十速

TM52F3288

DATA SHEET

Rev V0.90

tenx reserves the right to change or discontinue the manual and online documentation to this product herein to improve reliability, function or design without further notice. **tenx** does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. **tenx** products are not designed, intended, or authorized for use in life support appliances, devices, or systems. If Buyer purchases or uses tenx products for any such unintended or unauthorized application, Buyer shall indemnify and hold tenx and its officers, employees, subsidiaries, affiliates and distributors harmless against all claims, cost, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use even if such claim alleges that tenx was negligent regarding the design or manufacture of the part.

AMENDMENT HISTORY

Version	Date	Description
0.90	Apr, 2018	New release.

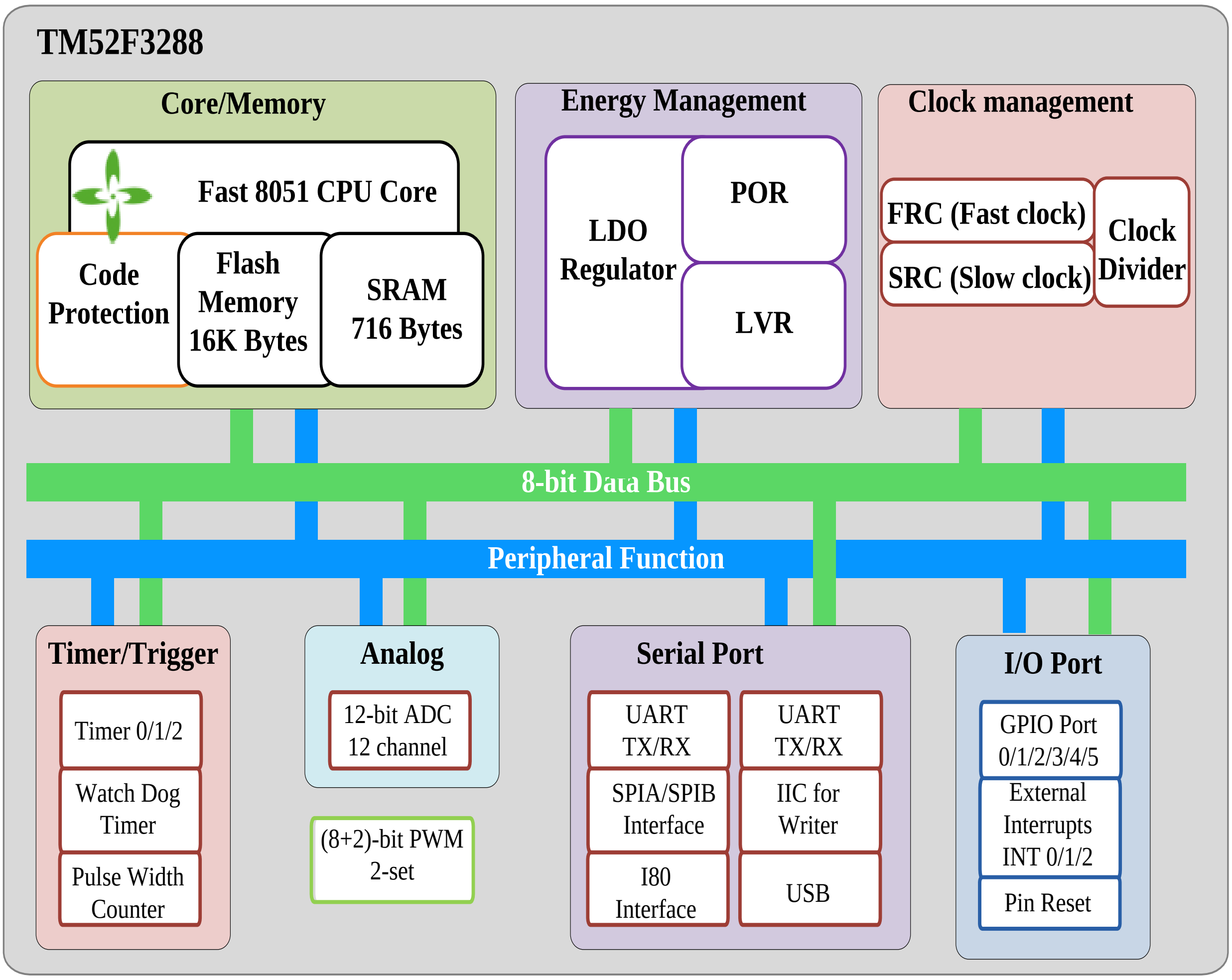
CONTENTS

GENERAL DESCRPTION

TM52F3288 is version of a new, fast 8051 architecture for an 8-bit microcontroller single chip with an instruction set fully compatible with industry standard 8051, C language development platform, and retains most 8051 peripheral function block and contain the USB full speed general purpose application. Typically, the TM52 executes instruction six times faster than the traditional 8051 architecture.

The **TM52F3288** provides improved performance, lower cost and fast time-to-market by integrating features on the chip, including 16K Bytes Flash program memory, 716 Bytes SRAM, Low Voltage Reset (LVR), dual clock power saving operation mode, 2 set SPI Interface, 8051 standard UART and Timer0/1/2, 2 set (8+2)-bit PWMs, 12 channels 12-bit A/D Convert and Watchdog Timer. Its high reliability and lower power consummation feature can be widely applied in consummation and home appliance products. It can works in USB mode by using USB power or works in standalone mode by using battery power.

BLOCK DIAGRAM



FEATURES

1. **Standard 8051 Instruction Set, fast machine cycle**
   - Execute instructions six times faster than standard 8051
2. **16 K Bytes Flash Program Memory**
   - Support “In Circuit Programming” (ICP) or “In System Programming” (ISP) for the Flash Code
   - Byte Write “In Application Programming” (IAP) mode is convenient as Data EEPROM access
   - Code Protection Capability
3. **Total 716 Bytes SRAM (IRAM + XRAM)**
   - 256 Bytes IRAM in the 8051 internal data memory area
   - 460 Bytes XRAM in the 8051 external data memory area (accessed by MOVX Instruction)
4. **Two System Clock type Selection**
   - Fast clock from Internal RC (FRC, 11.0592 MHz)
   - Slow clock from Internal RC (SRC, 32KHz)
   - System clock can be divided by 1/2/4/16 option
5. **8051 Standard Timer -- Timer0/1/2**
   - 16-bit Timer0, also supports T0O clock output for Buzzer application
   - 16-bit Timer1
   - 16-bit Timer2, also supports T2O clock output for Buzzer application
6. **USB Interface**
   - Compliance with the Universal Serial Bus specification v2.0 Full Speed
   - Support USB Power Delivery specification R2.0
   - Built-in USB Transceiver 3.3V regulator
   - Endpoint 0: Control SETUP /IN/OUT transfer (each 8 bytes)
   - Endpoint 1: INTERRUPT IN transfer (8 bytes)
   - Endpoint 2: INTERRUPT IN transfer (8 bytes)
   - Endpoint 3: BULK IN transfer with Ping-Pong feature (2*64 bytes)
   - Endpoint 4: BULK OUT transfer with Ping-Pong feature (2*64 bytes)
   - USB PD transfer (38 bytes)
   - USB PD receiver (38 bytes)
7. **12-bit ADC convert with 12 input channels**
8. **8051 Standard UART**
   - One Wire UART option can be used for ISP or other application

9. Two SPI Interface

SPIA:

- Master or Slave mode selectable
- Programmable transmit bit rate
- Serial clock phase and polarity options
- MSB-first or LSB-first selectable

SPIB:

- Support Master only
- Clock rate up to 12Mbps
- Read/Write DMA mode
- Serial clock phase and polarity options
- MSB-first or LSB-first selectable

10. I80 Interface (Nand-Flash Interface)

- compatible with 8-bit parallel interface
- Read/Write DMA mode

11. Two independent “8+2” bits PWMs with prescaler/period-adjust

- PWM0:

  Width differential output pair

  Non-overlap durations adjustable
- PWM1:

  3 output with different duty but share the same period

12. Pin Interrupt can Wake up CPU from Power-Down (Stop) mode

- P3.2 / P3.3 (INT0 / INT1) Interrupt & Wake-up
- P4.7 (INT2) Interrupt & Wake-up
- Each Port1 pin can be defined as Interrupt & Wake-up pin (by pin change)

13. 24 Sources, 4-level priority Interrupt

- Timer0/Timer1/Timer2 Interrupt
- INT0/INT1 Falling-Edge/Low-Level Interrupt
- Port1 Pin Change Interrupt
- UART TX/RX Interrupt
- P4.7 (INT2) Interrupt
- ADC End of Conversion Interrupt
- SPIA Interrupt
- USB1 Interrupt: SET0I/OUT0I/TX0I/TX1I/TX2I/SUSPI/TX3I/RC4I

- USB2 Interrupt: VDD5VRI/RSTI/RSMI/KBDI/PD_TXI/PD_RCI

14. Max. 41 Programmable I/O pins

- CMOS Output
- Pseudo-Open-Drain, or Open-Drain Output
- Schmitt Trigger Input
- Pin Pull-up can be Enabled or Disabled
- P4.4 and P4.5 support high drive/sink current(Max. 40mA)

15. Independent RC Oscillating Watchdog Timer

- 400ms/200ms/100ms/50ms Selectable WDT Timeout options

16. Six type Reset

- Power on Reset
- Selectable External Pin Reset
- Selectable Watch Dog Reset
- Software Command Reset
- Selectable Battery Low Voltage Reset
- USB Plug-out Reset

17. 3-level Low Voltage Reset

- 2.0V/2.3V/2.9V

18. 4 Power Saving Operation Modes

- Fast / Slow / Idle / Stop Mode

19. On-chip Debug/ICE interface

- Use P1.2/P1.3 pin or P3.6/P3.7 pin (in USB application)
- Share with ICP programming pin (P1.2/P1.3)

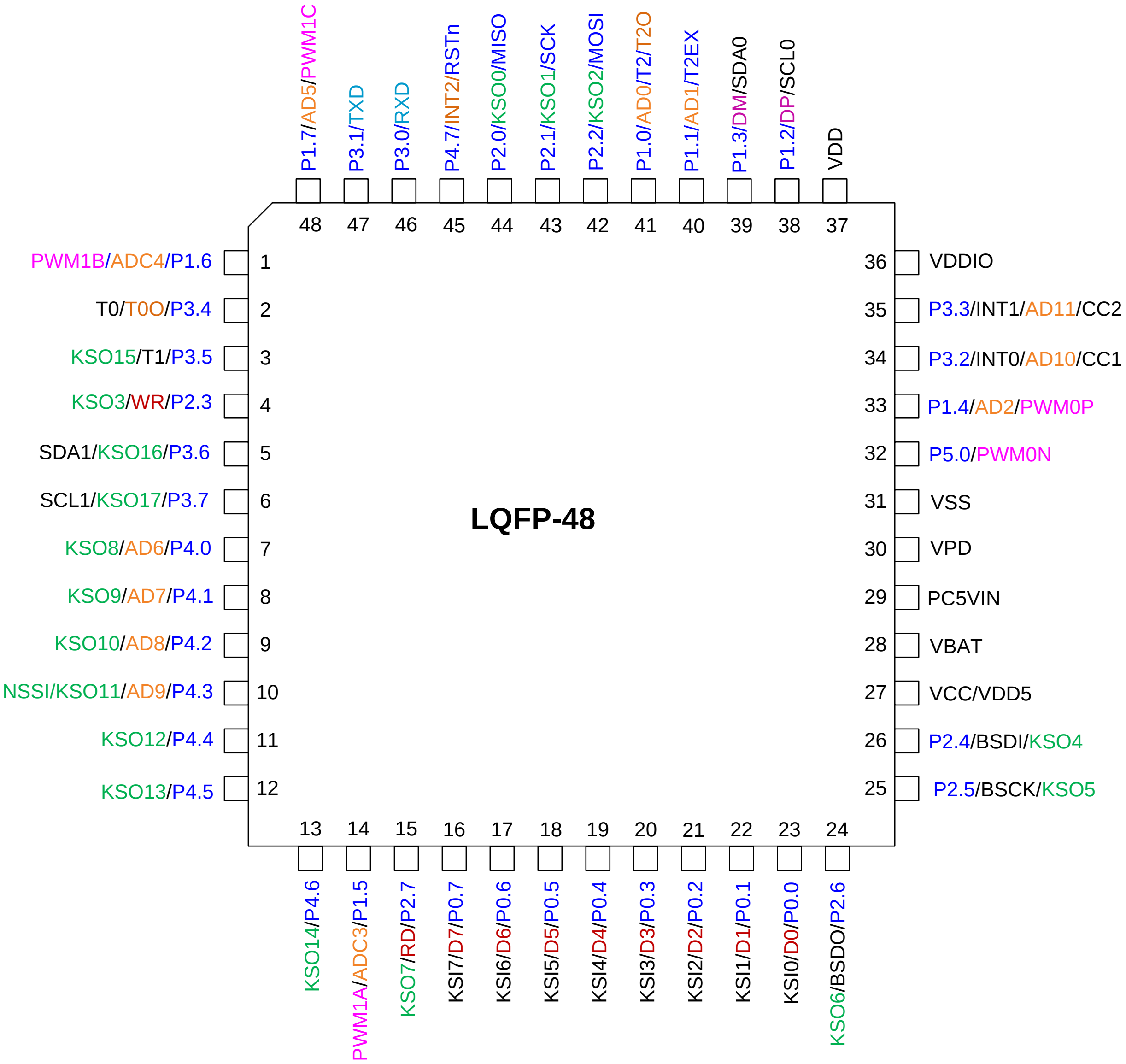
20. Operating Voltage: 4.5V~5.5V in USB application or 3.0~5.5 in non-USB application

21. Operating Temperature Range

- -40C ~+85C

22. 48-pin LQFP Package (7x7x1.4mm)

PIN ASSIGNMENT DIAGRAM



PIN DESCRIPTIONS

Name	In/Out	Pin Description
P0.0–P0.7	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output. Pull-up resistors are assignable by software.
P1.0~P1.7	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or “open-drain” output. Pull-up resistors are assignable by software. These pin’s level change can wake up CPU from Idle/Stop mode
P2.0~P2.7	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output. Pull-up resistors are assignable by software.
P3.0~P3.2	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or “pseudo open drain” output. Pull-up resistors are assignable by software.
P3.3~P3.7	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or “open-drain” output. Pull-up resistors are assignable by software.
P4.0~P4.7	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output or “open-drain” output. Pull-up resistors are assignable by software.
P5.0	I/O	Bit-programmable I/O port for Schmitt-trigger input, CMOS push-pull output. Pull-up resistors are assignable by software.
INT0, INT1	I	External Low level or falling edge Interrupt input, Idle/Stop mode wake up input
INT2	I	External falling edge Interrupt input, Idle/Stop mode wake up input
ADC11~ADC0	I	Analog to Digital Convert input pin
RXD	I/O	UART Mode0 transmit & receive data, Mode1/2/3 receive data
TXD	I/O	UART Mode0 transmit clock, Mode1/2/3 transmit data. IN One Wire UART mode, this pin transmits and receives serial data
T0, T1, T2	I	Timer0, Timer1, Timer2 event count pin input
T0O	O	Timer0 overflow divided by 64 output
T2O	O	Timer2 overflow divided by 2 output
T2EX	I	Timer2 external trigger input
PWM0P/PWM0N PWM1A/PWM1B/ PWM1C	O	8+2 bit PWM output
MISO	I/O	SPIA data input for master mode, data output for slave mode
MOSI	I/O	SPIA data output for master mode, data input for slave mode
SCK	I/O	SPIA clock output for master or clock input for slave mode
BSDI	I	SPIB data input
BSDO	O	SPIB data output
BSCK	O	SPIB clock out
D0~D7	I/O	I80 Data
RD	O	I80 Read signal output
WR	O	I80 Write signal output
KSI0~KSI7	I	Keyboard matrix input
KSO0~KSO17	O	Keyboard matrix output
PC5VIN	I	USB Power input pin
RSTn	I	External active low reset
VDD	P	3.3V output power generate by internal regulator; need connect 1uF to Ground
VDDIO	P	3.3V/5V input power for I/O port; need connect 1uF to Ground
VPD	P	1.1V output power generated by internal regulator; used for USB PD; need connect 1uF to Ground
VBAT	P	VBAT input power
VCC/VDD5, VSS	P	Power input pin and Ground

Pin Summary

Pin number	Pin Name	Type	Input			Output			Alternate Function							MISC
			Pull-up Control	Wake up	Ext. Interrupt	P.P.	P.O.D	O.D	PWM	ADC	UART	SPI	I80	KBD	Timer	
1	PWM1B/ADC4/P1.6	I/O	⊙	●		●		●	●	●						
2	T0/T0O/P3.4	I/O	⊙			●		●							●	
3	KSO15/T1/P3.5	I/O	⊙			●		●						●	●	
4	KSO3/WR/P2.3	I/O	◎			●							●	●		
5	SDA1/KDO16/P3.6	I/O	⊙			●		●						●		
6	SCL1/KSO17/P3.7	I/O	⊙			●		●						●		
7	KSO8/AD6/P4.0	I/O	⊙			●		●		●				●		
8	KSO9/AD7/P4.1	I/O	⊙			●		●		●				●		
9	KSO10/AD8/P4.2	I/O	⊙			●		●		●				●		
10	NSSI/KSO11/AD9/P4.3	I/O	⊙			●		●		●		●		●		
11	KSO12/ P4.4	I/O	⊙			●		●						●		
12	KSO13/ P4.5	I/O	⊙			●		●						●		
13	KSO14/ P4.6	I/O	⊙			●		●						●		
14	PWM1A/AD3/ P2.7	I/O	◎			●			●	●						
15	KSO7/RD/P2.7	I/O	◎			●							●	●		
16	KSI7/D7/P0.7	I/O	◎			●							●	●		
17	KSI6/D6/P0.6	I/O	◎			●							●	●		
18	KSI5/D5/P0.5	I/O	◎			●							●	●		
19	KSI4/D4/P0.4	I/O	◎			●							●	●		
20	KSI3/D3/P0.3	I/O	◎			●							●	●		
21	KSI2/D2/P0.2	I/O	◎			●							●	●		
22	KSI1/D1/P0.1	I/O	◎			●							●	●		
23	KSI0/D0/P0.0	I/O	◎			●							●	●		
24	KSO6/ BSDO /P2.6	I/O	◎			●						●		●		
25	KSO5/BSCK/P2.5	I/O	◎			●						●		●		
26	KSO4/BSDI/P2.4	I/O	◎			●						●		●		
27	VCC	P														
28	VBAT	P														
29	PC5VIN	I														
30	VPD	P														

31	VSS	P														
32	PWM0N/P5.0	I/O	◎			●			●							
33	PWM0P/AD2/P1.4	I/O	⊙	●		●		●	●	●						
34	CC1/AD10/INT0/P3.2	I/O	⊙	●	●	●	●			●						
35	CC2/AD11/INT1/P3.3	I/O	⊙	●	●	●		●		●						
36	VDDIO	P														
37	VDD	P														
38	SCL0/DP/P1.2	I/O	⊙	●		●		●								
39	SDA0/DM/P1.3	I/O	⊙	●		●		●								
40	T2EX/AD1/P1.1	I/O	⊙	●		●		●		●					●	
41	T2/T2O/AD0/P1.0	I/O	⊙	●		●		●		●					●	
42	AMOSI/KSO2/P2.2	I/O	◎			●						●		●		
43	ASCK/KSO1/P2.1	I/O	◎			●						●		●		
44	AMISO/KSO0/P2.0	I/O	◎			●						●		●		
45	RSTn/INT2/P4.7	I/O	⊙	●	●	●		●								
46	RXD/P3.0	I/O	⊙			●	●				●					
47	TXD/P3.1	I/O	⊙			●	●				●					
48	PWM1C/AD5/P1.7	I/O	⊙	●		●		●	●	●						

Symbol：

P.P. = Push-Pull Output
O.D. = Open Drain
P.O.D. = Pseudo Open Drain

PS:

1. ⊙ Port1, Port3, Port4 theses pins control Pull up resistor by operation modes
2. ◎ Port0, Port2, Port5 these pins control Pull up resistor while PxOE.n=0 and Px.n=1

Functional Description

1. CPU Core

In the 8051 architecture, the C programming language is used as a development platform. The **TM52F3288** features a fast 8051 core in a highly integrated microcontroller, allowing designers to be able to achieve improved performance compared to a classic 8051 device. TM52 series microcontrollers provide a complete binary code with standard 8051 instruction set compatibility, ensuring an easy migration path to accelerate the development speed of system products. The CPU core includes an ALU, a program status word (PSW), an accumulator (ACC), a B register, a stack point (SP), DPTRs, a programming counter, an instruction decoder, and core special function registers (SFRs).

1.1 Accumulator (ACC)

This register provides one of the operands for most ALU operations. Accumulators are generally referred to as A or Acc and sometimes referred to as Register A. In this document, the accumulator is represented as “A” or “ACC,” including the instruction table. The accumulator, as its name suggests, is used as a general register to accumulate the intermediate results of a large number of instructions. The accumulator is the most important and frequently used register to complete arithmetic and logical operations. It holds the intermediate results of most arithmetic and logic operations and assists in data transportation.

SFR **E0h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ACC	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

E0h.7~0 **ACC:** Accumulator

1.2 B Register (B)

The “B” register is very similar to the ACC and may hold a 1 Byte value. This register provides the second operand for multiply or divide instructions. Otherwise, it may be used as a scratch pad register. The B register is only used by two 8051 instructions, MUL and DIV. When A is to be multiplied or divided by another number, the other number is stored in B. For MUL and DIV instructions, it is necessary that the two operands be in A and B.

ex: DIV A,B

When this instruction is executed, data inside A and B are divided, and the answer is stored in A.

SFR **F0h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F0h.7~0 **B:** B register

1.3 Stack Pointer (SP)

The SP register contains the Stack Pointer. The Stack Pointer is used to load the program counter into memory during LCALL and ACALL instructions and is used to retrieve the program counter from memory in RET and RETI instructions. The stack may also be saved or loaded using PUSH and POP instructions, which also increment and decrement the Stack Pointer. The Stack Pointer points to the top location of the stack.

SFR **81h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SP	SP							
R/W	R/W							
Reset	0	0	0	0	0	1	1	1

81h.7~0 **SP:** Stack Point

1.4 Dual Data Pointer (DPTRs)

F2261/64 has two DPTRs, which share the same SFR address. Each DPTR is 16 bits in size and consists of two registers: the DPTR high byte (DPH) and the DPTR low byte (DPL). The DPTR is used for 16-bit-address external memory accesses, for offset code byte fetches, and for offset program jumps. Setting the DPSEL control bit allows the program code to switch between the two physical DPTRs.

SFR **82h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPL	DPL							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

82h.7~0 **DPL:** Data Point low byte

SFR **83h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DPH	DPH							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

83h.7~0 **DPH:** Data Point high byte

SFR **F8h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AUX1	CLRWDT	WDTE		ADSOC	CLRPWM0	PDDP	BOOTV	DPSEL
R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	u	0

F8h.0 **DPSEL:** Active DPTR Select

1.5 Program Status Word (PSW)

This register contains status information resulting from CPU and ALU operations. The PSW affected by instructions is listed below.

Instruction	Flag		
	C	OV	AC
ADD	X	X	X
ADDC	X	X	X
SUBB	X	X	X
MUL	0	X	
DIV	0	X	
DA	X		
RRC	X		
RLC	X		
SETB C	1		

Instruction	Flag		
	C	OV	AC
CLR C	0		
CPL C	X		
ANL C, bit	X		
ANL C. /bit	X		
ORL C, bit	X		
ORL C, /bit	X		
MOV C, bit	X		
CJNE	X		

A “0” means the flag is always cleared, a “1” means the flag is always set and an “X” means that the state of the flag depends on the result of the operation.

SFR **D0h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PSW	CY	AC	F0	RS1	RS0	OV	F1	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

D0h.7 **CY:** ALU carry flag

D0h.6 **AC:** ALU auxiliary carry flag

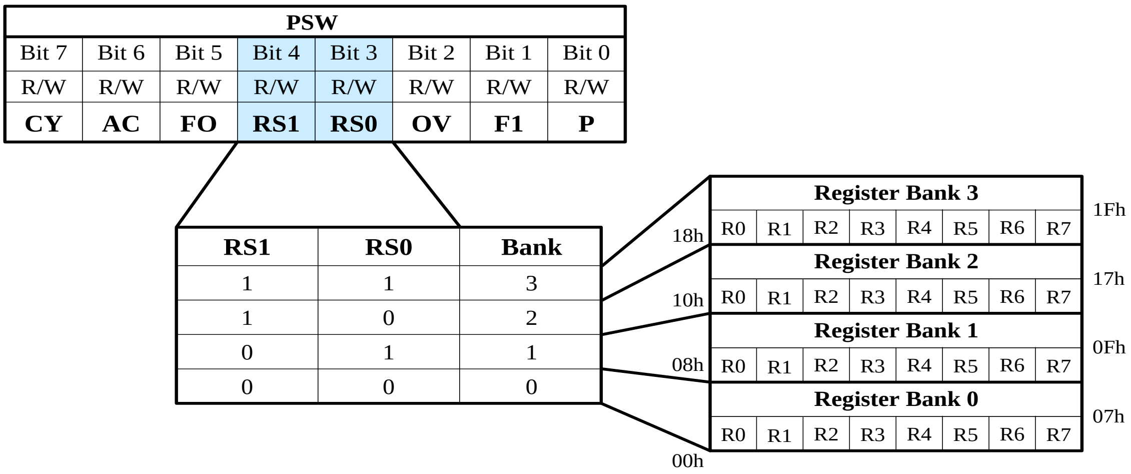
D0h.5 **F0:** General purpose user-definable flag

D0h.4~3 **RS1, RS0:** The contents of (RS1, RS0) enable the working register banks as:
00: Bank 0 (00h-07h)
01: Bank 1 (08h-0Fh)
10: Bank 2 (10h-17h)
11: Bank 3 (18h-1Fh)

D0h.2 **OV:** ALU overflow flag

D0h.1 **F1:** General purpose user-definable flag

D0h.0 **P:** Parity flag. Set/cleared by hardware each instruction cycle to indicate odd/even number of “one” bits in the accumulator.



PSW							
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
CY	**AC**	**FO**	**RS1**	**RS0**	**OV**	**F1**	**P**

RS1	**RS0**	**Bank**
1	1	3
1	0	2
0	1	1
0	0	0

2. Memory

2.1 Program Memory

The **F3288** has a 16K Bytes Flash program memory, which can support In Circuit Programming (ICP), In Application Programming (IAP) and In System Programming (ISP) function modes. The Flash write endurance is at least 50K cycle. The Flash program memory address continuous space (0000h–3FFFh) is partitioned to several sectors for device operation.

2.1.1 Program Memory Functional Partition

The last 4 bytes (3FFCh-3FFFh) of program memory is defined as chip Configuration Word (CFGW), which is loaded into the device control registers upon power on reset (POR). The address space 3F00h–3FFBh is the IAP free area, while the 0000h–005Fh is occupied by Reset/Interrupt vectors as standard 8051 definition. In the in-circuit emulation (ICE) mode, user also needs to reserve the address space 1D00h~1FFFh for ICE System communication.

16K Bytes program memory

0000h 005Fh	Reset/Interrupt Vector
0060h 1CFFh	User Code area
1D00h 1FFFh	ICE mode reserved area
2000h 3EFFh	User Code area
3F00h 3FFBh	IAP-Free area
3FFCh 3FFFh	CFGW

2.1.2 Flash ICP Mode

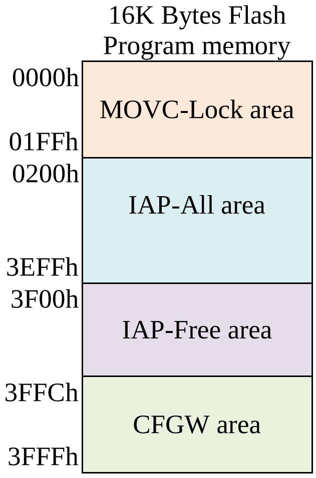
The Flash memory can be programmed by the tenx proprietary writer **(TWR98/TWR99)**, which needs at least four wires VCC, VSS, SCL and SDA pins to connect to this chip. If the user wants to program the Flash memory on the target circuit board (In Circuit Program, ICP), these pins must be reserved sufficient freedom to be connected to the Writer. More pins connected to Writer ensure more writing efficiency and speed. SCL and SDA can be P1.2 and P1.3 or can be P3.6 and P3.7.

Writer wire number	Pin connection
4-Wire	VCC, VSS, SCL, SDA
5-Wire	VCC, VSS, SCL, SDA, P1.0
7-Wire	VCC, VSS, SCL, SDA, P1.0, P1.1, P4.3

4-wire: Minimum program pin
5-Wire: add P1.0 (BUSY)
7-Wire: add P4.3 (TRIM), P1.1 (CKO)

2.1.3 Flash IAP Mode

The **F3288** has "In Application Program" (IAP) capability, which allows software to read/write data from/to the Flash memory during CPU run time as conveniently as data EEPROM access. The IAP function is byte writable, meaning that the **F3288** does not need to erase one Flash page before write. The available IAP data space is 254 Bytes after chip reset, and can be re-defined by the "MVCLOCK" and "IAPALL" control register as shown below.



Flash memory	MVCLOCK	IAPALL	MOVC Accessible	MOVX (IAP) Accessible
0000h~01FFh	1	X	No	No
	0	0	Yes	No
	0	1	Yes	Yes
0200h~3EFFh	X	0	Yes	No
	X	1	Yes	Yes
3F00h~3FFBh	X	X	Yes	Yes
3FFC~3FFEh	X	0	Yes	No
	X	1	Yes	Yes
3FFFh	X	X	Yes	No

In IAP mode, the program Flash memory is separated into four sectors: MOVC-Lock area, IAP-All area, IAP-Free area, and CFGW area. These four sectors are regulated differently.

In the **MOVC-Lock area,** IAP read/write is limited by MVCLOCK bit, which can be set to control the accessibility of the MOVC and MOVX instructions to this area. The size of this area is 512 Bytes. The lock function is made to protect the main program code against unconsciously writing Flash memory in IAP mode. Locking or unlocking the function should be performed by the tenx TWR98/99 writing to the CFGW in Flash memory.

The **IAP-All area** is protected by the IAPALL register to prevent IAP mode from writing application data to the program area, resulting in a program code error that cannot be repaired. The size of this area is 15,616 Bytes. Enabling IAPALL requires writing 65h to SFR SWCMD 97h to set the IAPALL control flag. Then, software can use MOVX instructions to write application data to flash memory from 0200h to 3EFFh. If user wants to disable IAPALL function, user can write other values to SFR SWCMD 97h to clear the IAPALL control flag. User must be careful not to overwrite program code which is already resided on the same Flash memory area.

The **IAP-Free area** has no control bit to protect. It can be used to reliably store system application data that needs to be programmed once or periodically during system operation. Other areas of Flash memory can be used to store data, but this area is usually the best. The size of this area is 252 Bytes, equivalent to an EEPROM, and Flash memory can provide byte access to read and write commands. In the past, storage of configuration data required an additional EEPROM or the other storage device. However, this functionality can now be provided by on-chip Flash, reducing the chip count of embedded applications. An external EEPROM or SRAM may not be needed.

The **CFGW area** has 4 data bytes (CFGWH1, CFGWH2, CFGWL1 and CFGWL2), which is located at the last 4 addresses of Flash memory. The CFGWH2 is not accessible to IAP, while the others can be read or written by IAP in case the IAPALL flag is set. CFGWH1(FRC Trimming value) is copied to the

SFR F7h after power on reset, software then take over CFGWH1's control capability by modifying the SFR F7h. CFGWL1(VBG Trimming value) is copied to the SFR F6h after power on reset, software then take over CFGWL1's control capability by modifying the SFR F6h. The CFGWL2 is reserved. BOOTV (CFGWH2[0]) bit is also copied to the SFR F8h[1].

2.1.4 IAP Mode Access Routines

Flash IAP write is simply achieved by a "MOVX @DPTR, A" instruction while the DPTR contains the target Flash address (0–3FFEH), and the ACC contains the data being written. The F3288 accepts IAP Write command only when the IAPWE SFR is enabled. Flash IAP writing requires approximately 500uS. Meanwhile, the CPU stays in a waiting state, but all peripheral modules (Timers and others) continue running during the writing time. The software must handle the pending interrupts after an IAP write. Flash IAP writing needs higher VCC voltage, VCC>2.8V.

Because the Program memory and the IAP data space share the same entity, a Flash IAP read can be performed by the "MOVX A, @DPTR" or "MOVC" instruction as long as the target address points to the 0~3FFFh area. A Flash IAP read does not require extra CPU wait time.

```
 ;  IAP example code
; need VCC > 2.8V
mov      dptr, #3f00h      ; dptr = 3f00h = target IAP address
mov      a, #5ah           ; a = 5ah = target IAP write data
mov      IAPWE, #47h       ; IAPWE SFR write 47h to enable IAPWE
movx     @dptr, a          ; Flash[3f00h] = 5ah,  after IAP write
                            ; 200us~500us H/W writing time, CPU wait
                            ;
mov      IAPWE, #00h       ; IAPWE = 0 immediately after IAP write
clr      a                  ; a= 0
movx     a, @dptr          ; a= 5ah
clr      a,                 ; a= 0
movc     a, @a+dptr        ; a= 5ah
```

Flash **3FFFh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CFGWH2	PROT	XRSTE	LVRE		-	PWRSAV	MVCLOCK	BOOTV

3FFFh.1 **MVCLOCK:** If 1, the MOVC & MOVX instruction's accessibility to MOVC-Lock area is limited.

3FFFh.0 BOOT vector select
0: Boot from ROM address 0000h
1: Boot from ROM address 3000h

SFR **97h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SWCMD	IAPALL / SWRST							
R/W	W							R/W
Reset	–							0

97h.7~0 **IAPALL (W):** Write 65h to set IAPALL control flag; Write other value to clear IAPALL flag.

97h.0 **IAPALL (R):** Flag indicates Flash memory sectors can be accessed by IAP or not. This bit combines with MVCLOCK to define the accessible IAP area.

SFR **C9h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IAPWE	IAPWE	IAPTO	-					
R/W	W/R	R	-					
Reset	0	0	-					

C9h.7 **IAPWE (W):** Write 47h to set IAPWE control flag; Write other value to clear IAPWE flag.

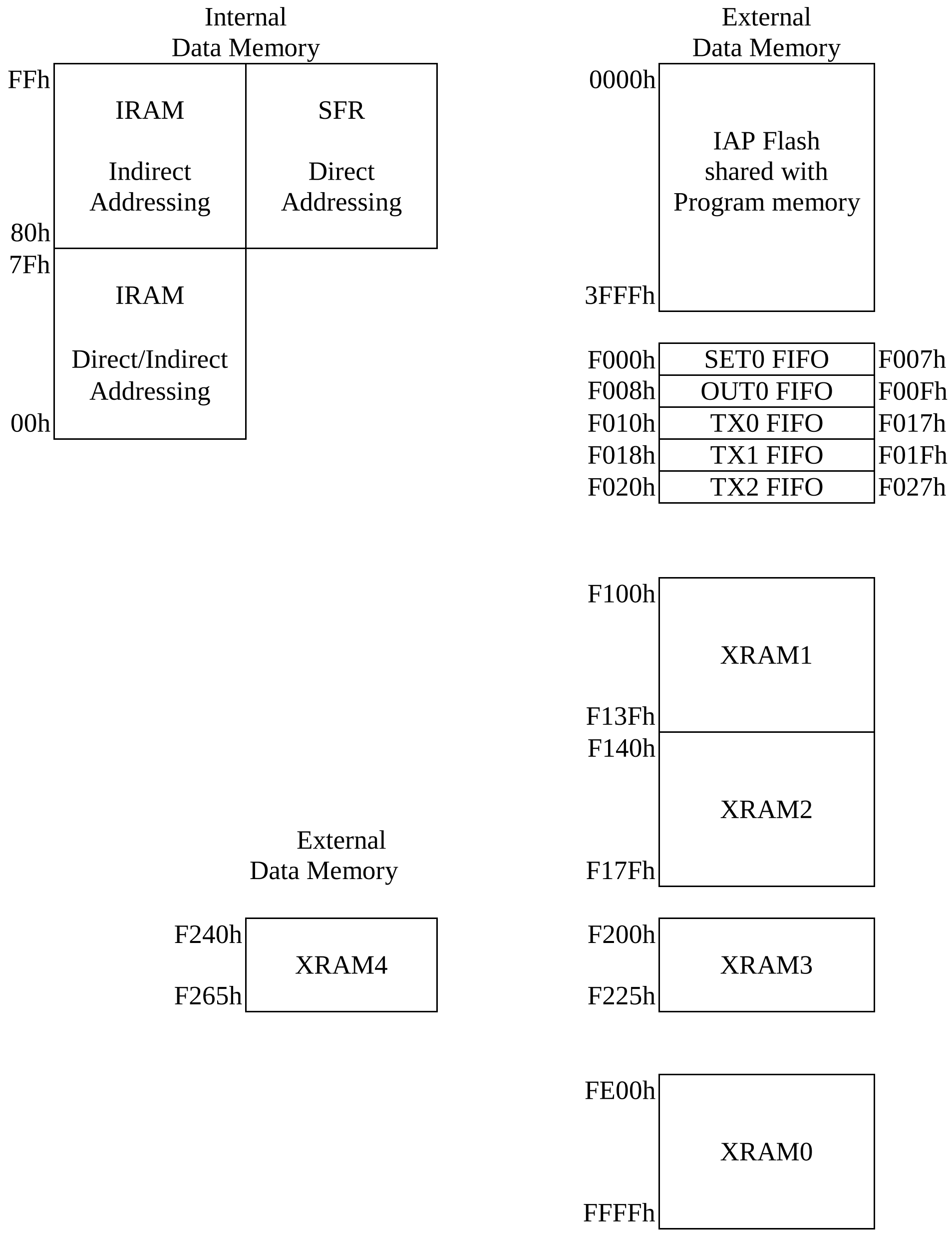
C9h.7 **IAPWE (R):** IAPWE flag

2.1.5 Flash ISP Mode

The "In System Program" (ISP) usage is similar to IAP, except the purpose is to refresh the Program code. User can use UART/SPI or other method to get new Program code from external host, then writes code as the same way as IAP. ISP operation is complicated; basically it needs to assign a Boot code area to the Flash which does not change during the ISP process. When BOOTV is set to high, the Boot address is changed to 3000h after chip reset. If BOOTV = 0, the Boot address is 0000h after chip reset.

2.2 Data Memory

As the standard 8051, the **F3288** has both Internal and External Data Memory space. The Internal Data Memory space consists of 256 Bytes IRAM and 63 SFRs, which are accessible through a rich instruction set. The External Data Memory space consists of 460 Bytes XRAM and IAP Flash, which can be only accessed by MOVX instruction.



