



## SH68F093

### 8-Bit Microcontroller

---

#### Features

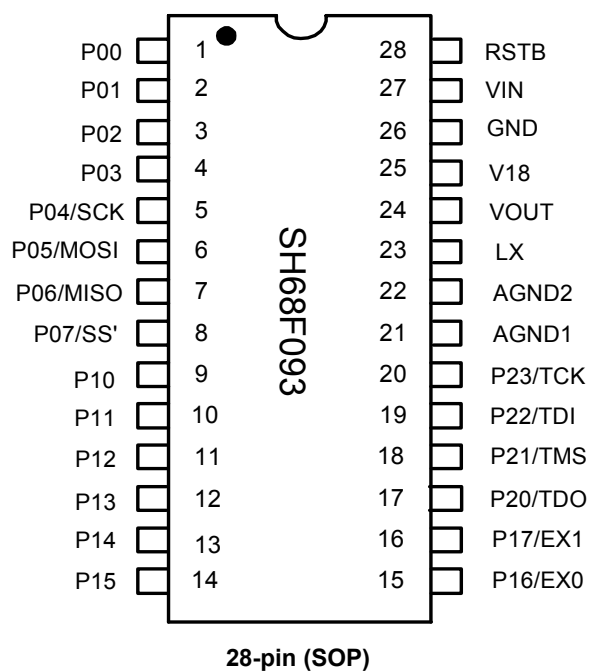
- CMOS technology for low power consumption
- Power Supply
  - DC-DC Input voltage ranges from 1.1V to 1.7V (static voltage without load)
  - Fixed DC-DC output voltage: 2.7V or 3.3V
  - Built-in 1.8V regulator for Micro-Processor ( $\mu$ P)
- 8-bit CMOS Micro-Processor ( $\mu$ P) core
  - Instruction set is fully compatible with standard 8051
- Program Memory
  - 16K Bytes Program Memory (flash ROM)
- Data Memory
  - 512-byte DATA Memory: internal 256-byte RAM and external 256-byte RAM
- Interrupts
  - 8 vector interrupt structure: two external interrupt, two base timer, resume Interrupt, wake-up Interrupt, OVL interrupt, and SPI interrupt
  - Two programmable priority levels
- On-Chip programmable RC Oscillator
  - System Clock: 8MHz/4MHz/2MHz/1MHz
  - Wake-up Timer: 8MHz/4MHz/2MHz/1MHz @IDLE mode
  - Wake-up Timer: 32KHz/16KHz @Power down mode
- Operation frequency: Programmable
- Two 8-Bit auto-reloadable base timers
- One programmable voltage comparator
- Reset
  - Hardware reset: Low-Voltage Reset, Power-on reset, and External Reset
  - Watch-dog Reset
  - Resume Reset
- General Purpose I/O
  - Up to 20 Selectable GPIO on 28 pin SOP package
- Master/Slave SPI serial interface
- Package
  - 28 pin SOP, CHIPFORM

#### General Description

SH68F093, an 8-bit micro-controller, is designed for the high-performance and low-power RF mouse application. The micro-controller contains on-chip flash-type program 16K bytes ROM, internal 512 bytes RAM, two 8-bit base timers, a wake-up timer, a watch-dog timer, a DC-DC converter, a 1.8V regulator for  $\mu$ P core, built-in 8MHz RC resonator, POR/LVR reset, resume reset, programmable voltage comparator, and master/slave SPI interface.

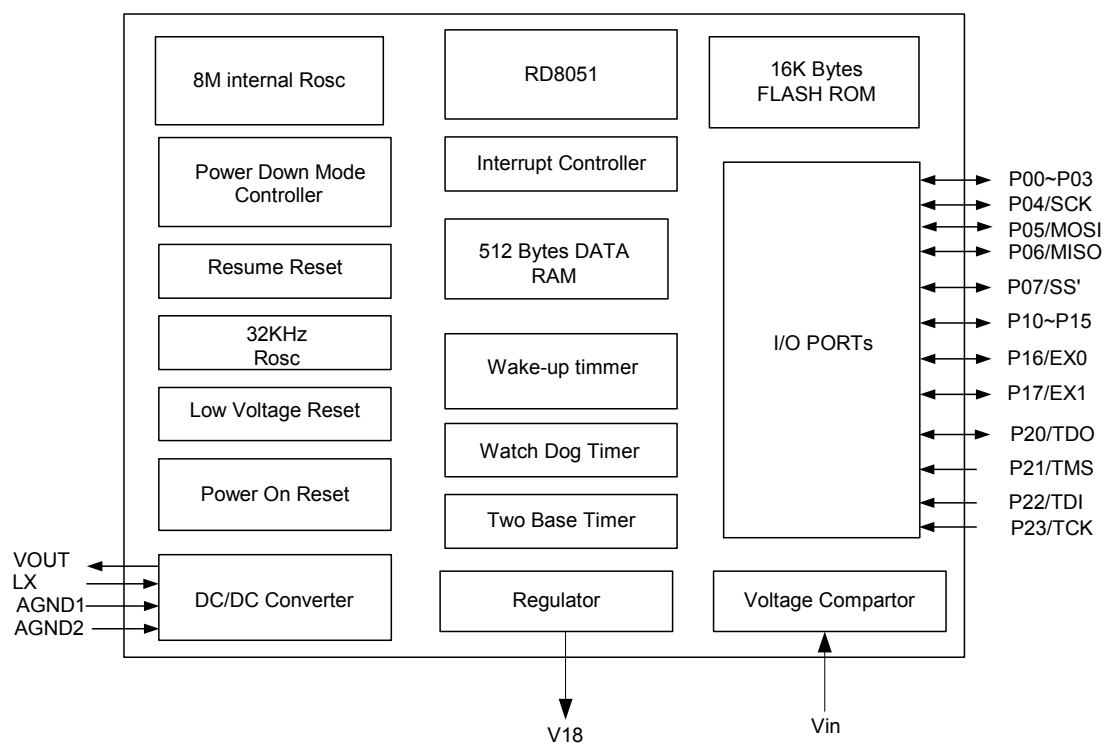


## Pin Configurations





## Block Diagram




**Pin and Pad Descriptions**

Package Pin No. 28pin SOP	Pad No.	Pad Name	I/O	Description
1	1	P00	I/O	Bi-directional I/O pin
2	2	P01	I/O	Bi-directional I/O pin
3	3	P02	I/O	Bi-directional I/O pin
4	4	P03	I/O	Bi-directional I/O pin
5	5	P04 SCK	I/O I/O	Bi-directional I/O pin SPI serial clock
6	6	P05 MOSI	I/O I/O	Bi-directional I/O pin SPI master output slave input
7	7	P06 MISO	I/O I/O	Bi-directional I/O pin SPI master input slave output
8	8	P07 /SS	I/O I	Bi-directional I/O pin SPI Slave Select
9	9	P10	I/O	Bi-directional I/O pin
10	10	P11	I/O	Bi-directional I/O pin
11	11	P12	I/O	Bi-directional I/O pin
12	12	P13	I/O	Bi-directional I/O pin
13	14	P14	I/O	Bi-directional I/O pin
14	15	P15	I/O	Bi-directional I/O pin
15	16	P16 EX0	I/O I	Bi-directional I/O pin External INT0
16	17	P17 EX1	I/O I	Bi-directional I/O pin External INT1
17	18	P20 TDO	I/O O	Bi-directional I/O pin JTAG DATA output pin
18	19	P21 TMS	I I	Uni-directional input pin JTAG MODE select pin
19	20	P22 TDI	I I	Uni-directional input pin JTAG DATA input pin
20	21	P23 TCK	I I	Uni-directional Input pin JTAG clock input pin
21	22/23	AGND1	I	Analog Ground
22	24/25	AGND2	I	Analog Ground
23	26/27	LX	I	DC-DC Switching pin
24	28/29/30	VOOUT/VDDIN	I/O	DC-DC Voltage output pin
25	31/32	V18/VCC	I/O	Regulator pin
26	33/34	GND	I	Ground
27	35	VIN	I	Voltage Comparator
28	36	RSTB	I	Reset pin



## Functional Description

### 1. Memory

#### 1.1. Memory Allocation

There are 16K bytes Program Memory and 512 bytes Data Memory.

#### 1.2. Program Memory

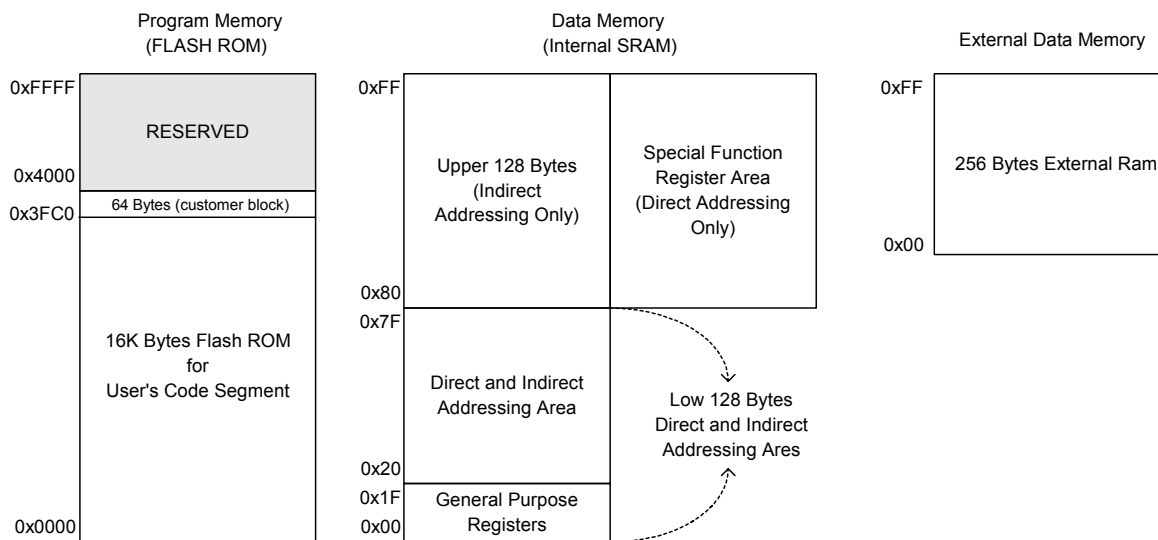
SH68F093 embeds 16K Bytes (0000H - 3FFFFH) on-chip program memory for program code. The flash program memory provides electrical erasure and programming.

**Note:** The 0<sup>th</sup> sector (0000H - 03FFFH) and the last 64 bytes for customer block (3FC0H - 3FFFFH) are reserved and can't be used as program memory.

#### 1.3. Data Memory

SH68F093 provides internal 256-byte RAM and external 256-byte RAM.

Internal Special Function Registers (SFR): There are 128 bytes SFR, which is the internal reserved memory for system registers in the  $\mu$ P.



**FIG. 1-1 SH68F093 Program/Data Memory Map**

**Note:** SH68F093 provides traditional method for accessing of external RAM. Use MOVX A, @Ri or MOVX @Ri, A to access 256-byte external RAM.



## 1.4. Registers

System Registers											
Address	Name	Initial Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00E0H	ACC	00H	R/W	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0
00F0H	B	00H	R/W	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0
00D0H	PSW	00H	R/W	CY	AC	F0	RS1	RS0	OV	0	P
0081H	SP	07H	R/W	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
0082H	DPL	00H	R/W	DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0
0083H	DPH	00H	R/W	DPH7	DPH6	DPH5	DPH4	DPH3	DPH2	DPH1	DPH0
00F7h	XPAGE	00H	R/W	XPAGE7	XPAGE6	XPAGE5	XPAGE4	XPAGE3	XPAGE2	XPAGE1	XPAGE0
Idle and Power-down Control Registers											
Address	Name	Initial Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0087H	PCON	00000000B	R/W	0	0	0	0	0	0	PD	IDL
008EH	SUSLO	00H	R/W	SUSL7	SUSL6	SUSL5	SUSL4	SUSL3	SUSL2	SUSL1	SUSL0
00AFH	PRCON	00000001B	R/W	0	0	0	0	0	ENWDI	0	ENLVR
General I/O Ports Registers											
Address	Name	Initial Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0080H	P0	11111111B	R/W	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
0090H	P1	11111111B	R/W	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
00A0H	P2	00011111B	R/W	0	0	0	0	P2.3	P2.2	P2.1	P2.0
00A2H	P0WK	00000000B	R/W	P0WK7	P0WK6	P0WK5	P0WK4	P0WK3	P0WK2	P0WK1	P0WK0
00A3H	P1WK	00000000B	R/W	P1WK7	P1WK6	P1WK5	P1WK4	P1WK3	P1WK2	P1WK1	P1WK0
00A4H	P2WK	00000000B	R/W	0	0	0	0	P2WK3	P2WK2	P2WK1	P2WK0
009AH	P0CON	00000000B	R/W	P0CON7	P0CON6	P0CON5	P0CON4	P0CON3	P0CON2	P0CON1	P0CON0
009BH	P1CON	00000000B	R/W	P1CON7	P1CON6	P1CON5	P1CON4	P1CON3	P1CON2	P1CON1	P1CON0
009CH	P2CON	00000000B	R/W	0	0	0	0	P2CON3	P2CON2	P2CON1	P2CON0
SPI Registers											
Address	Name	Initial Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
009DH	SPCON	00H	R/W	DIR	MSTR	CPHA	CPOL	SSDIS	SPR2	SPR1	SPR0
00F8H	SPSTA	00H	R/W	SPEN	SPIF	MODF	WCOL	RXOV	0	0	0
009EH	SPDAT	00H	R/W	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
Base Timers											
Address	Name	Initial Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00D2H	BT0	00H	R/W	BT07	BT06	BT05	BT04	BT03	BT02	BT01	BT00
00D3H	BT1	00H	R/W	BT17	BT16	BT15	BT14	BT13	BT12	BT11	BT10
00D4H	BTCON	00H	R/W	ENBT1	BT1M2	BT1M1	BT1M0	ENBT0	BT0M2	BT0M1	BT0M0
00D5H	BTCLK	00H	R/W	0	0	BT1CLK1	BT1CLK0	0	0	BT0CLK1	BT0CLK0
Wake-up Timer & Resume Control Registers											
Address	Name	Initial Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0095H	WKT_CON	F0H	R/W	WKT_SEL1	WKT_SEL0	CLK_SEL1	CLK_SEL0	WKT3	WKT2	WKT1	WKT0
0097H	RES_CON	00H	R/W	0	0	0	0	0	0	RES1	RES0
Reset & Resume Flag											
Address	Name	Initial Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0096H	MODE_FG	00000001B	R/W	0	0	FLVR1	FLVR2	PORT_RSU	WKUP_RS	WDT	POF
Watch-Dog Timer Control Registers											
Address	Name	Initial Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0093H	CLRWD	55H	W	CLRWD7	CLRWD6	CLRWD5	CLRWD4	CLRWD3	CLRWD2	CLRWD1	CLRWD0
0094H	PREWD	05H	R/W	0	0	0	0	0	WDT2	WDT1	WDT0
Interrupt Control Registers											
Address	Name	Initial Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00A8H	IE	00000000B	R/W	EA	EOVL	EWKT	ERSU	EBT1	EBT0	EX1	EX0
00A9	IE2	00000000B	R/W	0	0	0	0	0	0	0	ESPI
00B8H	IP	00000000B	R/W	0	POVL	PWKT	PRSU	PBT1	PBT0	PX1	PX0
00B9H	IP2	00000000B	R/W	0	0	0	0	0	0	0	PSPI
00D9H	EIT	00000000B	R/W	0	0	0	0	0	0	EIT1	EIT0
00DAH	IF	00000000B	R/W	0	IRQ_OVL	IRQ_WKT	IRQ_RSU	IRQ_BT1	IRQ_BT0	IRQ_X1	IRQ_X0



Register (continued)

Information Block Control Registers											
Address	Name	Initial Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00F2H	IBCON1	00H	R/W	IBCON17	IBCON16	IBCON15	IBCON14	IBCON13	IBCON12	IBCON11	IBCON10
00F3H	IBCON2	00H	R/W	0	0	0	0	IBCON23	IBCON22	IBCON21	IBCON20
00F4H	IBCON3	00H	R/W	0	0	0	0	IBCON33	IBCON32	IBCON31	IBCON30
00F5H	IBCON4	00H	R/W	0	0	0	0	IBCON43	IBCON42	IBCON41	IBCON40
00F6H	IBCON5	00H	R/W	0	0	0	0	IBCON53	IBCON52	IBCON51	IBCON50
00FBH	IBOFFSET	00H	R/W	IBOFFSET7	IBOFFSET6	IBOFFSET5	IBOFFSET4	IBOFFSET3	IBOFFSET2	IBOFFSET1	IBOFFSET0
00FCH	IB_DAT	00H	R/W	IBDATA7	IBDATA6	IBDATA5	IBDATA4	IBDATA3	IBDATA2	IBDATA1	IBDATA0
DC-DC & Voltage Comparator Control Registers											
Address	Name	Initial Value	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00C7H	VSEL_D	00100000B	R/W	0	0	DC_EN	DCPD2	DCPD1	0	VSEL_DC	VSEL_DC0
00C4H	VC_CO	01000000B	R/W	ENVC	VREF2	VREF1	VREF0	0	0	0	VCMP

**Note1:** Unlike the other resets (POR, LVR, and RST), resume reset will not reset some registers values

**Note2:** The read/write operations of reserved control registers may cause some unexpected events. Please remain aware during system software development.

**2. Interrupt and Reset Vectors**

■ External Interrupt 0

■ External Interrupt 1

■ Base Timer 0

■ Base Timer 1

■ Resume Interrupt

■ Wake-up Interrupt

■ OVL Interrupt

■ SPI Interrupt

Address	Interrupt Source	Enable	IRQ Flag	Description
0000H	Reset	-	-	System Reset
0003H	External Interrupt 0	IE.0	EX0	P16 (falling edge/low level voltage)
000BH	External Interrupt 1	IE.1	EX1	P17 (falling edge/low level voltage)
0013H	Base Timer 0	IE.2	EBT0	Base Timer 0 Interrupt
001BH	Base Timer 1	IE.3	EBT1	Base Timer 1 Interrupt
0023H	Resume Interrupt	IE.4	ERSU	Port Resume Interrupt (IDLE mode)
002BH	Wake-up Interrupt	IE.5	EWKT	Wake up timer Interrupt (IDLE mode)
0033H	OVL Interrupt	IE.6	EOVL	Invalid ROM address detector Interrupt
003BH	SPI Interrupt	IE2.0	ESPI	Serial peripheral interface Interrupt





## 3. Micro-Processor

### 3.1. General Description

SH68F093 is an 8-bit microprocessor optimized for control applications. Byte-processing and numerical operations on small data structures are facilitated by a variety of fast addressing modes for Internal RAM. The instruction set provides several byte instructions including multiply and divide instructions. In addition, several bit oriented instructions are also provided. This allows direct bit manipulation and testing in control and logic systems that require Boolean processing.

### 3.2. Special Function Registers (SFRs)

SH68F093 has a total of 46 SFR's, as shown in the figure below --- SFR Map for SH68F093. Note that not all the addresses are occupied by SFR's. The unoccupied addresses are not implemented and should not be used by the customer. Read access from these unoccupied locations will return unpredictable data, while write accesses will have no effect on the chip.

SFR Map for SH68F093									
F8H	SPSTA	-	-	IB_OFFSET	IB_DATA	-	-	-	FFH
F0H	B	-	IBCON1	IBCON2	IBCON3	IBCON4	IBCON5	XPAGE	F7H
E8H	-	-	-	-	-	-	-	-	EFH
E0H	ACC	-	-	-	-	-	-	-	E7H
D8H	-	EIT	IF	-	-	-	-	-	DFH
D0H	PSW	-	BT0	BT1	BTCON	BTCLK	-	-	D7H
C8H	-	-	-	-	-	-	-	-	CFH
C0H	-	-	-	-	VC_CON	-	-	VSEL_DC	C7H
B8H	IP	IP2	-	-	-	-	-	-	BFH
B0H	-	-	-	-	-	-	-	-	B7H
A8H	IE	IE2	-	-	-	-	-	PRCON	AFH
A0H	P2	-	P0WK	P1WK	P2WK	-	-	-	A7H
98H	-	-	P0CON	P1CON	P2CON	SPCON	SPDAT	-	9FH
90H	P1	-	-	CLRWDT	PREWDT	WKT_CON	MODE_FG	RES_CON	97H
88H	-	-	-	-	-	-	SUSLO	-	8FH
80H	P0	SP	DPL	DPH	-	-	-	PCON	87H

**Note1:** SFR's in marked column are bit addressable.  
**Note2:** SFR's in gray color are standard 8051 SFR's, and others are SFR's for SH68F093.

