Built-in 12 Bit ADC / 16 Bit PWM / Touch Key / 16K FLASH 8-bit MCU

CA51F152S3 Series MCU User Guide

REV1.2

Directory

1 Introduction

The CA51F152S3 series chip is an 8-bit microcontroller based on the 1T 8051 core, which not only retains the basic characteristics of traditional 8051 chips, but also runs 10 times faster than traditional 8051 chips in general, with superior performance. The chip is equipped with a 16KB Flash program memory, which can be programmed multiple times, and 768 bytes of SRAM to facilitate customer development of complex applications. It also integrates 14 channels of 12 Bit ADC, 14 channels of Touch Key (without external capacitors), 16 Bit PWM, I2C, 2 channels of UART, SPI, RTC, 6Bit DAC, TMC, low voltage detection (LVD) and other functional modules. Supports PWM, I2C, UART peripherals that can be mapped to different pins, and also supports three power-saving modes: IDLE, STOP, and low-speed operation to adapt to applications with different power consumption requirements. Its powerful functions and superior anti-interference performance make it widely applicable in household appliances.

2 Basic Features

- Core
  - CPU: 1T 8051, up to 10 times faster than traditional 8051
  - Compatible with 8051 instruction set, dual DPTR operation mode
- Memory
  - Flash: 16K bytes, supports multiple rewrites
  - Flash can be divided into program area and data area, the data area can be used to save data that needs to be saved after power down
  - RAM:256 bytes of internal RAM, 512 bytes of external RAM
- Operating Voltage
  - Operating voltage: 2.3V ~ 5.5V
- Operating temperature
  - Operating temperature: -40℃~+85℃
- Clock System
  - External low-speed oscillator: 32.768KHz
  - Built in low-speed RC oscillator: 128KHz
  - Built in high-speed RC oscillator: 16MHz, accuracy of ± 2% @ 5V/25℃ (factory calibrated)
    Voltage 2.7V~5.5V supports system clock 16MHz
    Voltage 2.3V~5.5V supports system clock 8MHz
- Timer
  - Three 16-bit general-purpose timers: Timer 0, Timer 1, Timer 2
- General purpose input and output ports (GPIO)
  - Supports up to 14 GPIO ports
  - Supports push-pull, open drain, pull-up , pull-down , and high resistance modes

- When pushing and pulling output, the maximum current of a single GPIO source can reach 18ma@5v The maximum current can reach 40ma@5v
- GPIO can be software simulated as a 1/2 BIAS LCD driver without the need for external resistors

◆ TMC Functions

- The clock source of TMC timer can be selected as IRCL or XOSCL
- The smallest unit of interruption time, when selecting IRCL as the clock source, it is 512 IRCL clock cycles, and when selecting XOSCL as the clock source, it is 128 XOSCL clock cycles
- Configurable interrupt time from 1 to 256 minimum units of time

◆ Interrupt system

- 7 valid interrupt sources
- Two levels of interrupt priority, supporting nested interrupts
- 5 external interrupt sources INT0~INT4
- External interrupt triggering edge selection: INT0~1 (rising edge, falling edge), INT2~4 (rising edge, falling edge, double edge)
- Interrupt input pin selection: INT0 (P0.0), INT1 (P0.1), INT2 (P0.2), INT3 (P0.3), INT4. Any GPIO pin except for P0.0~P0.3 can be selected as the interrupt input pin

◆ Analog/Digital Converters (ADC/DAC)

- Support 14 channel 12 bit SAR ADC
- Supports two reference voltage sources: VDD and internal reference
- Selecting internal voltage as the reference, VDD voltage can be measured
- Supports one 6-bit DAC, divided into 64 voltage stages, and can be used for AD acquisition button function

◆ Touch Key

- Built in touch sensing controller
- Supports up to 14 touch channels , without the need for external capacitors
- High anti-interference performance, in compliance with EMC (CS dynamic 10V, EFT 4KV) standards
- Support touch interruption
- Support parallel connection of touch channels, which can be used to achieve low power consumption mode

◆ PWM

- Supports 4 PWM outputs, each of which can be individually controlled, and the cycle and duty cycle can be freely configured within a 16 bit range
- Support for direct output of internal clock function
- Support PWM interrupt function
- Each PWM output pin can be mapped to different GPIO pins

◆ Universal Serial Interface（UART1/UART2）

- Supports 2 full duplex serial ports
- Support 1-byte receive cache

- The TX/RX of UART1/UART2 can be mapped to different GPIO pins

- I²C interface
  - Built in 1 I2C interface, supporting master-slave mode, standard/fast/high-speed mode
  - I2C pin SCL/SDA can be mapped to any GPIO pin
- SPI interface
  - Built in 1 4-wire SPI interface, supporting master-slave mode
- Low voltage detection（LVD）
  - The detection voltage can be set to 2.2V, 2.5V, 2.7V, 3.0V, 3.3V, 3.6V, 3.9V, and 4.2V.
  - Low voltage reset or interrupt can be set
- Reset mode
  - The chip supports multiple reset sources: power on/power off reset, soft reset, hard reset, watchdog reset, low voltage detection reset
- Watchdog
  - 27 bit watchdog timer, 16 bit adjustment accuracy, configurable watchdog reset or interrupt
- Program download and simulation
  - Support ISP and IAP
  - Support online simulation function
- Low power consumption
  - STOP mode, current<7uA
  - IDLE mode, current<30uA
  - Low speed operation mode, current<55uA
- Package Type：SOP16

3 Chip Model Function Introduction

Table 3-1 CA51F152S3X series specific model functional characteristics

Chip Model	Flash Capacity[BYTE]	SRam[BYTE]	Internal high-speed RC oscillator	Internal low-speed RC oscillator	external oscillator	Number of GPIO	Number of universal 16 bit timers	Number of UART	I²C	SPI	Number of 16 bit PWM channels	Capacitive touch buttons	Number of 12-bit ADC channels	ISP	On-chip emulation functions	Operating Voltage [V]	Package form
CA51F152S3A **CA51F152S3B**	16K	768	√	√	√	14	3	2	1	1	4	14	14	√	√	2.3~5.5	SOP28

4 System Block Diagram

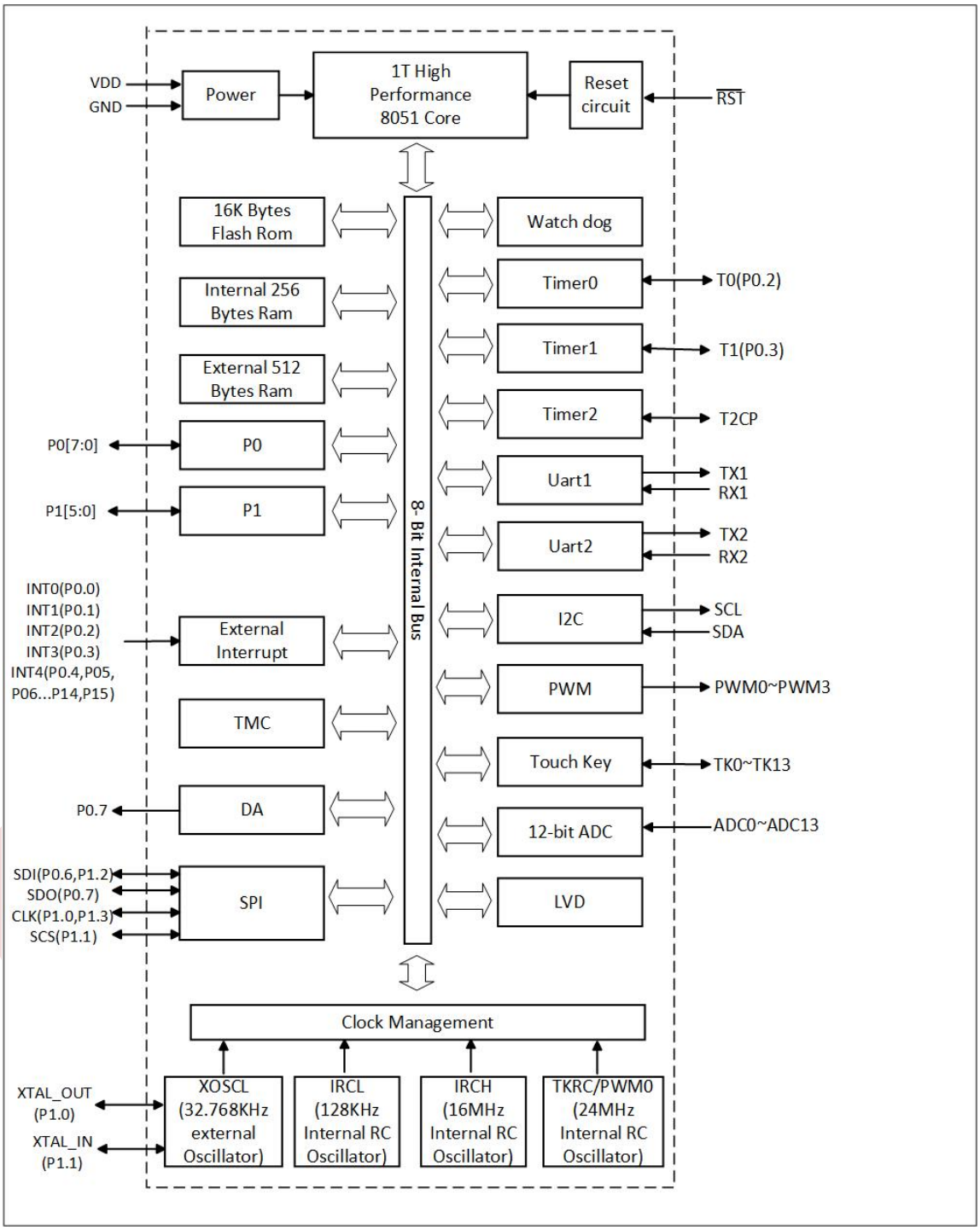


Figure 4-1-1 Chip Block Diagram

5 Pinout Packages and their Descriptions

5.1 Package Definition

Model：CA51F152S3A

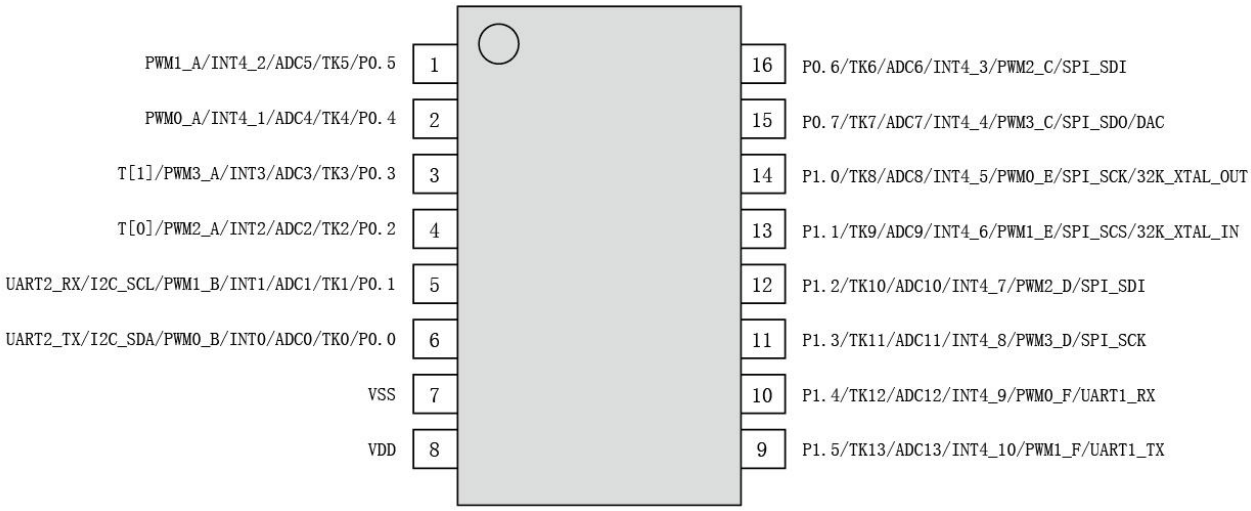


Figure 5-1-1 SOP16 Package Diagram

Model：CA51F152S3B

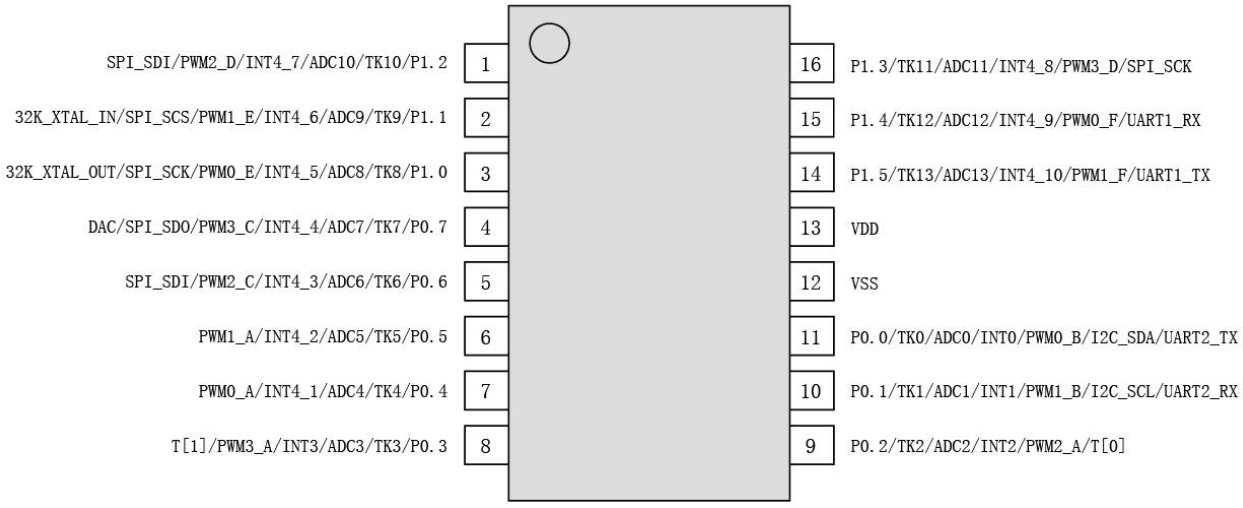


Figure 5-1-2 SOP16 Package Diagram

Remarks:

(1) For the convenience of design applications, the RX/TX of UART1/UART2 can be mapped to different GPIO pins through settings, as described in the "15-2-5 Pin Reuse Function Mapping Table" for details.

(2) For the convenience of designing applications, the SCL/SDA of IIC can be mapped to any GPIO pin through settings, as described in the "15-2-5 Pin Multiplexing Function Mapping Table" for details. The simulation pins for downloading the CA51F152S3A / CA51F152S3B program are P0.0 (IIC-SDA) and P0.1 (IIC-SCL)

5.2 Pin Description

Table 5-2-1 Pin Description

Pin Serial Number		Pin Name	Pin Function	Default Function
SOP16 (A)	SOP16 (B)			
1	6	P0.5/INT4_2/PWM1_A/ADC5/TK5/I2C_SDA/I2C_SCL/ UART2_RX/UART2_TX/T2CP	General purpose bi-directional I/O ports INT4 signal input PWM1 signal output ADC analog channel input Touch key analog channel input I^2C transfer port UART2 transfer port T2CP signal input	General purpose bi-directional I/O ports
2	7	P0.4/INT4_1/PWM0_A/ADC4/TK4/I2C_SDA/I2C_SCL/ UART2_RX/UART2_TX/T2CP	General purpose bi-directional I/O ports INT4 signal input PWM0 signal output ADC analog channel input Touch key analog channel input I^2C transfer port UART2 transfer port T2CP signal input	General purpose bi-directional I/O ports
3	8	P0.3/INT3/T1/PWM3_A/ADC3/TK3/I2C_SDA/I2C_SCL/ UART2_RX/UART2_TX/T2CP	General purpose bi-directional I/O ports INT3 signal input T1 input port PWM3 signal output ADC analog channel input Touch key analog channel input I^2C transfer port UART2 transfer port T2CP signal input	General purpose bi-directional I/O ports
4	9	P0.2/INT2/T0/PWM2_A/ADC2/TK2/I2C_SDA/I2C_SCL/ UART2_RX/UART2_TX/T2CP	General purpose bi-directional I/O ports INT2 signal input T0 input port PWM2 signal output ADC analog channel input Touch key analog channel input I^2C transfer port UART2 transfer port T2CP signal input	General purpose bi-directional I/O ports

5	10	P0.1/INT1/PWM1_B/ADC1/TK1/I2C_SDA/I2C_SCL/ UART2_RX/UART2_TX/T2CP	General purpose bi-directional I/O ports INT1 signal input PWM1 signal output ADC analog channel input Touch key analog channel input I²C transfer port UART2 transfer port T2CP signal input	I²C transfer port
6	11	P0.0/INT0/PWM0_B/ADC0/TK0/I2C_SDA/I2C_SCL/ UART2_RX/UART2_TX/T2CP	General purpose bi-directional I/O ports INT0 signal input PWM0 signal output ADC analog channel input Touch key analog channel input I²C transfer port UART2 transfer port T2CP signal input	I²C transfer port
7	12	VSS	Power ground pin	Power ground pin
8	13	VDD	Chip power supply pins	Chip power supply pins
9	14	P1.5/INT4_10/PWM1_F/ADC13/TK13/I2C_SDA/I2C_SCL/UART1_RX/UART1_TX/T2CP	General purpose bi-directional I/O ports INT4 signal input PWM1 signal output ADC analog channel input Touch key analog channel input I²C transfer port UART1 transfer port T2CP signal input	General purpose bi-directional I/O ports
10	15	P1.4/INT4_9/PWM0_F/ADC12/TK12/I2C_SDA/I2C_SCL/UART1_RX/UART1_TX/T2CP	General purpose bi-directional I/O ports INT4 signal input PWM0 signal output ADC analog channel input Touch key analog channel input I²C transfer port UART1 transfer port T2CP signal input	General purpose bi-directional I/O ports
11	16	P1.3/INT4_8/PWM3_D/ADC11/TK11/I2C_SDA/I2C_SCL/UART1_RX/UART1_TX/SPI_SCK	General purpose bi-directional I/O ports INT4 signal input PWM3 signal output ADC analog channel input	General purpose bi-directional I/O ports

			Touch key analog channel input I²C transfer port UART1 transfer port SPI_SCK port	
12	1	P1.2/INT4_7/PWM2_D/ADC10/TK10/I2C_SDA/I2C_SCL /UART1_RX/UART1_TX/SPI_SDI	General purpose bi-directional I/O ports INT4 signal input PWM2 signal output ADC analog channel input Touch key analog channel input I²C transfer port UART1 transfer port SPI_SDI port	General purpose bi-directional I/O ports
13	2	P1.1/INT4_6/PWM1_E/ADC9/TK9/I2C_SDA/I2C_SCL/ UART1_RX/UART1_TX/SPI_SCS/32K_XTAL_IN	General purpose bi-directional I/O ports INT4 signal input PWM1 signal output ADC analog channel input Touch key analog channel input I²C transfer port UART2 transfer port SPI_SCS port 32K External crystal oscillator input	General purpose bi-directional I/O ports
14	3	P1.0/INT4_5/PWM0_E/ADC8/TK8/I2C_SDA/I2C_SCL/ UART1_RX / UART1_TX/SPI_SCK/32K_XTAL_OUT	General purpose bi-directional I/O ports INT4 signal input PWM0 signal output ADC analog channel input Touch key analog channel input I²C transfer port UART2 transfer port SPI_SCK port 32K External crystal oscillator output	General purpose bi-directional I/O ports
15	4	P0.7/INT4_4/PWM3_C/ADC7/TK7/I2C_SDA/I2C_SCL/ UART2_RX/UART2_TX/DAC/SPI_SDO	General purpose bi-directional I/O ports INT4 signal input PWM3 signal output ADC analog channel input Touch key analog channel input I²C transfer port UART2 transfer port DAC Analog output port	General purpose bi-directional I/O ports

			SPI_SDO port	
16	5	P0.6/INT4_3/PWM2_C/ADC6/TK6/I2C_SDA/I2C_SCL/ UART2_RX/UART2_TX/SPI_SDI	General purpose bi-directional I/O ports INT4 signal input PWM2 signal output ADC analog channel input Touch key analog channel input I^2C transfer port UART2 transfer port SPI_SDI port	General purpose bi-directional I/O ports

Note: The method for setting the signal pin multiplexing function is detailed in Table 15-2-5

6 Central Processing Unit (CPU)

6.1 CPU Introduction

The CA51F152S3 series chips use a single-cycle 8051 CPU, which is fully compatible with the original MCS-51 instruction set. the CPU uses a pipelined architecture, and typically, the single-cycle 8051 CPU runs 10 times faster than a standard 8051 processor.

The CPU has the following characteristics：

- 1T 8051 CPU
- Compatible with 8051 instruction set, see instruction set appendix
- Dual DPTR for fast data migration

6.2 Register Description

Program counter PC

The program counter PC register is 16 bits, which is a special register used to control the order of instruction execution, and it has no register address. After the microcontroller is powered on or reset, the PC value is 0 and the microcontroller starts executing the program from zero address.

Accumulator ACC

Accumulator ACC is a commonly used special register. The instruction system uses A as the accumulator helper, and it is often used to store the operands of arithmetic or logical operations and the results of the operations.

General purpose register B

The MUL AB instruction multiplies ACC with an 8-bit unsigned number in B. The low byte of the resulting 16-bit product is stored in A and the high byte is stored in B. The DIV AB instruction divides B by A. The integer quotient is stored in A and the remainder is stored in B. Register B can also be used as a general-purpose temporary storage register.

Stack pointer SP

The stack pointer SP is an 8-bit dedicated register. It indicates the location of the top of the stack in the internal RAM block. After system reset, SP initializes bit 07H, making the stack in fact start from cell 08H. Considering that cells 08H~1FH belong to working register group 1~3 respectively, if these areas are used in programming, it is better to change SP to 80H or larger.

Data pointer DPTR

The data pointers DPTR0/DPTR1 are two 16-bit dedicated registers, their high byte registers are represented by DP0H/DP1H and low byte registers are represented by DP0L/DP1L, which can be used optionally via DPS (PSW.1). each DPTR can be handled either as one 16-bit register or as two independent 8-bit registers DP0H/DP1H and DP0L/DP1L.

Status Register PSW

Status register PSW is the status register of the CPU. When the CPU does arithmetic operations or logical operations, the corresponding PSW status bits will change

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 Status Register 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	Bit by bit flag 0：No rounding or borrowing occurs in arithmetic or logical operations 1：Arithmetic or logical operations with rounding or debit occurring
6	AC	Auxiliary feed flag bit 0：No auxiliary rounding or borrowing occurs in arithmetic or logical operations 1：Arithmetic or logical operations in which auxiliary rounding or borrowing occurs
5	F0	F0 flag bit User-defined flag bits
4~3	RS	R0~R7 register page selection bits 00：Page 0 (mapped to 00H-07H) 01：Page 1 (mapped to 08H-0FH) 10：Page 2 (mapped to 10H-17H) 11：Page 3 (mapped to 18H-1FH)
2	OV	Overflow flag bit 0：No overflow occurs 1：There is an overflow occurring
1	DPS	DPTR selection register, 0 to select DPTR0, 1 to select DPTR1
0	P	Parity check bits 0：The accumulator A value of 1 is an even number of bits 1：The number of bits with a totalizer A value of 1 is 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	Register SPMAX is used to record the maximum value of SP, the user can check this register in the application to determine whether the stack is at risk of overflow

7 Memory Systems

7.1 Random data memory (RAM)

The CA51F152S3 series chips provide 256 bytes of internal RAM and 512 bytes of external RAM with the following memory address assignments：

- The low 128 bytes of internal RAM (address: 00H ~ 7FH) can be addressed directly or indirectly.
- The high 128 bytes of internal RAM (address: 80H ~ FFH) can only be indirectly addressed.
- External 512 bytes of external RAM (addresses: 0000H~01FFH) can be indirectly addressed through MOVX instructions.

