Product List

OB38S003W20,

Description

The OB38S003 is a 1T (one machine cycle per clock) single-chip 8-bit microcontroller. It has 8KB embedded program memory, and executes all ASM51 instructions fully compatible with MCS-51.

OB38S003 contains 512B on-chip RAM, up to 18 GPIOs (20L package), various serial interfaces and many peripheral functions as described below. It can be programmed via writers. Its on-chip ICE is convenient for users in verification during development stage.

The high performance of OB38S003 can achieve complicated manipulation within short time. About one third of the instructions are pure 1T, and the average speed is 8 times of traditional 8051, the fastest one among all the 1T 51-series.Its excellent EMI and ESD characteristics are advantageous for many different applications.

Ordering Information

OB38S003 ihhkL
YWW
i: process identifier { W = 2.4V ~ 5.5V}
hh: pin count
k: package type postfix {as table below }
L:PB Free identifier
{No text is Non-PB free，”P” is PB free}
Y: Year Code
WW: Week Code (01-52)

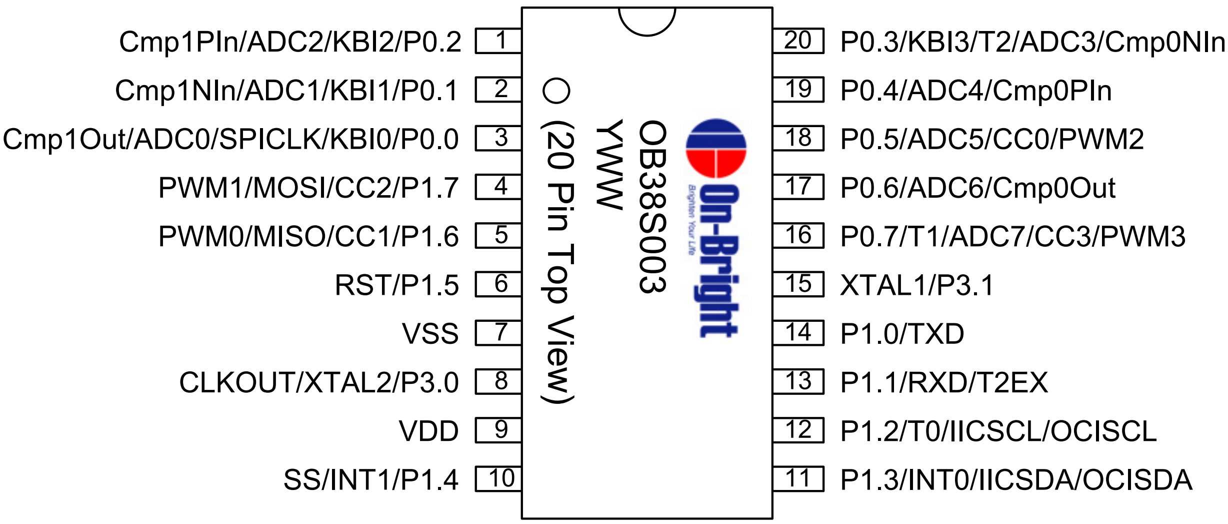
Postfix	Package
E	TSSOP(173mil)

Features

- Operating Voltage: 2.4V ~ 5.5V
- High speed architecture of 1 clock/machine cycle runs up to 16MHz.
- 1~8T can be switched on the fly.
- Instruction-set compatible with MCS-51.
- 16MHz Internal RC oscillator, with programmable clock divider
- 8KB on-chip program memory.
- 512B RAM as standard 8052,
- Dual 16-bit Data Pointers (DPTR0 & DPTR1).
- One serial peripheral interfaces in full duplex mode.
- Additional Baud Rate Generator
- Three 16-bit Timer/Counters. (Timer 0,1,2）
- 8 ~18 GPIOs(10L ~ 20L package)
- External interrupt 0,1 with four priority levels
- Programmable watchdog timer.
- One IIC interface. (Master/Slave mode)
- One SPI interface (Master/Slave mode)
- 4-channel PWM
- 4-channel 16-bit PCA for compare(PWM) / capture / reload functions
- 8-channel 12-bit analog-to-digital converter (ADC)
- CMP x1 Set (2 devices)
- ISP/IAP/ICP functions.
- ISP service program space configurable in N*128 byte (N=0 to 8) size.
- EEPROM function.
- On-Chip in-circuit emulator (ICE) functions with On-Chip Debugger (OCD).
- Keyboard interface (KBI) for four more interrupts.
- LVI/LVR (LVR deglitch 500ns)
- IO PAD ESD over 4KV
- Enhance user code protection.
- Power management unit for IDLE and power down modes.

Pin Configuration

20 Pin TSSOP



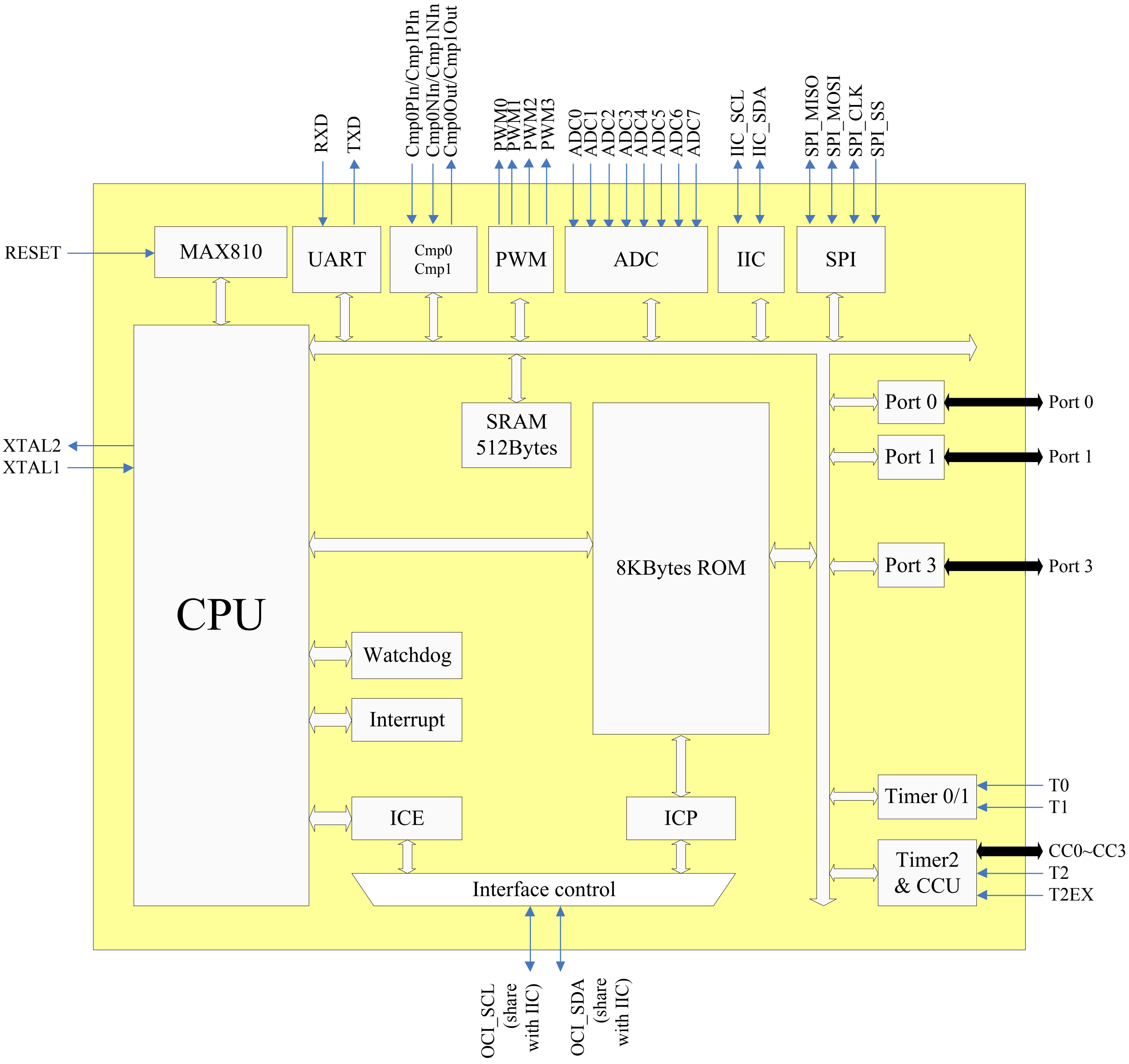
Notes：

(1) The pin Reset/P1.5 factory default is GPIO (P1.5). User can configure it to Reset by a programmer.

(2) To avoid accidentally entering ISP-Mode(refer to section 18.4), care must be taken not asserting pulse signal at RXD P1.1 during power-up while P1.6 are set to high.

(3) To apply ICP function, OSI_SDA/P1.3 and OCI_SCL/P1.2 must be set to Bi-direction mode if they are configured as GPIO in system.

Block Diagram



Pin Description

20L	Symbol	I/O	Description
1	P0.2/KBI2/ADC2/ CMP1PIn	I/O	Bit 2 of port 0 & KBI interrupt 2 & ADC input channel 2 & Cmp1 Positive Input
2	P0.1/KBI1/ADC1/ CMP1NIn	I/O	Bit 1 of port 0 & KBI interrupt 1 & ADC input channel 1 & Cmp1 Negative Input
3	P0.0/KBI0/SPICLK/ADC0/ CMP1Out	I/O	Bit 0 of port 0 & KBI interrupt 0 & SPI interface Clock pin & ADC input channel 0 & Cmp1 output
4	P1.7/CC2/MOSI/PWM1	I/O	Bit 7 of port 1 & Timer 2 compare/capture Channel 2 & SPI interface Serial Data Master Output or Slave Input pin & PWM Channel 1
5	P1.6/CC1/MISO/PWM0	I/O	Bit 6 of port 1 & Timer 2 compare/capture Channel 1 & SPI interface Serial Data Master Input or Slave Output pin & PWM Channel 0
6	P1.5/RST	I/O	Bit 5 of port 1 & Reset pin
7	VSS	I	Power supply
8	P3.0/XTAL2/CLKOUT	I/O	Bit 0 of port 3 & Crystal output & Clock Output
9	VDD	I	Power supply
10	P1.4/INT1/SS	I/O	Bit 4 of port 1 & External interrupt 1 & SPI interface Slave Select pin
11	P1.3/INT0/IICSDA/OCISDA	I/O	Bit 3 of port 1 & External interrupt 0 & IIC SDA pin & On-Chip Instrumentation Command and data I/O pin synchronous to OCI_SCL in ICE and ICP functions
12	P1.2/T0/IICSCL/OCISCL	I/O	Bit 2 of port 1 & Timer 0 external input & IIC SCL pin & On-Chip Instrumentation Clock I/O pin of ICE and ICP functions
13	P1.1/RXD/T2EX	I/O	Bit 1 of port 1 & Serial interface channel 0 receive/transmit data & Timer 2 capture trigger
14	P1.0/TXD	I/O	Bit 0 of port 1 & Serial interface channel 0 transmit data or receive clock in mode 0
15	P3.1/XTAL1	I/O	Bit 1 of port 3 & Crystal input
16	P0.7/T1/ADC7/CC3/PWM3	I/O	Bit 7 of port 0 & Timer 1 external input & ADC input channel 7& Timer 2 compare/capture Channel 3& PWM Channel 3
17	P0.6/ADC6/CMP0Out	I/O	Bit 6 of port 0 & ADC input channel 6 & Cmp0 Output
18	P0.5/ADC5/CC0/PWM2	I/O	Bit 5 of port 0 & ADC input channel 5 & Timer 2 compare/capture Channel 0& PWM Channel 2
19	P0.4/ADC4/CMP0PIn	I/O	Bit 4 of port 0 & ADC input channel 4 & Cmp0 Positive Input
20	P0.3/KBI3/T2/ADC3/CMP0NIn	I/O	Bit 3 of port 0 & KBI interrupt 3 & Timer 2 external input clock & ADC input channel 3 & Cmp0 Negative Input

Special Function Register (SFR)

A map of the Special Function Registers is shown as below：

Hex\Bin	X000	X001	X010	X011	X100	X101	X110	X111	Bin/Hex
F8	IICS	IICCTL	IICA1	IICA2	IICRWD	IICEBT	CMP0CON	CMP1CON	FF
F0	B	SPIC1	SPIC2	SPITXD	SPIRXD	SPIS	OPPIN	TAKEY	F7
E8				ISPFDH			INTDEG	ADCSH	EF
E0	ACC	ISPFAH	ISPFAL	ISPFDL	ISPFC	ENHIT	LVC	SWRES	E7
D8		PFCON	P3M0	P3M1					DF
D0	PSW	CCEN2	P0M0	P0M1	P1M0	P1M1			D7
C8	T2CON	CCCON	CRCL	CRCH	TL2	TH2	PWMMDH	PWMMDL	CF
C0	IRCON	CCEN	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3	C7
B8	IEN1	IP1	SRELH		PWMD0H	PWMD0L	PWMD1H	PWMD1L	BF
B0	P3	PWMD2H	PWMD2L	PWMD3H	PWMD3L	PWMC	WDTC	WDTK	B7
A8	IEN0	IP0	SRELL	ADCC1	ADCC2	ADCDH	ADCDL	ADCCS	AF
A0		RSTS							A7
98	SCON	SBUF	IEN2						9F
90	P1	AUX		KBLS	KBE	KBF	KBD	IRCON2	97
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	IFCON	8F
80	P0	SP	DPL	DPH	DPL1	DPH1		PCON	87
Hex\Bin	X000	X001	X010	X011	X100	X101	X110	X111	Bin/Hex

Note: Special Function Registers reset values and description for OB38S003

Register	Location	Reset value	Description
SYSTEM			
SP	81h	07h	Stack Pointer
ACC	E0h	00h	Accumulator
PSW	D0h	00h	Program Status Word
B	F0h	00h	B Register
DPL	82h	00h	Data Pointer 0 low byte
DPH	83h	00h	Data Pointer 0 high byte
DPL1	84h	00h	Data Pointer 1 low byte
DPH1	85h	00h	Data Pointer 1 high byte
AUX	91h	00h	Auxiliary register
PCON	87h	00h	Power Control
CKCON	8Eh	10h	Clock control register
INTERRUPT & PRIORITY			
IRCON	C0h	00h	Interrupt Request Control Register
IRCON2	97h	00h	Interrupt Request Control Register 2
IEN0	A8h	00h	Interrupt Enable Register 0
IEN1	B8h	00h	Interrupt Enable Register 1

Register	Location	Reset value	Description
IEN2	9Ah	00h	Interrupt Enable Register 2
IP0	A9h	00h	Interrupt Priority Register 0
IP1	B9h	00h	Interrupt Priority Register 1
ENHIT	E5h	07h	ENHance Interrupt Type Register
INTDEG	EEh	00h	External Interrupt Deglitch Register
KBI			
KBLS	93h	00h	Keyboard level selector Register
KBE	94h	00h	Keyboard input enable Register
KBF	95h	00h	Keyboard interrupt flag Register
KBD	96h	00h	Keyboard interface De-bounce control register
UART			
PCON	87h	00h	Power Control
AUX	91h	00h	Auxiliary register
SCON	98h	00h	Serial Port, Control Register
SBUF	99h	00h	Serial Port, Data Buffer
SRELL	AAh	00h	Serial Port, Reload Register, low byte
SRELH	BAh	00h	Serial Port, Reload Register, high byte
PFCON	D9h	00h	Peripheral Frequency control register
ADC			
ADCC1	ABh	00h	SADC Control 1 Register
ADCC2	ACh	0Fh	SADC Control 2 Register
ADCDH	ADh	00h	SADC data high byte
ADCDL	AEh	00h	SADC data low byte
ADCCS	AFh	00h	SADC clock select
ADCSH	EFh	00h	SADC Sample and Hold Time
WDT			
RSTS	A1h	00h	Reset status register
WDTC	B6h	04h	Watchdog timer control register
WDTK	B7h	00h	Watchdog timer refresh key.
TAKEY	F7h	00h	Time Access Key register
PWM			
PWMC	B5h	00h	PWM control register
PWMD0H	BCh	00h	PWM channel 0 data high byte
PWMD0L	BDh	00h	PWM channel 0 data low byte
PWMD1H	BEh	00h	PWM channel 1 data high byte
PWMD1L	BFh	00h	PWM channel 1 data low byte
PWMD2H	B1h	00h	PWM channel 2 data high byte
PWMD2L	B2h	00h	PWM channel 2 data low byte
PWMD3H	B3h	00h	PWM channel 3 data high byte
PWMD3L	B4h	00h	PWM channel 3 data low byte
PWMMDH	CEh	00h	PWM Max Data Register, high byte.

Register	Location	Reset value	Description
PWMMDL	CFh	FFh	PWM Max Data Register, low byte.
TIMER0/TIMER1			
TCON	88h	00h	Timer/Counter Control
TMOD	89h	00h	Timer Mode Control
TL0	8Ah	00h	Timer 0, low byte
TL1	8Bh	00h	Timer 1, low byte
TH0	8Ch	00h	Timer 0, high byte
TH1	8Dh	00h	Timer 1, high byte
PFCON	D9h	00h	Peripheral Frequency control register
PCA(TIMER2)			
CCEN	C1h	00h	Compare/Capture Enable Register
CCL1	C2h	00h	Compare/Capture Register 1, low byte
CCH1	C3h	00h	Compare/Capture Register 1, high byte
CCL2	C4h	00h	Compare/Capture Register 2, low byte
CCH2	C5h	00h	Compare/Capture Register 2, high byte
CCL3	C6h	00h	Compare/Capture Register 3, low byte
CCH3	C7h	00h	Compare/Capture Register 3, high byte
T2CON	C8h	00h	Timer 2 Control
CCCON	C9h	00h	Compare/Capture Control
CRCL	CAh	00h	Compare/Reload/Capture Register, low byte
CRCH	CBh	00h	Compare/Reload/Capture Register, high byte
TL2	CCh	00h	Timer 2, low byte
TH2	CDh	00h	Timer 2, high byte
CCEN2	D1h	00h	Compare/Capture Enable 2 register
GPIO			
P0	80h	FFh	Port 0
P1	90h	FFh	Port 1
P3	B0h	FFh	Port 3
P0M0	D2h	00h	Port 0 output mode 0
P0M1	D3h	00h	Port 0 output mode 1
P1M0	D4h	00h	Port 1 output mode 0
P1M1	D5h	00h	Port 1 output mode 1
P3M0	DAh	00h	Port 3 output mode 0
P3M1	DBh	00h	Port 3 output mode 1
ISP/IAP/EEPROM			
IFCON	8Fh	00h	Interface control register
ISPFAH	E1h	FFh	ISP Address-High register
ISPFAL	E2h	FFh	ISP Address-Low register
ISPFDL	E3h	FFh	ISP Data High register
ISPFDH	E3h	FFh	ISP Data Low register
ISPFC	E4h	00h	ISP control register

Register	Location	Reset value	Description
TAKEY	F7h	00h	Time Access Key register
LVI/LVR/SOFTRESET			
RSTS	A1h	00h	Reset status register
LVC	E6h	20h	Low voltage control register
SWRES	E7h	00h	Software Reset register
TAKEY	F7h	00h	Time Access Key register
SPI			
AUX	91h	00h	Auxiliary register
SPIC1	F1h	08h	SPI control register 1
SPIC2	F2h	00h	SPI control register 2
SPITXD	F3h	00h	SPI transmit data buffer
SPIRXD	F4h	00h	SPI receive data buffer
SPIS	F5h	40h	SPI status register
IIC			
AUX	91h	00h	Auxiliary register
IICS	F8h	00h	IIC status register
IICCTL	F9h	04h	IIC control register
IICA1	FAh	A0h	IIC channel 1 Address 1 register
IICA2	FBh	60h	IIC channel 1 Address 2 register
IICRWD	FCh	00h	IIC channel 1 Read / Write Data buffer
IICEBT	FDh	00h	IIC Enable Bus Transaction register
OPA			
OPPIN	F6h	00H	Comparator Pin Select register
CMP0CON	FEh	00h	Comparator 0 Control register
CMP1CON	FFh	00h	Comparator 1 Control register

Function Description

1. General Features

OB38S003 is an 8-bit micro-controller. All of its functions and the detailed meanings of SFR will be given in the following sections.

1.1 Embedded Programmable ROM

The program can be loaded into the embedded 8KB programmable ROM via its writer or In-System Programming (ISP).

1.2 IO Pads

The OB38S003 has Three I/O ports: Port 0, Port 1 and Port 3. Ports 0, 1 are 8-bit ports and Port 3 is a 2-bit port. These are: quasi-bidirectional (standard 8051 port outputs), push-pull, open drain, and input-only. As description in section 5.

All the pads for P0, P1 and P3 are with slew rate to reduce EMI. The IO pads can withstand 4KV ESD in human body mode guaranteeing the OB38S003 is quality in high electro-static environments.

The RESET Pin can define as General I/O P1.5 when user use Internal RESET.

The XTAL2 and XTAL1 can define as P3.0 and P3.1 by writer or ISP, when user use internal OSC as system clock; when user use external OSC as system clock and input into XTAL1, only XTAL2 can be defined as P3.0.

1.3 Instruction timing Selection

The conventional 52-series MCUs are 12T, i.e., 12 oscillator clocks per machine cycle. OB38S003 is a 1T to 8T MCU, i.e., its machine cycle is one-clock to eight-clock. In the other words, it can execute one instruction within one clock to only eight clocks.

Mnemonic: CKCON **Address: 8Eh**

7	6	5	4	3	2	1	0	Reset
-	ITS[2:0]			-	-	CLKOUT[1:0]		10H

ITS: Instruction timing select.

ITS [2:0]	Instruction timing
000	1T mode
001	2T mode (default)
010	3T mode
011	4T mode
100	5T mode
101	6T mode
110	7T mode
111	8T mode

The default is in 2T mode, and it can be changed to another Instruction timing mode if CKCON [6:4] (at address 8Eh) is change any time. Not every instruction can be executed with one machine cycle. The exact machine cycle number for all the instructions are given in the next section.

1.4 The Clock Out Selection

The OB38S003 can Generator a clock out signal at P3.0, when user use Oscillator (XTAL1 as clock input) or internal OSC as system clock. The CKCON [1:0] (at address 8Eh) can change any time.

CLKOUT: Clock output select.

CKCON [1:0]	Mode.
00	GPIO(default)
01	Fosc
10	Fosc/2
11	Fosc/4

1.5 RESET

1.5.1 Hardware RESET function

OB38S003 provides Internal reset circuit inside, the Internal reset time can set by writer or ISP.

Internal Reset time
25ms (default)
200ms
100ms
50ms
16ms
8ms
4ms

1.5.2 Software RESET function

OB38S003 provides one software reset mechanism to reset whole chip. To perform a software reset, the firmware must write three specific values 55h, AAh and 5Ah sequentially to the TAKEY register to enable the Software Reset register (SWRES) write attribute. After SWRES register obtain the write authority, the firmware can write FFh to the SWRES register. The hardware will decode a reset signal that "OR" with the other hardware reset. The SWRES register is self-reset at the end of the software reset procedure.

Mnemonic	Description	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
Software Reset function											
RSTS	Reset status register	A1h	-	LVRLP INTF	LVRL PF	PDRF	WDTF	SWRF	LVRF	PORF	00H
TAKEY	Time Access Key register	F7h	TAKEY [7:0]								00H
SWRES	Software Reset register	E7h	SWRES [7:0]								00H

1.5.3 Reset status

Mnemonic: RSTS **Address: A1h**

7	6	5	4	3	2	1	0	Reset
-	LVRL PINTF	LVRLP F	PDRF	WDTF	SWRF	LVRF	PORF	00H

LVRLPINTF: “Internal” Low voltage reset flag.

