SINO WEALTH

SH68F86

Low Speed USB Micro-controller

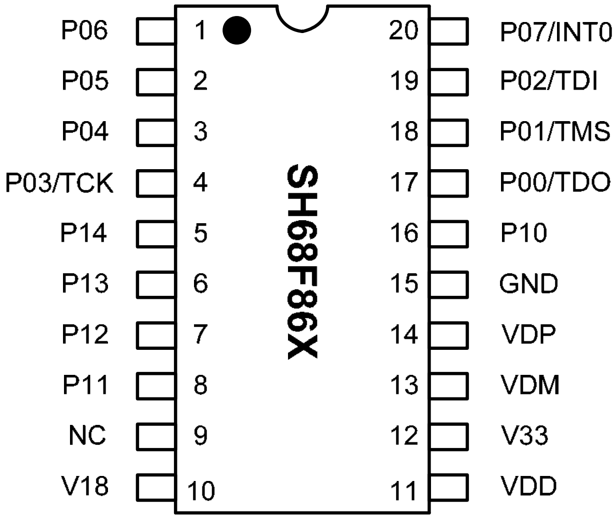
Features

- 8-bit CMOS Micro-Processor (uP) core
  - Instruction set is compatible with standard 8051
  - Build-in 12MHz RC for MCU clock
- Memory
  - 16K Bytes Flash
  - Support USB Self-Sectors-Programming (SSP)
  - 256 bytes internal data memory
- Operation voltage 4.4V - 5.25V
- 3.3V regulator output
  - Maximum driving current 35mA
- Up to 13 general purpose I/O ports
- Interrupt
  - 10 vectors interrupt structure
  - Timer0 & 1, USB, EX0
- Two 8-Bit Base Timer
- 1 external Interrupt: EX0
- Build-in 32KHz oscillator for programmable wake up timer
- USB Specification Compliance
  - Complies with USB specification 1.1
  - Support one Low-Speed USB Device Address with 3 endpoints (endpoint 0, 1, and 2)
  - Built-in 1.5Kohms USB pull-up resistor with DM
- Built-in Watch Dog Timer (WDT)
- Reset
  - Hardware reset
  - Power-on reset, Low-voltage reset
  - USB reset
  - Watch-Dog reset
  - Resume reset
- Two power-reducing modes:
  - IDLE mode
  - Power-Down mode
- Package:
  - 24 pad Chip Form
  - 20 pin TSSOP

General Description

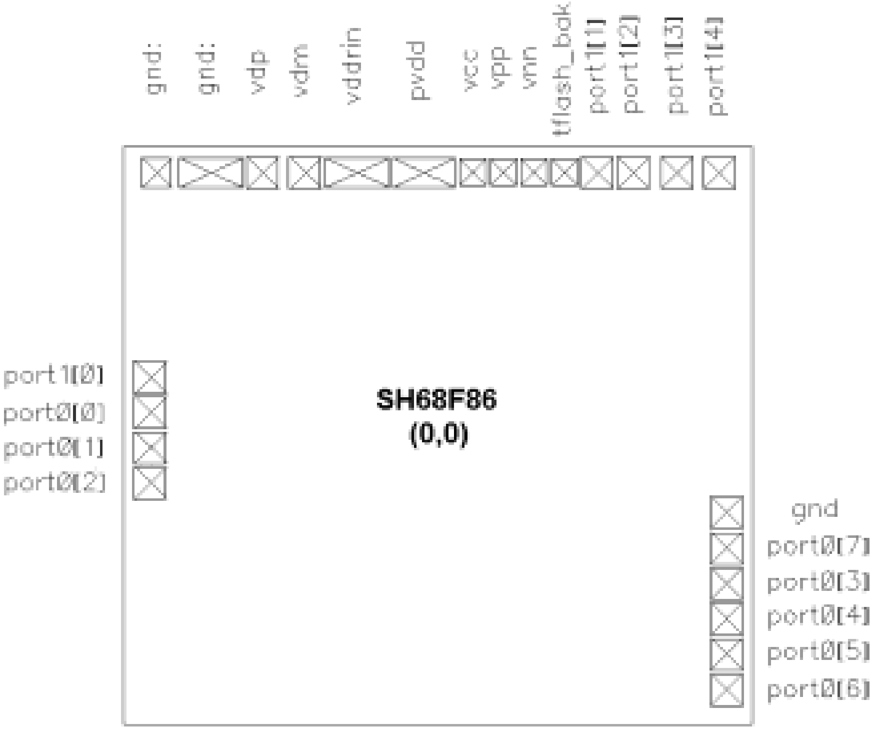
The SH68F86 is an enhanced 8051-compatiable micro-controller with integrated low-speed USB SIE, Transceiver and data FIFO, build-in 1.8v for CPU core and 3.3v for USB transceiver, IO and others. With on-chip 16K bytes flash memory and internal 256 bytes data memory suit for variable applications. Two 8-bits timer, programmable Watch-dog timer and Wake-up timer provide great F/W facility and power saving mode operation, 13 selectable GPIO in 20 pin TSSOP package, build-in 12MHz oscillator to eliminate external crystal. Besides, build-in 32K oscillator, POR and LVR circuit will further saving your external components cost. The SH68F86 is a highly integrated MCU designed for cost effective applications. Application can cover such items as Dongle and others.

Pin Configuration

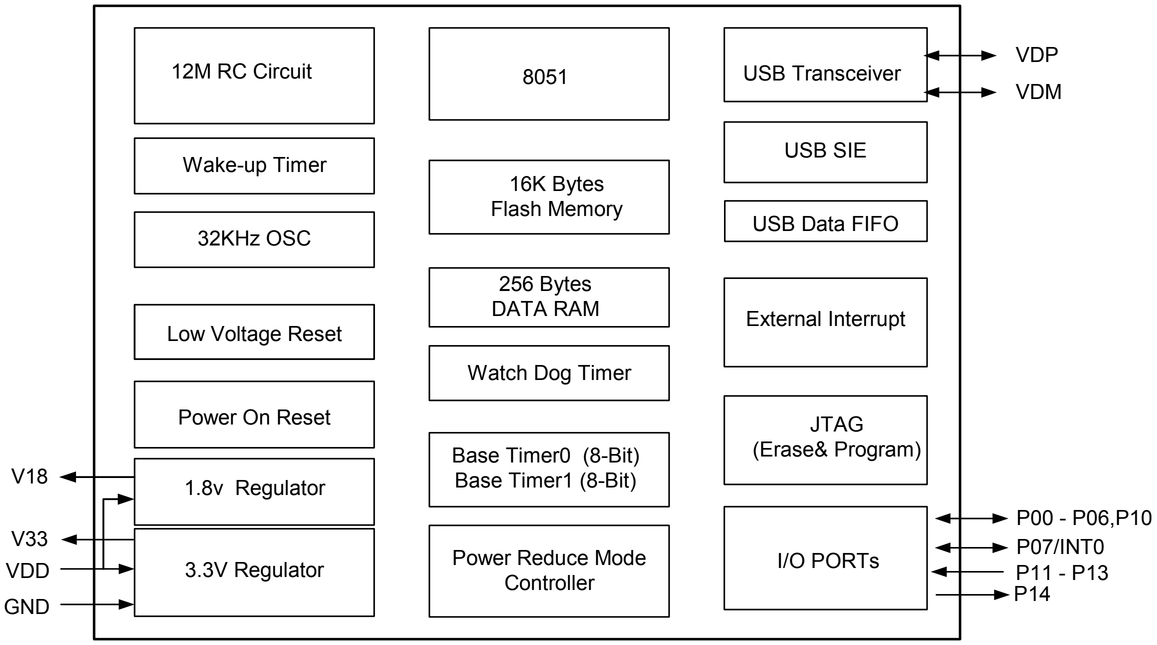


SH68F86 20-Pin (TSSOP) Package

Pad Configuration



Block Diagram



Pin and Pad Description

PIN No.	PAD No.	Designation	I/O	Description
1	5	P06	I/O	Bi-directional I/O pin (3.3V)
2	6	P05	I/O	Bi-directional I/O pin (3.3V)
3	7	P04	I/O	Bi-directional I/O pin (3.3V)
4	8	P03/TCK	I/O	Bi-directional I/O pin (3.3V)
5	11	P14	O	Uni-directional output pin (5.0V)
6	12	P13	I	Uni-directional input pin (5.0V)
7	13	P12	I	Uni-directional input pin (5.0V)
8	14	P11	I	Uni-directional input pin (5.0V)
9	-	NC	-	-
10	18	V18	P	Regulator output (1.8V)
11	19	V_{DD}	P	Power supply (5V)
12	20	V33	P	Regulator output (3.3V)
13	21	VDM	I/O	USB D-
14	22	VDP	I/O	USB D+
15	23, 24	GND	P	Ground
16	1	P10	I/O	Bi-directional I/O pin (3.3V)
17	2	P00/TDO	I/O	Bi-directional I/O pin (3.3V)
18	3	P01/TMS	I/O	Bi-directional I/O pin (3.3V)
19	4	P02/TDI	I/O	Bi-directional I/O pin (3.3V)
20	9	P07/INT0	I/O	Bi-directional I/O pin (3.3V)

Functional Description

1. Memory

1.1. Memory Allocation

There are 16K bytes Program Memory and 256 bytes Data Memory. These features are described as followed.

1.2. Program Memory

The SH68F86 embeds 16K Bytes (0000H - 3FFFH) on-chip program memory for program code. The flash program memory provides electrical erasure and programming.

1.3. Data Memory

The SH68F86 provides additional Bytes of RAM space for increased data parameter handling, high level language usage. The SH68F86 has internal data memory that is mapped into three separate segments.

The Three segments are:

1. The Lower 128 bytes of RAM (addresses 00H to 7FH) are directly and indirectly addressable.
2. The Upper 128 bytes of RAM (addresses 80H to FFH) are indirectly addressable only.
3. The Special Function Registers (SFR, addresses 80H to FFH) are directly addressable only.

The Upper 128 bytes of RAM occupy the same address space as SFR, but they are physically separate from SFR space. When an instruction accesses an internal location above address 7Fh, the CPU can distinguish whether to access the upper 128 bytes data RAM or to access SFR by different addressing mode of the instruction. Note the unused address is unavailable in SFR.

The Internal RAM configuration is shown as below:

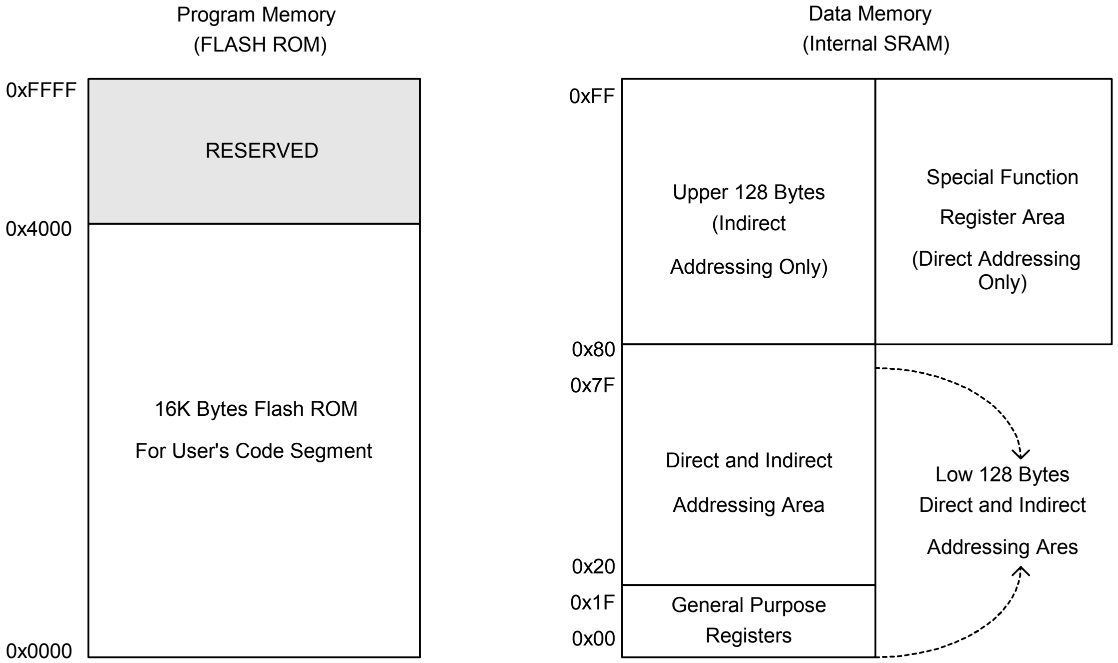


Figure 1-1. SH68F86 Program/Data Memory Map

1.4. Registers

System Registers

Address	Name	Init.	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

Idle and Power-down Control Registers

Address	Name	Init.	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	ENWDT	0	ENLVR1

General I/O Ports Registers

Address	Name	Init.	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0080H	P0	XXXXXXXXB	R/W	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
0090H	P1	0001XXXXB	R/W	0	0	0	P1.4	P1.3	P1.2	P1.1	P1.0
00A3H	P1WK	00000000B	R/W	0	0	0	0	0	0	0	P1WK0
009AH	P0CON	11111111B	R/W	P0CON7	P0CON6	P0CON5	P0CON4	P0CON3	P0CON2	P0CON1	P0CON0
009BH	P1CON	00000001B	R/W	0	0	0	0	0	0	0	P1CON0
0091H	P1PCON	00000000B	R/W	0	0	0	0	0	0	0	P1PCON0

Base Timer Registers

Address	Name	Init.	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

Wake-up Timer & Resume Control Register

Address	Name	Init.	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0095H	WKT_CON	20H	R/W	0	0	PERIOD1	PERIOD0	WKT3	WKT2	WKT1	WKT0

Reset & Resume Flag

Address	Name	Init.	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0096H	MODE_FG	02H	R/W	-	Nonidle	WKUPT	RES_TRG	WDT	USBRST	POF	SUSF

Watch-Dog Timer Control Register

Address	Name	Init.	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0093H	CLRWDT	55H	W	CLRWDT7	CLRWDT6	CLRWDT5	CLRWDT4	CLRWDT3	CLRWDT2	CLRWDT1	CLRWDT0
0094H	PREWDT	00H	R/W	0	0	0	0	0	0	PREWDT1	PREWDT0

Interrupt Control Register

Address	Name	Init.	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00A8H	IE	00H	R/W	EA	0	0	0	ET1	0	ET0	EEXT0
00A9H	IE2	00H	R/W	0	EFUN	ESIE	EOUT0	EIN0	EOT0ERR	EOWSTUP	ESTUP
00B8H	IP	00H	R/W	0	0	0	0	PT1	0	PT0	PEXT0
00B9H	IP2	00H	R/W	0	PFUN	PSIE	POUT0	PIN0	POT0ERR	POWSTUP	PSTUP
00DAH	IF	00H	R/W	0	0	0	0	T1	0	T0	EXT0
00DBH	IF2	00H	R/W	0	FUN	SIE	OUT0	IN0	OT0ERR	OWSTUP	STUP
00DCH	IRQEN	00H	R/W	EIN2	EIN1	ER0STL	ET0STL	ENAK2	ENAK1	ENAKR0	ENAKT0
00DDH	IRQEN2	00H	R/W	0	0	0	0	0	ESUSP	EOVL	0
00DEH	IRQFG	00H	R/W	IN2	IN1	R0STL	T0STL	NAK2	NAK1	NAKR0	NAKT0
00DFH	IRQFG2	00H	R/W	0	0	0	0	0	SUSP	OVL	0
00BDH	EX_IRQCR	00H	R/W	0	0	0	0	0	0	0	EX_IRQM

Registers (continued)