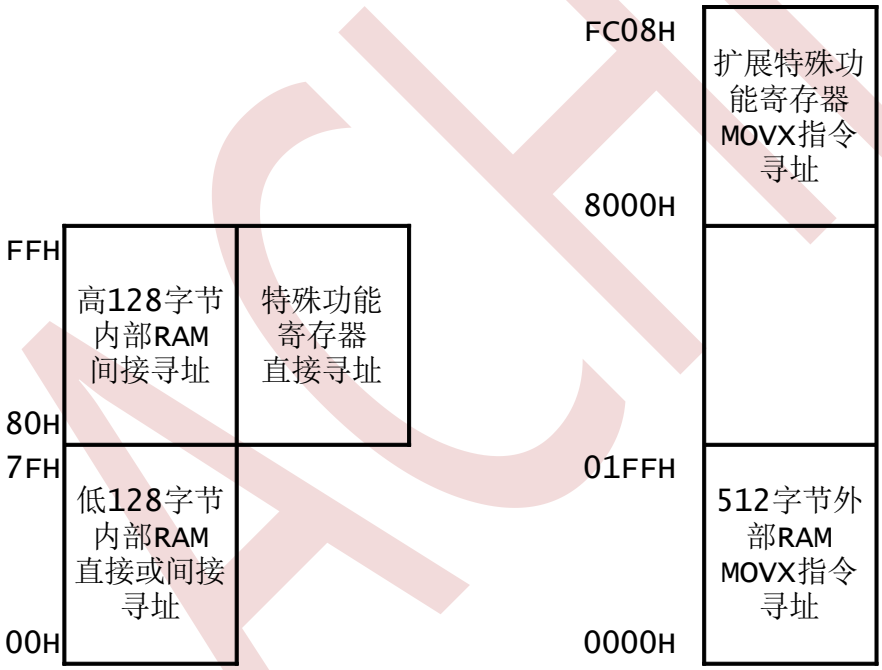


Figure 7-1-1 RAM Organizational Chart

7.2 Special Function Register (SFR)

CA51F152S3 series chips provide SFR distribution compatible with traditional 8051, SFR and high 128 bytes of internal RAM share the address 80H ~ FFH, can only be directly addressed, SFR mapping as shown in Table 7-2-1.

Table 7-2-1 Special Function Register (SFR) Mapping Table

	Bit-address able	Non-bit addressable						
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
F8H	TKST	TKCFG1	TKCFG2	TKCFG3	TKDL	TKDH	TKPULL	TKCFG4
F0H	B	CBYTE	TKPULLTRIM	SPMAX	TKIE	-	-	-
E8H	LVDCON	-	-	-	-	-	-	-
E0H	ACC	-	-	-	-	-	-	-
D8H	UDCKS1	-	-	-	-	-	-	-
D0H	PSW	EP0CON	EP1CON	EP2CON	EPIF	TMCON	TMSNU	SCLSEL
C8H	CKCON	CKDIV	IHCFG	TKCCFG	-	-	-	-
C0H	I2CCON	I2CADR	I2CADM	I2CCCR	I2CDAT	I2CSTA	I2CFLG	SDASEL
B8H	IP	S2CON	S2BUF	S2RELL	S2RELH	UDCKS2	DACON	-
B0H	-	-	-	-	-	-	-	-
A8H	IE	SPCON	SPDAT	SPSTA	-	-	-	-
A0H	-	WDFLG	WDVTHL	WDVTHH	T2CON	T2CRH	T2CRL	WDCON
98H	S1CON	S1BUF	PWMDUTH	PWMDUTL	-	-	-	INDEX
90H	P1	PWMEN	PWMIF	-	PWMCON	PWMCKD	PWMDIVL	PWMDIVH
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 CA51F152S3 series chips add extended special function registers to the external RAM address space, and the extended special function register mapping is shown in Figure 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	-	-
8010 H	-	-	-	-	-	-	-	-
8018H	-	-	-	-	-	-	-	-
8060H	ADCON	ADCFGL	ADCDL	ADCDH	ADCALL	ADCALH		
8068 H	S1RELL	S1RELH	-	-	-	-	-	-
8120H	P00C	P01C	P02C	P03C	P04C	P05C	P06C	P07C
8128H	P10C	P11C	P12C	P13C	P14C	P15C	-	-
8130H	-	-	-	-	-	-	-	-
8138H	-	-	-	-	-	-	-	-
FC00H	MECON	FSCMD	FSDAT	LOCK	PADRD	PTSL	PTSH	-
FC08H	-	-	-	-	-	-	-	-

7.3 Flash Memory

7.3.1 Function Introduction

Flash memory contains 16K bytes of Flash, which can be repeatedly erased. Flash memory is controlled by a specific set of registers, which users can use to perform operations such as read/write erasure and write protection

7.3.2 Flash memory organization

- Flash consists of several pages, which are the smallest unit for erasing operations, with each page size of 64 bytes.
- Flash write operations are performed on a page by page basis and must write 64 bytes at once. Single byte writing is not supported
- Flash can be divided into program area and data area by function, with a division unit of 128 bytes. The program area is used to store the user's program, while the data area is used to store some data that needs to be saved after power failure.

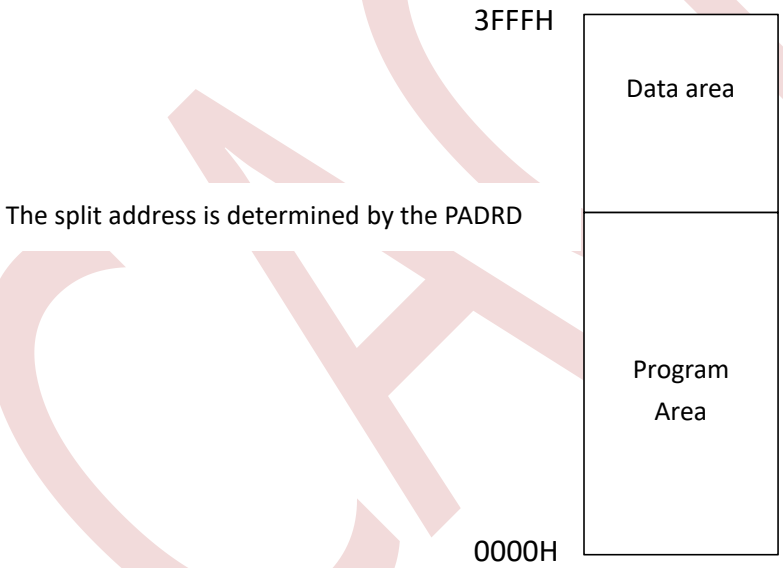


Figure 7-3-1 16KB Flash memory structure

7.3.3 Flash Register 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	IDLE/STOP mode Flash enter sleep mode control bit 0：IDLE/STOP mode, Flash is in normal operation mode 1：Flash enters sleep mode in IDLE/STOP mode Note: If DPSTB=1, when the chip enters IDLE/STOP mode, the Flash also enters sleep mode at the same time. The power consumption of the Flash in sleep mode is 50nA, and when the chip exits IDLE/STOP mode, the Flash also exits sleep mode at the same time.
5~1	-	-
0	BOOT	Set the program start space selection bit field after soft reset 0：Program runs from FLASH after soft reset 1：Program starts running 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	The information area access enable bit needs to be set to this position during access
6~4	-	-
3	CLRPL	Clear data from Flash latch
2~0	CMD	Command register 000: No operation 100: Flash sector erase 001: Read Flash data area 010: Write Flash data area 011: Erase one page of Flash data area 101: Read Flash Program Area 110: Writing Flash Program Area 111: Erase a page in the Flash program area *Remark.：*

		1. The CMD is automatically cleared after the erase command is executed. *2.The CMD remains unchanged after the read and write commands are written and then completed by reading and writing to 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]							
Initial Value	-	0	-	-	0	0	0	0
Bit Number	Bit Symbol	Description						
Write operation								
7~0	LOCK	28H：Unlocking of Flash programmable areas 29H：Unlocking the Flash program area 2AH：Unlocking the Flash data area AAH：Flash is locked, no write erase operation is possible						
Read operation								
7~4	-							
3	FLKF	Programmable zone unlock flag, 1 means unlocked						
2	PLKF	Program area unlock flag, 1 means unlocked						
1	DLKF	Data area unlock flag, 1 means 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 area 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 program area is: 0 ~ (PADRD×128 - 1), The address space of the data area is: (PADRD×128) ~ 3FFFH. *Remark：* *1. When PADRD=0, the whole Flash space is data space.* *2. The maximum value of PADRD is 90H respectively, and the setting value of PADRD cannot exceed the maximum value.*

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	The target address pointer register, when writing FSDAT operations, the data will be written to the PTS [5:0] related flash latch for temporary storage, and PTS [5:0] corresponds to the lower 6 bits of the actual write operation; When sending a write command, it is necessary to set the relevant page address PTS [13:6]. It is best to reconfigure the PTS address during each read, write, and erase operation. For continuous read operations, only the first address of the continuous read operation can be set.

7.3.4 Flash Control Routines

◆ **Flash divides program area and data area**

For example, the 16K Flash space is divided into the last 128 bytes for data space and the rest for program space, and the program is as follows.

```
---------------------------------------------------------------------------------
  PADRD = 127; //Program area space address is: 0~0x3F7F,data area space address is: 0x3F80~0x3FFF
---------------------------------------------------------------------------------
```

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

◆ **Data space page erasure**

For example, to erase data space page n, the program is as follows:

```
---------------------------------------------------------------------------------
  unsigned int address;
  address = 0x40*n;
  FSCMD = 0;        // Set CMD to 0
  LOCK = 0x2A;      // Data space unlocking
  FSCMD = 8;        // Set erase latch
  PTSH = (unsigned char)(address >>8); // Fill in high-order address
  PTSL = (unsigned char)(address);     // Fill in low order address
  FSCMD = 3;        // Set data area erase command
  LOCK = 0xAA;      // FLASH locking
---------------------------------------------------------------------------------
```

Note: Sector serial number n=0, 1, 2 .....

◆ **Writing data to the data space page**

For example, to write data 0xAA to a data space address of (0x40 * n), the program is as follows:

```
---------------------------------------------------------------------------------
  unsigned char i;
  unsigned int address;
  address = 0x40*n;
  FSCMD = 0;        // Set CMD to 0
  LOCK = 0x2A;      // Data space unlocking
  PTSH = 0;         //Set the starting address of the page latch
  PTSL = 0;         // Set the starting address of the page latch
  FSCMD = 8;        // Set erase latch
  for(i=0;i<64;i++)
  {
      FSDAT = 0xAA;           // Continuously write data for 1 page
  }
  PTSH = (unsigned char)(address >>8);     // Set the data first address 8 bits higher
  PTSL = (unsigned char)(address);         // Set data first address low 8 bits
  FSCMD = 2;        // Set Write Command
  LOCK = 0xAA;      // FLASH locking
---------------------------------------------------------------------------------
```

Note:

1．Page number n=0, 1, 2.

2．When writing data continuously, only the first address needs to be set. After each FSDAT write, the data pointer register PTS will automatically accumulate.

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

4．Data writing can only be done on a page by page basis, and 64 bytes must be written each time.

◆ **Data space readout data**

For example, reading data from the data space address n~(n+63) to the pointer dataBuf, the program is as follows:

```
---------------------------------------------------------------------------------
  unsigned int i,dataBuf[64];
  FSCMD =0;
  PTSH = (unsigned char)(n>>8);                    // Fill in high-order address
  PTSL = (unsigned char)n;                         // Fill in low order address
  FSCMD = 1;                                       // Execute read operation
  for(i = 0; i < Length; i++)
  {
        dataBuf[i]= FSDAT;
  }
  FSCMD = 0;
  LOCK = 0xAA;                                     // FLASH locking
---------------------------------------------------------------------------------
```

Note:

1．When continuously reading data, only the first address needs to be set. After each read of FSDAT, the data pointer register PTS will automatically accumulate.

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

3．Data reading does not need to be on a page basis and can read any number of bytes continuously.

◆ **Program space sector erase**

For example, to erase the program space sector n, the program is as follows:

```
---------------------------------------------------------------------------------
   unsigned int address;
   address = 0x40*n;
  FSCMD = 0;   // Set CMD to 0
  LOCK = 0x29; // Program space unlocking
  FSCMD = 8;    // Set erase latch
  PTSH = (unsigned char)((address)>>8); // Set the high bit address of the sector
  PTSL = (unsigned char)(address);         // Set the low bit address of the sector
  FSCMD = 7;      // Set erase command
  LOCK = 0xAA; // FLASH locking
---------------------------------------------------------------------------------
```

Note: Sector number n=0, 1, 2.

◆ **Writing data into program space**

For example, writing data 0xAA to program space address 0x40 * n, the program is as follows:

```
unsigned char i;
unsigned int address;
address = 0x40*n;
FSCMD = 0;          // Set CMD to 0
LOCK = 0x29;        // Program space unlocking
FSCMD = 8;          // Set erase latch
PTSH = 0;           // Set the starting address of the page latch
PTSL = 0;           // Set the starting address of the page latch
for(i=0;i<64;i++)
{
    FSDAT = 0xAA;               // Continuously write 1 page of data
}
PTSH = (unsigned char)(address >>8); // Set the top 8 bits of the data header address
PTSL = (unsigned char)(address);     // Set the low 8 bits of the data header address
FSCMD = 6;                           // Set write command
LOCK = 0xAA;                         // FLASH locking
```

Note:

1. Page number n=0, 1, 2.

2. When writing data continuously, only the first address needs to be set. After each FSDAT write, the data pointer register PTS will automatically accumulate.

3. Data writing can only be done on a page by page basis, and 64 bytes must be written each time.

◆ **Reading data from program space**

For example, reading data from program space address n~(n+63) to pointer dataBuf, the program is as follows:

```
unsigned char i, dataBuf[64];
FSCMD = 0; // Set CMD to 0
PTSH = (unsigned char)(n>>8); // Set the top 8 bits of the data header address
PTSL = (unsigned char)n;          // Set the low 8 bits of the data header address
FSCMD = 5;    // Set read command
for(i=0;i<64;i++)
{
    dataBuf[i] = FSDAT ;     // Continuously writing data
}
FSCMD = 0;
LOCK = 0xAA;                 // FLASH locking
```

Note:

1. When continuously reading data, only the first address needs to be set. After each read of FSDAT, the data pointer register PTS will automatically accumulate.

2. Data reading does not need to be on a page basis and can read any number of bytes continuously.

7.4 External RAM mapped to program space

A 512 byte external RAM can be mapped to program space usage, with mapping addresses ranging from 8000H to 81FFH. The mapping diagram is shown in Figure 7-4-1. Users can download programs to external RAM space and execute jump instructions directly to the mapped program area when the program is running. Similarly, the value of BOOT (see register MECON for details) can also be set to 1, and then a soft reset can be performed. After the reset, the program starts executing from the external RAM space (with mapped addresses ranging from 0000H to 01FFH). The mapping program area is particularly convenient for implementing IAP and other functions.

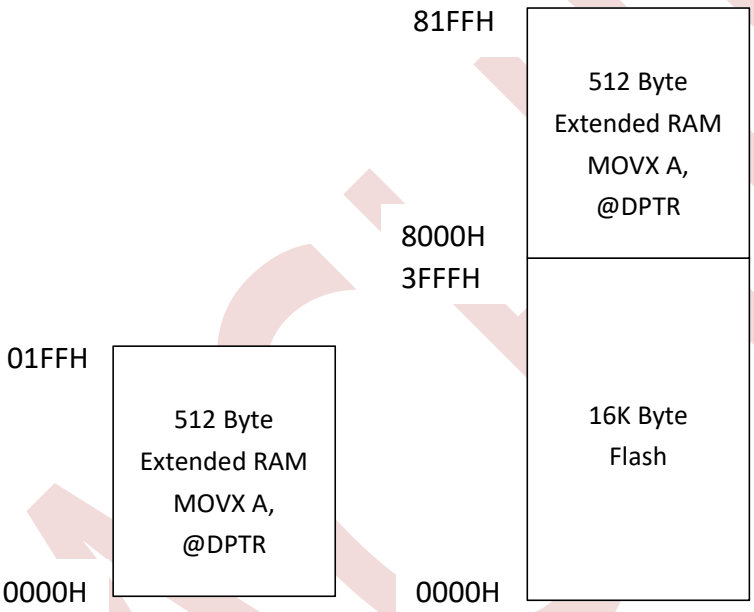


Figure 7-4-1 XRAM address mapping diagram

8 Interruption system

8.1 Function Introduction

CA51F152S3X series chip has an enhanced interrupt control system, there are 7 interrupt entry, each interrupt entry has a number of interrupt sources, each interrupt source has two 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

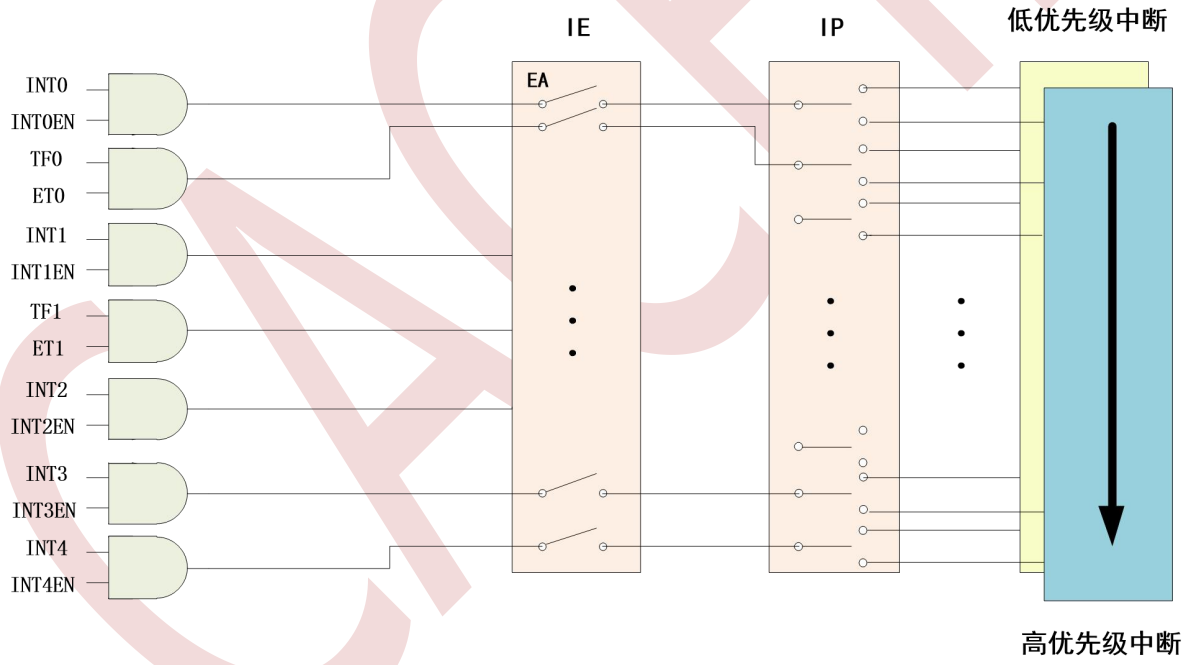


Table 8-2-1 Interrupt Logic Diagram

8.3 Interrupt vector table

Table 8-3-1 Interrupt vector table

Interruptions	Interrupt source	vector	Default Priority
INT0	INT0	03H	0
TF0	Timer 0	0BH	1
INT1	INT1	13H	2
TF1	Timer 1	1BH	3
INT2	External interrupt 2/UART1/ADC interrupt/PWM interrupt/SPI interrupt	23H	4
INT3	External interrupt 3/Timer 2/Touch interrupt/TMC interrupt	2BH	5
INT4	External interrupt 4/UART2/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：Shutdown 1：open
6	INT4EN	Interrupt 4 enable control bit (used for external interrupt 4/UART2/WDT interrupt/I2C interrupt/LVD interrupt) 0: Close 1: Open
5	INT3EN	Interrupt 3 enables control bit (Interrupt 3 is used for timer 2/TK/TMC/external interrupt 3) 0: Close 1: Open
4	INT2EN	Interrupt 2 enable control bit (Interrupt 2 is used for UART1/ADC/PWM/SPI/external interrupt 2) 0: Close 1: Open
3	ET1	Timer 1 interrupt enable control bit 0: Close 1: Open
2	EX1	Interrupt 1 enable control bit (Interrupt 1 is used for external interrupt 1) 0: Close

		1: Open
1	ET0	Timer 0 interrupt enable control bit 0: Close 1: Open
0	EX0	Interrupt 0 enables control bit (Interrupt 0 is used for external interrupt 0) 0: Close 1: Open

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

Table 8-4-2 Register IP

B8H	7	6	5	4	3	2	1	0
IP	-	PX4	PX3	PX2	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

Bit Number	Bit Symbol	Description
7	-	-
6	PX4	Interrupt INT4 priority control bit 0: Low priority 1: High priority
5	PX3	Interrupt INT3 priority control bit 0: Low priority 1: High priority
4	PX2	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

In addition to the standard 8051's INT0 and INT1, the system also extends three interrupt entry points INT2 to INT4 as external interrupts. Each external interrupt can be used to wake up in STOP mode. EPIF is an external interrupt status register from INT2 to INT4. The configuration registers corresponding to INT2~INT4 are EP0CON~EP2CON.

Interrupt input pin selection: INT0 (P0.0), INT1 (P0.1), INT2 (P0.2), INT3 (P0.3), INT4 can select any other I/O pin except for P0.0~P0.3 as the interrupt trigger source input pin. INT0~INT1 can choose to trigger interrupts along the rising or falling edge, while INT2~INT4 can choose to trigger interrupts along the rising, falling, or double edges.

