HC89S105xx
Datasheet

48/44/32Pin 8bit
FLASH Microcontroller with ADC
Peripheral function Ports total mapping

Contents

1 Description

HC89S105xx is an enhanced 8 bit microcontroller with high frequency and low power consumption CMOS process. 64K bytes flash program memory, 256 bytes IRAM and 2k bytes XRAM, 6 groups bi-directional I/O, 1 peripheral function Ports total mapping module, two 16-bit Timer/counters, 2 PCA, 3 groups 12 bits complementary PWM with dead-time control, 2 UART, 1 IIC, 1 RTC, 15/13 external interrupts, 23+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**
  - 64K bytes flash
  - Support IAP and ICP operation
  - Flexible code protection mode
- **RAM**
  - 256 bytes IRAM
  - 2K bytes XRAM
- **Clock**
  - Internal high precision (±1%) 32MHz RC
  - Internal 38 KHz RC
  - External high frequency oscillator 4MHz-16MHz
  - External low frequency oscillator 32.768KHz
  - Peripheral module clock can be switched separately
  - The maximum operating speed of the system clock is 16MHz
- **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
  - External pin low voltage reset
- **I/O**
  - 41/29 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,
  - Stage 2 port drive current optional
  - Peripheral function Ports total mapping module
- **interrupt**
  - 16 interrupt sources
  - 4 level interrupt priorities
  - 15/13 external interrupts
- **Timer/Counter**
  - two 16-bit Timer (T0/T1)
  - 16-bit auto reload
- **PWM**
  - Up to 3 groups 12 bits complementary PWM with dead-time control
  - Malfunction detection function
- **PCA**
  - 16-bit capture/compare/software timing
  - 16-bit PWM
- **PTC**
- **Communication interfaces**
  - 2 UART
  - 1 SPI
  - 1 IIC
- **Analog to digital converter (ADC)**
  - 12 bit ADC, up to 23+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
  - External pin voltage (1.2V) detection, interruptible
- **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°C to +85°C
- **Package**
  - LQFP48
  - LQFP44
  - LQFP32

✓ **Selection table**

Device	ROM	RAM	I/O	Timer	PWM	A/D	INT	PCA	IIC	UART	SPI	WDT	Voltage	TEMP	Package
HC89S105C8	64K	256+2K	41	16bit*2	12bit*3 group	23+2	15	2	1	2	1	1	2.0~5.5V	-40~+85℃	LQFP48
HC89S105S8	64K	256+2K	41	16bit*2	12bit*3 group	23+2	15	2	1	2	1	1	2.0~5.5V	-40~+85℃	LQFP44
HC89S105K8	64K	256+2K	29	16bit*2	12bit*3 group	17+2	13	2	1	2	1	1	2.0~5.5V	-40~+85℃	LQFP32

Device	Simulator	Programmer	Datasheet	Demo Code	Demo Board
HC89S105xx	HC-LINK	HC-PM51/ HC-ISP	√	√	√

HC89S105xx **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. Before IAP writes and enters power off mode, the FREQ_CLK register needs to be configured to indicate the current CPU clock frequency.
4. If RTCEN is 1, the RTC module will start to work. It will continue to work in the power down mode.When the RTC count overflows, the RTC will continue to work.Interruption will awaken the power mode, but after awakening it must operate for 30 s to enter the next power outage if RTC is not required.Working in mode requires RTCEN to be 0.
5. If you use the drop edge external interrupt to wake up the power mode, you must keep the external interrupt pin high before entering the power down mode,Both interrupt enable and interrupt allow registers need to be turned on.

1.2 System diagram

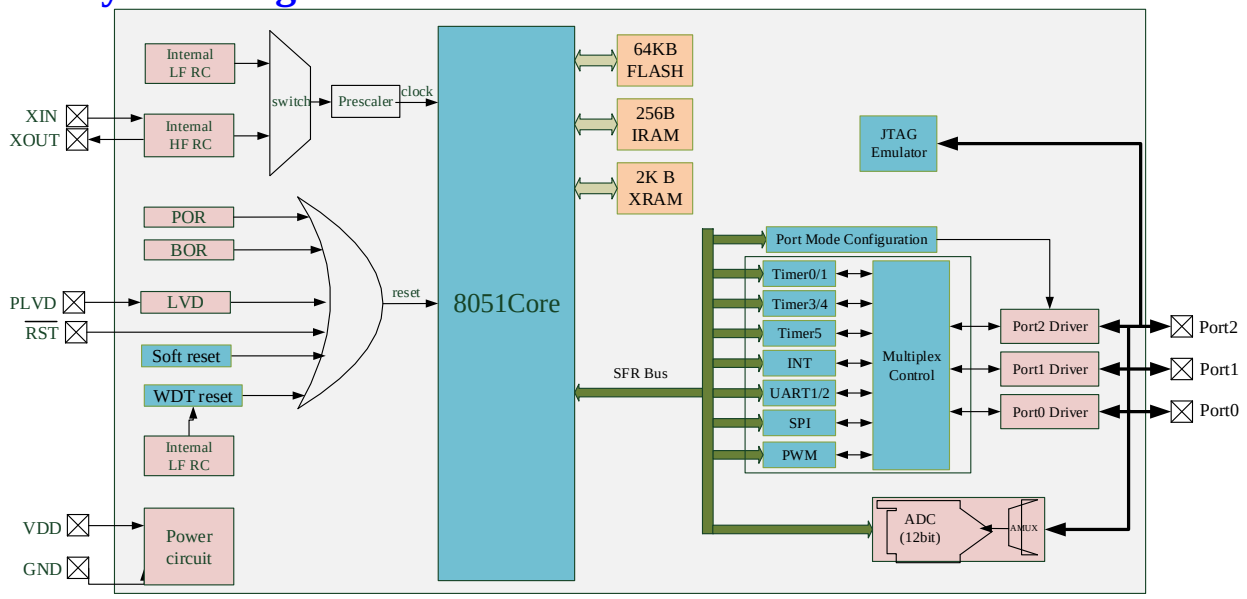


Figure 1-1 System diagram

1.3 Pin configuration

1.3.1 LQFP48 Pin configuration

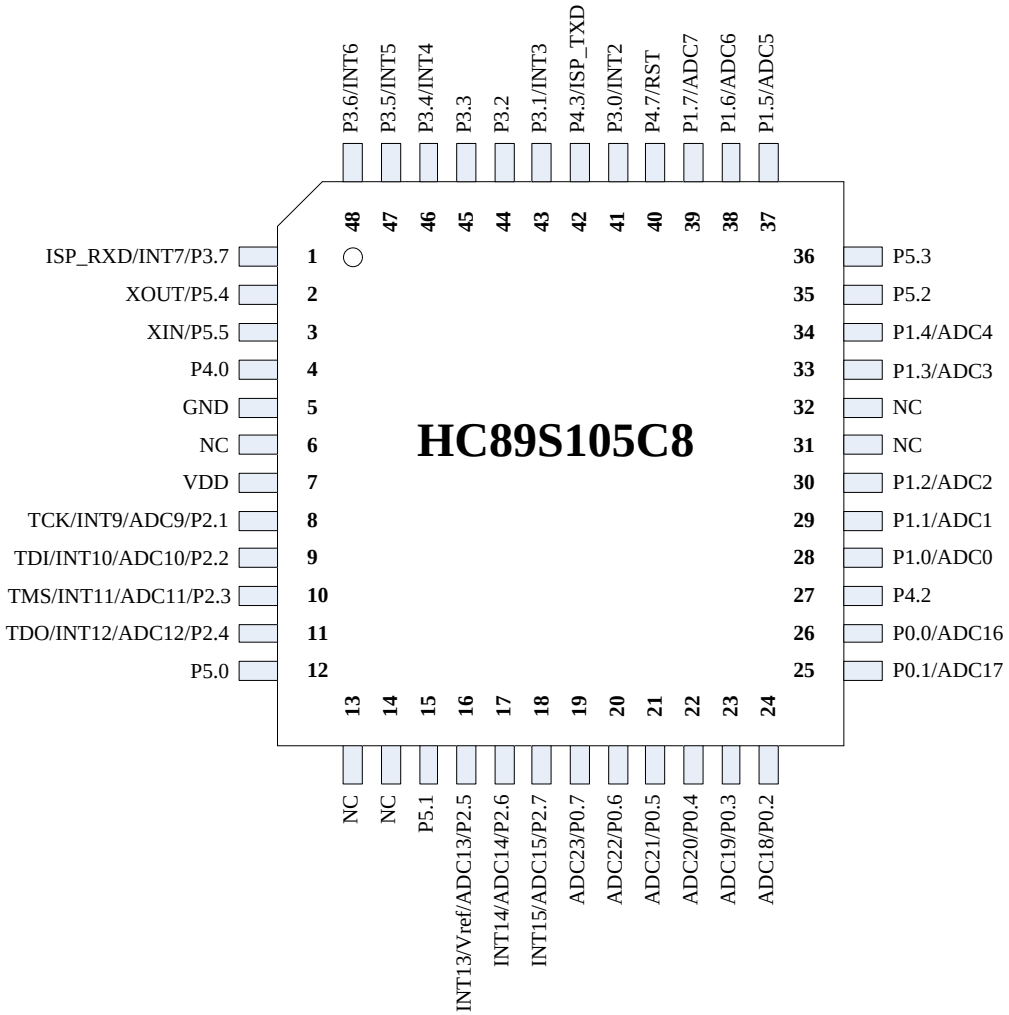


Figure 1-2 LQFP48 pin configuration diagram

1.3.2 LQFP44 Pin configuration

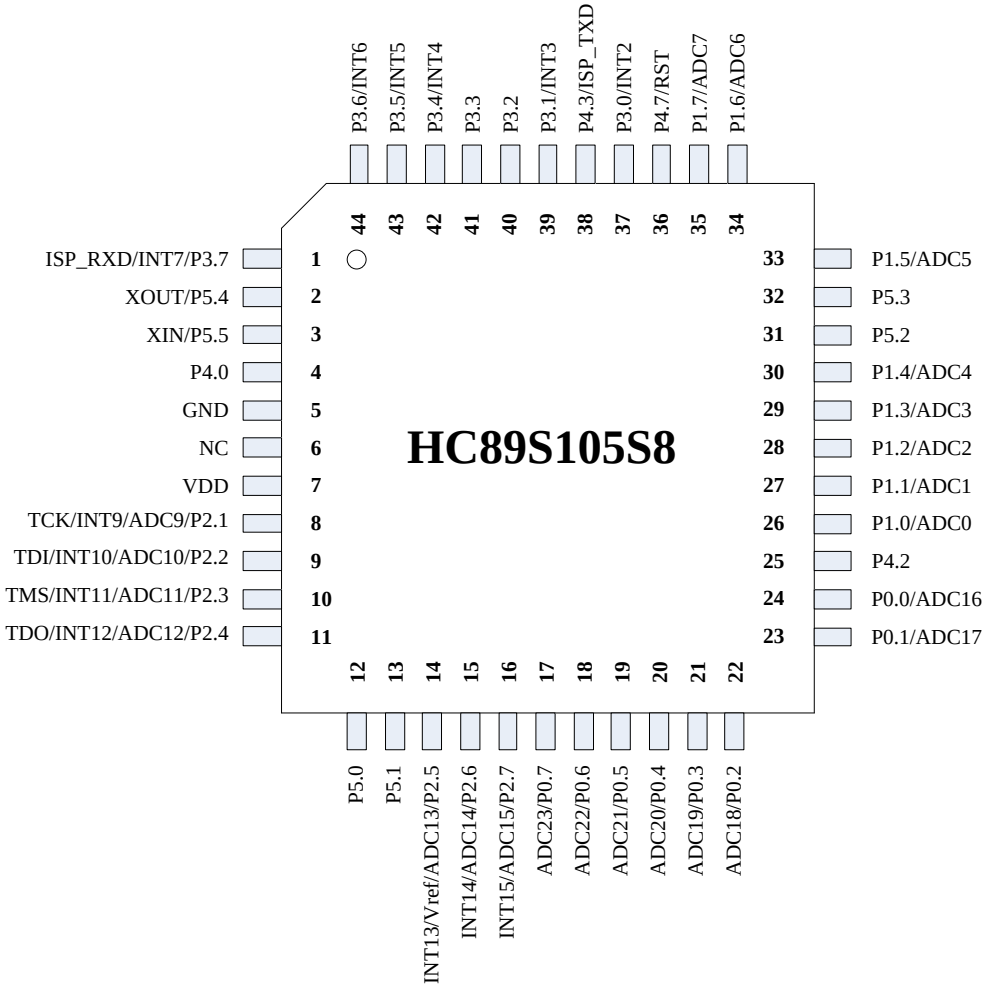


Figure 1-3 LQFP44 pin configuration diagram

1.3.3 LQFP32 Pin configuration

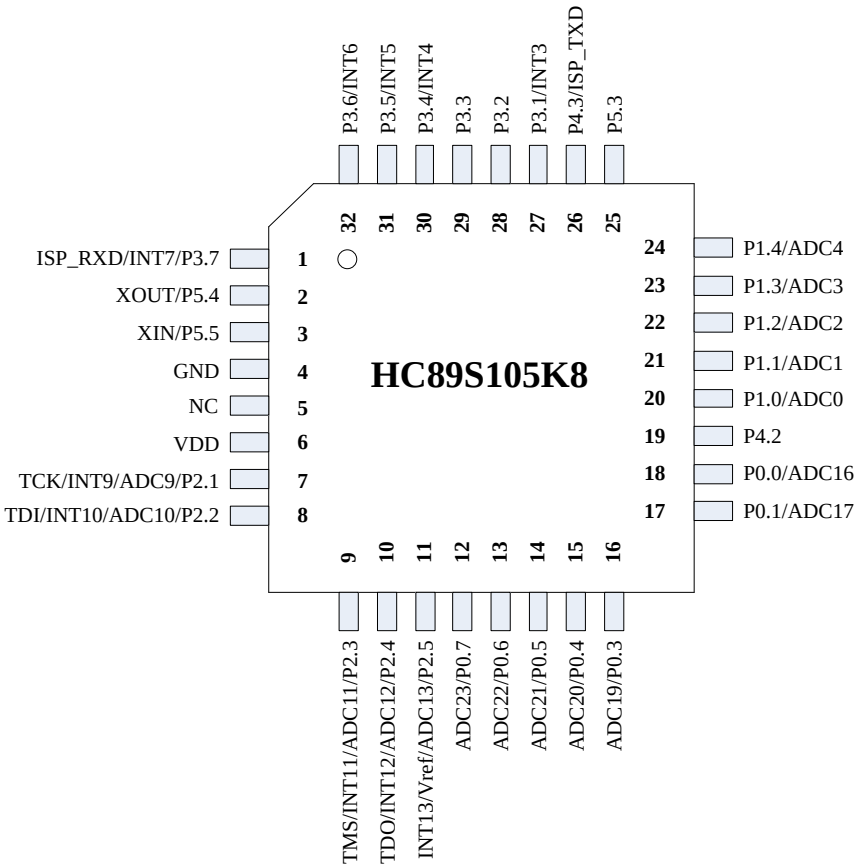


Figure 1-4 LQFP32pin configuration diagram

1.4 Pin description

1.4.1 LQFP48 Pin configuration

Pin	Name	Type	Introductions
1	P3.7 INT7 ISP_RXD	I/O I I	Input/output port External interrupt 7, input port ISP download RXD port (this port does not affect the normal use of UART)
2	P5.4 XOUT	I/O AN	Input/output port External crystal oscillator outlet
3	P5.5 XIN	I/O AN	Input/output port External crystal input port
4	P4.0	I/O	Input/output port
5	GND	P	Power ground
6	NC	-	Dangling feet
7	VDD	P	Power input
8	P2.1 AN9 INT9 TCK	I/O AN I I	Input/output port ADC9 input port External interrupt 9, input port JTAG clock input
9	P2.2 AN10 INT10 TDI	I/O AN I I	Input/output port ADC10 input port External interrupt 10, input port JTAG data input
10	P2.3 AN11 INT11 TMS	I/O AN I I	Input/output port ADC11 input port External interrupt 11, input port JTAG data input
11	P2.4 AN12 INT12 TDO	I/O AN I O	Input/output port ADC12 input port External interrupt 12, input port JTAG data output
12	P5.0	I/O	Input/output port
13	NC	-	Dangling feet
14	NC	-	Dangling feet
15	P5.1	I/O	Input/output port
16	P2.5 AN13 INT13 Vref	I/O AN I AN	Input/output port ADC13 input port External interrupt 13, input port ADC external reference voltage input port
17	P2.6 AN14 INT14	I/O I AN	Input/output port ADC14 input port External interrupt 14, input port
18	P2.7 AN15 INT15	I/O AN I	Input/output port ADC15 input port External interrupt 15, input port
19	P0.7 AN23	I/O AN	Input/output port ADC23 input port
20	P0.6 AN22	I/O AN	Input/output port ADC22 input port
21	P0.5 AN21	I/O AN	Input/output port ADC21 input port
22	P0.4 AN20	I/O AN	Input/output port ADC20 input port
23	P0.3	I/O	Input/output port

	AN19	AN	ADC19 input port
24	P0.2 AN18	I/O AN	Input/output port ADC18 input port
25	P0.1 AN17	I/O AN	Input/output port ADC17 input port
26	P0.0 AN16	I/O AN	Input/output port ADC16 input port
27	P4.2	I/O	Input/output port
28	P1.0 AN0	I/O AN	Input/output port ADC0 input port
29	P1.1 AN1	I/O AN	Input/output port ADC1 input port
30	P1.2 AN2	I/O AN	Input/output port ADC2 input port
31	NC	-	Dangling feet
32	NC	-	Dangling feet
33	P1.3 AN3	I/O AN	Input/output port ADC3 input port
34	P1.4 AN4	I/O AN	Input/output port ADC4 input port
35	P5.2	I/O	Input/output port
36	P5.3	I/O	Input/output port
37	P1.5 AN5	I/O AN	Input/output port ADC5 input port
38	P1.6 AN6	I/O AN	Input/output port ADC6 input port
39	P1.7 AN7	I/O AN	Input/output port ADC7 input port
40	P4.7 RST	I/O I	Input/output port External reset input port
41	P3.0 INT2	I/O I	Input/output port External interrupt 2, input port
42	P4.3 ISP_RXD	I/O O	Input/output port ISP download RXD port (this port does not affect the normal use of UART)
43	P3.1 INT3	I/O I	Input/output port External interrupt 3, input port
44	P3.2	I/O	Input/output port
45	P3.3	I/O	Input/output port
46	P3.4 INT4	I/O I	Input/output port External interrupt 4, input port
47	P3.5 INT5	I/O I	Input/output port External interrupt 5, input port
48	P3.6 INT6	I/O I	Input/output port External interrupt 6, input port

1.4.2 LQFP44 Pin configuration

Pin	Name	Type	Introductions
1	P3.7 INT7 ISP_RXD	I/O I I	Input/output port External interrupt 7, input port ISP download RXD port (this port does not affect the normal use of UART)
2	P5.4 XOUT	I/O AN	Input/output port External crystal oscillator outlet
3	P5.5	I/O	Input/output port

	XIN	AN	External crystal input port
4	P4.0	I/O	Input/output port
5	GND	P	Power ground
6	NC	-	Dangling feet
7	VDD	P	Power input
8	P2.1 AN9 INT9 TCK	I/O AN I I	Input/output port ADC9 input port External interrupt 9, input port JTAG clock input
9	P2.2 AN10 INT10 TDI	I/O AN I I	Input/output port ADC10 input port External interrupt 10, input port JTAG data input
10	P2.3 AN11 INT11 TMS	I/O AN I I	Input/output port ADC11 input port External interrupt 11, input port JTAG data input
11	P2.4 AN12 INT12 TDO	I/O AN I O	Input/output port ADC12 input port External interrupt 12, input port JTAG data output
12	P5.0	I/O	Input/output port
13	P5.1	I/O	Input/output port
14	P2.5 AN13 INT13 Vref	I/O AN I AN	Input/output port ADC13 input port External interrupt 13, input port ADC external reference voltage input port
15	P2.6 AN14 INT14	I/O I AN	Input/output port ADC14 input port External interrupt 14, input port
16	P2.7 AN15 INT15	I/O AN I	Input/output port ADC15 input port External interrupt 15, input port
17	P0.7 AN23	I/O AN	Input/output port ADC23 input port
18	P0.6 AN22	I/O AN	Input/output port ADC22 input port
19	P0.5 AN21	I/O AN	Input/output port ADC21 input port
20	P0.4 AN20	I/O AN	Input/output port ADC20 input port
21	P0.3 AN19	I/O AN	Input/output port ADC19 input port
22	P0.2 AN18	I/O AN	Input/output port ADC18 input port
23	P0.1 AN17	I/O AN	Input/output port ADC17 input port
24	P0.0 AN16	I/O AN	Input/output port ADC16 input port
25	P4.2	I/O	Input/output port
26	P1.0 AN0	I/O AN	Input/output port ADC0 input port
27	P1.1 AN1	I/O AN	Input/output port ADC1 input port
28	P1.2	I/O	Input/output port

	AN2	AN	ADC2 input port
29	P1.3 AN3	I/O AN	Input/output port ADC3 input port
30	P1.4 AN4	I/O AN	Input/output port ADC4 input port
31	P5.2	I/O	Input/output port
32	P5.3	I/O	Input/output port
33	P1.5 AN5	I/O AN	Input/output port ADC5 input port
34	P1.6 AN6	I/O AN	Input/output port ADC6 input port
35	P1.7 AN7	I/O AN	Input/output port ADC7 input port
36	P4.7 RST	I/O I	Input/output port External reset input port
37	P3.0 INT2	I/O I	Input/output port External interrupt 2, input port
38	P4.3 ISP_RXD	I/O O	Input/output port ISP download RXD port (this port does not affect the normal use of UART)
39	P3.1 INT3	I/O I	Input/output port External interrupt 3, input port
40	P3.2	I/O	Input/output port
41	P3.3	I/O	Input/output port
42	P3.4 INT4	I/O I	Input/output port External interrupt 4, input port
43	P3.5 INT5	I/O I	Input/output port External interrupt 5, input port
44	P3.6 INT6	I/O I	Input/output port External interrupt 6, input port

1.4.3 LQFP32 Pin configuration

Pin	Name	Type	Introductions
1	P3.7 INT7 ISP_RXD	I/O I I	Input/output port External interrupt 7, input port ISP download RXD port (this port does not affect the normal use of UART)
2	P5.4 XOUT	I/O AN	Input/output port External crystal oscillator outlet
3	P5.5 XIN	I/O AN	Input/output port External crystal input port
4	GND	P	Power ground
5	NC	-	Dangling feet
6	VDD	P	Power input
7	P2.1 AN9 INT9 TCK	I/O AN I I	Input/output port ADC9 input port External interrupt 9, input port JTAG clock input
8	P2.2 AN10 INT10 TDI	I/O AN I I	Input/output port ADC10 input port External interrupt 10, input port JTAG data input
9	P2.3 AN11 INT11	I/O AN I	Input/output port ADC11 input port External interrupt 11, input port

	TMS	I	JTAG data input
10	P2.4 AN12 INT12 TDO	I/O AN I O	Input/output port ADC12 input port External interrupt 12, input port JTAG data output
11	P2.5 AN13 INT13 Vref	I/O AN I AN	Input/output port ADC13 input port External interrupt 13, input port ADC external reference voltage input port
12	P0.7 AN23	I/O AN	Input/output port ADC23 input port
13	P0.6 AN22	I/O AN	Input/output port ADC22 input port
14	P0.5 AN21	I/O AN	Input/output port ADC21 input port
15	P0.4 AN20	I/O AN	Input/output port ADC20 input port
16	P0.3 AN19	I/O AN	Input/output port ADC19 input port
17	P0.1 AN17	I/O AN	Input/output port ADC17 input port
18	P0.0 AN16	I/O AN	Input/output port ADC16 input port
19	P4.2	I/O	Input/output port
20	P1.0 AN0	I/O AN	Input/output port ADC0 input port
21	P1.1 AN1	I/O AN	Input/output port ADC1 input port
22	P1.2 AN2	I/O AN	Input/output port ADC2 input port
23	P1.3 AN3	I/O AN	Input/output port ADC3 input port
24	P1.4 AN4	I/O AN	Input/output port ADC4 input port
25	P5.3	I/O	Input/output port
26	P4.3 ISP_RXD	I/O O	Input/output port ISP download RXD port (this port does not affect the normal use of UART)
27	P3.1 INT3	I/O I	Input/output port External interrupt 3, input port
28	P3.2	I/O	Input/output port
29	P3.3	I/O	Input/output port
30	P3.4 INT4	I/O I	Input/output port External interrupt 4, input port
31	P3.5 INT5	I/O I	Input/output port External interrupt 5, input port
32	P3.6 INT6	I/O I	Input/output port External interrupt 6, input port

1.5 Peripheral function Ports total mapping module PTM

HC89S105xx 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 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 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
PCA	ECI	I	PCA external input
	PCA0	I/O	PCA0 input/output
	PCA1	I/O	PCA1 input/output
PWM	FLT0	I	PWM0 fault detection input port
	PWM0	O	PWM0 output port
	PWM01	O	PWM01 output port
	FLT1	I	PWM1 fault detection input port
	PWM1	O	PWM1 output port
	PWM11	O	PWM11 output port
	FLT2	I	PWM2 fault detection input port
	PWM2	O	PWM2 output port
	PWM21	O	PWM21 output port
RTC	RTCO	O	RTC clock frequency divider output
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
	$\overline{SS}$	I	SPI chip select port
IIC	SCL	I/O	IIC clock port
	SDA	I/O	IIC data port
INT	INT0	I	External interrupt 0
	INT1	I	External interrupt 1

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), ADC input, INT0-15 function port, oscillator Pin(Xin,XOUT),external reset Port(RST).

2 CPU

2.1 CPU characteristics

HC89S105xx 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	R	R	R	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-			-	-	-	-	DPS

Bit	Flag	Introductions
7-1	-	Reserved (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 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.

