HC89S003F4

Datasheet

20Pin 8bit
FLASH Microcontroller with ADC
Peripheral function Ports total mapping

Contents

1 Description

HC89S003F4 is an enhanced 8 bit microcontroller with high frequency and low power consumption CMOS process. 16K bytes flash program memory, 256 bytes IRAM and 256 bytes XRAM, 18 bi-directional I/O, five 16-bit Timer/counters, 3 groups 12 bits complementary PWM with dead-time control, one 8bits PWM, 2 UART, 1 SPI, 16 external interrupts, 8+2 channels 12 bits ADC, 4 system work modes (normal, low frequency, power-down, idle) and 16 interrupt sources.

1.1 Features

◆ **CPU**
  ● Enhanced 1T 8051 core
◆ **ROM**
  ● 16K bytes flash
  ● Support IAP and ICP operation
  ● Flexible code protection mode
◆ **RAM**
  ● 256 bytes IRAM
  ● 256 bytes XRAM
◆ **Clock**
  ● Internal high precision ($\pm1\%$) 32MHz RC
  ● Internal 44 KHz RC
  ● External high frequency oscillator 4MHz-24MHz
  ● External low frequency oscillator 32.768KHz
  ● Multiple clock output
◆ **RESET**
  ● Power on reset (POR)
  ● Multistep low voltage reset (BOR) 4.2/3.9/3.6/3.0/2.6/2.4/2.0/1.8V
  ● Watchdog Timer reset
  ● Software reset
  ● Stack overflow reset
  ● External pin low voltage reset
  ● External pin voltage (1.2V) detection reset
◆ **I/O**
  ● 18 bi-directional IO
  ● Multiple modes configurable: input, pull-up input, pull-down input, Schmitt input, analog input, strong push pull output, open drain output, open drain output with pull-up,
  ● Peripheral function Ports total mapping module

◆ **interrupt**
  ● 16 interrupt sources
  ● 4 level interrupt priorities

  ● 16 external interrupts
◆ **Timer/Counter**
  ● T0/T1 compatible with standard 8051, 16-bit auto reload
  ● T3 can run in power-down mode
  ● T4 can be trigged by external signal
  ● T5 with capture function
◆ **PWM**
  ● Up to 3 groups 12 bits complementary PWM with dead-time control
    - Configurable 6 channels independent output
    - Can used as Timer
    - Malfunction detection function
  ● 1 channel 8 bit PWM output
◆ **Communication interfaces**
  ● 2 UART
  ● 1 SPI
◆ **Analog to digital converter (ADC)**
  ● 12 bit ADC, up to 8+2 multiple channels
  ● ADC reference voltage: internal VREF, external VREF, and VDD
  ● Power saving wakeup function (single channel)
◆ **Low voltage detection module**
  ● Multilevel voltage detection with interrupt 4.2/3.9/3.6/3.0/2.6/2.4/2.0/1.9V
◆ **Cyclic redundancy check(CRC)**
◆ **Power saving mode**
  ● Idle mode
  ● power-down mode
◆ **Operating conditions**
  ● Wide operating voltage 2.0V to 5.5V
  ● Temperature range -40℃ to +85℃
◆ **Package**
  ● TSSOP20
  ● SOP/TSSOP/QFN20

✓ **Selection table**

Device	ROM	RAM	I/O	Timer	PWM	A/D	INT	PCA	IIC	UART	SPI	WDT	Voltage	TEMP	Package
HC89S003F4	16K	256+256	18	16bit*5	12bit*3 group +8bit*1 group	8+2	16	/	/	2	1	1	2.0~ 5.5V	-40~ +85℃	TSSOP20

HC89S003F4 use attentions：

1. In order to ensure the system stability, user must connect a capacitor (≥0.1μF) between VDD and GND.
2. When user use ADC module, no matter what the reference voltage is selected, the system voltage VDD must above 2.7V.
3. When external interrupt in query mode, interrupt flag cannot be cleared normally. User must disable interrupt enable bit first, then clear interrupt flag, after completion of clearing the interrupt flag, then enable external interrupt to generate normal external interrupt query. No problem in interrupt mode.
4. When user use T3 counter clock source to select external clock of port input, TR3 and T3CLKS [1:0] should be configured simultaneously (that is to say, user should uses one instruction to complete the configuration).

1.2 System diagram
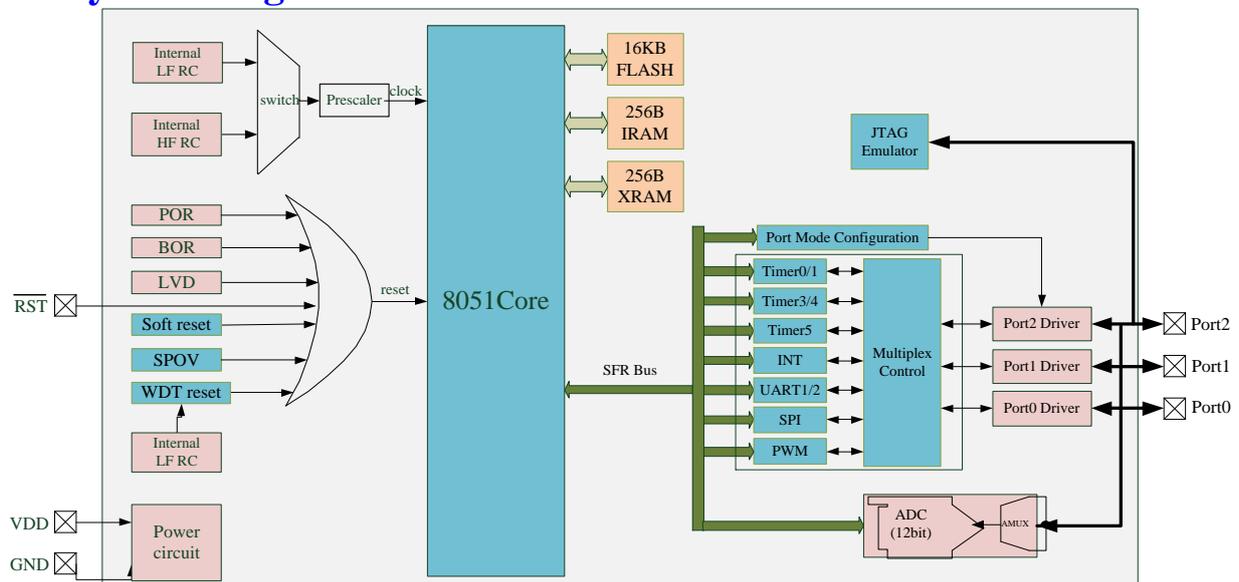
Figure 1-1 System diagram

TSSOP20
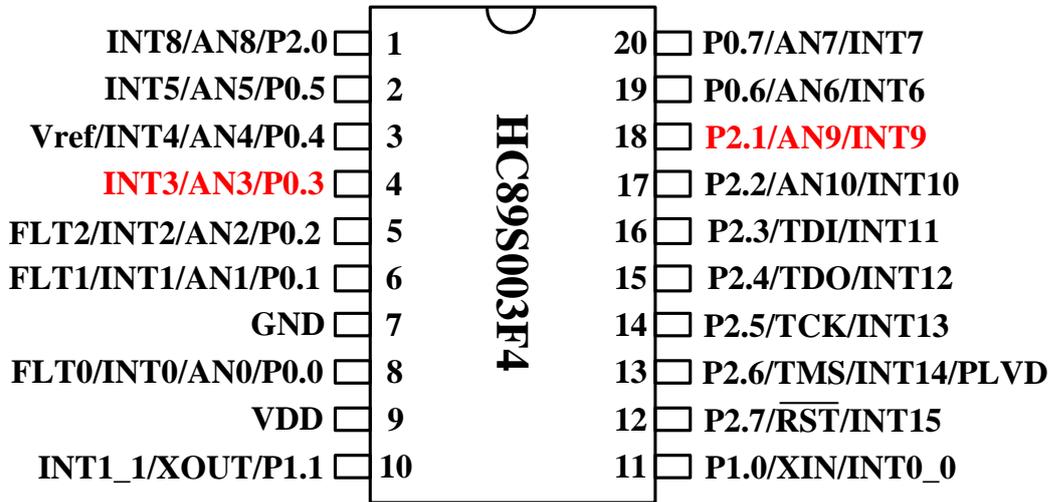
Figure 1-2 TSSOP20

QFN20
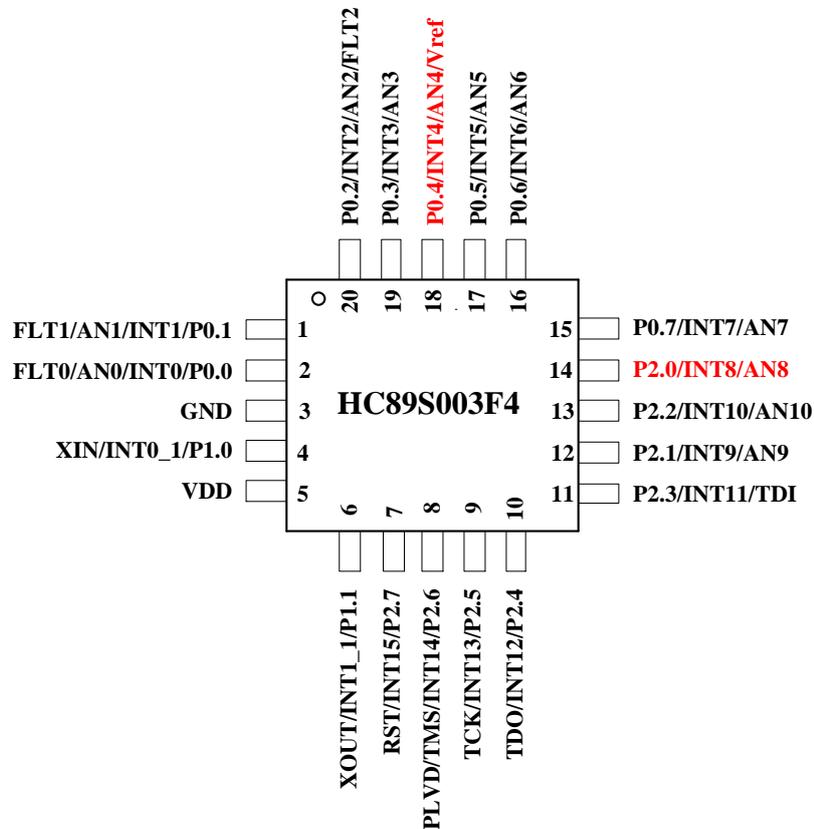
Figure 1-3 QFN20

1.3 Pin configuration

1.3.1 SOP20/TSSOP20 Pin configuration

```
INT8/AN8/P2.0  □  1            20  □  P0.7/AN7/INT7
INT5/AN5/P0.5  □  2            19  □  P0.6/AN6/INT6
Vref/INT4/AN4/P0.4  □  3       18  □  P2.1/AN9/INT9
INT3/AN3/P0.3  □  4    HC89S003F4    17  □  P2.2/AN10/INT10
FLT2/INT2/AN2/P0.2  □  5       16  □  P2.3/TDI/INT11
FLT1/INT1/AN1/P0.1  □  6       15  □  P2.4/TDO/INT12
GND  □  7                      14  □  P2.5/TCK/INT13
FLT0/INT0/AN0/P0.0  □  8       13  □  P2.6/TMS/INT14/PLVD
VDD  □  9                      12  □  P2.7/RST/INT15
XOUT/P1.1  □  10               11  □  P1.0/XIN
```
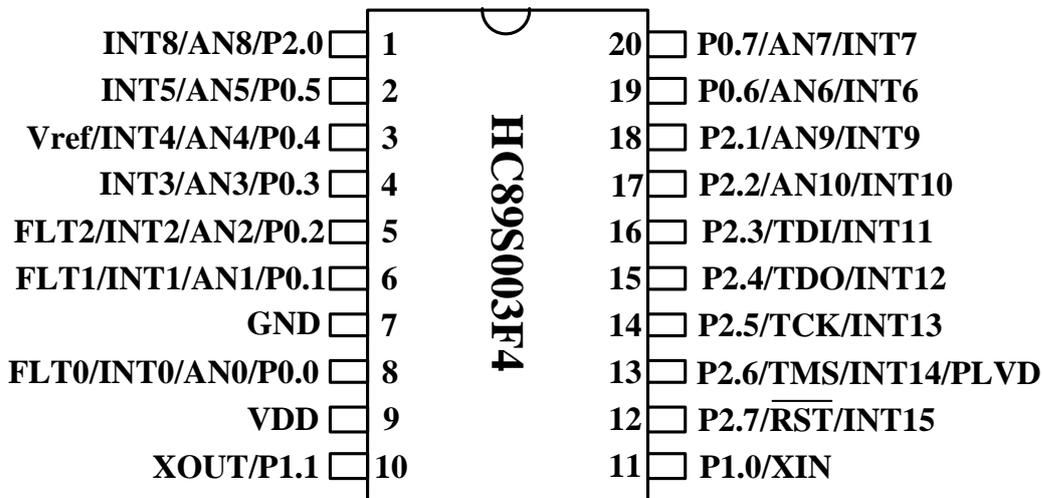
Figure 1-2 SOP20/TSSOP20 pin configuration diagram

1.4 Pin description

Pin	Name	Type	Introductions
1	P2.0	I/O	Input/output port
	AN8	AN	ADC8 input port
	INT8	I	External interrupt 8, input port
2	P0.5	I/O	Input/output port
	AN5	AN	ADC5 input port
	INT5	I	External interrupt 5, input port
3	P0.4	I/O	Input/output port
	AN4	AN	ADC4 input port
	Vref	AN	ADC external reference voltage input/output port
	INT4	I	External interrupt 4, input port
4	P0.3	I/O	Input/output port
	AN3	AN	ADC3 input port
	INT3	I	External interrupt 3, input port
5	P0.2	I/O	Input/output port
	AN2	I	ADC2 input port
	INT2	I	External interrupt 2, input port
	FLT2		PWM2 fault detection of input pins
6	P0.1	I/O	Input/output port
	AN1	I	ADC1 input port
	INT1	I	External interrupt 1, input port
	FLT1		PWM1 fault detection of input pins
7	GND	P	Power ground
8	P0.0	I/O	Input/output port
	AN0	AN	ADC0 input port
	INT0	I	External interrupt 0, input port
	FLT0	I	PWM0 fault detection of input pins
9	VDD	P	Power input
10	P1.1	I/O	Input/output port
	XOUT		External oscillator output
11	P1.0	I/O	Input/output port
	XIN		External oscillator input
12	P2.7	I/O	Input/output port

	$\overline{RST}$	I	External reset input port
	INT15	I	External interrupt 15, input port
13	P2.6	I/O	Input/output port
	TMS	I	JTAG mode input
	INT14	I	External interrupt 14, input port
	PLVD	AN	Port low voltage detection port
14	P2.5	I/O	Input/output port
	TCK	I	JTAG clock input
	INT13	I	External interrupt 13, input port
15	P2.4	I/O	Input/output port
	TDO	O	JTAG data output
	INT12	I	External interrupt 12, input port
16	P2.3	I/O	Input/output port
	TDI	I	JTAG data input
	INT11	I	External interrupt 11, input port
17	P2.2	I/O	Input/output port
	AN10	I	ADC10 input port
	INT10		External interrupt 10, input port
18	P2.1	I/O	Input/output port
	AN9	AN	ADC9 input port
	INT9	I	External interrupt 9, input port
19	P0.6	I/O	Input/output port
	AN6	AN	ADC6 input port
	INT6	I	External interrupt 6, input port
20	P0.7	I/O	Input/output port
	AN7	AN	ADC7 input port
	INT7	I	External interrupt 7, input port

1.5　Peripheral function Ports total mapping module PTM

HC89S003F4 has peripheral function Ports total mapping module internal, by software user can configure most peripheral function to arbitrary port except power port (VDD, GND).

1.5.1　PTM module characteristics

➢ When set peripheral port as input (T0/1/3/5 external input, RXD and so on) function, system permit multi to one mapping, that is multi-input peripheral functions port are distributed the same IO, the method will optimize the user's system.

➢ When set peripheral port as output (T0/1/4 clock output, TXD and so on) function, if multi-output peripheral functions port are distributed the same IO, it will follow fixed priority, only one output is valid.

➢ Software operation, use flexible, when use design system, don't care the Pins layout of peripheral functions, it can reduce the development cost.

➢ When user meets layout errors of peripheral function Pins on PCB, user can re-distribute peripheral functions by PTM module, and shorten development period.

➢ When user changes the peripheral components during system design, only need minimum changes, it will reduce the cost of system maintenance.

1.5.2　PTM support peripheral function Ports total mapping

Peripheral	Name	Type	Instructions
Timer	T0	I/O	T0 external input orT0 clock scale output
	T1	I/O	T1 external input orT1 clock scale output
	T3	I	T3 external input
	T4	O	T4 ouput

	T5	I	T5 external input
PWM	PWM0	O	PWM0 output port
	PWM01	O	PWM01 output port
	PWM1	O	PWM1 output port
	PWM11	O	PWM11 output port
	PWM2	O	PWM2 output port
	PWM21	O	PWM21 output port
	PWM3	O	PWM3 output port
CLK	CLKO	O	Clock output port
UART	TXD	O	UART1 data transmission port
	RXD	I/O	UART1 receive port
	TXD2	O	UART2 data transmission port
	RXD2	I	UART2 receive port
SPI	MOSI	I/O	SPI data port, master output and slave input
	MISO	I/O	SPI data port, master input and slave output
	SCK	I/O	SPI clock port
	$\overline{\text{SS}}$	I	SPI chip select port

1.5.3　PTM dose not support peripheral function Ports total mapping

PTM does not support peripheral function Ports total mapping include power port(VDD, GND),PWM fault detection Pin (FLT0/1/2), ADC input, INT0-15 function port, oscillator Pin(Xin,XOUT),external reset Port(RST), LVD voltage detection port(LVDI).

2 CPU

2.1 CPU characteristics

HC89S003F4 CPU is an enhanced 1T compatible with 8051 core, it run faster than traditional 8051 under the same system clock, and has better performance characteristics.

2.2 CPU registers

2.2.1 PC program counter PC

Program counter PC is independent physically, does not belong to SFR. PC word length is 16 bits, and used to control the execution sequence of instructions register. After microcontroller power on or reset,PC value is 0000H, program is executed from 0000H address, if second reset vector is enabled, then after power on or reset, microcontroller will execute program from the second reset vector address.

2.2.2 Accumulator ACC

Accumulator (ACC) as A in instruction system, and used to provide ALU operands and store the arithmetic result, it is CPU most frequent work register, most execution of the instructions via the accumulator ACC.

2.2.3 Register B

Register B is set for multiplication and division registers specifically, used to store the operands and result of the arithmetic of multiplication and division. at the time no multiplication or division, it can be used as a general purpose register.

2.2.4 Program state word register PSW

This register is used to save characteristics and the processing state of the ALU arithmetic result, and the characteristics and state as the condition of controlling program transfer, for program checking and querying, the bits are defined as follows:

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Reset values	0	0	0	0	0	0	0	0
Flag	CY	AC	F0	RS[1:0]		OV	F1	P

Bit	Flag	Introductions
7	CY	Carry/borrow flag 0 : In arithmetic, no a carry or borrow 1 : In arithmetic, carry or borrow has occurred
6	AC	Auxiliary carry/borrow flag 0 : In arithmetic, no auxiliary carry or borrow 1 : In arithmetic, auxiliary carry or borrow has occurred
5	F0	User-defined flag

4-3	RS[1:0]	register group selection flag
		    00 : 0 Group (00H ~ 07H)
		    01 : 1 Group (08H ~ 0FH)
		    10 : 2 Group (10H ~ 17H)
		    11 : 3 Group (18H ~ 1FH)

2	OV	Overflow flag 0: no overflow 1 : Overflow has occurred
1	F1	User-defined flag
0	P	Parity bit 0 : sum of 1 in ACC register is 0 or even 1 : Sum of 1 in ACC register is odd

2.2.5 Stack pointer SP

SP is a 8 bits special register, it indicates the top of the stack in the internal RAM position. After MCU reset,SP value is 07H, the stack was actually performed from the 08H unit, considering the 08H~1FH units belong to work register 1~3 respectively, and if in the program user needs to use these areas, the SP value better should be set a large value. 51MCU stack is upward generated, such as: SP=30H,after CPU execute a instruction or response a interrupt,PC push stack, PCL protected to 31H,PCH protected to 32H,SP=32H.

2.2.6 Data pointer DPTR

Data pointer DPTR is a 16 bits special register, it is consists of two 8 registers DPH (high 8 bits) and DPL (low 8 Bits). The series MCU has two 16 bits data pointer of DPTR0 and DPTR1, they share the same address, user can set DPS (INSCON. 0) to select the data pointer.

2.2.7 Data pointer select register INSCON

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R/W	R	R	R	R/W
Reset values	0	0	0	0	0	0	0	0
Flag		-		IAPS		-		DPS

Bit	Flag	Introductions
7-5	-	Reserved (read = 0b, write invalid)
4	IAPS	MOVC operation selection bit 0 : program area read/erase/write operation 1 : OPTION read operation
3-1	-	Keep (read = 0b, write invalid)
0	DPS	Data pointer selection bit 0: Data pointer DPTR0 1: Data pointer DPTR1

3 Memory

3.1 The program memory(flash)

3.1.1 Flash characteristics

- ➢ Support erase and program in all operating voltage
- ➢ In-circuit programming (ICP) support write, read, and erase operations
- ➢ ICP mode supports 32 bits password protection
- ➢ In-application programming (IAP) supports user-defined startup code and flash simulation of EEPROM
- ➢ Flexible code protection mode
- ➢ 100k erase times at least
- ➢ 10 years data retention at least

3.1.2 Flash data security

Flash operation is divided into two modes: first mode is flash read/erase/write through flash programmer, this is called in-circuit programming mode (ICP), JTAG is one of ICP; second mode is the user code run in flash code area, it can read/write/erase the other sectors of flash memory, but unable to erase the code in sector itself, which is called in-application programming mode (IAP).

3.1.2.1 User ID password protection

User can protect the ICP operation by setting password during PC software, password lengths are 4 bytes (32 bits), once password is set, only input the correct password, user can enter JTAG mode, otherwise any operation of flash is invalid, the password can protect the user's code availably.

3.1.2.2 ICP read/erase/write flash protection

ICP read protection unit is 4K bytes, when 4K bytes space read protection enabled, read data is all 0 by ICP read, but user can still do simulation by ICP operation.

ICP erase and write protection unit are 4K bytes too, when the corresponding 4K bytes erase and write protection enabled, ICP will not be able to erase and program 4K bytes, strong writing is disabled.

If the corresponding 4K bytes read protection is enabled, but erase and write operations are enabled, user can get read access after erase until reset or power-down.

ICP read, erase and write protections are configured by PC software, and the detail descriptions please see HC-51LINK user manual.

3.1.2.3 IAP read/erase/write flash protection

IAP read flash by the instruction MOVC, IAP read protection unit is 4K bytes, if the 4K bytes space read protection is enabled, MOVC instruction in other 4K bytes space only read out data 0 from this 4K bytes, but MOVC instruction in this 4K bytes can read the data itself.

IAP erase and write flash steps are described in flash IAP operation, IAP erase and write protection unit is 4K bytes, before IAP erase and write, the corresponding sectors erase and write protection must be disabled.

If the corresponding 4K bytes read protection is enabled, but erase and write operations are enabled, user can to get read access after erase until reset or power-down.

IAP read, erase and write protections are configured by PC software, and the detail descriptions please see HC-51LINK user manual.

3.1.3 OPTION

There is a read-only OPTION area besides 16K bytes ROM, storage data include: user setting data, user passwords, chip configurations data, the second reset vector data related. Address distribution in below table.

Address	Name	Address offset	Name	Address	Name	Address offset	Name
0x0000	SN_DATA0	0x0020	FLASH_SC0	0x0031	ERST_ENB	0x0100	CHIP_ID0
0x0001	SN_DATA1	0x0021	FLASH_SC1	0x0038	WAIT_TS	0x0101	CHIP_ID1
0x0002	SN_DATA2	0x0022	FLASH_SC2	0x0039	BORVS	0x0102	CHIP_ID2
0x0003	SN_DATA3	0x0023	FLASH_SC3	0x003E	RVCFG	0x0103	CHIP_ID3
0x0004	SN_DATA4	-	-	0x003F	nRVCFG	0x0104	CHIP_ID4
0x0005	SN_DATA5	-	-	-	-	0x0105	CHIP_ID5
0x0006	SN_DATA6	-	-	-	-	0x0106	CHIP_ID6
0x0007	SN_DATA7	-	-	-	-	0x0107	CHIP_ID7
0x0008	ID_DATA0	-	-	-	-	-	-
0x0009	ID_DATA1	-	-	-	-	-	-
0x000A	ID_DATA2	-	-	-	-	-	-
0x000B	ID_DATA3	-	-	-	-	-	-
0x000C	ID_DATA4	-	-	-	-	-	-
0x000D	ID_DATA5	-	-	-	-	-	-
0x000E	ID_DATA6	-	-	-	-	-	-
0x000F	ID_DATA7	-	-	-	-	-	-

HC89S003F4 will be configured a 8 bytes CHIP_ID before leave the factory, the CHIP_ID is unique and not repeated, user can read it by MOVC instruction in code.

SN_DATA and ID_DATA are user-defined data, FLASH_SC is user password, it is set by software tools, as well as setting code options, they can be erased or modified, and user can read them by MOVC instruction in code.

Note: 1. User must set register INSCON[IAPS] bit to 1 before read OPTION.

　　　2. First character "n" is complement of the corresponding data.

3.1.3.1 External reset enabled (ERST_ENB)

Bit	7	6	5	4	3	2	1	0
Flag	-							ERST_ENB

Bit	Flag	Introductions
7-1	-	Reserved bits
0	ERST_ENB	Reset pin enable bit 0: External $\overline{RST}$ input 1 : P2.7 as GPIO

3.1.3.2 Wait time of reread OPTION after reset (WAIT_TS)

Bit	7	6	5	4	3	2	1	0
Flag	-	-	-	-	-	-	WAIT_TS	

Bit	Flag	Introductions
7-2	-	Reserved bits
1-0	WAIT_TS	Wait time of reread option after reset selection bits 　　00 : 8ms 　　01 : 4ms 　　11 : 16ms

3.1.3.3 BOR detection voltage selection (BORVS)

Bit	7	6	5	4	3	2	1	0
Flag	-	-	-	-	-	BORVS		

Bit	Flag	Introductions
7-3	-	Reserved bits
2-0	BORVS	BOR detection of voltage selection bits 000 : 1.8V 001 : 2.0V 010 : 2.4V 011 : 2.6V 100 : 3.0V 101 : 3.6V 110 : 3.9V 111 : 4.2V

3.1.3.4 Second reset vector configuration (RVCFG)

Bit	7	6	5	4	3	2	1	0
Flag	RVSEN	-			RVADR[3:0]			

Bit	Flag	Introductions
7	RVSEN	The second reset vector enable bit 0: disable the second reset vector 1: enable the second reset vector
6-4	-	Reserved bits
3-0	RVADR[3:0]	The second reset vector configuration values The second reset vector address = {RVADR[3:0],0000000000B} Note: 1. When RVADR[3:0]=0, the second reset vector address coincide with 0x0000H. 2. RVADR[3:0] configuration data only equal 1000, 1100, 1110, 1111 four values, the second reset vector space only is 1K, 2K, 4K, 8K.

3.1.4 Flash IAP operation

Before flash IAP erase and write, user need to configure extension SFR FREQ_CLK register, and indicates the current CPU frequency, FREQ_CLK configuration value is equal to CPU clock frequency, the minimum value is 1MHz, If CPU current frequency is 16MHz, user must configure the value in register FREQ_CLK=0x10 . Recommended before IAP erase and write, CPU clock frequency division factor is an integer. When CPU clock frequency below 1MHz, flash IAP erase and write operation is disabled.

3.1.4.1 IAP data register (IAP_DATA)

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	IAP_DATA[7:0]							

Bit	Flag	Introductions
7-0	IAP_DATA[7:0]	IAP data register

3.1.4.2 IAP address register IAP_ADDRL, IAP_ADDRH

IAP_ADDRL

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	1	1	1	1	1	1	1	1
Flag	IAP_ADDR[7:0]							

Bit	Flag	Introductions
7-0	IAP_ADDR[7:0]	Low 8 bits of the IAP operation address register

IAP_ADDRH

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	1	1	1	1	1	1
Flag	-		IAP_ADDR[13:8]					

Bit	Flag	Introductions
7-6	-	Reserved
5-0	IAP_ADDR[13:8]	High 6 bits of the IAP operation address register

Note: User can modify IAP address register only in unlocked status, and once operation is completed, IAP address is pointed to 0x3FFF automatically.

3.1.4.3 IAP Command register IAP_CMDH, IAP_CMDL

IAP_CMDH

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	IAP_CMDH[7:0]							

Bit	Flag	Introductions
7-0	IAP_CMDH[7:0]	Operation mode selection bit 0xF0 : Unlock (22 CPU clock automatically locked, IAP_CMD[7:0] = 0x00) 0xE1 : Trigger one time action 0xD2 : Sector erase 0xB4 : Byte program 0x87 : Software reset, reset address 0000H, not reread codes options 0x78 : Software reset, reset address 0000H, reread codes options Other values: lock

IAP_CMDL

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	1	1	1	1	1	1	1	1
Flag	IAP_CMDL[7:0]								

Bit	Flag	Introductions
7-0	IAP_CMDL[7:0]	IAP_CMDH[7:0] complement code Note: Write into IAP_CMDL[7:0] data must equal the complement of IAP_CMDH[7:0] data previous, otherwise operations will be locked, meanwhile operation will fail.

Examples:

1, Program space sector erase

    IAP_CMDH = 0xF0;

    IAP_CMDL = 0x0F;

    IAP_ADDRL = 0x80;

    IAP_ADDRH = 0x00;   // Select first sector to be erased, a sector for 128 bytes

    IAP_CMDH = 0xD2;   // Select operation mode, sector erase

    IAP_CMDL = 0x2D;

    IAP_CMDH = 0xE1;   // Trigger

    IAP_CMDL = 0x1E;   // After trigger IAP_ADDRL Links to 0xFF, IAP_ADDRH Links to 0x3F,

meanwhile locked automatically

2, program space byte program

    IAP_DATA = 0x02; //Data ready to be programmed before writing data register must be unlocked

    IAP_CMDH = 0xF0;

    IAP_CMDL = 0x0F;

    IAP_ADDRH = 0x00;

    IAP_ADDRL = 0x00;

    IAP_CMDH = 0xB4;   // Select the mode of operation, byte program

    IAP_CMDL = 0x4B;

    IAP_CMDH = 0xE1;   // Trigger

    IAP_CMDL = 0x1E;   // After the trigger IAP_ADDRL Links to 0xFF, IAP_ADDRH Links to

0x3F, IAP_DATA Links to 0x00, meanwhile locked automatically

    Note: After unlocked, write address, select operation mode, trigger, between these three steps, any instruction cannot be inserted, and must be operated continuously.

3, Software reset (do not reread code options)

    IAP_CMDH = 0xF0;

    IAP_CMDL = 0x0F;

    IAP_CMDH = 0x87;

    IAP_CMDL = 0x78;

4, Software reset (reread code options)

    IAP_CMDH = 0xF0;

    IAP_CMDL = 0x0F;

    IAP_CMDH = 0x78;

    IAP_CMDL = 0x87;

3.1.5 Flash ICP operation

    User can use HC-51LINK emulator to program MCU, after MCU is already welded in the user board, if user uses power-on reset enter JTAG mode, only links 6 cables, and user must power-down the system, and power supplied by the emulator. When user does not want to power-down the system, it need 7 cables to enter the programming mode, add a reset Pin, detailed instructions of emulator, please see HC-51LINK user manual.

    In addition, because the programming signals are very sensitive, user needs to use 6 jumpers to separate programming Pins (VDD, TDO, TDI, TMS, TCK, $\overline{RST}$) from the circuit, as shown in below figure.
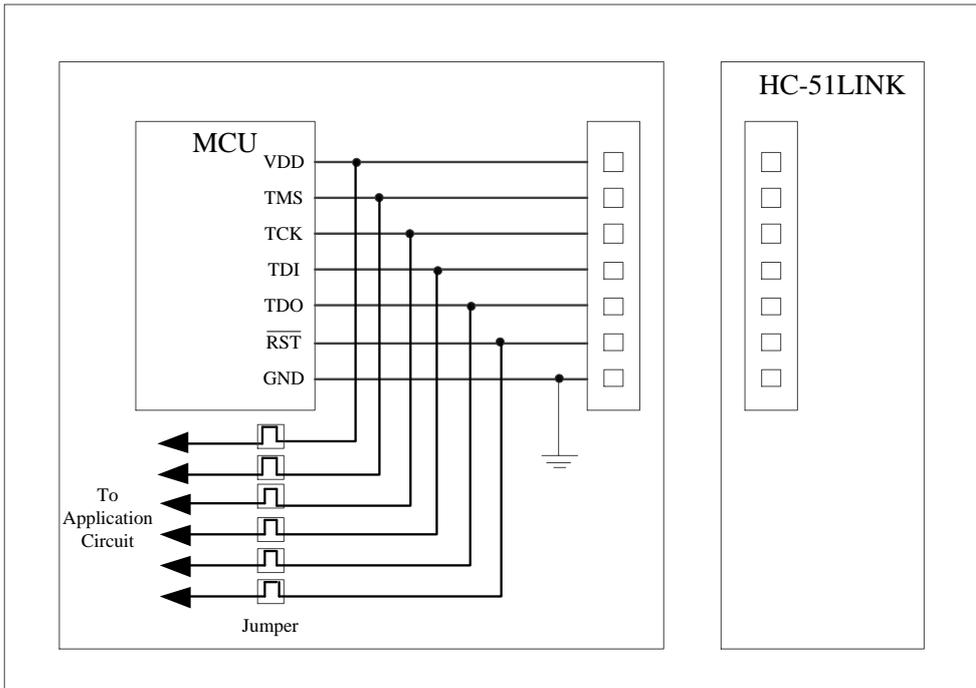
Figure 3 - 1 HC-51LINK programming hardware connection

When using ICP operation mode, it is recommended operation according to the following steps:

1, Disconnect jumpers (jumper) before start programming, , separate programming pins from the application circuit.

2, Connect the chip programming Pin flash program interface, then start programming.

3, After the program ends, disconnect flash Programming interfaces, connect jumper to application circuit.

Note: Chip power on, reset Pin is valid in default, and low level is reset, so when user use the reset Pin , it cannot be set pull-down, otherwise the chip is in the reset state always, and cannot be programming and simulation.

3.1.6 Second reset vector operations

If the user has configured second reset vector enabled in the code options and the second reset vector address, then after the on-chip power-on reset, PC first point to the second vector address, and begin to implement user's startup program, if at the end of user code need place a un-reread code item of software reset program, that user program will be reset to 0x0000H, start to implement the user application program.