When MCU is reset by LVR_LP_INT, LVRLPINTF flag will be set to one by hardware.

This flag clear by software.

LVRLPF: Low voltage reset(Low Power) flag.

When MCU is reset by LVR(Low Power), LVRLPF flag will be set to one by hardware.

This flag clear by software.

PDRF: Pad reset flag.

When MCU is reset by reset pad, PDRF flag will be set to one by hardware. This flag clear by software.

WDTF: Watchdog timer reset flag.

When MCU is reset by watchdog, WDTF flag will be set to one by hardware. This flag clear by software.

SWRF: Software reset flag.

When MCU is reset by software, SWRF flag will be set to one by hardware. This flag clear by software.

LVRF: Low voltage reset flag.

When MCU is reset by LVR, LVRF flag will be set to one by hardware. This flag clear by software.

PORF: Power on reset flag.

When MCU is reset by POR, PORF flag will be set to one by hardware. This flag clear by software.

1.5.4 Time Access Key register (TAKEY)

Mnemonic: TAKEY **Address: F7H**

7	6	5	4	3	2	1	0	Reset
TAKEY [7:0]								00H

Software reset register (SWRES) is read-only by default; software must write three specific values 55h, AAh and 5Ah sequentially to the TAKEY register to enable the SWRES register write attribute. That is:

```
MOV TAKEY, #55h
MOV TAKEY, #0AAh
MOV TAKEY, #5Ah
```

1.5.5 Software Reset register (SWRES)

Mnemonic: SWRES **Address: E7H**

7	6	5	4	3	2	1	0	Reset
SWRES [7:0]								00H

SWRES[7:0]: Software reset register bit. These 8-bit is self-reset at the end of the reset procedure.

SWRES [7:0] = FFh, software reset.

SWRES [7:0] = 00h ~ FEh, MCU no action.

1.5.6 Example of software reset

```
MOV TAKEY, #55h
MOV TAKEY, #0AAh
MOV TAKEY, #5Ah; enable SWRES write attribute
MOV SWRES, #0FFh ; software reset MCU
```

1.6 Clocks

The default clock is the 16MHz Internal OSC. This clock is used during the initialization stage. The major work of the initialization stage is to determine the clock source used in normal operation.

The internal clock sources are from the internal OSC with difference frequency division as given in Table 1-1，the clock source can set by writer or ICP.

Table 1-1: Selection of clock source

Clock source
external crystal (use XTAL1 and XTAL2 pins)
external crystal/2 (use XTAL1 and XTAL2 pins)
external crystal/4 (use XTAL1 and XTAL2 pins)
external crystal (only use XTAL1, the XTAL2 define as I/O)
16MHz from internal OSC
8MHz from internal OSC
4MHz from internal OSC
2MHz from internal OSC
1MHz from internal OSC

There may be having a little variance in the frequency from the internal OSC. The max variance as giving in Table 1-2

Table 1-2: Temperature with variance

Temperature	Max Variance
25℃	±2%

2. Instruction Set

All OB38S003 instructions are binary code compatible and perform the same functions as they do with the industry standard 8051. The following tables give a summary of the instruction set cycles of the OB38S003 Microcontroller core.

Table 2-1: Arithmetic operations

Mnemonic	Description	Code	Bytes	Cycles
ADD A,Rn	Add register to accumulator	28-2F	1	1
ADD A,direct	Add direct byte to accumulator	25	2	2
ADD A,@Ri	Add indirect RAM to accumulator	26-27	1	2
ADD A,#data	Add immediate data to accumulator	24	2	2
ADDC A,Rn	Add register to accumulator with carry flag	38-3F	1	1
ADDC A,direct	Add direct byte to A with carry flag	35	2	2
ADDC A,@Ri	Add indirect RAM to A with carry flag	36-37	1	2
ADDC A,#data	Add immediate data to A with carry flag	34	2	2
SUBB A,Rn	Subtract register from A with borrow	98-9F	1	1
SUBB A,direct	Subtract direct byte from A with borrow	95	2	2
SUBB A,@Ri	Subtract indirect RAM from A with borrow	96-97	1	2
SUBB A,#data	Subtract immediate data from A with borrow	94	2	2
INC A	Increment accumulator	04	1	1
INC Rn	Increment register	08-0F	1	2
INC direct	Increment direct byte	05	2	3
INC @Ri	Increment indirect RAM	06-07	1	3
INC DPTR	Increment data pointer	A3	1	1
DEC A	Decrement accumulator	14	1	1
DEC Rn	Decrement register	18-1F	1	2
DEC direct	Decrement direct byte	15	2	3
DEC @Ri	Decrement indirect RAM	16-17	1	3
MUL AB	Multiply A and B	A4	1	5
DIV	Divide A by B	84	1	5
DA A	Decimal adjust accumulator	D4	1	1

Table 2-2: Logic operations

Mnemonic	Description	Code	Bytes	Cycles
ANL A,Rn	AND register to accumulator	58-5F	1	1
ANL A,direct	AND direct byte to accumulator	55	2	2
ANL A,@Ri	AND indirect RAM to accumulator	56-57	1	2
ANL A,#data	AND immediate data to accumulator	54	2	2
ANL direct,A	AND accumulator to direct byte	52	2	3
ANL direct,#data	AND immediate data to direct byte	53	3	4
ORL A,Rn	OR register to accumulator	48-4F	1	1
ORL A,direct	OR direct byte to accumulator	45	2	2
ORL A,@Ri	OR indirect RAM to accumulator	46-47	1	2
ORL A,#data	OR immediate data to accumulator	44	2	2
ORL direct,A	OR accumulator to direct byte	42	2	3
ORL direct,#data	OR immediate data to direct byte	43	3	4
XRL A,Rn	Exclusive OR register to accumulator	68-6F	1	1
XRL A,direct	Exclusive OR direct byte to accumulator	65	2	2
XRL A,@Ri	Exclusive OR indirect RAM to accumulator	66-67	1	2
XRL A,#data	Exclusive OR immediate data to accumulator	64	2	2
XRL direct,A	Exclusive OR accumulator to direct byte	62	2	3
XRL direct,#data	Exclusive OR immediate data to direct byte	63	3	4
CLR A	Clear accumulator	E4	1	1
CPL A	Complement accumulator	F4	1	1
RL A	Rotate accumulator left	23	1	1
RLC A	Rotate accumulator left through carry	33	1	1
RR A	Rotate accumulator right	03	1	1
RRC A	Rotate accumulator right through carry	13	1	1
SWAP A	Swap nibbles within the accumulator	C4	1	1

Table 2-3: Data transfer

Mnemonic	Description	Code	Bytes	Cycles
MOV A,Rn	Move register to accumulator	E8-EF	1	1
MOV A,direct	Move direct byte to accumulator	E5	2	2
MOV A,@Ri	Move indirect RAM to accumulator	E6-E7	1	2
MOV A,#data	Move immediate data to accumulator	74	2	2
MOV Rn,A	Move accumulator to register	F8-FF	1	2
MOV Rn,direct	Move direct byte to register	A8-AF	2	4
MOV Rn,#data	Move immediate data to register	78-7F	2	2
MOV direct,A	Move accumulator to direct byte	F5	2	3
MOV direct,Rn	Move register to direct byte	88-8F	2	3
MOV direct1,direct2	Move direct byte to direct byte	85	3	4
MOV direct,@Ri	Move indirect RAM to direct byte	86-87	2	4
MOV direct,#data	Move immediate data to direct byte	75	3	3
MOV @Ri,A	Move accumulator to indirect RAM	F6-F7	1	3
MOV @Ri,direct	Move direct byte to indirect RAM	A6-A7	2	5
MOV @Ri,#data	Move immediate data to indirect RAM	76-77	2	3
MOV DPTR,#data16	Load data pointer with a 16-bit constant	90	3	3
MOVC A,@A+DPTR	Move code byte relative to DPTR to accumulator	93	1	3
MOVC A,@A+PC	Move code byte relative to PC to accumulator	83	1	3
MOVX A,@Ri	Move external RAM (8-bit addr.) to A	E2-E3	1	3
MOVX A,@DPTR	Move external RAM (16-bit addr.) to A	E0	1	3
MOVX @Ri,A	Move A to external RAM (8-bit addr.)	F2-F3	1	4
MOVX @DPTR,A	Move A to external RAM (16-bit addr.)	F0	1	4
PUSH direct	Push direct byte onto stack	C0	2	4
POP direct	Pop direct byte from stack	D0	2	3
XCH A,Rn	Exchange register with accumulator	C8-CF	1	2
XCH A,direct	Exchange direct byte with accumulator	C5	2	3
XCH A,@Ri	Exchange indirect RAM with accumulator	C6-C7	1	3
XCHD A,@Ri	Exchange low-order nibble indir. RAM with A	D6-D7	1	3

Table 2-4: Program branches

Mnemonic	Description	Code	Bytes	Cycles
ACALL addr11	Absolute subroutine call	xxx11	2	6
LCALL addr16	Long subroutine call	12	3	6
RET	from subroutine	22	1	4
RETI	from interrupt	32	1	4
AJMP addr11	Absolute jump	xxx01	2	3
LJMP addr16	Long iump	02	3	4
SJMP rel	Short jump (relative addr.)	80	2	3
JMP @A+DPTR	Jump indirect relative to the DPTR	73	1	2
JZ rel	Jump if accumulator is zero	60	2	3
JNZ rel	Jump if accumulator is not zero	70	2	3
JC rel	Jump if carry flag is set	40	2	3
JNC	Jump if carry flag is not set	50	2	3
JB bit,rel	Jump if direct bit is set	20	3	4
JNB bit,rel	Jump if direct bit is not set	30	3	4
JBC bit,direct rel	Jump if direct bit is set and clear bit	10	3	4
CJNE A,direct rel	Compare direct byte to A and jump if not equal	B5	3	4
CJNE A,#data rel	Compare immediate to A and jump if not equal	B4	3	4
CJNE Rn,#data rel	Compare immed. to reg. and jump if not equal	B8-BF	3	4
CJNE @Ri,#data rel	Compare immed. to ind. and jump if not equal	B6-B7	3	4
DJNZ Rn,rel	Decrement register and jump if not zero	D8-DF	2	3
DJNZ direct,rel	Decrement direct byte and jump if not zero	D5	3	4
NOP	No operation	00	1	1

Table 2-5: Boolean manipulation

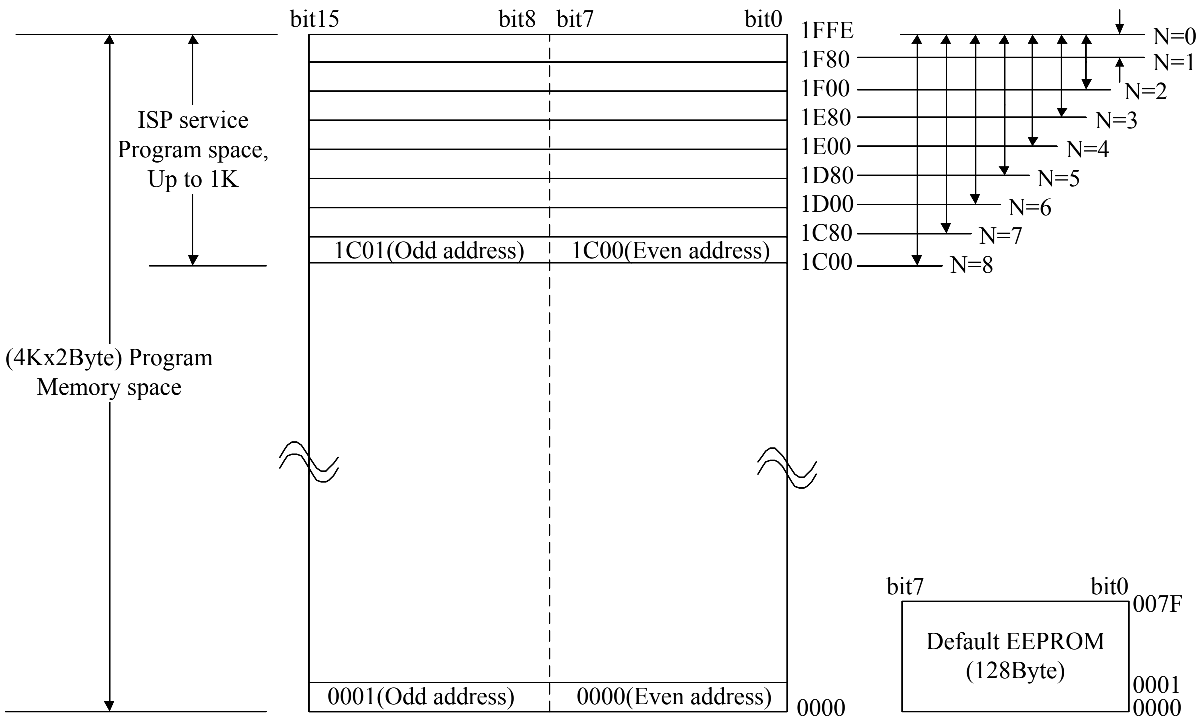
Mnemonic	Description	Code	Bytes	Cycles
CLR C	Clear carry flag	C3	1	1
CLR bit	Clear direct bit	C2	2	3
SETB C	Set carry flag	D3	1	1
SETB bit	Set direct bit	D2	2	3
CPL C	Complement carry flag	B3	1	1
CPL bit	Complement direct bit	B2	2	3
ANL C,bit	AND direct bit to carry flag	82	2	2
ANL C,/bit	AND complement of direct bit to carry	B0	2	2
ORL C,bit	OR direct bit to carry flag	72	2	2
ORL C,/bit	OR complement of direct bit to carry	A0	2	2
MOV C,bit	Move direct bit to carry flag	A2	2	2
MOV bit,C	Move carry flag to direct bit	92	2	3

3. Memory Structure

The OB38S003 memory structure follows general 8052 structure. It is 8KB program memory.

3.1 Program Memory

The OB38S003 has 8KB on-chip memory which can be used as general program memory, on which include up to 1K byte specific ISP service program memory space. The address range for the 8K byte is $0000 to $1FFF. The address range for the ISP service program is $1C00 to $1FFF. The ISP service program size can be partitioned as N blocks of 128 byte (N=0 to 8). When N=0 means no ISP service program space available, total 8KB memory used as program memory. When N=1 means address $1F80 to $1FFF reserved for ISP service program. When N=2 means memory address $1F00 to $1FFF reserved for ISP service program…etc. Value N can be set and programmed into OB38S003 by the writer.



EEPROM Area
(10K-write cycle life)

Programmable ROM Area
(1K-write cycle life);
Main Program Memory

Fig. 3-1: OB38S003 ROM

3.2 Data Memory

The OB38S003 has 256 Bytes + 256Bytes on-chip SRAM, 256 Bytes of it are the same as general 8052 internal memory structure while the expanded 256Bytes on-chip SRAM can be accessed by external memory addressing method (by instruction MOVX.).

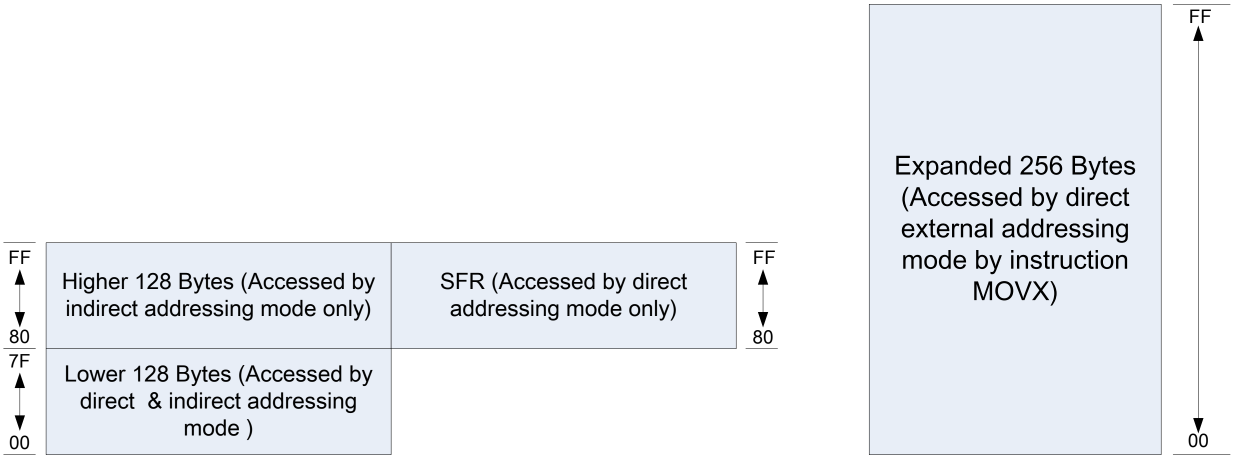

FF
80
7F
00
Higher 128 Bytes (Accessed by indirect addressing mode only)
SFR (Accessed by direct addressing mode only)
Lower 128 Bytes (Accessed by direct & indirect addressing mode)
FF
80
Expanded 256 Bytes (Accessed by direct external addressing mode by instruction MOVX)
FF
00

Fig. 3-2: RAM architecture

3.3 Data memory - lower 128 byte (00h to 7Fh)

Data memory 00h to FFh is the same as 8052.
The address 00h to 7Fh can be accessed by direct and indirect addressing modes.
Address 00h to 1Fh is register area.
Address 20h to 2Fh is memory bit area.
Address 30h to 7Fh is for general memory area.

3.4 Data memory - higher 128 byte (80h to FFh)

The address 80h to FFh can be accessed by indirect addressing mode.
Address 80h to FFh is data area.

3.5 Data memory - Expanded 256 bytes ($00 to $FF)

From external address 00h to FFh is the on-chip expanded SRAM area, total 256 Bytes. This area can be accessed by external direct addressing mode (by instruction MOVX).

4. CPU Engine

The OB38S003 engine is composed of four components:

(1) Control unit
(2) Arithmetic – logic unit
(3) Memory control unit
(4) RAM and SFR control unit

The OB38S003 engine allows to fetch instruction from program memory and to execute using RAM or SFR. The following chapter describes the main engine register.

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
8051 Core											
ACC	Accumulator	E0h	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00H
B	B register	F0h	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	00H
PSW	Program status word	D0h	CY	AC	F0	RS[1:0]		OV	PSW.1	P	00H
SP	Stack Pointer	81h	SP[7:0]								07H
DPL	Data pointer low 0	82h	DPL[7:0]								00H
DPH	Data pointer high 0	83h	DPH[7:0]								00H
DPL1	Data pointer low 0	84h	DPL1[7:0]								00H
DPH1	Data pointer high 0	85h	DPH1[7:0]								00H
AUX	Auxiliary register	91h	BRGS	-	-	-	-	-	-	DPS	00H
CKCON	Clock control register	8Eh	-	ITS[2:0]					CLKOUT[1:0]		10H
IFCON	Interface control register	8Fh	-	CDPR	-	-	-	-	-	ISPE	00H

4.1 Accumulator

ACC is the Accumulator register. Most instructions use the accumulator to store the operand.

Mnemonic: ACC **Address: E0h**

7	6	5	4	3	2	1	0	Reset
ACC.7	ACC.6	ACC05	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0	00h

ACC[7:0]: The A (or ACC) register is the standard 8052 accumulator.

4.2 B Register

The B register is used during multiply and divide instructions. It can also be used as a scratch pad register to store temporary data.

Mnemonic: B **Address: F0h**

7	6	5	4	3	2	1	0	Reset
B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0	00h

B[7:0]: The B register is the standard 8052 register that serves as a second accumulator.

4.3 Program Status Word

Mnemonic: PSW **Address: D0h**

7	6	5	4	3	2	1	0	Reset
CY	AC	F0	RS [1:0]		OV	F1	P	00h

CY: Carry flag.

AC: Auxiliary Carry flag for BCD operations.

F0: General purpose Flag 0 available for user.

RS[1:0]	Bank Selected	Location
00	Bank 0	00h – 07h
01	Bank 1	08h – 0Fh
10	Bank 2	10h – 17h
11	Bank 3	18h – 1Fh

OV: Overflow flag.

F1: General purpose Flag 1 available for user.

P: Parity flag, affected by hardware to indicate odd/even number of “one” bits in the Accumulator, i.e. even parity

4.4 Stack Pointer

The stack pointer is a 1-byte register initialized to 07h after reset. This register is incremented before PUSH and CALL instructions, causing the stack to start from location 08h.

Mnemonic: SP **Address: 81h**

7	6	5	4	3	2	1	0	Reset
SP [7:0]								07h

SP[7:0]: The Stack Pointer stores the scratchpad RAM address where the stack begins. In other words, it always points to the top of the stack.

4.5 Data Pointer

The data pointer (DPTR) is 2-bytes wide. The lower part is DPL, and the highest is DPH. It can be loaded as a 2-byte register (e.g. MOV DPTR, #data16) or as two separate registers (e.g. MOV DPL,#data8). It is generally used to access the external code or data space (e.g. MOVC A, @A+DPTR, @DPTR respectively).

Mnemonic: DPL **Address: 82h**

7	6	5	4	3	2	1	0	Reset
DPL [7:0]								00h

DPL[7:0]: Data pointer Low 0

Mnemonic: DPH **Address: 83h**

7	6	5	4	3	2	1	0	Reset
DPH [7:0]								00h

DPH [7:0]: Data pointer High 0

4.6 Data Pointer 1

The Dual Data Pointer accelerates the moves of data block. The standard DPTR is a 16-bit register that is used to address external memory or peripherals. In the OB38S003 core the standard data pointer is called DPTR, the second data pointer is called DPTR1. The data pointer select bit chooses the active pointer. The data pointer select bit is located in LSB of AUX register (DPS).

The user switches between pointers by toggling the LSB of AUX register. All DPTR-related instructions use the currently selected DPTR for any activity.

Mnemonic: DPL1 **Address: 84h**

7	6	5	4	3	2	1	0	Reset
DPL1 [7:0]								00h

DPL1[7:0]: Data pointer Low 1

Mnemonic: DPH1 **Address: 85h**

7	6	5	4	3	2	1	0	Reset
DPH1 [7:0]								00h

DPH1[7:0]: Data pointer High 1

Mnemonic: AUX **Address: 91h**

7	6	5	4	3	2	1	0	Reset
BRGS	-	-	-	-	-	-	DPS	00H

DPS: Data Pointer select register.

DPS = 1 is selected DPTR1.

4.7 Clock control register

Mnemonic: CKCON **Address: 8Eh**

7	6	5	4	3	2	1	0	Reset
-	ITS[2:0]			-	-	CLKOUT[1:0]		10H

ITS[2:0]: Instruction timing select.

ITS [2:0]	Mode
000	1T mode
001	2T mode (default)
010	3T mode
011	4T mode
100	5T mode
101	6T mode
110	7T mode
111	8T mode

CLKOUT: Clock output select.

CKCON [1:0]	Mode.
00	GPIO(default)
01	Fosc
10	Fosc/2
11	Fosc/4

It can be used when the system clock is the internal RC oscillator.

4.8 Interface control register

Mnemonic: IFCON **Address: 8Fh**

7	6	5	4	3	2	1	0	Reset
-	CDPR	-	-	-	-	-	ISPE	00H

CDPR: Code protect (Read Only)

ISPE: ISP function enable bit

ISPE = 1, enable ISP function

ISPE = 0, disable ISP function

5. GPIO

5.1 SFR Setting Method

The OB38S003 has three I/O ports: Port 0, Port 1 and Port 3. Ports 0, 1 are 8-bit ports and Port 3 is a 2-bit port. These are: quasi-bidirectional (standard 8051 port outputs), push-pull, open drain, and input-only. Two configuration registers for each port select the output type for each port pin. All I/O port pins on the OB38S003 may be configured by software to one of four types on a pin-by-pin basis, shown as below:

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
I/O port function register											
P0M0	Port 0 output mode 0	D2h	P0M0 [7:0]								00H
P0M1	Port 0 output mode 1	D3h	P0M1[7:0]								00H
P1M0	Port 1 output mode 0	D4h	P1M0[7:0]								00H
P1M1	Port 1 output mode 1	D5h	P1M1[7:0]								00H
P3M0	Port 3 output mode 0	DAh							P3M0[1:0]		00H
P3M1	Port 3 output mode 1	DBh							P3M1[1:0]		00H

PxM1.y	PxM0.y	Port output mode
0	0	Quasi-bidirectional (standard 8051 port outputs) (pull-up)
0	1	Push-pull
1	0	Input only (high-impedance)
1	1	Open drain

The RESET Pin can define as General I/O P1.5 when user use Internal RESET.