3.1.3.1 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

3.1.3.2 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;
```

3.1.4 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.

3.1.5 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.

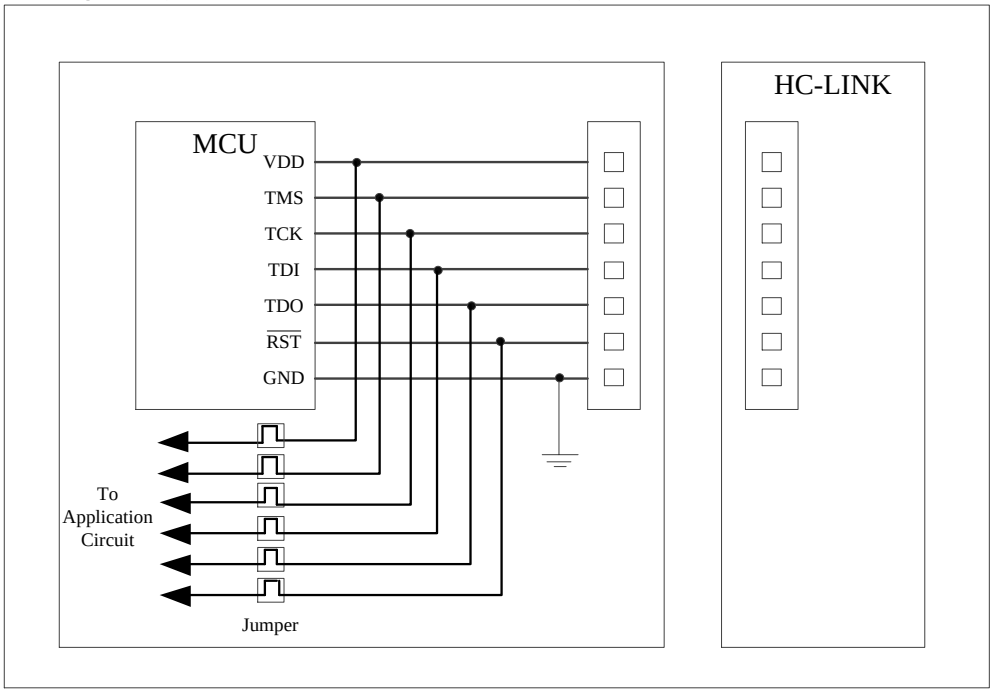


Figure 3 - 1 HC-51LINK programming hardware connection

3.2 Data storage (RAM)

HC89S105xx provide user with a 256 bytes internal RAM and 2K bytes internal expansion RAM as data memory. Below is data memory space allocation.

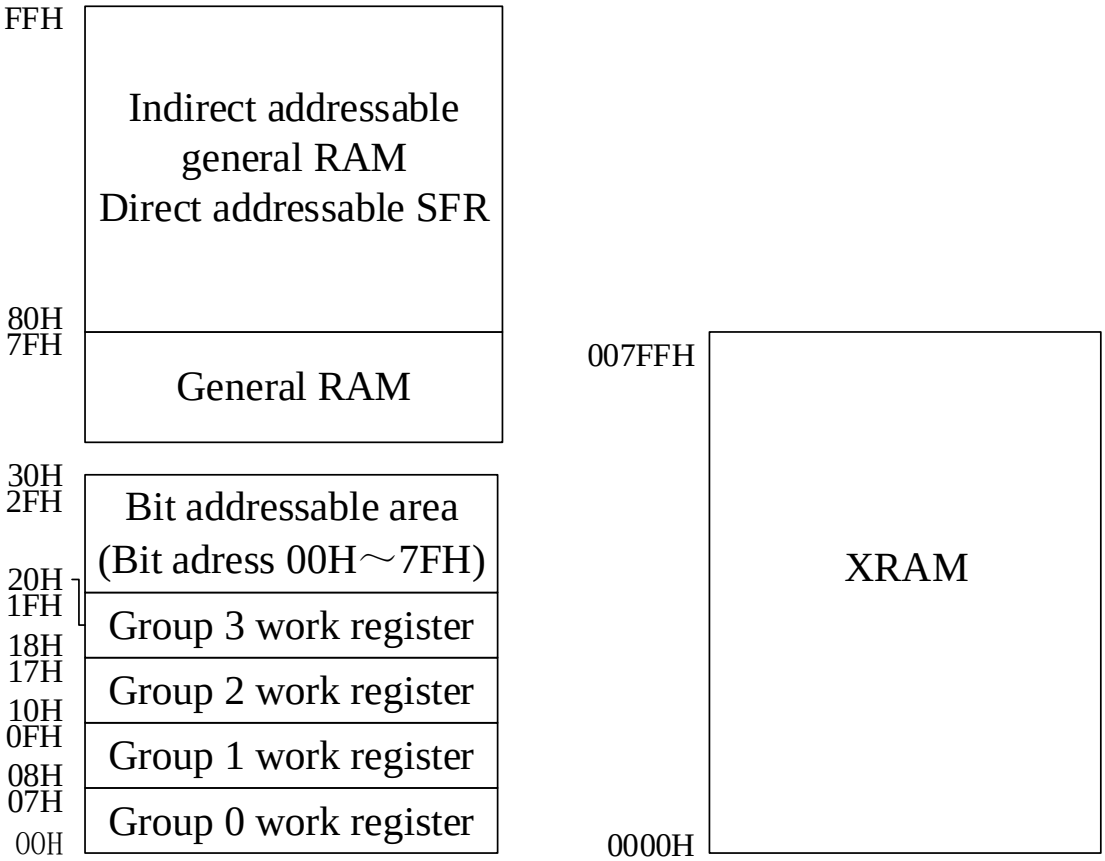


Figure 3 - 2 Data memory map

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

Internal expansion RAM (XRAM) addresses range is 0x0000 ~ 0x7FF, 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	-	PWM2C	PWM2PL	PWM2PH	PWM2DL	PWM2DH	PWM2DT
E8	-	PWMFLT	PWM1C	PWM1PL	PWM1PH	PWM1DL	PWM1DH	PWM1DT
E0	ACC	PWMEN	PWM0C	PWM0PL	PWM0PH	PWM0DL	PWM0DH	PWM0DT
D8	-	-	-	-	-	-	-	-
D0	PSW	-	-	-	-	-	-	-
C8	P5	PCACLK	PCAMOD0	PCAMOD1	CCAPL0	CCAPH0	CCAPL1	CCAPH1
C0	P4	PCACON	PCACL	PCACH	-	-	-	-
B8	IE1	IP2	IP3	LVDC	RTCC	WDTC	CRCL	CRCH
B0	P3	IP4	-	-	ADCC0	ADCC1	ADCRL	ADCRH
A8	IE	IP0	IP1	SPDAT	SPCTL	SPSTAT	IICDAT	IICADR
A0	P2	-	-	INSCON	-	-	IICCON	IICSTA
98	SCON	SBUF	SADDR	SADEN	SBRTL	SBRTH	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	-	0x0023	-	0x0033	PITS3
0x0004	PCA_PWM0	0x0014	-	0x0024	BORC	0x0034	-
0x0005	PCA_PWM1	0x0015	-	0x0025	BORDBC	0x0035	-
0x0006	-	0x0016	-	0x0026	-	0x0036	-
0x0007	-	0x0017	-	0x0027	LVDDBC	0x0037	-
0x0008	S2CON	0x0018	ADCWC0	0x0028	-	0x0038	PINTE0
0x0009	S2CON2	0x0019	ADCWC1	0x0029	-	0x0039	PINTE1
0x000A	S2BUF	0x001A	-	0x002A	RSTDBC	0x003A	PITS0
0x000B	S2ADDR	0x001B	ADCC2	0x002B	-	0x003B	-
0x000C	S2ADEN	0x001C	PWM0DBC	0x002C	CLKPCKEN0	0x003C	-
0x000D	S2BRTH	0x001D	PWM1DBC	0x002D	CLKPCKEN1	0x003D	-
0x000E	S2BRTL	0x001E	PWM2DBC	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	P4M0	0x0030	-
0x0001	P0M1	0x0011	P2M1	0x0021	P4M1	0x0031	-
0x0002	P0M2	0x0012	P2M2	0x0022	-	0x0032	-
0x0003	P0M3	0x0013	P2M3	0x0023	P4M3	0x0033	-
0x0004	P0HPU	0x0014	P2HPU	0x0024	P4HPU	0x0034	-
0x0005	P0LPU	0x0015	P2LPU	0x0025	P4LPU	0x0035	-
0x0006	-	0x0016	-	0x0026	-	0x0036	-
0x0007	-	0x0017	-	0x0027	-	0x0037	-
0x0008	P1M0	0x0018	P3M0	0x0028	P5M0	0x0038	-
0x0009	P1M1	0x0019	P3M1	0x0029	P5M1	0x0039	-
0x000A	P1M2	0x001A	P3M2	0x002A	P5M2	0x003A	-
0x000B	P1M3	0x001B	P3M3	0x002B	-	0x003B	-
0x000C	P1HPU	0x001C	P3HPU	0x002C	P5HPU	0x003C	-
0x000D	P1LPU	0x001D	P3LPU	0x002D	P5LPU	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	INT0_MAP
0x0001	T1_MAP	0x0011	PWM01_MAP	0x0021	RXD_MAP	0x0031	INT1_MAP
0x0002	-	0x0012	FLT0_MAP	0x0022	SCL_MAP	0x0032	-
0x0003	-	0x0013	-	0x0023	SDA_MAP	0x0033	-
0x0004	RTCO_MAP	0x0014	PWM1_MAP	0x0024	$\overline{SS}$_MAP	0x0034	-
0x0005	BRTO_MAP	0x0015	PWM11_MAP	0x0025	SCK_MAP	0x0035	-
0x0006	-	0x0016	FLT1_MAP	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	ECI_MAP	0x001A	FLT2_MAP	0x002A	-	0x003A	-
0x000B	PCA0_MAP	0x001B	-	0x002B	-	0x003B	-
0x000C	PCA1_MAP	0x001C	-	0x002C	-	0x003C	-
0x000D	-	0x001D	-	0x002D	-	0x003D	-
0x000E	-	0x001E	-	0x002E	-	0x003E	-
0x000F	-	0x001F	-	0x002F	-	0x003F	-

Read only:

Extension XSFR (base address is 0xFFC0)

Offset address	XSFR	Offset address	XSFR	Offset address	XSFR	Offset address	XSFR
0x0000	SN_DATA0	0x0010	CHIP_ID0	0x0020	-	0x0030	-
0x0001	SN_DATA1	0x0011	CHIP_ID1	0x0021	-	0x0031	-
0x0002	SN_DATA2	0x0012	CHIP_ID2	0x0022	-	0x0032	-
0x0003	SN_DATA3	0x0013	CHIP_ID3	0x0023	-	0x0033	-
0x0004	SN_DATA4	0x0014	CHIP_ID4	0x0024	-	0x0034	-
0x0005	SN_DATA5	0x0015	CHIP_ID5	0x0025	-	0x0035	-
0x0006	SN_DATA6	0x0016	CHIP_ID6	0x0026	-	0x0036	-
0x0007	SN_DATA7	0x0017	CHIP_ID7	0x0027	-	0x0037	-
0x0008	ID_DATA0	0x0018	-	0x0028	-	0x0038	-
0x0009	ID_DATA1	0x0019	-	0x0029	-	0x0039	-
0x000A	ID_DATA2	0x001A	-	0x002A	-	0x003A	-
0x000B	ID_DATA3	0x001B	-	0x002B	-	0x003B	-
0x000C	ID_DATA4	0x001C	-	0x002C	-	0x003C	-
0x000D	ID_DATA5	0x001D	-	0x002D	-	0x003D	-
0x000E	ID_DATA6	0x001E	-	0x002E	-	0x003E	-
0x000F	ID_DATA7	0x001F	-	0x002F	-	0x003F	-

4 The system clock

4.1 Characteristics of the system clock

HC89S105xx MCU system clock have 3 optional clock sources: external high-frequency RC clock (4MHz~16MHz), external low-frequency RC clock (32.768KHz), internal high-frequency RC clock (32 MHz) and internal low frequency RC clock (38KHz). The internal frequency RC scope of error is less than 1% during -40°C ~+85°C. The clock selected by the system is denoted as Fosc, and Fosc can perform frequency division of any value between 1-255. The clock after frequency division is denoted as CPU clock, that is, Fcpu.

If the internal 32MHz RC is used as the system clock, the system clock must be reduced to 16MHz or below by frequency division. If the system clock is higher than 16MHz, it may cause the system to work incorrectly.

Internal low frequency RC (38KHz) output clock marker as wdt_clk, for the watchdog timer count.

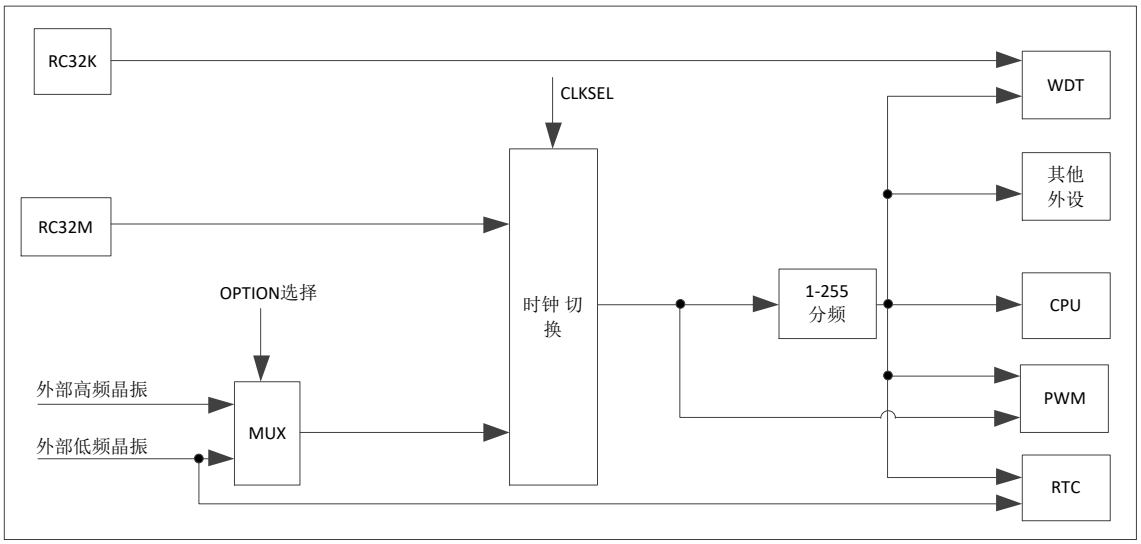


Figure 4 - 1 System clock block diagram

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	-			

Bit	Flag	Introductions
7-6	CLKSTA[1:0]	System clock state bits x0: 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	-	Reserved bit
4	CLKSEL[1:0]	System clock selection bit 0: Select system clock to internal high frequency RC 1: Select system clock to external crystal oscillator
3-0	-	Reserved bit

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 16 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 Peripheral clock gating register (CLKPCKEN0、CLKPCKEN1)

Power consumption can be reduced by turning off clock without peripherals, while clock gating registers of peripherals allow users to turn on or off the connection between system clock and peripherals at

any time in operation mode. When the user turns off a peripheral's clock, the module is disabled and registers that operate the module are disabled.

After the system is reset, all peripheral clocks are on. The user can turn off the corresponding peripheral clock by clearing the peripheral clock control bit in CLKPCKEN0 or CLKPCKEN1.

CLKPCKEN0

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	UART2_ CLKEN	UART1_ CLKEN	WDT_ CLKEN	RTC_ CLKEN	PWM_ CLKEN	PCA_ CLKEN	T1_ CLKEN	T0_ CLKEN

Bit	Flag	Introductions
7	UART2_CLKEN	UART2 clock enablement bit 0: Disable peripheral clock connection with UART2 1: Enable the peripheral clock to connect with UART2
6	UART1_CLKEN	UART1 clock enablement bit 0: Disable peripheral clock connection with UART1 1: Enable the peripheral clock to connect with UAR1
5	WDT_CLKEN	WDT clock enablement bit 0: Disable peripheral clock connection with WDT 1: Enable the peripheral clock to connect with WDT
4	RTC_CLKEN	RTC clock enablement bit 0: Disable peripheral clock connection with RTC 1: Enable the peripheral clock to connect with RTC
3	PWM_CLKEN	PWM clock enablement bit 0: Disable peripheral clock connection with PWM 1: Enable the peripheral clock to connect with PWM
2	PCA_CLKEN	PCA clock enablement bit 0: Disable peripheral clock connection with PCA 1: Enable the peripheral clock to connect with PCA
1	T1_CLKEN	T1 clock enablement bit 0: Disable peripheral clock connection with T1 1: Enable the peripheral clock to connect with T1
0	T0_CLKEN	T0 clock enablement bit 0: Disable peripheral clock connection with T0 1: Enable the peripheral clock to connect with T0

CLKPCKEN1

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	-						SPI_ CLKEN	IIC_ CLKEN

Bit	Flag	Introductions
3-0	-	Reserved bit
6	SPI__CLKEN	SPI clock enablement bit

		0: Disable peripheral clock connection with SPI 1: Enable the peripheral clock to connect with SPI
5	IIC_CLKEN	IIC clock enablement bit 0: Disable peripheral clock connection with IIC 1: Enable the peripheral clock to connect with IIC

4.2.5 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 HC89S105xx 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 HC89S105xx 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 HC89S105xx 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

HC89S105xx 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 enabled and permits working in power down mode, then the WDT 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, HC89S105xx 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, HC89S105xx 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 and RTC interrupt. After effective external interrupts and RTC interrupts occur, the oscillator is started, the CPU clock and peripheral clock are restored immediately, the PD bit in the PCON register is cleared by the hardware, and the program runs the external interrupt service program. After completing the external interrupt service routine, jump to the instruction after entering power off mode to continue running. Run 30 s for the next shutdown.

(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 HC89S105xx 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 HC89S105xx 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.

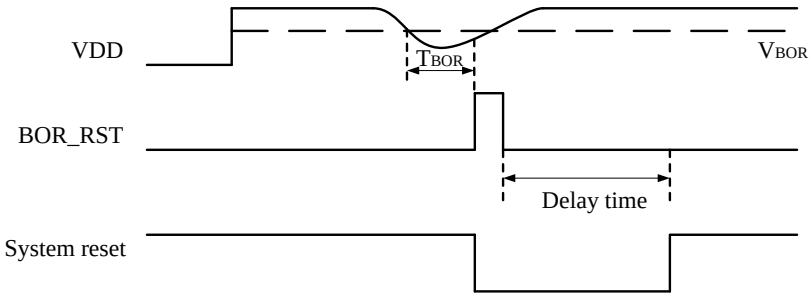


Figure 6 - 1 BOR schematic diagram

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. P4.7 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 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.

6.6 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.

6.7 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	R
POR Reset	1	x	x	x	x	0	0	0
EXRST Reset	u	1	u	u	u	0	0	0
BOR Reset	u	u	1	u	u	0	0	0
WDT Reset	u	u	u	1	u	0	0	0
Soft reset	u	u	u	u	1	0	0	0
Flag	PORF	EXRSTF	BORF	WDTRF	SWRF	-	-	-

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
0-2	-	Reserved

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.

7 General and multiplexed I/O

7.1 General and multiplexed I/O characteristics

- Provides 41/39 bi-directional I/O ports
- Multiple modes configuration

7.2 I/O mode

All HC89S105xx 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 HC89S105xx the power-on reset, all IO states is set as analog input default.

When HC89S105xx 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

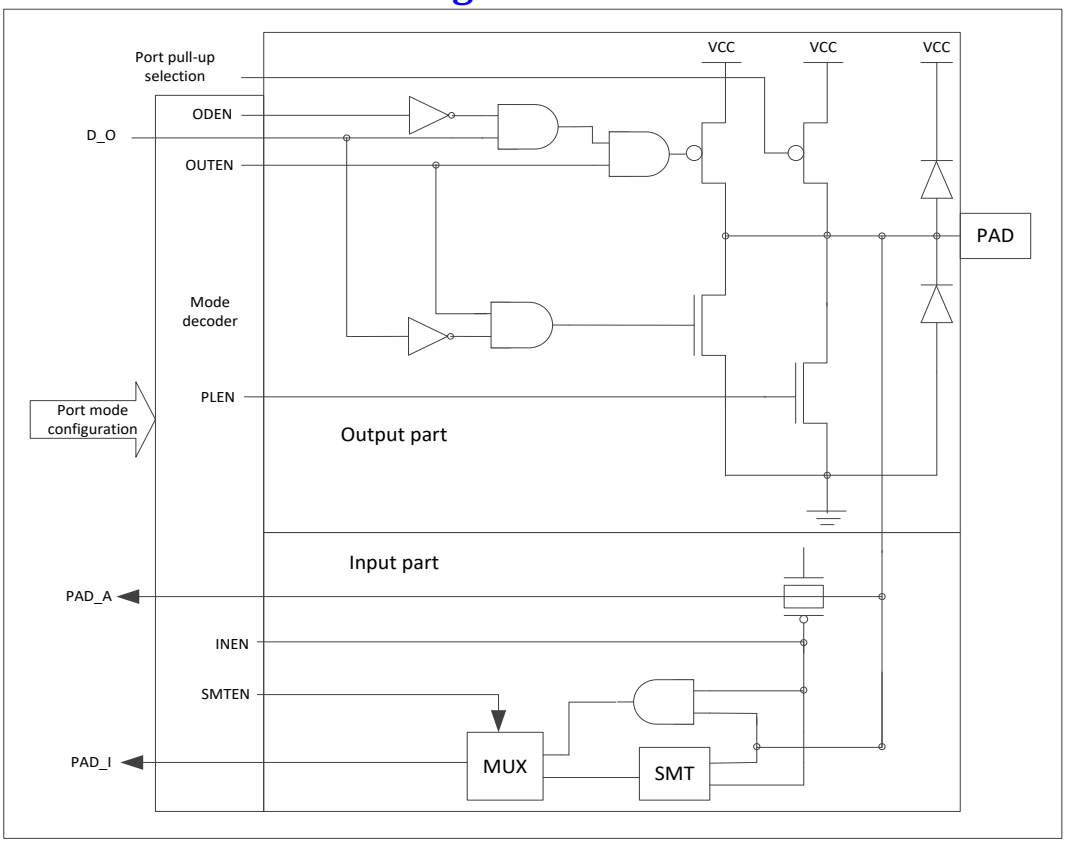


Figure 7 - 1 I/O function block diagram

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

Note: For HC89S105K8, P0 [2] cannot be operated. Please do not operate anything else

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

Note: For HC89S105K8, P1 [4:0] cannot be operated. Please do not operate anything else

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

Note: For HC89S105C8/S8, only P2 [7:1] can be operated. Please do not operate anything else
Note: For HC89S105K8, only P2 [5:1] can be operated. Please do not operate anything else

7.4.4 P3 port data register P3

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

7.4.5 P4 port data register P4

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
7-4	-	Reserved bit
3-2	P4[3:2]	P4 port data register
1	-	Reserved bit
0	P4[0]	P4 port data register

Note: For HC89S105C8/S8, only P4 [0] and P4 [3:2] can be operated. Please do not operate anything else
Note: For HC89S105K8, only P4 [2] can be operated. Please do not operate anything else

7.4.6 P5 port data register P5

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	-		P5 [5:0]					

Bit	Flag	Introductions
7-6	-	Reserved bit
5-0	P5 [5:0]	P5 port data register

Note: For HC89S105K8, only P3 [5:3] can be operated. Please do not operate anything else

7.4.7 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)

		1000 : Push-pull output(IO drives High Drive Mode) 1001 : Open drain output(IO drives High Drive Mode) 1010 : open drain output with pull-up(IO drives High Drive Mode) 1011 : Reserved 1100 : Push-pull output(IO drives Low Drive Mode) 1101: Open drain output(IO drives Low Drive Mode) 1110: open drain output with pull-up(IO drives Low Drive Mode) 1111: Reserved

7.4.8 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]			

P1M1

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	P13M[3:0]				P12M[3:0]			

P1M2

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	P15M[3:0]				P14M[3:0]			

P1M3

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	P17M[3:0]				P16[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) 1000 : Push-pull output(IO drives High Drive Mode) 1001 : Open drain output(IO drives High Drive Mode) 1010 : open drain output with pull-up(IO drives High Drive Mode) 1011 : Reserved 1100 : Push-pull output(IO drives Low Drive Mode)

		1101: Open drain output(IO drives Low Drive Mode) 1110: open drain output with pull-up(IO drives Low Drive Mode) 1111: Reserved

Note: For HC89S105K8, P1M2 [7:4] and P1M3 [7:0] cannot be operated.

7.4.9 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]				-			

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]			

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) 1000 : Push-pull output(IO drives High Drive Mode) 1001 : Open drain output(IO drives High Drive Mode) 1010 : open drain output with pull-up(IO drives High Drive Mode) 1011 : Reserved 1100 : Push-pull output(IO drives Low Drive Mode) 1101: Open drain output(IO drives Low Drive Mode) 1110: open drain output with pull-up(IO drives Low Drive Mode) 1111: Reserved

Note: For HC89S105C8/S8, P2M0 [3:0] cannot be operated.
Note: For HC89S105K8, P2M0 [3:0] and P2M3 [7:0] cannot be operated.

7.4.10 P3 port function select register P3M0, P3M1, P3M2, P3M3

P3M0

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	P31M[3:0]				P30M[3:0]			

P3M1

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	P33M[3:0]				P32M[3:0]			

P3M2

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	P35M[3:0]				P34M[3:0]			

P3M3

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	P37M[3:0]				P36M[3:0]			

Bit	Flag	Introductions
7-4 3-0	P3xM [3:0] (x = 0...7)	P3.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) 1000 : Push-pull output(IO drives High Drive Mode) 1001 : Open drain output(IO drives High Drive Mode) 1010 : open drain output with pull-up(IO drives High Drive Mode) 1011 : Reserved 1100 : Push-pull output(IO drives Low Drive Mode) 1101: Open drain output(IO drives Low Drive Mode) 1110: open drain output with pull-up(IO drives Low Drive Mode) 1111: Reserved

7.4.11 P4 port function select register P4M0, P4M1

P4M0

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	-				P40M[3:0]			

P4M1

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	P43M[3:0]				P42M[3:0]			

Bit	Flag	Introductions
7-4 3-0	P4xM [3:0] (x = 0,1,2,6)	P4.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) 1000 : Push-pull output(IO drives High Drive Mode) 1001 : Open drain output(IO drives High Drive Mode) 1010 : open drain output with pull-up(IO drives High Drive Mode) 1011 : Reserved 1100 : Push-pull output(IO drives Low Drive Mode) 1101: Open drain output(IO drives Low Drive Mode) 1110: open drain output with pull-up(IO drives Low Drive Mode) 1111: Reserved

Note: For HC89S105C8/S8, only P4M0 [3:0] and P4M1 [3:0] can be operated. Please do not operate anything else

Note: For HC89S105K8, only P4M1 [3:0] can be operated. Please do not operate anything else

7.4.12 P5 port function select register P5M0, P5M1, P5M2

P5M0

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	P51M[3:0]				P50M[3:0]			

P5M1

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	P53M[3:0]				P52M[3:0]			

P5M2

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	P55M[3:0]				P54M[3:0]			

Bit	Flag	Introductions
7-4 3-0	P5xM [3:0] (x = 0...5)	P5.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) 1000 : Push-pull output(IO drives High Drive Mode) 1001 : Open drain output(IO drives High Drive Mode) 1010 : open drain output with pull-up(IO drives High Drive Mode) 1011 : Reserved 1100 : Push-pull output(IO drives Low Drive Mode) 1101: Open drain output(IO drives Low Drive Mode) 1110: open drain output with pull-up(IO drives Low Drive Mode) 1111: Reserved

Note: For HC89S105K8, only P5M1 [7:4] can be operated. Please do not operate anything else

7.4.13 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	P03PU[1:0]		P02PU[1:0]		P01PU[1:0]		P00PU[1:0]	

Note: For HC89S105K8, P1LPU [5:4] cannot be operated.

P0HPU

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	P07PU[1:0]		P06PU[1:0]		P05PU[1:0]		P04PU[1:0]	

P1LPU

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	P13PU[1:0]		P12PU[1:0]		P11PU[1:0]		P10PU[1:0]	

P1HPU

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	P17PU[1:0]		P16PU[1:0]		P15PU[1:0]		P14PU[1:0]	

Note: For HC89S105K8, only P1HPU [1:0] can be operated. Please do not operate anything else

P2LPU

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	P23PU[1:0]		P22PU[1:0]		P21PU[1:0]		-	-

P2HPU

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	P27PU[1:0]		P26PU[1:0]		P25PU[1:0]		P24PU[1:0]	

Note: For HC89S105K8, only P2HPU [3:0] can be operated. Please do not operate anything else

P3LPU

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	P33PU[1:0]		P32PU[1:0]		P31PU[1:0]		P30PU[1:0]	

P3HPU

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	P37PU[1:0]		P36PU[1:0]		P35PU[1:0]		P34PU[1:0]	

P4LPU

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	P43PU[1:0]		P42PU[1:0]		-		P40PU[1:0]	

Note: For HC89S105C8/S8, only P4LPU [1:0] and P4HPU [7:4] can be operated. Please do not operate anything else

Note: For HC89S105K8, only P4LPU [5:4] can be operated. Please do not operate anything else

P5LPU

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	P53PU[1:0]		P52PU[1:0]		P51PU[1:0]		P50PU[1:0]	

Note: For HC89S105K8, only P5LPU [7:6] can be operated. Please do not operate anything else

P5HPU

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	-	-	-	-	P55PU[1:0]		P54PU[1:0]	

Bit	Flag	Introductions
7-6 5-4 3-2 1-0	PxyPU[1:0] (x=0,1,2,3,4,5) (y=0…7)	Port pull-up resistance selection bit 00 : 30 KΩ 01 : 50 KΩ 10 : 120 KΩ 11 : 230 KΩ Note: Resistance is the reference value at VDD @5V.

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	INT0_MAP
0xFF81	T1_MAP	0xFF91	PWM01_MAP	0xFFA1	RXD_MAP	0xFFB1	INT1_MAP
0xFF82	-	0xFF92	FLT0_MAP	0xFFA2	SCL_MAP	0xFFB2	-
0xFF83	-	0xFF93	-	0xFFA3	SDA_MAP	0xFFB3	-
0xFF84	RTCO_MAP	0xFF94	PWM1_MAP	0xFFA4	$\overline{SS}$_MAP	0xFFB4	-
0xFF85	BRTO_MAP	0xFF95	PWM11_MAP	0xFFA5	SCK_MAP	0xFFB5	-
0xFF86	-	0xFF96	FLT1_MAP	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	ECI_MAP	0xFF9A	FLT2_MAP	0xFFAA	-	0xFFBA	-
0xFF8B	PCA0_MAP	0xFF9B	-	0xFFAB	-	0xFFBB	-
0xFF8C	PCA1_MAP	0xFF9C	-	0xFFAC	-	0xFFBC	-
0xFF8D	-	0xFF9D	-	0xFFAD	-	0xFFBD	-
0xFF8E	-	0xFF9E	-	0xFFAE	-	0xFFBE	-
0xFF8F	-	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[2:0]			FPIN [2:0]		

Bit	Flag	Introductions
7-6	-	Reserved bit
5-3	FPORT[2:0]	Mapping port selection bit 000: P0 001: P1 010: P2 011: P3 100: P4 101: P5
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 0xFF, 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 P3.1 P3.2 pin, the user must do the following configuration before start to use UART1:

```
MOV A,        #0x31     //TXD-->P3.1
MOV DPTR,  #0XFFA0
MOVX          @DPTR,A
MOV A,        #0x32      //RXD-->P3.2
MOV DPTR,  #0XFFA1
MOVX          @DPTR,A
```

User need to map UART1 TXD RXD to P4.4 P4.5 pin in the next design, the user must do the following configuration:

```
MOV A,        #0x44     //TXD-->P4.4
MOV DPTR,  #0XFFA0
MOVX          @DPTR,A
MOV A,        #0x45     //RXD-->P4.5
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	T0
2	T1
3	RTCO
4	BRTO
5	PCA0
6	PCA1
7	PWM0
8	PWM01
9	PWM1
10	PWM11
11	PWM2
12	PWM21
13	TXD
14	RXD
15	SCL
16	SDA
17	SCK
18	MOSI
19	MISO
20	TXD2
21	RXD2

For example: The RTCO_MAP low 6 bit configuration is 000001, select P0.1 as RTCO output port, The BRTO _MAP low 6 bit also is configured to 000001, as the priorities above, the hardware will configure P0.1 as RTCO output port, and BRTO _MAP configuration is invalid.

When all the port-mapped control registers is not equal 000001, 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 0x13, need select P1.3 as T0 input port, FLT0_MAP is also configured to 0x13, and the signal into to P1.3 port is valid to T0 and FLT0 simultaneously..

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
- 15 external interrupts

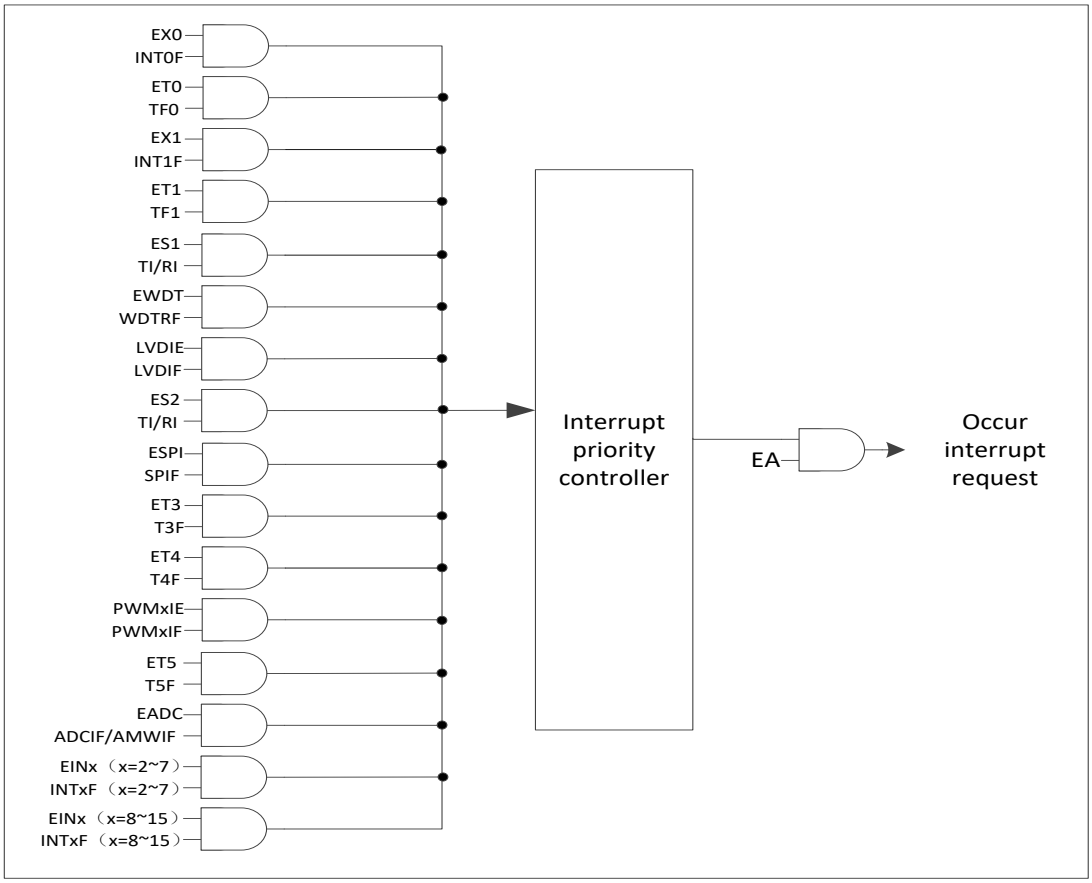


Figure 8 – 1 Interrupt function block diagram

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	ET1	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/Pattern of conflict	9	8
IIC	004BH	EIIC	SI	10	9
PCA	0053H	ECF	PCAF	11	10
PWM	005BH	PWMxIE (x =0/1/2)	PWMxIF (x =0/1/2)	12	11
RTC	0063H	ERTC	RTCF	13	12

ADC	0073H	EADC	ADCIF/AMxWIF (x = 0,1)	14	14
INT2-INT7	007BH	EINx (x =2...7)	INTxF (x = 2...7)	15	15
INT9-INT15	0083H	EINx (x =9...15)	INTxF (x = 9...15)	16	16

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

HC89S105xx 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.

If the external interrupt is triggered at a low level, the external interrupt source must remain valid until the requested interrupt is generated, a process that requires two system clock cycles. If the interrupt service completes and the external interrupt persists, the next interrupt occurs. If you wake up the power mode with the drop edge external interrupt, you must keep the external interrupt pin high before entering the power down mode, and you need to turn on the interrupt enable and interrupt allow registers.

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	-	Reserved bit
6	EX9_15	External interrupt 9~15 interrupt enable bit 0 : Disable INT9~INT15 interrupts 1 : Enable INT9~INT15 interrupts Note: INT9~INT15 share the same interrupt vector.
5	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.
4	EADC	A/D Conversion complete interrupt enable bit 0 : Disable A/D interrupt 1 : Enable A/D interrupt
3	-	Reserved bit
2	ERTC	RTC interrupt enable bit 0 : Disable RTC interrupt 1 : Enable RTC interrupt
1	EIIC	IIC interrupt enable bit 0 : Disable IIC interrupt 1 : Enable IIC 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]		PPCA[1:0]		PIIC[1:0]		PSPI[1:0]	

Bit	Flag	Introductions
7-6	PPWM [1:0]	PWM interrupt priority control bits
5-4	PPCA [1:0]	PPCA interrupt priority control bits
3-2	PIIC[1:0]	PIIC 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	PX2_7[1:0]		PADC[1:0]		-		PRTC[1:0]	

Bit	Flag	Introductions
7-6	PX2_7 [1:0]	PX2_7 interrupt priority control bits
5-4	PADC [1:0]	PADC interrupt priority control bits
3-2	-	Reserved bit
1-0	PRTC [1:0]	PRTC interrupt priority control bits

IP4

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	-						PX9_15[1:0]	

Bit	Flag	Introductions
7-2	-	Reserved bit
1-0	PX9_15 [1:0]	PRTC 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 values	0	0	0	0	0	0	0	0
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]		-	

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	-

Bit	Flag	Introductions
7-0	EINTx (x =8...15)	External interrupt control bits (INT9~INT15) 0 : Disable the port interrupts 1 : Enable the port interrupts Note: 1. As long as the corresponding EINTx (x =9...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	-

Bit	Flag	Introductions
7-0	INTxF (x =9...15)	INT9-INT15 interrupt request flag bits 0： Software clear 0 1 : When external interrupts occur, hardware set 1

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

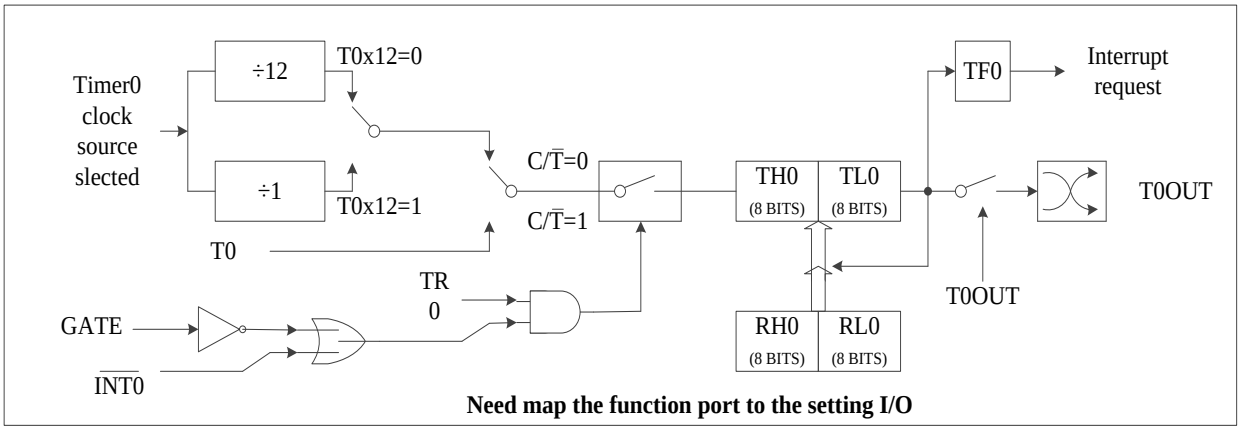


Figure 9 - 1 Timer0 mode0 function block diagram

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

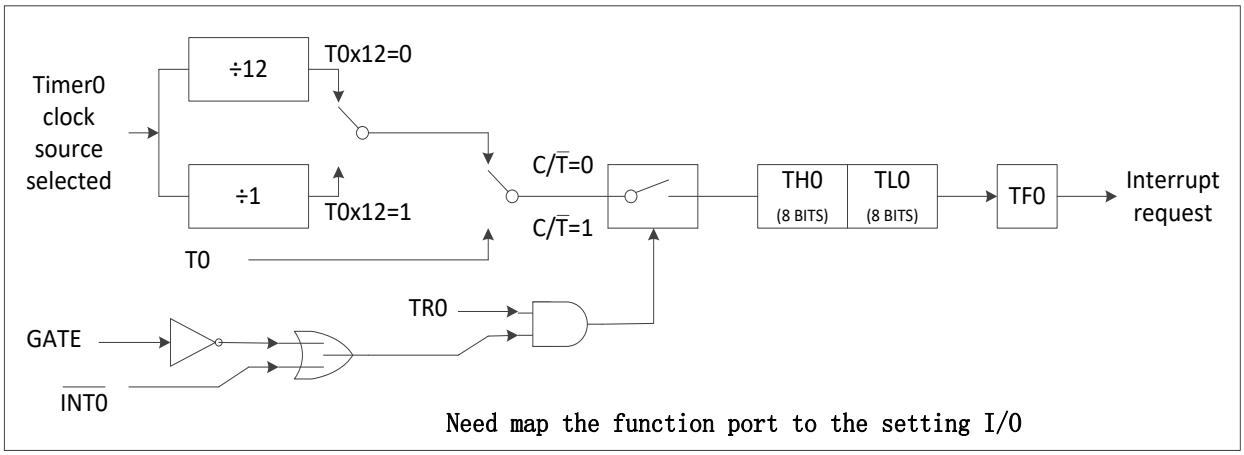


Figure 9 - 2 Timer0 1 mode1 function block diagram

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

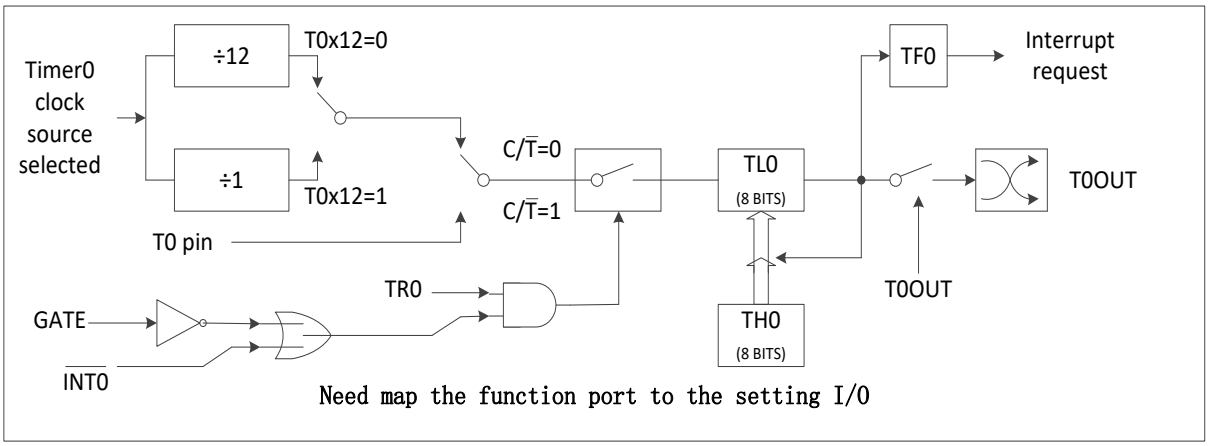


Figure 9 - 3 Timer0 mode2 Function block diagram

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)

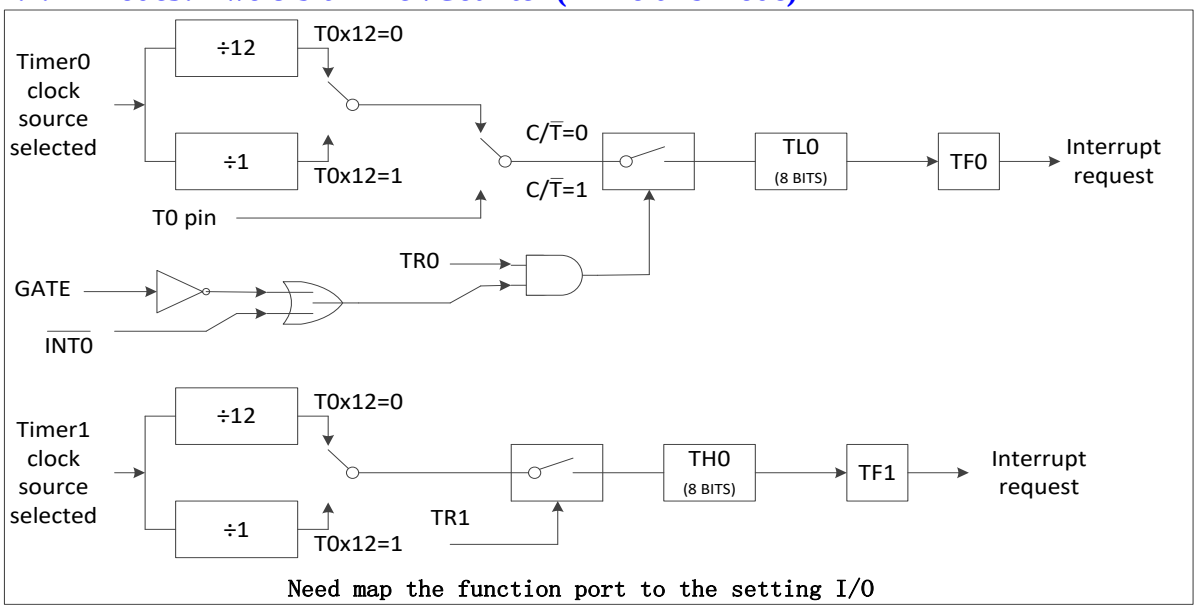


Figure 9 - 4 Timer0 3 function block diagram

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{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

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

HC89S105xx 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.

If EPWM0 is enabled and PWM0 is not mapped through the functional pin mapping register, then PWM0 will not be output from the chip pin,And this time PWM0 Counter can be used as a Timer, when counter overflow, PWM interrupt occur when interrupt is enabled.

Set EFLT0/EFLT1/EFLT2 to 1, PWM0/PWM1/PWM2 output and its complementary output can be closed by input signal variation on FLT0/FLT1/FLT2 pin automatically. Once detected valid input on FLT 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 FLT input signal valid period, FLTS cannot be cleared. Only when the FLT input signal invalid, FLTS status can be cleared by software, and PWM return to output.

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 registers

10.2.1 PWMEN

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	PWM0FLT_EN	EPWM21		EPWM11	EPWM01	-	EPWM2	EPWM1

Bit	Flag	Introductions
7	PWM0FLT_EN	PWM0 fault detection enable bit 0: Disable fault detection 1: Allow fault detection. Pin mapping and IO mode setting of FLT0 are required first
6	EPWM21	PWM21 enable control bit 0: disable PWM21 output 1: allow PWM21 output and configure the pin mapping for PWM21
4	EPWM11	PWM11 enable control bit 0: disable PWM11 output 1: allow PWM11 output and configure the pin mapping for PWM11
4	EPWM01	PWM01 enable control bit 0: disable PWM01 output 1: allow PWM01 output and configure the pin mapping for PWM01
3	-	Reserved bit
2	EPWM2	PWM2 enable control bit 0: disable PWM2 output

		1: allow PWM2 output and configure the pin mapping for PWM2
1	EPWM1	PWM1 enable control bit 0: disable PWM1 output 1: allow PWM1 output and configure the pin mapping for PWM1
0	EPWM0	PWM0 enable control bit 0: disable PWM0 output 1: allow PWM0 output and configure the pin mapping for PWM0

10.2.2 PWMELT

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	PWM2 FLT_EN	PWM1 FLT_EN	PWM2_FLT_MODE		PWM1_FLT_MODE		PWM0_FLT_MODE	

Bit	Flag	Introductions
7	PWM2FLT_EN	PWM2 fault detection enable bit 0: Disable fault detection 1: Allow fault detection. Pin mapping and IO mode setting of FLT2 are required first
6	PWM1FLT_EN	PWM1 fault detection enable bit 0: Disable fault detection 1: Allow fault detection. Pin mapping and IO mode setting of FLT1 are required first
5-4	PWM2_FLT_MODE	PWM2 output fault reserve status choose a 00: PWM2 & PWM21 during the fault are of low level 01: PWM2 fault during the low level, PWM21 fault during high level 10: PWM2 fault during the high level, PWM21 fault during low level 11: PWM2 & PWM21 during the fault are of high level
3-2	PWM1_FLT_MODE	PWM1 output fault reserve status choose a 00: PWM1 & PWM11 during the fault are of low level 01: PWM1 fault during the low level, PWM11 fault during high level 10: PWM1 fault during the high level, PWM11 fault during low level 11: PWM1 & PWM11 during the fault are of high level
1-0	PWM0_FLT_MODE	PWM0 output fault reserve status choose a 00: PWM0 & PWM01 during the fault are of low level 01: PWM0 fault during the low level, PWM01 fault during high level 10: PWM0 fault during the high level, PWM01 fault during low level 11: PWM0 & PWM01 during the fault are of high level

10.2.3 PWM0 module

10.2.3.1 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	PWM0FLTS	PWM0FLTC	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：Fosc/2 01：Fosc/4 10：Fosc/8 11：Fosc/16

10.2.3.2 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

10.2.3.3 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

10.2.3.4 PWM0 dead time register PWM0DT

PWM0DT

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	PWM0DT[7:0]							

Bit	Flag	Introductions
7-0	PWM0DT[7:0]	PWM0 Dead time register Note: The clock source is Fosc without any frequency division.

10.2.3.5 PWM0 fault input pin buffeting control register PWM0DBC

PWM0DBC

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	PWM0DBCLK[1:0]		PWM0DBCT[5:0]					

Bit	Flag	Introductions
7-6	PWM0DBCLK[1:0]	Port buffeting clock selection 00：F_{CPU} /1 01：F_{CPU} /4 10：F_{CPU} /16 11：F_{CPU} /64
5-0	PWM0DBCT[5:0]	The number of port buffering counting clocks, when configured to 00, indicates buffering.

Shaking time = elimination frequency coefficient * TCPU * PWM0DBCT [beat]

Note: shake time is not accurate, real number away shaking between the configuration value minus one and configuration values.

10.2.4 PWM1 module

10.2.4.1 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	PWM1FLTS	PWM1FLTC	PWM1S		CK0	

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	PWM1FLTS	PWM1 FLT state bits 0 : PWM normal state, software clear 0 1 : PWM output off, hardware set 1
4	PWM1FLTC	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
1-0	CK0	PWM1 clock source selection bits 00 : F_{osc} /2 01 : F_{osc}/4 10 : F_{osc}/8 11: F_{osc}/16

10.2.4.2 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

10.2.4.3 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

10.2.4.4 PWM1 dead time register PWM1DT

PWM1DT

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	PWM1DT[7:0]							

Bit	Flag	Introductions
7-0	PWM1DT[7:0]	PWM1 Dead time register Note: The clock source is Fosc without any frequency division.