USB Control Register											
Address	**Name**	**Init.**	**R/W**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
00F2H	DADDR	00H	R/W	0	DA6	DA5	DA4	DA3	DA2	DA1	DA0
00F3H	DFC	01H	R/W	PULL_UP	0	FW_K	RSU_SEL	USBEN	0	ERWUP	VPCON
00EAH	TXDAT0	XXH	W	T07	T06	T05	T04	T03	T02	T01	T00
00EBH	TXCNT0	0XH	W	0	0	0	0	TC03	TC02	TC01	TC00
00ECH	TXFLG0	00H	R/W	0	0	0	0	0	0	STLT0	T0FULL
00EDH	RXDAT0	XXH	R	R0.7	R0.6	R0.5	R0.4	R03	R02	R01	R00
00EEH	RXCNT0	0XH	R	0	0	0	0	RC03	RC02	RC01	RC00
00EFH	RXFLG0	00H	R/W	0	0	RXERR	R0_OW	R0SEQ	OUT0ENB	STLR0	R0FULL
00E2H	TXDAT1	XXH	W	T17	T16	T15	T14	T13	T12	T11	T10
00E3H	TXCNT1	0XH	W	0	0	0	0	CNT13	CNT12	CNT11	CNT10
00E4H	TXFLG1	00H	R/W	0	0	0	0	T1EPE	T1SEQC	STL1	T1FULL
00E5H	TXDAT2	XXH	W	T27	T26	T25	T24	T23	T22	T21	T20
00E6H	TXCNT2	0XH	W	0	0	0	0	CNT23	CNT22	CNT21	CNT20
00E7H	TXFLG2	00H	R/W	0	0	0	0	T2EPE	T2SEQC	STL2	T2FULL
00E9H	CRWCON	00H	R/W	0	0	0	0	0	CRSEQ	STLCR	STLCW
Flash Control Register											
Address	**Name**	**Init.**	**R/W**	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
00F7H	XPAGE	00H	R/W	XPAGE7	XPAGE6	XPAGE5	XPAGE4	XPAGE3	XPAGE2	XPAGE1	XPAGE0
00BEH	IB_OFFSET	00H	R/W	IB_OFFSET7	IB_OFFSET6	IB_OFFSET5	IB_OFFSET4	IB_OFFSET3	IB_OFFSET2	IB_OFFSET1	IB_OFFSET0
00BFH	IB_DATA	00H	R/W	IB_DATA7	IB_DATA6	IB_DATA5	IB_DATA4	IB_DATA3	IB_DATA2	IB_DATA1	IB_DATA0
00B3H	IB_CON1	00H	R/W	IB_CON17	IB_CON16	IB_CON15	IB_CON14	IB_CON13	IB_CON12	IB_CON11	IB_CON10
00B4H	IB_CON2	X0H	R/W	-	-	-	-	IB_CON23	IB_CON22	IB_CON21	IB_CON20
00B5H	IB_CON3	X0H	R/W	-	-	-	-	IB_CON33	IB_CON32	IB_CON31	IB_CON30
00B6H	IB_CON4	X0H	R/W	-	-	-	-	IB_CON43	IB_CON42	IB_CON41	IB_CON40
00B7H	IB_CON5	X0H	R/W	-	-	-	-	IB_CON53	IB_CON52	IB_CON51	IB_CON50

2. Interrupt and Reset Vectors

- External Interrupt 0
- Base Timer 0
- Base Timer 1
- SETUP Interrupt
- OWSTUP Interrupt
- OT0ERR Interrupt
- IN0 Interrupt
- OUT0 Interrupt
- SIE Interrupt (NAKT0, NAKR0, T0STL, R0STL, NAK1, NAK2, IN1, IN2)
- Suspend/OVL Interrupt

Address	Interrupt Source	Enable	IRQ Flag	Description
0000H	Reset	-	-	System Reset
0003H	External Interrupt0	IE.0	EXT0	P4.6 Falling Edge
000BH	Base Timer0	IE.1	T0	Base Timer0 Interrupt
0013H	Reserved	-	-	-
001BH	Base Timer1	IE.3	T1	Base Timer1 Interrupt
0023H	Reserved	-	-	-
002BH	Reserved	-	-	-
0033H	Reserved	-	-	-
0043H	Setup Interrupt	IE2.0	STUP	SETUP Token Interrupt
004BH	OWSTUP Interrupt	IE2.1	OWSTUP	-
0053H	OT0ERR Interrupt	IE2.2	OT0ERR	-
005BH	IN0 Interrupt	IE2.3	IN0	IN0 Token Interrupt
0063H	OUT0 Interrupt	IE2.4	OUT0	OUT0 Token Interrupt
006BH	SIE Interrupt	IE2.5	SIE	NAKT0, NAKR0, T0STL, R0STL, NAK1, NAK2, IN1, IN2
0073H	Suspend/OVL Interrupt	IE2.6	FUN	SUSP/OVL Interrupt
007BH	Reserved	-	-	-

3. Micro-Processor

3.1. General Description

The SH68F86 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)

The SH68F86 has a total of 56 SFRs, as shown in the figure below - SFR Map for SH68F86. 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 SH68F86

F8H								-	FFH
F0H	B	-	DADDR	DFC				XPAGE	F7H
E8H		CRWCON	TXDAT0	TXCNT0	TXFLG0	RXDAT0	RXCNT0	RXFLG0	EFH
E0H	ACC		TXDAT1	TXCNT1	TXFLG1	TXDAT2	TXCNT2	TXFLG2	E7H
D8H			IF	IF2	IRQEN	IRQEN2	IRQFG	IRQFG2	DFH
D0H	PSW		BT0	BT1	BTCON				D7H
C8H									CFH
C0H									C7H
B8H	IP	IP2				EX_IRQCR	IB_OFFSET	IB_DATA	BFH
B0H				IB_CON1	IB_CON2	IB_CON3	IB_CON4	IB_CON5	B7H
A8H	IE	IE2						PRCON	AFH
A0H				P1WK					A7H
98H	-	-	P0CON	P1CON					9FH
90H	P1	P1PCON	-	CLRWDT	PREWDT	WKT_CON	MODE_FG		97H
88H	-	-	-	-	-	-	SUSLO	-	8FH
80H	P0	SP	DPL	DPH	-	-	-	PCON	87H

Note 1: *SFR's in marked column are bit addressable.*

Note 2: *SFR's in gray color are standard 8051 SFR's, and others are SFR's for SH68F86.*

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		
Bit 7	CY	Carry flag
Bit 6	AC	Auxiliary Carry flag (for BCD operations)
Bit 5	F0	Flag 0 (Available to the user for general purposes)
Bit 4	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)
Bit 3	RS0	
Bit 2	OV	Overflow Flag
Bit 1	X	User definable flag
Bit 0	P	Parity Flag Set/Cleared by hardware each instruction cycle to indicate an odd/even number of “one” bit I 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 (P0, P1)

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

3.3. Instruction Set List

Arithmetic Instructions					
Opcode			**Bytes**	**Cycles**	**Meaning**
ADD	A, Rn		1	1	Add reg to acc
	A, @Ri		1	2	Add indir byte to acc
	A, direct		2	2	Add dir byte to acc
	A, #data		2	2	Add imm. Data to acc
ADDC	A, Rn		1	1	Add reg to acc with carry flag
	A, @Ri		1	2	Add indir byte to acc with carry flag
	A, direct		2	2	Add dir byte to acc with carry flag
	A, #data		2	2	Add imm. Data to acc with carry flag
SUBB	A, Rn		1	1	Subtract reg from acc with borrow
	A, @Ri		1	2	Subtract indir byte from acc with borrow
	A, direct		2	2	Subtract dir byte from acc with borrow
	A, #data		2	2	Subtract imm. Data from acc with borrow
INC	A		1	1	Increment acc
	Rn		1	2	Increment reg
	@Ri		1	3	Increment indir byte
	DPTR		1	4	Increment data pointer
	direct		2	3	Increment dir byte
DEC	A		1	1	Decrement acc
	Rn		1	2	Decrement reg
	@Ri		1	3	Decrement indir byte
	direct		2	3	Decrement dir byte
MUL	AB		1	11	Multiply A and B, 8-bit
			1	20	Multiply (AUXC A) and B, 16-bit
DIV	AB		1	11	Divide A by B, 8-bit
			1	20	Divide (AUXC A) by B, 16-bit
DA	A		1	1	Decimal adjust acc

Logical Instructions					
Opcode			**Bytes**	**Cycles**	**Meaning**
CLR	A		1	1	Clear acc
CPL	A		1	1	Complement acc
ANL	A, Rn		1	1	AND register to acc
	A, @Ri		1	2	AND indir byte to acc
	A, direct		2	2	AND dir byte to acc
	A, #data		2	2	AND imm. Data to acc
	direct, A		2	3	AND acc to dir byte
	direct, #data		3	3	AND imm. Data to dir byte
ORL	A, Rn		1	1	OR reg to acc
	A, @Ri		1	2	OR indir byte to acc
	A, direct		2	2	OR dir byte to acc
	A, #data		2	2	OR imm. Data to acc
	direct, A		2	3	OR acc to dir byte
	direct, #data		3	3	OR imm. Data to dir byte
XRL	A, Rn		1	1	Exclusive-OR reg to acc
	A, @Ri		1	2	Exclusive-OR indir byte to acc
	A, direct		2	2	Exclusive-OR dir byte to acc
	A, #data		2	2	Exclusive-OR imm. Data to acc
	direct, A		2	3	Exclusive-OR acc to dir byte
	direct, #data		3	3	Exclusive-OR imm. Data to dir byte

Instruction Set List (continued)

RL	A		1	1	Rotate acc left
RLC	A		1	1	Rotate acc left through the carry
RR	A		1	1	Rotate acc right
RRC	A		1	1	Rotate acc right throught the carry
SWAP	A		1	4	Swap nibbles within the acc
					Data Transfer
Opcode			**Bytes**	**Cycles**	**Meaning**
MOV	A, Rn		1	1	Move reg to acc
	A, @Ri		1	2	Move indir byte to acc
	Rn, A		1	2	Move acc to reg
	@Ri, A		1	2	Move acc to indir byte
	A, direct		2	2	Move dir byte to acc
	A, #data		2	2	Move imm. Data to acc
	Rn, #data		2	2	Move imm. Data to reg
	direct, A		2	2	Move acc to dir byte
	direct, Rn		2	2	Move reg to dir byte
	@Ri, #data		2	2	Move imm. Data to indir byte
	Rn, direct		2	3	Move dir byte to reg
	direct, @Ri		2	3	Move indir byte to dir byte
	@Ri, direct		2	3	Move dir byte to indir byte
	direct, direct		3	3	Move dir byte to dir byte
	direct, #data		3	3	Move imm. Data to dir byte
	DPTR, #data16		3	3	Load data pointer with 16-bit constant
MOVC	A, @A+DPTR		1	7	Move code byte relative to DPTR to acc
	A, @A+PC		1	8	Move code byte relative to PC to acc
MOVX	@Ri, A		1	4	Move acc to xdata byte (8 bit address)
	A, @Ri		1	5	Move xdata byte to acc (8 bit address)
	@DPTR, A		1	5	Move acc to xdata byte (16 bit address)
	A, @DPTR		1	6	Move xdata byte to acc (16 bit address)
PUSH	direct		2	5	Push dir byte to stack
POP	direct		2	4	Pop dir byte from stack
XCH	A, Rn		1	3	Exchange reg with acc
	A, @Ri		1	4	Exchange indir byte with acc
	A, direct		2	4	Exchange dir byte with acc
XCHD	A, @Ri		1	4	Exchange low-order digit in indir byte with acc
					Bit Manipulation
Opcode			**Bytes**	**Cycles**	**Meaning**
CLR	C		1	1	Clear carry
	bit		2	3	Clear dir bit
SETB	C		1	1	Set carry
	bit		2	3	Set dir bit
CPL	C		1	1	Complement carry
	bit		2	3	Complement dir bit
ANL	C, bit		2	2	AND dir bit to carry
	C, /bit		2	2	AND complement of dir bit to carry
ORL	C, bit		2	2	OR dir bit to carry
	C, /bit		2	2	OR complement of dir bit to carry
MOV	C, bit		2	2	Move dir bit to carry
	bit, C		2	3	Move carry to dir bit

Instruction Set List (continued)

Program Branching					
Opcode			Bytes	Cycles	Meaning
JC	rel	(not taken)	2	2	Jump if carry is set Jump if less than
		(taken)		4	
JNC	rel	(not taken)	2	2	Jump if carry is not set Jump if greater than or equal
		(taken)		4	
JB	bit, rel	(not taken)	3	4	Jump if dir bit is set
		(taken)		6	
JNB	bit, rel	(not taken)	3	4	Jump if dir bit is not set
		(taken)		6	
JBC	bit, rel	(not taken)	3	4	Jump if dir bit is set and clear bit
		(taken)		6	
JZ	rel	(not taken)	2	3	Jump if acc is zero
		(taken)		5	
JNZ	rel	(not taken)	2	3	Jump if acc is not zero
		(taken)		5	
SJMP	rel		2	4	Short jump (relative address)
ACALL	addr11		2	7	Absolute subroutine call
LCALL	addr16		3	7	Long subroutine call
RET			1	8	Return from subroutine
RETI			1	8	Return from interrupt
AJMP	addr11		2	4	Absolute jump
LJMP	addr16		3	5	Long jump
JMP	@A+DPTR		1	6	Jump indir relative to DPTR
CJNE	A, direct, rel	(not taken)	3	4	Compare dir byte to acc. And jump if not equal
		(taken)		6	
CJNE	A, #data, rel	(not taken)	3	4	Compare imm. Data to acc. And jump if not equal
		(taken)		6	
CJNE	Rn, #data, rel	(not taken)	3	4	Compare imm. Data to reg and jump if not equal
		(taken)		6	
CJNE	@Ri, #data, rel	(not taken)	3	4	Compare imm. Data to indir and jump if not equal
		(taken)		6	
DJNZ	Rn, rel	(not taken)	2	3	Decrement reg and jump if not zero
		(taken)		5	
DJNZ	direct, rel	(not taken)	3	4	Decrement dir byte and jump if not zero
		(taken)		6	
NOP			1	1	No operation

4. Oscillators

The SH68F86 has a built-in 12MHz RC resonator for system clock. The oscillator generates the system timing and control signal to be supplied to the CPU core and the on-chip peripherals, such as USB, Timer and so on.

Besides, the SH68F86 also has a built-in 32 KHz RC resonator to generate the clock for wake up timer.

5. Reset and Power-reducing Mode

There are totally four Reset Sources in the SH68F86 application.

- Hardware reset: Low-Voltage Reset, Power-On Reset or External Reset
- WDT (Watch-dog Timer) Reset
- Resume Reset
- USB Reset

5.1. Hardware Reset

5.1.1. Power-On Reset (POR) and LVRA

When power is first applied to the SH68F86, the internal Power-On Reset will be generated and reset the whole chip.

This process is fulfilled by a power-on reset circuit and an auxiliary Lower-voltage reset circuit (LVRA) monitoring V_{DD}. Once V_{DD} climb up from 0V and cross the V_{POR}, the internal POR signal will active and end after $T_{RST(POR)}$.

The LVRA will perform as a function Low-voltage Reset when system is normal running (under normal/idle/power-down mode). LVRA reset signal (this signal is shared with POR signal) will active when V_{DD} was less than V_{LVRA} and lasts for $T_{PW(LVRA)}$, LVRA signal will end after $T_{\mathbf{RST(LVR)}}$ when V_{DD} was larger than V_{LVRA}.

See Figure5-1 for the POR and LVRA behavior.

