Datasheet

FS98O12

8-bit MCU with 4k program EPROM, 256-byte SRAM,
1 low noise OPAMP, 8-ch 14-bit ADC

Fortune
SEMICONDUCTOR CORPORATION

Fortune Semiconductor Corporation
富晶電子股份有限公司
23F., No.29-5,Sec. 2, Zhongzheng E. Rd.,
Danshui Town, Taipei County 251, Taiwan
Tel.：886-2-28094742
Fax：886-2-28094874
www.ic-fortune.com

Contents

1. General Description

The FS98O12 is a 8-bit high performance and cost-efficient microcontroller, 12-channel 14-bit sigma-delta ADC, and 24-bit general purpose I/O port. The device is suited for use in low power measurement and control applications such as: NiMH charger…etc.

2. Features

High Performance RISC CPU

- 8-bit single chip microcontroller(MCU).
- Embedded 4k x 16 bits program memory with one-time programmable (OTP) ROM.
- 256-byte data memory (SRAM).
- Only 37 single word instructions to learn
- 6-level memory stacks.

Peripheral Features

- 24-bit bi-directional I/O port.
- Charger current control.
- I2C serial I/O port (slave mode only).
- 12-channel 14-bit fully differential input analog to digital converter(ADC)

Analog Features

- 12-channel Sigma-Delta ADC with programmable output rate and resolution.
- Low noise (1μV Vpp without chopper, 0.5μV Vpp with chopper, 0.1Hz~1Hz) OPAMP with chopper controller.

Special Microcontroller Features

- Embedded Low Voltage Reset (LVR) and Low Voltage Detector (LVD).
- Embedded voltage regulator (3.3V regulated output).
- 5 Interrupt sources (external: 2, internal: 3).
- Watchdog timer (WDT).
- Embedded 3.2 MHz oscillator.

CMOS Technology

- Voltage operation ranges from 3.6V to 6.0V.
- Operation current is less than 1.5 mA; sleep mode current is about 1.5μA.

3. Applications

- ADC measure and I/O controls.
- NiMH/NiCd charger.

4. Ordering Information

Product Number	Description	Package Type
FS98O12-PCD	MCU with OTP ROM; The customer has to program the compiled hex code into OTP ROM.	44-pin package form
FS98O12-GCD	MCU with OTP ROM; The customer has to program the compiled hex code into OTP ROM.	44-pin package form
FS98O12-nnn-PCD	MCU with program type; FSC programs the customer's compiled hex code into EPROM at factory before shipping.	44-pin package form

FS98O12-nnn-GCD	MCU with program type; FSC programs the customer's compiled hex code into EPROM at factory before shipping.	44-pin package form

Note1: Code number (nnn) is assigned for customer.
Note2: Code number (nnn = 001~999); Version (V = A~Z).

