Product List

SM59A16U1U48,

Description

The SM59A16U1 is a 1T (one machine cycle per clock) single-chip 8-bit microcontroller. It has 64KB embedded Flash for program, and executes all ASM51 instructions fully compatible with MCS-51.

SM59A16U1 contains 6K+256B on-chip RAM, up to 38 GPIOs (48L 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 SM59A16U1 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.

The SM59A16U1 offers outstanding features, like USB Interface, high performance PWM for motor control applications, high speed 10-bit A/D convert for barcode reader applications. The SM59A16U all features as below.

Ordering Information

SM59A16U1ihhkL yymmv
i: process identifier { U = 2.2V ~ 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}
yy: year
mm: month
v: version identifier{ A, B,…}

Postfix	Package
V	LQFP

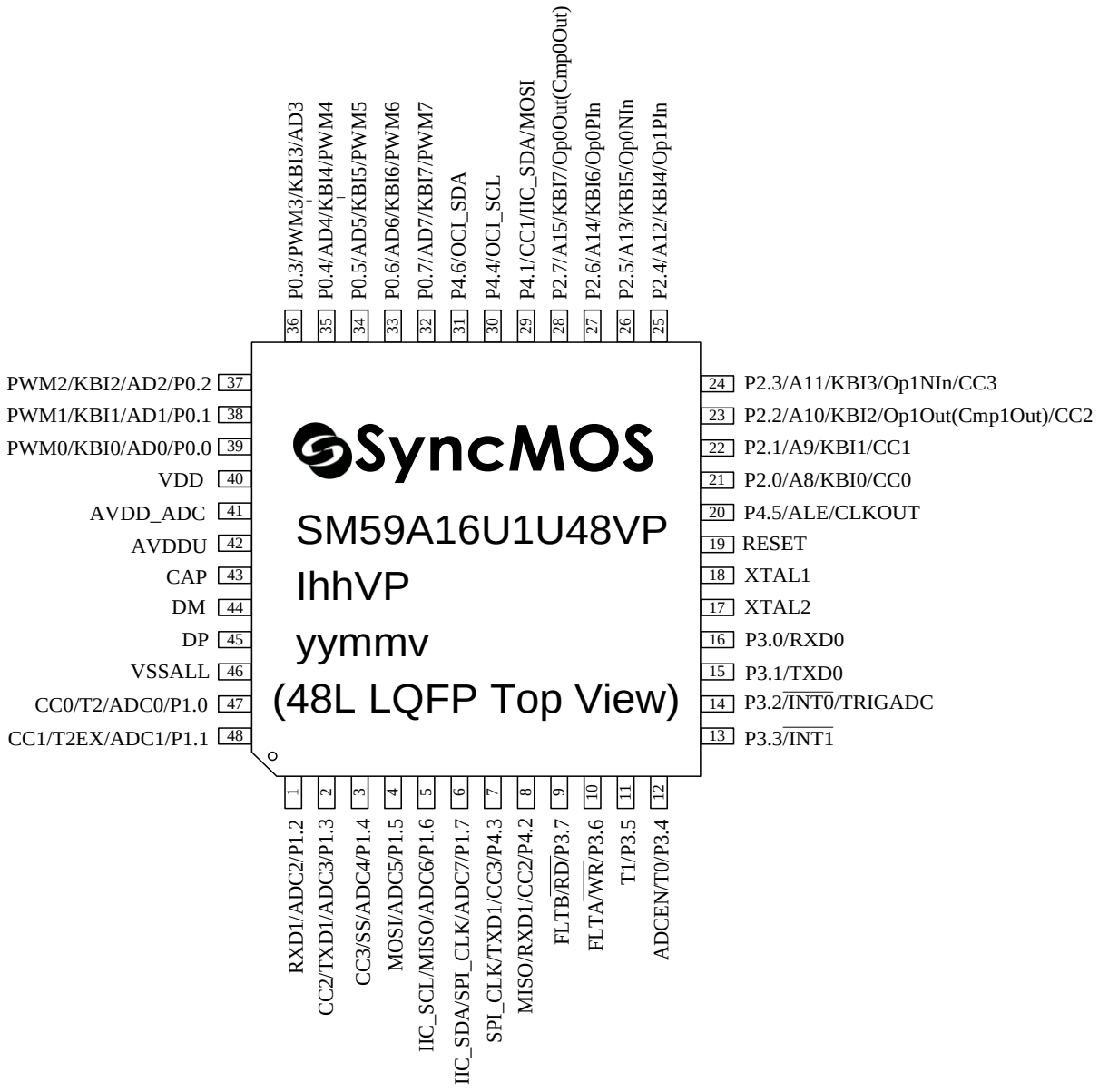
Features

- Architecture
  - Instruction-set compatible with MCS-51
  - 1T/2T can be switched on the fly
  - Dual 16-bit Data Pointers (DPTR0 & DPTR1)
  - 38 GPIOs (LQFP 48), GPIOs can select four types (quasi-bidirectional, push-pull, open drain, input-only),default is quasi-bidirectional(pull-up)
- Clock & Power
  - Operating Voltage: 2.2V ~ 5.5V.
  - Support Xtal, Internal RC Oscillator (22.1184MHz, 20KHz) and PLL to user select.
  - High speed architecture of 1 clock/machine cycle (1T), runs up to 25MHz
  - Power management unit for idle and power down modes.
- Memory.
  - 64KBytes on-chip flash program memory.
  - On–chip flash memories support ISP/IAP/ICP and EEPROM functions.
  - ISP service program space configurable in N*256 byte (N=0 to 16) size.
  - On-chip expandable RAM 6K bytes, 256 bytes RAM as standard 8052.
  - External RAM addresses up to 64K bytes.
- UART Interface.
  - Two serial peripheral interfaces in full duplex mode (UART0 & UART1),
  - Additional Baud Rate Generator for Serial 0
- IIC Interface
  - One IIC interface (Master/Slave mode).
- SPI Interface.
  - One SPI interface (Master/Slave mode)
- KBI (Keyboard Interface).
  - Keyboard interface (KBI) on port 0 or port 2 (default) for eight more interrupts.
- OP Controller
  - 2 On-Chip OPA/Comparator.
- Interrupt Controller
  - interrupts have four priority levels
  - External interrupt 0, 1
- Timer
  - Three 16-bit Timers/Counters. (Timer 0, 1, 2)
  - Programmable watchdog timer (WDT)

- USB Device 2.0
  - Low speed: 1.5Mbps, Full speed: 12Mbps
  - 1 port USB Device
  - 5 Endpoints for USB device
    Endpoint 0: Control IN/OUT. FIFO: 8 bytes
    Endpoint 1: Interrupt IN. FIFO: 8 bytes.
    Endpoint 2: Interrupt OUT. FIFO: 8byte.
    Endpoint 3: Bulk IN. FIFO: 64 bytes.
    Endpoint 4: Bulk OUT. FIFO: 64 bytes.
  - SyncMOS proprietary DFU provide firmware update function by USB
- CCU Controller
  - 4-channel 16-bit compare /capture /load functions
  - Comparator out can be CCU input source internally.
  - Noise filter with CCU input.
- PWM Controller
  - 8-channel 14-bit PWM for BLDC (Brushless DC motors) and CCD barcode reader control.
- A/D Converter
  - 8+1 channel 10-bit analog-to-digital converter
  - Independent ADC reference voltage
  - External I/O triggers ADC
  - ADC auto triggered by specific PWM interrupts.
  - ADC values by DMA dump into SRAM.
- Barcode decoding function.
  - ADC values convert to slope rate then dump into SRAM by DMA
  - Barcode decoding has rise / fall slope setting.
- MDU (Fast multiplication-division unit)
  - 16*16, 32/16, 16/16, 32-bit L/R shifting and 32-bit normalization
- Other
  - On-chip in-circuit emulator (ICE) function with On-Chip Debugger（OCD）
  - Enhanced user code protection
  - EMI reduction mode (ALE output inhibited).
  - LVI/LVR (deglitch 500ns).

Pin Configuration

48 Pin LQFP

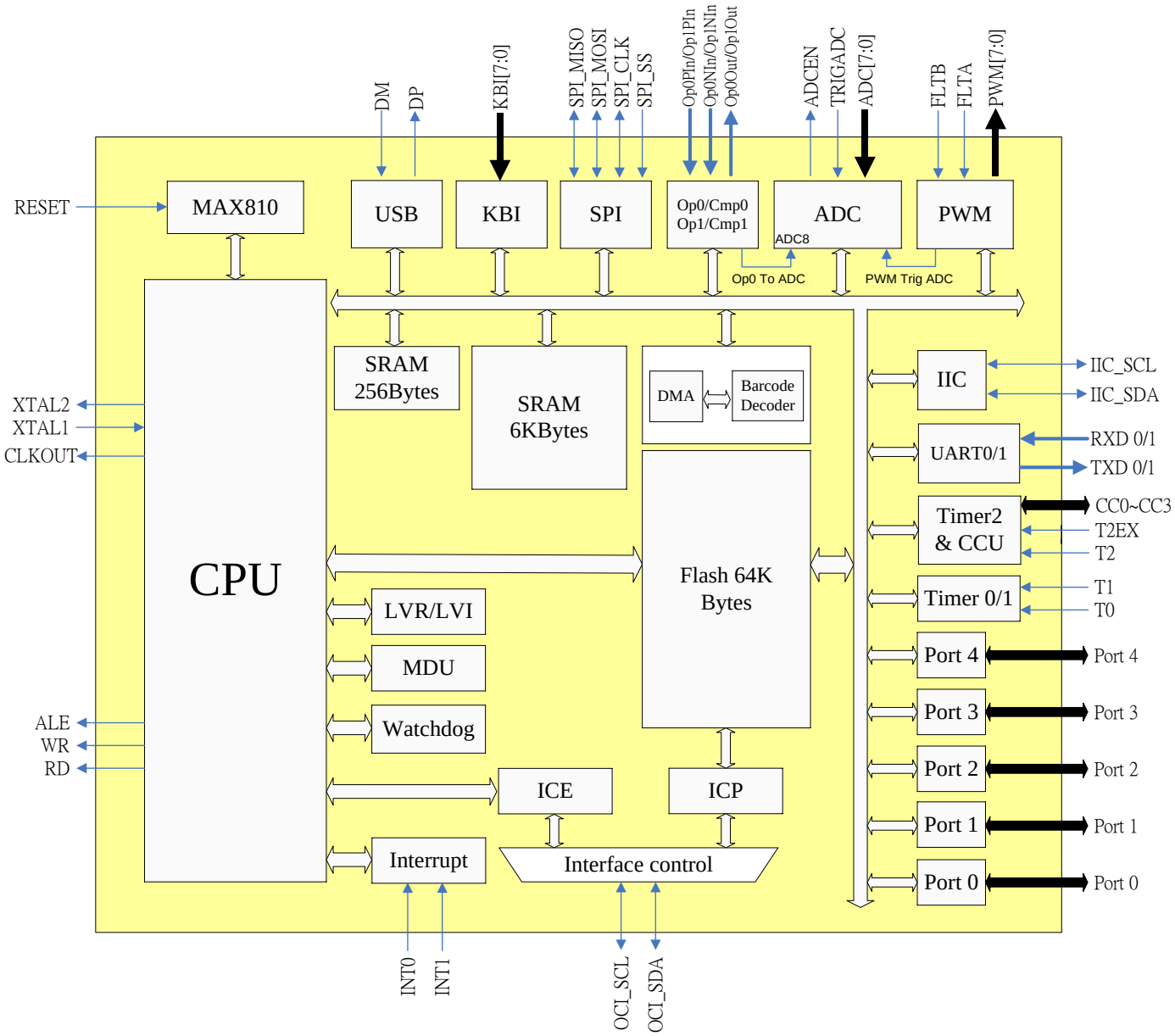


Notes：

(1) To avoid accidentally entering ISP-Mode(refer to section 18.4), care must be taken not asserting pulse signal at RXD P1.0 during power-up while P3.4 are set to high.

(2) To apply ICP function, OCI_SDA/P4.6 and OCI_SCL/P4.4 must be set to Bi-direction mode if they are configured as GPIO in system.

Block Diagram



Pin Description

48L LQFP	Symbol	I/O	Description
1	P1.2/ADC2/RXD1	I/O	* Bit 2 of port 1 * ADC input channel 2 * Serial interface channel 1 receive data
2	P1.3/ADC3/TXD1/CC2	I/O	* Bit 3 of port 1 * ADC input channel 3 * Serial interface channel 1 transmit data or receive clock in mode 0 * Timer 2 compare/capture Channel 2
3	P1.4/ADC4/SS/CC3	I/O	* Bit 4 of port 1 * ADC input channel 4 * SPI interface Slave Select pin * Timer 2 compare/capture Channel 3
4	P1.5/ADC5/MOSI	I/O	* Bit 5 of port 1 * ADC input channel 5 * SPI interface Serial Data Master Output or Slave Input pin
5	P1.6/ADC6/MISO/IIC_SCL	I/O	* Bit 6 of port 1 * ADC input channel 6 * SPI interface Serial Data Master Input or Slave Output pin * IIC SCL pin
6	P1.7/ADC7/SPI_CLK/IIC_SDA	I/O	* Bit 7 of port 1 * ADC input channel 7 * SPI interface Clock pin * IIC SDA pin
7	P4.3/CC3/TXD1/SPI_CLK	I/O	* Bit3 of port 4 * Timer 2 compare/capture Channel 3 * Serial interface channel 1 transmit data * SPI interface Clock pin
8	P4.2/CC2/RXD1/MISO	I/O	* Bit2 of port4 * Timer 2 compare/capture Channel 2 * Serial interface channel 1 receive/transmit data * SPI interface Serial Data Master Input or Slave Output pin
9	P3.7/#RD/FLTB	I/O	* Bit7 of port 3 * External memory Read signal * Fault Input pin
10	P3.6/#WR/FLTA	I/O	* Bit 6 of port 3 * External memory write signal * Fault Input pin
11	P3.5/T1	I/O	* Bit 5 of port 3 * Timer 1 external input
12	P3.4/T0/ADCEN	I/O	* Bit 4 of port 3 * Timer 0 external input * ADC monitor pin
13	P3.3/#INT1	I/O	* Bit 3 of port 3 * External interrupt 1
14	P3.2/#INT0/TRIGADC	I/O	* Bit 2 of port 3 * External interrupt 0 * Trigger ADC
15	P3.1/TXD0	I/O	* Bit 1 of port 3 * Serial interface channel 0 transmit data or receive clock in mode 0

48L LQFP	Symbol	I/O	Description
16	P3.0/RXD0	I/O	* Bit 0 of port 3 * Serial interface channel 0 receive/transmit data
17	XTAL2	O	* Crystal output
18	XTAL1	I	* Crystal input
19	RESET	I	* Reset pin
20	P4.5/ALE/CLKOUT	I/O	* Bit 5 of port 4 * Address latch enable * Internal clock output
21	P2.0/A8/KBI0/CC0	I/O	* Bit 0 of port 2 * Bit 8 of external memory address * KBI interrupt 0 * Timer 2 compare/capture Channel 0
22	P2.1/A9/KBI1/CC1	I/O	* Bit 1 of port 2 * Bit 9 of external memory address * KBI interrupt 1 * Timer 2 compare/capture Channel 1
23	P2.2/A10/KBI2/Op1 Out/CC2	I/O	* Bit 2 of port 2 * Bit 10 of external memory address * KBI interrupt 2 * Op1 output * Timer 2 compare/capture Channel 2
24	P2.3/A11/KBI3/Op1 NIn/CC3	I/O	* Bit 3 of port 2 * Bit 11 of external memory address * KBI interrupt 3 * Op1 Negative Input * Timer 2 compare/capture Channel 3
25	P2.4/A12/KBI4/Op1 PIn	I/O	* Bit 4 of port 2 * Bit 12 of external memory address * KBI interrupt 4 * Op1 Positive Input
26	P2.5/A13/KBI5/Op0 NIn	I/O	* Bit 5 of port 2 * Bit 13 of external memory address * KBI interrupt 5 * Op0 Negative Input
27	P2.6/A14/KBI6/Op0 PIn	I/O	* Bit 6 of port 2 * Bit 14 of external memory address * KBI interrupt 6 * Op0 Positive Input
28	P2.7/A15/KBI7/Op0 Out	I/O	* Bit 7 of port 2 * Bit 15 of external memory address * KBI interrupt 7 * Op0 Output
29	P4.1/CC1/IIC_SDA/ MOSI	I/O	* Bit 1 of port 4 * Timer 2 compare/capture Channel 1 * IIC SDA pin * SPI interface Serial Data Master Output or Slave Input pin
30	P4.4/OCI_SCL	I/O	* Bit 4 of port 4 * On-Chip Instrumentation Clock I/O pin of ICE and ICP functions
31	P4.6/OCI_SDA	I/O	* Bit 6 of port 4 * On-Chip Instrumentation Command and data I/O pin synchronous to OCI_SCL in ICE and ICP functions
32	P0.7/AD7/KBI7/PW M7	I/O	* Bit 7 of port 0 * Bit 7 of external memory address/ data

48L LQFP	Symbol	I/O	Description
			* KBI interrupt 7 * PWM channel 7
33	P0.6/AD6/KBI6/PWM6	I/O	* Bit 6 of port 0 * Bit 6 of external memory address/ data * KBI interrupt 6 * PWM channel 6
34	P0.5/AD5/KBI5/PWM5	I/O	* Bit 5 of port 0 * Bit 5 of external memory address/ data * KBI interrupt 5 * PWM channel 5
35	P0.4/AD4/KBI4/PWM4	I/O	* Bit 4 of port 0 * Bit 4 of external memory address/ data * KBI interrupt 4 * PWM channel 4
36	P0.3/AD3/KBI3/PWM3	I/O	* Bit 3 of port 0 * Bit 3 of external memory address/ data * KBI interrupt 3 * PWM channel 3
37	P0.2/AD2/KBI2/PWM2	I/O	* Bit 2 of port 0 * Bit 2 of external memory address/ data * KBI interrupt 2 * PWM channel 2
38	P0.1/AD1/KBI1/PWM1	I/O	* Bit 1 of port 0 * Bit 1 of external memory address/ data * KBI interrupt 1 * PWM channel 1
39	P0.0/AD0/KBI0/PWM0	I/O	* Bit0 of port 0 * Bit 1 of external memory address/ data * KBI interrupt 0 * PWM channel 0
40	VDD	I	* VDD, 10uF and 0.1uF to GND.
41	AVDD_ADC	I	* ADC VDD
42	AVDDU	I	* VDD
43	CAP	O	* 10uF and 0.1uF to GND.
44	DM	I/O	* USB DM
45	DP	I/O	* USB DP
46	VSSALL	I	* VSS
47	P1.0/ADC0/T2/CC0	I/O	* Bit 0 of port 1 * ADC input channel 0 * Timer 2 external input clock * Timer 2 compare/capture Channel 0
48	P1.1/ADC1/T2EX/CC1	I/O	* Bit 1 of port 1 * ADC input channel 1 * Timer 2 capture trigger * Timer 2 compare/capture Channel 1

Special Function Register (SFR)

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

In-direct access Mode

Hex\Bin	X000	X001	X010	X011	X100	X101	X110	X111	Bin/Hex
F8	IICS	IICCTL	IICA1	IICA2	IICRWD	IICEBT	Cmp0CON	Cmp1CON	**FF**
F0	B	SPIC1	SPIC2	SPITXD	SPIRXD	SPIS	OpPin	TAKEY	**F7**
E8	P4	MD0	MD1	MD2	MD3	MD4	MD5	ARCON	**EF**
E0	ACC	ISPFAH	ISPFAL	ISPFD	ISPFC		LVC	SWRES	**E7**
D8		PFCON	P3M0	P3M1	P4M0	P4M1			**DF**
D0	PSW	CCEN2	P0M0	P0M1	P1M0	P1M1	P2M0	P2M1	**D7**
C8	T2CON	CCCON	CRCL	CRCH	TL2	TH2	OpPin2		**CF**
C0	IRCON	CCEN	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3	**C7**
B8	IEN1	IP1	S0RELH	S1RELH		CLKSEL	PAGESEL		**BF**
B0	P3						WDTC	WDTK	**B7**
A8	IEN0	IP0	S0RELL	ADCC1	ADCC2	ADCDH	ADCDL	ADCCS	**AF**
A0	P2	RSTS	PWM ADDR	PWM DATA	BARCODE ADDR	BARCODE DATA	USB ADDR	USB DATA	**A7**
98	S0CON	S0BUF	IEN2	S1CON	S1BUF	S1RELL			**9F**
90	P1	AUX	AUX2	KBLS	KBE	KBF	KBD	IRCON2	**97**
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	IFCON	**8F**
80	P0	SP	DPL0	DPH0	DPL1	DPH1	RCON	PCON	**87**
Hex\Bin	**X000**	**X001**	**X010**	**X011**	**X100**	**X101**	**X110**	**X111**	**Hex\Bin**

Note: About SFRs correct setting, refer to PAGESEL register.

Page Mode: page0

Hex\Bin	X000	X001	X010	X011	X100	X101	X110	X111	Bin/Hex
F8	IICS	IICCTL	IICA1	IICA2	IICRWD	IICEBT	Cmp0CON	Cmp1CON	**FF**
F0	B	SPIC1	SPIC2	SPITXD	SPIRXD	SPIS	OpPin	TAKEY	**F7**
E8	P4	MD0	MD1	MD2	MD3	MD4	MD5	ARCON	**EF**
E0	ACC	ISPFAH	ISPFAL	ISPFD	ISPFC		LVC	SWRES	**E7**
D8	-	PFCON	P3M0	P3M1	P4M0	P4M1			**DF**
D0	PSW	CCEN2	P0M0	P0M1	P1M0	P1M1	P2M0	P2M1	**D7**
C8	T2CON	CCCON	CRCL	CRCH	TL2	TH2	OPPIN2		**CF**
C0	IRCON	CCEN	CCL1	CCH1	CCL2	CCH2	CCL3	CCH3	**C7**
B8	IEN1	IP1	S0RELH	S1RELH		CLKSEL	PAGESEL		**BF**
B0	P3						WDTC	WDTK	**B7**
A8	IEN0	IP0	S0RELL	ADCC1	ADCC2	ADCDH	ADCDL	ADCCS	**AF**
A0	P2	RSTS							**A7**
98	S0CON	S0BUF	IEN2	S1CON	S1BUF	S1RELL			**9F**
90	P1	AUX	AUX2	KBLS	KBE	KBF	KBD	IRCON2	**97**
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	IFCON	**8F**
80	P0	SP	DPL0	DPH0	DPL1	DPH1	RCON	PCON	**87**
Hex\Bin	**X000**	**X001**	**X010**	**X011**	**X100**	**X101**	**X110**	**X111**	**Bin/Hex**

Page Mode: page1

Hex\Bin	X000	X001	X010	X011	X100	X101	X110	X111	Bin/Hex
F8		PWMTBC0	PWMTBC1	PWMOPMOD	TBCOUNTERL	TBCOUNTERH	UCTRL1	UCTRL2	**FF**
F0	B	PERIODL	PERIODH	SEVTCMPL	SEVTCMPH	PWMEN	USTAT	TAKEY	**F7**
E8	P4	DEADTIME0	DEADTIME1	DEADTIME2	DEADTIME3	PWMSEV	PWMTBPOSTSCALE	LNGDATAL	**EF**
E0	ACC	ISPFAH	ISPFAL	ISPFD	ISPFC		LVC	SWRES	**E7**
D8	-	PFCON	LNGDATAH	FLTCONFIG	FLTNF	PWMPOLARITY	OVRIDEDIS	OVRIDEDATA	**DF**
D0	PSW	DUTY0L	DUTY0H	DUTY1L	DUTY1H	DUTY2L	DUTY2H	DUTY3L	**D7**
C8	T2CON	DUTY3H	BCCTRL	ADDR2ML	TL2	TH2	ADDR2MH		**CF**
C0	IRCON	RDATA	FDATA		DEVADR	FRMNUMH	FRMNUML	HSTALL	**C7**
B8	IEN1	IP1	S0RELH	S1RELH	PWMINTF	CLKSEL	PAGESEL	DSTALL	**BF**
B0	P3	HSKSTAT	UIER1	UIER2	UIFR1	UIFR2	EPDRDY	EP0CNT	**B7**
A8	IEN0	IP0	S0RELL	ADCC1	ADCC2	ADCDH	ADCDL	ADCCS	**AF**
A0	P2	EP1CNT	EP2CNT	EP3CNT	EP4CNT			EP0DATA	**A7**
98	S0CON	S0BUF	IEN2	S1CON	S1BUF	S1RELL	EP1DATA	EP2DATA	**9F**
90	P1	AUX		EP3DATA	EP4DATA			IRCON2	**97**
88	TCON	TMOD	TL0	TL1	TH0	TH1	CKCON	IFCON	**8F**
80	P0	SP	DPL0	DPH0	DPL1	DPH1	RCON	PCON	**87**
Hex\Bin	**X000**	**X001**	**X010**	**X011**	**X100**	**X101**	**X110**	**X111**	**Bin/Hex**

Note: About SFRs correct setting, refer to PAGESEL register.

Note: Special Function Registers reset values and description for SM59A16U1.

Register	Location: 80h ~ 8Fh			Reset value	Description
	Method 1	Method 2 Page 0	Method 2 Page 1		
P0	80h	80h	80h	FFh	Port 0
SP	81h	81h	81h	07h	Stack Pointer
DPL0	82h	82h	82h	00h	Data Pointer 0 Register, Low Byte
DPH0	83h	83h	83h	00h	Data Pointer 0 Register, High Byte
DPL1	84h	84h	84h	00h	Data Pointer 1 Register, Low Byte
DPH1	85h	85h	85h	00h	Data Pointer 1 Register, High Byte
RCON	86h	86h	86h	00h	Internal RAM Control Register
PCON	87h	87h	87h	40h	Power Control Register
TCON	88h	88h	88h	00h	Timer/Counter Control Register
TMOD	89h	89h	89h	00h	Timer Mode Control
TL0	8Ah	8Ah	8Ah	00h	Timer 0 Register, Low Byte
TL1	8Bh	8Bh	8Bh	00h	Timer 1 Register, Low Byte
TH0	8Ch	8Ch	8Ch	00h	Timer 0 Register, High Byte
TH1	8Dh	8Dh	8Dh	00h	Timer 1 Register, High Byte
CKCON	8Eh	8Eh	8Eh	10h	Clock Control Register
IFCON	8Fh	8Fh	8Fh	00h	Interface Control Register
Register	**Location: 90h ~ 9Fh**			**Reset value**	**Description**
	Method 1	**Method 2 Page 0**	**Method 2 Page 1**		
P1	90h	90h	90h	FFh	Port 1
AUX	91h	91h	91h	00h	Auxiliary Register
AUX2	92h	92h	-	00h	Auxiliary 2 Register
KBLS	93h	93h	-	00h	Keyboard Level Selection Register
KBE	94h	94h	-	00h	Keyboard input Enable Register
KBF	95h	95h	-	00h	Keyboard interrupt Flag Register
IRCON2	97h	97h	97h	00h	Interrupt Request Control Register 2
EP3DATA	-	-	93h	00h	USB Endpoint 3 Data Register
EP4DATA	-	-	94h	00h	USB Endpoint 4 Data Register
S0CON	98h	98h	98h	00h	Serial Port 0, Control Register
S0BUF	99h	99h	99h	00h	Serial Port 0, Data Buffer
IEN2	9Ah	9Ah	9Ah	00h	Interrupt Enable Register 2
S1CON	9Bh	9Bh	9Bh	00h	Serial Port 1, Control Register
S1RELL	9Dh	9Dh	9Dh	00h	Serial Port 1, Reload Register, Low Byte
EP1DATA	-	-	9Eh	00h	USB Endpoint 1 Data Register
EP2DATA	-	-	9Fh	00h	USB Endpoint 2 Data Register

Register	Location: A0h ~ AFh			Reset value	Description
	Method 1	Method 2 Page 0	Method 2 Page 1		
P2	A0h	A0h	A0h	FFh	Port 2
PWMADDR	A2h	-	-	00h	PWM Address Register
PWMDATA	A3h	-	-	00h	PWM Data Register
BARCODE ADDR	A4h	-	-	00h	Barcode Address Register
BARCODE DATA	A5h	-	-	00h	Barcode Data Register
USBDATA	A7h	-	-	00h	USB Data Register
EP1CNT	-	-	A1h	00h	USB Endpoint 1 Data Counter Register
EP2CNT	-	-	A2h	00h	USB Endpoint 2 Data Counter Register
EP3CNT	-	-	A3h	00h	USB Endpoint 3 Data Counter Register
EP4CNT	-	-	A4h	00h	USB Endpoint 4 Data Counter Register
EP0DATA	-	-	A7h	00h	USB Endpoint 0 Data Register
IEN0	A8h	A8h	A8h	00h	Interrupt Enable Register 0
IP0	A9h	A9h	A9h	00h	Interrupt Priority Register 0
S0RELL	AAh	AAh	AAh	00h	Serial Port 0, Reload Register, Low Byte
ADCC1	ABh	ABh	ABh	00h	ADC Control 1 Register
ADCC2	ACh	ACh	ACh	00h	ADC Control 2 Register
ADCDH	ADh	ADh	ADh	00h	ADC Data Register, High Byte
ADCDL	AEh	AEh	AEh	00h	ADC Data Register, Low Byte
ADCCS	AFh	AFh	AFh	00h	ADC Clock Select Register
Register	**Location: B0h ~ BFh**			**Reset value**	**Description**
	Method 1	**Method 2 Page 0**	**Method 2 Page 1**		
P3	B0h	B0h	B0h	FFh	Port 3
WDTC	B6h	B6h	-	04h	Watchdog Timer Control Register
WDTK	B7h	B7h	-	00h	Watchdog Timer Refresh Key Register
HSKSTAT	-	-	B1h	80h	USB Handshake Status Register
UIER1	-	-	B2h	00h	USB Interrupt Enable Register 1
UIER2	-	-	B3h	00h	USB Interrupt Enable Register 2
UIFR1	-	-	B4h	00h	USB Interrupt Flag Register 1
UIFR2	-	-	B5h	00h	USB Interrupt Flag Register 2
EPDRDY	-	-	B6h	2Ah	USB Endpoint Data Ready Register
EP0CNT	-	-	B7h	00h	USB Endpoint 0 Data Counter Register
IEN1	B8h	B8h	B8h	00h	Interrupt Enable Register 1
IP1	B9h	B9h	B9h	00h	Interrupt Priority Register 1
S0RELH	BAh	BAh	BAh	00h	Serial Port 0, Reload Register,

