SH88F516（SH88F54/SH89F52）

8051 Microcontroller with 10bit ADC

1. Features (All of the flowing contents are from SH88F516; The characteristics of SH88F54/SH89F52 is slightly different from SH88F516, see "Product Information" section)

- 8bits micro-controller with Pipe-line structured 8051 compatible instruction set
- Flash ROM: 64K Bytes
- RAM: internal 256 Bytes, external 1024 Bytes
- EEPROM-like: 1K Bytes
- Operation Voltage:
  V_{DD} = 3.6V - 5.5V，f_{OSC} = 30kHz - 16.6MHz
- Oscillator (code option)
  - Crystal oscillator: 32.768kHz
  - Crystal oscillator: 400kHz - 16.6MHz
  - Ceramic oscillator: 400kHz - 16.6MHz
  - Internal RC: 16.6MHz
  - External clock:30kHz-16.6MHz
- 40 CMOS bi-directional I/O pins (Quasi-Bi mode, Push-Pull mode, Input-Only mode and Open-Drain mode)
- Three 16-bit timer/counters T0, T1 & T2
- Powerful interrupt sources:
  - Timer0, 1, 2
  - INT0, INT1, INT4 (8 input)
  - EUART0, EUART1, SPI, PWM, SCM, LPD
  - ADC, CMP0, CMP1
- Three 8-bit PWM
- Two built-in comparator (CMP)
- EUART0 & EUART1
- SPI interface (Master/Slave Mode)
- 8channels 10-bits Analog Digital Converter (ADC)
- Low Voltage Detect (LPD)
- Low Voltage Reset (LVR) function (enabled by code option)
  - LVR voltage level 1: 4.3V
  - LVR voltage level 2: 3.7V
- CPU Machine cycle: 1 oscillator clock
- Watch Dog Timer (WDT)
- Warm-up Timer
- System Clock Monitor (SCM)
- Support Low power operation modes:
  - Idle Mode
  - Power-Down Mode
- Low power consumption
- Package:
  - QFP44
  - TQFP48
  - LQFP44

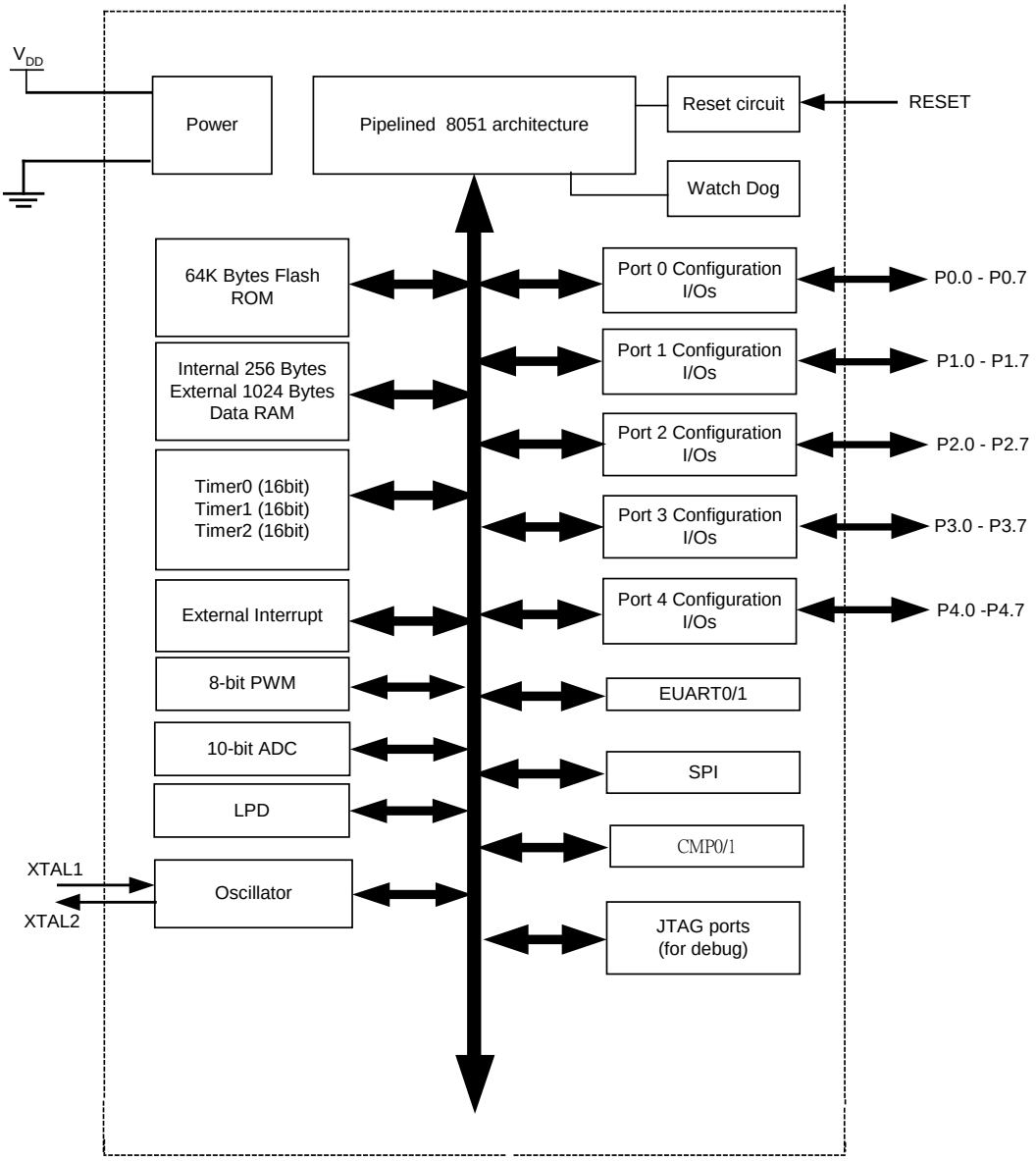
2. General Description

The SH88F516 is a high performance 8051 compatible micro-controller, regard to its build-in Pipe-line instruction fetch structure, that helps the SH88F516 can perform more fast operation speed and higher calculation performance, if compare SH88F516 with standard 8051 at same clock speed.

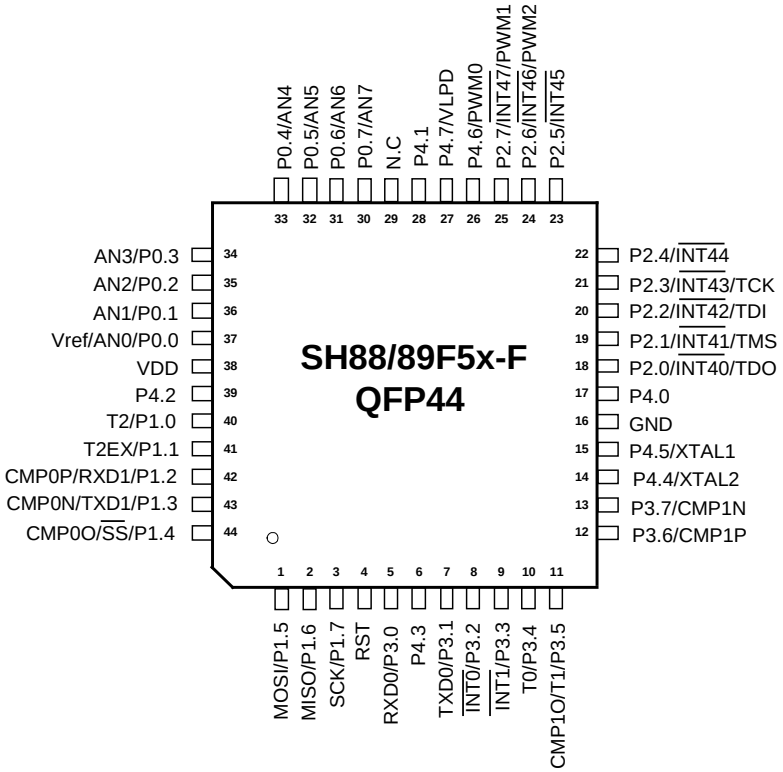
The SH88F516 retains most features of the standard 8051. These features include internal 256 bytes RAM, two 16-bit timer/counter,UART and INT0-1.In addition, SH88F516 provides external 1024 bytes RAM, It also contains 16-bit timer/counter (Timer2) and 64K bytes Flash memory block both for program and data. Also ADC, PWM and CMP are incorporated in SH88F516.

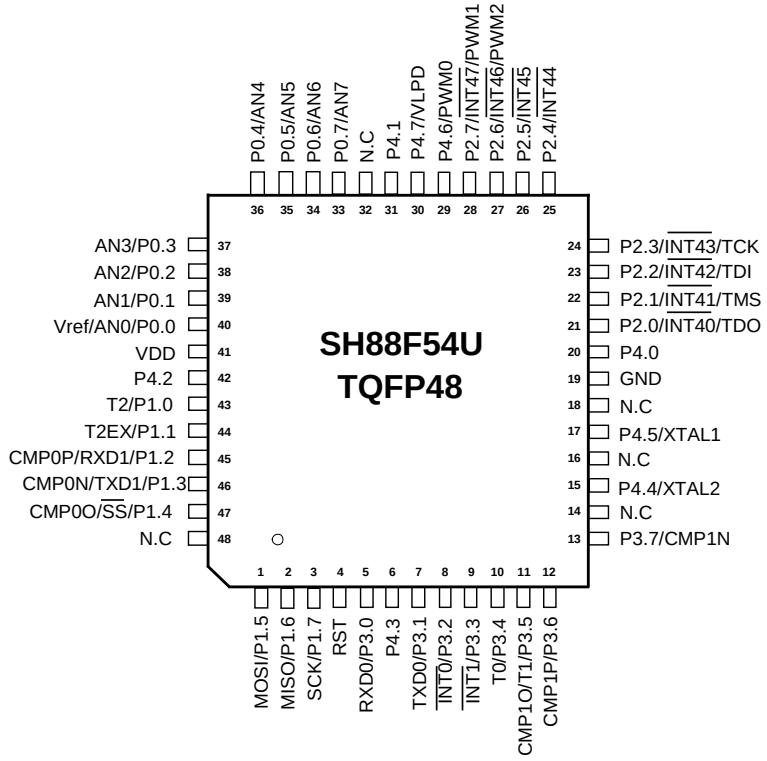
For high reliability and low power consumption, SH88F516 builds in Watchdog Timer, Low Voltage Reset function, LPD function and SCM function. And SH88F516 also supports two power saving modes to reduce power consumption.

3. Block Diagram



4. Pin Configuration





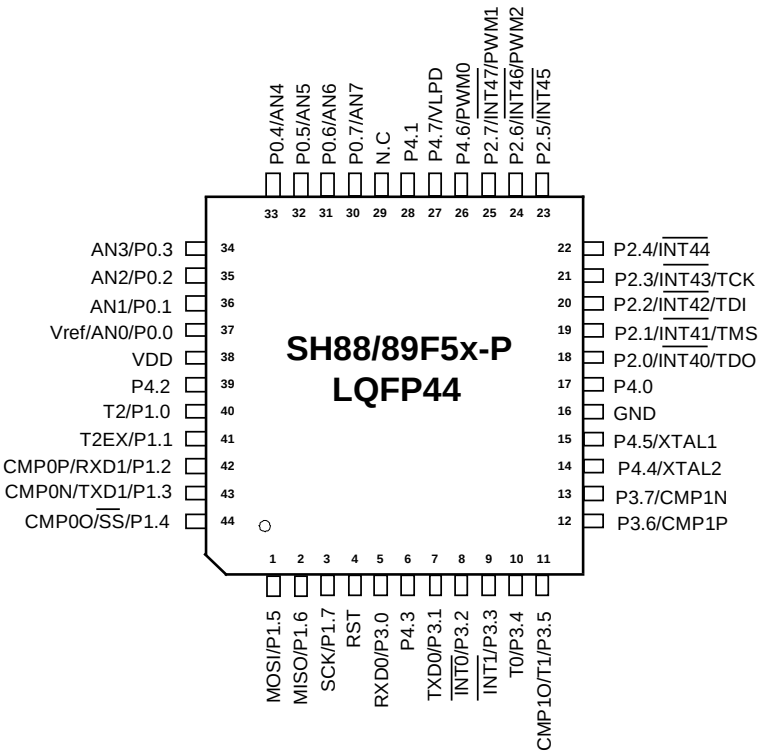

SH88/89F5x-P
LQFP44
P0.4/AN4
P0.5/AN5
P0.6/AN6
P0.7/AN7
N.C
P4.1
P4.7/VLPD
P4.6/PWM0
P2.7/INT47/PWM1
P2.6/INT46/PWM2
P2.5/INT45
AN3/P0.3
AN2/P0.2
AN1/P0.1
Vref/AN0/P0.0
VDD
P4.2
T2/P1.0
T2EX/P1.1
CMP0P/RXD1/P1.2
CMP0N/TXD1/P1.3
CMP0O/SS/P1.4
P2.4/INT44
P2.3/INT43/TCK
P2.2/INT42/TDI
P2.1/INT41/TMS
P2.0/INT40/TDO
P4.0
GND
P4.5/XTAL1
P4.4/XTAL2
P3.7/CMP1N
P3.6/CMP1P
MOSI/P1.5
MISO/P1.6
SCK/P1.7
RST
RXD0/P3.0
P4.3
TXD0/P3.1
INT0/P3.2
INT1/P3.3
T0/P3.4
CMP1O/T1/P3.5

Table 4.1 Pin Function

Pin No.			Pin Name	Default Function
TQFP 48	QFP 44	LQFP 44		
1	1	1	MOSI/P1.5	P1.5
2	2	2	MISO/P1.6	P1.6
3	3	3	SCK/P1.7	P1.7
4	4	4	RST	RST
5	5	5	RXD0/P3.0	P3.0
6	6	6	P4.3	P4.3
7	7	7	TXD0/P3.1	P3.1
8	8	8	$\overline{\text{INT0}}$/P3.2	P3.2
9	9	9	$\overline{\text{INT1}}$/P3.3	P3.3
10	10	10	T0/P3.4	P3.4
11	11	11	CMP1O/T1/P3.5	P3.5
12	12	12	CMP1P/P3.6	P3.6
13	13	13	CMP1N/P3.7	P3.7
14	-	-	N.C	Not connected
15	14	14	XTAL2/P4.4	P4.4 or oscillator output pin(controlled by Code Option)
16	-	-	N.C	Not connected
17	15	15	XTAL1/P4.5	P4.5 oscillator input pin(controlled by Code Option)
18	-	-	N.C	Not connected
19	16	16	GND	GND
20	17	17	P4.0	P4.0
21	18	18	TDO/$\overline{\text{INT40}}$/P2.0	P2.0
22	19	19	TMS/$\overline{\text{INT41}}$/P2.1	P2.1
23	20	20	TDI/$\overline{\text{INT42}}$/P2.2	P2.2
24	21	21	TCK/$\overline{\text{INT43}}$/P2.3	P2.3
25	22	22	$\overline{\text{INT44}}$/P2.4	P2.4
26	23	23	$\overline{\text{INT45}}$/P2.5	P2.5
27	24	24	PWM2/$\overline{\text{INT46}}$/P2.6	P2.6
28	25	25	PWM1/$\overline{\text{INT47}}$/P2.7	P2.7
29	26	26	PWM0/P4.6	P4.6
30	27	27	VLPD/P4.7	P4.7

(to be continued)

(continue)

Pin No.			Pin Name	Default Function
TQFP 48	QFP 44	LQFP 44		
31	28	28	P4.1	P4.1
32	29	29	N.C	Not connected
33	30	30	AN7/P0.7	P0.7
34	31	31	AN6/P0.6	P0.6
35	32	32	AN5/P0.5	P0.5
36	33	33	AN4/P0.4	P0.4
37	34	34	AN3/P0.3	P0.3
38	35	35	AN2/P0.2	P0.2
39	36	36	AN1/P0.1	P0.1
40	37	37	Vref/AN0/P0.0	P0.0
41	38	38	V_{DD}	V_{DD}
42	39	39	P4.2	P4.2
43	40	40	T2/P1.0	P1.0
44	41	41	T2EX/P1.1	P1.1
45	42	42	CMP0P/RXD1/P1.2	P1.2
46	43	43	CMP0N/TXD1/P1.3	P1.3
47	44	44	CMP0O/$\overline{SS}$/P1.4	P1.4
48	-	-	N.C	Not connected

Note:

1. Pin can be configured as open-drain output for N channel, but Pin voltage can't be higher than V_{DD} + 0.3V

2. The out most pin function has the highest priority, and the inner most pin function has the lowest priority (Refer to Pin Configuration Diagram. This means when one pin is occupied by a higher priority function (if enabled) cannot be used as the lower priority functional pin, even when the lower priority function is also enabled. Until the higher priority function is closed by software, can the corresponding pin be released for the lower priority function use.

5. Pin Description

Pin No.	Type	Description
I/O Port		
P0.0 - P0.7	I/O	8 bit General purpose CMOS I/O
P1.0 - P1.7	I/O	8 bit General purpose CMOS I/O
P2.0 - P2.7	I/O	8 bit General purpose CMOS I/O
P3.0 - P3.7	I/O	8 bit General purpose CMOS I/O
P4.0 - P4.7	I/O	8 bit General purpose CMOS I/O
Timer		
T0	I/O	Timer0 external input/Comparator output
T1	I/O	Timer1 external input/Comparator output
T2	I/O	Timer2 external input/Baud-Rate generator
T2EX	I	Timer2 Reload/Capture/Direction Control
PWM		
PWM0	O	Output pin for 8-bit PWM Timer
PWM1	O	Output pin for 8-bit PWM Timer
PWM2	O	Output pin for 8-bit PWM Timer
EUART		
RXD0	I/O	EUART0 data input
TXD0	O	EUART0 data output
RXD1	I/O	EUART1 data input
TXD1	O	EUART1 data output
SPI		
MOSI	I/O	SPI master output slave input
MISO	I/O	SPI master input slave output
SCK	I/O	SPI serial clock
$\overline{SS}$	I	SPI Slave Select
ADC		
AN0 - AN7	I	ADC input channel
V_{REF}	I	External ADC reference voltage input
Interrupt & Reset & Clock & Power		
$\overline{INT0}$ - $\overline{INT1}$	I	External interrupt 0-1 input source
$\overline{INT40}$ - $\overline{INT47}$	I	External interrupt 40-47 input source
RST	I	The device will be reset by A low voltage on this pin longer than 10us, CPU will reset
XTAL1	I	Oscillator input
XTAL2	O	Oscillator output
V_{DD}	P	Power supply (3.6V - 5.5V)
GND	P	Ground

(to be continued)

(continue)

Pin No.	Type	Description
LPD		
VLPD	I	Power Voltage Detect
CMP		
CMP0P	I	CMP0 positive input pin
CMP0N	I	CMP0 negative input pin
CMP0O	O	CMP0 output
CMP1P	I	CMP1 positive input pin
CMP1N	I	CMP1 negative input pin
CMP1O	O	CMP1 output
Programmer		
TDO (P2.0)	O	Debug interface: Test data out
TMS (P2.1)	I	Debug interface: Test mode select
TDI (P2.2)	I	Debug interface: Test data in
TCK (P2.3)	I	Debug interface: Test clock in

Note:
When P2.0-2.3 used as debug interface, functions of P2.0 - 2.3 are blocked

6. Product Information

SH88F516: QFP44、LQFP44

Part Num	RAM (byte)	Flash (byte)	E2 (byte)	EUARTx	CMPx	ADC (10bit)	PWMx	Timerx	SPI	ExINT	LPD Pin
SH88F516	1280	64K	1K	EUART0,1	CMP0,1	AN0-7	PWM0,1,2	Timer0,1,2	Y	2+1(8)	Y

SH88F54: QFP44、TQFP48、LQFP44

Part Num	RAM (byte)	Flash (byte)	E2 (byte)	EUARTx	CMPx	ADC (10bit)	PWMx	Timerx	SPI	ExINT	LPD Pin
SH88F54	768	16K	512	EUART0,1	CMP0,1	AN0-7	PWM0,1,2	Timer0,1,2	Y	2+1(8)	Y

SH89F52: QFP44、LQFP44

Part Num	RAM (byte)	Flash (byte)	E2 (byte)	EUARTx	CMPx	ADC (10bit)	PWMx	Timerx	SPI	ExINT	LPD Pin
SH89F52	512	8K	512	EUART0	CMP0	N	PWM0,1,2	Timer0,1,2	Y	2+1(8)	Y

7. SFR Mapping

The SH88F516 provides 256 bytes of internal RAM to contain general-purpose data memory and Special Function Register (SFR). The SFR of the SH88F516 fall into the following categories:

CPU Core Registers:	ACC, B, PSW, SP, DPL, DPH
Enhanced CPU Core Registers:	AUXC, DPL1, DPH1, INSCON, XPAGE
Power and Clock Control Registers:	PCON, SUSLO
LPD Registers:	LPDCON
Flash Registers:	IB_OFFSET, XPAGE, IB_DATA, IB_CON1, IB_CON2, IB_CON3, IB_CON4, IB_CON5, FLASHCON
Data Memory Register:	XPAGE
Hardware Watchdog Timer Registers:	RSTSTAT
System Clock Control Register:	CLKCON
Interrupt System Registers:	IEN0, IEN1, IENC, IPH0, IPL0, IPH1, IPL1, EXF0, EXF1
I/O Port Registers:	P0, P1, P2, P3, P4, P0M0, P0M1, P1M0, P1M1, P2M0, P2M1, P3M0, P3M1, P4M0, P4M1
Timer Registers:	TCON, TMOD, TH0, TH1, TL0, TL1, TCON1, T2CON, T2MOD, TH2, TL2, RCAP2L, RCAP2H
EUART0 Registers:	SCON, SBUF, SADEN, SADDR, PCON
EUART1 Registers:	SCON1, SBUF1, SADEN1, SADDR1, SBRT0/1, PCON
SPI Registers:	SPCON, SPSTA, SPDAT
ADC Registers:	ADCON, ADT, ADCH, ADDL, ADDH
PWM Registers:	PWMxCON, PWMxP, PWMxC (x = 0 - 2)
CMP Registers:	CMPCONx (x = 0,1)

Table 7.1 CPU Core SFRs

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ACC	E0H	Accumulator	00000000	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0
B	F0H	B Register	00000000	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0
AUXC	F1H	C Register	00000000	C.7	C.6	C.5	C.4	C.3	C.2	C.1	C.0
PSW	D0H	Program Status Word	00000000	C	AC	F0	RS1	RS0	OV	F1	P
SP	81H	Stack Pointer	00000111	SP.7	SP.6	SP.5	SP.4	SP.3	SP.2	SP.1	SP.0
DPL	82H	Data Pointer Low byte	00000000	DPL0.7	DPL0.6	DPL0.5	DPL0.4	DPL0.3	DPL0.2	DPL0.1	DPL0.0
DPH	83H	Data Pointer High byte	00000000	DPH0.7	DPH0.6	DPH0.5	DPH0.4	DPH0.3	DPH0.2	DPH0.1	DPH0.0
DPL1	84H	Data Pointer 1 Low byte	00000000	DPL1.7	DPL1.6	DPL1.5	DPL1.4	DPL1.3	DPL1.2	DPL1.1	DPL1.0
DPH1	85H	Data Pointer 1 High byte	00000000	DPH1.7	DPH1.6	DPH1.5	DPH1.4	DPH1.3	DPH1.2	DPH1.1	DPH1.0
INSCON	86H	Data pointer select	----00-0	-	-	-	-	DIV	MUL	-	DPS

Table 7.2 Power and Clock control SFRs

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	87H	Power Control	000-0000	SMOD	SSTAT	SSTAT1	-	GF1	GF0	PD	IDL
SUSLO	8EH	Suspend Mode Control	00000000	SUSLO.7	SUSLO.6	SUSLO.5	SUSLO.4	SUSLO.3	SUSLO.2	SUSLO.1	SUSLO.0

Table 7.3 Data Memory SFRs

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XPAGE	F7H	Memory Page	00000000	XPAGE.7	XPAGE.6	XPAGE.5	XPAGE.4	XPAGE.3	XPAGE.2	XPAGE.1	XPAGE.0

Table 7.4 Flash/EEPROM control SFRs

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_OFF SET	FBH	Low byte offset of flash memory for programming	00000000	IB_OFF SET.7	IB_OFF SET.6	IB_OFF SET.5	IB_OFF SET.4	IB_OFF SET.3	IB_OFF SET.2	IB_OFF SET.1	IB_OFF SET.0
IB_DATA	FCH	Data Register for programming flash memory	00000000	IB_DATA.7	IB_DATA.6	IB_DATA.5	IB_DATA.4	IB_DATA.3	IB_DATA.2	IB_DATA.1	IB_DATA.0
IB_CON1	F2H	Flash Memory Control Register 1	00000000	IB_CON1.7	IB_CON1.6	IB_CON1.5	IB_CON1.4	IB_CON1.3	IB_CON1.2	IB_CON1.1	IB_CON1.0
IB_CON2	F3H	Flash Memory Control Register 2	----0000	-	-	-	-	IB_CON2.3	IB_CON2.2	IB_CON2.1	IB_CON2.0
IB_CON3	F4H	Flash Memory Control Register 3	----0000	-	-	-	-	IB_CON3.3	IB_CON3.2	IB_CON3.1	IB_CON3.0
IB_CON4	F5H	Flash Memory Control Register 4	----0000	-	-	-	-	IB_CON4.3	IB_CON4.2	IB_CON4.1	IB_CON4.0
IB_CON5	F6H	Flash Memory Control Register 5	----0000	-	-	-	-	IB_CON5.3	IB_CON5.2	IB_CON5.1	IB_CON5.0
XPAGE	F7H	Memory Page	00000000	XPAGE.7	XPAGE.6	XPAGE.5	XPAGE.4	XPAGE.3	XPAGE.2	XPAGE.1	XPAGE.0
FLASHCON	A7H	Flash access control	-------0	-	-	-	-	-	-	-	FAC

Table 7.5 WDT SFR

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RSTSTAT	B1H	Watchdog Timer Control	0-100000*	WDOF	-	PORF	LVRF	CLRF	WDT.2	WDT.1	WDT.0

***Note:** RSTSTAT initial value is determined by different RESET,refer to "Watchdog Timer (WDT) and Reset State" section for details.*

Table 7.6 CLKCON SFR

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCON	B2H	System Clock Control Register	111000--	32K _SPDUP	CLKPS1	CLKPS0	SCMIF	RCON	FS	-	-

Table 7.7 Interrupt SFRs

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN0	A8H	Interrupt Enable Control 0	00000000	EA	EADC	ET2	ES0	ET1	EX1	ET0	EX0
IEN1	A9H	Interrupt Enable Control 1	00000000	ELPD	EX4	EPWM	ESCM	ECMP1	ES1	ECMP0	ESPI
IENC	BAH	Interrupt 8channel enable control	00000000	EXS47	EXS46	EXS45	EXS44	EXS43	EXS42	EXS41	EXS40
IPH0	B4H	Interrupt Priority Control High 0	-0000000	-	PADCH	PT2H	PUH	PT1H	PX1H	PT0H	PX0H
IPL0	B8H	Interrupt Priority Control Low 0	-0000000	-	PADCL	PT2L	PUL	PT1L	PX1L	PT0L	PX0L
IPH1	B5H	Interrupt Priority Control High 1	00000000	PLPDH	PX4H	PPWMH	PSCMH	PCMP1H	PS1H	PCMP0H	PSPIH
IPL1	B9H	Interrupt Priority Control Low 1	00000000	PLPDL	PX4L	PPWML	PSCML	PCMP1L	PS1L	PCMP0L	PSPIL
EXF0	AAH	External interrupt Control 0	00------	IT4.1	IT4.0	-	-	-	-	-	-
EXF1	D8H	External interrupt Control 1	00000000	IF47	IF46	IF45	IF44	IF43	IF42	IF41	IF40

Table 7.8 Port SFRs

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0	80H	8-bit Port 0	11111111 /00000000*	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
P1	90H	8-bit Port 1	11111111 /00000000*	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
P2	A0H	8-bit Port 2	11111111 /00000000*	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
P3	B0H	8-bit Port 3	11111111 /00000000*	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
P4	C0H	8-bit Port 4	11111111 /00000000*	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0

(to be continued)

(continue)

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0M0	E9H	Port0-4 mode control register	00000000**	P0M07	P0M06	P0M05	P0M04	P0M03	P0M02	P0M01	P0M00
P0M1	E1H		00000000	P0M17	P0M16	P0M15	P0M14	P0M13	P0M12	P0M11	P0M10
P1M0	EAH		00000000**	P1M07	P1M06	P1M05	P1M04	P1M03	P1M02	P1M01	P1M00
P1M1	E2H		00000000	P1M17	P1M16	P1M15	P1M14	P1M13	P1M12	P1M11	P1M10
P2M0	EBH		00000000**	P2M07	P2M06	P2M05	P2M04	P2M03	P2M02	P2M01	P2M00
P2M1	E3H		00000000	P2M17	P2M16	P2M15	P2M14	P2M13	P2M12	P2M11	P2M10
P3M0	ECH		00000000**	P3M07	P3M06	P3M05	P3M04	P3M03	P3M02	P3M01	P3M00
P3M1	E4H		00000000	P3M17	P3M16	P3M15	P3M14	P3M13	P3M12	P3M11	P3M10
P4M0	EDH		00000000**	P4M07	P4M06	P4M05	P4M04	P4M03	P4M02	P4M01	P4M00
P4M1	E5H		00000000	P4M17	P4M16	P4M15	P4M14	P4M13	P4M12	P4M11	P4M10