5. Functional Block Diagram

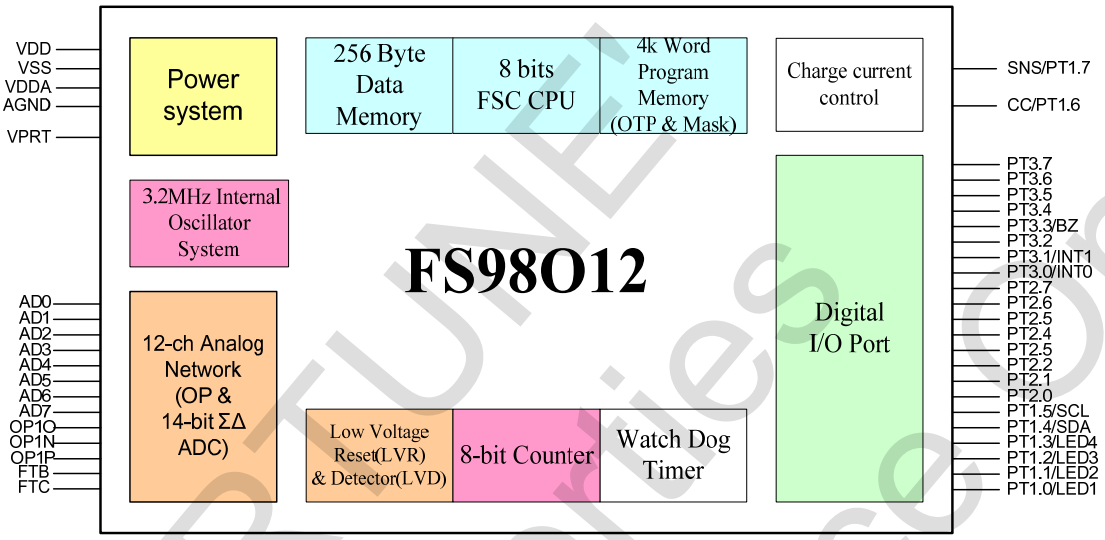
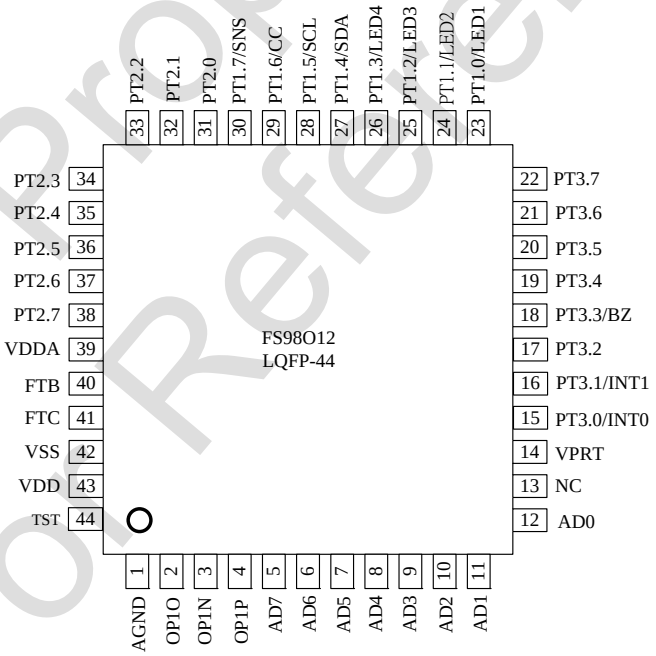


Figure 5-1: FS98O12 functional block

6. Pad Assignment



7. Pin Description

Name	Function	Input type	Description
VDD	VDD	PWR	Positive power supply for logic and I/O pins
VSS	VSS	PWR	Ground reference for logic and I/O pins
VDDA	VDD	PWR	Positive power supply for analog
AGND	AGND	PWR	Ground reference for analog
VPRT		PWR	The OTP ROM programming power and Reset pin.
TST	TST		Test pin
PT1.0/LED1	PT1.0	TTL	General I/O pin.
	LED1		Source or sink LED1 display.
PT1.1/LED2	PT1.1	TTL	General I/O pin.
	LED1		Source or sink LED2 display.
PT1.2/LED3	PT1.2	TTL	General I/O pin.
	LED3		Source or sink LED3 display.
PT1.3/LED4	PT1.3	TTL	General I/O pin.
	LED4		Source or sink LED4 display.
PT1.4/SDA	PT1.4		General I/O pin.
	SDA	ST	Serial data I/O for I2C.
PT1.5/SCL	PT1.5		General I/O pin.
	SCL	ST	Serial clock I/O for I2C.
PT1.6/CC	PT1.6	OD	Current sensing using an external sensing resistor RSNS
	CC	ST	General I/O pin.
PT1.7/SNS	PT1.7	TTL	Charge current control output
	SNS	AN	General I/O pin.
PT2.0~ PT2.7		TTL	General I/O pin.
PT3.2	PT3.2	ST	General I/O pin.
PT3.4~PT3.7		TTL	General I/O pin.
PT3.0/INT0	PT3.0		General I/O pin.
	INT0	ST	External Interrupt input
PT3.1/INT1	PT3.1		General I/O pin.
	INT1	ST	External Interrupt input
PT3.3/BZ	PT3.3	ST	General I/O pin.
	BZ		Buzzer Output
AD0~AD7		AN	Analog channel input.
FTB, FTC		AN	ADC Pre-Filter Capacitor Connection
OP1O		AN	OPAMP 1 Output
OP1P		AN	OPAMP 1 Input H
OP1N		AN	OPAMP 1 Input L

Legend: TTL= TTL input buffer, ST= Schmitt trigger input buffer, AN= Analog channel, OD= Open drain output, PWR= Power pin.

