CACHIP *Built - in 12 Bit ADC / PWM / Touch Key / 1T 8051 16K Flash MCU*

CA51F7 Series MCU
User Guide

REV 1.0

IMPORTANT STATEMET: We reserve the right to make further clarifications regarding the reliability, functionality and design of all products listed below. We also reserve the right to make changes to all documentation for this product without notice. Customers should ask our sales staff for the latest documentation when using this product. Hereby declare!

Table of Contents

1 Introduction

CA51F7 series chip is a special chip for humidifier developed based on 8-bit MCU with 1T 8051 core. Normally, the operation speed is 10 times faster than the traditional 8051 chip, and the performance is more superior. The Flash program memory embedded can be programmed for times and offers users with 16K FLASH, which brings great convenience to software development. CA51F7 not only retains the functions of the traditional 8051 chip, but also integrates 12bit ADC, touch key, 16-bit PMW, UART, I2C, LVD and other function modules. The chip supports connecting wireless modules for communication and intelligent product control. It can operate in three Power Save Modes (IDLE/STOP/LOW SPEED) in order to meet different power consumption needs. With an embedded programmable FPG clock frequency suppression function, CA51F7 could be widely used in nebulizers, humidifiers and medical beauty products to effectively reduce conducted interference.

2 Basic Features

- Core
  - CPU： 1T 8051, with highest speed 10 times faster than traditional 8051
  - Compatible with 8051 instruction set, with double DPTR mode
- Memory
  - Flash: 16K byte， can be erased and overwritten for times
  - Flash could be divided into a storage and a data storage. The data storage functions like an EEPROM and is used to store data that needs to be protected during power down
  - RAM: 256 bytes of internal RAM, 768 bytes of external RAM
- Operating Voltage
  - When the CPU clock is 8MHz, the operating voltage range is 2.2 ~ 5.5V
  - When the CPU clock is 16MHz, the operating voltage range is 2.7 ~ 5.5V
- Clock System
  - Internal Low Speed RC Oscillator： 100KHz（±50KHz）
  - Internal High Speed RC Oscillator： 16MHz， with 2% precision
  - Programmable RC oscillator PFG: 20-40MHz with internal automatic frequency hopping
  - PFG clock frequency correction module: enables the PFG clock to be counted against the system clock to correct the central frequency

- TMC Function
  - The clock source is the low speed internal RC oscillator and the minimum time unit for interrupt is 512 clock cycle periods of it
  - Configurable interrupt time ranges from 1 to 256 minimum time units

- Interrupt System
  - 7 effective interrupt sources
  - Two levels for interrupt priority, which also supports interrupt nesting
  - 5 external interrupt sources

- Timer
  - Two 16-bit general timers: Timer 0, Timer 1

- General Purpose IO（GPIO）
  - Supports 18 GPIO at most and also supports push-pull, open-drain, pull-up, pull-down and high-impedance mode
  - One PWM pin (P07) can be set for strong push current and strong sink current to drive MOS tubes directly（Specially apply to humidifiers）
  - Five PWM pins can be set for strong push current and can directly drive the cooling fan and LED indicators（Specially apply to humidifiers）

- Analogue/Digital Converters（ADC）
  - Supports 10-channel 12-bit SAR ADC
  - Supports 2 types of reference voltage sources: VDD and internal reference (1.5V)
  - Selecting the internal voltage as reference allows direct measurement of the VDD voltage

- Touch Key
  - Internal touch sensor controller and support up to 10 touch channels
  - Internal charging and internal reference and touch check water anti-dry burn function can be set
  - Integral touch frequency hopping function for significantly improved resistance to voltage pulse injection (CS), high immunity to interference and EMC (CS) compliance
  - Support for touch power saving mode

- PWM
  - Supports 6 channel PWM, any periods or duty cycles are configurable in 16 bits
  - Supports to output internal clock directly

- Low voltage detection（LVD）
  - The detection voltage can be set to 2.7V, 3.3V, 3.7V and 4.2V
  - Low voltage reset or interrupt can be set

- Reset mode
  - Supports variable reset sources：Hard Reset, Soft Reset, Watchdog Reset, LVD Reset, Power On/Down Reset

- Watchdog
  - 27 bits Watchdog Timer, 16 bits precision configurable, with Watchdog Reset and Interrupt configurable as well

- Universal Asynchronous Receiver/Transmitter（UART）
  - Supports 1 UART port
  - Supports 1 byte receive buffer

- I2C
  - One I^2C port embedded which supports Master-Slave mode and Standard mode

- Program Download and Simulation
  - Supports ISP and IAP
  - Support simulation online

- Low power consumption
  - For STOP mode, current <5uA
  - For IDLE mode, current<30Ua

- Package Type：TSSOP20/SOP16

3 Chip Models and Function Description

Table 3-1 CA51F7 series specific model functional features

Models	Flash Storage [BYTE]	External Ram[BYTE]	Internal High Speed Crystal Oscillator	Internal Low Speed Crystal Oscillator	Internal programmable PFG oscillator	GPIO	UART	I^{2}C	16 bit PWM Channels	Touch Key	12-bit ADC	General purpose 16-bit timer	ISP	Simulation On Chip	Working Voltage	Package Type
CA51F752T3	16K	768	√	√	√	18	√	√	6	10	10	2	√	√	2.2-5.5	TSSOP20
CA51F752S3	16K	768	√	√	√	14	√	√	6	6	7	2	√	√	2.2-5.5	SOP16

4 System Block Diagram

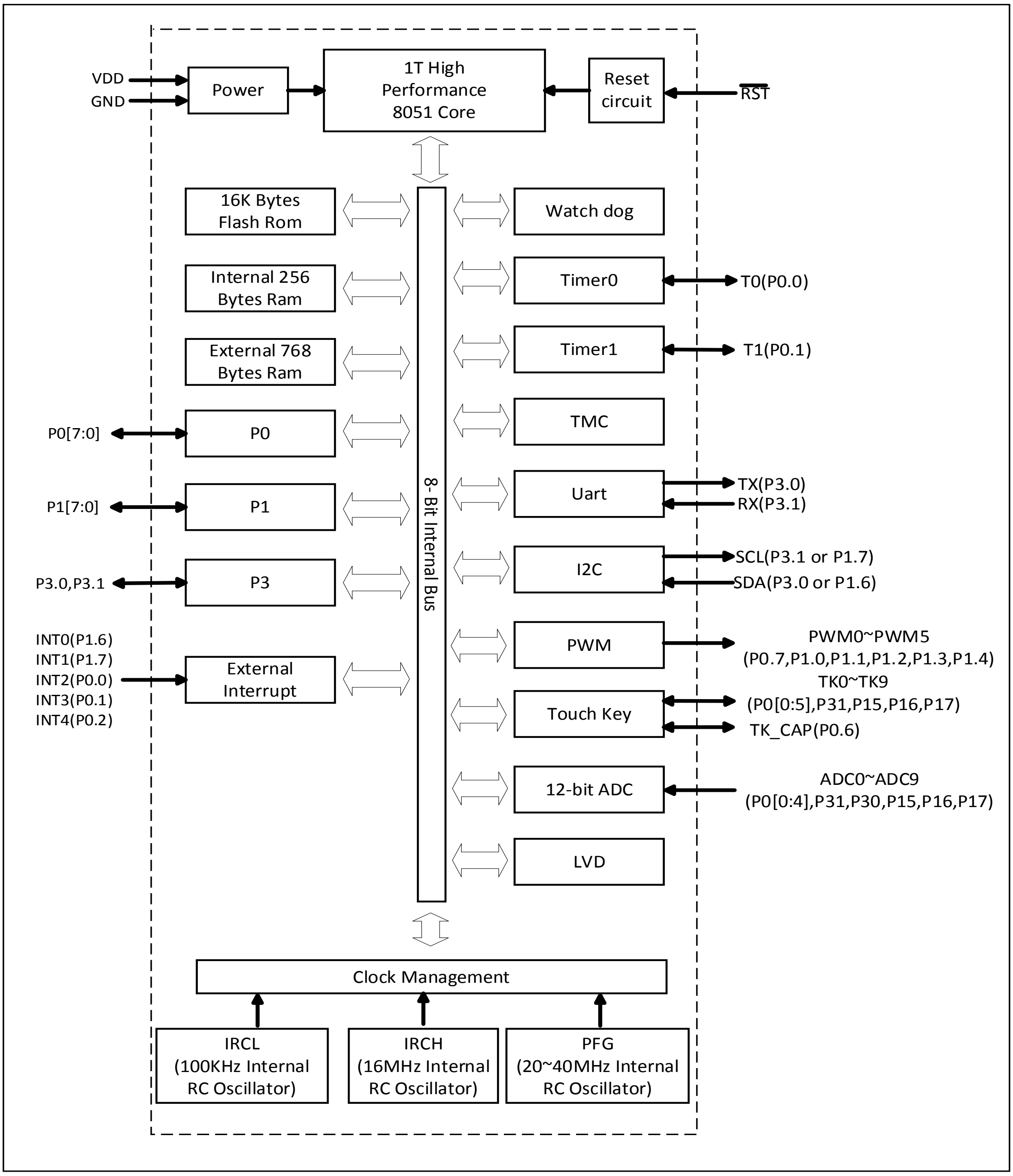


Figure 4-1-1 Chip Block Diagram

5 Pin Package and Description

5.1 Package Definition

Model：CA51F752T3

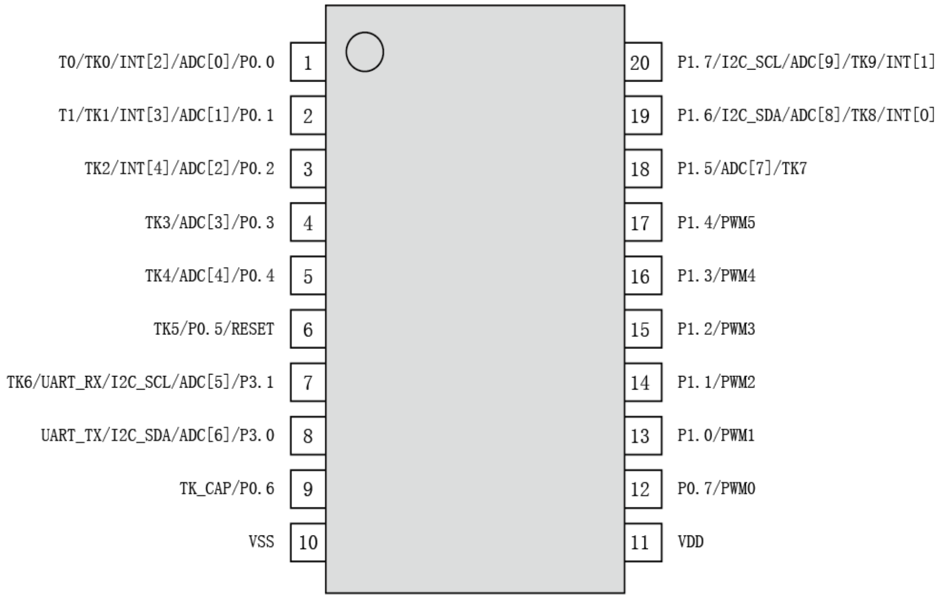


Figure 5-1-1 TSSOP20 Package

Model：CA51F752S3

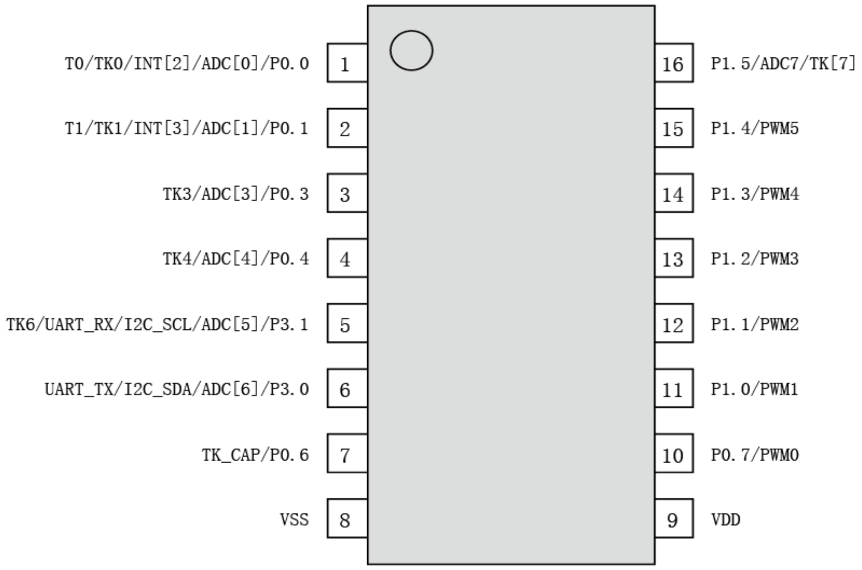


Figure 5-1-2 SOP16Package

5.2 Pin Description

Table 5-2-1 Pin Description

Pin Sequence Number		Pin Name	Pin Function	Default function
TSSOP20	SOP16			
1	1	P0.0/ADC[0] /TK[0]/INT2/T0	General bi-directional I/O port ADC analog channel input Touch key analog channel input	General bi-directional I/O port
2	2	P0.1/ADC[1] /TK[1]/INT3/T1	General bi-directional I/O port ADC analog channel input Touch key analog channel input	General bi-directional I/O port
3		P0.2/ADC[2] /TK[2]/INT4	General bi-directional I/O port ADC analog channel input Touch key analog channel input	General bi-directional I/O port
4	3	P0.3/ADC[3] /TK[3]	General bi-directional I/O port ADC analog channel input Touch key analog channel input	General bi-directional I/O port
5	4	P0.4/ADC[4]/TK[4]	General bi-directional I/O port ADC analog channel input Touch key analog channel input	General bi-directional I/O port
6		P0.5/RESET]/TK[5]	General bi-directional I/O port External reset pin Touch key analog channel input	Hardware reset pin
7	5	P3.1/I2C_SCL/ ADC[5]/TK[6] /UART_RX	General bi-directional I/O port UART0_RX transmission port ADC analog channel input External touch capacitor input port I^{2}C clock transfer port	I2C clock transfer port
8	6	P3.0/I2C_SDA/ADC[6]/UART_TX	General bi-directional I/O port UART0_RX transmission port ADC analog channel input I^{2}C data transfer port	I^{2}C data transfer port
9	7	P0.6/ TKCAP	General bi-directional I/O port External touch capacitor input port	General bi-directional I/O port
10	8	VSS	Power ground pin	Ground

11	9	VDD	Chip power supply pin	power supply for the chip
12	10	P0.7/PWM0	General bi-directional I/O port PWM signal output	General bi-directional I/O port
13	11	P1.0/PWM1	General bi-directional I/O port PWM signal output	General bi-directional I/O port
14	12	P1.1/PWM2	General bi-directional I/O port PWM signal output	General bi-directional I/O port
15	13	P1.2/PWM3	General bi-directional I/O port PWM signal output	General bi-directional I/O port
16	14	P1.3/PWM4	General bi-directional I/O port PWM signal output	General bi-directional I/O port
17	15	P1.4/PWM5	General bi-directional I/O port PWM signal output	General bi-directional I/O port
18	16	P1.5/ADC[7]/TK[7]	General bi-directional I/O port ADC analog channel input Touch key analog channel input	General bi-directional I/O port
19		P1.6/I2C_SDA/ADC[8]/TK[8]	General bi-directional I/O port ADC analog channel input Touch key analog channel input I^2C data transfer port	General bi-directional I/O port
20		P1.7/I2C_SCL/ADC[9]/TK[9]	General bi-directional I/O port ADC analog channel input Touch key analog channel input I^2C clock transfer port	General bi-directional I/O port

6 Central Processing Unit(CPU)

6.1 CPU Introduction

The core of CA51F7 Series is monocyclic 8051 CPU and make it fully compatible with original MCS-51 instruction set. A monocyclic 8051 CPU usually operates 10 times faster than standard 8051 one due to its pipeline structure

The features of this CPU are：

- 1T 8051 CPU
- Compatible with 8051 instructions set, for more you may refer to instruction in Appendix
- Double DPTR, so that the data could be moved quickly

6.2 Register Description

Program Counter (PC)

Program counter is a 16-bit register without register address which is specially used to control the sequence of instructions. It is set to 0 after reset or power on and the MCU will execute the program from the zero address.

Accumulator (ACC)

Accumulator ACC is a commonly used special register and 'A' is used as its instruction. It is often used to store the operands and results of logical/arithmetic computing.

General Register B

Register B needs to be used in conjunction with ACC in multiplication and division operations. The instruction MUL AB multiplies ACC and the 8-bit unsigned number in B. The low byte (16 bits) and high byte (16 bits) of the calculation result will be stored in A and B, respectively. In addition, the instruction DIV AB divides B by A. The quotient of the integer will be stored in A and the remainder in B. Register B can also be used as a general-purpose temporary storage register.

Stack Pointer (SP)

Stack Pointer (SP) is a 8 bit special register and indicates where the top of stack is in the internal RAM. It is initialized to 07H after a reset which makes stack actually starts from 08H. Since 08H~1FH belongs to working register group 1~3, if they are used in program development, SP is recommended to be set to 80H or even higher.

Data Pointer (DPTR)

Data pointer DPTR0/DPTR1 are two 16-bit special register with their higher stored in register DP0H/DP1H respectively and lower bytes stored in register DP0H/DP1H respectively. By setting DPS(PSW.1) either of them can be used. For each DPTR, it can be seen as one 16-bit register or two independent 8-bit registers DP0H/DP1H and DP0L/DP1L.

Program Status Word (PSW)

Program Status Word(PSW) is a register indicates the statues of the CPU. The status bit of it will change correspondingly when the CPU is doing arithmetic or logical operations.

Table 6-2-1 Accumulator ACC

E0H	7	6	5	4	3	2	1	0
ACC	ACC[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0

Table 6-2-2 General Register B

F0H	7	6	5	4	3	2	1	0
B	B[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0

Table 6-2-3 Stack Pointer SP

81H	7	6	5	4	3	2	1	0
SP	SP[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	1	1	1

Table 6-2-4 Data Pointer DP0L

82H	7	6	5	4	3	2	1	0
DP0L	DP0L[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0

Table 6-2-5 Data Pointer DP0H

83H	7	6	5	4	3	2	1	0
DP0H	DP0H[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0

Table 6-2-6 Data Pointer DP1L

84H	7	6	5	4	3	2	1	0
DP1L	DP1L[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0

Table 6-2-7 Data Pointer DP1H

85H	7	6	5	4	3	2	1	0
DP1H	DP1H[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0

Table 6-2-8 Program Status Word PSW

D0H	7	6	5	4	3	2	1	0
PSW	CY	AC	F0	RS[1:0]		OV	DPS	P
R/W	R/W	R/W	R/W	R/W		R/W	R	R
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit symbol	Description
7	CY	Carry flag bit 0： There is no carry or borrow happened in arithmetic/logical operation 1： There is carry or borrow happened in arithmetic/logical operation
6	AC	Auxiliary carry flag bit 0： There is no auxiliary carry or borrow happened in arithmetic/logical operation 1： There is auxiliary carry or borrow happened in arithmetic/logical operation
5	F0	F0 flag bit It is defined by the user
4~3	RS	R0~R7 registers' page selection bits 00： page 0（mapping to 00H-07H) 01： page 1（ mapping to 08H-0FH) 10： page 2（ mapping to 10H-17H) 11： page 3（ mapping to 18H-1FH)
2	OV	Overflow flag bit 0： no overflow 1： overflow happened
1	DPS	DPTR selector，0 for DPTR0, 1 for DPTR1
0	P	Parity flag bit 0： the number of 1 in ACC is even 1： the number of 1 in ACC in odd

Table 6-2-9 Register SPMAX

F3H	7	6	5	4	3	2	1	0
SPMAX	SPMAX[7:0]							
R/W	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit symbol	Description
7~0	SPMAX	SPMAX is used to record the maximum value of SP. Users can check this register using software to decide whether there is a risk that the stack may overflow

7 Memory Systems

7.1 Random Access Memory(RAM)

CA51F7 series chips offer both internal RAM(256 bytes) and external RAM(768 bytes) for the users and the corresponding address are shown as follows:

- Lower 128 bytes of internal RAM（address：00H ~ 7FH）supports both direct addressing or indirect addressing
- Higher 128 bytes of the internal RAM(address: 80H ~ FFH）only supports indirect addressing
- 768 bytes external RAM（address：0000H ~ 02FFH）supports indirect addressing by using MOVX instruction

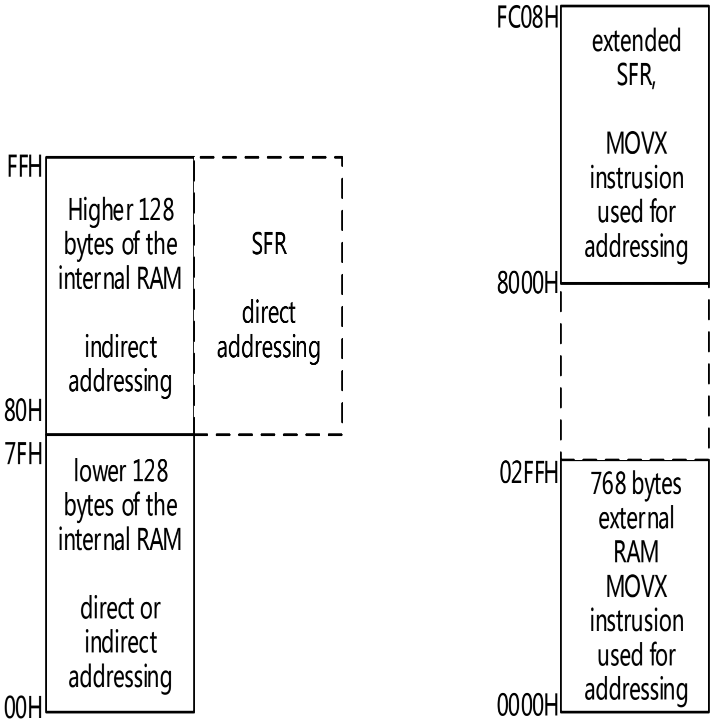

FFH
Higher 128 bytes of the internal RAM
indirect addressing
SFR
direct addressing
80H
7FH
lower 128 bytes of the internal RAM
direct or indirect addressing
00H
FC08H
extended SFR,
MOVX instrusion used for addressing
8000H
02FFH
768 bytes external RAM MOVX instrusion used for addressing
0000H

Figure 7-1-1 RAM Architecture

7.2 Special Function Register(SFR)

CA51F7 series chips provide SFR compatible with traditional 8051 chip. SFR and higher 128 bytes of the internal RAM both use the address 80H ~ FFH that only supports direct addressing, SFR mapping is shown in Table 7-2-1.

Table 7-2-1 Special Function Register Mapping Table

	Bit addressable	Not bit addressable						
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8H	TKCON	TKCFG	TKMTS	TKIF	TK0CHS	TK1CHS	TK2CHS	TK3CHS
F0H	B	TK4CHS	TK5CHS	SPMAX	RCMSLL	RCMSLH	RCMSHL	RCMSHH
E8H	LVDCON	RCCON	-	VCKDL	VCKDH	STEP	STEPNUM	-
E0H	ACC	TK0MSL	TK0MSH	TK1MSL	TK1MSH	TK2MSL	TK2MSH	TK3MSL
D8H	UDCKS	TK3MSH	TK4MSL	TK4MSH	TK5MSL	TK5MSH	TKCKS	TKPWC
D0H	PSW	EP0CON	EP1CON	EP2CON	EPIF	TMCON	TMSNU	I2CIOS
C8H	CKCON	CKDIV	IHCFG	PFGCFGL	PFGCFGH	PFGTRIML	PFGTRIMH	-
C0H	I2CCON	I2CADR	I2CADM	I2CCCR	I2CDAT	I2CSTA	I2CFLG	PWMEN
B8H	IP	PWM0CON	PWM1CON	PWM2CON	PWM3CON	PWM4CON	PWM5CON	PWMNUMH
B0H	P3	PWM0CKD	PWM1CKD	PWM2CKD	PWM3CKD	PWM4CKD	PWM5CKD	PWMNUML
A8H	IE	PWM0DIVL	PWM0DIVH	PWM1DIVL	PWM1DIVH	PWM2DIVL	PWM2DIVH	PWM3DIVL
A0H	WDCON	WDFLG	WDVTHL	WDVTHH	PWM3DIVH	PWM4DIVL	PWM4DIVH	PWM5DIVL
98H	SCON	SBUF	PWM5DIVH	PWM0DUTL	PWM0DUTH	PWM1DUTL	PWM1DUTH	PWM2DUTL
90H	P1	PWM2DUTH	PWM3DUTL	PWM3DUTH	PWM4DUTL	PWM4DUTH	PWM5DUTL	PWM5DUTH
88H	TCON	TMOD	TL0	TL1	TH0	TH1	IDLST	STPST
80H	P0	SP	DP0L	DP0H	DP1L	DP1H	PWCON	PCON

Due to the limited SFR address space, the CA51F7 series chips add extended special function registers to the external RAM address space. The extended special function register mapping is shown in Table 7-2-2.

Table 7-2-2 Extended Special Function Register Mapping Table

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
8000H	P00F	P01F	P02F	P03F	P04F	P05F	P06F	P07F
8008H	P10F	P11F	P12F	P13F	P14F	P15F	P16F	P17F
8018H	P30F	P31F	-	-	-	-	-	-
8060H	ADCON	ADCFGL	ADCDL	ADCDH	ADCALL	ADCALH	RCTAGL	RCTAGH
8068H	SRELL	SRELH						
8120H	P00C	P01C	P02C	P03C	P04C	P05C	P06C	P07C
8128H	P10C	P11C	P12C	P13C	P14C	P15C	P16C	P17C
8138H	P30C	P31C	-	-	-	-	-	-
FC00H	MECON	FSCMD	FSDAT	LOCK	PADRD	PTSL	PTSH	-

7.3 Flash

7.3.1 Function Introduction

Flash memory contains 16K bytes of Flash main data area, and it can be erased and overwritten repeatedly. Flash is also controlled by a group of specific registers; therefore, users may use these registers to erase/overwrite/set write protect to the Flash
and so on.

