Product List

OB38R08T1W28SP,
OB38R08T1W28EP,
OB38R08T1W24SP,
OB38R08T1W24EP,
OB38R08T1W20SP,
OB38R08T1W20EP,
OB38R08T1W16OP,
OB38R08T1W14OP,
OB38R08T1W10GP,

Description

The OB38R08T1 is a 1T (one machine cycle per clock) single-chip 8-bit microcontroller. It has 8KB +1KB embedded program memory, and executes all ASM51 instructions fully compatible with MCS-51.
OB38R08T1 contains 256B+256B on-chip RAM, up to 26 GPIOs (28L 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 OB38R08T1 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

OB38R08T1 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)

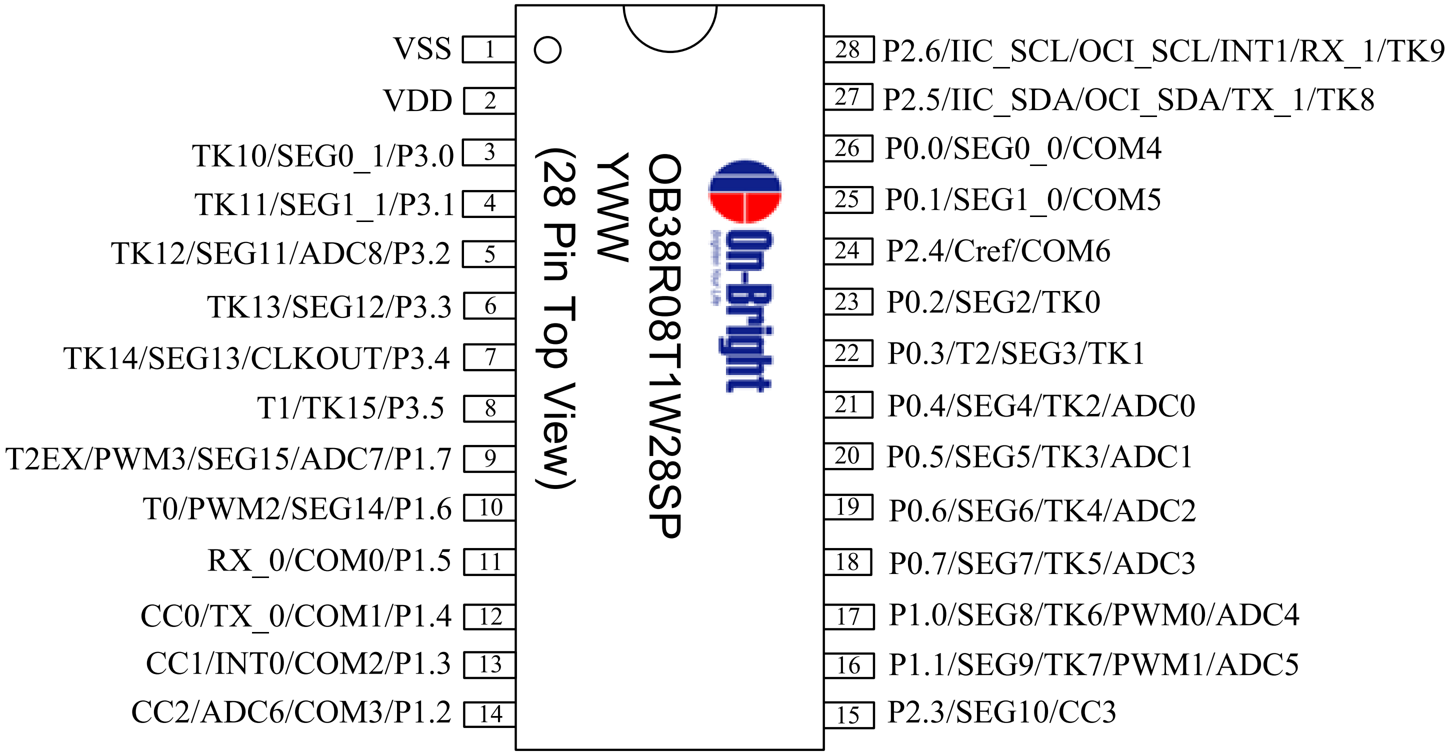
Postfix	Package
S	SOP (300 mil)
E	TSSOP(173 mil)
O	SOP (150 mil)
G	SSOP (150 mil)

Features

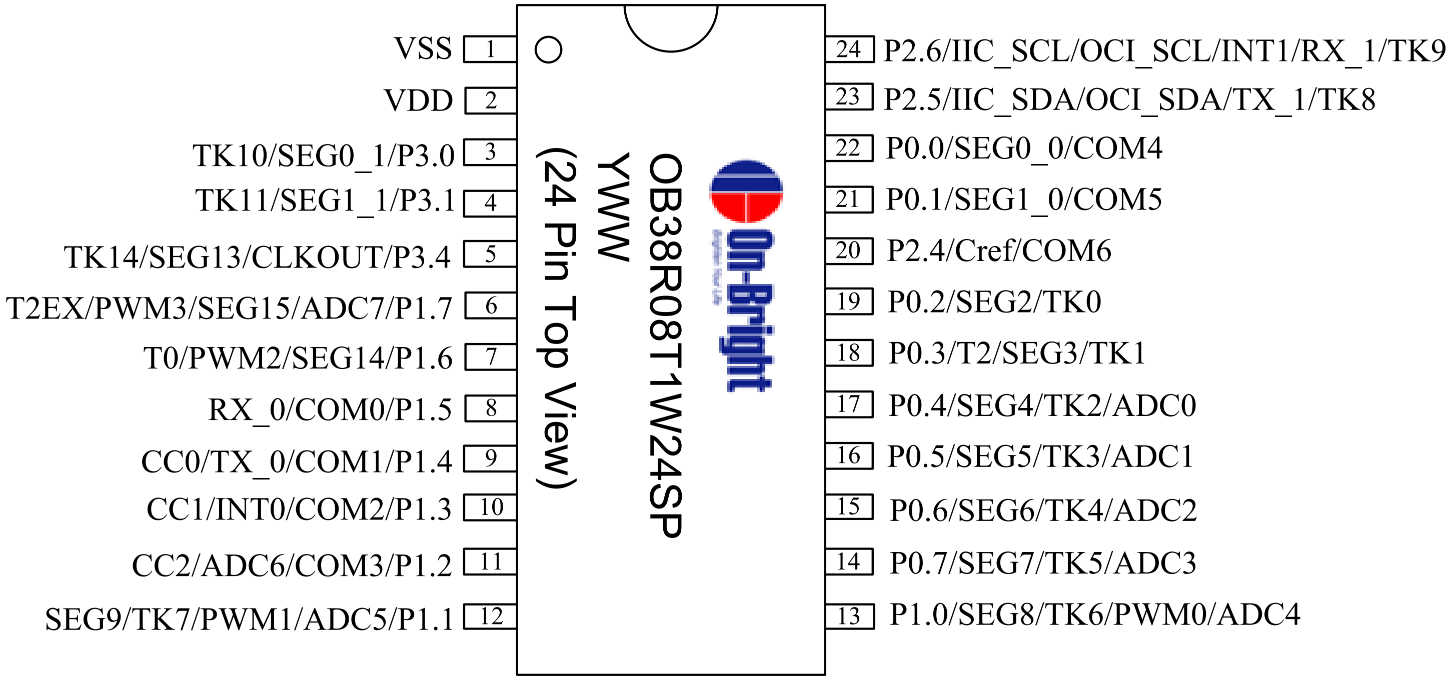
- Operating Voltage: 2.4V ~ 5.5V
- High speed architecture of 1 clock/machine cycle runs up to 16MHz.
- 1~8T can be switched on the fly.
- Instruction-set compatible with MCS-51.
- 16MHz Internal RC oscillator, with programmable clock divider
- 8KB +1KB on-chip program memory.
- 256 bytes SRAM as standard 8052, plus 256 bytes on-chip expandable SRAM.
- One serial peripheral interfaces in full duplex mode.
- Up to 16 touch sense inputs, support multiplexing I/O function.
- Lower power touch-key wakeup function.
- Additional Baud Rate Generator for Serial port.
- Three 16-bit Timer/Counters. (Timer 0,1,2）
- 4 channel 16-bit compare/capture functions
- Port 0~3, Up to 26 GPIO.
- External interrupt 0,1 with four priority levels
- Programmable watchdog reset and interrupt timer.
- One IIC interface. (Master/Slave mode)
- 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).
- 4-channel PWM with dead time
- 9 channel 12-bit ADC
- LED driver 7 COM*16 Segment, COM port sink current 80mA
- LVI/LVR (LVR deglitch 500ns)
  - POR and Programmable LVR(3.0 / 2.4 / 2.2 /2.0) & LVI (4.0 / 3.2 / 2.6/ 2.4)
- 2 external interrupt with rising/falling edge detection. (INT x 2)
- Power management unit for IDLE and power down modes.

Pin Configuration

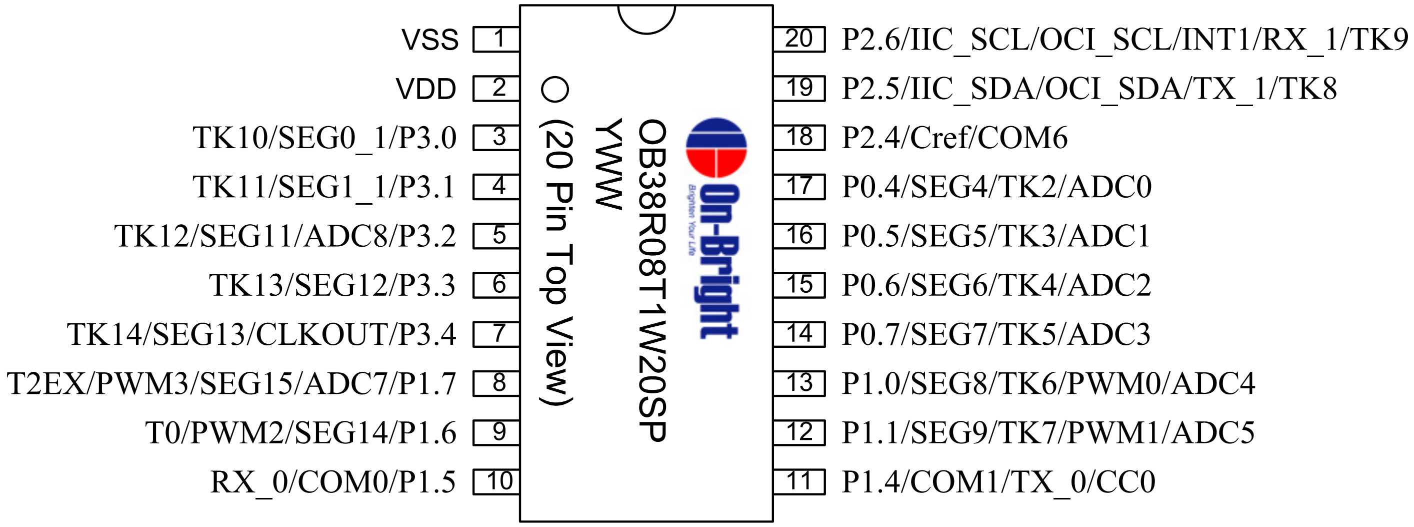
28 Pin SOP/TSSOP



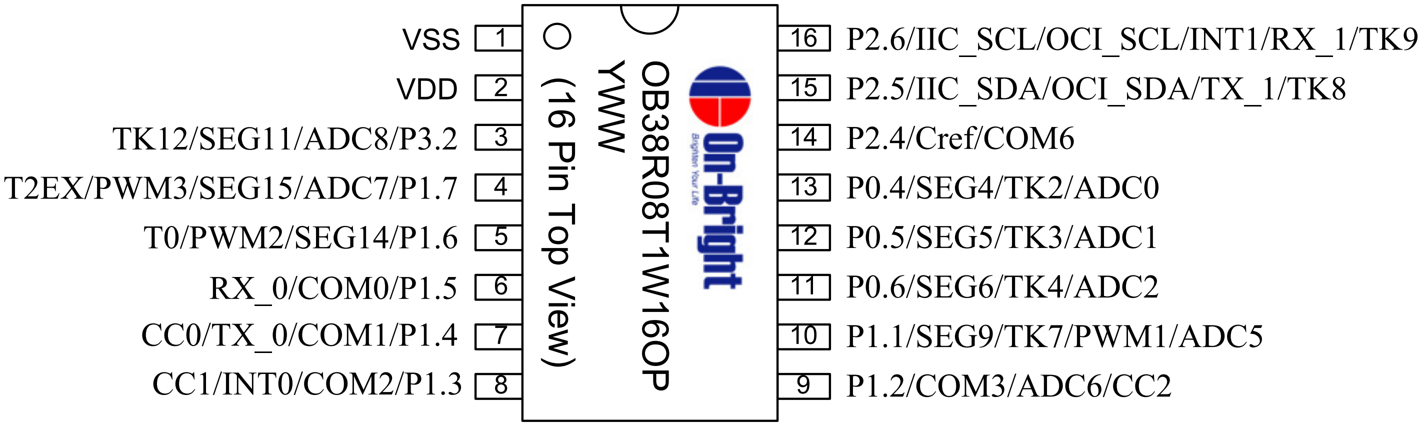
24 Pin SOP/TSSOP



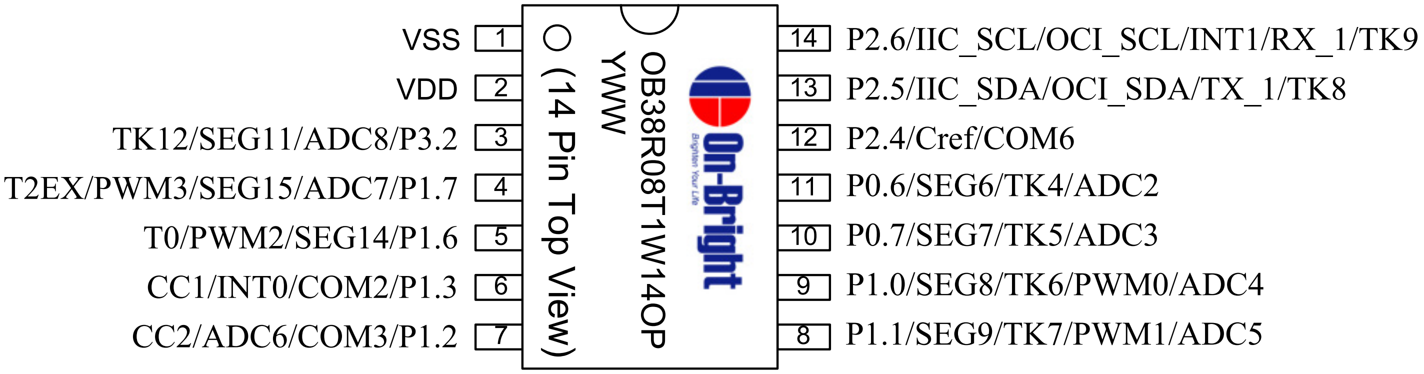
20 Pin SOP/TSSOP



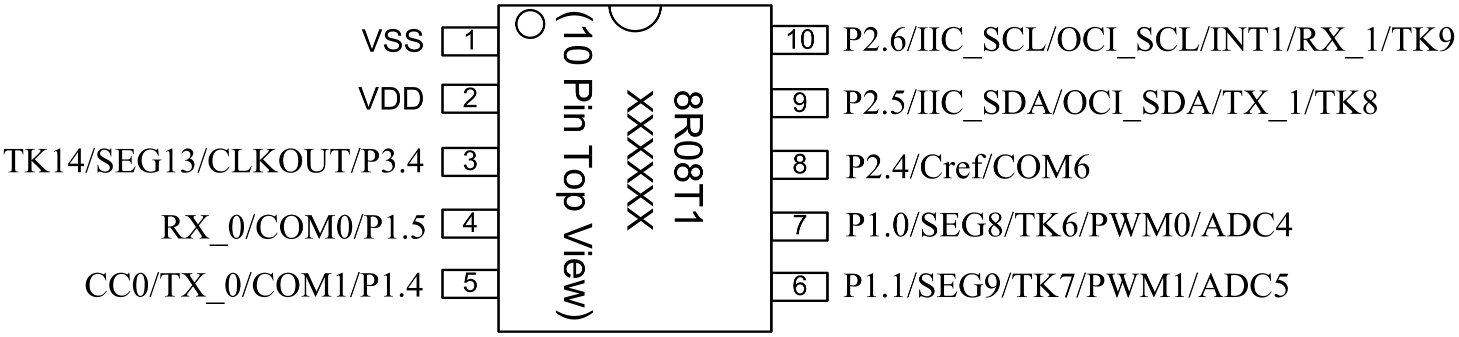
16 Pin SOP



14 Pin SOP



10 Pin SSOP


VSS
1
VDD
2
TK14/SEG13/CLKOUT/P3.4
3
RX_0/COM0/P1.5
4
CC0/TX_0/COM1/P1.4
5
8R08T1
XXXXXX
(10 Pin Top View)
10
P2.6/IIC_SCL/OCI_SCL/INT1/RX_1/TK9
9
P2.5/IIC_SDA/OCI_SDA/TX_1/TK8
8
P2.4/Cref/COM6
7
P1.0/SEG8/TK6/PWM0/ADC4
6
P1.1/SEG9/TK7/PWM1/ADC5