3.2 Data storage (RAM)

HC89S003F4 provide user with a 256 bytes internal RAM and 256 bytes internal expansion RAM as data memory. Below is data memory space allocation.
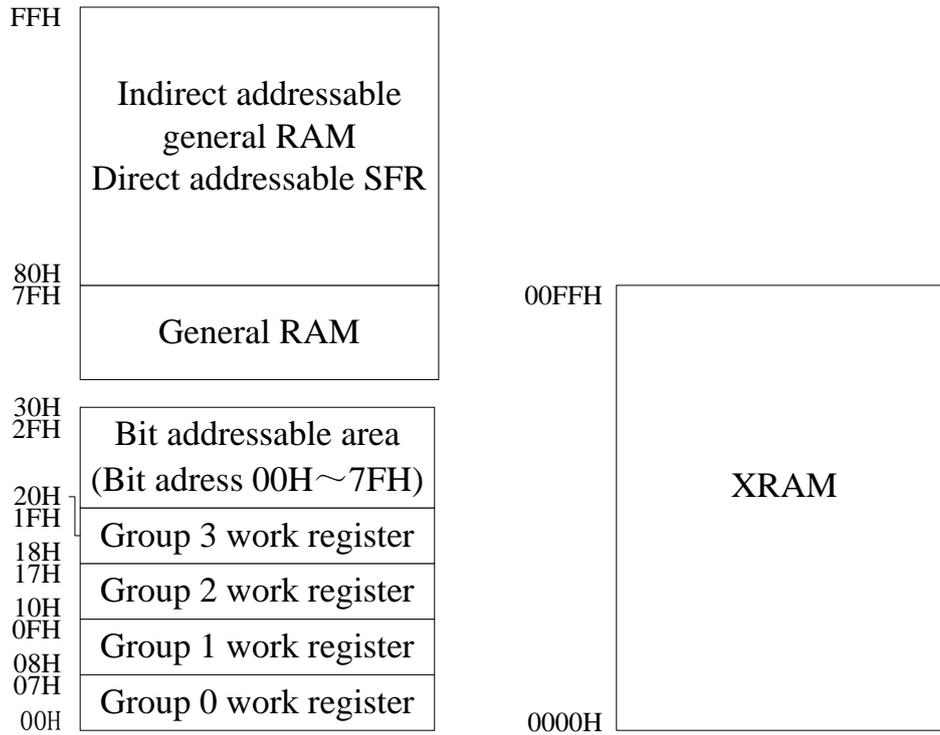
Figure 3 - 2 Data memory map

Internal RAM high 128 bytes (0x80 ~ 0xFF) must use the register indirect addressing modes.

Internal expansion RAM (XRAM) addresses range is 0x0000 ~ 0x00FF, and access to internal extensions RAM methods same as traditional 8051 access external extensions RAM, but it does not affect I/O port. In assembly language, access internal expansion RAM through MOVX instruction, as MOVX @DPTP or MOVX @Ri.

3.3 Special function registers (SFR)

3.3.1 Special function registers list

3.3.1.1 Direct addressing, read and write SFR

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8	RSTFR	IAP_ADDRL	IAP_ADDRH	IAP_DATA	IAP_CMDL	IAP_CMDH	-	-
F0	B	PWM2EN	PWM2PL	PWM2PH	PWM2DL	PWM2DH	PWM2DTL	PWM2DTH
E8	-	PWM1EN	PWM1PL	PWM1PH	PWM1DL	PWM1DH	PWM1DTL	PWM1DTH
E0	ACC	PWM0EN	PWM0PL	PWM0PH	PWM0DL	PWM0DH	PWM0DTL	PWM0DTH
D8	-	-	PWM0C	PWM1C	PWM2C	PWM3C	PWM3P	PWM3D
D0	PSW	-	-	-	-	-	-	-
C8	-	T3CON	TL3	TH3	T4CON	TL4	TH4	-
C0	-	T5CON	TL5	TH5	RCAP5L	RCAP5H	-	-
B8	IE1	IP2	IP3	LVDC	-	WDTC	CRCL	CRCH
B0	-	-	-	-	ADCC0	ADCC1	ADCRL	ADCRH
A8	IE	IP0	IP1	SPDAT	SPCTL	SPSTAT	-	-
A0	P2	-	INSCON	-	-	-	-	-
98	SCON	SBUF	SADDR	SADEN	-	-	SCON2	-
90	P1	-	-	-	-	-	PINTF0	PINTF1
88	TCON	TMOD	TL0	TL1	TH0	TH1	CLKSWR	CLKCON
80	P0	SP	DPL	DPH	-	-	-	PCON

3.3.1.2 External extension XSFR

The method to access extension XSFR is the same as XRAM, use MOVX A, @DPTR and MOVX @DPTR,A to read and write.

For example: write XSFR at address 0xFE88, operation as below:

MOV A, #wdata
MOV DPTR, #0xFE88
MOVX @DPTR, A

Read XSFR at address 0xFE89, operation as below:

MOV DPTR, #0xFE89
MOVX A, @DPTR

Extension XSFR (base address is 0xFE80)

Offset address	XSFR	Offset address	XSFR	Offset address	XSFR	Offset address	XSFR
0x0000	TCON1	0x0010	-	0x0020	WDTCCR	0x0030	PITS0
0x0001	-	0x0011	CLKDIV	0x0021	-	0x0031	PITS1
0x0002	-	0x0012	FREQ_CLK	0x0022	CRCC	0x0032	PITS2
0x0003	-	0x0013	CLKOUT	0x0023	-	0x0033	PITS3
0x0004	-	0x0014	RC_PD_EN	0x0024	BORC	0x0034	-
0x0005	T5CON1	0x0015	SPOV_RSTEN	0x0025	BORDBC	0x0035	-
0x0006	-	0x0016	-	0x0026	-	0x0036	-
0x0007	-	0x0017	-	0x0027	LVDDBC	0x0037	-
0x0008	S2CON	0x0018	ADCWC	0x0028	-	0x0038	PINTE0
0x0009	S2CON2	0x0019	-	0x0029	-	0x0039	PINTE1
0x000A	S2BUF	0x001A	-	0x002A	RSTDBC	0x003A	-
0x000B	-	0x001B	ADCC2	0x002B	-	0x003B	-
0x000C	-	0x001C	-	0x002C	-	0x003C	INT01_PINS
0x000D	-	0x001D	-	0x002D	-	0x003D	-
0x000E	-	0x001E	-	0x002E	-	0x003E	-
0x000F	-	0x001F	-	0x002F	-	0x003F	-

Extension XSFR (base address is 0xFF00)

Offset address	XSFR	Offset address	XSFR	Offset address	XSFR	Offset address	XSFR
0x0000	P0M0	0x0010	P2M0	0x0020	-	0x0030	-
0x0001	P0M1	0x0011	P2M1	0x0021	-	0x0031	-
0x0002	P0M2	0x0012	P2M2	0x0022	-	0x0032	-
0x0003	P0M3	0x0013	P2M3	0x0023	-	0x0033	-
0x0004	-	0x0014	-	0x0024	-	0x0034	-
0x0005	P0LPU	0x0015	-	0x0025	-	0x0035	-
0x0006	-	0x0016	-	0x0026	-	0x0036	-
0x0007	-	0x0017	-	0x0027	-	0x0037	-
0x0008	P1M0	0x0018	-	0x0028	-	0x0038	-
0x0009	-	0x0019	-	0x0029	-	0x0039	-
0x000A	-	0x001A	-	0x002A	-	0x003A	-
0x000B	-	0x001B	-	0x002B	-	0x003B	-
0x000C	-	0x001C	-	0x002C	-	0x003C	-
0x000D	-	0x001D	-	0x002D	-	0x003D	-
0x000E	-	0x001E	-	0x002E	-	0x003E	-
0x000F	-	0x001F	-	0x002F	-	0x003F	-

Extension XSFR (base address is 0xFF40)

Offset address	XSFR	Offset address	XSFR	Offset address	XSFR	Offset address	XSFR
0x0000	P00DBC	0x0010	-	0x0020	-	0x0030	-
0x0001	P01DBC	0x0011	-	0x0021	-	0x0031	-
0x0002	P02DBC	0x0012	-	0x0022	-	0x0032	-
0x0003	-	0x0013	-	0x0023	-	0x0033	-
0x0004	-	0x0014	-	0x0024	-	0x0034	-
0x0005	-	0x0015	-	0x0025	-	0x0035	-
0x0006	-	0x0016	-	0x0026	-	0x0036	-
0x0007	-	0x0017	-	0x0027	-	0x0037	-
0x0008	-	0x0018	-	0x0028	-	0x0038	-
0x0009	-	0x0019	-	0x0029	-	0x0039	-
0x000A	-	0x001A	-	0x002A	-	0x003A	-
0x000B	-	0x001B	-	0x002B	-	0x003B	-
0x000C	-	0x001C	-	0x002C	-	0x003C	-
0x000D	-	0x001D	-	0x002D	-	0x003D	-
0x000E	-	0x001E	-	0x002E	-	0x003E	-
0x000F	-	0x001F	-	0x002F	-	0x003F	-

Extension XSFR (base address is 0xFF80)

Offset address	XSFR	Offset address	XSFR	Offset address	XSFR	Offset address	XSFR
0x0000	T0_MAP	0x0010	PWM0_MAP	0x0020	TXD_MAP	0x0030	-
0x0001	T1_MAP	0x0011	PWM01_MAP	0x0021	RXD_MAP	0x0031	-
0x0002	-	0x0012	-	0x0022	-	0x0032	-
0x0003	T3_MAP	0x0013	-	0x0023	-	0x0033	-
0x0004	T4_MAP	0x0014	PWM1_MAP	0x0024	_MAP	0x0034	-
0x0005	T5_MAP	0x0015	PWM11_MAP	0x0025	SCK_MAP	0x0035	-
0x0006	-	0x0016	-	0x0026	MOSI_MAP	0x0036	-
0x0007	-	0x0017	-	0x0027	MISO_MAP	0x0037	-
0x0008	-	0x0018	PWM2_MAP	0x0028	TXD2_MAP	0x0038	-
0x0009	-	0x0019	PWM21_MAP	0x0029	RXD2_MAP	0x0039	-
0x000A	-	0x001A	-	0x002A	-	0x003A	-
0x000B	-	0x001B	-	0x002B	-	0x003B	-
0x000C	-	0x001C	PWM3_MAP	0x002C	-	0x003C	-
0x000D	-	0x001D	-	0x002D	-	0x003D	-
0x000E	-	0x001E	-	0x002E	-	0x003E	-
0x000F	CLKO_MAP	0x001F	-	0x002F	-	0x003F	-

4 The system clock

4.1 Characteristics of the system clock

HC89S003F4 MCU system clock have 2 optional clock sources: external high-frequency RC clock (4MHz~24MHz), external low-frequency RC clock (32.768KHz), internal high-frequency RC clock (32 MHz) and internal low frequency RC clock (44KHz). The internal frequency RC scope of error is less than 1% during -40℃ ~+85℃. Select the system clock (if user choose an internal high-frequency of RC, clock is divided after setting RC32M_DIV[1:0] as osc_clk, the frequency is f_{OSC}, period is t_{OSC}, mainly used for peripheral modules, osc_clk can be divided by any value between 1-255, clock divided as CPU clock, frequency is $f_{CPU,}$ period is t_{CPU}.

CPU can run under 20MHz highest frequency, if frequency of clock selected is higher than 20MHz, the clock need to be divided to meet CPU Clock equal to or less than 20MHz .

Internal low frequency RC (RC44K) output clock marker as wdt_clk, for the watchdog timer count, and can also be used for the system clock, internal high frequency RC (RC32M) output marker as rc32m_clk, and it can be divided by 1/2/4/8. Select external oscillator marker as xtal_clk by setting XTALCFG register.
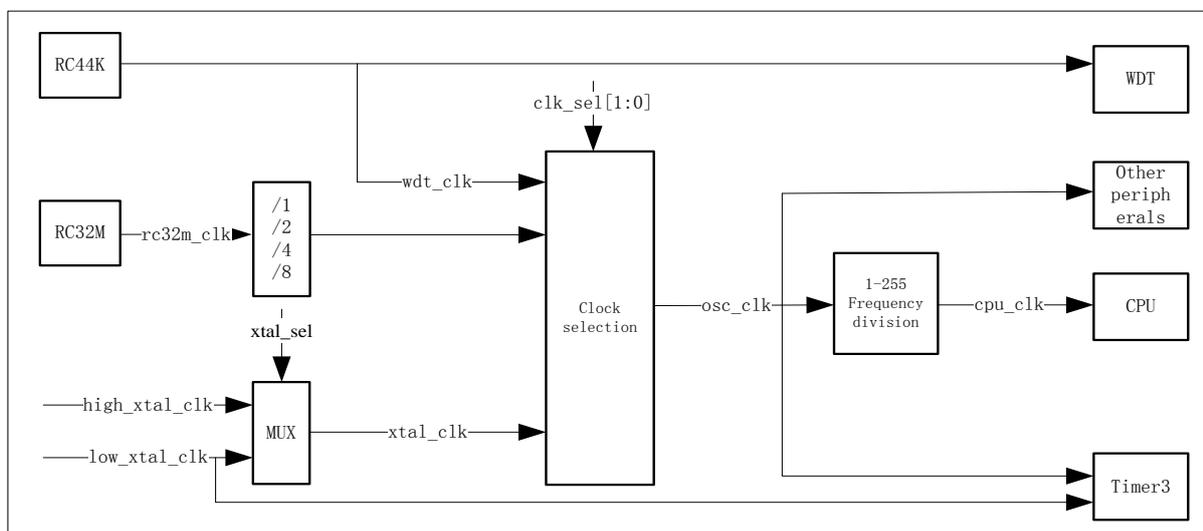
Figure 4 - 1 System clock block diagram

4.2 System clock registers

4.2.1 Clock control register CLKCON

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R/W	R/W	R
Reset values	0	0	1	1	0	0	1	0
Flag	HXTALRDY	LXTALRDY	HSRCRDY	LSRCRDY	-	XTALEN	HSRCEN	-

Bit	Flag	Introductions
7	HXTALRDY	External high-frequency RC oscillator state bit 0: External high-frequency RC is not ready 1: External high-frequency RC is ready Note: The hardware automatically clear 0 or set 1
6	LXTALRDY	External low-frequency RC oscillator state bit 0: External low-frequency RC is not ready 1: External low-frequency RC is ready Note: The hardware automatically clear 0 or set 1
5	HSRCRDY	Internal high-frequency RC oscillator state bit

		0: Internal high-frequency RC is not ready 1: Internal high-frequency RC is ready Note: the hardware automatically clear 0 or set 1
4	LSRCRDY	Internal low frequency RC oscillator state bit 0: Internal low frequency RC is not ready 1: Internal low frequency RC is ready Note: the bit hardware automatically clear 0 or set 1
3	-	Reserved bit

2	XTALEN	External oscillator enable bit
		
		   0：External oscillator close
		
		   1：External oscillator open
		
		Note: Shen enabled, need set the corresponding IO mode to analog channel by
		
		software.

1	HSRCEN	Internal high-frequency RC oscillator enable bit 0: Internal high-frequency RC close 1: Internal high-frequency RC open
0	-	Reserved bit

4.2.2　Select clock register CLKSWR

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W
Reset values	0	1	0	1	0	0	1	1
Flag	CLKSTA[1:0]		CLKSEL[1:0]		-		RC32M_DIV[1:0]	

Bit	Flag	Introductions
7-6	CLKSTA[1:0]	System clock state bits 00: Current system clock is internal low frequency RC 01: Current system clock is internal high frequency RC 10: Current system clock is external low frequency RC 11: Current system clock is external high frequency RC Notes: system automatically switches state based on current system clock
5-4	CLKSEL[1:0]	System clock selection bit 00: Select system clock to internal low frequency RC 01: Select system clock to internal high frequency RC Note: corresponding clock source state bits must be set to 1 when select system clock, or use previous clock, after switching, the original clock does not automatically close; select the system clock marker as osc_clk, the frequency is $F_{osc,}$ period is T_{osc} .
3-2	-	Reserved bit

1-0	RC32M_DIV[1:0]	Internal high frequency RC scale bits
		   00 : rc32m_clk
		   01 : rc32m_clk /2
		   10 : rc32m_clk /4
		   11 : rc32m_clk /8(default)

4.2.3 Clock scale register (CLKDIV)

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	1	0
Flag	CLKDIV[7:0]							

Bit	Flag	Introductions
7-0	CLKDIV[7:0]	CPU clock division factor, default value is 2 Configuration values is 0 or 1, clock is not divided; in other condition, the configuration value is equal to the frequency factor; Note: Clock after divided is CPU clock, frequency is F_{cpu}, period is T_{cpu}.

4.2.4 Clock output register CLKOUT

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R/W	R	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-			CLK_OUT_EN	-	CLK_OUT_ SEL[2:0]		

Bit	Flag	Introductions
7-5	-	Reserved bit
4	CLK_OUT_EN	Clock output enable bit 0 : Disable clock output 1 : Enable clock output
3	-	Reserved bit
2-0	CLK_OUT_ SEL [2:0]	output clock selection bits 000 : Select cpu_clk 001 : Select osc_clk 010 : Select wdt_clk 011 : Select xtal_clk 100 : Select rc32m_clk 101 : Select rc32m_clk/2 110 : Select rc32m_clk/4 111 : Select rc32m_clk/8

4.2.5 External oscillator configuration register XTALCFG

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	HXTAL_CFG		LXTAL_CFG		HXTAL_MODE_SEL		RC_PD_EN	XTAL_SEL

Bit	Flag	Introductions
7-6	HXTAL_CFG	External high frequency oscillator warmup count value selection bit 00：2048 01：256 10：16384 11：65536

5-4	LXTAL_CFG	External low frequency oscillator warmup count value selection bit 00：16384 01：4096 10：1024 11：65536
3-2	HXTAL_MODE_SEL	External high frequency oscillator selection bit 00：Select 4M/8M oscillator 01：Select 4M/8Moscillator strong drive mode, oscillator sartup time is short whenat low voltage, and power consumption is suitable. 11：select 6M/24M oscillator
1	RC_EN_PD	Internal high frequency RC closed when generate system BOR 0：Do not close internal high frequency RC when system BOR generated 1：Close internal high frequency RC when system BOR generated Note: this bit is used to BOR enabled, to reduce the system power consumption during VDD drops.
0	XTAL_SEL	External oscillator selection bit 0：External low frequency oscillator selection bit 32.768KHz 1：External high frequency oscillator

4.2.6　CPU clock frequency register FREQ_CLK

Before flash IAP erase and write, user need to configure extension SFR FREQ_CLK register, and indicates the current CPU frequency, FREQ_CLK configuration value is equal to CPU clock frequency, the minimum value is 1MHz, If CPU current frequency is 16MHz, user must configure the value in register FREQ_CLK=0x10

FREQ_CLK

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	FREQ_CLK[7:0]							

Bit	Flag	Introductions
7-0	FREQ_CLK[7:0]	Current CPU clock frequency register

5 Power management

5.1 Power management characteristics

➢ Provide idle mode (IDLE) and power-down mode (PD), as a power saving mode
➢ Provide a variety of ways to wake up from the idle/power-down mode
➢ Provide low frequency mode (it is clock division, described in the system clock chapter)

5.2 Idle mode

System power consumption can be reduced in idle mode, in this mode, the program terminate run, CPU clock stop, but external device clock continues to run. In idle mode, the CPU stop in determining state, and all CPU states was saved before entering idle mode, such as the PC, PSW, SFR, RAM and so on.

Set PCON register IDL bit to 1, then HC89S003F4 enters idle mode. IDL bit set 1 is the last instruction executed before CPU enter idle mode.

Two ways to exit the idle mode:

(1) All valid interrupts. When HC89S003F4 detects a valid interrupt, CPU clock is recovered immediately, hardware clear PCON register IDL bit automatically, and then execute the interrupt service program, then jump to execute the instruction after enter idle mode instruction.

(2) The reset signal (valid level on external reset Pin, WDT reset, BOR reset, low-voltage detection reset on external ports). After HC89S003F4 detects a valid reset signal, IDL in PCON register is reset to 0, system program will start to run from the reset address 0000H, RAM remains unchanged, SFR value changes depend on the value of different function module.

5.3 Power-down mode

HC89S003F4 will enter very low power consumption state in power-down mode. In power-down mode CPU and peripherals of all clock signal will stop, but if WDT and TIMER3 enabled and permits working in power down mode, then the WDT and TIMER3 module will continue to work. Before enter the power-down mode all the CPU states were saved, such as the PC, PSW, SFR, RAM and so on.

Set PCON register PD bit to 1, HC89S003F4 will enter the power-down mode. PD set 1 is the last instruction executed by CPU before enter the power-down mode.

Note: If user set IDL and PD bits at the same time, HC89S003F4 enter the power-down mode. After exit the power-down mode, CPU couldn't enter idle mode, and hardware will clear the IDL and the PD bits after exit from the power-down mode.

Multiple ways to exit the power-down mode:

(1) Valid external interrupts, LVD interrupt, WDT interrupt and TIMER3 (s) interrupt. Valid external interrupts and TIMER3 (elect external clock as count clock source) interrupt occur, internal high-frequency RC oscillator start up, CPU clock and the peripheral clock is immediately recovered, PCON register PD bit will be clear by hardware, and CPU running external interruption service program. After the completion of external interrupt service, and continue to run the instructions after jump to enter power-down mode.

(2) The reset signal (valid level on external reset Pin, WDT reset, BOR reset or low voltage detection reset on external ports). Valid reset signal will reset PCON register PD bit to 0, oscillator restart, CPU clock and the peripheral clock immediately recovered, system program will start to run from the reset address 0000H, RAM remains unchanged, SFR value changes depend on the value of different function module.

5.4 Power management registers

5.4.1 Power control register PCON

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-				GF1	GF0	PD	IDL

Bit	Flag	Introductions
7-4	-	Reserved (read = 0b,, write invalid)
3	GF1	User normal flag 1
2	GF0	User normal flag 0
1	PD	Power-down mode control bit 0 : Normal mode 1 : Enter power-down mode (clear to 0 automatically after exit)
0	IDL	Idle mode control bit 0 : Normal mode 1 : Enter idle mode (clear to 0 automatically after exit) Note: If set PD&IDL at the same time, the system will enter the power-down mode, meanwhile flag is clear after wake up.

6 Reset
6.1 Reset characteristics
➢ Provides multiple ways to reset
➢ All reset have special flags

6.2 POR (Power-on reset)
During HC89S003F4 power-on, a POR signal will be generated, this signal will reset the microcontroller, meanwhile PORF bit in RSTFR register will be set, and the user can read this flag to determine whether POR reset or not.

Note: After POR reset, RAM data is not stable, it is recommended that user need to reinitialize the RAM, other reset mode does not reset RAM.

6.3 BOR (Brown-out reset)
When VDD voltage drops below V_{BOR}, and continue time is more than T_{BOR}, the system generates undervoltage reset. when BOR reset , BORF bit in RSTFR register is set to 1, the user can read this flag to determine whether BOR reset or not.

User can select HC89S003F4 BOR voltage detection value by code option or register. When the configuration of BOR gear is completed in the code options, user can also reconfigure BOR voltage through the configuration registers. BOR gear: 4.2V/3.9V/3.6V/3.0V/2.6V/2.4V/2.0V/1.8V.

BOR voltage detection circuit has a certain hysteresis, hysteresis voltage is about 0.1V. When VDD voltage drops to BOR voltage gear selected, BOR is valid; and VDD voltages needed to rise to BOR voltage +0.1V, BOR reset removed.

Undervoltage reset diagram shown below, T_{BOR} configuration by register used to voltage debouncing.
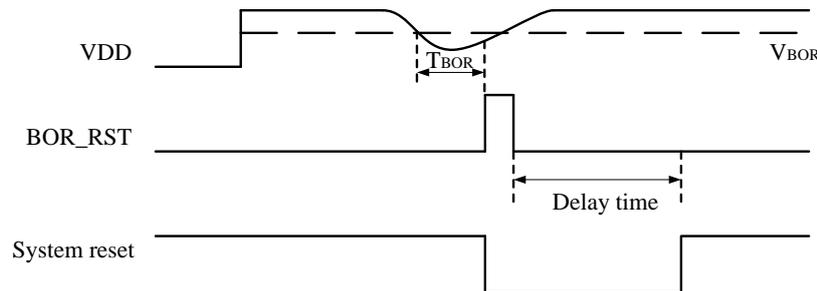
Figure 6 - 1 BOR schematic diagram

6.4 External reset
External $\overline{RST}$ Pin reset is from outside to the $\overline{RST}$ Pin applied a certain width pulse, so as to achieve the microcontroller reset, the Pin can be can be configured as I/O port when it is not used, the function need to be set in the code options.

When it as RST port, after $\overline{RST}$ Pin need be set low level and keep the setting time at least (software configuration), microcontroller will enter the reset state, after set $\overline{RST}$ Pin back to the high level, MCU exit reset state and the user program starts to run from 0000H. EXRSTF bit in RSTFR register is set to 1 when reset, the user can read this flag to determine whether external RST reset is generated or not.

Note: 1. P2.7 ports cannot be used as general I/O when as external reset $\overline{RST}$ port
2. If the external RST function is enabled and external $\overline{RST}$ port is in valid state, the system cannot enter simulation or program mode.

6.5 External port low-voltage detection reset
When external voltage is too low, it cannot guarantee the normal system working. At this time, user can use the external port low voltage detection (PLVD) to reset the microcontroller, external port detection voltage equal 1.2V, the reset function can be disabled. When PLVD reset, PLVRSTF bit in RSTFR register is

set to 1, and user can read the flag to determine whether the external ports low-voltage detection reset or not. In addition, by setting registers user can also implement external port voltage debouncing.

6.6 Software reset

Write corresponding value into IAP_CMDH and IAP_CMDL register as flow, the system will generate software reset, SWRF bit in RSTFR register will be set to 1 after reset, and the user can read the flag to determine whether the software reset or not. Detail operations see FLASH IAP operation chapter.

It is recommended to switch system clock to internal high frequency RC before software reset. Software reset does not switch the system clock, but will reset RC32M_DIV[1:0] bits to 01B in CLKSWR register, and reset CLKDIV register to 08H.

6.7 Watchdog (WDT) reset

In order to prevent system interfered in abnormal circumstances, when MCU program is broken, and the system work in abnormal state for a long time, usually the watchdog will be used, if MCU program is not in operation as required within the stipulated time, the MCU is considered in a unexpected state, the watchdog will force MCU reset, and program will re-run from 0000H.

Note: To generate WDT reset, user must set WDTRST to 1, that is to say WDT reset function enabled, otherwise, even WDT is enabled, and it can only set the overflow flag, but not generate reset.

6.8 Stack overflow reset

When the stack overflows, the system will reset, and set SPOVF overflow flag, it must be cleared by software.

Stack overflow include instack overflow and outstack overflow, instack overflow is the current top of the stack address is 0xFF, and have instack action at this time; outstack overflow is the current top of the stack address equal to the bottom of the stack address setting by user, and have outstack action at this time.

Stack overflow reset is configured by enable registers, when it is enabled, and only stack overflow can reset the system.

6.9 Reset registers
6.9.1 Reset flag register (RSTFR)

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	R/W
POR Reset	1	x	x	x	x	0	x	x
EXRST Reset	u	1	u	u	u	0	u	u
BOR Reset	u	u	1	u	u	0	u	u
WDT Reset	u	u	u	1	u	0	u	u
Soft reset	u	u	u	u	1	0	u	u
Stack overflow reset	u	u	u	u	u	0	1	u
PLVD Reset	u	u	u	u	u		u	1
Flag	PORF	EXRSTF	BORF	WDTRF	SWRF	-	SPOVF	PLVRSTF

Note: x is undefined value, u indicates the value is determined by the value before current reset, it is recommended to clear the registers after POR Reset.

Bit	Flag	Introductions
7	PORF	Power-on reset flag 0 : No power-on reset 1 : Power-on reset generated, software clear 0
6	EXRSTF	External RST reset flag 0 :No external RST reset 1 : External RST reset generated, software clear 0
5	BORF	Under voltage reset flag 0 : No undervoltage reset 1 : Undervoltage reset generated, software clear 0

4	WDTRF	WDT Reset flag 0 : No WDT reset 1 : WDT reset generated, software clear 0
3	SWRF	Software Reset flag 0 : No software reset 1 : Software reset generated , software clear 0
2	-	Reserved
1	SPOVF	Stack overflow flag 0 : No stack overflow reset 1 : Stack overflow reset generated, software clear 0
0	PLVRSTF	External port voltage detection reset flag 0 : External port voltage detection reset 1 : External port voltage detection reset generated, software clear0

6.9.2 BOR voltage detection control register (BORC)

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	R	R	R/W	R/W	R/W
Reset values	1	0	0	0	0	0	0	0
Flag	BOREN	BOR_DBC_EN	-			BORVS[2:0]		

Bit	Flag	Introductions
7	BOREN	BOR enable bit 0 : Disable BOR 1 : Enable BOR
6	BOR_DBC_EN	BOR debouncing enable bit 0 : Disabled 1 : Enabled
5-3	-	Reserved (read = 0b, write invalid)
2-0	BORVS[2:0]	BOR detection of voltage selection bit 000 : 1.8V 001 : 2.0V 010 : 2.4V 011 : 2.6V 100 : 3.0V 101 : 3.6V 110 : 3.9V 111 : 4.2V

6.9.3 BOR voltage detection debouncing control register (BORDBC)

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	BORDBC[7:0]							

Bit	Flag	Introductions
7-0	BORDBC[7:0]	BOR debouncing control bit Debouncing time = BORDBC[7:0] * $8T_{CPU}$ +2 T_{CPU} Note: need to enable BOR_DBC_EN, otherwise BOR no debouncing.

Note: In power-down mode BOR debouncing is turn off automatically, open it automatically when exit power-down mode.

6.9.4 External RST debouncing control register (RSTDBC)

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	1	1	1	1	1	1	1	1
Flag	RSTDBC[7:0]							

Bit	Flag	Introductions
7-0	RSTDBC[7:0]	External RST debouncing control bit debouncing time = RSTDBC[7:0] * $8T_{CPU}$ +2 T_{CPU}

Note: System turns off the external RST debouncing functions automatically in power-down mode, opens automatically after exit the power-down mode.

6.9.5 Stack overflow reset enable registers (SPOV_RSTEN)

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-							SPOV_RSTEN

Bit	Flag	Introductions
7-1	-	Reserved (read = 0b,,write invalid)
0	SPOV_RSTEN	Stack overflow reset enable bit 0 : Disable the stack overflow reset bit 1 : Enable the stack overflow reset bit

7 General and multiplexed I/O

7.1 General and multiplexed I/O characteristics

➢ Provides 18 bi-directional I/O ports
➢ Multiple modes configuration

7.2 I/O mode

All HC89S003F4 I/O ports can be configured into one of many working types by the software, include: input, pull-up input, pull-down input, Schmitt input, analog input, strong push pull output, open drain output, open drain output with pull-up.

After HC89S003F4 the power-on reset, all IO states is set as analog input default.

When HC89S003F4 in input mode (does not include analog input), when execute any read operations, the data sources are from the Pin level. But in output mode, the read data sources distinguished by instructions, "read - modify - write" commands are used to read registers, and other commands is used to read the Pin level.