7.3.2 Flash Memory Architecture

- Flash consists of several sectors, which are the smallest units for erasure and write operation, and each sector is 64 bytes.
- Flash write operations are performed in sectors units and must write 64 bytes at once. Single-byte writes are not supported
- Flash can be divided into PROGRAM and DATA area by function, and the division unit is 128 bytes. The PROGRAM area is used to store the user's program and the data is used to store data that needs to be saved during power down period.

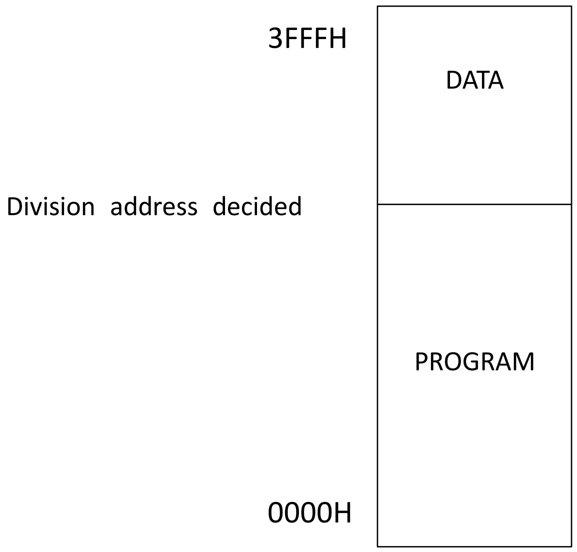


Figure 7-3-2-1 16K Flash Memory Structure

7.3.3 Flash Memory Description

Table 7-3-3-1 Register MECON

FC00H	7	6	5	4	3	2	1	0
MECON	-	DPSTB	-	-	-	-	-	BOOT
R/W	-	R/W	-	-	-	-	-	R/W
Initial Value	-	0	-	-	-	-	-	0

Bit number	Bit symbol	Description
7	-	-
6	DPSTB	Flash SLEEP mode control bit in IDLE/STOP mode 0： Flash in NORMAL mode while IDLE/STOP 1： Flash in SLEEP mode while IDLE/STOP *Note： if DPSTB=1， when the chip enters IDLE/STOP mode， the Flash will enter SLEEP mode simultaneously . When the chip exits IDLE/STOP mod, the Flash exits SLEEP mode as well.*
5~1	-	-
0	BOOT	Programs start area control bit after soft reset 0： Program starts from FLASH after soft reset 1： Program starts from XRAM after soft reset

Table 7-3-3-2 Register FSCMD

FC01H	7	6	5	4	3	2	1	0
FSCMD	IFEN	-	-	-	CLRPL	CMD[2:0]		
R/W	R/W	-	-	-	0	R/W		
Initial Value	0	-	-	-	0	0	0	0

Bit number	Bit symbol	Description
7	IFEN	Information area access enable bit, it needs to be set when accessing
6~4	-	-
3	CLRPL	Clear data from Flash latch
2~0	CMD	Command register 000： No operations 100： Erase the Flash sector （ Each sector size is 4K bytes ） 001： Read Flash DATA area 010： Write Flash DATA area 011： Erase one page of Flash DATA area 101： Read Flash PROGRAM area 110： Write Flash PROGRAM area 111： Erase one page of Flash PROGRAM area *note：* *1. CMD is automatically cleared after the erase and write commands are executed* *2. The CMD remains unchanged after the read command is written and then will be done by reading/writing FSDAT*

Table 7-3-3-3 Register FSDAT

FC02H	7	6	5	4	3	2	1	0
FSDAT	FSDAT[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit symbol	Description
7~0	FSDAT	Flash data register

Table 7-3-3-4 Register LOCK

FC03H	7	6	5	4	3	2	1	0
LOCK								
R	-	REPE	-	-	FLKF	PLKF	DLKF	ILKF
W	LOCK[7:0]							
初始值	-	0	-	-	0	0	0	0
Bit number	Bit symbol	Description						
Writer								
7~0	LOCK	28H：Unlock Flash programmable area 29H：Unlock Flash PROGRAM area 2AH：Unlock Flash DATA area AAH：Lock Flash, R/W forbidden						
Read								
7~4	-							
3	FLKF	Programmable area unlocked flag, 1 indicates unlocked						
2	PLKF	PROGRAM area unlocked flag, 1 indicates unlocked						
1	DLKF	DATA area unlocked flag, 1 indicates unlocked						
0	-	-						

Table 7-3-3-5 Register PADRD

FC04H	7	6	5	4	3	2	1	0
PADRD	PADRD[7:0]							
R/W	R/W							
Initial Value	1	0	0	0	0	0	0	0
Bit number	Bit symbol	Description						
7	-	-						
6~0	PADRD	PROGRAM and DATA area division configuration register The program area and data area are divided in units of 128 bytes and when PADRD>0: The address space of the PROGRAM area: 0 ~ (PADRD×128 - 1), The address space of the data area: (PADRD×128) ~ 3FFFH. *Note：* *1. PADRD=0 indicates the whole Flash is DATA area* 2. *The maximum value for PADRD is 80H. PADRD can not be set to any values greater than the maximum.*						

Table 7-3-3-6 Register PTS

FC05H	7	6	5	4	3	2	1	0
PTSL	PTS[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
FC06H	7	6	5	4	3	2	1	0
PTSH	-	-	PTS[13:8]					
R/W	-	-	R/W					
Initial Value	-	-	0	0	0	0	0	0
Bit number	Bit symbol	Description						
15~14	-	-						
13~0	PTS	Destination address pointer register, when writing FSDAT operation, the data will be written to the flash latch associated with PTS[5:0] for temporary storage, PTS[5:0] corresponds to the lower 6 bits of the actual write operation; when sending the write command, the associated page address PTS[13:6] needs to be set. It is best to reconfigure the PTS address once for each read/ write/erase operation, and for continuous read operations, you can just set the first address operation for continuous read operations.						

7.3.4 Flash Control Example

◆ Divide Flash into DATA area and PROGRAM area

For instance, if the user wants to divide a 16K Flash (128 byte DATA area and the remains for PROGRAM area), the program may like this:

```
---------------------------------------------------------------------------------------
PADRD = 127; // The address for PROGRAM area will be 0~0x3F7F, the address for DATA area will be
0x3F80~0x3FFF
---------------------------------------------------------------------------------------
```

Note: The physical address of the data area in FLASH set above is 0x3F80~0x3FFF, but the logical address is 0x0000~0x007F, the logical address should be filled in when reading and writing the data area.

◆ Sector erasure of DATA area

Sector n of DATA area needs to be erased, for example, the program may as follows:

```
---------------------------------------------------------------------------------------
FSCMD = 0; //set CMD=0
LOCK = 0x2A; //unlock DATA area
FSCMD = 8;    // Set erase latch
PTSH = (unsigned char) ((n*0x40)>>8); // Set sector high address
PTSL = (unsigned char) (n*0x40);         // Set sector low address
FSCMD = 3;   // Set data area erase command
LOCK = 0xAA; // FLASH locking
---------------------------------------------------------------------------------------
```

Note: page number n=0、1、2……

◆ Write data into DATA area

For instance, write data 0xAA to DATA of which address n~(n+63), the program will be:

```
---------------------------------------------------------------------------------------
unsigned char i;
FSCMD = 0; // set CMD=0
LOCK = 0x2A; // unlock DATA area
PTSH = 0;      // Set page latch start address
PTSL = 0;      // Set page latch start address
FSCMD = 8;   // Set erase latch
for(i=0;i<64;i++)
{
    FSDAT = 0xAA; // Write 1 page of data continuously
}
PTSH = (unsigned char)(n>>8); // Set the data first address 8 bits higher
PTSL = (unsigned char)n;      // Set data first address low 8 bits
FSCMD = 2;   // Set Write Command
LOCK = 0xAA; // FLASH locking
---------------------------------------------------------------------------------------
```

Note：

1． When writing data continuously, only the first address needs to be set, and the data pointer register PTS will be accumulated automatically after each FSDAT write.

2． When reading or writing the data area, the address set is the logical address of the data area, not the physical address of FLASH, and the logical address starts from 0.

3. Data can only be written in pages, and 64 bytes must be written at a time.

◆ **Read data from DATA area**

For instance, the pointer dataBuf reads data from DATA area of which address is n~(n+63) to, the program will be:

```
    ---------------------------------------------------------------------------------------
    unsigned char i， dataBuf[64]；
    FSCMD = 0; // Set CMD to 0
    LOCK = 0x2A; // unlock DATA area
    PTSH = (unsigned char)(n>>8); // Set page latch start address
    PTSL = (unsigned char)n;      // Set page latch start address
    FSCMD = 1;   // Set read command
    for(i=0;i<64;i++)
    {
        dataBuf[i] = FSDAT ; // Continuous reading
    }
    FSCMD = 0; // Set CMD to 0
    LOCK = 0xAA; // FLASH locking
    ---------------------------------------------------------------------------------------
```

Note：

1. *When reading out data continuously, only the first address needs to be set, and the data pointer register PTS will be accumulated automatically after each FSDAT read.*
2. *Data readout does not need to be in pages, any number of bytes can be read continuously.*

◆ Sector erasure of PROGRAM area

Sector n of PROGRAM area needs to be erased, for example, the program may as follows:

```
---------------------------------------------------------------------------------------
FSCMD = 0; // Set CMD to 0
```

```
LOCK = 0x29; // Unlock PROGRAM area
FSCMD = 8;   // Set erase latch
PTSH = (unsigned char)((n*0x40)>>8); // Set sector high address
PTSL = (unsigned char)(n*0x40);      // Set sector low address
FSCMD = 7;   // Set erase command
LOCK = 0xAA; // FLASH locking
----------------------------------------------------------------------------------------
```

Note：page number n=0、1、2……

Write data into PROGRAM area

For instance, write data 0xAA to PROGRAM area of which address is n~(n+63), the program will be:

```
----------------------------------------------------------------------------------------
unsigned char i;
FSCMD = 0; // Set CMD to 0
LOCK = 0x29; // Unlock PROGRAM area
FSCMD = 8;   //Set erase latch
PTSH = 0;     //Set page latch start address
PTSL = 0;     //Set page latch start address
for(i=0;i<64;i++)
{
      FSDAT = 0xAA; //Write 1 page of data continuously
}
PTSH = (unsigned char)(n>>8);    //Set the data first address 8 bits higher
PTSL = (unsigned char)n;         //Set data first address low 8 bits
FSCMD = 6;   //Set Write command
LOCK = 0xAA; //FLASH LOCKING
----------------------------------------------------------------------------------------
```

Note：

1．When writing data continuously, only the first address needs to be set, and the data pointer register PTS will be accumulated automatically after each FSDAT write.

2. Data can only be written in pages, and 64 bytes must be written at a time.

Read data from PROGRAM area

For instance, the pointer pBuf reads data from PROGRAM area of which address is n~(n+63) to pointer dataBuf, the program will be:

```
----------------------------------------------------------------------------------------
unsigned char i，dataBuf[64];
FSCMD = 0; // Set CMD to 0
LOCK = 0x29; // Unlock PROGRAM area
```

```
PTSH = (unsigned char)(n>>8); //Set the data first address 8 bits higher
PTSL = (unsigned char)n;      //Set data first address low 8 bits
FSCMD = 5;   // Set read command
for(i=0;i<64;i++)
{
      DataBuf[i] = FSDAT ; // Continuous writing of data
}
FSCMD = 0; // Set CMD to 0
LOCK = 0xAA; //FLASH LOCKING
-------------------------------------------------------------------------------------------
```

Note:

1. When reading out data continuously, only the first address needs to be set, and the data pointer register PTS will be accumulated automatically after each FSDAT read.

2. Data readout does not need to be in pages, and any number of bytes can be read continuously.

7.4 External RAM Mapped to Program Area

The 768 bytes external RAM can be mapped for PROGRAM area use, the mapping address is 4000H~42FFH, and the mapping diagram is shown in Figure 7-4-1. The user can download the program to the external RAM, When the program runs directly execute the jump instruction to jump to the mapped program area for execution. For the same effect, the value of BOOT (see register MECON for details) can also be set to 1, and then a soft reset is executed, and the program starts to execute from the external RAM after the reset (the mapped address is 0000H~02FFH at this time). The mapped program area is particularly convenient for implementing functions such as IAP.

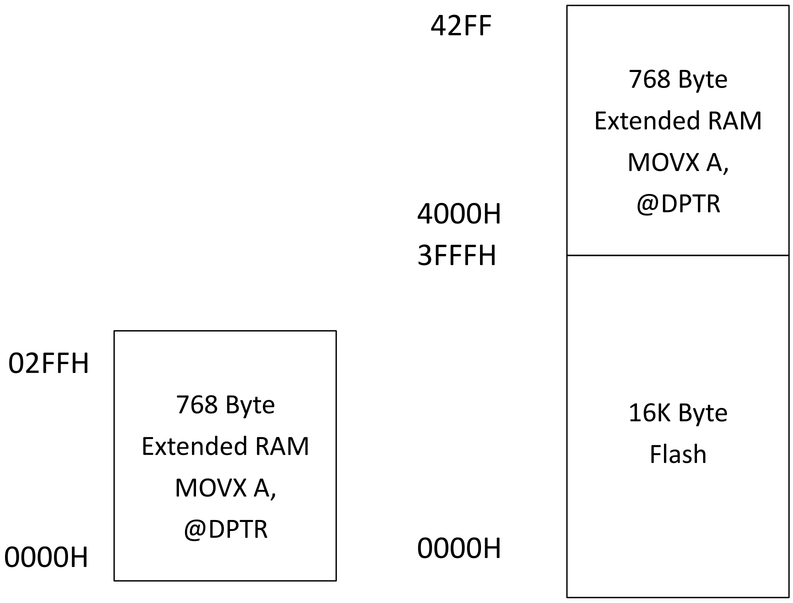


Figure 7-4-1 XRAM Address mapping

8 Interrupt System

8.1 Function Introduction

CA51F7 series chips have an enhanced interrupt control system with 7 interrupt entries, each with several interrupt sources, each with 2 levels of interrupt priority. Each interrupt source has its own interrupt vector, priority setting bit, interrupt enable bit, and interrupt flag, and after responding to the interrupt, the CPU will enter the interrupt service program corresponding to the interrupt and return to the pre-interrupt state after receiving the RETI instruction. If more than one valid interrupt generates interrupt requests at the same time, the CPU will respond in order according to the set interrupt priority; if the priority is the same, it will respond in order according to their natural priority (interrupt entry address from lowest to highest).

8.2 Interrupt Logic

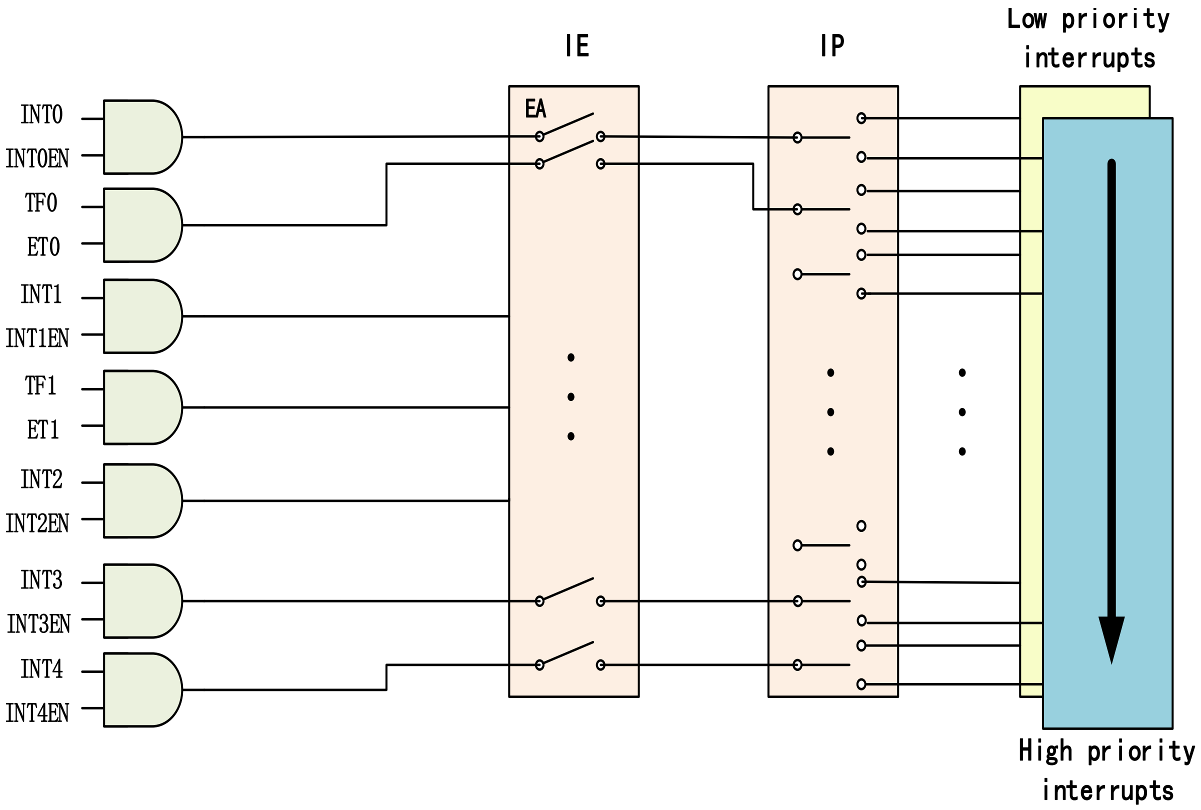


Figure 8-2-1 Interrupt Logic

8.3 Interrupt Vector Table

Table 8-3-1 Interrupt Vector Table

Interrupt	Interrupt source	Vector	Default Priority
INT0	INT0	03H	0
TF0	Timer 0	0BH	1
INT1	INT1	13H	2
TF1	Timer 1	1BH	3
INT2	UART/External Interrupt 2 /ADC Interrupt	23H	4
INT3	External Interrupt 3/Touch Key Interrupt/TMC Interrupt	2BH	5
INT4	External interrupt 4/WDT interrupt/I2C interrupt/LVD interrupt	33H	6

8.4 Interrupt Control Register

Table 8-4-1 Register IE

A8H	7	6	5	4	3	2	1	0
IE	EA	INT4EN	INT3EN	INT2EN	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit symbol	Description
7	EA	Global interrupt enable control bit 0：disable Global Interrupt 1：enable Global Interrupt
6	INT4EN	Interrupt 4 enable control bit （ Interrupt 4 for WDT/I2C/LVD/External Interrupt 4 ） 0：Disable 1：Enable
5	INT3EN	Interrupt 3 enable control bit （ Interrupt 3 for TMC/TK/External Interrupt 3） 0：Disable 1：Enable
4	INT2EN	Interrupt 2 enable control bit （ Interrupt 2 for UART/ADC/External Interrupt 2） 0：Disable 1：Enable
3	ET1	Timer 1 interrupt enable control bit 0：Disable Timer 1 Interrupt 1：Enable Timer 1 Interrupt
2	EX1	Interrupt 1 enable control bit （ Interrupt 1 for external interrupt 1） 0：Disable 1：Enable
1	ET0	Timer 0 interrupt enable control bit 0：Disable Timer 0 Interrupt 1：Enable Timer 0 Interrupt
0	EX0	Interrupt 0 enable control bit (interrupt 0 is used for external interrupt 0) 0：Disable 1：Enable

Note：The enable control bit of IE is corresponding to the interrupt vector, and the interrupt switch of each interrupt source should be turned on separately. For example, to turn on the interrupt of external interrupt 2, in addition to setting INT2EN to 1, EPIE0 (external interrupt 2 enable bit) should also be set to 1.

Table 8-4-2 Register IP

B8H	7	6	5	4	3	2	1	0
IP	-	PS1	PT2	PS0	PT1	PX1	PT0	PX0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit symbol	Description
7	-	-
6	PS1	Interrupt INT4 priority control bit 0：low priority 1： high priority
5	PT2	Interrupt INT3 priority control bit 0： low priority 1： high priority
4	PS0	Interrupt INT2 priority control bit 0：low priority 1：high priority
3	PT1	Timer 1 priority control bit 0：low priority 1：high priority
2	PX1	External interrupt 1 priority control bit 0：low priority 1：high priority
1	PT0	Timer 0 priority control bit 0：low priority 1：high priority
0	PX0	External interrupt 0 priority control bit 0：low priority 1：high priority

8.5 External Interrupt

8.5.1 External Interrupt Introduction

The interrupt pins of INT0/INT1 are P1.6/P1.7 respectively, which are basically compatible with the standard 8051; in addition, the system also extends three interrupt entries INT2~INT4 as external interrupts. Either rising or falling edge trigger can be selected. The External interrupt can also be wakened up in STOP mode. EPIF is the external interrupt status register of INT2~INT4, and the configuration registers of INT2~INT4 are EP0CON~EP2CON.

Note: INT0 and INT1 can be selected to be triggered by rising or falling edge, the selection bits are IT0 and IT1 respectively, see the description of register TCON for details.

8.5.2 External Interrupt Register

Table8-5-1 Register EPIF

D4H	7	6	5	4	3	2	1	0
EPIF	-	-	-	-	-	EPIF2	EPIF1	EPIF0
R/W	-	-	-	-	-	R/W	R/W	R/W
Initial Value	-	-	-	-	-	0	0	0

Bit number	Bit symbol	Description
7~3	-	-
2	EPIF2	External Interrupt 4 Interrupt Flag, cleared when 1 is written to it
1	EPIF1	External Interrupt 3 Interrupt Flag, cleared when 1 is written to it
0	EPIF0	External Interrupt 2 Interrupt Flag, cleared when 1 is written to it

Table 8-5-2 Register EPCON

D1H	7	6	5	4	3	2	1	0
EP0CON	EPIE0	EPPL0	-	-	-	-	-	-
R/W	R/W	R/W	-	-	-	-	-	-
Initial value	0	0	-	-	-	-	-	-
D2H	7	6	5	4	3	2	1	0
EP1CON	EPIE1	EPPL1	-	-	-	-	-	-
R/W	R/W	R/W	-	-	-	-	-	-
Initial value	0	0	-	-	-	-	-	-
D3H	7	6	5	4	3	2	1	0
EP2CON	EPIE2	EPPL2	-	-	-	-	-	-
R/W	R/W	R/W	-	-	-	-	-	-
Initial value	0	0	-	-	-	-	-	-
Note: "n" in the following table means 0/1/2								

Bit number	Bit symbol	Description
7	EPIEn	External interrupt enable bit 0：Disable 1：Enable *Note: n=0/1/2 corresponds to external interrupts 2/3/4 respectively*
6	EPPLn	External interrupt trigger edge select bit 0：Rising edge 1：Falling edge *Note: n=0/1/2 corresponds to external interrupts 2/3/4 respectively*
5~0	-	-

8.5.3 External Interrupt Control Method and Examples

- **External Interrupt 0/1 control example**

For instance, to enable external interrupt 0, the program is as follows:

```
-------------------------------------------------------------------------------------
void INT0_init(void)
{
        P16F = 1;       // The interrupt pin of external interrupt 0 is P16, set P16 as input pin
        EX0 = 1;        // INT0 interrupt enable
        IE0 = 1;        // External interrupt 0 enable
        IT0 = 1;        // Set to falling edge trigger
        PX0 = 1;        // Set INT0 to high priority
        EA = 1;         // /Enable Global Interrupt
}
void INT0_ISR (void) interrupt 0
{
        // External interrupt 0 interrupt service program
}
-------------------------------------------------------------------------------------
```

For example, to enable external interrupt 1, the program is as follows:

```
-------------------------------------------------------------------------------------
void INT1_init(void)
{
        P17F = 1;       // The interrupt pin of external interrupt 1 is P17, set P17 as input function
        EX1 = 1;        // INT1 interrupt enable
        IE1 = 1;        // External interrupt 1 enable
        IT1 = 1;        // Set to falling edge interrupt
        PX1 = 1;        // Set INT1 to high priority
        EA = 1;         // Enable Global Interrupt
}
void INT1_ISR (void) interrupt 2
{
        // External interrupt 1 interrupt service program
}

-------------------------------------------------------------------------------------
```

- **External interrupt 2~4 control example**

Take external interrupt 2 as an example, the program is as follows:

```
-------------------------------------------------------------------------------------
void INT2_init(void)
{
        P00F = 1;                            // Set P00 as input pin
        EP0CON = (1<<7) | (0<<6) ;          // Enable external interrupt 2 and set to rising edge trigger
        INT2EN = 1;                         //Enable INT2 interrupt
        EA = 1;                             // Enable Global Interrupt
}
void INT2_ISR (void) interrupt 4
{
        if(EPIF & 0x01)             // Judge the Interrupt Flag for External Interrupt 2
        {
                EPIF = 0x01;    // Write 1 to the Interrupt Flag to clear it
                // External interrupt 2 interrupt service program
                ......
        }
}
```

----

9 Clock System

9.1 Clock System Introduction

The CA51F7 series of chips support the following clock sources in total:

- 16MHz Internal RC Oscillator
- 100KHz internal RC Oscillator
- 20~40MHz Programmable Internal RC Oscillator

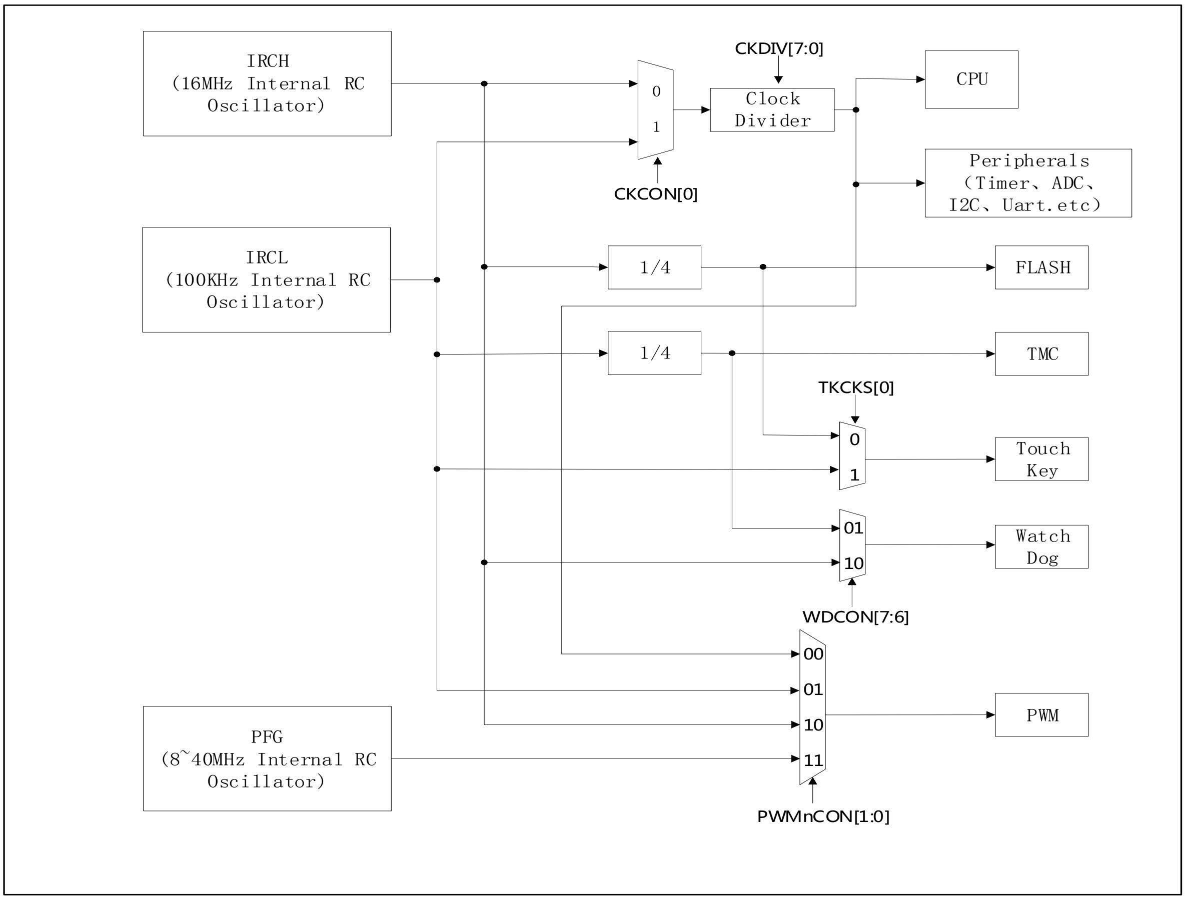


Figure 9-1-1 Clock Structure

Users can manage each clock source independently, and each clock source can be turned on or off individually, thus allowing flexible control of power consumption.

