00
IRQNUM	IRQ Number	0x07	–	–	–	IRQNUM					0x00
IRQCLR	IRQ Clear	0x08	–	–	–	–	–	–	–	IRQCLR	0x00
Delta Sigma ADC Base 0xE600											
RAWL	Raw Data L	0x00	RAWL								0x00
RAWM	Raw Data M	0x01	RAWM								0x00
RAWH	Raw Data H	0x02	RAWH								0x00
CLBL	Calibration Data L	0x03	CLBL								0x00
CLBM	Calibration Data M	0x04	CLBM								0x00
CLBH	Calibration Data H	0x05	CLBH								0x00
SDMGAINL	Sensor Clb. Gain L	0x06	S_CLB_GAINL								0x00
SDMGAINM	Sensor Clb. Gain M	0x07	S_CLB_GAINM								0x00
SDMGAINH	Sensor Clb. Gain H	0x08	S_CLB_GAINH								0x00

SDMOSL	Sensor Clb. OS L	0x09	S_CLB_OSL								0x00
SDMOSM	Sensor Clb. OS M	0x0A	S_CLB_OSM								0x00
SDMOSH	Sensor Clb. OS H	0x0B	S_CLB_OSH								0x00
SDMOFL	Sensor Clb. OF L	0x0C	S_CLB_OFL								0x00
SDMOFM	Sensor Clb. OF M	0x0D	S_CLB_OFM								0x00
SDMOFH	Sensor Clb. OF H	0x0E	S_CLB_OFH								0x00
TEMPGAINL	Temp. Clb. Gain L	0x0F	T_CLB_GAINL								0x00
TEMPGAINM	Temp. Clb. Gain M	0x10	T_CLB_GAINM								0x00
TEMPGAINH	Temp. Clb. Gain H	0x11	T_CLB_GAINH								0x00
TEMPOSL	Temp. Clb. OS L	0x12	T_CLB_OSL								0x00
TEMPOSM	Temp. Clb. OS M	0x13	T_CLB_OSM								0x00
TEMPOSH	Temp. Clb. OS H	0x14	T_CLB_OSH								0x00
TEMPOFL	Temp. Clb. OF L	0x15	T_CLB_OFL								0x00
TEMPOFM	Temp. Clb. OF M	0x16	T_CLB_OFM								0x00
TEMPOFH	Temp. Clb. OF H	0x17	T_CLB_OFH								0x00
OSR	Over-Sampling	0x18	—	—	—	—	OSR				0x00
INMUX	Input Mux	0x19	ENMUXS	—	VINLMUX			VINHMUX			0x00
RMUX	Refer. Mux	0x1A	ENMUXR	—	—	—	VRLMUX		VRHMUX		0x00
PGASET	Enable Set	0x1B	ENBUFCH	VRBUF	ENPGACH	INBUF	—	PGAGN			0x00
ENCFG	Config	0x1C	ENADC	ENADCCH	VRGN		ADCGN				0x00
DCSET	DC Set	0x1D	—	—	—	—	DCSET				0x00
CONFIG	Config. Set	0x1E	—	—	FCHSEL	FTCHK		ENDEM	ENSNSCH	ENTMPSNS	0x00
INTSTAT	INT. Status	0x1F	INTF	—	—	—	—	—	—	INTEN	0x00
TEST	Test Set	0x80	—	—	—	—	CALDVDC	INCH1	INCH0	INSC	0x00
Analog Controller Base 0xE800											
COMPCFG	Comp. Config.	0x00	CMPHYS			CMPTH		CMPIVTH	CMPHEN	CMPREN	0x00
COMPRES	Comp. Result	0x01	—	—	—	—	—	—	—	CMPOUT	0x00
LVCFG	Low Volt. Config.	0x02	LVDF	LVDLV			LVRLV		LVDEN	LVREN	0x00
LVDIEN	LVD Interrupt En.	0x03	—	—	—	—	—	—	—	LVDIEN	0x00
PWCFG	Power Config.	0x04	—	ENOPAMP	ENBODN	ENBODP	ENPDET	ENVCM	ENLDOA	ENBIAS	0x00
IGENCTRL	Cur. Gen. Contr.	0x05	—			PSW	DAC2N	DAC1N	DAC2P	DAC1P	0x00
IGEN1	Cur. Gen. En1.	0x06	ENIDAC1	IDAC1							0x00
IGEN2	Cur. Gen. En2	0x07	ENIDAC2	IDAC2							0x00
IGENTRIM1	Cur. Gen. Trim 1	0x08				IDAC1TR					0x00
IGENTRIM2	Cur. Gen. Trim 2	0x09				IDAC2TR					0x00
BGTRIM	BG Trim	0x0A	ENBGTR	—	—	BG_TRIM					0x00
Real-Time Clock Base 0xEA00											
CNTSEC	Second Counter	0x00	—	—	CNTSEC						0x00
CNTMIN	Minute Counter	0x01	—	—	CNTMIN						0x00
CNTHOUR	Hour Counter	0x02	—	—	—	CNTHOUR					0x00
CNTDAY	Day Counter	0x03	—	—	—	CNTDAY					0x00
ALMSEC	Second Alarm	0x04	—	—	ALMSEC						0x00
ALMMIN	Minute Alarm	0x05	—	—	ALMMIN						0x00
ALMHOUR	Hour Alarm	0x06	—	—	—	ALMHOUR					0x00
RTCCTRL	RTC Control	0x07	—	SECIEN	MINIEN	HORIEN	DAYIEN	ALMIEN	ALMWEN	RTCEN	0x00
RTCSTAT	RTC Status	0x08	—	SECINTF	MININTF	HORINTF	DAYINTF	ALMINTF			0x00
RTCWDONE	RTC Write Done	0x09	—	—	—	—	—	—	—	WDONE	0x00
Watchdog Base 0xEC00											
WDTCTRL0	WDT Control 0	0x00	—	RSTTIME			—	INTTIME			0x00
WDTCTRL1	WDT Control 1	0x01	—	—	—	—	RSTEN	INTEN	CLKSEL	WDTEN	0x00
RESTART	Restart	0x02	0x5A								0x5A
WRPROT	Write Protect	0x03	0xA5								0xA5
STATUS	Status	0x04	—	—	—	—	—	—	—	WDTINTF	0x00
I2C Base 0xEE00											
HWCFG	HW FIFO Config.	0x00	—	—	—	—	—	—	FIFOSIZE		0x00
INTEN0	INT. EN. 0	0x01	BTRN	START	STOP	ARBLOS	ADRHIT	FIFOHAF	FIFOFUL	FIFOEMP	0x00
INTEN1	INT. EN. 1	0x02	—	—	—	—	—	—	CMPL	BRCV	0x00
INTSTAT0	INT. Stat. 0	0x03	BTRNF	STARTF	STOPF	ARBLOSF	ADRHITF	FIFOHAFF	FIFOFULF	FIFOEMPF	0x00
INTSTAT1	INT. Stat. 1	0x04	—	LINESDA	LINESCL	GENCALL	BUSBUSY	ACK	CMPLF	BRCVF	0x00
SLVADR0	Slave Addr. 0	0x05	SLVADRL								0x00
SLVADR1	Slave Addr. 1	0x06	—	—	—	—	—	—	SLVADRH		0x00
I2CDATA	I2C Data	0x07	I2CDAT								0x00
DATACNT	I2C Data Counter	0x08	DATCNT								0x00
I2CCTRL	I2C Control	0x09	—	—	—	PHSTART	PHADR	PHDAT	PHSTOP	DIR	0x00
I2CCMD	I2C Command	0x0A	I2CCMD								0x00
I2CSETUP0	I2C Setup 0	0x0B	—	—	—	—	—	MASTER	ADDRING	I2CEN	0x00
I2CSETUP1	I2C Setup 1	0x0C	T_SCLHi								0x00
I2CSETUP2	I2C Setup 2	0x0D	—	T_HDDAT					T_SCLRAT	T_SCLHi	0x00
I2CSETUP3	I2C Setup 3	0x0E	T_DATSETUPT					T_SP			0x00
UART Base 0xF000											
HWCFG	HW FIFO Config.	0x00	—	—	—	—	—	—	FIFOSIZE		0x00
OSCRCTRL	Over-Sampl. Rate	0x01	—	—	—	OSCR					0x00
DATA_DLL	Receive Buffer	0x02	RBR								0x00
	Transmitter Holding		THR								
	Div. LSB		DLL								
IER_DLM	Interrupt Enable	0x03	—	—	—	—	EMSI	ELSI	ETHEI	ERBI	0x00
	Div. MSB		DLM								
IIR_FCR	Interrupt Iden.	0x04	FIFOED	FIFOED	—	—	INTRID				0x00
	FIFO Control		RFIFOT	RFIFOT	TFIFOT	TFIFOT	—	TFIFORST	RFIFORST	FIFOE	
LINECTRL	Line Control	0x05	DLAB	BC	SPS	EPS	PEN	STB	WLS		0x00
MODMCFG	Modem Control	0x06	—	—	AFCE	LOOP	—	—	RTS	—	0x00
LINESTAT	Line Status	0x07	ERRF	TEMT	THER	LBREAK	FERR	PERR	OERR	DR	0x00
MODMSTAT	Modem Status	0x08	—	—	—	CTS	—	—	—	DCTS	0x00
LCD Controller Base 0xF200											
LCMCTRL0	LCM Control 0	0x00	—	ENBUF	FREQ	LEVEL	VCTR				0x00
LCMCTRL1	LCM Control 1	0x01	—	—	—	—	—	—	—	LCMEN	0x00
SEQBUF00	Frame Buffer 00	0x10	—	—	—	—	FRAMEBUF00				0x00

SEQBUF01	Frame Buffer 01	0x11	–	–	–	–	FRAMEBUF01				0x00
SEQBUF02	Frame Buffer 02	0x12	–	–	–	–	FRAMEBUF02				0x00
SEQBUF03	Frame Buffer 03	0x13	–	–	–	–	FRAMEBUF03				0x00
SEQBUF04	Frame Buffer 04	0x14	–	–	–	–	FRAMEBUF04				0x00
SEQBUF05	Frame Buffer 05	0x15	–	–	–	–	FRAMEBUF05				0x00
SEQBUF06	Frame Buffer 06	0x16	–	–	–	–	FRAMEBUF06				0x00
SEQBUF07	Frame Buffer 07	0x17	–	–	–	–	FRAMEBUF07				0x00
SEQBUF08	Frame Buffer 08	0x18	–	–	–	–	FRAMEBUF08				0x00
SEQBUF09	Frame Buffer 09	0x19	–	–	–	–	FRAMEBUF09				0x00
SEQBUF10	Frame Buffer 10	0x1A	–	–	–	–	FRAMEBUF10				0x00
SEQBUF11	Frame Buffer 11	0x1B	–	–	–	–	FRAMEBUF11				0x00
SEQBUF12	Frame Buffer 12	0x1C	–	–	–	–	FRAMEBUF12				0x00
SEQBUF13	Frame Buffer 13	0x1D	–	–	–	–	FRAMEBUF13				0x00
SEQBUF14	Frame Buffer 14	0x1E	–	–	–	–	FRAMEBUF14				0x00
SEQBUF15	Frame Buffer 15	0x1F	–	–	–	–	FRAMEBUF15				0x00
SEQBUF16	Frame Buffer 16	0x20	–	–	–	–	FRAMEBUF16				0x00
SEQBUF17	Frame Buffer 17	0x21	–	–	–	–	FRAMEBUF17				0x00
SEQBUF18	Frame Buffer 18	0x22	–	–	–	–	FRAMEBUF18				0x00
SEQBUF19	Frame Buffer 19	0x23	–	–	–	–	FRAMEBUF19				0x00
Capture Base 0xF400											
PRESCALE	Pre-scale	0x00	PRESCALE								0x00
LPFSEL	Filter Num. Sel.	0x01	CAP3LPF		CAP2LPF		CAP1LPF		CAP0LPF		0x00
MODE	Mode	0x02	CAP3MODE		CAP2MODE		CAP1MODE		CAP0MODE		0x00
INTEN	Interrupt Enable	0x03	CAP3OIEN	CAP3IEN	CAP2OIEN	CAP2IEN	CAP1OIEN	CAP1IEN	CAP0OIEN	CAP0IEN	0x00
CAPCTRL	CAP Control	0x04	CAP3EN	CAP2EN	CAP1EN	CAP0EN	–	–	–	–	0x00
CAP0CNTL	CAP0 Counter Low	0x06	CAP0CNTL								0x00
CAP0CNTH	CAP0 Counter High	0x07	CAP0CNTH								0x00
CAP1CNTL	CAP1 Counter Low	0x08	CAP1CNTL								0x00
CAP1CNTH	CAP1 Counter High	0x09	CAP1CNTH								0x00
CAP2CNTL	CAP2 Counter Low	0x0A	CAP2CNTL								0x00
CAP2CNTH	CAP2 Counter High	0x0B	CAP2CNTH								0x00
CAP3CNTL	CAP3 Counter Low	0x0C	CAP3CNTL								0x00
CAP3CNTH	CAP3 Counter High	0x0D	CAP3CNTH								0x00
CAPINTF	CAP Interrupt Flag	0x0F	CAP3OF	CAP3F	CAP2OF	CAP2F	CAP1OF	CAP1F	CAP0OF	CAP0F	0x00
Timer 0 Base 0xF600 / Timer 1 Base 0xF800 / Timer 2 Base 0xFA00											
INTEN	INT. Enable	0x00	CH1IEN3	CH1IEN2	CH1IEN1	CH1IEN0	CH0IEN3	CH0IEN2	CH0IEN1	CH0IEN0	0x00
INTSTAT	INT. Status	0x01	CH1INF3	CH1INF2	CH1INF1	CH1INF0	CH0INF3	CH0INF2	CH0INF1	CH0INF0	0x00
CHEN	Channel Enable	0x02	CH1TEN3/ CH1PEN	CH1TEN2	CH1TEN1	CH1TEN0	CH0TEN3/ CH0PEN	CH0TEN2	CH0TEN1	CH0TEN0	0x00
CH0CTRL	CH0 Control	0x03	–	–	–	PWMPRK	PWMPOL	CHMODE			0x00
CH1CTRL	CH1 Control	0x04	–	–	–	PWMPRK	PWMPOL	CHMODE			0x00
CH0RL0	CH0 Reload 0	0x05	CH0RL0								0x00
CH0RL1	CH0 Reload 1	0x06	CH0RL1								0x00
CH0RL2	CH0 Reload 2	0x07	CH0RL2								0x00
CH0RL3	CH0 Reload 3	0x08	CH0RL3								0x00
CH1RL0	CH1 Reload 0	0x09	CH1RL0								0x00
CH1RL1	CH1 Reload 1	0x0A	CH1RL1								0x00
CH1RL2	CH1 Reload 2	0x0B	CH1RL2								0x00
CH1RL3	CH1 Reload 3	0x0C	CH1RL3								0x00
CH0CNT0	CH0 Counter 0	0x0D	CH0CNT0								0x00
CH0CNT1	CH0 Counter 1	0x0E	CH0CNT1								0x00
CH0CNT2	CH0 Counter 2	0x0F	CH0CNT2								0x00
CH0CNT3	CH0 Counter 3	0x10	CH0CNT3								0x00
CH1CNT0	CH1 Counter 0	0x11	CH1CNT0								0x00
CH1CNT1	CH1 Counter 1	0x12	CH1CNT1								0x00
CH1CNT2	CH1 Counter 2	0x13	CH1CNT2								0x00
CH1CNT3	CH1 Counter 3	0x14	CH1CNT3								0x00
SPI Base 0xFC00											
TRNFMT0	Transfer Format 0	0x00	–	–	–	MOSIBID	LSB	SLVMODE	CPOL	CPHA	0x00
TRNFMT1	Transfer Format 1	0x01	ADRLEN		–	DATLEN					0x00
TRNCTRL0	Transfer Control 0	0x02	RTRNCNT								0x00
TRNCTRL1	Transfer Control 1	0x03	WTRNCNT				TOKENVU	–	–	RTRNCNT	0x00
TRNCTRL2	Transfer Control 2	0x04	–	–	TOKENEN	WTRNCNT					0x00
TRNCTRL3	Transfer Control 3	0x05	–	–	ADREN	–	TRNMODE				0x00
SPICMD	SPI Command	0x06	CMD								0x00
SPIADR0	SPI Address 0	0x07	SPIADR0								0x00
SPIADR1	SPI Address 1	0x08	SPIADR1								0x00
SPIADR2	SPI Address 2	0x09	SPIADR2								0x00
SPIDATA	SPI Data	0x0A	SPIDAT								0x00
SPICTRL0	SPI Control 0	0x0B	–	–	–	–	–	TFIFORST	RFIFORST	SPIRST	0x00
SPICTRL1	SPI Control 1	0x0C	–	–	–	RTHRCV					0x00
SPICTRL2	SPI Control 2	0x0D	–	–	–	TTHTRN					0x00
SPISTAT0	SPI Status 0	0x0E	–	–	–	–	–	–	–	BUSY	0x00
SPISTAT1	SPI Status 1	0x0F	RFUL	REMP	–	RFIFONUM					0x00
SPISTAT2	SPI Status 2	0x10	TFUL	TEMP	–	TFIFONUM					0x00
INTEN	INT. Enable	0x11	–	–	SLVCMDIEN	ENDTIEN	TFIFOIEN	RFIFOIEN	TFIFOURIEN	RFIFOORIEN	0x00
INTSTAT	INT. Status	0x12	–	–	SLVCMDINTF	ENDTINTF	TFIFOINTF	RFIFOINTF	TFIFOURINTF	RFIFOORINTF	0x00
INTFTIM0	Interface Timing 0	0x13	SCLKDIV								0x00
INTFTIM1	Interface Timing 1	0x14	–	–	CS2SCLK		CSHT				0x00
SLVSTAT0	Slave Status 0	0x15	USERSTAT0								0x00
SLVSTAT1	Slave Status 1	0x16	USERSTAT1								0x00
SLVSTAT2	Slave Status 2	0x17	–	–	–	–	–	URRUN	ORRUN	RDY	0x00
RCVCNT0	Recev. Counter 0	0x18	RCVCNTL								0x00
RCVCNT1	Recev. Counter 1	0x19	–	–	–	–	–	–	–	RCVCNTH	0x00
TSMCNT0	Trans. Counter 0	0x1A	TRNCNTL								0x00
TSMCNT1	Trans. Counter 1	0x1B	–	–	–	–	–	–	–	TRNCNTH	0x00
SPICFG	SPI Configuration	0x1C	–	–	TFIFOSIZE		–	–	RFIFOSIZE		0x00

Multi-Function ALU Base 0xFE00											
MULCTRL	Multiply Control	0x00	–	–	–	–	–	–	–	SIGN	0x00

MULP0	Multiplier 0	0x01	MULTIPLIER0								0x00
MULP1	Multiplier 1	0x02	MULTIPLIER1								0x00
MULC0	Multiplicand 0	0x03	MULTIPLICAND0								0x00
MULC1	Multiplicand 1	0x04	MULTIPLICAND1								0x00
MULPROD0	Multiply Product 0	0x05	MULPRODUCT0								0x00
MULPROD1	Multiply Product 1	0x06	MULPRODUCT1								0x00
MULPROD2	Multiply Product 2	0x07	MULPRODUCT2								0x00
MULPROD3	Multiply Product 3	0x08	MULPRODUCT3								0x00
FPCTRL	FP. Control	0x09	–	–	–	–	–	ROUND			0x00
FPMULP0	FP. Multiplier 0	0x0A	FPMULTIPLIER0								0x00
FPMULP1	FP. Multiplier 1	0x0B	FPMULTIPLIER1								0x00
FPMULP2	FP. Multiplier 2	0x0C	FPMULTIPLIER2								0x00
FPMULP3	FP. Multiplier 3	0x0D	FPMULTIPLIER3								0x00
FPMULC0	FP. Multiplicand 0	0x0E	FPMULTIPLICAND0								0x00
FPMULC1	FP. Multiplicand 1	0x0F	FPMULTIPLICAND1								0x00
FPMULC2	FP. Multiplicand 2	0x10	FPMULTIPLICAND2								0x00
FPMULC3	FP. Multiplicand 3	0x11	FPMULTIPLICAND3								0x00
FPPROD0	FP. Mul. Product 0	0x12	FPMULPRODUCT0								0x00
FPPROD1	FP. Mul. Product 1	0x13	FPMULPRODUCT1								0x00
FPPROD2	FP. Mul. Product 2	0x14	FPMULPRODUCT2								0x00
FPPROD3	FP. Mul. Product 3	0x15	FPMULPRODUCT3								0x00
FPSTAT	FP. Status	0x16	FPSTAT								0x00
LOGIN0	Log2 Input 0	0x17	LOGINPUT0								0x00
LOGIN1	Log2 Input 1	0x18	LOGINPUT1								0x00
LOGIN2	Log2 Input 2	0x19	LOGINPUT2								0x00
LOGIN3	Log2 Input 3	0x1A	LOGINPUT3								0x00
LOGOUT0	Log2 Output 0	0x1B	LOGOUTPUT0								0x00
LOGOUT1	Log2 Output 1	0x1C	LOGOUTPUT1								0x00
LOGOUT2	Log2 Output 2	0x1D	LOGOUTPUT2								0x00
LOGOUT3	Log2 Output 3	0x1E	LOGOUTPUT3								0x00
LOGSTAT	Log2 Status	0x1F	LOGSTAT								0x00

Table 6.2-2 FE81 Series Peripheral Register Map Table

6.3 External Vectored Interrupt Controller (EVIC)

6.3.1 Overview

The 8051 core supports external vectored interrupt controller (EVIC), which is used where the interrupts are prioritized outside the 8051 core using external interrupt control logic. When an interrupt happens, the program flow will be directed to the interrupt handler that manipulates the interrupt event. To speed up the interrupt handling based on the importance of the interrupt event, an interrupt vectored table is provided to dispatch the interrupt according to the interrupt type. Each interrupt is taken according to the content of the interrupt vectored table, which is called entry point, to reducing the software effort of dispatching the interrupt handling jobs to interrupt handler. Section 6.3.2 describes the interrupt entry point and peripheral interrupt map.

The EVIC supports fixed interrupt priority levels for each hardware interrupt source ranging from 0 (highest priority) to 15 (lowest priority) as defined by IRQ number, that is, several interrupts are activated “at the same time” with the highest priority interrupt being triggered to execute related interrupt handler.

The EVIC supports nested interrupt mechanism, which means the program flow of an interrupt handler executed will be preempted by another interrupt being activated no matter the priority level. User can use global interrupt enable flag to disable the mechanism as described in section 6.3.3.

The FE81 series supports several low power modes with different current topologies according to different application requirements. Once a low power mode is entered, system is able to be wakeup by interrupts or resets.

Features of EVIC

- Priority level triggered Interrupt mechanism
- Nested interrupt mechanism controlled by global interrupt enable flag
- Recovery from low power mode by interrupts

6.3.2 FE81 Series Interrupt Entry Point and Interrupt Map

The FE81 series with 8051 core support 16 interrupts. Table 6.3-1 lists the interrupt model supported by FE81 series. Note that an entry point can be mapped to multiple interrupt types.

Offset	Entry Point	Supported Interrupt Types
0 ~ 15	Interrupt (IRQ0 ~ IRQ15)	Hardware Interrupts

Table 6.3-1 Interrupt Vectored Table

Table 6.3-2 depicts all the peripheral interrupt numbers with the priority level defined by the sequence of the IRQ number. External interrupts (ext0_int ~ ext3_int) have the highest priority levels than the other peripherals with the application requirements response to external events.

IRQ Num.	Interrupt Name	Description
0	ext0_int	External Interrupt From PB3
1	ext1_int	External Interrupt From PA5
2	ext2_int	External Interrupt From PA3
3	ext3_int	External Interrupt From PA2
4	Reserved	Reserved
5	sdm_int	SDM Interrupt
6	rtc_int	RTC Interrupt
7	wdt_int	WDT Interrupt
8	tmr0_int	TMR0 Interrupt
9	tmr1_int	TMR1 Interrupt
10	tmr2_int	TMR2 Interrupt
11	i2c_int	I2C Interrupt
12	uart_int	UART Interrupt
13	spi_int	SPI Interrupt
14	cap_int	Capture Interrupt
15	Reserved	Reserved

Table 6.3-2 Interrupt Map of FE81 Series

6.3.3 FE81 Series Interrupt Priority

Each peripheral has its own interrupt enable flag, and the interrupt signal of each peripheral is activated only when interrupt enable flag is set. The priority level of each peripheral is shown in Figure 6.3-1. Four external interrupts have the highest priority levels and SPI has the lowest priority level. Not only the interrupt enable flag should be set in advance, the global interrupt enable flag (GINTEN in GINTMASK Register) must also be set to activate the interrupt signal, and after interrupt handler is executed, the global interrupt enable flag (GINTEN) is cleared automatically for prohibiting another interrupt request. User can re-set the global interrupt flag at the beginning of interrupt service routine (ISR) to enable nested interrupt mechanism requested by another peripherals.

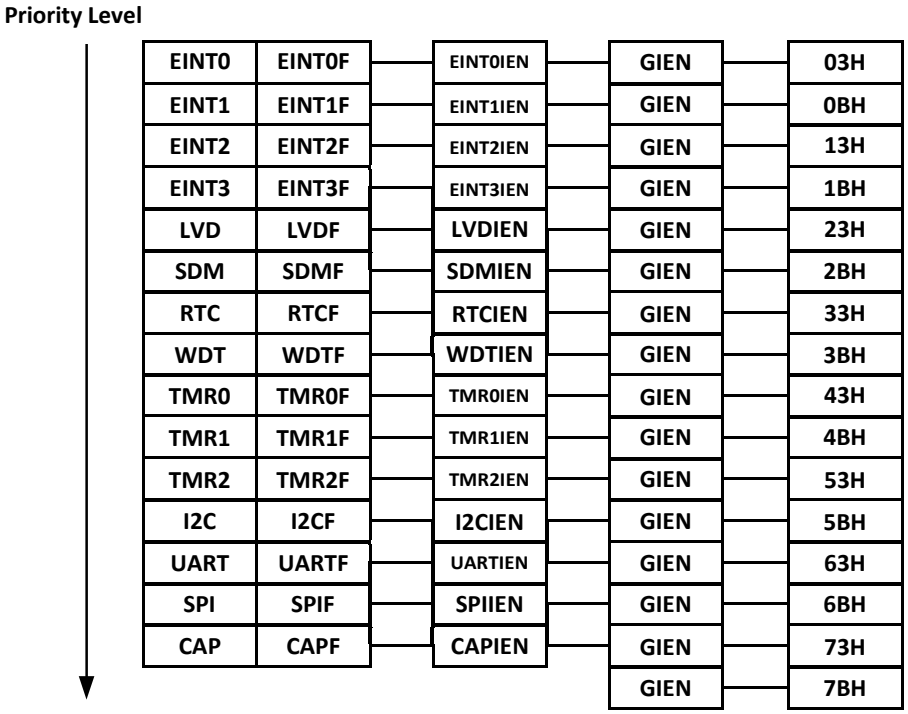


Figure 6.3-1 FE81 Series Interrupt Priority Level

Steps of nested interrupt triggered sequence :

1. Set corresponding interrupt enable flag of each peripheral
2. Set global interrupt enable flag (GINTEN flag in GINTMASK Register)
3. Enter ISR according to priority level → global interrupt enable flag is cleared automatically by hardware
4. Enable global interrupt enable flag for nested interrupt at the beginning of ISR
5. ISR executed or be preempted by another interrupt request

Steps of un-nested interrupt triggered sequence :

1. Set corresponding interrupt enable flag of each peripheral
2. Set global interrupt enable flag (GINTEN flag in GINTMASK Register)
3. Enter ISR according to priority level → global interrupt enable flag is cleared automatically by hardware
4. ISR executed
5. Enable global interrupt enable flag at the end of the ISR or main application code

6.3.4 External Interrupt

The FE81 series supports four external interrupts which can be used for external events. For example, a switch button to trigger the display of temperature of thermometer applications. The four external interrupt pins are pin-shared with GPIO pins as shown below :

GPIO	External Interrupt Pin
PB3	External interrupt 0
PA5	External interrupt 1
PA3	External interrupt 2
PA2	External interrupt 3

The external interrupt supports four edge type triggered interrupts, which are falling and low, rising, falling, and rising and falling. The external interrupt also supports hardware debounce feature as the external pulse signal will be sampled by continuous two debounce sample clocks. Section 6.3.5 depicts the external interrupt register setting. While external interrupt service routine is entered, the related external interrupt flag is cleared by software (external interrupt enable is set).

External interrupt trigger type defined in EINTCTRL Register

- Falling and low
- Rising
- Falling
- Rising and falling

External interrupt flag clear scheme

- If external interrupt enable is set
  - Be cleared by software
- If external interrupt enable is not set
  - Be cleared by software via IRQCLR flag in IRQCLR Register*1

*1 While using IRQCLR flag to clear external interrupt flag, all pending interrupts are cleared, too.

While using external interrupt as for waking up from low power “Halt mode”, the clock source of external interrupt must be selected to LSRC. Please refer to section 6.5.3 for more details.

6.3.5 Register Map and Description

Below lists external vectored interrupt controller register map. The memory base address of EVIC is 0xE400 with the offset indication of each register.

Base Address (INTC_BA) : 0xE400				
Register	**Offset**	**RW**	**Description**	**Reset Value**
GINTMSK	INTC_BA+0x00	R/W	Global Interrupt Mask Register	0x00
EINTCTRL	INTC_BA+0x01	R/W	External Interrupt Control Register	0x00
EINTEN	INTC_BA+0x02	R/W	External Interrupt Enable Register	0x00
DBCTRL	INTC_BA+0x03	R/W	External Interrupt Debounce Control Register	0x00
EINTSTAT	INTC_BA+0x04	R/W	External Interrupt Status Register	0x00
INTSTAT0	INTC_BA+0x05	R	Peripheral Interrupt Status 0 Register	0x00
INTSTAT1	INTC_BA+0x06	R	Peripheral Interrupt Status 1 Register	0x00
IRQNUM	INTC_BA+0x07	R	IRQ Number Register	0x00
IRQCLR	INTC_BA+0x08	W	IRQ Clear Register	0x00

Global Interrupt Mask Register

Register	Offset	RW	Description	Reset Value
GINTMASK	INTC_BA+0x00	R/W	Global Interrupt Mask Register	0x00

Bits	Flag	Description
[7:1]	Reserved	**Reserved.**
[0]	GINTEN	**Global Interrupt Enable Flag** This bit is set to enable global interrupt. This bit is set by software, and cleared by hardware automatically while entering interrupt service routine. GINTEN = 1, Global interrupt enabled. GINTEN = 0, Global interrupt disabled.

External Interrupt Control Register

Register	Offset	RW	Description	Reset Value
EINTCTRL	INTC_BA+0x01	R/W	External Interrupt Control Register	0x00

Bits	Flag	Description
[7:6]	EINT3CTL	**External Interrupt 3 Control Flag** This bit is set by software, and cleared by software. EINT3CTL = 0, Falling and low trigger mode. EINT3CTL = 1, Rising trigger mode. EINT3CTL = 2, Falling trigger mode. EINT3CTL = 3, Rising and Falling trigger mode.
[5:4]	EINT2CTL	**External Interrupt 2 Control Flag** This bit is set by software, and cleared by software. EINT2CTL = 0, Falling and low trigger mode. EINT2CTL = 1, Rising trigger mode. EINT2CTL = 2, Falling trigger mode. EINT2CTL = 3, Rising and Falling trigger mode.
[3:2]	EINT1CTL	**External Interrupt 1 Control Flag** This bit is set by software, and cleared by software. EINT1CTL = 0, Falling and low trigger mode. EINT1CTL = 1, Rising trigger mode. EINT1CTL = 2, Falling trigger mode. EINT1CTL = 3, Rising and Falling trigger mode.
[1:0]	EINT0CTL	**External Interrupt 0 Control Flag** This bit is set by software, and cleared by software. EINT0CTL = 0, Falling and low trigger mode. EINT0CTL = 1, Rising trigger mode. EINT0CTL = 2, Falling trigger mode. EINT0CTL = 3, Rising and Falling trigger mode.

External Interrupt Enable Register

Register	Offset	RW	Description	Reset Value
EINTEN	INTC_BA+0x02	R/W	External Interrupt Enable Register	0x00

Bits	Flag	Description
[7:4]	Reserved	**Reserved.**
[3]	EINT3EN	**External Interrupt 3 Enable Flag** This bit is set by software, and cleared by software. EINT3EN = 1, External interrupt 3 enabled. EINT3EN = 0, External interrupt 3 disabled.
[2]	EINT2EN	**External Interrupt 2 Enable Flag** This bit is set by software, and cleared by software. EINT2EN = 1, External interrupt 2 enabled. EINT2EN = 0, External interrupt 2 disabled.
[1]	EINT1EN	**External Interrupt 1 Enable Flag** This bit is set by software, and cleared by software. EINT1EN = 1, External interrupt 1 enabled. EINT1EN = 0, External interrupt 1 disabled.
[0]	EINT0EN	**External Interrupt 0 Enable Flag** This bit is set by software, and cleared by software. EINT0EN = 1, External interrupt 0 enabled. EINT0EN = 0, External interrupt 0 disabled.

External Interrupt Debounce Control Register

Register	Offset	RW	Description	Reset Value
DBCTRL	INTC_BA+0x03	R/W	External Interrupt Debounce Control Register	0x00

Bits	Flag	Description
[7:4]	Reserved	**Reserved.**
[3:0]	DBSAMPC	**External Interrupt Debounce Sampling Clock Number** This bit is set by software, and cleared by software. If the input signal pulse width cannot be sampled by continuous two debounce cycle, the interrupt will not be triggered. DBSAMPC = 0, Every 1 clocks. DBSAMPC = 1, Every 2 clocks. DBSAMPC = 2, Every 4 clocks. DBSAMPC = 3, Every 8 clocks. DBSAMPC = 4, Every 16 clocks. DBSAMPC = 5, Every 32 clocks. DBSAMPC = 6, Every 64 clocks. DBSAMPC = 7, Every 128 clocks. DBSAMPC = 8, Every 256 clocks.

External Interrupt Status Register

Register	Offset	RW	Description	Reset Value
EINTSTAT	INTC_BA+0x04	R/W	External Interrupt Status Register	0x00

Bits	Flag	Description
[7:4]	Reserved	**Reserved.**
[3]	EINT3F	**External Interrupt 3 Status Flag** This bit is set by hardware, and cleared by software (Write 1 to clear). User can use the flag for polling external signal if external interrupt enable is not set. EINT3F = 1, External interrupt 3 triggered. EINT3F = 0, Idle.
[2]	EINT2F	**External Interrupt 2 Status Flag** This bit is set by hardware, and cleared by software (Write 1 to clear). User can use the flag for polling external signal if external interrupt enable is not set. EINT2F = 1, External interrupt 2 triggered. EINT2F = 0, Idle.
[1]	EINT1F	**External Interrupt 1 Status Flag** This bit is set by hardware, and cleared by software (Write 1 to clear). User can use the flag for polling external signal if external interrupt enable is not set. EINT1F = 1, External interrupt 1 triggered. EINT1F = 0, Idle.
[0]	EINT0F	**External Interrupt 0 Status Flag** This bit is set by hardware, and cleared by software (Write 1 to clear). User can use the flag for polling external signal if external interrupt enable is not set. EINT0F = 1, External interrupt 0 triggered. EINT0F = 0, Idle.

Peripheral Interrupt Status 0 Register

Register	Offset	RW	Description	Reset Value
INTSTAT0	INTC_BA+0x05	R	Peripheral Interrupt Status 0 Register	0x00

Bits	Flag	Description
[7]	WDTINTF	**Watchdog Interrupt Status Flag** This bit is set by hardware, and cleared by hardware automatically. WDTINTF **= 1, WDT interrupt triggered.** WDTINTF **= 0, Idle.**
[6]	RTCINTF	**RTC Interrupt Status Flag** This bit is set by hardware, and cleared by hardware automatically. RTCINTF = 1, RTC interrupt triggered. RTCINTF = 0, Idle.
[5]	SDMINTF	**Delta Sigma Interrupt Status Flag** This bit is set by hardware, and cleared by hardware automatically. SDMINTF = 1, SDM interrupt triggered. SDMINTF = 0, Idle.
[4]	LVDINTF	**LVD Interrupt Status Flag** This bit is set by hardware, and cleared by hardware automatically. LVDINTF = 1, LVD interrupt triggered. LVDINTF = 0, Idle.
[3]	EINT3F	**External Interrupt 0 Status Flag** The same function as defined in EINTENSTAT.
[2]	EINT2F	**External Interrupt 0 Status Flag** The same function as defined in EINTENSTAT.
[1]	EINT1F	**External Interrupt 0 Status Flag** The same function as defined in EINTENSTAT.
[0]	EINT0F	**External Interrupt 0 Status Flag** The same function as defined in EINTENSTAT.

Peripheral Interrupt Status 1 Register

Register	Offset	RW	Description	Reset Value
INTSTAT1	INTC_BA+0x06	R	Peripheral Interrupt Status 1 Register	0x00

Bits	Flag	Description
[7]	Reserved	**Reserved.**
[6]	CAPINTF	**CAP Interrupt Status Flag** This bit is set by hardware, and cleared by hardware automatically. CAPINTF = 1, CAP interrupt triggered. CAPINTF = 0, Idle.
[5]	SPIINTF	**SPI Interrupt Status Flag** This bit is set by hardware, and cleared by hardware automatically. SPIINTF = 1, SPI interrupt triggered. SPIINTF = 0, Idle.
[4]	UARTINTF	**UART Interrupt Status Flag** This bit is set by hardware, and cleared by hardware automatically. UARTINTF = 1, UART interrupt triggered. UARTINTF = 0, Idle.
[3]	I2CINTF	**I2C Interrupt Status Flag** This bit is set by hardware, and cleared by hardware automatically. I2CINTF = 1, I2C interrupt triggered. I2CINTF = 0, Idle.
[2]	TMR2INTF	**Timer 2 Interrupt Status Flag** This bit is set by hardware, and cleared by hardware automatically. TMR2INTF = 1, Timer 2 interrupt triggered. TMR2INTF = 0, Idle.
[1]	TMR1INTF	**Timer 1 Interrupt Status Flag** This bit is set by hardware, and cleared by hardware automatically. TMR1INTF = 1, Timer 1 interrupt triggered. TMR1INTF = 0, Idle.
[0]	TMR0INTF	**Timer 0 Interrupt Status Flag** This bit is set by hardware, and cleared by hardware automatically. TMR0INTF = 1, Timer 0 interrupt triggered. TMR0INTF = 0, Idle.

IRQ Number Register

Register	Offset	RW	Description	Reset Value
IRQNUM	INTC_BA+0x07	R	IRQ Number Register	0x00

Bits	Flag	Description
[7:5]	Reserved	**Reserved.**
[4:0]	IRQNUM	**MCU IRQ Number Flag** This bit is set by hardware, and cleared by hardware. The register field stores the IRQ number that 8051 core has been finished executing related peripheral interrupt service routine.

IRQ Clear Register

Register	Offset	RW	Description	Reset Value
IRQCLR	INTC_BA+0x08	W	IRQ Clear Register	0x00

Bits	Flag	Description
[7:1]	Reserved	**Reserved.**
[0]	IRQCLR	**MCU Interrupt Clear Flag** This bit is set by software, and cleared by hardware. Note that, while setting this flag, all pending interrupts are cleared. IRQCLR = 1, All pending interrupts are cleared. IRQCLR = 0, Idle.

6.4 System Manager Controller (SMU)

6.4.1 Overview

The system manager controller is responsible for system controls of the FE81 series listed as follows.

Part number identification

Part number identification indicates the part number of FE81 series. For instance, E811 is an 8051 core, 8K-byte eMTP, and 1K-byte Data SRAM product.

Reset control

The FE81 series supports several reset mechanisms that provide flexible controls for customers, which is described in section 6.4.2.

General purpose I/O (GPIO) control

The FE81 series supports five GPIO modes, which are input mode, push-pull mode, open-drain mode, internal pull-up mode, and high-impendence mode for analog I/O. The GPIO is also pin-shared with system connectivity for die area reduction

Timer stall control

User is able to use this mechanism to pause timer counting.

SRAM Configuration

Base on various operation voltage setting, SRAM should be set to different parameters.

Contact ESMT for more detail information related to SRAM configuration.

6.4.2 Reset Control

The FE81 series provides flexible reset mechanism for customers to choose the most appropriate reset usage. The FE81 series supports rich reset control, which is listed as follows.

Power on reset

The reset is issued by power on reset circuit to reset the whole chip.

Low active external RST pin

The reset is issued by the external RST pin, which is pin-shared with PB0 to reset whole chip. For example, a reset push button on the PCB board. The RST pin function should be enabled by EXTR bit in SYSRSTC Register.

Chip reset

The reset is one of the system software reset that reset the whole chip. Write CHIPR bit in SYSRSTC Register will issue chip reset.

CPU reset

The reset is one of the system software reset that reset only the 8051 core. Write CPUR bit in SYSRSTC Register will issue the CPU reset.

CPU Reset and Hold

The reset is one of the system software reset that reset and gated the operation of 8051 core. Write CPURH bit in SYSRSTC Register will issue the CPU reset and hold.

Watchdog reset (WDT)

The reset is issued by watchdog timer controller to reset the whole chip.

Low voltage reset (LVR)

The reset is issued by low voltage reset circuit to reset the whole chip.

On-chip debugging (OCD) reset

The reset is issued by DAP controller to reset the whole chip. This reset is only used for OCD control by IDE tool.

In-system programming (ISP) reset

The reset is issued by ISP controller to reset the whole chip. This reset is only used for eMTP flash memory update by on-line update GUI tool provided by ESMT.

Peripheral resets

Each peripheral has its own reset control mechanism to reset each peripheral to default state.

Table 6-.4-1 lists the reset control of each register with corresponding bit field.

Reset	Register	Bit Field
Chip Reset	SYSRSTC	CHUPR : [0]
CPU Reset	SYSRSTC	CPUR : [1]
CPU Reset and Hold	SYSRSTC	CPURH : [2]
RST Pin Enable	SYSRSTC	EXTR : [3]
OCD Reset	DAPCMD	Command : 0x80
ISP Reset	ISP Command Code	Command : 0x99
SDM Reset	PERRSTC0	SDMR : [0]
ANAC Reset	PERRSTC0	ANAR : [1]
LCM Reset	PERRSTC0	LCMR : [2]
RTC Reset	PERRSTC0	RTCR : [4]
I2C Reset	PERRSTC0	I2CR : [6]
UART Reset	PERRSTC0	UARTR : [7]
Timer 0 Reset	PERRSTC1	TMR0R : [0]
Timer 1 Reset	PERRSTC1	TMR1R : [1]
Timer 2 Reset	PERRSTC1	TMR2R : [2]
SPI Reset	PERRSTC1	RSPI : [3]
MFA Reset	PERRSTC1	MFAR : [4]
CAP Reset	PERRSTC1	CAPR : [5]

Table 6.4-1 Reset Control of FE81 Series

6.4.3 General Purpose I/O

Table 6.4-2 depicts mode selection of GPIO. Each Port (PA, PB, PC, PD) has 3-byte mode selection registers for setting the IO Mode.

Mode Selection			IO Mode
PxMS2 (x = A, B, C, D)	PxMS1 (x = A, B, C, D)	PxMS0 (x = A, B, C, D)	
0	0	0	Input
0	0	1	Push-pull
0	1	0	Open-drain
0	1	1	Internal pull-up
1	0	0	High-impendence

Table 6.4-2 GPIO Mode Selection

System peripheral connectivity pins are pin-shared with GPIO pins. Figure 6.4-1 illustrates GPIO function selected by 2-byte GPIO control selection register. For instance, the GPIO function of PA0 can be selected by PACTRL1[0] and PACTRL0[0], respectively.