Pin Description

28L	24L	20L	16L	14L	10L	Symbol	I/O	Description
1	1	1	1	1	1	VSS	I	Ground
2	2	2	2	2	2	VDD	I	Power supply
3	3	3				P3.0/SEG0_1/TK10	I/O	Bit 0 of port 3 & LED driver segment 0_1 & Touch key 10
4	4	4				P3.1/SEG1_1/TK11	I/O	Bit 1 of port 3 & LED driver segment 1_1 & Touch key 11
5		5	3	3		P3.2/ADC8/SEG11/TK12	I/O	Bit 2 of port 3 & ADC input channel 8 & LED driver segment 11 & Touch key 12
6		6				P3.3/SEG12/TK13	I/O	Bit 3 of port 3 & LED driver segment 12 & Touch key 13
7	5	7			3	P3.4/CLKOUT/SEG13/TK14	I/O	Bit 4 of port 3 & Clock output & LED driver segment 13 & Touch key 14
8						P3.5/TK15/T1	I/O	Bit 5 of port 3 & Touch key 15 & Timer 1 external input
9	6	8	4	4		P1.7/ADC7/SEG15/PWM3/T2EX	I/O	Bit 7 of port 1 & ADC input channel 7 & LED driver segment 15 & PWM channel 3 & Timer 2 capture trigger
10	7	9	5	5		P1.6/SEG14/PWM2/T0	I/O	Bit 6 of port 1 & LED driver segment 14 & PWM Channel 2 & Timer 0 external input
11	8	10	6		4	P1.5/COM0/RX_0	I/O	Bit 5 of port 1 & LED driver common 0 & Serial interface channel_0 receive data
12	9	11	7		5	P1.4/COM1/TX_0/CC0	I/O	Bit 4 of port 1 & LED driver common 1 & Serial interface channel_0 transmit data & Timer 2 compare/capture channel 0
13	10		8	6		P1.3/COM2/INT0/CC1	I/O	Bit 3 of port 1 & LED driver common 2 & External interrupt 0 & Timer 2 compare/ capture channel 1
14	11		9	7		P1.2/COM3/ADC6/CC2	I/O	Bit 2 of port 1 & LED driver common 3 & ADC input channel 6 & Timer 2 compare/ capture channel 2
15						P2.3/SEG10/CC3	I/O	Bit 3 of port 2 & LED driver segment 10 & Timer 2 compare/capture channel 3
16	12	12	10	8	6	P1.1/SEG9/TK7/PWM1/ADC5	I/O	Bit 1 of port 1 & LED driver segment 9 & Touch key 7 & PWM channel 1 & ADC input channel 5
17	13	13		9	7	P1.0/SEG8/TK6/PWM0/ADC4	I/O	Bit 0 of port 1 & LED driver segment 8 & Touch key 6 & PWM channel 0 & ADC input channel 4

18	14	14		10		P0.7/SEG7/TK5/ADC3	I/O	Bit 7 of port 0 & LED driver segment 7 & Touch key 5 & ADC input channel 3
19	15	15	11	11		P0.6/SEG6/TK4/ADC2	I/O	Bit 6 of port 0 & LED driver segment 6 & Touch key 4 & ADC input channel 2
20	16	16	12			P0.5/SEG5/TK3/ADC1	I/O	Bit 5 of port 0 & LED driver segment 5 & Touch key 3 & ADC input channel 1
21	17	17	13			P0.4/SEG4/TK2/ADC0	I/O	Bit 4 of port 0 & LED driver segment 4 & Touch key 2 & ADC input channel 0
22	18					P0.3/SEG3/TK1/T2	I/O	Bit 3 of port 0 & LED driver segment 3 & Touch key 1 & Timer 2 external input
23	19					P0.2/SEG2/TK0	I/O	Bit 2 of port 0 & LED driver segment 2 & Touch key 0
24	20	18	14	12	8	P2.4/CREF/COM6	I/O	Bit 4 of port 2 & Touch key external capacitor & LED driver common 6
25	21					P0.1/SEG1_0/COM5	I/O	Bit 1 of port 0 & LED driver segment 1_0 & LED drIver common 5
26	22					P0.0/SEG0_0/COM4	I/O	Bit 0 of port 0 & LED driver segment 0_0 & LED driver common 4
27	23	19	15	13	9	P2.5/IIC_SDA/OCI_SDA/TX_1/TK8	I/O	Bit 5 of port 2 & IIC SDA & On-Chip Instrumentation Data I/O pin of ICE and ICP Functions & Serial interface channel_1 transmit data & Touch key 8
28	24	20	16	14	10	P2.6/IIC_SCL/OCI_SCL/INT1/RX_1/TK9	I/O	Bit 6 of port 2 & IIC SCL & On-Chip Instrumentation Clock I/O pin of ICE and ICP Functions & External interrupt 1 & Serial interface channel_1 receive data & Touch key 9

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	TKSW	WDTIC	**FF**
F0	B		TKPSSR	TKWKTRICNT				TAKEY	**F7**
E8				ISPFDH			INTDEG	ADCSH	**EF**
E0	ACC	ISPFAH	ISPFAL	ISPFDL	ISPFC	ENHIT	LVC	SWRES	**E7**
D8		PFCON	P3M0	P3M1		PWMC2	PWMDT0	PWMDT1	**DF**
D0	PSW	CCEN2	P0M0	P0M1	P1M0	P1M1	P2M0	P2M1	**D7**
C8	T2CON	CCCON	CRCL	CRCH	TL2	TH2	PWMMDL	PWMMDH	**CF**
C0	IRCON	CCEN	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3	**C7**
B8	IEN1	IP1	SRELH	TKSTATUS0	PWMD0L	PWMD0H	PWMD1L	PWMD1H	**BF**
B0	P3	PWMD2L	PWMD2H	PWMD3L	PWMD3H	PWMC	WDTRC	WDTK	**B7**
A8	IEN0	IP0	SRELL	ADCC1	ADCC2	ADCDH	ADCDL	ADCCS	**AF**
A0	P2	RSTS							**A7**
98	SCON	SBUF	IEN2	TKCON	TKRUNTIME	TKCHN	TKCDL	TKCDH	**9F**
90	P1	AUX		TKEN0	TKEN1			IRCON2	**97**
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	IFCON	**8F**
80	P0	SP	DPL	DPH		TKSTATUS1		PCON	**87**
Hex\Bin	X000	X001	X010	X011	X100	X101	X110	X111	**Bin/Hex**

Register map of xdata (External data memory):

Indirect Access Mode

Hex\Bin	X000	X001	X010	X011	X100	X101	X110	X111	Bin/Hex
FFF8								ADCCAL	FFFF
…									FFF7
FF20	LEDCLK								
FF18	COM5_CH	COM5_CL	COM6_CH	COM6_CL	COMEN	SEGEN0	SEGEN1	LEDCON	FF1F
FF10	COM1_CH	COM1_CL	COM2_CH	COM2_CL	COM3_CH	COM3_CL	COM4_CH	COM4_CL	FF17
FF08	COM4_AH	COM4_AL	COM5_AH	COM5_AL	COM6_AH	COM6_AL	COM0_CH	COM0_CL	FF0F
FF00	COM0_AH	COM0_AL	COM1_AH	COM1_AL	COM2_AH	COM2_AL	COM3_AH	COM3_AL	FF07
…									…
1F8	TK8TRICNT	TK9TRICNT	TK10TRICNT	TK11TRICNT	TK12TRICNT	TK13TRICNT	TK14TRICNT	TK15TRICNT	1FF
1F0	TK0TRICNT	TK1TRICNT	TK2TRICNT	TK3TRICNT	TK4TRICNT	TK5TRICNT	TK6TRICNT	TK7TRICNT	1F7
1E8	TK12TRIGH	TK12TRIGL	TK13TRIGH	TK13TRIGL	TK14TRIGH	TK14TRIGL	TK15TRIGH	TK15TRIGL	1EF
1E0	TK8TRIGH	TK8TRIGL	TK9TRIGH	TK9TRIGL	TK10TRIGH	TK10TRIGL	TK11TRIGH	TK11TRIGL	1E7
1D8	TK4TRIGH	TK4TRIGL	TK5TRIGH	TK5TRIGL	TK6TRIGH	TK6TRIGL	TK7TRIGH	TK7TRIGL	1DF
1D0	TK0TRIGH	TK0TRIGL	TK1TRIGH	TK1TRIGL	TK2TRIGH	TK2TRIGL	TK3TRIGH	TK3TRIGL	1D7
Hex\Bin	X000	X001	X010	X011	X100	X101	X110	X111	Bin/Hex

Note: Special Function Registers reset values and description for OB38R08T1

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 low byte
DPH	83h	00h	Data Pointer 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
IEN2	9Ah	00h	Interrupt Enable Register 2

Register	Location	Reset value	Description
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
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
WDT			
RSTS	A1h	00h	Reset status register
WDTRC	B6h	04H	Watchdog Timer Reset Control
WDTIC	FFh	00H	Watchdog Timer Interrupt Control
WDTK	B7h	00h	Watchdog timer refresh key.
TAKEY	F7h	00h	Time Access Key register
PWM			
PWMC	B5h	00H	PWM Control Register
PWMD0H	BDh	00H	PWM 0 Data High Register
PWMD0L	BCh	00H	PWM 0 Data Low Register
PWMD1H	BFh	00H	PWM 1 Data High Register
PWMD1L	BEh	00H	PWM 1 Data Low Register
PWMD2H	B2h	00H	PWM 2 Data High Register
PWMD2L	B1h	00H	PWM 2 Data Low Register
PWMD3H	B4h	00H	PWM 3 Data High Register
PWMD3L	B3h	00H	PWM 3 Data Low Register
PWMMDH	CFh	00H	PWM Max Data High Register
PWMMDL	CEh	FFH	PWM Max Data Low Register
PWMC2	DDh	00H	PWM Control 2
PWMDT0	DEh	00H	PWM 0 Dead Time
PWMDT1	DFh	00H	PWM 1 Dead Time
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

Register	Location	Reset value	Description
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	User define	Port 0
P1	90h	User define	Port 1
P2	A0h	User define	Port 2
P3	B0h	User define	Port 3
P0M0	D2h	User define	Port 0 output mode 0
P0M1	D3h	User define	Port 0 output mode 1
P1M0	D4h	User define	Port 1 output mode 0
P1M1	D5h	User define	Port 1 output mode 1
P2M0	D6h	User define	Port 2 output mode 0
P2M1	D7h	User define	Port 2 output mode 1
P3M0	DAh	User define	Port 3 output mode 0
P3M1	DBh	User define	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	EBh	FFh	ISP Data Low register
ISPFC	E4h	00h	ISP control register
TAKEY	F7h	00h	Time Access Key register
TOUCH KEY			
TKEN0	93h	00h	Touch Key Enable 0
TKEN1	94h	00h	Touch Key Enable 1
TKCON	9Bh	00h	Touch Key Control
TKCHN	9Dh	00h	Touch Key Channel Number

Register	Location	Reset value	Description
TKCDL	9Eh	00h	Touch Key Capture Data Low Byte
TKCDH	9Fh	00h	Touch Key Capture Data High Byte
TKSW	FEh	00h	Touch Key Switch
TKSTATUS0	BBh	00h	Touch Key Status 0
TKSTATUS1	85h	00h	Touch Key Status 1
TKPSSR	F2h	07h	Touch Key Samping Rate
TKWKTRICNT	F3h	02h	Touch Key Trigger Counter
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
IIC			
IICS	F8h	00h	IIC status register
IICCTL	F9h	03h	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
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
LED			
COMEN	0xFF1C	00h	LED COM Enable Register.
SEGEN0	0xFF1D	00h	LED SEG Enable 0 Register.
SEGEN1	0xFF1E	00h	LED SEG Enable 1 Register.
LEDCON	0xFF1F	00h	LED Control Register.
LEDCLK	0xFF20	00h	LED Clock Register.
ADC Caliration			
ADCCAL	0xFFFF	02h	ADC Calibration Register.

Function Description

1. General Features

OB38R08T1 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 +1KB programmable ROM via its writer or In-System Programming (ISP).

1.2 IO Pads

The OB38R08T1 has four I/O ports: Port 0~3. 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~P3 are with slew rate to reduce EMI. The IO pads can withstand 4KV ESD in human body mode guaranteeing the OB38R08T1 is quality in high electro-static environments.

1.3 Instruction timing Selection

The conventional 52-series MCUs are 12T, i.e., 12 oscillator clocks per machine cycle. OB38R08T1 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 Clock Out Selection

The OB38R08T1 can generate a clock out signal at P3.4. 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

OB38R08T1 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

OB38R08T1 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	-	LVRLPI NTF	-	-	WDT RF	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
-	LVRLPI NTF	-	-	WDTRF	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.

WDTRF: 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.

Table 1-1: Selection of clock source

Clock source
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 OB38R08T1 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 OB38R08T1 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 OB38R08T1 memory structure follows general 8052 structure. It is 8KB +1KB program memory.

3.1 Program Memory

The OB38R08T1 has 8KB +1KB 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 $3C00 to $3FFF. 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+1KB memory used as program memory. When N=1 means address $3F80 to $3FFF reserved for ISP service program. When N=2 means memory address $3F00 to $3FFF reserved for ISP service program…etc. Value N can be set and programmed into OB38R08T1 by the writer.

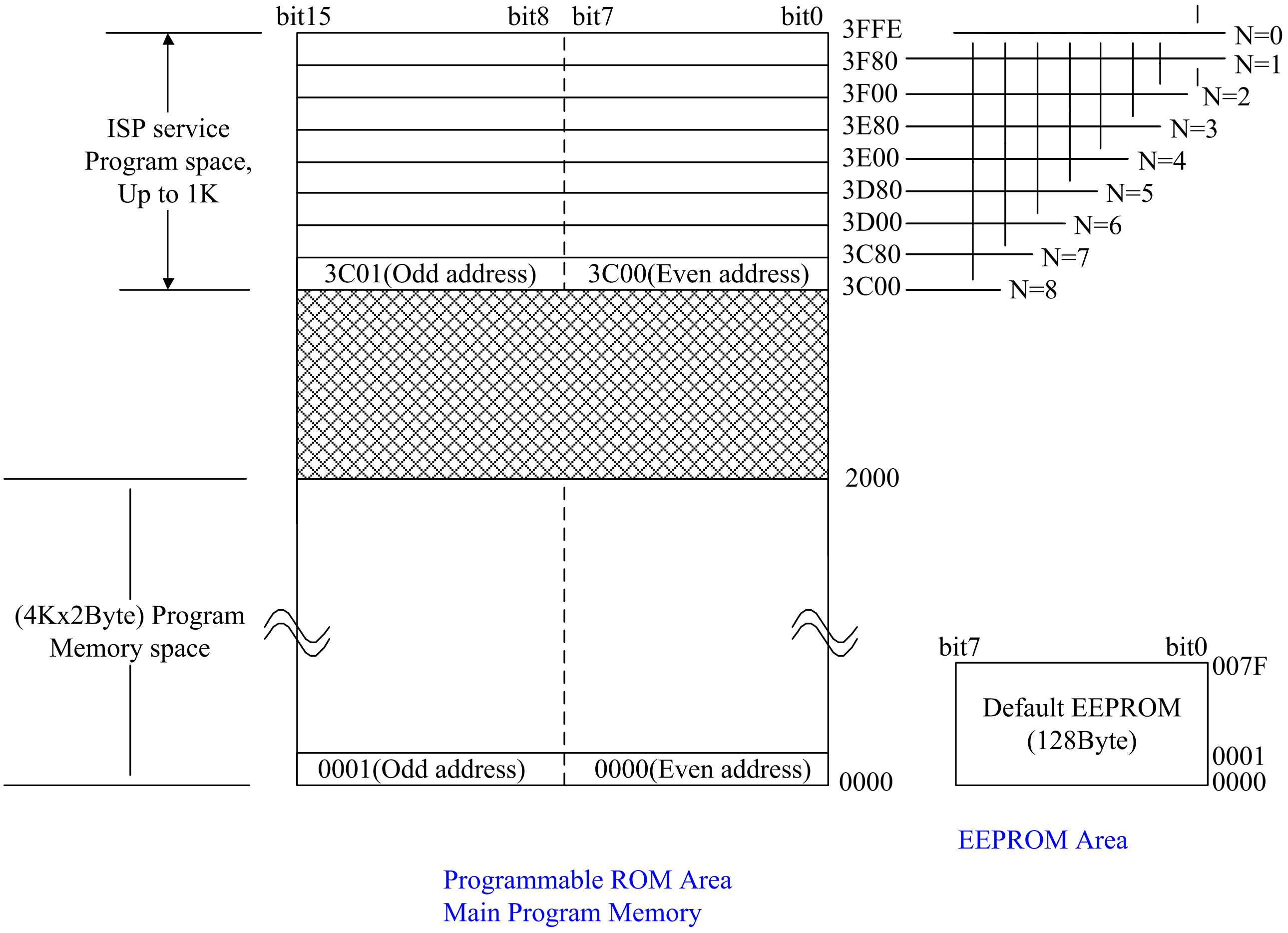


Fig. 3-1: OB38R08T1 ROM

3.2 Data Memory

The OB38R08T1 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.).

