Cmsemicon®

CMS8S006 Series Products

User Manual

Enhanced flash memory 1T 8051 microcontrollers

Rev. 1.0.0

Please note the following CMS IP policy

*China Micro Semicon Co., Ltd. (hereinafter referred to as the Company) has applied for patents and holds absolute legal rights and interests. The patent rights associated with the Company's MCUs or other products have not been authorized for use, and any company, organization, or individual who infringes the Company's patent rights through improper means will be subject to all possible legal actions taken by the Company to curb the infringement and to recover any damages suffered by the Company as a result of the infringement or any illegal benefits obtained by the infringer.

*The name and logo of Cmsemicon are registered trademarks of the Company.

*The Company reserves the right to further explain the reliability, functionality and design improvements of the products in the data sheet. All electrical parameters mentioned in the datasheet are typical values designed for standard room temperature conditions. The Company is not responsible for the use of the Specification Contents. The applications mentioned herein are for illustrative purposes only and the Company does not warrant and does not represent that these applications can be applied without further modification, nor does it recommend that its products be used in places that may cause harm to persons due to malfunction or other reasons. The Company's products are not authorized for use as critical components in lifesaving, life-sustaining devices or systems. The Company reserves the right to modify the products without prior notice. For the latest information, please visit the official website at www.mcu.com.cn.

Table of Contents

1. Central Processing Unit (CPU)

This series features an 8-bit microcontroller based on the 8051 architecture. The CPU is the core component of the microcontroller and consists of an arithmetic logic unit, a control unit, and a set of dedicated registers. The arithmetic logic unit primarily performs arithmetic operations, logical operations, bit manipulation, and data transfer operations. The control unit is responsible for decoding instructions and generating various control signals. The dedicated register set is mainly used to represent the memory address of the instruction currently being executed, store operands, and indicate the status after instruction execution. The dedicated register set includes the accumulator (ACC), general-purpose register (B), stack pointer (SP), data pointer (DPTR), program status word (PSW), and program counter (PC).

1.1 Reset vector (0000H)

The microcontroller has a word-length system reset vector at address 0000H. After a reset, the program will start executing from 0000H, and all system registers will be restored to their default values. The following example demonstrates how to define the reset vector in FLASH.

Example: defining the reset vector

```
            ORG         0000H           ;System reset vector
            LJMP        START
            ORG         0010H           ;User program start
      START:
            …                           ;User program
            …
            END                         ;Program end
```

1.2 BOOT partition

The BOOT space has a maximum size of 4KB*8Bit, and its size is allocated by user-configurable registers.

When the chip is powered on, if the program is to start from the BOOT area, the CONFIG register must be configured to define the address space allocation (details can be found in the corresponding data sheet). If not configured, the program will start from the APROM area.

Example for a 2K BOOT area: When configuring BOOT_2K in the CONFIG register, the program will start executing from address 7800H upon powering on. If the program needs to switch between the BOOT area and the APROM area, the BOOT control register BOOTCON must be written with values 0xAA or 0x55 (refer to the register description for details). After this, a software reset or a watchdog reset must be executed.

Upon power-on reset, external reset, or low-voltage reset, the default value of BOOTCON is 0x00. Note that a software reset and watchdog reset do not clear this register.

BOOT control register (BOOTCON)

F691H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BOOTCON	D7	D6	D5	D4	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	D<7:0>:	BOOT region control bit (it can only be written when the chip is configured to BOOT_2K/BOOT_4K/BOOT_8K/BOOT_16K)
	0x55=	To switch from the APROM region to the BOOT region, write 0x55 to it, then perform a software reset or trigger a watchdog reset.
	0xAA=	To switch from the BOOT region to the APROM region, write 0xAA to it, then perform a software reset or trigger a watchdog reset.
	Other values=	Invalid.

For example, if the chip is powered on and starts from the BOOT region, and you want to switch to the APROM region using a software reset, the configuration is as follows:

1) The BOOTCON register needs to be written with 0xAA:

```
MOV    DPTR,# BOOTCON
MOV    A,#0AAH
MOVX  @DPTR,A
```

2) Execute a software reset:

```
MOV    TA,#0AAH
MOV    TA,#055H
MOV    WDCON,#080H
```

For another example, using a software reset to switch from the APROM region to the BOOT region, the configuration is as follows:

1) The BOOTCON register needs to be written with 0x55:

```
MOV    DPTR, # BOOTCON
MOV    A,#055H
MOVX  @DPTR,A
```

2) Execute a software reset:

```
MOV    TA,#0AAH
MOV    TA,#055H
MOV    WDCON,#080H
```

1.3 Accumulator (ACC)

The ALU is an 8-bit wide Arithmetic Logic Unit, and all mathematical and logical operations in the MCU are completed through it. It can perform addition, subtraction, shifting, and logical operations; the ALU also controls the status bits (in the PSW status register) to indicate the status of the operation results.

The ACC register is an 8-bit register where the results of ALU operations can be stored.

1.4 B register (B)

The B register is used with multiplication and division instructions. If not used for multiplication or division, it can also serve as a general-purpose register.

1.5 Stack pointer register (SP)

The SP register points to the address of the stack, with a default value of 0x07 after reset, indicating that the stack area starts from RAM address 08H. The value of SP can be modified; for example, if the stack area is set to start at 0xC0, then after system reset, the SP value needs to be set to 0xBF.

Operations that affect SP include: PUSH, LCALL, ACALL, POP, RET, RETI, and entering interrupts.

The PUSH instruction occupies one byte in the stack, while LCALL, ACALL, and interrupts occupy two bytes. The POP instruction releases one byte, and RET/RETI instructions release two bytes.

Using the PUSH instruction automatically saves the current value of the operated register into RAM.

1.6 Data pointer registers (DPTR0/DPTR1)

The data pointers are mainly used in MOVX and MOVC instructions, serving to locate addresses in XRAM and ROM. The chip has two data pointer registers, DPTR0 and DPTR1, which are selected via the DPS register.

Each set of pointers includes two 8-bit registers: DPTR0 = {DPH0, DPL0}; DPTR1 = {DPH1, DPL1}.

For example, assembly code to operate on XRAM is as follows:

```
MOV         DPTR,#0001H
MOV         A,#5AH
MOVX        @DPTR,A             ;Write the data in A to XRAM address 0001H
```

1.7 Data pointer selection register (DPS)

Data pointer selection register DPS

0x86	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
DPS	ID1	ID0	TSL	AU	--	--	--	SEL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 ID<1:0>: Self-plus/minus select bit.
00= DPTR0 plus 1 or DPTR1 minus 1
01= DPTR0 minus 1 or DPTR1 plus 1
10= DPTR0 plus 1 or DPTR1 minus 1
11= DPTR0 minus 1 or DPTR1 minus 1

Bit5 TSL: Toggle select enable bit
1= After executing DPTR instructions, the SEL bit will automatically toggle.
0= DPTR-related instructions do not affect the SEL bit

Bit4 AU: Self-plus/minus enable bit
1= Enable the MOVX @DPTR or MOVC @DPTR instruction to run and then perform a self-subtracting/self-adding operation (determined by ID1-ID0).
0= DPTR-related instructions do not affect the SEL bit.

Bit3~Bit1 --: Reserved, set to 0.

Bit0 SEL: Data pointer selection bit
1= Select DPTR1
0= Select DPTR0

1.8 Program status register (PSW)

Program status register PSW

0xD0	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PSW	CY	AC	F0	RS1	RS0	OV	--	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Reset value	0	0	0	0	0	0	0	0

Bit7	CY:	Carry flag
	1=	Carry occurred
	0=	No carry
Bit6	AC:	Auxiliary carry flag bit (half carry flag bit)
	1=	Carry occurred
	0=	No carry
Bit5	F0:	General flag bit
Bit4~Bit3	RS<1:0>:	Register BANK selection bit
	00=	Select Bank0
	01=	Select Bank1
	10=	Select Bank2
	11=	Select Bank3
Bit2	OV:	Overflow flag bit
	1=	Overflow occurred in arithmetic or logical operations
	0=	No overflow in arithmetic or logical operations
Bit1	--	Reserved, set to 0.
Bit0	P:	Parity bit
	1=	The highest bit resulted in a carry
	0=	The highest bit did not result in a carry

1.9 Program counter (PC)

The Program Counter (PC) controls the execution sequence of instructions in the program memory (FLASH). It can address the entire range of FLASH memory. After fetching the instruction code, the Program Counter (PC) automatically increments by one, pointing to the address of the next instruction code. However, when executing operations such as jumps, conditional jumps, subroutine calls, initialization resets, interrupts, interrupt returns, and subroutine returns, the PC will load the address related to the instruction instead of the address of the next instruction.

When a conditional jump instruction is encountered and the jump condition is met, the next instruction read during the execution of the current instruction will be discarded, and a dummy instruction cycle will be inserted before the correct instruction can be fetched. Conversely, if the condition is not met, the next instruction will be executed sequentially.

1.10 Timing access register (TA)

Timing access register TA

0x96	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TA	TA7	TA6	TA5	TA4	TA3	TA2	TA1	TA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TA<7:0>: Timing access control bit

Certain protected registers must be written to only after performing the following operations on TA:

```
MOV    TA, #0AAH
MOV    TA, #055H
```

No other instructions can be inserted between these operations. To modify again, this sequence must be executed once more.

Protected registers: WDCON, CLKDIV, SCKSEL, MLOCK, WWCON0, WWCON1, WWCMPD.

2. Memory and Register Mapping

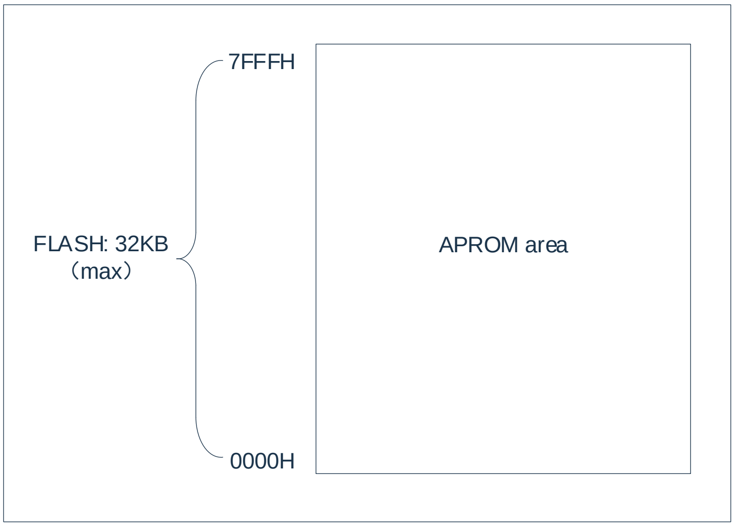
This series of microcontrollers includes the following types of memory:

- Up to 32KB of FLASH program memory (APROM area)
- Up to 1KB of non-volatile data memory (Data FLASH)
- Up to 256B of general internal data memory (RAM)
- Up to 2KB of general external data memory (XRAM)
- Special Function Registers (SFR) in BANK0 and BANK1
- External Special Function Registers (XSFR)

2.1 Program memory (APROM)

The program memory APROM is used to store source programs and table data, with the Program Counter (PC) serving as the address pointer. The PC is a 16-bit program counter, allowing it to address a memory space of 32KB.

The allocation structure of the FLASH space is shown in the diagram below:

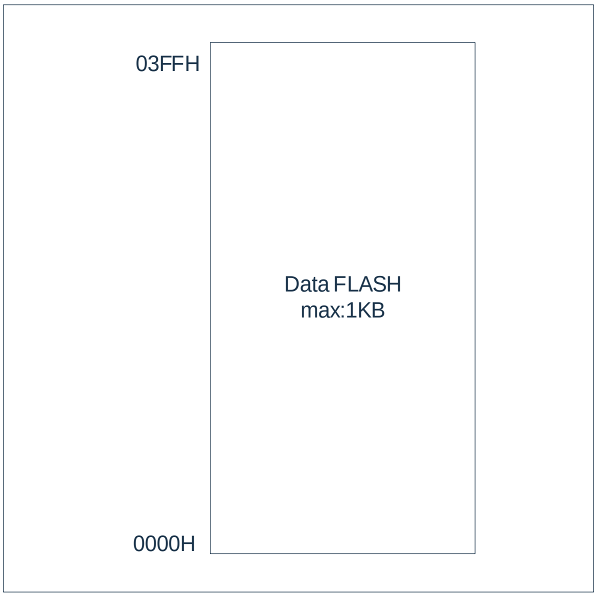


After the chip resets, the CPU starts executing from address 0000H. Each interrupt is assigned a fixed address in the program memory, and the interrupt causes the CPU to jump to that address to begin executing the service program.

For example, External Interrupt 1 is assigned the address 0013H. If External Interrupt 1 is used, its service program must start from the 0013H location. If the interrupt is not utilized, its service address can be used as a regular program storage address.

2.2 Non-volatile data memory (Data FLASH)

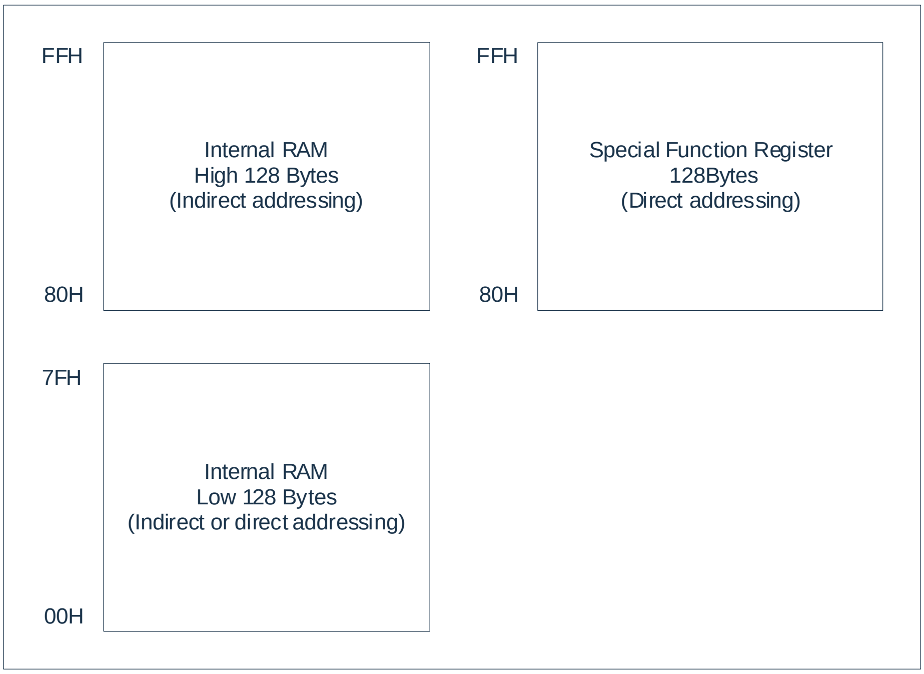
The non-volatile data memory, Data FLASH, can be used to store important data such as constant data, calibration data, and security-related information. Data stored in this area retains its integrity even during a power loss or unexpected power interruption. The allocation structure of the Data FLASH space is illustrated in the diagram below:



The read, write, and erase operations of the Data FLASH memory are implemented through the FLASH control interface.

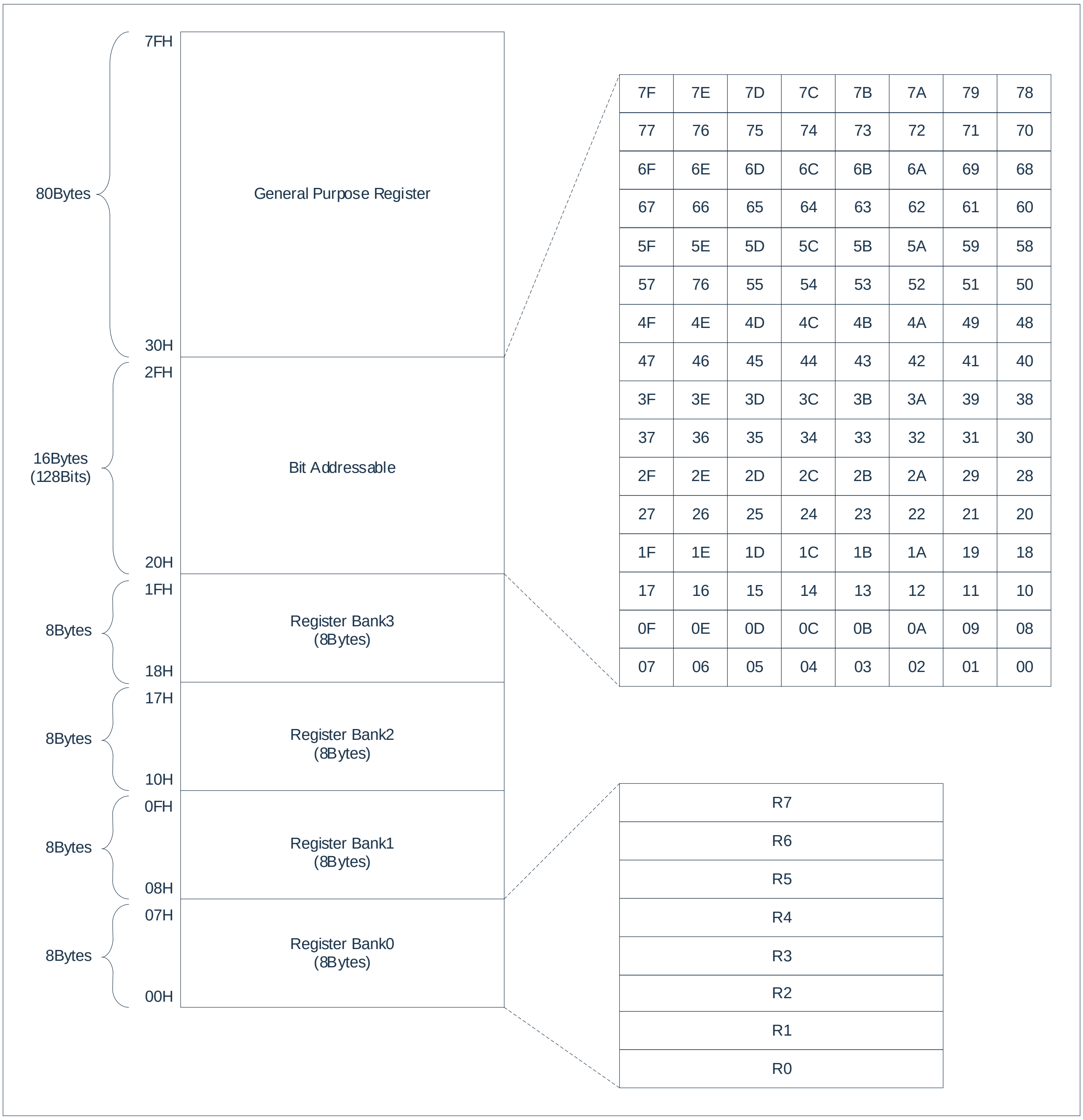
2.3 General-purpose data memory (RAM)

The internal data memory is divided into three parts: Low 128 Bytes, High 128 Bytes, and Special Function Register (SFR). The RAM space allocation structure is illustrated in the diagram below:



In the diagram, the High 128 Bytes and SFR occupy the same address range (80H~FFH), but they are independent entities. Direct addressing of storage space above 7FH (SFR) and indirect addressing above 7FH (High 128 Bytes) access different memory spaces. The SFR is divided into two pages: BANK0 and BANK1, each with 128 Bytes, occupying the same address region, and accessed through a page register to select the appropriate memory space.

The allocation of the Low 128 Bytes space registers is shown in the diagram below. The lowest 32 bytes (00H~1FH) consist of four register groups, each with 8 storage units labeled R0~R7, used for storing operands and intermediate results. After a reset, group 0 is selected by default; to choose other register groups, the program state must be changed. The 16 Bytes following the register groups (20H~2FH) form the bit-addressable memory space, where the RAM units can be operated on a byte basis as well as directly manipulated on a bit basis. The remaining 80 storage units (30H~7FH) can be used by the user to set up the stack area and store intermediate data.

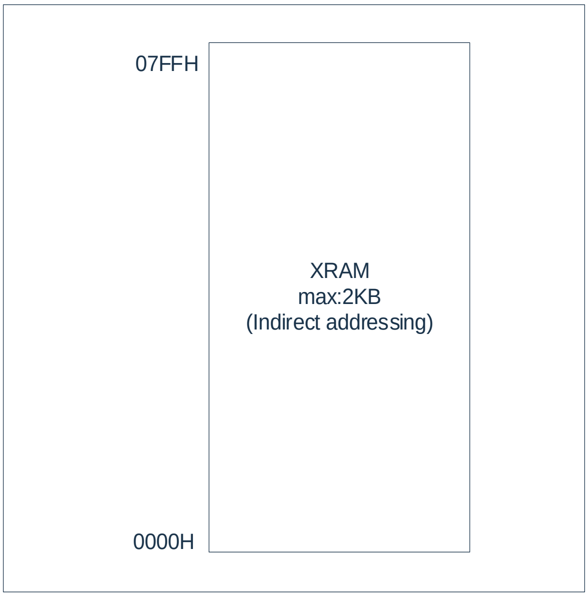


7F	7E	7D	7C	7B	7A	79	78
77	76	75	74	73	72	71	70
6F	6E	6D	6C	6B	6A	69	68
67	66	65	64	63	62	61	60
5F	5E	5D	5C	5B	5A	59	58
57	76	55	54	53	52	51	50
4F	4E	4D	4C	4B	4A	49	48
47	46	45	44	43	42	41	40
3F	3E	3D	3C	3B	3A	39	38
37	36	35	34	33	32	31	30
2F	2E	2D	2C	2B	2A	29	28
27	26	25	24	23	22	21	20
1F	1E	1D	1C	1B	1A	19	18
17	16	15	14	13	12	11	10
0F	0E	0D	0C	0B	0A	09	08
07	06	05	04	03	02	01	00

R7
R6
R5
R4
R3
R2
R1
R0

2.4 General-purpose external data register (XRAM)

The chip has a maximum of 2KB XRAM area, which is independent of FLASH/RAM. The space allocation structure of XRAM is illustrated in the diagram below:



Access to the XRAM/XSFR space is performed using the DPTR (Data Pointer). DPTR includes two sets of pointers: DPTR0 and DPTR1, which are selected by the DPS (Data Pointer Select) register. For example, using indirect addressing with the MOVX instruction, the assembly code is as follows:

```
MOV        R0,#01H
MOV        A,#5AH
MOVX       @R0,A          ;Write the data in A to the XRAM address 01H, with the high 8-bit address
                          determined by DPH0/1
```

In Keil51, when the Target-->Memory Model is set to Large, the C compiler will use XRAM for variable addresses. Typically, DPTR is used for XRAM/XSFR operations.

2.5 Special function register (SFR)

Special Function Registers (SFR) are registers with specific purposes, essentially acting as specific RAM cells within the microcontroller. They are distributed across the address range from 80H to FFH. Users can access them using direct addressing instructions for byte storage, where the lower four bits of the address are either 0000 or 1000, allowing bit-addressable access, such as for registers like P0, TCON, and P1.

SFRs are divided into two pages: BANK0 and BANK1. The paging functionality is controlled by the SFRS register. The address ranges 0xD1-0xD7, 0xD9-0xDF, 0xE1-0xE7, 0xE9-0xEF, and 0xF1-0xF7 contain a total of 35 addresses that represent different registers in BANK0 and BANK1, allowing for separate access through paging. Other addresses correspond to the same registers in both BANK0 and BANK1, meaning they can be accessed from either bank.

It is important to note that the system does not automatically switch between BANK0 and BANK1; this operation must be performed through software by writing to the paging register (SFRS). Particularly in interrupt service routines, users must manually save and restore the state of the paging register (SFRS).

SFR paging control register (SFRS)

0x92	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SFRS	SFRS7	SFRS6	SFRS5	SFRS4	SFRS3	SFRS2	SFRS1	SFRS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 SFRS<7:0>: Paging selection control
0x00= BANK0
0x01= BANK1
Others= Disable access

BANK0 register table:

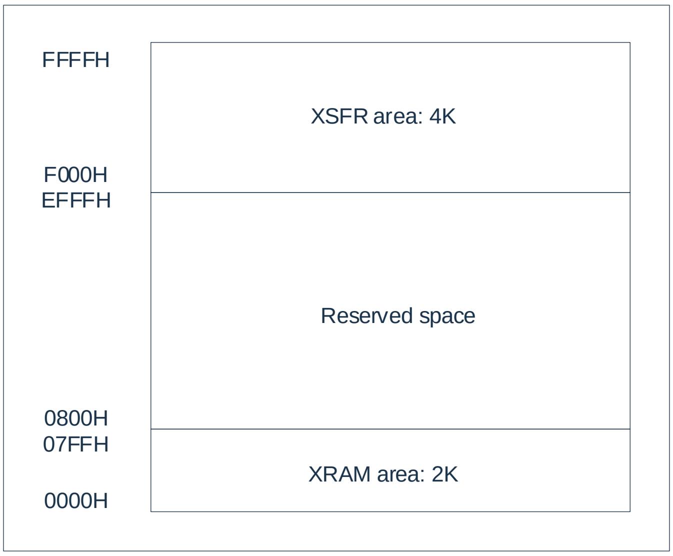
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0xF8	--	MREGION	MMODE	MDATA	MADRL	MADRH	MSTATUS	MLOCK
0xF0	B	I2CSADR	I2CSCR	I2CSBUF	I2CMSA	I2CMCR	I2CMBUF	I2CMTP
0xE8	--	ADCON2	SCON1	SBUF1	SPCR	SPSR	SPDR	SSCR
0xE0	ACC	--	TL4	TH4	--	WWCON0	WWCMPD	WWCON1
0xD8	--	ADCON3	TL3	TH3	ADRESL	ADRESH	ADCON1	ADCON0
0xD0	PSW	ADCMPC	T34MOD	ADDLYL	ADCMPL	ADCMPH	SCKSEL	SCKSTAU
0xC8	T2CON	T2IF	RLDL	RLDH	TL2	TH2	CCEN	T2IE
0xC0	--	CAP2CON	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3
0xB8	IP	EIP1	EIP2	EIP3	WUTCRL	WUTCRH	BUZDIV	BUZCON
0xB0	P3	--	EIF2	--	P0EXTIF	P1EXTIF	P2EXTIF	P3EXTIF
0xA8	IE	--	EIE2	--	P0EXTIE	P1EXTIE	P2EXTIE	P3EXTIE
0xA0	P2	P1TRIS	P2TRIS	P3TRIS	--	--	--	--
0x98	SCON0	SBUF	P0TRIS	--	--	--	--	--
0x90	P1	FUNCCR	SFRS	DPX0	--	DPX1	TA	WDCON
0x88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	CLKDIV
0x80	P0	SP	DPL0	DPH0	DPL1	DPH1	DPS	PCON

BANK1 register table:

	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F
0xF8	--	MREGION	MMODE	MDATA	MADRL	MADRH	MSTATUS	MLOCK
0xF0	B	--	--	--	--	--	--	--
0xE8	--	MD0	MD1	MD2	MD3	MD4	MD5	ARCON
0xE0	ACC	--	--	--	--	--	--	--
0xD8	--	--	--	--	--	--	--	--
0xD0	PSW	--	--	--	--	--	--	--
0xC8	T2CON	T2IF	RLDL	RLDH	TL2	TH2	CCEN	T2IE
0xC0	--	CAP2CON	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3
0xB8	IP	EIP1	EIP2	EIP3	WUTCRL	WUTCRH	BUZDIV	BUZCON
0xB0	P3	--	EIF2	--	P0EXTIF	P1EXTIF	P2EXTIF	P3EXTIF
0xA8	IE	--	EIE2	--	P0EXTIE	P1EXTIE	P2EXTIE	P3EXTIE
0xA0	P2	P1TRIS	P2TRIS	P3TRIS	--	--	--	--
0x98	SCON0	SBUF	P0TRIS	--	--	--	--	--
0x90	P1	FUNCCR	SFRS	DPX0	--	DPX1	TA	WDCON
0x88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	CLKDIV
0x80	P0	SP	DPL0	DPH0	DPL1	DPH1	DPS	PCON

2.6 External special function registers (XSFR)

The External Special Function Registers (XSFR) share the addressing space with the external RAM (XRAM). They primarily include port control registers and other functional control registers. The addressing range is shown below.



The external special function registers are listed as follows.

Address	Symbol	Description
F000H	P00CFG	P00 port configuration register
F001H	P01CFG	P01 port configuration register
F002H	P02CFG	P02 port configuration register
F003H	P03CFG	P03 port configuration register
F004H	P04CFG	P04 port configuration register
F005H	P05CFG	P05 port configuration register
F006H	--	--
F007H	--	--
F009H	P0OD	P0 port open drain control register
F00AH	P0UP	P0 port pull-up resistor control register
F00BH	P0RD	P0 port pull-down resistor control register
F00CH	P0DR	P0 port drive current control register
F00DH	P0SR	P0 port slope control register
F00EH	P0DS	P0 port data input selection register
F00FH	P0MODE	P0 port data input level selection register
F010H	P10CFG	P10 port configuration register
F011H	P11CFG	P11 port configuration register
F012H	P12CFG	P12 port configuration register
F013H	P13CFG	P13 port configuration register
F014H	P14CFG	P14 port configuration register
F015H	P15CFG	P15 port configuration register
F016H	P16CFG	P16 port configuration register
F017H	P17CFG	P17 port configuration register
F019H	P1OD	P1 port open drain control register
F01AH	P1UP	P1 port pull-up resistor control register
F01BH	P1RD	P1 port pull-down resistor control register
F01CH	P1DR	P1 port drive current control register
F01DH	P1SR	P1 port slope control register

Address	Symbol	Description
F01EH	P1DS	P1 port data input selection register
F01FH	P1MODE	P1 port data input level selection register
F020H	P20CFG	P20 port configuration register
F021H	P21CFG	P21 port configuration register
F022H	P22CFG	P22 port configuration register
F023H	P23CFG	P23 port configuration register
F024H	P24CFG	P24 port configuration register
F025H	P25CFG	P25 port configuration register
F026H	P26CFG	P26 port configuration register
F027H	P27CFG	P27 port configuration register
F029H	P2OD	P2 port open drain control register
F02AH	P2UP	P2 port pull-up resistor control register
F02BH	P2RD	P2 port pull-down resistor control register
F02CH	P2DR	P2 port drive current control register
F02DH	P2SR	P2 port slope control register
F02EH	P2DS	P2 port data input selection register
F02FH	P2MODE	P2 port data input level selection register
F030H	P30CFG	P30 port configuration register
F031H	P31CFG	P31 port configuration register
F032H	P32CFG	P32 port configuration register
F033H	P33CFG	P33 port configuration register
F034H	P34CFG	P34 port configuration register
F035H	P35CFG	P35 port configuration register
F036H	P36CFG	P36 port configuration register
F037H	P37CFG	P37 port configuration register
F039H	P3OD	P3 port open drain control register
F03AH	P3UP	P3 port pull-up resistor control register
F03BH	P3RD	P3 port pull-down resistor control register
F03CH	P3DR	P3 port drive current control register
F03DH	P3SR	P3 port slope control register
F03EH	P3DS	P3 port data input selection register
F03FH	P3MODE	P3 port data input level selection register
F080H	P00EICFG	P00 port interrupt control register
F081H	P01EICFG	P01 port interrupt control register
F082H	P02EICFG	P02 port interrupt control register
F083H	P03EICFG	P03 port interrupt control register
F084H	P04EICFG	P04 port interrupt control register
F085H	P05EICFG	P05 port interrupt control register
--	--	--
F088H	P10EICFG	P10 port interrupt control register
F089H	P11EICFG	P11 port interrupt control register
F08AH	P12EICFG	P12 port interrupt control register
F08BH	P13EICFG	P13 port interrupt control register
F08CH	P14EICFG	P14 port interrupt control register
F08DH	P15EICFG	P15 port interrupt control register
F08EH	P16EICFG	P16 port interrupt control register
F08FH	P17EICFG	P17 port interrupt control register
F090H	P20EICFG	P20 port interrupt control register

Address	Symbol	Description
F091H	P21EICFG	P21 port interrupt control register
F092H	P22EICFG	P22 port interrupt control register
F093H	P23EICFG	P23 port interrupt control register
F094H	P24EICFG	P24 port interrupt control register
F095H	P25EICFG	P25 port interrupt control register
F096H	P26EICFG	P26 port interrupt control register
F097H	P27EICFG	P27 port interrupt control register
F098H	P30EICFG	P30 port interrupt control register
F099H	P31EICFG	P31 port interrupt control register
F09AH	P32EICFG	P32 port interrupt control register
F09BH	P33EICFG	P33 port interrupt control register
F09CH	P34EICFG	P34 port interrupt control register
F09DH	P35EICFG	P35 port interrupt control register
F09EH	P36EICFG	P36 port interrupt control register
F09FH	P37EICFG	P37 port interrupt control register
--	--	--
F0C0H	PS_INT0	External interrupt 0 input port allocation register
F0C1H	PS_INT1	External interrupt 1 input port allocation register
F0C2H	PS_T0	Timer0 external clock input port allocation register
F0C3H	PS_T0G	Timer0 gate input port allocation register
F0C4H	PS_T1	Timer1 external clock input port allocation register
F0C5H	PS_T1G	Timer1 gate input port allocation register
F0C6H	PS_T2	Timer2 external event or gate input port allocation register
F0C7H	PS_T2EX	Timer2 falling edge auto-reload input port allocation register
F0C8H	PS_CAP0	Timer2 input capture channel 0 port allocation register
F0C9H	PS_CAP1	Timer2 input capture channel 1 port allocation register
F0CAH	PS_CAP2	Timer2 input capture channel 2 port allocation register
F0CBH	PS_CAP3	Timer2 input capture channel 3 port allocation register
F0CCH	PS_ADET	ADC external trigger input port allocation register
F0CDH	PS_FB	PWM external brake signal port allocation register
--	--	--
F120H	PWMCON	PWM control register
F121H	PWMOE	PWM output enable register
F122H	PWMPINV	PWM output polarity selection register
F123H	PWM01PSC	PWM0/PWM1 prescaler control register
F124H	PWM23PSC	PWM2/PWM3 prescaler control register
F125H	PWM45PSC	PWM4/PWM5 prescaler control register
F126H	PWMCNTE	PWM counter start control register
F127H	PWMCNTM	PWM counter mode selection register
F128H	PWMCNTCLR	PWM counter clear control register
F129H	PWMLOADEN	PWM load enable control register
F12AH	PWM0DIV	PWM0 prescaler control register
F12BH	PWM1DIV	PWM1 prescaler control register
F12CH	PWM2DIV	PWM2 prescaler control register
F12DH	PWM3DIV	PWM3 prescaler control register
F12EH	PWM4DIV	PWM4 prescaler control register
F12FH	PWM5DIV	PWM5 prescaler control register

Address	Symbol	Description
F130H	PWMP0L	PWM0 period data register low 8 bits
F131H	PWMP0H	PWM0 period data register high 8 bits
F132H	PWMP1L	PWM1 period data register low 8 bits
F133H	PWMP1H	PWM1 period data register high 8 bits
F134H	PWMP2L	PWM2 period data register low 8 bits
F135H	PWMP2H	PWM2 period data register high 8 bits
F136H	PWMP3L	PWM3 period data register low 8 bits
F137H	PWMP3H	PWM3 period data register high 8 bits
F138H	PWMP4L	PWM4 period data register low 8 bits
F139H	PWMP4H	PWM4 period data register high 8 bits
F13AH	PWMP5L	PWM5 period data register low 8 bits
F13BH	PWMP5H	PWM5 period data register high 8 bits
--	--	--
F140H	PWMD0L	PWM0 compare data register low 8 bits
F141H	PWMD0H	PWM0 compare data register high 8 bits
F142H	PWMD1L	PWM1 compare data register low 8 bits
F143H	PWMD1H	PWM1 compare data register high 8 bits
F144H	PWMD2L	PWM2 compare data register low 8 bits
F145H	PWMD2H	PWM2 compare data register high 8 bits
F146H	PWMD3L	PWM3 compare data register low 8 bits
F147H	PWMD3H	PWM3 compare data register high 8 bits
F148H	PWMD4L	PWM4 compare data register low 8 bits
F149H	PWMD4H	PWM4 compare data register high 8 bits
F14AH	PWMD5L	PWM5 compare data register low 8 bits
F14BH	PWMD5H	PWM5 compare data register high 8 bits
--	--	--
F150H	PWMDD0L	PWM0 asymmetric down compare data register low 8 bits
F151H	PWMDD0H	PWM0 asymmetric down compare data register high 8 bits
F152H	PWMDD1L	PWM1 asymmetric down compare data register low 8 bits
F153H	PWMDD1H	PWM1 asymmetric down compare data register high 8 bits
F154H	PWMDD2L	PWM2 asymmetric down compare data register low 8 bits
F155H	PWMDD2H	PWM2 asymmetric down compare data register high 8 bits
F156H	PWMDD3L	PWM3 asymmetric down compare data register low 8 bits
F157H	PWMDD3H	PWM3 asymmetric down compare data register high 8 bits
F158H	PWMDD4L	PWM4 asymmetric down compare data register low 8 bits
F159H	PWMDD4H	PWM4 asymmetric down compare data register high 8 bits
F15AH	PWMDD5L	PWM5 asymmetric down compare data register low 8 bits
F15BH	PWMDD5H	PWM5 asymmetric down compare data register high 8 bits
F15CH	PWMBRKC	PWM brake recovery control register
F15DH	PWMBRKRDTL	PWM delay recovery data register low 8 bits
F15EH	PWMBRKRDTH	PWM delay recovery data register high 8 bits
--	--	--
F160H	PWMDTE	PWM programmable dead zone delay control register
F161H	PWM01DT	PWM0/PWM1 programmable dead zone delay time register
F162H	PWM23DT	PWM2/PWM3 programmable dead zone delay time register
F163H	PWM45DT	PWM4/PWM5 programmable dead zone delay time register

Address	Symbol	Description
F164H	PWMMASKE	PWM mask enable control register
F165H	PWMMASKD	PWM mask data register
F166H	PWMFBKC	PWM brake control register
F167H	PWMFBKD	PWM brake data register
F168H	PWMPIE	PWM period interrupt enable register
F169H	PWMZIE	PWM zero-crossing interrupt enable register
F16AH	PWMUIE	PWM up compare interrupt enable register
F16BH	PWMDIE	PWM down compare interrupt enable register
F16CH	PWMPIF	PWM period interrupt flag register
F16DH	PWMZIF	PWM zero-crossing interrupt flag register
F16EH	PWMUIF	PWM up compare interrupt flag register
F16FH	PWMDIF	PWM down compare interrupt flag register
--	--	--
F500H	C0CON0	Comparator 0 control register 0
F501H	C0CON1	Comparator 0 control register 1
F502H	C0CON2	Comparator 0 control register 2
F503H	C1CON0	Comparator 1 control register 0
F504H	C1CON1	Comparator 1 control register 1
F505H	C1CON2	Comparator 1 control register 2
F506H	CNVRCON	Comparator reference voltage control register
F507H	CNFBCON	Comparator brake control register
F508H	CNIE	Comparator interrupt enable register
F509H	CNIF	Comparator interrupt flag register
--	--	--
F50CH	C0HYS	Comparator 0 hysteresis control register
F50DH	C1HYS	Comparator 1 hysteresis control register
F50EH	C0CON3	Comparator 0 control register 3
F50FH	C1CON3	Comparator 1 control register 3
--	--	--
F520H	OP0CON0	Operational amplifier 0 control register 0
F521H	OP0CON1	Operational amplifier 0 control register 1
F522H	--	--
F523H	OP1CON0	Operational amplifier 1 control register 0
F524H	OP1CON1	Operational amplifier 1 control register 1
F525H	--	Unused
--	--	--
F529H	PGACON0	PGA control register 0
--	--	--
F52BH	PGACON2	PGA control register 2
--	--	--
F550H	ADCCNTL	ADC multiple conversion count low 8 bits
F551H	ADCCNTH	ADC multiple conversion count high 8 bits
F552H	ADCRES0	ADC multiple conversion result low 8 bits
F553H	ADCRES1	ADC multiple conversion result middle 8 bits
F554H	ADCRES2	ADC multiple conversion result high 8 bits
--	--	--
F5C0H	BRTCON	BRT module control register
F5C1H	BRTDL	BRT timer load value low 8 bits

Address	Symbol	Description
F5C2H	BRTDH	BRT timer load value high 8 bits
--	--	--
F5E0H	UID0	UID<7:0>
F5E1H	UID1	UID<15:8>
F5E2H	UID2	UID<23:16>
F5E3H	UID3	UID<31:24>
F5E4H	UID4	UID<39:32>
F5E5H	UID5	UID<47:40>
F5E6H	UID6	UID<55:48>
F5E7H	UID7	UID<63:56>
F5E8H	UID8	UID<71:64>
F5E9H	UID9	UID<79:72>
F5EAH	UID10	UID<87:80>
F5EBH	UID11	UID<95:88>
--	--	--
F690H	LVDCON	Power monitoring register
F691H	BOOTCON	BOOT control register
F692H	ADCLDO	ADC reference voltage control register
F693H	LVDEICFG	Power monitoring control register
F694H	LSECRL	LSE timer data register low 8 bits
F695H	LSECRH	LSE timer data register high 8 bits
F696H	LSECON	LSE timer control register
F697H	XTSCM	Oscillator stop detection control register
--	--	--
F704H	SMODECON0	Idle mode control register 0
F705H	SMODECON1	Idle mode control register 1
F706H	PCRCDL	Program CRC calculation result data register low 8 bits
F707H	PCRCDH	Program CRC calculation result data register high 8 bits
--	--	--

3. Reset

Reset Time refers to the time from when the chip is reset to when it begins executing instructions, with a default design value of approximately 18 ms. This time includes the oscillator startup time and configuration time. Whether the chip is reset due to power-on or other reasons, this reset time will always be present. Additionally, when the oscillator is selected as an external low-speed crystal oscillator (32.768 kHz), the reset time (including startup time) is approximately 1.5 seconds (with external capacitance of 10 pF to 22 pF).

The chip can be reset using the following methods:

- Power-on reset
- External reset
- Low voltage reset
- Watchdog overflow reset
- Windowed watchdog reset
- Software reset
- CONFIG state protection reset
- Power-on configuration monitoring reset

Whenever any of these reset methods occur, all system registers will restore to their default states, the program will stop running, and the program counter (PC) will be cleared. After the reset, the program will start executing from the reset vector at 0000H.

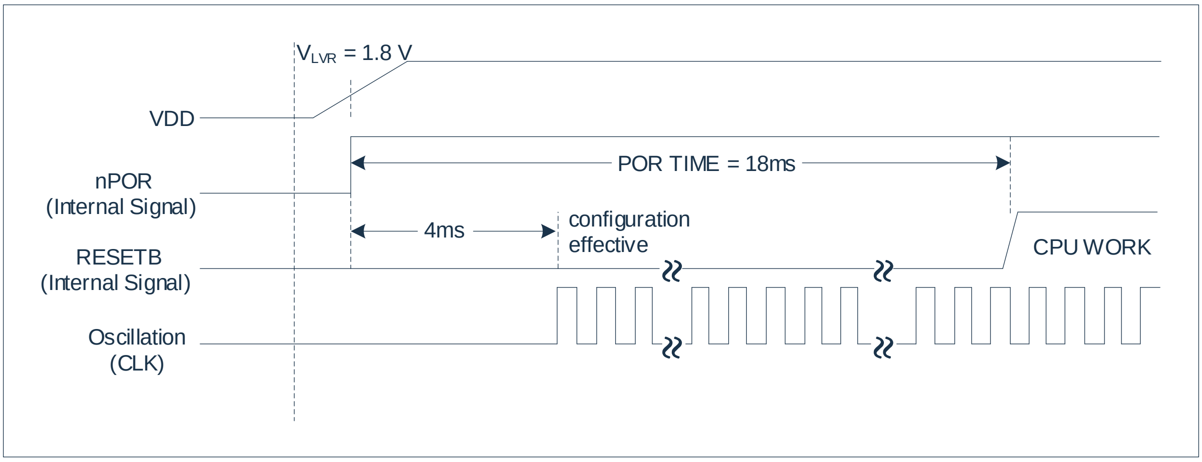
Any reset condition requires a certain response time, and the system provides a comprehensive reset process to ensure the smooth execution of the reset action.

3.1 Power-on reset

Power-on reset is closely related to Low Voltage Reset (LVR) operations. The power-on process of the system follows a gradually rising curve and requires some time to reach the normal voltage level. Below is the normal timing sequence for power-on reset:

- Power-On: The system detects the rising power supply voltage and waits for it to stabilize.
- System Initialization: All system registers are set to their initial values.
- Oscillator Operation: The oscillator begins to provide the system clock.
- Program Execution: Power-on concludes, and the program begins to run.

The stabilization time is typically 18 ms; if a 32.768 kHz crystal oscillator is selected, the stabilization time is approximately 1.5 seconds. The timing diagram for the power-on reset is shown below:



The system can determine if a power-on reset has occurred by checking the PORF (WDCON.6) flag. The types of resets that can set the PORF flag to 1 include: power-on reset, LVR reset, power monitoring reset, CONFIG protection reset, external reset, and windowed watchdog reset.

The relationship between reset flags and reset signals is shown in the table below:

Reset source / Flag bit	Power-on reset	LVR low voltage reset	Power on monitoring reset	CONFIG protection reset	Software reset	External reset	Watchdog reset	Windowed watchdog reset
SWRST	0	0	0	0	1	0	Unaffected	0
PORF	1	1	1	1	Unaffected	1	Unaffected	1
EXTIF	0	0	Unaffected	Unaffected	Unaffected	1	Unaffected	Unaffected
FXTIF	0	0	Unaffected	1	Unaffected	Unaffected	Unaffected	Unaffected
WDTRF	0	0	0	0	Unaffected	0	1	0
WWDTRF	0	0	Unaffected	Unaffected	Unaffected	Unaffected	Unaffected	1

0x97	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDCON	SWRST	PORF	EXTIF	FIXIF	WDTIF	WDTRF	WDTRE	WDTCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	1	0	0	0	0	0	0

Bit7	SWRST:	Software reset control bit
	1:	Execute a system software reset (write 0 to clear after reset)
	0:	--
Bit6	PORF:	Power-on reset flag bit
	1:	Power-on reset (write 0 to clear, no TA write timing)
	0:	--
Bit5	EXTIF:	External reset flag bit
	1=	External reset (write 0 to clear, no TA write timing)
	0=	--
Bit4	FIXIF:	CONFIG status protection reset flag bit
	1=	CONFIG status protection reset (write 0 to clear, no TA write timing)
	0=	--
Bit3	WDTIF:	WDT overflow interrupt flag bit
	1=	WDT overflow (write 0 to clear)
	0=	No WDT overflow
Bit2	WDTRF:	WDT reset flag bit
	1=	Reset by WDT (write 0 to clear)
	0=	Not reset by WDT
Bit1	WDTRE:	WDT reset enable bit
	1=	Enable WDT to reset CPU
	0=	Disable WDT to reset CPU
Bit0	WDTCLR:	WDT counter clear bit
	1=	Enable WDT counter (automatically cleared by hardware)
	0=	Disable WDT counter (writing 0 has no effect)

3.2 External reset

An external reset refers to a reset signal from the external port (NRST) that resets the chip after being input through a Schmitt trigger. A reset request will be triggered if the NRST pin is held low for more than 4 LSI clock cycles, provided that the operating voltage range and stable oscillation conditions are met. Once the internal state is initialized and the reset status changes to 1, an stabilization time of 16 ms is required before the internal RESETB signal changes to 1, allowing the program to start executing from the vector address 0000H.

During the stabilization time, the chip undergoes a reconfiguration process, which is the same as the power-on reset configuration process. The external reset pin NRST and its pull-up resistor are enabled through CONFIG configuration.

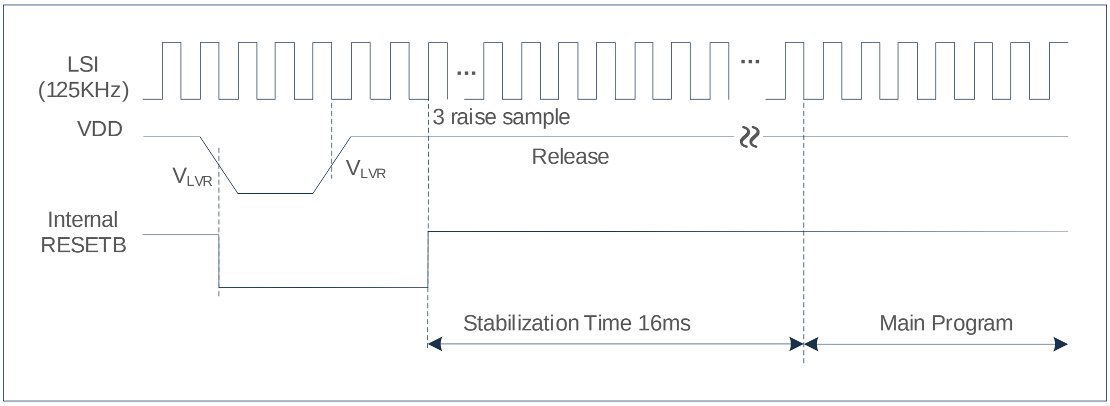
The system can determine whether it is in an external reset state by checking the EXTIF (WDCON.5) flag.

3.3 Low voltage reset (LVR)

The chip integrates a Low Voltage Reset (LVR) function. When the system voltage VDD drops below the LVR threshold voltage, the LVR is triggered, causing a system reset. The voltage point that triggers the reset can be set in the CONFIG.

When the LVR module detects that VDD<V_{LVR}, it will request a reset. In sleep mode (STOP), the LVR low voltage reset function is disabled.

The timing diagram for the LVR low voltage reset is shown below:



During the stabilization time, the chip undergoes a reconfiguration process that is the same as the power-on reset configuration process.

3.4 Watchdog reset

The watchdog reset is a protective feature of the system. Under normal conditions, the program resets the watchdog timer. If an error occurs and the system enters an unknown state, the watchdog timer will overflow, triggering a system reset. After the watchdog reset, the system restarts and returns to normal operation.

The WDT (Watchdog Timer) counter cannot be addressed directly. It starts counting after the power-on reset process is completed and the program begins execution. When setting the WDT register, it is recommended to first clear the WDT counter to accurately control the overflow timing.

The timing sequence for the watchdog reset is as follows:

1) Watchdog Timer Status: The system checks if the watchdog timer has overflowed; if it has, the system resets.
2) Initialization: All system registers are set to their default states.
3) Program: After the reset is completed, the program starts executing from address 0000H.

The clock source for the WDT is provided by the system clock, and the basic timing period for the WDT counter is Tsys. After a WDT overflow, the CPU and all registers are reset, and the program immediately begins execution from address 0000H one Tsys later. The watchdog reset does not reconfigure the power-on reset settings. The overflow time for the watchdog can be set by the program, with the overflow time selectable via the WTS2-WTS0 bits in the CKCON register. The watchdog overflow times are shown in the table below:

WTS[2:0]	Watchdog interval	Number of clocks	OVT@Fsys=16MHz	OVT@Fsys=48MHz
000	2^{17}	131072	8.192ms	2.731ms
001	2^{18}	262144	16.384ms	5.461ms
010	2^{19}	524288	32.768ms	10.923ms
011	2^{20}	1048576	65.536ms	21.845ms
100	2^{21}	2097152	131.072ms	43.691ms
101	2^{22}	4194304	262.144ms	87.381ms
110	2^{24}	16777216	1.048s	349.525ms
111	2^{26}	67108864	4.194s	1.398s

The WDT can also be configured to not reset the system, allowing it to generate an interrupt instead.

3.5 Windowed watchdog reset

The windowed watchdog reset is another protective feature of the system. In normal operation, the program clears the window watchdog timer within a specified window period. If an error occurs and the system enters an unknown state, clearing the windowed watchdog outside of the window period or allowing the windowed watchdog timer to overflow will trigger a system reset. After the window watchdog reset, the system restarts and returns to normal operation. The window watchdog can also be configured to not reset the system, allowing it to generate an interrupt instead. For more details, see the description below.

3.6 Software reset

The chip can implement a software reset through the program, which repositions the program flow to the reset address 0000H and then restarts the program. Users can achieve a custom software reset by writing to the software reset control bit WDCON[7] (SWRST=1). The software reset does not reconfigure the power-on reset settings.

3.7 CONFIG state protection reset

The CONFIG state protection reset is an enhanced protective mechanism for the system. During power-on reset, a set of 16-bit CONFIG registers is loaded with a fixed code (A569H) from FLASH. Under normal operation, this register is not modified. However, if the value of the register changes under specific non-program operation conditions and does not equal the original fixed code, the system will generate a reset if, after several clock samples, the register continues to hold a value different from the fixed code.

This reset mechanism prevents changes to configuration bits under certain conditions, ensuring that the system does not enter an unpredictable state.

During normal operation, the clock used to sample the register value is based on the internal fixed clock (Fixed_Clock at 8MHz, sourced from HSI) and the low-power clock (LSI at 125KHz). Once the value of the register is not equal to the fixed code, both the LSI and HSI oscillators are forcibly enabled, and the system clock switches to the LSI clock. If, after 12 Fixed_Clock samples or 3 LSI clock samples, the register still holds a value different from the fixed code, the system will generate a reset.

Using two types of clocks for sampling helps prevent issues related to oscillator stoppage under specific conditions.

The system can determine if a CONFIG state reset has occurred by checking the FIXIF flag (WDCON.4).

3.8 Power-on configuration monitoring reset

During the power-on configuration process, the chip contains a configuration monitoring circuit. If the configuration time exceeds a predefined duration or enters a state that cannot be reconfigured, the monitoring circuit begins timing from the start of the configuration. If the set time limit is exceeded, the monitoring circuit resets the configuration module, allowing it to restart the configuration process and preventing the system from entering an unpredictable state during power-on.

The monitoring circuit operates with the LSI clock (125KHz), and the default monitoring time is set to 65ms. If a 32.768KHz crystal oscillator is selected, the monitoring time extends to 2.1 seconds.

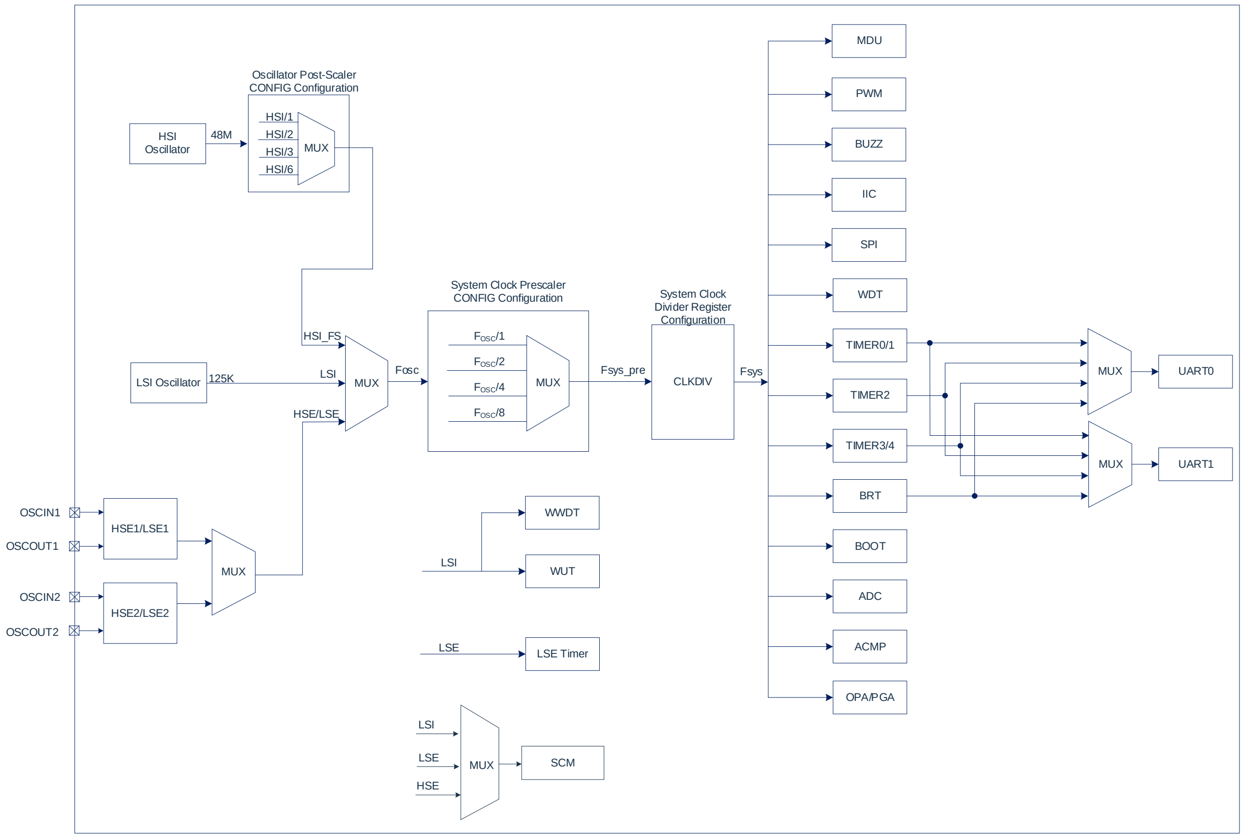
4. Clock Structure

The system clock has four types of clock sources, which can be selected through the configuration registers or user registers for clock source and clock division. The system clock sources are as follows:

- Internal high-speed oscillator (HSI) - 48MHz
- External high-speed oscillator (HSE) - 8MHz/16MHz
- External low-speed oscillator (LSE) - 32.768KHz
- Internal low-speed oscillator (LSI) – 125KHz

4.1 System clock structure

The block diagram of the system clock structure for various peripheral modules is shown below:



4.2 Relevant registers

4.2.1 Oscillator control register (CLKDIV)

0x8F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKDIV	CLKDIV7	CLKDIV6	CLKDIV5	CLKDIV4	CLKDIV3	CLKDIV2	CLKDIV1	CLKDIV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	CLKDIV<7:0>:	System clock Fsys prescaler bit
	00H=	Fsys=Fsys_pre
	Others=	Fsys=Fsys_pre/ (2*CLKDIV) (divided by 2, 4…510)

Sequence of instructions needed to modify CLKDIV (no other instructions can be inserted in between):

```
MOV                             TA,#0AAH
MOV                             TA,#055H
MOV                             CLKDIV,#02H
```

4.2.2 System clock switch register (SCKSEL)

0xD6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCKSEL	--	--	--	SEL	WRITE	CKSEL2	CKSEL1	CKSEL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit5	--	Reserved, set to 0.
Bit4	SEL:	System clock configuration selection
	1=	The clock source is determined by CKSEL<1:0>
	0=	The clock source is determined by CONFIG
Bit3	WRITE:	Write enable generates a pulse for clock switching
	1=	Switch clock
	0=	Do not switch clock
Bit2~Bit0	CKSEL<2:0>:	System clock source selection bit
	111=	LSI
	110=	LSE
	101=	HSE
	100=	HSI
	Other=	Invalid value, access disabled.

Sequence of instructions needed to modify SCKSEL (no other instructions can be inserted in between):

```
MOV                             TA,#0AAH
MOV                             TA,#055H
MOV                             SCKSEL, #05H
```

4.2.3 System clock status register (SCKSTAU)

0xD7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCKSTAU	LSI_F	LSE_F	HSE_F	HSI_F	--	--	--	--
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	LSI_F:	Low-speed internal stability status bit
	1=	Stable
	0=	Unstable
Bit6	LSE_F:	Low-speed external crystal stability status bit
	1=	Stable
	0=	Unstable
Bit5	HSE_F:	High-speed external crystal stability status bit
	1=	Stable
	0=	Unstable
Bit4	HSI_F:	High-speed internal clock stability status bit
	1=	Stable
	0=	Unstable
Bit3~Bit0	--	Reserved, set to 0.

4.2.4 System clock monitor register (SCM)

F697H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XTSCM	SCMEN	SCMIE	--	--	--	--	SCMIF	SCMSTA
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Reset value	0	0	0	0	0	0	0	0

Bit7	SCMEN:	Stop oscillation detection module enable
	1=	Enable
	0=	Disable
Bit6	SCMIE:	Stop oscillation detection interrupt enable bit (this interrupt shares an entry with the LSE timer interrupt)
	1=	Enable
	0=	Disable
Bit5~Bit2	--	Reserved, set to 0.
Bit1	SCMIF:	Stop oscillation interrupt flag bit
	1=	Stop oscillation
	0=	Cleared by software; after clearing, it will automatically switch to HSE/LSE main frequency (can only be cleared by software).
Bit0	SCMSTA:	Stop oscillation status bit, read-only:
	1=	Stop oscillation
	0=	Stop oscillation recovered

Note:

1) Both SCMIF and SCMSTA can reflect the status of HSE/LSE as the system clock. The main difference is that when HSE/LSE stops oscillating, SCMSTA will remain high until HSE/LSE recovers; SCMIF can also indicate the stop oscillation of HSE/LSE, but it can generate an interrupt (interrupt must be enabled). SCMIF can be cleared through the register, and after clearing, the main frequency will switch back to HSE/LSE. If it is still in a stop oscillation state, the interrupt will be triggered again.

2) After a stop oscillation, the main frequency will switch from HSE/LSE to HSI. If HSE/LSE recovers, SCMSTA will automatically be cleared, and the main frequency will automatically switch back from HSI to HSE/LSE.