IO	PACTRL1 PACtrl0	IO Function
PA.0	2'b00	GPIO
	2'b01	MISO
	2'b10	
	2'b11	
PA.1		GPIO
	2'b01	MOSI
	2'b10	
	2'b11	
PA.2	2'b00	GPIO
	2'b01	SCLK
	2'b10	INT3
	2'b11	
PA.3	2'b00	GPIO
	2'b01	SS
	2'b10	INT2
	2'b11	
PA.4	2'b00	GPIO
	2'b01	SDA
	2'b10	
	2'b11	
PA.5	2'b00	GPIO
	2'b01	SCL
	2'b10	INT1
	2'b11	
PA.6	2'b00	GPIO
	2'b01	RXD
	2'b10	
	2'b11	
PA.7	2'b00	GPIO
	2'b01	TXD
	2'b10	
	2'b11	

IO	PBCTRL1 PBCtrl0	IO Function
PB.0	2'b00	GPIO
	2'b01	RST
	2'b10	
	2'b11	SEF19
PB.1	2'b00	GPIO
	2'b01	XTAOUT
	2'b10	
	2'b11	SEG18
PB.2	2'b00	GPIO
	2'b01	XTAIN
	2'b10	
	2'b11	SEG17
PB.3	2'b00	GPIO
	2'b01	
	2'b10	INT0
	2'b11	SEG16
PB.4	2'b00	GPIO
	2'b01	PWM0
	2'b10	
	2'b11	SEG15
PB.5	2'b00	GPIO
	2'b01	PWM1
	2'b10	
	2'b11	SEG14
PB.6	2'b00	GPIO
	2'b01	PWM2
	2'b10	
	2'b11	SEG13
PB.7	2'b00	GPIO
	2'b01	PWM3
	2'b10	
	2'b11	SEG12

IO	PCCTRL1 PCCtrl0	IO Function
PC.0	2'b00	GPIO
	2'b01	PWM4
	2'b10	
	2'b11	SEG11
PC.1	2'b00	GPIO
	2'b01	PWM5
	2'b10	
	2'b11	SEG10
PC.2	2'b00	GPIO
	2'b01	nCTS
	2'b10	OCD_SDA
	2'b11	SEG09
PC.3	2'b00	GPIO
	2'b01	nRTS
	2'b10	OCD_SCK
	2'b11	SEG08
PC.4	2'b00	GPIO
	2'b01	
	2'b10	ISP_SDA
	2'b11	SEG07
PC.5	2'b00	GPIO
	2'b01	
	2'b10	ISP_SCL
	2'b11	SEG06
PC.6	2'b00	GPIO
	2'b01	
	2'b10	CMPP
	2'b11	SEG05
PC.7	2'b00	GPIO
	2'b01	
	2'b10	CMPN
	2'b11	SEG04

IO	PDCTRL1 PDCtrl0	IO Function
PD.0	2'b00	GPIO
	2'b01	CAPIN0
	2'b10	
	2'b11	SEG03
PD.1	2'b00	GPIO
	2'b01	CAPIN1
	2'b10	OPAO
	2'b11	SEG02
PD.2	2'b00	GPIO
	2'b01	CAPIN2
	2'b10	OPAP
	2'b11	SEG01
PD.3	2'b00	GPIO
	2'b01	CAPIN3
	2'b10	OPAN
	2'b11	SEG00
PD.4	2'b00	GPIO
	2'b01	
	2'b10	AP0
	2'b11	COM3
PD.5	2'b00	GPIO
	2'b01	
	2'b10	AN0
	2'b11	COM2
PD.6	2'b00	GPIO
	2'b01	
	2'b10	AP1
	2'b11	COM1
PD.7	2'b00	GPIO
	2'b01	
	2'b10	AN1
	2'b11	COM0

Figure 6.4-1 GPIO Control Selection

Other features of GPIO

- Mask function
- Fast bit output control

6.4.4 Register Map and Description

Table 6-5 lists the register map of system manager controller.

Base Address (SMC_BA) : 0xE000				
Register	**Offset**	**RW**	**Description**	**Reset Value**
PNID0	SMC_BA+0x00	R	Part Number Identification 0 Register	0x45
PNID1	SMC_BA+0x01	R	Part Number Identification 1 Register	0x38
PNID2	SMC_BA+0x02	R	Part Number Identification 2 Register	0x31
PNID3	SMC_BA+0x03	R	Part Number Identification 3 Register	0x31
RSTSRC	SMC_BA+0x04	R/W	Reset Source Register	0x01
SYSRSTC	SMC_BA+0x05	R/W	System Reset Control Register	0x00
PERRSTC0	SMC_BA+0x06	R/W	Peripheral Reset Control 0 Register	0x00
PERRSTC1	SMC_BA+0x07	R/W	Peripheral Reset Control 1 Register	0x00
PACTRL0	SMC_BA+0x08	R/W	PA Control 0 Register	0x00
PACTRL1	SMC_BA+0x09	R/W	PA Control 1 Register	0x00
Reserved				
Reserved				
PAMS0	SMC_BA+0x0C	R/W	PA Mode Select 0 Register	0x00
PAMS1	SMC_BA+0x0D	R/W	PA Mode Select 1 Register	0x00
PAMS2	SMC_BA+0x0E	R/W	PA Mode Select 2 Register	0x00
PADO0	SMC_BA+0x0F	R/W	PA Bit 0 Output Register	0x00
PADO1	SMC_BA+0x10	R/W	PA Bit 1 Output Register	0x00
PADO2	SMC_BA+0x11	R/W	PA Bit 2 Output Register	0x00
PADO3	SMC_BA+0x12	R/W	PA Bit 3 Output Register	0x00
PADO4	SMC_BA+0x13	R/W	PA Bit 4 Output Register	0x00
PADO5	SMC_BA+0x14	R/W	PA Bit 5 Output Register	0x00
PADO6	SMC_BA+0x15	R/W	PA Bit 6 Output Register	0x00
PADO7	SMC_BA+0x16	R/W	PA Bit 7 Output Register	0x00
PAMSK	SMC_BA+0x17	R/W	PA Data Out Write Mask Register	0x00
PADOUT	SMC_BA+0x18	R/W	PA Data Out Register	0x00
PAPIN	SMC_BA+0x19	R/W	PA PIN Value Register	0x00
PBCTRL0	SMC_BA+0x1A	R/W	PB Control 0 Register	0x00
PBCTRL1	SMC_BA+0x1B	R/W	PB Control 1 Register	0x00
Reserved				
Reserved				
PBMS0	SMC_BA+0x1E	R/W	PB Mode Select 0 Register	0x00
PBMS1	SMC_BA+0x1F	R/W	PB Mode Select 1 Register	0x00
PBMS2	SMC_BA+0x20	R/W	PB Mode Select 2 Register	0x00
PBDO0	SMC_BA+0x21	R/W	PB Bit 0 Output Register	0x00
PBDO1	SMC_BA+0x22	R/W	PB Bit 1 Output Register	0x00
PBDO2	SMC_BA+0x23	R/W	PB Bit 2 Output Register	0x00
PBDO3	SMC_BA+0x24	R/W	PB Bit 3 Output Register	0x00
PBDO4	SMC_BA+0x25	R/W	PB Bit 4 Output Register	0x00
PBDO5	SMC_BA+0x26	R/W	PB Bit 5 Output Register	0x00
PBDO6	SMC_BA+0x27	R/W	PB Bit 6 Output Register	0x00
PBDO7	SMC_BA+0x28	R/W	PB Bit 7 Output Register	0x00
PBMSK	SMC_BA+0x29	R/W	PB Data Out Write Mask Register	0x00

PBDOUT	SMC_BA+0x2A	R/W	PB Data Out Register	0x00
PBPIN	SMC_BA+0x2B	R/W	PB PIN Value Register	0x00
PCCTRL0	SMC_BA+0x2C	R/W	PC Control 0 Register	0x00
PCCTRL1	SMC_BA+0x2D	R/W	PC Control 1 Register	0x00
Reserved				
Reserved				
PCMS0	SMC_BA+0x30	R/W	PC Mode Select 0 Register	0x00
PCMS1	SMC_BA+0x31	R/W	PC Mode Select 1 Register	0x00
PCMS2	SMC_BA+0x32	R/W	PC Mode Select 2 Register	0x00
PCDO0	SMC_BA+0x33	R/W	PC Bit 0 Output Register	0x00
PCDO1	SMC_BA+0x34	R/W	PC Bit 1 Output Register	0x00
PCDO2	SMC_BA+0x35	R/W	PC Bit 2 Output Register	0x00
PCDO3	SMC_BA+0x36	R/W	PC Bit 3 Output Register	0x00
PCDO4	SMC_BA+0x37	R/W	PC Bit 4 Output Register	0x00
PCDO5	SMC_BA+0x38	R/W	PC Bit 5 Output Register	0x00
PCDO6	SMC_BA+0x39	R/W	PC Bit 6 Output Register	0x00
PCDO7	SMC_BA+0x3A	R/W	PC Bit 7 Output Register	0x00
PCMSK	SMC_BA+0x3B	R/W	PC Data Out Write Mask Register	0x00
PCDOUT	SMC_BA+0x3C	R/W	PC Data Out Register	0x00
PCPIN	SMC_BA+0x3D	R/W	PC PIN Value Register	0x00
PDCTRL0	SMC_BA+0x3E	R/W	PD Control 0 Register	0x00
PDCTRL1	SMC_BA+0x3F	R/W	PD Control 1 Register	0x00
Reserved				
Reserved				
PDMS0	SMC_BA+0x42	R/W	PD Mode Select 0 Register	0x00
PDMS1	SMC_BA+0x43	R/W	PD Mode Select 1 Register	0x00
PDMS2	SMC_BA+0x44	R/W	PD Mode Select 2 Register	0x00
PDDO0	SMC_BA+0x45	R/W	PD Bit 0 Output Register	0x00
PDDO1	SMC_BA+0x46	R/W	PD Bit 1 Output Register	0x00
PDDO2	SMC_BA+0x47	R/W	PD Bit 2 Output Register	0x00
PDDO3	SMC_BA+0x48	R/W	PD Bit 3 Output Register	0x00
PDDO4	SMC_BA+0x49	R/W	PD Bit 4 Output Register	0x00
PDDO5	SMC_BA+0x4A	R/W	PD Bit 5 Output Register	0x00
PDDO6	SMC_BA+0x4B	R/W	PD Bit 6 Output Register	0x00
PDDO7	SMC_BA+0x4C	R/W	PD Bit 7 Output Register	0x00
PDMSK	SMC_BA+0x4D	R/W	PD Data Out Write Mask Register	0x00
PDDOUT	SMC_BA+0x4E	R/W	PD Data Out Register	0x00
PDPIN	SMC_BA+0x4F	R/W	PD PIN Value Register	0x00
TMRSTAL	SMC_BA+0x50	R/W	Timer Stall Control Register	0x00
RAMCONF	SMC_BA+0x51	R/W	SRAM Configuration Register	0x02

Part Number Identification Register

Register	Offset	RW	Description	Reset Value
PNIDx	SMC_BA+(0x00 ~ 0x03)	R	Part Number Identification 0, 1, 2, 3 Register	0x45383131

Bits	Flag	Description
[7:0]	PNID0	**Part Number Identification 0** Part number identification is located in this register. User can read this register for identifying which device is used.

Bits	Flag	Description
[7:0]	PNID1	**Part Number Identification 1** Part number identification is located in this register. User can read this register for identifying which device is used.

Bits	Flag	Description
[7:0]	PNID2	**Part Number Identification 2** Part number identification is located in this register. User can read this register for identifying which device is used.

Bits	Flag	Description
[7:0]	PNID3	**Part Number Identification 3** Part number identification is located in this register. User can read this register for identifying which device is used.

Part Number	Package	Flash	Data SRAM	PNID
FE8116x		8KB	1KB	0x45383131

Reset Source Register

Register	Offset	RW	Description	Reset Value
RSTSRC	SMC_BA+0x04	R/W	Reset Source Register	0x01

Bits	Flag	Description
[7]	Reserved	**Reserved.**
[6]	LVR	**Low Voltage Reset Source Indication Flag** This bit is set by the low voltage circuit. Software writes 0 to clear this bit. LVR = 1, The reset source is from low voltage reset circuit. LVR = 0, No reset source is from low voltage reset circuit.
[5]	Reserved	**Reserved.**
[4]	WDT	**WatchDog Timer Reset Source Indication Flag** This bit is set by the "reset signal" from watchdog timer controller to indicate that the reset source is from watch dog timer controller. Software writes 0 to clear this bit. WDT = 1, The reset source is from watch dog timer controller. WDT = 0, No reset source is from watch dog timer controller.
[3]	CPU	**CPU Reset Source Indication Flag** This bit is set by the "reset signal" if software writes CPUR bit (SYSRSTC[1]). CPU = 1, The reset source is from CPU reset. CPU = 0, No reset source is from CPU reset.
[2]	CHIP	**Chip Reset Source Indication Flag** This bit is set by the "reset signal" if software writes CHIPR bit (SYSRSTC[0]). CHIP = 1, The reset source is from CHIP reset. CHIP = 0, No reset source is from CHIP reset.
[1]	EXT	**Active Low RESET Pin Reset Source Indication Flag** This bit is set by the "reset signal" from RST pin to indicate that the reset source is from external reset pin. Software writes 0 to clear this bit. EXT = 1, The reset source is from RST pin. EXT = 0, No reset source is from RST pin.
[0]	POR	**Power On Reset Source Indication Flag** This bit is set by the "reset signal" from power on reset circuit to indicate that the reset source is from power on reset circuit. Software writes 0 to clear this bit. POR = 1, The reset source is from power on reset circuit. POR = 0, No reset source is from power on reset circuit.

System Reset Control Register

Register	Offset	RW	Description	Reset Value
SYSRSTC	SMC_BA+0x05	R/W	System Reset Control Register	0x00

Bits	Flag	Description
[7:4]	Reserved	**Reserved.**
[3]	EXTR	**External Reset Enable Flag** This bit is set by software. After this bit field is set, and related PB0 control registers (PBCTRL1 and PBCTRL0 Registers) are set, external reset function is enabled. EXTR = 1, External reset enabled. EXTR = 0, External reset disabled.
[2]	CPURH	**CPU Reset and Hold Flag** This bit is set by software to reset and stop CPU execution. This bit will be cleared by hardware automatically after 3 clock cycles. CPURH = 1, MCU reset and hold. CPURH = 0, MCU normal operation.
[1]	CPUR	**CPU Reset Flag** This bit is set by software to reset MCU. This bit will be cleared by hardware automatically after 3 clock cycles. CPUR = 1, 8051 core reset. CPUR = 0, 8051 core normal operation.
[0]	CHIPR	**Whole Chip Reset Flag** This bit is set by software to reset whole chip like power on reset circuit does. This bit will be cleared by hardware automatically after 3 clock cycles. CHIPR = 1, Whole chip reset. CHIPR = 0, Whole chip normal operation.

Peripheral Reset Control 0 Register

Register	Offset	RW	Description	Reset Value
PERRSTC0	SMC_BA+0x06	R/W	Peripheral Reset Control 0 Register	0x00

Bits	Flag	Description
[7]	UARTR	**UART Controller Reset Flag** This bit is set by software to reset UART controller like power on reset circuit does. This bit will be cleared by hardware automatically after 3 clock cycles. UARTR = 1, UART reset. UARTR = 0, UART normal operation.
[6]	I2CR	**I2C Controller Reset Flag** This bit is set by software to reset I2C controller like power on reset circuit does. This bit will be cleared by hardware automatically after 3 clock cycles. I2CR = 1, I2C reset. I2CR = 0, I2C normal operation.
[5]	Reserved	Reserved.
[4]	RTCR	**RTC Controller Reset Flag** This bit is set by software to reset RTC controller like power on reset circuit does. This bit will be cleared by hardware automatically after 3 clock cycles. RTCR = 1, RTC reset. RTCR = 0, RTC normal operation.
[3]	Reserved	Reserved.
[2]	LCMR	**LCD Controller Reset Flag** This bit is set by software to reset STN LCD controller like power on reset circuit does. This bit will be cleared by hardware automatically after 3 clock cycles. LCMR = 1, LCM reset. LCMR = 0, LCM normal operation.
[1]	ANAR	**Analog IP Controller Reset Flag** This bit is set by software to reset analog controller like power on reset circuit does. This bit will be cleared by hardware automatically after 3 clock cycles. ANAR = 1, ANAC reset. ANAR = 0, ANAC normal operation.
[0]	SDMR	**Sigma-Delta Reset Flag** This bit is set by software to reset delta sigma ADC like power on reset circuit does. This bit will be cleared by hardware automatically after 3 clock cycles. SDMR = 1, Sigma-delta reset. SDMR = 0, Sigma-delta normal operation.

Peripheral Reset Control 1 Register

Register	Offset	RW	Description	Reset Value
PERRSTC1	SMC_BA+0x07	R/W	Peripheral Reset Control 1 Register	0x00

Bits	Flag	Description
[7:6]	Reserved	Reserved.
[5]	CAPR	**CAP Reset Flag** This bit is set by software to reset CAP like power on reset circuit does. This bit will be cleared by hardware automatically after 3 clock cycles. CAPR = 1, CAP reset. CAPR = 0, CAP normal operation.
[4]	MFAR	**MFA Reset Flag** This bit is set by software to reset MFA like power on reset circuit does. This bit will be cleared by hardware automatically after 3 clock cycles. MFAR = 1, MFA reset. MFAR = 0, MFA normal operation.
[3]	SPIR	**SPI Reset Flag** This bit is set by software to reset SPI like power on reset circuit does. This bit will be cleared by hardware automatically after 3 clock cycles. SPIR = 1, SPI reset. SPIR = 0, SPI normal operation.
[2]	TMR2R	**Timer 2 Reset Flag** This bit is set by software to reset Timer 2 like power on reset circuit does. This bit will be cleared by hardware automatically after 3 clock cycles. TMR2R = 1, Timer 2 reset. TMR2R = 0, Timer 2 normal operation.
[1]	TMR1R	**Timer 1 Reset Flag** This bit is set by software to reset Timer 1 like power on reset circuit does. This bit will be cleared by hardware automatically after 3 clock cycles. TMR1R = 1, Timer 1 reset. TMR1R = 0, Timer 1 normal operation.
[0]	TMR0R	**Timer 0 Reset Flag** This bit is set by software to reset Timer 0 like power on reset circuit does. This bit will be cleared by hardware automatically after 3 clock cycles. TMR0R = 1, Timer 0 reset. TMR0R = 0, Timer 0 normal operation.

PA Control Register

Register	Offset	RW	Description	Reset Value
PACTRL0	SMC_BA+0x08	R/W	PA Control 0 Register	0x00

Register	Offset	RW	Description	Reset Value
PACTRL1	SMC_BA+0x09	R/W	PA Control 1 Register	0x00

PA Num.	Description
PA0	The PACTRL0[0], and PACTRL1[0] are concatenated to distinguish the PA0 function. Function / PACTRL1[0] / PACTRL0[0] PA0 / 0 / 0 MISO / 0 / 1 Reserved / 1 / 0 Reserved / 1 / 1
PA1	The PACTRL0[1], PACTRL1[1] are concatenated to distinguish the PA1 function. Function / PACTRL1[1] / PACTRL0[1] PA1 / 0 / 0 MOSI / 0 / 1 Reserved / 1 / 0 Reserved / 1 / 1
PA2	The PACTRL0[2], PACTRL1[2] are concatenated to distinguish the PA2 function. Function / PACTRL1[2] / PACTRL0[2] PA2 / 0 / 0 SCLK / 0 / 1 INT3 / 1 / 0 Reserved / 1 / 1
PA3	The PACTRL0[3], PACTRL1[3] are concatenated to distinguish the PA3 function. Function / PACTRL1[3] / PACTRL0[3] PA3 / 0 / 0 SS / 0 / 1 INT2 / 1 / 0 Reserved / 1 / 1
PA4	The PACTRL0[4], PACTRL1[4] are concatenated to distinguish the PA4 function. Function / PACTRL1[4] / PACTRL0[4] PA4 / 0 / 0 SDA / 0 / 1 Reserved / 1 / 0 Reserved / 1 / 1
PA5	The PACTRL0[5], PACTRL1[5] are concatenated to distinguish the PA5 function. Function / PACTRL1[5] / PACTRL0[5] PA5 / 0 / 0 SCL / 0 / 1 INT1 / 1 / 0 Reserved / 1 / 1
PA6	The PACTRL0[6], PACTRL1[6] are concatenated to distinguish the PA6 function. Function / PACTRL1[6] / PACTRL0[6] PA6 / 0 / 0 RXD / 0 / 1 Reserved / 1 / 0 Reserved / 1 / 1
PA7	The PACTRL0[7], PACTRL1[7] are concatenated to distinguish the PA7 function. Function / PACTRL1[7] / PACTRL0[7] PA7 / 0 / 0 TXD / 0 / 1 Reserved / 1 / 0 Reserved / 1 / 1

PA Mode Selection Register

The PAMS0, PAMS1 and PAMS2 are used to selection one of the five GPIO modes, which are input mode, push-pull mode, open-drain mode, internal pull-up mode, and high-impedance mode.

Register	Offset	RW	Description	Reset Value
PAMS0	SMC_BA+0x0C	R/W	PA Mode Select 0 Register	0x00

Register	Offset	RW	Description	Reset Value
PAMS1	SMC_BA+0x0D	R/W	PA Mode Select 1 Register	0x00

Register	Offset	RW	Description	Reset Value
PAMS2	SMC_BA+0x0E	R/W	PA Mode Select 2 Register	0x00

PA Num.	Description
PA0	The PAMS0[0], PAMS1[0], and PSMS2[0] are concatenated to distinguish the PA0 IO function.

Function	PAMS2[0]	PAMS1[0]	PAMS0[0]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PA1: The PAMS0[1], PAMS1[1], and PAMS2[1] are concatenated to distinguish the PA1 IO function.

Function	PAMS2[1]	PAMS1[1]	PAMS0[1]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PA2: The PAMS0[2], PAMS1[2], and PAMS2[2] are concatenated to distinguish the PA2 IO function.

Function	PAMS2[2]	PAMS1[2]	PAMS0[2]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PA3: The PAMS0[3], PAMS1[3], and PAMS2[3] are concatenated to distinguish the PA3 IO function.

Function	PAMS2[3]	PAMS1[3]	PAMS0[3]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PA4: The PAMS0[4], PAMS1[4], and PAMS2[4] are concatenated to distinguish the PA4 IO function.

Function	PAMS2[4]	PAMS1[4]	PAMS0[4]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PA5: The PAMS0[5], PAMS1[5], and PAMS2[5] are concatenated to distinguish the PA5 IO function.

Function	PAMS2[5]	PAMS1[5]	PAMS0[5]

	Input	0	0	0
	Push-pull	0	0	1
	Open-drain	0	1	0
	Internal pull-up	0	1	1
	High-impendence	1	0	0

PA6

The PAMS0[6], PAMS1[6], and PAMS2[6] are concatenated to distinguish the PA6 IO function.

Function	PAMS2[6]	PAMS1[6]	PAMS0[6]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PA7

The PAMS0[7], PAMS1[7], and PAMS2[7] are concatenated to distinguish the PA7 IO function.

Function	PAMS2[7]	PAMS1[7]	PAMS0[7]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PA Bit Out Register

Register	Offset	RW	Description	Reset Value
PADO0	SMC_BA+0x0F	R/W	PA0 Bit Output Register	0x00
PADO1	SMC_BA+0x10	R/W	PA1 Bit Output Register	0x00
PADO2	SMC_BA+0x11	R/W	PA2 Bit Output Register	0x00
PADO3	SMC_BA+0x12	R/W	PA3 Bit Output Register	0x00
PADO4	SMC_BA+0x13	R/W	PA4 Bit Output Register	0x00
PADO5	SMC_BA+0x14	R/W	PA5 Bit Output Register	0x00
PADO6	SMC_BA+0x15	R/W	PA6 Bit Output Register	0x00
PADO7	SMC_BA+0x16	R/W	PA7 Bit Output Register	0x00

Bits	Flag	Description
[7:1]	Reserved	**Reserved.**
[0]	PADOn n = 0 ~ 7	**PA I/O Pin Bit Output Control** Writing this bit can control one PA pin output value. 1 = Set the corresponding PA pin to high. 0 = Set the corresponding PA pin to low. For example: Writing PADO0 will reflect the written value to bit PADOUT[0].

PA Data Out Write Mask Register

Register	Offset	RW	Description	Reset Value
PAMSK	SMC_BA+0x17	R/W	PA Data Out Write Mask Register	0x00

Bits	Flag	Description
[7:0]	PAxMSK x = 0 ~ 7	**Pax Data Output Write Mask** 1 = PA pin[x] is protected. 0 = PA pin[x] is able to be updated.

PA Data Out Register

Register	Offset	RW	Description	Reset Value
PADOUT	SMC_BA+0x18	R/W	PA Data Out Register	0x00

Bits	Flag	Description
[7:0]	PADOUT[x] x = 0 ~ 7	**PA Data Out** 1 = PA pin[x] will drive “HIGH” if the PA pin is configured as output, open-drain. 0 = PA pin[x] will drive “LOW” if the PA pin is configured as output, open-drain.

PA Pin Value Register

Register	Offset	RW	Description	Reset Value
PAPIN	SMC_BA+0x19	R/W	PA PIN Value Register	0x00

Bits	Flag	Description
[7:0]	PAPIN[x] x = 0 ~ 7	**PA Pin Value** 1 = Indicates PA pin[x] value is 1. 0 = Indicates PA pin[x] value is 0.

PB Control Register

Register	Offset	RW	Description	Reset Value
PBCTRL0	SMC_BA+0x1A	R/W	PB Control 0 Register	0x00

Register	Offset	RW	Description	Reset Value
PBCTRL1	SMC_BA+0x1B	R/W	PB Control 1 Register	0x00

PB Num.	Description
PB0	The PBCTRL0[0], and PBCTRL1[0] are concatenated to distinguish the PB0 function. Function / PBCTRL1[0] / PBCTRL0[0] PB0 / 0 / 0 RST / 0 / 1 Reserved / 1 / 0 SEG19 / 1 / 1
PB1	The PBCTRL0[1], PBCTRL1[1] are concatenated to distinguish the PB1 function. Function / PBCTRL1[1] / PBCTRL0[1] PB1 / 0 / 0 XTALOUT / 0 / 1 Reserved / 1 / 0 SEG18 / 1 / 1
PB2	The PBCTRL0[2], PBCTRL1[2] are concatenated to distinguish the PB2 function. Function / PBCTRL1[2] / PBCTRL0[2] PB2 / 0 / 0 XTALIN / 0 / 1 Reserved / 1 / 0 SEG17 / 1 / 1
PB3	The PBCTRL0[3], PBCTRL1[3] are concatenated to distinguish the PB3 function. Function / PBCTRL1[3] / PBCTRL0[3] PB3 / 0 / 0 Reserved / 0 / 1 INT0 / 1 / 0 SEG16 / 1 / 1
PB4	The PBCTRL0[4], PBCTRL1[4] are concatenated to distinguish the PB4 function. Function / PBCTRL1[4] / PBCTRL0[4] PB4 / 0 / 0 PWM0 / 0 / 1 Reserved / 1 / 0 SEG15 / 1 / 1
PB5	The PBCTRL0[5], PBCTRL1[5] are concatenated to distinguish the PB5 function. Function / PBCTRL1[5] / PBCTRL0[5] PB5 / 0 / 0 PWM1 / 0 / 1 Reserved / 1 / 0 SEG14 / 1 / 1
PB6	The PBCTRL0[6], PBCTRL1[6] are concatenated to distinguish the PB6 function. Function / PBCTRL1[6] / PBCTRL0[6] PB6 / 0 / 0 PWM2 / 0 / 1 Reserved / 1 / 0 SEG13 / 1 / 1
PB7	The PBCTRL0[7], PBCTRL1[7] are concatenated to distinguish the PB7 function. Function / PBCTRL1[7] / PBCTRL0[7] PB7 / 0 / 0 PWM3 / 0 / 1 Reserved / 1 / 0 SEG12 / 1 / 1

PB Mode Selection Register

The PBMS0, PBMS1 and PBMS2 are used to selection one of the five GPIO modes, which are input mode, push-pull mode, open-drain mode, internal pull-up mode, and high-impedance mode.

Register	Offset	RW	Description	Reset Value
PBMS0	SMC_BA+0x1E	R/W	PB Mode Select 0 Register	0x00

Register	Offset	RW	Description	Reset Value
PBMS1	SMC_BA+0x1F	R/W	PB Mode Select 1 Register	0x00

Register	Offset	RW	Description	Reset Value
PBMS2	SMC_BA+0x20	R/W	PB Mode Select 2 Register	0x00

PB Num.	Description
PB0	The PBMS0[0], PBMS1[0], and PSMS2[0] are concatenated to distinguish the PB0 IO function.

Function	PBMS2[0]	PBMS1[0]	PBMS0[0]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PB1: The PBMS0[1], PBMS1[1], and PBMS2[1] are concatenated to distinguish the PB1 IO function.

Function	PBMS2[1]	PBMS1[1]	PBMS0[1]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PB2: The PBMS0[2], PBMS1[2], and PBMS2[2] are concatenated to distinguish the PB2 IO function.

Function	PBMS2[2]	PBMS1[2]	PBMS0[2]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PB3: The PBMS0[3], PBMS1[3], and PBMS2[3] are concatenated to distinguish the PB3 IO function.

Function	PBMS2[3]	PBMS1[3]	PBMS0[3]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PB4: The PBMS0[4], PBMS1[4], and PBMS2[4] are concatenated to distinguish the PB4 IO function.

Function	PBMS2[4]	PBMS1[4]	PBMS0[4]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PB5: The PBMS0[5], PBMS1[5], and PBMS2[5] are concatenated to distinguish the PB5 IO function.

Function	PBMS2[5]	PBMS1[5]	PBMS0[5]

	Input	0	0	0
	Push-pull	0	0	1
	Open-drain	0	1	0
	Internal pull-up	0	1	1
	High-impendence	1	0	0

PB6

The PBMS0[6], PBMS1[6], and PBMS2[6] are concatenated to distinguish the PB6 IO function.

Function	PBMS2[6]	PBMS1[6]	PBMS0[6]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PB7

The PBMS0[7], PBMS1[7], and PBMS2[7] are concatenated to distinguish the PB7 IO function.

Function	PBMS2[7]	PBMS1[7]	PBMS0[7]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PB Bit Out Register

Register	Offset	RW	Description	Reset Value
PBDO0	SMC_BA+0x21	R/W	PB0 Bit Output Register	0x00
PBDO1	SMC_BA+0x22	R/W	PB1 Bit Output Register	0x00
PBDO2	SMC_BA+0x23	R/W	PB2 Bit Output Register	0x00
PBDO3	SMC_BA+0x24	R/W	PB3 Bit Output Register	0x00
PBDO4	SMC_BA+0x25	R/W	PB4 Bit Output Register	0x00
PBDO5	SMC_BA+0x26	R/W	PB5 Bit Output Register	0x00
PBDO6	SMC_BA+0x27	R/W	PB6 Bit Output Register	0x00
PBDO7	SMC_BA+0x28	R/W	PB7 Bit Output Register	0x00

Bits	Flag	Description
[7:1]	Reserved	**Reserved.**
[0]	PBDOn n = 0 ~ 7	**PB I/O Pin Bit Output Control** Writing this bit can control one PB pin output value. 1 = Set the corresponding PB pin to high. 0 = Set the corresponding PB pin to low. For example: Writing PBDO0 will reflect the written value to bit PBDOUT[0].

PB Data Out Write Mask Register

Register	Offset	RW	Description	Reset Value
PBMSK	SMC_BA+0x29	R/W	PB Data Out Write Mask Register	0x00

Bits	Flag	Description
[7:0]	PBxMSK x = 0 ~ 7	**PBx Data Output Write Mask** 1 = PB pin[x] is protected. 0 = PB pin[x] is able to be updated.

PB Data Out Register

Register	Offset	RW	Description	Reset Value
PBDOUT	SMC_BA+0x2A	R/W	PB Data Out Register	0x00

Bits	Flag	Description
[7:0]	PBDOUT[x] x = 0 ~ 7	**PB Data Out** 1 = PB pin[x] will drive “HIGH” if the PB pin is configured as output, open-drain. 0 = PB pin[x] will drive “LOW” if the PB pin is configured as output, open-drain.

PB Pin Value Register

Register	Offset	RW	Description	Reset Value
PBPIN	SMC_BA+0x2B	R/W	PB PIN Value Register	0x00

Bits	Flag	Description
[7:0]	PBPIN[x] x = 0 ~ 7	**PB Pin Value** 1 = Indicates PB pin[x] value is 1. 0 = Indicates PB pin[x] value is 0.

PC Control Register

Register	Offset	RW	Description	Reset Value
PCCTRL0	SMC_BA+0x2C	R/W	PC Control 0 Register	0x00

Register	Offset	RW	Description	Reset Value
PCCTRL1	SMC_BA+0x2D	R/W	PC Control 1 Register	0x00

<table>
<tr><th>PC Num.</th><th>Description</th></tr>
<tr><td>PC0</td><td>The PCCTRL0[0], and PCCTRL1[0] are concatenated to distinguish the PC0 function.
<table>
<tr><th>Function</th><th>PCCTRL1[0]</th><th>PCCTRL0[0]</th></tr>
<tr><td>PC0</td><td>0</td><td>0</td></tr>
<tr><td>PWM4</td><td>0</td><td>1</td></tr>
<tr><td>Reserved</td><td>1</td><td>0</td></tr>
<tr><td>SEG11</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>PC1</td><td>The PCCTRL0[1], PCCTRL1[1] are concatenated to distinguish the PC1 function.
<table>
<tr><th>Function</th><th>PCCTRL1[1]</th><th>PCCTRL0[1]</th></tr>
<tr><td>PC1</td><td>0</td><td>0</td></tr>
<tr><td>PWM5</td><td>0</td><td>1</td></tr>
<tr><td>Reserved</td><td>1</td><td>0</td></tr>
<tr><td>SEG10</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>PC2</td><td>The PCCTRL0[2], PCCTRL1[2] are concatenated to distinguish the PC2 function.
<table>
<tr><th>Function</th><th>PCCTRL1[2]</th><th>PCCTRL0[2]</th></tr>
<tr><td>PC2</td><td>0</td><td>0</td></tr>
<tr><td>nCTS</td><td>0</td><td>1</td></tr>
<tr><td>OCD_SWD</td><td>1</td><td>0</td></tr>
<tr><td>SEG09</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>PC3</td><td>The PCCTRL0[3], PCCTRL1[3] are concatenated to distinguish the PC3 function.
<table>
<tr><th>Function</th><th>PCCTRL1[3]</th><th>PCCTRL0[3]</th></tr>
<tr><td>PC3</td><td>0</td><td>0</td></tr>
<tr><td>nRTS</td><td>0</td><td>1</td></tr>
<tr><td>OCD_SCK</td><td>1</td><td>0</td></tr>
<tr><td>SEG08</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>PC4</td><td>The PCCTRL0[4], PCCTRL1[4] are concatenated to distinguish the PC4 function.
<table>
<tr><th>Function</th><th>PCCTRL1[4]</th><th>PCCTRL0[4]</th></tr>
<tr><td>PC4</td><td>0</td><td>0</td></tr>
<tr><td>Reserved</td><td>0</td><td>1</td></tr>
<tr><td>ISP_SDA</td><td>1</td><td>0</td></tr>
<tr><td>SEG07</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>PC5</td><td>The PCCTRL0[5], PCCTRL1[5] are concatenated to distinguish the PC5 function.
<table>
<tr><th>Function</th><th>PCCTRL1[5]</th><th>PCCTRL0[5]</th></tr>
<tr><td>PC5</td><td>0</td><td>0</td></tr>
<tr><td>Reserved</td><td>0</td><td>1</td></tr>
<tr><td>ISP_SCL</td><td>1</td><td>0</td></tr>
<tr><td>SEG06</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>PC6</td><td>The PCCTRL0[6], PCCTRL1[6] are concatenated to distinguish the PC6 function.
<table>
<tr><th>Function</th><th>PCCTRL1[6]</th><th>PCCTRL0[6]</th></tr>
<tr><td>PC6</td><td>0</td><td>0</td></tr>
<tr><td>Reserved</td><td>0</td><td>1</td></tr>
<tr><td>CMPP</td><td>1</td><td>0</td></tr>
<tr><td>SEG05</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>PC7</td><td>The PCCTRL0[7], PCCTRL1[7] are concatenated to distinguish the PC7 function.
<table>
<tr><th>Function</th><th>PCCTRL1[7]</th><th>PCCTRL0[7]</th></tr>
<tr><td>PC7</td><td>0</td><td>0</td></tr>
<tr><td>Reserved</td><td>0</td><td>1</td></tr>
<tr><td>CMPN</td><td>1</td><td>0</td></tr>
<tr><td>SEG04</td><td>1</td><td>1</td></tr>
</table></td></tr>
</table>

PC Mode Selection Register

The PCMS0, PCMS1 and PCMS2 are used to selection one of the five GPIO modes, which are input mode, push-pull mode, open-drain mode, internal pull-up mode, and high-impedance mode.

Register	Offset	RW	Description	Reset Value
PCMS0	SMC_BA+0x30	R/W	PC Mode Select 0 Register	0x00

Register	Offset	RW	Description	Reset Value
PCMS1	SMC_BA+0x31	R/W	PC Mode Select 1 Register	0x00

Register	Offset	RW	Description	Reset Value
PCMS2	SMC_BA+0x32	R/W	PC Mode Select 2 Register	0x00

PC Num.	Description
PC0	The PCMS0[0], PCMS1[0], and PSMS2[0] are concatenated to distinguish the PC0 IO function.

Function	PCMS2[0]	PCMS1[0]	PCMS0[0]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PC1: The PCMS0[1], PCMS1[1], and PCMS2[1] are concatenated to distinguish the PC1 IO function.

Function	PCMS2[1]	PCMS1[1]	PCMS0[1]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PC2: The PCMS0[2], PCMS1[2], and PCMS2[2] are concatenated to distinguish the PC2 IO function.

Function	PCMS2[2]	PCMS1[2]	PCMS0[2]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PC3: The PCMS0[3], PCMS1[3], and PCMS2[3] are concatenated to distinguish the PC3 IO function.

Function	PCMS2[3]	PCMS1[3]	PCMS0[3]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PC4: The PCMS0[4], PCMS1[4], and PCMS2[4] are concatenated to distinguish the PC4 IO function.

Function	PCMS2[4]	PCMS1[4]	PCMS0[4]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PC5: The PCMS0[5], PCMS1[5], and PCMS2[5] are concatenated to distinguish the PC5 IO function.

Function	PCMS2[5]	PCMS1[5]	PCMS0[5]

	Input	0	0	0
	Push-pull	0	0	1
	Open-drain	0	1	0
	Internal pull-up	0	1	1
	High-impendence	1	0	0

PC6

The PCMS0[6], PCMS1[6], and PCMS2[6] are concatenated to distinguish the PC6 IO function.

Function	PCMS2[6]	PCMS1[6]	PCMS0[6]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PC7

The PCMS0[7], PCMS1[7], and PCMS2[7] are concatenated to distinguish the PC7 IO function.

Function	PCMS2[7]	PCMS1[7]	PCMS0[7]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PC Bit Out Register

Register	Offset	RW	Description	Reset Value
PCDO0	SMC_BA+0x33	R/W	PC0 Bit Output Register	0x00
PCDO1	SMC_BA+0x34	R/W	PC1 Bit Output Register	0x00
PCDO2	SMC_BA+0x35	R/W	PC2 Bit Output Register	0x00
PCDO3	SMC_BA+0x36	R/W	PC3 Bit Output Register	0x00
PCDO4	SMC_BA+0x37	R/W	PC4 Bit Output Register	0x00
PCDO5	SMC_BA+0x38	R/W	PC5 Bit Output Register	0x00
PCDO6	SMC_BA+0x39	R/W	PC6 Bit Output Register	0x00
PCDO7	SMC_BA+0x3A	R/W	PC7 Bit Output Register	0x00

Bits	Flag	Description
[7:1]	Reserved	**Reserved.**
[0]	PCDOn n = 0 ~ 7	**PC I/O Pin Bit Output Control** Writing this bit can control one PC pin output value. 1 = Set the corresponding PC pin to high. 0 = Set the corresponding PC pin to low. For example: Writing PCDO0 will reflect the written value to bit PCDOUT[0].

PC Data Out Write Mask Register

Register	Offset	RW	Description	Reset Value
PCMSK	SMC_BA+0x3B	R/W	PC Data Out Write Mask Register	0x00

Bits	Flag	Description
[7:0]	PCxMSK x = 0 ~ 7	**PCx Data Output Write Mask** 1 = PC pin[x] is protected. 0 = PC pin[x] is able to be updated.

PC Data Out Register

Register	Offset	RW	Description	Reset Value
PCDOUT	SMC_BA+0x3C	R/W	PC Data Out Register	0x00

Bits	Flag	Description
[7:0]	PCDOUT[x] x = 0 ~ 7	**PC Data Out** 1 = PC pin[x] will drive “HIGH” if the PC pin is configured as output, open-drain. 0 = PC pin[x] will drive “LOW” if the PC pin is configured as output, open-drain.

PC Pin Value Register

Register	Offset	RW	Description	Reset Value
PCPIN	SMC_BA+0x3D	R/W	PC PIN Value Register	0x00

Bits	Flag	Description
[7:0]	PCPIN[x] x = 0 ~ 7	**PC Pin Value** 1 = Indicates PC pin[x] value is 1. 0 = Indicates PC pin[x] value is 0.

PD Control Register

Register	Offset	RW	Description	Reset Value
PDCTRL0	SMC_BA+0x3E	R/W	PD Control 0 Register	0x00

Register	Offset	RW	Description	Reset Value
PDCTRL1	SMC_BA+0x3F	R/W	PD Control 1 Register	0x00

PD Num.	Description
PD0	The PDCTRL0[0], and PDCTRL1[0] are concatenated to distinguish the PD0 function. Function / PDCTRL1[0] / PDCTRL0[0] PD0 / 0 / 0 CAPIN0 / 0 / 1 CMPO / 1 / 0 SEG03 / 1 / 1
PD1	The PDCTRL0[1], PDCTRL1[1] are concatenated to distinguish the PD1 function. Function / PDCTRL1[1] / PDCTRL0[1] PD1 / 0 / 0 CAPIN1 / 0 / 1 OPAP / 1 / 0 SEG02 / 1 / 1
PD2	The PDCTRL0[2], PDCTRL1[2] are concatenated to distinguish the PD2 function. Function / PDCTRL1[2] / PDCTRL0[2] PD2 / 0 / 0 CAPIN2 / 0 / 1 OPAP / 1 / 0 SEG01 / 1 / 1
PD3	The PDCTRL0[3], PDCTRL1[3] are concatenated to distinguish the PD3 function. Function / PDCTRL1[3] / PDCTRL0[3] PD3 / 0 / 0 CAPIN3 / 0 / 1 OPAN / 1 / 0 SEG00 / 1 / 1
PD4	The PDCTRL0[4], PDCTRL1[4] are concatenated to distinguish the PD4 function. Function / PDCTRL1[4] / PDCTRL0[4] PD4 / 0 / 0 Reserved / 0 / 1 AP0 / 1 / 0 COM3 / 1 / 1
PD5	The PDCTRL0[5], PDCTRL1[5] are concatenated to distinguish the PD5 function. Function / PDCTRL1[5] / PDCTRL0[5] PD5 / 0 / 0 Reserved / 0 / 1 AN0 / 1 / 0 COM2 / 1 / 1
PD6	The PDCTRL0[6], PDCTRL1[6] are concatenated to distinguish the PD6 function. Function / PDCTRL1[6] / PDCTRL0[6] PD6 / 0 / 0 Reserved / 0 / 1 AP1 / 1 / 0 COM1 / 1 / 1
PD7	The PDCTRL0[7], PDCTRL1[7] are concatenated to distinguish the PD7 function. Function / PDCTRL1[7] / PDCTRL0[7] PD7 / 0 / 0 Reserved / 0 / 1 AN1 / 1 / 0 COM0 / 1 / 1

PD Mode Selection Register

The PDMS0, PDMS1 and PDMS2 are used to selection one of the five GPIO modes, which are input mode, push-pull mode, open-drain mode, internal pull-up mode, and high-impedance mode.

Register	Offset	RW	Description	Reset Value
PDMS0	SMC_BA+0x42	R/W	PD Mode Select 0 Register	0x00

Register	Offset	RW	Description	Reset Value
PDMS1	SMC_BA+0x43	R/W	PD Mode Select 1 Register	0x00

Register	Offset	RW	Description	Reset Value
PDMS2	SMC_BA+0x44	R/W	PD Mode Select 2 Register	0x00

PD Num.	Description
PD0	The PDMS0[0], PDMS1[0], and PSMS2[0] are concatenated to distinguish the PD0 IO function.

Function	PDMS2[0]	PDMS1[0]	PDMS0[0]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PD1: The PDMS0[1], PDMS1[1], and PDMS2[1] are concatenated to distinguish the PD1 IO function.

Function	PDMS2[1]	PDMS1[1]	PDMS0[1]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PD2: The PDMS0[2], PDMS1[2], and PDMS2[2] are concatenated to distinguish the PD2 IO function.

Function	PDMS2[2]	PDMS1[2]	PDMS0[2]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PD3: The PDMS0[3], PDMS1[3], and PDMS2[3] are concatenated to distinguish the PD3 IO function.