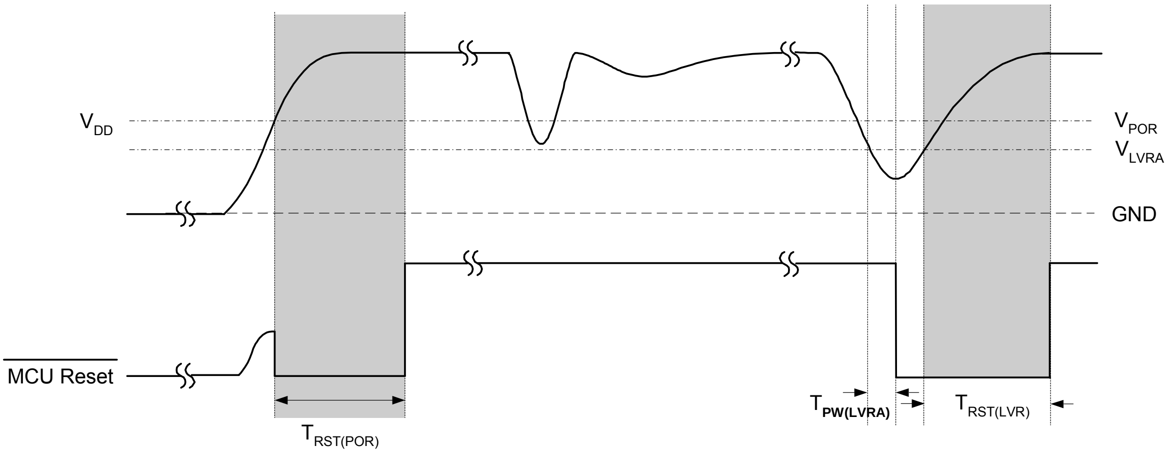


Figure 5-1. Power-on Reset and LVRA

Note:

$V_{POR(max.)}$ = 3.6V

$V_{LVRA(min.)}$ = 2.9V, $V_{LVRA(typ.)}$ = 3.0V, and $V_{LVRA(max.)}$ = 3.1V

$T_{PW(LVRA)}$ (Drop-Down Pulse Width for LVRA) = 2^{10} X T_{SYS}

$T_{RST(POR)}$ (Internal Power-on Reset Hold Time) = 2^{17} X T_{SYS}

$T_{RST(LVR)}$ (Internal Low-voltage Reset Hold Time) = 2^{17} X T_{SYS}

5.1.2. Low Voltage Reset (LVR)

(1) Low Voltage Reset 1 (LVR1)

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

The LVR1 circuit will monitor the 1.8V regulator output voltage to the MCU core.

LVR1 reset signal will active when the input power of MCU core was less than V_{LVR1} and lasts for $T_{PW(LVR1)}$, LVR1 signal will end after $T_{RST(LVR)}$ when the power was larger than V_{LVR1}. See Figure 5-2 for the LVR1 behavior.

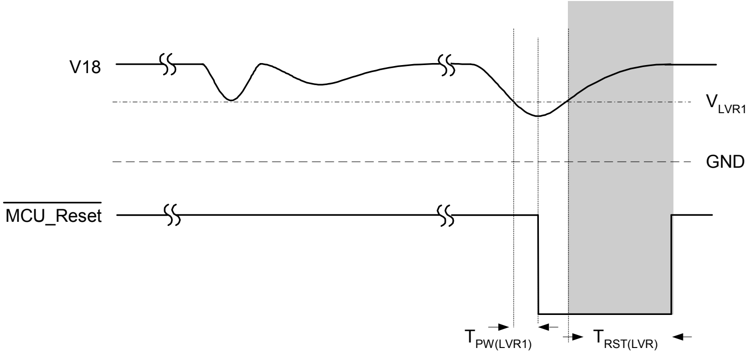


Figure 5-2. Low Voltage Reset 1

Note: *$V_{LVR1(min.)}$ = 1.4V, $V_{LVR1(typ.)}$ = 1.5V, and $V_{LVR1(max.)}$ = 1.6V*
$T_{PW(LVR1)}$ (Drop-Down Pulse Width for LVR1) = 2^{10} X T_{SYS}.
$T_{RST(LVR)}$ (Internal Low-voltage Reset Hold Time) = 2^{17} X T_{SYS}.

Under Power-down mode:

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

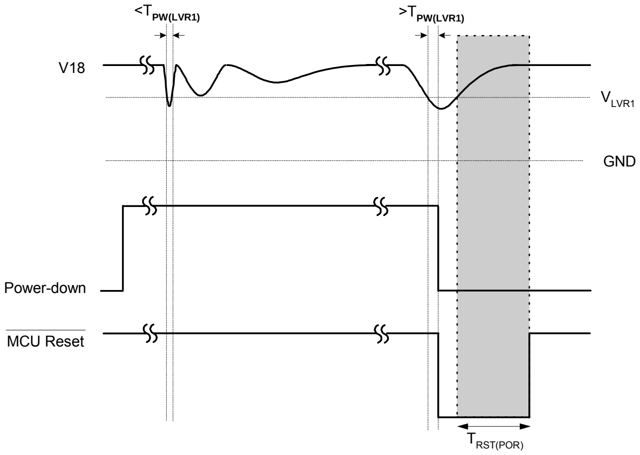


Figure 5-3. Low Voltage Reset 1 under Power-down Mode

(2) Low Voltage Reset (LVR2)

The embedded Low-Voltage Reset (LVR2) circuit monitors the 3.3V regulator output Voltage. It will generate a internal reset to the whole chip while heavy loads at 3.3V regulator output switched on which cause the regulator output voltage temporarily fall below the minimum specified operating voltage. This feature is can protect system from working under bad power supply environment.

LVR2 reset signal will active when the 3.3V regulator output was less than V_{LVR2} and lasts for $T_{PW(LVR2)}$, LVR2 signal will end after $T_{RST(LVR)}$ when the power was larger than V_{LVR2}. See Figure 5-4 for the LVR2 behavior.

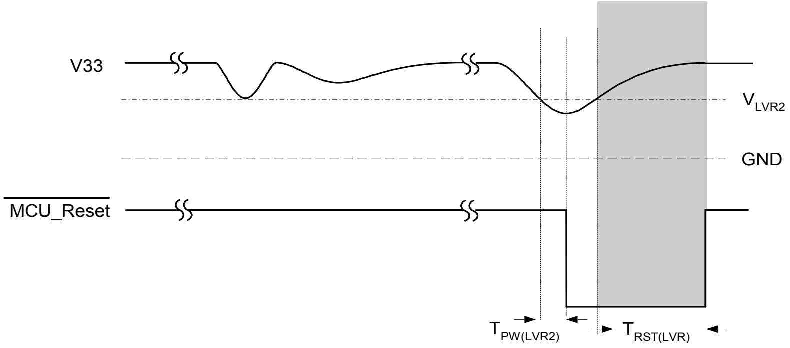


Figure 5-4. Low Voltage Reset 2

Note: *V_{LVR2} ($V_{LVR2(min.)}$ = 2.2V, $V_{LVR2(typ.)}$ = 2.4V, and $V_{LVR2\ (max.)}$ = 2.6V)*
$T_{PW(LVR2)}$ (Drop-Down Pulse Width for LVR2) = 2^{10} X T_{SYS}
$T_{RST(LVR)}$ (Internal Low-voltage Reset Hold Time) = 2^{17} X T_{SYS}

5.2. Watch-dog Timer Reset

The SH68F86 implements a Watchdog timer to avoid system stop or malfunction. The clock source of the WDT is FSYS. The time-out interval of Watchdog timer is selected by **PREWDT[1:0]**. The Watchdog timer must be cleared within time-out period; otherwise the Watchdog timer will overflow and cause a system reset. The Watchdog timer is cleared and enabled after the system is reset, and can be disabled by the software only on idle mode. Users can clear the Watchdog timer by writing a #55H to the **CLRWDT** (0093H) register.

0093H	CLRWDT	Initial Value	Clear Watch-dog Timer Control Register	
Bit[7:0]	CLRWDT [7:0]	55H	W	Write "55H" to clear watch-dog timer **Reset source: Hardware reset, USB reset, WDT reset, Resume reset**

0094H	PREWDT	Initial Value	Watch-dog Timer Pre-scalar Control Register	
Bit[7:2]	-	000000b	-	Reserved
Bit[1:0]	PREWDT [1:0]	00b	R/W	**Watch-dog timer Pre-scalar control register** 00: 2^{17} T_{SYS} (10.922ms) 01: 2^{18} T_{SYS} (21.845ms) 10: 2^{19} T_{SYS} (43.688ms) 11: 2^{20} T_{SYS} (87.376ms) **Reset source: Hardware reset, USB reset, WDT reset, Resume reset**

Note1: *The new Pre-scalar value will be loaded after the Watchdog Timer was cleared (write #55H to* ***CLRWDT*** *register)*

Note2: *When system enters Power-Down Mode, WDT will stop due to the lack of T_{SYS}. When system resumes from Power-Down Mode, the WDT control register will be cleared to the initial state*

5.3. Resume Reset

A resume reset holds SFR values, CPU status and Pin state, but program is re-run at 0000h. Port 1.0 can be set as a resume port by setting P1WK, The low level of enabled resume source is triggered in power-down mode will causes a resume reset.

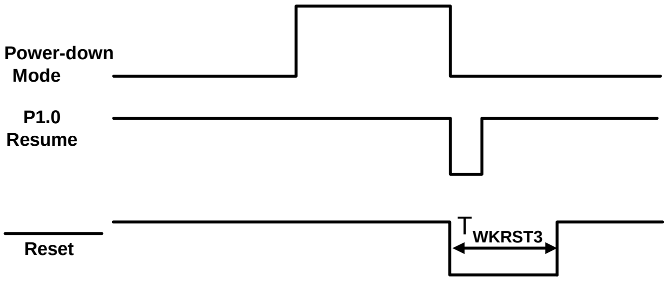


Figure 5-5. In power-down enabled resume ports are triggered at the low level

5.4. USB Reset

When USB Reset signal detected by the SH68F86, A USB reset will generate. It will initialize some SFR values and the program is re-run at 0000h.

5.5. IDLE and Power-Down Mode

The SH68F86 has two power-reducing modes:

- IDLE mode (**IDL** = 1 & **SUSLO** = 55H): The CPU is frozen, the 12MHz RC resonator continues to run and the Interrupt and Timer blocks continue to be clocked.
- Power-down mode (**PD** = 1 & **SUSLO** = 55H): The 12MHz RC resonator is frozen.

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
				Reset source: Hardware reset, USB reset, WDT reset, Resume reset

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

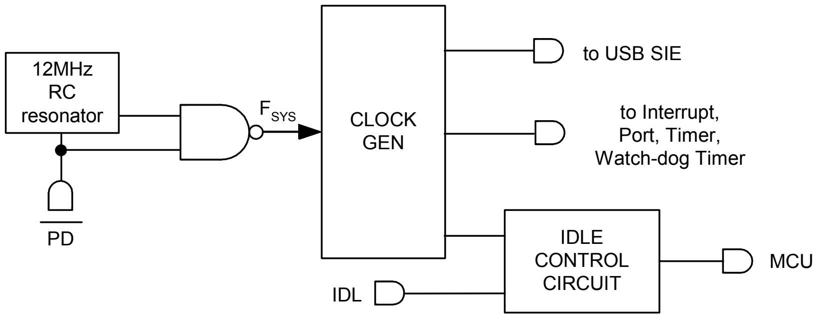


Figure 5-6. Sketch map for IDLE and Power-Down Mode implement

5.5.1. IDLE Mode

Two continuous instructions that set PCON.0 to ‘1’ and set SUSLO to ‘55H’ let the SH68F86 enter IDLE mode. In IDLE mode, the internal clock signal is gated off to the CPU only. The CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, and all other registers maintain their value during IDLE mode. The port pins hold the latest logical states before system enter IDLE mode.

There are four ways to terminate IDLE mode:

(1) **Activation of any enabled interrupt will terminate the 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 three NOPs after the instruction that put the device into IDLE mode. (Watchdog timer begins to count from the value where it was stopped.)

(2) **The wake-up Timer is Time-out in IDLE Mode**

When the wake-up timer is time-out in IDLE mode, 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 will execute properly, user should add three NOPs after the instruction that put the device into IDLE mode.

(3) **Hardware Reset, Watchdog Reset, USB Reset**

At this time, the CPU resumes program execution from the beginning of the whole program, which is 0000H.

(4) **P1.0 as Resume signal will terminate the IDLE mode**

When a low level of enabled resume source is triggered in IDLE mode, the next instruction to be executed will be the one following the instruction that put the device into IDLE mode. Note that, H/W will not issue K-State when I/O ports resume source is triggered in the IDLE mode. In order to make sure program will execute properly, user should add three NOPs after the instruction that put the device into IDLE mode.

00A3H	**P1WK**	**Initial Value**	**Port1 Resume Enable Register**	
Bit[7:1]	P1WK[7:1]	0000000b	-	Reserved
Bit [0]	P1WK[0]	0b	**R/W**	1: Enable wake-up function of PORT1’s pins 0: Disable wake-up function of PORT1’s pins **Reset source: Hardware reset**

Example: Set port 1.0 as the resume resource

```
IDLE1:  MOV     P1,        #01H    ; Set PORT1.
        MOV     P1WK,      #01H    ; Set PORT1 resume ability.
        ANL     PRCON,     #FBH    ; Disable Watch-Dog timer under idle mode.
        MOV     CLRWDT,    #55H    ; Clear Watch-Dog Timer
        ORL     PCON,      #01H    ; Set IDLE mode.
        MOV     SUSLO,     #55H    ; Enter IDLE mode.
        NOP                        ; 3 NOP instruction (make sure program will executes properly)
        NOP
        NOP
```

5.5.2. Power-down Mode

Method of entering Power-down mode: set PCON.1 = 1 and set SUSLO = 55h

- In the Power- down mode, the on-chip oscillator stops.
- With the clock frozen, all functions are stopped, but the on-chip RAM and Special function Registers are held.
- In order to make sure the program will resume properly, user should add three NOPs immediately after setting SUSLO to 55H.

There are two ways to exit from Power-down mode.

- Low Voltage Reset or Power-On Reset.
- Resume reset: A resume reset holds SFR values, CPU status and Pin state, but program is re-run at 0000h. **There are three ways to generate resume reset.**

(1) P1.0 can be set as a resume ports by setting **P1WK**. A low level of enabled resume source is triggered in Power-down mode will cause a Resume Reset.