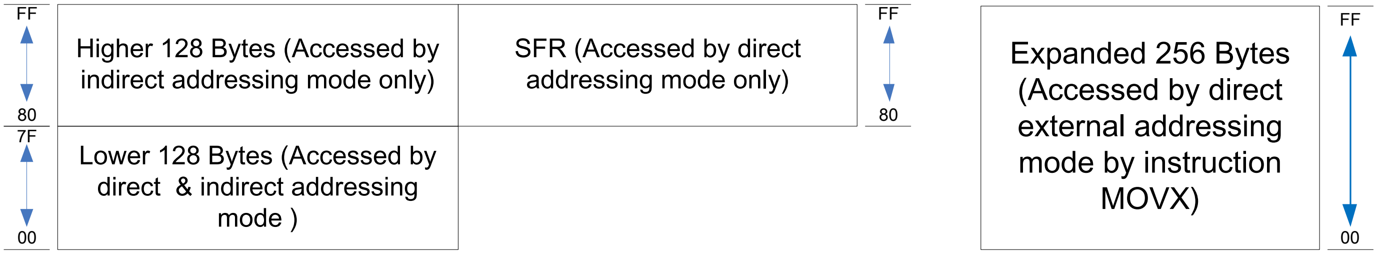

FF
80
7F
00
Higher 128 Bytes (Accessed by indirect addressing mode only)
Lower 128 Bytes (Accessed by direct & indirect addressing mode)
SFR (Accessed by direct addressing mode only)
FF
80
Expanded 256 Bytes (Accessed by direct external addressing mode by instruction MOVX)
FF
00

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 OB38R08T1 engine is composed of four components:

(1) Control unit

(2) Arithmetic – logic unit

(3) Memory control unit

(4) RAM and SFR control unit

The OB38R08T1 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	82h	DPL[7:0]								00H
DPH	Data pointer high	83h	DPH[7:0]								00H
AUX	Auxiliary register	91h	BRGS	-	SICS[1:0]		-	SEG1S	SEG0S	-	00H
CKCON	Clock control register	8Eh	-	ITS[2:0]			-	-	-	-	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.

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.

IT1=0: INT1 select level trigger.(high or low dependent on ENHIT1)

IT1=1: INT1 select edge trigger.(falling or rising or both edge dependent on ENHIT1).

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.

IT0=0: INT0 select level trigger.(high or low dependent on ENHIT0)

IT0=1: INT0 select edge trigger.(falling or rising or both edge dependent on ENHIT0)

6.3 ENHance Interrupt Type Register (ENHIT)

Mnemonic: ENHIT **Address: E5h**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	ENHIT1[1:0]		ENHIT0[1:0]		00h

	ENHIT0[1:0]=00	ENHIT0[1:0]=01	ENHIT0[1:0]=10	ENHIT0[1:0]=11
IT0=0	INT0 Low level trigger	INT0 High level trigger	--	--
IT0=1	INT0 Falling edge	INT0 Rising trigger	INT0 Both falling and rising	--

	ENHIT1[1:0]=00	ENHIT1[1:0]=01	ENHIT1[1:0]=10	ENHIT1[1:0]=11
IT1=0	INT1 Low level trigger	INT1 High level trigger	--	--
IT1=1	INT1 Falling edge	INT1 Rising trigger	INT1 Both falling and rising	--

6.4 External Interrupt Deglitch Register (INTDEG)

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

6.5 Peripheral Frequency control register

Mnemonic: PFCON **Address: D9h**

7	6	5	4	3	2	1	0	Reset
-	-	-	-	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

6.6 Mode 0 (13-bit Counter/Timer)

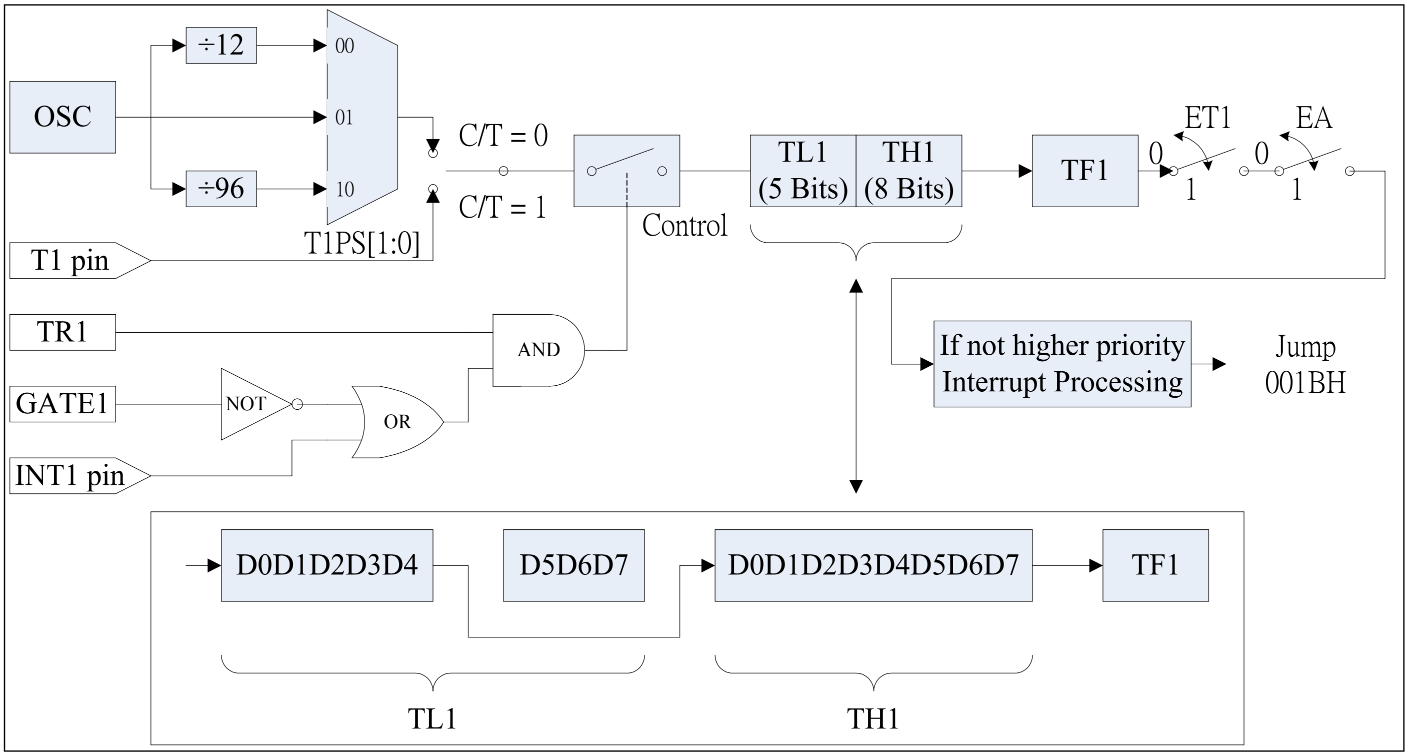


Fig. 6-1: Mode 0 -13 bit Timer / counter operation

6.7 Mode 1 (16-bit Counter/Timer)

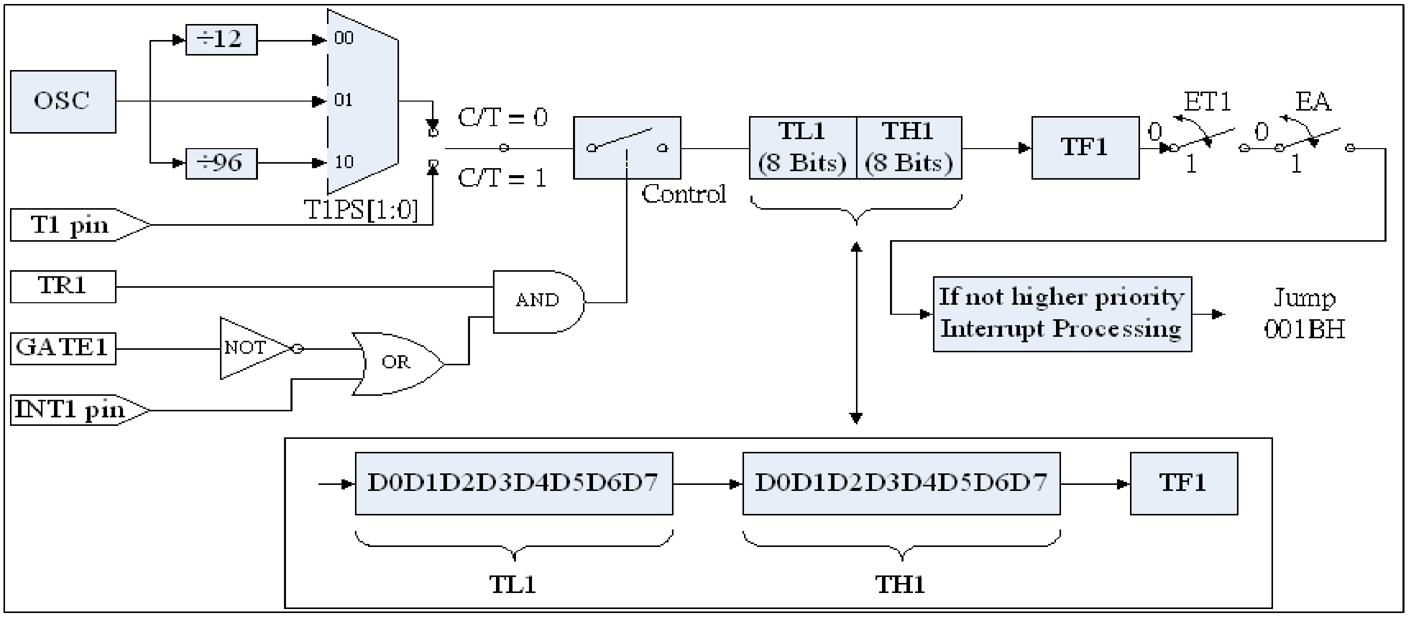


Fig. 6-2: Mode 1 -16 bit Timer / counter operation

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

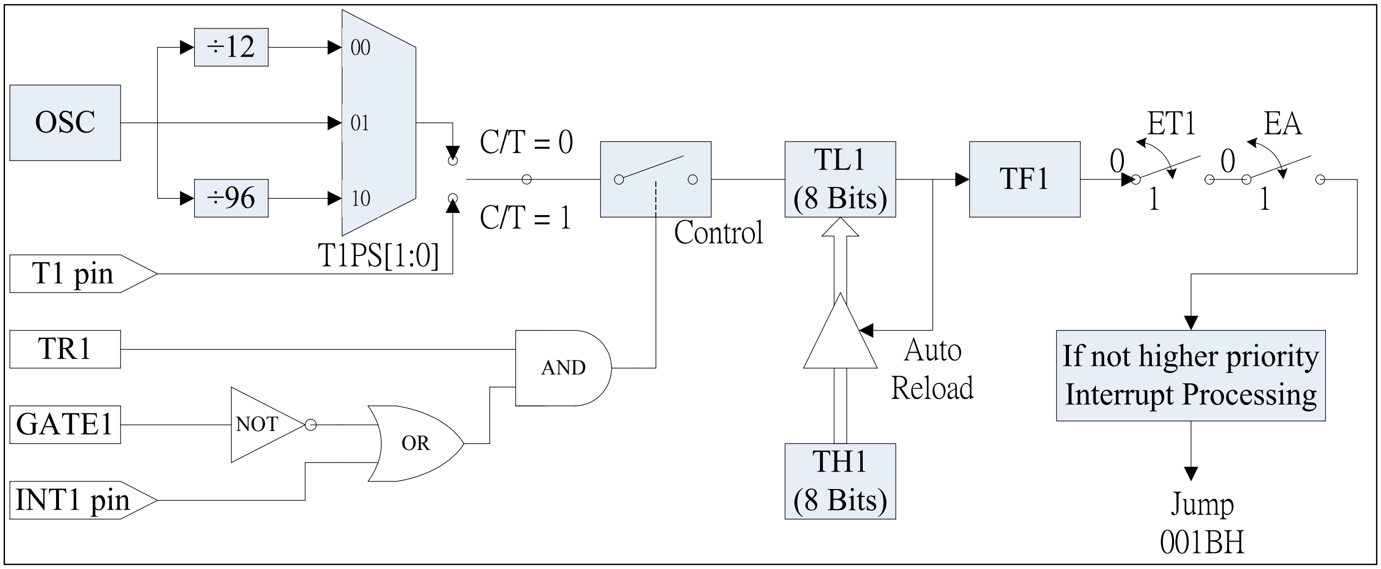


Fig. 6-3: Mode 2 8 bit Auto-reload Counter/Timer

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

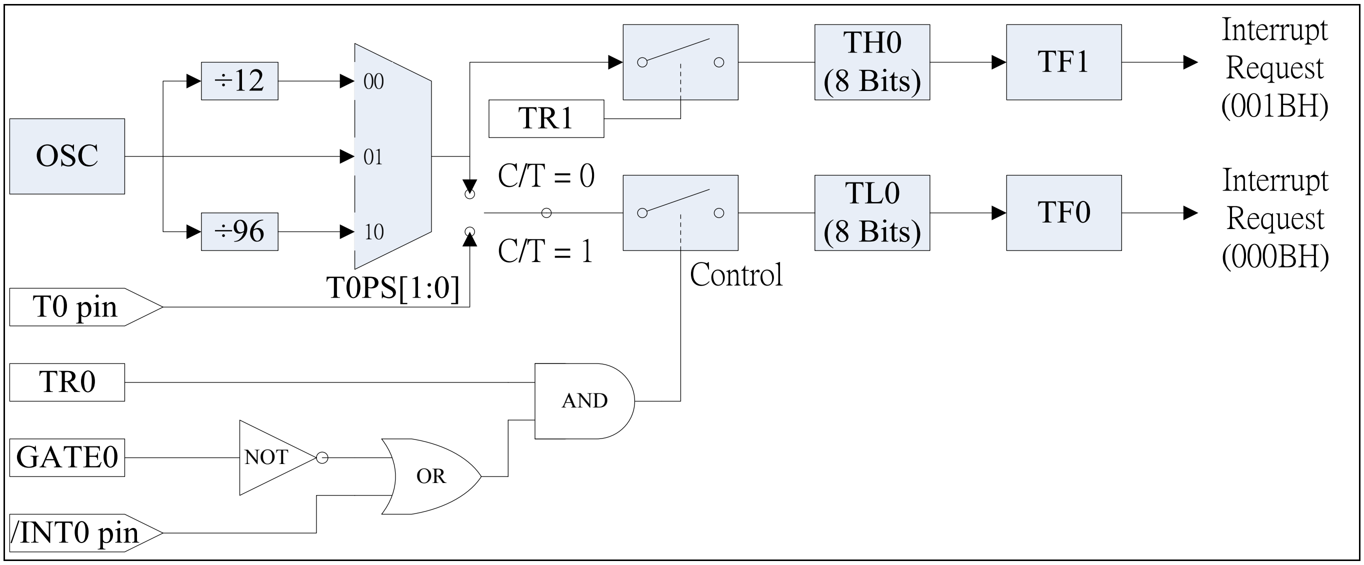
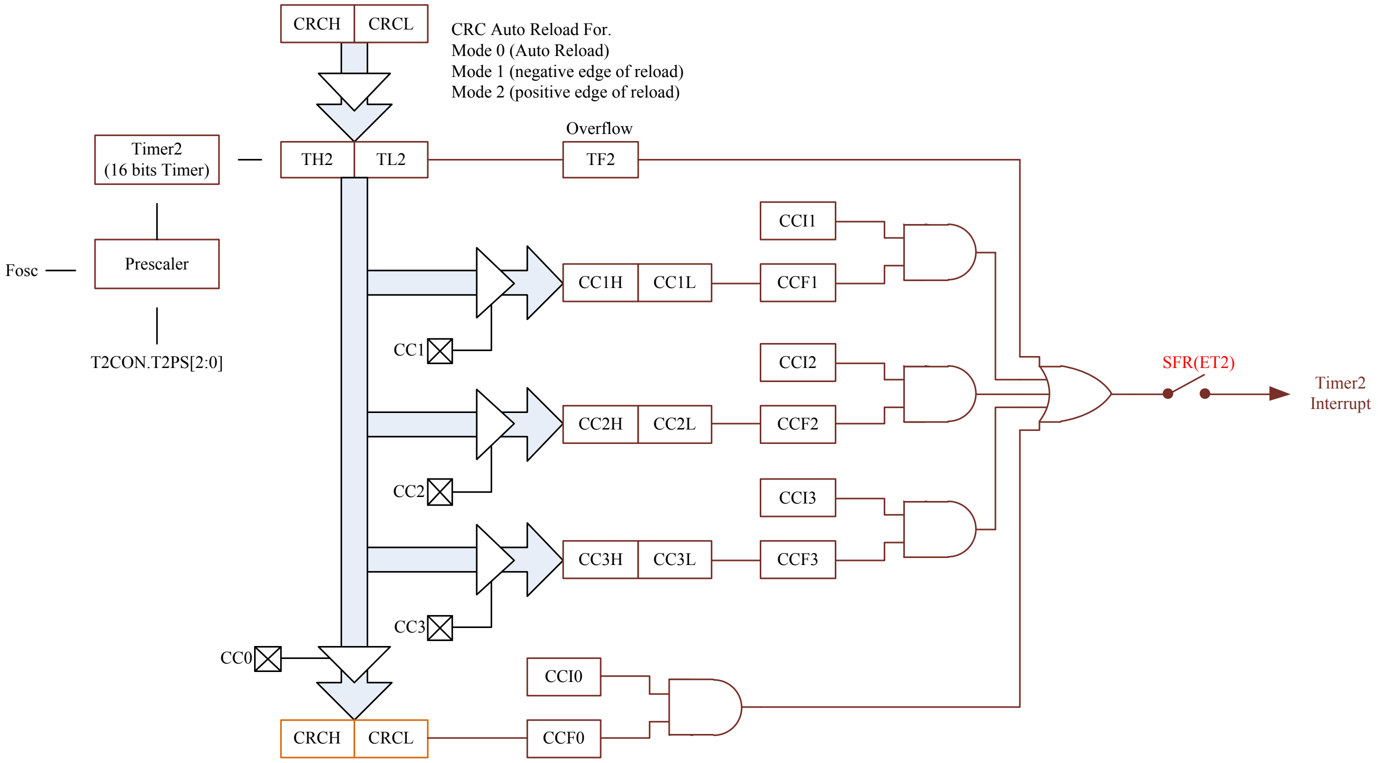


Fig. 6-4: Mode 3 - two independent 8 bit Timers / Counters (Only Timer 0)

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] = 00 – Reload disabled

T2R[1:0] = 01 –Mode 2: T2EX Rising Edge Reload

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.