2.2.1 IRAM

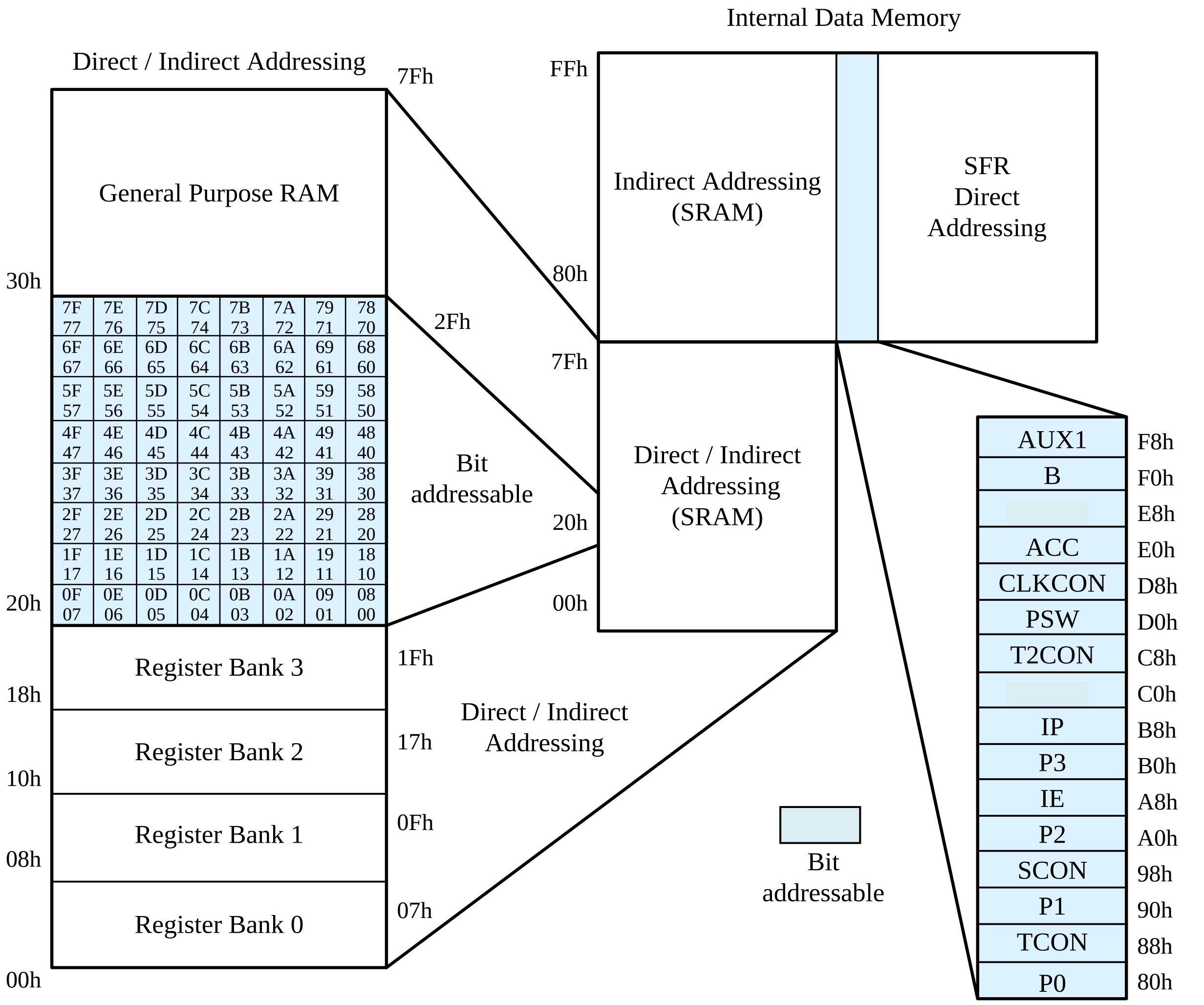
IRAM is located in the 8051 internal data memory space. The whole 256 Bytes IRAM are accessible using indirect addressing but only the lower 128 Bytes are accessible using direct addressing. There are four directly addressable register banks (switching by PSW), which occupy IRAM space from 0x00 to 0x1F. The address 0x20 to 0x2F 16 Bytes IRAM space is bit-addressable. IRAM can be used as scratch pad registers or program stack.

2.2.2 XRAM

XRAM0 is located in the 8051 external data memory space (address from FF00h to FFFFh). The 256 Bytes XRAM can be only accessed by "MOVX" instruction. XRAM1/2 is located in the 8051 external data memory space (address from F100h to F17Fh). The two 64 Bytes XRAM can be accessed by "MOVX" instruction and can be access by USB, SPI and I80 hardware DMA module. XRAM3/4 is located in the 8051 external data memory space (address from F200h to F225 and F240h to F265h). The two 38 Bytes XRAM can be accessed by "MOVX" instruction and can be access by USB PD module.

2.2.3 SFRs

All peripheral functional modules such as I/O ports, Timers and UART operations for the chip are accessed via Special Function Registers (SFRs). These registers occupy upper 128 Bytes of direct Data Memory space locations in the range 0x80 to 0xFF. There are 14 bit addressable SFRs (which means that eight individual bits inside a single byte are addressable), such as ACC, B register, PSW, TCON, SCON, and others. The remaining SFRs are only byte addressable. SFRs provide control and data exchange with the resources and peripherals of the **F3288**. The TM52 series of microcontrollers provides complete binary code with standard 8051 instruction set compatibility. Beside the standard 8051 SFRs, the **F3288** implements additional SFRs used to configure and access subsystems such as the USB/SPI, which are unique to the **F3288**.



7F 77	7E 76	7D 75	7C 74	7B 73	7A 72	79 71	78 70
6F 67	6E 66	6D 65	6C 64	6B 63	6A 62	69 61	68 60
5F 57	5E 56	5D 55	5C 54	5B 53	5A 52	59 51	58 50
4F 47	4E 46	4D 45	4C 44	4B 43	4A 42	49 41	48 40
3F 37	3E 36	3D 35	3C 34	3B 33	3A 32	39 31	38 30
2F 27	2E 26	2D 25	2C 24	2B 23	2A 22	29 21	28 20
1F 17	1E 16	1D 15	1C 14	1B 13	1A 12	19 11	18 10
0F 07	0E 06	0D 05	0C 04	0B 03	0A 02	09 01	08 00

	8/0	9/1	A/2	B/3	C/4	D/5	E/6	F/7
F8h	AUX1							
F0h	B					AUX2	CFGWL1	CFGWH1
E8h	P4							ANAT
E0h	ACC							
D8h	CLKCON		USB1IE	USB2IE	USBRSTM			PDCTRL
D0h	PSW	TX3CNT	RC4CNT	RC0REG	I80CTRL		XRAMCTRL	
C8h	T2CON	IAPWE	RCP2L	RCP2H	TL2	TH2	TX3CTRL	RC4CTRL
C0h	P5	USBADR	USB1I	USB2I	USBCTRL	TX0CTRL	TX1CTRL	TX2CTRL
B8h	IP	IPH	IP1	IP1H	SPACON	SPASTA	SPADAT	
B0h	P3	SPBCTRL	SPBCRS	DMALEN	P4MODL	P4MODH	KBMASK	
A8h	IE	INTE1	ADCDL	ADCDH	PWMDTYL		CHSEL	
A0h	P2	PWMCON	P1MODL	P1MODH	P3MODL	P3MODH	PINMOD	PWM0MODT
98h	SCON	SBUF	PWM0PRD	PWM0DTYH	PWM1PRD	PWM1ADTYH	PWM1BDTYH	PWM1CDTYH
90h	P1	P0OE		P2OE	OPTION	INTFLG	P1WKUP	SWCMD
88h	TCON	TMOD	TL0	TL1	TH0	TH1		
80h	P0	SP	DPL	DPH				PCON

3. Power

VCC pin is the main power supply for this chip. When MODE3V=0, the voltage regulator output 3.3V (VDD) power to the internal chip circuit. When MODE3V=1, the LDO is turned off, and the internal circuit receives a power supply directly from the VCC pin. Because the LDO consumes 150uA for operation, turning off LDO by setting MODE3V=1 can reduce the chip current consumption. However, setting MODE3V=1 is only valid for an operation of VCC < 3.6V. The PWRSAV also control the LDO. When MODE3V=0 and PWRSAV=1, the LDO is turned off in STOP mode for saving power consumption. In addition, set PWRSAV will affect the LVR setting. When **F3288** is used for USB application, the LDO output must keep at 3.3V; the LDO cannot be turned off in STOP mode for saving power consumption.

MODE3V=0

Operation Mode	**CFGWH2**		**LDO**	**Function**
	PWRSAV	**LVRE**		
Fast Slow Idle	X	00	ON	LV Reset 2.9V
	X	01	ON	LV Reset 2.3V
	X	10	ON	LV Reset 2.3V
	X	11	ON	LV Reset 2.1V
STOP	0	00	ON	LV Reset 2.9V
	0	01	ON	LV Reset 2.3V
	0	10	ON	LV Reset 2.3V
	0	11	ON	LV Reset 2.1V
	1	00	OFF	LV Reset 2.9V
	1	01	OFF	LV Reset 2.3V
	1	10	OFF	LV Reset 2.3V
	1	11	OFF	LV Reset 2.1V

MODE3V=1

Operation Mode	**CFGWH2**		**LDO**	**Function**
	PWRSAV	**LVRE**		
Fast Slow Idle	0	00	OFF	LV Reset 2.9V
	0	01	OFF	LV Reset 2.3V
	0	10	OFF	LV Reset 2.3V
	0	11	OFF	LV Reset 2.1V
	1	00	OFF	LV Reset 2.1V
	1	01	OFF	LV Reset 2.1V
	1	10	OFF	LV Reset 2.1V
	1	11	OFF	LV Reset 2.1V
STOP	0	00	OFF	LV Reset 2.9V
	0	01	OFF	LV Reset 2.3V
	0	10	OFF	LV Reset 2.3V
	0	11	OFF	LV Reset 2.1V
	1	00	OFF	LV Reset 2.1V
	1	01	OFF	LV Reset 2.1V
	1	10	OFF	LV Reset 2.1V
	1	11	OFF	LV Reset 2.1V

Flash **3FFFh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CFGWH2	PROT	XRSTE	LVRE		-	PWRSAV	MVCLOCK	BOOTV

3FFFh.2 **PWRSAV:** Power saving function control bit

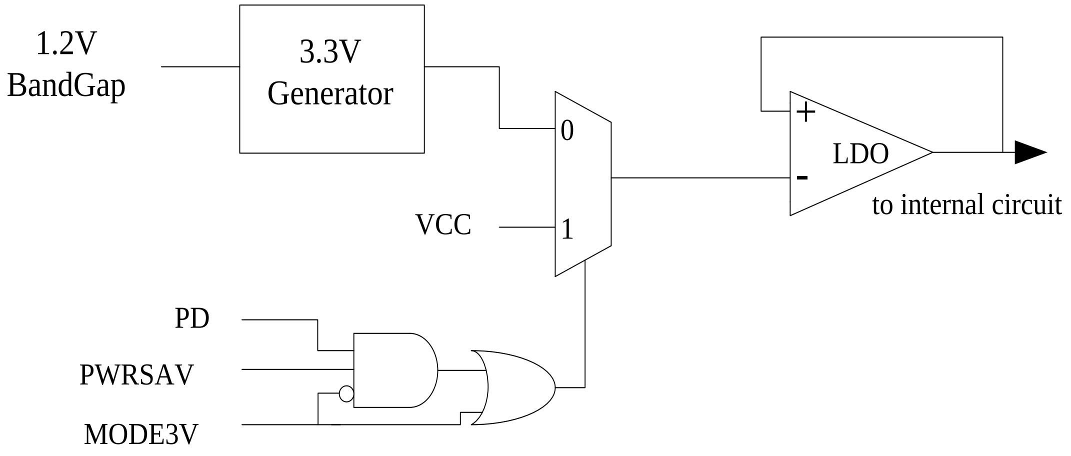
0: Disable Power saving function

1: Enable Power saving function

SFR **94h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	UART1W	MODE3V	WDTPSC		ADCKS		-	
R/W	R/W	R/W	R/W		R/W		-	
Reset	0	0	0	0	0	0	-	

94h.6 **MODE3V:**3V mode selection control bit

If this bit is set, the chip can be only operated in the condition od VCC < 3.6V, and LDO is turned off to save current



4. Reset

The **F3288** has six types of reset method. Resets can caused by Power on Reset (POR), External Pin Reset (XRST), Software Command Reset (SWRST), Watchdog Timer Reset (WDTR), Low Voltage Reset (LVR), or USB power plug-out/plug-in reset. The CFGW controls the Reset functionality. The SFRs are returned to their default value after Reset.

4.1 Power on Reset

After Power on Reset, the device stays on Reset state for 24 ms as chip warm up time, then downloads the CFGW register from ROM‘s last byte (Other Reset will not reload the CFGW). The Power on Reset needs VCC pin’s voltage first discharge to near VSS level, then rise beyond 2.1V.

4.2 External Pin Reset

Pin Reset is active low. It needs to keep at least 2 SRC clock cycle long to be seen by the chip. Pin Reset can be disabled or enabled by CFGWH2.

4.3 Software Reset

Software Reset is activated by writing the SFR 97h with data 56h.

4.4 Watch Dog Timer Reset (WDT)

WDT overflow Reset is disabled or enable by SFR F8h. The WDT uses SRC as its counting time base. It runs in Fast / Slow mode and runs or stops in Idle/ Stop mode. WDT overflow speed can be defined by WDTPSC SFR. WDT is cleared by device Reset or CLRWDT SFR bit.

4.5 Low Voltage Reset

The **F3288** offers three options for LVR function. The user can make a selection by CFGWH2, let LVR voltage of 2.9V, 2.3V and 1.9V be selected separately. The LVR can be disabled or enabled by CFGWH2.

4.6 USB Plug Reset

The **F3288** support USB application. USB cable plugs in or out can cause chip reset. By checking the register VDD5VFLG status, the F/W can determine **F3288** is working in USB or non-USB application. When VDD5VFLG is high, which means USB cable plugs in PC5V power supply to the chip, **F3288** can work in USB application. IF VDD5VFLG is low, **F3288** can work in non-USB application.

System Clock Frequency	12 MHz	6 MHz	3MHz	0.75MHz
Minimum LVR level	LVR=2.9V	LVR=2.9V	LVR=2.3V	LVR=2.1V

SFR **F8h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AUX1	CLRWDT	WDTE		ADSOC	CLRPWM0	PDADOP	BOOTV	DPSEL
R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F8h.7 **CLRWDT**: Set to clear WDT, H/W auto clear it at next clock cycle

F8h.6~5 **WDTE:** Watchdog Timer Reset control
0x: Watchdog Timer Reset disable
10: Watchdog Timer Reset enable in Fast/Slow mode, disable in Idle/Stop mode
11: Watchdog Timer Reset always enable

Flash **3FFFh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CFGWH2	PROT	XRSTE	LVRE		-	PWRSAV	MVCLOCK	BOOTV

3FFFh.6 **XRSTE:** External Pin Reset control
0: Disable External Pin Reset
1: Enable External Pin Reset

3FFFh.5~4 **LVRE:** Low Voltage Reset function select
00: Set LVR at 2.9V
01: Set LVR at 2.3V
10: Reserved
11: Set LVR at 1.9V

SFR **94h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	UART1W	MODE3V	WDTPSC		ADCKS		-	
R/W	R/W	R/W	R/W		R/W		-	
Reset	0	0	0	0	0	0	-	

94h.5~4 **WDTPSC:** Watchdog Timer pre-scaler time select
00: 400ms WDT overflow rate
01: 200ms WDT overflow rate
10: 100ms WDT overflow rate
11: 50ms WDT overflow rate

SFR **97h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SWCMD	IAPALL / SWRST							
R/W	W							R/W
Reset	–							0

97h.7~0 **SWRST:** Write 56h to generate S/W Reset.

SFR **F5h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AUX2	VDD5VFLG	VDD5FALL	–			IAPTE		VCCFLT
R/W	R	R/W	–			R/W		R/W
Reset	0	0	–			0		0

F5h.7 **VDD5VFLG:** PC5V status
0: PC5V Low
1: PC5V High (USB plug in)

F5h.6 **VDD5FALL:** USB Plug out flag
write 0 or power on reset to clear flag

5. Clock Circuitry & Operation Mode

5.1 System Clock

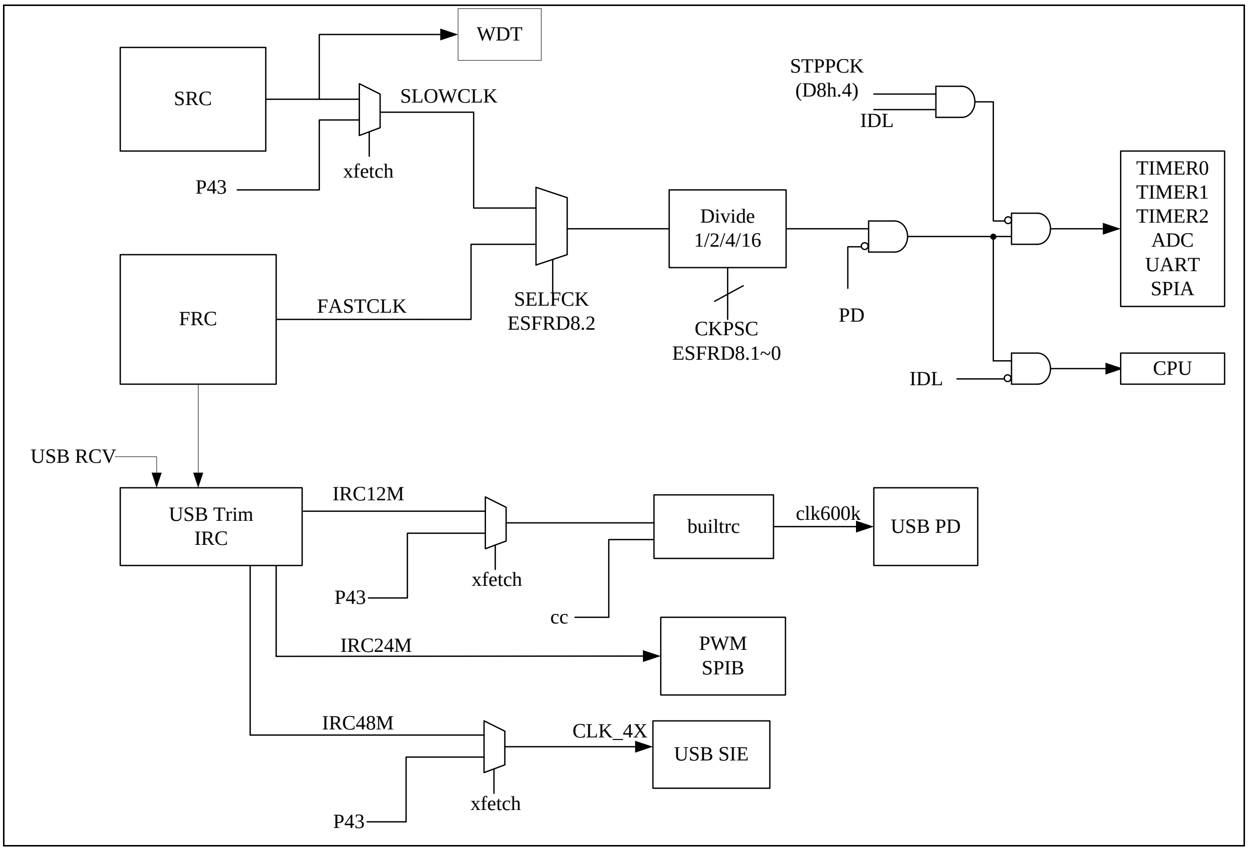
The **F3288** is designed with dual-clock system. During runtime, user can directly switch the System clock from Fast to Slow or from slow to fast. It also can directly select a clock divider of 1, 2, 4 or 16. The Fast clock is fixed to FRC (Fast Internal RC, 11.0592 MHz at VCC=5V). The Slow clock is SRC (Slow Internal RC, 32KHz). Fast mode and Slow mode are defined as the CPU running at Fast and Slow clock speeds.

After Reset, the device is running at Slow mode with 32KHz SRC. The user should select the proper clock rate for chip operation safety. The higher VCC allows the chip to run at a higher System clock frequency.

The **CLKCON** SFR controls the System clock operating. H/W automatically blocks the S/W abnormally setting for this register. Never to write both STPFCK=1 and SELFCK=1. It is recommended to write this SFR bit by bit.

5.2 USB Clock

The F3288 incorporates flexible internal oscillator and clock generators, including a 24 MHz accurate to 3% over temperature and voltage. The 24 MHz can also be doubled to 48 MHz for USB application, and the Internal RC clock generators will self-tune to ±0.25% accuracy when USB communication.



Clock Structure

5.3 Operation Mode

There are four operation modes for this device. **Fast Mode** is defined as the CPU running at Fast clock speed. **Slow Mode** is defined as the CPU running at Slow clock speed. When the System clock speed is lower, the power consumption is lower.

Idle Mode is entered by setting the IDL bit in PCON SFR. Both Fast and Slow clock can be set as the System clock source in Idle Mode, but Slow clock is better for power saving. In Idle mode, the CPU puts itself to sleep while the on-chip peripherals stay active. The STPPCK bit in CLKCON SFR can be set to furthermore reduce Idle mode current. If STPPCK=1, Timer0/1/2, ADC and UART are stopped in Idle mode. The slower System clock rate also helps current saving. It can be achieved by setup the CLKPSC SFR to divide System clock frequency. Idle mode is terminated by Reset or enabled Interrupts wake up.

Stop Mode is entered by setting the PD bit in PCON SFR. This mode is the so-called "Power Down" mode in standard 8051. In Stop mode, all clocks stop except the WDT is alive if it is enabled. Stop mode can be terminated by Reset or pin wake up.

Note: Chip cannot enter Stop Mode if INTn pin is low and wakeup is enable. (INTn=0 and EXn=1, n=0,1,2)

SFR **87h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	SMOD	–	–	–	GF1	GF0	PD	IDL
R/W	R/W	–	–	–	R/W	R/W	R/W	R/W
Reset	0	–	–	–	0	0	0	0

87h.1 **PD:** Stop bit. If 1 Stop mode is entered.

87h.0 **IDL:** Idle bit. If 1, Idle mode is entered.

SFR **D8h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CLKCON	-	-	STPSCK	STPPCK	STPFCK	SELFCK	CLKPSC	
R/W	-	-	R/W	R/W	R/W	R/W	R/W	
Reset	-	-	0	0	0	0	1	1

D8h.5 **STPSCK:** Set 1 to stop Slow clock

D8h.4 **STPPCK:** Set 1 to stop UART/Timer0/Timer1/Timer2/ADC clock in Idle mode

D8h.3 **STPFCK:** Set 1 to stop Fast clock for power saving in Slow/Idle mode.
This bit can be changed only in Slow mode.

D8h.2 **SELFCK:** System clock source selection. This bit can be changed only when STPFCK=0.
0: Slow clock
1: Fast clock

D8h.1~0 **CLKPSC:** System clock prescaler.
00: System clock is Fast/Slow clock divided by 16
01: System clock is Fast/Slow clock divided by 4
10: System clock is Fast/Slow clock divided by 2
11: System clock is Fast/Slow clock divided by 1

6. Interrupt and Wake-up

This **F3288** has a 25-source four-level priority interrupt structure. All enabled Interrupts can wake up CPU from Idle mode, but only the Pin Interrupts can wake up CPU from Stop mode. Each interrupt source has its own enable control bit. An interrupt event will set its individual Interrupt Flag, no matter whether its interrupt enable control bit is 0 or 1. The Interrupt vectors and flags are list below.

Vector	**Flag**	**Description**
0003	IE0	INT0 external pin Interrupt (can wake up Stop mode)
000B	TF0	Timer0 Interrupt
0013	IE1	INT1 external pin Interrupt (can wake up Stop mode)
001B	TF1	Timer1 Interrupt
0023	RI+TI	Serial Port (UART) Interrupt
002B	TF2+EXF2	Timer2 Interrupt
0033	NMI	Reserved for ICE mode use
003B	SPAI	SPIA Interrupt
0043	P1IF	Port1 external pin change Interrupt (can wake up Stop mode)
004B	IE2	INT2 external pin Interrupt (can wake up Stop mode)
0053	ADIF	ADC Interrupt
005B	USB1I	USB1 Interrupt Vector (8 sources)
0063	USB2I	USB2 Interrupt Vector (6 sources)

Interrupt Vector & Flag

6.1 Interrupt Enable and Priority Control

The IE and INTE1 SFRs decide whether the pending interrupt is serviced by CPU. The IP, IPH, IP1 and IP1H SFRs decide the interrupt priority. An interrupt will be serviced as long as an interrupt of equal or higher priority is not already being serviced. If an interrupt of equal or higher level priority is being serviced, the new interrupt will wait until it is finished before being serviced. If a lower priority level interrupt is being serviced, it will be stopped and the new interrupt serviced. When the new interrupt is finished, the lower priority level interrupt that was stopped will be completed.

SFR **96h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1WKUP	P1WKUP							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

96h.7~0 **P1WKUP:** P1.7~P1.0 pin individual Wake up/Interrupt enable control
0: Disable
1: Enable

SFR **A8h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IE	EA	–	ET2	ES	ET1	EX1	ET0	EX0
R/W	R/W	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	–	0	0	0	0	0	0

A8h.7 **EA:** Global interrupt enable
0: Disable all interrupts
1: Each interrupt is enabled or disabled by its individual interrupt control bit

A8h.5 **ET2:** Timer2 interrupt enable
0: Disable Timer2 interrupt
1: Enable Timer2 interrupt

A8h.4 **ES:** Serial Port (UART) interrupt enable
0: Disable Serial Port (UART) interrupt
1: Enable Serial Port (UART) interrupt

A8h.3 **ET1:** Timer1 interrupt enable
0: Disable Timer1 interrupt
1: Enable Timer1 interrupt

A8h.2 **EX1:** INT1 pin Interrupt enable and Stop mode wake up enable
0: Disable INT1 pin Interrupt and Stop mode wake up
1: Enable INT1 pin Interrupt and Stop mode wake up, it can wake up CPU from Stop mode no matter EA is 0 or 1.

A8h.1 **ET0:** Timer0 interrupt enable
0: Disable Timer0 interrupt
1: Enable Timer0 interrupt

A8h.0 **EX0:** INT0 pin Interrupt enable and Stop mode wake up enable
0: Disable INT0 pin Interrupt and Stop mode wake up
1: Enable INT0 pin Interrupt and Stop mode wake up, it can wake up CPU from Stop mode no matter EA is 0 or 1.

SFR **A9h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTE1	-	-	U2IE	U1IE	ADIE	EX2	P1IE	SPAIE
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	-	0	0	0	0	0	0

A9h.5 **U2IE:** USB2 interrupt enable
0: Disable USB2 interrupt
1: Enable USB2 interrupt

A9h.4 **U1IE:** USB1 interrupt enable
0: Disable USB1 interrupt
1: Enable USB1 interrupt

A9h.3 **ADIE:** ADC interrupt enable
0: Disable ADC interrupt
1: Enable ADC interrupt

A9h.2 **EX2:** INT2 pin Interrupt enable and Stop mode wake up enable
0: Disable INT2 pin Interrupt and Stop mode wake up
1: Enable INT2 pin Interrupt and Stop mode wake up, it can wake up CPU from Stop mode no matter EA is 0 or 1.

A9h.1 **P1IE:** Port1 pin change interrupt enable
0: Disable Port1 pin change interrupt
1: Enable Port1 pin change interrupt

A9h.0 **SPAIE:** SPIA interrupt enable
0: Disable SPIA interrupt
1: Enable SPIA interrupt

SFR **B9h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IPH	–	–	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	0	0	0	0	0	0

SFR **B8h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IP	–	–	PT2	PS	PT1	PX1	PT0	PX0
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	0	0	0	0	0	0

B9h.5, B8h.5 **PT2H, PT2:** Timer2 interrupt priority control. (PT2H, PT2) =
00: Level 0 (lowest priority)
01: Level 1
10: Level 2
11: Level 3 (highest priority)
B9h.4, B8h.4 **PSH, PS:** Serial Port (UART) interrupt priority control. Definition as above.
B9h.3, B8h.3 **PT1H, PT1:** Timer1 interrupt priority control. Definition as above.
B9h.2, B8h.2 **PX1H, PX1:** INT1 pin interrupt priority control. Definition as above.
B9h.1, B8h.1 **PT0H, PT0:** Timer0 interrupt priority control. Definition as above.
B9h.0, B8h.0 **PX0H, PX0:** INT0 pin interrupt priority control. Definition as above.

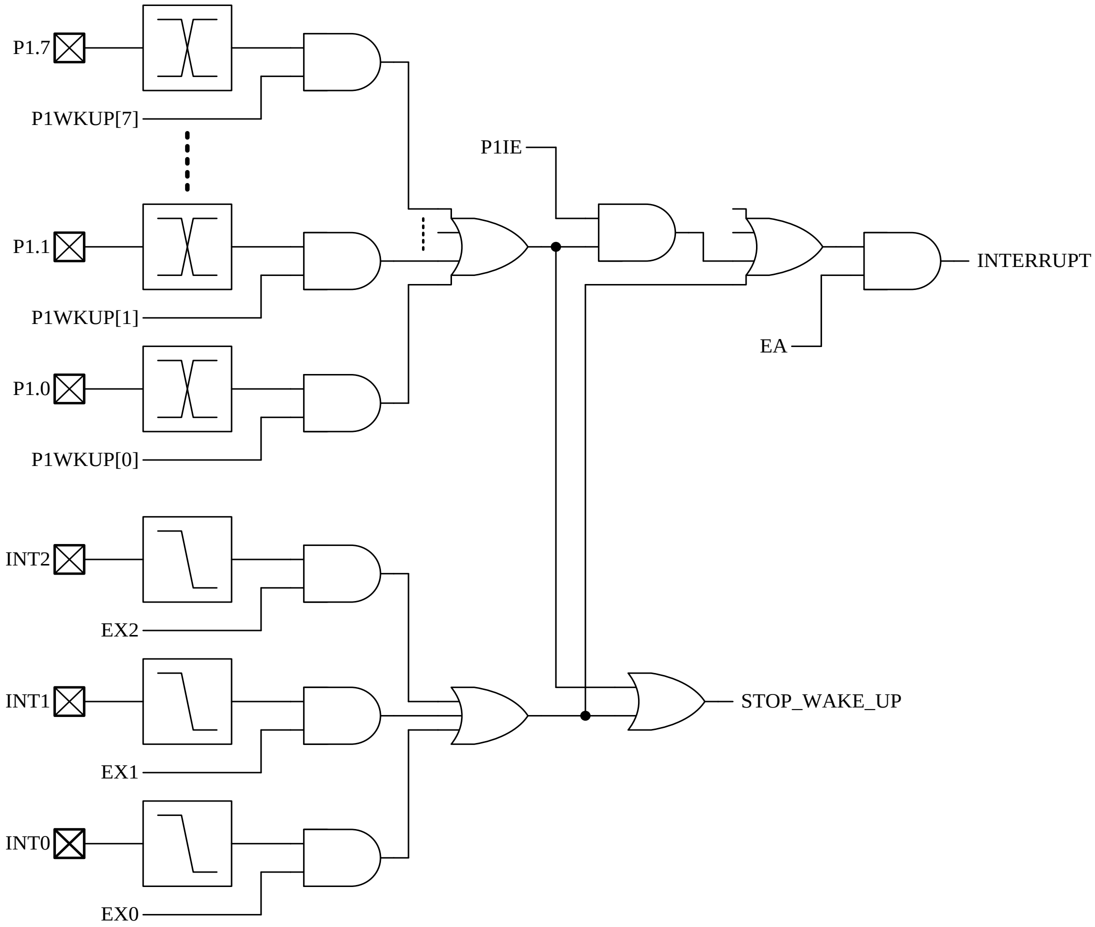
SFR **BBh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IP1H	–	–	PUSB2IH	PUSB1IH	PADIH	PX2H	PP1H	PSPAIH
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	0	0	0	0	0	0

SFR **BAh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IP1	–	–	PUSB2I	PUSB1I	PADI	PX2	PP1	PSPAI
R/W	–	–	R/W	R/W	R/W	R/W	R/W	R/W
Reset	–	–	0	0	0	0	0	0

BBh.5, BAh.5 **PUSB2IH, PUSB2I:** USB2 interrupt priority control. Definition as above.
BBh.4, BAh.4 **PUSB1IH, PUSB1I:** USB1 interrupt priority control. Definition as above.
BBh.3, BAh.3 **PADIH, PADI:** ADC interrupt priority control. Definition as above.
BBh.2, BAh.2 **PX2H, PX2:** INT2 pin interrupt priority control. Definition as above.
BBh.1, BAh.1 **PP1H, PP1:** Port1 pin change interrupt priority control. Definition as above.
BBh.0, BAh.0 **PSPAIH, PSPAI:** SPIA interrupt priority control. Definition as above.

6.2 Pin Interrupt

Pin Interrupts include INT0 (P3.2), INT1 (P3.3), INT2 (P4.7) and Port1 Change Interrupt. These pins also have the Stop mode wake up capability. INT0 and INT1 are falling edge or low level triggered as the 8051 standard. INT2 is falling edge triggered and Port1 Change Interrupt is triggered by any Port1 pin state change.



Pin Interrupt & Wake up

SFR 88h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

88h.3 **IE1:** External Interrupt 1 (INT1 pin) edge flag
Set by H/W when an INT1 pin falling edge is detected, no matter the EX1 is 0 or 1.
It is cleared automatically when the program performs the interrupt service routine.

88h.2 **IT1:** External Interrupt 1 control bit
0: Low level active (level triggered) for INT1 pin
1: Falling edge active (edge triggered) for INT1 pin

88h.1 **IE0:** External Interrupt 0 (INT0 pin) edge flag
Set by H/W when an INT0 pin falling edge is detected, no matter the EX0 is 0 or 1.
It is cleared automatically when the program performs the interrupt service routine.

88h.0 **IT0:** External Interrupt 0 control bit
0: Low level active (level triggered) for INT0 pin
1: Falling edge active (edge triggered) for INT0 pin

SFR 95h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTFLG	LVD	–	-	ADIF	–	IE2	P1IF	-
R/W	R	–	-	R/W	–	R/W	R/W	-
Reset	–	–	-	0	–	0	0	-

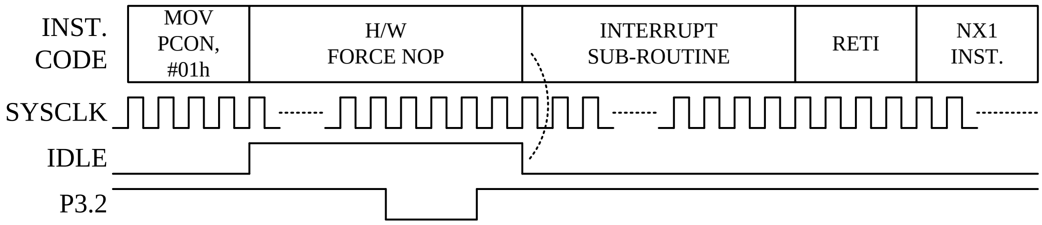
95h.2 **IE2:** External Interrupt 2 (INT2 pin) edge flag
Set by H/W when a falling edge is detected on the INT2 pin state, no matter the EX2 is 0 or 1.
It is cleared automatically when the program performs the interrupt service routine.
S/W can write FBh to INTFLG to clear this bit. (Note2)

95h.1 **P1IF:** Port1 pin change interrupt flag
Set by H/W when a P1 pin state change is detected, and its interrupt enable bit is set (P1WKUP).
P1IE does not affect this flag’s setting.
It is cleared automatically when the program performs the interrupt service routine.
S/W can write FDh to INTFLG to clear this bit. (Note2)

Note2: S/W can write 0 to clear a flag in the INTFLG, but writing 1 has no effect.

6.3 Idle Mode Wake up and Interrupt

Idle mode is wake up by enabled Interrupts, which means individual interrupt enable bit (ex: EX0) and EA bit must be both set to 1 to establish Idle mode wake up capability. All enabled Interrupts (Pins, Timers, ADC, SPIA, UART and USB) can wake up CPU from Idle mode. Upon Idle wake-up, Interrupt service routine is entered immediately. “The first instruction behind IDL (PCON.0) setting” is executed after interrupt service routine return.



EA=EX0=1, Idle mode wake-up and Interrupt by P3.2 (INT0)

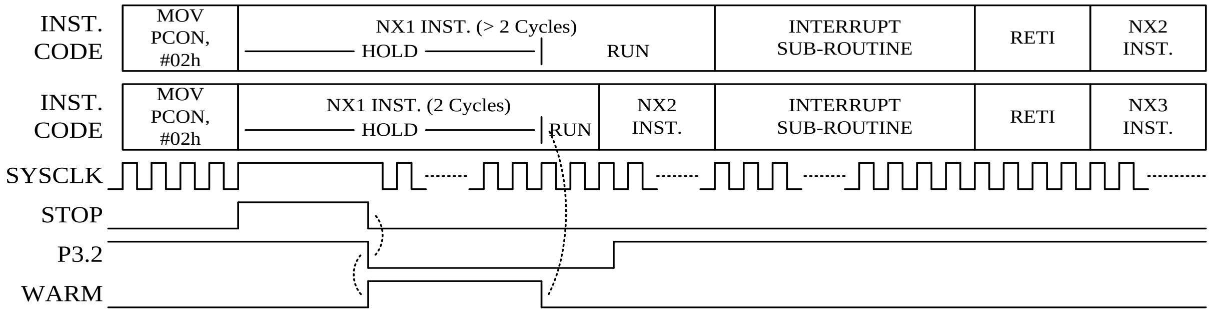
SFR 87h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	SMOD	–	–	–	GF1	GF0	PD	IDL
R/W	R/W	–	–	–	R/W	R/W	R/W	R/W
Reset	0	–	–	–	0	0	0	0

87h.1 **PD:** Stop bit. If 1, Stop mode is entered.
87h.0 **IDL:** Idle bit. If 1, Idle mode is entered.

6.4 Stop Mode Wake up and Interrupt

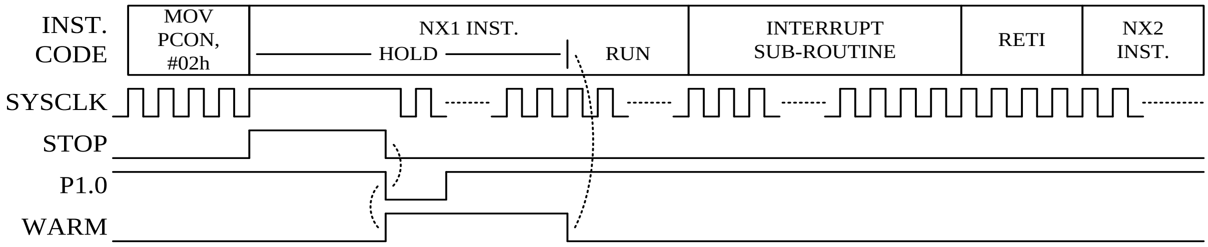
Stop mode wake up is simple, as long as the individual pin interrupt enable bit (ex: EX0) is set, the pin wake up capability is asserted. Set EX0/EX1/EX2 can enable INT0/INT1/INT2 pins’ Stop mode wake up capability. Set P1WKUP bit 7~0 can enable P1.7~P1.0’s Stop mode wake up capability. Upon Stop wake up, “ the first instruction behind PD (PCON.1) setting ” is executed immediately before Interrupt service. Interrupt entry needs EA=1 (P1WKUP also needs P1IE=1) and the trigger state of the pin staying sufficiently long to be observed by the

System clock. This feature allows CPU to enter or not enter Interrupt sub-routine after Stop mode wake up.



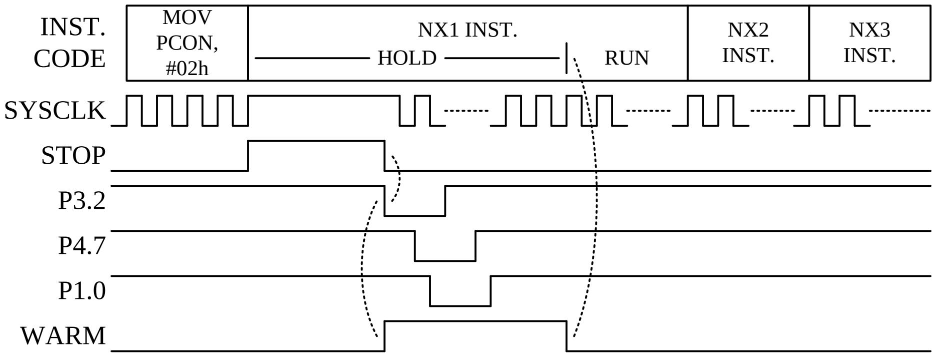
EA=EX0=1

P3.2 (INT0) is sampled after warm-up, Stop mode wake-up and Interrupt.