7.3 I/O function block diagram
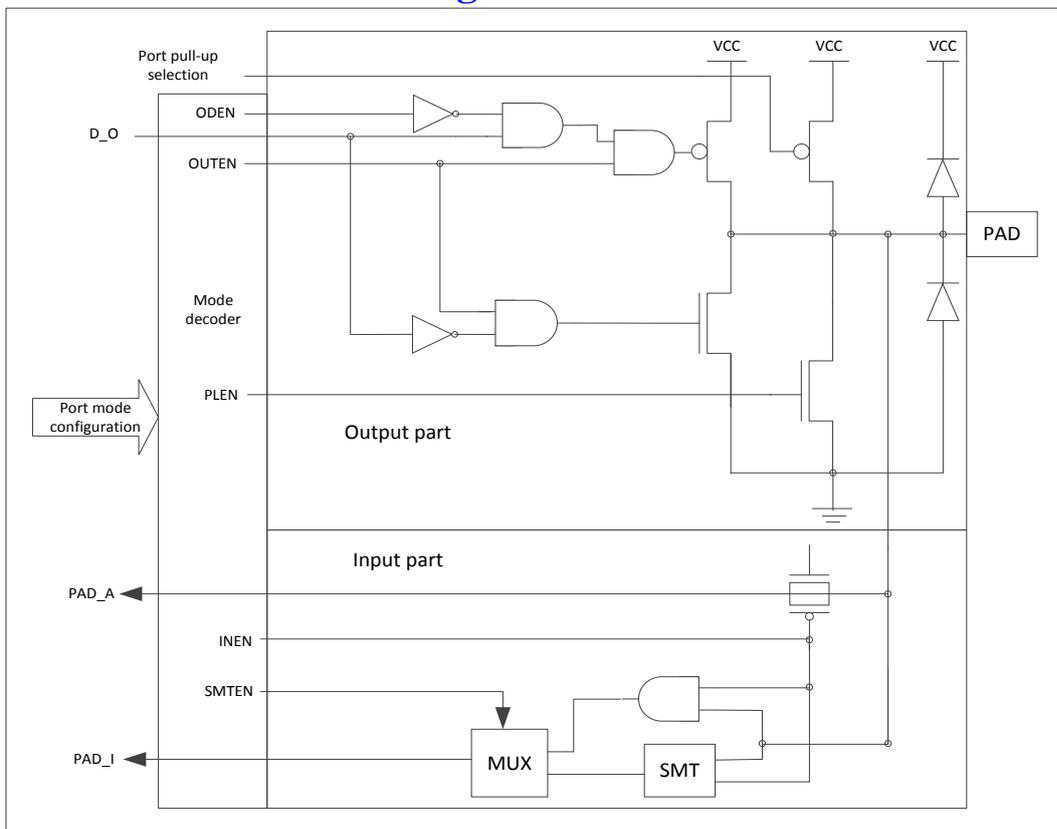
Figure 7 - 1 I/O function block diagram

7.4 I/O port registers

7.4.1 P0 port data register P0

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	P0 [7:0]							

Bit	Flag	Introductions
7-0	P0 [7:0]	P0 port data register

7.4.2 P1 port data register P1

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-	-	-	-	-	-	P1 [1:0]	

Bit	Flag	Introductions
7-0	P1 [1:0]	P1 port data register

7.4.3 P2 port data register P2

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	P2 [7:0]							

Bit	Flag	Introductions
7-0	P2 [7:0]	P2 port data register

7.4.4 P0 port function select register P0M0,P0M1, P0M2, P0M3

P0M0

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	1	1	0	0	1	1
Flag	P01M[3:0]				P00M[3:0]			

P0M1

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	1	1	0	0	1	1
Flag	P03M[3:0]				P02M[3:0]			

P0M2

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	1	1	0	0	1	1
Flag	P05M[3:0]				P04M[3:0]			

P0M3

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	1	1	0	0	1	1
Flag	P07M[3:0]				P06M[3:0]			

Bit	Flag	Introductions
7-4 3-0	P0xM [3:0] (x = 0...7)	P0.x port mode configuration bit 0000 : Input (no SMT) 0001 : Pull-down input (no SMT) 0010 : Pull-up input t (no SMT) 0011 : Analog input 0100 : Input (SMT) 0101 : Pull-down input (SMT) 0110 : Pull-up input (SMT) 0111 : Reserved (analog input) 1x00 : Push-pull output 1x01 : Open drain output 1x10 : open drain output with pull-up 1x11 : Reserved (push-pull output) Note: x is 0 or 1

7.4.5 P1 port function select register P1M0

P1M0

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	1	1	0	0	1	1
Flag	P11M[3:0]				P10M[3:0]			

Bit	Flag	Introductions
7-4 3-0	P1xM [3:0] (x = 0...7)	P1.x port mode configuration bit 0000 : Input (no SMT) 0001 : Pull-down input (no SMT) 0010 : Pull-up input t (no SMT) 0011 : Analog input 0100 : Input (SMT) 0101 : Pull-down input (SMT) 0110 : Pull-up input (SMT) 0111 : Reserved (analog input) 1x00 : Push-pull output 1x01 : Open drain output 1x10 : Open drain output with pull-up 1x11 : Reserved (push-pull output) Note: x is 0 or 1

7.4.6 P2 port function select register P2M0, P2M1, P2M2, P2M3

P2M0

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	1	1	0	0	1	1
Flag	P21M[3:0]				P20M[3:0]			

P2M1

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	1	1	0	0	1	1
Flag	P23M[3:0]				P22M[3:0]			

P2M2

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	1	1	0	0	1	1
Flag	P25M[3:0]				P24M[3:0]			

P2M3

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	1	1	0	0	1	1
Flag	P27M[3:0]				P26M[3:0]			

P2 group port P2.7, P2..5, P2.4, P2.3 support port pull-up and pull-down enable at the same time, output voltage of port is about 1/2VDD. The remained port of P2 group does not support the function described above. Detailed configuration as below:

Bit	Flag	Introductions
7-4 3-0	P2xM [3:0] (x = 0,1,2,6)	P2.x port mode configuration bit 0000 : Input (no SMT) 0001 : Pull-down input (no SMT) 0010 : Pull-up input t (no SMT) 0011 : Analog input 0100 : Input (SMT) 0101 : Pull-down input (SMT) 0110 : Pull-up input (SMT) 0111 : Reserved (analog input) 1x00 : Push-pull output 1x01 : Open drain output 1x10 : Open drain output with pull-up 1x11 : Reserved (push-pull output) Note: x is 0 or 1

Bit	Flag	Introductions
7-4 3-0	P2xM [3:0] (x = 3,4,5,7)	P2.x port mode configuration bit 0000, 0001, 0010 : Input (no SMT) 0011 : Analog input 0100 : Input (SMT) 0101 : Pull-down input (SMT) 0110 : Pull-up input (SMT) 0111: Analog channels, pull-up and pull-down at the same time enable 1x00 : Push-pull output 1101 : Open drain output 1110 : Open drain output with pull-up Other values: the system Reserved, forbid operation Note: x is 0 or 1

7.4.7　Port pull-up resistor selection register

P0LPU

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R/W	R	R	R	R
Reset values	0	0	0	0	0	0	0	0
Flag	-		P02PU[1:0]		-			

Bit	Flag	Introductions
7-6	-	Reserved bit
5-4	P02PU[1:0]	Port pull-up resistance selection bit 00 : 50 KΩ 01 : 100 KΩ 10 : 150 KΩ 11 : 300 KΩ Note: Resistance is the reference value at VDD @5V.
3-0	-	Reserved bit

7.4.8   Ports debouncing control register P00DBC, P01DBC, P02DBC

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	P0xDBCLK[1:0]		P0xDBCT[5:0]					

Bit	Flag	Introductions

7-6	P0xDBCLK [1:0]	Port debouncing clock select
		  00 : F_{osc} /1
		  01 : F_{osc} /4
		  10 : F_{osc} /16
		  11 : F_{osc} /64
		Note: x is 0, 1or 2.

5-0	P0xDBCT [5:0]	Port debouncing count number of clock, when configured as a 00, no debouncing. Time debouncing time is the time need to maintain for the level of its corresponding port when port input, in need of attention, assigned to the function foot of the three Pins, external interrupt input, fault detection Pin is affected by debouncing control, and P02DBC[7:0] is P0.2 debouncing control registers. Note: P0xDBCT [5:0] configuration for debouncing time is a range, scale factor * T_{osc}* P0xDBCT [5:0] - T_{osc} < debouncing time < scale factor * T_{osc}* (P0xDBCT [5:0] +1) - T_{osc}.

7.5 Peripheral function Ports total mapping control

7.5.1   Peripheral function Ports total mapping control registers

Extension SFR Address	Extension SFR	Extension SFR Address	Extension SFR	Extension SFR Address	Extension SFR	Extension SFR Address	Extension SFR
0xFF80	T0_MAP	0xFF90	PWM0_MAP	0xFFA0	TXD_MAP	0xFFB0	-
0xFF81	T1_MAP	0xFF91	PWM01_MAP	0xFFA1	RXD_MAP	0xFFB1	-
0xFF82	-	0xFF92	-	0xFFA2	-	0xFFB2	-
0xFF83	T3_MAP	0xFF93	-	0xFFA3	-	0xFFB3	-
0xFF84	T4_MAP	0xFF94	PWM1_MAP	0xFFA4	$\overline{SS}$_MAP	0xFFB4	-
0xFF85	T5_MAP	0xFF95	PWM11_MAP	0xFFA5	SCK_MAP	0xFFB5	-
0xFF86	-	0xFF96	-	0xFFA6	MOSI_MAP	0xFFB6	-
0xFF87	-	0xFF97	-	0xFFA7	MISO_MAP	0xFFB7	-
0xFF88	-	0xFF98	PWM2_MAP	0xFFA8	TXD2_MAP	0xFFB8	-
0xFF89	-	0xFF99	PWM21_MAP	0xFFA9	RXD2_MAP	0xFFB9	-

0xFF8A	-	0xFF9A	-	0xFFAA	-	0xFFBA	-
0xFF8B	-	0xFF9B	-	0xFFAB	-	0xFFBB	-
0xFF8C	-	0xFF9C	PWM3_MAP	0xFFAC	-	0xFFBC	-
0xFF8D	-	0xFF9D	-	0xFFAD	-	0xFFBD	-
0xFF8E	-	0xFF9E	-	0xFFAE	-	0xFFBE	-
0xFF8F	CLKO_MAP	0xFF9F	-	0xFFAF	-	0xFFBF	-

Note: the above SFR are external extension XSFR, use MOVX to read and write.

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R/W	R	R/W	R/W	R/W
Reset values	0	0	1	1	0	1	1	1
Flag	-	-	FPORT[1:0]		-	FPIN [2:0]		

Bit	Flag	Introductions
7-6	-	Reserved bit
5-4	FPORT[1:0]	Mapping port selection bit 00 : P0 01 : P1 10 : P2
3	-	Reserved bit
2-0	FPIN [2:0]	Mapping port output pin selection bit FPIN [2:0] = x(x = 0... 7), the corresponding port name selected x (x = 0 ... 7)pin

Note: As output function, many to one mapping will be prohibited by system, but as input function, system will enable many to one mapping.

Above registers reset value is 0x37, after reset All IO are GPIO, user must configure above registers before using the peripheral function Pin, otherwise the peripheral functions will not be available.

For example:

UART1 TXD RXD map to P2.1 P2.2 pin, the user must do the following configuration before start to use UART1:

MOV A,   #0x21   //TXD-->P2.1
MOV DPTR, #0XFFA0
MOVX   @DPTR,A
MOV A,   #0x22   //RXD-->P2.2
MOV DPTR, #0XFFA1
MOVX   @DPTR,A

User need to map UART1 TXD RXD to P0.4 P0.5 pin in the next design, the user must do the following configuration:

MOV A,   #0x04   //TXD-->P0.4
MOV DPTR, #0XFFA0
MOVX   @DPTR,A
MOV A,   #0x05   //RXD-->P0.5
MOV DPTR, #0XFFA1
MOVX   @DPTR,A

When more than one outputs are mapped to a port, there can be only one valid output, the following is the default priorities:

Sequence of priority	Multiplexed port function
1	PWM0
2	PWM01
3	PWM1
4	PWM11
5	PWM2
6	PWM21
7	PWM3
8	CLKO
9	T0_OUT
10	T1_OUT
11	T4_OUT
12	TXD
13	RXD
14	SCK
15	MOSI
16	MISO
17	TXD2
18	SCL
19	SDA

For example: CLKO_MAP is configured to 0x01, select P0.1 as CLKO output port, T4_MAP also is configured to 0x01, as the priorities above, the hardware will configure P0.1 as CLKO output port, and T4_MAP configuration is invalid.

When all the port-mapped control registers is not equal 0x01, that is to say all function ports are not select P0.1 as the input/output port, and at this time the port output is first bit of P0 port data register.

Input can be configured as multiple functions from the entry of a PAD Pin, such as:

Configure T0_MAP to 0x23, need select P2.3 as T0 input port, T5_MAP is also configured to 0x23, and the signal into to P2.3 port is valid to T3 and T5 simultaneously..

When configure TXD and RXD to one port, and if the ports are set to output this time, TXD and RXD will connected together internally.

When as input, regardless of any functions of port, read port data register equal read the values on Pin.

8 Interrupt

8.1 Interrupt characteristics

- ➢ 16 interrupt sources
- ➢ 4 level interrupt priorities
- ➢ 16 external interrupts
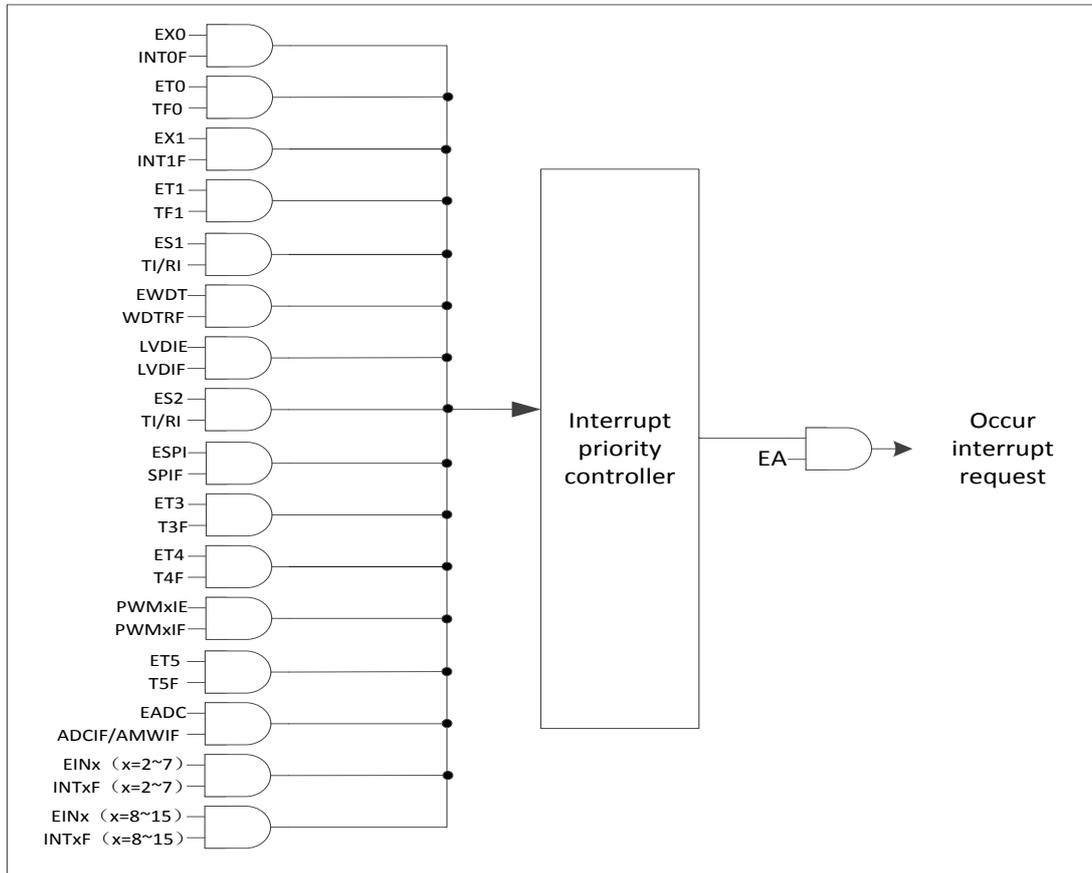
Figure 8 – 1 Interrupt function block diagram

8.2 Interrupt summary

interrupt sources	Vector address	Enable bit	Flag bit	Query priority	interrupt number (C Language)
INT0	0003H	EX0	INT0F	1 (the highest)	0
T0	000BH	ET0	TF0	2	1
INT1	0013H	EX1	INT1F	3	2
T1	001BH	BTV	TF1	4	3
UART1	0023H	ES1	TI/RI	5	4
WDT	002BH	EWDT	WDTRF	6	5
LVD	0033H	LVDIE	LVDF	7	6
UART2	003BH	ES2	TI/RI	8	7
SPI	0043H	ESPI	SPIF/MODF	9	8
T3	004BH	ET3	TF3	10	9
T4	0053H	BTS	TF4	11	10
PWM	005BH	PWMxIE (x = 0...3)	PWMxIF (x = 0...3)	12	11
T5	0063H	ET5	TF5	13	12
ADC	006BH	EADC	ADCIF/ AMWIF	14	13

| INT2-INT7 | 0073H | EINx (x =2...7) | INTxF (x = 2...7) | 15 | 14 |
| INT8-INT15 | 007BH | EINx (x=8...15) | INTxF (x = 8...15) | 16 | 15 |

Note: except the enable and flags bit above have been set, to respond to interrupts the interrupt switch bit EA is enabled, otherwise does not respond to any interrupt.

8.3 Interrupt vectors

When an interrupt occur, data in program counter is push to stack, the corresponding interrupt vector addresses are loaded in program counter. Entrance of the interrupt vector interrupt is described in interrupt summary chapter.

8.4 Interrupt priorities

Each interrupt source can be individually set to one of the 4 interrupt priorities, through the corresponding bit in IP0, IP1, IP2, IP3 to implementation. Interrupt priority service program description as below:

When system respond to an interrupt service program, can respond to higher-priority interrupts, but cannot respond another interrupt with same or low priority.

When system respond to the highest level interrupt service program, do not respond to any other interrupts. If different priorities interrupt sources in apply for interrupt at the same time, system will respond to higher priority interrupt request.

If the same priority interrupt sources in apply for interrupt at the beginning of instruction cycle, the internal query priority decide the interrupt response sequence. Query priority detailed reference to interrupt summary.

interrupt priority	
Priority control (X for the function module) Px[1:0]	Priority
00	Priority 0 (lowest)
01	Priority 1
10	Priority 2
11	Priority 3 (highest)

8.5 Interrupt handling

When an interrupt is generated and the CPU in response, the main program is interrupted, then execute the following operations:

1. Completion of all of the instructions currently being executed;
2. The PC is pushed into the stack;
3. Scene protection;
4. Prevent other interrupts of the same level;
5. The interrupt vector address is loaded into the program counter PC;
6. Execute the corresponding interrupt service program.

Interrupt service program ended by RETI (return from interrupt) instruction, PC value pop from the stack, and restore the original interrupt setting.

When an interrupt is responded, the value loaded into the program counter PC known as an interrupt vector, it is correspond to the starting address of the interrupt service program of the interrupt source. The entry address of the interrupt service program (interrupt vector) detail information, user can refer the interrupt summary.

Since the entry address of interrupt vector is located in the start of the program memory, so the main program first instruction is the jump instruction usually, over the interrupt vector area (LJMP MAIN).

Need to take attention, user can't use RET instead of RETI instruction, RET instruction can also control PC go back to where the original interrupt, but RET Instruction has not the function to clear interrupt priority level trigger, interrupt control system deem the interrupt is still in progress, the consequence is the same level or low-level interrupt request is not be responded.

If user executes the operation of push stack in the interrupt service program, the corresponding pop stack operation should be executed before RETI instruction, that is to say in the interrupt service program PUSH and POP Instruction must be used in pairs, otherwise the system cannot be returned correctly.

8.6 External interrupt

HC89S003F4 have 4 -external interrupt vector entrances, external interrupts 0 ~ 1 has a separate entrance to the interrupt vector respectively, and external interrupts 2 ~ 7 share a common interrupt vector entrance, external interrupts 8 ~ 15 share a common interrupt vector entrance, thus the total have 16 external interrupt inputs, all interrupts can be set 4 trigger modes, namely the rising edge, falling edge, double edge and low level.

When user call the interrupt service program, external interrupts 0 ~ 1 will be cleared to 0 by hardware automatically, external interrupt 2 ~ 15 flags must be cleared by software. If interrupt service is completed and an external interrupt is still maintained, the next interrupt will be generated.

The function of external interrupt 0~2 is on port P0.0~P0.2 respectively, when in the use of an external interrupt 0~2, the user can set external interrupts 0~2 to generate valid interrupt request that need the debouncing time, user can set debouncing time of P0.0~P0.2 port, detail description of ports debouncing control registers P00DBC,P01DBC, P02DBC.

8.7 Interrupt registers
8.7.1 Interrupt enable register IE,IE1
IE

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	EA	ES2	EWDT	ES1	BTV	EX1	ET0	EX0

Bit	Flag	Introductions
7	EA	CPU total interruption enable control bit 0 : Disable CPU interrupt 1 : Enable CPU interrupt
6	ES2	UART2 interrupt enable bit 0 : Disable UART2 interrupt 1 : Enable UART2 interrupt
5	EWDT	WDT interrupt enable bit 0 : Disable WDT interrupt 1 : Enable WDT interrupt
4	ES1	UART1 interrupt enable bit 0 : Disable UART1 interrupt 1 : Enable UART1 interrupt
3	BTV	T1 interrupt enable bit 0 : Disable T1 interrupt 1 : Enable T1 interrupt
2	EX1	Interrupt enable bit of external interrupt 1 0 : Disable INT1 interrupt 1 : Enable INT1 interrupt
1	ET0	T0 interrupt enable bit 0 : Disable T0 interrupt 1 : Enable T0 interrupt
0	EX0	Interrupt enable bit of external interrupt 0 0 : Disable INT0 interrupt 1 : Enable INT0 interrupt

IE1

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	EX8_15	EX2_7	EADC	ET5	-	BTS	ET3	ESPI

Bit	Flag	Introductions
7	EX8_15	External interrupt 8~15 interrupts enable bit 　0 : Disable INT8~INT15 interrupts 　1 : Enable INT8~INT15 interrupts Note: INT8~INT15 share the same interrupt vector.
6	EX2_7	External interrupt 2~7 interrupt enable bit 　0 : Disable INT2~INT7 interrupts 　1 : Enable INT2~INT7 interrupts Note: INT2~INT7 share the same interrupt vector.
5	EADC	A/D Conversion complete interrupt enable bit 　0 : Disable A/D interrupt 　1 : Enable A/D interrupt
4	ET5	T5 interrupt enable bit 　0 : Disable T5 interrupt 　1 : Enable T5 interrupt
3	-	Reserved bit
2	BTS	T4 interrupt enable bit 　0 : Disable T4 interrupt 　1 : Enable T4 interrupt
1	ET3	T3 interrupt enable bit 　0 : Disable T3 interrupt 　1 : Enable T3 interrupt
0	ESPI	SPI interrupt enable bit 　0 : Disable SPI interrupt 　1 : Enable SPI interrupt

8.7.2 Interrupt priority selection register IP0,IP1, IP2,IP3

IP0

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PT1 [1:0]		PX 1[1:0]		PT0 [1:0]		PX0 [1:0]	

Bit	Flag	Introductions
7-6	PT1 [1:0]	T1 interrupt priority control bits
5-4	PX 1[1:0]	INT1 interrupt priority control bits
3-2	PT0 [1:0]	T0 interrupt priority control bits
1-0	PX0 [1:0]	INT0 interrupt priority control bits

IP1

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PS2 [1:0]		PLVD [1:0]		PWDT [1:0]		PS1 [1:0]	

Bit	Flag	Introductions
7-6	PS2 [1:0]	UART2 interrupt priority control bits
5-4	PLVD [1:0]	LVD interrupt priority control bits
3-2	PWDT [1:0]	WDT interrupt priority control bits
1-0	PS1 [1:0]	UART1 interrupt priority control bits

IP2

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PPWM [1:0]		PT4 [1:0]		PT3 [1:0]		PSPI [1:0]	

Bit	Flag	Introductions
7-6	PPWM [1:0]	PWM interrupt priority control bits
5-4	PT4 [1:0]	T4 interrupt priority control bits
3-2	PT3 [1:0]	T3 interrupt priority control bits
1-0	PSPI [1:0]	SPI interrupt priority control bits

IP3

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PX8_15 [1:0]		PX2_7 [1:0]		PADC [1:0]		PT5 [1:0]	

Bit	Flag	Introductions
7-6	PX8_15 [1:0]	INT8_15 interrupt priority control bits
5-4	PX2_7 [1:0]	INT2_7 interrupt priority control bits
3-2	PADC [1:0]	ADC interrupt priority control bits
1-0	PT5 [1:0]	T5 interrupt priority control bits

interrupt priority	
Priority control (x for the function module) Px[1:0]	Priority
00	Priority 0 (lowest)
01	Priority 1
10	Priority 2
11	Priority 3 (highest)

8.7.3 External interrupt level selection registers PITS0,PITS1, PITS2, PITS3

PITS0

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	IT3[1:0]		IT2[1:0]		IT1[1:0]		IT0[1:0]	

PITS1

Bit	7	6	5	4	3	2	1	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

Bit	7	6	5	4	3	2	1	0
values								
Flag	IT7[1:0]		IT6[1:0]		IT5[1:0]		IT4[1:0]	

PITS2

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	IT11[1:0]		IT10[1:0]		IT9[1:0]		IT8[1:0]	

PITS3

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	IT15[1:0]		IT14[1:0]		IT13[1:0]		IT12[1:0]	

Bit	Flag	Introductions
7-6 5-4 3-2 1-0	ITx[1:0] (x = 0... 7)	External interrupt trigger edge selection bits 00 : Low level interrupts 01 : Falling edge interrupts 10 : Rising edge interrupts 11 : Double edge interrupts

8.7.4　External interrupt 2-15 enable control register PINTE0, PINTE1

PINTE0

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Reset values	0	0	0	0	0	0	0	0
Flag	EINT7	EINT6	EINT5	EINT4	EINT3	EINT2	-	

Bit	Flag	Introductions
7-2	EINTx (x =2...7)	External interrupt control bits (INT2~INT7) 0 : Disable the port interrupts 1 : Enable the port interrupts Note: As long as the corresponding EINTx (x =2...7) bits are enabled, the corresponding interrupt flags can be set to 1, otherwise, corresponding flags will not be set to 1.
1-0	-	Reserved (read = 0b,, write invalid)

PINTE1

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	EINT15	EINT14	EINT13	EINT12	EINT11	EINT10	EINT9	EINT8

Bit	Flag	Introductions
7-0	EINTx (x =8...15)	External interrupt control bits (INT8~INT15) 0 : Disable the port interrupts 1 : Enable the port interrupts Note: 1. As long as the corresponding EINTx (x =8...15) bits are enabled, the

corresponding interrupt flags can be set to 1, otherwise, corresponding flags will not be set to 1.

2 .When external interrupt use query mode, the interrupt flag cannot be cleared normally. User can disable interrupt enable bit and then clear the interrupt flag, upon completion of the interrupt flag is cleared, and then enable external interrupts to generate normal external interrupt queries. When using the interrupt mode, no problem.

8.7.5 External interrupt flag register PINTF0, PINTF1

PINTF0

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	INT7F	INT6F	INT5F	INT4F	INT3F	INT2F	INT1F	INT0F

Bit	Flag	Introductions
7-2	INTxF (x =2...7)	INT2-INT7 interrupt request flags 0: Software clear 0 1: When external interrupts occur, hardware set 1
1-0	INTxF (x = 0,1)	INT0 INT1 interrupt request flags 0 : When interrupt responded, hardware clear 0 automatically or software clear 0 1 : When external interrupts occur, hardware set 1

PINTF1

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	INT15F	INT14F	INT13F	INT12F	INT11F	INT10F	INT9F	INT8F

Bit	Flag	Introductions
7-0	INTxF (x =8...15)	INT8-INT15 interrupt request flag bits 0：Software clear 0 1 : When external interrupts occur, hardware set 1

8.7.6 External interrupt 01 Pin selection register INT01_PINS

INT01_PINS

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-						INT1_PINS	INT0_PINS

Bit	Flag	Introductions
7-2	-	Reserved
1	INT1_PINS	INT1 Pin selection bit 0 : P0.1 1 : P1.1
0	INT0_PINS	INT0 Pin selection bit 0 : P0.0 1 : P1.0

9 Timer/Counter

9.1 Timer/Counter characteristics

➢ Timer/Counter T0&T1 is not fully compatible with standard 8051, the difference mainly of function definition in the mode0.

➢ Timer/Counter T0&T1 support 16 bit automatic reload

9.2 Timer/Counter Tx(x = 0,1)

9.2.1 Timer/Counter Tx(x = 0,1) work mode

Two data of each Timer register (THx & TLx (x = 0, 1)) can be used as a 16 bit register to access, they are controlled by the register TCON TMOD. IE registers ET0 ET1 bits will enable Timer0 and Timer1 interrupt (See interrupt section chapter).

Select the Counter/Timer operation mode by selecting Mx[1:0] bit in counter/Timer mode register (TMOD).

Mx[1:0]	mode	Description
00	mode 0	16bit auto reload Timer/Counter
01	mode1	16 -bit Timer/Counter
10	mode2	8 Automatically reload Timer/Counter
11	mode 3	T0 divided into two (TL0/TH0) independent 8 -bit Timer/Counter (T1 no the mode)

9.2.1.1 Mode0: 16 bit auto reload Timer/Counter
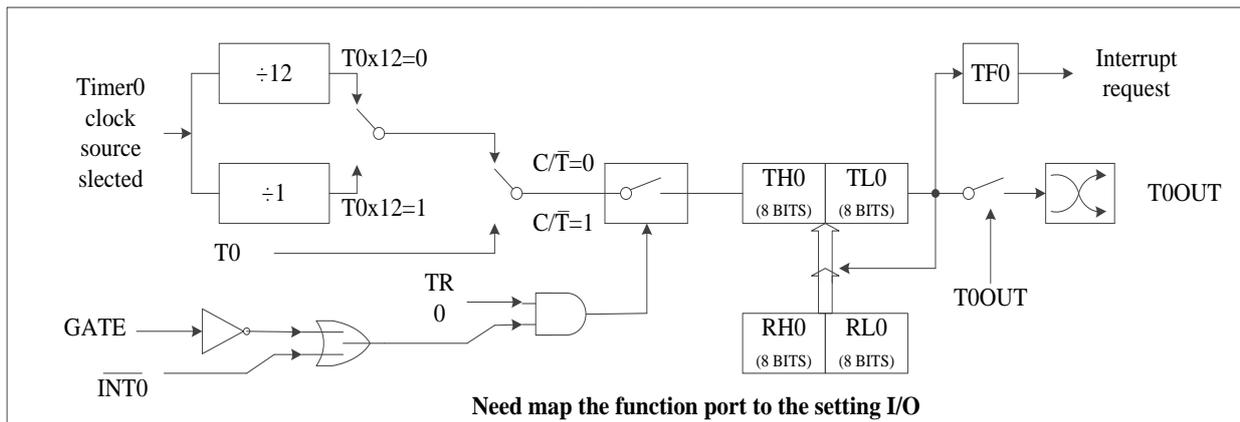
Figure 9 - 1 Timer0 mode0 function block diagram

Mode0 is not compatible with the standard 8051, it is a 16 bit automatically reload Timer/Counter, when THx & TLx(x = 0, 1) was written, it is used as a Timer reload register, when be read, it is used as count register. When TRx (x = 0, 1) value is 0, write THx & TLx (x = 0, 1) two registers sequentially, the write value is written into the reloaded register and count register at the same time, when TRx (x = 0, 1) is set to 1, the count register value increments the count from the written data, after count to 0xFFFF, count counter will overflow after an additional clock, TFx (x = 0, 1) is set to 1, while 16 Data of reloaded registers is automatically reloaded into the counter register, counter starts to increment the count from the reload value.

When TRx(x = 0,1) is 1, THx & TLx(x = 0,1) write operations will not affect the value of the counter, can only change the value for reload registers, this changed value is reloaded into the count register after the next overflow. Only when TRx (x = 0, 1) is 0, write operation of THx & TLx (x = 0, 1) will also change the count register and reload register value at the same time.

Because of the TLx (x = 0,1) THx (x = 0,1) write operation require 2 instructions to complete, in order to ensure the accurate count, THx(x = 0,1) TLx(x = 0,1) register write operation with the TLx(x = 0,1) register write operations as a baseline. When user write reloaded registers, write THx (x = 0, 1) register does not valid immediately, but store in a buffer register temporarily, only the TLx (x = 0, 1) register write operations will enable THx(x = 0, 1) and TLx(x = 0, 1) registers at the same time.

Therefore, THx (x = 0, 1) TLx (x = 0, 1) read and write operations flow the following sequence:

Write: high bit first then low

Read: high bit first then low

User need to take attention is during a write operation, when TRx(x = 0,1) is 0, start with high bit then low, reload data will directly reloaded to the counter register, when TRx(x = 0,1) is 1, start with high bit then low, reload data only will be reloaded to the count register in the next overflow . If user write low bit then high, high data is invalid (invalid: indicates that the data cannot be updated when an reload occur), until the next write operation to the low data, previously written high data to be valid (valid indicates reload data can be updated when an reload occur). If only write low bit, low data will also be available, for example, when T0 is executed as the following operation:

        (1) TH0 = 0x05;

        (2) TL0 = 0x08;   // In case of reload, reload to the counter data is 0x0508

        (3) TH0 = 0x06;   // In case of reload, reload to the counter data is still 0x0508

        (4) TL0 = 0x08;   // In case of reload, reload to the counter data is 0x0608

        (5) TL0 = 0x09;   // In case of reload, reload to the counter data is 0x0609

Apparently as long as modifying data reload, low bit has to be written once again, it is recommended they are modified at the same time every time.

Note: mode1, 2, 3 no this requirement.

9.2.1.2 Mode1: 16 bit Timer/Counter
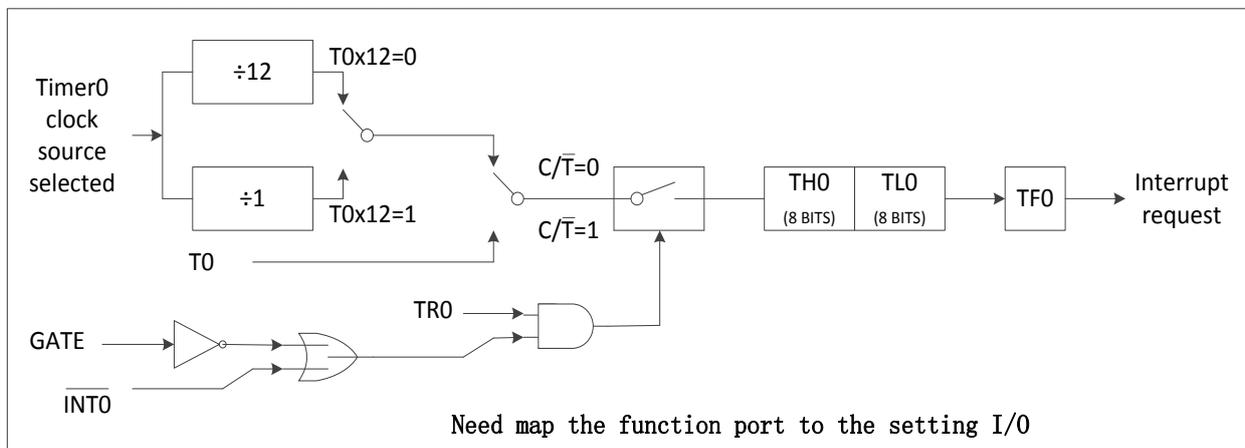
Figure 9 - 2 Timer0 1 mode1 function block diagram

In mode1, the Timer Tx (x = 0, 1) is 16 -bit counter/Timer. THx(x = 0, 1) register store high 8 bits data of 16 -bit counter/Timer, TLx (X = 0, 1) store low 8 bits. When 16 -bit Timer register increments to overflow, the system set Timer overflow flag TFx(x = 0, 1). If Timer x interrupt is enabled, it will generate an interrupt.

C/$\overline{T}$x (x = 0,1) bit select Counter/Timer function, if C/$\overline{T}$x (X = 0, 1) = 1,that will work in the external counter mode, when an external count clock falling edge occur, the Timer Tx data register will increment 1 . If c/$\overline{T}$x (x = 0, 1) = 0, select the system clock as the clock source of Timer Tx (x = 0, 1).

When GATEx (x = 0, 1) = 0, TRx set 1, open the Timer.

When GATEx (x = 0, 1) = 1, only when the external input signal INTx (x = 0, 1) is high level, TRx (x = 0,1) will be set to 1, the Timer Tx will count, which can be measured positive pulse width of INTx (x = 0,1). TRx(x = 0, 1) bit set 1 does not forcibly reset Timer, this means if TRx is set to 1, the Timer registers start to count from the value of TRx(x = 0, 1) is cleared to 0 last time. So before enable Timer, user should set the initial value of Timer registers.