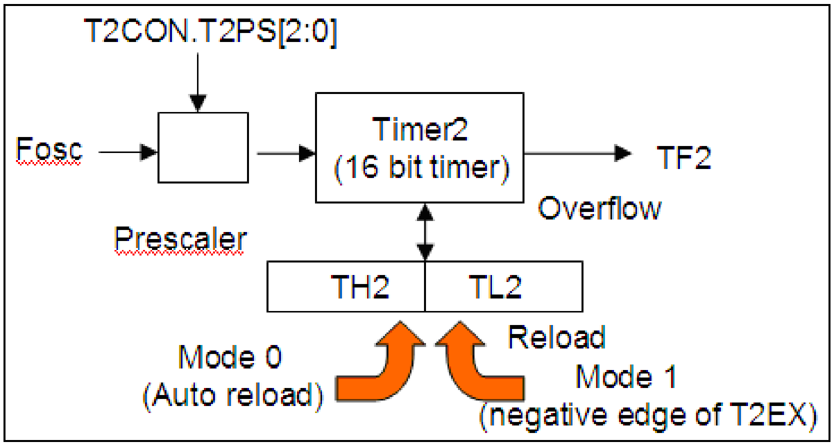


Fig. 7-1: Timer mode and Reload mode function

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.

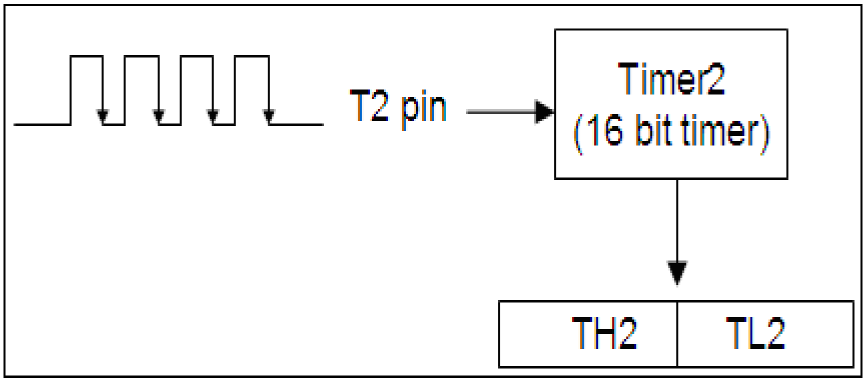


Fig. 7-2: Event counter mode function

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.

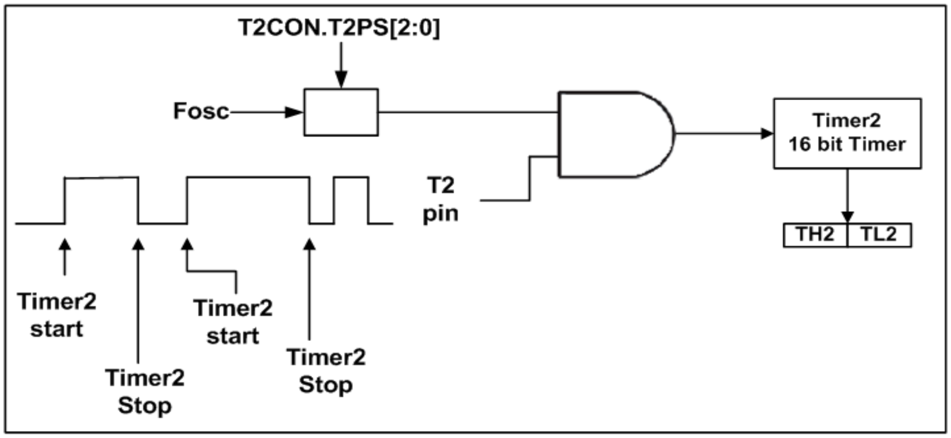


Fig. 7-3: Gated timer mode function

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.

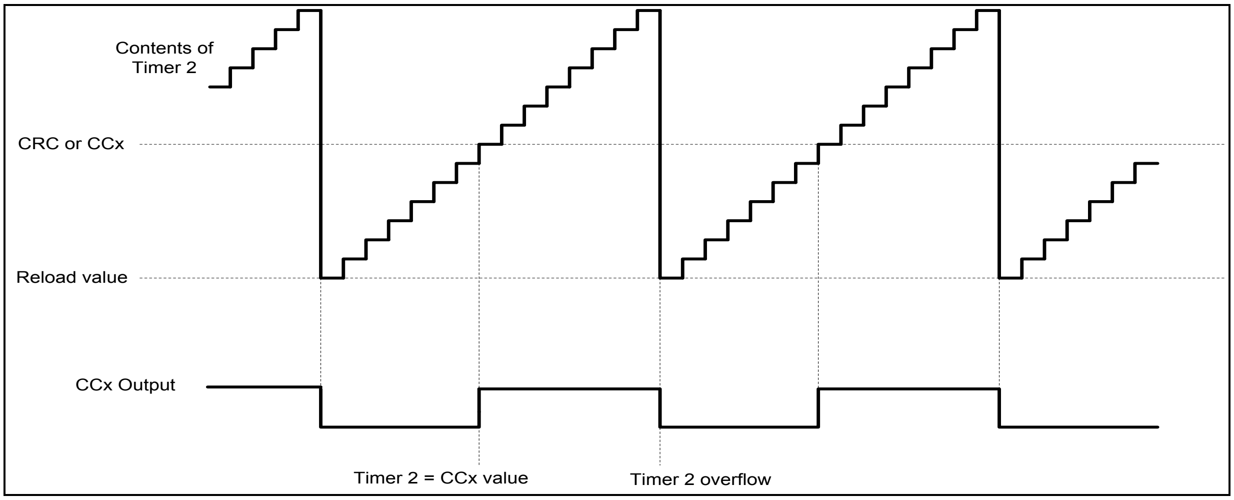


Fig. 7-4: Compare mode 0 function

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.

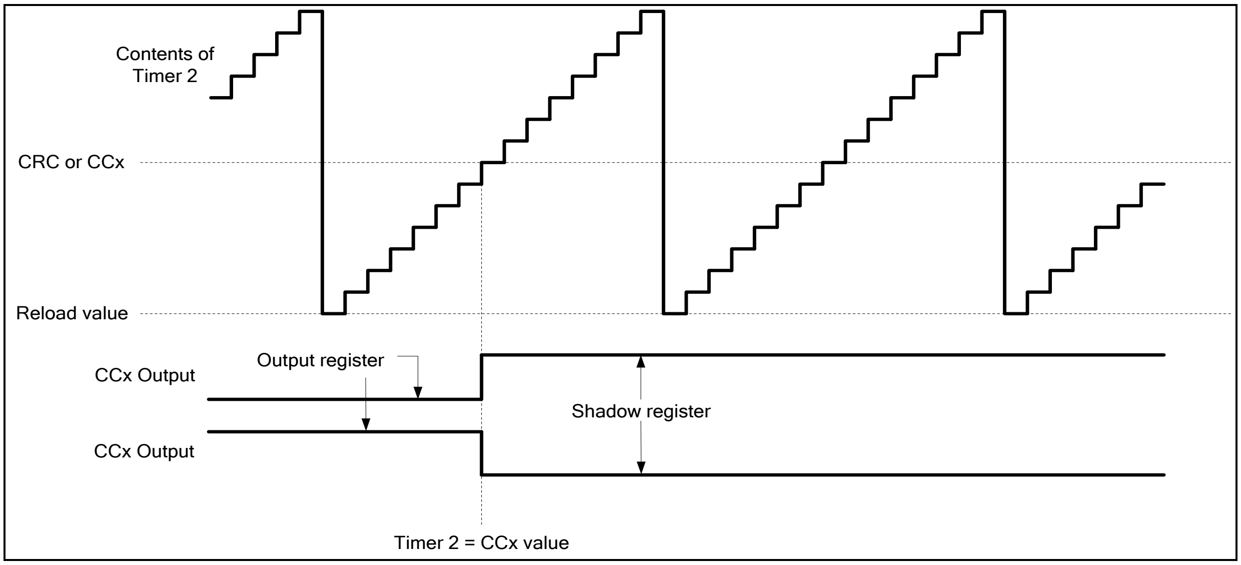


Fig. 7-5: Comparison mode 1 function

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.

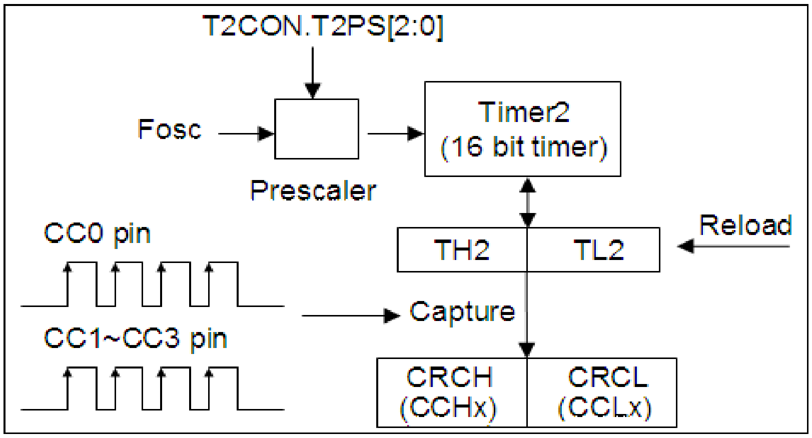


Fig. 7-6: Capture mode 0

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.

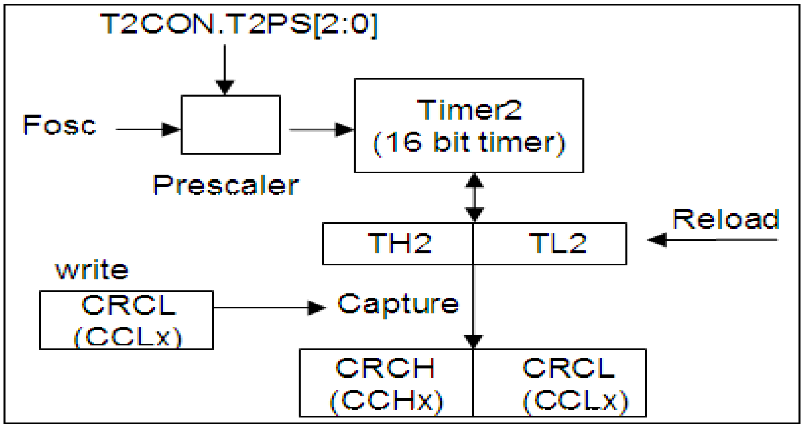


Fig. 7-7: Capture mode 1 function

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	-	SICS[1:0]		-	SEG1S	SEG0S	-	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	-	-	-	-	T1PS[1:0]		T0PS[1:0]		00H

Mnemonic: AUX **Address: 91h**

7	6	5	4	3	2	1	0	Reset
BRGS	-	SICS[1:0]		-	SEG1S	SEG0S	-	00H

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

SICS[1:0]: Serial interface channel selection control.

SICS[1:0]	Channel	Note
00	Port 1	RXD_0,TXD_0
01	Port 2	RXD_1,TXD_1
10	reserved	-
11	reserved	-

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.

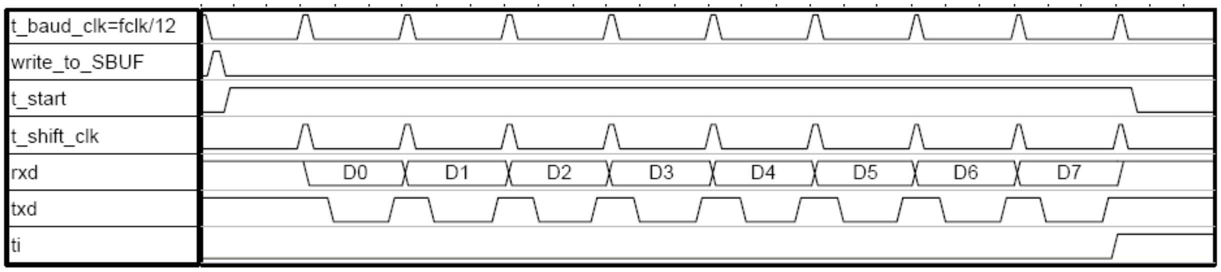


Fig. 8-1: Transmit mode 0

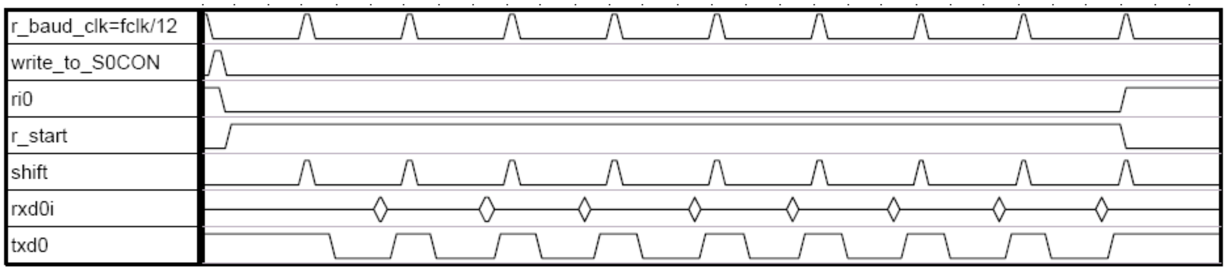


Fig. 8-2: Receive mode 0

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.

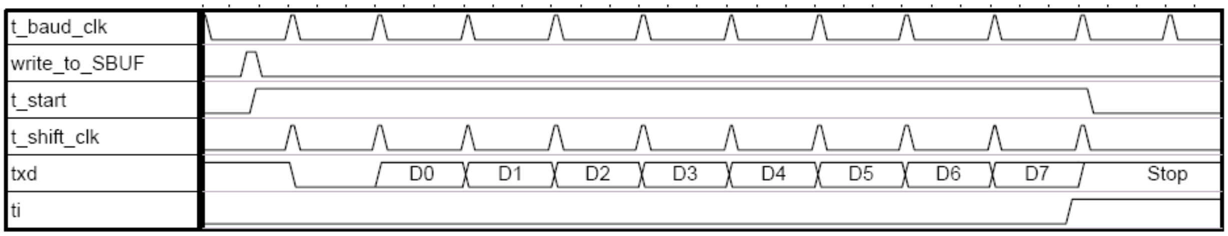


Fig. 8-3: Transmit mode 1

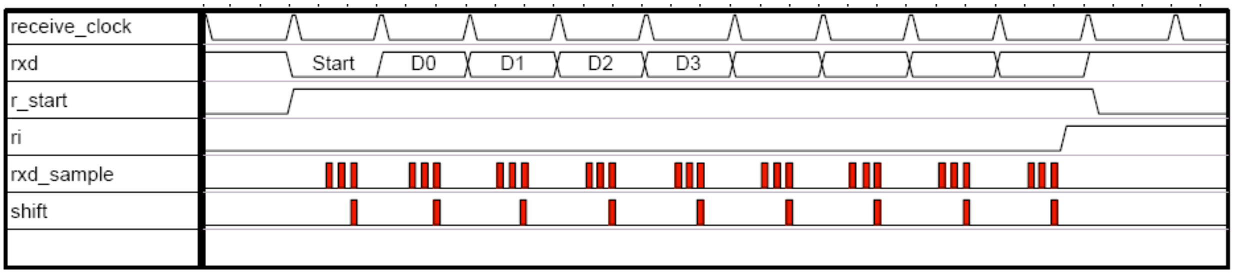


Fig. 8-4: Receive mode 0

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.