					High Byte
S1RELH	BBh	BBh	BBh	00h	Serial Port 1, Reload Register, High Byte
CLKSEL	BDh	BDh	BDh	00h	System Clock Select Register
PAGESEL	BEh	BEh	BEh	00h	SFR Page Mode Select Register
PWMINTF	-	-	BCh	00h	PWM Interrupt Flag Register
DSTALL	-	-	BFh	00h	USB Device Stall Register
Register	**Location: C0h ~ CFh**			**Reset value**	**Description**
	Method 1	**Method 2 Page 0**	**Method 2 Page 1**		
IRCON	C0h	C0h	C0h	00h	Interrupt Request Control Register
CCEN	C1h	C1h	-	00h	Compare/Capture Enable Register
CCH1	C3h	C3h	-	00h	Compare/Capture Register 1, High Byte
CCL2	C4h	C4h	-	00h	Compare/Capture Register 2, Low Byte
CCH2	C5h	C5h	-	00h	Compare/Capture Register 2, High Byte
CCL3	C6h	C6h	-	00h	Compare/Capture Register 3, Low Byte
CCH3	C7h	C7h	-	00h	Compare/Capture Register 3, High Byte
RDATA	-	-	C1h	19h	Barcode Rising of Data Register
FDATA	-	-	C2h	18h	Barcode Falling of Data Register
DEVADR	-	-	C4h	00h	USB Device Address Register
FRMNUMH	-	-	C5h	00h	USB Frame Number Register, High Byte
FRMNUML	-	-	C6h	00h	USB Frame Number Register, Low Byte
HSTALL	-	-	C7h	00h	USB Host Stall Register
T2CON	C8h	C8h	C8h	00h	Timer 2 Control Register
CCCON	C9h	C9h	-	00h	Compare/Capture Control Register
CRCL	CAh	CAh	-	00h	Compare/Reload/Capture Register, Low Byte
TL2	CCh	CCh	CCh	00h	Timer 2 Register, Low Byte
TH2	CDh	CDh	CDh	00h	Timer 2 Register, High Byte
OpPin2	CEh	CEh	-	00h	Op/Comparator Pin Select register 2
DUTY3H	-	-	C9h	00h	PWM 3 Duty Register, High Byte
BCCTRL	-	-	CAh	01h	Barcode Control Register
ADDR2ML	-	-	CBh	00h	Barcode Start address to SRAM Register, Low Byte
ADDR2MH	-	-	CEh	00h	Barcode Start address to SRAM Register, High Byte
Register	**Location: D0h ~ DFh**			**Reset value**	**Description**
	Method 1	**Method 2 Page 0**	**Method 2 Page 1**		

PSW	D0h	D0h	D0h	00h	Program Status Word
CCEN2	D1h	D1h	-	00h	Compare/Capture Enable 2 Register
P0M1	D3h	D3h	-	00h	Port 0 Output Mode 1
P1M0	D4h	D4h	-	00h	Port 1 Output Mode 0
P1M1	D5h	D5h	-	00h	Port 1 Output Mode 1
P2M0	D6h	D6h	-	00h	Port 2 Output Mode 0
P2M1	D7h	D7h	-	00h	Port 2 Output Mode 1
DUTY0L	-	-	D1h	00h	PWM 0 Duty Register, Low Byte
DUTY1L	-	-	D3h	00h	PWM 1 Duty Register, Low Byte
DUTY1H	-	-	D4h	00h	PWM 1 Duty Register, High Byte
DUTY2L	-	-	D5h	00h	PWM 2 Duty Register, Low Byte
DUTY2H	-	-	D6h	00h	PWM 2 Duty Register, High Byte
PFCON	D9h	D9h	D9h	00h	Peripheral Frequency Control Register
P3M0	DAh	DAh	-	00h	Port 3 Output Mode 0
P3M1	DBh	DBh	-	00h	Port 3 Output Mode 1
P4M0	DCh	DCh	-	00h	Port 4 Output Mode 0
P4M1	DDh	DDh	-	00h	Port 4 Output Mode 1
FLTCONFIG	-	-	DBh	80h	PWM Fault Configure Register
FLTNF	-	-	DCh	00h	PWM Fault Noise Filter Register
PWM POLARITY	-	-	DDh	FFh	PWM Polarity Register
OVRIDEDIS	-	-	DEh	FFh	PWM Override Disable Register
OVRIDE DATA		-	DFh	00h	PWM Override Data Register
Register	**Location: E0h ~ EFh**			**Reset value**	**Description**
	Method 1	**Method 2 Page 0**	**Method 2 Page 1**		
ACC	E0h	E0h	E0h	00h	Accumulator
ISPFAH	E1h	E1h	E1h	FFh	ISP Flash Address Register, High Byte
ISPFAL	E2h	E2h	E2h	FFh	ISP Flash Address Register, Low Byte
ISPFD	E3h	E3h	E3h	FFh	ISP Flash Data Register
ISPFC	E4h	E4h	E4h	00h	ISP Flash Control Register
LVC	E6h	E6h	E6h	20h	Low Voltage Control Register
SWRES	E7h	E7h	E7h	00h	Software Reset Register
P4	E8h	E8h	E8h	FFh	Port 4
MD0	E9h	E9h	-	00h	Multiplication/Division Register 0
MD1	EAh	EAh	-	00h	Multiplication/Division Register 1
MD2	EBh	EBh	-	00h	Multiplication/Division Register 2
MD3	ECh	ECh	-	00h	Multiplication/Division Register 3
MD4	EDh	EDh	-	00h	Multiplication/Division Register 4
MD5	EEh	EEh	-	00h	Multiplication/Division Register 5
ARCON	EFh	EFh	-	00h	Arithmetic Control Register

DEADTIME0	-	-	E9h	00h	PWM Dead Time 0 Register
DEADTIME1	-	-	EAh	00h	PWM Dead Time 1 Register
DEADTIME2	-	-	EBh	00h	PWM Dead Time 2 Register
DEADTIME3	-	-	ECh	00h	PWM Dead Time 3 Register
PWMSEV	-	-	EDh	00h	PWM Special Event Register
PWMTBPOST SCALE	-	-	EEh	00h	PWM Time Base Post Scale Register
LNGDATAL	-		EFh	00h	Barcode Length of Data Register, High Byte
Register	**Location: F0h ~ FFh**			**Reset value**	**Description**
	Method 1	**Method 2 Page 0**	**Method 2 Page 1**		
B	F0h	F0h	F0h	00h	B Register
SPIC1	F1h	F1h	-	08h	SPI Control Register 1
SPIC2	F2h	F2h	-	00h	SPI Control Register 2
SPITXD	F3h	F3h	-	00h	SPI Transmit Data Buffer
SPIRXD	F4h	F4h	-	00h	SPI Receive Data Buffer
SPIS	F5h	F5h	-	40h	SPI Status Register
OpPin	F6h	F6h	-	00h	Op/Comparator Pin Select Register
TAKEY	F7h	F7h	F7h	00h	Time Access Key Register
PERIODL	-	-	F1h	FFh	PWM Period Register, Low Byte
PERIODH	-	-	F2h	3Fh	PWM Period Register, High Byte
SEVTCMPL	-	-	F3h	FFh	PWM Special Event Compare Register, Low Byte
SEVTCMPH	-	-	F4h	3Fh	PWM Special Event Compare Register, High Byte
PWMEN	-	-	F5h	00h	PWM Output Enable Register
USTAT	-	-	F6h	00h	USB Status Register
IICS	F8h	F8h	-	00h	IIC Status Register
IICCTL	F9h	F9h	-	04h	IIC Control Register
IICA1	FAh	FAh	-	A0h	IIC Address 1 Register
IICA2	FBh	FBh	-	60h	IIC Address 2 Register
IICRWD	FCh	FCh	-	00h	IIC Read / Write Register
IICEBT	FDh	FDh	-	00h	IIC Enable Bus Transaction Register
Cmp0CON	FEh	FEh	-	00h	Comparator 0 Control Register
Cmp1CON	FFh	FFh	-	00h	Comparator 1 Control Register
PWMTBC0	-	-	F9h	00h	PWM Time Base Control 0 Register
PWMTBC1	-	-	FAh	00h	PWM Time Base Control 1 Register
PWMOPMOD	-	-	FBh	00h	PWM Output Pair Mode Register
TBCOUNTERL	-	-	FCh	00h	PWM Time Base Counter Register, Low Byte
TBCOUNTERH	-	-	FDh	00h	PWM Time Base Counter Register, High Byte
UCTRL1	-	-	FEh	20h	USB Control 1 Register

UCTRL2	-	-	FFh	02h	USB Control 2 Register

Function Description

1. General Features

SM59A16U1 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 Flash

The program can be loaded into the embedded 64KBFlash memory via its writer or In-System Programming (ISP). The high-quality Flash has a 100K-write cycle life，suitable for re-programming and data recording as EEPROM.

1.2 IO Pads

The SM59A16U1 has Five I/O ports: Port 0, Port 1, Port 2 , Port 3 and Port4. Ports 0, 1, 2, 3 are 8-bit ports.. These are: quasi-bidirectional (standard 8051 port outputs), push-pull, open drain, and input-only. As description in section 5.

All the pads for P0、P1、P2、P3 and P4 are with slew rate to reduce EMI. The IO pads can withstand 4KV ESD in human body mode guaranteeing the SM59A16U1's quality in high electro-static environments.

The OCI_SCL、ALE and OCI_SDA can be configured as I/O ports P4.4、P4.5 and P4.6 by writer or in ISP mode.

All the pins on P0 ~ P4 are with slew rate adjustment to reduce EMI. The other way to reduce EMI is to disable the ALE output if unused. This is selected by its SFR. The IO pads can withstand 4KV ESD in human body mode guaranteeing the SM59A16U1's quality in high electro-static environments.

1.3 2T/1T Selection

SM59A16U1 is a 2T or 1T MCU, i.e., its machine cycle is two-clock or one-clock. In the other words, it can execute one instruction within two clocks or only one clock. The difference between 2T mode and 1T mode are given in the example in Fig. 1-1.

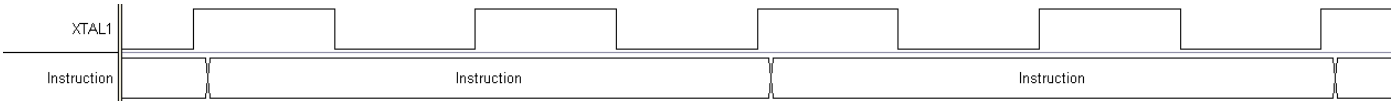


Fig. 1-1: (a)The waveform of internal instruction signal in 2T mode

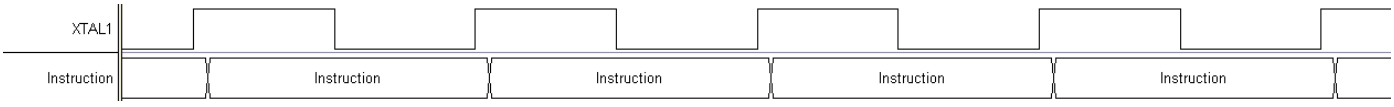


Fig. 1-2: (b)The waveform of internal instruction signal in 1T mode

The default is in 1T mode, 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 RESET

1.4.1 Hardware RESET Function

SM59A16U1 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

1.4.2 Software RESET Function

SM59A16U1 provides one software reset mechaniOB 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	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
Software Reset function											
TAKEY	Time Access Key register	F7h	TAKEY [7:0]								00H
SWRES	Software Reset register	E7h	SWRES [7:0]								00H

1.4.3 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, #AAh
MOV TAKEY, #5Ah
```

1.4.4 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.

1.4.5 Example Of Software Reset

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

1.5 Clocks

SM59A16U1 offers four modes to set the system clock. The system clock can set by writer or ICP.

- IRC: Internal RC-Oscillator and clock is 22.1184MHz fixed (Default).
- 20K: Internal RC-Oscillator and clock is 20K Hz fixed.
- Xtal: External crystal, and may be connected on XTAL1/XTAL2.
- PLL: According to the external crystal generates a fixed 48MHz frequency.
  - System divide clock can't be "DIVIDE 1" in PLL mode; otherwise the PLL (48MHz) will exceed MCU limitation (25MHz).
  - For example to using PLL for system clock:
    Crystal:12MHz
    System Clock: PLL (48MHz fixed).
    System Divide Clock: Divide 2.

MCU generates clock is 24MHz. (48MHz/2)

Note: Recommended to select 6, 12 or 24MHz crystal when USB is used.

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

Table 1-1: Selection of clock source

Clock source
external crystal (use XTAL1 and XTAL2 pins)
external crystal (only use XTAL1, the XTAL2 define as I/O)
22.1184MHz from internal OSC
22.1184MHz/2 from internal OSC
22.1184MHz/4 from internal OSC
22.1184MHz/16 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 SM59A16U1 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 SM59A16U1 Microcontroller core. As given in Table

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
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 SM59A16U1 memory structure follows general 8052 structure. It is integrate the expanded 6KB data memory and 64KB program memory.

3.1 Program Memory

The SM59A16U1 has 64KB on-chip flash memory which can be used as general program memory or EEPROM, on which include up to 4K byte specific ISP service program memory space. The address range for the 64K byte is $0000 to $FFFF. The address range for the ISP service program is $F000 to $FFFF. The ISP service program size can be partitioned as N blocks of 256 byte (N=0 to 16). When N=0 means no ISP service program space available, total 64K byte memory used as program memory. When N=1 means address $FF00 to $FFFF reserved for ISP service program. When N=2 means memory address $FE00 to $FFFF reserved for ISP service program…etc. Value N can be set and programmed into SM59A16U1 by the writer or ICP. It can be used to record any data as EEPROM. The procedure of this EEPROM application function is described in the section 21 on internal ISP. As shown in Fig. 3-1

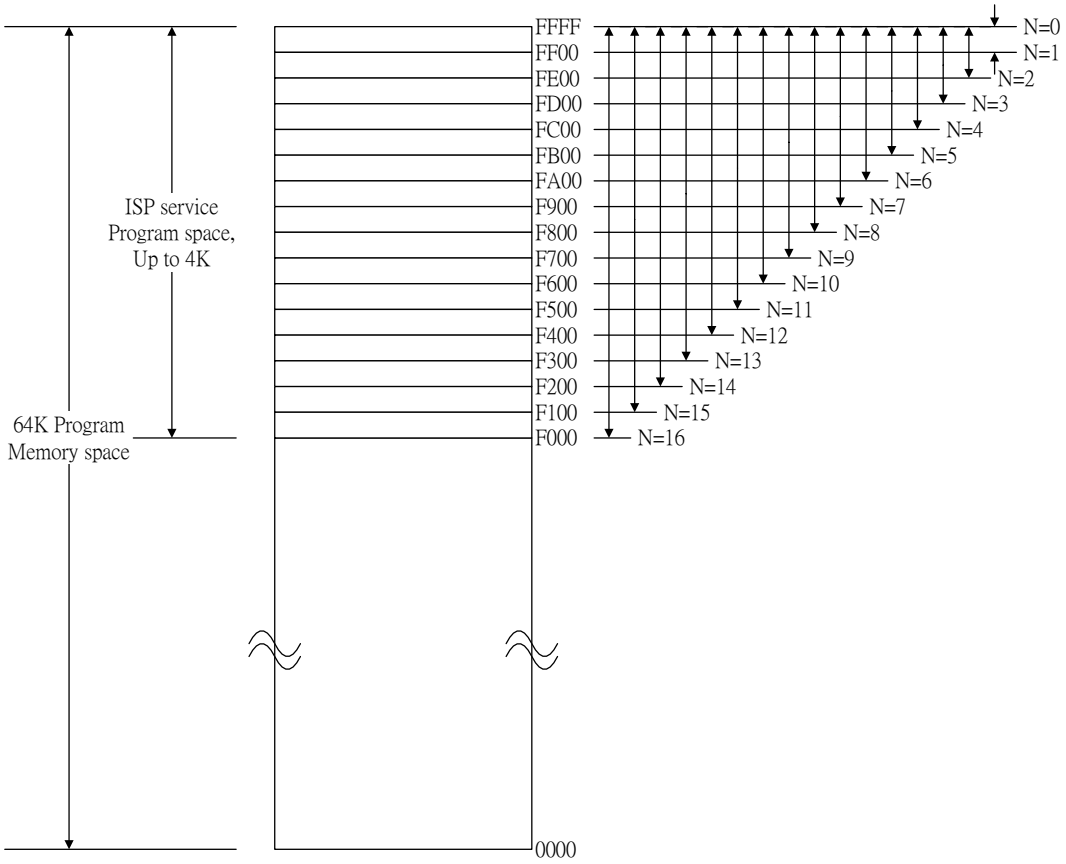


Fig. 3-1: SM59A16U1 programmable Flash

3.2 Data Memory

The SM59A16U1 has 6K+256B on-chip SRAM, 256 Bytes of it are the same as general 8052 internal memory structure while the expanded 6K Bytes on-chip SRAM can be accessed by external memory addressing method(by instruction MOVX.). As shown in Fig. 3-2、Fig. 3-3 and Fig. 3-4

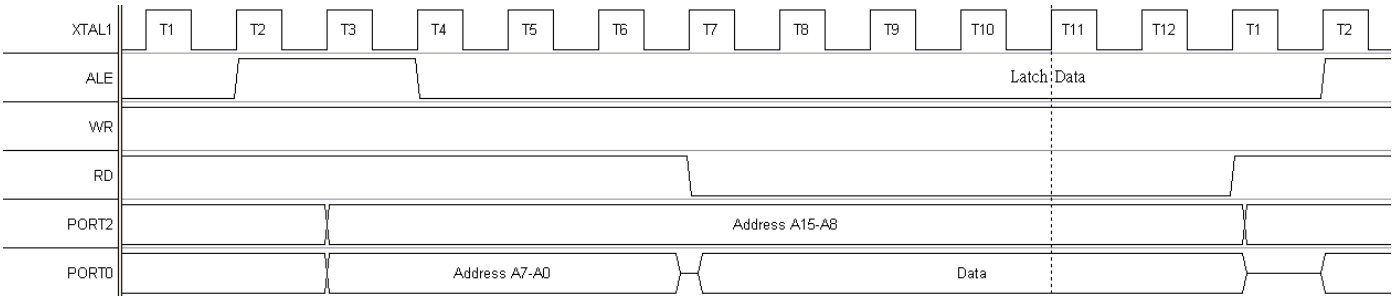


Fig. 3-2: (a)External memory access as read

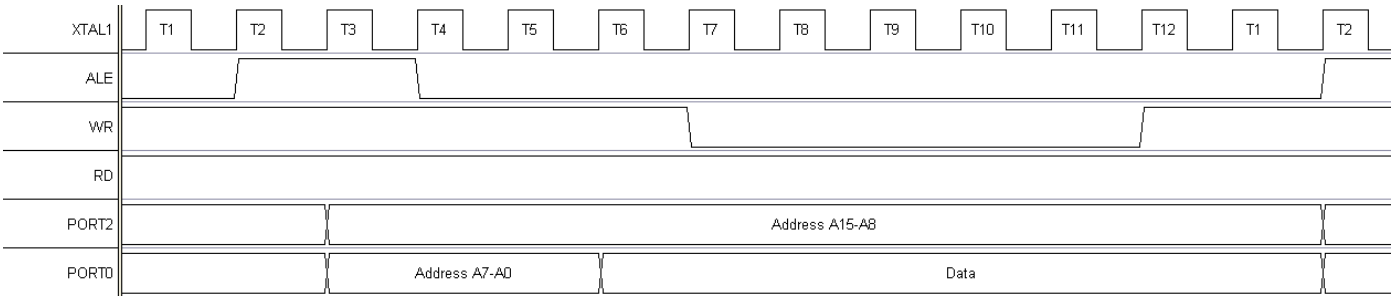


Fig. 3-3: (b)External memory access as write

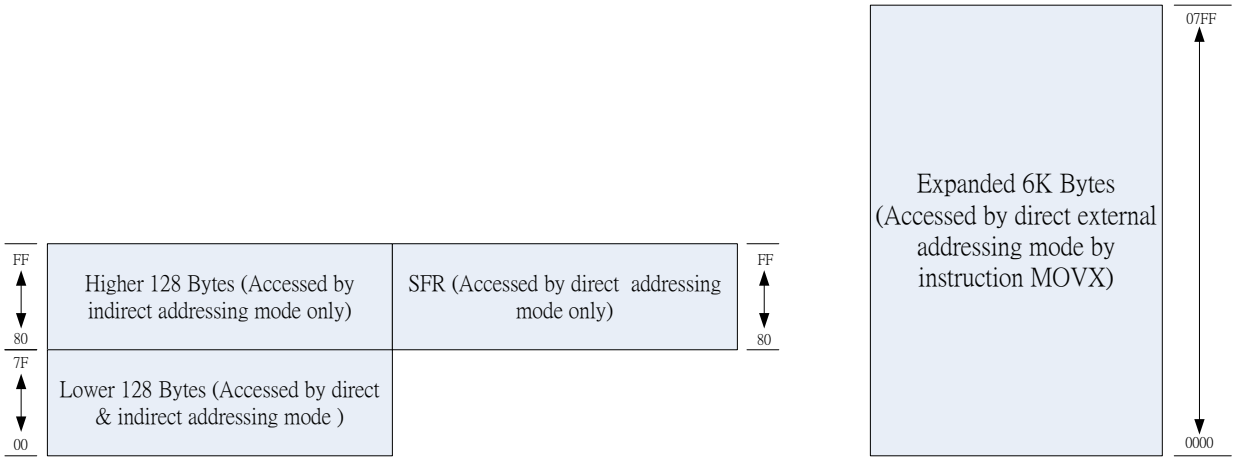


Fig. 3-4: RAM architecture

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 6K Bytes(0000h ~ 0x17FFh)

From external address 0000h to 17FFh is the on-chip expanded SRAM area, total 6K Bytes. This area can be accessed by external direct addressing mode (by instruction MOVX).

If the address of instruction MOVX @DPTR is larger then 17FFh, the SM59A16U1 will generate the external memory control signal automatically.

The address space of instruction MOVX @Ri, i=0, 1 is determined by RCON [7:0] of special function register $86 RCON (internal RAM control register). The default setting of RCON [7:0] is 00h (page0). One page of data RAM is 256 bytes.

When EMEN = 0, the internal 6K expanded RAM is enabled. If access memory space is more than 6K byte, the value of RCON is sent to Port2 to access external RAM.

When EMEN = 1, the internal 6K expanded RAM is disabled. The value of RCON is invalid and high byte address is decided by register context of Port2 register P2 [7:0].

MOVX @Ri, A MOVX A,@Ri	0 ≦ RCON[7:0] ≦ 23	24 ≦ RCON [7:0] ≦ 255
EMEN = 0	Addr [15:8] <= RCON[7:0]	Port2 [7:0] <= P2[7:0]
EMEN = 1	Port2 [7:0] <= P2 [7:0]	Port2 [7:0] <= P2 [7:0]

4. CPU Engine

The SM59A16U1 engine is composed of four components:

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

The SM59A16U1 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
DPL0	Data Pointer Low 0	82h	DPL0[7:0]								00H
DPH0	Data Pointer High 0	83h	DPH0[7:0]								00H
DPL1	Data Pointer Low 1	84h	DPL1[7:0]								00H
DPH1	Data Pointer High 1	85h	DPH1[7:0]								00H
AUX	Auxiliary Register	91h	BRGS	-	P4SPI	P4UR1	P4IIC	P0KBI	-	DPS	00H
RCON	Internal RAM Control Register	86h	RCON[7:0]								00H
CKCON	Clock Control Register	8Eh	CLOCK_READY	ITS[2:0]			-	CLKOUT[2:0]			00H
IFCON	Interface Control Register	8Fh	-	CDPR	F32K	F16K	-		EMEN	ISPE	00H
PAGESEL	SFR Page Mode Select Register	BEh	-						Page_num	Page_mode	00H
PWMADDR	PWM Address Register	A2h	PWMADDR[7:0]								00H
PWMDATA	PWM Data Register	A3h	PWMDATA[7:0]								00H
USBADDR	USB Address Register	A6h	USBADDR[7:0]								00H
USBDATA	USB Data Register	A7h	USBDATA[7:0]								00H

4.1 Accumulator

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

Mnemonic: ACC **Address: E0h**

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

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

4.2 B Register

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

Mnemonic: B **Address: F0h**

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

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

4.3 Program Status Word (PSW)

Mnemonic: PSW **Address: D0h**

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

CY: Carry flag.

AC: Auxiliary Carry flag for BCD operations.

F0: General purpose Flag 0 available for user.

RS[1:0]: Register bank select, used to select working register bank.

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

OV: Overflow flag.

F1: General purpose Flag 1 available for user.

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

4.4 Stack Pointer (SP)

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

Mnemonic: SP **Address: 81h**

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

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

4.5 Data Pointer(DP)

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

Mnemonic: DPL **Address: 82h**

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

DPL[7:0]: Data pointer Low 0

Mnemonic: DPH **Address: 83h**

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

DPH [7:0]: Data pointer High 0

4.6 Data Pointer 1(DP1)

The Dual Data Pointer accelerates the moves of data block. The standard DPTR is a 16-bit register that is used to address external memory or peripherals. In the SM59A16U1 core the standard data pointer is called DPTR, the second data pointer is called DPTR1. The data pointer select bit chooses the active pointer. The data pointer select bit is located in LSB of AUX register (DPS).

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

Mnemonic: DPL1 **Address: 84h**

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

DPL1[7:0]: Data pointer Low 1

Mnemonic: DPH1 **Address: 85h**

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

DPH1[7:0]: Data pointer High 1

4.7 Auxiliary Register(AUX)

Mnemonic: AUX **Address: 91h**

7	6	5	4	3	2	1	0	Reset
BRGS	-	P4SPI	P4UR1	P4IIC	P0KBI	-	DPS	00H

BRGS: 0 = Baud Rate Generator by Timer 1 Register.
1 = Baud Rate Generator by Serial Port Register.

P4SPI: 0 = SPI function on P1.
1 = SPI function on P4

P4UR1: 0 = Serial interface 1 function on P1.
1 = Serial interface 1 function on P4.

P4IIC: 0 = IIC function on P1.
1 = IIC function on P4.

P0KBI: 0 = KBI function on P2.
1 = KBI function on P0.

DPS: DPS = 0 is selected DPTR0.
DPS = 1 is selected DPTR1.

4.8 Internal RAM Control Register(RCON)

SM59A16U1 has 6K byte on-chip expanded RAM which can be accessed by external memory addressing method only (By instruction MOVX). The address space of instruction MOVX @Ri, i= 0, 1 is determined by RCON [7:0] of RCON. The default setting of RCON [7:0] is 00h (page0).

Mnemonic: RCON **Address: 86h**

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

4.9 Clock Control Register(CKCON)

The register is used to select instruction timing and clock out selected.

Mnemonic: CKCON **Address: 8Eh**

7	6	5	4	3	2	1	0	Reset
CLOCK_ READY	ITS[2:0]			-	CLKOUT[2:0]			00H

CLOCK_READY: Clock Ready flag

When change clock source on the fly, SW must check this flag;

If this bit be set, means clock source is stable, HW can keep working normally.

ITS[2:0]: Instruction timing select.

ITS [2:0]	Mode
000	1T instruction mode
001	2T instruction mode

CLKOUT[2:0]: Clock output select.

CLKOUT[2:0]	Mode
000	ALE (default)
100	P4.5
x01	Fosc
x10	Fosc/2
x11	Fosc/4

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

4.10 Interface Control Register(IFCON)

Mnemonic: IFCON **Address: 8Fh**

7	6	5	4	3	2	1	0	Reset
-	CDPR	F32K	F16K	-		EMEN	ISPE	00H

CDPR: code protect (Read Only)

F32K: flash size is 32KB+4KB (Read Only)

F16K: flash size is 16KB+4KB (Read Only)

EMEN: Internal 6K SRAM enable.(default is enable)

EMEN = 0, Enable internal 6K RAM and external 58K RAM.

EMEN = 1, Disable internal 6K RAM, Enable external 64K RAM.

ISPE: ISP function enable bit

ISPE = 1, enable ISP function

ISPE = 0, disable ISP function

4.11 Page Select(PAGESEL)

The SM59A16U1 provide two different methods to set Special Function Register (SFR) are as follow:

- SFR Method 1 (Indirect Mode): This method is only an SFR page. If you want to use PWM or USB registers of the Method 2, can be used indirectly addressable setting.
  Example: Write a data 0x80h to PWMEN Register in Method 1.

```
PAGESEL = 0x0h;     // Method 1.
PWMADDR = 0xF5h;    // PWMEN indirect address: 0xF5h (Indirect mode)
                    // (Refer Page1 Table of the Method 2)
PWMDATA = 0x80h;    // Write data 0x80h to PWMEN.
```

- SFR Method 2 (Page Mode): This method provides two SFR page to set the registers.
  Example: Write a data 0x80h to PWMEN Register in Method 2, Page 1.

```
PAGESEL = 0x3h;     // Method 2, Page 1 (Page mode)
PWMEN = 0x80h;      // Write data 0x80h to PWMEN.
```

- SFR Page Mode Table:

Page_mode	Page_num	SFR Select
0	0	SFR Method 1
0	1	SFR Method 1
1	0	SFR Method 2, Page 0
1	1	SFR Method 2, Page 1

Mnemonic: PAGESEL **Address: BEh**

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

Page_num: This flag is used only in the SFR method 2.

0: page 0 mode.

1: page 1 mode.

Page_mode: This flag is used to select SFR register table.

0: SFR Method 1 (indirect mode).

1: SFR Method 2 (page mode).

4.12 PWM Address Register(PWMADDR)

Mnemonic: PWMADDR **Address: A2h**

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

PWMADDR: PWM address register and can only use in SFR method 1 (Indirect mode).
PWMADDR and PWMDATA need to be used together.

4.13 PWM Data Register(PWMDATA)

Mnemonic: PWMDATA **Address: A3h**

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

PWMDATA: PWM data register and can only use in SFR method 1 (Indirect mode).
PWMDATA and PWMADDR need to be used together.

- Read a data from PWM register in SFR Method 1 (Indirect Mode):

Example: Read the PWMSEV data in SFR Method 1.

```
PAGESEL = 0x0h;       // Method 1.
PWMADDR = 0xEDh;      // PWMSEV indirect address: 0xEDh (Indirect mode)
                      // (Refer Page1 Table of the Method 2)
Val = PWMDATA;        // Val: Read data from PWMSEV.
```

- Read a data from PWM register in SFR Method 2 (Page Mode):
  Example: Read the PWMSEV data in SFR Method 2, Page 1.
  PAGESEL = 0x3h; // Method 2, Page 1 (Page mode)
  Val = PWMSEV; // Val: Read data from PWMSEV.
- The PWM Method1 and PWM Method 2 is same result.

4.14 USB Address Register(USBADDR)

Mnemonic: USBADDR **Address: A6h**

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

USBADDR: USB address register and can only use in SFR method 1 (Indirect mode).
USBADDR and USBDATA need to be used together.

4.15 USB Data Register(USBDATA)

Mnemonic: USBDATA **Address: A7h**

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

USBDATA: USB data register and can only use in SFR method 1 (Indirect mode).
USBDATA and USBADDR need to be used together.

- Write data to USB Register in SFR Method 1 (Indirect Mode):
  Example: Write data 0x1h to UCTRL1 Register in SFR Method 1.

```
PAGESEL = 0x0h;     // Method 1.
USBADDR = 0xFEh;    // UCTRL1 indirect address: 0xFEh (Indirect mode)
                    // (Refer Page1 Table of the Method 2)
USBDATA = 0x1h;     // Write data 0x01h to UCTRL1.
```

- Write a data to USB Register in SFR Method 2 (Page Mode):
  Example: Write data 0x1h to UCTRL1 Register in SFR Method 2, Page 1.