The XTAL2 and XTAL1 can define as P3.0 and P3.1 by writer or ISP, when user use internal OSC as system clock; when user use external OSC as system clock and input into XTAL1, only XTAL2 can be defined as P3.0.

For general-purpose applications, every pin can be assigned to either high or low independently as given below:

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
Ports											
Port 3	Port 3	B0h	-	-	-	-	-	-	P3.1	P3.0	FFh
Port 1	Port 1	90h	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	FFh
Port 0	Port 0	80h	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	FFh

Mnemonic: P0 **Address: 80h**

7	6	5	4	3	2	1	0	Reset
P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	FFh

P0.7~ 0: Port0 [7] ~ Port0[0]

Mnemonic: P1 **Address: 90h**

7	6	5	4	3	2	1	0	Reset
P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	FFh

P1.7~ 0: Port1 [7] ~ Port1 [0]

Mnemonic: P3 **Address: B0h**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	P3.1	P3.0	FFh

P3.1~ 0: Por3 [1] ~ Port3 [0]

5.2 Software of Writer Setting Method

Please Setting the “IO Output Mode" item in the "Configuration" window, it can change the I/O mode of P0~P3 to the “Quasi-bidirectional (standard 8051 port outputs) (pull-up)” or Input only (high-impedance) mode, When MCU after reset and initial.

6. Timer 0 and Timer 1

The OB38S003 has three 16-bit timer/counter registers: Timer 0, Timer 1 and Timer 2. All can be configured for counter or timer operations.

In timer mode, the Timer 0 register or Timer 1 register is incremented every 1/12/96 machine cycles, which means that it counts up after every 1/12/96 periods of the clk signal. It's dependent on SFR(PFCON).

In counter mode, the register is incremented when the falling edge is observed at the corresponding input pin T0or T1. Since it takes 2 machine cycles to recognize a 1-to-0 event, the maximum input count rate is 1/2 of the oscillator frequency. There are no restrictions on the duty cycle, however to ensure proper recognition of 0 or 1 state, an input should be stable for at least 1 machine cycle.

Four operating modes can be selected for Timer 0 and Timer 1. Two Special Function registers (TMOD and TCON) are used to select the appropriate mode.

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
Timer 0 and 1											
TL0	Timer 0, low byte	8Ah	TL0[7:0]								00H
TH0	Timer 0, high byte	8Ch	TH0[7:0]								00H
TL1	Timer 1, lowbyte	8Bh	TL1[7:0]								00H
TH1	Timer 1, high byte	8Dh	TH1[7:0]								00H
TMOD	Timer Mode Control	89h	GATE	C/T	M1	M0	GATE	C/T	M1	M0	00H
TCON	Timer/Counter Control	88h	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00H
PFCON	Peripheral Frequency control register	D9h	-	-	SRELPS[1:0]		T1PS[1:0]		T0PS[1:0]		00H

6.1 Timer/counter mode control register (TMOD)

Mnemonic: TMOD **Address: 89h**

7	6	5	4	3	2	1	0	Reset
GATE	C/T	M1	M0	GATE	C/T	M1	M0	00h
Timer 1				Timer 0				

GATE: If set, enables external gate control (pin INT0 or INT1 for Counter 0 or 1, respectively). When INT0 or INT1 is high, and TRx bit is set (see TCON register), a counter is incremented every falling edge on T0 or T1 input pin.

C/T: Selects Timer or Counter operation. When set to 1, a counter operation is performed, when cleared to 0, the corresponding register will function as a timer.

M[1:0]: Selects mode for Timer/Counter 0 or Timer/Counter 1

M1	M0	Mode	Function
0	0	Mode0	13-bit counter/timer, with 5 lower bits in TL0 or TL1 register and 8 bits in TH0 or TH1 register (for Timer 0 and Timer 1, respectively). The 3 high order bits of TL0 and TL1 are hold at zero.
0	1	Mode1	16-bit counter/timer.
1	0	Mode2	8 -bit auto-reload counter/timer. The reload value is kept in TH0 or TH1, while TL0 or TL1 is incremented every machine cycle. When TLx overflows, a value from THx is copied to TLx.
1	1	Mode3	If Timer 1 M1 and M0 bits are set to 1, Timer 1 stops. If Timer 0 M1 and M0 bits are set to 1, Timer 0 acts as two independent 8 bit timers / counters.

6.2 Timer/counter control register (TCON)

Mnemonic: TCON **Address: 88h**

7	6	5	4	3	2	1	0	Reset
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	00h

TF1: Timer 1 overflow flag set by hardware when Timer 1 overflows. This flag can be cleared by software and is automatically cleared when interrupt is processed.

TR1: Timer 1 Run control bit. If cleared, Timer 1 stops.

TF0: Timer 0 overflow flag set by hardware when Timer 0 overflows. This flag can be cleared by software and is automatically cleared when interrupt is processed.

TR0: Timer 0 Run control bit. If cleared, Timer 0 stops.

IE1: Interrupt 1 edge flag. Set by hardware, when falling edge on external pin INT1 is observed. Cleared when interrupt is processed.

IT1: Interrupt 1 type control bit. Selects falling edge or low level on input pin to cause interrupt. IT1=1, interrupt 1 select falling edge trigger. IT1=0, interrupt1 select low level trigger.

IE0: Interrupt 0 edge flag. Set by hardware, when falling edge on external pin INT0 is observed. Cleared when interrupt is processed.

IT0: Interrupt 0 type control bit. Selects falling edge or low level on input pin to cause interrupt. IT0=1, interrupt 0 select falling edge trigger. IT0=0, interrupt 0 select low level trigger.

6.3 Peripheral Frequency control register

Mnemonic: PFCON **Address: D9h**

7	6	5	4	3	2	1	0	Reset
-	-	SRELPS[1:0]		T1PS[1:0]		T0PS[1:0]		00H

T1PS[1:0]: Timer1 Prescaler select

T1PS[1:0]	Prescaler
00	Fosc/12
01	Fosc
10	Fosc/96
11	reserved

T0PS[1:0]: Timer0 Prescaler select

T0PS[1:0]	Prescaler
00	Fosc/12
01	Fosc
10	Fosc/96
11	reserved

6.4 Mode 0 (13-bit Counter/Timer)

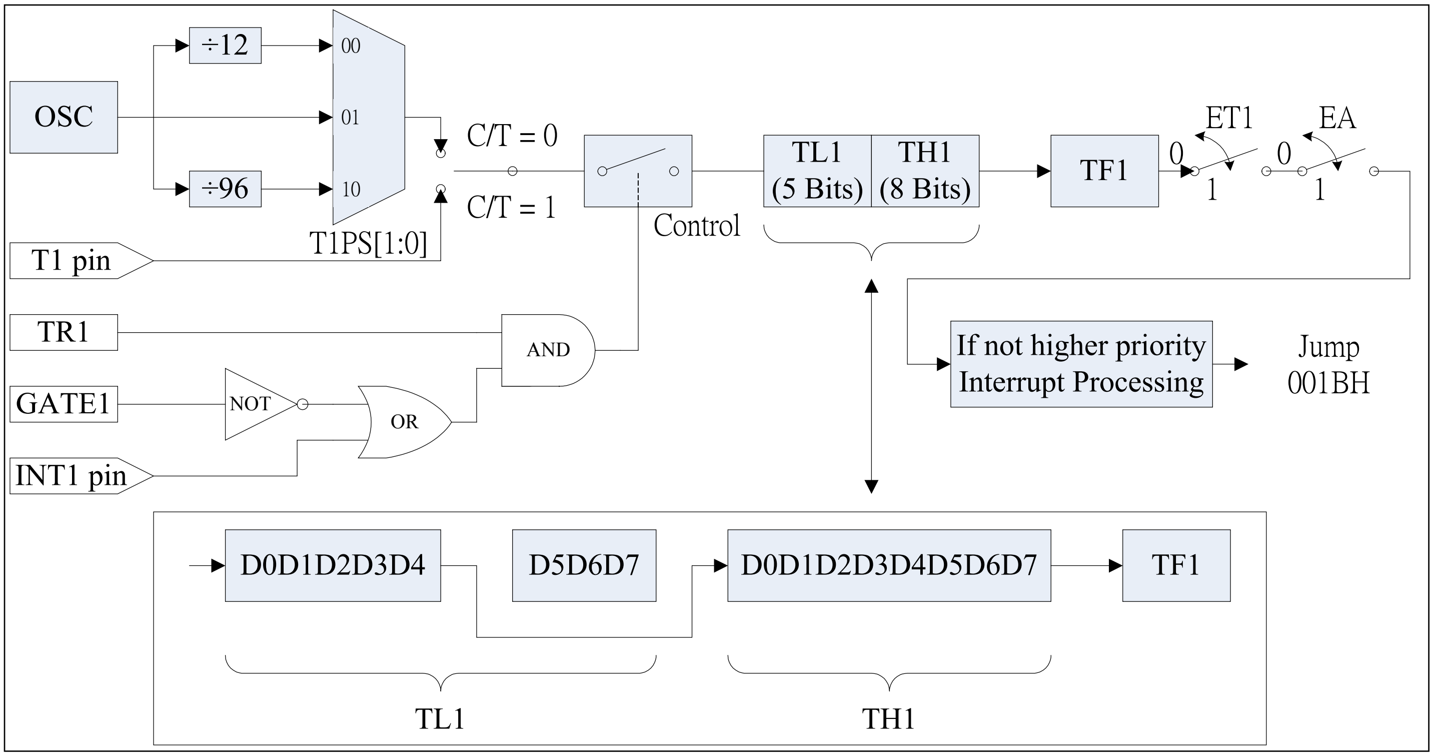


Fig. 6-1: Mode 0 -13 bit Timer / counter operation

6.5 Mode 1 (16-bit Counter/Timer)

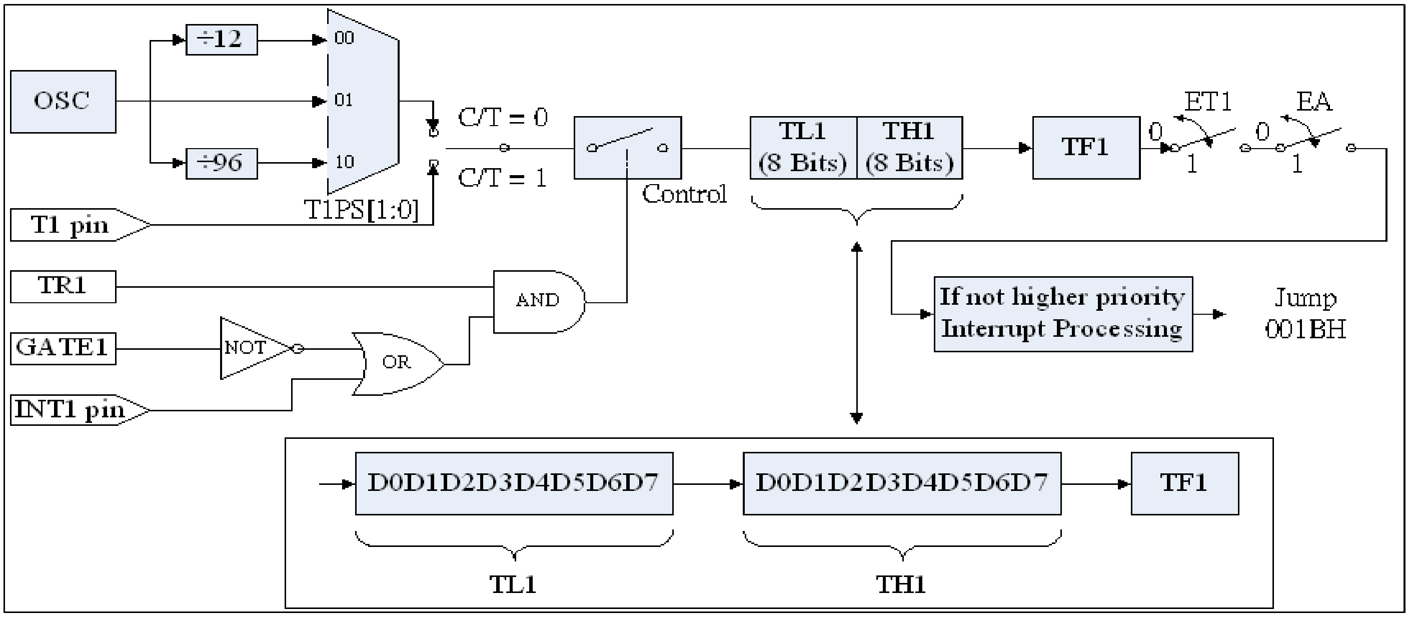


Fig. 6-2: Mode 1 -16 bit Timer / counter operation

6.6 Mode 2 (8-bit auto-reload Counter/Timer)

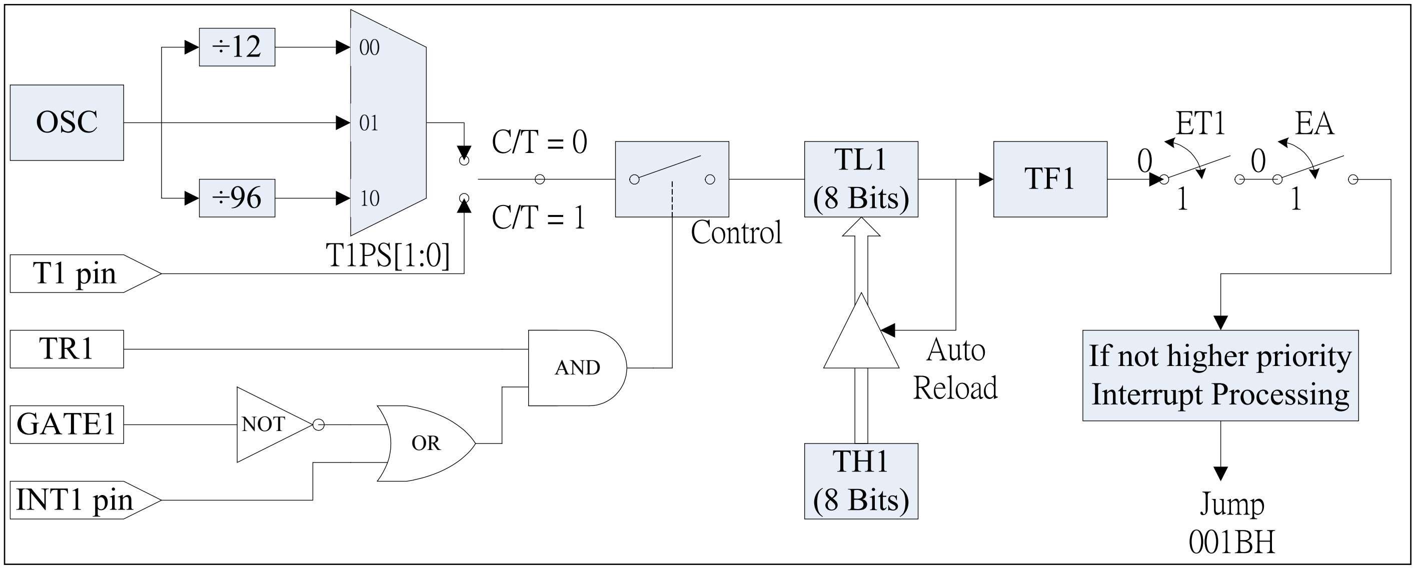


Fig. 6-3: Mode 2 8 bit Auto-reload Counter/Timer

6.7 Mode 3 (Timer 0 acts as two independent 8 bit Timers / Counters)

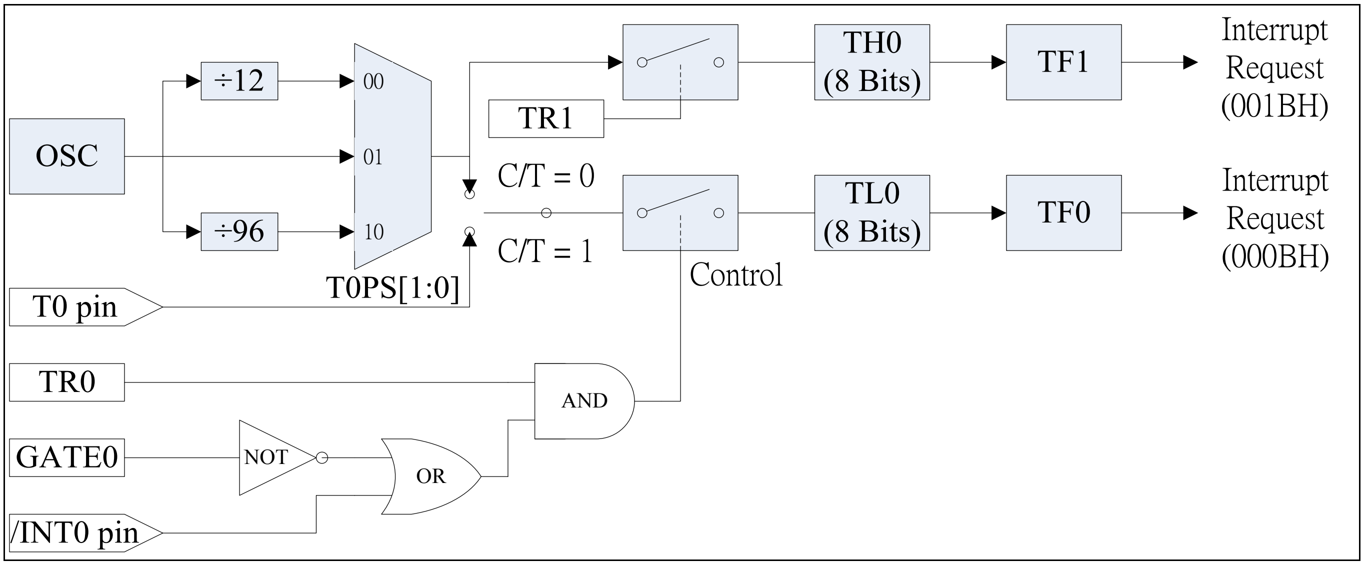


Fig. 6-4: Mode 3 - two independent 8 bit Timers / Counters (Only Timer 0)

7. Timer 2 and Capture Compare Unit

Timer 2 is not only a 16-bit timer, also a 4-channel unit with compare, capture and reload functions. It is very similar to the programmable counter array (PCA) in some other MCUs except pulse width modulation (PWM).

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
Timer 2 and Capture Compare Unit											
T2CON	Timer 2 control	C8h	T2PS[2:0]			T2R[1:0]		-	T2I[1:0]		00H
CCCON	Compare/Capture Control	C9h	CCI3	CCI2	CCI1	CCI0	CCF3	CCF2	CCF1	CCF0	00H
CCEN	Compare/Capture Enable register	C1h	-	COCAM1[2:0]			-	COCAM0[2:0]			00H
CCEN2	Compare/Capture Enable 2 register	D1h	-	COCAM3[2:0]			-	COCAM2[2:0]			00H
TL2	Timer 2, low byte	CCh	TL2[7:0]								00H
TH2	Timer 2, high byte	CDh	TH2[7:0]								00H
CRCL	Compare/Reload/ Capture register, low byte	CAh	CRCL[7:0]								00H
CRCH	Compare/Reload/ Capture register, high byte	CBh	CRCH[7:0]								00H
CCL1	Compare/Capture register 1, low byte	C2h	CCL1[7:0]								00H
CCH1	Compare/Capture register 1, high byte	C3h	CCH1[7:0]								00H
CCL2	Compare/Capture register 2, low byte	C4h	CCL2[7:0]								00H
CCH2	Compare/Capture register 2, high byte	C5h	CCH2[7:0]								00H
CCL3	Compare/Capture register 3, low byte	C6h	CCL3[7:0]								00H
CCH3	Compare/Capture register 3, high byte	C7h	CCH3[7:0]								00H

Mnemonic: T2CON **Address: C8h**

7	6	5	4	3	2	1	0	Reset
T2PS[2:0]			T2R[1:0]		-	T2I[1:0]		00H

T2PS[2:0]: Prescaler select bit:

T2PS = 000 – timer 2 is clocked with the oscillator frequency.

T2PS = 001 – timer 2 is clocked with 1/2 of the oscillator frequency.

T2PS = 010 – timer 2 is clocked with 1/4 of the oscillator frequency.

T2PS = 011 – timer 2 is clocked with 1/6 of the oscillator frequency.

T2PS = 100 – timer 2 is clocked with 1/8 of the oscillator frequency.

T2PS = 101 – timer 2 is clocked with 1/12 of the oscillator frequency.

T2PS = 110 – timer 2 is clocked with 1/24 of the oscillator frequency.

T2R[1:0]: Timer 2 reload mode selection

T2R[1:0] = 0X – Reload disabled

T2R[1:0] = 10 – Mode 0: Auto Reload

T2R[1:0] = 11 – Mode 1: T2EX Falling Edge Reload

T2I[1:0]: Timer 2 input selection

T2I[1:0] = 00 – Timer 2 stop

T2I[1:0] = 01 – Input frequency from prescaler (T2PS[2:0])

T2I[1:0] = 10 – Timer 2 is incremented by external signal at pin T2

T2I[1:0] = 11 – internal clock input is gated to the Timer 2

Mnemonic: CCCON **Address: C9h**

7	6	5	4	3	2	1	0	Reset
CCI3	CCI2	CCI1	CCI0	CCF3	CCF2	CCF1	CCF0	00H

CCI3: Compare/Capture 3 interrupt control bit.

“1” is enable.

CCI2: Compare/Capture 2 interrupt control bit.

“1” is enable.

CCI1: Compare/Capture 1 interrupt control bit.

“1” is enable.

CCI0: Compare/Capture 0 interrupt control bit.

“1” is enable.

CCF3: Compare/Capture 3 flag set by hardware. This flag can be cleared by software.

CCF2: Compare/Capture 2 flag set by hardware. This flag can be cleared by software.

CCF1: Compare/Capture 1 flag set by hardware. This flag can be cleared by software.

CCF0: Compare/Capture 0 flag set by hardware. This flag can be cleared by software.

Compare/Capture interrupt share T2 interrupt vector.

Mnemonic: CCEN **Address: C1h**

7	6	5	4	3	2	1	0	Reset
-	COCAM1[2:0]			-	COCAM0[2:0]			00H

COCAM1[2:0] 000: Compare/Capture disable

001: Compare enable but no output on Pin

010: Compare mode 0

011: Compare mode 1

100: Capture on rising edge at pin CC1

101: Capture on falling edge at pin CC1

110: Capture on both rising and falling edge at pin CC1

111: Capture on write operation into register CC1

COCAM0[2:0] 000: Compare/Capture disable

001: Compare enable but no output on Pin

010: Compare mode 0

011: Compare mode 1

100: Capture on rising edge at pin CC0

101: Capture on falling edge at pin CC0

110: Capture on both rising and falling edge at pin CC0

111: Capture on write operation into register CC0

Mnemonic: CCEN2 **Address: D1h**

7	6	5	4	3	2	1	0	Reset
-	COCAM3[2:0]			-	COCAM2[2:0]			00H

COCAM3[2:0] 000: Compare/Capture disable

001: Compare enable but no output on Pin

010: Compare mode 0

011: Compare mode 1

100: Capture on rising edge at pin CC3

101: Capture on falling edge at pin CC3

110: Capture on both rising and falling edge at pin CC3

111: Capture on write operation into register CC3

COCAM2[2:0] 000: Compare/Capture disable

001: Compare enable but no output on Pin

010: Compare mode 0

011: Compare mode 1

100: Capture on rising edge at pin CC2

101: Capture on falling edge at pin CC2

110: Capture on both rising and falling edge at pin CC2

111: Capture on write operation into register CC2

7.1 Timer 2 function

Timer 2 can operate as timer, event counter, or gated timer as explained later.