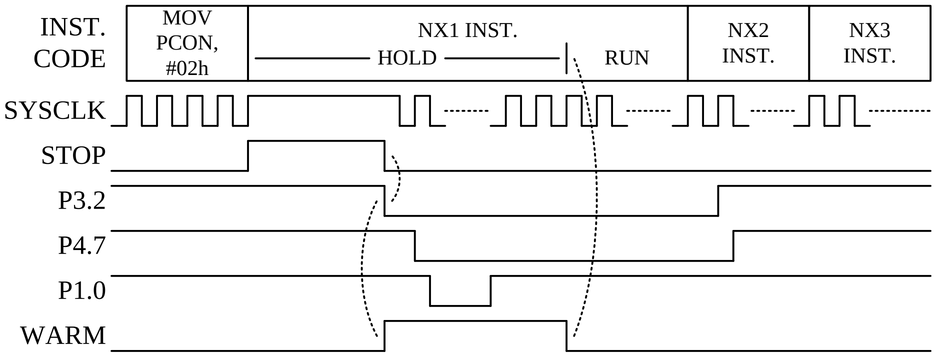
EA=P1IE=P1WKUP=1

P1.0 change (not need clock sample), Stop mode wake-up and Interrupt.



EA=EX0=EX2=P1WKUP=1, P1IE=0

Stop mode wake-up but not Interrupt, P3.2/P4.7 pulse too narrow.



EX0=EX2=P1WKUP=P1IE=1, EA=0
Stop mode wake-up but not Interrupt.

6.5 USB interrupt

There are several interrupts (USB1I and USB2I) generated by USB module when **F3288** works in USB application. USB1I has 8-source interrupts and USB2I has 6-source interrupts. If USB1I or USB2I interrupt occurs, F/W need to check SFR C2h or C3h to identify which USB interrupt source is generated.

SFR **C2h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
USB1I	SET0I	OUT0I	TX0I	TX1I	TX2I	SUSPI	TX3I	RC4I
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

C2h.7 Endpoint 0 SET0 Receive Interrupt flag, write 0 to clear flag.
C2h.6 Endpoint 0 OUT Receive Interrupt flag, write 0 to clear flag.
C2h.5 Endpoint 0 Transmit Interrupt flag, write 0 to clear flag.
C2h.4 Endpoint 1 Transmit Interrupt flag, write 0 to clear flag.
C2h.3 Endpoint 2 Transmit Interrupt flag, write 0 to clear flag.
C2h.2 USB Suspend Interrupt flag, write 0 to clear flag.
C2h.1 Endpoint 3 Bulk Transmit Interrupt flag, write 0 to clear flag.
C2h.0 Endpoint 4 Bulk Receive Interrupt flag, write 0 to clear flag.

SFR **C3h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
USB2I	VDD5RI	RSTI	RSMI	KBDI	PD_TXI	PD_RCI	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	-
Reset	0	0	0	0	0	0	-	-

C3h.7 VDD5V Rise Interrupt flag, write 0 to clear it
C3h.6 USB Bus Reset Interrupt flag, write 0 to clear flag.
C3h.5 USB Resume Interrupt flag, write 0 to clear flag.
C3h.4 Keyboard Interrupt flag, write 0 to clear flag.
C3h.3 USB PD Transmit Interrupt flag, write 0 to clear flag.
C3h.2 USB PD Receive Interrupt flag, write 0 to clear flag.

7. I/O Ports

The **F3288** has total 41 multi-function I/O pins. All I/O pins follow the standard 8051 “Read-Modify-Write” feature. The instructions that read the SFR rather than the Pin State are the ones that read a port or port bit value, possibly change it, and then rewrite it to the SFR. (ex: ANL P1, A; INC P2; CPL P3.0)

7.1 Port1 & Port3 & P4

These pins can operate in four different modes as below.

Mode	Port1, Port3, P4 pin function		Px.n SFR data	Pin State	Resistor Pull-up	Digital Input
	P3.2~P3.0	Others				
Mode 0	Pseudo Open Drain	Open Drain	0	Drive Low	N	N
			1	Pull-up	Y	Y
Mode 1	Pseudo Open Drain	Open Drain	0	Drive Low	N	N
			1	Hi-Z	N	Y
Mode 2	CMOS Output		0	Drive Low	N	N
			1	Drive High	N	N
Mode 3	Alternative Function, such as ADC		X (don’t care)	–	N	N

Port1, Port3, P4 I/O Pin Function Table

If a Port1 (except P1.2/P1.3), Port3 or P4 pin is used for Schmitt-trigger input, S/W must set the I/O pin to Mode0 or Mode1 and set the corresponding Port Data SFR to 1 to disable the pin’s output driving circuitry.

Beside I/O port function, each Port1 (except P1.2/P1.3), Port3 and P4 pin has one or more alternative functions, such as ADC. Most of the functions are activated by setting the individual pin mode control SFR to Mode3. Port1/Port3 pins have standard 8051 auxiliary definition such as INT0/1, T0/1/2, or RXD/TXD. These pin functions need to set the pin mode SFR to Mode0 or Mode1 and keep the P1.n/P3.n SFR at 1.

Pin Name	8051	Wake-up	CKO	ADC	others	Mode3
P1.0	T2	Y	T2O	AD0		AD0
P1.1	T2EX	Y		AD1		AD1
P1.2		Y			DP	
P1.3		Y			DM	
P1.4		Y		AD2	PWM0P	AD2
P1.5		Y		AD3	PWM1A	AD3
P1.6		Y		AD4	PWM1B	AD4
P1.7		Y		AD5	PWM1C	AD5
P3.0	RXD					
P3.1	TXD					
P3.2	INT0	Y		AD10		AD10/CC1
P3.3	INT1	Y		AD11		AD11/CC2
P3.4	T0		T0O		NSSI	
P3.5	T1				KSO15	
P3.6					KSO16	
P3.7					KSO17	
P4.0				AD6	KSO8	AD6
P4.1				AD7	KSO9	AD7

P4.2				AD8	KSO10	AD8
P4.3				AD9	KSO11	AD9
P4.4					KSO12	
P4.5					KSO13	
P4.6					KSO14	
P4.7	INT2	Y				

Port1, Port3, Port4 multi-function Table

The necessary SFR setting for Port1/Port3/Port4 pin's alternative function is list below.

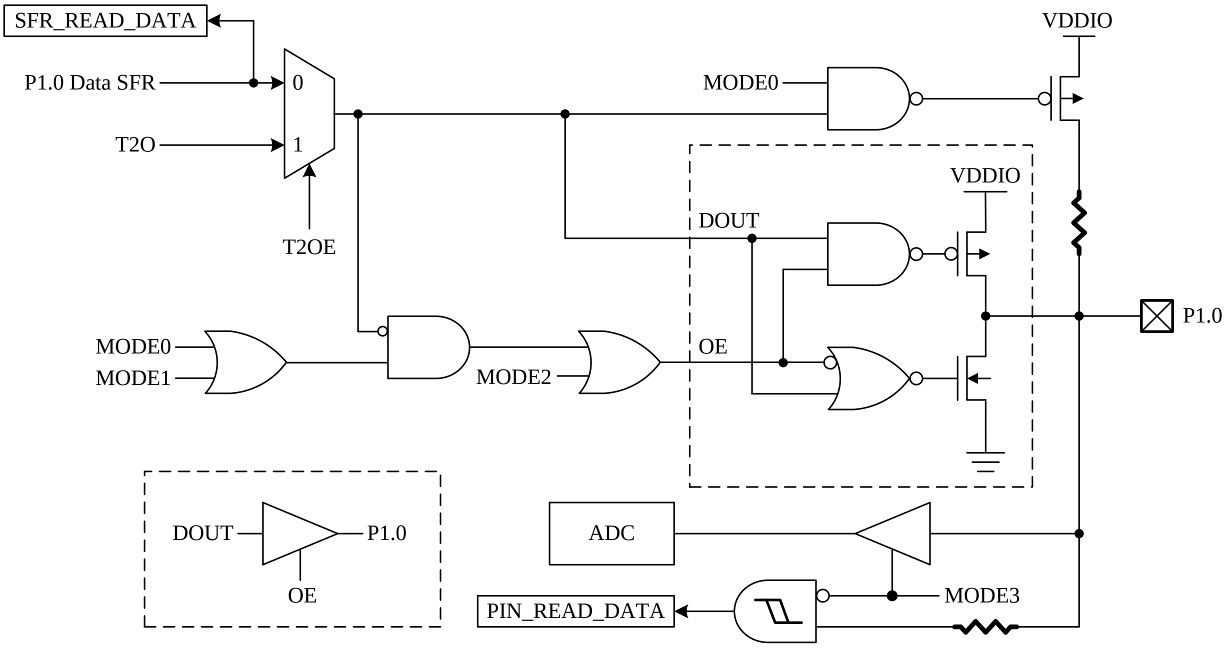
Alternative Function	Mode	Px.n SFR data	Pin State	Other necessary SFR setting
T0, T1, T2, T2EX, INT0, INT1	**0**	1	Input with Pull-up	
	1	1	Input	
RXD, TXD	**0**	1	Input with Pull-up/Pseudo Open Drain Output	
	1	1	Input/Pseudo Open Drain Output	
T0O, T2O	**0**	X	Clock Open Drain Output with Pull-up	PINMOD
	1	X	Clock Open Drain Output	
	2	X	Clock Output (CMOS Push-Pull)	
AD0~AD11	**3**	X	ADC Channel	
PWM1A, PWM1B, PWM1C, PWM0P	**0**	X	PWM Open Drain Output with Pull-up	PINMOD
	1	X	PWM Open Drain Output	
	2	X	PWM Output (CMOS Push-Pull)	
NSSI	**1**	1	SPIA Chip Selection	SPACON
Keyboard	**0**	X	Keyboard Open Drain Output with Pull-up	
	1	X	Keyboard Open Drain Output	
	2	X	Keyboard Output (CMOS Push-Pull)	
DP,DM	**X**	X	USB I/O	USBADR

Mode Setting for Port1, Port3, P4.1~P4.0 Alternative Function

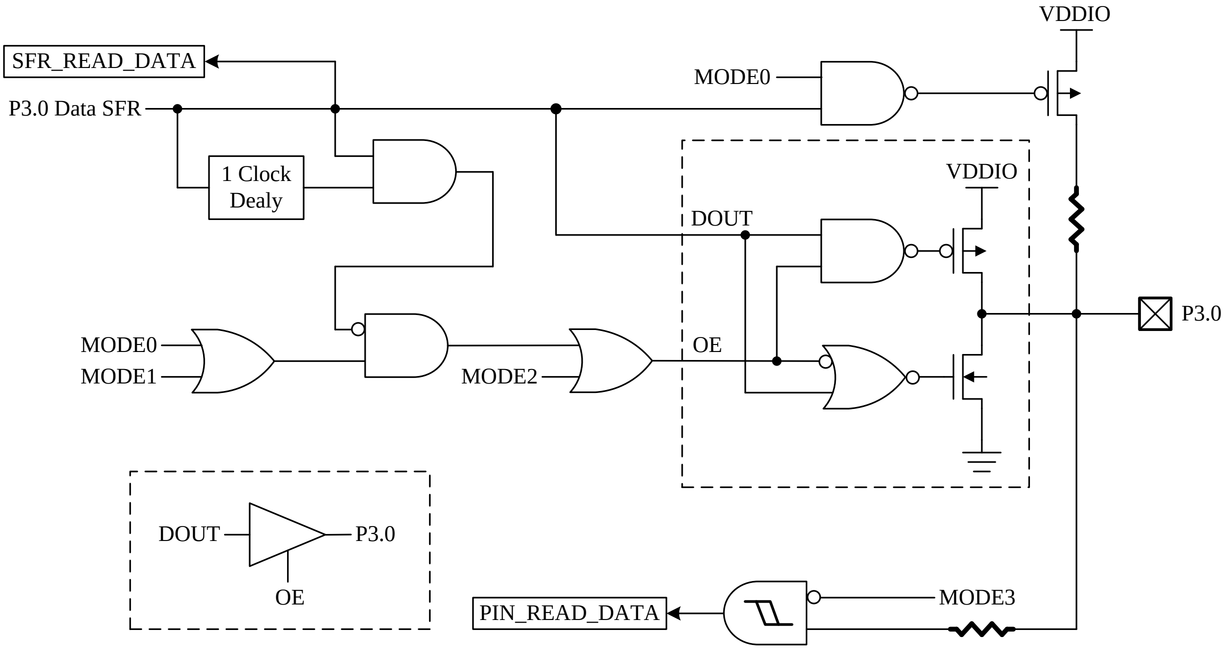
For tables above, a **"COMS Output"** pin means it can sink and drive at least 4mA current. It is not recommended to use such pin as input function.

An **"Open Drain"** pin means it can sink at least 4mA current but only drive a small current (<20μA). It can be used as input or output function and typically needs an external pull up resistor.

An 8051 standard pin is a **"Pseudo Open Drain"** pin. It can sink at least 4 mA current when output is at low level, and drives at least 4 mA current for 1~2 clock cycle when output transits from low to high, then keeps driving a small current (<20μA) to maintain the pin at high level. It can be used as input or output function.



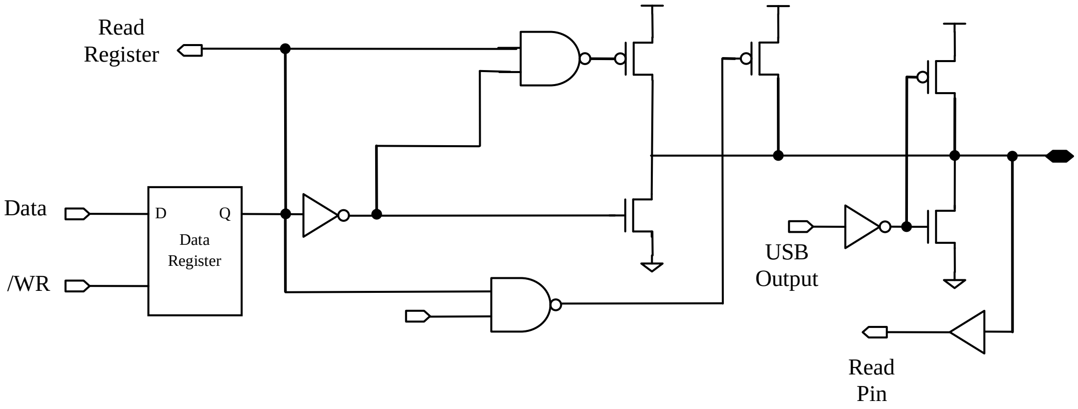
P1.0 Structure



P3.0 Structure

7.2 P1.2 (DP) and P1.3 (DM)

These pins are similar to other Port 1, except they share the pin with USB function. When **USBE** is set to "1" these two pin are used for USB function DP/DM. The maximum power of these two pin is VDD (internal power) not VDDIO (external input power).



DP/DM (P1.2/P1.3) Structure

SFR **90h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

90h.7~0 **P1:** Port1 data

SFR **B0h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

B0h.7~0 **P3:** Port3 data

SFR **E8h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P4	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

E8h.1~0 **P4:** Port 4 data

SFR **A2h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1MODL	P1MOD3		P1MOD2		P1MOD1		P1MOD0	
R/W	R/W		R/W		R/W		R/W	
Reset	0	0	0	0	0	0	0	0

A2h.7~6 **P1MOD3:** P1.3 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A2h.5~4 **P1MOD2:** P1.2 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A2h.3~2 **P1MOD1:** P1.1 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.1 is ADC input

A2h.1~0 **P1MOD0:** P1.0 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.0 is ADC input

SFR **A3h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P1MODH	P1MOD7		P1MOD6		P1MOD5		P1MOD4	
R/W	R/W		R/W		R/W		R/W	
Reset	0	0	0	0	0	0	0	0

A3h.7~6 **P1MOD7:** P1.7 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.7 is ADC input

A3h.5~4 **P1MOD6:** P1.6 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.6 is ADC input

A3h.3~2 **P1MOD5:** P1.5 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.5 is ADC input

A3h.1~0 **P1MOD4:** P1.4 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P1.4 is ADC input

SFR **A4h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3MODL	P3MOD3		P3MOD2		P3MOD1		P3MOD0	
R/W	R/W		R/W		R/W		R/W	
Reset	0	0	0	0	0	0	0	0

A4h.7~6 **P3MOD3:** P3.3 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P3.3 is ADC input

A4h.5~4 **P3MOD2:** P3.2 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P3.2 is ADC input

A4h.3~2 **P3MOD1:** P3.1 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A4h.1~0 **P3MOD0:** P3.0 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

SFR **A5h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P3MODH	P3MOD7		P3MOD6		P3MOD5		P3MOD4	
R/W	R/W		R/W		R/W		R/W	
Reset	0	0	0	0	0	0	0	0

A5h.7~6 **P3MOD7:** P3.7 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A5h.5~4 **P3MOD6:** P3.6 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A5h.3~2 **P3MOD5:** P3.5 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

A5h.1~0 **P3MOD4:** P3.4 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

SFR **B4h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P4MODL	P4MOD3		P4MOD2		P4MOD1		P4MOD0	
R/W	R/W		R/W		R/W		R/W	
Reset	0	0	0	0	0	0	0	0

B4h.7~6 **P4MOD3:** P4.3 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P4.3 is ADC input

B4h.5~4 **P4MOD2:** P4.2 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P4.2 is ADC input

B4h.3~2 **P4MOD1:** P4.1 pin control
00: Mode0
01: Mode1
10: Mode2

11: Mode3, P4.1 is ADC input

B4h.1~0 **P4MOD0:** P4.0 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3, P4.0 is ADC input

SFR **B5h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P4MODH	P4MOD7		P4MOD6		P4MOD5		P4MOD4	
R/W	R/W		R/W		R/W		R/W	
Reset	0	0	0	0	0	0	0	0

B5h.7~6 **P4MOD7:** P4.7 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

B5h.5~4 **P4MOD6:** P4.6 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

B5h.3~2 **P4MOD5:** P4.5 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

B5h.1~0 **P4MOD4:** P4.4 pin control
00: Mode0
01: Mode1
10: Mode2
11: Mode3

SFR **A6h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PINMOD	P5OE0	PWM0POE	PWM0NOE	T2OE	T0OE	PWM1AOE	PWM1BOE	PWM1COE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

A6h.6 **PWM0POE:** PWM0P signal output enable
0: Disable PWM0P signal output to P1.4
1: Enable PWM0P signal output to P1.4

A6h.4 **T2OE:** Timer2 signal output enable
0: Disable Timer2 overflow divided by 2 output to P1.0
1: Enable Timer2 overflow divided by 2 output to P1.0

A6h.3 **T0OE:** Timer0 signal output enable
0: Disable Timer0 overflow divided by 64 output to P3.4
1: Enable Timer0 overflow divided by 64 output to P3.4

A6h.2 **PWM1AOE:** PWM1A signal output enable
0: Disable PWM1A signal output to P1.5
1: Enable PWM1A signal output to P1.5

A6h.1 **PWM1BOE:** PWM1B signal output enable
0: Disable PWM1B signal output to P1.6
1: Enable PWM1B signal output to P1.6

A6h.0 **PWM1COE:** PWM1C signal output enable
0: Disable PWM1C signal output to P1.7
1: Enable PWM1C signal output to P1.7

SFR **BCh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPACON	SPAEN	SPAMSTR	SPACPOL	SPACPHA	SPASSDIS	SPALSBF	SPACR	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

BCh.3 **SPASSDIS:** SS pin disable

0: Enable SS pin, P3.4 is SPIA chip selection input.

1: Disable SS pin

SFR **C1h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
USBADR	USBE	FUNADR						
R/W	R/W	R/W						
Reset	0	0	0	0	0	0	0	0

C1h.7 **USBE:** USB function enable bit

0: Disable P1.2/P1.3 as USB DP/DM pin

1: Enable P1.2/P1.3 as USB DP/DM pin

7.3 Port0 & Port2 & Port5.0

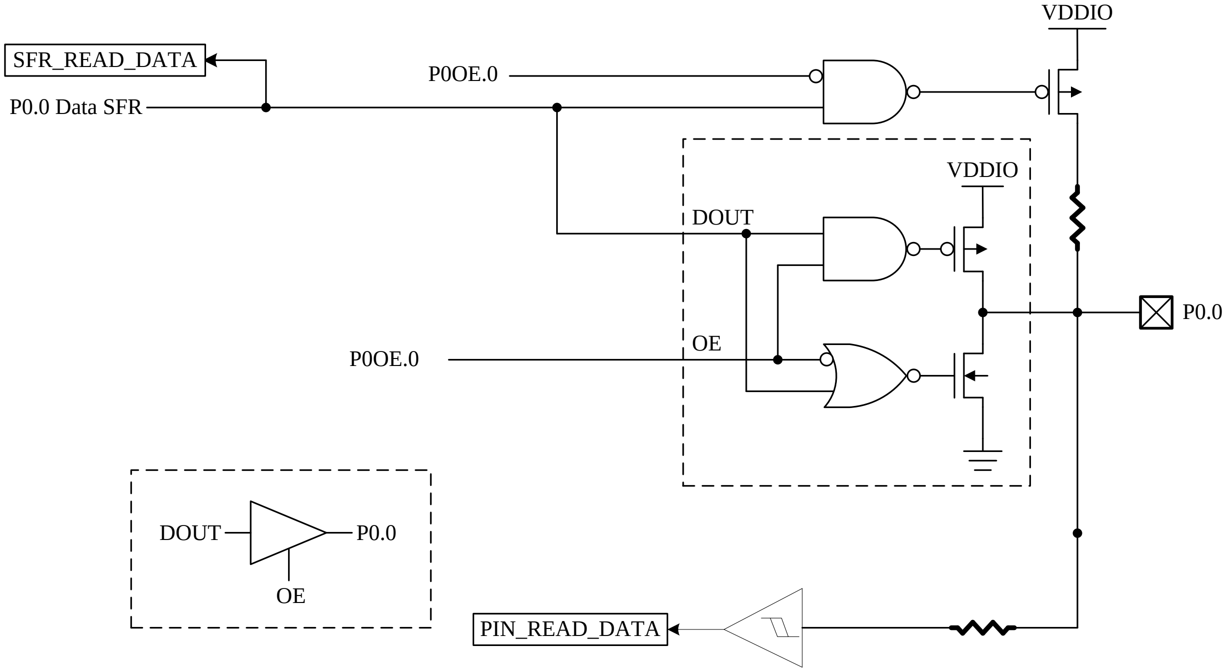
These pins are shared with SPI, ADC, I80 and Keyboard. If a Port0/Port2/Port5.0 pin is defined as I/O pin, it can be used as CMOS push-pull output or Schmitt-trigger input. The pin's pull up function is enable while SFR bit PxOE.n=0 and Px.n=1.

Port0, Port2, Port5.0 pin function	PxOE.n	Px.n SFR data	Pin State	Resistor Pull-up	Digital Input
Input	0	0	Hi-Z	N	Y
	0	1	Pull-up	Y	Y
CMOS Output	1	0	Drive Low	N	N
	1	1	Drive High	N	N

Port0, Port2, Port5.0 I/O Pin Function Table

Pin Name	I80	Key	SPI
P0.0	D0	KSI0	
P0.1	D1	KSI1	
P0.2	D2	KSI2	
P0.3	D3	KSI3	
P0.4	D4	KSI4	
P0.5	D5	KSI5	
P0.6	D6	KSI6	
P0.7	D7	KSI7	
P2.0		KSO0	MISO
P2.1		KSO1	SCK
P2.2		KSO2	MOSI
P2.3	WR	KSO3	
P2.4		KSO4	BSDI
P2.5		KSO5	BSCK
P2.6		KSO6	BSDO
P2.7	RD	KSO7	
P5.0			

Port0, Port2, Port5.0 multi-function Table



P0.0 Pin Structure

SFR **80h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

80h.7~0 **P0:** Port0 data, also controls the P0.n pin’s pull-up function. If the P0.n SFR data is “1” and the corresponding P0OE.n=0 (input mode), the pull-up is enabled.

SFR **A0h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P2	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	1	1	1	1	1	1	1

A0h.7~0 **P2:** Port2 data, also controls the P2.n pin’s pull-up function. If the P2.n SFR data is “1” and the corresponding P2OE.n=0 (input mode), the pull-up is enabled.

SFR **C0h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P5	-	-	-	-	-	-	-	P50D
R/W	-	-	-	-	-	-	-	R/W
Reset	-	-	-	-	-	-	-	1

C0h.0 **P50D:** P5.0 data, also controls the P5.0 pin’s pull-up function. If the P5.0 SFR data is “1” and the corresponding P5OE0=0 (input mode), the pull-up is enabled.

SFR **91h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P0OE	P0OE							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

91h.7~0 **P0OE:** Port0 CMOS Push-Pull output enable control
0: Disable
1: Enable

SFR **93h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
P2OE	P2OE							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

93h.7~0 **P2OE:** Port2 CMOS Push-Pull output enable control
0: Disable
1: Enable

SFR **A6h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PINMOD	P5OE0	PWM0POE	PWM0NOE	T2OE	T0OE	PWM1AOE	PWM1BOE	PWM1COE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

A6h.7 **P5OE0:** P5.0 CMOS Push-Pull output enable control
0: Disable
1: Enable

A6h.5 **PWM0NOE:** PWM0N signal output enable
0: Disable PWM0N signal output to P5.0
1: Enable PWM0N signal output to P5.0

8. Timers

Timer0, Timer1 and Timer2 are provided as standard 8051 compatible timer/counter. Compare to the traditional 12T 8051, the chip's Timer0/1/2 use 2 System clock cycle as the time base unit. That is, in timer mode, these timers increase at every “2 System clock” rate; in counter mode, T0/T1/T2 pin input pulse must be wider than 2 System clock to be seen by this device. In addition to the standard 8051 timers function. The T0O pin can output the “Timer0 overflow divided by 64” signal, and the T2O pin can output the “Timer2 overflow divided by 2” signal.

8.1 Timer0/1

TCON and TMOD are used to set the mode of operation and to control the running and interrupt generation of the Timer0/1, with the timer/counter values stored in two pairs of 8-bit registers (TL0, TH0, and TL1, TH1).

SFR **88h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

88h.7 **TF1:** Timer1 overflow flag
Set by H/W when Timer/Counter 1 overflows.
Cleared by H/W when CPU vectors into the interrupt service routine.

88h.6 **TR1:** Timer1 run control
0: Timer1 stops
1: Timer1 runs

88h.5 **TF0:** Timer0 overflow flag
Set by H/W when Timer/Counter 0 overflows.
Cleared by H/W when CPU vectors into the interrupt service routine.

88h.4 **TR0:** Timer0 run control
0: Timer0 stops
1: Timer0 runs

SFR **89h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TMOD	GATE1	CT1N	TMOD1		GATE0	CT0N	TMOD0	
R/W	R/W	R/W	R/W		R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

89h.7 **GATE1:** Timer1 gating control bit
0: Timer1 enable when TR1 bit is set
1: Timer1 enable only while the INT1 pin is high and TR1 bit is set

89h.6 **CT1N:** Timer1 Counter/Timer select bit
0: Timer mode, Timer1 data increases at 2 System clock cycle rate
1: Counter mode, Timer1 data increases at T1 pin’s negative edge

89h.5~4 **TMOD1:** Timer1 mode select
00: 8-bit timer/counter (TH1) and 5-bit prescaler (TL1)
01: 16-bit timer/counter
10: 8-bit auto-reload timer/counter (TL1). Reloaded from TH1 at overflow.
11: Timer1 stops

89h.3 **GATE0:** Timer0 gating control bit
0: Timer0 enable when TR0 bit is set
1: Timer0 enable only while the INT0 pin is high and TR0 bit is set

89h.2 **CT0N:** Timer0 Counter/Timer select bit
0: Timer mode, Timer0 data increases at 2 System clock cycle rate
1: Counter mode, Timer0 data increases at T0 pin’s negative edge

89h.1~0 **TMOD0:** Timer0 mode select
00: 8-bit timer/counter (TH0) and 5-bit prescaler (TL0)
01: 16-bit timer/counter
10: 8-bit auto-reload timer/counter (TL0). Reloaded from TH0 at overflow.
11: TL0 is an 8-bit timer/counter. TH0 is an 8-bit timer/counter using Timer1's TR1 and TF1 bits.

SFR **8Ah**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TL0	TL0							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

8Ah.7~0 **TL0:** Timer0 data low byte

SFR **8Bh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TL1	TL1							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

8Bh.7~0 **TL1:** Timer1 data low byte

SFR **8Ch**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TH0	TH0							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

8Ch.7~0 **TH0:** Timer0 data high byte

SFR **8Dh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TH1	TH1							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

8Dh.7~0 **TH1:** Timer1 data high byte

Note: also refer to Section 6 for more information about Timer0/1 Interrupt enable and priority.
Note: also refer to Section 14 for more information about T0O pin output setting.

8.2 Timer2

Timer2 is controlled through the TCON2 register with the low and high bytes of Timer/Counter 2 stored in TL2 and TH2 and the low and high bytes of the Timer2 reload/capture registers stored in RCAP2L and RCAP2H.

SFR **C8h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	CT2N	CPRL2N
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

C8h.7 **TF2:** Timer2 overflow flag
Set by H/W when Timer/Counter 2 overflows unless RCLK=1 or TCLK=1. This bit must be cleared by S/W.

C8h.6 **EXF2:** T2EX interrupt pin falling edge flag
Set when a capture or a reload is caused by a negative transition on T2EX pin if EXEN2=1. This bit must be cleared by S/W.

C8h.5 **RCLK:** UART receive clock control bit
0: Use Timer1 overflow as receive clock for serial port in mode 1 or 3
1: Use Timer2 overflow as receive clock for serial port in mode 1 or 3

C8h.4 **TCLK:** UART transmit clock control bit
0: Use Timer1 overflow as transmit clock for serial port in mode 1 or 3
1: Use Timer2 overflow as transmit clock for serial port in mode 1 or 3

C8h.3 **EXEN2:** T2EX pin enable
0: T2EX pin disable
1: T2EX pin enable, it cause a capture or reload when a negative transition on T2EX pin is detected if RCLK=TCLK=0

C8h.2 **TR2:** Timer2 run control
0: Timer2 stops
1: Timer2 runs

C8h.1 **CT2N:** Timer2 Counter/Timer select bit
0: Timer mode, Timer2 data increases at 2 System clock cycle rate
1: Counter mode, Timer2 data increases at T2 pin's negative edge

C8h.0 **CPRL2N:** Timer2 Capture/Reload control bit
0: Reload mode, auto-reload on Timer2 overflows or negative transitions on T2EX pin if EXEN2=1
1: Capture mode, capture on negative transitions on T2EX pin if EXEN2=1
If RCLK=1 or TCLK=1, CPRL2N is ignored and timer is forced to auto-reload on Timer2 overflow

SFR **CAh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RCP2L	RCP2L							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

CAh.7~0 **RCP2L:** Timer2 reload/capture data low byte

SFR **CBh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RCP2H	RCP2H							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

CBh.7~0 **RCP2H:** Timer2 reload/capture data high byte

SFR **CCh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TL2	TL2							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

CCh.7~0 **TL2:** Timer2 data low byte

SFR **CDh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TH2	TH2							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

CDh.7~0 **TH2:** Timer2 data high byte

Note: also refer to Section 6 for more information about Timer2 Interrupt enable and priority.
Note: also refer to Section 7 for more information about T2O pin output setting.

8.3 T0O and T2O Output Control

This device can generate various frequency waveform pin output (in CMOS push pull format) for Buzzer. The T0O and T2O waveform is divided by Timer0/Timer2 overflow signal. The T0O waveform is Timer0 overflow divided by 64, and T2O waveform is Timer2 overflow divided by 2. User can control their frequency by Timers auto reload speed. Set T0OE and T2OE SFRs can output these waveforms.

SFR **A6h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PINMOD	P5OE0	PWM0POE	PWM0NOE	T2OE	T0OE	PWM1AOE	PWM1BOE	PWM1COE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

A6h.4 **T2OE:** Timer2 signal output enable
0: Disable Timer2 overflow divided by 2 output to P1.0
1: Enable Timer2 overflow divided by 2 output to P1.0

A6h.3 **T0OE:** Timer0 signal output enable
0: Disable Timer0 overflow divided by 64 output to P3.4
1: Enable Timer0 overflow divided by 64 output to P3.4

9. UART

The UART uses SCON and SBUF SFRs. SCON is the control register, SBUF is the data register. Data is written to SBUF for transmission and SBUF is read to obtain received data. The received data and transmitted data registers are completely independent. In addition to standard 8051's full duplex mode, this chip also provides one wire mode. If the UART1W bit is set, both transmit and receive data use P3.1 pin.

SFR **87h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCON	SMOD	–	–	–	GF1	GF0	PD	IDL
R/W	R/W	–	–	–	R/W	R/W	R/W	R/W
Reset	0	–	–	–	0	0	0	0

87h.7 **SMOD:** UART double baud rate control bit
0: Disable UART double baud rate
1: Enable UART double baud rate

SFR **94h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	UART1W	MODE3V	WDTPSC		ADCKS		-	-
R/W	R/W	R/W	R/W		R/W		-	-
Reset	0	0	0	0	0	0	0	0

94h.7 **UART1W:** One wire UART mode enable, both TXD/RXD use P3.1 pin
0: Disable one wire UART mode
1: Enable one wire UART mode

SFR **98h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

98h.7~6 **SM0,SM1:** Serial port mode select bit 0,1
00: Mode0: 8 bit shift register, Baud Rate=$F_{SYSCLK}/2$
01: Mode1: 8 bit UART, Baud Rate is variable
10: Mode2: 9 bit UART, Baud Rate=$F_{SYSCLK}/32$ or /64
11: Mode3: 9 bit UART, Baud Rate is variable

98h.5 **SM2:** Serial port mode select bit 2
SM2 enables multiprocessor communication over a single serial line and modifies the above as follows. In Modes 2 & 3, if SM2 is set then the received interrupt will not be generated if the received ninth data bit is 0. In Mode 1, the received interrupt will not be generated unless a valid stop bit is received. In Mode 0, SM2 should be 0.

98h.4 **REN:** UART reception enable
0: Disable reception
1: Enable reception

98h.3 **TB8:** Transmit Bit 8, the ninth bit to be transmitted in Mode 2 and 3

98h.2 **RB8:** Receive Bit 8, contains the ninth bit that was received in Mode 2 and 3 or the stop bit in Mode 1 if SM2=0

98h.1 **TI:** Transmit interrupt flag
Set by H/W at the end of the eighth bit in Mode 0, or at the beginning of the stop bit in other modes. Must be cleared by S/W.

98h.0 **RI:** Receive interrupt flag
Set by H/W at the end of the eighth bit in Mode 0, or at the sampling point of the stop bit in other modes. Must be cleared by S/W.

SFR **99h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SBUF	SBUF							
R/W	R/W							
Reset	–	–	–	–	–	–	–	–

99h.7~0 **SBUF:** UART transmit and receive data. Transmit data is written to this location and receive data is read from this location, but the paths are independent.

F_{SYSCLK} denotes System clock frequency.

- Mode 0:
  Baud Rate=$F_{SYSCLK}/2$
- Mode 1, 3: if using Timer1 auto reload mode
  Baud Rate= (SMOD+1) x F_{SYSCLK}/ (32x2x (256–TH1))
- Mode 1, 3: if using Timer2
  Baud Rate=Timer2 overflow rate/16=F_{SYSCLK}/ (32x (65536–RCP2H, RCP2L))
- Mode 2:
  Baud Rate= (SMOD+1) x $F_{SYSCLK}/64$

Note: also refer to Section 6 for more information about UART Interrupt enable and priority.
Note: also refer to Section 8 for more information about how Timer2 controls UART clock.

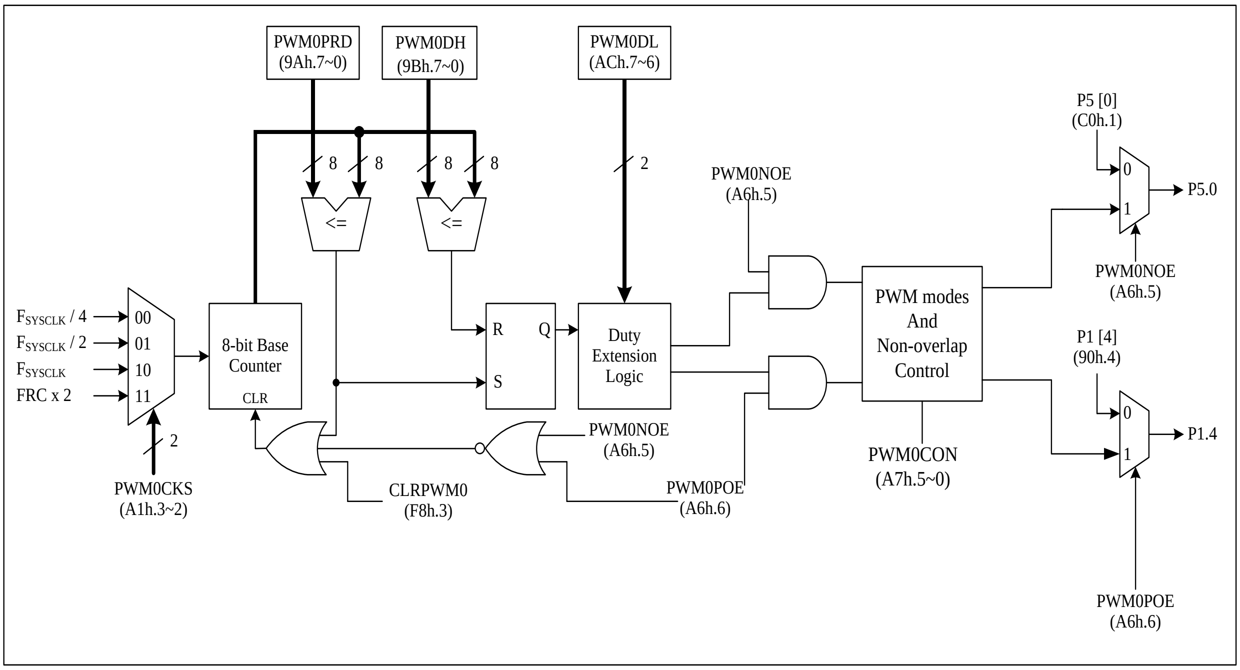
10. PWMs

The PWM of **F3288** is a simple structure, which switches its output high and low at uniform repeatable intervals. It is widely used to control the brightness of a light, in which case the PWM circuit is used to turn on/off a power line. If the line is repeatedly turned on for 500 ms and turned off for 500 ms per every second, the PWM would have an average output of half of the voltage or run at half speeds or half brightness. The PWM has been used in a wide variety of applications, such as voltage control, current control, motor speed control, light brightness control, and as a voltage inverter.