9.1.1 Clock Special name definition

Name abbreviation	Description
IRCH	16MHz Internal RC Oscillator
IRCL	100KHz internal RC Oscillator
PFG	20~40MHz Programmable Internal RC Oscillator

9.1.2 16MHz Internal RC Oscillator（IRCH）

IRCH is the default system clock after power on, which can be enabled or disabled by setting IHCKE bit of register CKCON. The clock accuracy error is 1% maximum, and the factory frequency is 16MHz@3.3V/25℃.

9.1.3 100KHz internal RC Oscillator（IRCL）

IRCL can be enabled or disabled by setting ILCKE bit of register CKCON. IRCL is set as the system clock to achieve low system power consumption. IRCL has a clock accuracy of ±25%@3.3V/25°C.

9.1.4 Internal PFG oscillator

The PFG oscillator is enabled by the PFGCKE and is primarily used as a clock source for PWM, specifically designed for humidifier applications.

- **PFG calibration mechanism**

The CA51F7 calibration module takes the system clock as the reference and can count or measure the PFG clock, then the application software calculates the current PFG clock frequency based on the count or measurement value, and then the frequency is adjusted. The application software then calculates the current PFG clock frequency based on the count or measurement value, and then performs the frequency adjustment. Repeated measurement and adjustment can eventually make the PFG clock frequency close to the target frequency.

The calibration module has two modes of operation: Count mode and Measure mode
Count mode is used to manually measure the PFG clock frequency. The PFG clock counting is started after setting MODE (RCCON[7:6]) to 1, and counting is stopped after MODE is set to 0. The count value is stored in register RCMS (RCMSHH/RCMSHL/RCMSLH/RCMSLL) after stopping counting. In the application, the user can start the PFG clock count within a determined time period, and the frequency of the PFG can be obtained by a simple calculation of the count value RCMS.

The measure mode counts the PFG clock in several system clock cycles, and the PFG clock frequency is derived from the count value. In order to improve the measurement accuracy, the measurement period should be extended as long as possible by configuring the register VCKD (VCKDH/VCKDL) and setting a measurement period to be VCKD times the system clock period. In this way, the frequency of PFG can be obtained after a simple calculation of the count value after the measurement. The calculation formula is as follows.

PFG cycle＝ (System clock cycle x VCKD) ÷RCMS

- **PFG clock frequency hopping function**

The PFG frequency hopping function is specifically designed for humidifier applications and is intended to reduce the magnitude of external RF interference to humidifier products.

PFG frequency hopping takes the calibrated clock frequency as the center point and swings back and forth in steps set by STEP, and the maximum swing amplitude is set by STEPNUM. After each frequency adjustment, the set value of PFG frequency is saved in PFGRT, that is, after the frequency hopping function is enabled, it is actually PFGRT that determines the frequency of PWM. The frequency hopping function is enabled by TRIMEN (RCCON[3]), and each frequency hopping is triggered by PWM, and PWM0NUM can set the number of cycles to trigger PFG adjustment.

For instance, the calibrated PFG clock frequency is 24MHz, the 24MHz corresponding PFGCFG value is the center point, STEP is set to 5, STEMNUM is set to 3, PWM0NUM is set to 2. After the frequency hopping function is enabled, every 3 (PWM0NUM+1) PWM cycles, the PFG clock frequency is adjusted once, and the difference between PFGRT and PFGCFG is changed in the following order Varies in the following order: 5, 10, 15, 10, 5, 0, -5, -10, -15, -10, -5, 0, 5, 10, 15 ......

Note: When you need to change the center point frequency, you must turn off the frequency hopping function first (set TRIMEN=0), and then enable the frequency hopping function after updating the center point frequency.

9.2.2 Calibration module control register

Table 9-2-2-1 Register RCCON

E9H	7	6	5	4	3	2	1	0
RCCON	MODE[1:0]		-	-	TRIMEN	-	-	-
R/W	R/W		-	-	R/W	-	-	-
Initial Value	0	0	-	-	0	-	-	-
Bit number	Bit Symbol	Description						
7~6	MODE	Operating mode selection bits 01： Count mode, when MODE is 0 then exit count mode 10： Measure mode, MODE automatically clears 0 after completion						
-	-	-						
-	-	-						
3	TRIMEN	PFG clock frequency adjustment enable, 1: means turn on the PFG clock frequency change; 0: means the frequency is fixed						
2-0	-	-						

Table 9-2-2-3 Register VCKDL、VCKDH

EBH	7	6	5	4	3	2	1	0
VCKDL	VCKD[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
ECH	7	6	5	4	3	2	1	0
VCKDH	VCKD[15:8]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
Bit number	Bit Symbol	Description						
15~0	VCKD	In the measure mode, the reference clock divides the frequency by a factor of VCKD (VCKD>1)						

Table 9-2-2-4 Register RCTAGL、RCTAGH

8066H	7	6	5	4	3	2	1	0
RCTAGL	RCTAGL[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
8067H	7	6	5	4	3	2	1	0
RCTAGH	RCTAGH[15:8]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
15~0	RCTAG	In measure mode, the target clock divides the frequency by a multiple of RCTAG (RCTAG>=1)

Table 9-2-2-5 Register RCMSLL、RCMSLH、RCMSHL、RCMSHH

F4H	7	6	5	4	3	2	1	0
RCMSLL	RCMS[7:0]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
F5H	7	6	5	4	3	2	1	0
RCMSLH	RCMS[15:8]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
F6H	7	6	5	4	3	2	1	0
RCMSHL	RCMS[23:16]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
F7H	7	6	5	4	3	2	1	0
RCMSHH	RCMS[31:24]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
31~0	RCMS	When the count mode is completed, the count result is stored When the measure mode is completed, the measurement results is stored

9.2.3 Clock Control Register Description

Table 9-2-2-6 Register IHCFG

CAH	7	6	5	4	3	2	1	0
IHCFG	IHCFG[7:0]							
R/W	R/W							
Initial Value	-	-	0	0	0	0	0	0
Bit number	Bit Symbol	Description						
7~0	IHCFG	IRCH frequency adjustment register						

Table 9-2-2-7 Register PFGCFGL、PFGCFGH

CBH	7	6	5	4	3	2	1	0
PFGCFGL	PFGCFG[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
CCH	7	6	5	4	3	2	1	0
PFGCFGH	-	-	-	-	PFGCFG[11:8]			
R/W	-	-	-	-	R/W			
Initial Value	-	-	-	-	0	0	0	0
Bit number	Bit Symbol	Description						
15~9	-	-						
11~0	PFGCFG	PFG reference frequency adjustment register						

Table 9-2-2-8 Register PFGTRIML、PFGTRIMH

CDH	7	6	5	4	3	2	1	0
PFGTRIML	PFGTRIM[7:0]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
CEH	7	6	5	4	3	2	1	0
PFGTRIMH	-	-	-	-	PFGTRIM[11:8]			
R/W	-	-	-	-	R			
Initial Value	-	-	-	-	0	0	0	0
Bit number	Bit Symbol	Description						
15~9	-	-						
11~0	PFGTRIM	PFG actual frequency adjustment register						

Table 9-2-2-9 Register STEP

EDH	7	6	5	4	3	2	1	0
STEP	STEP[7:0]							
R/W	R/W							
Initial Value	-	-	0	0-	0	0	0	0
Bit number	Bit Symbol	Description						
7~0	STEP	PFG frequency adjustment step value						

Table 9-2-2-10 Register STEPNUM

EEH	7	6	5	4	3	2	1	0
STEPNUM	STEPNUM[7:0]							
R/W	R/W							
Initial Value	-	-	0	0-	0	0	0	0

Bit number	Bit Symbol	Description
7~0	STEPNUM	PFG frequency step adjustment stage register

9.3 System Clock

System clock control is done by registers CKCON and CKDIV. With these register sets, the switching of each clock source and the switching and dividing of the system clock can be set individually.

The system clock has two clocks to choose from: IRCH and IRCL. After power on, the default system clock is IRCH and the CKDIV value is 1, which means the default system clock power on frequency is one-half of the IRCH frequency. If the CPU needs to run at a higher frequency, the software can set CKDIV to 0.

9.3.1 System Clock Architecture

Please refer to figure 9-3-1

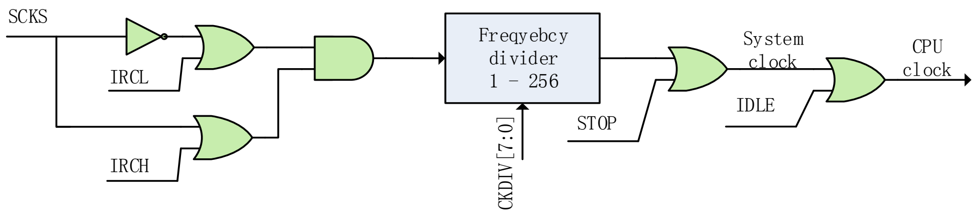


Figure 9-3-1 System Clock Architecture

9.3.2 System Clock Control Register Description

Table 9-3-1 Register CKCON

C8H	7	6	5	4	3	2	1	0
CKCON	IHCKE	ILCKE	PFGCKE	-	-	-	-	SCKS
R/W	R/W	R/W	R/W	-	-	-	-	R/W
Initial Value	0	0	0	-	-	-	-	0
Bit number	Bit Symbol	Description						
7	IHCKE	IRCH enable control bit 1: Enable 0: Disable *Note:* *When this bit is 1, the clock module is turned on, but when this bit is 0, the clock will still be turned on if the system or another module selects this clock source.*						
6	ILCKE	IRCL enable control bit 1： Enable 0： Disable *Note:* *When this bit is 1, the clock module is turned on, but when this bit is 0, the clock will still be turned on if the system or another module selects this clock source.*						
5	FPGCKE	FPG enable control bit 1： Enable 0： Disable						
1-0	SCKS	System clock selection bit 0： Select IRCH 1： Select IRCL						

Table 9-3-2 Register CKDIV

C9H	7	6	5	4	3	2	1	0
CKDIV	CKDIV[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	1
Bit number	Bit Symbol	Description						
7~0	CKDIV	System clock frequency division 00H： No division 01H： frequency divided by 2 02H： frequency divided by 3 03H： frequency divided by 4 …… FFH： frequency divided by 256						

9.3.3 System clock control methods and Example

- **Set IRCH as the system clock**

  To set IRCH as the system clock. The program is as follows:

```
--------------------------------------------------------------------------------
#define IHCKE              (1<<7)
#define CKSEL_IRCH     0
void Sys_Clk_Set_IRCH(void)
{
        CKCON |= IHCKE;                                      // Enable IRCH clock
        CKCON = (CKCON&0xFE) | CKSEL_IRCH;        // Set IRCH as system clock
}
--------------------------------------------------------------------------------
```

- **Set IRCL as the system clock**

  To set IRCL as the system clock. The program is as follows:

```
--------------------------------------------------------------------------------
#define ILCKE         (1<<6)
#define CKSEL_IRCL     1
void Sys_Clk_Set_IRCL(void)
{
        CKCON |= ILCKE;                                      // Enable IRCL
        Delay_ms(1);
        CKCON = (CKCON&0xFE) | CKSEL_IRCL;         // Set IRCL as system clock
}
--------------------------------------------------------------------------------
```

10 Power Supply and Reset System

10.1 Power Supply

A 2.2V - 5.5V power supply is connected to the VDD and VSS pins of the CA51F7 family of chips. This power supply can directly power the internal digital and analog systems of the chip. Note that the maximum frequency and power consumption supported by the chip are not the same for different supply voltages, as described in the Electrical Specification.

The chip is also designed with a BANDGAP reference voltage as a reference voltage source for the ADC internal reference voltage, LVD voltage, touch internal op-amp, etc. This reference voltage source is factory calibrated with an accuracy of ±30mV.

10.1.2 Internal Reference Voltage Control Register

Table 10-1-2-1 Register PWCON

86H	7	6	5	4	3	2	1	0
PWCON	FLEVEL[3:0]				**VREFS**	-	-	-
R/W	R/W				**R/W**	-	-	-
Initial Value	0	1	1	1	**0**	-	-	-

Bit number	Bit Symbol	Description
7~4	FLEVEL	Internal reference voltage (Bandgap) output adjustment bit field 0000：0.825V 0001：0.850V 0010：0.875V 0011：0.900V 0100：0.925V 0101：0.950V 0110：0.975V 0111：1.000V 1000：1.025V 1001：1.050V 1010：1.075V 1011：1.100V 1100：1.125V 1101：1.150V 1110：1.175V 1111：1.200V *Note: This reference voltage is factory calibrated with an accuracy of ±30mV, the calibration value is automatically loaded on power. The user is not allowed to change it.*
3	VREFS	Reference voltage drive selection. 0: 0.8uA drive capability, default 1: Op-amp driver
2-0	-	-

10.2 Reset System

The CA51F7 series chips have multiple internal and external reset sources, as shown in Figure 10-2-1.

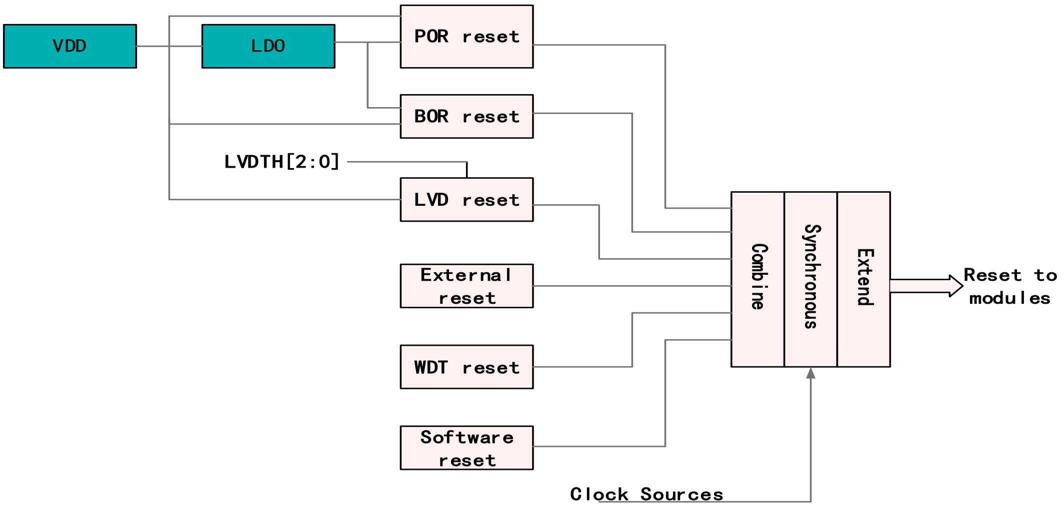


Figure 10-2-1 Reset system Architecture

- **Power On Rese（POR）**

The system power on presents a gradually rising curve form, and it takes some time to reach the normal operating voltage. The POR is based on VDD, and the POR signal is valid when the voltage is below the detection threshold.

The POR circuit ensures that the chip is in a reset state during the power on process and that the chip can start operating from a known stable state after power on. The POR signal is also spread by the chip's internal counter to ensure that the various internal analog modules can enter a stable operating state after power on.

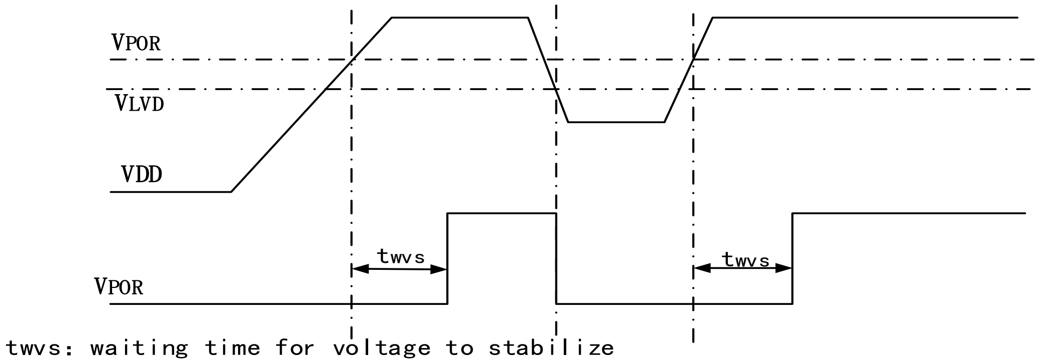
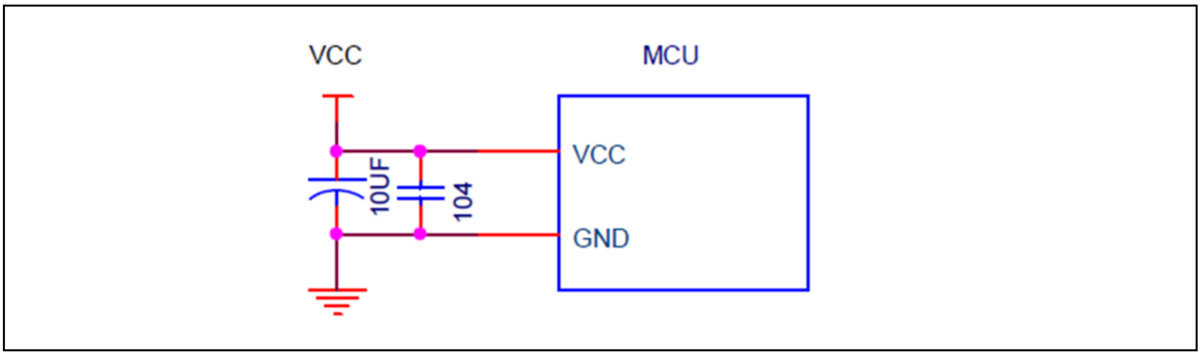


Figure 10-2-2 POR Circuit Example and Power On Stage



10-1-2 Typical circuit diagram of chip power supply

***Important Reminder*：**

1. In the above circuit, the filter capacitor 10uF and 104 is standard for the chip power supply circuit and cannot be omitted. This capacitor must be placed close to the chip power supply pins, otherwise it may cause the chip to work abnormally.

2. The above circuit and component parameters are for reference only and may need to be modified according to the peripheral operating environment and different voltage supply parameters.

- **Brown Out Reset（BOR）**

BOR offers alarm signal for the chip when the voltage drops (eg. Inference or load changes). Once the VDD or internal LDO output voltage is below a certain threshold, it will reset the chip to avoid program error or system abnormality.

- **Low Voltage Reset**

The Low Voltage Detection (LVD) can detect VDD in multiple working modes. When VDD voltage is below the threshold set by LVD for 20us it will generates reset signal (on the premise of that LVD is Reset mode).

- **External Reset**

By pulling down the reset pin(RESET), external device can reset the chip as well. RESET can reset the whole in normal working modes, while in STOP mode, the hard reset will awaken the chip first and then reset it. Usually, RESET is pulled up internally and will not influence the internal reset circuit.

- **Watchdog Reset**

The WDT (watchdog timer) is responsible for monitor the how processor do with instructions. With proper configuration, if the WDT is not refreshed in certain time, a reset signal will be generated. WDT is disabled after POR, but users can enable and configure it if

- **Soft Reset**

The program can soft reset the chip. When 1 is written to SWRST of register PCON, CPU sends out reset signal.

Power on power down reset and external hard reset will reset all circuits, LVD and WDT reset cannot reset their own circuits, but can reset other circuits (e.g. after WDT reset is generated, WDT module circuits are not reset, WDT registers remain in the state before reset, but circuits outside WDT have been reset). neither LVD/WDT nor soft reset can reset memory control circuits Neither LVD/WDT nor soft reset can reset the memory control circuit. After a soft reset, the program will start running from the location pointed by the BOOT configuration. After all resets are generated, the PC will point to address 0.

11 Power Consumption Management

The CA51F7 series chips have three different low power modes: IDLE, STOP and Low Speed mode. The system current is less than 30uA in IDLE mode, less than 5uA in STOP mode, and less than 70uA in Low Speed mode.

11.1 IDLE Mode

CPU stops working in this mode. All the clocks can be disabled to save power before entering IDLE mode except the main clock. Peripherals can also be enabled/disabled before entering IDLE mode according to user’s needs. Those enabled peripherals will operates normally in IDLE mode.

Register IDLST(IDLSTH and IDLSTL) needs to be checked before entering IDLE mode. If all the bits are 0, CPU will enter IDLE mode normally when the mode is set as IDLE. However, if NOT all the bits are 0, CPU will not enter IDLE mode and remains in normal working mode although the mode is set as IDLE. To deal with this situation, users must complete the IDLST corresponding interrupt processing first and then set the mode as IDLE again.

Any reset or interrupt will awake the chip. The clock will resume first and then the chip responds to the interrupt and enters the interrupt service routine after the CPU awakening. After the chip exits interrupt service routine ,it will execute the instructions after the instruction which set IDLE to 1. When it exits IDLE mode, IDLE will be cleared automatically.

What must be mentioned is that there should be two “nop” instructions after setting IDLE to avoid program error.

11.2 STOP Mode

The STOP mode can be seen as a deeper low power consumption mode than IDLE. STOP mode is able to stop all the clocks (include the main clock) and clock generation circuits. If WDT and RTC are enabled, their clock module will still work, hence users may disable them to save power.

Similar to IDLE mode, before entering STOP mode, you need to check the STPST (STPSTH and STPSTL) registers, if there are bits set to 1, you need to handle them first to ensure that you can enter STOP mode smoothly.

The STOP mode can be awoken by external interrupt, LVD reset or interrupt, hard reset, RTC interrupt, WDT interrupt or reset, clock monitor interrupt and touch key interrupt. If it is awoken by an interrupt, the chip will esume clock first and respond to the interrupt, and then enters corresponding the interrupt service routine. After the chip exits the interrupt service routine, it will execute the instructions after the setting STOP to I nstruction. The STOP will be cleared automatically when the chip exits STOP mode.

To arouse the chip better, it is recommended to set the internal clock as system clock before entering STOP mode because it will take longer time waiting for stable status when using external clock.

When the chip enters STOP mode, the last clock edge will disable system clock and then the chip enters STOP mode entirely. What must be mentioned is that there should be three “nop” instructions after setting STOP to avoid program error.

11.3 Low Speed Mode

Since the power consumption is influenced by the its speed, so it will reduce the power consumption effectively

if the main clock runs with low speed(131kHz IRCL). The current will be less than 70uA if IRCL is set as the system clock.