(2) Wake-up Timer time out

(3) USB Bus Non-idle State (VDM is low, or VDM & VDP both high)

Port resume reset example 1: Assume that P1.0 is resume source and H/W issues K-State when Resume Reset occurs.

```
PWRDN_HW:
        MOV     P1,         #01H    ; Set PORT1.
        MOV     P1WK,       #01H    ; Set PORT1 resume ability.
        ANL     PRCON,      #FBH    ; Disable Watch-Dog timer under idle mode.
        MOV     CLRWDT,     #55H    ; Clear Watch-Dog Timer
        ANL     DFC,        #EFH    ; RSU_SEL=0, H/W issue K-State to respond RESUME signal
        ORL     DFC,        #02H    ; ERWUP=1, Enable Remote Wake Up function
        ORL     PCON,       #02H    ; Set IDLE mode.
        MOV     SUSLO,      #55H    ; Enter IDLE mode.
        NOP                         ; 3 NOP instruction (make sure program will executes properly)
        NOP
        NOP
```

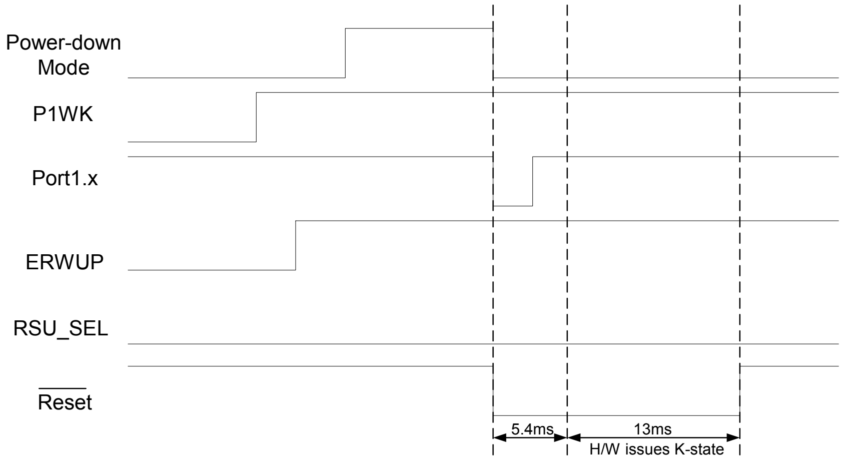


Figure 5-7. Select H/W Issues K-State by Resume Ports Reset

Port resume reset example 2: Assume that PORT1.0 is resume source and F/W issues K-State when Resume Reset occurs.

```
PWRDN_FW:
          MOV      P1,        #01H    ; Set PORT1.
          MOV      P1WK,      #01H    ; Set PORT1 resume ability.
          ANL      PRCON,     #FBH    ; Disable Watch-Dog timer under idle mode.
          MOV      CLRWDT,    #55H    ; Clear Watch-Dog Timer
          ORL      DFC,       #10H    ; RSU_SEL=1, F/W issue K-State to respond RESUME signal
          ORL      DFC,       #02H    ; ERWUP=1, Enable Remote Wake Up function
          ORL      PCON,      #02H    ; Set IDLE mode.
          MOV      SUSLO,     #55H    ; Enter IDLE mode.
          NOP                         ; 3 NOP instruction (make sure program will executes properly)
          NOP
          NOP
```

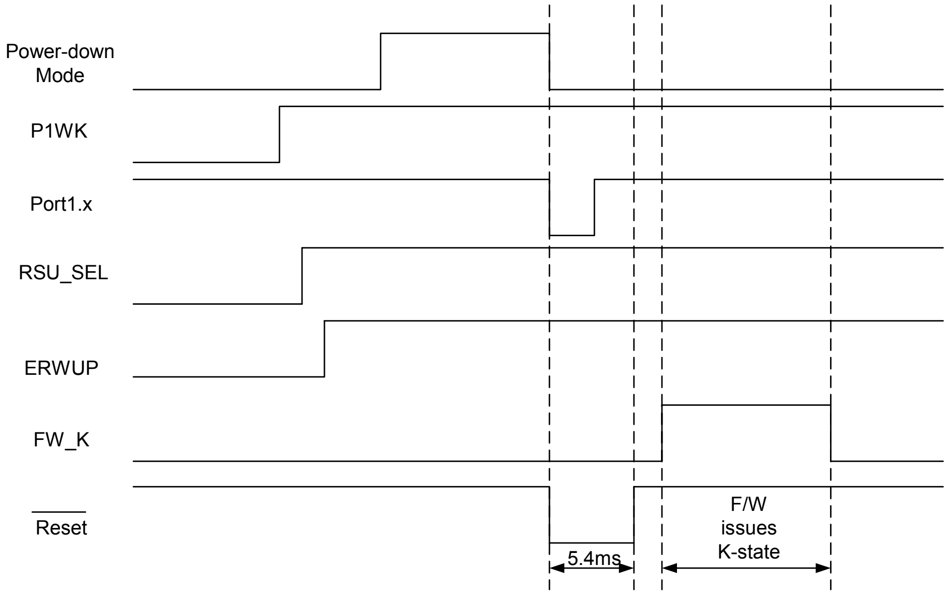


Figure 5-8. Select F/W Issues K-State by Resume Ports Reset

Wake-up Timer Time out Resume Reset

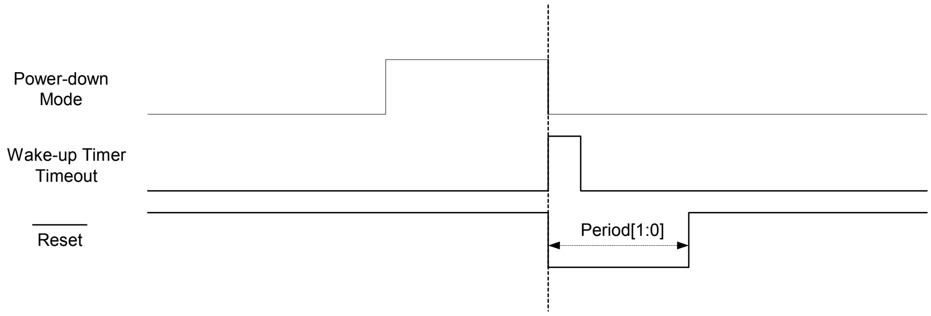


Figure 5-9. Wake-up Timer Time Out Waveform

USB Bus Non-idle State Resume Reset

- ■ Resume reset after Non-idle event

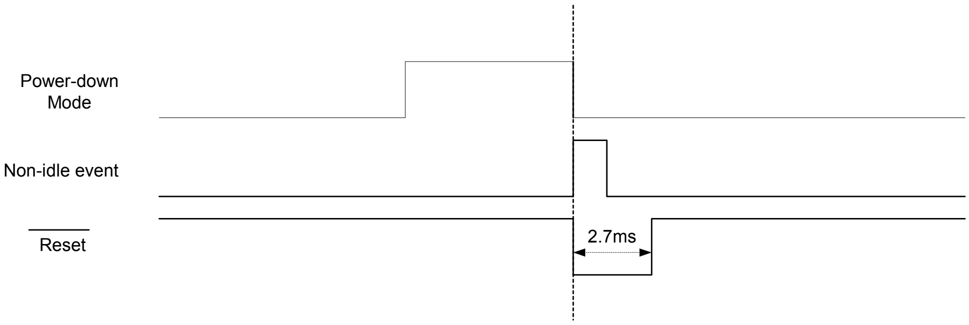


Figure 5-10. USB Non-idle Resume Reset Waveform

- ■ USB reset signal at Power-down mode

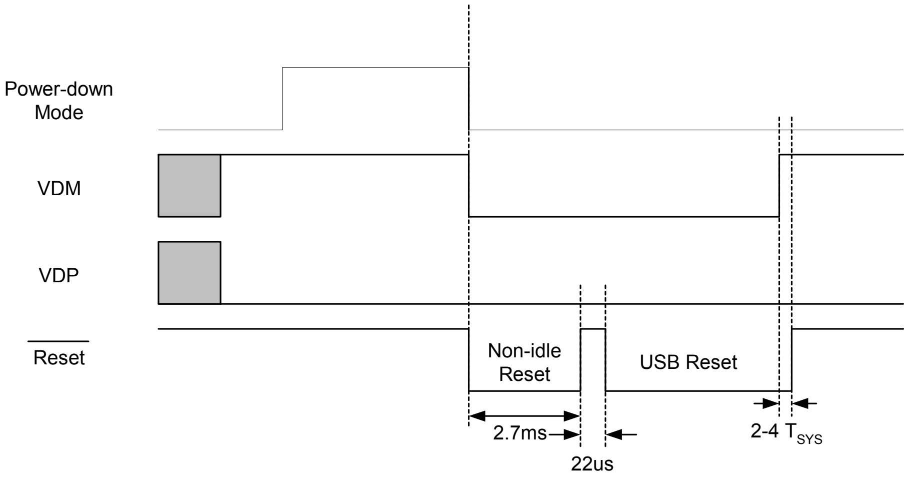


Figure 5-11. USB Reset Wake-up Waveform

5.6. Wake-up Timer

- The SH68F86 has a Built-in 32KHz Ring-Oscillator. It is the clock source of wake-up timer. The 32KHz Ring-Oscillator will start when the control register **WKT[3:2]** was not equal to #00b.
- The wake-up timer can only be enabled/disabled by **WKT[3:0]** (**WKT[3:0]** not equal to 00xxb).
- If the Wake-up timer is enabled and system enter idle/power-down mode, the wake-up timer will load the time-out period register **WKT[3:0]** and start to count.

0095H	WKT_CON	Initial Value		Wake-up Timer & Resume Reset Control Register
Bit[7:6]	-	00b	-	Reserved
Bit[5:4]	Period [1:0]	10b	R/W	**Internal Resume Reset period for Power-Down mode** (these times do not include resonator start-up time) 00: 2^{11} T_{SYS} (170us) 01: 2^{12} T_{SYS} (340us) 10: 2^{17} T_{SYS} (10.922ms) 11: 2^{18} T_{SYS} (21.845ms) **Reset source: Hardware reset**
Bit[3:0]	WKT[3:0]	0000b	R/W	**Wake-up timer** 00xx: disable Wake-up timer under **Power-down mode** or **IDLE mode** Others: enable Wake-up timer under **Power-down mode** or **IDLE mode** 0101: 2^{0} T_{RING} (31.25us@32KHz) 0110: 2^{7} T_{RING} (4ms@32KHz) 0111: 2^{8} T_{RING} (8ms@32KHz) 0100: 2^{9} T_{RING} (16ms@32KHz) 1001: 2^{10} T_{RING} (32ms@32KHz) 1010: 2^{11} T_{RING} (64ms@32KHz) 1011: 2^{12} T_{RING} (128ms@32KHz) 1000: 2^{13} T_{RING} (256ms@32KHz) 1101: 2^{14} T_{RING} (512ms@32KHz) 1110: 2^{15} T_{RING} (1.024s@32KHz) 1111: 2^{16} T_{RING} (2.048s@32KHz) 1100: 2^{17} T_{RING} (4.096s@32KHz) **Reset source: Hardware reset**

5.7. MODE_FG Flag

0096H	MODE_FG	Initial Value		Mode Register
Bit7	-	0b	-	Reserved
Bit6	Nonidle	0b	R/W	USB bus flag. Write “0” to clear, write “1” no effect. 1: set by non-idle event **Reset source: Hardware reset, USB reset**
Bit5	WKUPT	0b	R/W	Set “1” after wake-up timer time-out. Write “0” to clear, write “1” no effect. **Reset source: Hardware reset or USB reset**
Bit4	RES_TRG	0b	R/W	“1”: Remote wake up; “0”: Global wake up. Write “0” to clear, write “1” no effect. **Reset source: Hardware reset or USB reset**
Bit3	WDT	0b	R/W	Set “1” after Watchdog reset. Write “0” to clear, write “1” no effect. **Reset source: Hardware reset or USB reset**
Bit2	USBRST	0b	R/W	Set “1” after USB reset. Write “0” to clear, write “1” no effect. **Reset source: Hardware reset**
Bit1	POF	1b	R/W	Set “1” after power-on reset, Low voltage reset and External reset. Write “0” to clear, write “1” no effect. **Reset source: Hardware reset**
Bit0	SUSF	0b	R/W	Set “1” when entering Power-down mode. Write “0” to clear, write “1” no effect. **Reset source: Hardware reset or USB reset**

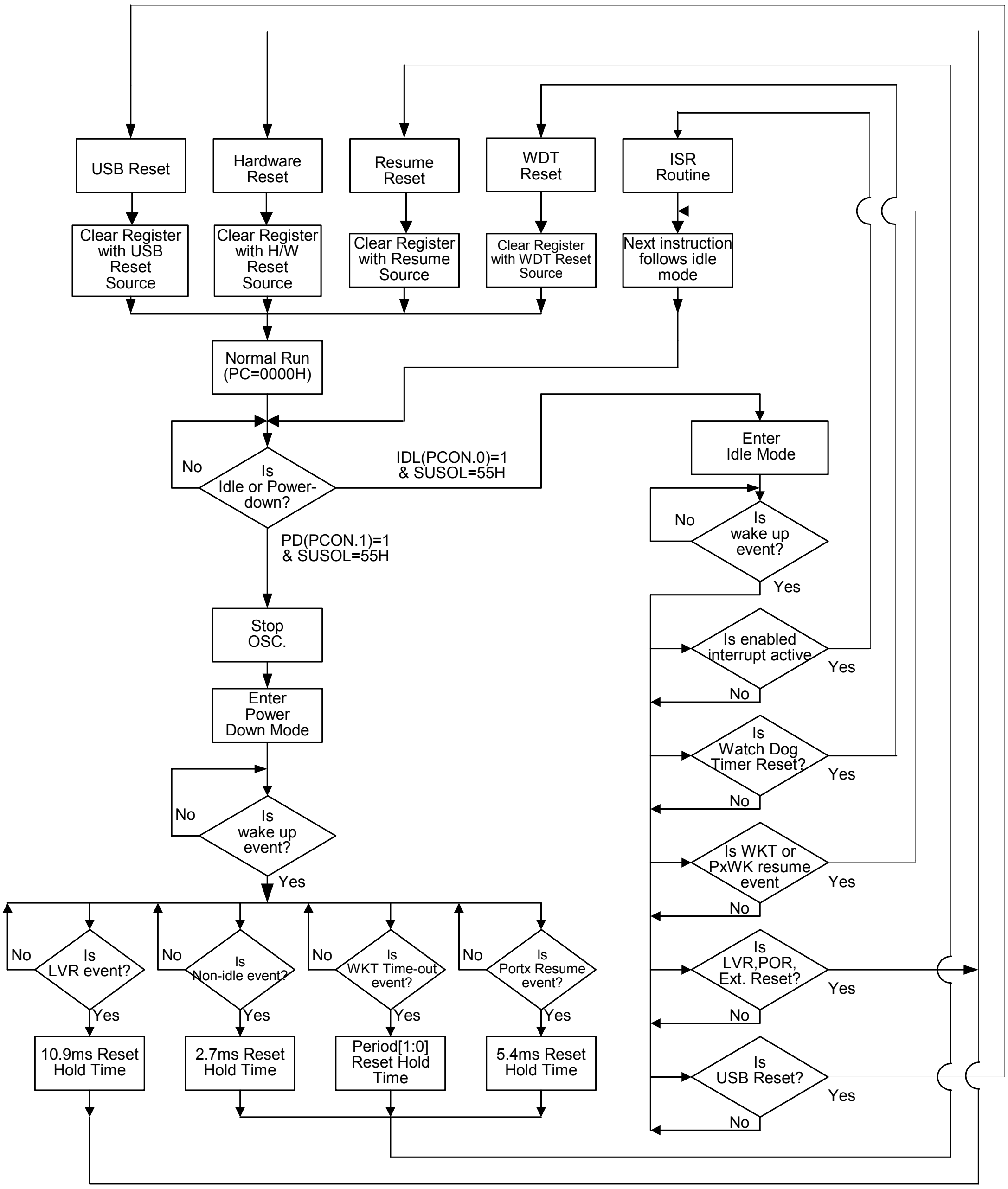


Figure 5-12. Event for exit from idle and power down mode

6. Input/Output Ports

6.1. 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	I/O	Shown in Figure 6-1	0	0	Output Low (0.4 X V33, 2mA)
				1	0	Output High (0.6 X V33, -2mA)
		I		X	1	Input (HI-Z)

6.2. Port-1 Configuration

Port-1 Configuration 1: (Reset source: Hardware reset)

I/O Port	Function	I/O	Circuit Structure	Control Bits			Description
				P1.x	P1CON.x	P1PCON.x	
Port1[0]	Port1	O	Shown in Figure 6-2	0	0	x	Output Low (0.4 X V33, 2mA)
				1	0	x	Output High (0.6 X V33, -2mA)
		I		X	1	0	Input (HI-Z)
						1	Input with Pull-up Resister

Port -1 Configuration 2: (Reset source: Hardware reset)

I/O Port	Function	I/O	Circuit Structure	Control Bits	Description
				P1.x	
Port1[3:1]	Port1	I	Shown in Figure 6-1	X	Input (HI-Z)

Port -1 Configuration 3: (Reset source: Hardware reset)