9.2.1.3 Mode2: 8 bit auto reload Timer/Counter
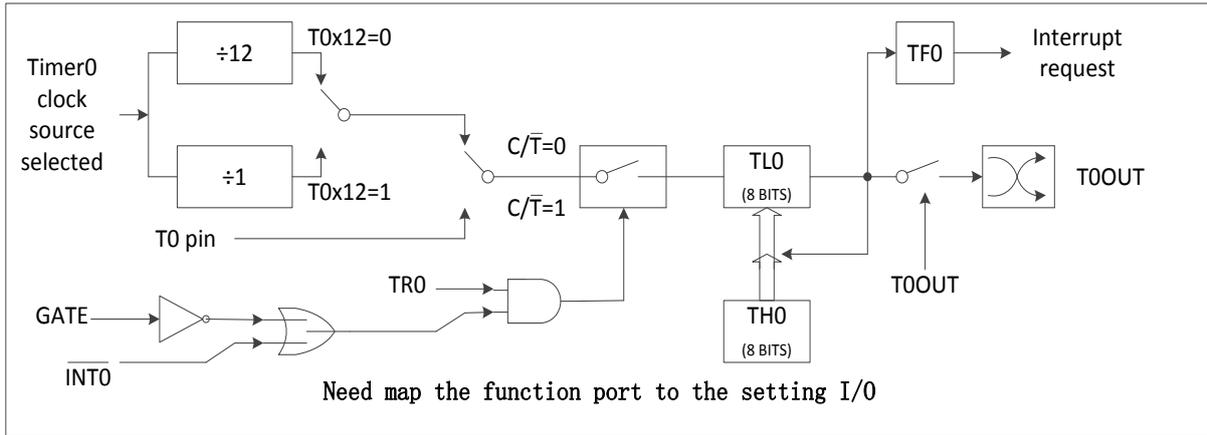
Figure 9 - 3 Timer0 mode2 Function block diagram

In mode2, the Timer Tx(x = 0, 1) is 8 bit auto reload counter/Timer. TLx (x = 0, 1) store the count value, THx (x = 0, 1) store the reload value. When TLx (x = 0, 1) counter increments to 0x00, Timer overflow flag TFx (x = 0, 1) is set, value in register THx (x = 0, 1) is reloaded into register TLx (x = 0, 1). If the Timer interrupt enabled, when TFx (x = 0, 1) bits are set to 1, an interrupt will generated, but the reload value in THx (x = 0, 1) do not change. Before enable Timer start counting, TLx (x = 0, 1) must be initialized to the value that user want.

In addition to auto reload function, in mode2, enable and configure to the counter/Timer mode0 is consistent with mode1. Configure TxX12 (x = 0, 1) bits in register TCON2 to select system clock or 1/12 system clock as clock source of Timer Tx (x = 0, 1).

When used as a Timer application, configure TxOUT[1:0](x = 0,1) bits in register TCON1 Tx(x = 0,1) to enable Timer Tx(x = 0,1) overflow, Pin of Timer Tx(x = 0,1) flip automatically.

9.2.1.4 Mode3: Two 8 bit Timer/Counter (T1 no this mode)
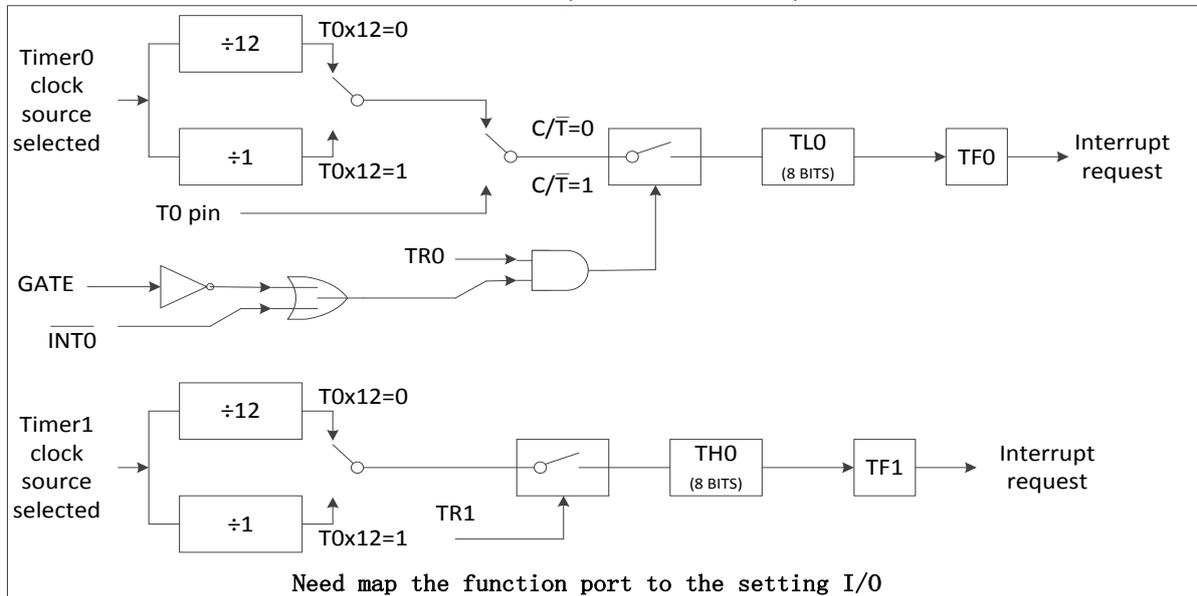
Figure 9 - 4 Timer0 3 function block diagram

In mode3 the Timer T0 as two independent 8 -bit counter/Timers, it is controlled by TL0 TH0 respectively. TL0 using Timer0 control (in TCON) and state (in TMOD) bits TR0, C/$\overline{\text{T}}$0, GATE0 and TF0. TL0 can use system clock or external input signals as clock source.

TH0 can only be used as a Timer function, clock source is from the system clock. TH0 is controlled enable by Timer T1 control bit TR1, Timer T1 overflow flag TF1 is set to 1 when overflow, and control Timer T1 interrupt.

When Timer0 work in mode3, Timer1 can work in mode 0/1/2, but can't set TF1 and generate interrupt.

It can be used to generate the baud rate of serial port. TH1 and TL1 can only be used as a Timer, clock source from the system clock, and GATE1 bit is invalid. The pull-up resistance on T1 input pin is invalid. Timer1 is controlled enable or not by mode, because TR1 is occupied by Timer0. Timer1 is enabled in mode0/1/2, and is closed in mode3.

Configure TxX12(x = 0, 1) bits in register TCON1 to select the system clock or 1/12 of system clock as clock source of Timer Tx(x = 0, 1).

9.2.2 Timer/Counter Tx(x = 0,1) registers
9.2.2.1 Timer Tx(x = 0,1) control register TCON, TCON1
TCON

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R
Reset values	0	0	0	0	0	0	0	0
Flag	TF1	TR1	TF0	TR0	-			

Bit	Flag	Introductions
7, 5	TFx (x = 0,1)	Tx(x = 0,1) overflow flag 0 : Hardware clear 0 automatically when interrupt response, or software clear 0 1 : Hardware set 1 when Counter overflow
6, 4	TRx (x = 0,1)	Tx(x = 0,1) operation control bit 0 : Stop Tx 1 : Start Tx
3-0	-	Reserved bit

TCON1

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R/W	R	R	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-		T1OUT	T1X12	-		T0OUT	T0X12

Bit	Flag	Introductions
7, 6, 3, 2	-	Reserved (read = 0b,, write invalid)
5, 1	TxOUT (x = 0,1)	Tx(x = 0,1) comparison output enable bits 0 : Disable Timer Tx comparison output function 1 : Enable Timer Tx comparison output function

4, 0	TxX12 (x = 0,1)	Tx(x = 0,1) Timer system clock scale frequency selection bits
		  0 : Tx Timer clock $F_{osc}/12$
		  1 : Tx Timer clock F_{osc}

9.2.2.2 Timer Tx(x = 0, 1) mode register TMOD

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	GATE1	C/$\overline{T}$1	M1[1:0]		GATE0	C/$\overline{T}$0	M0[1:0]	

Bit	Flag	Introductions
7, 3	GATEx (x = 0,1)	Tx(x = 0,1) door control bit 0 : Just need a software set TRx can start the Tx 1 : Only set TRx 1 when the INTx port is high level, Tx can work
6, 2	C/$\overline{T}x$ (x = 0,1)	Tx(x = 0,1) Timer/Count function selection bits 0 : Tx for internal Timer

		1 : Tx for external count
5-4 1-0	Mx[1:0] (x = 0,1)	Tx(x = 0,1) mode selection bits 00 : 0 16 -bit reload Timer/Counter 01 : 1 16 -bit Timer/Counter 10 : 2 8 auto reload initial value Timer /Counter 11 : 3 T0 divided into two (TL0/TH0) independent 8 -bit Timer/Counter; T1 stop count Note: T0 occupied the bits TR1/TF1 of T1 and interrupt source of in mode 3, because TR1 is occupied by T0, and needs to close T1 at this time, and user can set T1 to mode3.

9.2.2.3 Timer Tx(x = 0, 1) Data register TLx (x = 0, 1), THx (x = 0, 1)

TLx (x = 0, 1)

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	TLx[7:0] (x = 0,1)							

Bit	Flag	Introductions
7-0	TLx[7:0] (x = 0,1)	Tx(x = 0,1) low bytes of data register

THx (x = 0,1)

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	THx[7:0] (x = 0,1)							

Bit	Flag	Introductions
7-0	THx[7:0] (x = 0,1)	Tx(x = 0,1) high bytes of data register

9.3 Timer 3

Timer 3 is 16bit auto reload Timer, using two data register TH3 TL3 to access, and controlled by T3CON register. Set bit ET3 to 1 in IE1 registers enables Timer 3 interrupt (see interrupt chapter).

Timer 3 only has one mode: 16bit auto reload counter/Timer, user can set the prescaler ratio, and work in power-down mode.

Timer 3 has a 16 -bit counter/Timer register (TH3, TL3). When TH3 and TL3 are written, is used a reload Timer register; when are read, is used as a counter register. TR3 set 1 then Timer 3 start to increment count, from 0xFFFF to 0x0000 an overflow occurred, overflow will set the TF3 bit, and 16 –bit data in reload register is reloaded to count register at the same time.

When TR3 value is 1, write operation of TH3/TL3 will not affect the value of counter, and only change the reloaded register value, the changed value will be reloaded into count register at next overflow time. Only when TR3 is 0, write operation of TH3/TL3 will change the value of count and reload register synchronously.

Read or write operation of TH3 TL3 follow the following order: high bit first then low.

If T3CLKS[1:0] value is 00, the Timer 3 cannot work in power-down mode. If T3CLKS[1:0] value is 01,T3 port input an external clock, Timer 3 can work in normal mode or power-down mode (When system enter the power-down mode from high frequency mode (internal high-frequency RC)).

Note: When reading TH3 TL3, make sure TR3 = 0. (When TR3=1, because T3 is counting, read out TH3 and TL3 value is inaccurate).

Need map the function port to the setting I/O
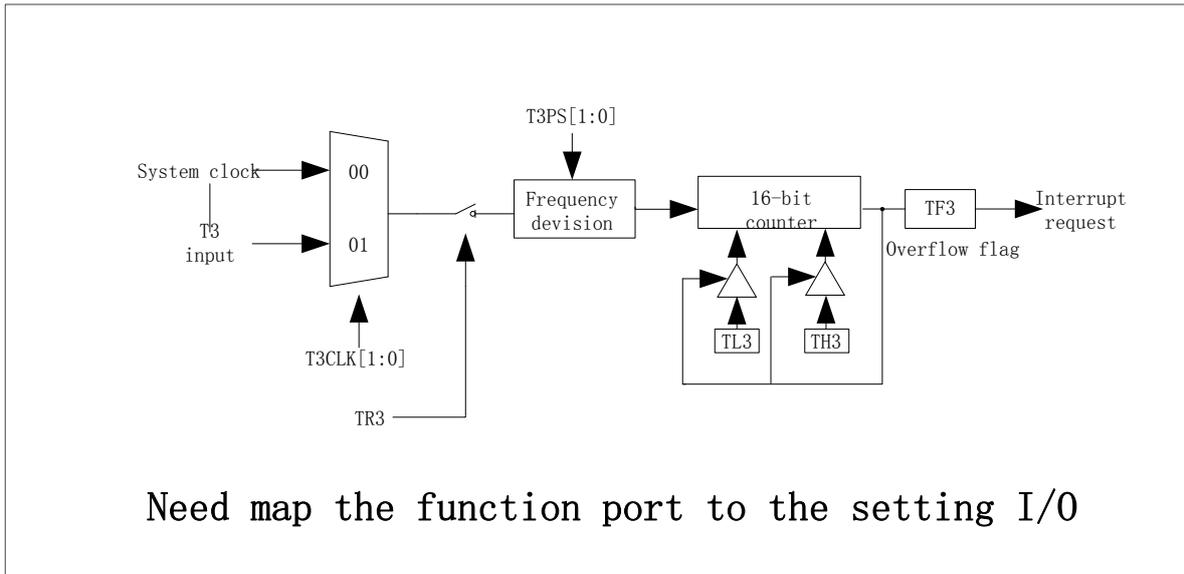
Figure 9 - 5 Timer3 function block diagram

9.3.1 Timer/Counter T3 registers

9.3.1.1 Timer T3 control register T3CON

T3CON

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	TF3	T3PD_EN	T3PS[1:0]		-	TR3	T3CLKS [1:0]	

Bit	Flag	Introductions
7	TF3	Timer 3 overflow flag 0 : Without overflow (hardware clear 0),software can also clear 0 1 : Overflow (hardware set 1)
6	T3PD_EN	Timer 3 Operation control bit in power-down mode 0 : Disable Timer power-down mode 3 1 : Enable Timer power-down mode 3, at this time T3CLKS [1:0] is 01
5-4	T3PS[1:0]	Timer 3 prescaler ratio selection bit 00 : 1/1 01 : 1/8 10 : 1/64 11 : 1/256
3	-	Reserved bit
2	TR3	Timer 3 control enable bit 0 : Disable Timer 3 1 : Enable Timer 3
1-0	T3CLKS [1:0]	Timer 3 count clock source selection bits 00 : The system clock F_{osc} 01 : T3 input an external clock 10 : Reserved 11 : Reserved (system clock F_{osc}) Note: when T3 Count clock source select the port input an external clock, TR3 and T3CLKS requires are configured at the same time, that is to say with one instruction to complete the configuration.

9.3.1.2 Timer T3 Data register TL3, TH3

TL3

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	TL3							

Bit	Flag	Introductions
7-0	TL3	T3 low bytes of data registers

TH3

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	TH3							

Bit	Flag	Introductions
7-0	TH3	T3 high bytes of data register

9.4 Timer/Counter 4

Timer 4 is 16 bit auto reload Timer. Two data registers TH4 and TL4 as a 16 -bit register to access, is controlled by T4CON register. Set IE1 register ET4 bit to 1 enables Timer 4 interrupt (see interrupt chapter).

When TH4 and TL4 are written, it is used as reload Timer register, when read, is used as count register. TR4 set 1 then Timer 4 start to increment count, from 0xFFFF to 0x0000 an overflow occurred, overflow will set the TF4 bit, and 16 –bit data in reload register is reloaded to count register at the same time.

TH4 TL4 Read or write operation follow the following order: high first then low.

9.4.1 Timer/Counter T4 work mode

Timer 4 there are three kinds of work **mode**: 16-bit auto reload Timer, serial port baud rate generator, and T4 edge trigger 16-bit auto reload Timer. The modes selection by T4CON register T4M[1:0] setting.

9.4.1.1 Mode0: 16 bit auto reload Timer/Counter

Timer 4 is 16 bit auto reload Timer. TH4 register store 16 -bit Timer high 8 bits,TL4 store low 8 bits. When TR4 = 0, write TH4 and TL4 two registers sequentially, write the value is written into reload and count registers. TR4 set 1, the count register value increments the count from the written data, after count to 0xFFFF, count counter will overflow after an additional clock, TF4 is set to 1, the same time 16 bits data of reloaded register is reloaded into count register automatically, counter starts to increment the count from the reload value. The interrupt will be generated If Timer 4 interrupt enabled.

When TR4 value is 1, write operation of TH4/TL4 will not affect the value of counter, and only change the reloaded register value, the changed value will be reloaded into count register at next overflow time. Only when TR4 is 0, write operation of TH4/TL4 will change the value of count and reload register synchronously.

T4CON.0 register T4CLKS bit select clock source. When T4CLKS = 1, Timer 4 clock source is external clock, after prescaler, counter data register increment. When T4CLKS = 0, clock source of Timer 4 is the system clock.

In comparison mode, T4 port is need be set output by software. Timer 4 counts to 0xFFFF from the default value in TH4 and TL4, when counter overflows, T4 port output level flips, and Timer 4 interrupt flag is set to 1. In comparison mode, Timer 4 has to work in Timer mode (T4CLKS = 0).
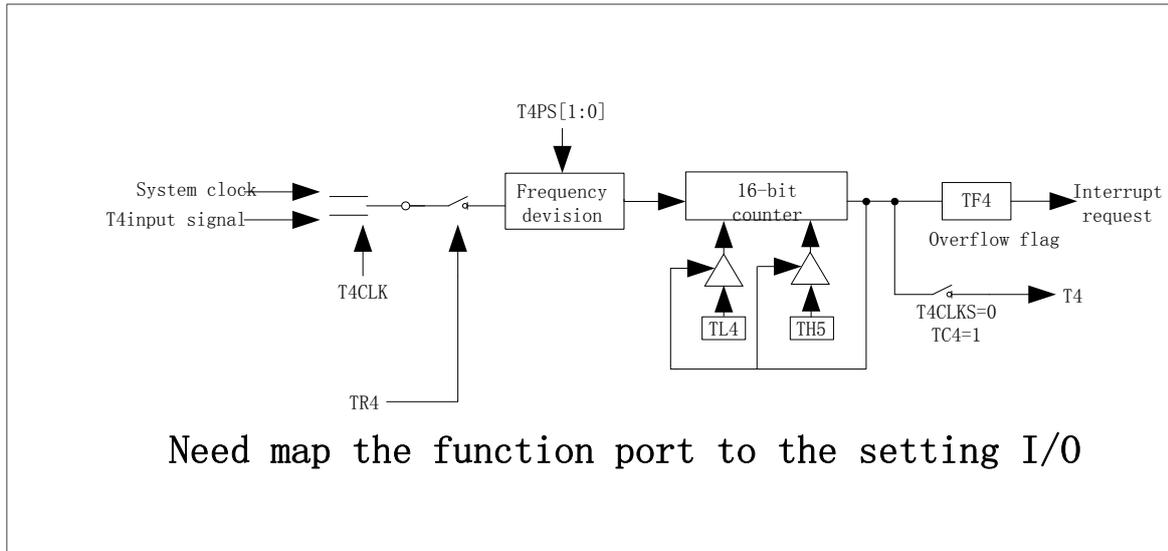
Figure 9 - 6 TIMER4 0 functional block diagram

9.4.1.2 Mode1: UART1 baud rate generator

By setting the T4CON register T4M[1:0] = 01 to select the Timer 4 as the baud rate generator. The mode is similar to auto reload mode. Overflow of Timer 4 will cause the 16 -bit value in software Timer 4 reloads register load to Timer 4 counter, and generate overflow interrupt at the same time. If user does not want to generate interrupt, he can close ET4. UART1 baud rate in mode1 and 3 is calculated by the following formula:

$$\text{BaudRate} = \frac{1}{16} \times \frac{f_{T4}/\text{PRESCALER}}{65536-[\text{TH4,TL4}]}$$, use Timer 4 as the baud rate generator.

In the formula, f_{T4} is Timer 4 Count selected clock source, PRESCALER Timer 4 prescaler ratio, TH4 and TL4 are Timer 4 data register.

When TC4=1, user can set the baud rate output from T4, frequency of the output waveform is 1/2 of baud rate, at this time Timer 4 must work in Timer mode(T4CLKS = 0).

9.4.1.3 Mode2/3: with edge-trigger 16 bit auto reload Timer

Timer 4 2/3 are 16 bit auto reload Timer. T4CLKS bit in T4CON. 0 registers keep 0 always, user can only select the system clock as clock source of Timer 4, and the other settings are same as mode 0.

In mode2, when TR4 is set to 1, Timer 4 wait for the trigger signal on T4 port (control rising/falling edge by T4M[1:0]), a valid trigger signal starts Timer 4. When the Timer 4 overflows from 0xFFFF to 0x0000, TF4 (T4CON.7) will be set, If Timer 4 interrupt enabled, Timer 4 interrupt will be generated. With overflow, 16 bits data in Timer reload register is reloaded into the count registers TH4 and TL4, Timer 4 maintains the state and wait for the next trigger.

  If TC4=0, when Timer 4 is counting, a trigger signal will not stop counter for counting, the counter will reload after overflow and maintain the state, and waiting for the next valid trigger signal;

If TC4=1, when Timer 4 is counting, a trigger signal will cause 16 bits data in reload register is reloaded into the count registers TH4 and TL4, and begin to count, but it will not generate an interrupt, interrupt occur only after the counter overflow.

TR4 set 1 don't clear Timer 4 counter, before enable the Timer, user write an initial value expected to reload register.
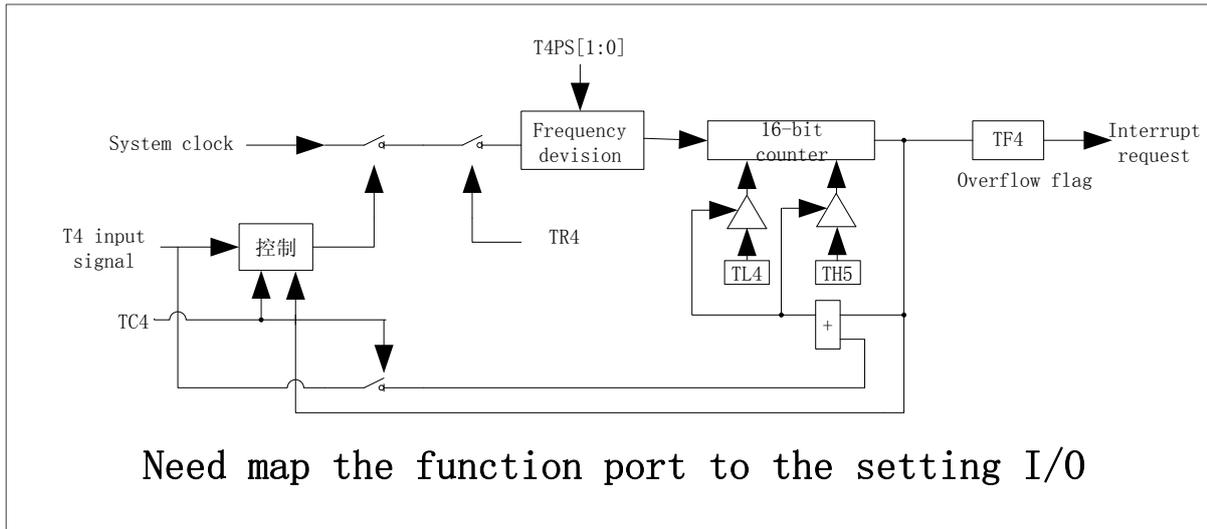
Figure 9 - 7 Timer4 mode2/3 functional block diagram

Note:
(1) When the Timer 4 working as a Timer (TR4 = 1) in baud rate generator mode, TH4 or TL4 cannot be read or written. Because the Timer increment all the time, read or write the results are not accurate. So before access TH4/TL4, Timer 4 must be closed (TR4 = 0).
(2) When the Timer 4 used as counter, input signal frequency of T4 pin must less than half of system clock.

9.4.2 Timer/Counter T4 registers
9.4.2.1 Timer T4 control register T4CON
T4CON

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	TF4	TC4	T4PS[1:0]		T4M [1:0]		TR4	T4CLKS

Bit	Flag	Introductions
7	TF4	Timer 4 overflow flag 0 : No overflow (hardware clear 0),software can also clear 0 1 : Overflow (hardware set 1)
6	TC4	Compare function enable bit When T4M[1:0] = 00 or 01 0: Disable Timer 4 Compare function 1: Enable Timer 4 Compare function When T4M[1:0] = 10 or 11 0 : Timer 4 not be triggered again 1 : Timer 4 can be triggered again
5-4	T4PS[1:0]	Timer 4 prescaler ratio (PRESCALER) selection bit 00 : 1/1 01 : 1/8 10 : 1/64 11 : 1/256
3-2	T4M [1:0]	Timer 4 mode selection bit 00 : Mode0, 16bit auto reload Timer 01 : Mode1, UART1 baud rate generator 10 : Mode2, T4 port rising edge triggered (only the system clock, T4CLKS invalid) 11 : Mode3, T4 port falling edge triggered (only the system clock, T4CLKS invalid)
1	TR4	Timer 4 Enable control bit 0: Disable Timer 4 1: Enable Timer 4
0	T4CLKS	Timer 4 Counting clock source selection bit 0 : The system clock F_{osc} 1 : T4 input an external clock Note: Count clock source after Timer 4 selected is called: f_{T4}.

9.4.2.2 Timer T4 data register TL4, TH4
TL4

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	TL4							

Bit	Flag	Introductions
7-0	TL4	T4 data register low byte

TH4

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset values	0	0	0	0	0	0	0	0
Flag	TH4							

Bit	Flag	Introductions
7-0	TH4	T4 data register high byte

9.5 Timer 5

Timer 5 is 16 bit auto reload Timer. T5CON register control to access the two data registers TH5 and TL5. IEN1 register ET5 bitset 1 enables Timer 5 interrupt (see interrupt chapter).

9.5.1 Timer T5 work mode
9.5.1.1 Mode0: 16 bit auto reload Timer/Counter

When Timer 5 in mode0, TH5 register stores high 8 bits of 16-bit Timer, TL5 stores low 8 bits.

When EXEN5=0, 16 bit Timer register is increased form 0xFFFF to 0x0000 and overflow, set TF5, Timer will automatically load 16 bit value in registers RCAP5H and RCAP5L to TH5 and TL5 registers, if enable Timer 5 interrupt, it will be generated.

When EXEN5=1, Overflow or the falling/Rising edge triggers on external input T5 can trigger a 16 bit reload, and set EXF5 Bit. If ET5 is enabled, TF5 and EXF5 can generate interrupts all.

T5CON.1 register TR5 bit set 1 can enable Timer 5, and don't clear counter of Timer 5. Before enable Timer5, user should write an initial value to reload register user want.
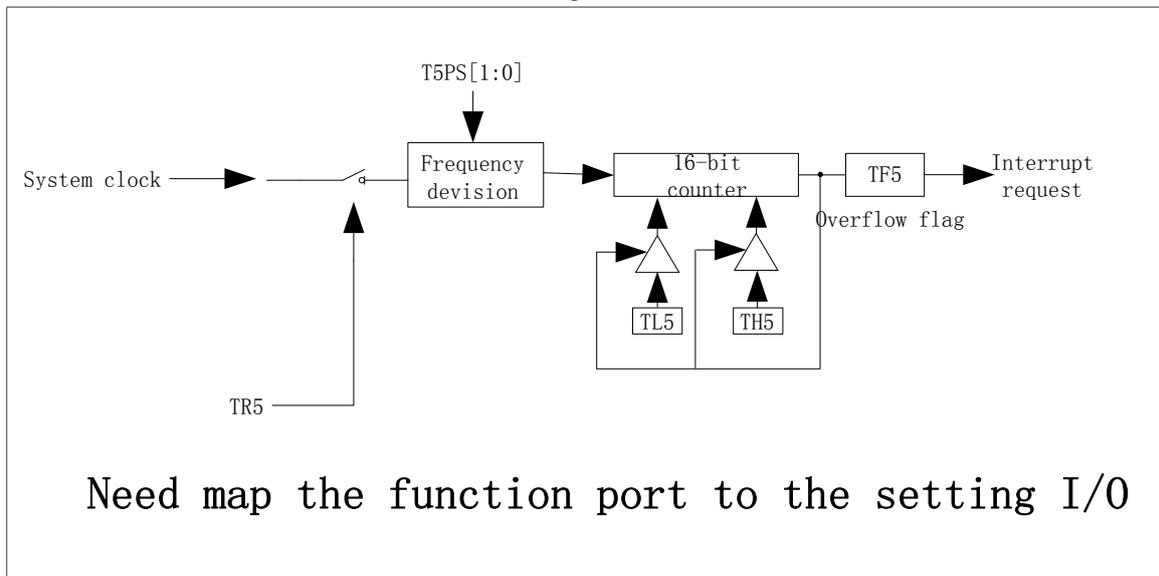
Figure 9 - 8 Timer5 Mode0 functional block diagram

9.5.1.2 Mode1: UART2 baud rate generator

By setting the T5CON register T5M to 01, Select Timer 5 as UART2's baud rate generator. The mode is similar to auto reload mode. Overflow of Timer 5 will cause the 16 -bit value in software Timer5 reload register load to Timer 5 counter, but overflow cannot generate interrupt.

When EXEN5=1 at this time,the falling edge of/rising edge on T5 pins will set EXF5, but not causes reload. So when the Timer 5 as baud rate generator, T5 pins can be used as an additional external interrupt.

UART2 baud rate is calculated by the following formula:

$$BaudRate = \frac{1}{16} \times \frac{f_{T5}/PRESCALER}{65536-[RCAP5H,RCAP5L]},$$ use Timers 5 as baud rate generator.

In the formula, f_{T5} is Fosc, PRESCALER is Timer 5 prescale ratio, RCAP5H and RCAP5L are Timer 5 reload capture register.

9.5.1.3 Mode2: 16 bit rising edge capture

In capture mode, EXEN5 of T5CON has two options.

EXEN5 = 0 The Timer T5 16 Timer, if ET5 Permitting, Timer T5 user can set TF5 Spillovers generate an interrupt. After the overflow count will not be reloaded, from 0 to count.

EXEN5 = 1 The Timer T5 Do the same thing, but External input T5 (Available from the T5CON1 To select the capture input types) the rising edge can cause TH5 TL5 Current value is captured RCAP5H RCAP5L And, in addition, T5 On the rising edge can also result in T5CON EXF5 Is set. ET5 enabled EXF5 like TF5 also generate an interrupt.

9.5.1.4 Mode3: 16 bit falling edge capture

In capture mode, EXEN5 of T5CON has two options.

EXEN5 = 0 the Timer T5 16 Timer, if ET5 Permitting, Timer T5 user can set TF5 spillovers generate an interrupt. After the overflow count will not be reloaded, from 0 to count.

EXEN5 = 1 the Timer T5 Do the same thing, but External input T5 (available from the T5CON1 to select the capture input types) can cause the falling edge TH5 TL5 Current value is captured RCAP5H RCAP5L And, in addition, T5 on the falling edge can also result in T5CON EXF5 Is set. ET5 Enabled EXF5 like TF5 also generates an interrupt.

9.5.2 Timer/Counter T5 registers
9.5.2.1 Timer T5 control register T5CON,T5CON1
T5CON

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	TF5	EXF5	T5PS[1:0]		T5M [1:0]		TR5	EXEN5

Bit	Flag	Introductions
7	TF5	Timer 5 overflow flag 0 : No overflow 1 : Hardware reset on overflow 1 Must be software clearance 0
6	EXF5	T5 flag pin external input event occur 0 : No external input event occur, must be cleared to 0 b software 1 : Detection of external input event and EXEN5= 1, hardware set 1, alse as interrupt request flag
5-4	T5PS[1:0]	Timer 5 prescaler frequency ratio (PRESCALER) selection bit 00 : 1/1 01 : 1/8 10 : 1/64 11 : 1/256
3-2	T5M [1:0]	Timer 5 mode selection flag 00 : Mode0, 16bit auto reload Timer 01 : Mode1, UART2 baud rate generator 10 : Mode2, 16 rising edge captured 11 : Mode3, 16 falling edge capture
1	TR5	Timer 5 enable control flag 0 : Disable Timer 5 1 : Enable Timer 5
0	EXEN5	T5 external event input as the reload on the Pin/capture trigger enable/disable control bit 0 : Ignore events on T5 Pin 1 : a falling or rising edge on T5 pin, get a capture or reload Note: When capturing low frequency RC RXD pin, also need to enable EXEN5 ， and meanwhile setting rising edge capture or falling edge capture.

T5CON1

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-						CAPM[1:0]	

Bit	Flag	Introductions
7-2	-	Reserved
1-0	CAPM[1:0]	Timer 5 capture type selection bit 00 : Edge of T5 change 01 : Internal low frequency RC, that is count clock of watchdog 10 : RXD1 Pin of UART1 11 : RXD2 Pin of UART2

9.5.2.2 Timer T5 data registers TL5, TH5

TL5

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	TL5							

Bit	Flag	Introductions
7-0	TL5	T5 low byte of data registers

TH5

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	TH5							

Bit	Flag	Introductions
7-0	TH5	T5 Data register high byte

9.5.2.3 Timer T5 reload capture registers RCAP5L, RCAP5H

RCAP5L

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	RCAP5L							
Bit	Flag	Introductions						
7-0	RCAP5L	T5 reload capture registers low byte						

RCAP5H

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0

Flag	RCAP5H

Bit	Flag	Introductions
7-0	RCAP5H	T5 reload capture register high byte

10 Pulse width modulation PWM

10.1 PWM Characteristics

➢ 3 complementary PWM outputs with dead-time insertion.6 Independent PWM outputs
➢ Provides each PWM a period overflow interrupt, but share the same interrupt vector
➢ output polarity is selectable
➢ Provides an error frame detection function to close PWM output immediately
➢ PWM clock can be set prescaler ratio
➢ PWM can be used as Timer/Counter

HC89S003F4 integrated three 12 bit PWM module of PWM0, PWM1, and PWM2, each module has a Counter, PWM0 Counter is controlled by PWM0_EN, the Counter start to count when PWM0_EN is enabled, the counter clock source is selected by CK0 bits in PWM0C register.

When user need to output PWM waveform from MCU Pin, user need to enable PWM0_OEN or PWM01_OEN, and need to set port at strong push-pull mode. No PWM waveform output when PWM0_OEN or PWM01_OEN disabled. And this time PWM0 Counter can be used as a Timer, when counter overflow, PWM interrupt occur when interrupt is enabled.

Set EFLT0 to 1, PWM0 output and its complementary output can be closed by input signal variation on FLT0 pin automatically. Once detected valid input on FLT0 pin,PWM output will closed immediately, but PWM internal Counter continue to run, after the error signal removed from FLT0 ,the PWM output continued. During FLT0 input signal valid period, FLT0S cannot be cleared. Only when the FLT0 input signal invalid, FLT0S status can be cleared by software, and PWM return to output.

PWM0 fault detection port FLT0 (P0.0 port) has debouncing function, user can set the appropriate debouncing time. Configure FLT0 debouncing time equal configure port P0.0 debouncing time, setting method in ports debouncing control register P00DBC, P01DBC and P02DBC chapter.

Three PWM modules function and operation are exactly the same. User can control register to 3 roads with dead complementary PWM or six roads single PWM output.

3 PWM modules share an interrupt vector entrance, but have their own control bit and flag. It is used for user modify the PWM module cycles or duty cycle of the next cycle.

10.2 PWM output mode

PWM module contains 3 independent waveform generate modules, the corresponding 3 PWM output are   PWM0/PWM01,PWM1/PWM11,PWM2/PWM21, by controlling the associated registers to provide each pair PWM output configured as a complementary output mode or independent mode.

10.2.1 Complementary output mode

Set PWMxM(x =0,1,2) to 0 : PWM will be working in complementary output mode, enable PWMx&PWMx (x =0,1,2) 1 output, and control the cycle registers, duty registers and dead-time registers to output the complementary waveform. The PWMx&PWMx1(x =0, 1, 2) polarity can be selected in complementary output mode. It is easy to user multiple level driven request.