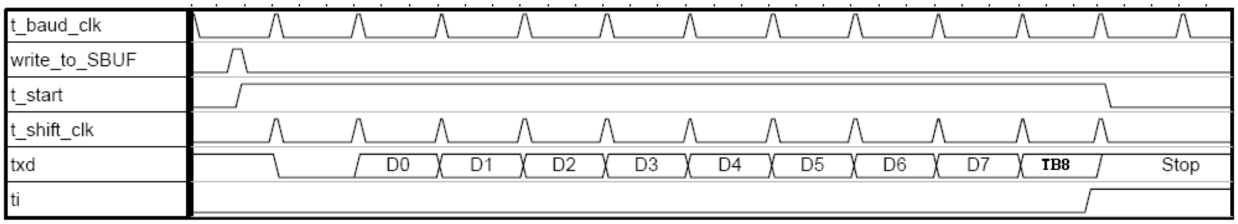


Fig. 8-5: Transfer Mode 2 and Mode 3

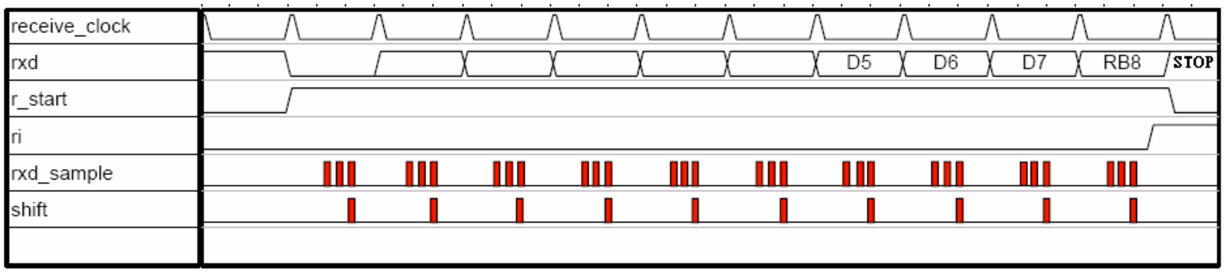


Fig. 8-6: The receiving modes 2 and 3

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
-	-	-	-	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

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

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

(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

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

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

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

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 WDTRF bit of RSTS 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 (23 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 WDTRE bit of WDTRC register. The default WDT time-out period is approximately 204.8ms (WDTRM [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 (WDTRM [3:0]) of Watch Dog Timer Control Register (WDTRC) should be set accordingly.

$$WDTRCLK = \frac{\mathrm{23KHz}}{2^{\mathrm{WDTRM}}} \qquad Watchdog\ reset\ time = \frac{256}{\mathrm{WDTRCLK}}$$

$$WDTICLK = \frac{\mathrm{23KHz}}{2^{\mathrm{WDTIM}}} \qquad Watchdog\ Interrupt\ time = \frac{256}{\mathrm{WDTICLK}}$$

Table 9-1: WDT time-out period

WDTRM [3:0]	Divider (23 KHz RC oscillator in)	Reset Time period @ 23KHz	WDTIM [3:0]	Divider (23 KHz RC oscillator in)	Interrupt Time period @ 23KHz
0000	1	11.1ms	0000	1	11.1ms
0001	2	22.2ms	0001	2	22.2ms
0010	4	44.5ms	0010	4	44.5ms
0011	8	89.0ms	0011	8	89.0ms
0100	16	178.0ms (default)	0100	16	178.0ms
0101	32	356.1ms	0101	32	356.1ms
0110	64	712.3ms	0110	64	712.3ms
0111	128	1.4246s	0111	128	1.4246s
1000	256	2.8493s	1000	256	2.8493s
1001	512	5.6987s	1001	512	5.6987s
1010	1024	11.397s	1010	1024	11.397s
1011	2048	22.795s	1011	2048	22.795s
1100	4096	45.590s	1100	4096	45.590s
1101	8192	91.180s	1101	8192	91.180s
1110	16384	182.36s	1110	16384	182.36s
1111	32768	364.72s	1111	32768	364.72s

Note: RC oscillator (23 KHz), about ± 20% of variation

The program can enable the WDT function by programming 1 to the WDTRE bit. After WDTRE set to 1, the 8 bit-counter starts to count with the selected time base source clock which set by WDTRM [3:0]. It will generate a reset signal when overflows. The WDTRE bit will be cleared to 0 automatically when MCU been reset, either hardware reset or WDT reset. As shown in Fig. 9-1.

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 WDTRF flag will set to one and automatically reset MCU. The WDTRF flag can be clear by software or external reset or power on reset.

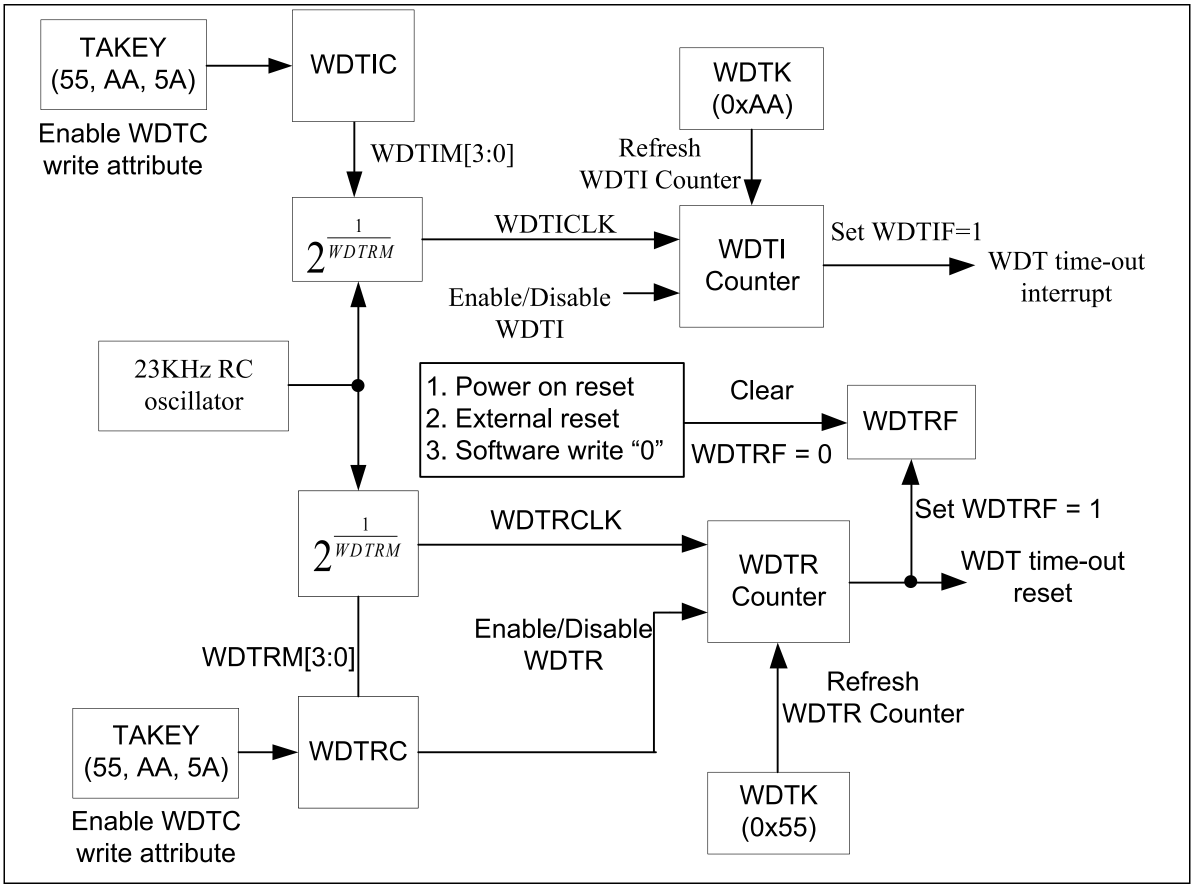


Fig. 9-1: Watchdog timer block diagram

Mnemonic	**Description**	**Dir.**	**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
WDTRC	Watchdog timer reset control register	B6h	-	-	WDTRE	-	WDTRM[3:0]				04H
WDTIC	Watchdog timer interrupt control register	FFh	-	-	WDTIE	WDTIOF	WDTIM[3:0]				00H
WDTK	Watchdog timer refresh key	B7h	WDTK[7:0]								00H
RSTS	Reset status register	A1h	-	LVRPINTF	-	-	WDTRF	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 (WDTRC & WDTIC)is read-only by default; software must write three specific values 55h, AAh and 5Ah sequentially to the TAKEY register to enable the WDTRC write attribute. That is:

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

Mnemonic: WDTRC **Address: B6h**

7	6	5	4	3	2	1	0	Reset
-	-	WDTRE	-	WDTRM [3:0]				04H

WDTRE: Control bit used to enable Watchdog reset timer.

WDTRE = 0 - Disable Watchdog reset timer.

WDTRE = 1 - Enable Watchdog reset timer.

WDTRM [3:0]: WDT clock source divider bit. As seen in Fig. 9-1 to reference the WDT time-out period.

Mnemonic: WDTIC **Address: FFh**

7	6	5	4	3	2	1	0	Reset
-	-	WDTIE	WDTIOF	WDTIM[3:0]				00H

WDTIE: Control bit used to enable Watchdog interrupt timer.

WDTIE = 0 - Disable Watchdog interrupt timer.

WDTIE = 1 - Enable Watchdog interrupt timer.

The function is support interrupt and stop mode wakeup.

WDTIOF: Watchdog Interrupt Timer overflow flag set by hardware when Watchdog Interrupt timer overflows. This flag can be cleared by software

WDTIM [3:0]: WDT clock source divider bit. As seen in Fig. 9-1 to reference the WDT time-out period.

Mnemonic: RSTS **Address: A1h**

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

WDTRF: Watchdog timer reset flag. When MCU is reset by watchdog, WDTRF 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 or 0xAA into WDTK register, and then the watchdog reset timer or interrupt timer will be cleared to zero.

For example 1, if enable WDT reset period is 2.8493s

```
MOV TAKEY, #55h

MOV TAKEY, #0AAh

MOV TAKEY, #5Ah    ; enable WDTRC write attribute.

MOV WDTRC, #28h    ; Set WDTRM [3:0] = 1000b. Set WDTRE =1 to enable WDT function.

.

.

.

MOV WDTK, #55h     ; Clear WDT reset timer to 0.
```

For example 2, if enable WDT interrupt period is 178.0ms.

```
MOV TAKEY, #55h

MOV TAKEY, #0AAh

MOV TAKEY, #5Ah     ; enable WDTIC write attribute.

MOV WDTIC, #24h      ;Set WDTIM [3:0] = 0100b. ;Set WDTIE =1 to enable WDT function
```

10. Interrupt

The OB38R08T1 provides 12 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	ADCIF – A/D converter interrupt	0053h	10
9	LVIIF – Low Voltage Interrupt	0063h	12
10	IICIF – IIC interrupt	006Bh	13
11	WDTIF – Watchdog interrupt	008Bh	17
12	TKIF – Touch interrupt	009Bh	19

* 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	EXEN2	-	IEIIC	IELVI	-	IEADC	-	IEPWM	00H
IEN2	Interrupt Enable 2 register	9AH	-	-	-	-	IETK	-	EWDT	-	00H
IRCON	Interrupt request register	C0H	EXF2	TF2	IICIF	LVIIF	-	ADCIF	-	PWMIF	00H
IRCON2	Interrupt request register 2	97H	-	-	-	-	TKIF	-	WDTIF	-	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

Mnemonic: IEN0 **Address: A8h**

7	6	5	4	3	2	1	0	Reset
EA	-	ET2	ES	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	-	IEADC	-	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.

IEADC: A/D converter interrupt enable

IEADC = 0 – Disable ADC interrupt.

IEADC = 1 – Enable ADC 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
-	-	-	-	IETK	-	EWDT	-	00H

IETK: Enable touch key interrupt.

IETK = 0 – Disable Touch Key interrupt.

IETK = 1 – Enable Touch Key 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	-	ADCIF	-	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.

ADCIF: A/D converter end interrupt flag. Hardware will clear this flag automatically when enter interrupt vector.

PWMIF: PWM interrut flag. Hardware will clear this flag automatically when enter interrupt vector.

Mnemonic: IRCON2 **Address: 97h**

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

TKIF: Touch Key interrupt flag. Must be cleared by software.
WDTIF: Watch dog interrupt flag

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	-
External interrupt 1	-	ADC interrupt
Timer 1 interrupt	-	-
Serial channel interrupt	-	LVI interrupt
Touch 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	-
IP1.2, IP0.2	External interrupt 1	-	ADC interrupt
IP1.3, IP0.3	Timer 1 interrupt	-	-
IP1.4, IP0.4	Serial channel interrupt	-	LVI interrupt
IP1.5, IP0.5	Touch interrupt	Timer 2 interrupt	IIC interrupt

Table 10-5: Polling sequence

Interrupt source	Sequence
External interrupt 0	↓
PWM interrupt	
Timer 0 interrupt	
Watchdog interrupt	
External interrupt 1	
ADC interrupt	
Timer 1 interrupt	
Serial channel interrupt	
LVI interrupt	
Touch Key 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. The following wake-up sources can be configured to wake the device from stop mode:

- External interrupt 0, 1
- LVI / LVR
- Watchdog timer reset / interrupt
- Touch key

12. 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[1:0]		03H
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[1:0]		-	-	-	-	-	-	00H

Mnemonic: IICCTL **Address: F9h**

7	6	5	4	3	2	1	0	Reset
IICEN	MSS	MAS	AB_EN	BF_EN	-	IICBR[1:0]		03H

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 FU_EN[1:0] is set to 10. Set this bit when multi-master and slave connection. Clear this bit when single master to single slave.

IICBR[1:0]: Baud rate selection (master mode only), where Fosc is the oscillator frequency.

IICBR[1:0]	Baud rate
00	Fosc/32
01	Fosc/64
10	Fosc/128
11	Fosc/256 (Default)

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)

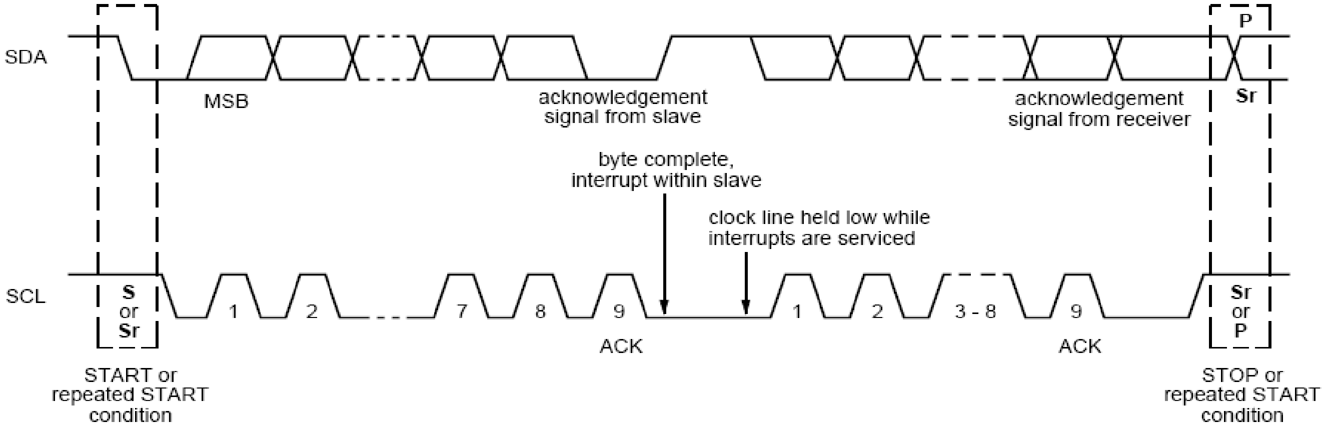

SDA
SCL
P
MSB
acknowledgement signal from slave
acknowledgement signal from receiver
Sr
byte complete, interrupt within slave
clock line held low while interrupts are serviced
S or Sr
1
2
7
8
9
ACK
1
2
3 - 8
9
ACK
Sr or P
START or repeated START condition
STOP or repeated START condition

Fig. 12-1: 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 FU_EN[1:0] = 10. It appears at the 8th bit after the IIC address as shown in Fig. 11-2. 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

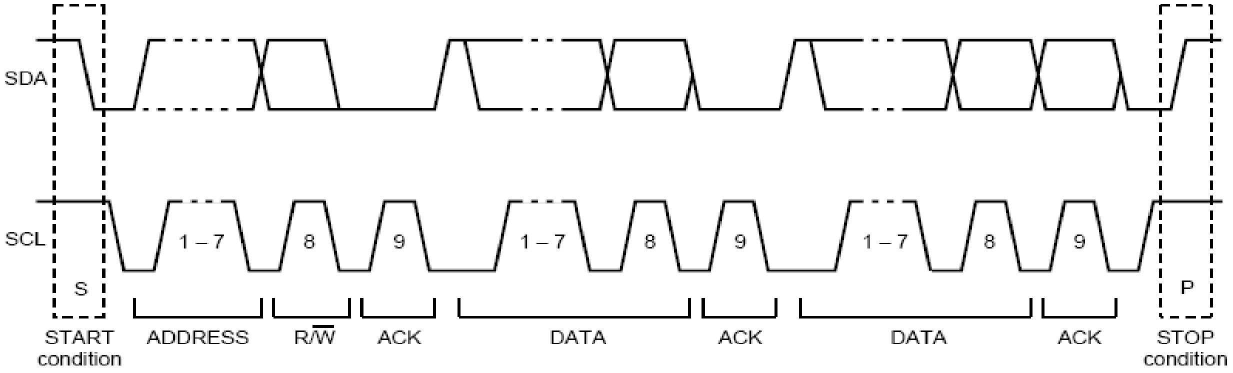


Fig. 12-2: RW bit in the 8th bit after IIC address

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 FU_EN[1:0] = 10. 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[1:0]		-	-	-	-	-	-	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[1:0] will be auto-clear by hardware, so setting FU_EN[1:0] repeatedly is necessary.

※ FU_EN[1:0] should be set as 01 before read/write data transfer for bus release; otherwise, SCL will be locked (pull low).

※ FU_EN[1:0] will be auto-clear by hardware, so setting FU_EN[1:0] repeatedly is necessary.