4.2.5 Function clock control register

Watchdog overflow time/timer clock source selection register CKCON

0x8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	WTS2	WTS1	WTS0	T1M	T0M	--	--	T0CNTM
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	1	1	1

Bit7~Bit5	WTS<2:0>:	WDT overflow time select bit
	000=	2^{17}*Tsys
	001=	2^{18}*Tsys
	010=	2^{19}*Tsys
	011=	2^{20}*Tsys
	100=	2^{21}*Tsys
	101=	2^{22}*Tsys
	110=	2^{24}*Tsys
	111=	2^{26}*Tsys
Bit4	T1M:	Timer1 clock source selection bit
	0=	Fsys/12
	1=	Fsys/4
Bit3	T0M:	Timer0 clock source selection bit
	0=	Fsys/12
	1=	Fsys/4
Bit2~Bit1	--	Reserved, set to 1.
Bit0	T0CNTM:	Timer0 count source select bit
	0=	PWM0 output
	1=	T0 pin input

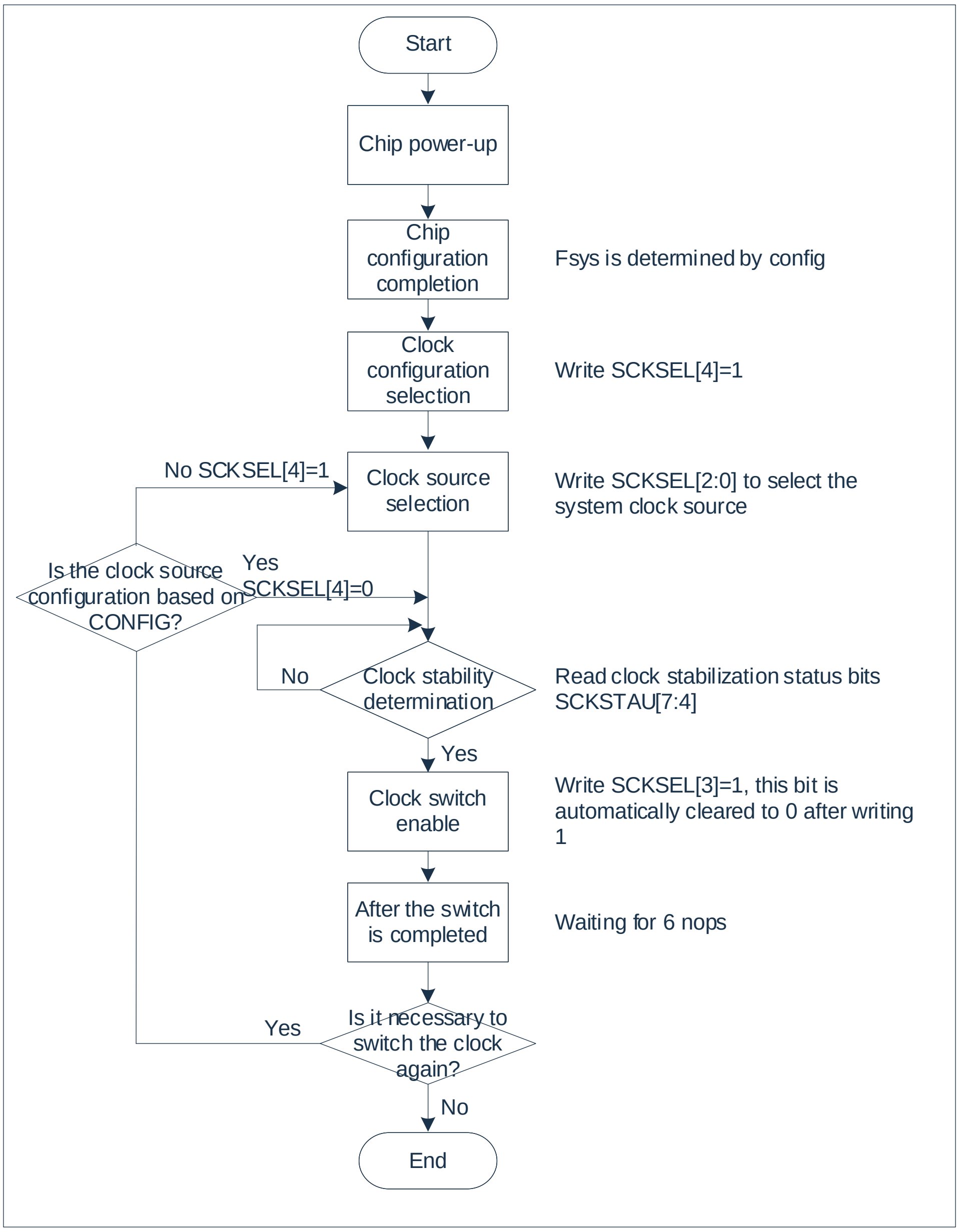
UART0/1 baud rate selection register FUNCCR

0x91	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FUNCCR	--	--	--	--	UART1_CKS1	UART0_CKS1	UART1_CKS0	UART0_CKS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4	--	Reserved, set to 0.
Bit3	UART1_CKS1:	UART1 timer clock source selection bits (high bits), {UART1_CKS1, UART1_CKS}:
	00=	Timer1 overflow clock
	01=	Timer4 overflow clock
	10=	Timer2 overflow clock
	11=	BRT overflow clock
Bit2	UART0_CKS1:	UART0 timer clock source selection bits (high bits), {UART0_CKS1, UART0_CKS}:
	00=	Timer1 overflow clock
	01=	Timer4 overflow clock
	10=	Timer2 overflow clock
	11=	BRT overflow clock
Bit1	UART1_CKS:	UART1 timer clock source selection bits (low bits), see UART1_CKS1 description
Bit0	UART0_CKS:	UART0 timer clock source selection bits (low bits), see UART1_CKS0 description

4.3 System clock switching

The procedure for switching the system clock is shown in the following figure.


Start
Chip power-up
Chip configuration completion
Fsys is determined by config
Clock configuration selection
Write SCKSEL[4]=1
No SCKSEL[4]=1
Clock source selection
Write SCKSEL[2:0] to select the system clock source
Is the clock source configuration based on CONFIG?
Yes
SCKSEL[4]=0
No
Clock stability determination
Read clock stabilization status bits SCKSTAU[7:4]
Yes
Clock switch enable
Write SCKSEL[3]=1, this bit is automatically cleared to 0 after writing 1
After the switch is completed
Waiting for 6 nops
Yes
Is it necessary to switch the clock again?
No
End

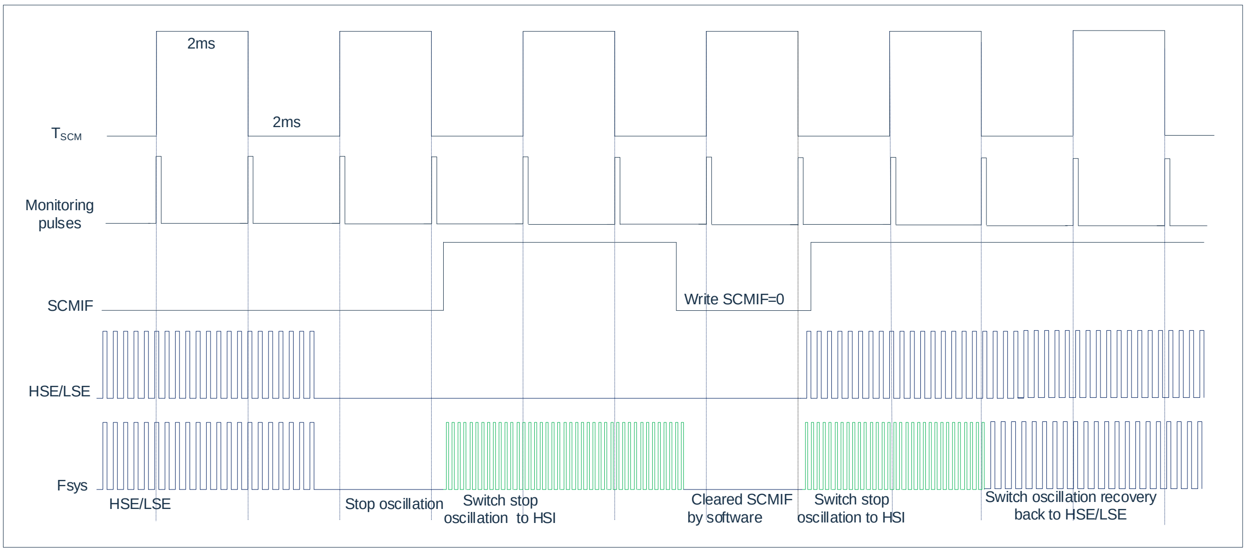
Note: When switching the system clock source, make sure that the corresponding clock source exists and has been configured correctly.

4.4 System clock monitoring

System Clock Monitoring (SCM) is designed to prevent system failures caused by the stopping of the crystal oscillator. When using High-Speed External (HSE) or Low-Speed External (LSE) clocks, if the SCM detects that the HSE/LSE clock has stopped, it forcibly switches the system to the High-Speed Internal (HSI) clock source. Once the HSI stabilizes, the system runs at an 8 MHz main frequency. If the HSE/LSE clock resumes and stabilizes, the system clock automatically switches back to the HSE/LSE.

The SCM module monitors the HSE/LSE system clock every 4ms, with a duty cycle of 1:1. During the high state of the monitoring cycle (T_{SCM}), SCM checks for the stop condition of the HSE/LSE. In the low state, it processes the monitoring results. If a stoppage is detected, the system clock switches to HSI, and the stop interrupt flag (SCMIF) is set to 1. Clearing the SCMIF means that even if HSE/LSE has stopped, the system clock will automatically switch back to HSE/LSE.

The system clock monitoring architecture is shown in the following diagram.


2ms
2ms
TSCM
Monitoring pulses
SCMIF
Write SCMIF=0
HSE/LSE
Fsys
HSE/LSE
Stop oscillation
Switch stop oscillation to HSI
Cleared SCMIF by software
Switch stop oscillation to HSI
Switch oscillation recovery back to HSE/LSE

5. Power Management

Low power modes are divided into two categories:

- IDLE: Idle mode

  This mode includes two sub-modes: Idle Mode 1 (IDLE1) and Idle Mode 2 (IDLE2). The selection of the mode is controlled by the registers SMODECON0 and SMODECON1.

- STOP: Sleep mode

When users are developing programs in C, it is strongly recommended to use the IDLE and STOP macros to control the system modes instead of directly setting the IDLE and STOP bits. The macros are as follows:

Enter idle mode: IDLE();

Enter sleep mode: STOP();

The available clocks and wake-up sources in different operating modes are as follows:

Comparison	Normal operation	Idle mode		Sleep mode
		Idle mode 1	Idle mode 2	

Definition	MCU operates normally;
	CP is active;
	Peripherals are operational;
	LDO is on;
	FLASH is on.
		MCU in idle mode 1;
		CPU stops working;
		Digital peripherals remain operational;
		Analog peripherals remain operational;
		LDO is on;
		FLASH is on.
			MCU in idle mode 2;
			CPU stops working;
			Digital peripherals remain operational;
			Analog peripheral bits are forcibly disabled;
			LDO is off;
			FLASH is off.
				MCU in sleep mode;
				CPU stops working;
				Digital peripherals stop working;
				Analog peripherals are disabled by software;
				LDO is off;
				FLASH is off.

Entry conditions	After system reset, the chip enters normal mode.	SMODECON0=0x00;
		SMODECON1=0x00;
		IDLE();
		Enter idle mode 1.
			When system clock is LSI or LSE,
			SMODECON0=0x55;
			SMODECON1=0xAA;
			IDLE();Enter idle mode 2.
				STOP();
				Enter sleep mode.

Wake-up sources	--	All interrupts	Other interrupts except ADC and ACMP	WUT timer wake-up;
				INT0/1 interrupt wake-up;
				GPIO interrupt wake-up;
				LSE timer wake-up;
				WWDT timer wake-up.

Available clocks	--	All clocks	System clock is LSI or LSE	No available clocks
Post-wake mode	--	MCU resumes normal mode, and program runs normally.	MCU resumes normal mode, and program runs normally.	MCU resumes normal mode, and program runs normally.
Wake wait time	--	Immediate execution	Wake wait time set by CONFIG	Wake wait time set by CONFIG
LVR	Supported	Supported	Supported	Supported
External reset	Supported	Supported	Supported	Supported
Windowed watchdog reset	Supported	Supported	Supported	Supported
Watchdog reset	Supported	Supported	Supported	Not supported

5.1 Power management registers

5.1.1 Power management register (PCON)

0x87	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD0	SMOD1	--	--	--	SWE	STOP	IDLE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SMOD0: UART0 baud rate doubling bit
0= UART0 normal baud rate
1= UART0 baud rate doubled

Bit6 SMOD1: UART1 baud rate doubling bit
0= UART1 normal baud rate
1= UART1 baud rate doubled

Bit5~Bit3 -- Reserved, set to 0.

Bit2 SWE: STOP state wake enable bit
(the system can be restarted by a power-on reset or an enabled external reset, regardless of the value of SWE.)
0= Disable function wakeup
1= Enable function wakeup (can be woken up by external interrupts and timer wakeup)

Bit1 STOP: Sleep state control bit
0= Not in sleep state
1= In sleep state (automatically cleared upon exiting stop mode).

Bit0 IDLE: Idle state control bit
0= Not in idle state
1= In idle state (automatically cleared upon exiting IDLE mode).

5.1.2 Idle mode control register (SMODECON0)

F704H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SMODECON0	SMODE07	SMODE06	SMODE05	SMODE04	SMODE03	SMODE02	SMODE0E1	SMODE00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 SMODE0<7:0>: Idle mode control register 0 (this register is hardware cleared to 0 upon waking from Idle mode)

5.1.3 Idle mode control register (SMODECON1)

F705H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SMODECON1	SMODE17	SMODE16	SMODE15	SMODE14	SMODE13	SMODE12	SMODE11	SMODE10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 SMODE1<7:0>: Idle mode configuration register 1 (This register is hardware cleared to 0 upon waking from Idle mode)
When {SMODE1, SMODE0}=0xaa55, Idle Mode 2 is enabled.
When {SMODE1, SMODE0}=other values, Idle Mode 1 is enabled.

5.2 Power monitoring registers

The MCU has built-in power detection functionality. If the LVD module is enabled (LVDEN=1) and the voltage monitoring point (LVDSEL) and voltage detection edge (LVDEICFG) are properly set, an LVDINTF trigger flag will be generated when the power supply voltage falls below or exceeds the set LVD threshold, alerting the user.

If the LVD module is enabled before entering sleep mode, the hardware will not turn off the module’s circuit during sleep; it must be disabled by software (LVDEN=0).

5.2.1 Power monitoring register (LVDCON)

0xF690	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LVDCON	--	LVDSEL2	LVDSEL1	LVDSEL0	LVDEN	LVDOUT	--	LVDINTF
R/W	R/W	R/W	R/W	R/W	R/W	R	R	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	--	Reserved, set to 0.
Bit6~Bit4	LVDSEL<2:0>:	LVD voltage monitoring point
	000=	2.0V
	001=	2.2V
	010=	2.4V
	011=	2.7V
	100=	3.0V
	101=	3.7V
	110=	4.0V
	111=	4.3V
Bit3	LVDEN:	LVD module enable
	0=	Disable
	1=	Enable
Bit2	LVDOUT	Power voltage monitoring bit
	0=	Power supply voltage is above the monitoring voltage
	1=	Power supply voltage is below the monitoring voltage
Bit1	--	Reserved, set to 0.
Bit0	LVDINTF:	LVD trigger flag bit
	0=	No trigger generated
	1=	Trigger generated (cleared by software)

5.2.2 Power monitoring control register (LVDEICFG)

F693H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LVDEICFG	--	--	--	--	--	--	LVDEICFG1	LVDEICFG0
R/W	R	R	R	R	R	R	R/W	R/W
Reset value	0	0	0	0	0	0	0	1

Bit7~Bit2	--	Reserved, set to 0.
Bit1~Bit0	LVDEICFG<1:0>:	Power monitoring trigger edge control bit
	00=	Trigger disabled
	01=	Power supply voltage is below the monitoring voltage
	10=	Power supply voltage is above the monitoring voltage
	11=	Power supply voltage is either below or above the monitoring voltage

5.3 Idle mode

The idle mode is divided into two types: Idle Mode 1 and Idle Mode 2. The selection of the mode is controlled by registers SMODECON0/SMODECON1. The configurations for the two modes are as follows:

Idle mode 1:

SMODECON0=0X00;

SMODECON1=0X00;

IDLE();

Idle mode 2:

SMODECON0=0X55;

SMODECON1=0XAA;

IDLE();

5.3.1 Idle mode 1

In this mode, only the CPU clock source is turned off. Therefore, peripheral functionalities (such as timers, PWM, and I^2C) and the clock generator continue to operate normally.

After entering Idle Mode 1, the system can be awakened by any interrupt. Upon waking up, it enters the interrupt service routine, and after returning from the interrupt, it continues executing the instructions following the sleep operation.

If Idle Mode 1 is entered in interrupt service routine, the system can only be awakened by interrupts of higher priority.

5.3.2 Idle mode 2

Idle Mode 2 can be configured when the chip uses either LSI or LSE as the main clock. In this mode, the enable bits for analog peripherals are forced to turn off; the CPU instruction fetching stops, and the LDO enters low-power mode while other digital peripherals continue to work normally. In this mode, the system can be awakened by any interrupt except for ADC and ACMP. After waking up from Idle Mode 2, it enters the interrupt service routine, and after returning from the interrupt, it continues executing the instructions following the sleep operation.

If Idle Mode 2 is entered in interrupt service routine, the system can only be awakened by interrupts of higher priority.

5.3.2.1 Wake-up wait state

After an interrupt occurs in Idle Mode 2, there is a waiting period before the system can wake up and execute the next instruction. The wait time for the CPU to wake up is set in the CONFIG during programming, and it can be configured to be between 50µs and 1s.

If the watchdog reset is enabled before entering Idle Mode 2, when the watchdog overflow interrupt occurs, it will wake up Idle Mode 2 regardless of whether the global interrupt enable bit (EA) is set to 1, and there will be no wake-up wait time.

5.3.2.2 Wake-up time

The total wake-up time when using external interrupts to wake up the system is:

Power manager stabilization time (200 µs) + wake-up wait time

The total wake-up time when using timer-based wake-up is:

Power manager stabilization time (200 µs) + wake-up timer counting time + wake-up wait time.

(Note: The above times apply when Fsys > 1MHz)

5.3.2.3 Reset operations

In Idle Mode 2, the system can also be restarted via power-off reset, external reset, WDT reset, or WWDT reset:

Power-Off Reset: No additional conditions are needed; after VDD drops to 0 V, powering back up to the operational voltage will enter the power-on reset state.

External reset: The external reset function needs to be enabled, and the related port must be configured as a dedicated reset pin. When the reset pin remains low for more than 4 LSI clock cycles, the system will generate a reset. Releasing the reset pin will cause the system to restart.

WDT Reset: When WDT reset is enabled, after the WDT overflow interrupt occurs, it will forcefully wake up Idle Mode 2, followed by a watchdog reset.

5.4 Sleep mode (STOP)

In this mode, all circuits are turned off except for the LVD and LSE modules (the LVD/LSE modules must be disabled via software). The system operates in a low-power state, and digital circuits do not function.

5.4.1 Sleep wake-up

After entering sleep mode, the sleep wake-up feature can be enabled (SWE must be set to 1) to wake the system from sleep. There are several ways to wake up from sleep mode:

1) INT0/1 interrupt

To use the INT0/1 interrupt to wake up from sleep mode, the global interrupt enable and INT0/1 interrupt enable must be activated before entering sleep. The related registers for INT0 and INT1 interrupts include IE, IP, TCON, and IO multiplexing mapping registers. INT0/1 interrupts can only wake the system with a falling edge.

2) External (GPIO) interrupt

For waking up using external GPIO interrupts, the global interrupt enable and port interrupt enable must be activated before entering sleep. External GPIO interrupts can be configured to wake the system on rising edge, falling edge, or both edges. The interrupt wake-up edge is controlled by the external interrupt configuration register PxnEICFG.

3) WUT wake-up

To wake up using the WUT timer, the timer wake-up function must be enabled before entering sleep, and the time duration from sleep to wake-up must be set. The clock source for the timer wake-up circuit is provided by the LSI (Low-Speed Internal Oscillator), which automatically turns on during sleep when the timer wake-up function is enabled.

4) LSE wake-up

To wake up using the LSE timer, the LSE module, counting enable, and timer wake-up functions must be enabled before entering sleep, along with setting the appropriate duration from sleep to wake-up.

5) WWDT wake-up

To wake up using the WWDT timer, the WWDT module must be enabled, and the WWDT sleep wake-up feature must be activated before entering sleep, along with setting the appropriate duration from sleep to wake-up.

5.4.2 Wake-up wait state

Regardless of whether it is an INT0/1 interrupt, external GPIO interrupt, WUT wake-up, LSE timer wake-up, or WWDT wake-up in sleep mode, after an interrupt occurs or the timer elapses, there is a wait time before the system can wake up and execute the next instruction. After an interrupt occurs or the timer reaches its set time, the system oscillator starts, but the oscillation frequency has not stabilized yet, and the CPU remains inactive, with the program counter (PC) still halted in sleep mode. The system needs to wait for a certain period before providing the clock to the CPU. The wake-up wait time is set in the configuration and can be adjusted from 50µs to 1s. After this wait time, the MCU determines that the system clock is stable and then provides the clock to the CPU to continue executing the program.

If both the internal wake-up timer and external interrupt wake-up functions are enabled, any of these wake-up methods can wake up the CPU after the system enters sleep mode. If the internal timer wakes up the oscillator first and then an external interrupt occurs, the program will execute the interrupt handler first after the wake-up wait time, followed by executing the instructions that came after the sleep operation.

5.4.3 Sleep wake-up time

The total wake-up time using external interrupts is:

Power Manager Stabilization Time (200 μs) + Wake-Up Wait Time

The total wake-up time using timer-based wake-up is:

Power Manager Stabilization Time (200 μs) + Wake-Up Timer Counting + Wake-Up Wait Time

(Note: The above times apply when Fsys>1MHz)

5.4.4 Reset operation in sleep mode

In sleep mode, the system can also be restarted through power-down reset, external reset, or WWDT reset. This reset method is independent of the SWE value; even if SWE = 0, the system can still be restarted through these reset operations.

Power-Down Reset: No additional conditions are needed; the system enters power-on reset state after VDD drops to 0V and then is powered back to the operating voltage.

External Reset: The external reset function needs to be enabled. The relevant port must be configured as a dedicated reset pin. During sleep, the reset pin should remain low for more than 1 μs to generate a reset, and releasing the reset pin will restart the system.

5.4.5 Sleep power consumption in debug mode

The sleep state in debug mode does not accurately reflect the actual chip’s sleep state.

In debug mode, after the system enters sleep state, the related power management circuits and oscillators are not turned off but remain active. Wake-up operations can also be performed in debug mode, and the wake-up methods are the same as in normal mode.

Therefore, the measured sleep current in this state does not represent true sleep power consumption. It is recommended to complete the development of wake-up functionality in debug mode and then turn off debug mode, followed by restarting the system. The measured current at this point will be the actual sleep power consumption.

5.4.6 Example applications of sleep mode

Before entering sleep mode, if users need to achieve a low sleep current, they should first check the status of all I/O ports. If there are floating I/O pins in the user's design, all floating pins should be set as output to ensure each input pin has a fixed state. This avoids undefined states on input pins which can increase sleep current. Additionally, disabling the ADC module, LSE module, LVD module, WWDT module, and other peripherals can further reduce sleep current.

Example: Using Timer Wake-Up, the assembly code for the sleep handling routine is as follows:

```
SLEEP_MODE:
            MOV                         WUTCRL,#31h
            MOV                         WUTCRH,#80h
            MOV                         P0TRIS,#0FFh
            MOV                         P0,#0FFh
            MOV                         P1TRIS,#0FFh
            MOV                         P1,#0FFh
            MOV                         P2TRIS,#0FFh
            MOV                         P2,#0FFh
            MOV                         P3TRIS,#0FFh
            MOV                         P3,#0FFh
            Instructions to disable other
            functions
            MOV                         PCON,#06H     ;Execute low-power sleep operation that allows
                                        wake-up
            NOP
            NOP
            NOP
            NOP
            NOP
            NOP                         ;Six NOP instructions must follow the sleep operation command
            Instructions after wake-up
```

6. Interrupt

6.1 Overview

The chip has 21 interrupt sources and interrupt vectors.

Interrupt source	Interrupt description	Interrupt vector	Same level priority sequence
INT0	External interrupt 0	0-0x0003	1
Timer0	Timer 0 interrupt	1-0x000B	2
INT1	External interrupt 1	2-0x0013	3
Timer1	Timer 1 interrupt	3-0x001B	4
UART0	TI0 or RI0	4-0x0023	5
Timer2	Timer 2 interrupt	5-0x002B	6
UART1	TI1 or RI1	6-0x0033	7
P0EXTIF<7:0>	P0 port external interrupt	7-0x003B	8
P1EXTIF<7:0>	P1 port external interrupt	8-0x0043	9
P2EXTIF<7:0>	P2 port external interrupt	9-0x004B	10
P3EXTIF<7:0>	P3 port external interrupt	10-0x0053	11
--	--	11-0x005B	12
--	--	12-0x0063	13
LSE_Timer/SCM	LSE timer/SCM interrupt	13-0x006B	14
ACMP	Comparator interrupt	14-0x0073	15
Timer3	Timer 3 interrupt	15-0x007B	16
Timer4	Timer 4 interrupt	16-0x0083	17
--	--	17-0x008B	18
PWM	PWM interrupt	18-0x0093	19
ADC	ADC interrupt	19-0x009B	20
WDT	WDT interrupt	20-0x00A3	21
I^{2}C	I^{2}C interrupt	21-0x00AB	22
SPI	SPI interrupt	22-0x00B3	23
--	--	--	--
WWDT	Windowed watchdog interrupt	28-0x00E3	29

The LSE Timer interrupt and the SCM (Stop Clock Monitor) interrupt share a common interrupt vector entry, but they have separate interrupt enable bits.

The chip defines two interrupt priority levels, allowing for two-level interrupt nesting. When one interrupt is already being serviced, if a higher-priority interrupt request comes in, it can preempt the current interrupt, enabling interrupt nesting.

6.2 External interrupts

6.2.1 INT0/INT1 interrupt

Each pin on the chip supports the native 8051 INT0 and INT1 external interrupts. INT0 and INT1 can be configured to trigger on falling edges or low levels, with the relevant control register being TCON. INT0 and INT1 occupy two separate interrupt vectors.

6.2.2 GPIO interrupt

Each GPIO pin on the chip supports external interrupts and can be configured for falling edge, rising edge, or dual edge triggers. The edge-triggering type is configured through the Px$_N$EICFG register. For example, to configure pin P13 for falling edge interrupt:

```
P13CFG=0x00;        //Set P13 as GPIO
P1TRIS&=0xF7;       //Set P13 as input
P13EICFG=0x02;      //Set P13 to falling edge trigger
```

GPIO interrupts occupy four interrupt vectors:

Pin P0 occupies interrupt vector 0x003B;

Pin P1 occupies interrupt vector 0x0043;

Pin P2 occupies interrupt vector 0x004B;

Pin P3 occupies interrupt vector 0x0053;

When an interrupt occurs, the interrupt service routine (ISR) can first determine which port triggered the interrupt and then proceed with the appropriate handling.

6.3 Interrupts and sleep wake-up

In sleep mode (STOP wake-up mode), each external interrupt can be set to wake the system.

For INT0/INT1 interrupts to wake the system, the corresponding interrupt enable bits and global interrupt enable must be activated. The wake-up method will be falling edge (the wake-up method for INT0/INT1 is independent of the interrupt trigger type selection bits IT0/IT1).

For GPIO interrupt wake system, it is recommended to configure the edge triggering method for the corresponding ports before entering sleep mode (the wake-up method for GPIO is the same as the interrupt trigger edge type and can be set for rising edge, falling edge, or dual edge). Additionally, the relevant interrupt enable bits and global interrupt enable must be turned on.

Once the system is awakened by an external interrupt, it first enters the interrupt service routine to handle the wake-up task. After exiting the ISR, the system continues executing instructions following the sleep operation.

6.4 Interrupt registers

6.4.1 Interrupt mask registers

6.4.1.1 Interrupt mask register (IE)

The Interrupt Enable Register (IE) is a read/write register that supports bit manipulation. When an interrupt condition occurs, the corresponding interrupt flag will be set to 1, regardless of the state of the corresponding interrupt enable bit or the global enable bit (EA). User software should ensure that the relevant interrupt flag is cleared before enabling an interrupt.

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 EA: Global interrupt enable bit
1= Enable all unmasked interrupts
0= Disable all interrupts

Bit6 ES1: UART1 interrupt enable bit
1= Enable UART1 interrupt
0= Disable UART1 interrupt

Bit5 ET2: TIMER2 global interrupt enable bit
1= Enable all TIMER2 interrupts
0= Disable all TIMER2 interrupts

Bit4 ES0: UART0 interrupt enable bit
1= Enable UART0 interrupt
0= Disable UART0 interrupt

Bit3 ET1: TIMER1 interrupt enable bit
1= Enable TIMER1 interrupt
0= Disable TIMER1 interrupt

Bit2 EX1: External interrupt 1 enable bit
1= Enable external interrupt 1
0= Disable external interrupt 1

Bit1 ET0: TIMER0 interrupt enable bit
1= Enable TIMER0 interrupt
0= Disable TIMER0 interrupt

Bit0 EX0: External interrupt 0 enable bit
1= Enable external interrupt 0
0= Disable external interrupt 0

6.4.1.2 Interrupt mask register (EIE2)

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	SPIIE:	SPI interrupt enable bit
	1=	Enable SPI interrupt
	0=	Disable SPI interrupt
Bit6	I2CIE:	I^{2}C interrupt enable bit
	1=	Enable I^{2}C interrupt
	0=	Disable I^{2}C interrupt
Bit5	WDTIE:	WDT interrupt enable bit
	1=	Enable WDT overflow interrupt
	0=	Disable WDT overflow interrupt
Bit4	ADCIE:	ADC interrupt enable bit
	1=	Enable ADC interrupt
	0=	Disable ADC interrupt
Bit3	PWMIE:	PWM global interrupt enable bit
	1=	Enable all PWM interrupts
	0=	Disable all PWM interrupts
Bit2	--	Reserved, set to 0.
Bit1	ET4:	Timer4 interrupt enable bit
	1=	Enable Timer4 interrupt
	0=	Disable Timer4 interrupt
Bit0	ET3:	Timer3 interrupt enable bit
	1=	Enable Timer3 interrupt
	0=	Disable Timer3 interrupt

6.4.1.3 Timer2 interrupt mask register (T2IE)

0xCF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IE	T2OVIE	T2EXIE	--	--	T2C3IE	T2C2IE	T2C1IE	T2C0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 T2OVIE: Timer2 overflow interrupt enable bit
1= Enable interrupt
0= Disable interrupt

Bit6 T2EXIE: Timer2 external load interrupt enable bit
1= Enable interrupt
0= Disable interrupt

Bit5~Bit4 --: Reserved, set to 0.

Bit3 T2C3IE: Timer2 compare/capture channel 3 interrupt enable bit
1= Enable interrupt
0= Disable interrupt

Bit2 T2C2IE: Timer2 compare/capture channel 2 interrupt enable bit
1= Enable interrupt
0= Disable interrupt

Bit1 T2C1IE: Timer2 compare/capture channel 1 interrupt enable bit
1= Enable interrupt
0= Disable interrupt

Bit0 T2C0IE: Timer2 compare/capture channel 0 interrupt enable bit
1= Enable interrupt
0= Disable interrupt

If Timer2’s interrupt is enabled, Timer2’s general interrupt enable bit ET2=1 (IE.5=1) must also be enabled.

6.4.1.4 P0 port interrupt control register (P0EXTIE)

0xAC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0EXTIE	--	--	P05IE	P04IE	P03IE	P02IE	P01IE	P00IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 --: Reserved, set to 0.

Bit5~Bit0 P0iIE: P0i port interrupt enable bit (i=0-5)
1= Enable interrupt
0= Disable interrupt

6.4.1.5 P1 port interrupt control register (P1EXTIE)

0xAD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1EXTIE	P17IE	P16IE	P15IE	P14IE	P13IE	P12IE	P11IE	P10IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 P1iIE: P1i port interrupt enable bit (i=0-7)
1= Enable interrupt
0= Disable interrupt

6.4.1.6 P2 port interrupt control register (P2EXTIE)

0xAE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2EXTIE	P27IE	P26IE	P25IE	P24IE	P23IE	P22IE	P21IE	P20IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 P2iIE: P2i port interrupt enable bit (i=0-7)
1= Enable interrupt
0= Disable interrupt

6.4.1.7 P3 port interrupt control register (P3EXTIE)

0xAF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3EXTIE	P37IE	P36IE	P35IE	P34IE	P33IE	P32IE	P31IE	P30IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 P3iIE: P3i port interrupt enable bit (i=0-7)
1= Enable interrupt
0= Disable interrupt

6.4.2 Interrupt priority control register

6.4.2.1 Interrupt priority control register (IP)

The Interrupt Priority Control Register IP is a read/write register that can be bit-manipulated.

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	PS1	PT2	PS0	PT1	PX1	PT0	PX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	--	Reserved, set to 0.
Bit6	PS1:	UART1 interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit5	PT2:	TIMER2 interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit4	PS0:	UART0 interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit3	PT1:	TIMER1 interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit2	PX1:	External interrupt 1 interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit1	PT0:	TIMER0 interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit0	PX0:	External interrupt 0 interrupt priority control bit
	1=	High priority level
	0=	Low priority level

6.4.2.2 Interrupt priority control register (EIP1)

0xB9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP1	PACMP	PLSE_SCM	--	--	PP3	PP2	PP1	PP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	PACMP:	Analog comparator interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit6	PLSE_SCM	Low-speed crystal timer and crystal stop detection interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit5~ Bit4	--	Reserved, set to 0.
Bit3	PP3:	P3 port interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit2	PP2:	P2 port interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit1	PP1:	P1 port interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit0	PP0:	P0 port interrupt priority control bit
	1=	High priority level
	0=	Low priority level

6.4.2.3 Interrupt priority control register (EIP2)

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI interrupt priority control bit
1= High priority level
0= Low priority level

Bit6 PI2C: I^{2}C interrupt priority control bit
1= High priority level
0= Low priority level

Bit5 PWDT: WDT interrupt priority control bit
1= High priority level
0= Low priority level

Bit4 PADC: ADC interrupt priority control bit
1= High priority level
0= Low priority level

Bit3 PPWM: PWM interrupt priority control bit
1= High priority level
0= Low priority level

Bit2 -- Reserved, set to 0.

Bit1 PT4: TIMER4 interrupt priority control bit
1= High priority level
0= Low priority level

Bit0 PT3: TIMER3 interrupt priority control bit
1= High priority level
0= Low priority level

6.4.2.4 Interrupt priority control register (EIP3)

0xBB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP3	--	--	PWWDT	--	--	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, set to 0.

Bit5 PWWDT WWDT interrupt priority control bit
1= High priority level
0= Low priority level

Bit4~Bit0 -- Reserved, set to 0.

6.4.3 Interrupt flag bit register

6.4.3.1 Timer0/1, INT0/1 interrupt flag bit register (TCON)

0x88	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
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
Reset value	0	0	0	0	0	0	0	0

Bit7 TF1: Timer1 counter overflow interrupt flag bit
1= Timer1 counter overflow, automatically cleared by hardware when entering the interrupt service routine, can also be cleared by software
0= No Timer1 counter overflow

Bit6 TR1: Timer1 run control bit
1= Timer1 start
0= Timer1 stop

Bit5 TF0: Timer0 counter overflow interrupt flag bit
1= Timer0 counter overflow, automatically cleared by hardware when entering the interrupt service routine, can also be cleared by software
0= No Timer0 counter overflow

Bit4 TR0: Timer0 run control bit
1= Timer0 start
0= Timer0 stop

Bit3 IE1: External interrupt 1 flag bit
1= External interrupt 1 triggered, automatically cleared by hardware when entering the interrupt service routine, can also be cleared by software
0= No external interrupt 1 triggered

Bit2 IT1: External interrupt 1 trigger mode control bit
1= Falling edge trigger
0= Low level trigger

Bit1 IE0: External interrupt 0 flag bit
1= External interrupt 0 triggered, automatically cleared by hardware when entering the interrupt service routine, can also be cleared by software
0= No external interrupt 0 triggered

Bit0 IT0: External interrupt 0 trigger mode control bit
1= Falling edge trigger
0= Low level trigger

6.4.3.2 Timer2 interrupt flag bit register (T2IF)

0xC9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IF	TF2	T2EXIF	--	--	T2C3IF	T2C2IF	T2C1IF	T2C0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 TF2: Timer2 counter overflow interrupt flag bit
1= Timer2 counter overflow, requires software to clear
0= No Timer2 counter overflow

Bit6 T2EXIF: Timer2 external load flag bit
1= A falling edge is generated at T2EX port of Timer2, requires software to clear.
0= --

Bit5~Bit4 --: Reserved, set to 0.

Bit3 T2C3IF: Timer2 compare/capture channel 3 flag bit
1= Timer2 compare channel 3 {CCH3:CCL3}={TH2:TL2} or capture channel 3 has generated a capture operation, requires software to clear.
0= --

Bit2 T2C2IF: Timer2 compare/capture channel 2 flag bit
1= Timer2 compare channel 2 {CCH2:CCL2}={TH2:TL2} or capture channel 2 has generated a capture operation, requires software to clear.
0= --

Bit1 T2C1IF: Timer2 compare/capture channel 1 flag bit
1= Timer2 compare channel 1 {CCH1:CCL1}={TH2:TL2} or capture channel 1 has generated a capture operation, requires software to clear.
0= --

Bit0 T2C0IF: Timer2 compare/capture channel 0 flag bit
1= Timer2 compare channel 0{RLDH:RLDL}={TH2:TL2} or capture channel 0 has generated a capture operation, requires software to clear.
0= --

6.4.3.3 Peripheral interrupt flag register (EIF2)

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	--	TF4	TF3
R/W	R	R	--	R/W	R	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIF: SPI global interrupt indicator bit, read only
1= SPI interrupt occurred (this flag is automatically cleared after clearing the specific interrupt flag)
0= No SPI interrupt occurred
Bit6 I2CIF: I^{2}C total interrupt indicator (read-only)
1= I^{2}C interrupt occurred (this flag is automatically cleared after clearing the specific interrupt flag)
0= No I^{2}C interrupt occurred
Bit5 -- Reserved, set to 0.
Bit4 ADCIF: ADC interrupt flag bit
1= ADC conversion completed, needs to be cleared by software
0= ADC conversion not completed
Bit3 PWMIF: PWM total interrupt indicator (read-only)
1= PWM interrupt occurred (this flag is automatically cleared after clearing the specific interrupt flag)
0= No PWM interrupt occurred.
Bit2 -- Reserved, set to 0.
Bit1 TF4: Timer4 overflow interrupt flag bit
1= Timer4 overflow, automatically cleared by hardware upon entering the interrupt service routine, can also be cleared by software
0= No Timer4 overflow
Bit0 TF3: Timer3 overflow interrupt flag bit
1= Timer3 overflow, automatically cleared by hardware upon entering the interrupt service routine, can also be cleared by software
0= No Timer3 overflow

6.4.3.4 SPI interrupt flag bit register (SPSR)

0xED	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSR	SPISIF	WCOL	--	--	--	--	--	SSCEN
R/W	R	R	--	--	--	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPISIF: SPI transmission complete interrupt flag bit, read only.
1= SPI transmission complete (first read SPSR, then read/write SPDR to clear)
0= SPI is not completed
Bit6 WCOL: SPI write collision interrupt flag, read only.
1= A write operation to SPDR causes a collision when SPI transmission is not completed (first read SPSR, then read/write SPDR to clear).
0= No write collision
Bit5~Bit1 -- Reserved, set to 0.
Bit0 SSCEN: SPI master mode NSS output control bit
1= NSS output high when SPI is in idle state
0= NSS output register SSCR contents

6.4.3.5 I^{2}C master mode interrupt flag bit register (I2CMCR/I2CMSR)

0xF5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMCR	RSTS	--	--	--	ACK	STOP	START	RUN
I2CMSR	I2CMIF	BUS_BUSY	IDLE	ARB_LOST	DATA_ACK	ADDR_ACK	ERROR	BUSY
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 RSTS: I^{2}C active module reset control bit
1= Reset the master control module (all I^{2}C registers of the master control module, including I2CMSR)
0= Clear the interrupt flag bit in I^{2}C master mode to 0
I2CMIF: I^{2}C master mode interrupt flag bit
1= In master mode, indicates completion of send/receive or a transmission error. (cleared by software; write 0 to clear)
0= No interrupt occurred

Bit6~Bit0 I^{2}C master mode control and status bits, see I2CM description for details.

6.4.3.6 I^{2}C slave mode status register (I2CSSR)

0xF2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSSR	--	--	--	--	--	SENDFIN	TREQ	RREQ
R/W	--	--	--	--	--	R	R	R
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit3 --: Reserved, set to 0.

Bit2 SENDFIN: I^{2}C slave mode transmission complete flag bit, read-only
1= The master device no longer needs data; TREQ is no longer set to 1, indicating that the current data transfer is completed. (automatically cleared after reading I2CSCR)
0= --

Bit1 TREQ: I^{2}C slave mode ready to send flag, read-only
1= The slave device has been addressed, and the master device is ready to receive data. (automatically cleared after writing to I2CSBUF)
0= --

Bit0 RREQ: I^{2}C slave mode reception complete flag bit, read-only
1= Reception is completed. (automatically cleared after reading I2CSBUF)
0= Not completed

The relevant status bits in I^{2}C slave mode are also interrupt flag bits

Caution: The I^{2}C master mode interrupt and slave mode interrupt share the same interrupt vector (00ABH).

6.4.3.7 UART control register (SCONn)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCONn	UnSM0	UnSM1	UnSM2	UnREN	UnTB8	UnRB8	TIn	RIn
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register SCON0 address: 0x98; Register SCON1 address: 0xEA.

Bit7~Bit2 U1SM0, U1SM1, U1SM2, U1REN, U1TB8, U1RB8: UART1 related control bits, see UARTn description

Bit1 TIn: Transmission interrupt flag (requires software to clear)
1= Indicates that the transmission buffer is empty, and a new frame of data can be sent.
0= --

Bit0 RIn: Reception interrupt flag (requires software to clear)
1= Indicates that the reception buffer is full, and after reading, the next frame of data can be received.
0= --

TIn and RIn share the same interrupt vector, and it is necessary to check to determine whether it is a reception interrupt or a transmission interrupt.

6.4.3.8 P0 port interrupt flag bit register (P0EXTIF)

0xB4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0EXTIF	--	--	P05IF	P04IF	P03IF	P02IF	P01IF	P00IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, set to 0.

Bit5~Bit0 P0iIF: P0i port interrupt flag bit (i=0-5)
1= P0i port generates an interrupt, requires software to clear.
0= P0i port has not generated an interrupt.

6.4.3.9 P1 port interrupt flag bit register (P1EXTIF)

0xB5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P1EXTIF	P17IF	P16IF	P15IF	P14IF	P13IF	P12IF	P11IF	P10IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 P1iIF: P1i port interrupt flag bit (i=0-7)
1= P1i port generates an interrupt, requires software to clear.
0= P1i port has not generated an interrupt.

6.4.3.10 P2 port interrupt flag bit register (P2EXTIF)

0xB6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P2EXTIF	P27IF	P26IF	P25IF	P24IF	P23IF	P22IF	P21IF	P20IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 P2iIF: P2i port interrupt flag bit (i=0-7)
1= P2i port generates an interrupt, requires software to clear.
0= P2i port has not generated an interrupt.

6.4.3.11 P3 port interrupt flag bit register (P3EXTIF)

0xB7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P3EXTIF	P37IF	P36IF	P35IF	P34IF	P33IF	P32IF	P31IF	P30IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 P3iIF: P3i port interrupt flag bit (i=0-7)
1= P3i port generates an interrupt, requires software to clear.
0= P3i port has not generated an interrupt.

6.4.4 Clearing interrupt flag bits

The clearing of interrupt flags can be divided into the following types:

- Hardware automatic clearing (requires entering the interrupt service routine)
- Software clearing
- Read/write operation clearing

1) Hardware automatic clearing flag bits

The flag bits that support hardware automatic clearing include those generated by interrupts from INT0, INT1, T0, T1, T3, and T4. The condition for hardware automatic clearing of these flag bits is that the global interrupt enable bit EA=1, and the corresponding interrupt enable bits are also set. After the interrupt is generated, the system enters the corresponding interrupt service routine, and the flag bits are automatically cleared. If the interrupt enable is turned off, these flag bits can also be cleared using software.

2) Software clearing flag bits

There are flag bits in the system that can only be cleared by software. These flag bits will not be automatically cleared upon entering the interrupt service routine and require the software to write 0 to clear them. Otherwise, the system may re-enter the interrupt service routine after exiting.

3) Read/write operation clearing flag bits

Some flag bits in the system cannot be cleared by simply writing 0 to them; they need to be cleared by reading/writing other registers. For example, the transmission complete flag SPISIF in the SPI interrupt flag register, which, when set to 1, needs to read SPSR first and then read/write SPDR to clear.

When performing software clearing operations, it should be noted that if multiple interrupt flag bits are in the same register and their occurrences are unrelated, it is not recommended to use the read-modify-write operation. For instance, in the PWMUIF interrupt flag register, which contains the up comparison interrupts for channels PG0-PG5, these interrupt flag bits are unrelated. When an up comparison interrupt occurs for PG0, the value of PWMUIF is 0x01. If a read-modify-write operation is performed to clear this flag:

```
PWMUIF &= 0xFE;
```

This operation is specifically implemented by first reading the value of PWMUIF back to the CPU, then performing the calculation, and finally writing it back to PWMUIF. If the interrupt flag PWMUIF[1] for PG1 is set to 1 after the CPU reading, while PWMUIF[1] was 0 during the read, then after the calculation, writing back to PWMUIF[1] will also result in 0, thus clearing the already generated up interrupt flag PWMUIF[1] for PG1.

To clear the interrupt flag bits of the above types, it is recommended to write 0 directly, and write 1 to the other unrelated flag bits: PWMUIF = 0xFE. This operation has no practical effect on writing 1 to the unrelated interrupt flag bits.

6.4.5 Special interrupt flag bits in debug mode

In the system, there are certain flags that cannot be cleared by simply writing 0 to them; instead, they need to be cleared by reading or writing to other registers.

In debug mode, after breakpoint execution, single-step operation or stopping operation, the emulator will read out the values of all registers from the system to the emulation software, and the read/write operation of the emulator is exactly the same as that in normal mode.

Therefore, during the debugging process, the interrupt flag bit should be set to 1 after a pause, but it is displayed as 0 in the watch window.

Example: The transmission complete flag (SPISIF) in the SPI interrupt flag register in debug mode:

```
...                              //Set ports and enable interrupts
SPDR = 0x56;                 //Transmit SPDR data
delay();
...

void SPI_int (void) interrupt SPI_VECTOR     // SPI interrupt service routine
{
        O1   _nop_();           //Set breakpoint 1
             _nop_();
        O2   k = SPSR;          //Set breakpoint 2
          _nop_();
        ...
}
```

When running at Breakpoint 1, after stopping, the SPI transmission operation is completed, and a transmission complete interrupt has been generated, so SPSR.7=1. At this point, the emulator has already completed a read operation for all registers (including reading SPSR).

Upon resuming execution and stopping at Breakpoint 2, the emulator performs another read operation for all registers (including SPDR), resulting in SPSR.7=0. This situation will also occur when running twice in single-step, which is important to be aware of in debug mode.

7. I/O Ports

7.1 GPIO function

The chip features four sets of I/O ports: PORT0, PORT1, PORT2, and PORT3.

Each PORTx is a bidirectional port, and its corresponding data direction register is denoted as PxTRIS. Setting a bit in PxTRIS to 1 configures the corresponding pin as an output. Clearing a bit in PxTRIS to 0 configures the corresponding PORTx pin as an input.

When PORTx operates as an output, writing to the Px register will write to the port latch. All write operations are performed as read-modify-write operations. Therefore, writing to a port means first reading the pin level of that port, then modifying the read value, and finally writing the modified value to the port data latch.

When PORTx operates as an output, reading the Px register is influenced by the settings of the PxDS register. If a bit in PxDS is set to 1, reading the corresponding bit in Px will yield the actual pin state. If a bit in PxDS is cleared to 0, reading the corresponding bit in Px will give the state of the port data latch. When PORTx functions as an input, reading the Px register reflects the actual pin state and is independent of the PxDS register settings.

When PORTx pins are used as analog inputs, users must ensure that the corresponding bits in the PxTRIS register remain set to 0. I/O pins configured as analog inputs will always read as 0.

Relevant registers associated with PORTx include Px, PxTRIS, PxOD, PxUP, PxRD, PxDR, PxSR, PxDS, PxMODE, etc.

7.1.1 PORTx data register (Px)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Px	Px7	Px6	Px5	Px4	Px3	Px2	Px1	Px0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	X	X	X	X	X	X	X	X

Register P0 address: 0x80; Register P1 address: 0x90; Register P2 address: 0xA0; Register P3 address: 0xB0.

Bit7~Bit0	Px<7:0>:		Px I/O pin bit
		1=	Port pin level>V_{IH} (positive threshold voltage)
		0=	Port pin level<V_{IL} (negative threshold voltage)

7.1.2 PORTx direction register (PxTRIS)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxTRIS	PxTRIS7	PxTRIS6	PxTRIS5	PxTRIS4	PxTRIS3	PxTRIS2	PxTRIS1	PxTRIS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register P0TRIS address: 0x9A; Register P1TRIS address: 0xA1;

Register P2TRIS address: 0xA2; Register P3TRIS address: 0xA3.

Bit7~Bit0	PxTRIS<7:0>:		Tri-state control bit
		1=	The pin is configured as an output
		0=	The pin is configured as an input (tri-state)

Note:

- When the port is configured as an output, reading the port returns the value of the output register.
- When the port is configured as an input, any read-modify-write type instructions applied to the port actually operate on the output register.

7.1.3 PORTx open-drain control register (PxOD)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxOD	PxOD7	PxOD6	PxOD5	PxOD4	PxOD3	PxOD2	PxOD1	PxOD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register P0OD address: F009H; Register P1OD address: F019H;

Register P2OD address: F029H; Register P3OD address: F039H.

Bit7~Bit0 PxOD<7:0>: Open-drain control bit
1= The pin is configured as open-drain (output is open-drain)
0= The pin is configured as normal (output is push-pull)

7.1.4 PORTx pull-up resistor control register (PxUP)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxUP	PxUP7	PxUP6	PxUP5	PxUP4	PxUP3	PxUP2	PxUP1	PxUP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register P0UP address: F00AH; Register P1UP address: F01AH;

Register P2UP address: F02AH; Register P3UP address: F03AH.

Bit7~Bit0 PxUP<7:0>: Pull-up resistor control bit
1= Pull-up resistor enabled
0= Pull-up resistor disabled

7.1.5 PORTx pull-down resistor control register (PxRD)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxRD	PxRD7	PxRD6	PxRD5	PxRD4	PxRD3	PxRD2	PxRD1	PxRD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register P0RD address: F00BH; Register P1RD address: F01BH;

Register P2RD address: F02BH; Register P3RD address: F03BH.

Bit7~Bit0 PxRD<7:0>: Pull-down resistor control bit
1= Pull-down resistor enabled
0= Pull-down resistor disabled

Note: The control of the pull-down resistor is independent of the GPIO configuration and multiplexing functions; it is controlled separately by the PxRD register.

7.1.6 PORTx drive current control register (PxDR)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxDR	PxDR7	PxDR6	PxDR5	PxDR4	PxDR3	PxDR2	PxDR1	PxDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register P0DR address: F00CH; Register P1DR address: F01CH;

Register P2DR address: F02CH; Register P3DR address: F03CH.

Bit7~Bit0 PxDR<7:0>: Drive current control bit (effective when the port is configured as an output)

1= Weak drive

0= Strong drive

7.1.7 PORTx slope control register (PxSR)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxSR	PxSR7	PxSR6	PxSR5	PxSR4	PxSR3	PxSR2	PxSR1	PxSR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register P0SR address: F00DH; Register P1SR address: F01DH;

Register P2SR address: F02DH; Register P3SR address: F03DH.

Bit7~Bit0 PxSR<7:0>: Px slope control register (effective when the port is configured as an output)

1= Px pin with slow slope

0= Px pin with fast slope

7.1.8 PORTx data input selection register (PxDS)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxDS	PxDS7	PxDS6	PxDS5	PxDS4	PxDS3	PxDS2	PxDS1	PxDS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register P0DS address: F00EH; Register P1DS address: F01EH;

Register P2DS address: F02EH; Register P3DS address: F03EH.

Bit7~Bit0 PxDS<7:0>: Data input select bit, when configured as GPIO, affects the value of the Px register read.

1= Both output and input modes read the pin state

(The Schmitt trigger remains enabled when the port is set to output.)

0= Output mode: data latch state

Input mode: pin state

Note: To read the pin state when the port is configured for multiplexed function input, the direction control for that port must be set to input mode.

7.1.9 PORTx input level selection register (PxMODE)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxMODE	PxMODE7	PxMODE6	PxMODE5	PxMODE4	PxMODE3	PxMODE2	PxMODE1	PxMODE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register P0MODE address: F00FH; Register P1MODE address: F01FH;

Register P2MODE address: F02FH; Register P3MODE address: F03FH.

Bit7~Bit0	PxMODE<7:0>:		Data input level selection
		1=	TTL level input
		0=	Schmitt input

7.2 Multiplexing function

7.2.1 Port multiplexing function configuration register

PORTx function configuration register (PxnCFG)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PxnCFG	--	--	--	PxnCFG4	PxnCFG3	PxnCFG2	PxnCFG1	PxnCFG0
R/W	--	--	--	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit5 -- Reserved, set to 0.

Bit4~Bit0 PxnCFG<4:0>: Function configuration bit, default is GPIO function. For details, please refer to Port Function Configuration.

The Px function configuration registers consist of 8 registers, including Px0CFG to Px7CFG, which control the functionality of Px0 to Px7.

Each port has a corresponding function configuration register, PxNCFG, allowing each port to be set to its respective digital function. For example, to configure P24 for the BEEP buzzer function, you would set: P24CFG = 0x18.

When the port is configured for multiplexed functions, there is no need to configure the port direction register PxTRIS.

- The pull-up resistor registers for SCL and SDA can be configured to force open-drain output.
- The RXD0 and RXD1 registers can be configured for pull-up resistors or synchronous mode to force the pull-up resistors on.

Other multiplexed functions have their pull-up resistors hardware-forced off and open-drain outputs disabled, making software settings for pull-up resistors (PxUP) or open-drain outputs (PxOD) ineffective.

When the port is multiplexed for SCL and SDA functions, the hardware forces the port to be an open-drain output, and the pull-up resistor can be enabled via software using PxUP.

The port functions are configured as follows.

Configuration value	Function	Direction	Function description
0x00	GPIO	I/O	General I/O ports can be configured through registers for input/output, pull-up/pull-down functions, and other features.
0x01	ANALOG		Analog function
0x02	--	--	--
0x03	--	--	--
0x04	CC0	O	Timer2 compare output channel 0
0x05	CC1	O	Timer2 compare output channel 1
0x06	CC2	O	Timer2 compare output channel 2
0x07	CC3	O	Timer2 compare output channel 3
0x08	TXD0	O	UART0 data output
0x09	RXD0	I/O	UART0 data input / synchronous mode data output
0x0A	TXD1	O	UART1 data output
0x0B	RXD1	I/O	UART1 data input / synchronous mode data output
0x0C	SCL	I/O	I^{2}C clock input/output
0x0D	SDA	I/O	I^{2}C data input/output

Configuration value	Function	Direction	Function description
0x0E	NSS	I/O	SPI slave mode chip select signal (input/output)
0x0F	SCLK	I/O	SPI clock input/output
0x10	MOSI	I/O	SPI master transmit / slave receive
0x11	MISO	I/O	SPI master receive / slave transmit
0x12	PG0	O	PWM channel 0 output
0x13	PG1	O	PWM channel 1 output
0x14	PG2	O	PWM channel 2 output
0x15	PG3	O	PWM channel 3 output
0x16	PG4	O	PWM channel 4 output
0x17	PG5	O	PWM channel 5 output
0x18	BEEP	O	Buzzer output
0x19	--	--	--
0x1A	C0_O	O	Comparator 0 output
0x1B	C1_O	O	Comparator 1 output
0x1C	--	--	--
0x1D	--	--	--
0x1E	--	--	--
0x1F	--	--	--

Note:

1) Configuration values marked as “--” in the table are reserved and must not be used.

2) The Function Configuration Register defaults to 0x00, which configures the port as a GPIO function. Different functionalities can be assigned by setting the corresponding input function configuration registers.

3) When the Function Configuration Register is set to 0x01, digital circuits are disabled to reduce power consumption, rendering GPIO-related register settings ineffective. The port supports multiple analog functions, as described in the table.

4) There are no priority restrictions for output functions among the multiplexed ports. If multiple ports are configured for the same output function, that function will be simultaneously outputted from all those ports.

5) Input functions have a priority order among multiplexed ports. If two or more ports are configured for the same input function, the selection will follow the priority order from high to low: P00, P01, ..., P32, P35.

For example, if both P00 and P32 are configured as RXD0: P00CFG = 0x09; P32CFG = 0x09; Since P00 has a higher priority, the RXD0 signal source will connect to P00, and any data waveform present on P32 will not be considered as the RXD0 signal source.

The corresponding analog functions of the port are as follows:

PIN	CONFIG	1(ANALOG)				Other digital function priority
P00		AN0	C0P1			Highest
P01		AN1	C0P2			
P02		AN2	C1P2			
P03		AN3	C1P1			
P04		AN4	C1P0			
P05		AN5	C1N			
P10		AN27				
P11		AN23				
P12		AN24				
P13		AN6	C0P0			
P14		AN7	C0N			
P15		AN18				
P16		AN19				
P17		AN20				
P20		AN25				
P21	DSCK1	AN21				
P22	OSCIN1	AN8		OP1_P		
P23	OSCOUT1	AN9		OP1_N		
P24	-	AN10	C0P5/C1P5	OP1_O		
P25	DSCK2	AN11				
P26	-	AN12			PGATO	
P27		AN26				
P30	-	AN22	C0P4/C1P4	OP0_O	-	
P31	OSCIN2	AN13	-	OP0_N	PGAP	
P32	OSCOUT2	AN14	-	OP0_P	PGAGND	
P33		AN28				
P34		AN29				
P35	DSDA	AN16				
P36		AN17				
P37		AN30				Lowest

7.2.2 Port input function allocation register

The chip has digital functions that are exclusively for input states, such as INT0/INT1, etc. These types of digital input functions are independent of the port multiplexing status. As long as the allocated port supports digital input (for example, RXD0 as a digital input or GPIO configured for input functionality), that port will support the specified function.

The input function port allocation registers are as follows.

Input function allocation register	Address	Function	Function description
PS_INT0	F0C0H	INT0	External interrupt 0 input port allocation register
PS_INT1	F0C1H	INT1	External interrupt 1 input port allocation register
PS_T0	F0C2H	T0	Timer 0 external clock input port allocation register
PS_T0G	F0C3H	T0G	Timer 0 gate control input port allocation register
PS_T1	F0C4H	T1	Timer 1 external clock input port allocation register
PS_T1G	F0C5H	T1G	Timer 1 gate control input port allocation register
PS_T2	F0C6H	T2	Timer 2 external event or gate control input port allocation register
PS_T2EX	F0C7H	T2EX	Timer 2 falling edge auto reload input port allocation register
PS_CAP0	F0C8H	CAP0	Timer 2 input capture channel 0 port allocation register
PS_CAP1	F0C9H	CAP1	Timer 2 input capture channel 1 port allocation register
PS_CAP2	F0CAH	CAP2	Timer 2 input capture channel 2 port allocation register
PS_CAP3	F0CBH	CAP3	Timer 2 input capture channel 3 port allocation register
PS_ADET	F0CCH	ADET	ADC external trigger input port allocation register
PS_FB	F0CDH	FB	PWM external brake signal input port allocation register

PS_XX input function port allocation register PS_XX (as described in the table above)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PS_XX	--	--	PS_XX5	PS_XX4	PS_XX3	PS_XX2	PS_XX1	PS_XX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	1	1	1	1	1	1	1

Bit7~Bit6	--	Reserved, set to 0.
Bit5~Bit0	PS_XX<5:0>:	Input function allocation control bit (based on the actual port of the chip, unused values are reserved and prohibited);
	0x00=	Allocated to port P00
	0x01=	Allocated to port P01
	……	……
	0x14=	Allocated to port P14
	0x15=	Allocated to port P15
	……	……
	0x35=	Allocated to port P35
	0x36=	Allocated to port P36
	……	……
	0x3F=	No port allocated

1) The input function allocation structure supports multiple functions being assigned to the same port. For example, both INT0 and CAP0 can be assigned to port P00 with the following configuration:

```
P00CFG = 0x00;      //Configure P00 as GPIO function
P0TRIS = 0x00;      //Set P00 as GPIO input
PS_INT0 = 0x00;     //Assign INT0 functionality to P00
PS_CAP0 = 0x00;     //Assign CAP0 functionality to P00
```

2) This input function allocation structure is relatively independent and can be used simultaneously with other multiplexed function ports. In this case, there is no need to configure the direction register of the corresponding port. For instance, RXD0 and INT0 can be assigned to port P00 as follows:

```
P00CFG = 0x09;      //Configure P00 for UART0 RXD0 function
PS_INT0 = 0x00;     //Assign INT0 functionality to P00
```

3) The input function configuration structure can also be used alongside external interrupt functionality. For example, both CAP0 and GPIO interrupt functions can be assigned to port P00 with the following configuration:

```
P00CFG = 0x00;      //Configure P00 as GPIO function
P0TRIS = 0x00;      //Set P00 as GPIO input
PS_CAP0 = 0x00;     //Assign CAP0 functionality to P00
P00EICFG = 0x01;    //Configure P00 for rising edge trigger interrupt
P0EXTIE = 0x01;     //Enable external interrupt on port P00
```

7.2.3 External interrupt control register

When using external interrupts, the port must be configured as a GPIO function and set to input direction. Alternatively, the port can be multiplexed for input functions (such as RXD0 or RXD1). Each port can be configured to support GPIO interrupt functionality.