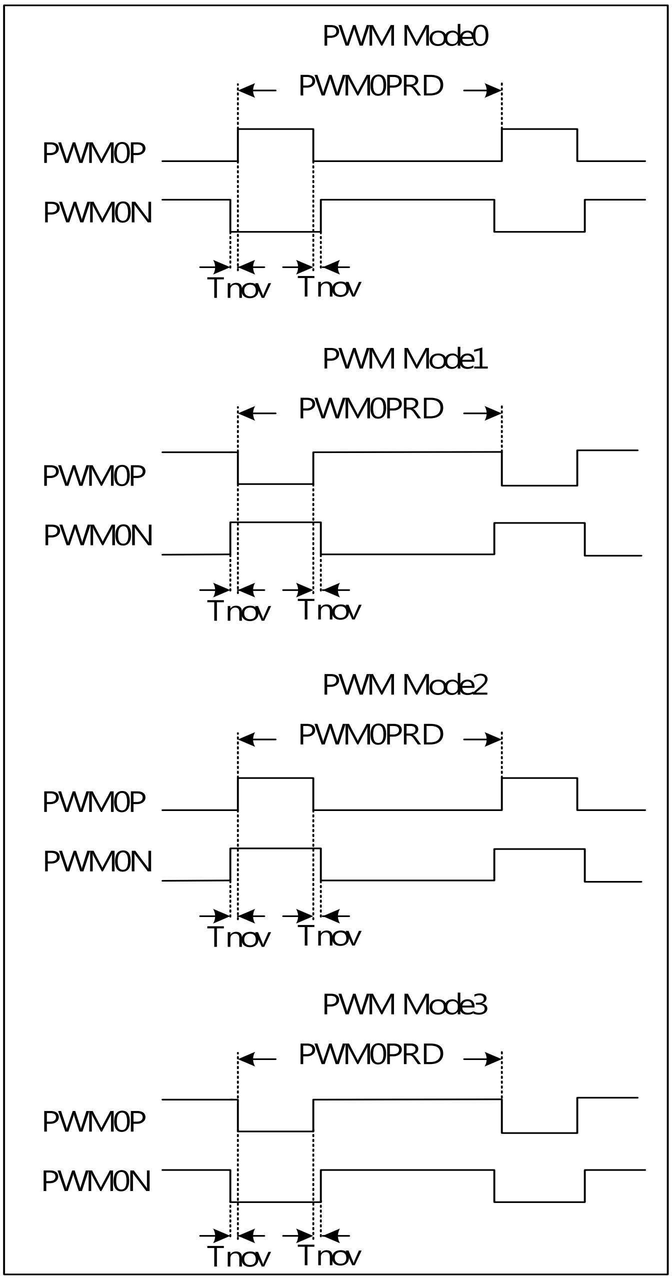
The **F3288** has two independent PWM modules, PWM0 and PWM1. The PWM can generate a fixed frequency waveform with 1024 duty resolution on the basis of the PWM clock. The PWM clock can select FRC24MHz or F_{SYSCLK} divided by 1, 2, or 4 by PWM0CKS. A spread LSB technique allows PWM to run its frequency at the “PWM clock divided by 256” instead of at the “PWM clock divided by 1024”, which means the PWM is four times faster than normal. The advantage of a higher PWM frequency is that the post RC filter can transform the PWM signal to a more stable DC voltage level. Because two PWMs have the same structure, the following describes only the functions of the PWM0.

10.1 PWM0

The PWM0 duty cycle can be changed by writing to PWM0DH (9Bh) and PWM0DL (Ach.7~6). The PWM0 output signal resets to a low level whenever the 8-bit base counter matches the 8-bit MSB of the PWM0 duty register PWM0DH. When the base counter rolls over, the 2-bit LSB of the PWM0 duty register PWM0DL decides whether to set the PWM0 output signal high immediately or set it high after one clock cycle delay. The PWM0 period can be set by writing the period value to the PWM0PRD (9Ah) register.

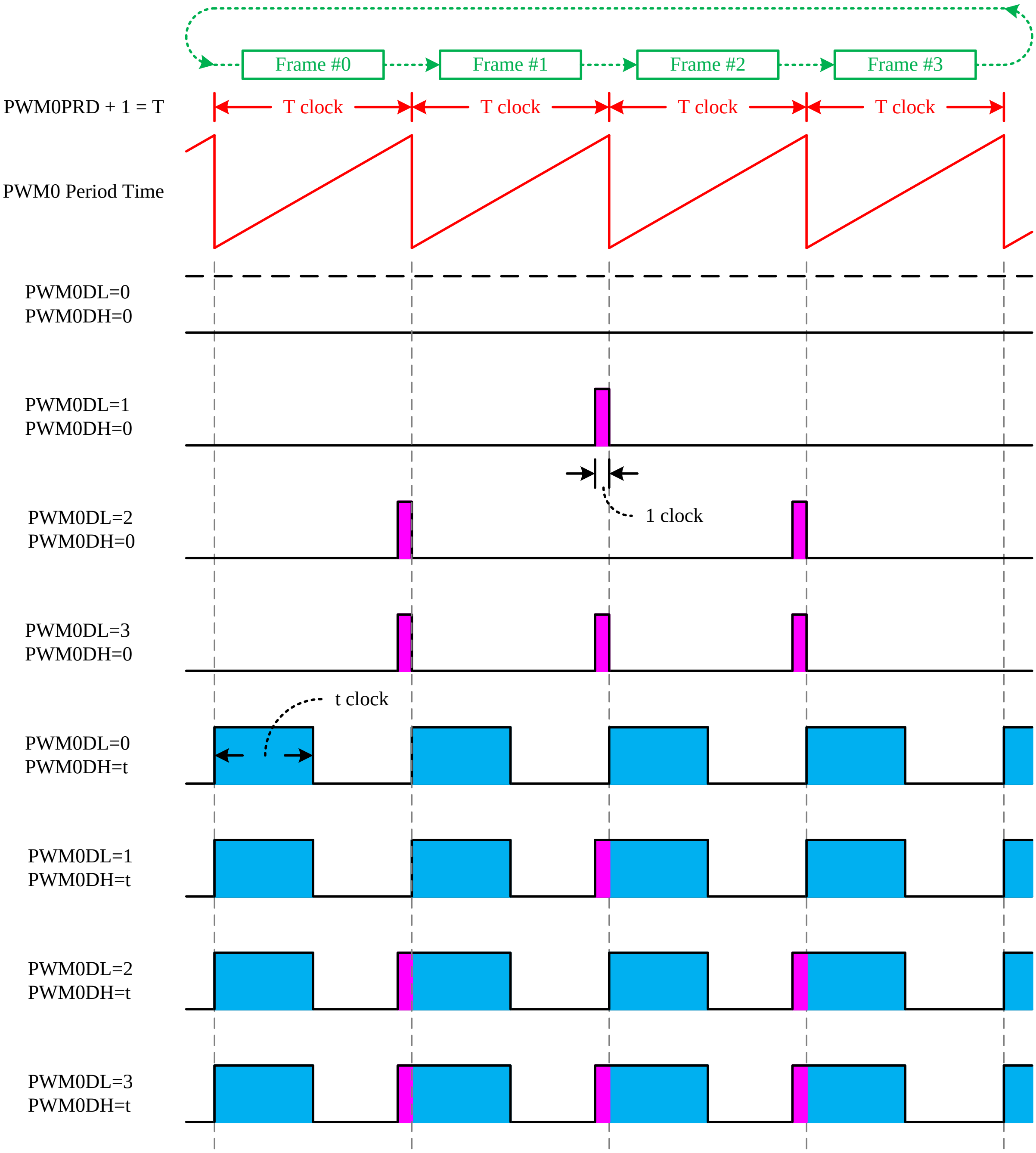


The PWM0 has two output pins and can be configured to use P1.4 and P5.0 with four different modes. The edges of the PWM0 pulse can be separated with 16 different non-overlap clocks time intervals. The PWM0 mode and non-overlap clocks time intervals are selected by PWM0CON (A7h). The default output is PWM Mode0 and the zero non-overlap clocks time intervals. User can select one of the pins as an output or let PWM0 signal output simultaneously on two output pins. The PWM0POE bit is used to select the output for PWM0P (P1.4), and the PWM0NOE bit is used to select the output for PWM0N (P5.0). These two bits also can be PWM0 control bit. If both bits are cleared, the PWM0 will be cleared and stopped, otherwise the PWM0 is running. The CLRPWM0 bit has the same function. When CLRPWM0 bit is set, the PWM0 will be cleared and held, otherwise the PWM0 is running.



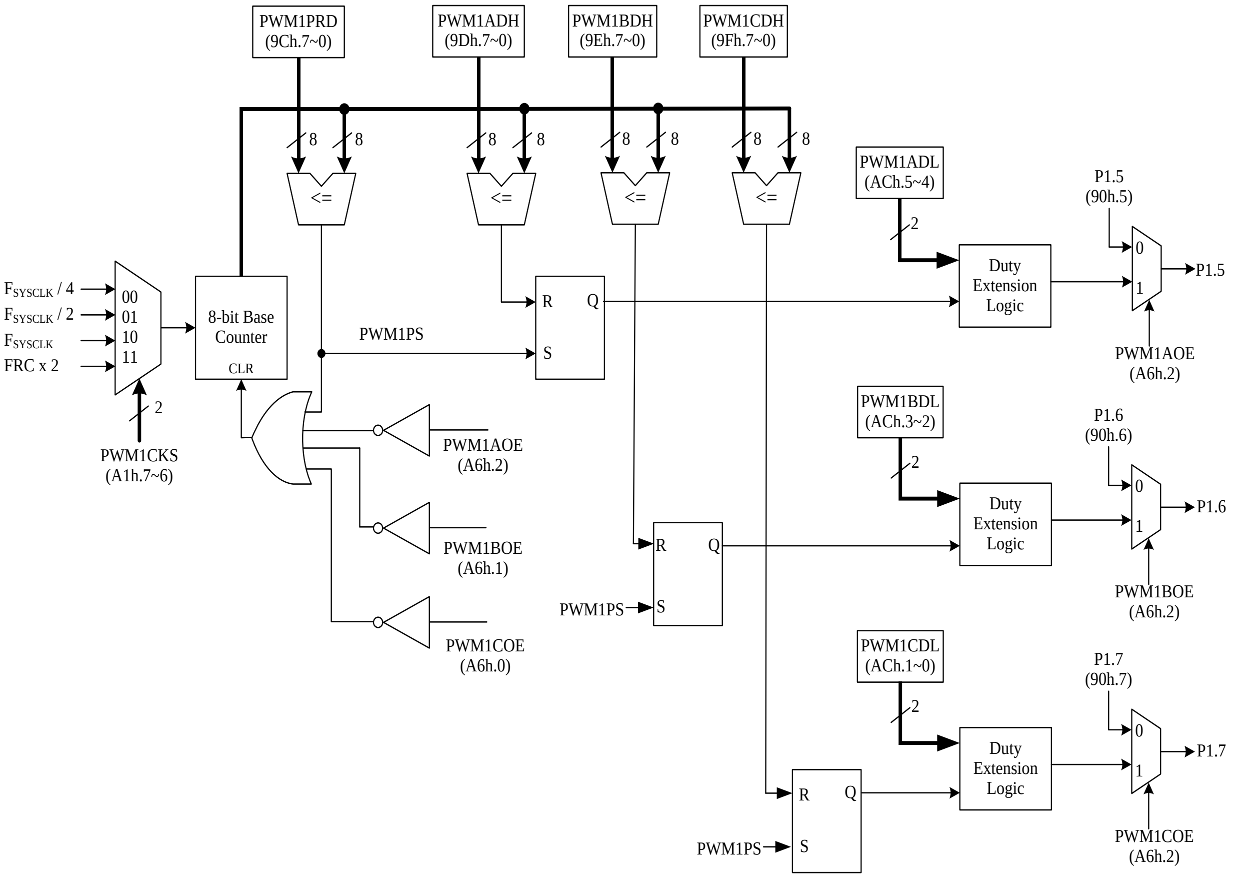
PWM0 Output Modes

The PWM0 offers 2-bit of an extension data register PWM0DL to provide an extension of the cycle of the PWM output. The extension cycle involves the placement of one extra clock period at specific intervals.



10.2 PWM1

The PWM1 has three outputs (PWM1A, PWM1B and PWM1C). These PWM1 outputs share the same period setting PWM1PRD (9Ch) but with the different duty setting PWM1ADTY, PWM1BDTY and PWM1CDTY.



SFR **9Ah**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0PRD	PWM0PRD							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

9Ah.7~0 **PWM0PRD:** PWM0 period

SFR **9Bh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0DH	PWM0DH							
R/W	R/W							
Reset	1	0	0	0	0	0	0	0

9Bh.7~0 **PWM0DH:** PWM0 duty high byte

The PWM0 output signal is reset to a low level whenever the 8-bit base counter matches the 8-bit PWM0DH.

SFR **9Ch**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1PRD	PWM1PRD							
R/W	R/W							
Reset	1	1	1	1	1	1	1	1

9Ch.7~0 **PWM1PRD:** PWM1 period

SFR **9Dh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1ADH	PWM1ADH							
R/W	R/W							
Reset	1	0	0	0	0	0	0	0

9Dh.7~0 **PWM1ADH:** PWM1A duty high byte

The PWM1A output signal is reset to a low level whenever the 8-bit base counter matches the 8-bit PWM1ADH.

SFR **9Eh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1BDH	PWM1BDH							
R/W	R/W							
Reset	1	0	0	0	0	0	0	0

9Eh.7~0 **PWM1BDH:** PWM1B duty high byte

The PWM1B output signal is reset to a low level whenever the 8-bit base counter matches the 8-bit PWM1BDH.

SFR **9Fh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM1CDH	PWM1CDH							
R/W	R/W							
Reset	1	0	0	0	0	0	0	0

9Fh.7~0 **PWM1CDH:** PWM1C duty high byte

The PWM1C output signal is reset to a low level whenever the 8-bit base counter matches the 8-bit PWM1CDH.

SFR **A1h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMCKS	PWM1CKS		-	-	PWM0CKS		-	-
R/W	R/W		-	-	R/W		-	-
Reset	1	0	-	-	1	0	-	-

A1h.7~6 **PWM1CKS:** PWM1 clock source

00: $F_{SYSCLK}/4$
01: $F_{SYSCLK}/2$
10: F_{SYSCLK}
11: FRC24MHz

A1h.3~2 **PWM0CKS:** PWM0 clock source

00: $F_{SYSCLK}/4$
01: $F_{SYSCLK}/2$
10: F_{SYSCLK}
11: FRC24MHz

SFR **A6h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PINMOD	P5OE0	PWM0POE	PWM0NOE	T2OE	T0OE	PWM1AOE	PWM1BOE	PWM1COE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

A6h.6 **PWM0POE:** PWM0P signal output enable
0: Disable PWM0P signal output to P1.4
1: Enable PWM0P signal output to P1.4

A6h.5 **PWM0NOE:** PWM0N signal output enable
0: Disable PWM0N signal output to P5.0
1: Enable PWM0N signal output to P5.0

A6h.2 **PWM1AOE:** PWM1A signal output enable
0: Disable PWM1A signal output to P1.5
1: Enable PWM1A signal output to P1.5

A6h.1 **PWM1BOE:** PWM1B signal output enable
0: Disable PWM1B signal output to P1.6
1: Enable PWM1B signal output to P1.6

A6h.0 **PWM1COE:** PWM1C signal output enable
0: Disable PWM1C signal output to P1.7
1: Enable PWM1C signal output to P1.7

SFR **A7h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWM0CON	-	-	PWM0MODE		PWM0NOVT			
R/W	-	-	R/W		R/W			
Reset	-	-	0		0			

A7h.5~4 **PWM0MODE:** PWM0 differential output mode
00: Mode 0
01: Mode 1
10: Mode 2
11: Mode 3

A7h.3~0 **PWM0NOVT:** PWM0 non-overlap time (dead time) select
0000: 0 pwm0 clock (original PWM0)
0001: 1 pwm0 clock
0010: 2 pwm0 clock
0011: 3 pwm0 clock
0100: 4 pwm0 clock
0101: 5 pwm0 clock
0110: 6 pwm0 clock
0111: 7 pwm0 clock
1000: 8 pwm0 clock
1001: 9 pwm0 clock
1010: 10 pwm0 clock
1011: 11 pwm0 clock
1100: 12 pwm0 clock
1101: 13 pwm0 clock
1110: 14 pwm0 clock
1111: 16 pwm0 clock

SFR **ACh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PWMDL	PWM0DL		PWM1ADL		PWM1BDL		PWM1CDL	
R/W	R/W		R/W		R/W		R/W	
Reset	0	0	0	0	0	0	0	0

ACh.7~6 **PWM0DL:** PWM0 duty low byte
When the base counter rolls over, the PWM0DL decides whether to set the PWM0 output signal high immediately or set it high after one clock cycle delay

ACh.5~4 **PWM1ADL:** PWM1A duty low byte
When the base counter rolls over, the PWM1ADL decides whether to set the PWM1A output signal high immediately or set it high after one clock cycle delay

ACh.3~2 **PWM1BDL:** PWM1B duty low byte
When the base counter rolls over, the PWM1BDL decides whether to set the PWM1B output signal high immediately or set it high after one clock cycle delay

ACh.7~6 **PWM1CDL:** PWM1C duty low byte
When the base counter rolls over, the PWM1CDL decides whether to set the PWM1C output signal high immediately or set it high after one clock cycle delay

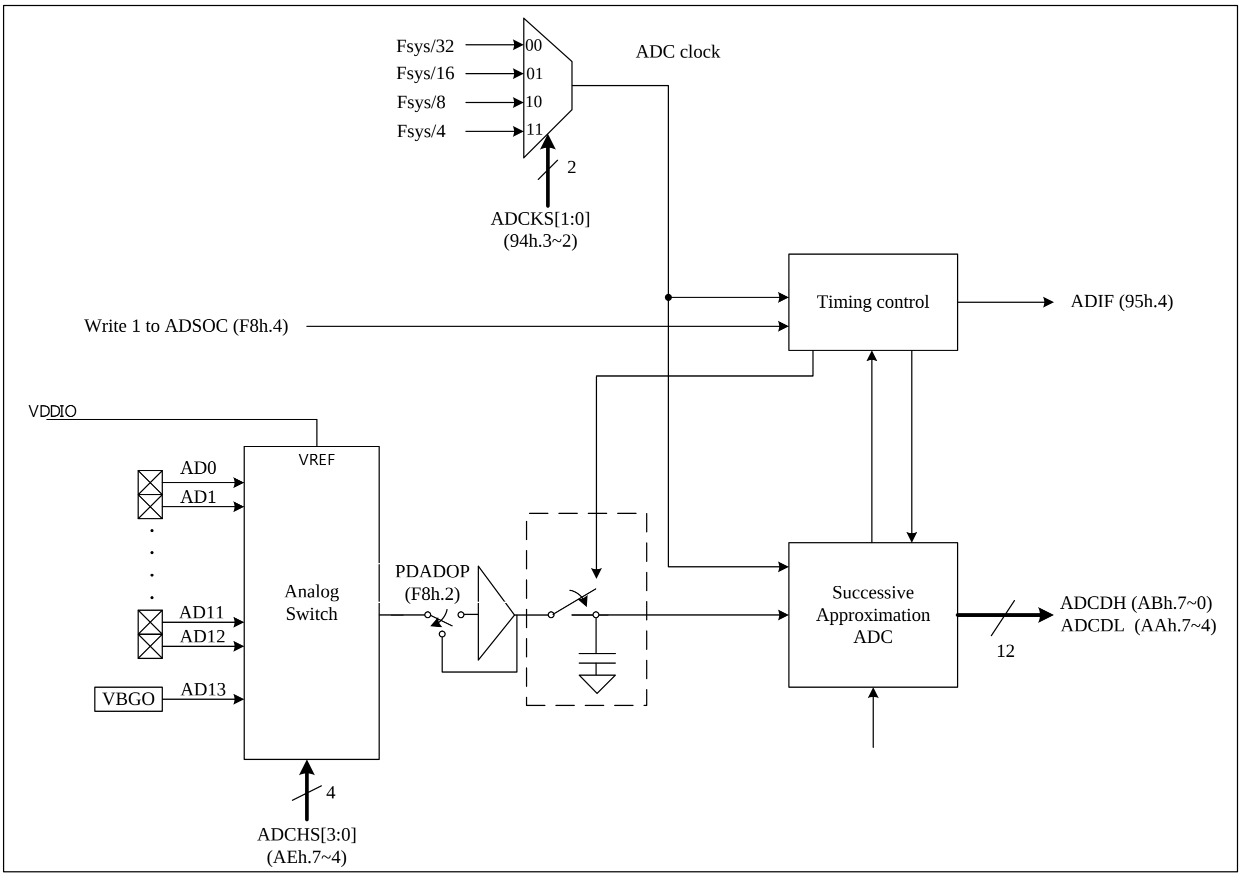
SFR **F8h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AUX1	CLRWDT	WDTE		ADSOC	CLRPWM0	PDADOP	BOOTV	DPSEL
R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

F8h.3 **CLRPWM0:** PWM0 clear enable
0: PWM0 is running
1: PWM0 is cleared and held

Note: also refer to Section 7 for more information about PWM pin output setting.

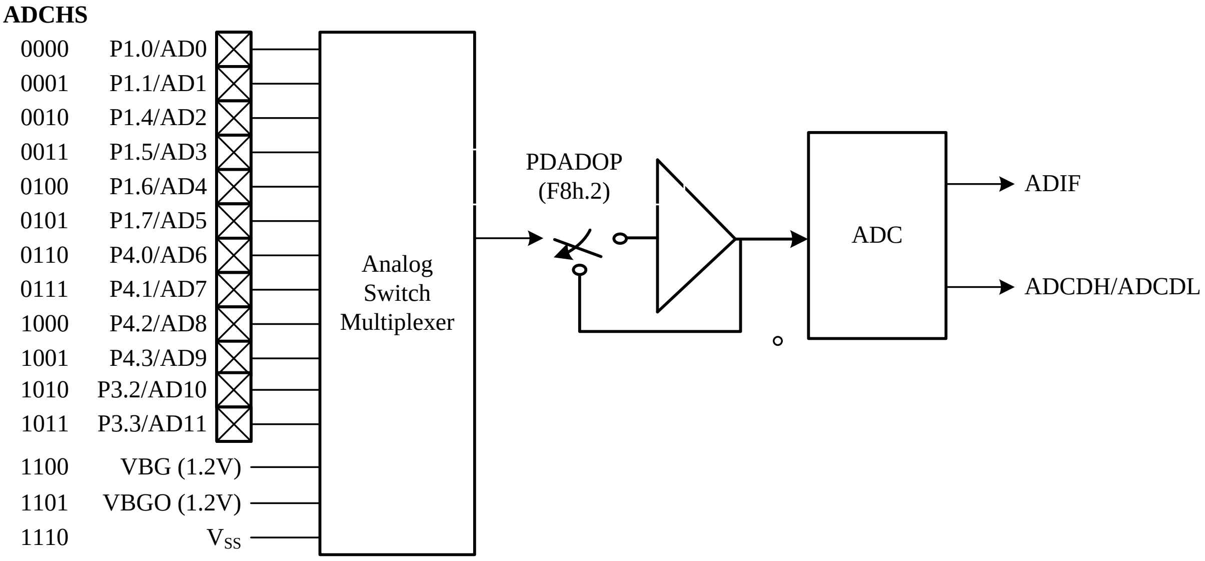
11. ADC

The **F3288** offers a 12-bit ADC consisting of a 14-channel analog input multiplexer, control register, clock generator, 12-bit successive approximation register, and output data register. To use the ADC, set the ADCKS bit first to choose a proper ADC clock frequency, which must be less than 1 MHz. Then, launch the ADC conversion by setting the ADSOC bit, and H/W will automatic clear it at the end of the conversion. After the end of the conversion, H/W will set the ADIF bit and generate an interrupt if an ADC interrupt is enabled. The ADIF bit can be cleared by writing 0 to this bit or 1 to the ADSOC bit. The analog input level must remain within the range from V_{SS} to V_{DDIO}.



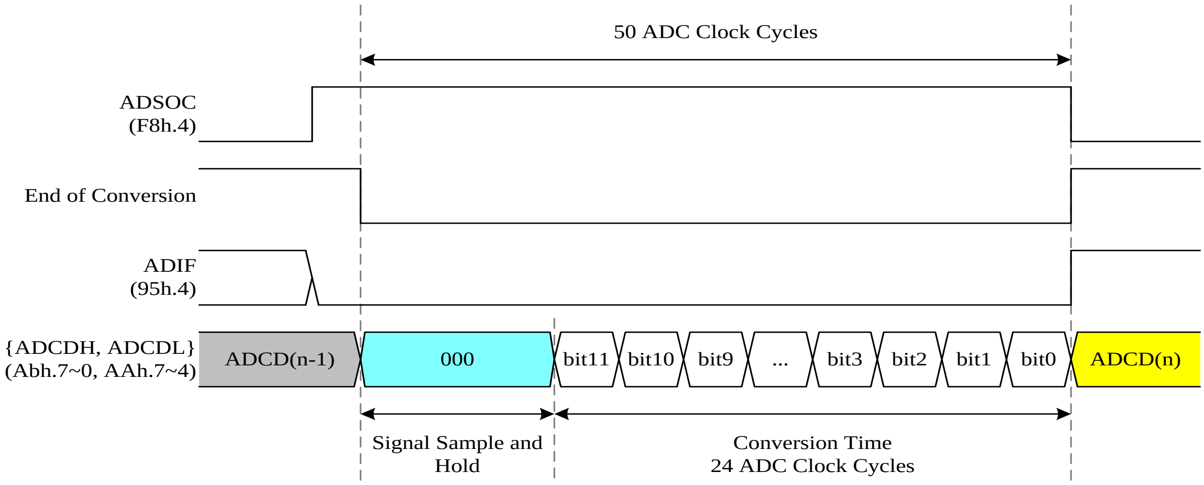
11.1 ADC Channels

The 12-bit ADC has a total of 12 channels, designated AD0~AD11, VBG, VBGO and VSS. The ADC channels are connected to the analog input pins via the analog switch multiplexer. The analog switch multiplexer is controlled by the ADCHS register. **F3288** offers up to 12 analog input pins, designated AD0~AD11. In addition, there are two analog input pins for voltage reference connections. When ADCHS is set to 1110b, the analog input will connect to V_{SS}, and when ADCHS is set to 1100b, the analog input will connect to VBG, and when ADCHS is set to 1101b, the analog input will connect to VBGO. VBG is an internal voltage reference at 1.25V, and VBGO is also the internal voltage reference at 1.25V with OP driving. If the PDADOP =0, the analog switch multiplexer output will go through the rail to rail OP-amp otherwise the multiplexer output will bypass the rail to rail OP-amp and into the ADC.



11.2 ADC Conversion Time

The conversion time is the time required for the ADC to convert the voltage. The ADC requires two ADC clock cycles to convert each bit and several clock cycles to sample and hold the input voltage. A total of 50 ADC clock cycles are required to perform the complete conversion. When the conversion time is complete, the ADIF interrupt flag is set by H/W, and the result is loaded into the ADCDH and ADCDL registers of the 12-bit A/D result.



SFR **94h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
OPTION	UART1W	MODE3V	WDTPSC		ADCKS		-	
R/W	R/W	R/W	R/W		R/W		-	
Reset	0	0	0	0	0	0	-	-

94h.3~2 **ADCKS:** ADC clock rate select

00: $F_{SYSCLK}/32$
01: $F_{SYSCLK}/16$
10: $F_{SYSCLK}/8$
11: $F_{SYSCLK}/4$

SFR **95h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTFLG	LVD	–	-	ADIF	–	IE2	P1IF	-
R/W	R	–	-	R/W	–	R/W	R/W	-
Reset	–	–	-	0	–	0	0	-

95h.4 **ADIF:** ADC interrupt flag

Set by H/W at the end of conversion. S/W writes EFh to INTFLG or sets the ADSOC bit to clear this flag.

SFR **AAh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCDL	ADCDL				-			
R/W	R				-			
Reset	–	–	–	–	-			

AAh.7~4 **ADCDL:** ADC data bit 3~0

SFR **ABh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ADCDH	ADCDH							
R/W	R							
Reset	–	–	–	–	–	–	–	–

ABh.7~0 **ADCDH:** ADC data bit 11~4

SFR **AEh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CHSEL	ADCHS				-			
R/W	R/W				-			
Reset	1	1	1	1	-	-	-	-

AEh.7~4 **ADCHS:** ADC channel select

0000: ADC0 (P1.0)
0001: ADC1 (P1.1)
0010: ADC2 (P1.4)
0011: ADC3 (P1.5)
0100: ADC4 (P1.6)
0101: ADC5 (P1.7)
0110: ADC6 (P4.0)
0111: ADC7 (P4.1)
1000: ADC8 (P4.2)
1001: ADC9 (P4.3)
1010: ADC10(P3.2)
1011: ADC11(P3.3)
1100: VBG (internal Bandgap reference voltage)
1101: VBGO(internal Bandgap reference voltage with OP driving)
1110: VSS

Note: FW must turn off Bandgap to obtain Tiny Current (ADCHS≠0b1100/0b1101)

SFR **F8h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AUX1	CLRWDT	WDTE		ADSOC	CLRPWM0	PDADOP	BOOTV	DPSEL
R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	–	0

F8h.4 **ADSOC:** Start ADC conversion

Set the ADSOC bit to start ADC conversion, and the ADSOC bit will be cleared by H/W at the end of conversion. S/W can also write 0 to clear this flag.

F8h.2 **PDADOP:** ADOP power down control

0: multiplexer output go through the R-to-R OP-amp
1: multiplexer output bypass the R-to-R OP-amp

Note: also refer to Section 6 for more information about ADC Interrupt enable and priority.
Note: also refer to Section 7 for more information about ADC pin input setting.

12. USB Interface

The USB engine includes the Serial Interface Engine (SIE), the full-speed USB I/O transceiver. The SIE block performs most of the USB interface function with only minimum support from F/W. There are 5 endpoints supported. Endpoint 0 is used to receive and transmit control (including SETUP) packets. Endpoint 1 and endpoint 2 are used for interrupt transfer. Endpoint 3 and endpoint 4 are used for bulk transfer.

The USB SIE handles the following USB bus activity independently:

1. Bit stuffing/unstuffing
2. CRC generation/checking
3. ACK/NAK
4. TOKEN type identification
5. Address checking

F/W handles the following tasks:

1. Coordinate enumeration by responding to SETUP packets
2. Fill and empty the FIFOs
3. Suspend/Resume coordination
4. Verify and select DATA toggle values

12.1 USB Device Address

The USB device address register FUNADR stores the device's address. This register is reset to all 0 after chip reset. F/W must write this register a valid value after the USB enumeration process. F/W also must to set the U2IE (A9h.5) and USB2IE (DBh.7~2) or U1IE (A9h.4) and USB1IE (DAh.7~0) to enable all of the related USB interrupt described below.

12.2 Endpoint 0 Receive (SET0/OUT0)

After receiving a SETUP packet and placing the data into the Endpoint 0 setup receive FIFO (SET0FIFO), **F3288** updates the Endpoint 0 status registers to record the receive status and then generates an Endpoint 0 setup receive interrupt (SET0I). The received data are always stored into SET0FIFO for DATA packets following SETUP token.

If received is a valid OUT packet, then generates Endpoint 0 out receive interrupt (OUT0I), data are stored into OUT0FIFO, F/W can read the status register USBCTRL, TX0CTRL and RC0REG for the recent transfer information, which includes the data byte count (OUT0CNT), packet toggle bit (RC0TGL) and data valid flag (RC0ERR). The data following an OUT token is written into OUT0FIFO and the OUT0CNT is updated unless Endpoint 0 STALL (EP0STALL) is set or Endpoint 0 receive ready (OUT0RDY) is not cleared. The data following an OUT token is written into the OUT0FIFO, and the OUT0CNT is updated unless Endpoint 0 STALL (EP0STALL) is set or Endpoint 0 receive ready (OUT0RDY) is cleared. The SIE clears the OUT0RDY automatically and generates OUT0I interrupt when the OUT0CNT or OUT0FIFO is updated. As long as the OUT0RDY is cleared, SIE keeps responding NAK to Host's Endpoint 0 OUT packet request. F/W should set the OUT0RDY flag after the OUT0I interrupt is asserted and OUT0FIFO is read out.

12.3 Endpoint 0 Transmit (TX0)

After detecting a valid Endpoint 0 IN token, **F3288** automatically transmits the data pre-stored in the Endpoint 0 transmit FIFO (TX0FIFO) to the USB bus if the Endpoint 0 transmit ready flag (TX0RDY) is set and the EP0STALL is cleared. The number of byte to be transmitted depends on the Endpoint 0 transmit byte count register (TX0CNT). The DATA0/1 token to be transmitted depends on the Endpoint 0 transmit toggle control bit (TX0TGL). After the TX0FIFO is updated, TX0RDY should be set to 1. This enables the **F3288** to respond to an Endpoint 0 IN packet. TX0RDY is cleared and an Endpoint 0 transmit interrupt (TX0I) is generated once the USB host acknowledges the data transmission. The interrupt service routine can check TX0RDY to confirm that the data transfer is successful.

12.4 Endpoint 1/2 Transmit (TX1/2)

Endpoint 1 and Endpoint 2 are capable of transmit only. These endpoints are enabled when the Endpoint 1 / Endpoint 2 configuration control bit (EP1CFG/EP2CFG) is set. After detecting a valid Endpoint 1/2 IN token, **F3288** automatically transmits the data pre-stored in the Endpoint 1/2 transmit FIFO (TX1FIFO/TX2FIFO) to the USB bus if the Endpoint 1/2 transmit ready flag (TX1RDY/TX2RDY) is set and the EP1STALL/EP2STALL is cleared. The number of byte to be transmitted depends on the Endpoint 3/4 transmit byte count register (TX1CNT/TX2CNT). The DATA0/1 token to be transmitted depends on the Endpoint 1/2 transmit toggle control bit (TX1TGL/TX2TGL). After the TX1FIFO/TX2FIFO is updated, TX1RDY/TX2RDY should be set to 1. This enables the **F3288** to respond to an Endpoint 1/2 IN packet. TX1RDY/TX2RDY is cleared and an Endpoint 1/2 transmit interrupt (TX1I/TX2I) is generated once the USB host acknowledges the data transmission. The interrupt service routine can check TX1RDY/TX2RDY to confirm that the data transfer is successful.

12.5 Endpoint 3 Transmit (TX3)

Endpoint 3 is capable of transmit only. Register TX3CTRL, TX3CNT and XRAMCTRL are used to control this endpoint. Endpoint 3 is enabled when the configuration control bit (EP3CFG) is set. To properly use this endpoint, F/W must set XRAM1USB=1 or XRAM2USB=1 to assign exactly one XRAM (XRAM1 or XRAM2) as USB Bulk In buffer. Once this endpoint is enabled, F/W should set the Toggle bit (TX3TGL) and set the transmit byte count register (TX3CNT). After detecting a valid Endpoint 3 IN token, **F3288** automatically transmits the data pre-stored in the Endpoint 3 XRAM buffer to the USB bus if the Endpoint 3 transmits ready flag (TX3RDY) is set and the EP3STALL is cleared. The number of byte to be transmitted depends on the Endpoint 3 transmit byte count register (TX3CNT). The DATA0/1 token to be transmitted depends on the Endpoint 1 transmit toggle control bit (TX3TGL). Once the USB host acknowledges the data transmission, Endpoint 3 transmit interrupt (TX3I) is generated and the TX3RDY will be cleared. The interrupt service routine can check TX3RDY to confirm that the data transfer is successful.

12.6 USB Endpoint 4 Receive (RC4)

Endpoint 4 is capable of receive only. Register RC4CTRL, RC4CNTand XRAMCTRL are used to control this endpoint. This endpoint is enable when Endpoint 4 configured control bit (EP4CFG) is set. To properly use this endpoint, F/W must set XRAM1USB=1 or XRAM2USB=1 to assign exactly one XRAM (XRAM1 or XRAM2) as USB Bulk out buffer. After detecting a valid Endpoint 4 OUT token, the **F3288** automatically stores the bulk out data into the specified Bulk out buffer and updates RC4CNT if the Endpoint 4 receiving ready flag (RC4RDY) is set and the EP4STALL is cleared. The DATA0/DATA1 token to be checked is toggled by F/W. When an Endpoint 4 receive interrupt (RC4I) is generated, the RC4RDY is cleared. During the packet transfer stage, if data are used to check error, the result will be responded on RC4ERR.

12.7 USB Control and Status

Other USB control bits include the USB enable (USBE), Suspend (SUSPND), Resume output (RSMO, Device Resistor (DEVICE_R), and corresponding interrupt enable bits. The DEVICE_R is set to enable DP pull-up resistor. Other USB status flag includes the USB reset interrupt (RSTI), Resume input interrupt (RSMI), and USB Suspend interrupt (SUSPI).

12.8 Suspend and Resume

Once the Suspend condition is asserted, F/W can set the SUSP bit to save the power consumption of USB Engine. F/W can further save the device power by forcing the CPU to go into the Power Down Mode. In the Power Down mode CPU can be waken-up by the trigger of any enabled interrupt's source or by USB bus reset or by USB bus resume. The **F3288** sends Resume signaling to USB bus when SUSPND =1 and RSMO=1.

SFR **A9h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
INTE1	-	-	U2IE	U1IE	ADIE	EX2	P1IE	SPAIE
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset	-	-	0	0	0	0	0	0

A9h.5 **U2IE:** USB2 interrupt enable
0: Disable USB2 interrupt
1: Enable USB2 interrupt

A9h.4 **U1IE:** USB1 interrupt enable
0: Disable USB1 interrupt
1: Enable USB1 interrupt

SFR **C1h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
USBADR	USBE	FUNADR						
R/W	R/W	R/W						
Reset	0	0	0	0	0	0	0	0

C1h.7 **USBE:** USB function enable bit
0: Disable P1.2/P1.3 as USB DP/DM pin
1: Enable P1.2/P1.3 as USB DP/DM pin

C1h.6~0 **FUNADR:** USB Function Address

SFR **C2h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
USB1I	SET0I	OUT0I	TX0I	TX1I	TX2I	SUSPI	TX3I	RC4I
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

C2h.7	**SET0I**	: USB Endpoint 0 SET0 Receive Interrupt flag; write 0 to clear flag
C2h.6	**OUT0I**	: USB Endpoint 0 OUT0 Receive Interrupt flag; write 0 to clear flag
C2h.5	**TX0I**	: USB Endpoint 0 Transmit Interrupt flag; write 0 to clear flag
C2h.4	**TX1I**	: USB Endpoint 1 Transmit Interrupt flag; write 0 to clear flag
C2h.3	**TX2I**	: USB Endpoint 2 Transmit Interrupt flag; write 0 to clear flag
C2h.2	**SUSPI**	: USB Suspend Interrupt flag; write 0 to clear flag
C2h.1	**TX3I**	: USB Endpoint 3 Bulk Transmit Interrupt flag; write 0 to clear flag
C2h.0	**RC4I**	: USB Endpoint 4 Bulk Receive Interrupt flag; write 0 to clear flag

SFR **C3h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
USB2I	VDD5VRI	RSTI	RSMI	KBDI	PD_TXI	PD_RCI	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	-
Reset	0	0	0	0	0	0	-	-

C3h.7 **VDD5VRI** : VDD5V Rise Interrupt flag; write 0 to clear flag
C3h.6 **RSTI** : USB Bus Reset Interrupt flag; write 0 to clear flag
C3h.5 **RSMI** : USB Resume Interrupt flag; write 0 to clear flag
C3h.3 **PD_TXI** : USB PD Transmit Interrupt flag; write 0 to clear flag
C3h.2 **PD_RCI** : USB PD Receive Interrupt flag; write 0 to clear flag

SFR **C4h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
USBCTRL	SUSPND	RSMO	EP1CFG	EP2CFG	DEVICE_R	OUT0RDY	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	-
Reset	0	0	0	0	0	0	-	-

C4h.7 **SUSPND:** USB Suspend bit
F/W set high to force USB interface into suspend mode

C4h.6 **RSMO:** USB Resume bit
F/W set high to force USB interface send RESUME signal in suspend mode

C4h.5 **EP1CFG:**
Write 1to set Endpoint 1 configed

C4h.4 **EP2CFG:**
Write 1to set Endpoint 2 configed

C4h.3 **DEVICE_R:** USB DP pull-up resistor enable bit
0: disable pull-up resistor
1: enable pull-up resistor

C4h.2 **OUT0RDY:** Endpoint 0 ready for receive control bit
0: Endpoint 0 is not ready for receive OUT packet.
1: Endpoint 0 is ready for receive from USB Host; cleared by H/W while OUT0I occurs

SFR **C5h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TX0CTRL	TX0RDY	TX0TGL	EP0STALL	IN0STALL	TX0CNT			
R/W	R/W	R/W	R/W	R/W	R/W			
Reset	0	0	0	0	0	0	0	0

C5h.7 **TX0RDY:** Endpoint 0 ready for transmit control bit
0: Endpoint 0 is not ready for IN packet.
1: Endpoint 0 is ready for transmit to USB Host; cleared by H/W while TX0I occurs

C5h.6 **TX0TGL:** Endpoint 0 transmit data type
0: DATA0
1: DATA1.