**3.2.1. Accumulator (ACC)**

ACC is the accumulator register used for most of the arithmetic and logical instructions. Its initial value is 00h.

**3.2.2. B Register (B)**

The B register is an SFR which is used primarily in the multiply and divide instructions. It can also be used as a temporary scratch pad register for the other instructions and its initial value is 00h.

**3.2.3. Program Status Word (PSW)**

The PSW is the register that holds information about the status of the Accumulator, the selected register banks and other information. Its initial value is 00h. This register is described in details in the following figure:

PSW - Program Status Word Register		
B7	CY	Carry flag
B6	AC	Auxiliary Carry flag (for BCD operations)
B5	F0	Flag 0 (Available to the user for general purposes)
B4	RS1	Register Bank select control bit 1 & 0 Set/cleared by software to determine working bank. (RS1, RS0): (00) - Bank 0 ⇔ Address → (00H - 07H) (01) - Bank 1 ⇔ Address → (08H - 0FH) (10) - Bank 2 ⇔ Address → (10H - 17H) (11) - Bank 3 ⇔ Address → (18H - 1FH)
B3	RS0	
B2	OV	Overflow Flag
B1	X	User definable flag
B0	P	Parity Flag Set/Cleared by hardware each instruction cycle to indicate an odd/even number of "one" bit in the Accumulator, i.e., even parity.

**3.2.4. Stack Pointer (SP)**

The Stack Pointer is an 8-bit wide register that is used to point to the top of the stack where addresses are stored. After a reset, the stack pointer is initialized to 07H, and so the stack begins at 08H. However the stack can reside at any location in the Internal RAM and stack pointer can be programmed to suit the user's needs.

**3.2.5. Data Pointers (DPH, DPL)**

One Data Pointers (DPTR) consist of DPH, DPL. Its intended function is to hold a 16-bit address. It may be manipulated as a 16-bit register or as two independent 8-bit registers.

**3.2.6. Port 0, Port1, and Port2 (P0, P1, and P2)**

The three ports have three SFR's associated with them. Data to be brought out onto the port pins is written to the latches.

**3.2.7. External Memory page register (XPAGE)**

The XPAGE is high byte of address bus when use MOVX with @ Ri.



SH68F093 has a built-in 8M RC resonator for system clock. The system clock can be selected as 8MHz, 4MHz, 2MHz, or 1MHz by Firmware. SH68F093 also has a built-in 32 KHz RC resonator to generate the wake-up timer clock source. The wake up timer clock can be selected as 16 KHz, 32 KHz, or system clock (8MHz/4MHz/2MHz/1MHz) with disabled 32 KHz RC resonator by firmware See the following table.

11



## 5. Reset

There are totally 3 MCU reset sources in the SH68F093 application.

- Hardware reset: Low-Voltage Reset, Power-On Reset or External Reset
- WDT (Watch-Dog Timer) Reset
- Resume Reset (power down mode)

### 5.1. Hardware Reset

#### Idle and Power-down Control Registers

00AFH	PRCON	Initial Value	Low-Voltage Control Register	
Bit[7:3]	-	00000b	-	Reserved
Bit 2	ENWDT	0b	R/W	1: Enable Watch-Dog timer under idle mode 0: Disable Watch-Dog timer under idle mode <b>Reset source: Hardware reset or Resume Reset</b>
Bit 1	-	0b	-	Reserved
Bit 0	ENLVR1	1b	R/W	1: Enable LVR1 under power-down mode 0: Disable LVR1 under power-down mode <b>Reset source: Hardware reset or Resume Reset</b>

#### 5.1.1. Low-Voltage Reset (LVR)

SH68F093 has two LVR circuits named LVR1 and LVR2. The LVR1 circuit will monitor the regulator output voltage to the MCU core. The LVR2 will monitor the output voltage of DC-DC.

##### 1. Low-Voltage Reset (LVR1)

###### (1) Normal Mode and IDLE:

When the regulator output power to the MCU core is less than  $V_{LVR1}$  (**Note**), the LVR1 reset will happen. During the reset, the MCU is going to reset and set the FLVR1 flag (MODE\_FG bit 5). If the regulator output power was larger than  $V_{LVR1}$ , the MCU will go back to be normal and regulator power on. MCU\_Reset signal is asserted until the end of  $T_{RST(POR)}$ . See Fig. 5-1 for the LVR1 behavior.

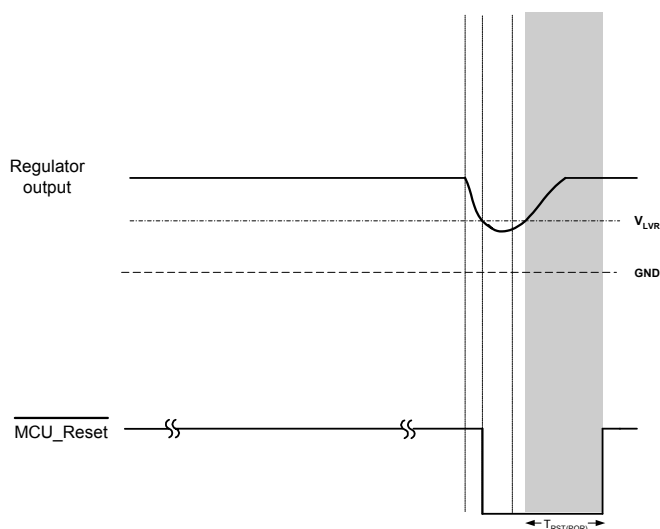


FIG.5-1 Programmable System Clock Diagram

**Note:**  $V_{LVR1}$  ( $V_{LVR1(min)} = 1.4V$ ,  $V_{LVR1(typ)} = 1.5V$ , and  $V_{LVR1(max)} = 1.6V$ )



## (2) Power-down Mode:

- ENLVR1 = 0: Disable LVR1 under power-down mode
- ENLVR1 = 1: Enable LVR1 under power-down mode

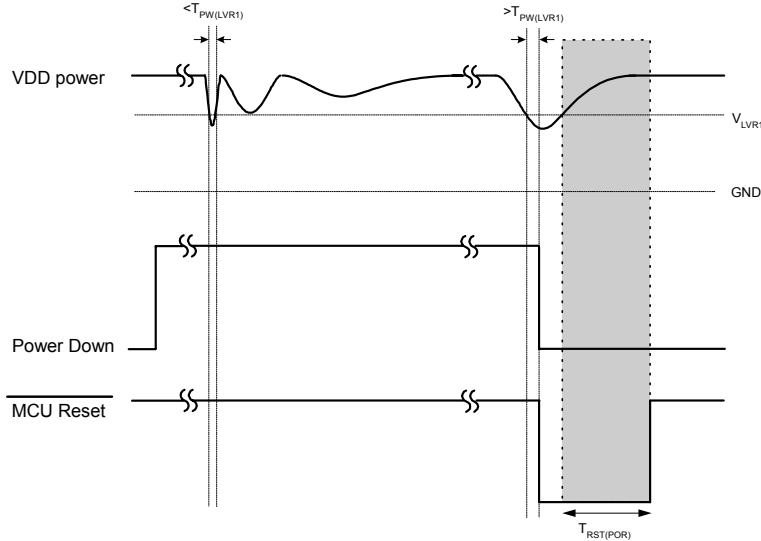


FIG.5-2 Low-Voltage Reset under Power-Down Mode (Note)

**Note1:**  $T_{PW(LVR1)}$  (Drop-Down Pulse Width for LVR1) =  $2^9 \times T_{SYS}$ .

**Note2:**  $V_{LVR1}$  ( $V_{LVR1(min)} = 1.4V$ ,  $V_{LVR1(typ)} = 1.5V$ , and  $V_{LVR1(max)} = 1.6V$ )

## 2. Low-Voltage Reset (LVR2)

The embedded Low-Voltage Reset (LVR2) circuit monitors the output voltage of DC-DC. When the output voltage was less than  $V_{LVR2}$  (**Note**), LVR2 signal will active. Then the MCU will be going to reset and set the FLVR2 flag (MODE\_FG bit4). Then the power of MCU will drop down slowly (because of the capacitor between  $V_{OUT}$  and GND). When the output voltage of DC-DC was larger than  $V_{LVR2}$ , then the MCU will go back to be normal and regulator power on. MCU\_Reset signal is asserted until the end of  $T_{RST(POR)}$ . See Fig. 5-3 for the LVR2 behavior.

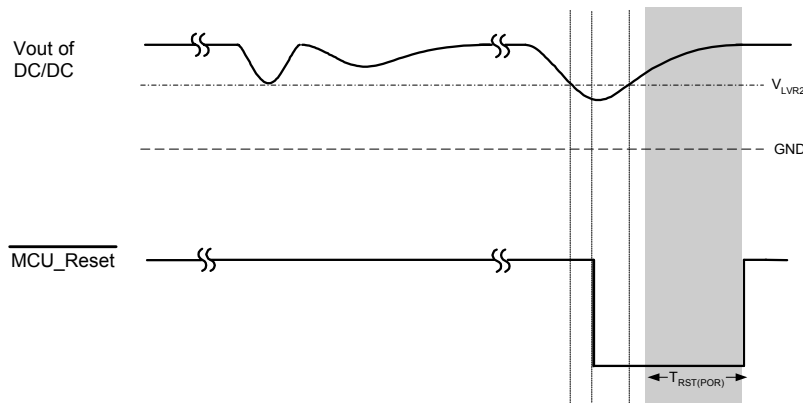


FIG.5-3 Low-Voltage Reset (LVR2)

**Note:**  $V_{LVR2}$  ( $V_{LVR2(min)} = 1.8V$ ,  $V_{LVR2(typ)} = 1.9V$ , and  $V_{LVR2(max)} = 2.0V$ )



### 5.1.2. Power-On Reset (POR)

The embedded Power-On Reset (POR) will monitor the output power of DC-DC to the MCU core. When the supply voltage rises above  $V_{UT(POR)}$  during power on. POR signal is asserted until the end of  $T_{RST(POR)}$  (**Note**). See FIG.5-4 for the POR behavior.

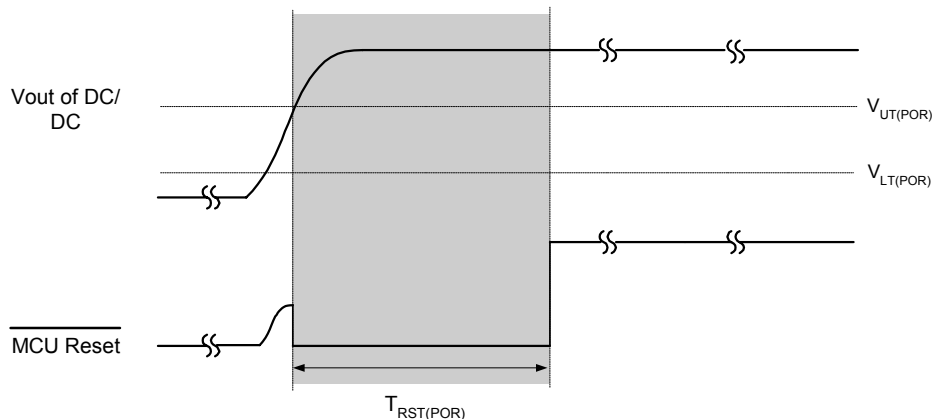


FIG.5-4 Power-On Reset (Note)

**Note1:**  $T_{RST(POR)}$  (Internal Power-on Reset Hold Time) = 11ms

**Note2:**  $V_{UT(POR)}$  (Upper-Threshold Voltage) =  $0.8 \pm 0.1V$

**Note3:**  $V_{LT(POR)}$  (Lower-Threshold Voltage) =  $0.6 \pm 0.1V$

### 5.1.3. External Reset

#### 1. Normal Mode and IDLE

The MCU will generate internal system reset when the pin level of the External Reset is less than the low-threshold voltage ( $V_{LT(RSTB)} = 0.3V_{OUT}$ ) and its pulse width larger than  $T_{PW(RSTB)}$  (**Note**). The reset cycle will end after  $T_{RST(RSTB)}$  (**Note**) when the RSTB pin level is Large than the upper threshold voltage ( $V_{UT(RSTB)} = 0.7V_{OUT}$ ).

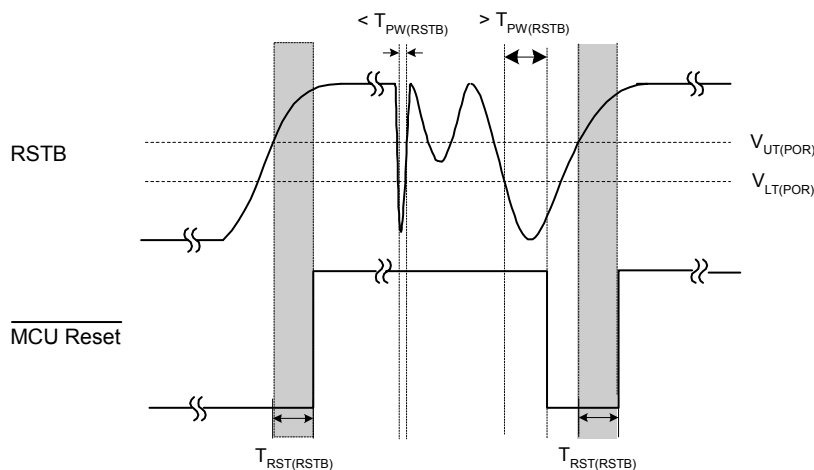


FIG.5-5 External Reset

**Note1:**  $T_{PW(RSTB)}$  (RESETB Input Low Pulse Width) =  $2^{13} \times T_{SYS}$ .

**Note2:**  $T_{RST(RSTB)}$  (Internal External Reset Hold Time) = 11ms.

#### 2. Power-down Mode

When the device was in Power-down mode, an External Reset can't force the device to exit its Power-down mode. SH68F093 has a built-in warm-up counter. Due to to eliminate unstable state, the counter provides a short time to make the MCU stable.



## 5.2. WDT (Watch-Dog Timer) Reset

This MCU implements a watchdog timer reset to avoid system stop or malfunction. The clock source of the WDT is from  $F_{SYS}$ . The Watchdog timer's time-out interval is selected by WDT[2:0]. If the software is in normal operation sequence, then the Watchdog timer must be cleared within time-out period; otherwise the Watchdog timer will overflow and cause a system reset. The Watchdog reset cycle will end after  $T_{RST(WDT)}$  (**Note**). The Watchdog timer is cleared and enabled after the system is reset, and can be disabled by the software under IDLE mode. Users can clear the Watchdog timer by writing a 55H to the CLRWDWT (0093H) register.

**Note:**  $T_{RST(WDT)}$  (internal Watch-Dog reset hold time) = 0.5ms.

0093H	CLRWDWT	Initial Value	Clear Watch-dog Timer Control Register	
Bit[7:0]	CLRWDWT [7:0]	55H	W	Write "55H" to clear watch-dog timer <b>Reset source: Hardware reset, WDT reset, or Resume reset</b>

00AFH	PRCON	Initial Value	Low-Voltage Control Register	
Bit[7:3]	-	00000b	-	<b>Reserved</b>
Bit2	ENWDT	0b	R/W	1: Enable Watch-Dog timer under idle mode 0: Disable Watch-Dog timer under idle mode <b>Reset source: Hardware reset or Resume Reset</b>
Bit1	-	0b	-	<b>Reserved</b>
Bit0	ENLVR1	1b	R/W	1: Enable LVR1 under power-down mode 0: Disable LVR1 under power-down mode <b>Reset source: Hardware reset or Resume Reset</b>

0094H	PREWDT	Initial Value	Watch-dog Timer Control Register	
Bit[7:3]	-	00000b	-	<b>Reserved</b>
Bit[2:0]	WDT[2:0]	101b	R/W	<b>Watch-dog Timer time-out selection</b> 000: $2^{15} \times T_{SYS}$ 001: $2^{16} \times T_{SYS}$ 010: $2^{17} \times T_{SYS}$ 011: $2^{18} \times T_{SYS}$ 100: $2^{19} \times T_{SYS}$ 101: $2^{20} \times T_{SYS}$ 110: $2^{21} \times T_{SYS}$ 111: $2^{22} \times T_{SYS}$ <b>Reset source: Hardware reset or WDT reset</b>



### 5.3. Resume Reset (Power Down Mode)

A resume reset holds SFR values, CPU status and Pin state, but program is re-run at 0000h. **Port 0, Port 1, and Port 2** can be set as a resume port by setting P0WK, P1WK, and P2WK. **The low level, raising edge, falling edge, or double edge** of enabled resume source is triggered in power-down mode will causes a resume reset.

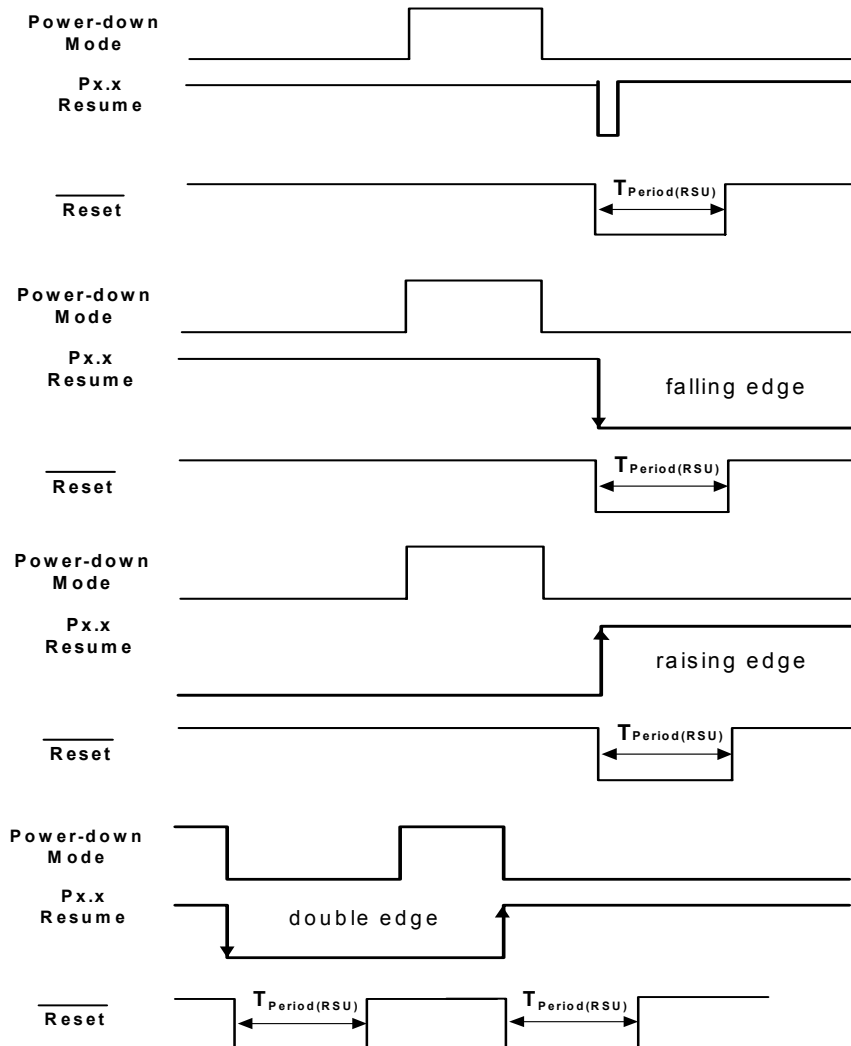


FIG.5-6 In power-down enabled resume ports are triggered at the low level, raising edge, falling edge, and double edge

**Note:**  $T_{Period(RSU)}$  (resume reset period) = 24 $\mu$ s.

0097H	RES_CON	Initial Value	Resume Control Register	
Bit[7:2]	-	000000b	-	Reserved
Bit[1:0]	RES[1:0]	00b	R/W	00: low level triggered resume port hold reset/interrupt 01: raising edge triggered resume port hold reset/interrupt 10: falling edge triggered resume port hold reset/interrupt 11: double edge triggered resume port hold reset/interrupt <b>Reset Source: Hardware reset or WDT Reset</b>





## 6. Power-reducing Mode

SH68F093 has two power-reducing modes: IDLE and Power-down. A sketch map below shows the internal circuitry, which implement these features. In IDLE mode (IDLE = 1 & SUSLO = 55H), the 8MHz RC resonator continues to run and the Interrupt and Timer blocks continue to be clocked. In Power-down (PD = 1 & SUSLO = 55H), the 8MHz RC resonator is frozen. Setting bits in SFR (Special Function Register) PCON activates IDLE and Power-down modes.