10.2.4.5 PWM1 fault input pin buffeting control register PWM1DBC

PWM1DBC

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	PWM1DBCLK[1:0]		PWM1DBCT[5:0]					

Bit	Flag	Introductions
7-6	PWM1DBCLK[1:0]	Port buffeting clock selection 00：F_{CPU} /1 01：F_{CPU} /4 10：F_{CPU} /16 11：F_{CPU} /64
5-0	PWM1DBCT[5:0]	The number of port buffering counting clocks, when configured to 00, indicates buffering.

Shaking time = elimination frequency coefficient * TCPU * PWM0DBCT [beat]

Note: shake time is not accurate, real number away shaking between the configuration value minus one and configuration values.

10.2.5 PWM2 module

10.2.5.1 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	PWM2FLTS	PWM2FLTC	PWM2S		CK0	

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	PWM2FLTS	PWM2 FLT state bit 0 : PWM normal state, software clear 0 1 : PWM output off, hardware set 1
4	PWM2FLTC	PWM2 FLT pin configuration bit 0 : When FLT2 is low level, PWM output off 1 : When FLT2 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	CK0	PWM2 Clock source selection bits 00 : F_{osc} /2 01 : F_{osc}/4 10 : F_{osc}/8 11 : F_{osc}/16

10.2.5.2 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

10.2.5.3 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

10.2.5.4 PWM2 dead time register PWM2DT

PWM2DT

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	PWM2DT[7:0]							

Bit	Flag	Introductions
7-0	PWM2DT[7:0]	PWM2 Dead time register Note: The clock source is Fosc without any frequency division.

10.2.5.5 PWM2 fault input pin buffeting control register PWM2DBC

PWM2DBC

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	PWM2DBCLK[1:0]		PWM2DBCT[5:0]					

Bit	Flag	Introductions
7-6	PWM2DBCLK[1:0]	Port buffeting clock selection 00：F_{CPU} /1 01：F_{CPU} /4 10：F_{CPU} /16 11：F_{CPU} /64
5-0	PWM2DBCT[5:0]	The number of port buffering counting clocks, when configured to 00, indicates buffering.

Shaking time = elimination frequency coefficient * TCPU * PWM0DBCT [beat]

Note: shake time is not accurate, real number away shaking between the configuration value minus one and configuration values.

11 Programmable counter array PCA

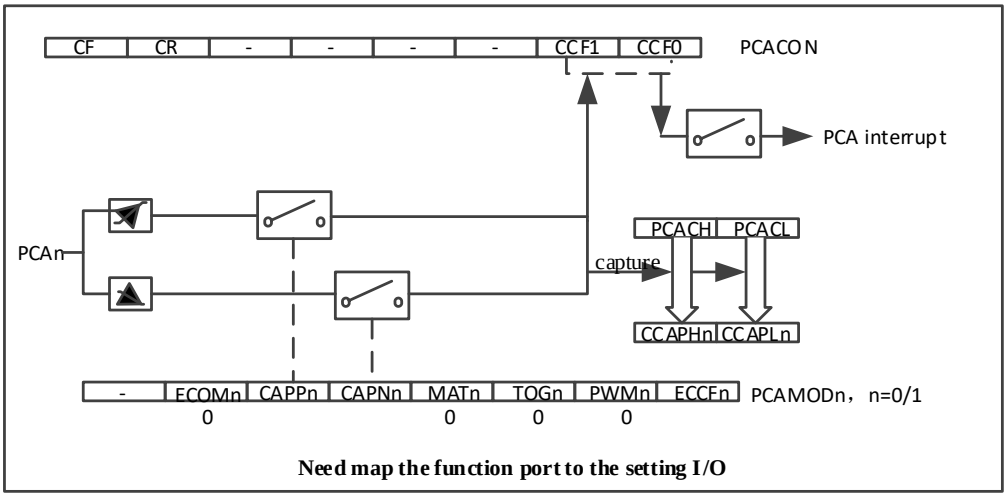
11.1 PCA Characteristics

The HC89S105xx MCU has a 2-way programmable counter array (PCA), which contains a special 16-bit timer and two 16-bit capture/comparison modules connected to it. Each module can be programmed to work in four modes: capture, software timer, high-speed output and modulable pulse output.

11.2 PCA working mode

11.2.1 Capture Mode

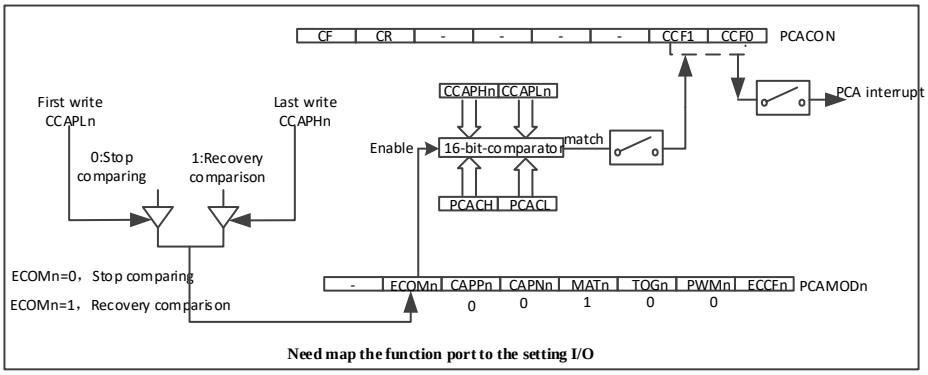
For a PCA module to work in capture mode, the two bits (CAPNn and CAPPn) of the register PCAMODn (n=0, 1) or any of them must be set to 1. When the PCA module works in the capture mode, it will sample the hops of the external PCAn (n=0, 1) input of the module. When the valid hops are sampled, the PCA counting registers (PCACH and PCACL) will be loaded into the capturing registers CCAPHn and CCAPLn (n=0, 1) of the module.



If the bits CCFn (n=0,1) and PCAMODn(n=0,1) in the PCACON register are set, an interrupt will occur. In the interrupt service program, we can determine which module caused the interrupt, and pay attention to the software zeroing of the interrupt flag bit.

11.2.2 Software timer mode

The 16-bit software timer mode structure is shown in the figure below:



By setting the ECOM and MAT bits of the PCAMODn (n=0,1) register, the PCA module can be used as a software timer. When the value of the PCA timer is compared with the value of the capture register, if both

bits CCFn (in the PCACON register, n=0,1) and ECCFn (in the PCAMODn register, n=0,1) are set, an interrupt will occur.

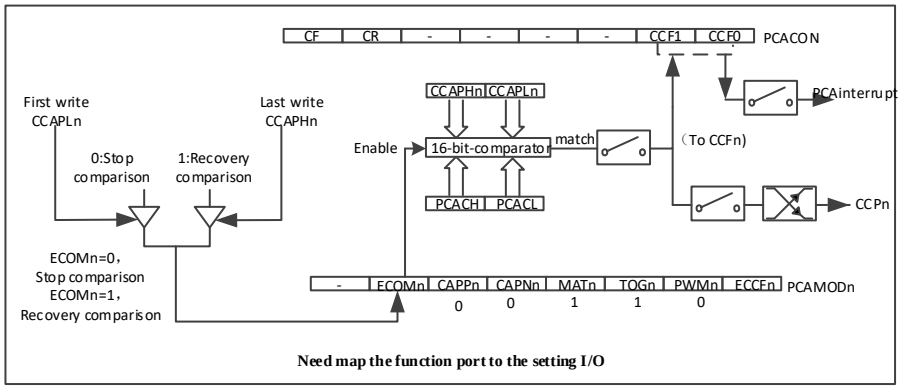
[PCACH,PCACL] automatically increments 1 at regular intervals, depending on the clock source selected. For example, when the clock source is selected as SYSclk/12, add 1 every 12 clock cycles [PCACH,PCACL], and when [PCACH,PCACL] increase to [CCAPHn, CCAPLn] (n=0,1), CCFn=1, resulting in an interrupt request. If [CCAPHn, CCAPLn] (n=0,1) is increased by the same value in the interrupt service program after each PCA module interrupt, then the time between the next interrupt is also the same, thus realizing the timing function. The timing time depends on the choice of clock source and the setting of PCA counter value. The following example illustrates the calculation method of PCA counter value.

Assume that the system clock frequency SYSclk = 18.432mhz, the clock source selected is SYSclk/12, and the timing time T is 5ms, then the PCA counter value is: PCA counter count = T/(1 / (SYSclk) x 12) = 0.005 / ((1/18432000) * 12) = 7680 (decimal) = 1 e00h (hexadecimal number) that is to say, PCA timer count 7680 times, timing time is 5 ms, which is every time to [CCAPHn CCAPLn] value (step).

In operation [CCAPHn,CCAPLn],CCAPLn must be written first, then CCAPHn (n=0,1).

11.2.3 High speed output mode

In this mode, the PCAn(n=0,1) output of the PCA module will be flipped when the count value of the PCA counter matches the value of the capture register. To enable high-speed output mode, the TOGn(n=0,1), MATn, and ECOMn(n=0,1) bits of the PCAMODn register must be set.



The values of CCAPLn and CCAPHn (n=0,1) determine the output pulse frequency of PCA module n. When PCA clock source is SYSclk / 2, the frequency of the output pulse F: F = SYSclk/([CCAPHn CCAPLn] - [PCACH PCACL]), SYSclk for the system clock frequency. Thus, the values of CCAPLn and CCAPHn (n=0,1) can be obtained.

If the result is not an integer, it is rounded to the whole. For example, suppose SYSclk = 20MHz, [PCACH,PCACL] starts counting at 0x0000. If the high speed pulse output frequency of PCA is 125KHz square wave, then the value in [CCAPHn,CCAPLn] (n=0,1) should be: [CCAPHn,CCAPLn] =(1000000/125000)*20/2= 80 = 50H.

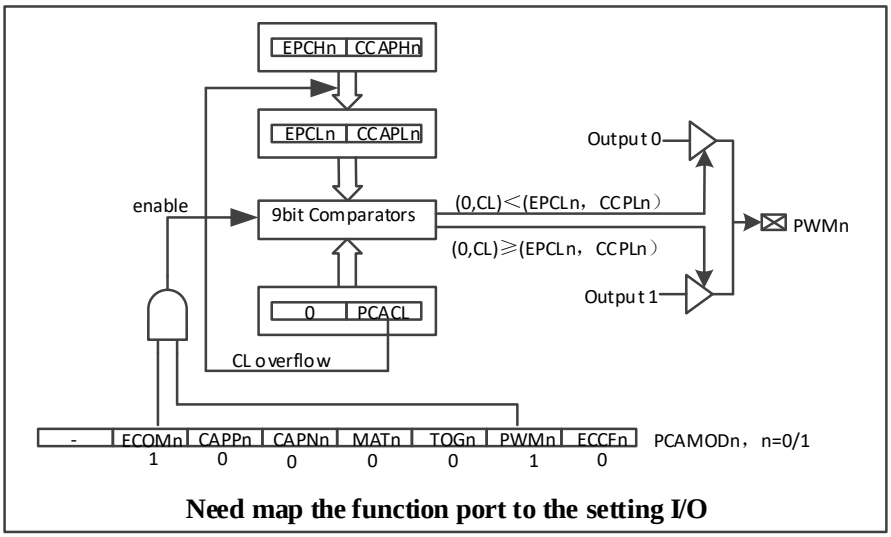
That is, set [CCAPHn,CCAPLn]=0x0050, [PCACH,PCACL] to count from 0x0000, and CCPn port level change occurs when the two match. Because [PCACH,PCACL] is continuously adding 1 action, in order to achieve stable 125KHZ PWM output, it is necessary to make [CCAPHn,CCAPLn] increase 0x0050 or zeroing [PCACH,PCACL] on the basis of the original [CCAPHn,CCAPLn] when the match occurs, to output the required PWM waveform.

in operation [CCAPHn,CCAPLn]，fiest write CCAPLn，last write CCAPHn(n=0,1)。

11.2.4 Pulse width Modulation (PWM)

The PCA module can be configured to operate in 8-bit PWM mode.

8-bit PWM mode is shown in the figure below:



Need map the function port to the setting I/O

All PCA modules can be used as PWM output, and the output frequency depends on the time source of PCA timer. Modules using the same PCA timer have the same output frequency. If different PCA timers are used, different output frequencies can be set. The output duty cycle of each module varies independently and is related to the capture register [EPCLn,CCAPLn] (n=0,1) used.

When the value of register PCACL is less than [EPCLn, CCAPLn] (n=0,1), the output is low; The output is high when the register PCACL is greater than or equal to [EPCLn,CCAPLn]. When the value of the PCACL changes from FF to 00 overflow, the contents of [EPCHn,CCAPHn] are loaded into [EPCLn,CCAPLn]. In this way, PWM can be updated without interference. To enable PWM mode, the PWMn(n=0,1) and ECOMn(n=0,1) bits of the module PCAMODn register must be set.

Because PWM is 8 bits, so the frequency of the PWM = PCA clock input source frequency present 256.

PCA clock input source can choose one from the following eight: SYSclk, SYSclk / 2, SYSclk / 4, SYSclk / 6, 8, SYSclk/SYSclk / 12, timer 0 overflow, ECI input.

Example: PWM output frequency is required to be 38KHz, SYSclk is selected as PCA clock input source, and the value of SYSclk can be obtained. 38000 = SYSclk / 256 by computing formula, get the external clock frequency SYSclk = 38000 x 256 = 9728000 if you want to realize adjustable frequency PWM output, optional timer 0 overflow ratio or ECI input as PCA clock input source.

When EPCLn =0 and CCAPLn = 00H, the PWM fixed output is high; when EPCLn =1 and CCAPLn =0FFH, the PWM fixed output is low.

11.3 PCA registers

11.3.1 PCA control register

PCACON

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	CF	CR	-				CCF0	CCF1

Bit	Flag	Introductions
7	CF	PCA counter array overflow flag bit 0: Software clear 0 1: Hardware set 1, PCA counter overflow when buy 1 if the ECF CMOD register location, CF marks can be used to generate interrupts.
6	CR	PCA run counter array control bit 0: close the PCA counter 1: start the PCA counter
5-2	-	Reserved (read = 0b,, write invalid)
1	CCF1	PCA module 1 interrupt flag 0: Software clear 0 1: Hardware set 1, This position is 1 when a match or capture occurs in PCA module 1
0	CCF0	PCA module 1 interrupt flag 0: Software clear 0 1: Hardware set 1, when there is match or capture PCA module 0 the position 1

11.3.2 PCA clock register

PCACLK

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	CIDL	CR			CPS[2:0]			ECF

Bit	Flag	Introductions
7	CIDL	Whether to stop the control bit of PCA count in idle mode 0: The PCA counter continues to work in idle mode 1: PCA counter stops working in idle mode
5-4	-	Reserved (read = 0b, write invalid)
3-1	CPS[2:0]	PCA count pulse source selection control bit 000：SYSclk/12 001：SYSclk/2 010：Overflow pulse of timer 0 011：The external clock entered by the ECI pin（maximum rate＝SYSclk/2） 100：SYSclk 101：SYSclk/4 110：SYSclk/6 111：SYSclk/8
0	ECF	The PCA count overflows the interrupt enabled bit 0: Disables the interrupt of CF bit in register PCACON 1: Allows the interrupt of CF bit in register PCACON

11.3.3 PCA mode register

PCAMODn (n=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	-	ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn

Bit	Flag	Introductions
7	-	Reserved (read = 0b, write invalid)
6	ECOMn	Comparator function control bit 0: Disable comparator function 1: Allow comparator function
5	CAPPn	The positive capture control bit 0: No rising edge capture 1: Allow rising edge capture
4	CAPNn	Negative capture control bits 0: Do not allow falling edge capture 1: Drop edge capture is allowed
3	MATn	Matching control bit 0: No matching 1: Matching the PCA meter value with the module comparison/capture register value will set the interrupt flag bit CCFn of the PCACON register
2	TOGn	Flip control bit 0: Disable flip 1: Working in PCA high speed output mode, PCAn feet will be flipped if the value of PCA counter matches the value of module comparison/capture register
1	PWMn	Pulse width adjustment mode 0：Disable PWM 1: Allows CEXn feet to be used as pulse width adjustment output
0	ECCFn	CCFn interrupt enabled bit 0: Disable an interrupt of the compare/capture flag CCFn in register PCACON 1: Allows an interrupt to compare/capture flag CCFn in register PCACON

ECOMn	CAPPn	CAPNn	MATn	TOGn	PWMn	ECCFn	module function
0	0	0	0	0	0	0	No such action
1	0	0	0	0	1	0	8 bit PWM，No interruption
1	1	0	0	0	1	1	8 bit PWM output, From low to high can generate interrupt
1	0	1	0	0	1	1	8 bit PWM output, From high to low can generate interrupt
1	1	1	0	0	1	1	8 bit PWM output, From high to low and low to high can also generate interrupt

x	1	0	0	0	0	x	16-bit capture mode, triggered by the rising edge of PCAn/PCAn
x	0	1	0	0	0	x	16-bit capture mode, triggered by the falling edge of PCAn/PCAn
x	1	1	0	0	0	x	16-bit capture mode, triggered by PCAn/PCAn transition
1	0	0	1	0	0	x	16-bit software timer
1	0	0	1	1	0	x	16-bit high-speed output

11.3.4 PCA count register

PCACL

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	PCACL[7:0]							

Bit	Flag	Introductions
7-0	PCACL [7:0]	count register low 8 bits

PCACH

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	PCACH[7:0]							

Bit	Flag	Introductions
7-0	PCACH [7:0]	count register low 8 bits

11.3.5 PCA capture/comparison register CCAPLn(n = 0,1)、CAPHn (n = 0,1)

CCAPLn (n = 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	CCAPLn[7:0] (n = 0,1)							

Bit	Flag	Introductions
7-0	CCAPLn[7:0] (n = 0,1)	PCAn [7:0] (n = 0,1) capture/comparison register low 8 bits

CCAPHn (n = 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	CCAPHn[7:0] (n = 0,1)							

Bit	Flag	Introductions
7-0	CCAPHn[7:0] (n = 0,1)	PCAn [7:0] (n = 0,1) capture/comparison register high 8 bits

11.3.6 PCA mode PWM register PCA_PWMn

PCA_PWMn (n = 0,1)

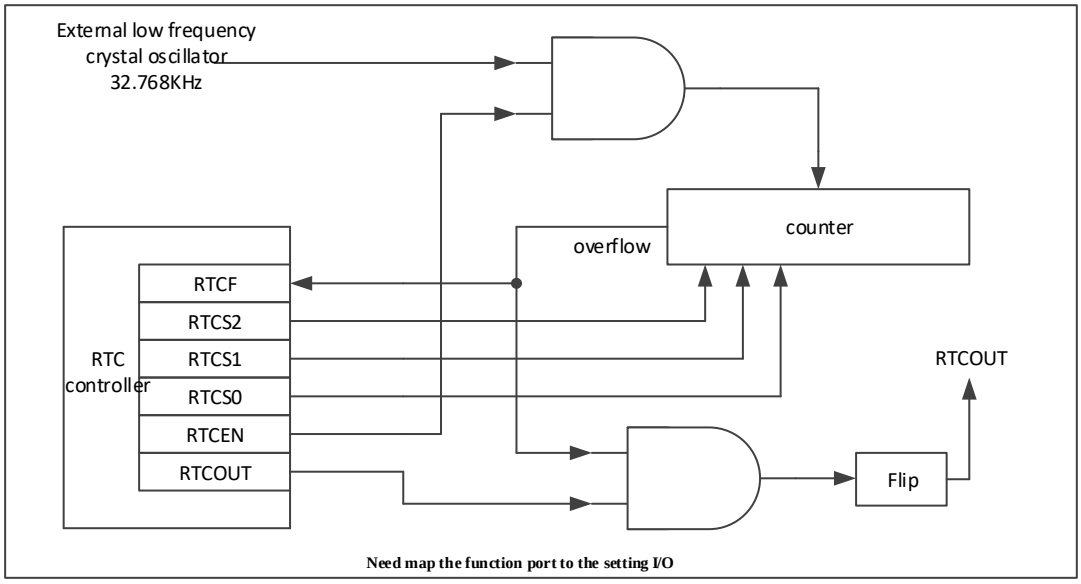
Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-						EPCHn	EPCLn

Bit	Flag	Introductions
7-2	-	Reserved (read = 0b, write invalid)
1	EPCHn	In PWM mode, it is composed of 9 digits with CCAPHn.
0	EPCLn	In PWM mode, it is composed of 9 digits with CCAPLn.

12 Real-time clock RTC

12.1 RTC Characteristics

This series of MCU has a BUILT-IN RTC module. In order to make the RTC module work, external low-frequency crystal clock 32.768khz must be enabled; otherwise, the RTC is invalid. External low-frequency crystal oscillator needs to be enabled in two steps. When RTC count overflows, the RTC interruption will wake up the power mode. However, after awakening, the RTC must operate for 30 s before entering the next power down mode. If RTC is not required, RTCEN 0 is required to operate in the power down mode. The RTC's control registers can be set to generate fixed periodic interrupts and continuous fixed clock output.



12.2 RTC Characteristics

12.2.1 RTC control register RTCC

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	RTCF			RTCS[2:0]			RTCEN	RTCOUT

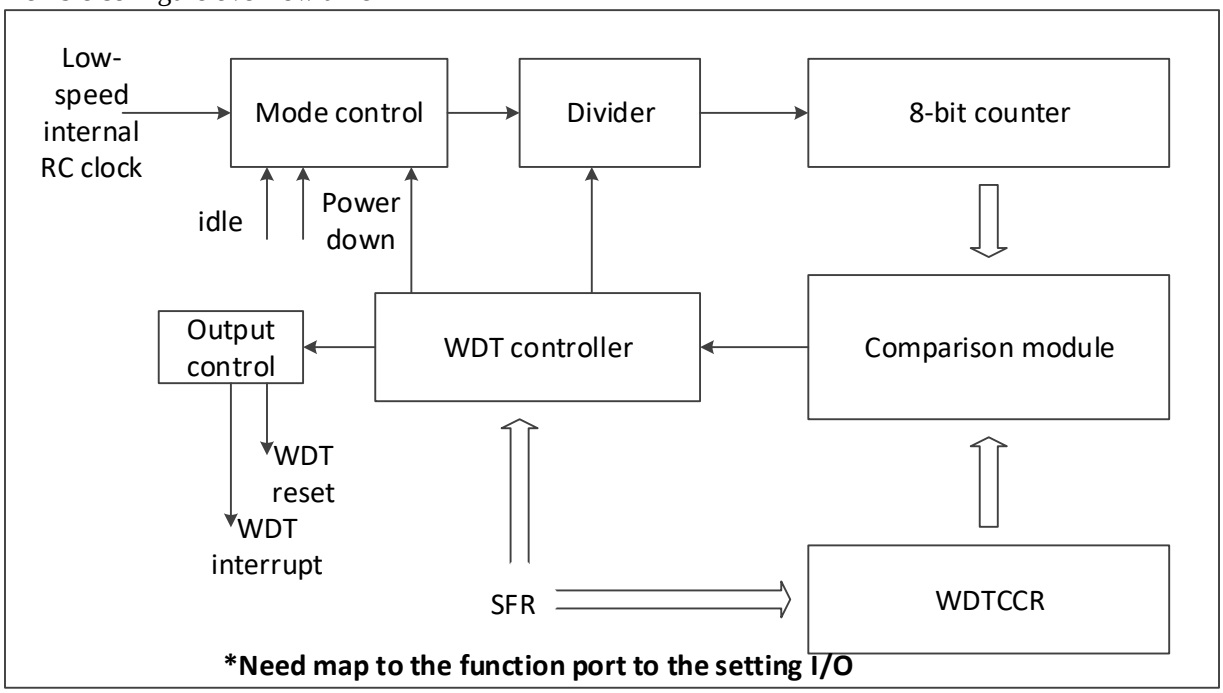
Bit	Flag	Introductions
7	RTCF	RTC interrupt flag bit 0：Software clear 0 1：Hardware set 1 when RTC counter overflows
6-5	-	Reserved (read = 0b, write invalid)
4-2	RTCS[2:0]	RTC break time selector bit 000：0.0625s 001：0.125s 010：0.25s 011：0.5s

		100：1.0s 101：2.0s 110：4.0s 111：8.0s
1	RTCEN	RTCEN Operation control bit 0：Stop RTC work 1：Start RTC work (re-counting)
0	RTCOUT	RTC clock output enable bit 0：Disable RTC clock output 1：Allows RTC clock output (this port should be set to output mode beforehand)

13 Watchdog timer WDT

13.1 WDT characteristics

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



HC89S105xx 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.

HC89S105xx 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.

13.2 WDT registers

13.2.1 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, you need to enable WDTRST. That is, only a watchdog reset can awaken the electrical mode. 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

13.2.2 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.

Overrun time = (WDT frequency division coefficient * (WDTCCR [7:0] + 1))/ 38K
WDTCCR[7:0] = 0xFF Watchdog overflow time table as below.

PS2	PS1	PS0	WDT Frequency division coefficient	WDT Maximum overflow time@38K
0	0	0	8	54 ms
0	0	1	16	108 ms
0	1	0	32	216 ms
0	1	1	64	432 ms
1	0	0	128	864 ms
1	0	1	256	1728 ms
1	1	0	512	3456 ms
1	1	1	1024	6912 ms

14 Universal asynchronous transceiver UART

14.1 UART characteristics

- 2 UART
- Multiple work modes
- Multiple errors detection

14.2 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 $F_{CPU}/12\times6^{UX6}$
0	1	1	Asynchronous	Timer 4 overflow rate /16
1	0	2	Asynchronous	$(2^{SMOD}/64)\times F_{CPU}$
1	1	3	Asynchronous	Timer 4 overflow rate /16

14.2.1 Mode0 : Synchronous half-duplex communication

Mode0 support synchronous communication of external devices, RXD pin send and receive serial data, TXD Pin send shift clock. HC89S105xx 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 $1/12*F_{osc}$ or $1/2*F_{osc}$. 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 HC89S105xx 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.

