HC89S105AC8
HC89S105AS8
HC89S105AK8
Datasheet

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

Contents

1 Description

HC89S105A is an enhanced 8 bit microcontroller with high frequency and low power consumption CMOS process. It has up to 64K Bytes FLASH program memory, 256 Bytes IRAM and 4K Bytes XRAM, 6 bi-directional I/O, 1 peripheral function Ports total mapping module ,six 16-bit Timer/counters,2 PCA,3 groups 16 bits complementary PWM with dead-time control, 1 channel 8 bit PWM output, 2 UART, 1 IIC, 1 SPI, up to 32+2 channels 12 bits ADC, Integrated 16 bit*16bit multiplier, 16 bit/16 bit divisor , 32 bit/16 bit divisor, 4 system work modes (normal, low-frequency, STOP, IDLE) and 20 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
  - 4K Bytes XRAM
- **Clock**
  - Internal high precision 32MHz RC
  - External high-frequency oscillator 4MHz-24MHz
  - External low-frequency oscillator 32.768KHz
  - Peripheral module clock can switch independently
- **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**
  - 46/42/30 bi-directional IO
  - Multiple modes configurable
  - 4 levels of port drive current to choose from
  - peripheral function Ports total mapping module
  - All ports can be configured as 1/2BIAS and 1/3BIAS software LCD drivers
- **interrupt**
  - Up to 20 interrupt sources
  - 4 level interrupt priorities
  - 16 external interrupts
- **Timer/Counter**
  - T0/T1 compatible with standard 8051, 16-bit auto reload
  - T3 can run in STOP mode
  - T4 can be trigged by external signal
  - T5 with capture function
- **PWM**
  - Up to 3 groups 16 bits complementary PWM with dead-time control
  - Malfunction detection function
  - 1 channel 8 bit PWM output
- **PCA**
  - 16 bit capture/compare/software timing
  - 8 bit PWM
- **Communication interfaces**
  - 2 UART
  - 1 SPI
  - 1 IIC
- **Analog to digital converter (ADC)**
  - 12 bit ADC, up to 32+2 multiple channels
  - ADC reference voltage is optional internal or external Vref
  - analog watchdog
- **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 (1.2 V), with the interrupt
  - Comparator function
- **Cyclic redundancy check(CRC)**
- **Multipliers and dividers**
  - Integrated 16 bit*16bit multiplier, 16 bit/16 bit divisor, 32 bit/16 bit divisor
- **Power saving mode**
  - IDLE mode
  - STOP mode
- Support SWD, JTAG simulation and download
- **Operating conditions**
  - Wide operating voltage 2.0V to 5.5V
  - Temperature range -40°C to +105°C
- **Package**
  - LQFP48，LQFP44，LQFP32

✓ **Selection table**

Product model	ROM	RAM	I/O	Timer	PWM	A/D	INT	PCA	IIC	UART	SPI	WDT	Voltage	TEMP	Package
HC89S105AC8T7M	64K	256+4K	46	6	16bit*3 groups	32+2	16	2	1	2	1	1	2.0~5.5V	-40~+105°C	LQFP48
HC89S105AS8T7CM	64K	256+4K	42	6	16bit*3 groups	31+2	15	2	1	2	1	1	2.0~5.5V	-40~+105°C	LQFP44
HC89S105AK8T7CM	64K	256+4K	30	6	16bit*3 groups	23+2	12	2	1	2	1	1	2.0~5.5V	-40~+105°C	LQFP32

Product model	Simulator	Programmer	Datasheet	DemoCode	DemoBoard
HC89S105AC8T7M HC89S105AS8T7CM HC89S105AK8T7CM	HC-LINK	HC-PM51	√	√	√

1.2 Naming rule

HC89	S	105	A	C	8	T	7	M
Code	**Product type**	**Product line**	**Version**	**Pin number**	**ROM**	**Package**	**TEMP**	**Logo**
Holychip 1T 8051	S: Standard L: Low-power dissipation P: Op-amp type	0xx：Value type 1xx：Basic type 2xx：Enhancement mode	Omit：First edition A: First upgrade B: Second upgrade	J: 8pin P: 16pin F: 20pin K: 32pin S: 44pin C: 48pin R: 64pin	3: 8KB 4: 16KB 5: 24KB 6: 32KB 7: 48KB 8: 64KB	P:TSSOP U:QFN M:SOP T:LQFP	6: -40℃~85℃ 7: -40℃~105℃	CM: With positive print 0.8mm package pitch M: With positive print 0.5mm package pitch

1.3 Use attention

1. In order to ensure the system stability, user must connect a capacitor (≥0.1μF) between VDD and GND.
2. The P4.7 of HC89S105AC8/S8 or P4.1 of HC89S105AK8 is a reset pin by default, and the port mode is schmidt-input strip pull. This port can be configured as a common IO pin by configuration code option.
3. If FLASH IAP is required, read the Precautions in 3.1.3.1 carefully.
4. Do not respond to any interruption while IAP is in operation.
5. After ADCEN is set to 1 or conversion channel is switched, it is recommended to start ADC conversion after a delay of 20us. If the external input impedance is large, this time should be extended.
6. When the reference voltage of the ADC is VDD, the ADC conversion clock can be 8MHz, and only 15 ADC_CLK are needed for one conversion, which can achieve the fastest ADC conversion speed.
   When UART2 uses full-duplex mode, you need to set the CPU frequency to 16MHz or higher, set the baud rate to 9600 or lower, and clear TI or RI in the UART2 interrupt service function as soon as possible.

1.4 System diagram

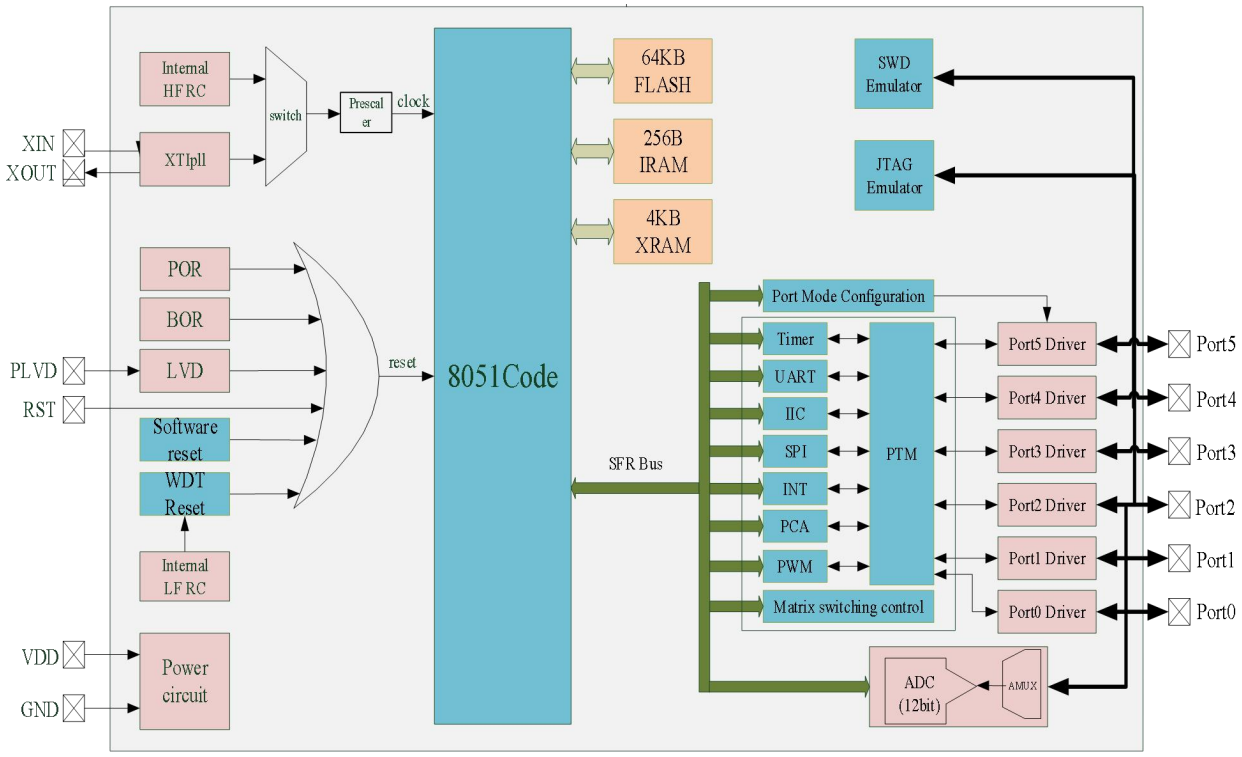


Figure 1- 1 System diagram

1.5 Pin configuration

1.5.1 LQFP48 Pin configuration

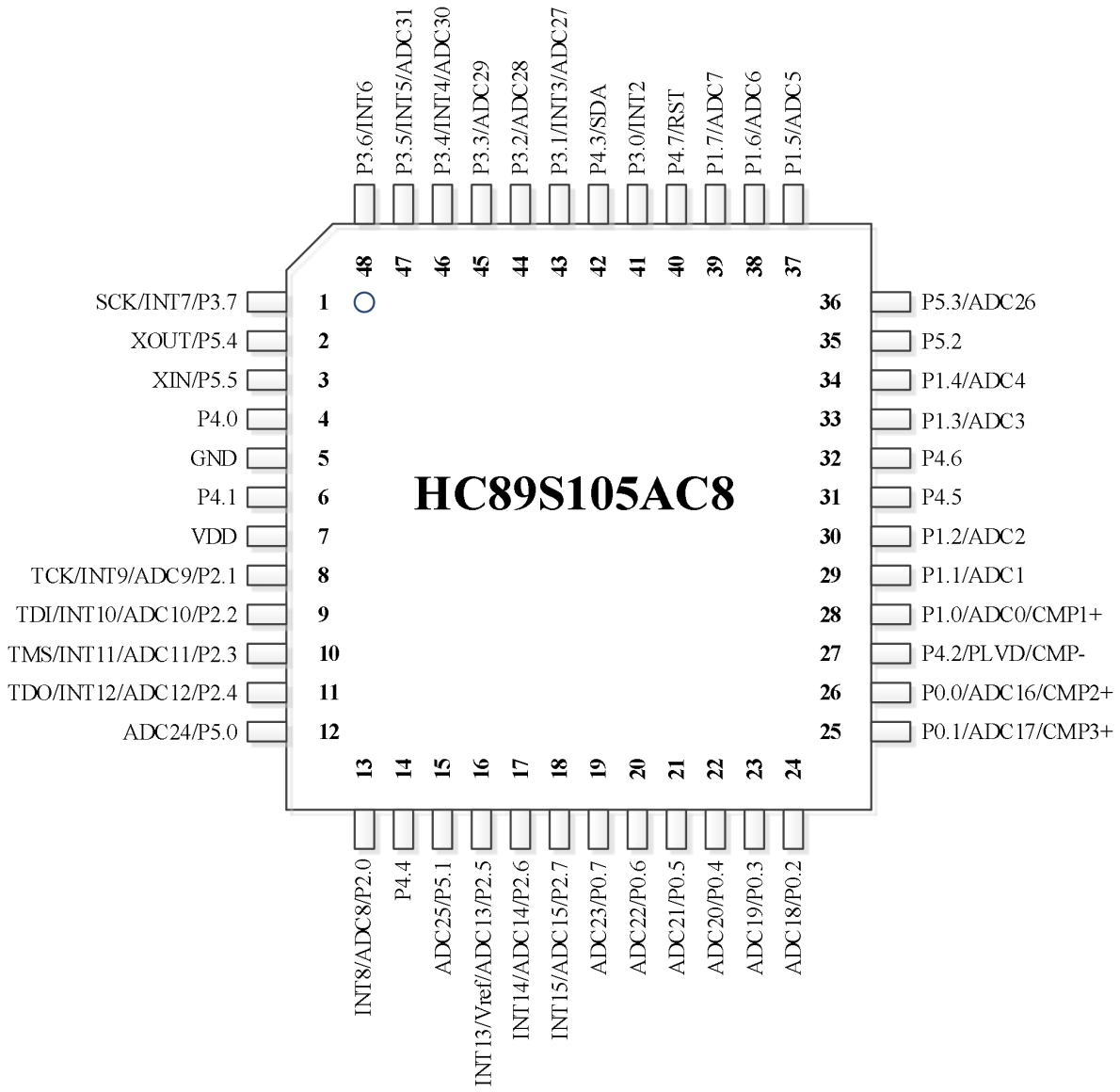


Figure 1-2 LQFP48 pin configuration diagram

1.5.2 LQFP44 Pin configuration

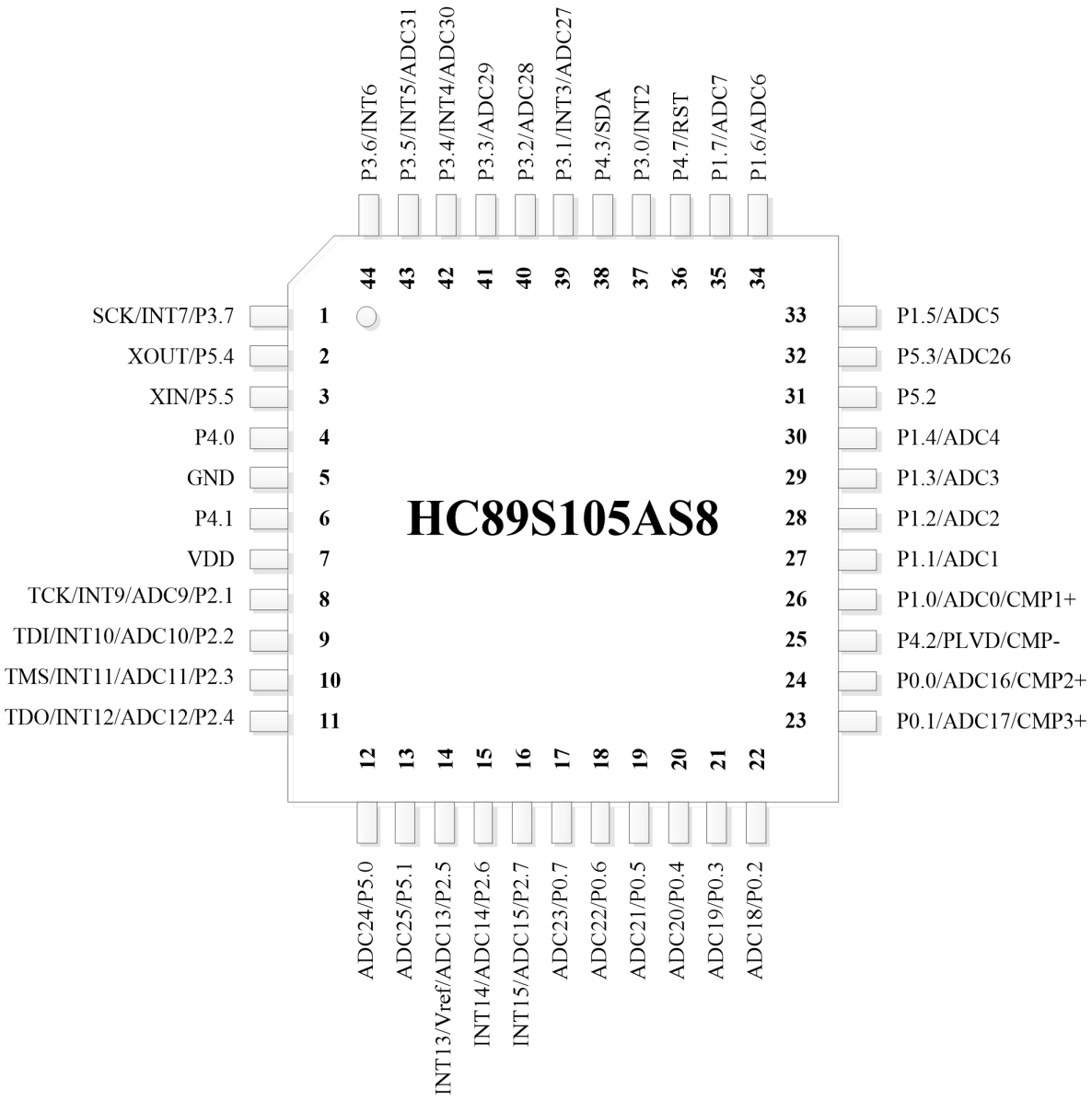


Figure 1-3 LQFP44 pin configuration diagram

1.5.3 LQFP32 Pin configuration

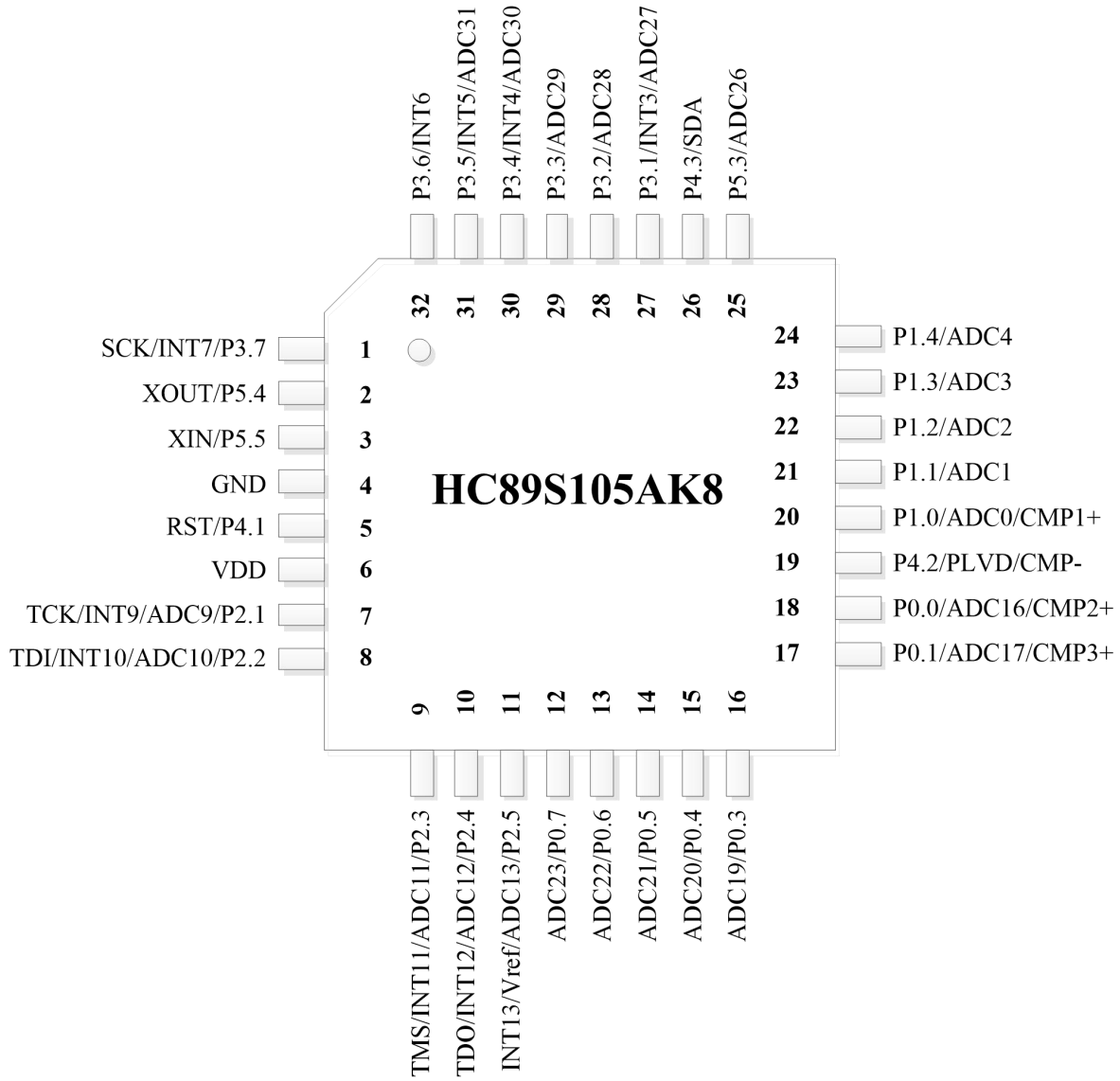


Figure 1-4 LQFP32 pin configuration diagram

1.6 Pin description

1.6.1 LQFP48 Pin description

Pin	Name	Type	Introductions
1	P3.7	I/O	Input/output port
	INT7	I	External interrupt 7, input port
	SCK	I	SWD mode clock input
2	P5.4	I/O	Input/output port
	XOUT	AN	External oscillator output
3	P5.5	I/O	Input/output port
	XIN	AN	External oscillator input
4	P4.0	I/O	Input/output port
5	GND	P	Power ground
6	P4.1	I/O	Input/output port
7	VDD	P	Power input
8	P2.1	I/O	Input/output port
	ADC9	AN	ADC input port
	INT9	I	External interrupt 9, input port
	TCK	I	JTAG clock input
9	P2.2	I/O	Input/output port
	ADC10	AN	ADC input port
	INT10	I	External interrupt 10, input port
	TDI	I	JTAG data input
10	P2.3	I/O	Input/output port
	ADC11	AN	ADC input port
	INT11	I	External interrupt 11, input port
	TMS	I	JTAG mode input
11	P2.4	I/O	Input/output port
	ADC12	AN	ADC input port
	INT12	I	External interrupt 12, input port
	TDO	O	JTAG data output
12	P5.0	I/O	Input/output port
	ADC24	AN	ADC input port
13	P2.0	I/O	Input/output port
	ADC8	AN	ADC input port
	INT8	I	External interrupt 8, input port
14	P4.4	I/O	Input/output port
15	P5.1	I/O	Input/output port
	ADC25	AN	ADC input port
16	P2.5	I/O	Input/output port
	ADC13	AN	ADC input port
	INT13	I	External interrupt 13, input port
	Vref	AN	ADC external reference voltage input/output port
17	P2.6	I/O	Input/output port

	ADC14	AN	ADC input port
	INT14	I	External interrupt 14, input port
18	P2.7	I/O	Input/output port
	ADC15	AN	ADC input port
	INT15	I	External interrupt 15, input port
19	P0.7	I/O	Input/output port
	ADC23	AN	ADC input port
20	P0.6	I/O	Input/output port
	ADC22	AN	ADC input port
21	P0.5	I/O	Input/output port
	ADC21	AN	ADC input port
22	P0.4	I/O	Input/output port
	ADC20	AN	ADC input port
23	P0.3	I/O	Input/output port
	ADC19	AN	ADC input port
24	P0.2	I/O	Input/output port
	ADC18	AN	ADC input port
25	P0.1	I/O	Input/output port
	ADC17	AN	ADC input port
	CMP3+	AN	Comparator positive end 3 input port
26	P0.0	I/O	Input/output port
	ADC16	AN	ADC input port
	CMP2+	AN	Comparator positive end 2 input port
27	P4.2	I/O	Input/output port
	PLVD	AN	Port low voltage detection port
	CMP-	AN	Comparator negative input port
28	P1.0	I/O	Input/output port
	ADC0	AN	ADC input port
	CMP1+	AN	Comparator positive end 1 input port
29	P1.1	I/O	Input/output port
	ADC1	AN	ADC input port
30	P1.2	I/O	Input/output port
	ADC2	AN	ADC input port
31	P4.5	I/O	Input/output port
32	P4.6	I/O	Input/output port
33	P1.3	I/O	Input/output port
	ADC3	AN	ADC input port
34	P1.4	I/O	Input/output port
	ADC4	AN	ADC input port
35	P5.2	I/O	Input/output port
36	P5.3	I/O	Input/output port
	ADC26	AN	ADC input port
37	P1.5	I/O	Input/output port
	ADC5	AN	ADC input port
38	P1.6	I/O	Input/output port
	ADC6	AN	ADC input port

39	P1.7	I/O	Input/output port
	ADC7	AN	ADC input port
40	P4.7	I/O	Input/output port
	RST	I	External reset input port
41	P3.0	I/O	Input/output port
	INT2	I	External interrupt 2, input port
42	P4.3	I/O	Input/output port
	SDA	I	SWD data input/output
43	P3.1	I/O	Input/output port
	INT3	I	External interrupt 3, input port
	ADC27	AN	ADC input port
44	P3.2	I/O	Input/output port
	ADC28	AN	ADC input port
45	P3.3	I/O	Input/output port
	ADC29	AN	ADC input port
46	P3.4	I/O	Input/output port
	INT4	I	External interrupt 4, input port
	ADC30	AN	ADC input port
47	P3.5	I/O	Input/output port
	INT5	I	External interrupt 5, input port
	ADC31	AN	ADC input port
48	P3.6	I/O	Input/output port
	INT6	I	External interrupt 6, input port

Note: I = Input, O =Output, I/O = Input/ Output, P =Power, AN = Analog input/output.

1.6.2 LQFP44 Pin description

Pin	Name	Type	Introductions
1	P3.7	I/O	Input/output port
	INT7	I	External interrupt 7, input port
	SCK	I	SWD mode clock input
2	P5.4	I/O	Input/output port
	XOUT	AN	External oscillator output
3	P5.5	I/O	Input/output port
	XIN	AN	External oscillator input
4	P4.0	I/O	Input/output port
5	GND	P	Power ground
6	P4.1	I/O	Input/output port
7	VDD	P	Power input
8	P2.1	I/O	Input/output port
	ADC9	AN	ADC input port
	INT9	I	External interrupt 9, input port
	TCK	I	JTAG clock input
9	P2.2	I/O	Input/output port
	ADC10	AN	ADC input port
	INT10	I	External interrupt 10, input port
	TDI	I	JTAG data input

10	P2.3	I/O	Input/output port
	ADC11	AN	ADC input port
	INT11	I	External interrupt 11, input port
	TMS	I	JTAG mode input
11	P2.4	I/O	Input/output port
	ADC12	AN	ADC input port
	INT12	I	External interrupt 12, input port
	TDO	O	JTAG data output
12	P5.0	I/O	Input/output port
	ADC24	AN	ADC input port
13	P5.1	I/O	Input/output port
	ADC25	AN	ADC input port
14	P2.5	I/O	Input/output port
	ADC13	AN	ADC input port
	INT13	I	External interrupt 13, input port
	Vref	AN	ADC external reference voltage input/output port
15	P2.6	I/O	Input/output port
	ADC14	AN	ADC input port
	INT14	I	External interrupt 14, input port
16	P2.7	I/O	Input/output port
	ADC15	AN	ADC input port
	INT15	I	External interrupt 15, input port
17	P0.7	I/O	Input/output port
	ADC23	AN	ADC input port
18	P0.6	I/O	Input/output port
	ADC22	AN	ADC input port
19	P0.5	I/O	Input/output port
	ADC21	AN	ADC input port
20	P0.4	I/O	Input/output port
	ADC20	AN	ADC input port
21	P0.3	I/O	Input/output port
	ADC19	AN	ADC input port
22	P0.2	I/O	Input/output port
	ADC18	AN	ADC input port
23	P0.1	I/O	Input/output port
	ADC17	AN	ADC input port
	CMP3+	AN	Comparator positive end 3 input port
24	P0.0	I/O	Input/output port
	ADC16	AN	ADC input port
	CMP2+	AN	Comparator positive end 2 input port
25	P4.2	I/O	Input/output port
	PLVD	AN	Port low voltage detection port
	CMP-	AN	Comparator negative input port
26	P1.0	I/O	Input/output port
	ADC0	AN	ADC input port
	CMP1+	AN	Comparator positive end 1 input port

27	P1.1 ADC1	I/O AN	Input/output port ADC input port
28	P1.2 ADC2	I/O AN	Input/output port ADC input port
29	P1.3 ADC3	I/O AN	Input/output port ADC input port
30	P1.4 ADC4	I/O AN	Input/output port ADC input port
31	P5.2	I/O	Input/output port
32	P5.3 ADC26	I/O AN	Input/output port ADC input port
33	P1.5 ADC5	I/O AN	Input/output port ADC input port
34	P1.6 ADC6	I/O AN	Input/output port ADC input port
35	P1.7 ADC7	I/O AN	Input/output port ADC 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 SDA	I/O I	Input/output port SWD data input/output
39	P3.1 INT3 ADC27	I/O I AN	Input/output port External interrupt 3, input port ADC input port
40	P3.2 ADC28	I/O AN	Input/output port ADC input port
41	P3.3 ADC29	I/O AN	Input/output port ADC input port
42	P3.4 INT4 ADC30	I/O I AN	Input/output port External interrupt 4, input port ADC input port
43	P3.5 INT5 ADC31	I/O I AN	Input/output port External interrupt 5, input port ADC input port
44	P3.6 INT6	I/O I	Input/output port External interrupt 6, input port

Note: I = Input, O =Output, I/O = Input/ Output, P =Power, AN = Analog input/output.

1.6.3 LQFP32 Pin description

Pin	Name	Type	Introductions
1	P3.7	I/O	Input/output port
	INT7	I	External interrupt 7, input port
	SCK	I	SWD mode clock input
2	P5.4	I/O	Input/output port
	XOUT	AN	External oscillator output
3	P5.5	I/O	Input/output port
	XIN	AN	External oscillator input
4	GND	P	Power ground
5	P4.1	I/O	Input/output port
	RST	I	External reset input port
6	VDD	P	Power input
7	P2.1	I/O	Input/output port
	ADC9	AN	ADC input port
	INT9	I	External interrupt 9, input port
	TCK	I	JTAG clock input
8	P2.2	I/O	Input/output port
	ADC10	AN	ADC input port
	INT10	I	External interrupt 10, input port
	TDI	I	JTAG data input
9	P2.3	I/O	Input/output port
	ADC11	AN	ADC input port
	INT11	I	External interrupt 11, input port
	TMS	I	JTAG mode input
10	P2.4	I/O	Input/output port
	ADC12	AN	ADC input port
	INT12	I	External interrupt 12, input port
	TDO	O	JTAG data output
11	P2.5	I/O	Input/output port
	ADC13	AN	ADC input port
	INT13	I	External interrupt 13, input port
	Vref	AN	ADC external reference voltage input/output port
12	P0.7	I/O	Input/output port
	ADC23	AN	ADC input port
13	P0.6	I/O	Input/output port
	ADC22	AN	ADC input port
14	P0.5	I/O	Input/output port
	ADC21	AN	ADC input port
15	P0.4	I/O	Input/output port
	ADC20	AN	ADC input port
16	P0.3	I/O	Input/output port
	ADC19	AN	ADC input port

17	P0.1	I/O	Input/output port
	ADC17	AN	ADC input port
	CMP3+	AN	Comparator positive end 3 input port
18	P0.0	I/O	Input/output port
	ADC16	AN	ADC input port
	CMP2+	AN	Comparator positive end 2 input port
19	P4.2	I/O	Input/output port
	PLVD	AN	Port low voltage detection port
	CMP-	AN	Comparator negative input port
20	P1.0	I/O	Input/output port
	ADC0	AN	ADC input port
	CMP1+	AN	Comparator positive end 1 input port
21	P1.1	I/O	Input/output port
	ADC1	AN	ADC input port
22	P1.2	I/O	Input/output port
	ADC2	AN	ADC input port
23	P1.3	I/O	Input/output port
	ADC3	AN	ADC input port
24	P1.4	I/O	Input/output port
	ADC4	AN	ADC input port
25	P5.3	I/O	Input/output port
	ADC26	AN	ADC input port
26	P4.3	I/O	Input/output port
	SDA	I	SWD data input/output
27	P3.1	I/O	Input/output port
	INT3	I	External interrupt 3, input port
	ADC27	AN	ADC input port
28	P3.2	I/O	Input/output port
	ADC28	AN	ADC input port
29	P3.3	I/O	Input/output port
	ADC29	AN	ADC input port
30	P3.4	I/O	Input/output port
	INT4	I	External interrupt 4, input port
	ADC30	AN	ADC input port
31	P3.5	I/O	Input/output port
	INT5	I	External interrupt 5, input port
	ADC31	AN	ADC input port
32	P3.6	I/O	Input/output port
	INT6	I	External interrupt 6, input port

Note: I = Input, O =Output, I/O = Input/ Output, P =Power, AN = Analog input/output.

1.7 Peripheral function Ports total mapping module (PTM)

HC89S105A has peripheral function Ports total mapping module internal, by software user can configure most peripheral function to arbitrary port except power port (VDD, GND).

1.7.1 PTM module characteristics

- When set peripheral port as input (T0/1/3/5 external input, RXD and so on) function, system permit multi to one mapping, that is multi-input peripheral functions port are distributed the same IO, the method will optimize the user's system.
- When set peripheral port as output (T0/1/4 clock output, TXD and so on) function, if multi-output peripheral functions port are distributed the same IO, it will follow fixed priority, only one output is valid.
- Software operation, use flexible, when use design system, don't care the Pins layout of peripheral functions, it can reduce the development cost.
- When user meets layout errors of peripheral function Pins on PCB, user can re-distribute peripheral functions by PTM module, and shorten development period.
- When user changes the peripheral components during system design, only need minimum changes, it will reduce the cost of system maintenance.

1.7.2 PTM support peripheral function Ports total mapping

Peripheral	Name	Type	Instructions
Timer	T0	I/O	T0 external input orT0 clock scale output
	T1	I/O	T1 external input orT1 clock scale output
	T3	I	T3 external input
	T4	O	T4 ouput
	T5	I	T5 external input
	T6	O	T6 clock divider output
	CAP0	I	Capture the input for channel 0
	CAP1	I	Capture the input for channel 1
PCA	ECI	I	PCA external input
	PCA0	I/O	PCA0 Input/output port
	PCA1	I/O	PCA1 Input/output port
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
CLK	CLKO	O	Clock output port
UART	TXD	O	UART1 data transmission port
	RXD	I/O	UART1 receive port
	TXD2	O	UART2 data transmission port
	RXD2	I/O	UART2 receive port
	BRTO	O	BRT clock divider output

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
ADC	ADCST	I	ADC external start input port
External interrupt	INT0	I	External interrupt 0
	INT1	I	External interrupt 1

1.7.3 PTM does not support peripheral function Ports total mapping

PTM does not support peripheral function Ports total mapping include power port(VDD, GND) ， ADC input port，Vref pin，oscillator Pin(Xin,XOUT)，External interrupt 2-15 input port，$\overline{RST}$.

2 CPU

2.1 CPU characteristics

HC89S105A 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 program counter PC

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

2.2.2 Accumulator ACC

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

2.2.3 Register B

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

2.2.4 Program state word register PSW

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

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

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

		0 : In arithmetic, no auxiliary carry or borrow 1 : In arithmetic, auxiliary carry or borrow has occurred
5	F0	User-defined flag
4-3	RS[1:0]	register group selection flag 00 : 0 Group (00H ~ 07H) 01 : 1 Group (08H ~ 0FH) 10 : 2 Group (10H ~ 17H) 11 : 3 Group (18H ~ 1FH)
2	OV	Overflow flag 0: no overflow 1 : Overflow has occurred
1	F1	User-defined flag
0	P	Parity bit 0 : sum of 1 in ACC register is 0 or EVEN 1 : Sum of 1 in ACC register is ODD

2.2.5 Stack pointer SP

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

2.2.6 Data pointer DPTR

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

2.2.7 Data pointer select register INSCON

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

Bit	Flag	Introductions
7-5	-	Reserved (read = 0b, write invalid)
4	IAPS	MOVC operation selection bit 0 : program area read/erase/write operation 1 : OPTION read operation
3-1	-	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
- 128 bytes for a sector, 8 sectors for a page(1KB)

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); 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 ICP operation 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 1K bytes, when 1K 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 1K bytes too, when the corresponding 1K bytes erase and write protection enabled, ICP will not be able to erase and program 4K bytes, strong writing is disabled.

If the corresponding 1K 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 1K bytes, before IAP erase and write, the corresponding sectors erase and write protection must be disabled.

If the corresponding 1K 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-LINK user manual.

3.1.3 OPTION

There is a read-only OPTION area besides 64K bytes ROM, storage data include: user setting data, user passwords, chip configurations data, the second reset vector data related. Address distribution in below table.

Address	Name	Address offset	Name	Address	Name	Address offset	Name
0x0000	SN_DATA0	0x0020	FLASH_SC0	0x0030	ERST_SEL	0x0100	CHIP_ID0
0x0001	SN_DATA1	0x0021	FLASH_SC1	0x0031	ERST_ENB	0x0101	CHIP_ID1
0x0002	SN_DATA2	0x0022	FLASH_SC2	0x0032	XTAL_CFG	0x0102	CHIP_ID2
0x0003	SN_DATA3	0x0023	FLASH_SC3	0x0038	WAIT_TS	0x0103	CHIP_ID3
0x0004	SN_DATA4	-	-	0x0039	BORVS	0x0104	CHIP_ID4
0x0005	SN_DATA5	-	-	0x003E	RVCFG	0x0105	CHIP_ID5
0x0006	SN_DATA6	-	-	0x003F	nRVCFG	0x0106	CHIP_ID6
0x0007	SN_DATA7	-	-	-	-	0x0107	CHIP_ID7
0x0008	ID_DATA0	-	-	-	-	-	-
0x0009	ID_DATA1	-	-	-	-	-	-
0x000A	ID_DATA2	-	-	-	-	-	-
0x000B	ID_DATA3	-	-	-	-	0x0128	rc32m_trim
0x000C	ID_DATA4	-	-	-	-	0x012C	rc24m_trim
0x000D	ID_DATA5	-	-	-	-	-	-
0x000E	ID_DATA6	-	-	-	-	-	-
0x000F	ID_DATA7	-	-	-	-	-	-

HC89S105A will be configured a 8 bytes CHIP_ID before leave the factory, the CHIP_ID is unique and not repeated, user can read it by MOVC instruction in code.

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

Note: 1. User must set register INSCON[IAPS] bit to 1 before read OPTION.
2. First character "n" is complement of the corresponding data.

The program to read CHIP_ID in C is as follows：

```
// Read arbitrary length of data from FLASH
void Flash_ReadArr(unsigned int fui_Address,unsigned char fuc_Length,unsigned char *fucp_SaveArr)
{
    while(fuc_Length--)
    *(fucp_SaveArr++)=*((unsigned char code *)(fui_Address++));
}
// Read CHIP_ID and save it to the read_chip_id array
unsigned char read_chip_id[8];
INSCON |= 0x10;
Flash_ReadArr(0x0100,8,read_chip_id);      //CHIP_ID start address is 0x0100
INSCON &=~ 0x10;
```

To read SN_DATA or ID_DATA, you only need to change the read address.

In addition, users can obtain sn_DATA0-SN_data7, ID_DATA0-ID_data7, and chip_ID0-chip_ID7 by

reading XSFR (see section 3.3.1.2).

3.1.3.1 External reset level selection ERST_SEL

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

Bit	Flag	Introductions
7-1	-	Reserved Bits
0	ERST_SEL	Reset the level select bit 0：High level reset 1：Low level reset

3.1.3.2 External reset enabled ERST_ENB

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

Bit	Flag	Introductions
7-1	-	Reserved Bits
0	ERST_ENB	Reset pin enable bit 0: External $\overline{\text{RST}}$ input 1 : P4.7 as GPIO

3.1.3.3 External crystal configuration register XTAL_CFG

Bit	7	6	5	4	3	2	1	0
Flag								xtal_sel

Bit	Flag	Introductions
7-1	-	Reserved Bits
0	xtal_sel	External crystal select bit 0：Low-frequency vibration crystal 32.768KHz 1：High-frequency vibration crystal

3.1.3.4 Wait time of reread OPTION after reset WAIT_TS

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

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

3.1.3.5 BOR detection voltage selection BORVS

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

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

3.1.3.6 Second reset vector configuration RVCFG

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

Bit	Flag	Introductions
7	RVSEN	The second reset vector enable bit 0: disable the second reset vector 1: enable the second reset vector
6	-	Reserved Bits
5-0	RVADR[5:0]	The second reset vector configuration values The second reset vector address = {RVADR[5:0] ,0000000000B} Note： 1. When RVADR[5:0]=0, the second reset vector address coincide with 0x0000H. 2.RVADR[5:0] configuration data only equal 100000，110000，111000，111100，111110，111111 six values, the second reset vector space only is 1K，2K，4K，8K，16K，32K.