008EH	SUSLO	Initial Value	Power saving Control Register 1	
Bit[7:0]	SUSLO[7:0]	00H	R/W	IDL = 1 & SUSLO = 55H: Enter idle mode PD = 1 & SUSLO = 55H: Enter Power-down mode <b>Reset source: Hardware reset, WDT reset, or Resume reset</b>

0087H	PCON	Initial Value	Power saving Control Register 2	
Bit[7:2]	-	000000b	-	<b>Reserved</b>
Bit1	PD	0b	R/W	PD = 1 & SUSLO = 55H: Enter Power-down mode <b>Reset source: Hardware reset, WDT reset, or Resume reset</b>
Bit0	IDL	0b	R/W	IDL = 1 & SUSLO = 55H: Enter idle mode <b>Reset source: Hardware reset, WDT reset, or Resume reset</b>

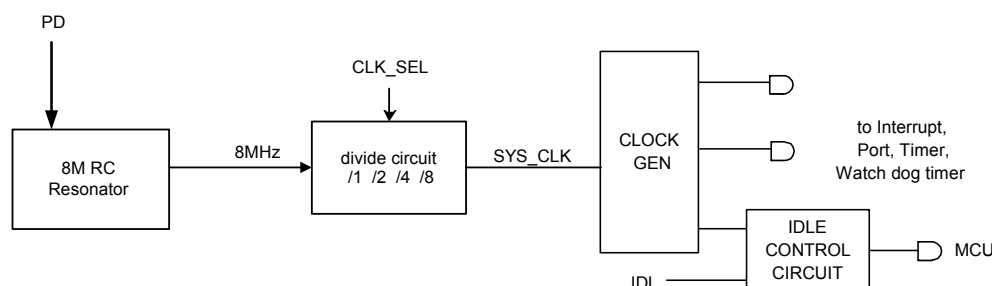


FIG.6-1: Sketch map for IDLE and Power-Down Mode implement

### 6.1. IDLE Mode

Two continuous instructions that set PCON.0 to 1 and set SUSLO to 55H are executed before IDLE mode begins. In IDLE mode, the internal clock signal is gated off to the CPU, but not to the Interrupt, Timer, and Serial Port functions. The CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, and all other registers maintain their data during IDLE mode. The port pins hold the logical states they had at the time IDLE mode was activated. Under IDLE Mode, the clock source of wake-up timer is from system clock.

There are four ways to terminate IDLE mode. (If Watchdog timer was disabled at IDLE mode, then it will restart to count from the value where it was stopped when entering IDLE Mode. When leave IDLE Mode, PCON.0 and SUSLO will be cleared by hardware)

- (1) Activation of any enabled interrupt will cause SH68F093 exit form IDLE Mode. The interrupt will be serviced, and following RETI the next instruction to be executed will be the one following the instruction that put the device into IDLE mode. In order to make sure program executed properly, user should add two NOPs after the instruction that put the device into IDLE mode. (Watchdog timer begins to count from the value where it was stopped.)
- (2) When the wake-up timer is time-out in IDLE mode, it will set the IRQ\_WKT (IF.5) flag. If the interrupt is enabled by writing "1" to the IE.5 before entering IDLE mode, then it will interrupt 8051 CPU for terminating IDLE mode. Wake-up timer can be disabled by set WKT\_SEL = 00b.
- (3) Port0, Port1, and Port2 can be set as a resume port by setting POWK, P1WK, and P2WK. When a low level of enabled resume source is triggered in IDLE mode, it will set the IRQ\_RSU (IF.4) flag if the interrupt is enabled by writing "1" to the IE.4 before entering IDLE mode, then it will interrupt 8051 CPU for terminating IDLE mode. In order to make sure program, it will executes properly, user should add two NOPs after the instruction that put the device into IDLE mode.
- (4) The other way of terminating IDLE mode is with a LVR, POR, External reset or Watchdog reset. At this time, the CPU resumes program execution from the beginning of the whole program, which is 0000H.



```

graph LR
    subgraph " "
        IDLE:
        MOV CLRWDT, #55H
        MOV SUSLO, #55h
        ORL PCON, #01H
        NOP
        NOP
        MOV CLRWDT, #55H
        :
        :
        :
    end
    subgraph " "
        INTSUB:
        :
        MOV A, MD_FG
        ANL A, #08H
        JZ NORMAL
        :
        RETI
        NORMAL:
        :
        :
        :
        RETI
    end
    IDLE --> INTSUB
    INTSUB --> NORMAL

```

00A2H	P0WK	Initial Value	Port0 Resume Enable Register	
Bit[7:0]	P0WK[7:0]	00h	R/W	1: Enable wake-up function of PORT0's pins 0: Disable wake-up function of PORT0's pins <b>Reset source: Hardware reset or WDT reset</b>

00A3H	P1WK	Initial Value	Port1 Resume Enable Register	
Bit [7:0]	P1WK[7:0]	00h	R/W	1: Enable wake-up function of PORT1's pins 0: Disable wake-up function of PORT1's pins <b>Reset source: Hardware reset or WDT reset</b>

00A4H	P2WK	Initial Value	Port2 Resume Enable Register	
Bit [3:0]	P2WK[3:0]	00h	R/W	1: Enable wake-up function of PORT2's pins 0: Disable wake-up function of PORT2's pins <b>Reset source: Hardware reset or WDT reset</b>

## 6.2. Power-Down Mode

In the Power-down mode, if the WKT\_SEL = 00b, the on-chip programmable 32KHz RC resonator clock will be frozen, all functions are stopped, but the on-chip RAM and Special function Registers are held. The port pins output the values held by their respective SFR. With the wake-up timer enable (WKT\_SEL ≠ 00b), the wake-up timer will be enabled, and wake-up system when wake-up timer overflow. In order to make sure program will resume properly, user should add two NOPs after setting PCON.1 to “1” and SUSLO = 55h. Under Power-Down Mode, the clock source of wake-up timer is set by WKT\_SEL. When the device was in Power-down mode, an External Reset can't force the device to exit its Power-down mode.

### 1. Low Voltage Reset

## 2. Resume Reset

A resume reset holds SFR values, CPU status and Pin state, but program is re-run at 0000h. There are two ways to generate resume reset.



(1) Port 0, Port 1, and Port 2 can be set as a resume port by setting P0WK, P1WK, and P2WK. The low level, raising edge, falling edge, or double edge of enabled resume source is triggered in power-down mode will causes a resume reset.

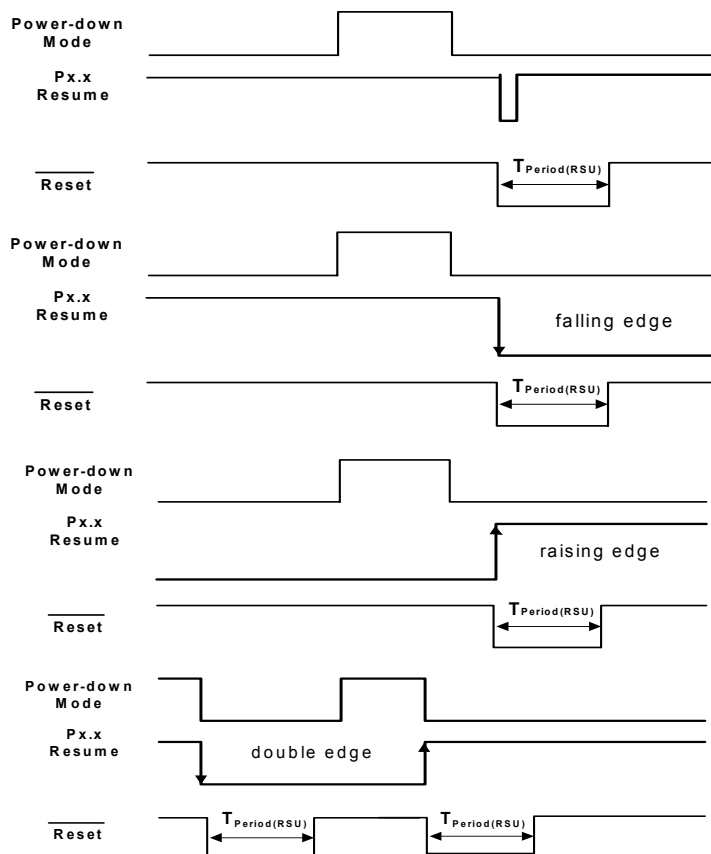


FIG.6-2 In power-down enabled resume ports are triggered at the low level, raising edge, falling edge, and double edge

**Note:**  $T_{Period(RSU)}$  (resume reset period) = 24 $\mu$ s.

0097H	RES_CON	Initial Value	Resume Control Register	
Bit[7:2]	-	000000b	-	Reserved
Bit[1:0]	RES[1:0]	00b	R/W	00: low level triggered resume port hold reset/interrupt 01: raising edge triggered resume port hold reset/interrupt 10: falling edge triggered resume port hold reset/interrupt 11: double edge triggered resume port hold reset/interrupt <b>Reset Source: Hardware reset or WDT Reset</b>

(2) Wake-up Timer Time-out

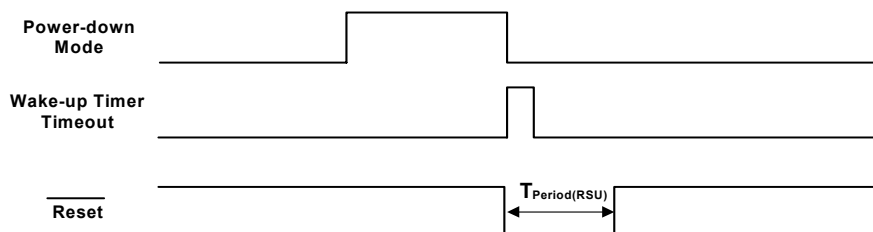


FIG.6-3 Wake-up timer time out waveform

**Note:**  $T_{Period(RSU)}$  (resume reset period) = 24 $\mu$ s.



### 6.3. Wake-up Timer

SH68F093 has a Built-in programmable RC resonator providing of wake-up timer. The wake-up timer can be enabled/disable by WKT\_SEL[1:0]. If the Wake-up timer is enabled, the wake-up timer will free run, and the time-out period can be adjusted by WKT[3:0]. See the following block diagram:

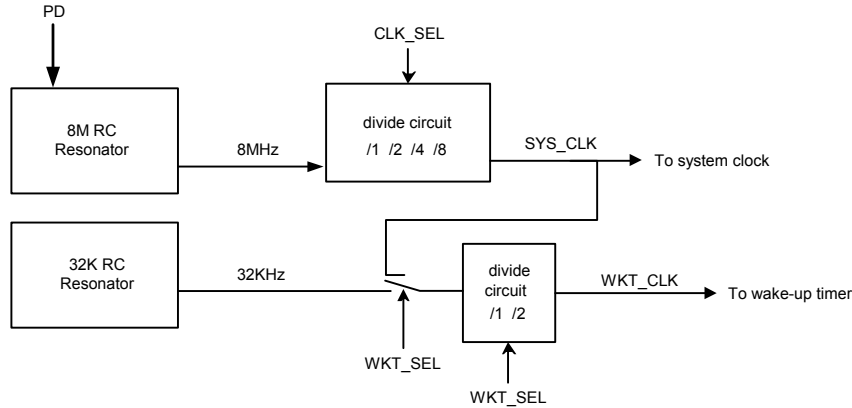


FIG.6-4 The Clock Block Diagram

0095H	WKT_CON	Initial Value	Wake-up Timer & Resume Reset Control Register	
Bit[7:6]	WKT_SEL[1:0]	11b	R/W	<b>Wake-up timer clock source(TWKT_CLK) selection</b> 00: Disable Wake-up Timer 01: 16KHz (In Power-down mode) 01: Disable Wake-up Timer (In IDLE & normal operation mode) 10: 32KHz (In Power-down mode) 10: Disable Wake-up Timer (In IDLE & normal operation mode) 11: System clock (In IDLE mode) 11: Disable Wake-up Timer (In Power-down & normal operation mode) <b>Reset source: Hardware reset or WDT reset</b>
Bit[5:4]	CLK_SEL[1:0]	11b	R/W	<b>System clock Frequency selection</b> 00: 1MHz 01: 2MHz 10: 4MHz 11: 8MHz <b>Reset source: Hardware reset or WDT reset</b>
Bit[3:0]	WKT[3:0]	0000b	R/W	<b>Wake up Time</b> 0000: Reserved 0001: $2^7 \times \text{TWKT\_CLK}$ 0010: $2^8 \times \text{TWKT\_CLK}$ 0011: $2^9 \times \text{TWKT\_CLK}$ 0100: $2^{10} \times \text{TWKT\_CLK}$ 0101: $2^{11} \times \text{TWKT\_CLK}$ 0110: $2^{12} \times \text{TWKT\_CLK}$ 0111: $2^{13} \times \text{TWKT\_CLK}$ 1000: $2^{14} \times \text{TWKT\_CLK}$ 1001: $2^{15} \times \text{TWKT\_CLK}$ 1010: $2^{16} \times \text{TWKT\_CLK}$ 1011: $2^{17} \times \text{TWKT\_CLK}$ 1100: $2^{18} \times \text{TWKT\_CLK}$ 1101: $2^{19} \times \text{TWKT\_CLK}$ 1110: $2^{20} \times \text{TWKT\_CLK}$ 1111: $2^{21} \times \text{TWKT\_CLK}$ <b>Reset source: Hardware reset or WDT reset</b>



## 6.4. Change System Clock Source

SH68F093 can select several system clock frequency modes that are 1MHz, 2MHz, 4MHz, or 8MHz. To change the system clock, user just needs to write CLK\_SEL[1:0] and the clock control will change the system clock when all clock source at rising edge.

## 6.5. MODE\_FG Flag

0096H	MODE_FG	Initial Value	Mode Register	
Bit[7:6]	-	00b	-	<b>Reserved</b>
Bit5	FLVR1	0b	R/W	Set "1": after LVR1 reset. Write "0" to clear, write "1" no effect. 1: set by LVR1 <b>Reset source: Power-on reset or External Reset</b>
Bit4	FLVR2	0b	R/W	Set "1": after LVR2 reset. Write "0" to clear, write "1" no effect. 1: set by LVR2 <b>Reset source: Power-on reset or External Reset</b>
Bit3	PORT_RSU	0b	R/W	Set "1" after Port Resume reset in Power down mode. Write "0" to clear, write "1" no effect. <b>Reset source: Hardware reset or WDT Reset</b>
Bit2	WKUP_RSU	0b	R/W	Set "1" after Wake-up timer time-out Resume reset in Power down mode, Write "0" to clear, write "1" no effect. <b>Reset source: Hardware reset or WDT Reset</b>
Bit1	WDT	0b	R/W	Set "1" after Watchdog reset. Write "0" to clear, write "1" no effect <b>Reset source: Hardware reset</b>
Bit0	POF	1b	R/W	Set "1" after power-on reset or External reset. Write "0" to clear, write "1" no effect. <b>Reset source: Power-on reset or External Reset</b>

**Note:** In the beginning SH68F093 applied the power until stable situation MODE\_FG (11H) is set both FLVR2 and POF flag to ONE. In order to avoid confusing, we suggest that the user be following examples.

**Example 1: POF, LVR1, and LVR2 are the same priority level.**

```
void Main(void)
{
    //POF //LVR1 // LVR2
    if((MODE_FG&0x01)||((MODE_FG&0x20)||((MODE_FG&0x10)
    {
        :
    }
    // other flag
    else if (MODE_FG..)
    {
        :
    }
}
```

**Example 2: POF is higher priority level than LVR1 and LVR2.**

```
void Main(void)
{
    if(MODE_FG&0x01) //POF
    {
        :
    }
    else if (MODE_FG&0x20) //LVR1
    {
        :
    }
    else if (MODE_FG&0x20) //LVR2
    {
        :
    }
    else if (MODE_FG..) // other flag
    {
        :
    }
}
```

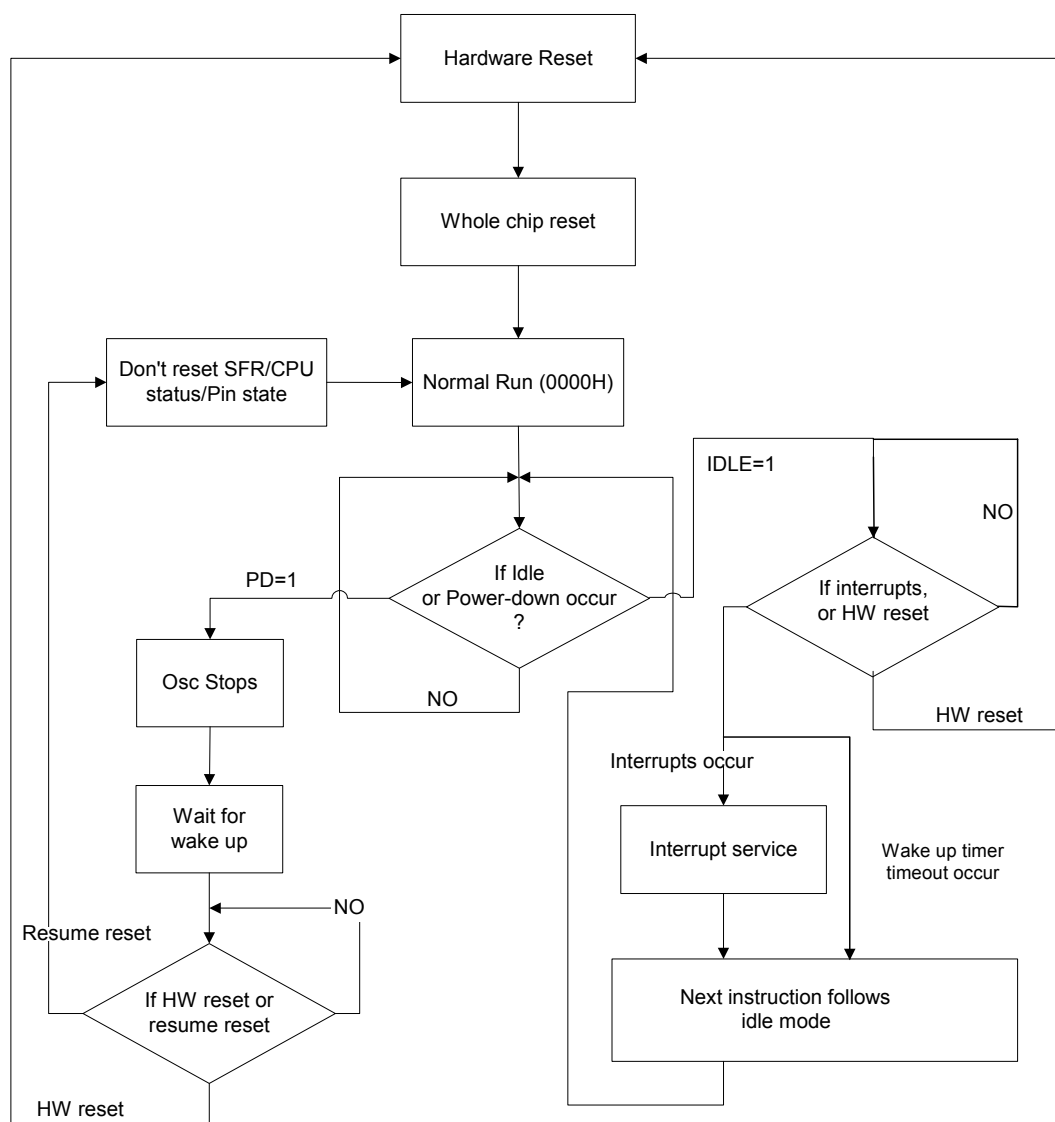


FIG.6-5 Flow chart of IDLE Mode and Power-down Mode



## Summary

### IDLE Mode

To enter this mode	Set PCON.0 = 1, SUSLO = 55h
Watch dog timer	ENWDT = 1, enable watch-dog timer ENWDT = 0, disable watch-dog timer
Function	Interrupt, Base Timer, and I/O port are available
Register	Internal RAM, external RAM, and SFR were held
To terminate IDLE mode	<p><b>(1) Interrupt:</b> Any activate interrupt will terminate IDLE Mode. If EA = 0, the interrupt will not be serviced, if EA = 1, the interrupt will be serviced.</p> <p><b>(2) Wake-up Timer:</b> Wake-up timer time-out (enable by setting WKT_SEL[1:0] = 11b) and <b>EWKT = 1</b></p> <p><b>(3) Port Wake-up:</b> <b>(4) Port0-2</b> are triggered by low level, raising edge, falling edge, or double edge when P0WK-P2WK are enabled and <b>ERSU = 1</b></p> <p><b>(5) Reset:</b> LVR1, LVR2, POR, external reset, and watch-dog reset</p>
Resume	Interrupt and go to interrupt vector (0023H)
Wake-up	Interrupt and go to interrupt vector (002BH)