I/O Port	Function	I/O	Circuit Structure	Control Bits	Description
				P1.x	
Port1[4]	Port1	O	Shown in Figure 6-1	0	Output Low (2.6V - 3.2V, typ: 9mA)
				1	Output High (2.4V, -50μA ~ -100uA)

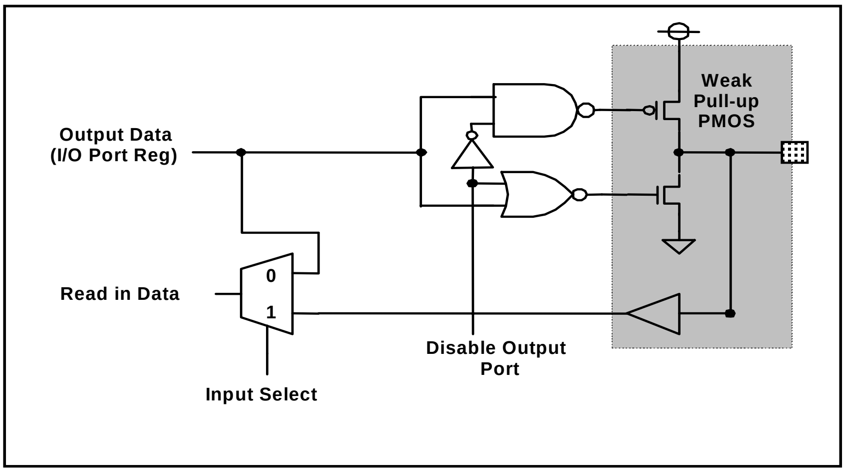


Figure 6-1. PORT Configuration-1

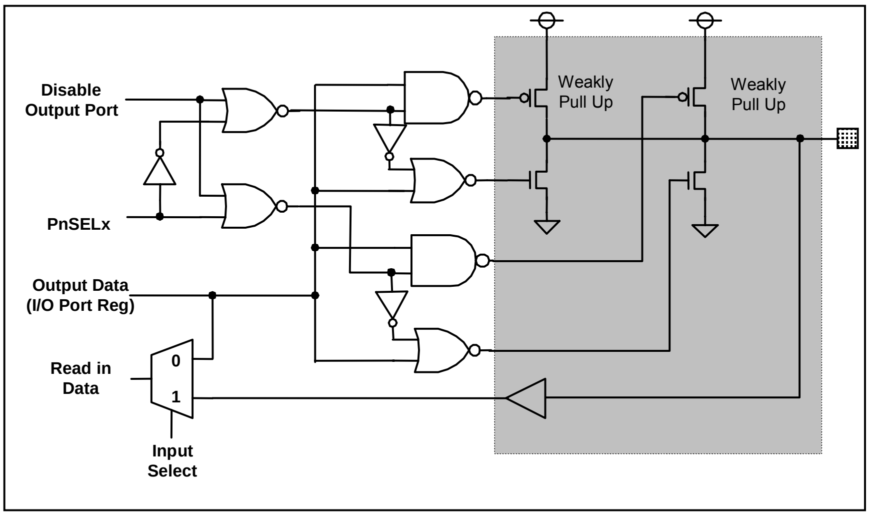


Figure 6-2. PORT Configuration-2

7. Interrupts

7.1. External Interrupt Registers

00BDH	EX_IRQCR	Initial Value	External INT0 Trigger Mode Control registers
Bit[7:1]	-	0000000b	Reserved
Bit[0]	EX_IRQM	0b	0: Falling edge 1: Rising edge **Reset Source: Hardware reset**

7.2. Interrupt Enables

Each of the interrupt sources can be individually enabled or disabled by setting or clearing a bit in the SFR named **IE**, **IE2**, **IRQEN**, and **IRQEN2**. The register **IE** also contains a global disable bit, which can be cleared to disable all interrupts at once. Figure 7-1 shows the interrupt register for the SH68F86.

Interrupt Enable Register1

00A8H	IE	Initial Value	Interrupt Enable Register	
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. **Reset Source: Hardware reset, USB reset or WDT reset**
Bit6	-	0b	-	Reserved
Bit5	-	0b	-	Reserved
Bit4	-	-	-	Reserved
Bit3	ET1	0b	R/W	Base Timer1 interrupt
Bit2	-	0b	-	Reserved
Bit1	ET0	0b	R/W	Base Timer0 interrupt
Bit0	EEXT0	0b	R/W	External interrupt0

Enable bit = 1, enables the interrupt
Enable bit = 0, disables the interrupt
Reset source: Hardware reset or USB reset
Note: *EA bit will also be clear by WDT reset*

Interrupt Enable Register 2

00A9H	IE2	Initial Value	Interrupt Enable Register	
Bit7	-	0b	-	Reserved
Bit6	EFUN	0b	R/W	SUSP/OVL interrupt
Bit5	ESIE	0b	R/W	SIE interrupt (NAKT0, NAKR0, NAK1, NAK2, T0STL, R0STL)
Bit4	EOUT0	0b	R/W	Out0 interrupt
Bit3	EIN0	0b	R/W	IN0 interrupt
Bit2	EOT0ERR	0b	R/W	OT0ERR interrupt
Bit1	EOWSTUP	0b	R/W	OWSTUP interrupt
Bit0	ESTUP	0b	R/W	Setup interrupt

Enable bit = 1, enables the interrupt
Enable bit = 0, disables the interrupt
Reset source: Hardware reset or USB reset

Interrupt Enable Register 3

00DCH	IRQEN	Initial Value	SIE Interrupt Enable Register	
Bit7	EIN2	0b	R/W	IN2 interrupt
Bit6	EIN1	0b	R/W	IN1 interrupt
Bit5	ER0STL	0b	R/W	R0 stall interrupt
Bit4	ET0STL	0b	R/W	T0 stall interrupt
Bit3	ENAK2	0b	R/W	T2 NAK interrupt
Bit2	ENAK1	0b	R/W	T1 NAK interrupt
Bit1	ENAKR0	0b	R/W	R0 NAK interrupt
Bit0	ENAKT0	0b	R/W	T0 NAK interrupt
Enable bit = 1, enables the interrupt **Enable bit = 0, disables the interrupt** **Reset Source: Hardware reset or USB reset**				

Interrupt Enable Register 4

00DDH	IRQEN2	Initial Value	FUN Interrupt Enable Register	
Bit7	-	0b	-	Reserved
Bit6	-	0b	-	Reserved
Bit5	-	0b	-	Reserved
Bit4	-	0b	-	Reserved
Bit3	-	0b	-	Reserved
Bit2	ESUSP	0b	R/W	Suspend interrupt (bus idle > 5ms)
Bit1	EOVL	0b	R/W	OVL interrupt
Bit0	-	0b	-	Reserved
Enable bit = 1, enables the interrupt **Enable bit = 0, disables the interrupt** **Reset Source: Hardware reset or USB reset**				

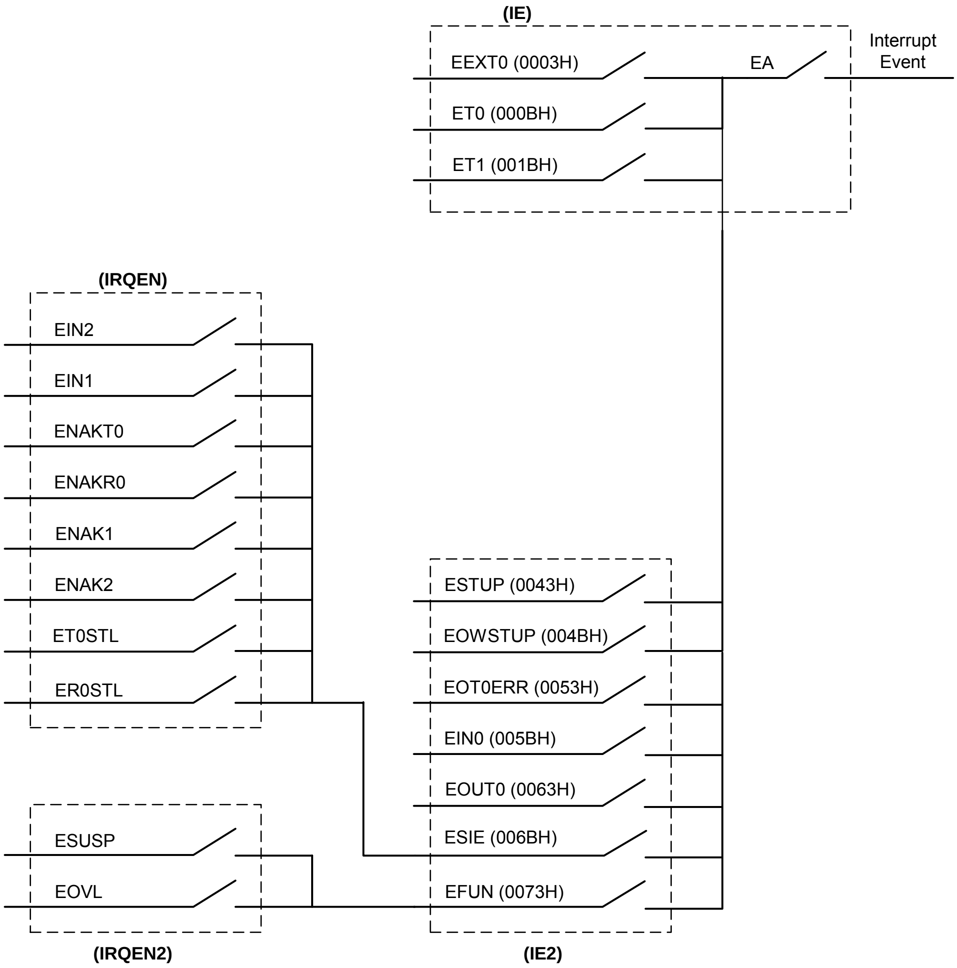


Figure 7-1. Interrupt Structure

7.2. Interrupt Priorities

- Each interrupt source can also be individually programmed to one of the two priority levels by setting or clearing a bit in the SFR named **IP** (Interrupt Priority) and **IP2**. The Following figure shows the **IP** & **IP2** register in the SH68F86.
- 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 the 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.

00B8H	IP	Initial Value	Interrupt Priority Register	
Bit7	-	0b	-	Reserved
Bit6	-	0b	-	Reserved
Bit5	-	0b	-	Reserved
Bit4	-	0b	-	Reserved
Bit3	PT1	0b	R/W	Base Timer1 interrupt priority bit
Bit2	-	0b	-	Reserved
Bit1	PT0	0b	R/W	Base Timer0 interrupt priority bit
Bit0	PEXT0	0b	R/W	External interrupt0 priority bit
1: high priority, 0: low priority **Reset Source: Hardware reset or USB reset**				

00B9H	IP2	Initial Value	Interrupt Priority Register	
Bit7	-	0b	-	Not implemented (always 0)
Bit6	PFUN	0b	R/W	SUSP/OVL interrupt priority bit
Bit5	PSIE	0b	R/W	SIE interrupt priority bit (NAKT0, NAKR0, NAK1, NAK2, T0_STL, R0_STL, IN1, IN2)
Bit4	POUT0	0b	R/W	Out0 interrupt priority bit
Bit3	PIN0	0b	R/W	IN0 interrupt priority bit
Bit2	POT0ERR	0b	R/W	OT0ERR interrupt priority bit
Bit1	POWSTUP	0b	R/W	OWSTUP interrupt priority bit
Bit0	PSTUP	0b	R/W	Setup interrupt priority bit
1: high priority, 0: low priority **Reset Source: Hardware reset or USB reset**				

7.3. Interrupt Flag

00DAH	IF1	Initial Value	Interrupt Control Flag	
Bit[7:5]	-	0b	-	Reserved
Bit4	TC0	0b	R/W	**Time Capture 0 Interrupt flag**. Set by hardware when the eight bits are received or end condition is detected. Cleared by hardware when interrupt is processed. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit3	T1	0b	R/W	**Base Timer 1 Interrupt flag**. Set by hardware when the Base timer1 overflow is detected. Cleared by hardware when interrupt is processed. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit2	-	0b	-	Reserved
Bit1	T0	0b	R/W	**Base Timer 0 Interrupt flag**. Set by hardware when the Base Timer0 over flow is detected. Cleared by hardware when interrupt is processed. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit0	EXT0	0b	R/W	**External Interrupt 0 flag**. Set by hardware when the P46 falling edge signal is detected. Cleared by hardware when interrupt is processed. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**

00DBH	IF2	Initial Value	Interrupt Control Flag	
Bit7	-	0b	-	Reserved
Bit6	FUN	0b	R/W	**FUN Interrupt flag**. Set by hardware when an invalid program ROM address is detected or the idle time of USB bus large then 5ms. Should be cleared by software at the beginning of interrupt service program. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit5	SIE	0b	R/W	When OUT0, IN0, IN1 or IN2 is responded by a NAK, responds ACK to IN1, IN2 or responds STALL to IN0 or OUT0 tokens, SIE will be set. Should be cleared by software at the beginning of interrupt service program. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit4	OUT0	0b	R/W	When OUT token for endpoint 0 is done, it will set the OUT0 flag. Must be cleared by software at the beginning of interrupt service program. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit3	IN0	0b	R/W	When IN token for endpoint 0 is done, it will set the IN0 flag. Cleared by hardware when interrupt is processed. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit2	OT0ERR	0b	R/W	When an Out token with wrong data sequence is received, OT0ERR will be set 1. Cleared by hardware when interrupt is processed. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit1	OWSTUP	0b	R/W	When a receiving setup token overwrites the existing data in FIFO, R0_OW will set 1. After the overwriting setup packet is received and a following IN or OUT token happens, OWSTUP is set. Cleared by hardware when interrupt is processed. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit0	STUP	0b	R/W	When a SETUP TOKEN for endpoint 0 is done, it will set the STUP flag. Cleared by hardware when interrupt is processed. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**

00DEH	IRQFG	Initial Value	Interrupt Control Flag	
Bit7	IN2	0b	R/W	When IN token for endpoint 2 is done, it will set the IN2 flag. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit6	IN1	0b	R/W	When IN token for endpoint 1 is done, it will set the IN1 flag. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit5	R0STL	0b	R/W	When SH68F86 responds STALL to OUT0 tokens, R0_STL will be set. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit4	T0STL	0b	R/W	When SH68F86 responds STALL to IN0 tokens, T0_STL will be set. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit3	NAK2	0b	R/W	When IN2 is responded by a NAK, NAK2 will be set. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit2	NAK1	0b	R/W	When IN1 is responded by a NAK, NAK1 will be set. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit1	NAKR0	0b	R/W	When OUT0 is responded by a NAK, NAKR0 will be set. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit0	NAKT0	0b	R/W	When IN0 is responded by a NAK, NAKT0 will be set. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**

00DFH	IRQFG2	Initial Value	Interrupt Control Flag	
Bit7	-	0b	-	Reserved
Bit6	-	0b	-	Reserved
Bit5	-	0b	-	Reserved
Bit4	-	0b	-	Reserved
Bit3	-	0b	-	Reserved
Bit2	SUSP	0b	R/W	When USB SIE detects a bus idle state (J state > 5ms), its sets the SUSP Flag. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit1	OVL	0b	R/W	OVL Interrupt 1 flag. Set by hardware when an invalid program ROM address is detected. Write “0” to clear, write “1” no effect. **Reset Source: Hardware reset or USB reset**
Bit0	-	0b	-	Reserved

8. 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. (x = 0, 1)
- ■ 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 **BTCON** (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 source of Base Timer-x is controlled by the BTxM[2:0] register. The following table shows 8 ranges of the Base Timer-x. For counting accuracy, please set the Base Timer-x register first, then preset the BTxM[2:0] register, last, enable 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 **Reset Source: Hardware reset or USB reset**