Function	PDMS2[3]	PDMS1[3]	PDMS0[3]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PD4: The PDMS0[4], PDMS1[4], and PDMS2[4] are concatenated to distinguish the PD4 IO function.

Function	PDMS2[4]	PDMS1[4]	PDMS0[4]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PD5: The PDMS0[5], PDMS1[5], and PDMS2[5] are concatenated to distinguish the PD5 IO function.

Function	PDMS2[5]	PDMS1[5]	PDMS0[5]

	Input	0	0	0
	Push-pull	0	0	1
	Open-drain	0	1	0
	Internal pull-up	0	1	1
	High-impendence	1	0	0

PD6

The PDMS0[6], PDMS1[6], and PDMS2[6] are concatenated to distinguish the PD6 IO function.

Function	PDMS2[6]	PDMS1[6]	PDMS0[6]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PD7

The PDMS0[7], PDMS1[7], and PDMS2[7] are concatenated to distinguish the PD7 IO function.

Function	PDMS2[7]	PDMS1[7]	PDMS0[7]
Input	0	0	0
Push-pull	0	0	1
Open-drain	0	1	0
Internal pull-up	0	1	1
High-impendence	1	0	0

PD Bit Out Register

Register	Offset	RW	Description	Reset Value
PDDO0	SMC_BA+0x45	R/W	PD0 Bit Output Register	0x00
PDDO1	SMC_BA+0x46	R/W	PD1 Bit Output Register	0x00
PDDO2	SMC_BA+0x47	R/W	PD2 Bit Output Register	0x00
PDDO3	SMC_BA+0x48	R/W	PD3 Bit Output Register	0x00
PDDO4	SMC_BA+0x49	R/W	PD4 Bit Output Register	0x00
PDDO5	SMC_BA+0x4A	R/W	PD5 Bit Output Register	0x00
PDDO6	SMC_BA+0x4B	R/W	PD6 Bit Output Register	0x00
PDDO7	SMC_BA+0x4C	R/W	PD7 Bit Output Register	0x00

Bits	Flag	Description
[7:1]	Reserved	**Reserved.**
[0]	PDDOn n = 0 ~ 7	**PD I/O Pin Bit Output Control** Writing this bit can control one PD pin output value. 1 = Set the corresponding PD pin to high. 0 = Set the corresponding PD pin to low. For example: Writing PDDO0 will reflect the written value to bit PDDOUT[0].

PD Data Out Write Mask Register

Register	Offset	RW	Description	Reset Value
PDMSK	SMC_BA+0x4D	R/W	PD Data Out Write Mask Register	0x00

Bits	Flag	Description
[7:0]	PDxMSK x = 0 ~ 7	**PDx Data Output Write Mask** 1 = PD pin[x] is protected. 0 = PD pin[x] is able to be updated.

PD Data Out Register

Register	Offset	RW	Description	Reset Value
PDDOUT	SMC_BA+0x4E	R/W	PD Data Out Register	0x00

Bits	Flag	Description
[7:0]	PDDOUT[x] x = 0 ~ 7	**PD Data Out** 1 = PD pin[x] will drive “HIGH” if the PD pin is configured as output, open-drain. 0 = PD pin[x] will drive “LOW” if the PD pin is configured as output, open-drain.

PD Pin Value Register

Register	Offset	RW	Description	Reset Value
PDPIN	SMC_BA+0x4F	R/W	PD PIN Value Register	0x00

Bits	Flag	Description
[7:0]	PDPIN[x] x = 0 ~ 7	**PD Pin Value** 1 = Indicates PD pin[x] value is 1. 0 = Indicates PD pin[x] value is 0.

Timer Stall Control Register

Register	Offset	RW	Description	Reset Value
TMRSTAL	SMC_BA+0x50	R/W	Timer Stall Control Register	0x00

Bits	Flag	Description
[7:3]	Reserved	**Reserved.**
[2]	TMR2ST	**Timer 2 Stall Flag** This bit is set by software and cleared by software. TMR2ST = 1, Timer 2 stall. TMR2ST = 0, Timer 2 normal operation.
[1]	TMR1ST	**Timer 1 Stall Flag** This bit is set by software and cleared by software. TMR1ST = 1, Timer 1 stall. TMR1ST = 0, Timer 1 normal operation.
[0]	TMR0ST	**Timer 0 Stall Flag** This bit is set by software and cleared by software. TMR0ST = 1, Timer 0 stall. TMR0ST = 0, Timer 0 normal operation.

SRAM Configuration Register

Register	Offset	RW	Description	Reset Value
RAMCONF	SMC_BA+0x51	R/W	SRAM Configuration Register	0x02

Bits	Flag	Description
[7:4]	MS	**Not used but register field reserved.**
[3]	Reserved	**Reserved.**
[2]	MSE	Not used but register reserved.
[1:0]	DVS	**SRAM Voltage Selection Flag** This bit is set by software and should not be modified except operation voltage is changed to 2V or below. DVS = 2'b10, 3.3v is selected. DVS = 2'b01, 2.0v or below is selected.

6.5 Clock Controller

6.5.1 Overview

The FE81 series supports 3 clock sources for general applications.

- Build-in 12MHz high speed RC oscillator (HSRC)
- Build-in 32KHz low speed RC oscillator (LSRC)
- External 12MHz/32KHz crystal input (XTAL)
  12MHz or 32KHz is selected by XTALSEL bit field in CLKEN Register.

The clock controller of the FE81 series controls the 3 clock sources listed as follows.

Clock stable status

The clock source stable flag indicates clock source status. After enabling high speed RC oscillator or external crystal, 1024 clock cycles must be maintained waiting for the stability of the clock source. After enabling low speed oscillator, 16-cycles must be maintained to wait for the stability of the clock source.

Clock selection control

System clock source is selectable from one of the three clock source. Moreover, several sources of peripheral clocks can be selected providing flexibilities for various applications. Please refer to Section 6.5.2 for detailed descriptions.

High performance bus (HPB) clock control

The clocks of HPB peripherals, such as 8051 core, HPB, on-chip SRAM, and FMC, are all controlled by the clock controller.

Peripheral bus (PFB) clock control

The clocks of PFB peripherals are all controlled by the clock controller.

Clock division control

System clock and several peripheral clocks can be divided to meet the requirements of various applications.

Low power mode control

Rich low power modes are supported by FE81 series. Please refer to Section 6.5.3 for detailed descriptions.

6.5.2 Clock Plan Block Diagram

The block diagram of clock plan is illustrated in Figure 6.5-1, and Figure 6.5-2. System clock (FSRC) is selected from one of the 3 clock sources, and can be divided by MCU_N. The HPB clock (HCLK), used by HPB peripherals, is at the same frequency of the 8051 core. The PFB clock (PCLK), used by PFB peripherals, is divided from HPB clock by PFB_N.

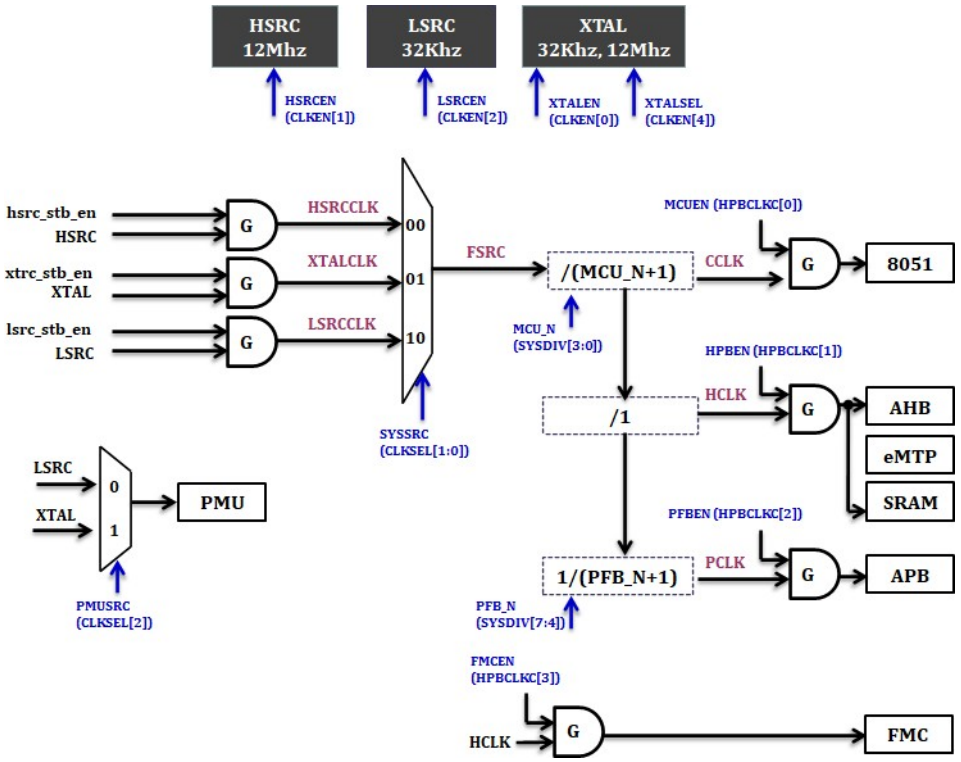


Figure 6.5-1 Clock Architecture of HPB Peripherals

The FE81 series provides flexible control of peripheral clock. For low power purpose, the clock of each peripheral default is gated with corresponding enable flag defined in related register bit field. For example, while GPIO PA port is used, PAEN flag should be set in PFBCLKC1 Register. Figure 6.5-2 illustrates clock architecture of PFB peripherals. Note that while using XTAL as one of the peripheral clock sources for low power mode, the XTAL clock should be set to 32KHz.

Below lists peripherals that using XTAL 32KHz as one of the clock sources. Note that the bit field of WDT clock source selection is defined in WDTCTRL1 Register (CLKSEL flag) in Section 6.10.2.

- Power management unit (PMU)
- Watchdog Timer (WDT)
- Real-time clock (RTC)
- STN LCD controller (LCM)

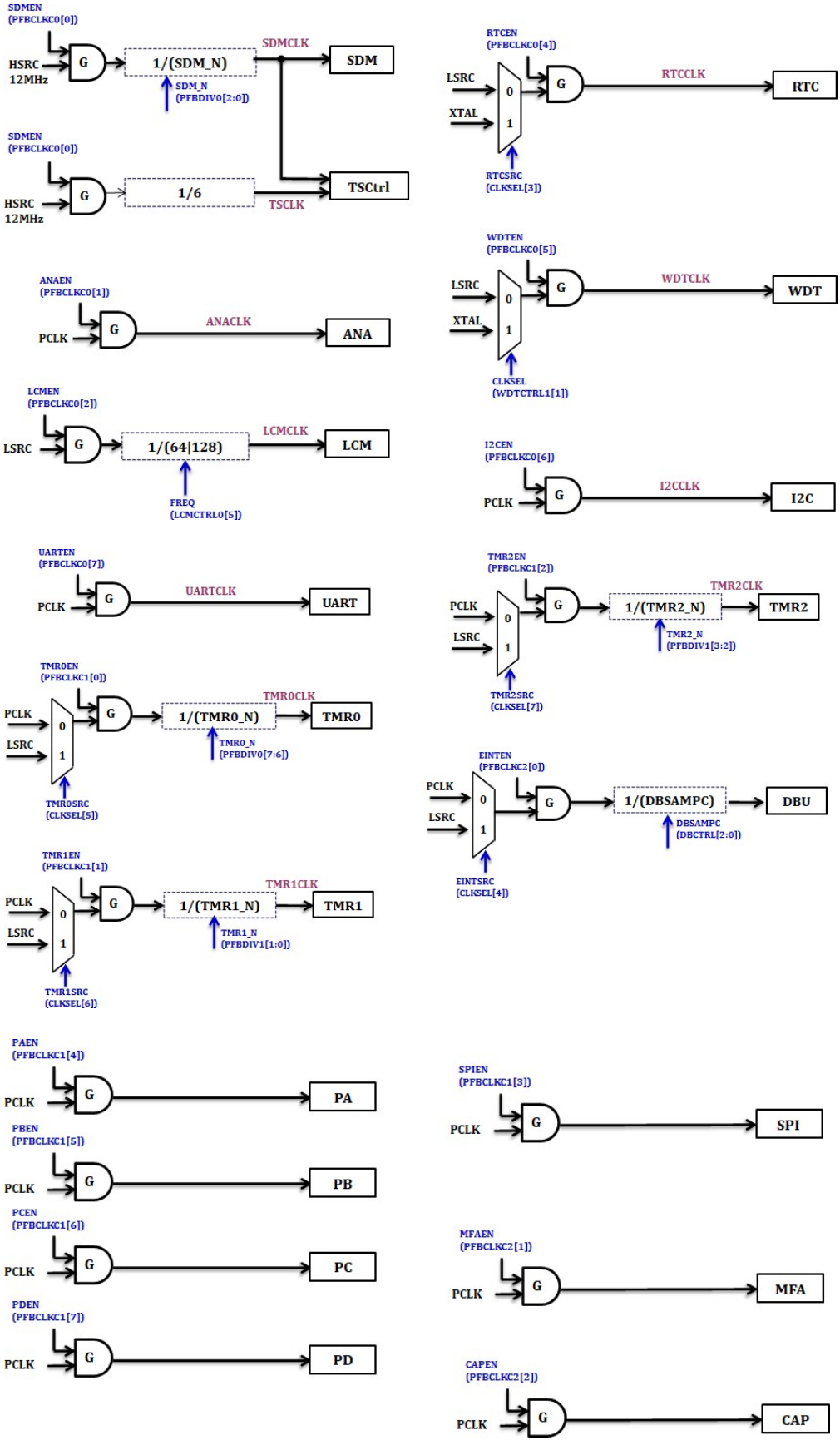


Figure 6.5-2 Clock Architecture of PFB Peripherals

6.5.3 Low Power Mode

The FE81 series supports several low power modes. Table 6.5-1 illustrates the low power mode and related wakeup mechanism.

Power Mode	PMCON	HSRC 8MHz	LSRC 32KHz	XTAL 8M/32K	8051	HPB Periph.	PFB Periph.	Wakeup Mechanism
Normal	00	Enabled	Selected	Selected	CCLK = FSRC	HCLK = CCLK	*PCLK = HCLK/N	None
Slow	00	Selected	Enabled	Selected	CCLK = 32KHz	HCLK = 32KHz	*PCLK = 32KHz	None
Wait	01	Enabled	Selected	Selected	OFF	HCLK = CCLK	*PCLK = HCLK/N	All Reset All Int. RTC Wakeup
Slow Wait	01	Selected	Enabled	Selected	OFF	HCLK = 32KHz	*PCLK = 32KHz	All Reset All Int. RTC Wakeup
Halt	10	Disabled	Enabled	Disabled	OFF	OFF	WDT RTC	Ext. Reset OCD Reset ISP Reset WDT Int. RTC Int. RTC Wakeup Ext. Int.
Shutdown	11	Disabled	Disabled	Disabled	OFF	OFF	OFF	Ext. Reset OCD Reset ISP Reset

Table 6.5-1 Low Power Mode

Figure 6.5-3 depicts the transition of state machine of low power mode. User can reference the diagram to choose the best low power policy for their applications.

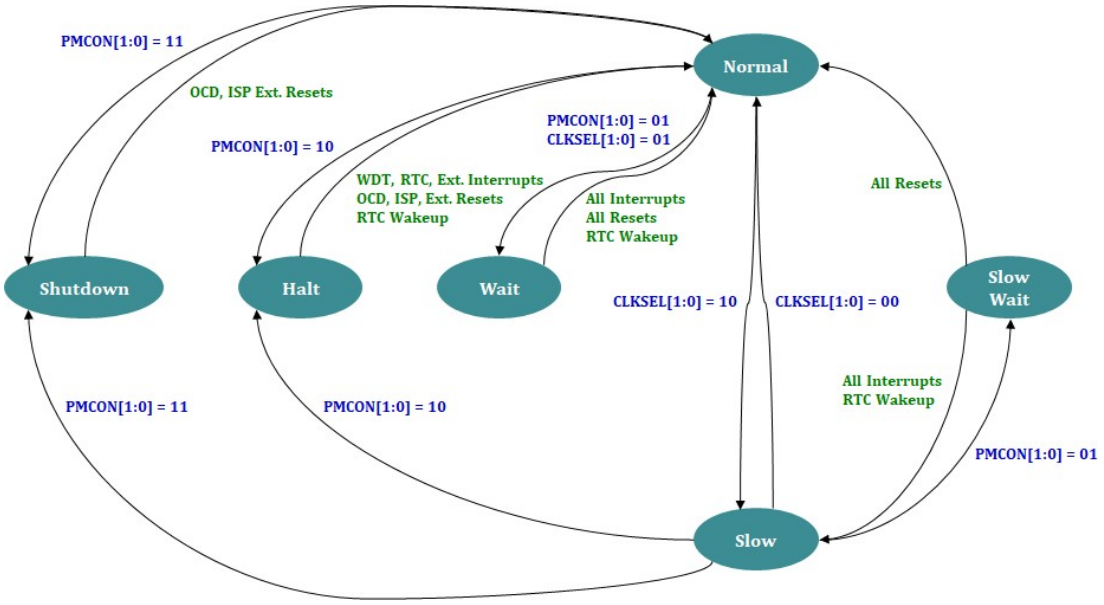


Figure 6.5-3 State Machine of Low Power Mode

Below lists the steps of entering each low power mode.

Slow mode

1. LSRC is on (LSRCEN = 1 in CLKEN Register)
2. Set FSRC source to LSRC (SYSSRC = 2 in CLKSEL Register)
3. HSRC is selected on or off (HSRCEN = 0/1 in CLKEN Register)
4. XTAL is selected on or off (XTALEN = 0/1 in CLKEN Register)

Wait mode

1. LSRC is selected on or off (LSRCEN = 0/1 in CLKEN Register)
2. XTAL is selected on or off (XTALEN = 0/1 in CLKEN Register)
3. Set PMCON to 01 : System clock is selected to HSRC by hardware automatically
4. CCLK is gated
5. CCLK is released by any reset or interrupt and system is returned back to “normal mode” with HSRC enabled

Slow wait mode

1. System enters “slow mode” in advance
2. Set PMCON to 01
3. CCLK is gated
4. CCLK is released by any reset or interrupt and system is returned back to “slow mode” with LSRC enabled

Halt mode

1. WDT/RTC/EINT is enabled in advance

   Note that while using external interrupt for waking up from low power “Halt mode”, the clock source of external interrupt controller must be selected to LSRC as the HSRC and XTAL are disabled by hardware.
2. Set PMCON to 10 : System clock is enabled and selected to LSRC by hardware automatically
3. HSRC is disabled by hardware
4. XTAL is disabled by hardware
5. FSRC, CCLK, HCLK and PCLK is gated
6. Clocks are released by corresponding resets and interrupts (refer to Table 6.5-1), and system is returned back to “normal mode” with HSRC enabled

Shutdown mode

1. Set PMCON to 11
2. All clocks are gated by hardware
3. Clocks are released by external, OCD, and ISP resets

6.5.4 Register Map and Description

Base Address (CLK_BA) : 0xE200				
Register	**Offset**	**RW**	**Description**	**Reset Value**
CLKEN	CLK_BA+0x00	R/W	Clock Enable Control Register	0x07
CLKSTB	CLK_BA+0x01	R	Clock Stable Status Register	0x00
CLKSEL	CLK_BA+0x02	R/W	Clock Select Control Register	0x00
HPBCLKC	CLK_BA+0x03	R/W	HPB Peripheral Clock Control Register	0x07
PFBCLKC0	CLK_BA+0x04	R/W	PFB Peripheral Clock Control 0 Register	0x00
PFBCLKC1	CLK_BA+0x05	R/W	PFB Peripheral Clock Control 1 Register	0x00
PFBCLKC2	CLK_BA+0x06	R/W	PFB Peripheral Clock Control 2 Register	0x00
SYSDIV	CLK_BA+0x07	R/W	System Clock Divider Register	0x00
PFBDIV0	CLK_BA+0x08	R/W	PFB Peripheral Clock Divider 0 Register	0x01
PFBDIV1	CLK_BA+0x09	R/W	PFB Peripheral Clock Divider 1 Register	0x00
PMCON	CLK_BA+0x0A	R/W	Power Mode Control Register	0x00
CLKTRIM	CLK_BA+0x10	R/W	Clock Trim Register	0x00

Clock Enable Control Register

Register	Offset	RW	Description	Reset Value
CLKEN	CLK_BA+0x00	R/W	Clock Enable Control Register	0x07

Bits	Flag	Description
[7:5]	Reserved	**Reserved.**
[4]	XTALSEL	**External Crystal Selection Flag** This bit is able to be set and cleared by software. XTALSEL = 1, External 32.768Mhz is enabled. XTALSEL = 0, External 8Mhz is enabled.
[3]	Reserved	**Reserved.**
[2]	LSRCEN	**Internal Low Speed Oscillator Enable Flag** This bit is set to 1 while internal low speed oscillator is enabled. This bit is able to be set and cleared by software. LSRCEN = 1, Internal low speed oscillator is enabled. LSRCEN = 0, Internal low speed oscillator is disabled.
[1]	HSRCEN	**Internal High Speed Oscillator Enable Flag** This bit is set to 1 while internal high speed oscillator is enabled. This bit is able to be set and cleared by software. HSRCEN = 1, Internal high speed oscillator is enabled. HSRCEN = 0, Internal high speed oscillator is disabled.
[0]	XTALEN	**External Crystal Enable Flag** This bit is set to 1 while external crystal is enabled. This bit is able to be set and cleared by software. XTALEN = 1, External crystal is enabled. XTALEN = 0, External crystal is disabled.

Clock Stable Status Register

Register	Offset	RW	Description	Reset Value
CLKSTB	CLK_BA+0x01	R	Clock Stable Status Register	0x00

Bits	Flag	Description
[7:3]	Reserved	**Reserved.**
[2]	LSRCSTB	**Low Speed Oscillator Clock Stable flag** This bit is read only and set by hardware. LSRCSTB = 1, Low speed oscillator clock is stable. LSRCSTB = 0, Low speed oscillator clock is not stable.
[1]	HSRCSTB	**High Speed Oscillator Clock Stable flag** This bit is read only and set by hardware. HSRCSTB = 1, High speed oscillator clock is stable. HSRCSTB = 0, High speed oscillator clock is not stable.
[0]	XTALSTB	**External Crystal Clock Stable flag** This bit is read only and set by hardware. XTALSTB = 1, External crystal clock is stable. XTALSTB = 0, External crystal clock is not stable.

Clock Select Control Register

Register	Offset	RW	Description	Reset Value
CLKSEL	CLK_BA+0x02	R/W	Clock Select Control Register	0x00

Bits	Flag	Description
[7]	TMR2SRC	**TMR0 Clock Source Select Flag** This bit is set by software, and cleared by software. TMR2SRC = 0, TMR0 clock source is from PCLK. TMR2SRC = 1, TMR0 clock source is from LSRC.
[6]	TMR1SRC	**TMR0 Clock Source Select Flag** This bit is set by software, and cleared by software. TMR1SRC = 0, TMR0 clock source is from PCLK. TMR1SRC = 1, TMR0 clock source is from LSRC.
[5]	TMR0SRC	**TMR0 Clock Source Select Flag** This bit is set by software, and cleared by software. TMR0SRC = 0, TMR0 clock source is from PCLK. TMR0SRC = 1, TMR0 clock source is from LSRC.
[4]	EINTSRC	**External Interrupt Sampling Source Clock Select Flag** This bit is set by software, and cleared by software. While in low power mode, the clock source must be selected to low speed oscillator. EINTSRC = 0, Sampling clock source is from PFB clock. EINTSRC = 1, Sampling clock source is from low speed oscillator.
[3]	RTCSRC	**RTC Clock Source Select Flag** This bit is set by software, and cleared by software. RTCSRC = 0, RTC clock source is from low speed oscillator. RTCSRC = 1, RTC clock source is from low speed crystal oscillator.
[2]	PMUSRC	**Power Management Unit Clock Source Selection flag** This bit is set by software, and cleared by software. PMUSRC = 0, Clock source is from low speed oscillator. PMUSRC = 1, Clock source is from low speed crystal oscillator.
[1:0]	SYSSRC	**FSRC Clock Source Selection flag** This bit is set by software or hardware, and cleared by software or hardware. SYSSRC = 00, Clock source is from high speed oscillator. SYSSRC = 01, Clock source is from external crystal. SYSSRC = 10, Clock source is from low speed oscillator.

HPB Peripheral Clock Control Register

Register	Offset	RW	Description	Reset Value
HPBCLKC	CLK_BA+0x03	R/W	HPB Peripheral Clock Control Register	0x07

Bits	Flag	Description
[7:4]	Reserved	**Reserved.**
[3]	FMCEN	**FMC Clock Enable flag** This bit is set or cleared by software to enable or disable FMC peripheral clock. FMCEN = 1, FMC clock is enabled. FMCEN = 0, FMC clock is disabled.
[2]	PFBEN	**PFB Clock Enable flag** This bit is set or cleared by software to enable or disable PFB peripheral clock. PFBEN = 1, PFB clock is enabled. PFBEN = 0, PFB clock is disabled.
[1]	HPBEN	**HPB Clock Enable flag** This bit is set or cleared by software to enable or disable HPB peripheral clock. HPBEN = 1, HPB clock is enabled. HPBEN = 0, HPB clock is disabled.
[0]	MCUEN	**MCU Clock Enable flag** This bit is set or cleared by software to enable or disable MCU clock. MCUEN = 1, MCU clock is enabled. MCUEN = 0, MCU clock is disabled.

PFB Peripheral Clock Control 0 Register

Register	Offset	RW	Description	Reset Value
PFBCLKC0	CLK_BA+0x04	R/W	PFB Peripheral Clock Control 0 Register	0x00

Bits	Flag	Description
[7]	UARTEN	**UART Clock Enable flag** This bit is set or cleared by software to enable or disable UART clock. UARTEN = 1, UART clock is enabled. UARTEN = 0, UART clock is disabled.
[6]	I2CEN	**I2C Clock Enable flag** This bit is set or cleared by software to enable or disable I2C clock. While setting I2C clock 400KHz, PCLK must at least 4MHz I2CEN = 1, I2C clock is enabled. I2CEN = 0, I2C clock is disabled.
[5]	WDTEN	**WDT Clock Enable flag** This bit is set or cleared by software to enable or disable WDT clock. WDTEN = 1, WDT clock is enabled. WDTEN = 0, WDT clock is disabled.
[4]	RTCEN	**RTC Clock Enable flag** This bit is set or cleared by software to enable or disable RTC clock. RTCEN = 1, RTC clock is enabled. RTCEN = 0, RTC clock is disabled.
[3]	Reserved	**Reserved.**
[2]	LCMCEN	**LCD Controller Clock Enable flag** This bit is set or cleared by software to enable or disable LCD controller clock. LCMCEN = 1, LCD controller clock is enabled. LCMCEN = 0, LCD controller clock is disabled.
[1]	ANAEN	**Analog Controller Clock Enable flag** This bit is set or cleared by software to enable or disable ANAC clock. ANAEN = 1, ANAC clock is enabled. ANAEN = 0, ANAC clock is disabled.
[0]	SDMEN	**Sigma-Delta ADC Controller Clock Enable flag** This bit is set or cleared by software to enable or disable SDM controller clock. SDMEN = 1, SDM clock is enabled. SDMEN = 0, SDM clock is disabled.

PFB Peripheral Clock Control 90 Register

Register	Offset	RW	Description	Reset Value
PFBCLKC1	CLK_BA+0x05	R/W	PFB Peripheral Clock Control 1 Register	0x00

Bits	Flag	Description
[7]	PDEN	**PD Controller Clock Enable flag** This bit is set or cleared by software to enable or disable PD controller clock. PDEN = 1, PD clock is enabled. PDEN = 0, PD clock is disabled.
[6]	PCEN	**PC Controller Clock Enable flag** This bit is set or cleared by software to enable or disable PC controller clock. PCEN = 1, PC clock is enabled. PCEN = 0, PC clock is disabled.
[5]	PBEN	**PB Controller Clock Enable flag** This bit is set or cleared by software to enable or disable PB controller clock. PBEN = 1, PB clock is enabled. PBEN = 0, PB clock is disabled.
[4]	PAEN	**PA Controller Clock Enable flag** This bit is set or cleared by software to enable or disable PA controller clock. PAEN = 1, PA clock is enabled. PAEN = 0, PA clock is disabled.
[3]	SPIEN	**SPI Controller Clock Enable flag** This bit is set or cleared by software to enable or disable SPI controller clock. SPIEN = 1, SPI clock is enabled. SPIEN = 0, SPI clock is disabled.
[2]	TMR2EN	**Timer 2 Controller Clock Enable flag** This bit is set or cleared by software to enable or disable TMR2 controller clock. TMR2EN = 1, Timer 2 clock is enabled. TMR2EN = 0, Timer 2 clock is disabled.
[1]	TMR1EN	**Timer 1 Controller Clock Enable flag** This bit is set or cleared by software to enable or disable TMR1 controller clock. TMR1EN = 1, Timer 1 clock is enabled. TMR1EN = 0, Timer 1 clock is disabled.
[0]	TMR0EN	**Timer 0 Controller Clock Enable flag** This bit is set or cleared by software to enable or disable TMR0 controller clock. TMR0EN = 1, Timer 0 clock is enabled. TMR0EN = 0, Timer 0 clock is disabled.

PFB Peripheral Clock Control 2 Register

Register	Offset	RW	Description	Reset Value
PFBCLKC2	CLK_BA+0x06	R/W	PFB Peripheral Clock Control 2 Register	0x00

Bits	Flag	Description
[7:3]	Reserved	**Reserved.**
[2]	CAPEN	**CAP Controller Clock Enable flag** This bit is set/cleared by software to enable/disable CAP controller clock. CAPEN = 1, CAP controller clock is enabled. CAPEN = 0, CAP controller clock is disabled.
[1]	MFAEN	**MFA Controller Clock Enable flag** This bit is set/cleared by software to enable/disable MFA controller clock. MFAEN = 1, MFA controller clock is enabled. MFAEN = 0, MFA controller clock is disabled.
[0]	EINTEN	**External Interrupt Sampling Source Clock Enable flag** This bit is set/cleared by software to enable/disable external interrupt controller clock. EINTEN = 1, External interrupt sampling clock is enabled. EINTEN = 0, External interrupt sampling clock is disabled.

System Clock Divider Register

Register	Offset	RW	Description	Reset Value
SYSDIV	CLK_BA+0x07	R/W	System Clock Divider Register	0x00

Bits	Flag	Description
[7:4]	PFB_N	**Peripheral Bus Clock Divider Number** This bit is set by software, cleared by software. Peripheral bus clock frequency = (HCLK) / (PFB_N + 1). PFB_N = 0, (HCLK) / 1 PFB_N = 1, (HCLK) / 2 PFB_N = 2, (HCLK) / 3 PFB_N = 3, (HCLK) / 4 ~ PFB_N = 14, (HCLK) / 15
[3:0]	MCU_N	**MCU Clock Divider Number** This bit is set by software, cleared by software. MCU clock frequency = (FSRC) / (MCU_N + 1). MCU_N = 0, (FSRC) / 1 MCU_N = 1, (FSRC) / 2 MCU_N = 2, (FSRC) / 3 MCU_N = 3, (FSRC) / 4 ~ MCU_N = 14, (FSRC) / 15

PFB Peripheral Clock Divider 0 Register

Register	Offset	RW	Description	Reset Value
PFBDIV0	CLK_BA+0x08	R/W	PFB Peripheral Clock Divider 0 Register	0x01

Bits	Flag	Description
[7:6]	TMR0_N	**Timer 0 Clock Divider Number** This bit is set by software, cleared by software. TMR0_N = 00, PCLK/4. TMR0_N = 01, PCLK/8. TMR0_N = 10, PCLK/16. TMR0_N = 11, PCLK/32.
[5:3]	Reserved	**Reserved.**
[2:0]	SDM_N	**Delta Sigma ADC Clock Divider Number** This bit is set by software, cleared by software. While setting SDMCLK = 1MHz, PCLK must at least 8MHz. While in “slow mode” or “slow wait mode”, SDM should not be enabled. SDM _N = 000, 1MHz SDM _N = 001, 500KHz SDM _N = 010, 250KHz SDM _N = 011, 125KHz SDM _N = 100, 62.5KHz SDM _N = 101, 31.25KHz Others, Reserved.

PFB Peripheral Clock Divider 1 Register

Register	Offset	RW	Description	Reset Value
PFBDIV1	CLK_BA+0x09	R/W	PFB Peripheral Clock Divider 1 Register	0x00

Bits	Flag	Description
[7:4]	Reserved	**Reserved**.
[3:2]	TMR2_N	**Timer 2 Clock Divider Number** This bit is set by software, cleared by software. TMR2_N = 00, PCLK/4. TMR2_N = 01, PCLK/8. TMR2_N = 10, PCLK/16. TMR2_N = 11, PCLK/32.
[1:0]	TMR1_N	**Timer 1 Clock Divider Number** This bit is set by software, cleared by software. TMR1_N = 00, PCLK/4. TMR1_N = 01, PCLK/8. TMR1_N = 10, PCLK/16. TMR1_N = 11, PCLK/32.

Power Mode Control Register

Register	Offset	RW	Description	Reset Value
PMCON	CLK_BA+0x0A	R/W	Power Mode Control Register	0x00

Bits	Flag	Description
[7:2]	Reserved	**Reserved**.
[1:0]	PMCON	**Power Mode Control Flag** PMCON = 00, Normal or Slow mode. PMCON = 01, Wait or Slow Wait mode. PMCON = 10, Halt mode. PMCON = 11, Shutdown mode.

Clock Trim Register

Register	Offset	RW	Description	Reset Value
CLKTRIM	CLK_BA+0x10	R/W	Clock Trim Register	0x00

Bits	Flag	Description
[7]	HSRCH	**High Speed RC Oscillator Trim High Flag**
[6:0]	HSRCL	**High Speed RC Oscillator Trim Flag**

The clock trim register should not be used by user (defined in eMTP system configuration block).

6.6 Flash Memory Controller (FMC)

6.6.1 Overview

The FE81 series integrates with an 8K-byte on-chip eMTP flash memory, which is accessed by flash memory controller (FMC). The 8K-byte eMTP flash memory has 2 blocks, main block and information block, where main block is stored with user application code, and information block is stored with system configuration and user configuration. Figure 6.6-1 shows the organization of eMTP blocks.

eMTP flash memory blocks

- Main block (4K x 16b)
  - Application code
- Information block (32 x 16b)
  - System configuration (16 x 16b)
  - User configuration (16 x 16b)

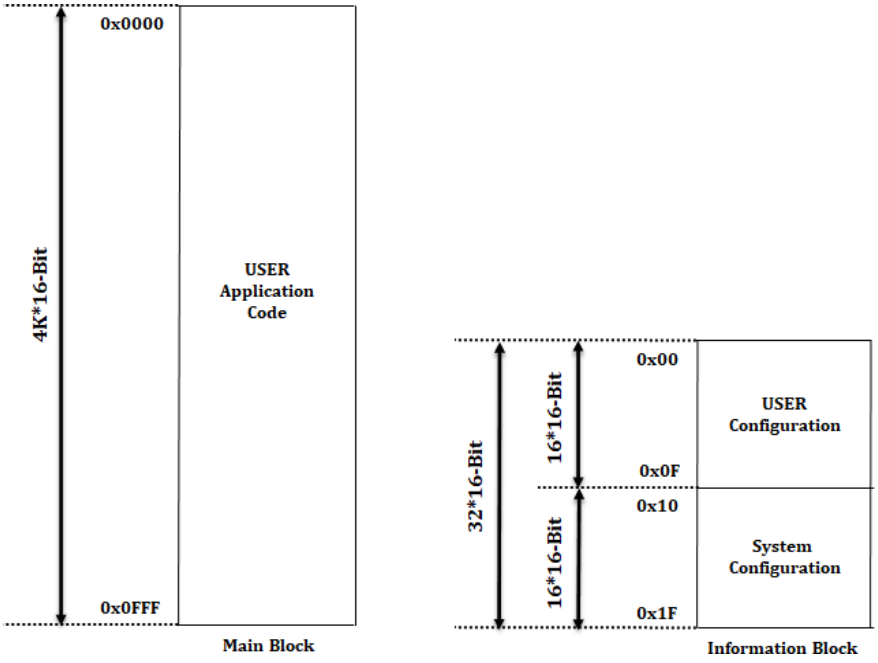


Figure 6.6-1 eMTP Block Organization

The flash memory controller has 2 slave ports (S0, and S1), one is for 8051 core reading the application code, and another is for 8051 core, DAP, and ISP accessing main block and information blocks. Table 6.6-1 describes the FMC port usages accessed by peripherals.

FMC Slave Port	Peripheral	Usage Methodology
S0	8051 core	Main block instruction fetch
S1	8051 core	Information block read/program*1
	DAP	1. Main block read/program 2. Information block read/program
	ISP	1. Main block read/program 2. Information block read/program

Table 6.6-1 FMC Port Access Methodology

*1 Note that S1 port should not be used by 8051 core to program main block.

Figure 6.6-2 depicts the diagram of access methodology by 8051 core, DAP, and ISP. The green arrow line shows the access path by 8051 core, the orange arrow line shows the access path by DAP, whereas the dark blue line shows the access path of ISP.

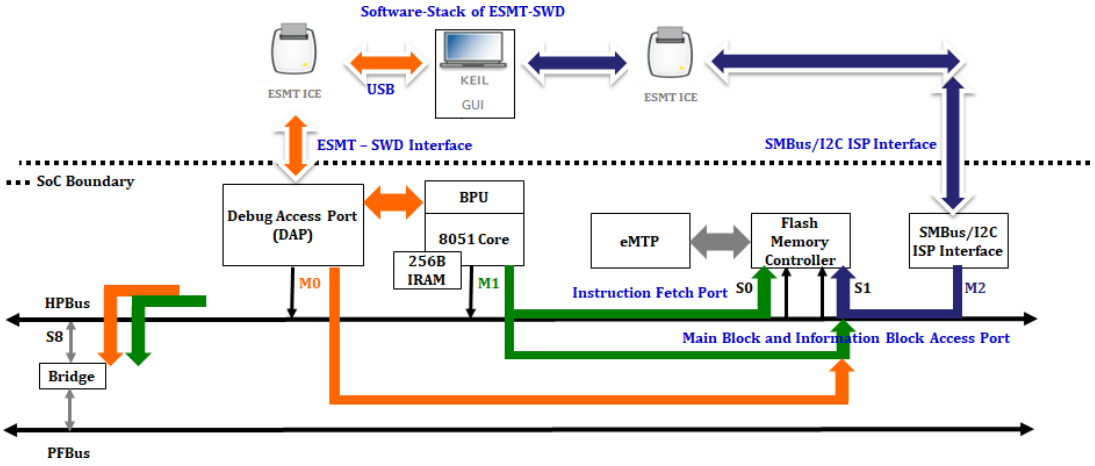


Figure 6.6-2 Block Diagram of eMTP Access

Note that while accessing eMTP via FMC S1 port, the FMC clock must be enabled first by FMCEN bit field in HBPCLKC Register.

6.6.2 System Configuration

The system configuration in information block stores parameters for customers to set initial operation environment. Table 6.6-2 illustrates the detail description of system configuration. User can use provides GUI tool or IDE tool (Keil) for system configuration programming.

Bits	Flag	Description
[31:28]	Reserved	**Reserved.**
[27]	WDTEN	**WDT Enable Flag** WDTEN = 0, WDT disabled (Default). WDTEN = 1, WDT enabled.
[26]	LVREN	**LVR Enable flag** LVREN = 0, LVR disabled (Default). LVREN = 1, LVR enabled.
[25:24]	LVRLV	**LVR Level Selection flag** LVRLV = 00, 2.7v (Default). LVRLV = 01, 2.3v. LVRLV = 01, 2.1v. LVRLV = 01, 1.9v.
[23]	LVDEN	**LVD Enable flag** LVDEN = 0, LVD disabled (Default). LVDEN = 1, LVD enabled.
[22:20]	LVDLV	**LVD Level Selection flag** LVDLV = 0, 2.9v (Default). LVDLV = 1, 2.75v. LVDLV = 2, 2.6v. LVDLV = 3, 2.45v. LVDLV = 4, 2.3v. LVDLV = 5, 2.2v. LVDLV = 6, 2.1v. LVDLV = 7, 1.9v. Others = reserved.
[19]	PB1PB2SEL	**PB1 &PB2 IO Function Selection flag** PB1PB2SEL = 0, GPIO. PB1PB2SEL = 1, XTAL input (Default).
[18]	PB0SEL	**PB0 IO Function Selection flag** PB0SEL = 0, GPIO. PB0SEL = 1, External reset (Default).
[17]	PC45SEL	**PC4 & PC5 IO Function Selection flag** PC45SEL = 0, GPIO. PC45SEL = 1, ISP (Default).
[16]	PC23SEL	**PC2 & PC3 IO Function Selection flag** PC23SEL = 0, GPIO. PC23SEL = 1, SWD (Default).
[15:8]	HSRCTRIM	High Speed RC Oscillator Trim Flag
[7:6]	Reserved	**Reserved**

[5:4]	HCLKSRC	**HCLK Clock Source Selection flag** HCLKSRC = 00, Clock source is from high speed oscillator (Default). HCLKSRC = 01, Clock source is from crystal. HCLKSRC = 10, Clock source is from low speed oscillator.
[3]	XTALSEL	**External Crystal Selection Flag** XTALSEL = 1, External 32.768Mhz is enabled. XTALSEL = 0, External 8Mhz is enabled (Default).
[2]	LSRCEN	**Internal Low Speed Oscillator Enable Flag** LSRCEN = 1, Internal low speed oscillator is enabled (Default). LSRCEN = 0, Internal low speed oscillator is disabled.
[1]	HSRCEN	**Internal High Speed Oscillator Enable Flag** HSRCEN = 1, Internal high speed oscillator is enabled (Default). HSRCEN = 0, Internal high speed oscillator is disabled.
[0]	XTLEN	**External Crystal Enable Flag** XTLEN = 1, External crystal is enabled (Default). XTLEN = 0, External crystal is disabled.

Table 6.6-2 System Configuration

6.6.3 Debug Access Port (DAP)

The FE81 series uses 2-wire ESMT-SWD interface (up to 4Mb/s) for on-chip debugging and in-circuit-programming (ICP). While using 2-wire ESMT-SWD, the FE81 series receives commands sent from host (IDE tool, for example, Keil), and these commands are decoded by debug access port (DAP) generating related system and 8051 core control signals. That is, all system peripheral registers and memory subsystem are able to be accessed by DAP.

Peripheral accessible by DAP

- Memory subsystem
  - I-RAM (256-byte)
  - X-RAM (1K-byte)
  - eMTP including main block and information block
- Peripheral
  - HPB peripherals
  - PFB peripherals

The 2-wire ESMT-SWD interface are pin-shared with GPIO pins as shown below. The Default setting of PC2, PC3 is OCD function, and users can re-set the function to GPIO by system configuration or application code.

OCD Pin	GPIO Pin	Pin Description
OCD_SWD	PC2	On-chip debugging data input/output pin
OCD_SCK	PC3	On-chip debugging clock input pin

6.6.4 In-System-Programming (ISP)

The FE81 series uses 2-wire ISP interface (up to 400Kb/s) for in system programming. With this ISP feature, IC bounded on PCBA need not to be de-soldered while having the requirement of updating application code. Users only preserve the ISP interface (ISP_SCL pin, ISP_SDA pin) on PCBA, and ESMT will provide GUI tool for customers.

Figure 6.6-3 depicts the ISP protocol, and Table 6.6-3 describes the ISP commands supported by FE81 series. Please refer application note for ISP programming while having the requirement of implementing user's own GUI tool.

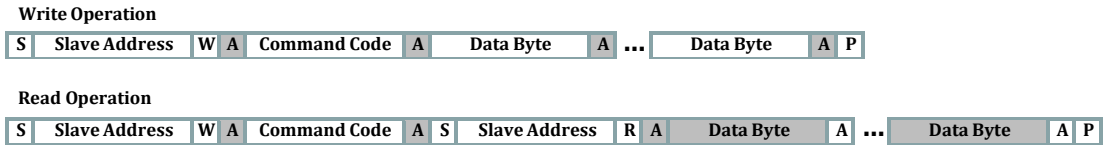


Figure 6.6-3 ISP Protocol

Command	Command Code	Description
Write Address	0x01	Write FMC address
Write Byte	0x10	Write 1 byte to FMC pointed by FMC address
Write Word	0x11	Write 2 bytes to FMC pointed by FMC address
Write Page	0x12	Write 32 bytes to FMC pointed by FMC address
Read Byte	0x20	Read 1 byte from FMC pointed by FMC address
Read Word	0x21	Read 2 bytes from FMC pointed by FMC address
Read Page	0x22	Read 32 bytes from FMC pointed by FMC address
ISPGATE	0x88	8051 core clock gated
ISPFREE	0x89	8051 core clock released
ISPRST	0x99	ISP reset enabled

Table 6.6-3 ISP Command

The 2-wire ISP interface pins are pin-shared with GPIO pins as shown below. Default setting of PC4, PC5 is ISP function, and users can re-set the function to GPIO by system configuration or program code.