### Power-down Mode

To enter this mode	Set PCON.1 = 1, SUSLO = 55h
Watch dog timer	Disable
Function	All function (Interrupt, Base Timer, and I/O port are available) are stopped
Register	Internal RAM, external RAM, and SFR were held
To terminate Power-down mode	<p><b>(1) LVR1 Reset</b> (Disabled if ENLVR1 = 0)</p> <p><b>(2) LVR2 Reset</b></p> <p><b>(3) Resume Reset:</b> Port0-2 are triggered by low level voltage, raising edge, falling edge, or double edge when P0WK-P2WK are enabled</p> <p><b>(4) Wake-up timer:</b> Wake-up timer time out (enabled by setting WKT_SEL[1:0] = 01b or WKT_SEL[1:0] = 10b)</p>
16/32KHz RC oscillator	<p><b>WKT_SEL[1:0]</b></p> <p>00b: disable programmable ring oscillator</p> <p>01b: Wake-up timer use 16KHz Ring Oscillator</p> <p>10b: Wake-up timer use 32KHz Ring Oscillator</p> <p>11b: Wake-up timer use system clock, define in CLK_SEL[1:0]</p>
Resume	Reset and re-run at 0000H
Wake-up	Reset and re-run at 0000H



## 7. Input/Output Ports

### 7.1. Port-0 Configuration: (Reset source: Hardware reset)

I/O Port	Function	I/O	Circuit Structure	Control Bits		Description
				P0.x	P0CON.x	
Port0 [7:0]	Port0	O	Shown in FIG.7-1	0	0	Output Low (MAX: 0.3Vout) @ Vout = 2.7V Output Low (MAX: 0.3Vout) @ Vout = 3.3V
				1	0	Output High (MIN: 0.7Vout) @ Vout = 2.7V Output High (MIN: 0.7Vout) @ Vout = 3.3V
		I		X	1	HI-Z ( <b>Note 1</b> )

### 7.2. Port-1 Configuration: (Reset source: Hardware reset)

I/O Port	Function	I/O	Circuit Structure	Control Bits		Description
				P1.x	P1CON.x	
Port1 [7:0]	Port1	O	Shown in FIG.7-1	0	0	Output Low (MAX: 0.3Vout) @ Vout = 2.7V Output Low (MAX: 0.3Vout) @ Vout = 3.3V
				1	0	Output High (MIN: 0.7Vout) @ Vout = 2.7V Output High (MIN: 0.7Vout) @ Vout = 3.3V
		I		X	1	HI-Z ( <b>Note 1</b> )

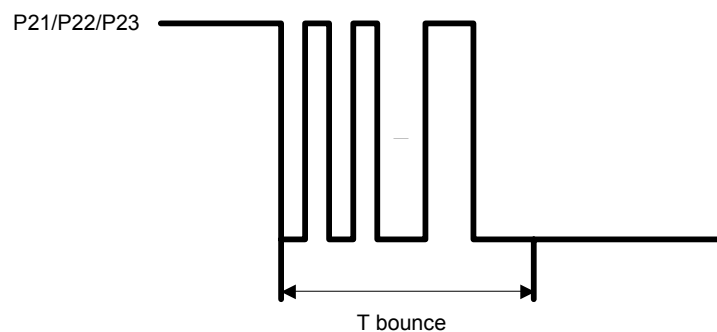
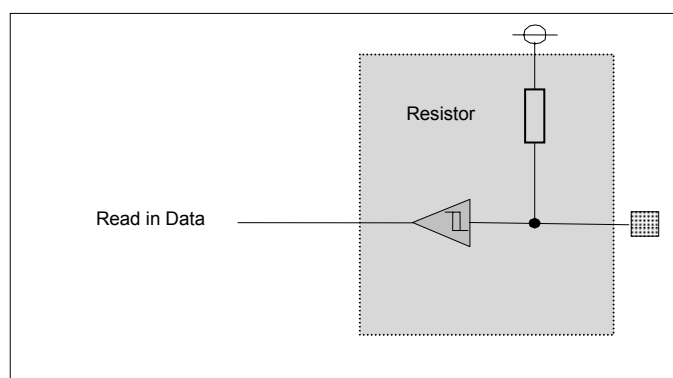
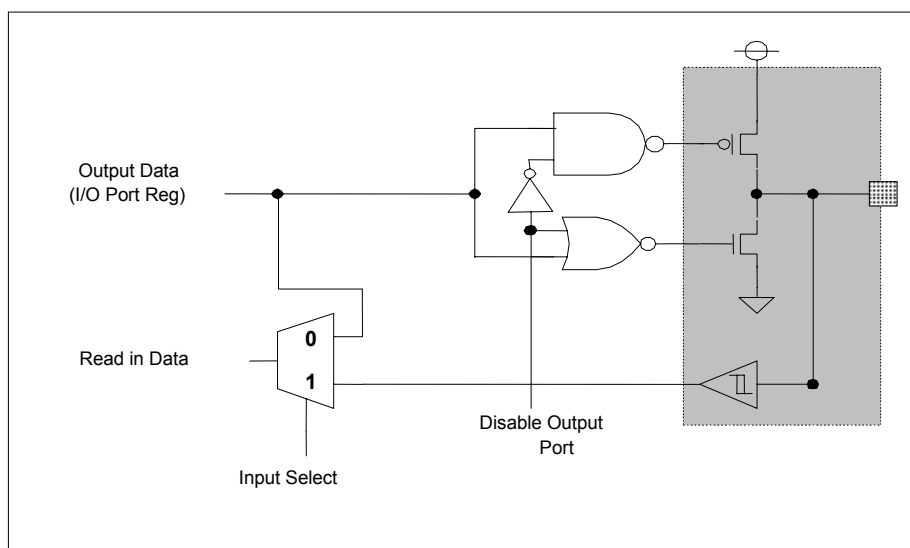
### 7.3. Port-2 Configuration: (Reset source: Hardware reset)

I/O Port	Function	I/O	Circuit Structure	Control Bits		Description
				P2.x	P2CON.x	
Port2 [0]	Port2	O	Shown in FIG.7-1	0	0	Output Low (MAX: 0.3Vout) @ Vout = 2.7V Output Low (MAX: 0.3Vout) @ Vout = 3.3V
				1	0	Output High (MIN: 0.7Vout) @ Vout = 2.7V Output High (MIN: 0.7Vout) @ Vout = 3.3V
				X	1	HI-Z ( <b>Note 1</b> )
Port2 [1:3]		I	Shown in FIG.7-2	X	X	Input High (0.7Vout) Input Low (0.3Vout) ( <b>Note 2</b> )

**Note1:** P00-P07, P10-P17, and P20 are set input pins to become those ports without pull resistors; these ports have better to be connected a pull resistor (about 10k ohm) to V<sub>DD</sub>.

**Note2:** Used to only unidirectional input pins from Port21 to Port23.





**Note1:** Assume P21, P22, and P23 connected to switches. The unidirectional input pins have  $T_{BOUNC}$  delay for switch bounce.

**Note2:**  $T_{BOUCNE} \doteq 15ms$ .



## 8. Interrupts

### 8.1. Interrupt Enables

Each of the interrupt sources can be individually enabled or disabled by setting or clearing a bit in the SFR named IE. The register IE also contains a global disable bit, which can be cleared to disable all interrupts at once. The following figure shows the interrupt register for SH68F093.

#### Interrupt Enable Register

00A8h	IE	Initial Value	Interrupt Enable Register in the SH68F093	
Bit7	EA	0b	R/W	Disable all interrupts. If EA = 0, no any interrupts will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
Bit6	EOVL	0b	R/W	Invalid ROM address detector Interrupt
Bit5	EWKT	0b	R/W	Wake up time interrupt (IDLE mode)
Bit4	ERSU	0b	R/W	Port resume interrupt (IDLE mode)
Bit3	EBT1	0b	R/W	Base Timer 1 interrupt
Bit2	EBT0	0b	R/W	Base Timer 0 interrupt
Bit1	EX1	0b	R/W	External interrupt 1
Bit0	EX0	0b	R/W	External interrupt 0
<b>Enable bit = 1, enables the interrupt</b> <b>Enable bit = 0, disables the interrupt</b> <b>Reset source: Hardware reset or WDT reset</b>				

00A9h	IE2	Initial Value	Interrupt Enable Register in the SH68F093	
Bit[7:1]	-	0000000b	-	Reserved
Bit0	ESPI	0b	R/W	SPI interrupt
<b>Enable bit = 1, enables the interrupt</b> <b>Enable bit = 0, disables the interrupt</b> <b>Reset source: Hardware reset or WDT reset</b>				



## 8.2. Interrupt Priorities

### Interrupt Enable Register

00B8H	IP	Initial Value	IP (Interrupt Priority) Register in the SH68F093	
Bit7	-	0b	-	Reserved
Bit6	POVL	0b	R/W	Invalid ROM address detector Interrupt priority bit
Bit5	PWKT	0b	R/W	Wake up time interrupt priority bit
Bit4	PRSU	0b	R/W	Port resume interrupt priority bit
Bit3	PBT1	0b	R/W	Base Timer 1 interrupt priority bit
Bit2	PBT0	0b	R/W	Base Timer 0 interrupt priority bit
Bit1	PX1	0b	R/W	External interrupt 1 priority bit
Bit0	PX0	0b	R/W	External interrupt 0 priority bit
1 = high priority; 0 = low priority Reset Source: Hardware reset, or WDT Reset				

00B9H	IP2	Initial Value	Interrupt Priority Register in the SH68F093	
Bit[7:1]	-	0b	-	Reserved
Bit0	PSPI	0b	R/W	SPI interrupt priority bit
1 = high priority; 0 = low priority Reset Source: Hardware reset or WDT Reset				

### 2 Priority Levels

IP0	Level
0	0
1	1

A low-priority interrupt can be interrupted by a high-priority interrupt, but cannot be interrupted by another low-priority interrupt. A high-priority interrupt can't be interrupted by any other interrupt source. If two interrupt requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If interrupt requests the same priority levels are received simultaneously, **an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by software polling sequence.**

In operation, all the interrupt flags are latched into the interrupt control system every machine cycle. The samples are polled during the following machine cycle. If the flag for an enabled interrupt is set to 1, the interrupt system generates an LCALL to the appropriate location in Program Memory, unless some other condition blocks an interrupt, such as an interrupt of equal or higher priority level already in progress.

The hardware-generated LCALL accesses the contents of the Program Counter pushed onto the stack, and reloads the PC with the beginning address of the service routine. As previously noted, the service routine for each interrupt begins at a fixed location.

**Only the Program Counter is automatically pushed onto the stack, not the PSW or any other register.** Having only the PC automatically saved allows the programmer to decide how much time to spend saving other registers. This enhances the interrupt response time, albeit at expense of increasing the programmer's burden of responsibility. As a result, many interrupt functions that are typical in control applications—toggling a port pin, for example, or reloading a timer, or unloading a serial buffer—can often be completed in less time than it takes other architectures to commence them.


**8.3. Interrupt Flag**

00DAH	IF	Initial Value	Interrupt Control Flag	
Bit7	-	0b	-	Reserved
Bit6	IRQ_OVL	0b	R/W	OVL Interrupt flag. Set by hardware when an invalid program ROM address is detected. Write "0" to clear, write "1" no effect. <b>Reset source: Hardware reset or WDT reset</b>
Bit5	IRQ_WKT	0b	R/W	Wake-up Timer Interrupt request flag. Set by hardware when the Wake-up timer overflow is detected in IDLE mode. Cleared by hardware when interrupt is processed. Write "0" to clear, write "1" no effect. <b>Reset source: Hardware reset or WDT reset</b>
Bit4	IRQ_RSU	0b	R/W	Port Resume Interrupt request flag. Set by hardware when the Port Resume signal is detected in IDLE mode. Cleared by hardware when interrupt is processed. Write "0" to clear, write "1" no effect. <b>Reset source: Hardware reset or WDT reset</b>
Bit3	IRQ_BT1	0b	R/W	Base Timer 1 Interrupt request flag. Set by hardware when the Base timer 1 overflow is detected. Cleared by hardware when interrupt is processed. Write "0" to clear, write "1" no effect. <b>Reset source: Hardware reset or WDT reset</b>
Bit2	IRQ_BT0	0b	R/W	Base Timer 0 Interrupt request flag. Set by hardware when the Base Timer 0 overflow is detected. Cleared by hardware when interrupt is processed. Write "0" to clear, write "1" no effect. <b>Reset source: Hardware reset or WDT reset</b>
Bit1	EXT1	0b	R/W	External Interrupt 0 flag. Set by hardware when the P17 Interrupt edge signal is detected. Cleared by hardware when interrupt is processed. Write "0" to clear, write "1" no effect. <b>Reset source: Hardware reset or WDT reset</b>
Bit0	EXT0	0b	R/W	External Interrupt 0 flag. Set by hardware when the P16 Interrupt edge signal is detected. Cleared by hardware when interrupt is processed. Write "0" to clear, write "1" no effect. <b>Reset source: Hardware reset or WDT reset</b>

00D9H	EIT	Initial Value	Interrupt Control Flag	
Bit[7:2]	-	000000b	-	Reserved
Bit1	EIT1	0b	R/W	External interrupt 1 type control bit: 0: falling edge triggered External Interrupt 1: low level triggered External Interrupt <b>Reset source: Hardware reset or WDT reset</b>
Bit0	EIT0	0b	R/W	External interrupt 0 type control bit: 0: falling edge triggered External Interrupt 1: low level triggered External Interrupt <b>Reset source: Hardware reset or WDT reset</b>

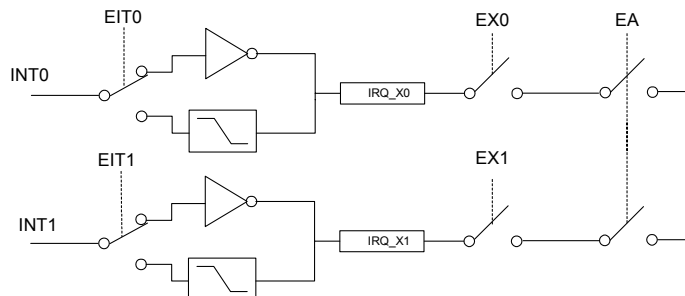
0097H	RES_CON	Initial Value	Resume Control Register	
Bit[7:2]	-	000000b	-	Reserved
Bit[1:0]	RES	00b	R/W	00: low level triggered resume port hold reset/interrupt 01: raising edge triggered resume port hold reset/interrupt 10: falling edge triggered resume port hold reset/interrupt 11: double edge triggered resume port hold reset/interrupt <b>Reset Source: Hardware reset or WDT Reset</b>



## 8.4. Interrupt Sources

### 8.4.1. External Interrupt Sources

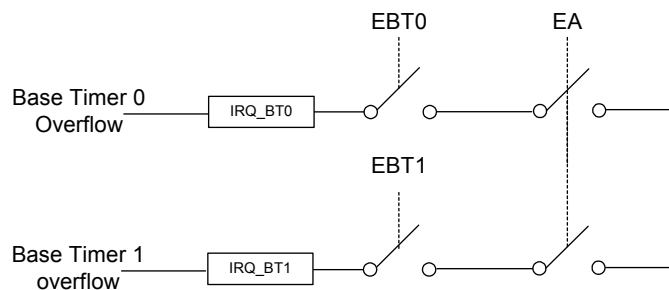
The External interrupt source is activated when the External Interrupt edge is detected. The following figure shows the detailed structure of the INT0 interrupt sources.



**FIG.8-1 External Interrupt Source**

### 8.4.2 Base Timer 0/Base Timer 1 Interrupt Sources

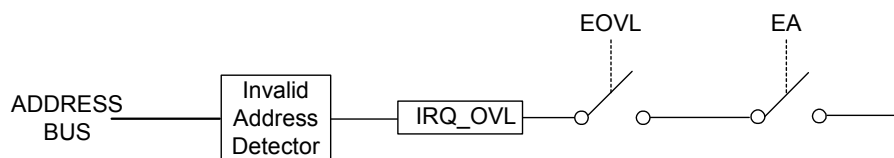
The Base Timer 0/Base Timer 1 interrupt source is activated when the Base Timer0/Base Timer 1 overflows. The following figure shows the detailed structure of the BT0/BT1 interrupt sources.



**FIG.8-2 Base Timer 0/Base Timer 1 Interrupt Source**

### 8.4.3 OVL Interrupt Sources

The OVL interrupt source is activated when the CPU detects an invalid program ROM address. The following figure shows the detailed structure of the OVL sources.

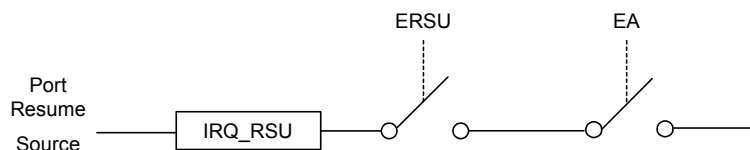


**FIG.8-3 OVL Interrupt Source**



#### 8.4.4. Resume Interrupt Sources

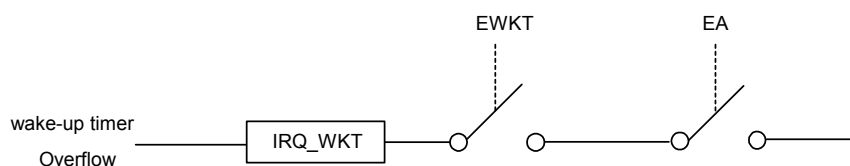
The RSU interrupt source is activated when a port resume source is triggered in IDLE mode. The following figure shows the detailed structure of the RSU sources.



**FIG.8-4 Resume Interrupt Source**

#### 8.4.5. WKT Interrupt Sources

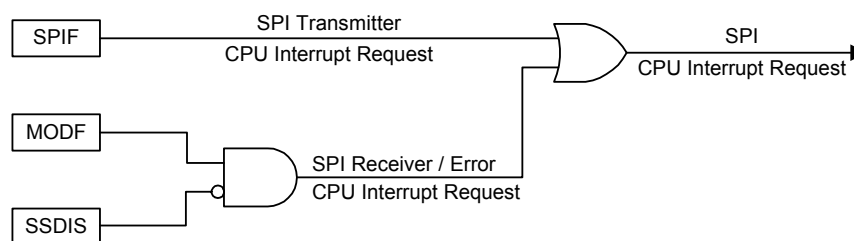
The WKT interrupt source is activated when Wake-up timer overflows in IDLE mode. The following figure shows the detailed structure of the WKT sources.