10.2.2 Independent output mode

Set PWMxM to 1: PWM will be working in independent mode, user can control PWMx&PWMx1 single or both output. their cycle time are same but the duty cycle can be set individually. Duty cycle register control PWMx duty cycle, dead-time control registers control PWMx1 duty cycle,
The PWMx&PWMx1 polarity can be selected in independent output mode. It is easy to user multiple level driven request(x =0, 1, 2).

10.3 PWM registers

10.3.1 PWM0 module
10.3.1.1 PWM0 enable register PWM0EN

Bit	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-	FLT0_MODE		EFLT0	PWM0M	PWM01_OEN	PWM0_OEN	PWM0_EN

Bit	Flag	Introductions
7	-	Reserved bit
6-5	FLT0_MODE	PWM0 fault output book status selection bit 00 : PWM0&PWM01 low level both during failure 01 : PWM0 low level during failure, PWM01 high level during failure 10 : PWM0 high level during failure, PWM01 low level during failure 11 : PWM0&PWM01 high level both during failure
4	EFLT0	PWM0 FLT0 control pin enable bit 0 : Disable fault detection, GPIO function or other functions 1 : Enable fault detection, PWM0 fault detection of input pins Note: the complementary and independent output mode can be controlled by the foot fault detection.
3	PWM0M	PWM0 working mode selection bit 0 : PWM0&PWM01 working in complementary output mode 1 : PWM0&PWM01 working in independent mode Note: Recommended closing PWM0 module before modify PWM0 working mode
2	PWM01_OEN	PWM01 output control bit 0：Disable PWM01 output 1：Enable PWM01 output
1	PWM0_OEN	PWM0 output control bit 0 :Disable PWM0 output 1 :Enable PWM0 output Note: PWM0 output is enabled when PWM0_EN is set to 1, otherwise PWM0 output disable. (the port output must be set output mode); Even prohibit output, as long as PWM0_EN is enabled, the PWM0 can occur overflow interrupt, PWM0 can be used as a Timer and the control bit is valid immediately when changed .
0	PWM0_EN	PWM0 module control enable bit 0 : Close PWM0 module 1 : Open PWM0 module (re-count) Note: When PWM0 close, counter stopped and output close immediately .When PWM0 open, PWM0 re-count from 1, output controlled by PWM0_OEN and PWM01_OEN bits.

10.3.1.2 PWM0 control register PWM0C

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PWM0IE	PWM0IF	FLT0S	FLT0C	PWM0S		CK0	

Bit	Flag	Introductions
7	PWM0IE	PWM0 interrupt enable bit 0 : Disable PWM0 interrupt 1 : Enable PWM0 interrupt
6	PWM0IF	PWM0 interrupt flag 0 : Software clear 0 1 : PWM0 cycle counter overflow, hardware set 1
5	FLT0S	PWM0 FLT status bit 0 : PWM normal status, software clear 0 1 : PWM output off, hardware set 1
4	FLT0C	PWM0 FLT pin configuration bit 0 : FLT0 low level, PWM output off 1 : FLT0 high level, PWM output off
3-2	PWM0S	PWM0 PWM01 output mode selection bits 00 : PWM0 and PWM01 high level valid 01 : PWM0 high level valid, PWM01 low level valid 10 : PWM0 low level valid, PWM01 high level valid 11 : PWM0 PWM01 low level valid Note: For independent mode, the output mode selection bits is also valid, but different with complementary mode is: valid period is duty cycle period, but in complementary mode, valid period of PWM0 is duty cycle period, valid period of PWM01 is complementary duty cycle period.
1-0	CK0	PWM0 clock source selection bits 00 : F_{osc} /1 01 : F_{osc}/8 10 : F_{osc}/32 11 : F_{osc}/128

PWM0S=00 & PWM0M=0: PWM0 and PWM01 work in a complementary mode and high level valid both
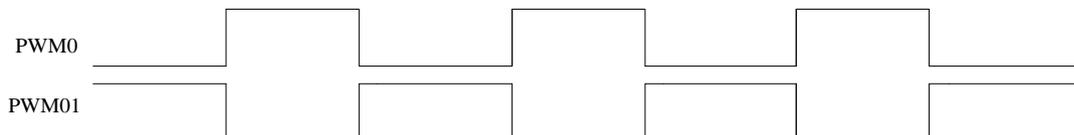
PWM0S=00 & PWM0M=1: PWM0 and PWM01 work in a mode and high level valid both
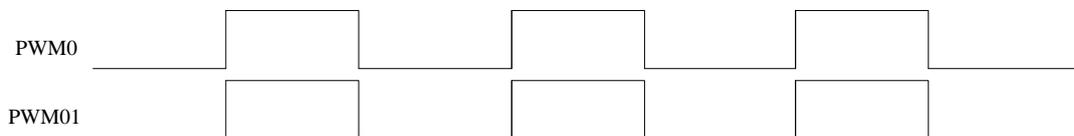
PWM0S=01 & PWM0M=0: PWM0 PWM01 work in a complementary mode, and PWM0 high level valid, PWM01 low level valid
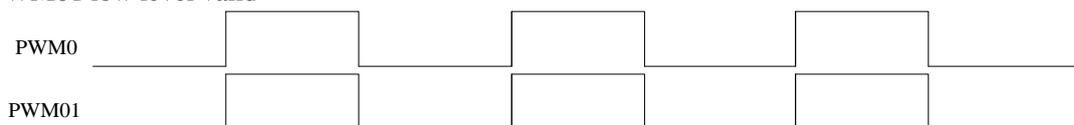
PWM0S=01 & PWM0M=1: PWM0 PWM01 work in an independent mode, and PWM0 high level valid, PWM01 low level valid
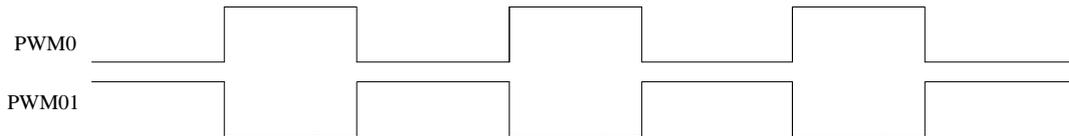
 PWM0S=10 & PWM0M=0: PWM0 PWM01 work in a complementary mode, and PWM0 low level valid, PWM01 high level valid
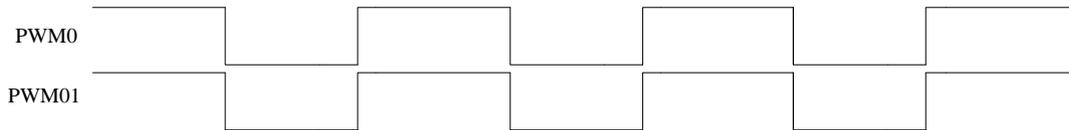
PWM0S=10 & PWM0M=1: PWM0 PWM01 work in an independent mode, and PWM0 low level valid, PWM01 high level valid
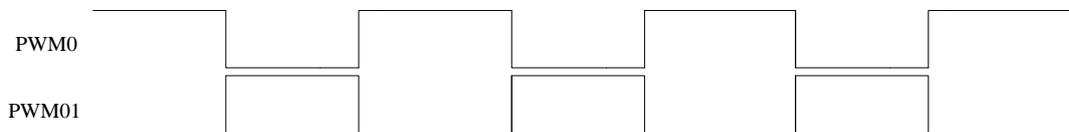
PWM0S=11 & PWM0M=0: PWM0 PWM01 work in a complementary mode, and low level valid both
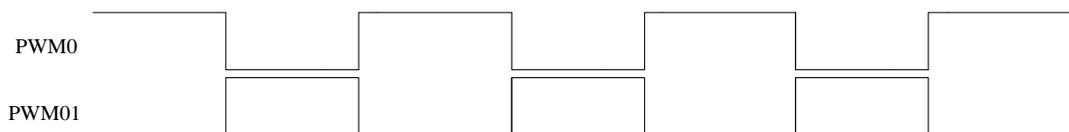
PWM0S=11 & PWM0M=1: PWM0 PWM01 work in an independent mode, and low level valid both
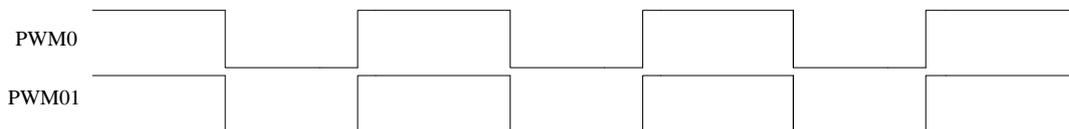
10.3.1.3 PWM0 period register PWM0PL,PWM0PH

PWM0PL

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PWM0PL[7:0]							

Bit	Flag	Introductions
7-0	PWM0PL[7:0]	PWM0 **cycle** register low 8 bits

PWM0PH

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-	-	-	-	PWM0PH[3:0]			

Bit	Flag	Introductions		
7-4	-	Reserved (read = 0b,, write invalid)		
3-0	PWM0PH[3:0]	PWM0 **cycle** register high 4 bits		

Note: modify high bits firstly when modify PWM0 cycle, then modify low bits, read as not restricted, such as:

(1) PWM0PH = 0x05;
(2) PWM0PL = 0x08;   // PWM Counter overflow, the cycle data is 0x0508 from the next cycle
(3) PWM0PH = 0x06;   // PWM Counter overflow, the cycle data is 0x0508 from the next cycle
(4) PWM0PL = 0x08;   // PWM Counter overflow, the cycle data is 0x0608 from the next cycle
(5) PWM0PL = 0x09;   // PWM Counter overflow, the cycle data is 0x0609 from the next cycle

As long as PWM Period modified, regardless of whether the low registers need to be modified, low bits has to be written one time, and cycle changes will valid only from the next PWM cycle.

PWM0 cycle = [PWM0PH : PWM0PL] * PWM0 Clock cycle

10.3.1.4 PWM0 duty cycle register PWM0DL,PWM0DH

PWM0DL

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PWM0DL[7:0]							

Bit	Flag	Introductions
7-0	PWM0DL[7:0]	PWM0 Duty cycle register low 8 bits

PWM0DH

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-	-	-	-	PWM0DH[3:0]			

Bit	Flag	Introductions		
7-4	-	Reserved (read = 0b,,write invalid)		
3-0	PWM0DH[3:0]	PWM0 duty cycle registers high 4 bits		

Note: modify PWM0 Duty cycle registers, similar to modify PWM0 cycle register, both are required to modify the high level first then low, and changes will valid from the next cycle.

PWM0 Duty cycle = [PWM0DH : PWM0DL] * PWM0 Clock cycle

10.3.1.5 PWM0 dead time register PWM0DTL, PWM0DTH

PWM0DTL

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PWM0DTL[7:0]							

Bit	Flag	Introductions
7-0	PWM0DTL[7:0]	PWM0 Dead time register low 8 bits

PWM0DTH

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Reset values	0	0	0	0	0	0	0	0
Flag	-	-	-	-	PWM0DTH[3:0]			

Bit	Flag	Introductions
7-4	-	Reserved (read = 0b,,write invalid)
3-0	PWM0DTH[3:0]	PWM0 Dead time register high 4

When PWM0M=1, PWM0 work in 2 road independent mode, dead time register is used as the PWM01 Duty cycle registers, namely independent mode PWM0 can output 2 road PWM waveform with the same cycle, but different duty cycle.

Complementary mode: PWM0 Dead time = [PWM0DTH:PWM0DTL] * PWM0 clock cycle.

Complementary mode: dead time must be less than the duty cycle time, sum of dead time and duty cycle time must be less than PWM0 cycle.

Independent mode: PWM01 Duty cycle time = [PWM0DTH:PWM0DTL] * PWM0 Clock cycle.

10.3.2 10.3.2 PWM1 module
10.3.2.1 PWM1 enable register PWM1EN

Bit	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-	FLT1_MODE	EFLT1	PWM1M	PWM11_OEN	PWM1_OEN	PWM1_EN

Bit	Flag	Introductions
7	-	Reserved bit
6-5	FLT1_MODE	PWM1 fault output predetermined status selection bits 00 : PWM1&PWM11 are low level during the fault 01 : PWM1 low level is during fault, PWM11 is high level during fault 10 : PWM1 high level is during fault, PWM11 is low level during fault 11 : PWM1&PWM11 are high level during fault
4	EFLT1	PWM1 FLT1 control pin enable bit 0 : Disable fault detection, GPIO function or other functions 1 : Enable fault detection, PWM1 fault detection input pin Note: the complementary output mode and independent output mode are controlled by the fault detection pin.
3	PWM1M	PWM1 work mode selection bit 0 : PWM1&PWM11 work in complementary output mode 1 : PWM1&PWM11 work in independent mode Note: Recommended to close PWM1 module when modifying PWM1 work mode
2	PWM11_OEN	PWM11 output control bit 0 : Disable PWM11 output 1 : Enable PWM11 output
1	PWM1_OEN	PWM1 output control 0 : Disable PWM1 output 1 : Enable PWM1 output Note: PWM1 output is valid only when PWM1_EN is set to 1, otherwise PWM1 output off (the corresponding output port must be set output mode); even output is disabled, as long as PWM1_EN I\is enabled, the PWM1 can occur overflow interrupt, that is to say PWM1 can be used as a Timer, this control bit is valid immediately once changed.
0	PWM1_EN	PWM1 module control bit 0 : Close PWM1 module 1 : Open PWM1 module (re-count) Note: When closed, PWM1 count stopped, the output is closed

immediately. When opened, PWM1 counter re-count from 1, output is controlled by the PWM1_OEN and PWM11_OEN bits.

10.3.2.2 PWM1 control register PWM1C

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PWM1IE	PWM1IF	FLT1S	FLT1C	PWM1S		CK1	

Bit	Flag	Introductions
7	PWM1IE	PWM1 interrupt enable bit 0 : Disable PWM1 interrupt 1 : Enable PWM1 interrupt
6	PWM1IF	PWM1 interrupt flag 0 : Software clearance 0 1 : PWM1 cycle counter overflow, set 1 by hardware
5	FLT1S	PWM1 FLT state bits 0 : PWM normal state, software clear 0 1 : PWM output off, hardware set 1
4	FLT1C	PWM1 FLT pin configuration bit 0 : When FLT1 is low level, PWM output off 1 : When FLT1 is high level, PWM output off
3-2	PWM1S	PWM1 and PWM11 output mode selection bits 00 : PWM1 and PWM11 are active high 01 : PWM1 is active high, PWM11 is active low 10 : PWM1 is active low, PWM11 is active high 11 : PWM1 and PWM11 are active low Note: for independent mode, the output mode selections bit is also valid, but different with complementary mode is: the valid period is duty cycle period, and in complementary mode, the valid period of PWM1 is duty cycle period, the valid period of PWM11 is complementary duty cycle period.
1-0	CK1	PWM1 clock source selection bits 00 : F_{osc} /1 01 : F_{osc}/8 10 : F_{osc}/32 11 : F_{osc}/128

10.3.2.3 PWM1 period register PWM1PL, PWM1PH
PWM1PL

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PWM1PL[7:0]							

Bit	Flag	Introductions
7-0	PWM1PL[7:0]	PWM1 **cycle** register low 8 bits

PWM1PH

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-	-	-	-	PWM1PH[3:0]			

Bit	Flag	Introductions
7-4	-	Reserved (read = 0b, write invalid)
3-0	PWM1PH[3:0]	PWM1 cycle register high 4 bits

Note: Modify PWM1 cycle register high bits first, then modify low bits, read is not unlimited, such as:

(6) PWM1PH = 0x05;

(7) PWM1PL = 0x08;   //PWM counter overflow, the cycle data is 0x0508 form the next cycle

(8) PWM1PH = 0x06;   //PWM counter overflow, the cycle data is 0x0508 form the next cycle

(9) PWM1PL = 0x08;   //PWM counter overflow, the cycle data is 0x0608 form the next cycle

(10) PWM1PL = 0x09;   //PWM counter overflow, the cycle data is 0x0609 form the next cycle

As long as PWM period is modified, regardless of whether the low registers need to be modified, low bits has to be written once, and cycle changes are valid only from the next PWM cycle.

PWM1 cycle = [PWM1PH : PWM1PL] * PWM1 work clock source cycle

10.3.2.4 PWM1 duty register PWM1DL, PWM1DH
PWM1DL

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PWM1DL[7:0]							

Bit	Flag	Introductions
7-0	PWM1DL[7:0]	PWM1 duty register low 8 bits

PWM1DH

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-	-	-	-	PWM1DH[3:0]			

Bit	Flag	Introductions
7-4	-	Reserved (read = 0b, write invalid)
3-0	PWM1DH[3:0]	PWM1 duty registers high 4 bits

Note: Modify PWM1 duty registers, similar to modify PWM1 cycle register, are required to modify the high then low bits, and changes are valid only from the next cycle.

PWM1 Duty cycle = [PWM1DH : PWM1DL] * PWM1 work clock source cycle

10.3.2.5 PWM1 dead time register PWM1DTL, PWM1DTH
PWM1DTL

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PWM1DTL[7:0]							

Bit	Flag	Introductions
7-0	PWM1DTL[7:0]	PWM1 dead time register low 8 bits

PWM1DTH

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Reset values	0	0	0	0	0	0	0	0
Flag	-				PWM1DTH[3:0]			

Bit	Flag	Introductions
7-4	-	Reserved (read = 0b, write invalid)
3-0	PWM1DTH[3:0]	PWM1 dead time register high 4 bits

When PWM1M=1, PWM1 is worked in 2 independent mode, dead time register is used as the PWM11 duty cycle registers, namely independent mode PWM1 can generate 2 ways same cycle, but different duty cycle PWM waveform.

In complementary mode: PWM1 dead time = [PWM1DTH: PWM1DTL] * PWM1 work clock cycle;

In complementary mode: dead time must be less than the duty cycle time, sum of dead time and duty time must be less than PWM1 cycle;

In independent mode: PWM11 duty cycle time = [PWM1DTH: PWM1DTL] * PWM1 work clock cycle;

10.3.3 PWM2 module
10.3.3.1 PWM2 enable register PWM2EN

Bit	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-	FLT2_MODE	EFLT2	PWM2M	PWM21_OEN	PWM2_OEN	PWM2_EN

Bit	Flag	Introductions
7	-	Reserved bit
6-5	FLT2_MODE	PWM2 fault output predetermined selection bits 00 : PWM2&PWM21 are low level during fault 01 : PWM2 is low level during fault, PWM21 is high level during fault 10 : PWM2 is high level during fault, PWM21 is low level during fault 11 : PWM2&PWM21 are high level during fault
4	EFLT2	PWM2 FLT2 control pin enable bit 0 : Disable fault detection, GPIO function or other functions 1 : Enable fault detection, PWM2 fault detection input pins Note: the complementary output mode and independent output mode are controlled by the fault detection pin.
3	PWM2M	PWM2 work mode selection bit 0 : PWM2&PWM21 work in complementary output mode 1 : PWM2&PWM21 work in independent mode Note: Recommended to close PWM1 module when modifying PWM1 work mode
2	PWM21_OEN	PWM21 output control bit 0 : Disable PWM21 output 1 : Enable PWM21 output
1	PWM2_OEN	PWM2 output control bit 0 : Disable PWM2 output 1 : Enable PWM2 output Note: PWM2 output is valid only when PWM2_EN is set to 1, otherwise PWM2 output off (the corresponding output port must be set output mode); even output is disabled, as long as PWM2_EN I\is enabled, the PWM2 can occur overflow interrupt, that is to say PWM2 can be used as a Timer, this control bit is valid immediately once changed.
0	PWM2_EN	PWM2 module control bit 0 : Closed PWM2 module 1 : Open PWM2 module (re-count) Note: When closed, PWM2 count stopped, the output is closed

immediately. When opened, PWM1 counter re-count from 1, output is controlled by the PWM2_OEN and PWM21_OEN bits.

10.3.3.2 PWM2 control register PWM2C

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PWM2IE	PWM2IF	FLT2S	FLT2C	PWM2S		CK2	

Bit	Flag	Introductions
7	PWM2IE	PWM2 interrupt enable bit 0 : Disable PWM2 interrupt 1 : Enable PWM2 interrupt
6	PWM2IF	PWM2 interrupt flag 0 : Software clearance 0 1 : PWM2 cycle counter overflow, set 1 by hardware
5	FLT2S	PWM2 FLT state bit 0 : PWM normal state, software clear 0 1 : PWM output off, hardware set 1
4	FLT2C	PWM2 FLT pin configuration bit 0 : When FLT1 is low level, PWM output off 1 : When FLT1 is high level, PWM output off
3-2	PWM2S	PWM2 and PWM21 output mode selection bits 00 : PWM2 and PWM21 are active high 01 : PWM2 is active high, PWM21 is active low 10 : PWM2 is active low, PWM21 is active high 11 : PWM2 and PWM21 are active low Note: for independent mode, the output mode selections bit is also valid, but different with complementary mode is: the valid period is duty cycle period, and in complementary mode, the valid period of PWM2 is duty cycle period, the valid period of PWM21 is complementary duty cycle period.
1-0	CK2	PWM2 Clock source selection bits 00 : F_{osc} /1 01 : F_{osc}/8 10 : F_{osc}/32 11 : F_{osc}/128

10.3.3.3 PWM2 period register PWM2PL, PWM2PH

PWM2PL

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PWM2PL[7:0]							

Bit	Flag	Introductions
7-0	PWM2PL[7:0]	PWM2 **cycle** register low 8 bits

PWM2PH

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-	-	-	-	PWM2PH[3:0]			

Bit	Flag	Introductions
7-4	-	Reserved (read = 0b, write invalid)
3-0	PWM2PH[3:0]	PWM2 cycle register high 4 bits

Note: Modify PWM2 cycle register high bits first, then modify low bits, read is not unlimited, such as:

(6) PWM2PH = 0x05;

(7) PWM2PL = 0x08; //PWM counter overflow, the cycle data is 0x0508 form the next cycle

(8) PWM2PH = 0x06; //PWM counter overflow, the cycle data is 0x0508 form the next cycle

(9) PWM2PL = 0x08; //PWM counter overflow, the cycle data is 0x0608 form the next cycle

(10) PWM2PL = 0x09; //PWM counter overflow, the cycle data is 0x0609 form the next cycle

As long as PWM period is modified, regardless of whether the low registers need to be modified, low bits has to be written once, and cycle changes are valid only from the next PWM cycle.

PWM2 cycle = [PWM2PH : PWM2PL] * PWM2 work clock source cycle

10.3.3.4 PWM2 duty register PWM2DL, PWM2DH

PWM2DL

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PWM2DL[7:0]							

Bit	Flag	Introductions
7-0	PWM2DL[7:0]	PWM2 duty register low 8 bits

PWM2DH

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-				PWM2DH[3:0]			

Bit	Flag	Introductions
7-4	-	Reserved (read = 0b, write invalid)
3-0	PWM2DH[3:0]	PWM2 duty register high 4 bits

Note: Modify PWM2 duty registers, similar to modify PWM2 cycle register, are required to modify the high then low bits, and changes are valid only from the next cycle.

PWM2 Duty cycle = [PWM2DH : PWM2DL] * PWM2 work clock source cycle

10.3.3.5 PWM2 dead time register PWM2DTL, PWM2DH

PWM2DTL

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PWM2DTL[7:0]							

Bit	Flag	Introductions
7-0	PWM2DTL[7:0]	PWM2 dead time register low 8 bits

PWM2DTH

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W

Reset values	0	0	0	0	0	0	0	0
Flag	-				PWM2DTH[3:0]			

Bit	Flag	Introductions
7-4	-	Reserved (read = 0b, write invalid)
3-0	PWM2DTH[3:0]	PWM2 dead time register high 4 bits

When PWM2M=1, PWM2 is worked in 2 independent mode, dead time register is used as the PWM21 duty cycle registers, namely independent mode PWM2 can occur 2 ways same cycle, but different duty cycle PWM waveform.

In complementary mode: PWM2 dead time = [PWM2DTH: PWM2DTL] * PWM2 work clock cycle;

In complementary mode: dead time must be less than the duty cycle time, sum of dead time and duty time must be less than PWM2 cycle;

In independent mode: PWM21 duty cycle time = [PWM2DTH: PWM2DTL] * PWM2 work clock cycle;

11 Single 8 bit PWM

11.1 PWM characteristics

➢ 8 bits PWM output
➢ Provides PWM cycle overflow interrupt, but the interrupt share the same vector with PWM0, PWM1 and PWM2.
➢ output polarity is selectable
➢ PWM can be used as Timer/Counter, namely cycle register used as Timer when write, read as counter when read.

11.2 PWM module registers

11.2.1 PWM3 control register PWM3C

PWM3C

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PWM3EN	PWM3IE	PWM3IF	PWM3OEN	PWM3S	PTCK3[2:0]		

Bit	Flag	Introductions
7	PWM3EN	PWM3 module enable control bit 0 : Close PWM3 module 1 : Open PWM3 module (re-count) Note: When PWM close, counter stopped and output close immediately .When PWM open, PWM re-count from 1, output controlled by PWM3OEN bit.
6	PWM3IE	PWM3 interrupt enable bit 0 :Disable PWM3 interrupt 1 :Enable PWM3 interrupt
5	PWM3IF	PWM3 interrupt flag 0: Software clear 0 1: Hardware set 1, only set to 1 when the PWM3 Counter overflow (greater than PWM3P)
4	PWM3OEN	PWM3 output enable bit 0 : PWM3 Disable output 1 : PWM3 Enable output Note: PWM3 output is enabled when PWM3EN is set to 1, otherwise PW3 output disable. (the port output must be set output mode); Even prohibit output, as long as PWM3EN is enabled, the PWM3 can occur overflow interrupt, PWM3 can be used as a Timer and the control bit is valid immediately when changed .
3	PWM3S	PWM3 output polarity selection bit 0 : PWM3 high level during valid period 1 : PWM3 low level during valid period Note: modify the control bit will be effective immediately, valid period is duty cycle period
2-0	PTCK3[2:0]	PWM3 clock source selection bits 000 : $F_{osc}/1$ 001 : $F_{osc}/2$ 010 : $F_{osc}/4$ 011 : $F_{osc}/8$ 100 : $F_{osc}/16$ 101 : $F_{osc}/32$ 110 : $F_{osc}/64$ 111 : $F_{osc}/128$ Note: modify the control bit will be effective immediately, modify is not recommended during output.

11.2.2 PWM3 period register PWM3P

PWM3 cycle register PWM3P

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PWM3P[7:0]							

Bit	Flag	Introductions
7-0	PWM3P[7:0]	PWM3P cycle register

11.2.3 PWM3D duty register PWM3D

PWM3 Duty register PWM3D

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	PWM3D[7:0]							

Bit	Flag	Introductions
7-0	PWM3D[7:0]	PWM3D duty cycle register When PWM3P ≤ PWM3D, the duty cycle 100%; PWM3D = 0x00, the duty cycle 0%

12 Watchdog timer WDT

12.1 WDT characteristics

➢ Can be configured for overflow reset
➢ Configurable idle/power-down mode enable or not
➢ Flexible configure overflow time

HC89S003F4 watchdog Timer is an incremental counter, the clock source is internal low frequency RC, You can configure register to select run or not in idle/power-down mode. When WDT overflow, user can get the chip reset or not by WDTRF in RSTFR register. If WDTRST is 1, the system will reset when WDT overflow, if WDTRST is 0, and WDT interrupt is enabled, then WDT interrupt occur.

Due to the internal low frequency RC Oscillator frequency has deviation with process. The real frequency of it can be measured via Timer 5 capture, and calculate the overflow time according to the actual frequency value.

HC89S003F4 watchdog Timer has overflow flag after overflow, reset has a special reset flag, frequency division and overflow value can be set, clear WDT only need to set the corresponding control bit, operation is flexible.

12.2 WDT registers

12.2.1 WDT control register WDTC

Bit	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	W	R/W	R/W	R/W	R/W
Reset values	0	1	0	0	1	1	1	1
Flag	-	WDTRST	WDTF	WDTCLR	WDTPD	WDTPS[2:0]		

Bit	Flag	Introductions
7	-	Reserved bit
6	WDTRST	WDT reset enable bit 0 : Disable WDT reset 1 : Enable WDT reset Note: Disable WDT reset, interrupt request flag can still set when WDT Count overflow
5	WDTF	WDT interrupt request flag 0 : No WDT count overflow, when interrupt response software clear 0 1 : WDT count overflow, WDTF hardware reset 1, can be used for interrupt request
4	WDTCLR	Watchdog clear 0 　Set 1 can clear WDT counter, hardware clear 0 automatically
3	WDTPD	WDT idle/power-down mode control bit 0 : Enable WDT in idle/power-down mode, if WDTRST=1 will reset wakeup the system, if WDTRST=0 and EA=1, EWDT=1 Will interrupt to wake up the system. 1 : disable WDT in idle/power-down mode
2-0	WDTPS[2:0]	The watchdog Timer clock source frequency division selection bits 000 : 1/8 001 : 1/16 010 : 1/32 011 : 1/64 100 : 1/128 101 : 1/256 110 : 1/512 111 : 1/1024

12.2.2 WDT count compare register WDTCCR

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	1	1	1	1	1	1	1	1
Flag	WDTCCR[7:0]							

Bit	Flag	Introductions
7-0	WDTCCR[7:0]	WDT Count compare register flags Note: When WDT Counter match with WDTCCR[7:0], overflow and counter clear 0 then Count again. Write 0 will be turned off WDT Function (don't close the internal low frequency RC), namely disable WDT. Write non-0 data, will start the WDT.

Below results is calculated under 44KHz, the real frequency of it can be measured via Timer 5 capture.
Overrun time = (WDT frequency division coefficient * (WDTCCR [7:0] + 1))/real internal low frequency RC frequency.
WDTCCR[7:0] = 0xFF Watchdog overflow time table as below.

PS2	PS1	PS0	WDT Frequency division coefficient	Adjusting the step value	WDT Maximum overflow time @44K
0	0	0	8	0.25 ms	46.55ms
0	0	1	16	0.5 ms	93.09ms
0	1	0	32	1 ms	186.18ms
0	1	1	64	2 ms	372.36ms
1	0	0	128	4 ms	744.73ms
1	0	1	256	8 ms	1489.45ms
1	1	0	512	16 ms	2978.91ms
1	1	1	1024	32 ms	5957.82ms

13 Universal asynchronous transceiver UART

13.1 UART characteristics

- ➢ 2 UART
- ➢ Multiple work modes
- ➢ Multiple errors detection

13.2 Work mode

UART has 4 kinds of work modes, in all modes, any SBUF write operations as a destination register will start transmission. In mode0 RI = 0 and REN = 1 used to initializes receiver. TXD Pin generates a clock signal, and RXD Pin shift 8 -bits data. In other modes the start bit of input initializes receiver (if RI = 0 and REN = 1). The communication of external transmitter started when sending the start bit. TXD pin must be set as output high before transmission.

SM0	SM1	Work mode	Type	Baud rate
0	0	0	Synchronous	The baud rate is $F_{osc}/12 \times 6^{UX6}$
0	1	1	Asynchronous	Timer 4 overflow rate /16
1	0	2	Asynchronous	$(2^{SMOD}/64) \times F_{osc}$
1	1	3	Asynchronous	Timer 4 overflow rate /16

13.2.1 Mode0 : Synchronous half-duplex communication

Mode0 support synchronous communication of external devices, RXD pin send and receive serial data, TXD Pin send shift clock. HC89S003F4 provides the shift clock on TXD pin, so this mode is half-duplex serial communications. In the mode, each frame receives 8 -bits, low bit received or sent first.

By set UX6 to 0 or 1, baud rate fix $1/12 \ast F_{osc}$ or $1/2 \ast F_{osc}$. When UX6=0, serial port with f_{OSC} 1/12 running when UX6 1 Shi, serial port F_{osc} 1/2 Running. The only difference with Standard 8051 is that HC89S003F4 has variable baud rate in mode0.

Function block diagram is shown as below figure, data RXD pin moves into and out of the serial port, the shift clock by TXD pin output.
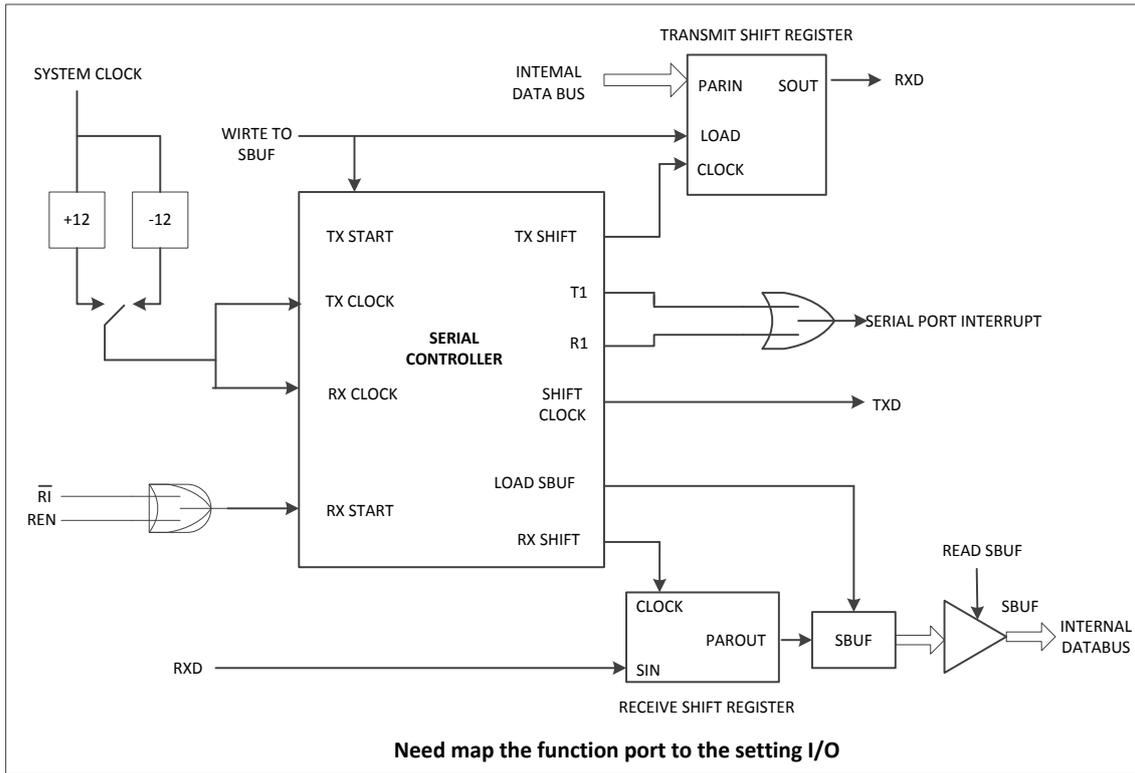
Figure 13 - 1 UART mode0 function block diagram

Any write operation with SBUF as a destination register will start transmission. TX control module start to transmit at next system clock. Data switch take place at the falling edge of the clock, data in shift register ordinal shifted from left to right, empty position set 0.When all 8 bits are sent,TX control modules send operation is stopped, and then TI set to 1 at the rising edge of next system clock.
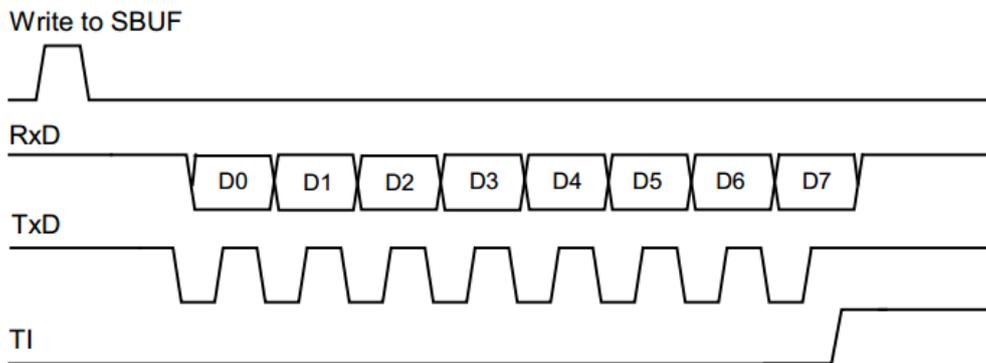
Figure 13 - 2 Mode0 data send timing diagram

REN set 1and RI clear 0 to initialize receiver. The next system clock start to receive, latch data at rising edge of the shift clock, and data in receive conversion register ordinal shifted to left. After all 8 -bit data moved to the shift register, and RX control module stop receiving, RI is set at the rising edge of next system clock, until it is cleared by software to enable the next reception.
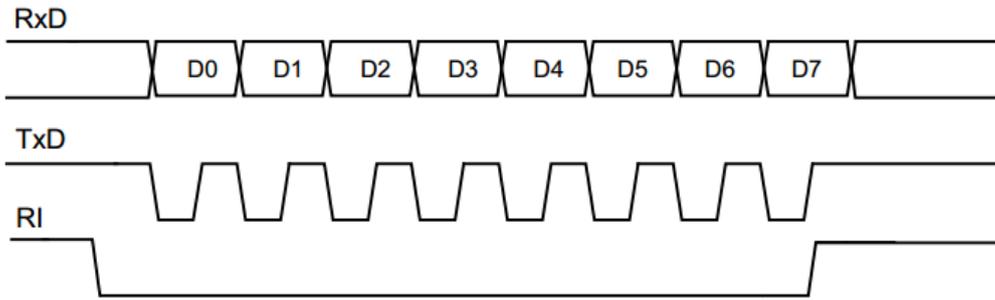
Figure 13 - 3 Mode0 Data receive timing diagram

13.2.2 Mode1 : 8 UART, variable baud rate, asynchronous full duplex

Mode1 provides 10 bits communication of full duplex asynchronous, 10 bits consist of a start bit (logical 0), 8 data bits (low bit first) and one stop bit (logic 1). When receiving, 8 data bits stored in SBUF and stop bit stored in RB8. Mode1 baud rate equals Timer 4 overflow rate /16.