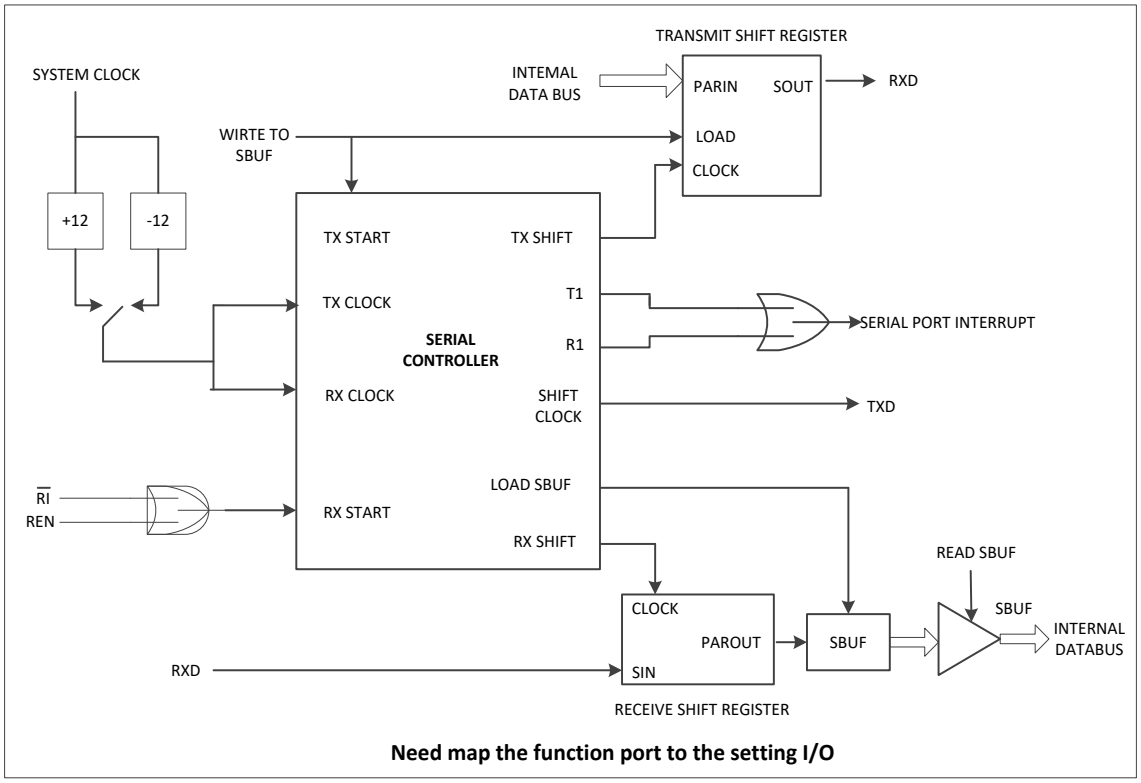


Figure 14- 1 UART mode0 function block diagram

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.

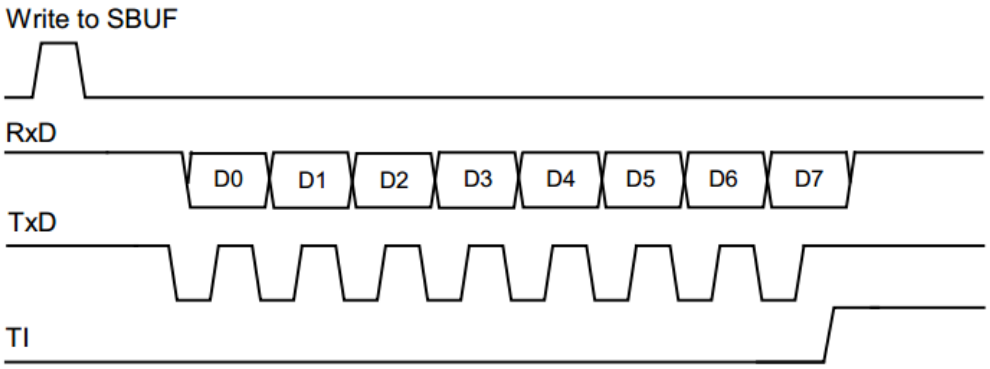


Figure 14- 2 Mode0 data send timing diagram

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.

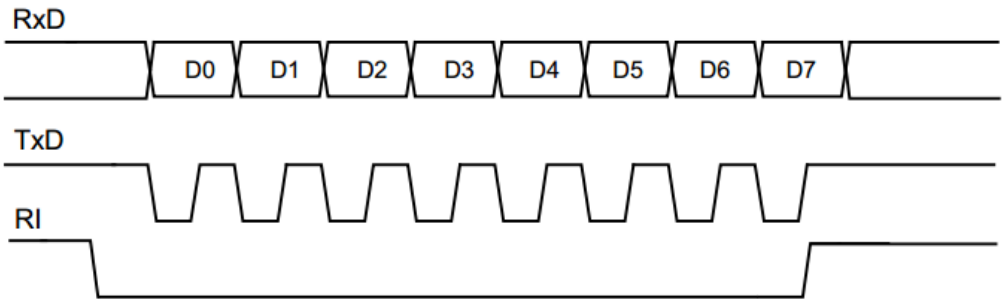


Figure 14 - 3 Mode0 Data receive timing diagram

14.2.2 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:

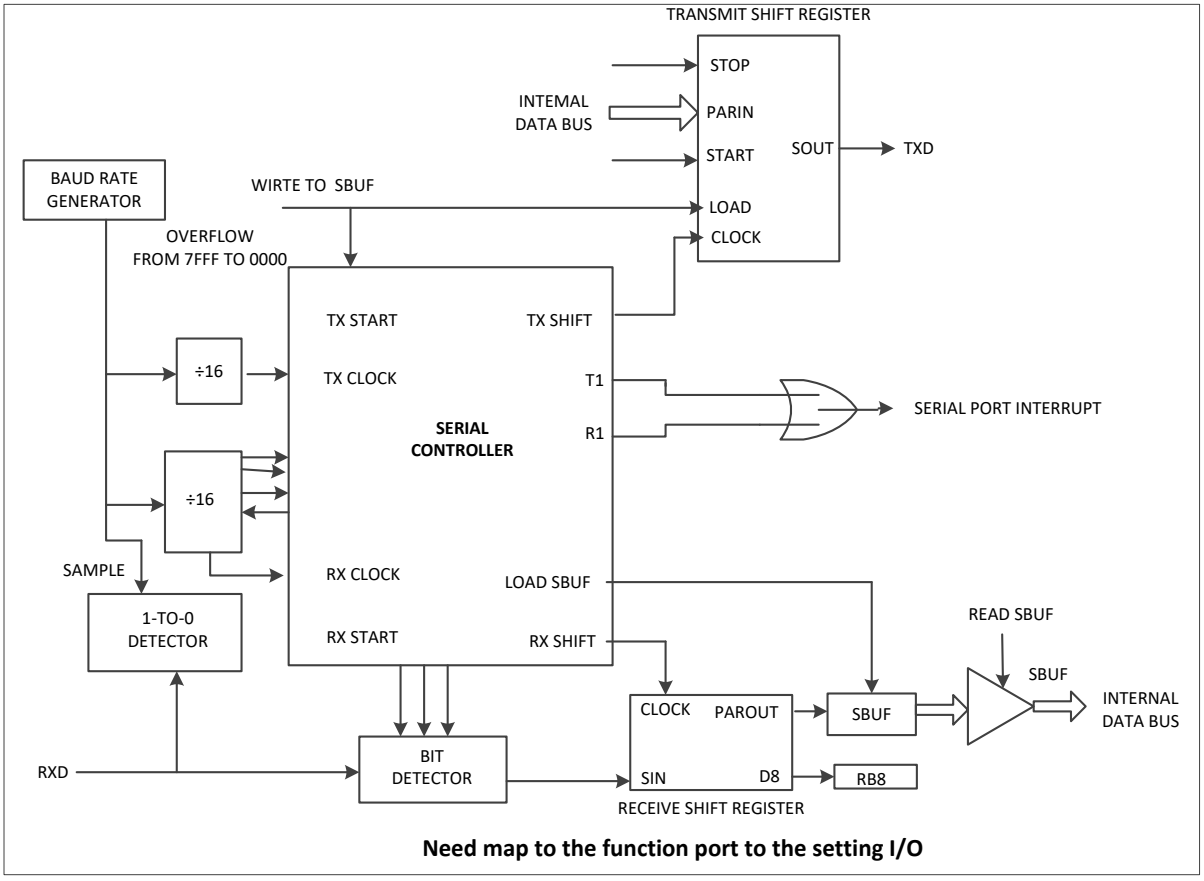


Figure 14 - 4 UART mode1 functional block diagram

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.

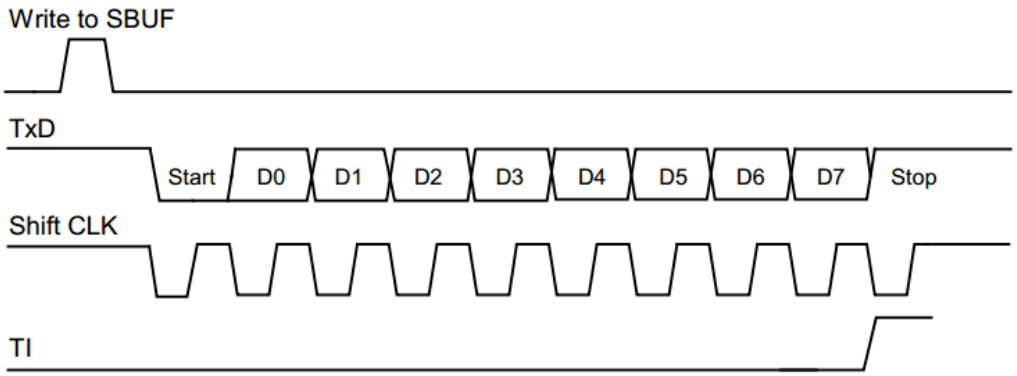


Figure 14 - 5 mode1 send the data time series block diagram

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.

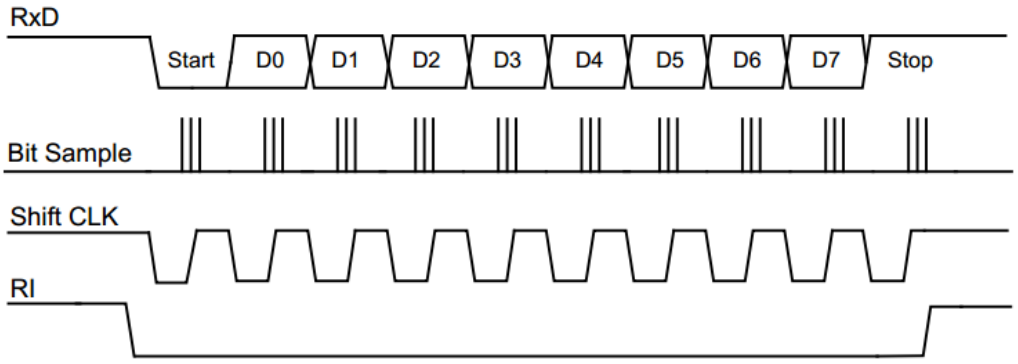


Figure 14 - 6 mode1 data receive timing diagram

14.2.3 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.

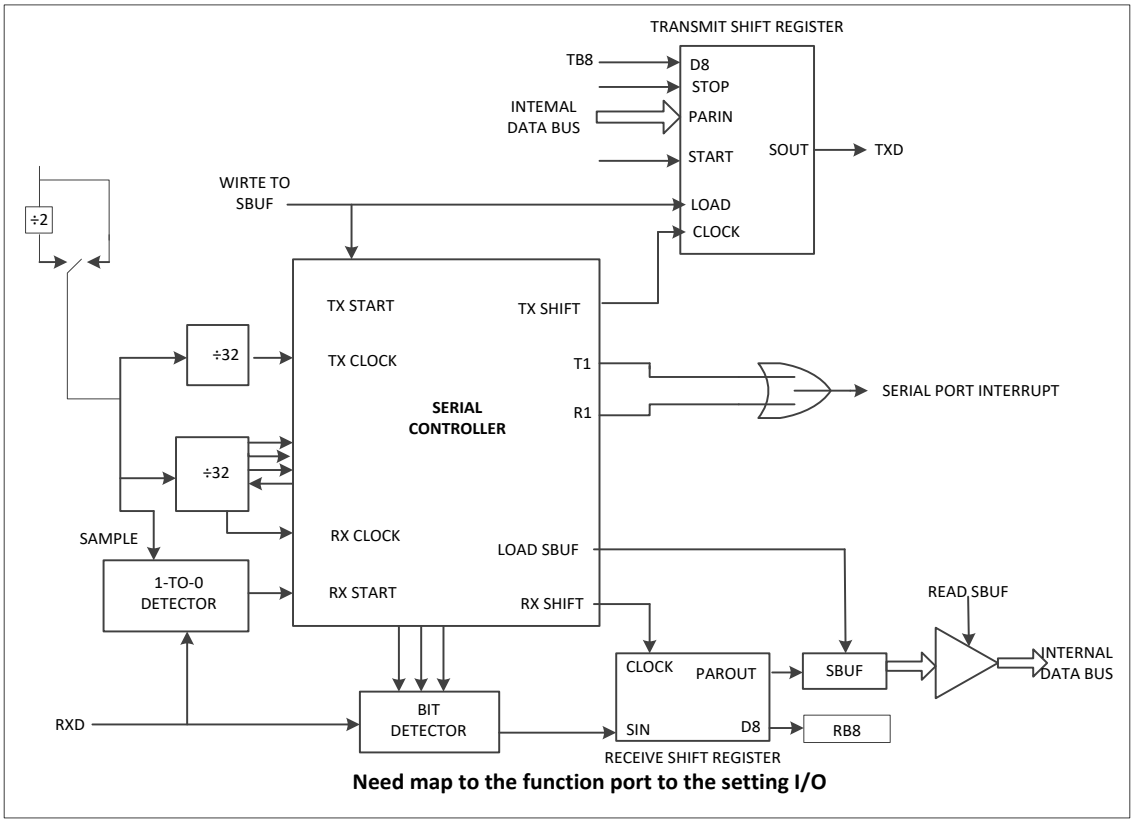


Figure 14 - 7 UART mode2 functional block diagram

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.

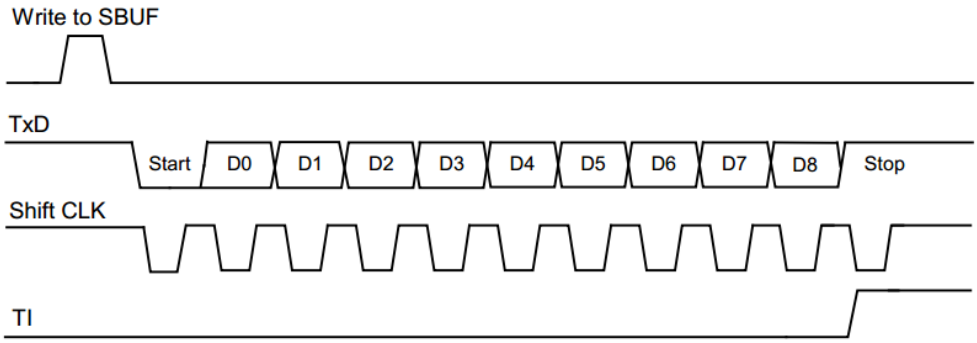


Figure 14 - 8 Mode2 Send the data time series block diagram

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.

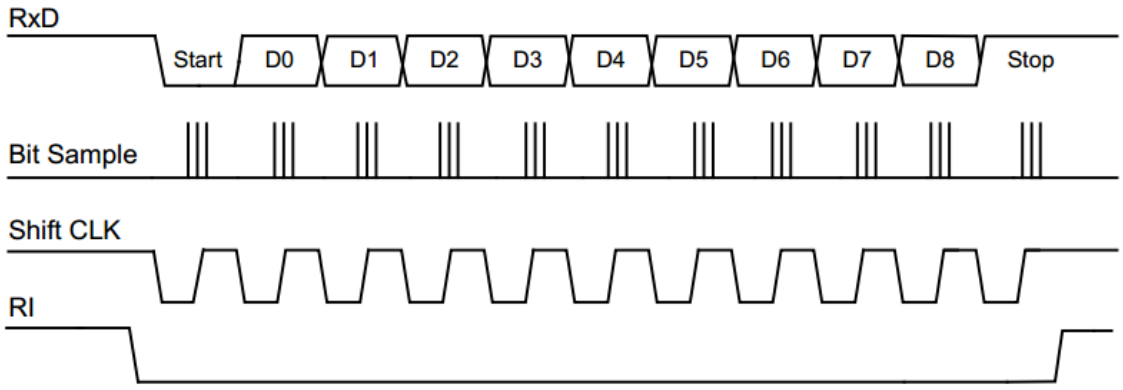


Figure 14 - 9 Mode2 receive data timing diagram

14.2.4Mode3 : 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.

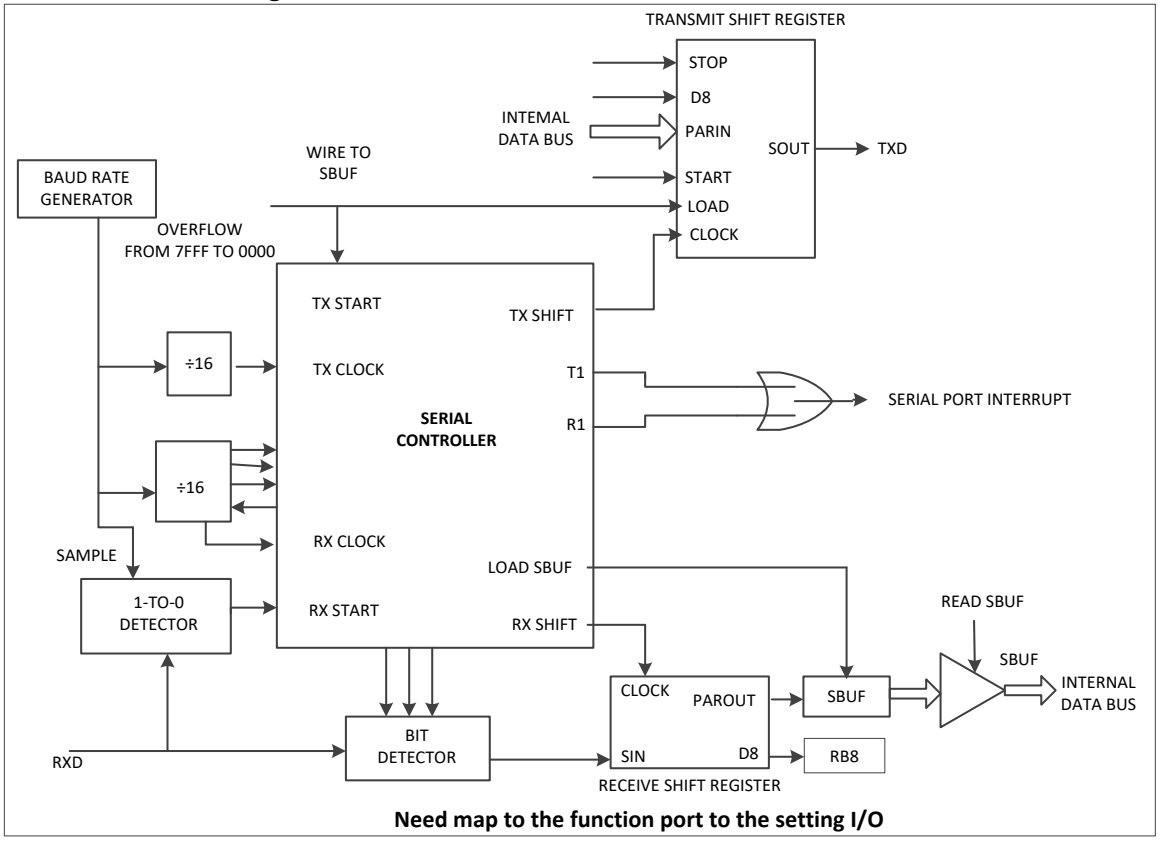


Figure 14 - 10 UART mode3 function block diagram

14.3 Baud rate generator

UART comes with a baud rate generator, which is essentially a 16-bit incrementing counter.

In Mode 0, the baud rate can be programmed to be 1/12 or 1/2 of the system clock, determined by the UX6 bits. When UX6 is 0, the serial port runs at 1/12 of the FCPU. When UX6 is 1, the serial port runs at 1/2 of the FCPU.

In Mode 2, the baud rate is fixed at 1/32 or 1/64 of the system clock, depending on the SMOD bit. When the SMOD bit is 0, UART runs at 1/64 of the FCPU. When the SMOD bit is 1, UART runs at 1/32 of FCPU.

$$\text{BaudRate} = 2^{\text{SMOD}} \times \left(\frac{F_{CPU}}{64}\right)$$

In Mode 1 and mode 3, baud rate formula is as follows:

$$\text{Baud} = \frac{F_{CPU}}{16 \times (65536 - \text{SBRT})}, \quad \text{SBRT} = [\text{SBRTH} : \text{SBRTL}]$$

Following table is common CPU frequency and baud rate corresponding to the baud rate generator overloaded value

Commonly used baud rate	Fcpu		
	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.

14.4 Multiprocessor communication

14.4.1 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.

14.4.2 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.

14.5 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.

14.5.1 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).

14.5.2 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.

14.6 UART1 registers

14.6.1 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	-	RXROV	TXCOL	REN	TB8	RB8	TI	RI

Bit	Flag	Introductions
7	-	Reserved
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	BRTR	BRTOUT	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_{cpu} /12 1 : Serial port mode0 Clock = F_{cpu} /2
4	BRTR	Independent baud rate generator BRT operation control bit 0: independent baud rate generator BRT stop working 1: start the baud rate generator BRT work independently
3	BRTOUT	Independent baud rate generator BRT output enable bit

		0: independent baud rate generator BRT clock output 1: allow independent baud rate generator BRT clock output
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_{cpu} /12 When UX6 = 1, the baud rate is F_{cpu} /2
0	1	1	8 bit UART, variable baud rate	Overflow rate of BRT independent Baud rate generator /16
1	0	2	9 bit UART	$(2^{SMOD}/64)\times F_{cpu}/2$
1	1	3	9 bit UART, variable baud rate	Overflow rate of BRT independent Baud rate generator /16

14.6.2 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

14.6.3 UART1 independent Baud rate generator registers SBRTL, SBRTH

SBRTL

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	SBRTL[7:0]							

Bit	Flag	Introductions
7-0	SBRTL[7:0]	Baud rate generator register BRT low 8 bit, used to hold the reload time constant

SBRTH

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	SBRTH[7:0]							

Bit	Flag	Introductions
7-0	SBRTH[7:0]	Baud rate generator register BRT high 8 bit, used to hold the reload time constant

Note: Modify SBRTL & SBRTH, similar to t0/T1 mode 0 when overloaded data is modified.

14.6.4 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

14.7 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 Baud rate clock cannot output;

14.7.1 UART2

14.7.1.1 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.

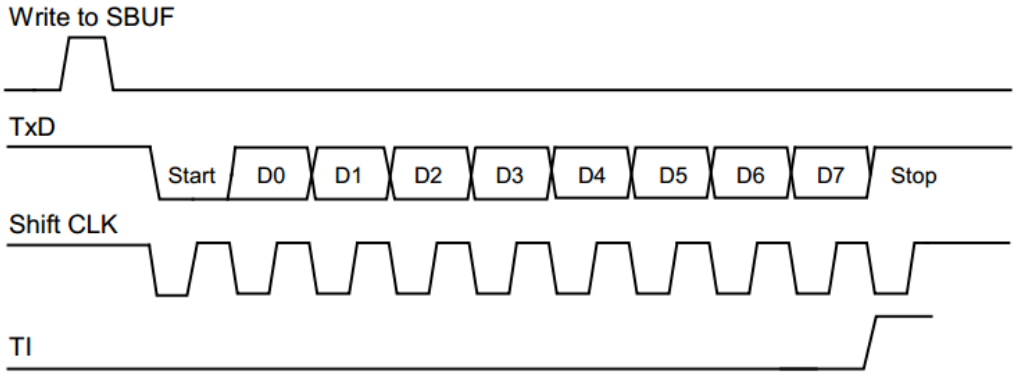


Figure 14 - 11 Send timing of mode0

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.

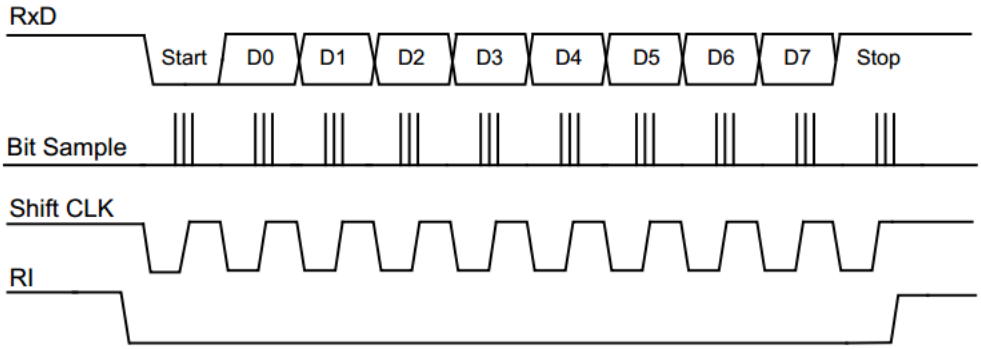


Figure 14 - 12 Receive timing of mode0

14.7.1.2 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.