11.4 Low Power Related Register Description

Table 11-4-1 Register PCON

87H	7	6	5	4	3	2	1	0
PCON	-	-	SWRST	-	-	TSMODE	STOP	IDLE
R/W	-	-	W	-	-	R	W	W
Initial Value	-	-	0	-	-	0	0	0

Bit number	Bit Symbol	Description
7~6	-	-
5	SWRST	Soft reset control ,1enable SWRST mode Setting SWRST=1 will generate soft reset signal, and it will be cleared to 0 automatically after the reset
4~3	-	-
2	TSMODE	Online test mode flag bit, 1 indicates the chip is working in online test mode
1	STOP	STOP mode control bit, and 1 enable STOP mode When STOP=1 and STPST=0, the chip will enter STOP mode. It will be cleared to 0 automatically after the chip exits STOP mode
0	IDLE	IDLE mode control, 1 enables IDLE mode When IDLE=1and IDLST=0, the chip will enter IDLE mode It will be cleared to 0 automatically after the chip exits IDLE mode

Table 11-4-2 Register IDLST

8EH	7	6	5	4	3	2	1	0
IDLST	-	IDLSTL[6:0]						
R/W	-	R						
Initial Value	-	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7	-	-
6	I2CINT/WDIF/LVDINT/EPIF[2]	Interrupt status of I^{2}C/WDT/LVD/External Interrupt 4 in IDLE mode
5	TKINT/TMINT/EPIF[1]	Interrupt status of touch key/TMC/external interrupt 3 in IDLE mode
4	UART/EPIF[0]/ADC	Interrupt status/ADC of UART/External Interrupt 2 in IDLE mode

3	TF1	Interrupt status of timer 1 in IDLE mode
2	PIF[1]	Interrupt status of external interrupt 1 in IDLE mode
1	TF0	Interrupt status of timer 0 in IDLE mode
0	PIF[0]	Interrupt status of external interrupt 0 in IDLE mode

Table 11-4-3 Register STPST

8FH	7	6	5	4	3	2	1	0
STPST	-	STPSTL [6:0]						
R/W	-	R						
Initial Value	-	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7	-	-
6	WDTWKF/LVDWKF/I2CWKF	Interrupt status of WDT/LVD/I2C in STOP mode
5	TKWKF/TMWKF	Interrupt status of touch button/TMC in STOP mode
4	EPWKF[2]	Interrupt status of external interrupt 4 in STOP mode
3	EPWKF[1]	Interrupt status of external interrupt 3 in STOP mode
2	EPWKF[0]	Interrupt status of external interrupt 2 in STOP mode
1	PWKF[1]	Interrupt status of external interrupt 1 in STOP mode
0	PWKF[0]	Interrupt status of external interrupt 0 in STOP mode

11.5 Low Power Consumption Control Example

- **STOP Mode Example**

The STOP mode program is as follows:

```
----------------------------------------------------------------------------------------------
#define IHCKE           (1<<7)
#define ILCKE           (1<<6)

#define CKSEL_IRCH      0
#define CKSEL_IRCL      1

void Stop(void)
{
    bit IE_EA;
    I2CCON = 0;         // disable I2C for it is the default enabled, otherwise the IRCH cannot be disabled
    CKCON = 0;          // disable all the clocks
    MECON |= (1<<6);    // set FLASH in deep sleep mode
    IE_EA = EA;         // Save global interrupt enable bit status
    EA = 0;
    PCON = (PCON&0x04) | 0x02;          // enters STOP mode
    _nop_();
    _nop_();
    _nop_();
    EA = IE_EA;
}
----------------------------------------------------------------------------------------------
```

IDLE Mode Example

The IDLE mode program is as follows.

```
------------------------------------------------------------------------------------------------
#define IHCKE            (1<<7)
#define ILCKE            (1<<6)

#define CKSEL_IRCH    0
#define CKSEL_IRCL    1

void Idle(void)
{
        CKCON |= ILCKE;                                    // Enable IRCL
        Delay_ms(1);
        CKCON = (CKCON&0xFE) | CKSEL_IRCL;         // Set IRCL as system clock
        I2CCON = 0;   // disable I2C for it is the default enabled, otherwise the IRCH cannot be disabled
        CKCON &= ~IHCKE;     // disable IRCH clock
        MECON |= (1<<6);         // set FLASH in deep sleep mode
        PCON = (PCON&0x04) | 0x01;        //enters IDLE mode
        _nop_();
        _nop_();
}
```

Note : Since the main clock is still on after entering IDLE, if the main clock is a high-speed clock before entering IDLE, the power consumption will still be high after entering IDLE mode, so you need to switch the main clock to a low-speed clock before entering IDLE.

Low Speed Mode Example

The low-speed operation mode program is as follows.

```
------------------------------------------------------------------------------------------------
#define IHCKE            (1<<7)
#define ILCKE            (1<<6)

#define CKSEL_IRCH    0
#define CKSEL_IRCL    1

void LowSpeedMode(void)
{
        I2CCON = 0;              // disable I2C for it is the default enabled, otherwise the IRCH cannot be disabled
        CKCON = ILCKE;                // enable IRCL clock and disable other clocks
        Delay_ms(1);
        CKCON |= CKSEL_IRCL;     // switch system clock to IRCL
}
------------------------------------------------------------------------------------------------
```

12 General Timer(Timer0, Timer1)

12.1 Timer0

12.1.1 Timer0 Introduction

The timer/counter function can be selected by CT0 (TMOD[2]). When CT0=0 it operates as a timer; when CT0=1, it functions as a counter. As a timer, its clock is the system clock with frequency divided by 12. As a counter, its clock is the input clock for T0. Because it takes 2 clock cycles to detect the T0 input signal edge change, so when it operates as a counter, the maximum input baud rate is 1/2 of the internal system clock frequency. There is no limit for T0 input signal's duty cycle. However, in order to identify the 0 and 1 clearly, the signal has to keep for at least one internal system clock cycle. Timer 0 has 4 operating modes, which are selected by T0M0, T0M1 bits (TMOD[1:0]).

- **Mode0**

Timer 0 is a 13 bits timer/counter in this mode. The higher 8 bits are stored in TH0 and the lower 5 bits are stored in TL0[4:0] with TL0[7:5] invalid and should be ignored when reading. When Timer0 overflows, the interrupt flag TF0 (TCON[5]) will be set to 1. TF0 will be cleared automatically after the interrupt response. When GATE0 (TCON[3])=0, the timer/counter's is enabled/disabled by TR0 (TCON[4]). When GATE0=1, the timer/counter's is enabled/disabled by INT0. INT0 signal with high level with enable the counting and vice verse.

- **Mode1**

Timer0 is a 16 bits timer/counter in this mode. The function is the same as Mode0.

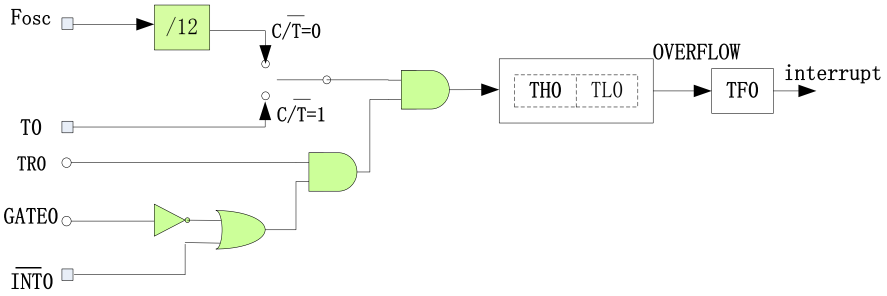


Figure 12-1-1-1 Timer0 Mode0/1

- **Mode2**

Timer0 is a 8 bit automatic reload counter/timer in this mode and only TL0 counts up automatically. When TL0 count overflows, there will be an interrupt flag TF0. The initial value for the count will be reloaded to TL0 from TH0 as well. The other settings are the same as mode0/1.

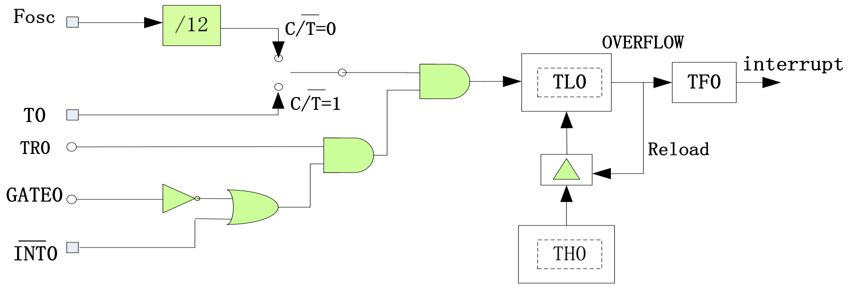


Figure 12-1-1-2 Timer0 Mode2

- **Mode3**

TL0 and TH0 are two independent 8 bits counter/timer in this mode. TL0 can be used as timer or counter while TH0 can only be used as counter. TL0 will be controlled by CT0,GATE0,TR0,TF0 and INT0 and TH0 will only be controlled by TR1 and TF1. The control method is the same as mode0/1. When Timer0 is working in mode3, Timer1 and TH0 both are controlled by TR1. Due to TF1 is used for TH0 already, at the same time, Timer1 can only be used when there is no need for interrupt.(eg, UART baud rate generation)

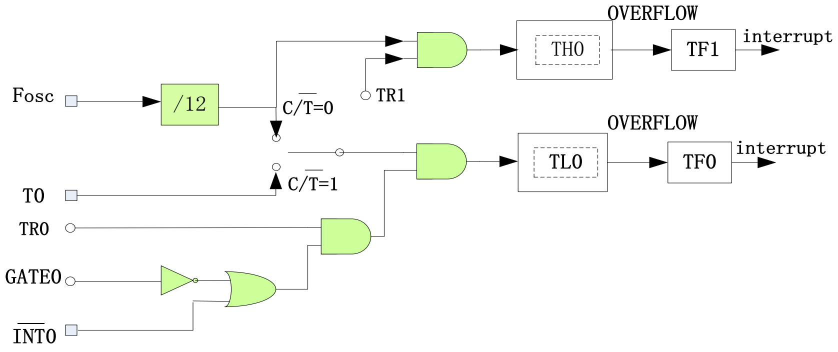


Figure 12-1-1-3 Timer0 Mode3

12.1.2 Timer0 Register Description

Table 12-1-2-1 Register TCON

88H	7	6	5	4	3	2	1	0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7	TF1	Timer0 TH0 overflow flag in mode3 /Timer1 overflow flag, it is cleared automatically after the interrupt response
6	TR1	Timer1 enable control bit, 1 enables it
5	TF0	Timer0 overflow flag bit, it is cleared automatically after the interrupt response
4	TR0	Timer0 enable control bit, 1 enables it
3	IE1	External Interrupt1 enable control bit, 1 enables it
2	IT1	External Interrupt1 trigger type control bit 0： External interrupt 1 is triggered on the rising edge of the input pin 1： External interrupt 1 is triggered on the falling edge of the input pin
1	IE0	External Interrupt0 enable control bit, 1 enables it
0	IT0	External Interrupt0 trigger type control bit 0： External interrupt 0 is triggered on the rising edge of the input pin 1： External interrupt 0 is triggered on the falling edge of the input pin

Table 12-1-2-2 Register TMOD

89H	7	6	5	4	3	2	1	0
TMOD	GATE1	CT1	T1M1	T1M0	GATE0	CT0	T0M1	T0M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7	GATE1	Timer1 gating control bit. When it equals 1, Timer1 is enabled/disabled by INT1
6	CT1	Timer1 Counter/Timer selection bit 0：Timer, the clock for it is the system clock with its frequency divided by 12 1：Counter, the clock for it is T1 input clock
5	T1M1	[T1M1,T1M0]for Timer1 mode selection bits 00：mode0, TL1 and TH1 make up a 13 bits Timer/Counter 01：mode1, TL1 and TH1 make up a 16 bits Timer/Counter 10：mode2, TL1 is an 8 bits Timer/Counter, TH1 is the automatic reload register 11：mode3, TH1/TL1 locked in this mode, and it is the same as TR1=0
4	T1M0	
3	GATE0	Timer0 gating control bit. When it equals 1, Timer0 is enabled/disabled by INT0
2	CT0	Timer0Counter/Timer selection bit 0：Timer, the clock for it is the system clock with its frequency divided by 12 1：Counter, the clock for it is T0 input clock
1	T0M1	[T0M1,T0M0] Timer0 mode selection bits 00：mode0, TL0 and TH0 make up a 13 bits Timer/Counter 01：mode1, TL0 and TH0 make up a 16 bits Timer/Counter 10：mode2, TL0 is an 8 bits Timer/Counter, TH0 is the automatic reload register 11：mode3, TL0 and TH0 are two independent 8 bits Timer/Counter
0	T0M0	

Table 12-1-2-3 Register TL0

8AH	7	6	5	4	3	2	1	0
TL0	TL0							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7~0	TL0	Lower byte of Timer0 count value in mode0/1, count value in mode2/3

Table 12-1-2-4 Register TH0

8CH	7	6	5	4	3	2	1	0
TH0	TH0							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7~0	TH0	Higher byte of Timer0’s count value in mode0/1, reload value in mode2, count value in mode3

12.2 Timer1

12.2.1 Timer1 Introduction

The timer/counter function can be selected by CT1 (TMOD[6]). When CT1=0 it operates as a timer; when CT1=1, it functions as a counter. As a counter, its clock is 1he input clock for T1. Because it takes 2 clock cycles to detect the T1 input signal edge change, so when it operates as a counter, the maximum input baud rate is 1/2 of the internal system clock frequency. There is no limit for T1 input signal's duty cycle. However, in order to identify the 0 and 1 clearly, the signal has to keep for at least one internal system clock cycle time. There are for modes for Timer1 which are selected by T1M0 andT1M1 (TMOD[5:4]).

- **Mode0**

Timer 1 is a 13-bit timer/counter in this mode. The higher 8 bits are stored in TH1 and the lower 5 bits are stored in TL1[4:0] with TL1[7:5] invalid and should be ignored when reading. When Timer1 overflows, the interrupt flag TF1 (TCON[7]) will be set to 1. TF1 will be cleared automatically after the interrupt response. When GATE1 (TCON[7])=0, the timer/counter's is enabled/disabled by TR1 (TCON[6]). When GATE1=1, the timer/counter's is enabled/disabled by INT1. INT1 signal with high level with enable the counting and vice verse.

- **Mode1**

Timer1 operates as a 16-bit timer/counter in this mode. TH1 stores the higher 8bits of the 16-bit timer/counter and TL1 stores the lower 8 bits. When Timer1 overflows, the interrupt flag TF1 (TCON[7]) will be set to 1. TF1 will be cleared automatically after the interrupt response. When GATE1 (TCON[7])=0, the Timer/Counter's is enabled/disabled by TR1 (TCON[6]). When GATE1=1, the timer/counter's is enabled/disabled by INT1. INT1 signal with high level with enable the counting and vice verse,

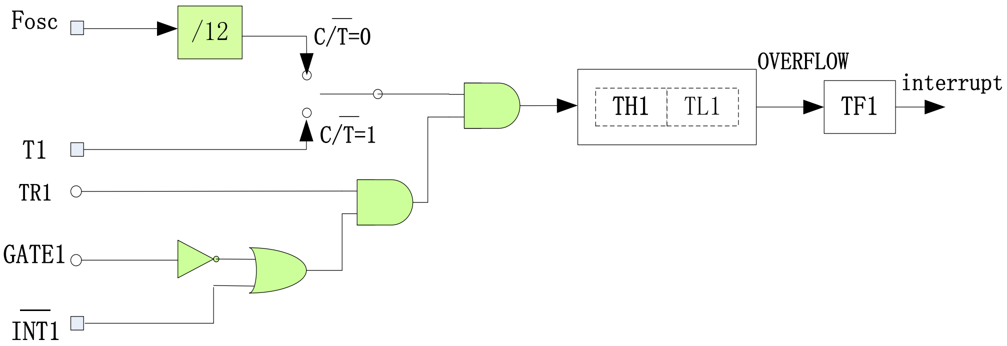


Figure 12-2-1 Timer1 Mode0 and Mode1

- **Mode2**

Timer1 is an 8 bits automatic reload counter/timer in this mode and only TL1 counts up automatically. When TL1 count overflows, there will be an interrupt flag TF1. The initial value for the count will be reloaded to TL1 from TH1 as well. The other settings are the same as mode0/1.

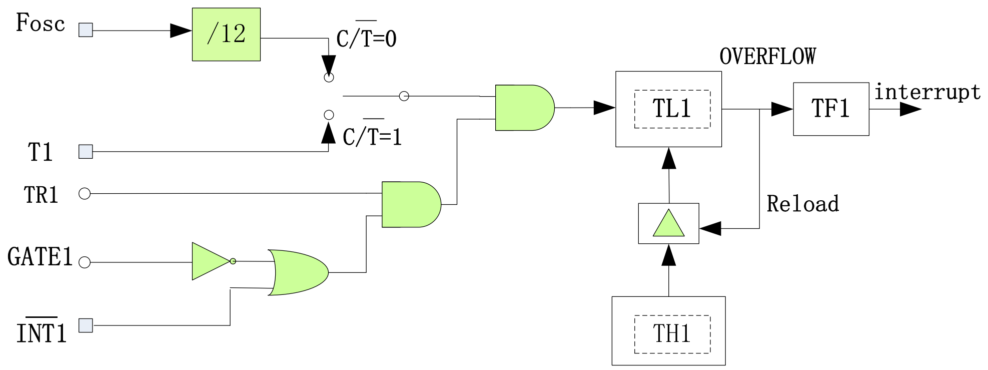


Figure 12-2-2 Timer1 Mode2

- **Mode3**

TH1 and TL1 are locked in this mode, which makes it the same as TR1=0.

12.2.2 Timer1 Register Description

For the register TCON and TMOD please refer to Table12-2-2-1 and Table 12-2-2-2

Table12-2-2-1 RegisterTL1

8BH	7	6	5	4	3	2	1	0
TL1	TL1							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
Bit number	Bit Symbol	Description						
7~0	TL1	Lower byte of Timer1 count value in mode0/1, count value in mode2/3						

Table 12-2-2-2 Register TH1

8DH	7	6	5	4	3	2	1	0
TH1	TH1							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
Bit number	Bit Symbol	Description						
7~0	TH1	Higher byte of Timer1's count value in mode0/1, reload value in mode2, count value in mode3						

13 Watchdog Timer(WDT)

13.1 Watchdog Timer (WDT) Function Introduction

The watchdog timer is a 27-bit backward counter with alternate clock sources. When the clock frequency is 16MHz, the count time can be 0.128ms - 8.388s with 16-bit adjustment precision. The watchdog is mainly used for monitoring the system so that CPU will not break down due to external interference. If the software can not refresh WDT before it overflows, the watchdog will generate internal reset or interrupt. Writing A5H to register WDFLG will refresh the watchdog and reading WDFLG will get the status of the watchdog. If the watchdog is enabled in STOP mode, then the clock selected by the watchdog will works normally. In addition, if the interrupt function is also enabled for watchdog, it will awaken CPU in STOP mode.

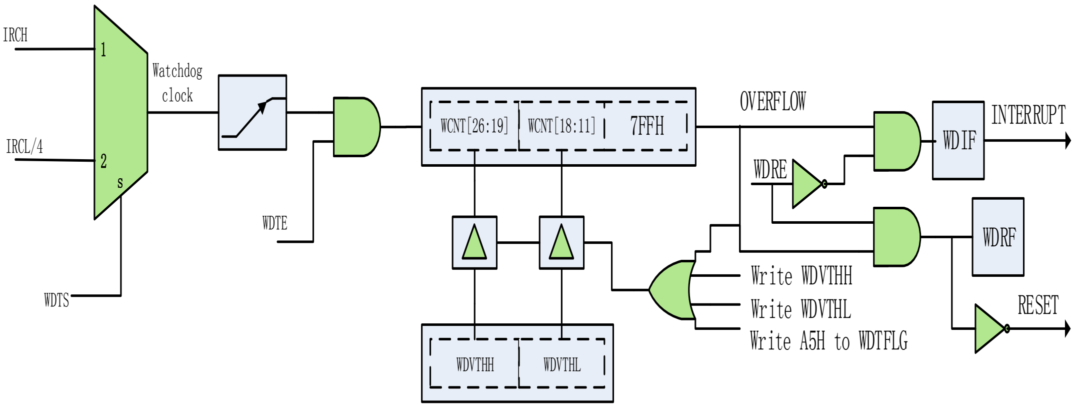


Figure 13-1-1 Watchdog Module Architecture

13.2 Watchdog Timer(WDT) Register Description

Table 13-2-1 Register WDCON

A0H	7	6	5	4	3	2	1	0
WDCON	WDTS[1:0]		-	-	-	-	-	WDRE
R/W	R/W		-	-	-	-	-	R/W
Initial Value	0	0	-	-	-	-	-	0
Bit number	Bit Symbol	Description						
7~6	WDTS	WDT clock selection bits 01： IRCH 10： IRCL with frequency divided by 4 Others： WDT disabled						
5~1	-							
0	WDRE	WDT function selection bit 0： interrupt happens when WDT overflows 1： reset happens when WDT overflows						

Table 13-2-2 Register WDFLG

A1H	7	6	5	4	3	2	1	0
WDFLG							WDIF	WDRF
R/W	-						R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7~2	-	-
1	WDIF	WDT interrupt flag bit, writing A5H to the register will clear it
0	WDRF	WDT reset flag bit, writing A5H to the register will clear it

Table 13-2-3 Register WDVTHL、WDVTHH

A2H	7	6	5	4	3	2	1	0
WDVTHL	WDVTH[7:0]							
R/W	R/W							
Initial Value	1	1	1	1	1	1	1	1
A3H	7	6	5	4	3	2	1	0
WDVTHH	WDVTH[15:8]							
R/W	R/W							
Initial Value	1	1	1	1	1	1	1	1

Bit number	Bit Symbol	Description
15~0	WDVTH	The WDT threshold setting register, the calculation formula is as follows: WDT trigger time ＝ (WDVTH * 800H+7FFH) * clock cycle

13.3 Watchdog Timer Control Example

- **Example for Watchdog interrupt mode**

  For instance, IRCH is set for the watchdog clock and the frequency for it is 16MHz. The watchdog works in reset mode and the overflow time is one second, the program is like：

```
--------------------------------------------------------------------------------------
#define WDTS_IRCH           (1<<6)
#define WDTS_IRCL           (2<<6)

#define WDRE_reset          (1<<0)
#define WDRE_int            (0<<0)
void WDT_init(void)
{
        WDCON = WDTS_IRCH | WDRE_int;    // set the clock as IRCH and watchdog in interrupt mode
        WDVTHH = 0x1E;                   // set one second as the time for watchdog
        WDVTHL = 0x83;
        WDFLG = 0xA5;                    // refresh the watchdog
        INT4EN = 1;                      // enable watchdog interrupt
        EA = 1;                          //enable globe interrupt
}
void WDT_ISR (void) interrupt 6
{
        if(WDFLG & 0x02)
        {
                // watchdog interrupt service program
                WDFLG = 0xA5;            // refresh the watchdog
        }
}
-----------------------------------------------------------------------------------
```

- **Example for watchdog reset mode**

  For instance, IRCH is set for the watchdog clock and the frequency for it is 16MHz. The watchdog works in reset mode and the overflow time is one second, the program is like：

```
--------------------------------------------------------------------------------------
#define WDTS_IRCH           (1<<6)
#define WDTS_IRCL           (2<<6)

#define WDRE_reset          (1<<0)
#define WDRE_int            (0<<0)
void WDT_init(void)
{
        WDCON = WDTS_IRCH | WDRE_reset;          / /set the clock as IRCH and watchdog in reset mode
        WDVTHH = 0x1e;                           // set one second as the time for watchdog
        WDVTHL = 0x83;
        WDFLG = 0xA5;                                    / /refresh the watchdog
}
```

----

14 TMC Timer

14.1 TMC Function Introduction

The clock source for TMC Timer is IRCL. The minimum time unit for its interrupt is 512 IRCL clock cycles, with1~256 minimum time units configurable for the interrupt. TMC interrupt is able to awaken CPU in STOP/IDLE mode.

14.2 TMC Register Description

Table 14-2-1 Register TMCON

D5H	7	6	5	4	3	2	1	0
TMCON	TME	-	-	-	-	-	-	TMF
R/W	R/W	-	-	-	-	-	-	R
Initial value	0	-	-	-	-	-	-	0
Bit number	Bit Symbol	Description						
7	RTCE	TME module enable control bit, 1 enables it						
6~1	-	-						
0	TMF	TMC interrupt flag bit, 1 enables it, cleared when 1 is written to it						

Table 14-2-2 Register TMSNU

D6H	7	6	5	4	3	2	1	0
TMSNU	TMSNU[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	-
Bit number	Bit Symbol	Description						
7~0	TMSNU	TMC interrupt time configuration register，the time for TMC interrupt is （TMSNU+1）×512×Tircl *Note：Tircl is one clock cycle period of IRCL*						

14.3 TMC Control Example

Set the minimum time for TMC interrupt which is 512 IRCL clock cycles，the program is like：

```
-----------------------------------------------------------------------------------------
#define TME(N)          (N<<7)    //N=0-1
#define TMF             (1<<0)

#define IHCKE           (1<<7)
#define ILCKE           (1<<6)

void INT3_ISR (void) interrupt 5
{
        if(TMCON & TMF)              // judge the TMC interrupt flag
        {
                TMCON |= TMF;        // clear TMC interrupt flag

        }
}


void TMC_init(void)
{
        CKCON |= ILCKE;     // enable IRCL clock
        TMCON = TME(1);     // enable TMC
        TMSNU = 0;          // set one time unit (512IRCL clock cycles) for the interrupt
        INT3EN =1;          // enable TMC interrupt
        EA = 1;             //enable global interrupt
}
-----------------------------------------------------------------------------------------
```

15 General Purpose Input/Output(GPIO)

15.1 Function Introduction

The CA51F7 series chips have a maximum package of 18 I/O pins, each of which is an alternate function pin that can be independently programmed not only as an input/output port, but also set as other function pins. Each pin is assigned a function setting register PnxF (corresponding to pin Pnx, where n=0, 1, 3, representing P0, P1, P3, and x=0~7, representing Pn.0~Pn.7), and the user can configure the main function and other options of the pin through register PnxF. For more information please refer to the register description.

Main features of GPIO:

- High impedance mode configurable
- The pull-up or pull-down can be set independently for the I/O structure
- Open-drain or push-pull output can be selected for the output mod
- The data output latch can be read/revised/written
- Supports 2.2~5.5V voltage
- 1 PWM pin (P0.7) with a maximum push current of 40mA and a sink current of 55mA for direct drive of MOS tubes (specifically designed for humidifier applications)
- 1 PWM pin (P1.0) with a sink current of 100mA current supply for direct drive of cooling fan (specifically designed for humidifier applications)
- 4 PWM pins (P1.1, P1.2, P1.3, P1.4) with 55mA sink current for direct drive of LED indicators (specifically designed for humidifier applications)

Note: Please refer to the Electrical Characteristics chapter for the above push/sink current test conditions.