Function block diagram is shown in the following figure:
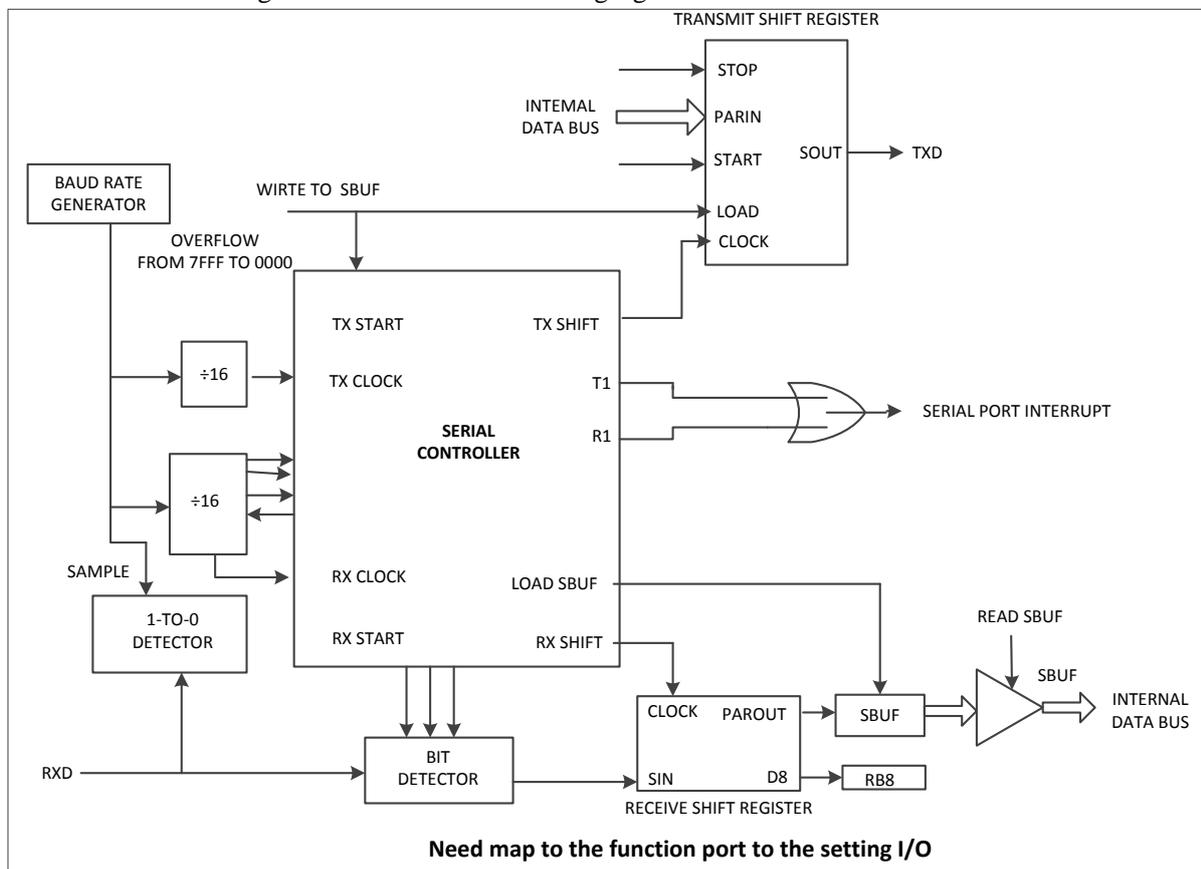

Figure 13 - 4 UART mode1 functional block diagram

Any SBUF write operations as a destination register will start transmission. Actually sending is started from the system clock after 16 scale frequency counter's next jump. So bit time is synchronous with 16 frequency division counter, but out-sync with SBUF write operation. Start bit shift out from TXD Pin first, and then shift 8 bits data. After all the 8 bits data in send shift register is sent, the stop bit shift out from TXD Pin, at the same time TI flag set.

Note: In this mode, when receiving data via UART1, RI cannot be set normally, but user can query RB8 by software for data reception.
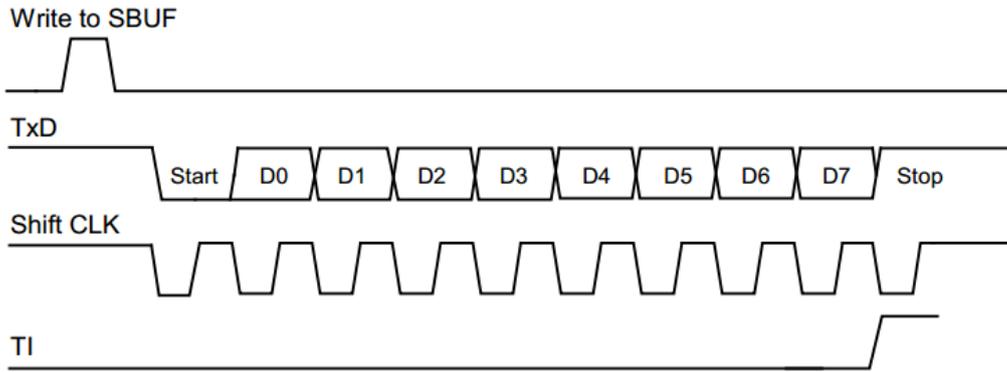
Figure 13 - 5 mode1 send the data time series block diagram

Receive is enabled only when REN set 1. UART start to receive data when the falling edge of RXD is detected. The CPU need to sample RXD pin continuously, sampling rate equal 16 times of baud rate. When detecting falling edge, 16 division frequency counter reset immediately to help 16 frequency counter and RXD pin serial data synchronization. 16 frequency counter's every bit time is divided into 16 states, at the 7, 8, and 9 states, the bit detector sample the level on RXD pin. To restrain noise, in this 3 sample states, at least 2 samples data are same，the data will be received. If first received bit is not 0, indicates that this is not a start of frame, this bit is ignored, the receive circuit is reset, wait for the RXD pin of another falling rise. If start bit is valid, then move into the shift register, and then move the other bits to shift register. 8 data bits and 1 stop bit (stop bits contain errors, as described in the description of register SM2) ,after moving, the data of the shift register and the stop bit (stop bits that contain errors) is loaded into SBUF and RB8   respectively,RI set 1, but it must meet the following conditions:

(1) RI = 0
(2) SM2 = 0 or stop bit received = 1

If these conditions are met, then stop bit (contain the error stop bit) id loaded into RB8, 8 data is loaded into SBUF, RI is set to 1. Otherwise the receive frame is lost. At this time, the receiver will return to detect RXD port if there has another falling edge. User must use the software to clear RI, and receive again.
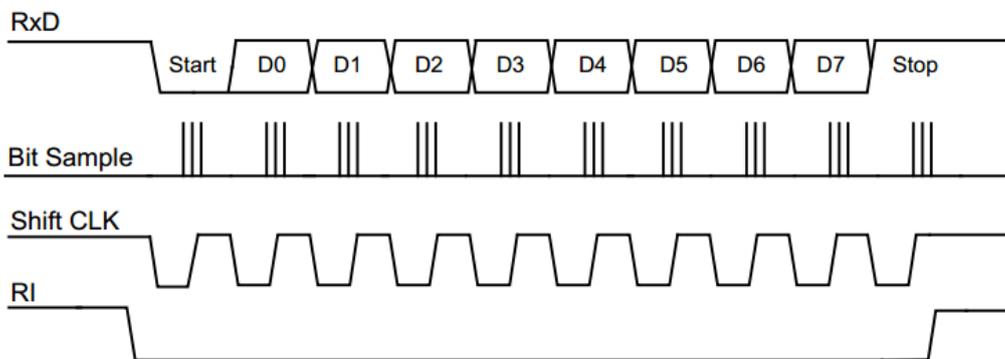
Figure 13 - 6 mode1 data receive timing diagram

13.2.3 Mode2 : 9 bits UART, fixed baud rate, asynchronous full duplex

In this mode, frame is 11 bits by asynchronous full duplex communication. A frame consists of a start bit (logic 0), 8 data bits (low in front), a programmable 9th data bit and one stop bit (logic 1). Mode2 support for multiprocessor communication and hardware address recognition (see multiprocessor communication chapter). At the time of data transmission, the 9th bits (TB8) can be written 0 or 1, for example, it can be written the parity bit P of PSW, or as a multiprocessor communication of data/address flag. When data is received, the 9th data is moved into RB8 and stop bits are not saved. Baud rate selection SMOD bit equal 1/32 or 1/64 of system work frequency. Function block diagram is shown below.
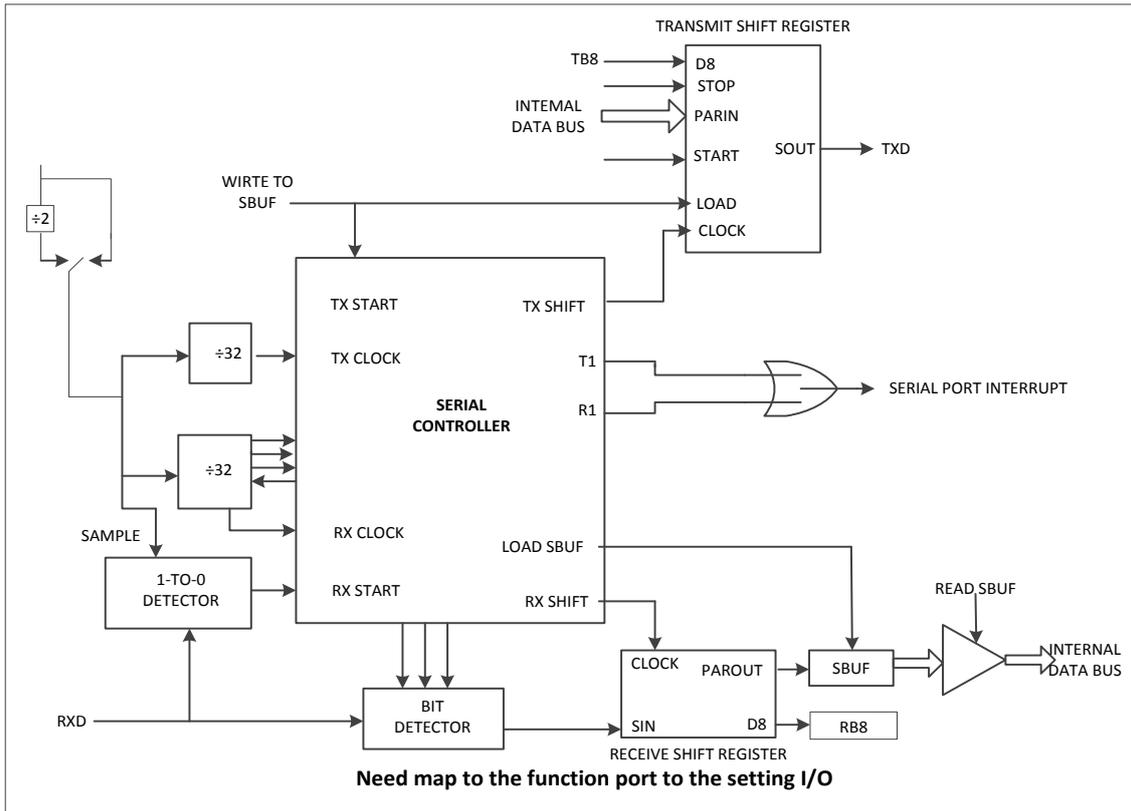
Figure 13 - 7 UART mode2 functional block diagram

Any SBUF write operations as a destination register will start transmission. Meanwhile TB8 is loaded into the sending shift register's 9th bits. Actually sending is started from the system clock after 16 scale frequency counter's next jump. So bit time is synchronous with 16 frequency division counter, but out-sync with SBUF write operation. A Start bit shift out from TXD Pin first, and then shift 9 bits data. After all the 9 bits data in send shift register is sent, the stop bit shift out from TXD Pin, at the same time TI flag set.
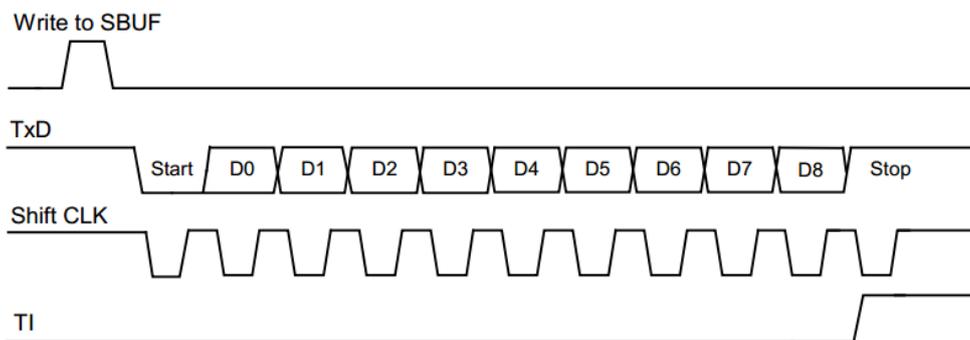
Figure 13 - 8 Mode2 Send the data time series block diagram

Receive is enabled only when REN set 1. UART start to receive data when the falling edge of RXD is detected. The CPU need to sample RXD pin continuously, sampling rate equal 16 times of baud rate. When detecting falling edge, 16 division frequency counter reset immediately to help 16 frequency counter and RXD pin serial data synchronization. 16 frequency counter's every bit time is divided into 16 states, at the 7, 8, and 9 states, the bit detector sample the level on RXD pin. To restrain noise, in this 3 sample states, at least 2 samples data are same，the data will be received. If first received bit is not 0, indicates that this is not a start of frame, this bit is ignored, the receive circuit is reset, wait for the RXD pin of another falling rise. If start bit is valid, then move into the shift register, and then move the other bits to shift register. 9 data bits and 1 stop bit after moving, the data of the shift register and the stop bit is loaded into SBUF and RB8 respectively, RI set 1, but it must meet the following conditions:

(1) RI = 0

(2) SM2 = 0 or 9th received bit= 1

If these conditions are met, then the 9th is loaded into RB8, 8 bits data is loaded into SBUF, RI is set to 1. Otherwise the receive frame is lost.

Among the stop bit, the receiver will return to detect RXD port if there has another falling edge. User must use the software to clear RI, and receive again.
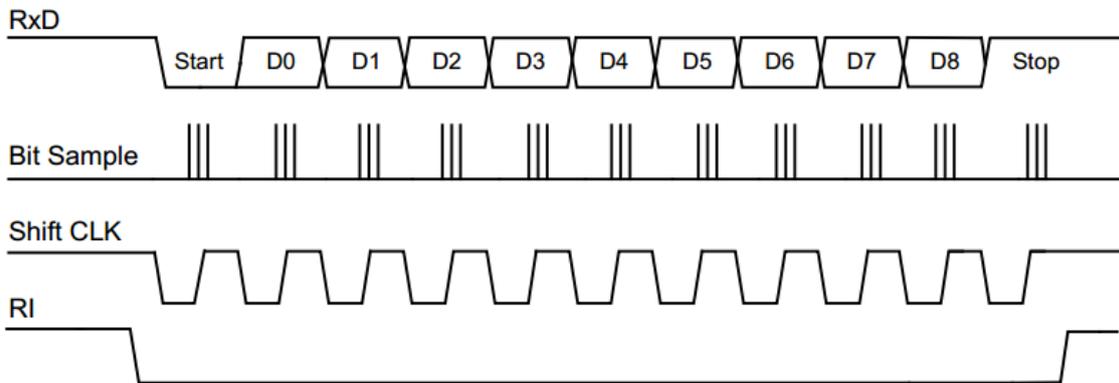
Figure 13 - 9 Mode2 receive data timing diagram

13.2.4 Mode3 : 9 bit UART, variable baud rate, asynchronous full duplex

Mode3 uses the transmission protocols of mode 2 and baud rate generation method of mode1.
Note: When receive data by using interrupt in mode2/3, one time interrupt request will occur several times breaks. Avoid method: Delay some time before cleared RI flag in the interrupt service routine, delay time is up to one of the current length of the communication baud rate at least.
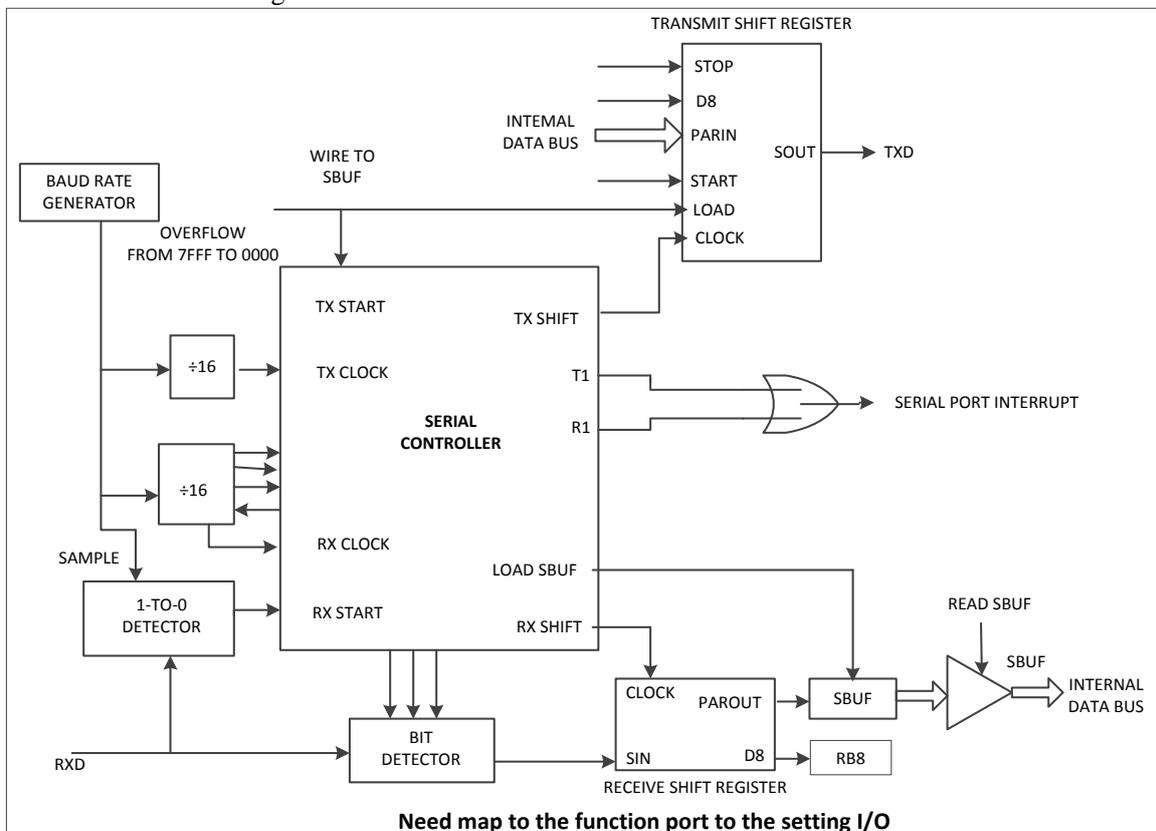
Figure 13 - 10 UART mode3 function block diagram

13.3 Baud rate generator

Timer 4 UART1 is used as the baud rate generator. Select Timer 4 work mode: as the baud rate generator. The mode is similar to automatic reload mode. Overflow of Timer 4 will cause 16 -bit value in Timer 4 reload register load into Timer 4 counter via software, and will generate overflow interrupt. If user does not want to generate interrupt, user can close ET4. The baud rate of UART1 mode1 and mode3 is calculated by the formula as below:

$$BaudRate = \frac{1}{16} \times \frac{f_{T4}/PRESCALER}{65536-[TH4,TL4]}$$ Timer 4 is the baud rate generator.

In the last formula, TH4 and TL4 are data register of Timer 4.

Following table is common Fosc frequency and baud rate corresponding to the Timer 4 count values:

Commonly used baud rate	Fosc		
	4MHz	8MHz	16MHz
1200	FF2F	FE5F	FCBF
2400	FF98	FF2F	FE5F
4800	FFCC	FF98	FF2F
9600	FFE6	FFCC	FF98
19200	FFF3	FFE6	FFCC
38400	/	FFF3	FFE6

Note: UART1 maximum communication baud rate up to 38400Bd.

13.4 Multiprocessor communication

13.4.1 Software address recognition

Mode 2 and mode 3 have applied to communication functions. In both modes, receive data is a 9 -bit data, after the 9th bit data is moved into RB8, next bit is stop. You can set UART: when receiving a stop bit and RB8 = 1, serial port interrupt is valid (request flag RI is set).This moment set SM2, UART work in multiprocessor communication mode.

In multiprocessor communication system, please use the functions as described below. When a host sends a data block to one of several slaves, first send an address byte for addressing the target slave. You can use the 9th bits to distinguish between address byte and data byte, the 9th bit of address byte is 1, and the 9th bit of data byte is 0.

If slave SM2 is 1, it cannot response the interrupt of data byte. Address bytes can enable the interrupt of all slaves, each received address byte is checked by slave, and distinguish whether or not this slave is the target slave. The slave is addressed clear SM2 to 0, and ready to receive incoming data bytes. When finished, once again slave set SM2. The slaves have not been addressed, reserved SM2 bit as 1, do not respond to the data bytes.

Note: in mode1, SM2 is used to detect whether or not the stop bit is valid, if SM2 = 1, and receive interrupt does not respond until it receives a valid stop bit.

13.4.2 Automatic (hardware) address recognition

In mode 2 and mode 3, SM2 is set, UART operation state is as follows: it receives the stop bit, the 9th bit is 1 (address byte), and the data bytes received in accordance with UART slave address, UART generate an interrupt. Slave clear SM2 to 0, the data bytes received subsequently.

The 9th is1 indicates that the byte is address, not data. When a host sends a group of data to one of slaves, it must be sent the target slave address first. All slaves wait to receive the address byte, in order to ensure interrupt occur only when the receiving address byte, SM2 bit must be set to 1. Automatic address recognition is only the address matched can generate interrupt, and comparison is completed by hardware.

After interrupt, the address matched slave clear SM2, continue to receive data bytes. The slave address does not matched is not affected, and will continue to wait to receive its matched address byte. After all information received, the addresses matched slave must set SM2 again, and ignore all non-address bytes transmission, until receiving the next address byte.

When user use automatic address recognition, by calling the appointed slave address the host select one or more slaves for communication. Host uses the broadcast address can address all slaves. There are two special function registers, the slave address (SADDR) and address shield (SADEN). Slave address is an 8 bits byte, saved in the SADDR register. SADEN defines SADDR bits is valid or not, if one bit in SADEN is 0, SADDR corresponding bit is ignored, if one bit in SADEN is 1, SADDR corresponding bit will be used to produce the appointed address. This user can flexible address more than one slaves without changing the slave address in SADDR register.

	From the slave1	From the slave2
SADDR	10100100	10100111
SADEN	11111010	11111001
Contract address	10100x0x	10100xx1
Broadcast address	1111111x	11111111

The slave1 and 2 address lowest bit is different. Slave1 the lowest bit is ignored, and slave2 lowest bit is 1. When only slave1 in communication, the host must send the lowest bit is 0 as address (10100000). Similarly, the slave1 lowest bit is 0, slave2 lowest bit is ignored. Therefore, only slave2 in communication, the host must send the lowest bit is1 as address (10100011). If the host will need to communicate with the two slaves, the bit0 equal 1, bit1 equal 0, bit2 is ignored by the two slaves, and two different addresses for the slave selection (1010 0001 and 1010 0101).

Host uses the broadcast address to communicate with all slaves at the same time. This address is equal bitwise or of the SADDR and SADEN, 0 in result indicates that the bits are ignored. In most cases, the broadcast address is 0xff, the address can be responded by all slaves.

After system reset, SADDR and SADEN registers are initialized to 0, these two results set the appointed address and broadcast address xxxxxxxx (all bits are ignored). By this way the characteristic of communication is removed effectively, and disable the automatic addressing mode. The UART will respond any address, and compatible with the 8051 controller that does not support automatic address recognition. User can implement software address recognition of multiprocessor communication in accordance with the methods above.

13.5 Frame error detection

After 3 error flags are set, only clear to 0 through software, although subsequent frames received without any errors and are not automatically cleared.

13.5.1 Send conflict

If one of the data send is in progress, when user writing data to SBUF, send conflict bit (TXCOL Bit) is set to 1. If send conflict occur, the new data will be ignored, and cannot be written to the send buffer (that do not affect the transmission).

13.5.2 Receive overflow

RI set 1, the data in the receive buffer is not being read, RI is cleared to 0, receive new data again, if user has not read the received data in the buffer before the new data is not received completion (RI set 1), then receive overflow bits (RXROV bit) is set. If receive overflow occur, does not affect the original data in the receive buffer, but subsequent data is lost.

13.5.3 Frame error

If it detects an invalid (low) stop bit, then frames error bit (FE bit) is set to 1.

13.6 UART1 registers

13.6.1 UART1 control register SCON, SCON2

SCON

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	FE	RXROV	TXCOL	REN	TB8	RB8	TI	RI

Bit	Flag	Introductions
7	FE	Frame error detection bit 0 : No frame errors, or software of clearance 0 1 : Frame errors, hardware set 1
6	RXROV	Receive overflow flag 0 : without receiving overflow or software of clearance 0 1 : Receive overflow, hardware set 1
5	TXCOL	Sending conflict flag 0 : No send conflict or software of clearance 0 1 : Send conflict, hardware set 1
4	REN	Serial receive enable control bit 0: Disable serial receive 1: Enables serial receive
3	TB8	In mode2/3, It is the 9th of send data, software set 1 or clear 0
2	RB8	In mode2/3, It is the 9th of send data, as frame flag of a parity bit or address frame/data
1	TI	send interrupt request flags bit 0: Software clear 0 1: In mode0, at the end of sending serial send 8th data, the hardware auto set 1, in other mode, when at start of sending stop bit, hardware set 1
0	RI	receive interrupt request flags bit 0: Software clear 0 1: In mode0, at the end of receiving serial send 8th data, the hardware auto set 1, in other mode, when at start of receiving stop bit, hardware set 1

SCON2

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R	R/W	R	R	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	SMOD	-	UX6	-		SM0	SM1	SM2

Bit	Flag	Introductions
7	SMOD	Double baud rate control bit 0 : In mode2, the baud rate = system clock F_{osc} 1/64 1 : In mode2, the baud rate = system clock F_{osc} 1/32
6	-	Reserved (read = 0b, write invalid)
5	UX6	Serial port mode0 communication speed bit 0 : Serial port mode0 Clock = $F_{osc}/12$ 1 : Serial port mode0 Clock = $F_{osc}/2$
4-3	-	Reserved (read = 0b, write invalid)
2-1	SM0:SM1	Serial mode, see the following table

0	SM2	Multiprocessor communication enable control bit (9th bit "1" validator) 0 : In mode1, does not detect stop bit, set RI whatever stop bit is 0 or 1 In mode2 and 3, not detect 9th bit, set RI whatever any bytes 1 : In mode1, enable stop checked, only a valid stop bit= "1" can set RI In mode2 and 3, only the address byte (9th bit ="1") can set RI

SM0	SM1	Work method	Function description	Baud rate
0	0	0	Synchronous shift transfer serial **mode**: shift register	When UX6 = 0, the baud rate is $F_{osc}/12$ When UX6 = 1, the baud rate is $F_{osc}/2$
0	1	1	8 bit UART, variable baud rate	Timer4 overflow rate /16
1	0	2	9 bit UART	$(2^{SMOD}/64)\times F_{osc}/2$
1	1	3	9 bit UART, variable baud rate	Timer4 overflow rate /16

13.6.2 UART1 data buffer register SBUF

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	SBUF[7:0]							

Bit	Flag	Introductions
7-0	SBUF[7:0]	Serial buffer register Write as the sent data needed, read as the received data

13.6.3 UART1 automatic address recognition SADDR, SADEN

Slave address register SADDR

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	SADDR[7:0]							

Bit	Flag	Introductions
7-0	SADDR[7:0]	Slave address register

Slave address mask register SADEN

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	SADEN [7:0]							

Bit	Flag	Introductions
7-0	SADEN [7:0]	slave address mask register

13.7 UART2

UART2 control and work with UART1 The same register please refer to UART1
Differences:

1. UART2 registers deposited in the extended SFR；
2. UART2 Only two works;
3. UART2 No error detection;
4. UART2 No automatic hardware address recognition.

13.7.1 UART2

13.7.1.1 Mode0 : 8 bit UART variable baud rate that asynchronous full duplex

Mode0 provides 10 bits full duplex asynchronous communication, 10 bits consist of a start bit (logical 0), 8 data bits (low bit first) and one stop bit (logic 1). When receiving, 8 data bits stored in SBUF and stop bit stored in RB8. Mode1 baud rate equals Timer 4 overflow rate /16.

Any S2BUF write operations as a destination register will start transmission. Actually sending is started from the system clock after 16 scale frequency counter's next jump. So bit time is synchronous with 16 frequency division counter, but out-sync with S2BUF write operation. Start bit shift out from TXD Pin first, and then shift 8 bits data. After all the 8 bits data in send shift register is sent, the stop bit shift out from TXD Pin, at the same time TI flag set.
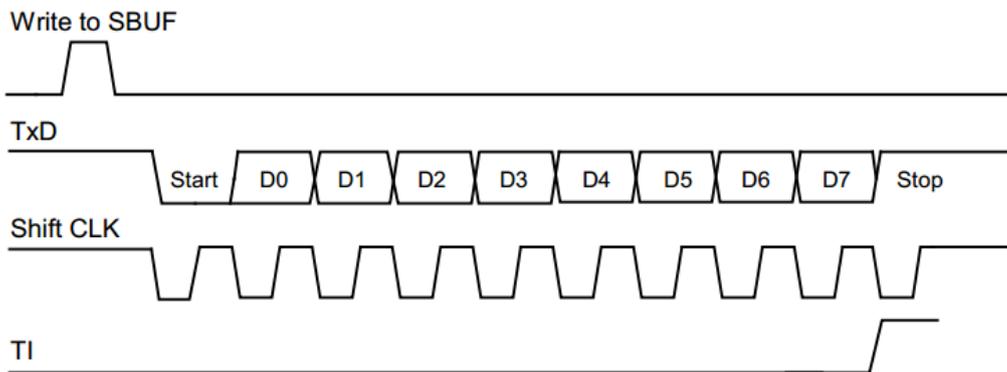
Figure 13 - 11 Send timing of mode0

Receive is enabled only when REN set 1. UART start to receive data when the falling edge of RXD is detected. The CPU need to sample RXD pin continuously, sampling rate equal 16 times of baud rate. When detecting falling edge, 16 division frequency counter reset immediately to help 16 frequency counter and RXD pin serial data synchronization. 16 frequency counter's every bit time is divided into 16 states, at the 7, 8, and 9 states, the bit detector sample the level on RXD pin. To restrain noise, in this 3 sample states, at least 2 samples data are same，the data will be received. If first received bit is not 0, indicates that this is not a start of frame, this bit is ignored, the receive circuit is reset, wait for the RXD pin of another falling rise. If start bit is valid, then move into the shift register, and then move the other bits to shift register. 8 data bits and 1 stop bit (stop bits contain errors, as described in the description of register SM2) ,after moving, the data of the shift register and the stop bit (stop bits that contain errors) is loaded into SBUF and RB8   respectively,RI set 1, but it must meet the following conditions:

(1) RI = 0
(2) SM2 = 0 don't judge stop bit or SM2=1 judge stop bit, and stop bit must be 1

If these conditions are met, then stop bit (contain the error stop bit) id loaded into RB8, 8 data is loaded into SBUF, RI is set to 1. Otherwise the receive frame is lost. At this time, the receiver will return to detect RXD port if there has another falling edge. User must use the software to clear RI, then receive again.
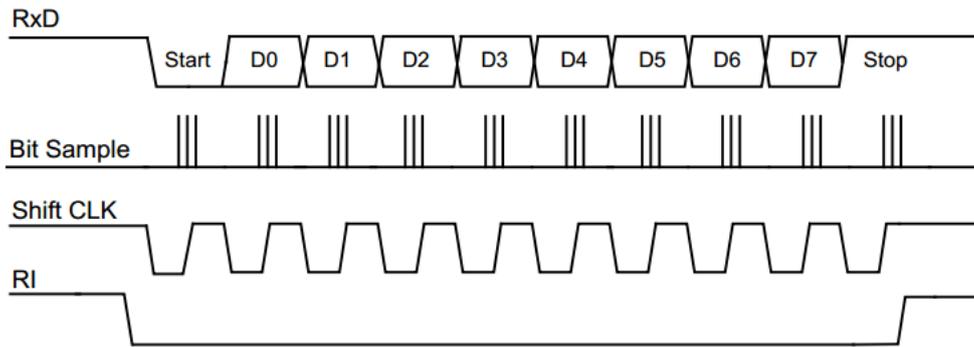
Figure 13 - 12 Receive timing of mode0

13.7.1.2 Mode1 : 9 bit UART variable baud rate, asynchronous full duplex

In this mode, frame is 11 bits by asynchronous full duplex communication. A frame consists of a start bit (logic 0), 8 data bits (low in front), a programmable 9th data bit and one stop bit (logic 1). Mode1 support multiprocessor communication. At the time of data transmission, the 9th bits (TB8) can be written 0 or 1, for example, it can be written the parity bit P of PSW, or as a multiprocessor communication of data/address flag. When data is received, the 9th data is moved into RB8 and stop bits are not saved.