**FIG.8-5 Wake-up timer Interrupt Source**

#### 8.4.6. SPI Interrupt Sources

The SPI interrupt source is activated when SPIF or MODF is triggered. The following figure shows the detailed structure of the SPI sources.



**FIG.8-6 SPI Interrupt Source**



## 9. Base Timer

The Timer-x is an 8-bit counter with a programmable clock source selection and the value of Base Timer-x counter can be read out any time. The Base Timer-x can be enabled/disabled by the CPU. After reset, the Base Timer-x is disabled and cleared. The Base Timer-x can be preset by writing a preset value to BTx register at any time. When the Base Timer-x is enabled, the Base Timer-x starts counting from the preset value to FFH and when the values reaches 00H, it generates a Base Timer-x interrupt if the Base Timer-x interrupt is enabled. When it reaches 00H, the Base Timer-x will auto-load the value in BTx register and begins counting. The Base Timer-x can be enabled by writing a "1" to "ENBTx" in the BTCN (Base Timer Control) register. The ENBTx is level trigger. If any value is written to BTx register when it is counting, Base Timer-x will reload that value immediately and continue counting from that written value. Every time ENBTx goes rising, the counter begins to count from the preset value in BTx register.

The input clock of Base Timer-x is controlled by the BTxCLK[1:0] register. The following table shows 4 types: 8MHz, 4MHz, 2MHz, or 1MHz. The clock source of Base Timer-x is controlled by the BTxM[2:0] register. The following table shows 8 ranges of the Base Timer-x.

00D2H	BT0	Initial Value	Base Timer-0 Control Register	
Bit[7:0]	BT0[7:0]	00h	R/W	Base Timer 0 register <b>Reset Source: Hardware reset or USB reset</b>

00D3H	BT1	Initial Value	Base Timer-1 Control Register	
Bit[7:0]	BT1[7:0]	00h	R/W	Base Timer 1 register <b>Reset Source: Hardware reset or USB reset</b>

00D5H	BTCLK	Initial Value	Base Timer Clock Register	
Bit[7:6]	-	00b	R/W	Reserved
Bit[5:4]	BT1CLK [1:0]	00b	R/W	Frequency Base Timer-1 clock register (FBT1 <sub>CLK</sub> ) 11: 8MHz 10: 4MHz 01: 2MHz 00: 1MHz <b>Reset Source: Hardware reset or WDT reset</b>
Bit[3:2]	-	00b	R/W	Reserved
Bit[1:0]	BT0CLK [1:0]	00b	R/W	Frequency Base Timer-0 clock register (FBT0 <sub>CLK</sub> ) 11: 8MHz 10: 4MHz 01: 2MHz 00: 1MHz <b>Reset Source: Hardware reset or WDT reset</b>



00D4H	BTCON	Initial Value	Base Timer Control Register	
Bit7	ENBT1	0b	R/W	0: Disable Base Timer-1 1: Enable Base Timer-1 <b>Reset Source: Hardware reset or WDT reset</b>
Bit[6:4]	BT1M[2:0]	000b	R/W	Base Timer-1 clock source 000: $FBT1_{CLK}/2^0$ 001: $FBT1_{CLK}/2^1$ 010: $FBT1_{CLK}/2^2$ 011: $FBT1_{CLK}/2^3$ 100: $FBT1_{CLK}/2^4$ 101: $FBT1_{CLK}/2^6$ 110: $FBT1_{CLK}/2^8$ 111: $FBT1_{CLK}/2^{10}$ <b>Reset Source: Hardware reset or WDT reset</b>
Bit3	ENBT0	0b	R/W	0: Disable Base Timer-0 1: Enable Base Timer-0 <b>Reset Source: Hardware reset or WDT reset</b>
Bit[2:0]	BT0M[2:0]	000b	R/W	Base Timer-0 clock source 000: $FBT0_{CLK}/2^0$ 001: $FBT0_{CLK}/2^1$ 010: $FBT0_{CLK}/2^2$ 011: $FBT0_{CLK}/2^3$ 100: $FBT0_{CLK}/2^4$ 101: $FBT0_{CLK}/2^6$ 110: $FBT0_{CLK}/2^8$ 111: $FBT0_{CLK}/2^{10}$ <b>Reset Source: Hardware reset or WDT reset</b>

**Note:** For counting accuracy, please set the Base Timer-x register first, then preset the BTxM[2:0] register; last, enable the Base Timer-x.





## 10. DC-DC Converter

### 10.1. Features

- Boost Type DC-DC converter
- Operating voltage range: 1.1V - 1.7V (static voltage without load)
- Output voltage: 2.7V or 3.3V
- Maximum output currents (built-in):
  - 70mA at  $V_{BAT} = 1.5V$ ,  $V_{OUT} = 3.3V$
  - 100mA at  $V_{BAT} = 1.5V$ ,  $V_{OUT} = 2.7V$ .... **(Note)**

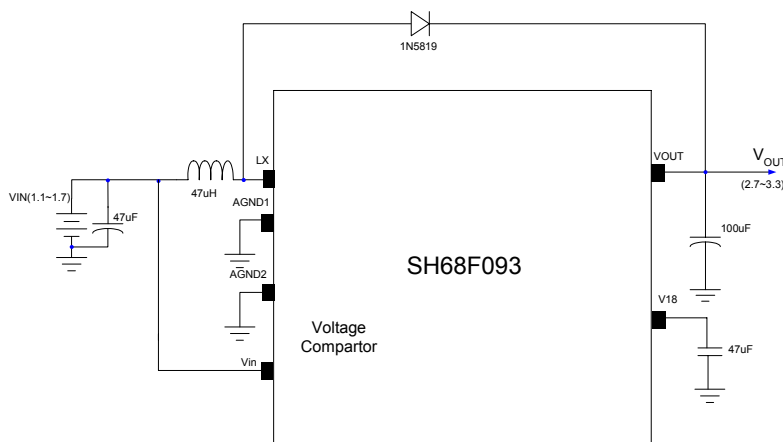
**Note:** DC-DC converter specification refers as page 46 and performance depends on external components and PCB layout.

### 10.2. General Description

The DC-DC converter is a high performance Boost type converter with low noise and low supply current. The DC-DC converter part consists of a switching transistor, oscillation circuit and comparator. As external parts, a coil, a diode, and a capacitor are available for obtaining a constant output voltage,  $V_{OUT}$  (2.7V or 3.3V) higher than the output voltage of the battery for the DC-DC converter part.

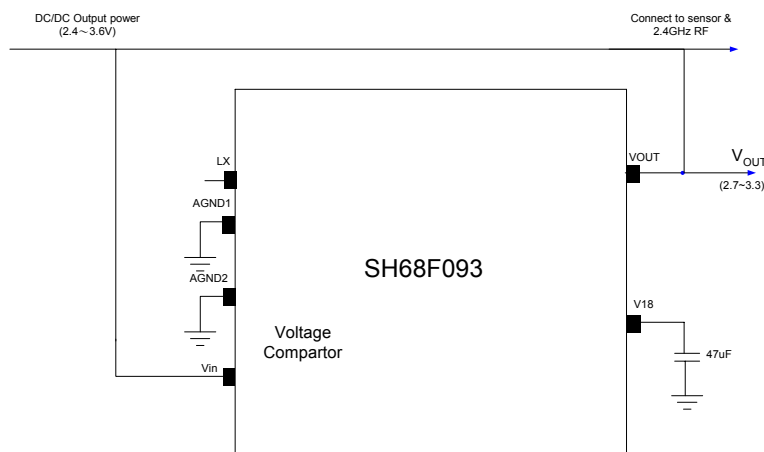
### 10.3. Block Diagram and Application Circuits

(1) Enable built-in DC-DC converter:



**FIG.10-1**

(2) Disable built-in DC-DC converter:



**FIG. 10-2**

**Note:** about voltage comparator usage, see page 36.



#### 10.4. Register

00C7H	VSEL_DC	Initial Value	DC-DC Feedback resistor Control Register	
Bit[7:6]	-	00b	-	Reserved
Bit5	EN_DC	1b	R/W	1: Enable built-in DC-DC 0: Disable built-in DC-DC
Bit[4:3]	DCPD[1:0]	00	R/W	<b>DC-DC converter power down mode select</b> 00: PWM mode (operating current: 600uA) 01: VFM mode (operating current: 255uA) 10: Stop mode (turn DC/DC off) 11: reserved <b>Reset source: Hardware reset or WDT reset</b>
Bit2	-	0b	-	Reserved
Bit[1:0]	VSEL_DC [1:0]	00b	R/W	<b>DC-DC output voltage control register</b> 00: DC-DC output: 2.7V 01: DC-DC output: 2.8V (optional) 10: DC-DC output: 3.3V 11: DC-DC output: 3.4V (optional) <b>Reset source: Hardware reset or WDT reset</b>

Compare enable built-in DC-DC with disable built-in DC-DC

Enable/Disable	Enable Built-in DC-DC Converter	Disable Built-in DC-DC Converter
DCPD	DCPD=00: PWM mode (operating current: 500uA) DCPD=01: VFM mode (operating current: 200uA) DCPD=10: Stop mode (turn DC/DC off)  (1) When waked up or resume, MCU immediately goes back to default (PWM mode) (2) In stop mode (DCPD = 10), Vbat = Vout.	DCPD = XX  Due to be turned built-in DC-DC off, setting DCPD does not work at DC-DC converter power down mode selection.



## 11. Voltage Comparator

The Voltage Comparator compares the voltage on  $V_{REF}$  and  $V_{IN}$ . Use the Voltage Comparator user can easily get the  $V_{IN}$  voltage level.

### 11.1. Control Procedure

- It compares the voltage input voltage ( $V_{IN}$ ) with the reference voltage ( $V_{REF}$ ). And then, it stores the results ( $V_{CMP}$ ) into the VC\_CON.0 register. The typical comparing time for voltage detector is 5us.
- The detection steps are shown below:
  - $V_{REF}[2:0]$
  - Set EN\_VC
  - Wait for 5us
  - Read VC\_CON.0

### 11.2. Register

00C4H	VC_CON	Initial Value	Voltage Comparator Control Register	
Bit7	EN_VC	0b	R/W	Enable or disable the programmable voltage comparator circuit. 1: enable 0: disable <b>Reset source: Hardware reset or WDT reset</b>
Bit[6:4]	VREF[2:0]	100b	R/W	Set reference voltage ( <b>Note</b> ) 000b - 010b: reserved 011b: $V_{REF} = 0.8V$ 100b: $V_{REF} = 0.9V$ 101b: $V_{REF} = 1.0V$ 110b - 111b: reserved <b>Reset source: Hardware reset or WDT reset</b>
Bit[3:1]	-	000b	-	Reserved
Bit0	VCMP	0b	R	Compare result of the voltage detector. 1: $V_{IN} \geq V_{REF}$ 0: $V_{IN} < V_{REF}$ <b>Reset source: Hardware reset or WDT reset</b>

**Note:**  $V_{BAT}$  from 1.1V to 1.7V that means static voltage without load so that at the condition of dynamic voltage with load the one-cell battery provides is lower voltage than that provides at the static voltage without load. Therefore the battery actually provides from 0.8V to 1.5V at the condition of the dynamic voltage with load.

### 11.3. Block Diagram

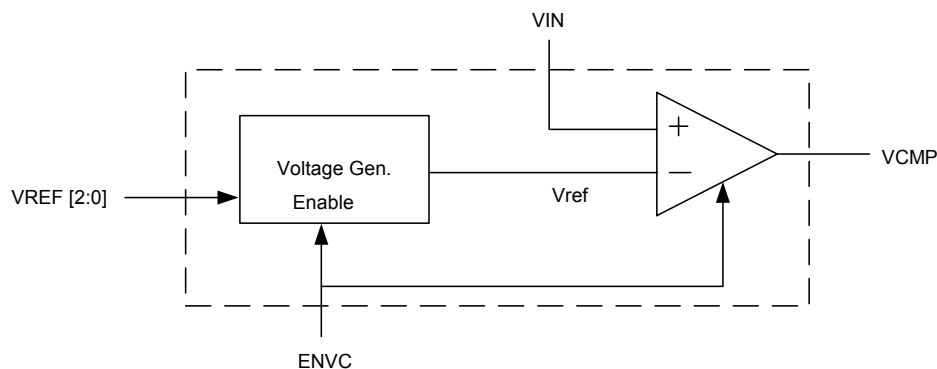
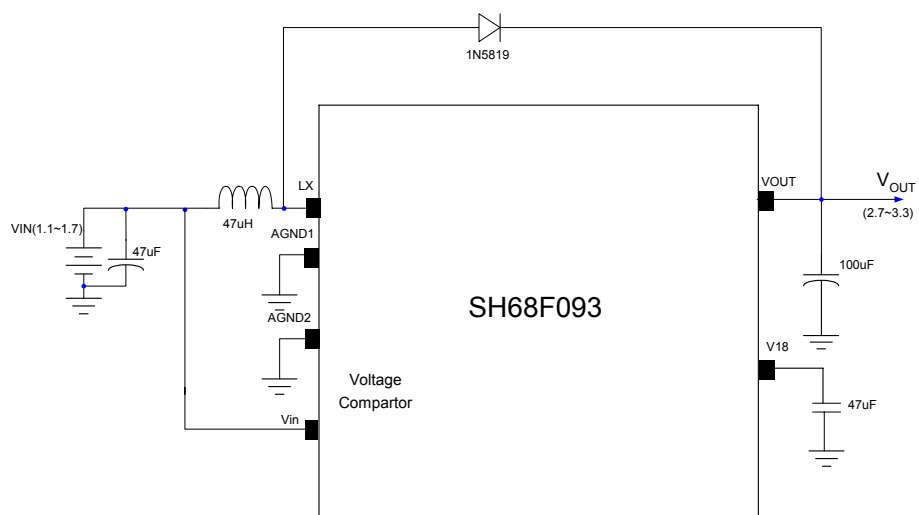


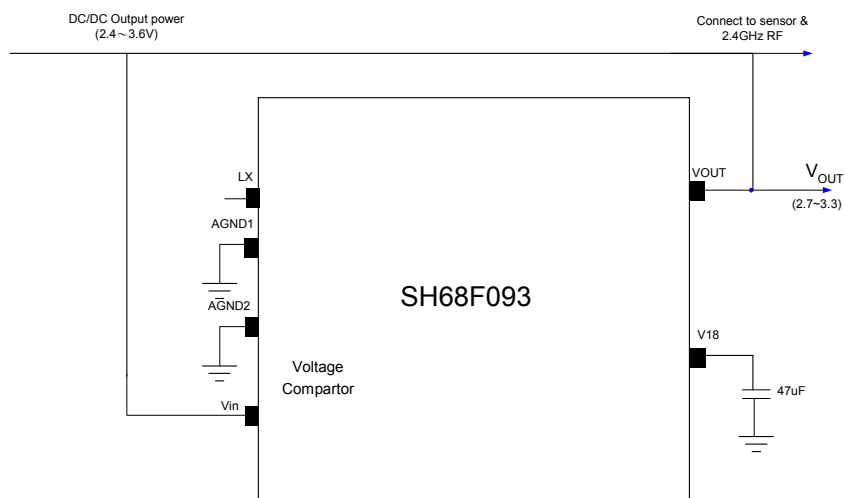
FIG.11-1 Voltage Comparator Block Diagram



#### 11.4. Voltage Comparator Application Circuit



**FIG.11-2 Voltage comparator connected to battery under enable built-in DC-DC converter**



**FIG.11-3 Voltage comparator connected to DC-DC output under disable built-in DC-DC converter**



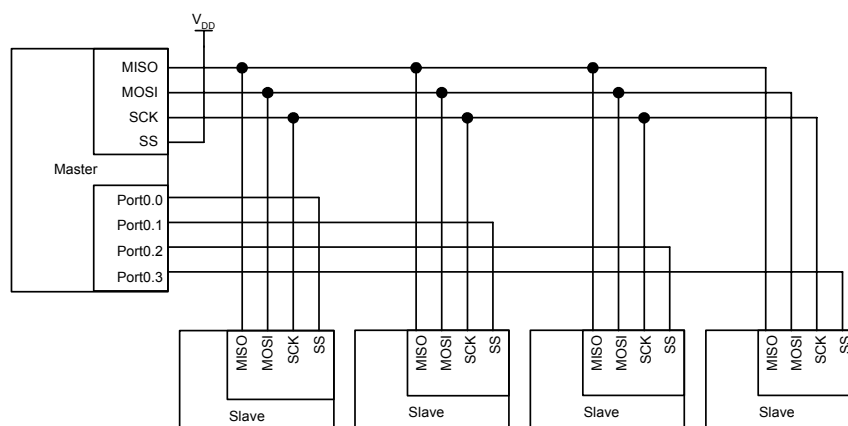
## 12. SPI (Serial Peripheral Interface)

The Serial Peripheral Interface Module (SPI) allows full-duplex, synchronous, serial communication between the MCU and peripheral devices, including other MCUs.

### 12.1. Features

- Full-duplex, three-wire synchronous transfers
- Master or Slave operation
- Six programmable Master clock rates
- Serial clock with programmable polarity and phase
- Master Mode fault error flag with MCU interrupt capability
- Write collision flag protection
- LSB or MSB transfer selectable

The following diagram shows a typical SPI bus configuration using one Master controller and many Slave peripherals. The bus is made of three wires connecting all the devices. The Master device selects the individual Slave devices by using four pins of a parallel port to control the four SS pins of the Slave devices.



**FIG.12-1 SPI bus connection between the master and the slaves**

### 12.2. Signal Description

**Master Output Slave Input (MOSI):** This 1-bit signal is directly connected between the master device and a slave device. The MOSI line is used to transfer data in series from the master to the slave. Therefore, it is an output signal from the master, and an input signal to a slave.

**Master Input Slave Output (MISO):** This 1-bit signal is directly connected between the slave device and a master device. The MISO line is used to transfer data in series from the Slave to the Master. Therefore, it is an output signal from the Slave, and an input signal to the Master. The MISO pin is placed in a high-impedance state when the SPI operates as a slave that is not selected. (/SS = high) A static high level on the /SS pin puts the MISO line of a slave in a high-impedance state.

**SPI Serial Clock (SCK):** This signal is used to synchronize the data movement both in and out of the devices through their MOSI and MISO lines. It is driven by the Master for eight clock cycles which allows to exchange one byte on the serial lines. The SCK signal is ignored by a SPI slave when the slave is not selected (/SS = 1).

**Slave Select (/SS):** Each Slave peripheral is selected by one Slave Select pin (/SS). This signal must stay low for any active Slave. It is obvious that only one Master (/SS high) can drive the network. The Master may select each Slave device by software through port pins. To prevent bus conflicts on the MISO line, only one slave should be selected at a time by the Master for a transmission. In a Master configuration, the /SS line can be used in conjunction with the MODF flag in the SPI Status register to prevent multiple masters from driving MOSI and SCK.

The /SS pin could be used as a general IO if the following conditions are met:

- The device is configured as a Master and the SSDIS control bit in SPCON is set. This kind of configuration can happen when only one Master is driving the network. Therefore, the MODF flag in the SPSTA will never be set.
- The Device is configured as a Slave with CPHA and SSDIS control bits set. This kind of configuration can happen when the network comprises only one Master and one Slave only. Therefore, the device should always be selected and the Master will never use the slave's /SS pin to select the target communication Slave.

**Note:** When CPHA = '0', the /SS pin is used to start the transmission.

**Baud rate:** In Master mode, the baud rate is chosen from one of the six clock rates by the division of the internal clock by 4, 8, 16, 32, 64 or 128 set by the three bits SPR[2:0] in the SPCON register.



The following diagram shows a detailed structure of the SPI module.