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

00D4H	**BTCON**	**Initial Value**	**Base Timer Control Register**	
Bit7	ENBT1	0b	R/W	0: Disable Base Timer-1 1: Enable Base Timer-1 **Reset Source: Hardware reset or USB reset**
Bit[6:4]	BT1M[2:0]	000b	R/W	**Base Timer-1 clock source** 000: $F_{BT}/2^0$ 001: $F_{BT}/2^1$ 010: $F_{BT}/2^2$ 011: $F_{BT}/2^3$ 100: $F_{BT}/2^4$ 101: $F_{BT}/2^5$ 110: $F_{BT}/2^6$ 111: $F_{BT}/2^7$ $F_{BT} = F_{SYS}/12$ **Reset Source: Hardware reset or USB reset**
Bit3	ENBT0	0b	R/W	0: Disable Base Timer-0 1: Enable Base Timer-0 **Reset Source: Hardware reset or USB reset**
Bit[2:0]	BT0M[2:0]	000b	R/W	**Base Timer-0 clock source** 000: $F_{BT}/2^0$ 001: $F_{BT}/2^1$ 010: $F_{BT}/2^2$ 011: $F_{BT}/2^3$ 100: $F_{BT}/2^4$ 101: $F_{BT}/2^5$ 110: $F_{BT}/2^6$ 111: $F_{BT}/2^7$ $F_{BT} = F_{SYS}/12$ **Reset Source: Hardware reset or USB reset**

9. USB Control Register

9.1. DADDR

USB Device Address Register

00F2H	DADDR	Initial Value		Device Address Register
Bit7	-	0B	-	Reserved
Bit[6:0]	DADDR[6:0]	0000000B	R/W	USB Device address **Reset Source: Hardware reset (External reset, Power-on reset and Low-Voltage reset) or USB reset**

9.2. DFC

USB Feature Control Register

00F3H	DFC	Initial Value		Device Feature Control Register
Bit7	PULL_UP	0B	R/W	**Internal 1.5K ohm pull up resistor On/Off control** 0: Disable internal USB D- pad 1.5K ohm pull-up resistor 1: Enable internal USB D- pad 1.5K ohm pull-up resistor This F/W controlled function will be mask and is controlled by H/W if USB Mode was enabled (USB_CON = 1) **Reset Source: Hardware reset**
Bit6	USB_CON	0B	R/W	0: Enable GPIO Mode 1: Enable USB Mode **Reset Source: Hardware reset**
Bit5	FW_K	0B	R/W	0: FW stops issuing K-state on USB bus 1: FW starts to issue K-state on USB bus **Reset Source: Hardware reset or USB reset**
Bit4	RSU_SEL	0B	R/W	0: Enable HW to response RESUME by issuing K-state 1: Disable HW to response RESUME by issuing K-state **Reset Source: Hardware reset or USB reset**
Bit3	USBEN	0B	R/W	After power on, USBEN is reset to 0. USBEN will be set to 1 after HOST issues USB reset and then the device starts to respond USB commands. This bit can be also read and written by F/W. 0: Disable USB functions 1: Enable USB functions **Reset Source: Hardware reset**
Bit2	-	0B	-	Reserved
Bit1	ERWUP	0B	R/W	**Remote Wake Up Enable Bit** 0: Disable remote wake-up 1: Enable remote wake-up ERWUP can be returned by SETUP command - GetStatus ( ) to a device ERWUP can be set by SETUP command - ClearFeature (DEVICE_REMOTE_WAKEUP) and SetFeature (DEVICE_REMOTE_WAKEUP). For remote wake-up function, H/W designer and F/W programmer must follow the below notes. Remote wake bit in DFC register can only be set/reset by HOST. **Reset Source: Hardware reset or USB reset**
Bit0	VPCON	1B	R/W	**USB Virtual Plug-off Control** 0: Perform USB plug-in only if the device is disconnected 1: Perform USB pseudo plug-off **Reset Source: Hardware reset**

9.3. TXDATx

USB Transmit FIFO Data Register, x = 0/1/2 for Endpoint 0/1/2. The byte count of the transmitted data must be equal to or less than 8.

00EAH	TXDAT0	Initial Value	USB TX FIFO 0 Data Register	
Bit[7:0]	TXDAT0[7:0]	XXH	W	Transmit FIFO 0 **Reset Source: no reset source**

00E2H	TXDAT1	Initial Value	USB TX FIFO 1 Data Register	
Bit[7:0]	TXDAT1[7:0]	XXH	W	Transmit FIFO 1 **Reset Source: no reset source**

00E5H	TXDAT2	Initial Value	USB TX FIFO 2 Data Register	
Bit[7:0]	TXDAT2[7:0]	XXH	W	Transmit FIFO 2 **Reset Source: no reset source**

9.4. TXCNTx

USB FIFO Transmit Bytes Count Register, x = 0/1/2 for Endpoint 0/1/2. The firmware writes the corresponding bytes count to this register after writing data to the **TXDATx**.

00EBH	TXCNT0	Initial Value	USB TX FIFO 0 Bytes Count Register	
Bit[7:4]	-	0000B	-	Reserved
Bit[3:0]	TXCNT0[3:0]	XXXXB	W	TX FIFO 0 Transmit Bytes Count **Reset Source: no reset source**

00E3H	TXCNT1	Initial Value	USB TX FIFO 1 Bytes Count Register	
Bit[7:4]	-	0000B	-	Reserved
Bit[3:0]	TXCNT1[3:0]	XXXXB	W	TX FIFO 1 Transmit Bytes Count **Reset Source: no reset source**

00E6H	TXCNT2	Initial Value	USB TX FIFO 2 Bytes Count Register	
Bit[7:4]	-	0000B	-	Reserved
Bit[3:0]	TXCNT2[3:0]	XXXXB	W	TX FIFO 2 Transmit Bytes Count **Reset Source: no reset source**

9.5. TXFLGx

USB Transmit FIFO Flag/Control Register, x = 0/1/2 for Endpoint 0/1/2.

00ECH	TXFLG0	Initial Value	USB TX FIFO 0 Flag/Control Register	
Bit[7:2]	-	000000B	-	Reserved
Bit1	STLT0	0	R/W	**Pipe 0 stall bit** 0: SIE responds ACK, NAK or not respond to pipe 0 IN token 1: STLT0 bit is used to stall the pipe 0 IN token. SIE will respond STALL to pipe 0 IN token as long as STLT0 bit is set **Reset source: Hardware reset or USB reset**
Bit0	T0FULL	0	R/W	**TXDAT0 FIFO full status bit**. F/W writes “1” to set H/W FIFO pointer. Clear to “0” by H/W after receiving ACK from host. 0: empty 1: full **Reset Source: Hardware reset or USB reset**

00E4H	**TXFLG1**	**Initial Value**	**USB TX FIFO 1 Flag/Control Register**	
Bit[7:4]	-	0000B	-	Reserved
Bit3	T1EPE	0b	R/W	**This bit is used to enable/disable the endpoint 1** 1: Enable endpoint 1 0: Disable, the corresponding endpoint does not respond to a valid IN Token **Reset source: Hardware reset or USB reset**
Bit2	T1SEQC	0b	W	The data sequence of each transmitted data packet is controlled by hardware and is toggled after receiving ACK from host. The F/W can reset the data sequence by writing "1" to T1SEQC for resetting the next transmitting data sequence on endpoint 1. Write "0" to no effect. Read this bit will always get value with "0" **Reset source: Hardware reset or USB reset**
Bit1	STLT1	0b	R/W	**Pipe 1 stall bit**, this bit is used to stall the pipe 1. STL1 is set by SETUP command - SetFeature (ENDPOINT_HALT) and STL1 is reset by SETUP command - ClearFeature (ENDPOINT_HALT). 0: responds ACK, NAK or not respond to IN1 1: STLT1 bit is used to stall the pipe 1 IN token. SIE will respond STALL to Host IN token as long as STLT1 bit is set **Reset source: Hardware reset or USB reset**
Bit0	T1FULL	0b	R/W	**TXDAT1 FIFO full status bit**. F/W writes "1" to set H/W FIFO pointer. Clear to "0" by H/W after receiving ACK form host. 0: Empty 1: Full **Reset Source: Hardware reset, USB reset**

00E7H	**TXFLG2**	**Initial Value**	**USB TX FIFO 2 Flag/Control Register**	
Bit[7:4]	-	0000B	-	Reserved
Bit3	T2EPE	0b	R/W	**This bit is used to enable the endpoint 2**. 1: Enable endpoint 2 0: Disable, the corresponding endpoint does not respond to a valid IN Token **Reset source: Hardware reset or USB reset**
Bit2	T2SEQC	0b	W	The data sequence of each transmitted data packet is controlled by hardware and is toggled after receiving ACK from host. The F/W can reset the data sequence by writing "1" to T2SEQC for resetting the next transmitting data sequence on endpoint 2. Write "0" to no effect. Read this bit will always get value with "0" **Reset source: Hardware reset or USB reset**
Bit1	STLT2	0b	R/W	**Pipe 2 stall bit**, this bit is used to stall the pipe 2. STL2 is set by SETUP command - SetFeature (ENDPOINT_HALT) and STL2 is reset by SETUP command - ClearFeature (ENDPOINT_HALT). 0: responds ACK, NAK or not respond to IN2 1: STLT2 bit is used to stall the pipe 2 IN token. SIE will respond STALL to Host IN token as long as STLT2 bit is set **Reset source: Hardware reset or USB reset**
Bit0	T2FULL	0b	R/W	**TXDAT 2 FIFO full status bit**. F/W writes "1" to set H/W FIFO pointer. Clear to "0" by H/W after receiving ACK form host. 0: Empty 1: Full **Reset Source: Hardware reset or USB reset**

The TX FIFO operational model refers to Figure 9-1.

In the following, the related F/W procedures and H/W actions are described.

(1) After Hardware Reset or USB Reset, the **TxFULL** bit in **TXFLGx** will reset to 0 to announce no data in FIFOs (x = 0/1/2).

(2) F/W writes up to n bytes of data to the **TXDATx** FIFO. (n = 0-8)

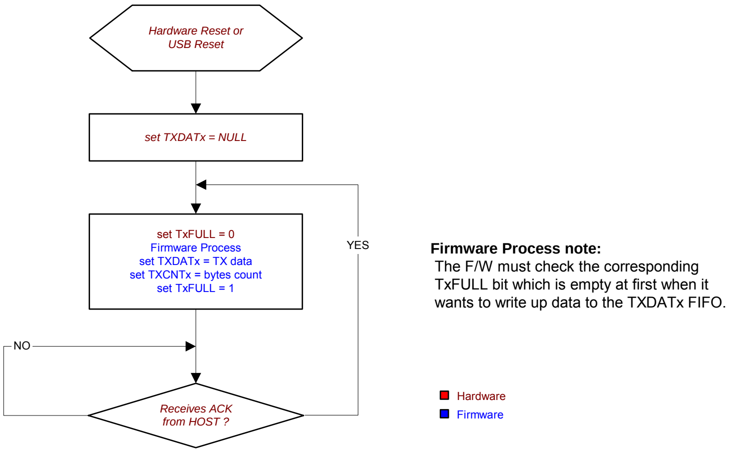
(3) F/W writes data byte count to the corresponding **TXCNTx** register.

(4) F/W sets the **TxFULL** bit.

(5) SIE issues data from the corresponding FIFO byte-by-byte after SIE receives a valid corresponding IN transaction.

(6) SIE waits the ACK.

(7) After SIE receives ACK package successively, the **TxFULL** bit is then reset to 0 by H/W. If SIE don‘t receive ACK, **TxFULL** is on its original status.



Firmware Process note:
The F/W must check the corresponding TxFULL bit which is empty at first when it wants to write up data to the TXDATx FIFO.

Figure 9-1. TX FIFO Operating Model (for a valid IN Transaction)

9.6. RXDAT0

USB Receive FIFO Data Register for Endpoint 0.SIE writes data to the RXDAT0 FIFO for Endpoint 0. CPU read data from the RXDAT0 for Endpoint 0. The operational model refers to Figure 9-2.

00EDH	RXDAT0	Initial Value	USB RX FIFO 0 Data Register	
Bit[7:0]	RXDAT0 [7:0]	XXH	R	RX FIFO Data Register for Endpoint 0 **Reset Source: no reset source**

9.7. RXCNT0

USB Received FIFO bytes count register for Endpoint 0. SIE writes the corresponding bytes count to this register after writing data to the RXDAT0.

00EEH	RXCNT0	Initial Value	USB RX FIFO 0 Bytes Count Register	
Bit[7:4]	-	0000B	-	Reserved
Bit[3:0]	RXCNT0 [3:0]	XXXXB	W	RX FIFO bytes count register for Endpoint 0 **Reset Source: no reset source**

9.8. RXFLG0

USB Receive FIFO Flag/Control Register for Endpoint 0

00EFH	RXFLG0	Initial Value	USB RX FIFO Flag/Control Register	
Bit[7:6]	-	00B	-	Reserved
Bit5	RXERR	0B	R/W	**Receiving error on pipe 0**. When device receives a DATA packet with CRC or bit stuffing errors, this bit is set. Write "0" to clear, Write "1" no effect. **Reset Source: Hardware reset or USB reset**
Bit4	R0_OW	0B	R	This bit is set as long as receiving FIFO is corrupted by setup token **Reset Source: Hardware reset or USB reset**
Bit3	R0SEQ	0B	R	**The data toggle bit of receiving transaction on pipe 0**. This bit is updated by hardware as long as pipe 0 receives a setup or out transaction. **Reset Source: Hardware reset or USB reset**
Bit2	OUT0ENB	0B	R/W	0: The device will receive the data of OUT0 packet when RX FIFO 0 is empty and respond ACK if no bit stuffing error or CRC error. 1: The SH68F86 will respond OUT0 token with NAK. **Reset Source: Hardware reset or USB reset**
Bit1	STLR0	0B	R/W	**Pipe 0 stall bits**. STLR0 bit is used to stall the pipe 0 OUT token. 0: responds ACK, NAK or not respond to OUT token. 1: SIE will respond STALL to HOST OUT token. **Reset Source: Hardware reset or USB reset**
Bit0	R0FULL	0B	R/W	**RXDAT0 FIFO full bit**. Set to "1" by H/W when the RX FIFO 0 fills with valid data. 0: Empty. 1: Full. Write "0" to clear, Write "1" no effect. **Reset Source: Hardware reset or USB reset**

9.9. CRWCON

EP0 Control Read/Write Function Control Register

00E9H	CRWCON	Initial Value	EP0 Control Read/Write Setup Register	
Bit[7:3]	-	00000B	-	**Reserved**
Bit2	CRSEQ	0B	R/W	Select "Valid OUT0 Token" for "STLCR" as Data 1 or Data 0/1. 0: "Valid OUT0 Token" include both OUT Token with "Data 1" & "Data 0" 1: "Valid OUT0 Token" means only OUT Token with "Data 1" **Reset Source: Hardware reset, USB reset, SETUP**
Bit1	STLCR	0B	R/W	1: Enable H/W set "STLR0" and "STLT0" bits when a "valid OUT0 token" was processed 0: Disabled **Reset Source: Hardware reset, USB reset, SETUP**
Bit0	STLCW	0B	R/W	1: Enable H/W set "STLT0" and "STLR0" bits when a "valid IN0 token" was processed 0: Disabled **Reset Source: Hardware reset, USB reset, SETUP**

CRSEQ	STLCR	STLCW	Valid OUT0 Data 0	Valid OUT0 Data 1	Valid IN0 Token	Note
0	0	0	STLT0 = 0 & STLR0 = xx	STLT0 = 0 & STLR0 = xx	STLT0 = xx & STLR0 = 0	
1	0	0	STLT0 = 0 & STLR0 = xx	STLT0 = 0 & STLR0 = xx	STLT0 = xx & STLR0 = 0	
0	1	0	STLT0 = 1 & STLR0 = 1	STLT0 = 1 & STLR0 = 1	STLT0 = xx & STLR0 = 0	
1	1	0	STLT0 = 0 & STLR0 = xx	STLT0 = 1 & STLR0 = 1	STLT0 = xx & STLR0 = 0	
0	0	1	STLT0 = 0 & STLR0 = xx	STLT0 = 0 & STLR0 = xx	STLT0 = 1 & STLR0 = 1	
1	0	1	STLT0 = 0 & TLR0 = xx	STLT0 = 0 & STLR0 = xx	STLT0 = 1 & STLR0 = 1	
0	1	1	STLT0 = 0 & STLR0 = 1	STLT0 = 0 & STLR0 = 1	STLT0 = 1 & STLR0 = 0	Illegal
1	1	1	STLT0 = 0 & STLR0 = xx	STLT0 = 0 & STLR0 = 1	STLT0 = 1 & STLR0 = 0	Illegal

***Note1**: xx means unchanged*

***Note2:** Set the control register in the illegal condition will result in abnormal state under EP0 Control Read/Write Transfer.*

The RX FIFO operational model refers to Figure 9-2.