ISP Pin	GPIO Pin	Pin Description
ISP_SDA	PC4	ISP data input/output pin
ISP_SCL	PC5	ISP clock input pin

6.6.5 Register Map and Description

Base Address (FMC_BA) : 0xD000				
Register	**Offset**	**RW**	**Description**	**Reset Value**
FMCCON	FMC_BA+0x00	R/W	Flash Memory Controller Control Register	0x20
FMCCMD	FMC_BA+0x01	R/W	Flash Memory Controller Command Register	0x00
FMCADR0	FMC_BA+0x02	R/W	Flash Memory Controller Address 0 Register	0x00
FMCADR1	FMC_BA+0x03	R/W	Flash Memory Controller Address 1 Register	0x00
FMCDATA0	FMC_BA+0x04	R/W	Flash Memory Controller Data 0 Register	0x00
FMCDATA1	FMC_BA+0x05	R/W	Flash Memory Controller Data 1 Register	0x00
FMCSTAT	FMC_BA+0x06	R	Flash Memory Controller Status Register	0x00
FMCCKSUM0	FMC_BA+0x07	R	Flash Memory Controller Checksum 0 Register	0x00
FMCCKSUM1	FMC_BA+0x08	R	Flash Memory Controller Checksum 1 Register	0x00

Flash Memory Controller Register

Register	Offset	RW	Description	Reset Value
FMCCON	FMC_BA+0x00	R/W	Flash Memory Controller Control Register	0x20

Bits	Flag	Description
[7]	FMCEN	**FMC Function Enable Flag** This bit is set by software, cleared by software, or set by DAP, ISP. FMCEN = 1, FMC function is enabled. FMCEN = 0, FMC function is disabled. Note : MCU writes eMTP information block do not set this flag.
[6]	INFOEN	**FMC Information Block Enable Flag** This bit is set by software, cleared by software, or set by DAP, ISP. INFOEN = 1, eMTP information block is selected. INFOEN = 0, eMTP main block is selected.
[5]	ISAVB	**eMTP Operation Speed Enable Flag** This bit is set by software, cleared by software, or set by DAP, ISP. ISAVB = 1, eMTP high speed mode is selected. ISAVB = 0, eMTP low power mode is selected.
[4:1]	Reserved	**Reserved**.
[0]	WDTOF	**WDT Disable Flag** This bit is set by software, cleared by software, or set by DAP, ISP. While updating eMTP, watchdog timer must be disabled by this flag. WDTOF = 0, WDT configuration is not changed. WDTOF = 1, WDT is disabled.

Flash Memory Controller Command Register

Register	Offset	RW	Description	Reset Value
FMCCMD	FMC_BA+0x01	R/W	Flash Memory Controller Command Register	0x00

Bits	Flag	Description
[7:0]	CMD	**Flash Memory Command Register** FMC supports “Word Program” function. FMC supports “Page Program” function. FMC supports “Word Read” function. FMC supports “Page Read” function. MCU support “Word Program” function. MCU support “Word Read” function. CMD = 0x02, Read flash memory word data (16-bit). **(Note: MCU read Information Block must set the command)** CMD = 0x08, Program flash memory word data (16-bit). CMD = 0x09, Program flash memory page data (32*16-bit). CMD = 0x91, Flash memory standby. CMD = 0x92, Flash memory static. CMD = 0x93, Flash bit cell current testing CMD = 0x99, Flash memory enable (default enabled).

Flash Memory Controller Address 0 Register

Register	Offset	RW	Description	Reset Value
FMCADR0	FMC_BA+0x02	R/W	Flash Memory Controller Address 0 Register	0x00

Bits	Flag	Description
[7:0]	FMADR0	**Flash Memory Address Low Byte** This register is set the address of eMTP flash memory.

Flash Memory Controller Address 1 Register

Register	Offset	RW	Description	Reset Value
FMCADR1	FMC_BA+0x03	R/W	Flash Memory Controller Address 1 Register	0x00

Bits	Flag	Description
[7:0]	FMADR1	**Flash Memory Address High Byte** This register is set the address of eMTP flash memory.

Flash Memory Controller Data 0 Register

Register	Offset	RW	Description	Reset Value
FMCDATA0	FMC_BA+0x04	R/W	Flash Memory Controller Data 0 Register	0x00

Bits	Flag	Description
[7:0]	FM**DAT0**	**Flash Memory Data Low Byte** This register is set the data of eMTP flash memory. Either flash memory read or program, the data are all located in this register.

Flash Memory Controller Data 1 Register

Register	Offset	RW	Description	Reset Value
FMCDATA1	FMC_BA+0x05	R/W	Flash Memory Controller Data 1 Register	0x00

Bits	Flag	Description
[7:0]	FM**DAT1**	**Flash Memory Data High Byte** This register is set the data of eMTP flash memory. Either flash memory read or program, the data are all located in this register.

Flash Memory Controller Status Register

Register	Offset	RW	Description	Reset Value
FMCSTAT	FMC_BA+0x06	R	Flash Memory Controller Status Register	0x00

Bits	Flag	Description
[7]	BUSY	**FMC Busy Flag** This bit is used for the FMC program ready check. This bit is set by hardware, and cleared by hardware. BUSY = 1, FMC program is not finished. BUSY = 0, FMC idle.
[6]	ECC	**FMC Error Correction Check Flag** This bit is set by hardware, and cleared by hardware. ECC = 1, FMC read ECC is set. ECC = 0, FMC read without ECC is executed.
[5:0]	Reserved	**Reserved.**

Flash Memory Controller Checksum 0 Register

Register	Offset	RW	Description	Reset Value
FMCCKSUM0	FMC_BA+0x07	R	Flash Memory Controller Checksum 0 Register	0x00

Bits	Flag	Description
[7:0]	CKSUM0	**Code Checksum 0 Register**

Note : CKSUM0 = CKSUM0 + DATA

Flash Memory Controller Checksum 1 Register

Register	Offset	RW	Description	Reset Value
FMCCKSUM1	FMC_BA+0x08	R	Flash Memory Controller Checksum 1 Register	0x00

Bits	Flag	Description
[7:0]	CKSUM1	**Code Checksum 1 Register**

Note : CKSUM1 = CKSUM1 ^ DATA

6.7 Delta-Sigma Controller (SDM)

6.7.1 Overview

The FE81 series contains a high accuracy multi-channel 24-bit Delta Sigma analog-to-digital converter which can directly interface to external analog signals, such as that from sensors or other control signals and convert these signals directly into a 24-bit digital value.

The following block diagram illustrates the SDM converter basic operational function. The SDM converter input channel can be arranged as seven differential input channels. The SDM converter module will output one bit converted data to FIR filter which can transform the converted one-bit data to 24 bits and store them into the specific data registers. With high accuracy and performance, the device is very suitable for differential output sensor applications such as temperature sensor and other related products.

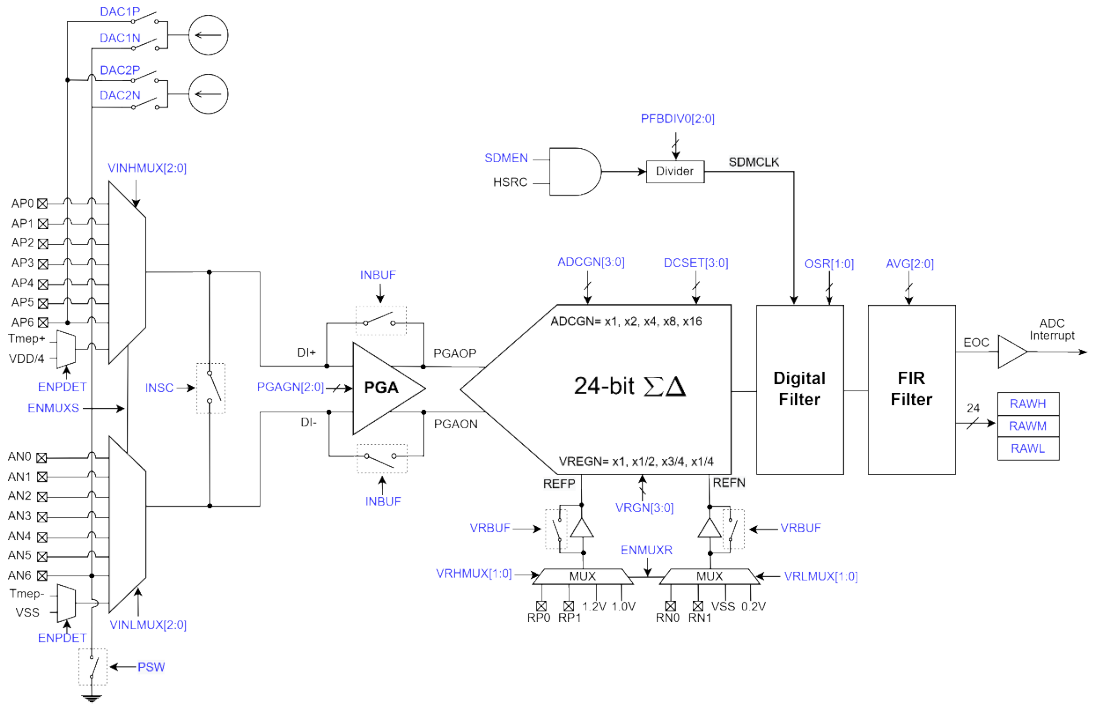


Figure 6.7-1 Block Diagram of SDM

6.7.2 Register Map and Description

Base Address (SDM_BA) : 0xE600				
Register	**Offset**	**RW**	**Description**	**Reset Value**
RAWL	SDM_BA+0x00	R	SDM Raw Data Low Register	0x00
RAWM	SDM_BA+0x01	R	SDM Raw Data Medium Register	0x00
RAWH	SDM_BA+0x02	R	SDM Raw Data High Register	0x00
CLBL	SDM_BA+0x03	R	SDM Calibration Data Low Register	0x00
CLBM	SDM_BA+0x04	R	SDM Calibration Data Medium Register	0x00
CLBH	SDM_BA+0x05	R	SDM Calibration Data High Register	0x00
SDMGAINL	SDM_BA+0x06	R/W	SDM Sensor Calibration Gain Low Register	0x00
SDMGAINM	SDM_BA+0x07	R/W	SDM Sensor Calibration Gain Medium Register	0x00
SDMGAINH	SDM_BA+0x08	R/W	SDM Sensor Calibration Gain High Register	0x00
SDMOSL	SDM_BA+0x09	R/W	SDM Sensor Calibration OS Low Register	0x00
SDMOSM	SDM_BA+0x0A	R/W	SDM Sensor Calibration OS Medium Register	0x00
SDMOSH	SDM_BA+0x0B	R/W	SDM Sensor Calibration OS High Register	0x00
SDMOFL	SDM_BA+0x0C	R/W	SDM Sensor Calibration OF Low Register	0x00
SDMOFM	SDM_BA+0x0D	R/W	SDM Sensor Calibration OF Medium Register	0x00
SDMOFH	SDM_BA+0x0E	R/W	SDM Sensor Calibration OF High Register	0x00
TEMPGAINL	SDM_BA+0x0F	R/W	SDM Temperature Calibration Gain Low Register	0x00
TEMPGAINM	SDM_BA+0x10	R/W	SDM Temperature Calibration Gain Medium Register	0x00
TEMPGAINH	SDM_BA+0x11	R/W	SDM Temperature Calibration Gain High Register	0x00
TEMPOSL	SDM_BA+0x12	R/W	SDM Temperature Calibration OS Low Register	0x00
TEMPOSM	SDM_BA+0x13	R/W	SDM Temperature Calibration OS Medium Register	0x00
TEMPOSH	SDM_BA+0x14	R/W	SDM Temperature Calibration OS High Register	0x00
TEMPOFL	SDM_BA+0x15	R/W	SDM Temperature Calibration OF Low Register	0x00
TEMPOFM	SDM_BA+0x16	R/W	SDM Temperature Calibration OF Medium Register	0x00
TEMPOFH	SDM_BA+0x17	R/W	SDM Temperature Calibration OF High Register	0x00
OSR	SDM_BA+0x18	R/W	OSR Register	0x00
INMUX	SDM_BA+0x19	R/W	Input Mux Register	0x00
RMUX	SDM_BA+0x1A	R/W	Reference Mux Register	0x00
PGASET	SDM_BA+0x1B	R/W	Enable Configuration Register	0x00
ENCFG	SDM_BA+0x1C	R/W	ADC Setting Register	0x00
ADCSET	SDM_BA+0x1D	R/W	PGA Setting Register	0x00
CONFIG	SDM_BA+0x1E	R/W	Configuration Register	0x00
INTSTAT	SDM_BA+0x1F	R/W	Interrupt Status Register	0x00
TEST	SDM_BA+0x80	R/W	Test Register	0x00

SDM Raw Data Register

Register	Offset	RW	Description	Reset Value
RAWL	SDM_BA+0x00	R	SDM Raw Data Low Register	0x00

Register	Offset	RW	Description	Reset Value
RAWM	SDM_BA+0x01	R	SDM Raw Data Medium Register	0x00

Register	Offset	RW	Description	Reset Value
RAWH	SDM_BA+0x02	R	SDM Raw Data High Register	0x00

Bits	Flag	Description
[7:0]	RAWL	**Sigma Delta Raw Data Low Byte Register**

Bits	Flag	Description
[7:0]	RAWM	**Sigma Delta Raw Data Medium Byte Register**

Bits	Flag	Description
[7:0]	RAWH	**Sigma Delta Raw Data High Byte Register**

SDM Calibration Data Register

Register	Offset	RW	Description	Reset Value
CLBL	SDM_BA+0x03	R	SDM Calibration Data Low Register	0x00

Register	Offset	RW	Description	Reset Value
CLBM	SDM_BA+0x04	R	SDM Calibration Data Medium Register	0x00

Register	Offset	RW	Description	Reset Value
CLBH	SDM_BA+0x05	R	SDM Calibration Data High Register	0x00

Bits	Flag	Description
[7:0]	CLBL	**Sigma Delta Calibrated Data Low Byte Register**

Bits	Flag	Description
[7:0]	CLBM	**Sigma Delta Calibrated Data Medium Byte Register**

Bits	Flag	Description
[7:0]	CLBH	**Sigma Delta Calibrated Data High Byte Register**

SDM Sensor Calibration Gain Register

Register	Offset	RW	Description	Reset Value
SDMGAINL	SDM_BA+0x06	R/W	SDM Sensor Calibration Gain Low Register	0x00

Register	Offset	RW	Description	Reset Value
SDMGAINM	SDM_BA+0x07	R/W	SDM Sensor Calibration Gain Medium Register	0x00

Register	Offset	RW	Description	Reset Value
SDMGAINH	SDM_BA+0x08	R/W	SDM Sensor Calibration Gain High Register	0x00

Bits	Flag	Description
[7:0]	CLBL	**Sigma Delta Sensor Calibration Gain Low Byte Register**

Bits	Flag	Description
[7:0]	CLBM	**Sigma Delta Sensor Calibration Gain Medium Byte Register**

Bits	Flag	Description
[7:0]	CLBH	**Sigma Delta Sensor Calibration Gain High Byte Register**

SDM Sensor Calibration OS Register

Register	Offset	RW	Description	Reset Value
SDMOSL	SDM_BA+0x09	R/W	SDM Sensor Calibration OS Low Register	0x00

Register	Offset	RW	Description	Reset Value
SDMOSM	SDM_BA+0x0A	R/W	SDM Sensor Calibration OS Medium Register	0x00

Register	Offset	RW	Description	Reset Value
SDMOSH	SDM_BA+0x0B	R/W	SDM Sensor Calibration OS High Register	0x00

Bits	Flag	Description
[7:0]	SDMOSL	**Sigma Delta Sensor Calibration OS Low Byte Register**

Bits	Flag	Description
[7:0]	SDMOSM	**Sigma Delta Sensor Calibration OS Medium Byte Register**

Bits	Flag	Description
[7:0]	SDMOSH	**Sigma Delta Sensor Calibration OS High Byte Register**

SDM Sensor Calibration OF Register

Register	Offset	RW	Description	Reset Value
SDMOFL	SDM_BA+0x0C	R/W	SDM Sensor Calibration OF Low Register	0x00

Register	Offset	RW	Description	Reset Value
SDMOFM	SDM_BA+0x0D	R/W	SDM Sensor Calibration OF Medium Register	0x00

Register	Offset	RW	Description	Reset Value
SDMOFH	SDM_BA+0x0E	R/W	SDM Sensor Calibration OF High Register	0x00

Bits	Flag	Description
[7:0]	SDMOFL	**Sigma Delta Sensor Calibration OF Low Byte Register**

Bits	Flag	Description
[7:0]	SDMOFM	**Sigma Delta Sensor Calibration OF Medium Byte Register**

Bits	Flag	Description
[7:0]	SDMOFH	**Sigma Delta Sensor Calibration OF High Byte Register**

SDM Temperature Calibration Gain Register

Register	Offset	RW	Description	Reset Value
TEMPGAINL	SDM_BA+0x0F	R/W	SDM Temperature Calibration Gain Low Register	0x00

Register	Offset	RW	Description	Reset Value
TEMPGAINM	SDM_BA+0x10	R/W	SDM Temperature Calibration Gain Medium Register	0x00

Register	Offset	RW	Description	Reset Value
TEMPGAINH	SDM_BA+0x11	R/W	SDM Temperature Calibration Gain High Register	0x00

Bits	Flag	Description
[7:0]	TEMPGAINL	**Sigma Delta Temperature Calibration Gain Low Byte Register**

Bits	Flag	Description
[7:0]	TEMPGAINM	**Sigma Delta Temperature Calibration Gain Medium Byte Register**

Bits	Flag	Description
[7:0]	TEMPGAINH	**Sigma Delta Temperature Calibration Gain High Byte Register**

SDM Temperature Calibration OS Register

Register	Offset	RW	Description	Reset Value
TEMPOSL	SDM_BA+0x12	R/W	SDM Temperature Calibration OS Low Register	0x00

Register	Offset	RW	Description	Reset Value
TEMPOSM	SDM_BA+0x13	R/W	SDM Temperature Calibration OS Medium Register	0x00

Register	Offset	RW	Description	Reset Value
TEMPOSH	SDM_BA+0x14	R/W	SDM Temperature Calibration OS High Register	0x00

Bits	Flag	Description
[7:0]	TEMPOSL	**Sigma Delta Temperature Calibration OS Low Byte Register**

Bits	Flag	Description
[7:0]	TEMPOSM	**Sigma Delta Temperature Calibration OS Medium Byte Register**

Bits	Flag	Description
[7:0]	TEMPOSH	**Sigma Delta Temperature Calibration OS High Byte Register**

SDM Temperature Calibration OF Register

Register	Offset	RW	Description	Reset Value
TEMPOFL	SDM_BA+0x15	R/W	SDM Temperature Calibration OF Low Register	0x00

Register	Offset	RW	Description	Reset Value
TEMPOFM	SDM_BA+0x16	R/W	SDM Temperature Calibration OF Medium Register	0x00

Register	Offset	RW	Description	Reset Value
TEMPOFH	SDM_BA+0x17	R/W	SDM Temperature Calibration OF High Register	0x00

Bits	Flag	Description
[7:0]	TEMPOFL	**Sigma Delta Temperature OF Low Byte Register**

Bits	Flag	Description
[7:0]	TEMPOFM	**Sigma Delta Temperature OF Medium Byte Register**

Bits	Flag	Description
[7:0]	TEMPOFH	**Sigma Delta Temperature OF High Byte Register**

SDM Over Sampling Register

Register	Offset	RW	Description	Reset Value
OSR	SDM_BA+0x18	R/W	OSR Register	0x00

Bits	Flag	Description
[7:4]	Reserved	**Reserved.**
[3:0]	OSR	**Sigma Delta Over Sampling Rate Selection Flag** This bit is set by software, cleared by software. OSR = 0, Over sampling rate is 32768. OSR = 1, Over sampling rate is 16384. OSR = 2, Over sampling rate is 8192. OSR = 3, Over sampling rate is 4096. OSR = 4, Over sampling rate is 2048. OSR = 5, Over sampling rate is 1024. OSR = 6, Over sampling rate is 512. OSR = 7, Over sampling rate is 256. OSR = 8, Over sampling rate is 128. Others, Reserved.

SDM Input Mux Selection Register

Register	Offset	RW	Description	Reset Value
INMUX	SDM_BA+0x19	R/W	Input Mux Selection Register	0x00

Bits	Flag	Description
[7]	ENMUXS	**Sigma Delta Input Mux Function Enable Flag** This bit is set by software, cleared by software. ENMUXS = 0, Input mux selection function disabled. ENMUXS = 1, Input mux selection function enabled.
[6]	Reserved	**Reserved.**
[5:3]	VINLMUX	**Sigma Delta Negative Input Mux Selection Flag** This bit is set by software, cleared by software. VINLMUX = 0, AN0. VINLMUX = 1, AN1. VINLMUX = 2, AN2. VINLMUX = 3, AN3. VINLMUX = 4, AN4. VINLMUX = 5, AN5. VINLMUX = 6, RN1. VINLMUX = 7, RN0.
[2:0]	VINHMUX	**Sigma Delta Positive Input Mux Selection Flag** This bit is set by software, cleared by software. VINHMUX = 0, AP0. VINHMUX = 1, AP1. VINHMUX = 2, AP2. VINHMUX = 3, AP3. VINHMUX = 4, AP4. VINHMUX = 5, AP5. VINHMUX = 6, RP1. VINHMUX = 7, RP0.

SDM Voltage Reference Mux Selection Register

Register	Offset	RW	Description	Reset Value
RMUX	SDM_BA+0x1A	R/W	Voltage Reference Mux Selection Register	0x00

Bits	Flag	Description
[7]	ENMUXR	**Sigma Delta Voltage Reference Mux Enable Flag** This bit is set by software, cleared by software. ENMUXR = 0, Reference mux selection function disabled. ENMUXR = 1, Reference mux selection function enabled.
[6]	Reserved	**Reserved.**
[5:4]	Reserved	**Reserved**
[3:2]	VRLMUX	**Sigma Delta Negative Voltage Reference Mux Selection Flag** This bit is set by software, cleared by software. VRLMUX = 0, Voltage Reference is RP0. VRLMUX = 1, Voltage Reference is RP1. VRLMUX = 2, Voltage Reference is 1.2V. VRLMUX = 3, Voltage Reference is 1.0V.
[1:0]	VRHMUX	**Sigma Delta Positive Voltage Reference Mux Selection Flag** This bit is set by software, cleared by software. VRHMUX = 0, Voltage Reference is RP0. VRHMUX = 1, Voltage Reference is RP1. VRHMUX = 2, Voltage Reference is 1.2V. VRHMUX = 3, Voltage Reference is 1.0V.

SDM PGA Setting Register

Register	Offset	RW	Description	Reset Value
PGASET	SDM_BA+0x1B	R/W	PGA Setting Register	0x00

Bits	Flag	Description
[7]	ENBUFCH	**Sigma Delta Buffer Chopper Enable Flag** This bit is set by software, cleared by software. ENBUFCH = 0, Buffer chopper disabled. ENBUFCH = 1, Buffer chopper enabled.
[6]	VRBUF	**Sigma Delta Voltage Reference Buffer Enable Flag** This bit is set by software, cleared by software. VRBUF = 0, Voltage reference buffer disabled. VRBUF = 1, Voltage reference buffer enabled.
[5]	ENPGACH	**Sigma Delta PGA Chopper Enable Flag** This bit is set by software, cleared by software. ENPGACH = 0, PGA chopper disabled. ENPGACH = 1, PGA chopper enabled.
[4]	INBUF	**Sigma Delta Input Buffer Enable Flag** This bit is set by software, cleared by software. INBUF = 0, Input buffer disabled. INBUF = 1, Input buffer enabled.
[3]	Reserved	**Reserved.**
[2:0]	PGAGN	**Sigma Delta PGA Gain Selection Flag** This bit is set by software, cleared by software. PGAGN = 0, PGA Gain is 1x. PGAGN = 1, PGA Gain is 2x. PGAGN = 2, PGA Gain is 4x. PGAGN = 3, PGA Gain is 8x. PGAGN = 4, PGA Gain is 16x. PGAGN = 5, PGA Gain is 32x. PGAGN = 6, PGA Gain is 64x. PGAGN = 7, PGA Gain is 128x.

SDM Enable Configuration Register

Register	Offset	RW	Description	Reset Value
ENCFG	SDM_BA+0x1C	R/W	Enable Configuration Register	0x00

Bits	Flag	Description
[7]	ENADC	**Sigma Delta ADC Enable Flag** This bit is set by software, cleared by software. ENADC = 0, SDM disabled. ENADC = 1, SDM enabled.
[6]	ENADCCH	**Sigma Delta ADC Chopper Enable Flag** This bit is set by software, cleared by software. ENADCCH = 0, SDM chopper disabled. ENADCCH = 1, SDM chopper enabled.
[5:4]	VRGN	**Sigma Delta Voltage Reference Gain Selection Flag** This bit is set by software, cleared by software. VRGN = 0, Voltage Reference Gain is 1x. VRGN = 1, Voltage Reference Gain is 1/2x. VRGN = 2, Voltage Reference Gain is 3/4x. VRGN = 3, Voltage Reference Gain is 1/4x.
[3:0]	ADCGN	**Sigma Delta ADC Gain Selection Flag** This bit is set by software, cleared by software. ADCGN = 0, ADC Gain is 1x. ADCGN = 1, ADC Gain is 2x. ADCGN = 3, ADC Gain is 4x. ADCGN = 7, ADC Gain is 8x. ADCGN = 15, ADC Gain is 16x.

SDM ADC Setting Register

Register	Offset	RW	Description	Reset Value
ADCSET	SDM_BA+0x1D	R/W	ADC Setting Register	0x00

Bits	Flag	Description
[7:3]	Reserved	**Reserved.**
[2:0]	DCSET	**Sigma Delta DC Offset Selection Flag** This bit is set by software, cleared by software. DCSET = 0, DC Offset is 0 x Vref. DCSET = 1, DC Offset is (1/16) x Vref. DCSET = 2, DC Offset is (2/16) x Vref. DCSET = 3, DC Offset is (3/16) x Vref. DCSET = 4, DC Offset is (4/16) x Vref. DCSET = 5, DC Offset is (5/16) x Vref. DCSET = 6, DC Offset is (6/16) x Vref. DCSET = 7, DC Offset is (7/16) x Vref.

SDM Configuration Register

Register	Offset	RW	Description	Reset Value
CONFIG	SDM_BA+0x1E	R/W	Configuration Register	0x00

Bits	Flag	Description
[7:3]	Reserved	**Reserved.**
[2]	ENDEM	**Sigma Delta DEM clock Enable Flag** This bit is set by software, cleared by software. ENDEM = 0, DEM clock disabled. ENDEM = 1, DEM clock enabled.
[1]	ENSNSCH	**Sigma Delta Temperature Chopper Enable Flag** This bit is set by software, cleared by software. ENSNSCH = 0, Temperature chopper disabled. ENSNSCH = 1, Temperature chopper enabled.
[0]	ENTMPSNS	**Sigma Delta Temperature Function Enable Register** This bit is set by software, cleared by software. ENTEMP = 0, Temperature function disabled. ENTEMP = 1, Temperature function enabled. **Note: This bit must be set when ADC measurement**

SDM Interrupt Status Register

Register	Offset	RW	Description	Reset Value
INTSTAT	SDM_BA+0x1F	R/W	Interrupt Status Register	0x00

Bits	Flag	Description
[7]	INTF	**Sigma Delta SDM Interrupt Flag** This bit is set by software, cleared by software. INTF = 0, SDM interrupt not happened. INTF = 1, SDM interrupt happened.
[6:1]	Reserved	**Reserved.**
[0]	INTEN	**Sigma Delta Interrupt Enable Flag** This bit is set by software, cleared by software. INTEN = 0, SDM interrupt disabled. INTEN = 1, SDM interrupt enabled.

SDM Test Register

Register	Offset	RW	Description	Reset Value
TEST	SDM_BA+0x80	R/W	Test Register	0x00

Bits	Flag	Description
[7:4]	Reserved	**Reserved.**
[3]	CALDVDC	**Sigma Delta Power Mux Function Flag** This bit is set by software, cleared by software. CALDVDC = 0, Power mux function disabled. CALDVDC = 1, Power mux function enabled.
[2:1]	INCH	**Input signal architecture configuration** 00：INH connects to SDM+ and INL connects to SDM- 01：INL connects to SDM+/SDM- 10：INH connects to SDM+/SDM- 11：INH connects to SDM- and INL connects to SDM+
[0]	INSC	**Sigma Delta Input Short Circuit Enable Flag** This bit is set by software, cleared by software. INSC = 0, SDM input short disabled. INSC = 1, SDM input short enabled.

6.8 Analog Function (ANA)

6.8.1 Overview

The FE81 series contains OPAMP and related function such that hysteresis comparator and rail to rail c omparator which user can use via control IO pin (PD1, PD2, PD3, PC6, PC7) and register (COMPCFG, C OMPRES) to implement related application. This device also has a Low Voltage Detector function, also known as LVD. This enabled the devices to monitor the power supply voltage, VDD, and provide a war ning signal if it falls below a certain level.

The FE81 series provides two current generation used for SDM measurement in analog input Channel 6, each current generation can be set 10uA~1200uA in register IGEN1, IGEN2, it is very useful for some application, like resistive sensor (RTD).

OPAMP and Comparator

The following block diagram illustrates the OPAMP and Comparator basic operational function.

The OPAMP operate configuration as follows:

- Enable OPAMP flag in register PWCFG.
- Set PD2 as OPAP, PD1 as APAN, PD3 as OPAO in register PDCTRL0, PDCTRL1.
- Set PD2(OPAMP+), PD1(OPAMP-), PD3(OPOUT) as high-impendence in register P3MS0~P3MS2.

The Comparator(R2R) operate configuration as follows:

- Enable CMPREN flag in register COMPCFG.
- Set IVTH flag as 1 in register COMPCFG (Select internal voltage level mux).
- Select voltage level in CMPTH.
- Hardware setting: PC6 (CMP+), PC7 (CMP-).
- The compare result showed in CMPOUT flag in register COMPRES.

The Comparator(HYS) operate configuration as follows:

- Enable CMPHEN flag in register COMPCFG.
- Set IVTH flag as 1 in register COMPCFG (Select internal voltage level mux).
- Select voltage level in CMPHYS.
- Hardware setting: PC6 (CMP+), PC7 (CMP-).
- The compare result showed in CMPOUT flag in register COMPRES.

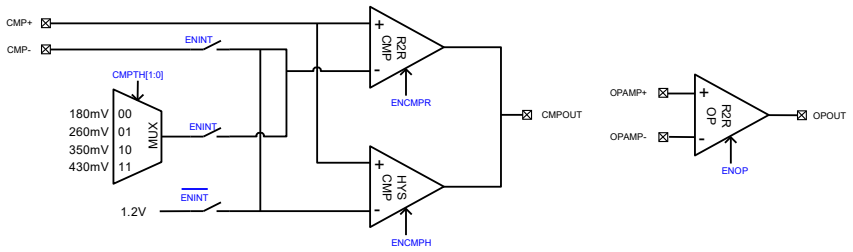


Figure 6.8-1 Block Diagram of OPAMP and Comparator

6.8.2 Register Map and Description

Base Address (ANA_BA) : 0xE800				
Register	**Offset**	**RW**	**Description**	**Reset Value**
COMPCFG	ANA_BA+0x00	R/W	Analog Comparator Configuration Register	0x00
COMPRES	ANA_BA+0x01	R	Analog Comparator Result Register	0x00
LVCFG	ANA_BA+0x02	R/W	Analog Low Voltage Configuration Register	0x00
LVDIEN	ANA_BA+0x03	R/W	Analog Low Voltage Interrupt Enable Register	0x00
PWCFG	ANA_BA+0x04	R/W	Analog Power Configuration Register	0x02
IGENCTRL	ANA_BA+0x05	R/W	Analog Current Generation Control Register	0x00
IGEN1	ANA_BA+0x06	R/W	Analog Current Generation1 Enable Register	0x00
IGEN2	ANA_BA+0x07	R/W	Analog Current Generation2 Enable Register	0x00
IGENTRIM1	ANA_BA+0x08	R/W	Analog Current Generation Trim 1 Register	0x00
IGENTRIM2	ANA_BA+0x09	R/W	Analog Current Generation Trim 2 Register	0x00
BGTRIM	ANA_BA+0x0A	R/W	Bandgap Trim Register	0x00

Analog Comparator Configuration Register

Register	Offset	RW	Description	Reset Value
COMPCFG	ANA_BA+0x00	R/W	Analog Comparator Configuration Register	0x00

Bits	Flag	Description
[7:5]	CMPHYS	**Hysteresis Comparator Voltage Threshold Selection Flag** This bit is set by software, cleared by software. 000：Minimum hysteresis level 111： Maximum hysteresis level
[4:3]	CMPTH	**Rail to Rail Internal Negative Voltage Input Selection Flag** This bit is set by software, cleared by software. CMPTH = 0, Internal 180mV. CMPTH = 1, Internal 260mV. CMPTH = 2, Internal 350mV. CMPTH = 3, internal 430mV.
[2]	CMPIVTH	**Comparator Negative Input Voltage Selection Flag** This bit is set by software, cleared by software. CMPIVTH = 0, External negative voltage input. CMPIVTH = 1, Internal negative voltage input.
[1]	CMPHEN	**Hysteresis Comparator Enable Flag** This bit is set by software, cleared by software. CMPHEN = 0, Hysteresis comparator disabled. CMPHEN = 1, Hysteresis comparator enabled.
[0]	CMPREN	**Rail to Rail Comparator Enable Flag** This bit is set by software, cleared by software. CMPREN = 0, Rail to rail comparator disabled. CMPREN = 1, Rail to rail comparator enabled.

Analog Comparator Result Register

Register	Offset	RW	Description	Reset Value
COMPRES	ANA_BA+0x01	R	Analog Comparator Result Register	0x00

Bits	Flag	Description
[7:1]	Reserved	**Reserved.**
[0]	CMPOUT	**Comparator Result Value** This bit is set by hardware, cleared by hardware. CMPOUT = 0, Comparator input positive voltage < input negative voltage. CMPOUT = 1, Comparator input positive voltage >= input negative voltage.

Analog Low Voltage Configuration Register

Register	Offset	RW	Description	Reset Value
LVCFG	ANA_BA+0x02	R/W	Analog Low Voltage Configuration Register	0x00

Bits	Flag	Description
[7]	LVDF	**Low Voltage Detect Flag** This bit is set by hardware, cleared by hardware. LVDF = 0, Idle. LVDF = 1, Low voltage detected.
[6:4]	LVDLV	**Low Voltage Detect Voltage Selection Flag** This bit is set by software, cleared by software. LVDLV = 0, 3.0V (Default). LVDLV = 1, 2.8v. LVDLV = 2, 2.6v. LVDLV = 3, 2.4v. LVDLV = 4, 2.2v. LVDLV = 5, 2.0v. LVDLV = 6, 1.8v. LVDLV = 7, 1.6v.
[3:2]	LVRLV	**Low Voltage Reset Voltage Selection Flag** This bit is set by software, cleared by software. LVRLV = 0, 2.6v. LVRLV = 1, 2.4v. LVRLV = 2, 2.2v. LVRLV = 3, 2.0v.
[1]	LVDEN	**Low Voltage Detect Enable Flag** This bit is set by software, cleared by software. LVDEN = 0, Low voltage detect disabled. LVDEN = 1, Low voltage detect enabled.
[0]	LVREN	**Low Voltage Reset Enable Flag** This bit is set by software, cleared by software. LVREN = 0, Low voltage reset disabled. LVREN = 1, Low voltage reset enabled.

Analog Low Voltage Interrupt Enable Register

Register	Offset	RW	Description	Reset Value
LVDIEN	ANA_BA+0x03	R	Analog Low Voltage Interrupt Enable Register	0x00

Bits	Flag	Description
[7:1]	Reserved	**Reserved.**
[0]	LVDIEN	**Low Voltage Interrupt Enable Flag** This bit is set by software, cleared by software. LVDIEN = 0, Interrupt disabled. LVDIEN = 1, Interrupt enabled.

Analog Power Configuration Register

Register	Offset	RW	Description	Reset Value
PWCFG	ANA_BA+0x04	R/W	Analog Power Configuration Register	0x02

Bits	Flag	Description
[7]	Reserved	**Reserved.**
[6]	ENOPAMP	**OPAmp Enable Flag** This bit is set by software, cleared by software. ENOPAMP = 0, OPAmp disabled. ENOPAMP = 1, OPAmp enabled.
[5]	ENBODN	**BOD Negative Voltage Input Enable Flag** This bit is set by software, cleared by software. ENBODN = 0, BOD negative voltage input disabled. ENBODN = 1, BOD negative voltage input enabled.
[4]	ENBODP	**BOD Positive Voltage Input Enable Flag** This bit is set by software, cleared by software. ENBODP = 0, BOD positive voltage input disabled. ENBODP = 1, BOD positive voltage input enabled.
[3]	ENPDET	**PDET Enable Flag** This bit is set by software, cleared by software. ENPDET = 0, PDET disabled. ENVPET = 1, PDET enabled.
[0]	ENVCM	**VCM Input Enable Flag** This bit is set by software, cleared by software. ENVCM = 0, External VCM input disabled. ENVCM = 1, External VCM input enabled.
[2]	ENLDOA	**Analog LDO Enable Flag** This bit is set by software, cleared by software. ENLDOA = 0, LDOA disabled. ENLDOA = 1, LDOA enabled.
[1]	ENBIAS	**Voltage Bias Enable Flag** This bit is set by software, cleared by software. ENBIAS = 0, Voltage bias disabled. ENBIAS = 1, Voltage bias enabled.

Analog Current Generation Enable Register

Register	Offset	RW	Description	Reset Value
IGENCTRL	ANA_BA+0x05	R/W	Analog Current Generation Control Register	0x00

Bits	Flag	Description
[7:5]	Reserved	**Reserved.**
[4]	PSW	**PSW Enable Flag** This bit is set by software, cleared by software. PSW = 0, PSW disabled. PSW = 1, PSW enabled.
[3]	DAC2N	**Current Generation 2 Negative Output Enable Flag** This bit is set by software, cleared by software. DAC2N = 0, Current generation 2 negative output disabled. DAC2N = 1, Current generation 2 negative output enabled.
[2]	DAC1N	**Current Generation 1 Negative Output Enable Flag** This bit is set by software, cleared by software. DAC1N = 0, Current generation 1 negative output disabled. DAC1N = 1, Current generation 1 negative output enabled.
[1]	DAC2P	**Current Generation 2 Positive Output Enable Flag** This bit is set by software, cleared by software. DAC2P = 0, Current generation 2 positive output disabled. DAC2P = 1, Current generation 2 positive output enabled.
[0]	DAC1P	**Current Generation 1 Positive Output Enable Flag** This bit is set by software, cleared by software. DAC1P = 0, Current generation 1 positive output disabled. DAC1P = 1, Current generation 1 positive output enabled.

Analog Current Generation Control Register

Register	Offset	RW	Description	Reset Value
IGEN1	ANA_BA+0x06	R/W	Analog Current Generation1 Enable Register	0x00

Bits	Flag	Description
[7]	ENIGEN1	**Current Generation 1 Enable Flag** This bit is set by software, cleared by software. ENIGEN1= 0, Current generation 1 disabled. ENIGEN1= 1, Current generation 1 enabled.
[6:0]	IDAC1	**Current Generation 1 Selection Flag** This bit is set by software, cleared by software. IGEN1 = 0, 10uA. IGEN1 = 1, 20uA. IGEN1 = 2, 30uA. IGEN1 = 4, 50uA. IGEN1 = 8, 100uA. IGEN1 = 16, 180uA. IGEN1 = 32, 330uA. IGEN1 = 64, 600uA. IGEN1 = 127, 1200uA.

Analog Current Generation Trim 0 Register

Register	Offset	RW	Description	Reset Value
IGEN2	ANA_BA+0x07	R/W	Analog Current Generation2 Enable Register	0x00

Bits	Flag	Description
[2]	ENIGEN2	**Current Generation 2 Enable Flag** This bit is set by software, cleared by software. ENIGEN2= 0, Current generation 2 disabled. ENIGEN2= 1, Current generation 2 enabled.
[6:0]	IDAC2	**Current Generation 2 Selection Flag** This bit is set by software, cleared by software. IGEN2 = 0, 10uA. IGEN2 = 1, 20uA. IGEN2 = 2, 30uA. IGEN2 = 4, 50uA. IGEN2 = 8, 100uA. IGEN2 = 16, 180uA. IGEN2 = 32, 330uA. IGEN2 = 64, 600uA. IGEN2 = 127, 1200uA.
[4:0]	IDAC1_TRIM	**Current Generation 1 Trim Value** This bit is set by software, cleared by software.

Analog Current Generation Trim 1 Register

Register	Offset	RW	Description	Reset Value
IGENTRIM1	ANA_BA+0x08	R/W	Analog Current Generation Trim 1 Register	0x00

Bits	Flag	Description
[7:5]	Reserved	**Reserved.**
[4:0]	IDAC1_TRIM	**Current Generation 1 Trim Value** This bit is set by software, cleared by software.

Analog Current Generation Trim 2 Register

Register	Offset	RW	Description	Reset Value
IGENTRIM2	ANA_BA+0x09	R/W	Analog Current Generation Trim 2 Register	0x00

Bits	Flag	Description
[7:5]	Reserved	**Reserved.**
[4:0]	IDAC2_TRIM	**Current Generation 2 Trim Value** This bit is set by software, cleared by software.

Analog Bandgap Trim Register

Register	Offset	RW	Description	Reset Value
BGTRIM	ANA_BA+0x0A	R/W	Bandgap Trim Register	0x00

Bits	Flag	Description
[7:6]	Reserved	**Reserved.**
[5:0]	BG_TRIM	**Bandgap Trim Value** This bit is set by software, cleared by software.

6.9 Real-Time Clock (RTC)

6.9.1 Overview

The FE81 series supports a low power real-time clock (RTC) which keeps track of current time information and provides periodic and alarm interrupts. Each counter (second, minute, hour, and day) has its own interrupt enable flag with corresponding interrupt flag providing flexible control of specified time information control. This RTC also supports alarm interrupt and alarm wakeup, where alarm interrupt is at the PFB clock domain and alarm wakeup is at the RTC clock domain which can be used to wakeup system while PFB clock domain being disabled in low power mode. Note that the pulse width of alarm wakeup signal is 1/32.768KHz. Figure 6.9-1 depicts RTC block diagram.