Note:

1. *INT0 and INT1 can be triggered by either the rising or falling edge, with the selection bits IT0 and IT1 respectively. Please refer to the description of register TCON for details.*
2. *In STOP mode, if the main clock is turned off, INT2~INT4 cannot wake up the CPU. For more precautions, please refer to the "STOP mode" chapter.*

8.5.2 External Interrupt Register

Table 8-5-1 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 bit, write 1 to reset to zero
1	EPIF1	External interrupt 3 interrupt flag bit, write 1 to reset to zero
0	EPIF0	External interrupt 2 interrupt flag, write 1 to reset to zero

Table 8-5-2 Register EPCON

D1H	7	6	5	4	3	2	1	0
EP0CON	EPIE0	EPPL0		-	-	-	-	-
R/W	R/W	R/W	RW	-	-	-	-	-
Initial Value	0	0	0	-	-	-	-	-
D2H	7	6	5	4	3	2	1	0
EP1CON	EPIE1	EPPL1-		-	-	-	-	-
R/W	R/W	R/W	R/W	-	-	-	-	-
Initial Value	0	0	0	-	-	-	-	-
D3H	7	6	5	4	3	2	1	0
EP2CON	EPIE2	EPPL2		EPPS2				
R/W	R/W	R/W	R/W	R/W				
Initial Value	0	0	0	0	0	0	0	0

Note: "n" in the table below represents 0/1/2

Bit Number	Bit Symbol	Description
7	EPIEn	External interrupt enable bit 0: Close 1: Open *Note: n=0/1/2 corresponds to external interrupts 2/3/4, respectively.*
6~5	EPPLn	External interrupt trigger edge selection bit 00: Rising edge 01: Descending edge 1x: Double edge *Note: n=0/1/2 corresponds to external interrupts 2/3/4, respectively.* *X=0/1*
4~0	EPPS2	External interrupt 4 selection interrupt input port 5'h00：Indicates selectionP1.0； 5'h01：Indicates selectionP1.1； 5'h02：Indicates selectionP1.2； 5'h03：Indicates selectionP1.3； 5'h04：Indicates selectionP1.4； 5'h05：Indicates selectionP1.5； 5'h06：Indicates selectionP0.4； 5'h07：Indicates selectionP0.5； 5'h08：Indicates selectionP0.6； 5'h09：Indicates selectionP0.7；

8.5.3 External Interrupt Control Routines

◆ **External interrupt 0 control routine**

For example, to enable external interrupt 0, the procedure is as follows:

```
---------------------------------------------------------------------------------
void INT0_init(void)
{
    P00F = 1;       // The interrupt pin for external interrupt 0 is P0.0, set P0.0 as the input function
    EX0 = 1;        // INT0 interrupt enable
    IE0 = 1;        // External interrupt 0 enable
    IT0 = 1;        // Set to falling edge interrupt
    EA = 1;         // Total interrupt enable
}
void INT0_ISR (void) interrupt 0
{
    // External interrupt 0 interrupt service program
}
---------------------------------------------------------------------------------
```

◆ **External interrupt 4 control routine**

Taking external interrupt 4 as an example, set P1.0 as the input pin for external interrupt 4 and enable external interrupt 4. The program is as follows:

```
---------------------------------------------------------------------------------
void INT4_init(void)
{
    P10F = 1;                       // Set P1.0 as input pin
    EP2CON = (1<<7) | (0<<5) | 0;   // Set to trigger the rising edge and set the interrupt pin index number, where 0
                                    //corresponds to P1.0. If set to below
                                    // The falling edge trigger is set to EP2CON=(1<<7) | (1<<5) | 0;
    INT4EN = 1;                     // INT4 interrupt enable
    EA = 1;                         // Total interrupt enable
}
void INT4_ISR (void) interrupt 6
{
    if(EPIF & 0x04)                 // Determine external interrupt 4 interrupt flag
    {
        EPIF = 0x04;                // Interrupt flag write 1 clear 0
        // External Interrupt 4 Interrupt Service Program
        ......
    }
}
---------------------------------------------------------------------------------
```

9 Clock System

9.1 Introduction of clock system

The CA51F152S3X family of chips supports the following clock sources in total：

- Built-in 16MHz RC oscillator
- Built-in 128KHz RC oscillator
- Built-in 24MHz RC oscillator
- Support external 32.768KHz crystal oscillator

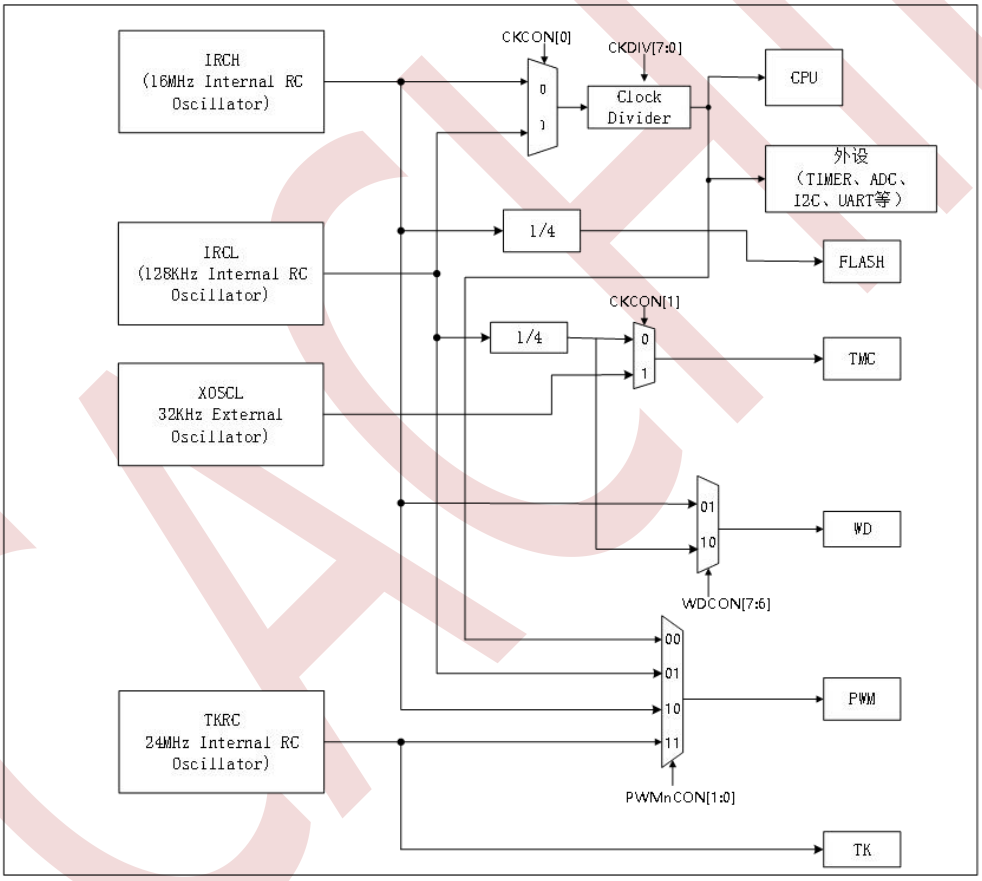


Figure 9-1-1 Clock source diagram

Users can control the clock sources independently. They can disable or enable any of the clock sources in order to manage the power consumption flexibly.

All the clock sources can be set as system alarm clock and assigned to various peripherals as their clock sources. For more information you may refer to the Peripherals part.

Note: The external 32.768KHz crystal oscillator (XOSCL) can only be used as a clock source for TMC timers.

9.1.1 Clock-specific Name Definition

Symbol	Description
IRCH	Built-in 16MHz RC oscillator
IRCL	Built-in 128KHz RC oscillator
TKRC	Built-in 24MHz RC oscillator
XOSCL	External 32.768MHz crystal oscillator

9.1.2 16 MHz Internal RC Oscillator(IRCH)

IRCH is the default system clock after the chip is powered on, and can be turned on or off by the IHCKE bit of register CKCON.The chip is shipped from IRCH frequency correction for 16MHz@5V/25° C with a clock accuracy of ±2%.

9.1.2 24 MHz Internal RC Oscillator(TKRC)

TKRC can be turned on or off through the TKCKE bit of register CKCON, serving as a dedicated clock for touch modules.

9.1.3 Built-in 128 KHz RC Oscillator (IRCL)

IRCL can be turned on or off by the ILCKE bit in register CKCON. IRCL is set to the system clock to achieve low system power consumption.

9.1.4 External 32.768KHz Crystal Resonator (XOSCL)

XOSCL is mainly used as a clock source for TMC, used for real-time timing, to achieve the clock function of the product. XOSCL is opened or closed through the XLCKE bit of register CKCON. It should be noted that XOSCL has a relatively long oscillation time, which takes about 1 second to reach stability. When applying, it is necessary to wait for the XOSCL clock to stabilize before it can be used.

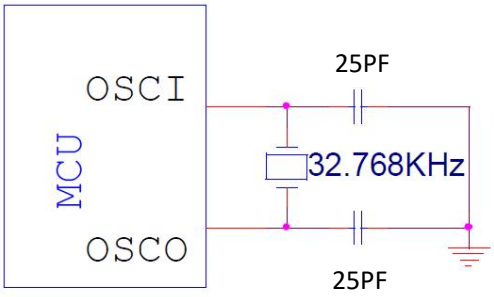


Figure 9-1-4-1 Typical circuit diagram of XOSCL

Important Reminder:

1. *During hardware design, the load capacitor of the crystal oscillator must be connected to the chip ground, and the compensation capacitor of the crystal oscillator should be as close as possible to the VSS pin of the chip.*

32.768KHz quartz crystal oscillator requires the use of crystal oscillator specifications with a diameter of 3mm x 8mm.

2. *The above circuit and component parameters are for reference only. The parameters may need to be modified when using crystal oscillators from different manufacturers in the circuit.*

9.2 Clock Control Register Description

Table 9-2-3-1 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	0	0
Bit Number	Bit Symbol	Description						
7~0	IHCFG	IRCH frequency adjustment register						

Table 9-2-3-2 Register TKCCFG

CBH	7	6	5	4	3	2	1	0
TKCCFG	TKCCFG[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
Bit Number	Bit Symbol	Description						
7~0	TKCCFG	TKRC frequency adjustment register						

9.3 System Clock

The system clock control is completed by registers CKCON and CKDIV. Through these register groups, it is possible to separately set the switches of each clock source, switch the system clock, and perform frequency division operations.

There are two clock options for the system clock: IRCH and IRCL. After power on, the default system clock is IRCH, and the CKDIV value is 1, which means that the system clock defaults to the binary frequency of IRCH when powered on. If the CPU needs to run at a higher frequency, the software can set the CKDIV to 0.

9.3.1 System clock structure diagram

The system clock structure diagram is shown in Figure 9-3-1.

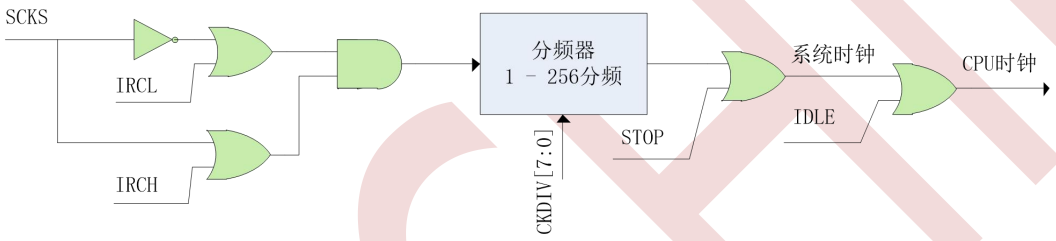


Figure 9-3-1 System clock structure diagram

9.3.2 System Clock Control Register Description

Table 9-3-2-1 Register CKCON

C8H	7	6	5	4	3	2	1	0
CKCON	IHCKE	ILCKE	TKCKE	XLCKE	-	-	TMCS	SCKS
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
Initial Value	0	0	0	0	-	-	0	0

Bit Number	Bit Symbol	Description
7	IHCKE	IRCH enable control bit 0：Shutdown 1：Open Note: When this bit is 1, the clock module is turned on, but when this bit is 0, if the system or other modules choose the clock source, the clock will still be turned on.
6	ILCKE	IRCL enable control bit 0：Shutdown 1：Open

		Note: When this bit is 1, the clock module is turned on, but when this bit is 0, if the system or other modules choose the clock source, the clock will still be turned on.
5	TKCKE	TK clock enable control bit 0：Shutdown 1：Open
4	XLCKE	XOSCL clock enable control bit 0：XOSCL clock off 1：XOSCL clock on
3~2	-	-
1	TMCS	TMC counting clock selection 0：Select IRCL 1：Select XOSCL
0	SCKS	System clock selection bit 0：Select IRCH 1：Select IRCL

Table 9-3-2-2 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

Note: After powering on, the system clock defaults to 2 divisions. If you need to modify it to a 16MHz clock, you need to set CKDIV to 8'h00;

9.3.3 System clock control methods and routines

- **Set the system clock to IRCH**

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
    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;                                 // Turn on the IRCL clock
    Delay_ms(1);                                    // Delay 1ms after enabling IRCL and wait for IRCL to stabilize
    CKCON = (CKCON&0xFE) | CKSEL_IRCL;      // Set the system clock to IRCL
}
-------------------------------------------------------------------------------
```

10 Power supply and reset system

10.1 Power supply system

Connect a 2.3V-5.5V power supply between the VDD and VSS pins of the CA51F152S3 series chip, which can directly supply power to the internal digital and analog systems of the chip. It should be noted that under different power supply voltage conditions, the maximum frequency and power consumption supported by the chip for operation are not the same. Please refer to the Electrical Characteristics section for details.

The chip is also designed with a BANDGAP reference voltage as the reference voltage source for ADC internal 1.5V reference voltage, LVD voltage, etc. After calibration at the factory, this reference voltage source has an accuracy of ± 50mV.

Figure 10-1-1 shows a typical circuit diagram of the chip power supply.

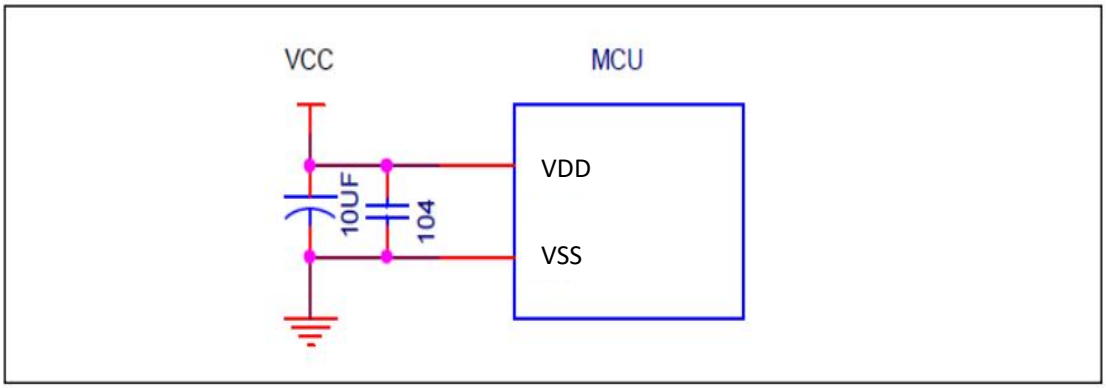


Figure 10-1-1 Typical circuit diagram of chip power supply

Important Reminder:

1. *In the above circuits, the filtering capacitors 10uF and 104 are standard configurations for chip power supply circuits and cannot be omitted. This capacitor must be close to the chip power supply Place your feet, otherwise it may cause abnormal chip operation.*
2. *The above circuit and component parameters are for reference only, and may need to be modified according to the peripheral working environment and different voltage power supply parameters*

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 has been calibrated at the factory with an accuracy of ± 30mV. The calibration value is automatically loaded when powered on and cannot be changed by the user.*
3	VREFS	Drive selection for reference voltage: 0:0.8uA driving capability, default 1: Operational amplifier drive
2-0		

10.2 Reset system

The CA51F152S3X series chips have multiple internal and external reset sources, as shown in Figure 10-2-1.

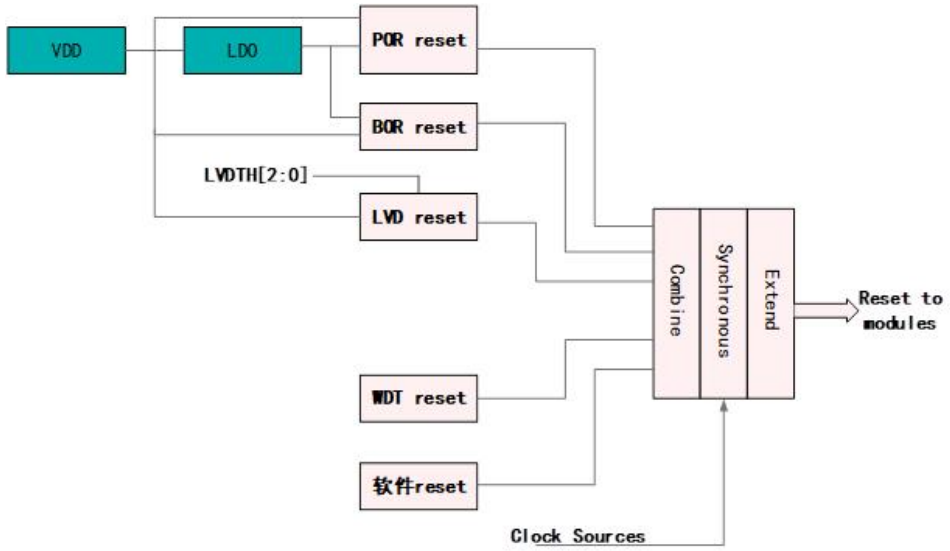
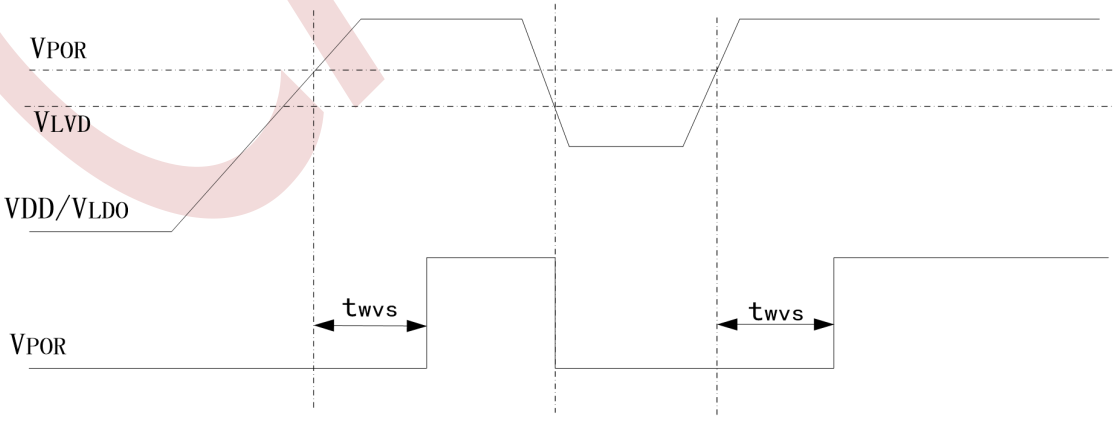


图 10-2-1 Reset system architecture diagram

- **Power-On Reset (POR)**

The system shows a gradually increasing curve when powered on, and it takes some time to reach the normal working voltage. Power on reset is based on the power supply voltage VDD. When the voltage is below the detection threshold, the power on reset signal is effective.

The power-on reset circuit ensures that the chip is in a reset state during the power-up process and that the chip can start operating from a known stable state after power-up. The power-on reset 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.



twvs：等待电压稳定时间

Figure 10-2-2 Power-on reset circuit example and power-on process

- **Power-Off Reset (BOR)**

By using power-off reset, a warning signal for power drop (such as interference or load changes) can be provided to the chip. Once the power supply voltage VDD drops to a certain threshold, the chip should be reset in a timely manner to avoid abnormal system operation or program execution errors.

- **Low voltage reset**

Low voltage detection (LVD) allows continuous monitoring of the supply voltage VDD in various operating modes. a reset signal can be generated when VDD falls below the domain voltage set by LVD for more than 20us (provided that LVD is set to reset mode).

- **Watchdog Reset**

The watchdog timer is responsible for monitoring the execution of instructions by the processor and, with the proper configuration, can generate a reset signal if the watchdog timer is not refreshed within a specific time period. After a power-on reset, the watchdog timer is off and then configured to be on when the user needs it.

- **Soft Reset**

The chip can perform a soft reset under program control. By writing 1 to the SWRST bit in the PCON register, the CPU can issue a reset command.

Power-on and power-down 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.

Note: After low voltage reset, watchdog reset, and soft reset occur, the values of IHCFG and TKCCFG will be reset to 0, causing a change in the clock frequency of IRCH and TKRC. In applications, it is required that the software must reload the calibration values of IRCH and TKRC after the above three resets occur. The reference program is as follows:

```
unsigned char ReadIHCFG(void)
{
    unsigned char D1,D2;
    FSCMD = 0x80;
    PTSH = 0x00;
    PTSL = 0x08;
    FSCMD = 0x81;
    D1 = FSDAT;
    D2 = FSDAT;
    FSCMD = 0;
    if(D1 == 'H')
    {
        return D2;
    }
    return IHCFG;
}
```

```
unsigned char ReadTKCCFG(void)
{
    unsigned char D1,D2;
    FSCMD = 0x80;
    PTSH = 0x00;
    PTSL = 0x06;
    FSCMD = 0x81;
    D1 = FSDAT;
    D2 = FSDAT;
    FSCMD = 0;
    if(D1 == 'F')
    {
        return D2;
    }
    return TKCCFG;
}

Void main(void)
{
    IHCFG = ReadIHCFG();
    TKCCFG = ReadTKCCFG();