7.1.1 Timer mode

As below Fig. 7-1; In this mode Timer 2 can by incremented in various frequency that depending on the prescaler. The prescaler is selected by bit T2PS[2:0] in register T2CON.

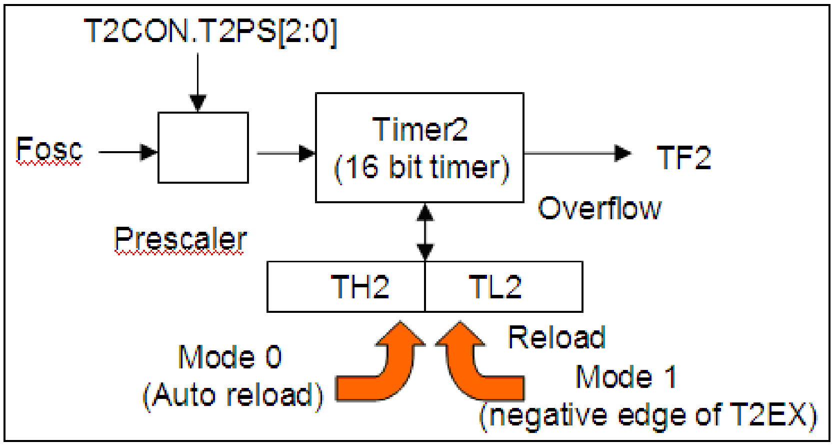


Fig. 7-1: Timer mode and Reload mode function

7.1.2 Event counter mode

As below Fig. 7-2; In this mode, the timer is incremented when external signal T2 change value from 1 to 0. The T2 input is sampled in every cycle. Timer 2 is incremented in the cycle following the one in which the transition was detected.

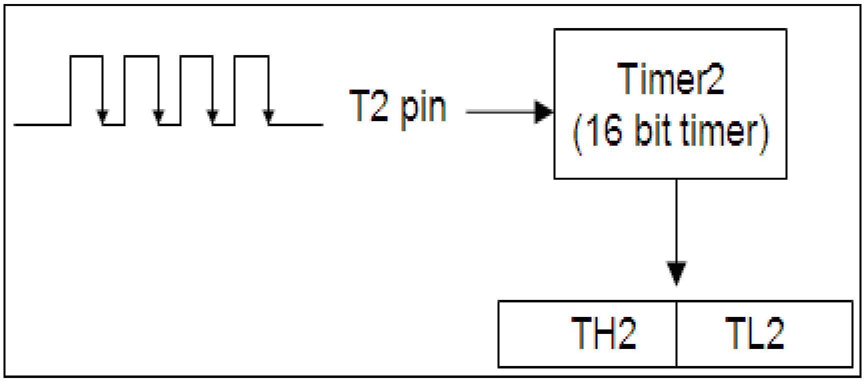


Fig. 7-2: Event counter mode function

7.1.3 Gated timer mode

As below Fig. 7-3; In this mode, the internal clock which incremented timer 2 is gated by external signal T2.

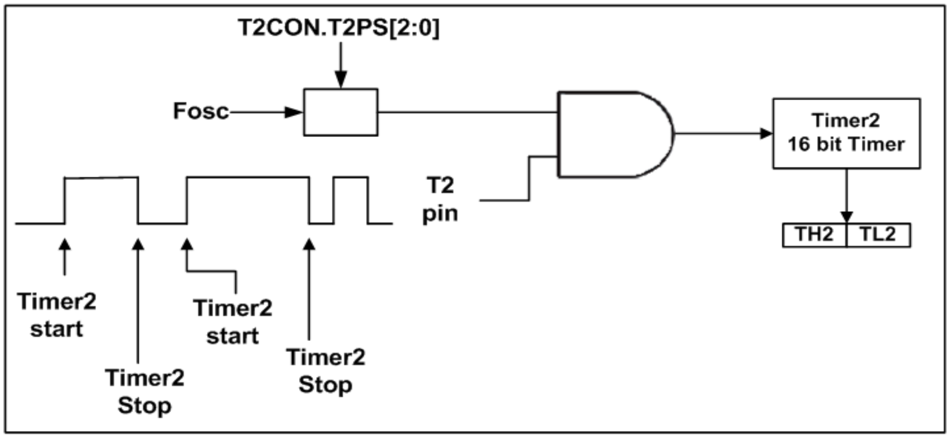


Fig. 7-3: Gated timer mode function

7.1.4 Reload of Timer 2

Reload (16-bit reload from the crc register) can be executed in the following two modes:
Mode 0: Reload signal is generate by a Timer 2 overflows - auto reload
Mode 1: Reload signal is generate by a negative transition at the corresponding input pin T2EX.

7.2 Compare function

In the four independent comparators, the value stored in any compare/capture register is compared with the contents of the timer register. The compare modes 0 and 1 are selected by bits C0CAMx . In both compare modes, the results of comparison arrives at Port 1 within the same machine cycle in which the internal compare signal is activated.

7.2.1 Compare Mode 0

As below Fig. 7-4; In mode 0, when the value in Timer 2 equals the value of the compare register, the output signal changes from low to high. It goes back to a low level on timer overflow. In this mode, writing to the port will have no effect, because the input line from the internal bus and the write-to-latch line are disconnected. The following figure illustrates the function of compare mode 0.

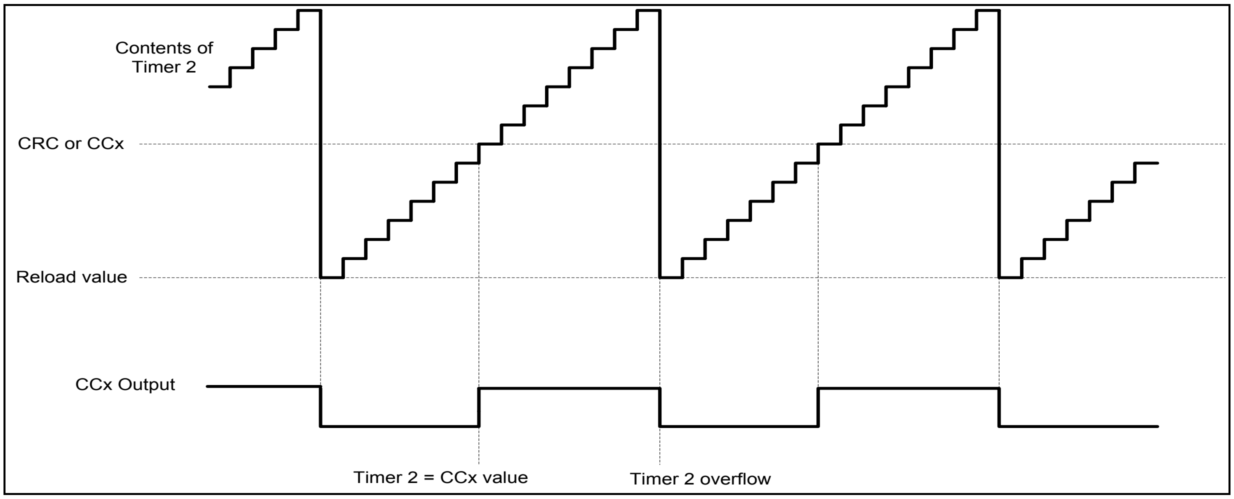


Fig. 7-4: Compare mode 0 function

7.2.2 Compare Mode 1

In compare mode 1, the transition of the output signal can be determined by software. A timer 2 overflow causes no output change. In this mode, both transitions of a signal can be controlled. Fig. 7-5 shows a functional diagram of a register/port configuration in compare Mode 1. In compare Mode 1, the value is written first to the “Shadow Register”, when compare signal is active, this value is transferred to the output register.

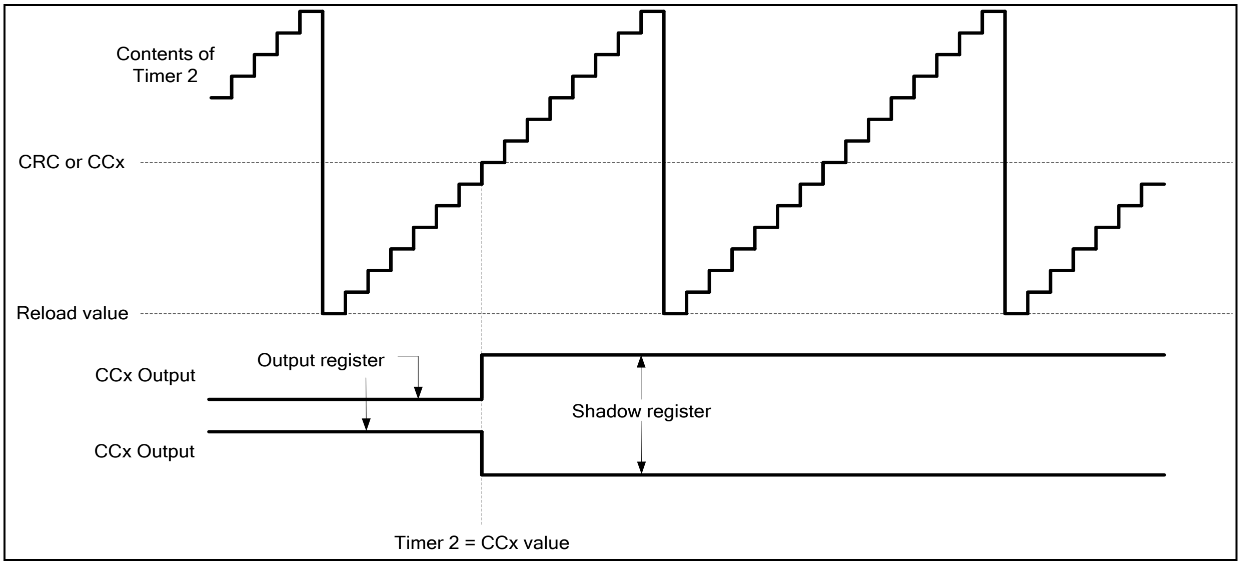


Fig. 7-5: Comparison mode 1 function

7.3 Capture function

Actual timer/counter contents can be saved into registers CCx or CRC upon an external event (mode 0) or a software write operation (mode 1).

7.3.1 Capture Mode 0 (by Hardware)

As below Fig. 7-6; In mode 0, value capture of Timer 2 is executed when:

(1) Rising edge on input CC0-CC3
(2) Falling edge on input CC0-CC3
(3) Both rising and falling edge on input CC0-CC3

The contents of Timer 2 will be latched into the appropriate capture register.

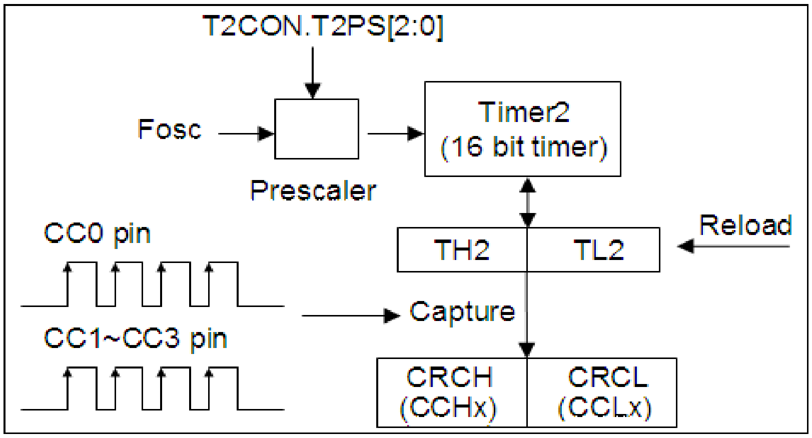


Fig. 7-6: Capture mode 0

7.3.2 Capture Mode 1(by Software)

As below Fig. 7-7; In mode 1, value capture of timer 2 is caused by writing any value into the low-order byte of the dedicated capture register. The value written to the capture register is irrelevant to this function. The contents of Timer 2 will be latched into the appropriate capture register.

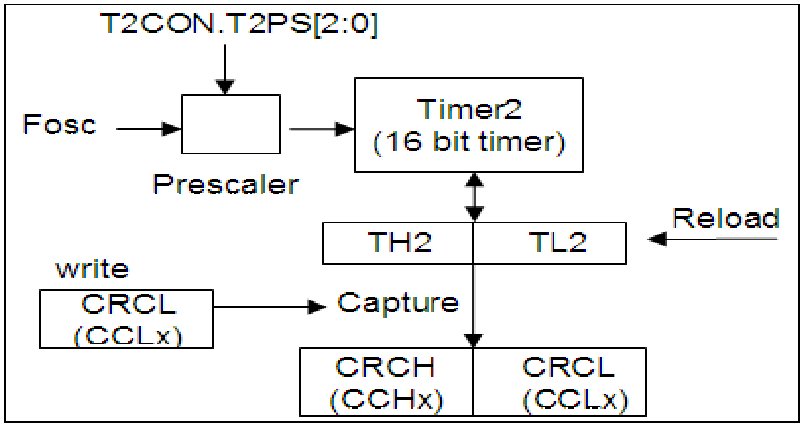


Fig. 7-7: Capture mode 1 function

8. Serial interface

The serial buffer consists of two separate registers, a transmit buffer and a receive buffer.

Writing data to the Special Function Register SBUF sets this data in serial output buffer and starts the transmission. Reading from the SBUF reads data from the serial receive buffer. The serial port can simultaneously transmit and receive data. It can also buffer 1 byte at receive, which prevents the receive data from being lost if the CPU reads the first byte before transmission of the second byte is completed.

Mnemonic	Description	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
Serial interface											
PCON	Power control	87H	SMOD	-	-	-	-	-	STOP	IDLE	00H
AUX	Auxiliary register	91h	BRGS	-	-	-	-	-	-	DPS	00H
SCON	Serial Port control register	98H	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00H
SRELL	Serial Port reload register low byte	AAH	SREL.7	SREL. 6	SREL. 5	SREL. 4	SREL. 3	SREL. 2	SREL. 1	SREL. 0	00H
SRELH	Serial Port reload register high byte	BAH	-	-	-	-	-	-	SREL. 9	SREL. 8	00H
SBUF	Serial Port data buffer	99H	SBUF[7:0]								00H
PFCON	Peripheral Frequency control register	D9h	-	-	SRELPS[1:0]		T1PS[1:0]		T0PS[1:0]		00H

Mnemonic: AUX **Address: 91h**

7	6	5	4	3	2	1	0	Reset
BRGS	-	-	-	-	-	-	DPS	00H

BRGS: BRGS = 0 – baud rate generator from Timer 1.

BRGS = 1 – baud rate generator by SREL.

Mnemonic: SCON **Address: 98h**

7	6	5	4	3	2	1	0	Reset
SM0	SM1	SM2	REN	TB8	RB8	TI	RI	00h

SM0,SM1: Serial Port 0 mode selection.

SM0	SM1	Mode
0	0	0
0	1	1
1	0	2
1	1	3

The 4 modes in UART, Mode 0 ~ 3, are explained later.

SM2: Enables multiprocessor communication feature

REN: If set, enables serial reception. Cleared by software to disable reception.

TB8: The 9th transmitted data bit in modes 2 and 3. Set or cleared by the CPU depending on the function it performs such as parity check, multiprocessor communication etc.

RB8: In modes 2 and 3, it is the 9th data bit received. In mode 1, if SM2 is 0, RB8 is the stop bit. In mode 0, this bit is not used. Must be cleared by software.

TI: Transmit interrupt flag, set by hardware after completion of a serial transfer. Must be cleared by software.

RI: Receive interrupt flag, set by hardware after completion of a serial reception. Must be cleared by software.

8.1 Serial interface

SM0	SM1	Mode	Description	Board Rate
0	0	0	Shift register	Fosc/12
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	Fosc/32 or Fosc/64
1	1	3	9-bit UART	Variable

Here Fosc is the crystal or oscillator frequency.

8.1.1 Mode 0

As below Figure. Pin RXD serves as input and output. TXD outputs the shift clock. 8 bits are transmitted with LSB first. The baud rate is fixed at 1/12 of the crystal frequency. Reception is initialized in Mode 0 by setting the flags in SCON as follows: RI = 0 and REN = 1. In other modes, a start bit when REN = 1 starts receiving serial data.

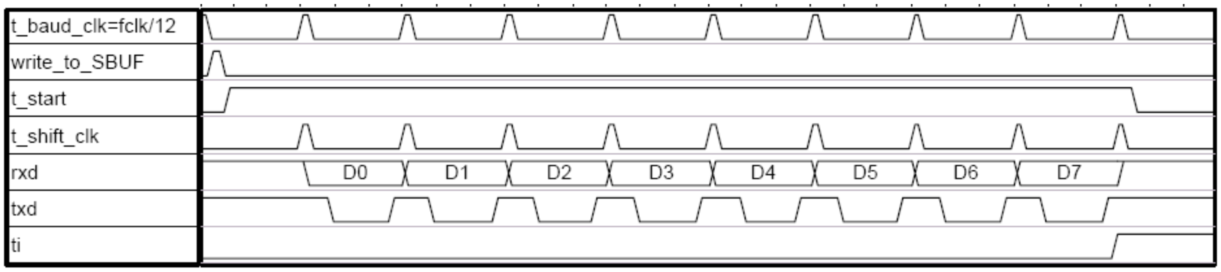


Fig. 8-1: Transmit mode 0

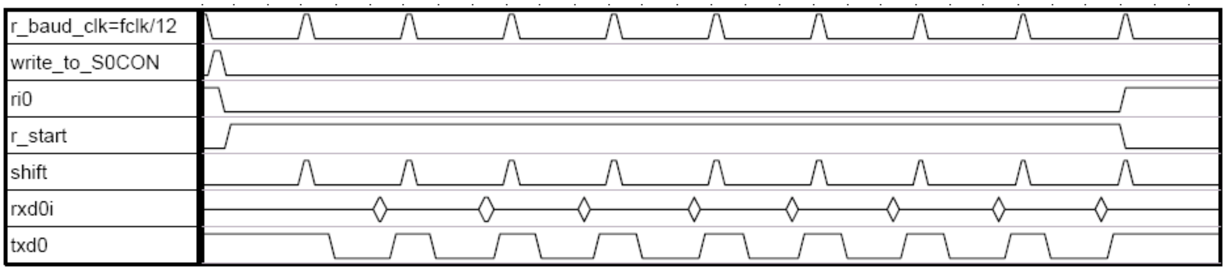


Fig. 8-2: Receive mode 0

8.1.2 Mode 1

As below Figure.Pin RXD serves as input, and TXD serves as serial output. No external shift clock is used, 10 bits are transmitted: a start bit (always 0), 8 data bits (LSB first), and a stop bit (always 1). On receive, a start bit synchronizes the transmission, 8 data bits are available by reading SBUF, and stop bit sets the flag RB8 in the Special Function Register SCON. In mode 1 either internal baud rate generator or timer 1 can be use to specify baud rate.

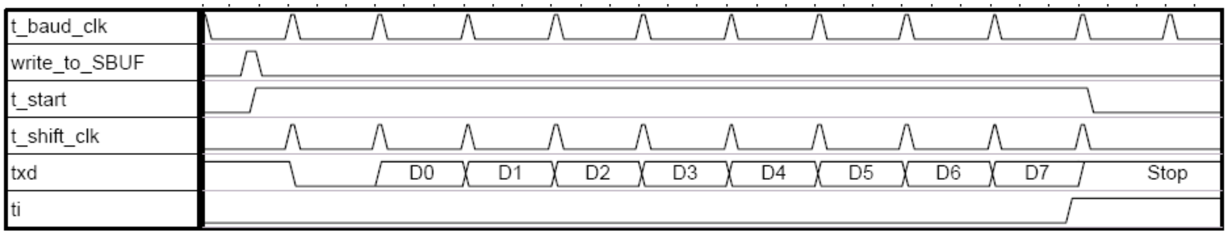


Fig. 8-3: Transmit mode 1

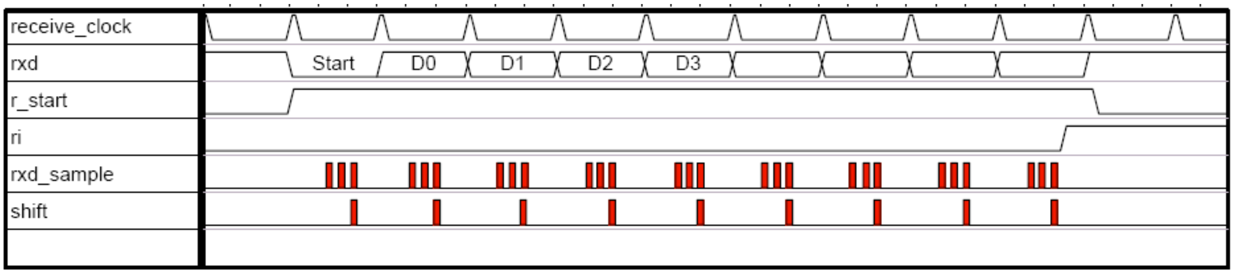


Fig. 8-4: Receive mode 0

8.1.3 Mode 2

This mode is similar to Mode 1, with two differences. The baud rate is fixed at 1/32 (SMOD=1) or 1/64(SMOD=0) of oscillator frequency and 11 bits are transmitted or received: 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 serial interface: at transmission, bit TB8 in SCON is output as the 9th bit, and at receive, the 9th bit affects RB8 in Special Function Register SCON.

8.1.4 Mode 3

As below Figure. The only difference between Mode 2 and Mode 3 is that in Mode 3 either internal baud rate generator or timer 1 can be use to specify baud rate.

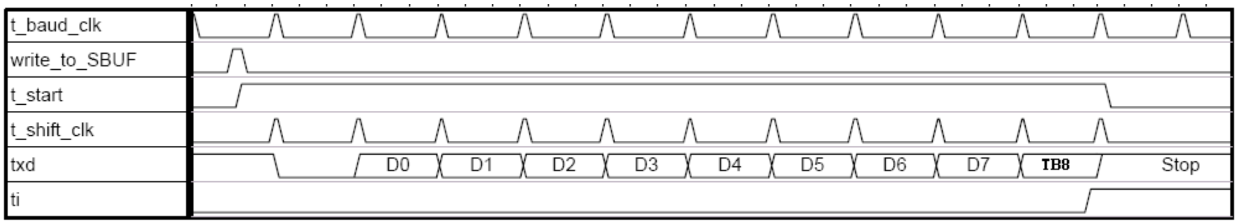


Fig. 8-5: Transfer Mode 2 and Mode 3

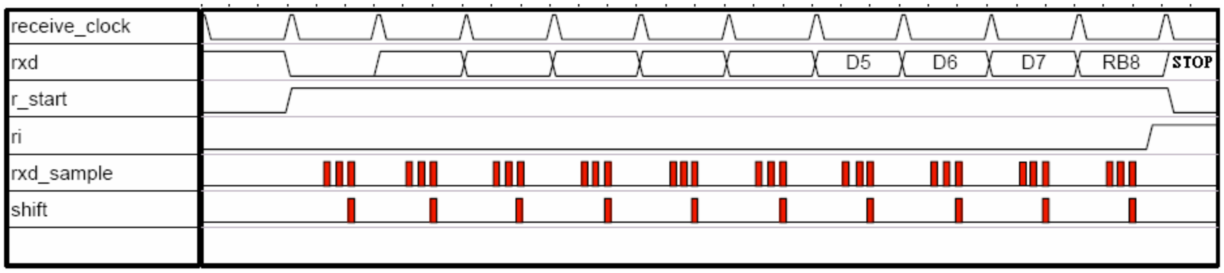


Fig. 8-6: The receiving modes 2 and 3

8.2 Multiprocessor Communication of Serial Interface

The feature of receiving 9 bits in Modes 2 and 3 of Serial Interface can be used for multiprocessor communication. In this case, the slave processors have bit SM2 in SCON set to 1. When the master processor outputs slave's address, it sets the 9th bit to 1, causing a serial port receive interrupt in all the slaves. The slave processors compare the received byte with their network address. If there is a match, the addressed slave will clear SM2 and receive the rest of the message, while other slaves will leave SM2 bit unaffected and ignore this message. After addressing the slave, the host will output the rest of the message with the 9th bit set to 0, so no serial port receive interrupt will be generated in unselected slaves.