Note: **means selecting quasi-bi mode by default when power-on,reset value will be 11111111; selecting input-only mode by default when power-on,reset value will be 00000000. **means selecting quasi-bi mode by default when power-on,reset value will be 00000000; selecting input-only mode by default when power-on,reset value will be 11111111.*

Table 7.9 Timer SFRs

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	88H	Timer/Counter0/1 Control	00000000	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
TMOD	89H	Timer/Counter0/1 Mode	00000000	GATE1	C/T1	M11	M10	GATE0	C/T0	M01	M00
TL0	8AH	Timer/Counter0 Low Byte	00000000	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0
TH0	8CH	Timer/Counter0 High Byte	00000000	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
TL1	8BH	Timer/Counter1 Low Byte	00000000	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.1
TH1	8DH	Timer/Counter1 High Byte	00000000	TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.1
T2CON	C8H	Timer/Counter2 Control	00000000	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/RL2
T2MOD	C9H	Timer/Counter2 Mode	------00	-	-	-	-	-	-	T2OE	DCEN
RCAP2L	CAH	Timer/Counter2 Reload /Caprure Low Byte	00000000	RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.0
RCAP2H	CBH	Timer/Counter2 Reload /Caprure High Byte	00000000	RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0
TL2	CCH	Timer/Counter2 Low Byte	00000000	TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0
TH2	CDH	Timer/Counter2 High Byte	00000000	TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0
TCON1	CEH	Timer/Counter x Control (x = 0,1)	-00-----	-	TCLKS1	TCLKS0	-	-	-	-	-

Table 7.10 EUART0 SFRs

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON	98H	Serial Control	00000000	SM0/FE	SM1/RXOV	SM2/TXCOL	REN	TB8	RB8	TI	RI
SBUF	99H	Serial Data Buffer	00000000	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0
SADEN	9BH	Slave Address Mask	00000000	SADEN.7	SADEN.6	SADEN.5	SADEN.4	SADEN.3	SADEN.2	SADEN.1	SADEN.0
SADDR	9AH	Slave Address	00000000	SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0
PCON	87H	Power & serial Control	000-0000	SMOD	SSTAT	SSTAT1	-	GF1	GF0	PD	IDL

Table 7.11 EUART1 SFRs

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON1	E8H	Serial Control	00000000	SM10 /FE1	SM11 /RXOV1	SM12 /TXCOL1	REN1	TB81	RB81	TI1	RI1
SBUF1	9DH	Serial Data Buffer	00000000	SBUF1.7	SBUF1.6	SBUF1.5	SBUF1.4	SBUF1.3	SBUF1.2	SBUF1.1	SBUF1.0
SADDR1	9EH	Slave Address	00000000	SADDR1.7	SADDR1.6	SADDR1.5	SADDR1.4	SADDR1.3	SADDR1.2	SADDR1.1	SADDR1.0
SADEN1	9FH	Slave Address Mask	00000000	SADEN1.7	SADEN1.6	SADEN1.5	SADEN1.4	SADEN1.3	SADEN1.2	SADEN1.1	SADEN1.0
PCON	87H	Power & serial Control	000-0000	SMOD	SSTAT	SSTAT1	-	GF1	GF0	PD	IDL
SBRT0	9CH	Baud-rate generator	00000000	SBRT0.7	SBRT0.6	SBRT0.5	SBRT0.4	SBRT0.3	SBRT0.2	SBRT0.1	SBRT0.0
SBRT1	A4H	Baud-rate generator	00000000	SBRTEN	SBRT1.6	SBRT1.5	SBRT1.4	SBRT1.3	SBRT1.2	SBRT1.1	SBRT1.0

Table 7.12 SPI SFRs

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPCON	A2H	SPI control register	00000000	DIR	MSTR	CPHA	CPOL	SSDIS	SPR2	SPR1	SPR0
SPSTA	F8H	SPI status register	00000---	SPEN	SPIF	MODF	WCOL	RXOV	-	-	-
SPDAT	A3H	SPI data register	00000000	SPD.7	SPD.6	SPD.5	SPD.4	SPD.3	SPD.2	SPD.1	SPD.0

Table 7.13 ADC SFRs

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON	93H	ADC Control	000-0000	ADON	ADCIF	EC	-	SCH2	SCH1	SCH0	GO/DONE
ADT	94H	ADC Time Configuration	000-0000	TADC2	TADC1	TADC0	-	TS3	TS2	TS1	TS0
ADCH	95H	ADC Channel Configuration	00000000	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
ADDL	96H	ADC Data Low Byte	------00	-	-	-	-	-	-	A1	A0
ADDH	97H	ADC Data High Byte	00000000	A9	A8	A7	A6	A5	A4	A3	A2

Table 7.14 LPD SFR

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LPDCON	B3H	LPD Control	000---00	LPDEN	LPDF	LPDV	-	-	-	LPDS1	LPDS0

Table 7.15 PWM SFRs

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMxCON	D9H -DBH	PWMx Control	0000--00	EPWMx	PWMxS	PWMxCK1	PWMxCK0	-	-	PWMxIF	PWMxSS
PWMxP	D1H -D3H	PWMx Period	00000000	PP.7	PP.6	PP.5	PP.4	PP.3	PP.2	PP.1	PP.0
PWMxD	C1H -C3H	PWMx Duty	00000000	PD.7	PD.6	PD.5	PD.4	PD.3	PD.2	PD.1	PD.0

(x = 0, 1, 2)

Table 7.16 CMP SFR

Mnem	Add	Name	POR/WDT/LVR/ PIN Reset Value	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCONx	91-92H	CMPx Control	00---000	CMPxEN	CMPxIF	-	-	-	CMPxOC	CINxV	COUTx

(x = 0, 1)

Note： *- :reserved.*

SFR Map

	Bit addressable	Non Bit addressable							
	0/8	1/9	2/A	3/B	4/C	5/D	6/E	7/F	
F8h	SPSTA			IB_OFFSET	IB_DATA				FFh
F0h	B	AUXC	IB_CON1	IB_CON2	IB_CON3	IB_CON4	IB_CON5	XPAGE	F7h
E8h	SCON1	P0M0	P1M0	P2M0	P3M0	P4M0			EFh
E0h	ACC	P0M1	P1M1	P2M1	P3M1	P4M1			E7h
D8h	EXF1	PWM0CON	PWM1CON	PWM2CON					DFh
D0h	PSW	PWM0P	PWM1P	PWM2P					D7h
C8h	T2CON	T2MOD	RCAP2L	RCAP2H	TL2	TH2	TCON1		CFh
C0h	P4	PWM0D	PWM1D	PWM2D					C7h
B8h	IPL0	IPL1	IENC						BFh
B0h	P3	RSTSTAT	CLKCON	LPDCON	IPH0	IPH1			B7h
A8h	IEN0	IEN1	EXF0						AFh
A0h	P2		SPCON	SPDAT	SBRT1			FLASHCON	A7h
98h	SCON	SBUF	SADDR	SADEN	SBRT0	SBUF1	SADDR1	SADEN1	9Fh
90h	P1	CMPCON0	CMPCON1	ADCON	ADT	ADCH	ADDL	ADDH	97h
88h	TCON	TMOD	TL0	TL1	TH0	TH1	SUSLO		8Fh
80h	P0	SP	DPL	DPH	DPL1	DPH1	INSCON	PCON	87h
	0/8	*1/9*	*2/A*	*3/B*	*4/C*	*5/D*	*6/E*	7/F	

Note: *The unused addresses of SFR are not available.*

8. Normal Function

8.1 CPU

Feature

- CPU core registers: ACC, B, PSW, SP, DPL, DPH

Accumulator

ACC is the Accumulator register. Instruction system adopts A as mnemonic symbol of accumulator.

B Register

The B register is used during multiply and divide operations. For other instructions it can be treated as another scratch pad register.

Stack Pointer (SP)

The Stack Pointer Register is 8 bits special register, It is incremented before data is stored during PUSH, CALL executions and interrupt response.And it is decremented after data is out of stack during POP, RET, RETI executions. The stack may reside anywhere in on-chip internal RAM (00H-FFH). On reset, the Stack Pointer is initialized to 07H causing the stack to begin at location 08H.

Program Status Word Register (PSW)

The PSW register contains program status information.

Table 8.1 PSW Register

D0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PSW	C	AC	F0	RS1	RS0	OV	F1	P
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**C**	**Carry flag bit** 0: no carry or borrow in an arithmetic or logic operation 1: a carry or borrow in an arithmetic or logic operation
6	**AC**	**Auxiliary Carry flag bit** 0: no auxiliary carry or borrow in an arithmetic or logic operation 1: an auxiliary carry or borrow in an arithmetic or logic operation
5	**F0**	**F0 flag bit** Available to the user for general purposes
4-3	**RS[1:0]**	**R0-R7 Register bank select bits** 00: Bank0 (Address to 00H-07H) 01: Bank1 (Address to 08H-0FH) 10: Bank2 (Address to 10H-17H) 11: Bank3 (Address to 18H-1FH)
2	**OV**	**Overflow flag bit** 0: no overflow happen 1: an overflow happen
1	**F1**	**F1 flag bit** Available to the user for general purposes
0	**P**	**Parity flag bit** 0: In the Accumulator,the bits whose value is 1 is even number 1: In the Accumulator,the bits whose value is 1 is odd number

Data Pointer Register (DPTR)

DPTR consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16-bit address, but it may be manipulated as a 16-bit register or as two independent 8-bit registers.

8.1.1 Enhanced CPU core SFRs

- Extended 'MUL' and 'DIV' instructions: 16bit*8bit, 16bit/8bit
- Dual Data Pointer
- Enhanced CPU core registers: AUXC, DPL1, DPH1, INSCON

The SH88F516 has modified 'MUL' and 'DIV' instructions. These instructions support 16 bit operand. A new register - the register AUXC is applied to hold the upper part of the operand/result.

The AUXC register is used during 16 bit operand multiply and divide operations. For other instructions it can be treated as another scratch pad register.

After reset, the CPU is in standard mode, which means that the 'MUL' and 'DIV' instructions are operating like the standard 8051 instructions. To enable the 16 bit mode operation, the corresponding enable bit in the INSCON register must be set.

	Operation		Result		
			A	B	AUXC
MUL	INSCON.2 = 0; 8 bit mode	(A)*(B)	Low Byte	High Byte	---
	INSCON.2 = 1; 16 bit mode	(AUXC A)*(B)	Low Byte	Middle Byte	High Byte
DIV	INSCON.3 = 0; 8 bit mode	(A)/(B)	Quotient Low Byte	Remainder	---
	INSCON.3 = 1; 16 bit mode	(AUXC A)/(B)	Quotient Low Byte	Remainder	Quotient High Byte

Dual Data Pointer

Using two data pointers can accelerate data memory moves. The standard data pointer is called DPTR and the new data pointer is called DPTR1.

DPTR1 is similar to DPTR, which consists of a high byte (DPH1) and a low byte (DPL1). Its intended function is to hold a 16-bit address, but it may be manipulated as a 16-bit register or as two independent 8-bit registers.

The DPS bit in INSTCON register is used to choose the active pointer by setting 1 or 0. And all DPTR-related instructions will use the currently selected data pointer.

8.1.2 Register

Table 8.2 Data Pointer Select Register

86H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
INSCON	-	-	-	-	DIV	MUL	-	DPS
R/W	-	-	-	-	R/W	R/W	-	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	0	0	-	0

Bit Number	Bit Mnemonic	Description
3	DIV	**16 bit/8 bit Divide Selection Bit** 0: 8 bit Divide 1: 16 bit Divide
2	MUL	**16 bit/8 bit Multiply Selection Bit** 0: 8 bit Multiply 1: 16 bit Multiply
0	DPS	**Data Pointer Selection Bit** 0: Data pointer 1: Data pointer1

8.2 RAM

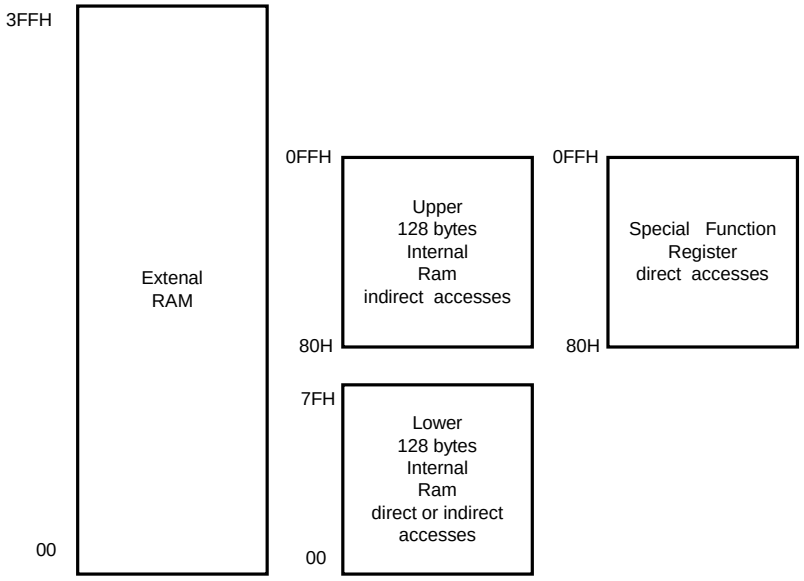
8.2.1 Feature

SH88F516 provides both internal RAM and external RAM for random data storage. The internal data memory is mapped into four separated segments:

- The Lower 128 bytes of RAM (addresses 00H to 7FH) are directly and indirectly addressable
- The Upper 128 bytes of RAM (addresses 80H to FFH) are indirectly addressable only
- The Special Function Registers (SFR, addresses 80H to FFH) are directly addressable only
- The 1024 bytes of external RAM(addresses 00H to 3FFH) are indirectly accessed by MOVX instructions

The Upper 128 bytes 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 addresses of SFR are not available.*



The Internal and External RAM Configuration

The SH88F516 provides traditional method for accessing of external RAM. Use *MOVXA, @Ri* or *MOVX @Ri, A;* to access external low 256 bytes RAM; *MOVX A, @DPTR* or *MOVX @DPTR, A* also to access external 1024 bytes RAM.

In SH88F516, the user can also use XPAGE register to access external RAM only with *MOVX A, @Ri* or *MOVX @Ri, A* instructions. The user can use XPAGE to represent the high byte address of RAM above 256 Bytes.

In Flash SSP mode, the XPAGE can also be used as sector selector (Refer to SSP Function).

8.2.2 Register

Table 8.3 Data Memory Page Register

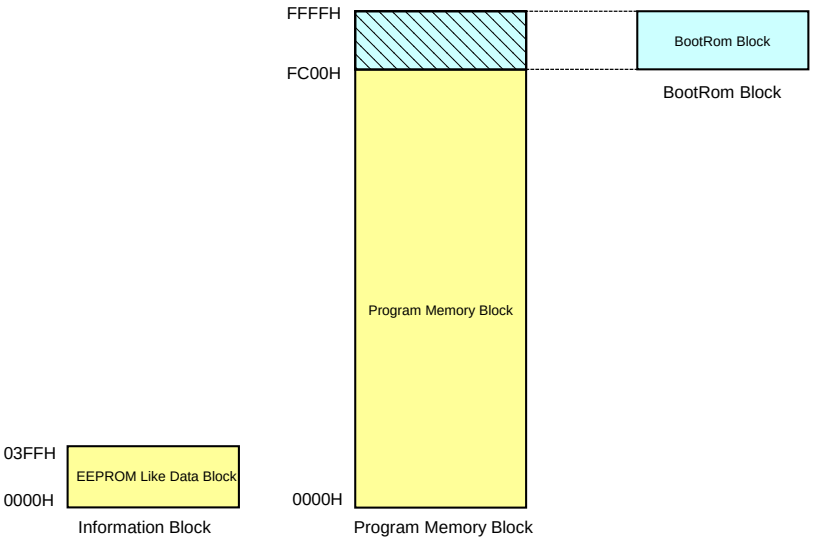
F7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XPAGE	XPAGE.7	XPAGE.6	XPAGE.5	XPAGE.4	XPAGE.3	XPAGE.2	XPAGE.1	XPAGE.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**XPAGE[7:0]**	**RAM Page Selector**

8.3 Flash Program Memory

8.3.1 Feature

- The program memory consists 64 X 1KB sectors, total 64KB
- Programming and erase can be done over the full operation voltage range
- Write, read and erase operation are all supported by In-Circuit Programming (ICP)
- Fast mass/sector erase and programming
- Minimum program/erase cycles: 100,000
- Minimum years data retention: 10
- Low power consumption



The SH88F516 embeds 64K flash program memory for program code. The flash program memory supports In-Circuit Programming (ICP) mode and Self-Sector Programming (SSP) mode, or operating Flash memory block In-System Programming.

Every sector is 1024 bytes.

The SH88F516 also embeds 1024 bytes EEPROM-like memory block for storing user data. Every sector is 256 bytes.It has 4 sectors.

The SH88F516 also embeds 1K bytes BootRom Block for In-System Programming function.

Flash operation definition:

In-Circuit Programming (ICP): Through the Flash programmer to wipe the Flash memory, read and write operations.

Self-Sector Programming (SSP): User Program code run in Program Memory to wipe the Flash memory, read and write operations.

In-System Programming(ISP): Program code run in BootRom to wipe the Flash memory, read and write operations.At present, the program in BootRom has been cured in it before leaving factory.Users can download user program to chips by UART port with corresponding PC software.

If the customer choose to use ISP function (Code Option OP_ISPEN will be set as 1, refer to “Code Option” for details), then the last sector address (0xFC00 - 0xFFFF) will be mapped as BootRom address, it can not be used as program memory block; If the customer choose to close ISP function (Code Option OP_ISPEN will be cleared, refer to “Code Option” for details), then the last sector (0xFC00 - 0xFFFF) can be used as program memory block.

Flash Memory Supports the Following Operations:

(1) Code Protection Control Mode

SH88F516 code protection function provides a high-performance security measures for the user. Each partition has two modes are available.

Code protection mode 0: allow/forbid any programmer write/read operations (not including overall erasure).

Code protection mode 1: allow/forbid through MOVC instructions to read operation in other sectors, or through SSP mode to erased/write operation.

The user must use one of the following two ways to complete code protection control mode Settings:

1. Flash programmer in ICP mode is set to corresponding protection bit to enter the protected mode.
2. The SSP mode does not support code protection control mode programming.

(2) Overall Erasure

Regardless of the state of the code protection control mode, the overall erasure operation will erase all programs, code options, the code protection bit, but they will not erase EEPROM-like memory block.

The user must use the following way to complete the overall erasure:

1. Flash programmer in ICP mode send overall erasure instruction to run overall erasure.
2. The SSP mode does not support overall erasure mode.

Note: *When SH88F516 enables ISP function (Code Option OP_ISPEN is set as 1, refer to "Code Option" for details), the last sector (sector 63) can not be erased by overall erase.*

(3) Sector Erasure

Sector erasure operations will erase the content of selected sector. The user program (SSP) and Flash programmer can perform this operation.

For user programs to perform the operation, code protection mode 1 in the selected sector must be forbidden.

For Flash programmer to perform the operation, code protection mode 0 in the selected sector must be forbidden.

The user must use one of the following two ways to complete sector erasure:

1. Flash programmer in ICP mode send sector erasure instruction to run sector erasure.
2. Through the SSP function send sector erasure instruction to run sector erasure (see chapter SSP)

Note: *When SH88F516 enables ISP function (Code Option OP_ISPEN is set as 1, refer to "Code Option" for details), the last sector (sector 63) can not be erased by overall erase.*

(4) EEPROM-like Memory Block Erasure

EEPROM-like memory block erasure operations will erase the content in EEPROM-like memory block.The user program (SSP) and Flash programmer can perform this operation.

The user must use one of the following two ways to complete EEPROM-like memory block erasure:

1. Flash programmer in ICP mode send EEPROM-like memory block erasure instruction to run EEPROM-like memory block erasure.
2. Through the SSP function send EEPROM-like memory block erasure instruction to run EEPROM-like memory block erasure (see chapter SSP).

(5) Write/Read Code

Write/read code operation can read or write code from flash memory block.The user program (SSP) and Flash programmer can perform this operation.

For user programs to perform the operation, code protection mode 1 in the selected sector must be forbidden. Regardless of the security bit Settings or not, the user program can read/write the sector which contains program itself.

For Flash programmer to perform the operation,code protection mode 0 in the selected sector must be forbidden.

The user must use one of the following two ways to complete write/read code:

1. Flash programmer in ICP mode send write/read code instruction to run write/read code.
2. Through the SSP function send write/read code instruction to run write/read code.

(6) Write/Read EEPROM-like Memory Block

EEPROM-like memory block operation can read or write data from EEPROM-like memory block.The user program (SSP) and Flash programmer can perform this operation.

The user must use one of the following two ways to complete write/read EEPROM-like memory block:

1. Flash programmer in ICP mode send write/read EEPROM-like memory block instruction to run write/read EEPROM-like memory block.
2. Through the SSP function send write/read EEPROM-like memory block instruction to run write/read EEPROM-like memory block.

Flash Memory Block Operation Summary

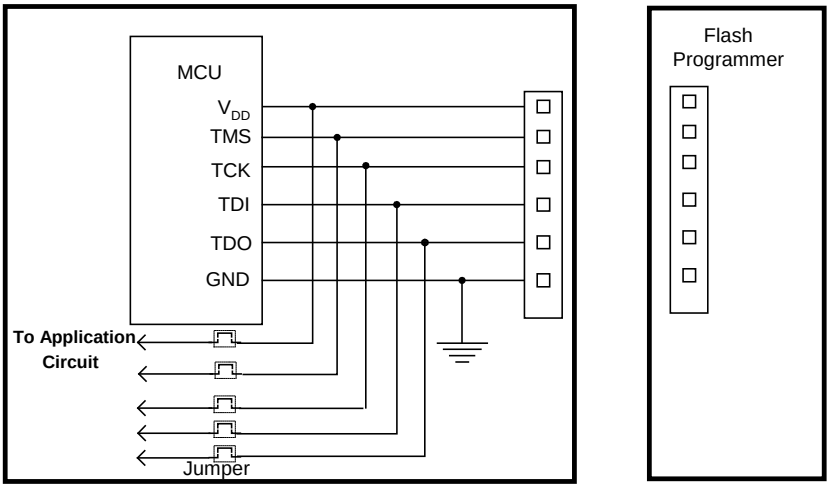
Operation	ICP	SSP	ISP
Code protection	support	Non support	support
Sector erasure	Support (no security bit)	Support (no security bit)	Support (no security bit)
Overall erasure	support	Non support	support
EEPROM-like memory block erasure	support	support	support
Write/read code	Support (no security bit)	Support (no security bit)	Support (no security bit)
Read/write EEPROM-like memory block	support	support	support

8.3.2 Flash Operation in ICP 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 programmer can refresh the program memory through ICP programming interface. The ICP programming interface consists of 6 pins (V_{DD}, GND, TCK, TDI, TMS, TDO).

At first the four JTAG pins (TDO, TDI, TCK, TMS) are used to enter the programming mode. Only after the four pins are inputted the specified waveform, the CPU will enter the programming mode. For more detail description please refers to the FLASH Programmer's user guide.

In ICP mode，all the flash operations are completed by the programmer through 6-wire interface. Since the program signal is very sensitive, 6 jumpers are needed (V_{DD}, GND, TDO, TDI, TCK, TMS) to separate the program pins from the application circuit, as show in the following diagram.



When using ICP mode to do operations, the recommended steps are as following:

(1) The jumpers must be open to separate the programming pins from the application circuit before programming.

(2) Connect the programming interface with programmer and begin programming.

(3) Disconnect programmer interface and connect jumpers to recover application circuit after programming is complete.

8.3.3 Using ISP Mode to Download Programs

SH88F516 has 1k BootRom Block. The program in BootRom has been cured in it before leaving factory.Users can download user program to chips by UART port with corresponding PC software. When using this function, users only need to connect TXD & RXD of UART port with the corresponding pin of the programmer. If users choose "Enter ISP mode only when P1.0 and P1.1 are connected to GND, simultaneously. (Default)", then they must make sure P1.0 and P1.1 are both GND. After that, the program can be downloaded correctly. See upper computer software instructions.

8.4 SSP Function

The SH88F516 provides SSP (Self Sector Programming) function, each sector can be sector erased or programmed by the user's code if the selected sector is not be protected. But once sector has been programmed, it cannot be reprogrammed before sector erase.

The SH88F516 builds in a complex control flow to prevent the code from carelessly modification. If the dedicated conditions are not met (IB_CON2-5), the SSP will be terminated.

8.4.1 SSP Register

Table 8.4 Memory Page Register for Programming

F7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
XPAGE	XPAGE.7	XPAGE.6	XPAGE.5	XPAGE.4	XPAGE.3	XPAGE.2	XPAGE.1	XPAGE.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

- For program memory block, a sector is 1024 bytes, registers are defined as follows:

Bit Number	Bit Mnemonic	Description
7-2	**XPAGE[7:2]**	Sector of the flash memory to be programmed, 000000 means sector 0, and so on
1-0	**XPAGE[1:0]**	High 2 Address of the flash memory sector to be programmed

- For EEPROM-like memory block, a sector is 256 bytes, registers are defined as follows:

Bit Number	Bit Mnemonic	Description
7-2	**XPAGE[7:2]**	Reserved
1-0	**XPAGE[1:0]**	For EEPROM-like sector, 00 means sector 0, and so on

Note:

For program memory block, a sector is 1024 bytes;

For EEPROM-like memory block, a sector is 256 bytes.

Table 8.5 Offset of Flash Memory for Programming

FBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_OFFSET	IB_OFFSET.7	IB_OFFSET.6	IB_OFFSET.5	IB_OFFSET.4	IB_OFFSET.3	IB_OFFSET.2	IB_OFFSET.1	IB_OFFSET.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

- For program memory block, a sector is 1024 bytes, registers are defined as follows:

Bit Number	Bit Mnemonic	Description
7-0	**IB_OFFSET[7:0]**	Low 8 Address of the flash memory sector to be programmed

- For EEPROM-like memory block, a sector is 256 bytes, registers are defined as follows:

Bit Number	Bit Mnemonic	Description
7-0	**IB_OFFSET[7:0]**	Address of the flash memory sector to be programmed

Table 8.6 Data Register for Programming

FCH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_DATA	IB_DATA.7	IB_DATA.6	IB_DATA.5	IB_DATA.4	IB_DATA.3	IB_DATA.2	IB_DATA.1	IB_DATA.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**IB_DATA[7:0]**	Data to be programmed

Table 8.7 SSP Type select Register

F2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CON1	IB_CON1.7	IB_CON1.6	IB_CON1.5	IB_CON1.4	IB_CON1.3	IB_CON1.2	IB_CON1.1	IB_CON1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**IB_CON1[7:0]**	**SSP Type select** 0xE6: Sector Erase 0x6E: Sector Programming

Table 8.8 SSP Flow Control Register1

F3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CON2	-	-	-	-	IB_CON2.3	IB_CON2.2	IB_CON2.1	IB_CON2.0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
3-0	**IB_CON2[3:0]**	Must be 05H, otherwise Flash Programming will terminate

Table 8.9 SSP Flow Control Register2

F4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CON3	-	-	-	-	IB_CON3.3	IB_CON3.2	IB_CON3.1	IB_CON3.0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
3-0	**IB_CON3[3:0]**	Must be 0AH, otherwise Flash Programming will terminate

Table 8.10 SSP Flow Control Register3

F5H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CON4	-	-	-	-	IB_CON4.3	IB_CON4.2	IB_CON4.1	IB_CON4.0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

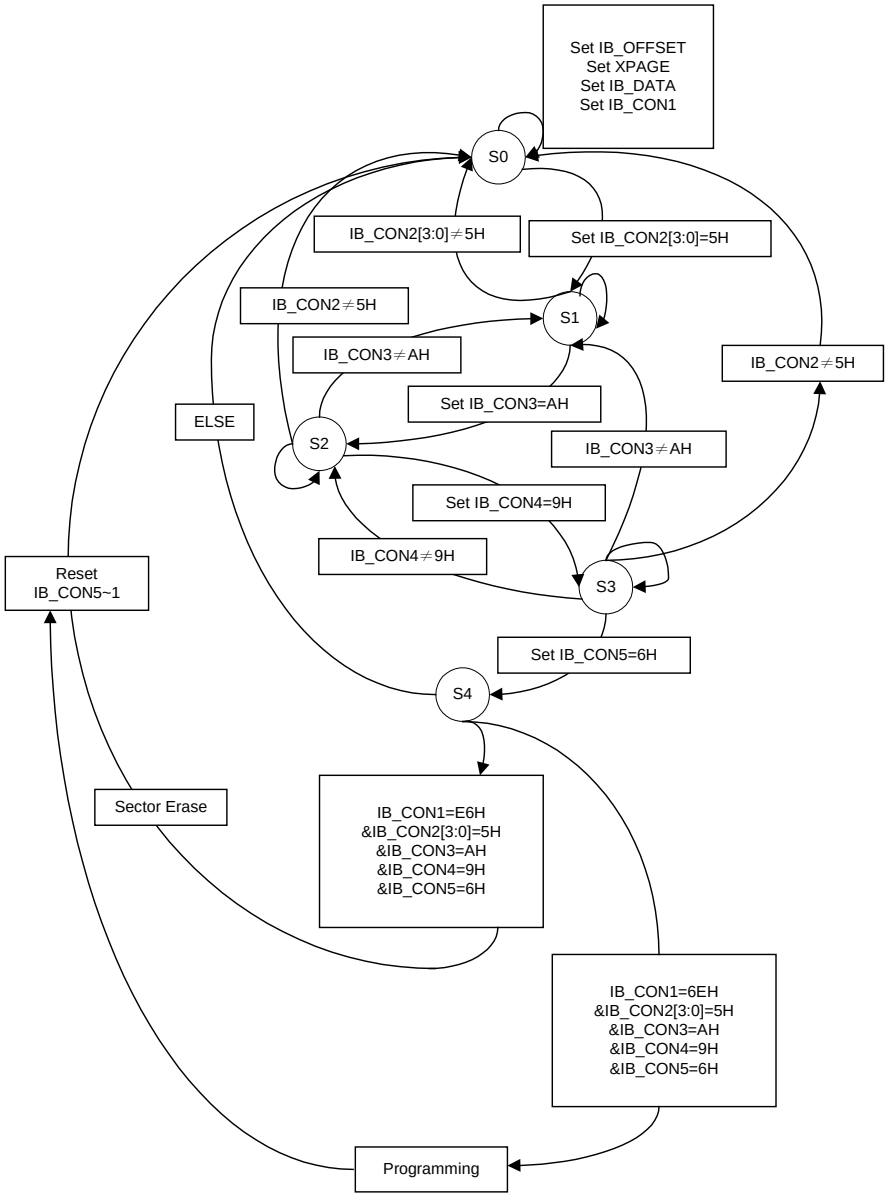
Bit Number	Bit Mnemonic	Description
3-0	**IB_CON4[3:0]**	Must be 09H, otherwise Flash Programming will terminate

Table 8.11 SSP Flow Control Register4

F6H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IB_CON5	-	-	-	-	IB_CON5.3	IB_CON5.2	IB_CON5.1	IB_CON5.0
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
3-0	**IB_CON5[3:0]**	Must be 06H, otherwise Flash Programming will terminate

8.4.2 Flash Control Flow



8.4.3 SSP Programming Notice

To successfully complete SSP programming, the user's software must be set as the following the steps:

(1) For Code/Data Programming:

1. Disable interrupt;
2. Fill in the XPAGE, IB_OFFSET for the corresponding address;
3. Fill in IB_DATA, if programming is wanted;
4. Fill in IB_CON1-5 sequentially;
5. Add 4 nops for more stable operation;
6. Code/Data programming, CPU will be in IDLE mode;
7. Go to Step 2, if more data are to be programmed;
8. Clear XPAGE; enable interrupt if necessary.