8. Typical Application Circuit

NI-MH Batteries Parallel Charging Application Circuit

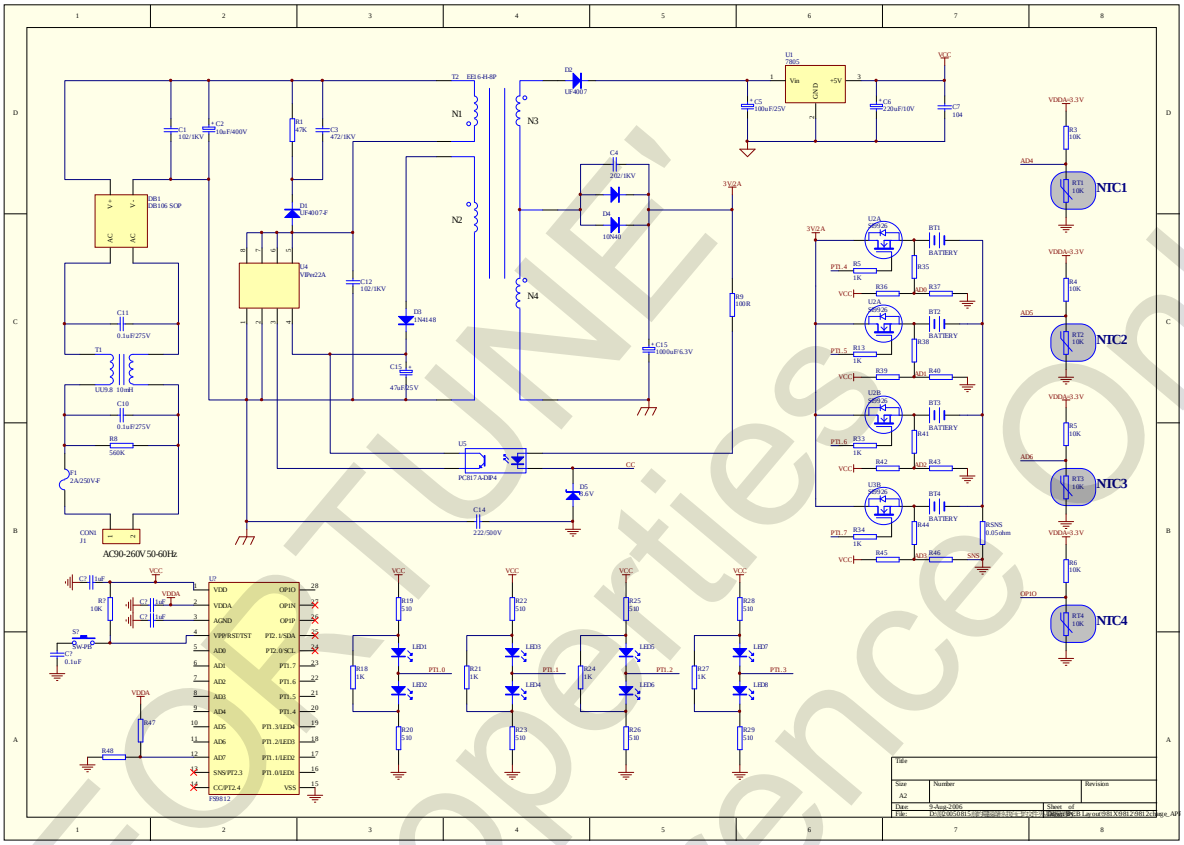


Figure 8-1: FS98O12 application circuit I

NI-MH Batteries Serial Charging Application Circuit

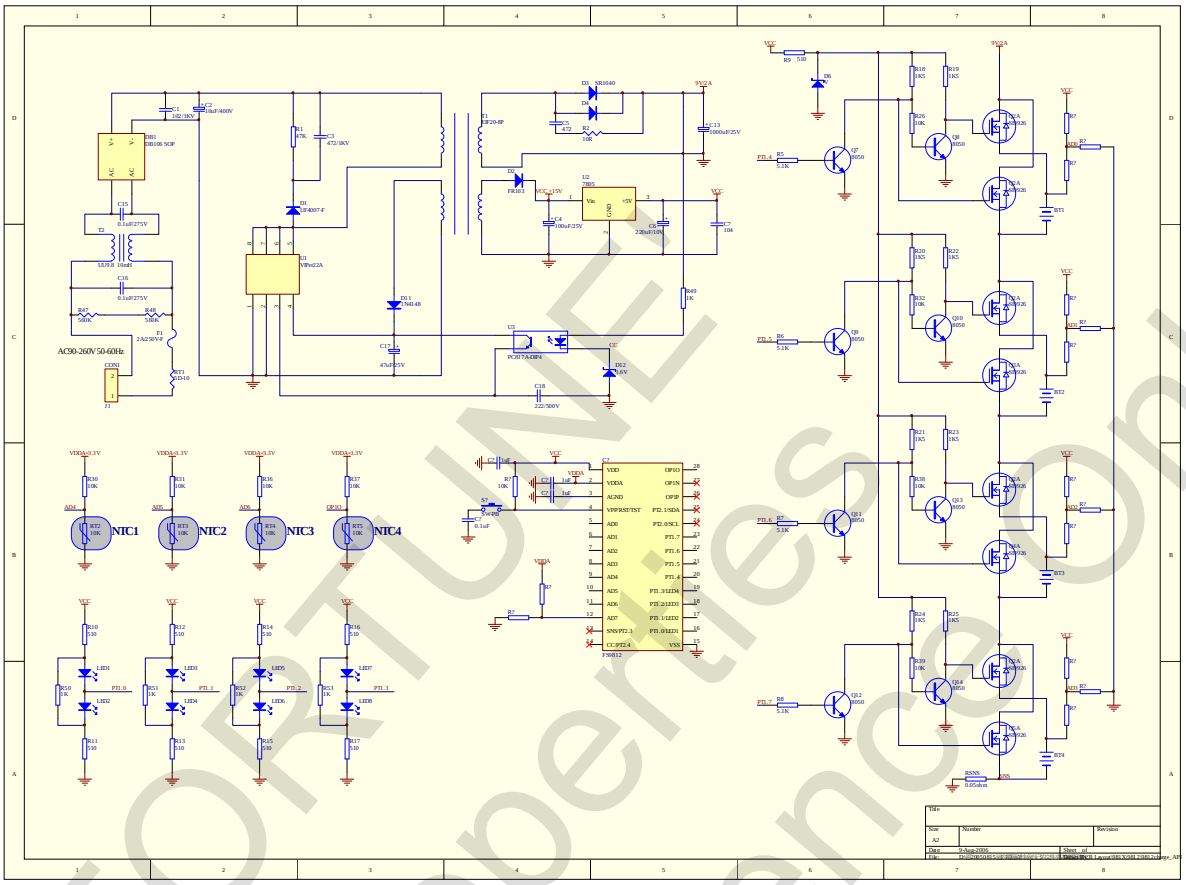


Figure 8-2: FS98O12 application circuit II

9. Electrical Characteristics

Absolute Maximum Ratings

Parameter	Rating	Unit
Supply Voltage to Ground Potential	-0.3 to 6.0	V
Applied Input/Output Voltage	VSS-0.3 to VDD+0.3	V
Ambient Operating Temperature	-40 to +85	°C
Storage Temperature	-55 to +150	°C
Soldering Temperature, Time	260°C, 10 Sec	
ESD immunity, Human Body Mode / Machine Model	≧2kV / 200V	
Latch-up immunity	≧100mA	