C5h.5 **EP0STALL:**
0: Endpoint 0 will not stall OUT/IN packet
1: Endpoint 0 will stall OUT/IN packet

C5h.4 **IN0STALL:**
0: Endpoint 0 will not stall IN packet
1: Endpoint 0 will stall IN packet

C5h.3~0 **TX0CNT:** Endpoint 0 transmit byte count
F/W set the transmit byte count for IN0 packet

SFR **C6h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TX1CTRL	TX1RDY	TX1TGL	EP1STALL	-	TX1CNT			
R/W	R/W	R/W	R/W	-	R/W			
Reset	0	0	0	-	0	0	0	0

C6h.7 **TX1RDY:** Endpoint 1 ready for transmit control bit
0: Endpoint 1 is not ready for IN packet.
1: Endpoint 1 is ready for transmit to USB Host; cleared by H/W while TX1I occurs

C6h.6 **TX1TGL:** Endpoint 1 transmit data type
0: DATA0.
1: DATA1.

C6h.5 **EP1STALL:**
0: Endpoint 1 will not stall IN packet
1: Endpoint 1 will stall IN packet

C6h.3~0 **TX1CNT:** Endpoint 1 transmit byte count
F/W set the transmit byte count for IN1 packet

SFR **C7h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TX2CTRL	TX2RDY	TX2TGL	EP2STALL	-	TX2CNT			
R/W	R/W	R/W	R/W	-	R/W			
Reset	0	0	0	-	0	0	0	0

C7h.7 **TX2RDY:** Endpoint 2 ready for transmit control bit
0: Endpoint 2 is not ready for IN packet.
1: Endpoint 2 is ready for transmit to USB Host; cleared by H/W while TX2I occurs

C7h.6 **TX2TGL:** Endpoint 2 transmit data type
0: DATA0
1: DATA.

C7h.5 **EP2STALL:**
0: Endpoint 2 will not stall IN packet
1: Endpoint 2 will stall IN packet

C7h.3~0 **TX2CNT:** Endpoint 2 transmit byte count
F/W set the transmit byte count for IN2 packet

SFR **CEh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TX3CTRL	TX3RDY	TX3TGL	EP3STALL	EP3CFG	-	-	-	-
R/W	R/W	R/W	R/W	R/W	-	-	-	-
Reset	0	0	0	0	-	-	-	-

CEh.7 **TX3RDY:** Endpoint 3 ready for transmit control bit
0: Endpoint 3 is not ready for IN packet.
1: Endpoint 3 is ready for transmit to USB Host; cleared by H/W while TX3I occurs

CEh.6 **TX3TGL:** Endpoint 3 transmit data type
0: DATA0
1: DATA1.

CEh.5 **EP3STALL:**
0: Endpoint 3 will not stall IN packet
1: Endpoint 3 will stall IN packet

CEh.4 **EP3CFG:**
Write 1to set Endpoint 3 configed

SFR **CFh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RC4CTRL	RC4RDY	EP4STALL	EP4CFG	RC4TGL	RC4ERR	-	-	-
R/W	R/W	R/W	R/W	R	R	-	-	-
Reset	0	0	0	-	-	-	-	-

CFh.7 **RC4RDY:** Endpoint 4 ready for receive control bit
0: Endpoint 4 is not ready for OUT packet.
1: Endpoint 4 is ready for receive from USB Host; cleared by H/W while RC4I occurs

CFh.6 **EP4STALL:**
0: Endpoint 3 will not stall OUT packet
1: Endpoint 4 will stall OUT packet

CFh.5 **EP4CFG:**
Write 1to set Endpoint 4 configed

CFh.4 **RC4TGL:** Endpoint 4 received data type
0: DATA0
1: DATA1.

CFh.3 **RC4ERR:** Endpoint 4 received data error status flag
0: Endpoint 4 received data is correct
1: Endpoint 4 received data is error

SFR **D1h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TX3CNT	-	TX3CNT						
R/W	-	R/W						
Reset	-	0	0	0	0	0	0	0

D1h.6~0 **TX3CNT:** Endpoint 3 transmit byte count
F/W set the transmit byte count for IN3 packet

SFR **D2h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RC4CNT	-	RC4CNT						
R/W	-	R						
Reset	-	-	-	-	-	-	-	-

D2h.6~0 **RC4CNT:** Endpoint 4 received byte count
F/W read the received byte count from OUT4 packet

SFR **D3h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RC0REG	RC0TGL	RC0ERR	EP0DIR	EP0SET	OUT0CNT			
R/W	R	R	R	R	R			
Reset	-	-	-	-	-	-	-	-

D3h.7 **RC0TGL:** Endpoint 0 received data type
0: DATA0
1: DATA1.

D3h.6 **RC0ERR:** Endpoint 0 received data error status flag
0: Endpoint 0 received data is correct
1: Endpoint 0 received data is error

D3h.5 **EP0DIR:** Endpoint 0 transfer direction
0: OUT/SETUP transfer
1: IN transfer

D3h.4 **EP0SET:** Endpoint 0 SETUP Token indicator
0: not SETUP Token
1: SETUP Token

D3h.3~0 **OUT0CNT:** Endpoint 0 received OUT packet byte count
F/W read the received byte count from OUT0 packet

SFR **DAh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
USB1IE	SET0IE	OUT0IE	TX0IE	TX1IE	TX2IE	SUSPIE	TX3IE	RC4IE
R/W	R/W	R/W	R/W	R/W	R/W`	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

DAh.7 **SET0IE:** Endpoint 0 SET0 Receive Interrupt enable bit
0: disable
1: enable interrupt service

DAh.6 **OUT0IE:** Endpoint 0 OUT0 Receive Interrupt enable bit
0: disable
1: enable interrupt service

DAh.5 **TX0IE:** Endpoint 0 Transmit Interrupt enable bit
0: disable
1: enable interrupt service

DAh.4 **TX1IE:** Endpoint 1 Transmit Interrupt enable bit
0: disable
1: enable interrupt service

DAh.3 **TX2IE:** Endpoint 2 Transmit Interrupt enable bit
0: disable
1: enable interrupt service

DAh.2 **SUSPIE:** USB Suspend Interrupt enable bit
0: disable
1: enable interrupt service

DAh.1 **TX3IE:** Endpoint 3 Bulk Transmit Interrupt enable bit
0: disable
1: enable interrupt service

DAh.0 **RC4IE:** Endpoint 4 Bulk receive Interrupt enable bit
0: disable
1: enable interrupt service

SFR **DBh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
USB2IE	VDD5VRIE	RSTIE	RSMIE	KBDIE	PD_TXIE	PD_RCIE	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	-
Reset	0	0	0	0	0	0	-	-

DBh.7 **VDD5VRIE:** USB Bus power VDD5V Rise Interrupt enable bit
0: disable
1: enable interrupt service

DBh.6 **RSTIE:** USB Bus Reset Interrupt enable bit
0: disable
1: enable interrupt service

DBh.5 **RSMIE:** USB Resume Interrupt enable bit
0: disable
1: enable interrupt service

12.9 USB Keyboard

The **F3288** supports Keyboard keyscan function. This function has a dedicated interrupt (KBDI) and can choose KSI[7:0] pin to do interrupt service, without interrupt of service to mask by KBMASK. The matrix key scan pin KSI[7:0] is shared with Port0[7:0], KSO[7:0] is shared with Port2[7:0], KSO[14:8] is shared with Port4[6:0] and KSO[17:16] is shares with Port3[7:5].

SFR **DBh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
USB2IE	VDD5VRIE	RSTIE	RSMIE	KBDIE	PD_TXIE	PD_RCIE	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	-
Reset	0	0	0	0	0	0	-	-

DBh.4 **KBDIE:** Keyboard Scan Interrupt enable bit
0: disable
1: enable interrupt service

SFR **C3h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
USB2I	VDD5VRI	RSTI	RSMI	KBDI	PD_TXI	PD_RCI	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	-
Reset	0	0	0	0	0	0	-	-

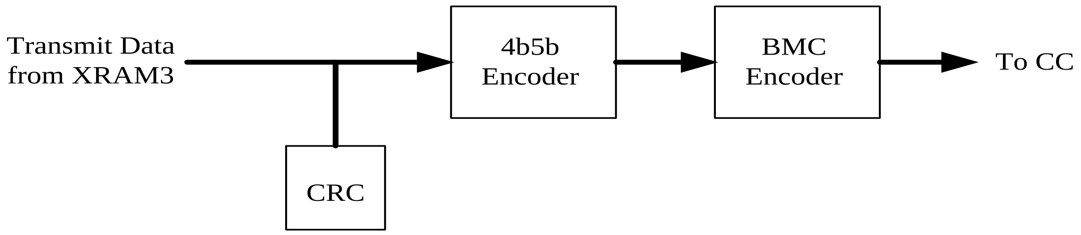
C3h.4 **KBDI** **:** Keyboard Interrupt flag; write 0 to clear flag

SFR **B6h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
KBMASK	KBMASK							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

B6h Mask KSI[7:0] interrupt (KBDI) function while the corresponding bit is "1"

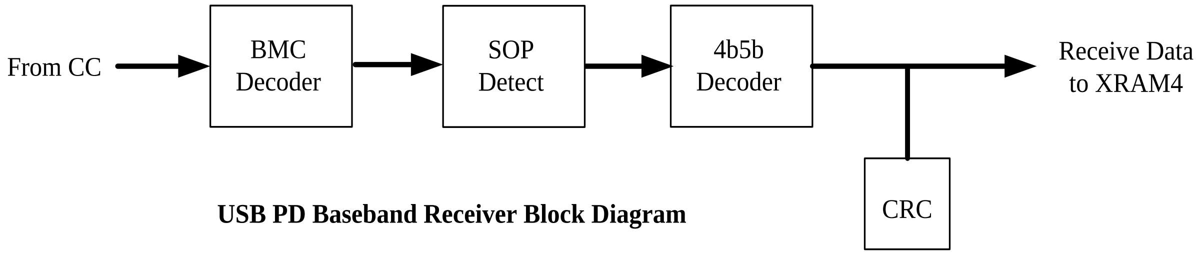
12.10 USB PD

F3288 implements USB PD physical layer to support USB Power Delivery Protocol R2.0 V1.2. **USBPDEN** bit must be set to high when **F3288** is used in USB PD mode. The USB PD messages are transmitted or received in a USB Type-C system using a BMC signaling. The BMC signal is output/input on the same CC pin (CC1 or CC2). **F3288** USB PD module starts to transmit the message when the PDTXRDY bit is set to high. **F3288** MCU need to fill the PD protocol message in the XRAM3 buffer before **F3288** USB PD module transmit message to the other device via CC pin and PD_TXI will be generated to inform MCU that the PD message is transmitted completed.



USB PD Baseband Transmitter Block Diagram

F3288 USB PD module is ready to receive the message when the PDRXRDY bit is set to high. The received USB PD message from the other device via CC pin is stored in the XRAM4 buffer. If the received data is CRC correct, the PD_RCI interrupt flag will be generated and inform the MCU to read the received USB PD message.



USB PD Baseband Receiver Block Diagram

The USB PD packet shall consist of a Preamble, an SOP*, packet data including the Message Header, a CRC and an EOP. The packet format is shown bellow. The entire packet shall be transmitted by using BMC over CC pin.

Preamble	SOP* (Start Of Packet)	Message Header (16 bit)	0..7 Data Object(s)	CRC	EOP (End Of Packet)

The Preamble shall consist of 64-bit sequence of alternating 0s and 1s. The Preamble shall start with a "0" and shall end with a "1".

The SOP (Start of Packet) is used as an addressing scheme to identify whether the Communications were intended for one of the Port partners (SOP Communication) or one of the Cable Plug (SOP' / SOP'' Communication)

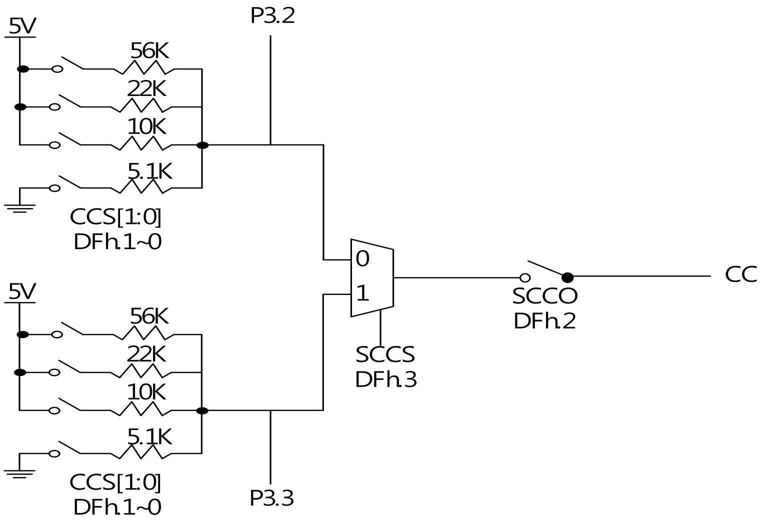
All Message shall be composed of a Message Header and a variable length (include zero) data portion. Every message shall start with a 16-bit Message Header. The Message Header contains basic information about the Message and the USB PD Port Capability. The Message Header may be used standalone as a Control Message when the Number of Data Object field is zero or as the first part of Data Message when the Number of Data Object field is non-zero.

Message Header

Bit(s)	Start of Packet	Field Name
15	N/A	Reserved
14..12	SOP*	Number of Data Objects
11..9	SOP*	Message ID
8	SOP only	Port Power Role
	SOP'/ SOP''	Cable Plug
7..6	SOP*	Specification Reversion
5	SOP only	Port Data Role
	SOP'/ SOP''	Reserved
4	N/A	Reserved
3..0	SOP*	Message Type

The Message Header and data shall be protected by a 32-bit CRC. CRC-32 protects the data integrity of the data payload.CRC-32 calculation shall begin at byte 0 bit 0 and continue to bit 7 of each of the bytes of the packet.

When P3.2 (CC1) or P3.3 (CC2) worked in USB PD mode, the P3.2 or P3.3 must set to MODE3 and SCC bit is used to select CC1 (P3.2) or CC2 (P3.3) as CC pin.



SFR **DBh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
USB2IE	VDD5VRIE	RSTIE	RSMIE	KBDIE	PD_TXIE	PD_RCIE	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	-
Reset	0	0	0	0	0	0	-	-

DBh.3 **PD_TXIE:** USB PD Transmit Interrupt enable bit
0: disable
1: enable interrupt service

DBh.2 **PD_RCIE:** USB PD Receive Interrupt enable bit
0: disable
1: enable interrupt service

SFR **DFh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDCTRL	USBPDEN	PDTXRDY	PDRXRDY	SCC	SCCO2	SCCO1	CCS	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

DFh.7 **USBPDEN:** USB PD mode enable bit
0: disable
1: enable USB PD mode

DFh.6 **PDTXRDY:** USB PD ready for transmit control bit
0: not ready
1: USB PD is ready for transmit PD message

DFh.5 **PDRXRDY:** USB PD ready to receive control bit
0: not ready
1: USB PD is ready to receive PD message

DFh.4 **SCC:** Select CC1 or CC2 as CC pin
0: select P3.2 (CC1) as CC pin
1: select P3.3 (CC2) as CC pin

DFh.3 **SCCO2:** Resistor connect to CC2 pin control bit
0: disconnect the resistor to CC2 pin
1: connect the resistor to CC2 pin

DFh.2 **SCCO1:** Resistor connect to CC1 pin control bit
0: disconnect the resistor to CC1 pin
1: connect the resistor to CC1 pin

DFh.1~0 **CCS:** CC pin connect resistor select
00: 5.1K ohm to Ground
01: 10K ohm to VDD5
10: 22K ohm to VDD5
11: 56K ohm to VDD5

XRAM3 and XRAM4 can be accessed by MCU or by USB PD module depending on the register XRAMCTRL. The data stored in XRAM3/XRAM4 is the USB PD packet.

SFR **D6h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
XRAMCTRL	XRAM4UPD	XRAM3UPD	XRAM1USB	XRAM2USB	XRAM1SPB	XRAM2SPB	XRAM1I80	XRAM2I80
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

D6h.7 **XRAM4UPD:** XRAM4 access control bit
0: XRAM4 is access by MCU
1: XRAM4 is access by USB PD Receive message function

D6h.6 **XRAM3UPD:** XRAM3 access control bit
0: XRAM3 is access by MCU
1: XRAM3 is access by USB PD Transmit message function

XRAM3/XRAM4

addr.	data	description
0	kcode1	**Start of Packet Sequence**
1	kcode2	
2	kcode3	
3	kcode4	
4	head0	**Message Header [7:0]**
5	head1	**Message Header[15:8]**
6	dat0	**Data Object 1**
7	dat1	
8	dat2	
9	dat3	
10	dat4	**Data Object 2**
11	dat5	
12	dat6	
13	dat7	
14	dat8	**Data Object 3**
15	dat9	
16	dat10	
17	dat11	
18	dat12	**Data Object 4**
19	dat13	
20	dat14	
21	dat15	
22	dat16	**Data Object 5**
23	dat17	
24	dat18	
25	dat19	
26	dat20	**Data Object 6**
27	dat21	
28	dat22	
29	dat23	
30	dat24	**Data Object 7**
31	dat25	
32	dat26	
33	dat27	
34	crc0	**CRC-32**
35	crc2	
36	crc3	
37	crc4	

Note. Please refer to USB Power Delivery Protocol R2.0 V1.2 section 5 & section 6 for Start of Packet Sequence
Note. Please refer to USB Power Delivery Protocol R2.0 V1.2 section 6 for Message Header and Data Object

13. Serial Peripheral Interface

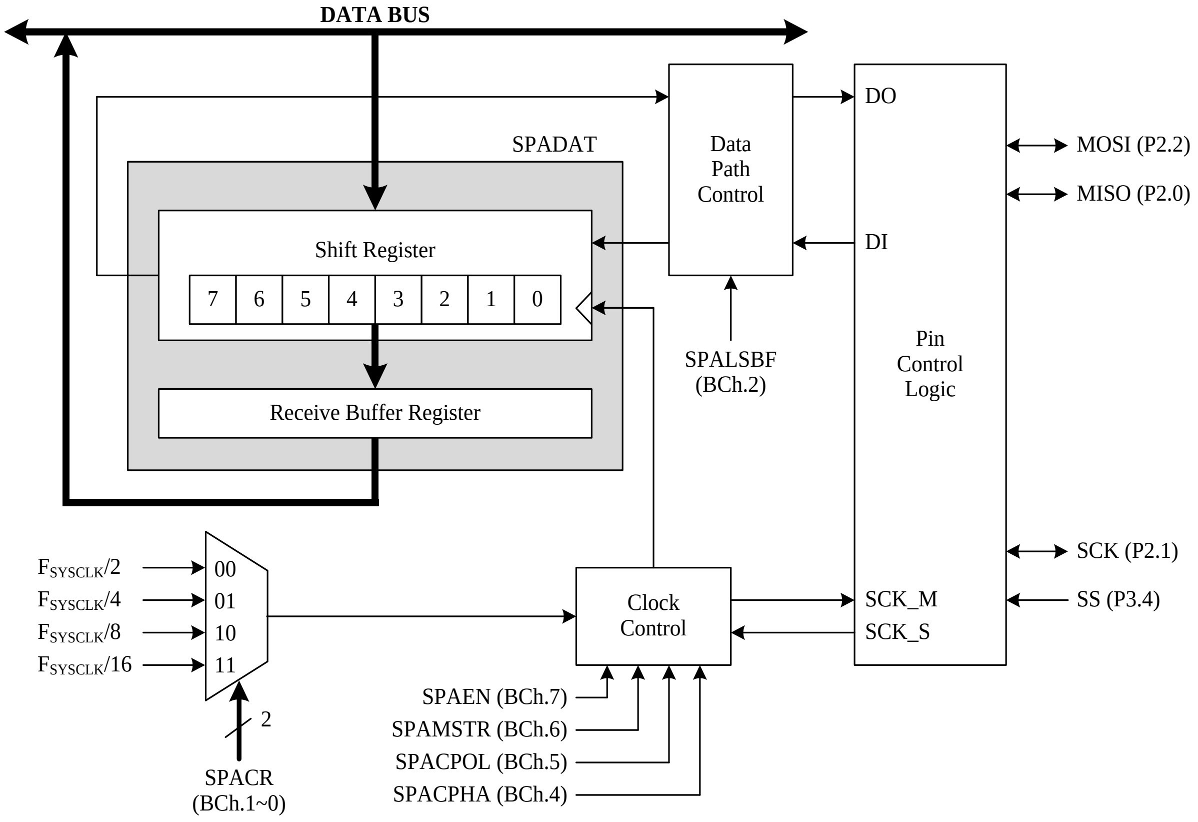
The **F3288** has two SPI modules. One SPI module (SPIA) is support for master or slave mode, the other one (SPIB) is support master mode only.

13.1 Serial Peripheral Interface (SPIA)

The Serial Peripheral Interface module A (SPIA) is capable of full-duplex, synchronous, serial communication between the MCU and peripheral devices. The peripheral devices can be other MCUs, A/D converter, sensors, or flash memory, etc. The SPI runs at a clock rate up to the system clock divided by two. Firmware can read the status flags, or the operation can be interrupt driven. Following figure shows the SPI system block diagram.

The features of the SPIA module include:

- Master or Slave mode operation
- 3-wire or 4-wire mode operation
- Full-duplex operation
- Programmable transmit bit rate
- Single buffer receive
- Serial clock phase and polarity options
- MSB-first or LSB-first shifting selectable



The four signals used by SPIA are described below. The MOSI (P2.2) signal is an output from a Master Device and an input to Slave Devices. The MISO (P2.0) signal is an output from a Slave Device and an input to a Master Device. Data is transferred most-significant bit (MSB) or least-significant bit (LSB) first by setting the SPALSBF bit. The SCK (P2.1) signal is an output from a Master Device and an input to Slave Devices. It is used to synchronize the data on the MOSI and MISO lines of Master and Slave. SPIA generates the signal with eight programmable clock rates in Master mode. The SS (P3.4) signal is a low active slave select pin. In 4-wire Slave mode, the signal is ignored when the Slave is not selected (SS=1). The SS is ignored when the SPASSDIS in SPACON is set in both Master and Slave modes. In Slave mode and the SPASSDIS is clear, the SPIA active when SS stay low. For multiple-slave mode, only one slave device is selected at a time to avoid bus collision on the MISO line. In Master mode and the SSDIS is cleared, the MODF in SPASTA is set when this signal is low. For multiple-master mode, enable SS line to avoid multiple driving on MOSI and SCK lines from multiple masters.

13.1.1 Master Mode

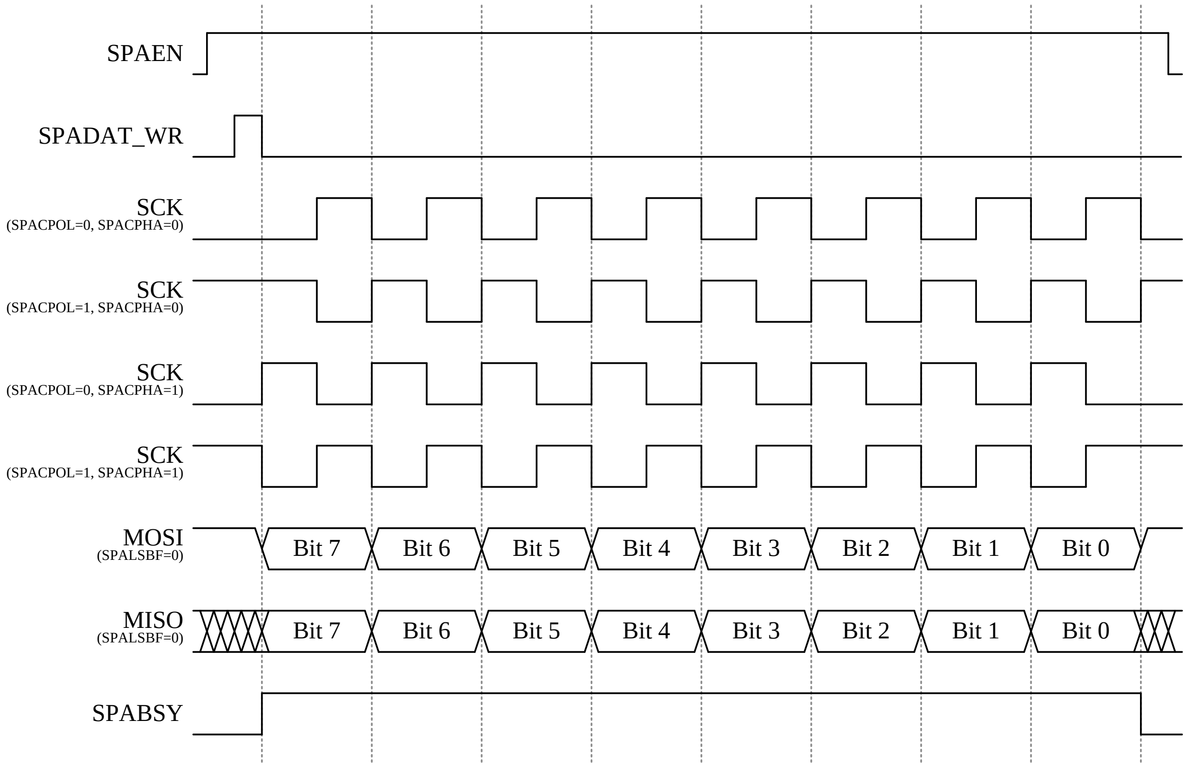
The SPIA operates in Master mode by setting the SPAMSTR bit in the SPACON. To start transmit, writing a data to the SPADAT. If the SPABSY bit is cleared, the data will be transferred to the shift register and starts shift out on the MOSI line. The data of the slave shift in from the MISO line at the same time. When the SPAIF bit in the SPASTA becomes set at the end of the transfer, the receive data is written to receiver buffer and the SPARCVBF bit in the SPASTA is set. To prevent an overrun condition, software must read the SPADAT before next byte enters the shift register. The SPABSY bit will be set when writing a data to SPADAT to start transmit, and be cleared at the end of the eighth SCK period in Master mode.

13.1.2 Slave Mode

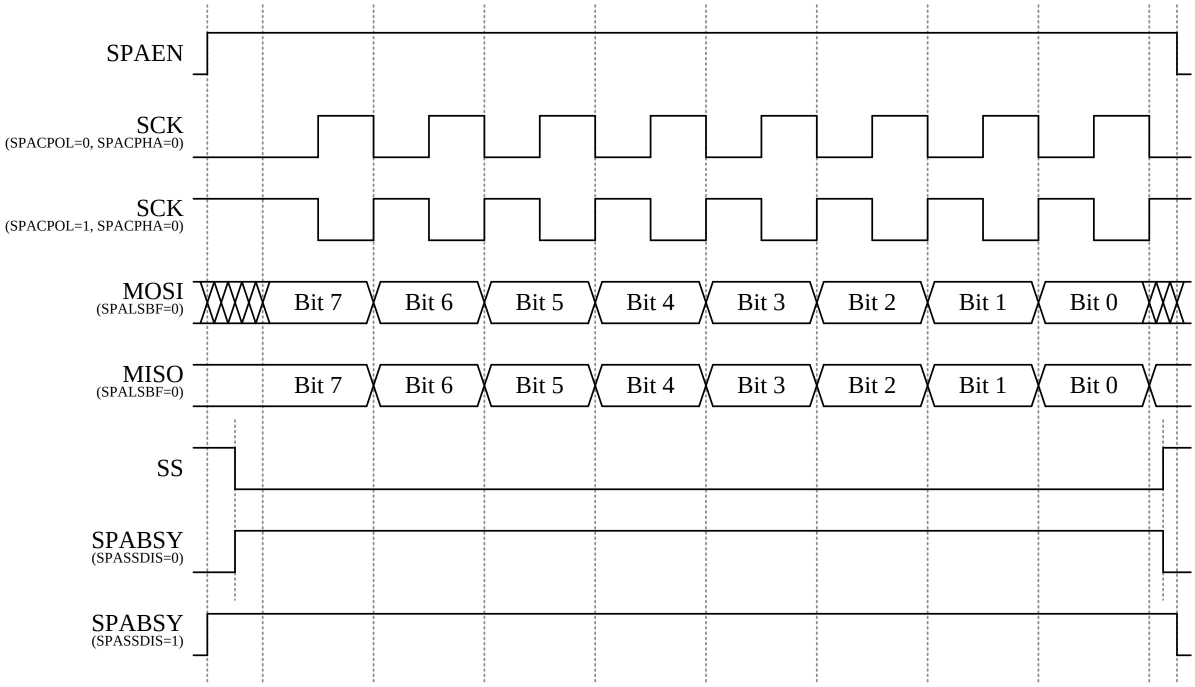
The SPIA operates in Slave mode by clearing the SPAMSTR bit in the SPACON. If the SPASSDIS is cleared, the transmission will start when the SS become low and remain low until the end of a data transfer. If the SPASSDIS is set, the transmission will start when the SPAEN bit in the SPACON is set, and don’t care the SS. The data from a master will shift into the shift register through the MOSI line, and shift out from the shift register on the MISO line. When a byte enters the shift register, the data will be transferred to receiver buffer if the SPARCVBF is cleared. If the SPARCVBF is set, the newer receive data will not be transferred to receiver buffer and the SPARCVOVF bit is set. After a byte enters the shift register, the SPAIF and SPARCVBF bits are set. To prevent an overrun condition, software must read the SPADAT or write 0 to SPARCVBF before next byte enters the shift register. The maximum SCK frequency allowed in Slave mode is $F_{SYSCLK}/4$. In Slave mode, the SPBSY bit refers to the SS pin when the SPASSDIS bit is cleared, and refer to the SPAEN bit when SPASSDIS bit is set.

13.1.3 Serial Clock

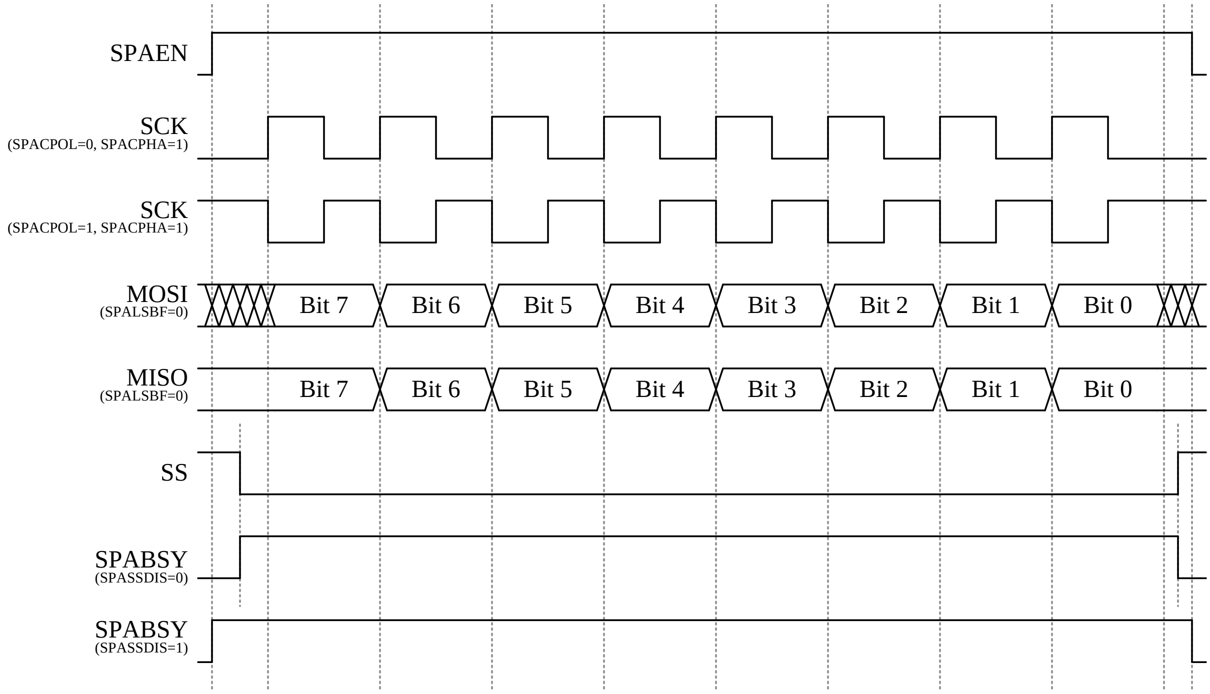
The SPI has four clock types by setting the SPACPOL and SPACPHA bits in the SPACON register. The SPACPOL bit defines the level of the SCK in SPIA idle state. The level of the SCK in idle state is low when the SPACPOL bit is cleared, and is high when the SPACPOL bit is set. The SPACPHA bit defines the edges used to sample and shift data. The SPIA sample data on the first edge of SCK period and shift data on the second edge of SCK period when the SPACPHA bit is cleared. The SPIA sample data on the second edge of SCK period and shift data on first edge of SCK period when the SPACPHA bit is set. The figures below show the detail timing in Master and Slave modes. Both Master and Slave devices must be configured to use the same clock type before the SPASPEN bit is set. The SPASPCR controls the Master mode serial clock frequency. This register is ignored when operating in Slave mode. The SPIA clock can select System clock divided by 2, 4, 8, or 16 in Master mode.


SPAEN
SPADAT_WR
SCK (SPACPOL=0, SPACPHA=0)
SCK (SPACPOL=1, SPACPHA=0)
SCK (SPACPOL=0, SPACPHA=1)
SCK (SPACPOL=1, SPACPHA=1)
MOSI (SPALSBF=0)
MISO (SPALSBF=0)
Bit 7
Bit 6
Bit 5
Bit 4
Bit 3
Bit 2
Bit 1
Bit 0
SPABSY

Master Mode Timing



Slave Mode Timing (SPACPHA=0)



Slave Mode Timing (SPACPHA=1)

In both Master and Slave modes, the SPAIF bit is set by H/W at the end of a data transfer and generates an interrupt if SPIA interrupt is enabled. The SPAIF bit is cleared automatically when the program performs the interrupt service routines. S/W can also write 0 to clear this flag. If write data to SPADAT when the SPABSY is set, the SPAWCOL bit will be set by H/W and generates an interrupt if SPIA interrupt is enabled. When this occurs, the data write to SPADAT will be ignored, and shift register will not be written. Write 0 to this bit or when SPABSY is cleared and rewrite data to SPADAT will clear this flag. The SPAMODF bit is set when SPASSDIS is cleared and SS pin is pulled low in Master mode. If SPIA interrupt is enabled, an interrupt will be generated. When this bit is set, the SPAEN and SPAMSTR in SPACON will be cleared by H/W. Write 0 to this bit will clear this flag.

SFR **BCh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPACON	SPAEN	SPAMSTR	SPACPOL	SPACPHA	SPASSDIS	SPALSBF	SPASPCR	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

BCh.7 **SPAEN:** SPIA enable
0: SPIA disable
1: SPIA enable

BCh.6 **SPAMSTR:** SPIA Master mode enable
0: Slave mode
1: Master mode

BCh.5 **SPACPOL:** SPIA clock polarity
0: SCK is low in idle state
1: SCK is high in idle state

BCh.4 **SPACPHA:** SPIA clock phase
0: Data sample on first edge of SCK period
1: Data sample on second edge of SCK period

BCh.3 **SPASSDIS:** SS pin disable
0: Enable SS pin
1: Disable SS pin

BCh.2 **SPALSBF:** SPIA LSB first
0: MSB first
1: LSB first

BCh.1~0 **SPASPCR:** SPIA clock rate
00: $F_{SYSCLK}/2$
01: $F_{SYSCLK}/4$
10: $F_{SYSCLK}/8$
11: $F_{SYSCLK}/16$

SFR **BDh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPASTA	SPAIF	SPAWCOL	SPAMODF	SPARCVOVF	SPARCVBF	SPABSY	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	–	–
Reset	0	0	0	0	0	0	–	–

BDh.7 **SPAIF:** SPIA interrupt flag
This is set by H/W at the end of a data transfer. Cleared by H/W when an interrupt is vectored into. Writing 0 to this bit will clear this flag.

BDh.6 **SPAWCOL:** Write collision interrupt flag
Set by H/W if write data to SPADAT when SPABSY is set. Write 0 to this bit or rewrite data to SPADAT when SPABSY is cleared will clear this flag.

BDh.5 **SPAMODF:** SPIA Mode fault interrupt flag
Set by H/W when SPASSDIS is cleared and SS pin is pulled low in Master mode. Write 0 to this bit will clear this flag. When this bit is set, the SPAEN and SPAMSTR in SPACON will be cleared by H/W.

BDh.4 **SPARCVOVF:** SPIA Received buffer overrun flag
Set by H/W at the end of a data transfer and SPARCVBF is set. Write 0 to this bit or read SPADAT register will clear this flag.

BDh.3 **SPARCVBF:** SPIA Receive buffer full flag
Set by H/W at the end of a data transfer. Write 0 to this bit or read SPADAT register will clear this flag.

BDh.2 **SPABSY:** SPIA busy flag
Set by H/W when a SPIA transfer is in progress.

SFR **BEh**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPADAT	SPADAT							
R/W	R/W							
Reset	0	0	0	0	0	0	0	0

BEh.7~0 **SPADAT:** SPIA transmit and receive data
The SPADAT register is used to transmit and receive data. Writing data to SPADAT place the data into shift register and start a transfer when in master mode. Reading SPADAT returns the contents of the receive buffer.

13.2 Serial Peripheral Interface B (SPIB)

Serial Peripheral Interface B (SPIB) is a synchronous serial data protocol, which is used for communicating with one or more peripheral slave devices on short distances. The **F3288**'s SPIB module is capable of full-duplex, synchronous, serial communication between MCU and peripheral devices. The peripheral devices can be other MCUs (slave mode), LCD module, Mouse IR sensor, A/D converter, MEMS sensors, SPI EEPROM or SPI flash memory, etc… The features of the SPIB module include the Master operation, Full-duplex operation, Programmable transmit bit rate, Serial clock phase and polarity options, receive data buffers (up to 64 bytes), transmit data buffer (up to 64 bytes), data transfer DMA mode between SPI to USB or I80 interface options. DMALEN register is used to inform the H/W how many byte data will be transferred to/from XRAM1/2.

13.2.1 SPIB Functional Description

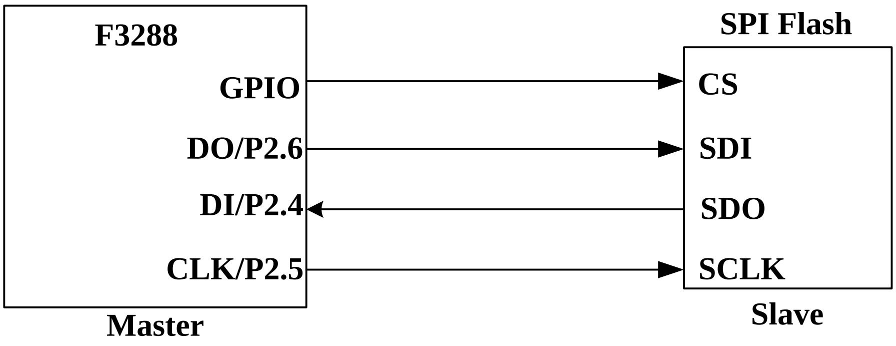


Figure above is the connection diagram between SPIB master device and slave device. The master device initiates all SPI data transfers, but it must follow slave device SPI data transfer specification, otherwise it cannot communicate to slave device. During a transfer, the master device transmits data to the slave device SDI input pin, while the slave device receives data from the master device DO output pin at the same time. When the master device receives data from the slave device SDO output pin, the slave device transmits data to the master device DI input pin at the same time. The CLK signal is a clock output to slave device from the master device, whether to transmit or receive data. During the SPI transfer, data is read only on the DI pin at one CLK edge and shifted, changing the bit value on the DO pin to output data, one-half CLK cycle later.