```
PAGESEL = 0x3h;     // Method 2, Page 1 (Page mode)
UCTRL1 = 0x01h;     // Write data 0x01h to UCTRL1.
```

- The USB Method1 and USB Method 2 is same result.

5. GPIO

The SM59A16U1 has four I/O ports: Port 0, Port 1, Port 2, Port 3 and Port 4. Ports 0, 1, 2, 3, are 8-bit ports and Port 4 is a 6-bit port. These are: quasi-bidirectional (standard 8051 port outputs), push-pull, open drain, and input-only. Two configuration registers for each port select the output type for each port pin. All I/O port pins on the SM59A16U1 may be configured by software to one of four types on a pin-by-pin basis, shown as below:

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
I/O port function register											
P0M0	Port 0 output mode 0	D2h	P0M0 [7:0]								00H
P0M1	Port 0 output mode 1	D3h	P0M1[7:0]								00H
P1M0	Port 1 output mode 0	D4h	P1M0[7:0]								00H
P1M1	Port 1 output mode 1	D5h	P1M1[7:0]								00H
P2M0	Port 2 output mode 0	D6h	P2M0[7:0]								00H
P2M1	Port 2 output mode 1	D7h	P2M1[7:0]								00H
P3M0	Port 3 output mode 0	DAh	P3M0[7:0]								00H
P3M1	Port 3 output mode 1	DBh	P3M1[7:0]								00H
P4M0	Port 4 output mode 0	DCh	-	P4M 0.6	P4M 0.5	P4M 0.4	P4M 0.3	P4M 0.2	P4M 0.1	-	00H
P4M1	Port 4 output mode 1	DDh	-	P4M 1.6	P4M 1.5	P4M 1.4	P4M 1.3	P4M 1.2	P4M 1.1	-	00H

Note: P0 is input only, when reset assert (even P1M0 reset value is 00H).

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

The OCI_SCL、ALE and OCI_SDA can be define as P4.4、P4.5 and P4.6 by writer or ISP。

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

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
Ports											
Port 0	Port 0	80h	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0	FFh
Port 1	Port 1	90h	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	FFh
Port 2	Port 2	A0h	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	FFh
Port 3	Port 3	B0h	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	FFh
Port 4	Port 4	E8h	-	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	-	FFh

5.1 P0 (Port 0 Register)

Mnemonic: P0 **Address: 80h**

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

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

5.2 P1 (Port 1 Register)

Mnemonic: P1 **Address: 90h**

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

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

5.3 P2 (Port 2 Register)

Mnemonic: P2 **Address: A0h**

7	6	5	4	3	2	1	0	Reset
P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	FFh

P2.7~ 0: Port2 [7] ~ Port2 [0]

5.4 P3 (Port 3 Register)

Mnemonic: P3 **Address: B0h**

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

P3.7~ 0: Port3 [7] ~ Port3 [0]

5.5 P4 (Port 4 Register)

Mnemonic: P4 **Address: E8h**

7	6	5	4	3	2	1	0	Reset
-	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	-	FFh

P4.6~ 1: Port4 [6] ~ Port4 [1]

6. Multiplication Division Unit(MDU)

This on-chip arithmetic unit provides 32-bit division, 16-bit multiplication, shift and normalize features, etc. All operations are unsigned integer operations.

Table 6-1: 乘除寄存器

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
The relevant registers of the Multiplication Division Unit											
PCON	Power control	87H	SMOD	MDUF	-				STOP	IDLE	40H
Multiplication Division Unit											
ARCON	Arithmetic Control register	EFh	MDEF	MDOV	SLR	SC[4:0]					00H
MD0	Multiplication/Division Register 0	E9h	MD0[7:0]								00H
MD1	Multiplication/Division Register 1	EAh	MD1[7:0]								00H
MD2	Multiplication/Division Register 2	EBh	MD2[7:0]								00H
MD3	Multiplication/Division Register 3	ECh	MD3[7:0]								00H
MD4	Multiplication/Division Register 4	EDh	MD4[7:0]								00H
MD5	Multiplication/Division Register 5	EEh	MD5[7:0]								00H

6.1 Operating Registers of the MDU

The MDU is handled by seven registers, which are memory mapped as special function registers. The arithmetic unit allows operations concurrently to and independent of the CPU's activity. Operands and results registers are MD0 to MD5. Control register is ARCON. Any calculation of the MDU overwrites its operands.

Mnemonic: ARCON **Address: EFh**

7	6	5	4	3	2	1	0	Reset
MDEF	MDOV	SLR	SC[4:0]					00H

MDEF: Multiplication Division Error Flag.

The MDEF is an error flag. The error flag is read only. The error flag indicates an improperly performed operation (when one of the arithmetic operations has been restarted or interrupted by a new operation). The error flag mechanism is automatically enabled with the first write to MD0 and disabled with the final read instruction from MD3 multiplication or shift/normalizing) or MD5 (division) in phase three.

The error flag is set when:

1. Phase two in process and write access to mdx registers (restart or interrupt calculations)

The error flag is reset only if:

Phase two finished (arithmetic operation successful completed) and read access to MDx registers.

MDOV: Multiplication Division Overflow flag. The overflow flag is read only.

The overflow flag is set when:

Division by Zero

Multiplication with a result greater then 0000FFFFh

Start of normalizing if the most significant bit of MD3 is set(MD3.7=1)

The overflow flag is reset when:

Write access to MD0 register(Start Phase one)

SLR: Shift direction bit.

SLR = 0 – shift left operation.

SLR = 1 – shift right operation.

SC[4:0]: Shift counter.

When preset with 00000b, normalizing is selected. After normalize sc.0 – sc.4 contains the number of normalizing shifts performed. When sc.4 – sc.0 ≠ 0, shift operation is started. The number of shifts performed is determined by the count written to sc.4 to sc.0.

sc.4 – MSB ... sc.0 – LSB

6.2 Operation of the MDU

The operation of the MDU consists of three phases:

6.2.1 First phase: loading the MDx registers, x = 0~5:

The type of calculation the MDU has to perform is selected following the order in which the mdx registers are written to.

Table 6-2: MDU registers write sequence

Operation	32bit/16bit	16bit/16bit	16bit x 16bit	shift/normalizing
First write	MD0 Dividend Low	MD0 Dividend Low	MD0 Multiplicand Low	MD0 LSB
	MD1 Dividend	MD1 Dividend High	MD4 Multiplicator Low	MD1
	MD2 Dividend		MD1 Multiplicand High	MD2
	MD3 Dividend High			MD3 MSB
	MD4 Divisor Low	MD4 Divisor Low		
Last write	MD5 Divisor High	MD5 Divisor High	MD5 Multiplicator High	ARCON start conversion

A write to md0 is the first transfer to be done in any case. Next writes must be done as shown in Table 6-1 to determine MDU operation. Last write finally starts selected operation.

6.2.2 Second phase: executing calculation.

During executing operation, the MDU works on its own parallel to the CPU. When MDU is finished, the MDUF register will be set to one by hardware and the flag will clear at next calculation.

Mnemonic: PCON **Address: 87h**

7	6	5	4	3	2	1	0	Reset
SMOD	MDUF	-				STOP	IDLE	40h

MDUF: MDU finish flag.

When MDU is finished, the MDUF will be set by hardware and the bit will clear by hardware at next calculation.

The following table gives the execution time in every mathematical operation.

Table 6-3: MDU execution times

Operation	Number of Tclk
Division 32bit/16bit	17 clock cycles
Division 16bit/16bit	9 clock cycles
Multiplication	11 clock cycles
Shift	Min. 3 clock cycles, Max. 18 clock cycles
Normalize	Min. 4 clock cycles, Max. 19 clock cycles

6.2.3 Third phase: reading the result from the MDx registers.

Read out sequence of the first MDx registers is not critical but the last read (from MD5 - division and MD3 - multiplication, shift and normalizing) determines the end of a whole calculation (end of phase three).

Table 6-4: MDU registers read sequence

Operation	32Bit/16Bit	16Bit/16Bit	16Bit x 16Bit	shift/normalizing
First read	MD0 Quotient Low	MD0 Quotient Low	MD0 Product Low	MD0 LSB
	MD1 Quotient	MD1 Quotient High	MD1 Product	MD1
	MD2 Quotient		MD2 Product	MD2
	MD3 Quotient High			
	MD4 Remainder L	MD4 Remainder Low		
Last read	MD5 Remainder H	MD5 Remainder High	MD3 Product High	MD3 MSB

Here the operation of normalization and shift will be explained more. In normalization, all reading zeroes in registers MD0 to MD3 are removed by shift left. The whole operation is completed when the MSB (most significant bit) of MD3 register contains a '1'. After normalizing, bits ARCON.4 (MSB) to ARCON.0 (LSB) contain the number of shift left operations. As for shift, SLR bit (ARCON.5) has to contain the shift direction, and ARCON.4 to ARCON.0 represent the shift count (which must not be 0). During shift, zeroes come into the left or right end of the registers MD0 or MD3, respectively.

6.3 Normalizing

All reading zeroes of integers variables in registers MD0 to MD3 are removed by shift left operations. The whole operation is completed when the MSB (most significant bit) of MD3 register contains a '1'. After normalizing, bits ARCON.4 (MSB) to ARCON.0 (LSB) contain the number of shift left operations, which were done.

6.4 Shifting

SLR bit (ARCON.5) has to contain the shift direction, and ARCON.4 to ARCON.0 the shift count (which must not be 0). During shift, zeroes come into the left or right end of the registers MD0 or MD3, respectively.

7. Timer 0 and Timer 1

The SM59A16U1 has three 16-bit timer/counter registers: Timer 0, Timer 1 and Timer 2. All can be configured for counter or timer operations.

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

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

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

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

7.1 Timer/Counter Mode Vontrol Register (TMOD)

Mnemonic: TMOD **Address: 89h**

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

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

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

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.

7.2 Timer/Counter Control Register(TCON)

Mnemonic: TCON **Address: 88h**

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

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

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

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

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

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

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

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

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

7.3 Timer 0 Register(TL0, TH0)

Mnemonic: TL0 **Address: 8Ah**

7	6	5	4	3	2	1	0	Reset
TL0[7:0]								00H

Mnemonic: TH0 **Address: 8Ch**

7	6	5	4	3	2	1	0	Reset
TH0[7:0]								00H

7.4 Timer 1 Register(TL1, TH1)

Mnemonic: TL1 **Address: 8Bh**

7	6	5	4	3	2	1	0	Reset
TL1[7:0]								00H

Mnemonic: TH0 **Address: 8Dh**

7	6	5	4	3	2	1	0	Reset
TH1[7:0]								00H

7.5 Peripheral Frequency Control Register

Mnemonic: PFCON **Address: D9h**

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

T1PS[1:0]: Timer1 Prescaler select

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

T0PS[1:0] Timer0 Prescaler select

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

7.6 Mode 0(13-bit Counter/Timer)

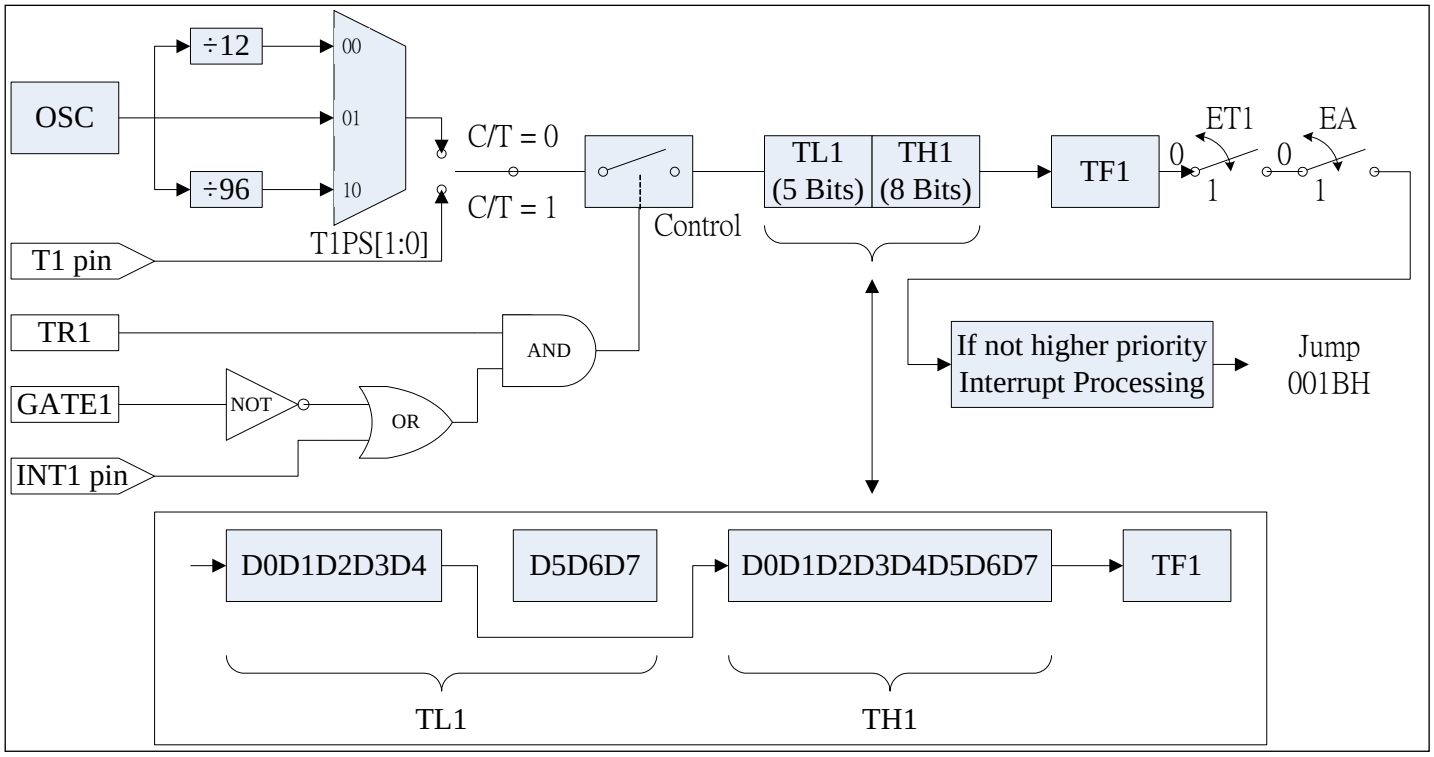


Fig. 7-1: Mode 0 -13 bit Timer / counter operation

7.7 Mode 1(16-bit Counter/Timer)

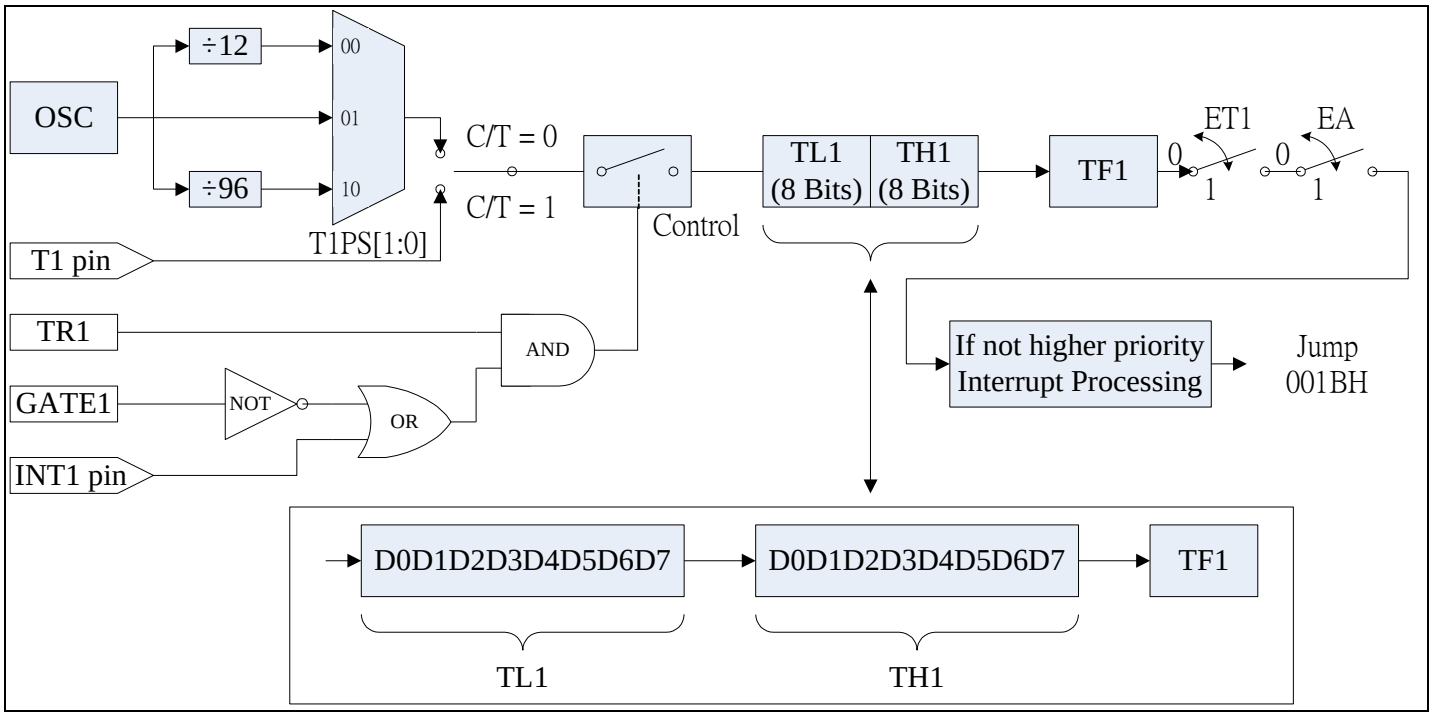


Fig. 7-2: Mode 1 16 bit Counter/Timer operation

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

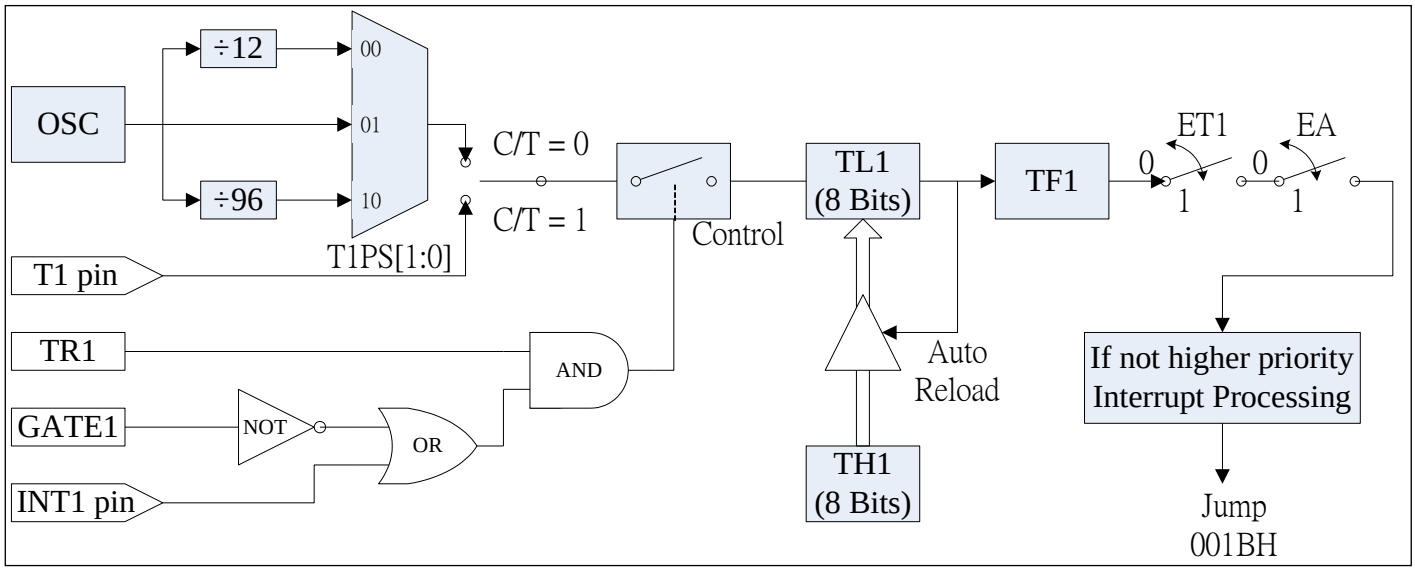


Fig. 7-3: Mode 2 8-bit auto-reload Counter/Timer operation.

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

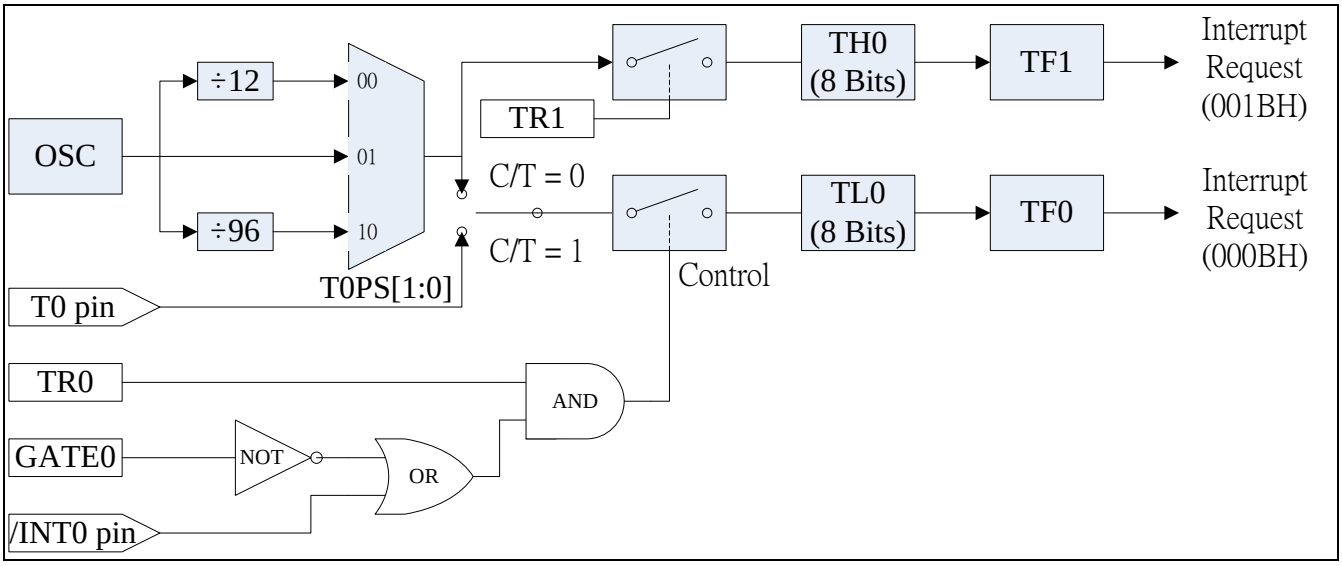


Fig. 7-4: Mode 3 Timer 0 acts as two independent 8 bit Timers / Counters operatin

8. 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).

Timer 2 and capture compare module features:

- The timer 2 is 16-bit timer / counter.
- 4-channel 16-bit compare / capture / reload functions.
- Comparator out can be CCU input source internally.
- Noise filter with CCU input.

The timer 2 interrupt vector is 2Bh.

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											
AUX2	Auxiliary 2 register	92h	-	CCU 2Source	CCU 1Source	-	CCUINF[1:0]		P42CC [1:0]		00H
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

8.1 Auxiliary 2 Register(AUX2)

Mnemonic: AUX2 **Address: 92h**

7	6	5	4	3	2	1	0	Reset
	CCU2 Source	CCU1 Source	-	CCUINF[1:0]		P42CC [1:0]		00H

The following Fig. 8-1 is set CCU action

CCU2 Capture input source 2

CCU2 = 0 - external Pin to be CCU2 capture input source

CCU2 = 1 - analog comparator 1 output to be CCU2 capture input source

CCU1 Capture input source 1

CCU1= 0 - external Pin to be CCU1 capture input source

CCU1= 1 - analog comparator 0 output to be CCU1 capture input source

CCUINF[1:0] CCU capture input Noise Filter(CCU1,CCU2)

CCUINF[1:0] = 00 - 1 consecutive same value recognize as valid data.

CCUINF[1:0] = 01 - 2 consecutive same value recognize as valid data.

CCUINF[1:0] = 10 - 4 consecutive same value recognize as valid data.

CCUINF[1:0] = 11 - 8 consecutive same value recognize as valid data.

P42CC [1:0] Capture/Compare port select function.

00: Capture/Compare function on Port1.

01: Capture/Compare function on Port2

10: Capture/Compare function on Port4 (The TQFP 64L Package Only)

11: reserved

Note: External pin CC0 and CC3 only capture input source.

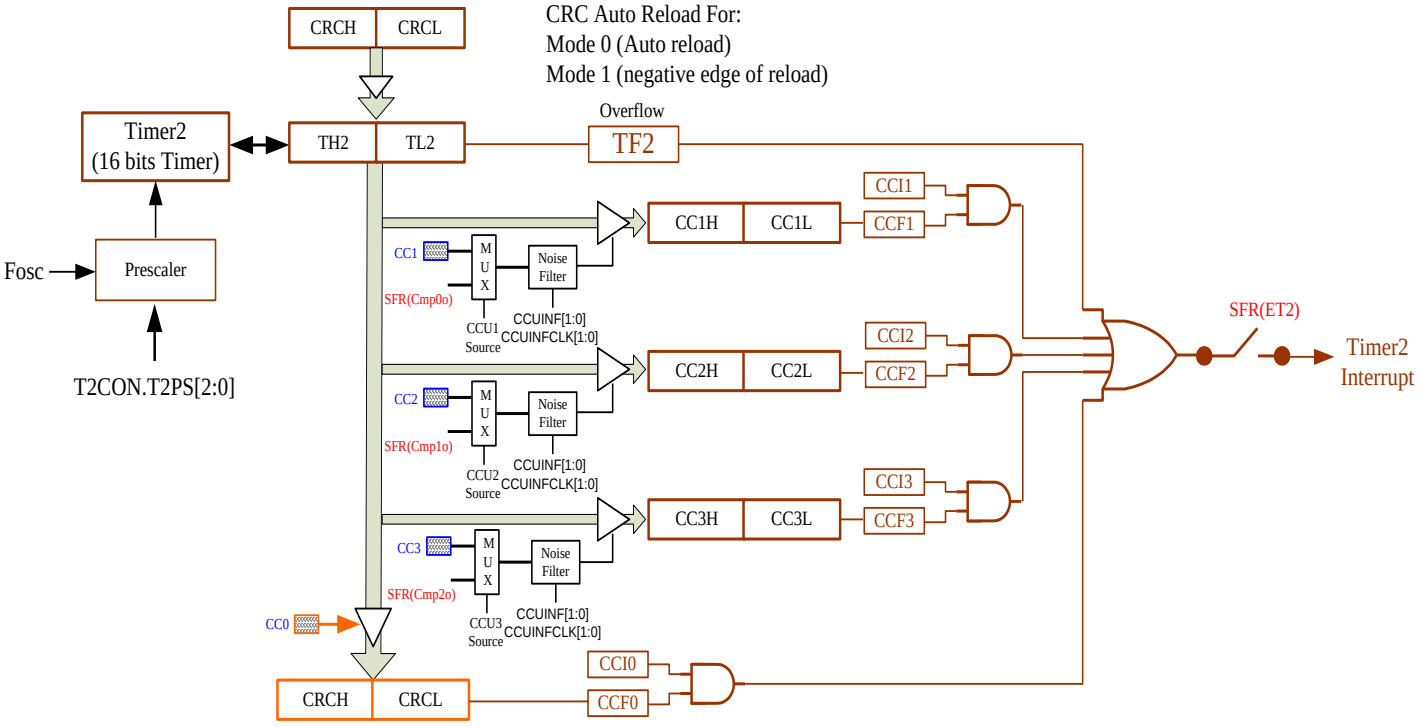


Fig. 8-1: CCU action diagram

8.2 Timer 2 Control Register(T2CON)

Mnemonic: T2CON **Address: C8h**

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

T2PS[2:0]: Prescaler select bit:

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

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

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

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

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

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

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

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

T2R[1:0] = 0X – Reload disabled

T2R[1:0] = 10 – Mode 0: Auto Reload

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

T2I[1:0]: Timer 2 input selection

T2I[1:0] = 00 – Timer 2 stop

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

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

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

8.3 Compare/Capture Control Register(CCCON)

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.

CCI3 = 1 is enable.

CCI2: Compare/Capture 2 interrupt control bit.

CCI3 = 1 is enable.

CCI1: Compare/Capture 1 interrupt control bit.

CCI3 = 1 is enable.

CCI0: Compare/Capture 0 interrupt control bit.

CCI3 = 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.

8.4 Compare/Capture Enable Register(CCEN)

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

8.5 Compare/Capture Enable 2 Register(CCEN2)

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

8.6 Timer 2 Register(TL2, TH2)

Mnemonic: TL2 **Address: CCh**

7	6	5	4	3	2	1	0	Reset
TL2[7:0]								00H

Mnemonic: TH2 **Address: CDh**

7	6	5	4	3	2	1	0	Reset
TH2[7:0]								00H

8.7 Compare/Reload/Capture Registers(CRCL, CRCH)

Mnemonic: CRCL **Address: CAh**

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

Mnemonic: CRCH **Address: CBh**

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

8.8 Compare/Capture Register 1(CCL1, CCH1)

Mnemonic: CCL1 **Address: C2h**

7	6	5	4	3	2	1	0	Reset
CCL1[7:0]								00H

Mnemonic: CCH1 **Address: C3h**

7	6	5	4	3	2	1	0	Reset
CCH1[7:0]								00H

8.9 Compare/Capture Register 2(CCL2, CCH2)

Mnemonic: CCL2 **Address: C4h**

7	6	5	4	3	2	1	0	Reset
CCL2[7:0]								00H

Mnemonic: CCH2 **Address: C5h**

7	6	5	4	3	2	1	0	Reset
CCH2[7:0]								00H

8.10 Compare/Capture Register 3(CCL3, CCH3)

Mnemonic: CCL3 **Address: C6h**

7	6	5	4	3	2	1	0	Reset
CCL3[7:0]								00H

Mnemonic: CCH3 **Address: C7h**

7	6	5	4	3	2	1	0	Reset

CCH3[7:0]	00H

8.11 Timer 2 Function

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

8.11.1 Timer Mode

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. As shown in Fig. 8-2

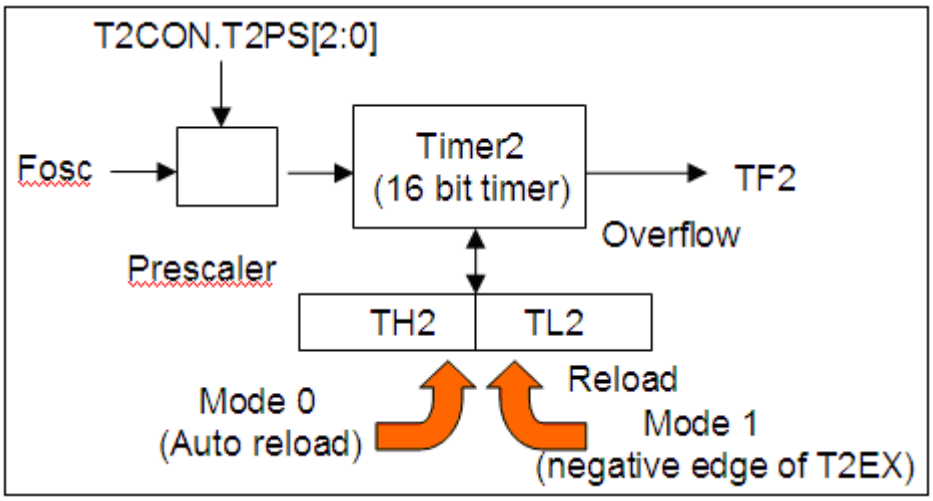


Fig. 8-2: Timer mode and Reload mode function

8.11.2 Event Counter Mode

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. As shown in Fig. 8-3

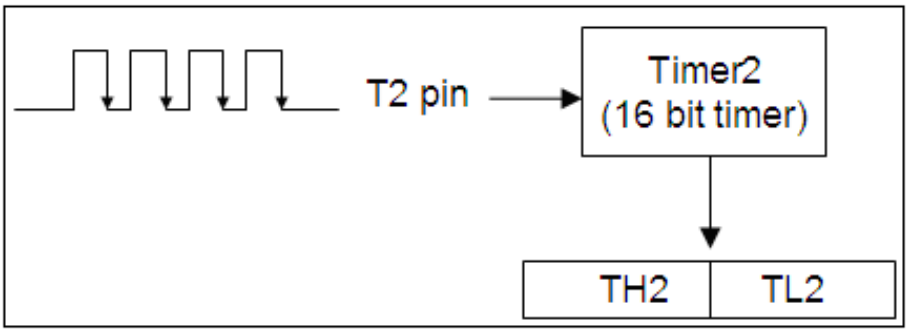


Fig. 8-3: Event counter mode function

8.11.3 Gated Timer Mode

In this mode, the internal clock which incremented timer 2 is gated by external signal T2. As shown in Fig. 8-4

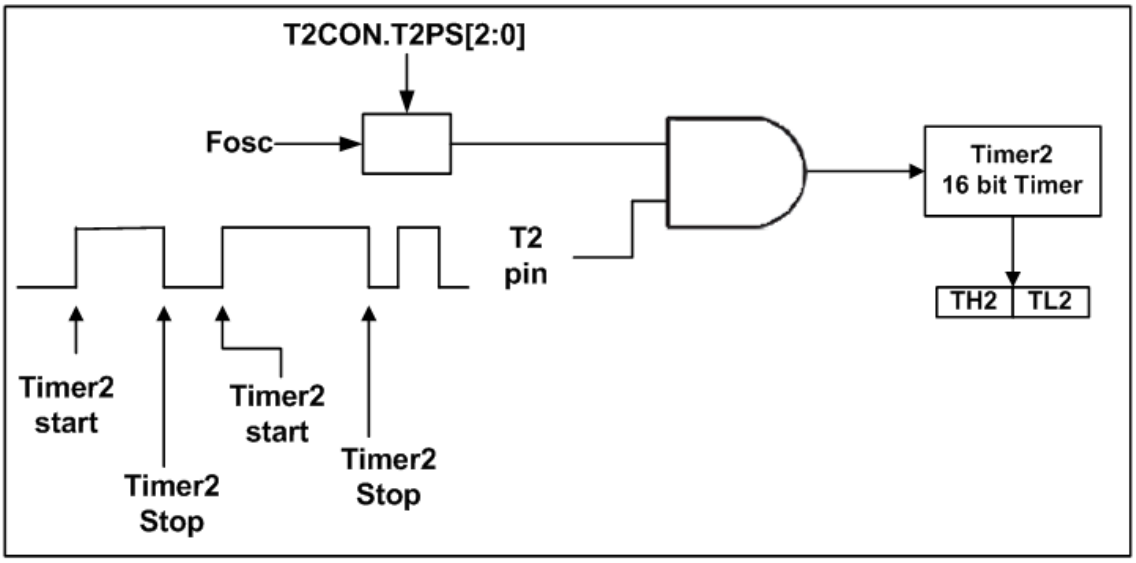


Fig. 8-4: Gated timer mode function

8.11.4 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.

8.12 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.

8.12.1 Compare Mode 0

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. As shown in Fig. 8-5 illustrates the function of compare mode 0.

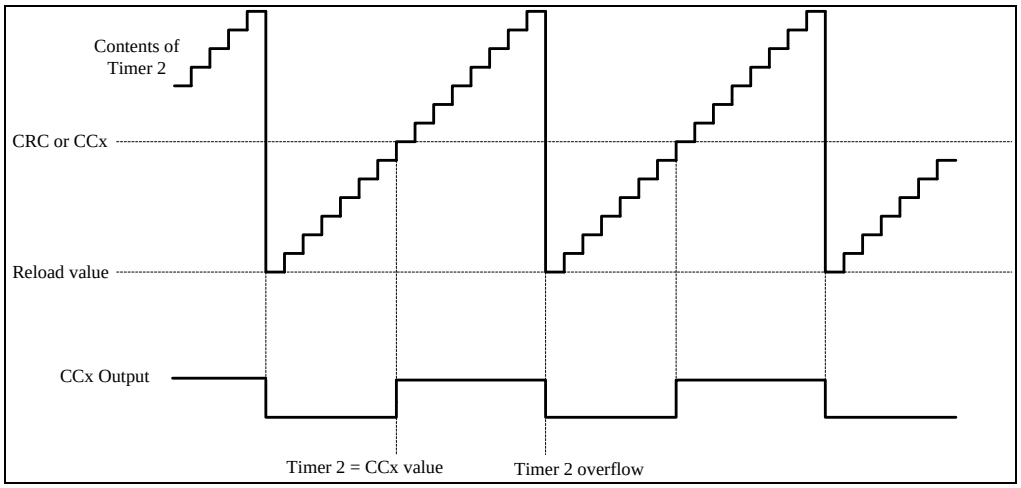


Fig. 8-5: Compare mode 0 function

8.12.2 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. As shown in Fig. 8-6 and Fig. 8-7 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.

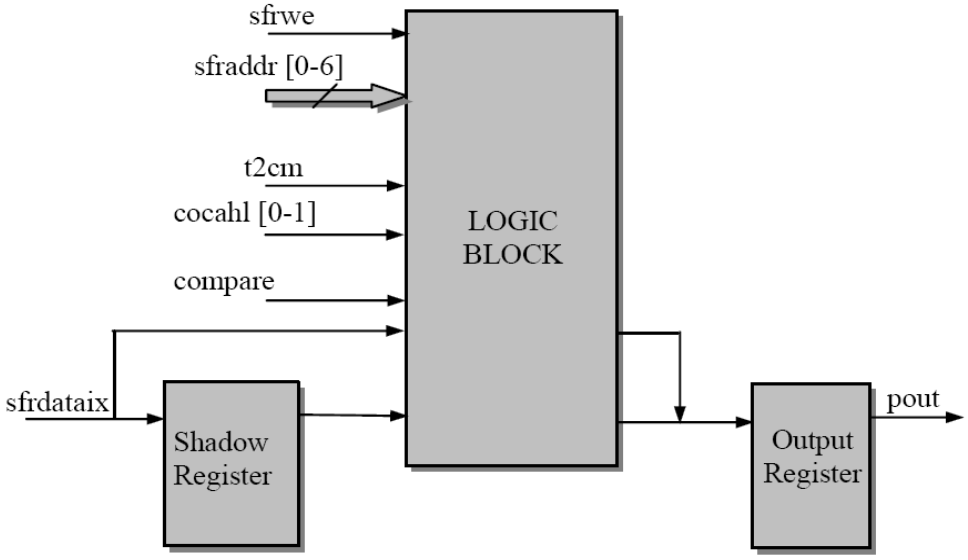


Fig. 8-6: Mode 1 Register/Port Function

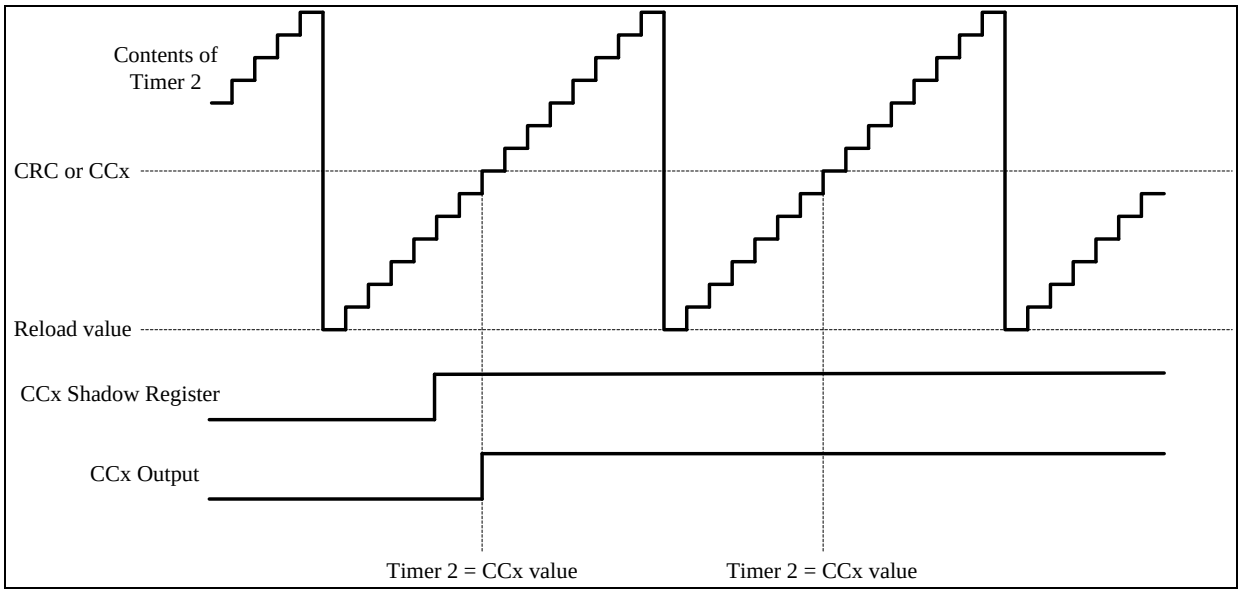


Fig. 8-7: Compare mode 1 function

8.13 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).

8.13.1 Capture Mode 0 (by Hardware)

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. As shown in Fig. 8-8

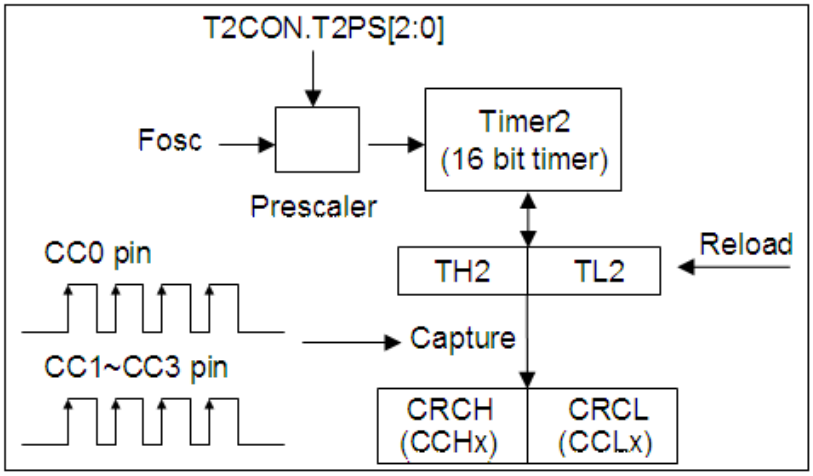


Fig. 8-8: Capture mode 0 function

8.13.2 Capture Mode 1(by Software)

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. As shown in Fig. 8-9

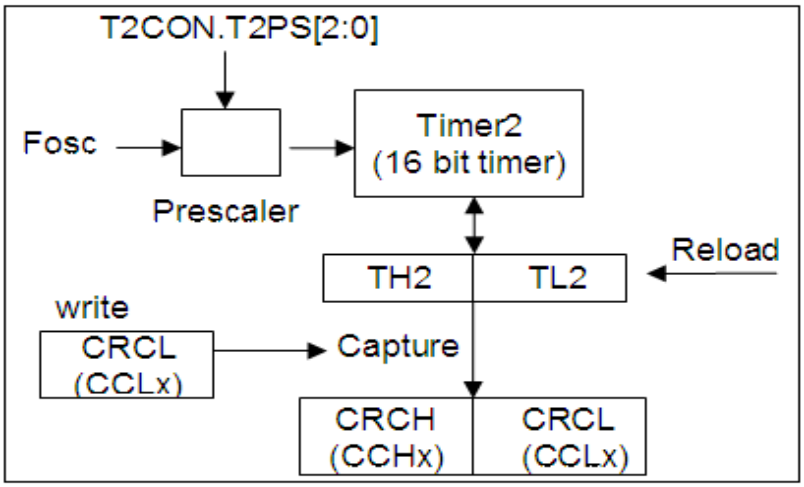


Fig. 8-9: Capture mode 1 function

9. Serial Interface 0 and 1

There are two serial interfaces for data communication in SM59A16U1, they are the so called UART0 and UART1.

As the conventional UART, the communication speed can be selected by configuring the baud rate in SFRs.

These two serial buffers consists of two separate registers, a transmit buffer and a receive buffer. Writing data to the SFR S0BUF or S1BUF sets this data in serial output buffer and starts the transmission. Reading from the S0BUF or S1BUF 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 second byte before the transmission of the first byte is completed.

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
The relevant registers of Serial interface 0 and 1											
PCON	Power control	87H	SMOD	MDUF	-				STOP	IDLE	40H
AUX	Auxiliary register	91h	BRGS	-	P4SPI	P4UR1	P4IIC	P0KBI	-	DPS	00H
PFCON	Peripheral Frequency control register	D9h	-		SRELPS[1:0]		T1PS[1:0]		T0PS[1:0]		00H
Serial interface 0 and 1											
S0CON	Serial Port 0 control register	98H	SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0	00H
S0RELL	Serial Port 0 reload register low byte	AAH	S0REL[7:0]								00H
S0RELH	Serial Port 0 reload register high byte	BAH	-						S0REL[9:8]		00H
S0BUF	Serial Port 0 data buffer	99H	S0BUF[7:0]								00H
S1CON	Serial Port 1 control register	9BH	SM	-	SM21	REN1	TB81	RB81	TI1	RI1	00H
S1RELL	Serial Port 1 reload register low byte	9DH	S1REL[7:0]								00H
S1RELH	Serial Port 1 reload register high byte	BBH	-						S1REL[9:8]		00H
S1BUF	Serial Port 1 data buffer	9CH	S1BUF[7:0]								00H

9.1 Serial Port 0 Control Register(S0CON)

Mnemonic: S0CON **Address: 98h**

7	6	5	4	3	2	1	0	Reset
SM0	SM1	SM20	REN0	TB80	RB80	TI0	RI0	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 UART0, Mode 0 ~ 3, are explained later.

SM20: Enables multiprocessor communication feature

REN0: If set, enables serial reception. Cleared by software to disable reception.

TB80: 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.

RB80: In modes 2 and 3, it is the 9th data bit received. In mode 1, if SM20 is 0, RB80 is the stop bit. In mode 0, this bit is not used. Must be cleared by software.

TI0: Transmit interrupt flag, set by hardware after completion of a serial transfer. Must be cleared by software.

RI0: Receive interrupt flag, set by hardware after completion of a serial reception. Must be cleared by software.

9.2 Serial Port 0 Reload Register(S0RELL, S0RELH)

Mnemonic: S0RELL **Address: AAh**

7	6	5	4	3	2	1	0	Reset
S0REL[7:0]								00h

Mnemonic: S0RELH **Address: BAh**

7	6	5	4	3	2	1	0	Reset
-						S0REL[9:8]		00h

9.3 Serial Port 0 Data Buffer(S0BUF)

Mnemonic: S0BUF **Address: 99h**

7	6	5	4	3	2	1	0	Reset
S0BUF[7:0]								00h

9.4 Serial Port 1 Control Register(S1CON)

Mnemonic: S1CON **Address: 9Bh**

7	6	5	4	3	2	1	0	Reset
SM	-	SM21	REN1	TB81	RB81	TI1	RI1	00h

SM: Serial Port 1 mode select.

SM	Mode
0	A
1	B

The 2 modes in UART1, Mode A and Mode B, are explained later.

SM21: Enables multiprocessor communication feature.

REN1: If set, enables serial reception. Cleared by software to disable reception.

TB81: The 9th transmitted data bit in mode A. Set or cleared by the CPU depending on the function it performs such as parity check, multiprocessor communication etc.

RB81: In mode A, it is the 9th data bit received. In mode B, if SM21 is 0, RB81 is the stop bit. Must be cleared by software.

TI1: Transmit interrupt flag, set by hardware after completion of a serial transfer. Must be cleared by software.

RI1: Receive interrupt flag, set by hardware after completion of a serial reception. Must be cleared by software.

9.5 Serial Port 1 Reload Register(S1RELL, S1RELH)