(2) For Sector Erase:

1. Disable interrupt;
2. Fill in the XPAGE for the corresponding sector;
3. Fill in IB_CON1-5 sequentially;
4. Add 4 NOPs for more stable operation;
5. Sector Erase, CPU will be in IDLE mode;
6. Go to step 2, if more sectors are to be erased;
7. Clear XPAGE; enable interrupt if necessary.

(3) For Code Reading:

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

(4) For EEPROM-Like:

SH88F516 has 1K(0000H-03FFH) EEPROM-Like. Steps is same as code programming,the diffenrences are:

1. Set FAC bit in FLASHCON register before programming or erase EEPROM-Like;
2. One sector of EEPROM-Like is 256 bytes.not 1024 bytes.

Note: *FAC must be cleared when you don't need to do EEPROM-like operation.*

8.4.4 Readable Random Code

Every chip is cured an 8-bit readable random code after production. Readable random code is 0-255 random value,and can not be erased, read by program or tools.

How to read random code: set FAC bit, Assigned to the DPTR as "0A7FH", clear A, then use "MOVC A, @A+DPTR" to read.

Note: *It is needed to clear FAC after reading readable random code, otherwise it will influence on the instructions execution of reading program ROM.*

FLASHCON register description is as follows:

Table 8.12 Flash Access Control Register

A7H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
FLASHCON	-	-	-	-	-	-	-	FAC
R/W	-	-	-	-	-	-	-	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	-	-	-	0

Bit Number	Bit Mnemonic	Description
7-1	**-**	**Reserved**
0	**FAC**	**FAC: Flash access control** 0: MOVC or SSP access Main Block 1: MOVC or SSP access EEPROM-like

8.5 System Clock and Oscillator

8.5.1 Feature

- 5 oscillator types: 32.768kHz crystal, crystal oscillator, ceramic oscillator, external clock and 16.6MHz internal RC
- Built-in 16.6MHz internal RC
- Built-in 32.768kHz speed up circuit
- Built-in system clock prescaler

8.5.2 Clock Definition

SH88F516 have several internal clocks defined as below:

OSCCLK: the oscillator clock is selected from the five oscillator types (32.768kHz crystal, crystal oscillator, ceramic oscillator, external clock and 16.6MHz internal RC). f_{OSC} is defined as the OSCCLK frequency. t_{OSC} is defined as the OSCCLK period.

WDTCLK: the internal WDT RC clock. f_{WDT} is defined as the WDTCLK frequency. t_{WDT} is defined as the WDTCLK period.

SYSCLK: system clock, the output clock of system clock prescaler. It is the CPU instruction clock. f_{SYS} is defined as the SYSCLK frequency. t_{SYS} is defined as the SYSCLK period.

8.5.3 Description

SH88F516 has 5 oscillator types: 32.768kHz crystal, crystal oscillator (30kHz-16.6MHz), ceramic Oscillator (30kHz-16.6MHz), external clock (30kHz-16.6MHz) and internal RC (16.6MHz), which is selected by code option OP_OSC (Refer to code option section for details). The oscillator generates the basic clock pulse that provides the system clock to supply CPU and on-chip peripherals.

8.5.4 Register

Table 8.13 System Clock Control Register

B2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCON	32K_SPDUP	CLKS1	CLKS0	SCMIF	RCON*	FS*	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	-
Reset Value (POR/WDT/LVR/PIN)	1	1	1	0	0	0	-	-

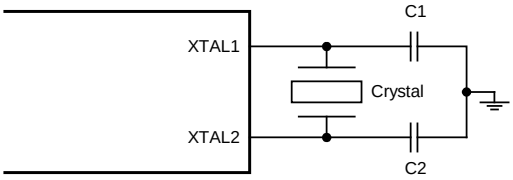
Bit Number	Bit Mnemonic	Description
7	**32K_SPDUP**	**32.768kHz oscillator speed up mode control bit** 0: 32.768kHz oscillator normal mode, cleared by software. 1: 32.768kHz oscillator speed up mode, set by hardware or software. This control bit is set by hardware automatically in all kinds of RESET such as Power on reset, watch dog reset etc. to speed up the 32.768kHz Oscillator oscillating, shorten the 32.768kHz oscillator start-oscillating time. And this bit also can be set or cleared by software if necessary. Such as set before entering Power-down mode and cleared when Power-down mode wakes up. It should be noticed that turning off 32.768kHz oscillator speed up (clear this bit) could reduce the system power consumption. Only when code option OP_OSC is 011, this bit is valid. (32.768kHz oscillator is selected, Refer to **code option** section for details)
6-5	**CLKS[1:0]**	**SYSCLK Prescaler Register** 00: $f_{SYS} = f_{OSCS}$ 01: $f_{SYS} = f_{OSCS}/2$ 10: $f_{SYS} = f_{OSCS}/4$ 11: $f_{SYS} = f_{OSCS}/12$ If 32.768kHz oscillator is selected as OSCSCLK, these control bits is invalid.
4	**SCMIF**	**System Clock Monitor flag bit** 0: Clear by hardware to indicate system clock is normal 1: Set by hardware to indicate system clock fails
3	**RCON**	**Internal RC On control Register** 0: Turn off Internal RC (Default) 1: Turn on Internal RC Only when code option OP_OSC is 011, this bit is valid.
2	**FS**	**Frequency Select Register** 0: 32.768kHz is selected as OSCSCLK 1: Internal RC is selected as OSCSCLK Only when code option OP_OSC is 011, this bit is valid.

Note:

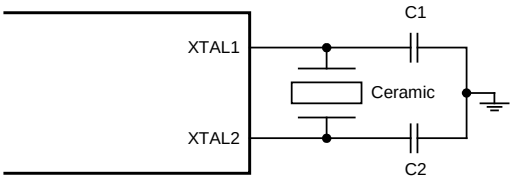
1. RCON and FS is valid only when code option OP_OSC[2:0] is 011;
2. When Internal RC is used as OSCSCLK (that is RCON = 1 and FS = 1), RCON is can't be cleared by software;
3. When OSCSCLK changed from 32.768kHz to Internal RC, the steps below must be done in sequence:
a. Set RCON = 1 to turn on the Internal RC;
b. Wait at least 2 Oscillator period;
c. Set FS = 1 to select SYSCLK as Internal RC.

8.5.5 Oscillator Type

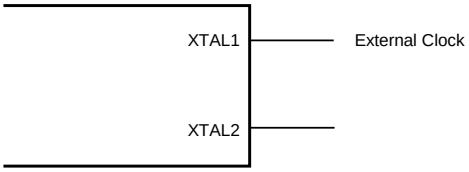
(1) Crystal Oscillator: 32.768kHz or 400kHz - 16.6MHz



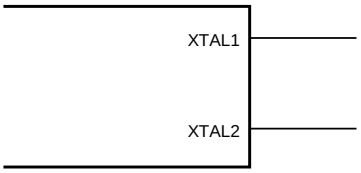
(2) Ceramic Oscillator: 400kHz - 16.6MHz



(3) External clock: 30kHz - 16.6MHz



(4) Internal RC: 16.6MHz



8.5.6 Capacitor Selection for Oscillator

Ceramic Oscillator		
Frequency	**C1**	**C2**
455kHz	47-100pF	47-100pF
3.58MHz	-	-
4MHz	-	-

Crystal Oscillator		
Frequency	**C1**	**C2**
32.768kHz	5-12.5pF	5-12.5pF
8MHz	8-15pF	8-15pF
16MHz	8-15pF	8-15pF

-: Load capacitor has been built in it.

Note:

(1) ***Capacitor values are used for design guidance only!***

(2) These capacitors were tested with the crystals listed above for basic start-up and operation. They are ***not optimized.***

(3) Be careful for the stray capacitance on PCB board, the user should test the performance of the oscillator over the expected V_{DD} and the temperature range for the application.

Before selecting crystal/ceramic, the user should consult the crystal/ceramic manufacturer for appropriate value of external component to get best performance, visit http://www.sinowealth.com for more recommended manufactures.

8.6 System Clock Monitor (SCM)

In order to enhance the system reliability, SH88F516 contains a system clock monitor (SCM) module. If the system clock fails (for example external oscillator stops oscillating), the built-in SCM will switch the OSCCLK to the internal WDTCLK and set system clock monitor bit (SCMIF) to 1. And the SCM interrupt will be generated when EA and ESCM is enabled. If the OSCCLK comes back, SCM will switch the OSCCLK back to external oscillator and clears the SCMIF automatically.

Notes:

The SCMIF is read-only register; it can be clear to 0 or set to 1 by hardware only.

If SCMIF is cleared, the SCM switches the system clock to the state before system clock fail automatically.

*If Internal RC is selected as OSCCLK by code option (Refer to **code option** section for detail), the SCM can not work.*

Table 8.14 System Clock Control Register

B2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CLKCON	32K_SPDUP	CLKS1	CLKS0	SCMIF	RCON*	FS*	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	-	-
Reset Value (POR/WDT/LVR/PIN)	1	1	1	0	0	0	-	-

Bit Number	Bit Mnemonic	Description
4	**SCMIF**	**System Clock Monitor bit** 0: Clear by hardware to indicate system clock is normal 1: Set by hardware to indicate system clock fails

8.7 I/O Port

8.7.1 Feature

- 40 bi-directional I/O ports
- Four selectable I/O mode
- Share with alternative functions

The SH88F516 has 40 bi-directional I/O ports. All I/O can be set as one of 4 modes by PxMy register: Quasi-Bi mode (Traditional 8051 mode), Push-Pull mode, Input-Only mode and Open-Drain output mode.

I/O reset status can be set by code option as Quasi-Bi mode or Input-Only mode.

In order to improve EMC capability, every input pin has a Schmitt Trigger. Even enter Power-down mode, Schmitt Trigger is never off.

For SH88F516, some I/O pins can share with alternative functions. There exists a priority rule in CPU to avoid these functions be conflict when all the functions are enabled. (Refer to **Port Share** Section for details). Only when the other function is turned off, it allows setting the corresponding register to change the I/O mode.

Note: *When I/O works in other functions, Writing PxMy register will not change the value of PxMy register and I/O mode.*

8.7.2 Register

Table 8.15 Port Control Register

E1H-E5H, E9H-EDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0M0 (E9H)	P0M07	P0M06	P0M05	P0M04	P0M03	P0M02	P0M01	P0M00
P0M1 (E1H)	P0M17	P0M16	P0M15	P0M14	P0M13	P0M12	P0M11	P0M10
P1M0 (EAH)	P1M07	P1M06	P1M05	P1M04	P1M03	P1M02	P1M01	P1M00
P1M1 (E2H)	P1M17	P1M16	P1M15	P1M14	P1M13	P1M12	P1M11	P1M10
P2M0 (EBH)	P2M07	P2M06	P2M05	P2M04	P2M03	P2M02	P2M01	P2M00
P2M1 (E3H)	P2M17	P2M16	P2M15	P2M14	P2M13	P2M12	P2M11	P2M10
P3M0 (ECH)	P3M07	P3M06	P3M05	P3M04	P3M03	P3M02	P3M01	P3M00
P3M1 (E4H)	P3M17	P3M16	P3M15	P3M14	P3M13	P3M12	P3M11	P3M10
P4M0 (EDH)	P4M07	P4M06	P4M05	P4M04	P4M03	P4M02	P4M01	P4M00
P4M1 (E5H)	P4M17	P4M16	P4M15	P4M14	P4M13	P4M12	P4M11	P4M10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	*	*	*	*	*	*	*	*

*: I/O reset status can be set by code option as Quasi-Bi mode or Input-Only mode (high impedance).

I/O Mode

PxM0n	PxM1n	Description
0	0	Quasi-Bi mode
0	1	Push-Pull mode
1	0	Input-Only mode (high impedance)
1	1	Open-Drain output mode

(x = 0, 1, 2 ,3 or 4; n = 7, 6, 5, 4, 3, 2, 1 or 0)

Table 8.16 Port Data Register

80H,90H,A0H,B0H,C0H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
P0 (80H)	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
P1 (90H)	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
P2 (A0H)	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
P3 (B0H)	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
P4 (C0H)	P4.7	P4.6	P4.5	P4.4	P4.3	P4.2	P4.1	P4.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN) *	*	*	*	*	*	*	*	*

**: I/O data reset status can be set by code option, if as Quasi-Bi mode I/O data reset value is 0FFH or as Input-Only mode. I/O data reset value is 00H.*

Bit Number	Bit Mnemonic	Description
7-0	**Px.y** **x = 0-4, y = 0-7**	Port Data Register

Note: *All can be configured as N-channel open drain I/O, but voltage provided for this pin can't exceed V_{DD} + 0.3V.*

8.7.3 Port Structure

Quasi-Bi Mode

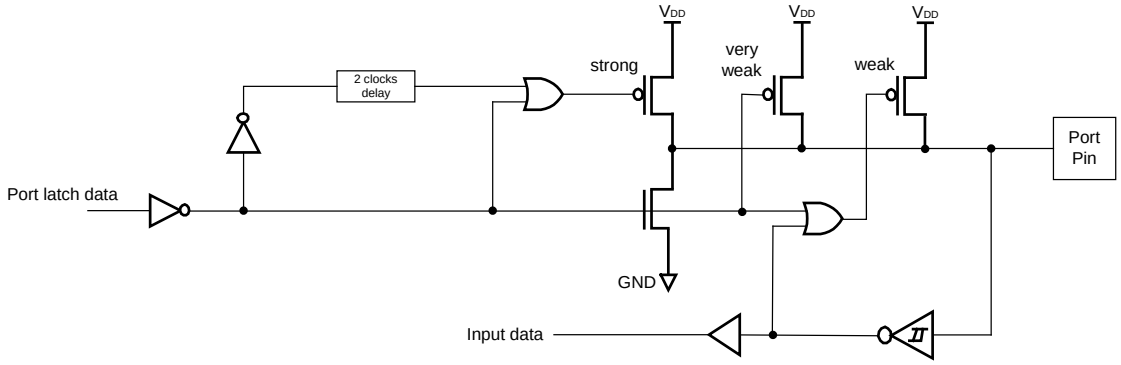
Quasi-Bi I/O has 3 pull-up MOS to adapt to different needs: weak pull-up,very weak pull-up and strong pull-up.

Weak pull-up MOS: When Data register and pin are set 1, this pull-up provides the basic drive current that quasi-bidirectional ports output high. External circuit pull the output-high pin to low, weak pull-up will be off and very weak pull-up will keep on. In order to pull this pin low intensity, external circuit must have sufficient sink current capability to drop the voltage of port below the threshold voltage.

Very weak pull-up MOS: Provide weak pull-up current to pull the pin high when port latch is 1 and the port is floating.

Strong pull-up Mos: When the port latch transition from 0 to 1, strong pull-up is used to speed up the quasi-bi port conversion from logic 0 to logic 1 in almost 2 machine cycles.

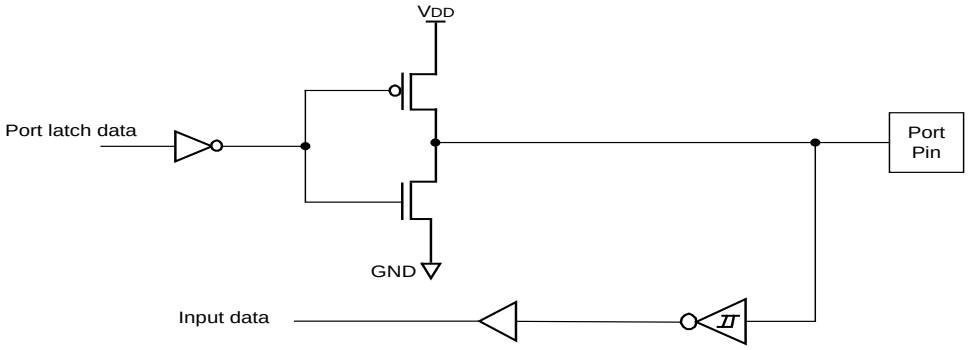
Quasi-bi model port structure diagram is shown below.



Quasi-Bi Mode

Push-Pull Mode

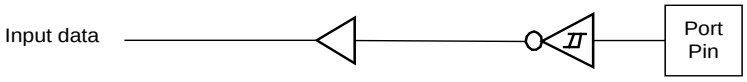
The pull-low structure in push-pull mode is same as open-drain and Quasi-Bi mode, but the port provides a continuous strong pull-up when the port latch is 1. Push-Pull mode port structure diagram is shown below:



Push-Pull Mode

Input-Only Mode

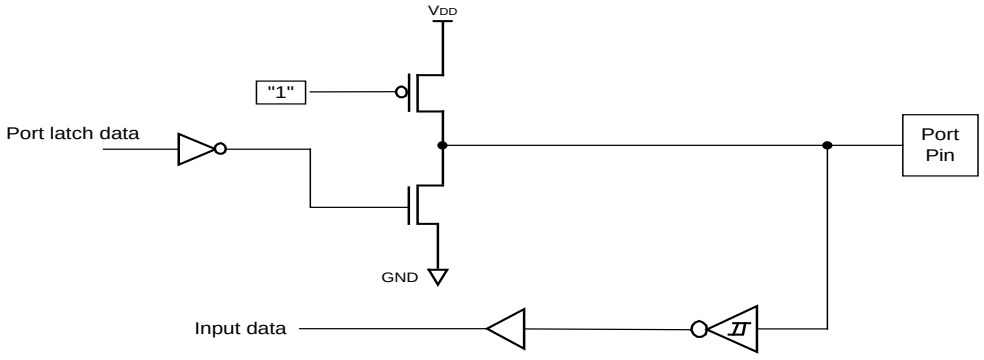
In Input-Only mode port is input only, no output capability. Input-Only mode port structure diagram is shown below:



Input-Only Mode

Open-Drain Mode

In Open-Drain mode the ports have no output high capability. The users should use pull-up resistor to output high. voltage provided for this pin can't exceed V_{DD} + 0.3V. Open-Drain mode port structure diagram is shown below:



Open-Drain Mode

8.7.4 Port Share

The 40 bi-directional I/O ports can also share second or third special function. But the share priority should obey the **Outer Most Inner Lest** rule:

The out most pin function in **Pin Configuration** has the highest priority, and the inner most pin function has the lowest priority. This means when one pin is occupied by a higher priority function (if enabled), it cannot be used as the lower priority functional pin , even the lower priority function is also enabled. Only until the higher priority function is closed by hardware or software, can the corresponding pin be released for the lower priority function use.

When port share function is enabled, any read or write operation to port will only affect the data register while the port pin keeps unchanged until all the share functions are disabled.

The 40 bi-directional I/O ports can provides some special functions:

PORT0:

- AN0 - AN7 (P0.0 - P0.7): ADC input channel
- Vref (P0.0): ADC external reference voltage

Table 8.17 PORT0 Share Table

Pin No.			Priority	Function	Enable bit
TQFP48	QFP44	LQFP44			
40	37	37	1	Vref	Set REFC bit in ADCON register
			2	AN0	Set ADCH.0 bit in ADCH register and SCH[2:0] = 000 in ADCON register
			3	P0.0	Clear REFC bit in ADCON register and ADCH.0 bit in ADCH register
39-33	36-30	36-30	1	AN1-7	Set ADCH.1-7 bits in ADCH register and SCH[2:0] = 001-111 in ADCON register
			2	P0.1-0.7	Clear ADCH.1 - 7 bits in ADCH register

PORT1:

- T2 (P1.0): Timer2 external input
- T2EX (P1.1): Timer2 capture function external input
- RXD1 (P1.2): EUART1 input
- TXD1 (P1.3): EUART1 output
- $\overline{SS}$ (P1.4): SPI slave select
- MOSI (P1.5): Master Output Slave Input
- MISO (P1.6): Master Input Slave Output
- SCK (P1.7): SPI clock
- CMP0P (P1.2): CMP0 positive input
- CMP0N (P1.3): CMP0 negative input
- CMP0O (P1.4): CMP0 output

Table 8.18 PORT1 Share Table

Pin No.			Priority	Function	Enable bit
TQFP48	QFP44	LQFP44			
43	40	40	1	T2	Set TR2 bit in T2CON register and set C/$\overline{T2}$ bit in T2MOD register
			2	P1.0	Clear TR2 bit in T2CON register and C/$\overline{T2}$ bit in T2MOD register
44	41	41	1	T2EX	Set TR2 bit in T2CON register, set C/$\overline{T2}$ bit in T2MOD register, EXEN2 = 1
			2	P1.1	Clear TR2 bit in T2CON register, set C/$\overline{T2}$ bit in T2MOD register, EXEN2 = 0
45	42	42	1	CMP0P	Set CMPEN bit in CMP0CON
			2	RXD1	Set REN1 bit in SCON1 register
			3	P1.2	Clear CMPEN bit in CMP0CON and REN1 bit in SCON1 register

(to be continued)

(continue)

Pin No.			Priority	Function	Enable bit
TQFP48	QFP44	LQFP44			
46	43	43	1	CMP0N	Set CMPEN bit in CMP0CON
			2	TXD1	When writing SBUF1 register
			3	P1.3	Clear CMPEN bit in CMP0CON and not write SBUF1 register
47	44	44	1	CMP0O	Set CMPEN bit in CMP0CON and set CMPOC bit
			2	$\overline{SS}$	Set SPEN bit in SPSTA register (When in slave mode, SPEN = 1, CPHA = 1 and SSDIS = 1, internal pull-up open automatically)
			3	P1.4	Clear CMPEN bit or CMPOC bit in CMP0CON, and clear SPEN bit in SPSTA register
1	1	1	1	MOSI	Set SPEN bit in SPSTA register (When in slave mode, SPEN = 1, CPHA = 1 and SSDIS = 1, internal pull-up open automatically)
			2	P1.5	Clear SPEN bit in SPSTA register
2	2	2	1	MISO	Set SPEN bit in SPSTA register (When in master mode, internal pull-up open automatically)
			2	P1.6	Clear SPEN bit in SPSTA register
3	3	3	1	SCK	Set SPEN bit in SPSTA register (When in slave mode, SPEN = 1, CPHA = 1 and SSDIS = 1, internal pull-up open automatically)
			2	P1.7	Clear SPEN bit in SPSTA register

PORT2:
- INT40 - INT47 (P2.0 - P2.7): external interrupt 4 input
- PWM1/2 (P2.6/P2.7): PWM1/2 output

Table 8.19 PORT2 Share Table

Pin No.			Priority	Function	Enable bit
TQFP48	QFP44	LQFP44			
21-26	18-23	18-23	1	INT40-45	Set EX4 bit in IEN1 and set EXS40 - 45 bit in IENC
			2	P2.0-2.5	Clear EX4 bit in IEN1 or clear EXS40 - 45 corresponding bit in IENC
27-28	24-25	24-25	1	PWM1/2	Set EPWM bit and PWMSS bit in PWM1/2CON
			2	INT46-47	Set EX4 bit in IEN1 and set EXS46-47 corresponding bit in IENC
			3	P2.6-2.7	Clear EPWM bit and PWMSS bit in PWM1/2CON, and Clear EX4 bit in IEN1 or clear EXS46 - 47 corresponding bit in IENC

PORT3:
- RXD0 (P3.0): EUART0 input
- TXD0 (P3.1): EUART0 output
- INT0 (P3.2): external interrupt 0
- INT1 (P3.3): external interrupt 1
- T0 (P3.4): Timer0 external input
- T1 (P3.5): Timer1 external input
- CMP1P (P3.6): CMP1 positive input
- CMP1N (P3.7): CMP1 negative input
- CMP1O (P3.5): CMP1 output

Table 8.20 PORT3 Share Table

Pin No.			Priority	Function	Enable Bit
TQFP48	QFP44	LQFP44			
5	5	5	1	RXD0	Set REN0 bit in SCON0 register
			2	P3.0	Clear REN0 bit in SCON0 register
7	7	7	1	TXD0	When writing SBUF0 register
			2	P3.1	When SBUF0 register is not written
8	8	8	1	INT0	Set EX0 bit in IEN0 register
			2	P3.2	Clear EX0 bit in IEN0 register
9	9	9	1	INT1	Set EX1 bit in IEN0 register
			2	P3.3	Clear EX1 bit in IEN0 register
10	10	10	1	T0	Set TR0 bit in TCON register and set C/T0 bit in TMOD register
			2	P3.4	Clear TR0 bit in TCON register and clear C/T0 bit in TMOD register
11	11	11	1	CMP1O	Set CMPEN bit and CMPOC bit in CMP1CON
			2	T1	Set TR1 bit in TCON register and set C/T1 bit in TMOD register
			3	P3.5	Clear CMPEN bit and CMPOC bit in CMP1CON,and clear TR1 bit in TCON register or clear C/T1 bit in TMOD register
12	12	12	1	CMP1P	Set CMPEN bit in CMP1CON
			2	P3.6	Clear CMPEN bit in CMP1CON
13	13	13	1	CMP1N	Set CMPEN bit in CMP1CON
			2	P3.7	Clear CMPEN bit in CMP1CON

PORT4：
- XTAL2 (P4.4): oscillator output
- XTAL1 (P4.5): oscillator input
- PWM0 (P4.6): PWM0 output
- VLPD (P4.7): LPD detection voltage input

Table 8.21 PORT4 Share Table

Pin No.			Priority	Function	Enable Bit
TQFP48	QFP44	LQFP44			
15	14	14	1	XTAL2	Selected by Code Option, when code option is external clock or built-in RC oscillator, P4.4 is I/O
			2	P4.4	Selected by Code Option
17	15	15	1	XTAL1	Selected by Code Option, when code option is built-in RC oscillator, P4.5 is I/O
			2	P4.5	Selected by Code Option
29	26	26	1	PWM0	Set EPWM bit and PWMSS bit in PWM0CON register
			2	P4.6	Clear EPWM bit and PWMSS bit in PWM0CON register
30	27	27	1	VLPD	Set ELPD bit and LPDV bit in LPDCON register
			2	P4.7	Clear ELPD bit and LPDV bit in LPDCON register
20	17	17	1	P4.0	Default
31	28	28	1	P4.1	Default
42	39	39	1	P4.2	Default
6	6	6	1	P4.3	Default

8.8 Timer

8.8.1 Feature

- The SH88F516 has three timers (Timer0, 1, 2)
- Timer0 is compatible with the standard 8051
- Timer1 is compatible with the standard 8051
- Timer2 is compatible with the standard 8052 and has up or down counting and programmable clock output function
- Timer0/1 add 32.768kHz as the clock source function option

8.8.2 Timer0 & Timer1

Each Timer is implemented as a 16-bit register accessed as two cascaded Data Registers: THx & TLx (x = 0, 1). They are controlled by the register TCON and TMOD. The Timer 0 & Timer 1 interrupts can be enabled by setting the ET0 & ET1 bit in the IEN0 register (Refer to **Interrupt** Section for details).

Timer x Mode（x = 0，1）

Both Timers operate in one of four primary modes selected by the Mode Select bits Mx1-Mx0 (x = 0, 1) in the Counter/Timer Mode register (TMOD).