PORTx external interrupt control register (Px$_N$EICFG)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Px$_N$EICFG	--	--	--	--	--	--	Px1EICFG1	Px0EICFG0
R/W	--	--	--	--	--	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit2 -- Reserved, set to 0.
Bit1~Bit0 Px$_N$EICFG<1:0>: Px$_N$ external interrupt control bit
00= External interrupt disable
01= Rising edge trigger interrupt
10= Falling edge trigger interrupt
11= Rising or falling edge trigger interrupt

There are 8 external interrupt control registers in Px, including Px0EICFG~Px7EICFG, which control the external interrupts of Px0~Px7 respectively.

To configure P00 for a falling edge trigger interrupt, the setup is as follows:

```
P00CFG =0x00;        //Configure P00 as GPIO function
P0TRIS =0x00;        //Set P00 as input
P00EICFG = 0x02;     //Configure P00 for falling edge trigger interrupt
EA = 1;              //Enable global interrupts
P0EXTIE = 0x01;      //Enable external interrupt functionality for P00
```

7.2.4 Considerations for multiplexed function applications

1) The input functionality of multiplexed functions and the external interrupt (GPIO interrupt) for the port are relatively independent in structure.

For example, if P01 is configured as RXD0 and the GPIO interrupt trigger for P01 is set to rising edge with the interrupt enabled, the GPIO interrupt for P01 will be triggered when the input on P01 changes from low to high.

2) The input structure for digital signals is not affected by the system configuration state.

For instance, if P01 is powered up and configured as an external reset pin, the input module for that port is enabled. If P01 is also configured as INT0 with the interrupt enabled in the program, the interrupt service routine will execute before the valid sampling time for the reset signal, followed by the reset operation.

3) It is important to note that in debug mode, if the multiplexed function is configured for the DSDA pin, its input function remains effective. It is recommended not to configure related multiplexed functions for the DSDA pin in debug mode.

4) When the port is used for analog functions and the function configuration register is set to 0x01, the hardware disables the digital circuitry to reduce power consumption, making the GPIO function-related registers ineffective.

8. Watchdog Timer (WDT)

8.1 Overview

The Watchdog Timer (WDT) is an on-chip timer with a selectable overflow time, sourced by the system clock (Fsys).

When the watchdog timer counts to the set overflow value, it generates a watchdog overflow interrupt flag (WDTIF=1). If global interrupts are enabled (EA=1) and the watchdog timer interrupt is enabled (EIE2[5]=1), the CPU will execute the interrupt service routine and clear the watchdog counter by writing to the register WDCON[0]=1. After the watchdog counter is cleared, it starts counting from 0 again until the next timer overflow.

When the watchdog timer overflows, if the watchdog overflow reset is enabled (WDCON[1]=1) and the watchdog counter is not cleared, a watchdog overflow reset will occur. The watchdog overflow reset is a protective mechanism for the system; when the system enters an unknown state, the watchdog can reset the system to prevent it from getting stuck in an indefinite loop. For details on the watchdog overflow reset, please refer to the reset section.

8.2 Relevant registers

8.2.1 Watchdog control register (WDCON)

0x97	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WDCON	SWRST	PORF	EXTIF	FIXIF	WDTIF	WDTRF	WDTRE	WDTCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	1	0	0	0	0	0	0

Bit7 SWRST: Software reset control bit
1: Execute a system software reset (write 0 to clear after reset).
0: --

Bit6 PORF: Power-on reset flag
1: Power-on reset (write 0 to clear, no TA write timing required).
0: --

Bit5 EXTIF: External reset flag
1= External reset (write 0 to clear, no TA write timing required);
0= --

Bit4 FIXIF: CONFIG state protection bit reset flag bit
1= CONFIG state protection reset (write 0 to clear, no TA write timing required);
0= --

Bit3 WDTIF: WDT overflow interrupt flag bit
1= WDT overflow (write 0 to clear)
0= No WDT overflow

Bit2 WDTRF: WDT reset flag bit
1= The system is reset by the WDT (write 0 to clear)
0= The system is not reset by the WDT

Bit1 WDTRE: WDT reset enable bit
1= Enable the WDT to reset the CPU
0= Disable the WDT to reset the CPU

Bit0 WDTCLR: WDT counter clear bit
1= Clear the WDT counter (automatically zeroed by hardware)
0= Disable WDT counter (write 0 is invalid).

Note:

1. If the WDT is configured as ENABLE in the CONFIG, then the WDT is always enabled, regardless of the state of the WDTRE control bit. Additionally, the WDT overflow reset function is forcibly enabled.
2. If the WDT is configured as SOFTWARE CONTROL in the CONFIG, then the WDTRE control bit can be used to enable or disable the WDT.

The instruction sequence required to modify the WDCON register (no other instructions may be inserted in between) is as follows:

```
MOV                 TA,#0AAH
MOV                 TA,#055H
ORL                 WDCON,#01H
```

8.2.2 Watchdog overflow control register (CKCON)

0x8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	WTS2	WTS1	WTS0	T1M	T0M	--	--	T0CNTM
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	1	1	1

Bit7~Bit5 WTS<2:0>: WDT overflow time select bit
000= 2^{17}*Tsys
001= 2^{18}*Tsys
010= 2^{19}*Tsys
011= 2^{20}*Tsys
100= 2^{21}*Tsys
101= 2^{22}*Tsys
110= 2^{24}*Tsys
111= 2^{26}*Tsys

Bit4 T1M: Timer1 clock source select bit
0= Fsys/12
1= Fsys/4

Bit3 T0M: Timer0 clock source select bit
0= Fsys/12
1= Fsys/4

Bit2~Bit1 --: Reserved, set to 1.

Bit0 T0CNTM: Timer0 count source select bit
0= PWM0 output
1= T0 pin input

8.3 WDT interrupt

The watchdog timer can enable or disable interrupts via the EIE2 register and set high/low priority through the EIP2 register. The interrupt-related bits are as follows:

8.3.1 Interrupt mask register (EIE2)

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	SPIIE:	SPI interrupt enable bit
	1=	Enable SPI interrupt
	0=	Disable SPI interrupt
Bit6	I2CIE:	I^{2}C interrupt enable bit
	1=	Enable I^{2}C interrupt
	0=	Disable I^{2}C interrupt
Bit5	WDTIE:	WDT interrupt enable bit
	1=	Enable WDT overflow interrupt
	0=	Disable WDT overflow interrupt
Bit4	ADCIE:	ADC interrupt enable bit
	1=	Enable ADC interrupt
	0=	Disable ADC interrupt
Bit3	PWMIE:	PWM global interrupt enable bit
	1=	Enable PWM global interrupt
	0=	Disable PWM global interrupt
Bit2	--	Reserved, set to 0.
Bit1	ET4:	Timer4 interrupt enable bit
	1=	Enable Timer4 interrupt
	0=	Disable Timer4 interrupt
Bit0	ET3:	Timer3 interrupt enable bit
	1=	Enable Timer3 interrupt
	0=	Disable Timer3 interrupt

8.3.2 Interrupt priority control register (EIP2)

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI interrupt priority control bit
1= High priority level
0= Low priority level

Bit6 PI2C: I^{2}C interrupt priority control bit
1= High priority level
0= Low priority level

Bit5 PWDT: WDT interrupt priority control bit
1= High priority level
0= Low priority level

Bit4 PADC: ADC interrupt priority control bit
1= High priority level
0= Low priority level

Bit3 PPWM: PWM interrupt priority control bit
1= High priority level
0= Low priority level

Bit2 --: Reserved, set to 0.

Bit1 PT4: TIMER4 interrupt priority control bit
1= High priority level
0= Low priority level

Bit0 PT3: TIMER3 interrupt priority control bit
1= High priority level
0= Low priority level

9. Windowed Watchdog Timer (WWDT)

9.1 Overview

The windowed watchdog timer is a 5-bit down-counting timer with an optional window comparison time. It is clocked by an LSI-provided clock source with selectable prescaling. This timer can generate interrupts, wake the system from sleep mode, and reset the chip.

9.2 Relevant registers

9.2.1 WWDT control register (WWCON0)

0xE5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WWCON0	WWDTPSC3	WWDTPSC2	WWDTPSC1	WWDTPSC0	WWDTEN	WWDTRE	WWDTCLR	WWDTRF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

This register is protected by TA.

Bit7~4 WWDTPSC<3:0>: Windowed watchdog prescaler bit

0000= $F_{LSI}/2^{8}$ 1000= $F_{LSI}/2^{16}$

0001= $F_{LSI}/2^{9}$ 1001= $F_{LSI}/2^{17}$

0010= $F_{LSI}/2^{10}$ 1010= $F_{LSI}/2^{18}$

0011= $F_{LSI}/2^{11}$ 1011= $F_{LSI}/2^{19}$

0100= $F_{LSI}/2^{12}$ 1100= $F_{LSI}/2^{20}$

0101= $F_{LSI}/2^{13}$ 1101= $F_{LSI}/2^{21}$

0110= $F_{LSI}/2^{14}$ 1110= $F_{LSI}/2^{22}$

0111= $F_{LSI}/2^{15}$ 1111= $F_{LSI}/2^{22}$

Bit3 WWDTEN: Windowed watchdog module enable bit

1= Enable

0= Disable

Bit2 WWDTRE: Windowed watchdog reset enable bit

1= Enable

0= Disable

Bit1 WWDTCLR: Windowed watchdog timer clear bit

1= Clear timer (write 1 to clear the timer, hardware automatically clears to 0)

0= Writing 0 is invalid

Bit0 WWDTRF: Windowed watchdog reset flag bit

1= Generate a reset

0= Write 0 to clear the reset flag

9.2.2 WWDT control register (WWCON1)

0xE7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WWCON1	FORCE3	FORCE2	FORCE1	FORCE0	MODE	WWDTSLE	WWDTIE	WWDTIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

This register is protected by TA.

Bit7~4 FORCE<3:0>: Windowed watchdog overflow force reset enable bit

A/F= Enable force reset for the window watchdog

1000

Flsi/217

Others= The watchdog reset enable is jointly determined by WWDTEN and WWDTRE.

Bit3 MODE: Windowed watchdog mode selection bit

1= Window feed dog mode (feeding the dog within the range of 0 < counter value < CMPDAT does not cause a reset; otherwise, a reset occurs)

0= Any time feed dog mode (reset occurs when the counter decrements from 1 to 0).

Bit2 WWDTSLE: Windowed watchdog sleep wake-up enable

1= Enable

0= Disable

Bit1 WWDTIE: Windowed watchdog compare interrupt enable

1= Enable

0= Disable

Bit0 WWDTIF: Windowed watchdog compare overflow flag

1= Compare overflow (enabling WWDTIE can generate an interrupt)

0= Write 0 to clear this flag

9.2.3 WWDT compare value register (WWCMPD)

0xE6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WWCMPD	--	--	--	CMPDAT4	CMPDAT3	CMPDAT2	CMPDAT1	CMPDAT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

This register is protected by TA.

Bit7~5 -- Reserved, set to 0.

Bit4~0 CMPDAT<4:0>: Window comparison values

9.3 WWDT interrupt and sleep wakeup

The windowed watchdog timer can enable or disable interrupts through the WWCON1 register, as described above. High/low priority can be set through the EIP3 register, with the interrupt-related bits as follows.

9.3.1 Interrupt priority control register (EIP3)

0xBB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP3	--	--	PWWDT	--	--	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, set to 0.
Bit5 PWWDT WWDT interrupt priority control bit
1= High level interrupt
0= Low level interrupt
Bit4~Bit0 -- Reserved, set to 0.

In any dog feeding mode, when the windowed watchdog timer counts to the window comparison value, it will set the hardware compare overflow flag WWCON1[0] to 1. If global interrupt enable (EA=1) and window watchdog compare interrupt enable (WWCON1[1]=1) are set, the CPU will execute the interrupt service routine. The formula for calculating the window comparison time is as follows:

Window Comparison Time $=\frac{PSC}{125}\times(\text{0x1F-WWCMPD[4:0]})$ ms

Where PSC is the prescaler for the window watchdog, set by WWCON0[7:4].

To use the windowed watchdog for waking from sleep mode, the windowed watchdog module enable bit WWDTEN and the sleep wake-up enable bit WWDTSLE must be enabled before entering sleep. Additionally, the comparison value WWCMPD[4:0] should be set. If global interrupt enable and watchdog compare interrupt enable are activated before sleep, upon waking, the interrupt service routine will be executed first. After returning from the interrupt, the next instruction following the sleep command will be executed.

9.4 Function description

The windowed watchdog has two feeding modes: window feeding mode and anytime feeding mode. When WWCON1[3] is set to 1, it operates in window feeding mode; when set to 0, it operates in anytime feeding mode.

- Window feeding mode

The windowed watchdog timer counts down from 0x1F. When the timer reaches the set window comparison value WWCMPD[4:0], the overflow flag WWCON1[0] is set to 1. At this point, a clear timer operation can be executed (set WWCON0[1] to 1 to clear the current value of the window watchdog timer), allowing the timer to restart counting from 0x1F. If the windowed watchdog timer is cleared before reaching the window comparison value or if the timer counts down to 0, a reset will occur under the condition that the windowed watchdog reset enable (WWCON0[2]=1) is set, and the hardware will set the window watchdog reset flag WWCON0[0] to 1. Therefore, in window feeding mode, the window watchdog timer must be cleared within the window period (0 < count value < WWCMPD).

- Anytime feeding mode

In anytime feeding mode, the windowed watchdog timer counts down from 0x1F, and the timer can be cleared at any time before it reaches 0, allowing it to restart counting from 0x1F. If the timer reaches 0, a reset will occur under the condition that the windowed watchdog reset enable is set, and the hardware will set the window watchdog reset flag to 1. Thus, in anytime feeding mode, the window watchdog timer can be cleared at any moment while the count value is greater than 0.

When the forced reset function of the window watchdog timer overflow is enabled (WWCON1[7:4]=0xa/0xf), the window watchdog timer will be activated regardless of whether the user has configured the window watchdog mode enable and reset enable. When the timer counts down from 0x1F to 0, a system reset will occur. The forced reset function of the windowed watchdog overflow also remains effective in sleep mode.

10. Timer 0/1

Timer 0 and Timer 1 are similar in type and structure, both being 16-bit timers. Timer 1 has three operating modes, while Timer 0 has four. They provide basic timing and event counting operations.

In Timer Mode, the timer register increments once every 12 or 4 system cycles when the timer clock is enabled.

In Counter Mode, the timer register of Timer 0 increments whenever a falling edge is detected on the corresponding input pin (T0 or PWM0). For Timer 1, the timer register increments on a falling edge detected on its corresponding input pin (T1).

10.1 Overview

Timer 0 and Timer 1 are fully compatible with the standard 8051 timers.

Each timer consists of two 8-bit registers: {TH0 (0x8C): TL0 (0x8A)} and {TH1 (0x8D): TL1 (0x8B)}. Both Timer 0 and Timer 1 operate in four identical modes. The modes for Timer 0 and Timer 1 are described as follows.

Mode	M1	M0	Functional description
0	0	0	THx[7:0] and TLx[4:0] form a 13-bit timer/counter.
1	0	1	THx[7:0] and TLx[7:0] together form a 16-bit timer/counter.
2	1	0	TLx[7:0] constitutes an 8-bit auto-reload timer/counter, which is reloaded from THx.
3	1	1	TL0 and TH0 are two 8-bit timer/counters, while Timer 1 stops counting.

The registers THx and TLx are special function registers that store the actual timer values. THx and TLx can be combined through mode options to form 13-bit or 16-bit registers. With each internal clock pulse or external timer pin state change, the value in the register increments by 1. The timer counts from the value loaded in the preload register until it overflows, at which point an internal interrupt signal is generated. In auto-reload mode, the timer value resets to the initial value from the preload register and continues counting; otherwise, the timer value resets to zero. Note that to achieve the maximum counting range for the timer/counter, the preload register must be cleared first.

10.2 Relevant register

10.2.1 Timer0/1 mode register (TMOD)

0x89	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
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
Reset value	0	0	0	0	0	0	0	0

Bit7 GATE1: Timer 1 gate control bit
1= Enable
0= Disable

Bit6 CT1: Timer 1 timer/counter selection bit
1= Count
0= Timer

Bit5~Bit4 T1M<1:0>: Timer 1 mode selection bit
00= Mode 0: 13-bit timer/counter
01= Mode 1: 16-bit timer/counter
10= Mode 2: 8-bit auto-reload timer/counter
11= Mode 3: Stop counting

Bit3 GATE0: Timer 0 gate control bit
1= Enable
0= Disable

Bit2 CT0: Timer 0 timer/counter selection bit
1= Count
0= Timer

Bit1~ Bit0 T0M<1:0>: Timer 0 mode selection bit
00= Mode 0: 13-bit timer/counter
01= Mode 1: 16-bit timer/counter
10= Mode 2: 8-bit auto-reload timer/counter
11= Mode 3: Two independent 8-bit timers/counters.

10.2.2 Timer0/1 control register (TCON)

0x88	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
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
Reset value	0	0	0	0	0	0	0	0

Bit7 TF1: Timer1 overflow interrupt flag
1= Timer1 overflow occurs, automatically cleared by hardware upon entering the interrupt service routine.
0= No Timer1 overflow

Bit6 TR1: Timer1 run control bit
1= Start Timer1
0= Stop Timer1

Bit5 TF0: Timer0 overflow interrupt flag
1= Timer0 overflow occurs, automatically cleared by hardware upon entering the interrupt service routine.
0= No Timer0 overflow

Bit4 TR0: Timer0 run control bit
1= Start Timer0
0= Stop Timer0

Bit3 IE1: External interrupt 1 flag
1= External interrupt 1 occurs, automatically cleared by hardware upon entering the interrupt service routine.
0= No interrupt for External Interrupt 1

Bit2 IT1: External interrupt 1 trigger control bit
1= Falling edge trigger
0= Low level trigger

Bit1 IE0: External interrupt 0 flag
1= External interrupt 0 occurs, automatically cleared by hardware upon entering the interrupt service routine.
0= No interrupt for External Interrupt 0

Bit0 IT0: External interrupt 0 trigger control bit
1= Falling edge trigger
0= Low level trigger

10.2.3 Timer0 low bit data register (TL0)

0x8A	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL0	TL07	TL06	TL05	TL04	TL03	TL02	TL01	TL00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~ Bit0 TL0<7:0>: Timer0 low bit data register (also used as counter low bit).

10.2.4 Timer0 high bit data register (TH0)

0x8C	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH0	TH07	TH06	TH05	TH04	TH03	TH02	TH01	TH00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TH0<7:0>: Timer0 high bit data register (also used as counter high bit).

10.2.5 Timer1 low bit data register (TL1)

0x8B	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL1	TL17	TL16	TL15	TL14	TL13	TL12	TL11	TL10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TL1<7:0>: Timer1 low bit data register (also used as counter low bit).

10.2.6 Timer1 high bit data register (TH1)

0x8D	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH1	TH17	TH16	TH15	TH14	TH13	TH12	TH11	TH10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TH1<7:0>: Timer1 high bit data register (also used as counter high bit).

10.2.7 Function clock control register (CKCON)

0x8E	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CKCON	WTS2	WTS1	WTS0	T1M	T0M	--	--	T0CNTM
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	1	1	1

Bit7~Bit5 WTS<2:0>: WDT overflow time select bit
000= 2^{17}*Tsys
001= 2^{18}*Tsys
010= 2^{19}*Tsys
011= 2^{20}*Tsys
100= 2^{21}*Tsys
101= 2^{22}*Tsys
110= 2^{24}*Tsys
111= 2^{26}*Tsys

Bit4 T1M: Timer1 clock source selection bit
0= Fsys/12
1= Fsys/4

Bit3 T0M: Timer0 clock source selection bit
0= Fsys/12
1= Fsys/4.

Bit2~Bit1 --: Reserved, set to 1.

Bit0 T0CNTM: Timer0 count source select bit
0= PWM0 output
1= T0 pin input

10.3 Timer 0/1 interrupt

Timer 0/1 can be enabled or disabled for interrupts through the IE register, and the priority (high/low) can be set using the IP register. The relevant interrupt bits are as follows:

10.3.1 Interrupt mask register (IE)

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	EA:	Global interrupt enable bit
	1=	Enable all unmasked interrupts
	0=	Disable all interrupts
Bit6	ES1:	UART1 interrupt enable bit
	1=	Enable UART1 interrupt
	0=	Disable UART1 interrupt
Bit5	ET2:	TIMER2 global interrupt enable bit
	1=	Enable all TIMER2 interrupts
	0=	Disable all TIMER2 interrupts
Bit4	ES0:	UART0 interrupt enable bit
	1=	Enable UART0 interrupt
	0=	Disable UART0 interrupt
Bit3	ET1:	TIMER1 interrupt enable bit
	1=	Enable TIMER1 interrupt
	0=	Disable TIMER1 interrupt
Bit2	EX1:	External interrupt 1 enable bit
	1=	Enable external interrupt 1
	0=	Disable external interrupt 1
Bit1	ET0:	TIMER0 interrupt enable bit
	1=	Enable TIMER0 interrupt
	0=	Disable TIMER0 interrupt
Bit0	EX0:	External interrupt 0 enable bit
	1=	Enable external interrupt 0
	0=	Disable external interrupt 0

10.3.2 Interrupt priority control register (IP)

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	PS1	PT2	PS0	PT1	PX1	PT0	PX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 -- Reserved, set to 0.
Bit6 PS1: UART1 interrupt priority control bit
1= High priority level
0= Low priority level
Bit5 PT2: TIMER2 interrupt priority control bit
1= High priority level
0= Low priority level
Bit4 PS0: UART0 interrupt priority control bit
1= High priority level
0= Low priority level
Bit3 PT1: TIMER1 interrupt priority control bit
1= High priority level
0= Low priority level
Bit2 PX1: External interrupt 1 interrupt priority control bit
1= High priority level
0= Low priority level
Bit1 PT0: TIMER0 interrupt priority control bit
1= High priority level
0= Low priority level
Bit0 PX0: External interrupt 0 interrupt priority control bit
1= High priority level
0= Low priority level

10.3.3 Timer0/1, INT0/1 interrupt flag register (TCON)

0x88	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
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
Reset value	0	0	0	0	0	0	0	0

Bit7 TF1: Timer1 overflow interrupt flag
1= Timer1 overflow occurs; automatically cleared by hardware upon entering the interrupt service routine, can also be cleared by software.
0= No overflow in Timer1

Bit6 TR1: Timer1 run control bit
1= Start Timer1
0= Stop Timer1

Bit5 TF0: Timer0 overflow interrupt flag
1= Timer0 overflow occurs; automatically cleared by hardware upon entering the interrupt service routine, can also be cleared by software.
0= No overflow in Timer0

Bit4 TR0: Timer0 run control bit
1= Start Timer0
0= Stop Timer0

Bit3 IE1: External interrupt 1 flag
1= External Interrupt 1 occurs; automatically cleared by hardware upon entering the interrupt service routine, can also be cleared by software.
0= No interrupt from External Interrupt 1

Bit2 IT1: External interrupt 1 trigger mode control bit
1= Trigger on falling edge
0= Trigger on low level

Bit1 IE0: External interrupt 0 flag
1= External Interrupt 0 occurs; automatically cleared by hardware upon entering the interrupt service routine, can also be cleared by software.
0= No interrupt from External Interrupt 0

Bit0 IT0: External interrupt 0 trigger mode control bit
1= Trigger on falling edge
0= Trigger on low level

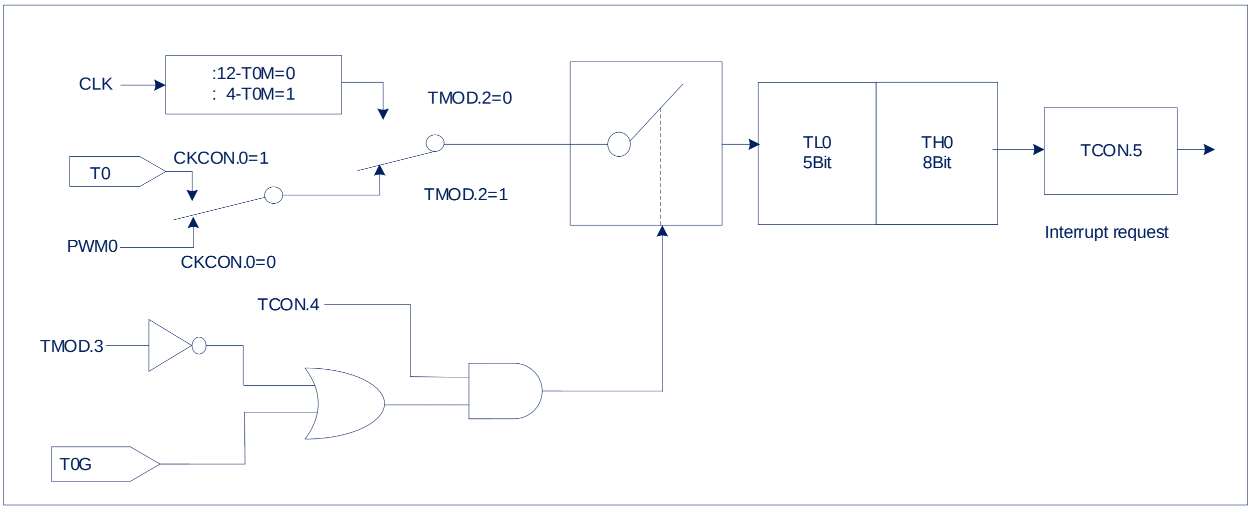
The interrupt flags can be cleared by software, which results in the same effect as clearing them by hardware. This means that it is possible to generate an interrupt through software (though it is not recommended to set the flag directly to generate an interrupt) or to cancel a pending interrupt.

The TF0 and TF1 flags can be cleared by writing a 0 to them even when interrupts are not enabled.

10.4 Timer0 operation modes

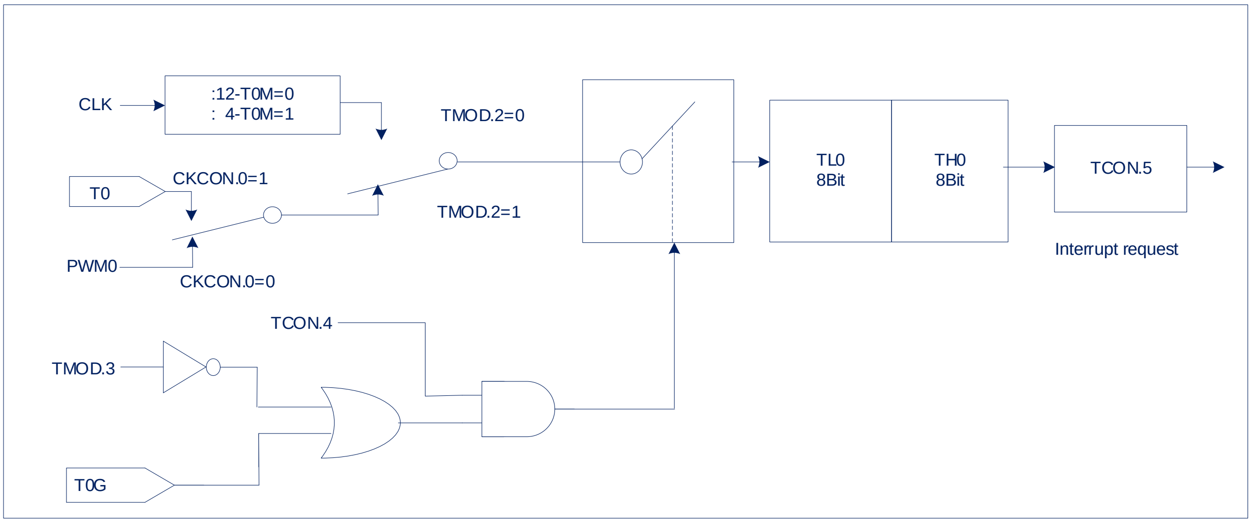
10.4.1 T0 -Mode 0 (13-bit timing/counting mode)

In this mode, Timer 0 is a 13-bit register. When all bits of the counter flip from 1 to 0, the Timer 0 interrupt flag (TF0) is set to 1. The counting input is enabled to Timer 0 when TCON.4 = 1 and either TMOD.3 = 0 or TCON.4 = 1 and TMOD.3 = 1, with T0G = 1. (Setting TMOD.3 = 1 allows Timer 0 to be controlled by the external pin T0G for pulse width measurement). The 13-bit register consists of TH0 and the lower 5 bits of TL0, while the upper 3 bits of TL0 should be ignored. The block diagram of Timer 0 in mode 0 is shown below:



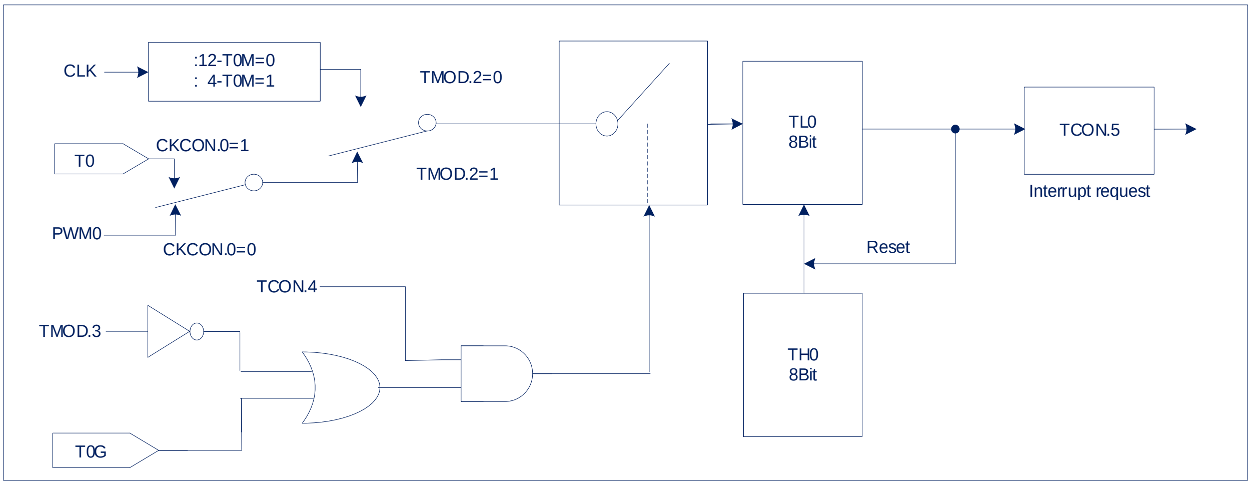
10.4.2 T0 -Mode 1 (16-bit timing/counting mode)

Mode 1 is similar to Mode 0, but in Mode 1, the Timer 0 data register operates as a full 16-bit register. The block diagram of Timer 0 in Mode 1 is shown below:



10.4.3 T0 -Mode 2 (8-bit auto-reload timing/counting mode)

In Mode 2, the timer register functions as an 8-bit counter (TL0) with an auto-reload feature, as shown in the diagram below. An overflow from TL0 not only sets the TF0 flag to 1 but also reloads the content of TH0 into TL0 via software. During the reload process, the value of TH0 remains unchanged. The block diagram of Timer 0 in Mode 2 is shown below:



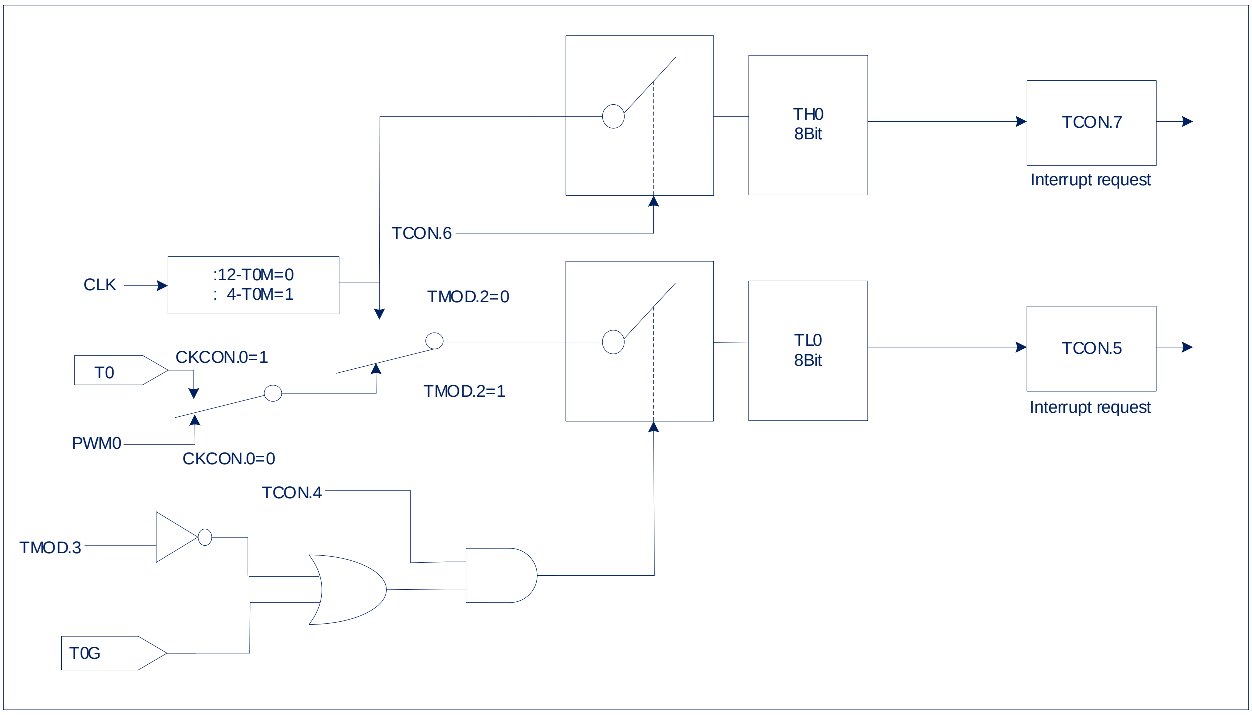
10.4.4 T0 -Mode 3 (two separate 8-bit timer/counters)

In Mode 3, Timer 0 configures TL0 and TH0 as two seperate counters. The logic for Timer 0 in Mode 3 is shown in the diagram below.

TL0 can operate as a timer or counter and use the control bits of Timer0: CT0, TR0, GATE0 and TF0.

TH0 can only work as a timer, and uses the TR1 and TF1 flags of Timer1 and controls the interrupt of Timer1.

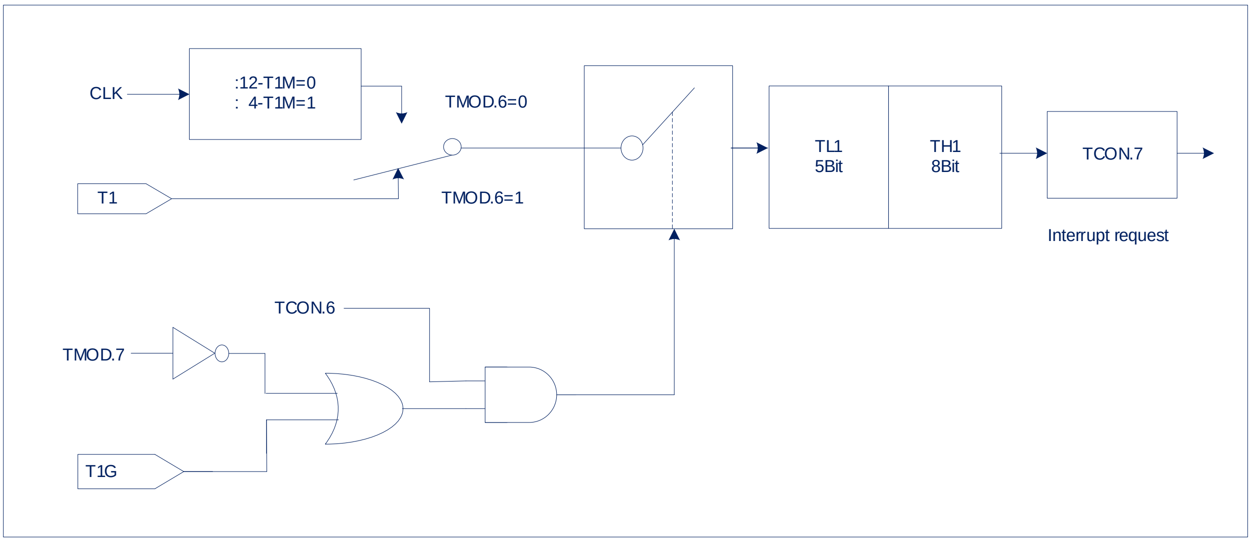
Mode 3 can be used when two 8-bit timer/counters are required. When Timer0 is in mode 3, Timer1 can be turned off by switching to its own mode 3, or can still be used by the serial channel as a baud rate generator, or in any application that does not require Timer1 interrupts. The Timer0 mode 3 architecture block diagram is shown below.



10.5 Timer1 operation modes

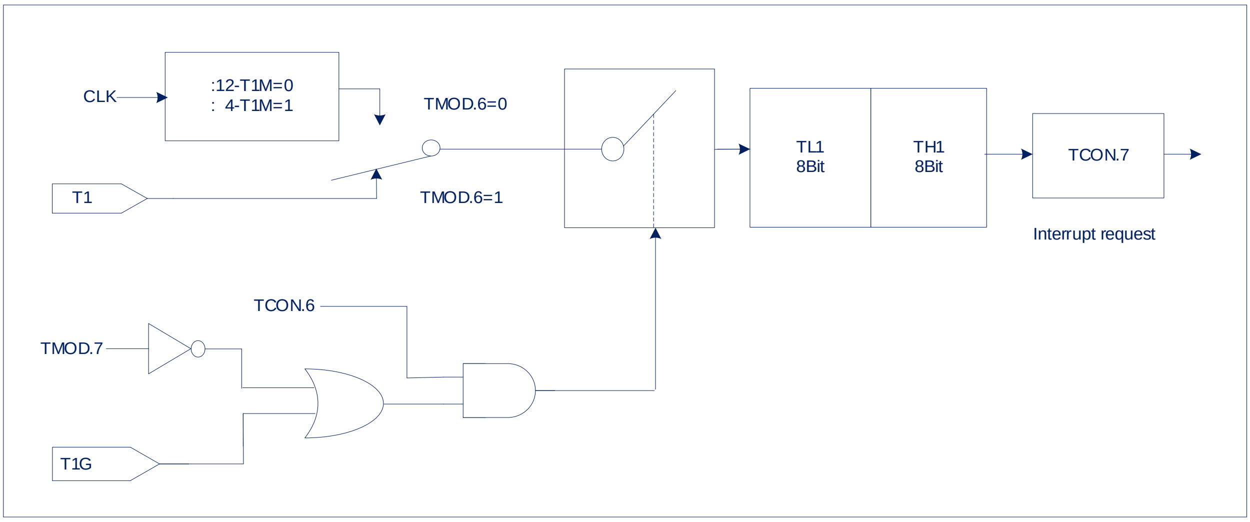
10.5.1 T1 -Mode 0 (13-bit timing/counting mode)

In this mode, Timer 1 is a 13-bit register. When all bits of the counter transition from 1 to 0, the Timer 1 interrupt flag (TF1) is set to 1. The counting input is enabled for Timer 1 when TCON.6 = 1 and TMOD.7 = 0, or when TCON.6 = 1, TMOD.7 = 1, and T1G = 1 (Setting TMOD.7 = 1 allows Timer 1 to be controlled by the external pin T1G for pulse width measurement). The 13-bit register consists of the 8 bits from TH1 and the lower 5 bits from TL1. The upper three bits of TL1 should be ignored. The block diagram of Timer 1 in Mode 0 is shown below:



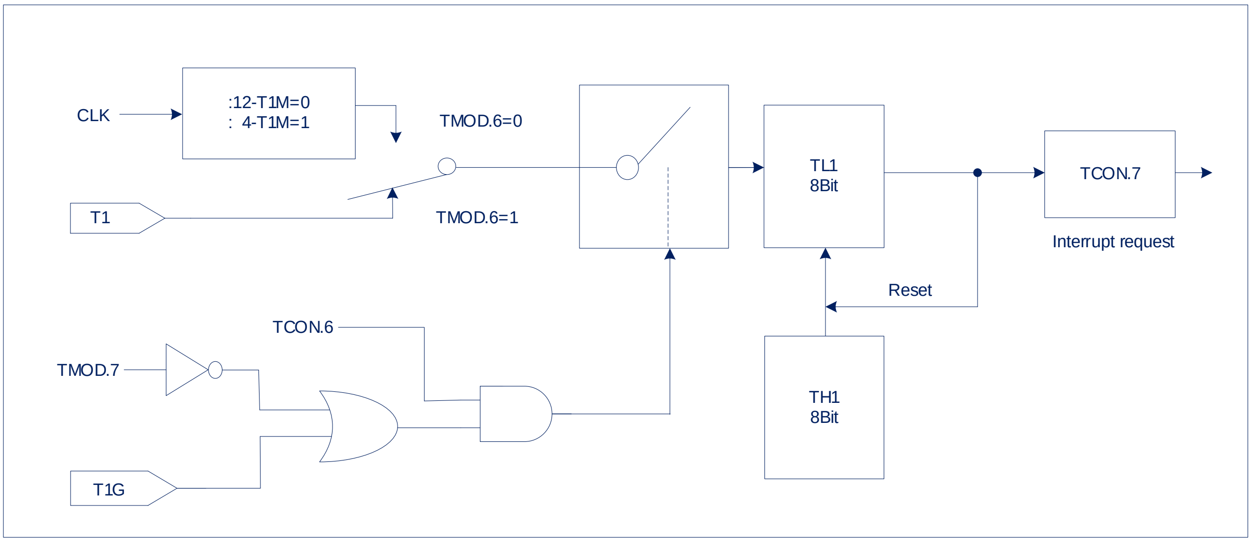
10.5.2 T1 -Mode 1 (16-bit timing/counting mode)

Mode 1 is similar to Mode 0, but in Mode 1, the Timer 1 register operates as a full 16-bit register. The block diagram of Timer 1 in Mode 1 is shown below:



10.5.3 T1 -Mode 2 (8-bit auto-reload timing/counting mode)

In Mode 2, Timer 1 operates as an 8-bit counter (TL1) with an automatic reload feature. When TL1 overflows, it not only sets the Timer 1 interrupt flag (TF1) but also reloads the content of TH1 into TL1. During this reload process, the value in TH1 remains unchanged. The block diagram of Timer 1 in Mode 2 is shown below:



10.5.4 T1 -Mode 3 (stop counting)

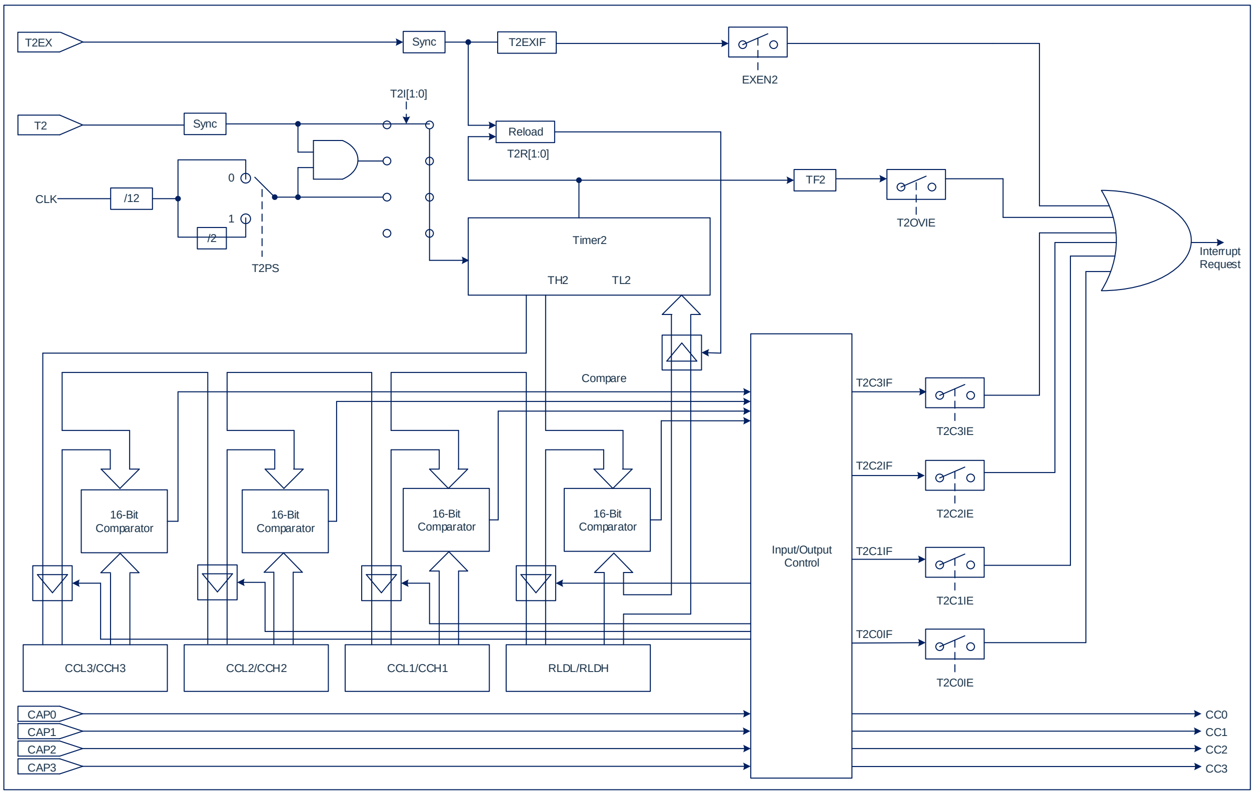
In Mode 3, Timer 1 stops counting, which has the same effect as setting TR1 to 0.

11. Timer 2

Timer 2, with its additional compare/capture/reload functionalities, is one of the core peripheral units. It can be used for various digital signal generation and event capture tasks, such as pulse generation, pulse width modulation (PWM), and pulse width measurement.

11.1 Overview

The block diagram of Timer 2, which includes additional compare/capture/reload register functionalities, is shown below:



11.2 Relevant registers

11.2.1 Timer2 control register (T2CON)

0xC8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CON	T2PS	I3FR	CAPES	T2R1	T2R0	T2CM	T2I1	T2I0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 T2PS: Timer2 clock prescaler selection bit
1= Fsys/24
0= Fsys/12

Bit6 I3FR: Capture channel 0 input single-edge select and compare interrupt timing select bit
Capture channel 0 mode
1= Capture rising edge to RLDL/RLDH register
0= Capture falling edge to RLDL/RLDH register
Compare channel 0 mode
1= An interrupt is generated when TL2/TH2 and RLDL/RLDH transition from not equal to equal.
0= An interrupt is generated when TL2/TH2 and RLDL/RLDH transition from equal to not equal.

Bit5 CAPES: Capture channels 1-3 input single-edge selection (affects all capture channels 1-3)
0= Capture rising edge to CCL1/CCH1-CCL3/CCH3 register
1= Capture falling edge to CCL1/CCH1-CCL3/CCH3 register
Capture mode 2 starting edge selection bit
0= Rising edge capture
1= Falling edge capture

Bit4~Bit3 T2R<1:0>: Timer2 load mode selection bit
0x= Reload disabled
10= Load mode 1: automatic reload on timer 2 overflow
11= Load mode 2: reload on falling edge ofT2EX pin

Bit2 T2CM: Compare mode selection
1= Compare mode 1
0= Compare mode 0

Bit1~Bit0 T2I<1:0>: Timer 2 clock input selection bit
00= Timer2 stopped
01= System clock prescaler (controlled byT2PS)
10= External pin T2 is used for event input (event counting mode)
11= External pin T2 is used for gate input (gate timing mode)

11.2.2 Timer2 low bit data register (TL2)

0xCC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL2	TL27	TL26	TL25	TL24	TL23	TL22	TL21	TL20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TL2<7:0>: Timer 2 low data register (also used as counter low bit).

11.2.3 Timer2 high bit data register (TH2)

0xCD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH2	TH27	TH26	TH25	TH24	TH23	TH22	TH21	TH20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TH2<7:0>: Timer 2 high data register (also used as counter low bit).

11.2.4 Timer2 compare/capture/auto reload low bit register (RLDL)

0xCA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RLDL	RLDL7	RLDL6	RLDL5	RLDL4	RLDL3	RLDL2	RLDL1	RLDL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 RLDL<7:0>: Timer 2 compare/capture/auto-reload register low bit.

11.2.5 Timer2 compare/capture/auto reload high bit register (RLDH)

0xCB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RLDH	RLDH7	RLDH6	RLDH5	RLDH4	RLDH3	RLDH2	RLDH1	RLDH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 RLDH<7:0>: Timer 2 compare/capture/auto-reload register high bit.

11.2.6 Timer2 compare/capture channel 1 low bit register (CCL1)

0xC2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCL1	CCL17	CCL16	CCL15	CCL14	CCL13	CCL12	CCL11	CCL10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 CCL1<7:0>: Timer 2 compare/capture channel 1 register low bit.

11.2.7 Timer2 compare/capture channel 1 high bit register (CCH1)

0xC3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCH1	CCH17	CCH16	CCH15	CCH14	CCH13	CCH12	CCH11	CCH10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 CCH1<7:0>: Timer 2 compare/capture channel 1 register high bit.

11.2.8 Timer2 compare/capture channel 2 low bit register (CCL2)

0xC4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCL2	CCL27	CCL26	CCL25	CCL24	CCL23	CCL22	CCL21	CCL20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 CCL2<7:0>: Timer 2 compare/capture channel 2 register low bit.

11.2.9 Timer2 compare/capture channel 2 high bit register (CCH2)

0xC5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCH2	CCH27	CCH26	CCH25	CCH24	CCH23	CCH22	CCH21	CCH20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 CCH2<7:0>: Timer 2 compare/capture channel 2 register high bit.

11.2.10 Timer2 compare/capture channel 3 low bit register (CCL3)

0xC6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCL3	CCL37	CCL36	CCL35	CCL34	CCL33	CCL32	CCL31	CCL30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 CCL3<7:0>: Timer 2 compare/capture channel 3 register low bit.

11.2.11 Timer2 compare/capture channel 3 high bit register (CCH3)

0xC7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCH3	CCH37	CCH36	CCH35	CCH34	CCH33	CCH32	CCH31	CCH30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 CCH3<7:0>: Timer 2 compare/capture channel 3 register high bit.

11.2.12 Timer2 compare/capture control register (CCEN)

0xCE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CCEN	CMH3	CML3	CMH2	CML2	CMH1	CML1	CMH0	CML0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6	CMH3-CML3:	Capture/compare mode control bit
	00=	Capture/compare disable
	01=	Capture operation triggered on the rising or falling edge of Channel 3 (CAPES selection)
	10=	Compare mode enable
	11=	Capture operation triggered when writing to CCL3 or on dual-edge triggering of Channel 3
Bit5~Bit4	CMH2-CML2:	Capture/compare mode control bit
	00=	Capture/compare disable
	01=	Capture operation triggered on the rising or falling edge of Channel 2 (CAPES selection)
	10=	Compare mode enable
	11=	Capture operation triggered when writing to CCL2 or on dual-edge triggering of Channel 2
Bit3~Bit2	CMH1-CML1:	Capture/compare mode control bit
	00=	Capture/compare disable
	01=	Capture operation triggered on the rising or falling edge of Channel 1 (CAPES selection)
	10=	Compare mode enable
	11=	Capture operation triggered when writing to CCL1 or on dual-edge triggering of Channel 1.
Bit1~Bit0	CMH0-CML0:	Capture/compare mode control bit
	00=	Capture/compare disable
	01=	Capture operation triggered on the rising or falling edge of Channel 0 (I3FR selection)
	10=	Compare mode enable
	11=	Capture operation triggered when writing to RLDL or on dual-edge triggering of Channel 0.

11.2.13 Timer2 capture mode 2 control register (CAP2CON)

0xC1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CAP2CON	CAP2ERR	--	--	--	CAP2LOCK	CAP2CLR	CAP2ST	CAP2EN
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 CAP2ERR: Timer2 overflow indication bit during the capture process (hardware sets to 1 and clears to 0; read-only bit);

1= In capture mode 2, counter overflows during the capture process.

0= When Timer2 CAP2EN/CAP2ST bit is 0, or when there is no overflow during the capture process.

Bit6~Bit4 -- Reserved, set to 0.

Bit3 CAP2LOCK: Timer2 capture mode 2 read latch function

1= Enable

0= Disable

Bit2 CAP2CLR: Timer2 capture completion clear timer enable control

1= Enable

0= Disable

Bit1 CAP2ST: Timer2 capture mode 2 start bit

1= Enable capture mode

0= Disable capture mode

Bit0 CAP2EN: Timer2 capture mode 2 enable control

1= Enable

0= Disable

11.3 Timer2 interrupts

Timer 2 can enable or disable the global interrupt through the IE register, and the priority can be set to high or low using the IP register. Timer 2 supports four types of interrupts:

- Timer overflow interrupt
- External pin T2EX falling edge interrupt
- Compare interrupt
- Capture interrupt

To configure Timer 2 interrupts, the following bits must be set: Global Interrupt Enable Bit (EA = 1), Timer 2 Global Interrupt Enable Bit (ET2 = 1), Timer 2 Specific Interrupt Enable Bit (T2IE). All four types of Timer 2 interrupts share a single interrupt vector. Upon entering the interrupt service routine, it is necessary to check the relevant flag bits to determine which type of interrupt has occurred.

11.3.1 Relevant registers

11.3.1.1 Interrupt mask register (IE)

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 EA: Global interrupt enable bit
1= Enable all unmasked interrupts
0= Disable all interrupts
Bit6 ES1: UART1 interrupt enable bit
1= Enable UART1 interrupts
0= Disable UART1 interrupts
Bit5 ET2: TIMER2 global interrupt enable bit
1= Enable all TIMER2 interrupts
0= Disable all TIMER2 interrupts
Bit4 ES0: UART0 interrupt enable bit
1= Enable UART0 interrupt
0= Disable UART0 interrupt
Bit3 ET1: TIMER1 interrupt enable bit
1= Enable TIMER1 interrupt
0= Disable TIMER1 interrupt
Bit2 EX1: External interrupt 1 interrupt enable bit
1= Enable external interrupt 1 interrupt
0= Disable external interrupt 1 interrupt
Bit1 ET0: TIMER0 interrupt enable bit
1= Enable TIMER0 interrupt
0= Disable TIMER0 interrupt
Bit0 EX0: External interrupt 0 interrupt enable bit
1= Enable external interrupt 0 interrupt
0= Disable external interrupt 0 interrupt

11.3.1.2 Timer2 interrupt mask register (T2IE)

0xCF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IE	T2OVIE	T2EXIE	--	--	T2C3IE	T2C2IE	T2C1IE	T2C0IE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	T2OVIE:	Timer2 overflow interrupt enable bit
	1=	Enable interrupt
	0=	Disable interrupt
Bit6	T2EXIE:	Timer2 external load interrupt enable bit
	1=	Enable interrupt
	0=	Disable interrupt
Bit5~Bit4	--	Reserved, set to 0.
Bit3	T2C3IE:	Timer2 compare/capture channel 3 interrupt enable bit
	1=	Enable interrupt
	0=	Disable interrupt
Bit2	T2C2IE:	Timer2 compare/capture channel 2 interrupt enable bit
	1=	Enable interrupt
	0=	Disable interrupt
Bit1	T2C1IE:	Timer2 compare/capture channel 1 interrupt enable bit
	1=	Enable interrupt
	0=	Disable interrupt
Bit0	T2C0IE:	Timer2 compare/capture channel 0 interrupt enable bit
	1=	Enable interrupt
	0=	Disable interrupt

If you enable the Timer 2 interrupts, you also need to set the Timer 2 global interrupt enable bit (ET2) to 1 (IE.5 = 1).

11.3.1.3 Interrupt priority control register (IP)

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	PS1	PT2	PS0	PT1	PX1	PT0	PX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 -- Reserved, set to 0.

Bit6 PS1: UART1 interrupt priority control bit
1= High priority level
0= Low priority level

Bit5 PT2: TIMER2 interrupt priority control bit
1= High priority level
0= Low priority level

Bit4 PS0: UART0 interrupt priority control bit
1= High priority level
0= Low priority level

Bit3 PT1: TIMER1 interrupt priority control bit
1= High priority level
0= Low priority level

Bit2 PX1: External interrupt 1 interrupt priority control bit
1= High priority level
0= Low priority level

Bit1 PT0: TIMER0 interrupt priority control bit
1= High priority level
0= Low priority level

Bit0 PX0: External interrupt 0 interrupt priority control bit
1= High priority level
0= Low priority level

11.3.1.4 Timer2 interrupt flag bit register (T2IF)

0xC9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2IF	TF2	T2EXIF	--	--	T2C3IF	T2C2IF	T2C1IF	T2C0IF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 TF2: Timer2 overflow interrupt flag
1= Timer2 counter overflow requires software reset
0= Timer2 counter no overflow

Bit6 T2EXIF: Timer2 external load flag
1= Timer2's T2EX pin generates a falling edge, requiring software reset.
0= --

Bit5~Bit4 -- Reserved, set to 0.

Bit3 T2C3IF: Timer2 ompare/capture channel 3 flag
1= Timer2 compare channel 3 {CCH3:CCL3}={TH2:TL2} or capture channel 3 has generated a capture operation, requiring software reset.
0= --

Bit2 T2C2IF: Timer2 compare/capture channel 2 flag
1= Timer2 compare channel 2 {CCH2:CCL2}={TH2:TL2} or capture channel 2 has generated a capture operation, requiring software reset.
0= --

Bit1 T2C1IF: Timer2 compare/capture channel 1 flag
1= Timer2 compare channel 1 {CCH1:CCL1}={TH2:TL2} or capture channel 1 has generated a capture operation, requiring software reset.
0= --

Bit0 T2C0IF: Timer2 compare/capture channel 0 flag
1= Timer2 compare channel 0 {RLDH:RLDL}={TH2:TL2} or capture channel 0 has generated a capture operation, requiring software reset.
0= --

11.3.2 Timer interrupt

The timer interrupt enable bit is set by register T2IE[7], and the interrupt flag can be checked via register T2IF[7]. When the Timer2 overflows, the timer overflow interrupt flag TF2 will be set to 1.

11.3.3 External trigger interrupt

The external pin T2EX falling edge trigger interrupt enable bit is set by register T2IE[6], and the interrupt flag can be checked via register T2IF[6]. When the T2EX pin experiences a falling edge, the external load interrupt flag T2EXIF will be set to 1.

11.3.4 Comparison interrupt

All four comparison channels support comparison interrupts. The comparison interrupt enable bits are set by register T2IE[3:0], and the interrupt flags can be checked via register T2IF[3:0].

For comparison channel 0, the timing of when the comparison interrupt occurs can be selected, and if an interrupt occurs, the comparison channel 0 interrupt flag T2C0IF will be set to 1.

When I3FR = 0, an interrupt is generated when TL2/TH2 transitions from unequal to equal with RLDL/RLDH.

When I3FR = 1, an interrupt is generated when TL2/TH2 transitions from equal to unequal with RLDL/RLDH.

For comparison channels 1 to 3, the timing for the interrupt cannot be selected and is fixed to when TL2/TH2 transitions from unequal to equal with CCxL/CCxH. If an interrupt occurs, the corresponding comparison channel interrupt flag T2CxIF will be set to 1.

11.3.5 Capture interrupt

All four capture channels support external capture interrupts. The capture interrupt enable bits are set by register T2IE[3:0], and the interrupt flags can be checked via register T2IF[3:0]. When a capture operation occurs, the corresponding capture channel interrupt flag T2CxIF will be set to 1.

Note that write operation capture mode does not generate an interrupt.

11.4 Timer2 function description

Timer2 is a 16-bit up-counting timer with its clock source derived from the system clock. Timer2 can be configured in the following operational modes:

- Timer mode.
- Reload mode.
- Gated timer mode.
- Event counting mode.
- Compare mode.
- Capture mode.

By setting Timer2 to different modes, it can be used for various digital signal generation and event capture tasks, such as pulse generation, pulse width modulation, and pulse width measurement.