Mnemonic: S1RELL **Address: 9Dh**

7	6	5	4	3	2	1	0	Reset
S1REL[7:0]								00h

Mnemonic: S1RELH **Address: BBh**

7	6	5	4	3	2	1	0	Reset
-						S1REL[9:8]		00h

9.6 Serial Port 1 Data Buffer(S1BUF)

Mnemonic: S0BUF **Address: 9Ch**

7	6	5	4	3	2	1	0	Reset
S1BUF[7:0]								00h

9.7 Serial Interface 0

The Serial Interface 0 can operate in the following 4 modes:

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.

9.7.1 Mode 0

Pin RXD0 serves as input and output. TXD0 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 S0CON as follows: RI0 = 0 and REN0 = 1. In the other modes, a start bit when REN0 = 1 starts receiving serial data.. As shown in Fig. 9-1 and Fig. 9-2

Fig. 9-1: Transmit mode 0 for Serial 0

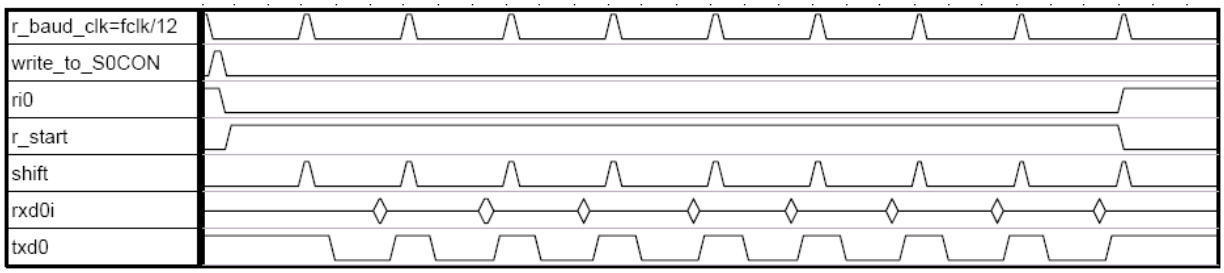


Fig. 9-2: Receive mode 0 for Serial 0

9.7.2 Mode 1

Here Pin RXD0 serves as input, and TXD0 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 S0BUF, and a stop bit sets the flag RB80 in the SFR S0CON. In mode 1, either internal baud rate generator or timer 1 can be use to specify the desired baud rate. As shown in Fig. 9-3 and Fig. 9-4

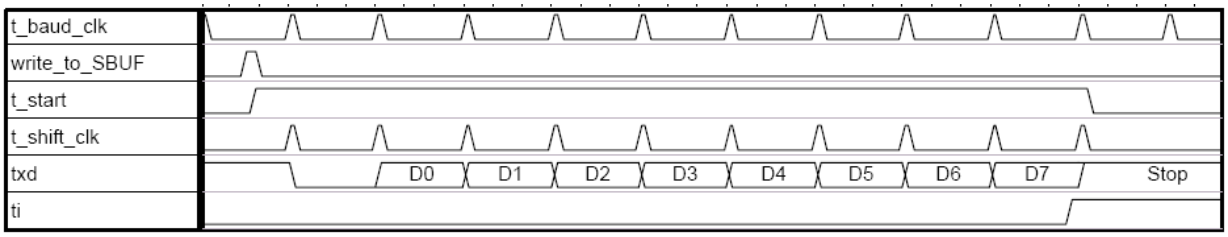


Fig. 9-3: Transmit mode 1 for Serial 0

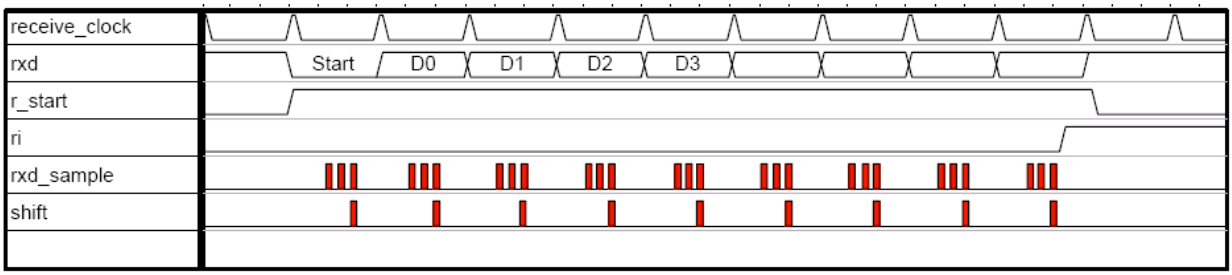


Fig. 9-4: Receive mode 1 for Serial 0

9.7.3 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 TB80 in S0CON is output as the 9th bit, and at receive, the 9th bit affects RB80 in Special Function Register S0CON.

9.7.4 Mode 3

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. As shown in Fig. 9-5 and Fig. 9-6.

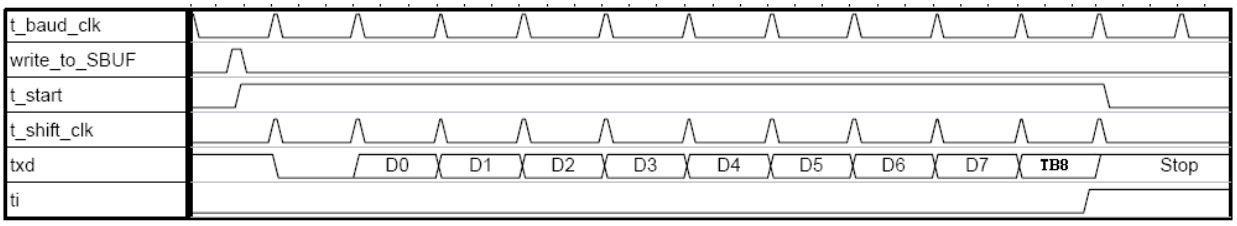


Fig. 9-5: Transmit modes 2 and 3 for Serial 0

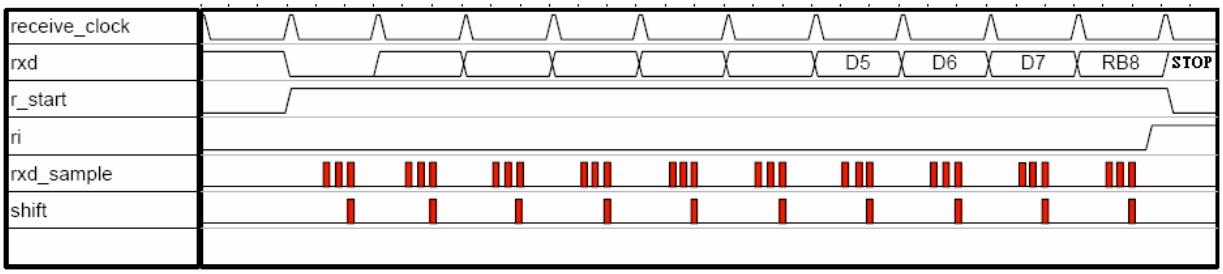


Fig. 9-6: Receive modes 2 and 3 for Serial 0

9.8 Serial Interface 1

The interrupt vector is 83h.

The Serial Interface 1 can operate in the following 2 modes:

SM	Mode	Description	Baud Rate
0	A	9-bit UART	Variable
1	B	8-bit UART	Variable

9.8.1 Mode A

This mode is similar to Mode 2 and 3 of Serial interface 0, 11 bits are transmitted or received: a start bit (0), 8 data bits (LSB first), a programmable Bit 9, and a stop bit (1). Bit 9 can be used to control the parity of the serial interface: at transmission, bit TB81 in S1CON is outputted as Bit 9, and at receive, Bit 9 affects RB81 in SFR S1CON. As shown in Fig. 9-7 and Fig. 9-8.

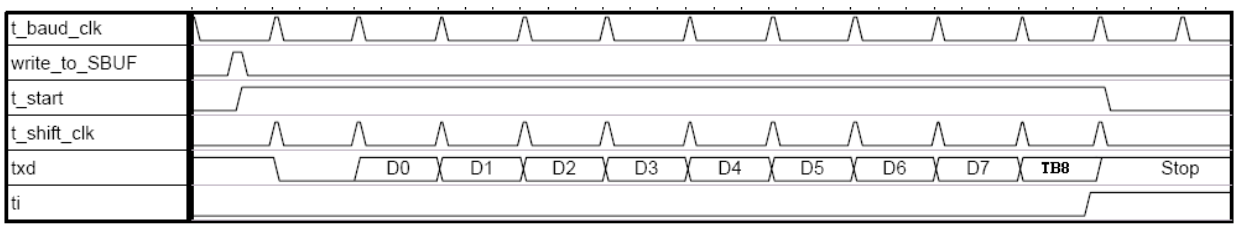


Fig. 9-7: Transmit mode A for Serial 1

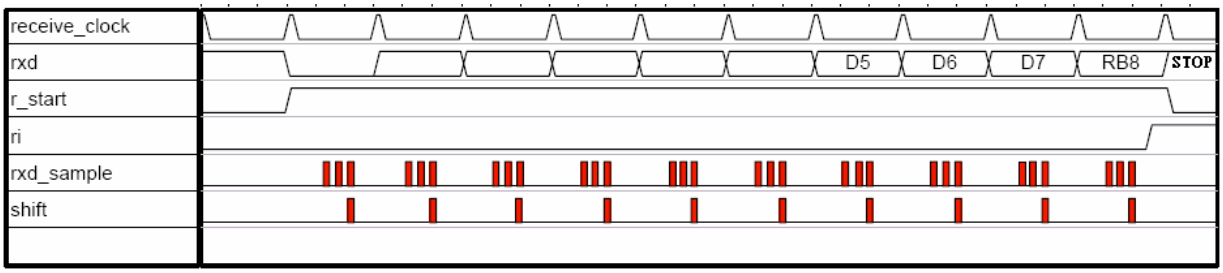


Fig. 9-8: Receive mode A for Serial 1

9.8.2 Mode B

This mode is similar to Mode 1 of Serial interface 0. Pin RXD1 serves as input, and TXD1 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 S1BUF, and stop bit sets the flag RB81 in the SFR S1CON. In mode B, internal baud rate generator is use to specify the baud rate. As shown in Fig. 9-9 and Fig. 9-10.

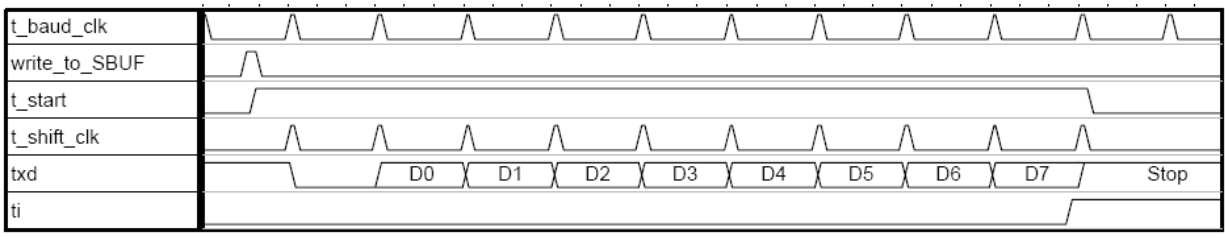


Fig. 9-9: Transmit mode B for Serial 1

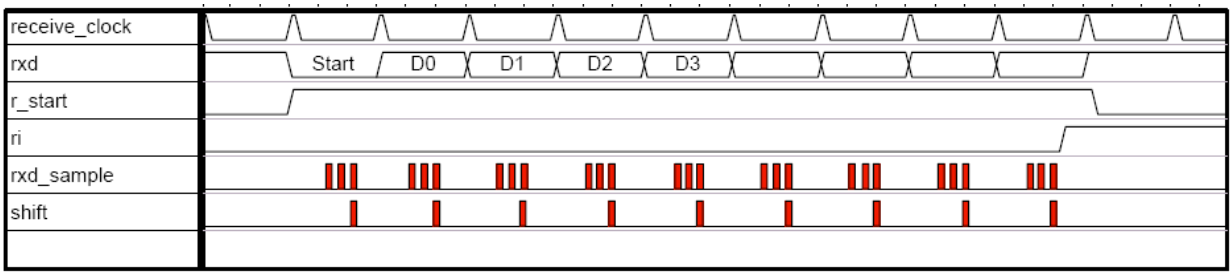


Fig. 9-10: Receive mode B for Serial 1

9.9 Multiprocessor communication of Serial Interface 0 and 1

The feature of receiving 9 bits in Modes 2 and 3 of Serial Interface 0 or in Mode A of Serial Interface 1 can be used for multiprocessor communication. In this case, the slave processors have bit SM20 in S0CON or SM21 in S1CON set to 1. When the master processor outputs slave's address, it sets the Bit 9 to 1, causing a serial port receive interrupt in all the slaves. The slave processors compare the received byte with their network address. If matched, the addressed slave will clear SM20 or SM21 and receive the rest of the message, while other slaves will leave SM20 or SM21 bit unaffected and ignore this message. After addressing the slave, the host will output the rest of the message with the Bit

9 set to 0, so no serial port receive interrupt will be generated in unselected slaves.

9.10 Baud Rate Generator

9.10.1 Serial Interface 0 modes 1 and 3

Mnemonic: PFCON **Address: D9h**

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

SRELPS[1:0] SREL Prescaler Select

SRELPS[1:0]	Prescaler
00	Fosc/64
01	Fosc/32
10	Fosc/16
11	Fosc/8

T1PS[1:0]: Timer1 Prescaler Select

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

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

(1) T1PS[1:0] is 00

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

(2) T1PS[1:0] is 01

$$\text{Baud Rate} = \frac{2^{\text{SMOD}} \times F_{OSC}}{32 \times (256 - TPH1)}$$

(3) T1PS[1:0] is 10

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

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

(1) SRELPS[1:0] is 00

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

(2) SRELPS[1:0] is 01

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

(3) SRELPS[1:0] is 10

$$\text{Baud Rate} = \frac{2^{\text{SMOD}} \times F_{osc}}{16 \times \left(2^{10} - \text{SREL}\right)}$$

(4) SRELPS[1:0] is 11

$$\text{Baud Rate} = \frac{2^{\text{SMOD}} \times F_{osc}}{8 \times \left(2^{10} - \text{SREL}\right)}$$

9.10.2 Serial Interface 1 modes A and B

$$\text{Baud Rate} = \frac{F_{OSC}}{32 \times \left(2^{10} - \text{S1REL}\right)}$$

9.11 Clock Source for baud rate

The on-chip RC-Oscillator frequency varies within ±5% after factory calibration. In case of application with higher clock precision requirement, external Crystal is usually recommended clock source.

10. Watchdog timer

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

The watchdog timer has a free running on-chip RC oscillator (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 WDTE bit of WDTC register. The default WDT time-out period is approximately 178.0ms (WDTM [3:0] = 0100b).

The WDT has selectable divider input for the time base source clock. To select the divider input, the setting of bit3 ~ bit0 (WDTM [3:0]) of Watch Dog Timer Control Register (WDTC) should be set accordingly. As shown in Table 10-1.

$$WDTCLK = \frac{23\text{KHz}}{2^{\text{WDTM}}}$$

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

Table 10-1: WDT time-out period

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

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

When MCU is reset, the MCU will be read WDTEN control bit status. When WDTEN bit is set to 1, the watchdog function will be disabled no matter what the WDTE bit status is. When WDTEN bit is clear to 0, the watchdog function will be enabled if WDTE bit is set to 1 by program. User can to set WDTEN on the writer or ISP.

The program can enable the WDT function by programming 1 to the WDTE bit premise that WDTEN control bit is clear to 0. After WDTE set to 1, the 8 bit-counter starts to count with the selected time base source clock which set by WDTM [3:0]. It will generate a reset signal when overflows. The WDTE bit will be cleared to 0 automatically when MCU been reset, either hardware reset or WDT reset. As shown in Fig. 10-1.

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

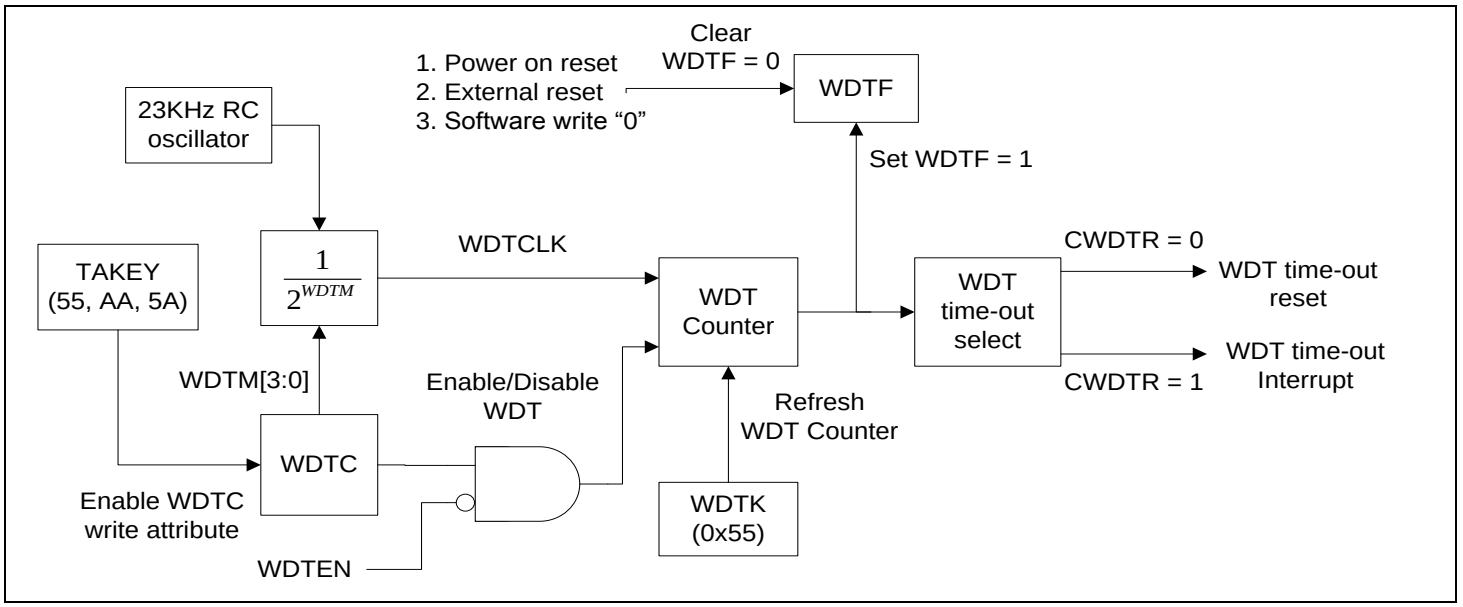


Fig. 10-1: Watchdog timer block diagram

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
WDTC	Watchdog timer control register	B6h	-	CWDTR	WDTE	-	WDTM [3:0]				04H
WDTK	Watchdog timer refresh key	B7h	WDTK[7:0]								00H

Mnemonic: TAKEY **Address: F7h**

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

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

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

10.1 Watchdog Timer Control Register(WDTC)

Mnemonic: WDTC **Address: B6h**

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

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

CWDTR = 0 - Enable watch dog reset.

CWDTR = 1 - Enable watch dog interrupt.

WDTE: Control bit used to enable Watchdog timer.
The WDTE bit can be used only if WDTEN is "0". If the WDTEN bit is "0", then WDT can be disabled / enabled by the WDTE bit.

WDTE = 0 - Disable WDT.

WDTE = 1 - Enable WDT.
The WDTE bit is not used if WDTEN is "1". That is, if the WDTEN bit is "1", WDT is always disabled no matter what the WDTE bit status is. The WDTE bit can be read and written.

WDTM [3:0]: WDT clock source divider bit. Please see Table 10-1 to reference the WDT time-out period.

10.2 Watchdog Timer Refresh Register(WDTK)

Mnemonic: WDTK **Address: B7h**

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

WDTK[7:0] : Watchdog timer refresh key.

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

For example, if enable WDT and select time-out reset period is 2.8493s.

First, programming the information block OP3 bit7 WDTEN to “0”.

Secondly,

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

For example 2, if enable WDT and select time-out Interrupt period is 178.0ms.
First, programming the information block OP3 bit7 WDTEN to “0”.
Secondly,

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

11. Interrupt

The SM59A16U1 provides 14 interrupt sources with four priority levels. Each source has its own request flag(s) located in a special function register. Each interrupt requested by the corresponding flag could individually be enabled or disabled by the enable bits in SFR's IEN0, IEN1, and IEN2.

When the interrupt occurs, the engine will vector to the predetermined address as given in Table 11-1. 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 11-1: 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	RI0/TI 0– Serial channel 0 interrupt	0023h	4
6	TF2/EXF2 – Timer 2 interrupt	002Bh	5
7	PWMIF – PWM interrupt	0043h	8
8	SPIIF – SPI interrupt	004Bh	9
9	ADCIF – A/D converter interrupt	0053h	10
10	KBIIF – keyboard Interface interrupt	005Bh	11
11	LVIIF – Low Voltage Interrupt	0063h	12
12	IICIF – IIC interrupt	006Bh	13
13	USB interrupt	0073h	14
14	USBRSM interrupt	007Bh	15
15	RI1/TI1 – Serial channel 1 interrupt	0083h	16
16	WDT – Watchdog interrupt	008Bh	17
17	Comparator interrupt	0093h	18

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

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
Interrupt											
IEN0	Interrupt Enable 0 register	A8H	EA	-	ET2	ES0	ET1	EX1	ET0	EX0	00H
IEN1	Interrupt Enable 1 register	B8H	B8H	EXEN 2	-	IEIIC	IELVI	IEKBI	IEADC	IESPI	00H
IEN2	Interrupt Enable 2 register	9AH	-	-	-	-	-	ECmpI	EWDT	ES1	00H
IRCON	Interrupt request register	C0H	EXF2	TF2	IICIF	LVIIF	KBIIF	ADCIF	SPIIF	PWMI F	00H
IRCON2	Interrupt request register 2	97H	-	-	-	-	-	CmpIF	WDT IF	-	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

11.1 Interrupt Enable 0 Register(IEN0)

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.

ES0: ES=0 – Disable Serial channel 0 interrupt.
ES=1 – Enable Serial channel 0 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.

11.2 Interrupt Enable 1 Register(IEN1)

Mnemonic: IEN1 **Address: B8h**

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

EXEN2: Timer 2 reload interrupt enable.

EXEN2 = 0 – Disable Timer 2 external reload interrupt.

EXEN2 = 1 – Enable Timer 2 external reload interrupt.

IEIIC: IIC interrupt enable.

IEIIC = 0 – Disable IIC interrupt.

IEIIC = 1 – Enable IIC interrupt.

IELVI: LVI interrupt enable.

IELVI = 0 – Disable LVI interrupt.

IELVI = 1 – Enable LVI interrupt.

IEKBI KBI interrupt enable

EKBI = 0 – Disable KBI interrupt.

IEKBI = 1 – Enable KBI interrupt.

IEADC: A/D converter interrupt enable

IEADC = 0 – Disable ADC interrupt.

IEADC = 1 – Enable ADC interrupt.

IESPI: SPI interrupt enable.

IESPI = 0 – Disable SPI interrupt.

IESPI = 1 – Enable SPI interrupt.

11.3 Interrupt Enable 2 Register(IEN2)

Mnemonic: IEN2 **Address: 9Ah**

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

ECmpI Enable Comparator interrupt(include comparator_0 and comparator_1).

ECmpI = 0 – Disable Comparator interrupt.

ECmpI = 1 – Enable Comparator interrupt.

EWDT: Enable Watch dog interrupt.

EWDT = 0 – Disable Watchdog interrupt.

EWDT = 1 – Enable Watchdog interrupt.

ES1: ES1=0 – Disable Serial channel 1 interrupt.

ES1=1 – Enable Serial channel 1 interrupt.

11.4 Interrupt Request Register(IRCON)

Mnemonic: IRCON **Address: C0h**

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

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

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

IICIF: IIC interrupt flag.

LVIIF: LVI interrupt flag.

KBIIF: KBI interrupt flag.

ADCIF: A/D converter end interrupt flag.

SPIIF: SPI interrupt flag.

PWMIF: PWM interrupt flag.

11.5 Interrupt Request Register 2(IRCON2)

Mnemonic:IRCON2 **Address: 97h**

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

CmpIF Comparator interrupt flag

HW will clear this flag automatically when enter interrupt vector.

SW can clear this flag also.(in case analog comparator INT disable)

WDTIF: Watch dog interrupt flag

11.6 Priority Level Structure

All interrupt sources are combined in groups, As given in Table 11-2.

Table 11-2: Priority level groups

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

Each group of interrupt sources can be programmed individually to one of four priority levels by setting or clearing one bit in the special function register ip0 and one in ip1. If requests of the same priority level will be received simultaneously, an internal polling sequence determines which request is serviced first. As given in Table 11-3 and Table 11-4 and Table 11-5.

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 11-3: Priority levels

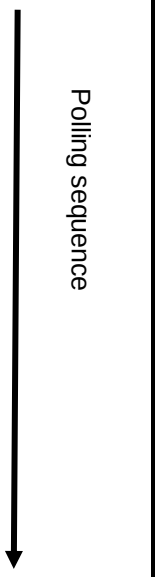
IP1.x	**IP0.x**	**Priority Level**
0	0	Level0 (lowest)
0	1	Level1
1	0	Level2

1	1	Level3 (highest)

Table 11-4: Groups of priority

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

Table 11-5: Polling sequence

Interrupt source	**Sequence**
External interrupt 0	
Serial channel 1 interrupt	
PWM interrupt	
Timer 0 interrupt	
Watchdog interrupt	
SPI interrupt	
External interrupt 1	
Comparator interrupt	
ADC interrupt	
Timer 1 interrupt	
USB interrupt	
KBI interrupt	
Serial channel 0 interrupt	
USBRSM interrupt	
LVI interrupt	
Timer 2 interrupt	
IIC interrupt	

12. 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	MDUF	-				STOP	IDLE	40h

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

12.1 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.

12.2 Stop Mode

Setting the STOP bit of PCON register invokes the STOP mode. All internal clocking in this mode is turn off. The CPU will exit this state only if interrupts asserted from external INT0/1, KBI, LVI, and USB, OPA Comparator or hardware reset by WDT and LVR

Internally generated interrupts (timer, serial port ...) have no effect on stop mode since they require clocking activity.

13. Pulse Width Modulation (PWM)

PWM module features:

- Eight-channel (four-pair) PWM output pins.
- 14-bit resolution.
- Center and Edge Alignment output mode.
- Dead time generator.
- PWM and Special Event Interrupt Trigger.
- Output Override Function for motor control.
- Overdrive current protect for the fault (FLTA and FLTB)

There under is the working module of the PWM, As shown in Fig. 13-1

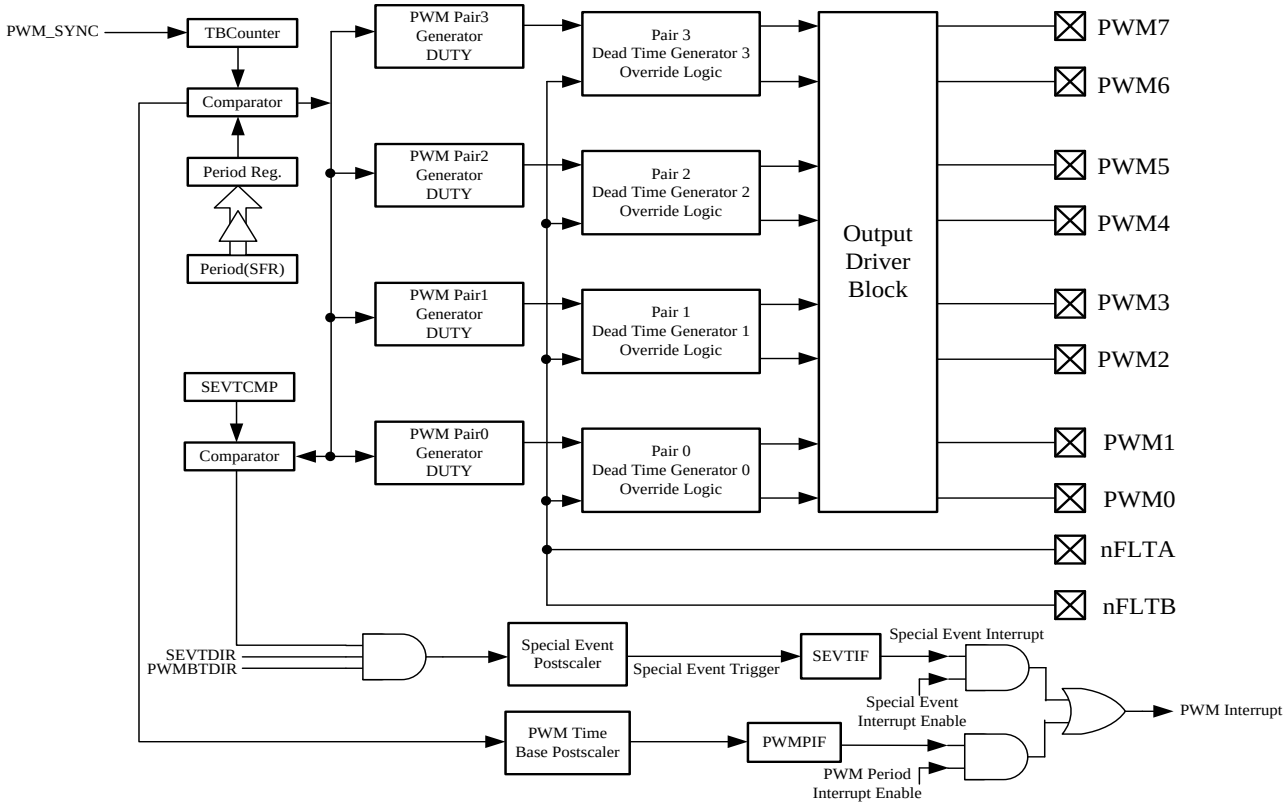


Fig. 13-1: Working module of the PWM

The interrupt vector is 43h.

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
The relevant registers of the PWM function											
PWMADDR	PWM Address Register	A2h	PWMADDR[7:0]								00H
PWMDATA	PWM Data Register	A3h	PWMDATA[7:0]								00H
ADCC2	ADC Control 2 Register	ACh	Start	Adjust	PWM TriggerEN	EXT TriggerEN	ADC MODE	ADCCH[2:0]			00H
PWM											
PWMTBC0	PWM Time Base Control 0 Register	F9h	-				PWMTBPRE[1:0]		PWMTBMOD[1:0]		00H
PWMTBC1	PWM Time Base Control 1 Register	FAh	PWMTBEN	-					SEVTIE	PWMPIE	00H
PWMOPMOD	PWM Output Pair Mode Register	FBh	-				PWMOP3MOD	PWMOP2MOD	PWMOP1MOD	PWMOP0MOD	00H
TBCOUNTERL	Time Base Counter (Low)	FCh	Time Base Counter Low 8 bit								00H
TBCOUNTERH	Time Base Counter (High)	FDh	-		Time Base Counter High 6 bit						00H
PERIODL	PWM Period (Low) Register	F1h	PWM Period Low 8 bit								FFH
PERIODH	PWM Period(High) Register	F2h	-		PWM Period High 6 bit						3FH
SEVTCMPL	Special Event Compare Low Register	F3h	Special Event Compare Low 8 bit								FFH
SEVTCMPH	Special Event Compare High Register	F4h	-		Special Event Compare High 6 bit						3FH
PWMEN	PWM Output Enable Register	F5h	PWM7EN	PWM6EN	PWM5EN	PWM4EN	PWM3EN	PWM2EN	PWM1EN	PWM0EN	00H
PWMSEV	PWM Special Event Register	EDh	SEVPOST[3:0]				SEVTDIR	-	UDIS	OSYNC	00H
PWMTBPOSTSCALE	PWM Time Base Post Scale Register	EEh	PWMTBPOST[7:0]								00H
PWMINTF	PWM Interrupt Flag Register	BCh	PWMTBDIR	-					SEVTIF	PWMPIF	00H
DEADTIME0	Dead Time 0 Register	E9h	DT0PRE[1:0]		DT0[5:0]						00H
DEADTIME1	Dead Time 1 Register	EAh	DT1PRE[1:0]		DT1[5:0]						00H
DEADTIME2	Dead Time 2 Register	EBh	DT2PRE[1:0]		DT2[5:0]						00H

DEADTIME3	Dead Time 3 Register	ECh	DT3PRE[1:0]		DT3[5:0]					00H
FLTCONFIG	Fault Configure Register	DBh	BRF EN	FLTB S	FLTB MOD	FLTB EN	FLT CON	FLTA S	FLTA MOD	FLTA EN	80H
FLTNF	Fault Noise Filter Register	DCh	-		FLTB LS	FLTA LS	FLTB NF	FLTA NF	-		00H
PWMPOLARITY	PWM Polarity Register	DDh	Polarity 7	Polarity 6	Polarity 5	Polarity 4	Polarity 3	Polarity2	Polarity1	Polarity 0	FFH
OVRIDEDIS	Override Disable Register	DEh	OV7 DIS	OV6 DIS	OV5 DIS	OV4 DIS	OV3 DIS	OV2 DIS	OV1 DIS	OV0 DIS	FFH
OVRIDEDATA	Override Data Register	DFh	OV7 DATA	OV6 DATA	OV5 DATA	OV4 DATA	OV3 DATA	OV2 DATA	OV1 DATA	OV0 DATA	00H
DUTY0L	PWM 0 Duty Low byte Register	D1h	PWM Pair 0 Duty Low 8 bit								00H
DUTY0H	PWM 0 Data High byte Register	D2h	-		PWM Pair 0 Duty High 6 bit						00H
DUTY1L	PWM 1 Duty Low byte Register	D3h	PWM Pair1 Duty Low 8 bit								00H
DUTY1H	PWM 1 Data High byte Register	D4h	-		PWM Pair1 Duty High 6 bit						00H
DUTY2L	PWM 2 Duty Low byte Register	D5h	PWM Pair 2 Duty Low 8 bit								00H
DUTY2H	PWM 2 Duty High byte Register	D6h	-		PWM Pair 2 Duty High 6 bit						00H
DUTY3L	PWM 3 Duty Low byte Register	D7h	PWM Pair 3 Duty Low 8 bit								00H
DUTY3H	PWM 3 Duty High byte Register	C9h	-		PWM Pair 3 Duty High 6 bit						00H

13.1 ADC Control Register 2(ADCC2)

Mnemonic:ADCC2 **Address: ACh**

7	6	5	4	3	2	1	0	Reset
Start	Adjust	PWM TriggerEN	EXT TriggerEN	ADC MODE	ADCCH[2:0]			00H

Start When this bit is set, the ADC will be start conversion.

(SW trigger conversion)

ADJUST Adjust the format of ADC conversion DATA.

ADJUST = 0: (default value)
ADC data high byte ADCD [9:2] = ADCDH [7:0].

ADC data low byte ADCD [1:0] = ADCDL [1:0].

ADJUST = 1: ADC data high byte ADCD [9:8] = ADCDH [1:0].

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

PWMTriggerEN PWM trigger ADC to start conversion.

(HW internal trigger conversion)

0 = disable

1 = enable

EXTTriggerEN External Pin trigger ADC to start conversion.

(HW external trigger conversion)

0 = disable

1 = enable

ADCMODE 0 = continuous mode

1 = single-shot mode

ADCCH[2:0] ADC channel select.

ADCCH [2:0]	Channel
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

13.2 PWM Time Base Control 0(PWMTBC0)

Mnemonic: PWMTBC0 **Address: F9h**

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

PWMTBPRE PWM Time Base PreScale

[1:0]

PWMTBPRE [1:0]	Mode
00	Fosc
01	Fosc/4
10	Fosc/16
11	Fosc/64

PWMTBMOD PWM Time Base Mode

PWMTBMOD [1:0] = 00 - Free Running mode

PWMTBMOD [1:0] = 01 - single-shot mode

PWMTBMOD [1:0] = 10 - continuous up/down counting mode

PWMTBMOD [1:0] = 11 - continuous up/down counting with interrupt for double PWM updates.

Minimum PWM Frequencies Example: Fosc= 24MHz, PERIOD = 0x3FFFh		
Prescale	PWM Frequency Edge-align	PWM Frequency center-align
1:1	1500 Hz	750 Hz
1:4	375 Hz	188 Hz
1:16	94 Hz	47 Hz
1:64	23 Hz	12 Hz

13.3 PWM Time Base Control 1(PWMTBC1)

Mnemonic: PWMTBC1 **Address: FAh**

7	6	5	4	3	2	1	0	Reset
PWMTBEN	-					SEVTIE	PWMPIE	00H

PWMTBEN PWM Time Base Enable

0 = PWM Time Base Disable

1 = PWM Time Base Enable

SEVTIE: Special Event Interrupt Enable

SEVTIE = 0 - Special Event Interrupt Disable

SEVTIE = 1 - Special Event Interrupt Enable

PWMPIE: PWM Period Interrupt Enable

PWMPIE = 0 - PWM Period Interrupt Diable

PWMPIE = 1 - PWM Period Interrupt Enable

13.4 PWM Output Pair Mode(PWMOPMOD)

Mnemonic: PWMOPMOD **Address: FBh**

7	6	5	4	3	2	1	0	Reset
-				PWMOP3MOD	PWMOP2MOD	PWMOP1MOD	PWMOP0MOD	00H

PWMOP3MOD PWM Output Pair 3 Mode

0 = (PWM6, PWM7) is complementary mode

1 = (PWM6, PWM7) is independent mode

PWMOP2MOD PWM Output Pair 2 Mode

0 = (PWM4, PWM5) is complementary mode

1 = (PWM4, PWM5) is independent mode

PWMOP1MOD PWM Output Pair 1 Mode

0 = (PWM2, PWM3) is complementary mode

1 = (PWM2, PWM3) is independent mode

PWMOP0MOD PWM Output Pair 0 Mode

0 = (PWM0, PWM1) is complementary mode

1 = (PWM0, PWM1) is independent mode

13.5 Time Base Counter by PWM clock(TBCOUNTERL, TBCOUNTERH)

Mnemonic: TBCOUNTERL **Address: FCh**

7	6	5	4	3	2	1	0	Reset
Time Base Counter Low 8 bit [7:0]								00H

Mnemonic: TBCOUNTERH **Address: FDh**

7	6	5	4	3	2	1	0	Reset
-	Time Base Counter High 6 bit [5:0]							00H

13.6 PWM Period(PERIODL, PERIODH)

Mnemonic: PERIODL **Address: F1h**

7	6	5	4	3	2	1	0	Reset
PWM Period Low 8 bit [7:0]								FFH

Mnemonic: PERIODH **Address: F2h**

7	6	5	4	3	2	1	0	Reset
-	PWM Period High 6 bit [5:0]							3FH

13.7 Special Event Compare(SEVTCMPL, SEVTCMPH)

Mnemonic: SEVTCMPL **Address: F3h**

7	6	5	4	3	2	1	0	Reset
Special Event Compare Low 8 bit [7:0]								FFH

Mnemonic: SEVTCMPH **Address: F4h**

7	6	5	4	3	2	1	0	Reset
-	Special Event Compare High 6 bit [5:0]							3FH

13.8 PWM Output Enable(PWMEN)