D.C. Characteristics (VDD= 5.0V, Ta=25°C, unless otherwise noted)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VDD	Recommend Operation Power Voltage	For MCU only (Disenable VDDA) *1	2.6		6.0	V
		For Reference Voltage (Enable VDDA) *2	3.6		6.0	
IDD1	Supply Current 1	VDD= 3.6V		1.5	4.5	mA
		VDD= 5.0V		2.5	4.5	mA
IDD2	Supply Current 2	Internal Oscillator Off, VDD= 3.6V		1.5	4.5	μA
		Internal Oscillator Off, VDD= 5.0V		2.5	4.5	μA
IPO	Sleep Mode Supply Current	Sleep Instruction, VDD= 3.6V		1.5	4.5	uA
		Sleep Instruction, VDD= 5.0V		2.5	4.5	μA
VIH	Digital Input High Voltage	PT1, Reset	0.7			VDD
VIL	Digital Input Low Voltage	PT1, Reset			0.3	VDD
VIHSH	Input Hys. High Voltage	Schmitt-trigger port		0.45		VDD
VIHSL	Input Hys. Low Voltage	Schmitt-trigger port		0.20		VDD
IPU	Pull up Current	VDD= 3.6V	20		32	uA
		VDD= 5.0V	48		68	uA
IOH	I/O Port source current	VDD= 3.6V VOH=0.9VDD	-7			mA
		VDD= 5.0V VOH=0.9VDD	-10			mA
	LED I/O Port source current	VDD= 3.6V VOH=0.9VDD	-10			mA
		VDD= 5.0V VOH=0.9VDD	-10			mA
IOL	I/O Port sink current	VDD= 3.6V VOL=0.1VDD	10			mA
		VDD= 5.0V VOL=0.1VDD	15			mA
	LED I/O Port sink current	VDD= 3.6V VOL=0.1VDD	15			mA
		VDD= 5.0V VOL=0.1VDD	20			mA
VDDA	Analog Power	3.6V < VDD < 6.0	3.27	3.3	3.33	V
ΔV_{LOAD}	VDDA Load Regulation	Output Current=10mA			0.1	mV/mA
AGND	Analog Ground Voltage	1/2VDDA	1.617	1.65	1.683	V
TCVDDA	Build in Reference Voltage Temperature Coefficient	Ta=(-40)~85℃		100		ppm/℃
VLBAT	Low Battery Detector Voltage	S_LB [1:0]=01		3.0		V
		S_LB [1:0]=10		3.3		
FRC	Internal RC oscillator			3.2		MHz
FWDT	Internal WDT Clock			21		kHz

*1. if VDDA is close, the operating voltage range for VDD voltage is 2.6V~6V.
*2. If VDDA is open, 3.6V~6V operating voltage is necessary for VDD voltage.

ADC Characteristics (VDD=5.0V, T_A=25℃, unless otherwise noted)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VACIN	ADC Common Mode Input Range	INH,INL,VRH,VRL to VSS	0.6		2.3	V
VADIN	ADC Differential Mode Input Range	(INH,INL), (VRH,VRL)			0.6	V
	Resolution			±15625		Counts
	ADC Linearity Error	VRFIN=0.44V	-0.1	0	+0.1	mV
	ADC Input Offset Voltage With Zero Cancellation	VRFIN=0.44V VAIN=0		0		V