Slave mode:

01: FU_EN[1:0] should be set as 01 only. The other value is inhibited.

Notice:

1. FU_EN[1:0] should be set as 01 before read/write data transfer for bus release; otherwise, SCL will be locked(pull low).
2. FU_EN[1:0] 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[1:0] as 01.
4. FU_EN[1:0] will be auto-clear by hardware, so setting FU_EN[1:0] repeatedly is necessary.

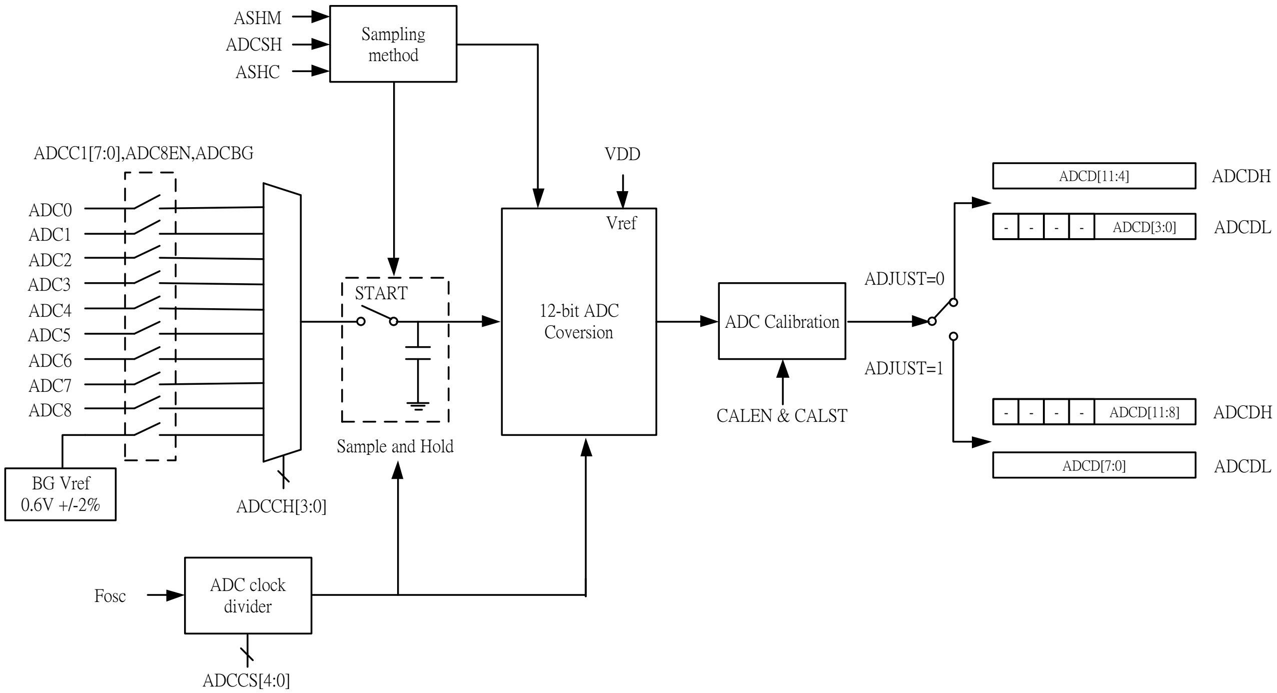
※ FU_EN[1:0] should be set as 01 after read/write data transfer for receiving a stop condition from bus master.

※ In Transmit data mode(slave mode), the output data should be filled into IICRWD before setting FU_EN[1:0] as 01.

※ FU_EN[1:0] will be auto-clear by hardware, so setting FU_EN[1:0] repeatedly is necessary.

13. ADC

The OB38R08T1 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
ADC											
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]				00H
ADCDH	ADC Data High	ADH	ADCDH [7:0]								00H
ADCDL	ADC Data Low	AEH	ADCDL [7:0]								00H
ADCCS	ADC Clock Select	AFH	ADCBGE	ADC8EN	-	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]				00H

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	8
Others	Bandgap reference voltage

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	ADC8EN	-	ADCCS[4:0]					00H

ADCBGE: ADC bandgap reference voltage channel enable.

(bandgap reference voltage 0.6V +/- 2%)

0: Disable

1: Enable

ADC8EN: ADC channel 8 enable.

0: Disable ADC channel 8

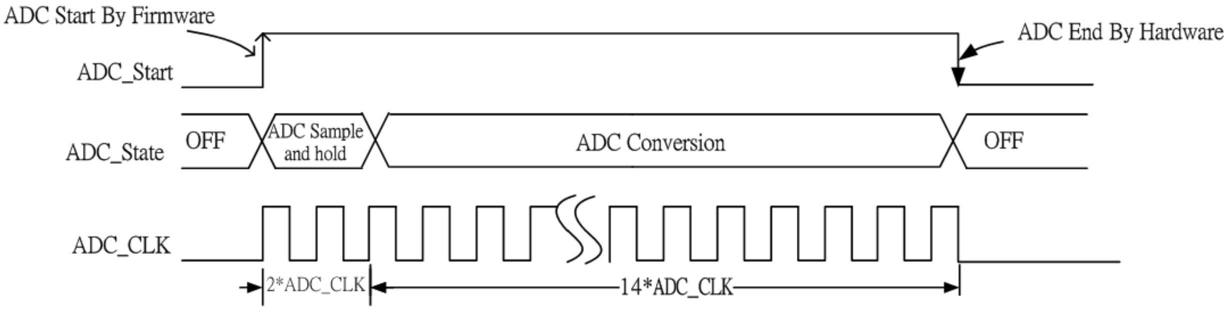
1: Enable ADC channel 8

ADCCS[4:0]: ADC clock select.

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

If ASHM=0 , ADCSH=0x00:

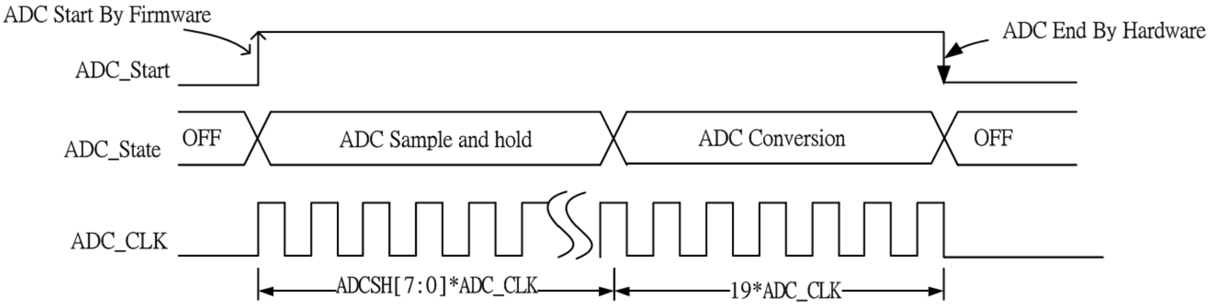
$$ADC\_Conversion\_Rate = \frac{\mathrm{ADC\_Clock}}{16}$$



If ASHM=0 , ADCSH≠0x00:

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

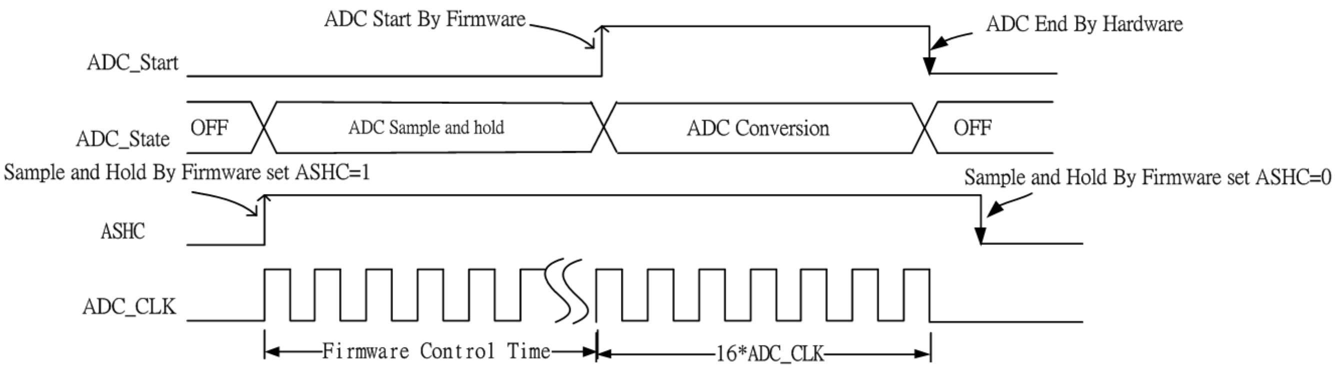
※Sample and hold time by hardware control decisions.