11.4.1 Timer mode

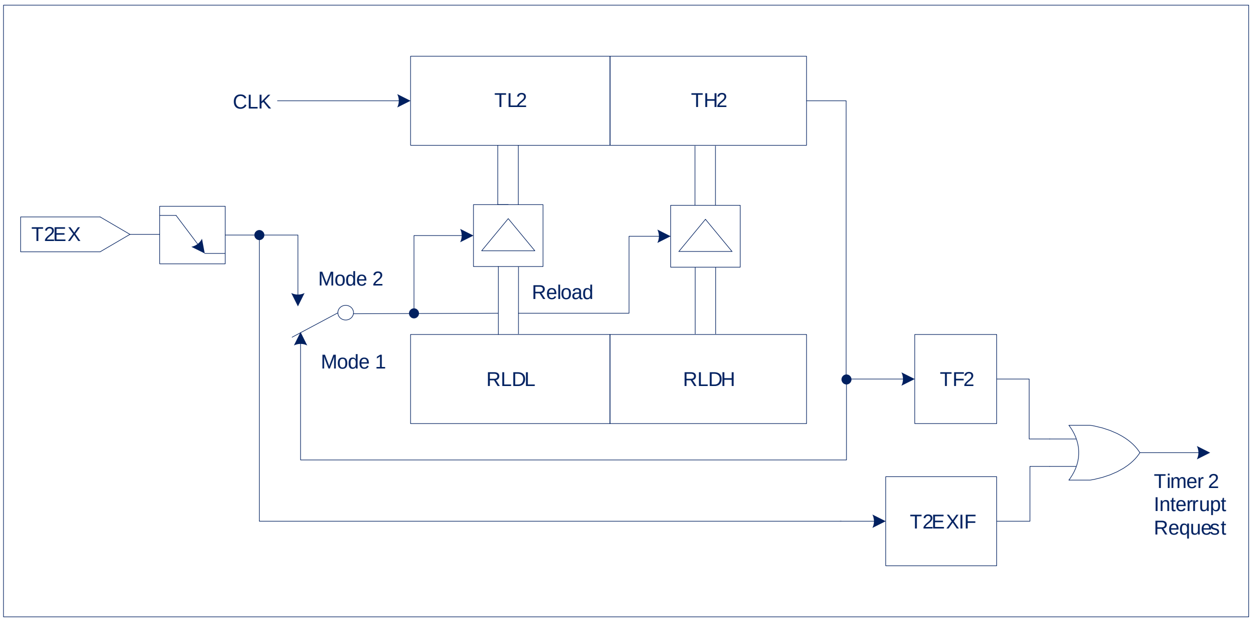
When used as a timer, the clock source comes from the system clock. The prescaler provides options for dividing the system frequency by 1/12 or 1/24, with the prescaler value selected by the T2PS bit in the T2CON register. Therefore, the 16-bit timer register (composed of TH2 and TL2) increments every 12 or 24 clock cycles.

11.4.2 Reload mode

The reload mode of Timer2 is chosen by the T2R0 and T2R1 bits in the T2CON register. The structure diagram for the reload operation is shown below.

In Load Mode 1: When the Timer2 counter transitions from all 1s to 0 (counter overflow), not only is the overflow interrupt flag TF2 set to 1, but the Timer2 register also automatically reloads a 16-bit value from the RLDL/RLDH registers, thereby overwriting the count value of 0x0000. The required RLDL/RLDH values can be preset by software.

In Load Mode 2: The 16-bit reload operation from the RLDL/RLDH registers is triggered by the falling edge of the corresponding T2EX input pin. When a falling edge on T2EX is detected, the external load interrupt flag T2EXIF is set to 1, and Timer2 automatically loads the 16-bit value from the RLDL/RLDH registers as the initial count value.



11.4.3 Gated timer mode

When Timer2 is used as a gated timer, the external input pin T2 serves as the gate input for Timer2. If the T2 pin is high, the internal clock input is allowed to pass through to the timer. If the T2 pin is low, counting is halted. This functionality is commonly used to measure pulse width.

11.4.4 Event counting mode

When Timer2 functions as an event counter, the timer counts up by 1 on each falling edge of the external input pin T2. The external input signal is sampled on every system clock cycle; when the sampled input shows high in one cycle and then transitions to low in the next, the count increases. If a high-to-low transition is detected on the T2 pin in the following cycles, the new count value is updated in the timer data register.

11.4.5 Compare mode

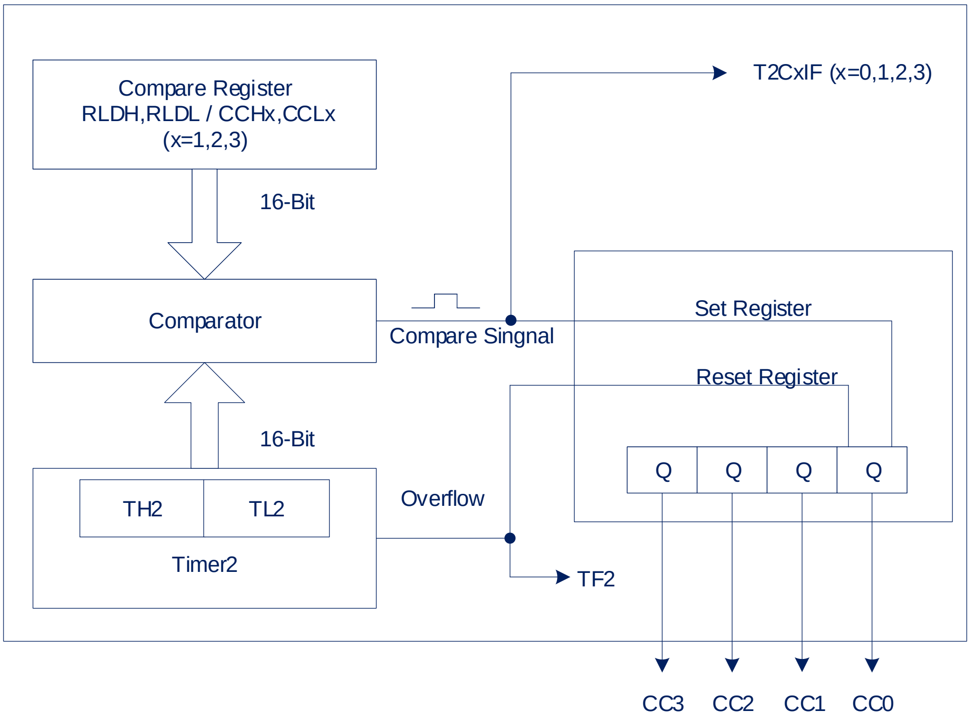
The compare function includes two modes: Compare Mode 0 and Compare Mode 1, selected by the T2CM bit in the special function register T2CON. These two comparison modes can generate periodic signals and alter the duty cycle control method, commonly used in pulse width modulation (PWM) and applications requiring continuous square wave generation.

The output channels for the comparison function are CC0, CC1, CC2, and CC3, corresponding to the 16-bit comparison registers {RLDH, RLDL}, {CCH1, CCL1}, {CCH2, CCL2}, and {CCH3, CCL3} along with the timer data registers {TH2, TL2}.

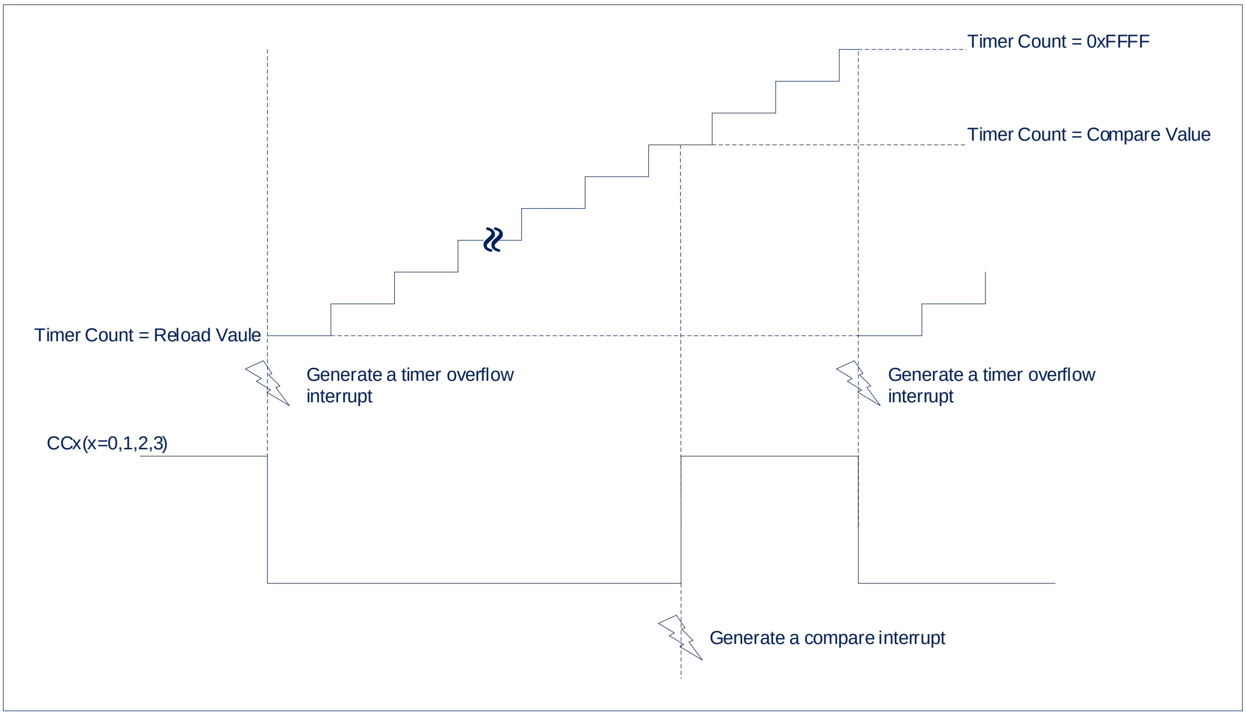
The 16-bit stored values in the comparison registers are compared with the timer’s count value; if the count value in the data register matches the stored value, an output signal transition occurs on the corresponding port pin, and an interrupt flag is generated.

11.4.5.1 Compare mode 0

In Mode 0, when the timer’s count value equals the value in the compare register, the comparison output signal transitions from low to high. When the timer count overflows, the comparison output signal returns to low. The comparison output channel is directly controlled by two events: timer overflow and comparison operation. The structural block diagram of Compare Mode 0 is shown below:



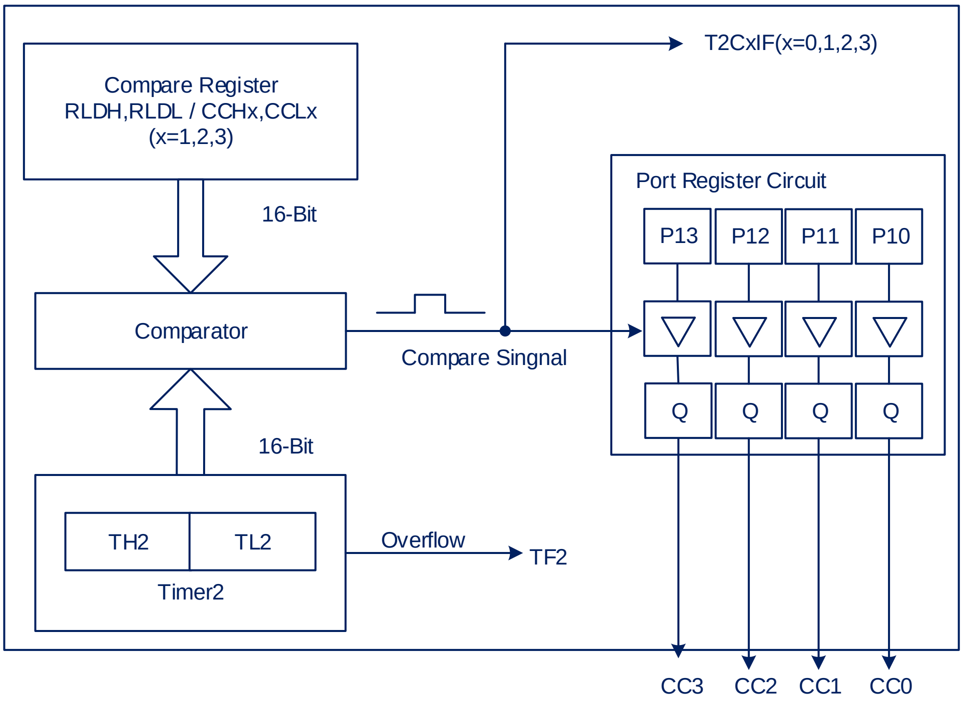
The output block diagram for Compare Mode 0 is shown below:


Timer Count = 0xFFFF
Timer Count = Compare Value
Timer Count = Reload Vaule
Generate a timer overflow interrupt
Generate a timer overflow interrupt
CCx(x=0,1,2,3)
Generate a compare interrupt

11.4.5.2 Compare mode 1

In Compare Mode 1, the output signal changes are determined adaptively by software and are independent of the periodicity of the constant signal.

When Mode 1 is enabled, the software writes to the corresponding output register for the CC3 port. The new value will not appear on the output pin until the next comparison match occurs. When the Timer2 counter matches the stored comparison value, the user can choose whether to change the output signal to the new value or maintain its previous value. The block diagram for Comparison Mode 1 is shown below:



11.4.6 Capture mode

The capture functionality includes four sets of 16-bit registers: {RLDH, RLDL}, {CCH1, CCL1}, {CCH2, CCL2}, and {CCH3, CCL3}. Each pair of registers can latch the current 16-bit value of {TH2, TL2}. This feature provides three different capture modes.

In Mode 0, an external event can latch the contents of Timer 2 into the capture registers.

In Mode 1, capture occurs when the low byte of the capture register (RLDL/CCL1/CCL2/CCL3) is written. This mode allows software to read the contents of {TH2, TL2} at runtime.

The capture channels 0 to 3 select the capture input pins CAP0 to CAP3 as the input signal sources.

In Mode 2, capture input pin CAP1 is used as the input signal source. Mode 2 is designed for continuous capturing of the signal from input channel CAP1, with three capture values sequentially loaded into CCL1/CCH1, CCL2/CCH2, and CCL3/CCH3 registers.

When capture mode 2 is enabled, channels 1, 2, and 3 are forced to operate in capture mode 2, while channel 0 remains unaffected and can perform any function.

11.4.6.1 Capture mode 0

In Capture Mode 0, a rising edge, falling edge, or both edges on the capture channels (CAP0 to CAP3) will generate a capture event. When a capture event occurs, the count value of the timer is latched into the corresponding capture register.

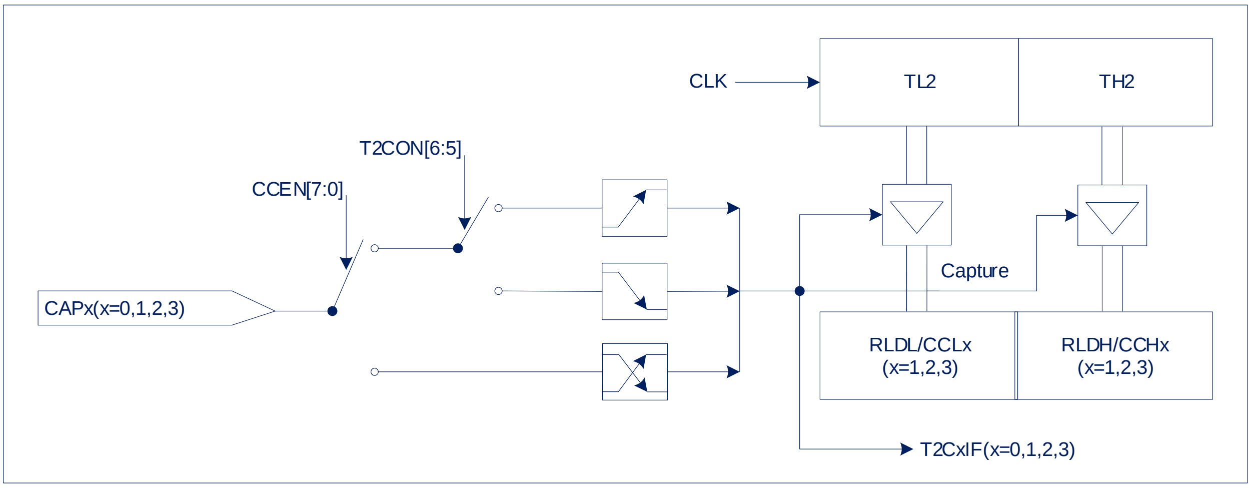
For Capture Channel 0, whether the capture operation is triggered by a rising or falling edge depends on the I3FR bit of the T2CON register. I3FR = 0: Falling edge triggers capture. I3FR = 1: Rising edge triggers capture.

For Capture Channels 1 to 3, whether the capture operation is triggered by a rising or falling edge depends on the CAPES bit of the T2CON register: CAPES = 0: Rising edge triggers capture. CAPES = 1: Falling edge triggers capture. The edge selection for capture channels 1 to 3 is the same.

Capture channels 0 to 3 also support dual-edge capture operations. If the control bits in the CCEN register are set to 11, the channel supports dual-edge capture operations. It is important to note that in this working mode, it also supports Mode 1, where a write operation can trigger a capture action.

In Capture Mode 0, external capture events on channels 0 to 3 can all generate interrupts.

A structural block diagram for Capture Mode 0 is shown as below.

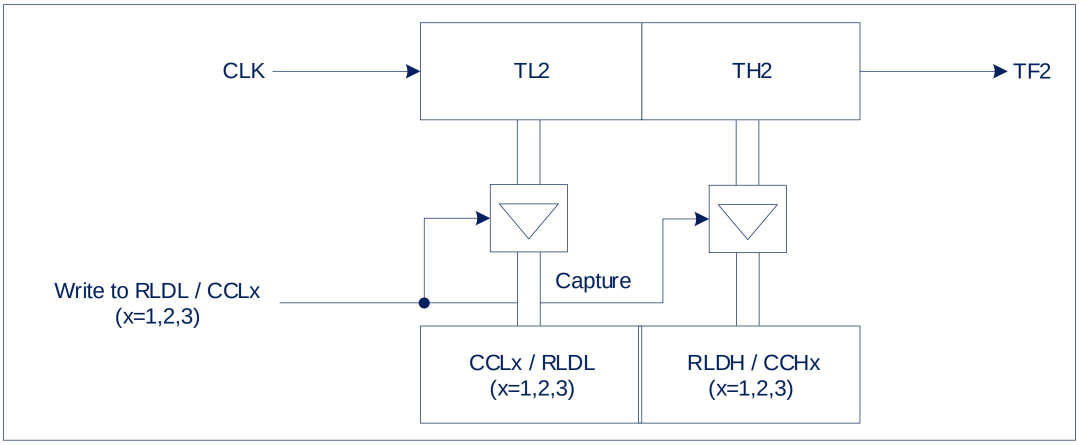


11.4.6.2 Capture mode 1

In Capture Mode 1, the capture event is triggered by the execution of a write instruction to the low byte of the capture register. The write signal (for example, writing to RLDL) initiates the capture operation, and the value being written is irrelevant to this function. After the write instruction is executed, the contents of Timer 2 will be latched into the corresponding capture register.

In Capture Mode 1, capture events for channels 0 to 3 do not generate interrupt request flags.

The structural block diagram for Capture Mode 1 is as follows:



11.4.6.3 Capture mode 2

In Capture Mode 2, the input signal on capture channel CAP1 is continuously captured for three edge events. The completion of these three captures constitutes a complete capture event, with the capture values being loaded into registers CCL1/CCH1, CCL2/CCH2, and CCL3/CCH3 in sequence.

The edge type for capturing is determined by the T2CON[5] bit. T2CON[5] = 1: First capture: falling edge; Second capture: rising edge; Third capture: falling edge. T2CON[5] = 0: First capture: rising edge; Second capture: falling edge; Third capture: rising edge. During continuous captures, the third capture edge from the previous cycle will serve as the first capture edge for the next cycle.

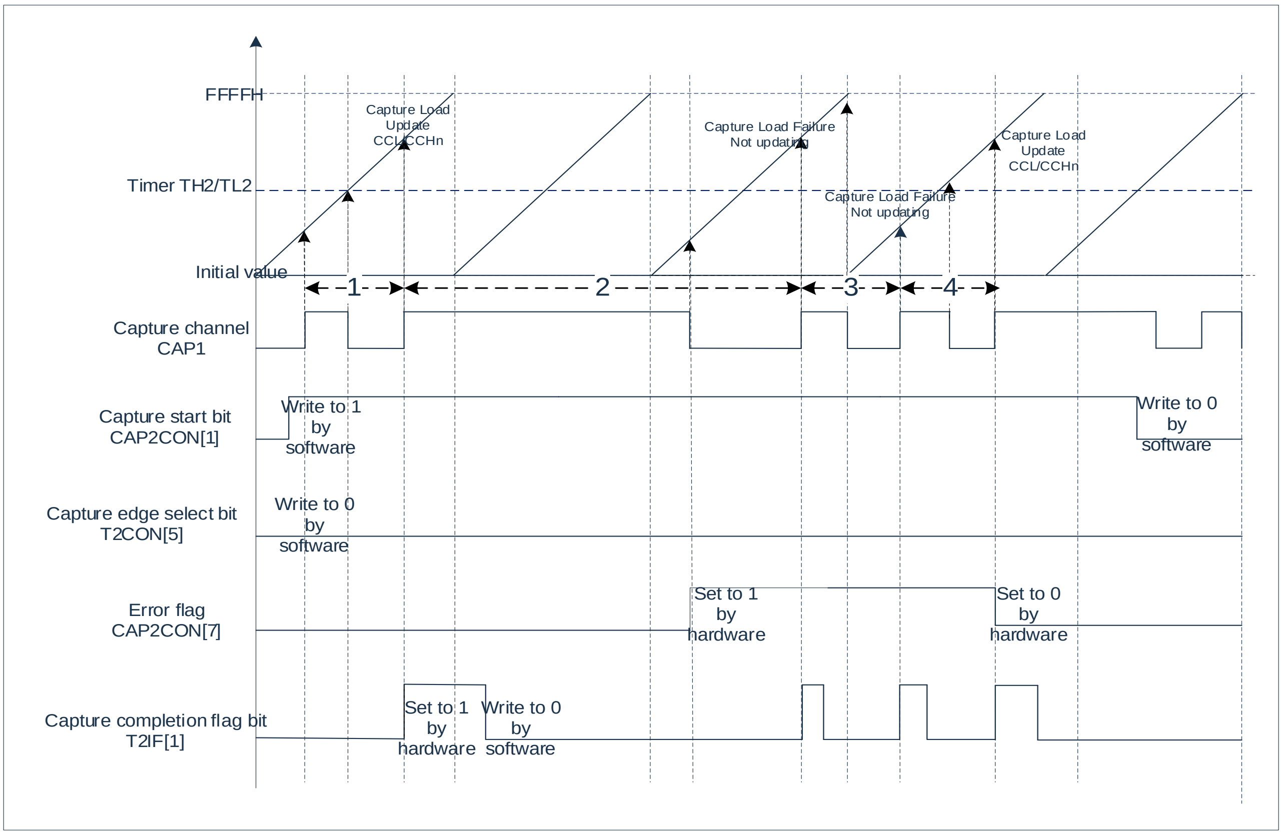
Capture mode 2 enabling is controlled by the CAP2CON[0] register. Capture start is controlled by the CAP2CON[1] register. After completing the three captures, the interrupt flag T2IF[1] is set to 1. If a timer overflow occurs during the second or third captures, the CAP2CON[7] will be set to 1 by hardware until the next complete capture process occurs without timer overflow, at which point CAP2CON[7] will be reset to 0.

Capture Mode 2 supports a read latch feature. When this feature is enabled (CAP2CON[3] = 1), the values in the registers CCL1/CCH1, CCL2/CCH2, and CCL3/CCH3 will correspond to the most recent capture values at the time CAP2CON[3] is set to 1, and these values will be locked and not updated during new captures. When the read latch function is disabled (CAP2CON[3] = 0), the CCL1/CCH1, CCL2/CCH2, CCL3/CCH3 registers will continuously update with new capture values. It is recommended to enable the read latch feature when reading captured data, and to disable it after completing the data reading.

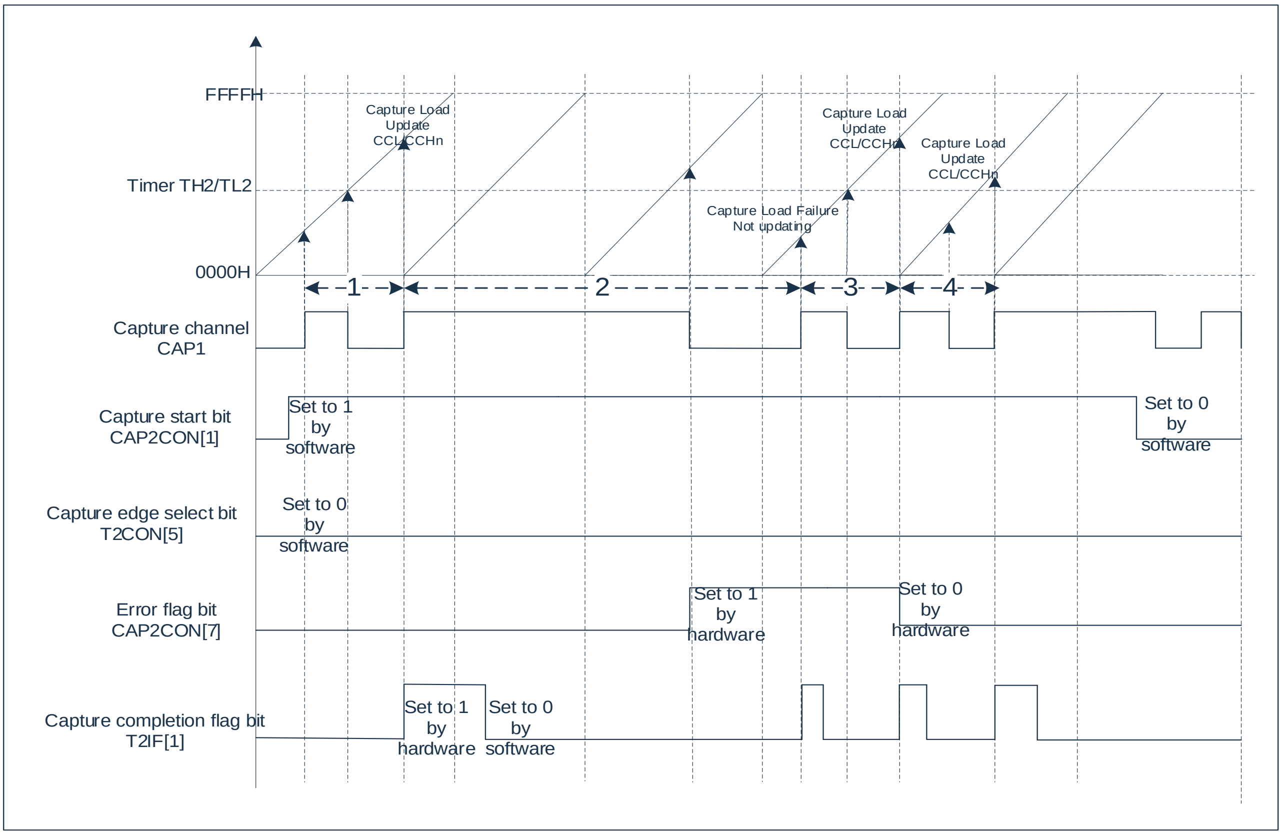
In Capture Mode 2, it is possible to configure the timer to clear upon capture event completion (CAP2CON[2] = 1). When this option is enabled, the timer will restart from 0 after the third capture of each capture cycle, meaning that for all captures following the first one, CCL1/CCH1 will typically be 0, except for the first capture which may not be 0.

Note: When enabling the capture completion clear timer function in Capture Mode 2, do not use it concurrently with T2EX falling edge automatic reload mode. This is because it becomes unclear whether the captured value corresponds to the reloaded state, which could lead to inaccuracies in the captured results.

When CAP2CON[2] = 0, the structural block diagram for Capture Mode 2 is as follows:



When CAP2CON[2]=1, the block diagram of capture mode 2 is shown below:


FFFFH
Timer TH2/TL2
0000H
Capture Load Update CCL/CCHn
Capture Load Failure Not updating
Capture Load Update CCL/CCHn
Capture Load Update CCL/CCHn
1
2
3
4
Capture channel CAP1
Capture start bit CAP2CON[1]
Set to 1 by software
Set to 0 by software
Capture edge select bit T2CON[5]
Set to 0 by software
Error flag bit CAP2CON[7]
Set to 1 by hardware
Set to 0 by hardware
Capture completion flag bit T2IF[1]
Set to 1 by hardware
Set to 0 by software

12. Timer 3/4

Timer 3 and Timer 4 are similar to Timer 0 and Timer 1, featuring two 16-bit timers. Timer 3 has four operating modes, while Timer 4 offers three operating modes. Unlike Timer 0 and Timer 1, Timer 3 and Timer 4 are designed solely for timing operations.

When the timers are active, the values in the registers increment every 12 or 4 system clock cycles.

12.1 Overview

Timers 3 and 4 are composed of two 8-bit registers each: {TH3, TL3} for Timer 3 and {TH4, TL4} for Timer 4. Both Timer 3 and Timer 4 operate in four identical modes. The modes for Timer 3 and Timer 4 are described as follows:

Mode	M1	M0	Function summary
0	0	0	13-bit Timer: Composed of THx[7:0] and TLx[4:0]
1	0	1	16-bit Timer: Composed of THx[7:0] and TLx[7:0].
2	1	0	8-bit Auto-Reload Timer: Composed of TLx[7:0], which reloads from THx.
3	1	1	TL3 and TH3 function as two 8-bit timers. Timer 4 can be configured to stop counting.

12.2 Relevant registers

12.2.1 Timer3/4 control register (T34MOD)

0xD2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T34MOD	TR4	T4M	T4M1	T4M0	TR3	T3M	T3M1	T3M0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	TR4:	Timer4 run control bit
	1=	Start Timer4
	0=	Stop Timer4
Bit6	T4M:	Timer 4 clock selection
	1=	Fsys/4
	0=	Fsys/12
Bit5~Bit4	T4M<1:0>:	Timer 4 mode selection
	00=	Mode0, 13-bit timer
	01=	Mode1, 16-bit timer
	10=	Mode2, 8-bit auto-reload timer
	11=	Mode3, stop counting
Bit3	TR3:	Timer3 run control bit
	1=	Start Timer3
	0=	Stop Timer3
Bit2	T3M:	Timer 3 clock selection
	1=	Fsys/4
	0=	Fsys/12
Bit1~Bit0	T3M<1:0>:	Timer 3 mode selection
	00=	Mode 0, 13-bit timer
	01=	Mode 1, 16-bit timer
	10=	Mode 2, 8-bit auto-reload timer
	11=	Mode 3, two independent 8-bit timers

12.2.2 Timer3 low bit data register (TL3)

0xDA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL3	TL37	TL36	TL35	TL34	TL33	TL32	TL31	TL30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TL3<7:0>: Timer 3 low data register (also used as timer low bit).

12.2.3 Timer3 high bit data register (TH3)

0xDB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH3	TH37	TH36	TH35	TH34	TH33	TH32	TH31	TH30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TH3<7:0>: Timer 3 high data register (also used as timer high bit).

12.2.4 Timer4 low bit data register (TL4)

0xE2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL4	TL47	TL46	TL45	TL44	TL43	TL42	TL41	TL40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TL4<7:0>: Timer 4 low data register (also used as timer low bit).

12.2.5 Timer4 high bit data register (TH4)

0xE3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TH4	TH47	TH46	TH45	TH44	TH43	TH42	TH41	TH40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 TH4<7:0>: Timer 4 high data register (also used as timer high bit).

12.3 Timer3/4 interrupts

Timers 3 and 4 can be enabled or disabled for interrupts through the EIE2 register, and the interrupt priority can be set to high or low using the EIP2 register. The relevant interrupt bits are as follows.

12.3.1 Interrupt mask register (EIE2)

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI interrupt enable bit
1= Enable SPI interrupt
0= Disable SPI interrupt

Bit6 I2CIE I^{2}C interrupt enable bit
1= Enable I^{2}C interrupt
0= Disable I^{2}C interrupt

Bit5 WDTIE: WDT interrupt enable bit
1= Enable WDT overflow interrupt
0= Disable WDT overflow interrupt

Bit4 ADCIE: ADC interrupt enable bit
1= Enable ADC interrupt
0= Disable ADC interrupt

Bit3 PWMIE: PWM global interrupt enable bit
1= Enable all PWM interrupts
0= Disable all PWM interrupts

Bit2 -- Reserved, set to 0.

Bit1 ET4: Timer4 interrupt enable bit
1= Enable Timer4 interrupt
0= Disable Timer4 interrupt

Bit0 ET3: Timer3 interrupt enable bit
1= Enable Timer3 interrupt
0= Disable Timer3 interrupt

12.3.2 Interrupt priority control register (EIP2)

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	PSPI:	SPI interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit6	PI2C:	I^{2}C interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit5	PWDT:	WDT interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit4	PADC:	ADC interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit3	PPWM:	PWM interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit2	--	Reserved, set to 0.
Bit1	PT4:	TIMER4 interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit0	PT3:	TIMER3 interrupt priority control bit
	1=	High priority level
	0=	Low priority level

12.3.3 Peripheral interrupt flag bit register (EIF2)

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	--	TF4	TF3
R/W	R	R	--	R/W	R	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIF: SPI global interrupt status bit (read only)
1= SPI interrupt occurred (automatically cleared after specific interrupt flags are cleared)
0= SPI no interrupt occurred

Bit6 I2CIF: I^{2}C global interrupt status bit (read only)
1= I^{2}C interrupt occurred (automatically cleared after specific interrupt flags are cleared)
0= I^{2}C no interrupt occurred

Bit5 --: Reserved, set to 0.

Bit4 ADCIF: ADC interrupt flag
1= ADC conversion completed (must be cleared by software)
0= ADC conversion not completed

Bit3 PWMIF: PWM global interrupt status bit (read only)
1= PWM interrupt occurred (automatically cleared after specific interrupt flags are cleared)
0= PWM no interrupt occurred

Bit2 --: Reserved, set to 0.

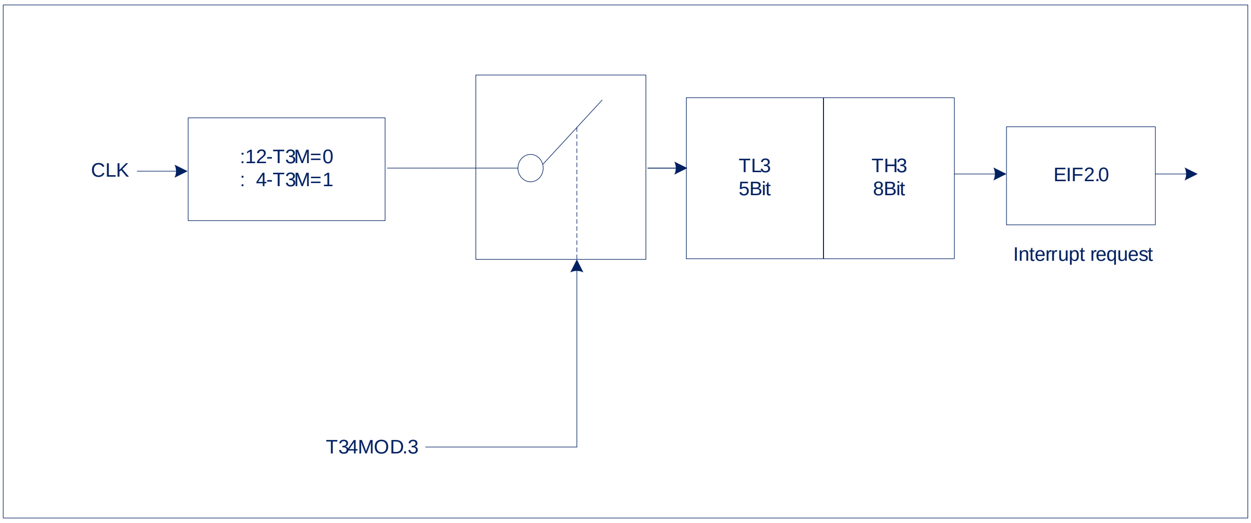
Bit1 TF4: Timer4 overflow interrupt flag
1= Timer4 overflow (automatically cleared by hardware upon entering the interrupt service routine; can also be cleared by software)
0= Timer4 no overflow

Bit0 TF3: Timer3 overflow interrupt flag
1= Timer3 overflow (automatically cleared by hardware upon entering the interrupt service routine; can also be cleared by software)
0= Timer3 no overflow

12.4 Timer3 operation modes

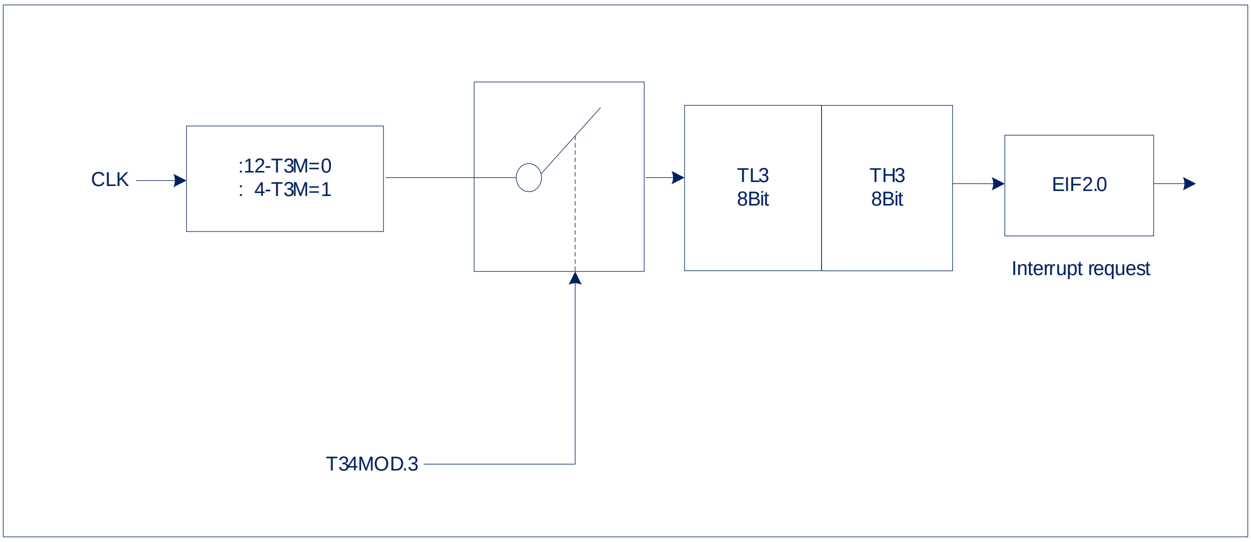
12.4.1 T3 -Mode 0 (13-bit timing mode)

In this mode, Timer 3 is a 13-bit register. When all bits of the timer flip from 1 to 0, the Timer 3 interrupt flag (TF3) is set to 1. The 13-bit register consists of TH3 and the lower 5 bits of TL3. The higher 3 bits of TL3 should be ignored. The structure diagram for Timer 3 Mode 0 is shown below:



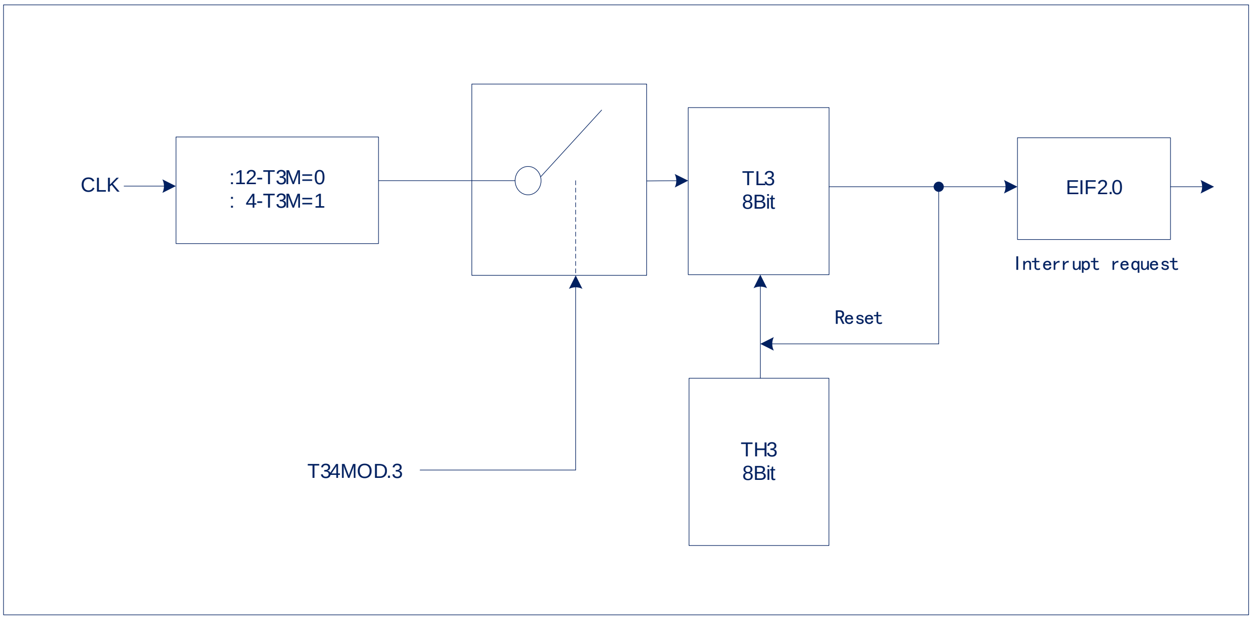
12.4.2 T3 -Mode 1 (16-bit timing mode)

Mode 1 is similar to Mode 0, except that in Mode 1, the Timer 3 register operates as a full 16-bit register. The structure diagram for Timer 3 Mode 1 is shown below:



12.4.3 T3 -Mode 2 (8-bit auto-reload timer mode)

In Mode 2, Timer 3 operates as an 8-bit timer (TL3) with an automatic reload feature, as shown in the diagram below. An overflow from TL3 not only sets TF3 to 1 but also reloads the content of TH3 into TL3 via software. During this reload process, the value of TH3 remains unchanged. The structure diagram for Timer 3 Mode 2 is shown below:



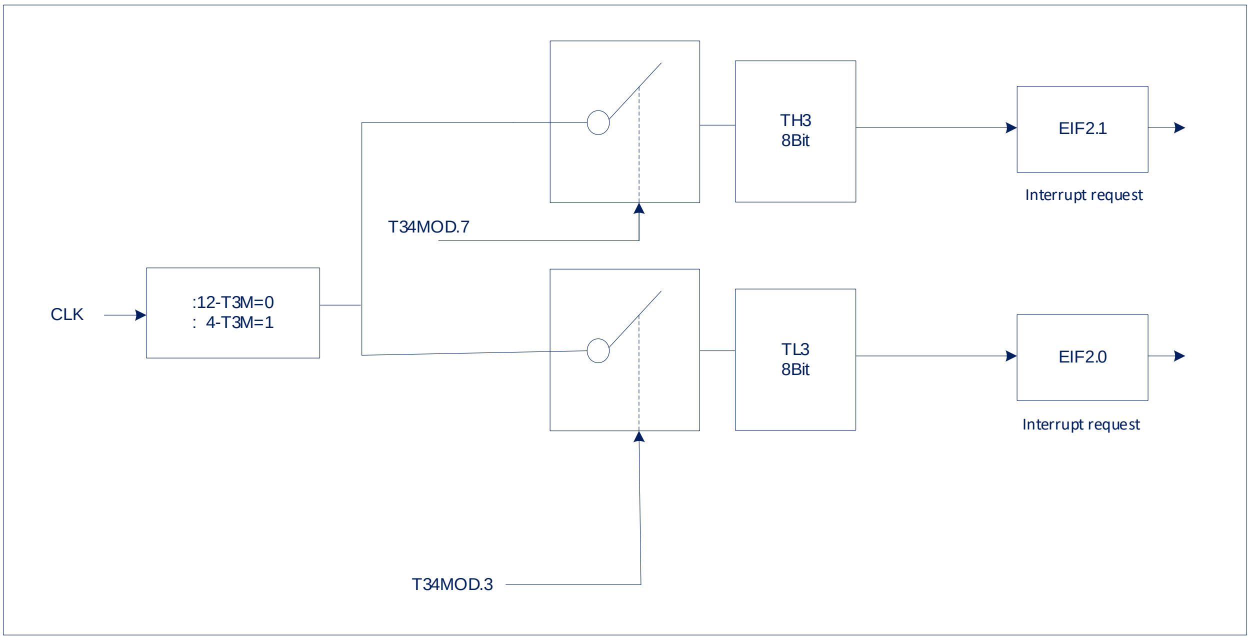
12.4.4 T3 -Mode 3 (two separate 8-bit timers)

In Mode 3, Timer 3 configures TL3 and TH3 as two independent timers. The logic for Timer 3 Mode 3 is illustrated in the diagram below.

TL3 functions as an 8-bit timer and uses Timer 3 control bits, such as TR3 and TF3.

TH3 also operates as an 8-bit timer, utilizing Timer 4's TR4 and TF4 flags to control Timer 4 interrupts.

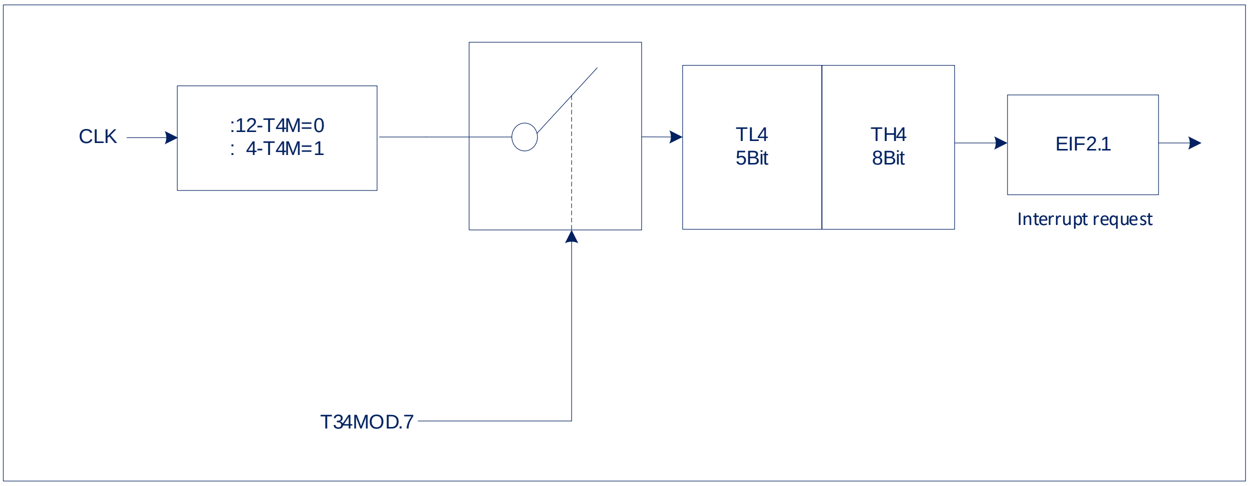
Mode 3 is useful when two 8-bit timers are needed. While Timer 3 is in Mode 3, Timer 4 can be disabled by switching to its own Mode 3 or can still be used as a baud rate generator through the serial channel, or in any application where Timer 4 interrupts are not required. The structure diagram for Timer 3 Mode 3 is shown below:



12.5 Timer4 operation modes

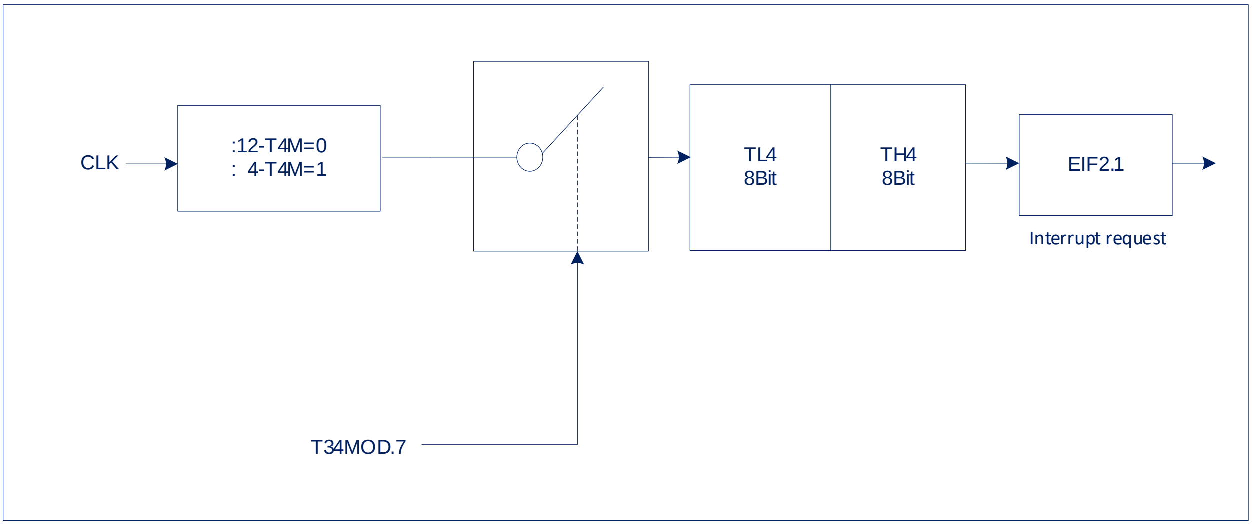
12.5.1 T4 -Mode 0 (13-bit timing mode)

In this mode, Timer 4 is a 13-bit register. When all bits of the timer transition from 1 to 0, the Timer 4 interrupt flag (TF4) is set to 1. The 13-bit register is composed of the 8-bit TH4 and the lower 5 bits of TL4. The higher three bits of TL4 are ignored. The structure diagram for Timer 4 Mode 0 is shown below:



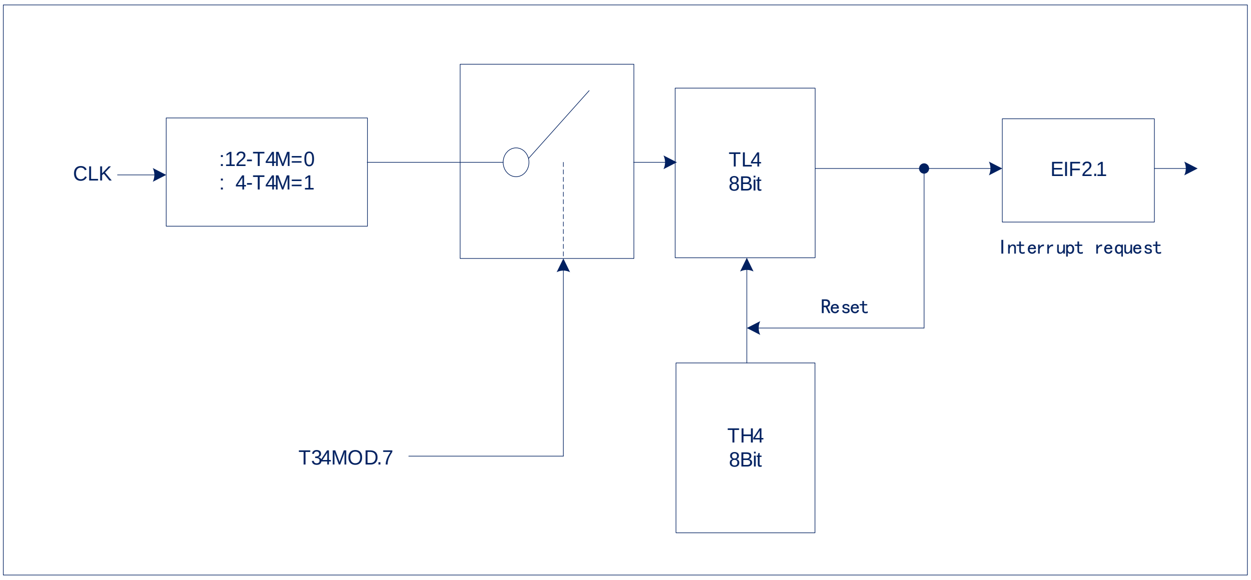
12.5.2 T4 -Mode 1 (16-bit timing mode)

Mode 1 is similar to Mode 0, but in Mode 1, the Timer 4 register operates as a full 16-bit register. The structure diagram for Timer 4 Mode 1 is shown below:



12.5.3 T4 -Mode 2 (8-bit auto-reload timer mode)

In Mode 2, Timer 4 functions as an 8-bit timer (TL4) with an auto-reload feature. When TL4 overflows, it not only sets the Timer 4 interrupt flag (TF4) to 1 but also automatically reloads the content of TH4 into TL4. During this reload process, the value of TH4 remains unchanged. The structure diagram for Timer 4 Mode 2 is shown below:



12.5.4 T4 -Mode 3 (stop counting)

In Mode 3, Timer 4 stops counting, which has the same effect as setting TR4 to 0.

13. LSE Timer

13.1 Overview

The LSE timer is a 16-bit up-counting timer with a clock source from an external low-speed clock (LSE). When using the LSE timer functionality, the LSE module should first be enabled, and then you must wait for the LSE clock to stabilize (approximately 1.5 seconds) before enabling the LSE count. The counter increments by 1 on the rising edge of the LSE clock. When the counter value equals the set timing value, the interrupt flag LSECON[0] is set to 1, and the counter restarts from 0. The timing value is configured using the register {LSECRH[7:0], LSECRL[7:0]}.

If the LSE timing function is configured before entering sleep mode, both the LSE oscillator and the LSE timer can continue to operate unaffected during sleep. If the LSE timer wake-up function is set before sleep, the system will be awakened when the counter value equals the timing value.

13.2 Relevant registers

13.2.1 LSE timer low 8-bit data register (LSECRL)

F694H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LSECRL	LSED7	LSED6	LSED5	LSED4	LSED3	LSED2	LSED1	LSED0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	1	1	1	1	1	1	1

Bit7~Bit0 LSED<7:0>: LSE timing/wakeup time data low 8 bits.

13.2.2 LSE timer high 8-bit data register (LSECRH)

F695H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LSECRH	LSED15	LSED14	LSED13	LSED12	LSED11	LSED10	LSED9	LSED8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	1	1	1	1	1	1	1

Bit7~Bit0 LSED<15:8>: LSE timing/wakeup time data high 8 bits.

13.2.3 LSE timer control register (LSECON)

F696H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LSECON	LSEEN	LSEWUEN	LSECNTEN	LSESTA	LSEIE	--	--	LSEIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	LSEEN:	LSEmodule enable control
	1=	Enable
	0=	Disable
Bit6	LSEWUEN:	LSE timer wake-up enable control
	1=	Enable
	0=	Disable
Bit5	LSECNTEN:	LSE timer count enable control
	1=	Enable
	0=	Disable
Bit4	LSESTA:	LSE stability status bit (read-only)
	1=	LSE is stable
	0=	LSE is unstable
Bit3	LSEIE:	LSE timer interrupt enable control
	1=	Enable
	0=	Disable
Bit2~Bit1	--	Reserved, set to 0.
Bit0	LSEIF:	LSE timer interrupt flag (cleared by software)
	1=	An interrupt generated
	0=	No interrupt or interrupt is cleared

13.3 Interrupt and sleep wake-up

The LSE timer can enable or disable interrupts through the LSECON register and set high/low priority using the EIP1 register. The relevant interrupt-related bits are as follows.

0xB9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP1	PACMP	PLSE_SCM	--	--	PP3	PP2	PP1	PP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit7 PACMP: Analog comparator interrupt priority control bit
1= High priority level
0= Low priority level

Bit6 PLSE_SCM Low-speed crystal timer and crystal stop detection interrupt priority control bit
1= High priority level
0= Low priority level

Bit5~ Bit4 -- Reserved, set to 0.

Bit3 PP3: P3 port interrupt priority control bit
1= High priority level
0= Low priority level

Bit2 PP2: P2 port interrupt priority control bit
1= High priority level
0= Low priority level

Bit1 PP1: P1 port interrupt priority control bit
1= High priority level
0= Low priority level

Bit0 PP0: P0 port interrupt priority control bit
1= High priority level
0= Low priority level

When the count value of the LSE timer equals the timing value, the timer interrupt flag LSEIF is set to 1. If global interrupts are enabled (EA=1) and LSE timer interrupts are enabled (LSEIE=1), the CPU will execute the interrupt service routine.

To use the LSE timer interrupt to wake up from sleep mode, it is necessary to enable LSEEN, LSECNT, and LSEWUEN before entering sleep, and to set the timing values for waking up as {LSECRH[7:0], LSECRL[7:0]}. If global interrupt enable and LSE interrupt enable are turned on before going to sleep, the interrupt service routine will be executed first upon waking up, and after returning from the interrupt, the next instruction following the sleep command will be executed.

13.4 Function description

To utilize the LSE timer functionality, follow these steps: Set LSEEN = 1 to enable the LSE timer module. Wait until the LSE clock stabilization status bit LSESTA = 1, indicating that the clock is stable. After the clock is stable, configure the timer value using {LSECRH[7:0], LSECRL[7:0]}. Finally, set LSECNT = 1 to enable the counting function of the LSE timer. The LSE timer counts from 0. When the count value equals the configured timer value, the interrupt flag is set to 1. At this moment, the timer value is updated to the last written value in the timer data register {LSECRH[7:0], LSECRL[7:0]} before the count and the timer value were equal. The minimum timer value is 1. If a value of 0 is set, the timer defaults to 1. The duration of the LSE timer can be calculated using the following formula:

LSE timer duration= $\frac{1}{32.768}$ ×({LSECRH[7:0], LSECRL[7:0]}+1) ms

If any of the bits LSEEN, LSECNTEN, or LSESTA are set to 0, the LSE counter value will be reset to 0.

14. Wake Up Timer (WUT)

14.1 Overview

The Wake Up Timer (WUT) is a 12-bit up-counting timer with a clock source from the internal low-speed clock (LSI), designed for waking the system from sleep mode. Before the system enters sleep mode, the wake-up time must be configured and the wake-up timer function enabled. Once the chip enters sleep mode, the WUT begins counting, and when the count value equals the set value, the chip enters the wake-up waiting state.

14.2 Relevant registers

14.2.1 WUTCRH register

0xBD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WUTCRH	WUTEN	TIMER_OV	WUTPS1	WUTPS0	WUTD11	WUTD10	WUTD9	WUTD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	WUTEN:	Wake-up timer enable bit
	1=	Wake-up timer enabled
	0=	Wake-up timer disabled
Bit6	TIMER_OV:	Timer overflow status bit
	1=	Counter overflow
	0=	Cleared by software
Bit5~Bit4	WUTPS<1:0>:	Wake-up timer counter clock prescaler bit
	00=	F/1
	01=	F/8
	10=	F/32
	11=	F/256
Bit3~Bit0	WUTD<11:8>:	Wake-up time data high 4 bits

14.2.2 WUTCRL register

0xBC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
WUTCRL	WUTD7	WUTD6	WUTD5	WUTD4	WUTD3	WUTD2	WUTD1	WUTD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	WUTD<7:0>:	Timer wake-up time data lower 8 bits.

14.3 Function description

The internal wake-up timer operates as follows: after the system enters sleep mode, the CPU and all peripheral circuits stop functioning, while the internal low-power oscillator (LSI) begins to operate, providing an oscillation clock of 125 kHz ($T_{LSI} \approx 8$ µs) to the WUT counter.

There are two internal wake-up timer registers: WUTCRH and WUTRCL.

The Bit7 of the WUTCRH register is the internal wake-up enable bit:

- WUTEN=1: Wake-up timer function enabled
- WUTEN=0: Wake-up timer function disabled

The {WUTCRH[3:0] and WUTCRL[7:0]} together form a 12-bit wake-up time data register. Once the system enters sleep mode, the WUT counter begins counting. When the value of the WUT counter equals the value in the wake-up time data register, the system oscillator is activated, entering the wake-up waiting state.

Wake-up time calculation: $T=(WUTD[11:0]+1)\times WUTPS\times T_{LSI}$

15. Baud Rate Timer (BRT)

15.1 Overview

The chip features a single 16-bit Baud Rate Timer (BRT), which primarily provides the clock for the UART module.

15.2 Relevant registers

15.2.1 BRT module control register (BRTCON)

F5C0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRTCON	BRTEN	--	--	--	--	BRTCKDIV 2	BRTCKDIV 1	BRTCKDIV 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	BRTEN:	BRT timer enable bit
	1=	Enable
	0=	Disable
Bit6~Bit3	--	Reserved, set to 0.
Bit2~Bit0	BRTCKDIV<2:0>	BRT timer prescaler selection bit
	000=	Fsys/1
	001=	Fsys/2
	010=	Fsys/4
	011=	Fsys/8
	100=	Fsys/16
	101=	Fsys/32
	110=	Fsys/64
	111=	Fsys/128

15.2.2 BRT timer data load low 8-bit register (BRTDL)

F5C1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRTDL	BRTDL7	BRTDL6	BRTDL5	BRTDL4	BRTDL3	BRTDL2	BRTDL1	BRTDL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	BRTDL<7:0>:	BRT timer load value low 8 bits

15.2.3 BRT timer data load high 8-bit register (BRTDH)

F5C2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BRTDH	BRTDH7	BRTDH6	BRTDH5	BRTDH4	BRTDH3	BRTDH2	BRTDH1	BRTDH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	BRTDH<7:0>:	BRT timer load value high 8 bits

15.3 Function description

The BRT contains a 16-bit incrementing counter, with its clock source coming from the prescaler circuit. The prescaler clock is determined by the timer prescaler selection bit (BRTCKDIV). The initial value of the counter is loaded from {BRTDH, BRTDL}.