Mnemonic: PWMEN **Address: F5h**

7	6	5	4	3	2	1	0	Reset
PWM7EN	PWM6EN	PWM5EN	PWM4EN	PWM3EN	PWM2EN	PWM1EN	PWM0EN	00H

PWM7EN PWM 7 Enable

PWM7EN = 0 - PWM7 Output Disable

PWM7EN = 1 - PWM7 Output Enable

PWM6EN PWM 6 Enable

PWM6EN = 0 - PWM6 Output Disable

PWM6EN = 1 - PWM6 Output Enable

PWM5EN PWM 5 Enable

PWM5EN = 0 - PWM5 Output Disable

PWM5EN = 1 - PWM5 Output Enable

PWM4EN PWM 4 Enable

PWM4EN = 0 - PWM4 Output Disable

PWM4EN = 1 - PWM4 Output Enable

PWM3EN PWM 3 Enable

PWM3EN =0 - PWM3 Output Disable

PWM3EN =1 - PWM3 Output Enable

PWM2EN PWM 2 Enable

PWM2EN =0 - PWM2 Output Disable

PWM2EN =1 - PWM2 Output Enable

PWM1EN PWM 1 Enable

PWM1EN =0 - PWM1 Output Disable

PWM1EN =1 - PWM1 Output Enable

PWM0EN PWM 0 Enable

PWM0EN =0 - PWM0 Output Disable

PWM0EN =1 - PWM0 Output Enable

13.9 PWM Special Event(PWMSEV)

Mnemonic: PWMSEV **Address: EDh**

7	6	5	4	3	2	1	0	Reset
SEVPOST[3:0]				SEVTDIR	-	UDIS	OSYNC	00H

SEVPOST Special Event Postscale Set

SEVPOST [3:0] 0000 = 1:1 Postscale

SEVPOST [3:0] 0001 = 1:2 Postscale

:

:

SEVPOST [3:0] 1111 = 1:16 Postscale

SEVTDIR Special event trigger time base direction

SEVTDIR = 0 - counting upwards

SEVTDIR = 1 - counting downwards

UDIS PWM update disable.(This bit affects PERIOD, DUTY, SEVTCMP; OVRIDEDIS, OVRIDEDATA)

UDIS = 0 - update from duty cycle and period buffer are Enable

UDIS = 1 - update from duty cycle and period buffer are Disable

OSYNC PWM output override synchronization

OSYNC = 0 - Asynchronous

OSYNC = 1 - synchronous

13.10 PWM Time Base Post Scale Register(PWMTBPOSTSCALE)

Mnemonic: PWMTBPOSTSCALE **Address: EEh**

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

PWMTBPOST[7:0] 0000_0000 = 1: 1 Postscale
0000_0001 = 1: 2 Postscale
:
:
0000_1111 = 1: 16 Postscale
0001_0000 = 1: 17 Postscale
:
:
1111_1111 = 1: 256 Postscale

13.11 PWM Interrupt Flag(PWMINTF)

Mnemonic: PWMINTF **Address: BCh**

7	6	5	4	3	2	1	0	Reset
PWMTBDIR	-					SEVTIF	PWMPIF	00H

PWMTBDIR:	PWM Time Base Count Direction Status(Read only) 0 = counts up 1 = counts down
SEVTIF:	Special Event Interrupt Flag
PWMPIF:	PWM Period Interrupt Flag

13.12 Dead Time

When the half-bridge circuit is applying, at the same time of upper and lower arm turn state period, due to characters of TON and TOFF, power crystal can not instantaneous complete turn state, so as to cause a short circuit, then must spare a certain time to allow power crystal turn state.

Each pair of complementary PWM output have a 6 bit down counter, due to produce dead time as below figure Fig. 13-2, each dead time unit has a rising edge and falling edge detector, according to the counter and when the value of number is zero, the output is just converted.

$$Td = \frac{2^{(DT0PRE+1)}}{Fosc} \times DT0$$

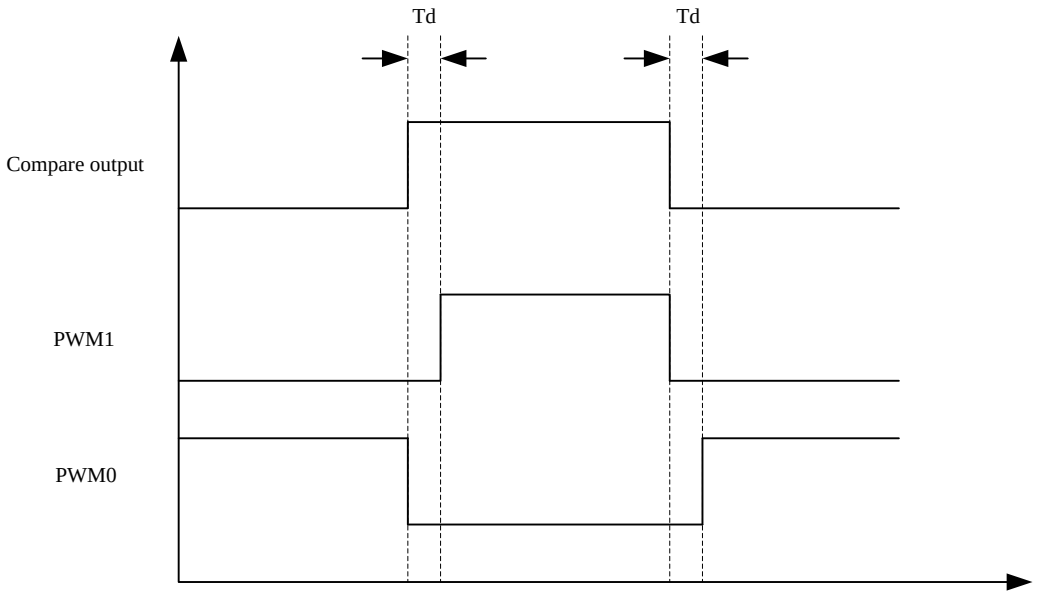


Fig. 13-2: PWM output Compare

13.12.1 Dead Time 0 for PWM Pair 0(DEADTIME0)

Mnemonic: DEADTIME0 **Address: E9h**

7	6	5	4	3	2	1	0	Reset
DT0PRE[1:0]				DT0[5:0]				00H

DT0PRE[1:0]	Dead Time 0 Prescale 00 = Fosc/2 01 = Fosc/4 10 = Fosc/8 11 = Fosc/16
DT0[5:0]	Dead Time 0

Freq = 24MHz, Period = 14 bit		
Prescale	**Dead Time Min**	**Dead Time Max**
1:2	83 ns	5.3 us (To apply to CCD)
1:4	166 ns	10.6 us
1:8	332 ns	21.2 us
1:16	664 ns	42.4 us

13.12.2 Dead Time 1 for PWM Pair 1(DEADTIME1)

Mnemonic: DEADTIME1 **Address: EAh**

7	6	5	4	3	2	1	0	Reset
DT1PRE[1:0]				DT1[5:0]		00H		

DT1PRE[1:0]	Dead Time 1 Prescale 00 = Fosc/2 01 = Fosc/4 10 = Fosc/8 11 = Fosc/16
DT1[5:0]	Dead Time 1 00_0000 = 1 Dead Time 1 Unit. 00_0001 = 2 Dead Time 1 Units ……. 11_1111 = 64 Dead Time 1 Units.

13.12.3 Dead Time 2 for PWM Pair 2(DEADTIME2)

Mnemonic: DEADTIME2 **Address: EBh**

7	6	5	4	3	2	1	0	Reset
DT2PRE[1:0]				DT2[5:0]				00H

DT2PRE[1:0]	Dead Time 2 Prescale 00 = Fosc/2 01 = Fosc/4 10 = Fosc/8 11 = Fosc/16
DT2[5:0]	Dead Time 2 00_0000 = 1 Dead Time 2 Unit. 00_0001 = 2 Dead Time 2 Units ……. 11_1111 = 64 Dead Time 2 Units.

13.12.4 Dead Time 3 for PWM Pair 3(DEADTIME3)

Mnemonic: DEADTIME3 **Address: ECh**

7	6	5	4	3	2	1	0	Reset
DT3PRE[1:0]				DT3[5:0]				00H

DT3PRE[1:0]	Dead Time 3 Prescale 00 = Fosc/2 01 = Fosc/4 10 = Fosc/8 11 = Fosc/16
DT3[5:0]	Dead Time 3 00_0000 = 1 Dead Time 3 Unit. 00_0001 = 2 Dead Time 3 Units ……. 11_1111 = 64 Dead Time 3 Units.

13.12.5 Override Disable(OVRIDEDIS)

Mnemonic: OVRIDEDIS **Address: DEh**

7	6	5	4	3	2	1	0	Reset
OV7DIS	OV6DIS	OV5DIS	OV4DIS	OV3DIS	OV2DIS	OV1DIS	OV0DIS	FFH

OV7DIS Override Disable 7 Action Selection

OV7DIS = 0 - PWM7 Override Enable

OV7DIS = 1 - PWM7 Override Disable

OV6DIS Override Disable 6 Action Selection

OV6DIS = 0 - PWM6 Override Enable

OV6DIS = 1 - PWM6 Override Disable

OV5DIS Override Disable 5 Action Selection

OV5DIS = 0 - PWM5 Override Enable

OV5DIS = 1 - PWM5 Override Disable

OV4DIS Override Disable 4 Action Selection

OV4DIS = 0 - PWM4 Override Enable

OV4DIS = 1 - PWM4 Override Disable

OV3DIS Override Disable 3 Action Selection

OV3DIS = 0 - PWM3 Override Enable

OV3DIS = 1 - PWM3 Override Disable

OV2DIS Override Disable 2 Action Selection

OV2DIS = 0 - PWM2 Override Enable

OV2DIS = 1 - PWM2 Override Disable

OV1DIS Override Disable 1 Action Selection

OV1DIS = 0 - PWM1 Override Enable

OV1DIS = 1 - PWM1 Override Disable

OV0DIS Override Disable 0 Action Selection

OV0DIS = 0 - PWM0 Override Enable

OV0DIS = 1 - PWM0 Override Disable

13.12.6 Override Data (OVRIDEDATA)

Mnemonic: OVRIDEDATA **Address: DFh**

7	6	5	4	3	2	1	0	Reset
OV7DATA	OV6DATA	OV5DATA	OV4DATA	OV3DATA	OV2DATA	OV1DATA	OV0DATA	00H

OV7DATA Ovride Data 7

OV7DATA = 0 - PWM7 Override Data

OV7DATA = 1 - PWM7 Override Data

OV6DATA Ovride Data 6

OV6DATA = 0 - PWM6 Override Data

OV6DATA = 1 - PWM6 Override Data

OV5DATA Ovride Data 5

OV5DATA = 0 - PWM5 Override Data

OV5DATA = 1 - PWM5 Override Data

OV4DATA Ovride Data 4

OV4DATA = 0 - PWM4 Override Data

OV4DATA = 1 - PWM4 Override Data

OV3DATA Ovride Data 3

OV3DATA = 0 - PWM3 Override Data

OV3DATA = 1 - PWM3 Override Data

OV2DATA Ovride Data 2

OV2DATA = 0 - PWM2 Override Data

OV2DATA = 1 - PWM2 Override Data

OV1DATA Ovride Data 1

OV1DATA = 0 - PWM1 Override Data

OV1DATA =1 - PWM1 Override Data

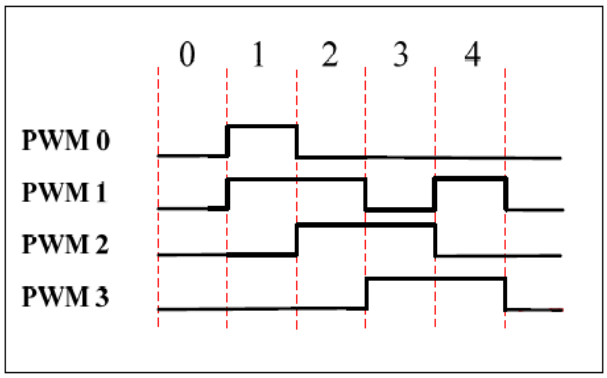
OV0DATA Ovride Data 0

OV0DATA = 0 - PWM0 Override Data

OV0DATA = 1 - PWM0 Override Data

Example: PWM Output overrides waveform.

Period Cycle	OVRIDEDIS[7:0]	OVRIDEDATA[7:0]	PWMPOLARITY[7:0]
1	11110000b	00000011b	11111111b
2	11110000b	00000110b	11111111b
3	11110000b	00001100b	11111111b
4	11110000b	00001010b	11111111b



13.12.7 PWM Polarity (PWMPOLARITY)

Mnemonic: PWMPOLARITY **Address: DDh**

7	6	5	4	3	2	1	0	Reset
POLARITY7	POLARITY6	POLARITY5	POLARITY4	POLARITY3	POLARITY2	POLARITY1	POLARITY0	FFH

POLARITY7 PWM Polarity 7

POLARITY7 = 0 - PWM7 Polarity active low

POLARITY7 = 1 - PWM7 Polarity active high

POLARITY6 PWM Polarity 6

POLARITY6 =0 - PWM6 Polarity active low

POLARITY6 =1 - PWM6 Polarity active high

POLARITY5 PWM Polarity 5

POLARITY5 = 0 - PWM5 Polarity active low

POLARITY5 = 1 - PWM5 Polarity active high

POLARITY4 PWM Polarity 4

POLARITY4 = 0 - PWM4 Polarity active low

POLARITY4 = 1 - PWM4 Polarity active high

POLARITY3 PWM Polarity 3

POLARITY3 = 0 - PWM3 Polarity active low

POLARITY3 = 1 - PWM3 Polarity active high

POLARITY2 PWM Polarity 2

POLARITY2 = 0 - PWM2 Polarity active low

POLARITY2 = 1 - PWM2 Polarity active high

POLARITY1 PWM Polarity 1

POLARITY1 = 0 - PWM1 Polarity active low

POLARITY1 = 1 - PWM1 Polarity active high

POLARITY0 PWM Polarity 0

POLARITY0 = 0 - PWM0 Polarity active low

POLARITY0 = 1 - PWM0 Polarity active high

13.13 Fault Configure (FLTCONFIG)

When FLTA or FLTB are in use, if hardware detects any abnormal signals, the status of PMW will shift to inactive automatically.

13.14 PWM Fault Inputs

The PWM module provides a fault function via FLTA and FLTB tw0 output. To disable the output signals of the PWM is their main function and as well as to enter an inactive status. When the fault occurs, the hardware will performer forthwith and shift the PWM in an inactive status; and meanwhile remain power-on connected to the PWM . Under normal working status, either low or high active can be directed by the users by simple operations.

Each of the fault inputs have two modes of operation

Inactive Mode

If the Fault occurs, the output signals of the PWM is deactivated. The status of the PWM will remain in inactive and correspond to flag of the FLTxS flag and also set it up. If the PWM need to be recovered in a normal output working status by the time the Fault flag of the FLTxS status must be cleared by the software.

Cycle-by-Cycle Mode

When the Fault function occurs, the output of the PWM is deactived. The status of the PWM pin will remain in inactive status and correspond to flag of the FLTxS flag and set it up. When the Fault is relieved, the FLTxS will be relatively cleared, and the output of the PWM will be recovered to normal working status.

Mnemonic: FLTCONFIG **Address: DBh**

7	6	5	4	3	2	1	0	Reset
BRFEN	FLTBS	FLTBMOD	FLTBEN	FLTCON	FLTAS	FLTAMOD	FLTAEN	80H

BRFEN Breakpoint Fault Enable

BRFEN = 0 - Disable

BRFEN = 1 - Enable

FLTBS Fault B status, must be cleared by SW(inactive mode)

FLTBS = 0 - No Fault

FLTBS = 1 - /FLTB is asserted.

FLTBMOD FLTB Mode Set

FLTBMOD = 0 - inactive mode

FLTBMOD = 1 - cycle-by-cycle mode

FLTBEN FLTB Active Set

FLTBEN = 0 - Disable Fault B function

FLTBEN = 1 - Enable Fault B function

FLTCON 0 = inactive PWM[5:0]

1 = inactive PWM[7:0]

FLTAS Fault A status, must be cleared by SW(inactive mode)

FLTAS = 0 - No Fault

FLTAS = 1 - /FLTA is asserted.

FLTAMOD FLTA Mode Set

FLTAMOD = 0 - inactive mode

FLTAMOD = 1 - cycle-by-cycle mode

FLTAEN FLTA Active Set

FLTAEN = 0 - Disable Fault A function

FLTAEN = 1 - Enable Fault A function

13.15 Fault Noise Filter(FLTNF)

Mnemonic: FLTNF **Address: DCh**

7	6	5	4	3	2	1	0	Reset
-	-	FLTBLS	FLTALS	FLTBNF[1:0]		FLTANF[1:0]		00H

FLTBLS:	Fault B level select 0: active low 1: active high
FLTALS:	Fault A level select 0: active low 1: active high
FLTANF [1:0]	Fault A noise filter 00 = Fosc/1 01 = Fosc/2 10 = Fosc/4 11 = Fosc/8
FLTBNF [1:0]	Fault B noise filter 00 = Fosc/1 01 = Fosc/2 10 = Fosc/4 11 = Fosc/8

13.16 PWM Pair 0 Duty(DUTY0L, DUTY0H)

Mnemonic: DUTY0L **Address: D1h**

6	5	4	3	2	1	0	Reset
PWM Pair 0 Duty Low 8 bit [7:0]							00H

Mnemonic: DUTY0H **Address: D2h**

6	5	4	3	2	1	0	Reset
-	PWM Pair 0 Duty High 6 bit [5:0]						00H

13.17 PWM Pair 1 Duty(DUTY1L, DUTY1H)

Mnemonic: DUTY0L **Address: D3h**

6	5	4	3	2	1	0	Reset
PWM Pair 1 Duty Low 8 bit [7:0]							00H

Mnemonic: DUTY0H **Address: D4h**

7	6	5	4	3	2	1	0	Reset
-				PWM Pair 1 Duty High 6 bit [5:0]				00H

13.18 PWM Pair 2 Duty(DUTY2L, DUTY2H)

Mnemonic: DUTY2L **Address: D5h**

6	5	4	3	2	1	0	Reset
PWM Pair 2 Duty Low 8 bit [7:0]							00H

Mnemonic: DUTY0H **Address: D6h**

6	5	4	3	2	1	0	Reset
-	PWM Pair 2 Duty High 6 bit [5:0]						00H

13.19 PWM Pair 3 Duty(DUTY3L, DUTY3H)

Mnemonic: DUTY3L **Address: D7h**

7	6	5	4	3	2	1	0	Reset
PWM Pair 3 Duty Low 8 bit [7:0]								00H

Mnemonic: DUTY3H **Address: C9h**

7	6	5	4	3	2	1	0	Reset
-	PWM Pair 3 Duty High 6 bit [5:0]							00H

14. 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											
AUX	Auxiliary register	91h	BRGS	-	P4SPI	P4UR1	P4IIC	P0KBI	-	DPS	00H
IICCTL	IIC control register	F9h	IICEN	MSS	MAS	AB_EN	BF_EN	IICBR[2:0]			04H
IICS	IIC status register	F8h	-	MPIF	LAIF	RXIF	TXIF	RXAK	TXAK	RW or BB	00H
IICA1	IIC Address 1 register	FAh	IICA1[7:1]							MATCH1or RW1	A0H
IICA2	IIC Address 2 register	FBh	IICA2[7:1]							MATCH2 or RW2	60H
IICRWD	IIC Read/Write register	FCh	IICRWD[7:0]								00H
IICEBT	IIC Enaable Bus Transaction	FDh	FU_EN		-	-	-	-	-	-	00H

Mnemonic: AUX **Address: 91h**

7	6	5	4	3	2	1	0	Reset
BRGS	-	P4SPI	P4UR1	P4IIC	P0KBI	-	DPS	00H

P4IIC: P4IIC = 0 – IIC function on P1.

P4IIC = 1 – IIC function on P4.

14.1 IIC Control Register(IICCTL)

Mnemonic: IICCTL **Address: F9h**

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

IICEN: Enable IIC module

IICEN = 1 is Enable

IICEN = 0 is Disable.

MSS: Master or slave mode select.

MSS = 1 is master mode.

MSS = 0 is slave mode.

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

MAS: Master address select (master mode only)

MAS = 0 is to use IICA1.

MAS = 1 is to use IICA2.

AB_EN: Arbitration lost enable bit. (Master mode only)

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

BF_EN: Bus busy enable bit. (Master mode only)

If set BF_EN bit, hardware will not generate a start condition to bus until BF=0. Clear this bit will always generate a start condition to bus when MStart is set. Set this bit when multi-master and slave connection. Clear this bit when single master to single slave.

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

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

14.2 IIC Status Register(IICS)

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).
As shown in Fig. 14-1

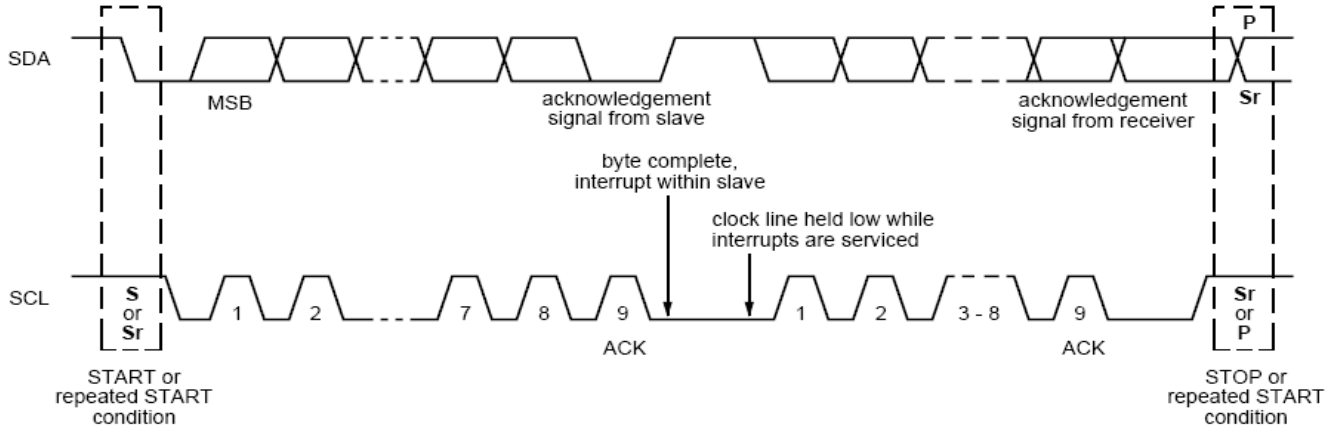


Fig. 14-1: Acknowledgement bit in the 9th bit of a byte transmission

14.3 IIC Address1 Register(IICA1)

Mnemonic: IICA1 **Address: FA**

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

Slave mode:

IICA1[7:1]: IIC Address registers

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

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

Master mode:

IICA1[7:1]: IIC Address registers

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

RW1: This bit will be sent out as RW of the slave side if the module has set the MStart or RStart bit. It appears at the 8th bit after the IIC address as below figure. 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. As shown in Fig. 14-2

RW1=1, master receive mode

RW1=0, master transmit mode

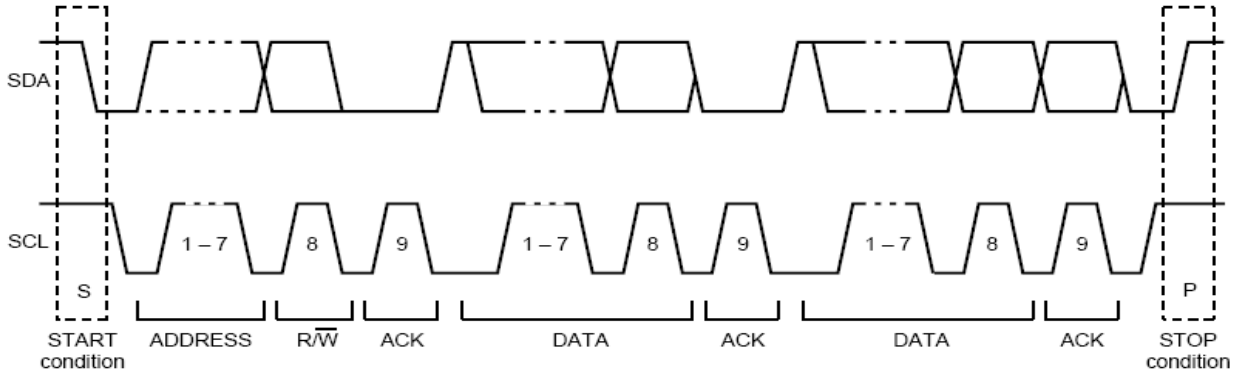


Fig. 14-2: RW bit in the 8th bit after IIC address

14.4 IIC Address2 Register(IICA2)

Mnemonic: IICA2 **Address: FB**

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

Slave mode:

IICA2[7:1]: IIC Address registers

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

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

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

Master mode:

IICA2[7:1]: IIC Address registers

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

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

RW2=1, master receive mode

RW2=0, master transmit mode

14.5 IIC Read Write Register(IICRWD)

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.

14.6 IIC Enable Bus Transaction Register(IICEBT)

Mnemonic: IICEBT **Address: FDH**

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

Master Mode：

00: reserved

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

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

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

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

Slave mode:

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

Notice:

FU_EN[7:6] should be set as 01 before read/write data transfer for bus release; otherwise, SCL will be locked(pull low).

FU_EN[7:6] 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[7:6] as 01.

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

15. SPI Function - Serial Peripheral Interface

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

The interrupt vector is 4Bh.

There are 4 signals used in SPI, they are
SPI_MOSI: data output in the master mode, data input in the slave mode,
SPI_MISO: data input in the master mode, data output in the master mode,
SPI_SCK: clock output from the master, the above data are synchronous to this signal
SPI_SS: input in the slave mode.

This slave device detects this signal to judge if it is selected by the master. As shown in Fig. 15-1

In the master mode, it can select the desired slave device by any IO with value = 0. As below figure is an example showing the relation of the 4 signals between master and slaves.

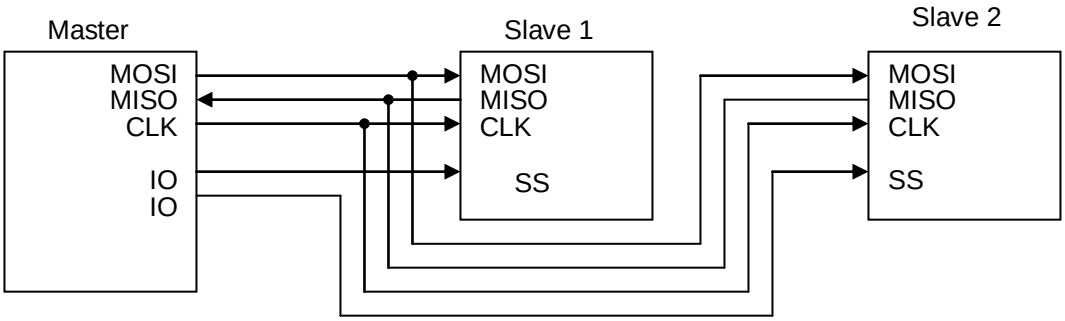


Fig. 15-1: SPI signals between master and slave devices

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

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
The relevant registers of the SPI function											
AUX	Auxiliary register	91h	BRGS	-	P4SPI	P4UR1	P4IIC	P0KBI	-	DPS	00H
SPI function											
SPIC1	SPI control register 1	F1h	SPIEN	SPIMSS	SPISSP	SPICKP	SPICKE	SPIBR[2:0]			08H
SPIC2	SPI control register 2	F2h	SPIFD	TBC[2:0]			SPIRST	RBC[2:0]			00H
SPIS	SPI status register	F5h	SPIRF	SPIMLS	SPIOV	SPITXIF	SPITDR	SPIRXIF	SPIRDR	SPIRS	40H
SPITXD	SPI transmit data buffer	F3h	SPITXD[7:0]								00H
SPIRXD	SPI receive data buffer	F4h	SPIRXD[7:0]								00H

Mnemonic: AUX **Address: 91h**

7	6	5	4	3	2	1	0	Reset
BRGS	-	P4SPI	P4UR1	P4IIC	P0KBI	-	DPS	00H

P4SPI: P4SPI = 0 – SPI function on P1.

P4SPI = 1 – SPI function on P4.

15.1 SPI Control Register 1(SPIC1)

Mnemonic:SPIC1 **Address:F1H**

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

SPIEN: Enable SPI module.

SPIEN = 1 - is Enable.

SPIEN = 0 - is Disable.

SPIMSS: Master or Slave mode Select

SPIMSS = 1 - is Master mode.

SPIMSS = 0 - is Slave mode.

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

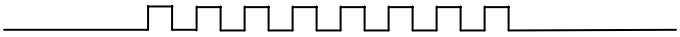
SPISSP = 1 - high active.

SPISSP = 0 - low active.

SPICKP: Clock idle polarity select. (Master mode used only)

SPICKP = 1 - SCK will idle high. Ex :

SPICKP = 0 - SCK will idle low. Ex :

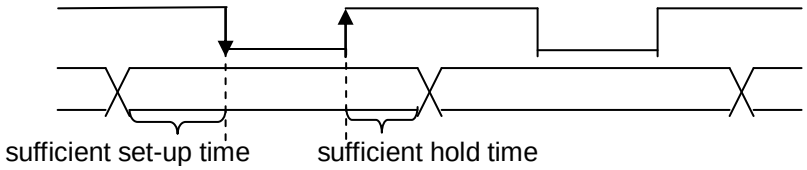


SPICKE: Clock sample edge select.

SPICKE = 1 - rising edge latch data.

SPICKE = 0 - falling edge latch data.

* To ensure the data latch stability, SM59A16U1 generate the output data As shown in the following example, the other side can latch the stable data no matter in rising or falling edge.



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

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

15.2 SPI Control Register 2(SPIC2)

Mnemonic: SPIC2 **Address: F2H**

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

SPIFD: Full-duplex mode enable.

SPIFD = 1 is enable full-duplex mode.

SPIFD = 0 is disable full-duplex mode.

When it is set, the TBC[2:0] and RBC[2:0] will be reset and keep to zero. When the Master device transmits data to the Slave device via the MOSI line, the Slave device responds by sending data to the Master device via the MISO line. This implies full-duplex transmission with both data out and data in synchronized with the same clock. As shown in Fig. 15-2

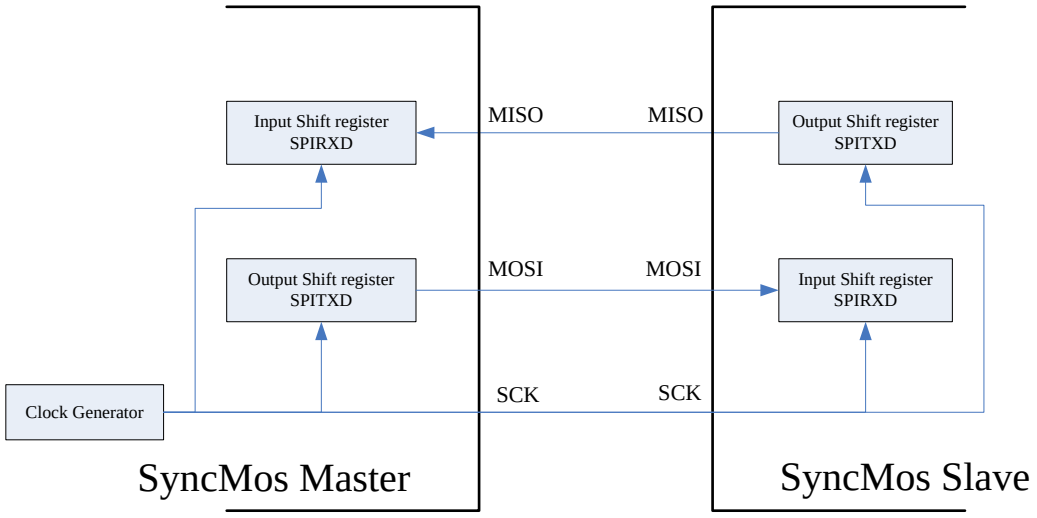


Fig. 15-2: SPI Mater and slave transfer method

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

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

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

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

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

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

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

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

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

15.3 SPI Status Register (SPIS)

Mnemonic: SPIS **Address:F5H**

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

SPIRF: SPI SS pin Release Flag.

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

SPIMLS: MSB or LSB first output /input Select.

SPIMLS = 1 is MSB first output/input.

SPIMLS = 0 is LSB first output/input.

SPIOV: Overflow flag.

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

It is clear by hardware, when SPIRDR is cleared.

SPITXIF: Transmit Interrupt Flag.

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

SPITDR: Transmit Data Ready.

When MCU finish writing data to SPITXD register, the MCU needs to set this bit to '1' to inform the SPI module to send the data. After SPI module finishes sending the data

from SPITXD, this bit will be cleared automatically.

SPIRXIF: Receive Interrupt Flag.

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

SPIRDR: Receive Data Ready.

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

SPIRS: Receive Start.

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

15.4 SPI Transmit Data Buffer (SPITXD)

Mnemonic: SPITXD **Address: F3H**

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

SPITXD[7:0]: SPI Receive Data Buffer

15.5 SPI Receive Data Buffer (SPIRXD)

Mnemonic: SPIRXD **Address: F4H**

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

SPIRXD[7:0]: Receive data buffer.

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

16. KBI – Keyboard Interface

Keyboard interface (KBI) can be connected to an 8 x n matrix keyboard or any similar devices. It has 8 inputs with programmable interrupt capability on either high or low level. These 8 inputs are through P2 or P0 and can be the external interrupts to leave from the idle and stop modes. As shown in Fig. 16-1 and Fig. 16-2 , The 8 inputs are independent from each other but share the same interrupt vector 5Bh.

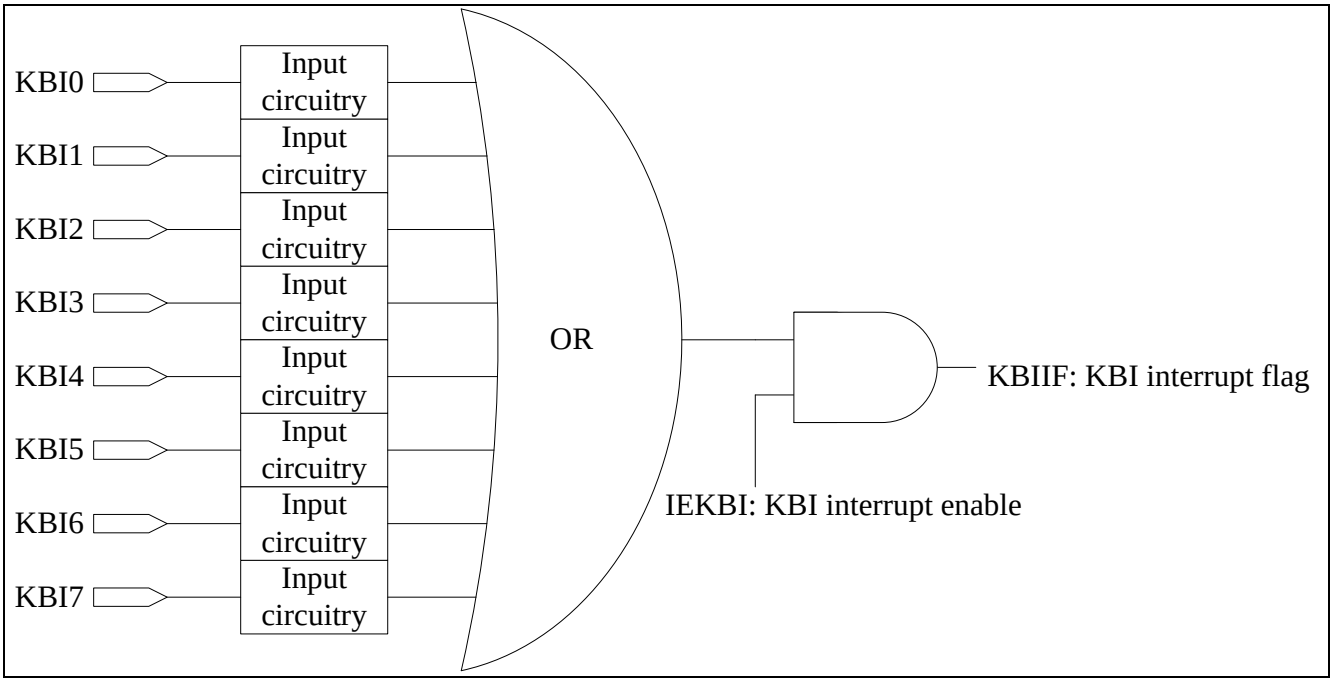


Fig. 16-1: keyboard interface block diagram

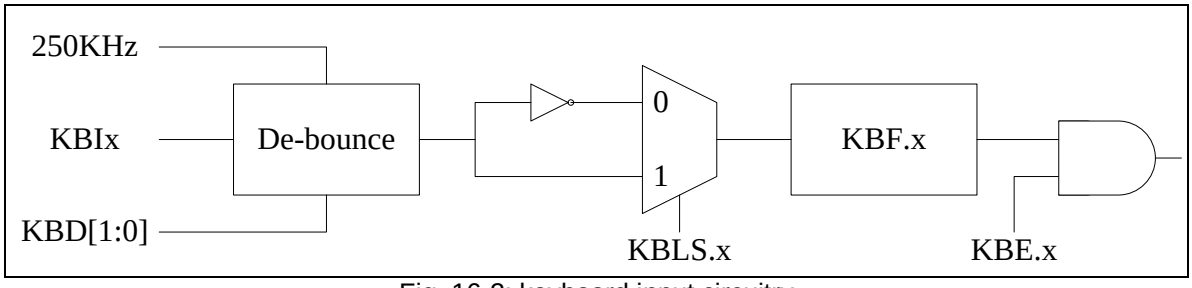


Fig. 16-2: keyboard input circuitry

KBI	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
The relevant registers of the KBI function											
AUX	Auxiliary register	91h	BRGS	-	P4SPI	P4UR1	P4IIC	P0KBI	-	DPS	00H
KBI function											