The Figure 15-1-1 shows GPIO Push-pull Mode Structure

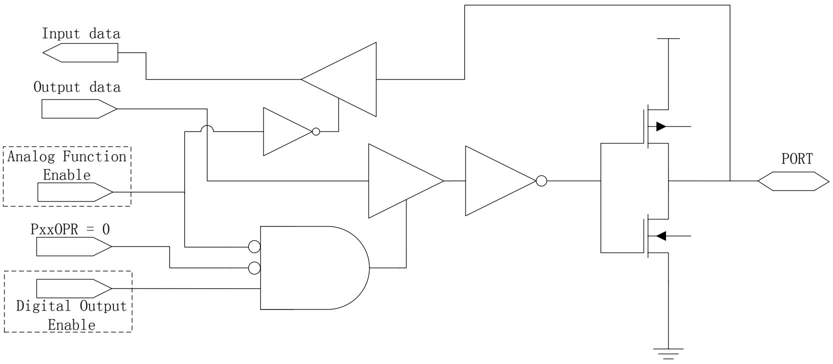


Figure 15-1-1 I/O Push-pull Mode Structure

The Figure 15-1-2 shows GPIO Open-drain Mode Structure

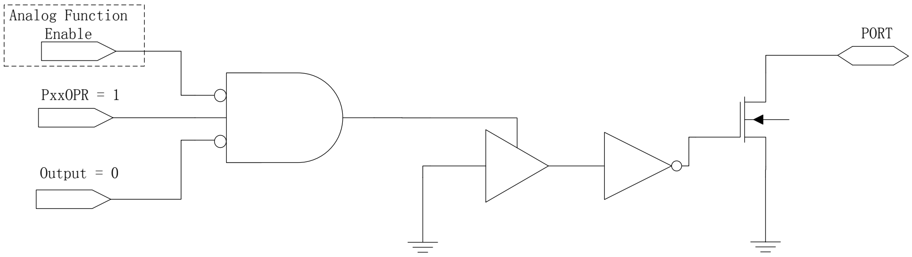


Figure 15-1-2 I/O Open-drain Mode Structure

The Figure 15-1-3 shows GPIO Pull-down Mode Structure

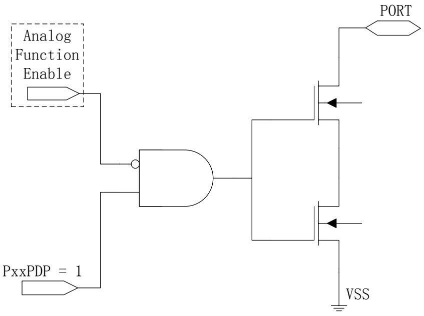


Figure 15-1-3 I/O I/O Pull-down Mode Structure

The Figure 15-1-4 shows GPIO Pull-up Mode Structure

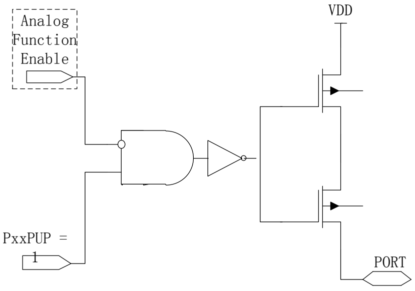


Figure 15-1-4 I/O I/O Pull-up Mode Structure

15.2 Pin Register Description

Table 15-2-1 Register P0

80H	7	6	5	4	3	2	1	0
P0	P07	P06	P05	P04	P03	P02	P01	P00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7~0	P0x	Data register for pin P0x, valid when the pin function is set to GPIO 0: When the pin is set to input, P0x is low level; when the pin set to output,P0x outputs low level signal 1: When the pin is set to input, P0x is high level; when the pin set to output,P0x outputs high level signal

Table 15-2-2 Register P1

90H	7	6	5	4	3	2	1	0
P1	P17	P16	P15	P14	P13	P12	P11	P10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7~0	P1x	Data register for pin P1x, valid when the pin function is set to GPIO 0: When the pin is set to input, P1x is low level; when the pin set to output,P1x outputs low level signal 1: When the pin is set to input, P1x is high level; when the pin set to output,P1x outputs high level signal

Table 15-2-3 Register P3

B0H	7	6	5	4	3	2	1	0
P3	-	-	-	-	-	-	P31	P30
R/W	-	-	-	-	-	-	R/W	R/W
Initial Value	-	-	-	-	-	-	0	0

Bit number	Bit Symbol	Description
7~6	-	-
5~0	P3x	Data register for pin P3x, valid when the pin function is set to GPIO 0: When the pin is set to input, P3x is low level; when the pin set to output,P3x outputs low level signal 1: When the pin is set to input, P3x is high level; when the pin set to output,P3x outputs high level signal

Table 15-2-4 Pin Function Control Register

8000H	7	6	5	4	3	2	1	0
P00F	P00PUP	P00PDP	P00OPR	-	-	P00S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial Value	0	0	0	-	-	0	0	0
8001H	7	6	5	4	3	2	1	0
P01F	P01PUP	P01PDP	P01OPR	-	-	P01S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial Value	0	0	0	-	-	0	0	0
8002H	7	6	5	4	3	2	1	0
P02F	P02PUP	P02PDP	P02OPR	-	-	P02S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial Value	0	0	0	-	-	0	0	0
8003H	7	6	5	4	3	2	1	0
P03F	P03PUP	P03PDP	P03OPR	-	-	P03S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial Value	0	0	0	-	-	0	0	0
8004H	7	6	5	4	3	2	1	0
P04F	P04PUP	P04PDP	P04OPR	-	-	P04S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial Value	0	0	0	-	-	0	0	0
8005H	7	6	5	4	3	2	1	0
P05F	P05PUP	P05PDP	P05OPR	-	-	P05S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial Value	0	0	0	-	-	0	1	1
8006H	7	6	5	4	3	2	1	0
P06F	P06PUP	P06PDP	P06OPR	-	-	P06S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial Value	0	0	0	-	-	0	0	0
8007H	7	6	5	4	3	2	1	0
P07F	P07PUP	P07PDP	P07OPR	-	-	P07S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial Value	0	0	0	-	-	0	0	0
8008H	7	6	5	4	3	2	1	0
P10F	P10PUP	P10PDP	P10OPR	-	-	P10S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial Value	0	0	0	-	-	0	0	0
8009H	7	6	5	4	3	2	1	0
P11F	P11PUP	P11PDP	P11OPR	-	-	P11S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial Value	0	0	0	-	-	0	0	0
800AH	7	6	5	4	3	2	1	0
P12F	P12PUP	P12PDP	P12OPR	-	-	P12S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial Value	0	0	0	-	-	0	0	0
800BH	7	6	5	4	3	2	1	0
P13F	P13PUP	P13PDP	P13OPR	-	-	P13S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial Value	0	0	0	-	-	0	0	0

800CH	7	6	5	4	3	2	1	0
P14F	P14PUP	P14PDP	P14OPR	-	-	P14S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial value	0	0	0	-	-	0	0	0

800DH	7	6	5	4	3	2	1	0
P15F	P15PUP	P15PDP	P15OPR	-	-	P15S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial value	0	0	0	-	-	0	0	0

800EH	7	6	5	4	3	2	1	0
P16F	P16PUP	P16PDP	P16OPR	-	-	P16S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial value	0	0	0	-	-	0	0	0

800FH	7	6	5	4	3	2	1	0
P17F	P17PUP	P17PDP	P17OPR	-	-	P17S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial value	0	0	0	-	-	0	0	0

8018H	7	6	5	4	3	2	1	0
P30F	P30PUP	P30PDP	P30OPR	-	-	P30S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial value	0	0	0	-	-	1	1	0

8019H	7	6	5	4	3	2	1	0
P31F	P31PUP	P31PDP	P31OPR	-	-	P31S		
R/W	R/W	R/W	R/W	-	-	R/W		
Initial value	0	0	0	-	-	1	1	0

Bit number	Bit symbol	Description
7	PnxPUP	Pull-up resistor enable control 0: disable pull-up resistor 1: enable pull-up resistor *Note: The pull-up resistor is 30K*
6	PnxPDP	Pull-down resistor enable control 0: disable pull-down resistor 1: enable pull-down resistor *Note: The pull-down resistor is 30K*
5	PnxOPR	Open-drain enable control, only valid when the pin is set to be digital output 0: disable open-drain 1: enable open-drain

Table 15-2-5 Register PnxC

8120H	7	6	5	4	3	2	1	0
P00C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial value	-	1	-	-	-	-	0	0

8121H	7	6	5	4	3	2	1	0
P01C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial value	-	1	-	-	-	-	0	0

8122H	7	6	5	4	3	2	1	0
P02C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial value	-	1	-	-	-	-	0	0

8123H	7	6	5	4	3	2	1	0
P03C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial value	-	1	-	-	-	-	0	0

8124H	7	6	5	4	3	2	1	0
P04C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial value	-	1	-	-	-	-	0	0

8125H	7	6	5	4	3	2	1	0
P05C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial value	-	1	-	-	-	-	0	0

8126H	7	6	5	4	3	2	1	0
P06C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial value	-	1	-	-	-	-	0	0

8127H	7	6	5	4	3	2	1	0
P07C	-	SMIT_EN	GDRV[2:0]			DRV_EN	DRV	SR
R/W	-	R/W	R/W			R/W	R/W	R/W
Initial value	-	1	0	0	0	0	0	0

8128H	7	6	5	4	3	2	1	0
P10C	-	SMIT_EN	SINK[2:0]			SINK_EN	DRV	SR
R/W	-	R/W	R/W			R/W	R/W	R/W
Initial value	-	1	0	0	0	0	0	0

8129H	7	6	5	4	3	2	1	0
P11C	-	SMIT_EN	-	SINK[1:0]		SINK_EN	DRV	SR
R/W	-	R/W	-	R/W		R/W	R/W	R/W
Initial value	-	1	-	0	0	0	0	0

812AH	7	6	5	4	3	2	1	0
P12C	-	SMIT_EN	-	SINK[1:0]		SINK_EN	DRV	SR
R/W	-	R/W	-	R/W		R/W	R/W	R/W
Initial value	-	1	-	0	0	0	0	0

812BH	7	6	5	4	3	2	1	0
P13C	-	SMIT_EN	-	SINK[1:0]		SINK_EN	DRV	SR
R/W	-	R/W	-	R/W		R/W	R/W	R/W
Initial value	-	1	-	0	0	0	0	0

812CH	7	6	5	4	3	2	1	0
P14C	-	SMIT_EN	-	SINK[1:0]		SINK_EN	DRV	SR
R/W	-	R/W	-	R/W		R/W	R/W	R/W
Initial value	-	1	-	0	0	0	0	0

812DH	7	6	5	4	3	2	1	0
P15C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial value	-	1	-	-	-	-	0	0

812EH	7	6	5	4	3	2	1	0
P16C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial value	-	1	-	-	-	-	0	0

812FH	7	6	5	4	3	2	1	0
P17C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial value	-	1	-	-	-	-	0	0

8138H	7	6	5	4	3	2	1	0
P30C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial value	-	1	-	-	-	-	0	0

8139H	7	6	5	4	3	2	1	0
P31C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial value	-	1	-	-	-	-	0	0

Bit number	Bit Symbol	Description
7	-	-
6	SMIT_EN	when SMIT_EN=1, input SMIT enable; when SMIT_EN=0, input inverter enable.
5~3	GDRV/ SINK [2:0]	GDRV：Drive current intensity selection SINK：Sink current intensity selection *Note: For details on the maximum drive current, please see the Electrical Specification section*
2	DRV_EN/SINK_EN	DRV_EN: Drive current enable control SINK_EN: Sink current enable control
1	DRV	Output intensity selection 0：Weak drive 1：Strong drive
0	SR	Output slope control 0：Slowest slope control 1：Fastest slope control

Note：Pnx → n=0、1、3, stands for P0、P1、P3
x=0~7, stands for Pn.0~Pn.7

Table 15-2-6 Pin Alternate Function Mapping

Value Name	0	1	2	3	4	5	6	7
P00S	high impedance	digital input	digital output	ADC[0]	TK[0]	T0	high impedance	high impedance
P01S	high impedance	digital input	digital output	ADC[1]	TK[1]	T1	high impedance	high impedance
P02S	high impedance	digital input	digital output	ADC[2]	TK[2]	high impedance	high impedance	high impedance
P03S	high impedance	digital input	digital output	ADC[3]	TK[3]	high impedance	high impedance	high impedance
P04S	high impedance	digital input	digital output	ADC[4]	TK[4]	high impedance	high impedance	high impedance
P05S	high impedance	digital input	digital output	RESET	TK[5]	high impedance	high impedance	high impedance
P06S	high impedance	digital input	digital output	high impedance	TK_CAP	high impedance	high impedance	high impedance
P07S	high impedance	digital input	digital output	PWM0	high impedance	high impedance	high impedance	high impedance
P10S	high impedance	digital input	digital output	PWM1	high impedance	high impedance	high impedance	high impedance
P11S	high impedance	digital input	digital output	PWM2	high impedance	high impedance	high impedance	high impedance

P12S	high impedance	digital input	digital output	PWM3	high impedance	high impedance	high impedance	high impedance
P13S	high impedance	digital input	digital output	PWM4	high impedance	high impedance	high impedance	high impedance
P14S	high impedance	digital input	digital output	PWM5	high impedance	high impedance	high impedance	high impedance
P15S	high impedance	digital input	digital output	ADC[7]	TK[7]	high impedance	high impedance	high impedance
P16S	high impedance	digital input	digital output	ADC[8]	TK[8]	high impedance	I2C_SDA	high impedance
P17S	high impedance	digital input	digital output	ADC[9]	TK[9]	high impedance	I2C_SCL	high impedance
P30S	high impedance	digital input	digital output	ADC[6]	high impedance	UART_TX	I2C_SDA	high impedance
P31S	high impedance	digital input	digital output	ADC[5]	TK[6]	UART_RX	I2C_SCL	high impedance

15.3 Pin control Example

◆ **Set the Pin function**

For instance, P00 is set to be push-pull output, the program is like：

```
P00F = 2;
```

P00 is set to be open-drain output, the program is like：

```
P00F = (1<<5)|2;
```

P00 is set to be open-drain output with pull-up enabled, the program is like：

```
P00F = (1<<7)｜(1<<5)｜2;
```

P00 is set to be input with pull-up enabled, the program is like：

```
P00F = (1<<7)｜1;
```

16 Universal Asynchronous Receiver/Transmitter(UART)

16.1 Function Introduction

UART is a full duplex asynchronous serial data transceiver, UART receiver includes a one byte buffer. There are two working modes for UART which is shown as the table 16-1-1-1.

SM	Mode	Description	Baud rate
0	A	9 bit asynchronous mode	CPUCLK/(32*(1024-SREL))
1	B	8 bit asynchronous mode	CPUCLK/(32*(1024-SREL))

Table 16-1-1-1 UART Communication Mode

The UART is designed with a dedicated baud rate generator, and the baud rate is configured through registers SRELL, SRELH.

Figure 16-1-1-1 shows the schematic of UART.

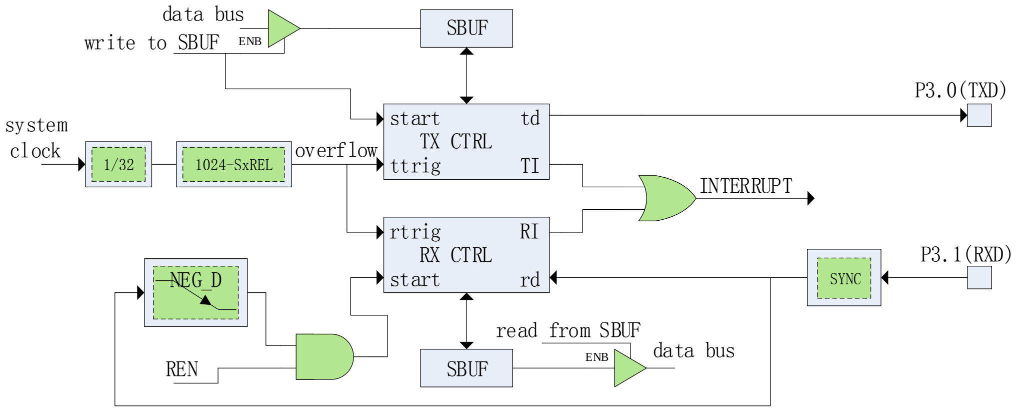


图 16-1-1-1 UART Schematic

- **Mode A**

In mode A, the UART can transmit and receive 9 bits of data asynchronously and simultaneously. Writing data to register SBUF will start UART data transmitting. The first bit transmitted is the start bit (which is 0), followed by 9 bits of data (low bit first), The 9th data bit is the TB81 bit of register SCON, The last bit transmitted is the stop bit (which is 1)。 In the receive state, the UART is synchronized by detecting the falling edge of pin RX. After the transmit process is completed, the low 8 bits of data are stored in register SBUF and the 9th bit of data is stored in RB8 bit.

- **Mode B**

Mode B differs from Mode A in that Mode B is an 8-bit data transmit, and the stop bit holds a valid stop bit. Its other functions are consistent with Mode A.

- **UART Multi-computer Communication**

Multi-computer Communication can be realized by UART in mode A. If SM21 of register SCON is set to 1, only when the 9th data is 1 (RB8=1), the slave will generate receive interrupt, which makes multi-computer communication possible. The slaves set their SM21 bits to 1, and the host transmits the slave's address with bit 9 data also set to 1, so that all slaves will generate a receive interrupt; The slaves' software then compare the address received to their own addresses. It the address matches, the matched slave will set SM21=0. The host then set the 9th data bit to 0 for the following data transmission. Due to the other slaves remain SM21 = 1, thus only the address matched slave will generate receive interrupt.

16.2 Register Description

Table 16-2-1 Register SCON

98H	7	6	5	4	3	2	1	0
SCON	SM	UIE	SM2	REN	TB8	RB8	TI	RI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7	SM	UART mode selection, for more information please refer to Table 16-1-1-1
6	UIE	UART interrupt enable control, 1 enables
5	SM2	Multi-computer communication enable control, 1 enables
4	REN	Serial receive enable control, 1 enables
3	TB8	The 9th data bit to transmit This bit is used for UART data transfer. It will be transmitted as the 9th bit of the data in mode A and it is controlled by the software (For instance, parity check or multi-computer communication)
2	RB8	The 9th bit of the data received This bit is used for UART data receive. It will be received as the 9th bit of the data in mode A . It is the stop bit in mode B.
1	TI	Transmit interrupt flag, 1 indicates the interrupt, cleared by writing 1 to it.
0	RI	Receive interrupt flag, 1 indicates the interrupt, cleared by writing 1 to it.

Table 16-2-2 Register SBUF

99H	7	6	5	4	3	2	1	0
SBUF	SBUF[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7~0	SBUF	Receiver/Transmitter buffer Writing data to SBUF will starts the data transmission Reading SBUF will reads the data received

Table 16-2-3 Register UDCKS

D8H	7	6	5	4	3	2	1	0
UDCKS	UDE	-	-	DNUM[4:0]				
R/W	R/W	-	-	R/W				
Initial Value	0	-	-	0	0	0	0	0

Bit number	Bit Symbol	Description
7	UDE	Rapid baud rate setting enable control, 1 enables it. ***Note：*** *When UDE=0, the configuration of the baud rate for UART remains the same; while UDE=1，the baud rate for UART is configured by DNUM.*
6~5	-	-
4~0	DNUM	Rapid baud rate setting register, valid only when UDE=1. For transmission, it is a must that DNUM>=0； for receiving, it is a must that DNUM>=6. BR = F_{sys}/((DNUM+1)*(1024-SREL))

Table 16-2-4 Register SRELL、SRELH

8068H	7	6	5	4	3	2	1	0
SRELL	SREL[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
8069H	7	6	5	4	3	2	1	0
SRELH	-	-	-	-	-	-	SREL[9:8]	
R/W	-	-	-	-	-	-	R/W	
Initial Value	-	-	-	-	-	-	0	0

Bit number	Bit Symbol	Description
9~0	SREL	Baud rate configuration register When UDE=0, BR = Fsys/(32 * (1024 – SREL)) When UDE=1, BR = F_{sys}/((DNUM+1)*(1024-SREL))

17 I²C Interface

17.1 Function Introduction

I²C modules enables the chip to communicate with peripheral I²C devices by serial transmission standard which complies with standard I²C specification. It can be set to either slave or master and configured to standard/fast/high speed mode.

17.2 I²C Main Features

- Simple but strong communication port, bi-directional bus with 2 wires
- Slave/Master mode configurable
- Able to operate in receiver/transmitter mode
- 7-bit slave address
- Supports multimeters' arbitration
- Broadcast function supported

17.3 I²C Function Description

I²C modules supports I²C standard bus specification. I²C bus includes 2 wires to transfer data among devices, one is SCL(Serial Clock) and the other is SDA(Serial Data), as Figure 17-3-1 shows. Since the it is open-drain port for I²C, there must be pull-up resistor on I²C bus. The pull-up resistor can be connected externally or enabled internally. Each device that connects to the bus has its own 7-bit address.

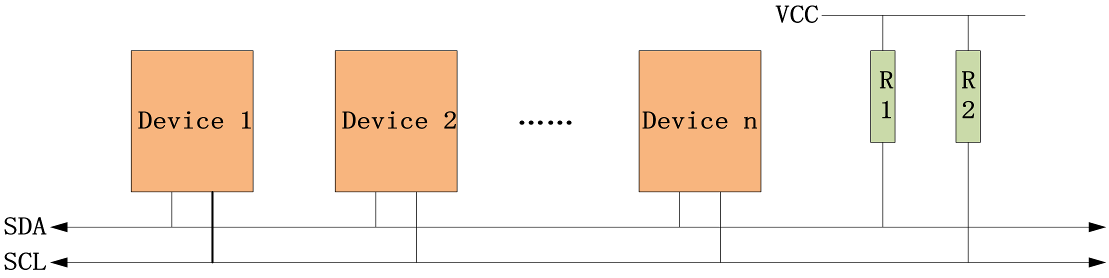


Figure 17-3-1 I²C Bus Interconnection

I²C module principle is as Figure 17-3-2 shows.

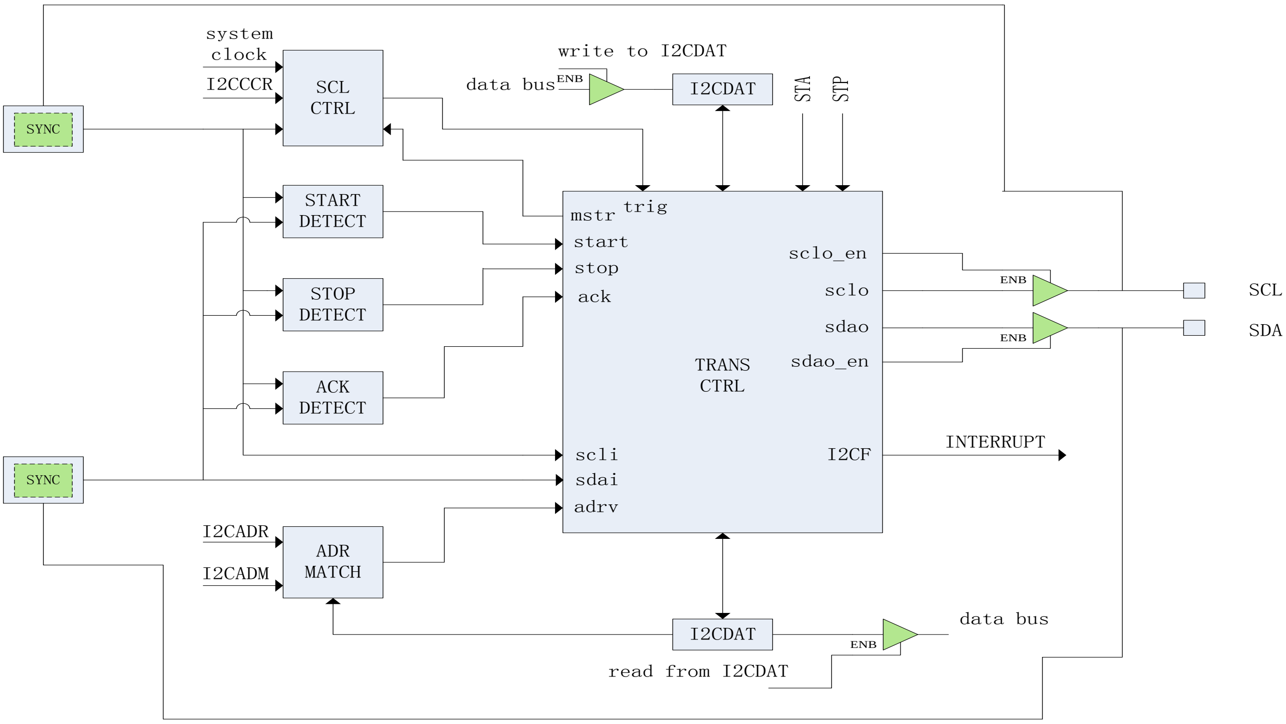


Figure 17-3-2 I²C Module Schematic

● I²C Mode Selection

I²C can operates in the following 4 modes： slave transmit mode, slave receive mode, master transmit mode, master receive mode. I²C operates in slave mode by default. I²C changes to master mode after the START signal generated and returns slave mode when the arbitration fails or STOP signal is generated.

● I²C Bus Data Transmission Pattern

There are usually 4 stages for the standard I²C communication: START signal, slave address transfer, data transmission and STOP signal. The data transmitted on I²C bus is always 8 bits with the most significant bit sent first. There must be a ACK following every one byte data. However, there is no byte limits for the data transmission. The master sends STOP signal after the transmission is over and terminates the communication.

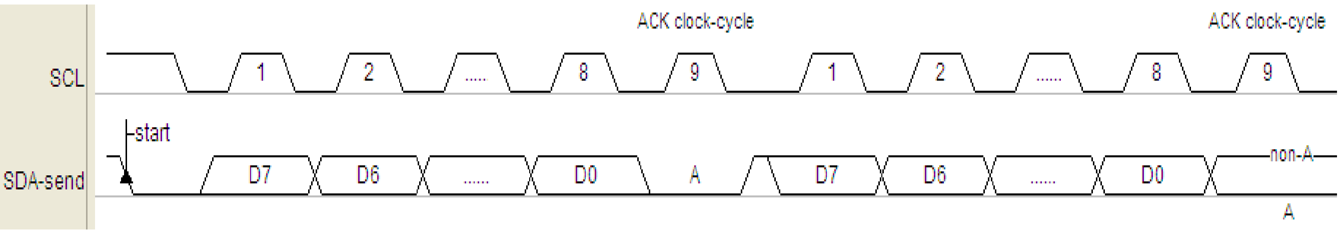


Figure 17-3-3 I²C Bus Data Transmission Format

● Communication Process

I²C enables the data transmission and generates the clock signal in Master mode. The serial data transmission always begins with START signal and ends with STOP signal. Both START and STOP signals are generated by the software in master mode. Setting STA=1 generates START signal and setting STP=1 generates STOP signal.

I²C can distinguish its address (7 bits) and the broadcast address in slave mode. Software can enable/disable its ability to recognize broadcast address by setting GCE.