When the timer enable bit BRTEN = 1, the counter starts operating. When the 16-bit counter value equals FFFFH, the BRT counter overflows. After the overflow, the initial count value is automatically reloaded into the counter, and counting resumes.

The overflow signal from the BRT counter is specifically provided to the UART module as the clock source for baud rate generation. The overflow does not generate an interrupt, nor is there a corresponding interrupt structure. In debug mode, the clock for the BRT does not stop. If the UART module has already begun sending or receiving data, it will complete the entire send or receive process even if the chip enters a pause state.

BRT overflow rate:

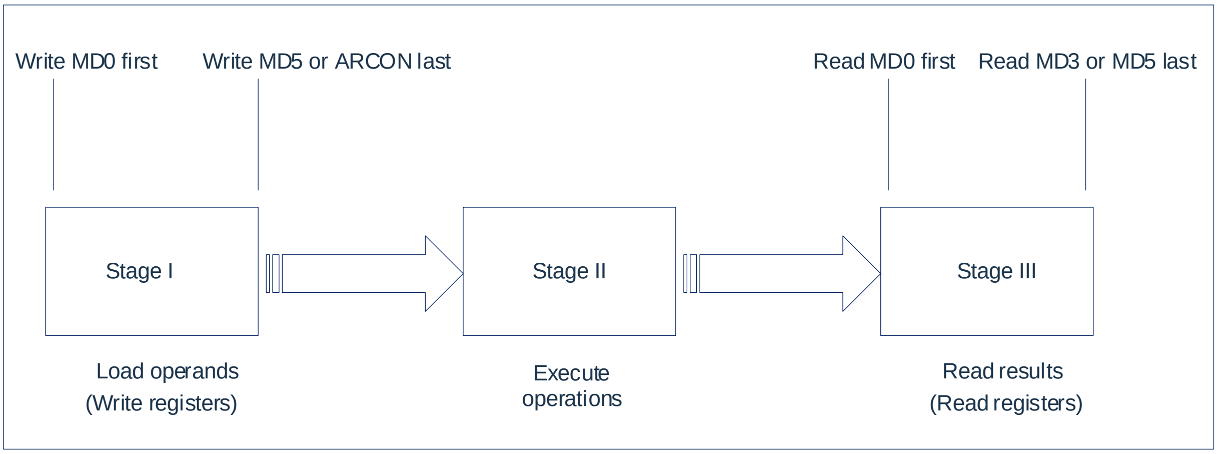
$$\mathrm{BRTov}=\frac{\mathrm{Fsys}}{(65536-\{\mathrm{BRTDH},\mathrm{BRTDL}\})\times 2^{\mathrm{BRTCKDIV}}}$$

16. Multiplication/Division Unit (MDU)

16.1 Overview

The MDU module provides functionalities for 32-bit/16-bit and 16bit/16bit division, 16bit*16bit multiplication, 32-bit shifting operations, and 32-bit normalization. All operations are performed using unsigned integer arithmetic. The shifting operations support both left and right shifts for 32-bit data. The MDU module is controlled by seven registers (MD0/MD1/MD2/MD3/MD4/MD5/ARCON). Registers MD0 to MD5 are used to store operands before the operation, as well as the results and remainders after the operation. ARCON serves as the control register.

The operations of the MDU module can be divided into three stages: loading operands (write registers), executing operations, and reading results (read registers). The first and third stages require register operations and depend on the CPU's execution, while the second stage can operate independently of the CPU. When the MD5 or ARCON register is written to start the MDU operation, the MDU will take a fixed amount of time before the result can be read from the registers. The block diagram of the MDU module operation stages is as follows:



16.2 Relevant registers

The operation of the MDU module is controlled by registers MD0, MD1, MD2, MD3, MD4, MD5, and ARCON, each of which is described as follows.

16.2.1 Manipulation register (MD0)

0XE9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD0	MD07	MD06	MD05	MD04	MD03	MD02	MD01	MD00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 MD0<7:0>: 32bit/16bit division: Write bit7-bit0 for the divisor, read bit7-bit0 for the quotient.
16bit/16bit division: Write bit7-bit0 for the divisor, read bit7-bit0 for the quotient.
16bit*16bit multiplication: Write bit7-bit0 for the first multiplier, read bit7-bit0 for the product.
Shifting: Write/read data bit7-bit0.
Normalization: Write/read data bit7-bit0.

16.2.2 Manipulation register (MD1)

0XEA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD1	MD17	MD16	MD15	MD14	MD13	MD12	MD11	MD10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 MD1<7:0>: 32bit/16bit division: Write bit15-bit8 for the divisor, read bit15-bit8 for the quotient.
16bit/16bit division: Write bit15-bit8 for the divisor, read bit15-bit8 for the quotient.
16bit*16bit multiplication: Write bit15-bit8 for the first multiplier, read bit15-bit8 for the product.
Shifting: Write/read data bit15-bit8.
Normalization: Write/read data bit15-bit8.

16.2.3 Manipulation register (MD2)

0XEB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD2	MD27	MD26	MD25	MD24	MD23	MD22	MD21	MD20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 MD2<7:0>: 32bit/16bit division: Write bit23-bit16 for the divisor, read bit23-bit16 for the quotient.
16bit*16bit multiplication: Read bit23-bit16 for the product.
Shifting: Write/read data bit23-bit16.
Normalization: Write/read data bit23-bit16.

16.2.4 Manipulation register (MD3)

0XEC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD3	MD37	MD36	MD35	MD34	MD33	MD32	MD31	MD30
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 MD3<7:0>: 32bit/16bit division: Write bit31-bit24 for the divisor, read bit31-bit24 for the quotient.
16bit*16bit multiplication: Read bit31-bit24 for the product.
Shifting: Write/read data bit31-bit24.
Normalization: Write/read data bit31-bit24.

16.2.5 Manipulation register (MD4)

0XED	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD4	MD47	MD46	MD45	MD44	MD43	MD42	MD41	MD40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 MD4<7:0>: 32bit/16bit division: Write bit7-bit0 for the divisor, read bit7-bit0 for the remainder.
16bit/16bit division: Write bit7-bit0 for the divisor, read bit7-bit0 for the remainder.
16bit*16bit multiplication: Write bit7-bit0 for the second multiplier.

16.2.6 Manipulation register (MD5)

0XEE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MD5	MD57	MD56	MD55	MD54	MD53	MD52	MD51	MD50
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 MD5<7:0>: 32bit/16bit division: Write bit15-bit8 for the divisor, read bit15-bit8 for the remainder.
16bit/16bit division: Write bit15-bit8 for the divisor, read bit15-bit8 for the remainder.
16bit*16bit multiplication: Write bit15-bit8 for the second multiplier.

16.2.7 Manipulation register (ARCON)

0XEF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ARCON	MDEF	MDOV	SLR	SC4	SC3	SC2	SC1	SC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 MDEF: Error flag (set to 1 by hardware)

1= The error flag is set to 1 during the data loading process when writing to the MD0 to MD5 registers

0= This flag is cleared to 0 when reading from the ARCON register

Bit6 MDOV: Overflow flag (set to 1 by hardware)

1= The divisor is zero

The multiplication result exceeds 0xffff

The normalization operation sets MD37 to 1

0= This bit is 0 when the condition of 1 is not matched (this bit is cleared by writing to the MD0 register, and writing to 0 by software is not valid).

Bit5 SLR: Direction control bit of the shift operation

1= Right shift

0= Left shift

Bit4~Bit0 SC<4:0>: Shift count/normalization count

Writing 00000 initializes the normalization operation functionality, and after completion, it stores the normalization count.

Writing a value other than 00000 specifies the number of bits to shift, initiating the shift operation functionality.

16.3 Function description

The MDU module's division and multiplication operations are determined by the order in which MD0 to MD5 registers are written. The shifting and normalization functions are controlled by the ARCON register. The operation sequence for the MDU module is as follows:

Operation phase	Operation order	32bit/16bit	16bit/16bit	16bit*16bit
First phase	First write	MD0 (Dividend bits 7-0)	MD0 (Dividend bits 7-0)	MD0 (Multiplier 0 bits 7-0)
		MD1 (Dividend bits 15-8)	MD1 (Dividend bits 15-8)	MD4 (Multiplier 1 bits 7-0)
		MD2 (Dividend bits 23-16)	--	--
		MD3 (Dividend bits 31-24)	--	--
		MD4 (Divisor bits 7-0)	MD4 (Divisor bits 7-0)	MD1 (Multiplier 0 bits 15-8)
	Last write	MD5 (Divisor bits 15-8)	MD5 (Divisor bits 15-8)	MD5 (Multiplier 1 bits 15-8)
Third phase	First read	MD0 (Quotient bits 7-0)	MD0 (Quotient bits 7-0)	MD0 (Product bits 7-0)
		MD1 (Quotient bits 15-8)	MD1 (Quotient bits 15-8)	MD1 (Product bits 15-8)
		MD2 (Quotient bits 23-16)	--	--
		MD3 (Quotient bits 31-24)	--	--
		MD4 (Remainder bits 7-0)	MD4 (Remainder bits 7-0)	MD2 (Product bits 23-16)
	Last read	MD5 (Remainder bits 15-8)	MD5 (Remainder bits 15-8)	MD3 (Product bits 31-24)

All operations of the MDU are performed by hardware, providing high conversion rates and saving significant time during program execution. The operational timing for the second phase of the five MDU operations is as follows:

Function	Result bit width	Remainder bit width	Operation time (second phase)
32bit/16bit division	32bit	16 bit	16 Tsys
16bit/16bit division	16 bit	16 bit	8 Tsys
16bit*16bit division	32bit	--	8 Tsys
32bit shift operation	32bit	--	2~17 Tsys
32bit normalization operation	32bit	--	1~17 Tsys

In the tables above, Tsys represents the clock cycle of the MDU module; the operation time indicates the computation time of the MDU module, excluding register read/write times. The timing for shift and normalization operations varies based on the number of bits shifted or the operands used, with shift operation timing ranging from a minimum of 2 Tsys to a maximum of 17 Tsys, and normalization operation timing ranging from 1 Tsys to 17 Tsys.

16.3.1 32bit/16bit division operation

The steps for a 32-bit/16-bit division operation are as follows:

1) Write to register MD0 (bits 7-0 of the dividend)
2) Write to register MD1 (bits 15-8 of the dividend)
3) Write to register MD2 (bits 23-16 of the dividend)
4) Write to register MD3 (bits 31-24 of the dividend)
5) Write to register MD4 (bits 7-0 of the divisor);
6) Write to register MD5 (bits 15-8 of the divisor), then start the division operation.
7) Wait for 16 clock cycles of the MDU module to ensure the calculation is completed.
8) Read from register MD0 (bits 7-0 of the quotient).
9) Read from register MD1 (bits 15-8 of the quotient).
10) Read from register MD2 (bits 23-16 of the quotient).
11) Read from register MD3 (bits 31-24 of the quotient).
12) Read from register MD4 (bits 7-0 of the remainder).
13) Read from register MD5 (bits 15-8 of the remainder). This completes one division operation.

If the result is not read after the operation is completed, you can rewrite MD0 to start the next operation.

For example, if the dividend is 0x87654321 and the divisor is 0x1234, write: MD0=0X21, MD1=0X43, MD2=0X65, MD3=0X87, MD4=0X34, MD5=0X12. After the MDU operation is completed, the results will be: MD0=0X23, MD1=0X70, MD2=0X07, MD3=0X00, MD4=0X05, MD5=0X06.

16.3.2 16-bit/16-bit division operation

The steps for a 16-bit/16-bit division operation are as follows:

1) Write to register MD0 (bits 7-0 of the dividend).
2) Write to register MD1 (bits 15-8 of the dividend).
3) Write to register MD4 (bits 7-0 of the divisor).
4) Write to register MD5 (bits 15-8 of the divisor), then start the division operation.
5) Wait for 8 clock cycles of the MDU module.
6) Read from register MD0 (bits 7-0 of the quotient).
7) Read from register MD1 (bits 15-8 of the quotient).
8) Read from register MD4 (bits 7-0 of the remainder).
9) Read from register MD5 (bits 15-8 of the remainder). This completes one division operation.

If the result is not read after the operation is completed, you can rewrite MD0 to start the next operation.

For example, if the dividend is 0x4321 and the divisor is 0x1234, write: MD0=0X21, MD1=0X43, MD4=0X34, MD5=0X12. After the MDU operation is completed, the results will be: MD0=0X03, MD1=0X00, MD4=0X85, MD5=0X0C.

16.3.3 16bit*16bit multiplication operation

The steps for a 16bit*16bit multiplication operation are as follows:

1) Write to register MD0 (bits 7-0 of the first multiplier).
2) Write to register MD4 (bits 7-0 of the first multiplier).
3) Write to register MD1 (bits 15-8 of the first multiplier).
4) Write to register MD5 (bits 15-8 of the second multiplier), then start the multiplication operation.
5) Wait for 8 clock cycles of the MDU module.
6) Read from register MD0 (bits 7-0 of the product).
7) Read from register MD1 (bits 15-8 of the product).
8) Read from register MD2 (bits 23-16 of the product).
9) Read from register MD3 (bits 31-24 of the product). This completes one multiplication operation.

If the result is not read after the operation is completed, you can rewrite MD0 to start the next operation.

For example, if the first multiplier is 0x8765 and the second multiplier is 0x1234, write: MD0=0X65, MD4=0X34, MD1=0X87, MD5=0X12. After the MDU operation is completed, the results will be: MD0=0X84, MD1=0X9A, MD2=0XA0, MD3=0X09.

16.3.4 32bit shift operation

The steps for the 32-bit shift operation are as follows:

1) Write to register MD0 (bits 7-0 of the operand).
2) Write to register MD1 (bits 15-8 of the operand).
3) Write to register MD2 (bits 23-16 of the operand).
4) Write to register MD3 (bits 31-24 of the operand).
5) Write to register ARCON, then start the shift operation.
6) Wait for 17 clock cycles of the MDU module to ensure the operation is completed.
7) Read from register MD0 (bits 7-0 of the shifted result).
8) Read from register MD1 (bits 15-8 of the shifted result).
9) Read from register MD2 (bits 23-16 of the shifted result).
10) Read from register MD3 (bits 31-24 of the shifted result). This completes one shift operation.

If the result is not read after the operation is completed, you can rewrite MD0 to start the next operation.

For example, if the operand is 0x12345678 and you want to right shift by 5 bits, write: MD0=0X78, MD1=0X56, MD2=0X34, MD3=0X12, ARCON=0X25. After the MDU operation is completed, the results will be: MD0=0XB3, MD1=0XA2, MD2=0X91, MD3=0X00.

16.3.5 32bit normalization operation

Normalization is the process of left-shifting an operand until the highest bit is 1. The steps for the 32-bit normalization operation are as follows:

1) Write to register MD0 (bits 7-0 of the operand).
2) Write to register MD1 (bits 15-8 of the operand).
3) Write to register MD2 (bits 23-16 of the operand).
4) Write to register MD3 (bits 31-24 of the operand).
5) Write to register ARCON = 0x00, then start the normalization operation.
6) Wait for 17 clock cycles of the MDU module to ensure the operation is completed.
7) Read from register MD0 (bits 7-0 of the normalized result).
8) Read from register MD1 (bits 15-8 of the normalized result).
9) Read from register MD2 (bits 23-16 of the normalized result).
10) Read from register MD3 (bits 31-24 of the normalized result). This completes one normalization operation.

If the result is not read after the operation is completed, you can rewrite MD0 to start the next operation.

For example, if the operand is 0x12345678, during the normalization operation, write: MD0=0X78, MD1=0X56, MD2=0X34, MD3=0X12, ARCON=0X00. After the MDU operation is completed, the results will be: MD0=0XC0, MD1=0XB3, MD2=0XA2, MD3=0X91, ARCON=0X03.

17. Buzzer Driver (BUZZER)

17.1 Overview

The buzzer driver module consists of an 8-bit counter, a clock driver, and control registers. The buzzer output generates a 50% duty cycle square wave, with the frequency set by the BUZCON and BUZDIV registers. This allows for a wide range of frequency outputs.

17.2 Relevant registers

17.2.1 BUZZER control register (BUZCON)

0xBF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BUZCON	BUZEN	--	--	--	--	--	BUZCKS1	BUZCKS0
R/W	R/W	R	R	R	R	R	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	BUZEN:	BUZZER enable bit
	1=	Enable
	0=	Disable
Bit6~Bit2	--	Reserved, set to 0.
Bit1~Bit0	BUZCKS<1:0>:	BUZZER frequency division ratio selection bit
	00=	Fsys/8
	01=	Fsys/16
	10=	Fsys/32
	11=	Fsys/64

17.2.2 BUZZER frequency control register (BUZDIV)

0xBE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
BUZDIV	BUZDIV7	BUZDIV6	BUZDIV5	BUZDIV4	BUZDIV3	BUZDIV2	BUZDIV1	BUZDIV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	BUZDIV<7:0>:	BUZZER frequency select bit
	0x00=	No square wave output
	Others=	Fbuz =Fsys/(2*CLKDIV*BUZCKS)

Note: It is not recommended to modify BUZDIV when BUZEN=1.

17.3 Function description

When using the buzzer, the corresponding port must first be configured as a buzzer driver output port. For example, to configure P24 as a buzzer driver output port, use the following configuration:

P24CFG = 0x18; // Configure P24 as a buzzer driver output port

By configuring the relevant registers of the buzzer driver module, different frequencies can be set for the buzzer driver output. For example:

1) Set Fsys= 8MHz, BUZCKS<1:0>=01, BUZDIV=125

   The buzzer driver output frequency is: Fbuz=8MHz/(2*125)/16= 2KHz

2) Set Fsys=16MHz, BUZCKS<1:0>=11, BUZDIV=125

   The buzzer driver output frequency is: Fbuz=16MHz/(2*125)/64= 1KHz

3) Set Fsys=24MHz, BUZCKS<1:0>=11, BUZDIV=94

   The buzzer driver output frequency is: Fbuz=24MHz/(2*94)/64= 2KHz

By selecting different system clock frequencies and buzzer driver clock division ratios, various output frequencies can be achieved. The buzzer driver output frequencies are shown in the table below:

BUZCKS<1:0>	Fbuz@Fsys=8MHz	Fbuz@Fsys=16MHz	Fbuz@Fsys=24MHz	Fbuz@Fsys=48MHz
00	2KHz~500KHz	4KHz~1MHz	6KHz~1.5MHz	12KHz~3MHz
01	1KHz~250KHz	2KHz~500KHz	3KHz~750KHz	6KHz~1.5MHz
10	0.5KHz~125KHz	1KHz~250KHz	1.5KHz~375KHz	3KHz~750KHz
11	0.25KHz~62.5KHz	0.5KHz~125KHz	0.75KHz~187.5KHz	1.5KHz~375KHz

18. Enhanced PWM Module

18.1 Overview

18.1.1 Functions

The Enhanced PWM module supports six PWM generators, which can be configured for six independent PWM outputs (PG0-PG5) or as three pairs of complementary PWM signals with programmable dead time (PG0-PG1, PG2-PG3, PG4-PG5).

Each pair of PWM signals shares an 8-bit prescaler, with six groups of clock dividers providing five division factors (1, 1/2, 1/4, 1/8, 1/16). Each PWM output is controlled by an independent 16-bit counter, and a separate 16-bit comparator is used to adjust the duty cycle. The six PWM generators provide 25 interrupt flags; when the period or duty cycle of the relevant PWM channel matches the counter, an interrupt flag is generated. Each PWM has its own enable bit.

Each PWM can be configured for one-shot mode (to produce a single PWM signal cycle) or continuous mode (to continuously output the PWM waveform).

18.1.2 Features

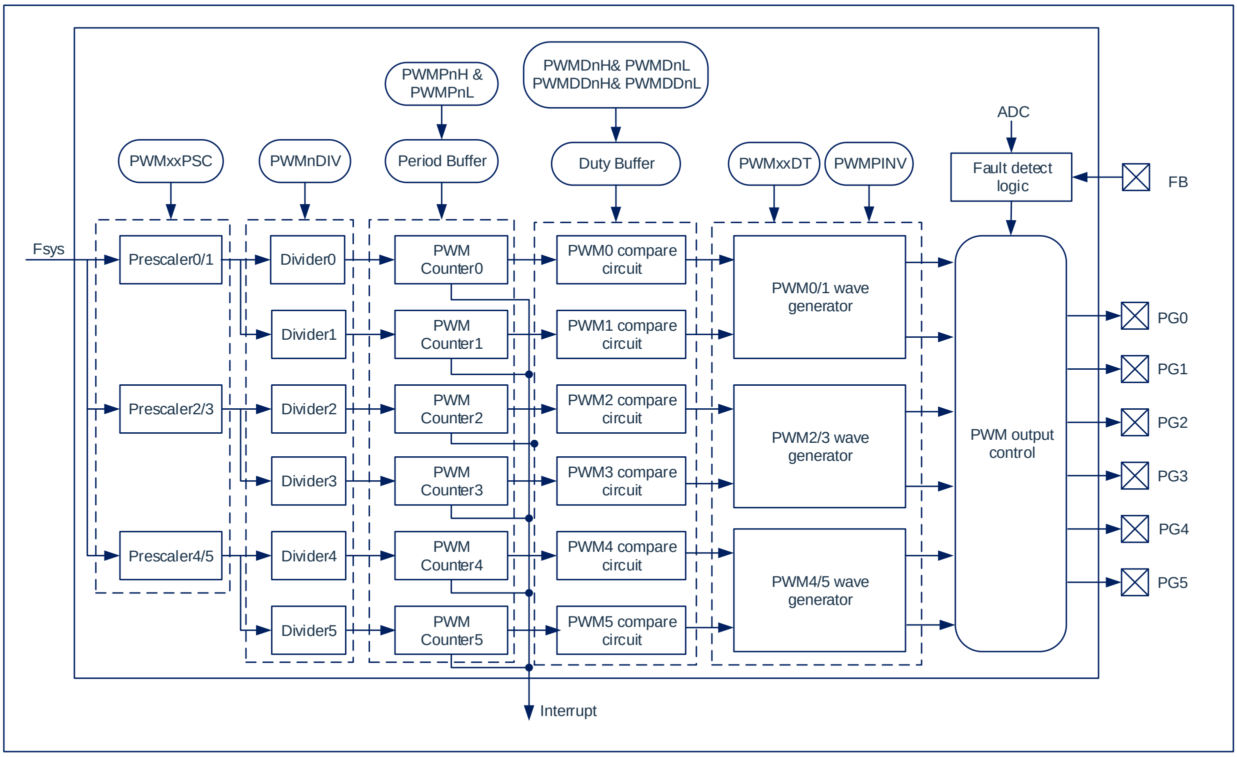
The Enhanced PWM module has the following features:

- ◆ 6 independent 16-bit PWM control modes:
  - 6 independent outputs: PG0, PG1, PG2, PG3, PG4, PG5.
  - Complementary PWM pairs output: (PG0-PG1), (PG2-PG3), (PG4-PG5), with programmable dead time.
  - Synchronized PWM pairs output: (PG0-PG1), (PG2-PG3), (PG4-PG5), with synchronized pins for each pair.
- ◆ Group control support: Outputs PG0, PG2, and PG4 are synchronized; outputs PG1, PG3, and PG5 are synchronized.
- ◆ One-shot mode or auto-load mode.
- ◆ Edge alignment and center alignment modes supported.
- ◆ Center alignment mode: Supports both symmetric and asymmetric counting.
- ◆ Programmable dead time generator in Complementary PWM.
- ◆ Independent polarity control for each PWM.
- ◆ Hardware brake protection and recovery functions: Triggered by external FB, software, or ADC compare events.
- ◆ PWM edge or period can trigger ADC conversion.

18.2 Configuration

18.2.1 Block diagram

The enhanced PWM function block diagram is shown below.



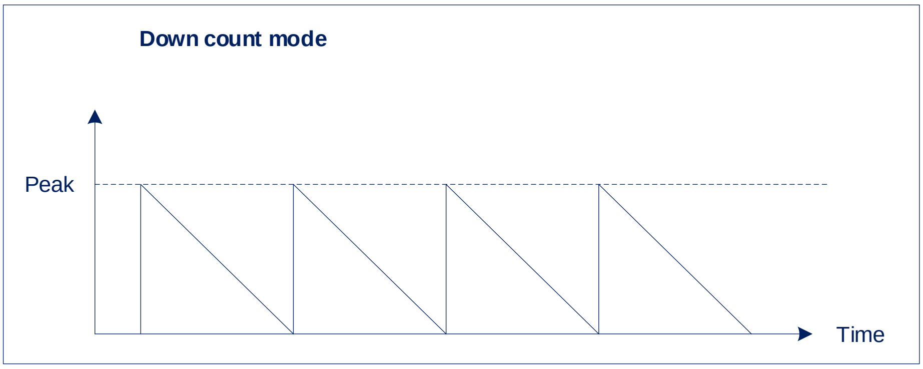
18.2.2 Functional module descriptions

The Enhanced PWM module consists of the PWM counter module, output comparison unit, waveform generator, fault detection, and output controller.

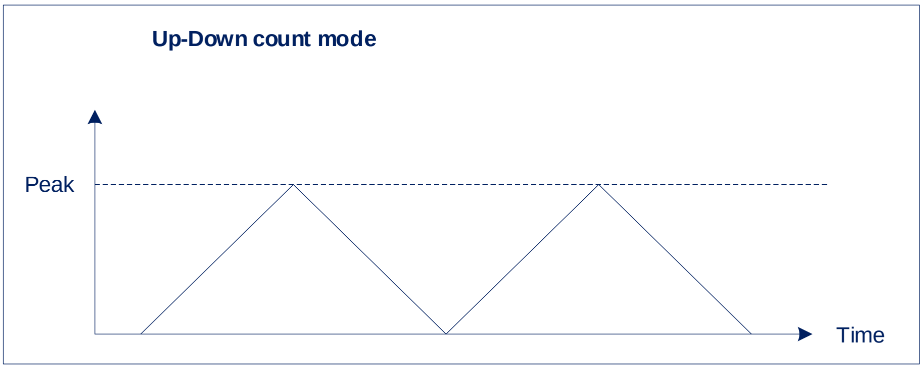
PWM counter:

The system clock is input to the Enhanced PWM module, which uses a prescaler and clock divider to derive the counting clock for the six PWM counters. The period registers (PWMPnH, PWMPnL) form a 16-bit control register used to set the counting period for the six PWM counters. To prevent arbitrary modifications to the PWM period settings during operation, a buffer register (Period Buffer) is employed. If PWM is set to continuous operation mode (PWMnCNTM=1), the value from the period register is automatically loaded into the buffer register (Period Buffer) at each zero-crossing of the PWM.

The PWM counter supports two counting modes: Down Count Mode and Up-Down Count Mode. The down count mode is shown below.



The up-down count mode is shown below.



OCU:

The Output Comparison Unit (OCU) consists of the duty cycle registers (PWMDnH, PWMDnL), which are used to set the duty cycles for the six channels of PWM. To prevent arbitrary modifications to the duty cycle settings during PWM operation, a duty buffer is employed, along with the PWM counter for comparison to flip the output levels. When PWM is set to continuous operation mode (PWMnCNTM=1), the values from the duty cycle registers are automatically loaded into the Duty Buffer at each zero-crossing of the PWM signal.

WFG:

The waveform generator is composed of the dead-time control unit and the output polarity control unit. For complementary outputs with dead time, the PWM01DT, PWM23DT, and PWM45DT registers are used to set the dead time for the PWM signals. Additionally, the output polarity is controlled using the polarity control register (PWMPINV).

Fault detection (brake function):

The fault detection module is integrated within the enhanced PWM circuitry and is configured for input fault detection to protect the system from component damage. Upon detecting a valid fault signal input, the PWM output is forcibly shut down. To accommodate different driving requirements, the shut-down level can be configured via the PWM brake data register (PWMFBKD).

Mask output:

Mask output is particularly important for specific applications such as square wave motor control. Each PWM channel has its own mask control bit and mask data bit, which are set using the mask control register (PWMMASKE) and the mask data register (PWMMASKD).

When mask output is disabled (PWMnMASKE=0), PWMn outputs the normal PWM waveform.

When mask output is enabled (PWMnMASKE=1), PWMn outputs the data from the mask register (PWMnMASKD).

Output controller:

The output controller is responsible for controlling the output state of the PWM signals. The PWM output enable control register (PWMOE) is used to set the output enable state for each channel. In the event of a fault requiring forced shutdown of the PWM, the MCU can output the corresponding level based on the settings in the brake data register (PWMFBKD) to meet the needs of various peripherals.

18.2.3 Related I/O port description

Before using the enhanced PWM module, it is necessary to configure the relevant ports as PWM channels. The PWM channels are labeled PG0-PG5 on the pin allocation diagram, corresponding to PWM channels 0-5. It can be observed that different PWM channels can correspond to the same port, and the same PWM channel can be allocated to different ports. This feature allows the enhanced PWM functionality to adapt to various package types and flexible PCB layout requirements.

The allocation of PWM channels is controlled by the corresponding port configuration registers. For example:

```
P13CFG = 0x12; // Configure P13 as PG0 channel
P14CFG = 0x13; // Configure P14 as PG1 channel
P15CFG = 0x14; // Configure P15 as PG2 channel
P16CFG = 0x15; // Configure P16 as PG3 channel
P17CFG = 0x16; // Configure P17 as PG4 channel
P22CFG = 0x17; // Configure P22 as PG5 channel
```

18.3 Enhanced PWM operation

18.3.1 Load update mode

There are two types of counter loading modes: one-shot load mode and automatic load mode. In one-shot load mode, the period and duty cycle related data are loaded into the counter once when it starts. In automatic load mode, the period and duty cycle related data are automatically reloaded at zero-crossing within the PWM cycle.

Due to the double-buffering structure of the PWM, changes made to the relevant operational registers (PWMPnL/PWMPnH/PWMDnL/PWMDnH/PWMDDnL/PWMDDnH) during PWM operation do not immediately reflect in the PWM output waveform. Instead, the values of these registers are only loaded into the corresponding buffers at zero-crossing. This structure ensures that any changes to the period or duty cycle data will not affect the output waveform of the current PWM cycle; the adjustments will take effect in the subsequent cycle. Thus, any modification of PWM-related data will not impact the current complete PWM cycle.

In high-speed applications, there might be instances where the loading point has arrived, but the operation of writing to the control registers has not yet completed. In such cases, we do not want a situation where part of the operational data has been loaded while another part has not. To address this high-speed application scenario, the PWM module provides a load enable bit.

After changing the relevant operational registers, it is necessary to set the load register PWMLOADEN enable bit PWMnLE to 1. Once the period and duty cycle loading is completed, the PWMnLE bit will automatically reset to 0. This can be used to determine whether the values of the associated registers have been loaded into the actual circuit. If PWMnLE=0, it indicates that the loading has been completed, and the currently outputted PWM waveform will be affected. If PWMnLE=1, it means that the loading has not yet occurred, so the current PWM waveform has not changed and will reflect the modified register values at the next loading point. If the relevant operational register values are changed again, PWMnLE must be reset to 1.

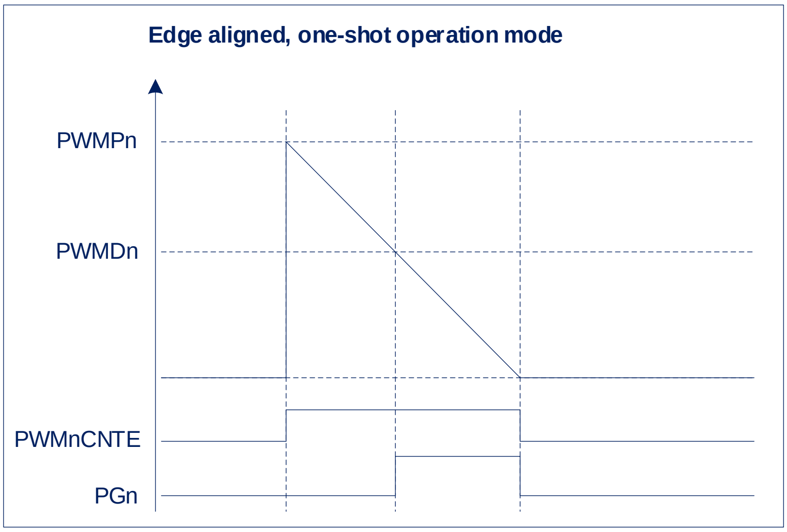
Note: When PWMnLE=1, changes to the period and duty cycle register contents may lead to unpredictable results.

It is recommended to first change the contents of the period and duty cycle registers, then set the load enable bit PWMnLE to 1, and finally wait for the loading to complete (PWMnLE=0).

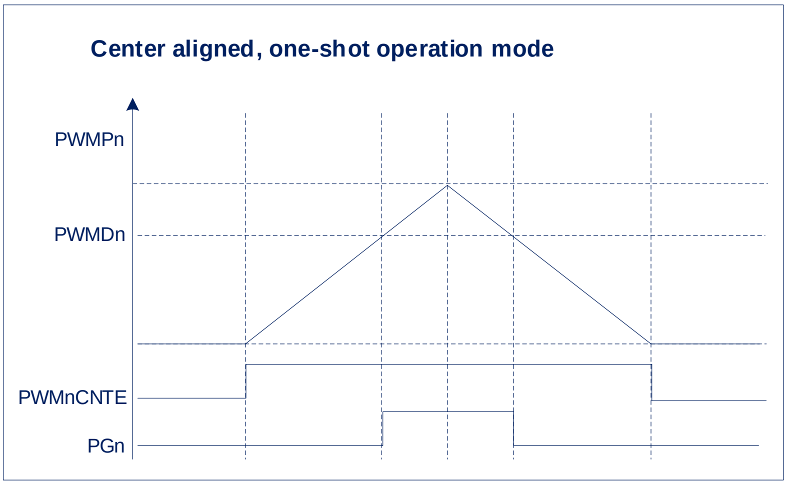
18.3.2 One-shot count mode

One-shot Count Mode is a mode in which the PWM counter operates only for one PWM cycle, and then the PWM counter stops operating. When the one-shot mode is completed, the PWM counter enable control bit is cleared by hardware (PWMnCNTE=0), and if one-shot mode is enabled again, the PWM counter enable control bit needs to be enabled (PWMnCNTE=1). The PWM counter mode can be selected via PWMCNTM.

The timing diagram for the one-shot mode with edge alignment is shown in the following figure.



The timing diagram for one-shot operation mode with center alignment is shown in the following figure.



18.3.3 Edge alignment mode

In edge-aligned mode, the PWM counter operates in a down-counting mode. The 16-bit PWM counter, CNTn, is initialized with the value from the period register PWMPn and begins counting down until it reaches 0. At this point, the MCU automatically loads the value from the period register into CNTn, starting the next PWM cycle.

When the value of CNTn equals the value in the duty cycle register PWMDn, the output PGn goes high. CNTn continues counting down to 0, at which point PGn will output low. If the PWM is configured for inverted output, the output levels will be the opposite of what is described above.

The parameters related to edge alignment are as follows:

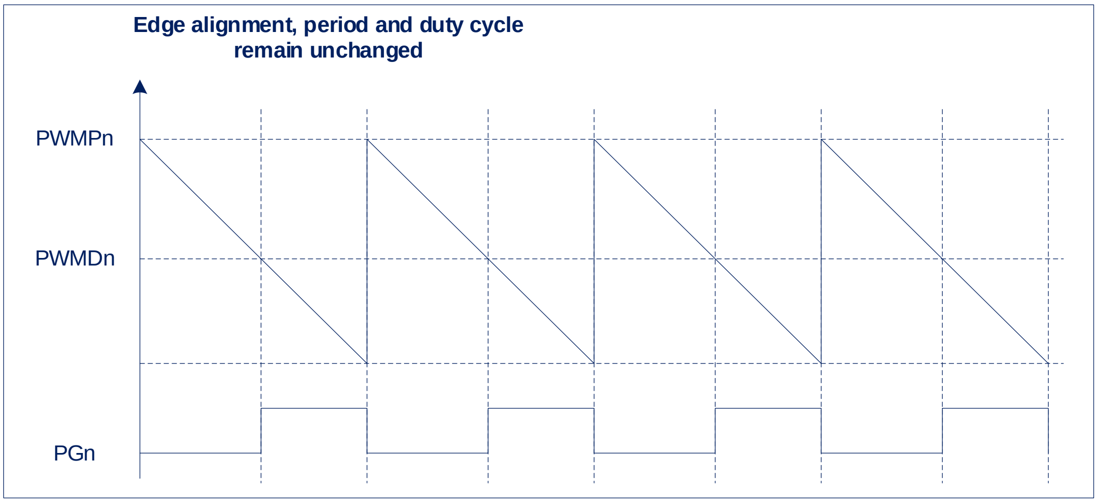
$$\text{High Level Time} = (\text{PWMDn+1})\times\text{Tpwm}$$

$$\text{Period} = (\text{PWMPn+1})\times\text{Tpwm}$$

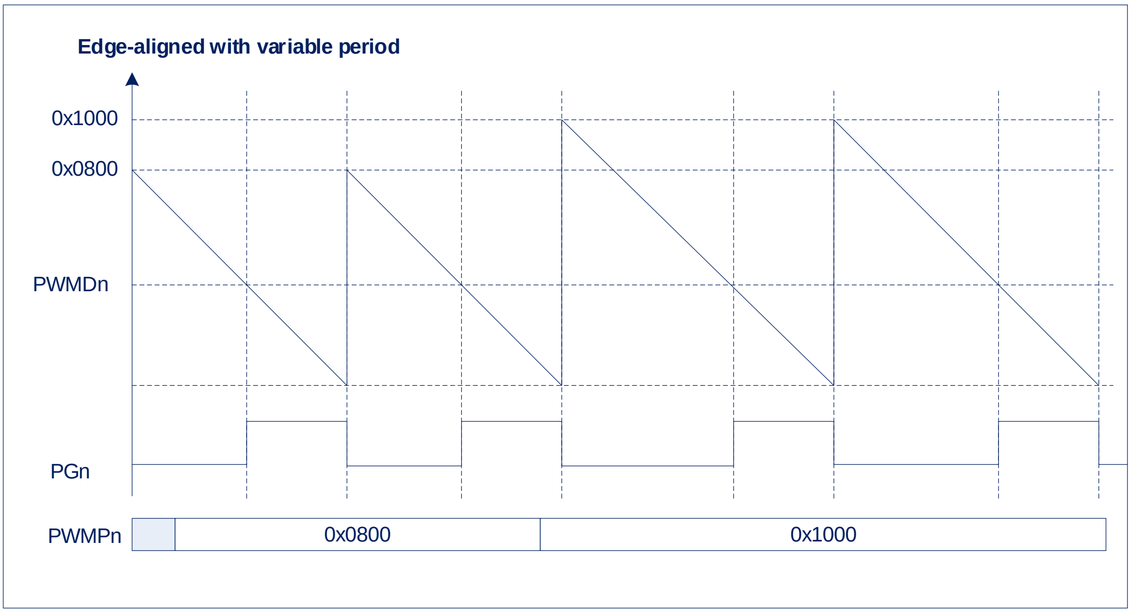
$$\text{Duty Cycle} = \frac{\text{PWMDn+1}}{\text{PWMPn+1}}$$

When PWMDn = 0, the duty cycle is 0%.

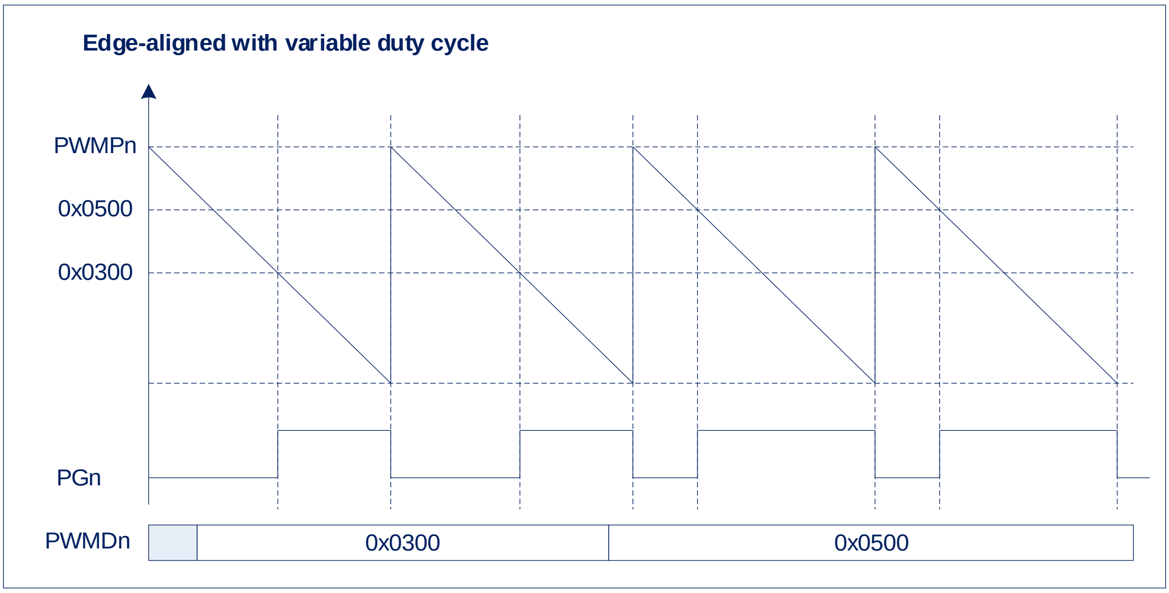
The timing diagram for edge-aligned PWM, where the period and duty cycle remain unchanged, is illustrated as follows:



Edge alignment, the cycle change timing diagram is shown below.



With the edges aligned, the duty cycle variation timing diagram is shown below.



18.3.4 Center-aligned mode

18.3.4.1 Symmetric counting

In center-aligned symmetric counting mode, the PWM counter operates in an up-down counting mode. The 16-bit PWM counter, CNTn, starts counting up from 0. When CNTn equals the value in the period register (PWMPn), it automatically begins counting down to 0. This counting process is repeated for subsequent PWM cycles.

During the up-counting edge, when the value of CNTn equals the value in the duty cycle register (PWMDn), the level of PGn toggles to high. During the down-counting edge, when CNTn equals the value in the duty cycle register, the output level of PGn toggles to low. If PWM is configured for inverted output, the output levels will be the opposite of what is described above.

The parameters related to symmetric counting are as follows:

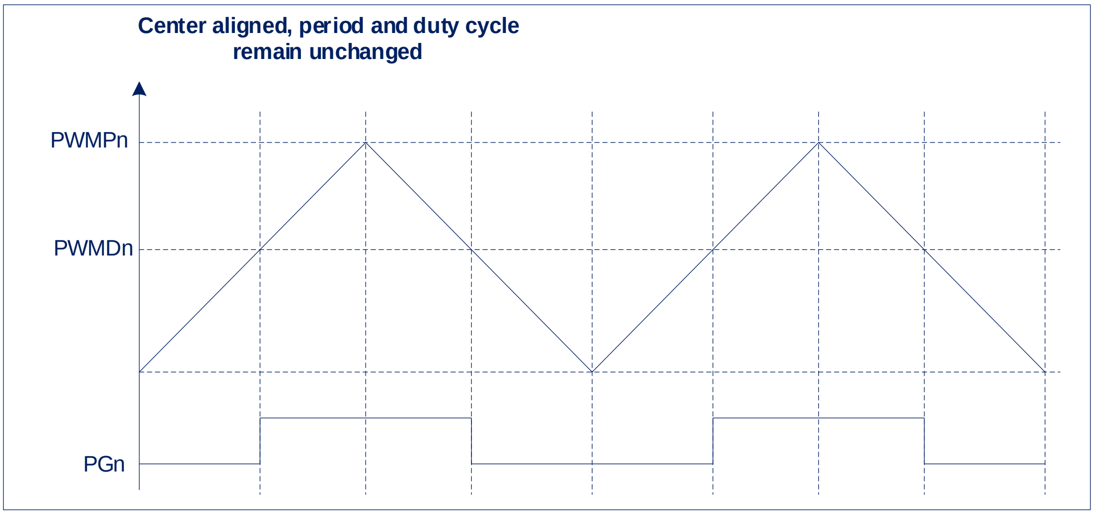
$$\text{High Level Time} = (\text{PWMPn}\times2\text{-PWMDn}\times2\text{-1})\times\text{Tpwm}$$

$$\text{Period} = (\text{PWMPn})\times2\times\text{Tpwm}$$

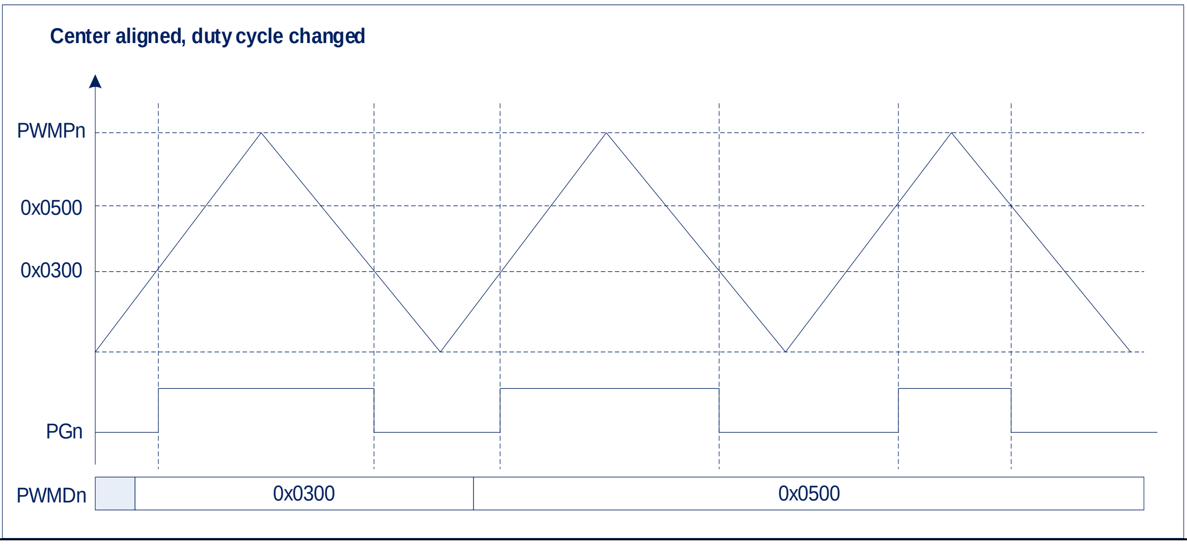
$$\text{Duty Cycle} = \frac{\text{PWMPn}\times2\text{-PWMDn}\times2\text{-1}}{\text{PWMPn}\times2}$$

When PWMDn = 0, the duty cycle is 100%.

The timing diagram for center-aligned PWM, where the period and duty cycle remain unchanged, is illustrated as follows:



The timing diagram for the center-aligned symmetric counting and duty cycle change is shown below.



18.3.4.2 Asymmetric counting

The center-aligned asymmetric PWM mode is a critical feature in motor control applications. In this mode, the PWM counter continues to operate in an up-down counting mode.

There are two comparison registers: PWMDn and PWMDDn. The 16-bit PWM counter CNTn starts counting up from 0. When CNTn equals the value in the PWMDn register, the output level of PGn toggles from low to high. The counter then continues counting up to the value in the PWMPn register before starting to count down. During the down-counting phase, when CNTn equals the value in the PWMDDn register, PGn toggles back to low. The counter continues counting down to 0. To enable the asymmetric PWM mode, the control bit ASYMEN must be set to 1.

The related parameters in asymmetric mode are as follows:

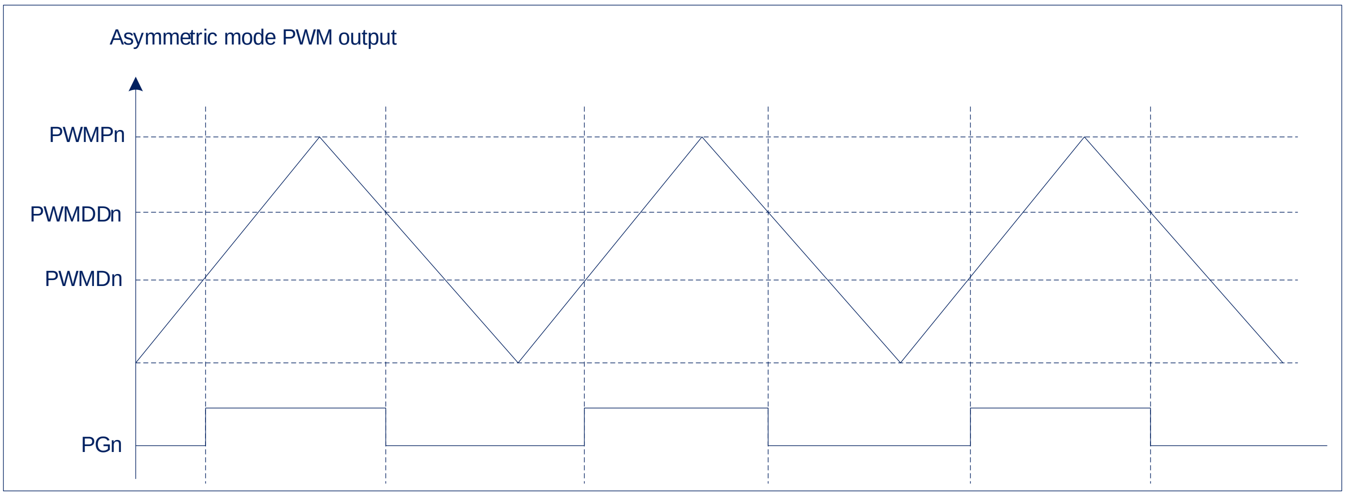
$$\text{High Level Time} = (\text{PWMPn}\times 2\text{-PWMDn-PWMDDn-1})\times\text{Tpwm}$$

$$\text{Period} = (\text{PWMPn})\times 2\times\text{Tpwm}$$

$$\text{Duty Cycle} = \frac{\text{PWMPn}\times 2\text{-PWMDn-PWMDDn-1}}{\text{PWMPn}\times 2}$$

When both PWMDn = 0 and PWMDDn = 0, the duty cycle is 100%.

The timing diagram for the center-aligned asymmetric PWM mode is illustrated below:

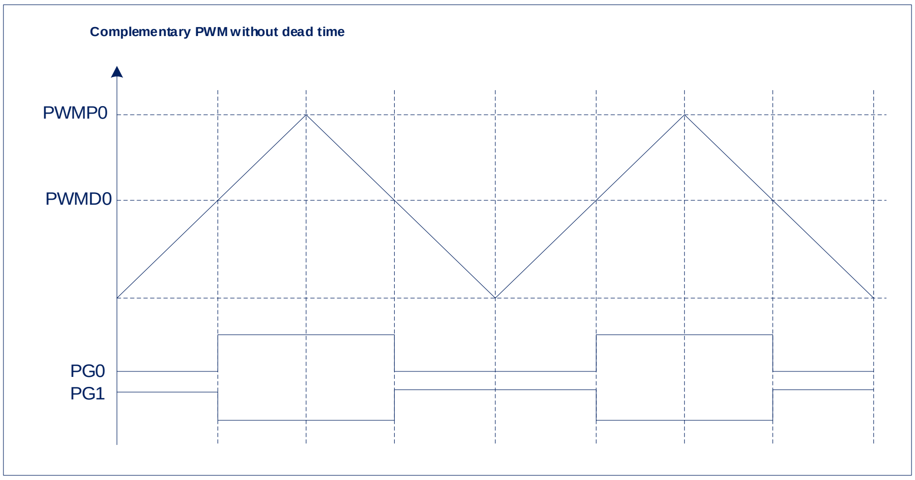


18.3.5 Complementary mode with dead time

In practical motor control applications, the PWM signals used to drive the inverter bridge must have a complementary output mode, meaning that the driving signal for the upper arm of the bridge is the exact inverse of the driving signal for the lower arm.

In the enhanced PWM module, six PWM channels can be configured into three pairs of complementary signals: PWM0 and PWM1, PWM2 and PWM3, PWM4 and PWM5. The periods and duty cycles of PWM1, PWM3, and PWM5 are determined by the corresponding registers of PWM0, PWM2, and PWM4, respectively.

The timing diagram for the complementary mode without dead time is shown below:



In motor control applications, the ideal PWM signal exhibits level transitions simultaneously. However, due to the delays associated with turning MOSFETs on and off, there is a risk of a direct short circuit between the power supply and ground, potentially damaging the power transistors. To avoid this phenomenon, PWM with dead time becomes especially important. In complementary mode, each pair of complementary PWM signals supports the insertion of dead time, defined as follows:

Dead time for PWM0/1: (PWM01DT+1)*T_{PWM0}

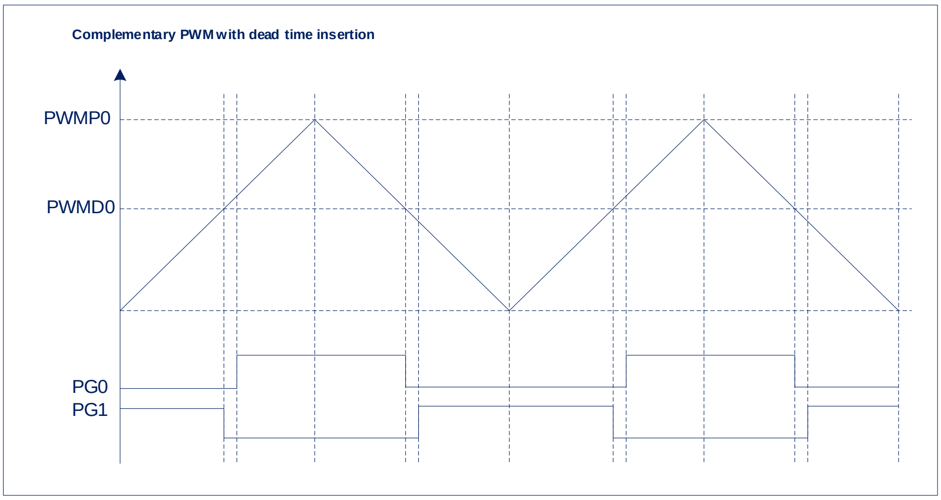
Dead time for PWM2/3: (PWM23DT+1)*T_{PWM2}

Dead time for PWM4/5: (PWM45DT+1)*T_{PWM4}

T_{PWM0}, T_{PWM2}, T_{PWM4} are the clock source periods for PWM0, PWM2, PWM4.

Note: Both center-aligned and edge-aligned configurations support complementary mode.

The complementary PWM waveform with inserted dead time is shown in the diagram below:



18.3.6 Braking function

The PWM braking signal sources can be triggered by several methods:

- Software trigger
- ADC result comparison output
- External trigger port (FB) (high/low level trigger)

The PWM braking control register PWMFBCK can be configured for software braking, while the external trigger port FB can enable PWM braking with the trigger type (high level or low level). The ADC comparator control register ADCMPC can be configured to control PWM braking enable based on the ADC comparator results.

PWM braking (fault protection) related flags:

- Fault flag PWMFBF (cleared by software)

  When a valid braking trigger signal is detected, the fault interrupt flag PWMFBF is set to 1 and must be cleared by software.
- Fault signal flag BRKAF (read-only)

  The fault signal flag BRKAF is set to 1; it is automatically cleared to 0 after the braking signal is removed. BRKAF is a read-only bit.
- Fault protection output status flag BRKOSF (read-only)

  If BRKOSF = 1, it indicates that the PWMn channel is outputting the PWMFBKD data state.

  If BRKOSF = 0, it indicates that PWMn is in a normal output state.

This flag shows whether the PWM output is in a braking state or a normal state. When a valid braking signal is detected, BRKOSF will be set to 1. In software recovery mode, executing the braking status clear operation (BRKCLR = 1) will affect the status of this bit.

The PWM braking recovery modes can be categorized into four types to adapt to different fault protection scenarios. The recovery conditions for the four braking recovery modes are described in the table below:

Braking recovery mode	Register PWMBRKC [1:0] setting mode	Counter status during braking	Recovery conditions				Recovery point
			Cancel the braking signal	Clear brake status	Counter enable	Delay	
Stop mode	00	Stop	Required	Required	Required	Not Required	Restart
Pause Mode	01	Continue counting	Required	Required	Not Required	Not Required	After clearing the brake status, the most recent loading point.
Recovery mode	10	Continue counting	Required	Not Required	Not Required	Not Required	The most recent loading point.
Delayed recovery	11	Continue counting	Required	Not Required	Not Required	Required	After the delay time elapses, the most recent loading point.

Note: After the brake protection is activated, the PWMn channel outputs data from PWMFBKD (each channel can be individually set to output high/low levels).

Stop mode: Generates a fault protection and fault interrupt flag, clears the PWMCNTE bit to zero, and stops the counter operation. To restore output, the braking signal must be removed, and a fault state clearing operation (PWMBRKC[3]=1) must be executed, followed by setting the PWMCNTE bit to 1 again.

Pause mode: Generates a fault protection and fault interrupt flag, but the counter continues to run. To restore output, the braking signal must be removed, and a fault state clearing operation (PWMBRKC[3]=1) must be executed, after which normal output will resume at the most recent loading update point.

Recovery mode: Generates a fault protection and fault interrupt flag, but the counter continues to run. After the braking signal is removed, normal output will automatically resume at the most recent loading update point. No fault state clearing operation is needed.

Delayed recovery mode: Generates a fault protection and fault interrupt flag, but the counter continues to run. After the braking signal is removed, normal output will resume at the most recent loading update point after a delay. No fault state clearing operation is needed.

The delay time can be freely set and controlled by the register {PWMBRKRDTH[1:0], PWMBRKRDTL[7:0]} (BRKRDT[9:0]). The delay time is defined as follows:

$$T_{delay} = BRKRDT[9:0]*T_{CLK} \ (T_{CLK} \text{ is the system clock cycle})$$

It is essential to distinguish between whether the brake signal is a pulse signal or a level signal: If the brake source is a level signal, the output cannot be restored until the brake signal is removed. If it is a pulse signal, the PWM output will resume from the most recent loading update point after the brake is triggered, unless another brake pulse signal is generated in the meantime.

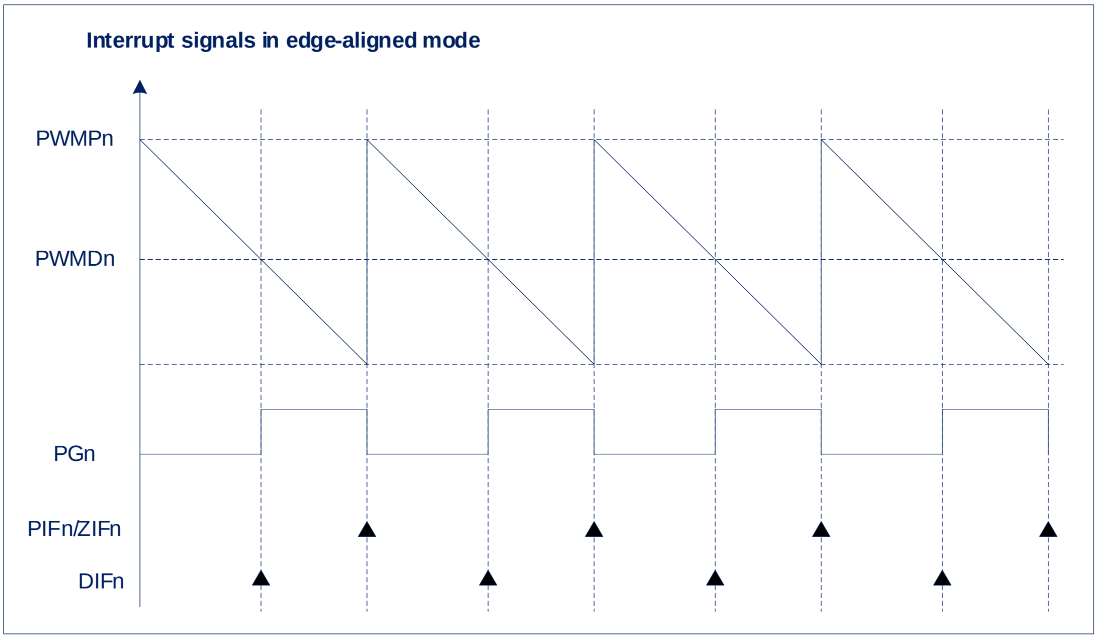
In the case of level braking, when the brake signal sets PWMFBF to 1, if the brake signal is not removed and the software writes PWMFBF to 0, then PWMFBF will remain 0 until the brake signal is removed and another brake signal is generated. To avoid clearing PWMFBF while the brake signal is still active, you can check the BRKAF bit to determine whether the brake signal has been removed.

18.3.7 Interrupt function

The enhanced PWM has a total of 25 interrupt flags, which include: 6 period interrupt flags, 6 zero-crossing interrupt flags, 6 up comparison interrupt flags, 6 down comparison interrupt flags, 1 brake interrupt flag. The generation of interrupt flags is independent of whether the corresponding interrupt enable bits are active. To enable any type of PWM interrupt, the global interrupt enable bit (EA=1) and the PWM global interrupt enable bit (PWMIE) must be activated to successfully configure the PWM interrupt functionality. All PWM interrupts share a single interrupt vector entry; therefore, upon entering the interrupt service routine, the user can determine which type of interrupt was triggered based on the interrupt flags.