3.1.4 FLASH IAP operation

The FLASH of HC89S105A has 512 sectors, 128 Bytes constitute a sector, 8*128 Bytes = 1K Bytes constitute a page, 1K Bytes*4 = 4KBytes constitute a block.

IAP must be erased before writing The IAP erases one sector at a time (128 bytes), and the address register can be any address in the sector. IAP writes are single-byte writes, one byte at a time.

It takes 5ms to erase a sector by IAP. When CPU_CLK is 16MHz, IAP takes 23μs to write a byte, and when CPU_CLK is 2MHz, IAP takes 37μs to write a byte.

3.1.4.1 IAP operation precautions

HC89S105A user program code can read, erase, write to the FLASH, as the user to update the code or store data for use, in order to ensure the security of the user's FLASH operation, please note during use:

1、 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.

2、 The system does not respond to any interruption during IAP operations.

3、 Setting relevant IAP erasure protection in Option and enabling the protection bit of the sector where the user program is located can effectively ensure that the program area will not be overwritten or erased by mistake.

4、 Before the write operation, it is recommended to turn off the interruption (EA=0) to ensure that the interruption will not be affected during the IAP operation, and will resume the interruption after the IAP write operation is completed.

5、 During IAP operation, it is inevitable that the power may fail before data is written after data erasure. Therefore, you are advised to save data in two areas. Even if data in one area is erased, data in the other area can be read normally.

3.1.4.2 IAP data register IAP_DATA

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

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

3.1.4.3 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]							

IAP_ADDRH

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[15:8]							

Bit	Flag	Introductions
7-0	IAP_ADDR[15:8]	High 8 bits of the IAP operation address register
7-0	IAP_ADDR[7:0]	High 8 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 0xFFFF automatically.

3.1.4.4 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	0	0	0	0	0	0	0	0
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, at the same time automatic locking
```

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&IAP_ADDRH Links to 0xFF, IAP_DATA Links to 0x00, at the same time automatic locking
```

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

3. Software reset (do not reread code options)

```
IAP_CMDH = 0xF0;
IAP_CMDL = 0x0F;
IAP_CMDH = 0x87;
IAP_CMDL = 0x78;
```

4. Software reset (reread code options)

```
IAP_CMDH = 0xF0;
```

```
IAP_CMDL = 0x0F;
IAP_CMDH = 0x78;
IAP_CMDL = 0x87;
```

3.1.5 FLASH ICP operation

3.1.5.1 JTAG mode

User can use HC-LINK 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-LINK user manual.

In addition, because the programming signals are very sensitive, user needs to use 5 jumpers to separate programming Pins (VDD, TDO, TDI, TMS, TCK) from the circuit, as shown in below figure.

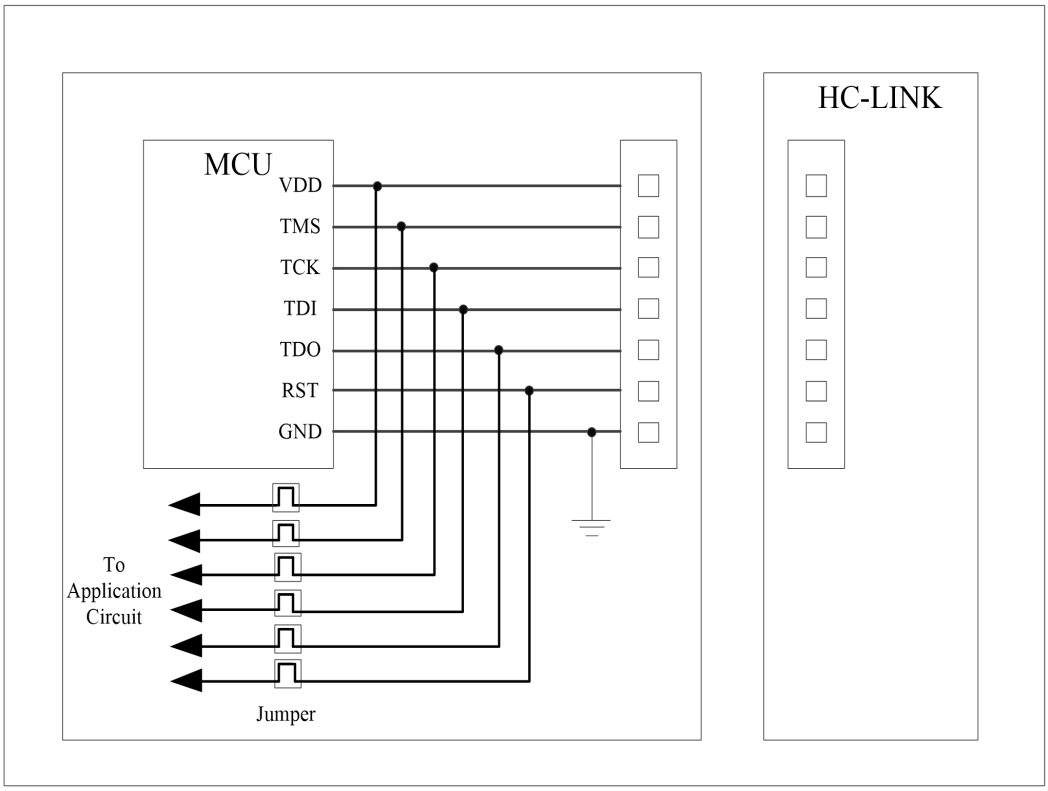


Figure 3- 1 HC-LINK programming hardware connection

3.1.5.2 SWD mode

User can use HC-LINK emulator to program MCU in SWD mode, after MCU is already welded in the user board, if user uses power-on reset, only links 4 cables(VDD，GND，SDA，SCK), 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 5 cables to enter the programming mode, add a reset Pin, detailed instructions of emulator, please see HC-LINK user manual.

In addition, because the programming signals are very sensitive, user needs to use 4 jumpers to separate programming Pins (VDD,SDA,SCK, $\overline{RST}$) from the circuit, as shown in below figure. In addition, if the external reset pin is used to enter, the external reset pin needs to be separated from the jumper.

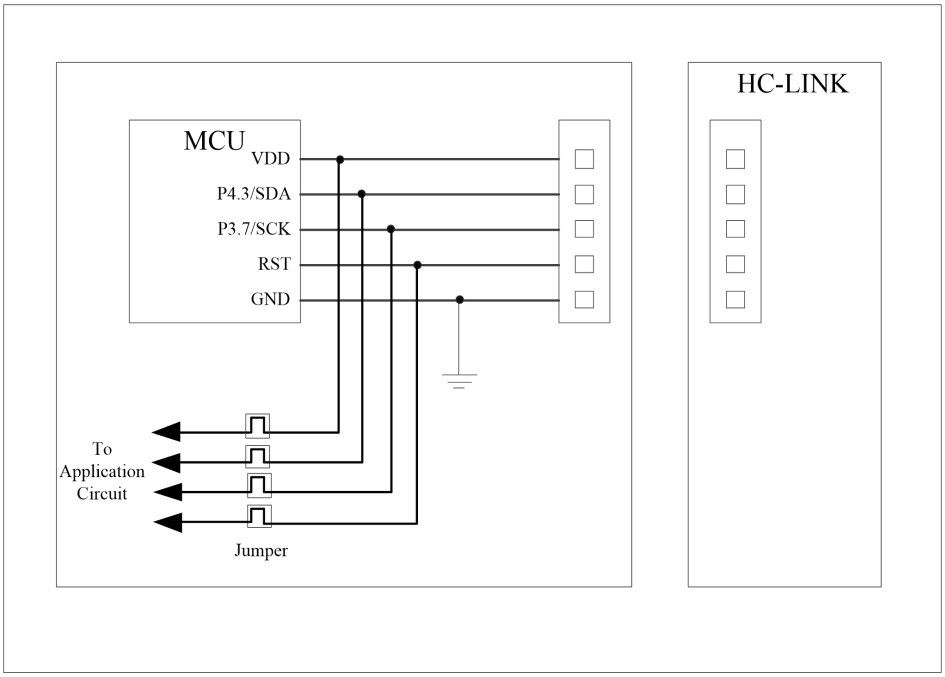


Figure 3- 2 HC-LINK programming hardware connection

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

1. Disconnect jumpers before start programming, separate programming pins from the application circuit.
2. Connect the chip programming pin Flash program interface, then start programming.
3. After the program ends, disconnect flash Programming interfaces, connect jumper to application circuit.

3.1.6 Second reset vector operations

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

3.2 Data storage（RAM）

HC89S105A provide user with a 256 bytes internal RAM and 4K bytes internal expansion RAM as data memory. Below is data memory space allocation.

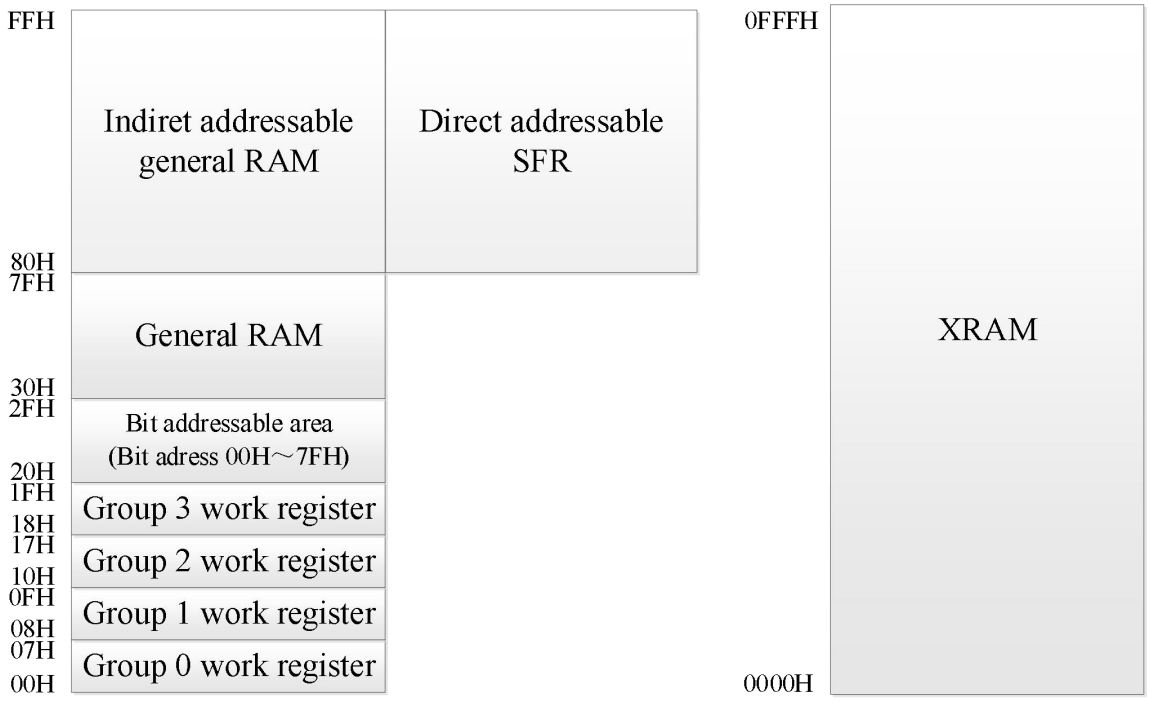


Figure 3- 3 Data memory map

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

Internal expansion RAM (XRAM) addresses range is 0x000～0xFFF 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	PWM2DTL
E8	-	PWMFLT	PWM1C	PWM1PL	PWM1PH	PWM1DL	PWM1DH	PWM1DTL
E0	ACC	PWMEN	PWM0C	PWM0PL	PWM0PH	PWM0DL	PWM0DH	PWM0DTL
D8	-	MCLDIVA0	MCLDIVA1	MCLDIVA2	MCLDIVA3	MCLDIVA4	MCLDIVA5	MCLDIVC
D0	PSW	LCDCON	T5CON	TL5	TH5	RCAP5L	RCAP5H	-
C8	P5	PCACLK	PCAMOD0	PCAMOD1	CCAPL0	CCAPH0	CCAPL1	CCAPH1
C0	P4	PCACON	PCACL	PCACH		PWM0DTH	PWM1DTH	PWM2DTH
B8	IE1	IP2	IP3	LVDC	T6CON	WDTC	CRCL	CRCH
B0	P3	IP4	IE2	LVDCMP	ADCC0	ADCC1	ADCRL	ADCRH
A8	IE	IP0	IP1	SPDAT	SPCTL	SPSTAT	IICDAT	IICADR
A0	P2	T4CON		INSCON	TL4	TH4	IICCON	IICSTA
98	SCON	SBUF	SADDR	SADEN	SBRTL	SBRTH	SCON2	PWM3C
90	P1	T3CON	TL3	TH3	-	-	PINTF0	PINTF1
88	TCON	TMOD	TL0	TL1	TH0	TH1	CLKSWR	CLKCON
80	P0	SP	DPL	DPH	PWM3P	PWM3D	-	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
```

XSFR can be assigned as if it were a direct addressing register by using #define ALLOCATE_EXTERN and #include "HC89S105A. H ", for example:

```
ADCC2 = 0x4D;
```

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	T4CON1	0x0011	CLKDIV	0x0021	-	0x0031	PITS1
0x0002	T5CON1	0x0012	FREQ_CLK	0x0022	CRCC	0x0032	PITS2
0x0003	T5CON2	0x0013	CLKOUT	0x0023	-	0x0033	PITS3
0x0004	PCA_PWM0	0x0014	UART_CLKS	0x0024	BORC	0x0034	-
0x0005	PCA_PWM1	0x0015	SPOV	0x0025	BORDBC	0x0035	-
0x0006	-	0x0016	-	0x0026	-	0x0036	-
0x0007	-	0x0017	PORB_IAPF	0x0027	LVDDBC	0x0037	-
0x0008	S2CON	0x0018		0x0028	-	0x0038	PINTE0
0x0009	S2CON2	0x0019	AWDCON	0x0029	-	0x0039	PINTE1
0x000A	S2BUF	0x001A	ADCC3	0x002A	RSTDBC	0x003A	
0x000B	S2ADDR	0x001B	ADCC2	0x002B		0x003B	
0x000C	S2ADEN	0x001C	PWM0DBC	0x002C	CLKPCKEN0	0x003C	
0x000D	S2BRTH	0x001D	PWM1DBC	0x002D	CLKPCKEN1	0x003D	TRMEN
0x000E	S2BRTL	0x001E	PWM2DBC	0x002E	ADCCONTV	0x003E	TRMV
0x000F	-	0x001F	-	0x002F	ADCGAPV	0x003F	-

Extension XSFR (base address is 0xFFC0)

Offset address	XSFR	Offset address	XSFR	Offset address	XSFR	Offset address	XSFR
0x0000	SCRH0	0x0010	CAPH0	0x0020	PWM0PHASEH	0x0030	PWM0CMPH
0x0001	SCRL0	0x0011	CAPL0	0x0021	PWM0PHASEL	0x0031	PWM0CMPL
0x0002	SCRH1	0x0012	CAPH1	0x0022	PWM1PHASEH	0x0032	-
0x0003	SCRL1	0x0013	CAPL1	0x0023	PWM1PHASEL	0x0033	-
0x0004	SCRH2	0x0014	-	0x0024	PWM2PHASEH	0x0034	-
0x0005	SCRL2	0x0015	-	0x0025	PWM2PHASEL	0x0035	-
0x0006	SCRH3	0x0016	-	0x0026	-	0x0036	-
0x0007	SCRL3	0x0017	-	0x0027	-	0x0037	-
0x0008	SCRH4	0x0018	CAPCON0	0x0028	PWM0INTDIV	0x0038	-
0x0009	SCRL4	0x0019	CAPCON1	0x0029	PWM1INTDIV	0x0039	PWMCON0
0x000A	SCRH5	0x001A	ADCHTRH	0x002A	PWM2INTDIV	0x003A	PWMCON1
0x000B	SCRL5	0x001B	ADCHTRL	0x002B	-	0x003B	PWMCON2
0x000C	SCRH6	0x001C	ADCLTRH	0x002C	-	0x003C	PWMCON3
0x000D	SCRL6	0x001D	ADCLTRL	0x002D	-	0x003D	PWMCON4
0x000E	SCRH7	0x001E	ADCDLYH	0x002E	PWM3CLKS	0x003E	-
0x000F	SCRL7	0x001F	ADCDLYL	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	P4M2	0x0032	-
0x0003	P0M3	0x0013	P2M3	0x0023	P4M3	0x0033	-
0x0004	-	0x0014	-	0x0024	-	0x0034	-
0x0005	-	0x0015	-	0x0025	-	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	-	0x001C	-	0x002C	-	0x003C	-
0x000D	-	0x001D	-	0x002D	-	0x003D	-
0x000E	-	0x001E	-	0x002E	-	0x003E	-
0x000F	-	0x001F	-	0x002F	-	0x003F	-

Extension XSFR (base address is 0XFF40)

Offset address	XSFR	Offset address	XSFR	Offset address	XSFR	Offset address	XSFR
0x0000	P00DBC	0x0010	P0OUT	0x0020	COMP0EN	0x0030	P0DRENH
0x0001	P01DBC	0x0011	P1OUT	0x0021	COMP1EN	0x0031	P0DRENL
0x0002	P02DBC	0x0012	P2OUT	0x0022	COMP2EN	0x0032	P1DRENH
0x0003	-	0x0013	P3OUT	0x0023	COMP3EN	0x0033	P1DRENL
0x0004	-	0x0014	P4OUT	0x0024	COMP4EN	0x0034	P2DRENH
0x0005	-	0x0015	P5OUT	0x0025	COMP5EN	0x0035	P2DRENL
0x0006	-	0x0016	-	0x0026	-	0x0036	P3DRENH
0x0007	-	0x0017	-	0x0027	-	0x0037	P3DRENL
0x0008	-	0x0018	-	0x0028	SEGP0EN	0x0038	P4DRENH
0x0009	-	0x0019	-	0x0029	SEGP1EN	0x0039	P4DRENL
0x000A	-	0x001A	-	0x002A	SEGP2EN	0x003A	P5DRENH
0x000B	-	0x001B	-	0x002B	SEGP3EN	0x003B	P5DRENL
0x000C	-	0x001C	-	0x002C	SEGP4EN	0x003C	PDREN_SEL
0x000D	-	0x001D	-	0x002D	SEGP5EN	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	T3_MAP	0x0012	FLT0_MAP	0x0022	SCL_MAP	0x0032	-
0x0003	T4_MAP	0x0013	-	0x0023	SDA_MAP	0x0033	-
0x0004	T6_MAP	0x0014	PWM1_MAP	0x0024	$\overline{SS}$_MAP	0x0034	-
0x0005	BRTO_MAP	0x0015	PWM11_MAP	0x0025	SCK_MAP	0x0035	-
0x0006	T5_MAP	0x0016	FLT1_MAP	0x0026	MOSI_MAP	0x0036	-
0x0007	-	0x0017	-	0x0027	MISO_MAP	0x0037	-
0x0008	CAP0_MAP	0x0018	PWM2_MAP	0x0028	TXD2_MAP	0x0038	-
0x0009	CAP1_MAP	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	PWM3_MAP	0x002C	-	0x003C	-
0x000D	ADCST_MAP	0x001D	-	0x002D	-	0x003D	-
0x000E	-	0x001E	-	0x002E	-	0x003E	-
0x000F	CLKO_MAP	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	-

HC89S105A will be solidified with a CHIP_ID, which is 8 bytes in total and one ID for each chip. It will not be repeated. Users can read it directly in the program just like reading XSFR.

SN_DATA and ID_DATA are user-defined data, which can be erased and modified by tools like setting code options. Users can also read XSFR directly in the program.

4 The system clock

4.1 Characteristics of the system clock

HC89S105A MCU system clock have 3 optional clock sources:
1. external high-frequency RC clock (4MHz~16MHz),
2. external low-frequency RC clock (32.768KHz),
3. internal high-frequency RC clock (32 MHz)

The internal low-frequency RC clock (44KHz) can only be used as a WDT clock, not as a system clock. Fosc can perform frequency division with any value between 1 and 255. The clock after frequency division is recorded as the CPU clock, that is, Fcpu. And the clock of other peripherals as the peripheral clock, that is Fper.

After the chip is powered on and reset, the internal high-frequency RC is selected as the system clock by default, and its Fosc is 32MHz and Fcpu is 2MHz. The frequency of CPU_CLK can be changed by configuring the frequency division register.

CPU can run under 24MHz highest frequency, if frequency of clock selected is higher than 24MHz, the clock need to be divided to meet CPU Clock equal to or less than 24MHz .

You can switch RC32M to RC24M by configuring the TRMV register. For details, see theTRMV Register.

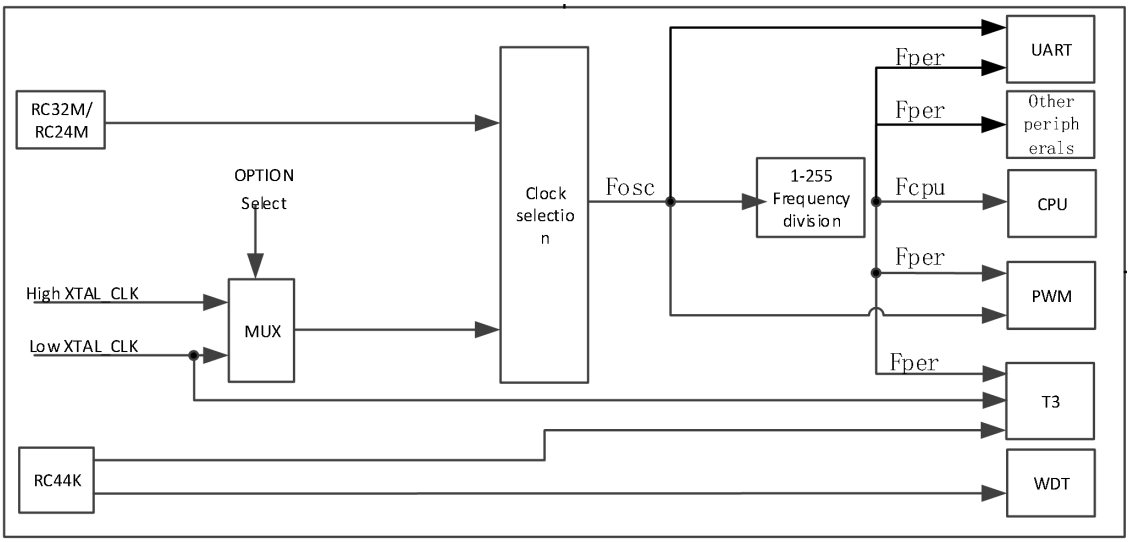


Figure 4- 1 System clock block diagram

4.2 System clock registers

4.2.1 Clock control register CLKCON

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R	R/W	R/W	R
Reset values	0	0	1	1	0	0	1	0
Flag	HXTAL RDY	LXTAL RDY	HSRC RDY	LSRC RDY	-	XTALEN	HSRCEN	-

Bit	Flag	Introductions
7	HXTALRDY	External high-frequency crystal oscillator state bit 0：External high-frequency crystal oscillator is not ready 1：External high-frequency crystal oscillator is ready Note: The hardware automatically clear 0 or set 1
6	LXTALRDY	External low-frequency crystal oscillator state bit 0：External low-frequency crystal oscillator is not ready 1：External low-frequency crystal oscillator 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 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: When enabling the function, the software needs to set the IO mode of the corresponding pin as analog channel, and select low-frequency crystal oscillator by default. If you want to use high-frequency crystal oscillator, you need to configure the code option.
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	R/W	R	R	R	R
Reset values	0	0	0	0	0	0	0	0
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 01：Current system clock is external low-frequency crystal oscillator 11：Current system clock is external high-frequency crystal oscillator system automatically switches state based on current system clock
5	-	Reserved Bit
4	CLKSEL	System clock selection bit 0：Select system clock to internal high-frequency RC 1：system clock is external crystal oscillator Note： 1. When the system clock is selected, the corresponding clock source status bit in the CLKCON register must be 1; otherwise, the previous clock will continue. 2. Before switching the system clock, you are advised to enable the clock to be switched and check whether the clock source status bit is 1. Configure the clock source status bit to complete the clock switch.
3-0	-	Reserved Bits

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	1	0	0	0	0
Flag	CLKDIV[7:0]							

Bit	Flag	Introductions
7-0	CLKDIV[7:0]	CPU clock division factor, default value is 2 Configuration values is 0 or 1, clock is not divided; in other condition, the configuration value is equal to the frequency factor; Note: The divided clock is recorded as Fcpu for CPU module and Fper for other peripherals.

4.2.4 Clock output register CLKOUT

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

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

4.2.5 Peripheral clock gating register CLKPCKEN0，CLKPCKEN1

Power consumption can be reduced by turning off clocks that are not used by peripherals. The clock gating register of the peripherals enables the user to turn on or off the system clock connection to the peripherals at any time in operation mode. When the user turns off the clock on a peripheral, the module is disabled and all registers that operate it are disabled.

After the system is reset, all peripheral clocks are on. You can disable 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	T6_CLKEN	PWM_CLKEN	PCA_CLKEN	T1_CLKEN	T0_CLKEN

Bit	Flag	Introductions
7	UART2_CLKEN	UART2 clock enable bit 0：Peripheral clock connection to UART2 is forbidden 1：Enable the peripheral clock to connect to UART2
6	UART1_CLKEN	UART1 clock enable bit 0：Peripheral clock connection to UART1 is forbidden 1：Enable the peripheral clock to connect to UART1
5	WDT_CLKEN	WDT clock enable bit 0：Peripheral clock connection to WDT is forbidden 1：Enable the peripheral clock to connect to WDT
4	T6_CLKEN	T6 clock enable bit 0：Peripheral clock connection to T6 is forbidden 1：Enable the peripheral clock to connect to T6
3	PWM_CLKEN	PWM clock enable bit 0：Peripheral clock connection to PWM is forbidden 1：Enable the peripheral clock to connect to PWM
2	PCA_CLKEN	PCA clock enable bit 0：Peripheral clock connection to PCA is forbidden 1：Enable the peripheral clock to connect to PCA
1	T1_CLKEN	Timer 1 clock enable bit 0：Peripheral clock connection to Timer 1 is forbidden 1：Enable the peripheral clock to connect to Timer 1
0	T0_CLKEN	Timer 0 clock enable bit 0：Peripheral clock connection to Timer 0 is forbidden 1：Enable the peripheral clock to connect to Timer 0

CLKPCKEN1

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	1	1	1	1	1
Flag	-			T5_ CLKEN	T4_ CLKEN	T3_ CLKEN	SPI_ CLKEN	IIC_ CLKEN

Bit	Flag	Introductions
7-5	-	Reserved Bits
4	T5_CLKEN	Timer 5 clock enable bit 0：Peripheral clock connection to Timer 5 is forbidden 1：Enable the peripheral clock to connect to Timer 5
3	T4_CLKEN	Timer 4 clock enable bit 0：Peripheral clock connection to Timer 4 is forbidden 1：Enable the peripheral clock to connect to Timer 4
2	T3_CLKEN	Timer 3 clock enable bit 0：Peripheral clock connection to Timer 3 is forbidden 1：Enable the peripheral clock to connect to Timer 3
1	SPI_CLKEN	SPI clock enable bit 0：Peripheral clock connection to SPI is forbidden 1：Enable the peripheral clock to connect to SPI
0	IIC_CLKEN	IIC clock enable bit 0：Peripheral clock connection to IIC is forbidden 1：Enable the peripheral clock to connect to IIC

4.2.6 Serial clock selection register UART_CLKS

Through this register can choose RC32M makes the serial clock, CPU frequency can be obtained by dividing RC32M by 2.So the serial port can use 115200 baud rate.

UART_CLKS

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	-						UART2_CLKS	UART1_CLKS

Bit	Flag	Introductions
7-2	-	Reserved Bits
1	UART2_CLKS	UART2 clock select bit 0：The clock of UART2 is Fper 1：The clock of UART2 is Fosc The selected clock is recorded as Fuart
0	UART1_CLKS	UART1 clock select bit 0：The clock of UART1 is Fper 1：The clock of UART1 is Fosc The selected clock is recorded as Fuart

4.2.7 CPU clock frequency register FREQ_CLK

Before performing a flash IAP wipe write or the system entering stop mode, it is necessary to configure FREQ_CLK register in the extended SFR to indicate the frequency of the current CPU clock, FREQ_CLK register configuration value is equal to the frequency value of the CPU clock, the minimum is 1MHz. If the current CPU is operating at 16MHz, configure registers FREQ_CLK = 0x10. It is recommended to divide the system clock frequency into integers before IAP is erased. When the system clock frequency is lower than 1MHz, flash IAP erase operations cannot be performed.

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 Examples are as follows: If the CPU frequency is 24MHz, the configuration value is 0x18 If the CPU frequency is 16MHz, the configuration value is 0x10 If the CPU frequency is 8MHz, the configuration value is 0x08 If the CPU frequency is 4MHz, the configuration value to 0x04 If the CPU frequency is 2MHz, the configuration value is 0x02

		If the CPU frequency is less than or equal to 1MHz, the configuration value is 0x01

4.2.8 Internal high-frequency RC adjustment enable register TRMEN

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	-							RCTRMEN

Bit	Flag	Introductions
7-1	-	Reserved Bits
0	RCTRMEN	Internal high-frequency RC adjustment enable bit 1：enable internal high-frequency RC adjustment 0：Disable internal high-frequency RC adjustment Note： After this register is enabled, the TRMV register must be configured immediately, otherwise the enabled register will be cleared after the next instruction, and the internal high frequency RC adjustment will be invalid.

4.2.9 Internal high-frequency RC adjustment configuration register TRMV

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	x	x	x	x	x	x	x
Flag	RC24M_SEL	RCTRMV						

Bit	Flag	Introductions
7	RC24M_SEL	0：RC32M 1：RC24M Note：If the CPU works at 24MHz, you need to switch RC32M to RC24M so that both the peripheral clock and CPU clock work at 24MHz
6-0	RCTRMV	Internal high-frequency RC adjustment configuration value Note： 1. "x" is undefined value, and the power-on reset value of this register is the factory calibration value. 2.When configuring this register value, the internal high-frequency RC adjustment enable bit needs to be set to 1 first. 3. According to the calibration curve software, first to enable RCTRMEN, followed by the configuration of RCTRMV, after the adjustment of RCTRMEN automatically clear, to prevent repeated operation

(1) The code to switch RC32M to RC24M is as follows:

```
unsigned char code *hs24m_trim = 0x012c;
INSCON = 0X10;
TRMEN = 0x01;
TRMV = *hs24m_trim ^ 0x7f;
INSCON = 0X00;
```

(2) The code to switch RC24M to RC32M is as follows:：