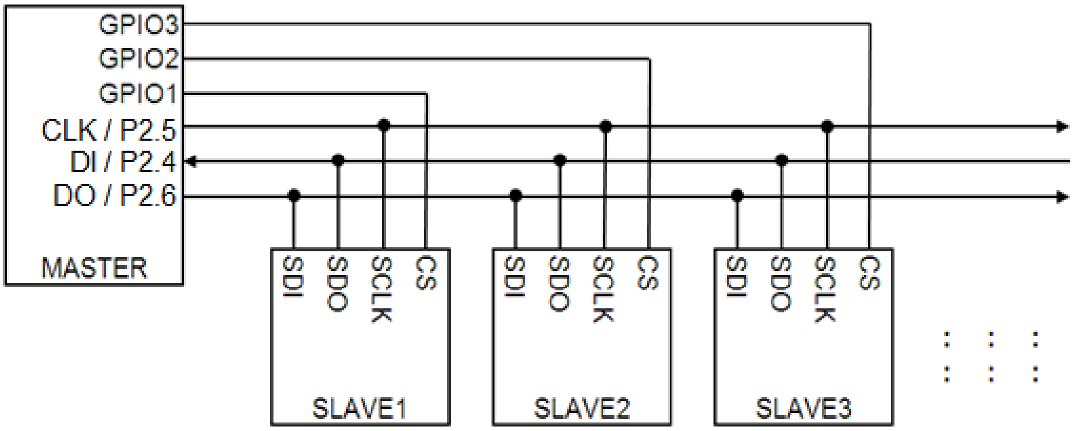
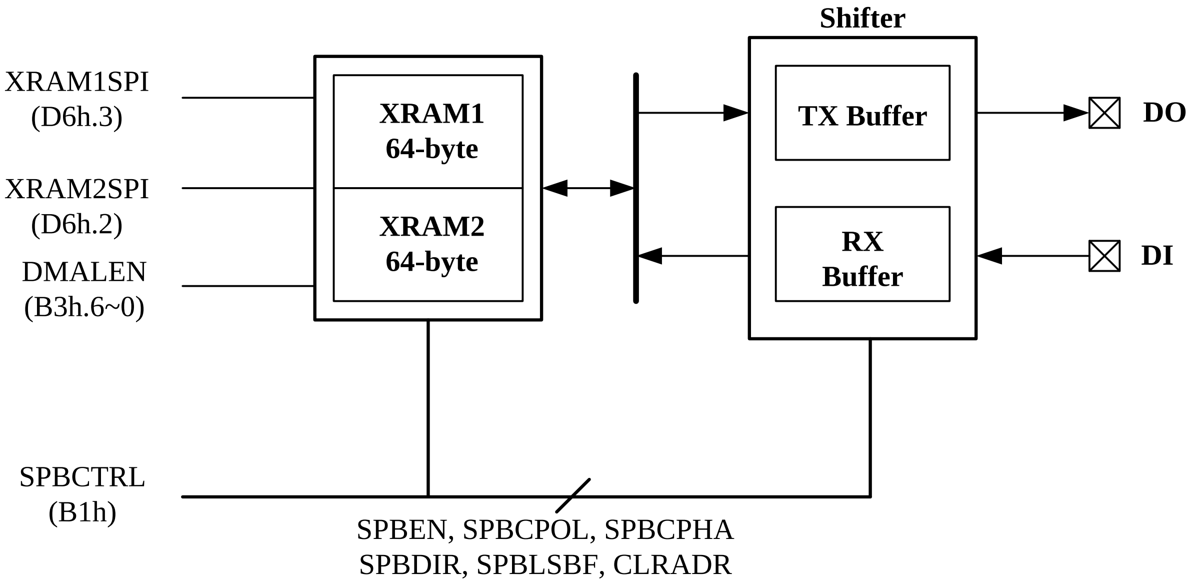


Figure above shows that the system contains one master and at least 2 slaves, DO pin of master connects to SDI pins of all slaves, DI pin of master connects to SDO pins of all slaves, CLK pins are connected each other. CS pins of slave should be connected by GPIO pins of the master.

Master device uses F/W in GPIO pins to access these slave devices (Chip Select). The CS in slave device is controlled by master device when GPIO pin goes low. The chip select pin is necessary when master device needs to connect multiple slave devices. FW can use GPIO to select multiple slave devices in CS pin, all of slave device can be chosen at one time, and when chip select pin (CS) is set low, that means this device is chosen to do data transfer.

13.2.2 SPIB System Block diagram and Register Control

The SPBMOD register indicates the IO Pin P2.6~4 is in GPIO or in SPI function. The SPIB transmit data is stored in XRAM1 or XRAM2. SPIB can select XRAM1 or XRAM2 according to XRAM1SPI bit or XRAM2SPI bit. Only one of these two bits can be set “1” at the mean time, since SPIB can access only one data source. SPIB can continuous access XRAM1 or XRAM2 until the designated length (DMALEN) is transferred completely. When SPBDIR bit is set to “1”, the SPIB will receive data from SPI device and stored to XRAM1 or XRAM2 via RX Buffer. When SPBDIR bit is set to “0”, the SPIB will transmit data to SPI device from XRAM1 or XRAM2 via TX Buffer. The SPBCTRL register is used to control the SPIB module. F/W sets SPBEN register to start the SPI data transmission. The SPBLSF register which indicates the SPI data transmission is LSB first or MSB first. The CLRADR register is used to clear the SPIB XRAM buffer address to 0.



SFR **B1h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPBCTRL	SPBMOD	SPBEN	SPBCPOL	SPBCPHA	SPBDIR	SPBLSBF	-	CLRADR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	R/W
Reset	0	0	0	0	0	0	-	0

B1h.7 **SPBMOD:** SPIB mode
0: P2.6~4 in GPIO
1: P2.6~4 in SPIB I/F function

B1h.6 **SPBEN:** SPIB transfer enable
0: automatically clears when data transfer complete
1: start transfer data and check busy state

B1h.5 **SPBCPOL:** SPIB clock polarity
0: BSCK is low in idle state
1: BSCK is high in idle state

B1h.4 **SPBCPHA:** SPIB clock phase
0: Data sample on first edge of BSCK period
1: Data sample on second edge of BSCK period

B1h.3 **SPBDIR:** SPIB transfer direction
0: transmit data to SPI device
1: receive data from SPI device

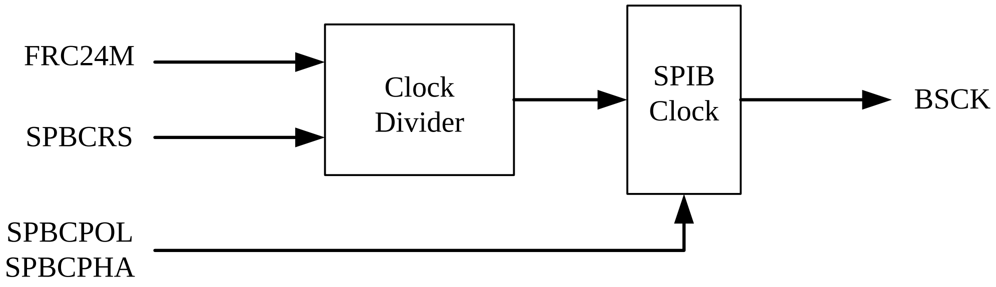
B1h.2 **SPBLSBF:** SPIB transfer LSB first
0: MSB first
1: LSB first

B1h.0 **CLRADR:** SPIB clear XRAM address
Set 1 to clear buffer address

13.2.3 SPIB Clock and Data Format

The **F3288**’s SPIB module can be used as master only. The clock rate and data transfer length are also adjustable. The figure below shows the SPI system block diagram. The SPIB clock output is divided by SPBCRS register. There are 6-bit prescaler to generate desired SPIB baud clock. The fastest bit rate is SPBCRS=000, it means the transfer bit rate is (FRC24M clock) / 2. The SPBCRS[6..0] is used to set SPIB clock select register.

SPIB clock rate = FRC24M / (2 * (SPICRS[6..0] + 1))



SFR **B2h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPBCRS	-	SPBCRS						
R/W	-							
Reset	-	0	0	0	0	0	0	0

B2h.6~0 **SPBCRS:** SPIB Clock select

SPICRS(0): 12 MHz	SPICRS(1): 6 MHz	SPICRS(2):4 MHz	SPICRS(3): 3 MHz
SPICRS(4): 2.4 MHz	SPICRS(5): 2 MHz	SPICRS(6): 1.71MHz	SPICRS(7): 1.5 MHz

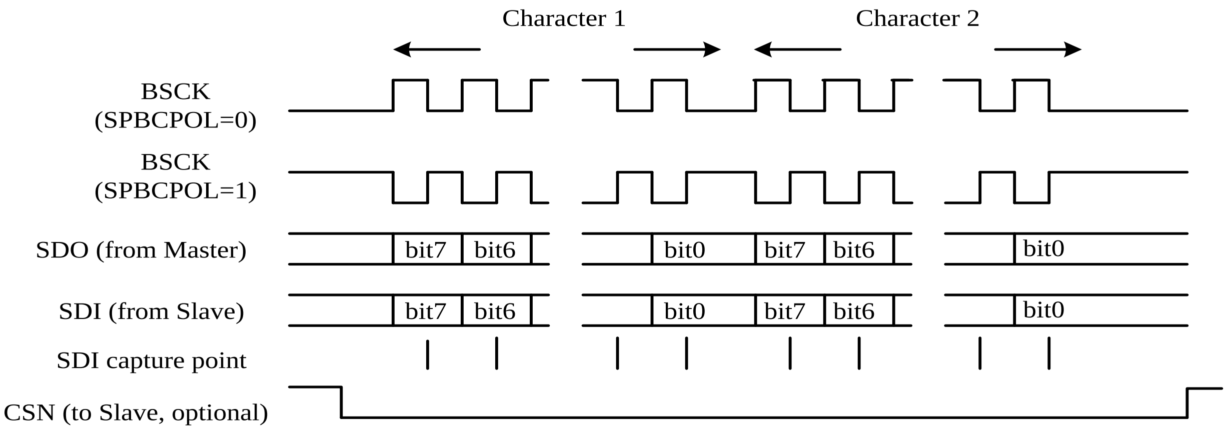
…….

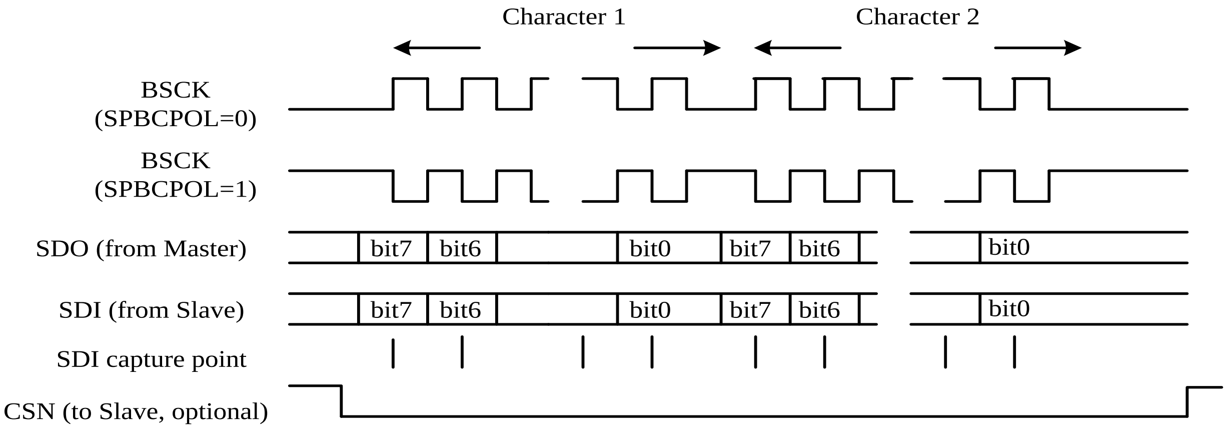
SFR **B3h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DMALEN	-	DMALEN						
R/W	-							
Reset	-	0	0	0	0	0	0	0

B3h.6~0 **DMALEN:** DMA data transfer length

The figure below shows that before transfer data needs to figure which registers in device control which functions, it also sets the clock frequency (SPBCRS), data transfer length (DMALEN), configures the clock polarity (SPBCPOL) and phase (SPBCPHA) with respect to the data. These modes control data in and out on the rising or falling edge of the data clock signal. The two modes combine polarity and phase.

SPIBTiming





The figure above shows the four modes of SPIB clock and data format. CS will be asserted low while a transfer is started. SPBCPOL is the BSCK pin priority select. BSCK pin is in a logic high state while not transferring if SPBCPOL=1, on the other hand, BSCK pin is in a logic low while not transferring if SPBCPOL=0.

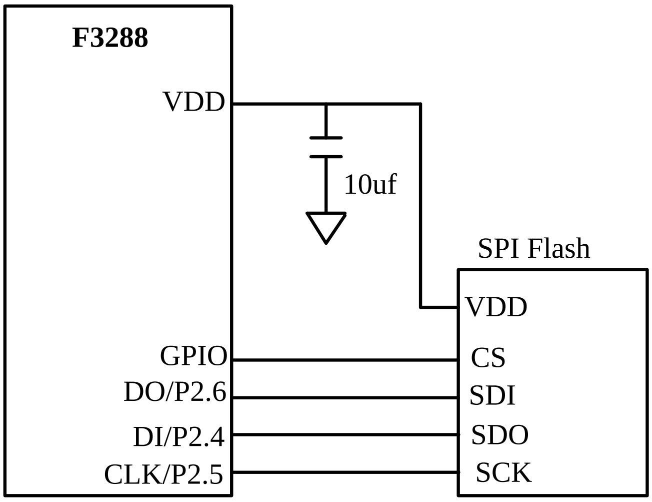
When SPBCPHA=0, the first edge of BSCK samples the data into TX shifter. The successive bit is placed on master’s DO pin at the second edge of BSCK. In summary, when SPBCPHA=0, the slave samples the data bits on odd number of BSCK clock edge, and the even number of BSCK edge is the data preparation time of the master device. **When SPBCPHA=1**, The DO pin is in unknown level until the first edge of BSCK is coming. When the first edge of BSCK is coming, the master device places the data output on DO. The slave device uses the second edge of BSCK to sample the data into RX shifter. The successive bit is placed on master’s DO pin at the third edge of

BSCK. In summary, when SPBCPHA=1, the slave samples the data bits on even number of BSCK clock edge, while the odd number of BSCK clock edge is the data preparation time of the master.

Mode	SPBCPOL	SPBCPHA	Description
1	0	0	The base value of the clock is 0 and the data are captured on the clock's rising edge and data are propagated on a falling edge.
2	0	1	The base value of the clock is 0 and the data are captured on the clock's falling edge and data are propagated on a rising edge.
3	1	0	The base value of the clock is 1 and the data are captured on clock's falling edge and data are propagated on a rising edge.
4	1	1	The base value of the clock is 1 and the data are captured on clock's rising edge and data are propagated on a falling edge.

When Date transfer IN/OUT on the rising or falling edge of the data clock signal calls the clock phase (SPBCPHA), the clock is idle when high or low is called at the clock polarity (SPBCPOL). The two modes combine polarity and phase.

13.2.4 SPIB Power Circuit



The **F3288** has a built-in low current output regulator (max.50mA) which is designed to provide 3.3V (VDD) from a USB 5V supply. This regulator is ideally suited output current for external device in 3.3V logic. The output capacitor is critical to maintain regulator stability, and must meet the required conditions of minimum amount of capacitance. The minimum output capacitance is required to maintain stability is 10µF. Larger values of output capacitance will give improved transient response.

13.2.5 SPIB DMA Transfer Mode

Generally, data transfer length is according to how many bytes data will be transferred. The length value is stored in **DMALEN** and the bulk transfer data is stored in the XRAM 64 bytes (XRAM1 or XRAM2). The **F3288** supports DMA mode between USB and I80

SFR **D6h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
XRAMCTRL	XRAM4UPD	XRAM3UPD	XRAM1USB	XRAM2USB	XRAM1SPB	XRAM2SPB	XRAM1I80	XRAM2I80
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

D6h.5 **XRAM1USB:** XRAM1 access control bit
1: assign XRAM1 as USB Bulk transfer buffer. *Note 1*

D6h.4 **XRAM2USB:** XRAM2 access control bit
1: assign XRAM2 as USB Bulk transfer buffer. *Note 2*

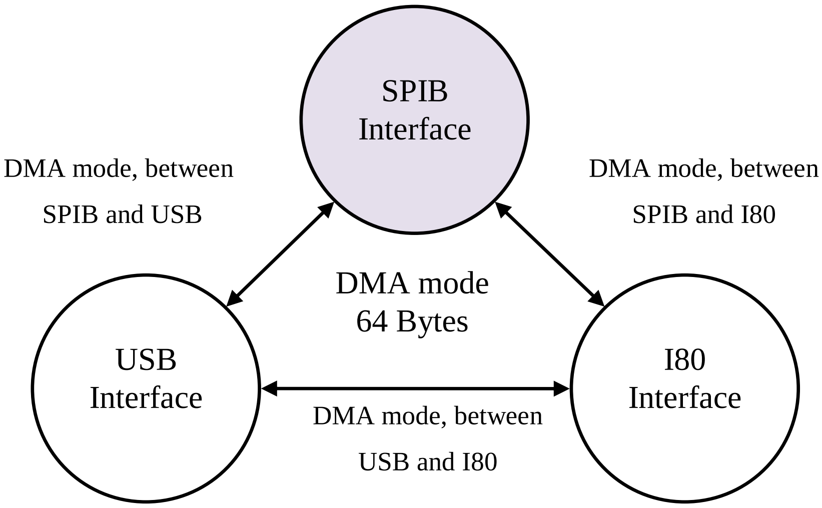
D6h.3 **XRAM1SPB:** XRAM1 access control bit
1: assign XRAM1 as SPIB DMA transfer buffer. *Note 1*

D6h.2 **XRAM2SPB:** XRAM2 access control bit
1: assign XRAM2 as SPIB DMA transfer buffer. *Note 2*

Note 1 : If MCU need to access XRAM1, F/W must clear { XRAM1USB, XRAM1SPB, XRAM1I80}=3'b000
Note 2 : If MCU need to access XRAM2, F/W must clear { XRAM2USB, XRAM2SPB, XRAM2I80}=3'b000

The **XRAMCTRL** register indicates XRAM1/XRAM2 data transfer direction to USB I/F, I80 I/F and SPIB I/F. The **XRAM1SPI** bit is used to assign the XRAM1 as SPIB I/F DMA Transfer buffer. The **XRAM2SPI** is used to assign the XRAM2 as SPIB I/F DMA Transfer buffer.

Figure below shows the data transfer DMA mode between SPIB to USB or I80 interface.



13.2.6 SPIB Initial Sample Code

Figure below shows the sample code of SPIB initialization, initializes the SPIB firmware code, making it ready to communicate to the SPI Slave Device. After reset the SPIB and all registers to default state, register initialization is completed, which indicating the XRAMCTRL register stored in the XRAM1 or XRAM2.

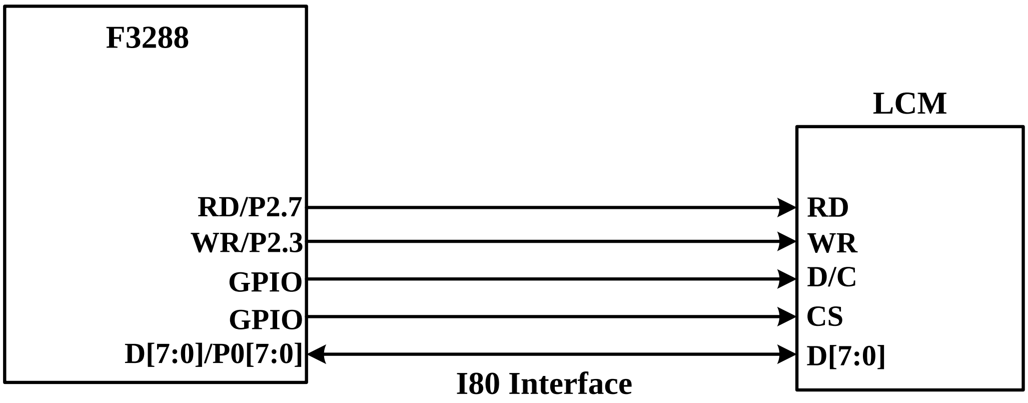
Example : SPIB Sample Code

```
;=============================================================
;Function:  master only
;=============================================================
Start:
        .
        .
SPIBTX:
        mov     SPBCTRL, #81h     ; SPIB module in SPIB mode
                                  ; clear xram address to zero
        mov     SPBCTRL, #80h     ;
        mov     DMALEN,  #40h     ; set DMA transfer length =40h = 64-byte
        mov     XRAMCTRL, #08h    ; XRAM1 as SPIB DMA transfer buffer
        CS_LO                     ; enable chip-select to slave device pin
        mov     SPBCTRL, #d0h        ; SPBCPOL=0, SPBCPHA=1,
                                  ; and start to transmit data from XRAM1 to
                                  ; slave device
CHKSPB:
        mov     A, SPBCTRL
        jb      A.6, CHKSPB       ; F/W polling SPBEN register bit ,
                                  ; if transfer complete, goto next instruction

        mov     XRAMCTRL, #04h    ; XRAM2 as SPIB DMA transfer buffer
        mov     SPBCTRL, #d0h     ; SPBCPOL=0, SPBCPHA=1,
                                  ; and start to transmit data from XRAM1 to
                                  ; slave device
CHK2SPB:
        mov     A, SPBCTRL
        jb      A.6, CHK2SPB      ; F/W polling SPBEN register bit ,
                                  ; if transfer complete, goto next instruction
        CS_HI                     ; disable chip-select
        .                         ;
        ret
        .
        END                     ; end of user program
```

14. I80 Peripheral Interface (I80)

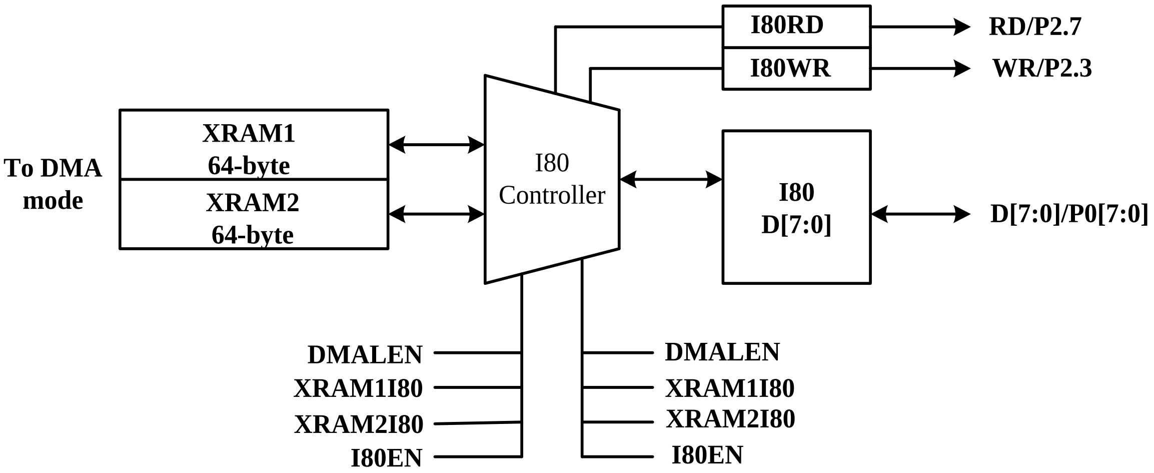
The **F3288** supports I80 interface for LCD Display module, NAND-Flash of parallel bus, NOR-Flash of parallel bus and printer of parallel bus, etc… The I80 interface is an 8-bit parallel bus with read (RD) and write (WR) control line, master only, and data transfer DMA mode.

14.1 I80 Functional Description



In I80 Parallel Interface, the WR pin is used as Write selection output. The Data write operation is initiated; output this pin to low to transmit data or command on D[7:0] bus. The RD pin is used as Read selection output. The Data read operation is initiated; output this pin to low to receive the data or command on D[7:0] bus. The **F3288** uses GPIO control D/C and CS pin. The D/C is a Data and Command control pin and CS is chip enable control pin. In D/C control pin, when it is active HIGH, the input at D7-D0 is treated as Data. When active is LOW, the input at D[7:0] is transferred to the command registers. These D[7:0] are 8-bit bi-directional data bus to be connected to the slave device data bus. The **F3288**'s I80 interface is master only to send data from the MCU.

14.2 I80 System Block Diagram and Register Control



The I80 interface is enabled by setting the **I80CTRL** (D4h) control register and the **DMALEN** (B3h) setting data length register. The DMALEN indicates how many bytes data will be transmitted to device or received from device. The I80 interface has maximum 64-byte data transfer buffer XRAM. The I80CTRL contains the I80BUSY, I80EN, I80START and I80DIR control register. The **I80EN** – (D4h.2) is an enable control bit which is used to enable the I80 DMA mode. When I80EN is set to high, Port0[7:0] becomes the I80 DATA bus D[7..0] and Port 2.7 becomes the I80 RD signal and Port 2.3 becomes the I80 WR signal. The **I80DIR** (D4h.0) is the data transfer direction control bit. When I80DIR is set to low, the data will be transmitted to device, and when it is set to high, the data will be received from device. The **I80START** (D4h.1) is the read/write start control bit. When I80START is set to high, the I80 controller will start to transfer data. If the I80 completes the data transfer, the I80START will be cleared to 0. The I80BUSY (D4.3) is an I80 interface state of data transfer. If the I80BUSY is read as high, it means the I80 interface is transferring data. If it is read as low, the I80 interface is idle.

SFR **D4h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
I80CTRL	-	-	-	-	I80BUSY	I80EN	I80START	I80DIR
R/W	-	-	-	-	R	R/W	R/W	R/W
Reset	-	-	-	-	-	0	0	0

D4h.3 **I80BUSY:** I80 Interface state of data transfer
0: I80 interface is idle
1: I80 interface is busy

D4h.2 **I80EN:** Enable I80 I/F DMA mode
0: disable I80 DMA mode
1: enable I80 DMA mode

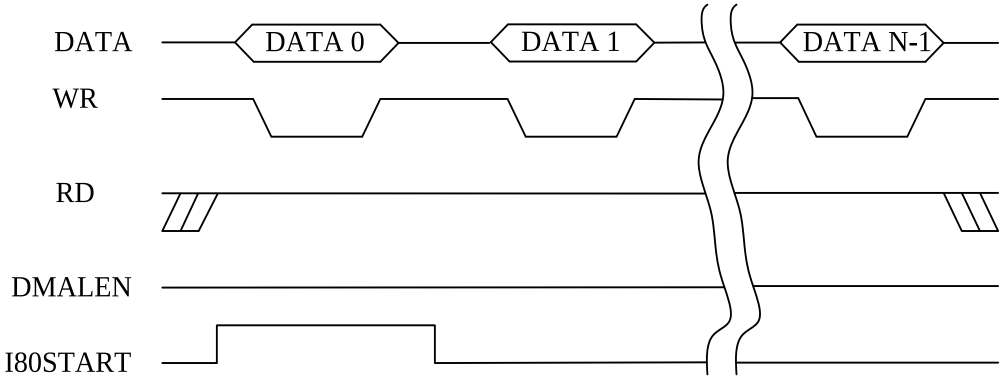
D4h.1 **I80START:** I80 I/F data transfer start bit
0: write 0 to clear it if data transfer completes
1: write 1 to set it to start data transfer

D4h.0 **I80DIR:** I80 I/F data transfer direction
0: write data to device
1: read data from device

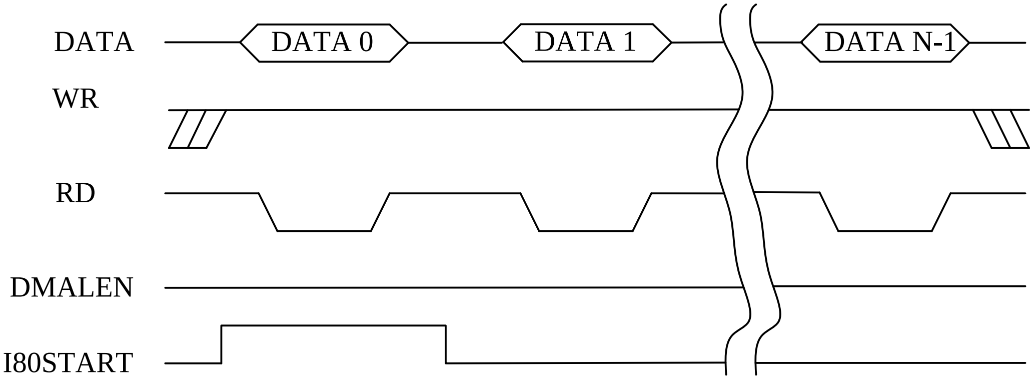
14.3 I80 Clock and Data Format

The **F3288**'s I80 is an 8-bit Hardware Interface. When I80 interface is selected, the Write Data Format is set by I80DIR (D4h.0). The written data is expanded into 64-byte internally and then written into XRAM1 or XRAM2.

I80 Timing for data Write



I80 Timing for data Read



14.4 I80 DMA Transfer Mode

The **F3288**'s I80 interface is a parallel 8-bit data transfer. The bulk transfer data is stored in the XRAM 64 bytes (XRAM1 or XRAM2). The **F3288** supports DMA mode between USB and I80. The **XRAM1I80** register is a control bit, which is used to assign the XRAM1 as I80 I/F DMA Transfer buffer. The **XRAM2I80** register is a control bit, which is used to assign the XRAM2 as I80 I/F DMA Transfer buffer.

SFR **D6h**	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
XRAMCTRL	XRAM4UPD	XRAM3UPD	XRAM1USB	XRAM2USB	XRAM1SPB	XRAM2SPB	XRAM1I80	XRAM2I80
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

D6h.5 **XRAM1USB:** XRAM1 access control bit
1: assign XRAM1 as USB Bulk transfer buffer. *Note 1*

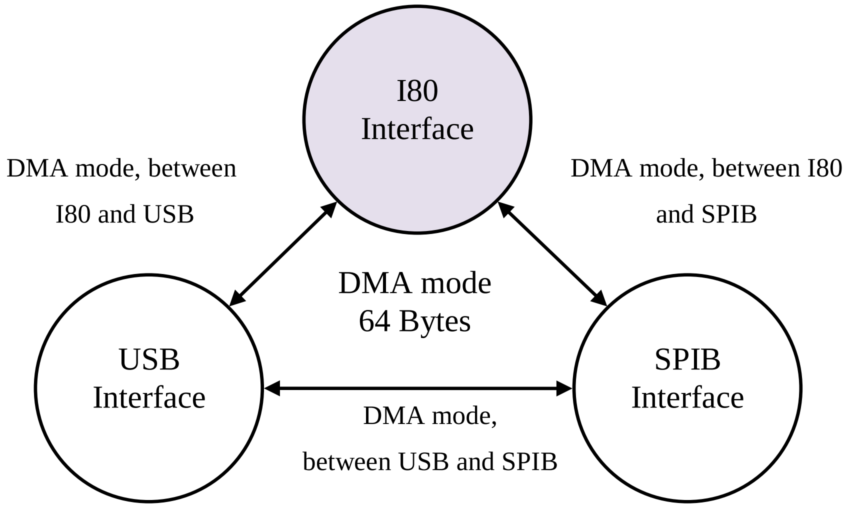
D6h.4 **XRAM2USB:** XRAM2 access control bit
1: assign XRAM2 as USB Bulk transfer buffer. *Note 2*

D6h.1 **XRAM1I80:** XRAM1 access control bit
1: assign XRAM1 as I80 DMA transfer buffer. *Note 1*

D6h.0 **XRAM2I80:** XRAM2 access control bit
1: assign XRAM2 as I80 DMA transfer buffer. *Note 2*

Note 1 : If MCU need to access XRAM1, F/W must clear { XRAM1USB, XRAM1SPB, XRAM1I80}=3'b000
Note 2 : If MCU need to access XRAM2, F/W must clear { XRAM2USB, XRAM2SPB, XRAM2I80}=3'b000

Figure below shows the data transfer DMA mode between I80 to USB or SPIB interface.



14.5 I80 Initial Sample Code

Figure below shows the sample code of I80 initialization, initializes the I80 firmware code, making it ready to communicate to the I80 Device. After reset the I80 and all registers to default state, register initialization is completed, indicating the XRAMCTRL register stored in the XRAM1 or XRAM2. In order to receive the data packet, status and control register is set to enable command and data configuration according to hardware.

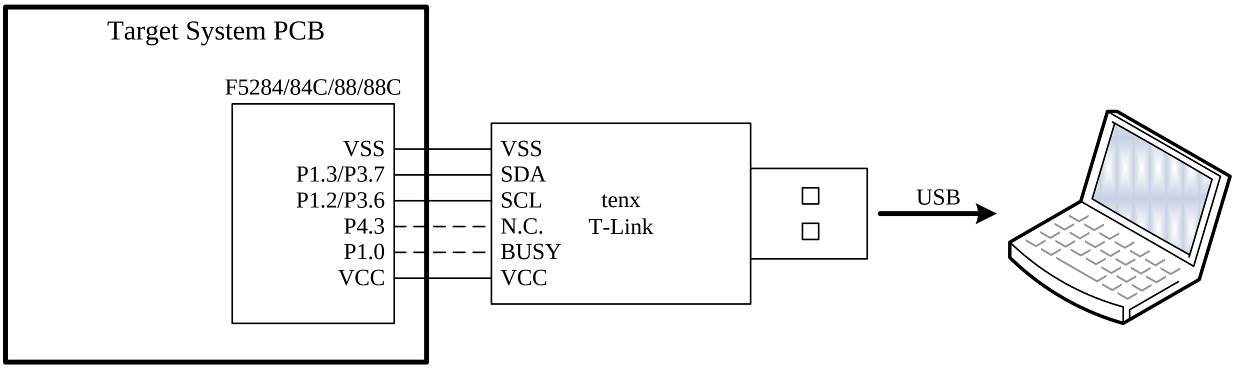
Example : I80 Sample Code

```
;=============================================================
;Function:
;=============================================================
I80TX:
         mov     XRAMCTRL, #02h  ; XRAM1 as I80 I/F DMA transfer buffer
         mov     DMALEN,  #40h    ; set DMA transfer length =40h = 64-byte
         mov     I80CTRL, #06h    ; Enable I80 DMA , and start to transfer data to device
         mov     I80CTRL, #04h    ; clear start
CHKI80W:
         mov     A, I80CTRL
         jb      A.3, CHKI80W     ; F/W polling I80BUSY register bit ,
                                  ; if transfer complete, goto next instruction
         mov     I80CTRL, #00h     ; disable I80 I/F DMA mode

I80RX:
         mov     XRAMCTRL, #01h  ; XRAM2 as I80 I/F DMA transfer buffer
         mov     DMALEN,  #40h    ; set DMA transfer length =40h = 64-byte
         mov     I80CTRL, #07h    ; Enable I80 DMA , and start to receive data from device
         mov     I80CTRL, #05h    ; clear start

CHKI80R:
         mov     A, I80CTRL
         jb      A.3, CHKI80R      ; F/W polling I80BUSY register bit ,
                                  ; if transfer complete, goto next instruction
         mov     I80CTRL, #00h     ; disable I80 I/F DMA mode                    ;
         ret
```

15. In Circuit Emulation (ICE) Mode

This device can support the In Circuit Emulation Mode. To use the ICE Mode, user just needs to connect P1.2 and P1.3 or P3.6 and P3.7 pin to the tenx proprietary EV Module. The benefit is that user can emulate the whole system without changing the on board target device. But there are some limits for the ICE mode as below.