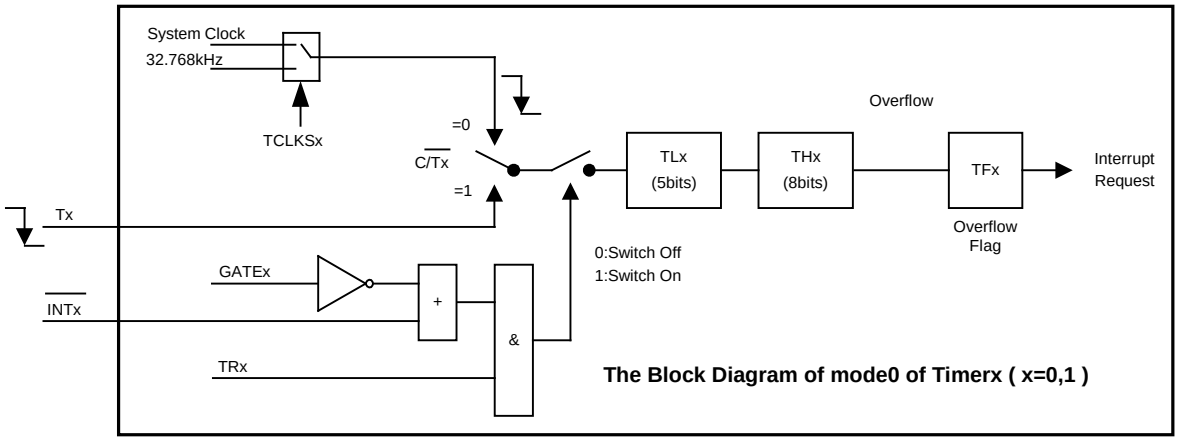
Mode0: 13-bit Counter/Timer

Timer x operate as 13-bit counter/timers in Mode 0. The THx register holds the high eight bits of the 13-bit counter/timer, TLx holds the five low bits(TLx.4 - TLx.0). The three upper bits(TLx.7- TLx.5) of TLx are indeterminate and should be ignored when reading. As the 13-bit timer register increments and overflows, the timer overflow flag TFx is set and an interrupt will occur if Timer interrupts is enabled. The C/$\overline{Tx}$ bit selects the counter/timer's clock source.

If C/$\overline{Tx}$ = 1, high-to-low transitions at the Timer input pin (Tx) will increase the timer/Counter Data register. Else if C/$\overline{Tx}$ = 0, selects the system clock to increase the timer/Counter Data register.

Setting the TRx bit enables the timer when either GATEx = 0, or GATEx = 1 and the input signal $\overline{INTx}$ is active. Setting GATEx to '1' allows the timer to be controlled by the external input signal $\overline{INTx}$, facilitating positive pulse width in $\overline{INTx}$ measurements. Setting TRx does not force the timer to reset, This means that if TRx is set, the timer register will count from the old value that was last stopped by clearing TRx. So the timer registers should be loaded with the desired initial value before the timer is enabled.

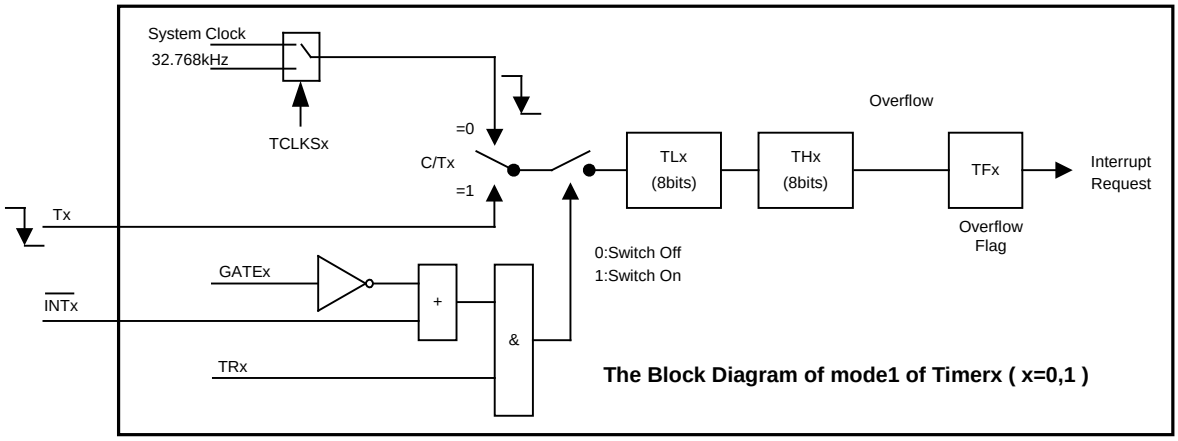
When as Timer, system clock or 32.768kHz can be used as clock source of Timer x (x = 0, 1) by configuring TCLKSx (x = 0, 1) bit in TCON1 register. TCLKSx (x = 0, 1) is valid when selecting 32.768kHz crystal oscillator in code option.



The Block Diagram of mode0 of Timerx (x=0,1)

Mode1: 16-bit Counter/Timer

Mode1 operation is the same as Mode0, except that the counter/timer registers use all 16 bits. The enable and configuration of Counter/Timer in Mode1 is same as Mode0.



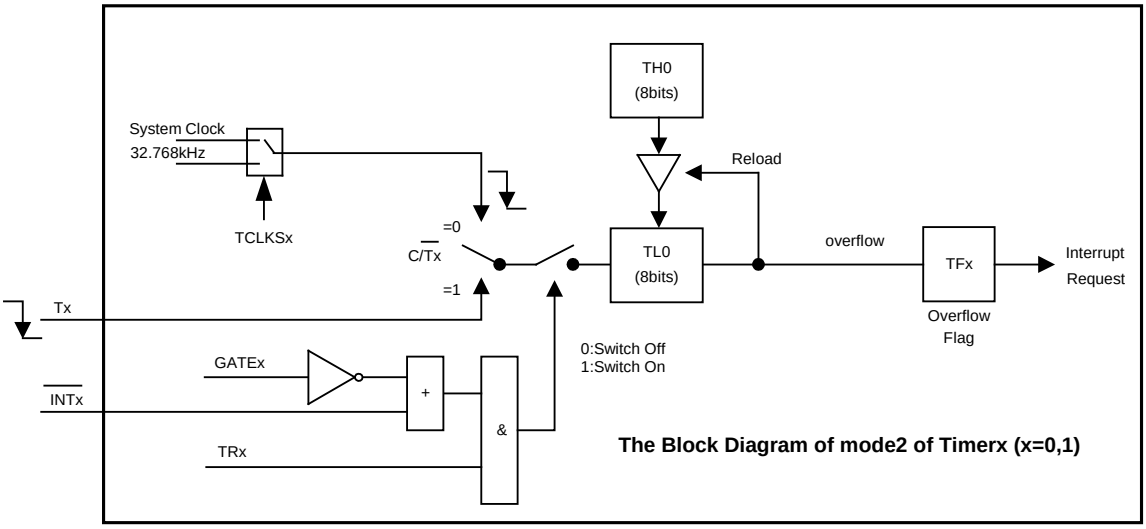
The Block Diagram of mode1 of Timerx (x=0,1)

Mode2: 8-bit Counter/Timer with Auto-Reload

In Mode2, Timer x is configured as 8-bit Counter/Timer with automatic reload of the start value. TLx holds the count and THx holds the reload value. When the counter in TLx overflows from 0xFF to THx, the timer overflow flag TFx is set and the counter in TLx is reloaded from THx. If Timer interrupts are enabled, an interrupt will occur when the TFx flag is set. The reload value in THx is not changed. TLx must be initialized to the desired value before enabling the timer to count correctly.

Except the Auto-Reload function, the enable and configuration of Counter/Timer in Mode2 is same as Mode0 and Mode 1.

When as Timer, system clock or 32.768kHz can be used as clock source of Timer x (x = 0, 1) by configuring TCLKSx (x = 0, 1) bit in TCON1 register. TCLKSx (x = 0, 1) is valid when selecting 32.768kHz crystal oscillator in code option.



The Block Diagram of mode2 of Timerx (x=0,1)

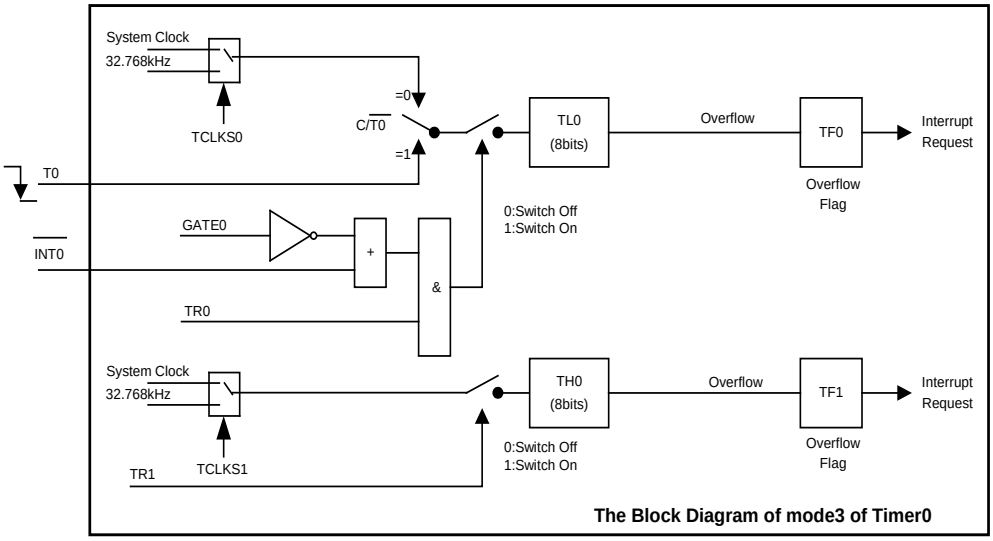
Mode3: Two 8-bit Counter/Timers (Timer0 Only)

In Mode3, Timer0 is configured as two separate 8-bit counter/timers: TH0 and TL0. TL0 is controlled by using the Timer0 control/status bits in TCON and TMOD: TR0, C/$\overline{T0}$, GATE0 and TF0. TL0 can use either the system clock or 32.768kHz or an external input signal as its clock source.

The TH0 is restricted as timer function, system clock is clock source. TH0 is enabled by using the Timer control bit TRx. The overflow flag TF1 will be set to control Timer interrupt when Timer overflows.

When Timer0 works in Mode3, Timer1 can work in Mode0/1/2, but it can not set the TF1 flag and generate an interrupt.The overflow of Timer1 can be used to generatr Baud-Rate of serial ports for EUART.TH1 and TL1 can be used as Timer function only.System clock provides clock source.GATE1 bit is unvalid. And the pull high resistor of T1 input pin is also disabled. Timer1 run control is handled through its mode settings, because TR1 is used by Timer0. When the Timer1 is in Mode0, 1, or 2, Timer1 is enable. When the Timer1 is in Mode3, Timer1 is disable.

When as Timer, system clock or 32.768kHz can be used as clock source of Timer 0 by configuring TCLKS0 bit in TCON1 register. TCLKS0 is valid when selecting 32.768kHz crystal oscillator in code option.



The Block Diagram of mode3 of Timer0

Register

Table 8.22 Timer/Counter x Control Register (x = 0, 1)

88H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7, 5	**TFx** **x = 0, 1**	**Timer x overflow flag** 0: Timer x no overflow, can be cleared by software 1: Timer x overflow, set by hardware; set by software will cause a timer interrupt
6, 4	**TRx** **x = 0, 1**	**Timer x start, stop control bits** 0: Stop timer x 1: Start timer x
3, 1	**IEx** **x = 0, 1**	**External interrupt x request flag**
2, 0	**ITx** **x = 0, 1**	**External interrupt x trigger mode select bits**

Table 8.23 Timer/Counter x Mode Register (x = 0,1)

89H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TMOD	GATE1	C/$\overline{T1}$	M11	M10	GATE0	C/$\overline{T0}$	M01	M00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7, 3	**GATEx x = 0, 1**	**Timer x Gate Control bits** 0: Timer x is enabled whenever TRx control bit is set 1: Timer x is enabled only while $\overline{INTx}$ pin is high and TRx control bit is set
6, 2	**C/$\overline{Tx}$ x = 0, 1**	**Timer x Timer/Counter mode selected bits** 0: Timer Mode, T0 or T1 pin is used as I/O port 1: Counter Mode
5-4 1-0	**Mx[1:0] x = 0, 1**	**Timer x Timer mode selected bits** 00: Mode0, 13-bit up counter/timer, bit7- 5 of TLx is ignored 01: Mode1, 16-bit up counter/timer 10: Mode2, 8-bit auto-reload up counter/timer 11: Mode3 (only for Timer0), two 8-bit up timer

Table 8.24 Timer/Counter x Data Register (x = 0, 1)

8AH-8DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TL0 (8AH)	TL0.7	TL0.6	TL0.5	TL0.4	TL0.3	TL0.2	TL0.1	TL0.0
TH0 (8CH)	TH0.7	TH0.6	TH0.5	TH0.4	TH0.3	TH0.2	TH0.1	TH0.0
TL1 (8BH)	TL1.7	TL1.6	TL1.5	TL1.4	TL1.3	TL1.2	TL1.1	TL1.0
TH1 (8DH)	TH1.7	TH1.6	TH1.5	TH1.4	TH1.3	TH1.2	TH1.1	TH1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**TLx.y, THx.y x=0-1, y=0-7**	Timer x Low & High byte counter

Table 8.25 定时器/计数器x控制寄存器（x = 0，1）

CEH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON1	-	TCLKS1	TCLKS0	-	-	-	-	-
R/W	-	R/W	R/W	-	-	-	-	-
Reset Value (POR/WDT/LVR/PIN)	-	0	0	-	-	-	-	-

Bit Number	Bit Mnemonic	Description
6-5	**TCLKSx x = 0,1**	**Timer x Clock Source Control bits** 0: Select system clock as Timer x Clock Source 1: Select 32.768kHz as Timer x Clock Source

8.8.3 Timer2

The Timer 2 is implemented as a 16-bit register accessed as two cascaded data registers: TH2 and TL2. It is controlled by the register T2CON and T2MOD. The Timer2 interrupt can be enabled by setting the ET2 bit in the IEN0 register. (Refer to Interrupt Section for details)

The operating modes of Timer2 is similar with Timer0/1. C/$\overline{T2}$ selects system clock (Timer) or external pin T2 (Counter) as the timer clock input. Setting TR2 allows Timer 2/Counter 2 Data Register to increment by the selected input.

Timer2 Modes

Timer 2 has 4 operating modes: Capture/Reload, Auto-reload mode with up or down counter, Baud Rate Generator and Programmable clock-output. These modes are selected by the combination of RCLK, TCLK and CP/RL2.

Timer2 Mode select

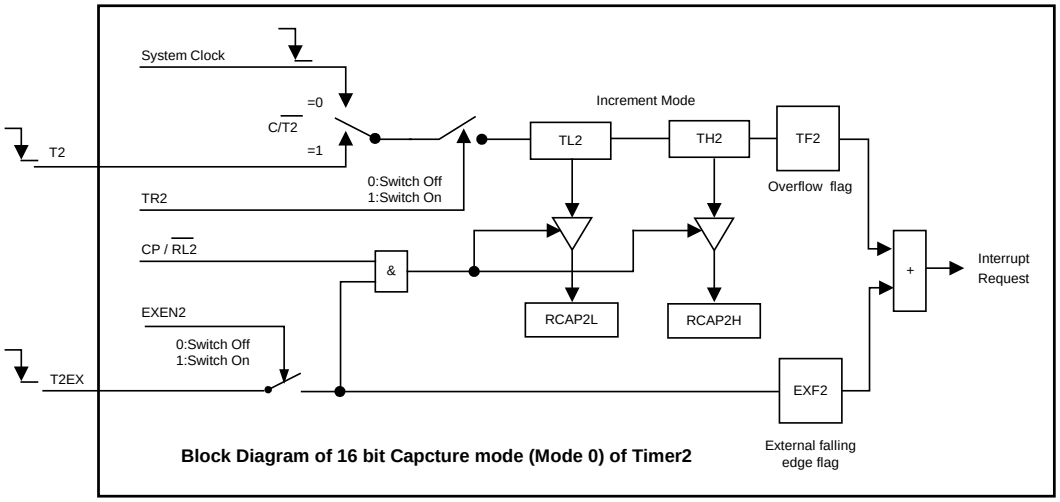
C/$\overline{T2}$	T2OE	DCEN	TR2	CP/RL2	RCLK	TCLK	Mode	
X	0	X	1	1	0	0	0	16 bit capture
X	0	0	1	0	0	0	1	16 bit auto-reload timer
X	0	1	1	0	0	0		
X	0	X	1	X	1	X	2	Baud-Rate generator
					X	1		
0	1	X	1	X	0	0	3	Programmable clock-output only
					1	X	3	Programmable clock-output, with Baud-rate generator
					X	1		
X	X	X	0	X	X	X	X	Timer2 stop, the T2EX path still enable

Mode0: 16 bit Capture

In the capture mode, two options are selected by EXEN2 bit in T2CON.

If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which will set TF2 on overflow to generate an interrupt if ET2 is enabled.

If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L respectively. In addition, a 1-to-0 transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can also generate an interrupt if ET2 is enabled.



Block Diagram of 16 bit Capcture mode (Mode 0) of Timer2

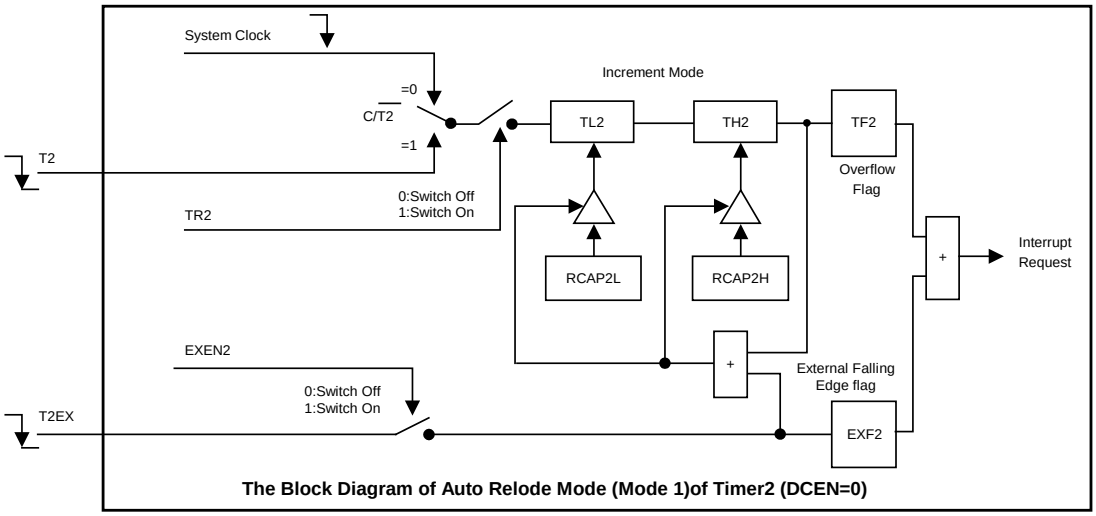
Mode1: 16 bit Auto-reload Timer

Timer2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit in T2MOD. After reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer2 can count up or down, depending on the value of the T2EX pin.

When DCEN = 0, two options are selected by bit EXEN2 in T2CON.

If EXEN2 = 0, Timer2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L, which are pressed by software.

If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if ET2 is enabled.



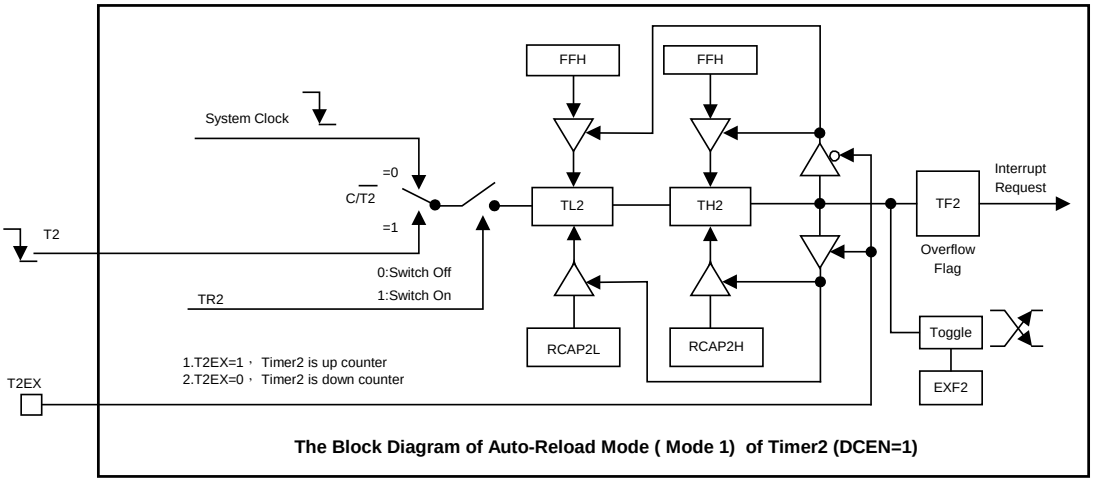
The Block Diagram of Auto Relode Mode (Mode 1)of Timer2 (DCEN=0)

Setting the DCEN bit enables Timer2 to count up or down. When DCEN = 1, the T2EX pin controls the direction of the count, and EXEN2's control is invalid.

A logical "1" at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logical "0" at T2EX makes Timer2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit can be used as a 17th bit of resolution whenever Timer2 overflows or underflows. In this operating mode, EXF2 does not flag an interrupt.



The Block Diagram of Auto-Reload Mode (Mode 1) of Timer2 (DCEN=1)

Mode2: Baud-Rate Generator

Timer2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON. The baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer1 is used for the other.

Setting RCLK and/or TCLK will put Timer2 into its baud rate generator mode, which is similar to the auto-reload mode.

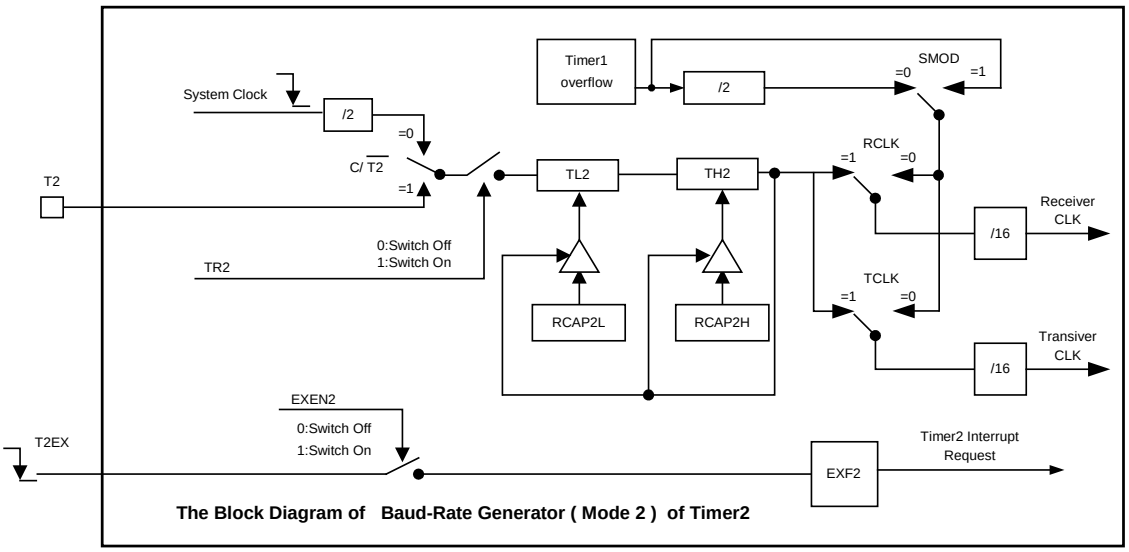
Over flow of Timer2 will causes the Timer2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L that preset by software. But this will not generate an interrupt.

If EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload. Thus when Timer 2 is used as a baud rate generator, T2EX can be used as an extra external interrupt.

The baud rates in EUART0 Mode1 and 3 are determined by Timer2's overflow rate according to the following equation.

$$\text{BaudRate} = \frac{1}{2X16} X \frac{\text{System Clock}}{65536 - [\text{RCAP2H,RCAP2L}]}; \ C/\overline{T2} = 0$$

$$\text{BaudRate} = \frac{1}{16} X \frac{\text{T2 frequency}}{65536 - [\text{RCAP2H,RCAP2L}]}; \ C/\overline{T2} = 1$$



The Block Diagram of Baud-Rate Generator (Mode 2) of Timer2

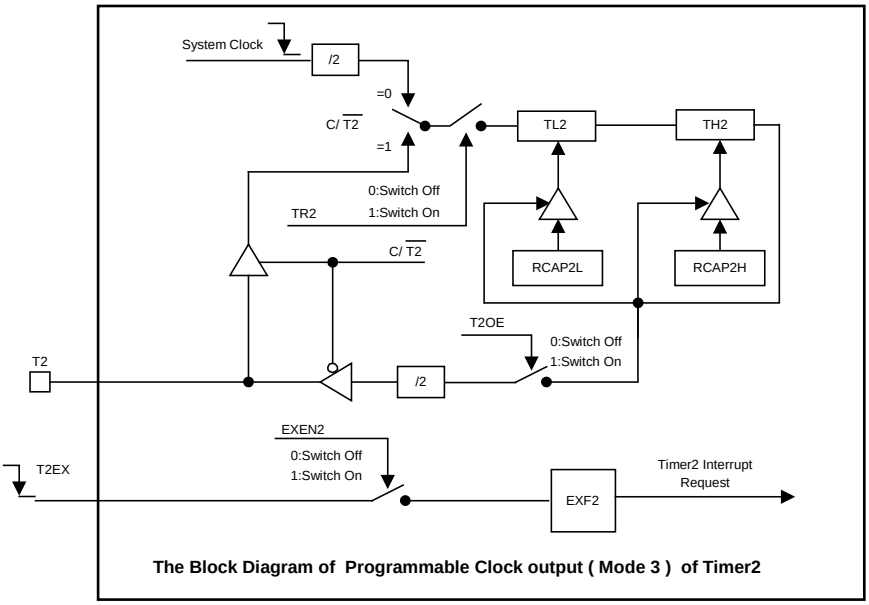
Mode3: Programmable Clock Output

A 50% duty cycle clock can be programmed to come out on T2 pin. To configure the Timer 2 as a clock generator, bit C/$\overline{T2}$ must be cleared and bit T2OE must be set. Bit TR2 starts and stops the timer.

In this mode T2 will output a 50% duty cycle clock.

$$Clock\ \ Out\ \ Frequency = \frac{1}{2\times 2}\times\frac{System\ \ Clock}{65536-[RCAP2H,RCAP2L]}$$

Timer2 overflow will not generate an interrupt, so it is possible to use Timer2 as a baud-rate generator and a clock output simultaneously with the same frequency.



The Block Diagram of Programmable Clock output (Mode 3) of Timer2

Note:

(1) Both TF2 and EXF2 can cause timer2 interrupt request, and they have the same vector address.

(2) TF2 and EXF2 are set as 1 by hardware while event occurs. But they can also be set by software at any time. Only the software and the hardware reset will be able to clear TF2 & EXF2 to 0.

(3) When EA = 1 & ET2 = 1, setting TF2 or EXF2 as 1 will cause a timer2 interrupt.

(4) While Timer2 is used as baud rate generator, writing TH2/TL2, writing RCAPH2/RCAPL2 will affect the accuracy of baud rate, thus might make cause communication error.

Register

Table 8.26 Timer2 Control Register

C8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$\overline{C/T2}$	$\overline{CP/RL2}$
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**TF2**	**Timer2 overflow flag bit** 0: No overflow (Must be cleared by software) 1: Overflow (Set by hardware if RCLK = 0 & TCLK = 0)
6	**EXF2**	**External event input (falling edge) from T2EX pin detected flag bit** 0: No external event input (Must be cleared by software) 1: Detected external event input (Set by hardware if EXEN2 = 1)
5	**RCLK**	**EUART0 Receive Clock control bit** 0: Timer1 generates receiveing baud-rate 1: Timer2 generates receiveing baud-rate
4	**TCLK**	**EUART0 Transmit Clock control bit** 0: Timer1 generates transmitting baud-rate 1: Timer2 generates transmitting baud-rate
3	**EXEN2**	**External event input (falling edge) from T2EX pin used as Reload/Capture trigger enable/disable control bit** 0: Ignore events on T2EX pin 1: Cause a capture or reload when a negative edge on T2EX pin is detected, when Timer2 is not used as clock of EUART0 (T2EX always has a pull up resistor)
2	**TR2**	**Timer2 start/stop control bit** 0: Stop Timer2 1: Start Timer2
1	**$\overline{C/T2}$**	**Timer2 Timer/Counter mode selected bit** 0: Timer Mode, T2 pin is used as I/O port 1: Counter Mode, the internal pull-up resister is turned on
0	**$\overline{CP/RL2}$**	**Capture/Reload mode selected bit** 0: 16 bits timer/counter with reload function 1: 16 bits timer/counter with capture function

Table 8.27 Timer2 Mode Control Register

C9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
T2MOD	-	-	-	-	-	-	T2OE	DCEN
R/W	-	-	-	-	-	-	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
1	**T2OE**	**Timer2 Output Enable bit** 0: Set P1.0/T2 as clock input or I/O port 1: Set P1.0/T2 as clock output (Baud-Rate generator mode)
0	**DCEN**	**Down Counter Enable bit** 0: Disable Timer2 as up/down counter, Timer2 is an up counter 1: Enable Timer2 as up/down counter

Table 8.28 Timer2 Reload/Capture & Data Register

CAH - CDH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RCAP2L	RCAP2L.7	RCAP2L.6	RCAP2L.5	RCAP2L.4	RCAP2L.3	RCAP2L.2	RCAP2L.1	RCAP2L.0
RCAP2H	RCAP2H.7	RCAP2H.6	RCAP2H.5	RCAP2H.4	RCAP2H.3	RCAP2H.2	RCAP2H.1	RCAP2H.0
TL2	TL2.7	TL2.6	TL2.5	TL2.4	TL2.3	TL2.2	TL2.1	TL2.0
TH2	TH2.7	TH2.6	TH2.5	TH2.4	TH2.3	TH2.2	TH2.1	TH2.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**RCAP2L.x**	**Timer2 Reload/Capturer Data, x = 0 - 7**
	RCAP2H.x	
7-0	**TL2.x**	**Timer2 Low & High byte counter, x = 0 - 7**
	TH2.x	

8.9 Interrupt

8.9.1 Feature

- 15 interrupt sources
- 4 interrupt priority levels

The SH88F516 provides total 15 interrupt sources: 3 external interrupts (INT0/1/4), INT4 has 8 interrupt sources (INT40-47, which share the same vector address), 3 timer interrupts (Timer0, 1, 2), LPD interrupt, 2 CMP interrupts, 2 EUART interrupts, ADC Interrupt, SPI Interrupt, SCM interrupt, and PWM interrupt (PWM0/1/2, which share the same vector address).