```
unsigned char code *hs32m_trim = 0x0128;
INSCON = 0X10;
TRMEN = 0x01;
TRMV = *hs32m_trim ^ 0x7f;
INSCON = 0X00;
```

5 Power management

5.1 Power management characteristics

- Provide idle mode (IDLE) and STOP mode , as a power saving mode
- Provide a variety of ways to wake up from the IDLE/STOP 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 HC89S105A 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 HC89S105A 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 HC89S105A 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 STOP mode

HC89S105A will enter very low power consumption state in STOP mode. In STOP mode CPU and peripherals of all clock signal will stop, but if WDT and TIMER3 enabled and permits working in STOP mode, then the WDT and TIMER3 module will continue to work. Before enter the STOP mode all the CPU states were saved, such as the PC, PSW, SFR, RAM and so on.

Before the system goes intoSTOP mode, 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, The user must set register FREQ_CLK to 0x10.

Set PCON register PD bit to 1, HC89S105A will enter the STOP mode. PD set 1 is the last instruction executed by CPU before enter the STOP mode.

Note: If user set IDL and PD bits at the same time, HC89S105A enter the STOP mode. After exit the STOP mode, CPU couldn’t enter IDLE mode, and hardware will clear the IDL and the PD bits after exit from the STOP mode.

Multiple ways to exit the STOP mode:

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

(2) The reset signal (valid level on external reset Pin, WDT reset, BOR reset or low voltage detection

reset on external ports). Valid reset signal will reset PCON register PD bit to 0, oscillator restart, CPU clock and the peripheral clock immediately recovered, system program will start to run from the reset address 0000H, RAM remains unchanged, SFR value changes depend on the value of different function module.

5.4 Power management registers

5.4.1 Power control register PCON

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

Bit	Flag	Introductions
7-4	-	Reserved (read = 0b,, write invalid)
3	GF1	User normal flag 1
2	GF0	User normal flag 0
1	PD	STOP mode control bit 0 : Normal mode 1 : Enter STOP 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 STOP 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 HC89S105A 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 VBOR, and continue time is more than TBOR, 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 HC89S105A BOR voltage detection value by code option or register. BOR gear: 4.2V/3.9V/3.6V/3.0V/2.6V/2.4V/2.0V/1.8V.

Undervoltage reset diagram shown below, TBOR configuration by register used to voltage debouncing.

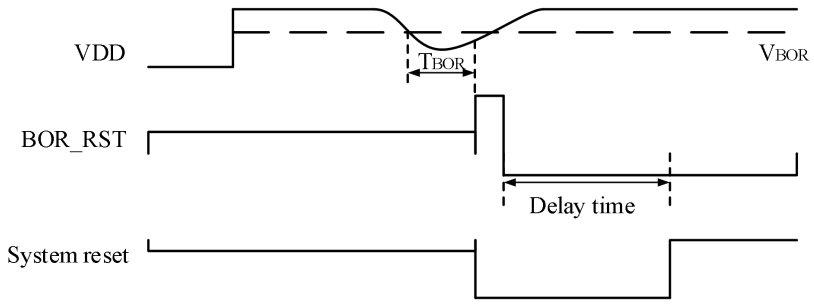


Figure 6- 1 BOR schematic diagram

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. The effective level for external reset can also be set by code option, and the chip factory defaults to high level reset.

When it as RST port, after $\overline{RST}$ Pin need be set high level and keep the setting time at least (software

configuration), microcontroller will enter the reset state, after set $\overline{RST}$ Pin back to the low 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 of HC89S105AC8 and HC89S105AS8 cannot be used as general-purpose input/output when as external reset $\overline{RST}$ port. P4.1 ports of HC89S105AK8 cannot be used as general-purpose input/output when as external reset $\overline{RST}$ port.

6.5 External port low-voltage detection reset

When external voltage is too low, it cannot guarantee the normal MCU working. At this time, user can use the external port low voltage detection (PLVD) to reset the microcontroller, external port detection voltage equal 1.2V, the reset function can be disabled. When PLVD reset, PLVRSTF bit in RSTFR register is set to 1, and user can read the flag to determine whether the external ports low-voltage detection reset or not. In addition, by setting registers user can also implement external port voltage debouncing.

6.6 Software reset

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

6.7 Watchdog (WDT) reset

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

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

6.8 Stack overflow reset

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

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

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

6.9 Reset registers

6.9.1 Reset flag register RSTFR

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

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

Bit	Flag	Introductions
7	PORF	Power-on reset flag 0 : No power-on reset 1 : Power-on reset generated, software clear 0
6	EXRSTF	External RST reset flag 0 :No external RST reset 1 : External RST reset generated, software clear 0
5	BORF	Under voltage reset flag 0 : No undervoltage reset 1 : Undervoltage reset generated, software clear 0
4	WDTRF	WDT Reset flag 0 : No WDT reset 1 : WDT reset generated, software clear 0
3	SWRF	Software Reset flag 0 : No software reset 1 : Software reset generated , software clear 0
2	-	Reserved
1	SPOVF	Stack overflow flag 0 : No stack overflow reset 1 : Stack overflow reset generated, software clear 0
0	PLVRSTF	External port voltage detection reset flag 0 : External port voltage detection reset 1 : External port voltage detection reset generated, software clear0

6.9.2 BOR voltage detection control register BORC

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R	R	R/W	R/W	R/W
Reset values	1	0	0	0	0	0	0	0
Flag	BOREN	BOR_DBC_EN	BOR_PD_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	BOR_PD_EN	BOR power enable bit 0 : Disabled 1 : Enabled Note: After this bit is enabled, the chip enters the STOP mode when BOR is reset
4-3	-	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 STOP mode BOR debouncing is turn off automatically, open it automatically when exit STOP 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 STOP mode, opens automatically after exit the STOP mode.

6.9.5 Stack overflow reset enable register SPOV_RSTEN

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

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

6.9.6 PORB_IAPF register

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	1
Flag	-							PORB_IAP

Bit	Flag	Introductions
7-1	-	Reserved Bits
0	PORB_IAP	PORB_IAP flag bit When this bit is 1, the voltage is higher than 1.6V, and the power-on reset is complete

7 General and multiplexed I/O

7.1 General and multiplexed I/O characteristics

- Provides 46/42/30 bi-directional I/O ports
- Multiple modes configuration

7.2 I/O mode

HC89S105A all I/O ports can be configured by software into one of a variety of work types, specifically:input, pull-up input, pull-down input,analog input,strong push pull output,open drain output,open drain output with pull-up, and the input can be configured as Schmidt input.

If P4.7 of 105AC8 and 105AS8 or P4.1 of 105AK8 is configured as a reset pin and low level reset is selected, the port mode is Schmidt input pull-up. If high level reset is selected, the port mode is Schmidt input pull down.

When HC89S105A 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.

The HC89S105A adds a set of read-only registers P0OUT, P1OUT, P2OUT, P3OUT, P4OUT, P5OUT. In output mode, you can read this set of registers directly to get the values written to the port data registers.

HC89S105A will first read the contents of the register to be modified back to ALU, modify the corresponding bits, and then write the whole back to the original register address. The instruction to complete this function is called "read - modify - write" instruction.

"Read - modify - write" instruction is the internal execution of the MCU, it occurs when writing IO port, when writing IO port, it first read the current state of IO back, according to the data to be written to modify the read back data, and then write IO port; The read pin is the current state of the direct read pin. If the current pin is high level, the read back is high level, and when the current pin is low, the read back is low level.

The read-modify-write command includes the following commands: INC direct, DEC direct, ANL direct,A, ANL direct, #data, ORL direct,A, ORL direct, Data, DJNZ direct,rel, MOV bit, C, CLR bit, SETB bit, CPL bit, JBC bit,rel See the instruction list in section 23.

7.3 I/O function block diagram

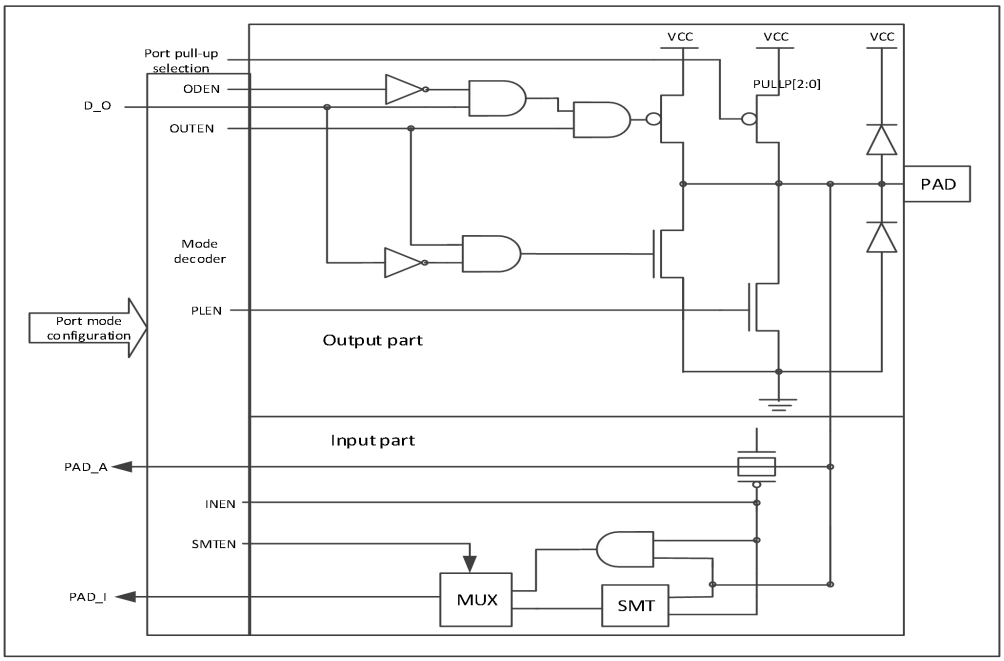


Figure 7- 1 I/O function block diagram

7.4 I/O port registers

7.4.1 P0 port data register P0

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

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

7.4.2 P1 port data register P1

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

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

7.4.3 P2 port data register P2

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

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

7.4.4 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	P3[7:0]							

Bit	Flag	Introductions
7-0	P3[7:0]	P3 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	P4[7:0]							

Bit	Flag	Introductions
7-0	P4[7:0]	P4 port data register

7.4.6 P5 port data register P5

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	0	0	0	0	0	0
Flag	-	-	P5[5:0]					

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

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 1000:Push-pull output (Sourcing Current:10mA/ sink current:28mA,PDREN_SEL=0) 1001:OutputOD (Sourcing Current:10mA/ sink current:28mA,PDREN_SEL=0) 1010: Open Drain Output with pull-up (Sourcing Current: 10mA/ sink current: 28mA, PDREN_SEL=0) 1011: Reserved 1100:Push-pull output (Sourcing Current:4mA/ sink current:7mA,PDREN_SEL=0) 1101: OutputOD (Sourcing Current：4mA/ sink current：7mA,PDREN_SEL=0) 1110:Open Drain Output with pull-up (Sourcing Current:4mA/ sink current:7mA,PDREN_SEL=0) 1111: Reserved

7.4.8 P0 port driver select register P0DRENL，P0DRENH

P0DRENL

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	P03D[1:0]		P02D[1:0]		P01D[1:0]		P00D[1:0]	

P0DRENH

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	P07D[1:0]		P06D[1:0]		P05D[1:0]		P04D[1:0]	

Bit	Flag	Introductions
7-6 5-4 3-2 1-0	P0xD[1:0] (x = 0...7)	P0.x port driver mode configuration bit 00： port driver 1（Sourcing Current: 4mA/ sink current:7mA） 01： port driver 2（Sourcing Current: 7mA/ sink current:14mA） 10： port driver 3（Sourcing Current: 10mA/ sink current:28mA） 11： port driver 4（Sourcing Current: 20mA/ sink current:70mA）

7.4.9 P1 port function select register P1M0，P1M1，P1M2，P1M3

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]				P16M[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 1000:Push-pull output (Sourcing Current:10mA/ sink current:28mA,PDREN_SEL=0) 1001:OutputOD (Sourcing Current:10mA/ sink current:28mA, PDREN_SEL=0) 1010: Open Drain Output with pull-up (Sourcing Current: 10mA/ sink current: 28mA, PDREN_SEL=0) 1011: Reserved 1100:Push-pull output (Sourcing Current:4mA/ sink current:7mA,PDREN_SEL=0) 1101: OutputOD (Sourcing Current：4mA/ sink current：7mA,PDREN_SEL=0) 1110:Open Drain Output with pull-up (Sourcing Current:4mA/ sink current:7mA,PDREN_SEL=0) 1111: Reserved

7.4.10 P1 port driver select register P1DRENL，P1DRENH

P1DRENL

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	P13D[1:0]		P12D[1:0]		P11D[1:0]		P10D[1:0]	

P1DRENH

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	P17D[1:0]		P16D[1:0]		P15D[1:0]		P14D[1:0]	

Bit	Flag	Introductions
7-6 5-4 3-2 1-0	P1xD[1:0] (x = 0...7)	P1.x port driver mode configuration bit 00： port driver 1（Sourcing Current: 4mA/ sink current:7mA） 01： port driver 2（Sourcing Current: 7mA/ sink current:14mA） 10： port driver 3（Sourcing Current: 10mA/ sink current:28mA） 11： port driver 4（Sourcing Current: 20mA/ sink current:70mA）

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

P2M0

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

P2M1

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

P2M2

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

P2M3

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

Bit	Flag	Introductions
7-4 3-0	P2xM[3:0] (x = 0...7)	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 1000:Push-pull output (Sourcing Current:10mA/ sink current:28mA,PDREN_SEL=0) 1001:OutputOD (Sourcing Current:10mA/ sink current:28mA,PDREN_SEL=0) 1010: Open Drain Output with pull-up (Sourcing Current: 10mA/ sink current: 28mA, PDREN_SEL=0) 1011: Reserved 1100:Push-pull output (Sourcing Current:4mA/ sink current:7mA,PDREN_SEL=0) 1101: OutputOD (Sourcing Current：4mA/ sink current：7mA,PDREN_SEL=0) 1110:Open Drain Output with pull-up (Sourcing Current:4mA/ sink current:7mA,PDREN_SEL=0) 1111: Reserved

7.4.12 P2 port driver select register P2DRENL，P2DRENH

P2DRENL

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	P23D[1:0]		P22D[1:0]		P21D[1:0]		P20D[1:0]	

P2DRENH

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	P27D[1:0]		P26D[1:0]		P25D[1:0]		P24D[1:0]	

Bit	Flag	Introductions
7-6 5-4 3-2 1-0	P2xD[1:0] (x = 0...7)	P2.x port driver mode configuration bit 00： port driver 1（Sourcing Current: 4mA/ sink current:7mA） 01： port driver 2（Sourcing Current: 7mA/ sink current:14mA） 10： port driver 3（Sourcing Current: 10mA/ sink current:28mA） 11： port driver 4（Sourcing Current: 20mA/ sink current:70mA）

7.4.13 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.xport 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 1000:Push-pull output (Sourcing Current:20mA/ sink current:70mA,PDREN_SEL=0) 1001:OutputOD (Sourcing Current:20mA/ sink current:70mA,PDREN_SEL=0) 1010: Open Drain Output with pull-up (Sourcing Current: 20mA/ sink current: 70mA, PDREN_SEL=0) 1011: Reserved 1100:Push-pull output (Sourcing Current:10mA/ sink current:28mA,PDREN_SEL=0) 1101: OutputOD (Sourcing Current：10mA/ sink current：28mA,PDREN_SEL=0) 1110:Open Drain Output with pull-up (Sourcing Current:10mA/ sink current:28mA,PDREN_SEL=0) 1111: Reserved

7.4.14 P3 port driver select register P3DRENL，P3DRENH

P3DRENL

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	P33D[1:0]		P32D[1:0]		P31D[1:0]		P30D[1:0]	

P3DRENH

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	P37D[1:0]		P36D[1:0]		P35D[1:0]		P34D[1:0]	

Bit	Flag	Introductions
7-6 5-4 3-2 1-0	P3xD[1:0] (x = 0...7)	P3.x port driver mode configuration bit 00： port driver 1（Sourcing Current: 4mA/ sink current:7mA） 01： port driver 2（Sourcing Current: 7mA/ sink current:14mA） 10： port driver 3（Sourcing Current: 10mA/ sink current:28mA） 11： port driver 4（Sourcing Current: 20mA/ sink current:70mA）

7.4.15 P4 port function select register P4M0，P4M1，P4M2，P4M3

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	P41M[3:0]				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]			

P4M2

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	P45M[3:0]				P44M[3:0]			

P4M3

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	P47M[3:0]				P46M[3:0]			

Bit	Flag	Introductions
7-4 3-0	P4xM[3:0] (x = 0...7)	P4.xport 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 1000:Push-pull output (Sourcing Current:10mA/ sink current:28mA,PDREN_SEL=0) 1001:OutputOD (Sourcing Current:10mA/ sink current:28mA,PDREN_SEL=0) 1010: Open Drain Output with pull-up (Sourcing Current: 10mA/ sink current: 28mA, PDREN_SEL=0) 1011: Reserved 1100:Push-pull output (Sourcing Current:4mA/ sink current:7mA,PDREN_SEL=0) 1101: OutputOD (Sourcing Current：4mA/ sink current：7mA,PDREN_SEL=0) 1110:Open Drain Output with pull-up (Sourcing Current:4mA/ sink current:7mA,PDREN_SEL=0) 1111: Reserved

7.4.16 P4 port driver select register P4DRENL，P4DRENH

P4DRENL

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	P43D[1:0]		P42D[1:0]		P41D[1:0]		P40D[1:0]	

P4DRENH

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	P47D[1:0]		P46D[1:0]		P45D[1:0]		P44D[1:0]	

Bit	Flag	Introductions
7-6 5-4 3-2 1-0	P4xD[1:0] (x = 0...7)	P4.x port driver mode configuration bit 00：port driver 1（Sourcing Current: 4mA/ sink current:7mA） 01：port driver 2（Sourcing Current: 7mA/ sink current:14mA） 10：port driver 3（Sourcing Current: 10mA/ sink current:28mA） 11：port driver 4（Sourcing Current: 20mA/ sink current:70mA）

7.4.17 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.xport 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 1000:Push-pull output (Sourcing Current:10mA/ sink current:28mA,PDREN_SEL=0) 1001:OutputOD (Sourcing Current:10mA/ sink current:28mA,PDREN_SEL=0) 1010: Open Drain Output with pull-up (Sourcing Current: 10mA/ sink current: 28mA, PDREN_SEL=0) 1011: Reserved 1100:Push-pull output (Sourcing Current:4mA/ sink current:7mA,PDREN_SEL=0) 1101: OutputOD (Sourcing Current：4mA/ sink current：7mA,PDREN_SEL=0) 1110:Open Drain Output with pull-up (Sourcing Current:4mA/ sink current:7mA,PDREN_SEL=0) 1111: Reserved

7.4.18 P5 port driver select register P5DRENL，P5DRENH

P5DRENL

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	P53D[1:0]		P52D[1:0]		P51D[1:0]		P50D[1:0]	

P5DRENH

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	-		-		P55D[1:0]		P54D[1:0]	

Bit	Flag	Introductions
7-6 5-4 3-2 1-0	P5xD[1:0] (x = 0...5)	P5.x port driver mode configuration bit 00： port driver 1（Sourcing Current: 4mA/ sink current:7mA） 01： port driver 2（Sourcing Current: 7mA/ sink current:14mA） 10： port driver 3（Sourcing Current: 10mA/ sink current:28mA） 11： port driver 4（Sourcing Current: 20mA/ sink current:70mA）

7.4.19 port driver select register PDREN_SEL

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	-	-	-	-	-	-	-	PDREN_SEL

Bit	Flag	Introductions
7-1	-	Reserved Bits
0	PDREN_SEL	port driver select configuration bit 0： Use PxMx register controls I/O driver capability 1： Use PxDRENH ， PxDRENL register controls I/O driver capability

7.4.20 Ports debouncing control register P00DBC，P01DBC，P02DBC

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

Bit	Flag	Introductions
7-6	P0xDBCLK [1:0]	Port debouncing clock select 00：F_{osc} /1 01：F_{osc} /4 10：F_{osc} /16 11：F_{osc} /64 Note：x is 0，1 or 2.
5-0	P0xDBCT [5:0]	Port debouncing count number of clock, when configured as a 00, no debouncing. Time debouncing time is the time need to maintain for the level of its corresponding port when port input, in need of attention, assigned to the function foot of the three Pins, external interrupt input, fault detection Pin is affected by debouncing control, and P02DBC[7:0] is P0.2 debouncing control registers. Note: P0xDBCT [5:0] configuration for debouncing time is a range, scale factor * T_{osc}* P0xDBCT [5:0] - T_{osc} < debouncing time < scale factor * T_{osc}* (P0xDBCT [5:0] +1) - T_{osc}.

7.5 Peripheral function Ports total mapping control

7.5.1 Peripheral function Ports total mapping control register

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	T3_MAP	0xFF92	FLT0_MAP	0xFFA2	SCL_MAP	0xFFB2	-
0xFF83	T4_MAP	0xFF93	-	0xFFA3	SDA_MAP	0xFFB3	-
0xFF84	T6_MAP	0xFF94	PWM1_MAP	0xFFA4	$\overline{SS}$_MAP	0xFFB4	-
0xFF85	BRTO_MAP	0xFF95	PWM11_MAP	0xFFA5	SCK_MAP	0xFFB5	-
0xFF86	T5_MAP	0xFF96	FLT1_MAP	0xFFA6	MOSI_MAP	0xFFB6	-
0xFF87	-	0xFF97	-	0xFFA7	MISO_MAP	0xFFB7	-
0xFF88	CAP0_MAP	0xFF98	PWM2_MAP	0xFFA8	TXD2_MAP	0xFFB8	-
0xFF89	CAP1_MAP	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	PWM3_MAP	0xFFAC	-	0xFFBC	-
0xFF8D	ADCST_MAP	0xFF9D	-	0xFFAD	-	0xFFBD	-
0xFF8E	-	0xFF9E	-	0xFFAE	-	0xFFBE	-
0xFF8F	CLKO_MAP	0xFF9F	-	0xFFAF	-	0xFFBF	-

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

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

Bit	Flag	Introductions
7-6	-	Reserved Bits
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 0x3F, 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:

```
TXD_MAP = 0x19;      //TXD-->P3.1
RXD_MAP = 0x1A;      //RXD-->P3.2
```

User need to map UART1 TXD RXD to P2.4 P2.5 pin in the next design, the user must do the following configuration:

```
TXD_MAP = 0x14;      //TXD-->P2.4
RXD_MAP = 0x15;      //RXD-->P2.5
```

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	T6
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
22	T4
23	T5
24	PWM3
25	CLKO

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

When all the port-mapped control registers is not equal 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:

If the lower six bits of T0_MAP are set to 010011, P2.3 is selected as the input port of T0, and the lower six bits of FLT0_MAP are also set to 010011. In this way, signals from port P2.3 also affect FLT0.When

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

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

8 Interrupt

8.1 Interrupt characteristics

- 20 interrupt sources
- 4 level interrupt priorities
- 16 external interrupts

8.2 Interrupt summary

中断源	向量地址	允许位	标志位	查询优先级	中断号（C 语言）
INT0	0003H	EX0	INT0F	1(最高)	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	WDTF	6	5
LVD	0033H	LVDIE	LVDF	7	6
UART2	003BH	ES2	TI/RI	8	7
SPI	0043H	ESPI	SPIF/Pattern	9	8
IIC	004BH	EIIC	SI	10	9
PCA	0053H	ECF	CCF1/ CCF0	11	10
PWM	005BH	PWMxIE (x =0/1/2)	PWMxIF (x =0/1/2)	12	11
T6	0063H	ET6	TF6	13	12
T3	006BH	ET3	TF3	14	13
ADC	0073H	EADC	ADCIF/ AWDIF	15	14
INT2-INT7	007BH	EINTx (x =2...7)	INTxF (x = 2...7)	16	15
INT8-INT15	0083H	EINTx (x =8...15)	INTxF (x = 8...15)	17	16
T4	008BH	ET4	TF4	18	17
T5	0093H	ET5	TF5/ EXF5/ CAPF1/ CAPF0	19	18
FLT	009BH	FLT_INT_ EN	PWMxFLTS (x =0...2)	20	19

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)	Priority
Px[1:0]	
00	Priority 0 (lowest)
01	Priority 1
10	Priority 2
11	Priority 3 (highest)

8.5 Interrupt handling

Interrupt flags are sampled on the rising edge of the CPU clock. If a flag is set to up, then the CPU catches the interrupt and the system calls a long transfer instruction (LCALL) which interrupts the service, but the hardware-generated LCALL is blocked by any of the following conditions:

1. Interrupts of the same level or higher priority are running.

2. The current period is not the last period of the executing instruction. In other words, any interrupt request will not be answered until the executing instruction completes.

3. An instruction is being executed to RETI or access the special registers IE/IE1 or IP0/IP1/IP2/IP3/IP4. In other words, an interrupt request is not answered immediately after RETI or IE/IE1 or IP0/IP1/IP2/IP3/IP4, but only after at least one other instruction is executed.

The interrupt service routine ISR performs some operations corresponding to this interrupt. The ISR ends with the RETI (interrupt return) instruction, which takes the PC value back from the stack and restores the original interrupt setting, after which execution continues from the breakpoint of the main program.

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.

The entry address of the interrupt service program (interrupt vector) detail information, user can refer the interrupt summary.

Note that the RET directive cannot be used instead of the RETI directive. Although the RET instruction can also control the PC to return to the original interrupt place, the RET instruction does not have the function of clearing the interrupt priority status trigger, The interrupt control system will assume that the interrupt is still in progress, with the consequence that interrupt requests of the same level or lower level will not be responded to.

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 Interrupt response time

Each interrupt has a different response time, depending on the nature of the interrupt and the instructions being executed at the time of the interrupt. If an interrupt is detected, the interrupt request flag bit is set, the internal circuit holds the flag bit, and the CPU generates an interrupt during the second clock cycle. If the response is valid and conditions permit, the hardware LCALL instruction invokes the service routine requesting the interrupt when the next instruction executes, otherwise the interrupt is suspended. Three clock cycles are required to execute the LCALL command. Therefore, at least 5 clock cycles are required from the time the interrupt flag is set to the time the interrupt service routine begins to execute.

When interrupt requests are blocked by any of the above three conditions, the interrupt response time increases. If interrupts of the same or higher priority are executing, the additional wait time depends on the length of the interrupt service routine being executed.

If the instructions are being executed is not in the last cycle, if is to execute commands RETI, complete RETI instructions are being executed, need four clock cycles, and for the longest time needed to complete the next instruction four clock cycles, if only one interrupt source in the system, plus LCALL call instruction 3 clock cycle, The maximum response time is 13 clock cycles.

Therefore, a simple interrupt system response time is always greater than 5 clock cycles and no more than 13 clock cycles.

8.7 External interrupt

HC89S105A 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 the interrupt service program is invoked, the flag bits from 0 to 15 of the external interrupt must be cleared by software. If the external interrupt persists after the interrupt service completes, the next interrupt occurs.

External interrupts 0 and 1 can be arbitrarily mapped to all ports, while external interrupts 2-15 are fixed.

8.8 Interrupt registers

8.8.1 Interrupt enable register IE，IE1，IE2

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	ET1	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	ET1	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/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	ET4	EX8_15	EX2_7	EADC	ET3	ET6	EIIC	ESPI

Bit	Flag	Introductions
7	ET4	T4 interrupt enable bit 0 : Disable T4 interrupt 1 : Enable T4 interrupt
6	EX8_15	External interrupt 8~15 interrupts enable bit 0 : Disable INT8~INT15 interrupts 1 : Enable INT8~INT15 interrupts Note: INT8~INT15 share the same interrupt vector.
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	ET3	T3 interrupt enable bit 0 : Disable T3 interrupt 1 : Enable T3 interrupt
2	ET6	T6 interrupt enable bit 0 : Disable T6 interrupt 1 : Enable T6 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

IE2

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	-	-	-	-	-	-	-	ET5

Bit	Flag	Introductions
7-1	-	Reserved Bits
0	ET5	T5 interrupt enable bit 0 : Disable T5 interrupt 1 : Enable T5 interrupt

8.8.2 Interrupt priority selection register IP0，IP1，IP2，IP3，IP4

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]		PX1[1:0]		PT0[1:0]		PX0[1:0]	

Bit	Flag	Introductions
7-6	PT1[1:0]	T1 interrupt priority control bits
5-4	PX1[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]	PCA interrupt priority control bits
3-2	PIIC [1:0]	IIC 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]		PT3[1:0]		PT6[1:0]	

Bit	Flag	Introductions
7-6	PX2_7[1:0]	INT2_7 interrupt priority control bits
5-4	PADC[1:0]	ADC interrupt priority control bits
3-2	PT3[1:0]	T3 interrupt priority control bits
1-0	PT6[1:0]	T6 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	PFLT[1:0]		PT5[1:0]		PT4[1:0]		PX8_15[1:0]	

Bit	Flag	Introductions
7-6	PFLT[1:0]	PWM fault detect interrupt priority control bits
5-4	PT5[1:0]	T5 interrupt priority control bits
3-2	PT4[1:0]	T4 interrupt priority control bits
1-0	PX8_15 [1:0]	INT8_15 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)

8.8.3 External interrupt level selection registers PITSx (x=0~3)

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]		IT8[1:0]	

PITS3

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

Bit	Flag	Introductions
7-6 5-4 3-2 1-0	ITx[1:0] (x = 0…15)	External interrupt trigger edge selection bits 00 : Low level interrupts 01 : Falling edge interrupts 10 : Rising edge interrupts 11 : Double edge interrupts

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

PINTE0

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

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

PINTE1

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

Bit	Flag	Introductions
7-0	EINTx (x =8...15)	External interrupt control bits (INT8~INT15) 0 : Disable the port interrupts 1 : Enable the port interrupts Note: As long as the corresponding EINTx (x =8...15) bits are enabled, the corresponding interrupt flags can be set to 1, otherwise, corresponding flags will not be set to 1.

8.8.5 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 and INT1 interrupt request flags 0 : When interrupt responded, hardware clear 0 automatically or software clear 0 1 : When external interrupts occur, hardware set 1

PINTF1

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

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

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 and ET1 bits will enable Timer 0 and Timer 1 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	16bit Timer/Counter
10	mode2	8bit 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

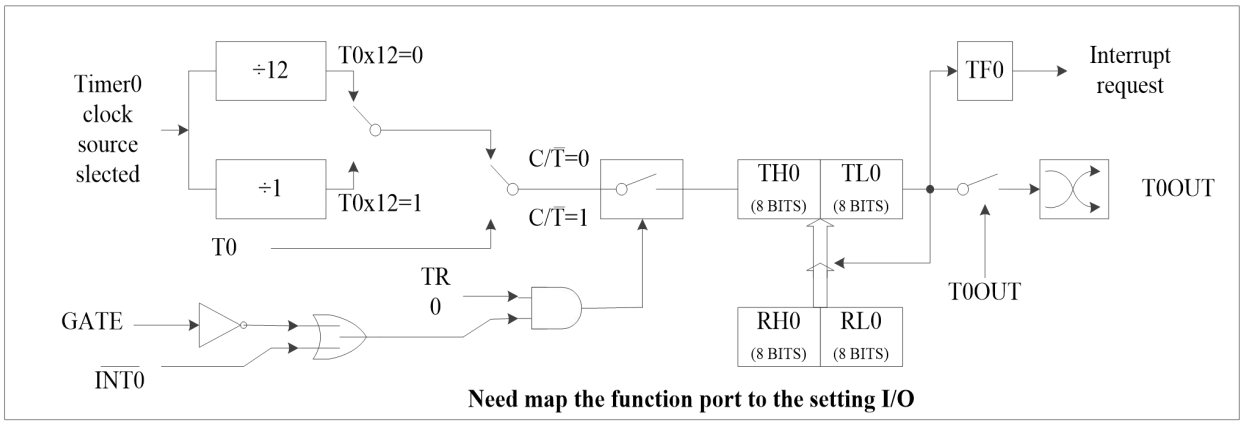


Figure 9- 1 TIMER0 mode0 function block diagram

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.

9.2.1.1 Mode1: 16 bit Timer/Counter

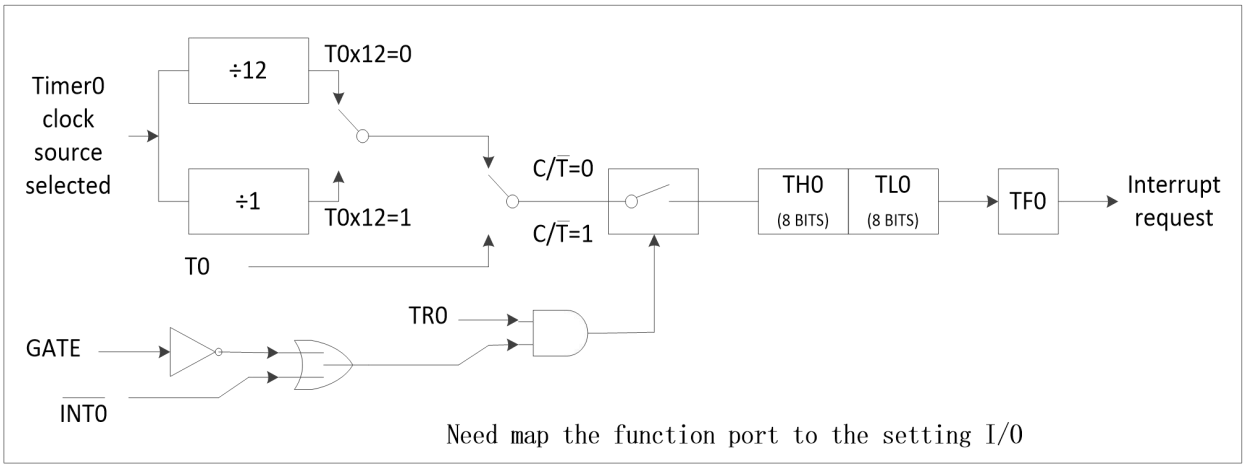


Figure 9- 2 TIMER0 mode1 function block diagram

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.

9.2.1.2 Mode2: 8 bit auto reload Timer/Counter

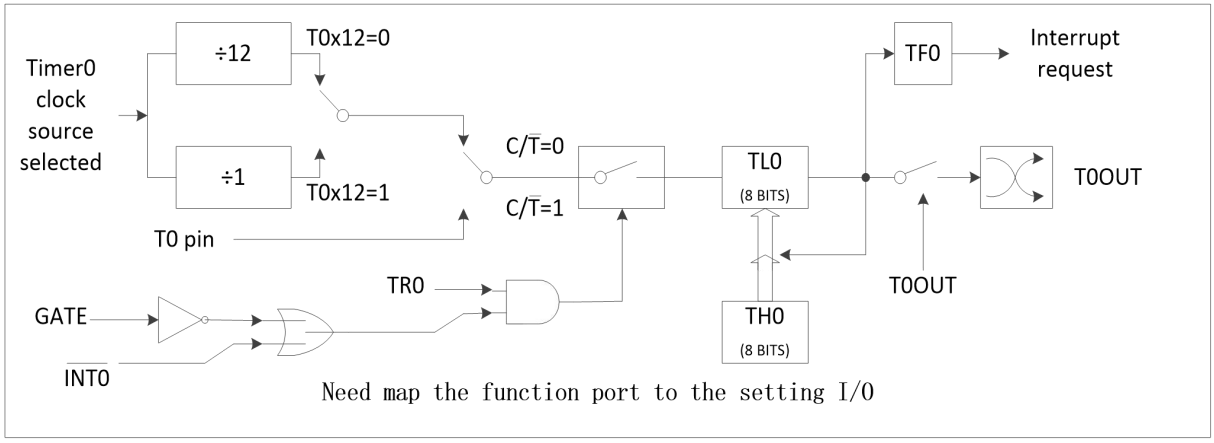


Figure 9- 3 TIMER0 mode2 Function block diagram

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.

9.2.1.3 Mode3: Two 8 bit Timer/Counter (T1 no this mode)

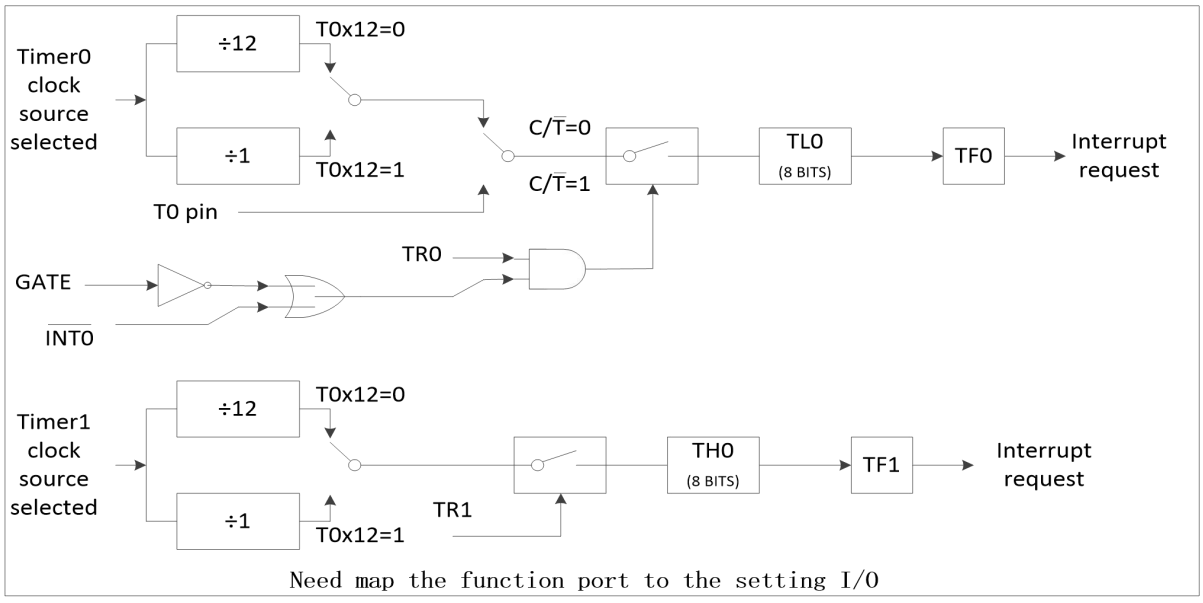


Figure 9- 4 TIMER0 mode 3 function block diagram

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).

When used as a timer, the T0OUT bit in register TCON1 can be configured to automatically flip the T0 pin when timer T0 overflows.

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 Bits

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 Fper/12 1 : Tx Timer clock Fper

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

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

TLx (x = 0,1)

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

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

THx (x = 0,1)

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

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

9.3 Timer/Counter 3

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

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

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

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

When T3CLKS[1:0] is 01, port T3 inputs the external clock, and timer 3 can work in normal mode or STOP mode. When T3CLKS[1:0] is 10, that is, the counting clock source of timer 3 is the external 32.768KHz low-frequency crystal oscillator (the external low-frequency crystal oscillator will not close in STOP mode), timer 3 can also work in normal mode or STOP mode at this time. When T3CLKS[1:0] is 11, that is, timer 3 counts clock source as internal low-frequency RC(RC44K), timer 3 can also work in normal mode or STOP mode.

When T3PD_EN is 1 and T3CLKS[1:0] is 01 or 10 or 11, T3 can work in STOP mode. When the T3 internal counter overflows from 0xFFFF to 0x0000, the chip wakes up from STOP mode. If interrupts are allowed, the awakened chip will enter timer 3 to interrupt.

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

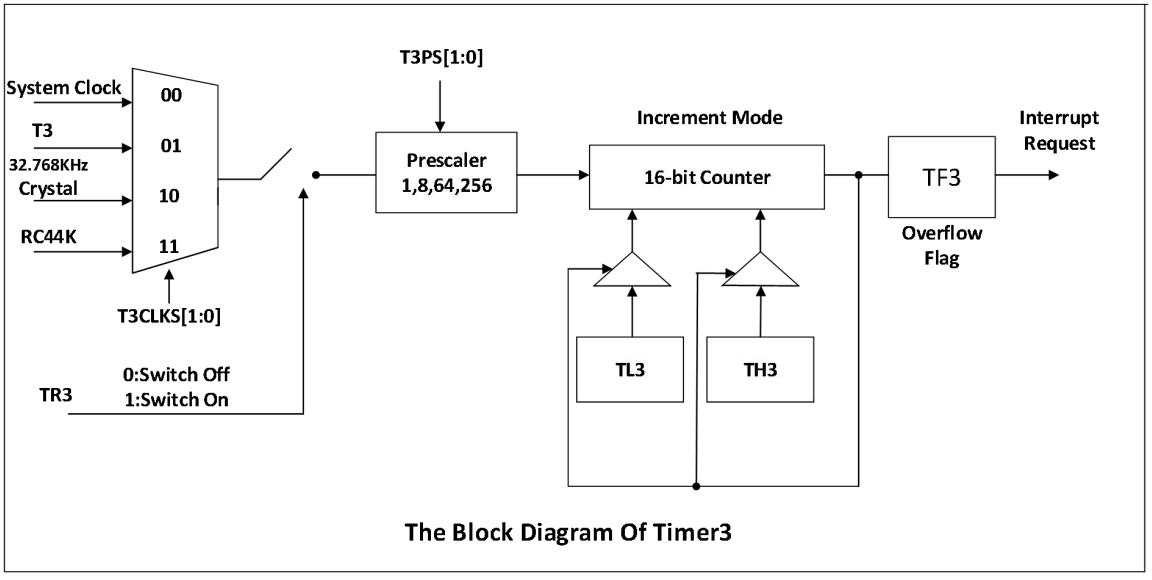


Figure 9- 5 TIMER3 function block diagram

9.3.1 Timer/Counter T3 registers

9.3.1.1 Timer/Counter T3 register T3CON

T3CON

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

Bit	Flag	Introductions
7	TF3	Timer 3 overflow flag 0 : Without overflow (hardware clear 0),software can also clear 0 1 : Overflow (hardware set 1)
6	T3PD_EN	Timer 3 Operation control bit in power-down mode 0 : Disable Timer power-down mode 3 1 : Enable Timer power-down mode 3, at this time T3CLKS [1:0] is 01 or 10 or 11 Note: When external crystal oscillator enable bit XTALEN set 1, crystal oscillator select low-frequency crystal oscillator, and timer 3 is allowed to work in STOP mode,and T3CLKS[1:0] set 10, if EA=1 and ET3=1, timer 3 will wake up after overflow and execute interrupt program of timer 3.
5-4	T3PS[1:0]	Timer 3 prescaler ratio selection bit 00：1/1 01：1/8 10：1/64 11：1/256
3	-	Reserved Bits
2	TR3	Timer 3 control enable bit 0 : Disable Timer 3 1 : Enable Timer 3
1-0	T3CLKS[1:0]	Timer 3 count clock source selection bits 00：The system clock Fper 01：T3 input an external clock 10：External low-frequency crystal oscillator 32.768KHz 11：Internal low-frequency RC (RC44K)

9.3.1.2 Timer T3 Data register TL3，TH3

TL3

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

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

TH3

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

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

9.4 Timer/Counter 4

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

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

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

9.4.1 Timer/Counter T4 work mode

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

9.4.1.1 Mode0: 16 bit auto reload Timer/Counter

Timer 4 is 16 bit auto reload Timer. TH4 register store 16 -bit Timer high 8 bits,TL4 store low 8 bits. When TR4 = 0, write TH4 and TL4 two registers sequentially, write the value is written into reload and count registers. TR4 is set to 1, and the internal 16-bit counter starts incrementing from the written value and overflows at 0xFFFF to 0x0000, at which point TF4 is set to 1. At the same time, the 16-bit data in the overloaded register is automatically reloaded into the internal 16-bit counter, which starts counting incrementally from the overloaded value.The interrupt will be generated If Timer 4 interrupt enabled.

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

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

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

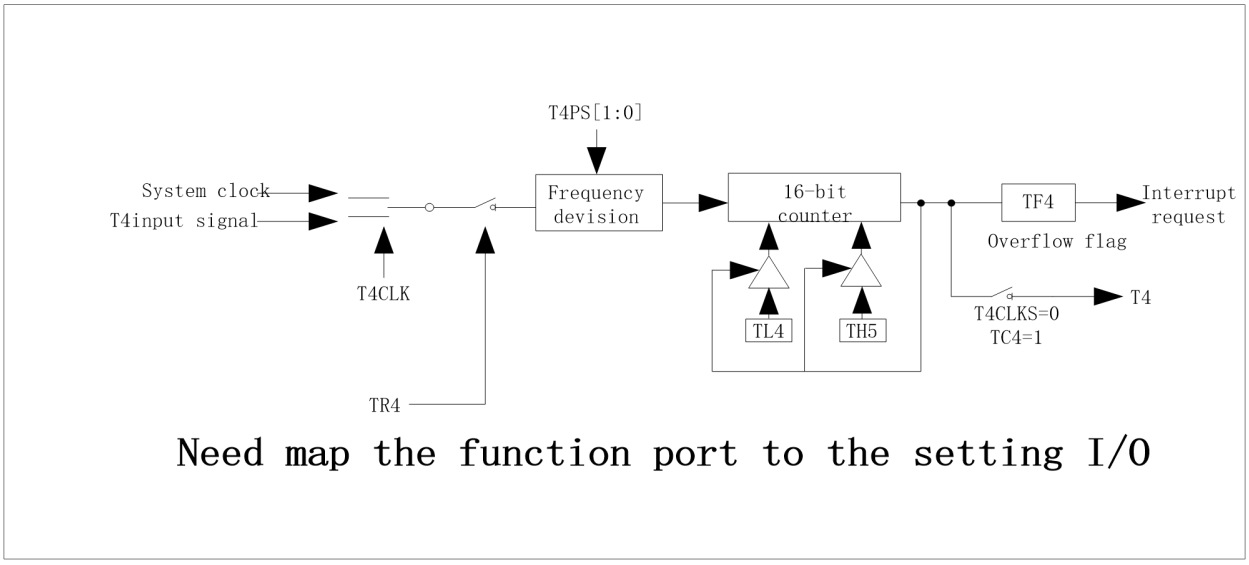


Figure 9- 6 TIMER4 mode0 functional block diagram

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

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

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

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

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

TR4 set 1 don't clear Timer 4 counter, before enable the Timer, user write an initial value expected to reload register.

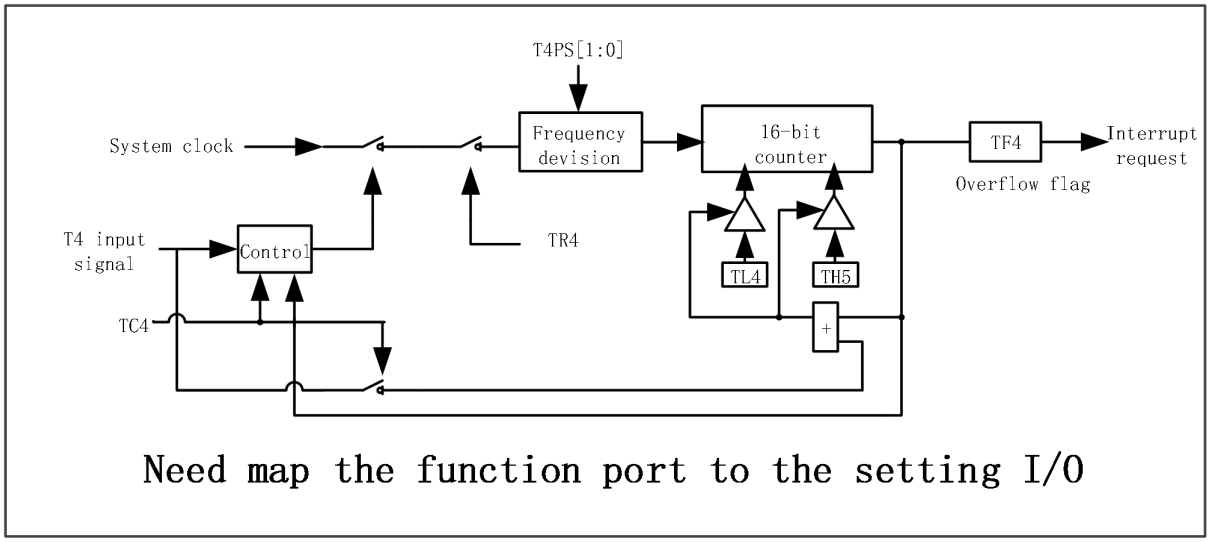


Figure 9- 7 TIMER4 mode2/3 functional block diagram

Note:

When the Timer 4 used as counter, input signal frequency of T4 pin must less than half of system clock.

9.4.2 Timer/Counter T4 registers

9.4.2.1 Timer T4 control register T4CON

T4CON

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

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

T4CON1

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	-						T4_PWMS	T4_PWM

Bit	Flag	Introductions
7-2	-	Reserved Bits
1	T4_PWMS	PWMOutput polarity selector bit 0: Indicates the high level during the validity period 1: Low level during the valid period Note: modification of this control bit is effective immediately. The period of validity refers to the duty cycle period
0	T4_PWM	Timer 4 acts as an 8-bit PWM 0: Normal timer function, compatible with previous 003 1: 8-bit PWM output, when T4_PWM=1, T4M=00, T4CLKS=0, TC4=0, T4_OUT output PWM waveform, where TH4 is the cycle register, TL4 is the duty cycle register, TR4 is used to start PWM output

9.4.2.2 Timer T4 data register TL4，TH4

TL4

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

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

TH4

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	TH4							

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

9.5 Timer 5

Timer 5 is 16 bit auto reload Timer. T5CON register control to access the two data registers TH5 and TL5. IE2 register ET5 bit set 1 enables Timer 5 interrupt (see interrupt chapter).

9.5.1 Timer T5 work mode

9.5.1.1 Mode0: 16 bit auto reload Timer/Counter

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

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

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

T5CON.1 register TR5 bit set 1 can enable Timer 5, and don't clear counter of Timer 5. Before enable Timer5, user should write an initial value to reload register user want.

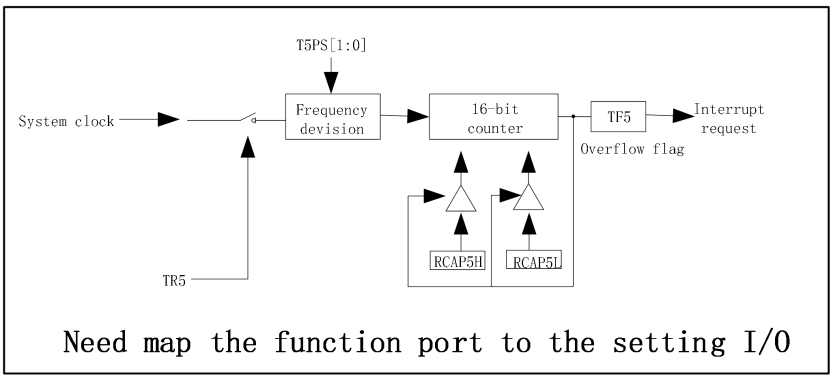


Figure 9- 8 TIMER5 Mode0 functional block diagram

9.5.1.2 Mode2: 16 bit rising edge captur

In capture mode, EXEN5 of T5CON has two options.

If EXEN5 = 0, the Timer5 as a 16-bit Timer, ET5 will be Permitting, Timer5 user can set TF5 Spillovers generate an interrupt. After the overflow count will not be reloaded, from 0 to count.

If EXEN5 = 1,the Timer5 do the same thing, but External input T5 (Available from the T5CON1 To select the capture input types) the rising edge can cause TH5 TL5 Current value is captured RCAP5H RCAP5L And, in addition, T5 On the rising edge can also result in T5CON EXF5 is set. ET5 enabled EXF5 like TF5 also generate an interrupt.

9.5.1.3 Mode3: 16 bit falling edge capture

In capture mode, EXEN5 of T5CON has two options.

If EXEN5 = 0, the Timer5 as a 16-bit Timer, ET5 will be Permitting, Timer5 user can set TF5 Spillovers generate an interrupt. After the overflow count will not be reloaded, from 0 to count.

If EXEN5 = 1,the Timer5 do the same thing, but External input T5 (Available from the T5CON1 To select the capture input types) the falling edge can cause TH5 TL5 Current value is captured RCAP5H RCAP5L And, in addition, T5 On the falling edge can also result in T5CON EXF5 is set. ET5 enabled EXF5 like TF5 also generate an interrupt.

9.5.2 Timer/Counter T5 registers

9.5.2.1 Timer T5 control register T5CON，T5CON1，T5CON2

T5CON

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

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

T5CON1

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

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

T5CON2

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	T5_PWMS	Reload_Sel			T5_MODE		CMPCR	T5CAPCR

Bit	Flag	Introductions
7	T5_PWMS	PWM output polarity selector bit 0：During the effective period of high level 1：During the effective period of low level Note: modification of this control bit is effective immediately. The period of validity refers to the duty cycle period
6-4	Reload_Sel	T5 Overload function selection bit 000: When timer 5 is used as a 16-bit auto overload timer, the count overflow of T5 causes an overload. 001: Timer 5 When used as a 16-bit automatic overload timer, EXEN5=1, a rising edge on the T5 pin will generate a overload. 010: When timer 5 is used as a 16-bit autoreload timer, EXEN5=1, a falling edge on the T5 pin will cause a reload. 011: When timer 5 is used as a 16-bit automatic overload timer, EXEN5=1, a falling edge or rising edge on the T5 pin will generate a overload. 100: Timer 5 when used as a 16-bit autoreload timer, an event on CAP0 causes a reload. 101: Timer 5 When used as a 16-bit autoreload timer, an event on CAP1 causes a reload.
3-2	T5_MODE	T5 extension mode selection 00: Determined by T5M[1:0] 01: rising edge capture or falling edge capture

		10: comparator mode 11: PWM mode In comparator mode, RCAP5H and RCAP5L prestore data to be compared. Since timer 5 counts up, once TH5 and TL5 match the set values of RCAP5H and RCAP5L, TF5 will be set to 1 by the hardware to indicate that a comparative match event has occurred. If CMPCR is set to 1, the timer 5 counter will automatically clear 0 after a comparison match event occurs. In PWM mode, set T5M[1:0]=00,TH0=0,TL0=0 {CAPH0, CAPL0} for storing PWM cycle; {CAPH1, CAPL1} for storing PWM duty cycle;
1	CMPCR	Comparison matches are cleared automatically This bit is valid only when timer 5 is in comparison mode. The hardware will automatically clear the TH5 and TL5 counters when they match. 0: After the comparison and match, timer 5 counts the value and continues counting according to the previous value. 1: After the comparison and match, timer 5 counts the value to clear 0.
0	T5CAPCR	T5 Port capture is automatically cleared This bit enables the hardware to automatically clear the TH5 and TL5 count registers when a capture event occurs and data in TH5 and TL5 is moved into the capture registers. Note: at position 1, if a capture event occurs, only TH5 and TL5 values are cleared. 0: T5 after the capture event occurs, timer 5 continues to accumulate the previous value 1: T5 port capture event occurs after the timer 5 count value automatically clear 0

9.5.2.2 Timer T5 data registers TL5，TH5

TL5

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

Bit	Flag	Introductions
7-0	TL5	T5 Data register low byte

TH5

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

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

9.5.2.3 Timer T5 reload capture registers RCAP5L，RCAP5H

RCAP5L

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	RCAP5L							

Bit	Flag	Introductions
7-0	RCAP5L	T5 reload capture registers low byte

RCAP5H

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	RCAP5H							

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

9.5.2.4 Capture control register CAPCON0，CAPCON1

CAPCON0

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	CAPEN1	CAPF1	CAP1LS[1:0]		CAPEN0	CAPF0	CAP0LS[1:0]	

Bit	Flag	Introductions
7	CAPEN1	T5 Input capture channel 1 enable control bit 1: T5 input capture channel 1 is enabled 0: disables T5 from entering capture channel 1
6	CAPF1	T5 Input capture channel 1 flag bit 1: this bit is set by hardware if input capture channel 1 detects the occurrence of edge triggering events 0: indicates that the software is cleared.
5-4	CAP1LS[1:0]	T5 Input capture channel 1 capture condition selection bit 00: falling edge 01: rising edge 10: rising edge or falling edge 11: keep
3	CAPEN0	T5 Input capture channel 0 enable control bit 1: enables T5 input capture channel 0 0: disables T5 from entering capture channel 0
2	CAPF0	T5 Input capture channel 0 flag bit 1: this bit is set by hardware if input capture channel 0 detects an edge triggering event 0: indicates that the software is cleared.
1-0	CAP0LS[1:0]	T5 Input capture channel 0 capture condition selection bit 00: falling edge 01: rising edge 10: rising edge or falling edge 11: keep

CAPCON1

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	-				CAP1_INTE	CAP0_INTE	CAP1CR	CAP0CR

Bit	Flag	Introductions
7-4	-	Reserved Bits
3	CAP1_INTE	CAP1 captures the interrupt enable bit 0: disable interruption 1: enable interruption
2	CAP0_INTE	CAP0 captures the interrupt enable bit 0: disable interruption 1: enable interruption
1	CAP1CR	CAP1 Port capture is automatically cleared This bit enables the hardware to automatically clear the TH5 and TL5 count registers when a capture event occurs and data in TH5 and TL5 is moved into the capture registers. 0: After CAP1 port capture event occurs timer 5 Accumulates the previous value 1: After CAP1 port capture event occurs, timer 5 automatically clears 0
0	CAP0CR	CAP0 Port capture is automatically cleared This bit enables the hardware to automatically clear the TH5 and TL5 count registers when a capture event occurs and data in TH5 and TL5 is moved into the capture registers. 0: After CAP0port capture event occurs timer 5 Accumulates the previous value 1: After CAP0 port capture event occurs, timer 5 automatically clears 0

Map CAP0 and CAP1 to a pin, and then set CAP0 to capture the rising edge, CAP1 to capture the falling edge, CAP0 configuration after capture zero, CAP1 configuration to capture not clear zero, so capture register {CAPH0, CAPL0} can capture the PWM cycle, {CAPH1, The duty cycle can be obtained in the CAPL1} register.

Introductions: Set this parameter to 0, T5_MODE=00, T5M[1:0]=00, and RCAP5L/RCAP5H to 0.

If interrupt enable ET5 of timer 5 is turned on, overflow interrupts will also occur when the counter is counted from 0 to FFFF.

CAP0 and CAP1 capture interrupts do not need to use ET5 in the IE2 register.

9.5.2.5 Capture register CAPL0，CAPH0，CAPL1，CAPH1

CAPLn (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	CAPLn[7:0] (n = 0~1)							

Bit	Flag	Introductions
7-0	CAPLn [7:0] (n = 0~1)	CAPn(n = 0~1) Input capture register low bytes

CAPHn (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	CAPHn[7:0] (n = 0~1)							

Bit	Flag	Introductions
7-0	CAPHn [7:0] (n = 0~1)	CAPn(n = 0~1) Input capture register high bytes

9.6 Timer 6

Timer 6 can generate fixed periodic interrupts and continuous fixed clock output.

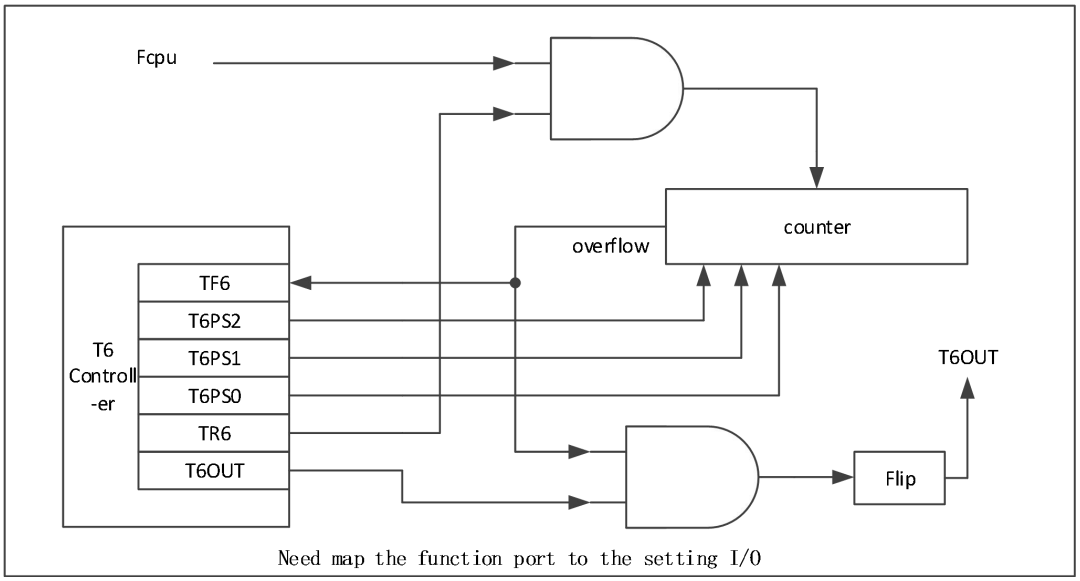


Figure 9- 9 T6 functional block diagram

9.6.1 Timer T6 control register T6CON

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R	R	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	TF6	-		T6PS[2:0]			TR6	T6OUT

Bit	Flag	Introductions
7	TF6	Timer 6 Interrupt flag bit 0: Software clearance 0 1: When the timer 6 counter overflows, the hardware is set to 1
6-5	-	Reserved (read = 0b, write invalid)
4-2	T6PS[2:0]	Timer 6 Counts the number of selected bits 000：2048 CPU clock cycles 001：4096 CPU clock cycles 010：8192 CPU clock cycles 011：16384 CPU clock cycles 100：32768 CPU clock cycles 101：65536 CPU clock cycles 110：131072 CPU clock cycles 111：262144 CPU clock cycles
1	TR6	Timer 6 runs the control bit 0: Stop timer 6 1: Start timer 6 (recount)
0	T6OUT	Timer 6 Enables the clock output bit 0：Disable clock output of timer 6 1： Allows timer 6 clock output(This port should be set to output mode in advance)

10 Pulse width modulation PWM

10.1 PWM Characteristics

- 3 complementary PWM outputs with dead-time insertion or 6 channels duty cycle independent PWM output
- 12 bit PWM or 16 bit PWM
- 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

HC89S105A integrates three 12/16 bit PWM modules PWM0, PWM1, and PWM2. Each of the three modules has a counter. The counter of PWM0 is controlled by EPWM0 or EPWM01 in PWMEN. The clock source of the counter is selected by CK0 in the PWM0C control register.

If EPWM0 is enabled and PWM0 is not mapped through the function pin mapping register, it will not output PWM0 from the chip pin. In this case, the PWM0 counter can be used as a timer. When the counter overflows, it will also generate PWM interrupts if interrupts are allowed.

The function and operation of the three PWM modules are exactly the same. The user can generate three PWM outputs with complementary deadband or six PWM outputs with independent duty cycle through the control registers.

If PWM0FLT_EN or PWM1FLT_EN or PWM2FLT_EN is set to 1, the PWM0/PWM1/PWM2 output and its complementary output can be turned off automatically by changing the FLT0/FLT1/FLT2 pin input signal. As soon as the effective level of the FLT pin input is detected, the PWM output is immediately turned off, but the PWM internal counter continues to operate, which is convenient to continue the PWM output after the FLT pin error is removed. FLTS bits cannot be cleared while the FLT input signal is active. Only when the FLT input signal disappears, the software can clear the FLTS status bit, and then the PWM returns to normal output.

Three PWM modules share an interrupt vector entry, but have their own control bit and flag bit, convenient for users to modify the PWM module next cycle cycle or duty cycle.

10.2 PWM output mode

The PWM output of the HC89S105A consists of two types: edge alignment and center alignment. Decide by setting TYPx (PWMCON0[7:5]) (x=0,1,2).

10.2.1 Edge alignment mode

In edge-aligned mode, the module generates edge-aligned PWM signals. The cycle of PWM output signal is determined by [PWM0PH:PWM0PL], and its duty cycle is determined by the corresponding duty cycle register (in the case of PWM independent output, the duty cycle registers of PWM01, PWM11, and PWM21 are their dead-time registers).

The 12/16 bit counter is in single-cycle mode, counting upward from 0000H, and all enabled PWM output is driven to a valid state at the beginning of the PWM cycle. When the value of the counter matches

the value of the PWM duty cycle register, the PWM output is driven to an invalid state. The counter continues until it matches [PWM0PH:PWM0PL], and then starts counting upwards again from 0000H.

PWMx period = [PWM0PH: PWM0PL] * PWM0 Period of the working clock source

PWMx duty cycle = [PWMxDH: PWMxDL] * PWM0 Working clock period

PWMx1 duty cycle = [PWMxDTH: PWMxDTL] * PWM0 Working clock period

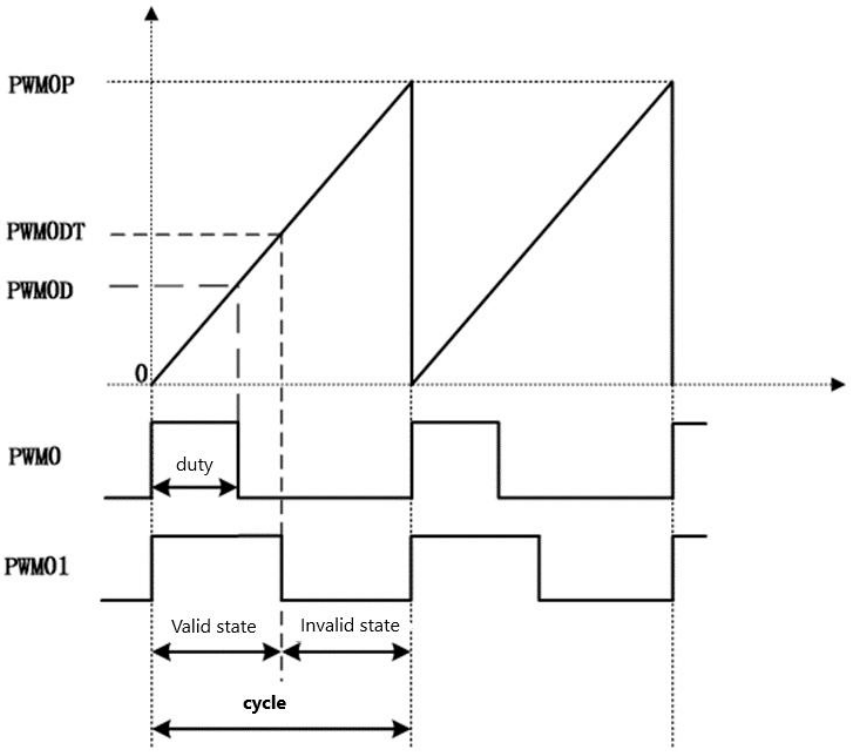


Figure 10- 1 PWM edge alignment waveform

10.2.2 Center alignment mode

In center aligned mode, the module generates a center aligned PWM signal. The cycle of PWM output signal is determined by [PWM0PH:PWM0PL], and its duty cycle is determined by the corresponding duty cycle register (in the case of PWM independent output, the duty cycle registers of PWM01, PWM11, and PWM21 are their dead-time registers).

The 12/16 bit counter is in dual-cycle mode, counting up from 0000H to [PWM0PH:PWM0PL] and down from [PWM0PH:PWM0PL] to 0000H, which is a full cycle of PWM. All enabled PWM outputs are driven to a valid state at the start of the PWM cycle. In the process of counting up, the PWM output is driven to an invalid state when the value of the counter matches the value of the PWM duty ratio register. The PWM output is not driven to a valid state until the counter is converted to a downward count and the value of the counter matches the value of the PWM duty ratio register.

PWMx period = [PWM0PH:PWM0PL] * PWM0 Working clock source period *2

PWMx duty cycle = [PWMxDH:PWMxDL] * PWM0 Working clock period *2

PWMx1 duty cycle = [PWMxDTH:PWMxDTL] * PWM0 Working clock period *2

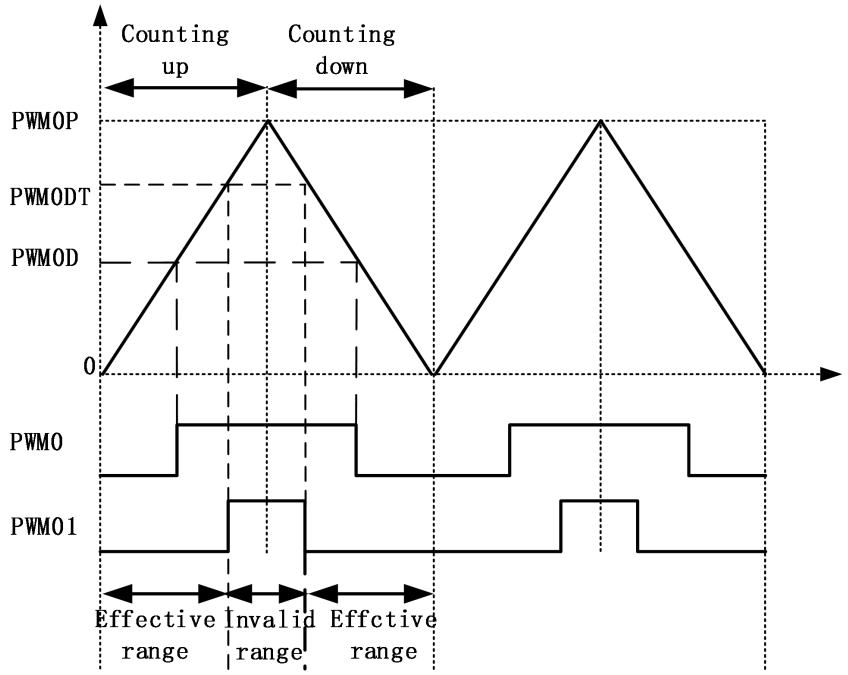


Figure 10- 2 PWM center alignment waveform

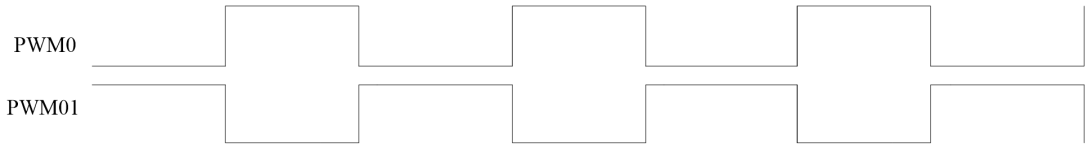
10.3 PWM output mode

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

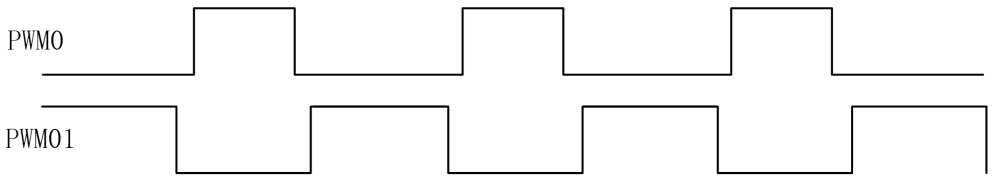
10.3.1 Complementary output mode

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

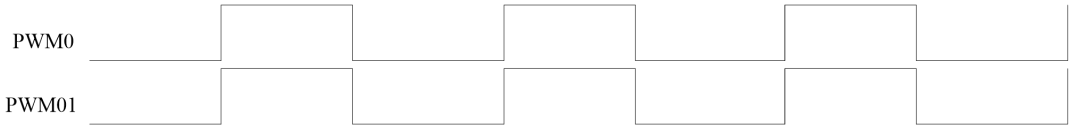
PWM0S=00& PWM0M=0：PWM0 and PWM01 work in complementary mode and are high level valid



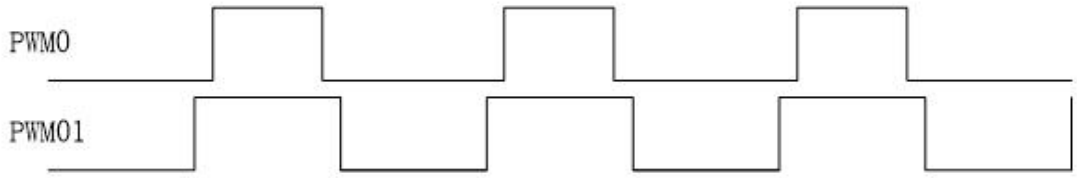
PWM0S=00& PWM0M=0： PWM0 and PWM01 work in complementary mode (with dead zone) and are highly level valid



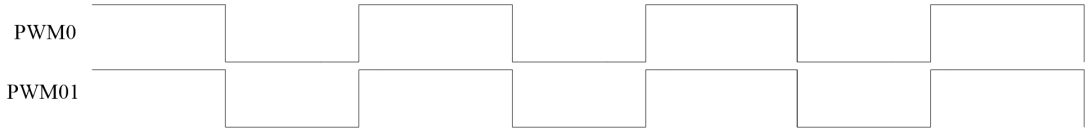
PWM0S=01& PWM0M=0：PWM0 and PWM01 work in complementary mode and PWM0 is high level valid and PWM01 is low level valid



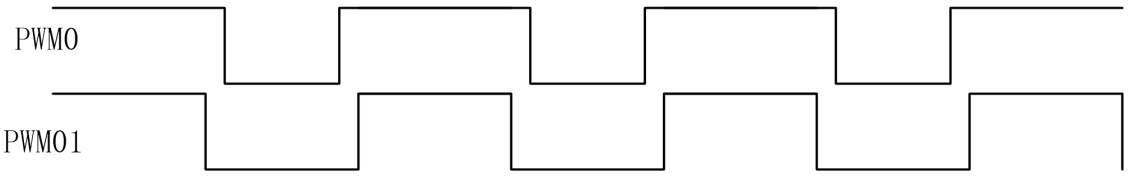
PWM0S=01& PWM0M=0：PWM0 and PWM01 work in complementary mode (with dead zone) and PWM0 is high efficiency and PWM01 is low level valid



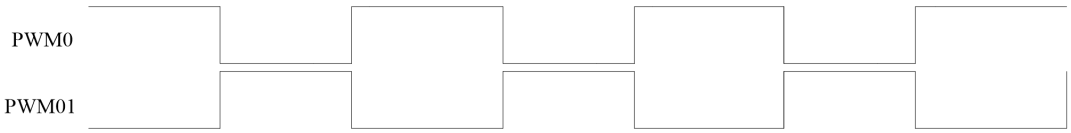
PWM0S=10& PWM0M=0：PWM0 and PWM01 work in complementary mode and PWM0 is low level valid and PWM01 is high level valid



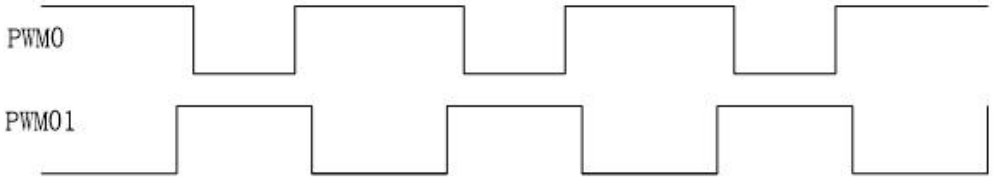
PWM0S=10& PWM0M=0：PWM0 and PWM01 work in complementary mode (with dead zone) and PWM0 is low level valid and PWM01 is high level valid



PWM0S=11& PWM0M=0：PWM0 and PWM01 work in complementary mode and both PWM0 and PWM01have low level valid



PWM0S=11& PWM0M=0：PWM0 and PWM01 work in complementary mode (with dead zone) and both PWM0 and PWM01 have low level valid

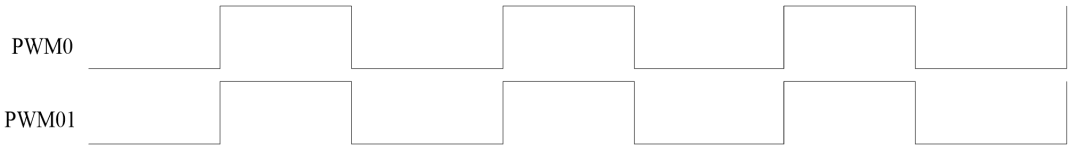


10.3.2 Independent output mode

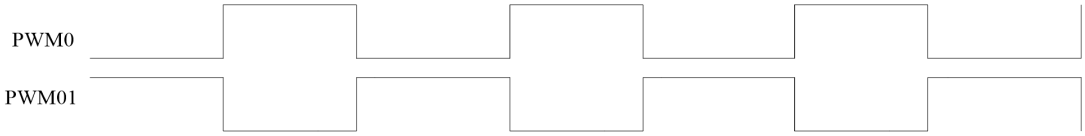
Set PWMxM to 1: PWM will be working in independent mode, user can control PWMx&PWMx1 single or both output. their cycle time are same but the duty cycle can be set individually. Duty cycle register control PWMx duty cycle, dead-time control registers control PWMx1 duty cycle,

The PWMx&PWMx1 polarity can be selected in independent output mode. It is easy to user multiple level driven request(x =0, 1, 2).

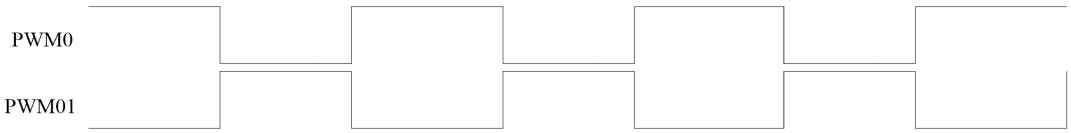
PWM0S=00& PWM0M=1：PWM0 and PWM01 work in independent mode , and both PWM0 and PWM01 are highly level valid



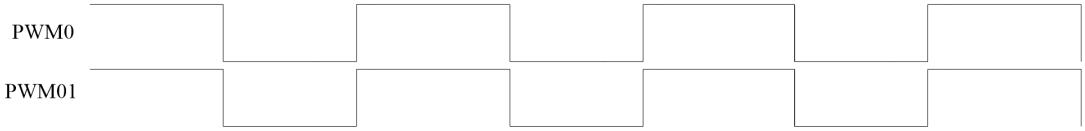
PWM0S=01& PWM0M=1：PWM0 and PWM01 work in independent mode .PWM0 is high level valid and PWM01 is low level valid



PWM0S=10& PWM0M=1：PWM0 and PWM01 work in independent mode and PWM0 is low level valid and PWM01 is high level valid



PWM0S=11& PWM0M=1：PWM0 and PWM01 work in independent mode and both PWM0 and PWM01 are low level valid



10.4 PWM registers

10.4.1 PWM control register PWMCON0，PWMCON1，PWMCON2

PWMCON0

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	TYP2	TYP1	TYP0	RLOAD2	RLOAD1	RLOAD0	PWMLEN	-

Bit	Flag	Introductions
7	TYP2	PWM2 Output type selection bit 0：PWM2 edge alignment 1：PWM2 center alignment
6	TYP1	PWM1 Output type selection bit 0：PWM1 edge alignment 1：PWM1 center alignment
5	TYP0	PWM0 Output type selection bit 0：PWM0 edge alignment 1：PWM0 center alignment
4	RLOAD2	PWM2 Auto overload enable bit 0：Disable automatic reloading 1：Enable automatic reloading Note: The default value is 0. By default, parameters cannot be loaded automatically after modification. Prohibit automatic overload before modifying parameters, enable after modifying parameters, can realize the synchronization between multiple groups of PWM, before the output state during the prohibition. **Suitable for three groups of PWM period is the same, want to modify the PWM period or duty cycle, if the three groups of PWM period is different, then PWM_OV signal is different**
3	RLOAD1	PWM1 Auto overload enable bit 0：Disable automatic reloading 1：Enable automatic reloading
2	RLOAD0	PWM0 Auto overload enable bit 0：Disable automatic reloading 1：Enable automatic reloading
1	PWMLEN	PWM precision selector bit 0：12 bit PWM count 1：16 bit PWM count Note: when the user selects 12-bit PWM count, the high 4 bits of

		PWMxPH, PWMxDH, and PWMxDTH(x = 0,1,2) are invalid, and the operation of the high 4 bits has no effect on the customer.
0	-	Reserved Bits

PWMCON1

Bit	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-	PWM0_ CMP_INTF	PWM0_ CMP_INTEN	PWM0_ CMPEN	-	PWM2_ PHASE	PWM1_ PHASE	PWM0_ PHASE

Bit	Flag	Introductions
7	-	Reserved Bits
6	PWM0_CMP_INTF	PWM0 counter matching flag 0：Software reset 1：When the PWM0 counter is the same as PWM0_CMP,set 1
5	PWM0_CMP_INTEN	PWM0 counter matching interrupt enabled 0：Disable 1：enabled
4	PWM0_CMPEN	PWM0 enable the counter matching mode 0：Disable 1：enabled
3	-	Reserved Bits
2	PWM2_PHASE	PWM2 phase shift contol bit 0：No phase shift 1： PWM21 phase shifts PWM2_PHASE_CNT PWM CLK relative to PWM2
1	PWM1_PHASE	PWM1 phase shift contol bit 0：No phase shift 1： PWM11 phase shifts PWM1_PHASE_CNT PWM CLKS relative to PWM1
0	PWM0_PHASE	PWM0 phase shift contol bit 0：No phase shift 1：Phase shift PWM01 relative to PWM0 PWM0_PHASE_CNT Number of PWM CLKS Note: Phase shifts work only if the independent mode edges are aligned. The number of phase shifts cannot be greater than the value assigned to the periodic register

PWMCON2

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	0	0	0	0	0	0
Flag			FLT_IO_SEL		DBG_STBEN	FLT_CTRL_MODE		FLT_INT_EN

Bit	Flag	Introductions
7-6	-	Reserved Bits
5-4	FLT_IO_SEL	FLT port selection 0x：Map register decision 10：FLT0/1/2 fixed from ADC analog watchdog 11：FLT0/1/2 fixed comparator from LVD
3	DBG_STBEN	In emulation mode PWM is disabled 0: simulation mode does not stop PWM output 1: Stop PWM output in simulation mode and restore output at full speed
2-1	FLT_CTRL_MODE	After the FTL valid signal is withdrawn, the PWM output is selected 00：Automatically restores the previous output 01：Restores the previous output after the current PWM cycle ends 10： the previous output cannot be restored. You need to reconfigure the PWM enabling position 11：Reserved
0	FLT_INT_EN	FLT interrupt enable bit 0：disable 1：enabled FLT is a separate interrupt vector.

10.4.2 PWM control register PWMCON3，PWMCON4

PWMCON3

Bit	7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-	FLT2_DBC_ENB	FLT1_DBC_ENB	FLT0_DBC_ENB	-	PWM2M	PWM1M	PWM0M

Bit	Flag	Introductions
7	-	Reserved Bits
6	FLT2_DBC_ENB	FLT2 port debouncing enable 0：enabled 1：disable
5	FLT1_DBC_ENB	FLT1 port debouncing enable 0：enabled 1：disable
4	FLT0_DBC_ENB	FLT0 port debouncing enable 0：enabled 1：disable
3	-	Reserved Bits
2	PWM2M	PWM2 work mode selection bit 0：PWM2&PWM21 work in complementary output mode 1：PWM2&PWM21 work in independent output mode Note: You are advised to disable the PWM2 module before changing the PWM2 working mode.
1	PWM1M	PWM1 work mode selection bit 0：PWM1&PWM11 work in complementary output mode 1：PWM1&PWM11 work in independent output mode Note: You are advised to disable the PWM1 module before changing the PWM1 working mode.
0	PWM0M	PWM0 work mode selection bit 0：PWM0&PWM01 work in complementary output mode 1：PWM0&PWM01 work in independent output mode Note: You are advised to disable the PWM0 module before changing the PWM0 working mode.

PWMCON4

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	0	0	0	0	0	0
Flag	-	-	PWM2_CKS		PWM1_CKS		PWM0_CKS	

Bit	Flag	Introductions
7-6	-	Reserved Bits
5-4	PWM2_CK1	PWM2 clock source selection bit 00：Determined by CK2 in the PWM2C register 01：Fosc/32 （CK0=00） 10：Fosc/64 （CK0=00） 11：Fosc/128（CK0=00） 11：Fosc（CK0=01） Note: Fosc is the system clock without any frequency division.
3-2	PWM1_CK1	PWM1 clock source selection bit 00：Determined by CK1 in the PWM1C register 01：Fosc/32 （CK0=00） 10：Fosc/64 （CK0=00） 11：Fosc/128（CK0=00） 11：Fosc（CK0=01） Note: Fosc is the system clock without any frequency division.
1-0	PWM0_CK1	PWM0 clock source selection bit 00：Determined by CK0 in the PWM0C register 01：Fosc/32 （CK0=00） 10：Fosc/64 （CK0=00） 11：Fosc/128（CK0=00） 11：Fosc（CK0=01） Note: Fosc is the system clock without any frequency division.

10.4.3 PWM enable register PWMEN

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	PWM0FLT_EN	EPWM21	EPWM11	EPWM01	-	EPWM2	EPWM1	EPWM0

Bit	Flag	Introductions
7	PWM0FLT_EN	PWM0 fault detection enable bit 0：disables fault detection 1：allows fault detection. You need to set FLT0 pin mapping and IO mode first Note: Both complementary and independent output modes can be controlled by fault detection pins.
6	EPWM21	PWM21 enable control bit 0: disables the output of PWM21 1: allows the output of PWM21 and requires the pin mapping of PWM21
5	EPWM11	PWM11 enable control bit 0: disables the output of PWM11 1: allows the output of PWM11 and requires the pin mapping of PWM11
4	EPWM01	PWM01 enable control bit 0: disables the output of PWM01 1: allows the output of PWM01 and requires the pin mapping of PWM01
3	-	Reserved Bits
2	EPWM2	PWM2 enable control bit 0: disables the output of PWM2 1: allows the output of PWM2 and requires the pin mapping of PWM2
1	EPWM1	PWM1 enable control bit 0: disables the output of PWM1 1: allows the output of PWM1 and requires the pin mapping of PWM1
0	EPWM0	PWM0 enable control bit 0: disables the output of PWM0 1: allows the output of PWM0 and requires the pin mapping of PWM0

10.4.4 PWM FLTcontrol register PWMFLT

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	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：disables fault detection 1： allows fault detection. You need to set FLT2 pin mapping and IO mode first Note: Both complementary and independent output modes can be controlled by fault detection pins.
6	PWM1FLT_EN	PWM1 fault detection enable bit 0：disables fault detection 1： allows fault detection. You need to set FLT1 pin mapping and IO mode first Note: Both complementary and independent output modes can be controlled by fault detection pins.
5-4	PWM2_FLT_MODE	PWM2 Fault output preset status selection bit 00: PWM2 and PWM21 are both low levels during the fault period 01: Low level during PWM2 failure, high level during PWM21 failure 10: High level during PWM2 failure, low level during PWM21 failure 11: PWM2 and PWM21 are both high levels during faults
3-2	PWM1_FLT_MODE	PWM1 Fault output preset status selection bit 00：PWM1 and PWM11 are both low levels during the fault period 01: Low level during PWM1 failure, high level during PWM11 failure 10: High level during PWM1 failure, low level during PWM11 failure 11: PWM1 and PWM11 are both high levels during faults
1-0	PWM0_FLT_MODE	PWM0 Fault output preset status selection bit 00：PWM0 and PWM01 are both low levels during the fault period 01: Low level during PWM0 failure, high level during PWM01 failure 10: High level during PWM0 failure, low level during PWM01 failure 11: PWM0 and PWM01 are both high levels during faults

10.4.5 PWM0 module

10.4.5.1 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 bit 0：Software reset 1：PWM0 Cycle counter overflows and is set to 1 by hardware
5	PWM0FLTS	PWM0 FLT status bit 0：PWM In normal state, the software is cleared 1：PWM The output is off and the hardware is set to 1
4	PWM0FLTC	PWM0 FLTPin configuration bits 0：FLT0 is at low power, PWM output is off 1：FLT0 is at high power, PWM output is off
3-2	PWM0S	PWM0 and PWM01 Duty cycle output mode selection bit 00: PWM0 and PWM01 are both highly valid 01: PWM0 indicates high validity, and PWM01 indicates low validity 10: PWM0 indicates low validity, and PWM01 indicates high validity 11: PWM0 and PWM01 are both low valid
1-0	CK0	PWM0 clock source selection register 00：Fosc/2 01：Fosc/4 10：Fosc/8 11：Fosc/16 Note: Fosc is the system clock without any frequency division.

10.4.5.2 PWM0 Periodic 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 is 8 bits lower

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 is 4 bits higher

Note: When modifying the PWM0 cycle, modify the high value first and then the low value. The read time is not restricted, for example：