## 12.4. Registers

009DH	SPCON	Initial Value	Serial Peripheral Interface Control Register	
Bit7	DIR	0b	R/W	<b>Transfer Direction Selection</b> 0 = MSB first 1 = LSB first <b>Reset source: Hardware reset or WDT reset</b>
Bit6	MSTR	0b	R/W	<b>Serial Peripheral Master</b> 0 = Configure the SPI as a Slave 1 = Configure the SPI as a Master <b>Reset source: Hardware reset or WDT reset</b>
Bit5	CPHA	0b	R/W	<b>Clock Phase</b> 0 = Data sampled on first edge of SCK period 1 = Data sampled on second edge of SCK period <b>Reset source: Hardware reset or WDT reset</b>
Bit4	CPOL	0b	R/W	<b>Clock Polarity</b> 0 = SCK line low in idle state 1 = SCK line high in idle state <b>Reset source: Hardware reset or WDT reset</b>
Bit3	SSDIS	0b	R/W	<b>/ss Disable</b> 0 = Enable /SS pin in both Master and Slave modes 1 = Disable /SS pin in both master and slave modes MODF interrupt request will not generate, if SSDIS is set In Slave mode, this bit has no effect if CPHA = 0 <b>Reset source: Hardware reset or WDT reset</b>
Bit[2:0]	SPR[2:0]	000b	R/W	<b>Serial Peripheral Clock Rate</b> 000 = system clock/4 (default) 001 = system clock/8 010 = system clock/16 011 = system clock/32 100 = system clock/64 others = system clock/128 <b>Reset source: Hardware reset or WDT reset</b>



00F8H	SPSTA	Initial Value	Serial Peripheral Interface Status Register	
Bit7	SPEN	0b	R/W	<b>SPI enable</b> 0 = Disable the SPI interface 1 = Enable the SPI interface <b>Reset source: Hardware reset or WDT reset</b>
Bit6	SPIF	0b	R/W	<b>Serial peripheral data transfer flag</b> 0 = Clear by software 1 = Set by hardware to indicate that the data transfer has been completed <b>Reset source: Hardware reset or WDT reset</b>
Bit5	MODF	0b	R/W	<b>Mode fault</b> 0 = Cleared by software 1 = Set by hardware to indicate that the /SS pin is at inappropriate logic level <b>Reset source: Hardware reset or WDT reset</b>
Bit4	WCOL	0b	R/W	<b>Write collision flag</b> 0 = Cleared by software to indicate write collision has be processed 1 = Set by hardware to indicate that a collision has been detected <b>Reset source: Hardware reset or WDT reset</b>
Bit3	RXOV	0b	R/W	<b>Receive Overrun</b> 0 = Cleared by software to indicate receive overrun has be processed 1 = Set by hardware to indicate that a receive overrun has been detected <b>Reset source: Hardware reset or WDT reset</b>
Bit[2:0]	-	000b	-	Reserved

009EH	SPDAT	Initial Value	Serial Peripheral Interface Data Register	
Bit[7:0]	SPDAT[7:0]	00H	R/W	A write to SPDAT places data directly into the transfer shift register. A read of the SPDAT returns the value located in the receive shift register. <b>Reset source: Hardware reset or WDT reset</b>

## 12.5. SPI Modes

The Serial Peripheral Interface can be configured as one of the two modes: Master mode or Slave mode. Once the SPI is configured, the data exchange is made using SPCON (The Serial Peripheral Control register), SPSTA (The Serial Peripheral Status register), and SPDAT (The Serial Peripheral Data register).

During an SPI transmission, data is simultaneously transmitted (shifted out serially) and received (shifted in serially). A serial clock line (SCK) synchronizes shifting and sampling on the two serial data lines (MOSI and MISO). A Slave Select line (/SS) allows individual selection of a SPI Slave; SPI Slave that are not selected do not interfere with SPI bus activities.

When the SPI Master transmits data to the SPI Slave via the MOSI line, the SPI Slave responds by sending data to the SPI Master via the MISO line. This implies full-duplex transmission with both data out and data in synchronized with the same clock. Both transmit shift register and receive shift register uses the same SFR Address = a write operation to SPDATA will write to the transmit shift register, and a read operation from SPDATA will retrieve the data in receive shift register.

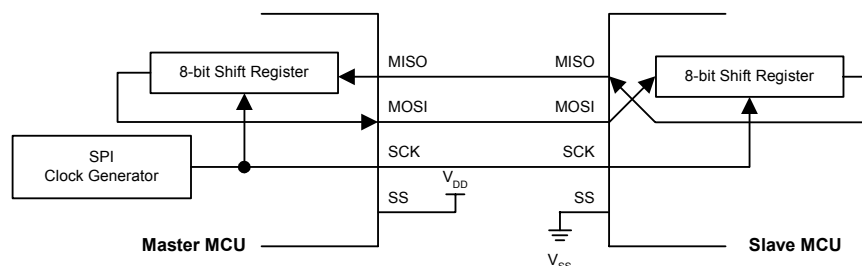


FIG.12-3 Full-Duplex Master-Slave Interconnection Diagram





### 12.5.1. Master Mode

**Enable:** A SPI master device initiates all data transfers on a SPI bus. The SPI operates in master mode when the MSTR is set in SPCON register. Only one master can initiate transmission.

**Transmit:** When in SPI master mode, writing a byte of data to the SPI data register (SPDAT) will write to the transmit shift buffer. If the transmit shift register already contains data, the SPI master will generate a WCOL signal to indicate writing too fast. But the data in transmit shift register will not be affected, and the transmission continues uninterrupted. Else if the transmit shift register is empty, the SPI master will immediately shifts out the data serially on the MOSI line while providing the serial clock on SCK. The SPIF flag in SPSTA register is set to logic '1' at the end of the transfer. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set.

**Receive:** While the master transfers data to a slave on the MOSI line, the addressed slave simultaneously transfers the contents of its transmit shift register to the master's receive shift register on the MISO line in a full-duplex operation. Therefore, the SPIF flag serves as both a transmit-complete and receive-data-ready flag. The data byte received from the slave is transferred MSB-first into the master's shift register. When a byte is fully shifted into the register, it is moved to the receive buffer where it can be read by the processor by reading SPDAT. If an overrun occurs, RXOV signal will be set to indicate data over-run occurs, and the receive shift register keep the byte that SPIF was lastly set, also the SPI master will not receive any further data until SPIF was cleared.

### 12.5.2. Slave Mode

**Enable:** The SPI operates in Slave mode when the MSTR is cleared in the SPCON register. Before a data transmission occurs, the Slave Select (/SS) pin of the Slave device must be set to '0'. The /SS pin must remain low until the 1-byte transmission is complete.

**Transmit & Receive:** When in SPI slave mode, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter counts SCK edges. When 8 bits have been shifted in the receive shift register and another 8 bits have been shifted out the transmit shift register, the SPIF flag is set to logic '1'. Data is read from the receive shift register by reading SPDAT. If interrupts are enabled, an interrupt request is generated when the SPIF flag is set.

To prevent an overflow condition, the SPI slave software must read the SPDAT before another byte enters the shift register. Else a RXOV signal will be set to indicate data over-run occurs, and the receive shift register keep the byte that SPIF was lastly set, also the SPI slave will not receive any further data until SPIF was cleared.

A SPI slave cannot initiate transfers. Data to be transferred to the master device is pre-loaded into the shift register by writing to SPDAT. Writes to SPDAT are placed in the transmit buffer first. So a SPI slave must complete the write to the SPDAT (transmit shift register) in one SPI clock before the master starts a new transmission. If the write to SPDAT is late, the SPI will transmit a '0x00' byte in the following transmission, if the write operation occurs during this time, a WCOL signal will be set. If the transmit shift register already contains data, the SPI slave will generate a WCOL signal to indicate writing too fast. But the data in transmit shift register will not be affected, and the transmission continues uninterrupted.



## 12.6. Transmission Formats

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL.

CPOL is defined the default SCK line level in idle state. It has no significant effect on the transmission format. CPHA is defined the edges on which the input data are sampled and the edges on which the output data are shifted. The clock phase and polarity should be identical for the master and the communicating slave.

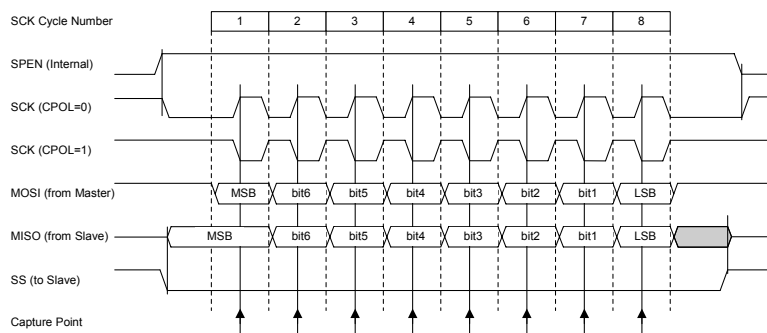


FIG.12-4 Data Transmission Format (CPHA = 0)

If CPHA = 0, the first SCK edge is the capture strobe. Therefore the Slave must begin driving its data before the first SCK edge, and a falling edge on the /SS pin is used to start the transmission. The /SS pin must be toggled high and then low between each byte transmitted. So SSDIS bit is invalid when CPHA = 0.

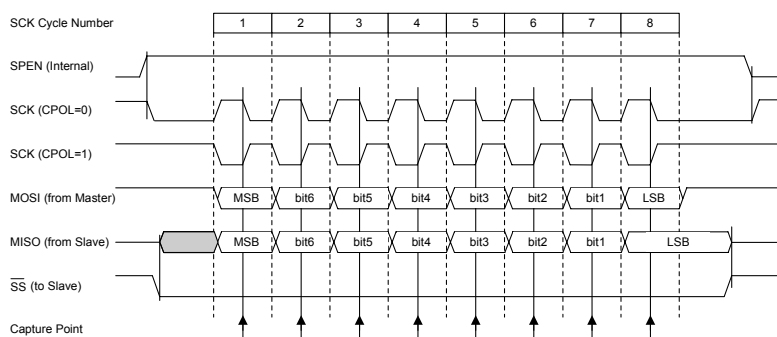


FIG.12-5 Data Transmission Format (CPHA = 1)

If CPHA = 1, the Master begins driving its MOSI pin on the first SCK edge. Therefore the slave uses the first SCK edge as a start transmission signal. So the user must put the SPDAT during the two edge of first SCK. The /SS pin can remain low between transmissions. This format may be preferred in systems with only one master and only one slave.

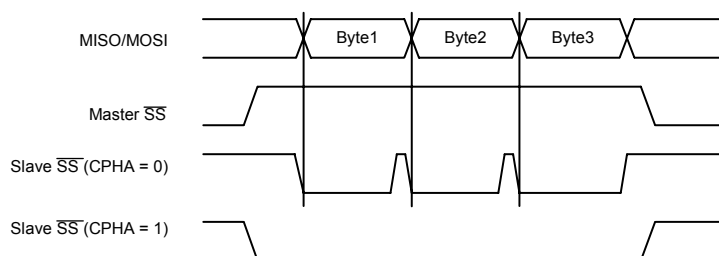


FIG.12-6 CPHA and /SS Timing

**Note:** When SPI is configured as Slave mode and CPOL bit in SPCON is cleared, the SCK pin must be set to input mode and enable pull-high resistor before SPEN bit in SPSTA is set to logic '1'.



## 12.7. Error Conditions

The following flags in the SPSTA signal SPI error conditions:

**Mode Fault (MODF):** Mode Fault error in master mode SPI indicates that the level on the Slave Select (/SS) pin is inconsistent with the actual mode of the device. MODF is set to warn that there may be a multi-master conflict for system control. In this case, the SPI system is affected in the following ways:

- An SPI receiver/error CPU interrupt request is generated
- The SPEN bit in SPSTA is cleared. This disables the SPI
- The MSTR bit in SPCON is cleared

When /SS Disable (SSDIS) bit in the SPCON register is cleared, the MODF flag is set when the /SS signal becomes '0'. However, as stated before, for a system with one Master, if the /SS pin of the master device is pulled low, there is no way that another master attempts to drive the network. In this case, to prevent the MODF flag from being set, software can set the SSDIS bit in the SPCON register and therefore making the /SS pin as a general-purpose I/O pin.

The user must clear the MODF bit by software, and enable SPEN in SPSTA register again for further communication, and enable MSTR bit to continue master mode.

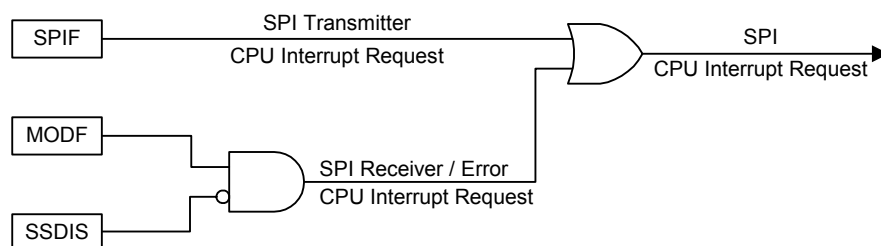
**Write Collision (WCOL):** A Write Collision (WCOL) flag in the SPSTA is set when a write to the SPDAT register is done during a transmit sequence. WCOL does not cause an interruption, and the transfer continues uninterrupted. The WCOL bit is cleared by software.

**Overrun Condition (RXOV):** An overrun condition occurs when the master or slave tries to send several data bytes and the slave or master has not cleared the SPIF bit issuing from the previous data byte transmitted. In this case, the receive shift register keep the byte that SPIF was lastly set, also the SPI device will not receive any further data until SPIF was cleared. The SPIF still keep on invoke interrupt before it is cleared, though the transmission can still be driven by SCK. RXOV does not cause an interruption, the RXOV bit is cleared by software.

## 12.8. Interrupts

Two SPI status flags can generate a CPU interrupt requests SPIF & MODF.

- Serial Peripheral data transfer flag, SPIF is set by hardware when a transfer has been completed.
- Mode Fault flag, MODF bit becomes set to indicate that the level on the /SS pin is inconsistent with the mode of the SPI. MODF with SSDIS reset will generate receiver/error CPU interrupt requests. When SSDIS is set, no MODF interrupt request is generated.



**FIG.12-7 SPI Interrupt Requests Generation**



## 13. Flash Program Memory

### 13.1. General Description

SH68F093 embeds 16K flash program memory for program code. The flash program memory provides electrical erasure and programming. Each of the sectors is equal to 1024 bytes. SH68F093 provides the user with SSP (self sector programming) function to write or erase the sector. To prevent the data from carelessly writing information block, the user must do five states (S0, S1, S2, S3, and S4) in sequence. If the dedicated conditions are not met from IBCON1 to IBCON5, the SSP will be terminated. Used to enable/disable the read operation through MOVC instruction, user want to read other sectors.

**Note:** The 0<sup>th</sup> sector (0000H - 03FFH) and the last 64 bytes for customer block (3FC0H - 3FFFH) and it cannot be rewritten before sector erase.

### 13.2. Registers

00F2H	IBCON1	Initial Value	Information Block Control Register 1	
Bit[7:0]	IBCON1	00h	R/W	Enable or disable the information block controller E6H: Erase the information block 6EH: Write to the information block Other: don't care <b>Reset source: Hardware reset or WDT reset</b>

00F3H	IBCON2	Initial Value	Information Block Control Register 2	
Bit[7:4]	-	0000b	-	Reserved
Bit[3:0]	IBCON2[3:0]	0000b	R/W	5H: enter S1 Other: enter S0 <b>Reset source: Hardware reset or WDT reset</b>

00F4H	IBCON3	Initial Value	Information Block Control Register 3	
Bit[7:4]	-	0000b	-	Reserved
Bit[3:0]	IBCON3	00h	R/W	AH: enter S2 Other: enter S1 <b>Reset condition: common reset   IBCON2 ≠ 5H</b>

00F5H	IBCON4	Initial Value	Information Block Control Register 4	
Bit[7:4]	-	0000b	-	Reserved
Bit[3:0]	IBCON4	0000b	R/W	9H: enter S3 Other: enter S2 <b>Reset condition: common reset   IBCON2 ≠ 5H   IBCON3 ≠ AH</b>

00F6H	IBCON5	Initial Value	Information Block Control Register 5	
Bit[7:4]	-	0000b	-	Reserved
Bit[3:0]	IBCON5	0000b	R/W	6H: enter S4 Other: enter S3 <b>Reset condition: common reset   IBCON2 ≠ 5H   IBCON3 ≠ AH   IBCON4 ≠ 9H</b>



00F7H	XPAGE	Initial Value	System Registers		
Bit[7:2]	XPAGE[7:2]	000000b	R/W	Sector of the flash memory to be programmed 000000 - 001111: Sector #0 - Sector #15 ( <b>Note 1</b> ) 010000 - 111111: reserved <b>Reset source: Hardware reset or WDT reset</b>	
Bit[1:0]	XPAGE[1:0]	00b	R/W	High address of offset of the flash memory sector to be programmed ( <b>Note 2</b> ) <b>Reset source: Hardware reset or WDT reset</b>	

00FBH	IBOFFSET	Initial Value	Information Block Control Register		
Bit[7:0]	IBOFFSET	00h	R/W	Offset in the flash memory Low address of offset of the flash memory sector to be programmed ( <b>Note 2</b> ) <b>Reset source: Hardware reset or WDT reset</b>	

00FCH	IB_DATA	Initial Value	Data Register for Programming Information Block		
B[7:0]	IB_DATA	00h	R/W	<b>Reset source: Hardware reset or WDT reset</b>	

**Note1:** Define the number of sector in the XPAGE register

XPAGE	
XPAGE[7:2]	XPAGE[1:0]
The number of sector	OFFSET BYTE

**Note2:** Combine XPAGE[1:0] and IBOFFSET[7:0] to become 10-bit offset byte so that the user can assign one byte of the information block for writing or reading.

offset byte	High address		Low address of offset of the flash memory sector to be programmed							
	XPAGE[1:0]		IBOFFSET[7:0]							
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	1	0
:	:	:	:	:	:	:	:	:	:	:
1022	1	1	1	1	1	1	1	1	1	0
1023	1	1	1	1	1	1	1	1	1	1

**Notice:**

To successfully complete SSP programming, the user has to follow the steps below:

A. for WRITE programming the information block

Step 1: Disable interrupt

Step 2: Fill in the XPAGE and IBOFFSET for the corresponding sector

Step 3: Fill in IB\_DATA if programming is wanted

Step 4: Fill in IBCON1 - 5 sequentially

Step 5: Code/Data programming, CPU will be in IDLE mode

Step 6: Add 4 nops; if more bytes want to be programmed, go back to step 2.

Step 7: Enable interrupt

B. for ERASE programming the information block

Step 1: Disable interrupt

Step 2: Fill in the XPAGE for the corresponding sector

Step 3: Fill in IBCON1 - 5 sequentially

Step 4: Erase the information block, CPU will be in IDLE mode

Step 5: Add 4 nop; if one more sector want to be erased, go back to step 2.