8.9.2 Interrupt Enable Control

Each interrupt source can be individually enabled or disabled by setting or clearing the corresponding bit in the interrupt enable registers IEN0 or IEN1. The IEN0 register also contains global interrupt enable bit, EA, which can enable/disable all the interrupts at once. Generally, after reset, all interrupt enable bits are set to 0, which means that all the interrupts are disabled.

8.9.3 Register

Table 8.29 Primary Interrupt Enable Register

A8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN0	EA	EADC	ET2	ES0	ET1	EX1	ET0	EX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**EA**	**All interrupt enable bit** 0: Disable all interrupt 1: Enable all interrupt
6	**EADC**	**ADC interrupt enable bit** 0: Disable ADC interrupt 1: Enable ADC interrupt
5	**ET2**	**Timer2 overflow interrupt enable bit** 0: Disable Timer2 overflow interrupt 1: Enable Timer2 overflow interrupt
4	**ES0**	**EUART0 interrupt enable bit** 0: Disable EUART0 interrupt 1: Enable EUART0 interrupt
3	**ET1**	**Timer1 overflow interrupt enable bit** 0: Disable Timer1 overflow interrupt 1: Enable Timer1 overflow interrupt
2	**EX1**	**External interrupt 1 enable bit** 0: Disable external interrupt1 1: Enable external interrupt1
1	**ET0**	**Timer0 overflow interrupt enable bit** 0: Disable Timer0 overflow interrupt 1: Enable Timer0 overflow interrupt
0	**EX0**	**External interrupt 0 enable bit** 0: Disable external interrupt0 1: Enable external interrupt0

Table 8.30 Secondary Interrupt Enable Register

A9H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IEN1	ELPD	EX4	EPWM	ESCM	ECMP1	ES1	ECMP0	ESPI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**ELPD**	**LPD interrupt enable bit** 0: Disable LPD interrupt (Default) 1: Enable LPD interrupt
6	**EX4**	**External interrupt4 enable bit** 0: Disable external interrupt4 (Default) 1: Enable external interrupt4
5	**EPWM**	**PWM interrupt enable bit** 0: Disable PWM interrupt (Default) 1: Enable PWM interrupt
4	**ESCM**	**SCM interrupt enable bit** 0: Disable SCM interrupt (Default) 1: Enable SCM interrupt
3	**ECMP1**	**Comparator1 output interrupt enable bit** 0: Disable Comparator1 interrupt (Default) 1: Enable Comparator1 interrupt
2	**ES1**	**EUART1 interrupt enable bit** 0: Disable EUART1 interrupt (Default) 1: Enable EUART1 interrupt
1	**ECMP0**	**Comparator0 output interrupt enable bit** 0: Disable Comparator0 interrupt (Default) 1: Enable Comparator0 interrupt
0	**ESPI**	**SPI interrupt enable bit** 0: Disable SPI interrupt (Default) 1: Enable SPI interrupt

Table 8.31 Interrupt channel Enable Register

BAH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IENC	EXS47	EXS46	EXS45	EXS44	EXS43	EXS42	EXS41	EXS40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**EXS4x (x = 0-7)**	**External interrupt4 select bit (x = 7-0)** 0: Disable external interrupt 4x (Default) 1: Enable external interrupt 4x

8.9.4 Interrupt Flag

Each Interrupt source has its own interrupt flag, when interrupt occurs, corresponding flag will be set by hardware, the interrupt flag bits are listed in interrupt abstract table.

For **external interrupt (INT0/1)**, when an external interrupt0/1 is generated, if the interrupt was edge trigged, the flag (IE0-1 in TCON) that generated this interrupt is cleared by hardware when the service routine is vectored. If the interrupt was level trigged, then the requesting external source directly controls the request flag, rather than the on-chip hardware.

When INT4 generates an interrupt, the flag bit IF4x (x = 0-7) in EXF1 register will be set.The flag bit should be cleared by user's program because the same vector entrance was used in INT4.But if INT4 is level triggered, the flag can't be cleared by user's program, it only be controlled by peripheral signal level that connect to interrupt source pin.

The **Timer0/1 interrupt** is generated when they overflows, the flag (TFx, x = 0, 1) in TCON register, which is set by hardware, and will be automatically be cleared by hardware when the service routine is vectored.

The **Timer2 interrupt** is generated by the logical OR of flag TF2 and bit EXF2 in T2CON register, which is set by hardware. None of these flags can be cleared by hardware when the service routine is vectored. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, so the flag must be cleared by software.

The **EUART interrupt** is generated by the logical OR of flag RI and TI in SCON register, which is set by hardware. Neither of these flags can be cleared by hardware when the service routine is vectored. In fact, the service routine will normally have to determine whether it was the receive interrupt flag or the transmission interrupt flag that generated the interrupt, so the flag must be cleared by software.

The **ADC interrupt** is generated by ADCIF bit in ADCON. If an interrupt is generated, the converted result in ADCDH/ADCDL will be valid. If continuous compare function in ADC module is Enable, ADCIF will not be clear at each conversion when conversion results is less than the compare value. But if converted result is larger than compare value, ADCIF bit will be 1. The flag must be cleared by software.

The **SPI interrupt** is generated by setting SPIF bit or MODF bit in SPSTA register, the flag bit must be cleared by software.

The **SCM interrupt** is generated by setting SCMIF bit in CLKCON register, the flag bit must be cleared by hardware.

The **PWM interrupt** is generated by setting PWMIF bit in PWMxCON register, the flag bit must be cleared by software.

The **LPD interrupt** is generated by setting LPDF bit in LPDCON register, the flag bit can be cleared by hardware or software.But, the flag bit is set by hardware.

The **CMP interrupt** is generated by setting CMPIF bit in CMPCON register.The flag bit must be cleared by software.

Table 8.32 Timer/Counter x Control Register (x = 0, 1)

88H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7, 5	**TFx (x= 0, 1)**	**Timer x overflow flag** 0: Timer x no overflow 1: Timer x overflow
6, 4	**TRx (x= 0, 1)**	**Timer x start, stop control bits** 0: Stop timer x 1: Start timer x
3, 1	**IEx (x= 0, 1)**	**External interrupt x request flag bit** 0: No interrupt pending 1: Interrupt is pending
2, 0	**ITx (x= 0, 1)**	**External interrupt x trigger mode selection bit** 0: Low level trigger 1: Falling edge trigger

Table 8.33 External Interrupt Flag Register

AAH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EXF0	IT4.1	IT4.0	-	-	-	-	-	-
R/W	R/W	R/W	-	-	-	-	-	-
Reset Value (POR/WDT/LVR/PIN)	0	0	-	-	-	-	-	-

Bit Number	Bit Mnemonic	Description
7-6	**IT4[1:0]**	**External interrupt 4 trigger mode selection bit** 00: Low Level trigger 01: Trigger on falling edge 10: Trigger on rising edge 11: Trigger on both edge IT4[1:0] configuration external interrupt 4x for the same trigger mode

Table 8.34 External Interrupt4 Flag Register

D8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
EXF1	IF47	IF46	IF45	IF44	IF43	IF42	IF41	IF40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**IF4x (x = 0-7)**	**External interrupt4 request flag bit** 0: No interrupt pending 1: Interrupt is pending IF4x is cleared by software

8.9.5 Interrupt Vector

When an interrupt occurs, the program counter is pushed onto the stack and the corresponding interrupt vector address is loaded into the program counter. The interrupt vector addresses are listed in **Interrupt Summary table**.

8.9.6 Interrupt Priority

Each interrupt source can be individually programmed to one of four priority levels by setting or clearing corresponding bits in the interrupt priority control registers: IPL0, IPH0, IPL1, and IPH1.The interrupt priority service is described below.

An interrupt service routine in progress can be interrupted by a higher priority interrupt, but can not by another interrupt with the same or lower priority.

The highest priority interrupt service cannot be interrupted by any other interrupt source. If two requests of different priority levels are received simultaneously, the request of higher priority level is serviced.

If the same priority interrupt source apply for the interrupt at the beginning of the instruction cycle at the same time, an internal polling sequence determines which request is serviced.

Interrupt Priority		
Priority bits		**Interrupt Lever Priority**
IPHx	IPLx	
0	0	Level 0 (lowest priority)
0	1	Level 1
1	0	Level 2
1	1	Level 3 (highest priority)

Table 8.35 Interrupt Priority Control Registers

B8H, B4H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
IPL0 (B8H)	-	PADCL	PT2L	PSL	PT1L	PX1L	PT0L	PX0L
IPH0 (B4H)	-	PADCH	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	0	0	08	0	0	0	0
B9H, B5H	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
IPL1 (B9H)	PLPDL	PX4L	PPWML	PSCML	PCMP1L	PS1L	PCMP0L	PSPIL
IPH1 (B5H)	PLPDH	PX4H	PPWMH	PSCMH	PCMP1H	PS1H	PCMP0H	PSPIH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
-	**PxxxL/H**	Corresponding interrupt source xxx's priority level selection bits

8.9.7 Interrupt Handling

The interrupt flags are sampled and captured at each machine cycle. All interrupts are sampled at the rising edge of the clock. If one of the flags was set, the CPU will find it and the interrupt system will generate a LCALL to the appropriate service routine, LCALL generated by hardware will be prevented by any of the following conditions:

An interrupt of equal or higher priority is already in progress.

The current cycle is not in the final cycle of the instruction in progress. In other words, any interrupt request can not get response before executing instructions to complete.

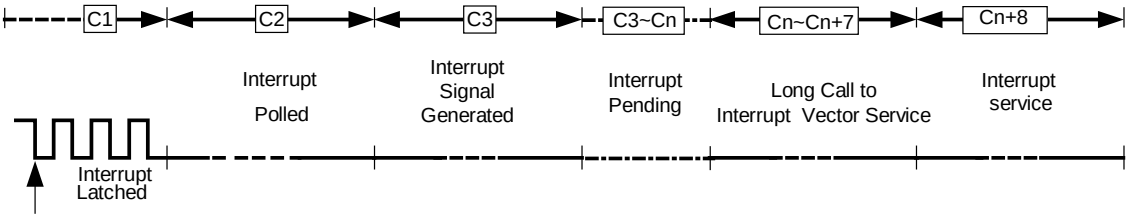
The instruction in progress is RETI or visit the special register IEN0/1 or IPL/H instruction. This ensures that if the instruction in progress is RETI or read and write IEN0/1 or IPL/H then at least one more instruction except RETI will be executed before any interrupt is vectored to; this delay guarantees that the CPU can observe the changes of the interrupt status.

Note:

Since priority change normally needs 2 instructions, it is recommended to disable corresponding Interrupt Enable flag to avoid interrupt between these 2 instructions during the change of priority.

If the flag is no longer active when the blocking condition is removed, the denied interrupt will not be serviced. Every polling cycle interrogates only the valid interrupt requests.

The polling cycle/LCALL sequence is illustrated below:



Interrupt Response Time

The hardware-generated LCALL pushes the contents of the program counter onto the stack (but it does not save the PSW) and reloads the program counter with corresponding address that depends on the source of the interrupt being vectored too, as shown in Interrupt Summary table.

Interrupt service execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that the interrupt routine is no longer in progress, and then pops the top two bytes from the stack and reloads the program counter. Execution of the interrupted program continues from the point where it was stopped. Note that the RETI instruction is very important because it informs the processor that the program left the current interrupt service. A simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking an inte.

8.9.8 Interrupt Response Time

If an interrupt is recognized, its request flag is set in every machine cycle after recognize. The value will be polled by the circuitry until the next machine cycle; the CPU will generate an interrupt at the third machine cycle. If the request is active and conditions are right for it to be acknowledged, hardware LCALL to the requested service routine will be the next instruction to be executed. Else the interrupt will pending. The call itself takes 7 machine cycles. Thus a minimum of 3+7 complete machine cycles will elapse between activation and external interrupt request and the beginning of execution of the first instruction of the service routine.

A longer response time would be obtained if the request was blocked by one of the above three previously listed conditions. If an interrupt of equal or higher priority is already in progress, the additional wait time obviously depends on the nature of the other interrupt's service routine.

If the instruction in progress is not in its final cycle and the instruction in progress is RETI， the additional wait time is 8 machine cycles. For a single interrupt system, if the next instruction is 20 machine cycles long (the longest instructions DIV & MUL are 20 machine cycles long for 16 bit operation), adding the LCALL instruction 7 machine cycles the total response time is 2+8+20+7 machine cycles.

Thus interrupt response time is always more than 10 machine cycles and less than 37 machine cycles.

8.9.9 External Interrupt Inputs

The SH88F516 has 3 external interrupt inputs. External interrupt0-1 each has one vector address, External interrupt 4 has 8 inputs sharing an interrupt vector address. External interrupt0-1 can be programmed to be level-triggered or edge-triggered by clearing or setting bit IT1 or IT0 in TCON register. If ITx = 0 (x = 0, 1), external interrupt INTx (x = 0, 1) is triggered by a low level detected. If ITx = 1 (x = 0, 1), external interrupt INTx (x = 0, 1) is edge triggered. In this mode if consecutive samples of the INTx (x = 0, 1)pin show a high level in one cycle and a low level in the next cycle, interrupt request flag in register TCON is set, causing an interrupt request. Since the external interrupt pins are sampled once each machine cycle, an input high or low level should be held for at least one machine cycle to ensure proper sampling.

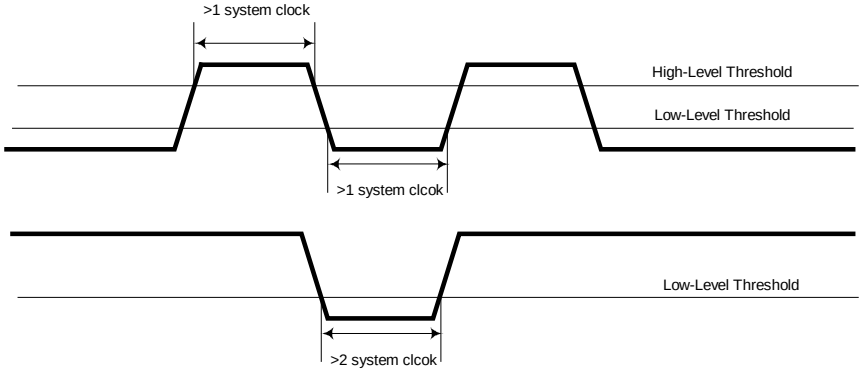
If the external interrupt is edge-triggered, the external source has to hold the request pin high for at least one machine cycle, and then hold it low for at least one machine cycle. This is to ensure that the transition is detected and that interrupt request flag is set. Notice that IE0-1 is automatically cleared by CPU when the service routine is called.

If the external interrupt is level-triggered, the external source must hold the request active until the requested interrupt is generated, which will take 2 machine cycles. If the external interrupt is still valid when the interrupt service routine is completed, another interrupt will be generated. It is not necessary to clear the interrupt flag IEx (x = 0, 1), when the interrupt is triggered by level, it simply tracks the input pin level.

External interrupt4 has more interrupt trigger modes than INT0/1, operation is similar to INT0/1.

If an external interrupt is enabled when the SH88F516 is put into Power down or Idle mode, the interrupt occurrence will cause the processor to wake up and resume operation. (Refer to "Power Management" section for details)

***Note**: IE0-1 is automatically cleared by hardware when the service routine is called while IF40-47 should be cleared by software.*



8.9.10 Interrupt Summary

Source	Vector Address	Enable bits	Flag bits	Polling Priority	Interrupt number(C51)
Reset	0000h			0 (highest)	-
INT0	0003h	EX0	IE0	1	0
Timer0	000Bh	ET0	TF0	2	1
INT1	0013h	EX1	IE1	3	2
Timer1	001Bh	ET1	TF1	4	3
EUART0	0023h	ES0	RI+TI	5	4
Timer2	002Bh	ET2	TF2+EXF2	6	5
ADC	0033h	EADC	ADCIF	7	6
SPI	003Bh	ESPI	SPIF	8	7
CMP0	0043h	ECMP0	CMP0IF	9	8
EUART1	004Bh	ES1	RI1+TI1	10	9
CMP1	0053h	ECMP1	CMP1IF	11	10
SCM	005Bh	ESCM	SCMIF	12	11
PWM	0063h	EPWM	PWMIF	13	12
INT4	006Bh	EX4	IF47-40	14	13
LPD	0073h	ELPD	LPDF	15 (lowest)	14

9. Enhanced Function

9.1 PWM (Pulse Width Modulation)

The SH88F516 has 3 8-bit PWM module. Which can provide the pulse width modulation waveform with the period and the duty being controlled individually by corresponding register.

PWM timer also provides 3 interrupts for PWM0. This makes it possible to change period or duty of next cycle in every PWM period.

Table 9.1 PWMx (x = 0-2) Timer Control Register

D9H - DBH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMxCON (x=0-2)	EPWMx	PWMxS	PWMxCK1	PWMxCK0	-	-	PWMxIF	PWMxSS
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	-	-	0	0

Bit Number	Bit Mnemonic	Description
7	**EPWMx**	**PWMx enable bit** 0: PWMx Timer off (Default) 1: PWMx Timer on
6	**PWMxS**	**PWMx output normal mode of duty cycle** 0: high active (Default) 1: low active
5-4	**PWMxCK1-0**	**PWMx clock source selector** 00: system frequency/2 (Default) 01: system frequency/4 10: system frequency/8 11: system frequency/16
3-2	**-**	**-**
1	**PWMxIF**	**PWMx interrupt flag bit** 0: no interrupt, cleared by software (Default) 1: interrupt occur, set by hardware to indicate that the PWM0 period counter overflow
0	**PWMxSS**	**PWMx output pin function selection** 0: IO (Default) 1: PWM output

Table 9.2 PWMx (x = 0-2)Period Control Register

D1H - D3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMxP (x=0-2)	PWMxP.7	PWMxP.6	PWMxP.5	PWMxP.4	PWMxP.3	PWMxP.2	PWMxP.1	PWMxP.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**PWMxP[7:0]**	**PWMx Period Registers** PWM output period = PWMxP[7:0] X PWM clock no matter how much PWM duty When PWMxP[7:0] = 000H, if PWMS = 0, PWMx output low level, no matter how much PWM duty When PWMxP[7:0] = 000H, if PWMS = 1, PWMx output high level, no matter how much PWM duty

Table 9.3 PWMx (x = 0-2) Duty Control Register

C1H - C3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PWMxD (x=0-2)	PWMxD.7	PWMxD.6	PWMxD.5	PWMxD.4	PWMxD.3	PWMxD.2	PWMxD.1	PWMxD.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**PWMxD[7:0]**	**PWMx Duty Register** PWM output duty = PWMxD[7:0] X PWM clock When PWMxP[7:0] ≤ PWMxD[7:0], if PWMS = 0, PWMx output high level When PWMxP[7:0] ≤ PWMxD[7:0], if PWMS = 1, PWMx output low level

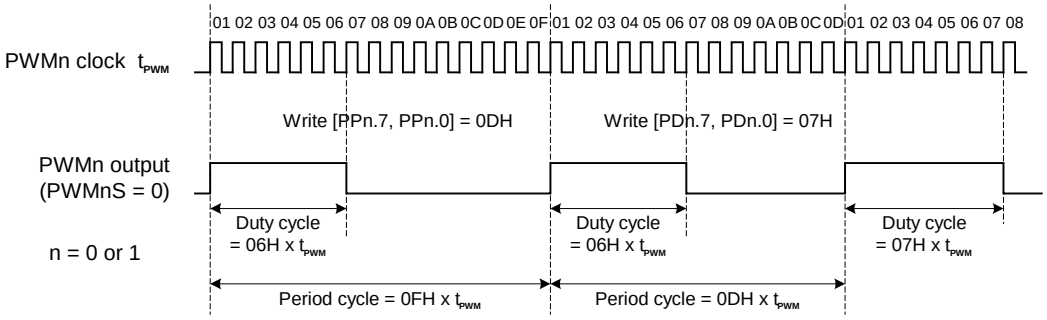
Programming Note:

(1) Set the PWM period/duty cycle by writing proper value to the PWM period control register (PWMP) or PWM duty control register (PWMD).

(2) Select the PWM output mode (high level valid or low level valid) by writing the PWMS bit in the PWM control register (PWMCON).

(3) In order to output the desired PWM waveform, enable the PWM module by writing "1" to the EPWMx bit in the PWM control register (PWMCON).

(4) If the PWM period cycle or duty cycle is to be changed, the writing flow should be followed as described in step 1 or step 2. The modified reloading counter value will take effect in the next period.



PWM output Period or Duty cycle changing example

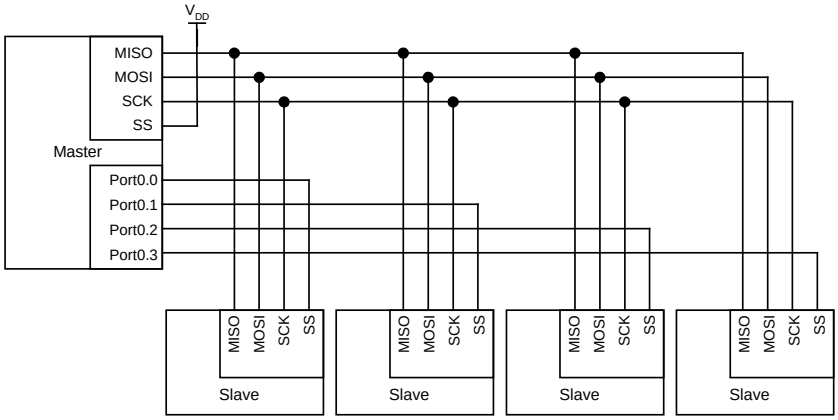
9.2 Serial Peripheral Interface (SPI)

9.2.1 Feature

- 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
- Selectable LSB or MSB transfer

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

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 $\overline{SS}$ pins of the Slave devices.



9.2.2 Signal Description

(1) Master Output Slave Input (MOSI)

This 1-bit signal is directly connected between the master device and slave devices. 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.

(2) Master Input Slave Output (MISO)

This 1-bit signal is directly connected between the slave devices and 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 ($\overline{SS}$ high).

A static high level on the $\overline{SS}$ pin puts the MISO line of a slave in a high-impedance state.

(3) 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 exchanging one byte on the serial lines. The SCK signal is ignored by a SPI slave when the slave is not selected ($\overline{SS}$ high).

(4) Slave Select ($\overline{SS}$)

Each slave peripheral is selected by one slave select pin ($\overline{SS}$). This signal must stay low for any active slave. It is obvious that only one master ($\overline{SS}$ 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 $\overline{SS}$ 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 $\overline{SS}$ pin could be used as a general IO if the following conditions are met:

(a) 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.

(b) 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 $\overline{SS}$ pin to select the target communication slave.

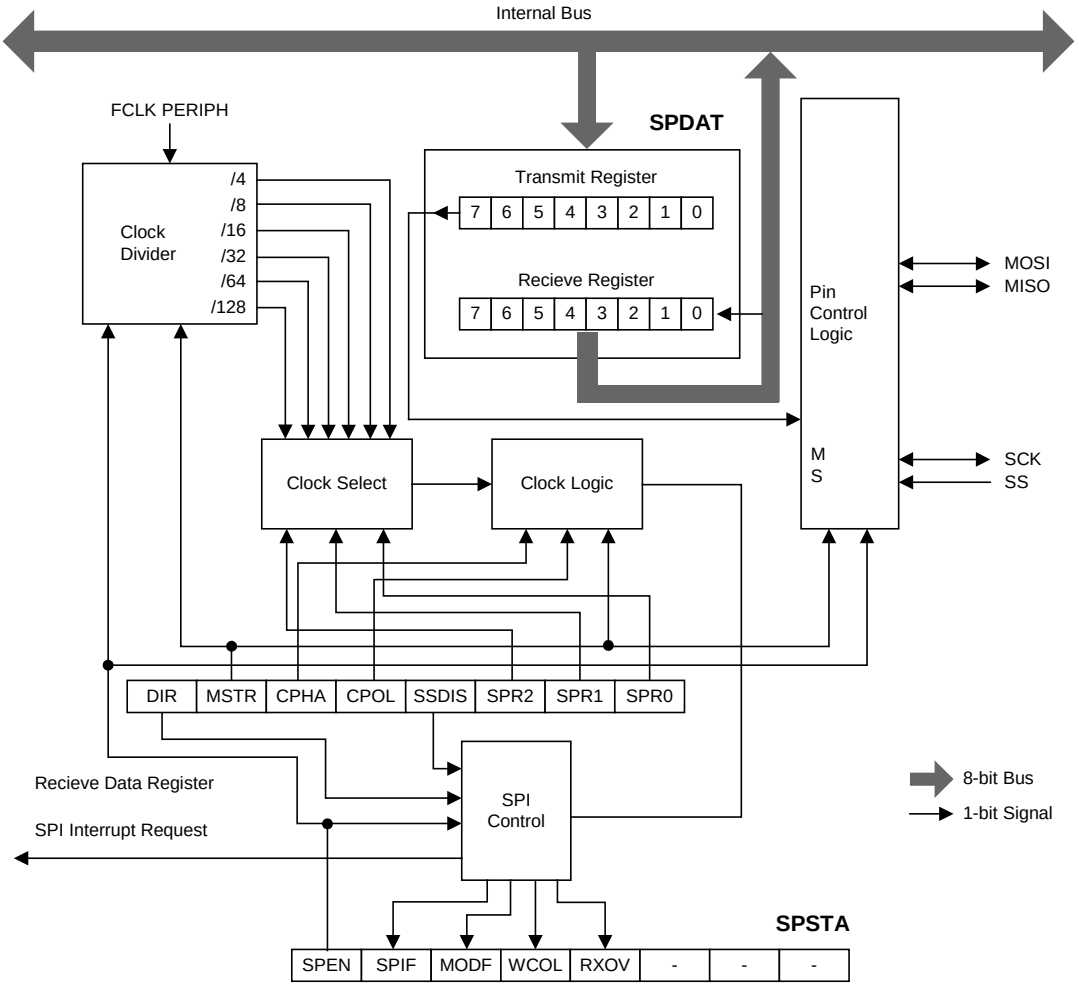
Note: *When CPHA = '0', a falling edge of $\overline{SS}$ pin is used to start the transmission.*

9.2.3 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.

9.2.4 Functional Description

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



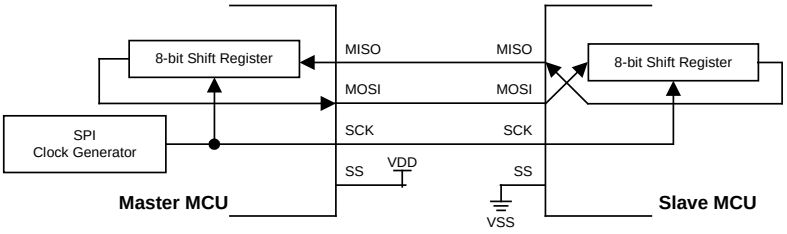
SPI Module Block Diagram

9.2.5 Operating Modes

The Serial Peripheral Interface can be configured as one of the two modes, master mode or slave mode. The configuration and initialization of the SPI module is made through SPCON (the serial peripheral control register) and SPSTA (the serial peripheral status register). Once the SPI is configured, the data exchange is made using SPCON, SPSTA 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 ($\overline{SS}$) allows individual selection of a SPI slave; SPI slaves 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 SPDAT will write to the transmit shift register, and a read operation from SPDAT will retrieve the data in receive shift register.



Full-Duplex Master-Slave Interconnection Diagram

Master Mode

(1) 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.

(2) 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.

(3) 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 or LSB-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.

Slave Mode

(1) Enable

The SPI operates in slave mode when the MSTR is cleared in the SPCON register. Before a data transmission occurs, the slave select ($\overline{SS}$) pin of the Slave device must be set to '0'. The $\overline{SS}$ pin must remain low until the 1-byte transmission is complete.

(2) Transmit & Receive

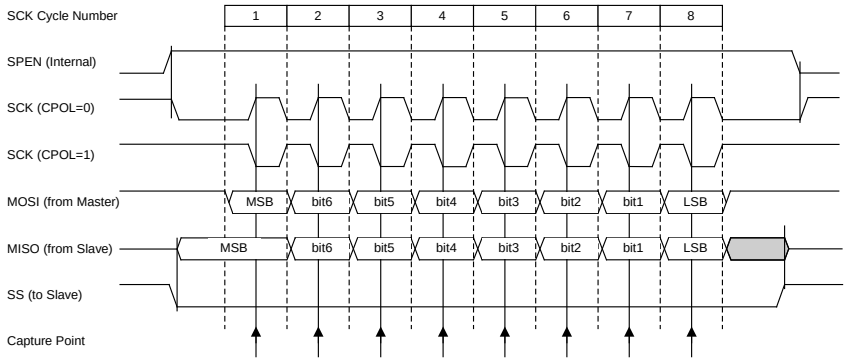
When in SPI slave mode, bytes are shifted in through the MOSI pin and out through the MISO pin by a master device controlling the SCK signal. A bit counter counts SCK edges. When 8 bits have been shifted in the receive shift register and another 8 bits have been shifted out the transmit shift register, 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 clear the **SPIF bit in SPSTA register** before another byte enters the receive 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 in the first transmission, the SPI slave 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.