KBLS	KBI level selection	93h	KBLS7	KBLS6	KBLS5	KBLS4	KBLS3	KBLS2	KBLS1	KBLS0	00H
KBE	KBI input enable	94h	KBE7	KBE6	KBE5	KBE4	KBE3	KBE2	KBE1	KBE0	00H
KBF	KBI flag	95h	KBF7	KBF6	KBF5	KBF4	KBF3	KBF2	KBF1	KBF0	00H
KBD	KBI De-bounce control register	96h	KBDEN	-					KBD1	KBD0	00H

Mnemonic: AUX **Address: 91h**

7	6	5	4	3	2	1	0	Reset
BRGS	-	P4SPI	P4UR1	P4IIC	P0KBI	-	DPS	00H

P0KBI: P0KBI = 0 – KBI function on P2.

P0KBI = 1 – KBI function on P0.

16.1 Keyboard Level Selector Register(KBLS)

Mnemonic: KBLS **Address: 93h**

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

KBLS.7: Keyboard Line 7 level selection bit

0 : enable a low level detection on KBI7.

1 : enable a high level detection on KBI7.

KBLS.6: Keyboard Line 6 level selection bit

0 : enable a low level detection on KBI6.

1 : enable a high level detection on KBI6.

KBLS.5: Keyboard Line 5 level selection bit

0 : enable a low level detection on KBI5.

1 : enable a high level detection on KBI5.

KBLS.4: Keyboard Line 4 level selection bit

0 : enable a low level detection on KBI4.

1 : enable a high level detection on KBI4.

KBLS.3: Keyboard Line 3 level selection bit

0 : enable a low level detection on KBI3.

1 : enable a high level detection on KBI3.

KBLS.2: Keyboard Line 2 level selection bit

0 : enable a low level detection on KBI2.

1 : enable a high level detection on KBI2.

KBLS.1: Keyboard Line 1 level selection bit

0 : enable a low level detection on KBI1.

1 : enable a high level detection on KBI1.

KBLS.0: Keyboard Line 0 level selection bit

0 : enable a low level detection on KBI0.

1 : enable a high level detection on KBI0.

16.2 Keyboard Interrupt Enable Register(KBE)

Mnemonic: KBE **Address: 94h**

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

KBE.7: Keyboard Line 7 enable bit

0 : enable standard I/O pin.

1 : enable KBF.7 bit in KBF register to generate an interrupt request.

KBE.6: Keyboard Line 6 enable bit

0 : enable standard I/O pin.

1 : enable KBF.6 bit in KBF register to generate an interrupt request.

KBE.5: Keyboard Line 5 enable bit

0 : enable standard I/O pin.

1 : enable KBF.5 bit in KBF register to generate an interrupt request.

KBE.4: Keyboard Line 4 enable bit

0 : enable standard I/O pin.

1 : enable KBF.4 bit in KBF register to generate an interrupt request.

KBE.3: Keyboard Line 3 enable bit

0 : enable standard I/O pin.

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

KBE.2: Keyboard Line 2 enable bit

0 : enable standard I/O pin.

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

KBE.1: Keyboard Line 1 enable bit

0 : enable standard I/O pin.

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

KBE.0: Keyboard Line 0 enable bit

0 : enable standard I/O pin.

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

16.3 Keyboard Interrupt Flag Register(KBF)

Mnemonic: KBF **Address: 95h**

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

KBF.7: Keyboard Line 7 flag

This is set by hardware when KBI7 detects a programmed level.

It generates a Keyboard interrupt request if KBE.7 is also set. It must be cleared by software.

KBF.6: Keyboard Line 6 flag

This is set by hardware when KBI6 detects a programmed level.

It generates a Keyboard interrupt request if KBE.6 is also set. It must be cleared by software.

KBF.5: Keyboard Line 5 flag

This is set by hardware when KBI5 detects a programmed level.

It generates a Keyboard interrupt request if KBE.5 is also set. It must be cleared by software.

KBF.4: Keyboard Line 4 flag

This is set by hardware when KBI4 detects a programmed level.

It generates a Keyboard interrupt request if KBE.4 is also set. It must be cleared by software.

KBF.3: Keyboard Line 3 flag

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

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

KBF.2: Keyboard Line 2 flag

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

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

KBF.1: Keyboard Line 1 flag

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

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

KBF.0: Keyboard Line 0 flag

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

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

16.4 Keyboard De-bounce Control Register(KBD)

Mnemonic: KBD **Address: 96H**

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

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

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

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

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

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

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

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

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

17. 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
Watchdog Timer											
LVC	Low voltage control register	E6h	LVI_EN	LVRLPE	LVRE	LVSIF	-	-	LVIS[1:0]		20H

17.1 Low Voltage Control Register(LVC)

Mnemonic: LVC **Address: E6h**

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

LVRLPE: Internal low voltage reset low power function enable bit. (Refer MCU core voltage)

Suggest using this function when MCU is on power down mode.

LVRLPE = 0 - disable Internal low voltage reset function.

LVRLPE = 1 - enable Internal low voltage reset function.

(When the internal voltage is below 1 V, it will generate reset.

LVRE: External low voltage reset function enable bit. (Refer MCU's VDD voltage).

LVRE = 0 - disable external low voltage reset function.

LVRE = 1 - enable external low voltage reset function.

(When the external voltage is below 1.6V, it will generate reset.)

LVSIF: Low Voltage Status Flag.

LVIS[1:0]: Low Voltage Interrupt Select:

00: 1.75V

01: 2.75V

10: 3.35V

11: 4.20V

18. 10-bit Analog-to-Digital Converter (ADC)

ADC module features:

- The SM59A16U1 provides 8+1 channels 10-bit ADC.
- The Digital output DATA [9:0] were put into ADCD [9:0].
- The ADC channel 8 is only for OP0 convert to ADC function.
- The ADC interrupt vector is 53H.

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
ADC											
ADCC1	ADC Control register 1	ABh	ADC7EN	ADC6EN	ADC5EN	ADC4EN	ADC3EN	ADC2EN	ADC1EN	ADC0EN	00H
ADCC2	ADC Control register 2	ACh	Start	ADJUST	PWMTriggerEN	EXTTriggerEN	ADC MODE	ADCCH[2:0]			00H
ADCDH	ADC data high byte	ADh	ADCDH [7:0]								00H
ADCDL	ADC data low byte	AEh	ADCDL [7:0]								00H
ADCCS	ADC clock select	AFh	OP0ToADC	-	ADCENToP34	ADCCS[4:0]					00H

18.1 ADC Control Register 1(ADCC1)

Mnemonic: ADCC1 **Address: ABh**

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

ADC7EN: ADC Channels 7 Enable.
1 = Enable ADC channel 7
0 = Disable ADC channel 7

ADC6EN: ADC Channels 6 Enable.
1 = Enable ADC channel 6
0 = Disable ADC channel 6

ADC5EN: ADC Channels 5 Enable.
1 = Enable ADC channel 5
0 = Disable ADC channel 5

ADC4EN: ADC Channels 4 Enable.
1 = Enable ADC channel 4
0 = Disable ADC channel 4

ADC3EN: ADC Channels 3 Enable.
1 = Enable ADC channel 3
0 = Disable ADC channel 3

ADC2EN: ADC Channels 2 Enable.

1 = Enable ADC channel 2

0 = Disable ADC channel 2

ADC1EN: ADC Channels 1 Enable.

1 = Enable ADC channel 1

0 = Disable ADC channel 1

ADC0EN: ADC Channels 0 Enable.

1 = Enable ADC channel 0

0 = Disable ADC channel 0

18.2 ADC Control Register 2(ADCC2)

Mnemonic: ADCC2 **Address: ACh**

7	6	5	4	3	2	1	0	Reset
Start	Adjust	PWM Trigger EN	EXT Trigger EN	ADC MODE	ADCCH[2:0]			00H

Start: ADC conversion by SW trigger.
0 = ADC conversion stop.
1 = ADC conversion start.

ADJUST: Adjust the format of ADC conversion DATA.
0 = ADC data format 1. (Default)
ADC data high byte ADCD [9:2] = ADCDH [7:0].
ADC data low byte ADCD [1:0] = ADCDL [1:0].
1 = ADC data format 2.
ADC data high byte ADCD [9:8] = ADCDH [1:0].
ADC data low byte ADCD [7:0] = ADCDL [7:0].

PWMTriggerEN: PWM trigger ADC conversion. (HW internal trigger by PWM0 ~ PMW7)
0 = Disable
1 = Enable

EXTTriggerEN: External Pin triggers ADC conversion. (HW external trigger by TRIGADC Pin)
0 = Disable
1 = Enable

ADCMODE: 0 = Continuous mode.
1 = Single-shot mode. (For barcode solution: PWM trigger ADC)

ADCCH[2:0]: ADC channel select.

ADCCH [2:0]	Channel
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

18.3 ADC Data(ADCDH, ADCDL)

When ADJUST = 0, the ADC data format 1 as below:

Mnemonic: ADCDH **Address: ADh**

7	6	5	4	3	2	1	0	Reset
ADCD[9]	ADCD[8]	ADCD[7]	ADCD[6]	ADCD[5]	ADCD[4]	ADCD[3]	ADCD[2]	00H

Mnemonic: ADCDL **Address: AEh**

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

When ADJUST = 1, the ADC data format 2 as below:

Mnemonic: ADCDH **Address: ADh**

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

Mnemonic: ADCDL **Address: AEh**

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

18.4 ADC Clock Select(ADCCS)

Mnemonic: ADCCS **Address: AFh**

7	6	5	4	3	2	1	0	Reset
OP0 ToADC	-	ADCEN ToP34	ADCCS[4:0]					00H

OP0ToADC: Select ADC channel 8 as input source

0 = Set ADC input source as decided by ADCC2.

1 = Set ADC input source as Op0 output.

ADCENToP34: ADC internal signal test and monitor.

0 = Disable ADC internal signal output to P3.4

1 = Enable ADC internal signal output to P3.4

ADCCS[4:0]: ADC clock select.

$$ADC\_Clock = \frac{\text{Fosc}}{2 \times (ADCCS + 1)}$$

$$ADC\_Conversion\_Rate = \frac{\text{ADC\_Clock}}{13}$$

19. USB function

USB peripheral module supports USB Control In/Out, Interrupt In/Out and Bulk In/Out transfers. I.e., the implementation supports 5 endpoint numbers (0, 1, 2, 3, 4) for a total of 5 endpoints. The Serial Interface Engine (SIE) handles all the USB 2.0 protocol and provides a simple Read/Write protocol for the MCU.

The MCU is a USB function device, and as a result is always a slave to the USB host. The USB host initiates all USB data transfers to and from the MCU USB port. There are 5 USB endpoints associated with MCU:

Endpoint 0: This control endpoint is used to initialize the device, and provides access to USB configuration, control and status registers. This endpoint is an 8 bytes bi-direction FIFO.

Endpoint 1: This endpoint supports interrupt transfers from the MCU transmit mailboxes to the USB host.

Endpoint 1 is 8 bytes interrupt endpoint.

Endpoint 2: This endpoint supports interrupt transfers from the USB host to the MCU device.

Endpoint 2 is 8 bytes interrupt endpoint.

Endpoint 3: This endpoint supports bulk data transfers from the MCU transmit FIFO to the USB host.

Endpoint 3 is 64 bytes transmitted FIFO.

Endpoint 4: This endpoint supports bulk data transfers from the USB host to the MCU receive FIFO.

Endpoint 4 is 64 bytes received FIFO.

19.1 USB Device Enumeration Transfer

The endpoint 0 is playing an important role in enumeration step. The Serial Interface Engine is designed for handling the entire most USB standard commands exclude Class/Vendor, GetDescriptor, SetDescriptor, and SynchFrame command. If MCU is received these 4 types command, the SIE will pass the command to endpoint 0, the USB interrupt flag register 1 (UIFR1) bit 0 will be set to notify MCU to read endpoint zero command. MCU program should have the ability to decode these commands and respond with relationship descriptors (MCU write device descriptor to endpoint 0). After completed these steps, MCU should be set Endpoint Data Ready Control Register (EPDRDY) bit 1. The SIE will fetch these descriptors data and transfer to host by USB cable. Two index pointers are used to access the endpoint 0. It must be initialized by the MCU, and is automatically incremented after the MCU (or host) reads (read pointer) or writes (write pointer) the endpoint 0 data register.

19.2 USB Interrupt In Transfer

Endpoint 1 is used for interrupt transfers to the USB host from a set of 8 bytes FIFO registers that are written by the MCU. To transfer a 8 bytes packet, the MCU writes data into the 8 bytes registers and set the transmit flag bit (EPDRDY bit 2, endpoint 1 transmitted data ready). The SIE will fetch endpoint 1 data and transfer data to host. If the USB host tries to read endpoint 1 when the endpoint 1 transmitted data ready bit is not set, a NAK acknowledge is returned. After the USB interrupt transfer completed, the UIFR1 bit 1 will be set to notify MCU that endpoint 1 registers have been read by the USB host. This guarantees that a previous interrupt transfer has completed before the endpoint 1 data are changed.

19.3 USB Interrupt Out Transfer

Endpoint 2 is used for interrupt transfers from the USB host to a set of 8 bytes receive registers which are read by the MCU. To transfer a 8bytes packet, the host first performs a USB 8-byte interrupt transfer to the endpoint 2 receive

mailbox registers. UIFR1 bit 2 is then set which can cause a local bus interrupt to signaling MCU check the EPDRDY bit 3 (endpoint 2 received data ready) is set or not. If this bit is set, MCU can read the data from endpoint 2 safely. After MCU completed read stage, it should be cleared the EPDRDY bit 3. Then it can wait until the next packet complete interrupt, and read the entire packet once again. If the USB host tries to write to these registers when the EPDRDY bit 3 is set, a NAK acknowledge will be returned, signaling host that the next data packet could not be accepted.

19.4 USB Bulk Transfer From Host to Device

For host to device transfers, the host first arranges to transfer a block of data from host memory to local shared memory. The USB host performs a bulk-out data transfer over the USB bus to the receive FIFO endpoint 4 in the MCU.

After MCU completed receive data, an interrupt will be generated to signaling MCU check the status register. For example, MCU program should be check EPDRDY bit 5 (endpoint 4 received data ready) is set or not. If this bit is set, MCU can read the data from endpoint 4 safely. After MCU completed read step, it should be cleared EPDRDY bit 5. Then it can wait until the next packet complete interrupt, and read the entire packet once again. If the data ready control bit (EPDRDY bit 5) from the previous packet is not cleared, then the MCU will return a USB NAK acknowledge to the host, signaling that the next data packet could not be accepted.

The MCU can also read handshake status register to detect whether the packet was acknowledged with an ACK, NAK. If these acknowledge bits are set, then a timeout has occurred. For NAK or timeout conditions at the completion of bulk transfers, the USB host will send another OUT token, and MCU should receive the same packet again.

19.5 USB Bulk Transfer From Device to Host

For device to host transfers, the MCU first writes the data block from local memory into the transmit FIFO endpoint 3. While writing data into the endpoint 3, the MCU must keep track of whether there is space available in the FIFO by monitoring the index write pointer. After the block has been loaded into the transmit FIFO, the MCU should be set the transmit flag (EPDRDY bit 4, endpoint 3 transmitted data ready) to notify SIE that FIFO data ready. The USB host sends an IN token to the MCU and starts a USB bulk-in transfer, SIE will fetch endpoint 3 data and transfer data to host.

When the transmit FIFO becomes empty, the SIE will terminate the packet with an End Of Packet, signaling that there is no more data available. Once an end of packet occurs, an interrupt can be generated to the MCU. The MCU can read handshake status register to detect whether the packet was acknowledge with ACK from the host, or whether the MCU respond to the IN token with a NAK.

If these acknowledge bits are set, then a timeout has occurred. For NAK or timeout conditions at the completion of bulk transfers, the USB host will send another IN token, and the MCU should re-transmit the same packet.

USB Module Features:

- Low speed (1.5Mbps) or Full speed (12Mbps).
- Supports control, interrupt and bulk transfer.
- Five endpoints with FIFO:
  - EP0: Control IN/OUT. FIFO: 8 bytes
  - EP1: Interrupt IN. FIFO: 8 bytes.
  - EP2: Interrupt OUT. FIFO: 8 bytes.
  - EP3: Bulk IN. FIFO: 64 bytes.
  - EP4: Bulk OUT. FIFO: 64 bytes.
- The USB interrupt vector is 73h.
- The USBRSM interrupt vector is 7Bh.

Note: Crystal should be 6, 12 or 24MHz to use USB device controller stably.

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
The relevant registers of the USB											
USBADDR	USB Address Register	A6h	USBADDR[7:0]								00h
USBDATA	USB Data Register	A7h	USBDATA[7:0]								00h
USB											
UCTRL1	USB Control 1 Register	FEh	-				EXT_PHY	PD_LDO33	UDC_EN	USB_CTRL_EN	20H
UCTRL2	USB Control 2 Register	FFh	-			RWKUP EN	RWKUP	PUR2_FW_CTRL	PUR2_SW	PUR1_SW	02H
USTAT	USB Status Register	F6h	SPEED	PLL_FS[1:0]		ALT_VAL	INTF_VAL	CFG_VAL	EP0_DTYPE	SETUP	00H
DEVADR	USB Device Address Register	C4h	-	DEVADR[6:0]							00H
FRMNUMH	USB Frame Number Register, High Byte	C5h	-					FRMNUM[10:8]			00H
FRMNUML	USB Frame Number Register, Low Byte	C6h	FRMNUM[7:0]								00H
HSTALL	USB Host Stall Register	C7h	USB function			EP4 HSTALL	EP3 HSTALL	EP2 HSTALL	EP1 HSTALL	EP0 HSTALL	00H
DSTALL	USB Device Stall Register	BFh	-			EP4 DSTALL	EP3 DSTALL	EP2 DSTALL	EP1 DSTALL	EP0 DSTALL	00H
HSKSTAT	USB Handshake Status Register	B1h	-			EP4HSK	EP3HSK	EP2HSK	EP1HSK	EP0HSK	00H
UIER1	USB Interrupt Enable Register 1	B2h	-			EP4IE	EP3IE	EP2IE	EP1IE	EP0IE	00H
UIER2	USB Interrupt Enable Register 2	B3h	RSUMIE	USBIE	-		CFGIE	SOFIE	SUSIE	RSTIE	00H
UIFR1	USB Interrupt Flag Register 1	B4h	-			EP4IF	EP3IF	EP2IF	EP1IF	EP0IF	00H
UIFR2	USB Interrupt Flag Register 2	B5h	RSUMIF	USBIF	-		CFGIF	SOFIF	SUSIF	RSTIF	00H
EPDRDY	USB Endpoint Data Ready Register	B6h	RDYSW	-		EP4RDY	EP3TDY	EP2RDY	EP1TDY	EP0TRDY	2AH
EP0CNT	USB Endpoint 0 Data Counter Register	B7h	-				EP0CNT[3:0]				00H

EP1CNT	USB Endpoint 1 Data Counter Register	A1h	-	EP1CNT[3:0]	00H
EP2CNT	USB Endpoint 2 Data Counter Register	A2h	-	EP2CNT[3:0]	00H
EP3CNT	USB Endpoint 3 Data Counter Register	A3h	-	EP3CNT[6:0]	00H
EP4CNT	USB Endpoint 4 Data Counter Register	A4h	-	EP4CNT[6:0]	00H
EP0DATA	USB Endpoint 0 Data Register	A7h	EP0DATA[7:0]		00H
EP1DATA	USB Endpoint 1 Data Register	9Eh	EP1DATA[7:0]		00H
EP2DATA	USB Endpoint 2 Data Register	9Fh	EP2DATA[7:0]		00H
EP3DATA	USB Endpoint 3 Data Register	93h	EP3DATA[7:0]		00H
EP4DATA	USB Endpoint 4 Data Register	94h	EP4DATA[7:0]		00H

19.6 USB Control 1 Register(UCTRL1)

Mnemonic: UCTRL1 **Address: FEh**

7	6	5	4	3	2	1	0	Reset
-				EXT_PHY	PD_LDO33	UDC_EN	USB_CTRL_EN	20H

EXT_PHY: External PHY enable.

When this bit is set, internal PHY will be disable and all digital signal switch to external to external to connect external PHY.

PD_LDO33: 3.3V LDO power down bit.

0 = normal.

1 = power down.

UDC_EN: UDC Enable.

0 = reset UDC block.

1 = enable.

USB_CTRL_EN: USB control block enable.

0 = reset.

1 = enable.

19.7 USB Control 2 Register(UCTRL2)

Mnemonic: UCTRL2 **Address: FFh**

7	6	5	4	3	2	1	0	Reset
-			RWKUPEN	RWKUP	PUR2_FW_CTRL	PUR_SW	PUR_SW	02H

RWKUP_EN: Remote wakeup function enabled flag.

This enable function was set by HOST using set feature command and cleared by HOST using clear feature command.

0 = Disable.

1 = Enable.

RWKUP_EN: Remote wakeup signal.

Write this bit 1 to generate 1 pulse to inform USB_IF and UDC to do remote wakeup procedure

RUP2_FW_CTRL: Pull up resistor 2 firmware control enable bit.

0 = follow hardware pull up circuit.

1 = control by RUP2_SW (UCTRL2 bit 1).

RUP2_SW: Pull up resistor 2 switch control bit.

0 = open.

1 = close.

RUP1_SW: Pull up resistor 1 switch control bit.

0 = open.

1 = close.

19.8 USB Status Register(USTAT)

Mnemonic: USTAT **Address: F6h**

7	6	5	4	3	2	1	0	Reset
SPEED	PLL_FS[1:0]		ALT_VAL	INTF_VAL	CFG_VAL	EP0_DTYPE	SETUP	00H

SPEED: USB speed status.

0 = Low speed.

1 = Full speed.

PLL_FS[1:0]: PLL input reference clock status:

00: 6MHz (external crystal).

01: 12MHz (external crystal).

10: 24MHz (external crystal).

ALT_VALL: Current alternate value.

INTF_VAL: Current interface value.

CFG_VAL: Current configuration value.

EP0_DTYPE: EP0 data type indicator. This bit is used to indicate EP0 received pkt is IN pkt or OUT pkt.

0 = OUT data.

1 = IN data.

SETUP: SETUP pkt indicator. This bit is used to indicate EP0 received pkt is SETUP command or DATA.

0 = DATA pkt.

1 = SETUP pkt.

19.9 USB Device Address Register(DEVADR)

Mnemonic: DEVADR **Address: C4h**

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

DEVADR[6:0] Device address set by host.

19.10 USB Frame Number Register(FRMNUMH, FRMNUML)

Mnemonic: FRMNUMH **Address: C5h**

7	6	5	4	3	2	1	0	Reset
-					FRMNUM[10:8]			00H

Mnemonic: FRMNUML **Address: C6h**

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

19.11 USB Host Stall Register(HSTALL)

Mnemonic: HSTALL **Address: C7h**

7	6	5	4	3	2	1	0	Reset
-			EP4H STALL	EP3H STALL	EP2H STALL	EP1HS TALL	EP0 HSTALL	00H

EP4HSTALL: EP4 stall bit set by host using set feature command.

EP3HSTALL: EP3 stall bit set by host using set feature command.

EP2HSTALL: EP2 stall bit set by host using set feature command.

EP1HSTALL: EP1 stall bit set by host using set feature command.

EP0HSTALL: EP0 stall bit set by host using set feature command.

19.12 USB Device Stall Register(DSTALL)

Mnemonic: DSTALL **Address: BFh**

7	6	5	4	3	2	1	0	Reset
-			EP4D STALL	EP3D STALL	EP2D STALL	EP1D STALL	EP0D STALL	00H