1. The device must be un-protect.
2. The device's P1.2 and P1.3 or P3.6 an P3.7 pins must work in input Mode (P1MOD2=0/1 and P1MOD3=0/1 or P3MOD6=0/1 and P3MOD7=0/1).
3. During Program Code download, P4.3 is controlled by T-Link unit and P1.0 sent acknowledge signal to T-Link unit. After download stage, P4.3 and P1.0 can be emulated as any other pins.
4. During Program Code download, P1.1 output FRC/4 and P1.4 always output Low. After download stage, P1.1 and P1.4 can be emulated as any other pins.
5. The Program Memory's addressing space 1D00h~1FFFh and 0033h~003Ah are occupied by tenx EV Module. So user Program cannot access these spaces.
6. The P1.2 and P1.3 or P3.6 and P3.7 pin's function cannot be emulated.
7. When the device is used in USB application, P1.2 and P1.3 ares used as USB DP and DM, only P3.6 and P3.7 can be used for ICE communication



SFR & CFGW MAP

Adr	Rst	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
80h	1111-1111	**P0**	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
81h	0000-0111	**SP**	SP							
82h	0000-0000	**DPL**	DPL							
83h	0000-0000	**DPH**	DPH							
87h	0xxx-0000	**PCON**	SMOD	–	–	–	GF1	GF0	PD	IDL
88h	0000-0000	**TCON**	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
89h	0000-0000	**TMOD**	GATE1	CT1N	TMOD1		GATE0	CT0N	TMOD0	
8Ah	0000-0000	**TL0**	TL0							
8Bh	0000-0000	**TL1**	TL1							
8Ch	0000-0000	**TH0**	TH0							
8Dh	0000-0000	**TH1**	TH1							
90h	1111-1111	**P1**	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
91h	0000-0000	**P0OE**	P0OE							
93h	0000-0000	**P2OE**	P2OE							
94h	0000-0000	**OPTION**	UART1W	MODE3V	WDTPSC		ADCKS		-	-
95h	xx00-x000	**INTFLG**	LVDO	-	-	ADIF	-	IE2	P1IF	-
96h	0000-0000	**P1WKUP**	P1WKUP							
97h	xxxx-xxx0	**SWCMD**	IAPALL / SWRST							
98h	0000-0000	**SCON**	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
99h	xxxx-xxxx	**SBUF**	SBUF							
9Ah	1111-1111	**PWM0PRD**	PWM0PRD							
9Bh	1000-0000	**PWM0DH**	PWM0DH							
9Ch	1111-1111	**PWM1PRD**	PWM1PRD							
9Dh	1000-0000	**PWM1ADH**	PWM1ADH							
9Eh	1000-0000	**PWM1BDH**	PWM1BDH							
9Fh	1000-0000	**PWM1CDH**	PWM1CDH							
A0h	1111-1111	**P2**	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
A1h	1000-1000	**PWMCKS**	PWM1CKS		-		PWM0CKS		-	-
A2h	0000-0000	**P1MODL**	P1MOD3		P1MOD2		P1MOD1		P1MOD0	
A3h	0000-0000	**P1MODH**	P1MOD7		P1MOD6		P1MOD5		P1MOD4	
A4h	0000-0000	**P3MODL**	P3MOD3		P3MOD2		P3MOD1		P3MOD0	
A5h	0000-0000	**P3MODH**	P3MOD7		P3MOD6		P3MOD5		P3MOD4	
A6h	0000-0000	**PINMOD**	P5OE0	PWM0POE	PWM0NOE	T2OE	T0OE	PWM1AOE	PWM1BOE	PWM1COE
A7h	xx00-0000	**PWM0CON**	-	-	PWM0MODE		PWM0NOVT			
A8h	0x00-0000	**IE**	EA	-	ET2	ES	ET1	EX1	ET0	EX0
A9h	0000-0000	**INTE1**			U2IE	U1IE	ADIE	EX2	P1IE	SPAIE
AAh	xxxx-xxxx	**ADCDL**	ADCDL				-	-	-	-
ABh	xxxx-xxxx	**ADCDH**	ADCDH							
ACh	0000-0000	**PWMDL**	PWM0DL		PWM1ADL		PWM1BDL		PWM1CDL	
AEh	1111-1111	**CHSEL**	ADCHS				-			
B0h	1111-1111	**P3**	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
B1h	0000-0000	**SPBCTRL**	SPBMOD	SPBEN	SPBCPOL	SPBCPHA	SPBDIR	SPBLSF	-	CLRADR
B2h	x000-0000	**SPBCRS**	-	SPBCRS						
B3h	x000-0000	**DMALEN**	-	DMALEN						
B4h	0000-0000	**P4MODL**	P4MOD4		P4MOD2		P4MOD1		P4MOD0	

Adr	Rst	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
B5h	0000-0000	**P4MODH**	P4MOD7		P4MOD6		P4MOD5		P4MOD4	
B6h	0000-0000	**KBMASK**	KBMASK							
B8h	xx00-0000	**IP**	-	-	PT2	PS	PT1	PX1	PT0	PX0
B9h	xx00-0000	**IPH**	-	-	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
BAh	xxx0-0000	**IP1**	-	-	-	PSPI	PADTKI	PX2	PP1	PT3
BBh	xxx0-0000	**IP1H**	-	-	-	PSPIH	PADTKIH	PX2H	PP1H	PT3H
BCh	0000-0000	**SPACON**	SPAEN	SPAMSTR	SPACPOL	SPACPHA	SPASSDIS	SPALSBF	SPASPCR	
BDh	0000-0xxx	**SPASTA**	SPAIF	SPAWCOL	SPAMODF	SPARCVOVF	SPARCVBF	SPABSY	-	-
BEh	0000-0000	**SPADAT**	SPDAT							
C0h	1111-1111	**P5**	-	-	-	-	-	-	-	P5.0
C1h	0000-0000	**USBADR**	USBE	FUNADR						
C2h	0000-0000	**USB1I**	SET0I	OUT0I	TX0I	TX1I	TX2I	SUSPI	TX3I	RC4I
C3h	0000-0000	**USB2I**	VDD5VRI	RSTI	RSMI	KBDI	PD_TXI	PD_RCI	-	-
C4h	0000-0000	**USBCTRL**	SUSPND	RSMO	EP1CFG	EP2CFG	DEVICE_R	OUT0RDY	-	-
C5h	0000-0000	**TX0CTRL**	TX0RDY	TX0TGL	EP0STALL	IN0STALL	TX0CNT			
C6h	0000-0000	**TX1CTRL**	TX1RDY	TX1TGL	EP1STALL	-	TX1CNT			
C7h	0000-0000	**TX2CTRL**	TX2RDY	TX2TGL	EP2STALL	-	TX2CNT			
C8h	0000-0000	**T2CON**	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	CT2N	CPRL2N
C9h	0xxx-xxxx	**IAPWE**	IAPWE	IAPTO	-	-	-	-	-	
CAh	0000-0000	**RCP2L**	RCP2L							
CBh	0000-0000	**RCP2H**	RCP2H							
CCh	0000-0000	**TL2**	TL2							
CDh	0000-0000	**TH2**	TH2							
CEh	0000-xxxx	**TX3CTRL**	TX3RDY	TX3TGL	EP3STALL	EP3CFG	-	-	-	-
CFh	000x-xxxx	**RC4CTRL**	RC4RDY	EP4STALL	EP4CFG	RC4TGL	RC4ERR	-	-	-
D0h	0000-0000	**PSW**	CY	AC	F0	RS1	RS0	OV	F1	P
D1h	x000-0000	**TX3CNT**	-	TX3CNT						
D2h	xxxx-xxxx	**RC4CNT**	-	RC4CNT						
D3h	xxxx-xxxx	**RC0REG**	RC0TGL	RC0ERR	EP0DIR	EP0SET	OUT0CNT			
D4h	xxxx-x000	**I80CTRL**	-	-	-	-	I80BUSY	I80EN	I80START	I80DIR
D6h	0000-0000	**XRAMCTRL**	XRAM4UPD	XRAM3UPD	XRAM1USB	XRAM2USB	XRAM1SPB	XRAM2SPB	XRAM1I80	XRAM2I80
D8h	xx00-0011	**CLKCON**	-	-	STPSCK	STPPCK	STPFCK	SELFCK	CLKPSC	
DAh	0000-0000	**USB1IE**	SET0IE	OUT0IE	TX0IE	TX1IE	TX2IE	SUSPIE	TX3IE	RC4IE
DBh	0000-00xx	**USB2IE**	VDD5VRIE	RSTIE	RSMIE	KBDIE	PD_TXIE	PD_RCIE	-	-
DCh	xxxx-x000	**USBRSTM**	-	-	-	-	-	UTMEN	UTMS	
DFh	00xx-0000	**PDCTRL**	USBPDEN	PDTXRDY	PDRXRDY	SCC	SCCO2	SCCO1	CCS	
E0h	0000-0000	**ACC**	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0
E8h	1111-1111	**P4**	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0
EFh	xxxx-x000	**ANAT**	-	-	-	-	-	ANTOE	SANTO	
F0h	0000-0000	**B**	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0
F5h	00xx-x000	**AUX2**	VDD5VFLG	VDD5FALL	-	-	-	IAPTE		VCCFLT
F6h	xxxx-xxxx	**VBGTM**	-	-	-	-	VBGTM			
F7h	xxxx-xxxx	**FRCTM**	-	-	-	FRCTM				
F8h	0000-00x0	**AUX1**	CLRWDT	WDTE		ADSOC	CLRPWM0	PDADOP	BOOTV	DPSEL

Flash Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
3FFCh	**CFGWL1**	-	-	-	-	VBGTM			
3FFEh	**CFGWH1**	-	-	-	FRCTM				
3FFFh	**CFGWH2**	PROT	XRSTE	LVRE		-	PWRSAV	MVCLOCK	BOOTV

SFR & CFGW DESCRIPTION

SFR Adr	SFR Name	Bit #	Bit Name	R/W	Rst	Description
80h	P0	7~0	P0	R/W	FFh	Port0 data, also controls the P0.n pin's pull-up function. If the P0.n SFR data is "1" and the corresponding P0OE.n=0 (input mode), the pull-up is enabled.
81h	SP	7~0	SP	R/W	07h	Stack Point
82h	DPL	7~0	DPL	R/W	00h	Data Point low byte
83h	DPH	7~0	DPH	R/W	00h	Data Point high byte
87h	PCON	7	SMOD	R/W	0	UART double baud rate control bit 0: Disable UART double baud rate 1: Enable UART double baud rate
		3	GF1	R/W	0	General purpose flag bit
		2	GF0	R/W	0	General purpose flag bit
		1	PD	R/W	0	Stop bit. If 1 Stop mode is entered.
		0	IDL	R/W	0	Idle bit. If 1, Idle mode is entered.
88h	TCON	7	TF1	R/W	0	Timer1 overflow flag Set by H/W when Timer/Counter 1 overflows. Cleared by H/W when CPU vectors into the interrupt service routine.
		6	TR1	R/W	0	Timer1 run control 0: Timer1 stops 1: Timer1 runs
		5	TF0	R/W	0	Timer0 overflow flag Set by H/W when Timer/Counter 0 overflows. Cleared by H/W when CPU vectors into the interrupt service routine.
		4	TR0	R/W	0	Timer0 run control 0: Timer0 stops 1: Timer0 runs
		3	IE1	R/W	0	External Interrupt 1 (INT1 pin) edge flag Set by H/W when an INT1 pin falling edge is detected, no matter the EX1 is 0 or 1. It is cleared automatically when the program performs the interrupt service routine.
		2	IT1	R/W	0	External Interrupt 1 control bit 0: Low level active (level triggered) for INT1 pin 1: Falling edge active (edge triggered) for INT1 pin
		1	IE0	R/W	0	External Interrupt 0 (INT0 pin) edge flag Set by H/W when an INT0 pin falling edge is detected, no matter the EX0 is 0 or 1. It is cleared automatically when the program performs the interrupt service routine.
		0	IT0	R/W	0	External Interrupt 0 control bit 0: Low level active (level triggered) for INT0 pin 1: Falling edge active (edge triggered) for INT0 pin
89h	TMOD	7	GATE1	R/W	0	Timer1 gating control bit 0: Timer1 enable when TR1 bit is set 1: Timer1 enable only while the INT1 pin is high and TR1 bit is set
		6	CT1N	R/W	0	Timer1 Counter/Timer select bit 0: Timer mode, Timer1 data increases at 2 System clock cycle rate 1: Counter mode, Timer1 data increases at T1 pin's negative edge

SFR Adr	SFR Name	Bit #	Bit Name	R/W	Rst	Description
89h	TMOD	5~4	TMOD1	R/W	00	Timer1 mode select 00: 8-bit timer/counter (TH1) and 5-bit prescaler (TL1) 01: 16-bit timer/counter 10: 8-bit auto-reload timer/counter (TL1). Reloaded from TH1 at overflow. 11: Timer1 stops
		3	GATE0	R/W	0	Timer0 gating control bit 0: Timer0 enable when TR0 bit is set 1: Timer0 enable only while the INT0 pin is high and TR0 bit is set
		2	CT0N	R/W	0	Timer0 Counter/Timer select bit 0: Timer mode, Timer0 data increases at 2 System clock cycle rate 1: Counter mode, Timer0 data increases at T0 pin's negative edge
		1~0	TMOD0	R/W	00	Timer0 mode select 00: 8-bit timer/counter (TH0) and 5-bit prescaler (TL0) 01: 16-bit timer/counter 10: 8-bit auto-reload timer/counter (TL0). Reloaded from TH0 at overflow. 11: TL0 is an 8-bit timer/counter. TH0 is an 8-bit timer/counter using Timer1's TR1 and TF1 bits.
8Ah	TL0	7~0	TL0	R/W	00h	Timer0 data low byte
8Bh	TL1	7~0	TL1	R/W	00h	Timer1 data low byte
8Ch	TH0	7~0	TH0	R/W	00h	Timer0 data high byte
8Dh	TH1	7~0	TH1	R/W	00h	Timer1 data high byte
90h	P1	7~0	P1	R/W	FFh	Port1 data
91h	P0OE	7~0	P0OE	R/W	00h	Port0 CMOS Push-Pull output enable control 0: Disable 1: Enable
93h	P2OE	7~0	P2OE	R/W	00h	Port2 CMOS Push-Pull output enable control 0: Disable 1: Enable
94h	OPTION	7	UART1W	R/W	0	One wire UART mode enable, both TXD/RXD use P3.1 pin 0: Disable one wire UART mode 1: Enable one wire UART mode
		6	MODE3V	R/W	0	3V mode selection control bit If this bit is set, the chip can be only operated in the condition of V_{CC}<3.6V, and LDO is turned off to save current
		5~4	WDTPSC	R/W	00	Watchdog Timer pre-scalar time select 00: 400ms WDT overflow rate 01: 200ms WDT overflow rate 10: 100ms WDT overflow rate 11: 50ms WDT overflow rate
		3~2	ADCKS	R/W	00	ADC clock rate select 00: F_{SYSCLK}/32 01: F_{SYSCLK}/16 10: F_{SYSCLK}/8 11: F_{SYSCLK}/4

SFR Adr	SFR Name	Bit #	Bit Name	R/W	Rst	Description
95h	INTFLG	7	LVDO	R		Low Voltage Detect flag Set by H/W when a low voltage occurs. The flag is valid when LVR is 1.9V or disabled. This flag is disabled in Stop mode or if MODE3V=1 and PWRSAV=1.
		4	ADIF	R/W	0	ADC interrupt flag Set by H/W at the end of conversion. S/W writes EFh to INTFLG or sets the ADSOC bit to clear this flag.
		2	IE2	R/W	0	External Interrupt 2 (INT2 pin) edge flag Set by H/W when a falling edge is detected on the INT2 pin state, no matter the EX2 is 0 or 1. It is cleared automatically when the program performs the interrupt service routine. S/W can write FBh to INTFLG to clear this bit.
		1	P1IF	R/W	0	Port1 pin change interrupt flag Set by H/W when a P1 pin state change is detected, and its interrupt enable bit is set (P1WKUP). P1IE does not affect this flag's setting. It is cleared automatically when the program performs the interrupt service routine. S/W can write FDh to INTFLG to clear this bit.
96h	P1WKUP	7~0	P1WKUP	R/W	00h	P1.7~P1.0 pin individual Wake up/Interrupt enable control 0: Disable 1: Enable
97h	SWCMD	7~0	SWRST	W		Write 56h to generate S/W Reset
		7~0	IAPALL	W		Write 65h to set IAPALL control flag; Write other value to clear IAPALL flag. It is recommended to clear it immediately after IAP access.
		0	IAPE	R	0	Flag indicates Flash memory sectors can be accessed by IAP or not. This bit combines with MVCLOCK to define the accessible IAP area.
98h	SCON	7	SM0	R/W	0	Serial port mode select bit 0,1 00: Mode0: 8 bit shift register, Baud Rate = F_{SYSCLK}/2 01: Mode1: 8 bit UART, Baud Rate is variable 10: Mode2: 9 bit UART, Baud Rate=F_{SYSCLK}/32 or /64 11: Mode3: 9 bit UART, Baud Rate is variable
		6	SM1	R/W	0	
		5	SM2	R/W	0	Serial port mode select bit 2 SM2 enables multiprocessor communication over a single serial line and modifies the above as follows. In Modes 2 & 3, if SM2 is set then the received interrupt will not be generated if the received ninth data bit is 0. In Mode 1, the received interrupt will not be generated unless a valid stop bit is received. In Mode 0, SM2 should be 0.
		4	REN	R/W	0	UART reception enable 0: Disable reception 1: Enable reception
		3	TB8	R/W	0	Transmit Bit 8, the ninth bit to be transmitted in Mode 2 and 3
		2	RB8	R/W	0	Receive Bit 8, contains the ninth bit that was received in Mode 2 and 3 or the stop bit in Mode 1 if SM2=0
		1	TI	R/W	0	Transmit interrupt flag Set by H/W at the end of the eighth bit in Mode 0, or at the beginning of the stop bit in other modes. Must be cleared by S/W.
		0	RI	R/W	0	Receive interrupt flag

SFR Adr	SFR Name	Bit #	Bit Name	R/W	Rst	Description
						Set by H/W at the end of the eighth bit in Mode 0, or at the sampling point of the stop bit in other modes. Must be cleared by S/W.
99h	SBUF	7~0	SBUF	R/W		UART transmit and receive data. Transmit data is written to this location and receive data is read from this location, but the paths are independent.
9Ah	PWM0PRD	7~0	PWM0PRD	R/W	FFh	PWM0 period
9Bh	PWM0DH	7~0	PWM0DH	R/W	80h	PWM0 duty high byte The PWM0 output signal is reset to a low level whenever the 8-bit base counter matches the 8-bit PWM0DH.
9Ch	PWM1PRD	7~0	PWM1PRD	R/W	FFh	PWM1 period
9Dh	PWM1ADH	7~0	PWM1ADH	R/W	80h	PWM1A duty high byte The PWM1A output signal is reset to a low level whenever the 8-bit base counter matches the 8-bit PWM1ADH.
9Eh	PWM1BDH	7~0	PWM1BDH	R/W	80h	PWM1B duty high byte The PWM1B output signal is reset to a low level whenever the 8-bit base counter matches the 8-bit PWM1BDH.
9Fh	PWM1CDH	7~0	PWM1CDH	R/W	80h	PWM1C duty high byte The PWM1C output signal is reset to a low level whenever the 8-bit base counter matches the 8-bit PWM1CDH.
A0h	P2	7~0	P2	R/W	FFh	Port2 data, also controls the P2.n pin’s pull-up function. If the P2.n SFR data is “1” and the corresponding P2OE.n=0 (input mode), the pull-up is enabled.
A1h	PWMCKS	7~6	PWM1CKS	R/W	10	PWM1 clock source 00: $F_{SYSCLK}/4$ 01: $F_{SYSCLK}/2$ 10: F_{SYSCLK} 11: FRCx2
		3~2	PWM0CKS	R/W	10	PWM0 clock source 00: $F_{SYSCLK}/4$ 01: $F_{SYSCLK}/2$ 10: F_{SYSCLK} 11: FRCx2
A2h	P1MODL	7~6	P1MOD3	R/W	00	P1.3 pin control
		5~4	P1MOD2	R/W	00	P1.2 pin control
		3~2	P1MOD1	R/W	00	P1.1 pin control
		1~0	P1MOD0	R/W	00	P1.0 pin control
A3h	P1MODH	7~6	P1MOD7	R/W	00	P1.7 pin control
		5~4	P1MOD6	R/W	00	P1.6 pin control
		3~2	P1MOD5	R/W	00	P1.5 pin control
		1~0	P1MOD4	R/W	00	P1.4 pin control
A4h	P3MODL	7~6	P3MOD3	R/W	00	P3.3 pin control
		5~4	P3MOD2	R/W	00	P3.2 pin control
		3~2	P3MOD1	R/W	00	P3.1 pin control
		1~0	P3MOD0	R/W	00	P3.0 pin control
A5h	P3MODH	7~6	P3MOD7	R/W	00	P3.7 pin control
		5~4	P3MOD6	R/W	00	P3.6 pin control
		3~2	P3MOD5	R/W	00	P3.5 pin control
		1~0	P3MOD4	R/W	00	P3.4 pin control

SFR Adr	SFR Name	Bit #	Bit Name	R/W	Rst	Description
A6h	PINMOD	7	P5OE0	R/W	0	P5.0 CMOS Push-Pull output enable control 0: Disable 1: Enable
		6	PWM0POE	R/W	0	PWM0P signal output enable 0: Disable PWM0P signal output to P1.4 1: Enable PWM0APsignal output to P1.4
		5	PWM0NOE	R/W	0	PWM0N signal output enable 0: Disable PWM0N signal output to P5.0 1: Enable PWM0N signal output to P5.0
		4	T2OE	R/W	0	Timer2 signal output enable 0: Disable Timer2 overflow divided by 2 output to P1.0 1: Enable Timer2 overflow divided by 2 output to P1.0
		3	T0OE	R/W	0	Timer0 signal output enable 0: Disable Timer0 overflow divided by 64 output to P3.4 1: Enable Timer0 overflow divided by 64 output to P3.4
		2	PWM1AOE	R/W	0	PWM1A signal output enable 0: Disable PWM1A signal output to P1.5 1: Enable PWM1A signal output to P1.5
		1	PWM1BOE	R/W	0	PWM1B signal output enable 0: Disable PWM1B signal output to P1.6 1: Enable PWM1B signal output to P1.6
		0	PWM1COE	R/W	0	PWM1C signal output enable 0: Disable PWM1C signal output to P1.7 1: Enable PWM1C signal output to P1.7
A7h	PWMCON	5~4	PWM0MODE	R/W	0	PWM0 differential output mode 00: Mode 0 01: Mode 1 10: Mode 2 11: Mode 3
		3~0	PWM0NOVT	R/W	0	PWM0 non-overlap time select; 0:0, 1:1 , 2:2 , 3:3 , 4:4 , 5:5 , 6:6 , 7:7 , 8:8 , 9:9 , 10:10 , 11:11 , 12:12 , 13:13 , 14:14 , 15:16 PWM0 clock
A8h	IE	7	EA	R/W	0	Global interrupt enable 0: Disable all interrupts 1: Each interrupt is enabled or disabled by its individual interrupt control bit
		5	ET2	R/W	0	Timer2 interrupt enable 0: Disable Timer2 interrupt 1: Enable Timer2 interrupt
		4	ES	R/W	0	Serial Port (UART) interrupt enable 0: Disable Serial Port (UART) interrupt 1: Enable Serial Port (UART) interrupt
		3	ET1	R/W	0	Timer1 interrupt enable 0: Disable Timer1 interrupt 1: Enable Timer1 interrupt
		2	EX1	R/W	0	INT1 pin Interrupt enable and Stop mode wake up enable 0: Disable INT1 pin Interrupt and Stop mode wake up 1: Enable INT1 pin Interrupt and Stop mode wake up, it can wake up CPU from Stop mode no matter EA is 0 or 1.
		1	ET0	R/W	0	Timer0 interrupt enable 0: Disable Timer0 interrupt 1: Enable Timer0 interrupt

SFR Adr	SFR Name	Bit #	Bit Name	R/W	Rst	Description
		0	EX0	R/W	0	INT0 pin Interrupt enable and Stop mode wake up enable 0: Disable INT0 pin Interrupt and Stop mode wake up 1: Enable INT0 pin Interrupt and Stop mode wake up, it can wake up CPU from Stop mode no matter EA is 0 or 1.
A9h	INTE1	5	U2IE	R/W	0	USB2 interrupt enable, reference to USB2IE(DBh) for more detail
		4	U1IE	R/W	0	USB1 interrupt enable, reference to USB1IE(DAh) for more detail
		3	ADIE	R/W	0	ADC/Touch Key (F5284/84C Only) interrupt enable 0: Disable ADC/Touch Key interrupt 1: Enable ADC/Touch Key interrupt
		2	EX2	R/W	0	INT2 pin Interrupt enable and Stop mode wake up enable 0: Disable INT2 pin Interrupt and Stop mode wake up 1: Enable INT2 pin Interrupt and Stop mode wake up, it can wake up CPU from Stop mode no matter EA is 0 or 1.
		1	P1IE	R/W	0	Port1 pin change interrupt enable 0: Disable Port1 pin change interrupt 1: Enable Port1 pin change interrupt
		0	SPAIE	R/W	0	SPIA interrupt enable 0: Disable SPIA interrupt 1: Enable SPIA interrupt
AAh	ADCDL	7~4	ADCDL	R		ADC data bit 3~0
ABh	ADCDH	7~0	ADCDH	R		ADC data bit 11~4
ACh	PWM0DL	7~6	PWM0DL	R/W	0	PWM0 duty 1~0
	PWM1ADL	5~4	PWM1ADL	R/W	0	PWM1A duty 1~0
	PWM1BDL	3~2	PWM1BDL	R/W	0	PWM1B duty 1~0
	PWM1CDL	1~0	PWM1CDL	R/W	0	PWM1C duty 1~0
AEh	CHSEL	7~4	ADCHS	R/W	1111	ADC channel select 0000: ADC0 (P1.0) 0001: ADC1 (P1.1) 0010: ADC2 (P1.4) 0011: ADC3 (P1.5) 0100: ADC4 (P1.6) 0101: ADC5 (P1.7) 0110: ADC6 (P4.0) 0111: ADC7 (P4.1) 1000: ADC8 (P4.2) 1001: ADC9 (P4.3) 1010: ADC10(P3.2) 1011: ADC11(P3.3) 1100: VBG (internal Bandgap voltage without drive) 1101: VBGO(internal Bandgap with voltage drive) 111x: VSS
B0h	P3	7~0	P3	R/W	FFh	Port3 data
B1h	SPBCTRL	7	SPBMOD	R/W	0	SPIB Mode 0: P2.6~4 in GPIO 1: P2.6~4 in SPIB I/F function

SFR Adr	SFR Name	Bit #	Bit Name	R/W	Rst	Description
		6	SPBEN	R/W	0	SPIB transfer enable 0: automatically clears when data transfer complete 1: start transfer data and check busy state
		5	SPBCPOL	R/W	0	SPIB clock polarity 0: BSCK is low in idle state 1: BSCK is high in idle state
		4	SPBCPHA	R/W	0	SPIB clock phase 0: Data sample on first edge of BSCK period 1: Data sample on second edge of BSCK period
		3	SPBDIR	R/W	0	SPIB transfer direction 0: transmit data to SPI device 1: receive data from SPI device
		2	SPBLSBF	R/W	0	SPIB transfer LSB first 0: MSB first 1: LSB first
		0	CLRADR	R/W	0	SPIB Clear XRAM address 0: not clear SPIB buffer address 1: clear SPIB buffer address to 0

B2h	SPBCRS	6~0	SPBCRS	R/W	0	SPIB Clock select;
						0: 12MHz; 1: 6MHz; 2: 4MHz; 3: 3MHz;
						4: 2.4MHz; 5: 2MHz; 6: 1.71MHz; 7: 1.5MHz;
						SPIB clock rate = FRC24M / (2 * (SPICRS[6:0] + 1))

B3h	DMALEN	6~0	DMALEN	R/W	0	DMA data transfer length
B4h	P4MODL	7~6	P4MOD3	R/W	00	P4.3 pin control
		5~4	P4MOD2	R/W	00	P4.2 pin control
		3~2	P4MOD1	R/W	00	P4.1 pin control
		1~0	P4MOD0	R/W	00	P4.0 pin control
B5h	P4MODH	7~6	P4MOD7	R/W	00	P4.7 pin control
		5~4	P4MOD6	R/W	00	P4.6 pin control
		3~2	P4MOD5	R/W	00	P4.5 pin control
		1~0	P4MOD4	R/W	00	P4.4 pin control
B6h	KBMASK	7~0	KBMASK			Mask KSI[7:0] interrupt (KBDI) function while the corresponding bit is “1”
B8h	IP	5	PT2	R/W	0	Timer2 interrupt priority low bit
		4	PS	R/W	0	Serial Port interrupt priority low bit
		3	PT1	R/W	0	Timer1 interrupt priority low bit
		2	PX1	R/W	0	INT1 interrupt priority low bit
		1	PT0	R/W	0	Timer0 interrupt priority low bit
		0	PX0	R/W	0	INT0 interrupt priority low bit
B9h	IPH	5	PT2H	R/W	0	Timer2 interrupt priority high bit
		4	PSH	R/W	0	Serial Port interrupt priority high bit
		3	PT1H	R/W	0	Timer1 interrupt priority high bit
		2	PX1H	R/W	0	INT1 interrupt priority high bit
		1	PT0H	R/W	0	Timer0 interrupt priority high bit
		0	PX0H	R/W	0	INT0 interrupt priority high bit

SFR Adr	SFR Name	Bit #	Bit Name	R/W	Rst	Description
BAh	IP1	5	PUSB2I	R/W	0	USB2 interrupt priority low bit
		4	PUSB1I	R/W	0	USB1 interrupt priority low bit
		3	PADI	R/W	0	ADC interrupt priority low bit
		2	PX2	R/W	0	INT2 interrupt priority low bit
		1	PP1	R/W	0	Port1 pin change interrupt priority low bit
		0	PSPAI	R/W	0	SPIA interrupt priority low bit
BBh	IP1H	5	PUSB1IH	R/W	0	USB2 interrupt priority high bit
		4	PUSB1IH	R/W	0	USB1 interrupt priority high bit
		3	PADIH	R/W	0	ADC interrupt priority high bit
		2	PX2H	R/W	0	INT2 interrupt priority high bit
		1	PP1H	R/W	0	Port1 interrupt priority high bit
		0	PSPIH	R/W	0	SPIA interrupt priority high bit
BCh	SPACON	7	SPAEN	R/W	0	SPIA enable 0: SPIA disable 1: SPIA enable
		6	SPAMSTR	R/W	0	SPIA Master mode enable 0: Slave mode 1: Master mode
		5	SPACPOL	R/W	0	SPIA clock polarity 0: SCK is low in idle state 1: SCK is high in idle state
		4	SPACPHA	R/W	0	SPIA clock phase 0: Data sample on first edge of SCK period 1: Data sample on second edge of SCK period
		3	SPASSDIS	R/W	0	SS pin disable 0: Enable SS pin 1: Disable SS pin
		2	SPALSBF	R/W	0	SPIA LSB first 0: MSB first 1: LSB first
		1~0	SPASPCR	R/W	00	SPIA clock rate 00: F_{SYSCLK}/2 01: F_{SYSCLK}/4 10: F_{SYSCLK}/8 11: F_{SYSCLK}/16
BDh	SPASTA	7	SPAIF	R/W	0	SPIA interrupt flag This is set by H/W at the end of a data transfer. Cleared by H/W when an interrupt is vectored into. Writing 0 to this bit will clear this flag.
		6	SPAWCOL	R/W	0	Write collision interrupt flag Set by H/W if write data to SPADAT when SPABSY is set. Write 0 to this bit or rewrite data to SPDAT when SPABSY is cleared will clear this flag.
		5	SPAMODF	R/W	0	Mode fault interrupt flag Set by H/W when SPASSDIS is cleared and SS pin is pulled low in Master mode. Write 0 to this bit will clear this flag. When this bit is set, the SPAEN and SPAMSTR in SPACON will be cleared by H/W.
		4	SPARCVOVF	R/W	0	Received buffer overrun flag Set by H/W at the end of a data transfer and SPARCVBF is set. Write 0 to this bit or read SPADAT register will

SFR Adr	SFR Name	Bit #	Bit Name	R/W	Rst	Description
						clear this flag.
BDh	SPASTA	3	SPARCVBF	R/W	0	Receive buffer full flag Set by H/W at the end of a data transfer. Write 0 to this bit or read SPADAT register will clear this flag.
		2	SPABSY	R		SPIA busy flag Set by H/W when a SPIA transfer is in progress.
BEh	SPADAT	7~0	SPADAT	R/W	0	SPIA transmit and receive data The SPADAT register is used to transmit and receive data. Writing data to SPADAT place the data into shift register and start a transfer when in master mode. Reading SPADAT returns the contents of the receive buffer.
C0h	P5	0	P5.0	R/W	1	P5.0 data, also controls the P5.n pin's pull-up function. If the P5.n SFR data is "1" and the corresponding P5OE0=0 (input mode), the pull-up is enabled.
C1h	USBADR	7	USBE	R/W	0	USB function enable bit 0: disable P1.2/P1.3 as USB DP/DM pin 1: enable P1.2/P1.3 as USB DP/DM pin
		6~0	FUNADR	R/W	0	USB Function Address
C2h	USB1I	7	SET0I	R/W	0	Endpoint 0 SET0 Receive Interrupt flag, write 0 to clear flag.
		6	OUT0I	R/W	0	Endpoint 0 OUT Receive Interrupt flag, write 0 to clear flag.
		5	TX0I	R/W	0	Endpoint 0 Transmit Interrupt flag, write 0 to clear flag.
		4	TX1I	R/W	0	Endpoint 1 Transmit Interrupt flag, write 0 to clear flag.
		3	TX2I	R/W	0	Endpoint 2 Transmit Interrupt flag, write 0 to clear flag.
		2	SUSPI	R/W	0	USB Suspend Interrupt flag, write 0 to clear flag.
		1	TX3I	R/W	0	Endpoint 3 Bulk Transmit Interrupt flag, write 0 to clear flag.
		0	RC4I	R/W	0	Endpoint 4 Bulk Receive Interrupt flag, write 0 to clear flag.
C3h	USB2I	7	VDD5VRI	R/W	0	VDD5V Rise Interrupt flag, write 0 to clear it
		6	RSTI	R/W	0	USB Bus Reset Interrupt flag, write 0 to clear flag.
		5	RSMI	R/W	0	USB Resume Interrupt flag, write 0 to clear flag.
		4	KBDI	R/W	0	Keyboard Interrupt flag, write 0 to clear flag.
		3	PD_TXI	R/W	0	USB PD Transmit Interrupt flag, write 0 to clear flag.
		2	PD_RCI	R/W	0	USB PD Receive Interrupt flag, write 0 to clear flag.
C4h	USBCTRL	7	SUSPND	R/W	0	F/W set high to force USB interface into suspend mode
		6	RSMO	R/W	0	F/W set high to force USB interface send RESUME signal in suspend mode
		5	EP1CFG	R/W	0	Write 1to set Endpoint 1 configed
		4	EP2CFG	R/W	0	Write 1to set Endpoint 2 configed
		3	DEVICE_R	R/W	0	USB DP pull-up resistor enable bit 0: disable pull-up resistor 1: enable pull-up resistor
		2	OUT0RDY	R/W	0	Endpoint 0 ready for receive control bit 0: Endpoint 0 is not ready for receive OUT packet. 1: Endpoint 0 is ready for receive from USB Host; cleared by H/W while OUT0I occurs
C5h	TX0CTRL	7	TX0RDY	R/W	0	Endpoint 0 ready for transmit control bit 0: Endpoint 0 is not ready for IN packet.

SFR Adr	SFR Name	Bit #	Bit Name	R/W	Rst	Description
						1: Endpoint 0 is ready for transmit to USB Host; cleared by H/W while TX0I occurs
		6	TX0TGL	R/W	0	Endpoint 0 transmit data type 0: DATA0 1: DATA1.
		5	EP0STALL	R/W	0	0: Endpoint 0 will not stall OUT/IN packet 1: Endpoint 0 will stall OUT/IN packet
		4	IN0STALL	R/W	0	0: Endpoint 0 will not stall IN packet 1: Endpoint 0 will stall IN packet
		3~0	TX0CNT	R/W	0	Endpoint 0 transmit byte count F/W set the transmit byte count for IN0 packet
C6h	TX1CTRL	7	TX1RDY	R/W	0	Endpoint 1 ready for transmit control bit 0: Endpoint 1 is not ready for IN packet. 1: Endpoint 1 is ready for transmit to USB Host; cleared by H/W while TX1I occurs
		6	TX1TGL	R/W	0	Endpoint 1 transmit data type 0: DATA0. 1: DATA1.
		5	EP1STALL	R/W	0	0: Endpoint 1 will not stall IN packet 1: Endpoint 1 will stall IN packet
		3~0	TX1CNT	R/W	0	Endpoint 1 transmit byte count F/W set the transmit byte count for IN1 packet
C7h	TX2CTRL	7	TX2RDY	R/W	0	Endpoint 2 ready for transmit control bit 0: Endpoint 2 is not ready for IN packet. 1: Endpoint 2 is ready for transmit to USB Host; cleared by H/W while TX2I occurs
		6	TX2TGL	R/W	0	Endpoint 2 transmit data type 0: DATA0. 1: DATA1.
		5	EP2STALL	R/W	0	0: Endpoint 2 will not stall IN packet 1: Endpoint 2 will stall IN packet
		3~0	TX2CNT	R/W	0	Endpoint 2 transmit byte count F/W set the transmit byte count for IN2 packet
C8h	T2CON	7	TF2	R/W	0	Timer2 overflow flag Set by H/W when Timer/Counter 2 overflows unless RCLK=1 or TCLK=1. This bit must be cleared by S/W.
		6	EXF2	R/W	0	T2EX interrupt pin falling edge flag Set when a capture or a reload is caused by a negative transition on T2EX pin if EXEN2=1. This bit must be cleared by S/W.
		5	RCLK	R/W	0	UART receive clock control bit 0: Use Timer1 overflow as receive clock for serial port in mode 1 or 3 1: Use Timer2 overflow as receive clock for serial port in mode 1 or 3
		4	TCLK	R/W	0	UART transmit clock control bit 0: Use Timer1 overflow as transmit clock for serial port in mode 1 or 3 1: Use Timer2 overflow as transmit clock for serial port in mode 1 or 3
		3	EXEN2	R/W	0	T2EX pin enable 0: T2EX pin disable 1: T2EX pin enable, it cause a capture or reload when a

SFR Adr	SFR Name	Bit #	Bit Name	R/W	Rst	Description
						negative transition on T2EX pin is detected if RCLK=TCLK=0
		2	TR2	R/W	0	Timer2 run control 0: Timer2 stops 1: Timer2 runs
		1	CT2N	R/W	0	Timer2 Counter/Timer select bit 0: Timer mode, Timer2 data increases at 2 System clock cycle rate 1: Counter mode, Timer2 data increases at T2 pin’s negative edge
		0	CPRL2N	R/W	0	Timer2 Capture/Reload control bit 0: Reload mode, auto-reload on Timer2 overflows or negative transitions on T2EX pin if EXEN2=1 1: Capture mode, capture on negative transitions on T2EX pin if EXEN2=1 If RCLK=1 or TCLK=1, CPRL2N is ignored and timer is forced to auto-reload on Timer2 overflow
C9h	IAPWE	7	IAPWE	R/W	0	(W): Write 47h to set IAPWE control flag; Write other value to clear IAPWE flag (R) : IAPWE flag
		6	IAPTO	R		IAP TimeOut flag
CAh	RCP2L	7~0	RCP2L	R/W	00h	Timer2 reload/capture data low byte
CBh	RCP2H	7~0	RCP2H	R/W	00h	Timer2 reload/capture data high byte
CCh	TL2	7~0	TL2	R/W	00h	Timer2 data low byte
CDh	TH2	7~0	TH2	R/W	00h	Timer2 data high byte
CEh	TX3CTRL	7	TX3RDY	R/W	0	Endpoint 3 ready for transmit control bit 0: Endpoint 3 is not ready for IN packet. 1: Endpoint 3 is ready for transmit to USB Host; cleared by H/W while TX3I occurs
		6	TX3TGL	R/W	0	Endpoint 3 transmit data type 0: DATA0 1: DATA1.
		5	EP3STALL	R/W	0	0: Endpoint 3 will not stall IN packet 1: Endpoint 3 will stall IN packet
		4	EP3CFG	R/W	0	Write 1to set Endpoint 3 configed
CFh	RC4CTRL	7	RC4RDY	R/W	0	Endpoint 4 ready for receive control bit 0: Endpoint 4 is not ready for OUT packet. 1: Endpoint 4 is ready for receive from USB Host; cleared by H/W while RC4I occurs
		6	EP4STALL	R/W	0	0: Endpoint 3 will not stall OUT packet 1: Endpoint 4 will stall OUT packet
		5	EP4CFG	R/W	0	Write 1to set Endpoint 4 configed
		4	RC4TGL	R		Endpoint 4 received data type 0: DATA0 1: DATA1.
		3	RC4ERR	R		Endpoint 4 received data error status flag 0: Endpoint 4 received data is correct 1: Endpoint 4 received data is error
D0h	PSW	7	CY	R/W	0	ALU carry flag
		6	AC	R/W	0	ALU auxiliary carry flag
		5	F0	R/W	0	General purpose user-definable flag

SFR Adr	SFR Name	Bit #	Bit Name	R/W	Rst	Description
		4	RS1	R/W	0	The contents of (RS1, RS0) enable the working register banks as: 00: Bank 0 (00h~07h) 01: Bank 1 (08h~0Fh) 10: Bank 2 (10h~17h) 11: Bank 3 (18h~1Fh)
		3	RS0	R/W	0	
		2	OV	R/W	0	ALU overflow flag
		1	F1	R/W	0	General purpose user-definable flag
		0	P	R/W	0	Parity flag. Set/cleared by hardware each instruction cycle to indicate odd/even number of “one” bits in the accumulator.
D1h	TX3CNT	6~0	TX3CNT	R/W	0	Endpoint 3 transmit byte count F/W set the transmit byte count for IN3 packet
D2h	RC4CNT	6~0	RC4CNT	R		Endpoint 4 received byte count F/W read the received byte count from OUT4 packet
D3h	RC0REG	7	RC0TGL	R		Endpoint 0 received data type 0: DATA0 1: DATA1.
		6	RC0ERR	R		Endpoint 0 received data error status flag 0: Endpoint 0 received data is correct 1: Endpoint 0 received data is error
		5	EP0DIR	R		Endpoint 0 transfer direction 0: OUT/SETUP transfer 1: IN transfer
		4	EP0SET	R		Endpoint 0 SETUP Token indicator 0: not SETUP Token 1: SETUP Token
		3~0	OUT0CNT	R		Endpoint 0 received OUT packet byte count F/W read the received byte count from OUT0 packet
D4h	I80CTRL	3	I80BUSY	R		I80 Interface state of data transfer 0: I80 interface is idle 1: I80 interface is busy
		2	I80EN	R/W	0	Enable I80 I/F DMA mode 0: disable I80 DMA mode 1: enable I80 DMA mode
		1	I80START	R/W	0	I80 I/F data transfer start bit 0: write 0 to clear it if data transfer completes 1: write 1 to set it to start data transfer
		0	I80DIR	R/W	0	I80 I/F data transfer direction 0: write data to device 1: read data from device
D6h	XRAMCTRL	7	XRAM4UPD	R/W	0	XRAM4 access control bit 0: XRAM4 is access by MCU 1: XRAM4 is access by USB PD Receive message function
		6	XRAM3UPD	R/W	0	XRAM3 access control bit 0: XRAM3 is access by MCU 1: XRAM3 is access by USB PD Transmit message function
		5	XRAM1USB	R/W	0	XRAM1 access control bit 1: assign XRAM1 as USB Bulk transfer buffer.