Any SBUF write operations as a destination register will start transmission. Meanwhile TB8 is loaded into the sending shift register's 9th bits. Actually sending is started from the system clock after 16 scale frequency counter's next jump. So bit time is synchronous with 16 frequency division counter, but out-sync with SBUF write operation. A Start bit shift out from TXD Pin first, and then shift 9 bits data. After all the 9 bits data in send shift register is sent, the stop bit shift out from TXD Pin, at the same time TI flag set.
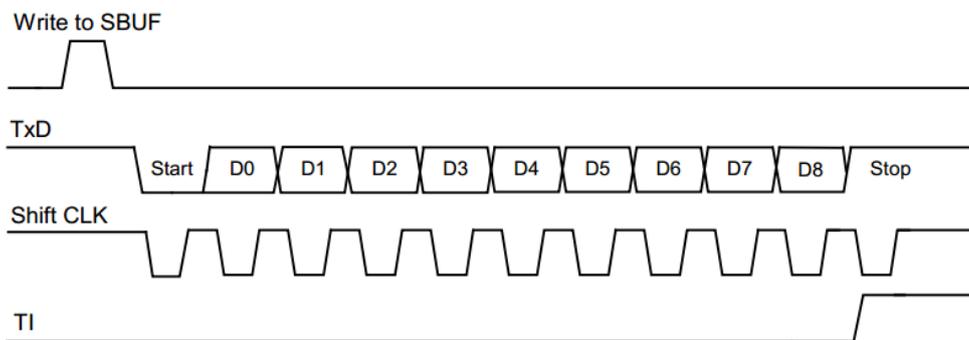
Figure 13 - 13 Send Timing of mode1

Receive is enabled only when REN set 1. UART start to receive data when the falling edge of RXD is detected. The CPU need to sample RXD pin continuously, sampling rate equal 16 times of baud rate. When detecting falling edge, 16 division frequency counter reset immediately to help 16 frequency counter and RXD pin serial data synchronization. 16 frequency counter's every bit time is divided into 16 states, at the 7, 8, and 9 states, the bit detector sample the level on RXD pin. To restrain noise, in this 3 sample states, at least 2 samples data are same，the data will be received. If first received bit is not 0, indicates that this is not a start of frame, this bit is ignored, the receive circuit is reset, wait for the RXD pin of another falling rise. If start bit is valid, then move into the shift register, and then move the other bits to shift register. 9 data bits and 1 stop bit after moving, the data of the shift register and the stop bit is loaded into SBUF and RB8 respectively, but it must meet the following conditions:
(1) RI = 0
(2) SM2 = 0

If these conditions are met, then the 9th is loaded into RB8, 8 bits data is loaded into SBUF. But need to detect stop bit, only stop bit is 1, RI can be set, if stop bit is 0,RI will not be set.
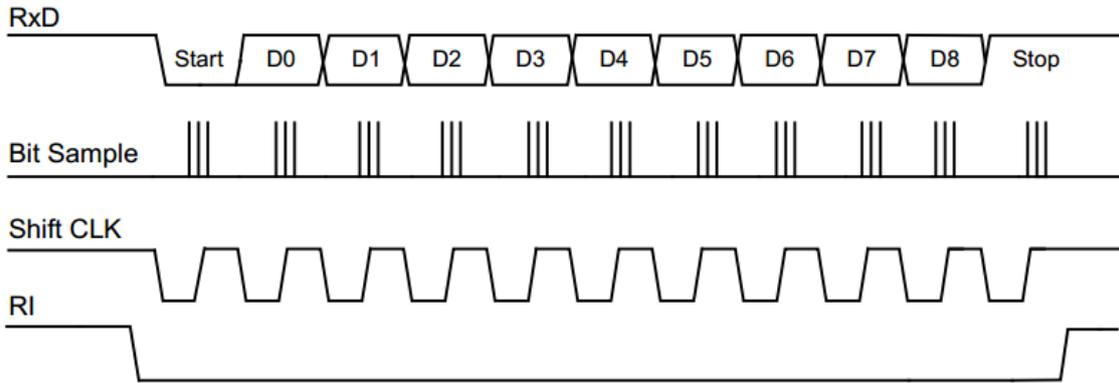
Figure 13 - 14 Receive timing of mode1

13.7.2 UART2 control register S2CON, S2CON2

S2CON

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	FE			REN	TB8	RB8	TI	RI

Bit	Flag	Introductions
7	FE	Frame error detection bit 0 : No frame errors, or software of clearance 0 1 : Frame errors, hardware set 1
6-5	-	Reserved (read = 0b, write invalid)
4	REN	Serial receive enable control bit 0: Disable serial receive 1: Enables serial receive
3	TB8	In mode1, It is the 9th of send data, software set 1 or clear 0
2	RB8	In mode1, It is the 9th of send data, as frame flag of a parity bit or address frame/data
1	TI	send interrupt request flags bit 0: Software clear 0 1 : when at start of sending stop bit, hardware set 1
0	RI	receive interrupt request flags bit 0: Software clear 0 1 : when at start of serial receiving stop bit, hardware set 1

S2CON2

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-						SM1	SM2

Bit	Flag	Introductions
7-2	-	Reserved (read = 0b, write invalid)
1	SM1	0 : 8 UART the Timer 5 overflow rate /16 1 : 9 UART the Timer 5 overflow rate /16 Note: 1, UART2 Sets the baud rate as described in Timer 5 1: UART2 Baud rate generator Chapters.

		2, UART2 maximum communication baud rate can be achieved 38400Bd .

0	SM2	The 9th bit detection enable bit
		0 : In mode0, does not detect stop bit, set RI whatever stop bit is 0 or 1
		   In mode1, not detect 9th bit, set RI whatever the 9th bit is 0 or 1
		1 : In mode0, enable stop checked, only stop bit is 1 can set RI
		   In mode1, only the 9th bit is1 can set RI

13.7.3 UART2 data buffer register S2BUF

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	S2BUF[7:0]							

Bit	Flag	Introductions
7-0	S2BUF[7:0]	Serial buffer register Write as the sent data needed, read as the received data

14 Serial peripheral interface SPI

14.1 SPI characteristics

➢ Full duplex, three/Four-wire synchronous transmission
➢ Master and slave operation
➢ Level programmable master clock frequency
➢ Polar and phase programmable serial clock
➢ Selectable data transfer direction
➢ Write conflict and receive the overflow flag
➢ MCU interrupt main mode conflict detection
➢ MCU interrupt transmission end flag
➢ Host mode supports up to 8Mbps transmission rate (F_{osc}=32MHz), slave mode speed must equal F_{osc} /16 or below F_{osc} /16

14.2 SPI signal description

Master output and slave input (MOSI): the signal connected master and a slave, data from master serial sent to the slave by MOSI, and master output, slave input.

Master input and slave output (MISO): the signal connected master and a slave, data from slave serial sent to the master by MISO, and slave output, master input. When the device is slave and has not been selected, MISO pin of slave in a high impedance state.

Serial clock (SCK): the signal used for control MOSI MISO synchronous operations of the input and output data, each 8 clock cycles MOSI and MISO transmits a byte, if the salve is not selected, SCK signal will be ignored. Note: only the master device can generate the SCK signal.

Slave device select pin ($\overline{SS}$): each slave devices is selected by pin ($\overline{SS}$). When the signal is low level, indicating that the slave is selected. Master can control the pin ($\overline{SS}$) port level of slave device by software to select each of slaves, clearly that only a master device can drive total transmission network. In order to avoid the MISO bus conflict, only enable one slave device to communicate with master device at the same time. In master **mode**, the ($\overline{SS}$) Pin state is associated MODF flag in SPI state register SPSTAT to avoid more than one master device driver MOSI and SCK.

The following conditions, ($\overline{SS}$) pin can be used as normal port or other functions:

(1) Device as the master equipment, SSIG flag in SPI control register SPCTL is set to 1. This configuration only support one master device in the communication network, therefore, the MODF flag in SPI state register SPSTA will not be set to 1.

(2) Device as the slave device, CPHA and SSIG flags in SPI control register SPCTL are set to 1. This configuration only support one master and one slave device in the communication network, therefore, the device are always selected, master device does not need to control the slave device ($\overline{SS}$) pin as the communication goal.

When the slave device ($\overline{SS}$) pin is enabled, other devices can enable the pin to maintain a low level to select the device. In order to avoid the MISO bus conflict, in principle, don't enables two or more devices are selected.

When the master device ($\overline{SS}$) pin is enabled, If ($\overline{SS}$) is pulled down will set the mode error flag MODF (interrupt), and MSTR bit will also be cleared to 0, the device will be switched to slave device compulsorily.

When MSTR = 0 (slave model) and CPHA = 0, SSIG must be 0, because the data transmission need cooperation with ($\overline{SS}$) pin at this time.

14.3 SPI clock rate

In the master mode, SPI transmission rate have 4 levels, namely the internal clock 4, 16, 64, 128 frequency division, user can select by SPR[1:0] bit in SPCTL register.

14.4 SPI functional block diagram
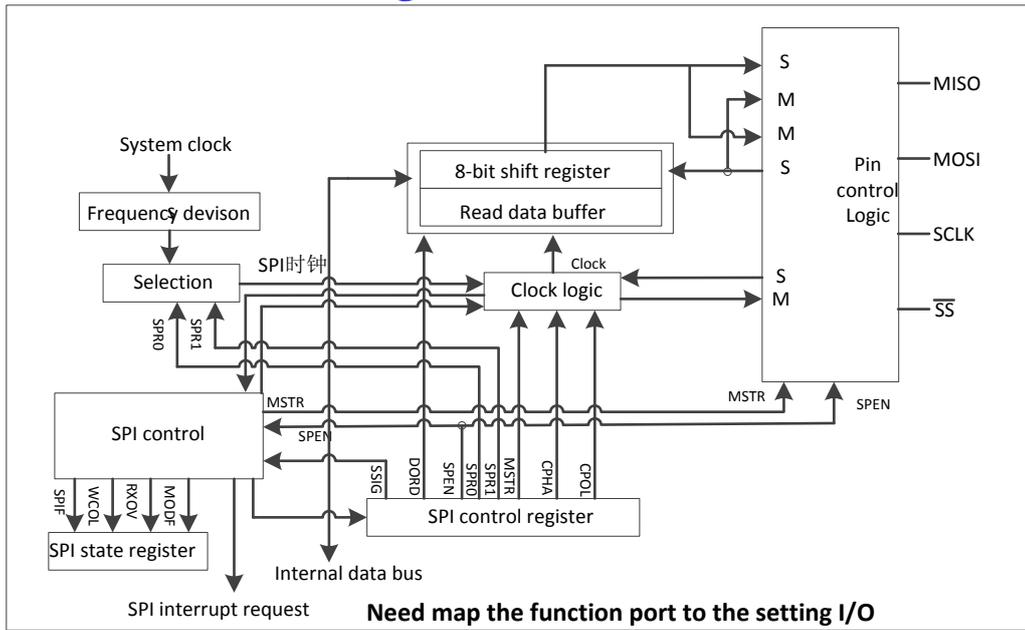
Figure 14 - 1 SPI functional block diagram

14.5 SPI work mode

SPI can be configured in master mode or slave mode. SPI module configuration and initialization by setting the register related. Further setting that is used to complete the data transfer.

During SPI communication, data is moved serial in and out synchronously, serial clock (SCK) is used to keep data movement and sample synchronization on two serial data lines (MOSI & MISO). The slave device ($\overline{SS}$) pin can be selected slave device independently, if the device is not selected, user cannot participate in the SPI activity on the bus.

When SPI master device transmits data to the slave device by MOSI, as response the slave device send data to master device by MISO, and achieve the data at the same clock sending and receiving of synchronous full duplex transmission. Send shift register and receive register use the same SFR address, the write operation of SPI data register SPDAT will write into send shift register, the read operation will get the receive shift register data.

Note: the data written does not affect the read data needed.
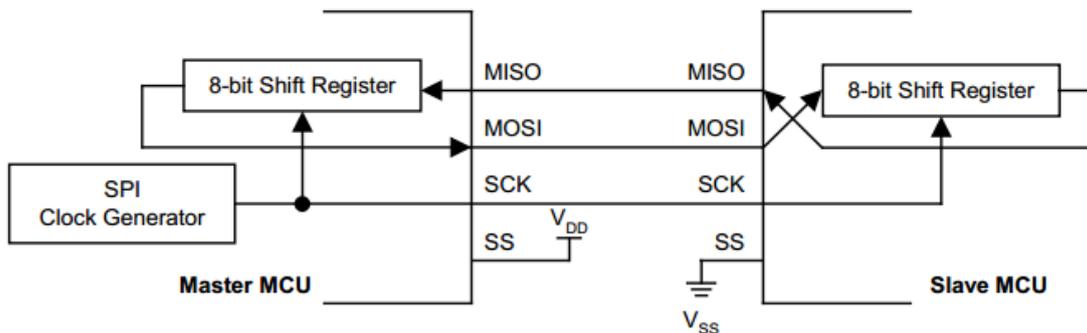
Figure 14 - 2 Full-duplex master/slave interconnect diagram

Master mode

(1) Mode startup

SPI master control the startup of all data transfer on the SPI bus. Only one master device can enable transfer on one SPI bus.

(2) Send

SPI master mode, when write a byte of data to the SPI data register SPDAT, data will be written to the send shift buffer. If one data already in the send shift register or is being transferring, SPI will generates a

WCOL signal to indicate that writing is too fast. But the data in send shift register will not be affected, send is not disrupted.

(3) Receive

When SPI master device transmits data to the slave device by MOSI, via MISO pin, data in sent shift register of it can also be transfer to the receive shift register of the master device, and achieve full-duplex operation. So SPIF flag is set to 1 indicates the data sent completed and the data received is complete also. The SPI module is receive double buffer, that is to say, data can be read out after SPIF is set to 1, but it must be read out before the next byte of data receive completed, otherwise it will reset the receiver overflow flag RXOV, If receive the overflow occur, subsequent data will not be moved into the receive registers, when receive overflow, SPIF could properly set to 1.

Slave mode

(1) Mode start up

Set MSTR to 0 (If ($\overline{SS}$) is enabled it must be pulled low), the device run in slave **mode**, **mode** cannot be changed during data transfer (($\overline{SS}$) pins must maintain low level), or the data transfer will fail (SPIF will not be set to 1).

(2) Send

SPI slave the device cannot start the data transfer, so SPI salve device must write the data is transmitted to master into send shift register before the master starts a new data transfer of data. If they are not data be written to a send shift register before sending, slave device will transfer data "0x00" to the master device. If the shift register has data when writing data (or in transmitting), the WCOL flag of SPI slave device will be set to 1, indicates the SPDAT writing is conflicted. But the data in shift register will not be affected, transmission is not disrupted. SPIF will be set to 1 when transfer is done.

(3) Receive

IN Slave mode, it is controlled by SCK signal of master device, data shift via MOSI, when the counter count SCK Edge to 8, represents a byte of data is received, SPIF will be set to 1, data can be read from SPDAT register, but it must be read out before next data receive completion, otherwise the receiver overflow flag RXOV will be set, if receiving overflow has occurred, subsequent data will not be moved into the receive registers, when receive overflow, SPIF could properly set to 1.

14.6 SPI transfer form

By software setting the CPOL and CPHA bit in register, the user can choose SPI the four combinations of clock polarity and phase. CPOL bit define clock polarity and that the level of free time. CPHA bit define clock phase, as define the sampling clock edge that enables data transfer. In two master and slave devices communication, clock polarity and phase settings should keep consistent.
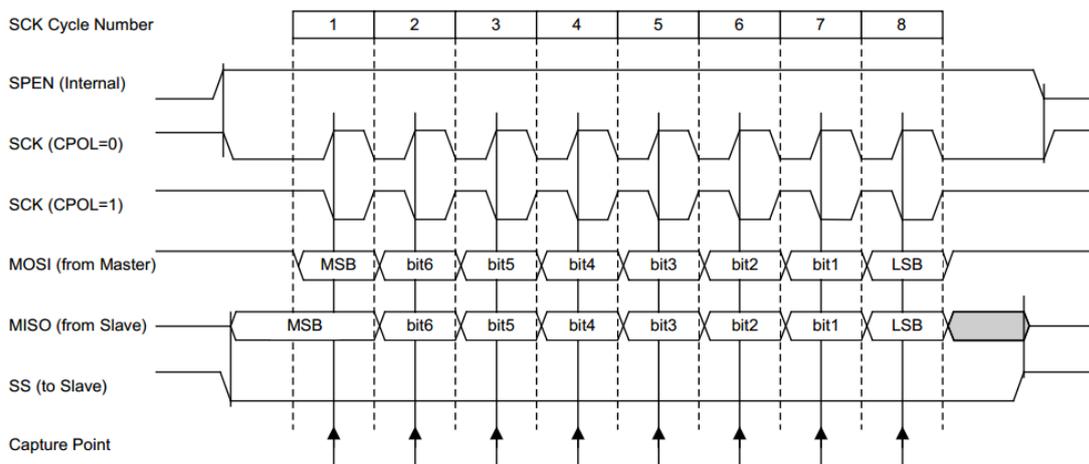

Figure 14 - 3 Data transfer form (CPHA=0)

If CPHA = 0, data is captured at the SCK first edge, so the slave device must be ready before SCK first edge, so the slave device start to sample data from the falling edge of ($\overline{SS}$) pin. ($\overline{SS}$) pin must be pulled high after one byte transmit every time, and be pulled down before sending the next byte again, so when CPHA = 0, SSIG is not valid, that is to say, ($\overline{SS}$) pin is forced to enabled.
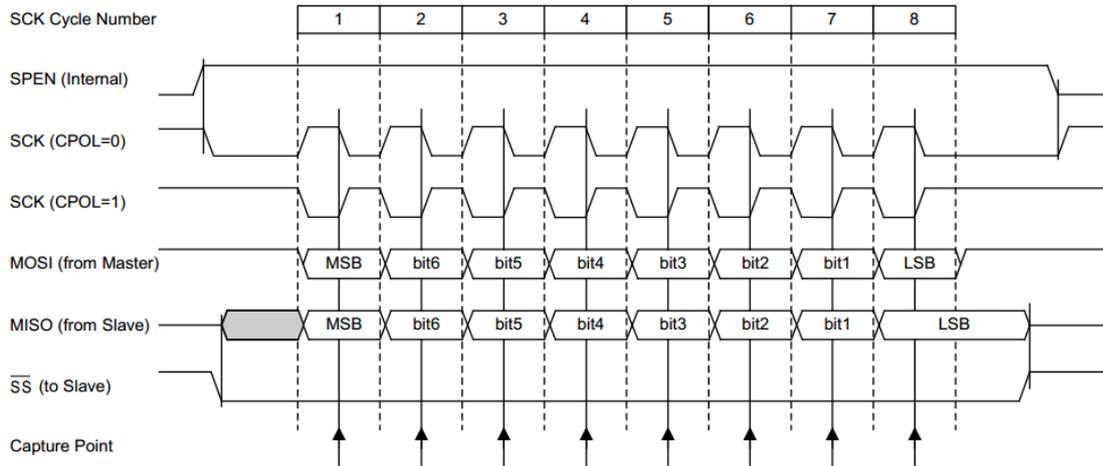
Figure 14 - 4 Send form data (CPHA=1)

If CPHA = 1, Master device output data to MOSI at SCK first edge, the slave device keep the SCK first edge as a start signal. User must complete the SPDAT write operation during first 2 edges of first clock,. Transfer each other modes cannot be changed, or the sending and receiving of data will fail, the mode changed of register data (send data), and state (receive empty) are unchanged. This form of data transfer is the first forms of a single between master-slave communication devices.
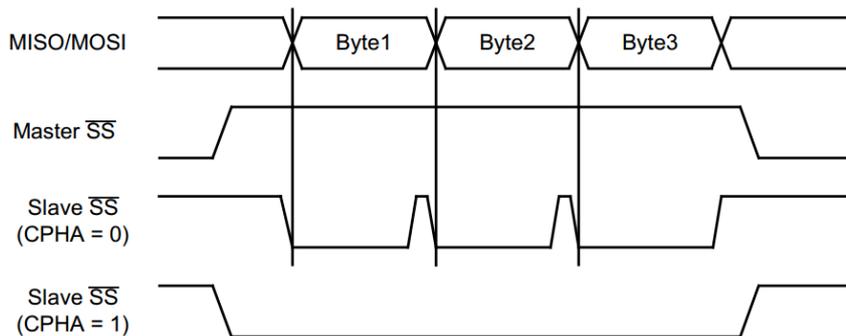
Figure 14 - 5 CPHA/($\overline{SS}$) timing sequence

14.7 SPI Error detection

The flags in SPSTA register define SPI communication errors:

(1) Mode fault (MODF)

Mode fault in SPI master mode shows ($\overline{SS}$) Pin level is inconsistent with the actual mode of device, MODF flag will be set to 1 (can occur interrupt), and indicates there have multi-master devices conflict in SPI control system, this moment the hardware clear the SPEN bit automatically, namely close SPI modules first and hardware clear MSTR bits automatically. If need to restart the SPI module, MODF must be written 1 and cleared 0 by software, then set SPEN.

(2) Write conflict (WCOL)

When data is not being sent or in sending, continue write data operation to SPDAT will cause a write conflict, WCOL bit will be set to 1, but sending does not terminate. Need software write 1 and clear 0.

(3) Receive the overflow (RXOV)

When before the completion receive of the second data, if previous received data SPIF flag has not been cleared, the receive overflow flag RXOV will be set, when SPIF is set to 1, later received data will not be saved in register, so before saving the receive data into SPDAT, SPIF must be cleared, RXOV need software write 1 and clear 0.

14.8 SPI interrupt

wo SPI state flags SPIF&MODF can generate a CPU interrupt request.

Serial data transmission completion flag SPIF: hardware set to 1 after one byte of data sent/receive is

completed.

Fault mode flag MODF: the bit is set to 1 is the device mode (master) is inconsistent with ($\overline{SS}$) pin levels, SSIG bit is 1 (($\overline{SS}$) has not been enabled), no MODF interrupt request.
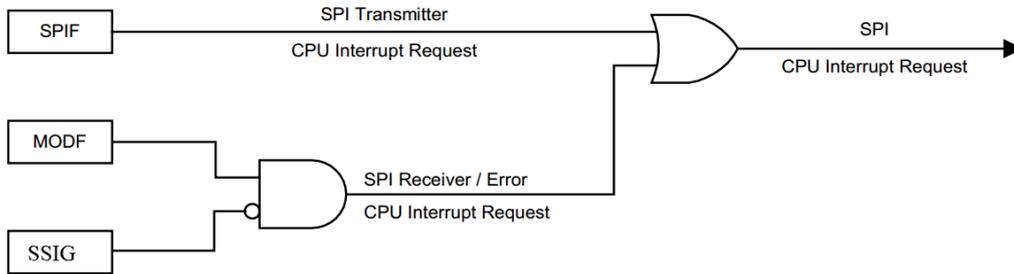

Figure 14 - 6 SPI interrupt request generation

14.9 SPI configuration table

SPEN	SSIG		MSTR	Master or slave mode	MISO	MOSI	SCK	Notes
0	x	I/O	x	SPI function disable	I/O	I/O	I/O	SPI disable
1	0	0	0	Slave mode	Output	Input	Input	Select slave
1	0	1	0	Slave mode not selected	High impedance	Input	Input	Not selected. MISO is high impedance to avoid bus conflict
1→0	0	0	1→0	Close SPI	Output	Input	Input	SS configured as input, SSIG is 0 . If SS is driven as low level, the device is selected as slave. This moment MSTR clear 0 and set the mode error flag MODF, and it can be used to interrupt request.
1	0	1	1	Master (free)	Input	High impedance	High impedance	When the master is idle, MOSI and SCK are high impedance state to avoid a bus conflict. User must pull up or pull down SCK (according to CPOL value) to avoid SCK in floating.
				Master (active)		Output	Output	As a master in active, the MOSI and SCK are push-pull output.
1	1	I/O	0	Slave	Output	Input	Input	CPHA Cannot be 0
1	1	I/O	1	Master	Input	Output	Output	-

14.10 SPI registers

14.10.1 SPI control register SPCTL

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	SSIG	SPEN	DORD	MSTR	CPOL	CPHA	SPR[1:0]	

Bit	Flag	Introductions
7	SSIG	($\overline{SS}$) pin enable bit 0 : ($\overline{SS}$) pin is used to determine the device is master or slave 1 : MSTR determine the device is master or slave, ($\overline{SS}$) pin as normal I/O
6	SPEN	SPI enable bit 0 : Disable SPI module, related pins are general I/O (recommended I/O set high impedance) 1 : Enable SPI module, related pins are SPI communication pins
5	DORD	Transfer direction selection bit 0 : MSB send first 1 : LSB send first
4	MSTR	Master/slave mode selection bit 0 : Slave mode 1 : Master mode
3	CPOL	SPI Clock polarity selection bit 0 : Low level when SCK idle 1 : High level when SCK idle
2	CPHA	SPI Clock phase selection bit 0 : Data sample at the SPI the first edge of the clock 1 : Data sample at the SPI the second edge of the clock Note: When SSIG = 0&CPHA = 0, at ($\overline{SS}$) low the data is driven; and when CPHA = 1, the data is driven at the edge of the previous SCK.
1-0	SPR[1:0]	SPI clock rate selection control bit 00 : F_{osc} /4 01 : F_{osc} /16 10 : F_{osc} /64 11 : F_{osc} /128

14.10.2 SPI state register SPSTAT

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R	R	R	R
Reset values	0	0	0	0	0	0	0	0
Flag	SPIF	WCOL	RXOV	MODF	-			

Bit	Flag	Introductions
7	SPIF	SPI transfer complete flag 0 : Software write 1 clear 0 1 : One time transfer is completed, the hardware set 1,and also as interrupt request flag
6	WCOL	SPI write conflict flag 0 : Software write 1 clear 0 1 : Implement SPDAT writing operation during transfer, hardware set 1, (the data

5	RXOV	SPI receive overflow flag 0 : Software write 1 clear 0 1 : Received overflow occur, hardware set 1 Note: receive is double BUFF and receive overflow occur at the time of previous received data SPIF flag has not been cleared before the completion receive of the second data. So SPIF must be cleared before ready to receive next data every time, otherwise RXOV will be set to 1, it does not affect the SPI receive.
4	MODF	Fault mode flag 0 : Software write 1 clear 0 1 : When ($\overline{SS}$) pin level is inconsistent with SPI mode, hardware set 1 (and switch to slave mode immediately), as interrupt request flag.
3-0	-	Reserved (read = 0b, write invalid)

14.10.3   SPI data register SPDAT

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	SPDAT[7:0]							

Bit	Flag	Introductions
7-0	SPDAT[7:0]	SPI data register

15 Analog to digital converter ADC

15.1 ADC characteristics

➢ Up to 8 external channels and 2 internal channels (include GND) 12/10 bits ADC detection
➢ Optional internal reference voltage 2V,3V,4V,VDD and external Vref
➢ Optional convert data align orientation
➢ Optional convert data bit
➢ ADC Conversion complete interrupt
➢ Single channel (P0.2 port) ADC wakeup interrupt
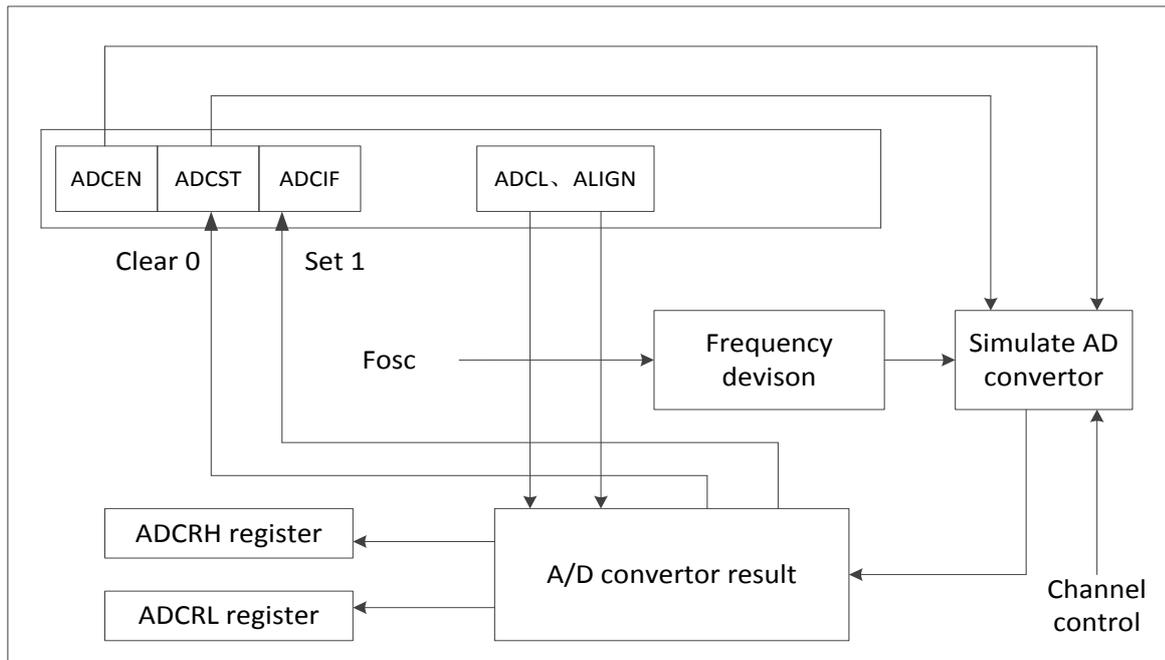
Figure 15 - 1 ADC functional block diagram

15.2 ADC power saving wakeup

After chip enters IDLE or PD mode, ADC power saving wakeup function can wake chip from the mode, operation as below:

1. ADC wakeup control register configuration can enable wakeup
2. Configure P0.2 as an analog channel
3. Configure the wakeup resistance by select P0.2 pull-up resistor register
4. Enter PD mode
5. When the key is pressed, if the voltage on port is less than 4.2V (@VDD=5V),Chip will be woken up from power-down mode, and set AMWIF Flag, ADC interrupt occur if the interrupt enabled
6. After wake up, turn off the wakeup module and output a high level on P0.2, R1 is used to equivalent internal wakeup resistance
7. Open ADC, sampling the voltage on wakeup channel, then distinguish the different buttons depend on different voltage.
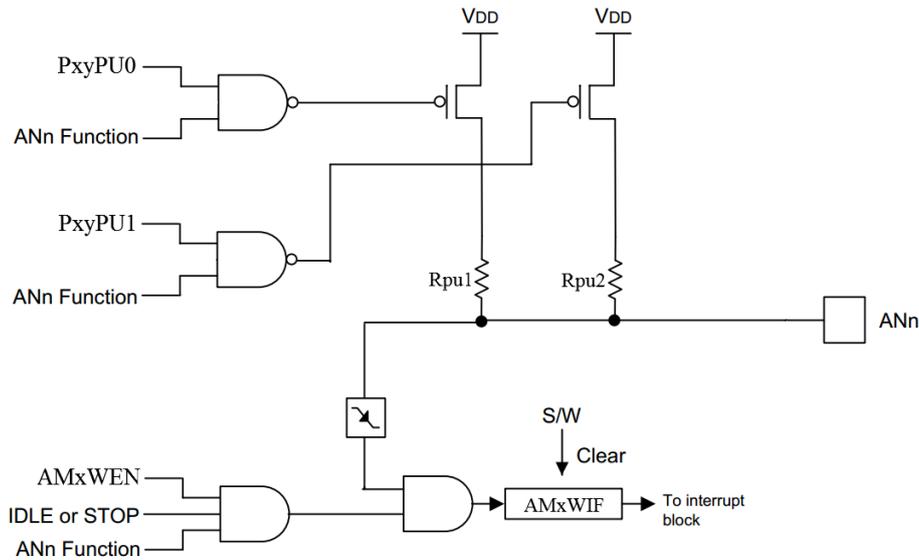
Figure 15 - 2 ADC power saving wakeup function block diagram
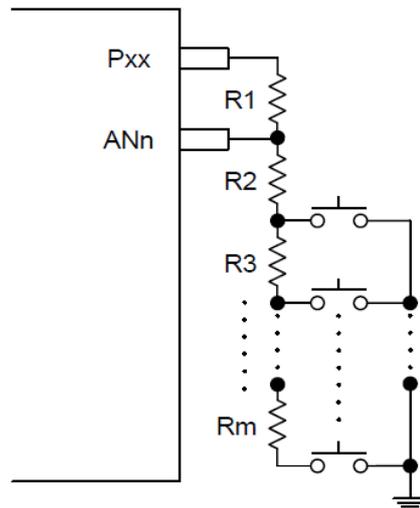
Figure 15 - 3 ADC series resistor key input application reference circuit

15.3 ADC registers

15.3.1 ADC control register ADCC0,ADCC1,ADCC2

ADCC0

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	1	1
Flag	ADCEN	ADCST	ADCIF	-	VREFO	VREFS	INREF_S[1:0]	

Bit	Flag	Introductions
7	ADCEN	ADC module power control bit 0 : Close ADC conversion power 1 : Open ADC conversion power Note: 1. ADCEN set 1 or after switch conversion channel, recommended start ADC

		Conversion after delay some time.
		2. In power-down mode, ADCEN force to 0 .
		3. When start ADC conversion, ADC power saving wakeup function must be closed.
6	ADCST	ADC start control bit 0 : After the conversion is complete, hardware clear 0 automatically, during the conversion, software clear 0 will stop the conversion. 1 : Start conversion Note: ADCIF need to clear 0 before start conversion, when ADCIF equal 1, set ADCST cannot start a new conversion.
5	ADCIF	ADC interrupt flag 0 : No ADC conversion interrupt 1 : After conversion, hardware set 1, can be used for interrupt request (must be software clear 0)
4	-	Reserved (read = 0b, write invalid)
3	VREFO	VREF output enable bit 0 : VREF no output 1 : From P0.4 output VREF. (this moment P0.4 mast be set analog input, and VREFS Must be 0)
2	VREFS	VREF selection bit 0 : Select internal VREF 1 : Select external VREF (this moment P0.4 as ADC reference voltage input only, and port must be set analog input)
1-0	INREF_S	ADC internal reference voltage selection bit 00: VDD 01 : Internal 4V 10 : Internal 3V 11 : Internal 2V Note: When the internal Vref selection for 2V, VDD must above 2.7V; selection for 3/4V, VDD must above the internal Vref+ 0.5V.