Step 6: Enable interrupt

C. for Code Reading

Use "MOVC A, @A+DPTR" or "MOVC A, @A+PC"



### 13.3. Flash Control Flow

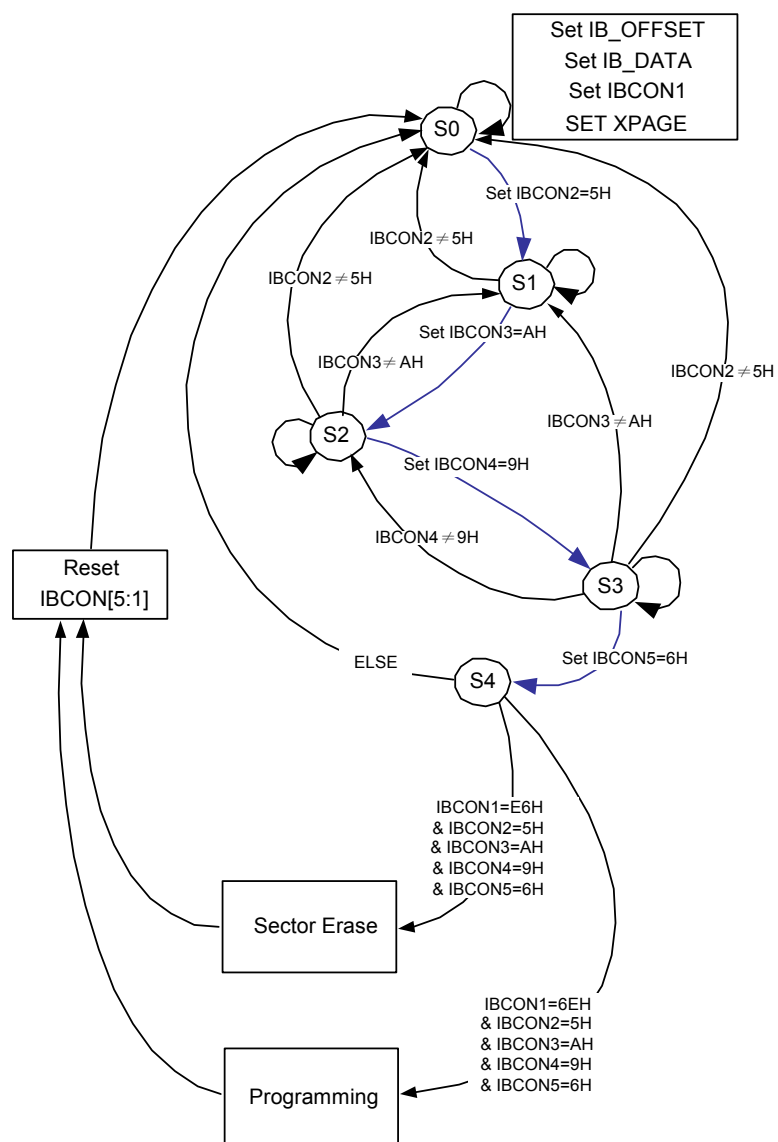


FIG.13-1 Flash Control Flow

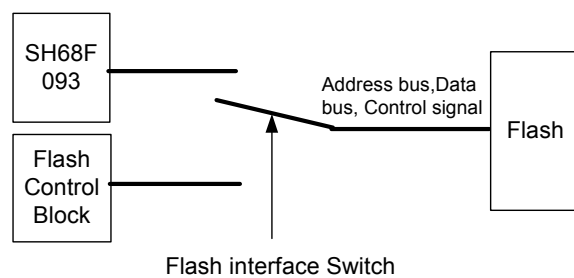


FIG.13-2 Flash interface Block Diagram

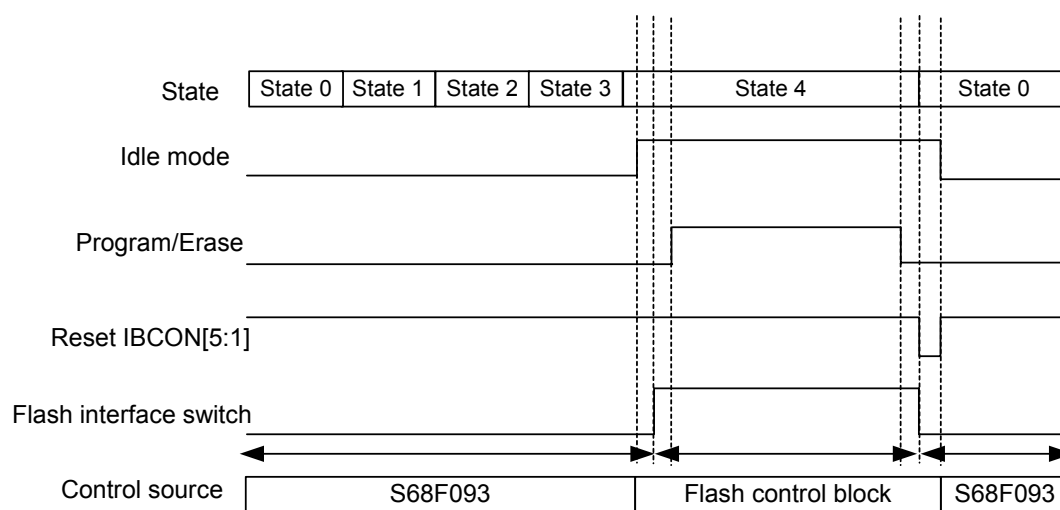


FIG.13-3 Flash Control Timing Diagram







15. Instruction Set Table

ARITHMETIC OPERATIONS				
Opcode	Description	Code	Byte	Cycle
ADD A, Rn	Add register to accumulator	0x28-0x2F	1	1
ADD A, direct	Add direct byte to accumulator	0x25	2	2
ADD A, @Ri	Add indirect RAM to accumulator	0x26-0x27	1	2
ADD A, #data	Add immediate data to accumulator	0x24	2	2
ADDC A, Rn	Add register to accumulator with carry flag	0x38-0x3F	1	1
ADDC A, direct	Add direct byte to A with carry flag	0x35	2	2
ADDC A, @Ri	Add indirect RAM to A with carry flag	0x36-0x37	1	2
ADDC A, #data	Add immediate data to A with carry flag	0x34	2	2
SUBB A, Rn	Subtract register from A with borrow	0x98-0x9F	1	1
SUBB A, direct	Subtract direct byte from A with borrow	0x95	2	2
SUBB A, @Ri	Subtract indirect RAM from A with borrow	0x96-0x97	1	2
SUBB A, #data	Subtract immediate data from A with borrow	0x94	2	2
INC A	Increment accumulator	0x04	1	1
INC Rn	Increment register	0x08-0x0F	1	2
INC direct	Increment direct byte	0x05	2	3
INC @Ri	Increment indirect RAM	0x06-0x07	1	3
DEC A	Decrement accumulator	0x14	1	1
DEC Rn	Decrement register	0x18-0x1F	1	2
DEC direct	Decrement direct byte	0x15	1	3
DEC @Ri	Decrement indirect RAM	0x16-0x17	2	3
INC DPTR	Increment data pointer	0xA3	1	4
MUL AB 8 X 8	Multiply A and B	0xA4	1	11
MUL AB 16 X 8				20
DIV AB 8 / 8	Divide A by B	0x84	1	11
DIV AB 16 / 8				20
DA A	Decimal adjust accumulator	0xD4	1	1



<b>LOGIC OPERATIONS</b>				
<b>Opcode</b>	<b>Description</b>	<b>Code</b>	<b>Byte</b>	<b>Cycle</b>
ANL A, Rn	AND register to accumulator	0x58-0x5F	1	1
ANL A, direct	AND direct byte to accumulator	0x55	2	2
ANL A, @Ri	AND indirect RAM to accumulator	0x56-0x57	1	2
ANL A, #data	AND immediate data to accumulator	0x54	2	2
ANL direct, A	AND accumulator to direct byte	0x52	2	3
ANL direct, #data	AND immediate data to direct byte	0x53	3	3
ORL A, Rn	OR register to accumulator	0x48-0x4F	1	1
ORL A, direct	OR direct byte to accumulator	0x45	2	2
ORL A, @Ri	OR indirect RAM to accumulator	0x46-0x47	1	2
ORL A, #data	OR immediate data to accumulator	0x44	2	2
ORL direct, A	OR accumulator to direct byte	0x42	2	3
ORL direct, #data	OR immediate data to direct byte	0x43	3	3
XRL A, Rn	Exclusive OR register to accumulator	0x68-0x6F	1	1
XRL A, direct	Exclusive OR direct byte to accumulator	0x65	2	2
XRL A, @Ri	Exclusive OR indirect RAM to accumulator	0x66-0x67	1	2
XRL A, #data	Exclusive OR immediate data to accumulator	0x64	2	2
XRL direct, A	Exclusive OR accumulator to direct byte	0x62	2	3
XRL direct, #data	Exclusive OR immediate data to direct byte	0x63	3	3
CLR A	Clear accumulator	0xE4	1	1
CPL A	Complement accumulator	0xF4	1	1
RL A	Rotate accumulator left	0x23	1	1
RLC A	Rotate accumulator left through carry	0x33	1	1
RR A	Rotate accumulator right	0x03	1	1
RRC A	Rotate accumulator right through carry	0x13	1	1
SWAP A	Swap nibbles within the accumulator	0xC4	1	4

<b>BOOLEAN MANIPULATION</b>				
<b>Opcode</b>	<b>Description</b>	<b>Code</b>	<b>Byte</b>	<b>Cycle</b>
CLR C	Clear carry flag	0xC3	1	1
CLR bit	Clear direct bit	0xC2	2	3
SETB C	Set carry flag	0xD3	1	1
SETB bit	Set direct bit	0xD2	2	3
CPL C	Complement carry flag	0xB3	1	1
CPL bit	Complement direct bit	0xB2	2	3
ANL C, bit	AND direct bit to carry flag	0x82	2	2
ANL C, /bit	AND complement of direct bit to carry	0xB0	2	2
ORL C, bit	OR direct bit to carry flag	0x72	2	2
ORL C, /bit	OR complement of direct bit to carry	0xA0	2	2
MOV C, bit	Move direct bit to carry flag	0xA2	2	2
MOV bit, C	Move carry flag to direct bit	0x92	2	3



<b>DATA TRANSFERS</b>				
<b>Opcode</b>	<b>Description</b>	<b>Code</b>	<b>Byte</b>	<b>Cycle</b>
MOV A, Rn	Move register to accumulator	0xE8-0xEF	1	1
MOV A, direct	Move direct byte to accumulator	0xE5	2	2
MOV A, @Ri	Move indirect RAM to accumulator	0xE6-0xE7	1	2
MOV A, #data	Move immediate data to accumulator	0x74	2	2
MOV Rn, A	Move accumulator to register	0xF8-0xFF	1	2
MOV Rn, direct	Move direct byte to register	0xA8-0xAF	2	3
MOV Rn, #data	Move immediate data to register	0x78-0x7F	2	2
MOV direct, A	Move accumulator to direct byte	0xF5	2	2
MOV direct, Rn	Move register to direct byte	0x88-0x8F	2	2
MOV direct1, direct2	Move direct byte to direct byte	0x85	3	3
MOV direct, @Ri	Move indirect RAM to direct byte	0x86-0x87	2	3
MOV direct, #data	Move immediate data to direct byte	0x75	3	3
MOV @Ri, A	Move accumulator to indirect RAM	0xF6-0xF7	1	2
MOV @Ri, direct	Move direct byte to indirect RAM	0xA6-0xA7	2	3
MOV @Ri, #data	Move immediate data to indirect RAM	0x76-0x77	2	2
MOV DPTR, #data16	Load data pointer with a 16-bit constant	0x90	3	3
MOVC A, @A+DPTR	Move code byte relative to DPTR to A	0x93	1	7
MOVC A, @A+PC	Move code byte relative to PC to A	0x83	1	8
MOVX A, @Ri	Move external RAM (8-bit address) to A	0xE2-0xE3	1	5
MOVX A, @DPTR	Move external RAM (16-bit address) to A	0xE0	1	6
MOVX @Ri, A	Move A to external RAM (8-bit address)	0xF2-F3	1	4
MOVX @DPTR, A	Move A to external RAM (16-bit address)	0xF0	1	5
PUSH direct	Push direct byte onto stack	0xC0	2	5
POP direct	Pop direct byte from stack	0xD0	2	4
XCH A, Rn	Exchange register with accumulator	0xC8-0xCF	1	3
XCH A, direct	Exchange direct byte with accumulator	0xC5	2	4
XCH A, @Ri	Exchange indirect RAM with accumulator	0xC6-0xC7	1	4
XCHD A, @Ri	Exchange low-order nibble indirect RAM with A	0xD6-0xD7	1	4



<b>PROGRAM BRANCHES</b>				
<b>Opcode</b>	<b>Description</b>	<b>Code</b>	<b>Byte</b>	<b>Cycle</b>
ACALL addr11	Absolute subroutine call	0x11-0xF1	2	7
LCALL addr16	Long subroutine call	0x12	3	7
RET	Return from subroutine	0x22	1	8
RETI	Return from interrupt	0x32	1	8
AJMP addr11	Absolute jump	0x01-0xE1	2	4
LJMP addr16	Long jump	0x02	3	5
SJMP rel	Short jump (relative address)	0x80	2	4
JMP @A+DPTR	Jump indirect relative to the DPTR	0x73	1	6
JZ rel (not taken)	Jump if accumulator is zero	0x60	2	3
JZ rel (taken)				5
JNZ rel (not taken)	Jump if accumulator is not zero	0x70	2	3
JNZ rel (taken)				5
JC rel (not taken)	Jump if carry flag is set	0x40	2	2
JC rel (taken)				4
JNC (not taken)	Jump if carry flag is not set	0x50	2	2
JNC (taken)				4
JB bit, rel (not taken)	Jump if direct bit is set	0x20	3	4
JB bit, rel (taken)				6
JNB bit, rel (not taken)	Jump if direct bit is not set	0x30	3	4
JNB bit, rel (taken)				6
JBC bit, rel (not taken)	Jump if direct bit is set and clear bit	0x10	3	4
JBC bit, rel (taken)				6
CJNE A, direct rel (not taken)	Compare direct byte to A and jump if not equal	0xB5	3	4
CJNE A, direct rel (taken)				6
CJNE A, #data rel (not taken)	Compare immediate to A and jump if not equal	0xB4	3	4
CJNE A, #data rel (taken)				6
CJNE Rn, #data rel (not taken)	Compare immediate to reg. And jump if not equal	0xB8-0xBF	3	4
CJNE Rn, #data rel (taken)				6
CJNE @Ri, #data rel (not taken)	Compare immediate to Ri and jump if not equal	0xB6-0xB7	3	4
CJNE @Ri, #data rel (taken)				6
DJNZ Rn, rel (not taken)	Decrement register and jump if not zero	0xD8-0xDF	2	3
DJNZ Rn, rel (taken)				5
DJNZ direct, rel (not taken)	Decrement direct byte and jump if not zero	0xD5	3	4
DJNZ direct, rel (taken)				6
NOP	No operation	0	1	1



## DC/AC Specifications

### Absolute Maximum Rating\*

- Operating Voltage ( $V_{BAT}$ ) . . . . . +1.1V to +1.7V
- Input/Output Voltage . . . . . GND - 0.2V to  $V_{BAT}$  + 0.2V
- Operating Ambient Temperature . . . . . -25°C to 60°C
- Storage Temperature . . . . . -55°C to +125°C

### \*Comments

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. These are stress ratings only. Functional operation of this device at these or any other conditions above those indicated in the operational sections of this specification is not implied and exposure to absolute maximum rating conditions for extended periods may affect device reliability.

**DC Electrical Characteristics** (Enable built-in DC-DC,  $V_{BAT} = 1.5V$ , GND = 0V,  $T_A = 25^\circ C$ , unless otherwise noted)

Main Power						
Symbol	Parameters	Min.	Typ.	Max.	Unit	Conditions
$V_{BAT}$	Operating Voltage 1	1.1	1.5	1.7	V	Battery output voltage (enable built-in DC-DC)
$I_{OP1}$	Operating Current 1	-	3.7	4.2	mA	No load ( $F_{SYS} = 8MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 2.7V$ )
$I_{OP2}$	Operating Current 2	-	2.5	3	mA	No load ( $F_{SYS} = 4MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 2.7V$ )
$I_{OP3}$	Operating Current 3	-	1.9	2.4	mA	No load ( $F_{SYS} = 2MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 2.7V$ )
$I_{OP4}$	Operating Current 4	-	1.5	2	mA	No load ( $F_{SYS} = 1MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 2.7V$ )
$I_{OP5}$	Operating Current 5	-	3.9	4.4	mA	No load ( $F_{SYS} = 8MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 3.3V$ )
$I_{OP6}$	Operating Current 6	-	2.5	3	mA	No load ( $F_{SYS} = 4MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 3.3V$ )
$I_{OP7}$	Operating Current 7	-	1.9	2.4	mA	No load ( $F_{SYS} = 2MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 3.3V$ )
$I_{OP8}$	Operating Current 8	-	1.5	2	mA	No load ( $F_{SYS} = 1MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 3.3V$ )
$I_{IDLE1}$	Idle Current 1	-	2	2.5	mA	No load ( $F_{SYS} = 8MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 2.7V$ )
$I_{IDLE2}$	Idle Current 2	-	1.6	2.1	mA	No load ( $F_{SYS} = 4MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 2.7V$ )
$I_{IDLE3}$	Idle Current 3	-	1.4	1.9	mA	No load ( $F_{SYS} = 2MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 2.7V$ )
$I_{IDLE4}$	Idle Current 4	-	1.3	1.8	mA	No load ( $F_{SYS} = 1MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 2.7V$ )
$I_{IDLE5}$	Idle Current 5	-	2	2.5	mA	No load ( $F_{SYS} = 8MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 3.3V$ )
$I_{IDLE6}$	Idle Current 6	-	1.6	2.1	mA	No load ( $F_{SYS} = 4MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 3.3V$ )
$I_{IDLE7}$	Idle Current 7	-	1.4	1.9	mA	No load ( $F_{SYS} = 2MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 3.3V$ )
$I_{IDLE8}$	Idle Current 8	-	1.3	1.8	mA	No load ( $F_{SYS} = 1MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 3.3V$ )
$I_{PD1}$	Power down Current 1	-	605	805	uA	No Load ( $F_{SYS} = 8MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 2.7V$ ), Enable built-in DC-DC, In Power-down mode ( <b>PWM mode</b> ): Enable LVR1
$I_{PD2}$	Power down Current 2	-	600	800	uA	No Load ( $F_{SYS} = 8MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 2.7V$ ), Enable built-in DC-DC, In Power-down mode ( <b>PWM mode</b> ): Disable LVR1
$I_{PD3}$	Power down Current 3	-	255	355	uA	No Load ( $F_{SYS} = 8MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 2.7V$ ), Enable built-in DC-DC, In Power-down mode ( <b>VFM mode</b> ): Enable LVR1
$I_{PD4}$	Power down Current 4	-	250	350	uA	No Load ( $F_{SYS} = 8MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 2.7V$ ), Enable built-in DC-DC, In Power-down mode ( <b>VFM mode</b> ): Disable LVR1
$I_{PD5}$	Power down Current 5	-	10	-	uA	No Load ( $F_{SYS} = 8MHz$ , $V_{BAT} = 1.5V$ , $V_{OUT} = 1.5V$ ), Enable built-in DC-DC, In Power-down mode ( <b>STOP mode</b> ): Enable LVR1



(continued)

Main Power						
Symbol	Parameters	Min.	Typ.	Max.	Unit	Conditions
I <sub>PD6</sub>	Power down Current 6	-	10	-	uA	No Load (F <sub>SYS</sub> = 8Mhz, V <sub>BAT</sub> = 1.5V, V <sub>OUT</sub> = 1.5V), Enable built-in DC-DC, In Power-down mode ( <b>STOP mode</b> ): Disable LVR1
I <sub>PD7</sub>	Power down Current 7	-	1.005	1.205	mA	No Load (F <sub>SYS</sub> = 8Mhz, V <sub>BAT</sub> = 1.5V, V <sub>OUT</sub> = 3.3V), Enable built-in DC-DC, In Power-down mode ( <b>PWM mode</b> ), Enable LVR1
I <sub>PD8</sub>	Power down Current 8	-	1	1.2	mA	No Load (F <sub>SYS</sub> = 8Mhz, V <sub>BAT</sub> = 1.5V, V <sub>OUT</sub> = 3.3V), Enable built-in DC-DC, In Power-down mode ( <b>PWM mode</b> ): Disable LVR1
I <sub>PD9</sub>	Power down Current 9	-	355	455	uA	No Load (F <sub>SYS</sub> = 8Mhz, V <sub>BAT</sub> = 1.5V, V <sub>OUT</sub> = 3.3V), Enable built-in DC-DC, In Power-down mode ( <b>VFM mode</b> ): Enable LVR1
I <sub>PD10</sub>	Power down Current 10	-	350	450	uA	No Load (F <sub>SYS</sub> = 8Mhz, V <sub>BAT</sub> = 1.5V, V <sub>OUT</sub> = 3.3V), Enable built-in DC-DC, In Power-down mode ( <b>VFM mode</b> ): Disable LVR1
I <sub>PD11</sub>	Power down Current 11	-	10	-	uA	No Load (F <sub>SYS</sub> = 8Mhz, V <sub>BAT</sub> = 1.5V, V <sub>OUT</sub> = 1.5V), Enable built-in DC-DC, In Power-down mode ( <b>STOP mode</b> ): Enable LVR1
I <sub>PD12</sub>	Power down Current 12	-	10	-	uA	No Load (F <sub>SYS</sub> = 8Mhz, V <sub>BAT</sub> = 1.5V, V <sub>OUT</sub> = 1.5V), Enable built-in DC-DC, In Power-down mode ( <b>STOP mode</b> ): Disable LVR1