The interrupt mechanism of the enhanced PWM is highly flexible. For the center-aligned mode, there are up to four types of interrupt: zero-crossing interrupt, up comparison interrupt, period interrupt, and down comparison interrupt. For the edge-aligned mode, there are three types of interrupts: period interrupt, comparison interrupt, and zero-crossing interrupt, where the period interrupt and zero-crossing interrupt are the same.

Below is the timing diagram of the interrupt signal for the edge-aligned mode:



18.4 PWM related registers

18.4.1 PWM control register (PWMCON)

F120H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCON	--	PWMRUN	PWMMODE1	PWMMODE0	GROUPEN	ASYMEN	CNTTYPE	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	--	Reserved, set to 0.
Bit6	PWMRUN:	PWM clock pre-scaler, clock division enable bit
	1=	Disable (PWMmnPSC and PWMmnDIV are both cleared to 0)
	0=	Enable
Bit5~Bit4	PWMMODE<1:0>:	PWM mode control bit
	00=	Independent mode
	01=	Complementary mode
	10=	Synchronous mode
	11=	Reserved
Bit3	GROUPEN:	PWM group function enable bit
	1=	PG0 controls PG2, PG4; PG1 controls PG3, PG5.
	0=	All PWM channel signals are independent of each other.
Bit2	ASYMEN:	Asymmetric counting enable bit in PWM center alignment mode
	1=	Asymmetric counting enable
	0=	Symmetric counting enable
Bit1	CNTTYPE:	PWM counting alignment mode selection bit
	1=	Center alignment mode
	0=	Edge alignment mode
Bit0	--	Reserved, set to 0.

18.4.2 PWM output enable control register (PWMOE)

F121H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMOE	--	--	PWM5OE	PWM4OE	PWM3OE	PWM2OE	PWM1OE	PWM0OE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6	--	Reserved, set to 0.
Bit5	PWM5OE:	PWM channel 5 output enable bit
	1=	Enable
	0=	Disable
Bit4	PWM4OE:	PWM channel 4 output enable bit
	1=	Enable
	0=	Disable
Bit3	PWM3OE:	PWM channel 3 output enable bit
	1=	Enable
	0=	Disable
Bit2	PWM2OE:	PWM channel 2 output enable bit
	1=	Enable
	0=	Disable
Bit1	PWM1OE:	PWM channel 1 output enable bit
	1=	Enable
	0=	Disable
Bit0	PWM0OE:	PWM channel 0 output enable bit
	1=	Enable
	0=	Disable

18.4.3 PWM0/1 clock prescaler control register (PWM01PSC)

F123H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM01PSC	PWM01PSC7	PWM01PSC6	PWM01PSC5	PWM01PSC4	PWM01PSC3	PWM01PSC2	PWM01PSC1	PWM01PSC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	PWM01PSC<7:0>:	PWM channel 0/1 prescaler control bit
	00=	The pre-scaler clock stops, and the counters for PWM0/1 stop.
	Others=	The system clock is divided by (PWM01PSC+1).

18.4.4 PWM2/3 clock prescaler control register (PWM23PSC)

F124H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM23PSC	PWM23PSC7	PWM23PSC6	PWM23PSC5	PWM23PSC4	PWM23PSC3	PWM23PSC2	PWM23PSC1	PWM23PSC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	PWM23PSC<7:0>:	PWM channel 2/3 prescaler control bit
	00=	The pre-scaler clock stops, and the counters for PWM2/3 stop.
	Others=	The system clock is divided by (PWM23PSC+1).

18.4.5 PWM4/5 clock prescaler control register (PWM45PSC)

F125H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM45PSC	PWM45PSC7	PWM45PSC6	PWM45PSC5	PWM45PSC4	PWM45PSC3	PWM45PSC2	PWM45PSC1	PWM45PSC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM45PSC<7:0>: PWM channel 4/5 prescaler control bit
00= The pre-scaler clock stops, and the counters for PWM4/5 stop.
Others= The system clock is divided by (PWM45PSC+1).

18.4.6 PWM clock prescaler control register (PWMnDIV(n=0-5))

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMnDIV	--	--	--	--	--	PWMnDIV2	PWMnDIV1	PWMnDIV0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register PWMnDIV(n=0-5) address: F12AH, F12BH, F12CH, F12DH, F12EH, F12FH.

Bit7~Bit3 -- Reserved, set to 0.
Bit2~Bit0 PWMnDIV<2:0>: PWM channel n clock prescaler control bit
000= Fpwmn-PSC/2
001= Fpwmn-PSC/4
010= Fpwmn-PSC/8
011= Fpwmn-PSC/16
100= Fpwmn-PSC
Others= Fsys (system clock)
(PSC is the pre-scaler clock)

18.4.7 PWM data load enable control register (PWMLOADEN)

F129H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMLOADEN	--	--	PWM5LE	PWM4LE	PWM3LE	PWM2LE	PWM1LE	PWM0LE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, set to 0.
Bit5~Bit0 PWMnLE: PWM channel n data load enable bit (n=0-5) (hardware clears after loading is completed). When PWMnLE=1, changes to the period and duty cycle registers may cause unpredictable results.
1= Enable the loading of period and duty cycle data (PERIODn, CMPn, CMPDn).
0= Writing 0 is invalid.

18.4.8 PWM output polarity control register (PWMPINV)

F122H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPINV	--	--	PWM5PINV	PWM4PINV	PWM3PINV	PWM2PINV	PWM1PINV	PWM0PINV
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, set to 0.

Bit5~Bit0 PWMnPINV: PWM channel n outputs polarity control bit (n=0-5)

1= Inverted output

0= Normal output

18.4.9 PWM counter mode control register (PWMCNTM)

F127H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCNTM	--	--	PWM5CNTM	PWM4CNTM	PWM3CNTM	PWM2CNTM	PWM1CNTM	PWM0CNTM
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, set to 0.

Bit5~Bit0 PWMnCNTM: PWM channel n counter mode control bit (n=0-5)

1= Auto load mode.

0= One-shot mode.

18.4.10 PWM counter enable control register (PWMCNTE)

F126H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCNTE	--	--	PWM5CNTE	PWM4CNTE	PWM3CNTE	PWM2CNTE	PWM1CNTE	PWM0CNTE
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, set to 0.

Bit5~Bit0 PWMnCNTE: PWM channel n counter enable control bit (n=0-5)

1= PWMn counter enabled (PWMn starts outputting)

0= PWMn counter stopped (writing 0 in software stops the counter and clears its value). (The brake triggers hardware to clear this bit to 0; hardware clears this bit to 0 upon completion of one-shot-mode operation)

18.4.11 PWM counter mode control register (PWMCNTCLR)

F128H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMCNTCLR	--	--	PWM5CNTCLR	PWM4CNTCLR	PWM3CNTCLR	PWM2CNTCLR	PWM1CNTCLR	PWM0CNTCLR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, set to 0.

Bit5~Bit0 PWMnCNTCLR: PWM channel n channel n counter clear control bit (n=0-5) (hardware automatically clears to 0)

1= PWMn counter clear

0= Writing 0 is invalid

18.4.12 PWM period data register low 8 bits (PWMPnL (n=0-5))

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPnL	PWMPnL7	PWMPnL6	PWMPnL5	PWMPnL4	PWMPnL3	PWMPnL2	PWMPnL1	PWMPnL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register PWMPnL (n=0-5) address: F130H, F132H, F134H, F136H, F138H, F13AH.

Bit7~Bit0 PWMPnL<7:0>: PWM channel n period data register low 8 bits.

18.4.13 PWM period data register high 8 bits (PWMPnH (n=0-5))

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPnH	PWMPnH7	PWMPnH6	PWMPnH5	PWMPnH4	PWMPnH3	PWMPnH2	PWMPnH1	PWMPnH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register PWMPnH (n=0-5) address: F131H, F133H, F135H, F137H, F139H, F13BH.

Bit7~Bit0 PWMPnH<7:0>: PWM channel n period data register high 8 bits.

18.4.14 PWM compare data register low 8 bits (PWMDnL (n=0-5))

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDnL	PWMDnL7	PWMDnL6	PWMDnL5	PWMDnL4	PWMDnL3	PWMDnL2	PWMDnL1	PWMDnL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register PWMDnL (n=0-5) address: F140H, F142H, F144H, F146H, F148H, F14AH.

Bit7~Bit0 PWMDnL<7:0>: PWM channel n comparasion data (duty cycle data) register low 8 bits.

18.4.15 PWM compare data register high 8 bits (PWMDnH (n=0-5))

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDnH	PWMDnH7	PWMDnH6	PWMDnH5	PWMDnH4	PWMDnH3	PWMDnH2	PWMDnH1	PWMDnH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register PWMDnH (n=0-5) address: F141H, F143H, F145H, F147H, F149H, F14BH.

Bit7~Bit0 PWMDnH<7:0>: PWM channel n comparasion data (duty cycle data) register high 8 bits.

18.4.16 PWM down compare data register low 8 bits (PWMDDnL (n=0-5))

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDDnL	PWMDDnL7	PWMDDnL6	PWMDDnL5	PWMDDnL4	PWMDDnL3	PWMDDnL2	PWMDDnL1	PWMDDnL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register PWMDDnL (n=0-5) address: F150H, F152H, F154H, F156H, F158H, F15AH.

Bit7~Bit0 PWMDDnL<7:0>: PWM channel n comparasion data (duty cycle data with asymmetric counting) register lower 8 bits.

18.4.17 PWM down compare data register high 8 bits (PWMDDnH (n=0-5))

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDDnH	PWMDDnH7	PWMDDnH6	PWMDDnH5	PWMDDnH4	PWMDDnH3	PWMDDnH2	PWMDDnH1	PWMDDnH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register PWMDDnH (n=0-5) address: F151H, F153H, F155H, F157H, F159H, F15BH.

Bit7~Bit0 PWMDDnH<7:0>: PWM channel n down compare data (duty cycle data with asymmetric counting) register lower 8 bits.

18.4.18 PWM dead time enable control register (PWMDTE)

F160H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDTE	--	--	--	--	--	PWM45DTE	PWM23DTE	PWM01DTE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit3 --: Reserved, set to 0.

Bit2 PWM45DTE: PWM4/5 channel dead time delay enable bit
1= Enable
0= Disable

Bit1 PWM23DTE: PWM2/3 channel dead time delay enable bit
1= Enable
0= Disable

Bit0 PWM01DTE: PWM0/1 channel dead time delay enable bit
1= Enable
0= Disable

18.4.19 PWM0/1 dead time delay data register (PWM01DT)

F161H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM01DT	PWM01DT7	PWM01DT6	PWM01DT5	PWM01DT4	PWM01DT3	PWM01DT2	PWM01DT1	PWM01DT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM01DT<7:0>: PWM channel 0/1 dead time delay data register.

18.4.20 PWM2/3 dead time delay data register (PWM23DT)

F162H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM23DT	PWM23DT7	PWM23DT6	PWM23DT5	PWM23DT4	PWM23DT3	PWM23DT2	PWM23DT1	PWM23DT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 PWM23DT<7:0>: PWM channel 2/3 dead time delay data register.

18.4.21 PWM4/5 dead time delay data register (PWM45DT)

F163H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWM45DT	PWM45DT7	PWM45DT6	PWM45DT5	PWM45DT4	PWM45DT3	PWM45DT2	PWM45DT1	PWM45DT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	PWM45DT<7:0>:	PWM channel 4/5 dead time delay data register.

18.4.22 PWM mask control register (PWMMASKE)

F164H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMMASKE	--	--	PWM5MASKE	PWM4MASKE	PWM3MASKE	PWM2MASKE	PWM1MASKE	PWM0MASKE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6	--	Reserved, set to 0.
Bit5~Bit0	PWMnMASKE:	PWM channel n mask control enable bit (n=0-5)
	1=	PWMn channel enable mask data output
	0=	PWMn channel disable mask data output (normal PWM waveform output)

18.4.23 PWM mask data register (PWMMASKD)

F165H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMMASKD	--	--	PWM5MASKD	PWM4MASKD	PWM3MASKD	PWM2MASKD	PWM1MASKD	PWM0MASKD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6	--	Reserved, set to 0.
Bit5~Bit0	PWMnMASKD:	PWM channel n mask data bit (n=0-5)
	1=	PWMn channel output high
	0=	PWMn channel output low

18.4.24 PWM brake control register (PWMFBKC)

F166H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMFBKC	PWMFBIE	PWMFBF	BRKAF	PWMFBKSW	PWMFBES	--	PWMFBEN	--
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 PWMFBIE: PWM brake interrupt mask bit
1= Enable interrupt
0= Disable interrupt
Bit6 PWMFBF: PWM brake flag bit (cleared by writing 0)
1= Brake operation occurred (value from PWM output brake data register)
0= No brake operation occurred
Bit5 BRKAF: EPWM fault signal flag bit (read-only)
1= Fault signal or brake signal remains valid
0= No fault occurred
Bit4 PWMFBKSW: PWM software brake signal start bit
1= PWM generates a software brake signal
0= Disable
Bit3 PWMFBES: PWM external hardware brake channel (FB) trigger level selection bit
1= High level
0= Low level
Bit2 --: Reserved, set to 0.
Bit1 PWMFBEN: PWM external hardware brake channel (FB) enable bit
1= Enable
0= Disable
Bit0 --: Reserved, set to 0.

18.4.25 PWM brake data register (PWMFBKD)

F167H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMFBKD	--	--	PWM5FBKD	PWM4FBKD	PWM3FBKD	PWM2FBKD	PWM1FBKD	PWM0FBKD
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 --: Reserved, set to 0.
Bit5~Bit0 PWMnFBKD: PWM channel n brake data bit (n=0-5)
1= PWMn channel outputs high after generating a brake operation
0= PWMn channel outputs low after generating a brake operation

18.4.26 PWM brake recovery control register (PWMBRKC)

F15CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMBRKC	BRKOSF	BRKRCS2	BRKRCS21	BRKRCS20	BRKCLR	BRKEN	BRKMS1	BRKMS0
R/W	R	R/W	R/W	R/W	W	R/W	R/W	R/W
Reset value	0	0	0	0	0	1	0	0

Bit7	BRKOSF:	EPWM fault protection output status flag bit (read-only)
	0=	EPWMn channel is in normal output state
	1=	EPWMn channel is in the data state of outputting BRKODn
Bit6~Bit4	BRKRCS<2:0>:	EPWM fault recovery load point selection bit
	000=	Recovery at EPWM0 load point
	001=	Recovery at EPWM1 load point
	010=	Recovery at EPWM2 load point
	011=	Recovery at EPWM3 load point
	100=	Recovery at EPWM4 load point
	101=	Recovery at EPWM5 load point
	Others=	Reserved
Bit3	BRKCLR:	EPWM fault protection clear bit (write-only)
	0=	Invalid
	1=	Clear fault protection status Note: Write 1 for fault clearing operation only when BRKAF=0; otherwise, the operation is invalid.
Bit2	BRKEN:	EPWM fault protection enable bit
	0=	Disable
	1=	Enable
Bit1~Bit0	BRKMS<1:0>:	Fault protection mode selection bit
	00=	Stop mode
	01=	Pause mode
	10=	Recovery mode
	11=	Delayed recovery mode

18.4.27 PWM delay recovery low 8 bits data register (PWMBRKRDTL)

F15DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMBRKRDTL	BRKRDT7	BRKRDT6	BRKRDT5	BRKRDT4	BRKRDT3	BRKRDT2	BRKRDT1	BRKRDT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	BRKRDT <7:0>:	Fault protection recovery delay data low 8 bits (valid only in delay recovery mode)

18.4.28 delay recovery high 2 bits data register (PWMBRKRDTH)

F15EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMBRKRDTH	-	-	-	-	-	-	BRKRDT9	BRKRDT8
R/W	R	R	R	R	R	R	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	BRKRDT <9:8>:	Fault protection recovery delay data high 2 bits (valid only in delay recovery mode) Delay time = BRKRDT[9:0] ×T_{CLK}

18.5 PWM interrupt related registers

18.5.1 Interrupt mask register (EIE2)

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI interrupt enable bit
1= Enable SPI interrupt
0= Disable SPI interrupt

Bit6 I2CIE: I^{2}C interrupt enable bit
1= Enable I^{2}C interrupt
0= Disable I^{2}C interrupt

Bit5 WDTIE: WDT interrupt enable bit
1= Enable WDT overflow interrupt
0= Disable WDT overflow interrupt

Bit4 ADCIE: ADC interrupt enable bit
1= Enable ADC interrupt
0= Disable ADC interrupt

Bit3 PWMIE: PWM global interrupt enable bit
1= Enable all PWM interrupts
0= Disable all PWM interrupts

Bit2 --: Reserved, set to 0.

Bit1 ET4: Timer4 interrupt enable bit
1= Enable Timer4 interrupt
0= Disable Timer4 interrupt

Bit0 ET3: Timer3 interrupt enable bit
1= Enable Timer3 interrupt
0= Disable Timer3 interrupt

18.5.2 Interrupt priority control register (EIP2)

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI interrupt priority control bit
1= High priority level
0= Low priority level
Bit6 PI2C: I^{2}C interrupt priority control bit
1= High priority level
0= Low priority level
Bit5 PWDT: WDT interrupt priority control bit
1= High priority level
0= Low priority level
Bit4 PADC: ADC interrupt priority control bit
1= High priority level
0= Low priority level
Bit3 PPWM: PWM interrupt priority control bit
1= High priority level
0= Low priority level
Bit2 --: Reserved, set to 0.
Bit1 PT4: TIMER4 interrupt priority control bit
1= High priority level
0= Low priority level
Bit0 PT3: TIMER3 interrupt priority control bit
1= High priority level
0= Low priority level

18.5.3 PWM period interrupt mask register (PWMPIE)

F168H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPIE	--	--	PWM5PIE	PWM4PIE	PWM3PIE	PWM2PIE	PWM1PIE	PWM0PIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 --: Reserved, set to 0.
Bit5~Bit0 PWMnPIE: PWM channel n period interrupt mask bit (n=0-5)
1= Enable interrupt
0= Disable interrupt

18.5.4 PWM zero-crossing interrupt mask register (PWMZIE)

F169H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMZIE	--	--	PWM5ZIE	PWM4ZIE	PWM3ZIE	PWM2ZIE	PWM1ZIE	PWM0ZIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, set to 0.
Bit5~Bit0 PWMnZIE: PWM channl n zero-crossing interrupt mask bit (n=0-5)
1= Enable interrupt
0= Disable interrupt

18.5.5 PWM up compare interrupt mask register (PWMUIE)

F16AH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMUIE	--	--	PWM5UIE	PWM4UIE	PWM3UIE	PWM2UIE	PWM1UIE	PWM0UIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, set to 0.
Bit5~Bit0 PWMnUIE: PWM channel n up compare interrupt mask bit (n=0-5)
1= Enable interrupt
0= Disable interrupt

18.5.6 PWM down compare interrupt mask register (PWMDIE)

F16BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDIE	--	--	PWM5DIE	PWM4DIE	PWM3DIE	PWM2DIE	PWM1DIE	PWM0DIE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, set to 0.
Bit5~Bit0 PWMnDIE: PWM channel n down compare interrupt mask bit (n=0-5)
1= Enable interrupt
0= Disable interrupt

18.5.7 PWM period interrupt flag register (PWMPIF)

F16CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMPIF	--	--	PWM5PIF	PWM4PIF	PWM3PIF	PWM2PIF	PWM1PIF	PWM0PIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, set to 0.
Bit5~Bit0 PWMnPIF: PWM channel n period interrupt flag bit (n=0-5)
1= Interrupt generated (software clear)
0= No interrupt generated

18.5.8 PWM zero-crossing interrupt flag register (PWMZIF)

F16DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMZIF	--	--	PWM5ZIF	PWM4ZIF	PWM3ZIF	PWM2ZIF	PWM1ZIF	PWM0ZIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6	--	Reserved, set to 0.
Bit5~Bit0	PWMnZIF:	PWM channel n zero-crossing interrupt flag bit (n=0-5)
	1=	Interrupt generated (software clear)
	0=	No interrupt generated

18.5.9 PWM up compare interrupt flag register (PWMUIF)

F16EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMUIF	--	--	PWM5UIF	PWM4UIF	PWM3UIF	PWM2UIF	PWM1UIF	PWM0UIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6	--	Reserved, set to 0.
Bit5~Bit0	PWMnUIF:	PWM channel n up compare interrupt flag bit (n=0-5)
	1=	Interrupt generated (software clear)
	0=	No interrupt generated

18.5.10 PWM down compare interrupt flag register (PWMDIF)

F16FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMDIF	--	--	PWM5DIF	PWM4DIF	PWM3DIF	PWM2DIF	PWM1DIF	PWM0DIF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6	--	Reserved, set to 0.
Bit5~Bit0	PWMnDIF:	PWM channel n down compare interrupt flag bit (n=0-5)
	1=	Interrupt generated (software clear)
	0=	No interrupt generated

19. SPI Module

19.1 Overview

This SPI is a fully configurable SPI master/slave device, allowing users to configure the polarity and phase of the serial clock signal (SCLK). The serial clock line (SCLK) is synchronized with the shifting and sampling of information on two independent serial data lines, enabling simultaneous sending and receiving of SPI data. SPI allows the MCU to communicate with serial peripheral devices and also facilitates inter-processor communication in multi-master systems. It is a technology-independent design that can be implemented in various process technologies.

The SPI system is flexible enough to directly connect to many standard peripheral products from multiple manufacturers. To accommodate most available synchronous serial peripherals, the clock control logic allows for the selection of clock polarity and phase. The system can be configured as either a master device or a slave device. When configured as a master device, software selects one of eight different bit rates for the serial clock, with rates up to the system clock divided by 4 (Fsys/4).

The SPI slave chip select is used to address SPI slave devices for exchanging serial data. When SPI operates as a master device, it automatically drives the Slave Select Control Register (SSCR) selected by the slave. The SPI controller includes logic error detection to support inter-processor communication, such as a write collision detector that indicates when data is written to the serial shift register during transmission.

The features of SPI include:

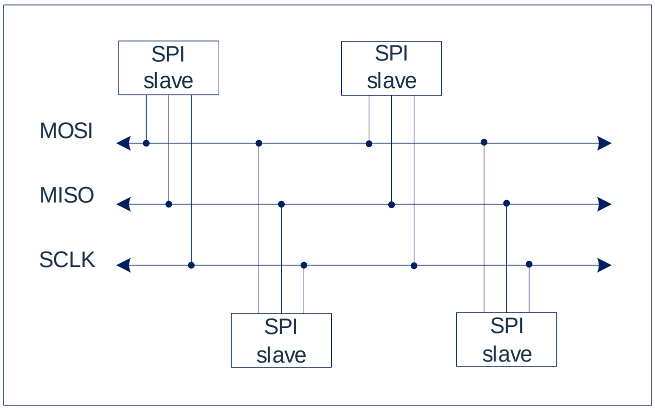
- Full-duplex synchronous serial data transfer.
- Support for master/slave modes.
- Support for multi-master systems.
- System error detection.
- Generation of interrupts.
- Supports speeds up to 1/4 of the system clock (FSYS ≤ 24 MHz).
- Bit rates can generate 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256, and 1/512 of the system clock.
- Supports four transfer formats.
- Simple interface allows for easy connection to microcontrollers.

19.2 SPI port configuration

To use the SPI functionality, the relevant ports need to be configured as SPI channels, and the corresponding input port must be selected through the communication input port register. For example, configure P14, P15, P16, and P17 as the SPI communication ports. The configuration code is as follows:

```
P14CFG = 0x0E;      //Configure P14 as NSS channel
P15CFG = 0x0F;      //Configure P15 as SCLK channel
P16CFG = 0x10;      //Configure P16 as MOSI channel
P17CFG = 0x11;      //Configure P17 as MISO channel
```

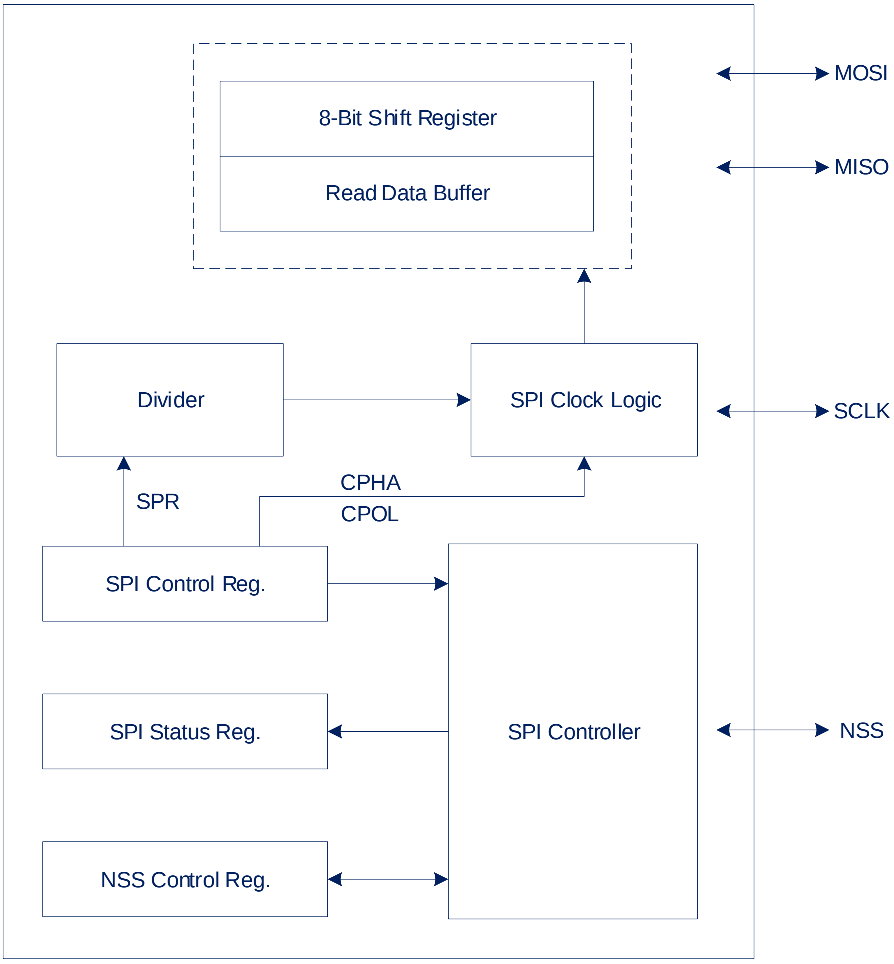
These ports are configured as SCLK, MOSI, MISO, and NSS, with the pull-up resistors and open-drain output switches forced to be disabled. The schematic diagram of the multi-master SPI communication structure is shown below:



19.3 SPI hardware description

During SPI transfer, as an 8-bit character is shifted out from one data pin, another data pin shifts in another 8-bit character. The 8-bit shift register in the master device and another 8-bit shift register in the slave device are connected as a circular 16-bit shift register. When a transfer occurs, this distributed shift register is shifted by 8 bits, effectively exchanging characters between the master and the slave.

The central element in the SPI system is a module that contains the shift register and a data read buffer. The system has a single buffer for the transmit direction and a double buffer for the receive direction. This means that new data can only be written to the shift register once the previous data transfer is completed; however, the received data is sent to a parallel read data buffer, allowing the shift register to freely accept a second serial character. As long as the first character is read from the read data buffer before the next serial character is ready for transfer, no data overlap will occur. The control block diagram of the SPI system is shown below:



The pins associated with SPI are: NSS, SCLK, MOSI, and MISO.

In master mode, the NSS output pin is used to select the slave device, while in slave mode, the NSS input pin is used to enable the transfer.

In master mode, the SCLK pin serves as the reference for the SPI clock signal. When the master device initiates a transfer, eight clock cycles are automatically generated on the SCLK pin.

When SPI is configured as a slave device, the SI pin serves as the input data line for the slave device, while the SO pin serves as the output data line for the slave device.

When SPI is configured as a master device, the MI pin serves as the input data line for the master device, while the MO pin serves as the output data line for the master device.

19.4 SPI related registers

19.4.1 SPI control register (SPCR)

0xEC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPCR	--	SPEN	SPR2	MSTR	CPOL	CPHA	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	1	0	0

Bit7	--	Reserved, set to 0.
Bit6	SPEN:	SPI module enable bit
	1=	Enable
	0=	Disable
Bit5	SPR2:	SPI clock frequency select bit: bit [2]
Bit4	MSTR:	SPI mode select bit
	1=	Master mode
	0=	Slave mode
Bit3	CPOL:	SPI clock polarity select bi
	1=	SCLK idle high
	0=	SCLK idle low
Bit2	CPHA:	SPI clock phase select bit
Bit1~Bit0	SPR<1:0>:	SPI clock frequency select bits [1:0] (for frequency control, see the table below)

SPR2-SPR0 control SPI clock divider

SPR2	SPR1	SPR0	System clock division
0	0	0	4
0	0	1	8
0	1	0	16
0	1	1	32
1	0	0	64
1	0	1	128
1	1	0	256
1	1	1	512

19.4.2 SPI data register (SPDR)

0xEE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPDR	SPIDATA7	SPIDATA6	SPIDATA5	SPIDATA4	SPIDATA3	SPIDATA2	SPIDATA1	SPIDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	SPIDATA:	SPI data transmission and reception
	Write:	Write the data to be transmitted (the transmission order is from the most significant bit to the least significant bit)
	Read:	Data already received

19.4.3 SPI slave select control register (SSCR)

The Slave Select Control Register (SSCR) can be read or written at any time. It is used to configure which slave select output should be driven when acknowledging an SPI master transfer. When an SPI master transfer is initiated, the contents of the SSCR register will be automatically assigned to the NSS pin.

0xEF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SSCR	--	--	--	--	--	--	--	NSSO0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	1	1	1	1	1	1	1

Bit7~Bit1	--	Reserved, set to 1.
Bit0	NSSO0:	SPI slave select control bit (master chip select output NSS is NSSO0)
	0=	When the SPI master transfer is initiated, NSSO0 outputs 0.
	1=	When the SPI master transfer is initiated, NSSO0 outputs 1.

19.4.4 SPI status register (SPSR)

0xED	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSR	SPISIF	WCOL	--	--	--	--	--	SSCEN
R/W	R	R	--	R	--	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	SPISIF:	SPI transfer complete interrupt flag (read-only)
	1=	SPI transfer is completed (read SPSR first, then read/write SPDR to clear)
	0=	SPI transfer is not yet completed
Bit6	WCOL:	SPI write collision interrupt flag (read-only)
	1=	Write SPDR operation collision: occurs when the SPI transfer is incompleted (read SPSR first, then read/write SPDR to clear).
	0=	No write collision
Bit5~Bit1	--	Reserved, set to 0.
Bit0	SSCEN:	SPI master mode NSS output control bit
	1=	SPI is in idle state, NSS outputs a high level.
	0=	NSS output register SSCR contents

SPI Status Register (SPSR) contains flags indicating the completion of transfer or the occurrence of system errors. When the corresponding events occur and are cleared sequentially by software, all flags will be automatically set. By reading SPSR, followed by accessing SPDR, the flags SPISIF and WCOL will be automatically cleared.

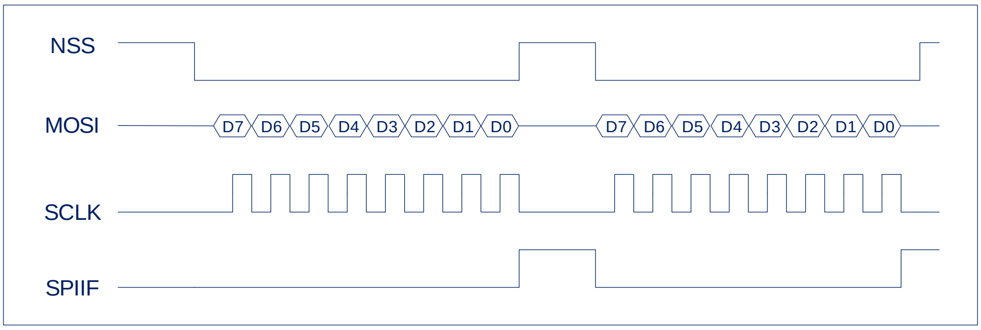
The SSCEN bit is the enable bit for automatic slave select output. When SSCEN is set to 1, the NSS line outputs the contents of the SSCR register during transfer, and the NSS is high when the transfer is idle. When the SSCEN bit is cleared to 0, the NSS line always displays the contents of the SSCR register.

19.5 SPI master mode

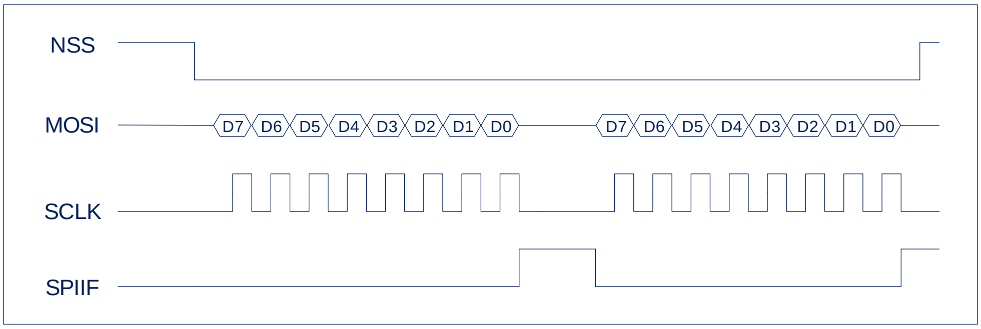
When the SPI is configured in master mode, transfer is initiated by writing to the SPDR register. When a new byte is written to the SPDR register, the SPI begins the transfer. The serial clock (SCLK) is generated by the SPI and is output when SPI is enabled in master mode.

In master mode, the SPI can select slave devices through the NSS line. The NSS line, which is the slave select output line, loads the contents of the SSCR register. The SSCEN bit in the SPSR register selects between automatic NSS line control and software control. When SSCEN is set in master mode, if SSCEN is set to 1, the NSS line outputs the contents of the SSCR register while transfer is in progress, and NSS is high when the transfer is idle. When SSCEN is cleared to 0, the NSS line is controlled by software and will always display the contents of the SSCR register, regardless of whether transfer is in progress or the SPI is idle.

When SSCEN = 1, the SPI's clock polarity (CPOL) is configured to 0, and the clock phase (CPHA) is also set to 0. The slave select line is used as shown in the diagram below:



When SSCEN = 0, the SPI's clock polarity (CPOL) is configured to 0, and the clock phase (CPHA) is also set to 0. The slave select line is used as shown in the diagram below:



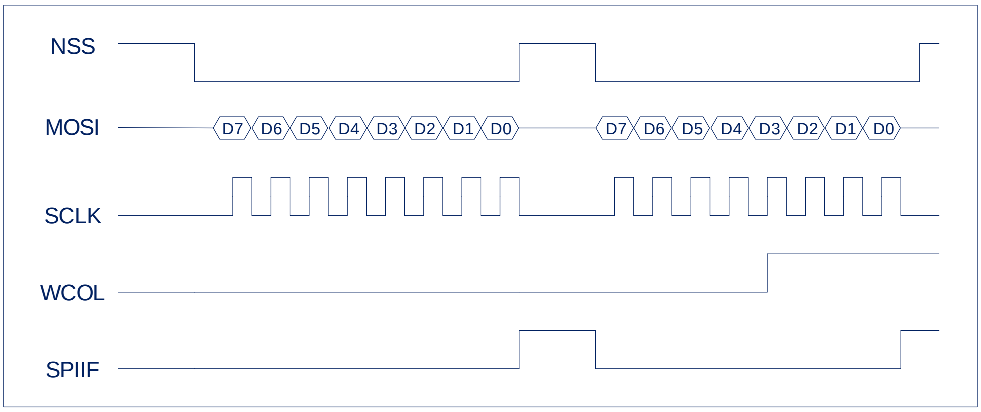
19.5.1 Write collision error

A write collision occurs if the SPI data register is written to during an ongoing transfer. The transfer continues uninterrupted, resulting in erroneous write data not being shifted into the shift register. Write collisions are indicated by the WCOL flag in the SPSR register.

When a WCOL error occurs, the WCOL flag is automatically set to 1 by the hardware. To clear the WCOL bit, the user should follow these steps:

- Read the contents of the SPSR register.
- Access the SPDR register (read or write).

In SPI master mode, when the SPI’s clock polarity (CPOL) is set to 0 and the clock phase (CPHA) is also set to 0, the write collision error is illustrated in the diagram below:



The specific conditions for a write collision to occur are as follows: during data transfer, when NSS is low, if SPDR is written to from the time the first data begins to be sent until the falling edge of the 8th SCLK, a write collision will occur, and WCOL will be set to 1.

Caution: At the start of data transmission, NSS does not immediately go low after writing to SPDR; it may take up to one SPI clock cycle before NSS goes low. Once NSS is low, a system clock delay is required before the first data is actually transmitted. Only after this delay does the transmission enter the actual data transfer state. Writing to SPDR again during the period between writing to SPDR and entering the actual data transfer state will not cause a write collision; however, this operation will update the data that is ready to be sent. If multiple writes to SPDR are performed, the data sent will be the value from the last write to SPDR.

Since SPI has only one transmission buffer, it is recommended to check whether the previous data transmission is completed before writing to SPDR. This ensures that the transmission is completed before updating the SPDR register to prevent write collisions.

19.6 SPI slave mode

When configured as an SPI slave device, SPI transfer is initiated by an external SPI master module using the SPI slave select input and generates the SCLK serial clock.

Before the transfer begins, it is necessary to determine which SPI slave will be used for data exchange. The NSS is used (active low = 0); the clock signal connected to the SCLK line will cause the SPI slave device to transfer the contents of its receive shift register to the MOSI line and drive the MISO line with the contents of the transmit shift register. When all 8 bits have been shifted in/out, SPI generates an interrupt request by setting the IRQ output. The contents of the shift register drive the MISO line.

In SPI slave mode, there can only be one transfer error-write collision error.

19.6.1 Addressing error

In slave mode, only write collision errors can be detected by the SPI.

When a write operation to the SPDR register is executed during SPI transfer, a write collision error will occur.

In slave mode, when CPHA is cleared, a write collision error may occur as long as the NSS slave select line is driven low, even if all bits have been transmitted. This is because the start of the transfer is not explicitly indicated, and the driving low of NSS after a full byte transfer may indicate the beginning of the next byte transfer.

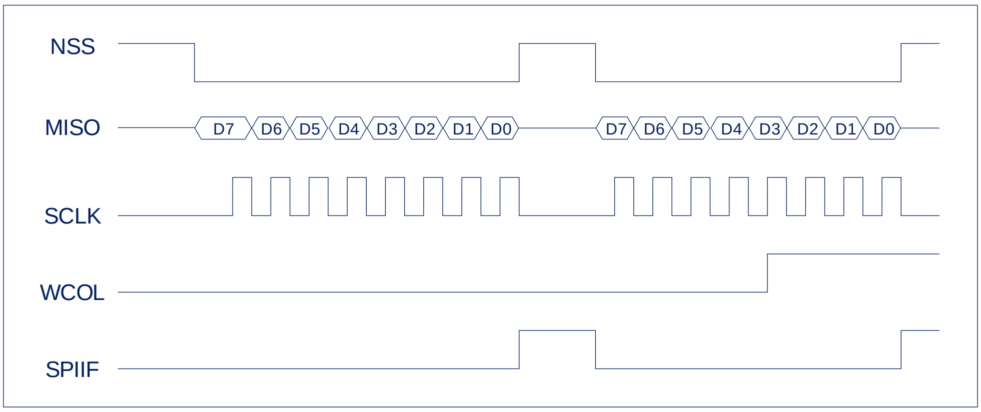
19.6.2 Write collision error

If the SPI Data Register is written to during the transfer, a write collision will occur. The transfer continues uninterrupted, and the erroneous written data will not be shifted into the shift register. The write collision is indicated by the WCOL flag in the SPSR register.

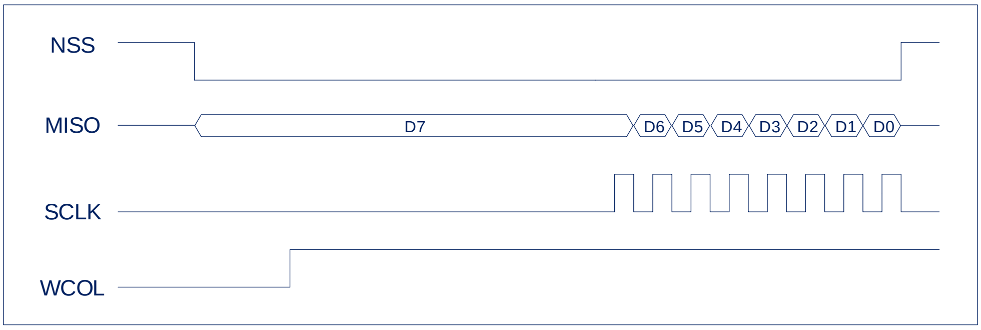
When a WCOL error occurs, the WCOL flag is automatically set to 1 by hardware. To clear the WCOL bit, the user should perform the following sequence:

- Read the contents of the SPSR register;
- Access the SPDR register (read or write).

The following diagram illustrates write collision errors during the transfer process in SPI slave mode:



In case CPHA is cleared, the generation of WCOL can also be triggered by writing to the SPDR register when any NSS line is driven low, even if the SPI master has not generated the serial clock SCLK. This is because the start of the transfer is not explicitly indicated, and driving NSS low after a full byte transfer may indicate the start of the next byte transfer. When the NSS transfer line is low, and the clock phase CPHA = 0, writing to the SPDR leads to a write collision error, as illustrated in the following diagram:



Additionally, in slave mode, after writing to the SPDR, the master-controlled NSS does not immediately go low. After NSS is driven low, it is necessary to wait for the second edge of SCLK before entering the actual data transfer state.

During the time between writing to SPDR and starting to send the first data, writing to SPDR again does not cause a write collision. However, this operation will update the data prepared for transmission. If there are multiple write operations to SPDR, the data sent will be the value of the last write to SPDR.

From the start of sending the first data until the second edge of SCLK, writing to SPDR again will not cause a write collision, nor will it update the data being transmitted. Therefore, this write to SPDR operation will be ignored.

Since SPI only has a single transmit buffer, it is recommended to check whether the previous data has been completely sent before writing to the SPDR register to avoid write collisions.

19.7 SPI clock control logic

19.7.1 SPI clock phase and polarity control

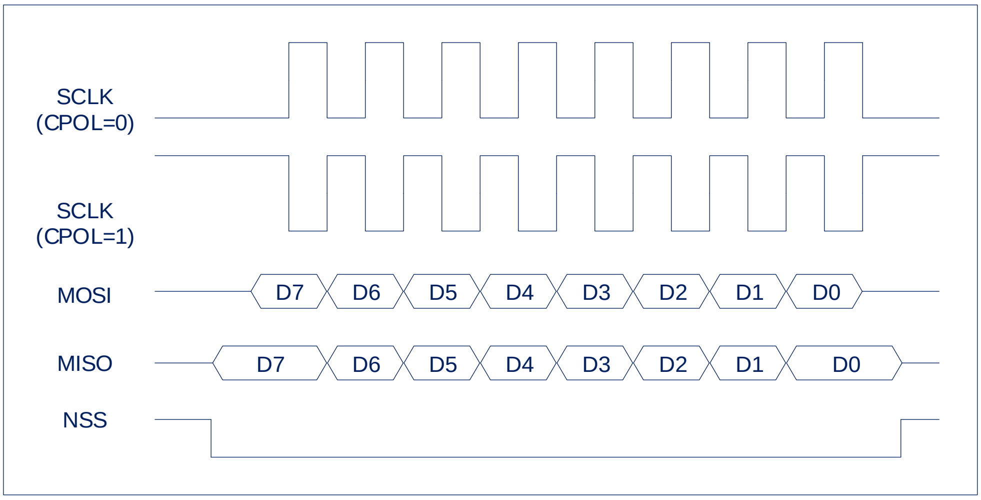
Software can select any of the four combinations of clock phase and polarity using two control bits in the SPI Control Register (SPCR). The clock polarity is specified by the CPOL control bit, which selects a high or low level during idle transfer. The clock phase (CPHA) control bit selects one of the two fundamentally different transfer formats. The clock phase and polarity of the master SPI device and the slave device must be the same. In some cases, the phase and polarity can be changed during transfer to allow the master device to communicate with slave peripherals that have different features. The flexibility of the SPI system allows for direct connections to almost all existing synchronous serial peripherals.

19.7.2 SPI transfer format

During SPI transfer, data is sent (serially shifted out) and received (serially shifted in) simultaneously. The serial clock line is synchronized with the two serial data lines for shifting and sampling. The slave select line allows for individual selection of slave SPI devices; unselected slave devices do not interfere with SPI bus activity. On the SPI master device, the slave select line can selectively indicate multi-master bus contention.

19.7.3 CPHA=0 transfer format

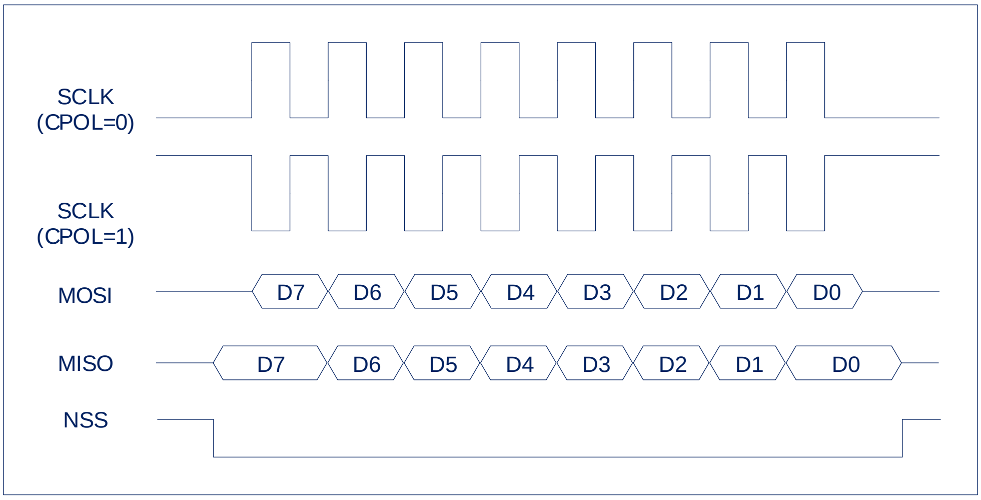
The diagram below shows the timing diagram for SPI transfer when CPHA is set to 0. The SCLK shows two waveforms: one for CPOL equal to 0 and the other for CPOL equal to 1. This diagram can be described as a timing diagram for either the master or slave device, with the Master In/Slave Out (MISO) and Master Out/Slave In (MOSI) pins directly connected between the master and slave. The MISO signal is the output from the slave, while the MOSI signal is the output from the master. The NSS line is the slave's select input; the master's NSS pin is not shown but is assumed to be invalid. The master's NSS pin must be at a high level. This timing diagram functionally describes how transfer is conducted and should not be used as a substitute for parameter information in data sheets.



When CPHA = 0, the NSS line must be released and then reset to 1 between each consecutive serial byte. Furthermore, if the NSS line is low and the slave writes data to the SPI Data Register (SPDR), a write collision error will occur. When CPHA = 1, the NSS line can remain low between consecutive transfers (it can always be held low). In systems with a single fixed master and a single slave driving the MISO data line, this format is sometimes preferred.

19.7.4 CPHA=1 transfer format

The diagram below shows the timing diagram for SPI transfer when CPHA = 1. The SCLK line displays two waveforms: one for CPOL = 0 and the other for CPOL = 1. Since the SCLK, MISO, and MOSI pins are directly connected between the master and slave, this diagram can be interpreted as a timing diagram for either the master or the slave device. The MISO signal is the output from the slave, while the MOSI signal is the output from the master. The NSS line serves as the slave select input; the master's NSS pin is not shown but is assumed to be invalid. The master's NSS pin must be held high or reconfigured to a general-purpose output that does not affect SPI communication.



19.8 SPI data transfer

19.8.1 SPI transfer start

All SPI transfers are initiated and controlled by the master SPI device. As a slave device, SPI considers the transfer to start on the first SCLK edge or the falling edge of NSS, depending on the selected CPHA format. When CPHA = 0, the falling edge of NSS indicates the start of the transfer. When CPHA = 1, the first edge of SCLK indicates the start of the transfer. In either CPHA mode, the transfer can be aborted by bringing the NSS line high, but this will reset the SPI slave logic and counters. The selected SCLK rate does not affect slave operation, as the host clock controls the transfer.

When the SPI is configured as a master, transfer is initiated by writing to the SPDR.

19.8.2 SPI transfer end

The SPI transfer is technically completed when the SPIF flag is set to 1. However, depending on the configuration of the SPI system, there may still be other tasks. Since the SPI bit rate does not affect the duration of the end period, discussions during the end period only consider the fastest rate. When the SPI is configured as a master, the SPIF is set at the end of the eighth SCLK cycle. When CPHA equals 1, SCLK remains inactive during the last half of the eighth SCLK cycle.

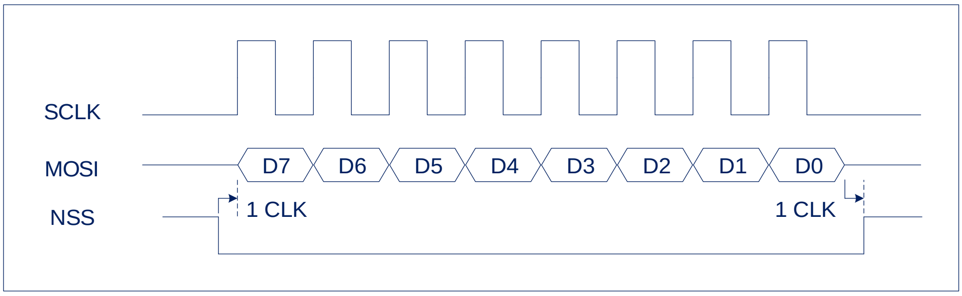
Because the SCLK line can be asynchronous with the slave’s MCU clock, and the slave cannot access as much information as the master can during the SCLK cycles, the end period behaves differently when SPI is operated in slave mode. For example, when CPHA = 1, the last edge of SCLK occurs in the middle of the eighth SCLK cycle, and the slave cannot know when the previous SCLK cycle ended. For these reasons, the slave assumes that the transfer is completed after sampling the last bit of serial data, which corresponds to the midpoint of the eighth SCLK cycle.

The SPIF flag is set at the end of the transfer, but when the NSS line is still low, the slave is not allowed to write new data to the SPDR.

19.9 SPI timing diagrams

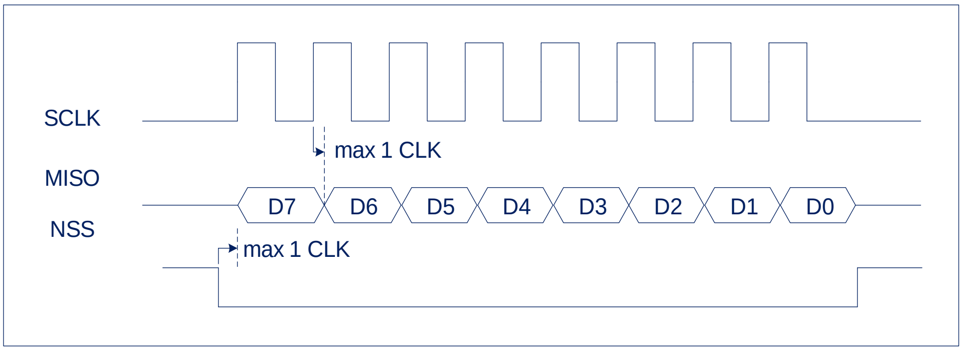
19.9.1 Master mode transfer

When the SPI clock polarity (CPOL) is 0 and the clock phase (CPHA) is 1, in master mode, after the NSS is low for one system clock (CLK), the MOSI starts to output. The data on the MOSI is output on the rising edge of the SCLK clock. The timing diagram for master mode is shown in the figure below:



19.9.2 Slave mode transfer

When the SPI clock polarity (CPOL) is 0 and the clock phase (CPHA) is 1, the data on the MISO line begins to output after the falling edge of the NSS line. The output of the MISO data may differ from the falling edge of the NSS by a maximum of one system clock (CLK). The timing diagram for slave mode is shown in the figure below:



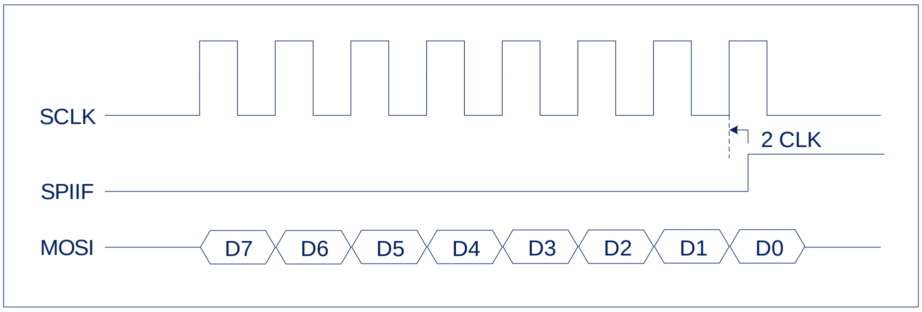
19.10 SPI interrupt

The interrupt number for SPI is 22, with the interrupt vector at 0x00B3. To enable the SPI interrupt, the enable bit SPIIE must be set to 1, and the global interrupt enable bit EA must also be set to 1.

If all the SPI-related interrupts are enabled and the SPI global interrupt flag SPIIF is set to 1, the CPU will enter the interrupt service routine. The SPIIF flag is read-only and is independent of the status of SPIIE.

In the SPI status register (SPSR), when either the transfer complete flag (SPISIF) or the write collision flag (WCOL) is set to 1, the SPI global interrupt flag (SPIIF) will be set to 1. The SPIIF flag will automatically clear to 0 only when all three flags are 0.

When the SPI clock polarity (CPOL) is 0 and the clock phase (CPHA) is 1, in master mode, the SPIIF flag is generated two system clocks (CLK) after the rising edge of the eighth SCLK clock of each data frame. The timing diagram is shown in the figure below:



19.10.1 Interrupt mask register (EIE2)

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI interrupt enable bit
1= Enable SPI interrupt
0= Disable SPI interrupt

Bit6 I2CIE: I^{2}C interrupt enable bit
1= Enable I^{2}C interrupt
0= Disable I^{2}C interrupt

Bit5 WDTIE: WDT interrupt enable bit
1= Enable WDT overflow interrupt
0= Disable WDT overflow interrupt

Bit4 ADCIE ADC interrupt enable bit
1= Enable ADC interrupt
0= Disable ADC interrupt

Bit3 PWMIE: PWM global interrupt enable bit
1= Enable all PWM interrupts
0= Disable all PWM interrupts

Bit2 -- Reserved, set to 0.

Bit1 ET4: Timer4 interrupt enable bit
1= Enable Timer4 interrupt
0= Disable Timer4 interrupt

Bit0 ET3: Timer3 interrupt enable bit
1= Enable Timer3 interrupt
0= Disable Timer3 interrupt

19.10.2 Interrupt priority control register (EIP2)

0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI interrupt priority control bit
1= High priority level
0= Low priority level

Bit6 PI2C: I^{2}C interrupt priority control bit
1= High priority level
0= Low priority level

Bit5 PWDT: WDT interrupt priority control bit
1= High priority level
0= Low priority level

Bit4 PADC: ADC interrupt priority control bit
1= High priority level
0= Low priority level

Bit3 PPWM: PWM interrupt priority control bit
1= High priority level
0= Low priority level

Bit2 -- Reserved, set to 0.

Bit1 PT4: TIMER4 interrupt priority control bit
1= High priority level
0= Low priority level

Bit0 PT3: TIMER3 interrupt priority control bit
1= High priority level
0= Low priority level

19.10.3 Peripheral interrupt flag bit register (EIF2)

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	--	TF4	TF3
R/W	R	R	--	R/W	R	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIF: SPI global interrupt indicator bit (read-only)
1= SPI interrupt occurred (this bit is automatically cleared after the specific interrupt flag is cleared)
0= No SPI interrupt occurred

Bit6 I2CIF: I^{2}C global interrupt indicator bit (read-only)
1= I^{2}C interrupt occurred (this bit is automatically cleared after the specific interrupt flag is cleared)
0= No I^{2}C interrupt occurred

Bit5 --: Reserved, set to 0.

Bit4 ADCIF: ADC interrupt flag bit
1= ADC conversion is completed (needs to be cleared by software)
0= ADC conversion is not completed

Bit3 PWMIF: PWM global interrupt indicator bit (read-only)
1= PWM interrupt occurred (this bit is automatically cleared after the specific interrupt flag is cleared)
0= No PWM interrupt occurred

Bit2 --: Reserved, set to 0.

Bit1 TF4: Timer4 overflow interrupt flag bit
1= Timer4 overflow (automatically cleared by hardware when entering the interrupt service routine, can also be cleared by software)
0= No Timer4 overflow

Bit0 TF3: Timer3 overflow interrupt flag bit
1= Timer3 overflow (automatically cleared by hardware when entering the interrupt service routine, can also be cleared by software)
0= No Timer3 overflow

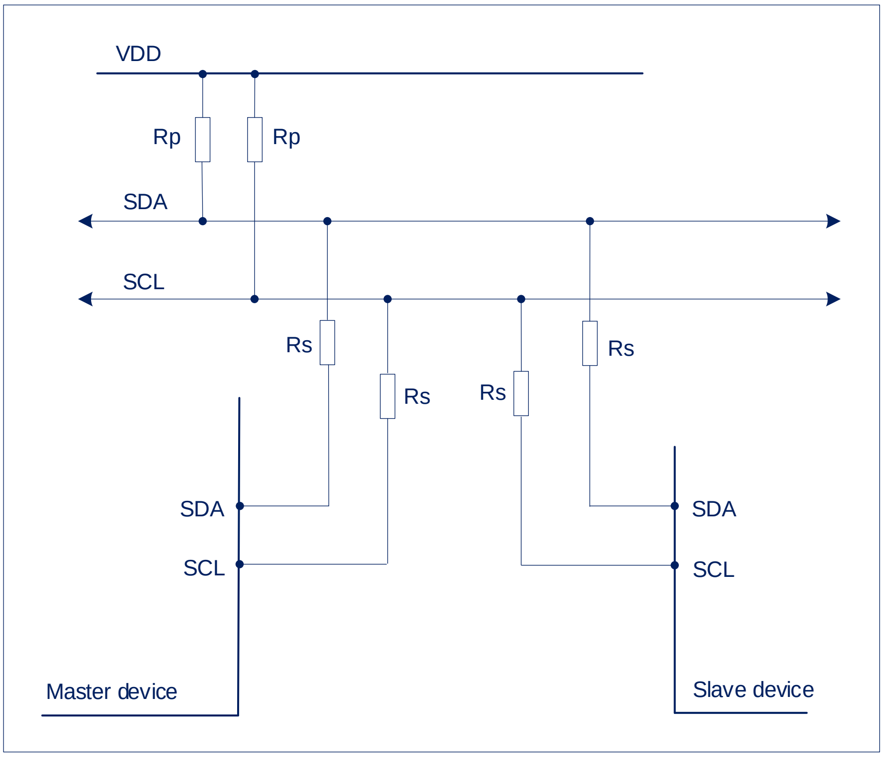
20. I²C Module

20.1 Overview

This module provides an interface between the microcontroller and the I²C bus, as illustrated in the diagram below. It supports arbitration and clock synchronization, allowing it to operate in multi-master systems. I²C supports standard and fast modes.

The I²C module has the following features:

- Supports four operational modes: master transmit, master receive, slave transmit, and slave receive.
- Supports two transfer speed modes:
  - Standard (up to 100 kb/s);
  - Fast (up to 400 kb/s);
- Executes arbitration and clock synchronization.
- Supports multi-master systems.
- In master mode, it supports both 7-bit and 10-bit addressing modes (software supported).
- In slave mode, it supports 7-bit addressing mode.
- Interrupt generation.
- Allows operation over a wide clock frequency range (with a built-in 8-bit timer).



20.2 I²C port configuration

To use the I²C function, the corresponding ports must first be configured as SCL and SDA channels. For example, to configure ports P04 and P05 for I²C functionality:

```
P04CFG=0x0C;    //Set P04 as the SCL channel
P05CFG=0x0D;    //Set P05 as the SDA channel
```

After configuring the I²C channels, the ports are set to open-drain mode by default. You can enable the internal pull-up resistors for the SCL and SDA lines by configuring PxUP, or you can add external pull-up resistors.

In master mode, the IIC outputs SCL to the slave. After sending the address or data, the slave needs to pull SCL low and return an acknowledgment signal to the master. The master must read the state of the SCL line to check whether the slave has released SCL, which indicates whether it is necessary to send the next frame of data. If the pull-up resistors connected to SCL or the board-level parasitic capacitance are too large, it may increase the read-back time, thereby affecting the communication speed of the IIC. For more details, please refer to the IIC application manual.

20.3 I²C master mode

There are six registers used to connect with the master: control, status, slave address, transmit data, receive data, and timer period registers.

Register		Address
Write	Read	
Slave Address Register I2CMSA	Slave Address Register I2CMSA	0xF4
Master Mode Control Register I2CMCR	Slave Address Register I2CMSR	0xF5
Master Transmit Data Register I2CMBUF	Master Receive Data Register I2CMBUF	0xF6
Timer Period Register I2CMTP	Timer Period Register I2CMTP	0xF7

The Master Mode Control Register (I2CMCR) and the Master Mode Status Register (I2CMSR) share the same register address, but they are physically two different registers.