ADCC1

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	R	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	ICHS[1:0]		-	-	XCHS[3:0]			

Bit	Flag	Introductions
7-6	ICHS[1:0]	ADC internal input channel selection bits 00 : Disable internal channel 01 : 1/4VDD as ADC input channel 10 : Reserved 11 : GND Note: when internal channel selection, external channel selection XCHS[3:0] must be configured to 1111b, otherwise internal and external channel may be opened at the same time.
5-4	-	Reserved (read = 0b, write invalid)
3-0	XCHS[3:0]	ADC external input channel selection bits XCHS[3:0] = x(x = 0,2... 9), x defines the current test channel as ANx, such as XCHS[3:0] = 3, the current test channel is external channel AN3. Except external channel must be set XCHS[3:0], corresponding Pin need be set analog input.

ADCC2

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	ADCL	ALIGN	ADCTS [2:0]			ADCS[2:0]		

Bit	Flag	Introductions

7	ADCL	ADC conversion data length control bit
		  0 : ADC conversion result is 12 bit data
		  1 : ADC conversion result is 12 bit data (get 12 bits high 10 bits)

6	ALIGN	ADC data alignment direction control bit
		  0 : ADC conversion results high 8 bits put in ADCRH registers, low 4 bits put in ADCRL register low 4 bits (or low 2 bits)
		  1 : ADC conversion results high 4 bits (Or high 2 bits)put in ADCRH register low 4bits, low 8 bits put in ADCRL register

5-3	ADCTS [2:0]	When ADC clock is 4MHZ configure the 3bits to 000b, one time conversion needs 22 ADC_CLK When ADC Clock is 2MHZ&1MHZ, configure the 3bits to 001b or 010b, one time conversion needs 19 ADC_CLK When ADC clock <1MHZ, configure the 3bits to 011b/100b/101b/110b/ 111b, one time conversion needs 15 ADC_CLK Note: in order to ensure ADC accuracy, recommended ADC conversion frequency at 2MHz or below 2MHz.

2-0	ADCS[2:0]	ADC clock selection bit
		  000 : F_{osc}/2
		  001 : F_{osc}/4
		  010 : F_{osc}/6
		  011 : F_{osc}/8
		  100 : F_{osc}/12
		  101 : F_{osc}/16
		  110 : F_{osc}/24
		  111 : F_{osc}/32

15.3.2 ADC conversion result register ADCRL, ADCRH

ADCRL

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	ADCRL[7:0]							

ADCRH

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	ADCRH[7:0]							

Bit	Flag	Introductions
7-0	ADCRH[7:0]	When ALIGN = 0 ADCRH[7:0] is ADC conversion high 8 bits, ADCRL[3:0] is ADC conversion of

7-0	ADCRL[7:0]	low 4/2 bits When ALIGN = 1 ADCRH[3:0] is ADC conversion high 4/2 bits, ADCRL[7:0] is ADC conversion of low 8 bits

Start ADC conversion steps:

(1) Enable ADC module;

(2) Select analog input channel, voltage reference, conversion clock, conversion result align orientation;

(3) Set 1 ADCST to start ADC conversion;

(4) Waiting for ADCST = 0 or ADCIF = 1, if ADC interrupt is enabled, ADC interrupt will be generated, user need to clear ADCIF by software;

(5) Get conversation data from ADCRH/ADCRL;

(6) Repeat steps 3-5 to start another conversion.

Note: in order to ensure ADC works reliably, VDD Operating voltage must above 2.7V.

15.3.3 ADC wakeup control register ADCWC

ADCWC

Bit	7	6	5	4	3	2	1	0
R/W	W	W	W	W	W	W	W	W
Reset values	0	0	0	0	0	0	0	0
Flag	AMWEN	AMWIF	-					

Bit	Flag	Introductions
7	AMWEN	ADC wakeup module enable bit 0 : Disable ADC wakeup module 1 : Enable ADC wakeup module Note: when enable ADC wake up module, ADC must be closed.
6	AMWIF	ADC wakeup module interrupt flag(the same interrupt vector with the ADC conversation) 0 : No ADC wakeup module is interrupted, software clear 0 1 : ADC wakeup module is interrupted, hardware set 1 Note: when the corresponding wakeup module is prohibited, even if the conditions are met, the corresponding flag will not be set 1 . must enable EADC .
5-0	-	Reserved bit

Note: Wake-up resistance is configured by port pull-up resistance.

16 Low voltage detection LVD

16.1 LVD characteristics

Support internal VDD multi-level voltage detection, and can generate an interrupt

Support port voltage detection, and can generate an interrupt or reset

LVD point: 4.2V/3.9V/3.6V/3.0V/2.6V/2.4V/2.0V/1.9V

Same as BOR ,the internal voltage detection is used to detect VDD voltage, but independent to BOR, so it can detect multi-level voltage that are above BOR voltage, by register, user can set the voltage point, start/stop work, enable/disable interruptions.

LVD voltage detection circuit has a certain hysteresis, hysteresis voltage equal 0.1V or so. When detection voltage drops to the LVD voltage selected, LVD will generates an interrupt request or reset, then only the detection voltage needed to rise to LVD voltage +0.1V, the LVD interrupt request or reset be removed.

LVD detect the voltage on port P2.6, when the voltage is below 1.2V to detection voltage, set the corresponding flag, if the interrupt enable, an interrupt request is generated, if the interruption disable, port voltage detection will generate reset. Port voltage detection will generate valid interrupt and reset, it can wake up the chip from PD and IDLE mode.

When LVD detecting VDD voltage, no reset occur, but user can wake up chip from PD and IDLE mode by valid interrupt.

16.2 LVD registers

16.2.1 LVD control register LVDC

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	LVDEN	LVDS	LVDIE	-	LVDF	LVDV		

Bit	Flag	Introductions
7	LVDEN	LVD enable bit 0 : Disable LVD 1 : Enable LVD
6	LVDS	LVD detect selection bit 0 : Detect VDD Voltage 1 : Detect P2.6 Voltage (1.2V, ±5%)
5	LVDIE	LVD interrupt enable bit 0 : Disable LVD interrupt 1 : Enable LVD interrupt Note: when disable interrupt, as long as detection enabled, LVDF can also be set 1, but even if EA is set to 1 at this time,no interrupt request is generated. When LVDS is 1,detect P2.6 port voltage: LVDIE=0 : port voltage detection reset LVDIE=1 : port voltage detection interrupt
4	-	Reserved bit
3	LVDF	Low-voltage detection flag 0 : Must software clear 0 1 : When VDD voltage is lower than detection voltage, hardware set 1, also as interrupt request Note: When VDD voltage below detection voltage, the time is more than the debouncing time set in LVDDBC register, LVDF will be set; VDD is higher than detection voltage, LVDF do not automatically clear, the bit must be software clear, only VDD is higher than detection voltage continuously, software clear is valid, if VDD is lower than detection voltage continuously, software is unable to clear

		LVDF.
2-0	LVDV[2:0]	VDD voltage detection voltage selection bit 000 : 1.9V 001 : 2.0V 010 : 2.4V 011 : 2.6V 100 : 3.0V 101 : 3.6V 110 : 3.9V 111 : 4.2V Note: Only setting LVD detection voltage above BOR voltages is valid.

16.2.2 LVD debouncing control register LVDDBC

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	LVDDBC[7:0]							

Bit	Flag	Introductions
7-0	LVDDBC[7:0]	LVD debouncing control bit Debouncing time = LVDDBC[7:0] * $8T_{CPU}$ + $2T_{CPU}$

Note: In power-down and idle mode automatically turns off, and opens automatically when exit the power-down and idle mode.

17 Cyclic redundancy check CRC

17.1 CRC characteristics

- 16 bit CRC
- CRC check compliance with CRC-CCITT polynomials, that is 0x1021
- The initial value can be set 0x0000 or 0xFFFF
- Calculation and results share the same registers

Every write to data register CRCL, the calculated result is a previous CRC results combination of the new results.

Each time the read data from register [CRCH: CRCL], its value is the last CRC calculation results.

User can set CRCRSV bit of register CRCC to select initial calculation value, but not effects the CRC calculating data, only set CRCRST bit of register CRCC can reset CRC calculator, then write data will use new initial value to calculate CRC results.
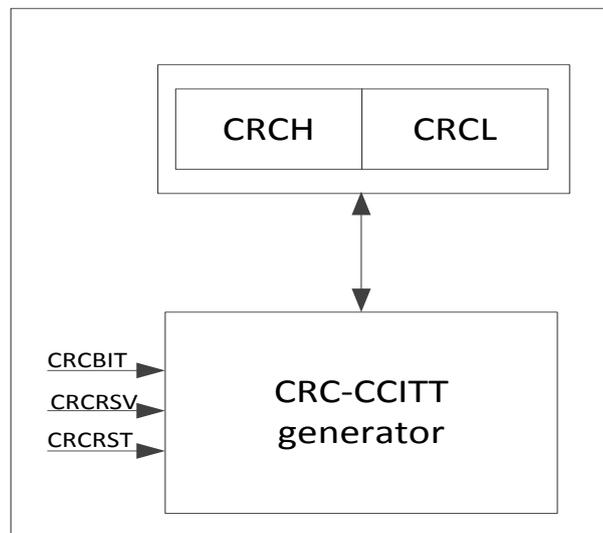
Figure 17 - 1 CRC functional block diagram

17.2 CRC registers

17.2.1 CRC control register CRCC

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R/W	R/W	W
Reset values	0	0	0	0	0	0	0	0
Flag	-					CRCBIT	CRCRSV	CRCRST

Bit	Flag	Introductions
7-3	-	Reserved (read as 0, write invalid)
2	CRCBIT	CRC BIT flip control bits 0 : MSB first 1 : LSB first
1	CRCRSV	CRC reset initial value selection bit 0 : reset initial value as 0x0000 1 : reset initial value as 0xFFFF
0	CRCRST	CRC calculator reset control bit Set 1 reset CRC calculator, hardware clear 0 automatically

17.2.2 CRC data register CRCL, CRCH

CRCL

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	CRCL[7:0]							

Bit	Flag	Introductions
7-0	CRCL[7:0]	As CRC calculator input data when write data As low bytes of CRC result when read data Note: when write data, start CRC calculated automatically, then close automatically when finished.

CRCH

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R	R	R
Reset values	0	0	0	0	0	0	0	0
Flag	CRCH[7:0]							

Bit	Flag	Introductions
7-0	CRCH[7:0]	Write data to the register is invalid As high bytes of CRC result when read data

Note: every time write data to be calculated, the calculation results are generated by common with previous results together.

18 Code options

1. External reset enable

 ➢　　P2.7 as external reset Pin (default). When the port as external reset Pin, it cannot as a normal I/O.

 ➢　　P2.7 as normal IO Pin

2. BOR detection voltage point

 ➢　　1.8V

 ➢　　2.0V

 ➢　　2.4V

 ➢　　2.6V

 ➢　　3.0V

 ➢　　3.6V

 ➢　　3.9V

 ➢　　4.2V

3.　Wait time after reset

 ➢　　4ms

 ➢　　8ms (Default)

 ➢　　16ms

4. The second reset vector configuration

　　　User can define the startup code address by the configuration, configuration values must be 1K bytes as a unit, so the second reset vector address lower 10 bits must be zero, the second reset vector is disabled default.

19 Electrical characteristics

19.1 Limit parameter

Parameter	Symbol	Min	Typical	Max	unit
DC power supply voltage	VDD	-0.3	-	+ 6.0	V
Input/output voltage	V_I/V_O	GND-0.3	-	VDD+0.3	V
Operating environment temperature	T_{OTG}	-40	-	+85	℃
Storage temperature	T_{STG}	-55	-	+125	℃

Note: (1) Maximum current through VDD <100mA @25℃ VDD=5V.
　　　(2) Maximum current through GND <150mA @25℃ VDD=5V.

19.2 DC characteristics

Parameter	Symbol	Condition (VDD=5V)	Min	Typical	Max	Unit
Operating voltage	VDD	F_{CPU} =16MHz 44KHz, ADC module closed	2.0	5.0	5.5	V
Operating current	I_{OP1}	F_{OSC} =32MHz, F_{CPU} =16MHz, No load, no floating input pins, execute NOP instructions, close the other modules	-	4.4	-	mA
		F_{OSC} =32MHz, F_{CPU} =8MHz, No load, no floating input pins, execute NOP instructions, close the other modules	-	3.5	-	
		F_{OSC} =32MHz, F_{CPU} =4MHz, No load, no floating input pins, execute NOP instructions, close the other modules	-	3.0	-	
		F_{OSC} =32MHz, F_{CPU} =2MHz, No load, no floating input pins, execute NOP Instructions, close the other modules	-	2.5	-	
		F_{OSC} =32MHz, F_{CPU} =1MHz, No load, no floating input pins, execute NOP Instructions, close the other modules	-	2.4	-	
		F_{OSC} =32MHz, F_{CPU} =512KHz, No load, no floating input pins, execute NOP Instructions, close the other modules	-	2.3	-	
		F_{OSC} =16MHz, F_{CPU} =16MHz, No load, no floating input pins, execute NOP instructions, close the other modules		3.6		
		F_{OSC} =16MHz, F_{CPU} =8MHz, No load, no floating input pins, execute NOP instructions, close the other modules		2.9		
		F_{OSC} =16MHz, F_{CPU} =4MHz, No load, no floating input pins, execute NOP instructions, close the other modules		2.1		
		F_{OSC} =16MHz, F_{CPU} =2MHz, No load, no floating input pins, execute NOP Instructions, close the other modules		1.8		
		F_{OSC} =16MHz, F_{CPU} =1MHz, No load, no floating input pins, execute NOP Instructions, close the other modules		1.6		
		F_{OSC} =16MHz, F_{CPU} =512KHz, No load, no floating input pins, execute NOP Instructions, close the other modules		1.5		
		F_{OSC} =8MHz, F_{CPU} =8MHz, No load, no floating input pins, execute NOP instructions, close the other modules		2.3		
		F_{OSC} =8MHz, F_{CPU} =4MHz, No load, no floating input pins, execute NOP instructions, close the other modules		1.7		
		F_{OSC} =8MHz, F_{CPU} =2MHz, No load, no		1.4		

		floating input pins, execute NOP Instructions, close the other modules				
		F_{OSC} =8MHz, F_{CPU} =1MHz, No load, no floating input pins, execute NOP Instructions, close the other modules		1.3		
		F_{OSC} =8MHz, F_{CPU} =512KHz, No load, no floating input pins, execute NOP Instructions, close the other modules		1.2		
		F_{OSC} =4MHz, F_{CPU} =4MHz, No load, no floating input pins, execute NOP instructions, close the other modules		1.5		
		F_{OSC} =4MHz, F_{CPU} =2MHz, No load, no floating input pins, execute NOP Instructions, close the other modules		1.2		
		F_{OSC} =4MHz, F_{CPU} =1MHz, No load, no floating input pins, execute NOP Instructions, close the other modules		1.1		
		F_{OSC} =4MHz, F_{CPU} =512KHz, No load, no floating input pins, execute NOP Instructions, close the other modules		1.0		
	I_{OP2}	F_{OSC} =44KHz, No load, no floating input pins, execute NOP Instructions, close the other modules	-	0.2	-	mA
	I_{PD}	Enter the power-down mode, no load, no floating input pins, close all modules	-	7.0	-	μA
	I_{IDLE1}	F_{OSC} =16MHz, Enter the idle mode, no load, no floating input pins, close all modules	-	2.0	-	mA
	I_{IDLE2}	F_{OSC} =8MHz, Enter the idle mode, no load, no floating input pins, close all modules	-	1.3	-	mA
	I_{IDLE3}	F_{OSC} =4MHz, Enter the idle mode, no load, no floating input pins, close all modules	-	900	-	μA
	I_{IDLE4}	F_{OSC} =2MHz, Enter the idle mode, no load, no floating input pins, close all modules	-	700	-	μA
	I_{IDLE5}	F_{OSC} =44KHz, Enter the idle mode, no load, no floating input pins, all closed, internal high-frequency RC Clock close	-	85	-	μA
power-down Timer interrupt wakeup current	I_{PW}	F_{CPU} =16MHz, close the BOR, TIMER3 count clock source select external low-frequency crystal oscillator, the system enter power-down mode, the average current with TIMER3 timing 1S interrupts to wake up the system	-	12	-	μA
WDT current	I_{WDT}	VDD = 5V	-	2.0	-	μA
LVD current	I_{LVD}	VDD = 5V	-	8.0	-	
BOR current	I_{BOR}	VDD = 5V	-	8.0	-	μA
Input low voltage 1	V_{IL1}	I/O port non-Schmitt input	GND	-	0.3*VDD	V
Input high voltage 1	V_{IH1}	I/O port non-Schmitt input	0.7*VDD	-	VDD	V
Input low voltage 2	V_{IL2}	I/O port Schmitt input	GND	-	0.2*VDD	V
Input high voltage 2	V_{IH2}	I/O port Schmitt input	0.8*VDD	-	VDD	V
Input leakage current	I_{ILC}	I/O port input mode, V_{IN} = VDD or GND	-1	0	1	μA
output leakage current	I_{OLC}	I/O port output mode V_{OUT} = VDD or GND	-1	0	1	μA
Sink	I_{OL}	Vout=GND+0.6	-	25	-	mA
Current	I_{OH}	Vout=VDD-0.6	-	21	-	
Pull-up resistor	R_{PU1}	P0.2 port, VIN=GND	-	50	-	kΩ

	R$_{PU2}$	P0.2 port, VIN=GND	-	100	-	
	R$_{PU3}$	P0.2 port, VIN=GND	-	150	-	
	R$_{PU4}$	P0.2 port, VIN=GND	-	300	-	
	R$_{PU5}$	Common port, VIN=GND	-	50	-	
Pull-down resistance	R$_{PD}$	Common port, VIN=VDD	-	50	-	
Pull-up resistor value	R$_{PUPD}$	P2.3, P2.4, P2.5, P2.7 port, VIN=GND	-	Pull-up: 60 Pull-down: 55	-	kΩ
ADC Wake up voltage	V$_{AW1}$	room temperature, VDD = 5V	4.0	4.2	4.4	
	V$_{AW2}$	room temperature, VDD = 3V	2.3	2.5	2.6	V
RAM maintain voltage	V$_{RAM}$	-	-	0.7	-	

Note: Subject to general operating conditions for VDD=5.0V GND=0V, 25 ℃ unless otherwise specified.

19.3 AC characteristics

Parameter	Symbol	Conditions	Min	Typical	Max	Unit
Internal RC 32M Startup time	Tset1	room temperature, VDD=5V	-	-	5	μs
External high-frequency oscillator Startup time	Tset3	16MHz, room temperature, VDD=5V	-	200	-	μs
high-frequency oscillator work volatge	Vset3	16MHz	2.5	-	5.5	V
External low-frequency oscillator Startup time	Tset4	room temperature, VDD=5V	-	2	-	s
Internal RC 44K startup time	Tset2	room temperature, VDD=5V	-	-	150	μs
Frequency accuracy	FIRC1	VDD=2V~5.5V, 25℃	32 (1-1%)	32	32 (1+1%)	MHz
	FIRC2	VDD=5.0V,-40℃ ~+85℃	32 (1-1%)	32	32 (1+1%)	MHz
	FWRC	-	31	44	58	KHz

19.4 ADC characteristics

Parameter	Symbol	Conditions	Min	Typical	Max	Unit
power supply voltage	VAD	-	2.7	5.0	5.5	V
Precision	NR	GND ≤ VAIN ≤ Vref	-	10	12	bit
ADC input voltage	VAIN	-	GND	-	Vref	V
ADC input resistance	RAIN	VAIN=5V	2	-	-	MΩ
Analog voltage sources recommended impedance	ZAIN	-	-	-	10	kΩ
ADC switching current	IAD	ADC module opened, VDD=5.0V	-	0.6	1	mA
ADC input current	IADIN	VDD=5.0V	-	-	10	μA
Differential nonlinearity error	DLE	VDD=5.0V	-	-	±2	LSB
Integral nonlinearity error (1MHz switching frequency)	ILE	VDD=5.0V, Vref =2V	-	-	-5~2	LSB
		VDD=5.0V, Vref =3V	-	-	-4~2	
		VDD=5.0V, Vref =4V	-	-	-3~2	
		VDD=5.0V, Vref =VDD	-	-	±2	
		VDD=5.0V, Vref = External reference voltage	-	-	±2	
Full scale error	EF	VDD=5.0V	-	-	±5	LSB

Offset error	EZ	VDD=5.0V	-	-	±3	LSB
Total error	EAD	VDD=5.0V	-	-	±5	LSB
Total conversion time	TCON	VDD=5.0V	10	-	-	μs
Internal reference voltage	VADREF	±1%,room temperature	-	2	-	V

19.5 BOR detection voltage characteristics

Parameter	Symbol	Condition	Min	Typ	Max	Unit
BOR Set voltage 1	VBOR1	BOR is enabled, VDD=2V~5.5V	1.7	1.8	1.9	V
BOR Set voltage 2	VBOR2		1.9	2.0	2.1	V
BOR Set voltage 3	VBOR3		2.3	2.4	2.5	V
BOR Set voltage 4	VBOR4		2.5	2.6	2.7	V
BOR Set voltage 5	VBOR5		2.9	3.0	3.1	V
BOR Set voltage 6	VBOR6		3.5	3.6	3.7	V
BOR Set voltage 7	VBOR7		3.8	3.9	4.0	V
BOR Set voltage 8	VBOR8		4.1	4.2	4.3	V

19.6 LVD/PLVD detection voltage characteristics

Parameter	Symbol	Condition	Min	Typ	Max	Unit
LVD Set voltage 0	VPLVD	LVD is enabled, VDD=2V~5.5V	-	1.2	-	V
LVD Set voltage 1	VLVD1		1.8	1.9	2.0	V
LVD Set voltage 2	VLVD2		1.9	2.0	2.1	V
LVD Set voltage 3	VLVD3		2.3	2.4	2.5	V
LVD Set voltage 4	VLVD4		2.5	2.6	2.7	V
LVD Set voltage 5	VLVD5		2.9	3.0	3.1	V
LVD Set voltage 6	VLVD6		3.5	3.6	3.7	V
LVD Set voltage 7	VLVD7		3.8	3.9	4.0	V
LVD Set voltage 8	VLVD8		4.1	4.2	4.3	V

19.7 System power consumption during power off

1, System shutdown BOR and enter the power-down mode
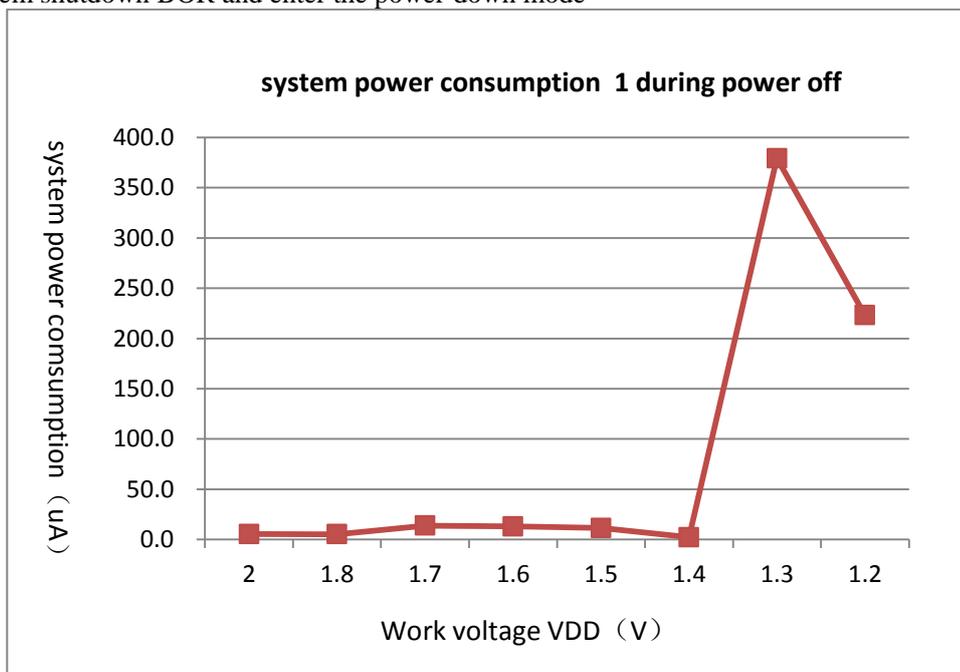
Figure 19 - 1 system power consumption 1 during power off

2, System enable BOR and enter power-down mode, enable/disable RC_EN_PD (See system BOR status, high frequency internal RC closed signals control register RC_PD_EN Chapter) signal
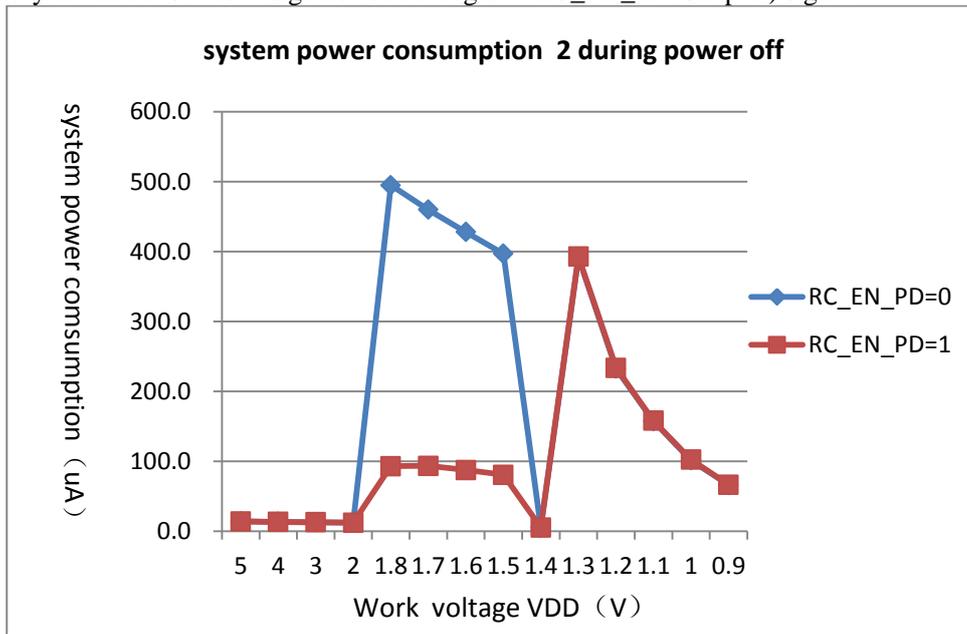

Figure 19 - 2 system power consumption 2 during power off

19.8 Other electrical characteristics

1, ESD (HBM): CLASS 3A (≥4000V)
2, ESD (MM): CLASS 2 (≥200V)
3, Latch_up : CLASS Ⅰ (100mA)

QFN20

Top View

Bottom View

Side View
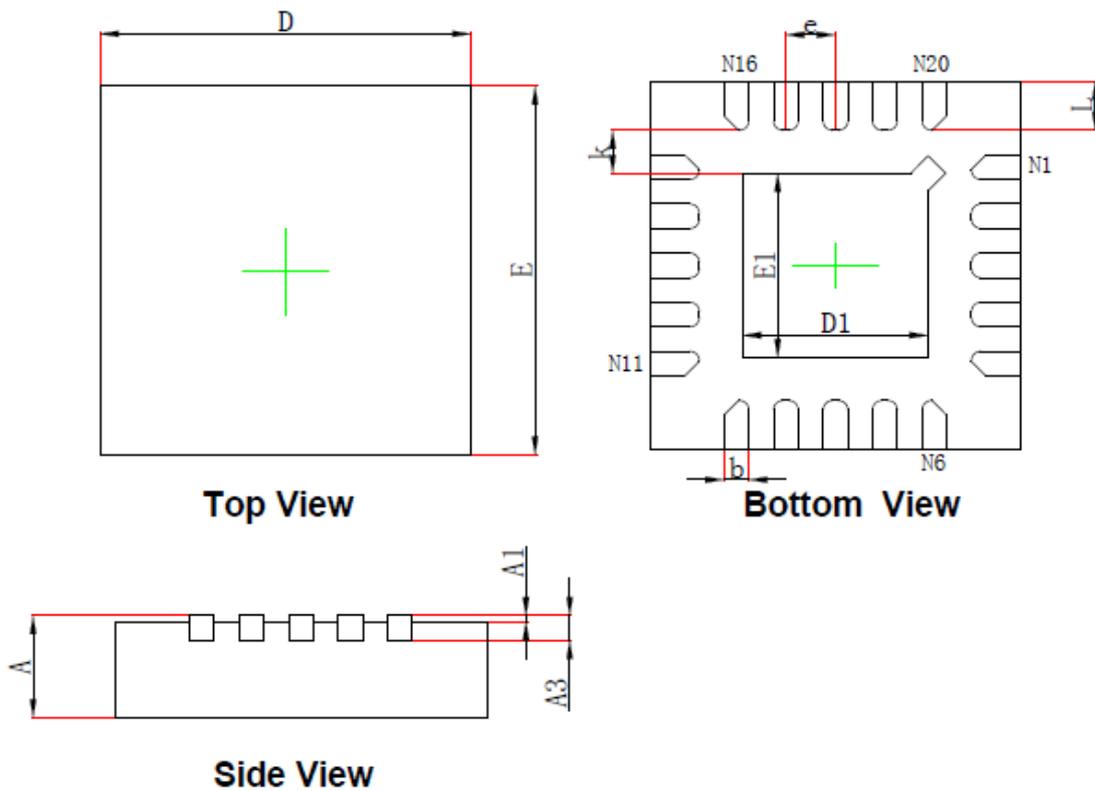
Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min.	Max.	Min.	Max.
A	0.700/0.800	0.800/0.900	0.028/0.031	0.031/0.035
A1	0.000	0.050	0.000	0.002
A3	0.203REF.		0.008REF.	
D	2.924	3.076	0.115	0.121
E	2.924	3.076	0.115	0.121
D1	1.400	1.600	0.055	0.063
E1	1.400	1.600	0.055	0.063
k	0.200MIN.		0.008MIN.	
b	0.150	0.250	0.006	0.010
e	0.400TYP.		0.016TYP.	
L	0.324	0.476	0.013	0.019

Figure 22-2 QFN20 封装尺寸

20 Development tools

20.1 Emulator characteristics

HC89S003F4 use HC-51LINK/HC-LINK emulator to program download and simulation, By JTAG Interface emulator implement the enhanced 8051 MCU of Holychip program download simulation. About the emulator, please refer the emulator's user manual.

Emulator characteristics
 ➢ Support Keil C51 integration build environment (uVision2.34 and above Ver.)
 ➢ Support all Holychip 8051 MCU
 ➢ Support FLASH erase, program and verify
 ➢ Support encryption bit and code option program
 ➢ Get power from USB directly, no external power supply

20.2 Programmer tools

HC-PM51 is Holychip new programmer for mass production, supports the program of all the enhanced 8051 MCU of Holychip. About the programmer, please refer the HC-PM51's user manual.
Programmer characteristics:
 ➢ USB port connection
 ➢ Support signal channel off-line programming

20.3 Software download

Software downloads address: http://www.holychip.cn

21 Package

21.1 TSSOP20

SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.20
A1	0.05	—	0.15
A2	0.80	1.00	1.05
A3	0.39	0.44	0.49
b	0.20	—	0.29
b1	0.19	0.22	0.25
c	0.13	—	0.18
c1	0.12	0.13	0.14
D	6.40	6.50	6.60
E1	4.30	4.40	4.50
E	6.20	6.40	6.60
e	0.65BSC		
L	0.45	0.60	0.75
L1	1.00BSC		
θ	0	—	8°
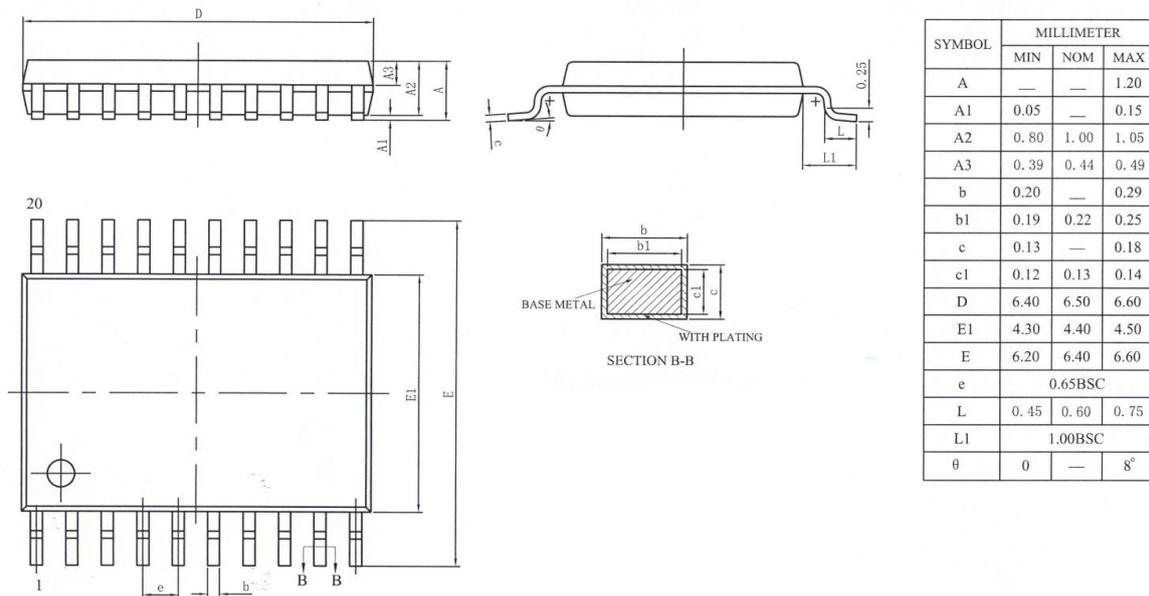
Figure 21 - 1 TSSOP20 package size

22 Reversion history

Document revision history

Version	Date	Description
Ver1.00	2017-09-04	First version
Ver1.01	2018-02-26	1. Added oscillator port and descriptions 2. Added ADC channels and descriptions 3. Translate from Chinese version v1.01