Both address and data are transmitted in bytes. The address will be sent by the master after the START signal. Both address and data are transmitted in bytes. The address will be sent by the master after the START signal. Both address and data are transmitted in bytes. The address will be sent by the master after the START signal. When the one byte information is received, the ACK signal will be generated automatically.

Every time when one byte data is received/transmitted or arbitration fails (and etc.) there will be an interrupt flag I2CF. The status of the event will be indicated by register I2CSTA (for more information please refer to register I2CSTA). The software decides the next operation according to the status of the event when interrupt occurs. Clearing the interrupt flag I2CF will start the next operation.

The STOP signal generated by the master at the end of communication also generates the interrupt flag I2CSTP on the slave, indicating the completion of the communication process. When there occurs interrupt I2CF, if SHD=1, SCL will be set to low by slave. After the master detects that SCL is released, it master will then continue the next operation; On the other hand, if SHD=0, SCL will not be set to low by the slave, which makes it compatible with applications when the master I^{2}C is simulated by software. Thus, the master's software must wait long enough so that the slave can deal with the response.

When I^{2}C is set as slave, the master outputs SCL clock, and it has nothing to do with the slave's clock configuration. As for the slave, SCL must remain low for at least 6.5 system clock cycles periods and high for at least 2.5 system clock cycles periods. In the end, the frequency of SCL sent by external master can be at most 1/9.

17.4 I^{2}C Communication Pin Mapping

There are different mappings for I2C communication pins which could be selected by register I2CIOS. For more information, please refer to register I2CIOS description.

17.5 Register Description

Table 17-5-1 Register I2CCON

C0H	7	6	5	4	3	2	1	0
I2CCON	I2CE	I2CIE	STA	STP	SHD	AAK	CBSE	STFE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	1	0	0	0	0	1	0	0

Bit number	Bit Symbol	Description
7	I2CE	I^{2}C module enable control, 1 enables it
6	I2CIE	I^{2}C module enable control, 1 enables it
5	STA	I^{2}C START signal transfer control, valid when it is 1, it will be cleared automatically when START signal
4	STP	I2C STOP signal transfer control, valid when it is 1, it will be cleared automatically when STOP signal
3	SHD	When it is 1, if I2CF=1, I2CF will make SCL remain low after SCL becomes low
2	AAK	I^{2}C ACK signal transfer control, 1 enables it *Note：*

		When I^{2}C is configured as slave, this bit must be set to 1 beforehand, otherwise even the address matches it will not reply ACK, and thus cannot be addressed
1	CBSE	CBUS compatible enable control When it is set to 1, the ACK will be ignored during the transmission to be compatible with CBUS bus
0	STFE	When STFE=1, I2CF will be set to 1 if I^{2}C module detects the START signal

Table 17-5-2 Register I2CADR

C1H	7	6	5	4	3	2	1	0
I2CADR	GCE	I2CADRL[6:0]						
R/W	R/W	R/W						
Initial Value	1	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7	GCE	Broadcast address recognition(00H)enable control, 1 enables it
6~0	I2CADRL	I^{2}C slave address, only valid when it operates as slave *Note：* *(When AAK=1) when the address is 7 bits and the higher 7 bits of first received address matches I2CADR, reply with ACK and enters slave mode*

Table 17-5-2 Register I2CADR

C2H	7	6	5	4	3	2	1	0
I2CADM	SPFE	I2CADML[6:0]						
R/W	R/W	R/W						
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7	SPFE	Broadcast address recognition(00H)enable control, 1 enables it
6~0	I2CADML	I^{2}C slave address, only valid when it operates as slave *Note：* *(When AAK=1) when the address is 7 bits and the higher 7 bits of first received address matches I2CADR, reply with ACK and enters slave mode*

Table 17-5-4 Register I2CCR

C3H	7	6	5	4	3	2	1	0
I2CCR	I2CCR[7:0]							
R/W	R/W							
Initial Value	0	0	1	0	0	0	0	0

Bit number	Bit Symbol	Description
7~0	I2CCR	I^{2}C clock setting register The sampling frequency is $2^{I2CCR[7:5]}$ divisions of the I^{2}C operating clock, when I2CCR[7:5] is equal to 000：$F_{sample}=F_{i2cclk}$ 001：$F_{sample}=F_{i2cclk}/2$ 010：$F_{sample}=F_{i2cclk}/4$ … 111：$F_{sample}=F_{i2cclk}/128$ The output frequency is the (I2CCCR[4:0]+1) division of the sampling frequency, $F_{scl}=F_{i2cclk}/(2^{I2CCCR[7:5]}*(I2CCCR[4:0]+1))$ For example I2CCCR[4:0]=9, when I2CCR[7:5] is equal to 000：$F_{scl}=F_{i2cclk}/(1*10)$ 001：$F_{scl}=F_{i2cclk}/(2*10)$

		010：$F_{scl}=F_{i2cclk}/(4*10)$ … 111：$F_{scl}=F_{i2cclk}/(128*10)$ ***Note：*** *1. When I2CCCR[7:5] = 0, if a value less than 9 is written to I2CCR[4:0], I2CCR[4:0] is automatically counted as a value of 9.* *2. When I2CCCR[7:5] > 0, if a value less than 7 is written to I2CCR[4:0], I2CCR[4:0] is automatically counted as a value of 7.*

Table 17-5-5 Register I2CDAT

C4H	7	6	5	4	3	2	1	0
I2CDAT	I2CDAT[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
Bit number	Bit Symbol	Description						
7~0	I2CDAT	Buffer for receiving/transmission ***Note：*** *When I2CF is 1, it is recommended to make I2CF remain 1 when users overwrite/read I2CDAT. I2CF should be cleared after the process is over, and then the transmission continues so that there will be no transmission errors.*						

Table 17-5-6 Register I2CSTA

C5H	7	6	5	4	3	2	1	0
I2CSTA	I2CSTA[7:0]							
R/W	R							
Initial Value	1	1	1	1	1	0	0	0
Bit number	Bit Symbol	Description						
7~0	I2CSTA	I^{2}C status register 00H: (master/slave) bus error 08H: (master/slave)START signal detected (valid only when STFE=1) 18H: (master)address and write bit sent, ACK signal received 20H: (master)address and write bit sent, no ACK signal received 28H: (master)one byte data received/transmitted, ACK signal detected 30H: (master)one byte data received/transmitted, no ACK signal detected 38H: (master)arbitration lost(master will change to slave after arbitration lost) 40H: (master)address and read bit transmitted, ACK signal received 48H: (master)address and read bit transmitted, no ACK signal received 60H: (slave)address and write bit received, with ACK signal is sent 70H: (master/slave)broadcast address received with ACK signal is sent(master/slave will become slave) 80H: (slave)one byte data received/transmitted, ACK signal detected 88H: (slave)one byte data received/transmitted, no ACK signal detected A0H: (master/slave)STOP signal detected(valid only when SPFE=1) A8H: (slave)address and read bit received, with ACK signal is sent F8H: (master/slave) bus is idle						

Table 17-5-7 Register I2CFLG

C6H	7	6	5	4	3	2	1	0
I2CFLG	-	-	-	-	-	-	-	I2CF
R/W	-	-	-	-	-	-	-	R
Initial Value	-	-	-	-	-	-	-	0
Bit number	Bit Symbol		Description					

7~1	-	-
0	I2CF	I²Cinterrupt flag, 1 indicates the interrupt, cleared by writing 1 to it *Note：* *1. I2CF will be set to 1 every time after a one-byte data or the address transmission completes (with ACK/NAK received/sent).* *2. I2CF will be set to 1 when there is bus error.* *3. If STFE=0, I2CF will not be set to 1 when START signal detected.* *4. If SPFE=0, I2CF will not be set to 1 when*

Table 17-5-8 Register I2CIOS

D7H	7	6	5	4	3	2	1	0
I2CIOS	-	-	-	-	-	-	-	I2CIOS
R/W	-	-	-	-	-	-	-	R/W
Initial Value	-	-	-	-	-	-	-	0

Bit number	Bit Symbol	Description
7~1	-	-
0	I2CIOS	2C pin selection control 0：P30/P31 1：P16/P17

18 PWM

18.1 PWM Function Introduction

CA51F7 series chip can include at most 6 channels PWM outputs. PWM period and duty cycle can be configured with 16-bit range.

There is a 16-bit counter for each PWM channel and the cycle is set by register PWMnDIV, Register PWMnDUT sets the corresponding PWM's duty cycle. PWM is enabled by register PWMEN with each bit of it corresponds to one channel in PWM. Whether the PWM pin outputs reversed signal is set by PWMnTOG. The clock source for each PWM channel can be selected by corresponding PWMnCKS of register PWMnCON, with the frequency division set by PWMnCKD independently.

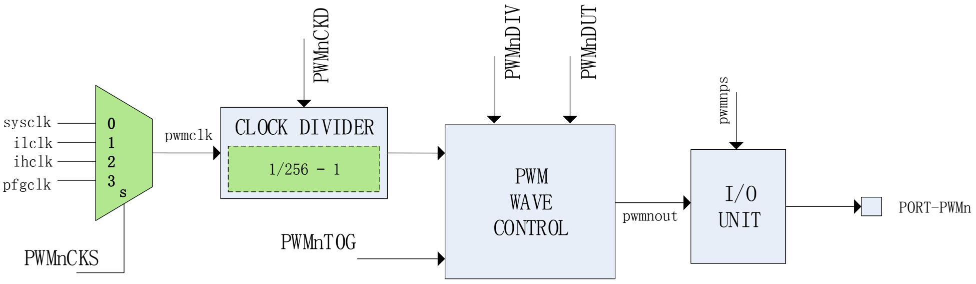


Figure 18-1-1 PWM Schematic

Note:

The 'n' in PWMnDIV, PWMnDUT and etc indicates 0/1/2/3/4/5, which stands for the control/configuration register for PWM channel 0/1/2/3/4/5.

- **PWM output waveform**

When PWM is enabled, PWM starts counting. When the count is less than or equal to PWMnDUT, PWM pin outputs high level signal (PWMnTOG=0); when the count value is greater than PWMnDUT, PWM pin outputs low level signal (PWMnTOG=0). When the count equals to PWMnDIV, a PWM cycle completes and the counter will starts counting again. PWM.

When PWMnDIV>PWMnDUT>0, the PWM waveform is as follows.

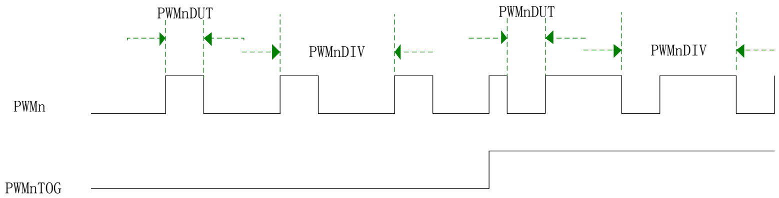


Figure 18-1-2 PWM Output Waveform

When PWMnDIV=0, PWM pin will output the PWM clock directly. If PWMnCKD=0 then, PWM pin outputs the clock source selected. On the other hand, if PWMnCKD ！=0, PWM pin will output the clock source with frequency divided by (PWMnCKD+1).

When PWMnDIV ≠ 0 and PWMnDUT=0, PWM pin outputs low/high level (PWMnTOG ＝ 0/1); when PWMnDUT>=PWMnDIV>0, PWM pin outputs high/low.

PWM0 can adjust the frequency of PFG clock by configuring register PWM0NUM, the step of each PFG frequency configuration is determined by the register STEP, the maximum number of steps adjusted up / down is determined by the register STEPNUM, For more details, please refer to the Integrated PFG oscillator section.。

18.2 PWM Register Description

Table 18-2-1 Register PWMEN

C7H	7	6	5	4	3	2	1	0
PWMEN	-	-	PWM5EN	PWM4EN	PWM3EN	PWM2EN	PWM1EN	PWM0EN
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial	-	-	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7~6	-	-
5	PWM5EN	PWM5 enable control，1 enables it
4	PWM4EN	PWM4 enable control，1 enables it
3	PWM3EN	PWM3 enable control，1 enables it
2	PWM2EN	PWM2 enable control，1 enables it
1	PWM1EN	PWM1 enable control，1 enables it
0	PWM0EN	PWM0 enable control，1 enables it

Table 18-2-2 Register PWMCON

B9H	7	6	5	4	3	2	1	0
PWM0CON	-	PWM0TOG	-	-	-	-	-	-
R/W	-	R/W	-	-	-	-	-	-
Initial value	-	0	-	-	-	-	-	-
BAH	7	6	5	4	3	2	1	0
PWM1CON	-	PWM1TOG	PWM1MOD[2:0]			PWM1POL	PWM1CKS[1:0]	
R/W	-	R/W	R/W			R/W	R/W	
Initial value	-	0	0	0	0	0	0	0
BBH	7	6	5	4	3	2	1	0
PWM2CON	-	PWM2TOG	PWM2MOD[2:0]			PWM2POL	PWM2CKS[1:0]	
R/W	-	R/W	R/W			R/W	R/W	
Initial value	-	0	0	0	0	0	0	0
BCH	7	6	5	4	3	2	1	0

PWM3CON	-	PWM3TOG	-	-	-	-	PWM3CKS[1:0]	
R/W	-	R/W	-	-	-	-	R/W	
Initial value	-	0	-	-	-	-	0	0

BDH	7	6	5	4	3	2	1	0
PWM4CON	-	PWM4TOG	-	-	-	-	PWM4CKS[1:0]	
R/W	-	R/W	-	-	-	-	R/W	
Initial value	-	0	-	-	-	-	0	0

BEH	7	6	5	4	3	2	1	0
PWM5CON	-	PWM5TOG	-	-	-	-	PWM5CKS[1:0]	
R/W	-	R/W	-	-	-	-	R/W	
Initial value	-	0	-	-	-	-	0	0

Bit number	Bit Symbol	Description
7	-	-
6	PWMnTOG	PWMn output reverse control, 1 reverses it
5~2	-	-
1~0	PWMnCKS	PWM working clock selection 00： System clock 01： IRCL 10： IRCH 11： PFG ***Note:*** *This setting is only valid for PWM1~5, PWM0 clock is fixed to PFG*

Table 18-2-3 Register PWMCKD

B1H	7	6	5	4	3	2	1	0
PWM0CKD	PWM0CKD[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

B2H	7	6	5	4	3	2	1	0
PWM1CKD	PWM1CKD[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

B3H	7	6	5	4	3	2	1	0
PWM2CKD	PWM2CKD[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

B4H	7	6	5	4	3	2	1	0
PWM3CKD	PWM3CKD[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

B5H	7	6	5	4	3	2	1	0
PWM4CKD	PWM4CKD[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

B6H	7	6	5	4	3	2	1	0
PWM5CKD	PWM5CKD[7:0]							

R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7~0	PWMnCKD	PWM Operating Clock Prescaler Configuration Register 00H：No division 01H：frequency divided by 2 02H：frequency divided by 3 …… FEH：frequency divided by 255 FFH：frequency divided by 256

Table 18-2-4 Register PWMDIVL、PWMDIVH

A9H	7	6	5	4	3	2	1	0
PWM0DIVL	PWM0DIV[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

AAH	7	6	5	4	3	2	1	0
PWM0DIVH	PWM0DIV[15:8]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

ABH	7	6	5	4	3	2	1	0
PWM1DIVL	PWM1DIV[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

ACH	7	6	5	4	3	2	1	0
PWM1DIVH	PWM1DIV[15:8]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

ADH	7	6	5	4	3	2	1	0
PWM2DIVL	PWM2DIV[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

AEH	7	6	5	4	3	2	1	0
PWM2DIVH	PWM2DIV[15:8]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

AFH	7	6	5	4	3	2	1	0
PWM3DIVL	PWM3DIV[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

A4H	7	6	5	4	3	2	1	0
PWM3DIVH	PWM3DIV[15:8]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

A5H	7	6	5	4	3	2	1	0
PWM4DIVL	PWM4DIV[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

A6H	7	6	5	4	3	2	1	0
PWM4DIVH	PWM4DIV[15:8]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0
A7H	7	6	5	4	3	2	1	0
PWM5DIVL	PWM5DIV[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0
9AH	7	6	5	4	3	2	1	0
PWM5DIVH	PWM5DIV[15:8]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0
Bit number	Bit Symbol		Description					
15~0	PWMnDIV		PWMn Cycle Configuration Register					

Table 18-2-5 Register PWMDUTL、PWMDUTH

9BH	7	6	5	4	3	2	1	0
PWM0DUTL	PWM0DUT[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0
9CH	7	6	5	4	3	2	1	0
PWM0DUTH	PWM0DUT[15:8]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0
9DH	7	6	5	4	3	2	1	0
PWM1DUTL	PWM1DUT[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0
9EH	7	6	5	4	3	2	1	0
PWM1DUTH	PWM1DUT[15:8]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0
9FH	7	6	5	4	3	2	1	0
PWM2DUTL	PWM2DUT[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0
91H	7	6	5	4	3	2	1	0
PWM2DUTH	PWM2DUT[15:8]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0
92H	7	6	5	4	3	2	1	0
PWM3DUTL	PWM3DUT[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0
93H	7	6	5	4	3	2	1	0
PWM3DUTH	PWM3DUT[15:8]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

94H	7	6	5	4	3	2	1	0
PWM4DUTL	PWM4DUT[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0
95H	7	6	5	4	3	2	1	0
PWM4DUTH	PWM4DUT[15:8]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0
96H	7	6	5	4	3	2	1	0
PWM5DUTL	PWM5DUT[7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0
97H	7	6	5	4	3	2	1	0
PWM5DUTH	PWM5DUT[15:8]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
15~0	PWMnDUT	PWMn Duty Cycle Configuration Register

Table 18-2-6 Register PWM0NUM

B7H	7	6	5	4	3	2	1	0
PWM0NUML	PWM0NUM [7:0]							
R/W	R/W							
Initial value	0	0	0	0	0	0	0	0
BFH	7	6	5	4	3	2	1	0
PWM0NUMH	PWM0NUM[15:8]							
R/W	R/W							
Initial value	0	0	0	1	0	0	0	0

Bit number	Bit Symbol	Description
15~0	PWM0NUM	The number of PWM cycles in which the PWM0 frequency changes (PWM0NUM + 1 PWM cycle frequency change once), The minimum value of PWM0NUM is 1. When set to 0, the PFG clock frequency is the reference frequency and no frequency adjustment will be made. *Note: This function is specially designed for PFG clock hopping function, please refer to PFG clock chapter for detailed function description.*

18.3 PWM Control Example

- **Single channel PWM output**

Taking PWM1 for example, the clock source for PWM is IRCH（frequency of IRCH is 16MHz）, the output clock's frequency is 30K with duty cycle 30%, the program is like:

```
    //PWMxCON
#define TOG(n)                (n<<6)
#define PWM_CKS_SYS           (0<<0)
#define PWM_CKS_IL            (1<<0)
#define PWM_CKS_IH            (2<<0)
#define PWM_CKS_PG            (3<<0)
#define  IHCKE                (1<<7)

void PWM_init(void)
{
    P10F = 3;                 //set P10 as PWM pin
    CKCON |= IHCKE;           //enables IRCH clock

    PWM1CON = TOG(0) | PWM_CKS_IH;   //set IRCH as the clock source for PWM1, PWM output is not inverted
    PWM1CKD = 0;                     //set the frequency division

    PWM1DIVH = 0x02;          //set the DIV， 16000000/30000=0x215
    PWM1DIVL = 0x15;

    PWM1DUTH = 0x00;          //set DUT， the duty cycle is 30
    PWM1DUTL = 0xA0;

    PWMEN |= (1<<1);                 //enables PWM1
}
```

19 Analog / Digital Converter(ADC)

19.1 Function Introduction

Analog/digital converter is a 12-bit successive approximation(SAR) ADC, with at most 10 input channels. The clock source for ADC is the system clock with frequency division configurable. There are ADC multiple reference voltages for ADC. When internal voltage is selected as the reference voltage, it can be used to test the power supply voltage for the chip and there will be correction to ensure the chip's consistency as well. There is also a compare mode for it with threshold configurable. Once it goes beyond the threshold, a corresponding interrupt occurs. The signal can be amplified/narrowed before the conversion when using ADC and OPAMP together.

19.2 Main Feature

- 12-bit resolution
- 10 input channels at most
- Supports ADC interrupt
- ADC clock frequency division configurable
- Multiple reference voltages available: internal reference voltage, VDD.
- Supports VDD and reference ground voltage measurements.
- Automatic data correction supported when internal reference voltage is selected
- Input voltage range：VSS<=VIN<=VDD.

19.3 Block Diagram

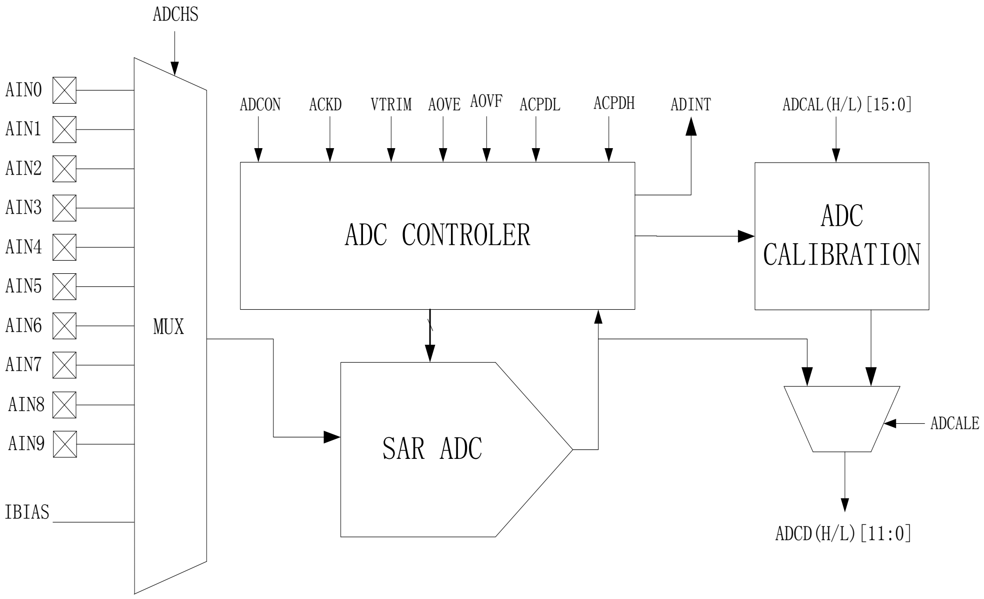


Figure 19-3-1 ADC Architecture

19.4 Function Introduction

ADC can be enabled by AST. When AST=1, the input voltage selected by ADCHS will be analog/digital converted. The clock for ADC is the system clock with frequency division set by ACKD beforehand. When ADC clock is constant, the time for single conversion is set by HTME. The conversion time is (13+2^HTME) ADC clock cycle periods. 12-bit A/D will be stored in register ADCDH and ADCDL after the conversion. AST will be cleared automatically 2.5 clock cycles later. The interrupt flag ADIF will be set to 1 at the same time. If ADC interrupt is enabled then, ADC interrupt occurs. Figure 19-4-1 is the sequence diagram for ADC conversion.

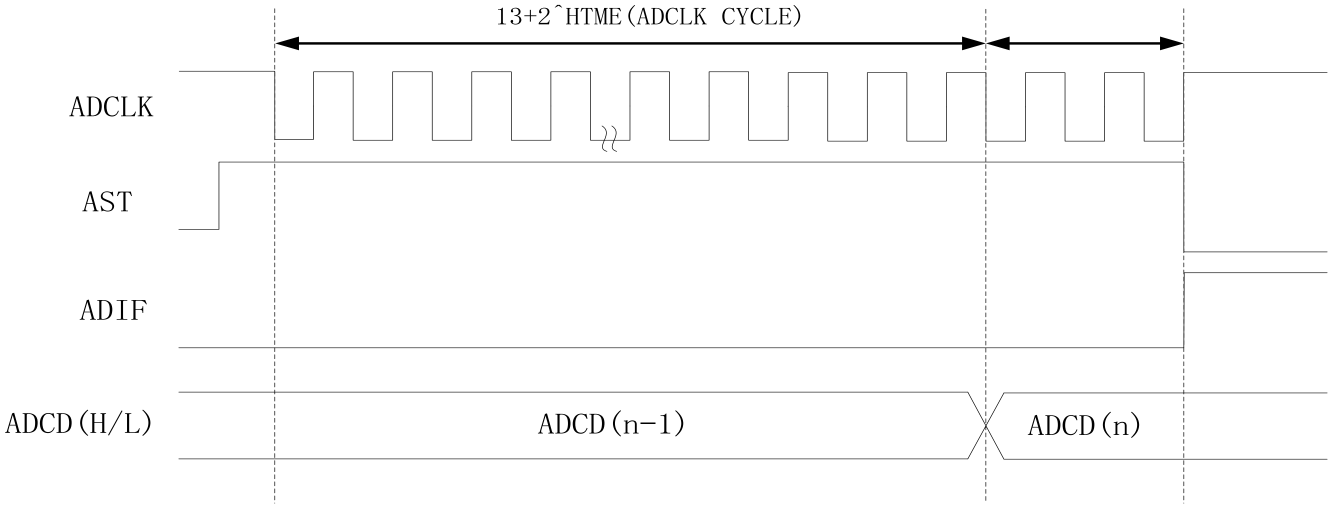


Figure 19-4-1 ADC Sequence Diagram

- **ADC Data Calibration**

When internal voltage(1.5V) is selected as the reference voltage, due to the discreteness of the chips, the internal voltage in each chip cannot be exactly the same which induces different ADC conversion results consequentially. Thus, it is necessary to correct the AD value after the conversion. The internal voltage will be tested and a correction value will be obtained when chips leave factory. When the chip's powered on, the correction value will be loaded into register ADCALL and ADCALH. The accurate AD value will be obtained by calculation according. The final accurate result for AD will be stored in register ADCD. The function can be enabled by ADCALE. Users only need to set ADCALE=1 and the correction will be done automatically.

19.5 Register Description

Table 19-5-1 Register ADCON

8060H	7	6	5	4	3	2	1	0
ADCON	AST	ADIE	ADCIF	HTME			-	VSEL
R/W	R/W	R/W	R/W	R/W			-	R/W
Initial Value	0	0	0	0	1	0	0	0
Bit number	Bit Symbol	Description						
7	AST	ADC conversion enable control, the conversion starts when 1 is written to it, the hardware will clear it automatically after the conversion.						