8.3 Peripheral Frequency control register

Mnemonic: PFCON **Address: D9h**

7	6	5	4	3	2	1	0	Reset
-	-	SRELPS[1:0]		T1PS[1:0]		T0PS[1:0]		00H

SRELPS[1:0]: SREL Prescaler select

SRELPS[1:0]	Prescaler
00	Fosc/64
01	Fosc /32

T1PS[1:0]: Timer1 Prescaler select

T1PS[1:0]	Prescaler
00	Fosc/12
01	Fosc
10	Fosc/96
11	reserved

8.4 Baud rate generator

8.4.1 Serial interface modes 1 and 3

8.4.1.1 When BRGS = 0 (in Special Function Register AUX).

(1) T1PS[1:0] is 00

$$\text{Baud Rate} = \frac{2^{\text{SMOD}} \times \text{F}_{\text{osc}}}{32 \times 12 \times (256 - \text{TH1})}$$

(2) T1PS[1:0] is 01

$$\text{Baud Rate} = \frac{2^{\text{SMOD}} \times F_{\text{osc}}}{32 \times (256 - \text{TH1})}$$

(3) T1PS[1:0] is 10

$$\text{Baud Rate} = \frac{2^{\text{SMOD}} \times F_{\text{osc}}}{32 \times 96 \times (256 - \text{TH1})}$$

8.4.1.2 When BRGS = 1 (in Special Function Register AUX).

(1) SRELPS[1:0] is 00

$$\text{Baud Rate} = \frac{2^{\text{SMOD}} \times F_{\text{osc}}}{64 \times (2^{10} - \text{SREL})}$$

(2) SRELPS[1:0] is 01

$$\text{Baud Rate} = \frac{2^{\text{SMOD}} \times F_{\text{osc}}}{32 \times (2^{10} - \text{SREL})}$$

9. Watchdog timer

The Watch Dog Timer (WDT) is an 8-bit free-running counter that generate reset signal if the counter overflows. The WDT is useful for systems which are susceptible to noise, power glitches, or electronics discharge which causing software dead loop or runaway. The WDT function can help user software recover from abnormal software condition. The WDT is different from Timer0, Timer1 and Timer2 of general 8052. To prevent a WDT reset can be done by software periodically clearing the WDT counter. User should check WDTF bit of WDTC register whenever un-predicted reset happened. After an external reset the watchdog timer is disabled and all registers are set to zeros.

The watchdog timer has a free running on-chip RC oscillator (20 KHz). The WDT will keep on running even after the system clock has been turned off (for example, in sleep mode). During normal operation or sleep mode, a WDT time-out (if enabled) will cause the MCU to reset. The WDT can be enabled or disabled any time during the normal mode. Please refer the WDTE bit of WDTC register. The default WDT time-out period is approximately 178.0ms (WDTM [3:0] = 0100b).

The WDT has selectable divider input for the time base source clock. To select the divider input, the setting of bit3 ~ bit0 (WDTM [3:0]) of Watch Dog Timer Control Register (WDTC) should be set accordingly.

$$WDTCLK = \frac{20\text{KHz}}{2^{\text{WDTM}}}$$

$$\text{Watchdog reset time} = \frac{256}{\text{WDTCLK}}$$

Table 9-1: WDT time-out period

WDTM [3:0]	Divider (20 KHz RC oscillator in)	Time period @ 20KHz
0000	1	12.8ms
0001	2	25.6ms
0010	4	51.2ms
0011	8	102.4ms
0100	16	204.8ms (default)
0101	32	409.6ms
0110	64	819.2ms
0111	128	1.6384s
1000	256	3.2768s
1001	512	6.5536s
1010	1024	13.10s
1011	2048	26.21s
1100	4096	52.42s
1101	8192	104.85s
1110	16384	209.71s
1111	32768	419.43s

Note: RC oscillator (20 KHz), about ± 20% of variation

After WDTE set to 1, the 8 bit-counter starts to count with the selected time base source clock which set by WDTM [3:0]. It will generate a reset signal when overflows. The WDTE bit will be cleared to 0 automatically when MCU been reset, either hardware reset or WDT reset.
Once the watchdog is started it cannot be stopped. User can refreshed the watchdog timer to zero by writing 0x55 to Watch Dog Timer refresh Key (WDTK) register. This will clear the content of the 8-bit counter and let the counter re-start to count from the beginning. The watchdog timer must be refreshed regularly to prevent reset request signal from becoming active.

When Watchdog timer is overflow, the WDTF flag will set to one and automatically reset MCU. The WDTF flag can be clear by software or external reset or power on reset.

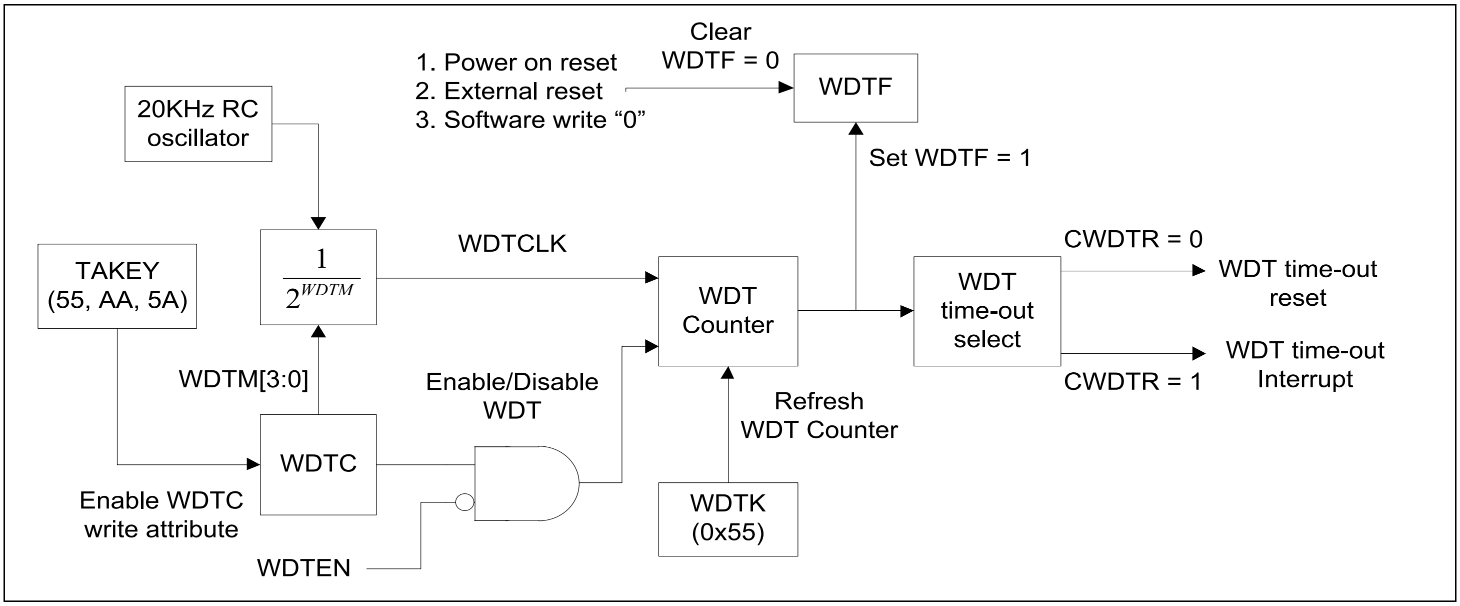


Fig. 9-1: Watchdog timer block diagram

Mnemonic	**Description**	**Address**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**RST**
Watchdog Timer											
TAKEY	Time Access Key register	F7h	TAKEY [7:0]								00H
WDTC	Watchdog timer control register	B6h	-	CWDTR	WDTE	-	WDTM [3:0]				04H
WDTK	Watchdog timer refresh key	B7h	WDTK[7:0]								00H
RSTS	Reset status register	A1h	-	LVRLPINTF	LVRLPF	PDRF	WDTF	SWRF	LVRF	PORF	00H

Mnemonic: TAKEY **Address: F7h**

7	6	5	4	3	2	1	0	Reset
TAKEY [7:0]								00H

Watchdog timer control register (WDTC) is read-only by default; software must write three specific values 55h, AAh and 5Ah sequentially to the TAKEY register to enable the WDTC write attribute. That is

```
MOV TAKEY, #55h
MOV TAKEY, #0AAh
MOV TAKEY, #5Ah
```

Mnemonic: WDTC **Address: B6h**

7	6	5	4	3	2	1	0	Reset
-	CWDTR	WDTE	-	WDTM [3:0]				04H

CWDTR: Watch dog states select bit(Support stop mode wakeup)

0: Enable watch dog reset.

1: Enable watch dog interrupt.

WDTE: Control bit used to enable Watchdog timer.

WDT can be disabled / enabled by the WDTE bit.

0: Disable WDT.

1: Enable WDT.

WDTM [3:0]: WDT clock source divider bit. Please see Table 9-1 to reference the WDT time-out period.

Mnemonic: RSTS **Address: A1h**

7	6	5	4	3	2	1	0	Reset
-	LVRLP INTF	LVRLP F	PDRF	WDTF	SWRF	LVRF	PORF	00h

WDTF: Watchdog timer reset flag. When MCU is reset by watchdog, WDTF flag will be set to one by hardware. This flag clear by software

Mnemonic: WDTK **Address: B7h**

7	6	5	4	3	2	1	0	Reset
WDTK[7:0]								00h

WDTK: Watchdog timer refresh key.

A programmer must write 0x55 into WDTK register, and then the watchdog timer will be cleared to zero.

For example 1, if enable WDT and select time-out reset period is 3.2768s

```
MOV TAKEY, #55h
MOV TAKEY, #0AAh
MOV TAKEY, #5Ah      ; enable WDTC write attribute.
MOV WDTC, #28h       ; Set WDTM [3:0] = 1000b. Set WDTE =1 to enable WDT
                     ; function.
.
.
.
MOV WDTK, #55h       ; Clear WDT timer to 0.
.
.
```

For example 2, if enable WDT and select time-out Interrupt period is 204.8ms.

```
MOV TAKEY, #55h
MOV TAKEY, #0AAh
MOV TAKEY, #5Ah     ; enable WDTC write attribute.
MOV WDTC, #64h       ; Set WDTM [3:0] = 0100b. Set WDTE =1 to enable WDT function
                    ; and Set CWDTR =1 to enable period interrupt function
```

10. Interrupt

The OB38S003 provides 14 interrupt sources with four priority levels. Each source has its own request flag(s) located in a special function register. Each interrupt requested by the corresponding flag could individually be enabled or disabled by the enable bits in SFR's IEN0, IEN1, and IEN2.

When the interrupt occurs, the engine will vector to the predetermined address as shown in Table 10-1. Once interrupt service has begun, it can be interrupted only by a higher priority interrupt. The interrupt service is terminated by a return from instruction RETI. When an RETI is performed, the processor will return to the instruction that would have been next when interrupt occurred.

When the interrupt condition occurs, the processor will also indicate this by setting a flag bit. This bit is set regardless of whether the interrupt is enabled or disabled. Each interrupt flag is sampled once per machine cycle, and then samples are polled by hardware. If the sample indicates a pending interrupt when the interrupt is enabled, then interrupt request flag is set. On the next instruction cycle the interrupt will be acknowledged by hardware forcing an LCALL to appropriate vector address.

Interrupt response will require a varying amount of time depending on the state of microcontroller when the interrupt occurs. If microcontroller is performing an interrupt service with equal or greater priority, the new interrupt will not be invoked. In other cases, the response time depends on current instruction. The fastest possible response to an interrupt is 7 machine cycles. This includes one machine cycle for detecting the interrupt and six cycles for perform the LCALL.

Table 10-1: Interrupt vectors

	Interrupt Request Flags	**Interrupt Vector Address**	**Interrupt Number *(use Keil C Tool)**
1	IE0 – External interrupt 0	0003h	0
2	TF0 – Timer 0 interrupt	000Bh	1
3	IE1 – External interrupt 1	0013h	2
4	TF1 – Timer 1 interrupt	001Bh	3
5	RI/TI – Serial channel interrupt	0023h	4
6	TF2/EXF2 – Timer 2 interrupt	002Bh	5
7	PWMIF – PWM interrupt	0043h	8
8	SPIIF – SPI interrupt	004Bh	9
9	ADCIF – A/D converter interrupt	0053h	10
10	KBIIF – keyboard Interface interrupt	005Bh	11
11	LVIIF – Low Voltage Interrupt	0063h	12
12	IICIF – IIC interrupt	006Bh	13
13	WDTIF – Watchdog interrupt	008Bh	17
14	Comparator interrupt	0093h	18

* See Keil C about C51 User's Guide about Interrupt Function description

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
Interrupt											
IEN0	Interrupt Enable 0 register	A8H	EA	-	ET2	ES	ET1	EX1	ET0	EX0	00H
IEN1	Interrupt Enable 1 register	B8H	EXEN 2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	IEPW M	00H
IEN2	Interrupt Enable 2 register	9AH	-	-	-	-	-	ECmpl	EWDT	-	00H
IRCON	Interrupt request register	C0H	EXF2	TF2	IICIF	LVIIF	KBIIF	ADCIF	SPIIF	PWMI F	00H
IRCON2	Interrupt request register 2	97H	-	-	-	-	-	CmpIF	WDTI F	-	00H
IP0	Interrupt priority level 0	A9H	-	-	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	00H
IP1	Interrupt priority level 1	B9H	-	-	IP1.5	IP1.4	IP1.3	IP1.2	IP1.1	IP1.0	00H
ENHIT	ENHance Interrupt Type Register	E5H	-	ENHIT 1	-	ENHIT 0	-	-	-	-	07H
INTDEG	External Interrupt Deglitch Register	EEh	-	-	-	-	INT1DEG[1:0]		INT0DEG[1:0]		00H

Mnemonic: IEN0 **Address: A8h**

7	6	5	4	3	2	1	0	Reset
EA	-	ET2	ES0	ET1	EX1	ET0	EX0	00h

EA: EA=0 – Disable all interrupt.

EA=1 – Enable all interrupt.

ET2: ET2=0 – Disable Timer 2 overflow or external reload interrupt.

ET2=1 – Enable Timer 2 overflow or external reload interrupt.

ES: ES=0 – Disable Serial channel interrupt.

ES=1 – Enable Serial channel interrupt.

ET1: ET1=0 – Disable Timer 1 overflow interrupt.

ET1=1 – Enable Timer 1 overflow interrupt.

EX1: EX1=0 – Disable external interrupt 1.

EX1=1 – Enable external interrupt 1.

ET0: ET0=0 – Disable Timer 0 overflow interrupt.

ET0=1 – Enable Timer 0 overflow interrupt.

EX0: EX0=0 – Disable external interrupt 0.

EX0=1 – Enable external interrupt 0.

Mnemonic: IEN1 **Address: B8h**

7	6	5	4	3	2	1	0	Reset
EXEN2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	IEPWM	00H

EXEN2: Timer 2 reload interrupt enable.

EXEN2 = 0 – Disable Timer 2 external reload interrupt.

EXEN2 = 1 – Enable Timer 2 external reload interrupt.

IEIIC: IIC interrupt enable.

IEIIC = 0 – Disable IIC interrupt.

IEIIC = 1 – Enable IIC interrupt.

IELVI: LVI interrupt enable.

IELVI = 0 – Disable LVI interrupt.

IELVI = 1 – Enable LVI interrupt.

IEKBI: KBI interrupt enable.

IEKBI = 0 – Disable KBI interrupt.

IEKBI = 1 – Enable KBI interrupt.

IEADC: A/D converter interrupt enable

IEADC = 0 – Disable ADC interrupt.

IEADC = 1 – Enable ADC interrupt.

IESPI: SPI interrupt enable.

IESPI = 0 – Disable SPI interrupt.

IESPI = 1 – Enable SPI interrupt.

IEPWM: PWM interrupt enable.

IEPWM = 0 – Disable PWM interrupt.

IEPWM = 1 – Enable PWM interrupt.

Mnemonic: IEN2 **Address: 9Ah**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	ECmpI	EWDT	-	00H

ECmpI: Enable Comparator interrupt(include comparator_0 and comparator_1).

ECmpI = 0 – Disable Comparator interrupt.

ECmpI = 1 – Enable Comparator interrupt.

EWDT: Enable Watch dog interrupt.

EWDT = 0 – Disable Watch dog interrupt.

EWDT = 1 – Enable Watch dog interrupt.

Mnemonic: IRCON **Address: C0h**

7	6	5	4	3	2	1	0	Reset
EXF2	TF2	IICIF	LVIIF	KBIIF	ADCIF	SPIIF	PWMIF	00H

EXF2: Timer 2 external reload flag. Must be cleared by software.

TF2: Timer 2 overflow flag. Must be cleared by software.

IICIF: IIC interrupt flag.

LVIIF: LVI interrupt flag.

KBIIF: KBI interrupt flag.

ADCIF: A/D converter end interrupt flag.

SPIIF: SPI interrupt flag.

PWMIF: PWM interrupt flag.

Mnemonic: IRCON2 **Address: 97h**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	CmpIF	WDTIF	-	00H

CmpIF: Comparator interrupt flag
HW will clear this flag automatically when enter interrupt vector.
SW can clear this flag also.(in case analog comparator INT disable)

WDTIF: Watch dog interrupt flag

Mnemonic: ENHIT **Address: E5h**

7	6	5	4	3	2	1	0	Reset
-	ENHIT1	-	ENHIT0	-	-	-	-	07H

ENHIT1: Interrupt 1 edge trigger control bit.

When ENHIT1 is set to 0 and IT1 is set to 1, The method of edge trigger is falling edge trigger.

When ENHIT1 and IT1 both are set to 1, The method of edge trigger is rising edge trigger.

	ENHIT1=0	**ENHIT1=1**
IT1=0	INT1 low level trigger	INT1 low level trigger
IT1=1	INT1 failing edge trigger	INT1 rising edge trigger

ENHIT0: Interrupt 0 edge trigger control bit.

When ENHIT0 is set to 0 and IT0 is set to 1, The method of edge trigger is falling edge trigger.

When ENHIT0 and IT0 both are set to 1, The method of edge trigger is rising edge trigger.

	ENHIT0=0	**ENHIT0=1**
IT0=0	INT0 low level trigger	INT0 low level trigger
IT0=1	INT0 failing edge trigger	INT0 rising edge trigger

Mnemonic: INTDEG **Address: EEh**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	INT1DEG[1:0]		INT0DEG[1:0]		00H

INT1DEG[1:0] Select INT1 deglitch time.
00: no deglitch.
01: 5us
10: 10us
11: 15us

INT0DEG[1:0] Select INT0 deglitch time.
00: no deglitch.
01: 5us
10: 10us
11: 15us

10.1 Priority level structure

All interrupt sources are combined in groups:

Table 10-2: Priority level groups

Groups		
External interrupt 0	-	PWM interrupt
Timer 0 interrupt	Watchdog interrupt	SPI interrupt
External interrupt 1	Comparator interrupt	ADC interrupt
Timer 1 interrupt	-	KBI interrupt
Serial channel interrupt	-	LVI interrupt
Timer 2 interrupt	-	IIC interrupt

Each group of interrupt sources can be programmed individually to one of four priority levels by setting or clearing one bit in the special function register ip0 and one in ip1. If requests of the same priority level will be received simultaneously, an internal polling sequence determines which request is serviced first

Mnemonic: IP0 **Address: A9h**

7	6	5	4	3	2	1	0	Reset
-	-	IP0.5	IP0.4	IP0.3	IP0.2	IP0.1	IP0.0	00h

Mnemonic: IP1 **Address: B9h**

7	6	5	4	3	2	1	0	Reset
-	-	IP1.5	IP1.4	IP1.3	IP1.2	IP1.1	IP1.0	00h

Table 10-3: Priority levels

IP1.x	IP0.x	Priority Level
0	0	Level0 (lowest)
0	1	Level1
1	0	Level2
1	1	Level3 (highest)

Table 10-4: Groups of priority

Bit	Group		
IP1.0, IP0.0	External interrupt 0	-	PWM interrupt
IP1.1, IP0.1	Timer 0 interrupt	Watchdog interrupt	SPI interrupt
IP1.2, IP0.2	External interrupt 1	Comparator interrupt	ADC interrupt
IP1.3, IP0.3	Timer 1 interrupt	-	KBI interrupt
IP1.4, IP0.4	Serial channel 0 interrupt	-	LVI interrupt
IP1.5, IP0.5	Timer 2 interrupt	-	IIC interrupt

Table 10-5: Polling sequence

Interrupt source	Sequence
External interrupt 0	
PWM interrupt	
Timer 0 interrupt	

Watchdog interrupt	
SPI interrupt	
External interrupt 1	
Comparator interrupt	
ADC interrupt	
Timer 1 interrupt	
KBI interrupt	
Serial channel 0 interrupt	
LVI interrupt	
Timer 2 interrupt	
IIC interrupt	

11. Power Management Unit

Power management unit serves two power management modes, IDLE and STOP, for the users to do power saving function.

Mnemonic: PCON **Address: 87h**

7	6	5	4	3	2	1	0	Reset
SMOD	-	-	-	-	-	STOP	IDLE	00h

STOP: Stop mode control bit. Setting this bit turning on the Stop Mode.
Stop bit is always read as 0

IDLE: Idle mode control bit. Setting this bit turning on the Idle Mode.
Idle bit is always read as 0

11.1 Idle mode

Setting the IDLE bit of PCON register invokes the IDLE mode. The IDLE mode leaves internal clocks and peripherals running. Power consumption drops because the CPU is not active. The CPU can exit the IDLE state with any interrupts or a reset.

11.2 Stop mode

Setting the STOP bit of PCON register invokes the STOP mode. All internal clocking in this mode is turn off. The CPU will exit this state from a no-clocked interrupt (external INT0/1 and LVI, KBI, Comparator interrupt, Watchdog interrupt) or a reset (WDT and LVR) condition. Internally generated interrupts (timer, serial port ...) have no effect on stop mode since they require clocking activity.

12. Pulse Width Modulation (PWM)

OB38S003 provides four -channel PWM outputs.
The interrupt vector is 43h.

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
PWM											
PWMC	PWM Control register	B5h	PWMCS[2:0]			-	PWM3 EN	PWM2 EN	PWM1 EN	PWM0 EN	00H
PWMD0H	PWM 0 Data register high byte	BCh	PWMP 0	-	-	-	-	-	PWMD0[9:8]		00H
PWMD0L	PWM 0 Data register low byte	BDh	PWMD0[7:0]								00H
PWMD1H	PWM 1 Data register high byte	BEh	PWMP 1	-	-	-	-	-	PWMD1[9:8]		00H
PWMD1L	PWM 1 Data register low byte	BFh	PWMD1[7:0]								00H
PWMD2H	PWM 2 Data register high byte	B1h	PWMP 2	-	-	-	-	-	PWMD2[9:8]		00H
PWMD2L	PWM 2 Data register low byte	B2h	PWMD2[7:0]								00H
PWMD3H	PWM 3 Data register high byte	B3h	PWMP 3	-	-	-	-	-	PWMD3[9:8]		00H
PWMD3L	PWM 3 Data register low byte	B4h	PWMD3[7:0]								00H
PWMMDH	PWM Max Data register high byte	CEh	-	-	-	-	-	-	PWMMD[9:8]		00H
PWMMDL	PWM Max Data register low byte	CFh	PWMMD[7:0]								FFH

Mnemonic: PWMC **Address: B5h**

7	6	5	4	3	2	1	0	Reset
PWMCS[2:0]			-	PWM3EN	PWM2EN	PWM1EN	PWM0EN	00H

PWMCS[2:0]: PWM clock select.