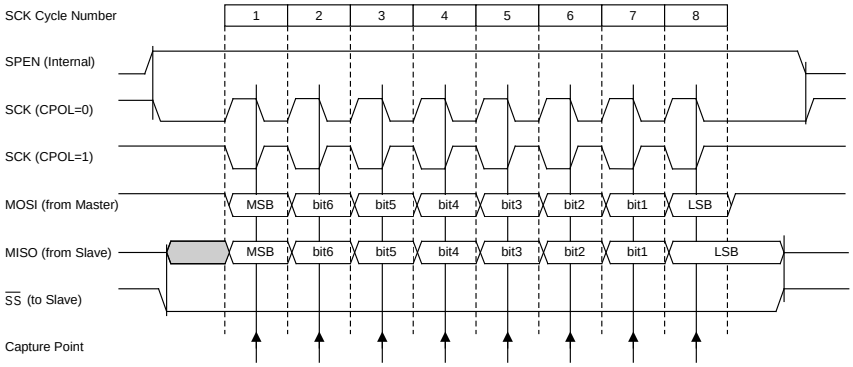
9.2.6 Transmission Formats

Software can select any of four combinations of serial clock (SCK) phase and polarity using two bits in the SPCON, the clock polarity CPOL and the clock phase CPHA. CPOL defines the default SCK line level in idle state. It has no significant effect on the transmission format. CPHA defines 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.



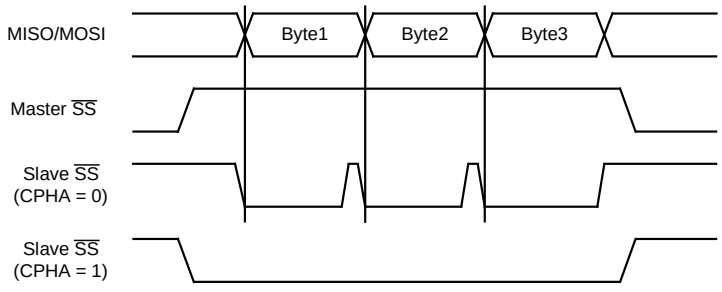
Data Transmission Format (CPHA = 0)

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 $\overline{SS}$ pin is used to start the transmission. The $\overline{SS}$ pin must be toggled high and then low between each byte transmitted. So SSDIS bit is invalid when CPHA = 0.



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 before the second edge of the first SCK. The $\overline{SS}$ pin can remain low between transmissions. This format may be preferred in systems with only one master and only one slave.



CPHA/$\overline{SS}$ Timing

Note: *Before SPI is configured as Slave mode and CPOL bit in SPCON is cleared, the P1.7/SCK pin must be set to input mode and enable pull-high resistor before SPEN bit in SPSTA is set to logic '1'.*

9.2.7 Error Conditions

The following flags in the SPSTA signal SPI error conditions:

(1) Mode Fault (MODF)

Mode fault error in master mode SPI indicates that the level on the $\overline{SS}$ 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:

- A 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 $\overline{SS}$ Disable (SSDIS bit in the SPCON register) is cleared, the MODF flag is set when the $\overline{SS}$ signal becomes '0'. However, as stated before, for a system with one Master, if the $\overline{SS}$ 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 $\overline{SS}$ pin as a general-purpose I/O pin.

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

(2) 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.

(3) Overrun Condition (RXOV)

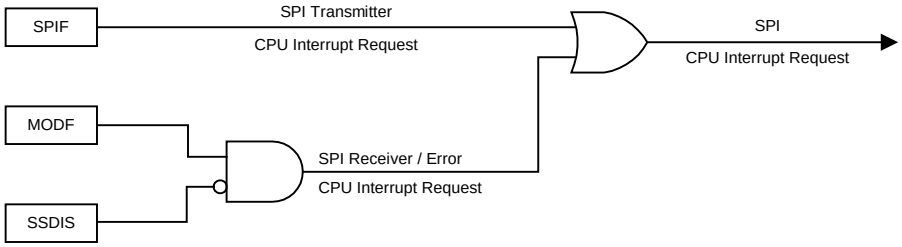
An overrun condition occurs when the master or slave tries to send several data bytes and the slave or master has not cleared the SPIF bit issuing from the previous data byte transmitted. In this case, the receive shift register keep the byte that SPIF was lastly set, also the SPI device will not receive 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 generate an interruption, the RXOV bit is cleared by software.

9.2.8 Interrupts

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

Serial Peripheral data transfer flag: SPIF. This bit is set by hardware when a transfer has been completed.

Mode Fault flag: MODF. This bit becomes set to indicate that the level on the $\overline{SS}$ 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.



9.2.9 Register

Table 9.4 Serial Peripheral Control Register

A2H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPCON	DIR	MSTR	CPHA	CPOL	SSDIS	SPR2	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**DIR**	**Transfer Direction Selection** 0: MSB first 1: LSB first
6	**MSTR**	**Serial Peripheral Master** 0: Configure the SPI as a Slave 1: Configure the SPI as a Master
5	**CPHA**	**Clock Phase** 0: Data sampled on first edge of SCK period 1: Data sampled on second edge of SCK period
4	**CPOL**	**Clock Polarity** 0: SCK line low in idle state 1: SCK line high in idle state
3	**SSDIS**	**$\overline{SS}$ Disable** 0: Enable $\overline{SS}$ pin in both Master and Slave modes 1: Disable $\overline{SS}$ 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.
2-0	**SPR[2:0]**	**Serial Peripheral Clock Rate** 000: $f_{SYS}/4$ 001: $f_{SYS}/8$ 010: $f_{SYS}/16$ 011: $f_{SYS}/32$ 100: $f_{SYS}/64$ Other: $f_{SYS}/128$

Table 9.5 SPI状态寄存器

F8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPSTA	SPEN	SPIF	MODF	WCOL	RXOV	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	-	-	-
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	-	-	-

Bit Number	Bit Mnemonic	Description
7	**SPEN**	**SPI Enable** 0: Disable the SPI interface 1: Enable the SPI interface
6	**SPIF**	**Serial Peripheral data transfer flag** 0: Clear by software 1: Set by hardware to indicate that the data transfer has been completed
5	**MODF**	**Mode Fault** 0: Cleared by software 1: Set by hardware to indicate that the $\overline{SS}$ pin is at inappropriate logic level
4	**WCOL**	**Write Collision flag** 0: Cleared by software to indicate write collision has bee processed 1: Set by hardware to indicate that a collision has been detected
3	**RXOV**	**Receive Overrun** 0: Cleared by software to indicate receive overrun has bee processed 1: Set by hardware to indicate that a receive overrun has been detected

Table 9.6 Serial Peripheral Data Register

A3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SPDAT	SPDAT7	SPDAT6	SPDAT5	SPDAT4	SPDAT3	SPDAT2	SPDAT1	SPDAT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**SPDAT[7:0]**	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.

***Note**: When SPI is disabled, the data of SPDAT is invalid.*

9.3 EUART

9.3.1 Feature

- SH88F516 has two enhanced EUART (EUART0/1) which are compatible with the conventional 8051
- The baud rate can be selected from the divided frequency of the system clock or Timer1/2 overflow rate, EUART1 itself has a baud-rate generator, EUART1 don't need to take up timer
- Enhancements over the standard 8051 the EUART include Framing Error detection and automatic address recognition
- The EUART can be operated in four modes

9.3.2 EUART0

The EUART0 can be operated in 4 modes. Users must initialize the SCON before any communication can take place. This involves selection of the Mode and the baud rate. The Timer1/2 should also be initialized if the mode 1 or the mode 3 is used.

In all of the 4 modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. This will generate a clock signal on the TXD pin and shift in 8 bits on the RxD pin. Reception is initiated in the other modes by the input start bit if REN = 1. The external transmitter will start the communication by transmitting the start bit.

EUART0 Mode Summary

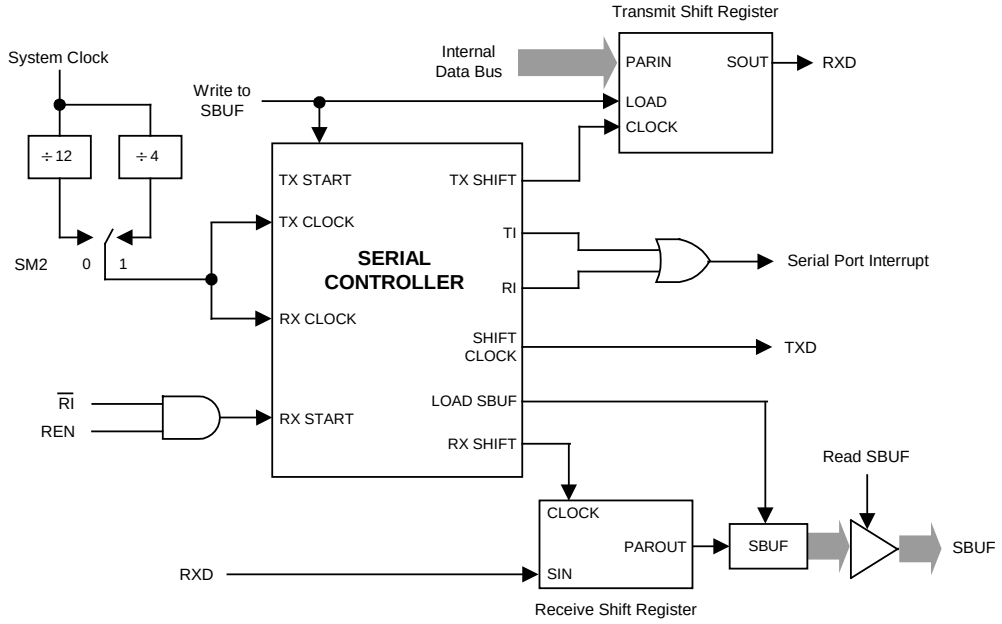
SM0	SM1	Mode	Type	Baud Clock	Frame Size	Start Bit	Stop Bit	9th bit
0	0	0	Synch	SYSCLK/(4 or 12)	8 bits	NO	NO	None
0	1	1	Asynch	Timer1 or 2 overflow rate/(16 or 32)	10 bits	1	1	None
1	0	2	Asynch	SYSCLK/(32 or 64)	11 bits	1	1	0,1
1	1	3	Asynch	Timer1 or 2 overflow rate/(16 or 32)	11 bits	1	1	0,1

Mode0: Synchronous Mode, Half duplex

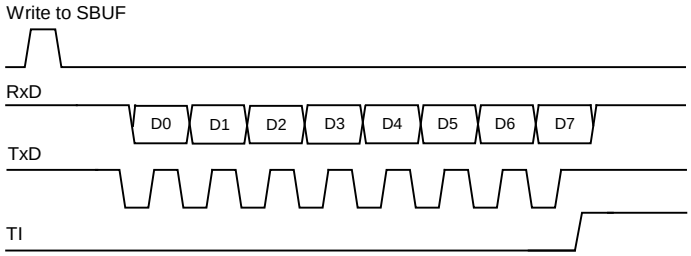
This mode provides synchronous communication with external devices. In this mode serial data is transmitted and received on the RxD line. TxD is used to output the shift clock. The TxD clock is provided by the SH88F516 whether the device is transmitting or receiving. This mode is therefore a half duplex mode of serial communication. In this mode, 8 bits are transmitted or received per frame. The LSB is transmitted/received first.

The baud rate is programmable to either 1/12 or 1/4 of the system clock. This baud rate is determined in the SM2 bit (SCON.5). When this bit is set to 0, the serial port runs at 1/12 of the system clock. When set to 1, the serial port runs at 1/4 of the system clock.

The functional block diagram is shown below. Data enters and exits the serial port on the RxD line. The TxD line is used to output the SHIFT CLOCK. The SHIFT CLOCK is used to shift data into and out of the SH88F516.

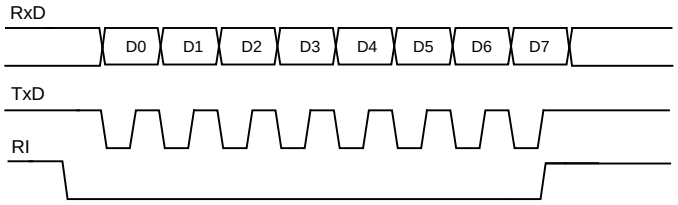


Any instruction that uses SBUF as a destination register (“write to SBUF” signal) will start the transmission. The next system clock tells the Tx control block to commence a transmission. The data shift occurs at the falling edge of the SHIFT CLOCK, and the contents of the transmit shift register is shifted one position to the right. As data bits shift to the right, zeros come in from the left. After transmission of all 8 bits in the transmit shift register, the Tx control block will deactivates SEND and sets TI (SCON.1) at the rising edge of the next system clock.



Send Timing of Mode 0

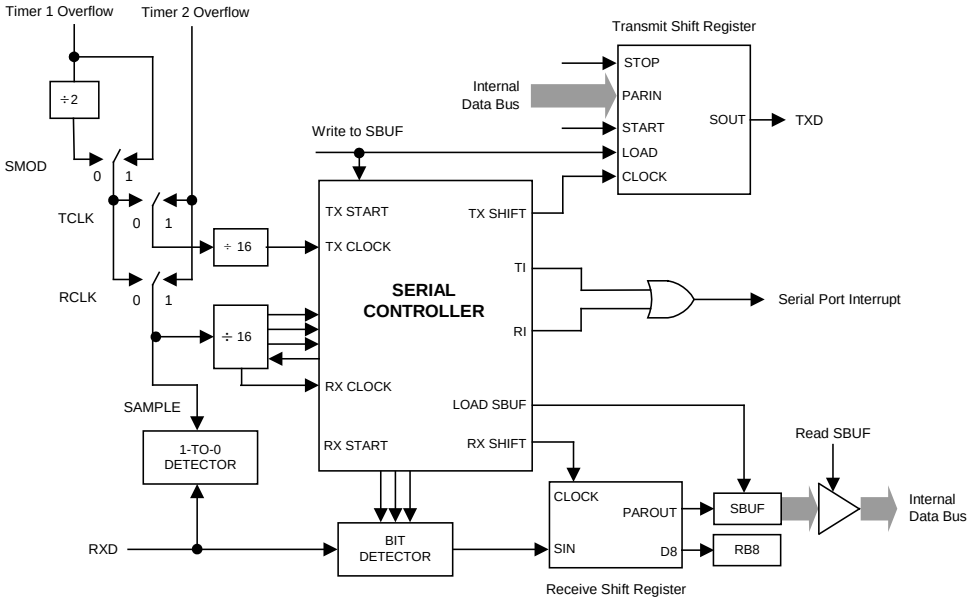
Reception is initiated by the condition REN (SCON.4) = 1 and RI (SCON.0) = 0. The next system clock activates RECEIVE. The data latch occurs at the rising edge of the SHIFT CLOCK, and the contents of the receive shift register are shifted one position to the left. After the receiving of all 8 bits into the receive shift register, the RX control block will deactivates RECEIVE and sets RI at the rising edge of the next system clock, and the reception will not be enabled till the RI is cleared by software.



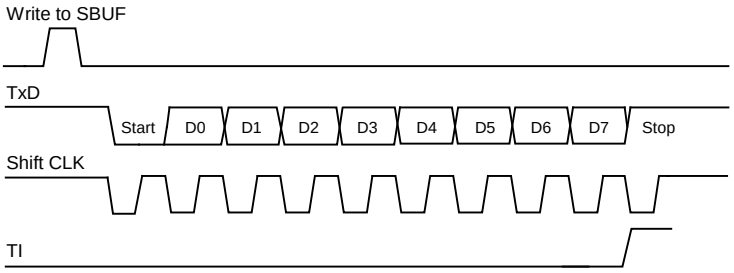
Receive Timing of Mode 0

Mode1: 8-Bit EUART, Variable Baud Rate, Asynchronous Full-Duplex

This mode provides the 10 bits full duplex asynchronous communication. The 10 bits consist of a start bit (logical 0), 8 data bits (LSB first), and a stop bit (logical 1). When receiving, the eight data bits are stored in SBUF and the stop bit goes into RB8 (SCON.2). The baud rate in this mode is variable. The serial receive and transmit baud rate can be programmed to be 1/16 of the Timer1/2 overflow (Refer to **Baud Rate** Section for details). The functional block diagram is shown below.



Transmission begins with a “write to SBUF” signal, and it actually commences at the next system clock following the next rollover in the divide-by-16 counter (divide baud-rate by 16), thus, the bit times are synchronized to the divide-by-16 counter, not to the “write to SUBF” signal. The start bit is firstly put out on TxD pin, then are the 8 bits of data. After all 8 bits of data in the transmit shift register are transmitted, the stop bit is put out on the TxD pin, and the TI flag is set at the same time that the stop is send.



Send Timing of Mode 1

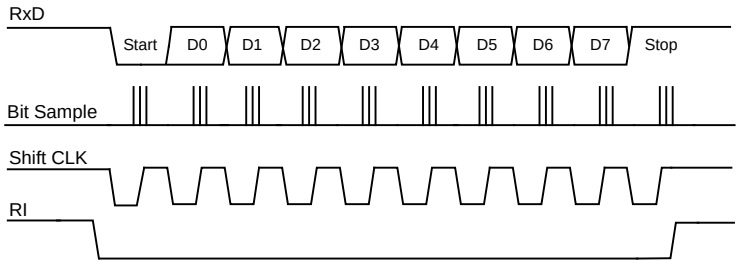
Reception is enabled only if REN is high. The serial port actually starts the receiving of serial data with the detection of a falling edge on the RxD pin. For this purpose RxD is sampled at the rate of 16 times baud rate. When a falling edge is detected, the divide-by-16 counter is immediately reset. This helps to align the bit boundaries with the rollovers of the divide-by-16 counter.The 16 states of the counter divide each bit time into 16ths. The bit detector samples the value of RxD at the 7^{th}, 8^{th} and 9^{th} counter states of each bit time. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the first bit after the falling edge of RxD pin is not 0, which indicates an invalid start bit, and the reception is immediately aborted. The receive circuits are reset and again waiting for a falling edge in the RxD line. If a valid start bit is detected, then the rest of the bits are also detected and shifted into the shift register. After shifting in 8 data bits and the stop bit, the SBUF and RB8 are loaded and RI are set if the following conditions are met:

1. RI must be 0

2. Either SM2 = 0, or the received stop bit = 1

If these conditions are met, then the stop bit goes to RB8, the 8 data bits go into SBUF and RI is set. Otherwise the received frame may be lost.

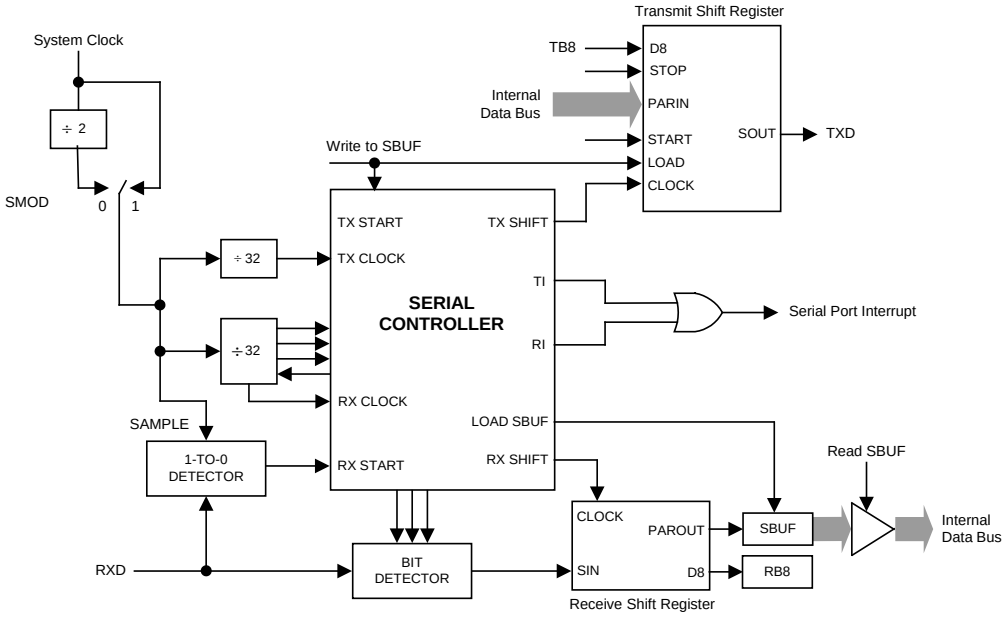
At the time, the receiver goes back to looking for another falling edge on the RxD pin. And the user should clear RI by software for further reception.



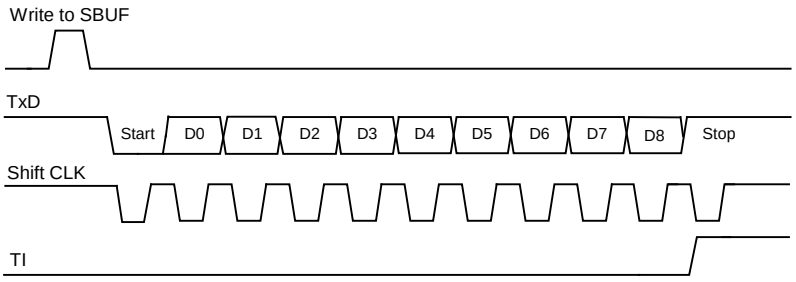
Receive Timing of Mode 1

Mode2: 9-Bit EUART, Fixed Baud Rate, Asynchronous Full-Duplex

This mode provides the 11 bits full duplex asynchronous communication. The 11 bit consists of one start bit (logical 0), 8 data bits (LSB first), a programmable 9th data bit, and a stop bit (logical 1). Mode2 supports multiprocessor communications and hardware address recognition (Refer to Multiprocessor Communication Section for details). When data is transmitted, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1, for example, the parity bit P in the PSW or used as data/address flag in multiprocessor communications. When data is received, the 9th data bit goes into RB8 and the stop bit is not saved. The baud rate is programmable to either 1/32 or 1/64 of the system working frequency, as determined by the SMOD bit in PCON. The functional block diagram is shown below.



Transmission begins with a “write to SBUF” signal, the “write to SBUF” signal also loads TB8 into the 9th bit position of the transmit shift register. Transmission actually commences at the next system clock following the next rollover in the divide-by-16 counter (thus, the bit times are synchronized to the divide-by-16 counter, not to the “write to SUBF” signal). The start bit is firstly put out on TxD pin, then are the 9 bits of data. After all 9 bits of data in the transmit shift register are transmitted, the stop bit is put out on the TxD pin, and the TI flag is set at the same time, this will be at the 11th rollover of the divide-by-16 counter after a write to SBUF.



Send Timing of Mode 2

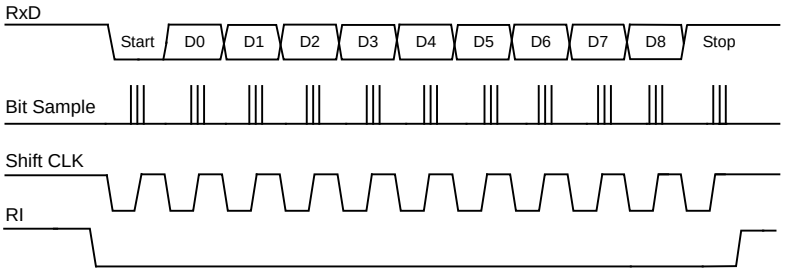
Reception is enabled only if REN is high. The serial port actually starts the receiving of serial data, with the detection of a falling edge on the RxD pin. For this purpose RxD is sampled at the rate of 16 times baud rate. When a falling edge is detected, the divide-by-16 counter is immediately reset. This helps to align the bit boundaries with the rollovers of the divide-by-16 counter. The 16 states of the counter divide each bit time into 16ths. The bit detector samples the value of RxD at the 7th, 8th and 9th counter state of each bit time. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done for noise rejection. If the first bit detected after the falling edge of RxD pin is not 0, which indicates an invalid start bit, and the reception is immediately aborted. The receive circuits are reset and again looks for a falling edge in the RxD line. If a valid start bit is detected, then the rest of the bits are also detected and shifted into the shift register. After shifting in 9 data bits and the stop bit, the SBUF and RB8 are loaded and RI is set if the following conditions are met:

1. RI must be 0

2. Either SM2 = 0, or the received 9th bit = 1 and the received byte accords with Given Address

If these conditions are met, then the 9th bit goes to RB8, the 8 data bits go into SBUF and RI is set. Otherwise the received frame may be lost.

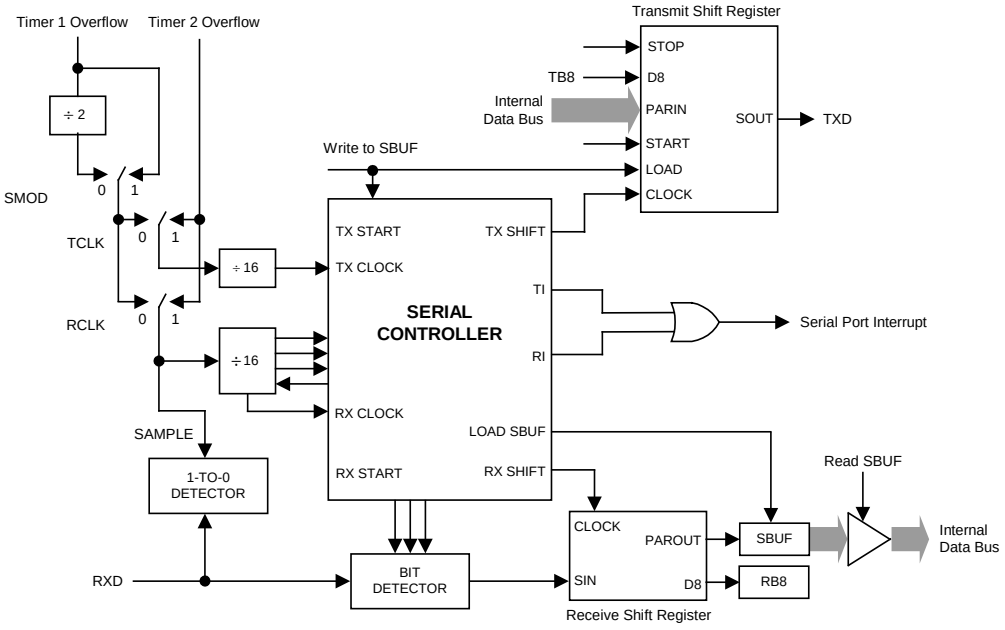
At the time, the receiver goes back to looking for another falling edge on the RxD pin. And the user should clear RI by software for further reception.



Receive Timing of Mode 2

Mode3: 9-Bit EUART, Variable Baud Rate, Asynchronous Full-Duplex

Mode3 uses transmission protocol of the Mode 2 and baud rate generation of the Mode1.



Baud Rate Generate

In Mode0, the baud rate is programmable to either 1/12 or 1/4 of the system frequency. This baud rate is determined by SM2 bit. When set to 0, the serial port runs at 1/12 of the system clock. When set to 1, the serial port runs at 1/4 of the system clock.

In Mode1 & Mode3, the baud rate can be selected from Timer1/2 overflow rate.

The Mode1 & 3 baud rate equations are shown below, where [RCAP2H, RCAP2L] is the 16-bit reload register for Timer2, SMOD is the EUART baud rate doubler (PCON.7), T1CLK is the clock source of Timer1. T2CLK is the clock source of Timer2.

$BaudRate = \frac{2^{SMOD}}{32} \times \frac{f_{T1}}{256 - TH1}$, Baud Rate using Timer1, working in Mode2.

$BaudRate = \frac{1}{2 \times 16} \times \frac{f_{T2}}{65536 - [RCAP2H, RCAP2L]}$, Baud Rate using Timer2, the clock source of Timer2 is system clock.

$BaudRate = \frac{1}{16} \times \frac{f_{T2}}{65536 - [RCAP2H, RCAP2L]}$, Baud Rate using Timer2, the clock source of Timer2 is input clock of T2 pin

In Mode2, the baud rate is programmable to either 1/32 or 1/64 of the system clock. This baud rate is determined by the SMOD bit (PCON.7). When this bit is set to 0, the serial port runs at 1/64 of the clock. When set to 1, the serial port runs at 1/32 of the clock.

$$\text{BaudRate} = 2^{\text{SMOD}} \times (\frac{f_{SYS}}{64})$$

Multi-Processor Communication

Software Address Recognition

Modes 2 and 3 of the EUART have a special provision for multi-processor communication. In these modes, 9 data bits are received. The 9th bit goes into RB8. Then a stop bit follows. The EUART can be programmed such that when the stop bit is received, the EUART interrupt will be activated (i.e. the request flag RI is set) only if RB8 = 1. This feature is enabled by setting the bit SM2 in SCON.

A way to use this feature in multiprocessor communications is as follows. If the master processor wants to transmit a block of data to one of the several slaves, it first sends out an address byte which identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte.

With SM2 = 1, no other slave will be interrupted by a data byte. An address byte, however, will interrupt all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave will clear its SM2 bit and prepare to receive the data bytes that will be coming. After having received a complete message, the slave sets SM2 again. The slaves that were not addressed leave their SM2 set and go on with their business, ignoring the incoming data bytes.