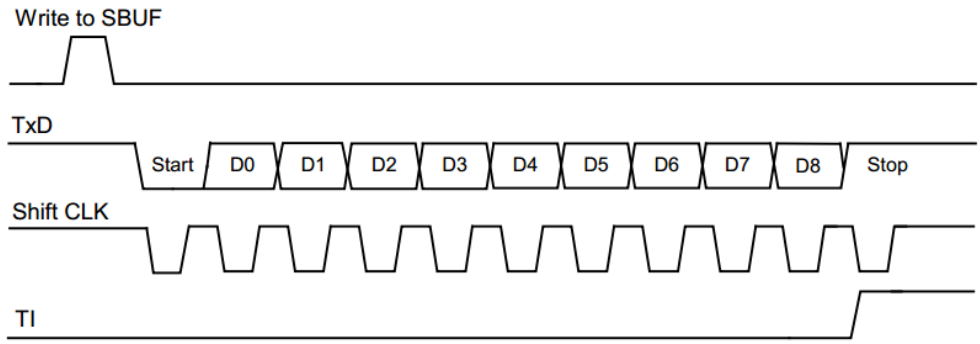


Figure 14 - 13 Send Timing of mode1

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.

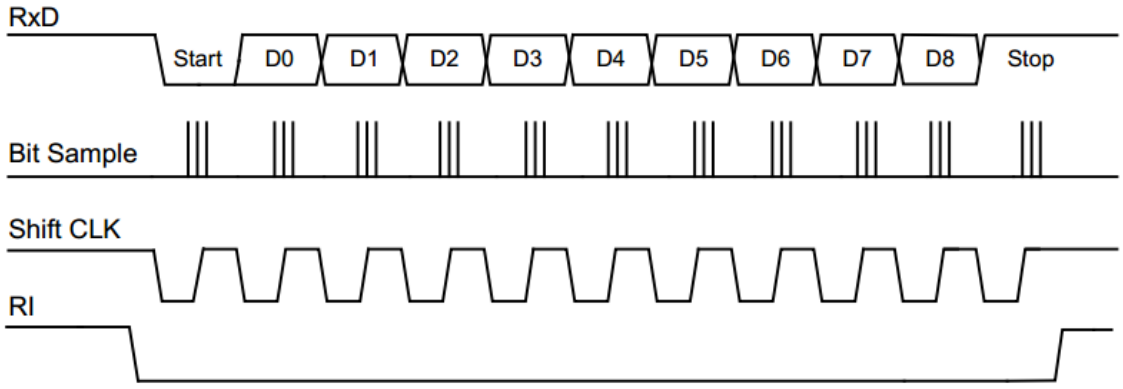


Figure 14 - 14 Receive timing of mode1

14.7.2 UART2 control register S2CON, S2CON2

S2CON

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	BRTR	-	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_{cpu} /12 1 : Serial port mode0 Clock = F_{cpu} /2
4	BRTR	Independent baud rate generator BRT operation control bit 0: independent baud rate generator BRT stop working 1: start the baud rate generator BRT work independently
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_{cpu} /12 When UX6 = 1, the baud rate is F_{cpu} /2
0	1	1	8 bit UART, variable baud rate	Overflow rate of BRT independent Baud rate generator /16
1	0	2	9 bit UART	$(2^{SMOD}/64)\times F_{cpu}/2$
1	1	3	9 bit UART, variable baud rate	Overflow rate of BRT independent Baud rate generator /16

14.7.3 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

14.7.4 UART1 independent Baud rate generator registers S2BRTL, S2BRTH

S2BRTL

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	S2BRTL[7:0]							

Bit	Flag	Introductions
7-0	S2BRTL[7:0]	Baud rate generator register BRT low 8 bit, used to hold the reload time constant

S2BRTH

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	S2BRTH[7:0]							

Bit	Flag	Introductions
7-0	S2BRTH[7:0]	Baud rate generator register BRT high 8 bit, used to hold the reload time constant

Note: To modify S2BRTL & S2BRTH, the higher level S2BRTH should be modified first, and then the lower level S2BRTL

14.7.5 UART2 automatic address recognition S2ADDR, S2ADEN

Slave address register S2ADDR

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	S2ADDR[7:0]							

Bit	Flag	Introductions
7-0	S2ADDR[7:0]	Slave address register

Slave address mask register S2ADEN

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	S2ADEN [7:0]							

Bit	Flag	Introductions
7-0	S2ADEN [7:0]	slave address mask register

15 Serial peripheral interface SPI

15.1 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

15.2 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.

15.3 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.

15.4 SPI functional block diagram

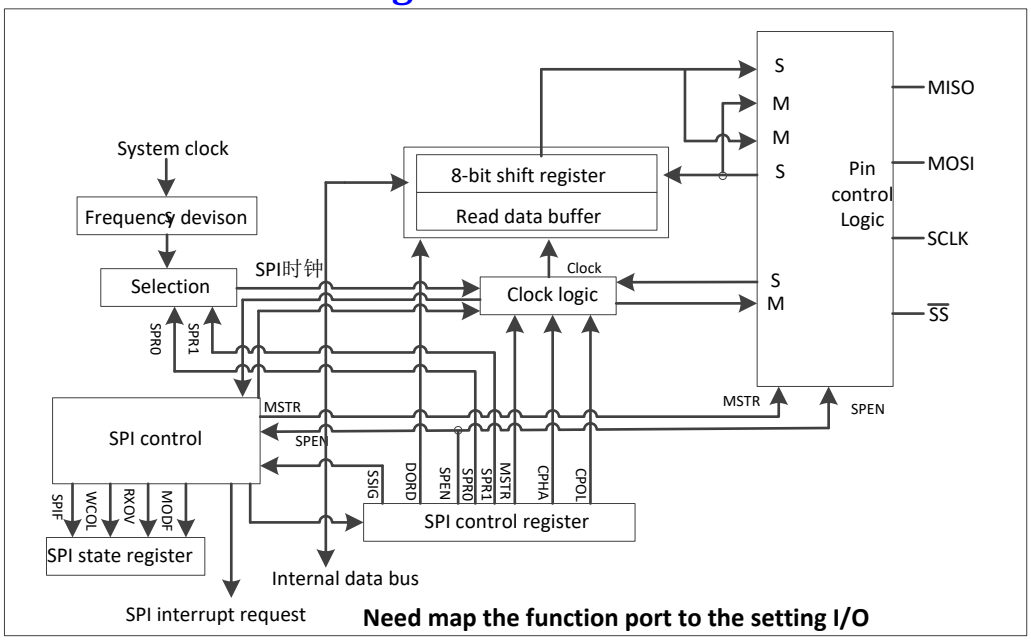


Figure 14 - 1 SPI functional block diagram

15.5 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.

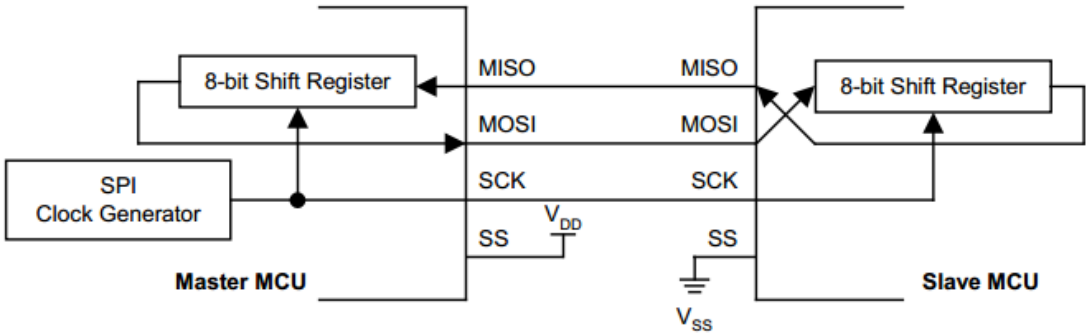


Figure 14 - 2 Full-duplex master/slave interconnect diagram

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.

15.6 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.

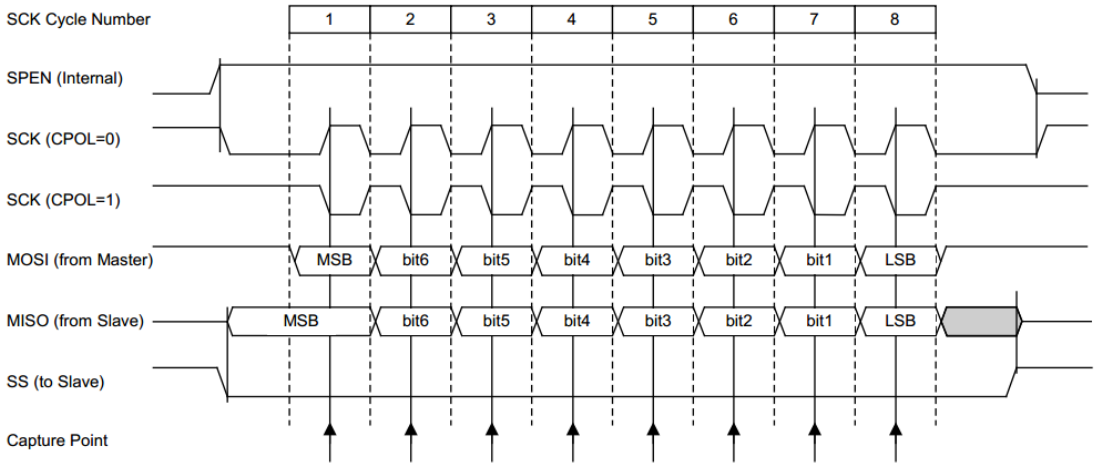


Figure 14 - 3 Data transfer form (CPHA=0)

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.

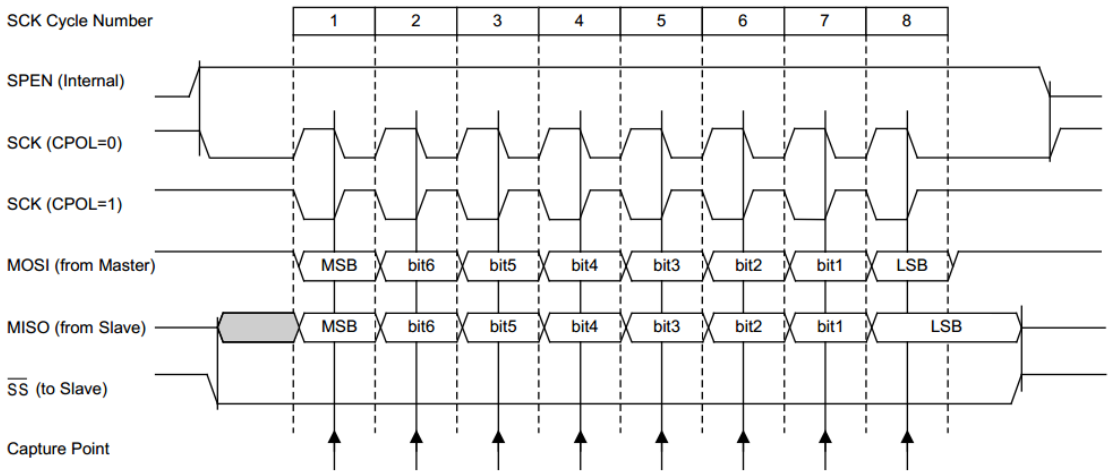


Figure 14 - 4 Send form data (CPHA=1)

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.

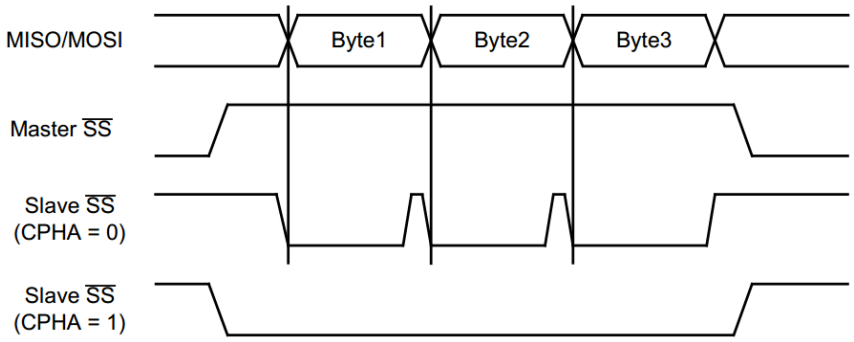


Figure 14 - 5 CPHA/($\overline{SS}$) timing sequence

15.7 SPI Error detection

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.

15.8 SPI interrupt

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.

Model conflict: the main equipment of the SS pin is enabled, if the SS is lower, the battle for the bus at this time will happen. The SPIF flag bit of SPSTAT will be set (interruptible), and the MSTR bit will be cleared to 0, forcing the device to switch to a slave device. Therefore, the user software must always detect the MSTR bit. If it is cleared by a slave selection and the user wants to continue using SPI as the host, the MSTR bit must be reset, or enter slave mode.

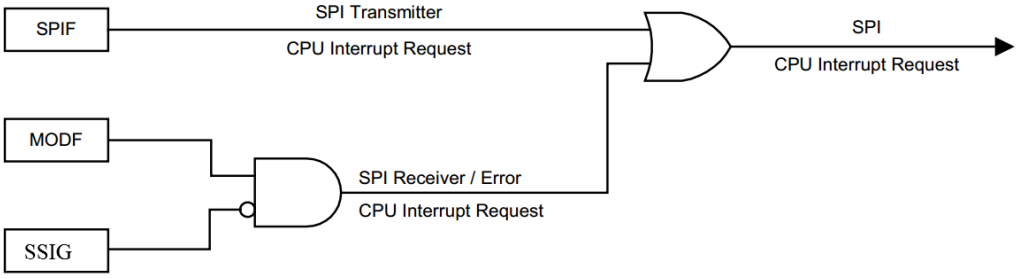


Figure 14 - 6 SPI interrupt request generation

15.9 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	-

15.10 SPI registers

15.10.1 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

15.10.2 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	-					

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

		being transmitted is not affected)
5-0	-	Reserved (read = 0b, write invalid)

15.10.3 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

16 IIC bus

16.1 IIC characteristics

- Double line communication
- Support master mode and slave mode
- Support multi-master communication with clock arbitration function
- Support address programmable
- Supportt standard data rate (up to 100kbps) and fastest data rate (up to 400kpbs)

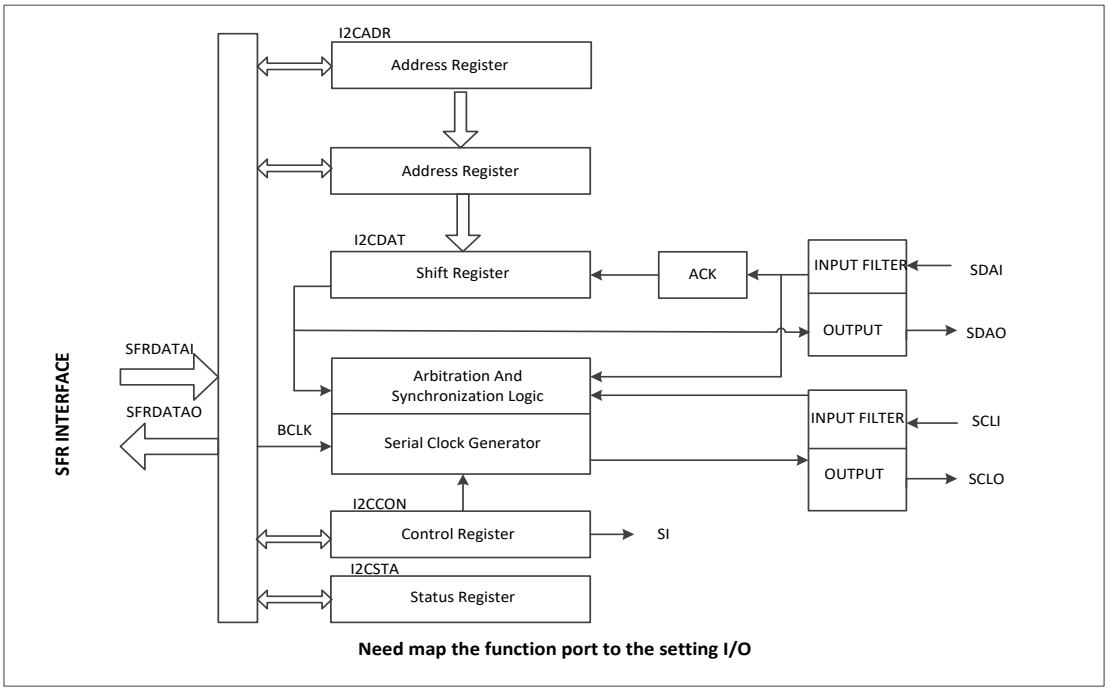


Figure 16-1 IIC function diagram

16.2 IIC bus work principle

In physical architecture, IIC system is consist of a serial data line SDA and clck line SCL. Master transmmit information as regular communication protocal, during data transmission, the initialization is cpmpleted by master. Master transmit data via SDA, meanwhile transmit clock via SCL. Transmission target and direction, start and end of transmission are all determined by master.

Every device has a unique address, and it could be single receiver or transceiver device. Transmitter or receiver could be operated in master or slave mode. It is determined whether or not the chip must be start up data transmission or be addressed only.

Below is general, typical IIC bus connection mode.

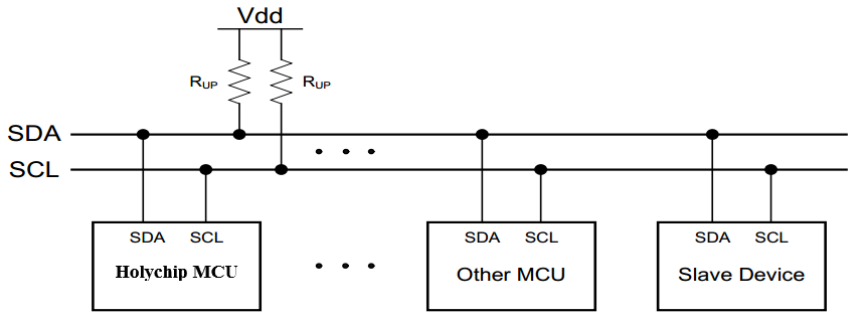


Figure 16-2 IIC bus connection diagram

16.3 Bus data availability

IIC bus transmits data by serial. High bit of byte is tranmitted first, each bit has a corresponding clock edgeon SCL. A stable logic level must be maintained on data line during clock high level, high level is data 1, low level is data 0, the level of data line is permitted to change only during clock is low. As figure16-3 below:

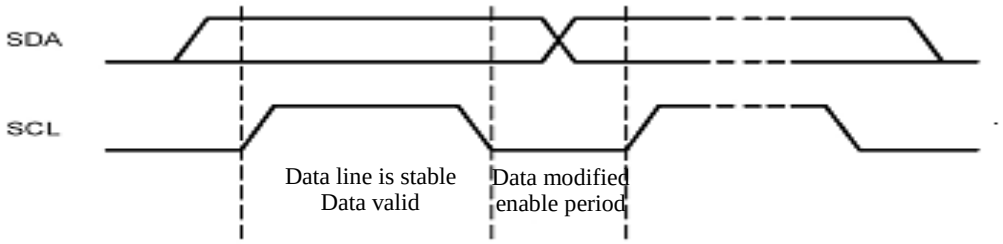


Figure 16-3 IIC bus data availability

16.4 Bus signal

IIC bus data transmission includes 4 types signal, they are: start signal, stop signal, restart signal, acknowledge signal.

Start signal (START): As Figure 16-4 shown, when SCL is high level, SDA transition from high level to low, it is start signal. When bus is idle, for example, no device is using the bus (SDA and SCL are high), master send start signal to establish communication.

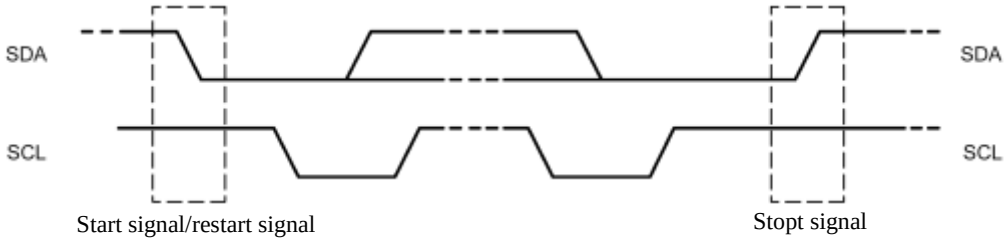


Figure 16-4 Start, restart, stop signal

Stop signal(STOP): as figure 16-4 shown, when SCL is high level, SDA transition from low level to

high, it is stop signal. Master end the data communication by sending a stop signal.

Restart signal (repeated START): on IIC bus, master send a start signal to start-up one time communication, before first time sending stop signal, by sending a repeated start, master can change the communication mode with current slave or switch to communicate with other slaves. As figure 16-5 shown, when SCL is high, SDA transition from high to low, a repeated start signal is generated, it is a start signal essentially.

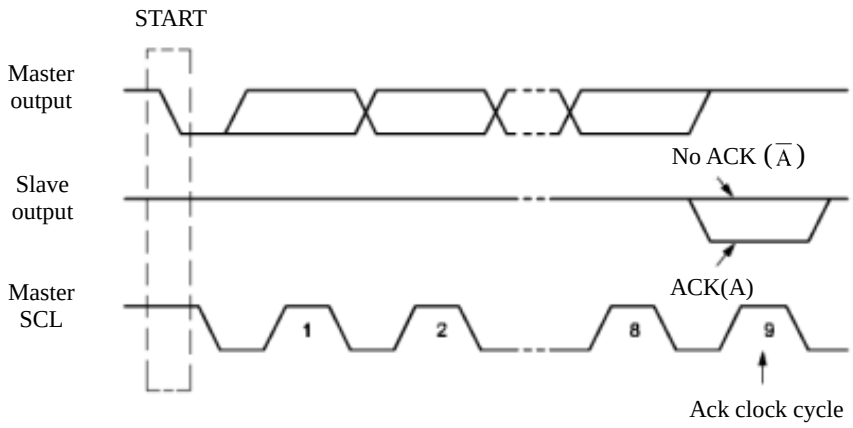


Figure 16-5 IIC bus acknowledge signal

Acknowledge signal (A): after slave received 8 bit data, it will send to master a special low level. Every byte must be followed by a acknowledge bit to indicates data has been received. Acknowledge appeared at the 9th clock cycle, the master must release data line at this time, and slave pull down the SDA line to generate acknowledge signal, or slave maintain the SDA line as high to generate a no acknowledge($\overline{A}$), as Figure16-5 shown. So one byte transmission needs 9 clock cycle. If slave as receiver send no acknowledge signal to master, the slave will end the transmission, and release SDA line. Any above cases will end the data transmission. At this time master sends stop signal to release bus, or generates repeated start signal to restart a new transmission. Start signal, repeated start and stop signal are all controlled by master, acknowledge signal is generated by receiver.

16.5 Bus data Initialization format

In general, a standard IIC communication is consist of 4 parts：start signal, slave address transmission, data transmission, stop signal.

Master send a start signal to start up one time IIC communication; after master address slave, then transmit data on bus. Every data is 8 bits, high bit sent first, and every byte must be followed by a acknowledge bit. The lengths of data are not limited; after end of all data transmission, master send a stop signal to end the communication.

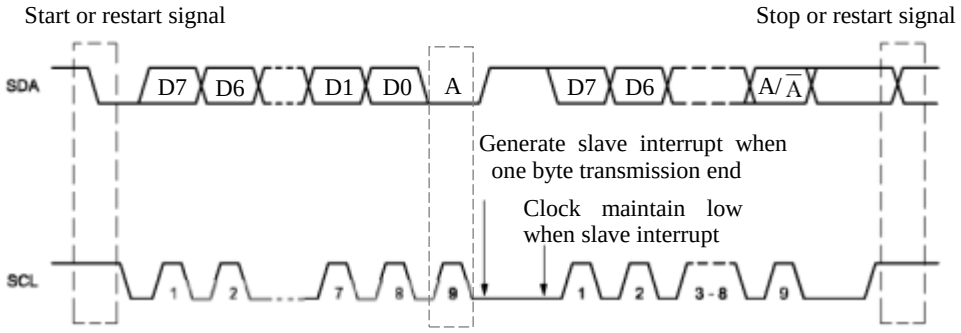


Figure 16-6 IIC Bus data transmission format

As Figure 16-6 shown, data transmission will be stopped when clock is low. After one byte received, this case can be used to the receiver need some other operation but cannot receive next data immediately, and force bus to idle status, until the receiver ready to receive new data, it will release clock signal to enable transmission again. For example, when receiver got one byte from master, system will generate and enter interrupt processing, after the interrupt next byte can be received, and in interrupt procedure the receiver will maintain low level of SCL until the interrupt end.

16.6 IIC bus addressing appointment

Slave device on IIC bus has a special 7 bits slave address usually, it has up to 128 coded space when use 7 bits slave address, so based on original 7 bits address, 10 bits address code format. It is match IIC bus protocol too.

“Broadcast call” is an exception, it can address all devices by writing 0 to the first byte. Broadcast call is used to the case that master need send the same information to several slaves. When the address is using, the other devices will respond or ignore as software configuration. If device responds broadcast call, the operation is same as slave receive mode.

16.7 Process of master write one byte to slave

As Figure 16-7 shown, when master send one byte to slave, first master send a start signal, and a slave address followed, the address has 7bits, then the 8th bit followed is data direction bit(R/W), 0 indicates master send data (write), 1 indicates master receive data (read), this time master wait slave give a acknowledge(A), when master received an acknowledge signal, send the address will be accessed, and wait slave give an acknowledge again, then master will send one byte data after received an acknowledge, and continuous to wait slave give an acknowledge, when master received the acknowledge, it will generate a stop signal, and end the transmission.