SFR Adr	SFR Name	Bit #	Bit Name	R/W	Rst	Description
		4	XRAM2USB	R/W	0	XRAM2 access control bit 1: assign XRAM2 as USB Bulk transfer buffer.
		3	XRAM1SPB	R/W	0	XRAM1 access control bit 1: assign XRAM1 as SPIB DMA transfer buffer.
		2	XRAM2SPB	R/W	0	XRAM2 access control bit 1: assign XRAM2 as SPIB DMA transfer buffer
		1	XRAM1I80	R/W	0	XRAM1 access control bit 1: assign XRAM1 as I80 DMA transfer buffer
		0	XRAM2I80	R/W	0	XRAM2 access control bit 1: assign XRAM2 as I80 DMA transfer buffer
D8h	CLKCON	5	STPSCK	R/W	0	Set 1 to stop SRC
		4	STPPCK	R/W	0	Set 1 to stop UART/Timer0/Timer1/Timer2/ADC clock in Idle mode
		3	STPFCK	R/W	0	Set 1 to stop Fast clock for power saving in Slow / Idle mode. This bit can be changed only in Slow mode.
		2	SELFCK	R/W	0	System clock source selection. This bit can be changed only when STPFCK=0. 0: Slow clock 1: Fast clock
		1~0	CLKPSC	R/W	11	System clock prescaler. 00: System clock is Fast/Slow clock divided by 16 01: System clock is Fast/Slow clock divided by 4 10: System clock is Fast/Slow clock divided by 2 11: System clock is Fast/Slow clock divided by 1
DAh	USB1IE	7	SET0IE	R/W	0	Endpoint 0 SET0 Receive Interrupt enable bit 0: disable 1: enable interrupt service
		6	OUT0IE	R/W	0	Endpoint 0 OUT0 Receive Interrupt enable bit 0: disable 1: enable interrupt service
		5	TX0IE	R/W	0	Endpoint 0 Transmit Interrupt enable bit 0: disable 1: enable interrupt service
		4	TX1IE	R/W	0	Endpoint 1 Transmit Interrupt enable bit 0: disable 1: enable interrupt service
		3	TX2IE	R/W	0	Endpoint 2 Transmit Interrupt enable bit 0: disable 1: enable interrupt service
		2	SUSPIE	R/W	0	USB Suspend Interrupt enable bit 0: disable 1: enable interrupt service
		1	TX3IE	R/W	0	Endpoint 3 Bulk Transmit Interrupt enable bit 0: disable 1: enable interrupt service
		0	RC4IE	R/W	0	Endpoint 4 Bulk receive Interrupt enable bit 0: disable 1: enable interrupt service
DBh	USB2IE	7	VDD5CRIE	R/W	0	USB Bus power VDD5V Rise Interrupt enable bit 0: disable 1: enable interrupt service
		6	RSTIE	R/W	0	USB Bus Reset Interrupt enable bit

SFR Adr	SFR Name	Bit #	Bit Name	R/W	Rst	Description
						0: disable 1: enable interrupt service
		5	RSMIE	R/W	0	USB Resume Interrupt enable bit 0: disable 1: enable interrupt service
		4	KBDIE	R/W	0	Keyboard Scan Interrupt enable bit 0: disable 1: enable interrupt service
		3	PD_TXIE	R/W	0	USB PD Transmit Interrupt enable bit 0: disable 1: enable interrupt service
		2	PD_RCIE	R/W	0	USB PD Receive Interrupt enable bit 0: disable 1: enable interrupt service
DCh	USBRSTM	2	UTMEN	R/W	0	USB Reset Timeout enable 0: disable 1: enable
		1~0	UTMS	R/W	0	USB Reset Timeout time select 00: 2ms; 01: 3ms; 10: 4ms; 11: 5ms
DFh	PDCTRL	7	USBPDEN	R/W	0	USB PD mode enable bit 0: disable 1: enable USB PD mode
		6	PDTXRDY	R/W	0	SB PD ready for transmit control bit 0: not ready 1: USB PD is ready for transmit PD message
		5	PDRXRDY	R/W	0	SB PD ready to receive control bit 0: not ready 1: USB PD is ready to receive PD message
		4	SCC	R/W	0	Select CC1 or CC2 as CC pin 0: select CC1 as CC pin 1: select CC2 as CC pin
		3	SCCO2	R/W	0	Resistor connect to CC2 pin control bit 0: disconnect the resistor to CC1 pin 1: connect the resistor to CC1 pin
		2	SCCO1	R/W	0	Resistor connect to CC1 pin control bit 0: disconnect the resistor to CC1 pin 1: connect the resistor to CC1 pin
		1~0	CCS	R/W	3h	CC pin connect resistor select 00: 5.1K ohm to Ground 01: 10K ohm to VDD5 10: 22K ohm to VDD5 11: 56K ohm to VDD5
E0h	ACC	7~0	ACC	R/W	00h	Accumulator
E8h	P4	7~0	P4	R/W	FFh	Port4 data
EF	ANAT	2	ANTOE	R/W	0	Enable Analog Test pin to P4.6 0: disable 1: enable
		1~0	SANTO	R/W	0	Select Analog Test pin source 00: VBG 01: VBGO 10: VPD 11: ADOPO

SFR Adr	SFR Name	Bit #	Bit Name	R/W	Rst	Description
F0h	B	7~0	B	R/W	00h	B register
F5h	AUX2	7	VDD5VFLG	R		PC5V status 0: PC5V Low 1: PC5V High (USB plug in)
		6	VDD5FALL	R/W	0	USB Plug out flag, write 0 or power on reset to clear flag
		2~1	IAPTE	R/W	0	IAP Timeout
		0	VCCFLT	R/W	0	VCC line filter enable
F6h	VBGTM	3~0	VBGTM	R/W		Band-gap Voltage adjustment value, load from configuration Word after power on reset
F7h	FRCTM	4~0	FRCTM	R/W		Fast RC adjustment value, load from configuration Word after power on reset
F8h	AUX1	7	CLRWDT	R/W	0	Set to clear WDT, H/W auto clear it at next clock cycle
		6~5	WDTE	R/W		Watchdog Timer Reset control 0x: Watchdog Timer Reset disable 10: Watchdog Timer Reset enable in Fast/Slow mode, disable in Idle/Stop mode 11: Watchdog Timer Reset always enable
		4	ADSOC	R/W	0	Start ADC conversion Set the ADSOC bit to start ADC conversion, and the ADSOC bit will be cleared by H/W at the end of conversion. S/W can also write 0 to clear this flag.
		3	CLRPWM0	R/W	0	PWM0 clear enable 0: PWM0 is running 1: PWM0 is cleared and held
		2	PDADOP	R/W	0	ADC input bypass ADCOP 0: ADC input through ADCOP 1: ADC input bypass ADCOP
		1	BOOTV	R/W	0	BOOT vector select; load from Configuration Word 0: Boot from ROM address 0000h 1: BootT from ROM address 3000h
		0	DPSEL	R/W	0	Active DPTR Select

Flash Adr	Bit #	Name	Description
3FFCh	3~0	VBGTM	Band-gap Voltage adjustment
3FFEh	4~0	FRCTM	Fast RC frequency adjustment
3FFFh	7	PROT	Flash Memory Code Protect 0: Disable protect 1: Enable protect
	6	XRSTE	External Pin Reset control 0: Disable External Pin Reset 1: Enable External Pin Reset
	5~4	LVRE	Low Voltage Reset function select 00: Set LVR at 2.9V 01: Set LVR at 2.3V 10: reserved 11: Set LVR at 2.1V
	2	PWRSAV	Power save function control bit 0: Disable Power save function 1: Enable Power save function

	1	MVCLOCK	If 1, the MOVC & MOVX instruction’s accessibility to MOVC-Lock area is limited.
	0	BOOTV	BOOT vector select 0: Boot from ROM address 0000h 1: Boot from ROM address 3000h

INSTRUCTION SET

Instructions are 1, 2 or 3 Bytes long as listed in the ‘byte’ column below. Each instruction takes 2~8 System clock cycles to execute as listed in the ‘cycle’ column below.

ARITHMETIC

Mnemonic	Description	byte	cycle	opcode
ADD A, Rn	Add register to A	1	2	28-2F
ADD A, dir	Add direct byte to A	2	2	25
ADD A, @Ri	Add indirect memory to A	1	2	26-27
ADD A, #data	Add immediate to A	2	2	24
ADDC A, Rn	Add register to A with carry	1	2	38-3F
ADDC A, dir	Add direct byte to A with carry	2	2	35
ADDC A, @Ri	Add indirect memory to A with carry	1	2	36-37
ADDC A, #data	Add immediate to A with carry	2	2	34
SUBB A, Rn	Subtract register from A with borrow	1	2	98-9F
SUBB A, dir	Subtract direct byte from A with borrow	2	2	95
SUBB A, @Ri	Subtract indirect memory from A with borrow	1	2	96-97
SUBB A, #data	Subtract immediate from A with borrow	2	2	94
INC A	Increment A	1	2	04
INC Rn	Increment register	1	2	08-0F
INC dir	Increment direct byte	2	2	05
INC @Ri	Increment indirect memory	1	2	06-07
DEC A	Decrement A	1	2	14
DEC Rn	Decrement register	1	2	18-1F
DEC dir	Decrement direct byte	2	2	15
DEC @Ri	Decrement indirect memory	1	2	16-17
INC DPTR	Increment data pointer	1	4	A3
MUL AB	Multiply A by B	1	8	A4
DIV AB	Divide A by B	1	8	84
DA A	Decimal Adjust A	1	2	D4

LOGICAL

Mnemonic	Description	byte	cycle	opcode
ANL A, Rn	AND register to A	1	2	58-5F
ANL A, dir	AND direct byte to A	2	2	55
ANL A, @Ri	AND indirect memory to A	1	2	56-57
ANL A, #data	AND immediate to A	2	2	54
ANL dir, A	AND A to direct byte	2	2	52
ANL dir, #data	AND immediate to direct byte	3	4	53
ORL A, Rn	OR register to A	1	2	48-4F
ORL A, dir	OR direct byte to A	2	2	45
ORL A, @Ri	OR indirect memory to A	1	2	46-47
ORL A, #data	OR immediate to A	2	2	44
ORL dir, A	OR A to direct byte	2	2	42
ORL dir, #data	OR immediate to direct byte	3	4	43
XRL A, Rn	Exclusive-OR register to A	1	2	68-6F
XRL A, dir	Exclusive-OR direct byte to A	2	2	65
XRL A, @Ri	Exclusive-OR indirect memory to A	1	2	66-67

LOGICAL				
Mnemonic	**Description**	**byte**	**cycle**	**opcode**
XRL A, #data	Exclusive-OR immediate to A	2	2	64
XRL dir, A	Exclusive-OR A to direct byte	2	2	62
XRL dir, #data	Exclusive-OR immediate to direct byte	3	4	63
CLR A	Clear A	1	2	E4
CPL A	Complement A	1	2	F4
SWAP A	Swap Nibbles of A	1	2	C4
RL A	Rotate A left	1	2	23
RLC A	Rotate A left through carry	1	2	33
RR A	Rotate A right	1	2	03
RRC A	Rotate A right through carry	1	2	13

DATA TRANSFER				
Mnemonic	**Description**	**byte**	**cycle**	**opcode**
MOV A, Rn	Move register to A	1	2	E8-EF
MOV A, dir	Move direct byte to A	2	2	E5
MOV A, @Ri	Move indirect memory to A	1	2	E6-E7
MOV A, #data	Move immediate to A	2	2	74
MOV Rn, A	Move A to register	1	2	F8-FF
MOV Rn, dir	Move direct byte to register	2	4	A8-AF
MOV Rn, #data	Move immediate to register	2	2	78-7F
MOV dir, A	Move A to direct byte	2	2	F5
MOV dir, Rn	Move register to direct byte	2	4	88-8F
MOV dir, dir	Move direct byte to direct byte	3	4	85
MOV dir, @Ri	Move indirect memory to direct byte	2	4	86-87
MOV dir, #data	Move immediate to direct byte	3	4	75
MOV @Ri, A	Move A to indirect memory	1	2	F6-F7
MOV @Ri, dir	Move direct byte to indirect memory	2	4	A6-A7
MOV @Ri, #data	Move immediate to indirect memory	2	2	76-77
MOV DPTR, #data	Move immediate to data pointer	3	4	90
MOVC A, @A+DPTR	Move code byte relative DPTR to A	1	4	93
MOVC A, @A+PC	Move code byte relative PC to A	1	4	83
MOVX A, @Ri	Move external data (A8) to A	1	4	E2-E3
MOVX A, @DPTR	Move external data (A16) to A	1	4	E0
MOVX @Ri, A	Move A to external data (A8)	1	4	F2-F3
MOVX @DPTR, A	Move A to external data (A16)	1	4	F0
PUSH dir	Push direct byte onto stack	2	4	C0
POP dir	Pop direct byte from stack	2	4	D0
XCH A, Rn	Exchange A and register	1	2	C8-CF
XCH A, dir	Exchange A and direct byte	2	2	C5
XCH A, @Ri	Exchange A and indirect memory	1	2	C6-C7
XCHD A, @Ri	Exchange A and indirect memory nibble	1	2	D6-D7

BOOLEAN				
Mnemonic	**Description**	**byte**	**cycle**	**opcode**
CLR C	Clear carry	1	2	C3
CLR bit	Clear direct bit	2	2	C2
SETB C	Set carry	1	2	D3
SETB bit	Set direct bit	2	2	D2
CPL C	Complement carry	1	2	B3
CPL bit	Complement direct bit	2	2	B2
ANL C, bit	AND direct bit to carry	2	4	82

ANL C, /bit	AND direct bit inverse to carry	2	4	B0
ORL C, bit	OR direct bit to carry	2	4	72
ORL C, /bit	OR direct bit inverse to carry	2	4	A0
MOV C, bit	Move direct bit to carry	2	2	A2
MOV bit, C	Move carry to direct bit	2	4	92
BRANCHING				
Mnemonic	**Description**	**byte**	**cycle**	**opcode**
ACALL addr 11	Absolute jump to subroutine	2	4	11-F1
LCALL addr 16	Long jump to subroutine	3	4	12
RET	Return from subroutine	1	4	22
RETI	Return from interrupt	1	4	32
AJMP addr 11	Absolute jump unconditional	2	4	01-E1
LJMP addr 16	Long jump unconditional	3	4	02
SJMP rel	Short jump (relative address)	2	4	80
JC rel	Jump on carry=1	2	4	40
JNC rel	Jump on carry=0	2	4	50
JB bit, rel	Jump on direct bit=1	3	4	20
JNB bit, rel	Jump on direct bit=0	3	4	30
JBC bit, rel	Jump on direct bit=1 and clear	3	4	10
JMP @A+DPTR	Jump indirect relative DPTR	1	4	73
JZ rel	Jump on accumulator=0	2	4	60
JNZ rel	Jump on accumulator … 0	2	4	70
CJNE A, dir,rel	Compare A,direct, jump not equal relative	3	4	B5
CJNE A, #data, rel	Compare A,immediate, jump not equal relative	3	4	B4
CJNE Rn, #data, rel	Compare register,immediate, jump not equal relative	3	4	B8-BF
CJNE @Ri, #data, rel	Compare indirect,immediate, jump not equal relative	3	4	B6-B7
DJNZ Rn, rel	Decrement register, jump not zero relative	2	4	D8-DF
DJNZ dir, rel	Decrement direct byte, jump not zero relative	3	4	D5

MISCELLANEOUS				
Mnemonic	**Description**	**byte**	**cycle**	**opcode**
NOP	No operation	1	2	00

In the above table, an entry such as E8-EF indicates a continuous block of hex opcodes used for 8 different registers, the register numbers of which are defined by the lowest three bits of the corresponding code. Non-continuous blocks of codes, shown as 11-F1 (for example), are used for absolute jumps and calls with the top 3 bits of the code being used to store the top three bits of the destination address.

ELECTRICAL CHARACTERISTICS

1. Absolute Maximum Ratings (T_A=25°C)

Parameter	Rating	Unit
Supply voltage	V_{SS} –0.3 ~ V_{SS} +5.5	V
Input voltage	V_{SS} –0.3 ~ V_{CC} +0.3	
Output voltage	V_{SS} –0.3 ~ V_{CC} +0.3	
Output current high per 1 PIN	–25	mA
Output current high per all PIN	–80	
Output current low per 1 PIN	+30	
Output current low per all PIN	+150	
Maximum Operating Voltage	5.5	V
Operating temperature	–40 ~ +85	°C
Storage temperature	–65 ~ +150	

2. DC Characteristics (T_A=25°C, V_{CC}=2.0V ~ 5.5V)

Parameter	Symbol	Conditions		Min.	Typ.	Max.	Unit
Operating Voltage	V_{CC}	Fast mode, F_{SYSCLK}=11.0592 MHz		2.9	–	5.5	V
		Fast mode, F_{SYSCLK}=5.5296 MHz		1.9	–	5.5	
		Fast mode, F_{SYSCLK}=2.7648 MHz		1.4	–	5.5	
		Fast mode, F_{SYSCLK}=0.6912 MHz		1.3	–	5.5	
		Slow mode, SRC		1.3	–	5.5	
Input High Voltage	V_{IH}	All Input, except P1.2, P1.3	V_{CC}=5V	0.6V_{CC}	–	–	V
			V_{CC}=3V	0.6V_{CC}	–	–	
		P1.2, P1.3	V_{CC}=5V	0.8V_{CC}	–	–	
			V_{CC}=3V	0.8V_{CC}	–	–	
Input Low Voltage	V_{IL}	All Input	V_{CC}=5V	–	–	0.2V_{CC}	V
			V_{CC}=3V	–	–	0.2V_{CC}	
I/O Port Source Current	I_{OH}	All Output, except P4.4, P4.5	V_{CC}=5V V_{OH}=0.9V_{CC}	4	8	–	mA
			V_{CC}=3V V_{OH}=0.9V_{CC}	2	4		
		P4.4, P4.5	V_{CC}=5V V_{OH}=0.9V_{CC}	30	40	–	
			V_{CC}=3V V_{OH}=0.9V_{CC}	15	20		
I/O Port Sink Current	I_{OL}	All Output, except P4.4, P4.5	V_{CC}=5V V_{OL}=0.1V_{CC}	8	16	–	mA
			V_{CC}=3V V_{OL}=0.1V_{CC}	4	8	–	
		P4.4, P4.5	V_{CC}=5V V_{OL}=0.1V_{CC}	30	40		
			V_{CC}=3V V_{OL}=0.1V_{CC}	20	25		
Input Leakage Current (pin high)	I_{ILH}	All Input	V_{in}=V_{CC}	–	–	1	μA

Input Leakage Current (pin low)	I_{ILL}	All Input	V_{in}=0V	–	–	–1	
Supply Current	I_{CC}	Fast, V_{CC}=5V LVR enable MODE3V=0	FRC= 11.0592MHz	–	7.0	–	mA
			FRC= 5.5296MHz	–	5.5	–	
			FRC= 2.7648 MHz	–	4.5	–	
		Fast, V_{CC}=3V LVR enable MODE3V=0	FRC= 11.0592MHz	–	5.5	–	
			FRC= 5.5296MHz	–	4.5	–	
			FRC= 2.7648 MHz	–	3.4	–	
		Fast, V_{CC}=3V LVR enable MODE3V=1	FRC= 11.0592MHz	–	5.5	–	
			FRC= 5.5296MHz	–	4.5	–	
			FRC= 2.7648 MHz	–	3.4	–	
		Slow, V_{CC}=5V LVR enable MODE3V=0	SRC=32 KHz	–	2.5	–	mA
		Slow, V_{CC}=3V LVR enable MODE3V=0	SRC=32 KHz	–	1.9	–	
		Slow, V_{CC}=3V MODE3V=1 PWRSAV=1	SRC=32 KHz	–	1.3	–	
		Slow, V_{CC}=3V MODE3V=1 PWRSAV=0	SRC=32 KHz	–	1.8	–	
		Idle, V_{CC}=5V LVR enable MODE3V=0	SRC=32 KHz	–	820	–	uA
		Idle, V_{CC}=3V LVR enable MODE3V=0	SRC=32 KHz	–	430	–	
		Idle, V_{CC}=3V MODE3V=1 PWRSAV=1	SRC=32 KHz	–	60	–	
		Idle, V_{CC}=3V MODE3V=1 PWRSAV=0	SRC=32 KHz	–	350	–	
		Stop, V_{CC}=5V MODE3V=0	PWRSAV=1	–	10	–	
			PWRSAV=0	–	800	–	
		Stop, V_{CC}=3V MODE3V=0	PWRSAV=1	–	3	-	
			PWRSAV=0	–	400	–	
System Clock Frequency	F_{SYSCLK}	V_{CC}>LVR_{th}	V_{CC}=5V	–	11.0592	-	MHz
			V_{CC}=4V	–	11.0592	-	
			V_{CC}=3V	–	11.0592	-	
LVR Reference Voltage	V_{LVR}	T_A=25°C		–	2.9	–	V
				–	2.3	–	
				–	2.1	–	
LVR Hysteresis Voltage	V_{HYST}	T_A=25°C		–	±0.1	–	V
Pull-Up Resistor	R_P	V_{IN}=0V All except P1.2, P1.3	V_{CC}=5V	–	26	–	KΩ
			V_{CC}=3V		48		

		V_{IN}=0V P1.2, P1.3	V_{CC}=3V	–	120	–	

3. Clock Timing (T_A= –40˚C ~ +85˚C, V_{CC}=3.0V ~ 5.5V)

Parameter	Conditions	Min.	Typ.	Max.	Unit
FRC Frequency	25°C, V_{CC}=3.0 ~ 5.5V	-1.6%	11.0592	+1.6%	MHz
	0°C~ 70°C, V_{CC}=3.0 ~ 5.5V	-3.8%	11.0592	+3.0%	
	–40°C ~ 85°C, V_{CC}=3.0 ~ 5.5V	-8.0%	11.0592	+3.0%	

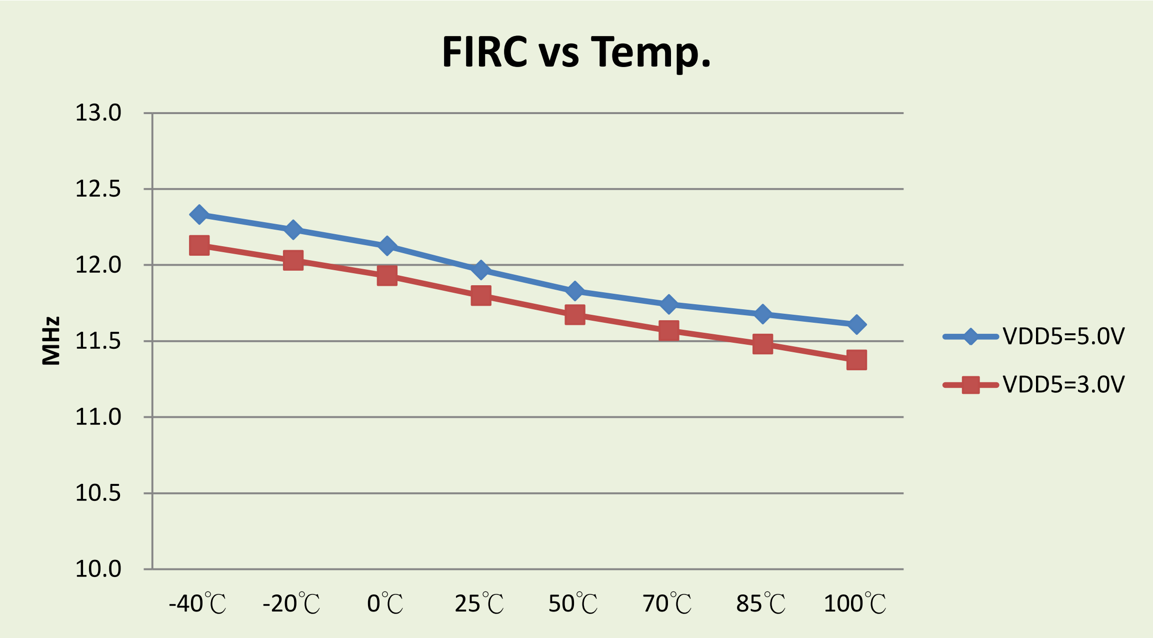
4. Reset Timing Characteristics (T_A= –40˚C ~ +85˚C, V_{CC}=3.0V ~ 5.0V)

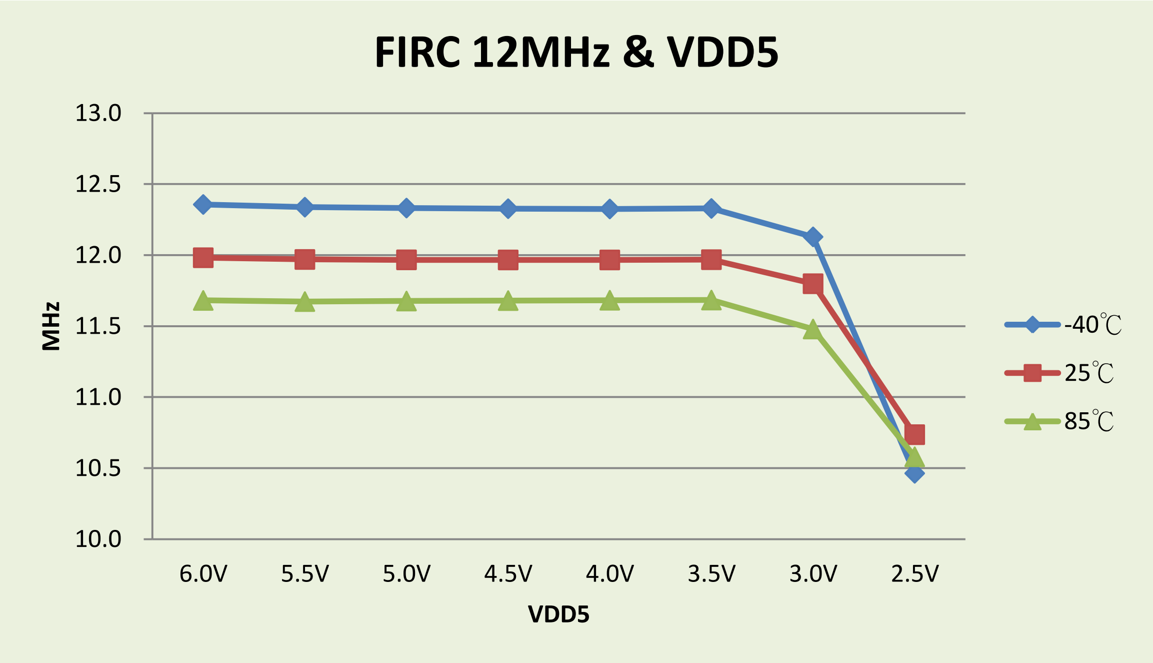
Parameter	Conditions	Min.	Typ.	Max.	Unit
RESET Input Low width	Input V_{CC}=5.0V ±10 %	30	–	–	μs
WDT wakeup time	V_{CC}=5.0V, WDTPSC=11	–	150	–	ms

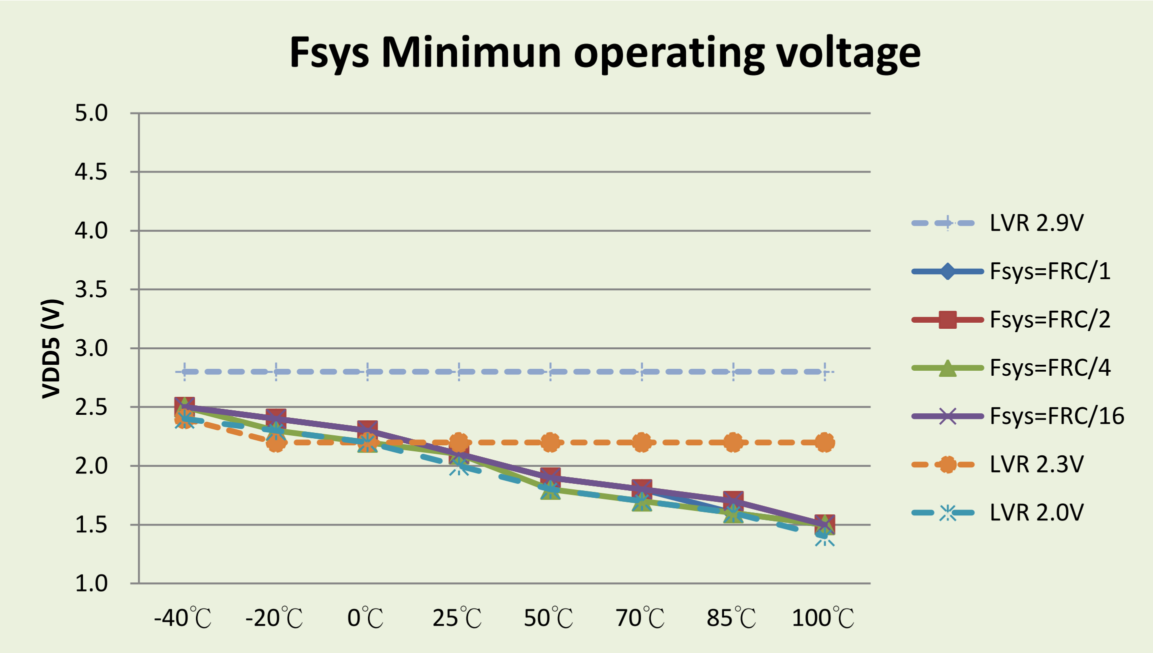
5. ADC Electrical Characteristics (T_A=25˚C, V_{CC}=3.0V ~ 5.5V, V_{SS}=0V)

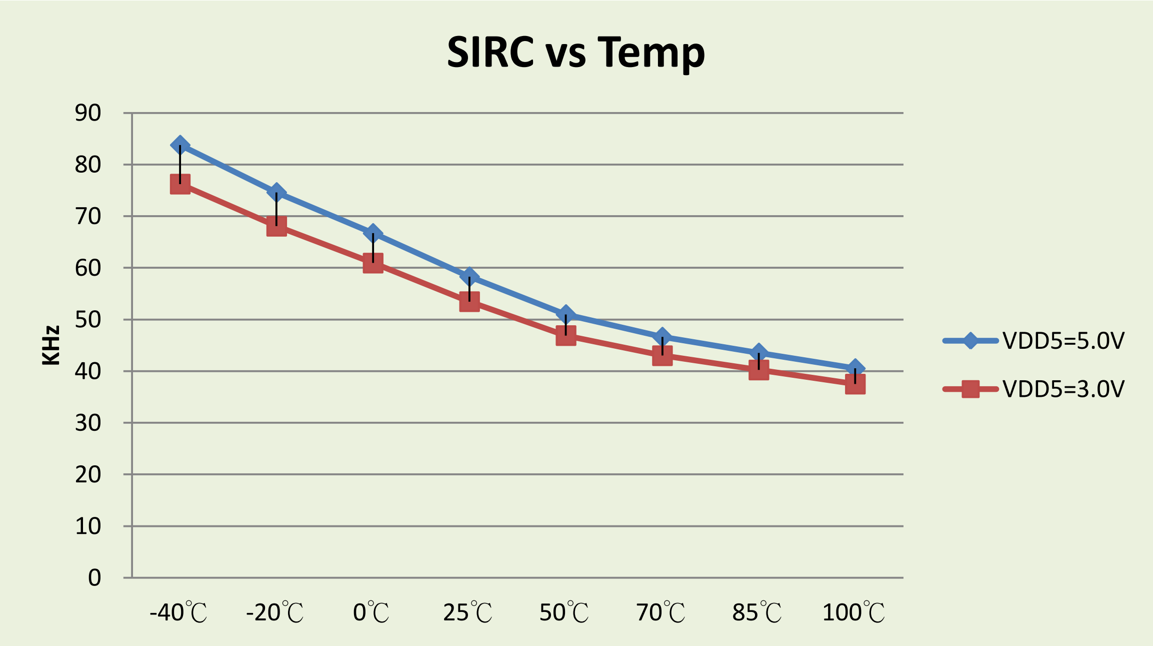
Parameter	Conditions	Min.	Typ.	Max.	Unit
Total Accuracy	V_{CC}=5V, V_{SS}=0V	–	±2.5	±4	LSB
Integral Non-Linearity		–	±3.2	±5	
Max Input Clock (f_{ADC})	–	–	–	1	MHz
Conversion Time	f_{ADC}=1MHz	–	50	–	μs
BandGap Voltage Reference	V_{CC}=3V	1.14	1.22	1.30	V
	V_{CC}=5V	1.15	1.25	1.35	
Input Voltage	–	V_{SS}	–	V_{CC}	V

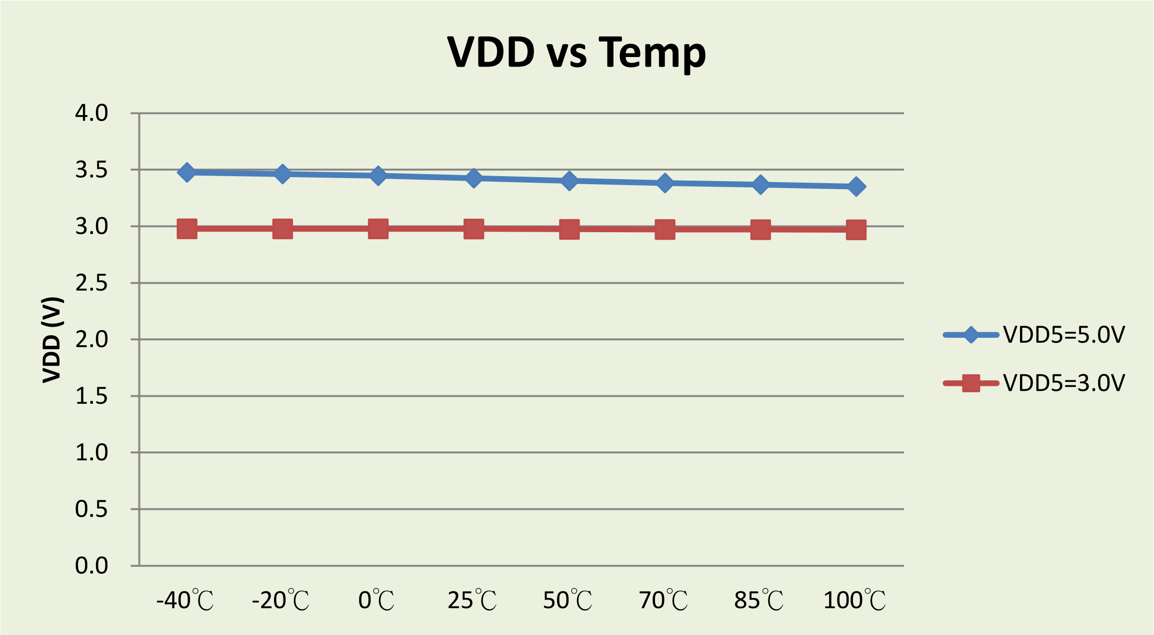
6. Characteristics Graphs






Fsys Minimun operating voltage
VDD5 (V)
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
-40℃
-20℃
0℃
25℃
50℃
70℃
85℃
100℃
LVR 2.9V
Fsys=FRC/1
Fsys=FRC/2
Fsys=FRC/4
Fsys=FRC/16
LVR 2.3V
LVR 2.0V


SIRC vs Temp
KHz
90
80
70
60
50
40
30
20
10
0
-40℃
-20℃
0℃
25℃
50℃
70℃
85℃
100℃
VDD5=5.0V
VDD5=3.0V


VDD vs Temp
VDD (V)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
-40℃
-20℃
0℃
25℃
50℃
70℃
85℃
100℃
VDD5=5.0V
VDD5=3.0V

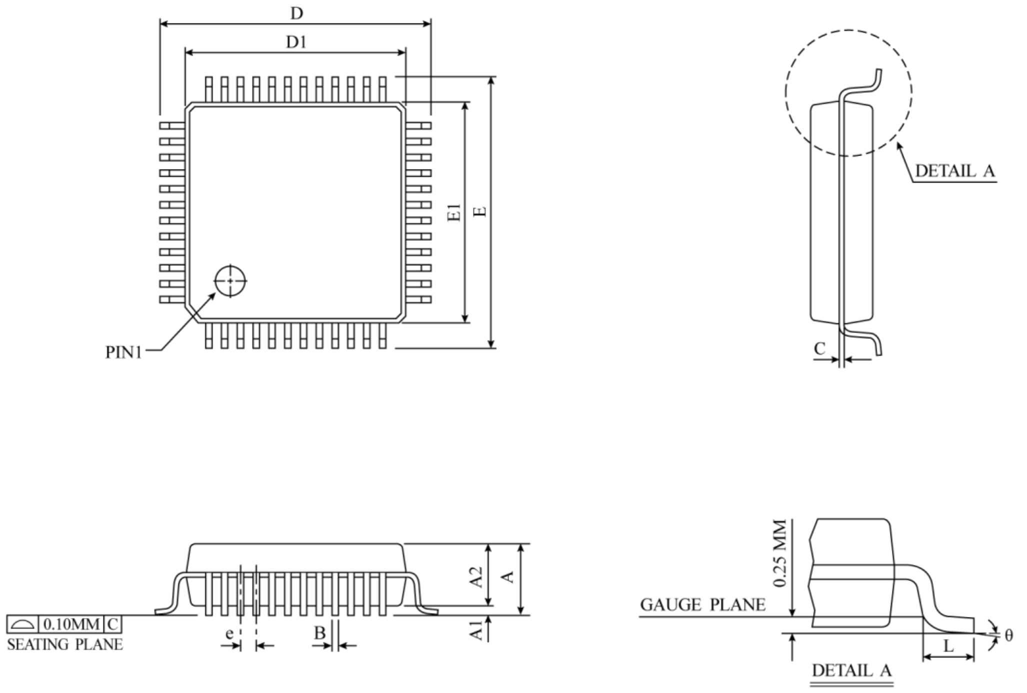
PACKAGE INFORMATION

Ordering Information

Ordering Number	Package
TM52F5284-MTP	Wafer/Dice blank chip
TM52F5284-COD	Wafer/Dice with code
TM52F5284-MTP-72	LQFP 48-pin (7x7 mm)

Package Information

LQFP 48-pin (7x7 mm) Package Dimensions



SYMBOL	DIMENSION IN MM			DIMENSION IN INCH		
	MIN	NOM	MAX	MIN	NOM	MAX
A	-	-	1.60	-	-	0.063
A1	0.05	0.10	0.15	0.001	0.004	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
B	0.17	0.22	0.27	0.007	0.009	0.011
C	0.09	0.15	0.20	0.004	0.006	0.008
D	9.00 BSC			0.354 BSC		
D1	7.00 BSC			0.276 BSC		
E	9.00 BSC			0.354 BSC		
E1	7.00 BSC			0.276 BSC		
e	0.50 BSC			0.020 BSC		
L	0.45	0.60	0.75	0.018	0.024	0.030
θ	0°	3.5°	7°	0°	3.5°	7°
JEDEC	MS-026 (BBC)					

△ * NOTES : DIMENSION 〝D1〞 AND 〝E1〞 DO NOT INCLUDE MOLD PROTRUSIONS. ALLOWABLE PROTRUSIONS IS 0.25 mm PER SIDE. 〝D1〞 AND 〝E1〞 ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS INCLUDING MOLD MISMACH.