In the following, the related F/W procedures and H/W actions are described.

*(1) After Hardware Reset or USB Reset, the **R0FULL** bit in **RXFLG0** will reset to 0 to announce no data in **RXDAT0** FIFO.*

(2) SIE receives data (a valid SETUP Transaction or a valid OUT Transaction) byte-by-byte from USB transceiver.

(3) SIE issues ACK.

*(4) A SETUP or OUT IRQ occurs and H/W writes data and bytes count to the **RXDAT0** and **RXCNT0** registers.*

*(5) H/W sets the **R0FULL** bit to "1".*

*(6) After F/W read data from **RXDAT0** FIFO, F/W has to set the **R0FULL** bit to "0".*

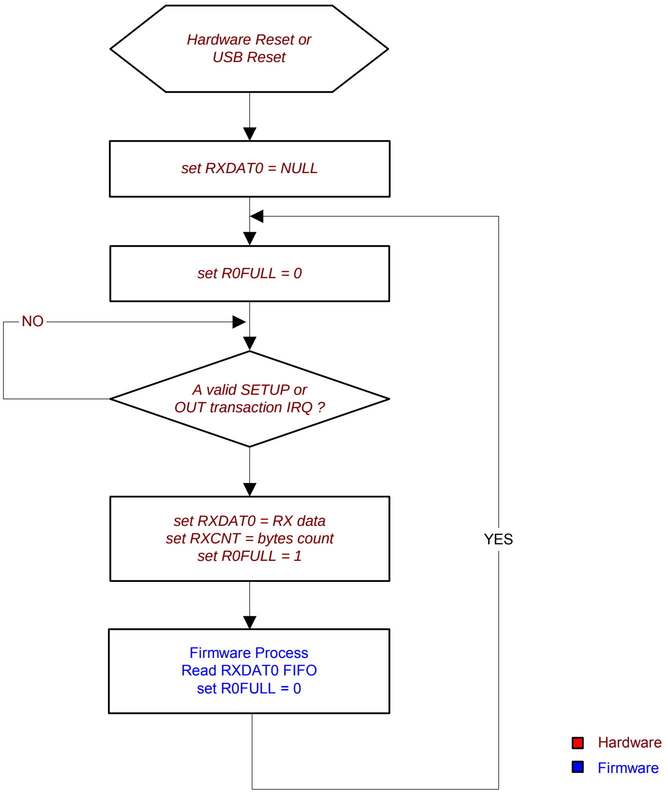


Figure 9-2. RXFIFO Operation Model

10. Flash Program Memory

10.1. General Description

The SH68F86 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.

User can program the 16K flash-type ROM in:

ICP (In-Circuit Programming) mode: Using an external Flash Programmer to perform all the operations to the flash-type ROM and the Information Block, such as erase or write. The read or write operation of ICP mode is done by byte, but the erase operation is done by sector or whole area.

SSP (Self Sector Programming) mode: SSP codes in Program Memory could erase read or write the flash-type ROM. The read or write operation of SSP mode is done by byte, but the erase operation is done by sector.

The flash-type ROM of SH68F86 supports the following operations:

10.1.1. Mass Erase

Mass Erase operation will erase all the contents of program code.

Mass Erase is available in ICP mode only.

11.1.2. Sector Erase

Sector Erase operation will erase the contents of program code of selected sector. This operation can be achieved in ICP mode or SSP mode.

To Sector Erase in SSP mode, the program can not erase its own sector.

See Self Sector Programming chapter for more details.

10.1.3. Write/Read Code

Write/Read Code operation will write the user code into the flash-type ROM or read the user code from ROM. This operation can be achieved in ICP mode or SSP mode.

To Write/Read Code in SSP mode, the program can read/write its own sector.

See Self Sector Programming chapter for more details.

Summary Table

Operation	ICP	SSP
Mass Erase	Support	Not support
Sector Erase	Support	Support
Write/Read Code	Support	Support

10.2. Operation in ICP (In-Circuit Programming) Mode

ICP mode is performed without removing the micro-controller from the system. In ICP mode, the user system must be power-off, and the Flash Programmer can refresh the program memory through ICP programming interface. The ICP programming interface consists of 6 wires (V_{DD}, GND, TCK, TDI, TMS, TDO).

The SH68F86 will enter ICP mode once specified waveform of TCK, TDI, TMS and TDO pins is detected within a limited period after system POR. To get more details, please refer Flash Programmer's user manual.

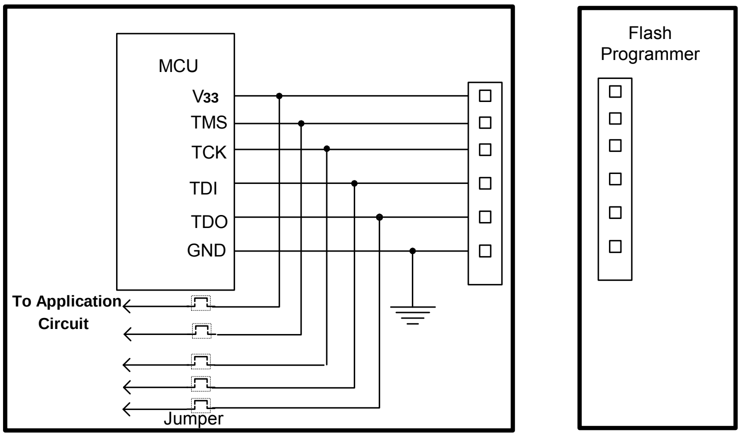


Figure 10-1. Typical application circuit for ICP mode

The recommended operating flow:

(1) The jumper must be open to separate the programming pins from the application circuit before programming.
(2) Connect the Flash Programmer can MCU through the 6-wire interface, and begin programming.
(3) Disconnect the Flash Programmer and short these jumpers after programming is complete.

11.3. SSP (Self Sector Programming)

The SH68F86 provides SSP function, and each sector of flash-type ROM can be sector erased or programmed by user code. But once sector has been programmed, it cannot be reprogrammed before Sector Erase.

To prevent the data from carelessly writing main block, the user must do five states (S0, S1, S2, S3, and S4) in sequence. If the dedicated conditions are not met from **IB_CON1** to **IB_CON5**, the SSP will be terminated. User can read other sectors through MOVC instruction.

10.3.1. Registers

00B3H	IB_CON1	Initial Value		Information block Control Register 1
Bit[7:0]	IB_CON1	00h	R/W	**Enable or disable Write/Erase operation** E6H: Erase the selected block 6EH: Write to the selected block Other: don't care **Reset source: Hardware reset or WDT reset**

00B4H	IB_CON2	Initial Value		Information block Control Register 2
Bit[7:4]	-	0000b	-	Reserved
Bit[3:0]	IB_CON2	0000b	R/W	5H: enter S1 Other: enter S0 **Reset source: Hardware reset or WDT reset**

00B5H	IB_CON3	Initial Value	Information block Control Register 3	
Bit[7:4]	-	0000b	-	Reserved
Bit[3:0]	IB_CON3	00h	R/W	AH: enter S2 Other: enter S1 **Reset condition: common reset|IB_CON2≠5H**

00B6H	IB_CON4	Initial Value	Information block Control Register 4	
Bit[7:4]	-	0000b	-	Reserved
Bit[3:0]	IB_CON4	0000b	R/W	9H: enter S3 Other: enter S2 **Reset condition: common reset|IB_CON2≠5H|IB_CON3≠AH**

00B7H	IB_CON5	Initial Value	Information block Control Register 5	
Bit[7:4]	-	0000b	-	Reserved
Bit[3:0]	IB_CON5	0000b	R/W	6H: enter S4 Other: enter S3 **Reset condition:** **common reset|IB_CON2≠5H|IB_CON3≠AH|IB_CON4≠9H**

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 (**Note1**) 010000 - 111111: reserved **Reset source: Hardware reset or WDT reset**
Bit[1:0]	XPAGE[1:0]	00b	R/W	High address of offset of the flash memory sector to be programmed (**Note2**) **Reset source: Hardware reset or WDT reset**

Note 1*: Define the number of sector in the* ***XPAGE*** *register. For 16K flash-type ROM, the size of each sector is 1024 bytes.*

Note 2: *Combine* ***XPAGE****[1:0] and* ***IB_OFFSET[****7:0] to become 10-bit offset byte so that the user can assign one byte of the information block for writing or reading.*

00BEH	IB_OFFSET	Initial Value	Information Block Control Register	
Bit[7:0]	IB_OFFSET	00h	R/W	Low address of offset of the flash memory sector to be programmed … (**Note2**) **Reset source: Hardware reset or WDT reset**

00BFH	IB_DATA	Initial Value	Data Register for programming information block	
Bit[7:0]	IB_DATA	00h	R/W	**Reset source: Hardware reset or WDT reset**

Offset byte	High address		Low address of offset of the flash memory sector to be programmed							
	XPAGE[1:0]		IB_OFFSET[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

10.3.2. Notice of SSP

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

A. For Code/Data programming

Step 1: Disable interrupt
Step 2: Fill in the **XPAGE** and **IB_OFFSET** for the corresponding address
Step 3: Fill in **IB_DATA** if programming is wanted
Step 4: Fill in **IB_CON1-5** 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 Sector Erase

Step 1: Disable interrupt
Step 2: Fill in the **XPAGE** for the corresponding sector
Step 3: Fill in **IB_CON1-5** sequentially
Step 4: Sector Erase, CPU will be in IDLE mode
Step 5: Add 4 NOPs. (If one more sector wants 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”

10.4 Flash Control Flow

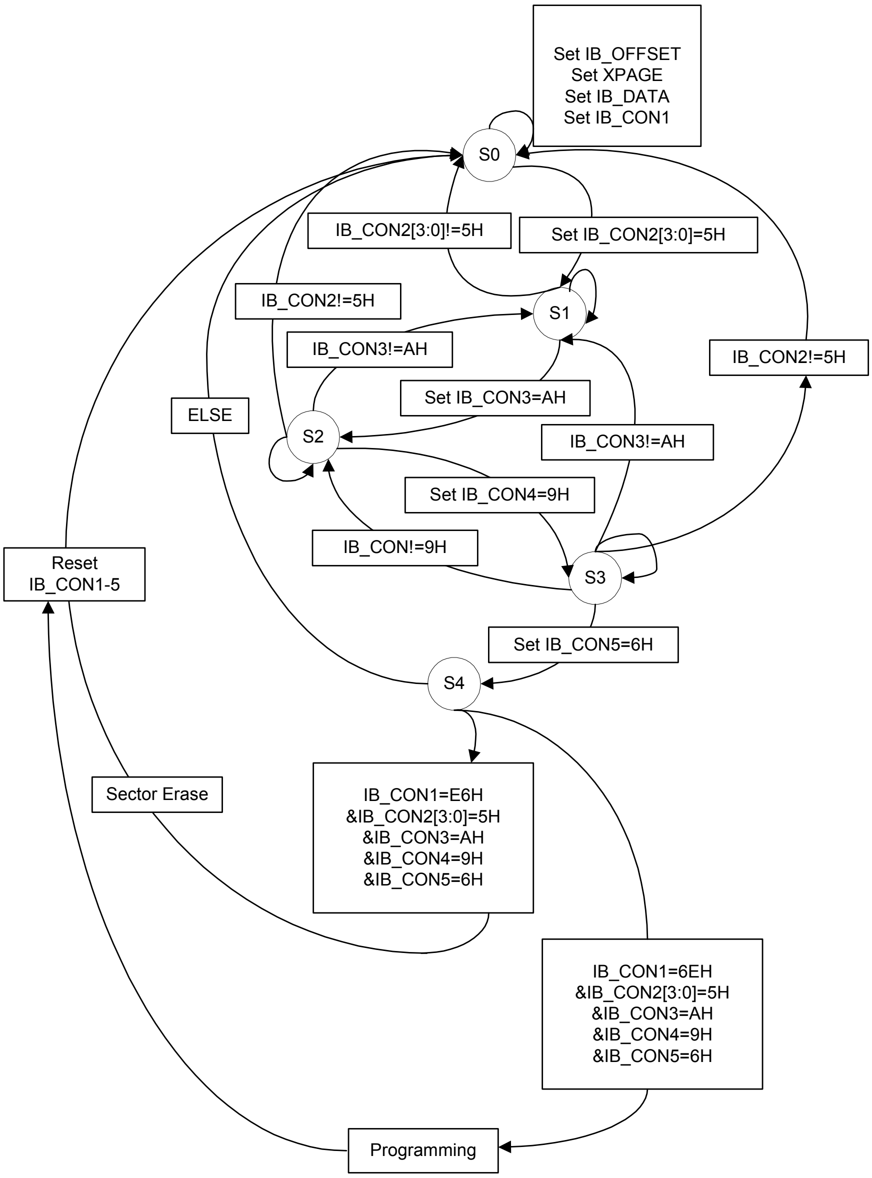


Figure 10-2. Flash Control Flow

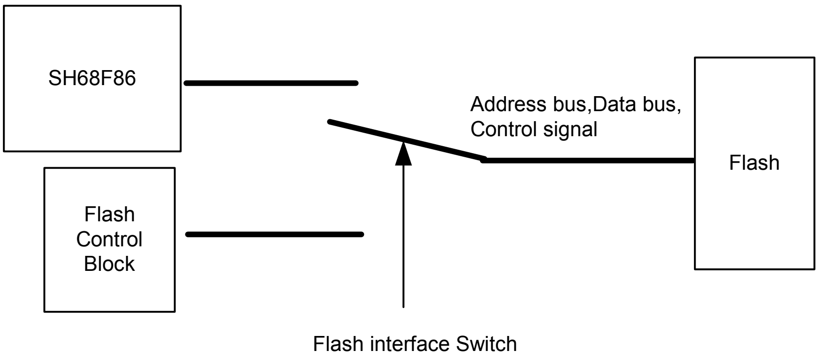


Figure 10-3. Flash interface Block Diagram

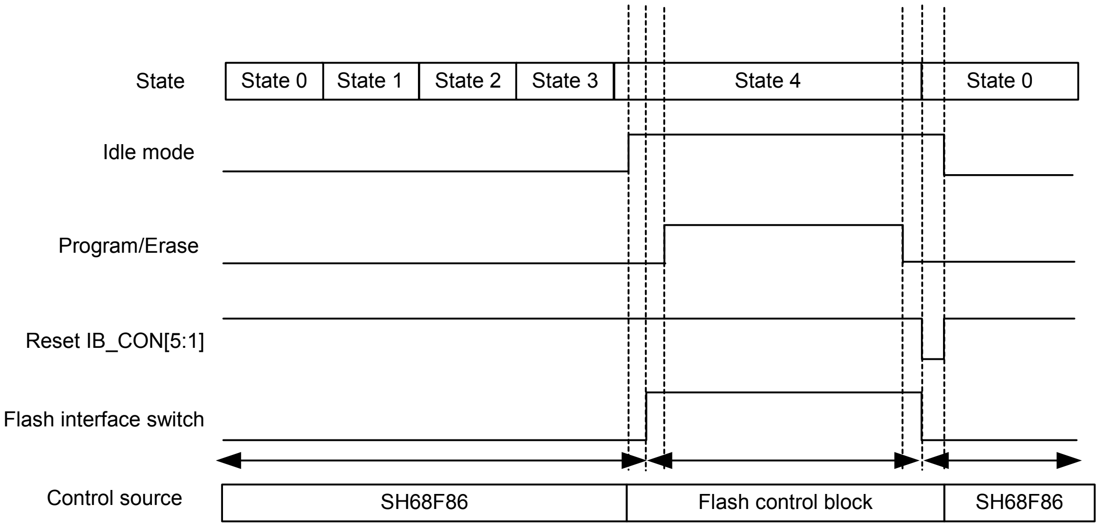


Figure 10-4. Flash Control Timing Diagram

11. Electrical Characteristics

Absolute Maximum Ratings*

Input/Output Voltage . . . . GND - 0.2V to V_{DD} + 0.2V

Operating Ambient Temperature . . . . 0℃ to +70℃

Storage Temperature . . . . -55℃ to +125℃

Operating Voltage (V_{DD}) . . . . +4.4V to 5.5V

*Comments

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to this 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 or intended. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