```
PWM0PH = 0x05;
PWM0PL = 0x08;  //PWM counter overflow, the cycle data is 0x0508 form the next cycle
PWM0PH = 0x06;  //PWM counter overflow, the cycle data is 0x0508 form the next cycle
PWM0PL = 0x08;  //PWM counter overflow, the cycle data is 0x0608 form the next cycle
PWM0PL = 0x09;  ///PWM counter overflow, the cycle data is 0x0609 form the next cycle
```

Obviously, whenever the PWM cycle is changed, the low register must be written once regardless of whether the low register needs to be changed or not, and the period change will only take effect in the next PWM cycle (modification is not restricted when the PWM is off).

PWM0 Period = [PWM0PH: PWM0PL] * PWM0 Period of the working clock source

10.4.5.3 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

10.4.5.4 PWM0 dead time register PWM0DTL，PWM0DTH

PWM0DTL

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

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

PWM0DTH

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-	-	-	-	PWM0DTH[3:0]			

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

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

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

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

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

10.4.5.5 PWM0 fault input pin debouncing 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 debouncing clock selection 00：Fper/1 01：Fper/4 10：Fper/16 11：Fper/64
5-0	PWM0DBCT[5:0]	Number of debouncing counter clocks on a port. If this parameter is set to 00, debouncing does not occur.

Debouncing time = frequency division coefficient * T_{CPU} * PWM0DBCT [5:0]

Note: the time of debouncing elimination is not accurate, the real number of debouncing elimination is between the configured value minus one and the configured value.

10.4.5.6 PWM0 phase shift number register PWM0PHASEH，PWM0PHASEL

PWM0PHASEH

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	PWM0PHASE[15:8]							

Bit	Flag	Introductions
7-0	PWM0PHASE[15:8]	PWM0 phase shift number register high 8 bits

PWM0PHASEL

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	PWM0PHASE [7:0]							

Bit	Flag	Introductions
7-0	PWM0PHASE [7:0]	PWM0 phase shift number register low 8 bits

10.4.5.7 PWM0 Count matching register PWM0CMPH，PWM0CMPL

PWM0CMPH

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	PWM0CMP[15:8]							

Bit	Flag	Introductions
7-0	PWM0CMP[15:8]	PWM0 Count matching register high 8 bits

PWM0CMPL

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	PWM0CMP[7:0]							

Bit	Flag	Introductions
7-0	PWM0CMP[7:0]	PWM0 Count matching register low 8 bits

Note: When the PWM0 count matches PWM0_CMP, raise the PWM0 counter matching flag

10.4.5.8 PWM0 number of interrupt frequency division PWM0INTDIV

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	PWM0_CMP_INT_DIV				PWM0_OV_INT_DIV			

Bit	Flag	Introductions
7-4	PWM0_CMP_INT_DIV	PWM0 Matching interrupt frequency division
3-0	PWM0_OV_INT_DIV	PWM0 Overflow interrupt frequency division

10.4.6 PWM1 module

10.4.6.1 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		CK1	

Bit	Flag	Introductions
7	PWM1IE	PWM1 interrupt enable bit 0：Disable PWM1 interrupt 1：Enable PWM1 interrupt
6	PWM1IF	PWM1 interrupt flag bit 0：Software reset 1：PWM1 Cycle counter overflows and is set to 1 by hardware
5	PWM1FLTS	PWM1 FLT status bit 0：PWM In normal state, the software is cleared 1：PWM The output is off and the hardware is set to 1
4	PWM1FLTC	PWM1 FLTPin configuration bits 0：FLT1 is at low power, PWM output is off 1：FLT1 is at high power, PWM output is off
3-2	PWM1S	PWM1 and PWM11 Duty cycle output mode selection bit 00: PWM1 and PWM11 are both highly valid 01: PWM1 indicates high validity, and PWM11 indicates low validity 10: PWM1 indicates low validity, and PWM11 indicates high validity 11: PWM1 and PWM11 are both low valid
1-0	CK1	PWM1 clock source selection register 00：Fosc/2 01：Fosc/4 10：Fosc/8 11：Fosc/16 Note: Fosc is the system clock without any frequency division.

10.4.6.1 PWM1 periodic 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 is 8 bits lower

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 is 4 bits higher

Note: When modifying the PWM1 cycle, modify the high value first and then the low value. The read time is not restricted, for example：

```
PWM1PH = 0x05;
PWM1PL = 0x08;  //PWM counter overflow, the cycle data is 0x0508 form the next cycle
PWM1PH = 0x06;  //PWM counter overflow, the cycle data is 0x0508 form the next cycle
PWM1PL = 0x08;  //PWM counter overflow, the cycle data is 0x0608 form the next cycle
PWM1PL = 0x09;  ///PWM counter overflow, the cycle data is 0x0609 form the next cycle
```

Obviously, whenever the PWM cycle is changed, the low register must be written once regardless of whether the low register needs to be changed or not, and the period change will only take effect in the next PWM cycle (modification is not restricted when the PWM is off).

PWM1 Period = [PWM1PH: PWM1PL] * PWM1 Period of the working clock source

10.4.6.2 PWM1 duty cycle 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 cycle 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 cycle register 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.

10.4.6.3 PWM1 dead time register PWM1DTL，PWM1DTH

PWM1DTL

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

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

PWM1DTH

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-				PWM1DTH[3:0]			

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

When PWM1M=1, PWM1 work in 2 road independent mode, dead time register is used as the PWM11 Duty cycle registers, namely independent mode PWM1 can output 2 road PWM waveform with the same cycle, but different duty cycle.

Complementary mode: PWM1 Dead time = [PWM1DTH:PWM1DTL] * PWM1 clock cycle.

Complementary mode: dead time must be less than the duty cycle time, sum of dead time and duty cycle time must be less than PWM1 cycle.

Independent mode: PWM11 Duty cycle time = [PWM1DTH:PWM1DTL] * PWM1 Clock cycle.

10.4.6.4 PWM1 fault input pin debouncing 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 debouncing clock selection 00：Fper/1 01：Fper/4 10：Fper/16 11：Fper/64
5-0	PWM1DBCT[5:0]	Number of debouncing counter clocks on a port. If this parameter is set to 00, debouncing does not occur.

Debouncing time = frequency division coefficient * T_{PER} * PWM1DBCT [5:0]

Note: the time of debouncing elimination is not accurate, the real number of debouncing elimination is between the configured value minus one and the configured value.

10.4.6.5 PWM1 phase shift number register PWM1PHASEH，PWM1PHASEL

PWM1PHASEH

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	PWM1PHASE[15:8]							

Bit	Flag	Introductions
7-0	PWM1PHASE[15:8]	PWM1 phase shift number register high 8 bits

PWM1PHASEL

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	PWM1PHASE [7:0]							

Bit	Flag	Introductions
7-0	PWM1PHASE [7:0]	PWM1 phase shift number register low 8 bits

10.4.6.6 PWM1 number of interrupt frequency division PWM1INTDIV

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	-				PWM1_OV_INT_DIV			

Bit	Flag	Introductions
7-4	-	Reserved Bits
3-0	PWM1_OV_INT_DIV	PWM1 Overflow interrupt frequency division

10.4.7 PWM2 module

10.4.7.1 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		CK2	

Bit	Flag	Introductions
7	PWM2IE	PWM2 interrupt enable bit 0：Disable PWM2 interrupt 1：Enable PWM2 interrupt
6	PWM2IF	PWM2 interrupt flag bit 0：Software reset 1：PWM2 Cycle counter overflows and is set to 1 by hardware
5	PWM2FLTS	PWM2 FLT status bit 0：PWM In normal state, the software is cleared 1：PWM The output is off and the hardware is set to 1
4	PWM2FLTC	PWM2 FLTPin configuration bits 0：FLT2 is at low power, PWM output is off 1：FLT2 is at high power, PWM output is off
3-2	PWM2S	PWM2 and PWM21 Duty cycle output mode selection bit 00: PWM2 and PWM21 are both highly valid 01: PWM2 indicates high validity, and PWM21 indicates low validity 10: PWM2 indicates low validity, and PWM21 indicates high validity 11: PWM2 and PWM21 are both low valid
1-0	CK2	PWM2 clock source selection register 00：Fosc/2 01：Fosc/4 10：Fosc/8 11：Fosc/16 Note: Fosc is the system clock without any frequency division.

10.4.7.2 PWM2 Periodic 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 Periodic 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 Periodic register high 4 bits

Note: When modifying the PWM0 cycle, modify the high value first and then the low value. The read time is not restricted, for example：