    …….
}
```

11 Power Management

The CA51F152S3 series chips have three different low-power modes: IDLE mode, STOP mode, and low-speed operation mode. The system power consumption is less than 30uA in IDLE mode, less than 7uA in STOP mode, and less than 55uA in low-speed operation.

11.1 IDLE mode

In IDLE mode, the CPU will stop working. Before entering IDLE mode, all other clock sources except the master clock can be selected to be turned off as needed to save power. Likewise, before entering IDLE mode, certain peripherals of the chip can be set to be switched on and off as needed. The open peripherals can still work normally in IDLE state.

Before setting into IDLE mode, you need to check the registers IDLST (IDLSTH and IDLSTL), if all the bits are 0, the CPU will enter IDLE mode normally after setting into IDLE mode. If the bits of IDLST are not all 0, even if there is an operation to set into IDLE mode, the CPU will not enter IDLE mode, but continue to stay in normal operation mode. In this case, the user needs to finish the interrupt processing of the corresponding bit of IDLST before resetting the action to enter IDLE mode.

All reset events and any interrupt events will wake up the chip. After the interrupt wakes up the CPU, the chip will first restore the clock, then respond to the interrupt and enter the service program of the interrupt. After exiting the interrupt service program, the chip will execute the instruction following the reset IDLE instruction. When exiting the IDLE mode, the IDLE bit will be automatically cleared to zero.

Note that two nop instructions need to follow immediately after the reset IDLE instruction to prevent program errors.

11.2 STOP mode

STOP mode is a deeper low-power mode than IDLE. STOP mode stops all clocks (including the master clock) and clock generation circuits. If the WDT and TMC are on, the clock modules they use will be in operation and the WDT and TMC can be selectively turned off 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.

STOP mode can be woken up by external interrupt, LVD interrupt or reset, TMC interrupt, WDT interrupt or reset, clock monitor interrupt and touch interrupt. If it is wake-up by interrupt, then after waking up the MCU, the chip will first restore the clock, then respond to the interrupt and enter the service program of the interrupt. After exiting the interrupt service program, the chip will execute the instruction after the set STOP instruction. When exiting STOP mode, the STOP bit will be automatically cleared to zero.

For better wake-up of the chip, it is recommended to switch the system clock to the internal clock before entering STOP mode, because the external clock needs more time to wait for stability when waking up.

When entering STOP mode, the last clock edge will turn off the system clock, and then the chip enters STOP mode completely. It should be noted that the instruction to set STOP needs to be followed immediately

by three nop instructions to prevent program errors.

Remarks:
If there is an application scenario where an external interrupt INT2/INT3/INT4 is used to wake up the MCU, the system clock needs to be set to the IRCL clock before entering STOP, while keeping the IRCL clock on. Keeping the IRCL clock running in STOP mode results in an increase in power consumption compared to when all clocks are turned off. The power consumption of a randomly selected typical chip test result is less than 20 uA (for reference only). If there is no application scenario where external interrupts INT2/INT3/INT4 are used to wake up the MCU, all clocks can be turned off before entering STOP, and the static power consumption at this time is less than 7uA. The above operations can be found in the "STOP Mode Routine".

11.3 Low-speed operation mode

Since the power consumption of the chip is directly related to the operating speed, switching the master clock to run at a low clock speed can also significantly reduce power consumption. The current when the system is set to IRCL (frequency of 128KHz) is less than 55uA.

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 bit, 1 valid Set SWRST=1 to generate a soft reset, and clear 0 automatically after the reset is generated.
4~3	-	-
2	TSMODE	Online emulation mode flag bit, 1 means the chip is working in online emulation mode
1	STOP	STOP mode control bit, 1 valid When STOP=1 and STPST is 0, the chip enters STOP mode and automatically clears 0 after exiting STOP mode
0	IDLE	IDLE mode control bit, 1 valid When setting IDLE=1 and IDLST is 0, the chip enters IDLE mode and automatically clears 0 after exiting IDLE mode

Table 11-4-2 Registers 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/UART2/EPIF[2]	When in IDLE mode, the interrupt status of I2C/WDT/LVD/UART2/external interrupt 4
5	TKINT/TMINT/TIMER2/EPIF[1]	When in IDLE mode, touch the interrupt status of buttons/TMC/timer 2/external interrupt 3
4	UART1/ADC/PWM/SPI/EPIF[0]	When in IDLE mode, the interrupt status of UART1/PWM/SPI/ADC/external interrupt 2
3	TF1	When in IDLE mode, the interrupt state of timer 1
2	PIF[1]	When in IDLE mode, the interrupt status of external interrupt 1
1	TF0	When in IDLE mode, the interrupt state of timer 0
0	PIF[0]	When in IDLE mode, the interrupt state of external interrupt 0

Table 11-4-3 Registers 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	When in STOP mode, the interrupt status of WDT/LVD/I2C
5	TKWKF/TMWKF	When in STOP mode, touch the interrupt status of the button/TMC
4	EPWKF[2]	When in STOP mode, the interrupt status of external interrupt 4
3	EPWKF[1]	When in STOP mode, the interrupt status of external interrupt 3
2	EPWKF[0]	When in STOP mode, the interrupt status of external interrupt 2
1	PWKF[1]	When in STOP mode, the interrupt status of external interrupt 1
0	PWKF[0]	When in STOP mode, the interrupt state of external interrupt 0

11.5 Low power mode control routines

- **STOP Mode Routine**

The STOP mode procedure 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;
    unsigned char ck_bak;
    I2CCON = 0;             // Turn off the I2C function, because I2C is enabled by default, if I2C is not turned off
    MECON |= (1<<6);        // will not be able to turn off the IRCH clock
    ck_bak = CKCON & 0xFE;  // Set FLASH into deep sleep state
    #if 0
/*
Remarks:
If there is an application scenario where an external interrupt INT2/INT3/INT4 is used to wake up the MCU, the system clock needs to be set to the IRCL clock before entering STOP, while keeping the IRCL clock on. Keeping the IRCL clock running in STOP mode results in an increase in power consumption compared to when all clocks are turned off. The power consumption of a randomly selected typical chip test result is less than 20 uA (for reference only).*/
    CKCON = (CKCON&0x01)| ILCKE;          // IRCL clock enable
    Delay_ms(1);                          // Enable IRCL and delay for 1ms, wait for IRCL to stabilize
    CKCON = (CKCON&0xFE) | CKSEL_IRCL; // System clock switched to IRCL
    #else
/*
Remarks:
If there is no application scenario where external interrupts INT2/INT3/INT4 are used to wake up the MCU, all clocks can be turned off before entering STOP, and the static power consumption at this time is less than 7uA.*/
    CKCON = 0;                            // Turn off all clocks
    #endif

    IE_EA = EA;                           // Save global interrupt enable bit status
    EA = 0;
    PCON = (PCON&0x04)|0x02;              // Enter STOP mode
    _nop_();                              // Three nop instructions need to be followed immediately after the
                                          // STOP instruction to prevent program errors.
    _nop_();
    _nop_();
```

```
    EA = IE_EA;                     // Restore the original global interrupt switch state
    Sys_Clk_Set_IRCH();
    CKCON |= ck_bak;                // Restore the closed clock
}
```

Test conditions:

All clocks are turned off, all output pins are unloaded, all digital input pins are not floating, all peripherals are turned off, Flash enters deep sleep mode, and CPU enters STOP mode.

◆ **IDLE Mode Routine**

The IDLE mode procedure is as follows.

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

#define CKSEL_IRCH    0
#define CKSEL_IRCL    1

void Idle(void)
{
    unsigned char ck_bak;
    I2CCON = 0;         // Disable I2C module, because I2C is enabled by default, if I2C is not disabled, IRCH
                        // clock will not be disabled
    ck_bak = CKCON & 0xFE;              // Backup clock status
    CKCON = (CKCON&0x41) | ILCKE;       // Enable IRCL clock and turn off other clocks
    Delay_ms(1);                        // Enable IRCL and delay for 1ms, wait for IRCL to stabilize
    CKCON = (CKCON&0xFE) | CKSEL_IRCL; // System clock switched to IRCL
    MECON |= (1<<6);                    // Set FLASH to enter deep sleep mode
    while(IDLST&0x7F);                  // If there is an interrupt that is not responding, wait for the interrupt to be
                                        // responded to
    PCON = (PCON&0x04)|0x01;            // Enter IDLE mode
    _nop_();
    _nop_();
    Sys_Clk_Set_IRCH();
    CKCON |= ck_bak;                    // Restore the closed clock
}
```

Note:

Since the master clock is still open after entering IDLE, if the master clock is a high-speed clock before entering IDLE, the power consumption will still be very large after entering IDLE mode, so you need to switch the master clock to a low-speed clock before entering IDLE.

---

12 General purpose timer (timer 0, timer 1, timer 2)

12.1 Timer 0

12.1.1 Timer 0 Introduction

The timer or counter function is selected by the CT0 bit (TMOD[2]), with CT0=0 selected as a timer and CT0=1 selected as a counter. When used as a timer, the clock is 12 divisions of the system clock. When used as a counter, the clock is the input clock of T0. Since it takes 2 clock cycles to detect a change in the input edge of T0, the maximum input baud rate when used as a counter is 1/2 the internal system clock frequency. the T0 input signal is not limited in duty cycle, however, in order to fully recognize a 0 or 1 state, the signal needs to be held for at least 1 internal system clock cycle time. Timer 0 has 4 operating modes, which are selected by the T0M0, T0M1 bits (TMOD[1:0]).

- **Mode 0**

In this mode, timer 0 acts as a 13-bit timer/counter. TH0 holds the high 8 bits of the 13-bit timer/counter, TL0[4:0] holds the low 5 bits, and TL0[7:5] is invalid and should be ignored when read. When timer 0 overflows, interrupt flag bit TF0 (TCON[5]) will be set to 1. After the interrupt is responded, TF0 bit will be cleared to 0 automatically. When GATE0 (TCON[3]) = 0, the timer/counter is enabled to count by TR0 (TCON[4]) bit, when GATE0 = 1, the timer/counter is enabled by pin INT0 control, when INT0 is high count, and when INT0 is low, it stops counting.

- **Mode 1**

In this mode, Timer 0 functions as a 16-bit timer/counter and is otherwise identical to Mode 0.

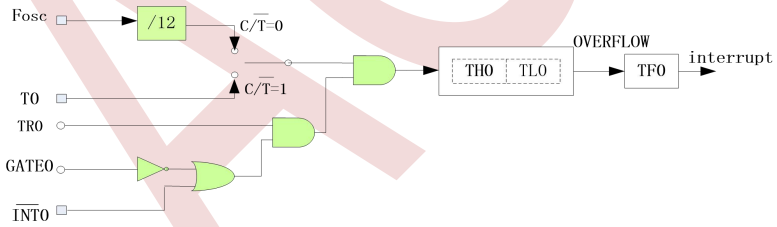
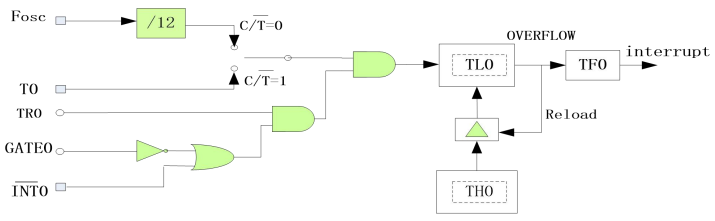


Figure 12-1-1-1 Mode 0 and 1 of Timer 0

- **Mode 2**

In this mode, timer 0 acts as an 8-bit auto-reload timer/counter, and only TL0 is auto-accumulated. When TL0 count overflows, not only the interrupt flag TF0 is generated, but also the initial value of count is automatically loaded from TH0 to TL0. Other setting methods are the same as modes 0 and 1.



- **Figure 12-1-1-2 Mode 2 of Timer 0**

- **Mode 3**

In this mode, TL0 and TH0 act as two independent 8-bit timers/counters. TL0 can act as timer or counter, while TH0 can only act as timer. Where TL0 occupies the control bits CT0, GATE0, TR0, TF0, INT0 of timer 0, while TH0 can only occupy the control bits TR1, TF1 of timer 1. Other control methods are the same as modes 0 and 1. When timer 0 works in mode 3, timer 1 and TH0 share control bit TR1, but timer 1 can only work in situations where interrupt generation is not required because TF1 is already occupied by TH0, for example, as a UART's baud rate generator.

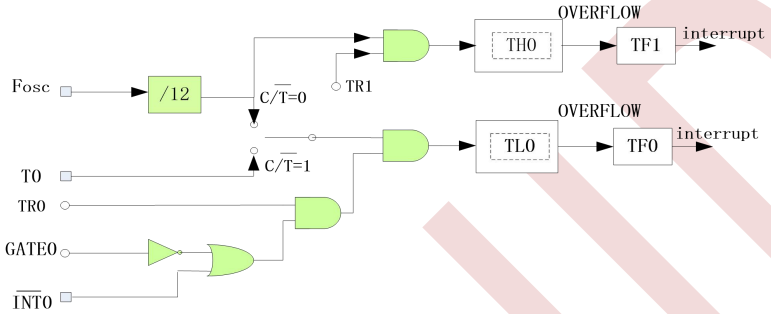


Figure 12-1-1-3 Mode 3 of Timer 0

12.1.2 Timer 0 Register Descriptions

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	TH0 overflow/timer 1 overflow flag bit of timer 0 mode 3, which is automatically cleared to 0 when the interrupt is responded.
6	TR1	Timer 1 operation control bit, 1 valid
5	TF0	Timer 0 overflow flag bit, automatically clear 0 when interrupt is responded.
4	TR0	Timer 0 run control bit, 1 valid
3	IE1	External interrupt 1 enable bit, 1 active
2	IT1	External interrupt 1 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 interrupt 0 enable bit, 1 active
0	IT0	External interrupt 0 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	Timer 1 gated control bit, 1 is valid. Timer 1 is switched by INT1 control when valid
6	CT1	Timer 1 counter/timer selection bit 0: timer, clocked at 12 divisions of the system clock 1: Counter, clocked as T1 input clock
5	T1M1	[T1M1,T1M0] is timer 1 mode selection bit 00: Mode 0, TL1 and TH1 form a 13-bit timer/counter 01: Mode 1, TL1 and TH1 form a 16-bit timer/counter 10: Mode 2, TL1 as 8-bit timer/counter, TH1 as auto reload register 11: Mode 3, this mode will lock TH1/TL1, equivalent to TR1=0
4	T1M0	
3	GATE0	Timer 0 gated control bit, 1 is valid. Timer 0 is switched by INT0 control when valid
2	CT0	Timer 0 counter / timer selection bit 0: Timer, clocked at 12 divisions of the system clock 1: Counter, clocked as T0 input clock
1	T0M1	[T0M1,T0M0] is timer 0 mode selection bit 00: Mode 0, TL0 and TH0 form a 13-bit timer/counter 01: Mode 1, TL0 and TH0 form a 16-bit timer/counter 10: Mode 2, TL0 as 8-bit timer/counter, TH0 as auto reload register 11: Mode 3, TL0 and TH0 as two completely independent 8-bit timer/counters
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	Timer 0 low byte of mode 0/1 count value, mode 2/3 count value

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	Timer 0 high byte of mode 0/1 count value, mode 2 reload value, mode 3 count value

12.2 Timer 1

12.2.1 Timer 1 Introduction

The timer or counter function is selected by the CT1 bit (TMOD[6]), with CT1=0 selected as a timer and CT1=1 selected as a counter. When used as a timer, the clock is 12 divisions of the system clock. When used as a counter, the clock is the input clock of T1. Since it takes 2 clock cycles to detect T1 input edge changes, the maximum input baud rate as a counter is 1/2 of the internal system clock frequency. the T1 input signal is not limited in duty cycle, however, in order to fully identify a 0 or 1 state, the signal needs to be held for at least 1 internal system clock cycle time. Timer 1 has 4 operating modes, which are selected by T1M0, T1M1 bits (TMOD[5:4])

- **Mode 0**

In this mode, timer 1 acts as a 13-bit timer/counter. TH1 holds the high 8 bits of the 13-bit timer/counter, TL1[4:0] holds the low 5 bits, and TL1[7:5] is invalid and should be ignored when read. When timer 1 overflows, interrupt flag bit TF1 (TCON[7]) will be set to 1. TF1 bit will be automatically cleared to 0 after the interrupt is responded. when GATE1 (TCON[7]) = 0, the timer/counter is enabled to count by TR1 (TCON[6]) bit, when GATE1 = 1, the timer/counter is enabled by pin INT1 control, when INT1 is high When GATE1=1, the timer/counter is enabled by pin INT1 and stops counting when INT1 is high.

- **Mode 1**

In this mode, timer 1 acts as a 16-bit timer/counter, TH1 holds the high 8 bits of the 16-bit timer/counter and TL1 holds the low 8 bits. When timer 1 overflows, interrupt flag bit TF1 (TCON[7]) will be set to 1. TF1 bit will be cleared to 0 automatically when the interrupt is responded. When GATE1=1, the timer/counter is enabled by pin INT1 and stops counting when INT1 is high.

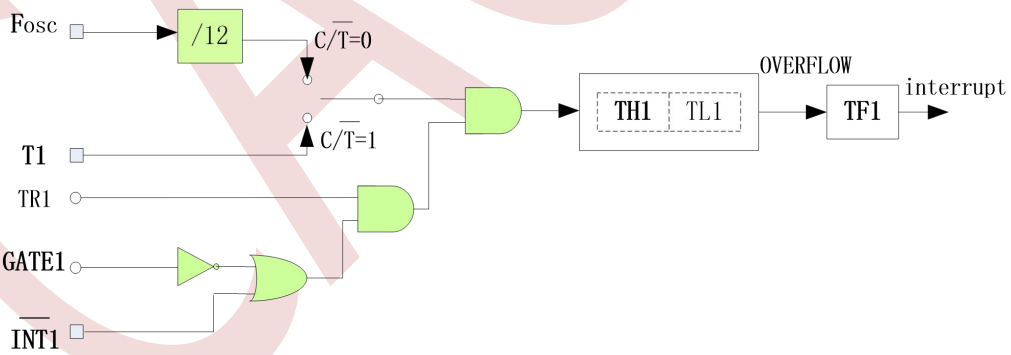


Figure 12-2-1 Mode 0 and 1 of Timer 1

- **Mode 2**

In this mode, Timer 1 serves as an 8-bit automatic overload timer/counter, and only TL1 automatically accumulates. When the TL1 count overflows, not only does it generate the interrupt flag TF1, but it also automatically loads the count Initial Value from TH1 to TL1. The other setting methods and modes 0 and 1 are the same.

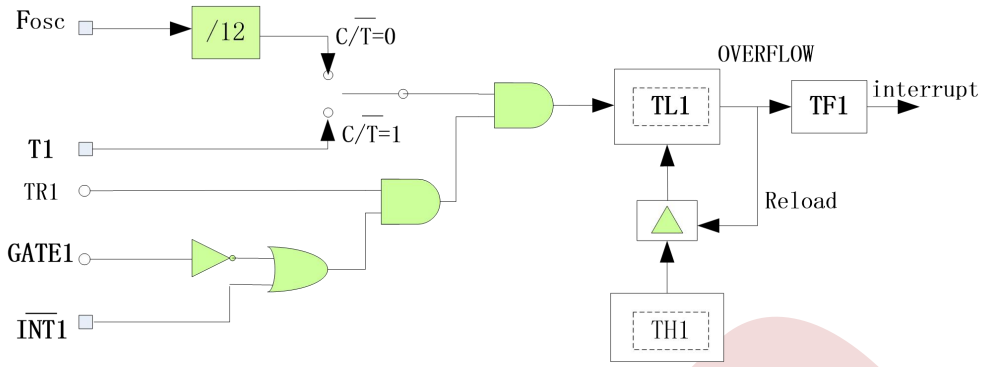


Figure 12-2-2 Mode 2 of Timer 1

- **Mode 3**
  In this mode, TH1 and TL1 will be locked, which is equivalent to TR1=0.

12.2.2 Timer 1 Register Description

The registers TCON and TMOD are shown in Table 12-1-2-1 and Table 12-1-2-2.

Table 12-2-2-1 Register TL1

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	Timer 1 low byte of mode 0/1 count value, mode 2/3 count value						

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	Timer 1 high byte of mode 0/1 count value, mode 2 reload value, mode 3 count value						

12.3 Timer 2

12.3.1 Timer2 Introduction

Timer 2 is a 16 bit timer, and the clock of Timer 2 is the system clock (note: unlike timers 0 and 1, the clock has not undergone a 12 division). Set different working modes through T2M.

When T2M=0, timer 2 operates in timer mode, and T2CRH and T2CRL store the timer set values. When Timer 2 is enabled, the 16 bit counter automatically accumulates. When the counter value reaches the set value (T2CRH, T2CRL stored values), a timed interrupt is generated, and the 16 bit counter is reset to zero. The F2 interrupt flag is cleared by writing 1 to TF.

When T2M=1, timer 2 operates in capture mode. When the trigger edge of pin T2CP occurs, the count value of timer 2 is locked to T2CRH and T2CRL. The trigger edge can be set through the CPEDGESEL bit. After the capture event occurs, the capture interrupt flag CF2 is set to 1. If timer 2 interrupts, it will trigger the capture interrupt, and CF2 will clear 0 by writing 1. In capture mode, counter overflow will generate overflow interrupt flag OF2, which is cleared by writing 1 to OF2.

Note: The T2CP pin can be mapped to different GPIO pins, as detailed in the "15-2-5 Pin Multiplexing Function Mapping Table".

12.3.2 Timer2 Register Description

Table 12-3-2-1 Register T2CON

A4H	7	6	5	4	3	2	1	0
T2CON	TR2	IE2	TF2	OF2	CF2	CPEDGESEL		T2M
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	TR2	Timer 2 working enabled, 1 effective
6	IE2	Timer 2 interrupt enable, 1 valid
5	TF2	Timer interrupt flag, write 1 clear 0
4	OF2	Overflow interrupt flag, write 1 clear 0
3	CF2	Capture interrupt flag bit, write 1 clear 0
2~1	CPEDGESEL	Capture mode trigger edge selection: 0, 3: Double edge trigger 1: Descending edge 2: Rising edge
0	T2M	Timing/capture function selection: 0: Timer mode 1: Capture mode

Table 12-3-2-2 Register T2CRL

A6H	7	6	5	4	3	2	1	0
T2CRL	TL2							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
Bit Number	Bit Symbol	Description						
7~0	T2CRL	Timer 2 overload value low byte or capture mode capture result low byte						

Table 12-3-2-3 Register T2CRH

A5H	7	6	5	4	3	2	1	0
T2CRH	TH2							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
Bit Number	Bit Symbol	Description						
7~0	T2CRH	Timer 2 overload value high byte or capture mode capture result high byte						

13 Watchdog Timer (WDT)

13.1 Watchdog Timer (WDT) Function Introduction

The watchdog timer is a 27 bit subtraction counter with an optional clock source, with a counting time range of 0.128ms-8.389s at 16MHz and 16 bit adjustment accuracy. Watchdogs are mainly used for monitoring systems to prevent CPU crashes due to external interference. If the software cannot refresh the watchdog timer before overflow, the watchdog will generate an internal reset or interrupt. Writing A5H to register WDFLG will refresh the watchdog, and reading WDFLG will obtain the watchdog status. In STOP mode, if the watchdog is in the enabled state, the clock source selected by the watchdog is working normally. At this time, if the watchdog is set to interrupt, the watchdog interrupt can wake up the CPU.

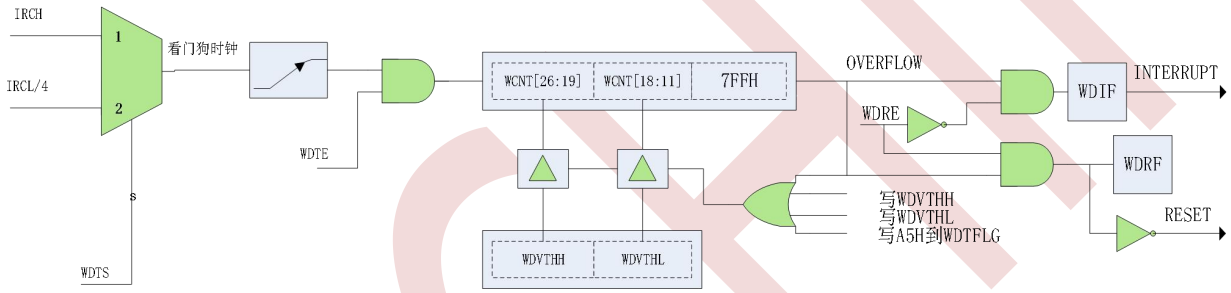


Figure 13-1-1 Watchdog Module Architecture

13.2 Watchdog Timer (WDT) Register Description

Table 13-2-1 Register WDCON

A7H	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 001: IRCH 010: IRCL with frequency divided by 4 Other: WDT disabled
5~1	-	-
0	WDRE	WDT function selection 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

Bit Number	Bit Symbol	Description
7~2	-	-
1	WDIF	WDT interrupt flag, writing A5H to the register will clear it
0	WDRF	WDT reset flag, 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	WDT threshold setting, the equation is as follows: WDT trigger time ＝ (WDVTH * 800H+7FFH) * clock cycle

13.3 Watchdog Timer Control Example

◆ Example for Watchdog interrupt mode

For example, if the watchdog clock is set to IRCH, the frequency of IRCH is 16MHz, the watchdog is set to interrupt mode, and the overflow time is 1 second, the program is as follows:

```
--------------------------------------------------------------------------------
void WDT_init(void)
{
    WDCON = (1<<6) | 0;              // 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;                      // Turn on watchdog interrupt
    EA = 1;                          // Turn on total interruption
}
void WDT_ISR (void) interrupt 6
{
    if(WDFLG & 0x02)
    {
        WDFLG = 0xA5;                // Refresh Watchdog
    }
}
--------------------------------------------------------------------------------
```

◆ Example for watchdog reset mode

For example, if the watchdog clock is set to IRCH, the frequency of IRCH is 16MHz, the watchdog is set to reset mode, and the overflow time is 1 second, the program is as follows:

```
--------------------------------------------------------------------------------
void WDT_init(void)
{
    WDCON = (1<<6) | 1;              // Set watchdog clock to IRCH, watchdog 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 of TMC timer can be IRCL or XOSCL. When the clock source is IRCL, the minimum unit of interrupt time is 512 clock cycles. When the clock source is XOSCL, the minimum unit of interrupt is 128 clock cycles, and the configurable interrupt time is 1-256 minimum unit times. In STOP/IDLE mode, TMC interrupts can wake up the CPU.