DC Electrical Characteristics (V_{DD} = 5V, GND = 0V, T_A = 25°C, f_{OSC} = 6MHz, unless otherwise noted)

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNIT	CONDITIONS
Main Power						
Operating Voltage	V_{DD}	4.4	5	5.25	V	
Operating Current 1 (f_{SYS} = 6Mhz)	I_{OP}	-	-	10	mA	f_{OSC} = 12MHz, V_{DD} = 5.0V All output pins unload (including all digital input pins un-floating) CPU on (execute NOP instruction), WDT on, 32K Wake-up timer off, all other function block on
Idle mode Current	I_{IDLE}	-	-	6	mA	f_{OSC} = 12MHz, V_{DD} = 5.0V All output pins unload, CPU off (IDLE), WDT on, 32K Wake-up timer off, all other function block off
Power down Current	I_{PD}	-	-	200	μA	OSC off, V_{DD} = 5.0V All output pins unload, CPU off (Power-Down), WDT off, 32K Wake-up timer on, LDO on, all other function block off
Regulator						
3.3V Regulator Voltage	V_{33} (V33)	-	3.3	-	V	I_{OUT33} = 1mA, V_{DD} = 5V
3.3V Maximum Output Current	I_{OUT33}	-	25	35	mA	ΔV_{33} = -0.1V (max)
V33 dropout voltage	V_{DROP33}	-	-	0.1	V	V_{DD} = 5V, I_{OUT33} = 35mA
1.8V Regulator Voltage	V_{18}	1.7	1.8	1.9	V	I_{18} = 1mA, V_{DD} = 5V
1.8V Maximum Output Current	I_{OUT18}	-	-	10	mA	ΔV_{18} = -0. 05V (max)
V18 dropout voltage	V_{DROP18}	-	-	0.05	V	V_{DD} = 5V, I_{OUT18} = 10mA
GPIO and LED Port						
Input Low Voltage Port0, Port1[0]	V_{IL1}	GND	-	0.2*V_{33}	V	IO, INT0 (V_{33} = 3.3V) (Schmitt Trigger)
Input Low Voltage Port1[3:1]	V_{IL2}	GND	-	0.2*V_{DD}	V	IO (V_{DD} = 5V)
Input High Voltage Port0, Port1[0]	V_{IH1}	0.8*V_{33}	-	V_{33}	V	IO, INT0 (V_{33} = 3.3V) (Schmitt Trigger)
Input High Voltage Port1[3:1]	V_{IH2}	0.8*V_{DD}	-	V_{DD}	V	IO (V_{DD} = 5V)
Output Low Voltage Port0[7:0], Port1[0]	V_{OL1}	0	-	0.4*V_{33}	V	I/O Ports I_{OL} = 2mA
Output Low Voltage Port1[4]	V_{OL2}	2.6	-	3.2	V	I_{OL} = 9mA (Typ.) (LED)
Output high voltage Port0[7:0], Port1[0]	V_{OH1}	0.6*V_{33}	-	V_{33}	V	V_{33} = 3.3V I_{OH} = -2mA
Output high voltage Port1[4]	V_{OH2}	2.4	-	-	V	I_{OH} = -50μA ~ -100μA
Input pull-up Register For P10	R_L	20	30	40	KΩ	

(continued)

Reset (DC)						
Power-on Reset Level	V_{POR}			3.6	V	
Auxiliary Lower-voltage Reset Level	V_{LVRA}	2.9	3.0	3.1	V	
Low Voltage Reset 1 Level	V_{LVR1}	1.4	1.5	1.6	V	
Low Voltage Reset 2 Level	V_{LVR2}	2.2	2.4	2.6	V	
Upper Threshold Voltage for external Reset	$V_{UT(RESET)}$	2	-	-	V	
Lower Threshold Voltage for external Reset	$V_{LT(RESET)}$	-	-	0.8	V	

AC Electrical Characteristics (V_{DD} = 5V, GND = 0V, T_A = 25°C, f_{OSC} = 6MHz, unless otherwise noted)

Oscillator						
PARAMETER	**SYMBOL**	**MIN.**	**TYP.**	**MAX.**	**UNIT**	**CONDITIONS**
Internal RC Frequency 1	F_{SYS}	11.82	12	12.18	MHz	±1.5%
Internal RC Frequency 2	F_{RING}	27.2	32	36.8	KHz	±15%

Reset (AC)						
PARAMETER	**SYMBOL**	**MIN.**	**TYP.**	**MAX.**	**UNIT**	**CONDITIONS**
Power On Reset Time	$T_{RST(POR)}$	-	2^{17}	-	T_{SYS}	F_{SYS} = 12MHz
Low Voltage Reset Time	$T_{RST(LVR)}$	-	2^{17}	-	T_{SYS}	F_{SYS} = 12MHz
Drop-Down Width for LVR1	$T_{PW(LVR1)}$	-	2^{10}	-	T_{SYS}	F_{SYS} = 12MHz
Drop-Down Width for LVR2	$T_{PW(LVR2)}$	-	2^{10}	-	T_{SYS}	F_{SYS} = 12MHz
Drop-Down Width for LVRA	$T_{PW(LVRA)}$	-	2^{10}	-	T_{SYS}	F_{SYS} = 12MHz
Watch-Dog Reset Hold Time	$T_{RST(WDT)}$	-	500	-	μs	
Internal USB Reset Hold Time	$T_{RST(USB)}$	2	-	4	T_{SYS}	F_{SYS} = 12MHz
SE0 Width for USB Reset	T_{URST}	22	-	-	μs	
SE0 Width for USB Reset (power-down mode)	T_{URST1}	3	-	-	ms	
Internal Resume Reset Width (Global wake-up)	T_{WKRST1}	-	2.7	-	ms	
Internal Resume Reset Width (Remote Wakeup, **RSU_SEL** = 0) (HW Issue K)	T_{WKRST2}	-	18.4	-	ms	
Internal Resume Reset Width (Remote Wakeup, **RSU_SEL** = 1)	T_{WKRST3}	-	5.4	-	ms	
Internal Resume Reset Width (Wake-Up timer)	T_{WKRST4}	2^{11}	-	2^{18}	T_{SYS}	F_{SYS} = 12MHz, (Depend on Period[1:0])
Noise cancellation for ExT0	$T_{PW(EXT0)}$	-	-	2^{3}	T_{SYS}	F_{SYS} = 12MHz

USB DC/AC Specifications

- Please refer to the UNIVERSAL SERIAL BUS specification Version 1.1 Chapter 7.
- Some items are listed in the following table.
- In addition, the crossover point voltage should meet the following specifications.

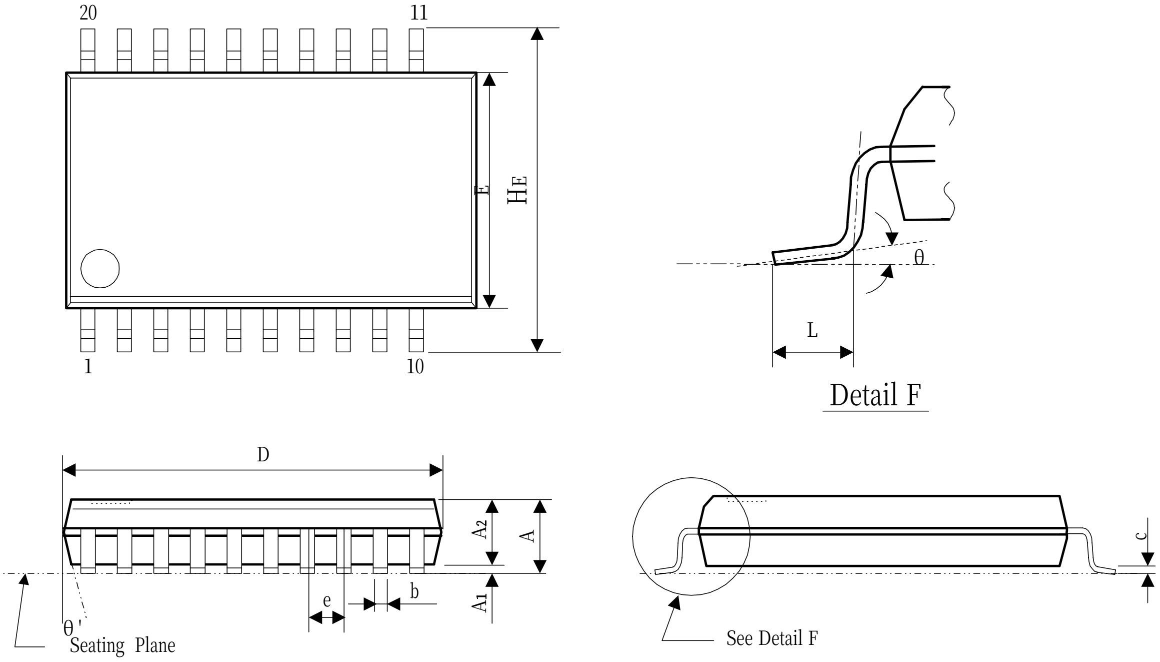
PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNIT	CONDITIONS
Input High Voltage (Driven)	V_{IH} (USB)	2.0	-	-	V	DM, DP
Input High Voltage (Floating)	V_{IHZ} (USB)	2.7	-	3.6	V	DM, DP
Input Low Voltage	V_{IO} (USB)	-	-	0.8	V	DM, DP
Differential Input Sensitivity	V_{DI} (USB)	0.2	-	-	V	DM, DP (|VDP - VDM|)
Differential Common Mode Range	V_{DM} (USB)	0.8	-	2.5	V	DM, DP (Includes V_{DI} Range)
Output Low Voltage	V_{OL} (USB)	0.0	-	0.3	V	DM, DP
Output High Voltage (Drive)	V_{OH} (USB)	2.8	-	3.6	V	DM, DP
Output Crossover Voltage	V_{CRS} (USB)	1.3	-	2.0	V	DP, DM, V_{DD} = 4.4V - 5.25V
Delay time of USB data output from output enable	$T_{USB(0DALEY)}$		334		ns	
Transition Time. Rise Time/Fall Time	$T_{LR/LF}$	75		300	ns	Between 10% and 90% Device Load CL = 50pF to 150pF, with external 15K ohm pull down R

Ordering Information

Part No.	Package
SH68F86H	Chip Form
SH68F86X/020XU	20 TSSOP

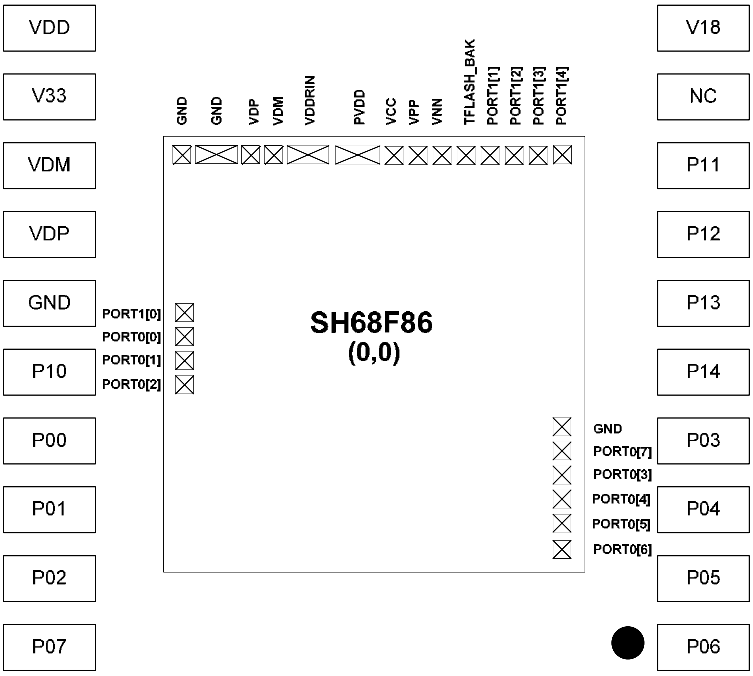
Package Information

TSSOP 20L Outline Dimensions unit: inches/mm



Symbol	Dimensions in inches		Dimensions in mm	
	MIN	**MAX**	**MIN**	**MAX**
A	---	0.048	---	1.2
A1	0.002	0.006	0.05	0.15
A2	0.031	0.041	0.8	1.05
b	0.007	0.012	0.18	0.3
C	0.004	0.008	0.09	0.2
D	0.252	0.26	6.4	6.6
E	0.169	0.177	4.3	4.5
HE	0.246	0.258	6.25	6.55
e	0.026(BSC)		0.65(BSC)	
L	0.018	0.03	0.45	0.75
θ	0°	8°	0°	8°

Bonding Diagram



Pad Location unit: µm

Pad NO.	Pad Name	X	Y	20TSSOP	Pad NO.	Pad Name	X	Y	20TSSOP
1	PORT1[0]	655.07	-132.7	16	13	PORT1[2]	-442.6	-599.15	7
2	PORT0[0]	655.07	-51.7	17	14	PORT1[1]	-361.6	-599.15	8
3	PORT0[1]	655.07	29.3	18	15	TFLASH_BAK	-286.45	-599.15	nc
4	PORT0[2]	655.07	110.3	19	16	VNN	-217.15	-599.15	nc
5	PORT0[6]	-655.07	582.37	1	17	VPP	-147.85	-599.15	nc
6	PORT0[5]	-655.07	501.37	2	18	VCC	-78.55	-599.15	10
7	PORT0[4]	-655.07	420.37	3	19	PVDD	32.6	-599.15	11
8	PORT0[3]	-655.07	339.37	4	20	VDDRIN	185.6	-599.15	12
9	PORT0[7]	-655.07	258.37	20	21	VDM	302.6	-599.15	13
10	GND	-655.07	177.37	Bonding to frame	22	VDP	399.8	-599.15	14
11	PORT1[4]	-637.54	-599.15	5	23	GND	516.8	-599.15	15
12	PORT1[3]	-542.14	-599.15	6	24	GND	640.55	-599.15	15& bonding to frame

Data Sheet Revision History

Revision No.	History	Date
2.0	1. Modify Pin Configuration, Ordering Information, Package Information, and Bonding Information as the package form altered into TSSOP20 2. Modify Pad Configuration 3. Modify the range of Input pull-up Register For P10	Apr. 2013
1.0	Original	Nov. 2012