EP4DSTALL: EP4 stall bit set by software when EP4 has any error.

0 = Endpoint 4 device work.

1 = Endpoint 4 device stall.

EP3DSTALL: EP3 stall bit set by software when EP3 has any error.

0 = Endpoint 3 device work.

1 = Endpoint 3 device stall.

EP2DSTALL: EP2 stall bit set by software when EP2 has any error.

0 = Endpoint 2 device work.

1 = Endpoint 2 device stall.

EP1DSTALL: EP1 stall bit set by software when EP1 has any error.

0 = Endpoint 1 device work.

1 = Endpoint 1 device stall.

EP0DSTALL: EP0 stall bit set by software when EP0 has any error.

0= Endpoint 0 device work.

1= Endpoint 0 device stall.

19.13 USB Handshake Status Register(HSKSTAT)

Mnemonic: HSKSTAT **Address: B1h**

7	6	5	4	3	2	1	0	Reset
-			EP4HSK	EP3HSK	EP2HSK	EP1HSK	EP0HSK	80H

EP4HSK USB Endpoint 4 Handshake status.

If the last handshake packet is STALL, Error in Data packet, time out on USB bus or NACK this bit will be set to ‘1’ automatically. If the last handshake packet is ACK, this bit will be cleared automatically. You can check this flag to know communication Status.

EP3HSK USB Endpoint 3 Handshake status.

If the last handshake packet is STALL, Error in Data packet, time out on USB bus or NACK this bit will be set to ‘1’ automatically. If the last handshake packet is ACK, this bit will be cleared automatically. You can check this flag to know communication Status.

EP2HSK: USB Endpoint 2 Handshake status.

If the last handshake packet is STALL, Error in Data packet, time out on USB bus or NACK this bit will be set to ‘1’ automatically. If the last handshake packet is ACK, this bit will be cleared automatically. You can check this flag to know communication Status.

EP1HSK: USB Endpoint 1 Handshake status.

If the last handshake packet is STALL, Error in Data packet, time out on USB bus or NACK this bit will be set to '1' automatically. If the last handshake packet is ACK, this bit will be cleared automatically. You can check this flag to know communication Status.

EP0HSK: USB Endpoint 0 Handshake status.

If the last handshake packet is STALL, Error in Data packet, time out on USB bus or NACK this bit will be set to '1' automatically. If the last handshake packet is ACK, this bit will be cleared automatically. You can check this flag to know communication Status.

19.14 USB Interrupt Enable Register 1(UIER1)

Mnemonic: UIER1 **Address: B2h**

7	6	5	4	3	2	1	0	Reset
-			EP4IE	EP3IE	EP2IE	EP1IE	EP0IE	00H

EP4IE: Endpoint 4 Interrupt Enable bit.

0 = Disable.

1 = Enable. This bit enables a local interrupt to be set when a USB Endpoint 4 data packet has been received by the MCU.

EP3IE: Endpoint 3 Interrupt Enable bit.

0 = Disable.

1 = Enable. This bit enables a local interrupt to be set when a USB Endpoint 3 data packet has been sent by the MCU.

EP2IE: Endpoint 2 Interrupt Enable bit.

0 = Disable.

1 = Enable. This bit enables a local interrupt to be set when a USB Endpoint 2 data packet has been received by the MCU.

EP1IE: Endpoint 1 Interrupt Enable bit.

0 = Disable.

1 = Enable. This bit enables a local interrupt to be set when the USB host has read the endpoint 1 register.

EP0IE: Endpoint 0 Interrupt Enable bit.

0 = Disable.

1 = Enable. This bit enables a local interrupt to be set when the USB Endpoint 0 received the class / vender command by the MCU.

19.15 USB Interrupt Enable Register 2(UIER2)

Mnemonic: UIER2 **Address: B3h**

7	6	5	4	3	2	1	0	Reset
RSUM IE	USBIE	-		CFGIE	SOFIE	SUSIE	RSTIE	00H

RSUMIE: Resume Interrupt Enable.

0 = Disable

1 = Enable

USBIE: All USB interrupt except resume enable bit.

0 = Disable

1 = Enable

CFGIE: Configuration change interrupt enable bit.

0 = Disable

1 = Enable

SOFIE: SOF(Start of Frame) interrupt enable bit.

0 = Disable

1 = Enable

SUSIE: Suspend interrupt enable bit.

0 = Disable

1 = Enable

RSTIE: USB reset interrupt enable bit.

0 = Disable

= Enable

19.16 USB Interrupt Flag Register 1(UIFR1)

Mnemonic: UIFR1 **Address: B4h**

7	6	5	4	3	2	1	0	Reset
-			EP4IF	EP3IF	EP2IF	EP1IF	EP0IF	00H

EP4IF: Endpoint 4 Interrupt Flag.

If set, this bit indicates when a USB Endpoint 4 data packet has been received by the MCU. This bit is cleared by the firmware.

EP3IF: Endpoint 3 Interrupt Flag.

If set, this bit indicates when a USB Endpoint 3 data packet has been sent by the MCU. This bit is cleared by the firmware.

EP2IF: Endpoint 2 Interrupt Flag.

If set, this bit indicates when a USB Endpoint 2 data packet has been received by the MCU. This bit is cleared by the firmware.

EP1IF: Endpoint 1 Interrupts Flag.

If set, this bit indicates when the USB host has read the Endpoint 1 registers. This bit is cleared by the firmware.

EP0IF: Endpoint 0 Interrupts Flag.

It will be set when the MCU receives Class / Vender command to endpoint 0. This bit is cleared by the firmware.

Note: When write “0”, these bit will be cleared.

When write “1”, these bit will not have any change.

19.17 USB Interrupt Flag Register 2(UIFR2)

Mnemonic: UIFR2 **Address: B5h**

7	6	5	4	3	2	1	0	Reset
RSUMIF	USBIF	-		CFGIF	SOFIF	SUSIF	RSTIF	00H

RSUMIF: Resume interrupt flag.

USBIF: All USB interrupt except resume interrupt flag.

CFGIF: Configuration change interrupt flag.

SOFIF: SOF(Start of Frame) interrupt flag.

SUSIF: Suspend interrupt flag.

RSTIF: USB reset interrupt flag.

Note: When write “0”, these bit will be cleared.
When write “1”, these bit will not have any change.

19.18 USB Endpoint Data Ready Register(EPDRDY)

Mnemonic: EPDRDY **Address: B6h**

7	6	5	4	3	2	1	0	Reset
RDYSW	-		EP4RDY	EP3TDY	EP2RDY	EP1TDY	EP0TRDY	2AH

RDYSW: EPDRDY write function switch.
0 = EPDRDY only can write “0”
1 = EPDRDY only can write “1”

EP4RDY: EP4 receive data FIFO ready.
0: SIE write EP4 FIFO.
1: MCU can read EP4 FIFO data.

EP3TDY: EP3 transmit data FIFO ready.
0: SIE read EP3 FIFO.
1: MCU can write EP3 FIFO data.

EP2RDY: EP2 receive data FIFO ready.
0: SIE write EP2 FIFO
1: MCU can read EP2 FIFO data.

EP1TDY: EP1 transmit data FIFO ready.
0: SIE read EP1 FIFO.
1: MCU can write EP1 FIFO data.

EP0TDRY: EP0 transmit / receive data FIFO ready.
0: SIE write/read EP0 FIFO
1: MCU can read/write EP0 FIFO.

19.19 USB Endpoint 0 Data Counter Register(EP0CNT)

Mnemonic: EP0CNT **Address: B7h**

7	6	5	4	3	2	1	0	Reset
-				EP0CNT[3:0]				00H

EP0CNT[3:0]: The EP0CNT is automatically incremented by 1 after the endpoint 0 receive FIFO register (EP0DATA) is written by SIE. The EP0CNT will count to 8H when it reaches the maximum count. The EP0CNT is automatically decreased by 1 after the MCU to read EP0DATA register. If EP0CNT = 00h, the Endpoint 0 FIFO is empty.

When endpoint 0 transmit mailbox (EP0DATA) data port is written by MCU, the pointer is automatically incremented by 1. The EP1CNT will count to 8H when it reaches the maximum count.

19.20 USB Endpoint 1 Data Counter Register(EP1CNT)

Mnemonic: EP1CNT **Address: A1h**

7	6	5	4	3	2	1	0	Reset
-				EP1CNT[3:0]				00H

EP1CNT[3:0]: This register determines which address of endpoint 1 transmit register is accessed when the Endpoint 1 transmit mailbox (EP1DATA) data port is written by MCU. This pointer is automatically incremented by 1 after the endpoint 1 transmit mailbox data port is written by MCU. The EP1CNT will count to 8H when it reaches the maximum count.

19.21 USB Endpoint 2 Data Counter Register(EP2CNT)

Mnemonic: EP2CNT **Address: A2h**

7	6	5	4	3	2	1	0	Reset
-				EP2CNT[3:0]				00H

EP2CNT[3:0]: The EP2CNT is automatically incremented by 1 after the endpoint 2 receive FIFO register (EP2DATA) is written by SIE. The EP2CNT will count to 8H when it reaches the maximum count. The EP2CNT is automatically decreased by 1 after the MCU to read EP2DATA register. If EP2CNT = 00h, the Endpoint 2 FIFO is empty.

19.22 USB Endpoint 3 Data Counter Register(EP3CNT)

Mnemonic: EP3CNT **Address: A3h**

7	6	5	4	3	2	1	0	Reset
-	EP3CNT[6:0]							00H

EP3CNT[6:0]: Endpoint 3 device write data counter register.

This register determines which address of endpoint 3 transmit register is accessed when the Endpoint 3 transmit FIFO (EP3DATA) data port is written by MCU. This EP3CNT is automatically incremented by 1 after the endpoint3 transmit FIFO data port is written by MCU. The EP3CNT pointer will count to 40H when it reaches the maximum count.

19.23 USB Endpoint 4 Data Counter Register(EP4CNT)

Mnemonic: EP4CNT **Address: A4h**

7	6	5	4	3	2	1	0	Reset
-	EP4CNT[6:0]							00H

EP4CNT[6:0]: The EP4CNT is automatically incremented by 1 after the endpoint 4 received FIFO register (EP4DATA) is written by SIE. The EP4CNT will count to 40H when it reaches the maximum count. The EP4CNT is automatically decreased by 1 after the MCU to read EP4DATA register. If EP4CNT = 00h, the Endpoint 4 FIFO is empty.

19.24 USB Endpoint 0 Data Register(EP0DATA)

Mnemonic: EP0DATA **Address: A7h**

7	6	5	4	3	2	1	0	Reset
EP0DATA[7:0]								00H

EP0DATA[7:0]: Endpoint 0 transmit/receive register.

This register is responsible to store the USB standard command from the USB host, or be written with the descriptor contents by MCU and waiting for SIE to fetch them. When USB host sends the 8-byte standard command to endpoint 0 FIFO, an interrupt (UIFR1 bit 0) should be generated to inform the MCU. When MCU need to send the descriptors via endpoint 0, the MCU must write the descriptors to this register.

19.25 USB Endpoint 1 Data Register(EP1DATA)

Mnemonic: EP1DATA **Address: 9Eh**

7	6	5	4	3	2	1	0	Reset
EP1DATA[7:0]								00H

EP1DATA[7:0]: Endpoint 1 transmits FIFO data register

This port is used to read or write one of the transmit mailbox registers. The register being accessed is selected by the endpoint 1 FIFO pointer register. The eight transmit registers are written by the MCU and are read by a USB interrupt transfer from endpoint 1. They can be used to pass messages from the MCU to the USB host. The format and content of the messages is user defined. If enabled, USB host reads from this register can generate a local interrupt.

Note: These registers are writing only during USB operation.

19.26 USB Endpoint 2 Data Register(EP2DATA)

Mnemonic: EP2DATA **Address: 9Fh**

7	6	5	4	3	2	1	0	Reset
EP2DATA[7:0]								00H

EP2DATA[7:0]: Endpoint 2 Received FIFO data register.

This register is used by the MCU to read data from the USB receive FIFO. The FIFO data

is written by the USB host using interrupt transfers to endpoint 2. The Endpoint 2 data FIFO is first – in – first – out.

Note: These registers are reading only during USB operation.

19.27 USB Endpoint 3 Data Register(EP3DATA)

Mnemonic: EP3DATA **Address: 93h**

7	6	5	4	3	2	1	0	Reset
EP3DATA[7:0]								00H

EP3DATA[7:0]: Endpoint 3 Transmit FIFO data register

This register is used by the MCU to write data to the transmit FIFO. The FIFO is read by the USB host using bulk transfers from endpoint 3. The Endpoint 3 data FIFO is first – in – first – out.

Note: These registers are writing only during USB operation.

19.28 USB Endpoint 4 Data Register(EP4DATA)

Mnemonic: EP4DATA **Address: 94h**

7	6	5	4	3	2	1	0	Reset
EP4DATA[7:0]								00H

EP4DATA[7:0]: Endpoint 4 received FIFO data register.

This register is used by the MCU to read data from the USB receive FIFO. The FIFO is written by the USB host using bulk transfers to endpoint 4. The Endpoint 4 data FIFO is first – in – first – out.

Note: These registers are reading only during USB operation.

20. Barcode

Barcode Module Features:

- ADC values store to SRAM directly by DMA.
- ADC converts to barcode raw data for barcode decoding.
- Barcode raw data filter.
- Barcode raw data slope rate setting.

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RDT
Barcode											
BCCTRL	BarCode Control Register	CAh	Bcc_en[1:0]		BarcodeINF[1:0]		RAWBITNUM	ADCDS[1:0]		bcdidv	01H
ADDR2ML	Start address to SRAM Low Byte Register	CBh	ADDR2M[7:0]								00H
ADDR2MH	Start address to SRAM High Byte Register	CEh	-			ADDR2M[12:8]					00H
LNGDATAL	Length of Data Low Byte Register	EFh	LNGDATA[7:0]								00H
LNGDATAH	Length of Data High Byte Register	DAh	-			LNGDATA[12:8]					00H
RDATA	Rising of Data Register	C1h	RDATA[7:0]								19H
FDATA	Falling of Data Register	C2h	FDATA[7:0]								18H

20.1 Barcode Control Register(BCCTRL)

Mnemonic: BCCTRL **Address: CAh**

7	6	5	4	3	2	1	0	Reset
Bcc_en[1:0]		BarcodeINF[1:0]		RAWBITNUM	ADCDS[1:0]		bcdidv	01H

Bcc_en Barcode control mode

[1:0] If ADC to DMA reach SRAM's address = 6K, HW will auto stop "write action" and this bit will be clear to 0 when barcode mode enabled.

00 = disable barcode controller

01 = Enable ADC value store to SRAM function, write ADC value into SRAM directly.

10 = Enable barcode decoder. [Decode the 8 bits of ADC data (MSB)]

Analysis/count barcode raw data then write the results into SRAM.

The results data format as following:

b7 : convert data

b[6:0]: length

ex: 1000_0111 means 7 successive "1".

1000_1000 means 8 successive "1".

1000_0000 means 128 successive “1”

[※Greater than 128 will continue to increase])

11 = Enable barcode decoder. [Decode the 8 bits of ADC data (MSB)]

Analysis/count barcode raw data then write the results into SRAM.

The results data format as following:

b7 : convert data

b[6:0]: length

ex: 1000_0111 means 7 successive “1”.

1000_1000 means 8 successive “1”.

1000_0000 means greater then 128 successive “1”

[※Will not increase more than 128])

BarcodeINF[1:0] Barcode Input Noise Filter

00 = 1 consecutive same value recognize as valid data.

01 = 2 consecutive same value recognize as valid data.

10 = 3 consecutive same value recognize as valid data.

11 = 4 consecutive same value recognize as valid data.

RAWBITNUM Raw Data Bit Number

0 = 10 bit; HW write raw data 10 bit into SRAM.

1 = 8 bit; HW write raw data 8 bit (MSB) into SRAM.

ADCDS The ADC of data select

[1:0] 00 = Compared with the previous first ADC of data.

01 = Compared with the previous second ADC of data.

10 = Compared with the previous third ADC of data.

11 = Compared with the previous fourth ADC of data.

bcdidv Initial digital value of Barcode decoder

bcdidv = 1 after system reset

20.2 Start Address to SRAM Register(ADDR2ML, ADDR2MH)

Mnemonic: ADDR2ML **Address: CBh**

7	6	5	4	3	2	1	0	Reset
ADDR2M[7:0]								00H

Mnemonic: ADDR2MH **Address: CEh**

7	6	5	4	3	2	1	0	Reset
-			ADDR2M[12:8]					00H

ADDR2M: Starting address of data written to SRAM;

[12:0] The content of this register will not be updated when barcode controller write data to SRAM.

data written to the address of SRAM arbitrary.

20.3 Length of Data Register(LNGDATAL, LNGDATAH)

Barcode raw data length when data written to SRAM completed. (Read Only)

Mnemonic: LNGDATAL **Address: EFh**

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

Mnemonic: LNGDATAH **Address: DAh**

7	6	5	4	3	2	1	0	Reset
-			LNGDATA[12:8]					00H

20.4 Rising of Data Register(RDATA)

Mnemonic: RDATA **Address: C1h**

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

RDATA[7:0] The rising of data of slope rate

20.5 Falling of Data Register(FDATA)

Mnemonic: FDATA **Address: C2h**

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

FDATA[7:0] The falling of data of slope rate

21. In-System Programming (Internal ISP)

The SM59A16U1 can generate flash control signal by internal hardware circuit. Users utilize flash control register, flash address register and flash data register to perform the ISP function without removing the SM59A16U1 from the system. The SM59A16U1 provides internal flash control signals which can do flash program/chip erase/page erase/protect functions. User need to design and use any kind of interface which SM59A16U1 can input data. User then utilize ISP service program to perform the flash program/chip erase/page erase/protect functions.

21.1 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 SM59A16U1 for the ISP purpose.

The ISP service programs were developed by user so that it should includes any features which relates to the flash memory programming function as well as communication protocol between SM59A16U1 and host device which output data to the SM59A16U1. For example, if user utilize UART interface to receive/transmit data between SM59A16U1 and host device, the ISP service program should include baud rate, checksum or parity check or any error-checking mechaniOB to avoid data transmission error.

The ISP service program can be initiated under SM59A16U1 active or idle mode. It can not be initiated under power down mode.

21.2 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 flash erase function.

The ISP service program spaces address range 0xF000h to 0xFFFFh. It can be divided as blocks of N*256 byte. (N=0 to 16). When N=0 means no ISP function, all of 64K byte flash memory can be used as program memory. When N=1 means ISP service program occupies 256 byte while the rest of 63.75K byte flash memory can be used as program memory. The maximum ISP service program allowed is 4K byte when N=16. Under such configuration, the usable program memory space is 60K byte.

After N determined, SM59A16U1 will reserve the ISP service program space downward from the top of the program address 0xFFFFh. The start address of the ISP service program located at 0xFx00h, x is depends on the lock bit N. As given in Table 21-1.

The lock bit N function is different from the flash protect function. The flash erase function can erase all of the flash memory except for the locked ISP service program space. If the flash not has been protected, the content of ISP service program still can be read. If the flash has been protected, the overall content of flash program memory space including ISP service program space cannot be read.

Table 21-1 ISP code area

	ISP service program address
0	No ISP service program
1	256 bytes (0xFF00h ~ 0xFFFFh)
2	512 bytes (0xFE00h ~ 0xFFFFh)
3	768 bytes (0xFD00h ~ 0xFFFFh)
4	1.0 K bytes (0xFC00h ~ 0xFFFFh)
5	1.25 K bytes (0xFB00h ~ 0xFFFFh)
6	1.5 K bytes (0xFA00h ~ 0xFFFFh)
7	1.75 K bytes (0xF900h ~ 0xFFFFh)
8	2.0 K bytes (0xF800h ~ 0xFFFFh)
9	2.25 K bytes (0xF700h ~ 0xFFFFh)
10	2.5 K bytes (0xF600h ~ 0xFFFFh)
11	2.75 K bytes (0xF500h ~ 0xFFFFh)
12	3.0 K bytes (0xF400h ~ 0xFFFFh)
13	3.25 K bytes (0xF300h ~ 0xFFFFh)
14	3.5 K bytes (0xF200h ~ 0xFFFFh)
15	3.75 K bytes (0xF100h ~ 0xFFFFh)
16	4.0 K bytes (0xF000h ~ 0xFFFFh)

ISP service program configurable in N*256 byte (N= 0 ~ 16)

21.3 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/erase timing. It is different from the flash memory program/erase timing so the locked ISP service program can not be erased by flash erase function. 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 SM59A16U1 was in system.

21.4 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 flash address blank (0x0000h = 0xFFH) will load the PC with start address of ISP service program. The hardware reset includes Internal (power on reset) and external pad reset.

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

(3) Enters ISP service program by hardware setting. User can force SM59A16U1 enter ISP service program by setting P3.4 “active low” during hardware reset period. The hardware reset includes Internal (power on reset) and external pad reset. In application system design, user should take care of the setting of P3.4 at reset period to prevent SM59A16U1 from entering ISP service program.

(4) Enter’s ISP service program by UART setting, the RXD0 received 0x55h data (baud rate: 57600bps) during hardware reset period. The hardware reset includes internal (power on reset) and external pad reset.

(5) Enter’s ISP service program by USB setting, the USB setup command (Endpoint 0) received sequence data: 0x40, 0x20, 0x5A, 0xA5, 0x69, 0x96, 0x02, 0x00 during hardware reset period. The hardware reset includes internal (power on reset) and external pad reset.

During hardware reset period, if they meet one of above conditions, chip will switch to ISP mode automatically. After ISP service program executed, user need to reset the SM59A16U1, either by hardware reset or by WDT, or jump to the address 0x0000h to re-start the firmware program.

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

(1) First Address Blank. i.e. 0x0000h = 0xFF. And triggered by Internal reset signal.
(2) First Address Blank. i.e. 0x0000h = 0xFF. And triggered by PAD reset signal.
(3) P3.4 = 0. And triggered by Internal reset signal.
(4) P3.4 = 0. And triggered by PAD reset signal.
(5) RXD0 received 0x55 data (baud rate is 57600bps). And triggered by Internal reset signal.
(6) RXD0 received 0x55 data (baud rate is 57600bps). And triggered by PAD reset signal.
(7) USB Endpoint 0 received sequence data: 0x40, 0x20, 0x5A, 0xA5, 0x69, 0x96, 0x02, 0x00. And triggered by Internal reset signal.
(8) USB Endpoint 0 received sequence data: 0x40, 0x20, 0x5A, 0xA5, 0x69, 0x96, 0x02, 0x00. And triggered by Internal reset signal.

21.5 ISP register – TAKEY, IFCON, ISPFAH, ISPFAL, ISPFD 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	ITS	CDPR	F32K	F16K	ALEC. 1	ALEC. 0	EMEN	ISPE	00H
ISPFAH	ISP Flash Address – High register	E1h	ISPFAH [7:0]								FFH
ISPFAL	ISP Flash Address - Low register	E2h	ISPFAL [7:0]								FFH
ISPFD	ISP Flash Data register	E3h	ISPFD [7:0]								FFH
ISPFC	ISP Flash Control register	E4h	EMF1	EMF2	EMF3	EMF4	-	ISPF[2:0]			00H

21.6 Time Access Key Register(TAKEY)

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
```