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	TME	TME module enable, 1 active
6~1	-	-
0	TMF	TMC interrupt flag, 1 valid, write 1 to clear 0

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 When the clock source of the TMC timer is IRCL, the interrupt time of the TMC is (TMSNU+1) × 512 × Tircl; When the clock source of the TMC timer is XOSCL, the interrupt time of the TMC is (TMSNU+1) x128 × Txoscl

14.3 TMC Control Routines

program as follows.

```
--------------------------------------------------------------------------------
#define TME(N)                  (N<<7)    // N=0-1 TME module enabled, 1 effective
#define TMF                     (1<<0)    // TMC interrupt flag, 1 valid, write 1 clear 0

#define XLCKE                   (1<<4)    // XOSC clock enable control bit
#define ILCKE                   (1<<6)    // IRCL enable control bit
#define TMCS(N)                 (N<<1)    // N=0-1, TMC counting clock selection
#define TMC_CLK_SELECT                    ILCKE
void INT3_ISR (void) interrupt 5
{
    if(TMCON & TMF)                       // Determine TMC interrupt flag
    {
        TMCON |= TMF;                     // Clear TMC interrupt flag

    }
}

void TMC_Init(void)
{
#if (TMC_CLK_SELECT == ILCKE)
    CKCON |= ILCKE;                       // Turn on IRCL clock
    CKCON |= TMCS(0);                     // TMC counting clock selection IRCL clock
#elif (TMC_CLK_SELECT == XLCKE)
    P10F = 11;                            // Set P1.0 as the crystal oscillator pin
    P11F = 11;                            // Set P1.1 as the crystal oscillator pin
    CKCON |= XLCKE;                       // Enable XOSCL clock
    CKCON |= TMCS(1);                     // TMC counting clock XOSCL clock
#endif
    TMCON = TME(1);                       // TMC enable
    TMSNU = 0;                            // Set interruption time
    INT3EN =1;                            // Enable TMC interrupt
    EA = 1;                               // Enable Total Interrupt
}
--------------------------------------------------------------------------------
```

15 General Purpose Input/Output(GPIO) and Alternate Functions

15.1 Function Introduction

The CA51F152S3X series chip has a maximum package of 14 I/O pins, each of which is a multiplexed function pin. It can be independently programmed as an input/output port and can also be set as other function pins. Each pin is assigned a function setting register PnxF (corresponding to pin Pnx, where n=0~5 represents P0, P1, P2, P3, P4, P5, x=0~7 represents Pn. 0~Pn. 7), and users can configure the main function and other options of the pin through the register PnxF. Please refer to the register section for details.

Main features of GPIO：

- High impedance mode configurable
- The I/O structure can independently set pull-up and pull-down resistors
- Output mode can be open-drain or push-pull output
- Data output latch supports read modify write
- When pushing and pulling output, the maximum current of a single GPIO source can reach 18mA@5v The maximum current can reach 40mA@5v
- GPIO can be software simulated as a 1/2 BIAS LCD driver without the need for external resistors

The GPIO push-pull mode structure diagram is shown in Figure

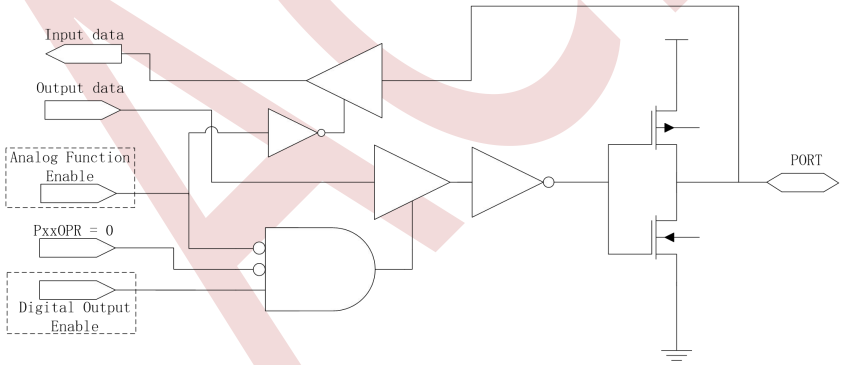


Figure 15-1-1 I/O Push-pull Mode Structure

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

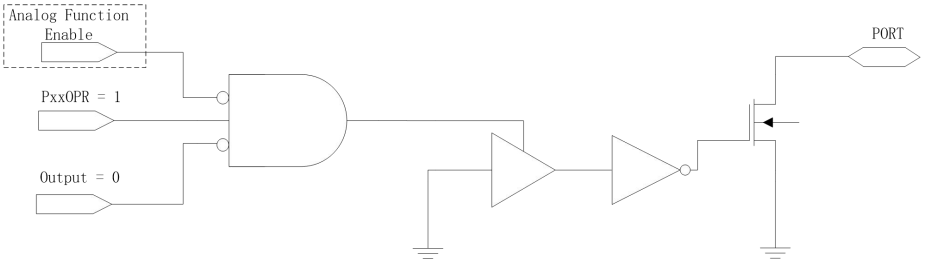


Figure 15-1-2 I/O Open-drain Mode Structure

The GPIO pull-down structure diagram is shown in Figure 15-1-3.

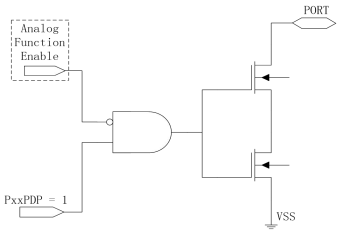


Figure 15-1-3 I/O Pull-down Mode Structure

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

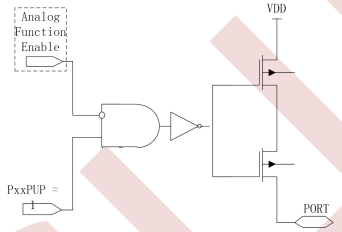


Figure 15-1-4 I/O Pull-up Mode Structure

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:P0x is low level when the pin is set to input; when the pin set to output,P0x outputs low level signal 1:P0x is high level when the pin is set to input; 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	-	-	P15	P14	P13	P12	P11	P10
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	-	-	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: P1x is low level when the pin is set to input; when the pin set to output,P1x outputs low level signal 1: P1x is high level when the pin is set to input; when the pin set to output,P1x outputs high level signal

Table 15-2-3 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	1	1	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	1	1	1
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	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	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	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	0	0	0
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	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	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	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	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	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	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	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	0

Bit Number	Bit Symbol	Description
7	PnxPUP	Pull-up resistor enable control 0: disable pull-up resistor 1: enable pull-up resistor
6	PnxPDP	Pull-down resistor enable

		0: disable pull-down resistor 1: enable pull-down resistor
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
4	-	-
3~0	PnxS	The pin reuse function settings are detailed in Table 15-2-5

Note: Pnx → n=0,1, representing P0 and P1

X=0~7, representing Pn. 0~Pn. 7

Table 15-2-4 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	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial Value	-	1	-	-	-	-	0	0

8128H	7	6	5	4	3	2	1	0
P10C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial Value	-	1	-	-	-	-	0	0
8129H	7	6	5	4	3	2	1	0
P11C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial Value	-	1	-	-	-	-	0	0
812AH	7	6	5	4	3	2	1	0
P12C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial Value	-	1	-	-	-	-	0	0
812BH	7	6	5	4	3	2	1	0
P13C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial Value	-	1	-	-	-	-	0	0
812CH	7	6	5	4	3	2	1	0
P14C	-	SMIT_EN	-	-	-	-	DRV	SR
R/W	-	R/W	-	-	-	-	R/W	R/W
Initial Value	-	1	-	-	-	-	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

Bit Number	Bit Symbol	Description
7	-	-
6	SMIT_EN	SMIT enable for input 1, inverter enable for input 0
5~2	-	-
1	DRV	Output intensity selection Note: The two-stage drive current is optional, please refer to the Electrical Characteristics section for details
0	SR	Output slope control 0: Slowest slope control 1: Fastest slope control

Table 15-2-5 Pin Multiplexing Function Mapping Table

Take value / Name	0	1	2	3	4	5	6	7	8	9	10	11
P00S	High resistance	Digital Input	Digital Output	PWM0_B	ADC0	TK0	I2C_SDA	I2C_SCL	UART2_RX	UART2_TX	T2CP	High resistance
P01S	High resistance	Digital Input	Digital Output	PWM1_B	ADC1	TK1	I2C_SDA	I2C_SCL	UART2_RX	UART2_TX	T2CP	High resistance
P02S	High resistance	Digital Input/T0	Digital Output	PWM2_A	ADC2	TK2	I2C_SDA	I2C_SCL	UART2_RX	UART2_TX	T2CP	High resistance
P03S	High resistance	Digital Input/T1	Digital Output	PWM3_A	ADC3	TK3	I2C_SDA	I2C_SCL	UART2_RX	UART2_TX	T2CP	High resistance
P04S	High resistance	Digital Input	Digital Output	PWM0_A	ADC4	TK4	I2C_SDA	I2C_SCL	UART2_RX	UART2_TX	T2CP	High resistance
P05S	High resistance	Digital Input	Digital Output	PWM1_A	ADC5	TK5	I2C_SDA	I2C_SCL	UART2_RX	UART2_TX	T2CP	High resistance
P06S	High resistance	Digital Input	Digital Output	PWM2_C	ADC6	TK6	I2C_SDA	I2C_SCL	UART2_RX	UART2_TX	SPI_SDI	High resistance
P07S	DAC	Digital Input	Digital Output	PWM3_C	ADC7	TK7	I2C_SDA	I2C_SCL	UART2_RX	UART2_TX	SPI_SDO	High resistance
P10S	High resistance	Digital Input	Digital Output	PWM0_E	ADC8	TK8	I2C_SDA	I2C_SCL	UART1_RX	UART1_TX	SPI_SCK	32K_OUT
P11S	High resistance	Digital Input	Digital Output	PWM1_E	ADC9	TK9	I2C_SDA	I2C_SCL	UART1_RX	UART1_TX	SPI_SCS	32K_IN
P12S	High resistance	Digital Input	Digital Output	PWM2_D	ADC10	TK10	I2C_SDA	I2C_SCL	UART1_RX	UART1_TX	SPI_SDI	High resistance
P13S	High resistance	Digital Input	Digital Output	PWM3_D	ADC11	TK11	I2C_SDA	I2C_SCL	UART1_RX	UART1_TX	SPI_SCK	High resistance
P14S	High resistance	Digital Input	Digital Output	PWM0_F	ADC12	TK12	I2C_SDA	I2C_SCL	UART1_RX	UART1_TX	T2CP	High resistance

P15S	High resistance	Digital Input	Digital Output	PWM1_F	ADC13	TK13	I2C_SDA	I2C_SCL	UART1_RX	UART1_TX	T2CP	High resistance

15.3 Pin control Example

- **Set the Pin function**

  For example, P0.0 is set to push-pull output, outputting high and low levels. The program is as follows:

```
P00F = 2;
```

  P0.0 is set to open drain output, and the program is as follows:

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

  P0.0 is set to open drain output and pull-up is turned on. The program is as follows:

```
P00F = (1<<7) | (1<<5) | 2;
```

  P0.0 is set as the input function and pull-up is turned on. The program is as follows:

```
P00F = (1<<7) | 1;
```

16 Universal Asynchronous Receiver/Transmitter (UART1/UART2)

16.1 Function Introduction

UART1 and UART2 are two fully duplex asynchronous serial data transceivers with identical designs, and UARTx (x=1,2, referring to UART1 and UART2) also has a one byte receive cache. UARTx has two different working modes, as shown in Table 16-1-1.

The TX/RX function of UART1/UART2 can be mapped to different GPIO pins, as detailed in the "15-2-5 Pin Multiplexing Function Mapping Table".

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

Table 16-1-1 UARTx Working mode

The UARTx is designed with a dedicated baud rate generator and the baud rate is configured through registers SxRELL, SxRELH.

Figure 16-1-1 shows the schematic diagram of UARTx.

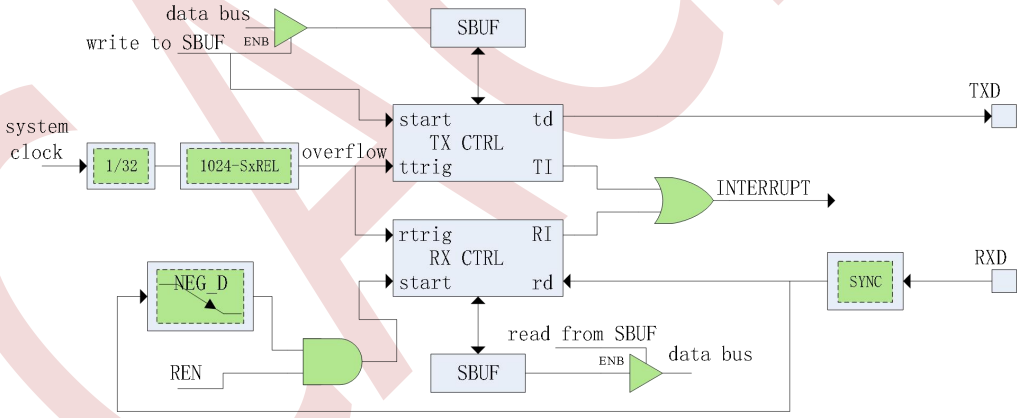


Figure 16-1-1 Schematic diagram of UARTx working principle

- **Mode A**

In mode A, the UARTx can send and receive 9 bits of data simultaneously asynchronously. Writing data to register SxBUF initiates UARTx data transmission. The first bit transmitted is the start bit (for 0), followed by 9 bits of data (low first), the 9th bit of data is the TB81 bit of register SxCON, and the last bit transmitted is the stop bit (for 1). In the receive state, UARTx is synchronized by detecting the falling edge of pin RX. After the transmission process is completed, the low 8 bits of data are stored in register SxBUF and the 9th bit of data is

stored in bit RB8x.

- **Mode B**

Mode B differs from Mode A in that Mode B is an 8-bit data transfer, and the stop bit holds a valid stop bit. Other functions are the same as Mode A.

- **UARTx Multi-Machine Communication**

There is a mechanism in UARTx mode A that is specifically applicable to multi-machine communication. When the SM2x position of register SxCON is 1, only the slave that receives the 9th bit data as 1 (RB8x=1) will generate the receive interrupt. Using this function, multi-machine communication can be performed; the slaves set all their SM2x bits to 1, and the host sets the 9th bit data to 1 when transmitting the slave's address, so that all the slaves will generate the receive interrupt; the slave's software uses their own address and the If they agree, the addressed slave sets SM2x=0, and then the host sets bit 9 to 0 when it continues to transmit the next data, because the other slaves still have SM2x set to 1, so that only the addressed slave generates a receive interrupt.

16.2 UARTx Register Description

Table 16-2-1 Register S1CON

98H	7	6	5	4	3	2	1	0
S1CON	SM1	U1IE	SM21	REN1	TB81	RB81	TI1	RI1
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	SM1	UART1 mode selection , for more information please refer to Table 16-1-1
6	U1IE	UART1 interrupt enable , 1 valid
5	SM21	Multi-computer communication enable control, 1 enables
4	REN1	Serial receive enable control, 1 enables
3	TB81	The 9th data bit to transmit In mode A, this bit is used for UART1 to transmit data and corresponds to bit 9 of the transmitted data (e.g. parity or multi-host communication), controlled by software
2	RB81	The 9th bit of the data received In mode A, this bit is used for UART1 to receive data, corresponding to bit 9 of the received data In mode B, this bit is the received stop bit
1	TI1	Transmit interrupt flag bit, 1 valid, write 1 to clear 0
0	RI1	Receive interrupt flag bit, 1 valid, write 1 to clear 0

Table 16-2-2 Register S1BUF

99H	7	6	5	4	3	2	1	0
S1BUF	S1BUF[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0

Bit Number	Bit Symbol	Description
7~0	S1BUF	UART1 transceiver buffer Writing S1BUF will start sending the written data Reading S1BUF will result in the received data

Table 16-2-3 Register UDCKS1

D8H	7	6	5	4	3	2	1	0
UDCKS1	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	Fast baud rate configuration enable control bit, 1 valid Note: When UDE=0, the UART1 baud rate is configured according to the original configuration, and when UDE=1, the UART1 baud rate is configured by DNUM.
6~5	-	-
4~0	DNUM	Fast baud rate configuration register, only valid when UDE=1 When sending, DNUM>=0 must be met; When receiving, DNUM>=6 BR=Fsys * (1/(DNUM+1) * (1024-S1REL))

Table 16-2-4 Register S1RELL、S1RELH

8068H	7	6	5	4	3	2	1	0
S1RELL	S1RELL[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
S1RELH	-	-	-	-	-	-	S1REL[9:8]	
R/W	-	-	-	-	-	-	R/W	
Initial Value	-	-	-	-	-	-	0	0

Bit Number	Bit Symbol	Description
9~0	S1REL	Baud rate configuration register Baud rate is CPUCLK/(32 * (1024 – S1REL))

Table 16-2-5 Register S2CON

B9H	7	6	5	4	3	2	1	0
S2CON	SM2	U2IE	SM22	REN2	TB82	RB82	TI2	RI2
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	SM2	UART2 mode selection , for more information please refer to Table 16-1-1
6	U2IE	Serial port 2 interrupt enable bit, 1 valid
5	SM22	Multi-computer communication enable control, 1 enables
4	REN2	Serial receive enable control, 1 enables
3	TB82	The 9th data bit to transmit In mode A, this bit is used for UART1 to transmit data and corresponds to bit 9 of the transmitted data (e.g. parity or multi-host communication), controlled by software
2	RB82	The 9th bit of the data received In mode A, this bit is used for UART2 to receive data, corresponding to bit 9 of the received data In mode B, this bit is the received stop bit
1	TI2	Transmit interrupt flag bit, 1 valid, write 1 to clear 0
0	RI2	Receive interrupt flag bit, 1 valid, write 1 to clear 0

Table 16-2-6 Register S2BUF

BAH	7	6	5	4	3	2	1	0
S0BUF	S0BUF[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0

Bit Number	Bit Symbol	Description
7~0	S0BUF	Receiver/Transmitter buffer Writing data to S2BUF will starts the data transmission Reading S2BUF will reads the data received

Table 16-2-7 Register UDCKS2

BDH	7	6	5	4	3	2	1	0
UDCKS2	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	Fast baud rate configuration enable control bit, 1 valid Note: When UDE=0, the UART2 baud rate is configured according to the original configuration, and when UDE=1, the UART2 baud rate is configured by DNUM.
6~5	-	-
4~0	DNUM	Fast baud rate configuration register, only valid when UDE=1 When sending, DNUM>=0 must be met; When receiving, DNUM>=6 BR=Fsys * (1/(DNUM+1) * (1024-S2REL))

Table 16-2-8 Register S2RELL、S2RELH

BBH	7	6	5	4	3	2	1	0
S2RELL	S2RELL[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
BCH	7	6	5	4	3	2	1	0
S2RELH	-	-	-	-	-	-	S2REL[9:8]	
R/W	-	-	-	-	-	-	R/W	
Initial Value	-	-	-	-	-	-	0	0

Bit Number	Bit Symbol	Description
9~0	S2REL	Baud rate configuration register Baud rate is CPUCLK/(32 * (1024 – S2REL))

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.

The SCL/SDA function of the I2C pin can be mapped to any GPIO pin and selected by setting different values in the registers SCLSEL/SDASEL. Please refer to the description of registers SCLSEL/SDASEL for details.

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 multimaster's arbitration
- Broadcast function supported
- Support SCL/SDA mapping to any GPIO

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.

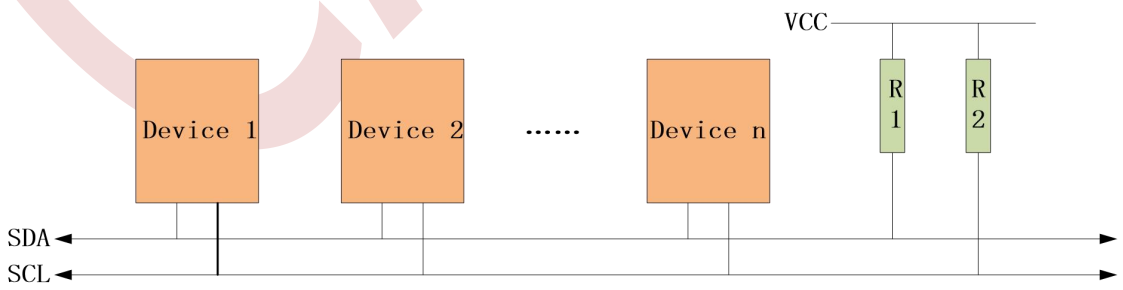


Figure 17-3-1 I2C Bus Interconnection Diagram

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

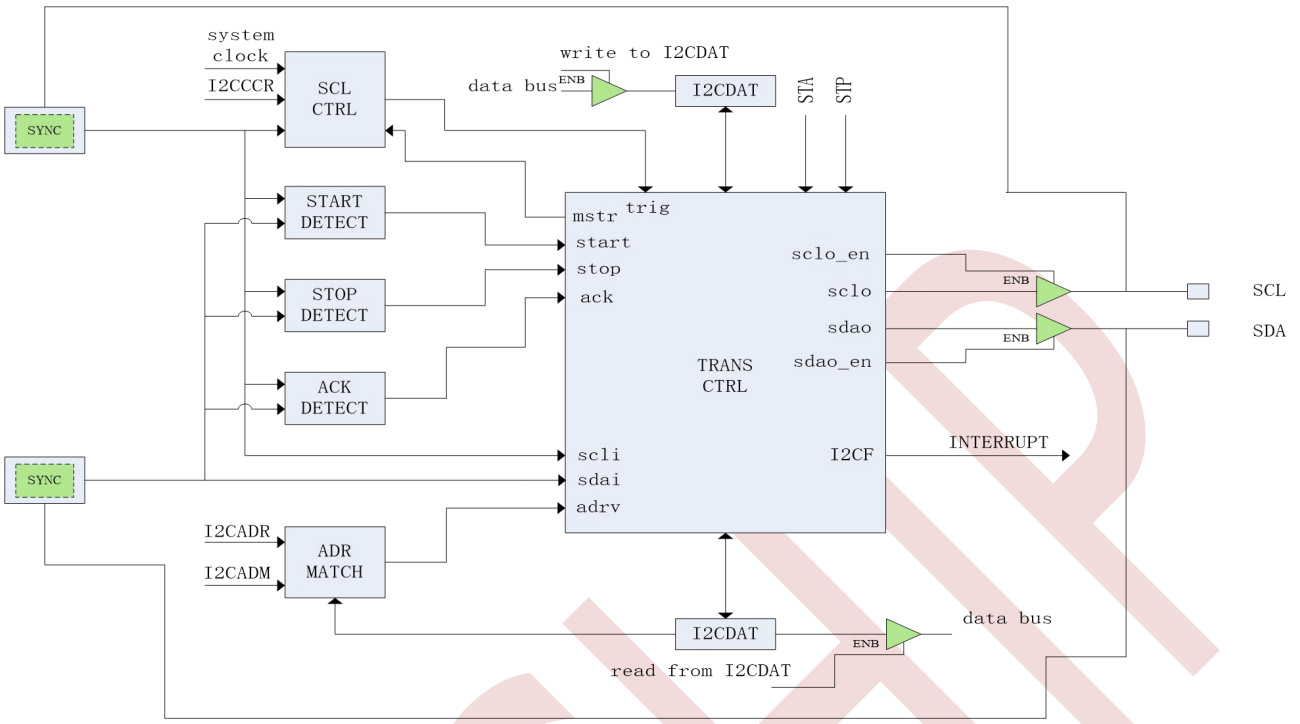


Figure 17-3-2 I²C Schematic diagram of the module princip

- **I²C Mode Selection**

I2C can operate in one of the following four modes: slave transmit mode, slave receive mode, host transmit mode, and host receive mode. By default, I2C is in slave mode. I2C automatically switches from slave mode to host mode when a start signal is generated, and then automatically switches back to slave mode when arbitration fails or a STOP signal is generated.