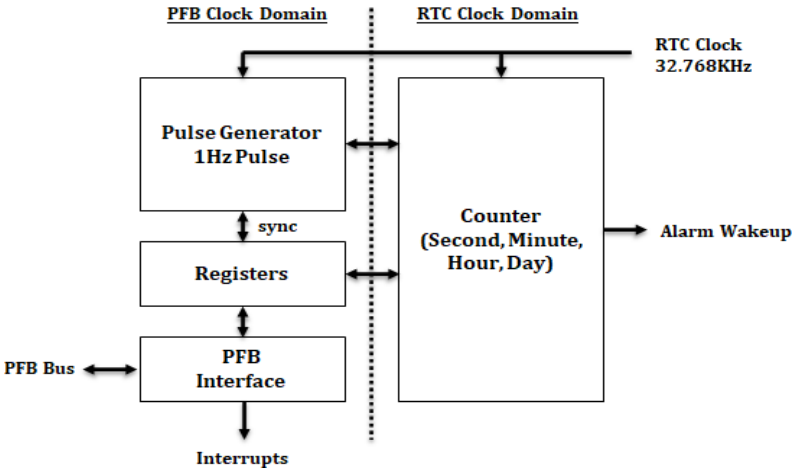


Figure 6.9-1 RTC Block Diagram

RTC Features

- Selective clock sources
  - Internal low speed RC oscillator
  - External low speed crystal (PB1, PB2)
- Periodic interrupts
  - Second interrupt
  - Minute interrupt
  - Hour interrupt
  - Day interrupt
- Programmable alarm interrupt
  - Second alarm
  - Minute alarm
  - Hour alarm
- Alarm wakeup signal to wakeup system in low power mode

6.9.2 Register Map and Description

Base Address (RTC_BA) : 0xEA00				
Register	**Offset**	**RW**	**Description**	**Reset Value**
CNTSEC	RTC_BA+0x00	R/W	RTC Counter Second Register	0x00
CNTMIN	RTC_BA+0x01	R/W	RTC Counter Minute Register	0x00
CNTHOUR	RTC_BA+0x02	R/W	RTC Counter Hour Register	0x00
CNTDAY	RTC_BA+0x03	R/W	RTC Counter Day Register	0x00
ALMSEC	RTC_BA+0x04	R/W	RTC Alarm Second Register	0x00
ALMMIN	RTC_BA+0x05	R/W	RTC Alarm Minute Register	0x00
ALMHOUR	RTC_BA+0x06	R/W	RTC Alarm Hour Register	0x00
RTCCTRL	RTC_BA+0x07	R/W	RTC Control Register	0x00
RTCSTAT	RTC_BA+0x08	R/W	RTC Status Register	0x00
RTCWDONE	RTC_BA+0x09	R	RTC Write Done Register	0x00

RTC Counter Second Register

Register	Offset	RW	Description	Reset Value
CNTSEC	RTC_BA+0x00	R/W	RTC Counter Second Register	0x00

Bits	Flag	Description
[7:6]	Reserved	**Reserved.**
[5:0]	CNTSEC	**Second Field of Current Time** This bit is set by software, cleared by software. CNTSEC = 0 ~ 59

RTC Counter Minute Register

Register	Offset	RW	Description	Reset Value
CNTMIN	RTC_BA+0x01	R/W	RTC Counter Minute Register	0x00

Bits	Flag	Description
[7:6]	Reserved	**Reserved.**
[5:0]	CNTMIN	**Minute Field of Current Time** This bit is set by software, cleared by software. CNTMIN = 0 ~ 59

RTC Counter Hour Register

Register	Offset	RW	Description	Reset Value
CNTHOUR	RTC_BA+0x02	R/W	RTC Counter Hour Register	0x00

Bits	Flag	Description
[7:5]	Reserved	**Reserved.**
[4:0]	CNTHOUR	**Hour Field of Current Time** This bit is set by software, cleared by software. CNTHOUR = 0 ~ 23

RTC Counter Day Register

Register	Offset	RW	Description	Reset Value
CNTDAY	RTC_BA+0x03	R/W	RTC Counter Day Register	0x00

Bits	Flag	Description
[7:5]	Reserved	**Reserved.**
[4:0]	CNTDAY	**Day Field of Current Time** This bit is set by software, cleared by software. CNTDAY = 0 ~ 30

RTC Alarm Second Register

Register	Offset	RW	Description	Reset Value
ALMSEC	RTC_BA+0x04	R/W	RTC Alarm Second Register	0x00

Bits	Flag	Description
[7:6]	Reserved	**Reserved.**
[5:0]	ALMSEC	**Second Field of Alarm** This bit is set by software, cleared by software. ALMSEC = 0 ~ 59

RTC Alarm Minute Register

Register	Offset	RW	Description	Reset Value
ALMMIN	RTC_BA+0x05	R/W	RTC Alarm Minute Register	0x00

Bits	Flag	Description
[7:6]	Reserved	**Reserved.**
[5:0]	ALMMIN	**Minute Field of Alarm** This bit is set by software, cleared by software. ALMMIN = 0 ~ 59

RTC Alarm Hour Register

Register	Offset	RW	Description	Reset Value
ALMHOUR	RTC_BA+0x06	R/W	RTC Alarm Hour Register	0x00

Bits	Flag	Description
[7:5]	Reserved	**Reserved.**
[4:0]	ALMHOUR	**Hour Field of Alarm** This bit is set by software, cleared by software. ALMHOUR = 0 ~ 23

RTC Control Register

Register	Offset	RW	Description	Reset Value
RTCCTRL	RTC_BA+0x07	R/W	RTC Control Register	0x00

Bits	Flag	Description
[7]	Reserved	**Reserved.**
[6]	SECIEN	**RTC Second Interrupt Enable Flag** This bit is set by software, cleared by software. Second interrupt is generated when one second passed. SECIEN = 0, RTC second interrupt disabled. SECIEN = 1, RTC second interrupt enabled.
[5]	MINIEN	**RTC Minute Interrupt Enable Flag** This bit is set by software, cleared by software. Minute interrupt is generated when second of RTC time changes from 59 to 0. MINIEN = 0, RTC minute interrupt disabled. MINIEN = 1, RTC minute interrupt enabled.
[4]	HORIEN	**RTC Hour Interrupt Enable Flag** This bit is set by software, cleared by software. Hour interrupt is generated when minute of RTC time changes from 59 to 0. HORIEN = 0, RTC hour interrupt disabled. HORIEN = 1, RTC hour interrupt enabled.
[3]	DAYIEN	**RTC Day Interrupt Enable Flag** This bit is set by software, cleared by software. Day interrupt is generated when hour of RTC time changes from 23 to 0. DAYIEN = 0, RTC day interrupt disabled. DAYIEN = 1, RTC day interrupt enabled.
[2]	ALMIEN	**RTC Alarm Interrupt Enable Flag** This bit is set by software, cleared by software. ALMIEN = 0, RTC wakeup interrupt disabled. ALMIEN = 1, RTC wakeup interrupt enabled.
[1]	ALMWEN	**RTC Alarm Wakeup Enable Flag** This bit is set by software, cleared by software. ALMWEN = 0, RTC wakeup function disabled. ALMWEN = 1, RTC wakeup function enabled.
[0]	RTCEN	**RTC Enable Flag** This bit is set by software, cleared by software. RTCEN = 0, RTC idle. RTCEN = 1, RTC enabled.

RTC Status Register

Register	Offset	RW	Description	Reset Value
RTCSTAT	RTC_BA+0x08	R/W	RTC Status Register	0x00

Bits	Flag	Description
[7]	Reserved	**Reserved.**
[6]	SECINTF	**RTC Second Interrupt Flag** This bit is set by software, cleared by software (Write 1 to clear). SECINTF = 0, Idle. SECINTF = 1, RTC second interrupt triggered.
[5]	MININTF	**RTC Minute Interrupt Flag** This bit is set by software, cleared by software (Write 1 to clear). MININTF = 0, Idle. MININTF = 1, RTC minute interrupt triggered.
[4]	HORINTF	**RTC Hour Interrupt Flag** This bit is set by software, cleared by software (Write 1 to clear). HORINTF = 0, Idle. HORINTF = 1, RTC hour interrupt triggered.
[3]	DAYINTF	**RTC Day Interrupt Flag** This bit is set by software, cleared by software (Write 1 to clear). DAYINTF = 0, Idle. DAYINTF = 1, RTC day interrupt triggered.
[2]	ALMINTF	**RTC Alarm Interrupt Flag** This bit is set by software, cleared by software (Write 1 to clear). ALMINTF = 0, Idle. ALMINTF = 1, RTC wakeup interrupt triggered.
[1:0]	Reserved	**Reserved**.

RTC Write Done Register

Register	Offset	RW	Description	Reset Value
RTCWDONE	RTC_BA+0x09	R	RTC Write Done Register	0x00

Bits	Flag	Description
[7:1]	Reserved	**Reserved.**
[0]	WDONE	**RTC Write Done Flag** This bit is set by hardware. See section 6.9.3 for detail description. WDONE = 0, Synchronous in progress. WDONE = 1, Registers write is synchronized to the RTC clock domain.

6.9.3 Programming Model

While updating RTC registers (Counters, Alarm, and Control registers, etc), all registers must be synchronized to the RTC clock domain. The RTC provides a RTCWDONE Register for user to check the process of synchronous. While updating registers, the WDONE flag in RTCWDONE Register becomes zero and returns to one when all prior updates have been successfully synchronized to the RTC clock domain. For usage convenient, each of the RTC registers is synchronized independently while their synchronization status is lumped into this single bit. Thus, writes to different RTC registers can be done in a batch before checking this bit. Note that while an RTC register update is being synchronized to the RTC clock domain, a second update to the same register may be dropped.

Since the frequency of the RTC clock is quite slow when compared to the typical frequency of the PFB clock, the synchronization period can be pretty long. The PFB clock domain should not be shut down while the synchronization is still in progress.

Adjust time and interrupts on the “Hour”

Step 1 :

Wait until the WDONE field of the RTCWDONE Register equals 1.

Step 2 :

Set the counter register to the current time (CNTSEC, CNTMIN, CNTHOUR Registers); set the CNTDAY to 0 to count from day 0.

Step 3 :

Enable the RTC and hourly interrupt: set the RTCCTRL Register to 0x11.

Trigger an alarm interrupt at a specific time

Step 1 :

Wait until the WDONE field of the RTCWDONE Register equals 1.

Step 2 :

Set the alarm register (ALMSEC, ALMMIN, ALMHOUR Registers) to the time you want to alarm.

Step 3 :

Enable the RTC and alarm: set the RTCCTRL Register to 0x5.

Trigger an alarm wakeup signal at a specific time

Step 1 :

Wait until the WDONE field of the RTCWDONE Register equals 1.

Step 2 :

Set the alarm register (ALMSEC, ALMMIN, ALMHOUR Registers) to the time you want to wake up the system.

Step 3 :

Enable the RTC and alarm wakeup: set the RTCCTRL Register to 0x3.

6.10 Watchdog Timer Controller (WDT)

6.10.1 Overview

Watchdog timer (WDT) is used to prevent a system from lock-up when program execution goes astray. Furthermore, system is able to be wakeup by WDT while in low power mode. Figure 6.10-1 shows the block diagram of WDT.

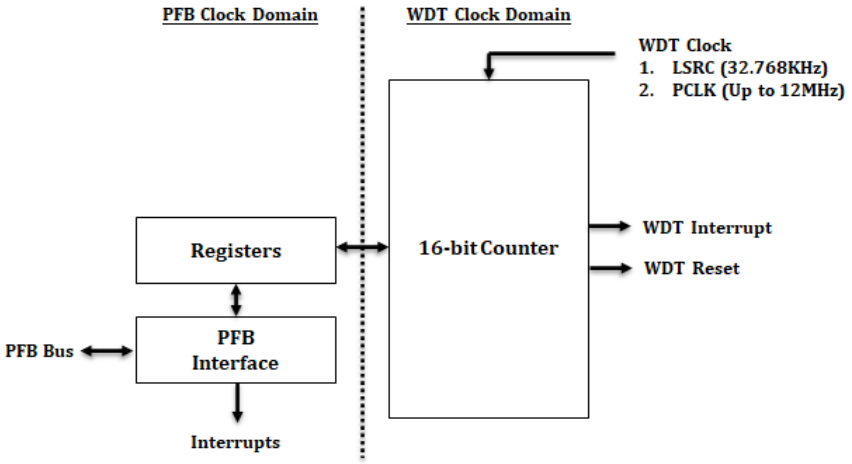


Figure 6.10-1 WDT Block Diagram

WDT Features

- Programmable source of timer clock
- Provides combinations of interrupt and reset when the watchdog timer expires
- Programmable interrupt time for interrupt generation
  - (1/WDTCLK) x (2^n, n = 6 ~ 15)
- Programmable reset time for reset generation
  - (1/WDTCLK) x (2^n, n = 7 ~ 10)
- Provides a write protection mechanism for the Control or Restart registers
- Specific magic numbers for write protection of registers and restart of the timer
- Watchdog timer could be externally paused by FMCCON Register (WDTOF) of FMC

The watchdog timer (WDT) provides a two-stage mechanism to prevent a system from lock-up. The first stage is called “interrupt stage”. If the watchdog interrupt is enabled and the watchdog timer is not restarted during the interrupt stage, the interrupt signal, WDT Interrupt, will be asserted. The second stage, reset stage, begins right after the interrupt stage. If the watchdog reset stage is enabled and the watchdog timer is not restarted during the reset stage, the reset signal, WDT Reset, will be asserted. Figure 6.10-2 depicts the 2 stages of the WDT.

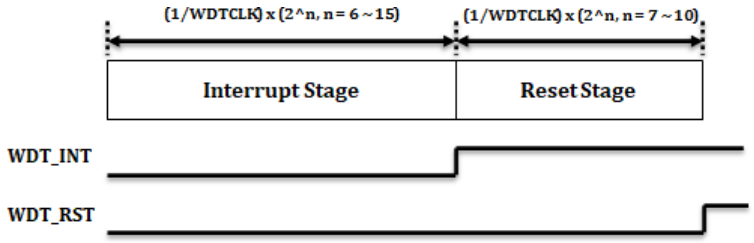


Figure 6.10-2 2 Stages of WDT

The WDTCTRL0, WDTCTRL1 Registers and RESTART Register (Section 6.10.2) should be programmed through a two-step write scheme. The WRPROT Register (Section 6.10.2) should be programmed with a magic number “0xA5” to disable the write protection before either of the registers could be updated. The register write-protection is enabled again upon the subsequent write to any of the watchdog registers is received.

When the RESTART Register is written with the preconfigured value “0x5A”, it restarts the interrupt timer and cancels the system reset timer. When the value is not “0x5A”, the controller just ignores the write.

6.10.2 Register Map and Description

Base Address (WDT_BA) : 0xEC00				
Register	**Offset**	**RW**	**Description**	**Reset Value**
WDTCTRL0	WDT_BA+0x00	R/W	WDT Control 0 Register	0x00
WDTCTRL1	WDT_BA+0x01	R/W	WDT Control 1 Register	0x00
RESTART	WDT_BA+0x02	W	WDT Restart Register	0x00
WRPROT	WDT_BA+0x03	W	WDT Write Protect Register	0x00
STATUS	WDT_BA+0x04	R	WDT Status Register	0x00

WDT Control 0 Register

Register	Offset	RW	Description	Reset Value
WDTCTRL0	WDT_BA+0x00	R/W	WDT Control 0 Register	0x00

Bits	Flag	Description
[7:6]	Reserved	**Reserved.**
[5:4]	RSTTIME	**WDT Reset Time Interval Selection Flag** This bit is set by software, cleared by software. RSTTIME = 00, Clock period x 128. RSTTIME = 01, Clock period x 256. RSTTIME = 10, Clock period x 512. RSTTIME = 11, Clock period x 1024
[3]	Reserved	**Reserved.**
[2:0]	INTTIME	**WDT Interrupt Time Interval Selection Flag** This bit is set by software, cleared by software. INTTIME = 000, Clock period x 64. INTTIME = 001, Clock period x 256. INTTIME = 010, Clock period x 1024. INTTIME = 011, Clock period x 2048 INTTIME = 100, Clock period x 4096. INTTIME = 101, Clock period x 8192. INTTIME = 110, Clock period x 16384. INTTIME = 111, Clock period x 32768

WDT Control 1 Register

Register	Offset	RW	Description	Reset Value
WDTCTRL1	WDT_BA+0x01	R/W	WDT Control 1 Register	0x00

Bits	Flag	Description
[7:4]	Reserved	**Reserved.**
[3]	RSTEN	**WDT Reset Enable Flag** This bit is set by software, cleared by software. RSTEN = 0, WDT reset disable. RSTEN = 1, WDT reset enable.
[2]	INTEN	**WDT Interrupt Enable Flag** This bit is set by software, cleared by software. INTEN = 0, WDT interrupt disable. INTEN = 1, WDT interrupt enable.
[1]	CLKSEL	**WDT Internal Clock Selection Flag** This bit is set by software, cleared by software. Note that while using WDT for low power mode wakeup, the clock source should be selected to LSRC. CLKSEL = 0, LSRC. CLKSEL = 1, PCLK.
[0]	WDTEN	**WDT Enable Flag** This bit is set by software, cleared by software. WDTEN = 0, WDT disable. WDTEN = 1, WDT enable.

WDT Restart Register

Register	Offset	RW	Description	Reset Value
RESTART	WDT_BA+0x02	W	WDT Restart Register	0x00

Bits	Flag	Description
[7:0]	RESTART	**WDT RESTART Magic Number** This bit is set by software, cleared by software. RESTART = 0x5A to restart the WDT.

WDT Write Protect Register

Register	Offset	RW	Description	Reset Value
WRPROT	WDT_BA+0x03	W	WDT Write Protect Register	0x00

Bits	Flag	Description
[7:0]	WRPROT	**WDT Write Protect Magic Number** This bit is set by software, cleared by software. WRPROT = 0xA5 to disable write protection of the registers.

WDT Status Register

Register	Offset	RW	Description	Reset Value
STATUS	WDT_BA+0x04	R	WDT Status Register	0x00

Bits	Flag	Description
[7:1]	Reserved	**Reserved.**
[0]	WDTINTF	**WDT Interrupt Flag** This bit is set by Hardware, cleared by Software (Write 1 to clear). WDTINTF = 0, WDT is not expired. WDTINTF = 1, WDT is expired.

6.10.3 Programming Model

Setup and enable WDT

Step 1 :

Write “0xA5” to the WRPROT Register.

Step 2 :

Select the clock source (CLKSEL bit field in WDTCTRL1 Register).

Set the intervals of interrupt (INTTIME) and reset (RSTTIME).

Enable the WDT interrupt (INTEN), reset (RSTEN).

Enable the WDT (WDTEN = 1).

Restart the WDT

Once WDT is enabled, it must be restarted by software to avoid the WDT interrupt/reset.

Step 1 :

Write “0xA5” to the WRPROT Register.

Step 2 :

Write “0x5A” to the RESTART Register.

Disable the WDT

Step 1 :

Write “0xA5” to the WRPROT Register.

Step 2 :

Disable the WDT (WDTEN = 0).

6.11 Timer Controller (TMR)

6.11.1 Overview

Timer controller can be used for counting, clock generation, event counting, and frequency measurement, etc. The FE81 series supports 3 timer controllers, which is TMR0, TMR1, and TMR2. Each timer (TMRx, x = 0 ~ 2) has 2-channel, and each channel supports 6 multi-function modes. Figure 6.11-1 illustrates the block diagram of TMR.

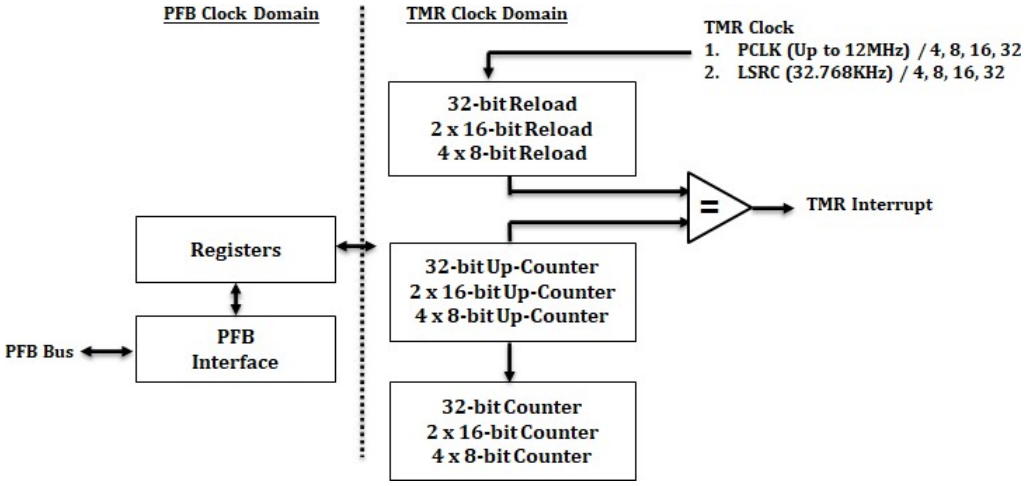


Figure 6.11-1 Timer Block Diagram

TMR Features

- Programmable source of timer clock
- 2-channel with each supports 6 multi-function timers
  - 1 32-bit Timer / 2 16-bit Timers / 4 8-bit Timers / PWM / PWM+16-bit Timer / PWM+8-bit Timers
- Externally paused by TMR0ST, TMR1ST, TMR2ST bit field defined in SMU TMRSTAL Register

Channel Mode	32-bit Timer	16-bit Timers	8-bit Timers	PWM	Mixed PWM/16-bit Timer	Mixed PWM/8-bit Timers
Channel Function	32-bit Timer 0	16-bit Timer 0 16-bit Timer 1	8-bit Timer 0 8-bit Timer 1 8-bit Timer 2 8-bit Timer 3	16-bit PWM	16-bit Timer 0 8-bit PWM	8-bit Timer 0 8-bit Timer 1 8-bit PWM

An n-bit timer means the timer contains an n-bit counter to generate periodic interrupts. An n-bit PWM means the PWM contains two n-bit counters to generate periodic square waves with programmable duty cycles. The more bits a counter contains, the larger period it can support. Table above shows the effective devices of the corresponding channel modes.

The reload register (CHxRL3, CHxRL2, CHxRL1, CHxRL0, x = 0, 1) keeps the initial/reload value(s) for timer/PWM counter(s). The definition of the reload register varies according to the channel mode. The period of timer or PWM is equal to the reload value plus one. For example, in the 32-bit timer mode, a timer interrupt will be generated every ({CH0RL3, CH0RL2, CH0RL1, CH0RL0} + 1) cycles. At the PWM mode, the high period is ({CH0RL3, CH0RL2} + 1) cycles and the low period is ({CH0RL1, CH0RL0} + 1) cycles.

Channel counter register (CHxCNT3, CHxCNT2, CHxCNT1, CHxCNT0, x = 0, 1) indicates the remaining cycles for the next timer interrupt or PWM toggle. Like the reload register, the field definition of counter register varies according to the channel mode.

6.11.2 Register Map and Description

Base Address (TMR_BA) : 0xF600, 0xF800, 0xFA00				
Register	**Offset**	**RW**	**Description**	**Reset Value**
INTEN	TMR_BA+0x00	R/W	Timer Interrupt Enable Register	0x00
INTSTAT	TMR_BA+0x01	R/W	Timer Interrupt Status Register	0x00
CHEN	TMR_BA+0x02	R/W	Timer Channel Enable Register	0x00
CH0CTRL	TMR_BA+0x03	R/W	Timer Channel 0 Control Register	0x00
CH1CTRL	TMR_BA+0x04	R/W	Timer Channel 1 Control Register	0x00
CH0RL0	TMR_BA+0x05	R/W	Timer Channel 0 Reload 0 Register	0x00
CH0RL1	TMR_BA+0x06	R/W	Timer Channel 0 Reload 1 Register	0x00
CH0RL2	TMR_BA+0x07	R/W	Timer Channel 0 Reload 2 Register	0x00
CH0RL3	TMR_BA+0x08	R/W	Timer Channel 0 Reload 3 Register	0x00
CH1RL0	TMR_BA+0x09	R/W	Timer Channel 1 Reload 0 Register	0x00
CH1RL1	TMR_BA+0x0A	R/W	Timer Channel 1 Reload 1 Register	0x00
CH1RL2	TMR_BA+0x0B	R/W	Timer Channel 1 Reload 2 Register	0x00
CH1RL3	TMR_BA+0x0C	R/W	Timer Channel 1 Reload 3 Register	0x00
CH0CNT0	TMR_BA+0x0D	R/W	Timer Channel 0 Counter 0 Register	0x00
CH0CNT1	TMR_BA+0x0E	R/W	Timer Channel 0 Counter 1 Register	0x00
CH0CNT2	TMR_BA+0x0F	R/W	Timer Channel 0 Counter 2 Register	0x00
CH0CNT3	TMR_BA+0x10	R/W	Timer Channel 0 Counter 3 Register	0x00
CH1CNT0	TMR_BA+0x11	R/W	Timer Channel 1 Counter 0 Register	0x00
CH1CNT1	TMR_BA+0x12	R/W	Timer Channel 1 Counter 1 Register	0x00
CH1CNT2	TMR_BA+0x13	R/W	Timer Channel 1 Counter 2 Register	0x00
CH1CNT3	TMR_BA+0x14	R/W	Timer Channel 1 Counter 3 Register	0x00

Timer Interrupt Enable Register

Register	Offset	RW	Description	Reset Value
INTEN	TMR_BA+0x00	R/W	Timer Interrupt Enable Register	0x00

Bits	Flag	Description
[7]	CH1IEN3	**Channel 1 Timer 3 Interrupt Enable Flag** This bit is set by software, cleared by software. If 4x8-bit timer mode is used, the bit field is set. CH1IEN3 = 0, Idle. CH1IEN3 = 1, Channel 1 timer 3 interrupt enabled.
[6]	CH1IEN2	**Channel 1 Timer 2 Interrupt Enable Flag** This bit is set by software, cleared by software. If 4x8-bit timer mode is used, the bit field is set. CH1IEN2 = 0, Idle. CH1IEN2 = 1, Channel 1 timer 2 interrupt enabled.
[5]	CH1IEN1	**Channel 1 Timer 1 Interrupt Enable Flag** This bit is set by software, cleared by software. If 2x16-bit timer, 4x8-bit timer mode is used, the bit field is set. CH1IEN1 = 0, Idle. CH1IEN1 = 1, Channel 1 timer 1 interrupt enabled.
[4]	CH1IEN0	**Channel 1 Timer 0 Interrupt Enable Flag** This bit is set by software, cleared by software. If 1x32-bit timer, 2x16-bit timer, 4x8-bit timer mode is used, the bit field is set. CH1IEN0 = 0, Idle. CH1IEN0 = 1, Channel 1 timer 0 interrupt enabled.
[3]	CH0IEN3	**Channel 0 Timer 3 Interrupt Enable Flag** This bit is set by software, cleared by software. If 4x8-bit timer mode is used, the bit field is set. CH0IEN3 = 0, Idle. CH0IEN3 = 1, Channel 0 timer 3 interrupt enabled.
[2]	CH0IEN2	**Channel 0 Timer 2 Interrupt Enable Flag** This bit is set by software, cleared by software. If 4x8-bit timer mode is used, the bit field is set. CH0IEN2 = 0, Idle. CH0IEN2 = 1, Channel 0 timer 2 interrupt enabled.
[1]	CH0IEN1	**Channel 0 Timer 1 Interrupt Enable Flag** This bit is set by software, cleared by software. If 2x16-bit timer, 4x8-bit timer mode is used, the bit field is set. CH0IEN1 = 0, Idle. CH0IEN1 = 1, Channel 0 timer 1 interrupt enabled.
[0]	CH0IEN0	**Channel 0 Timer 0 Interrupt Enable Flag** This bit is set by software, cleared by software. If 1x32-bit timer, 2x16-bit timer, 4x8-bit timer mode is used, the bit field is set. CH0IEN0 = 0, Idle. CH0IEN0 = 1, Channel 0 timer 0 interrupt enabled.

Timer Interrupt Status Register

Register	Offset	RW	Description	Reset Value
INTSTAT	TMR_BA+0x01	R/W	Timer Interrupt Status Register	0x00

Bits	Flag	Description
[7]	CH1INF3	**Channel 1 Timer 3 Interrupt Status Flag** This bit is set by hardware, cleared by software (Write 1 to clear). CH1INF3 = 0, No effect. CH1INF3 = 1, Channel 1 timer 3 time up.
[6]	CH1INF2	**Channel 1 Timer 2 Interrupt Status Flag** This bit is set by hardware, cleared by software (Write 1 to clear). CH1INF2 = 0, No effect. CH1INF2 = 1, Channel 1 timer 2 time up.
[5]	CH1INF1	**Channel 1 Timer 1 Interrupt Status Flag** This bit is set by hardware, cleared by software (Write 1 to clear). CH1INF1 = 0, No effect. CH1INF1 = 1, Channel 1 timer 1 time up.
[4]	CH1INF0	**Channel 1 Timer 0 Interrupt Status Flag** This bit is set by hardware, cleared by software (Write 1 to clear). CH1INF0 = 0, No effect. CH1INF0 = 1, Channel 1 timer 0 time up.
[3]	CH0INF3	**Channel 0 Timer 3 Interrupt Status Flag** This bit is set by hardware, cleared by software (Write 1 to clear). CH0INF3 = 0, No effect. CH0INF3 = 1, Channel 0 timer 3 time up.
[2]	CH0INF2	**Channel 0 Timer 2 Interrupt Status Flag** This bit is set by hardware, cleared by software (Write 1 to clear). CH0INF2 = 0, No effect. CH0INF2 = 1, Channel 0 timer 2 time up.
[1]	CH0INF1	**Channel 0 Timer 1 Interrupt Status Flag** This bit is set by hardware, cleared by software (Write 1 to clear). CH0INF1 = 0, No effect. CH0INF1 = 1, Channel 0 timer 1 time up.
[0]	CH0INF0	**Channel 0 Timer 0 Interrupt Status Flag** This bit is set by hardware, cleared by software (Write 1 to clear). CH0INF0 = 0, No effect. CH0INF0 = 1, Channel 0 timer 0 time up.

Timer Channel Enable Register

Register	Offset	RW	Description	Reset Value
CHEN	TMR_BA+0x02	R/W	Timer Channel Enable Register	0x00

Bits	Flag	Description
[7]	CH1TEN3 / CH1PEN	**Channel 1 Timer 3 / PWM Enable Flag** This bit is set by software, cleared by software. If 4x8-bit timer mode is used, the bit field is set. CH1TEN3 = 0, Idle. CH1TEN3 = 1, Channel 1 timer 3 enable. If PWM mode is used, the bit field is set. CH1PEN = 0, Idle. CH1PEN = 1, Channel 1 PWM enable.
[6]	CH1TEN2	**Channel 1 Timer 2 Enable Flag** This bit is set by software, cleared by software. If 4x8-bit timer mode is used, the bit field is set. CH1TEN2 = 0, Idle. CH1TEN2 = 1, Channel 1 timer 2 enable.
[5]	CH1TEN1	**Channel 1 Timer 1 Enable Flag** This bit is set by software, cleared by software. If 2x16-bit timer, 4x8-bit timer mode is used, the bit field is set. CH1TEN1 = 0, Idle. CH1TEN1 = 1, Channel 1 timer 1 enable.
[4]	CH1TEN0	**Channel 1 Timer 0 Enable Flag** This bit is set by software, cleared by software. If 1x32-bit timer, 2x16-bit timer, 4x8-bit timer mode is used, the bit field is set. CH1TEN0 = 0, Idle. CH1TEN0 = 1, Channel 1 timer 0 enable.
[3]	CH0TEN3 / CH0PEN	**Channel 0 Timer 3 / PWM Enable Flag** This bit is set by software, cleared by software. If 4x8-bit timer mode is used, the bit field is set. CH0TEN3 = 0, Idle. CH0TEN3 = 1, Channel 0 timer 3 enable. If PWM mode is used, the bit field is set. CH0PEN = 0, Idle. CH0PEN = 1, Channel 0 PWM enable.
[2]	CH0TEN2	**Channel 0 Timer 2 Enable Flag** This bit is set by software, cleared by software. If 4x8-bit timer mode is used, the bit field is set. CH0TEN2 = 0, Idle. CH0TEN2 = 1, Channel 0 timer 2 enable.
[1]	CH0TEN1	**Channel 0 Timer 1 Enable Flag** This bit is set by software, cleared by software. If 2x16-bit timer, 4x8-bit timer mode is used, the bit field is set. CH0TEN1 = 0, Idle. CH0TEN1 = 1, Channel 0 timer 1 enable.
[0]	CH0TEN0	**Channel 0 Timer 0 Enable Flag** This bit is set by software, cleared by software. If 1x32-bit timer, 2x16-bit timer, 4x8-bit timer mode is used, the bit field is set. CH0TEN0 = 0, Idle. CH0TEN0 = 1, Channel 0 timer 0 enable.

Note that a timer or PWM cannot be enabled if the corresponding channel does not exist or it is not a valid device in the channel mode. For example, Timer 1 of Channel 0 cannot be enabled when Channel 0 is set to the 32-bit Timer mode.

Timer Channel 0 Control Register

Register	Offset	RW	Description	Reset Value
CH0CTRL	TMR_BA+0x03	R/W	Timer Channel 0 Control Register	0x00

Bits	Flag	Description
[7:5]	Reserved	**Reserved**.
[4]	PWMPRK	**Channel 0 PWM Park Value** This bit is set by software, cleared by software. PWMPark = PWM park value, output of PWM when PWM is disabled.
[3]	PWMPOL	**Channel 0 PWM Polarity Flag** This bit is set by hardware, cleared by software. PWMPOL = 0, PWM high pulse first. PWMPOL = 1, PWM low pulse first.
[2:0]	CHMODE	**Channel 0 Channel Mode Selection** This bit is set by hardware, cleared by software. CHMode = 0, Reserved. CHMode = 1, 32-bit timer. CHMode = 2, 16-bit timers. CHMode = 3, 8-bit timers. CHMode = 4, PWM. CHMode = 5, Reserved. CHMode = 6, Mixed PWM/16-bit timer. CHMode = 7, Mixed PWM/8-bit timer.

Timer Channel 1 Control Register

Register	Offset	RW	Description	Reset Value
CH1CTRL	TMR_BA+0x04	R/W	Timer Channel 1 Control Register	0x00

Bits	Flag	Description
[7:5]	Reserved	**Reserved**.
[4]	PWMPRK	**Channel 1 PWM Park Value** This bit is set by hardware, cleared by software. PWMPark = PWM park value, output of PWM when PWM is disabled.
[3]	PWMPOL	**Channel 1 PWM Polarity Flag** This bit is set by hardware, cleared by software. PWMPOL = 0, PWM high pulse first. PWMPOL = 1, PWM low pulse first.
[2:0]	CHMODE	**Channel 1 Channel Mode Selection** This bit is set by hardware, cleared by software. CHMode = 0, Reserved. CHMode = 1, 32-bit timer. CHMode = 2, 16-bit timers. CHMode = 3, 8-bit timers. CHMode = 4, PWM. CHMode = 5, Reserved. CHMode = 6, Mixed PWM/16-bit timer. CHMode = 7, Mixed PWM/8-bit timer.

Timer Channel 0 Reload x Register, x = 0, 1, 2, 3

Register	Offset	RW	Description	Reset Value
CH0RL0	TMR_BA+0x05	R/W	Timer Channel 0 Reload 0 Register	0x00

Register	Offset	RW	Description	Reset Value
CH0RL1	TMR_BA+0x06	R/W	Timer Channel 0 Reload 1 Register	0x00

Register	Offset	RW	Description	Reset Value
CH0RL2	TMR_BA+0x07	R/W	Timer Channel 0 Reload 2 Register	0x00

Register	Offset	RW	Description	Reset Value
CH0RL3	TMR_BA+0x08	R/W	Timer Channel 0 Reload 3 Register	0x00

Bits	Flag	Description
[7:0]	CH0RLx, x = 0, 1, 2, 3	**Channel 0 Reload x Value, x = 0, 1, 2, 3**

1. 1x32-bit Reload Value
   - CH0RL32[31:0] = {CH0RL0, CH0RL1, CH0RL2, CH0RL3}

     CH0RL0 is the most significant byte and CH0RL3 is the least significant byte.
2. 2x16-bit Reload Value
   - CH0RL16_0[15:0] = {CH0RL0, CH0RL1}

     CH0RL0 is the most significant byte and CH0RL1 is the least significant byte.
   - CH0RL16_1[15:0] = {CH0RL2, CH0RL3}

     CH0RL2 is the most significant byte and CH0RL3 is the least significant byte.
3. 4x8-bit Reload Value
   - CH0RL8_0[7:0] = {CH0RL0}
   - CH0RL8_1[7:0] = {CH0RL1}
   - CH0RL8_2[7:0] = {CH0RL2}
   - CH0RL8_3[7:0] = {CH0RL3}

Timer Channel 1 Reload x Register, x = 0, 1, 2, 3

Register	Offset	RW	Description	Reset Value
CH1RL0	TMR_BA+0x09	R/W	Timer Channel 1 Reload 0 Register	0x00

Register	Offset	RW	Description	Reset Value
CH1RL1	TMR_BA+0x0A	R/W	Timer Channel 1 Reload 1 Register	0x00

Register	Offset	RW	Description	Reset Value
CH1RL2	TMR_BA+0x0B	R/W	Timer Channel 1 Reload 2 Register	0x00

Register	Offset	RW	Description	Reset Value
CH1RL3	TMR_BA+0x0C	R/W	Timer Channel 1 Reload 3 Register	0x00

Bits	Flag	Description
[7:0]	CH1RLx, x = 0, 1, 2, 3	**Channel 1 Reload x Value, x = 0, 1, 2, 3**

1. 1x32-bit Reload Value
   - CH1RL32[31:0] = {CH1RL0, CH1RL1, CH1RL2, CH1RL3}

     CH1RL0 is the most significant byte and CH1RL3 is the least significant byte.
2. 2x16-bit Reload Value
   - CH1RL16_0[15:0] = {CH1RL0, CH1RL1}

     CH1RL0 is the most significant byte and CH1RL1 is the least significant byte.
   - CH1RL16_1[15:0] = {CH1RL2, CH1RL3}

     CH1RL2 is the most significant byte and CH1RL3 is the least significant byte.
3. 4x8-bit Reload Value
   - CH1RL8_0[7:0] = {CH1RL0}
   - CH1RL8_1[7:0] = {CH1RL1}
   - CH1RL8_2[7:0] = {CH1RL2}
   - CH1RL8_3[7:0] = {CH1RL3}

Timer Channel 0 Counter x Register, x = 0, 1, 2, 3

Register	Offset	RW	Description	Reset Value
CH0CNT0	TMR_BA+0x0D	R/W	Timer Channel 0 Counter 0 Register	0x00

Register	Offset	RW	Description	Reset Value
CH0CNT1	TMR_BA+0x0E	R/W	Timer Channel 0 Counter 1 Register	0x00

Register	Offset	RW	Description	Reset Value
CH0CNT2	TMR_BA+0x0F	R/W	Timer Channel 0 Counter 2 Register	0x00

Register	Offset	RW	Description	Reset Value
CH0CNT3	TMR_BA+0x10	R/W	Timer Channel 0 Counter 3 Register	0x00

Bits	Flag	Description
[7:0]	CH0CNTx, x = 0, 1, 2, 3	**Channel 0 Counter x Value, x = 0, 1, 2, 3**

1. 1x32-bit Counter Value
   - CH0CNT32[31:0] = {CH0CNT0, CH0CNT1, CH0CNT2, CH0CNT3}

     CH0CNT0 is the most significant byte and Ch0CNT3 is the least significant byte.
2. 2x16-bit Counter Value
   - CH0CNT16_0[15:0] = {CH0CNT0, CH0CNT1}

     CH0CNT0 is the most significant byte and Ch0CNT1 is the least significant byte.
   - CH0CNT16_1[15:0] = {CH0CNT2, CH0CNT3}

     CH0CNT2 is the most significant byte and Ch0CNT3 is the least significant byte.
3. 4x8-bit Counter Value
   - CH0CNT8_0[7:0] = {CH0CNT0}
   - CH0CNT8_1[7:0] = {CH0CNT1}
   - CH0CNT8_2[7:0] = {CH0CNT2}
   - CH0CNT8_3[7:0] = {CH0CNT3}

Timer Channel 1 Counter x Register, x = 0, 1, 2, 3

Register	Offset	RW	Description	Reset Value
CH1CNT0	TMR_BA+0x11	R/W	Timer Channel 1 Counter 0 Register	0x00

Register	Offset	RW	Description	Reset Value
CH1CNT1	TMR_BA+0x12	R/W	Timer Channel 1 Counter 1 Register	0x00

Register	Offset	RW	Description	Reset Value
CH1CNT2	TMR_BA+0x13	R/W	Timer Channel 1 Counter 2 Register	0x00

Register	Offset	RW	Description	Reset Value
CH1CNT3	TMR_BA+0x14	R/W	Timer Channel 1 Counter 3 Register	0x00

Bits	Flag	Description
[7:0]	CH1CNTx, x = 0, 1, 2, 3	**Channel 1 Counter x Value, x = 0, 1, 2, 3**

1. 1x32-bit Counter Value
   - CH1CNT32[31:0] = {CH1CNT0, CH1CNT1, CH1CNT2, CH1CNT3}

     CH1CNT0 is the most significant byte and Ch1CNT1 is the least significant byte.
2. 2x16-bit Counter Value
   - CH1CNT16_0[15:0] = {CH1CNT0, CH1CNT1}

     CH1CNT0 is the most significant byte and Ch1CNT1 is the least significant byte.
   - CH1CNT16_1[15:0] = {CH1CNT2, CH1CNT3}

     CH1CNT2 is the most significant byte and Ch1CNT3 is the least significant byte.
3. 4x8-bit Counter Value
   - CH1CNT8_0[7:0] = {CH1CNT0}
   - CH1CNT8_1[7:0] = {CH1CNT1}
   - CH1CNT8_2[7:0] = {CH1CNT2}
   - CH1CNT8_3[7:0] = {CH1CNT3}

6.11.3 Programming Model (Please refer to “Sample code project”)

Below describes the usages of TMR.

One-shot : TMR channel 0 32-bit Timer

Set CHMODE field of CH0CTRL Register to 1 (32-bit timer mode)

Set CH0RLx, x = 0, 1, 2, 3, to the value reloaded

Set CH0IEN0 field of INTEN Register to 1

Set CH0TEN0 field of CHEN Register to 1

TMR ISR :

Set CH0INTF0 field of INTSTAT to 1 to clear interrupt flag

Set CH0TEN0 field of CHEN Register to 0

Periodic : TMR channel 0 16-bit Timer, channel 1 16-bit Timer

Set CHMODE field of CH0CTRL Register to 2 (16-bit timer mode)

Set CHMODE field of CH1CTRL Register to 2 (16-bit timer mode)

Set CH0RLx, x = 0, 1, to the value reloaded

Set CH1RLx, x = 0, 1, to the value reloaded

Set CH0IEN0, CH1IEN0 field of INTEN Register to 1

Set CH0TEN0, CH1TEN0, field of CHEN Register to 1

TMR ISR :

Set CH0INTF0, CH1INTF0, field of INTSTAT to 1 to clear interrupt flag

PWM : TMR channel 0 PWM, channel 1 PWM

Set CHMODE field of CH0CTRL Register to 4 (PWM mode)

Set CHMODE field of CH1CTRL Register to 4 (PWM mode)

Set CH0RLx, x = 0, 1, to channel 0 PWM low period

Set CH0RLx, x = 2, 3, to channel 0 PWM high period

Set CH1RLx, x = 0, 1, to channel 1 PWM low period

Set CH1RLx, x = 2, 3, to channel 1 PWM high period

Set CH0PEN, CH1PEN, field of CHEN Register to 1

Event counting with capture (software assist)

TMR Setting :

Set CHMODE field of CH0CTRL Register to 1 (32-bit timer mode)

Set CH0RLx, x = 0, 1, 2, 3, to the value 0xFF

EINT Setting :

Set corresponding EINT pin to “Rising and Falling trigger mode”

EINT ISR : (Rising Edge Triggered)

Set CH0TEN0 field of CHEN Register to 1

EINT ISR : (Falling Edge Triggered)

*1Read CH0CNTx, x = 0, 1, 2, 3, for 32-bit capture counter value

Set CH0TEN0 field of CHEN Register to 0

*1 User must read counter value first and then disable CH0TEN0 flag.

6.12 STN LCD Controller (LCM)

6.12.1 Overview

The FE81 series incorporates a STN LCD driver with the display requirements in various applications. This LCD driver has 4-COM and 20-SEG signals with several configurations providing direct driving and easy to use of interfacing to wide range of STN LCDs. Below lists the configuration of the LCD driver supported by FE81 series.

COM#	SEG#	Refresh Rate	Bias Type	Duty	Bias
4	20	60/120 (Hz)	R	1/4	1/3 or 1/2

Figure 6.12-1 illustrates R-type configurations with 1/3 and 1/2 bias type. User can choose one of the configurations based on the LCD selected.