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
```

14. 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	-	LVRLPINTF	-	-	WDRTF	SWRF	LVRF	PORF	00H
LVC	Low voltage control	E6h	LVI_EN	-	LVRE	-	-	-	LVIS[1:0]		20H

Mnemonic: RSTS **Address: A1h**

7	6	5	4	3	2	1	0	Reset
-	LVRLPINTF	-	-	WDRTF	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.

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.

Mnemonic: LVC **Address: E6h**

7	6	5	4	3	2	1	0	Reset
LVI_EN	-	LVRE	-	-	-	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.

LVIS LVI level select:

00: 2.4V.

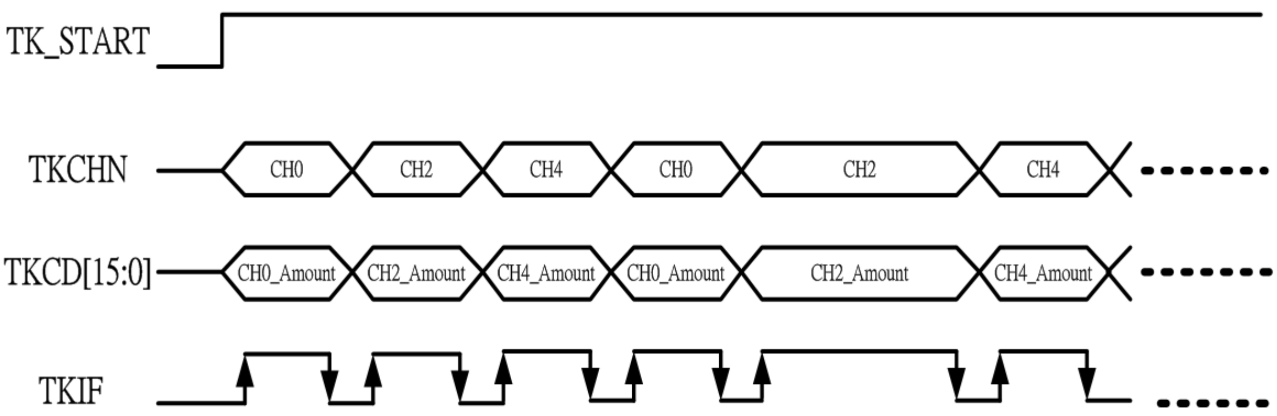
01: 2.6V

10: 3.2V

11: 4.0V

15. Touch Sense Unit

Enable Touch Key Channel 0、2、4



TKIF is set to high level by hardware.

TKIF is cleared to low level by firmware.

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
Capacitor Sensor											
IEN2	Interrupt Enable 2 register	9AH	-	-	-	-	IETK	-	EWDT	-	00H
IRCON2	Interrupt request register 2	97H	-	-	-	-	TKIF	-	WDTIF	-	00H
TKEN0	Touch Key Enable 0	93H	TK7EN	TK6EN	TK5EN	TK4EN	TK3EN	TK2EN	TK1EN	TK0EN	00H
TKEN1	Touch Key Enable 1	94H	TK15EN	TK14EN	TK13EN	TK12EN	TK11EN	TK10EN	TK9EN	TK8EN	00H
TKCON	Touch Key Control Reg.	9BH	TK_START	TK_PS	-	-	TK_SPEED[3:0]				00H
TKSW	Touch Key Switch Reg.	FEH	TKSIF	LEDSIF	-	NSP	VTK[1:0]		DTS[1:0]		00H
TKCHN	Touch Key Channel Number Reg.	9DH	-	-	-	-	TKCHN[3:0]				00H
TKCDL	Touch Key Capture Data Low-byte Reg.	9EH	TKCD[7:0]								00H
TKCDH	Touch Key Capture Data Hi-byte Reg.	9FH	TKCD[15:8]								00H
TKSTATUS0	Touch Key Status 0	BBH	TKSTATUS0[7:0]								00H
TKSTATUS1	Touch Key Status 1	85H	TKSTATUS1[7:0]								00H
TKPSSR	Touch Key Samping Rate	F2H	TKPSSR[7:0]								07H
TKWKTRICNT	Touch Key Trigger Counter	F3H	TKWKTRICNT[7:0]								02H
TKRUNTIME	Touch Key Running Time	9CH	TKRUNTIME[7:0]								00H

Description	SRAM Addr.	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	Reset
Touch Key Threshold Value Registers										
TK0TRIGH	0x1D0	TK0TRIG[15:8]								-
TK0TRIGL	0x1D1	TK0TRIG[7:0]								-
TK1TRIGH	0x1D2	TK1TRIG[15:8]								-
TK1TRIGL	0x1D3	TK1TRIG[7:0]								-
TK2TRIGH	0x1D4	TK2TRIG[15:8]								-
TK2TRIGL	0x1D5	TK2TRIG[7:0]								-
TK3TRIGH	0x1D6	TK3TRIG[15:8]								-
TK3TRIGL	0x1D7	TK3TRIG[7:0]								-
TK4TRIGH	0x1D8	TK4TRIG[15:8]								-
TK4TRIGL	0x1D9	TK4TRIG[7:0]								-
TK5TRIGH	0x1DA	TK5TRIG[15:8]								-
TK5TRIGL	0x1DB	TK5TRIG[7:0]								-
TK6TRIGH	0x1DC	TK6TRIG[15:8]								-
TK6TRIGL	0x1DD	TK6TRIG[7:0]								-
TK7TRIGH	0x1DE	TK7TRIG[15:8]								-
TK7TRIGL	0x1DF	TK7TRIG[7:0]								-
TK8TRIGH	0x1E0	TK8TRIG[15:8]								-
TK8TRIGL	0x1E1	TK8TRIG[7:0]								-
TK9TRIGH	0x1E2	TK9TRIG[15:8]								-
TK9TRIGL	0x1E3	TK9TRIG[7:0]								-
TK10TRIGH	0x1E4	TK10TRIG[15:8]								-
TK10TRIGL	0x1E5	TK10TRIG[7:0]								-
TK11TRIGH	0x1E6	TK11TRIG[15:8]								-
TK11TRIGL	0x1E7	TK11TRIG[7:0]								-
TK12TRIGH	0x1E8	TK12TRIG[15:8]								-
TK12TRIGL	0x1E9	TK12TRIG[7:0]								-
TK13TRIGH	0x1EA	TK13TRIG[15:8]								-
TK13TRIGL	0x1EB	TK13TRIG[7:0]								-
TK14TRIGH	0x1EC	TK14TRIG[15:8]								-
TK14TRIGL	0x1ED	TK14TRIG[7:0]								-
TK15TRIGH	0x1EE	TK15TRIG[15:8]								-
TK15TRIGL	0x1EF	TK15TRIG[7:0]								-
Touch Key Counter Registers										
TK0TRICNT	0x1F0	TK0TRICNT[7:0]								-
TK1TRICNT	0x1F1	TK1TRICNT[7:0]								-
TK2TRICNT	0x1F2	TK2TRICNT[7:0]								-
TK3TRICNT	0x1F3	TK3TRICNT[7:0]								-
TK4TRICNT	0x1F4	TK4TRICNT[7:0]								-
TK5TRICNT	0x1F5	TK5TRICNT[7:0]								-
TK6TRICNT	0x1F6	TK6TRICNT[7:0]								-
TK7TRICNT	0x1F7	TK7TRICNT[7:0]								-
TK8TRICNT	0x1F8	TK8TRICNT[7:0]								-
TK9TRICNT	0x1F9	TK9TRICNT[7:0]								-
TK10TRICNT	0x1FA	TK10TRICNT[7:0]								-
TK11TRICNT	0x1FB	TK11TRICNT[7:0]								-
TK12TRICNT	0x1FC	TK12TRICNT[7:0]								-

TK13TRICNT	0x1FD	TK13TRICNT[7:0]	-
TK14TRICNT	0x1FE	TK14TRICNT[7:0]	-
TK15TRICNT	0x1FF	TK15TRICNT[7:0]	-

Mnemonic: IEN2 **Address: 9Ah**

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

IETK: IETK = 0 – Disable Touch Key interrupt.
IETK = 1 – Enable Touch Key interrupt.

Mnemonic:IRCON2 **Address: 97h**

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

TKIF: Touch Key interrupt flag. Must be cleared by software.

Mnemonic: TKEN0 **Address: 93h**

7	6	5	4	3	2	1	0	Reset
TK7EN	TK6EN	TK5EN	TK4EN	TK3EN	TK2EN	TK1EN	TK0EN	00H

TK7EN: Touch key channels 7 enable.
TK7EN = 1 – Enable touch key channel 7
TK6EN: Touch key channels 6 enable.
TK6EN = 1 – Enable touch key channel 6
TK5EN: Touch key channels 5 enable.
TK5EN = 1 – Enable touch key channel 5
TK4EN: Touch key channels 4 enable.
TK4EN = 1 – Enable touch key channel 4
TK3EN: Touch key channels 3 enable.
TK3EN = 1 – Enable touch key channel 3
TK2EN: Touch key channels 2 enable.
TK2EN = 1 – Enable touch key channel 2
TK1EN: Touch key channels 1 enable.
TK1EN = 1 – Enable touch key channel 1
TK0EN: Touch key channels 0 enable.
TK0EN = 1 – Enable touch key channel 0

Mnemonic: TKEN1 **Address: 94h**

7	6	5	4	3	2	1	0	Reset
TK15EN	TK14EN	TK13EN	TK12EN	TK11EN	TK10EN	TK9EN	TK8EN	00H

TK15EN: Touch key channels 15 enable.
TK15EN = 1 – Enable touch key channel 15
TK14EN: Touch key channels 14 enable.
TK14EN = 1 – Enable touch key channel 14
TK13EN: Touch key channels 13 enable.
TK13EN = 1 – Enable touch key channel 13

TK12EN: Touch key channels 12 enable.
TK12EN = 1 – Enable touch key channel 12
TK11EN: Touch key channels 11 enable.
TK11EN = 1 – Enable touch key channel 11
TK10EN: Touch key channels 10 enable.
TK10EN = 1 – Enable touch key channel 10
TK9EN: Touch key channels 9 enable.
TK9EN = 1 – Enable touch key channel 9
TK8EN: Touch key channels 8 enable.
TK8EN = 1 – Enable touch key channel 8

Mnemonic: TKCON **Address: 9BH**

7	6	5	4	3	2	1	0	Reset
TK_ START	TK_PS	-	-	TK_SPEED[3:0]				00H

TK_START: 0 = The touch key will disable.
1 = The touch key will enable.
TK_PS: This register is select input clock for the touch key counter.
0 — Input clock=32MHz
1 — Input clock=16MHz
TK_SPEED[3:0]: This register is select the charging speed of touch sensor.
TK_SPEED="0000" Charge Level 0 (fastest)
TK_SPEED="0001" Charge Level 1
:
TK_SPEED="1111" Charge Level 15 (slowest)

Mnemonic: TKSW **Address: FEH**

7	6	5	4	3	2	1	0	Reset
TKSIF	LEDSIF	-	NSP	VTK[1:0]		DTS[1:0]		00H

TKSIF: The flag is set by hardware when touch function starts operating.
Must be cleared by software.
LEDSIF: The flag is set by hardware when LED function starts operating and touch function is turned on.
Must be cleared by software.
NSP: Port status of TK that currently have not been scanned (starting TK)
0 — Floating.
1 — Determined by the GPIO port setting.
VTK[1:0]: Touch the module voltage selection.

VTK[1:0]	Power Select	VDD working range
00	MCU VDD	2.4V ~ 5.5V
01	TK LDO	2.4V~ 5.5V
10	TK LDO	3.2V~ 5.5V

DTS[1:0]: Discharge time selection.
00 — 30*62.5ns
01 — 60*62.5ns
10 — 90*62.5ns

11－105*62.5ns

Mnemonic: TKCHN **Address: 9Dh**

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

TKCHN[3:0]: This register indicates the counter scanning channels (Read only).

Mnemonic: TKCDL **Address: 9Eh**

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

TKCD[7:0]: This register for 16 bits counter low byte contents (Read only).

Mnemonic: TKCDH **Address: 9Fh**

7	6	5	4	3	2	1	0	Reset
TKCD[15:8]								00H

TKCD[15:8]: This register for 16 bits counter high byte contents (Read only).

Mnemonic: TKSTATUS0 **Address: BBh**

7	6	5	4	3	2	1	0	Reset
TKSTATUS0[7:0]								00H

TKSTATUS0[7:0]: Record touch key channel 0~7 status.

Mnemonic: TKSTATUS1 **Address: 85h**

7	6	5	4	3	2	1	0	Reset
TKSTATUS1[7:0]								00H

TKSTATUS1[7:0]: Record touch key channel 8~15 status.

Mnemonic: TKRUNTIME **Address:9Ch**

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

TKRUNTIME[7:0]: Touch key running time.

This setting is valid only when LED and Touch function is enable. This value determines the touch running time.

Touch Key Running Time:

$$Running\_Time = \frac{\mathrm{TKRUNTIME[7:0]}}{LED\_Clock}$$

※ If TKRUNTIME=” 0x00” , LED and Touch key share pad is disable.

Mnemonic: TKPSSR **Address: F2h**

7	6	5	4	3	2	1	0	Reset
TKPSSR [7:0]								07H

TKPSSR [7:0]: This setting is for the sampling rate of the touch key in power saving mode.
Sampling Rate=23KHz/[256*(TKPSSR [7:0]+1)]

Mnemonic: TKWKTRICNT **Address: F3h**

7	6	5	4	3	2	1	0	Reset
TKWKTRICNT [7:0]								02H

TKWKTRICNT [7:0]: This setting is for wake-up after the touch key reaches the trigger value in power saving mode.

Mnemonic: TKxTRIGH、TKxTRIGL **Indirect Address: 0x01D0~0x01EF**

7	6	5	4	3	2	1	0	Reset
TKxTRIG[15:8]								00H
TKxTRIG[7:0]								00H

TKxTRIG [15:0]: This setting the touch key trigger value.

Mnemonic: TKxTRICNT **Indirect Address: 0x01F0~0x01FF**

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

TKxTRICNT [7:0]: This records the number of triggers.

16. PWM - Pulse Width Modulation

The OB38R08T1 provides four channel PWM output.

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]			PWMOMS	PWM3EN	PWM2EN	PWM1EN	PWM0EN	00H
PWMD0H	PWM 0 Data register high byte	BDh	-	-	-	-	-	-	PWMD0[9:8]		00H
PWMD0L	PWM 0 Data register low byte	BCh	PWMD0[7:0]								00H
PWMD1H	PWM 1 Data register high byte	BFh	-	-	-	-	-	-	PWMD1[9:8]		00H
PWMD1L	PWM 1 Data register low byte	BEh	PWMD1[7:0]								00H
PWMD2H	PWM 2 Data register high byte	B2h	-	-	-	-	-	-	PWMD2[9:8]		00H
PWMD2L	PWM 2 Data register low byte	B1h	PWMD2[7:0]								00H
PWMD3H	PWM 3 Data register high byte	B4h	-	-	-	-	-	-	PWMD3[9:8]		00H
PWMD3L	PWM 3 Data register low byte	B3h	PWMD3[7:0]								00H
PWMMDH	PWM Max Data register high byte	CFh	-	-	-	-	-	-	PWMMD[9:8]		00H
PWMMDL	PWM Max Data register low byte	CEh	PWMMD[7:0]								FFH
PWMC2	PWM Control 2	DDH	UDIS	-	-	-	PWM3PS	PWM2PS	PWM1PS	PWM0PS	00H
PWMDT0	PWM 0 Dead Time	DEH	DTP0[1:0]		DT0[5:0]						00H
PWMDT1	PWM 1 Dead Time	DFH	DTP1[1:0]		DT1[5:0]						00H

Mnemonic: PWMC **Address: B5h**

7	6	5	4	3	2	1	0	Reset
PWMCS[2:0]			PWMOMS	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 (P1.6/T0)

PWMOMS: PWM output mode select

PWMOMS = 1 –Complementary mode.

· Channel 0~3 will be classified into 2 groups (channel 0~1 & 2~3)

· Duty cycle is determined by PWMD0, PWMD2. (PWMD1, PWMD3 is invalid)

· Dead time function Enable.

PWMOMS = 0 –Independent mode.

PWM3EN: PWM channel 3 enable control bit.

PWM3EN = 1 – PWM channel 3 enable.

PWM3EN = 0 – PWM channel 3 disable.

PWM2EN: PWM channel 2 enable control bit.

PWM2EN = 1 – PWM channel 2 enable.

PWM2EN = 0 – PWM channel 2 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: BDh**

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

Mnemonic: PWMD0L **Address: BCh**

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

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

Mnemonic: PWMD1H **Address: BFh**

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

Mnemonic: PWMD1L **Address: BEh**

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

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

Mnemonic: PWMD2H **Address: B2h**

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

Mnemonic: PWMD2L **Address: B1h**

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

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

Mnemonic: PWMD3H **Address: B4h**

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

Mnemonic: PWMD3L **Address: B3h**

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

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

Mnemonic: PWMMDH **Address: CFh**

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

Mnemonic: PWMMDL **Address: CEh**

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.

Mnemonic: PWMC2 **Address: DDh**

7	6	5	4	3	2	1	0	Reset
UDIS	-	-	--	PWM3PS	PWM2PS	PWM1PS	PWM0PS	00H

UDIS: PWM update disable bit
0: Update from duty cycle and period buffer registers are enabled
1: Update from duty cycle and period buffer registers are disabled
※ The UDIS bit affects all duty cycle buffer registers (PWMD0x, PWMD1x, PWMD2x, PWMD3x) and the PWM period buffer register (PWMMDx). No duty cycle changes or period value changes will have effect while UDIS=1.

PWM3PS: PWM channel 3 polarity select
0: Active low
1: Active high

PWM2PS: PWM channel 2 polarity select
0: Active low
1: Active high

PWM1PS: PWM channel 1 polarity select
0: Active low
1: Active high

PWM0PS: PWM channel 0 polarity select
0: Active low
1: Active high

A
nemonic: PWMDT0 ddress: DEH

7	6	5	4	3	2	1	0	Reset
DTP0[1:0]		DT0[5:0]						00H

DTP0[1:0]: Dead time prescaler for channel 0/1

DTP0 [1:0]	Mode
00	Fosc
01	Fosc/8
10	Fosc/16
11	Fosc/32

DT0[5:0]: PWM channel 0/1 dead time value
※ Dead time：

$$\text{Dead Time} = \frac{\text{DT0}[5:0]+1}{\text{DTP0 clock}}$$

※ Increasing the Dead time will shorten the active time, extend the inactive time, and cause a change in the duty cycle.

A
nemonic: PWMDT1 ddress: DFH

7	6	5	4	3	2	1	0	Reset
DTP1[1:0]		DT1[5:0]						00H

DTP1[1:0]: Dead time prescaler for channel 2/3

DTP1 [1:0]	Mode
00	Fosc
01	Fosc /8
10	Fosc /16
11	Fosc/32

DT1[5:0]: PWM channel 2/3 dead time value

※ Dead time：

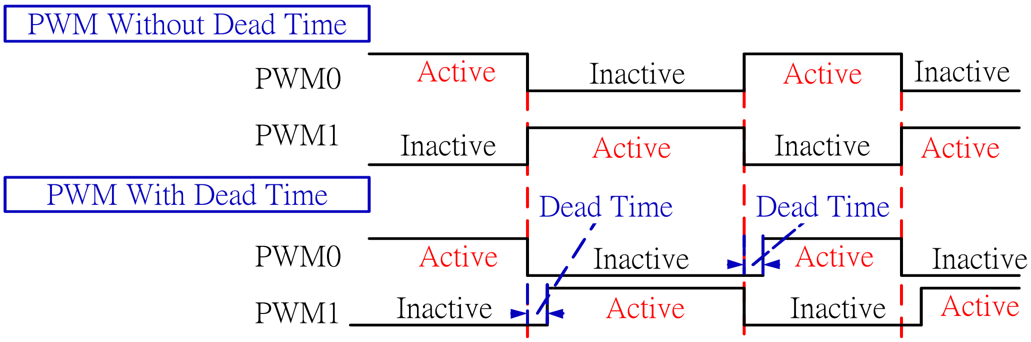
$$\text{Dead Time} = \frac{\text{DT1}[5:0]+1}{\text{DTP1 clock}}$$

※ Increasing the Dead time will shorten the active time, extend the inactive time, and cause a change in the duty cycle.

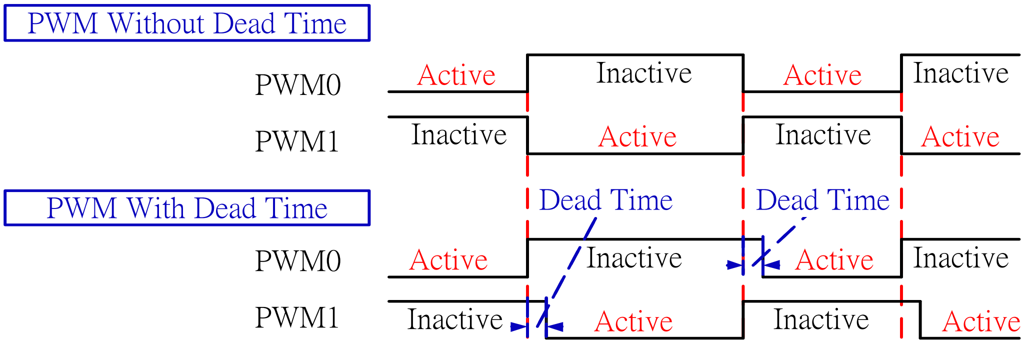
$$\text{PWM period} = \frac{\text{PWMMD}+1}{\text{PWM clock}}$$

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

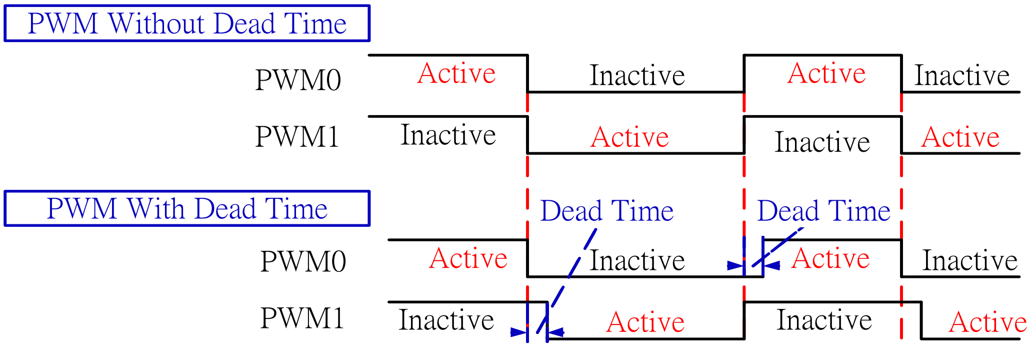
PWM0(Active Hi)
PWM1(Active Hi)



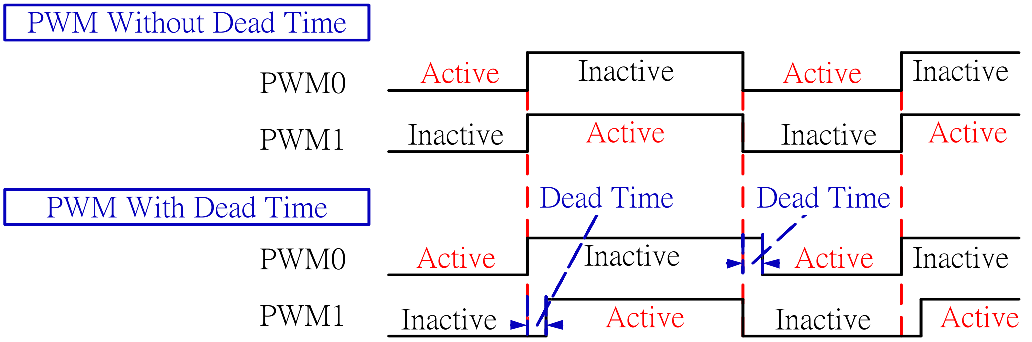
PWM0(Active Low)
PWM1(Active Low)



PWM0(Active Hi)
PWM1(Active Low)



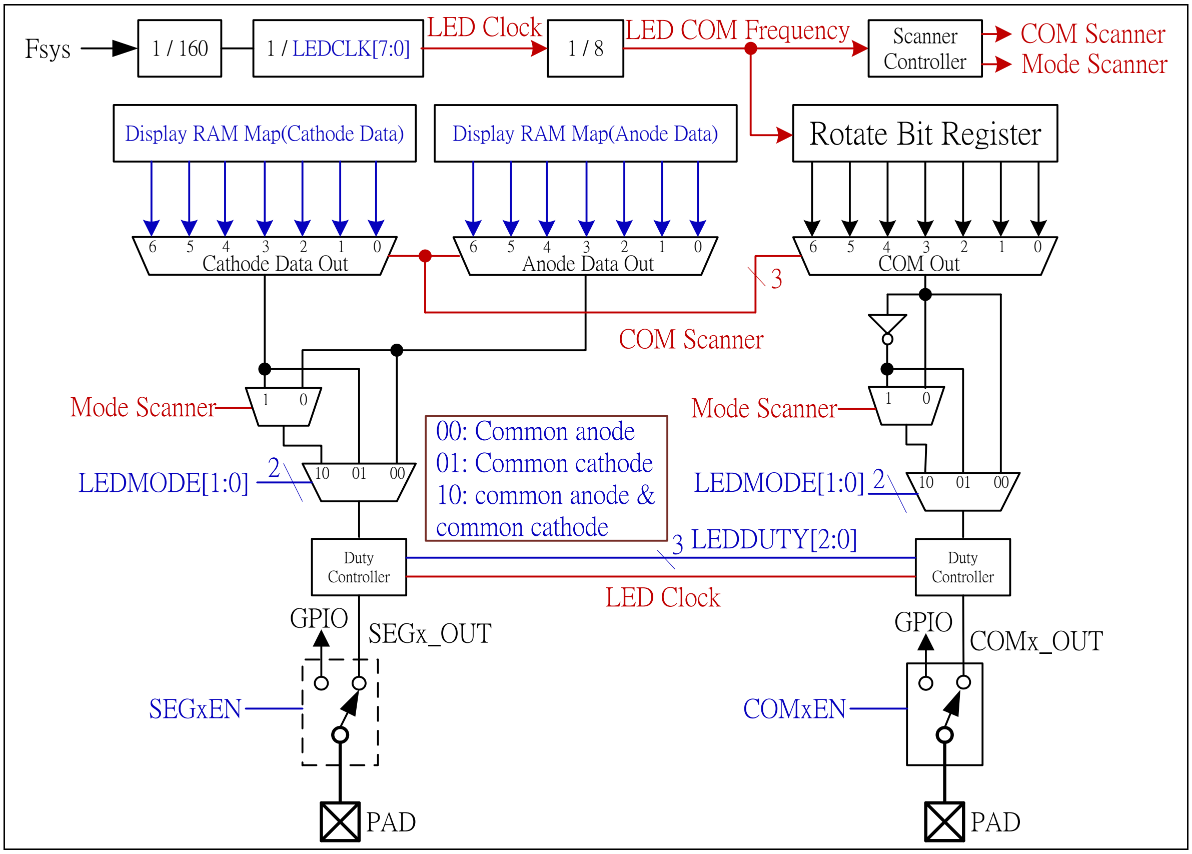
PWM0(Active Low)
PWM1(Active Hi)



17. LED Driver

- **LED selectable duty : 1/8~8/8(full-duty)**
- **LED maximum SEG-driver 16 pins, COM-driver 7 pins**
- **Support common anode、 common cathode and multi-mode waveform.**

Block Diagram



Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
Special Calculation Unit											
AUX	Auxiliary register	91H	BRGS	-	SICS[1:0]		-	SEG1S	SEG0S	-	00H

Description	Indirect	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	RST
COMEN	0xFF1C	-	COM6EN	COM5EN	COM4EN	COM3EN	COM2EN	COM1EN	COM0EN	0x00
SEGEN0	0xFF1D	SEG7EN	SEG6EN	SEG5EN	SEG4EN	SEG3EN	SEG2EN	SEG1EN	SEG0EN	0x00
SEGEN1	0xFF1E	SEG15EN	SEG14EN	SEG13EN	SEG12EN	SEG11EN	SEG10EN	SEG9EN	SEG8EN	0x00
LEDCON	0xFF1F	LEDEN	-	LEDMODE[1:0]		-	LEDDUTY[2:0]			0x00
LEDCLK	0xFF20	LEDCLK[7:0]								0x00

17.1 Display RAM Map

The display RAM is a static 28x8 bits RAM which stores the LED data. A logic “1” state in the RAM bit cell indicates the “ON” state of the corresponding LED segment. Similarly, logic “0” indicates the “OFF” state.

Description	Indirect	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0	RST
COM0_AH	0xFF00	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM0_AL	0xFF01	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM1_AH	0xFF02	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM1_AL	0xFF03	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM2_AH	0xFF04	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM2_AL	0xFF05	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM3_AH	0xFF06	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM3_AL	0xFF07	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM4_AH	0xFF08	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM4_AL	0xFF09	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM5_AH	0xFF0A	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM5_AL	0xFF0B	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM6_AH	0xFF0C	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM6_AL	0xFF0D	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM0_CH	0xFF0E	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM0_CL	0xFF0F	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM1_CH	0xFF10	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM1_CL	0xFF11	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM2_CH	0xFF12	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM2_CL	0xFF13	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM3_CH	0xFF14	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM3_CL	0xFF15	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM4_CH	0xFF16	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM4_CL	0xFF17	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM5_CH	0xFF18	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM5_CL	0xFF19	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00
COM6_CH	0xFF1A	SEG15	SEG14	SEG13	SEG12	SEG11	SEG10	SEG9	SEG8	0x00
COM6_CL	0xFF1B	SEG7	SEG6	SEG5	SEG4	SEG3	SEG2	SEG1	SEG0	0x00

Mnemonic: AUX **Address: 91h**

7	6	5	4	3	2	1	0	Reset
BRGS	-	SICS[1:0]		-	SEG1S	SEG0S	-	00H

SEG1S: Segment 1 pin selection

0: P0.1

1: P3.1

SEG0S: Segment 0 pin selection

0: P0.0

1: P3.0

Mnemonic: COMEN **Indirect Address: 0xFF1C**

7	6	5	4	3	2	1	0	Reset
-	COM6EN	COM5EN	COM4EN	COM3EN	COM2EN	COM1EN	COM0EN	00H

COM6EN: 1: COM6 enable

0: COM6 disable

COM5EN: 1: COM5 enable

0: COM5 disable

COM4EN: 1: COM4 enable

0: COM4 disable

COM3EN: 1: COM3 enable

0: COM3 disable

COM2EN: 1: COM2 enable

0: COM2 disable

COM1EN: 1: COM1 enable

0: COM1 disable

COM0EN: 1: COM0 enable

0: COM0 disable

Mnemonic: SEGEN0 **Indirect Address: 0xFF1D**

7	6	5	4	3	2	1	0	Reset
SEG7EN	SEG6EN	SEG5EN	SEG4EN	SEG3EN	SEG2EN	SEG1EN	SEG0EN	00H

SEG7EN: 1: SEG7 enable

0: SEG7 disable

SEG6EN: 1: SEG6 enable

0: SEG6 disable

SEG5EN: 1: SEG5 enable

0: SEG5 disable

SEG4EN: 1: SEG4 enable

0: SEG4 disable

SEG3EN: 1: SEG3 enable

0: SEG3 disable

SEG2EN: 1: SEG2 enable

0: SEG2 disable

SEG1EN: 1: SEG1 enable

0: SEG1 disable

SEG0EN: 1: SEG0 enable

0: SEG0 disable

Mnemonic: SEGEN1 **Indirect Address: 0xFF1E**

7	6	5	4	3	2	1	0	Reset
SEG15EN	SEG14EN	SEG13EN	SEG12EN	SEG11EN	SEG10EN	SEG9EN	SEG8EN	00H

SEG15EN: 1: SEG15 enable

0: SEG15 disable

SEG14EN: 1: SEG14 enable

0: SEG14 disable

SEG13EN: 1: SEG13 enable

0: SEG13 disable

SEG12EN: 1: SEG12 enable

0: SEG12 disable

SEG11EN: 1: SEG11 enable

0: SEG11 disable

SEG10EN: 1: SEG10 enable

0: SEG10 disable

SEG9EN: 1: SEG9 enable

0: SEG9 disable

SEG8EN: 1: SEG8 enable

0: SEG8 disable

Mnemonic: LEDCON **Indirect Address: 0xFF1F**

7	6	5	4	3	2	1	0	Reset
LEDEN	-	LEDMODE[1:0]		-	LEDDUTY[2:0]			00H

LEDEN: 1: LED scan enable

0: LED scan disable

LEDMODE[1:0]: LED mode Select:

00: Select common anode.

01: Select common cathode.

10: Select common anode & common cathode. (Does not support the use of touch buttons at the same time)

11: Reserve

LEDDUTY[2:0]: LED duty select:

001:1/8 duty

010:2/8 duty

011:3/8 duty

100:4/8 duty

101:5/8 duty

110:6/8 duty

111:7/8 duty

000:8/8duty

Mnemonic: LEDCLK **Indirect Address: 0xFF20**

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

LEDCLK: LED clock select:

LED Clock = Fsys / [160*(LEDCLK[7:0]+1)]

LED COM Frequency = LED Clock/8

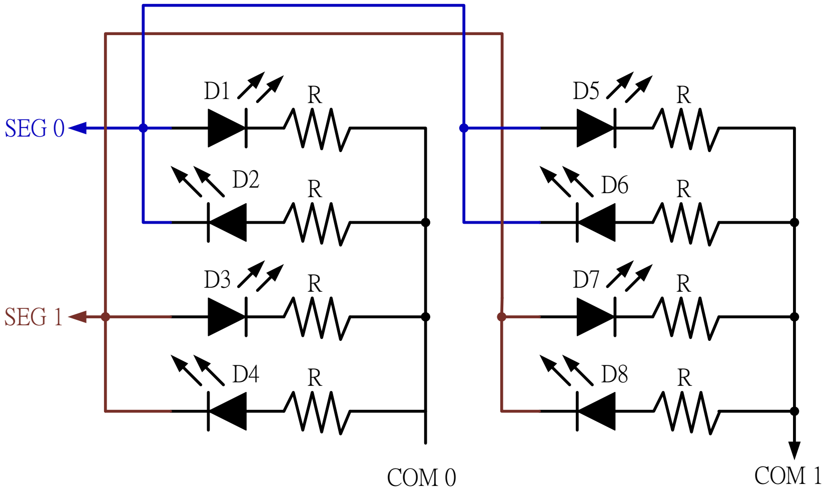
LED Frame Rata= LED COM Frequency /(COM enable Number)

LED Frame Time=(COM enable Number) / LED COM Frequency

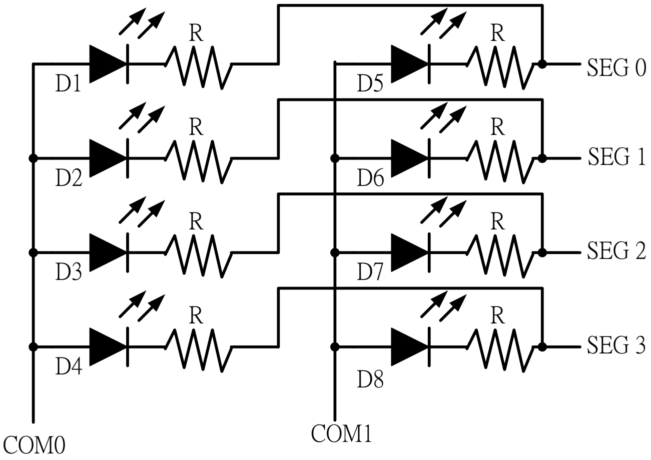
※Setting 0 and 1 are the same.

17.2 LED application circuit

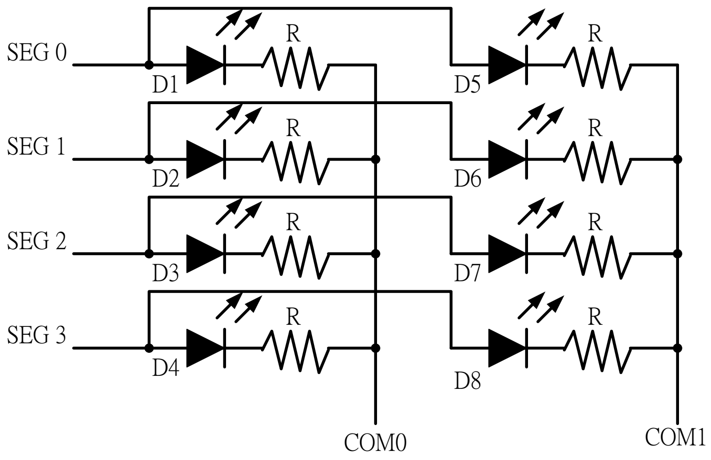
17.2.1 Common Anode & Common Cathode Mode



17.2.2 Common Anode Mode



17.2.3 Common Cathode Mode



18. In-System Programming (Internal ISP)

The OB38R08T1 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 OB38R08T1 from the system. The OB38R08T1 provides internal ROM control signals which can do program /protect functions. User need to design and use any kind of interface which OB38R08T1 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 OB38R08T1 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 OB38R08T1 and host device which output data to the OB38R08T1. For example, if user utilize UART interface to receive/transmit data between OB38R08T1 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 OB38R08T1 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 $3C00 to $3FFF. It can be divided as blocks of N*128 byte. (N=0 to 8). When N=0 means no ISP function, all of 8KB+1KB programmable ROM can be used as program memory. When N=1 means ISP service program occupies 128 byte while the rest of 8.875K 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 8K byte.

After N determined, OB38R08T1 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 $3x00 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 ($3F80h ~ $3FFFh)
2	256 bytes ($3F00h ~ $3FFFh)
3	384 bytes ($3E80h ~ $3FFFh)
4	512 bytes ($3E00h ~ $3FFFh)
5	640 bytes ($3D80h ~ $3FFFh)
6	768 bytes ($3D00h ~ $3FFFh)

7	896 bytes ($3C80h ~ $3FFFh)
8	1024 bytes ($3C00h ~ $3FFFh)

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 OB38R08T1 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 OB38R08T1 enter ISP service program by setting P1.6/P1.7 or P0.7 " 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/P1.7 or P0.7 at reset period to prevent OB38R08T1 from entering ISP service program.

(4) Enter's ISP service program by hardware setting, the P1.5 or P1.6(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, P1.7, P0.7, P1.5 and P2.6. If they meet one of above conditions, chip will switch to ISP mode automatically. After ISP service program executed, user need to reset the OB38R08T1, either by hardware reset or by WDT, or jump to the address $0000 to re-start the firmware program.

There are 5 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 power on reset signal.

(2) Force entry. And triggered by power on reset signal.

(3) (P1.6&P1.7) =0 or P0.7. And triggered by power on reset signal.

(4) P1.5 input 2 clocks. And triggered by power on reset signal.

(5) P2.6 input 2 clocks. And triggered by power on 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	EMF2	EMF3	EMF4	EMF5	ISPF[2:0]			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 OB38R08T1 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	EMF2	EMF3	EMF4	EMF5	ISPF[2:0]			00H

ISP entry mechanism
(1) First Address Blank. I.e. $0000 = FFh
(2) P1.6 = 0 & P1.7=0
(3) P0.7=0
(4) P1.5 input 2 clock
(5) P2.6 input 2 clock
(6) Forced entry

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

EMF2: Entry mechanism (2) 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)

EMF5: Entry mechanism (5) 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 $0F04H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah          ; enable ISPE write attribute
MOV IFCON, #01H          ; enable  ISP function
MOV ISPFAH, #0FH         ; set address-high, 0FH
MOV ISPFAL, #04H         ; set address-low, 04H
MOV ISPFDL, #22H         ; set data to be programmed, data = 22H
MOV ISPFC, #00H             ; start to program #22H to the address $0F04H
```