- **I2C Bus Data Transmission Pattern**

In general, the standard I2C communication consists of four parts: start signal, slave address transmission, data transmission and end signal. 8 bits of data are transmitted on the I2C bus, high bit first, each byte sent must be followed by an answer bit, there is no limit to the number of data bytes per communication; at the end of all data transmission, the host sends a stop signal to end the communication.

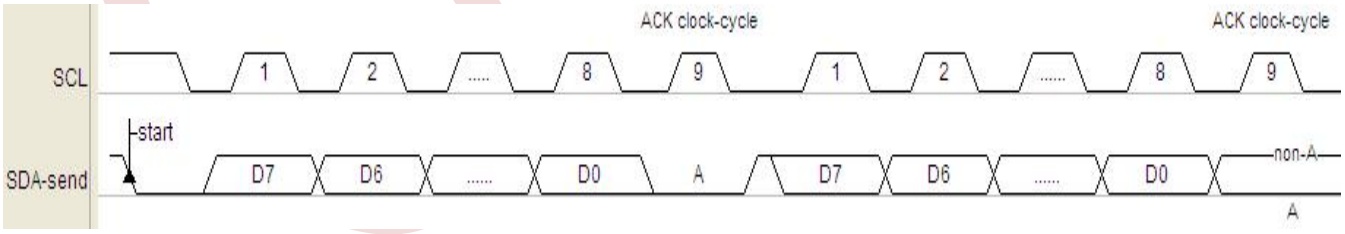


Figure 17-3-3 I2C Bus Data Transfer Format

- **Communication Process**

In host mode, the I2C interface initiates the data transfer and generates the clock signal. The serial data transfer always starts with the START signal and ends with the STOP signal. both the START signal and the STOP signal are generated in host mode by software control, the START signal is generated by setting STA=1 and the STOP signal is generated by setting STP=1.

In slave mode, the I2C interface can recognize its own address (7-bit address) and the broadcast address.

The software can enable or disable the recognition of the broadcast address via the GCE bit.

Address and data are transmitted in bytes, and the address is sent by the host after the START signal. In the 9th clock cycle after 8 clocks of a byte transmission, the receiver must send back an answer bit to the transmitter. The answer bit is set via the AAK bit. Setting the answer bit must be set before a byte is finished transmitting, and the answer signal is automatically generated when the receiver finishes receiving a byte. During data transmission, events such as data sending/receiving finished one byte, arbitration failure will generate interrupt flag I2CF, and the status of the event is indicated by register I2CSTA (please refer to register I2CSTA introduction for details), the software should set the next operation of data transmission according to the status of the event after generating interrupt flag, clearing interrupt flag I2CF will start the next operation. After the communication ends the host generates the STOP signal will also generate the interrupt flag I2CSTP at the slave side to indicate the completion of the communication process. When the interrupt flag I2CF is generated, if SHD=1, SCL will be pulled low by the slave before clearing I2CF, and the host will detect that SCL is released before the next operation; if SHD=0, the slave will not pull low SCL, which is designed to be compatible with the application that the host is software emulating I2C, at this time, the host's software must wait long enough for the slave to respond to each byte of data transfer is processed.

When the I2C interface serves as a slave, the SCL clock is input by the master and is independent of the slave's clock configuration. As a slave, it is necessary to ensure that the width of SCL at low levels is at least 6.5 system clocks, while at high levels it is at least 2.5 system clocks. So, the SCL frequency sent by the external host is up to 1/9 of the system clock frequency.

17.4 I^2C Communication Pin Mapping

For the convenience of hardware design, the I2C communication pins can have different mappings, which can be selected by setting different values in the registers SCLSEL/SDASEL. Please refer to the register SCLSEL/SDASEL description for details.

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	I2C module enable control, 1 enables it
6	I2CIE	I2C interrupt enable control, 1 enables it1
5	STA	I2C START signal transfer control,valid when it is 1, it will be cleared automatically when START signal detected
4	STP	I2C STOP signal transfer control, valid when it is 1, it will be cleared automatically when STOP signal detected

3	SHD	When it is 1, if I2CF=1, I2CF will make SCL remain low after SCL becomes low
2	AAK	I2C ACK signal transfer control, 1 enables it Note : When I2C is configured as slave, this bit must be set to 1 beforehand, otherwise even the address matches it will not reply ACK
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. Since the address for CBUS bus is 7 bits, thus GCE must be set to 0.
0	STFE	When STFE=1, I2CF will be set to 1 if I2C 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	I2C 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 17-5-3 Register I2CADM

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	When SPFE=1, I2CF will be set to 1 if I2C module detects the STOP signal
6~0	I2CADML	I2C address mask by bit control, valid only when it operates as slave When I2CADM[n](n=0~6)=1, the corresponding address bit I2CADR[n] will not be compared (which means no matter what is received, it is seen as matched)

Table 17-5-4 Register I2CCCR

C3H	7	6	5	4	3	2	1	0
I2CCCR	I2CCCR[7:0]							
R/W	R/W							
Initial Value	0	0	1	0	0	0	0	0

Bit Number	Bit Symbol	Description
7~0	I2CCCR	I2C clock configuration register The sampling frequency is the division of the $2^{I2CCCR\ [7:5]}$ of the I2C working clock, when I2CCCR [7:5] is equal to 000:$F_{sample}=F_{i2cclk}$ 001:$F_{sample}=F_{i2cclk}/2$ 010:$F_{sample}=F_{i2cclk}/4$ ... 111:$F_{sample}=F_{i2cclk}/128$ (I2CCCR [4:0]+1) division with an output frequency of sampling frequency, $F_{scl}=F_{i2cclk}/(2^{I2CCCR}$ [7:5] * (I2CCCR [4:0]+1)) For example, when I2CCR [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: When I2CCCR [7:5]=0, if a value less than 9 is written to I2CCCR [4:0], it will be automatically calculated based on the value of 9. When I2CCCR [7:5]>0, if a value less than 7 is written to I2CCCR [4:0], it will be automatically calculated based on the 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	Data 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	I2C 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	I2C interrupt 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 STOP signal detected

Table 17-5-8 Register SCLSEL

D7H	7	6	5	4	3	2	1	0
SCLSEL	I2CKS	-	-	I2CS				
R/W	R/W	-	-	R/W				
Initial Value	0	-	-	0	0	0	0	0

Bit Number	Bit Symbol	Description
7	I2CKS	I2C working clock selection bit 0: System clock 1: Internal high-speed clock
6~5	-	-
4~0	I2CS	I2C Pin selection bit 00000： SCL at pin P0.0 00001： SCL at pin P0.1 00010： SCL at pin P0.2 00011： SCL at pin P0.3 00100： SCL at pin P0.4 00101： SCL at pin P0.5 00110： SCL at pin P0.6 00111： SCL at pin P0.7 01000： SCL at pin P1.0 01001： SCL at pin P1.1 01010： SCL at pin P1.2 01011： SCL at pin P1.3 01100： SCL at pin P1.4 01101： SCL at pin P1.5 Other: SCL pin not selected

Table 17-5-9 Register SDASEL

C7H	7	6	5	4	3	2	1	0
SDASEL	-	-	-	I2CS				
R/W	-	-	-	R/W				
Initial Value	-	-	-	0	0	0	0	0

Bit Number	Bit Symbol	Description
7~5	-	-
4~0	I2CS	I2C SDA Pin selection bit 00000： SDA at pin P0.0 00001： SDA at pin P0.1

		00010： SDA at pin P0.2 00011： SDA at pin P0.3 00100： SDA at pin P0.4 00101： SDA at pin P0.5 00110： SDA at pin P0.6 00111： SDA at pin P0.7 01000： SDA at pin P1.0 01001： SDA at pin P1.1 01010： SDA at pin P1.2 01011： SDA at pin P1.3 01100： SDA at pin P1.4 01101： SDA at pin P1.5 Other: SDA pin not selected

18 PWM

18.1 PWM Function Introduction

The CA51F152S3 series chip supports up to 4 PWM outputs, each of which can be individually controlled. The cycle and duty cycle can be configured arbitrarily within a 16 bit range. Each PWM output pin can be mapped to different pins, as detailed in the "15-2-5 Pin Reuse Function Mapping Table".

18.2 PWM(0~3) Function Description

Each PWM channel has a dedicated 16 bit counter, and the PWM cycle is set through register PWMDIV, while register PWMDUT corresponds to the PWM duty cycle. PWM is enabled through register PWMEN, and each bit of register PWMEN corresponds to a channel of PWM. PWM can set the PWM pin output to invert through the PWMTOG bit. PWM has multiple clock sources to choose from, and each clock source is set separately, with the corresponding control register being the PWMCKS bit of PWMCON. In addition, the clock division of each PWM can be independently set through PWMCKD.

It is worth noting that when PWMDIV=0, the PWM pin directly outputs a PWM clock. If PWMCKD=0, the PWM pin outputs the clock signal of the selected clock source; When PWMDIV is not 0 and PWMDUT is 0, the PWM pin outputs a low level (PWMTOG=0); When PWMDUT>=PWMDIV>0, the PWM pin outputs a high level (PWMTOG=0).

PWM interrupt

PWM interrupt is enabled through the PWMIE bit of register PWMCON. When PWMTOG=0, the channel generates an interrupt on the rising edge; When PWMTOG=1, the channel generates an interrupt for the falling edge. The low 4 of register PWMIF corresponds to the interrupt status of each of the 4 channels.

18.3 PWM Register Description

Table 18-3-1 Register PWMEN

91H	7	6	5	4	3	2	1	0
PWMEN	-	-	-	-	PWMEN[3:0]			
R/W	-	-	-	-	R/W			
Initial Value	-	-	-	-	0	0	0	0
Bit Number	Bit Symbol	Description						
7~4	-	-						
3~0	PWMEN	3~0 bit correspond to PWM channel 3~0's enable control, 1 enables it						

Table 18-3-2 Register PWMCON

94H	7	6	5	4	3	2	1	0
PWMCON	PWMIE	PWMTOG	-	-	-	-	PWMCKS[1:0]	
R/W	R/W	R/W	-	-	-	-	R/W	
Initial Value	0	0	-	-	-	-	0	0

Bit Number	Bit Symbol	Description
Remarks: *1. PWMCON is an indexed register, with INDEX=0~3 corresponding to PWMCON0~3, respectively*		
7	PWMTIE	PWM interrupt enable control bit, 1 valid (PWMTOG=0, rising edge interrupt; PWMTOG=1, falling edge interrupt)
6	PWMTOG	PWM output reverse enable register, 1 valid
5~2	-	-
1~0	PWMCKS	PWM working clock selection bit 00: System clock 01: IRCL 10: IRCH 11: When INDEX=0, select TKRC; When INDEX is a different value, select the system clock.

Table 18-3-3 Registers PWMCKD

95H	7	6	5	4	3	2	1	0
PWMCKD	PWMCKD[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0

Note: PWMCKD is an indexed register, set INDEX=0~3 to correspond to PWMCKD0~PWMCKD3 respectively

Bit Number	Bit Symbol	Description
7~0	PWMCKD	PWM working clock pre division configuration register 000000: No frequency division 00000001:2 frequency division 0000010:3 Frequency division ..... 11111110:255 frequency division 11111111:256 Frequency division

Table 18-3-4 Register PWMDIVL、PWMDIVH

96H	7	6	5	4	3	2	1	0
PWMDIVL	PWMDIV[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
PWMDIVH	PWMDIV[15:8]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
Note: PWMDIV is an indexed register, set INDEX=0~3 to correspond to PWMDIV0~PWMDIV3 respectively								
Bit Number	Bit Symbol	Description						
15~0	PWMDIV	PWM cycle configuration register						

Table 18-3-5 Registers PWMDUTL、PWMDUTH

9BH	7	6	5	4	3	2	1	0
PWMDUTL	PWMDUT[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
PWMDUTH	PWMDUT[15:8]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
Note: PWMDUT is an indexed register, set INDEX=0~3 to correspond to PWMDUT0~PWMDUT3 respectively								
Bit Number	Bit Symbol	Description						
15~0	PWMDUT	PWM duty cycle configuration register						

Table 18-3-6 Registers PWMIF

92H	7	6	5	4	3	2	1	0
PWMIF	-	-	-	-	PWM3IF	PWM2IF	PWM1IF	PWM0IF
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Initial Value	-	-	-	-	0	0	0	0

Bit Number	Bit Symbol	Description
3	PWM3IF	PWM3 interrupt flag, write 1 clear 0
2	PWM2IF	PWM2 interrupt flag, write 1 clear 0
1	PWM1IF	PWM1 interrupt flag, write 1 clear 0
0	PWM0IF	PWM0 interrupt flag, write 1 clear 0

19 SPI interface

19.1 Function Introduction

The SPI interface enables the chip to transfer data with other devices in half/full duplex synchronization. The peripheral devices can be other MCUs, ADCs, sensors, or flash memory, etc. The SPI can be three or four wires and has the following features.

- Supports host or slave operation
- Selectable lowest or highest bit priority transmission
- 4 programmable bit rates
- Programmable polarity and phase
- Send end of interrupt flag
- Write to conflict flag protection mechanism
- Support main mode failure error interrupt

Figures 19-1-1 and 19-1-2 are schematic diagrams of the SPI master mode and slave mode, respectively.

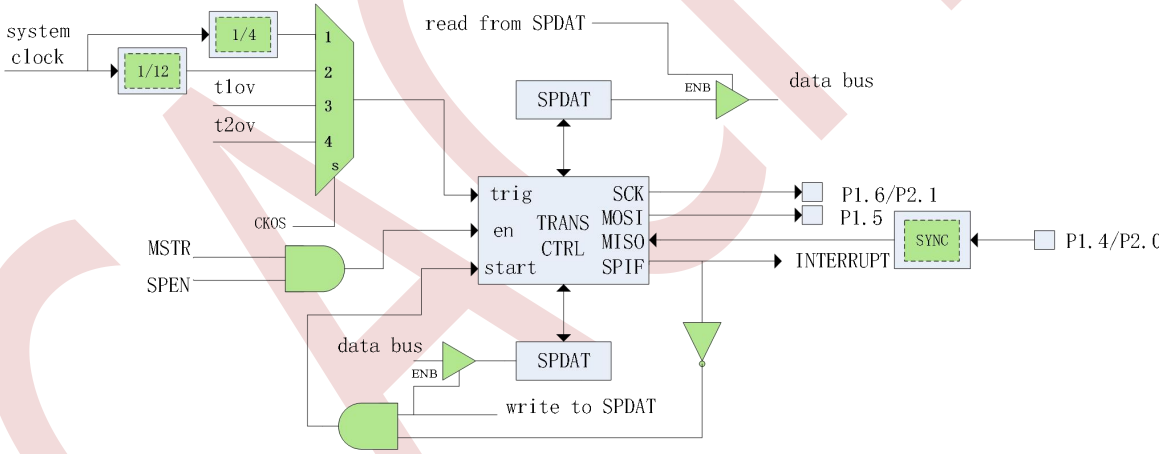


Figure 19-1-1 SPI host mode schematic

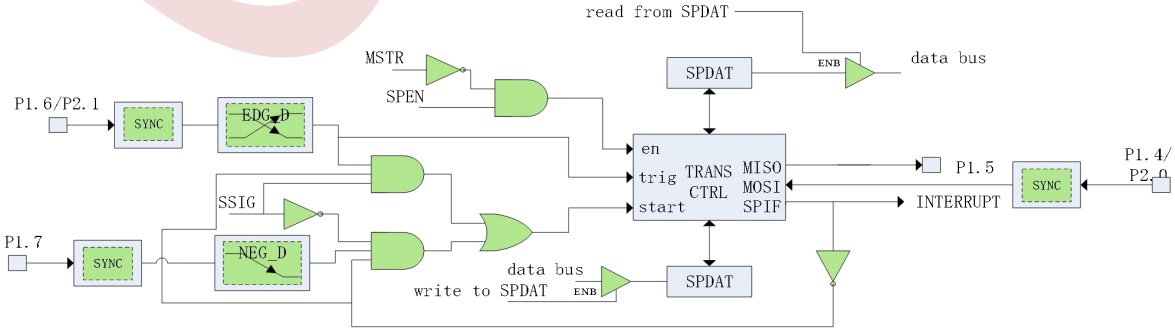


Figure 19-1-2 Schematic diagram of SPI slave mode

Table 19-1-1 SPI Operating Modes

Name	Description
Host Mode	All transmission actions are initiated by the host, including the generation of SCK and SSB signals, etc. When the MSTR (SPCON[4]) bit is set to 1, the SPI is in master mode. The user needs to select another GPIO as a chip select pin to connect to the slave SSB, and the host pulls this pin low before the data transfer starts and high after the transfer ends. In host mode, writing to register SPDAT initiates data transfer. Data is shifted out from MOSI on the valid edge of the clock.
Slave Mode	When the MSTR bit is set to 0, the SPI is in slave mode. When SSIG (SPCON[5]) is 1, the SSB pin is invalid, the SPI is three-wire communication, and the slave is valid for chip selection by default; when SSIG is 0, the SSB pin is valid and SSB is low to indicate that the slave is chip selected.

Table 19-1-2 SPI Interface Pin Descriptions

Name	Description
MOSI	Host output, slave input This pin is the master data output port when the SPI is used as a host and the slave data input port when it is used as a slave.
MISO	Host output, slave input This pin is the master data input port when the SPI is the host and the slave data output port when the SPI is the slave.
SCK	Serial Clock This pin is the serial clock output port when the SPI is the host and the serial clock input port when the SPI is the slave
SSB	Slave Selection This pin selects the input port for the slave when the SPI pin is the master, and the input port for the slave when it is the slave.

Table 19-1-3 SPI Phase and Polarity

Name	Description
CPHA	Phase control bits 0：denotes the data sampled at the odd edges of SCK (1,3,5,... ,15) sampled data 1：denotes data sampled at the even edges of SCK (2,4,6,... ,16) sampled data
CPOL	Polarity control bit 0：Indicates a low level when SCK is idle 1：Indicates that SCK is high when idle

Combined with Table 19-1-3, the waveforms during actual transmission are shown in Figure 19-1-3 and Figure 19-1-4.

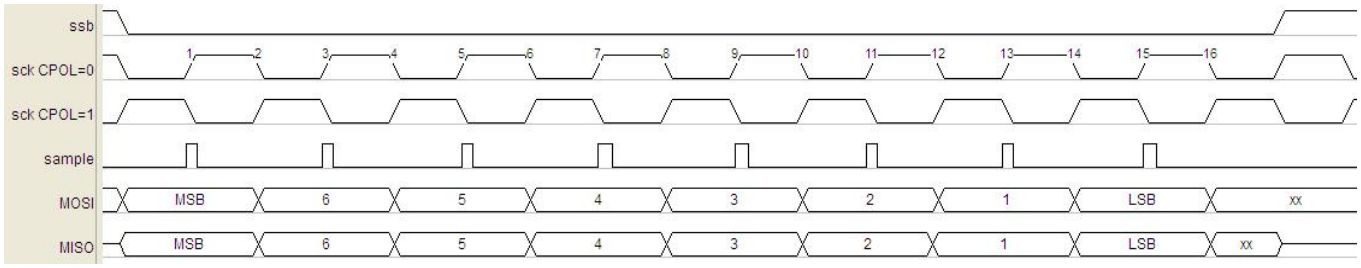


Figure 19-1-3 SPI Timing Diagram when CPHA=0

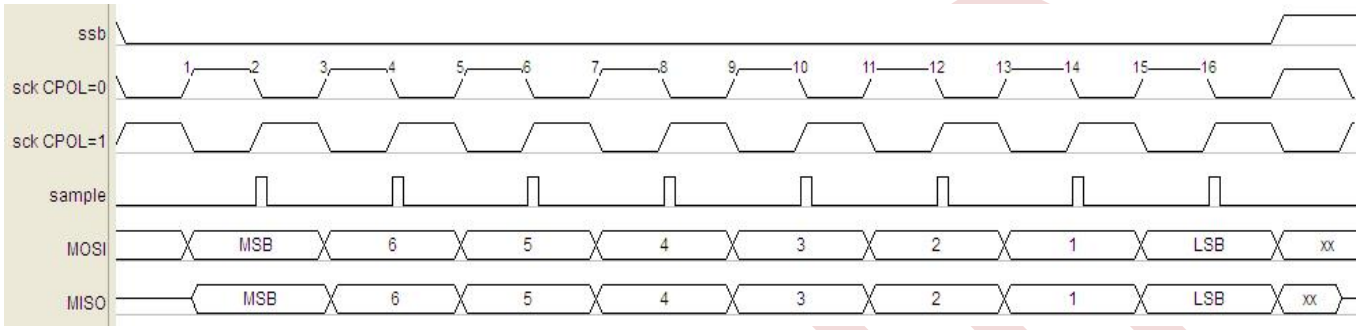


Figure 19-1-4 SPI Timing Diagram when CPHA=1

19.2 Register Description

Table 19-2-1 Register SPCON

A9H	7	6	5	4	3	2	1	0
SPCON	SPEN	LSBF	SSIG	MSTR	CPOL	CPHA	CKOS[1:0]	
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0

Bit Number	Bit Symbol	Description
7	SPEN	SPI module enable bit, 1 valid
6	LSBF	Low or high priority transmit/receive selection bit 0: High position first 1: Low position first
5	SSIG	SSB pin invalid control bit, the default is 0, when the SSB signal is valid.
4	MSTR	Host/Slave selection bit 0： Slave 1： Host
3	CPOL	Clock polarity selection bit 0: Clock is low by default 1: Clock is high by default
2	CPHA	Clock phase selection bit 0: Sample data when the clock leaves the default case 1: Sample data when the clock returns to the default case

1~0	CKOS	SPI output clock select bit 00:1/8 system clock 01: 1/24 system clock 10: Use timer 1 overflow flag to transfer data every two overflows

Table 19-2-2 Register SPDAT

AAH	7	6	5	4	3	2	1	0
SPDAT	RBUF[7:0]							
R/W	R							
Initial Value	0	0	0	0	0	0	0	0
SPDAT	TBUF[7:0]							
R/W	W							
Initial Value	0	0	0	0	0	0	0	0

Bit Number	Bit Symbol	Description
7~0	SPDAT	When writing SPDAT, write to internal TBUF, when reading SPDAT, read from RBUF

Table 19-2-3 Register SPSTA

ABH	7	6	5	4	3	2	1	0
SPSTA	SPIE	-	-	-	-	WCOL	MODF	SPIF
R/W	R/W	-	-	-	-	R/W	R/W	R/W
Initial Value	0	-	-	-	-	0	0	0

Bit Number	Bit Symbol	Description
7	SPIE	SPI interrupt enable bit, 1 valid
6~3	-	-
2	WCOL	The write conflict flag bit is generated when the data is being sent, if there is a software operation to write SPDAT, then the data cannot be written, that is, the write conflict flag is generated. This bit is valid for 1, clear 0 for write 1, no interrupt will be generated when it is valid.
1	MODF	Fault mode flag bit, 1 valid, indicates SSB at incorrect logic level, write 1 to clear 0, valid will generate interrupt
0	SPIF	Data transfer completion flag bit, 1 is valid, write 1 to clear 0, interrupt will be generated when valid

20 Analog/Digital Converter (ADC)

20.1 Function Introduction

The analog/digital converter is a 12 bit successive approximation register (SAR) ADC that provides up to 14 input channels. The ADC clock source is the system clock, which can be set for clock pre division. ADC has multiple reference voltage sources to choose from, among which selecting the internal voltage as the reference voltage can be used to detect the chip power supply voltage. ADC has an automatic correction function when selecting internal voltage as the reference voltage to avoid chip consistency issues.