PWMCS [2:0]	Mode
000	Fosc
001	Fosc/2
010	Fosc/4
011	Fosc/6
100	Fosc/8
101	Fosc/12
110	Timer 0 overflow
111	Timer 0 external input (P1.2/T0)

PWM3EN PWM channel 3 enable control bit.

PWM3EN = 1 – PWM channel 1 enable.

PWM3EN = 0 – PWM channel 1 disable.

PWM2EN PWM channel 2 enable control bit.

PWM2EN = 1 – PWM channel 1 enable.

PWM2EN = 0 – PWM channel 1 disable.

PWM1EN: PWM channel 1 enable control bit.

PWM1EN = 1 – PWM channel 1 enable.

PWM1EN = 0 – PWM channel 1 disable.

PWM0EN: PWM 0 enable control bit.

PWM0EN = 1 – PWM channel 0 enable.

PWM0EN = 0 – PWM channel 0 disable.

Mnemonic: PWMD0H **Address: BCh**

7	6	5	4	3	2	1	0	Reset
PWMP0	-	-	-	-	-	PWMD0[9:8]		00H

Mnemonic: PWMD0L **Address: BDh**

7	6	5	4	3	2	1	0	Reset
PWMD0[7:0]								00h

PWMP0: PWM channel 0 idle polarity select.

"0" – PWM channel 0 will idle low.

"1" – PWM channel 0 will idle high.

PWMD0[9:0]: PWM channel 0 data register.

Mnemonic: PWMD1H **Address: BEh**

7	6	5	4	3	2	1	0	Reset
PWMP1	-	-	-	-	-	PWMD1[9:8]		00H

Mnemonic: PWMD1L **Address: BFh**

7	6	5	4	3	2	1	0	Reset
PWMD1[7:0]								00H

PWMP1: PWM channel 1 idle polarity select.

"0" – PWM channel 1 will idle low.

"1" – PWM channel 1 will idle high.

PWMD1[9:0]: PWM channel 1 data register.

Mnemonic: PWMD2H **Address: B1h**

7	6	5	4	3	2	1	0	Reset
PWMP2	-	-	-	-	-	PWMD2[9:8]		00H

Mnemonic: PWMD2L **Address: B2h**

7	6	5	4	3	2	1	0	Reset
PWMD2[7:0]								00H

PWMP2: PWM channel 2 idle polarity select.

"0" – PWM channel 2 will idle low.

"1" – PWM channel 2 will idle high.

PWMD2[9:0]: PWM channel 2 data register.

Mnemonic: PWMD3H **Address: B3h**

7	6	5	4	3	2	1	0	Reset
PWMP3	-	-	-	-	-	PWMD3[9:8]		00H

Mnemonic: PWMD3L **Address: B4h**

7	6	5	4	3	2	1	0	Reset
PWMD3[7:0]								00H

PWMP3: PWM channel 3 idle polarity select.

“0” – PWM channel 3 will idle low.

“1” – PWM channel 3 will idle high.

PWMD3[9:0]: PWM channel 3 data register.

Mnemonic: PWMMDH **Address: CEh**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	PWMMD[9:8]		00H

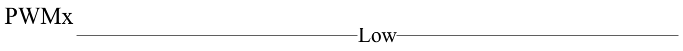
Mnemonic: PWMMDL **Address: CFh**

7	6	5	4	3	2	1	0	Reset
PWMMD[7:0]								FFH

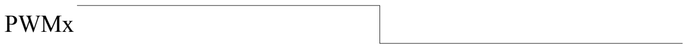
PWMMD[9:0]: PWM Max Data register.

PWM count from 0000h to PWMMD[9:0]. When PWM count data equal PWMMD[9:0] is overflow.

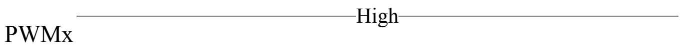
PWMPx = 0 & PWMDx = 00h



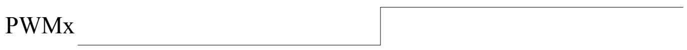
PWMPx = 0 & PWMDx ≠ 00h



PWMPx = 1 & PWMDx = 00h



PWMPx = 1 & PWMDx ≠ 00h



$$\text{PWM period} = \frac{\text{PWMMD} + 1}{\text{PWM clock}}$$

$$\text{Leader pulse} = \frac{\text{PWMDx}}{\text{PWM clock}}$$

13. IIC function

The IIC module uses the SCL (clock) and the SDA (data) line to communicate with external IIC interface. Its speed can be selected to 400Kbps (maximum) by software setting the IICBR [2:0] control bit. The IIC module provided 2 interrupts (RXIF, TXIF). It will generate START, repeated START and STOP signals automatically in master mode and can detects START, repeated START and STOP signals in slave mode. The maximum communication length and the number of devices that can be connected are limited by a maximum bus capacitance of 400pF.

The interrupt vector is 6Bh.

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
IIC function											
IICCTL	IIC control register	F9h	IICEN	MSS	MAS	AB_EN	BF_EN	IICBR[2:0]			04H
IICS	IIC status register	F8h	-	MPIF	LAIF	RXIF	TXIF	RXAK	TXAK	RW or BB	00H
IICA1	IIC Address 1 register	FAh	IICA1[7:1]							MATCH1or RW1	A0H
IICA2	IIC Address 2 register	FBh	IICA2[7:1]							MATCH2 or RW2	60H
IICRWD	IIC Read/Write register	FCh	IICRWD[7:0]								00H
IICEBT	IIC Enaable Bus Transaction	FDh	FU_EN		-	-	-	-	-	-	00H

Mnemonic: IICCTL **Address: F9h**

7	6	5	4	3	2	1	0	Reset
IICEN	MSS	MAS	AB_EN	BF_EN	IICBR[2:0]			04h

IICEN: Enable IIC module

IICEN = 1 is Enable

IICEN = 0 is Disable.

MSS: Master or slave mode select.

MSS = 1 is master mode.

MSS = 0 is slave mode.

*The software must set this bit before setting others register.

MAS: Master address select (master mode only)

MAS = 0 is to use IICA1.

MAS = 1 is to use IICA2.

AB_EN: Arbitration lost enable bit. (Master mode only)
If set AB_EN bit, the hardware will check arbitration lost. Once arbitration lost occurred, hardware will return to IDLE state. If this bit is cleared, hardware will not care arbitration lost condition. Set this bit when multi-master and slave connection. Clear this bit when single master to single slave.

BF_EN: Bus busy enable bit. (Master mode only)
If set BF_EN bit, hardware will not generate a start condition to bus until BF=0. Clear this bit will always generate a start condition to bus when MStart is set. Set this bit when multi-master

and slave connection. Clear this bit when single master to single slave.

IICBR[2:0]: Baud rate selection (master mode only), where Fosc is the external crystal or oscillator frequency. The default is Fosc/512 for users' convenience.

IICBR[2:0]	Baud rate
000	Fosc/32
001	Fosc/64
010	Fosc/128
011	Fosc/256
100	Fosc/512
101	Fosc/1024
110	Fosc/2048
111	Fosc/4096

Mnemonic: IICS **Address: F8H**

7	6	5	4	3	2	1	0	Reset
-	MPIF	LAIF	RXIF	TXIF	RXAK	TxAK	RW or BB	00H

MPIF: The Stop condition Interrupt Flag

The stop condition occurred and this bit will be set. Software need to clear this bit

LAIF: Arbitration lost bit. (Master mode only)

The Arbitration Interrupt Flag, the bus arbitration lost occurred and this bit will be set. Software need to clear this bit

RxIF: The data Receive Interrupt Flag (RXIF) is set after the IICRWD (IIC Read Write Data Buffer) is loaded with a newly receive data.

TxIF: The data Transmit Interrupt Flag (TXIF) is set when the data of the IICRWD (IIC Read Write Data Buffer) is downloaded to the shift register.

RxAK: The Acknowledge Status indicate bit. When clear, it means an acknowledge signal has been received after the complete 8 bits data transmit on the bus.

TxAK: The Acknowledge status transmit bit. When received complete 8 bits data, this bit will set (NoAck) or clear (Ack) and transmit to master to indicate the receive status.

RW or BB: **Master Mode:**

BB : Bus busy bit

If detect scl=0 or sda=0 or bus start, this bit will be set. If detect stop,this bit will be cleared. This bit can be cleared by software to return ready state.

Slave Mode:

RW:The slave mode read (received) or wrote (transmit) on the IIC bus. When this bit is clear, the slave module received data on the IIC bus (SDA).(Slave mode only)

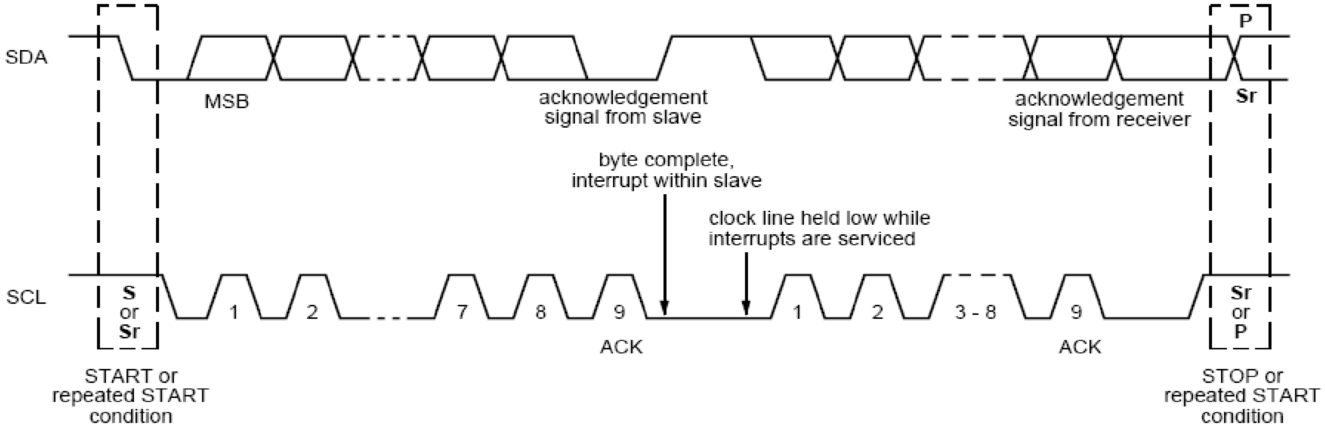

SDA
SCL
P
Sr
MSB
acknowledgement signal from slave
acknowledgement signal from receiver
byte complete, interrupt within slave
clock line held low while interrupts are serviced
S or Sr
1
2
7
8
9
ACK
1
2
3 - 8
9
ACK
Sr or P
START or repeated START condition
STOP or repeated START condition

Fig. 13-1: Acknowledgement bit in the 9th bit of a byte transmission

Mnemonic: IICA1 **Address: FAH**

7	6	5	4	3	2	1	0	Reset
IICA1[7:1]							Match1 or RW1	A0H
R/W							R or R/W	

Slave mode:

IICA1[7:1]: IIC Address registers

This is the first 7-bit address for this slave module. It will be checked when an address (from master) is received

Match1: When IICA1 matches with the received address from the master side, this bit will set to 1 by hardware. When IIC bus gets first data, this bit will clear.

Master mode:

IICA1[7:1]: IIC Address registers

This 7-bit address indicates the slave with which it wants to communicate.

RW1: This bit will be sent out as RW of the slave side if the module has set the MStart or RStart bit. It appears at the 8th bit after the IIC address as shown in Fig. 13-2. It is used to tell the salve the direction of the following communication. If it is 1, the module is in master receive mode. If 0, the module is in master transmit mode.

RW1=1, master receive mode

RW1=0, master transmit mode

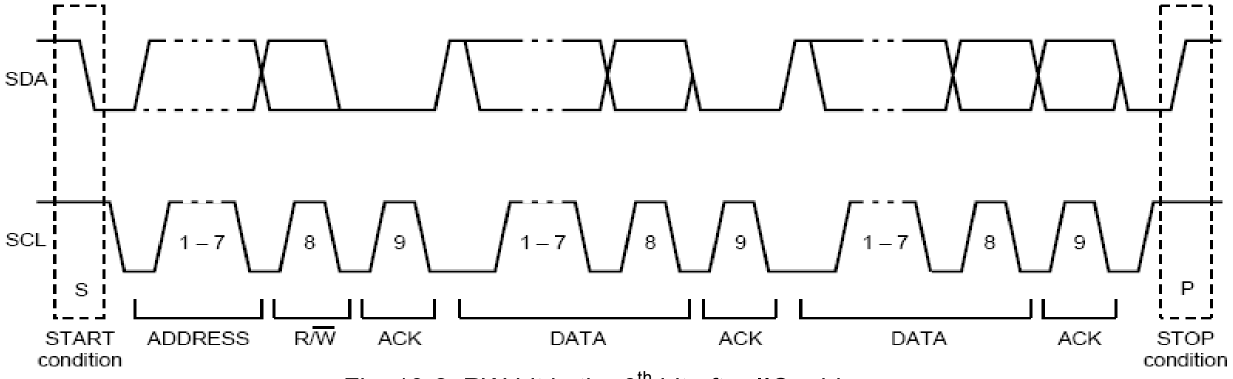


Fig. 13-2: RW bit in the 8th bit after IIC address

Mnemonic: IICA2 **Address: FBh**

7	6	5	4	3	2	1	0	Reset
IICA2[7:1]							Match2 or RW2	60h
R/W							R or R/W	

Slave mode:

IICA2[7:1]: IIC Address registers

This is the second 7-bit address for this slave module.

It will be checked when an address (from master) is received

Match2: When IICA2 matches with the received address from the master side, this bit will set to 1 by hardware. When IIC bus gets first data, this bit will clear.

Master mode:

IICA2[7:1]: IIC Address registers

This 7-bit address indicates the slave with which it wants to communicate.

RW2: This bit will be sent out as RW of the slave side if the module has set the MStart or RStart bit. It is used to tell the salve the direction of the following communication. If it is 1, the module is in master receive mode. If 0, the module is in master transmit mode.

RW2=1, master receive mode

RW2=0, master transmit mode

Mnemonic: IICRWD **Address: FCh**

7	6	5	4	3	2	1	0	Reset
IICRWD[7:0]								00h

IICRWD[7:0]: IIC read write data buffer.

In receiving (read) mode, the received byte is stored here.

In transmitting mode, the byte to be shifted out through SDA stays here.

Mnemonic: IICEBT **Address: FDH**

7	6	5	4	3	2	1	0	Reset
FU_EN		-	-	-	-	-	-	00H

Master Mode:

00: reserved

01: IIC bus module will enable read/write data transfer on SDA and SCL.

10: IIC bus module generate a start condition on the SDA/SCL, then send out address which is stored in the IICA1/IICA2(selected by MAS control bit)

11: IIC bus module generates a stop condition on the SDA/SCL.

FU_EN[7:6] will be auto-clear by hardware, so setting FU_EN[7:6] repeatedly is necessary.

Slave mode:

01: FU_EN[7:6] should be set as 01 only. The other value is inhibited.

Notice:

1. FU_EN[7:6] should be set as 01 before read/write data transfer for bus release; otherwise, SCL will be locked(pull low).
2. FU_EN[7:6] should be set as 01 after read/write data transfer for receiving a stop condition from bus master.
3. In transmit data mode(slave mode), the output data should be filled into IICRWD before setting FU_EN[7:6] as 01.
4. FU_EN[7:6] will be auto-clear by hardware, so setting FU_EN[7:6]

   repeatedly is necessary.

14. SPI Function - Serial Peripheral Interface

Serial Peripheral Interface (SPI) is a synchronous protocol that allows a master device to initiate communication with slave devices.

The interrupt vector is 4Bh.

There are 4 signals used in SPI, they are

SPI_MOSI: data output in the master mode, data input in the slave mode,
SPI_MISO: data input in the master mode, data output in the master mode,
SPI_SCK: clock output from the master, the above data are synchronous to this signal
SPI_SS: input in the slave mode.

This slave device detects this signal to judge if it is selected by the master.
In the master mode, it can select the desired slave device by any IO with value = 0. Fig. 14-1 is an example showing the relation of the 4 signals between master and slaves.

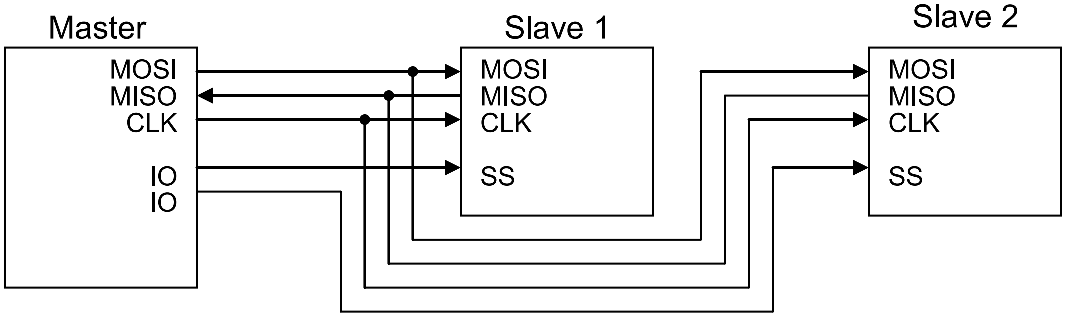


Fig. 14-1: SPI signals between master and slave devices

There is only one channel SPI interface. The SPI SFRs are shown as below:

SPI	Description	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
SPI function											
SPIC1	SPI control register 1	F1h	SPIEN	SPIMSS	SPISSP	SPICKP	SPICKE	SPIBR[2:0]			08H
SPIC2	SPI control register 2	F2h	SPIFD	TBC[2:0]			SPIRST	RBC[2:0]			00H
SPIS	SPI status register	F5h	SPIRF	SPIMLS	SPIOV	SPITXIF	SPITDR	SPIRXIF	SPIRDR	SPIRS	40H
SPITXD	SPI transmit data buffer	F3h	SPITXD[7:0]								00H
SPIRXD	SPI receive data buffer	F4h	SPIRXD[7:0]								00H

Mnemonic: SPIC1 **Address: F1h**

7	6	5	4	3	2	1	0	Reset
SPIEN	SPIMSS	SPISSP	SPICKP	SPICKE	SPIBR[2:0]			08h

SPIEN: Enable SPI module.

"1" is Enable.

"0" is Disable.

SPIMSS: Master or Slave mode Select

"1" is Master mode.

"0" is Slave mode.

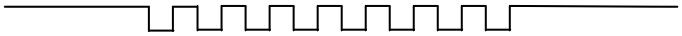
SPISSP: SS or CS active polarity.(Slave mode used only)

"1" - high active.

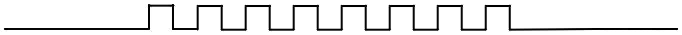
"0" - low active.

SPICKP: Clock idle polarity select.

"1" - SCK will idle high. Ex :



"0" - SCK will idle low. Ex :

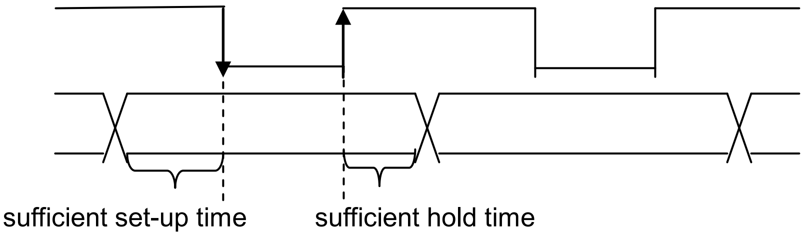


SPICKE: Clock sample edge select.

"1" – rising edge latch data.

"0" – falling edge latch data.

* To ensure the data latch stability, OB38S003 generate the output data as given in the following example, the other side can latch the stable data no matter in rising or falling edge.



SPIBR[2:0]: SPI baud rate select. (Master mode used only)

SPIBR[2:0]	Baud rate
0:0:0	Fosc/4
0:0:1	Fosc /8
0:1:0	Fosc /16
0:1:1	Fosc /32
1:0:0	Fosc /64
1:0:1	Fosc /128
1:1:0	Fosc /256
1:1:1	Fosc /512

Mnemonic: SPIC2 **Address: F2h**

7	6	5	4	3	2	1	0	Reset
SPIFD	TBC[2:0]			SPIRST	RBC[2:0]			00h

SPIFD: Full-duplex mode enable.

"1" is enable full-duplex mode.

"0" is disable full-duplex mode.

When it is set, the TBC[2:0] and RBC[2:0] will be reset and keep to zero. When the Master device transmits data to the Slave device via the MOSI line, the Slave device responds by sending data to the Master device via the MISO line. This implies full-duplex transmission with both data out and data in synchronized with the same clock.

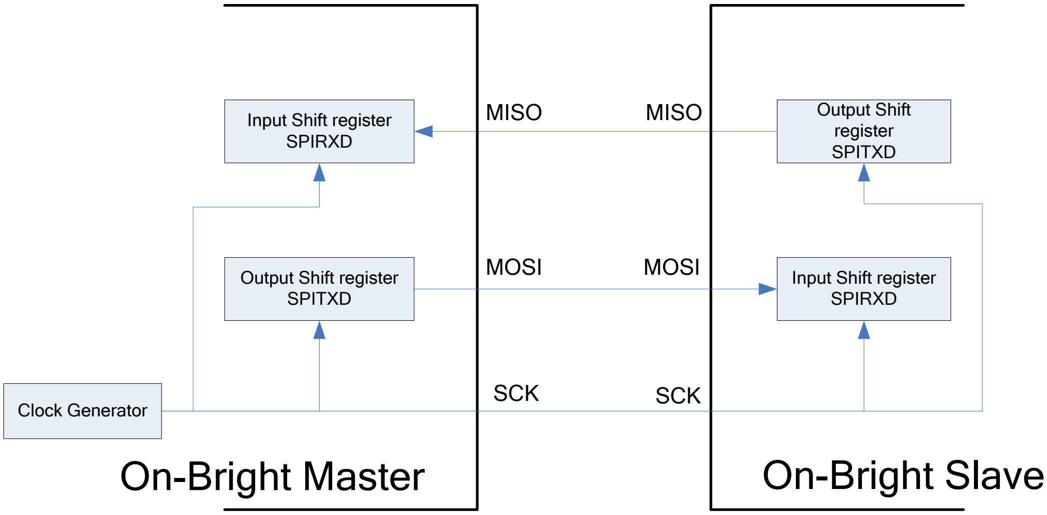


Fig. 14-2: SPI master and slave transmission method

TBC[2:0]: SPI transmitter bit counter.

TBC[2:0]	Bit counter
0:0:0	8 bits output
0:0:1	1 bit output
0:1:0	2 bits output
0:1:1	3 bits output
1:0:0	4 bits output
1:0:1	5 bits output
1:1:0	6 bits output
1:1:1	7 bits output

SPIRST: SPI Re-start (Slave mode used only)

SPIRST=0:Re-start function disable.SPI transmit/receive data when SS active.

In SPITXD/SPIRXD buffer, data got from previous SS active period will not be removed (i.e. it's valid).