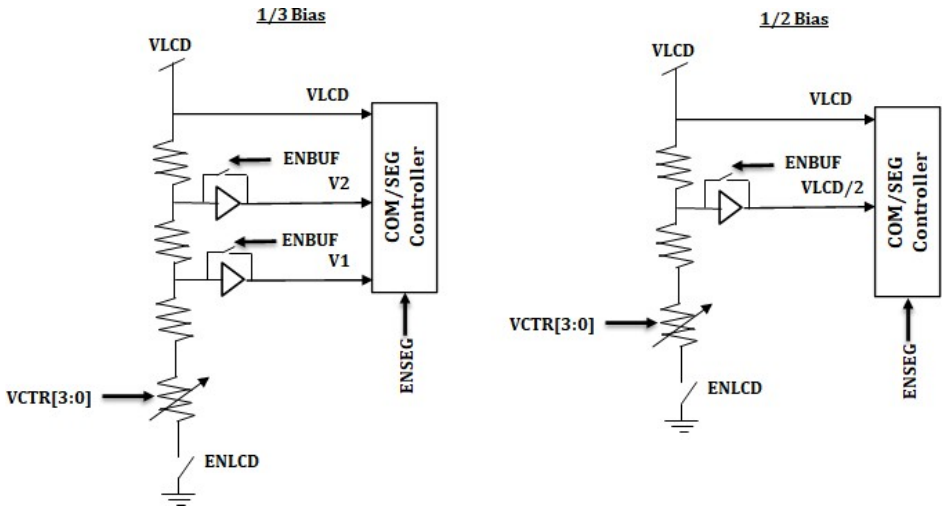


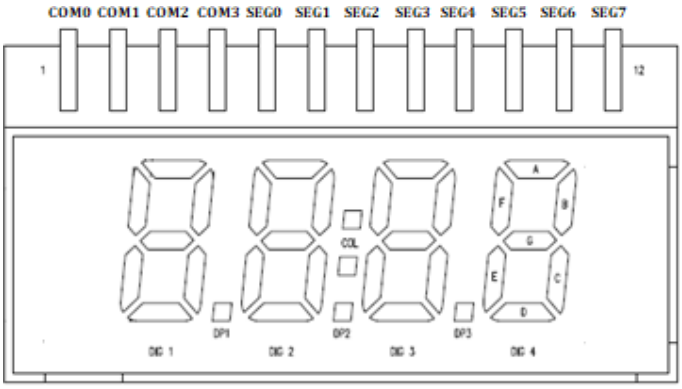
Figure 6.12-1 R-Type 1/3 or 1/2 Bias Configuration

While using the LCD driver, a frame buffer must be maintained for the data to be displayed on the LCD. While setting the corresponding bit field of the frame buffer, the related COM and SEG signal are selected to light on the corresponding pixel of the LCD. The FE81 series has dedicated frame buffer memory without having to be shared with application data, which decreasing the available size of SRAM memory. Figure 6.12-2 depicts the memory map of the frame buffer. The base address of LCD driver is 0xF200.

	bit3	bit2	bit1	bit0	
LCM_BASE + 0x10					SEQ0
LCM_BASE + 0x11					SEQ1
LCM_BASE + 0x12					SEQ2
LCM_BASE + 0x13					SEQ3
.					.
.					.
.					.
LCM_BASE + 0x22					SEQ18
LCM_BASE + 0x23					SEQ19
	COM3	COM2	COM1	COM0	

Figure 6.12-2 LCD Frame Buffer Memory Map

Figure 6.12-3 illustrates an example for lighting on the LCD with frame buffer setting. This LCD is controlled by 4 COMs (COM0 ~ COM4) and 8 SEGs (SEG0 ~ SEG7). If 36.75 is displayed on the LCD, the corresponding COM and SEG signals should be selected as the following table shows.



PIN	1	2	3	4	5	6	7	8	9	10	11	12
COM0	COM0	/	/	/	1D	DP1	2D	DP2	3D	DP3	4D	COL
COM1	/	COM1	/	/	1E	1C	2E	2C	3E	3C	4E	4C
COM2	/	/	COM2	/	1G	1B	2G	2B	3G	3B	4G	4B
COM3	/	/	/	COM3	1F	1A	2F	2A	3F	3A	4F	4A

Figure 6.12-3 LCD Example of Frame Buffer

Segment	DIG1	DIG2	DIG3	DIG4
A	1A	2A	3A	4A
B	1B	-	3B	-
C	1C	2C	3C	4C
D	1D	2D		4D
E	-	2E	-	-
F	-	2F	3F	4F
G	1G	2G	-	4G
DPx	-	DP2	-	-

As the table indicated above, the frame buffer should be set as follows. With the setting, 36.75 is showed on LCD.

		COM0	COM1	COM2	COM3
DIG1	SEG0	0	0	1 (1G)	0
	SEG1	0	1 (1C)	1 (1B)	1 (1A)
DIG2	SEG2	1 (2D)	1 (2E)	1 (2G)	1 (2F)
	SEG3	1 (DP2)	1 (2C)	0	1 (2A)
DIG3	SEG4	0	0	0	1 (3F)
	SEG5	0	1 (3C)	1 (3B)	1 (3A)
DIG4	SEG6	1 (4D)	0	1 (4G)	1 (4F)
	SEG7	0	1 (4C)	0	1 (4A)

6.12.2 Register Map and Description

Base Address (LCM_BA) : 0xF200				
Register	**Offset**	**RW**	**Description**	**Reset Value**
LCMCTRL0	LCM_BA+0x00	R/W	LCD Controller Control 0 Register	0x00
LCMCTRL1	LCM_BA+0x01	R/W	LCD Controller Control 1 Register	0x00
SEQBUF00	LCM_BA+0x10	R/W	LCD Controller Frame Buffer 0 Register	0x00
SEQBUF01	LCM_BA+0x11	R/W	LCD Controller Frame Buffer 1 Register	0x00
SEQBUF02	LCM_BA+0x12	R/W	LCD Controller Frame Buffer 2 Register	0x00
SEQBUF03	LCM_BA+0x13	R/W	LCD Controller Frame Buffer 3 Register	0x00
SEQBUF04	LCM_BA+0x14	R/W	LCD Controller Frame Buffer 4 Register	0x00
SEQBUF05	LCM_BA+0x15	R/W	LCD Controller Frame Buffer 5 Register	0x00
SEQBUF06	LCM_BA+0x16	R/W	LCD Controller Frame Buffer 6 Register	0x00
SEQBUF07	LCM_BA+0x17	R/W	LCD Controller Frame Buffer 7 Register	0x00
SEQBUF08	LCM_BA+0x18	R/W	LCD Controller Frame Buffer 8 Register	0x00
SEQBUF09	LCM_BA+0x19	R/W	LCD Controller Frame Buffer 9 Register	0x00
SEQBUF10	LCM_BA+0x1A	R/W	LCD Controller Frame Buffer 10 Register	0x00
SEQBUF11	LCM_BA+0x1B	R/W	LCD Controller Frame Buffer 11 Register	0x00
SEQBUF12	LCM_BA+0x1C	R/W	LCD Controller Frame Buffer 12 Register	0x00
SEQBUF13	LCM_BA+0x1D	R/W	LCD Controller Frame Buffer 13 Register	0x00
SEQBUF14	LCM_BA+0x1E	R/W	LCD Controller Frame Buffer 14 Register	0x00
SEQBUF15	LCM_BA+0x1F	R/W	LCD Controller Frame Buffer 15 Register	0x00
SEQBUF16	LCM_BA+0x20	R/W	LCD Controller Frame Buffer 16 Register	0x00
SEQBUF17	LCM_BA+0x21	R/W	LCD Controller Frame Buffer 17 Register	0x00
SEQBUF18	LCM_BA+0x22	R/W	LCD Controller Frame Buffer 18 Register	0x00
SEQBUF19	LCM_BA+0x23	R/W	LCD Controller Frame Buffer 19 Register	0x00

LCD Controller Control 0 Register

Register	Offset	RW	Description	Reset Value
LCMCTRL0	LCM_BA+0x00	R/W	LCD Controller Control 0 Register	0x00

Bits	Flag	Description
[7]	Reserved	**Reserved**.
[6]	ENBUF	**LCD High Driving Strength Enable Flag** This bit is set by software, cleared by software. High driving strength is set by this flag. ENBUF = 0, Low driving strength. ENBUF = 1, High driving strength.
[5]	FREQ	**LCD Frame Rate Selection Flag** This bit is set by software, cleared by software. FREQ = 0, 60Hz. FREQ = 1, 120Hz.
[4]	LEVEL	**LCD Bias Selection Flag** This bit is set by software, cleared by software. LEVEL = 0, 1/3 bias. LEVEL = 1, ½ bias.
[3:0]	VCTR	**LCD DC Voltage TRIM Flag** This bit is set by software, cleared by software. VCTR = 0, TRIM value 0. VCTR = 1, TRIM value 1. VCTR = 2, TRIM value 2. VCTR = 3, TRIM value 3. VCTR = 4, TRIM value 4. VCTR = 5, TRIM value 5. VCTR = 6, TRIM value 6. VCTR = 7, TRIM value 7.

LCD Controller Control 1 Register

Register	Offset	RW	Description	Reset Value
LCMCTRL1	LCM_BA+0x01	R/W	LCD Controller Control 1 Register	0x00

Bits	Flag	Description
[7:1]	Reserved	**Reserved**.
[0]	LCMEN	**LCD Enable Flag** This bit is set by software, cleared by software. LCMEN = 0, LCD driver disabled. LCMEN = 1, LCD driver enabled.

LCD Controller Frame Buffer Register

Register	Offset	RW	Description	Reset Value
SEQBUF00	LCM_BA+0x10	R/W	LCD Controller Frame Buffer 0 Register	0x00
SEQBUF01	LCM_BA+0x11	R/W	LCD Controller Frame Buffer 1 Register	0x00
SEQBUF02	LCM_BA+0x12	R/W	LCD Controller Frame Buffer 2 Register	0x00
SEQBUF03	LCM_BA+0x13	R/W	LCD Controller Frame Buffer 3 Register	0x00
SEQBUF04	LCM_BA+0x14	R/W	LCD Controller Frame Buffer 4 Register	0x00
SEQBUF05	LCM_BA+0x15	R/W	LCD Controller Frame Buffer 5 Register	0x00
SEQBUF06	LCM_BA+0x16	R/W	LCD Controller Frame Buffer 6 Register	0x00
SEQBUF07	LCM_BA+0x17	R/W	LCD Controller Frame Buffer 7 Register	0x00
SEQBUF08	LCM_BA+0x18	R/W	LCD Controller Frame Buffer 8 Register	0x00
SEQBUF09	LCM_BA+0x19	R/W	LCD Controller Frame Buffer 9 Register	0x00
SEQBUF10	LCM_BA+0x1A	R/W	LCD Controller Frame Buffer 10 Register	0x00
SEQBUF11	LCM_BA+0x1B	R/W	LCD Controller Frame Buffer 11 Register	0x00
SEQBUF12	LCM_BA+0x1C	R/W	LCD Controller Frame Buffer 12 Register	0x00
SEQBUF13	LCM_BA+0x1D	R/W	LCD Controller Frame Buffer 13 Register	0x00
SEQBUF14	LCM_BA+0x1E	R/W	LCD Controller Frame Buffer 14 Register	0x00
SEQBUF15	LCM_BA+0x1F	R/W	LCD Controller Frame Buffer 15 Register	0x00
SEQBUF16	LCM_BA+0x20	R/W	LCD Controller Frame Buffer 16 Register	0x00
SEQBUF17	LCM_BA+0x21	R/W	LCD Controller Frame Buffer 17 Register	0x00
SEQBUF18	LCM_BA+0x22	R/W	LCD Controller Frame Buffer 18 Register	0x00
SEQBUF19	LCM_BA+0x23	R/W	LCD Controller Frame Buffer 19 Register	0x00

Bits	Flag	Description
[7:4]	Reserved	**Reserved.**
[3:0]	FRAMEBUFn n = 0 ~ 19	**Frame Buffer Field** Writing this field can control corresponding COM and SEG signals. Refer to 6.12.1 and external PCB circuit for controlling the LCD module.

6.12.3 Programming Model (Please refer to “Sample code project”)

LCD light on procedure

Step 1 :

Set corresponding ENBUF, FREQ, LEVEL, VCTR field of LCMCTRL0 Register.

Step 2 :

Reset frame buffer, SEQBUF00 ~ SEQBUF19

Step 3 :

Set LCMEN field of LCMCTRL1 Register

Step 4 :

Based on the circuit of PCBA, lighting on the LCD by setting the bit field of frame buffer.

LCD light off procedure

Step 1 :

Disable LCMEN field of LCMCTRL1 Register

6.13 I2C Interface Controller (I2C)

6.13.1 Overview

The FE81 series supports an I2C controller with master or slave device.

Features of I2C Controller

- Support Standard-mode (100 Kb/s), Fast-mode (400 Kb/s) and Fast-mode Plus (1 Mb/s) protocols*1*2
- Programmable Master or Slave mode
- Support 7-bit and 10-bit addressing mode
- Support general call address
- Auto clock stretching
- Programmable clock or data timing
- 8-byte hardware transmit or receive FIFOs

*1 Note that, while Fast-mode is selected, the I2C clock source must be at least 4MHz

*2 Note that, while Fast-mode Plus is selected, the I2C clock source must be at least 12MHz

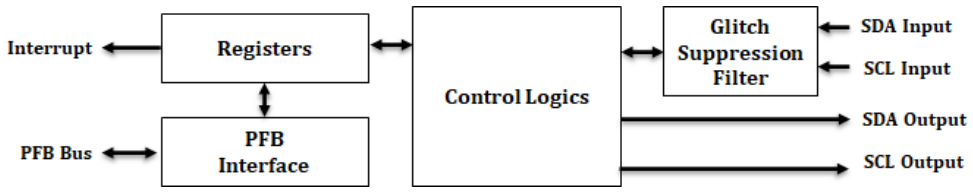


Figure 6.13-1 I2C Block Diagram

I2C Master

As an I2C master, the controller provides an efficient way to initiate I2C transactions. Every transaction can be delineated by four phases: Start, Address, Data and Stop. At the Start phase, a START condition is generated. At the Address phase, an address is sent. At the Data phase, one or more data bytes are transferred. At the Stop phase, a STOP condition is generated. The existence of each phase can be controlled independently.

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 General Call Address is a special address to address all slave devices on the I2C-bus. The I2C controller at the slave mode will respond with an ACK to the general call address and set the GENCALL field of the INTSTAT1 Register.

Auto Clock Stretch

The I2C controller can automatically pause bus transactions by stretching clocks on the I2C-bus when the software is not ready for the next byte of data or when the FIFO is full. Auto Clock Stretch is supported at both the master mode and the slave mode.

Auto-ACK

With Auto-ACK, the I2C controller automatically generates proper acknowledgements for each byte received. Every received byte will be responded with an ACK, except for the last byte, which should be responded with a NACK according to the I2C-bus protocol. On the other hand, if the software needs to determine each byte's acknowledgement status, Auto-ACK can be turned off by enabling the Byte Receive Interrupt.

6.13.2 Register Map and Description

Base Address (I2C_BA) : 0xEE00				
Register	**Offset**	**RW**	**Description**	**Reset Value**
HWCFG	I2CS_BA+0x00	R	I2C Hardware FIFO Configuration Register	0x00
INTEN0	I2CS_BA+0x01	R/W	I2C Interrupt Enable 0 Register	0x00
INTEN1	I2CS_BA+0x02	R/W	I2C Interrupt Enable 1 Register	0x00
INTSTAT0	I2CS_BA+0x03	R/W	I2C Interrupt Status 0 Register	0x00
INTSTAT1	I2CS_BA+0x04	R/W	I2C Interrupt Status 1 Register	0x00
SLVADR0	I2CS_BA+0x05	R/W	I2C Slave Address 0 Register	0x00
SLVADR1	I2CS_BA+0x06	R/W	I2C Slave Address 1 Register	0x00
I2CDAT	I2CS_BA+0x07	R/W	I2C Data Register	0x00
DATCNT	I2CS_BA+0x08	R/W	I2C Data Count Register	0x00
I2CCTRL	I2CS_BA+0x09	R/W	I2C Control Register	0x00
I2CCMD	I2CS_BA+0x0A	R/W	I2C Command Register	0x00
I2CSETUP0	I2CS_BA+0x0B	R/W	I2C Setup 0 Register	0x00
I2CSETUP1	I2CS_BA+0x0C	R/W	I2C Setup 1 Register	0x00
I2CSETUP2	I2CS_BA+0x0D	R/W	I2C Setup 2 Register	0x00
I2CSETUP3	I2CS_BA+0x0E	R/W	I2C Setup 3 Register	0x00

I2C Hardware FIFO Configuration Register

Register	Offset	RW	Description	Reset Value
HWCFG	I2CS_BA+0x00	R	I2C Hardware FIFO Configuration Register	0x00

Bits	Flag	Description
[7:2]	Reserved	**Reserved**.
[1:0]	FIFOSIZE	**FIFO Size** FIFOSIZE = 2, FIFO size is 8-byte.

I2C Interrupt Enable 0 Register

Register	Offset	RW	Description	Reset Value
INTEN0	I2CS_BA+0x01	R/W	I2C Interrupt Enable 0 Register	0x00

Bits	Flag	Description
[7]	BTRN	**I2C Byte Transmit Interrupt Enable Flag** This bit is set by software, cleared by software. BTRN = 0, Idle. BTRN = 1, I2C byte transmit interrupt enabled.
[6]	START	**I2C Start Condition Interrupt Enable Flag** This bit is set by software, cleared by software. START = 0, Idle. START = 1, I2C start condition interrupt enabled.
[5]	STOP	**I2C Stop Condition Interrupt Enable Flag** This bit is set by software, cleared by software. STOP = 0, Idle. STOP = 1, I2C stop condition interrupt enabled.
[4]	ARBLOS	**I2C Arbitration Lose Interrupt Enable Flag** This bit is set by software, cleared by software. Master : Interrupts when the controller loses the arbitration. Slave : Arbitration lose interrupt is not available at the slave mode. ARBLOS = 0, Idle. ARBLOS = 1, I2C arbitration lose interrupt enabled.
[3]	ADRHIT	**I2C Address Hit Interrupt Enable Flag** This bit is set by software, cleared by software. Master : Interrupts when the addressed slave returned an ACK. Slave : Interrupts when the controller is addressed. ADRHIT = 0, Idle. ADRHIT = 1, I2C address hit interrupt enabled.
[2]	FIFOHAF	**I2C FIFO Half Interrupt Enable Flag** This bit is set by software, cleared by software. 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. FIFOHAF = 0, Idle. FIFOHAF = 1, I2C FIFO half interrupt enabled.
[1]	FIFOFUL	**I2C FIFO Full Interrupt Enable Flag** This bit is set by software, cleared by software. FIFOFUL = 0, Idle. FIFOFUL = 1, I2C FIFO full interrupt enabled.
[0]	FIFOEMP	**I2C FIFO Empty Interrupt Enable Flag** This bit is set by software, cleared by software. FIFOEMP = 0, Idle. FIFOEMP = 1, I2C FIFO empty interrupt enabled.

I2C Interrupt Enable 1 Register

Register	Offset	RW	Description	Reset Value
INTEN1	I2CS_BA+0x02	R/W	I2C Interrupt Enable 1 Register	0x00

Bits	Flag	Description
[7:2]	Reserved	**Reserved**.
[1]	CMPL	**I2C Completion Interrupt Enable Flag** This bit is set by software, cleared by software. Master : Interrupts when a transaction is issued without losing the bus arbitration. Slave : Interrupts when a transaction addressing the controller completes. CMPL = 0, Idle. CMPL = 1, I2C completion interrupt enabled.
[0]	BRCV	**I2C Byte Receive Interrupt Enable Flag** This bit is set by software, cleared by software. Auto-ACK will be disabled if this interrupt is enabled, that is, the software needs to ACK/NACK the received byte manually. BRCV = 0, Idle. BRCV = 1, I2C byte receive interrupt enabled.

I2C Interrupt Status 0 Register

Register	Offset	RW	Description	Reset Value
INTSTAT0	I2CS_BA+0x03	R/W	I2C Interrupt Status 0 Register	0x00

Bits	Flag	Description
[7]	BTRNF	**I2C Byte Transmit Interrupt Flag** This bit is read by software, cleared by software (Write 1 to clear). BTRNF = 0, Idle. BTRNF = 1, I2C byte transmit interrupt triggered.
[6]	STARTF	**I2C Start Condition Interrupt Flag** This bit is read by software, cleared by software (Write 1 to clear). STARTF = 0, Idle. STARTF = 1, I2C start condition interrupt triggered.
[5]	STOPF	**I2C Stop Condition Interrupt Flag** This bit is read by software, cleared by software (Write 1 to clear). STOPF = 0, Idle. STOPF = 1, I2C stop condition interrupt triggered.
[4]	ARBLOSF	**I2C Arbitration Lose Interrupt Flag** This bit is read by software, cleared by software (Write 1 to clear). ARBLOSF = 0, Idle. ARBLOSF = 1, I2C arbitration lose interrupt triggered.
[3]	ADRHITF	**I2C Address Hit Interrupt Flag** This bit is read by software, cleared by software (Write 1 to clear). ADRHITF = 0, Idle. ADRHITF = 1, I2C address hit interrupt triggered.
[2]	FIFOHAFF	**I2C FIFO Half Interrupt Flag** This bit is read only. FIFOHAFF = 0, Idle. FIFOHAFF = 1, I2C FIFO half interrupt triggered.
[1]	FIFOFULF	**I2C FIFO Full Interrupt Flag** This bit is read only. FIFOFULF = 0, Idle. FIFOFULF = 1, I2C FIFO full interrupt triggered.
[0]	FIFOEMPF	**I2C FIFO Empty Interrupt Flag** This bit is read only. FIFOEMPF = 0, Idle. FIFOEMPF = 1, I2C FIFO empty interrupt triggered.

I2C Interrupt Status 1 Register

Register	Offset	RW	Description	Reset Value
INTSTAT1	I2CS_BA+0x04	R/W	I2C Interrupt Status 1 Register	0x00

Bits	Flag	Description
[7]	Reserved	**Reserved**.
[6]	LINESDA	**I2C SDA Line Status Flag** This bit is read only. LINESDA = 0, Low. LINESDA = 1, High.
[5]	LINESCL	**I2C SCL Line Status Flag** This bit is read only. LINESCL = 0, Low. LINESCL = 1, High.
[4]	GENCALL	**I2C General Call Flag** This bit is read only. GENCALL = 0, Not general call. GENCALL = 1, General call.
[3]	BUSBUSY	**I2C Bus Busy Flag** This bit is read only. The bus is busy when a START condition is on I2C bus and it ends when a STOP condition is seen on bus. BUSBUSY = 0, Not busy. BUSBUSY = 1, Busy.
[2]	ACK	**I2C ACK Flag** This bit is read only. ACK = 0, NACK. ACK = 1, ACK.
[1]	CMPLF	**I2C Completion Interrupt Flag** This bit is read by software, cleared by software (Write 1 to clear). Master : Indicate a transaction is issued without losing arbitration. Slave : Indicate a transaction addressing the controller completes. This bit must be cleared to receive the next transaction; otherwise, the next incoming transaction will be blocked. CMPLF = 0, Idle. CMPLF = 1, I2C completion interrupt triggered.
[0]	BRCVF	**I2C Byte Receive Interrupt Flag** This bit is read by software, cleared by software (Write 1 to clear). BYTERECVF = 0, Idle. BYTERECVF = 1, I2C byte receive interrupt triggered.

I2C Slave Address 0 Register

Register	Offset	RW	Description	Reset Value
SLVADR0	I2CS_BA+0x05	R/W	I2C Slave Address 0 Register	0x00

Bits	Flag	Description
[7:0]	SLVADRL	**I2C Slave Address Low Register** This bit is set by software, cleared by software. For 7-bit addressing mode, the most significant 3-bit are ignored.

I2C Slave Address 1 Register

Register	Offset	RW	Description	Reset Value
SLVADR1	I2CS_BA+0x06	R/W	I2C Slave Address 1 Register	0x00

Bits	Flag	Description
[7:0]	SLVADRH	**I2C Slave Address High Register** This bit is set by software, cleared by software. For 7-bit addressing mode, the most significant 3-bit are ignored.

I2C Data Register

Register	Offset	RW	Description	Reset Value
I2CDAT	I2CS_BA+0x07	R/W	I2C Data Register	0x00

Bits	Flag	Description
[7:0]	I2CDAT	**I2C Data Register** This bit is set by software. Write this register to put one byte to the FIFO. Read this register to get one byte from the FIFO.

I2C Data Count Register

Register	Offset	RW	Description	Reset Value
DATCNT	I2CS_BA+0x08	R/W	I2C Data Count Register	0x00

Bits	Flag	Description
[7:0]	DATCNT	**I2C Data Count Register** This bit is set by software. 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 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.

I2C Control Register

Register	Offset	RW	Description	Reset Value
I2CCTRL	I2CS_BA+0x09	R/W	I2C Control Register	0x00

Bits	Flag	Description
[7:5]	Reserved	**Reserved**.
[4]	PHSTART	**I2C Send Start Condition Enable Flag (Master Mode Only)** This bit is set by software, cleared by hardware. PHSTART = 0, Idle. PHSTART = 1, I2C send start condition.
[3]	PHADR	**I2C Send Address After Start Condition Enable Flag (Master Mode Only)** This bit is set by software, cleared by hardware. PHADR = 0, Idle. PHADR = 1, I2C send address phase after start condition.
[2]	PHDAT	**I2C Send Data After Address Phase Enable Flag (Master Mode Only)** This bit is set by software, cleared by hardware. PHDAT = 0, Idle. PHDAT = 1, I2C send data phase after address phase.
[1]	PHSTOP	**I2C Send Stop Condition Enable Flag (Master Mode Only)** This bit is set by software, cleared by hardware. PHSTOP = 0, Idle. PHSTOP = 1, I2C send stop condition.
[0]	DIR	**I2C Transaction Direction Flag** This bit is set by software, cleared by software. Master : DIR = 0, Transmitter. DIR = 1, Receiver. Slave : DIR = 0, Receiver. DIR = 1, Transmitter.

I2C Command Register

Register	Offset	RW	Description	Reset Value
I2CCMD	I2CS_BA+0x0A	R/W	I2C Command Register	0x00

Bits	Flag	Description
[7:0]	I2CCMD	**I2C Command Register** This bit is set by software. 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 I2C controller (abort current transaction, set the SDA and SCL line to the open-drain mode, clear the Status Register, Interrupt Enable Register and empty 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.

I2C Setup 0 Register

Register	Offset	RW	Description	Reset Value
I2CSETUP0	I2CS_BA+0x0B	R/W	I2C Setup 0 Register	0x00

Bits	Flag	Description
[7:3]	Reserved	**Reserved**.
[2]	MASTER	**I2C Master/Slave Selection Flag** This bit is set by software, cleared by hardware. MASTER = 0, Master mode. MASTER = 1, Slave mode.
[1]	ADDRING	**I2C Addressing Mode Flag** This bit is set by software, cleared by hardware. ADDRING = 0, 7-bit addressing mode. ADDRING = 1, 10-bit addressing mode.
[0]	I2CEN	**I2C Enable Flag** This bit is set by software, cleared by software. I2CEN = 0, I2C idle. I2CEN = 1, I2C enabled.

I2C Setup 205 Register

Register	Offset	RW	Description	Reset Value
I2CSETUP1	I2CS_BA+0x0C	R/W	I2C Setup 1 Register	0x00

Bits	Flag	Description
[7:0]	T_SCLHi	**I2C SCL High Period Register (Master Mode Only)** This bit is set by software, cleared by software. SCL HIGH period = *(4 + T_SP + T_SCLHi) * tpclk* The T_SCLHi value must be greater than T_SP and T_HDDAT values.

I2C Setup 206 Register

Register	Offset	RW	Description	Reset Value
I2CSETUP2	I2CS_BA+0x0D	R/W	I2C Setup 2 Register	0x00

Bits	Flag	Description
[7]	Reserved	**Reserved**.
[6:2]	T_HDDAT	**I2C Data Hold Time Register** This bit is set by software, cleared by software. T_HDDAT defines the data hold time after SCL goes LOW Hold time = *(4 + T_SP + T_HDDAT) * tpclk*
[1]	T_SCLRAT	**I2C SCL Low Period Register (Master Mode Only)** This bit is set by software, cleared by software. The LOW period of the generated SCL clock is defined by the combination of T_ SCLRAT and T_ SCLHi values. When T_ SCLRAT = 0, the LOW period is equal to HIGH period. When T_ SCLRAT = 1, the LOW period is roughly two times of HIGH period. SCL LOW period = *(4 + T_SP + T_SCLHi * ratio) * tpclk* T_SCLRAT = 0, LOW period is equal to HIGH period (*ratio* = 1). T_SCLRAT = 1, LOW period is roughly two times of HIGH period (*ratio* = 2).
[0]	T_SCLHi	**I2C SCL High Period Register (Master Mode Only)** This bit is set by software, cleared by software. SCL HIGH period = *(4 + T_SP + T_SCLHi) * tpclk* Note : The T_SCLHi value must be greater than T_SP and T_HDDAT values.

I2C Setup 207 Register

Register	Offset	RW	Description	Reset Value
I2CSETUP3	I2CS_BA+0x0E	R/W	I2C Setup 3 Register	0x00

Bits	Flag	Description
[7:3]	T_DATSETUPT	**I2C Data Setup Time Register** This bit is set by software, cleared by software. Setup time = *(4 + T_SP + T_DATSETUPT) * tpclk* tpclk = PCLK period
[2:0]	T_SP	**I2C Pulse Width of Spikes Register** This bit is set by software, cleared by software. T_SP defines the pulse width of spikes that must be suppressed by the input filter Pulse width = *T_SP * tpclk*

6.13.3 Programming Model (Please refer to "Sample code project")

Before enabling the I2C controller, user must setup the I2C-bus timing parameters by programming the setup register. As an I2C slave, the spike suppression width, the data setup time and the data hold time must be programmed properly according to the PFB clock frequency and the speed of the I2C-bus. As an I2C master, the I2C-bus clock frequency must be programmed as well.

The following descriptions show how to determine the setup register to meet the I2C-bus timing parameters. All the examples assume that the PFB clock frequency is 12MHz, i.e. the PFB clock period is 83ns. If the PFB clock frequency of your design is not 12MHz, please derive the register fields accordingly.

Spike Suppression Width

Table below shows the pulse width of spikes that must be suppressed by the input filter. For the Fast-mode and the Fast-mode Plus, spikes less than 50ns must be suppressed. i.e.

T_SP = 50ns/83ns = 0

Symbol	Standard		Fast		Fast-Plus	
	Min	Max	Min	Max	Min	Max
T_SP	-	-	0ns	50ns	0ns	50ns

Data Setup Time

Data setup time defines the time in which the SDA should be held steady before the SCL rising edge. Table below shows the timing parameters for the data setup time. The equation of data setup time is:

Symbol	Standard		Fast		Fast-Plus	
	Min	Max	Min	Max	Min	Max
T_DATSETUPT	250ns	-	100ns	-	50ns	-

Setup time = (4 + T_SP + T_ DATSETUPT) * tpclk

For the Standard-mode,

250ns = (4 + 0 + T_DATSETUPT) * 83ns

Then, T_DATSETUPT = 0

Data Hold Time

Data hold time defines the time in which the SDA should be held steady after the SCL falling edge. Table below shows the timing parameters for the data hold time. The equation of data hold time is:

Symbol	Standard		Fast		Fast-Plus	
	Min	Max	Min	Max	Min	Max
T_HDDAT	300ns	-	300ns	-	0	-

Hold time = (4 + T_SP + T_HDDAT) * tpclk

For the Standard-mode,

300ns = (4 + 0 + T_HDDAT) * 83ns

Then, T_HDDAT = 0

I2C Bus Clock Frequency

The I2C-bus clock frequency is specified by the tHIGH and tLOW parameters, which are shown in Table below and can be achieved through the T_SCLHi and T_SCLRAT fields of the setup register.

Symbol	Standard		Fast		Fast-Plus	
	Min	Max	Min	Max	Min	Max
tHIGH	4.0us	-	0.6us	-	0.26us	-
tLOW	4.7us	-	1.3us	-	0.5us	-

For the Standard-mode, the minimum requirements of tHIGH and tLOW are close, so T_ SCLRAT can be set to 0 (i.e. ratio = 1) to simplify the settings. The equations for the SCL periods are:

SCL HIGH period = (4 + T_SP + T_SCLHi) * tpclk >= 4000ns

SCL LOW period = (4 + T_SP + T_SCLHi * ratio) * tpclk >= 4700ns

Substitute 0 for T_SP, 1 for ratio and 83ns for tpclk, the equations become:

(4 + 0 + T_SCLHi) * 83ns >= 4000ns

(4 + 0 + T_SCLHi * 1) * 83ns >= 4700ns

Then,

T_SCLHi >= 53

For the Fast-mode, the minimum requirement of tLOW is about 2 times of tHIGH, so T_ SCLRAT can be set to 1 (i.e. ratio = 2). The equations for the SCL periods are:

The following examples show two bus clock setups :

SCL HIGH period = (4 + T_SP + T_SCLHi) * tpclk >= 600ns

SCL LOW period = (4 + T_SP + T_SCLHi * ratio) * tpclk >= 1300ns

Substitute 0 for T_SP, 2 for ratio and 83ns for tpclk, the equations become:

(4 + 0 + T_SCLHi) * 83ns >= 600ns

(4 + 0 + T_SCLHi * 2) * 83ns >= 1300ns

Then,

T_SCLHi >= 6

Master Transmit Mode

Step 1 : Set the setup register (I2CSETUP0 ~ I2CSETUP3)

Set Master = 1, I2CEN = 1, and related timing parameters.

Step 2 : Set data counter, direction and phase choices in the I2CCTRL Register and DATCNT Register

Set PHSTART = 1, PHADR = 1, PHDAT = 1, PHSTOP = 1, DIR = 0.

DATCNT = data counts in bytes.

Step 3 : Write I2C slave address SLVADR0 Register

Step 4 : Enable the completion interrupt and FIFO empty interrupt in the interrupt enable register

Set CMPL = 1 in INTEN1 Register.

Set FIFOEMP = 1 in INTEN0 Register.

Step 5: Write to the I2CCMD Register to issue the transaction

Set 0x01 in I2CCMD Register.

Step 6 : Wait for interrupts

1. FIFO empty interrupt : push data into the FIFO by writing the data to the I2CDAT Register until the FIFO becomes full. If all data are pushed into the FIFO, disable the FIFO Empty Interrupt. Otherwise, repeat Step 6.
2. Completion interrupt : write 1 to the CMPLF field of the INTSTAT1 Register to clear the completion status and go to Step 7.

Step 7 : Disable all interrupts and check the DATCNT field of the DATCNT Register to know if all data are successfully transmitted

Master Receive Mode

Step 1 : Set the setup register (I2CSETUP0 ~ I2CSETUP3)

Set Master = 1, I2CEN = 1, and related timing parameters.

Step 2 : Set data counter, direction and phase choices in the I2CCTRL Register and DATCNT Register

Set PHSTART = 1, PHADR = 1, PHDAT = 1, PHSTOP = 1, DIR = 1.

DATCNT = data counts in bytes.

Step 3 : Write I2C slave address SLVADR0 Register

Step 4 : Enable the completion interrupt and FIFO full interrupt in the interrupt enable register

Set CMPL = 1 in INTEN1 Register.

Set FIFOFUL = 1 in INTEN0 Register.

Step 5: Write to the I2CCMD Register to issue the transaction

Set 0x01 in I2CCMD Register.

Step 6 : Wait for interrupts

1. FIFO full interrupt (FIFOFULF) : get data from the FIFO by reading the I2CDAT Register until the FIFO becomes empty; repeat Step 6.
2. Completion interrupt (CMPLF) : get all of the remaining data from the FIFO, write 1 to the CMPLF field of the INTSTAT1 Register to clear the completion status and go to Step 7.

Step 7 : Disable all interrupts and check the DATCNT field of the DATCNT Register to know if all data are successfully received

Slave Transaction Mode

Step 1 : Set the address SLVADR0 Register to the slave controller's I2C-bus address

Step 2 : Set the setup register (I2CSETUP0 ~ I2CSETUP3)

Set Master = 0, I2CEN = 1, and related timing parameters.

Step 3 : Enable the address hit interrupt and completion interrupt in the interrupt enable register

Set ADRHIT = 1 in INTEN0 Register.

Set CMPL = 1 in INTEN1 Register.

Step 4 : Wait for ADRHIT interrupt

1. Receiver : enable FIFOFUL interrupt and go to Step 5.
   Set FIFOFUL = 1 in INTEN0 Register.
2. Transmitter : enable FIFOEMP interrupt and go to Step 6.
   Set FIFOEMP = 1 in INTEN0 Register.

Step 5 : Receiver

1. Wait for FIFO full interrupt (FIFOFULF) until the completion interrupt (CMPLF) asserted.
2. If the FIFO full interrupt (FIFOFULF) asserts : get data from the FIFO by reading the I2CDAT Register until the FIFO becomes empty, then go to (1.)

Step 6 : Transmitter

1. Wait for the FIFO empty (FIFOEMPF) until the completion interrupt (CMPLF) asserted.
2. If the FIFO empty (FIFOEMPF) asserts : push data to the FIFO by writing data to the I2CDAT Register until the FIFO becomes full, then go to (1.)

Step 7 : Whether Step 5 or Step 6, if the completion interrupt (CMPLF) asserts, clear the FIFO and go to Step 8.

Step 8 : Check the DATCNT of the DATCNT Register to know how many data are transferred. Clear the completion interrupt flag (CMPLF) of the INTSTAT1 Register when the software is ready to receive the next transaction

6.14 UART Controller (UART)

6.14.1 Overview

Features of UART Controller

- Over-sampling frequency is programmable (even multiples ranging from 8x to 32x)
- Programmable sequence compatible with the 16C550D UART
  - Support 5 to 8 bits per character
  - Support 1, 1.5, 2 STOP bits
  - Support even, odd and stick parity bits
  - Support programmable baud rate
  - Support modem control interface
  - Support complete status reporting capabilities
  - Support line breaks, parity errors, framing errors, and data overrun detection
- 16-byte hardware transmit / receive FIFOs

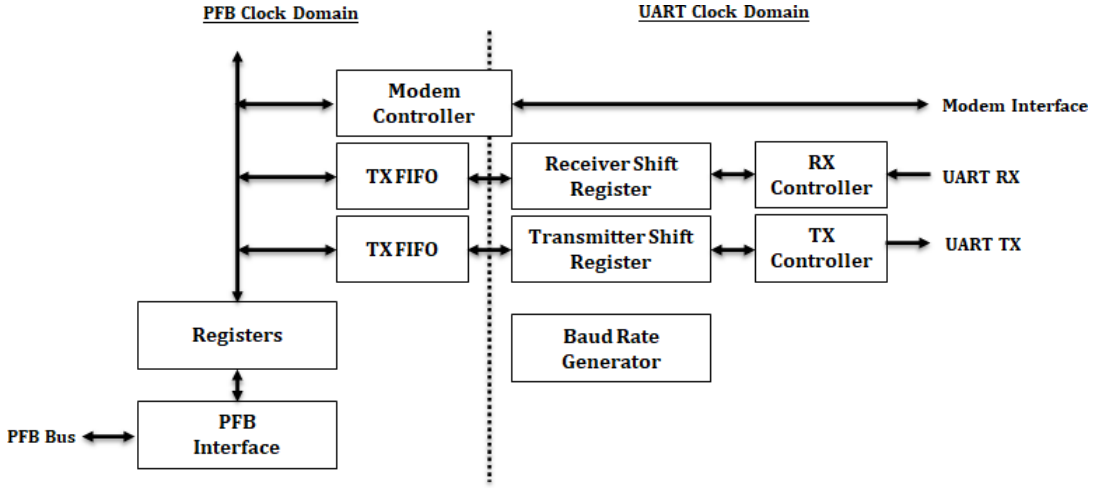


Figure 6.14-1 UART Block Diagram

The UART controller is a serial communication controller which provides asynchronous serial interface for a peripheral device or a modem. The controller comprises a transmitter, a receiver, a Baud Rate Generator, a Modem Controller, a Register File and an PFB interface, as shown in Figure 6.14-1.

Transmitter Mode

The transmitter comprises a transmitter FIFO (TX FIFO), a transmitter shift register (TSR), and a transmitter controller (TX controller). The TX FIFO holds data to be transferred through the serial interface. The TX FIFO can store up to 16 characters depending on hardware configurations. The TSR reads a character from the TX FIFO for the next transmission. The TSR functions as a parallel-to-serial

data converter, converting the outgoing character to serial bit streams. For each character transmission, the TX Controller generates a START bit, an optional parity bit, and some number of STOP bits. The generation of parity bit and STOP bit can be programmed by the line control register. The TX FIFO is by default a one entry buffer called transmitter holding register (THR). It needs to be enabled to work in multi-entry FIFO mode (FIFOE in the FIFO control register).

Receiver Mode

The receiver comprises a receiver FIFO (RX FIFO), receiver shift register (RSR), and a receiver controller (RX controller). The RX controller uses the oversampling clock generated by baud rate generator to perform sampling at the center of each bit transmission. The received bits are shifted into the RSR for serial-to-parallel data conversion and the received character is stored into the RX FIFO. The RX FIFO is by default a one entry buffer called the receiver buffer register (RBR). It needs to be enabled to work in multi-entry FIFO mode (FIFOE in the FIFO control register) with up to 16 characters depending on hardware configurations. The RX controller also detects some error conditions for each data transmission including parity error, framing error, data overrun, or line break.

Baud Rate Generator

The baud rate generator takes the UART clock as the source clock and divides it by a divisor. The divisor value is 16-bit in size and stored in two separate programming registers, each holding an 8-bit value. The most significant byte (MSB) is held in the divisor latch MSB (DLM) register and the least significant byte (LSB) is held in the divisor latch LSB (DLL) register.

The ratio of the sampling clock frequency to the baud rate is the oversampling ratio, which is stored in the over sample rate control register. The default value for OSCR is 16. This is typically good enough and do not need further adjustment. The formula for the divisor value is as follows :

The divisor value = frequency of UART / (desired baud rate × OSCR)

The oversampling clock is used by the RX controller to detect the leading edge of the START bit and to sample the data at the center of each bit transfer. Assume that the oversampling ratio is 16. The RX controller uses the oversampling clock to operate a counter, which starts counting from 1 to 16 after a falling edge of SIN (beginning of a START bit) is detected. The RX controller samples the value of SIN as a data bit when the counter value is 8. The counter will be reset to 1 after its value reaches 16 for sampling the next bit. The process repeats until the STOP bit is received. The oversampling clock is also used by the TX controller to generate its output data stream.

The oversampling technique provides better tolerance of clock variation. Suppose that :

- T is the period of one bit transmission as perceived by the UART Rx controller

- $T_{transmitter}$ is the period of one bit transmission of the transmitter
- N is the bit number for one frame of data – the START bit, data bits, parity bit (if any), and the STOP bit(s)

Then, the clock period tolerance for $T_{transmitter}$ is as follows :

$$\left(1 - \frac{\left(0.5 - \frac{1}{OSCR}\right)}{N}\right) \times T \leq T_{transmitter} \leq (N - 0.5)/(N - 1) \times T$$

Since T is the inverse of the baud rate, the actual baud rate generated by this controller in relation to the actual baud rate of the transmitter (the tolerance factor) can be within the range below :

$$\left(1 - \frac{\left(0.5 - \frac{1}{OSCR}\right)}{N}\right) \leq \frac{Actual\ Baud\ Rate}{Actual\ Transmitter\ Baud\ Rate} \leq (N - 0.5)/(N - 1)$$

If the character has one START bit, 8 data bits, one parity bit and one STOP bit, then N is 11 (1 + 8 + 1 + 1). The tolerance factor is from 0.9602 to 1.05. The table below shows clock tolerance factors as percentage of the actual transmitter baud rates for typical values of N and OSCR.

OSCR	N = 9	N = 10	N = 11	N = 12
8	95.83% - 106.25%	96.25% - 105.56%	96.59% - 105.00%	96.88% - 104.55%
16	95.14% - 106.25%	95.63% - 105.56%	96.02% - 105.00%	96.35% - 104.55%
32	94.79% - 106.25%	95.31% - 105.56%	95.74% - 105.00%	96.09% - 104.55%

Modem Controller

The modem controller provides the modem control function. Furthermore, an auto flow control function is provided to reduce the software management effort.

The flow control of UART can be achieved by the RTS/CTS handshaking. Without the flow control, overrun errors may occur when the data transmission rate exceeds the data consumption rate. The flow control guarantees that the data transmission will not proceed unless the receiver has enough space to accept the data.

The auto flow control function of UART controller comprises auto-RTS and auto-CTS. The former is for the incoming data while the latter is for the outgoing data.

With auto-RTS, the RTS output of the controller should be connected to the CTS input of the other end of the UART connection.

- ***n*** is the depth of the FIFO
- When the receiver FIFO ≥ ***n*** – 1, the RTS is de-asserted
- When the receiver FIFO < ***n*** – 1, the RTS is automatically asserted to request the other end of the UART connection to send more data

With auto-CTS, the CTS input of the controller is connected to the RTS output of the other end of the UART connection. The UART controller waits for the assertion of CTS before sending the next character. To stop the controller from transmitting the next character, CTS must be de-asserted before the STOP bit of the current character.

Loopback Mode

The UART controller provides a loopback mode for diagnostic testing without connecting an external device. When the loopback mode is enabled, the behavior of the controller is as follows :

- The output signals are disconnected from the TX controller and driven HIGH to avoid confusing the other end of the serial connection in case the connection exists
- The input signals are disconnected from the RX controller and ignored
- The TX controller output values originally intended for the TX output signals is routed internally to replace the input signal of RX for the RX controller, so every bit sent by the TX controller is looped back and received by the RX controller
- The modem controller output values which are intended for the signals are routed internally to replace the input signals for the modem controller

The value written to the transmitter holding register will be internally received by the receiver buffer register and the value written to the modem control register (bit3~bit0) will be routed to the modem status register (bit7~bit4). The entire bit transmission path is exercised in the loopback mode, only the input/output port is isolated from the diagnostic activity. The relationship between the bit fields in the modem control register and those in the modem status register are illustrated in Figure 6.14-2.