**DC Electrical Characteristics** (Disable built-in DC-DC, V<sub>OUT</sub> = 3V, GND = 0V, T<sub>A</sub> = 25°C, unless otherwise noted)

Main Power						
Symbol	Parameters	Min.	Typ.	Max.	Unit	Conditions
V <sub>OUT</sub>	Operating Voltage 2	2.4		3.6	V	(disable built-in DC-DC)
I <sub>OP9</sub>	Operating Current 9	-	1.8	2.2	mA	No load (F <sub>SYS</sub> = 8Mhz, V <sub>OUT</sub> = 3V)
I <sub>OP10</sub>	Operating Current 10	-	1.2	1.6	mA	No load (F <sub>SYS</sub> = 4Mhz, V <sub>OUT</sub> = 3V)
I <sub>OP11</sub>	Operating Current 11	-	0.9	1.3	mA	No load (F <sub>SYS</sub> = 2Mhz, V <sub>OUT</sub> = 3V)
I <sub>OP12</sub>	Operating Current 12	-	0.8	1.1	mA	No load (F <sub>SYS</sub> = 1Mhz, V <sub>OUT</sub> = 3V)
I <sub>IDLE9</sub>	Idle Current 9	-	1	1.4	mA	No load (F <sub>SYS</sub> = 8Mhz, V <sub>OUT</sub> = 3V)
I <sub>IDLE10</sub>	Idle Current 10	-	0.9	1.2	mA	No load (F <sub>SYS</sub> = 4Mhz, V <sub>OUT</sub> = 3V)
I <sub>IDLE11</sub>	Idle Current 11	-	0.8	1.1	mA	No load (F <sub>SYS</sub> = 2Mhz, V <sub>OUT</sub> = 3V)
I <sub>IDLE12</sub>	Idle Current 12	-	0.7	1	mA	No load (F <sub>SYS</sub> = 1Mhz, V <sub>OUT</sub> = 3V)
I <sub>PD13</sub>	Power down Current 13	-	65	85	uA	No Load (F <sub>SYS</sub> = 8Mhz, V <sub>OUT</sub> = 3V) Disable built-in DC-DC, In Power-down mode, enable wake-up timer, Enable LVR1
I <sub>PD14</sub>	Power down Current 14	-	60	80	uA	No Load (F <sub>SYS</sub> = 8Mhz, V <sub>OUT</sub> = 3V) Disable built-in DC-DC, In Power-down mode, enable wake-up timer, Disable LVR1
I <sub>PD15</sub>	Power down Current 15	-	10	-	uA	No Load (F <sub>SYS</sub> = 8Mhz, V <sub>OUT</sub> = 3V) Disable built-in DC-DC, In Power-down mode: disable wake-up timer, Enable LVR1
I <sub>PD16</sub>	Power down Current 16	-	10	-	uA	No Load (F <sub>SYS</sub> = 8Mhz, V <sub>OUT</sub> = 3V) Disable built-in DC-DC, In Power-down mode: disable wake-up timer, Disable LVR1



(continued)

DC-DC Converter						
Symbol	Parameters	Min.	Typ.	Max.	Unit	Conditions
V <sub>OUT</sub>	Output Voltage	-	3.3 /2.7	-	V	Depend on VSEL_DC[1:0], not include output Variation. V <sub>OUT</sub> = 2.7V(default)/V <sub>OUT</sub> = 3.3V
V <sub>BAT</sub>	Operation Start-up Voltage	1.1	-	1.7	V	
V /V	V <sub>OUT</sub> Variation	-	-	3	%	V <sub>OUT</sub> - Target V <sub>OUT</sub>  /Target V <sub>OUT</sub> , I <sub>OUT</sub> = 5mA
I <sub>OUTMAX1</sub>	Maximum output current 1	70	-	-	mA	V <sub>BAT</sub> = 1.5V, V <sub>OUT</sub> = 3.3V
I <sub>OUTMAX2</sub>	Maximum output current 2	100	-	-	mA	V <sub>BAT</sub> = 1.5V, V <sub>OUT</sub> = 2.7V
I <sub>OUTMIN1</sub>	Minimum output current 1	50	-	-	mA	V <sub>BAT</sub> = 1.1V, V <sub>OUT</sub> = 3.3V
I <sub>OUTMIN2</sub>	Minimum output current 2	60	-	-	mA	V <sub>BAT</sub> = 1.1V, V <sub>OUT</sub> = 2.7V
R <sub>SWON</sub>	LX Switch-On Resistance	-	1.4	2.4	Ohm	Same as I <sub>DD1</sub> . V <sub>LX</sub> = 0.4V. (Note 1)
T <sub>DC-DCST</sub>	DC-DC start up time	-	4	6	ms	From 10% V <sub>OUT</sub> to 90% V <sub>OUT</sub>
EFFI1	Efficiency 1 (Note 2)	-	85	-	%	V <sub>LX</sub> = 1.5V, V <sub>OUT</sub> = 2.7V, I <sub>OUT</sub> = 30mA
EFFI2	Efficiency 2 (Note 2)	-	85	-	%	V <sub>LX</sub> = 1.5V, V <sub>OUT</sub> = 3.3V, I <sub>OUT</sub> = 30mA

**Note1:** V<sub>LX</sub> means the DC-DC Input voltage

**Note2:** EFFI =  $\{[(\text{Output Voltage}) \times (\text{Output Current})] \div [(\text{Input Voltage}) \times (\text{Input Current})]\} \times 100\%$

Regulator						
Symbol	Parameters	Min.	Typ.	Max.	Unit	Conditions
V <sub>O</sub>	Output Voltage	1.7	1.8	1.9	V	Output voltage used for RD8051 core

GPIO						
Symbol	Parameters	Min.	Typ.	Max.	Unit	Conditions
V <sub>OH</sub>	Output High Voltage (Port0, Port1, and Port2)	V <sub>OUT</sub> - 0.6	-	-	V	@V <sub>OUT</sub> = 2.7/3.3V (I <sub>OH1</sub> = -4mA)
V <sub>OL</sub>	Output Low Voltage (Port0, Port1, and Port2)	-	-	0.4	V	@V <sub>OUT</sub> = 2.7/3.3V (I <sub>OL1</sub> = 4mA)
R <sub>RST</sub>	RSTB internal pull-up resistor	-	30	-	kohm	@0V
V <sub>STIH</sub>	Schmitt Trigger Input High Voltage (all GPIO)	0.8 V <sub>OUT</sub>	-	V <sub>OUT</sub>	V	@V <sub>OUT</sub> = 2.7V/3.3V
V <sub>STIL</sub>	Schmitt Trigger Input Low Voltage (all GPIO)	GND	-	0.2 V <sub>OUT</sub>	V	@V <sub>OUT</sub> = 2.7V/3.3V

Reset						
Symbol	Parameters	Min.	Typ.	Max.	Unit	Conditions
V <sub>LVR1</sub>	LVR1 Threshold Voltage	1.4	1.5	1.6	V	Testing Pulse Width > T <sub>PW(LVR)</sub>
V <sub>LVR2</sub>	LVR2 Threshold Voltage	1.8	1.9	2.0	V	
V <sub>UT(RSTB)</sub>	Schmitt Upper-Threshold Voltage for external reset	0.8 V <sub>OUT</sub>		V <sub>OUT</sub>	V	@V <sub>OUT</sub> = 2.7V/3.3V
V <sub>LT(RSTB)</sub>	Schmitt Lower-Threshold Voltage for external reset	GND		0.2 V <sub>OUT</sub>	V	@V <sub>OUT</sub> = 2.7V/3.3V

Power Down Mode						
Symbol	Parameters	Min.	Typ.	Max.	Unit	Conditions
V <sub>PDL1</sub>	Input Low Voltage 1	-	0.3 V <sub>OUT</sub>	-	V	P2.1 - P2.3
T <sub>Bounce</sub>	Bounce time After triggered	-	15	-	ms	P2.1 - P2.3



**AC Electrical Characteristics** ( $V_{BAT} = 1.5V$ ,  $GND = 0V$ ,  $T_A = 25^\circ C$ ,  $F_{SYS} = 8MHz$  (internal RC), unless otherwise noted)

Oscillator						
Symbol	Parameters	Min.	Typ.	Max.	Unit	Conditions
$F_{RC1}$	Internal RC Frequency 1	7.6	8	8.4	MHz	$\pm 5\% @ V_{OUT} = 2.7V/3.3V$ , $T_A = 0 - 60^\circ C$
$F_{RC2}$	Internal RC Frequency 2	30.08	32	33.92	KHz	$\pm 6\% @ V_{OUT} = 2.7V/3.3V$ , $T_A = 0 - 60^\circ C$

Reset						
Symbol	Parameters	Min.	Typ.	Max.	Unit	Conditions
$T_{PW(RSTB)}$	RESETB Input Low Pulse Width	-	$2^{13}$	-	$T_{SYS}$	$F_{SYS} = 8MHz$
$T_{PW(LVR1)}$	Drop-Down Pulse Width for LVR1	-	$2^9$	-	$T_{SYS}$	$F_{SYS} = 8MHz$ , Power < $V_{LT(LVR1)}$
$T_{RST(POR)}$	Internal Power-on Reset Hold Time	-	11	-	ms	$F_{SYS} = 8MHz$
$T_{RST(WDT)}$	Internal Watch-Dog Reset Hold Time	-	0.5	-	ms	$F_{SYS} = 8MHz$
$T_{RST(RSTB)}$	Internal External Reset Hold Time	-	11	-	ms	$F_{SYS} = 8MHz$
$T_{Period(RSU)}$	Resume Reset Period	-	24	-	$\mu s$	16 clock/2MHz plus 128clock/8MHz RC Resonator

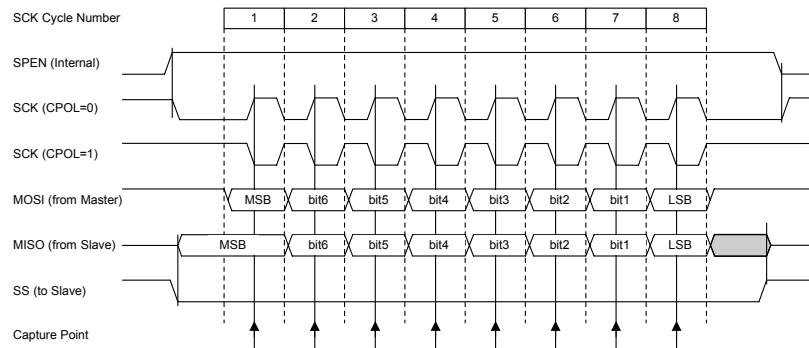
GPIO						
Symbol	Parameters	Min.	Typ.	Max.	Unit	Conditions
$t_{PW(Ext0/Ext1)}$	Noise cancellation for EXT0/EXT01	-	$2^2$	-	$T_{SYS}$	$F_{SYS} = 8MHz$

Voltage Comparator						
Symbol	Parameters	Min.	Typ.	Max.	Unit	Conditions
$T_{CONV}$	Voltage comparator compare time	-	-	5	$\mu s$	
$VC_{VAR}$	Voltage reference variation			$\pm 5$	%	@ 0.8V, 0.9V, and 1.0V





**SPI Interface Timing Requirements (Master Mode)**



SPI Interface						
Symbol	Parameters	Min.	Typ.	Max.	Unit	Conditions
SCK1	SCK frequency	-	-	2	MHz	@ Master mode ( $f_{sck} = 1/t_{sck}$ )
SCK2	SCK high/low	-	50%	-		duty cycle @ Master mode
SCK3	SCK rise/fall time	-	3.6	100	ns	@ Master mode ( $C_{load} = 5pF$ )
SCK4	MOSI to SCK	120	$0.5 \cdot t_{sck}$	-	ns	@ Master mode
SCK5	SCK to MOSI (hold)	200	-	-	ns	@ Master mode
SCK6	SCK to MISO (hold)		10	120	ns	@ Master mode



**SH68F093**

---

**Ordering Information**

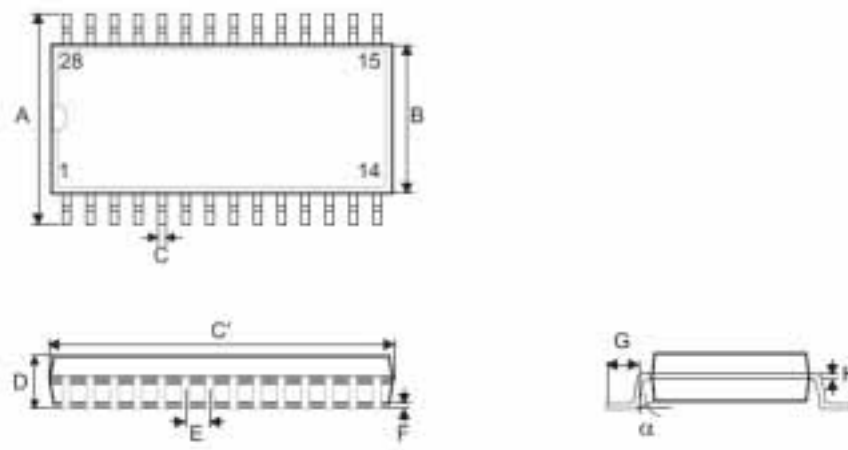
Part No.	Package
SH68F093H	CHIPFORM
SH68F093M	28 SOP



**Package Information**

**28-Pin SOP Outline Dimensions**

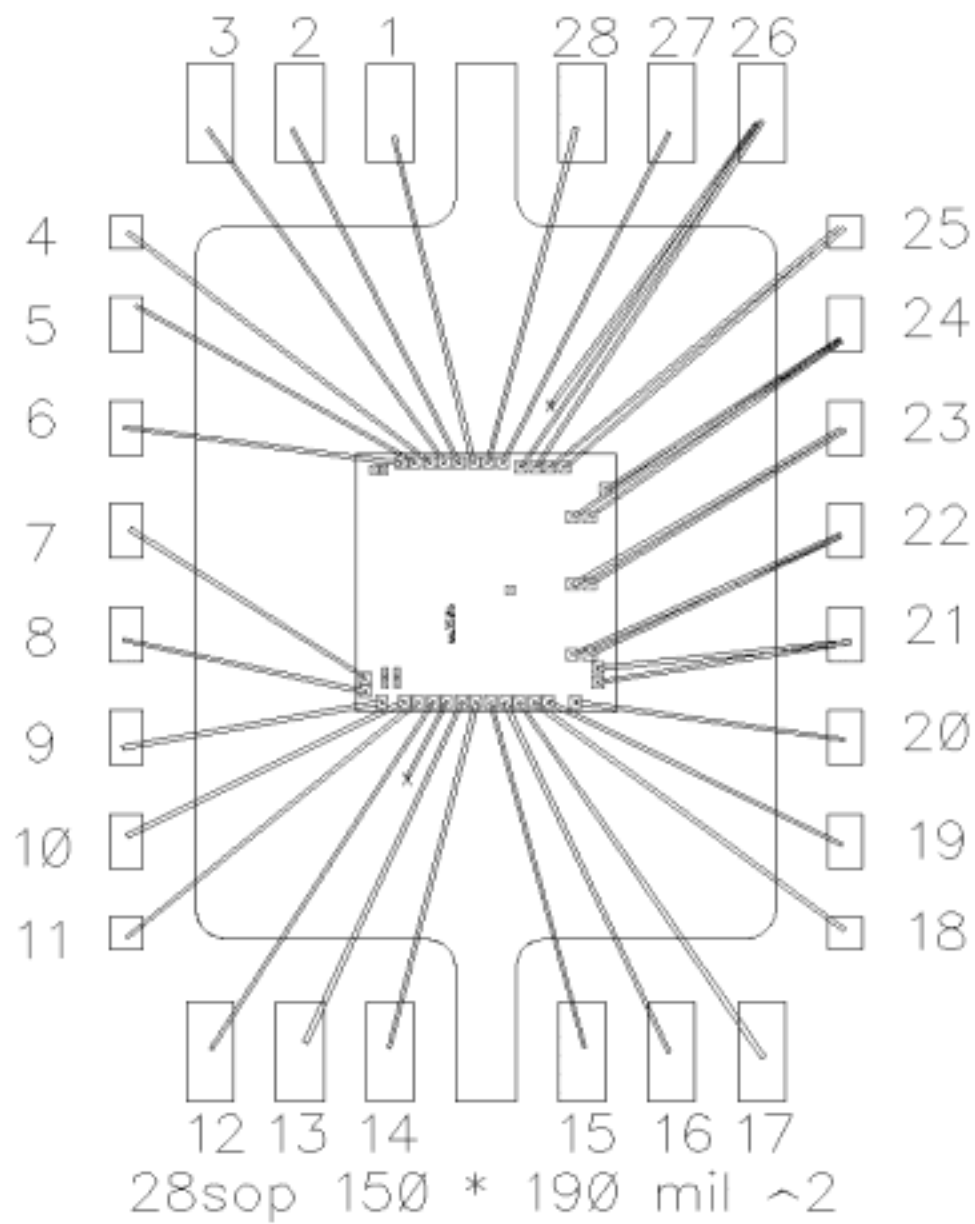
unit: mil



Symbol	Dimensions in mil		
	Min.	Nom.	Max.
A	394	—	419
B	290	—	300
C	14	—	20
C'	697	—	713
D	92	—	104
E	—	50	—
F	4	—	—
G	16	—	50
H	4	—	12
$\alpha$	0°	—	10°



Bonding Diagram





## SH68F093

### Pad Location

unit:  $\mu\text{m}$

Pad NO.	Pad Name	X	Y	28 SOP Pin NO.
1	port0[0]	809.64	81.98	1
2	port0[1]	809.64	176.98	2
3	port0[2]	809.64	271.98	3
4	port0[3]	809.64	366.98	4
5	port0[4]	809.64	461.98	5
6	port0[5]	809.64	556.98	6
7	port0[6]	-643.46	788.64	7
8	port0[7]	-738.46	788.64	8
9	port1[0]	-809.64	681.53	9
10	port1[1]	-809.64	539.1	10
11	port1[2]	-809.64	444.09	11
12	port1[3]	-809.64	349.09	12
13	GND	-809.64	254.09	bound to frame
14	port1[4]	-809.64	159.09	13
15	port1[5]	-809.64	64.09	14
16	port1[6]	-809.64	-30.91	15
17	port1[7]	-809.64	-125.92	16
18	port2[0]	-809.64	-220.91	17
19	port2[1]	-809.64	-315.91	18
20	port2[2]	-809.64	-410.91	19
21	port2[3]	-809.64	-586.64	20
22	GND3	-674.76	-731.64	21
23	GND2	-579.76	-731.64	21
24	GND	-484.76	-681.6	22
25	GND	-484.76	-571.6	22
26	LX	-15.23	-681.6	23
27	LX	-15.23	-571.6	23
28	VOUT	442.77	-681.6	24
29	VOUT	446.77	-571.6	24
30	VDDIN	632.96	-788.64	24
31	V18	779.64	-517.02	25
32	VCC	779.64	-422.02	25
33	GND	779.64	-327.02	26
34	GND	779.64	-232.02	26
35	VIN	809.64	-108.02	27
36	RSTB	809.64	-13.02	28

**Data Sheet Revision History**

Revision No.	History	Date
2.0	Modify I <sub>PD5</sub> /I <sub>PD6</sub> /I <sub>PD11</sub> /I <sub>PD12</sub> /I <sub>PD15</sub> /I <sub>PD16</sub> Current to 10uA Typ	May. 2010
1.0	Original	Aug. 2009