21.7 Interface Control Register(IFCON)

Mnemonic: IFCON **Address: 8FH**

7	6	5	4	3	2	1	0	Reset
ITS	CDPR	F32K	F16K	ALEC[1]	ALEC[0]	EMEN	ISPE	00H

The bit 0 (ISPE) of IFCON is ISP enable bit. User can enable overall SM59A16U1 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 erased accidentally. ISP registers ISPFAH, ISPFAL, ISPFD and ISPFC are read-only by default. Software must be set ISPE bit to 1 to enable these 4 registers write attribute.

21.8 ISP Flash Address Register(ISPFAH, ISPFAL)

Mnemonic: ISPFAH **Address: E1H**

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

ISPFAH [7:0]: Flash 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]: Flash address-Low for ISP function

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

21.9 ISP Flash Data Register(ISPFD)

Mnemonic: ISPFD **Address: E3H**

7	6	5	4	3	2	1	0	Reset
ISPFD7	ISPFD6	ISPFD5	ISPFD4	ISPFD3	ISPFD2	ISPFD1	ISPFD0	FFH

ISPFD [7:0]: Flash data for ISP function.

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

21.10 ISP Flash Control Register(ISPFC)

Mnemonic: ISPFC **Address: E4H**

7	6	5	4	3	2	1	0	Reset
EMF1	EMF2	EMF3	EMF4	-	ISPF[2]	ISPF[1]	ISPF[0]	00H

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

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

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

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

ISPF[2:0]	ISP function
000	Byte program
001	Chip protect
010	Page erase
011	Chip erase
100	Write option
101	Read option
110	Erase option
111	reserved

One page of flash memory is 128byte

When chip protected or no ISP service, option can only read.

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

To perform byte program/page erases ISP function, user need to specify flash address at first. When performing page erase function, SM59A16U1 will erase entire page which flash address indicated by ISPFAH & ISPFAL registers located within the page.

e.g. flash address: $ XYMN

page erase function will erase from $XY00 to $XYFF

To perform the chip erase ISP function, SM59A16U1 will erase all the flash program memory except the ISP service program space. To perform chip protect ISP function, the SM59A16U1 flash memory content will be read #00H.

e.g. ISP service program to do the byte program - to program #22H to the address $1005H

```
MOV TAKEY, #55h
MOV TAKEY, #0AAh
MOV TAKEY, #5Ah    ; enable ISPE write attribute
ORL IFCON, #01H    ; enable SM39R08A3 ISP function
MOV ISPFAH, #10H   ; set flash address-high, 10H
MOV ISPFAL, #05H   ; set flash address-low, 05H
MOV ISPFD, #22H    ; set flash data to be programmed, data = 22H
MOV ISPFC, #00H    ; start to program #22H to the flash address $1005H
MOV TAKEY, #55h
MOV TAKEY, #0AAh
MOV TAKEY, #5Ah    ; enable ISPE write attribute
ANL IFCON, #0FEH   ; disable SM39R08A3 ISP function
```

22. OPA/Comparator

SM59A16U1 had integrated an OPA/Comparator module on chip. This module supports OPA and Comparator modes individually according to user's configuration. When OPA Mode enabled, dual OP-Amps may be applied to single or two-stage amplifier network, and may be applied as a front-end signal process and internally routed to specific ADC channel. When Comparator Mode enabled, an internal reference voltage is available to be configured on comparator terminals. As shown in Fig. 22-1.

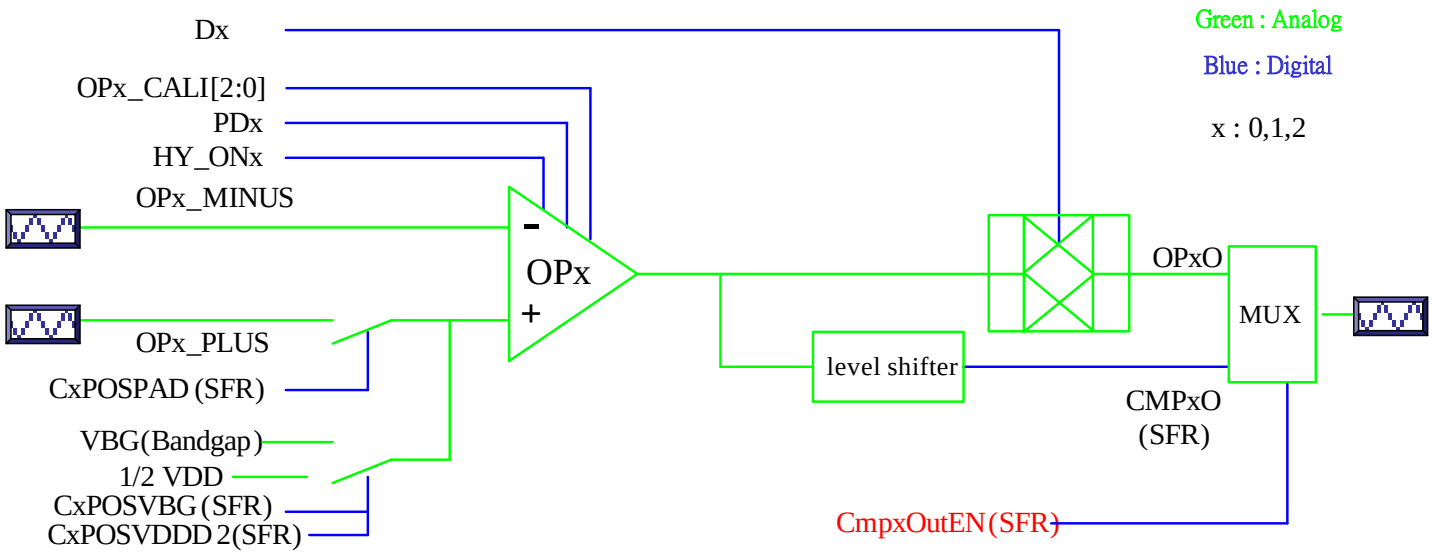


Fig. 22-1: Operation of Comparator Mode

If OPA and Comparator Mode both are enabled at same module, the OPA Mode has higher priority.

The Comparator interrupt vector is 93h.

The OPA/Comparator SFR show as below:

Mnemonic	Description	Dir.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	RST
Comparator											
OpPin	OpCmp Pin Select	F6h	Op0 En	Cmp0E n	C0Pos VBG	C0Pos PAD	Op1 En	Cmp1En	C1Pos VBG	C1Pos PAD	00h
OpPin2	OpCmp Pin Select 2	CEh	-					C1Pos Vddd2	C0Pos Vddd2	-	00h
Cmp0CON	Comparator_ 0 control	FEh	Hys0 En	Cmp0o	CMF0 MS[1:0]	CMF0	Cmp 0 Out EN	Hys0En	-		00h
Cmp1CON	Comparator_ 1 control	FFh	Hys1 En	Cmp1o	CMF1 MS[1:0]	CMF1	Cmp 1 Out EN	Hys1En	-		00h

OpxEn, CmpxEn, CxPosVBG, CxPosPad and CxPosVdd2 setting table.

OpxEn	CmpxEn	CxPos VBG	CxPos Pad	CxPos Vddd2	OP/Comparator	positive input source
0	0	X	X	X	GPIO	N/A
0	1	0	0	0	Not Allowed	N/A
0	1	0	0	1	Comparator	Reference 1/2 Vdd Voltage
0	1	0	1	0	Comparator	Positive pin input voltage
0	1	0	1	1	Not Allowed	N/A
0	1	1	0	0	Comparator	Internal Reference Voltage
0	1	1	0	1	Not Allowed	N/A
0	1	1	1	0	Not Allowed	N/A
0	1	1	1	1	Not Allowed	N/A
1	X	0	0	0	Not Allowed	N/A
1	X	0	0	1	OP	Reference 1/2 Vdd Voltage
1	X	0	1	0	OP	Positive pin input voltage
1	X	0	1	1	OP	OPxPIn output 1/2 Vdd Voltage
1	X	1	0	0	OP	Internal Reference Voltage
1	X	1	0	1	Not Allowed	N/A
1	X	1	1	0	OP	OPxPIn ouput Internal Reference Voltage
1	X	1	1	1	Not Allowed	N/A

Note: “X” Don’t care.

22.1 Op/Comparator Pin Select(OpPin)

Mnemonic: OpPin **Address: F6h**

7	6	5	4	3	2	1	0	Reset
Op0En	Cmp0En	C0Pos VBG	C0Pos PAD	Op1En	Cmp1En	C1Pos VBG	C1Pos PAD	00H

Op0En: Op0 enable function.

0 = Op0 circuit disable.

1 = Op0 circuit enable and switch to corresponding signal in multi-function pin P2.5/P2.6/P2.7 by HW automatically.

Cmp0En: Cmp0 enable function.

0 = Comparator_0 circuit disable.

1 = Comparator_0 circuit enable and switch to corresponding signal in multi-function pin P2.5/P2.6/P2.7 by HW automatically.

C0PosVBG: Enable Comparator_0 positive input source as internal reference voltage. (1.2V±10%)

0 = Disable positive input source as internal reference voltage.

1 = Enable positive input source as internal reference voltage.

C0PosPAD: Enable Comparator_0 positive input source as external pin.

0 = Disable positive input source as external pin.

1 = Enable positive input source as external pin.

Op1En: Op1 enable function.

0 = Op1 circuit disable.

1 = Op1 circuit enable and switch to corresponding signal in multi-function pin P2.2/P2.3/P2.4 by HW automatically.

Cmp1En: Cmp1 enable function.

0 = Comparator_1 circuit disable.

1 = Comparator_1 circuit enable and switch to corresponding signal in multi-function pin P2.2/P2.3/P2.4 by HW automatically.

C1PosVBG: Enable Comparator_1 positive input source as internal reference voltage. (1.2V±10%)

0 = Disable positive input source as internal reference voltage.

1 = Enable positive input source as internal reference voltage.

C1PosPAD: Enable Comparator _1 positive input source as external pin.

0 = Disable positive input source as external pin.

1 = Enable positive input source as external pin.

22.2 Op/Comparator Pin Select 2(OpPin2)

Mnemonic: OpPin2 **Address: CEh**

7	6	5	4	3	2	1	0	Reset
-					C1PosVddd2	C0PosVddd2	-	00H

C1PosVddd2: Enable Comparator_1 positive input source as Vdd divide 2.(1/2 Vdd)

0 = Disable positive input source as Vdd divide 2.

1 = Enable positive input source as Vdd divide 2.

C0PosVddd2: Enable Comparator_0 positive input source as Vdd divide 2.(1/2 Vdd)

0 = Disable positive input source as Vdd divide 2.

1 = Enable positive input source as Vdd divide 2.

22.3 Comparator 0 Control(Cmp0CON)

Mnemonic: Cmp0CON **Address: FEh**

7	6	5	4	3	2	1	0	Reset
Hys0En	Cmp0o	CMF0MS[1:0]		CMF0	Cmp0OutEN	-		00H

Hys0En: Comparator_0 hysteresis function enable

0 = Disable hysteresis at comparator_0 input

1 = Enable hysteresis at comparator_0 input

Cmp0o: Comparator_0 output. (read only)

0 = The positive input source was lower then negative input source

1 = The positive input source was higher then negative input source

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

00: CMF0(comprator_0 flag) will be set when comprator_0 output toggle

01: CMF0(comprator_0 flag) will be set when comprator_0 output rising

10: CMF0(comprator_0 flag) will be set when comprator_0 output falling

11: reserved

CMF0: Comparator_0 Flag.

This bit is set by hardware according to CMF0MS [1:0] and must be clear by software.

Cmp0OutEN: Comparator_0 output enable.

0 = Comparator_0 will not output to external pin.

1 = Comparator_0 will output to external pin.

22.4 Comparator 1 Control(Cmp1CON)

Mnemonic: Cmp1CON **Address: FFh**

7	6	5	4	3	2	1	0	Reset
Hys1En	Cmp1o	CMF1MS[1:0]		CMF1	Cmp1OutEN	-		00H

Hys1En: Comparator_1 hysteresis function enable

0 = Disable hysteresis at comparator_1 input

1 = Enable hysteresis at comparator_1 input

Cmp1o: Comparator_1 output. (read only)

0 = The positive input source was lower then negative input source

1 = The positive input source was higher then negative input source

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

00: CMF1(comprator_1 flag) will be set when comprator_1 output toggle

01: CMF1(comprator_1 flag) will be set when comprator_1 output rising

10: CMF1(comprator_1 flag) will be set when comprator_1 output falling

11: reserved

CMF1: Comparator_1 Flag.

This bit is set by hardware according to CMF1MS [1:0] and must be clear by software.

Cmp1OutEN: Comparator_1 output enable.

0 = Comparator_1 will not output to external pin.

1 = Comparator_1 will output to external pin.

Operating Conditions

Symbol	Description	Min.	Typ.	Max.	Unit.	Remarks
TA	Operating temperature	-40	25	85	℃	Ambient temperature under bias
VDD	Supply voltage	2.2	-	5.5	V	
Vref	Internal reference voltage	1.1	1.2	1.3	V	

DC Characteristics

TA = -40℃ to 85℃, VCC = 5.0V

Symbol	Parameter	Valid	Min	Typical	Max	Units	Conditions
VIL1	Input Low-voltage	Port 0,1,2,3,4 RESET	-	-	0.3Vdd	V	
VIL2	Input Low-voltage	XTAL1	-	-	0.2Vdd	V	
VIH1	Input High-voltage	Port 0,1,2,3,4 RESET	0.7Vdd	-	-	V	
VIH2	Input High-voltage	XTAL1	0.8Vdd	-	-	V	
Vhys	Hysteresis voltage	RESET	-	0.6		V	
VOL	Output Low-voltage	Port 0,1,2,3,4			0.4	V	IOL= 5.5mA
VOH1	Output High-voltage using Strong Pull-up(1)	Port 0,1,2,3,4	2.6V	-	-	V	IOH= -4.3mA
VOH2	Output High-voltage using Weak Pull-up(2)	Port 0,1,2,3,4	2.6V	-	-	V	IOH= -100uA
IIL	Logic 0 Input Current	Port 0,1,2,3,4	-	-	-75	uA	Vin= 0.45V
ITL	Logical Transition Current	Port 0,1,2,3,4	-	-	-650	uA	Vin= 2.0V
ILI	Input Leakage Current	Port 0,1,2,3,4	-	-	±10	uA	0.45V<Vin<Vcc
RRST	Reset Pull-down Resistor	RES	50	-	300	kΩ	
CIO	Pin Capacitance	-	-	-	10	pF	Freq= 1MHz, Ta= 25℃
ICC	Power Supply Current	VDD	-	16	25	mA	Active mode ,IRC=22.1184MHz
			-	10	15	mA	Active mode, 12MHz VCC =5V 25 ℃
			-	9	14	mA	Idle mode, 12MHz VCC =5V 25 ℃
			-	3	9	uA	Power down mode VCC =5V 25 ℃

Notes:
(1) Port in Push-Pull Output Mode
(2) Port in Quasi-Bidirectional Mode
(3) To Be Defined

TA = -40℃ to 85℃, VCC = 3.0V

Symbol	Parameter	Valid	Min	Typical	Max	Units	Conditions
VIL1	Input Low-voltage	Port 0,1,2,3,4 RESET	-	-	0.3Vdd	V	
VIL2	Input Low-voltage	XTAL1	-	-	0.2Vdd	V	
VIH1	Input High-voltage	Port 0,1,2,3,4 RESET	0.7Vdd	-	-	V	
VIH2	Input High-voltage	XTAL1	0.8Vdd	-	-	V	
Vhys	Hysteresis voltage	RESET	-	0.6		V	
VOL	Output Low-voltage	Port 0,1,2,3,4	-	-	0.4	V	IOL= 5.5mA
VOH1	Output High-voltage using Strong Pull-up(1)	Port 0,1,2,3,4	2.6V	-	-	V	IOH= -4.3mA
VOH2	Output High-voltage using Weak Pull-up(2)	Port 0,1,2,3,4	2.6V	-	-	V	IOH= -100uA
IIL	Logic 0 Input Current	Port 0,1,2,3,4	-	-	-75	uA	Vin= 0.45V
ITL	Logical Transition Current	Port 0,1,2,3,4	-	-	-650	uA	Vin= 2.0V
ILI	Input Leakage Current	Port 0,1,2,3,4	-	-	±10	uA	0.45V<Vin<Vcc
RRST	Reset Pull-down Resistor	RES	50	-	300	kΩ	
CIO	Pin Capacitance	-	-	-	10	pF	Freq= 1MHz, Ta= 25℃
ICC	Power Supply Current	VDD	-	12	18	mA	Active mode ,IRC=22.1184MHz
			-	5	10	mA	Active mode ,12MHz VCC = 3.0 V 25 ℃
			-	4	9	mA	Idle mode, 12MHz VCC =3.0V 25 ℃
			-	3	9	uA	Power down mode VCC =3.0V 25 ℃

Notes:

(1) Port in Push-Pull Output Mode

(2) Port in Quasi-Bidirectional Mode

(3) To Be Defined

Absolute Maximum Ratings

SYMBOL	PARAMETER	MAX	UNIT
Maximum sourced current	An I/O pin	N/A	mA
	Total I/O pins	150	mA
Maximum sink current	An I/O pin	N/A	mA
	Total I/O pins	150	mA
Tj	Max. Junction Temperature	150	℃

OPA Characteristics

Parameters	Min	Typ	Max	Units
Supply voltage/Operating voltage	2.5	-	5.5	V
Operating current (single Op)	-	-	200	uA
CMRR	60	-	-	dB
PSRR	60	-	-	dB
Input offset voltage	-	5	-	mV
Gain bandwidth product	-	500	-	KHz
Phase marge	-	55	-	
Slew rate(V/us)	-	0.03	-	V/us
MAX. load	-	10K Ohm 100 pF	-	
Output source current	-	500	-	uA
Output sink current	-	500	-	uA

Comparator Characteristics

Ta=25℃

Symbol	Description	Test Condition		MIN	TPY	MAX	Unit
		V_{DD}	Condition				
I_{OP}	Operating current	5	-	-	10	10	uA
-	Power Down Current	5	-	-	-	0.1	uA
-	Offset voltage	5	-	-10	-	+10	mV
V_{CM}	Input voltage commom mode range	-	-	Vss	-	Vdd-1.5	V
Tp	Propagation delay	5	△ Vin=10mV	-	3	6	us
	Hysteresis			-	+/- 20	-	mV

LVR (Low Voltage Reset) Characteristics

LVR detect voltage range

Reset Function	Min	Typical	Max
LVRE	VIL=1.52V	VIL=1.60V	VIL=1.68V
LVRLPE	VIH=0.92V	VIH=1.0V	VIH=1.08V

LVI (Low Voltage Interrupt) Characteristics

LVI detect voltage range

	Min	Typical	Max
LVIS[1:0] = 00	VIL=1.66V VIH=1.86V	VIL=1.75V (VIH=1.95V)	VIL=1.83V (VIH=2.03V)
LVIS[1:0] = 01	VIL=2.61V VIH=2.81V	VIL=2.75V VIH=2.95V	VIL=2.88V (VIH=3.08V
LVIS[1:0] = 10	VIL=3.18V VIH=3.38V	VIL=3.35V VIH=3.55V	VIL=3.51V VIH=3.71V
LVIS[1:0] = 11	VIL=3.99V VIH=4.19V	VIL=4.20V VIH=4.40V	VIL=4.41V VIH=4.61V