```
PWM0PH = 0x05;
PWM0PL = 0x08;  //PWM counter overflow, the cycle data is 0x0508 form the next cycle
PWM0PH = 0x06;  //PWM counter overflow, the cycle data is 0x0508 form the next cycle
PWM0PL = 0x08;  //PWM counter overflow, the cycle data is 0x0608 form the next cycle
PWM0PL = 0x09;  ///PWM counter overflow, the cycle data is 0x0609 form the next cycle
```

Obviously, whenever the PWM cycle is changed, the low register must be written once regardless of whether the low register needs to be changed or not, and the period change will only take effect in the next PWM cycle (modification is not restricted when the PWM is off).

PWM0 Period = [PWM0PH: PWM0PL] * PWM0 Period of the working clock source

10.4.7.3 PWM2 duty cycle 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 cycle 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 cycle register high 4 bits

Note: modify PWM2 Duty cycle registers, similar to modify PWM2 cycle register, both are required to modify the high level first then low, and changes will valid from the next cycle.

10.4.7.4 PWM2 dead time register PWM2DTL，PWM2DTH

PWM2DTL

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

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

PWM2DTH

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-				PWM2DTH[3:0]			

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

When PWM2M=1, PWM2 work in 2 road independent mode, dead time register is used as the PWM21 Duty cycle registers, namely independent mode PWM2 can output 2 road PWM waveform with the same cycle, but different duty cycle.

Complementary mode: PWM2 Dead time = [PWM2DTH:PWM2DTL] * PWM2 clock cycle.

Complementary mode: dead time must be less than the duty cycle time, sum of dead time and duty cycle time must be less than PWM2 cycle.

Independent mode: PWM21 Duty cycle time = [PWM2DTH:PWM2DTL] * PWM2 Clock cycle.

10.4.7.5 PWM2 fault input pin debouncing 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 debouncing clock selection 00：Fper/1 01：Fper/4 10：Fper/16 11：Fper/64
5-0	PWM2DBCT[5:0]	Number of debouncing counter clocks on a port. If this parameter is set to 00, debouncing does not occur.

Debouncing time = frequency division coefficient * T_{PER} * PWM2DBCT [5:0]

Note: the time of debouncing elimination is not accurate, the real number of debouncing elimination is between the configured value minus one and the configured value.

10.4.7.6 PWM2 phase shift number register PWM2PHASEH，PWM2PHASEL

PWM2PHASEH

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	PWM2PHASE[15:8]							

Bit	Flag	Introductions
7-0	PWM2PHASE[15:8]	PWM2 phase shift number register high 8 bits

PWM2PHASEL

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	PWM2PHASE [7:0]							

Bit	Flag	Introductions
7-0	PWM2PHASE [7:0]	PWM2 phase shift number register low 8 bits

10.4.7.7 PWM2 number of interrupt frequency division PWM2INTDIV

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	-				PWM2_OV_INT_DIV			

Bit	Flag	Introductions
7-4	-	Reserved Bits
3-0	PWM2_OV_INT_DIV	PWM2 Overflow interrupt frequency division

11 Single 8 bit PWM

11.1 PWM characteristics

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

11.2 PWM module registers

11.2.1 PWM3 control register PWM3C

PWM3C

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

Bit	Flag	Introductions
7	PWM3EN	PWM3 module enable control bit 0 : Close PWM3 module 1 : Open PWM3 module (re-count) Note: When PWM close, counter stopped and output close immediately .When PWM open, PWM re-count from 1, output controlled by PWM3OEN bit.
6	PWM3IE	PWM3 interrupt enable bit 0：Disable PWM3 interrupt 1：Enable PWM3 interrupt
5	PWM3IF	PWM3 interrupt flag 0: Software clear 0 1: Hardware set 1, only set to 1 when the PWM3 Counter overflow (greater than PWM3P)
4	PWM3OEN	PWM3 output enable bit 0 : PWM3 Disable output 1 : PWM3 Enable output Note: PWM3 output is enabled when PWM3EN is set to 1, otherwise PWM3 output disable. When PWM3OEN=0, if PWM3EN=1, PWM3 can overflow interrupt, that is, PWM3 can be used as a timer. The change of the control bit takes effect immediately.
3	PWM3S	PWM3 output polarity selection bit 0 : PWM3 high level during valid period 1 : PWM3 low level during valid period

		Note: modify the control bit will be effective immediately, valid period is duty cycle period
2-0	PTCK3[2:0]	PWM3 working clock source frequency division ratio selection bit, The clock source can be selected by PWM3_CLKS 000：1 001：2 010：4 011：8 100：16 101：32 110：64 111：128 Note: Modifying this control bit takes effect immediately. It is not recommended to modify it during output.

11.2.2 PWM3 period register PWM3P

PWM3P

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

Bit	Flag	Introductions
7-0	PWM3P[7:0]	PWM3 period register

11.2.3 PWM3 duty register PWM3D

PWM3D

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

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

11.2.4 PWM3 count clock selection register PWM3CLKS

PWM3CLKS

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	-							PWM3_CLKS

Bit	Flag	Introductions
7-1	-	Reserved Bits
0	PWM3_CLKS	PWM3 Count Clock Selection Bit 0：cpu_clk 1：Overflow rate of timer 4 When the PWM3 counting clock selects the overflow rate of timer 4, the comparison function of enabling timer 4 is required.

12 Programmable counter array PCA

12.1 PCA Characteristics

The HC89S105Ax 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.

12.2 PCA working mode

12.2.1 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.

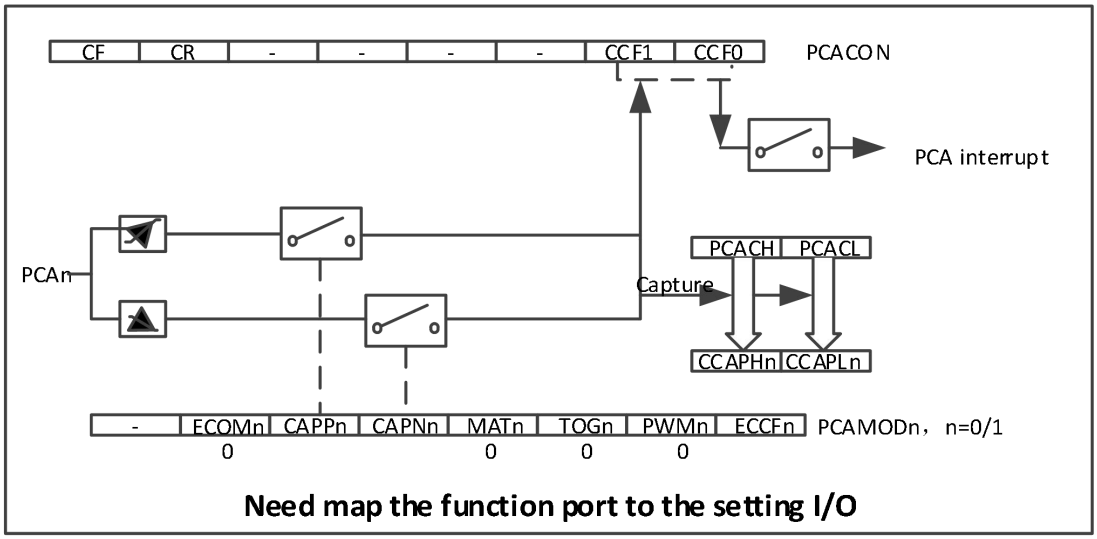


Figure 12- 1 PCA Capture the schema diagram

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.

12.2.2 Software timer mode

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

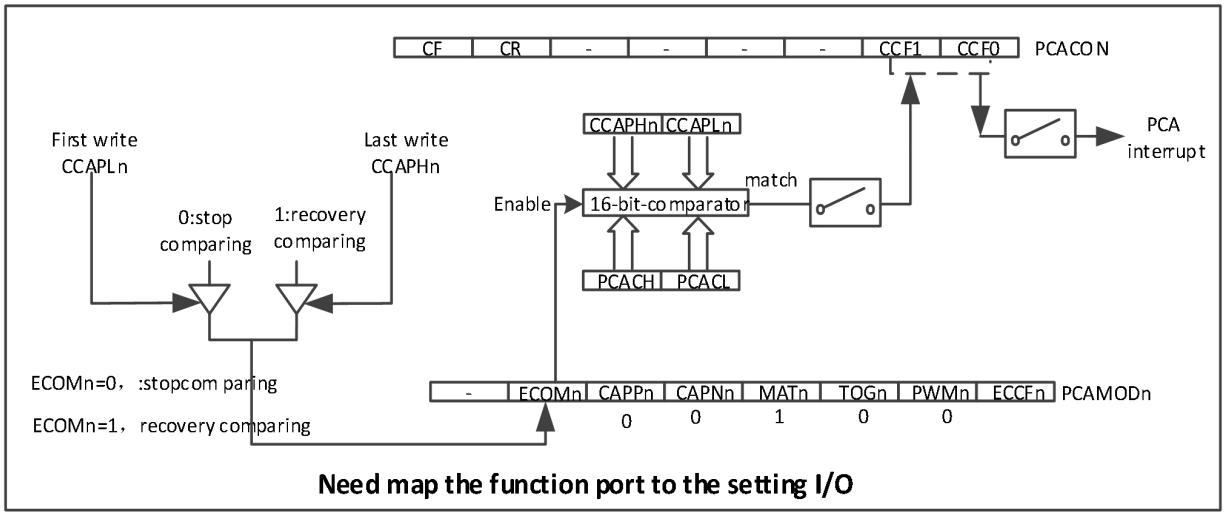


Figure 12-2 16 bit software timer mode/ PCA comparison mode for PCA module

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 HC89S105xx 71 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).

12.2.3 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.

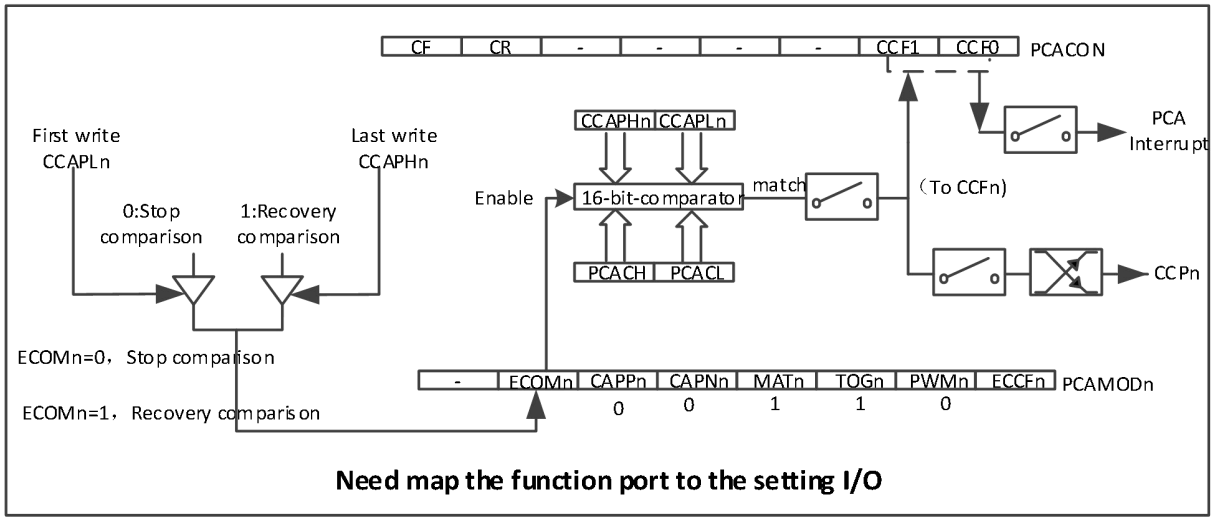


Figure 12- 3 PCA high speed output mode

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).

12.2.4 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:

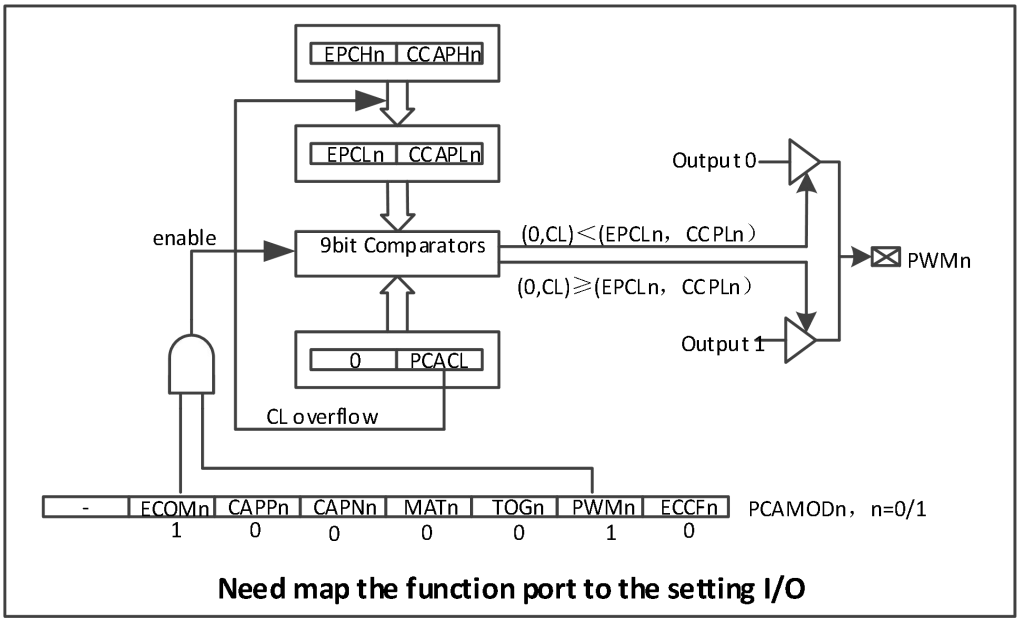


Figure 12- 4 PCA PWM mode

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.

12.3 PCA registers

12.3.1 PCA control registers

PCACON

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R	R	R	R	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	CF	CR	-				CCF1	CCF0

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

12.3.2 PCA clock register

PCACLK

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

12.3.3 PCA mode register

PCAMODn (n=0,1)

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	-	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	Comparator function control bit 0: Disable comparator function 1: Allow comparator function
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 the falling edge of PCAn/PCAn
1	0	0	1	0	0	x	16-bit software timer
1	0	0	1	1	0	x	16-bit high-speed output

12.3.4 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 high 8 bits

12.3.5 PCA capture/comparison register CCAPLn(n = 0,1)，CCAPHn (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(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(n = 0,1) capture/comparison register high 8 bits

12.3.6 PCA mode PWM register PCA_PWMn

PCA_PWMn (n = 0,1)

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

Bit	Flag	Introductions
7-2	-	Reserved Bits
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.

13 Watchdog timer WDT

13.1 WDT characteristics

- Can be configured for overflow reset
- Configurable IDLE/STOP mode enable or not
- Flexible configure overflow tim

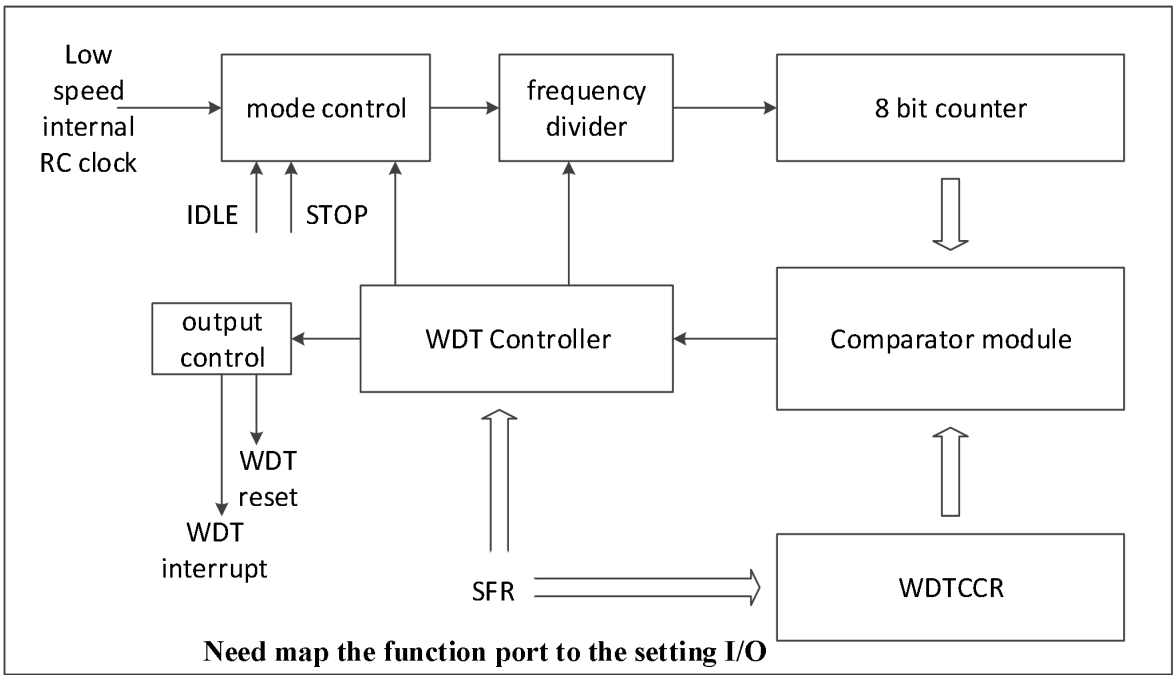


Figure 13- 1 WDT functional block diagram

HC89S105A watchdog timer is an incremental counter with a separate built in 44KHz RC oscillator as its clock source and can be registried to choose whether to run in IDLE/STOP mode. In case of WDT overflow, whether the chip is reset can also be configured through the register.

HC89S105A Watchdog timer overflow after the overflow mark, reset has a special reset mark, WDT overflow time can be set, WDT operation only need to set the corresponding control bit, flexible operation.

13.2 WDT Correlation 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 (read = 0b, write invalid)
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/STOP mode control bit 0：Enable WDT in IDLE/STOP mode, WDTRST flag needs to be enabled The shutdown mode can only be awakened if the watchdog is reset. 1：Disable WDT in IDLE/STOP mode
2-0	WDTPS[2:0]	The watchdog Timer clock source frequency division selection bits 000：8 001：16 010：32 011：64 100：128 101：256 110：512 111：1024

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.

Overflow time = WDT frequency division coefficient * (WDTCCR[7:0]+1))/44K.

WDTCCR[7:0] = 0xFF Watchdog overflow time table as below：

PS2	PS1	PS0	WDT Frequency division coefficient	WDT Maximum overflow time @44K
0	0	0	8	47 ms
0	0	1	16	93 ms
0	1	0	32	186 ms
0	1	1	64	372ms
1	0	0	128	745 ms
1	0	1	256	1489 ms
1	1	0	512	2979 ms
1	1	1	1024	5958 ms

14 Universal asynchronous transceiver UART

14.1 UART characteristics

- 2 UART incorporating baud rate generator
- baud rate generator is a 16 - bit up counter
- UART has 4 kinds of work modes
- UART adds frame error, receive overflow and write collision detection
- UART added automatic address identification

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	Baud rate is Fuart /12×6^{UX6}
0	1	1	Asynchronous	BRT overflow rate of independent baud rate generator /16
1	0	2	Asynchronous	(2^{SMOD} /64)×Fuart
1	1	3	Asynchronous	BRT overflow rate of independent baud rate generator /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. HC89S105A 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 set 1, serial port F_{osc} 1/2 Running. The only difference with Standard 8051 is that HC89S105A 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.

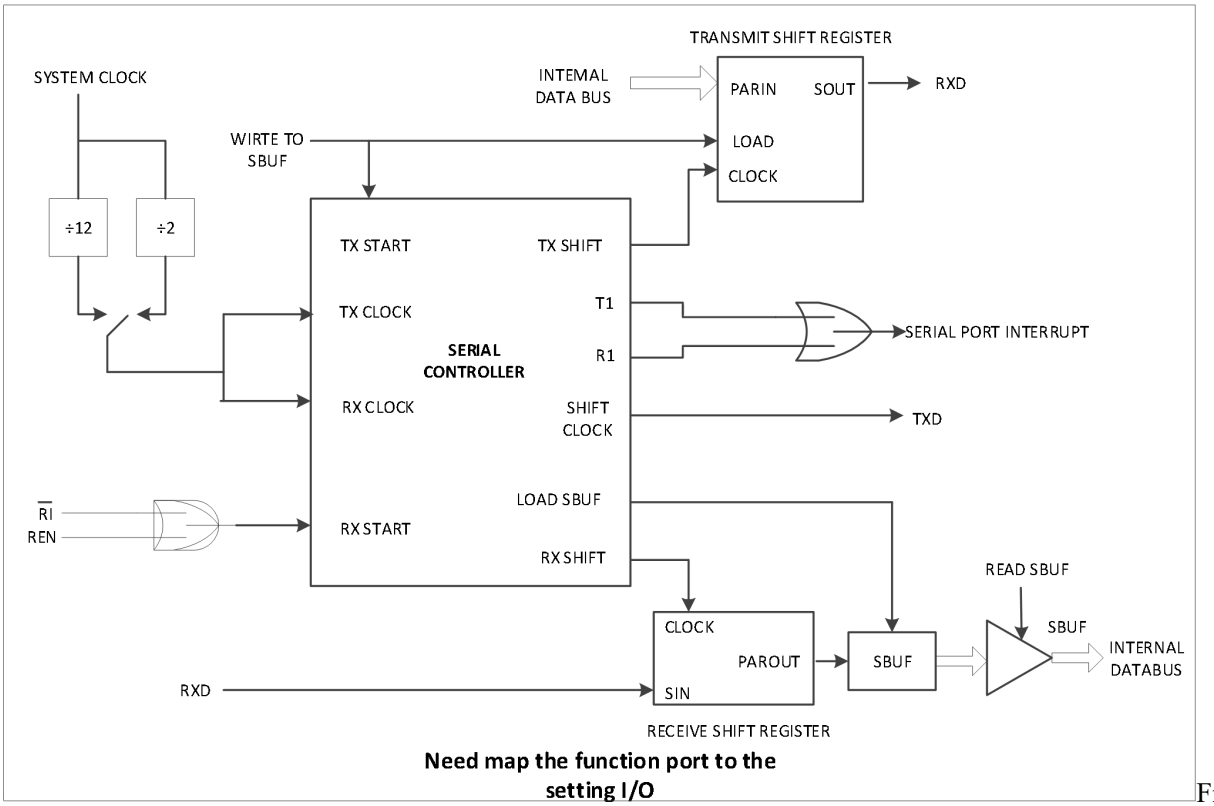


Figure 14- 1 UART mode0 function block diagram

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.

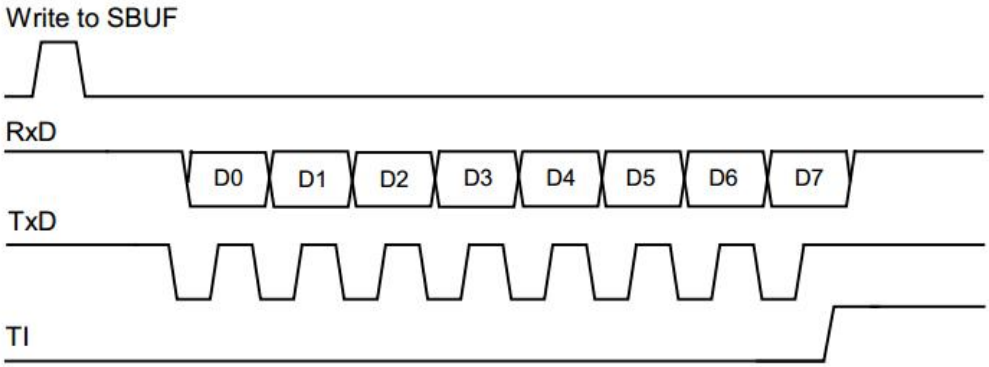


Figure 14- 2 Mode0 data send timing diagram

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.

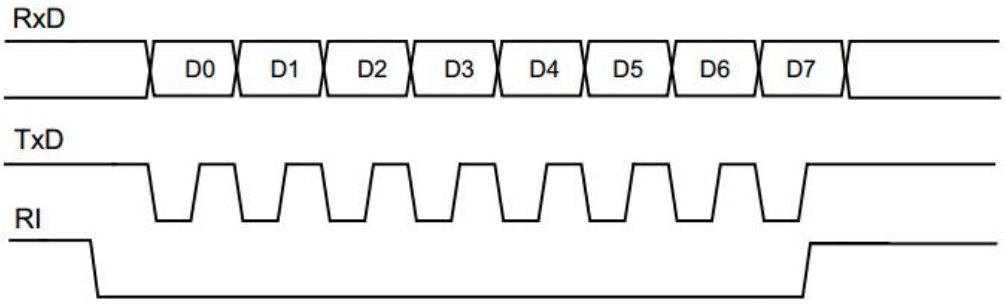


Figure 14- 3 Mode0 Data receive timing diagram

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 (logical 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:

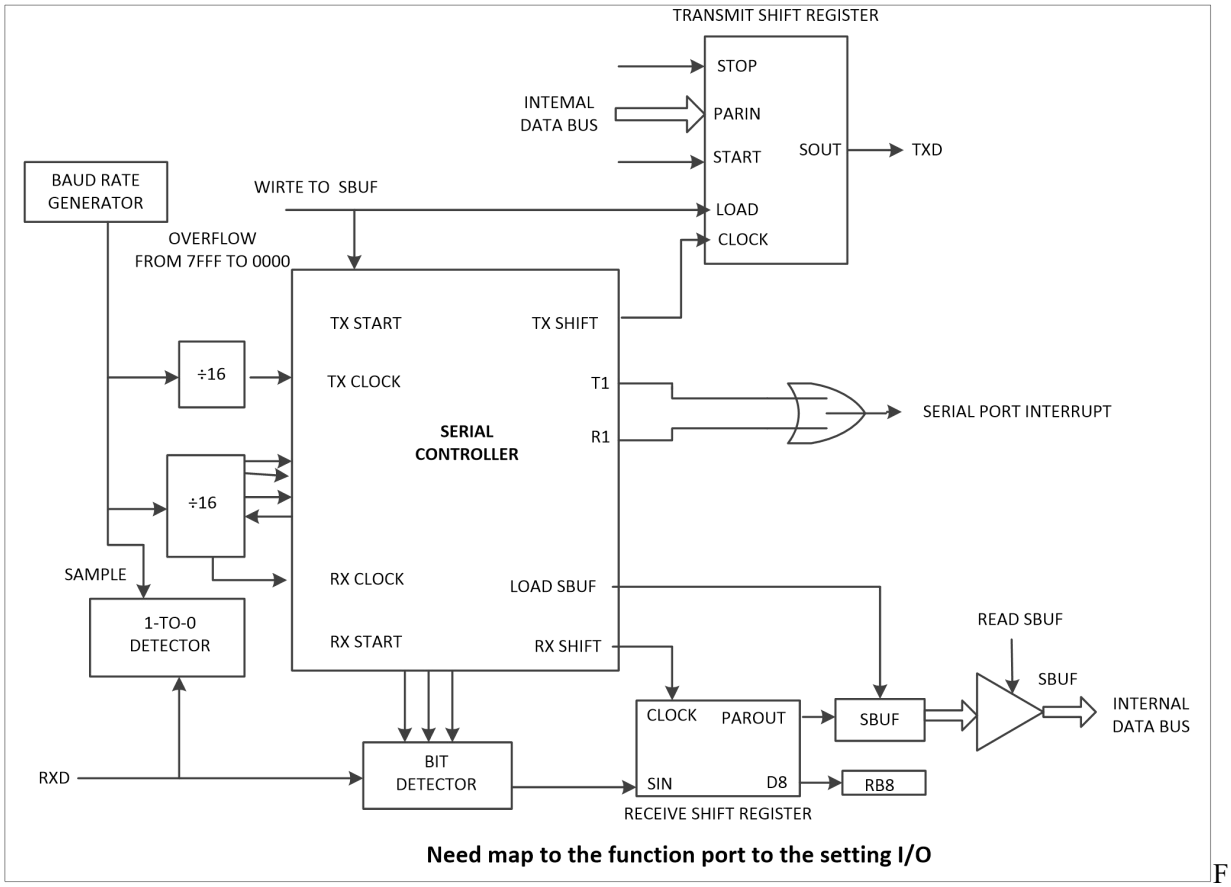


Figure 14- 4 UART mode1 functional diagram

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.

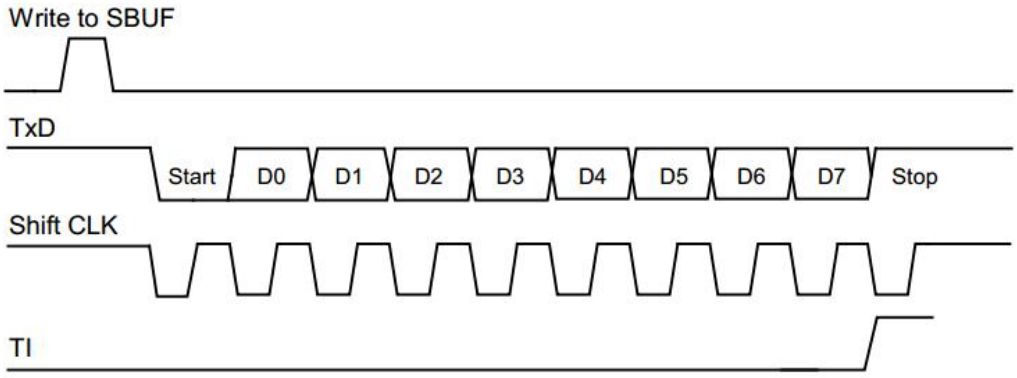


Figure 14- 5 mode1 send the data time series block diagram

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.

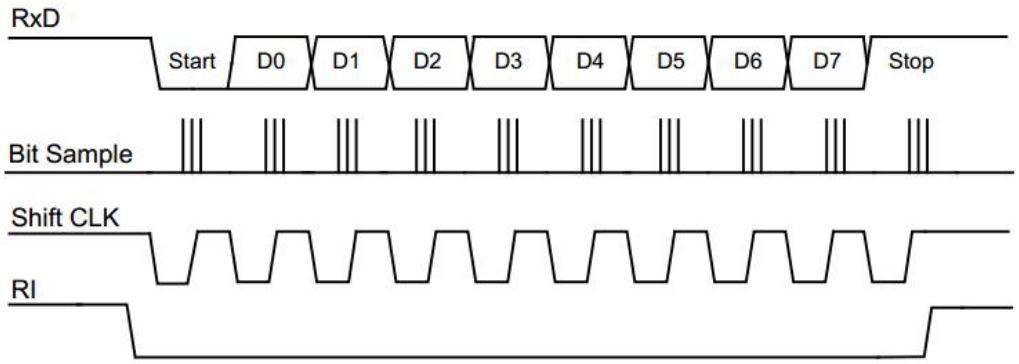


Figure 14- 6 mode1 data receive timing diagram

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.

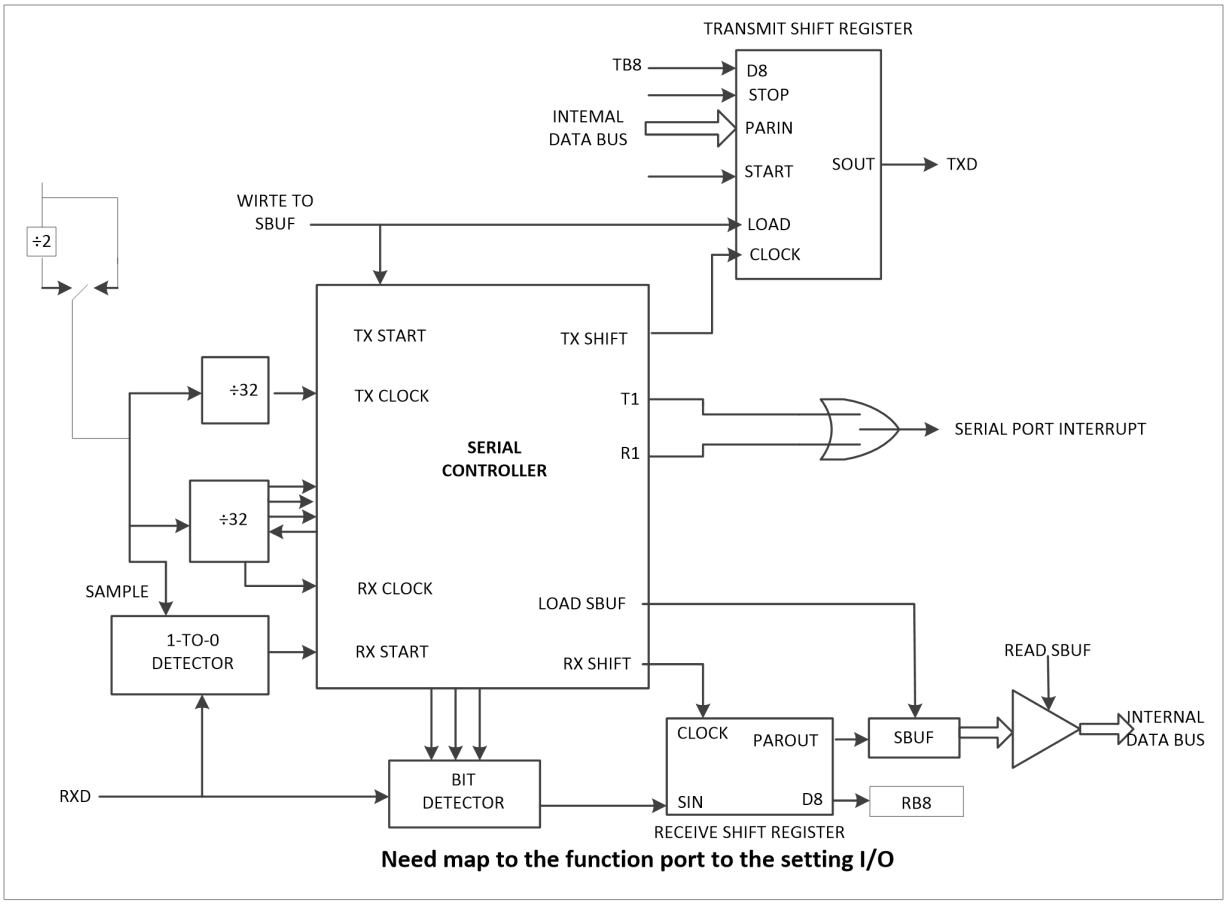


Figure 14- 7 UART mode2 functional block diagram

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.

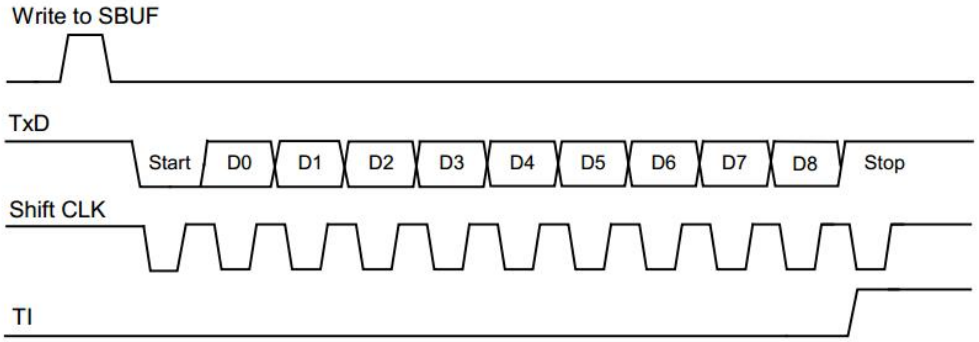


Figure 14- 8 Mode2 Send the data time series block diagram

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.

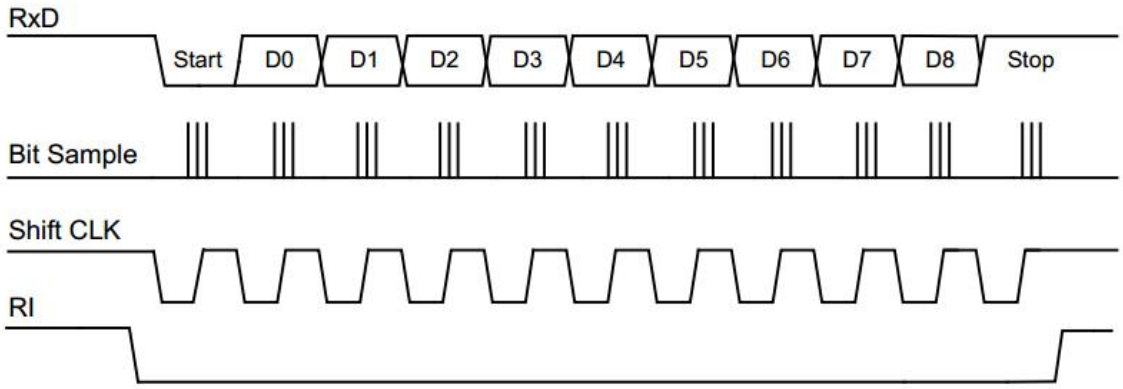


Figure 14- 9 Mode2 receive data timing diagram

14.2.4 9 bit UART, variable baud rate, asynchronous full duplex

Mode 3 uses the transmission protocol of Mode 2 and the generation mode of baud rate of mode 1.

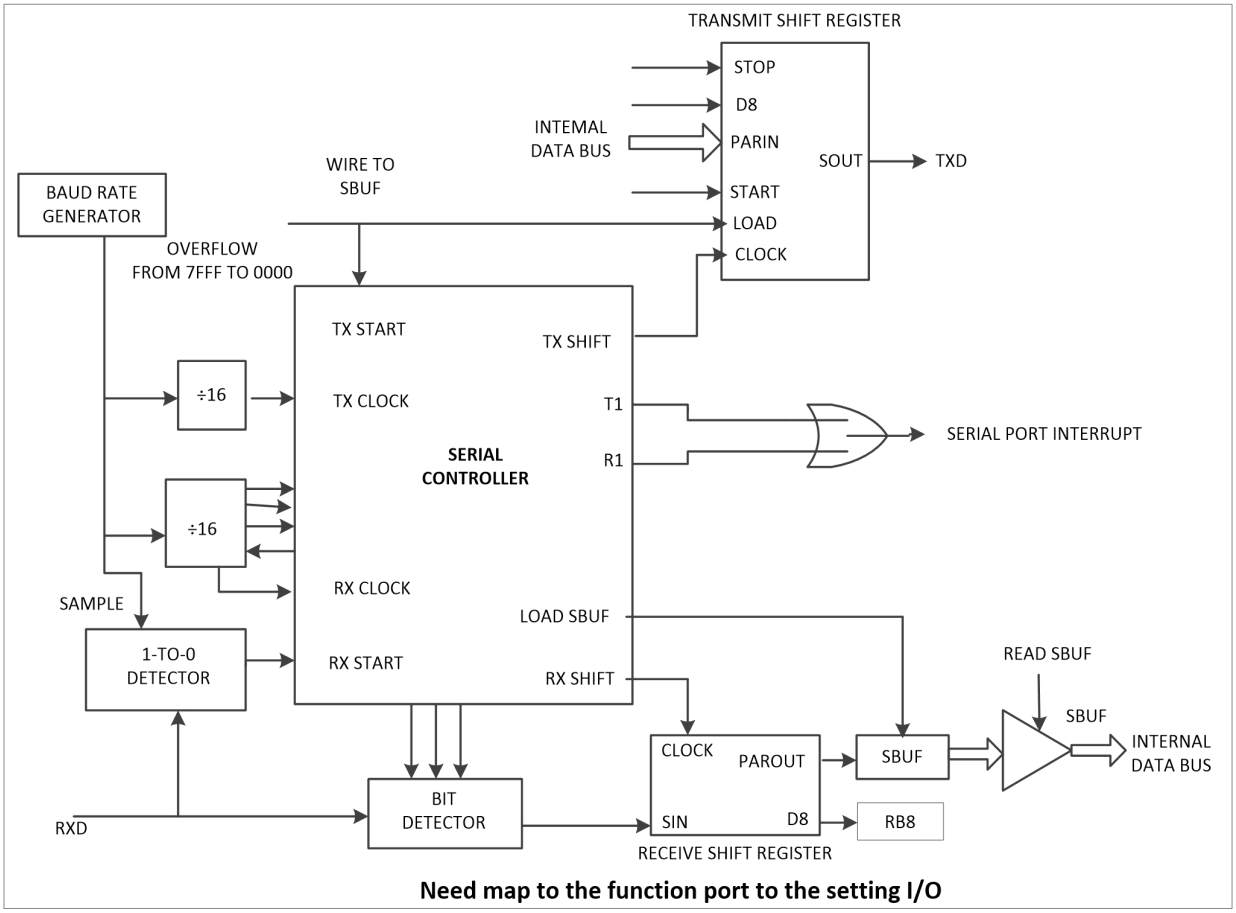


Figure 14- 10 UART mode3 function block diagram

14.3 Baud rate generator

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

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

In mode 2, the baud rate is fixed at 1/32 or 1/64 of the system clock, as determined by SMOD bits. When SMOD bit is 0, UART runs at 1/64 of Fuart. When SMOD bit is 1, UART runs at 1/32 of F_{CPU}.

The baud rate clock can be selected at 32MHz through the UART_CLKS register, so that 115200 baud rate can be used.

$$\text{Baud} = 2^{\text{SMOD}} \times \left(\frac{Fuart}{64}\right)$$

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

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

The following table shows the overload values of the baud rate generator corresponding to Fuart and common baud rate:

Commonly used baud rate	Fuart			Fosc
	4MHz	8MHz	16MHz	32MHz
1200	FF30	FE5F	FCBF	F97D
2400	FF98	FF30	FE5F	FCBF
4800	FFCC	FF98	FF30	FE5F
9600	FFE6	FFCC	FF98	FF30
19200	FFF3	FFE6	FFCC	FF98
38400	/	FFF3	FFE6	FFCC
57600	/	/	FFEF	FFDD
115200	/	/	/	FFEF

14.4 Multiprocessor communication

14.4.1 Software address recognition

Mode 2 and mode 3 are applicable to multiprocessor communication. In both cases, nine bits of data are received, and the ninth bit is moved into RB8, followed by the stop bit. The UART can be set so that when the stop bit is received and RB8 = 1, the serial port interrupt is valid (request flag RI is set). In this case, SM2 bit is set, and UART works in multi-machine 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: The stop bit is received, bit 9 of RB8 is 1 (address byte), and the received data byte matches the slave address of the UART, and the UART generates an interrupt. Slave clear SM2 to 0, the data bytes received subsequently.

Bit 9 being 1 indicates that the byte is an address rather than 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 slave 1 and 2 address lowest bit is different. Slave 1 the lowest bit is ignored, and slave 2 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, slave 2 lowest bit is ignored. Therefore, only slave 2 in communication, the host must send the lowest bit is 1 as address (10100011). If the host will need to communicate with the two slaves, the bit 0 equal 1, bit1 equal 0, bit 2 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

When the two error flag bits are set, they can only be cleared by software, although subsequent frames received have no errors and will not be cleared automatically.

14.5.1 Send conflict

If the user software writes data to the SBUF register while a data send is in progress, the send conflict bit (TXCOL bit) is set to 1. If a conflict occurs, the new data is ignored and cannot be written to the send buffer (without affecting transmission).

14.5.2 Receive overflow

RI is set to 1, the data in the receiving buffer is not read, RI is cleared to 0, and new data is received. If the data in the receiving buffer is not read before the new data is received (RI is set to 1), then the receive overflow bit (RXROV bit) is set. If a receive overflow occurs, the original data in the receive buffer is not affected and subsequent data is lost.

14.5.3 Frame error

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

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	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/W	R/W	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 Fuart 1/64 1 : In mode2, the baud rate = system clock Fuart 1/32
6	-	Reserved (read = 0b, write invalid)
5	UX6	Serial port mode0 communication speed bit 0 : Serial port mode0 Clock = Fuart/12 1 : Serial port mode0 Clock = Fuart/2
4	BRTR	Independent baud rate generator BRT operation control bit 0: Stops the independent baud rate generator BRT 1: Start independent baud rate generator BRT work
3	BRTOUT	Independent baud rate generator BRT output enable bit 0: disables clock output of independent baud rate generator BRT 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

UART Mode of work selection table：

SM0	SM1	mode	功能Introductions	波特率
0	0	0	Synchronous shift transfer serial mode: shift register	When UX6 = 0, the baud rate is Fuart /12 When UX6 = 1, the baud rate is Fuart /2
0	1	1	8 bit UART, variable baud rate	BRT overflow rate of independent baud rate generator /16
1	0	2	9 bit UART	$(2^{SMOD}/64)\times$Fuart
1	1	3	9 bit UART, variable baud rate	BRT overflow rate of 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 There are actually two separate registers on this address, one for receiving data and one for sending data. When data is written to SBUF, this is a send register and is shifted for serial transmission. When data is read from SBUF, this is a receive register.

14.6.3 UART1 independent baud rate generator register SBRTL，SBRTH

SBRTL

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	SBRTL[7:0]							

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

SBRTH

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	SBRTH[7:0]							

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

Note: to modify SBRTL & SBRTH, modify the high SBRTH first, and then modify the low SBRTL.

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 registers

UART2 controls and works in basically the same way as UART1.
Differences:

1. UART2 registers deposited in the extended SFR;
2. URAT2 baud rate clock cannot output.

14.7.1 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 Note: Frame errors refer to invalid stop bits
6	RXROV	Receive overflow flag bit 0: no receive overflow or software clearing 0 1: Receive overflow, hardware is set to 1
5	TXCOL	Sends conflicting flag bits 0: no sending conflict or software clearing 0 1: send conflict, set hardware to 1
4	REN	Serial receive enable control bit 0: Disable serial receive 1: Enables serial receive
3	TB8	In mode 2 or mode 3, the software sets 1 or zeros for the 9th bit of data to be sent
2	RB8	In mode 2 or mode 3, for the 9th bit of data received as the parity bit or address frame/data frame flag bit
1	TI	send interrupt request flags bit 0: Software clear 0 1: In mode 0, the hardware automatically sets 1 when the 8th bit of serial data transmission ends, and in other modes, the hardware sets 1 when the stop bit begins to send
0	RI	receive interrupt request flags bit 0: Software clear 0 1: In mode 0, the hardware will set 1 automatically when the 8th bit of the serial received data ends. In other modes, the hardware will set 1 when the serial received stop bit begins

S2CON2

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R	R/W	R/W	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 Fuart 1/64 1 : In mode2, the baud rate = system clock Fuart 1/32
6	-	Reserved (read = 0b, write invalid)
5	UX6	Serial port mode0 communication speed bit 0 : Serial port mode0 Clock = Fuart/12 1 : Serial port mode0 Clock = Fuart/2