6	ADIE	ADC interrupt enable control, 1 enables it
5	ADCIF	ADC interrupt flag, cleared when 1 is written to it
4~2	HTME	The number of sampling periods is 2 power HTME
1	-	-
0	VSEL	ADC reference voltage selection 0: internal 1.5V(INNER_VREF)as reference 1: external VDD

Table 19-5-2 Register ADCFGL

8061H	7	6	5	4	3	2	1	0
ADCFGL	ACKD			ADCALE	ADCHS			
R/W	R/W			R/W	R/W			
Initial Value	0	0	0	1	0	0	0	0

Bit number	Bit Symbol	Description
7~5	ACKD	ADC clock frequency division setting 000：no division 001：frequency divided by 2 010：frequency divided by 4 … 111：frequency divided by 14
4	ADCALE	ADC calibration enable control, 1 enables it Valid only when the internal 1.5V is selected as the reference voltage. When ADCALE=1, ADC conversion result will be calibrated according to register ADCAL. For more information please refer to register ADCAL description
3~0	ADCHS	ADC channel enable selection 0000： disable the channels 0001： enable channel AD_CH[0](P00) 0010： enable channel AD_CH[1](P01) 0011： enable channel AD_CH[2](P02) 0100： enable channel AD_CH[3](P03) 0101： enable channel AD_CH[4](P04) 0110： enable channel AD_CH[5](P31) 0111： enable channel AD_CH[6](P30) 1000： enable channel AD_CH[7](P15) 1001： enable channel AD_CH[8](P16) 1010： enable channel AD_CH[9](P17) 1011： enable 1/4 VDD detection 1100： enable VSS detection Others： disable the channels

Table 19-5-4 Register ADCAL

8064H	7	6	5	4	3	2	1	0
ADCALL	ADCAL[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
8065H	7	6	5	4	3	2	1	0
ADCALH	ADCAL[15:8]							
R/W	R/W							
Initial Value	0	0	0	1	0	0	0	0

Bit number	Bit Symbol	Description
15~0	ADCAL	ADC calibration register, valid only when ADCALE=1 and the internal 1.5V is selected as reference voltage. When it is valid, the ADC output is： ADCDL= (ADC conversion result*ADCAL)/32768

Table 19-5-5 Register ADCD

8062H	7	6	5	4	3	2	1	0
ADCDL	ADCDL[3:0]				-			
R/W	R				-			
Initial Value	0	0	0	0	-	-	-	-
8063H	7	6	5	4	3	2	1	0
ADCDH	ADCDH[11:4]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
11~0	ADCD	ADC conversion result

20 Capacitive touch key(Touch Key)

20.1 Function Introduction

With great anti-jamming performance, CA51F7 series chip can pass EFT, CS test and etc. The Touch Key module supports 10 channels at most (Different chip models support different number of touch channels). To meet the low power consumption requirement, it is designed to be able to work and wake up in STOP mode, to achieve product power saving function.

20.2 Main Features

- Great anti-jamming performance which meets the EMC(CS) Standard
- Supports 10 channels at most
- Supports low power consumption mode
- Touch interrupt supported
- Clock division supported for charging/discharging
- Supports manual control and automatic mode
- Selective levels for comparator’s threshold
- Waking up threshold configurable in STOP mode

20.3 Architecture

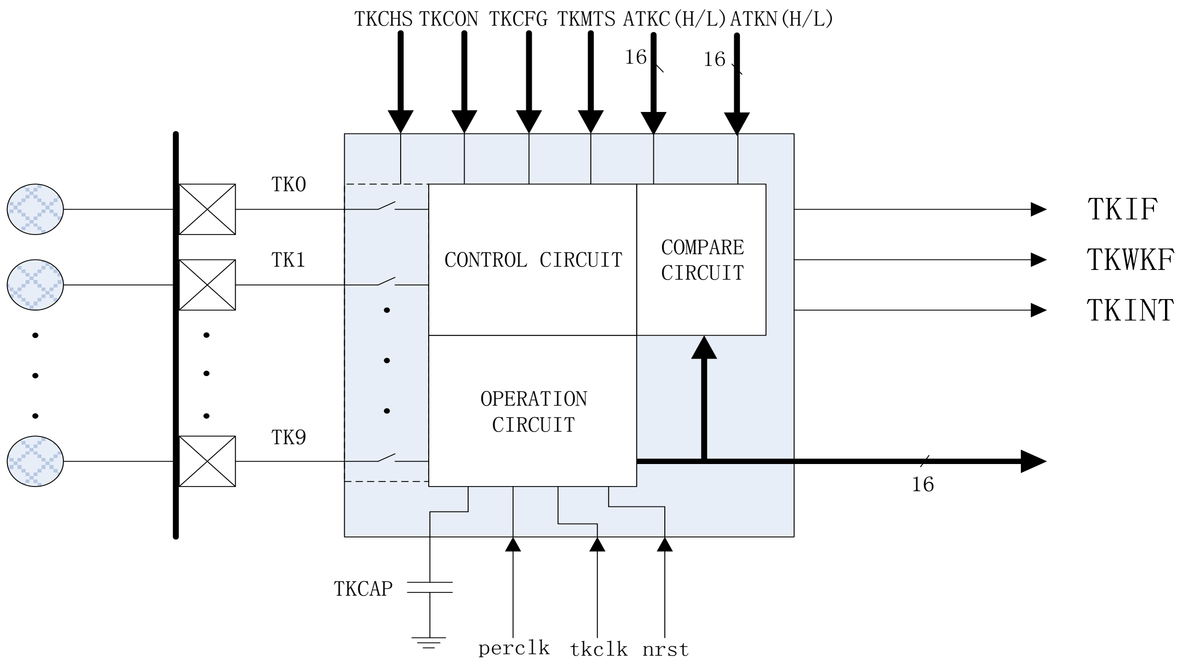


Figure 20-3-1 Touch Key Module Architecture

20.4 Function Description

20.4.2 Manual Control Mode and Automatic Mode

The touch data collection can be enabled by TKST in manual control mode. When TKST=1, the module starts to collect the data through channel selected. There are at most 6 channels for one group for the channels selection, which is set by the register TKnCHS. Every time the collection is enabled, one group of channels' data will be collected. TKST will be cleared automatically when the collection is over. The corresponding channel's interrupt flag TKIF will be set to 1. Then, the touch data can be read from register TKnMS.

Manual control mode and automatic mode can be selected by TMEN. In automatic mode, unlike in manual control mode, the touch data collection is enabled by timer's timing. The clock source for the timer is IRCL; the timing can be set by register TKMTS.

Note: *The "n" in TKnCHS and other registers means 0/1/2/3/4/5.*

20.4.3 Touch Key Clock Frequency Division

The clock source for electrode charging/discharging is IRCH with frequency divided by 4 or IRCL. The clock source for electrode charging/discharging is extremely important for the touch module's performance. When the clock frequency for charging/discharging is too high, the touch electrode may not be charged properly, which makes the data change too small when fingers touch the key. The frequency prescale can be set by TKDIV. With proper frequency, touch module will perform even better. When the touch key frequency hopping function is enabled, the setting of TKDIV is invalid.

20.4.4 Low Power consumption Mode

In order to realize the low power consumption application of the touch key function, the touch key module is designed with the corresponding power saving mechanism. In STOP mode, as long as the touch key charge and discharge clock source (IRCH or IRCL, due to the high static power consumption of IRCH, touch key power saving mode generally choose IRCL as the touch key charge and discharge clock) and low speed clock (IRCL) is on, the touch key module can maintain normal charge and discharge and counting. When the touch acquisition is completed, the touch acquisition completion interrupt will wake up the CPU, and the software can read the touch data after the CPU wakes up, and then enter STOP mode again.

20.4.5 Touch key frequency hopping function

To enhance the touch anti-voltage pulse injection performance, the chip is designed to touch frequency hopping function.

Touch frequency hopping function is enabled by the FAEN bit, after enabling, touch the charge and discharge clock in the touch charge and discharge source clock (IRCH divided by 4 or IRCL) on the basis of each charge and discharge in order to 2, 3, 4, 5 divisions of the order of change, can effectively reduce the voltage pulse injection when a frequency band injection interference signal on the touch interference amplitude.

20.5 Register Description

Table 20-5-1 Register TKCON

F8H	7	6	5	4	3	2	1	0
TKCON	TKST	TKIE	TMEN	FAEN	-	VRS[2:0]		
R/W	R/W	R/W	R/W	R/W	-	R/W		
Initial Value	0	0	0	0	-	1	0	0

Bit number	Bit Symbol	Description
7	TKST	Data collection start enable control, 1 enables it, cleared automatically after the data collection
6	TKIE	TK interrupt enable control, 1 enables it
5	TMEN	Start mode selection 0: enabled by TKST 1: enabled by Timer
4	FAEN	0： Frequency regulation disabled 1： Frequency regulation enabled, the touch clock is switched once per cycle, and the frequency is switched by 1-2-3-4 divisions of the reference frequency in rotation, once per cycle.
3	-	-
2~0	VRS	Reference voltage selection for comparator’s threshold voltage (the threshold voltage is directly proportional with VDD) 0： maximum threshold voltage … 7： minimum threshold voltage

Table 20-5-2 Register TKPWC

DFH	7	6	5	4	3	2	1	0
TKPWC	TKPC		VDS		VIRS		TKPWS	TKCVS
R/W	R/W		R/W		R/W		R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7~6	-	-
5~4	VDS	Internal op-amp output voltage selection 00： 2V 01： 2.5V 10： 3V 11： 4V
3~2	VIRS	Internal voltage reference selection

		00：1.5V 01：2.0V 10：2.5V 11：3.0V
1	TKPWS	Charging power selection 0：Select external power supply 1：Select internal op-amp output
0	TKCVS	Charging reference voltage selection 0：Selecting an external voltage reference 1：Selecting an Internal voltage reference

Table 20-5-3 Register TKCKS

DEH	7	6	5	4	3	2	1	0
TKCKS	-	-	-	-	-	-	-	TKSIL
R/W	-	-	-	-	-	-	-	R/W
Initial Value	-	-	-	-	-	-	-	0
Bit number	Bit Symbol	Description						
7~1	-	-						
0	TKSIL	Touch key sampling clock selection 0：Select IRCH clock divided by 4 (4M) 1：Select IRCL clock (100K)						

Table 20-5-4 Register TKCFG

F9H	7	6	5	4	3	2	1	0
TKCFG	TKDIV			TKTMS				
R/W	R/W			R/W				
Initial Value	0	0	0	1	1	1	1	1
Bit number	Bit Symbol		Description					
7~5	TKDIV		Frequency division selection for touch key clock 000：no division 001：frequency divided by 2 010：frequency divided by 3 … 111：frequency divided by 8					
4~0	TKTMS		The discharging time setting for external modulation capacitor Discharging time = TKTMS x 128 x clock cycle period When TKDIV=0, the discharging time ranges from 32us to 992us ***Note：*** *TKTMS cannot be set to 0*					

Table 20-5-5 Register TKMTS

FAH	7	6	5	4	3	2	1	0
TKMTS	TKMTS[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
Bit number	Bit Symbol		Description					
7~0	TKMTS		The start time setting register in timing mode The start time=(TKMTS+1) × 128 × IRCL cycle period Because the IRCL's frequency is 100KHz, the start time ranges from 1.28ms to 328ms.					

Table 20-5-6 Register TKCHS

FCH	7	6	5	4	3	2	1	0
TK0CHS	-	-	-	-	TKPS0[3:0]			
R/W	-	-	-	-	R/W			
Initial Value	-	-	-	-	0	0	0	0
FDH	7	6	5	4	3	2	1	0
TK1CHS	-	-	-	-	TKPS1[3:0]			
R/W	-	-	-	-	R/W			
Initial Value	-	-	-	-	0	0	0	0
FEH	7	6	5	4	3	2	1	0
TK2CHS	-	-	-	-	TKPS2[3:0]			
R/W	-	-	-	-	R/W			
Initial Value	-	-	-	-	0	0	0	0
FFH	7	6	5	4	3	2	1	0
TK3CHS	-	-	-	-	TKPS3[3:0]			
R/W	-	-	-	-	R/W			
Initial Value	-	-	-	-	0	0	0	0
F1H	7	6	5	4	3	2	1	0
TK4CHS	-	-	-	-	TKPS4[3:0]			
R/W	-	-	-	-	R/W			
Initial Value	-	-	-	-	0	0	0	0
F2H	7	6	5	4	3	2	1	0
TK5CHS	-	-	-	-	TKPS5[3:0]			
R/W	-	-	-	-	R/W			
Initial Value	-	-	-	-	0	0	0	0

Bit number	Bit Symbol	Description
7~4	-	-
3~0	TKPSn	Channel selection 0000： disable TK0~TK9 0001：TK0 selected 0010：TK1 selected 0011： TK2 selected …… 1010： TK9 selected 1011： Internal reference capacitor selected

Table 20-5-9 Register TKMS

E1H	7	6	5	4	3	2	1	0
TK0MSL	TK0MS[7:0]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
E2H	7	6	5	4	3	2	1	0
TK0MSH	TK0MS[15:8]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
E3H	7	6	5	4	3	2	1	0
TK1MSL	TK1MS[7:0]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
E4H	7	6	5	4	3	2	1	0
TK1MSH	TK1MS[15:8]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
E5H	7	6	5	4	3	2	1	0

TK2MSL	TK2MS[7:0]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
E6H	7	6	5	4	3	2	1	0
TK2MSH	TK2MS[15:8]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
E7H	7	6	5	4	3	2	1	0
TK3MSL	TK3MS[7:0]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
D9H	7	6	5	4	3	2	1	0
TK3MSH	TK3MS[15:8]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
DAH	7	6	5	4	3	2	1	0
TK4MSL	TK4MS[7:0]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
DBH	7	6	5	4	3	2	1	0
TK4MSH	TK4MS[15:8]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
DCH	7	6	5	4	3	2	1	0
TK5MSL	TK5MS[7:0]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
DDH	7	6	5	4	3	2	1	0
TK5MSH	TK5MS[15:8]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
Bit number	Bit Symbol		Description					
15~0	TKnMS		Touch key sampling data register					

Table 20-5-10 Register TKIF

FBH	7	6	5	4	3	2	1	0
TKIF	-	-	TKIF5	TKIF4	TKIF3	TKIF2	TKIF1	TKIF0
R/W	-	-	R	R	R	R	R	R
Initial Value	-	-	0	0	0	0	0	0
Bit number	Bit Symbol		Description					
7~6	-		-					
5	TKIF5		TK data collection interrupt flag for the 6th channel, cleared when 1 is written to it					
4	TKIF4		TK data collection interrupt flag for the 5th channel, cleared when 1 is written to it					
3	TKIF3		TK data collection interrupt flag for the 4th channel, cleared when 1 is written to it					
2	TKIF2		TK data collection interrupt flag for the 3rd channel, cleared when 1 is written to it					
1	TKIF1		TK data collection interrupt flag for the 2nd channel, cleared when 1 is written to it					
0	TKIF0		TK data collection interrupt flag for the 1st channel, cleared when 1 is written to it					

21 Low Voltage Detection(LVD)

21.1 Function Introduction

Low voltage Detection (LVD) is used to monitor the chip's power supply VDD, with detectable range 2.7V/3.3V/3.7V/4.2V (Four levels to choose from). When VDD is lower than the voltage set, either interrupt or reset occurs.

Note：*Due the manufacturing process, the LVD trigger voltage may be slightly different.*

Figure 21-1-1 shows the architecture of LVD.

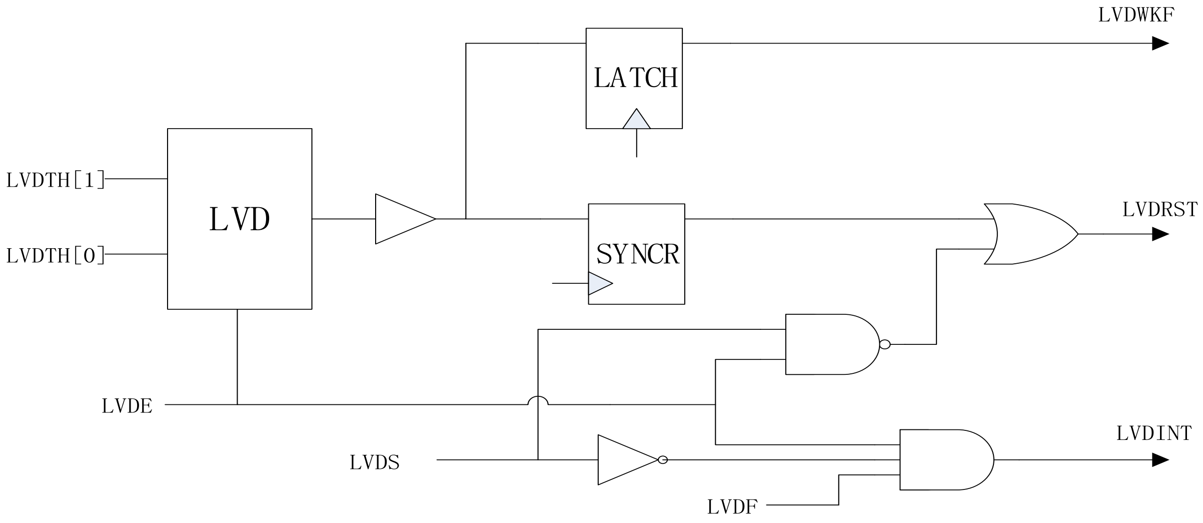


Figure 21-1-1 LVD Schematic

21.2 Function Description

LVD function is enabled by LVDE and the trigger voltage is set by LVDTH. When VDD is lower than the trigger voltage, the LVDF will be set to1. If LVDS=0 then, there will be an LVD interrupt; if LVDS=1, it will generate a reset signal. However, LVD reset signal will not reset itself, which means register LVDCON remains its status. As a result, if VDD is still lower than the trigger voltage set by users after the reset, it will be reset forever. Similarly, the interrupt will occur repeatedly if VDD is still lower than the trigger voltage set by users after the interrupt.

21.3 Register Description

Table 21-3-1 Register LVDCON

EFH	7	6	5	4	3	2	1	0
LVDCON	LVDE	LVDS	LVDF				LVDTH[1:0]	
R/W	R/W	R/W	R/W				R/W	
Initial Value	0	0	0	0	0	0	0	0

Bit number	Bit Symbol	Description
7	LVDE	LVD enable control, 1 enables it
6	LVDS	LVD function selection 0： interrupt 1： reset
5	LVDF	LVD flag, cleared when 1 is written to it
1~0	LVDTH	LVD trigger level selection 00： 2.7V 01： 3.3V 10： 3.7V 11： 4.2V

21.4 LVD Control Example

LVD interrupt example

For instance , set LVD to interrupt mode with trigger voltage 2.7V, the program is like：

```
------------------------------------------------------------------------------------------
#define LVDE(N)          (N<<7)    //N=0~1
#define LVDS_reset       (1<<6)
#define LVDS_int         (0<<6)
#define LVDF             (1<<5)
#define LVDTH_2_7V          0
void LVD_init(void)
{
        LVDCON = LVDE(1) | LVDS_int | LVDTH_2_7V;// enables LVD and set it to interrupt  mode,  set  the  trigger voltage to 2.7V
        INT4EN = 1;    // enables INT4 interrupt
        EA = 1;        // enable total interruption
}
void INT4_ISR (void) interrupt 6
{
        if(LVDCON & LVDF)
        {
                LVDCON |= LVDF;        // clear LVD interrupt flag
                // LVD interrupt service routine
                ...

        }
        ...
}
------------------------------------------------------------------------------------------
```

LVD reset example

For instance , set LVD to reset mode with trigger voltage 2.7V V, the program is like：

```
------------------------------------------------------------------------------------------
#define LVDE(N)          (N<<7)    //N=0~1
#define LVDS_reset       (1<<6)
#define LVDS_int         (0<<6)
#define LVDF             (1<<5)
#define LVDTH_2_7V          0
void LVD_init(void)
{
        LVDCON = LVDE(1) | LVDS_reset | LVDTH_2_7V;// enables LVD and set it to reset mode, set the trigger voltage to 2.7V
}
```

22 Program Download and Simulation

22.1 Program Download

CA51F7 Series chip download programs using ISP method. The chip can connect to the download tool with I2C port. The default upgrade interface is P3.0 (I2C SDA) or P3.1(I2C SCL).

For more download steps please refer to *CACHIP development tools manual*.

22.2 Online Simulation

CA51F7 Series chip supports online simulation. Chip can communicate with the emulator with IIC interface. The default port for IIC is P3.0(IIC SDA) and P3.1(IIC SCL). Since the IIC is used for communication between the chip and emulator, the IIC port cannot be set as other functions and IIC function cannot be used in software either, otherwise the simulation will not be enabled. The speed of IIC is decided by the main clock, so the main clock cannot be set as low speed clock by the software. In addition, it cannot enter power save mode either, otherwise the communication between the chip and emulator will be influenced.

When TSME=0(PCON[3]), the chip is forbidden to enter simulation mode. TSMODE(PCON[2]) will be set to 1 in simulation mode. The software can decide whether to enter power save mode or switch to low speed clock according to the status of TSMODE.

For more details about the simulation function please refer to the documents related to emulator.

23 Electrical Specification

23.1 Limit Parameter

Parameter	Minimum	Maximum	Unit
DC voltage for power supply	-0.3	6	V
Input voltage for I/O pin	-0.3	VDD+0.3	V
Working temperature	-40	85	℃
Storage temperature	-55	125	℃
CPU working frequency		16	MHz

Note：When the parameters exceed the limits above, the working status of the chip is unpredictable which may lead to severe damage to the chip. Working in such environment for a long time will influence the reliability of the chip.

23.2 DC Electrical Specification

DC Electrical Specification（VDD=2.2-5.5V, TA=25℃, unless there are other explanations）

Parameter	Symbol	**Working Voltage**	Minimum	Typical	Maximum	Unit	Condition
Working current	Iop1	VDD=2.5V		3.04		mA	The system clock is IRCH (16MHz), with other clocks disabled. No load for all the output pins. No floating for digital input pins. All the peripherals are disabled, with CPU executing instruction NOP
		VDD=3.3V		3.89			
		VDD=5V		5.95			
	Iop3	VDD=2.5V		30.8		uA	The system clock is IRCL(100kHZ),， with other clocks disabled. No load for all the output pins, No floating for digital input pins. All the peripherals are disabled, with CPU executing instruction NOP
		VDD=3.3V		36.8			
		VDD=5V		68.9			
Current for STOP mode	Istp	VDD=2.5V		5.6		uA	All the clocks disabled. No load for all the output pins, No floating for digital input pins. All the peripherals are disabled, flash enters sleep mode and CPU enters STOP mode.
		VDD=3.3V		5.7			
		VDD=5V		5.8			
Current for IDLE mode	Iidl1	VDD=2.5V		1.48		mA	The system clock is IRCH (16MHz), with other clocks disabled. No load for all the output pins. No floating for digital input pins. All the peripherals are disabled, flash enters sleep mode and CPU enters IDLE mode.
		VDD=3.3V		1.96			
		VDD=5V		3.14			
	Iidl2	VDD=2.5V		15.6		uA	The system clock is IRCL(100kHZ),， with other clocks disabled. No load for all the output pins, No floating for digital input pins. All the peripherals are disabled, CPU enters IDLE mode.
		VDD=3.3V		19.1			
		VDD=5V		27.7			

High voltage for IO port input (Schmidt mode on)	Vhi1	VDD=2.5V	0.75	-	1.8	V	-
		VDD=3.3V	1.20		3.3		
		VDD=5V	1.50		5		
High voltage for IO port input (Schmidt mode close)	Vhi2	VDD=1.8V		0.5*VDD	VDD	V	-
		VDD=3.3V					
		VDD=5V					
Low voltage for IO port input (Schmidt mode on)	Vlo1	VDD=1.8V	0	-	0.62	V	-
		VDD=3.3V	0	-	0.85		
		VDD=5V	0	-	1.20		
Low voltage for IO port input (Schmidt mode close)	Vlo2	VDD=1.8V	0	0.5*VDD		V	-
		VDD=3.3V					
		VDD=5V					
IO port push current	Ipu	VDD=3.3V	-	3.08	-	mA	IO set to push-pull output mode, drive power set to maximum, Vol=VDD-0.3V
		VDD=5V	-	4.1	-		
IO port sink current	Iol	VDD=3.3V	-	6.01	-	mA	IO set to push-pull output mode, drive power set to maximum, Vol=GND+0.3V
		VDD=5V	-	6.72	-		
P0.7 strong push current	Ipu	VDD=3.3V		33.6		mA	IO set to push-pull output mode or PWM pin function, DRV_EN/SINK_En is set to 1, GDRV/ SINK is set to 7, Vol=VDD-0.3V
		VDD=5V		41.1			
P0.7 strong sink current	Isi	VDD=3.3V		47.8		mA	IO set to push-pull output mode or PWM pin function, DRV_EN/SINK_EN is set to 1, GDRV/ SINK is set to 7， Vol=GND+0.3V
		VDD=5V		55.5			
P1.0 strong sink current	Isi	VDD=3.3V		82.7		mA	IO set to push-pull output mode or PWM pin function, DRV_EN/SINK_EN is set to 1， GDRV/ SINK is set to 7， Vol=GND+0.3V
		VDD=5V		99.6			
P1.1、P1.2、P1.3、P1.4 strong sink current	Isi	VDD=3.3V		45.3		mA	IO set to push-pull output mode or PWM pin function, DRV_EN/SINK_EN is set to 1, GDRV/ SINK is set to 3， Vol=GND+0.3V
		VDD=5V		55.8			
Pull-down resistor for IO port	Rd1	VDD=2.5~5.5V		30		KΩ	-
Pull-up resistor for IO port	Ru1	VDD=2.5~5.5V	-	30	-	KΩ	-

23.3 AC Electrical Specification

AC Electrical Specification (VDD=2.2-5.5V, TA=25℃, unless there are other explanations)

Parameter	Symbol	Minimum	Typical	Maximum	Unit	Condition
Time to start oscillation for IRCL	Trc1	-	50	-	us	IRCL frequency 100K
Time to start oscillation for IRCH	Trc2	-	10	-	us	IRCH frequency 16MHz
Time of the reset pulse	Trst	-	0.5	-	us	

Note： VDD=3.3V,TA=25□, the factory frequency for internal high speed clock is 16MHz, with deviation less than 1%.