20.2 Main Features

- 12 bit resolution
- 14 input channels at most
- Supports ADC interrupt
- ADC clock frequency division configurable
- Alternate reference voltage：internal reference voltage, VDD
- Support the measurement of VDD and reference ground voltage
- Automatic data correction supported when internal reference voltage is selected
- When selecting an internal reference voltage, VDD voltage can be measured
- Input voltage range：VSS<=VIN<=VDD

20.3 Block Diagram

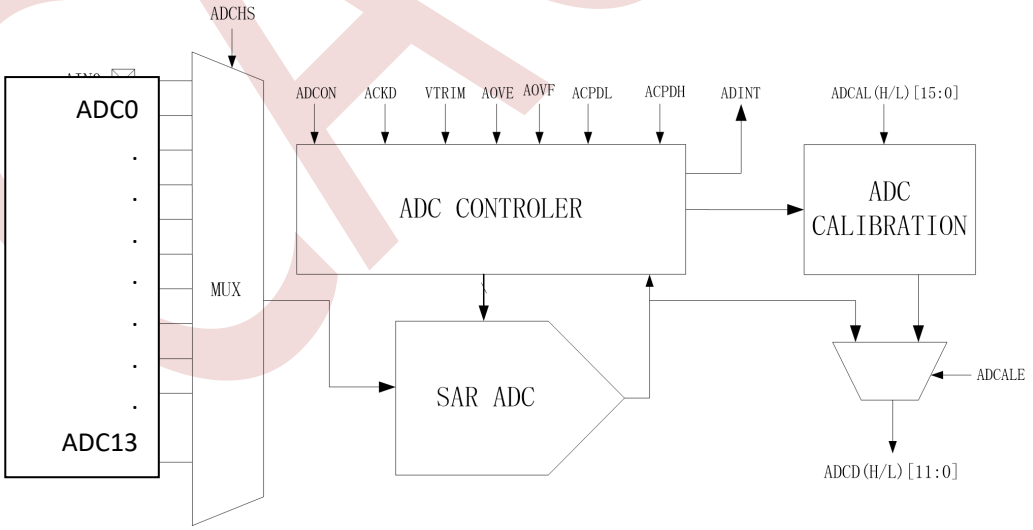


Figure 20-3-1 ADC Architecture

20.4 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 ADCIF will be set to 1 at the same time. If ADC interrupt is enabled then, ADC interrupt occurs. The shortest ADC conversion time is 0.5us. Figure 20-4-1 is the sequence diagram for ADC conversion.

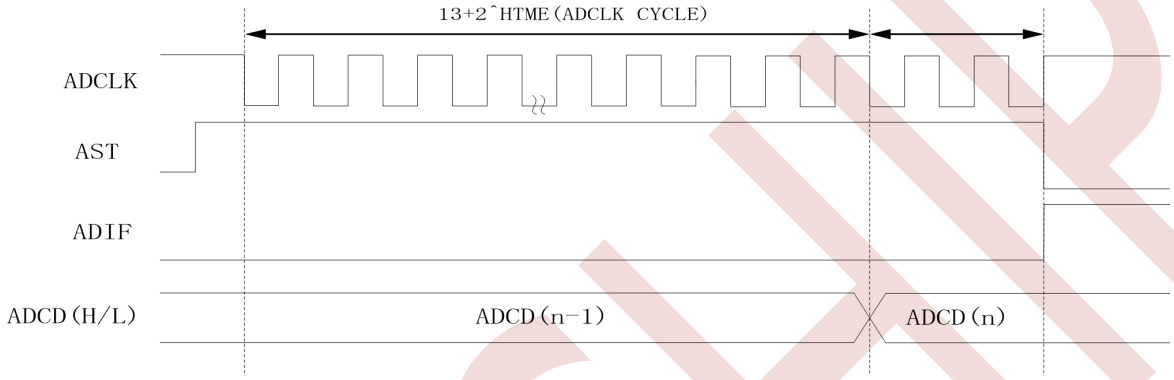


Figure 20-4-1 ADC Sequence Diagram

- **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 can not 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 to the correction value. 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.

20.5 Register Description

Table 20-5-1 Register ADCON

8060H	7	6	5	4	3	2	1	0
ADCON	AST	ADIE	ADCIF	HTME			ADCALE	VSEL
R/W	R/W	R/W	R/W	R/W			R/W	R/W
Initial Value	0	0	0	0	1	0	1	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 bit, write 1 to clear 0
4~2	HTME	The number of sampling periods is 2 power HTME
1	ADCALE	ADC calibration enable bit, 1 valid This bit is only valid when the reference voltage is selected as internal 1.5V. When ADCALE=1, the ADC conversion result will be calibrated based on the value of the ADCAL register. Please refer to register ADCAL Description for specific details.
0	VSEL	ADC reference voltage selection 0: internal 1.5V(INNER_VREF)as reference voltage 1: Power supply as reference voltage

Table 20-5-2 Register ADCFGL

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

Bit Number	Bit Symbol	Description
7~5	ACKD	ADC clock division setting 000: No frequency division 001: 2 Frequency division 010: 4 frequency division … 111: 14 frequency division
4~0	ADCHS	ADC channel enable selection bit field 00000: Channel closed 00001: Channel AD_ CH [0] (P0.0) enable 00010: Channel AD_ CH [1] (P0.1) enable 00011: Channel AD_ CH [2] (P0.2) enable 00100: Channel AD_ CH [3] (P0.3) enable 00101: Channel AD_ CH [4] (P0.4) enable 00110: Channel AD_ CH [5] (P0.5) enable 00111: Channel AD_ CH [6] (P0.6) enable 01000: Channel AD_ CH [7] (P0.7) enable 01001: Channel AD_ CH [8] (P1.0) enable 01010: Channel AD_ CH [9] (P1.1) enable 01011: Channel AD_ CH [10] (P1.2) enable 01100: Channel AD_ CH [11] (P1.3) enable 01101: Channel AD_ CH [12] (P1.4) enable 01110: Channel AD_ CH [13] (P1.5) enable 01111: Detect 1/4 enable of VDD Other: Channel closed

Table 20-5-4 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, only ADCALE=1 and the reference voltage is selected as internal 1.5V is valid. When valid, the ADC output is in accordance with the following formula: ADCDL=(ADC conversion result*ADCAL)/32768 Remarks: ADCAL is automatically loaded by the system after power-on, and users are not allowed to modify it.

Table 20-5-5 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 value

21 Capacitive touch key (Touch Key)

21.1 Function Introduction

The touch function module of the CA51F152S3 series chip has superior anti-interference performance and can pass tests such as EFT and CS. The touch module can support up to 14 channels at most. Equipped with a built-in touch sensing controller, there is no need for external capacitors during application. For applications with low power consumption requirements, support touch low power mode to achieve product power-saving function.

21.2 Main features

- Built in touch sensing controller
- Supports up to 14 touch channels without the need for external capacitors
- High anti-interference performance, in compliance with EMC (CS) standards
- Support touch interruption
- Support parallel connection of touch channels, enabling low-power touch mode

21.3 Function Description

The touch function requires setting corresponding parameters through a series of registers, including charging and discharging clock frequency, counting clock frequency, counting digits, discharge resistance, comparator voltage, charging current source current, etc. After setting the parameters, enable the touch function through TKEN. After setting the touch channel, set TKST=1 to start the touch data collection of the corresponding channel. After data collection is completed, the 16 bit touch data is stored in TKDH and THDL, and the touch interrupt flag TKIF is generated.

The touch module supports multiple touch channels in parallel, which can significantly reduce power consumption when used in touch power-saving mode.

For more touch related content, please refer to the touch library and related documentation, and it is recommended to conduct secondary development based on the touch library provided by our company.

21.4 Register description

Table 21-4-1 Register TKST

F8H	7	6	5	4	3	2	1	0
TKST	-	-	-	-	-	-	-	TKST
R/W	-	-	-	-	-	-	-	RW
Initial Value	-	-	-	-	-	-	-	0

Bit Number	Bit Symbol	Description
7~1	-	-
0	TKST	Data collection start enable bit, 1 is valid, automatically cleared to 0 after collection is completed

Table 21-4-2 Register TKCFG1

F9H	7	6	5	4	3	2	1	0
TKCFG1	-	TKEN	TKDIV					
R/W	-	R/W	R/W					
Initial Value	-	0	0	0	0	0	0	0

Bit Number	Bit Symbol	Description
7	-	-
6	TKEN	Enable touch simulation module
5~0	TKDIV	Charging and discharging clock selection: 0-61: The charging and discharging clock has a fixed frequency, and the clock frequency is F=24/(TKDIV+4) 62~63: Charge and discharge clock frequency hopping, maximum frequency 3M, minimum frequency 1M, center frequency 1.5MHz

表 21-4-3 寄存器 TKCFG2

FAH	7	6	5	4	3	2	1	0
TKCFG2	-	TKPULL8	PAREN	TKCADDR				
R/W	-	R/W	R/W	R/W				
Initial Value	-	1	0	0				

Bit Number	Bit Symbol	Description
7	-	-
6	TKPULL8	Charging current source configuration highest bit
5	PAREN	0: Single channel 1: All enabled channels are connected in parallel
4~0	TKCADDR	Channel selection bit: 00000: TK0

		00001:TK1 …… 01101:TK13

Table 21-4-4 Register TKCFG3

FBH	7	6	5	4	3	2	1	0
TKCFG3	-	RESO			TKCKSEL		CHGSEL	DCHSEL
R/W	-	R/W			R/W		R/W	R/W
Initial Value	-	0	0	1	1	1	1	1

Bit Number	Bit Symbol	Description
7~5	-	-
6~4	RESO	Counter bit selection 000: 9 digits 001: 10 digits 010: 11 digits 011: 12 digits 100: 13 digits 101: 14 digits 110: 15 digits 111: 16 digits
3~2	TKCKSEL	Count clock selection 0: 24Mhz 3: 4Mhz Other values: invalid
1	CHGSEL	Pre charging time selection 0:20 us 1: 40us
0	DCHSEL	Pre discharge time selection 0:2 us 1: 10us

Table 21-4-5 Register TKPULL

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

Bit Number	Bit Symbol	Description
7~0	TKPULL	Setting the charging current source to 8 low bits

Table 21-4-6 Register TKDL/TKDH

FCH	7	6	5	4	3	2	1	0
TKDL	TKD[7:0]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0
FDH	7	6	5	4	3	2	1	0
TKDH	TKD[15:8]							
R/W	R/W							
Initial Value	0	0	0	0	0	0	0	0

Bit Number	Bit Symbol	Description
7~0	TKD	The current sampling value of the touch channel

Table 21-4-7 Register TKPULLTRIM

F2H	7	6	5	4	3	2	1	0
TKPULLTRIM	-	-	TKPULLTRIM [5:0]					
R/W	-	-	RW					
Initial Value	-	-	0	0	0	0	0	0

Bit Number	Bit Symbol	Description
7~0	TKPULLTRIM	Charging current compensation value

Table 21-4-8 Register TKIE

F4H	7	6	5	4	3	2	1	0
TKIE	TKIE	-	-	-	-	-	-	TKIF
R/W	RW	-	-	-	-	-	-	R
Initial Value	0	-	-	-	-	-	-	0

Bit Number	Bit Symbol	Description
7	TKIE	Touch interrupt enable bit, 1 valid
6~1	-	-
0	TKIF	Touch to collect interrupt flag bit, write 1 clear 0

Table 21-4-9 Register TKCFG4

FFH	7	6	5	4	3	2	1	0
TKCFG4	-	-	RBSEL			VRSEL		
R/W	-	-	RW			RW		
Initial Value	-	-	0	0	0	0	0	0

Bit Number	Bit Symbol	Description
7~6	-	-
5~3	RBSEL	Selection of discharge resistance size 000:20K 001:30K 010:40K 011:50K 100:70K 101:90K 110:200K 111:350K
2~0	VRSEL	Comparator threshold voltage selection 000:1.8V 001:2.1V 010:2.5V 011:2.8V 100:3.2V 101:3.5V 110:3.9V 111:4.1V

22 Low Voltage Detection (LVD)

22.1 Function Introduction

Low voltage detection (LVD) is used to monitor the power supply VDD of the chip itself, and the detection voltage range can be set to 2.2V, 2.5V, 2.7V, 3.0V, 3.3V, 3.6V, 3.9V, and 4.2V. When the VDD is less than the set voltage value, a trigger interrupt or reset can be set.

Note: Due to the production process, there is a certain difference in LVD trigger voltage between chips.

Figure 22-1-1 shows the architecture of LVD.

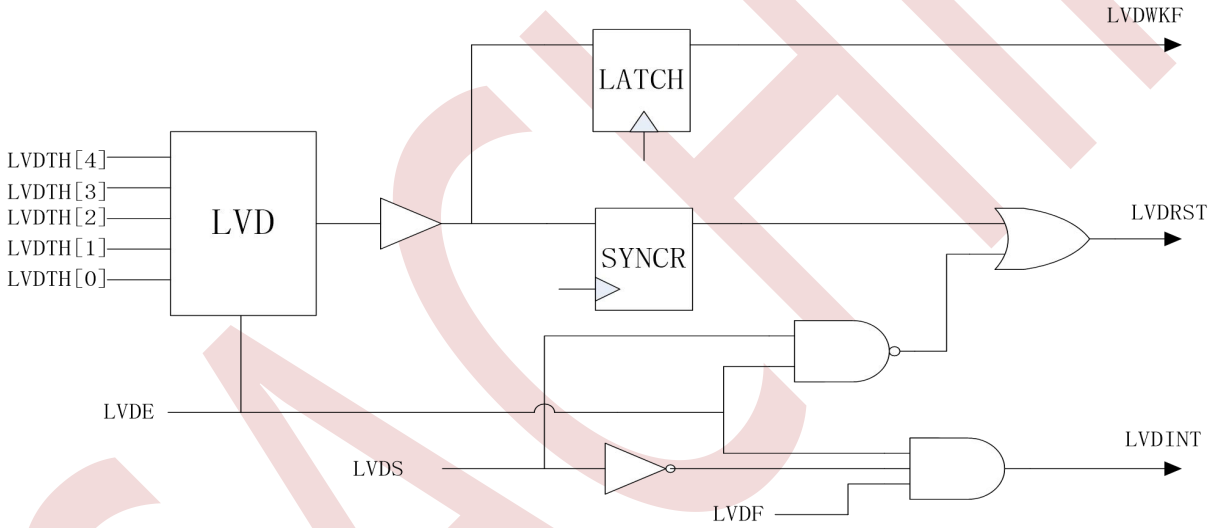


Figure 22-1-1 LVD Schematic

22.2 Function Description

The LVD function is enabled via the LVDE bit, while the detected voltage is set via the LVDTH bit field. When the chip VDD is less than the set voltage, the flag LVDF bit generated by the LVD function will set to 1. If LVDS = 0, an LVD interrupt will be generated, and if LVDS = 1, a reset will be generated. It should be noted that after LVD reset is generated, the circuit of LVD itself will not be reset, and the register LVDCON will keep the previous state. Therefore, if VDD continues to fall below the set voltage after LVD reset is generated, the chip will remain in the reset state. Similarly, when the LVD interrupt is generated, the LVD interrupt will be generated repeatedly if VDD continues to fall below the set voltage.

22.3 Register Description

Table 22-3-1 Register LVDCON

E8H	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 generate flag ,cleared when 1 is written to it
1~0	LVDTH	LVD trigger level selection bit field 000:2.2V 001: 2.5V 010:2.7V 011:3.0V 100:3.3V 101:3.6V 110:3.9V 111:4.2V

22.4 LVD control routines

LVD interrupt routines

For example, if LVD is set to interrupt mode and the detection voltage is 3V, the program is as follows：

```
---------------------------------------------------------------------------------------
#define LVDE(N)          (N<<7)    //N=0~1
#define LVDS_reset       (1<<6)
#define LVDS_int         (0<<6)
#define LVDF             (1<<5)
#define LVDTH_3V         3
void LVD_init(void)
{
    LVDCON = LVDE(1) | LVDS_int | LVDF | LVDTH_3V;// Set LVD enable, LVD is interrupt mode, detection
                                                  // voltage is 3V
    INT4EN = 1;   // INT4 interrupt enable
    EA = 1;       // Turn on total interruption
}
void INT4_ISR (void) interrupt 6
{
    if(LVDCON & LVDF)
    {
        LVDCON |= LVDF;       // Clear the LVD interrupt flag
    }
}
---------------------------------------------------------------------------------------
```

LVD Reset Routine

For example, setting LVD to reset mode and detecting a voltage of 3V, the program is as follows.

```
---------------------------------------------------------------------------------------
#define LVDE(N)          (N<<7)    //N=0~1
#define LVDS_reset       (1<<6)
#define LVDS_int         (0<<6)
#define LVDF             (1<<5)
#define LVDTH_3V         3
void LVD_init(void)
{
    LVDCON = LVDE(1) | LVDS_reset | LVDF | LVDTH_3V;// Set LVD enable, LVD is interrupt mode, detection
                                                    // voltage is 3V
}
---------------------------------------------------------------------------------------
```

23 Digital to analog converter (DAC)

23.1 Function Introduction

This module contains 1 DA output (P0.7), which can be set to 64 voltage levels. The main purpose of designing this module is to use the CA51F152S3 chip as a touch chip, which can output different voltages through DA to represent different key value information, for the main control chip to detect through ADC.

DAC is enabled through the DAEN bit in DACON. After setting DAEN=1 and selecting the DAC function for the corresponding pin, the DAC module can output the corresponding voltage on the corresponding pin based on the user configured DAVS value.

23.2 Register Description

Table 23-2-1 Register DACON

BEH	7	6	5	4	3	2	1	0
DACON	DAEN	-	DAVS					
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

Bit Number	Bit Symbol	Description
7	DAEN	DAC channel enable signal, 1 enable, 0 not enable
6	-	-
5~0	DAVS	DAC output voltage configuration register The output voltage of DAC is: (DAVS+1) ÷ 64) × Vdd

23.3 DAC control routines

The program is as follows:

```
-------------------------------------------------------------------------------------------
// DACON Definition
#define DAEN(N)          (N<<7)
#define DAVS(N)          N                //N=0-63
void DAC_init(void)
{
    P07F = 0;
    DACON = DAEN(1) | DAVS(32);
}
```

24 Program download and simulation

24.1 Program Download

The CA51F152S3 series chips mainly use ISP to download programs, and the chips are connected to the download tool through the I2C interface. The upgrade download interfaces for the chips are P0.0 (I2C SDA) and P0.1 (I2C SCL).

For more details on the program download steps, please refer to the "CACHIP Development Download Tool Usage Description".

24.2 Online Simulation

The CA51F152S3 series chip supports online simulation, and communication between the chip and the simulator is carried out through the IIC interface. The I2C interface of the chip is P0.0 (IIC-SDA) and P0.1 (IIC-SCL). It should be noted that due to the IIC communication between the chip and the simulator, the I2C interface pins connected to the simulator cannot be set to other functions, and the IIC function cannot be used in the application program, otherwise it will not be able to enter simulation mode. In addition, since the communication speed of I2C is determined by the main clock, the application cannot set the main clock to a low-speed clock or enter power-saving mode, otherwise it will affect the communication between the chip and the simulator.

When the chip enters simulation mode, the TSMODE bit (PCON [2]) is set to 1, and the application program can determine whether to switch to low-speed clock or enter power-saving mode by judging the status of this bit.

For more details on simulation functions, please refer to the relevant documentation of the simulator.

25 Electrical Characteristics

25.1 Limit parameters

Parameters	Minimum value	Maximum value	Unit
DC supply voltage	-0.3	6	V
I/O pin input voltage	-0.3	VDD+0.3	V
Working environment temperature	-40	85	℃
Storage temperature	-55	125	℃
CPU operating frequency	-	16	MHz

Note: Exceeding the "limit parameter" range may cause damage to the chip, and it is not possible to anticipate the working state of the chip outside the above range.