4	BRTR	Independent baud rate generator BRT operation control bit 0: Stops the independent baud rate generator BRT 1: Start independent baud rate generator BRT work
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 mode 1, stop bit validation is disabled and any stop bit is set RI In mode 2 or in mode 3, any byte is set to RI 1：In mode 1, stop bit confirmation checks are allowed, and only valid stop bit "1" can be set to RI ;In mode 2 or in mode 3, only the address byte (bit 9 = "1") can be set to RI

SM0	SM1	Mode	Introductions	波特率
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	BRT overflow rate of independent baud rate generator /16
1	0	2	9 bit UART	$(2^{SMOD}/64)\times F_{CPU}$
1	1	3	9 bit UART, variable baud rate	BRT overflow rate of independent baud rate generator /16

14.7.2 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 There are actually two separate registers on this address, one for receiving data and one for sending data. When data is written to S2BUF, this is a send register and is shifted for serial transmission. When data is read from S2BUF, this is a receive register.

14.7.3 UART2 independent baud rate generator register S2BRTL，S2BRTH

S2BRTL

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	S2BRTL[7:0]							

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

S2BRTH

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	S2BRTH[7:0]							

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

Note: to modify S2BRTL & S2BRTH, modify the high S2BRTH first, and then modify the low S2BRTL.

14.7.4 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
- 4 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

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.

master device driver MOSI and SCK.

The following conditions, (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 (SS) pin as the communication goal.

When the slave device (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 (SS) pin is enabled, If (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 (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

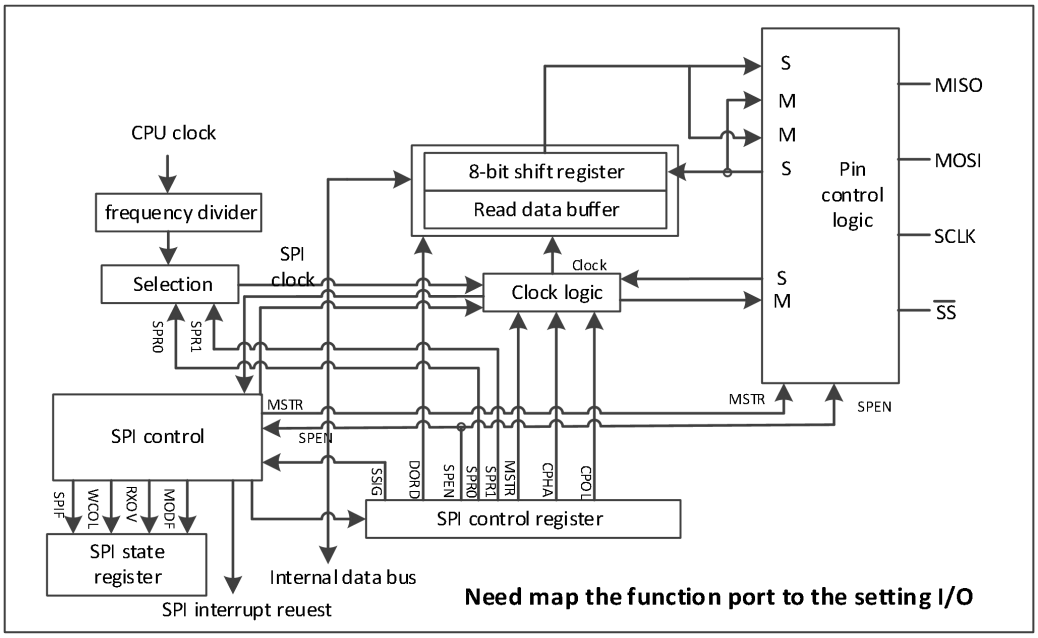


Figure 15- 1 SPI functional block diagram

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.

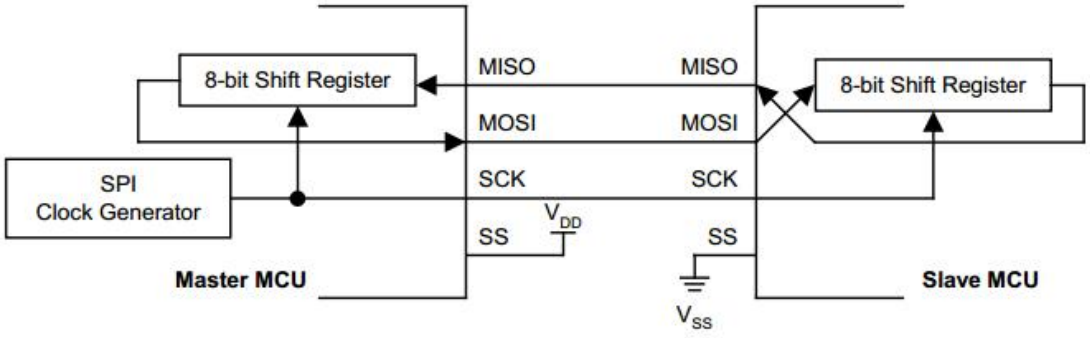


Figure 15- 2 Full-duplex master/slave interconnect diagram

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. This SPI module receives as a double buffer. Data can be read after SPIF is set to 1. In case of receive overflow, subsequent data will not be moved to the receive register. In case of receive overflow, SPIF can be set to 1 normally.

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.

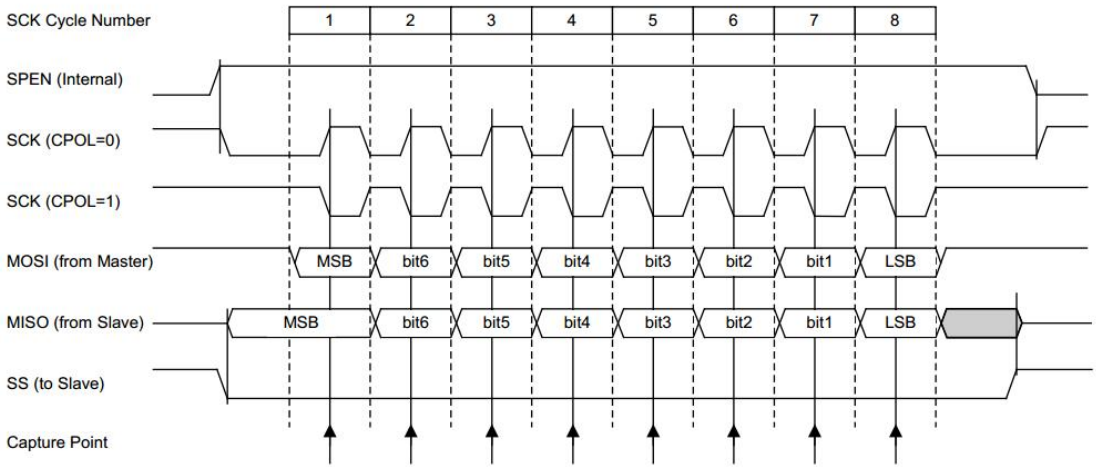


Figure 15- 3 Data transfer form（CPHA=0）

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.

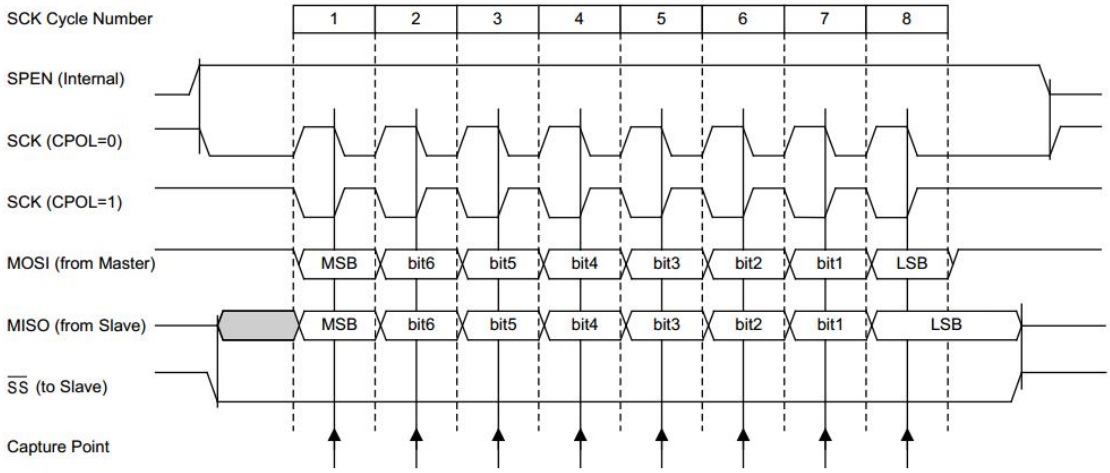


Figure 15- 4 Send form data（CPHA=1）

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.

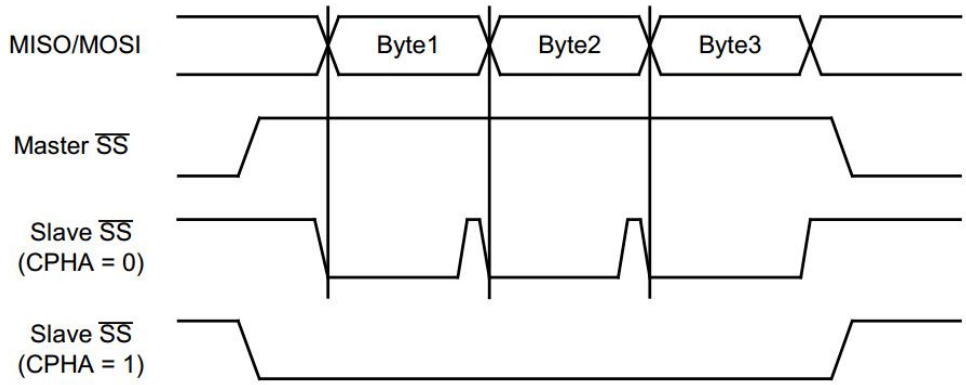


Figure 15- 5 CPHA/SS timing sequence

15.7 SPI Error detection

Continuing to write to SPDAT before or during data transmission will cause write collisions, and the WCOL bit will be set to 1, but the transmission will not stop, requiring software to 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.

Mode conflict: When the SS pins of the master device are enabled, the SS is pulled low, and contention for the bus occurs. 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 slave. Therefore, the user software must always detect the MSTR bit, and if this is cleared by a slave selection and the user wants to continue using SPI as the host, the MSTR must be reset, otherwise it goes into slave mode.

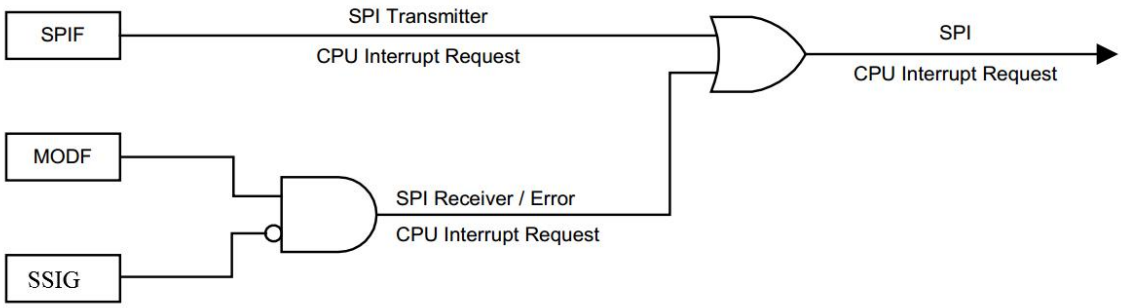


Figure 15- 6 SPI interrupt request generation

15.9 SPI configuration table

SPEN	SSIG	$\overline{SS}$	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：Fper/4 01：Fper/16 10：Fper/64 11：Fper/128

15.10.2 SPI state register SPSTAT

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

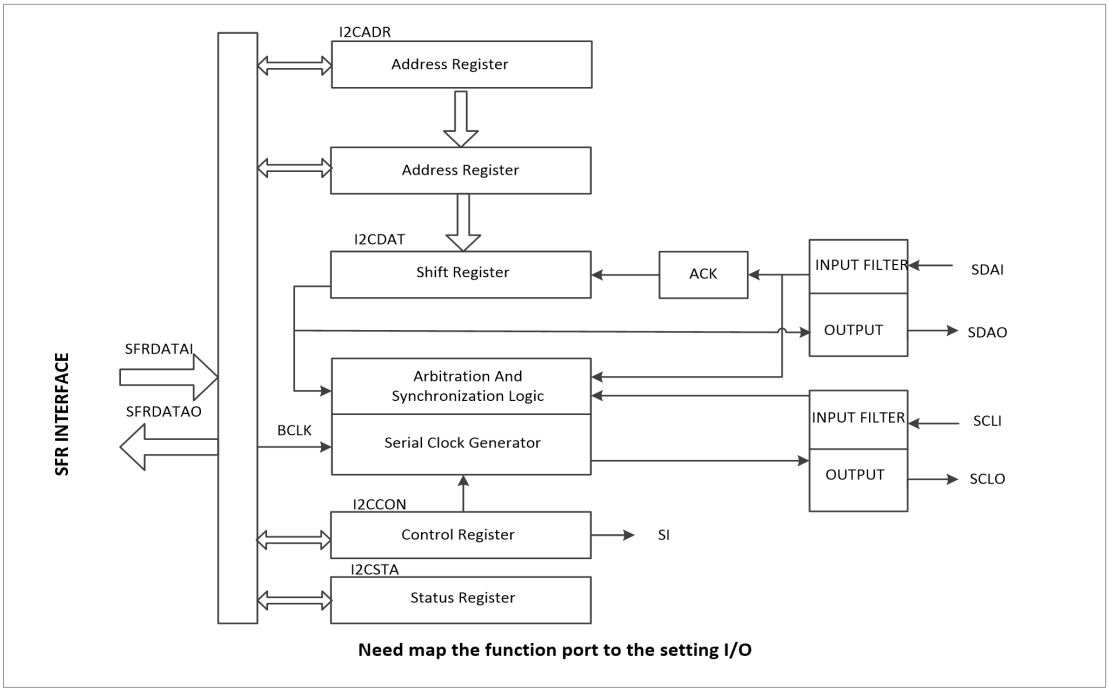


Figure 16- 1 IIC function diagram

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.

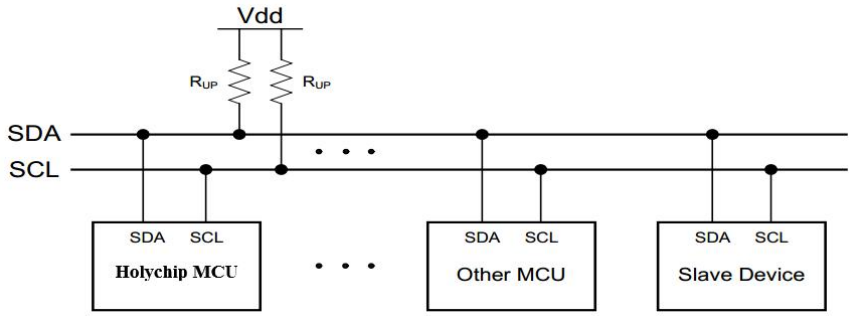


Figure 16- 2 IIC bus connection diagram

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:

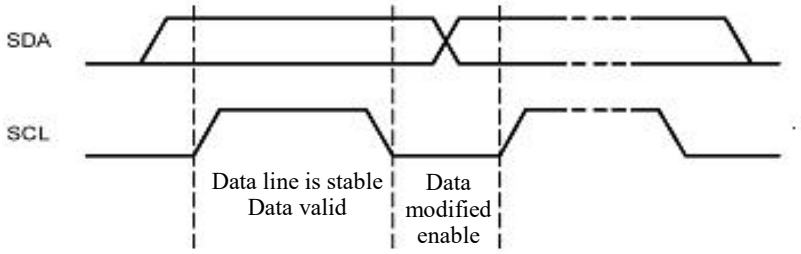


Figure 16- 3 IIC bus data availability

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.

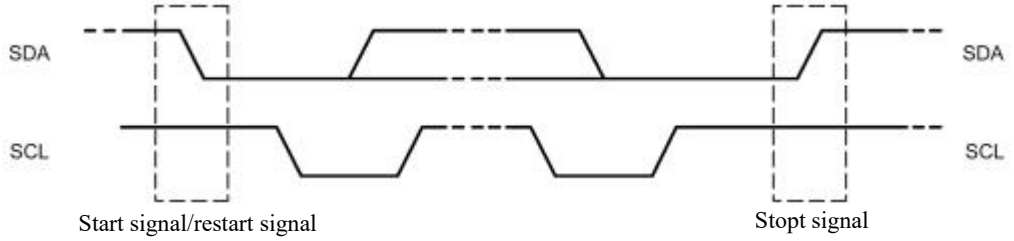


Figure 16- 4 Start, restart, stop signal

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.

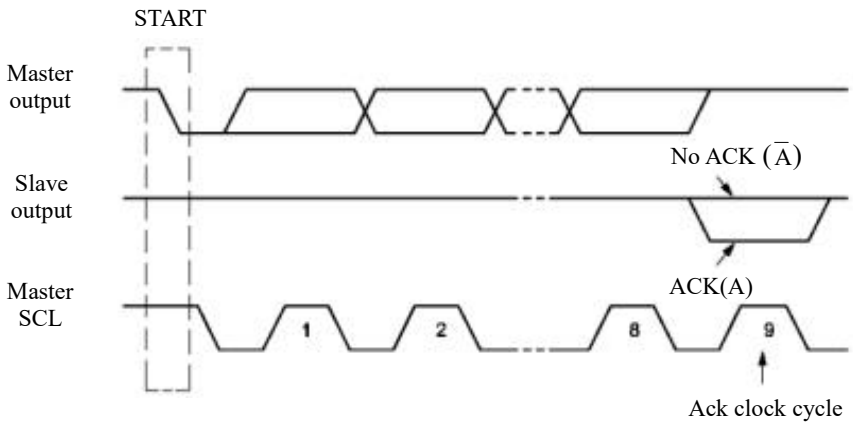


Figure 16- 5 IIC bus acknowledge signal

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.

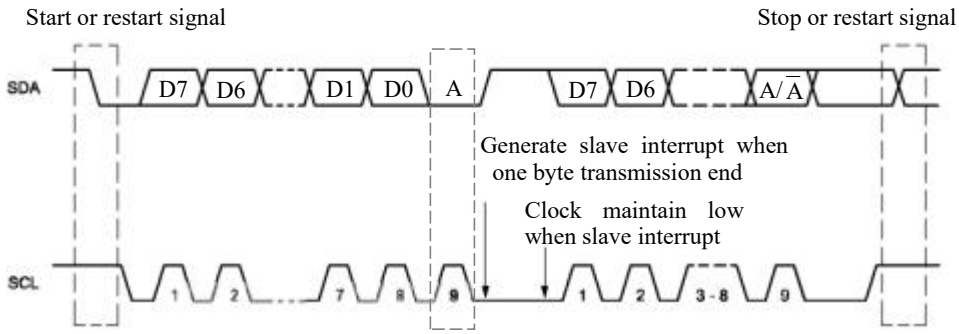


Figure 16- 6 IIC Bus data transmission format

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.

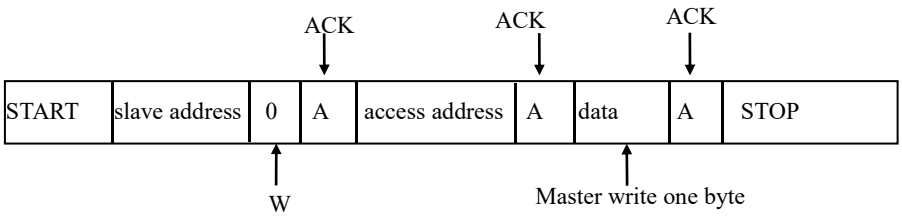


Figure 16- 7 Master write one byte data to slave

As Figure 16-8 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.

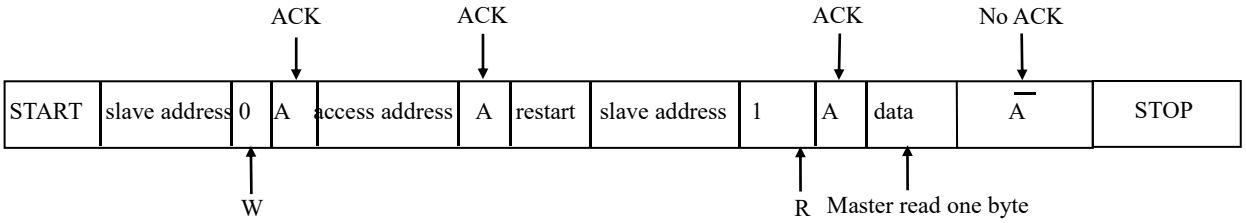


Figure 16- 8 Master read one byte data from slave

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.

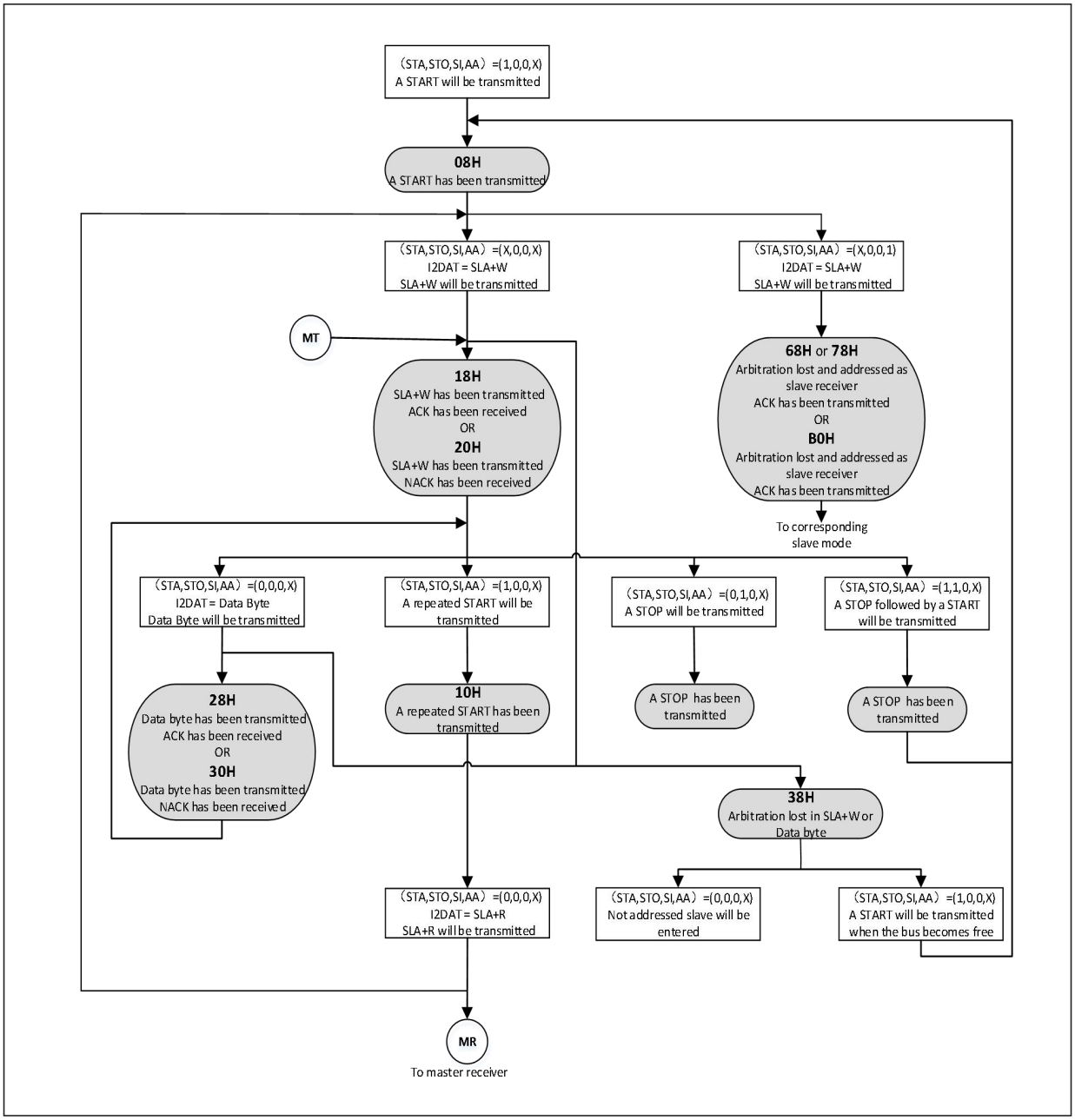


Figure 16- 9 Master send mode flow and status

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.

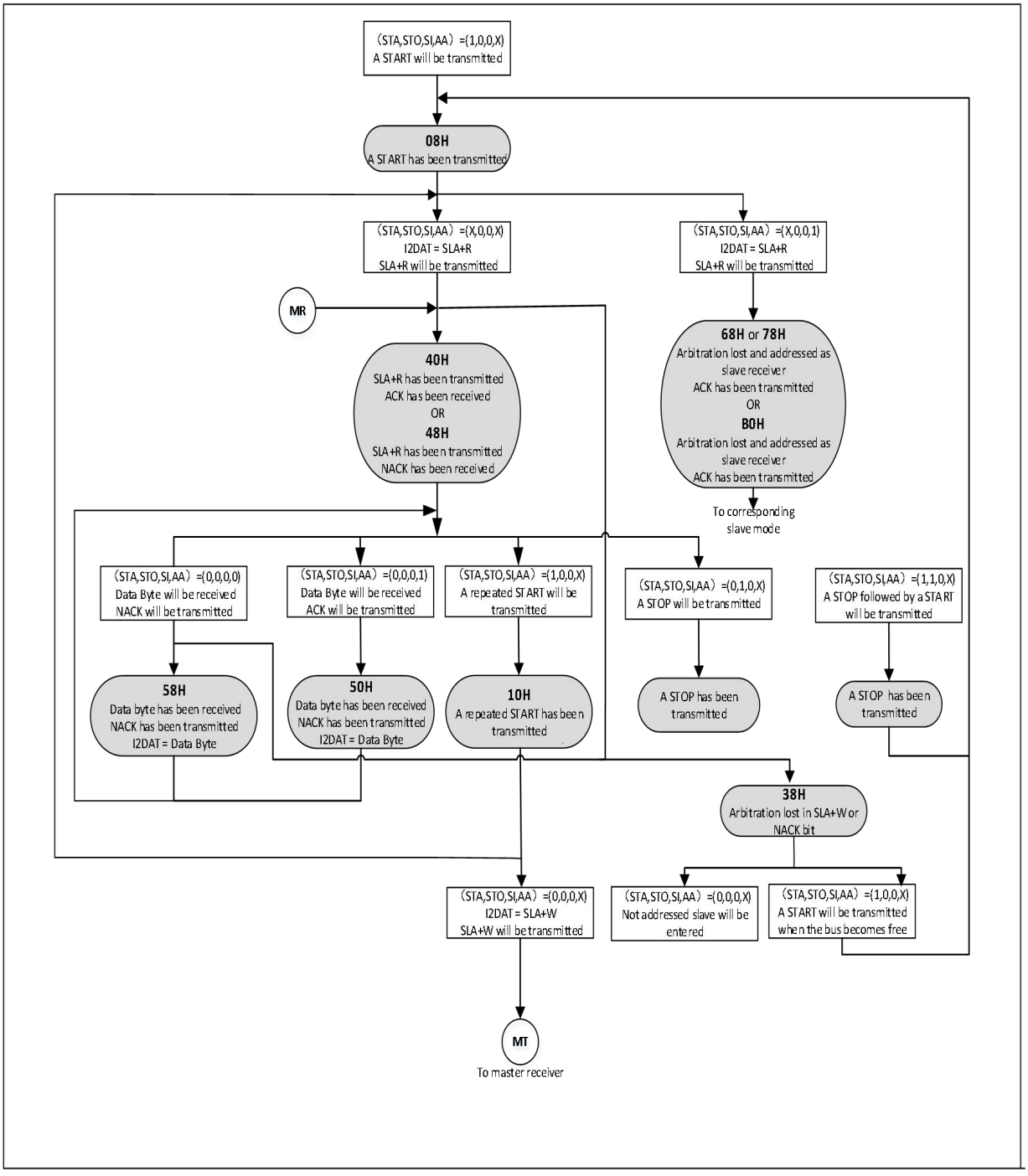


Figure 16- 10 Master receive mode flow and status

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.

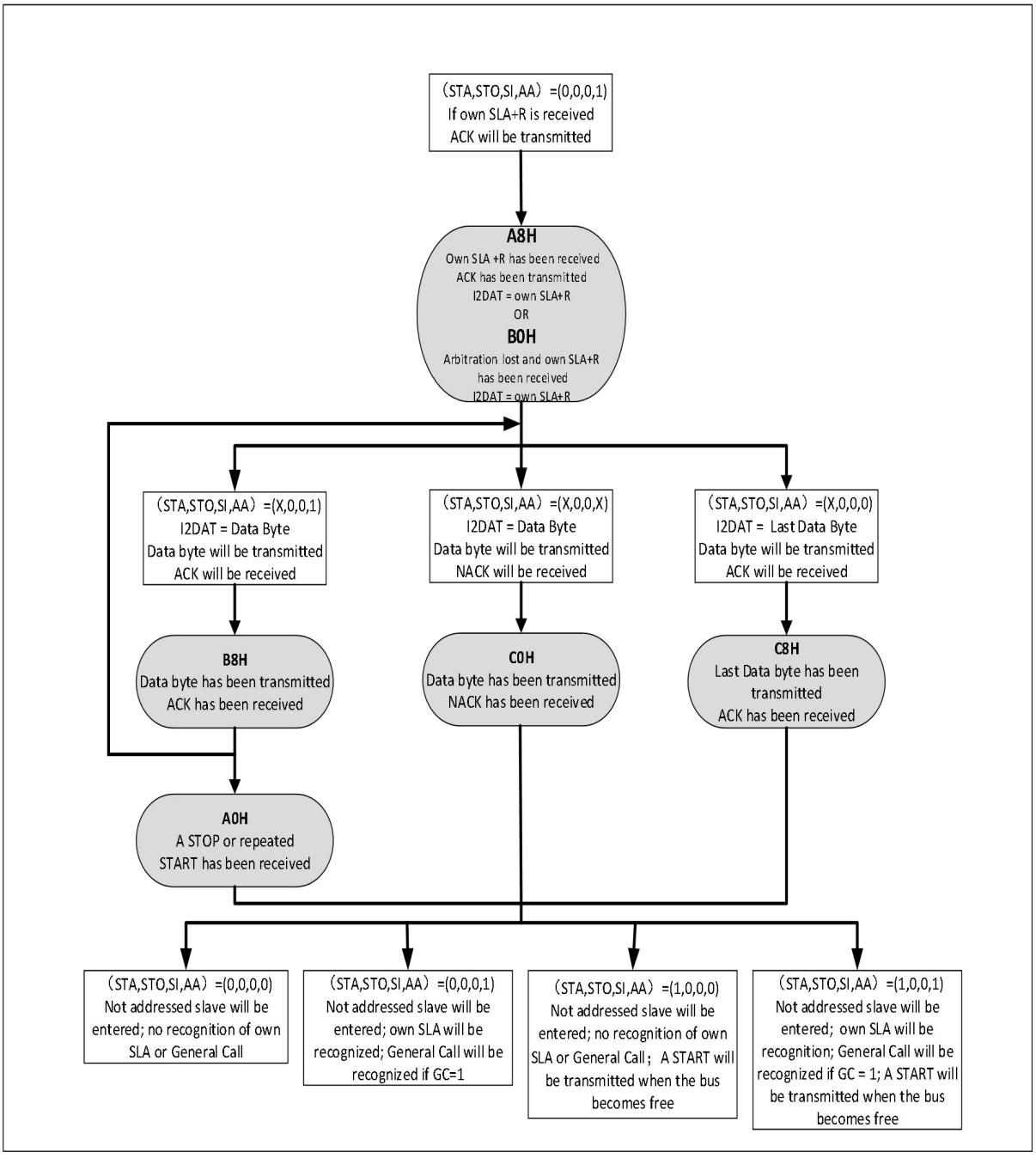


Figure 16- 11 Slave send mode flow and status

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.

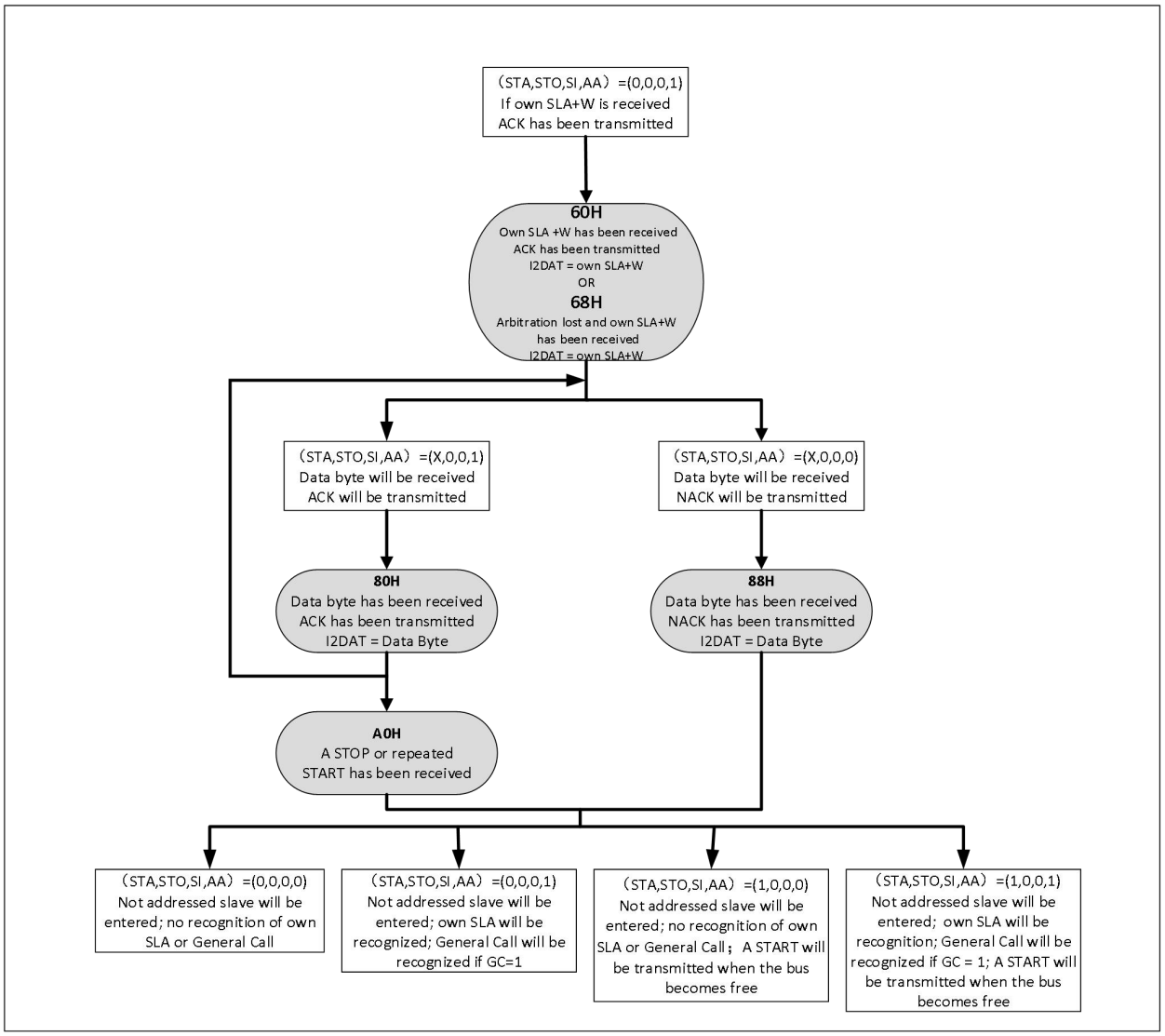


Figure 16- 12 Slave receive mode flow and status

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.

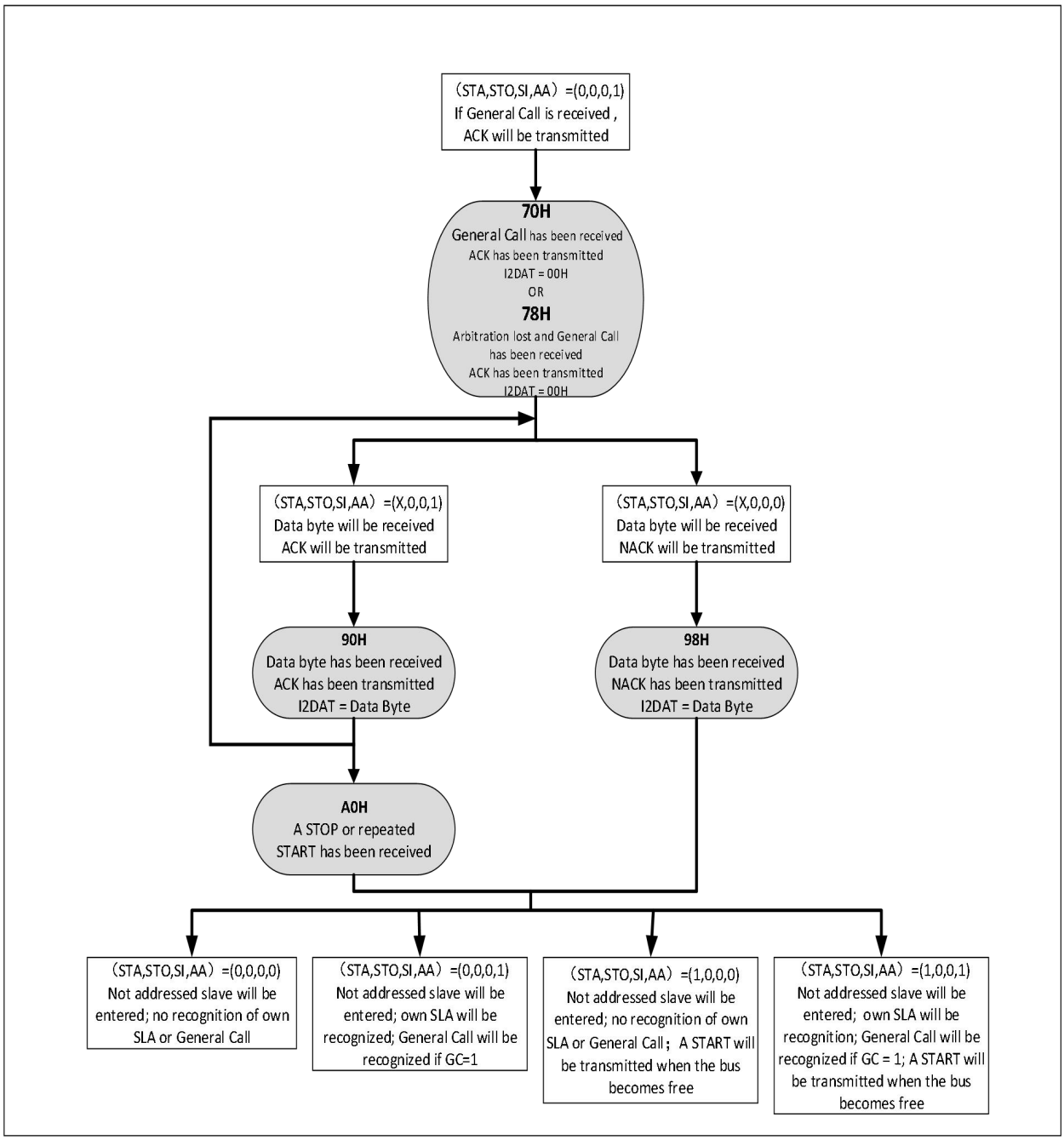


Figure 16- 13 Broadcast call mode flow and status

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 IIC control 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	Fper				frequency dividing coefficients
			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	UART2 BRT independent baud rate generator overflow rate divided by 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 Bits

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.

16.9.4 IIC 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 32 external channels and 2 internal channels (include GND) 12 bits or 10 bits ADC detection
- Optional internal reference voltage 2V,3V,4V,VDD and external Vref
- Optional convert data align orientation
- Optional convert data bit
- Analog-to-digital Conversion complete interrupt
- Analog watchdog
- Multiple trigger modes
- Analog-to-digital converter continuous conversion

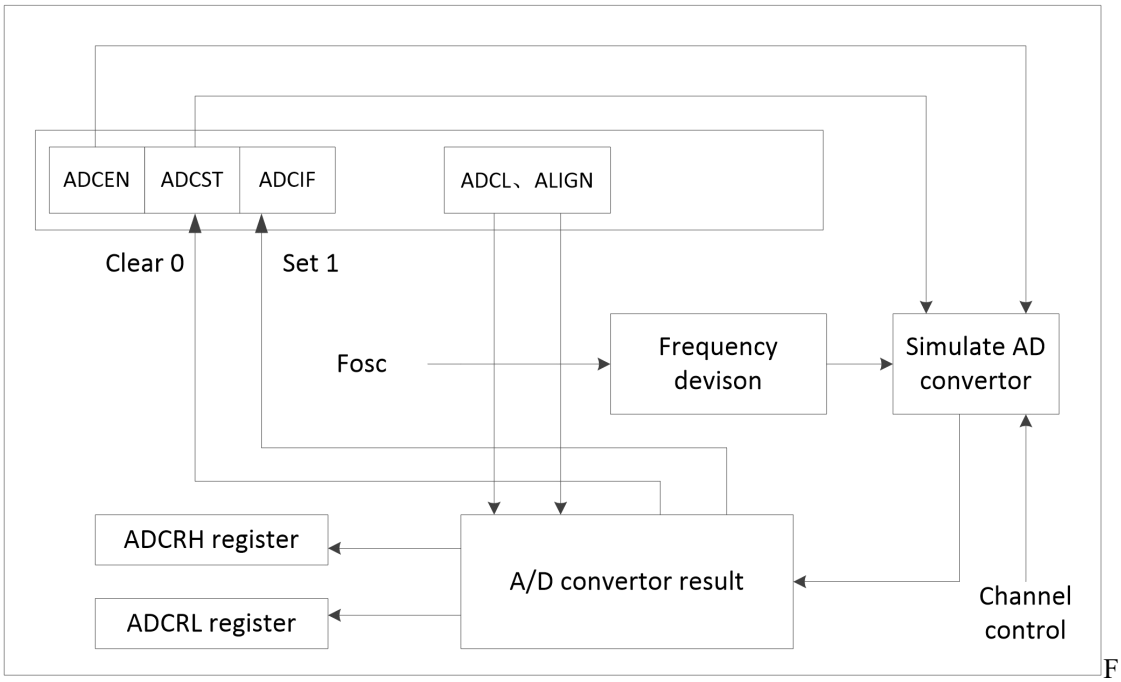


Figure 17- 17 functional block diagram

17.2 ADC registers

17.2.1 ADC control register ADCC0，ADCC1

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	0	0
Flag	ADCEN	ADCST	ADCIF	-	VREFO	VREFS	INREF_S[1:0]	

Bit	Flag	Introductions
7	ADCEN	ADC module power control bit 0 : Close ADC conversion power 1 : Open ADC conversion power Note：1. In STOP mode, ADCEN force to 0. 2. ADCEN set 1 or after switch conversion channel, recommended start ADC Conversion after delay 20us.
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) Note: software clear ADCIF and ADCST set to 1 cannot be operated at the same time
4	-	Reserved (read = 0b, write invalid)
3	VREFO	VREF output enable bit 0 : VREF no output 1： From P2.5 output VREF, this moment P2.5 mast be set analog input, and VREFS must be 0 注：VREF has a weak output driver and is only used for testing.
2	VREFS	VREF selection bit 0: Select internal VREF 1: Select external VREF(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 or3V or 4V, VDD must above the internal Vref 0.5V. Before the system enters the STOP mode, it is recommended to set the ADC reference voltage to non-VDD to further reduce the power consumption of the system.

ADCC1

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

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[4:0] 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[4:0]	ADC external input channel selection bits XCHS[4:0] = x(x = 0…31),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.

17.2.2 ADC control register ADCC2，ADCC3

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 10 bit data (get 12 bits high 10 bits)
6	ALIGN	ADC data alignment direction control bit, Refer to the ADC conversion data format description table below(Introductions)
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: 1. When the reference voltage is internal 2V, 3V, 4V,in order to ensure ADC accuracy, recommended ADC conversion frequency at 2MHz or below 2MHz.

		2. When the reference voltage is VDD, the ADC clock can be 8MHz, and only 15 ADC_CLK are required for one conversion, thus achieving the fastest ADC conversion speed.
2-0	ADCS[2:0]	ADC clock selection bi 000：Fper/2 001：Fper/4 010：Fper/6 011：Fper/8 100：Fper/12 101：Fper/16 110：Fper/24 111：Fper/32

ADC conversion data format description 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

ADCC3

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	-	FCLKEN	ADCST_OEN	TRIGSEL[4:0]				

Bit	Flag	Introductions
7	-	Reserved Bits
6	FCLKEN	Acceleration of ADC conversion when internal reference voltage is selected 0: An ADC with internal parameters can operate at a maximum frequency of 2MHz 1: An ADC with internal parameters can work at a frequency of up to 4MHz
5	ADCST_OEN	ADCST output enable bit 0：ADCST signal is disabled from P0.1 output 1：ADCST signal output from P0.1 is enabled
4-0	TRIGSEL[4:0]	ADC trigger signal selection bit 00000: ADC conversion start is controlled by ADCST(ADCC0.6) only 00001: PWM0 rising edge 00010: PWM0 falling edge 00011: PWM0 midpoint 00100: PWM0 terminal

		00101: PWM1 rising edge 00110: PWM1 falling edge 00111: PWM1 midpoint 01000: PWM1 terminal 01001: PWM2 rising edge 01010: PWM2 falling edge 01011: PWM2 midpoint 01100: PWM2 terminal 01101: PWM0 matching interrupt 01110: ADCST pin rising edge 01111: ADCST pin drop edge 10000:ADCST pin double edge 10001: An input capture event occurred on the T5 channel of timer 5 10010: An input capture event occurs on input capture channel 0 of timer 5 10011: An input capture event occurs on input capture channel 1 of timer 5 10100: A comparison match event occurs on timer 5 Other values: reserved Note: 1.PWM midpoint trigger is only applicable to center alignment mode PWM output 2. When ADCST is 1 (under conversion), the external trigger signal will not affect the ADC until the ADCST is cleared by hardware at the end of this ADC conversion 3.When using PWMx endpoint trigger mode, the ADC triggers the delay timer to write a value other than zero

17.2.3 ADC Trigger delay timer ADCDLYH，ADCDLYL

ADCDLYH

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	-				ADCDLY[11:8]			

Bit	Flag	Introductions
7-4	-	Reserved Bits
3-0	ADCDLY[11:8]	ADC external triggers the high 4 bits of the delay start timer

ADCDLYL

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	ADCDLY[7:0]

Bit	Flag	Introductions
7-0	ADCDLY[7:0]	The lower 8 bits of the ADC External Trigger Delay start timer, used to insert a delay before the external trigger starts the ADC, and to start the ADC conversion at the end of the delay timer Delay Time = Adcdly [11:0]* ADC Clock

17.2.4 ADC analog watchdog control register AWDCON

AWDCON

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R	R/W	R/W	R/W	R/W	R/W
Reset values	0	0	0	0	0	0	0	0
Flag	-			CONT	AWDIF	AWDIE	AWDMOD	AWDEN

Bit	Flag	Introductions
7-5	-	Reserved Bits
4	CONT	ADC continuous conversion enabled Configure the ADC channel through ADCC1 and configure CONT=1. The first ADC conversion requires software to start, and after the completion of ADC, ADCIF will be set. The software reads the results through query or interrupt, and the ADCIF will be cleared to zero at the same time, and the hardware will automatically start the next conversion.Until the conversion count reaches the register configured by the user or the user zeros CONT or turns off ADCE
3	AWDIF	Analog watchdog flag bit This bit is set by the hardware according to the AWDMOD bit and cleared by the software 0: There was no analog watchdog incident 1: The occurrence of analog watchdog event
2	AWDIE	Analog watchdog interrupt enable bit This bit is set and cleared by software to disable or allow a simulated watchdog. 0: Disallow simulated watchdog interrupts 1: Allows simulated watchdog interrupts.
1	AWDMOD	Analog watchdog mode selection bit 0: If the analog voltage being converted by the ADC is below the low threshold or above the high threshold, AWDIF is set 1: If the analog voltage converted by ADC is higher than the low threshold and lower than the high threshold, AWDIF is set
0	AWDEN	Awden: Turn on the analog watchdog in the ADC channel This bit is set and cleared by the software. 0: Disable analog watchdog 1: Use a simulated watchdog

17.2.5 ADC high threshold comparison register ADC_HTRH，ADC_HTRL

ADC_HTRH

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	-				HTR[11:8]			

Bit	Flag	Introductions
7-4	-	Reserved Bits
3-0	HTR[11:8]	ADC high threshold is 4 bits higher than the comparison register

ADC_HTRL

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	HTR[7:0]							

Bit	Flag	Introductions
7-0	HTR[7:0]	ADC high threshold compares the register to 8 bits lower

17.2.6 ADC low threshold comparison register ADC_LTRH，ADC_LTRL

ADC_LTRH

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	-				LTR[11:8]			

Bit	Flag	Introductions
7-4	-	Reserved Bits
3-0	LTR[11:8]	ADC low threshold Compares the register 4 bits higher

ADC_LTRL

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	LTR[7:0]							

Bit	Flag	Introductions
7-0	LTR[7:0]	ADC low threshold Compares the register to 8 bits lower

17.2.7 ADC continuous conversion frequency register ADCCONTV

ADCCONTV

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	CONTV							

Bit	Flag	Introductions
7-0	CONTV	ADC continuous conversion times When configured to 0, convert mode continuously as long as CONT and ADCEN are not closed

17.2.8 ADC convert interval registers continuously ADCGAPV

ADCGAPV

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	GAPV							

Bit	Flag	Introductions
7-0	GAPV	ADC continuously converts interval configuration values When the interval is configured to 0, the number of successive ADC conversions is up, the ADCIF is set, and the hardware automatically performs the next successive conversion. When the interval configuration value is non-0, the number of consecutive ADC conversions is up, the ADCIF is set, and then the ADC conversion stops, waiting for the software to start the next conversion.

17.2.9 ADC conversion result register SCRHx，SCRLx(x=0...7)

Bit	Flag	Introductions
7-0	SCRHx[7:0]	Detect the high 8 bits of data
7-0	SCRLx[7:0]	Detect the low 8 bits of data

Note: ADC only uses 12/10 bits and is also controlled by data alignment direction control bits.

When converting continuously, eight converted values can be stored. The first converted value is stored in SCRH0, SCRL0, ADCRL, and ADCRH.

When continuous conversion is not enabled, SCRHx, SCRLx(x=1...7) Can be used as RAM, a total of 14 bytes.

17.2.10 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.

18 Low voltage detector/comparator

18.1 LVD/ comparator 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
- Support comparator function

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, thenonly the detection voltage needed to rise to LVD voltage +0.1V, the LVD interrupt request or reset be removed.

LVD detect the voltage on port P4.2, when the voltage is below 1.2V to detection voltage, set the corresponding flag, if the interrupt enable, an interrupt request is generated, if the interruption disable, port voltage detection will generate reset. Port voltage detection will generate valid interrupt and reset, it can wake up the chip from PD and IDLE mode.

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

An LVD interrupt cannot wake the chip from stop mode, but an interrupt can wake the chip from IDLE mode.

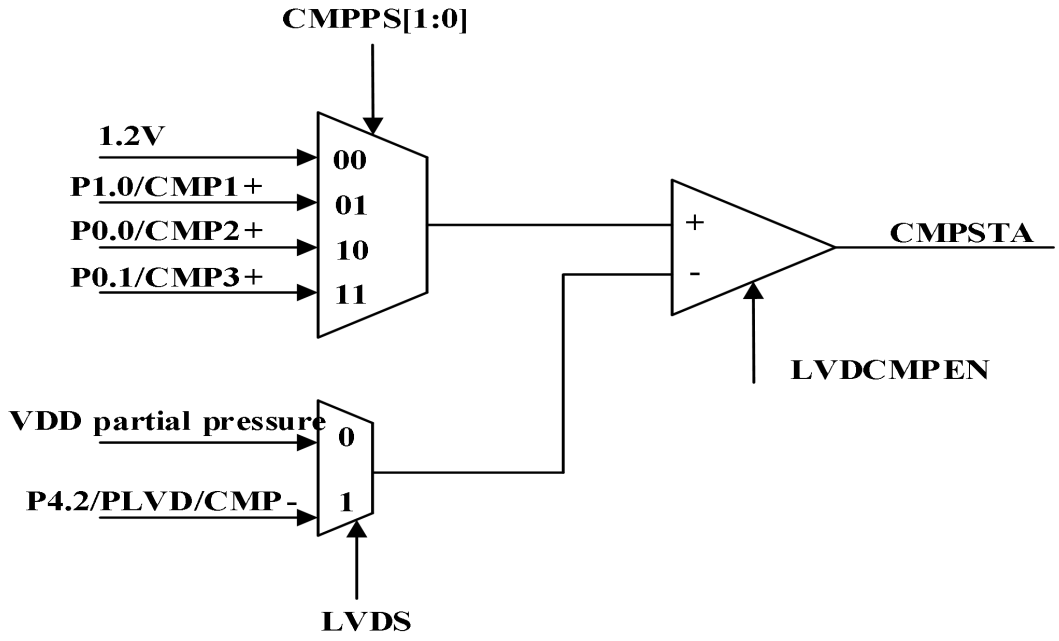


Figure 18- 1 LVD/ Comparator block diagram

18.2 Low voltage detection/comparator related register

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	LVDCMPEN	LVDS	LVDIE	-	LVDF	LVDV		

Bit	Flag	Introductions
7	LVDCMPEN	LVD/Comparator enable bit 0 : Disable LVD/Comparator 1 : Enable LVD/Comparator, It doesn't automatically shut down in stop mode
6	LVDS	LVD detect selection bit 0 : Detect VDD Voltage 1 : Detect P4.2 Voltage (1.2V, ±5%)
5	LVDIE	LVD interrupt enable bit 0 : Disable LVD interrupt 1 : Enable LVD interrupt Note: when disable interrupt, as long as detection enabled, LVDF can also be set 1, but even if EA is set to 1 at this time,no interrupt request is generated. When LVDS is 1, detect the voltage on the P4.2 port: LVDIE=0: Port voltage detection reset LVDIE=1: Port voltage detection interrupt
4	-	Reserved Bits
3	LVDF	Low-voltage detection flag 0: Must software clear 0 1:when the VDD voltage or port voltage is lower than (or higher than) the detection selection voltage, the hardware is set to 1 and the interrupt request is also made Note: The VDD voltage is lower than the detection voltage time is greater than the debounce time set by the LVDDBC register LVDF will be set LVDF, when it is higher than the detection voltage, the bit will not be automatically cleared, it must be cleared by software, only when the VDD voltage is continuously higher than the detection voltage, the software clear can work, if the VDD voltage continues to be lower than the detection voltage, the software cannot clear the 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 Compare function control register LVDCMP

Bit	7	6	5	4	3	2	1	0
R/W	R	R	R/W	R	R/W	R/W	R/W	R/W
Reset values	0	0	1	0	0	0	0	0
Flag	-	-	DBEN	CMPSTA	CMPIM[1:0]		CMPPS[1:0]	

Bit	Flag	Introductions
7-6	-	Reserved Bits
5	DBEN	debouncing enable bit 0：not debouncing 1：debouncing Note： 1. Both LVD and CMP are applicable； 2. In STOP mode and IDLE mode, chattering will not be eliminated automatically. When exiting STOP mode and IDLE mode, DBEN will control whether chattering will be eliminated
4	CMPSTA	Comparator output state 0: the positive voltage of the comparator is less than the negative voltage 1: the positive voltage of the comparator is greater than the negative voltage