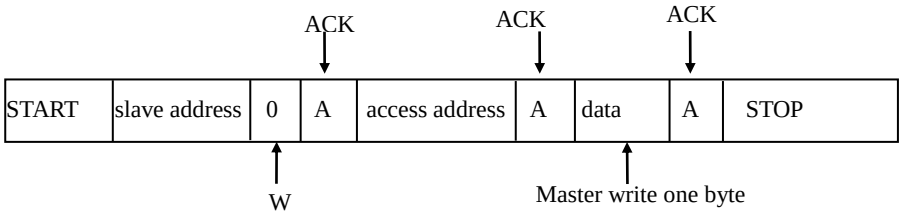


Figure 16-7 Master write one byte data to slave

As Figure 16-7 shown, master read one byte data from slave, first master sends a START signal, then follows a slave address, the 8th bit of the address is 0, it is indicates a write command to slave, the master wait slave give an acknowledge(A) at this time, when master received an acknowledge signal, send the address will be accessed, and wait slave give an acknowledge again, then master will change the communication mode(master changed from transmitter to receiver, slave changed from receiver to transmitter) after received an acknowledge. So master send a restart signal, then follows a slave address, the 8th bit is 1, it is indicates master has been set receive mode and start to receive data, this time master wait an acknowledge from slave, when master received acknowledge signal, then it can receive one byte data, when receive is completed, master send a no acknowledge signal, it is indicates receive end, master generates a stop signal, and end the transmission.

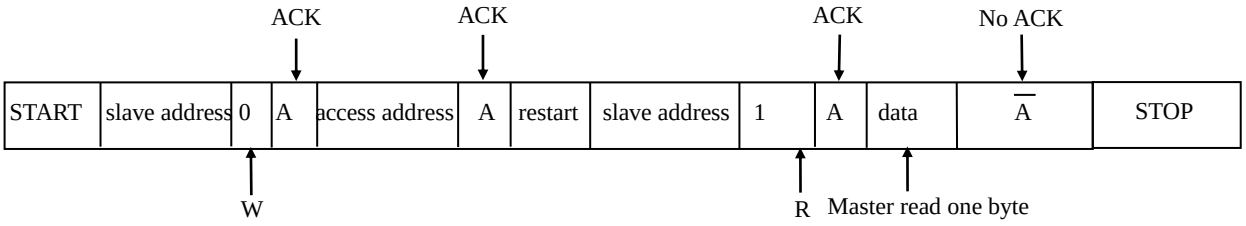


Figure 16-8 Master read one byte data from slave

16.8 IIC work mode

16.8.1 Master send mode

In master send mode, master send data to slave as next steps. Master write CR[2:0] to set expected clock rate and set IICEN bit to enable IIC bus, set STA bit to enter master send mode, as long as bus is idle, hardware will test bus and generates start signal, after the start signal is generated, SI bit will be set and status code of IICSTA is 08H, then load target address and data direction it ”write” (SLA + W) into IICDAT, SI bit must be clear to 0 when SLA +W start to transmit.

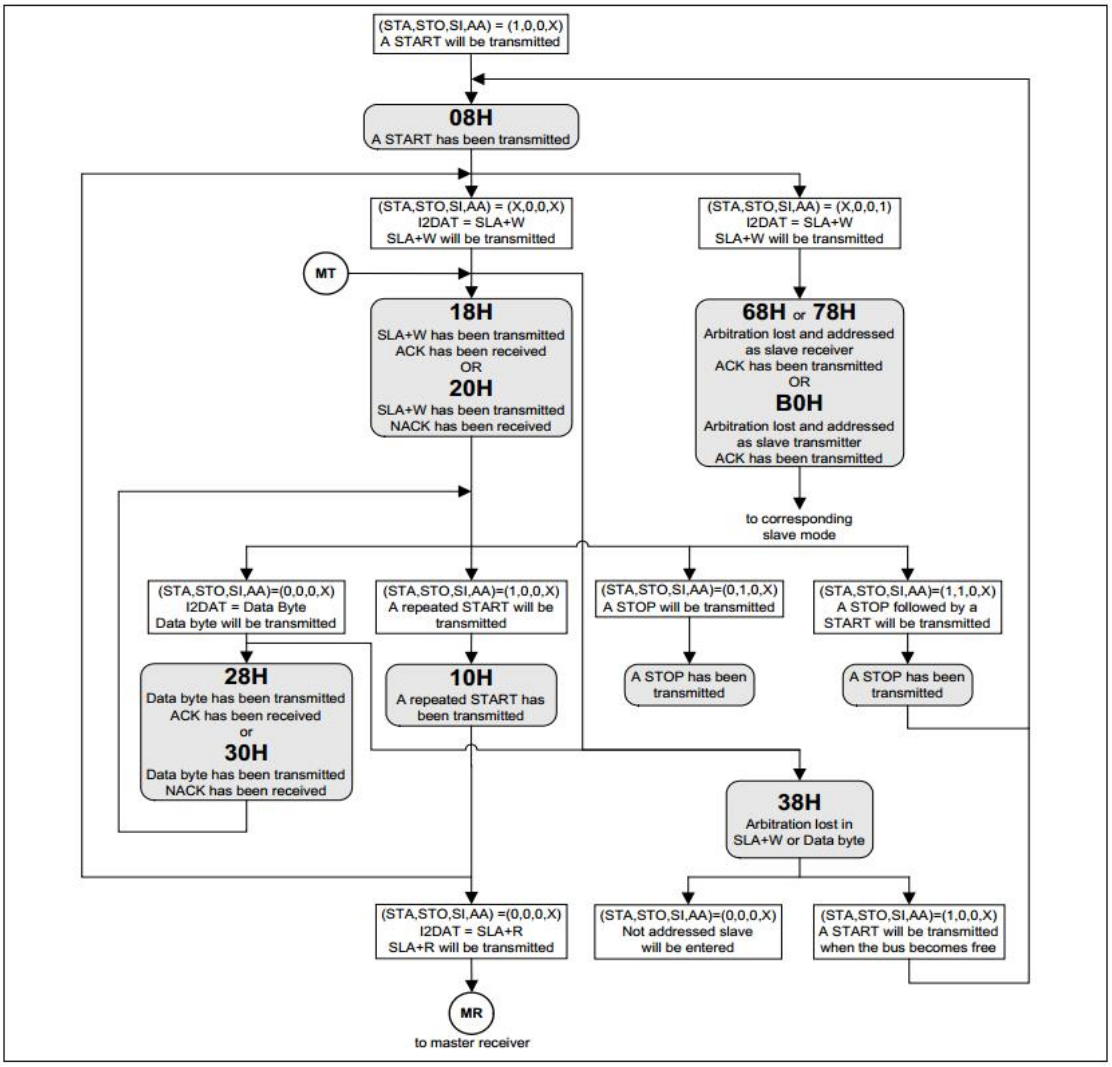


Figure 16-9 Master send mode flow and status

16.8.2 Master receive mode

In master receive mode, master receive data from slave as next steps. Start of transmission is same as master send mode, the target address and data direction it ”read” (SLA + R) will be loaded into IICDAT, after SLA + R byte is sent, and return an acknowledge, reset SI bit and read IICSTA as 40H, SI bit must be cleared to 0 to receive data from slave, if AA is set, master receiver will respond slave transmitter, if AA is cleared, master receiver will not respond slave, and release slave receiver as no-addressed slave, then master generates stop signal, repeat start signal to terminate transmission or start another transmission.

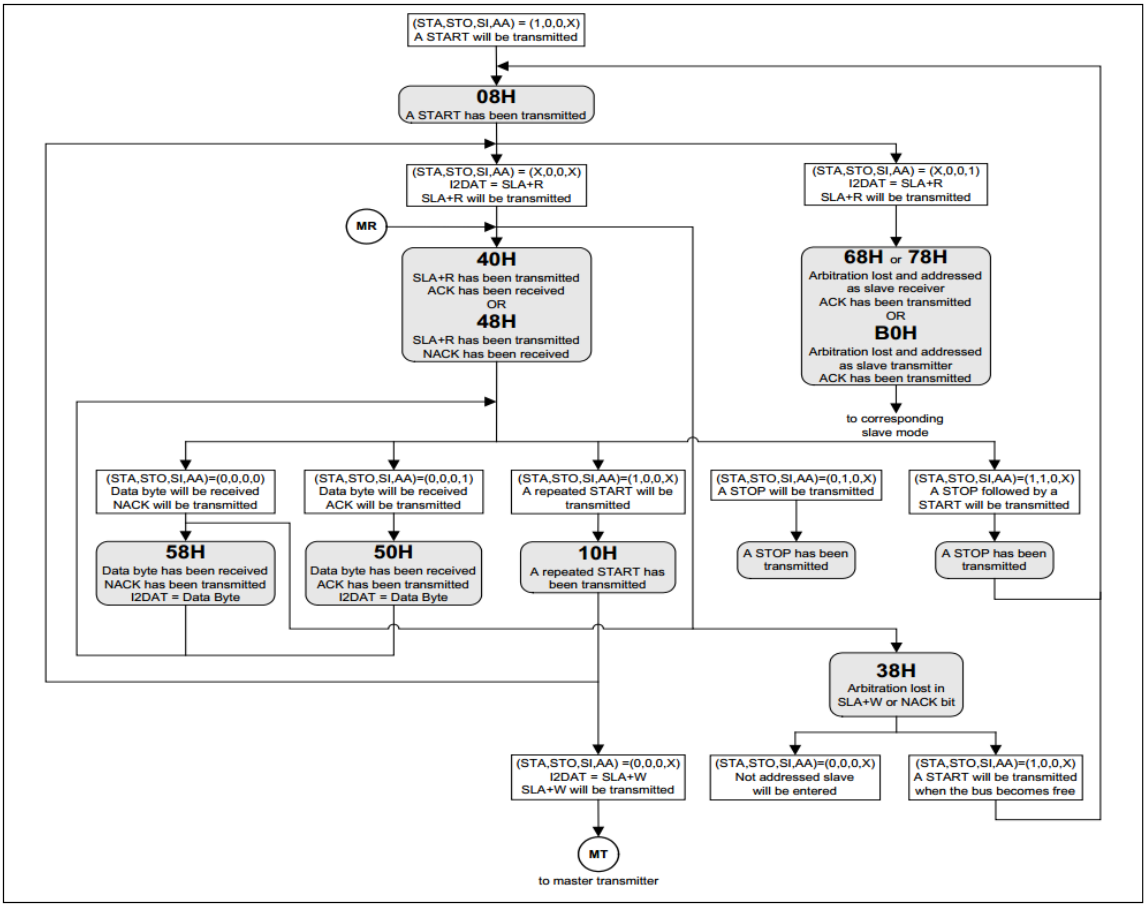


Figure 16-10 Master receive mode flow and status

16.8.3 Slave send mode

In slave send mode, slave send some data to master as next steps: after configure IICADR and IICCON register value, IIC wait itself address is addressed “read” (SLA + R). if arbitration fails, it can enter slave transmit mode.

After slave is addressed by SLA+W, user should clear SI flag to transmitt data to master transmitter, in general, master receiver return repoonse after slave send every byte, if the acknowledge is not been received, and if the transmission continuous it will send all “1”, it will become no-addressed slave, if AA flag is cleared during transmission, slave send the last byte, next time the transmission data are all “1”, slave is no-addressed.

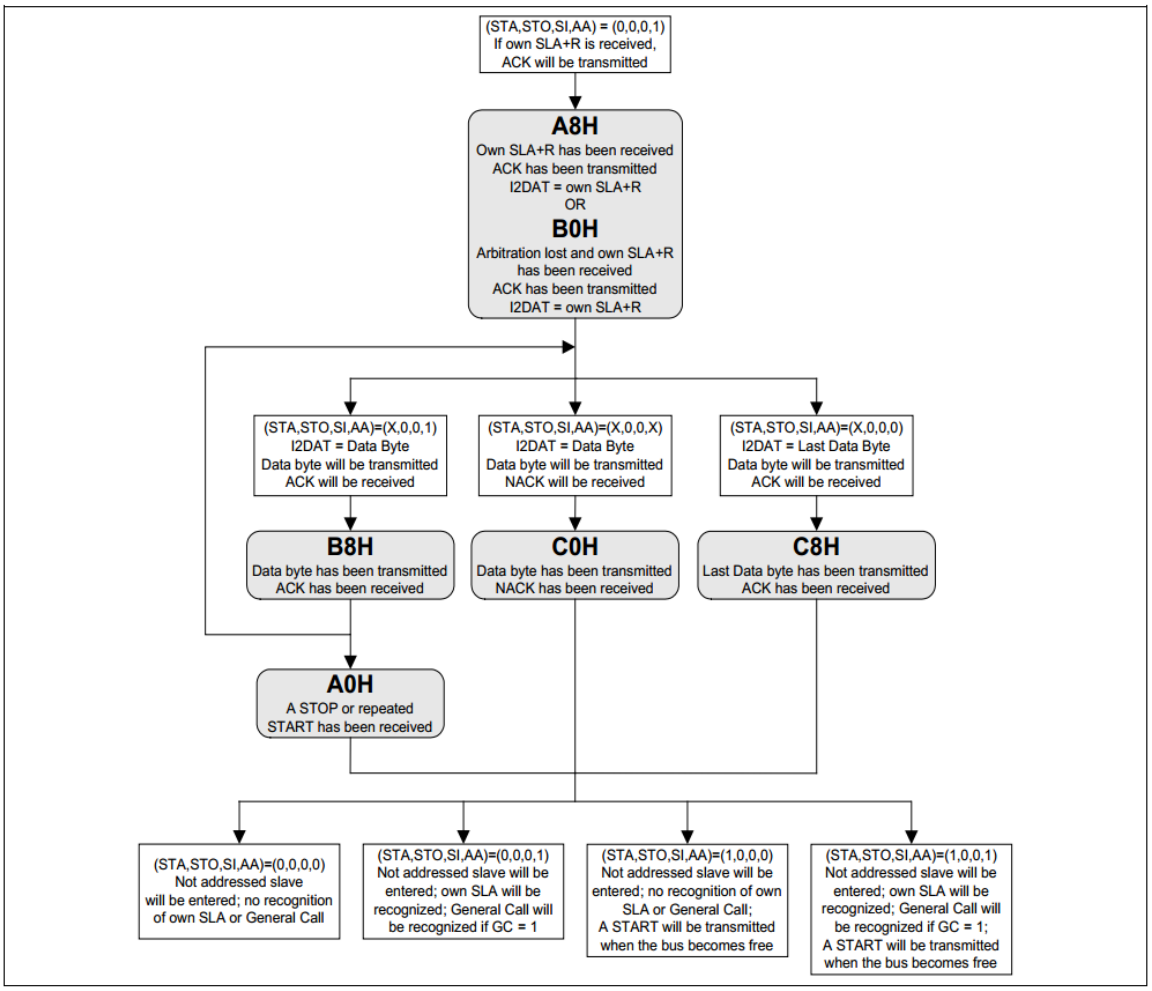


Figure 16-11 Slave send mode flow and status

16.8.4 Slave receive mode

In slave receive mode, slave receive some data from master as next steps: before starting, IICADR must be loaded reponse device address to addressed by master, AA bit must set to enable repond itself slave address or broadcast call, and after above initialization completed, IIC wait itself address is addressed and data direction bit “write” (SLA + W) or addressed by broadcast call. If arbitration fails, it can enter slave receive mode.

After slave is addressed by SLA + W, user should clear SI flag to receive data from master, During transmission, if AA flag equal 0, slave will return no-acknowledge after the data received next time, slave is no-addressed and separate from master, cannot receive any data in IICDAT, and maintain the current data received.

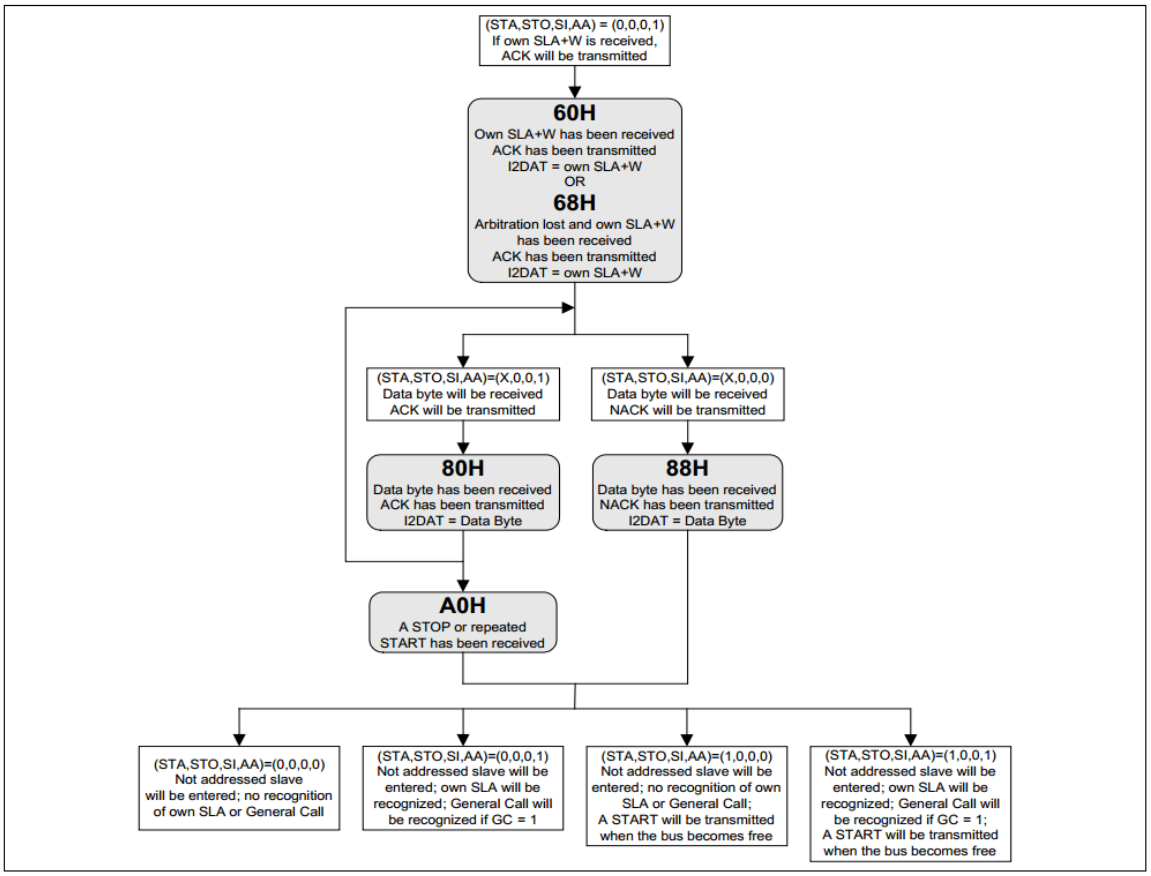


Figure 16-12 Slave receive mode flow and status

16.8.5 Broadcast call

Broadcast call is one of special slave receive modes, that is slave address and data direction bit are all 0, the slave is addressed by broadcast call has different status code in IICSTA register of normal slave receive mode, arbitration fails, it can generates broadcast call.

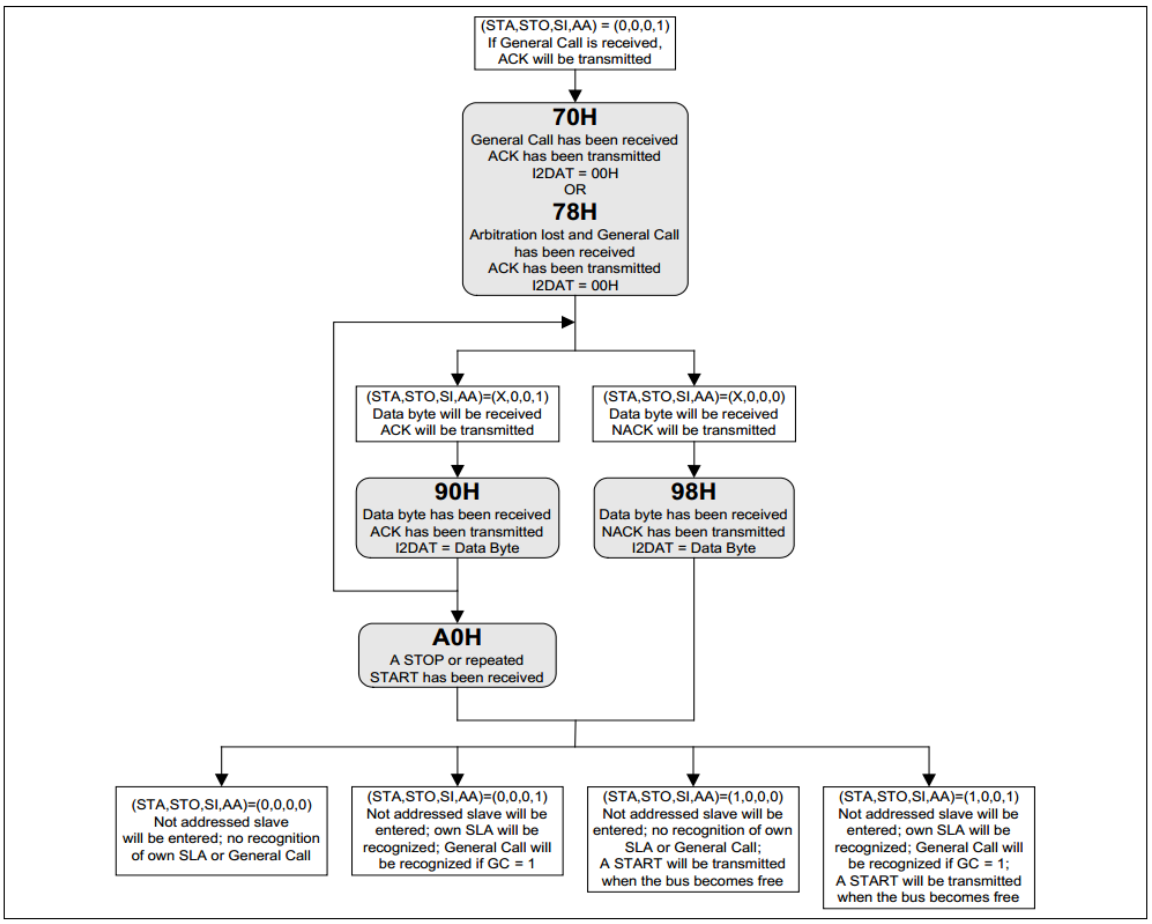


Figure 16-13 Broadcast call mode flow and status

16.8.6 Other status

There have 2 status code different with 24 defined status, that are 0F8H and 00H mentioned above.

The first status code 0F8H indicates no remated information is got during transmission, meanwhile, SI flag is 0 and no IIC interrupt request.

The other status code 00H indicates errors occur during transmission, bus error is generated when START or stop signal appeared at illegal positon temporarily, for example the second bit change to 8th bit in address byte, or data byte and reponse bit error on bus, SI is set immediately, when IIC bus error is detected, the device immediately change to no-addressed slave mode, and release SDA and SCL line, set SI flag, load 00H to IICSTA. User want recover from bus error status, STO bit must be set logic 1 and SI muat be cleared to 0, then STO is cleared by hardware and release IIC bus when no stop signal.

Special case: if no successful generation of START or repeated start signal, IIC bus is resisted by low level of SDA, for example one slave CPU clock has not synchronization bit, user can send extra clock pulse on SCL to solve the problem. When STA is set, IIC hardware send extra clock pulse, but because SDA is pull down to 0, it can not generate start signal, shen SDA bus is released finally and send a normal START

condition, then enter status 08H, continuous to excute serial transmission. Shen SDA is low, if send repeated start signal, IIC hardware will excute the same operation above. Under this condition, after successfully send start signal, bus will enter status 08H, and not 10H.

Note: software can not solve these kind of bus problem.

16.9 IIC bus registers

16.9.1 IICcontrol register IICCON

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	CR2	IICEN	STA	STO	SI	AA	CR1	CR0

Bit	Flag	Introductions
7	CR2	IIC communication clock selection bit 2
6	IICEN	IIC module enble bit 0: diable IIC module 1: start up IIC module
5	STA	Start bit 0: Don’t send start signal 1: When bus idle generate start signal. When busy, wait stop signal then generate a start signal. In master mode, when IIC prepare tranmmit or receive one or multi-bytes, set 1 to generate a repeated start signal.
4	STO	Stop bit 0: Don’t send stop signal 1: Master mode generates stop signal, when detect stop signal apppeared on bus, IIC hardware clear STO flag. STO flag is used to recover IIC device from error status (IICSTA is 00H). Under this condition, no stop is sent on IIC bus. If STA and STO is set 1 all, and in master mode the device is original, IIC bus will generate stop signal followed with a start signal immediately, If the device in slave mode, set STO will return to no-addressed slave, STO will be cleared by hardware.
3	SI	IIC serial interrupt flag 0: no IIC serial interrupto ccur 1: Set 1 when generate IIC communication status code except 0F8H, must be cleared 0 by software.
2	AA	Acknowledge flag 0: Respond NACK (SDA is high) 1: Repond ACK (SDA is low)
1	CR1	IIC communication selection bit 1
0	CR0	IIC communication selection bit 0

CR[2:0] IIC communication clock selection bit:

CR2	CR1	CR0	F_{osc}				分频系数
			6MHz	12 MHz	16 MHz	24 MHz	
0	0	0	23KHz	47KHz	63KHz	92KHz	256
0	0	1	27KHz	54KHz	71KHz	108KHz	224
0	1	0	31KHz	63KHz	83KHz	124KHz	192
0	1	1	37KHz	75KHz	100KHz	148KHz	160
1	0	0	6.25KHz	12.5KHz	17KHz	25KHz	960
1	0	1	50KHz	100KHz	133KHz	200KHz	120
1	1	0	100KHz	200KHz	266KHz	400KHz	60
1	1	1	Overflow rate of UART2 BRT independent Baud rate generator /8				

16.9.2 IIC state register IICSTA

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

Bit	Flag	Introductions
7-3	IICSTA[7:3]	IIC status code, total have 26 possible status codes, SI bit can be set except status code 0F8H
2-0	-	Reserved

16.9.3 IIC data register IICDAT

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	IICDAT[7:0]							

Bit	Flag	Introductions
7:0	IICDAT[7:0]	IIC data IICDAT include one byte will be transmitted or received IIC data just now. only SI = 1, data in IICDAT will maintain, during IIC send/receive, the result to read or write IICDAT are all uncertain. When data in IICDAT is removed, data on bus is updated to IICDAT synchronously. IICDAT shows current last byte on IIC bus. So when lost arbitration, IICDAT original value will be changed after transmission.