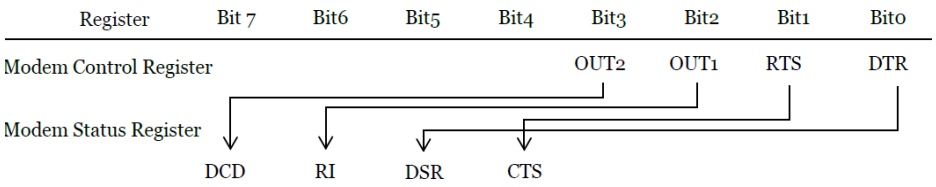


Figure 6.14-2 Relationship Between MCR and MSR in Loopback Mode

6.14.2 Register Map and Description

Base Address (UART_BA) : 0xF000				
Register	**Offset**	**RW**	**Description**	**Reset Value**
HWCFG	UART_BA+0x00	R/W	UART Hardware FIFO Configuration Register	0x00
OSCRCTRL	UART_BA+0x01	R/W	UART Over-Sampling Rate Control Register	0x00
DAT_DLL	UART_BA+0x02	R	DLAB = 0 : Receive Buffer Register (RBR)	0x00
		W	DLAB = 0 : Transmitter Holding Register (THR)	
		R/W	DLAB = 1 : UART Divisor Latch LSB Register	
IER_DLM	UART_BA+0x03	R/W	DLAB = 0 : UART Interrupt Enable Register	0x00
		R/W	DLAB = 1 : UART Divisor Latch MSB Register	
IIR_FCR	UART_BA+0x04	R	UART Interrupt Identification Register	0x00
		W	UART FIFO Control Register	
LINECTRL	UART_BA+0x05	R/W	UART Line Control Register	0x00
MODEMCTRL	UART_BA+0x06	R/W	UART Modem Control Register	0x00
LINESTAT	UART_BA+0x07	R/W	UART Line Status Register	0x00
MODEMSTAT	UART_BA+0x08	R/W	UART Modem Status Register	0x00

UART Hardware FIFO Configuration Register

Register	Offset	RW	Description	Reset Value
HWCFG	UART_BA+0x00	R/W	UART Hardware FIFO Configuration Register	0x00

Bits	Flag	Description
[7:2]	Reserved	**Reserved**.
[1:0]	FIFOSIZE	**FIFO Size** FIFOSIZE = 0, FIFO size is 16-byte.

UART Over-Sampling Rate Control Register

Register	Offset	RW	Description	Reset Value
OSCRCTRL	UART_BA+0x01	R/W	UART Over-Sampling Rate Control Register	0x00

Bits	Flag	Description
[7:5]	Reserved	Reserved.
[4:0]	OSCR	**UART Over-Sampling Rate Control Register** This bit is set by software, cleared by software. The value must be an even number; any odd value writes to this field will be converted to an even value. OSCR = **0**: The over-sample ratio is 32. OSCR ≤ 8: The over-sample ratio is 8. **8** < OSCR < **3**: The over sample ratio is OSCR.

UART Data or Divisor Latch LSB Register

Register	Offset	RW	Description	Reset Value
DAT_DLL	UART_BA+0x02	R	DLAB = 0 : Receive Buffer Register (RBR)	0x00
		W	DLAB = 0 : Transmitter Holding Register (THR)	
		R/W	DLAB = 1 : UART Divisor Latch LSB Register	

Bits	Flag	Description
[7:0]	RBR (Read Only : RX)	**DLAB = 0 (DLAB bit field in LINECTRL Register)** **UART Receive Buffer Register** UART receive data read port.

The RBR has two modes, the FIFO mode and the Buffer mode. Bit0 of the FIFO control register (FIFOE) controls the selection between these two modes. When FIFOE is 1 (FIFO mode), the RBR is a RXFIFO. The depth of RXFIFO is 16 characters. When FIFOE is 0 (Buffer mode), the RBR is just a byte buffer.

Bits	Flag	Description
[7:0]	THR (Write Only : TX)	**DLAB = 0 (DLAB bit field in LINECTRL Register)** **UART Transmitter Holding Register** UART transmit data write port.

The THR has two modes, the FIFO mode and the Buffer mode. Bit0 of the FIFO control register (FIFOE) controls the selection between these two modes. When FIFOE is 1 (FIFO mode), the THR is a TXFIFO. The depth of TXFIFO is 16 characters. When FIFOE is 0 (Buffer mode), the THR is a byte buffer.

Bits	Flag	Description
[7:0]	DLL	**DLAB = 1 (DLAB bit field in LINECTRL Register)** **UART Least Significant Byte of Divisor Latch Register** UART divisor latch LSB byte.

The divisor latch holds the divisor value for generating the sampling clock from the UART clock source. The size of the divisor latch is 16 bits (two bytes) and this register holds the least significant byte of the divisor latch. The valid value of the divisor latch should be between 1 and 65535 (2^{16}-1), inclusive.

UART Interrupt Enable or Divisor Latch MSB Register

UART Interrupt Enable or Divisor Latch MSB Register

Register	Offset	RW	Description	Reset Value
IER_DLM	UART_BA+0x03	R/W	DLAB = 0 : UART Interrupt Enable Register	0x00
		R/W	DLAB = 1 : UART Divisor Latch MSB Register	

Bits	Flag	Description
[7:4]	Reserved	**Reserved**.
[3]	EMSI	**DLAB = 0 (DLAB bit field in LINECTRL Register)** **UART Modem Status Interrupt Enable Flag** This bit is set by software, cleared by software. The interrupt asserts when the status of one of the following occurs : If the auto-cts mode is disabled (MCR bit4 (AFE) = 0), the status of modem_ctsn has been changed. If the auto-cts mode is enabled (MCR bit4 (AFE) = 1), modem_ctsn would be used to control the transmitter. EMSI = 0, Modem status interrupt disabled. EMSI = 1, Modem status interrupt enabled.
[2]	ELSI	**DLAB = 0 (DLAB bit field in LINECTRL Register)** **UART Receiver Line Status Interrupt Enable Flag** This bit is set by software, cleared by software. ELSI = 0, Receiver line status interrupt disabled. ELSI = 1, Receiver line status interrupt enabled.
[1]	ETHEI	**DLAB = 0 (DLAB bit field in LINECTRL Register)** **UART Transmitter Holding Register Enable Flag** This bit is set by software, cleared by software. ETHEI = 0, Transmitter holding register interrupt disabled. ETHEI = 1, Transmitter holding register interrupt enabled.
[0]	ERBI	**DLAB = 0 (DLAB bit field in LINECTRL Register)** **UART Receive Data Interrupt Enable Flag** This bit is set by software, cleared by software. ERBI = 0, Receive data interrupt disabled. ERBI = 1, Receive data interrupt enabled.

Bits	Flag	Description
[7:0]	DLM	**DLAB = 1 (DLAB bit field in LINECTRL Register)** **UART Most Significant Byte of Divisor Latch Register** UART divisor latch MSB byte.

The divisor latch holds the divisor value for generating the sampling clock from the UART clock source. The size of the divisor latch is 16 bits (two bytes) and this register holds the most significant byte of the divisor latch. The valid value of the divisor latch should be between 1 and 65535 (2^{16}-1), inclusive.

UART Interrupt Identification or FIFO Control Register

Register	Offset	RW	Description	Reset Value
IIR_FCR	UART_BA+0x04	R	UART Interrupt Identification Register	0x00
		W	UART FIFO Control Register	

Bits	Flag	Description
[7:6]	FIFOED	**While Read IIR_FCR** **UART FIFO Enabled Flag** This bit is read only. FIFOED = 0, UART FIFO is disabled. FIFOED = 3, UART FIFO is enabled (These two bits are 1 when bit 0 of the FIFO control register (FIFOE) is set to 1).
[5:4]	Reserved	**Reserved**.
[3:0]	INTRID	**While Read IIR_FCR** **UART Interrupt ID Flag** This bit is read only. INTRID = 0110, Receiver line status. ● Overrun errors, parity errors, framing errors, or line breaks ● Read the line status register (LSR) to clear the interrupt INTRID = 0100, Receive data available. ● If FIFOE is disabled, there is one received data available in the DAT. ● If FIFOE is enabled, the numbers of received data available reach the trigger level (RFIFOT). ● The interrupt signal will stay active until the number of data available becoming smaller than the trigger level. ● Read the receiver data register (RBR) to clear the interrupt INTRID = 1100, Character timeout. ● When FIFOE is enabled, and no character have been removed from or input to receive FIFO and there is at least one character in receive FIFO during the last four characters times. ● Read the receiver data register (RBR) to clear the interrupt INTRID = 0010, Transmitter holding register empty. ● If FIFOE is disabled, the 1-byte THR is empty. ● If FIFOE is enabled, the whole 16-byte transmit FIFO is empty. ● Write the transmitter holding register (THR) or read the interrupt identification register (IIR) to clear INTRID = 0000, Modem status. ● The Modem Status Register (MSR) bit[3:0] is not 0. ● One of the following events occurred: Clear To Send (CTS), Data Set Ready (DSR), Ring Indicator (RI), or Data Carrier Detect (DCD) Read the modem status register (MSR) to clear INTRID flag.

Note: When multiple events would trigger interrupts at the same time, the priority level determines the value for the INTRID field.

Bits	Flag	Description
[7:6]	RFIFOT	**While Write IIR_FCR** **UART Receiver FIFO Trigger Level Flag** This bit is write only, and cleared by software. RFIFOT = 0, Not empty. RFIFOT = 1, More than 3. RFIFOT = 2, More than 7. RFIFOT = 3, More than 13.
[5:4]	TFIFOT	**While Write IIR_FCR** **UART Transmitter FIFO Trigger Level Flag** This bit is write only, and cleared by software. TFIFOT = 0, Not full. TFIFOT = 1, Less than 12. TFIFOT = 2, Less than 8. TFIFOT = 3, Less than 4.
[3]	Reserved	**Reserved**.
[2]	TFIFORST	**While Write IIR_FCR** **UART Transmitter FIFO Reset Enable Flag** This bit is write only, and cleared by hardware. TFIFORST = 0, UART transmitter FIFO not reset. TFIFORST = 1, UART transmitter FIFO reset.
[1]	RFIFORST	**While Write IIR_FCR** **UART Receiver FIFO Reset Enable Flag** This bit is write only, and cleared by hardware. RFIFORST = 0, UART receiver FIFO not reset. RFIFORST = 1, UART receiver FIFO reset.
[0]	FIFOE	**While Write IIR_FCR** **UART FIFO Enable Flag** This bit is write only, and cleared by software. FIFOE = 0, UART FIFO disabled. FIFOE = 1, UART FIFO enabled.

UART Line Control Register

Register	Offset	RW	Description	Reset Value
LINECTRL	UART_BA+0x05	R/W	UART Line Control Register	0x00

Bits	Flag	Description
[7]	DLAB	**UART Divisor Latch Access Flag** This bit is set by software, cleared by software. DLAB = 0, Norma register is selected. DLAB = 1, Divisor latch register is selected.
[6]	BC	**UART Break Control Flag** This bit is set by software, cleared by software. BC = 0, Break control disabled. BC = 1, Break control enabled.
[5]	SPS	**UART Stick Parity Selection Flag** This bit is set by software, cleared by software. SPS = 0, Stick parity disabled. SPS = 1, Stick parity enabled.
[4]	EPS	**UART Even Parity Selection Flag** This bit is set by software, cleared by software. EPS = 0, Odd parity. EPS = 1, Even parity.
[3]	PEN	**UART Parity Bit Enable Flag** This bit is set by software, cleared by software. PEN SPS EPS 0 x x -> No Parity Bit 1 0 0 -> Odd Parity 1 0 1 -> Even Parity 1 1 0 -> Parity = 1 1 1 1 -> Parity = 0 PEN = 0, Parity bit disabled. PEN = 1, Parity bit enabled.
[2]	STB	**UART Number of Stop Bit Flag** This bit is set by software, cleared by software. STB = 0, 1-bit stop. STB = 1, WLS = 0 -> 1.5-bit stop. STB = 1, WLS = 1, 2, 3 -> 2-bit stop.
[1:0]	WLS	**UART Word Length Setting Flag** This bit is set by software, cleared by software. WLS = 0, 5-bit. WLS = 1, 6-bit. WLS = 2, 7-bit. WLS = 3, 8-bit.

UART Modem Control Register

Register	Offset	RW	Description	Reset Value
MODEMCTRL	UART_BA+0x06	R/W	UART Modem Control Register	0x00

Bits	Flag	Description
[7:6]	Reserved	**Reserved**.
[5]	AFCE	**UART Auto Flow Control Enable Flag** This bit is set by software, cleared by software. AFCE = 0, Auto flow control disabled. AFCE = 1, Auto flow control enabled.
[4]	LOOP	**UART Loopback Mode Enable Flag** This bit is set by software, cleared by software. LOOP = 0, Loopback mode disabled. LOOP = 1, Loopback mode enabled.
[3:2]	Reserved	Reserved.
[1]	RTS	**UART Modem Request to Send Flag** This bit is set by software, cleared by software. This bit controls the modrm_rtsn output. RTS = 0, Modem_rtsn output signal will be driven HIGH. RTS = 1, Modem_rtsn output signal will be driven LOW.
[0]	Reserved	**Reserved**.

UART Line Status Register

Register	Offset	RW	Description	Reset Value
LINESTAT	UART_BA+0x07	R/W	UART Line Status Register	0x00

Bits	Flag	Description
[7]	ERRF	**UART Error in RXFIFO Flag** This bit is set by hardware, and cleared when this register is read. In the FIFO mode, this bit is set when there is at least one parity error, framing error, or line break associated with data in the RXFIFO. ERRF = 0, Otherwise. ERRF = 1, At least one error in the RXFIFO.
[6]	TEMT	**UART Transmitter Empty Flag** This bit is set by hardware. TEMT = 0, Otherwise. TEMT = 1, THR (TXFIFO in the FIFO mode) and the transmitter shift register (TSR) are both empty.
[5]	THER	**UART Transmitter Holding Register Empty Flag** This bit is set by hardware. If the THRE interrupt is enabled, an interrupt is triggered when THRE becomes 1. THER = 0, Otherwise. THER = 1, THR (TXFIFO in the FIFO mode) is empty.
[4]	LBREAK	**UART Line Break Flag** This bit is set when the RX input signal was held LOW for longer than the time for a full-word transmission. A full-word transmission is the transmission of the START, data, parity, and STOP bits. It is cleared when this register is read. In the FIFO mode, this bit indicates the line break for the received data at the top of the RXFIFO. LBREAK = 0, Otherwise. LBREAK = 1, Line break.
[3]	FERR	**UART Frame Error Flag** This bit is set by hardware, and cleared when this register is read. This bit is set when the received STOP bit is not HIGH. In the FIFO mode, this bit indicates the framing error for the received data at the top of the RXFIFO. PERR = 0, Otherwise. PERR = 1, Frame error.
[2]	PERR	**UART Parity Error Flag** This bit is set by hardware, and cleared when this register is read. In the FIFO mode, this bit indicates the parity error for the received data at the top of the RXFIFO. PERR = 0, Otherwise. PERR = 1, Received parity is not matched.
[1]	OERR	**UART Overrun Error Flag** This bit is set by hardware. OERR = 0, Otherwise. OERR = 1, Data in the receiver data register is overrun.
[0]	DR	**UART Data Ready Flag** This bit is set by hardware, and cleared when all of the received data are read. This bit is set when there are incoming received data in the DAT register. DR = 0, Otherwise. DR = 1, Data in the receiver data register.

UART Modem Status Register

Register	Offset	RW	Description	Reset Value
MODEMSTAT	UART_BA+0x08	R/W	UART Modem Status Register	0x00

Bits	Flag	Description
[7:5]	Reserved	**Reserved**.
[4]	CTS	**UART Clear to Send Flag** This bit is set by hardware, and cleared when this register is read. CTS = 0, modem_ctsn input signal is HIGH. CTS = 1, modem_ctsn input signal is LOW.
[3:1]	Reserved	Reserved.
[0]	DCTS	**UART Modem Delta Clear to Send Flag** This bit is set by hardware, and cleared when this register is read This bit is set when the state of the modem_ctsn input signal has been changed since the last time this register is read. DCTS = 0, Otherwise. DCTS = 1, mode_ctsn has been changed.

6.14.3 Programming Model (Please refer to “Sample code project”)

UART Transmitter

Step 1 :

Set FIFOE in the FIFO control register (FCR) to enable the FIFO.

Step 2 :

Wait for FIFO empty by looping until the THRE bit in the line status register (LINESTAT) becomes 1.

Step 3 :

Write at most 16 bytes of data to THR.

Step 4 :

If there are more data to send, go to Step 2.

UART Receiver

Step 1 :

Set FIFOE in the FIFO control register (FCR) to enable the FIFO.

Step 2 :

Wait until the DR bit in the line status register (LINESTAT) becomes 1.

Step 3 :

Read one byte of data from the RBR.

Step 4 :

Go to Step 2 to read more data.

6.15 Serial Peripheral Interface Controller (SPI)

6.15.1 Overview

The FE81 series supports a Serial Peripheral Interface (SPI) controller which serves as a SPI master or a SPI slave. As a SPI master, the controller connects various SPI devices. As a SPI slave, the controller responds to the master requests for data exchange. Figure 6.15-1 depicts the SPI block diagram.

Features of SPI Controller

- Support MSB or LSB first transfer
- Support programmable SPI SCLK
- Support programmable master or slave mode
  - Master maximum@6MHz with PFB clock 12MHz
  - Slave maximum@3MHz with PFB clock 12MHz
- Support 8- or 16- or 24-bit address bus
- 8-byte hardware transmit or receive FIFO

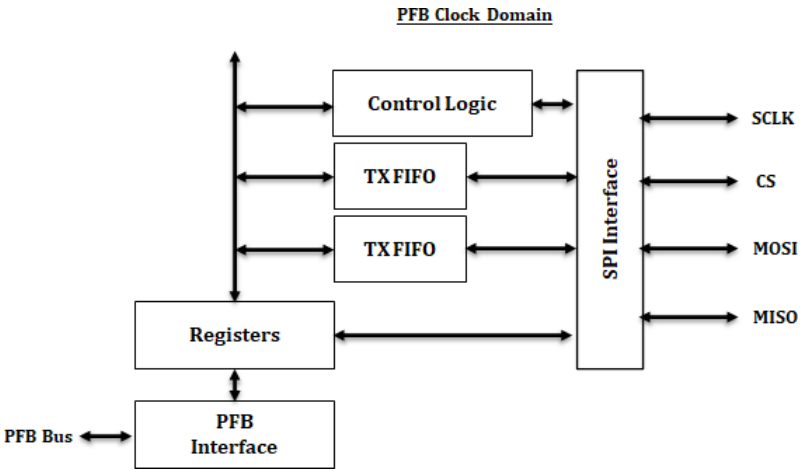


Figure 6.15-1 SPI Controller Block Diagram

Master Mode

The SPI controller can act as a SPI master, initiating SPI transfers on the SPI bus. The SPI transfer format and interface timing are programmable via the PFB programing port. The SPI transfers are initiated through direct register programming.

Figure 6.15-2 shows an example of SPI transfer format which includes command, address, and data phases for TX/RX data transfer. The controller provides dedicated registers to specify the contents of command, address, and data fields. The data register is shared by both TX and RX data transfers. The SPI controller provides TX/RX FIFO threshold interrupts to ease flow control under the PIO

programming. The controller also has a programmable bit to issue an interrupt once the transfer completes.

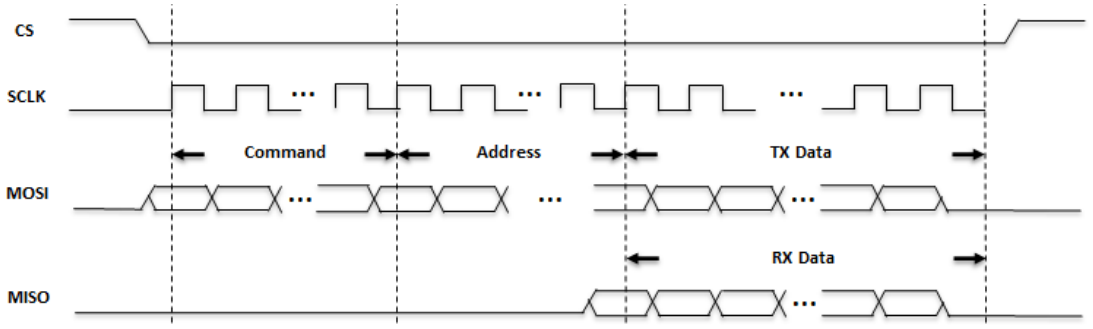


Figure 6.15-2 Transfer Format

Slave Mode

The SPI slave interprets the packet on the SPI I/O interface as three fields: slave command, slave dummy and slave data. The slave command field and slave dummy field are always 8-bit in length. The slave data field format is defined by the transfer control register (TRNMODE field in TRANSCTRL3 Register). Figure 6.15-3, Figure 6.15-4, Figure 6.15-5, and Figure 6.15-6 depict the timing diagram of read status command, read or write data command and user-defined command.

Slave Command Name	OP Code	Slave Data
Read Status IO	0x05	Reply 32-bit slave status registers (SLVSTAT2 ~ SLVSTAT0)
Read Data IO	0x0B	Reply data from the data register in the FIFO manner
Write Data IO	0x51	Data saved to the data register in the FIFO manner
User-defined	Any 8-bit numbers except the listed OP codes above	Depending on the transfer control register

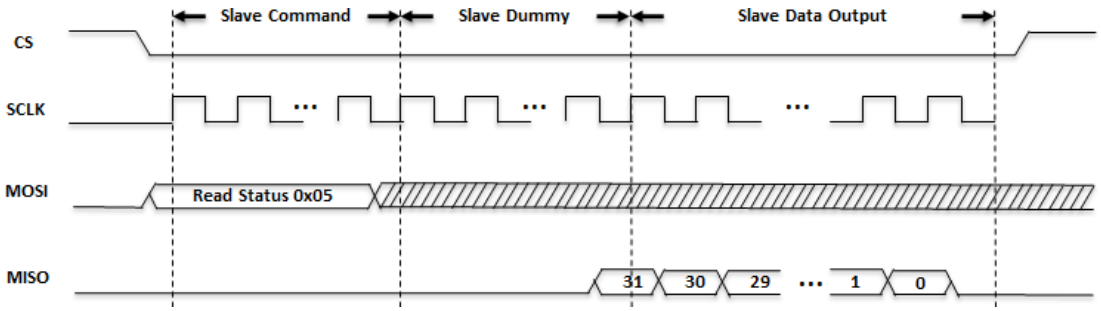


Figure 6.15-3 Timing Diagram of Read Slave Status Command

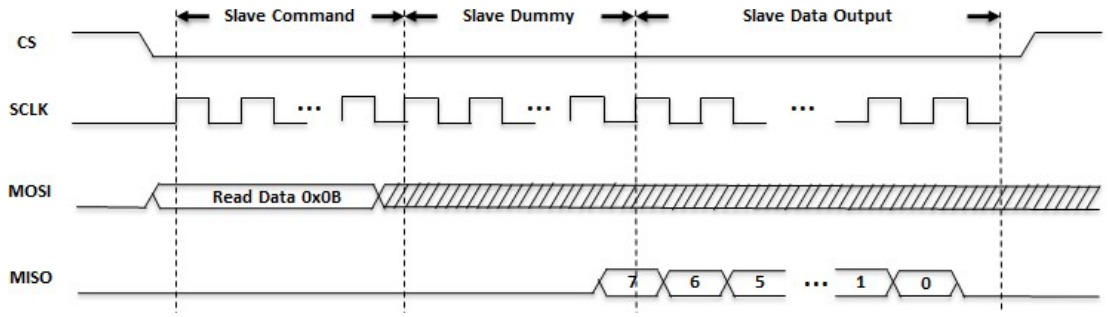


Figure 6.15-4 Timing Diagram of Data-Reading Command

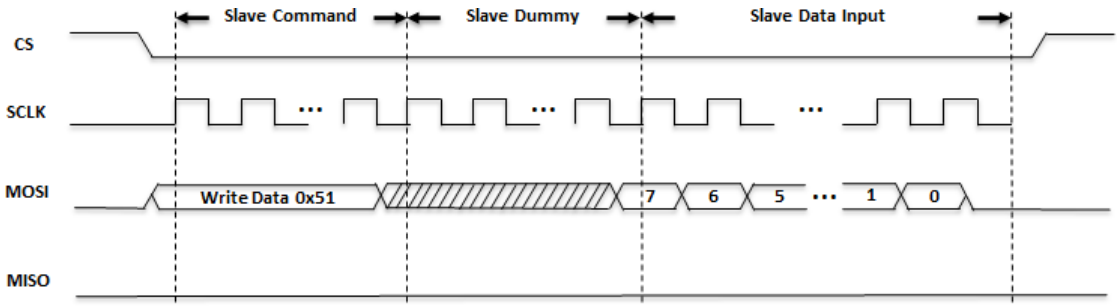


Figure 6.15-5 Timing Diagram of Data-Writing Command

For the user-defined command, the “slave data field” format is defined by the transfer control register (TRNMODE field in TRANSCTRL3 Register). For example, if the transfer mode is {Read Only}, only the data read field will be logged into the data register.

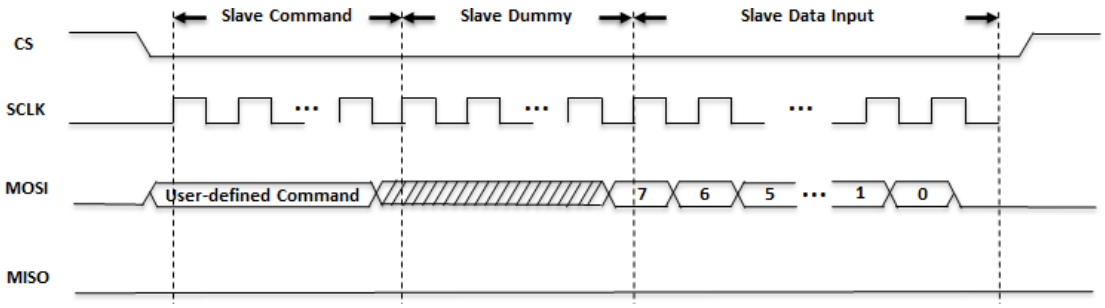


Figure 6.15-6 Timing Diagram of User-defined Command

Note that the user define command provides flexibility for customer to transceiver the data field as compared with read or write data command. Please see section 6.15.2 TRANSCTRL3 Register for more details.

6.15.2 Register Map and Description

Base Address (SPI_BA) : 0xFC00				
Register	**Offset**	**RW**	**Description**	**Reset Value**
TRNFMT0	SPI_BA+0x00	R/W	SPI Transfer Format 0 Register	0x00
TRNFMT1	SPI_BA+0x01	R/W	SPI Transfer Format 1 Register	0x00
TRNCTRL0	SPI_BA+0x02	R/W	SPI Transfer Control 0 Register	0x00
TRNCTRL1	SPI_BA+0x03	R/W	SPI Transfer Control 1 Register	0x00
TRNCTRL2	SPI_BA+0x04	R/W	SPI Transfer Control 2 Register	0x00
TRNCTRL3	SPI_BA+0x05	R/W	SPI Transfer Control 3 Register	0x00
SPICMD	SPI_BA+0x06	R/W	SPI Command Register	0x00
SPIADR0	SPI_BA+0x07	R/W	SPI Address 0 Register	0x00
SPIADR1	SPI_BA+0x08	R/W	SPI Address 1 Register	0x00
SPIADR2	SPI_BA+0x09	R/W	SPI Address 2 Register	0x00
SPIDATA	SPI_BA+0x0A	R/W	SPI Data Register	0x00
SPICTRL0	SPI_BA+0x0B	R/W	SPI Control 0 Register	0x00
SPICTRL1	SPI_BA+0x0C	R/W	SPI Control 1 Register	0x00
SPICTRL2	SPI_BA+0x0D	R/W	SPI Control 2 Register	0x00
SPISTAT0	SPI_BA+0x0E	R/W	SPI Status 0 Register	0x00
SPISTAT1	SPI_BA+0x0F	R/W	SPI Status 1 Register	0x00
SPISTAT2	SPI_BA+0x10	R/W	SPI Status 2 Register	0x00
INTEN	SPI_BA+0x11	R/W	SPI Interrupt Enable Register	0x00
INTSTAT	SPI_BA+0x12	R/W	SPI Interrupt Status Register	0x00
INTFTIM0	SPI_BA+0x13	R/W	SPI Interface Timing 0 Register	0x00
INTFTIM1	SPI_BA+0x14	R/W	SPI Interface Timing 1 Register	0x00
SLVSTAT0	SPI_BA+0x15	R/W	SPI Slave Status 0 Register	0x00
SLVSTAT1	SPI_BA+0x16	R/W	SPI Slave Status 1 Register	0x00
SLVSTAT2	SPI_BA+0x17	R/W	SPI Slave Status 2 Register	0x00
RCVCNT0	SPI_BA+0x18	R/W	SPI Slave Receive Counter 0 Register	0x00
RCVCNT1	SPI_BA+0x19	R/W	SPI Slave Receive Counter 1 Register	0x00
TSMCNT0	SPI_BA+0x1A	R/W	SPI Transmit Counter 0 Register	0x00
TSMCNT1	SPI_BA+0x1B	R/W	SPI Transmit Counter 1 Register	0x00
SPICFG	SPI_BA+0x1C	R/	SPI Configuration Register	0x00

SPI Transfer Format 0 Register

Register	Offset	RW	Description	Reset Value
TRNFMT0	SPI_BA+0x00	R/W	SPI Transfer Format 0 Register	0x00

Bits	Flag	Description
[7:5]	Reserved	**Reserved**.
[4]	MOSIBID	**SPI Bi-directional MOSI in Regular Mode Selection Flag** This bit is set by software, cleared by software. MOSIBID = 0, MOSI is uni-directional signal in regular (single) mode. MOSIBID = 1, MOSI is bi-directional signal in regular (single) mode.
[3]	LSB	**SPI LSB/MSB Bit First Selection Flag** This bit is set by software, cleared by software. LSB = 0, Most significant bit first. LSB = 1, Least significant bit first.
[2]	SLVMODE	**SPI Master/Slave Mode Selection Flag** This bit is set by software, cleared by software. SLVMODE = 0, Master mode. SLVMODE = 1, Slave mode.
[1]	CPOL	**SPI Clock Polarity Phase Flag** This bit is set by software, cleared by software. CPOL = 0, SCLK is LOW in the idle states. CPOL = 1, SCLK is HIGH in the idle states.
[0]	CPHA	**SPI Clock Sampling Phase Flag** This bit is set by software, cleared by software. CPHA = 0, Sampling data at pose-edge SCLK. CPHA = 1, Sampling data at neg-edge SCLK.

SPI Transfer Format 1 Register

Register	Offset	RW	Description	Reset Value
TRNFMT1	SPI_BA+0x01	R/W	SPI Transfer Format 1 Register	0x00

Bits	Flag	Description
[7:6]	ADRLEN	**SPI Address Length Selection Flag** This bit is set by software, cleared by software. ADRLEN = 0, 1-byte address. ADRLEN = 1, 2-byte address. ADRLEN = 2, 3-byte address. ADRLEN = 3, Reserved.
[5]	Reserved	**Reserved**.
[4:0]	DATLEN	**SPI Data Length Selection Flag** This bit is set by software, cleared by software. The actual bit number of a data unit is (DATLEN + 1) : Bit length of data

SPI Transfer Control 0 Register

Register	Offset	RW	Description	Reset Value
TRNCTRL0	SPI_BA+0x02	R/W	SPI Transfer Control 0 Register	0x00

Bits	Flag	Description
[7:0]	RTRNCNT	**SPI Transfer Count for Read Data** This bit is set by software, cleared by software. RTRNCNT indicates the number of units of data to be received from SPI bus and stored to the Date Register. The actual received count is (RTRNCNT+1). RTRNCNT only takes effect when TRNMODE is 0, 2, 3, or 4. The size (bit-width) of a data unit is defined by the DATLEN field of the transfer format register (TRANFMT1). For TRNMODE 0, WTRNCNT must be equal to RTRNCNT.

SPI Transfer Control 235 Register

Register	Offset	RW	Description	Reset Value
TRNCTRL1	SPI_BA+0x03	R/W	SPI Transfer Control 1 Register	0x00

Bits	Flag	Description
[7:4]	WTRNCNT	**SPI Transfer Count for Write Data** This bit is set by software, cleared by software. WTRNCNT indicates the number of units of data to be transmitted to the SPI bus from the Data Register. The actual transfer count is (WrTranCnt+1). WTRNCNT only takes effect when TRNMODE is 0, 1, 3, or 4. The size (bit-width) of a data unit is defined by the DATLEN field of the Transfer Format Register. For TRNMODE 0, WTRNCNT must be equal to RTRNCNT.
[3]	TOKENVU	**SPI Special Token Value Following the Address Phase (Master Mode Only)** This bit is set by software, cleared by software. TOKENVU = 0, Token value = 0x00. TOKENVU = 1, Token value = 0x69.
[2:1]	Reserved	**Reserved**.
[0]	RTRNCNT	**SPI Transfer Count for Read Data** This bit is set by software, cleared by software. RTRNCNT indicates the number of units of data to be received from SPI bus and stored to the Date Register. The actual received count is (RTRNCNT+1). RTRNCNT only takes effect when TRNMODE is 0, 2, 3, or 4. The size (bit-width) of a data unit is defined by the DATLEN field of the transfer format register (TRANFMT1). For TRNMODE 0, WTRNCNT must be equal to RTRNCNT.

SPI Transfer Control 236 Register

Register	Offset	RW	Description	Reset Value
TRNCTRL2	SPI_BA+0x04	R/W	SPI Transfer Control 2 Register	0x00

Bits	Flag	Description
[7:6]	Reserved	**Reserved**.
[5]	TOKENEN	**SPI Special Token Value Enable Flag (Master Mode Only)** This bit is set by software, cleared by software. TOKENEN = 0, One-byte special token disabled. TOKENEN = 1, One-byte special token enabled.
[4:0]	WTRNCNT	**SPI Transfer Count for Write Data** This bit is set by software, cleared by software. WTRNCNT indicates the number of units of data to be transmitted to the SPI bus from the Data Register. The actual transfer count is (WrTranCnt+1). WTRNCNT only takes effect when TRNMODE is 0, 1, 3, or 4. The size (bit-width) of a data unit is defined by the DATLEN field of the Transfer Format Register. For TRNMODE 0, WTRNCNT must be equal to RTRNCNT.

SPI Transfer Control 237 Register

Register	Offset	RW	Description	Reset Value
TRNCTRL3	SPI_BA+0x05	R/W	SPI Transfer Control 3 Register	0x00

Bits	Flag	Description
[7]	Reserved	**Reserved**.
[6]	CMDEN	**SPI Command Phase Enable Flag (Master Mode Only)** This bit is set by software, cleared by software. CMDEN = 0, Command phase disabled. CMDEN = 1, Command phase enabled.
[5]	ADREN	**SPI Address Phase Enable Flag (Master Mode Only)** This bit is set by software, cleared by software. ADREN = 0, Address phase disabled. ADREN = 1, Address phase enabled.
[4]	Reserved	**Reserved**.
[3:0]	TRNMODE	**SPI Transfer Mode Flag** This bit is set by software, cleared by software. TRNMODE = 0, Write and read at the same time. TRNMODE = 1, Write only. TRNMODE = 2, Read only. TRNMODE = 3, Write then read. TRNMODE = 4, Read then write. TRNMODE = 5, Reserved. TRNMODE = 6, Reserved. TRNMODE = 7, None data (must enable CMDEN or ADDREN in master mode). TRNMODE > 8, Reserved.

SPI Command Register

Register	Offset	RW	Description	Reset Value
SPICMD	SPI_BA+0x06	R/W	SPI Command Register	0x00

Bits	Flag	Description
[7:0]	CMD	**SPI Command Register** This bit is set by software, cleared by software. Write operations on this register trigger SPI transfers. This register must be written with a dummy value to start a SPI transfer even when the command phase is not enabled. When the SPI controller is programmed to the slave mode, the command field of the last received SPI transaction is stored in this SPI command register.

SPI Address Register

Register	Offset	RW	Description	Reset Value
SPIADR0	SPI_BA+0x07	R/W	SPI Address 0 Register	0x00

Bits	Flag	Description
[7:0]	SPIADRx, x = 0, 1, 2	**SPI Address x Register, x = 0, 1, 2** This bit is set by software, cleared by software. SPI address.

SPI Data Register

Register	Offset	RW	Description	Reset Value
SPIDATA	SPI_BA+0x0A	R/W	SPI Data Register	0x00

Bits	Flag	Description
[7:0]	SPIDAT	**SPI Data Register** This bit is set by software, cleared by software. For writes, data is enqueued to the TX FIFO. The least significant byte is always transmitted first. For reads, data is read and dequeued from the RX FIFO. The least significant byte is the first received byte. The FIFOs decouple the speed of the SPI transfers and the software's generation/consumption of data. When the TX FIFO is empty, SPI transfers will hold until more data is written to the TX FIFO; when the RX FIFO is full, SPI transfers will hold until there is more room in the RX FIFO. If more data is written to the TX FIFO than the write transfer count (WRTRANCNT), the remaining data will stay in the TX FIFO for the next transfer or until the TX FIFO is reset.

SPI Control 0 Register

Register	Offset	RW	Description	Reset Value
SPICTRL0	SPI_BA+0x0B	R/W	SPI Control 0 Register	0x00

Bits	Flag	Description
[7:3]	Reserved	**Reserved**.
[2]	TFIFORST	**SPI Transmit FIFO Reset Enable Flag** This bit is set by software, cleared by software. TFIFORST = 0, Idle. TFIFORST = 1, Transmit FIFO reset.
[1]	RFIFORST	**SPI Receive FIFO Reset Enable Flag** This bit is set by software, cleared by software. RFIFORST = 0, Idle. RFIFORST = 1, Receive FIFO reset.
[0]	SPIRST	**SPI Reset Enable Flag** This bit is set by software, cleared by hardware. SPIRST = 0, Idle. SPIRST = 1, SPI reset.

SPI Control 242 Register

Register	Offset	RW	Description	Reset Value
SPICTRL1	SPI_BA+0x0C	R/W	SPI Control 1 Register	0x00

Bits	Flag	Description
[7:5]	Reserved	**Reserved**.
[4:0]	RTHRCV	**SPI Receive FIFO Threshold Flag** This bit is set by software, cleared by software. The RXFIFOINT interrupt would be issued for consuming the RX FIFO when the RX data count is more than or equal to the RX FIFO threshold.

SPI Control 243 Register

Register	Offset	RW	Description	Reset Value
SPICTRL2	SPI_BA+0x0D	R/W	SPI Control 2 Register	0x00

Bits	Flag	Description
[7:5]	Reserved	**Reserved**.
[4:0]	TTHTRN	**SPI Transmit FIFO Threshold Flag** This bit is set by software, cleared by software. The TXFIFOINT interrupt would be issued to replenish the TX FIFO when the TX data count is less than or equal to the TX FIFO threshold.

SPI Status 0 Register

Register	Offset	RW	Description	Reset Value
SPISTAT0	SPI_BA+0x0E	R/W	SPI Status 0 Register	0x00

Bits	Flag	Description
[7:1]	Reserved	**Reserved**.
[0]	BUSY	**SPI In Progress Flag** This bit is set by hardware, cleared by hardware. BUSY = 0, Idle. BUSY = 1, SPI direct register programming is in progress.

SPI Status 245 Register

Register	Offset	RW	Description	Reset Value
SPISTAT1	SPI_BA+0x0F	R/W	SPI Status 1 Register	0x00

Bits	Flag	Description
[7]	RFUL	**SPI Receive FIFO Full Flag** This bit is set by hardware, cleared by hardware. RFUL = 0, Receive FIFO not full. RFUL = 1, Receive FIFO full.
[6]	REMP	**SPI Receive FIFO Empty Flag** This bit is set by hardware, cleared by hardware. REMP = 0, Receive FIFO not empty. REMP = 1, Receive FIFO empty.
[5]	Reserved	**Reserved**.
[4:0]	RFIFONUM	**SPI Number of Entries in the Receive FIFO** This bit is set by hardware, cleared by hardware.

SPI Status 246 Register

Register	Offset	RW	Description	Reset Value
SPISTAT2	SPI_BA+0x10	R/W	SPI Status 2 Register	0x00

Bits	Flag	Description
[7]	TFUL	**SPI Transmit FIFO Full Flag** This bit is set by hardware, cleared by hardware. TFUL = 0, Transmit FIFO not full. TFUL = 1, Transmit FIFO full.
[6]	TEMP	**SPI Transmit FIFO Empty Flag** This bit is set by hardware, cleared by hardware. TEMP = 0, Transmit FIFO not empty. TEMP = 1, Transmit FIFO empty.
[5]	Reserved	Reserved.
[4:0]	TFIFONUM	**SPI Number of Entries in the Transmit FIFO** This bit is set by hardware, cleared by hardware.

SPI Interrupt Enable Register

Register	Offset	RW	Description	Reset Value
INTEN	**SPI_BA+0x11**	**R/W**	SPI Interrupt Enable Register	**0x00**

Bits	Flag	Description
[7:6]	Reserved	**Reserved**.
[5]	SLVCMDIEN	**SPI Slave Command Interrupt Enable Flag (Slave Mode Only)** This bit is set by software, cleared by software. SLVCMDIEN = 0, Slave command interrupt disabled. SLVCMDIEN = 1, Slave command interrupt enabled.
[4]	ENDTIEN	**SPI End of Transfer Interrupt Enable Flag** This bit is set by software, cleared by software. ENDTIEN = 0, End of transfer interrupt disabled. ENDTIEN = 1, End of transfer interrupt enabled.
[3]	TFIFOIEN	**SPI Transmit FIFO Interrupt Enable Flag** This bit is set by software, cleared by software. In slave mode, end of read status transaction does not trigger this interrupt TFIFOIEN = 0, Receive FIFO interrupt disabled. TFIFOIEN = 1, Receive FIFO interrupt enabled.
[2]	RFIFOIEN	**SPI Receive FIFO Interrupt Enable Flag** This bit is set by software, cleared by software. RFIFOIEN = 0, Receive FIFO interrupt disabled. RFIFOIEN = 1, Receive FIFO interrupt enabled.
[1]	TFIFOURIEN	**SPI Transmit FIFO Underrun Interrupt Enable Flag (Slave Mode Only)** This bit is set by software, cleared by software. TFIFOURIEN = 0, Receive FIFO underrun interrupt disabled. TFIFOURIEN = 1, Receive FIFO underrun interrupt enabled.
[0]	RFIFOORIEN	**SPI Receive FIFO Overrun Interrupt Enable Flag (Slave Mode Only)** This bit is set by software, cleared by software. RFIFOORIEN = 0, Receive FIFO overrun interrupt disabled. RFIFOORIEN = 1, Receive FIFO overrun interrupt enabled.

SPI Interrupt Status Register

Register	Offset	RW	Description	Reset Value
INTSTAT	SPI_BA+0x12	R/W	SPI Interrupt Status Register	0x00

Bits	Flag	Description
[7:6]	Reserved	**Reserved**.
[5]	SLVCMDINTF	**SPI Slave Command Interrupt Flag (Slave Mode Only)** This bit is set by hardware, cleared by software. SLVCMDINTF = 0, Otherwise. SLVCMDINTF = 1, Slave command interrupt triggered.
[4]	ENDTINTF	**SPI End of Transfer Interrupt Flag** This bit is set by hardware, cleared by software. ENDTINTF = 0, Otherwise. ENDTINTF = 1, End of transfer interrupt triggered.
[3]	TFIFOINTF	**SPI Transmit FIFO Interrupt Flag** This bit is set by hardware, cleared by software. In slave mode, end of read status transaction does not trigger this interrupt TFIFOINTF = 0, Otherwise. TFIFOINTF = 1, Receive FIFO interrupt triggered.
[2]	RFIFOINTF	**SPI Receive FIFO Interrupt Flag** This bit is set by hardware, cleared by software. RFIFOINTF = 0, Otherwise. RFIFOINTF = 1, Receive FIFO interrupt triggered.
[1]	TFIFOURINTF	**SPI Transmit FIFO Underrun Interrupt Flag (Slave Mode Only)** This bit is set by hardware, cleared by software. TFIFOURINTF = 0, Otherwise. TFIFOURINTF = 1, Receive FIFO underrun interrupt triggered.
[0]	RFIFOORINTF	**SPI Receive FIFO Overrun Interrupt Flag (Slave Mode Only)** This bit is set by hardware, cleared by software. RFIFOORINTF = 0, Otherwise. RFIFOORINTF = 1, Receive FIFO overrun interrupt triggered.