3-2	CMPIM[1:0]	When CMPPS[1:0] does not select 00, this register needs to be configured: 00: no LVDF 01: WHEN CMP+ is smaller than CMP- to greater than CMP-, LVDF will be set; 10: LVDF will be set when CMP+ is greater than CMP- and less than CMP-; 11: CMP+ from smaller than CMP- to greater than CMP- or CMP+ from greater than CMP- to less than CMP- are set to LVDF When used as the SOURCE of PWM FLT, this register needs to be set to 01 or 10. 01: CMP+ voltage is greater than CMP- voltage, will produce a PWM FLT high mark; 10: CMP+ voltage is less than CMP- voltage, will produce a PWM FLT high mark;
1-0	CMPPS[1:0]	Positive-end selection bit of comparator 00：1.2V 01: CMP1 pin is the positive input of the comparator 10: CMP2 pin is the positive input of the comparator

		11: CMP3 pin is the positive input of the comparator

18.2.3 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 STOP and IDLE mode automatically turns off, and opens automatically when exit the STOP and IDLE mode.

19 Software LCD

19.1 LCD characteristics

- Supports 1/2 BIAS and 1/3 BIAS LCD dot matrix
- The driver capability is configurable
- The number of COM ports and SEG ports can be set arbitrarily
- LCD control signal (COM and SEG) is realized by software program
- No need to set the mode register for the port when enabling LCD driver operation

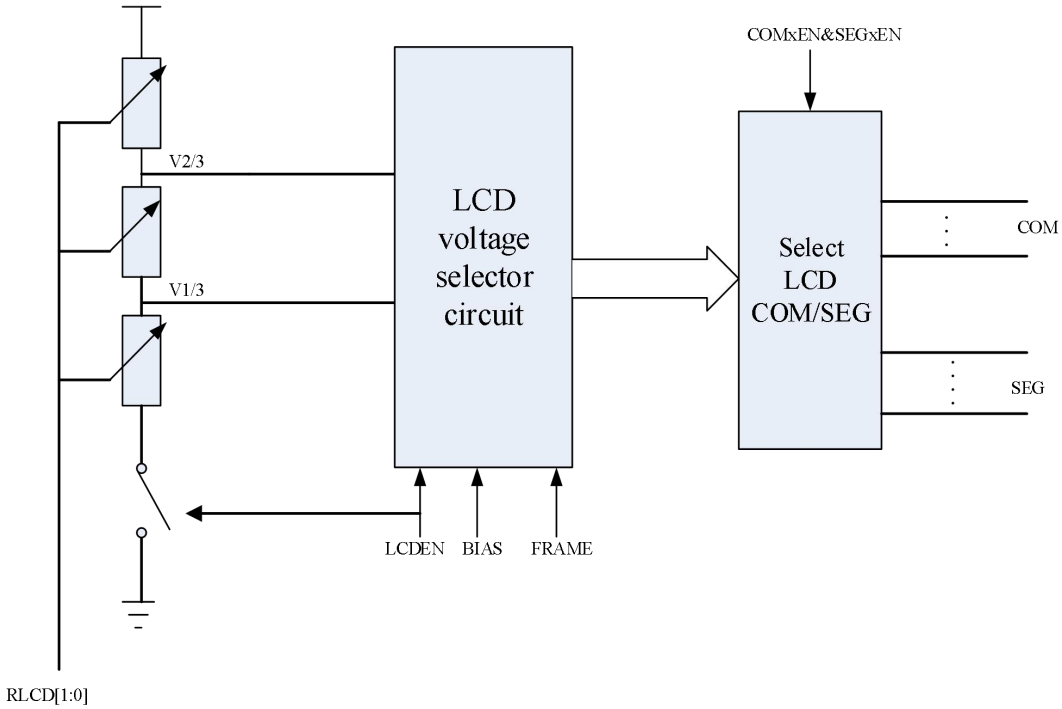


Figure 19- 1 LCD system chart

19.2 Software process description

Take COM[3:0] = P0[3:0] , SEG[7:0] = P1[7:0] for example：

1. Set LCD master enable, LCDEN=1, this is the total enable, open the resistor voltage divider circuit;
2. Set driving capacity and select different resistance partial voltage RLCD[1:0];
3. Set COM port enable control register or SEG port enable control register, COM0EN=0x0F, SEG1EN=0xFF, this is the specific setting of a IO state, enable LCD analog channel;
4. Set Frame0 (FRAME=0) to determine the lit and unlit levels;
5. Set timer start, COM port data register P0=0x01, SEG data register P1=0xXX, wait for timer end;
6. Set timer start, COM port data register P0=0x02, SEG data register P1=0xXX, wait for timer end;

7. Set timer start, COM port data register P0=0x04, SEG data register P1=0xXX, wait for timer end;
8. Set timer start, COM port data register P0=0x08, SEG data register P1=0xXX, wait for timer end;
9. Set Frame1 (FRAME=1) to determine the lit and unlit levels;
10. Set timer start, COM port data register P0=0x01, SEG data register P1=0xXX, wait for timer end;
11. Set timer start, COM port data register P0=0x02, SEG data register P1=0xXX, wait for timer end;
12. Set timer start, COM port data register P0=0x04, SEG data register P1=0xXX, wait for timer end;
13. Set timer start, COM port data register P0=0x08, SEG data register P1=0xXX, wait for timer end;
14. Cycle 4-13.

19.3 LCD Frame

A complete LCD waveform cycle consists of two frames, namely Frame0 and Frame1.

Frame 0

To output the waveform of Frame0, set FRAME to 0 in the LCDCON register.

In Frame0, the COM signal output can be VDD, or VBIAS=1/3VDD (1/2vDD);

In Frame0, the SEG signal output can be GND, or VBIAS=2/3VDD(1/2VDD).

Frame 1

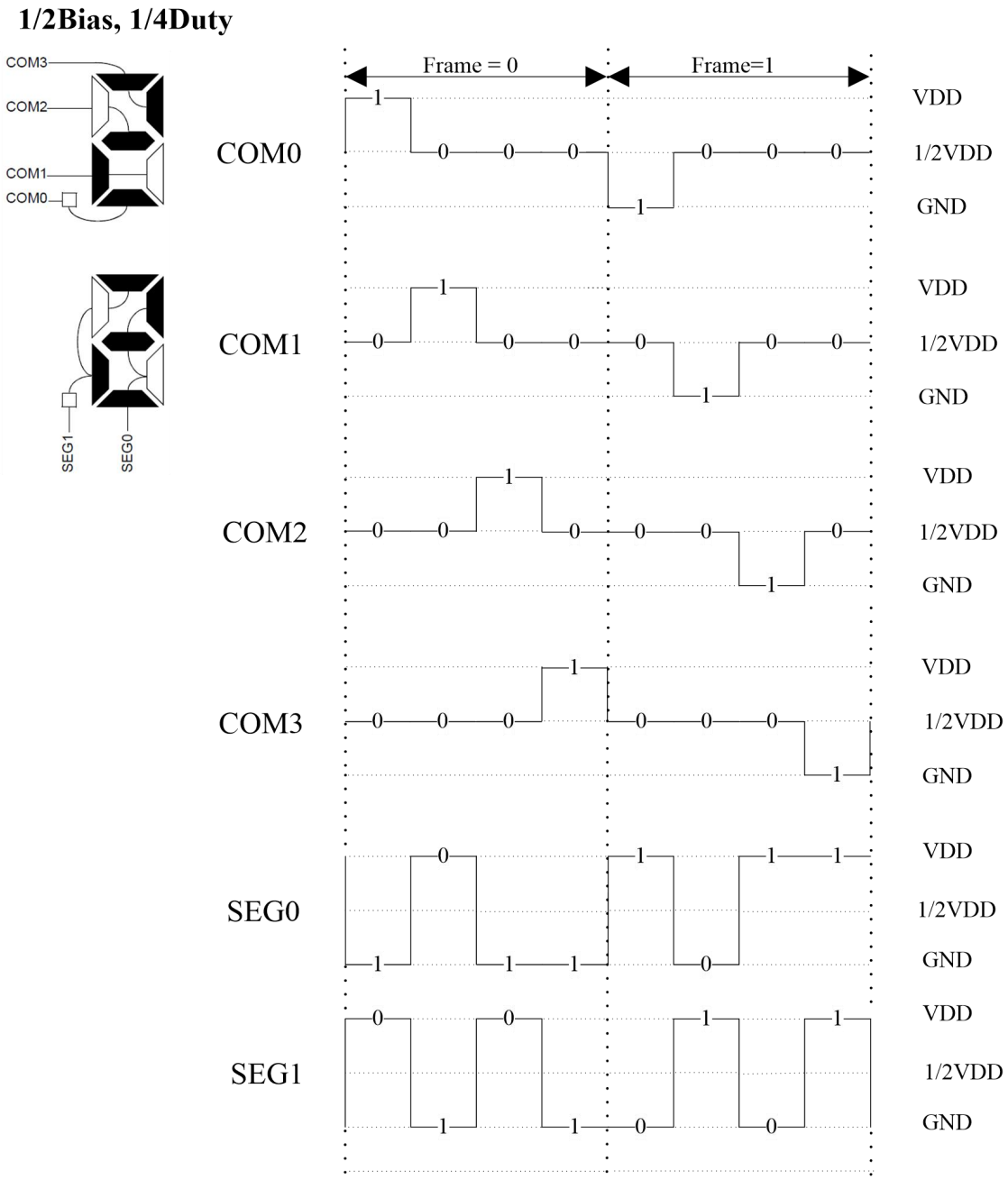
To output the waveform of Frame1, set FRAME to 1 in the LCDCON register.

In Frame1, the COM signal output can be GND, or VBIAS =2/3VDD(1/2VDD);

In Frame1, the SEG signal output can be VDD, or VBIAS=1/3VDD(1/2VDD).

The FRAME bit and the corresponding I/O data register are set by software to determine whether the COM port output is VDD, GND or VBIAS. The software sets the FRAME bit and the corresponding I/O data register to determine whether the current output of the SEG port is VDD, GND or VBIAS (at 1/2BIAS, SEG only outputs VDD or GND).

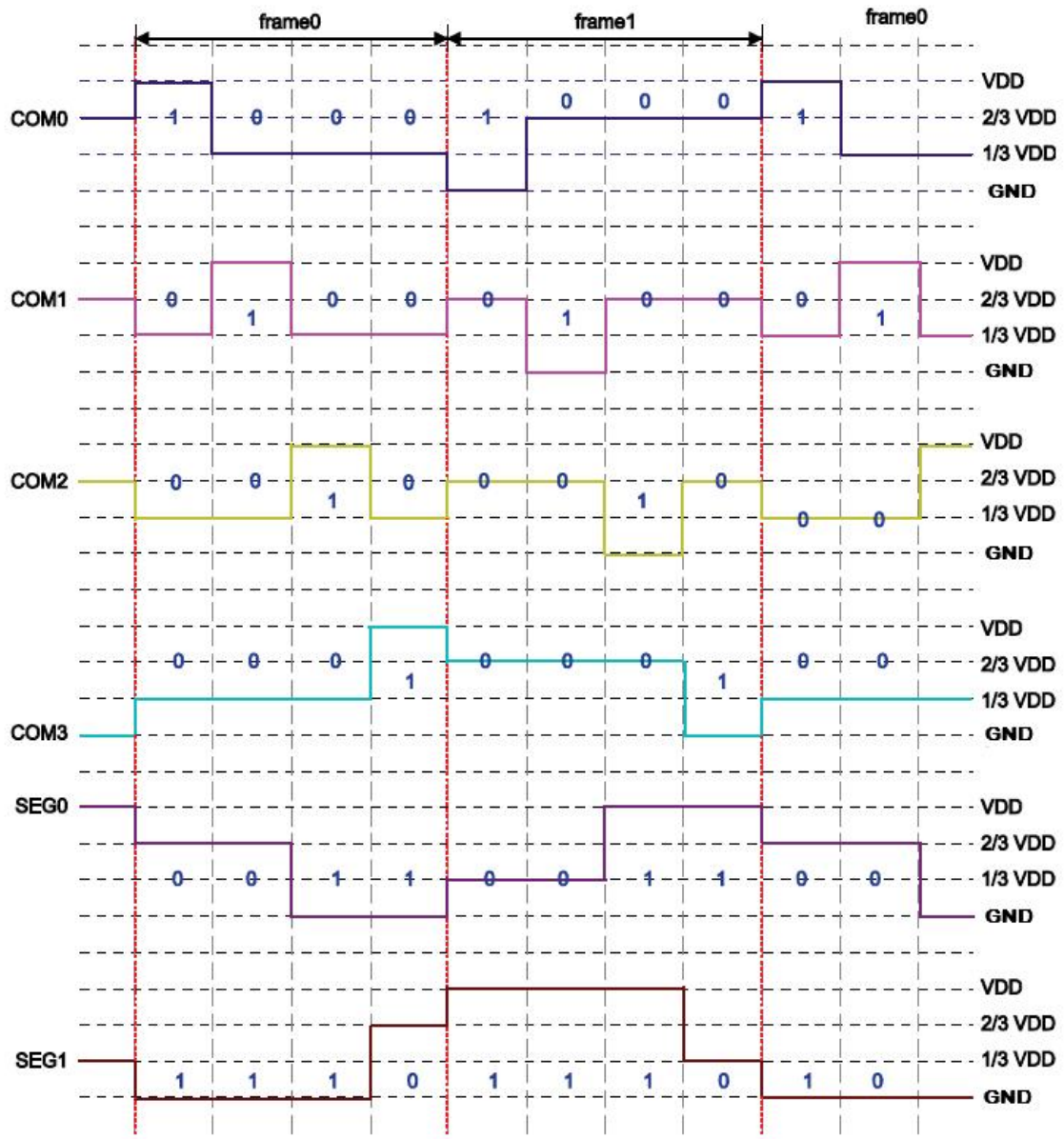
The waveform diagram below shows a typical 1/2Bias LCD waveform generated using an application.The "1" represents the lighting of the LCD.The COM and SEG signal polarity (0 or 1) generated on the COMN and SEGM pins is generated by the corresponding port data register bits.



Note: In the graph, logic value is COM or SEG is corresponding with the bit value of port data register.

Figure 19- 2 1/2bias LCD waveform diagram

The waveform diagram below shows a typical 1/3Bias LCD waveform generated using an application.The "1" represents the lighting of the LCD.The COM and SEG signal polarity (0 or 1) generated on the COMN and SEGM pins is generated by the corresponding port data register bits.



Note: In the graph, logic value is COM or SEG is corresponding with the bit value of port data register.

Figure 19- 3 1/3bias LCD waveform diagram

19.4 LCD related registers

19.4.1 LCD control register LCDCON

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R
Reset values	0	0	0	0	0	0	0	0
Flag	LCDEN	RLCD1	RLCD0	FRAME	BIAS	-		

Bit	Flag	Introductions
7	LCDEN	Software LCD Enable control bits 0: Disable 1: Enable
6-5	RLCD[1:0]	Software LCD resistance selection bit 00: 600kΩ 01: 300kΩ 10: 100kΩ 11: 50kΩ
4	FRAME	Frame0 or Frame1 output enable bit 0: Frame0 1: Frame1
3	BIAS	LCD bias select bit 0:1/2bias 1:1/3bias
2-0	-	Reserve

19.4.2 COM port enable control register COMP0EN-COMP5EN

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	COMPxENy[7:0]							

Bit	Flag	Introductions
7-0	COMPxENy	Software LCD COM function enable bit 0：Disable, standard IO 1：Enable Note：x = 0~5; y = 0-7

Note：COMPxENy[7:0] registers are only writable

For example：COMP1EN=0x11; //COM port enabled P14, P10

19.4.3 SEG port enable control register SEGP0EN-SEGP5EN

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	SEGPxENy[7:0]							

Bit	Flag	Introductions
7-0	SEGPxENy	Software LCD SEG function enable bit 0：Disable, standard IO 1：Enable Note：x = 0~5; y = 0-7

Note：SEGPxENy[7:0] registers are only writable

For example：SEGP1EN=0x11; //SEG port enabled P14, P10

20 Cyclic redundancy check CRC

20.1 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.

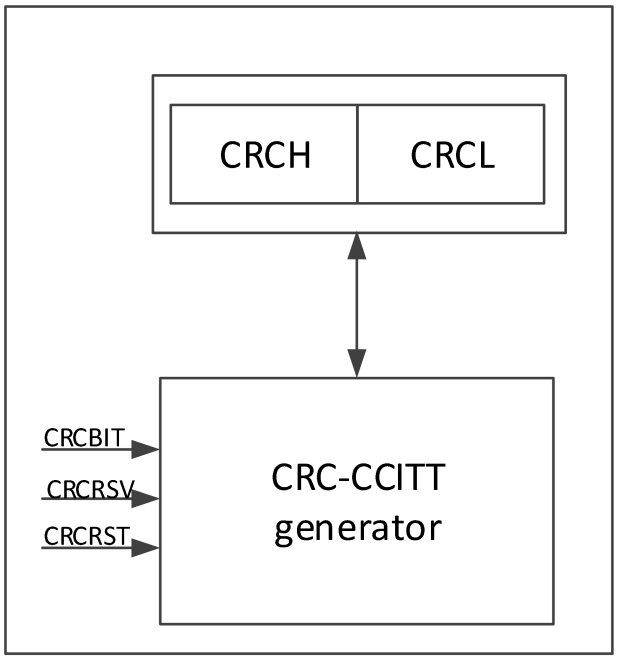


Figure 20- 1 CRC functional block diagram

20.2 CRC registers

20.2.1 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 = 0b, 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

20.2.2 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.

21 Multiplier and divider MCLDIV

21.1 MCLDIV(Features of Multipliers and Dividers)

- 16bit/16-bit hardware divider, 16bit*16-bit hardware multiplier, 32bit/16-bit hardware divider
- 32bit/16-bit up to 20 CPU cycles
- 16bit*16-bit up to 13 CPU cycles
- 16bit/16-bit up to 12 个 CPU cycles

21.2 MCLDIV Related registers

21.2.1 MCLDIV control register MCLDIVC

Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R	R	R	R	R
Reset values	0	0	0	0	0	0	0	0
Flag	OPERS	MD[6:5]		MDEF	MDOV	-	-	-

Bit	Flag	Introductions
7	OPERS	Operation start flag 0: Computation not started or ended 1: Start the operation, the flag bit is 1 during the operation, and the flag bit is cleared by hardware after the operation is completed.
6-5	MD[6:5]	Multiplication and division select register 00: 16*16 Multiplication operation 01: 32/16 division operation 10: 16/16 division operation 11: Reserve
4	MDEF	MDEF will be set to 1 when reading and writing MCLDIV0- MCLDIV5 during the calculation process, but it will not affect the calculation result, and the hardware will automatically clear it to 0
3	MDOV	MDOV will be set to 1 when the product of 16*16 exceeds 16 bits (does not affect the calculation result) or the divisor is 0, and will be automatically cleared to 0 after the calculation is over.
2-0	-	Reserve

21.2.2 MCLDIVO peration/Result Register Reserve MCLDIVAx（x=0~5）

MCLDIVA0

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	MCLDIVA0[7:0]							

Bit	Flag	Introductions
7-0	MCLDIVA0[7:0]	Multiplicand L8 or Dividend LL8 or Product LL8 or Quotient LL8

MCLDIVA1

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	MCLDIVA1[7:0]							

Bit	Flag	Introductions
7-0	MCLDIVA1[7:0]	Multiplicand H8 or Dividend LH8 or Product LH8 or Quotient LH8

MCLDIVA2

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	MCLDIVA2[7:0]							

Bit	Flag	Introductions
7-0	MCLDIVA2[7:0]	Dividend HL8 or Product HL8 or Quotient HL8

MCLDIVA3

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	MCLDIVA3[7:0]							

Bit	Flag	Introductions
7-0	MCLDIVA3[7:0]	Dividend HH8 or Product HH8 or Quotient HH8

MCLDIVA4

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	MCLDIVA4[7:0]							

Bit	Flag	Introductions
7-0	MCLDIVA4[7:0]	Multiplier L8 or Divisor L8 or Remainder L8

MCLDIVA5

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	MCLDIVA5[7:0]							

Bit	Flag	Introductions
7-0	MCLDIVA5[7:0]	Multiplier H8 or Divisor H8 or Remainder H8

Writing and reading of 16bit*16bit multiplication are as follows

Byte / Operand	Byte 3	Byte 2	Byte 1	Byte 0
Multiplicand 16bit	-	-	MCLDIVA1	MCLDIVA0
Multiplier 16bit	-	-	MCLDIVA5	MCLDIVA4
Product 32bit	MCLDIVA3	MCLDIVA2	MCLDIVA1	MCLDIVA0

Writing and reading of 16bit/16-bit division operation are as follows

Byte / Operand	Byte 3	Byte 2	Byte 1	Byte 0
Dividend 16bit	-	-	MCLDIVA1	MCLDIVA0
Divisor 16bit	-	-	MCLDIVA5	MCLDIVA4
Quotient 16bit			MCLDIVA1	MCLDIVA0
Remainder 16bit	-	-	MCLDIVA5	MCLDIVA4

Writing and reading of 32bit/16-bit division operation are as follows

Operand \ Byte	Byte 3	Byte 2	Byte 1	Byte 0
Dividend 32bit	MCLDIVA3	MCLDIVA2	MCLDIVA1	MCLDIVA0
Divisor 16bit	-	-	MCLDIVA5	MCLDIVA4
Quotient 16bit	MCLDIVA3	MCLDIVA2	MCLDIVA1	MCLDIVA0
Remainder 16bit	-	-	MCLDIVA5	MCLDIVA4

22 Code options

1. **External reset enable**
   0：P4.7 as external reset Pin (default). When the port as external reset Pin, it cannot as a normal I/O.
   1：P4.7 as normal IO Pin
2. **External reset level selection**
   0：High level reset（Default）
   1：Low level reset
3. **External crystal selection**
   0：Low-frequency crystal oscillator 32.768KHz（Default）
   1：High-frequency crystal oscillator
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.

23 Electrical characteristics

23.1 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	-	+105	°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.

23.2 DC characteristics

Parameter	Symbol	Condition (VDD=5V)	Min	Typical	Max	Unit
Operating voltage	VDD	F_{CPU} =16MHz or 32KHz	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	-	3.9	-	mA
		F_{CPU} =8MHz,No load, no floating input pins, execute NOP instructions, close the other modules	-	2.4	-	
		F_{CPU} =4MHz,No load, no floating input pins, execute NOP instructions, close the other modules	-	1.6	-	
		F_{CPU} =2MHz,No load, no floating input pins, execute NOP instructions, close the other modules	-	1.2	-	
		F_{CPU} =1MHz,No load, no floating input pins, execute NOP instructions, close the other modules	-	1.0	-	
		F_{CPU} =500KHz,No load, no floating input pins, execute NOP instructions, close the other modules	-	0.9	-	
		F_{CPU} =125KHz,No load, no floating input pins, execute NOP instructions, close the other modules	-	0.8	-	
	I_{OP2}	F_{OSC} =32KHz,No load, no floating input pins, execute NOP instructions, close the other modules	-	72	-	uA
	I_{ID1}	F_{per} =16MHz,no load, no floating input pins, close all modules	-	3.9	-	mA
	I_{ID2}	F_{per} =32KHz,no load, no floating input	-	72	-	uA

		pins, close all modules				
	I_{PD}	Enter stop mode ,no load, no floating input pins, close all modules	-	5.9	-	μA
	I_{IDLE1}	F_{per} =16MHz,enter IDLE mode,no load, no floating input pins, close all modules	-	1.4	-	mA
	I_{IDLE2}	F_{per} =8MHz,enter IDLE mode,no load, no floating input pins, close all modules	-	1.1	-	mA
	I_{IDLE3}	F_{per} =4MH,enter IDLE mode,no load, no floating input pins, close all modules	-	1.0	-	mA
	I_{IDLE4}	F_{per} =2MHz,enter IDLE mode,no load, no floating input pins, close all modules	-	0.9	-	mA
	I_{IDLE5}	F_{per} =1MHz,enter IDLE mode,no load, no floating input pins, close all modules	-	0.8	-	mA
	I_{IDLE6}	F_{per} =125KHz,enter IDLE mode,no load, no floating input pins, close all modules	-	0.8	-	mA
WDT current	I_{WDT}	VDD = 5V	-	1	-	μA
LVD current	I_{LVD}	VDD = 5V	-	0.2	-	
power-down Timer interrupt wakeup current	I_{PW}	F_{CPU} =16MHz,close BOR,TIMER3 count clock source select RC44K,system enter stop mode, TIMER3 timing 1S timing	-	13	-	μA
Input low voltage 1	V_{IL1}	I/O port non-Schmitt input	GND	-	0.55*VDD	V
Input high voltage 1	V_{IH1}	I/O port non-Schmitt input	0.5*VDD	-	VDD	V
Input low voltage 2	V_{IL2}	I/O port Schmitt input	GND	-	0.3*VDD	V
Input high voltage 2	V_{IH2}	I/O port Schmitt input	0.7*VDD	-	VDD	V
Input leakage current 1	I_{ILC1}	I/O port input mode ,V_{IN} = VDD or GND	-1	0	1	μA
output leakage current 2	I_{OLC1}	I/O port input mode ,V_{OUT} = VDD or GND	-1	0	1	μA
Sink	I_{OL1}	Vout=0.1VDD (DREN=00)	-	7	-	mA
	I_{OL2}	Vout=0.1VDD (DREN=01)	-	14	-	
	I_{OL3}	Vout=0.1VDD (DREN=10)	-	28	-	
	I_{OL4}	Vout=0.1VDD (DREN=11)	-	70	-	
Current	I_{OH1}	Vout=0.9VDD (DREN=00)	-	4	-	
	I_{OH2}	Vout=0.9VDD (DREN=01)		7		
	I_{OH3}	Vout=0.9VDD (DREN=10)		10		
	I_{OH4}	Vout=0.9VDD (DREN=11)	-	20	-	

Pull-up resistor	R_{PU1}	Common port,VIN=GND	-	50	-	kΩ
	R_{PU2}	External reset port low level effectively,VIN=GND	-	50	-	
Pull-down resistance	R_{PD1}	Common port,VIN=VDD	-	50	-	
	R_{PD2}	External reset port high level effectively,VIN=GND	-	50	-	
RAM maintain voltage	V_{RAM}	-	-	0.7	-	V

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

23.3 AC characteristics

Parameter	Symbol	Conditions	Min	Typical	Max	Unit
Internal RC 32M startup time	Tset1	room temperature,VDD=5V	-	-	5	μs
Internal RC 44M startup time	Tset2	room temperature,VDD=5V	-	-	150	μs
External high-frequency oscillator startup time	Tset3	16MHz,room temperature,VDD=5V	-	200	-	μs
high-frequency oscillator work volatge	Vset3	16MHz	2.5	-	5.5	V
External low-frequency oscillator startup time	Tset4	room temperature,VDD=5V	-	2	-	s
Frequency accuracy	FIRC1	VDD=2V~5.5V, 25°C	32(1-1%)	32	32(1+1%)	MHz
	FIRC2	VDD=5.0V,-20°C ~+85°C	32(1-2%)	32	32(1+2%)	MHz
	FIRC2	VDD=5.0V,-40°C ~+105°C	32(1-3%)	32	32(1+3%)	MHz
	FWRC	-	22	44	66	KHz

23.4 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	-	10	12	bit
ADC input voltage	VAIN	-	GND	-	Vref	V
ADC input resistance	RAIN	VAIN=5V	2	-	-	MΩ
Analog voltage sources recommended impedance	ZAIN	-	-	-	10	kΩ

ADC switching current	IAD	Open the ADC module ,VDD=5.0V	-	0.6	1	mA
ADC input current	IADIN	VDD=5.0V	-	-	10	μA
Differential nonlinearity error	DLE	VDD=5.0V	-	-	±2	LSB
Integral nonlinearity error	ILE	VDD=5.0V, Vref =2V	-	-	-5~2	LSB
		VDD=5.0V, Vref =3V	-	-	-4~2	
		VDD=5.0V, Vref =4V	-	-	-3~2	
		VDD=5.0V, Vref =VDD	-	-	±2	
		VDD=5.0V, Vref = External parameters	-	-	±2	
Full scale error	EF	VDD=5.0V	-	-	±5	LSB
Offset error	EZ	VDD=5.0V	-	-	±3	LSB
Total error	EAD	VDD=5.0V	-	-	±5	LSB
Total conversion time 1	TCON1	VDD=5.0V Vref =2/3/4V	10	-	-	μs
Total conversion time 2	TCON2	VDD=5.0V Vref =VDD	2	-	-	μs
Internal reference voltage	VADREF	VDD=5.0V, Vref =2V	2(1-1%)	2	2(1+1%)	V

23.5 FLASH memory characteristics

Parameter	Symbol	Conditions	Min	Typical	Max	Unit
Write and Read Test	N_{ENDUR}	-	100000	-	-	Cycle
PaUSe Test	T_{RET}	T=25°C	-	10	-	year
FLASH sector erase time	T_{ERASE}	1 sector（128 bytes）	-	5	-	ms
EEPROM sector erase time	T_{PROG}	1 bytes, Fcpu=16MHz	-	23	-	us
Byte write time	I_{DD1}	Fcpu=16MHz	-	4	-	mA
Consumption Current by Read	I_{DD2}	-	-	4	-	mA
Consumption Current by writing	I_{DD3}	-	-	2	-	mA

Note: Subject to general operating conditions for $V_{DD=5.0V}$ GND=0V, 25 °C unless otherwise specified.

23.6 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

23.7 LVD detection voltage characteristics

Parameter	Symbol	Condition	Min	Typ	Max	Unit
LVD set voltage 1	VLVD1	LVD is enabled ,VDD=2V~5.5V	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

23.8 System power down consumption

1、 System shutdown BOR, Enter stop mode

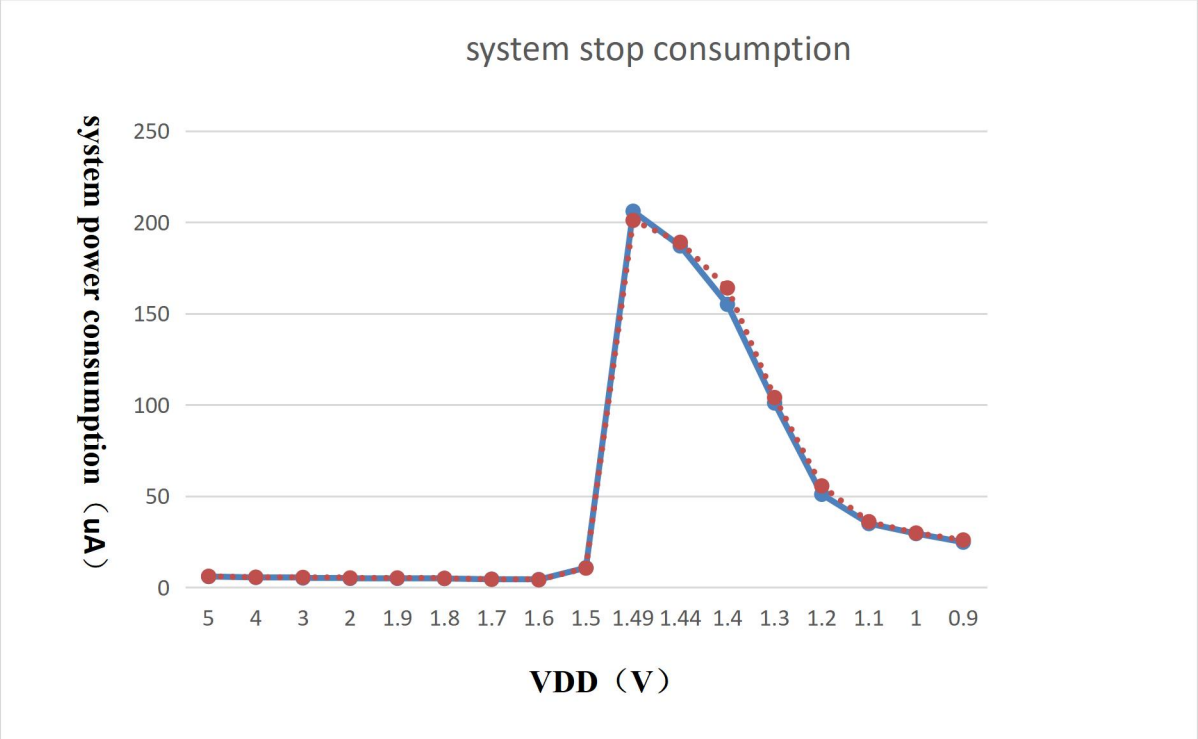


Figure 23- 1 system power down consumption

Figure 23- 2 HF RC/8-V characteristic curve

23.9 Frequency-Temperature characteristic curve

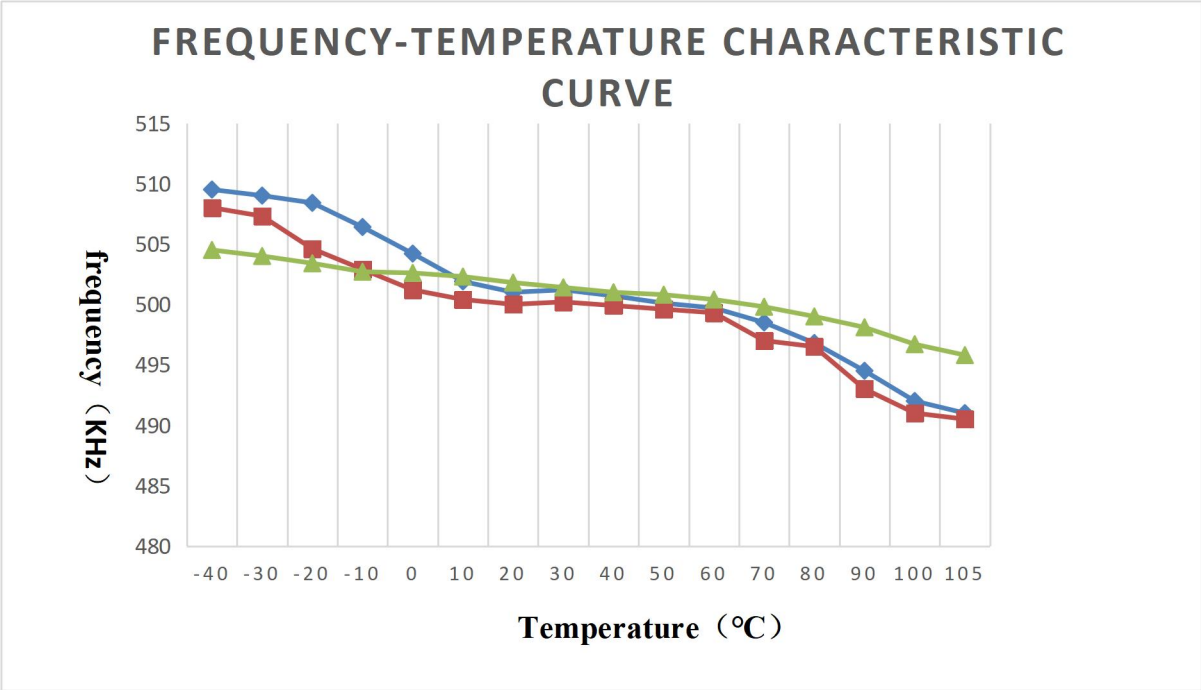


Figure 23- 3 HF IRC32M/500 - Temperature characteristic curve

23.10 ADC VREF 2V- Temperature characteristic curve

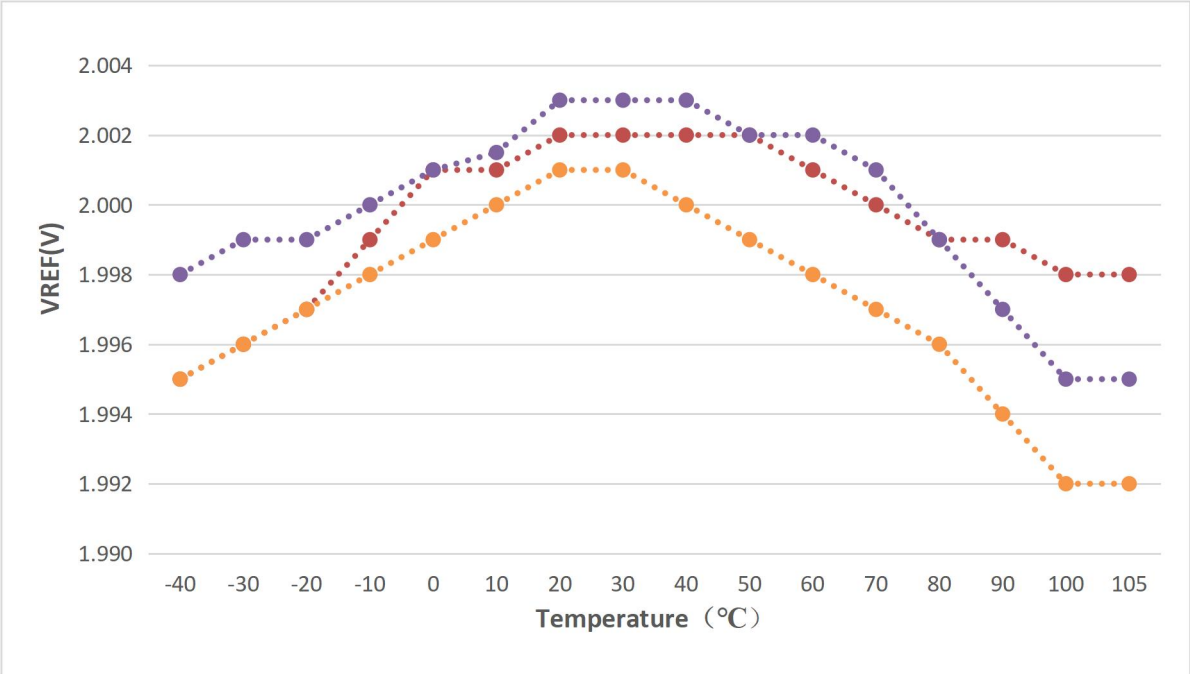


Figure 23- 4 VREF 2V – Temperature characteristic curve

23.11 Other electrical characteristics

1、ESD（HBM）：CLASS 3A（≥8000V）
2、ESD（MM）：CLASS 2（≥200V）
3、Latch_up：CLASSI（250mA）
4、EFT：±4800V

24 Development tools

24.1 HC-LINK

HC89S105A uses HC-Link emulator to download and simulation procedures, HC-Link through JTAG /SWD interface can be holychip all enhanced 8051 core MCU download and simulation. For details about how to use HC-Link, see the HC-Link User Manual.

- upport Keil C51 integration build environment (uVision4.0 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

24.2 HC-PM51

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.

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

24.3 ISP

HC89S105A series chips support the function of ISP serial port burning. Users can use HC-Link or HC-PM51 to burn ISP firmware programs. For details, see related application manuals.

HC-Link V4.0 supports the function of USB to serial port, with the host computer software HC-ISP, through the serial port to solidification of ISP program FLASH microcontroller to achieve one-click download function, without cold boot, very convenient for users to download programs. For details about how to use HC-ISP, see HC-ISP Tool Usage instructions.

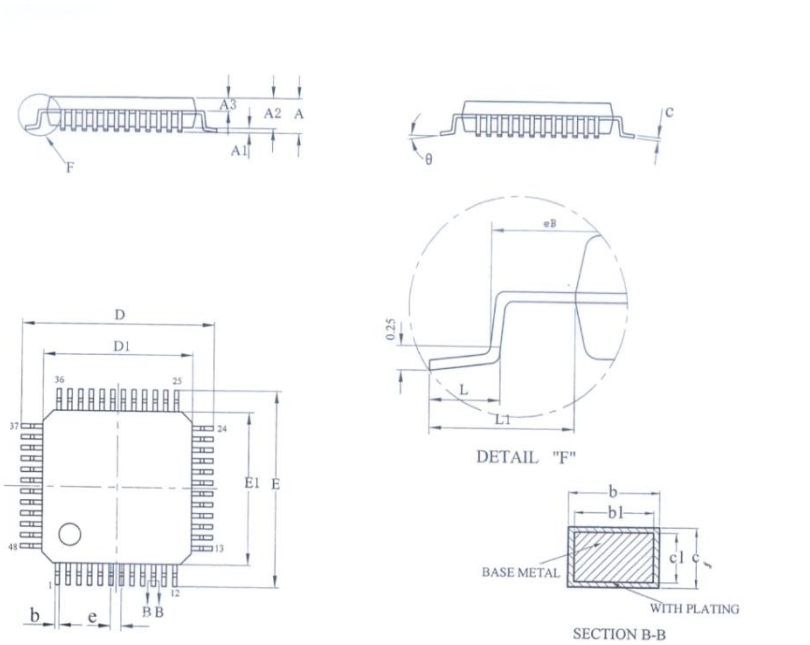
Figure 24- 1 ISP Serial download diagram

24.4 Software download

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

25 Package size

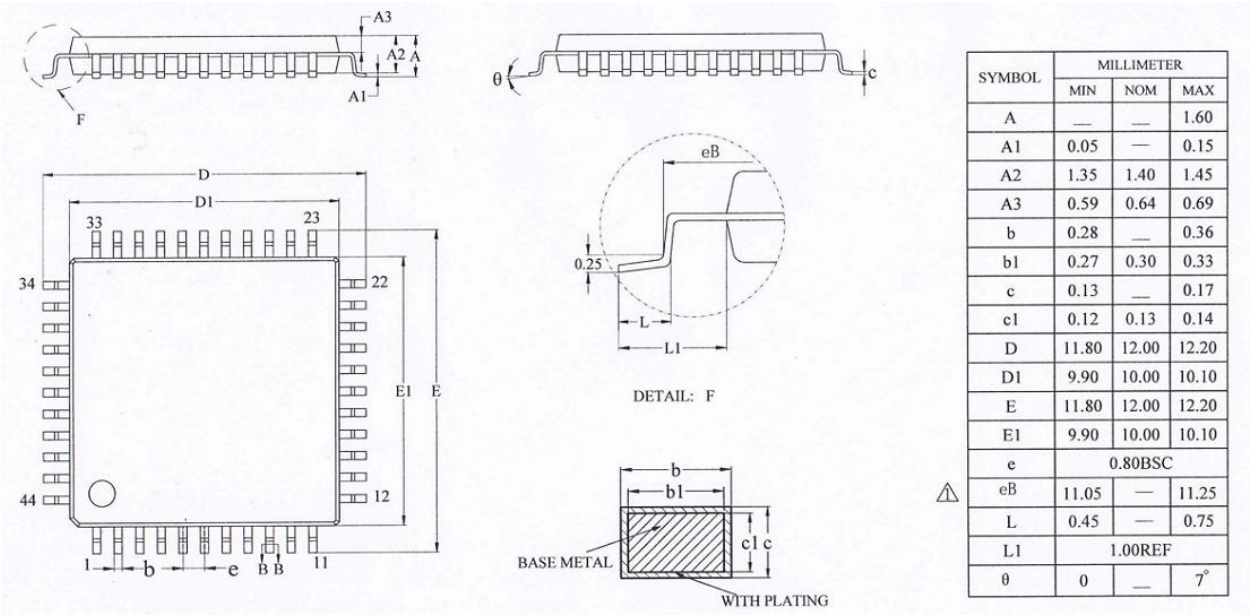
25.1 LQFP48



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 25- 1 LQFP48 package size

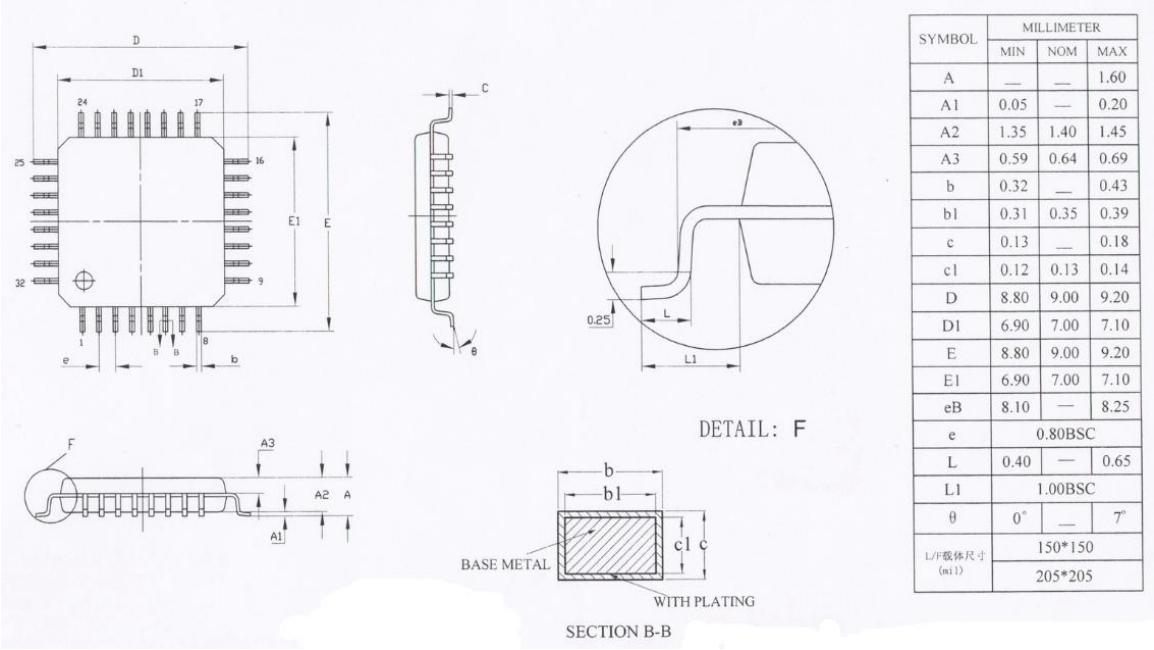
25.2 LQFP44



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 25- 2 LQFP44 package size

25.3 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 25- 3 LQFP32 package size

26 Reversion history

Version	Date	Description
Ver1.00	2021-12-07	First version
Ver1.01	2022-04-12	Revised some errors and improved part of the description
Ver1.02	2022-06-16	Revised some errors
Ver1.03	2022-06-28	Revised some errors
Ver1.04	2023-02-21	1. Precautions for improving the selection position of ADC trigger signals 2. Add COM and SEG port enable control register use examples 3. Modify the ADCST output enable bit ADCST signal output pin error, from P1.4 pin output enable to P0.1 pin output 4. Precautions for modifying the low-voltage detection flag of the register LVDC 5. XRAM changed from 2K to 4K 6. The priority order of modifying the multiplexing function is incorrect 7. Latch_up: CLASSI. (300mA) changed to Latch_up: CLASSI. (250mA)