16.9.4IIC address register IICADR

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	IICADR[7:1]							GC

Bit	Flag	Introductions
7-1	IICADR[7:1]	Slave mode: IIC device slave address itself Master mode: no effect
0	GC	Broadcast call bit 0: Broadcast call is ignored 1: If AA flag is1, broadcast call is recognized, if AA is 0, it is ignored. Note: the bit is valid in slave mode only, and no effect to master mode. When as slave, and set AA flag, in idle mode, if other master addressing address matchs to slave address, and slave will be woken up.

17 Analog to digital converter ADC

17.1 ADC characteristics

- Up to 23 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

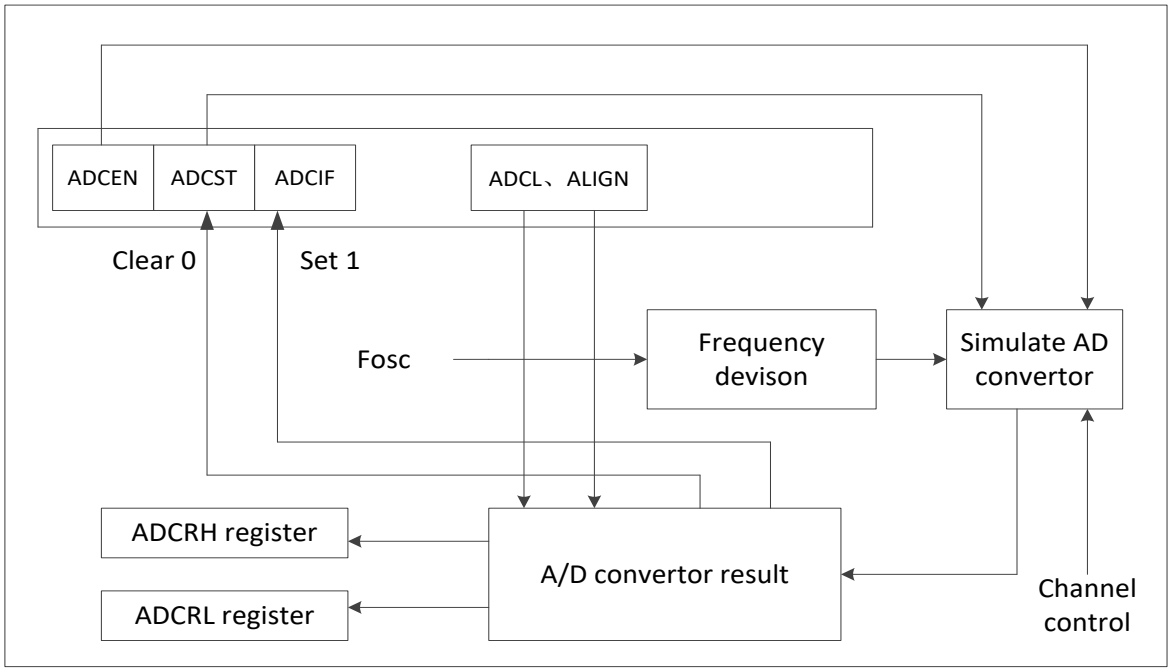


Figure 17 - 1 ADC functional block diagram

17.2 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.

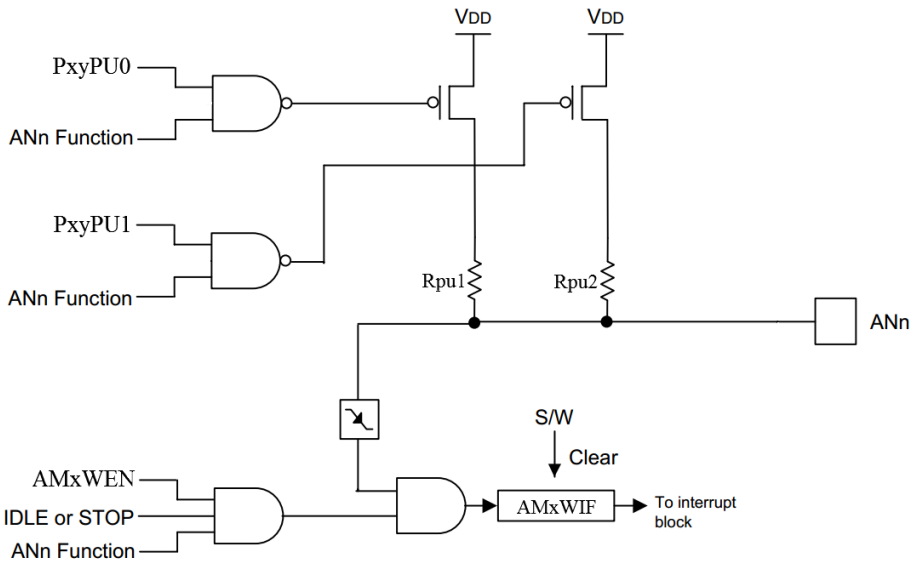


Figure 15 - 2 ADC power saving wakeup function block diagram

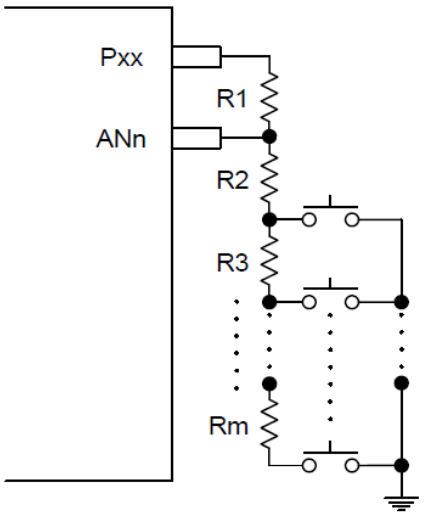


Figure 15 - 3 ADC series resistor key input application reference circuit

17.3 ADC registers

17.3.1 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	-	-	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)
4-3	-	Reserved (read = 0b, write invalid)
2	VREFS	VREF selection bit 0 : Select internal VREF 1 : Select external VREF_H (this moment P2.5 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	-	Reserved (read = 0b, write invalid)
4-0	XCHS[3:0]	ADC external input channel selection bits XCHS[4:0] = x(x = 0,2... 23), x defines the current test channel as ANx, such as XCHS[4:0] = 3, the current test channel is external channel AN3. Except external channel must be set XCHS[4:0], 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_{cpu}/2 001 : F_{cpu} /4 010 : F_{cpu} /6 011 : F_{cpu} /8 100 : F_{cpu} /12 101 : F_{cpu} /16 110 : F_{cpu} /24 111 : F_{cpu} /32

ADC conversion data format specification table

ADCL	ALIGN	ADCRH								ADCRL							
		7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0	0	D11	D10	D9	D8	D7	D6	D5	D4	/	/	/	/	D3	D2	D1	D0
0	1	/	/	/	/	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1	0	D11	D10	D9	D8	D7	D6	D5	D4	/	/	/	/	/	/	D3	D2
1	1	/	/	/	/	/	/	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2

17.3.2 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 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
7-0	ADCRL[7:0]	

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.

17.3.3 ADC wakeup control register ADCWC0、ADCWC1

ADCWC0

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	AM0WEN	AM0WIF	-	CHSW0[4:0]				

Bit	Flag	Introductions
7	AM0WEN	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	AM0WIF	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	-	Reserved bit
4-0	CHSW0[4:0]	ADC wake-up module x (x = 0, 1) input channel selection CHSWx [4:0] = y (y = 0... 23), indicating that the wake up channel of wake-up module 0 is selected as ANy, such as CHSW0[4:0] = 3, indicating that the wake up channel of wake-up module 0 is AN3. Note: channel selection must be allowed to awaken the module is valid, and wake up the channel ports must be set in advance of the need to use for analog input port. The wakeup resistor is configured through the port pull resistor.

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

18 Low voltage detection LVD

18.1 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 interrupt cannot wake the chip from power off mode, but interrupt can wake IDLE mode.

18.2 LVD registers

18.2.1 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	-	LVDIE	LVDM	LVDF	LVDV		

Bit	Flag	Introductions
7	LVDEN	LVD enable bit 0 : Disable LVD 1 : Enable LVD
6	-	Reserved bit
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.
4	LVDM	LVD test pattern choice bit 0: VDD voltage or port voltage is less than the check points, the LVDF hardware set 1 1: VDD voltage or port when the voltage is greater than the first, LVDF hardware set 1
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 or port voltage below detection voltage, the time is more than the debouncing time set in LVDDBC register, LVDF will be set; VDD or port voltage is higher than detection voltage, LVDF do not automatically clear, the bit must be software clear, only VDD or port voltage is higher than detection voltage continuously, software clear is valid, if VDD or port voltage 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.

18.2.2 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.

19 Cyclic redundancy check CRC

19.1 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.

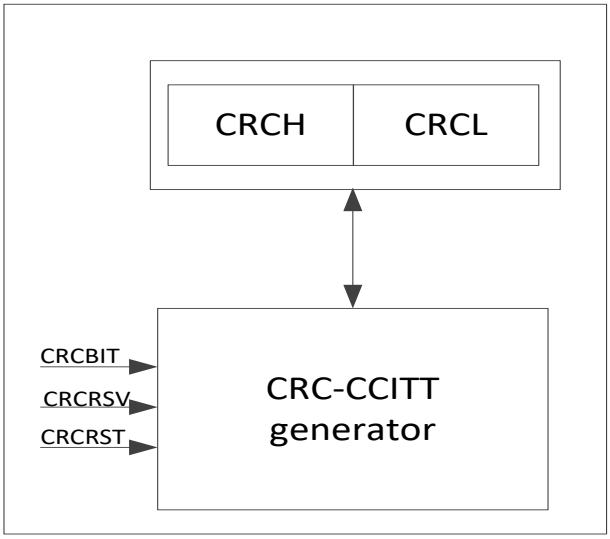


Figure 17 - 1 CRC functional block diagram

19.2 CRC registers

19.2.1 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

19.2.2 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.

20 Code options

1. External reset enable

- P4.7 as external reset Pin (default). When the port as external reset Pin, it cannot as a normal I/O.
- P4.7 as normal IO Pin

2. External reset level

- high level reset (default)
- low level reset

3. Wait time after reset

- 1ms
- 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.

21 Electrical characteristics

21.1 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	°C
Storage temperature	T_{STG}	-55	-	+125	°C

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

21.2 DC characteristics

Parameter	Symbol	Condition (VDD=5V)	Min	Typical	Max	Unit
Operating voltage	VDD	F_{CPU} =16MHz 32KHz, ADC module closed	2.0	5.0	5.5	V
Operating current	I_{OP1}	F_{CPU} =16MHz, No load, no floating input pins, execute NOP instructions, close the other modules	-	1.0	-	mA
		F_{CPU} =8MHz, No load, no floating input pins, execute NOP instructions, close the other modules	-	3.2	-	
		F_{CPU} =4MHz, No load, no floating input pins, execute NOP instructions, close the other modules	-	2.1	-	
		F_{CPU} =2MHz, No load, no floating input pins, execute NOP Instructions, close the other modules	-	1.5	-	
		F_{CPU} =1MHz, No load, no floating input pins, execute NOP Instructions, close the other modules	-	1.2	-	
		F_{CPU} =500KHz, No load, no floating input pins, execute NOP Instructions, close the other modules	-	1.1	-	
		F_{CPU} =125KHz, No load, no floating input pins, execute NOP Instructions, close the other modules	-	0.9	-	
	I_{OP2}	F_{OSC} =32MHz, No load, no floating input pins, execute NOP instructions, close the other modules		1.0	-	mA
	I_{ID1}	F_{CPU} =16MHz, No load, no floating input pins, execute NOP instructions, close the other modules		7.0	-	mA
	I_{ID2}	F_{OSC} =32KHz, No load, no floating input pins, execute NOP instructions, close the other modules		1.0	-	mA
	I_{PD}	Enter the power-down mode, no load, no floating input pins, close all modules	-	7.0	-	μA
	I_{IDLE1}	F_{CPU} =16MHz, Enter the idle mode, no load, no floating input pins, close all modules	-	2	-	mA
	I_{IDLE2}	F_{CPU} =8MHz, Enter the idle mode, no load, no floating input pins, close all modules	-	1.3	-	mA
	I_{IDLE3}	F_{CPU} =4MHz, Enter the idle mode, no load, no floating input pins, close all modules	-	1.0	-	mA
	I_{IDLE4}	F_{CPU} =2MHz, Enter the idle mode, no load, no floating input pins, close all modules	-	0.8	-	mA
	I_{IDLE5}	F_{CPU} =1MHz, Enter the idle mode, no load, no floating input pins, all closed,	-	0.7	-	mA

		internal high-frequency RC Clock close				
	I_{IDLE6}	F_{CPU} =125KHz, Enter the idle mode, no load, no floating input pins, all closed, internal high-frequency RC Clock close		0.7		mA
	I_{RTC}	RTC enable interrupt, enter power off mode, no floating input pin, ADC reference voltage select non-VDD, other modules shut down		50		μA
WDT Current	I_{WDT}	VDD = 5V	-	1.2	-	μA
LVD current	I_{LVD}	VDD = 5V	-	11.0	-	
RTC current	I_{RTC}	VDD = 5V, The system goes into power off mode	-	11.0	-	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_{OL1}	Vout=GND+0.6，High Drive Mode	-	25	-	mA
	I_{OL2}	Vout=GND+0.6，Low Drive Mode	-	8	-	
	I_{OL3}	Vout=GND+1.5，High Drive Mode onlyP3.0-P3.7(+1.5V@5V)	-	100	-	
	I_{OL4}	Vout=GND+1.5，Low Drive Mode onlyP3.0-P3.7(+1.5V@5V)	-	35	-	
Current	I_{OH1}	Vout=VDD-0.6，High Drive Mode	-	15	-	
	I_{OH2}	Vout=VDD-0.6，Low Drive Mode	-	5	-	
Pull-up resistor	R_{PU1}	Common port, VIN=GND	-	30	-	kΩ
	R_{PU2}	Common port, VIN=GND	-	50	-	
	R_{PU3}	Common port, VIN=GND	-	120	-	
	R_{PU4}	Common port, VIN=GND	-	230	-	
	R_{PU5}	External reset port low efficiency, VIN=GND	-	30	-	
Pull-down resistance	R_{PD1}	Common port, VIN=VDD	-	45	-	
	R_{PD2}	External reset port low efficiency, VIN=GND	-	45	-	
ADC Wake up voltage	V_{AW1}	room temperature, VDD = 5V	4.0	4.2	4.4	V
	V_{AW2}	room temperature, VDD = 3V	2.3	2.5	2.6	
RAM maintain voltage	V_{RAM}	-	-	0.7	-	

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

21.3 AC characteristics

Parameter	Symbol	Conditions	Min	Typical	Max	Unit
Internal RC 32M Startup time	Tset1	room temperature, VDD=5V	-	-	5	μs
Internal RC 38K startup time	Tset2	room temperature, VDD=5V	-	-	150	μs
External	Tset3	16MHz, room temperature, VDD=5V	-	200	-	μs

high-frequency oscillator startup time						
External 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
Frequency accuracy	FIRC1	VDD=2V~5.5V, 25°C	32 (1-1%)	32	32 (1+1%)	MHz
	FIRC2	VDD=5.0V,-40°C ~+85°C	32 (1-1%)	32	32 (1+1%)	MHz
	FWRC	-	31	44	19	38

57

21.4 ADC characteristics

Parameter	Symbol	Conditions	Min	Typical	Max	Unit
power supply voltage	VAD	-	2.5	5.0	5.5	V
Precision	NR	GND ≤ VAIN ≤ Vref	-	12	-	bit
ADC input voltage	VAIN	-	GND	-	Vref	V
ADC input resistance	RAIN	VAIN=5V	2	-	-	MΩ
Analog voltage sources recommended impedance	ZAIN	-	-	-	1	kΩ
ADC switching current	IAD	ADC module opened, VDD=5.0V	-	1	3	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	-	-	±4	LSB
		VDD=5.0V, Vref =3V	-	-	±5	
		VDD=5.0V, Vref =4V	-	-	±6	
		VDD=5.0V, Vref =VDD	-	-	±7	
		VDD=5.0V, Vref = External reference voltage	-	-	±7	
Full scale error	EF	VDD=5.0V	-	-	±10	LSB
Offset error	EZ	VDD=5.0V	-	-	±5	LSB
Total error	EAD	VDD=5.0V	-	-	±10	LSB
Total conversion time1	TCON1	VDD=5.0V	10	-	-	μs
Total conversion time2	TCON2	VDD=5.0V, Vref =VDD	2	-	-	μs
Internal reference voltage	VADREF	±1%,room temperature	2	-	-	V

5.0
12
V
-
1
3
-
-

21.5 FLASH Characteristics

Parameter	Symbol	Condition	Min	Typ	Max	Unit
Write and Read Test	N_{ENDUR}	-	100000	-	-	Cycle
Data retention time	T_{RET}	T=25°C	-	10	-	year
Sector erase time	T_{ERASE}	1 sector（1281byte）	-	5	-	ms
Byte was written time	T_{PROG}	1byte, Fcpu=16MHz	-	23	-	us
Read current consumption	I_{DD1}	Fcpu=16MHz	-	4	-	mA
Write current consumption	I_{DD2}	-	-	4	-	mA

Erase current consumption	I_{DD3}	-	-	2	-	mA

21.6 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

21.7 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

21.8 Other electrical characteristics

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

22 Development tools

22.1 Emulator characteristics

HC89S105xx 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

22.2 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:

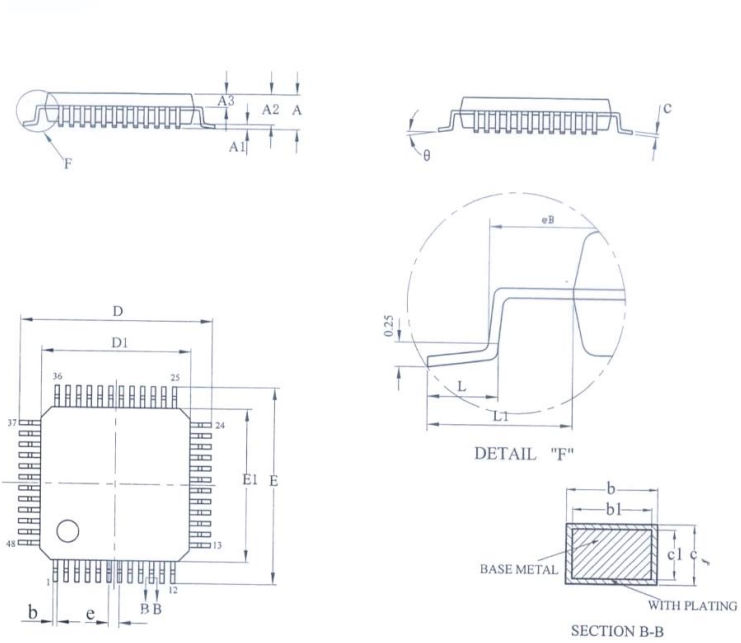
- USB port connection
- Support signal channel off-line programming

22.3 Software download

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

23 Package

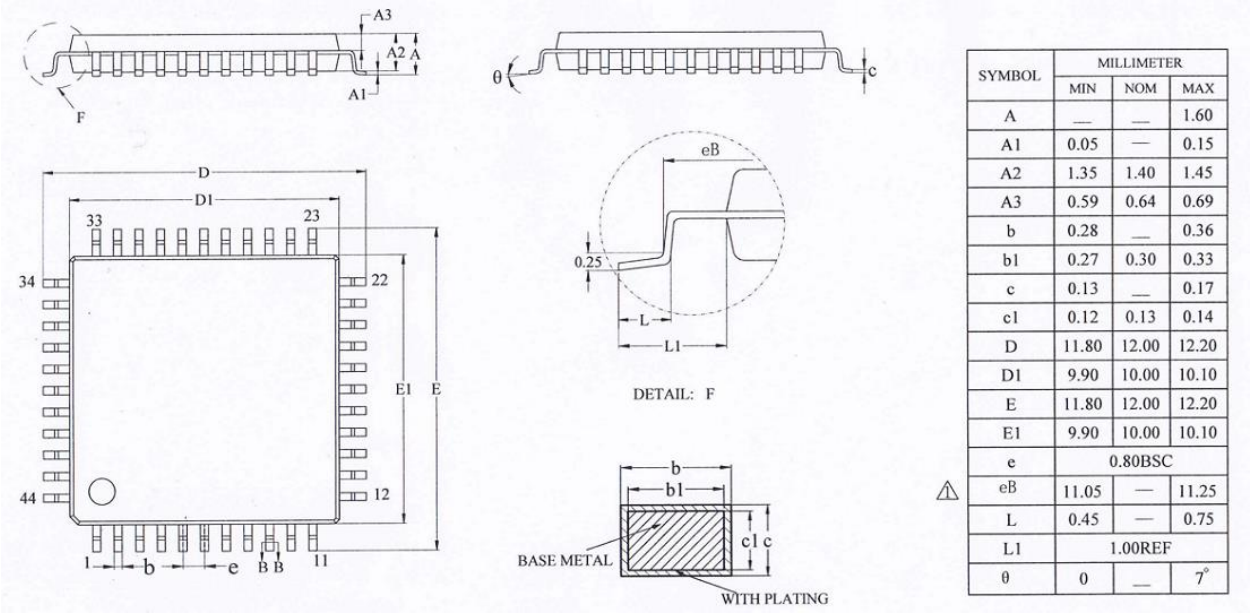
23.1 LQFP48



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.19	—	0.27
b1	0.18	0.20	0.23
c	0.13	—	0.18
c1	0.12	0.13	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
eB	8.10	—	8.25
e	0.50BSC		
L	0.45	—	0.75
L1	1.00BSC		
θ	0	—	7°

Figure 23 - 1 LQFP48 package size

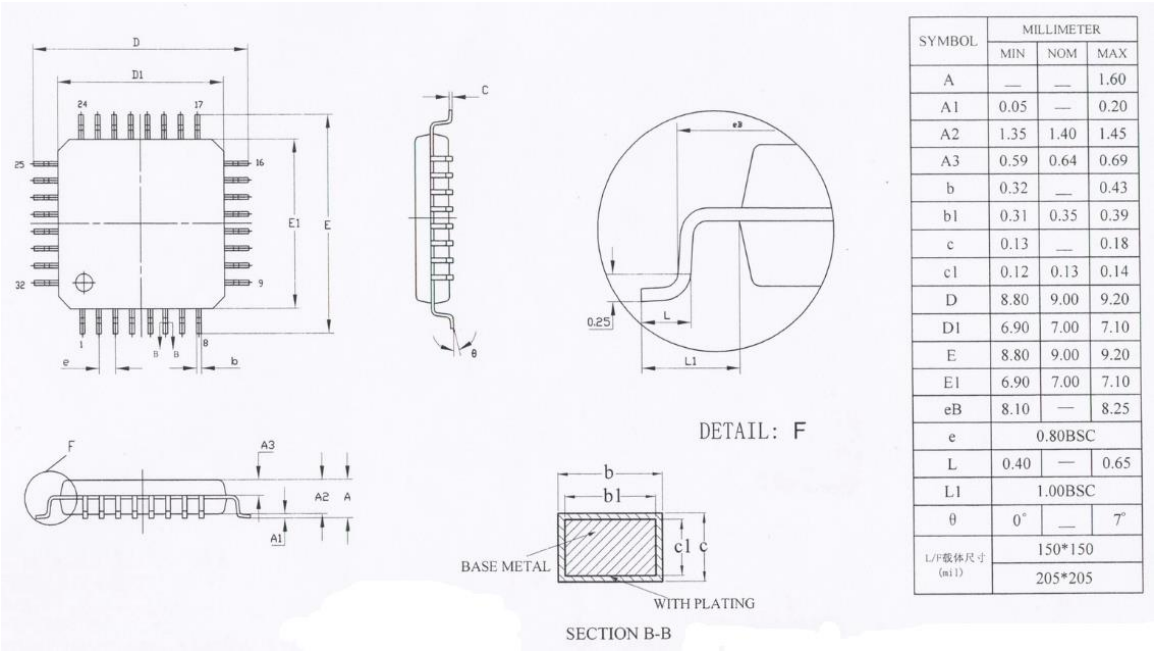
23.2 LQFP44



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.28	—	0.36
b1	0.27	0.30	0.33
c	0.13	—	0.17
c1	0.12	0.13	0.14
D	11.80	12.00	12.20
D1	9.90	10.00	10.10
E	11.80	12.00	12.20
E1	9.90	10.00	10.10
e	0.80BSC		
⚠ eB	11.05	—	11.25
L	0.45	—	0.75
L1	1.00REF		
θ	0	—	7°

Figure 23-1 LQFP44 package size

23.3 LQFP32



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	—	—	1.60
A1	0.05	—	0.20
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.32	—	0.43
b1	0.31	0.35	0.39
c	0.13	—	0.18
c1	0.12	0.13	0.14
D	8.80	9.00	9.20
D1	6.90	7.00	7.10
E	8.80	9.00	9.20
E1	6.90	7.00	7.10
eB	8.10	—	8.25
e	0.80BSC		
L	0.40	—	0.65
L1	1.00BSC		
θ	0°	—	7°
L/F载体尺寸 (mil)	150*150		
	205*205		

Figure 23-3 LQFP32 package size

24 Reversion history

Document revision history

Version	Date	Description
Ver1.00	2020-12-14	First version