Note: *In mode 0, SM2 is used to select baud rate doubling. In mode 1, SM2 can be used to check the validity of the stop bit. If SM2 = 1 in mode 1, the receive interrupt will not be activated unless a valid stop bit is received.*

Automatic (Hardware) Address Recognition

In Mode2 & 3, setting the SM2 bit will configure EUART act as following: when a stop bit is received, EUART will generate an interrupt only if the 9th bit that goes into RB8 is logic 1 (address byte) and the received data byte matches the EUART slave address. Following the received address interrupt, the slave should clear its SM2 bit to enable interrupts on the reception of the following data byte(s).

The 9-bit mode requires that the 9th information bit is a 1 to indicate that the received information is an address and not data. When the master processor wants to transmit a block of data to one of the slaves, it first sends out the address of the targeted slave (or slaves). All the slave processors should have their SM2 bit set high when waiting for an address byte, which ensures that they will be interrupted only by the reception of an address byte. The Automatic address recognition feature further ensures that only the addressed slave will be interrupted. The address comparison is done by hardware not software.

After being interrupted, the addressed slave clears the SM2 bit to receive data bytes. The un-addressed slaves will be unaffected, as they will be still waiting for their address. Once the entire message is received, the addressed slave should set its SM2 bit to ignore all transmissions until it receives the next address byte.

The Automatic Address Recognition feature allows a master to selectively communicate with one or more slaves by invoking the Given Address. All of the slaves may be contacted by using the Broadcast address. Two special Function Registers are used to define the slave's address, SADDR, and the address mask, SADEN. The slave address is an 8-bit value specified in the SADDR register. The SADEN register is actually a mask for the byte value in SADDR. If a bit position in SADEN is 0, then the corresponding bit position in SADDR is don't care. Only those bit positions in SADDR whose corresponding bits in SADEN are 1 are used to obtain the Given Address. This gives the user flexibility to address multiple slaves without changing the slave address in SADDR. Use of the Given Address allows multiple slaves to be recognized while excluding others.

	Slave 1	Slave 2
SADDR	10100100	10100111
SADEN (0 mask)	11111010	11111001
Given Address	10100x0x	10100xx1
Broadcast Address (OR)	1111111x	11111111

The Given address for slave 1 and 2 differ in the LSB. For slave 1, it is a don't care, while for slave 2 it is 1. Thus to communicate only with slave 1, the master must send an address with LSB = 0 (10100000). Similarly the bit 1 is 0 for slave 1 and don't care for slave 2. Hence to communicate only with slave 2 the master has to transmit an address with bit 1 = 1 (1010 0011). If the master wishes to communicate with both slaves simultaneously, then the address must have bit 0 = 1 and bit 1 = 0. The bit 2 position is don't care for both the slaves. This allows two different addresses to select both slaves (1010 0001 and 1010 0101).

The master can communicate with all the slaves simultaneously with the Broadcast Address. This address is formed from the logical OR of the SADDR and SADEN. The zeros in the result are defined as don't cares. In most cases, the Broadcast Address is FFh, this address will be acknowledged by all slaves.

On reset, the SADDR and SADEN are initialized to 00h. This results in Given Address and Broadcast Address being set as XXXXXXXX (all bits don't care). This effectively removes the multiprocessor communications feature, since any selectivity is disabled. This ensures that the EUART 0 will reply to any address, which it is backwards compatible with the 80C51 microcontrollers that do not support automatic address recognition. So the user may implement multiprocessor by software address recognition mentioned above.

Error Detection

Error detection is available when the SSTAT bit in register PCON is set to logic 1.The SSTAT bit must be logic 1 to access any of the status bits (FE, RXOV, and TXCOL). The SSTAT bit must be logic 0 to access the Mode Select bits (SM0, SM1, and SM2). All the 3 bits should be cleared by software after they are set, even when the following frames received without any error will not be cleared automatically.

Transmit Collision

The Transmit Collision bit (TXCOL bit in register SCON) reads '1' if RI is set 0 and user software writes data to the SBUF register while a transmission is still in progress. If this occurs, the new data will be ignored and the transmit buffer will not be written.

Receive Overrun

The Receive Overrun bit (RXOV in register SCON) reads '1' if a new data byte is latched into the receive buffer before software has read the previous byte. The previous data is lost when this happen.

Frame Error

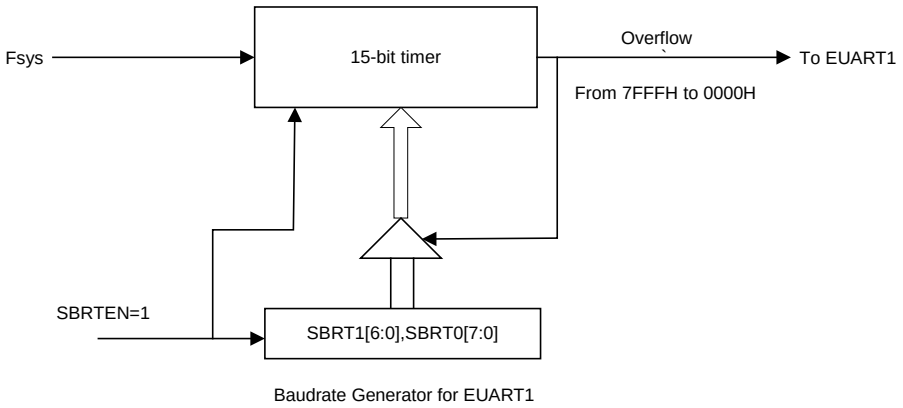
The Frame Error bit (FE in register SCON) reads '1' if an invalid (low) STOP bit is detected.

Break Detection

A break is detected when any 11 consecutive bits are sensed low. Since a break condition also satisfies the requirements for a framing error, a break condition will also result in reporting a framing error. Once a break condition has been detected, the EUART will go into an idle state and remain in this idle state until a valid stop bit (rising edge on RxD line) has been received.

9.3.3 EUART1

The control and operation mode of EUART0 are similar to EUART0.The difference is that EUART1 has a baud-rate generator In other words, EUART1 can't use Timer1/2 as baud-rate generator.Actually, the baud-rate generator of EUART1 is a 15-bit up counter.



Baudrate Generator for EUART1

$SBRToverflowrate = \frac{Fsys}{32768 - SBRT}$ ， $SBRT = [SBRT1, SBRT0]$ 。

Mode0:

The operation mode of EUART1 is similar to EUART0, the baud rate is programmable to either 1/12 or 1/4 of the system frequency. This baud rate is determined by SM12 bit. When set to 0, the serial port runs at 1/12 of the system clock. When set to 1, the serial port runs at 1/4 of the system clock.

Mode1:

The operation mode of EUART1 is similar to EUART0, baud-rate equation is shown below:

$$BaudRate = \frac{SBRToverflowrate}{16}$$

Mode2:

The operation mode of EUART1 is similar to EUART0, baud-rate equation is shown below:

$BaudRate = 2^{SMOD} \times \frac{Fsys}{64}$, (SMOD is PCON.7)

Mode3:

The operation mode of EUART1 is similar to EUART0, baud-rate equation is shown below:

$$BaudRate = \frac{SBRToverflowrate}{16}$$

9.3.4 Register

Table 9.7 Power Control Register

87H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD	SSTAT	SSTAT1	-	GF1	GF0	PD	IDL
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**SMOD**	**Baud rate doubler** If set in Mode1 & 3, the baud-rate of EUART0 is doubled if using Timer1 as baud-rate generator If set in Mode2, the baud-rate of EUART is doubled
6	**SSTAT**	**SCON[7:5] function select bit** 0: SCON[7:5] operates as SM0, SM1, SM2 1: SCON[7:5] operates as FE, RXOV, TXCOL
5	**SSTAT1**	**SCON1[7:5] function select bit** 0: SCON1[7:5] operates as SM10, SM11, SM12 1: SCON1[7:5] operates as FE1, RXOV1, TXCOL1
3-2	**GF[1:0]**	**General purpose flags for software use**
1	**PD**	**Power-Down mode control bit**
0	**IDL**	**Idle mode control bit**

Table 9.8 EUART0 Control & Status Register

98H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON	SM0 /FE	SM1 /RXOV	SM2 /TXCOL	REN	TB8	RB8	TI	RI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-6	**SM[0:1]**	**EUART Serial mode control bit, when SSTAT = 0** 00: mode 0, Synchronous Mode, fixed baud rate 01: mode 1, 8 bit Asynchronous Mode, variable baud rate 10: mode 2, 9 bit Asynchronous Mode, fixed baud rate 11: mode 3, 9 bit Asynchronous Mode, variable baud rate
7	**FE**	**EUART Frame Error flag, when FE bit is read, SSTAT bit must be set 1** 0: No Frame Error, clear by software 1: Frame error occurs, set by hardware
6	**RXOV**	**EUART Receive Over flag, when RXOV bit is read, SSTAT bit must be set 1** 0: No Receive Over, clear by software 1: Receive over occurs, set by hardware
5	**SM2**	**EUART Multi-processor communication enable bit (9th bit '1' checker), when SSTAT = 0** 0: In Mode0, baud-rate is 1/12 of system clock In Mode1, disable stop bit validation check, any stop bit will set RI to generate interrupt In Mode2 & 3, any byte will set RI to generate interrupt 1: In Mode0, baud-rate is 1/4 of system clock In Mode1, Enable stop bit validation check, only valid stop bit (1) will set RI to generate interrupt In Mode2 & 3, only address byte (9th bit = 1) will set RI to generate interrupt
5	**TXCOL**	**EUART Transmit Collision flag, when TXCOL bit is read, SSTAT bit must be set 1** 0: No Transmit Collision, clear by software 1: Transmit Collision occurs, set by hardware
4	**REN**	**EUART Receiver enable bit** 0: Receive Disable 1: Receive Enable
3	**TB8**	**The 9th bit to be transmitted in Mode2 & 3 of EUART, set or clear by software**
2	**RB8**	**The 9th bit to be received in Mode1, 2 & 3 of EUART** In Mode0, RB8 is not used In Mode1, if receive interrupt occurs, RB8 is the stop bit that was received In Modes2 & 3 it is the 9th bit that was received
1	**TI**	**Transmit interrupt flag of EUART0** 0: cleared by software 1: Set by hardware at the end of the 8th bit time in Mode0, or at the beginning of the stop bit in other modes
0	**RI**	**Receive interrupt flag of EUART0** 0: cleared by software 1: Set by hardware at the end of the 8th bit time in Mode0, or during the stop bit time in other modes

Table 9.9 EUART0 Data Buffer Register

99H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SBUF	SBUF.7	SBUF.6	SBUF.5	SBUF.4	SBUF.3	SBUF.2	SBUF.1	SBUF.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**SBUF[7-0]**	This SFR accesses two registers; a transmit shift register and a receive latch register A write of SBUF will send the byte to the transmit shift register and then initiate a transmission A read of SBUF returns the contents of the receive latch

Table 9.10 EUART0 Slave Address & Address Mask Register

9AH-9BH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SADDR	SADDR.7	SADDR.6	SADDR.5	SADDR.4	SADDR.3	SADDR.2	SADDR.1	SADDR.0
SADEN	SADEN.7	SADEN.6	SADEN.5	SADEN.4	SADEN.3	SADEN.2	SADEN.1	SADEN.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**SADDR[7:0]**	**SFR SADDR defines the EUART0's slave address**
7-0	**SADEN[7:0]**	**SFR SADEN is a bit mask to determine which bits of SADDR are checked against a received address** 0: Corresponding bit in SADDR is a "don't care" 1: Corresponding bit in SADDR is checked against a received address

Table 9.11 EUART1 Control & Status Register

E8H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SCON1	SM10 /FE	SM11 /RXOV	SM12 /TXCOL	REN	TB8	RB8	TI	RI
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-6	**SM1[0:1]**	**EUART1 Serial mode control bit, when SSTAT = 0** 00: mode 0, Synchronous Mode, fixed baud rate 01: mode 1, 8 bit Asynchronous Mode, variable baud rate 10: mode 2, 9 bit Asynchronous Mode, fixed baud rate 11: mode 3, 9 bit Asynchronous Mode, variable baud rate
7	**FE1**	**EUART1 Frame Error flag, when FE1 bit is read, SSTAT1 bit must be set 1** 0: No Frame Error, clear by software 1: Frame error occurs, set by hardware
6	**RXOV1**	**EUART1 Receive Over flag, when RXOV1 bit is read, SSTAT1 bit must be set 1** 0: No Receive Over, clear by software 1: Receive over occurs, set by hardware
5	**SM12**	**EUART1 Multi-processor communication enable bit (9th bit '1' checker), when SSTAT1 = 0** 0: In Mode0, baud-rate is 1/12 of system clock In Mode1, disable stop bit validation check, any stop bit will set RI1 to generate interrupt In Mode2 & 3, any byte will set RI1 to generate interrupt 1: In Mode0, baud-rate is 1/4 of system clock In Mode1, Enable stop bit validation check, only valid stop bit (1) will set RI1 to generate interrupt In Mode2 & 3, only address byte (9th bit = 1) will set RI1 to generate interrupt
5	**TXCOL1**	**EUART1 Transmit Collision flag, when TXCOL1 bit is read, SSTAT1 bit must be set 1** 0: No Transmit Collision, clear by software 1: Transmit Collision occurs, set by hardware
4	**REN1**	**EUART1 Receiver enable bit** 0: Receive Disable 1: Receive Enable
3	**TB18**	**The 9th bit to be transmitted in Mode2 & 3 of EUART1, set or clear by software**
2	**RB18**	**The 9th bit to be received in Mode1, 2 & 3 of EUART1** In Mode0, RB8 is not used In Mode1, if receive interrupt occurs, RB8 is the stop bit that was received In Modes2 & 3 it is the 9th bit that was received
1	**TI1**	**Transmit interrupt flag of EUART1** 0: cleared by software 1: Set by hardware at the end of the 8th bit time in Mode0, or at the beginning of the stop bit in other modes
0	**RI1**	**Receive interrupt flag of EUART1** 0: cleared by software 1: Set by hardware at the end of the 8th bit time in Mode0, or during the stop bit time in other modes

Table 9.12 EUART1 Data Buffer Register

9DH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SBUF1	SBUF1.7	SBUF1.6	SBUF1.5	SBUF1.4	SBUF1.3	SBUF1.2	SBUF1.1	SBUF1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**SBUF1[7-0]**	This SFR accesses two registers; a transmit shift register and a receive latch register A write of SBUF1 will send the byte to the transmit shift register and then initiate a transmission A read of SBUF1 returns the contents of the receive latch

Table 9.13 EUART1 Slave Address & Address Mask Register

9EH-9FH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SADDR1	SADDR1.7	SADDR1.6	SADDR1.5	SADDR1.4	SADDR1.3	SADDR1.2	SADDR1.1	SADDR1.0
SADEN1	SADEN1.7	SADEN1.6	SADEN1.5	SADEN1.4	SADEN1.3	SADEN1.2	SADEN1.1	SADEN1.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**SADDR1[7:0]**	**SFR SADDR defines the EUART1's slave address**
7-0	**SADEN1[7:0]**	**SFR SADEN1 is a bit mask to determine which bits of SADDR1 are checked against a received address** 0: Corresponding bit in SADDR1 is a "don't care" 1: Corresponding bit in SADDR1 is checked against a received address

Table 9.14 EUART1 Baudrate generator Register

A4H,9CH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SBRT1	SBRTEN	SBRT1.6	SBRT1.5	SBRT1.4	SBRT1.3	SBRT1.2	SBRT1.1	SBRT1.0
SBRT0	SBRT0.7	SBRT0.6	SBRT0.5	SBRT0.4	SBRT0.3	SBRT0.2	SBRT0.1	SBRT0.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**SBRTEN**	**EUART Baudrate generator control bit** 0: disable (default) 1: enable
6-0	**SBRT1.6:0**	**High 7-bit of EUART1 baud-rate generator counter**
7-0	**SBRT0.7:0**	**Low 8-bit of EUART1 baud-rate generator counter**

9.4 Analog Digital Converter (ADC)

9.4.1 Feature

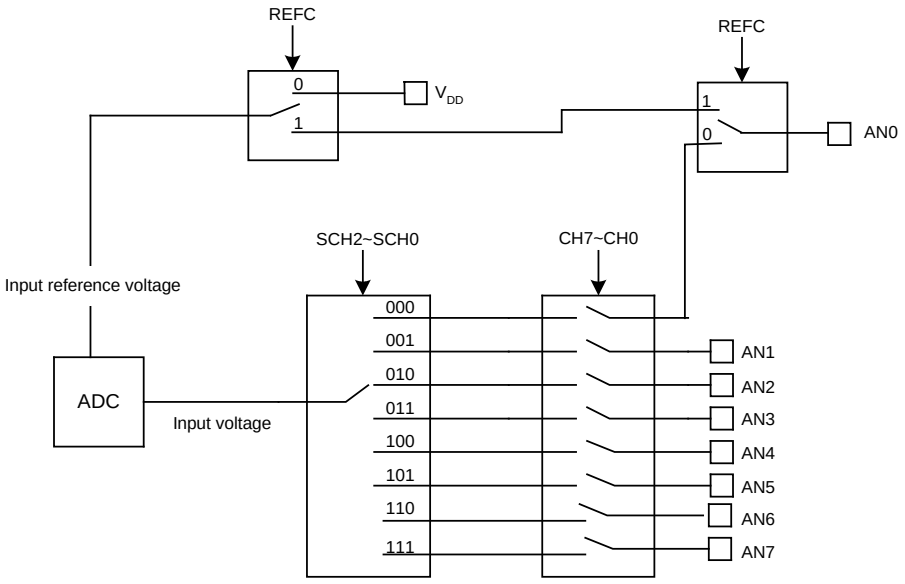
- 10-bit Resolution
- Build in V_{REF}
- Selectable external or built-in V_{REF}
- 8 Multiplexed Input Channels

The SH88F516 includes a single ended, 10-bit SAR Analog to Digital Converter (ADC) with build in reference voltage connected to the V_{DD}, users also can select the AVREF port input reference voltage. The 8 ADC channels are shared with 1 ADC module; each channel can be programmed to connect with the analog input individually. Only one channel can be available at one time. GO/$\overline{DONE}$ signal is available to start convert, and indicate end of convert. When conversion is completed, the data in AD convert data register will be updated and ADCIF bit in ADCON register will be set. If ADC Interrupt is enabled, the ADC interrupt will be generated.

The ADC integrates a digital compare function to compare the value of analog input with the digital value in the AD converter. If this function is enabled (set EC bit in ADCON register) and ADC module is enabled (set ADON bit in ADCON register). When the corresponding digital value of analog input is larger than the value in compare value register (ADDH/L), the ADC interrupt will occur, otherwise no interrupt will be generated. The digital comparator can work continuously when GO/$\overline{DONE}$ bit is set until software clear, which behaviors different with the AD converter operation mode.

The ADC module including digital compare module can wok in Idle mode and the ADC interrupt will wake up the Idle mode, but is disabled in Power-Down mode.

9.4.2 ADC Diagram



ADC Diagram

9.4.3 ADC Register

Table 9.15 ADC Control Register

93H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCON	ADON	ADCIF	EC	REFC	SCH2	SCH1	SCH0	$\overline{\text{GO/DONE}}$
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**ADON**	**ADC Enable bit** 0: Disable the ADC module 1: Enable the ADC module
6	**ADCIF**	**ADC Interrupt Flag bit** 0: No ADC interrupt, cleared by software. 1: Set by hardware to indicate that the AD Convert has been completed, or analog input is larger than ADDH/ADDL if compare is enabled
5	**EC**	**Compare Function Enable bit** 0: Compare function disabled 1: Compare function enabled
4	**REFC**	**Reference Voltage Select bit** 0: the reference voltage connected to V_{DD} 1: the reference voltage input from V_{REF} pin
3-1	**SCH[2:0]**	**ADC Channel Select bits** 000: ADC channel AN0(Default) 001: ADC channel AN1 010: ADC channel AN2 011: ADC channel AN3 100: ADC channel AN4 101: ADC channel AN5 110: ADC channel AN6 111: ADC channel AN7
0	**$\overline{\text{GO/DONE}}$**	**ADC Status Flag bit** 0: Automatically cleared by hardware when AD convert is completed. Clearing this bit during converting time will stop current conversion. If Compare function is enabled, this bit will not be cleared by hardware until software clear. 1: Set to start AD convert or digital compare.

Notes:

When select the reference voltage input from V_{REF} pin (REFC = 1), the P0.0 is shared as V_{REF} input rather than AN0 input.

Table 9.16 ADC Clock Configure Register

94H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADT	TADC2	TADC1	TADC0	-	TS3	TS2	TS1	TS0
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-5	**TADC[2:0]**	**ADC Clock Period Select bits** 000: ADC Clock Period t_{AD} = 2 t_{SYS} 001: ADC Clock Period t_{AD} = 4 t_{SYS} 010: ADC Clock Period t_{AD} = 6 t_{SYS} 011: ADC Clock Period t_{AD} = 8 t_{SYS} 100: ADC Clock Period t_{AD} = 12 t_{SYS} 101: ADC Clock Period t_{AD} = 16 t_{SYS} 110: ADC Clock Period t_{AD} = 24 t_{SYS} 111: ADC Clock Period t_{AD} = 32 t_{SYS}
3-0	**TS[3:0]**	**Sample time select bits** 2 t_{AD} ≤ Sample time = (TS[3:0]+1) * t_{AD} ≤ 15 t_{AD}

Note:

(1) Make sure that t_{AD} ≧ 1μs;

(2) The minimum sample time is 2 t_{AD}, even TS[3:0] = 0000;

(3) The maximum sample time is 15 t_{AD} , even TS[3:0] = 1111;

(4) Evaluate the series resistance connected with ADC input pin before set TS[3:0];

(5) Be sure that the series resistance connected with ADC input pin is no more than 10kΩ when 2 t_{AD} sample time is selected;

(6) Total conversion time is: 12 t_{AD} + sample time.

For Example

System Clock (SYSCLK)	TADC[2:0]	t_{AD}	TS[3:0]	Sample Time	Conversion Time
32.768kHz	000	30.5*2=61μs	0000	2*61=122μs	12*61+122=854μs
	000	30.5*2=61μs	0111	8*61=488μs	12*61+488=1220μs
	000	30.5*2=61μs	1111	15*61=915μs	12*61+915=1647μs
	111	30.5*32=976μs	0000	2*976=1952μs	12*976+1952=13664μs
	111	30.5*32=976μs	0111	8*976=7808μs	12*976+7808=19520μs
	111	30.5*32=976μs	1111	15*976=14640μs	12*976+14640=26352μs
12MHz	000	0.083*2=0.166μs	-	-	(t_{AD} < 1μs, not recommended)
	100	0.083*12=1μs	0000	2*1=2μs	12*1+2=14μs
	100	0.083*12=1μs	0111	8*1=8μs	12*1+8=20μs
	100	0.083*12=1μs	1111	15*1=15μs	12*1+15=27μs
	111	0.083*32=2.7μs	0000	2*2.7=5.4μs	12*2.7+5.4=37.8μs
	111	0.083*32=2.7μs	0111	8*2.7=21.6μs	12*2.7+21.6=54μs
	111	0.083*32=2.7μs	1111	15*2.7=40.5μs	12*2.7+40.5=72.9μs

Table 9.17 ADC Channel Configure Register

95H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCH	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN))	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**CH[7:0]**	**Channel Configuration bits** 0: P0.x (x = 0 - 7) are I/O port or other function 1: P0.x (x = 0 - 7) are ADC input port (When P0x isn't configured as other function)

Table 9.18 AD Converter Data Register (Compare Value Register)

96H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADDL	-	-	-	-	-	-	A1	A0
R/W	-	-	-	-	-	-	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	-	-	-	-	-	-	0	0
97H	**Bit7**	**Bit6**	**Bit5**	**Bit4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
ADDH	A9	A8	A7	A6	A5	A4	A3	A2
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
1-0 **7-0**	**A9-A0**	**ADC Data register** Digital Value of sampled analog voltage, updated when conversion is completed If ADC Compare function is enabled (EC = 1), the value will be compared with the analog input

The Approach for AD Conversion:

(1) Select the analog input channels and reference voltage.
(2) Enable the ADC module with the selected analog channel.
(3) Set GO/$\overline{DONE}$ = 1 to start the AD conversion.
(4) Wait until GO/$\overline{DONE}$ = 0 or ADCIF = 1, if the ADC interrupt is enabled, the ADC interrupt will occur, user need clear ADCIF by software.
(5) Acquire the converted data from ADDH/ADDL.
(6) Repeat step 3-5 if another conversion is required.

The Approach for Digital Compare Function:

(1) Select the analog input channels and reference voltage.
(2) Write ADDH/ADDL to set the compare value.
(3) Set EC = 1 to enable compare function.
(4) Enable the ADC module with the selected analog channel.
(5) Set GO/$\overline{DONE}$ = 1 to start the compare function.
(6) If the analog input is lager than compare value set in ADDH/ADDL, the ADCIF will be set to 1. If the ADC interrupt is enabled, the ADC interrupt will occur, user need clear ADCIF by software.
(7) The compare function will continue work until the GO/$\overline{DONE}$ bit is cleared to 0.

9.5 Low Power Detect (LPD)

9.5.1 Feature

- Low power detect and generate interrupt
- LPD detect voltage is selectable

The low power detect (LPD) is used to monitor the supply voltage and generate an internal flag if the voltage decrease below the specified value. It is used to inform CPU whether the power is shut off or the battery is used out, so the software may do some protection action before the voltage drop down to the minimal operation voltage.

9.5.2 Register

Table 9.19 Low Power Detection Control Register

B3H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LPDCON	LPDEN	LPDF*	LPDV	-	-	-	LPDS1	LPDS0
R/W	R/W	R/W*	R/W	-	-	-	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	-	-	-	0	0

*: LPDF can only be cleared, it can't be set.

Bit Number	Bit Mnemonic	Description
7	**LPDEN**	**LPD Enable bit** 0: Disable lower power detection 1: Enable lower power detection
6	**LPDF**	**LPD status Flag bit** 0: No LPD happened, clear by hardware 1: LPD happened, set by hardware
5	**LPDV**	**LPD Detect Voltage source** 0: Detect supply voltage 1: Detect VLPD (P4.7) pin voltage
1-0	**LPDS[1:0]**	**LPD Voltage Select bit** 00: 3.7V 01: 3.9V 10: 4.2V 11: 4.4V

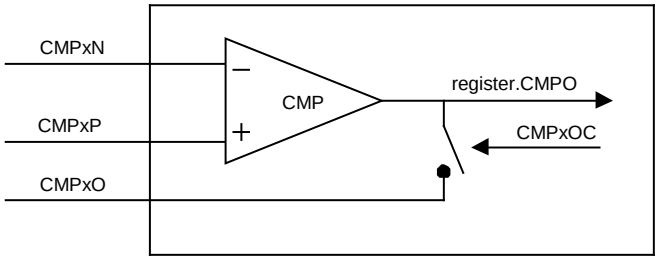
9.6 Comparator (CMP)

9.6.1 Feature

- Single power operation
- Output positive/negative control
- Work in Idle or Power-Down mode

SH88F516 consists of two independent precision voltage comparators. The CMPxP pin is the positive input of the Comparator. The CMPxN pin is the negative input of the Comparator. The CMPxO pin is the output of the Comparator, and it can be changed as the normal I/O port or comparator output pin under the condition of the comparator being enabled.

If CMPEN = 1 and CMPIE = 1, any change on the output value of the Comparator would generate an interrupt request (CMPIF = 1). The Comparator interrupt can also wake the CPU from IDLE or Power-Down mode.



Built-in CMP

Note: *It will take 2ms for the first start of comparator;Users need to clear CMPxIF before using comparator for the first time.*

9.6.2 Register

Table 9.20 CMP Control Register

91H,92H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CMPCONx (x=0,1)	CMPxEN	CMPxIF	-	-	-	CMPxOC	CINxV	COUTx
R/W	R/W	R/W	-	-	-	R/W	R/W	R
Reset Value (POR/WDT/LVR/PIN)	0	0	-	-	-	0	0	0

Bit Number	Bit Mnemonic	Description
7	**CMPxEN**	**Comparator Enable Control bit** 0: Disable Comparator 1: Enable Comparator
6	**CMPxIF**	**Comparator output Interrupt Flag** 0: Comparator output has not changed 1: Comparator output has changed (must be cleared by software)
2	**CMPxOC**	**Comparator Output Control bit** 0: Comparator without output (CMPxO is I/O or other function) 1: Comparator with output (CMPxO is comparator output)
1	**CINxV**	**Comparator output Inversion bit** 0: Comparator output not Inverted 1: Comparator output Inverted
0	**COUTx**	**Comparator output bit** COUT = 0, when CMPP < CMPN and CINV = 0 COUT = 1, when CMPP > CMPN and CINV = 0 COUT = 0, when CMPP > CMPN and CINV = 1 COUT = 1, when CMPP < CMPN and CINV = 1

9.7 Low Voltage Reset (LVR)

9.7.1 Feature

- Enabled by the code option and V_{LVR} is 4.3V or 3.7V
- LVR de-bounce timer T_{LVR} is about 30-60μs
- An internal reset flag indicates low voltage reset generates

The LVR function is used to monitor the supply voltage and generate an internal reset in the device when the supply voltage below the specified value V_{LVR}. The LVR de-bounce timer T_{LVR} is about 30-60μs.

The LVR circuit has the following functions when the LVR function is enabled: (t means the time of the supply voltage below V_{LVR})

Generates a system reset when $V_{DD} \leq V_{LVR}$ and $t \geq T_{LVR}$;

Cancels the system reset when $V_{DD} > V_{LVR}$ or $V_{DD} < V_{LVR}$, but $t < T_{LVR}$.

The LVR function is enabled by the code option.