The Master Transmit Data Register and the Master Receive Data Register also share the same register address. Write operations access the transmit register (I2CMBUF), while read operations access the receive register (I2CMBUF).

During write operations, the register is treated as a control register, and during read operations, it is treated as a status register.

20.3.1 I²C master mode timer period register

To generate a wide range of SCL frequencies, this module features an 8-bit timer used for standard and fast transfer.

When TIMER_PRD ≠ 0, the ideal clock period for SCL is: 2* (1+TIMER_PRD)* 10* Tsys

When TIMER_PRD ＝ 0, the ideal clock period for SCL is: 3* 10* Tsys

For detailed calculations of SCL, please refer to the IIC application manual.

0xF7	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMTP	--	MTP6	MTP5	MTP4	MTP3	MTP2	MTP1	MTP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	1

Bit7	--	Reserved, set to 0.
Bit6~Bit0	MTP<6:0>:	The bits 6-0 of the period timer register for standard and fast modes are: TIMER_PRD[6:0].

20.3.2 I²C master mode control and status register

The control register includes four bits: RUN, START, STOP, and ACK. The START bit will generate a START or REPEATED START condition. The STOP bit determines whether the data transfer stops at the end of the cycle or continues. To generate a single transfer cycle, write the desired address to the slave address register, set the R/S bit to 0, and write to the control register with ACK=x, STOP=1, START=1, RUN=1 (I2CMCR=xxx0_x111x) to execute the operation and stop. When the operation is completed (or an error occurs), an interrupt is generated. Data can be read from the receive data register.

When the I²C operates in master mode, the ACK bit must be set to 1. This will enable the I²C bus controller to automatically send an acknowledgment after each byte. When the I²C bus controller no longer requires data from the slave, this bit must be cleared to 0.

Master mode control register

0xF5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMCR	RSTS	--	--	--	ACK	STOP	START	RUN
R/W	W	R	R	W	W	W	W	W
Reset value	0	0	1	0	0	0	0	0

Bit7	RSTS:	I²C active module reset control bit
	1=	Resets the master control module (all I²C registers of the master control module, including I2CMSR).
	0=	Clears the interrupt flag bit in I²C master mode to 0.
Bit6~Bit5	--	Reserved
Bit4	--	Reserved, set to 0.
Bit3	ACK:	Acknowledge enable bit
	1=	Enable
	0=	Disable
Bit2	STOP:	Stop enable bit
	1=	Enable
	0=	Disable
Bit1	START:	Start enable bit
	1=	Enable
	0=	Disable
Bit0	RUN:	Operation enable bit
	1=	Enable
	0=	Disable

Various operations in master mode can be achieved through the following combination of control bits:

START: Send a start signal.

SEND: Send data or address.

RECEIVE: Receive data.

STOP: Send a stop signal.

Combination of control bits (IDLE status)

R/S	ACK	STOP	START	RUN	OPERATION
0	-	0	1	1	START followed by SEND (the master remains in send mode)
0	-	1	1	1	START followed by SEND and STOP
1	0	0	1	1	STARTfollowed by receive with no acknowledgment (the master remains in receiver mode)
1	0	1	1	1	START followed by RECEIVE and STOP
1	1	0	1	1	START followed by RECEIVE (the master remains in receiver mode)
1	1	1	1	1	Disable combinations
0	0	0	0	1	Disable combinations

Combination of control bits (master send state)

R/S	ACK	STOP	START	RUN	OPERATION
-	-	0	0	1	SEND operation
-	-	1	0	0	STOP
-	-	1	0	1	SEND followed bySTOP
0	-	0	1	1	Repeat START followed bySEND
0	-	1	1	1	Repeat START, followed by SEND and STOP
1	0	0	1	1	Repeat START condition followed by acknowledge RECEIVE operation (the master remains in receiver mode)
1	0	1	1	1	Repeat START followed by SEND and STOP condition
1	1	0	1	1	Repeat START condition followed by RECEIVE (the master remains in receiver mode)
1	1	1	1	1	Disable combinations

Combination of control bits (master receive state)

R/S	ACK	STOP	START	RUN	OPERATION
-	0	0	0	1	Receive operation with acknowledgment (the master remains in receiver mode)
-	-	1	0	0	STOP
-	0	1	0	1	RECEIVE followed by STOP
-	1	0	0	1	RECEIVE operation (the master remains in receiver mode)
-	1	1	0	1	Disable combinations
1	0	0	1	1	Repeat START, followed by acknowledge RECEIVE operation (the master remains in receiver mode)
1	0	1	1	1	Repeat START, followed by RECEIVE and STOP
1	1	0	1	1	Repeat START followed by RECEIVE (the master remains in receiver mode)
0	-	0	1	1	Repeat START followed by SEND (the master remains in send mode)
0	-	1	1	1	Repeat START, followed by SEND and STOP

Master mode status register (I2CMSR)

0xF5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMSR	I2CMIF	BUS_BUSY	IDLE	ARB_LOST	DATA_ACK	ADD_ACK	ERROR	BUSY
R/W	R	R	R	R	R	R	R	R
Reset value	0	0	1	0	0	0	0	0

Bit7 I2CMIF: I^{2}C master mode interrupt flag
1= In master mode, the transmission/reception is completed, or a transmission error has occurred. (cleared by software, writing 0 clears the flag.)
0= No interrupt generated

Bit6 BUS_BUSY: I^{2}C bus busy flag in master/slave mode
1= I^{2}C bus is busy and cannot perform transfer (set by a start bit in bus and a stop condition cleared).
0= --

Bit5 IDLE: I^{2}C master mode idle flag
1= Idle state
0= Run state

Bit4 ARB_LOST: I^{2}C master mode arbitration flag
1= Lost control of the bus
0= --

Bit3 DATA_ACK: I^{2}C master mode data transmission acknowledge flag
1= The last data sent did not receive an acknowledgment
0= --

Bit2 ADD_ACK: I^{2}C master mode address acknowledge flag
1= The last address sent did not receive an acknowledgment
0= --

Bit1 ERROR: I^{2}C master mode error flag
1= The addressed slave did not acknowledge/the sent data did not receive an acknowledgment/I^{2}C bus arbitration collision
0= --

Bit0 BUSY: I^{2}C master module busy flag
1= I^{2}C module is currently transmitting data
0= --

20.3.3 I^{2}C slave address register

The slave address register consists of 8 bits: 7 address bits (A6-A0) and a receive/send bit (R/S). The R/S bit determines whether the next operation is a receive (1) or a send (0).

Master mode slave address register I2CMSA

0xF4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMSA	SA6	SA5	SA4	SA3	SA2	SA1	SA0	R/S
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit1	SA<6:0>:	I^{2}C slave address in master mode
Bit0	R/S:	In I^{2}C master mode, receive/transmit status selection bit after sending slave address
	1=	Receive data after correct addressing
	0=	Send data after correct addressing

20.3.4 I^{2}C master mode transmit and receive data register

The data transmission register consists of eight data bits, which will be sent on the bus during the next send or burst send operation. The first sent bit is MD7 (MSB).

Master mode data buffer register I2CMBUF

0xF6	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CMBUF	MD7	MD6	MD5	MD4	MD3	MD2	MD1	MD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0	MD<7:0>:	Send/receive data in I^{2}C master mode.

20.4 I²C slave mode

There are five registers used to connect to the target device: the own address register, control register, status register, data transmission register, and data reception register.

Register		Address
Write	Read	
Self Address Register I2CSADR	Self Address Register I2CSADR	0xF1
Control Register I2CSCR	Status Register I2CSSR	0xF2
Data Transmission Register I2CSBUF	Data Reception Register I2CSBUF	0xF3

20.4.1 I²C self address register (I2CSADR)

The Self Address Register consists of seven address bits that identify the I²C core on the I²C bus. This register can be read from and written to.

Self address register (I2CSADR)

0xF1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSADR	--	SA6	SA5	SA4	SA3	SA2	SA1	SA0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	--	Reserved, set to 0.
Bit6~Bit0	SA<6:0>:	Self address for I²C slave mode

20.4.2 I²C slave mode control and status register (I2CSCR/I2CSSR)

The Slave Mode Control Register and the Slave Mode Status Register occupy the same register address, using different operations to distinguish access to these two registers:

Write operation: write to I2CSCR (write-only)

Read operation: read from I2CSSR (read-only)

The control register consists of two bits: RSTS and DA. The RSTS bit controls the reset of the entire I²C slave module. When the I²C bus encounters certain issues, the software can enable this bit to reinitialize the I2CS. The DA bit enables and disables the operation of the I2CS. Reading this address places the status register on the data bus.

Slave mode control register I2CSCR

0xF2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSCR	RSTS	--	--	--	--	--	--	DA
R/W	W	R	R	R	R	R	R	W
Reset value	0	0	0	0	0	0	0	0

Bit7	RSTS:	I²C slave module reset control bit
	1=	Reset slave module
	0=	No effect
Bit6~Bit1	--	Reserved, set to 0.
Bit0	DA:	I²C slave mode enable bit
	1=	Enable
	0=	Disable

The status register consists of three bits: SENDFIN, RREQ, and TREQ. The SENDFIN bit indicates that the master I²C controller has completed the reception of data during a single or continuous transmission operation of the I2C slave. The RREQ bit indicates that the I2C slave device has received a data byte from the I²C master, and the I2C slave device should read one data byte from the receive data register (I2CSBUF). The TREQ bit indicates that the I2C slave device is addressing itself as a transmitter, and the I2C slave device should write one data byte to the transmit data register (I2CSBUF). If the I²C interrupt is enabled, setting any of these three flags to 1 will generate an interrupt.

In slave mode, the bus busy flag is determined by Bit 6 (BUS_BUSY) of the master mode status register (I2CMSR). When the bus is idle, I2CMSR reads 0x20. When the start condition is generated until the stop condition is generated, the I2CMSR register reads 0x60. When the stop condition occurs, I2CMSR reads 0x20.

Slave mode status register I2CSSR

0xF2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSSR	--	--	--	--	--	SENDFIN	TREQ	RREQ
R/W	--	--	--	--	--	R	R	R
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit3 -- Reserved, set to 0.

Bit2 SENDFIN: I^{2}Cslave mode transmission complete flag (read-only)

1= The master device no longer needs data; TREQ is no longer set to 1, and the current data transfer is completed. (Automatically cleared after reading I2CSCR).

0= --

Bit1 TREQ: I^{2}C slave mode ready transmit flag (read-only)

1= The slave device has been addressed as a transmitter or the master device is ready to receive data. (Automatically cleared after writing to I2CSBUF)

0= --

Bit0 RREQ: I^{2}C slave mode reception complete flag (read-only)

1= The reception is completed. (Automatically cleared after reading from I2CSBUF)

0= The reception is not yet completed

20.4.3 I^{2}C slave mode transmit and receive buffer register (I2CSBUF)

0xF3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
I2CSBUF	I2CSBUF7	I2CSBUF6	I2CSBUF5	I2CSBUF4	I2CSBUF3	I2CSBUF2	I2CSBUF1	I2CSBUF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 I2CSBUF<7:0>: I^{2}C data to be sent or received

Write: Write the data to be sent (sent from the most significant bit to the least significant bit)

Read: Data that has already been received

20.5 I^{2}C interrupt

The interrupt number for I^{2}C is 21, with an interrupt vector of 0x00AB. To enable the I^{2}C interrupt, the enable bit I2CIE must be set to 1, and the global interrupt enable bit EA must also be set to 1.

When all relevant I^{2}C interrupts are enabled and the I^{2}C global interrupt flag I2CIF is set to 1, the CPU will enter the interrupt service routine. The operation attribute of I2CIF is read-only and is independent of the state of I2CIE.

If any of the following interrupt flags are set to 1—I^{2}C master mode interrupt flag I2CMIF, slave mode transmission complete flag SENDFIN, slave mode ready to transmit flag TREQ, or slave mode reception complete flag RREQ—then the global I^{2}C interrupt flag I2CIF will be set to 1. I2CIF will only be automatically cleared to 0 when all four of these flags are 0.

20.5.1 Interrupt mask register (EIE2)

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI interrupt enable bit
1= Enable SPI interrupt
0= Disable SPI interrupt

Bit6 I2CIE: I^{2}C interrupt enable bit
1= Enable I^{2}C interrupt
0= Disable I^{2}C interrupt

Bit5 WDTIE: WDT interrupt enable bit
1= Enable WDT overflow interrupt
0= Disable WDT overflow interrupt

Bit4 ADCIE: ADC interrupt enable bit
1= Enable ADC interrupt
0= Disable ADC interrupt

Bit3 PWMIE: PWM global interrupt enable bit
1= Enable all PWM interrupts
0= Disable all PWM interrupts

Bit2 --: Reserved, set to 0.

Bit1 ET4: Timer4 interrupt enable bit
1= Enable Timer4 interrupt
0= Disable Timer4 interrupt

Bit0 ET3: Timer3 interrupt enable bit
1= Enable Timer3 interrupt
0= Disable Timer3 interrupt

20.5.2 Interrupt priority control register (EIP2)

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	PSPI:	SPI interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit6	PI2C:	I^{2}C interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit5	PWDT:	WDT interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit4	PADC:	ADC interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit3	PPWM:	PWM interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit2	--	Reserved, set to 0.
Bit1	PT4:	TIMER4 interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit0	PT3:	TIMER3 interrupt priority control bit
	1=	High priority level
	0=	Low priority level

20.5.3 Peripheral interrupt flag bit register (EIF2)

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	--	TF4	TF3
R/W	R	R	--	R/W	R	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIF: SPI global interrupt indicator bit (read-only)
1= SPI generated an interrupt (this bit is automatically cleared after clearing the specific interrupt flags).
0= No SPI interrupt generated

Bit6 I2CIF: I^{2}C global interrupt indicator bit (read-only)
1= I^{2}C generated an interrupt (this bit is automatically cleared after clearing the specific interrupt flags)
0= No I^{2}C interrupt generated

Bit5 --: Reserved, set to 0.

Bit4 ADCIF: ADC interrupt flag bit
1= ADC conversion completed; must be cleared by software.
0= ADC conversion not completed.

Bit3 PWMIF: PWM global interrupt indicator bit (read-only)
1= PWM generated an interrupt (this bit is automatically cleared after clearing the specific interrupt flags)
0= No PWM interrupt generated

Bit2 --: Reserved, set to 0.

Bit1 TF4: Timer4 overflow interrupt flag bit
1= Timer4 overflow occurred; automatically cleared by hardware when entering the interrupt service routine, can also be cleared by software.
0= Timer4 no overflow

Bit0 TF3: Timer3 overflow interrupt flag bit
1= Timer3 overflow occurred; automatically cleared by hardware when entering the interrupt service routine, can also be cleared by software.
0= Timer3 no overflow

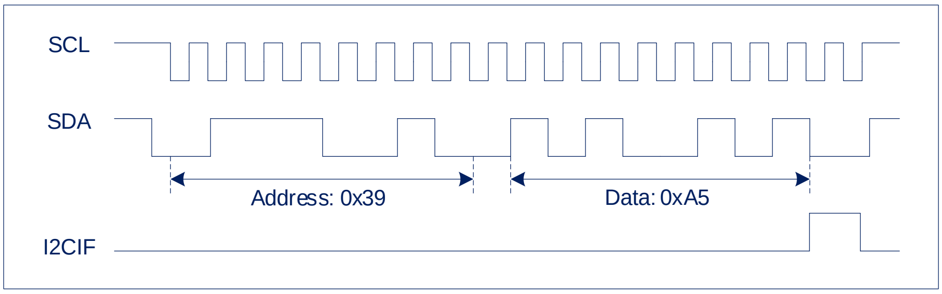
20.6 I²C slave mode transfer

In this section, all presented waveforms assume the I²C address is 0x39 (“00111001”).

20.6.1 One-shot reception

The following diagram shows the signal sequence received by I²C during a single data period. The one-shot reception sequence is as follows:

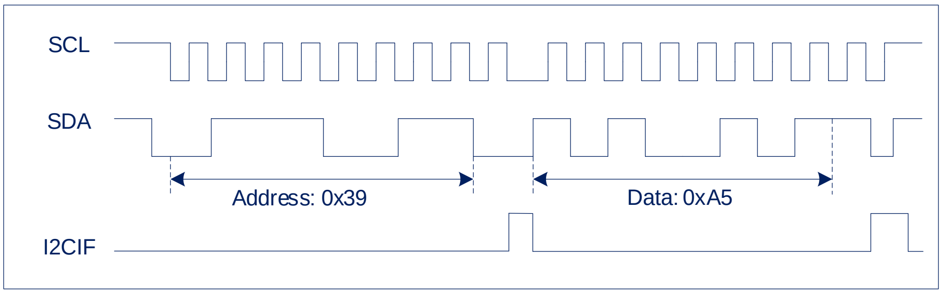
- Start condition;
- I2C is addressed as a receiver by the I2C master;
- The address is acknowledged by the I2C;
- Data is received by the I2C;
- Data is acknowledged by the I2C;
- Stop condition.



20.6.2 One-shot transmission

The following diagram shows the signal sequence sent by I²C during a single data period. The one-shot transmission sequence is as follows:

- Start condition;
- I2C is addressed as a transmitter by the I2C master;
- The address is acknowledged by the I2C;
- Data is transmitted by the I2C;
- Data is not acknowledged by the I2C master;
- Stop condition.



20.6.3 Continuous reception

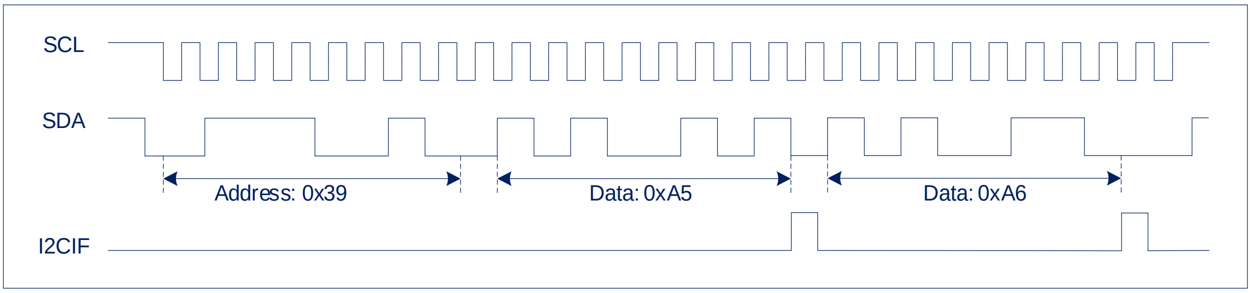
The following diagram shows the signal sequence received by I^{2}C during continuous data reception. The continuous reception sequence is as follows:

- Start condition.
- I2C is addressed as a receiver by the I2C master.
- The address is acknowledged by the I2C.

1) Data is received by the I2C.
2) Data is acknowledged by the I2C.

- Stop condition.

Sequences 1) and 2) repeat until the stop condition occurs.



20.6.4 Continuous transmission

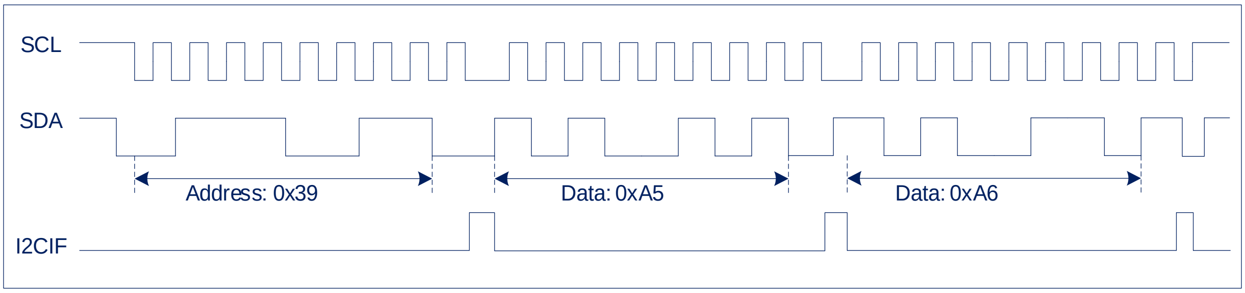
The following diagram shows the signal sequence sent by I^{2}C during continuous data transmission. The continuous transmission sequence is as follows:

- Transmission condition.
- I2C is addressed as a transmitter by the I2C master.
- The address is acknowledged by the I2C.

1) Data is sent by the I2C.
2) The I2C master acknowledges the data.
3) The last data is not acknowledged by the I2C master.

- Stop condition.

Sequences 1) and 2) are repeated until the last transmitted data is not acknowledged by the I^{2}C master in 3).



21. UARTn Module

21.1 Overview

The Universal Asynchronous Receiver-Transmitter (UART0 / UART1) provides a flexible method for full-duplex data exchange with external devices.

Inside UARTn, there are two physically independent receive and transmit buffers, SBUFn, which can be distinguished through read and write instructions to either the receive buffer or the transmit buffer. When writing to SBUFn, data is loaded into the transmit buffer; when reading SBUFn, the contents of the receive buffer are read.

UARTn has four operational modes: one synchronous mode and three asynchronous modes. Modes 2 and 3 have multi-slave communication capabilities, which are enabled by setting the SMn2 bit in the SCONn register to 1. The host processor first sends an address byte to identify the target slave. The address byte is different from the data byte because the 9th bit of the address byte is 1, while the data byte is 0. When SMn2=1, the slave will not be interrupted by the data byte. The address byte will interrupt all slaves. The addressed slave will clear its SMn2 bit and prepare to receive the forthcoming data byte. The unaddressed slave will set SMn2 to 1 and ignore the incoming data.

21.2 UARTn port configuration

Before using the UARTn module, the corresponding ports need to be configured as the TXDn and RXDn channels for UARTn. For example, the port configuration for UART0/1 is as follows:

P25CFG = 0x08;   //Configure P25 as TXD0 channel

P26CFG =0x09;   //Configure P26 as RXD0 channel; in master synchronous mode, the port is automatically configured as open-drain with a pull-up resistor

P35CFG = 0x0A;   //Configure P35 as TXD1 channel

P21CFG =0x0B;   //Configure P21 as RXD1 channel; in master synchronous mode, the port is automatically configured as open-drain with a pull-up resistor

When using, it is recommended to first set the working mode and then configure the respective ports as serial ports. When the serial port is configured for asynchronous mode, the pull-up resistor on the RXD0/RXD1 ports can be configured using the pull-up resistor control register PxUP.

21.3 UARTn baud rate

In mode 0, the baud rate of UARTn is fixed to be Fsys/12. In mode 2, the baud rate is fixed to be either Fsys/32 or Fsys/64. In modes 1 and 3, the baud rate is generated by Timer1, Timer4, Timer2, or the BRT module. The chip selects which timer to use as the baud rate clock source determined by the FUNCCR register.

21.3.1 Baud rate clock source

When UARTn is in modes 1 and 3, the baud rate clock source can be selected as follows:

When {FUNCCR[2],FUNCCR[0]}=00, select Timer1 as the baud rate generator for UART0.

When {FUNCCR[2],FUNCCR[0]}=01, select Timer4 as the baud rate generator for UART0.

When {FUNCCR[2],FUNCCR[0]}=10, select Timer2 as the baud rate generator for UART0.

When {FUNCCR[2],FUNCCR[0]}=11, select BRT as the baud rate generator for UART0.

When {FUNCCR[3],FUNCCR[1]}=00, select Timer1 as the baud rate generator for UART1.

When {FUNCCR[3],FUNCCR[1]}=01, select Timer4 as the baud rate generator for UART1.

When {FUNCCR[3],FUNCCR[1]}=10, select Timer2 as the baud rate generator for UART1.

When {FUNCCR[3],FUNCCR[1]}=11, select BRT as the baud rate generator for UART1.

21.3.2 Baud rate calculation

In modes 1 and 3, the baud rate calculation for UARTn varies depending on the selected clock source. The formulas for calculating the baud rate are as follows:

1) Timer1 or Timer4 in 8-bit auto-reload mode. The baud rate formula is:

$$BaudRate=\frac{Fsys\times 2^{SMODn}}{32\times\left(4\times 3^{1-TxM}\right)\times(256-THx)}(x=1,4)$$

SMODn is the baud rate selection bit, set by the PCON register. T1M is the Timer1 clock selection bit, set by the CKCON[4] register. T4M is the Timer4 clock selection bit, set by the T34MOD[6] register. To set the TH1/TH4 values for Timer1 or Timer4 at the corresponding baud rate:

$$THx=256-\frac{Fsys\times 2^{SMODn}}{32\times\left(4\times 3^{1-TxM}\right)\times BaudRate}(x=1,4)$$

2) Timer2 in overflow auto-reload mode. The baud rate formula is:

$$BaudRate=\frac{Fsys\times 2^{SMODn}}{32\times 12\times 2^{T2PS}\times(65536-\{RLDH,RLDL\})}$$

T2PS is the Timer2 clock prescaler selection bit, set by the T2CON[7] register. To set the {RLDH, RLDL} values for Timer2 at the corresponding baud rate:

$$\{RLDH,RLDL\}=65536-\frac{Fsys\times 2^{SMODn}}{32\times 12\times 2^{T2PS}\times BaudRate}$$

3) BRT as baud rate generator. The baud rate formula is:

$$BaudRate=\frac{Fsys\times 2^{SMODn}}{32\times(65536-\{BRTDH,BRTDL\})\times 2^{BRTCKDIV}}$$

BRTCKDIV is the BRT timer prescaler selection bit, set by the BRTCON register. To set the {BRTDH, BRTDL} values for BRT at the corresponding baud rate:

$$\{BRTDH,BRTDL\}=65536-\frac{Fsys\times 2^{SMODn}}{32\times 2^{BRTCKDIV}\times BaudRate}$$

21.3.3 Baud rate error

In modes 1 and 3 of UARTn, the errors at different baud rates when selecting different baud rate clock sources are as follows:

Table 1), 2) shows some baud rate related information in variable baud rate mode with 8-bit auto-reload mode for timer 1/4. Table 3), 4) shows some baud rate information of BRT timer overflow rate as UART clock source in variable baud rate mode.

1) SMODn=0, T1M=1/T4M=1

Baud rate	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz			Fsys=48MHz		
bps	{TH1, TH4}	Actual Rate	% Error	{TH1, TH4}	Actual Rate	% Error	{TH1, TH4}	Actual Rate	% Error	{TH1, TH4}	Actual Rate	% Error
4800	243	4808	-0.16	230	4808	-0.16	217	4808	-0.16	178	4808	-0.16
9600	--	--	--	247	9615	-0.16	236	9375	2.34	217	9615	-0.16
19200	--	--	--	--	--	--	246	18750	2.34	236	18750	2.34
38400	--	--	--	--	--	--	251	37500	2.34	246	37500	2.34
115200	--	--	--	--	--	--	--	--	--	--	--	--
250000	--	--	--	--	--	--	--	--	--	--	--	--
500000	--	--	--	--	--	--	--	--	--	--	--	--

2) SMODn=1, T1M=1/T4M=1

Baud rate	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz			Fsys=48MHz		
bps	{TH1, TH4}	Actual Rate	% Error	{TH1, TH4}	Actual Rate	% Error	{TH1, TH4}	Actual Rate	% Error	{TH1, TH4}	Actual Rate	% Error
4800	230	4808	-0.16	204	4808	-0.16	178	4808	-0.16	100	4808	-0.16
9600	243	9615	-0.16	230	9615	-0.16	217	9615	-0.16	178	9615	-0.16
19200	--	--	--	243	19230	-0.16	236	18750	2.34	217	19231	-0.16
38400	--	--	--	--	--	--	246	37500	2.34	236	37500	2.34
115200	--	--	--	--	--	--	--	--	--	--	--	--
250000	--	--	--	--	--	--	--	--	--	--	--	--
500000	--	--	--	--	--	--	--	--	--	--	--	--

3) SMODn=0, BRTCKDIV=0

Baud rate	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz			Fsys=48MHz		
bps	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual Rate	% Error
4800	65484	4808	-0.16	65432	4808	-0.16	65380	4808	-0.16	65224	4808	-0.16
9600	65510	9615	-0.16	65484	9615	-0.16	65458	9615	-0.16	65380	9615	-0.16
19200	65523	19231	-0.16	65510	19231	-0.16	65497	19231	-0.16	65458	19231	-0.16
38400	--	--	--	65523	38462	-0.16	65516	37500	2.34	65497	38462	-0.16
115200	--	--	--	--	--	--	--	--	--	65523	115385	-0.16
250000	--	--	--	--	--	--	--	--	--	65530	250000	0
500000	--	--	--	--	--	--	--	--	--	65533	500000	0

4) SMODn=1, BRTCKDIV=0

Baud rate	Fsys=8MHz			Fsys=16MHz			Fsys=24MHz			Fsys=48MHz		
bps	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual Rate	% Error	{BRTH, BRTL}	Actual Rate	% Error
4800	65432	4808	-0.16	65328	4808	-0.16	65224	4792	0.16	64911	4800	0
9600	65484	9615	-0.16	65432	9615	-0.16	65380	9615	-0.16	65224	9615	-0.16
19200	65510	19231	-0.16	65484	19231	-0.16	65458	19231	-0.16	65380	19231	-0.16
38400	65523	38462	-0.16	65510	38462	-0.16	65497	38462	-0.16	65458	38462	-0.16
115200	--	--	--	--	--	--	65523	115385	-0.16	65510	115385	-0.16
250000	--	--	--	--	--	--	--	--	--	65524	250000	0
500000	--	--	--	--	--	--	--	--	--	65530	500000	0
1000000	--	--	--	--	--	--	--	--	--	65533	1000000	0

21.4 UARTn registers

UARTn has the same functionality as the standard 8051 UART. The related registers are: FUNCCR, SBUFn, SCONn, PCON, IE, and IP. The UARTn data buffer (SBUFn) consists of two independent registers: the transmit and receive registers. Writing data to SBUFn sets this data in the UARTn output register and starts the transmission; reading data from SBUFn retrieves data from the UARTn receive register. The SCON0 register supports bit-addressable operations, while the SCON1 register does not support bit-addressable operations, which should be noted when using assembly language. The baud rate doubling is set through the PCON register.

21.4.1 UART0/1 baud rate selection register (FUNCCR)

0x91	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FUNCCR	--	--	--	--	UART1_CKS1	UART0_CKS1	UART1_CKS0	UART0_CKS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, set to 0.

Bit3 UART1_CKS1: UART1 clock source selection high bits, {UART1_CKS1, UART1_CKS}:
00= Timer1 overflow clock
01= Timer4 overflow clock
10= Timer2 overflow clock
11= BRT overflow clock

Bit2 UART0_CKS1: UART0 clock source selection high bits, {UART0_CKS1, UART0_CKS}:
00= Timer1 overflow clock
01= Timer4 overflow clock
10= Timer2 overflow clock
11= BRT overflow clock

Bit1 UART1_CKS: For the UART1 timer clock source selection low bits, see the description of UART1_CKS1.

Bit0 UART0_CKS: For the UART0 timer clock source selection low bits, see the description of UART1_CKS0.

21.4.2 UARTn buffer register (SBUFn)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SBUFn	BUFFERn7	BUFFERn6	BUFFERn5	BUFFERn4	BUFFERn3	BUFFERn2	BUFFERn1	BUFFERn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	X	X	X	X	X	X	X	X

Register SBUF0 address: 0x99; Register SBUF1 address: 0xEB.

Bit7~Bit0 BUFFERn<7:0>: Buffer data register
Write: UARTn start transmit data
Read: Read received data

21.4.3 UART control register (SCONn)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCONn	UnSM0	UnSM1	UnSM2	UnREN	UnTB8	UnRB8	TIn	RIn
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Register SCON0 address: 0x98; Register SCON1 address: 0xEA.

Bit7~Bit6 UnSM0-UnSM1: Multi-machine communication control bit
00= Master control synchronous mode
01= 8-bit asynchronous mode, variable baud rate
10= 9-bit asynchronous mode, baud rate Fsys/32 or Fsys/64
11= 9-bit asynchronous mode, variable baud rate

Bit5 UnSM2: Multi-machine communication control bit
1= Enable
0= Disable

Bit4 UnREN: Receive enable bit
1= Enable
0= Disable

Bit3 UnTB8: 9th bit of sent data (primarily used for sending in 9-bit asynchronous mode)
1= 9th bit data is 1
0= 9th bit data is 0

Bit2 UnRB8: 9th bit of received data (primarily used for sending in 9-bit asynchronous mode)
1= Received 9th bit data is 1
0= Received 9th bit data is 0

Bit1 TIn: Transmit interrupt flag bit (requires software to clear)
1= The transmit buffer is empty and ready to send the next frame of data.
0= --

Bit0 RIn: Receive interrupt flag bit (requires software to clear)
1= The receive buffer is full, and after reading, the next frame of data can be received.
0= --

The UARTn modes are listed in the following table.

SMn0	SMn1	Mode	Description	Baud rate
0	0	0	Shift register	Fsys/12
0	1	1	8-Bit UART	Controlled by Timer4/Timer1/Timer2/BRT
1	0	2	9-Bit UART	SMODn=0: Fsys/64; SMODn=1: Fsys/32
1	1	3	9-Bit UART	Controlled by Timer4/Timer1/Timer2/BRT

21.4.4 PCON register

0x87	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD0	SMOD1	--	--	--	SWE	STOP	IDLE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	SMOD0:	UART0 baud rate doubling bit
	1=	UART0 baud rate is doubled
	0=	UART0 baud rate is normal
Bit6	SMOD1:	UART1 baud rate doubling bit
	1=	UART1 baud rate is doubled
	0=	UART1 baud rate is normal
Bit5~Bit3	--	Reserved, set to 0.
Bit2	SWE:	STOP state function wake-up enable bit (Regardless of the value of SWE, the system can be restarted by power-down reset or enabled external reset.)
	0=	Disable function wake-up
	1=	Enable function wake-up (can be woken up by external interrupts and timer wake-up)
Bit1	STOP:	Sleep state control bit
	1=	Enter sleep state (automatically cleared upon exiting stop mode)
	0=	Not entered sleep state
Bit0	IDLE:	Idle state control bit
	1=	Enter idle state (automatically cleared upon exiting idle mode)
	0=	Not entered idle state

21.5 UARTn interrupt

The interrupt number for UART0 is 4, with an interrupt vector of 0x0023.

The interrupt number for UART1 is 6, with an interrupt vector of 0x0033.

To enable the UARTn interrupt, the enable bit ESn must be set to 1, and the global interrupt enable bit EA must also be set to 1. If all the relevant interrupt enables for UARTn are activated, when TIn = 1 or RIn = 1, the CPU will enter the corresponding interrupt service routine. The status of TIn/RIn is independent of the state of ESn and needs to be cleared by software. For detailed descriptions, refer to the register SCONn.

21.5.1 Interrupt mask register (IE)

0xA8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IE	EA	ES1	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	EA:	Global interrupt enable bit
	1=	Enable all unmasked interrupts
	0=	Disable all interrupts
Bit6	ES1:	UART1 interrupt enable bit
	1=	Enable UART1 interrupt
	0=	Disable UART1 interrupt
Bit5	ET2:	TIMER2 global interrupt enable bit
	1=	Enable TIMER2 all interrupts
	0=	Disable TIMER2 all interrupts
Bit4	ES0:	UART0 interrupt enable bit
	1=	Enable UART0 interrupt
	0=	Disable UART0 interrupt
Bit3	ET1:	TIMER1 interrupt enable bit
	1=	Enable TIMER1 interrupt
	0=	Disable TIMER1 interrupt
Bit2	EX1:	External interrupt 1 enable bit
	1=	Enable external interrupt 1
	0=	Disable external interrupt 1
Bit1	ET0:	TIMER0 interrupt enable bit
	1=	Enable TIMER0 interrupt
	0=	Disable TIMER0 interrupt
Bit0	EX0:	External interrupt 0 interrupt enable bit
	1=	Enable external interrupt 0
	0=	Disable external interrupt 0

21.5.2 Interrupt priority control register (IP)

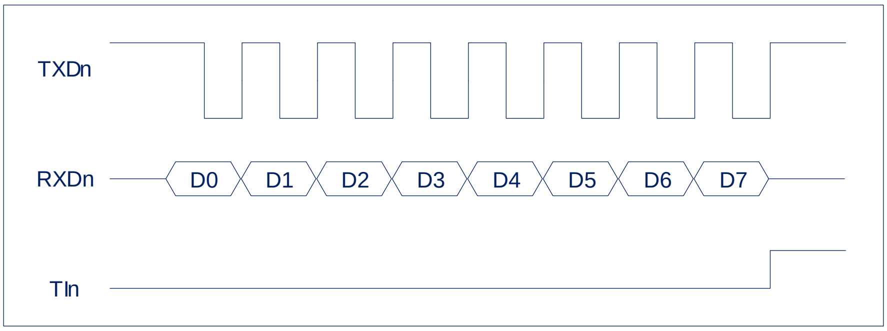
0xB8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IP	--	PS1	PT2	PS0	PT1	PX1	PT0	PX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	--	Reserved, set to 0.
Bit6	PS1:	UART1 interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit5	PT2:	TIMER2 interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit4	PS0:	UART0 interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit3	PT1:	TIMER1 interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit2	PX1:	External interrupt 1 priority control bit
	1=	High priority level
	0=	Low priority level
Bit1	PT0:	TIMER0 interrupt priority control bit
	1=	High priority level
	0=	Low priority level
Bit0	PX0:	External interrupt 0 priority control bit
	1=	High priority level
	0=	Low priority level

21.6 UARTn modes

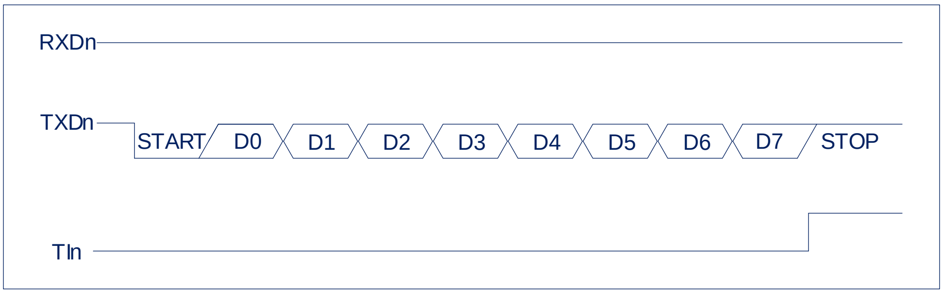
21.6.1 Mode 0- synchronous mode

In this mode, the RXDn pin acts as either an input or output, while the TXDn pin functions as a clock output. The TXDn output provides a shift clock, and the baud rate is fixed at 1/12 of the system clock frequency. Data is transmitted in 8 bits with the Least Significant Bit (LSB) sent first. To initialize reception, the flags in SCONn must be set: RIn = 0 and RENn = 1. The timing diagram for Mode 0 is shown in the figure below:



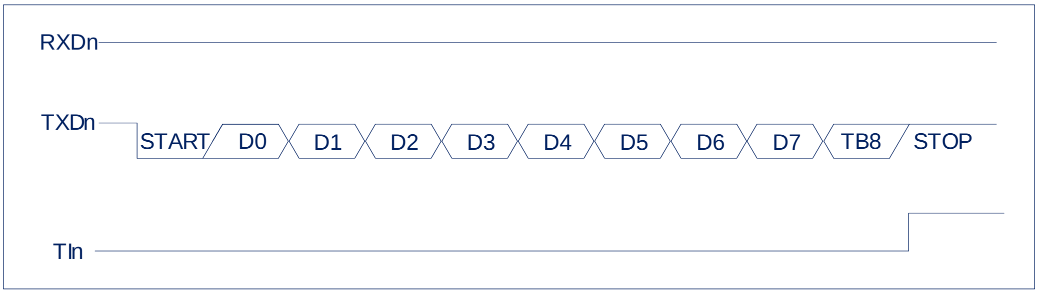
21.6.2 Mode 1-8-bit asynchronous mode (variable baud rate)

The RXDn pin is used as an input, while the TXDn pin serves as the serial output. The transmission consists of 10 bits: a start bit (always 0), 8 data bits (LSB first), and a stop bit (always 1). During reception, the start bit is synchronized for transmission, and the 8 data bits can be obtained by reading SBUFn. The stop bit sets the flag RBn8 in SCONn. The baud rate is variable and depends on the Timer4/Timer1/Timer2/BRT mode. The timing diagram for Mode 1 is shown in the figure below:



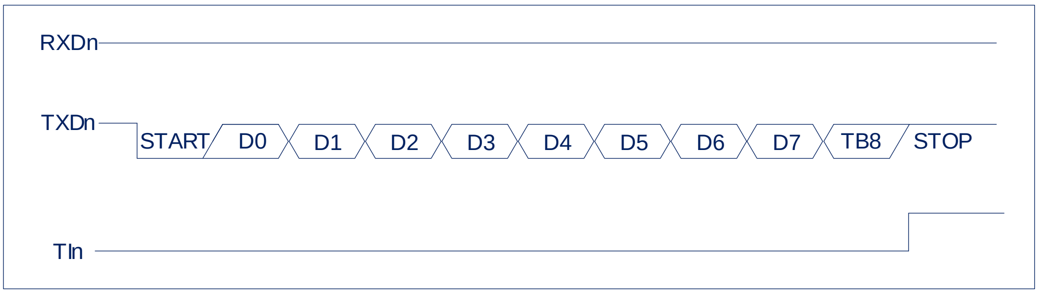
21.6.3 Mode 2-9-bit asynchronous mode (fixed baud rate)

This mode is similar to Mode 1 but has two differences. The baud rate is fixed at 1/32 or 1/64 of the CLK clock frequency, and it transmits 11 bits: a start bit (0), 8 data bits (LSB first), a programmable 9th bit, and a stop bit (1). The 9th bit can be used to control the parity of the UARTn interface: during transmission, the bit TBn8 in SCONn is output as the 9th bit, and during reception, the 9th bit affects the RBn8 in SCONn. The timing diagram for Mode 3 is shown in the figure below:



21.6.4 Mode 3-9-bit asynchronous mode (variable baud rate)

The only difference between Mode 2 and Mode 3 is that the baud rate in Mode 3 is variable. When REN0 = 1, data reception is enabled. The baud rate is variable and depends on Timer4/Timer1/Timer2/BRT mode. The timing diagram for Mode 4 is shown in the figure below:



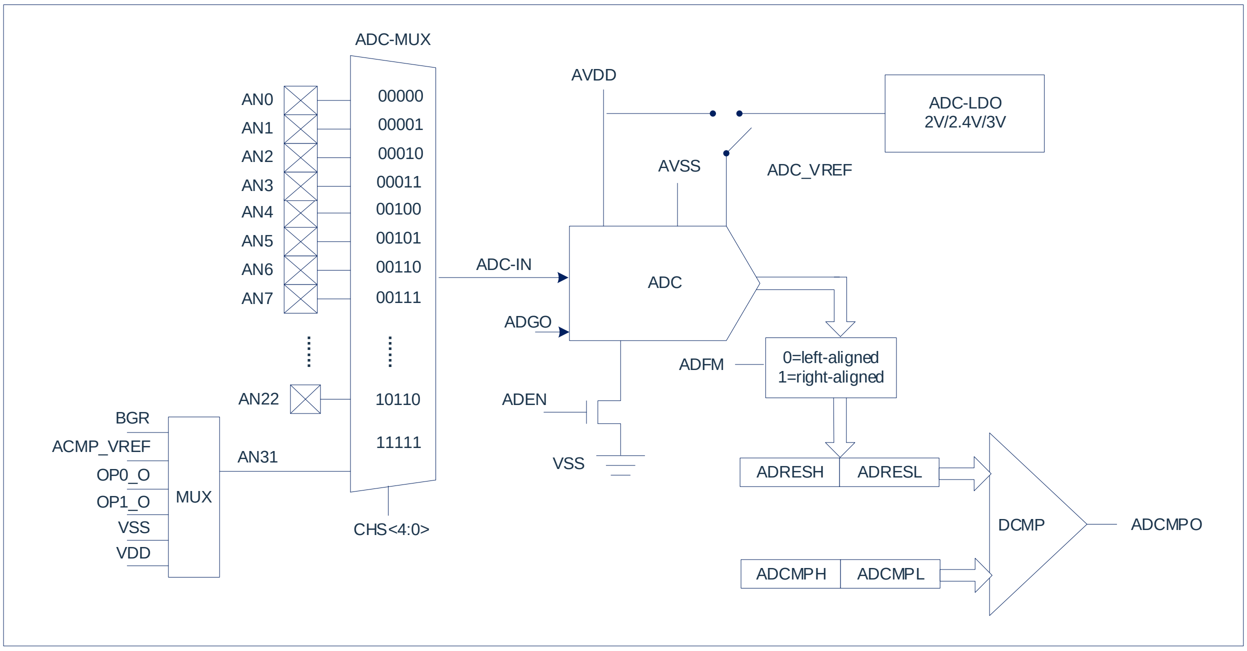
22. Analog-to-Digital Converter (ADC)

22.1 Overview

The Analog-to-Digital Converter (ADC) can convert an analog input signal into a 12-bit binary number that represents that signal. The block diagram of the ADC structure is shown in the figure below.

The port's analog input signals and internal analog signals are connected to the ADC's input through a multiplexer. The ADC uses the successive approximation method to produce a 12-bit binary result, which is then stored in the ADC result register (ADRESL and ADRESH). After the conversion is completed, the ADC can generate an interrupt. The ADC conversion result is compared with the values in the ADC comparison data registers (ADCMPL and ADCMPH), and the comparison result is stored in the ADCMPO flag.

The ADC reference voltage is always generated internally and can be supplied by AVDD or provided by the internal ADC-LDO.



22.2 ADC configuration

When configuring and using the ADC, the following factors must be considered:

- Port configuration.
- Channel selection.
- ADC conversion clock source.
- Interrupt control.
- Result storage format.

22.2.1 Port configuration

The ADC can convert both analog and digital signals. When converting analog signals, the corresponding port must be configured as an analog port.

Note: Applying an analog voltage to pins defined as digital inputs may cause overcurrent in the input buffer.

22.2.2 Channel selection

The ADCHS bits in the ADCON1 register determine which channel is connected to the analog-to-digital converter.

If the channel is changed, a certain delay is required before the next conversion begins. The ADC delay times are shown in the following table:

Delay time	Operating voltage
500ns	2.5~4.5V
200ns	4.5~5.5V

22.2.3 ADC reference voltage

The reference voltage for the ADC is provided by the chip’s VDD by default, but it can also be supplied by the internal ADC-LDO. The ADC-LDO can output one of three selectable voltages: 2.0V, 2.4V, or 3.0V.

22.2.4 Conversion clock

The conversion clock source can be selected through the software by configuring the ADCKS bit in the ADCON1 register and the ADCKS4 bit in the ADCON3 register.

The time to complete one bit conversion is defined as T_{ADCK}. The total time for a full conversion is influenced by the ADC result update selection and the sampling clock settings. The duration for a full conversion (when ADGO remains high) is shown in the table below:

ADC result update control	Time to complete a conversion	
ADCON3[3:2]	ADCON3[1]=1	ADCON3[1]=0
00	25*T_{ADCK}	29*T_{ADCK}
01	88*T_{ADCK}	104*T_{ADCK}
10	172*T_{ADCK}	204*T_{ADCK}
11	340*T_{ADCK}	404*T_{ADCK}

To achieve correct conversion results, the corresponding T_{ADCK} specification must be followed. Below is an example of proper ADC clock selection:

Fsys	F_{ADCK}（T_A=25℃）	
	V_{REF}=V_{REF1}=AVDD (AVDD=VDD)	V_{REF}=V_{REF2}=2.0V V_{REF}=V_{REF3}=2.4V V_{REF}=V_{REF4}=3.0V
8MHz	F_{sys}/1	F_{sys}/16
16MHz	F_{sys}/2	F_{sys}/32
24MHz	F_{sys}/4	F_{sys}/48
48MHz	F_{sys}/6	F_{sys}/96

Note: Any change in the system clock frequency will affect the ADC clock frequency, which could negatively impact the ADC conversion results.

22.2.5 Result formatting

The 12-bit ADC conversion result can be presented in two formats: left-aligned or right-aligned. This output format is controlled by the ADFM bit in the ADCON0 register.

When ADFM = 0, the ADC result is left-aligned.

When ADFM = 1, the ADC result is right-aligned.

22.3 ADC hardware trigger start

In addition to software-triggered ADC conversions, the ADC module also provides hardware-triggered start options. One method utilizes external port edge triggering, while the other method employs PWM edge or period triggering.

To use hardware-triggered ADC, the ADCEX bit must be set to 1, enabling the external triggering feature. After a certain delay, the hardware trigger signal sets the ADGO bit to 1, and once the conversion is completed, it will automatically reset to zero. Enabling hardware trigger functionality does not disable software triggering; thus, even when the ADC is idle, writing a 1 to the ADGO bit can still initiate an ADC conversion.

22.3.1 ADC triggered by external port edges

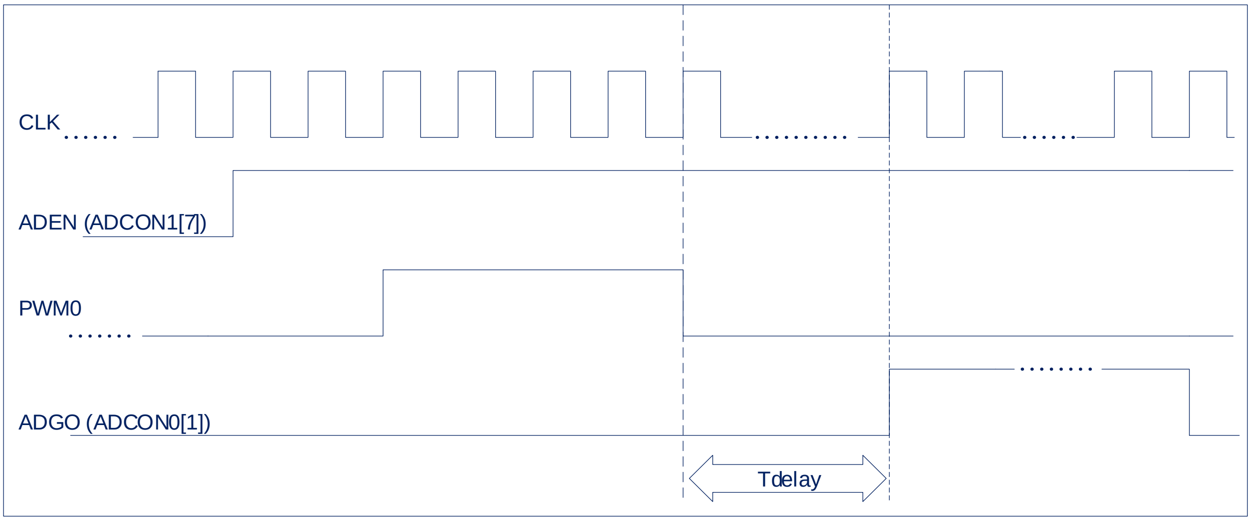
The ADET pin will automatically trigger the ADC conversion upon an edge event. At this time, the ADTGS[1:0] bits must be set to 11 (selecting external port edge triggering), and the ADEGS[1:0] bits can be configured to choose the type of edge triggering.

22.3.2 ADC triggered by PWM

PWM can trigger the ADC conversion based on either edge or period (zero crossing) detection. The ADTGS[1:0] bits select the PWM channel (PG0, PG2, PG4), while the ADEGS[1:0] bits determine whether the trigger is based on edge or period type.

22.3.3 Hardware trigger start delay

After the hardware trigger signal is generated, it does not immediately start the ADC conversion; a certain delay is required before setting the ADGO bit to 1. This delay is determined by the ADDLY[9:0] bits. The delay time for the hardware trigger signal is calculated as follows: (ADDLY+3)*Tsys. The timing structure for the delay-triggered start is illustrated in the accompanying diagram.



22.4 ADC result comparison

The ADC module provides a set of digital comparators to compare the ADC results with preset values stored in {ADCMPH, ADCMPL}. Each time an ADC conversion is completed, the result is compared against the preset value ADCMP, and the comparison result is stored in the ADCMPO flag, which automatically updates after the conversion. The ADCMPPS bit can change the polarity of the output result.

The ADC comparison result can trigger the enhanced PWM fault brake. To enable this feature, the ADFBEN bit must be set to 1.

When the enhanced PWM functionality is enabled (ADFBEN = 1), the ADC conversion result is compared with the preset values {ADCMPH, ADCMPL}. If the comparison result ADCMPO is 1, a fault brake operation is immediately initiated, clearing the start bits of all PWM channels and terminating the output of all PWM channels.

22.5 ADC operation principles

22.5.1 Starting conversion

To enable the ADC module, the ADEN bit in the ADCON1 register must be set to 1. Then, set the ADGO bit in the ADCON0 register to 1 for starting the analog-to-digital conversion (ADGO cannot be set to 1 when ADEN is 0).

22.5.2 Completing conversion

When the conversion is completed, the ADC module will:

- Clear the ADGO bit.
- Set the ADCIF flag to 1.
- Update the ADRESH:ADRESL register with the new conversion result.

22.5.3 Terminating conversion

After the ADC has started, it is necessary to wait for the ADC conversion to complete before terminating the conversion. It is prohibited to terminate the ADC conversion while it is in progress.

Note: A device reset will force all registers into their reset state. Therefore, a reset will disable the ADC module and terminate any pending conversions.

22.5.4 A/D conversion steps

The configuration steps for performing analog-to-digital conversion (ADC) using the ADC module are as follows:

1) Port configuration:
   - Disable the output driver for the pins (refer to the PxTRIS register).
   - Configure the pins as analog input pins.
2) Configure ADC interrupt (optional):
   - Clear the ADC interrupt flag.
   - Enable the ADC interrupt.
   - Enable peripheral interrupts.
   - Enable global interrupts.
3) Configure the ADC module:
   - Select the ADC conversion clock.
   - Select the ADC input channel.
   - Choose the format of the result.
   - Enable the ADC module.
4) Wait for the required acquisition time.
5) Set ADGO to 1 to start conversion.
6) Wait for ADC conversion to complete using one of the following methods:
   - Poll the ADGO bit.
   - Wait for the ADC interrupt (if interrupts are enabled).
7) Read the ADC result.
8) Clear the ADC interrupt flag (if interrupts are enabled).

Note: If the user attempts to resume sequential code execution after waking the device from sleep mode, global interrupts must be disabled.

22.5.5 Entering sleep during conversion

When the system enters sleep mode, it must wait for the ongoing ADC conversion to complete before transitioning to the sleep state. Entering sleep mode during ADC conversion is prohibited.

22.5.6 Multiple conversions

When enabling the ADC multiple conversion results accumulation feature (ADCON3[4] = 1), after setting ADGO to 1 to start the conversion, the ADC will automatically perform n consecutive conversions (where n is set by the {ADCCNTH[1:0], ADCCNTL[7:0]} register). After each conversion is completed, ADGO will be held low for one system clock cycle before being automatically set high by hardware to initiate the next conversion until n conversions are completed. Once all n conversions are finished, the ADCIF flag will be set to 1, and the accumulated value of the n conversion results will be loaded into the register {ADCRES2[7:0], ADCRES1[7:0], ADCRES0[7:0]}. To perform another series of multiple conversions, simply reset the ADGO bit to 1.

Note: When using the multiple conversion accumulation feature, it is recommended to first set the number of conversions, enable the accumulation feature, and then start the ADC conversion.

22.6 Relevant registers

There are a total of 16 registers related to A/D conversion, including:

- AD control registers: ADCON0, ADCON1, ADCON2, ADCON3, ADCLDO;
- Comparator control register: ADCMPC;
- Delay data register: ADDLYL;
- AD result data registers: ADRESH/L;
- AD multiple conversion result data registers: ADRES0/1/2;
- Conversion count registers: ADCCNTL/H;
- Comparator data registers: ADCMPH/L.

22.6.1 AD control register (ADCON0)

0xDF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON0	ADCHS4	ADFM	--	AN31SEL2	AN31SEL1	AN31SEL0	ADGO	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 ADCHS4 ADC analog channel selection bit 4
1= The channel allocation is described in the following register ADCON1.
0= --

Bit6 ADFM: ADC conversion result format selection bit
1= Right alignment
0= Left alignment

Bit5 -- Reserved, set to 0.

Bit4~Bit2 AN31SEL<2:0>: ADC channel 31 input source selection bit
000= BGR(1.2V);
001= ACMP_VREF comparator negative terminal reference voltage (see the ACMP section for details).
010= OP0_O;
011= OP1_O;
100= Reserved: Not for use.
101= VSS (ADC reference ground);
110= Reserved: Not for use.
111= VDD (ADC default reference voltage).

Bit1 ADGO: ADC conversion start bit (When this bit is set to 1, ADEN must also be set to 1; otherwise, the operation will be ineffective.);
1= Writing: Starts the ADC conversion. (This bit will also be set to 1 when the ADC is triggered by hardware.)
Reading: Indicates that the ADC is currently converting.
0= Writing: Invalid.
Reading: Indicates that the ADC is idle/completed the conversion.
During the ADC conversion period (ADGO=1), any software and hardware trigger signals will be ignored.

Bit0 -- Reserved, set to 0.

22.6.2 AD control register (ADCON1)

0xDE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON1	ADEN	ADCKS2	ADCKS1	ADCKS0	ADCHS3	ADCHS2	ADCHS1	ADCHS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	1	0	0	0	0	0	0

Bit7 ADEN: ADC enable bit
1= Enable ADC
0= Disable ADC, no operating current consumption.

Bit6~Bit4 ADCKS<2:0>: ADC conversion clock select bit (jointly controlled with ADCCON3[0])

ADCCON3[0]=0				ADCCON3[0]=1			
000=	Fsys/2	100=	Fsys/32	000=	Fsys	100=	Fsys/48
001=	Fsys/4	101=	Fsys/64	001=	Fsys/6	101=	Fsys/96
010=	Fsys/8	110=	Fsys/128	010=	Fsys/12	110=	Fsys/192
011=	Fsys/16	111=	Fsys/256	011=	Fsys/24	111=	Reserved.

Bit3~Bit0 ADCHS<3:0>: Simulated channel selection bits for the lower 4 bits, combined with ADCHS<4> to form a 5-bit channel selection bit, ADCHS<4:0>.

00000=	AN0;	10000=	AN16;
00001=	AN1;	10001=	AN17;
00010=	AN2;	10010=	AN18;
00011=	AN3;	10011=	AN19;
00100=	AN4;	10100=	AN20;
00101=	AN5;	10101=	AN21;
00110=	AN6;	10110=	AN22;
00111=	AN7;	10111=	AN23;
01000=	AN8;	11000=	AN24;
01001=	AN9;	11001=	AN25;
01010=	AN10;	11010=	AN26;
01011=	AN11;	11011=	AN27;
01100=	AN12;	11100=	AN28;
01101=	AN13;	11101=	AN29;
01110=	AN14;	11110=	AN30;
01111=	AN15;	11111=	See ADCON0.AN31SEL description.

22.6.3 AD control register (ADCON2)

0xE9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON2	ADCEX	--	ADTGS1	ADTGS0	ADEGS1	ADEGS0	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 ADCEX: ADC hardware trigger enable bit
1= Enable
0= Disable
Bit6 -- Reserved, set to 0.
Bit5~Bit4 ADTGS: ADC hardware trigger source select bit
00= PG0(PWM0)
01= PG2 (PWM2)
10= PG4 (PWM4)
11= Port pin (ADET)
Bit3~Bit2 ADEGS: ADC hardware trigger edge select bit
00= Falling edge
01= Rising edge
10= PWM period point
11= PWM period zero-crossing point
Bit1~Bit0 -- Reserved, set to 0.

22.6.4 AD control register (ADCON3)

0xD9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON3	--	--	--	SUMEN	ADCTIMES1	ADCTIMES0	SPTIME	ADCKS4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit5 -- Reserved, set to 0.
Bit4 SUMEN: ADC conversion result accumulation control bit
1= Enable
0= Disable
Bit3~Bit2 ADCTIMES<1:0>: ADC conversion result update selection bit
00= Update the data register (ADCRESL/ADCRESH) after completing one conversion
01= Update the data register after averaging the results of four conversions
10= Update the data register after averaging the results of eight conversions
11= Update the data register after averaging the results of sixteen conversions
Bit1 SPTIME: Sampling clock count selection bit
1= 4 sampling clocks T_{ADCK}
0= 8 sampling clocks T_{ADCK}
Bit0 ADCKS4: ADC conversion clock selection bit, refer to the description of ADCON1[6:4] for specific details.

22.6.5 AD comparator control register (ADCMPC)

0xD1	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCMPC	ADFBEN	ADCMPPS	--	ADCMPO	--	--	ADDLY9	ADDLY8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 ADFBEN: ADC comparator result control PWM brake enable control bit
1= Enable
0= Disable

Bit6 ADCMPPS: ADC comparator output polarity selection bit
1= If ADRES<ADCMP, ADCMPO=1.
0= If ADRES>=ADCMP, ADCMPO=1.

Bit5 --: Reserved, set to 0.

Bit4 ADCMPO: ADC comparator output bit
This bit reflects the result of the ADC comparator output and is updated at the end of each ADC conversion.

Bit3~Bit2 --: Reserved, set to 0.