SPIRST=1:Re-start function enable.SPI transmit/receive new data when SS re-active;

In SPITXD/SPIRXD buffer, data got from previous SS active period will be removed (i.e. It's invalid).

RBC[2:0]: SPI receiver bit counter.

RBC[2:0]	Bit counter
0:0:0	8 bits input
0:0:1	1 bit input
0:1:0	2 bits input
0:1:1	3 bits input
1:0:0	4 bits input
1:0:1	5 bits input
1:1:0	6 bits input
1:1:1	7 bits input

Mnemonic: SPIS **Address:F5H**

7	6	5	4	3	2	1	0	Reset
SPIRF	SPIMLS	SPIOV	SPITXIF	SPITDR	SPIRXIF	SPIRDR	SPIRS	40H

SPIRF: SPI SS pin Release Flag.

This bit is set when SS pin release & SPIRST as ‘1’.

SPIMLS: MSB or LSB first output /input Select.

“1” is MSB first output/input.

“0” is LSB first output/input.

SPIOV: Overflow flag.

When SPIRDR is set and next data already into shift register, this flag will be set.

It is clear by hardware, when SPIRDR is cleared.

SPITXIF: Transmit Interrupt Flag.

This bit is set when the data of the SPITXD register is downloaded to the shift register.

SPITDR: Transmit Data Ready.

When MCU finish writing data to SPITXD register, the MCU needs to set this bit to ‘1’ to inform the SPI module to send the data. After SPI module finishes sending the data from SPITXD, this bit will be cleared automatically.

SPIRXIF: Receive Interrupt Flag.

This bit is set after the SPIRXD is loaded with a newly receive data.

SPIRDR: Receive Data Ready.

The MCU must clear this bit after it gets the data from SPIRXD register. The SPI module is able to write new data into SPIRXD only when this bit is cleared.

SPIRS: Receive Start.

This bit set to “1” to inform the SPI module to receive the data into SPIRXD register.

Mnemonic: SPITXD **Address: F3h**

7	6	5	4	3	2	1	0	Reset
SPITXD[7:0]								00h

SPITXD[7:0]: Transmit data buffer.

Mnemonic: SPIRXD **Address: F4h**

7	6	5	4	3	2	1	0	Reset
SPIRXD[7:0]								00h

SPIRXD[7:0]: Receive data buffer.

P.S. MISO pin must be float when SS or CS no-active in slave mode.

15. KBI – Keyboard Interface

Keyboard interface (KBI) can be connected to a 4 x n matrix keyboard or any similar devices. It has 4 inputs with programmable interrupt capability on either high or low level. These 4 inputs can be the external interrupts to leave from the idle and stop modes.

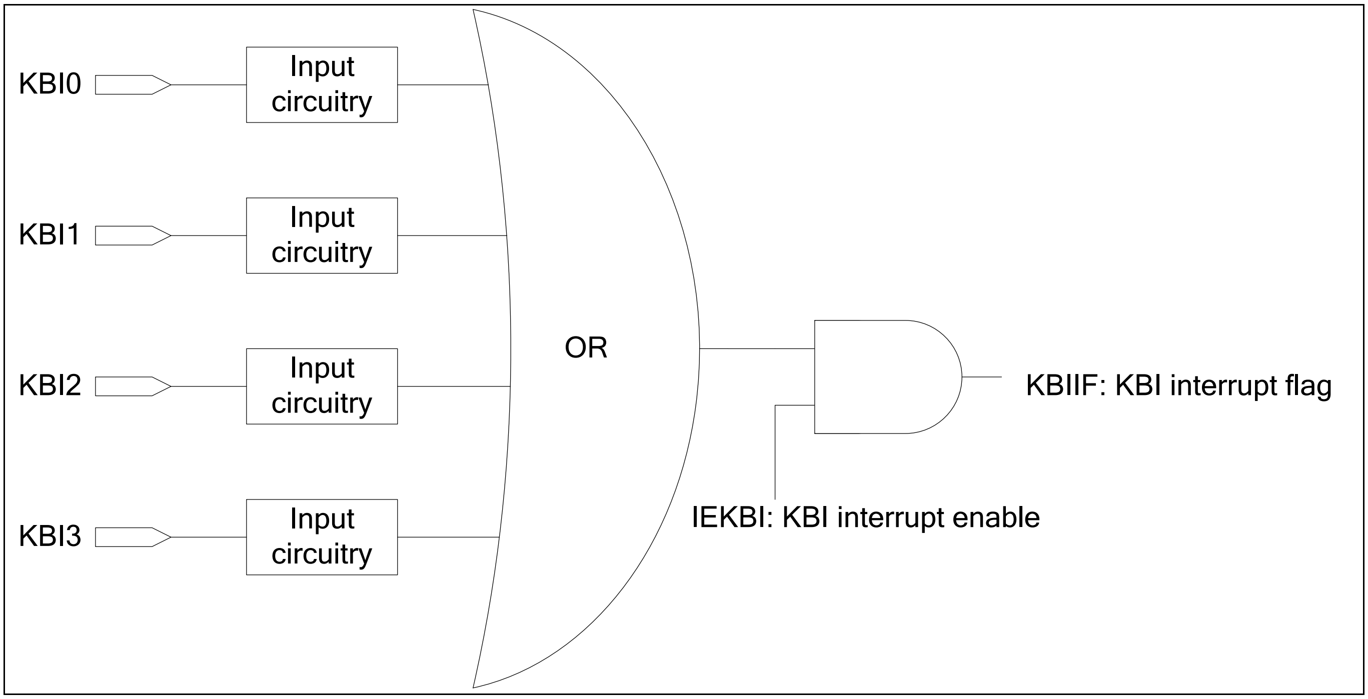


Fig. 15-1: keyboard interface block diagram

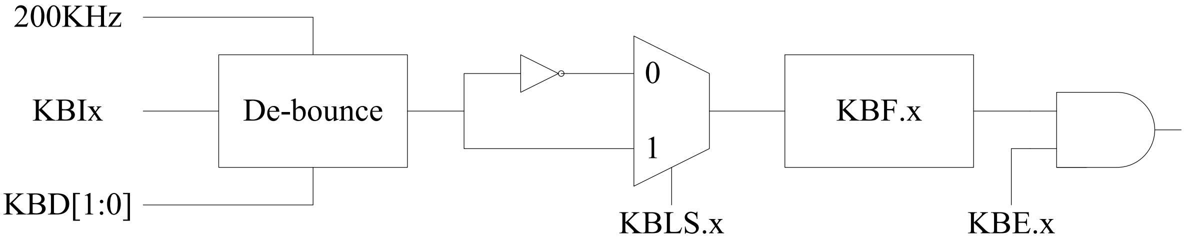


Fig. 15-2: keyboard input circuitry

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
KBI function											
KBLS	KBI level selection	93h	-	-	-	-	KBLS3	KBLS2	KBLS1	KBLS0	00H
KBE	KBI input enable	94h	-	-	-	-	KBE3	KBE2	KBE1	KBE0	00H
KBF	KBI flag	95h	-	-	-	-	KBF3	KBF2	KBF1	KBF0	00H
KBD	KBI De-bounce control register	96h	KBDEN	-	-	-	-	-	KBD1	KBD0	00H

Mnemonic: KBLS **Address: 93h**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	KBLS.3	KBLS.2	KBLS.1	KBLS.0	00h

KBLS.3: Keyboard Line 3 level selection bit

0 : enable a low level detection on KBI3.

1 : enable a high level detection on KBI3.

KBLS.2: Keyboard Line 2 level selection bit

0 : enable a low level detection on KBI2.

1 : enable a high level detection on KBI2.

KBLS.1: Keyboard Line 1 level selection bit

0 : enable a low level detection on KBI1.

1 : enable a high level detection on KBI1.

KBLS.0: Keyboard Line 0 level selection bit

0 : enable a low level detection on KBI0.

1 : enable a high level detection on KBI0.

Mnemonic: KBE **Address: 94h**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	KBE.3	KBE.2	KBE.1	KBE.0	00h

KBE.3: Keyboard Line 3 enable bit

0 : enable standard I/O pin.

1 : enable KBF.3 bit in KBF register to generate an interrupt request.

KBE.2: Keyboard Line 2 enable bit

0 : enable standard I/O pin.

1 : enable KBF.2 bit in KBF register to generate an interrupt request.

KBE.1: Keyboard Line 1 enable bit

0 : enable standard I/O pin.

1 : enable KBF.1 bit in KBF register to generate an interrupt request.

KBE.0: Keyboard Line 0 enable bit

0 : enable standard I/O pin.

1 : enable KBF.0 bit in KBF register to generate an interrupt request.

Mnemonic: KBF **Address: 95h**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	KBF.3	KBF.2	KBF.1	KBF.0	00h

KBF.3: Keyboard Line 3 flag

This is set by hardware when KBI3 detects a programmed level.

It generates a Keyboard interrupt request if KBE.3 is also set. It must be cleared by software.

KBF.2: Keyboard Line 2 flag

This is set by hardware when KBI2 detects a programmed level.

It generates a Keyboard interrupt request if KBE.2 is also set. It must be cleared by software.

KBF.1: Keyboard Line 1 flag

This is set by hardware when KBI1 detects a programmed level.

It generates a Keyboard interrupt request if KBE.1 is also set. It must be cleared by software.

KBF.0: Keyboard Line 0 flag

This is set by hardware when KBI0 detects a programmed level.

It generates a Keyboard interrupt request if KBE.0 is also set. It must be cleared by software.

Mnemonic: KBD **Address: 96H**

7	6	5	4	3	2	1	0	Reset
KBDEN	-	-	-	-	-	KBD.1	KBD.0	00H

KBDEN: Enable KBI de-bounce function. The default KBI function is enabled.

KBDEN = 0, enable KBI de-bounce function. The de-bounce time is selected by KBD [1:0].

KBDEN = 1, disable KBI de-bounce function. The KBI input pin without de-bounce mechanism.

KBD[1:0]: Select KBI de-bounce time. If KBDEN = “0”, the default de-bounce time is 320 ms.

KBD[1:0] = 00, the de-bounce time is 320 ms.

KBD[1:0] = 01, the de-bounce time is 160 ms.

KBD[1:0] = 10, the de-bounce time is 80 ms.

KBD[1:0] = 11, the de-bounce time is 40 ms.

16. LVI & LVR – Low Voltage Interrupt and Low Voltage Reset

The interrupt vector 63h.

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
LVI function											
RSTS	Reset status register	A1h	-	LVRL PINTF	LVRL PF	PDRF	WDTF	SWRF	LVRF	PORF	00H
LVC	Low voltage control	E6h	LVI_E N	-	LVRE	LVIF	-	-	LVIS[1:0]		20H

Mnemonic: RSTS **Address: A1h**

7	6	5	4	3	2	1	0	Reset
-	LVRL PINTF	LVRLP F	PDRF	WDTF	SWRF	LVRF	PORF	00H

LVRLPINTF: “Internal” Low voltage reset flag.

When MCU is reset by LVR_LP_INT, LVRLPINTF flag will be set to one by hardware.

This flag clear by software.

Note: LVRLPINT(VIL) = 1.45V

LVRLPF: Low voltage reset(Low Power) flag.

When MCU is reset by LVR(Low Power), LVRLPF flag will be set to one by hardware.

This flag clear by software.

PDRF: Pad reset flag.

When MCU is reset by reset pad, PDRF flag will be set to one by hardware. This flag clear by software.

LVRF: Low voltage reset flag.

When MCU is reset by LVR, LVRF flag will be set to one by hardware. This flag clear by software.

PORF: Power on reset flag.

When MCU is reset by POR, PORF flag will be set to one by hardware. This flag clear by software.

Mnemonic: LVC **Address: E6h**

7	6	5	4	3	2	1	0	Reset
LVI_EN	-	LVRE	LVIF	-	-	LVIS[1:0]		20H

LVI_EN: Low voltage interrupt function enable bit.

LVI_EN = 0 - disable low voltage detect function.

LVI_EN = 1 - enable low voltage detect function.

LVRE: External low voltage reset function enable bit.

LVRE = 0 - disable external low voltage reset function.

LVRE = 1 - enable external low voltage reset function.

LVIF: Low Voltage interrupt Flag

LVIS LVI level select:

00: 2.4V

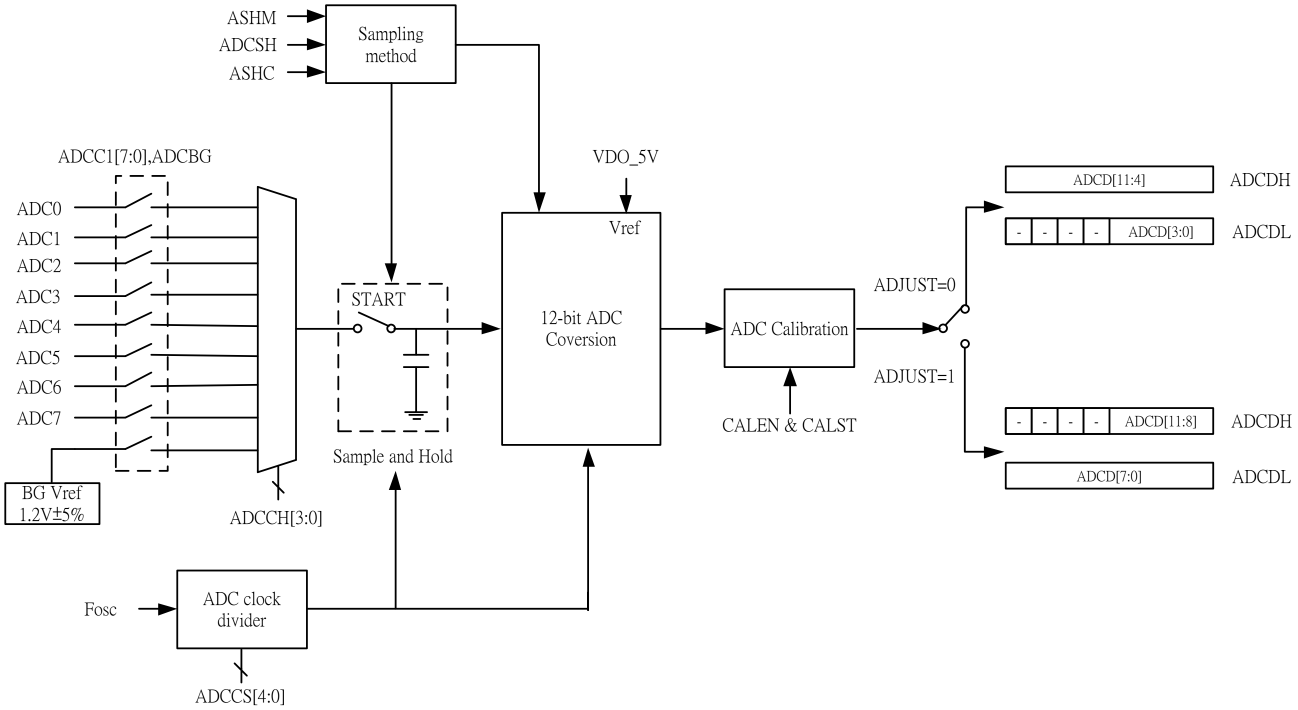
01: 2.6V

10: 3.2V

11: 4.0V

17. 12-bit Analog-to-Digital Converter (ADC)

The OB38S003 provides nine channels 12-bit ADC. The Digital output DATA [11:0] were put into ADCD [11:0].



The ADC SFR show as below:

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
SADC											
ADCC1	ADC Control 1	ABH	ADC7EN	ADC6EN	ADC5EN	ADC4EN	ADC3EN	ADC2EN	ADC1EN	ADC0EN	00H
ADCC2	ADC Control 2	ACH	START	ADJUST	ASHM	ASHC	ADCCH[3:0]				0FH
ADCDH	ADC Data High	ADH	ADCDH [7:0]								00H
ADCDL	ADC Data Low	AEH	ADCDL [7:0]								00H
ADCCS	ADC Clock Select	AFH	ADCBGE	-	-	ADCCS[4:0]					00H
ADCSH	ADC Sample and Hold Time	EFH	ADCSH[7:0]								00H

ADC Calibration:

Mnemonic	Description	Indirect	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
ADCCAL	ADC Calibration	FFFFH	-	-	-	-	-	-	CALEN	CALST	02H

Mnemonic: ADCC1 **Address: ABH**

7	6	5	4	3	2	1	0	Reset
ADC7EN	ADC6EN	ADC5EN	ADC4EN	ADC3EN	ADC2EN	ADC1EN	ADC0EN	00H

ADC7EN: ADC channel 7 enable.

0: Disable ADC channel 7

1: Enable ADC channel 7

ADC6EN: ADC channel 6 enable.

0: Disable ADC channel 6

1: Enable ADC channel 6

ADC5EN: ADC channel 5 enable.

0: Disable ADC channel 5

1: Enable ADC channel 5

ADC4EN: ADC channel 4 enable.

0: Disable ADC channel 4

1: Enable ADC channel 4

ADC3EN: ADC channel 3 enable.

0: Disable ADC channel 3

1: Enable ADC channel 3

ADC2EN: ADC channel 2 enable.

0: Disable ADC channel 2

1: Enable ADC channel 2

ADC1EN: ADC channel 1 enable.

0: Disable ADC channel 1

1: Enable ADC channel 1

ADC0EN: ADC channel 0 enable.

0: Disable ADC channel 0

1: Enable ADC channel 0

Mnemonic: ADCC2 **Address: ACH**

7	6	5	4	3	2	1	0	Reset
START	ADJUST	ASHM	ASHC	ADCCH[3:0]				0FH

START: When this bit is set, the ADC will be start single-shot conversion.

ADJUST: Adjust the format of ADC conversion DATA.

0: (default value)

ADC data high byte ADCDH [7:0] = ADCD [11:4].

ADC data low byte ADCDL [3:0] = ADCD [3:0].

1:

ADC data high byte ADCDH [3:0] = ADCD [11:8].

ADC data low byte ADCDL [7:0] = ADCD [7:0].

ASHM: ADC sample and hold mode :

ASHM=0: ADC sampling time is controlled by hardware.

ASHM=1: ADC sampling time is controlled by firmware.

ASHC: ADC sample and hold control bit. This control bit for ASHM=1.

ASHC=0: Disable ADC sampling.

ASHC=1: Enable ADC sampling.

ADCCH[3:0]: ADC channel select.

ADCCH [3:0]	Channel
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	Bandgap reference voltage
Others	-

ADJUST = 0:

Mnemonic: ADCDH **Address: ADH**

7	6	5	4	3	2	1	0	Reset
ADCD[11:4]								00H

Mnemonic: ADCDL **Address: AEH**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	ADCD[3:0]				00H

ADJUST = 1:

Mnemonic: ADCDH **Address: ADH**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	ADCD[11:8]				00H

Mnemonic: ADCDL **Address: AEH**

7	6	5	4	3	2	1	0	Reset
ADCD[7:0]								00H

ADCD[11:0]: ADC data register.

Mnemonic: ADCSH **Address: EFH**

7	6	5	4	3	2	1	0	Reset
ADCSH[7:0]								00H

ADCSH[7:0]: ADC sample and hold time register. This register for ASHM=0.

Mnemonic: ADCCS **Address: AFH**

7	6	5	4	3	2	1	0	Reset
ADCBGE	-	-	ADCCS[4:0]					00H

ADCBGE: ADC bandgap reference voltage channel enable

0: Disable

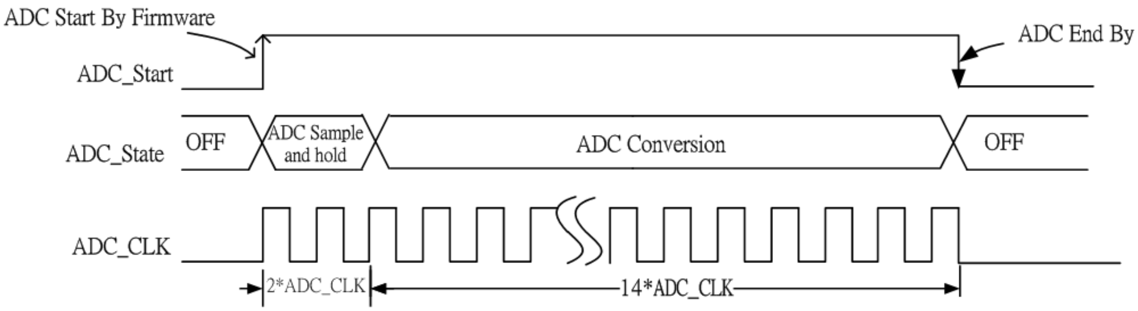
1: Enable

ADCCS[4:0]: ADC clock select.

$$ADC\_Clock = \frac{\mathrm{Fclk}}{6 \times (ADCCS[4:0]+1)}$$

If ASHM=0 , ADCSH=0x00:

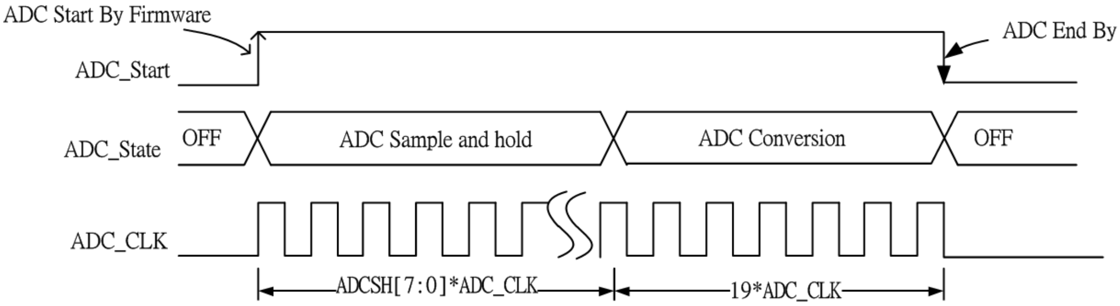
$$ADC\_Conversion\_Rate = \frac{\mathrm{ADC\_Clock}}{16}$$



If ASHM=0 , ADCSH≠0x00:

$$ADC\_Conversion\_Rate = \frac{\mathrm{ADC\_Clock}}{19+\mathrm{ADCSH}[7:0]}$$

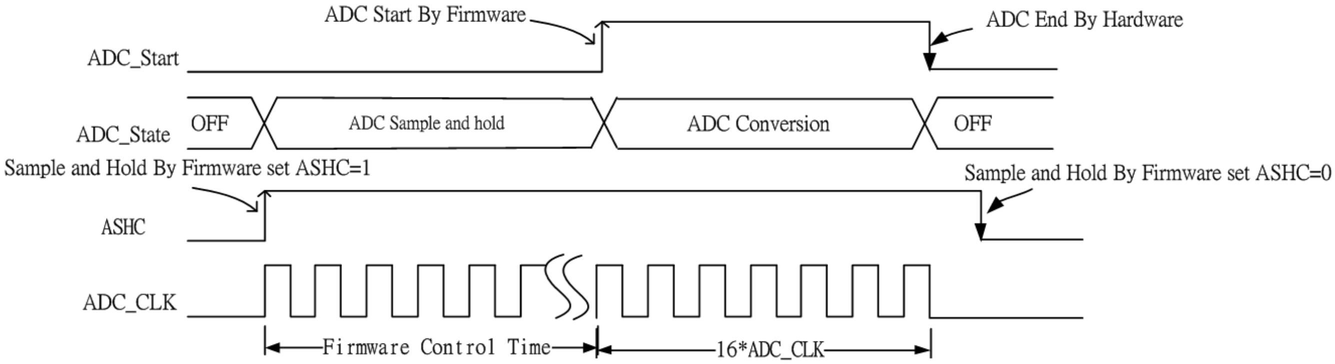
※Sample and hold time by hardware control decisions.