It is typically used in AC line or large battery supplier applications, where heavy loads may be switched on and cause the MCU supply-voltage temporarily falls below the minimum specified operating voltage. This feature can protect system from working under bad power supply environment.

9.8 Watchdog Timer (WDT) and Reset State

9.8.1 Feature

- Auto detect Program Counter (PC) over range, and generate OVL Reset
- WDT runs even in the Power-Down mode
- Selectable different WDT overflow frequency

OVL Reset

To enhance the anti-noise ability, SH88F516 built in Program Counter (PC) over range detect circuit, if program counter value is larger than flash romsize, or detect operation code equal to A5H which is not exist in 8051 instruction set, a OVL reset will be generate to reset CPU, and set WDOF bit. So, to make use of this feature, you should fill unused flash rom with A5H.

Watchdog Timer

The watchdog timer is a down counter, and its clock source is an independent built-in RC oscillator, so it always runs even in the Power-Down mode. The watchdog timer will generate a device reset when it overflows. It can be enabled or disabled permanently by the code option.

The watchdog timer control bits (WDT.2-0) are used to select different overflow frequency. The watchdog timer overflow flag (WDOF) will be automatically set to “1” by hardware when overflow happens. To prevent overflow happen, by reading or writing the WDT register RSTSTAT, the watchdog timer should re-count before the overflow happens.

There are also some reset flags in this register as below:

9.8.2 Register

Table 9.21 Reset Control Register

B1H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
RSTSTAT	WDOF	-	PORF	LVRF	CLRF	WDT.2	WDT.1	WDT.0
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR)	0	-	1	0	0	0	0	0
Reset Value (WDT)	1	-	u	u	u	0	0	0
Reset Value (LVR)	u	-	u	1	u	0	0	0
Reset Value (PIN)	u	-	u	u	1	0	0	0

U: unchanged

Bit Number	Bit Mnemonic	Description
7	**WDOF**	**Watch Dog Timer Overflow or OVL Reset Flag** Set by hardware when WDT overflow or OVL reset happened, cleared by software or Power On Reset 0: Watch Dog not overflows and no OVL reset generated 1: Watch Dog overflow or OVL reset occurred
5	**PORF**	**Power On Reset Flag** Set only by Power On Reset, cleared only by software 0: No Power On Reset 1: Power On Reset occurred
4	**LVRF**	**Low Voltage Reset Flag** Set only by Low Voltage Reset, cleared by software or Power On Reset 0: No Low Voltage Reset occurs 1: Low Voltage Reset occurred
3	**CLRF**	**Pin Reset Flag** Set only by pin reset, cleared by software or Power On Reset 0: No Pin Reset occurs 1: Pin Reset occurred
2-0	**WDT[2:0]**	**WDT Overflow period control bit** 000: Overflow period minimal value = 4096ms 001: Overflow period minimal value = 1024ms 010: Overflow period minimal value = 256ms 011: Overflow period minimal value = 128ms 100: Overflow period minimal value = 64ms 101: Overflow period minimal value = 16ms 110: Overflow period minimal value = 4ms 111: Overflow period minimal value = 1ms ***Notes:*** *If WDT_opt is enable in application, you must clear WatchDog periodically, and the interval must be less than the value list above.*

9.9 Power Management

9.9.1 Feature

- Two power saving modes: Idle mode and Power-Down mode
- Two ways to exit Idle and Power-Down mode: interrupt and reset

To reduce power consumption, SH88F516 supplies two power saving modes: Idle mode and Power-Down mode. These two modes are controlled by PCON & SUSLO register.

9.9.2 Idle Mode

In this mode, the clock to CPU is frozen, the program execution is halted, and the CPU will stop at a defined state. But the peripherals continue to be clocked. When entering idle mode, all the CPU status before entering will be preserved. Such as: PSW, PC, SFR & RAM are all retained.

By two consecutive instructions: setting SUSLO register as 0x55, and immediately followed by setting the IDL bit in PCON register, will make SH88F516 enter Idle mode. If the consecutive instruction sequence requirement is not met, the CPU will clear either SUSLO register or IDL bit in the next machine cycle. And the CPU will not enter Idle mode. The setting of IDL bit will be the last instruction that CPU executed.

There are two ways to exit Idle mode:

(1) An interrupt generated. After warm-up time, the clock to the CPU will be restored, and the hardware will clear SUSLO register and IDL bit in PCON register. Then the program will execute the interrupt service routine first, and then jumps to the instruction immediately following the instruction that activated Idle mode.

(2) Reset signal (logic high on the RESET pin, WDT RESET if enabled, LVR REST if enabled), this will restore the clock to the CPU, the SUSLO register and the IDL bit in PCON register will be cleared by hardware, finally the SH88F2051/4051 will be reset. And the program will execute from address 0000H. The RAM will keep unchanged and the SFR value might be changed according to different function module.

9.9.3 Power-Down Mode

The Power-Down mode places the SH88F516 in a very low power state. Power-Down mode will stop all the clocks including CPU and peripherals. If WDT is enabled, WDT block will keep on working. When entering Power-Down mode, all the CPU status before entering will be preserved. Such as: PSW, PC, SFR & RAM are all retained.

By two consecutive instructions: setting SUSLO register as 0x55, and immediately followed by setting the PD bit in PCON register, will make SH88F516 enter Power-Down mode. If the consecutive instruction sequence requirement is not met, the CPU will clear either SUSLO register or PD bit in the next machine cycle. And the CPU will not enter Power-Down mode.

The setting of PD bit will be the last instruction that CPU executed.

Note: *If IDL bit and PD bit are set simultaneously, the SH88F516 enters Power-Down mode. The CPU will not go in Idle mode when exiting from Power-Down mode, and the hardware will clear both IDL & PD bit after exit form Power-Down mode.*

There are three ways to exit the Power-Down mode:

(1) An active external Interrupt such as INT0, INT1 & INT4 will make SH88F516 exit Power-Down mode. The oscillator will start after interrupt happens, after warm-up time, the clocks to the CPU and peripheral will be restored, the SUSLO register and PD bit in PCON register will be cleared by hardware. Program execution resumes with the interrupt service routine. After completion of the interrupt service routine, program execution resumes with the instruction immediately following the instruction that activated Power-Down mode.

(2) Reset signal (logic high on the RESET pin, WDT RESET if enabled, LVR REST if enabled). This will restore the clock to the CPU after warm-up time, the SUSLO register and the PD bit in PCON register will be cleared by hardware, finally the SH88F516 will be reset. And the program will execute from address 0000H. The RAM will keep unchanged and the SFR value might be changed according to different function module.

(3) CMP Interrupt will make SH88F516 exit Power-Down mode. The oscillator will start after interrupt happens, after warm-up time, the clocks to the CPU and peripheral will be restored, the SUSLO register and PD bit in PCON register will be cleared by hardware. Program execution resumes with the interrupt service routine. After completion of the interrupt service routine, program execution resumes with the instruction immediately following the instruction that activated Power-Down mode.

Note: *In order to entering Idle/Power-Down, it is necessary to add 3 NOPs after setting IDL/PD bit in PCON.*

9.9.4 Register

Table 9.22 Power Control Register

87H	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
PCON	SMOD	SSTAT	SSTAT1	-	GF1	GF0	PD	IDL
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	-	0	0	0	0

Bit Number	Bit Mnemonic	Description
7	**SMOD**	**Baud rate double bit**
6	**SSTAT**	**SCON[7:5] function selection bit**
5	**SSTAT1**	**SCON1[7:5] function selection bit**
3-2	**GF[1:0]**	**General purpose flags for software use**
1	**PD**	**Power-Down mode control bit** 0: Cleared by hardware when an interrupt or reset occurs 1: Set by software to activate the Power-Down mode
0	**IDL**	**Idle mode control bit** 0: Cleared by hardware when an interrupt or reset occurs 1: Set by software to activate the Idle mode

Table 9.23 Suspend Mode Control Register

8EH	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
SUSLO	SUSLO.7	SUSLO.6	SUSLO.5	SUSLO.4	SUSLO.3	SUSLO.2	SUSLO.1	SUSLO.0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset Value (POR/WDT/LVR/PIN)	0	0	0	0	0	0	0	0

Bit Number	Bit Mnemonic	Description
7-0	**SUSLO[7:0]**	This register is used to control the CPU enter suspend mode (Idle or Power-Down). Only consecutive instructions like below will make CPU enter suspend mode. Other wise the either SUSLO, IDL or PD bit will be cleared by hardware in the next machine cycle.

Example:

```
IDLE_MODE:
        MOV     SUSLO, #55H
        ORL     PCON, #01H
        NOP
        NOP
        NOP
```

```
POWERDOWN_MODE:
        MOV     SUSLO, #55H
        ORL     PCON, #02H
        NOP
        NOP
        NOP
```

9.10 Warm-up Timer

9.10.1 Feature

- Built-in power on warm-up counter to eliminate unstable state of power on
- Built-in oscillator warm-up counter to eliminate unstable state when oscillation startup

SH88F516 has a built-in power warm-up counter; it is designed to eliminate unstable state after power on or to do some internal initial operation such as read customer option etc.

SH88F516 has also a built-in oscillator warm-up counter, it is designed to eliminate unstable state when oscillator starts oscillating in the following conditions: Power-on reset, Pin reset, LVR reset, Watchdog Reset and Wake up from Power-down mode.

After power-on, SH88F516 will start power warm-up procedure first, and then oscillator warm-up procedure.

Power Warm-up Time

Power On Reset/ Pin Reset/ Low Voltage Reset		**WDT Reset (Not in Power-Down Mode)**		**WDT Reset (Wakeup from Power-Down Mode)**		**Wakeup from Power-Down Mode (Only for interrupt)**	
TPWRT	OSC Warm up	TPWRT	OSC Warm up	TPWRT	OSC Warm up	TPWRT	OSC Warm up
11ms	YES	1000CKs	NO	11ms	YES	1000CKs	NO

OSC Warm-up Time

Option: OP_WMT / Oscillator Type	**00**	**01**	**10**	**11**
Ceramic	2^{13} X Tosc	2^{11} X Tosc	2^{9} X Tosc	2^{7} X Tosc
Crystal	2^{17} X Tosc	2^{15} X Tosc	2^{13} X Tosc	2^{11} X Tosc
32kHz Crystal	2^{13} X Tosc			
Internal RC	2^{7} X Tosc			

9.11 Code Option

OP_WDT[7]:
0: Disable WDT function (Default)
1: Enable WDT function

OP_WDTPD[6]:
0: Disable WDT function in Power-Down mode (Default)
1: Enable WDT function in Power-Down mode
Note: *When OP_WDT[7] = 1 is available.*

OP_WMT[4:3]: (unavailable for 32kHz crystal and Internal RC)
00: longest warm up time (Default)
01: longer warm up time
10: shorter warm up time
11: shortest warm up time

OP_OSC[2:0]:
000: Internal RC (16.6MHz) (Default)
010: External clock (30kHz - 16.6MHz)
011: 32.768kHz crystal oscillator
101: Crystal oscillator (400kHz - 16.6MHz) or Ceramic resonator (2MHz - 16.6MHz)
110: Ceramic resonator (400kHz - 2MHz)
Others: Internal RC (16.6MHz)

OP_LVREN[7]:
0: Disable LVR function (Default)
1: Enable LVR function

OP_LVRLE[6]:
0: 4.3V LVR level 1 (Default)
1: 3.7V LVR level 2

OP_SCM[3]:
0: SCM is invalid in warm up period (Default)
1: SCM is valid in warm up period

OP_IO[0]:
0: IO structure is only input mode after power-on reset
1: IO structure is Quasi-Bi mode after power-on reset (Default)

OP_ISP[7]:
0: Enable ISP function (Default)
1: Disable ISP function

OP_ISPPIN[6]:
0: Enter ISP mode only when P1.0 and P1.1 are connected to GND, simultaneously
1: Enter ISP mode directly regardless the condition of P1.0 and P1.1 (Default)
Note: *When OP_ISP[7] = 0 is available.*

10. Instruction Set

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	2	3
DEC @Ri		Decrement indirect RAM	0x16-0x17	1	3
INC DPTR		Increment data pointer	0xA3	1	4
MUL AB	8 X 8 16 X 8	Multiply A and B	0xA4	1	11 20
DIV AB	8 / 8 16 / 8	Divide A by B	0x84	1	11 20
DA A		Decimal adjust accumulator	0xD4	1	1

LOGIC OPERATIONS				
Opcode	**Description**	**Code**	**Byte**	**Cycle**
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

DATA TRANSFERS				
Opcode	**Description**	**Code**	**Byte**	**Cycle**
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

PROGRAM BRANCHES				
Opcode	**Description**	**Code**	**Byte**	**Cycle**
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) (taken)	Jump if accumulator is zero	0x60	2	3 5
JNZ rel (not taken) (taken)	Jump if accumulator is not zero	0x70	2	3 5
JC rel (not taken) (taken)	Jump if carry flag is set	0x40	2	2 4
JNC rel (not taken) (taken)	Jump if carry flag is not set	0x50	2	2 4
JB bit, rel (not taken) (taken)	Jump if direct bit is set	0x20	3	4 6
JNB bit, rel (not taken) (taken)	Jump if direct bit is not set	0x30	3	4 6
JBC bit, rel (not taken) (taken)	Jump if direct bit is set and clear bit	0x10	3	4 6
CJNE A, direct, rel (not taken) (taken)	Compare direct byte to A and jump if not equal	0xB5	3	4 6
CJNE A, #data, rel (not taken) (taken)	Compare immediate to A and jump if not equal	0xB4	3	4 6
CJNE Rn, #data, rel (not taken) (taken)	Compare immediate to reg. and jump if not equal	0xB8-0xBF	3	4 6
CJNE @Ri, #data, rel (not taken) (taken)	Compare immediate to Ri and jump if not equal	0xB6-0xB7	3	4 6
DJNZ Rn, rel (not taken) (taken)	Decrement register and jump if not zero	0xD8-0xDF	2	3 5
DJNZ direct, rel (not taken) (taken)	Decrement direct byte and jump if not zero	0xD5	3	4 6
NOP	No operation	0	1	1

BOOLEAN MANIPULATION				
Opcode	**Description**	**Code**	**Byte**	**Cycle**
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

11. Electrical Characteristics

Absolute Maximum Ratings*

DC Supply Voltage. . . . . -0.3V to +6.0V

DC Supply Voltage. . . . . .GND-0.3V to V_{DD}+0.3V

Operating Ambient Temperature. . . . . -40℃to +85℃

FLASH write/erase operating . . . . . 0℃to +85℃

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

***Comments**

Stresses exceed 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} = 3.6 - 5.5V, GND = 0V, T_A = +25°C, working in Quasi-Bi mode, unless otherwise specified)

Parameter	Symbol	Min.	Typ.*	Max.	Unit	Condition
Operating Voltage	V_{DD}	3.6	5.0	5.5	V	30kHz ≤ f_{OSC} ≤ 16.6MHz
Operating Current	I_{OP}	-	5	10	mA	f_{OSC} = 16.6MHz，V_{DD} = 5.0V All output pins unload (including all digital input pins unfloating) CPU on (execute NOP instruction), all other function block off
Stand by Current (IDLE)	I_{SB1}	-	25	35	μA	f_{OSC} = 32.768kHz，V_{DD} = 5.0V All output pins unload (including all digital input pins unfloating), all other function block off
		-	3	5	mA	f_{OSC} = 16.6MHz，V_{DD} = 5.0V All output pins unload (including all digital input pins unfloating), all other function block off
Stand by Current (Power-Down)	I_{SB2}	-	-	10	μA	f_{OSC} = 16.6MHz，V_{DD} = 5.0V All output pins unload(including all digital input pins unfloating), CPU off (Power-Down), all other function block off
WDT Current	I_{WDT}	-	1	3	μA	V_{DD} = 5.0V, All output pins unload, WDT on
LPD Current	I_{LPD}	-	3	5	μA	V_{DD} = 5.0V
Input Low Voltage	V_{IL}	GND	-	0.2 X V_{DD}	V	I/O Ports (all pin have schmitt trigger)
Input High Voltage	V_{IH}	0.8 X V_{DD}	-	V_{DD}	V	I/O Ports (all pin have schmitt trigger)
Input Leakage Current	I_{IL}	-1	-	1	μA	Input pad, V_{IN} = V_{DD} or GND (Input Only mode)
Output Leakage Current	I_{OL}	-1	-	1	μA	Open-drain, V_{DD} = 5.0V V_{OUT} = V_{DD} or GND (Open-Drain mode)
Very weak Pull-high Resistor	R_{PH1}	-	300	-	kΩ	V_{DD} = 5.0V，V_{IN} = GND
Weak Pull-high Resistor	R_{PH2}	-	10	-	kΩ	V_{DD} = 5.0V，V_{IN} = GND
Output High Voltage	V_{OH}	V_{DD} - 0.7	-	-	V	I/O Ports, I_{OH} = -10mA, V_{DD} = 5.0V (Push-Pull mode)
Output Low Voltage	V_{OL}	-	-	GND + 0.6	V	I/O Ports, I_{OL} = 10mA, V_{DD} = 5.0V (Push-Pull mode)

Note:

1. "" Data in "Typ." Column is at 5.0V, 25°C, unless otherwise specified.*

2. Maximum value of the supply current to V_{DD} is 100mA.

3. Maximum value of the output current from GND is 150mA.

5V A/D Converter Electrical Characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Supply Voltage	V_{AD}	4.5	5.0	5.5	V	
Resolution	N_R	-	10	-	bit	$GND \le V_{AIN} \le V_{REF}$
A/D Input Voltage*	V_{AIN}	GND	-	V_{REF}	V	
A/D Input Resistor	R_{AIN}	2	-	-	MΩ	V_{IN} = 5.0V
Recommended impedance of analog voltage source	Z_{AIN}	-	-	10	kΩ	
A/D conversion current	I_{AD}	-	1	3	mA	ADC module operating, V_{DD} = 5.0V
A/D Input current	I_{ADIN}	-	-	10	μA	V_{DD} = 5.0V
Differential linearity error	D_{LE}	-	-	±1	LSB	f_{OSC} = 16.6MHz, V_{DD} = 5.0V
Integral linearity error	I_{LE}	-	-	±2	LSB	f_{OSC} = 16.6MHz, V_{DD} = 5.0V
Full scale error	E_F	-	±1	±3	LSB	f_{OSC} = 16.6MHz, V_{DD} = 5.0V
Offset error	E_Z	-	±0.5	±2	LSB	f_{OSC} = 16.6MHz, V_{DD} = 5.0V
Total Absolute error	E_{AD}	-	-	±3	LSB	f_{OSC} = 16.6MHz, V_{DD} = 5.0V
Total Conversion time**	T_{CON}	14	-	-	t_{AD}	10 bit Resolution, V_{DD} = 5.0V

Note:

1. "" Here the A/D input Resistor is the DC input-resistance of A/D itself.*

*2. "**" Be sure that the series resistance connected with ADC input pin is no more than 10kΩ.*

Analog Comparator Electrical Characteristics (V_{DD} = 3.6V - 5.5V, GND = 0V, T_A = +25°C, f_{OSC} = 30KHz - 16.6MHz, unless otherwise specified)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Input Offset Voltage	$\|V_{IO}\|$	-	-	10	mV	
Input Common-Mode Voltage Range	V_{CM}	GND	-	V_{DD} - 1.0	V	
Response time	T_{RES}	-	250	500	ns	
Comparator enable to output valid time	T_{OV}	-	-	10	μs	
Input leakage current	I_{IL}	-	-	1	μA	$0 < V_{IN} < V_{DD}$

AC Electrical Characteristics (V_{DD} = 3.6V - 5.5V, GND = 0V, T_A = +25°C, f_{OSC} = 30KHz - 16.6MHz, unless otherwise specified)

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Oscillator start time	T_{OSC1}	-	1	2	s	f_{OSC}= 32.768kHz
Oscillator start time	T_{OSC2}	-	-	2	ms	f_{OSC} = 16.6MHz
RESET pulse width	t_{RESET}	10	-	-	μs	High active
RESET Pull-high Resistor	R_{RPH}	-	30	-	kΩ	V_{DD} = 5.0V, V_{IN} = GND
Frequency Stability (RC)*	F_{RC}	-1	-	1	%	V_{DD} = 3.6V - 5.5V, T_A = +25°C
		-3	-	3	%	V_{DD} = 3.6V - 5.5V, T_A = -10°C - +70°C
		-5		5	%	V_{DD} = 3.6V - 5.5V, T_A = -40°C - +85°C

***Note:** "*" RC frequency stability of ± 1% is for design guidance only and not tested.*

Low Voltage Reset Electrical Characteristics (V_{DD} = 3.6V - 5.5V, GND = 0V, T_A = +25°C, unless otherwise specified)

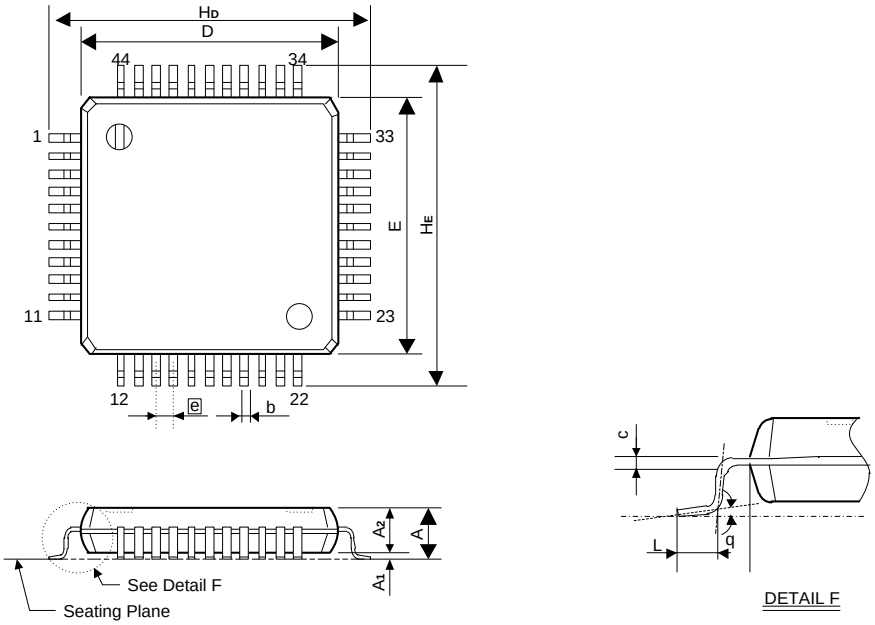
Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
LVR Voltage1	V_{LVR}	4.2	4.3	4.4	V	LVR enabled, V_{DD} = 3.6V - 5.5V
LVR Voltage2	V_{LVR}	3.6	3.7	3.8	V	LVR enabled, V_{DD} = 3.6V - 5.5V

12. Ordering Information

Part No.	Package
SH88F516F/044FR	QFP-44
SH88F516P/044PR	LQFP-44
SH88F54F/044FR	QFP-44
SH88F54U/048UR	TQFP-48
SH88F54P/044PR	LQFP-44
SH89F52F/044FR	QFP-44
SH89F52P/044PR	LQFP-44

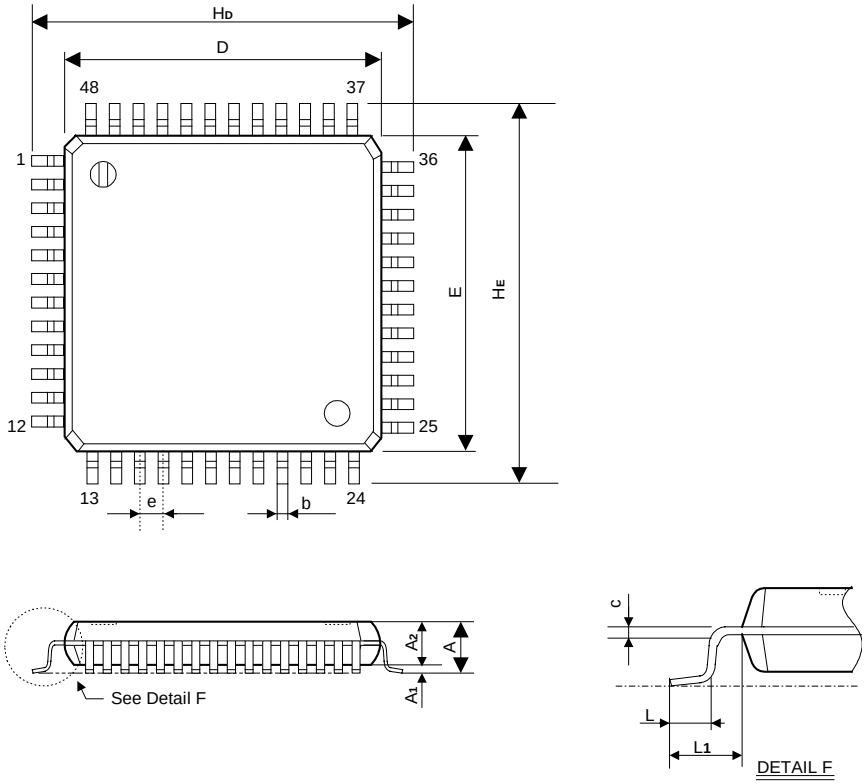
13. Package Information

QFP 44 Outline Dimensions unit: inch/mm



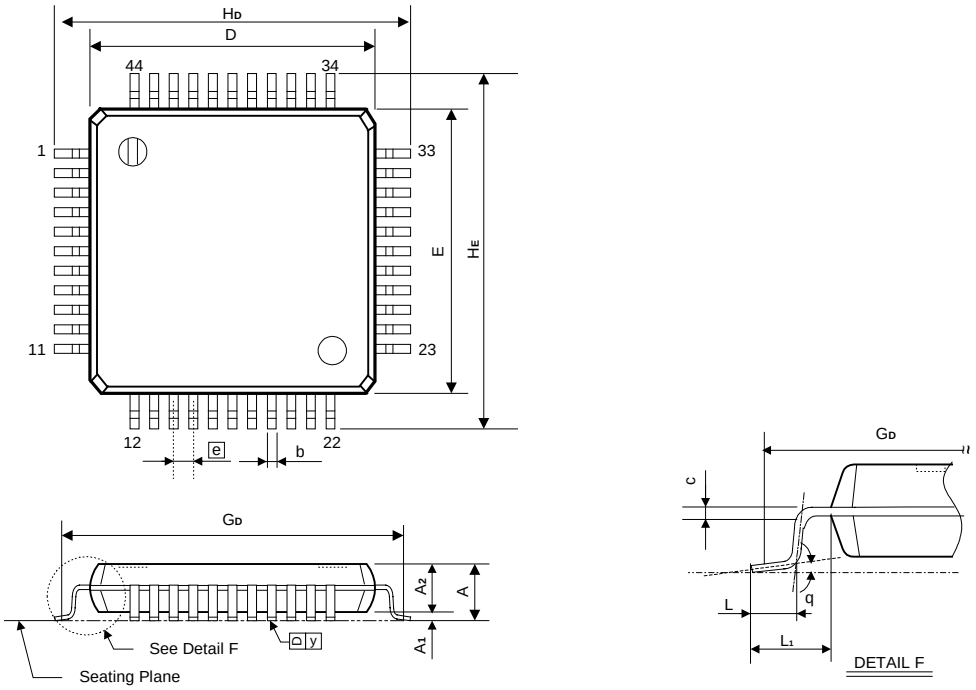
Symbol	Dimensions in inches	Dimensions in mm
A	0.106 Max.	2.70 Max.
A_1	0.012 Max.	0.3 Max
A_2	0.079 ± 0.004	2.00 ± 0.10
b	0.134 ± 0.001	0.35 ± 0.03
c	0.006 ± 0.002	0.15 ± 0.05
D	0.394 ± 0.006	10.00 ± 0.15
E	0.394 ± 0.006	10.00 ± 0.15
e	0.031 Typ.	0.80 Typ.
H_D	0.519 ± 0.014	13.20 ± 0.35
H_E	0.519 ± 0.014	13.20 ± 0.35
L	0.035 ± 0.006	0.9 ± 0.15
θ	0° ~ 11°	0° ~ 11°

TQFP48 Outline Dimensions unit: inch/mm



Symbol	Dimensions in inches		Dimensions in mm	
	MIN	MAX	MIN	MAX
A	---	0.047	---	1.2
A1	0.002	0.006	0.05	0.15
A2	0.035	0.041	0.9	1.05
D	0.270	0.281	6.85	7.15
E	0.270	0.281	6.85	7.15
H_D	0.346	0.362	8.8	9.2
H_E	0.346	0.362	8.8	9.2
b	0.007	0.010	0.19	0.26
e	0.020 TYP		0.500 TYP	
c	0.004	0.008	0.090	0.200
L	0.018	0.030	0.45	0.75
L1	0.033	0.045	0.85	1.15
θ	0°	10°	0°	10°

LQFP44 Outline Dimensions **unit: inch/mm**



Symbol	Dimensions in inches		Dimensions in mm	
	MIN	MAX	MIN	MAX
A	0.057	0.065	1.45	1.65
A1	0.000	0.001	0.01	0.21
A2	0.051	0.059	1.3	1.5
D	0.388	0.400	9.85	10.15
E	0.388	0.400	9.85	10.15
H_D	0.465	0.48	11.8	12.2
H_E	0.465	0.48	11.8	12.2
b	0.010	0.017	0.25	0.44
e	0.031 TYP		0.8 TYP	
c	0.005 TYP		0.127 TYP	
L	0.017	0.028	0.42	0.78
L1	0.037	0.045	0.95	1.15
θ	0°	10°	0°	10°

14. Product SPEC. Change Notice

Version	Content	Date
2.5	Revised Package Information	July. 2015
2.4	Original	Mar. 2014

Content