Bit1~Bit0 ADDLY<9:8>: ADC hardware trigger delay data [9:8] bits

22.6.6 AD hardware trigger delay data register (ADDLYL)

0xD3	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADDLYL	ADDLY7	ADDLY6	ADDLY5	ADDLY4	ADDLY3	ADDLY2	ADDLY1	ADDLY0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 ADDLY<7:0>: ADC hardware trigger delay data low 8 bits.

22.6.7 AD data register high ADRESH, ADFM=0 (left aligned)

0xDD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESH	ADRES11	ADRES10	ADRES9	ADRES8	ADRES7	ADRES6	ADRES5	ADRES4
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 ADRES<11:4>: ADC result register bits
Bits 11-4 of the 12-bit conversion result

22.6.8 AD data register low ADRESL, ADFM=0 (left aligned)

0xDC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESL	ADRES3	ADRES2	ADRES1	ADRES0	--	--	--	--
R/W	R	R	R	R	--	--	--	--
Reset value	X	X	X	X	--	--	--	--

Bit7~Bit4 ADRES<3:0>: ADC result register bits
Bits 3-0 of the 12-bit conversion result

Bit3~Bit0 Unused.

22.6.9 AD data register high ADRESH, ADFM=1 (right aligned)

0xDD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESH	--	--	--	--	ADRES11	ADRES10	ADRES9	ADRES8
R/W	--	--	--	--	R	R	R	R
Reset value	--	--	--	--	X	X	X	X

Bit7~Bit4 Unused.

Bit3~Bit0 ADRES<11:8>: ADC result register bits
Bits 11-8 of the 12-bit conversion result

22.6.10 AD data register low ADRESL, ADFM = 1 (right aligned)

0xDC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADRESL	ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 ADRES<7:0>: ADC result register bits
Bits 7-0 of the 12-bit conversion result

22.6.11 AD comparator data register (ADCMPH)

0xD5	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCMPH	D11	D10	D9	D8	D7	D6	D5	D4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	1	1	1	1	1	1	1

Bit7~Bit0 ADCMP<11:4>: ADC comparator data high 8 bits

22.6.12 AD comparator data register (ADCMPL)

0xD4	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCMPL	--	--	--	--	D3	D2	D1	D0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	1	1	1	1	1	1	1

Bit7~Bit4 Unused.

Bit3~Bit0 ADCMP<3:0>: ADC comparator data low 4 bits

22.6.13 AD reference voltage control register

F692H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCLDO	LDOEN	VSEL1	VSEL0	--	--	--	--	--
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 LDOEN ADC_LDO enable
1= LDO enabled, the reference voltage can only select the voltage corresponding to VSEL[1:0]
0= LDO disabled, the reference voltage is the chip supply voltage.
Bit6~Bit5 VSEL<1:0>: ADC reference voltage selection bit
00= Reserved, not allowed for use.
01= 2.0V
10= 2.4V
11= 3.0V
Bit4~Bit0 -- Reserved, set to 0.

22.6.14 AD multiple conversion count low 8 bits (ADCCNTL)

F550H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCNTL	CNT7	CNT6	CNT5	CNT4	CNT3	CNT2	CNT1	CNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 CNT<7:0> ADC conversion result accumulation count low 8 bits.

22.6.15 AD multiple conversion count high 8 bits (ADCCNTH)

F551H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCCNTH	--	--	--	--	CNT11	CNT10	CNT9	CNT8
R/W	R	R	R	R	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4 -- Reserved, set to 0.
Bit3~Bit0 CNT<11:8> ADC conversion result accumulation count high 4 bits
0x000/0x001= Accumulate 1 conversion result
0x002= Accumulate 2 conversion results
…… ……
0xFFF= Accumulate 4095 conversion results

Note: When changing the conversion count, it is recommended to disable the accumulation enable before changing the conversion count.

22.6.16 AD multiple conversion result low 8 bits (ADCRES0)

F552H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCRES0	RES7	RES6	RES5	RES4	RES3	RES2	RES1	RES0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 RES<7:0> ADC multiple conversion result low 8 bits

22.6.17 AD multiple conversion result mid 8 bits (ADCRES1)

F553H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCRES1	RES15	RES14	RES13	RES12	RES11	RES10	RES9	RES8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 RES<15:8> AD multiple conversion result mid 8 bits

22.6.18 AD multiple conversion result high 8 bits (ADCRES2)

F554H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCRES2	RES23	RES22	RES21	RES20	RES19	RES18	RES17	RES16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 RES<23:16> ADC multiple conversion result 8 bits

22.7 ADC interrupt

The ADC module allows for an interrupt to be generated upon the completion of the analog-to-digital conversion. The ADC interrupt enable bit is the ADCIE bit in the EIE2 register, and the ADC interrupt flag bit is the ADCIF bit in the EIF2 register. The ADCIF bit must be cleared by software, and it will be set to 1 after each conversion is completed, regardless of whether the ADC interrupt is enabled. The interrupt enable and priority for the ADC can be set through the relevant register bits as follows.

22.7.1 Interrupt mask register (EIE2)

0xAA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIE2	SPIIE	I2CIE	WDTIE	ADCIE	PWMIE	--	ET4	ET3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIE: SPI interrupt enable bit
1= Enable SPI interrupt
0= Disable SPI interrupt

Bit6 I2CIE: I^{2}C interrupt enable bit
1= Enable I^{2}C interrupt
0= Disable I^{2}C interrupt

Bit5 WDTIE: WDT interrupt enable bit
1= Enable WDT overflow interrupt
0= Disable WDT overflow interrupt

Bit4 ADCIE: ADC interrupt enable bit
1= Enable ADC interrupt
0= Disable ADC interrupt

Bit3 PWMIE: PWM global interrupt enable bit
1= Enable all PWM interrupt
0= Disable all PWM interrupt

Bit2 --: Reserved, set to 0.

Bit1 ET4: Timer4 interrupt enable bit
1= Enable Timer4 interrupt
0= Disable Timer4 interrupt

Bit0 ET3: Timer3 interrupt enable bit
1= Enable Timer3 interrupt
0= Disable Timer3 interrupt

22.7.2 Interrupt priority control register (EIP2)

0xBA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP2	PSPI	PI2C	PWDT	PADC	PPWM	--	PT4	PT3
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 PSPI: SPI interrupt priority control bit
1= High priority level
0= Low priority level

Bit6 PI2C: I^{2}C interrupt priority control bit
1= High priority level
0= Low priority level

Bit5 PWDT: WDT interrupt priority control bit
1= High priority level
0= Low priority level

Bit4 PADC: ADC interrupt priority control bit
1= High priority level
0= Low priority level

Bit3 PPWM: PWM interrupt priority control bit
1= High priority level
0= Low priority level

Bit2 --: Reserved, set to 0.

Bit1 PT4: TIMER4 interrupt priority control bit
1= High priority level
0= Low priority level

Bit0 PT3: TIMER3 interrupt priority control bit
1= High priority level
0= Low priority level

22.7.3 Peripheral interrupt flag bit register (EIF2)

0xB2	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIF2	SPIIF	I2CIF	--	ADCIF	PWMIF	--	TF4	TF3
R/W	R	R	--	R/W	R	--	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 SPIIF: SPI global interrupt indication bit, read-only.
1= SPI interrupt occurred, (this bit is automatically cleared after clearing the specific interrupt flag)
0= No SPI interrupt occurred

Bit6 I2CIF: I^{2}C global interrupt indication bit, read-only
1= I^{2}C interrupt occurred, (this bit is automatically cleared after clearing the specific interrupt flag)
0= No I^{2}C interrupt occurred

Bit5 --: Reserved, set to 0.

Bit4 ADCIF: ADC interrupt flag bit
1= ADC conversion completed, must be cleared by software
0= ADC conversion not completed

Bit3 PWMIF: PWM global interrupt indication bit, read-only
1= PWM interrupt occurred, (this bit is automatically cleared after clearing the specific interrupt flag)
0= No PWM interrupt occurred

Bit2 --: Reserved, set to 0.

Bit1 TF4: Timer4 overflow interrupt flag bit
1= Timer4 overflow, automatically cleared by hardware upon entering the interrupt service routine, can also be cleared by software
0= Timer4 no overflow

Bit0 TF3: Timer3 overflow interrupt flag bit
1= Timer3 overflow, automatically cleared by hardware upon entering the interrupt service routine, can also be cleared by software.
0= No Timer3 overflow

23. Analog Comparators (ACMP0/1)

The chip contains two internal analog comparators, ACMP0 and ACMP1. When the voltage at the positive terminal is greater than that at the negative terminal, the comparator outputs a logic 1; otherwise, it outputs a 0. The output polarity can also be changed through the output polarity selection bit. When the output value of the comparator changes, each comparator can generate an interrupt.

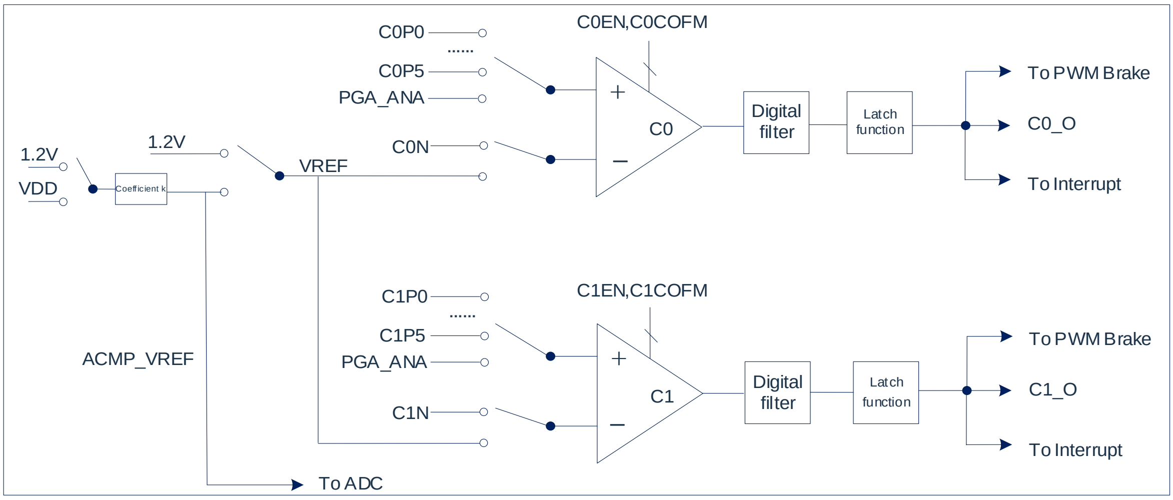
23.1 Comparator features

The comparators have the following features:

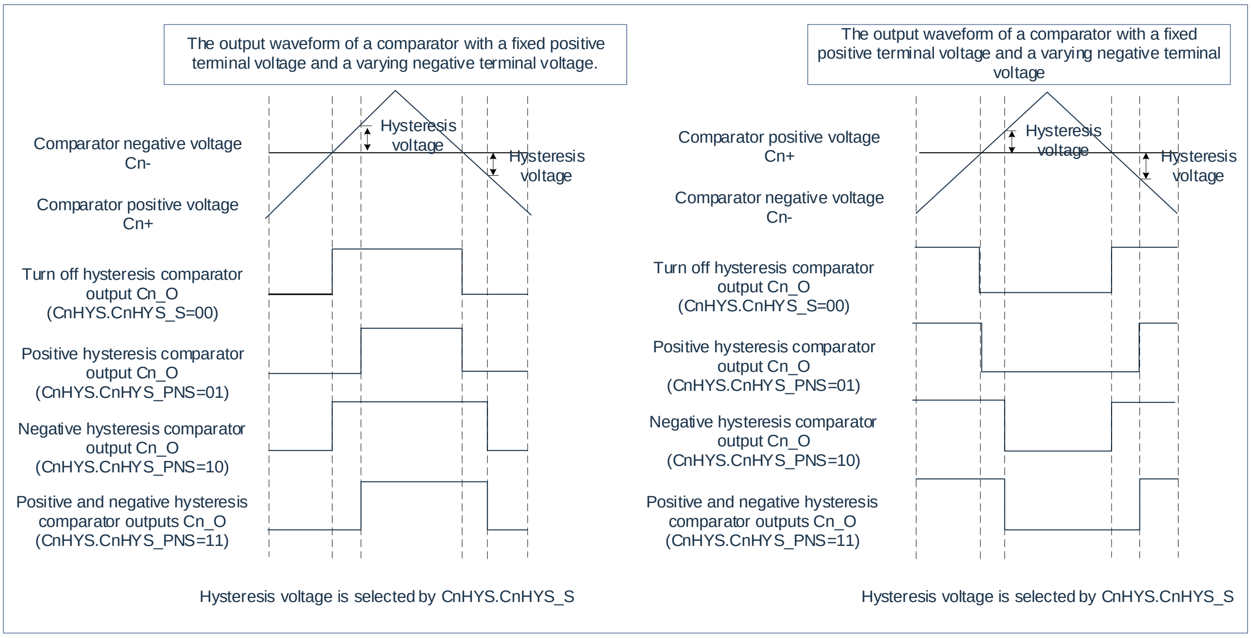
- Each comparator’s positive terminal can select from 5 input ports and 1 PGA output.
- Each comparator’s negative terminal can select between port input CnN and the internal reference voltage VREF.
- The internal reference voltage can select between the internal Bandgap (1.2V) and the ACMP_VREF output.
- ACMP_VREF reference source voltage division range: k=(2/20)~(17/20), with a total of 16 levels of selection.
- The output filter time can be selected: 0~512*Tsys.
- Supports single-side (positive/negative) and dual-side (positive and negative) hysteresis control.
- Hysteresis voltage options: 10mV, 20mV, 60mV.
- The output can be used as a braking trigger signal for enhanced PWM.
- Output changes can generate interrupts.
- The output can be latched.

23.2 Comparator structure

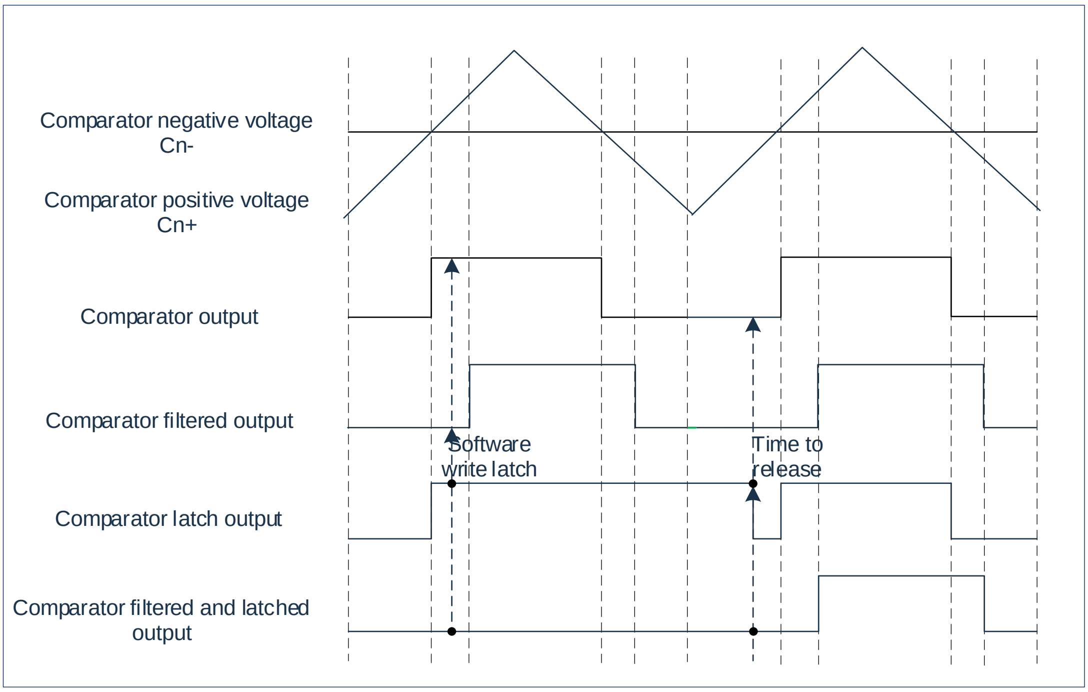
The structural block diagram of the comparator is shown below:



The comparator hysteresis control block diagram is shown below.


The output waveform of a comparator with a fixed positive terminal voltage and a varying negative terminal voltage.
Comparator negative voltage Cn-
Comparator positive voltage Cn+
Hysteresis voltage
Hysteresis voltage
Turn off hysteresis comparator output Cn_O (CnHYS.CnHYS_S=00)
Positive hysteresis comparator output Cn_O (CnHYS.CnHYS_PNS=01)
Negative hysteresis comparator output Cn_O (CnHYS.CnHYS_PNS=10)
Positive and negative hysteresis comparator outputs Cn_O (CnHYS.CnHYS_PNS=11)
Hysteresis voltage is selected by CnHYS.CnHYS_S
The output waveform of a comparator with a fixed positive terminal voltage and a varying negative terminal voltage
Comparator positive voltage Cn+
Comparator negative voltage Cn-
Hysteresis voltage
Hysteresis voltage
Turn off hysteresis comparator output Cn_O (CnHYS.CnHYS_S=00)
Positive hysteresis comparator output Cn_O (CnHYS.CnHYS_PNS=01)
Negative hysteresis comparator output Cn_O (CnHYS.CnHYS_PNS=10)
Positive and negative hysteresis comparator outputs Cn_O (CnHYS.CnHYS_PNS=11)
Hysteresis voltage is selected by CnHYS.CnHYS_S

The block diagram of the comparator latch function is shown below:


Comparator negative voltage Cn-
Comparator positive voltage Cn+
Comparator output
Comparator filtered output
Software write latch
Time to release
Comparator latch output
Comparator filtered and latched output

23.3 Relevant registers

23.3.1 Comparator control register (CnCON0)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CnCON0	CnEN	--	--	CnNS1	CnNS0	CnPS2	CnPS1	CnPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

C0CON0 address: F500H; C1CON0 address: F503H.

Bit7 CnEN: Comparator n enable bit
1= Enable
0= Disable
Bit6~Bit5 -- Reserved, set to 0.
Bit4~Bit3 CnNS<1:0>: Comparator n negative terminal channel selection bit
00= Comparator n negative terminal port
01= Internal voltage (Bandgap or ACMP_VREF)
1x= Reserved, not to be used.
Bit2~Bit0 CnPS<1:0>: Comparator n positive terminal channel selection bitCnPS<2:0>
000= CnP0
001= CnP1
010= CnP2
011= Reserved, not to be used.
100= C0P4
101= C0P5
110= PGA_ANA
111= Reserved, not to be used.

23.3.2 Comparator control register (CnCON1)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CnCON1	CnOUT	--	--	--	--	--	--	--
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

C0CON1 address: F501H; C1CON1 address: F504H.

Bit7 CnOUT: Comparator n result bit, read-only.
Bit6~ Bit0 -- Reserved, set to 0.

23.3.3 Comparator control register (CnCON2)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CnCON2	--	--	CnPOS	CnFE	CnFS3	CnFS2	CnFS1	CnFS0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

C0CON2 address: 0xF502; C1CON2 address: 0xF505.

Bit7~Bit6 -- Reserved, set to 0.

Bit5 CnPOS: Comparator n output polarity selection bit (may cause interrupt flag to be set during switching)

1= Inverted output

0= Normal output

Bit4 CnFE: Comparator n output filtering enable bit

1= Enable filtering

0= Disable filtering

Bit3~Bit0 CnFS<3:0>: Comparator n output filtering time selection bit

0000= (0~1)*Tsys

0001= (1~2)Tsys

0010= (2~3)Tsy

0011= (4~5)Tsys

0100= (8~9)Tsys

0101= (16~17)Tsys

0110= (32~33)Tsys

0111= (64~65)Tsys

1000= (128~129)Tsys

1001= (256~257)Tsys

1010= (512~513)Tsys

Other= (0~1)*Tsys

23.3.4 Comparator hysteresis control register (CnHYS)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CnHYS	--	--	--	--	CnHYS_PNS1	CnHYS_PNS0	CnHYS_S1	CnHYS_S0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

C0HYS address: F50CH; C1HYS address: F50DH.

Bit7~Bit4 -- Reserved, set to 0.

Bit3~Bit2 CnHYS_PNS<1:0> Positive and negative hysteresis selection bit

00= Disable hysteresis

01= Positive hysteresis (single-sided hysteresis)

10= Negative hysteresis (single-sided hysteresis)

11= Positive and negative hysteresis (dual-sided hysteresis)

Bit1~Bit0 CnHYS_S<1:0> Hysteresis control bit

00= Disable hysteresis

01= 10mV

10= 20mV

11= 60mV

23.3.5 Comparator reference voltage control register (CNVRCON)

F506H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CNVRCON	--	--	CNDIVS	CNSVR	CNVS3	CNVS2	CNVS1	CNVS0
R/W	R	R	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit6 -- Reserved, set to 0.
Bit5 CNDIVS: ACMP_VREF reference source selection bit
1= Select 1.2V (Bandgap) for voltage division
0= Select VDD for voltage division
Bit4 CNSVR: Comparator negative terminal internal voltage VREF selection bit
1= Select ACMP_VREF (voltage divider circuit enabled, independent of the comparator module);
0= Select 1.2V (Bandgap)
Bit3~Bit0 CNVS<3:0>: ACMP_VREF reference source voltage divider coefficient k selection bit
0000-1111= 2/20 ~ 17/20.

23.3.6 Comparator brake control register (CNFBCON)

F507H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CNFBCON	C1FBPEN	C0FBPEN	C1FBPS	C0FBPS	C1FBEN	C0FBEN	C1FBLS	C0FBLS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 C1FBPEN: Comparator 1 output level control PWM brake enable bit
0= Disable
1= Enable
Bit6 C0FBPEN: Comparator 0 output level control PWM brake enable bit
0= Disable
1= Enable
Bit5 C1FBPS: Comparator 1 output control PWM brake level selection bit
0= High level
1= Low level
Bit4 C0FBPS: Comparator 0 output control PWM brake level selection bit
0= High level
1= Low level
Bit3 C1FBEEN: Comparator 1 output event control PWM brake enable bit
0= Disable
1= Enable
Bit2 C0FBEEN: Comparator 0 output event control PWM brake enable bit
0= Disable
1= Enable
Bit1 C1FBES: Comparator 1 output event control PWM brake edge selection bit
0= Rising edge
1= Falling edge
Bit0 C0FBES: Comparator 0 output event control PWM brake edge selection bit
0= Rising edge
1= Falling edge

23.3.7 Comparator latch function control register (CnCON3)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CnCON3	-	-	CTM1	CTM0	-	-	-	SSET
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

C0CON3 address: F50EH; C1CON3 address: F50FH.

Bit7~Bit6	--	Reserved, set to 0.
Bit5~Bit4	CTM<1:0>:	Latch recovery timer selection
	0x1=	Select Timer 0 overflow recovery
	0x2=	Select Timer 1 overflow recovery
	Others=	Disable (cannot recover)
Bit3~Bit1	--	Reserved, set to 0.
Bit0	SSET:	Latch enable bit
	1=	Enable (comparator holds current output)
	0=	Disable (comparator operates normally)

23.4 Comparator interrupt

Both Comparator 0 and Comparator 1 can be configured to trigger interrupts. They share a single interrupt vector entry. After entering the interrupt service routine, the user can determine the type of interrupt by checking the interrupt flag. The comparator interrupt priority and interrupt enable can be configured using the following related register bits.

23.4.1 Interrupt priority control register (EIP1)

0xB9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EIP1	PACMP	PLSE_SCM	--	--	PP3	PP2	PP1	PP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit7 PACMP: Analog comparator interrupt priority control bit
1= High priority level
0= Low priority level
Bit6 PLSE_SCM Low-speed crystal timer and crystal stop detection interrupt priority control bit
1= High priority level
0= Low priority level
Bit5~Bit4 -- Reserved, set to 0.
Bit3 PP3: P3 port interrupt priority control bit
1= High priority level
0= Low priority level
Bit2 PP2: P2 port interrupt priority control bit
1= High priority level
0= Low priority level
Bit1 PP1: P1 port interrupt priority control bit
1= High priority level
0= Low priority level
Bit0 PP0: P0 port interrupt priority control bit
1= High priority level
0= Low priority level

23.4.2 Comparator interrupt mask register (CNIE)

F508H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CNIE	--	--	--	--	--	--	C1IE	C0IE
R/W	R	R	R	R	R	R	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit2 -- Reserved, set to 0.
Bit1 C1IE: Comparator 1 interrupt enable bit
0= Disable
1= Enable
Bit0 C0IE: Comparator 0 interrupt enable bit
0= Disable
1= Enable

23.4.3 Comparator interrupt flag register (CNIF)

F509H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CNIF	--	--	--	--	--	--	C1IF	C0IF
R/W	R	R	R	R	R	R	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~ Bit2	--	Reserved, set to 0.
Bit1	C1IF:	Comparator 1 interrupt flag (write 0 to clear)
	1=	Comparator 1 output has changed
	0=	--
Bit0	C0IF:	Comparator 0 interrupt flag (write 0 to clear)
	1=	Comparator 0 output has changed
	0=	--

24. Operational Amplifiers (OP0/1)

The chip includes two operational amplifier modules, OP0 and OP1, which can achieve basic signal amplification and signal processing functions with minimal peripheral devices.

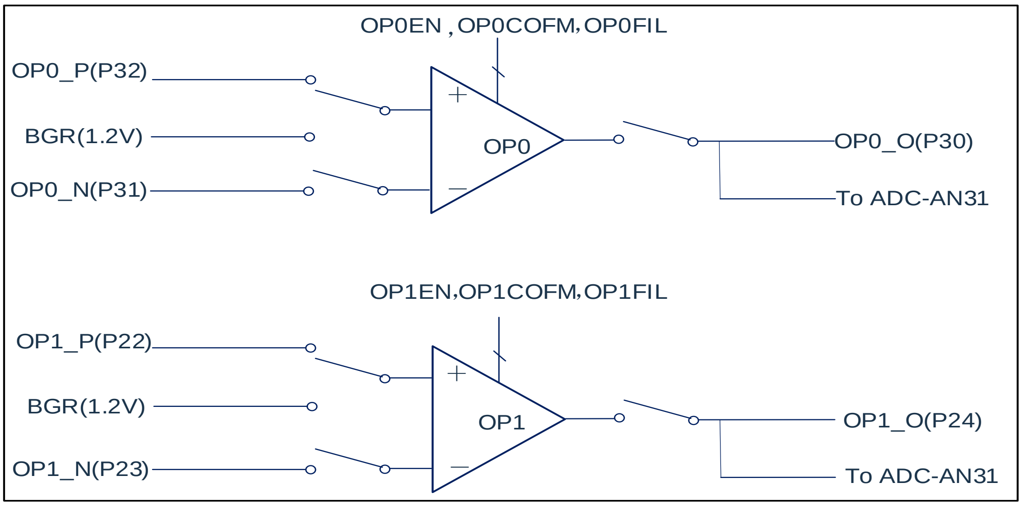
24.1 Operational amplifier features

The operational amplifiers have the following features:

- OP0 and OP1 are triple-ended and share GPIO port multiplexing.
- The non-inverting input supports internal 1.2V voltage input.
- Supports both comparator and operational amplifier modes.
- The output of OP0/1 can be connected to the internal ADC’s 31 channels for measurement.

24.2 Operational amplifier structure

The block diagram of the operational structure is shown below:



OP0/OP1 structural block diagram

24.3 Relevant registers

24.3.1 Op-Amp control register (OPnCON0(n=0-1))

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPnCON0	OPnEN	-	OPnMOD0	OPnOS	OPnNS1	OPnNS0	OPnPS1	OPnPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	1	0	0	0	0	0

OP0CON0 address: F520H; OP1CON0 address: F523H.

Bit7	OPnEN:	OPn enable bit
	1=	Enable
	0=	Disable
Bit6	-	Reserved, set to 0.
Bit5	OPnMOD<0>:	OPn adjustment mode selection
	00=	Comparator mode
	01=	Operational amplifier mode
Bit4	OPnOS:	OPn output channel selection bit
	1=	OPn_O
	0=	Disable
Bit3~Bit2	OPnNS<3:2>:	OPn negative channel selection bit
	00=	OPn_N
	Others=	Disable
Bit1~Bit0	OPnPS<1:0>:	OPn positive channel selection bit
	00=	OPn_P
	01=	BGR(1.2V)

24.3.2 Op-Amp control register (OPnCON1(n=0-1))

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
OPnCON1	OPnDOUT	--	--	--	--	--	--	--
R/W	R	R	R	R	R	R	R	R
Reset value	0	0	0	0	0	0	0	0

OP0CON1 address: F521H; OP1CON1 address: F524H.

Bit7	OPnDOUT:	OPn compare mode output, read-only
Bit6~ Bit0	--	Reserved, set to 0.

25. Programmable Gain Amplifier (PGA)

The chip includes a programmable gain amplifier module that can achieve basic signal amplification functions internally.

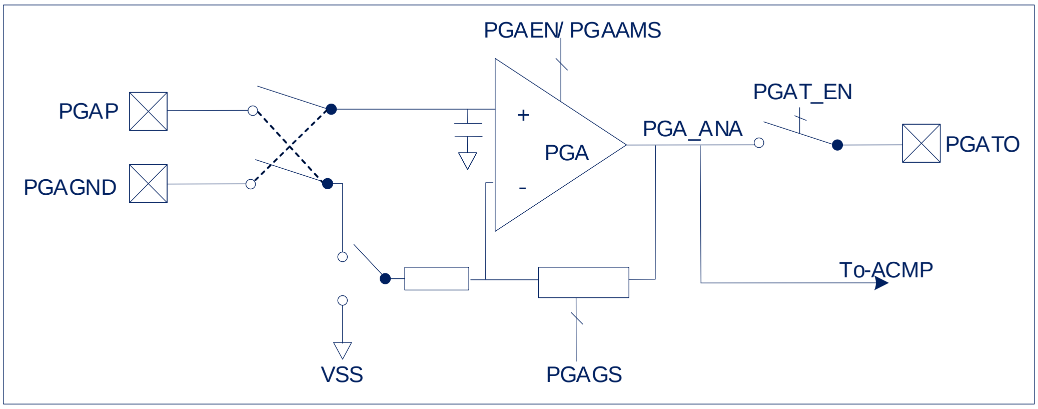
25.1 PGA features

The programmable gain amplifier has the following features:

- Multiple selectable gain levels (1/2/4/8/16/32/64/128).
- Supports single-ended and pseudo-differential inputs.
- The PGA output can be connected to the internal analog comparator input for shaping.
- Supports PGA output testing.

25.2 PGA structure

The block diagram of the PGA structure is shown below:



25.3 PGA related registers

25.3.1 PGACON0 register

F529H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PGACON0	PGAEN	PGAGS2	PGAGS1	PGAGS0	PGAIMS1	PGAIMS0	PGAIPS1	PGAIPS0
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7	PGAEN:	PGA enable bit
	1=	Enable
	0=	Disable
Bit6~Bit4	PGAGS:	PGA amplification selection
	000=	1
	001=	2
	010=	4
	011=	8
	100=	16
	101=	32
	110=	64
	111=	128
Bit3~Bit2	PGAIMS<1:0>:	PGA input mode selection bit
	00=	Single-ended input
	Others=	Differential input (pseudo-differential)
Bit1~Bit0	PGAIPS<1:0>:	PGA input port selection
	00=	Differential input: low-end input port: PGAP, high-end input port: PGAGND. Single-ended input: input port: PGAGND.
	Others=	Differential input: high-end input port: PGAP, low-end input port: PGAGND. Single-ended input: input port: PGAP.

25.3.2 PGACON2 register

F52BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PGACON2	--	--	--	--	PGAT_EN	--	--	--
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit4	--	Reserved, set to 0.
Bit3	PGAT_EN	PGATO output enable control
	1=	Enable
	0=	Disable
Bit2~Bit0	--	Reserved, set to 0.

26. Flash Memory

26.1 Overview

FLASH memory includes program memory (APROM) and non-volatile data memory (Data FLASH). The program memory space is a maximum of 64KB, divided into 128 sectors, with each sector containing 512B. The data memory space is a maximum of 1KB, divided into 2 sectors, with each sector containing 512B.

Access operations to the FLASH memory for In-Application Programming (IAP) functionality can be performed through related Special Function Registers (SFR). The SFR registers used for accessing FLASH space are as follows:

- MLOCK
- MSTATUS
- MDATA
- MADRL
- MADRH
- PCRCDL
- PCRCDH
- MREGION
- MMODE

The MLOCK register is used to enable memory operations, the MSTATUS register indicates the status of FLASH operations and sets the address to be accessed, the MDATA register forms a byte used to store the 8-bit data to be read/written, and the MADRL/MADRH registers store the address of the accessed MDATA unit. The PCRCDL/PCRCDH registers store the CRC calculation results, the MREGION register is used for memory region selection, and the MMODE register is used for selecting the memory operation mode.

Through the memory module interface, operations such as read/write/erase/CRC verification can be performed on the memory. The memory allows byte read and write, with the write time controlled by an on-chip timer. It is necessary to ensure that the data in the address has been erased before writing new data. The write and erase voltages are generated by an on-chip charge pump, which operates at a rated voltage within the device’s voltage range for byte operations.

Flash memory erase operations only support sector erase and do not support byte erase. Before modifying the data at a certain address, it is recommended to first save other data, then erase the current sector, and finally perform the data writing operation.

26.2 Relevant registers

26.2.1 FLASH protection lock register (MLOCK)

0xFF	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MLOCK	MLOCK7	MOCK6	MLOCK5	MLOCK4	MLOCK3	MLOCK2	MLOCK1	MLOCK0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 MLOCK<7:0>: Memory operation enable bit
AAH= Enable memory-related R/W/E/CRC operations
55H= Enable memory-related R/W/E/CRC operations, and increments MADDR by 1 after writing to MDATA.
Others= Disable memory-related R/W/E/CRC operations

Instruction sequence required to modify MLOCK (no other instructions can be inserted):

```
MOV                 TA,#AAH
MOV                 TA,#55H
MOV                 MLOCK,#AAH
```

26.2.2 FLASH status register (MSTATUS)

0xFE	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MSTATUS	MLOCKF	ERROR	CRCASEL	CRCCLR	START	-	-	-
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7 MLOCKF: Memory operation enable status indication bit (when MLOCK is enabled, this bit is set to 1; otherwise, it is set to 0.)
1= Enable, flash can be operated through register operations.
0= Disable, flash operations are not permitted.

Bit6 ERROR: Operation error flag bit (this bit can be cleared by writing 0)
1= If the data in the programmed address is not “FFH” (not erased) before the start of the programmed operation, the write operation is terminated immediately.
0= --

Bit5 CRCASEL: CRC verification end address selection bit
1= Select end address
0= Select start address

Bit4 CRCCLR: CRC calculation result clear bit
1= Clear the CRC calculation result register (this bit is automatically cleared to 0 by hardware).
0= --.

Bit3 START: Operation start control bit
1= Start memory R/W/E/CRC verification operations (this bit can be automatically cleared by hardware after operation completion).
0= Write: terminate or do not start program memory R/W/E check operations.
Read: operation completion or that the operation is not started.

Bit2~Bit0 -- Reserved, set to 0.

Note: The CRCASEL must be cleared after the CRC verification is completed.

26.2.3 FLASH memory data register (MDATA)

0xFB	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MDATA	MDATA7	MDATA6	MDATA5	MDATA4	MDATA3	MDATA2	MDATA1	MDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 MDATA<7:0>: Data read or written to program memory

26.2.4 FLASH memory address register (MADRL)

0xFC	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MADRL	MADRL7	MADRL6	MADRL5	MADRL4	MADRL3	MADRL2	MADRL1	MADRL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 MADRL<7:0>: Specifies the low 8 bits of the address for memory read/write operations

26.2.5 FLASH memory address register (MADRH)

0xFD	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MADRH	MADRH7	MADRH6	MADRH5	MADRH4	MADRH3	MADRH2	MADRH1	MADRH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 MADRH<7:0>: Specifies the high 8 bits of the address of the memory read/write operation

26.2.6 Program CRC result data register low 8 bits (PCRCDL)

F706H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCRCDL	PCRCD<7:0>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 PCRCD<7:0> Program CRC operation result low 8 bits data

26.2.7 Program CRC result data register high 8 bits (PCRCDH)

F707H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCRCDH	PCRCD<15:8>							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	0	0	0	0	0	0	0

Bit7~Bit0 PCRCD<15:8> Program CRC operation result high 8 bits data

26.2.8 FLASH memory area control register (MREGION)

0xF9	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MREGION	REGION7	REGION6	REGION5	REGION4	REGION3	REGION2	REGION1	REGION0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	1	0	1	0	1	0	1	0

Bit7~Bit0	REGION<7:0>	Flash area selection bit
	55H=	Select APROM area
	AAH=	Select DATA area
	69H=	Select user CONFIG area (only read operations are allowed; write and erase operations are prohibited).
	Others=	Invalid: This will exit all ongoing operations.

26.2.9 FLASH memory mode control register (MMODE)

0xFA	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
MMODE	MODE7	MODE6	MODE5	MODE4	MODE3	MODE2	MODE1	MODE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset value	0	1	1	0	1	0	0	1

Bit7~Bit0	MODE<7:0>	Flash mode selection bit
	55H=	Erase operation
	AAH=	Write operation
	69H=	Read operation
	96H=	CRC verification
	Others=	Invalid: This will exit all ongoing operations.

26.3 Function description

During the read/write/erase operations of the FLASH memory, the CPU is in a paused state. Once the operation is completed, the CPU resumes executing instructions.

Note: The read/write/erase/CRC verification operations of the FLASH must ensure that MLOCK, MREGION, and MMODE are all valid configurations to initiate the operation. If any of these registers are invalid during the operation, the operation will be prohibited.

26.3.1 FLASH read operation

The steps for the FLASH memory read operation are as follows:

1) Enable access to memory registers:

TA = 0xAA;

TA = 0x55;

MLOCK=0xAA;

2) Set the memory address to access:

Set the address using MADRL/MADRH.

3) Set the corresponding area for the memory address being accessed:

Configure the verification area using the MREGION register.

4) Set the read command:

MMODE=0x69.

5) Start the read operation:

Set MSTATUS[3] to 1.

6) Wait for 6 NOP instructions, then check if the read operation has ended:

After the read operation is completed, MSTATUS[3] will be automatically cleared to 0 by hardware.

7) Read the result:

MDATA holds the stored data from the corresponding program address.

8) Disable memory access operations:

TA = 0xAA;

TA = 0x55;

MLOCK=0x00.

26.3.2 FLASH write operation

The steps for writing to FLASH memory are as follows:

1) Enable access to memory registers:

   TA = 0xAA;

   TA = 0x55;

   MLOCK=0xAA;

2) Set the address to access the memory:

   Set the address using MADRL and MADRH.

3) Set the data to be written:

   Store the data to be written in MDATA.

4) Set the region corresponding to the memory address:

   Set the verification region using the MREGION register.

5) Set the write command:

   MMODE=0xAA.

6) Start the write operation:

   Set MSTATUS[3] to 1.

7) Wait for 6 NOP instructions, then check if the write operation is completed:

   After the write operation is completed, MSTATUS[3] will be cleared to 0 by hardware.

8) Disable memory access operations:

   TA = 0xAA;

   TA = 0x55;

   MLOCK=0x00.

FLASH self-increment write operation steps:

1) Enable access to memory registers:

   TA = 0xAA;

   TA = 0x55;

   MLOCK=0x55;

2) Set the initial memory address to access:

   Set the address using MADRL and MADRH.

3) Set the region corresponding to the memory address:

   Set the verification region using the MREGION register.

4) Set the write command:

   MMODE=0xAA.

5) Set the data to be written:

   Store the data to be written in MDATA.

6) Wait for 6 NOP instructions, then check if the write operation is completed:

   After the write operation is completed, MSTATUS[3] will be cleared to 0 by hardware.

7) Repeat steps 5 and 6 until all data is written.

8) Disable memory access operations:

   TA = 0xAA;

   TA = 0x55;

   MLOCK=0x00.

26.3.3 FLASH erase operation

The steps for erasing FLASH memory are as follows:

1) Enable access to memory registers:

   TA = 0xAA;

   TA = 0x55;

   MLOCK=0xAA;

2) Set the address to erase the memory:

   Set the address using MADRL and MADRH.

3) Set the region corresponding to the memory address:

   Set the verification region using the MREGION register.

4) Set the erase command:

   MMODE=0x55.

5) Start the erase operation:

   Set MSTATUS[3] to 1.

6) Wait for 6 NOP instructions, then check if the erase operation is completed:

   After the erase operation is completed, MSTATUS[3] will be cleared to 0 by hardware.

7) Disable memory access operations:

   TA = 0xAA;

   TA = 0x55;

   MLOCK=0x00.

26.3.4 CRC verification

The CRC verification command for the program is set by the MMODE register. The starting and ending addresses can be freely configured using the MADRL and MADRH registers. The result is stored in the PCRCDL and PCRCDH registers. During the CRC verification process in the program space, the CPU stops working and waits until the CRC calculation is completed before resuming operation. The CRC verification checks bytes in sequence from the initial address to the ending address. The steps for the CRC verification operation are as follows:

1) Enable access to program memory registers:

TA = 0xAA;

TA = 0x55;

MLOCK=0xAA;

2) Clear the previous CRC verification result:

PCRCDL=0x00; PCRCDH=0x00 (Alternatively, this can be done by writing 1 to MSTATUS[4] to clear PCRCDL and PCRCDH)

3) Set the starting and ending addresses for CRC verification:

Set MSTATUS[5] = 0, and configure the starting address using MADRL and MADRH.

Set MSTATUS[5] = 1, and configure the ending address using MADRL and MADRH.

4) Select the CRC verification region:

Configure the verification region using the MREGION register.

5) Select the CRC verification command:

MMODE=0x96.

6) Start the CRC verification:

Set MSTATUS[3] to 1.

7) Wait for 6 NOP instructions, then check if the CRC verification is completed:

After the CRC verification is completed, MSTATUS[3] will be cleared to 0 by hardware.

8) Read the CRC verification result:

PCRCDL contains the lower 8 bits of the CRC calculation result.

PCRCDH contains the higher 8 bits of the CRC calculation result.

9) Clear the CRC verification ending address selection bit:

After the CRC verification is completed, software should clear MSTATUS[5] to 0.

10) Disable memory access operations:

TA = 0xAA;

TA = 0x55;

MLOCK=0x00.

27. Unique ID (UID)

27.1 Overview

Each chip has a unique 96-bit identification number, referred to as the Unique ID (UID). This ID is set at factory and cannot be modified by the user.

27.2 UID register description

UID0

F5E0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID0	UID7	UID6	UID5	UID4	UID3	UID2	UID1	UID0
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0	UID<7:0>

UID1

F5E1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID1	UID15	UID14	UID13	UID12	UID11	UID10	UID9	UID8
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0	UID<15:8>

UID2

F5E2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID2	UID23	UID22	UID21	UID20	UID19	UID18	UID17	UID16
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0	UID<23:16>

UID3

F5E3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID3	UID31	UID30	UID29	UID28	UID27	UID26	UID25	UID24
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0	UID<31:24>

UID4

F5E4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID4	UID39	UID38	UID37	UID36	UID35	UID34	UID33	UID32
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<39:32>

UID5

F5E5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID5	UID47	UID46	UID45	UID44	UID43	UID42	UID41	UID40
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<47:40>

UID6

F5E6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID6	UID55	UID54	UID53	UID52	UID51	UID50	UID49	UID48
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<55:48>

UID7

F5E7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID7	UID63	UID62	UID61	UID60	UID59	UID58	UID57	UID56
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<63:56>

UID8

F5E8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID8	UID71	UID70	UID69	UID68	UID67	UID66	UID65	UID64
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<71:64>

UID9

F5E9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID9	UID79	UID78	UID77	UID76	UID75	UID74	UID73	UID72
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<79:72>

UID10 (0xF5EA)

F5EAH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID10	UID87	UID86	UID85	UID84	UID83	UID82	UID81	UID80
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<87:80>

UID11

F5EBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
UID11	UID95	UID94	UID93	UID92	UID91	UID90	UID89	UID88
R/W	R	R	R	R	R	R	R	R
Reset value	X	X	X	X	X	X	X	X

Bit7~Bit0 UID<95:88>

28. User Configuration

The System Configuration Register (CONFIG) contains the FLASH options for the initial conditions of the MCU. The program cannot access or manipulate this register. It includes the following content:

1. WDT (watchdog operation mode selection)
   - ENABLE
     Force enable WDT
   - SOFTWARE CONTROL (default)
     The WDT (Watchdog Timer) operation mode is controlled by the WDTRE bit in the WDCON register.
2. PROTECT
   - ENABLE
     FLASH code encryption; the read-out code will be 00H, and entry into debug mode is prohibited.
   - DISABLE (default)
     FLASH code is not encrypted.
3. FLASH_DATA_PROTECT
   - DISABLE
     FLASH data area is not encrypted.
   - ENABLE (default)
     FLASH data area is encrypted; the value read by the programmer after encryption will be 00H.
4. LVR (low voltage reset)
   - 1.8V (default)
   - 2.5V
   - 2.0V
5. DEBUG (debug mode)
   - DISABLE (default)
     Debug mode is disabled; DSCK1/2 and DSDA pins are used as general I/O ports.
   - ENABLE
     Debug mode is enabled; DSCK1/2 and DSDA pins are configured as debug ports, and the corresponding functions of these pins are disabled.
6. DEBUG_SEL (debug IO selection)
   - DSCK1/DSDA (default)
     P21/P35
   - DSCK2/DSDA
     P25/P35
7. OSC (oscillation mode)
   - HSI (default)
     48MHz
   - HSE
   - LSE(32.768KHz)
   - LSI(125KHz)
     125KHz
8. HSE/LSE_SEL (crystal port selection)
   - OSCIN1/OSCOUT1 AS HSE
     P22/P23
   - OSCIN1/OSCOUT1 AS LSE
     P22/P23
   - OSCIN2/OSCOUT2 AS HSE (default)
     P31/P32
   - OSCIN2/OSCOUT2 AS LSE
     P31/P32
9. SYS_PRESCALE (system clock pre-scaler selection)
   - FOSC/1 (default)
   - FOSC/2
   - FOSC/4
   - FOSC/8
10. HSI_FS (internal RC oscillator frequency division selection)

F_{HSI}/1	48MHz
F_{HSI}/2	24MHz
F_{HSI}/3	16MHz
F_{HSI}/6 (default)	8MHz

11. EXT_RESET (external reset configuration)

DISABLE (default)	External reset disabled
ENABLE	External reset enabled
ENABLE(OPEN PULLUP)	External reset enabled with internal pull-up resistor enabled on the reset pin

12. EXT_RESETSEL (external reset port selection)

- P00
- P01
- P02
- P03
- P04
- P05
- -
- -
- P10
- P11
- P12
- P13
- P14
- P15
- P16
- P17
- P20
- P21
- P22
- P23
- P24
- P25
- P26
- P27
- P30
- P31
- P32
- P33
- P34
- P35
- P36
- P37

13. WAKE UP_WAIT TIME (sleep-wakeup wait time for oscillator stabilization defaults to 1.0s)

- 50us
- 100us
- 500us
- 1ms
- 5ms
- 10ms
- 500ms
- 1.0s (default)

14. CPU_WAITCLOCK (memory wait clock selection)

- 1*System Clock (1T) (default)
- 2*System Clock (2T)
- 3*System Clock (3T)
- 4*System Clock (4T)
- 5*System Clock (5T)
- 6*System Clock (6T)
- 7*System Clock (7T)
- 8*System Clock (8T)

15. WRITE_ROM_PROTECT program memory area protection (protectable ranges, all default ranges are unprotected)

0000H-07FFH (protected/not protected)
0800H-0FFFH (protected/not protected)
1000H-17FFH (protected/not protected)
1800H-1FFFH (protected/not protected)
2000H-27FFH (protected/not protected)
2800H-2FFFH (protected/not protected)
3000H-37FFH (protected/not protected)
3800H-3FFFH (protected/not protected)
4000H-47FFH (protected/not protected)
4800H-4FFFH (protected/not protected)
5000H-57FFH (protected/not protected)
5800H-5FFFH (protected/not protected)
6000H-67FFH (protected/not protected)
6800H-6FFFH (protected/not protected)
7000H-77FFH (protected/not protected)
7800H-7FFFH (protected/not protected)

16. WRITE_DATA_PROTECT data area protection (protectable ranges, all default ranges are unprotected)

0000H-03FFH (protected/not protected)

17. BOOT (BOOT space selection)

● BOOT_DIS (default)	BOOT area disabled
● BOOT_1K	BOOT area size is 1K
● BOOT_2K	BOOT area size is 2K
● BOOT_4K	BOOT area size is 4K

Note:

1) The machine cycle is related to the memory wait clock selection (CPU_WAITCLOCK): Machine Cycle = T_{SYS}/CPU_WAITCLOCK.
2) When the oscillator mode is selected as HSI, the internal RC oscillator division is set to FHSI/1, and the system clock pre-scaler is set to F_{OSC}/1. When all three conditions are met, if the memory wait clock selection is set to 1*System Clock (1T) or 2*System Clock (2T), the actual memory wait clock selection becomes 3T, and the machine cycle is given by: Machine Cycle = T_{SYS}/3.
3) When the oscillator mode is selected as HSI, the internal RC oscillator division is set to FHSI/1, and the system clock pre-scaler is set to F_{OSC}/2. When all three conditions are met, if the memory wait clock selection is set to 1*System Clock (1T), the actual memory wait clock selection becomes 2T, and the machine cycle is given by: Machine Cycle = T_{SYS}/2.
4) When the oscillator mode is selected as HSI, the internal RC oscillator division is set to FHSI/2, and the system clock pre-scaler is set to F_{OSC}/1. When all three conditions are met, if the memory wait clock selection is set to 1*System Clock (1T), the actual memory wait clock selection becomes 2T, and the machine cycle is given by: Machine Cycle = T_{SYS}/2.

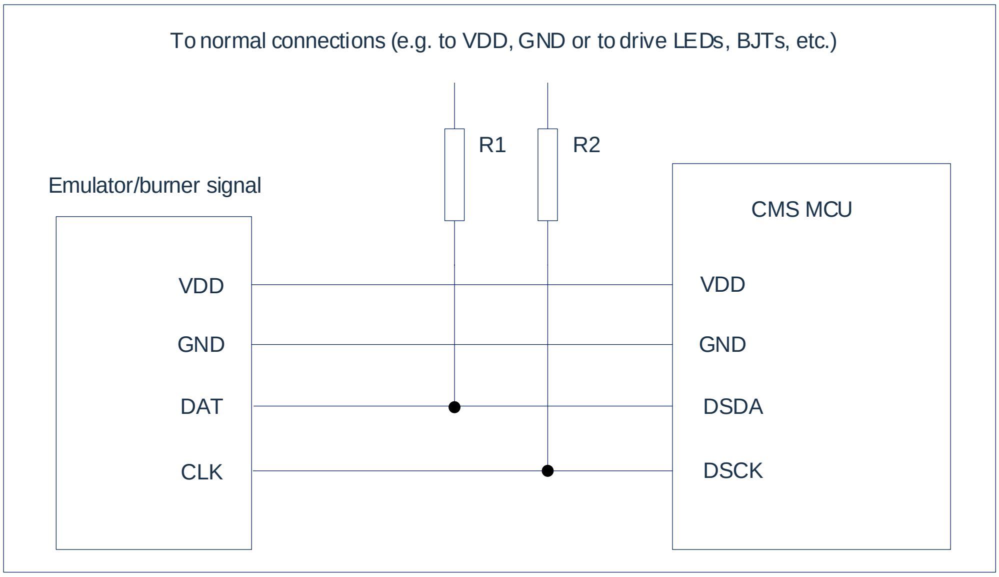
29. Online Programming and Debugging

29.1 Online programming mode

The chip can be programmed in the final application circuit using serial programming. Programming can be simply accomplished through the following four lines:

- Power line
- Ground line
- Data line
- Clock line

Online serial programming allows users to manufacture circuit boards with unprogrammed devices and only program the chips just before product delivery. This approach enables the latest version of firmware or customized firmware to be burned into the chip. A typical method for connecting online serial programming is shown in the diagram below:


To normal connections (e.g. to VDD, GND or to drive LEDs, BJTs, etc.)
R1
R2
Emulator/burner signal
CMS MCU
VDD
GND
DAT
CLK
VDD
GND
DSDA
DSCK

In the diagram above, R1 and R2 are electrical isolation devices, commonly replaced by resistors with the following values: R1⩾4.7K, R2⩾4.7K.

Note that during programming and debugging, the DSDA line must not be connected to pull-down resistors. If the actual circuit requires pull-down resistors, it is recommended to use a jumper structure. Disconnect the pull-down resistors during programming/debugging and reconnect them afterward.

29.2 Online debugging mode

The chip supports a two-wire (DSCK, DSDA) online debugging function. If the online debugging feature is used, the DEBUG in the system configuration register must be set to ENABLE. When using the debugging mode, please note the following points:

- In debugging mode, the DSCK and DSDA pins serve as dedicated debugging ports and cannot perform their GPIO or multiplexing functions.
- When entering sleep mode or idle mode during debugging, the system power and oscillator will not stop working. In this state, the wake-up functionality from sleep can be simulated. If power consumption is a concern, it is recommended to disable the debugging feature before testing the actual sleep current of the chip.
- In debugging mode, when paused, other peripheral functions will continue to run, while the WDT and Timer0/1/2/3/4 counters will stop. However, if Timer1/4 is used as the baud rate generator for UART0/1, it will continue to run during the paused state. Peripherals that continue to operate in paused mode may generate interrupts, so caution is advised during debugging.
- It is recommended not to use the WDT/WWDT reset or software reset functions in debugging mode, as the chip may lose connection with the debugger during the reset process.

30. Description of Instructions

There are a total of five categories of assembly instructions: arithmetic operations, logical operations, data transfer operations, Boolean operations, and program branching instructions. All these instructions are compatible with the standard 8051.

30.1 Description of symbols

Symbol	Description
Rn	Working registers: R0-R7
Direct	Address units of internal data memory (RAM) (00H-FFH) or addresses in special function registers (SFR)
@Ri	Indirect addressing registers (@R0 or @R1)
#data	8-bit binary constants
#data16	16-bit binary constants in instructions
Bit	Bit addresses in internal data memory (RAM) or special function registers (SFR)
Addr16	16-bit addresses with a range of 0-64KB address space
Addr11	11-bit addresses with a range of 0-2KB address space
Rel	Relative addresses
A	Accumulator

30.2 Instruction set

Mnemonic	Description
Arithmetic Operation	
ADD A,Rn	Add register to accumulator.
ADD A,direct	Add directly addressed data to accumulator.
ADD A,@Ri	Add indirectly addressed data to accumulator.
ADD A,#data	Add immediate data to accumulator.
ADDC A,Rn	Add register to accumulate or with carry.
ADDC A,direct	Add directly addressed data to accumulator with carry.
ADDC A,@Ri	Add indirectly addressed data to accumulator with carry.
ADDC A,#data	Add immediate data to accumulator with carry.
SUBB A,Rn	Subtract register from accumulator with borrow.
SUBB A,direct	Subtract directly addressed data from accumulate or with borrow.
SUBB A,@Ri	Subtract in directly addressed data from accumulator with borrow.
SUBB A,#data	Subtract immediate data from accumulate or with borrow.
INC A	Increment accumulator.
INC Rn	Increment register.
INC direct	Increment directly addressed location.
INC @Ri	Increment indirectly addressed location.
INC DPTR	Increment data pointer.
DEC A	Decrement accumulator.
DEC Rn	Decrement register.
DEC direct	Decrement directly addressed location.
DEC @Ri	Decrement indirectly addressed location.
MUL A,B	Multiply A and B.
DIV A,B	Divide A by B.
DA A	Decimally adjust accumulator.
Logic operation	
ANL A,Rn	AND register to accumulator.
ANL A,direct	AND directly addressed data to accumulator.
ANL A,@Ri	AND indirectly addressed data to accumulator.
ANL A,#data	AND immediate data to accumulator.
ANL direct,A	AND accumulator to directly addressed location.
ANL direct,#data	AND immediate data to directly addressed location.
ORL A,Rn	OR register to accumulator.
ORL A, direct	OR directly addressed data to accumulator.
ORL A,@Ri	OR indirectly addressed data to accumulator.
ORL A, #data	OR immediate data to accumulator.
ORL direct,A	OR accumulator to directly addressed location.
ORL direct,#data	OR immediate data to directly addressed location.
XRL A,Rn	Exclusive OR (XOR) register to accumulator.
XRL A,direct	XOR directly addressed data to accumulator.
XRL A,@Ri	XOR indirectly addressed data to accumulator.
XRL A,#data	XOR immediate data to accumulator.
XRL direct,A	XOR accumulator to directly addressed location.
XRL direct,#data	XOR immediate data directly addressed location.
CLR A	Clear accumulator.
CPL A	Complement accumulator.
RL A	Rotate accumulator left.

Mnemonic	Description
RLC A	Rotate accumulator left through carry.
RR A	Rotate accumulator right.
RRC A	Rotate accumulator right through carry.
SWAP A	Swap nibbles within the accumulator.
Data transfer	
MOV A,Rn	Move register to accumulator.
MOV A,direct	Move directly addressed data to accumulator.
MOV A,@Ri	Move indirectly addressed data to accumulator.
MOV A,#data	Move immediate data to accumulator.
MOV Rn,A	Move accumulator to register.
MOV Rn,direct	Move directly addressed data to register.
MOV Rn,#data	Move immediate data to register.
MOV direct,A	Move accumulator to direct.
MOV direct,Rn	Move register to direct.
MOV direct1,direct2	Move directly addressed data to directly addressed location.
MOV direct,@Ri	Move indirectly addressed data to directly addressed location.
MOV direct,#data	Move immediate data to directly addressed location.
MOV @Ri,A	Move accumulator to indirectly addressed location.
MOV @Ri,direct	Move directly addressed data to indirectly addressed location.
MOV @Ri,#data	Move immediate dato to102n directly addressed location.
MOV DPTR,#data16	Load data pointer with a 16-bitimmediate.
MOVC A,@A+DPTR	Load accumulator with a code byte relative to DPTR.
MOVC A,@A+PC	Load accumulator with a code byte relative to PC.
MOVX A,@Ri	Move external RAM (8-bitaddress) to accumulator.
MOVX A,@DPTR	Move external RAM (16-bitaddress) to accumulator.
MOVX @Ri,A	Move accumulator to external RAM (8-bitaddress).
MOVX @DPTR,A	Move accumulator to external RAM(16-bitaddress).
PUSH direct	Push directly addressed data onto stack.
POP direct	Pop directly addressed data location fromstack.
XCH A,Rn	Exchange register with accumulator.
XCH A, direct	Exchange directly addressed location with accumulator.
XCH A,@Ri	Exchange indirect RAM with accumulator.
XCHD A,@Ri	Exchange low-order nibbles of indirect and accumulator.
Boolean operation	
CLR C	Clear carry flag.
CLR bit	Clear directly addressed bit.
SETB C	Set carry flag.
SETB bit	Set directly addressed bit.
CPL C	Complement carry flag.
CPL bit	Complement directly addressed bit.
ANL C,bit	AND directly addressed bitto carry flag.
ANL C,/bit	AND complement of directly addressed bit to carry.
ORL C,bit	OR directly addressed bit to carry flag.
ORL C,/bit	OR complement of directly addressed bit to carry.
MOV C,bit	Move directly addressed bit to carry flag.
MOV bit,C	Move carry flag to directly addressed bit.
Program branching	
ACALL addr11	Absolute subroutine call
LCALL addr16	Long subroutine call

Mnemonic	Description
RET	Return from subroutine
RETI	Return from interrupt
AJMP addr11	Absolute jump
LJMP addr16	Long jump
SJMP rel	Short jump (relative address)
JMP @A+DPTR	Jump indirect relative to the DPTR
JZ rel	Jump if accumulator is zero
JNZ rel	Jump if accumulator is not zero
JC rel	Jump if carry flag is set
JNC rel	Jump if carry flag is not set
JB bit,rel	Jump if directly addressed bit is set
JNB bit,rel	Jump if directly addressed bit is not set
JBC bit,rel	Jump if directly addressed bit is set and clear bit
CJNE A,direct,rel	Compare directly addressed data to accumulator and jump if not equal
CJNE A,#data,rel	Compare immediate data to accumulator and jump if not equal
CJNE Rn,#data,rel	Compare immediate data to register and jump if not equal
CJNE @Ri,#data,rel	Compare immediate to indirect and jump if not equal
DJNZ Rn,rel	Decrement register and jump if not zero
DJNZ direct,rel	Decrement directly addressed lacation and jump if not zero
NOP	No operation for one cycle
Read-Modify-Write	
ANL	Logical AND. (ANL direct, A and ANL direct, #data)
ORL	Logical OR. (ORL direct, A and ORL direct, #data)
XRL	Logical exclusive OR. (XRL direct, A and XRL direct, #data)
JBC	Jump if bit = 1 and clear it. (JBC bit, rel)
CPL	Complement bit. (CPL bit)
INC	Increment. (INC direct)
DEC	Decrement. (DEC direct)
DJNZ	Decrement and jump if not zero. (DJNZ direct, rel)
MOV bit,C	Move carry to bit. (MOV bit, C)
CLR bit	Clear bit. (CLR bit)
SETB bit	Set bit. (SETB bit)

31. Revision History

Version	Date	Description of changes
V1.0.0	Jan 2025	1) Modified the cover page 2) Revised the description of external reset in section 3.2/5.3.2.3.