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 $0F05H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah             ; enable ISPE write attribute
MOV IFCON, #01H             ; enable  ISP function
MOV ISPFAH, #0FH            ; set address-high, 0FH
MOV ISPFAL, #05H            ; set address-low, 05H
MOV ISPFDL, #33H            ; set data to be programmed, data = 33H
MOV ISPFC, #00H             ; start to program #33H to the address $0F05H
```

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 $0F04H

```
MOV TAKEY, #55h
MOV TAKEY, #AAh
MOV TAKEY, #5Ah             ; enable ISPE write attribute
MOV IFCON, #01H             ; enable  ISP function
MOV ISPFAH, #0FH            ; set address-high, 0FH
MOV ISPFAL, #04H            ; set address-low, 04H
MOV ISPFDL, #22H            ; set data to be programmed, data = 22H
MOV ISPFDH, #33H            ; set data to be programmed, data = 33H
MOV ISPFC, #01H             ; start to program #3322H to the address $0F04H
```

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 address-high, 00H
MOV ISPFAL, #04H            ; set address-low, 04H
MOV ISPFDL, #22H            ; set 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 address-high, 00H
MOV ISPFAL, #05H         ; set address-low, 05H
MOV ISPFDL, #33H         ; set 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 address-high, 00H
MOV ISPFAL, #04H         ; set 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 address-high, 00H
MOV ISPFAL, #04H         ; set address-low, 04H
MOV ISPFDL, #22H         ; set data to be programmed, data = 22H
MOV ISPFC, #05H          ; start to program #22H to the 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 address-high, 00H
MOV ISPFAL, #05H        ; set address-low, 05H
MOV ISPFDL, #33H        ; set data to be programmed, data = 33H
MOV ISPFC, #05H         ; start to program #33H to the 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 address-high, 00H
MOV ISPFAL, #04H        ; set 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 $0F04H

```
CLR A
MOV DPTR,#0F04h         ;
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 $0F05H

```
CLR A
MOV DPTR,#0F05h         ;
MOVC A,@A+DPTR          ; read the data of address $0F05 and store it in Accumulator
```

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
VIL	Input Low-voltage	Port 0,1,2,3			0.2Vcc	V	
VIH	Input High-voltage	Port 0,1,2,3	0.8Vcc			V	-
Vhys1	Hysteresis voltage	Port 0,1,2,3		0.2		V	VCC=5V
Vhys2	Hysteresis voltage	I2C		0.4		V	VCC=5V
IOL1	Sink Current	Port 0,1,2,3	40			mA	VOL=0.45V VCC=5V
			25			mA	VOL=0.45V VCC=3V
IOL2	Sink Current	COM0~6	80			mA	VOL=0.45V VCC=5V
						mA	VOL=0.45V VCC=3V
IOH1	Source Current (Pull-Up)	Port 0,1,2,3	0.36			mA	VOH=2.6V VCC=5V
			0.07			mA	VOH=2.4V VCC=3V
IOH2	Source Current (Push-Pull)	Port 0,1,2,3	14			mA	VOH=4.5V VCC=5V
			7			mA	VOH=2.6V VCC=3V
IOH3	Source Current (Push-Pull)	COM0~6	80			mA	VOH=4.8V VCC=5V
						mA	VOH=2.6V VCC=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 ℃

ADC Characteristics

	Symbol	Test Condition	MIN	TYP	MAX	Unit
Operation	VDD	VDD	3		5.5	V
Resolution					12	bit
Conversion time				16tADC		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