25.2 DC electrical characteristics

DC electrical characteristics (VDD=2.3-5.5V, TA=25 ℃, unless otherwise described)

Chip Parameters	Symbols	Operating Voltage	Minimum value	Typical values	Maximum value	Unit	Test conditions
Operating current	Iop1	VDD=2.5V	-	3.04	-	mA	The system clock is IRCH (16MHz), all other clocks are turned off, all output pins are unloaded, all Digital Input pins are not floating, all peripherals are turned off, and the CPU executes NOP instructions
		VDD=3.3V	-	3.69	-		
		VDD=5V	-	5.85	-		
	Iop3	VDD=2.5V	-	30	-	uA	The system clock is IRCL (128kHZ), all other clocks are turned off, all output pins are unloaded, all Digital Input pins are not floating, all peripherals are turned off, and the CPU executes NOP instructions
		VDD=3.3V	-	41	-		
		VDD=5V	-	58	-		
STOP mode current	I_{stp}	VDD=2.5V	-	4.6	-	uA	All clocks are turned off, all output pins are unloaded, all Digital Input pins are not floating, all peripherals are turned off, Flash enters sleep mode, and CPU enters STOP mode.
		VDD=3.3V	-	4.7	-		
		VDD=5V	-	4.8	-		

IDLE mode current	I_{idl1}	VDD=2.5V	-	1.48	-	mA	The system clock is set to IRCH (16MHz), other clocks are turned off, all output pins are unloaded, all Digital Input pins are not floating, all peripherals are turned off, Flash enters sleep mode, and CPU enters IDLE mode.
		VDD=3.3V	-	1.96	-		
		VDD=5V	-	3.14	-		
	I_{idl2}	VDD=2.5V	-	17	-	uA	The system clock is set to IRCL (128KHz), other clocks are turned off, all output pins are unloaded, all Digital Input pins are not floating, all peripherals are turned off, and the CPU enters IDLE mode.
		VDD=3.3V	-	20	-		
		VDD=5V	-	28	-		
IO port input high voltage (Smit mode on)	V_{hi1}	VDD=2.5V	1.17	-	2.5	V	-
		VDD=3.3V	1.75	-	3.3		
		VDD=5V	2.60	-	5		
IO port input high voltage (Smit mode off)	V_{hi2}	VDD=2.5V	-	0.5*VDD	VDD	V	-
		VDD=3.3V					
		VDD=5V					
IO port input low voltage (Smit mode on)	V_{lo1}	VDD=2.5V	0	-	1.04	V	-
		VDD=3.3V	0	-	1.32		
		VDD=5V	0	-	1.75		
IO port input low voltage (smit mode off)	V_{lo2}	VDD=2.5V	0	0.5*VDD	-	V	-
		VDD=3.3V					
		VDD=5V					
IO port push current	I_{pu}	VDD=3.3V	-	4mA @DRV=0 8mA @DRV=1	-	mA	IO set to push-pull output mode, with optional two-stage driving capability (DRV), Voh=0.9 * VDD
		VDD=5V	-	10mA @DRV=0 18mA @DRV=1	-		
IO port sink current	I_{ol}	VDD=3.3V	-	11mA @DRV=0 24mA @DRV=1	-	mA	IO set to push-pull output mode, with optional two-stage driving capability (DRV), Vol=0.1 * VDD
		VDD=5V	-	20mA @DRV=0 40mA @DRV=1	-		
IO port pull-down resistor	R_{d1}	VDD=2.5~5.5V	-	30	-	KΩ	-
IO port pull-up resistor	R_{u1}	VDD=2.5~5.5V	-	30	-	KΩ	-

Note: The above parameters are typical chip test results randomly selected for reference.

25.3 ESD/EFT Characteristics

symbol	mode	Conditions	package	Maximum value	Unit
V_{ESD}	electrostatic discharge (Human discharge mode HBM)	TA=+25℃	SOP16	7500	V
	electrostatic discharge (Component discharge mode CDM)			1000	V

symbol	Conditions		package	pass through	Unit
EFT	F_{sys} = 16MHz / UART communicate	TA=+25℃	SOP16	±4000	V

25.4 AC Electrical Characteristics

AC electrical characteristics (VDD=2.3-5.5V, TA=25°C, unless otherwise specified)

Chip Parameters	Symbol	Minimum value	Typical values	Maximum value	Unit	Conditions
Internal low speed clock (IRCL) start time	Trc1	-	50	-	us	IRCL frequency is 128Khz
Internal high speed clock (IRCH) start time	Trc2	-	10	-	us	IRCH frequency is 16MHz

Note: VDD=5V,TA=25 ℃,internal high speed clock factory frequency is 16MHz,accuracy is ±2%.

25.5 Internal RC clock temperature characteristics

- **IRCH temperature characteristic**

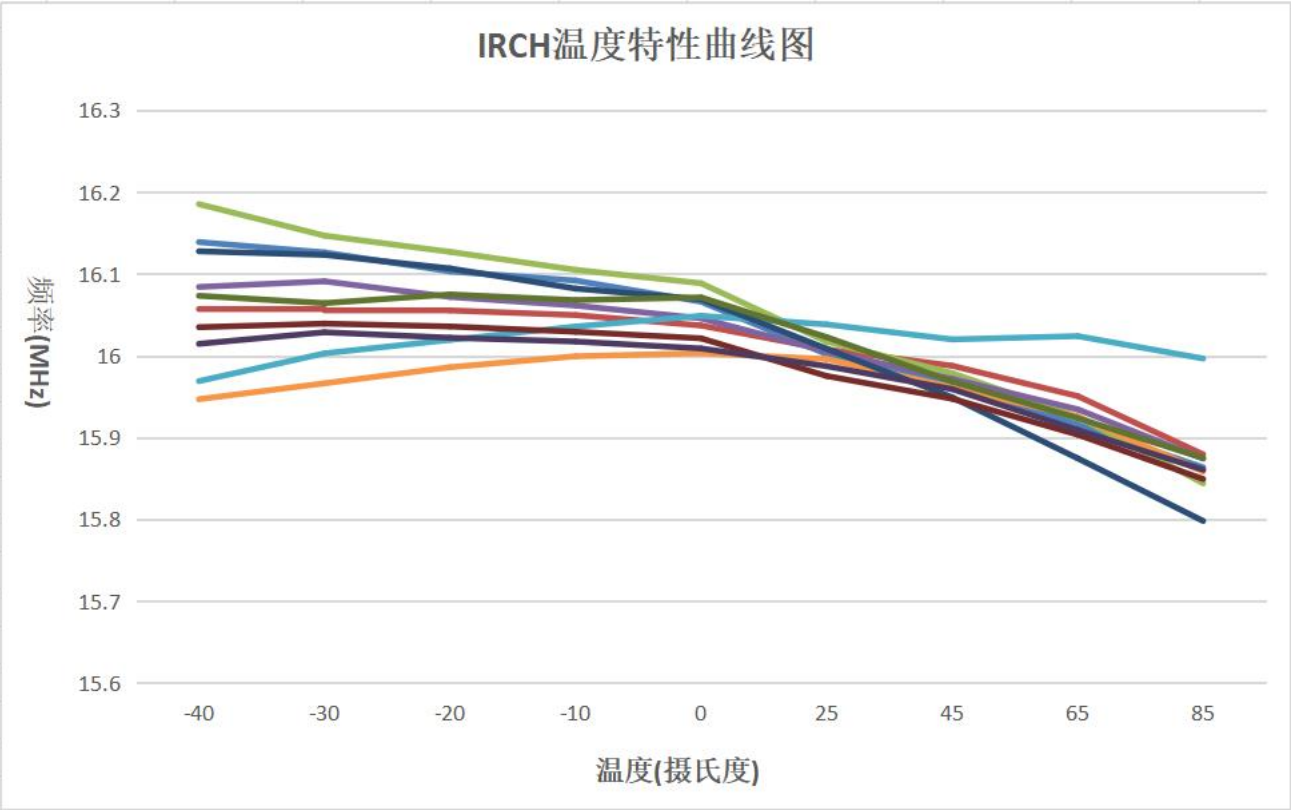
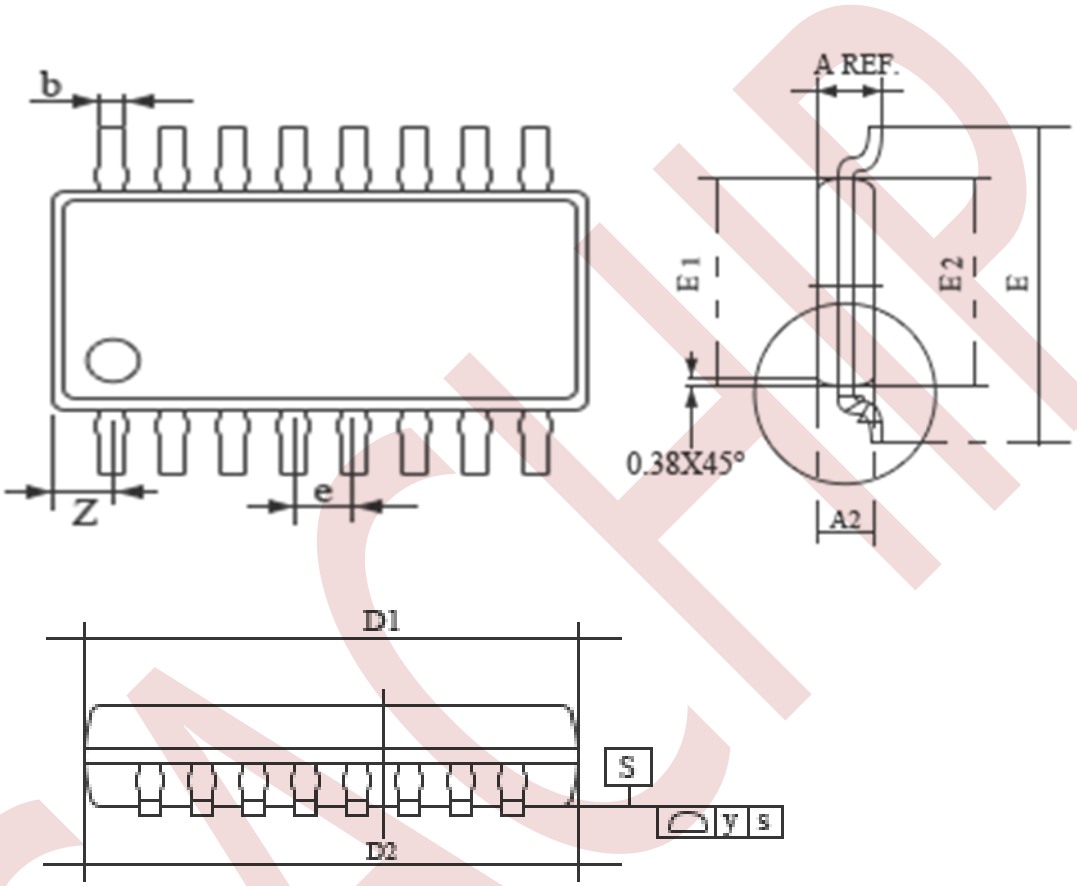


Figure 25-5-1 IRCH temperature characteristic curve

Note: The above graphic data is a randomly selected portion of chip test data, and the data is for reference only.

26 Package Type

Package form:SOP16



Serial number	Minimum value (mm)	Standard value (mm)	Maximum value (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	————

27 Appendix

Appendix 1 Instruction Set Quick Reference Table

Instruction	Description	Description	Periodicity
Data transfer command			
MOV A,Rn	The register contents are fed to the accumulator	(A) ← (Rn)	1
MOV A,direct	The data in the direct address cell is fed to the accumulator	(A) ← (direct)	1
MOV A,@Ri	Data from the indirect RAM is fed to the accumulator	(A) ← ((Ri))	1
MOV A,#data8	8-bit immediate number fed to accumulator	(A) ← #data	1
MOV Rn,A	The contents of the accumulator are fed into the register	(Rn) ← (A)	1
MOV Rn,direct	The data in the direct address cell is fed into the register	(Rn) ← (direct)	2
MOV Rn,#data8	8-bit immediate number sent to register	(Rn) ← #data	1
MOV direct,A	The contents of the accumulator are fed to the direct address unit	(direct) ← (A)	1
MOV direct,Rn	The register contents are fed to the direct address unit	(direct) ← (Rn)	2
MOV direct,direct	Data in the direct address unit is fed to the direct address unit	(direct) ← (direct)	2
MOV direct,@Ri	Data from the indirect RAM is fed to the direct address cell	(direct) ← ((Ri))	2
MOV direct,#data8	8-bit immediate number sent to direct address unit	(direct) ← #data	2
MOV @Ri,A	The contents of the accumulator are fed to the indirect RAM cell	((Ri)) ← (A)	1
MOV @Ri,direct	Data from the direct address cell is fed to the indirect RAM cell	((Ri)) ← (direct)	2
MOV @Ri,#data8	8-bit immediate number sent to indirect RAM cell	((Ri)) ← #data	1
MOV DPTR,#data16	16-bit immediate address fed into address register	(DPTR) ← #data116	2
MOV A,@A+DPTR	The data in the DPTR-based variable address addressing unit is fed to the accumulator	(A) ← ((A)) + (DPTR)	2
MOV A,@A+PC	The data in the PC-based address-variable addressing unit is fed to the accumulator	(PC) ← (PC) + 1 (A) ← ((A) + (PC))	2

MOVX A,@Ri	External RAM (8-bit address) is fed to the accumulator	(A) ← ((Ri))	2
MOVX A,@DPTR	External RAM (16-bit address) is fed to the accumulator	(A) ← ((DPTR))	2
MOVX @Ri,A	Accumulator feed to external RAM (8-bit address)	((Ri)) ← (A)	2
MOVX @DPTR,A	Accumulator feed to external RAM (16-bit address)	(DPTR) ← (A)	2
PUSH direct	Data in the direct address cell is pressed onto the stack	(SP) ← (SP) + 1 ((SP)) ← (direct)	2
POP DIRECT	Data in the stack is popped to the direct address unit	(direct) ← ((SP)) (SP) ← (SP) - 1	2
XCH A,Rn	Register and accumulator exchange	(A) ↔ (Rn)	1
XCH A,direct	Direct address unit exchange with accumulator	(A) ↔ (direct)	1
XCH A,@Ri	Indirect RAM to accumulator swap	(A) ↔ ((Ri))	1
XCHD A,@Ri	Indirect RAM with accumulator for low half-byte swap	(A.3,...,A.0) ↔ ((Ri).3,...,(Ri).0)	1
SWAP A	Accumulator half-byte swap	(A.3,...,A.0) ↔ (A.7,...,A.4)	1
Arithmetic operation class instruction			
ADD A, Rn	Register contents are added to the accumulator	(A) ← (A) + (Rn)	1
ADD A, direct	Direct address unit added to accumulator	(A) ← (A) + (direct)	1
ADD A, @Ri	Indirect RAM content added to accumulator	(A) ← (A) + ((Ri))	1
ADD A, #data8	8-bit immediate number added to accumulator	(A) ← (A) + #data	1
ADDC A, Rn	Register contents are added to the accumulator with rounding	(A) ← (A) + (C) + (Rn)	1
ADDC A, direct	Direct address unit with advance added to accumulator	(A) ← (A) + (C) + (direct)	1
ADDC A, @Ri	Indirect RAM content with rounding added to accumulator	(A) ← (A) + (C) + ((Ri))	1
ADDC A, #data8	8-bit immediate numbers with rounding added to accumulator	(A) ← (A) + (C) + #data	1
SUBB A, Rn	Accumulator with debit minus register content	(A) ← (A) - (C) - (Rn)	1
SUBB A, direct	Accumulator with debit minus direct address unit	(A) ← (A) - (C) - (direct)	1
SUBB A, @Ri	Accumulator with debit minus indirect RAM content	(A) ← (A) - (C) - ((Ri))	1

SUBB A, #data8	Accumulator with debit minus 8-bit immediate	(A) ← (A) - (C) - #data	1
INC A	Totalizer plus 1	(A) ← (A) + 1	1
INC Rn	Register plus 1	(Rn) ← (Rn) + 1	1
INC direct	Add 1 to the contents of the direct address unit	(direct) ← (direct) + 1	1
INC @Ri	Indirect RAM content plus 1	((Ri)) ← ((Ri)) + 1	1
INC DPTR	DPTR plus 1	(DPTR) ← (DPTR) + 1	2
DEC A	Totalizer minus 1	(A) ← (A) – 1	1
DEC Rn	Register minus 1	(Rn) ← (Rn) - 1	1
DEC direct	Direct address cell content minus 1	(direct) ← (direct) - 1	1
DEC @Ri	Indirect RAM content minus 1	((Ri)) ← ((Ri)) - 1	1
MUL AB	A multiplied by B	temp16 ← (A) X (B) (A)←(temp.7,temp.6,...,temp.0) (B)←(temp.15,temp.14,...,temp.8)	4
DIV AB	A divided by B	QUO ← (A) / (B) .....REM (A) ← QUO (B) ← REM	4

DA A	Accumulator for decimal conversion	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
		(A) ←
			1

		(temp.7,...,temp.0) IF (temp16) > 0xFF THEN CY ← 1	
Logic operation class instruction			
ANL A, Rn	The accumulator is ANDed with the register	(A) ← (A) & (Rn)	1
ANL A, direct	The accumulator is ANDed with the direct address unit	(A) ← (A) & (direct)	1
ANL A, @Ri	The accumulator is ANDed with the indirect RAM content	(A) ← (A) & ((Ri))	1
ANL A, #data8	The accumulator is ANDed with 8-bit immediate data	(A) ← (A) & #data	1
ANL direct, A	The direct address unit and the accumulator are "AND''	(direct) ← (direct) & (A)	1
ANL direct, #data8	Direct address unit and 8-bit immediate data "AND"	(direct) ← (direct) & #data	2
ORL A, Rn	The accumulator is ORed with the register	(A) ← (A) | (Rn)	1
ORL A, direct	The accumulator is ORed with the direct address unit	(A) ← (A) | (direct)	1
ORL A, @Ri	Accumulator and indirect RAM content "OR"	(A) ← (A) | ((Ri))	1
ORL A, #data8	The accumulator is ORed with 8-bit immediate data	(A) ← (A) | #data	1
ORL direct, A	The direct address unit is ORed with the accumulator	(direct) ← (direct) | (A)	1
ORL direct, #data8	Direct address unit and 8-bit immediate data "OR"	(direct) ← (direct) | #data	2
XRL A, Rn	"Exclusive OR" between accumulator and register	(A) ← (A) ^ (Rn)	1
XRL A, direct	"Exclusive OR" between the accumulator and the direct address unit	(A) ← (A) ^ (direct)	1
XRL A, @Ri	"Exclusive OR" between the accumulator and the indirect RAM content	(A) ← (A) ^ ((Ri))	1
XRL A, #data8	"Exclusive OR" between accumulator and 8-bit immediate data	(A) ← (A) ^ #data	1
XRL direct, A	"Exclusive OR" between the direct address unit and the accumulator	(direct) ← (direct) ^ (A)	1
XRL direct, #data8	"Exclusive OR" between direct address unit and 8-bit immediate data	(direct) ← (direct) ^ #data	2
CLR A	Accumulator is cleared to 0	(A) ← 0	1
CPL A	Accumulator negation	(A) ← /(A)	1
RL A	Accumulator rotate left	(A) ←	1

		(A.6,A.5,...,A.0,A.7)	
RLC A	Accumulator rotates left with carry	C ← A.7 (A) ← (A.6,A.5,...,A.0,C)	1
RR A	The accumulator rotates right	(A) ← (A.0,A.7,...,A.2,A.1)	1
RRC A	Accumulator with carry cycle shift right	C ← A.0 (A) ← (C,A.7,...,A.2,A.1)	1
Control transfer instructions			
ACALL addr11	Absolute short call subroutine	(PC) ← (PC) + 2 (SP) ← (SP) + 1 ((SP)) ← (PC7-0) (SP) ← (SP) + 1 ((SP)) ← (PC15-8) (PC10-0) ← page address	2
LACLL addr16	Long call subroutine	(PC) ← (PC) + 3 (SP) ← (SP) + 1 ((SP)) ← (PC7-0) ((SP)) ← (PC15-8) (PC) ←addr15-0	2
RET	The subroutine returns	(PC15-8) ← ((SP)) (SP) ← (SP) - 1 (PC7-0) ← ((SP)) (SP) ← (SP) - 1	2
RETI	Interrupt return	(PC15-8) ← ((SP)) (SP) ← (SP) - 1 (PC7-0) ← ((SP)) (SP) ← (SP) - 1	2
AJMP addr11	Absolute short transfer	(PC) ← (PC) + 2 (PC10-0) ← page address	2
LJMP addr16	Long transfer	(PC) ← (PC) + 3 (SP) ← (SP) + 1 ((SP)) ← (PC7-0)	2

		(SP) ← (SP) + 1 ((SP)) ← (PC15-8) (PC10-0) ←addr15-0	
SJMP rel	Relative Transfer	(PC) ← (PC) + 2 (PC) ← (PC) + rel	2
JMP @A+DPTR	Indirect transfer relative to DPTR	(PC) ← (A) + (DPTR)	2
JZ rel	Accumulator for zero transfer	(PC) ← (PC) + 2 IF (A) = 0 THEN (PC) ← (PC) + rel	2
JNZ rel	Accumulator non-zero transfer	(PC) ← (PC) + 2 IF (A) <> 0 THEN (PC) ← (PC) + rel	2

CJNE A, direct, rel	Compare the accumulator with the direct address unit, transfer 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	Accumulator compares with 8-bit immediate numbers, transfer if unequal	(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 register with 8-bit immediate number, transfer 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	Indirect RAM unit, unequal transfer	(PC) ← (PC) + 3
		IF ((Ri)) <> data
		THEN
		(PC) ← (PC) +
		relative offset
		IF ((Ri)) < data
		THEN
		(C) ← 1
		ELSE
		(C) ← 0
			2

DJNZ Rn, rel	Register minus 1, non-zero transfer	(PC) ← (PC) + 2 (Rn) ← (Rn) - 1 IF (Rn) <> 0 THEN (PC) ← (PC) + rel	2
DJNZ direct, rel	Direct address unit minus 1, non-zero transfer	(PC) ← (PC) + 2 (direct) ← (direct) - 1 IF (direct) <> 0 THEN (PC) ← (PC) + rel	2
NOP	No operation	(PC) ← (PC) + 1	1
Boolean variable operation instructions			
CLR C	Clear carry bit	(C) ← 0	1
CLR bit	Clear direct address bit	(bit) ← 0	1
SETB C	Set carry bit	(C) ← 1	1
SETB bit	Direct address bit	(bit) ← 1	1
CPL C	Carry negation	(C) ← /(C)	1
CPL bit	Direct address bit negation	(bit) ← /(bit)	1
ANL C, bit	Carry bit and direct address bit phase "AND"	(C) ← (C) & (bit)	2
ANL C, /bit	Carry bit and the inverse code of the direct address bit are "ANDed"	(C) ← (C) & /(bit)	2
ORL C, bit	Carry bit and direct address bit phase "OR"	(C) ← (C) | (bit)	2
ORL C, /bit	Carry bit and the inverse code of the direct address bit are ORed	(C) ← (C) | /(bit)	2
MOV C, bit	Direct address bit into carry bit	(C) ← (bit)	1

MOV bit, C	Carry bit into direct address bit	(bit) ← (C)	2
JC rel	If the carry bit is 1, then transfer (CY=O does not transfer, =1 transfer)	(PC) ← (PC) + 2 IF (C) = 1 THEN (PC) ← (PC) + rel	2
JNC rel	If the carry bit is 0, then branch	(PC) ← (PC) + 2 IF (C) = 0 THEN (PC) ← (PC) + rel	2
JB bit, rel	Direct address bit is 1, then transfer	(PC) ← (PC) + 3 IF (bit) = 1 THEN (PC) ← (PC) + rel	2
JNB bit, rel	Direct address bit is 0 then transfer	(PC) ← (PC) + 3 IF (bit) = 0 THEN (PC) ← (PC) + rel	2
JBC bit, rel	If the direct address bit is 1, it will be transferred, and the bit will be cleared.	(PC) ← (PC) + 3 IF (bit) = 1 THEN (bit) ← 0 (PC) ← (PC) + rel	2

Pseudo-instruction	
ORG	Set program start address
END	Mark the end of source code
EQU	Define constant
SET	Define an integer number
DATA	Set value for data address
BYTE	Assign value to byte type symbol
WROD	Assign value to word type symbol
BIT	Name the bit address
ALTNAME	Replace reserved words with custom names
DB	Load byte data into a continuous storage area
DW	Load font data to a continuous memory area
DS	Reserve a continuous storage area or load the specified byte
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 assembled, the list is not generated if the condition is false