SPI Interface Timing 0 Register

Register	Offset	RW	Description	Reset Value
INTFTIM0	SPI_BA+0x13	R/W	SPI Interface Timing 0 Register	0x00

Bits	Flag	Description
[7:0]	SCLKDIV	**SPI Clock Divider Flag** This bit is set by software, cleared by software. SCLK period = ((SCLKDIV+1)×2)×(Period of the SPI clock source) The SCLKDIV value 0xff is a special value which indicates that the SCLK frequency should be the same as the SPI source clock frequency.

SPI Interface Timing 1 Register

Register	Offset	RW	Description	Reset Value
INTFTIM1	SPI_BA+0x14	R/W	SPI Interface Timing 1 Register	0x00

Bits	Flag	Description
[7:6]	Reserved	**Reserved**.
[5:4]	CS2SCLK	**SPI Minimum Time Between Edges of CS and Edges of SCLK** This bit is set by software, cleared by software. The actual duration is (SCLK_Period/2)×(CS2CLK+1)
[3:0]	CSHT	**SPI Minimum Time of CS Stay HIGH** This bit is set by software, cleared by software. The actual duration is (SCLK_Period/2)×(CSHT+1)

SPI Slave Status 0 Register

Register	Offset	RW	Description	Reset Value
SLVSTAT0	SPI_BA+0x15	R/W	SPI Slave Status 0 Register	0x00

Bits	Flag	Description
[7:0]	USERSTAT0	**SPI User Defined Status 0 Flag** This bit is set by software, cleared by software.

The Slave Status Register keeps slave statuses. An SPI master can get these statuses by issuing status-reading commands.

SPI Slave Status 251 Register

Register	Offset	RW	Description	Reset Value
SLVSTAT1	SPI_BA+0x16	R/W	SPI Slave Status 1 Register	0x00

Bits	Flag	Description
[7:0]	USERSTAT1	**SPI User Defined Status 1 Flag** This bit is set by software, cleared by software.

The Slave Status Register keeps slave statuses. An SPI master can get these statuses by issuing status-reading commands.

SPI Slave Status 2 Register

Register	Offset	RW	Description	Reset Value
SLVSTAT2	SPI_BA+0x17	R/W	SPI Slave Status 2 Register	0x00

Bits	Flag	Description
[7:3]	Reserved	**Reserved**.
[2]	URRUN	**SPI Data Underrun Flag** This bit is set by hardware, cleared by software. URRUN = 0, Data underrun not occurs. URRUN = 1, Data underrun occurs in the last transaction.
[1]	ORRUN	**SPI Data Overrun Flag** This bit is set by hardware, cleared by software. ORRUN = 0, Data overrun not occurs. ORRUN = 1, Data overrun occurs in the last transaction.
[0]	RDY	**SPI Ready Flag** This bit is set by software, cleared by hardware. When an SPI transaction other than slave status-reading command ends, this bit will be cleared to 0. RDY = 0, SPI busy. RDY = 1, SPI ready.

The Slave Status Register keeps slave statuses. An SPI master can get these statuses by issuing status-reading commands.

SPI Slave Receive Counter 0 Register

Register	Offset	RW	Description	Reset Value
RCVCNT0	SPI_BA+0x18	R/W	SPI Slave Receive Counter 0 Register	0x00

Bits	Flag	Description
[7:0]	RCVCNTL	**SPI Received Data Count Low Field** This bit is set by hardware.

SPI Slave Receive Counter 1 Register

Register	Offset	RW	Description	Reset Value
RCVCNT1	SPI_BA+0x19	R/W	SPI Slave Receive Counter 1 Register	0x00

Bits	Flag	Description
[7:1]	Reserved	**Reserved**.
[0]	RCVCNTH	**SPI Received Data Count High Field** This bit is set by hardware.

SPI Slave Transmit Counter 0 Register

Register	Offset	RW	Description	Reset Value
TSMCNT0	SPI_BA+0x1A	R/W	SPI Transmit Counter 0 Register	0x00

Bits	Flag	Description
[7:0]	TRNCNTL	**SPI Transmit Data Count Low Field** This bit is set by hardware.

SPI Slave Transmit Counter 1 Register

Register	Offset	RW	Description	Reset Value
TSMCNT1	SPI_BA+0x1B	R/W	SPI Transmit Counter 1 Register	0x00

Bits	Flag	Description
[7:1]	Reserved	**Reserved**.
[0]	TRNCNTH	**SPI Transmit Data Count High Field** This bit is set by hardware.

SPI Configuration Register

Register	Offset	RW	Description	Reset Value
SPICFG	SPI_BA+0x1C	R	SPI Configuration Register	0x00

Bits	Flag	Description
[7:6]	Reserved	**Reserved**.
[5:4]	TFIFOSIZE	**SPI Transmit FIFO Size Flag** This bit is set by hardware. TFIFOSIZE = 0, 8-byte transmit FIFO size.
[3:2]	Reserved	**Reserved**.
[1:0]	RFIFOSIZE	**SPI Receive FIFO Size Flag** This bit is set by hardware. RFIFOSIZE = 0, 8-byte receive FIFO size.

6.15.3 Programming Model (Please refer to “Sample code project”)

Master Transmit

Step 1 : Set SPI transfer format

Set SLVMODE = 0 (master mode) in TRNFMT0 Register.

Set ADRLEN = 1 (2-byte address), DATLEN = 7 (8-bit data) in TRNFMT1 Register.

Step 2 : Set SPI transfer control

Set WTRNCNT = 0 (1-byte data) in TRNCTRL1 Register.

Set CMDEN = 0 (no command phase), ADREN = 1 (address phase), TRNMODE = 1 (write only) in TRNCTRL3 Register.

Step 3 : Set SPI control

Set SPIRST = 1 (SPI reset), RFIFORST = 1 (receive FIFO reset), TFIFORST = 1 (transmit FIFO reset) in SPICTRL0 Register.

Step 4 : Set SPI interrupt enable

Set ENDTIEN = 1 in INTEN Register.

Step 5 : Set SPI interface timing

Set SCLKDIV = 0 ((SCLKDIV+1) x 2 x 80ns = 6MHz) in INTFTIM0 Register.

Step 6 : Set SPI address (2-byte address)

Set SPIADR0 in SPIADR0 Register.

Set SPIADR1 in SPIADR1 Register.

Step 7 : Set SPI data

Set SPIDAT in SPIDATA Register.

Step 8 : Start SPI data transfer

Set CMD = 0x01 in SPICMD Register to start SPI transmit.

Master Receive

Step 1 : Set SPI transfer format

Set SLVMODE = 0 (master mode) in TRNFMT0 Register.

Set ADRLEN = 1 (2-byte address), DATLEN = 7 (8-bit data) in TRNFMT1 Register.

Step 2 : Set SPI transfer control

Set RTRNCNT = 0 (1-byte data) in TRNCTRL0 Register.

Set CMDEN = 0 (no command phase), ADREN = 1 (address phase), TRNMODE = 2 (read only) in TRNCTRL3 Register.

Step 3 : Set SPI control

Set SPIRST = 1 (SPI reset), RFIFORST = 1 (receive FIFO reset), TFIFORST = 1 (transmit FIFO reset) in SPICTRL0 Register.

Step 4 : Set SPI interrupt enable

Set ENDTIEN = 1 in INTEN Register.

Step 5 : Set SPI interface timing

Set SCLKDIV = 0 ((SCLKDIV+1) x 2 x 80ns = 6MHz) in INTFTIM0 Register.

Step 6 : Set SPI address (2-byte address)

Set SPIADR0 in SPIADR0 Register.

Set SPIADR1 in SPIADR1 Register.

Step 7 : Start SPI data transfer

Set CMD = 0x01 in SPICMD Register to start SPI receive.

Slave Transmit

Step 1 : Set SPI transfer format

Set SLVMODE = 1 (slave mode) in TRNFMT0 Register.

Set ADRLEN = 0 (1-byte address), DATLEN = 7 (8-bit data) in TRNFMT1 Register.

Step 2 : Set SPI transfer control

Set WTRNCNT = 0 (1-byte data) in TRNCTRL1 Register.

Set TRNMODE = 1 (write only) in TRNCTRL3 Register.

Step 3 : Set SPI control

Set SPIRST = 1 (SPI reset), RFIFORST = 1 (receive FIFO reset), TFIFORST = 1 (transmit FIFO reset) in SPICTRL0 Register.

Step 4 : Set SPI interrupt enable

Set ENDTIEN = 1 in INTEN Register.

Step 5 : Set SPI data

Set SPIDAT in SPIDATA Register.

Step 6 : Set ready flag

Set RDY = 1 in SLVSTAT2 Register.

Slave Receive

Step 1 : Set SPI transfer format

Set SLVMODE = 1 (slave mode) in TRNFMT0 Register.

Set ADRLEN = 0 (1-byte address), DATLEN = 7 (8-bit data) in TRNFMT1 Register.

Step 2 : Set SPI transfer control

Set RTRNCNT = 0 (1-byte data) in TRNCTRL0 Register.

Set TRNMODE = 2 (read only) in TRNCTRL3 Register.

Step 3 : Set SPI control

Set SPIRST = 1 (SPI reset), RFIFORST = 1 (receive FIFO reset), TFIFORST = 1 (transmit FIFO reset) in SPICTRL0 Register.

Step 4 : Set SPI interrupt enable

Set ENDTIEN = 1 in INTEN Register.

6.16 Multi-Function ALU (MFA)

6.16.1 Overview

Features of Multi-Function ALU (MFA)

- 16-bit signed- or unsigned- multiplier
- IEEE754 floating point multiplier
- IEEE754 Log2 unit

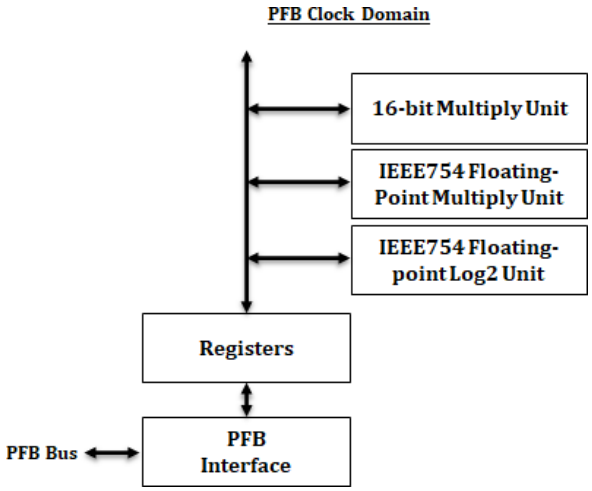


Figure 6.16-1 MFA Block Diagram

16-bit Multiply Unit

MFA supports a single cycle 16-bit signed or unsigned multiply for fast multiply computation without software library overhead. The product of the multiply unit is a 32-bit data with friendly interface for accessing.

IEEE754 Floating Point Multiply Unit

MFA supports IEEE754 compatible floating-point standard for multiply operation with single, double precision, and rounding mode. The round mode is set by floating-point control register (FPCTRL). The product of the floating point multiply unit is a 32-bit data with status information defined in floating-point status register (FPSTAT).

IEEE754 Log2 Unit

MFA supports IEEE754 compatible floating-point standard for Log2 operation. The result of the floating-point log2 unit is a 32-bit data with status information define in log2 status register (LOGSTAT).

6.16.2 Register Map and Description

Base Address (MFA_BA) : 0xFE00				
Register	**Offset**	**RW**	**Description**	**Reset Value**
MULCTRL	MFA_BA+0x00	R/W	Multiply Control Register	0x00
MULP0	MFA_BA+0x01	R/W	Multiply Multiplier Data 0 Register	0x00
MULP1	MFA_BA+0x02	R/W	Multiply Multiplier Data 1 Register	0x00
MULC0	MFA_BA+0x03	R/W	Multiply Multiplicand Data 0 Register	0x00
MULC1	MFA_BA+0x04	R/W	Multiply Multiplicand Data 1 Register	0x00
MULPROD0	MFA_BA+0x05	R	Multiply Product Data 0 Register	0x00
MULPROD1	MFA_BA+0x06	R	Multiply Product Data 1 Register	0x00
MULPROD2	MFA_BA+0x07	R	Multiply Product Data 2 Register	0x00
MULPROD3	MFA_BA+0x08	R	Multiply Product Data 3 Register	0x00
FPCTRL	MFA_BA+0x09	R/W	Floating Point Multiply Control Register	0x00
FPMULP0	MFA_BA+0x0A	R/W	Floating Point Multiply Multiplier Data 0 Register	0x00
FPMULP1	MFA_BA+0x0B	R/W	Floating Point Multiply Multiplier Data 1 Register	0x00
FPMULP2	MFA_BA+0x0C	R/W	Floating Point Multiply Multiplier Data 2 Register	0x00
FPMULP3	MFA_BA+0x0D	R/W	Floating Point Multiply Multiplier Data 3 Register	0x00
FPMULC0	MFA_BA+0x0E	R/W	Floating Point Multiply Multiplicand Data 0 Register	0x00
FPMULC1	MFA_BA+0x0F	R/W	Floating Point Multiply Multiplicand Data 1 Register	0x00
FPMULC2	MFA_BA+0x10	R/W	Floating Point Multiply Multiplicand Data 2 Register	0x00
FPMULC3	MFA_BA+0x11	R/W	Floating Point Multiply Multiplicand Data 3 Register	0x00
FPPROD0	MFA_BA+0x12	R	Floating Point Multiply Product Data 0 Register	0x00
FPPROD1	MFA_BA+0x13	R	Floating Point Multiply Product Data 1 Register	0x00
FPPROD2	MFA_BA+0x14	R	Floating Point Multiply Product Data 2 Register	0x00
FPPROD3	MFA_BA+0x15	R	Floating Point Multiply Product Data 3 Register	0x00
FPSTAT	MFA_BA+0x16	R	Floating Point Multiply Status Register	0x00
LOGIN0	MFA_BA+0x17	R/W	Floating Point Log2 Input Data 0 Register	0x00
LOGIN1	MFA_BA+0x18	R/W	Floating Point Log2 Input Data 1 Register	0x00
LOGIN2	MFA_BA+0x19	R/W	Floating Point Log2 Input Data 2 Register	0x00
LOGIN3	MFA_BA+0x1A	R/W	Floating Point Log2 Input Data 3 Register	0x00
LOGOUT0	MFA_BA+0x1B	R	Floating Point Log2 Output Data 0 Register	0x00
LOGOUT1	MFA_BA+0x1C	R	Floating Point Log2 Output Data 1 Register	0x00
LOGOUT2	MFA_BA+0x1D	R	Floating Point Log2 Output Data 2 Register	0x00
LOGOUT3	MFA_BA+0x1E	R	Floating Point Log2 Output Data 3 Register	0x00
LOGSTAT	MFA_BA+0x1F	R	Floating Point Log2 Status Register	0x00

MFA Multiply Control Register

Register	**Offset**	**RW**	**Description**	**Reset Value**
MULCTRL	MFA_BA+0x00	R/W	Multiply Control Register	0x00

Bits	**Flag**	**Description**
[7:1]	Reserved	**Reserved**.
[0]	SIGN	**Multiply Sign Flag** This bit is set by software, and cleared by software. SIGN = 0, unsigned multiply. SIGN = 1, signed multiply.

MFA Multiply Multiplier Data Register

Register	Offset	RW	Description	Reset Value
MULP0	MFA_BA+0x01	R/W	Multiply Multiplier Data 0 Register	0x00
MULP1	MFA_BA+0x02	R/W	Multiply Multiplier Data 1 Register	0x00

Bits	Flag	Description
[7:0]	MULTIPLIERx, x = 0, 1	**Multiply Multiplier Data** This byte is set by software, and cleared by software.

MFA Multiply Multiplicand Data Register

Register	Offset	RW	Description	Reset Value
MULC0	MFA_BA+0x03	R/W	Multiply Multiplicand Data 0 Register	0x00
MULC1	MFA_BA+0x04	R/W	Multiply Multiplicand Data 1 Register	0x00

Bits	Flag	Description
[7:0]	MULTIPLICANDx, x = 0, 1	**Multiply Multiplicand Data** This byte is set by software, and cleared by software.

MFA Multiply Product Data Register

Register	Offset	RW	Description	Reset Value
MULPROD0	MFA_BA+0x05	R	Multiply Product Data 0 Register	0x00
MULPROD1	MFA_BA+0x06	R	Multiply Product Data 1 Register	0x00
MULPROD2	MFA_BA+0x07	R	Multiply Product Data 2 Register	0x00
MULPROD3	MFA_BA+0x08	R	Multiply Product Data 3 Register	0x00

Bits	Flag	Description
[7:0]	MULPRODUCTx, x = 0, 1, 2, 3	**Multiply Product Data** This byte is set by hardware within 1-T.

MFA Floating-Point Multiply Control Register

Register	Offset	RW	Description	Reset Value
FPCTRL	MFA_BA+0x09	R/W	Floating Point Multiply Control Register	0x00

Bits	Flag	Description
[7:3]	Reserved	**Reserved**.
[2:0]	ROUND	**Floating-Point Multiply Round Mode** This field is set by software, and cleared by software. ROUND = 0, Round to nearest (even). ROUND = 1, Round to zero. ROUND = 2, Round to positive infinity. ROUND = 3, Round to negative infinity. ROUND = 4, Round to nearest up. ROUND = 5, Round away from zero. ROUND = 6, Reserved. ROUND = 7, Reserved.

MFA Floating-Point Multiply Multiplier Data Register

Register	Offset	RW	Description	Reset Value
FPMULP0	MFA_BA+0x0A	R/W	Floating Point Multiply Multiplier Data 0 Register	0x00
FPMULP1	MFA_BA+0x0B	R/W	Floating Point Multiply Multiplier Data 1 Register	0x00
FPMULP2	MFA_BA+0x0C	R/W	Floating Point Multiply Multiplier Data 2 Register	0x00
FPMULP3	MFA_BA+0x0D	R/W	Floating Point Multiply Multiplier Data 3 Register	0x00

Bits	Flag	Description
[7:0]	FPMULTIPLIERx, x = 0, 1, 2, 3	**Floating-Point Multiply Multiplier Data** This byte is set by software, and cleared by software.

MFA Floating-Point Multiply Multiplicand Data Register

Register	Offset	RW	Description	Reset Value
FPMULC0	MFA_BA+0x0E	R/W	Floating Point Multiply Multiplicand Data 0 Register	0x00
FPMULC1	MFA_BA+0x0F	R/W	Floating Point Multiply Multiplicand Data 1 Register	0x00
FPMULC2	MFA_BA+0x10	R/W	Floating Point Multiply Multiplicand Data 2 Register	0x00
FPMULC3	MFA_BA+0x11	R/W	Floating Point Multiply Multiplicand Data 3 Register	0x00

Bits	Flag	Description
[7:0]	FPMULTIPLICANDx, x = 0, 1, 2, 3	**Floating-Point Multiply Multiplicand Data** This byte is set by software, and cleared by software.

MFA Floating-Point Multiply Product Data Register

Register	Offset	RW	Description	Reset Value
FPPROD0	MFA_BA+0x12	R	Floating Point Multiply Product Data 0 Register	0x00
FPPROD1	MFA_BA+0x13	R	Floating Point Multiply Product Data 1 Register	0x00
FPPROD2	MFA_BA+0x14	R	Floating Point Multiply Product Data 2 Register	0x00
FPPROD3	MFA_BA+0x15	R	Floating Point Multiply Product Data 3 Register	0x00

Bits	Flag	Description
[7:0]	FPMULPRODUCTx, x = 0, 1, 2, 3	**Floating-Point Multiply Product Data** This byte is set by hardware within 4-T.

MFA Floating-Point Multiply Status Register

Register	Offset	RW	Description	Reset Value
FPSTAT	MFA_BA+0x16	R	Floating Point Multiply Status Register	0x00

Bits	Flag	Description
[7]	Reserved	**Reserved**.
[6]	HUGEINT	Integer result after rounding has a magnitude greater than the largest representable 2's complement integer with the same sign.
[5]	INEXACT	floating-point multiply output is not equal to the infinitely precise result.
[4]	HUGE	Rounded floating-point number with unbounded exponent has a magnitude greater than the maximum normalized number.
[3]	TINY	Rounded floating-point number with unbounded exponent has a magnitude less than the minimum normalized number.
[2]	INVALID	Floating-point multiply operation is invalid.
[1]	INFINITY	Floating-point multiply output is infinity.
[0]	ZERO	Floating-point multiply output is zero.

MFA Floating-Point Log2 Input Data Register

Register	Offset	RW	Description	Reset Value
LOGIN0	MFA_BA+0x17	R/W	Floating Point Log2 Input Data 0 Register	0x00
LOGIN1	MFA_BA+0x18	R/W	Floating Point Log2 Input Data 1 Register	0x00
LOGIN2	MFA_BA+0x19	R/W	Floating Point Log2 Input Data 2 Register	0x00
LOGIN3	MFA_BA+0x1A	R/W	Floating Point Log2 Input Data 3 Register	0x00

Bits	Flag	Description
[7:0]	LOGINPUTx, x = 0, 1, 2, 3	**Floating-Point Log2 Input Data** This byte is set by software, and cleared by software.

MFA Floating-Point Log2 Output Data Register

Register	Offset	RW	Description	Reset Value
LOGOUT0	MFA_BA+0x1B	R	Floating Point Log2 Output Data 0 Register	0x00
LOGOUT1	MFA_BA+0x1C	R	Floating Point Log2 Output Data 1 Register	0x00
LOGOUT2	MFA_BA+0x1D	R	Floating Point Log2 Output Data 2 Register	0x00
LOGOUT3	MFA_BA+0x1E	R	Floating Point Log2 Output Data 3 Register	0x00

Bits	Flag	Description
[7:0]	LOGOUTPUTx, x = 0, 1, 2, 3	**Floating-Point Log2 Output Data** This byte is set by hardware within 4-T.

MFA Floating-Point Log2 Status Register

Register	Offset	RW	Description	Reset Value
LOGSTAT	MFA_BA+0x1F	R	Floating Point Log2 Status Register	0x00

Bits	Flag	Description
[7]	Reserved	**Reserved**.
[6]	HUGEINT	Integer result after rounding has a magnitude greater than the largest representable 2's complement integer with the same sign.
[5]	INEXACT	floating-point multiply output is not equal to the infinitely precise result.
[4]	HUGE	Rounded floating-point number with unbounded exponent has a magnitude greater than the maximum normalized number.
[3]	TINY	Rounded floating-point number with unbounded exponent has a magnitude less than the minimum normalized number.
[2]	INVALID	Floating-point multiply operation is invalid.
[1]	INFINITY	Floating-point multiply output is infinity.
[0]	ZERO	Floating-point multiply output is zero.

6.16.3 Programming Model (Please refer to “Sample code project”)

16- bit Multiply

Step 1 :

Set SIGN bit in MULCTRL Register.

Step 2 :

Set 16-bit multiplier data with most significant byte in MULP1 Register and least significant byte in MULP0 Register.

Step 3 :

Set 16-bit multiplicand data with most significant byte in MULC1 Register and least significant byte in MULC0 Register.

Step 4 :

SIGN = 0

1. Define global variables of unsigned char mul_data_char and unsigned long int mul_data_lint with the same SRAM address.
   unsigned char xdata mul_data_char[4] _at_ 0x8000;
   unsigned long int xdata mul_data_lint _at_ 0x8000;
2. Read multiply product data with most significant byte in MULPROD3 Register and least significant byte in MULPROD0 Register, and stores the MULPRODx, x = 0, 1, 2, 3, data to mul_data_char[4].
   mul_data_char[3] = MFA->MULPROD[0];
   mul_data_char[2] = MFA->MULPROD[1];
   mul_data_char[1] = MFA->MULPROD[2];
   mul_data_char[0] = MFA->MULPROD[3];
3. User can use mul_data_lint directly for further application computing.

SIGN = 1

1. Define global variables of unsigned char mul_data_char and signed long int mul_data_lint with the same SRAM address.
   unsigned char xdata mul_data_char[4] _at_ 0x8000;
   signed long int xdata mul_data_lint _at_ 0x8000;
2. Read multiply product data with most significant byte in MULPROD3 Register and least significant byte in MULPROD0 Register, and stores the MULPRODx, x = 0, 1, 2, 3, data to mul_data_char[4].
   mul_data_char[3] = MFA->MULPROD[0];
   mul_data_char[2] = MFA->MULPROD[1];
   mul_data_char[1] = MFA->MULPROD[2];
   mul_data_char[0] = MFA->MULPROD[3];

3. User can use mul_data_lint directly for further application computing.

Floating-Point Multiply

Step 1 :

Set ROUND field in FPCTRL Register.

Step 2 :

Set floating-point multiply multiplier data with most significant byte in FPMULP3 Register and least significant byte in FPMULP0 Register.

Step 3 :

Set floating-point multiply multiplicand data with most significant byte in FPMULC3 Register and least significant byte in FPMULC0 Register.

Step 4 :

1. Define global variables of unsigned char mul_data_char and float mul_data_fp with the same SRAM address.
   unsigned char xdata mul_data_char[4] _at_ 0x8000;
   float xdata mul_data_fp _at_ 0x8000;
2. Read floating-point multiply product data with most significant byte in MULPROD3 Register and least significant byte in MULPROD0 Register, and stores the MULPRODx, x = 0, 1, 2, 3, data to mul_data_char[4].
   mul_data_char[3] = MFA->FPPROD[0];
   mul_data_char[2] = MFA->FPPROD[1];
   mul_data_char[1] = MFA->FPPROD[2];
   mul_data_char[0] = MFA->FPPROD[3];
3. User can use mul_data_ fp directly for further application computing.

Floating-Point Log2

Step 1 :

Set floating-point log2 input data with most significant byte in LOGIN3 Register and least significant byte in LOGIN0 Register.

Step 2 :

1. Define global variables of unsigned char log2_data_char and float log2_data_fp with the same SRAM address.
   unsigned char xdata log2_data_char[4] _at_ 0x8000;
   float xdata log2_data_fp _at_ 0x8000;
2. Read log2 output data with most significant byte in LOGOUT3 Register and least significant byte in LOGOUT0 Register, and stores the LOGOUTx, x = 0, 1, 2, 3, data to log2_data_char[4].
   log2_data_char[3] = MFA->LOGOUT[0];

```
log2_data_char[2] = MFA->LOGOUT[1];
log2_data_char[1] = MFA->LOGOUT[2];
log2_data_char[0] = MFA->LOGOUT[3];
```

3. User can use log2_data_ fp directly for further application computing.

6.17 Capture (CAP)

6.17.1 Overview

The FE81 series supports a Capture (CAP) controller which can be used for sampling the pulse width of external input signal. The CAP controller supports 4-channel with each 4 sampling modes.

Features of Capture Controller

- 4 channels for input capture function
- 4 capture modes of edge, rising to falling, falling to rising, falling to falling
- Input signal filter function

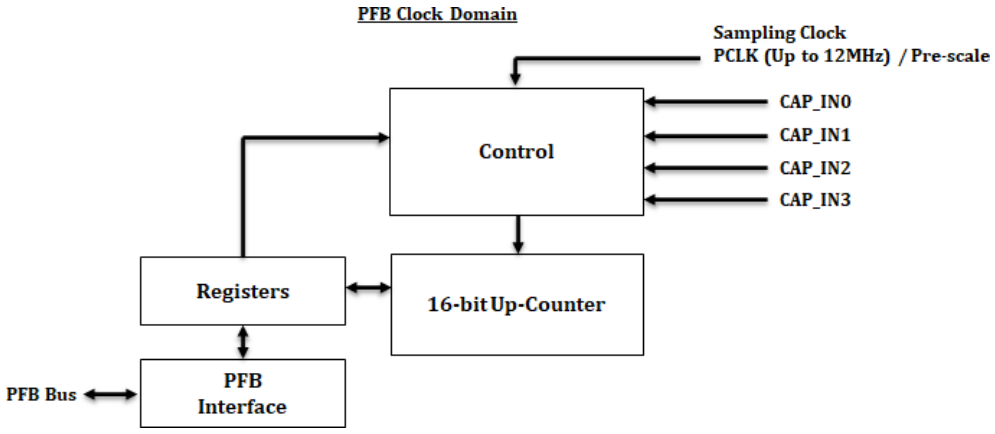


Figure 6.17-1 Capture Controller Block Diagram

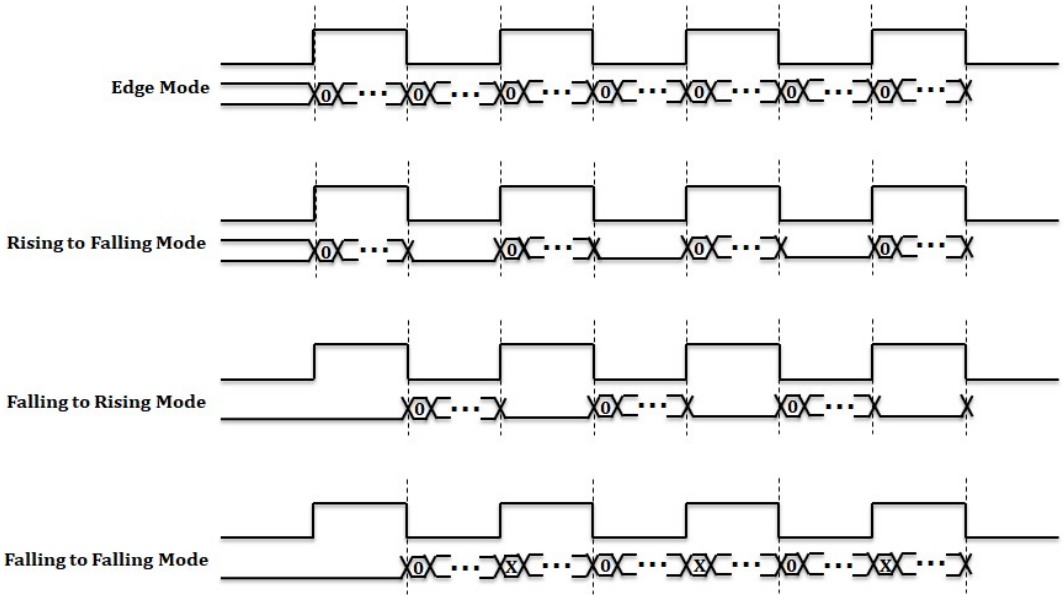


Figure 6.17-2 Capture Mode

Sampling Clock

The clock source of CAP controller is PFB clock (PCLK) and is able to be divided by PRESCALE Register depicted in Section 6.17.2.

Capture edge mode

While input signal toggling, the CAP counter value is preserved, and is auto-zero by hardware without software setting overhead. While edge mode interrupt is enabled, the counter value can be accessed directly by application in ISR for counting the “high pulse width” and “low pulse width” of the input signal.

Capture rising to falling mode

While input signal transitions from low to high, the CAP counter is auto-zero and starting counting. While input signal transition from high to low, the CAP counter value is preserved, and interrupt signal is asserted if interrupt is enabled. Application code can access the CAP counter in ISR for counting the “high pulse width” of the input signal.

Capture falling to rising mode

While input signal transitions from high to low, the CAP counter is auto-zero and starting counting. While input signal transition from low to high, the CAP counter value is preserved, and interrupt signal is asserted if interrupt is enabled. Application code can access the CAP counter in ISR for counting the “low pulse width” of the input signal.

Capture falling to falling mode

While input signal transitions from high to low, the CAP counter value is preserved, and is auto-zero by hardware. While interrupt is enabled, the interrupt signal is asserted, and application code can access the CAP counter in ISR for counting the “high + low pulse width” of the input signal.

Input Signal Filter

The CAP input signal is filtered according to CAPxLPF field, x = 0, 1, 2, 3, defined in LPFSEL Register, which is used for glitch immunization for stability of the input signal.

Capture Input Signal

The capture input signal is pin-shared with GPIO which is illustrated as follows.

CAP Pin	GPIO Pin
CAP_IN0	PD0
CAP_IN1	PD1
CAP_IN2	PD2
CAP_IN3	PD3

6.17.2 Register Map and Description

Base Address (CAP_BA) : 0xFE00				
Register	**Offset**	**RW**	**Description**	**Reset Value**
PRESCALE	CAP_BA+0x00	R/W	CAP Pre-scale Register	0x00
LPFSEL	CAP _BA+0x01	R/W	CAP Low Pass Filter Selection Register	0x00
MODE	CAP _BA+0x02	R/W	CAP Mode Selection Register	0x00
INTEN	CAP _BA+0x03	R/W	CAP Interrupt Enable Register	0x00
CATCTRL	CAP _BA+0x04	R/W	CAP Control Register	0x00
CAP0CNTL	CAP _BA+0x06	R	CAP Channel 0 Counter Low Register	0x00
CAP0CNTH	CAP _BA+0x07	R	CAP Channel 0 Counter High Register	0x00
CAP1CNTL	CAP _BA+0x08	R	CAP Channel 1 Counter Low Register	0x00
CAP1CNTH	CAP _BA+0x09	R	CAP Channel 1 Counter High Register	0x00
CAP2CNTL	CAP _BA+0x0A	R	CAP Channel 2 Counter Low Register	0x00
CAP2CNTH	CAP _BA+0x0B	R	CAP Channel 2 Counter High Register	0x00
CAP3CNTL	CAP _BA+0x0C	R	CAP Channel 3 Counter Low Register	0x00
CAP3CNTH	CAP _BA+0x0D	R	CAP Channel 3 Counter High Register	0x00
CAPINTF	CAP _BA+0x0F	R	CAP Interrupt Flag Register	0x00

Capture Pre-Scale Register

Register	Offset	RW	Description	Reset Value
PRESCALE	CAP_BA+0x00	R/W	CAP Pre-scale Register	0x00

Bits	Flag	Description
[7:0]	PRESCALE	**Capture Counter Clock Pre-Scale Flag** This field is set by software, cleared by software. **Sampling Clock = (PCLK) / (PRESCALE + 1)**

Capture Low Pass Filter Selection Register

Register	Offset	RW	Description	Reset Value
LPFSEL	CAP _BA+0x01	R/W	CAP Low Pass Filter Selection Register	0x00

Bits	Flag	Description
[7:6]	CAP3LPF	**CAP Channel 3 Input Signal Low Pass Filter Number** This field is set by software, cleared by software. CAP3LPF = 0, 1 consecutive input signal sampling without glitch. CAP3LPF = 1, 2 consecutive input signal sampling without glitch. CAP3LPF = 2, 3 consecutive input signal sampling without glitch. CAP3LPF = 3, 4 consecutive input signal sampling without glitch.
[5:4]	CAP2LPF	**CAP Channel 2 Input Signal Low Pass Filter Number** This field is set by software, cleared by software. CAP2LPF = 0, 1 consecutive input signal sampling without glitch. CAP2LPF = 1, 2 consecutive input signal sampling without glitch. CAP2LPF = 2, 3 consecutive input signal sampling without glitch. CAP2LPF = 3, 4 consecutive input signal sampling without glitch.
[3:2]	CAP1LPF	**CAP Channel 1 Input Signal Low Pass Filter Number** This field is set by software, cleared by software. CAP1LPF = 0, 1 consecutive input signal sampling without glitch. CAP1LPF = 1, 2 consecutive input signal sampling without glitch. CAP1LPF = 2, 3 consecutive input signal sampling without glitch. CAP1LPF = 3, 4 consecutive input signal sampling without glitch.
[1:0]	CAP0LPF	**CAP Channel 0 Input Signal Low Pass Filter Number** This field is set by software, cleared by software. CAP0LPF = 0, 1 consecutive input signal sampling without glitch. CAP0LPF = 1, 2 consecutive input signal sampling without glitch. CAP0LPF = 2, 3 consecutive input signal sampling without glitch. CAP0LPF = 3, 4 consecutive input signal sampling without glitch.

Capture Mode Selection Register

Register	Offset	RW	Description	Reset Value
MODE	CAP _BA+0x02	R/W	CAP Mode Selection Register	0x00

Bits	Flag	Description
[7:6]	CAP3MODE	**CAP Channel 3 Mode Selection Flag** This field is set by software, cleared by software. CAP3MODE = 0, Edge toggle. CAP3MODE = 1, Rising to Falling. CAP3MODE = 2, Falling to rising. CAP3MODE = 3, Falling to falling.
[5:4]	CAP2MODE	**CAP Channel 2 Mode Selection Flag** This field is set by software, cleared by software. CAP2MODE = 0, Edge toggle. CAP2MODE = 1, Rising to Falling. CAP2MODE = 2, Falling to rising. CAP2MODE = 3, Falling to falling.
[3:2]	CAP1MODE	**CAP Channel 1 Mode Selection Flag** This field is set by software, cleared by software. CAP1MODE = 0, Edge toggle. CAP1MODE = 1, Rising to Falling. CAP1MODE = 2, Falling to rising. CAP1MODE = 3, Falling to falling.
[1:0]	CAP0MODE	**CAP Channel 0 Mode Selection Flag** This field is set by software, cleared by software. CAP0MODE = 0, Edge toggle. CAP0MODE = 1, Rising to Falling. CAP0MODE = 2, Falling to rising. CAP0MODE = 3, Falling to falling.

Capture Interrupt Enable Register

Register	Offset	RW	Description	Reset Value
INTEN	CAP _BA+0x03	R/W	CAP Interrupt Enable Register	0x00

Bits	Flag	Description
[7]	CAP3OIEN	**CAP Channel 3 Counter Overflow Interrupt Enable Flag** This bit is set by software, cleared by software. CAP3OIEN = 0, Idle. CAP3OIEN = 1, Counter overflow.
[6]	CAP3IEN	**CAP Channel 3 Counting Ready Interrupt Enable Flag** This bit is set by software, cleared by software. CAP3IEN = 0, Idle. CAP3IEN = 1, Counting ready while mode condition is satisfied.
[5]	CAP2OIEN	**CAP Channel 2 Counter Overflow Interrupt Enable Flag** This bit is set by software, cleared by software. CAP2OIEN = 0, Idle. CAP2OIEN = 1, Counter overflow.
[4]	CAP2IEN	**CAP Channel 2 Counting Ready Interrupt Enable Flag** This bit is set by software, cleared by software. CAP2IEN = 0, Idle. CAP2IEN = 1, Counting ready while mode condition is satisfied.
[3]	CAP1OIEN	**CAP Channel 1 Counter Overflow Interrupt Enable Flag** This bit is set by software, cleared by software. CAP1OIEN = 0, Idle. CAP1OIEN = 1, Counter overflow.
[2]	CAP1IEN	**CAP Channel 1 Counting Ready Interrupt Enable Flag** This bit is set by software, cleared by software. CAP1IEN = 0, Idle. CAP1IEN = 1, Counting ready while mode condition is satisfied.
[1]	CAP0OIEN	**CAP Channel 0 Counter Overflow Interrupt Enable Flag** This bit is set by software, cleared by software. CAP0OIEN = 0, Idle. CAP0OIEN = 1, Counter overflow.
[0]	CAP0IEN	**CAP Channel 0 Counting Ready Interrupt Enable Flag** This bit is set by software, cleared by software. CAP0IEN = 0, Idle. CAP0IEN = 1, Counting ready while mode condition is satisfied.

Capture Control Register

Register	Offset	RW	Description	Reset Value
CAPCTRL	CAP _BA+0x04	R/W	CAP Control Register	0x00

Bits	Flag	Description
[7]	CAP3EN	**CAP Channel 3 Enable Flag** This bit is set by software, cleared by software. CAP3EN = 0, Idle. CAP3EN = 1, Capture channel 0 is enabled.
[6]	CAP2EN	**CAP Channel 2 Enable Flag** This bit is set by software, cleared by software. CAP2EN = 0, Idle. CAP2EN = 1, Capture channel 0 is enabled.
[5]	CAP1EN	**CAP Channel 1 Enable Flag** This bit is set by software, cleared by software. CAP1EN = 0, Idle. CAP1EN = 1, Capture channel 0 is enabled.
[4]	CAP0EN	**CAP Channel 0 Enable Flag** This bit is set by software, cleared by software. CAP0EN = 0, Idle. CAP0EN = 1, Capture channel 0 is enabled.
[3:0]	Reserved	**Reserved.**

Capture Control Register

Register	Offset	RW	Description	Reset Value
CAP0CNTL	CAP _BA+0x06	R	CAP Channel 0 Counter Low Register	0x00
CAP0CNTH	CAP _BA+0x07	R	CAP Channel 0 Counter High Register	0x00

Register	Offset	RW	Description	Reset Value
CAP1CNTL	CAP _BA+0x08	R	CAP Channel 1 Counter Low Register	0x00
CAP1CNTH	CAP _BA+0x09	R	CAP Channel 1 Counter High Register	0x00

Register	Offset	RW	Description	Reset Value
CAP2CNTL	CAP _BA+0x0A	R	CAP Channel 2 Counter Low Register	0x00
CAP2CNTH	CAP _BA+0x0B	R	CAP Channel 2 Counter High Register	0x00

Register	Offset	RW	Description	Reset Value
CAP3CNTL	CAP _BA+0x0C	R	CAP Channel 3 Counter Low Register	0x00
CAP3CNTH	CAP _BA+0x0D	R	CAP Channel 3 Counter High Register	0x00

Bits	Flag	Description
[7:0]	CAPxCNTL, x = 0, 1, 2, 3	**CAP Channel x Counter Low Byte Value** This bit is set by hardware.

Bits	Flag	Description
[7:0]	CAPxCNTH, x = 0, 1, 2, 3	**CAP Channel x Counter High Byte Value** This bit is set by hardware.

Capture Interrupt Flag Register

Register	Offset	RW	Description	Reset Value
CAPINTF	CAP _BA+0x0F	R	CAP Interrupt Flag Register	0x00

Bits	Flag	Description
[7]	CAP3OF	**CAP Channel 3 Counter Overflow Interrupt Flag (Read for Clear)** This bit is set by hardware, cleared by software. CAP3OF = 0, Idle. CAP3OF = 1, Counter overflow interrupt flag.
[6]	CAP3F	**CAP Channel 3 Counting Ready Interrupt Flag (Read for Clear)** This bit is set by hardware, cleared by software. CAP3F = 0, Idle. CAP3F = 1, Counting ready interrupt flag.
[5]	CAP2OF	**CAP Channel 2 Counter Overflow Interrupt Flag (Read for Clear)** This bit is set by hardware, cleared by software. CAP2OF = 0, Idle. CAP2OF = 1, Counter overflow interrupt flag.
[4]	CAP2F	**CAP Channel 2 Counting Ready Interrupt Flag (Read for Clear)** This bit is set by hardware, cleared by software. CAP2F = 0, Idle. CAP2F = 1, Counting ready interrupt flag.
[3]	CAP1OF	**CAP Channel 1 Counter Overflow Interrupt Flag (Read for Clear)** This bit is set by hardware, cleared by software. CAP1OF = 0, Idle. CAP1OF = 1, Counter overflow interrupt flag.
[2]	CAP1F	**CAP Channel 1 Counting Ready Interrupt Flag (Read for Clear)** This bit is set by hardware, cleared by software. CAP1F = 0, Idle. CAP1F = 1, Counting ready interrupt flag.
[1]	CAP0OF	**CAP Channel 0 Counter Overflow Interrupt Flag (Read for Clear)** This bit is set by hardware, cleared by software. CAP0OF = 0, Idle. CAP0OF = 1, Counter overflow interrupt flag.
[0]	CAP0F	**CAP Channel 0 Counting Ready Interrupt Flag (Read for Clear)** This bit is set by hardware, cleared by software. CAP0F = 0, Idle. CAP0F = 1, Counting ready interrupt flag.

6.17.3 Programming Model (Please refer to “Sample code project”)

Step 1: Set clock pre-scale register by PRESCALE Register

Step 2: Set CAPxLPF, x = 0, 1, 2, 3, of input signal filter in LPFSEL Register

Step 3: Set capture mode by CAPxMODE, x = 0, 1, 2, 3, field of Mode Register

Step 4: Set interrupt enable flag by CAPxEN, x = 0, 1, 2, 3, field of INTEN Register

Step 5: Enable capture channel by CAPxEN, x = 0, 1, 2, 3, field of CAPCTRL Register

Step 6: Wait for Interrupt

Step 7: In capture ISR, read CAPINTF Register for clearing related interrupt flag and read related channel counter value by CAPxCNTL and CAPxCNTH, x = 0, 1, 2, 3 Register