If ASHM=1:

$$ADC\_Conversion\_Rate = \frac{\mathrm{ADC\_Clock}}{16} + F/W\_Sampling\_Time$$

※Sample and hold time by firmware control decisions.



Mnemonic: ADCCAL **Indirect Address: FFFFH**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	-	-	CALEN	CALST	02H

CALEN: ADC calibration enable

0: Disable

1: Enable

※ If CALEN and ADCxEN both set to “1”, ADC execute conversion according to the calibrated parameter. It will cost some conversion time to complete ADC conversion.

CALST: ADC calibration start

0: Calibration complete (Hardware will clear this bit automatically to indicate the calibration has been finished)

1: Start calibration (Software set this bit to start calibration)

Calibration for sample code:
C language:

```
unsigned char xdata ADCCAL   _at_ 0xFFFF;
void ADC_Calibration(bit bEnable)
{
   if(bEnable==1)
   {
        ADCCAL=0x00;   // Set ADCCAL=0x00;
        ADCCAL=0x03;      // Set ADCCAL=0x03;
        while((ADCCAL&0x01)==0x01);      // End wait for ADC calibration
   }
}
void main(void)
{
        :
        :
        ADC_Calibration(1);          //ADC Calibration
}
```

Assembly language:

```
ADCCAL equ 0xFFFF
    :
ADC_Calibration:
    mov DPTR,#ADCCAL
    mov A,#0x00
    movx @DPTR,A         ;set ADCCAL=0x00;
    mov A,#0x03
    movx @DPTR,A         ;set ADCCAL=0x03;
WAIT:
    movx A,@DPTR
    anl A,#0x01
    jnz WATT            ; End wait for ADC calibration
    ret
```

18. In-System Programming (Internal ISP)

The OB38S003 can generate ROM control signal by internal hardware circuit. Users utilize control register, address register and data register to perform the ISP function without removing the OB38S003 from the system. The OB38S003 provides internal ROM control signals which can do program /protect functions. User need to design and use any kind of interface which OB38S003 can input data. User then utilize ISP service program to perform the program /protect functions.

18.1 ISP service program

The ISP service program is a user developed firmware program which resides in the ISP service program space. After user developed the ISP service program, user then determine the size of the ISP service program. User need to program the ISP service program in the OB38S003 for the ISP purpose.

The ISP service programs were developed by user so that it should includes any features which relates to the programming function as well as communication protocol between OB38S003 and host device which output data to the OB38S003. For example, if user utilize UART interface to receive/transmit data between OB38S003 and host device, the ISP service program should include baud rate, checksum or parity check or any error-checking mechanism to avoid data transmission error.

The ISP service program can be initiated under OB38S003 active or idle mode. It can not be initiated under power down mode.

18.2 Lock Bit (N)

The Lock Bit N has two functions: one is for service program size configuration and the other is to lock the ISP service program space from program function.

The ISP service program space address range $1C00 to $1FFF. It can be divided as blocks of N*128 byte. (N=0 to 8). When N=0 means no ISP function, all of 8KB programmable ROM can be used as program memory. When N=1 means ISP service program occupies 128 byte while the rest of 7.875K byte programmable ROM can be used as program memory. The maximum ISP service program allowed is 1K byte when N=8. Under such configuration, the usable program memory space is 7K byte.

After N determined, OB38S003 will reserve the ISP service program space downward from the top of the program address $3FFF. The start address of the ISP service program located at $1x00 while x is depending on the lock bit N. Please see Table 18-1. program memory diagram for this ISP service program space structure.

If the programmable ROM not has been protected, the content of ISP service program still can be read. If the programmable ROM has been protected, the overall content of programmable ROM program memory space including ISP service program space can not be read.

Table 18-1: ISP code area.

N	ISP service program address
0	No ISP service program
1	128 bytes ($1F80h ~ $1FFFh)
2	256 bytes ($1F00h ~ $1FFFh)
3	384 bytes ($1E80h ~ $1FFFh)
4	512 bytes ($1E00h ~ $1FFFh)
5	640 K bytes ($1D80h ~ $1FFFh)
6	768 K bytes ($1D00h ~ $1FFFh)
7	896 K bytes ($1C80h ~ $1FFFh)
8	1.0 K bytes ($1C00h ~ $1FFFh)

ISP service program configurable in N*128byte (N= 0 ~ 8)

18.3 Program the ISP Service Program

After Lock Bit N is set and ISP service program been programmed, the ISP service program memory will be protected (locked) automatically. The lock bit N has its own program timing. If user needs to erase the locked ISP service program, he can do it by writer only. User can not change ISP service program when OB38S003 was in system.

18.4 Initiate ISP Service Program

To initiate the ISP service program is to load the program counter (PC) with start address of ISP service program and execute it. There are four ways to do so:

(1) Blank reset. Hardware reset with first programmable ROM address blank ($0000=#FFH) will load the PC with start address of ISP service program. The hardware reset includes MAX810 (power on reset) and external pad reset. The hardware will issue a strobe window about 256us after hardware reset.

(2) Execute jump instruction can load the start address of the ISP service program to PC.

(3) Enter's ISP service program by hardware setting. User can force OB38S003 enter ISP service program by setting P1.6 " active low" during hardware reset period. The hardware reset includes MAX810 (power on reset) and external pad reset. The hardware will issue after hardware reset. In application system design, user should take care of the setting of P1.6 at reset period to prevent OB38S003 from entering ISP service program.

(4) Enter's ISP service program by hardware setting, the P1.1(RXD) will be detected the two clock signals during hardware reset period. The hardware reset includes MAX810 (power on reset) and external pad reset. The hardware will issue to detect 2 clock signals after hardware reset.

During the strobe window, the hardware will detect the status of P1.6/P1.1. If they meet one of above conditions, chip will switch to ISP mode automatically. After ISP service program executed, user need to reset the OB38S003, either by hardware reset or by WDT, or jump to the address $0000 to re-start the firmware program.

There are 8 kinds of entry mechanisms for user different applications. This entry method will select on the writer or ISP.

(1) First Address Blank. i.e. $0000 = 0xFF. And triggered by Internal reset signal.
(2) First Address Blank. i.e. $0000 = 0xFF. And triggered by PAD reset signal.
(3) P1.6 = 0. And triggered by Internal reset signal.
(4) P1.6 = 0. And triggered by PAD reset signal.
(5) P1.1 input 2 clocks. And triggered by Internal reset signal.
(6) P1.1 input 2 clocks. And triggered by PAD reset signal.

18.5 ISP register – TAKEY, IFCON, ISPFAH, ISPFAL, ISPFH, ISPFL and ISPFC

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
ISP function											
TAKEY	Time Access Key register	F7h	TAKEY [7:0]								00H
IFCON	Interface Control register	8Fh	-	CDPR	-	-	-	-	-	ISPE	00H
ISPFAH	ISP Address – High register	E1h	ISPFAH [7:0]								FFH
ISPFAL	ISP Address – Low register	E2h	ISPFAL [7:0]								FFH
ISPFDH	ISP Hogh Data register	EBh	ISPFDH[7:0]								FFH
ISPFDL	ISP Low Data register	E3h	ISPFDL[7:0]								FFH
ISPFC	ISP Control register	E4h	EMF1	-	EMF3	EMF4	-	ISPF.2	ISPF.1	ISPF.0	00H

Mnemonic: TAKEY **Address: F7H**

7	6	5	4	3	2	1	0	Reset
TAKEY [7:0]								00H

ISP enable bit (ISPE) is read-only by default, software must write three specific values 55h, AAh and 5Ah sequentially to the TAKEY register to enable the ISPE bit write attribute. That is:

```
MOV TAKEY, #55h
MOV TAKEY, #0AAh
MOV TAKEY, #5Ah
```

Mnemonic: IFCON **Address: 8FH**

7	6	5	4	3	2	1	0	Reset
-	CDPR	-	-	-	-	-	ISPE	00H

The bit 0 (ISPE) of IFCON is ISP enable bit. User can enable overall OB38S003 ISP function by setting ISPE bit to 1, to disable overall ISP function by set ISPE to 0. The function of ISPE behaves like a security key. User can disable overall ISP function to prevent software program be modified accidentally. ISP registers ISPFAH, ISPFAL, ISPFDH, ISPFDL and ISPFC are read-only by default. Software must be set ISPE bit to 1 to enable these 5 registers write attribute.

Mnemonic: ISPFAH **Address: E1H**

7	6	5	4	3	2	1	0	Reset
ISPFAH7	ISPFAH6	ISPFAH5	ISPFAH4	ISPFAH3	ISPFAH2	ISPFAH1	ISPFAH0	FFH

ISPFAH [5:0]: ISP address-high for ISP function

Mnemonic: ISPFAL **Address: E2H**

7	6	5	4	3	2	1	0	Reset
ISPFAL7	ISPFAL6	ISPFAL5	ISPFAL4	ISPFAL3	ISPFAL2	ISPFAL1	ISPFAL0	FFH

ISPFAL [7:0]: ISP address-Low for ISP function

The ISPFAH & ISPFAL provide the 16-bit memory address for ISP function. The memory address should not include the ISP service program space address. If the memory address indicated by ISPFAH & ISPFAL registers overlay with the ISP service program space address, the program of ISP function executed thereafter will have no effect.

Mnemonic: ISPFDH **Address: EBH**

7	6	5	4	3	2	1	0	Reset
ISPFDH7	ISPFDH6	ISPFDH5	ISPFDH4	ISPFDH3	ISPFDH2	ISPFDH1	ISPFDH0	FFH

ISPFDH [7:0]: ISP data for ISP function.

The ISPFDH provide the 8-bit data register for ISP function.

Mnemonic: ISPFDL **Address: E3H**

7	6	5	4	3	2	1	0	Reset
ISPFDL7	ISPFDL6	ISPFDL5	ISPFDL4	ISPFDL3	ISPFDL2	ISPFDL1	ISPFDL0	FFH

ISPFDL [7:0]: ISP data for ISP function.

The ISPFDL provide the 8-bit data register for ISP function.

Mnemonic: ISPFC **Address: E4H**

7	6	5	4	3	2	1	0	Reset
EMF1	-	EMF3	EMF4	-	ISPF[2]	ISPF[1]	ISPF[0]	00H

EMF1: Entry mechanism (1) flag, clear by reset. (Read only)

EMF3: Entry mechanism (3) flag, clear by reset. (Read only)

EMF4: Entry mechanism (4) flag, clear by reset. (Read only)

ISPF [2:0]: ISP function select bit.

ISPF[2:0]	ISP function
000	1-byte program ROM
001	2-byte program ROM
010	Chip protect
011	Write EEPROM
100	Read EEPROM
101	Write option
110	Read option
111	reserved

The choice ISP function will start to execute once the software write data to ISPFC register.

To perform byte program ISP function, user need to specify ROM address at first.

ISPF[2:0]=000 (1-byte program ROM)

EX1: ISP service program to do the 1-byte program ROM - to program #22H to the address $1004H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah          ; enable ISPE write attribute
MOV IFCON, #01H          ; enable  ISP function
MOV ISPFAH, #10H         ; set flash address-high, 10H
MOV ISPFAL, #04H         ; set flash address-low, 04H
MOV ISPFDL, #22H         ; set flash data to be programmed, data = 22H
MOV ISPFC, #00H          ; start to program #22H to the flash address $1004H
```

ISPF[2:0]=000 (1-byte program ROM)

EX2: ISP service program to do the 1-byte program ROM - to program #33H to the address $1005H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah          ; enable ISPE write attribute
MOV IFCON, #01H          ; enable  ISP function
MOV ISPFAH, #10H         ; set flash address-high, 10H
MOV ISPFAL, #05H         ; set flash address-low, 05H
MOV ISPFDL, #33H         ; set flash data to be programmed, data = 33H
MOV ISPFC, #00H          ; start to program #33H to the flash address $1005H
```

ISPF[2:0]=001 (2-byte program ROM)

EX3: ISP service program to do the 2-byte program ROM - to program #3322H to the address $1004H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah          ; enable ISPE write attribute
MOV IFCON, #01H          ; enable  ISP function
MOV ISPFAH, #10H         ; set flash address-high, 10H
MOV ISPFAL, #04H         ; set flash address-low, 04H
MOV ISPFDL, #22H         ; set flash data to be programmed, data = 22H
MOV ISPFDH, #33H         ; set flash data to be programmed, data = 33H
MOV ISPFC, #01H          ; start to program #3322H to the flash address $1004H
```

ISPF[2:0]=011 (EEPROM write)

EX4: ISP service program to do the EEPROM write - to program #22H to the address $0004H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah          ; enable ISPE write attribute
MOV IFCON, #01H          ; enable  ISP function
MOV ISPFAH, #00H         ; set flash address-high, 00H
MOV ISPFAL, #04H         ; set flash address-low, 04H
MOV ISPFDL, #22H         ; set flash data to be programmed, data = 22H
MOV ISPFC, #03H          ; start to program #22H to the EEPROM address $0004H
```

ISPF[2:0]=011 (EEPROM write)

EX5: ISP service program to do the EEPROM write - to program #33H to the address $0005H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah          ; enable ISPE write attribute
MOV IFCON, #01H          ; enable  ISP function
MOV ISPFAH, #00H         ; set flash address-high, 00H
MOV ISPFAL, #05H         ; set flash address-low, 05H
MOV ISPFDL, #33H         ; set flash data to be programmed, data = 33H
MOV ISPFC, #03H          ; start to program #33H to the EEPROM address $0005H
```

ISPF[2:0]=100 (EEPROM read)

EX6: ISP service program to do the EEPROM read - to read EEPROM program the address $0004H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah         ; enable ISPE write attribute
MOV IFCON, #01H         ; enable  ISP function
MOV ISPFAH, #00H        ; set flash address-high, 00H
MOV ISPFAL, #04H        ; set flash address-low, 04H
MOV ISPFC, #04H         ; start to program read the EEPROM address $0004H to ISPFDL
MOV A, ISPFDL           ; read ISPFDL and store it in Accumulator
```

ISPF[2:0]=101 (write option)

EX7: ISP service program to do the write option - to program #22H to the address $0004H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah         ; enable ISPE write attribute
MOV IFCON, #01H         ; enable  ISP function
MOV ISPFAH, #00H        ; set flash address-high, 00H
MOV ISPFAL, #04H        ; set flash address-low, 04H
MOV ISPFDL, #22H        ; set flash data to be programmed, data = 22H
MOV ISPFC, #05H         ; start to program #22H to the flash address $0004H
```

ISPF[2:0]=101 (write option)

EX8: ISP service program to do the write option - to program #33H to the address $0005H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah         ; enable ISPE write attribute
MOV IFCON, #01H         ; enable  ISP function
MOV ISPFAH, #00H        ; set flash address-high, 00H
MOV ISPFAL, #05H        ; set flash address-low, 05H
MOV ISPFDL, #33H        ; set flash data to be programmed, data = 33H
MOV ISPFC, #05H         ; start to program #33H to the flash address $0005H
```

ISPF[2:0]=110 (read option)

EX9: ISP service program to do the option read - to read option program the address $0004H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah         ; enable ISPE write attribute
MOV IFCON, #01H         ; enable  ISP function
MOV ISPFAH, #00H        ; set flash address-high, 00H
MOV ISPFAL, #04H        ; set flash address-low, 04H
MOV ISPFC, #06H         ; start to program read the option address $0004H to ISPFDL
MOV A, ISPFDL           ; read ISPFDL and store it in Accumulator
```

EX10: ISP service program to do the read program ROM - to read program ROM the address $1004H

```
CLR A
MOV DPTR,#1004h         ;
MOVC A,@A+DPTR          ; read the data of address $1004 and store it in Accumulator
```

EX11: ISP service program to do the read program ROM - to read program ROM the address $1005H

```
CLR A
MOV DPTR,#1005h         ;
MOVC A,@A+DPTR          ; read the data of address $1005 and store it in Accumulator
```

19. Comparator

OB38S003 had integrated two Comparator module on chip. This module supports Comparator modes individually according to user's configuration. When Comparator Mode enabled, an internal reference voltage is available to be configured on comparator terminals.

Comparator SFRs as follows:

Mnemonic	Description	Addr	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
Comparator											
OpPin	OpCmp Pin Select	F6h	-	Cmp0_En	C0PosVBG	C0PosPad	-	Cmp1_En	C1PosVBG	C1PosPad	00h
Cmp0CON	Comparator_0 control	FEh	Hys0En	Cmp0o	CMF0MS[1:0]		CMF0	Cmp0 OutEN	-	-	00h
Cmp1CON	Comparator_1 control	FFh	Hys1En	Cmp1o	CMF1MS[1:0]		CMF1	Cmp1 OutEN	-	-	00h

Mnemonic: OpPin **Address: F6h**

7	6	5	4	3	2	1	0	Reset
-	Cmp0_En	C0PosVBG	C0PosPad	-	Cmp1_En	C1PosVBG	C1PosPad	00h

Cmp0_En : Cmp0 enable.

1: Comparator_0 circuit enables and switch to corresponding signal in multi-function pin P0.3/P0.4/P0.6 by HW automatically.

C0PosVBG : Select Comparator_0 positive input source

1: set positive input source as internal reference voltage (1.2V±5%)

C0PosPad: Select Comparator_0 positive input source

1: set positive input source as external pin

Cmp1_En : Cmp1 enable.

1: Comparator_1 circuit enables and switch to corresponding signal in multi-function pin P0.0/P0.1/P0.2 by HW automatically.

C1PosVBG: Select Comparator_1 positive input source

1: set positive input source as internal reference voltage (1.2V±5%)

C1PosPad: Select Comparator_1 positive input source

1: set positive input source as external pin

Setting table：

Cmpx_En	CxPosPad	CxPosVBG	CmpxOut_En	Comparator		
				CmpxPIn	CmpxNIn	CmpxOUT
0	X	X	X	IO	IO	IO
1	0	1	0	IO	CMP	IO
1	0	1	1	IO	CMP	CMP
1	1	0	0	CMP	CMP	IO
1	1	0	1	CMP	CMP	CMP

Mnemonic: Cmp0CON **Address:FEh**

7	6	5 4	3	2	1	0	Reset
Hys0En	Cmp0o	CMF0MS[1:0]	CMF0	Cmp0 OutEN	-	-	00h

Hys0En: Hysteresis function enable

0: disable Hysteresis at comparator_0 input

1: enable

Cmp0o: Comparator_0 output (read only)

0: The positive input source was lower than negative input source

1: The positive input source was higher than negative input source

CMF0MS[1:0] : CMF0(Comparator_0 Flag) setting mode select

00: CMF0 will be set when comprator_0 output toggle

01: CMF0 will be set when comprator_0 output rising

10: CMF0 will be set when comprator_0 output falling

11: reserved

CMF0: Comparator_0 Flag

This bit is setting by hardware according to meet CMF0MS [1:0] select condition.

This bit must clear by software.

Cmp0OutEN: Comparator_0 Output Enable

0: Comparator_0 will not output to external Pin

1: Comparator_0 will output to external Pin

Mnemonic: Cmp1CON **Address:FFh**

7	6	5 4	3	2	1	0	Reset
Hys1En	Cmp1o	CMF1MS[1:0]	CMF1	Cmp1 OutEN	-	-	00h

Hys1En: Hysteresis function enable

0: disable Hysteresis at comparator_1 input

1: enable

Cmp1o: Comparator_1 output (read only)

0: The positive input source was lower than negative input source

1: The positive input source was higher than negative input source

CMF1MS[1:0] : CMF1(Comparator_1 Flag) setting mode select

00: CMF1 will be set when comprator_1 output toggle

01: CMF1 will be set when comprator_1 output rising

10: CMF1 will be set when comprator_1 output falling

11: reserved

CMF1: Comparator_1 Flag

This bit is setting by hardware according to meet CMF1MS [1:0] select condition.

This bit must clear by software.

Cmp1OutEN: Comparator_1 Output Enable

0: Comparator_1 will not output to external Pin

1: Comparator_1 will output to external Pin

Operating Conditions

Symbol	Description	Min.	Typ.	Max.	Unit.	Remarks
TA	Operating temperature	-40	25	85	℃	Ambient temperature under bias
VDD	Supply voltage	2.4	-	5.5	V	

DC Characteristics

TA = -40℃ to 85℃, VCC =2.4~ 5.5V

Symbol	Parameter	Valid	Min	Typical	Max	Units	Conditions
VIL1	Input Low-voltage	Port 0,1,3			0.2Vcc	V	
VIL2	Input Low-voltage	XTAL1			0.2Vcc	V	-
VIH1	Input High-voltage	Port 0,1,3	0.8Vcc			V	-
VIH2	Input High-voltage	XTAL1	0.8Vcc			V	-
IOL1	Sink Current (Open drain)	P0.0/P0.1/P0.2/P0.3 P0.4/P0.6/P1.0/P1.1 P1.4	20			mA	VOL=0.45V VCC=5V
			17			mA	VOL=0.45V VCC=3.3V
IOL2	Sink Current (Open drain)	P0.5/P0.7/P1.2/P1.3 P1.5/P1.6/P1.7/P3.0 P3.1	40			mA	VOL=0.45V VCC=5V
			28			mA	VOL=0.45V VCC=3.3V
IOH1	Source Current (Pull-Up)	Port 0,1,3	0.3			mA	VOH=2.6V VCC=5V
			0.09			mA	VOH=2.4V VCC=3.3V
IOH2	Source Current (Push-Pull)	P0.0/P0.1/P0.2/P0.3 P0.4/P0.6/P1.0/P1.1 P1.4	7			mA	VOH=4.6V VCC=5V
			5			mA	VOH=2.9V VCC=3.3V
IOH3	Source Current (Push-Pull)	P0.5/P0.7/P1.2/P1.3 P1.5/P1.6/P1.7/P3.0 P3.1	10			mA	VOH=4.6V VCC=5V
			7			mA	VOH=2.9V VCC=3.3V
CIO	Pin Capacitance	-	-	-	10	pF	Freq= 1MHz, Ta= 25℃
ICC	Power Supply Current	VCC=5V	-	5	6	mA	Active mode, IRC=16MHz 25 ℃
			-	4	5	mA	Idle mode, IRC=16MHz 25 ℃
			-	2	5	uA	Power down mode 25 ℃

LVI& LVR Characteristics

	LVR		
	Min	**Typical**	**Max**
2.0V	VIL=1.90V	VIL=2.00V	VIL=2.10V
2.2V	VIL=2.09V	VIL=2.20V	VIL=2.31V
2.4V	VIL=2.28V	VIL=2.40V	VIL=2.52V
3.0V	VIL=2.85V	VIL=3.00V	VIL=3.15V

	LVI		
	Min	**Typical**	**Max**
LVIS[1:0] = 00	VIL=2.28V	VIL=2.40V	VIL=2.52V
LVIS[1:0] = 01	VIL=2.47V	VIL=2.60V	VIL=2.73V
LVIS[1:0] = 10	VIL=3.04V	VIL=3.20V	VIL=3.36V
LVIS[1:0] = 11	VIL=3.80V	VIL=4.00V	VIL=4.20V

ADC Characteristics

	Symbol	**Test Condition**	**MIN**	**TYP**	**MAX**	**Unit**
Operation	VDD	VDD	3		5.5	V
Resolution					12	bit
Conversion time				$16t_{ADC}$		us
Sample rate				167k		Hz
Integral Non-Linearity Error	INL		-2		2	LSB
Differential Non-Linearity	DNL		-1		1	LSB
Clock frequency	ADCCLK				4	MHz

Comparator Characteristics

Symbol	**Description**	**Test Condition**		**MIN**	**TPY**	**MAX**	**Unit**
		V5V	**Condition**				
I_{OP}	Operating current	5	-	-	-	2	mA
-	Power Down Current	5	-	-	-	0.1	nA
-	Offset voltage	5	-	-10	-	+10	mV
V	Input voltage range	-	-	Vss	-	V5V	V
Tp	Propagation delay	5	△ Vin=10mV	-	4	6	us