23.4 Minimum working voltage

CPU frequency **(**Unit： MHz**)**	**Minimum working voltage** **(**Unit**:V)**
8	2.2~5.5
16	2.7~5.5

23.5 Temperature Characteristic for Internal RC Oscillator

◆ Temperature Characteristic for IRCH

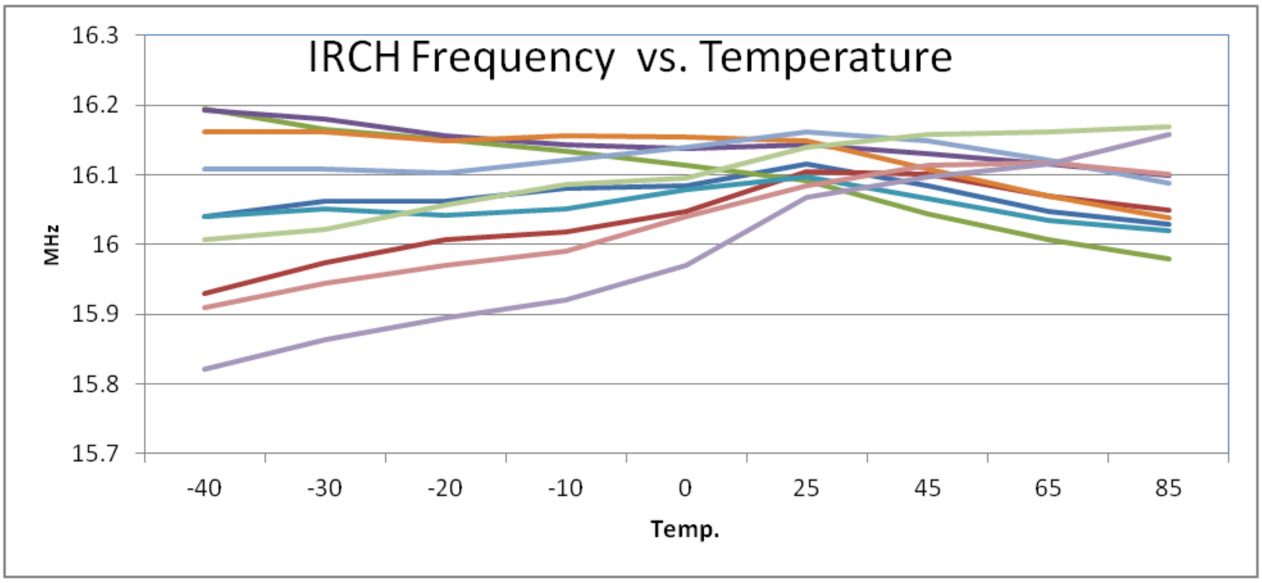


Figure 23-5-1 Temperature characteristic curve of IRCH

Note： The data above comes from single chip and can be only used for reference.

◆ Temperature Characteristic for IRCL

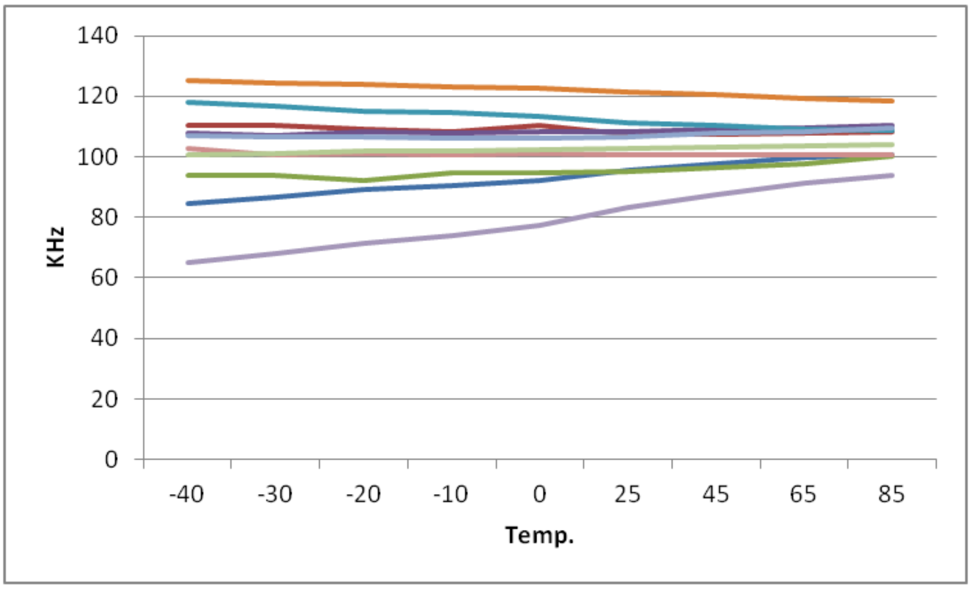
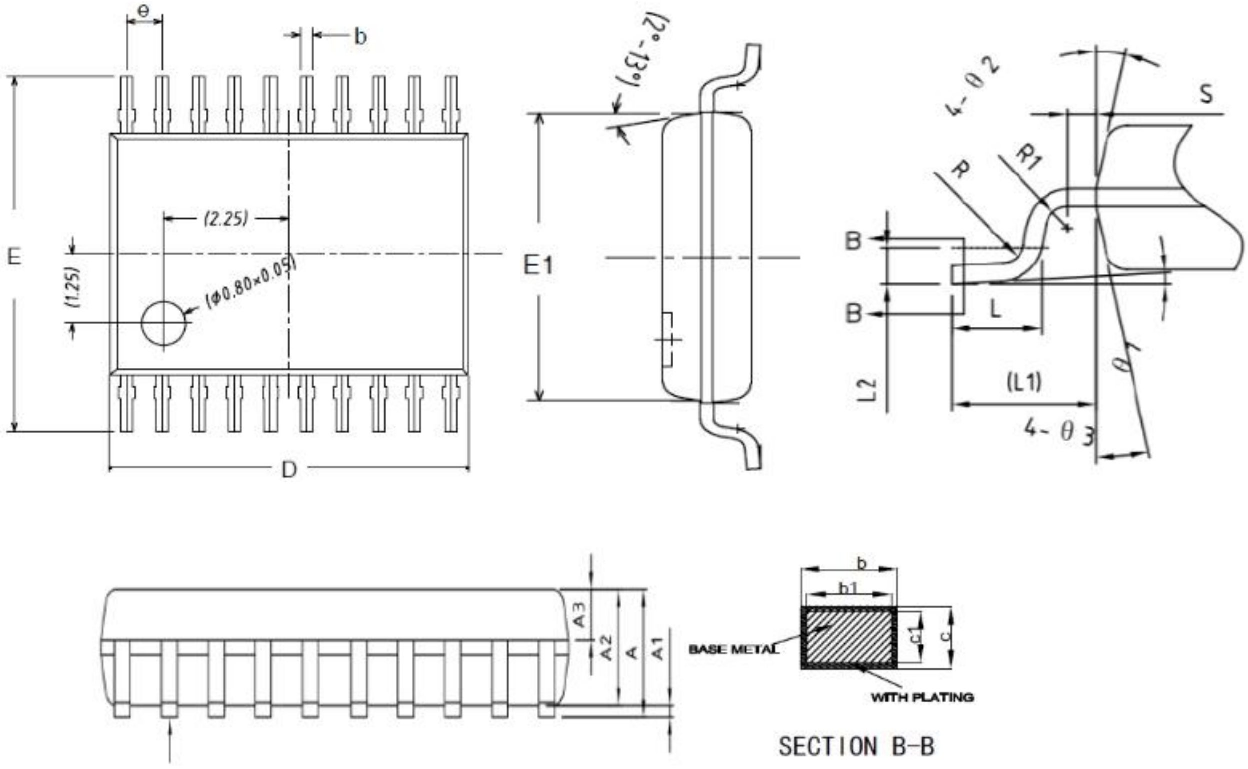


Figure 23-5-2 Temperature characteristic curve of IRCL

Note： The data above comes from single chip and can be only used for reference.

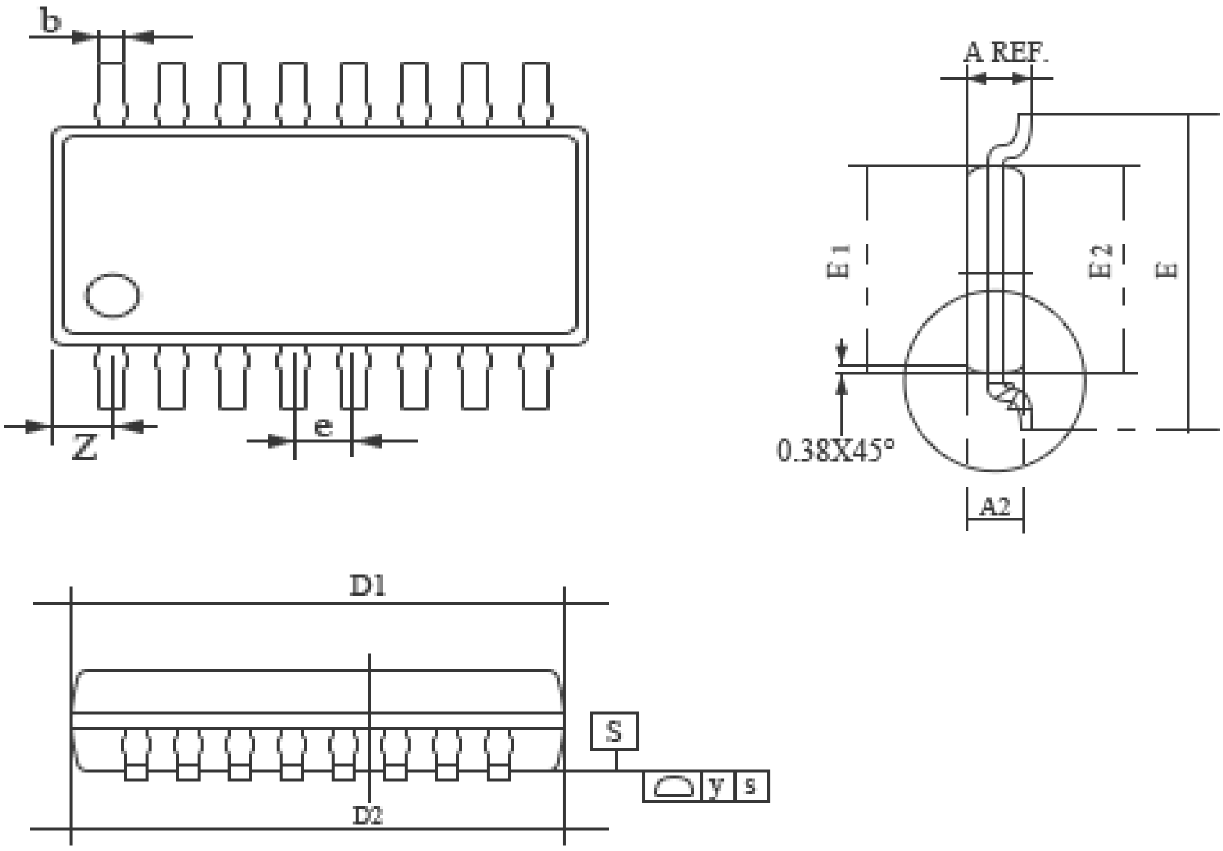
24 Package Type

Package (I) (TSSOP20)



Sequence number	Minimum (mm)	Standard (mm)	Maximum (mm)
A	1.0	---	1.1
A1	0.05	---	0.15
A2	---	---	0.95
A3	0.39	---	0.40
b	0.20	0.22	0.24
c	0.10	---	0.19
c1	0.10		0.15
D	6.40	6.45	6.50
E	6.25	6.40	6.55
E1		4.35	4.40
L	0.50	0.60	0.70
e	0.55	0.65	0.75
L2	0.25BSC		
R	0.09		
L1	1.0REF		

Package(II) (SOP16)



Symbol	Minimum(mm)	Typical(mm)	Maximum(mm)
A	1.500	1.600	1.700
A2	1.400	1.450	1.500
b	0.356	0.406	0.456
D1	9.70	9.90	10.10
D2	9.75	9.95	10.15
E	5.90	6.000	6.100
E1	3.800	3.900	4.000
E2	3.850	3.950	4.050
e	————	1.27	————
Z	————	0.505	————

25 Appendix

Appendix 1 Instruction set Quick Reference Table

Mnemonic	Description	Description	Cycles
DATA TRANSFER			
MOV A,Rn	Move register to A	(A) ← (Rn)	1
MOV A,direct	Move direct byte to A	(A) ← (direct)	1
MOV A,@Ri	Move indirect RAM to A	(A) ← ((Ri))	1
MOV A,#data8	Move 8-bit immediate data to A	(A) ← #data	1
MOV Rn,A	Move A to register	(Rn) ← (A)	1
MOV Rn,direct	Move direct byte to register	(Rn) ← (direct)	2
MOV Rn,#data8	Move 8-bit immediate data to register	(Rn) ← #data	1
MOV direct,A	Move A to direct byte	(direct) ← (A)	1
MOV direct,Rn	Move register to direct byte	(direct) ← (Rn)	2
MOV direct,direct	Move direct byte to direct byte	(direct) ← (direct)	2
MOV direct,@Ri	Move indirect RAM to direct byte	(direct) ← ((Ri))	2
MOV direct,#data8	Move 8-bit immediate data to direct byte	(direct) ← #data	2
MOV @Ri,A	Move A to indirect RAM	((Ri)) ← (A)	1
MOV @Ri,direct	Move direct byte to indirect RAM	((Ri)) ← (direct)	2
MOV @Ri,#data8	Move 8-bit immediate data to indirect RAM	((Ri)) ← #data	1
MOV DPTR,#data16	Load Data Pointer with 16-bit constant	(DPTR) ← #data116	2
MOV A,@A+DPTR	Move Code byte relative to DPTR to A	(A) ← ((A)) + (DPTR)	2
MOV A,@A+PC	Move Code byte relative to FFe to A	(PC) ← (PC) + 1 (A) ← ((A) + (PC))	2
MOVX A,@Ri	Move External RAM (8-bit addr) to A	(A) ← ((Ri))	2
MOVX A,@DPTR	Move External RAM (16-bit addr) to A	(A) ← ((DPTR))	2
MOVX @Ri,A	Move A to External RAM (8-bit addr)	((Ri)) ← (A)	2
MOVX @DPTR,A	Move A to External RAM (16-bit addr)	(DPTR) ← (A)	2
PUSH direct	Push direct byte onto stack	(SP) ← (SP) + 1 ((SP)) ← (direct)	2
POP DIRECT	Pop direct byte from stack	(direct) ← ((SP)) (SP) ← (SP) - 1	2
XCH A,Rn	Exchange register with A	(A) ↔ (Rn)	1
XCH A,direct	Exchange direct byte with A	(A) ↔ (direct)	1
XCH A,@Ri	Exchange indirect RAM with A	(A) ↔ ((Ri))	1
XCHD A,@Ri	Exchange low-order Digit indirect RAM with A	(A.3,...,A.0) ↔ ((Ri).3,...,(Ri).0)	1
SWAP A	Swap nibbles within A	(A.3,...,A.0) ↔ (A.7,...,A.4)	1
ARITHMETIC OPERATIONS			
ADD A, Rn	Add register to A	(A) ← (A) + (Rn)	1
ADD A, direct	Add direct byte to A	(A) ← (A) + (direct)	1
ADD A, @Ri	Add indirect RAM to A	(A) ← (A) + ((Ri))	1

ADD A, #data8	Add 8-bit immediate data to A	(A) ← (A) + #data	1
ADDC A, Rn	Add register to A with Carry	(A) ← (A) + (C) + (Rn)	1
ADDC A, direct	Add direct byte to A with Carry	(A) ← (A) + (C) + (direct)	1
ADDC A, @Ri	Add indirect RAM to A with Carry	(A) ← (A) + (C) + ((Ri))	1
ADDC A, #data8	Add 8-bit immediate data to A with Carry	(A) ← (A) + (C) + #data	1
SUBB A, Rn	Subtract register from A with Borrow	(A) ← (A) - (C) - (Rn)	1
SUBB A, direct	Subtract direct byte from A with Borrow	(A) ← (A) - (C) - (direct)	1
SUBB A, @Ri	Subtract indirect RAM from A with Borrow	(A) ← (A) - (C) - ((Ri))	1
SUBB A, #data8	Subtract immediate data from A with Borrow	(A) ← (A) - (C) - #data	1
INC A	Increment A	(A) ← (A) + 1	1
INC Rn	Increment register	(Rn) ← (Rn) + 1	1
INC direct	Increment direct byte	(direct) ← (direct) + 1	1
INC @Ri	Increment indirect RAM	((Ri)) ← ((Ri)) + 1	1
INC DPTR	Increment Data Pointer	(DPTR) ← (DPTR) + 1	2
DEC A	Decrement A	(A) ← (A) – 1	1
DEC Rn	Decrement register	(Rn) ← (Rn) - 1	1
DEC direct	Decrement direct byte	(direct) ← (direct) - 1	1
DEC @Ri	Decrement indirect RAM	((Ri)) ← ((Ri)) - 1	1
MUL AB	Multiply A & B (A × B => BA)	temp16 ← (A) X (B) (A)←(temp.7,temp.6,...,temp.0) (B)←(temp.15,temp.14,...,temp.8)	4
DIV AB	Divide A by B(A/B => A +B)	QUO ← (A) / (B) ......REM (A) ← QUO (B) ← REM	4

DA A	Decimal Adjust A	IF (A.3,...,A.0) > 9 || AC = 1
		THEN
		temp16 ← (A) + 0x06
		(A) ← (temp.7,...,temp.0)
		
		IF (temp16) > 0xFF
		THEN
		CY ← 1
		
		IF (A.7,...,A.4) > 9 || CY = 1
		THEN
		temp16 ← (A) + 0x60
			1

		(A) ← (temp.7,...,temp.0)
		IF (temp16) > 0xFF
		THEN
		CY ← 1	

LOGICAL OPERATIONS			
ANL A, Rn	AND register to A	(A) ← (A) & (Rn)	1
ANL A, direct	AND direct byte to A	(A) ← (A) & (direct)	1
ANL A, @Ri	AND indirect RAM to A	(A) ← (A) & ((Ri))	1
ANL A, #data8	AND 8-bit immediate data to A	(A) ← (A) & #data	1
ANL direct, A	AND A to direct byte	(direct) ← (direct) & (A)	1
ANL direct, #data8	AND 8-bit immediate data to direct byte	(direct) ← (direct) & #data	2
ORL A, Rn	OR register to A	(A) ← (A) | (Rn)	1
ORL A, direct	OR direct byte to A	(A) ← (A) | (direct)	1
ORL A, @Ri	OR indirect RAM to A	(A) ← (A) | ((Ri))	1
ORL A, #data8	OR 8-bit immediate data to A	(A) ← (A) | #data	1
ORL direct, A	OR A to direct byte	(direct) ← (direct) | (A)	1
ORL direct, #data8	OR 8-bit immediate data to direct byte	(direct) ← (direct) | #data	2
XRL A, Rn	Exclusive-OR register to A	(A) ← (A) ^ (Rn)	1
XRL A, direct	Exclusive-OR direct byte to A	(A) ← (A) ^ (direct)	1
XRL A, @Ri	Exclusive-OR indirect RAM to A	(A) ← (A) ^ ((Ri))	1
XRL A, #data8	Exclusive-OR 8-bit immediate data to A	(A) ← (A) ^ #data	1
XRL direct, A	Exclusive-OR A to direct byte	(direct) ← (direct) ^ (A)	1
XRL direct, #data8	Exclusive-OR 8-bit immediate data to direct byte	(direct) ← (direct) ^ #data	2
CLR A	Clear A	(A) ← 0	1
CPL A	Complement A	(A) ← /(A)	1
RL A	Rotate A Left	(A) ← (A.6,A.5,...,A.0,A.7)	1
RLC A	Rotate A Left through Carry	C ← A.7 (A) ← (A.6,A.5,...,A.0,C)	1
RR A	Rotate A Right	(A) ← (A.0,A.7,...,A.2,A.1)	1
RRC A	Rotate A Right through Carry	C ← A.0 (A) ← (C,A.7,...,A.2,A.1)	1
PROGRAM AND MACHINE CONTROL			
ACALL addr11	Absolute subroutine call	(PC) ← (PC) + 2 (SP) ← (SP) + 1 ((SP)) ← (PC7-0) (SP) ← (SP) + 1 ((SP)) ← (PC15-8) (PC10-0) ← page address	2
LACLL addr16	Long subroutine call	(PC) ← (PC) + 3 (SP) ← (SP) + 1	2

		((SP)) ← (PC7-0) ((SP)) ← (PC15-8) (PC) ←addr15-0	
RET	Return from subroutine	(PC15-8) ← ((SP)) (SP) ← (SP) - 1 (PC7-0) ← ((SP)) (SP) ← (SP) - 1	2
RETI	Return from interrupt	(PC15-8) ← ((SP)) (SP) ← (SP) - 1 (PC7-0) ← ((SP)) (SP) ← (SP) - 1	2
AJMP addr11	Absolute Jump	(PC) ← (PC) + 2 (PC10-0) ← page address	2
LJMP addr16	Long Jump	(PC) ← (PC) + 3 (SP) ← (SP) + 1 ((SP)) ← (PC7-0) (SP) ← (SP) + 1 ((SP)) ← (PC15-8) (PC10-0) ←addr15-0	2
SJMP rel	Short Jump (relative addr)	(PC) ← (PC) + 2 (PC) ← (PC) + rel	2
JMP @A+DPTR	Jump indirect relative to DPTR	(PC) ← (A) + (DPTR)	2

JZ rel	Jump if A is Zero	(PC) ← (PC) + 2
		IF (A) = 0
		THEN
		(PC) ← (PC) + rel
			2

JNZ rel	Jump if A is Not Zero	(PC) ← (PC) + 2
		IF (A) <> 0
		THEN
		(PC) ← (PC) + rel
			2

CJNE A, direct, rel	Compare direct to A & Jump if Not Equal	(PC) ← (PC) + 3
		IF (A) <> (direct)
		THEN
		(PC) ← (PC) + relative offset
		IF (A) < (direct)
		THEN
		(C) ← 1
		ELSE
		(C) ← 0
			2

CJNE A, #data8, rel	Compare 8-bit immediate to A & Jump if Not Equal	(PC) ← (PC) + 3
		IF (A) <> data
		THEN
		(PC) ← (PC) + relative offset
		IF (A) < data
		THEN
		(C) ← 1
		ELSE
		(C) ← 0
			2

CJNE Rn, #data8, rel	Compare 8-bit immediate to reg. & Jump if Not Equal	(PC) ← (PC) + 3
		IF (Rn) <> data
		THEN
		(PC) ← (PC) +
			2

		relative offset
		IF (Rn) < data
		THEN
		(C) ← 1
		ELSE
		(C) ← 0	

CJNE @Ri, #data8, rel	Compare 8-bit immediate to ind. & Jump if Not Equal	(PC) ← (PC) + 3
		IF ((Ri)) <> data
		THEN
		(PC) ← (PC) + relative offset
		IF ((Ri)) < data
		THEN
		(C) ← 1
		ELSE
		(C) ← 0
			2

DJNZ Rn, rel	Decrement register & Jump if Not Zero	(PC) ← (PC) + 2 (Rn) ← (Rn) - 1 IF (Rn) <> 0 THEN (PC) ← (PC) + rel	2
DJNZ direct, rel	Decrement direct byte & Jump if Not Zero	(PC) ← (PC) + 2 (direct) ← (direct) - 1 IF (direct) <> 0 THEN (PC) ← (PC) + rel	2
NOP	No operation	(PC) ← (PC) + 1	1
BOOLEAN VARIABLE MANIPULATION			
CLR C	Clear Carry flag	(C) ← 0	1
CLR bit	Clear direct bit	(bit) ← 0	1
SETB C	Set Carry flag	(C) ← 1	1
SETB bit	Set direct bit	(bit) ← 1	1
CPL C	Complement Carry flag	(C) ← /(C)	1
CPL bit	Complement direct bit	(bit) ← /(bit)	1
ANL C, bit	AND direct bit to Carry flag	(C) ← (C) & (bit)	2
ANL C, /bit	AND complement of direct bit to Carry flag	(C) ← (C) & /(bit)	2
ORL C, bit	OR direct bit to Carry flag	(C) ← (C) | (bit)	2
ORL C, /bit	OR complement of direct bit to Carry flag	(C) ← (C) | /(bit)	2
MOV C, bit	Move direct bit to Carry flag	(C) ← (bit)	1
MOV bit, C	Move Carry flag to direct bit	(bit) ← (C)	2
JC rel	Jump if Carry flag is set	(PC) ← (PC) + 2 IF (C) = 1 THEN (PC) ← (PC) + rel	2
JNC rel	Jump if No Carry flag	(PC) ← (PC) + 2 IF (C) = 0 THEN (PC) ← (PC) + rel	2
JB bit, rel	Jump if direct Bit is set	(PC) ← (PC) + 3 IF (bit) = 1 THEN (PC) ← (PC) + rel	2
JNB bit, rel	Jump if direct Bit is Not set	(PC) ← (PC) + 3 IF (bit) = 0 THEN (PC) ← (PC) + rel	2
JBC bit, rel	Jump if direct Bit is set & Clear bit	(PC) ← (PC) + 3 IF (bit) = 1 THEN (bit) ← 0 (PC) ← (PC) + rel	2

Pseudo-command	
ORG	Set program start address
END	Mark the end of source code
EQU	Define constants
SET	Define integer numbers
DATA	Assign a value to the data address
BYTE	Assigning values to byte type symbols
WROD	Assigning values to word type symbols
BIT	Name the address of the bit
ALTNAME	Replace reserved words with custom names
DB	Load a contiguous block of memory with byte-type data
DW	Load a contiguous block of memory with word data
DS	Set aside a contiguous storage area or load specified bytes
INCLUDE	Insert a source file into the program
TITLE	Add a header row to the list file
NOLIST	No list file is generated during assembly
NOCODE	When the condition is compiled, the list is not generated if the condition is false