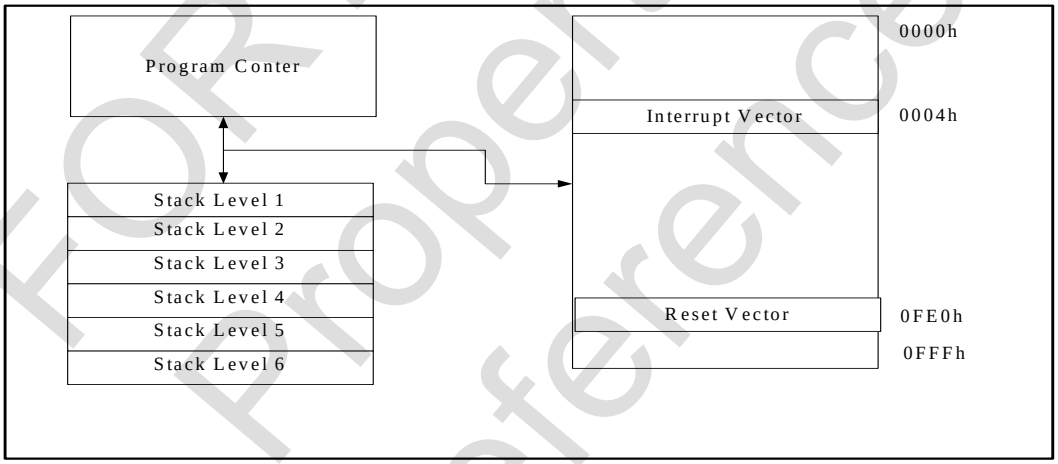
OPAMP Characteristics (VDD=5.0V, T_A=25℃, unless otherwise noted)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
	Input Offset			1		mV
	Input Offset Voltage with Chopper	Rs<100		20		V
	Input Reference Noise	Rs=100 , 0.1Hz~1Hz		1.0		Vpp
	Input Reference Noise with Chopper	Rs=100 , 0.1Hz~1Hz		0.5		Vpp
	Input Bias Current			10	30	pA
	Input Bias Current with Chopper			100	300	pA
	Input Common Mode Range		0.5		2.4	V
	Output Voltage Range		0.5		2.4	V
	Chopper Clock Frequency	S_CHCK[1:0]=11		1k		Hz
	Capacitor Load			50	100	pF

10. Memory Organization

Program Memory Structure

FS98O12 has an 4k x 16bits program memory space and a 6 level depth 12bits Stack Register. The Start up/Reset Vector is at 0x0FE0H. When FS98O12 is started or its program is reset, the Program Counter will point to Reset Vector. The Interrupt Vector is at 0x0004H. No matter what ISR is processed, the Program Counter will point to Interrupt Vector.



Data Memory Structure

FS98O12 has a 256 byte SRAM for Data Memory. The data memory is partitioned into three parts. The area with address 00h~07h is reserved for system special registers, such as indirect address, indirect address pointer, status register, working register, interrupt flag, interrupt control register. The address 08h~7Fh areas are peripheral special registers, such as I/O ports, timer, ADC, signal conditional network control register, LCD driver. The address 80h~17Fh areas are general data memory.

Start Address	End Address	Data Memory
0X00H	**0X07H**	***System Special Registers***
0X08H	**0X7FH**	***Peripheral Special Registers***
0X80H	**0X17FH**	***General Data Memory***

System Special Registers

The System Special Registers are designed to complete CPU Core functions, and consists of indirect address, indirect address pointer, status register, work register, interrupt flag, and interrupt control register.

Address	Name	Content (u mean unknown or unchanged)								Reset State
00H	IND0	Use contents of FSR0 to address data memory								uuuuuuuu
01H	IND1	Use contents of FSR1 to address data memory								uuuuuuuu
02H	FSR0	Indirect data memory, address point 0								uuuuuuuu
03H	FSR1	Indirect data memory, address point 1								uuuuuuuu
04H	STATUS	IRP1	IRP0		PD	TO	DC	C	Z	00u00uuu
05H	WORK	WORK register								uuuuuuuu
06H	INTF				TMIF	I2CIF	ADIF	E1IF	E0IF	00000000
07H	INTE	GIE			TMIE	I2CIE	ADIE	E1IE	E0IE	00000000
08h~7Fh		Peripheral special registers								-
80h~FFh		General data Memory 0 page								uuuuuuuu

- IND0, IND1: indirect addressing mode address
- FSR0, FSR1: indirect addressing mode point
- IRP0: Indirect address 0 page select.
- IRP1: Indirect address 1 page select.
- PD: Power down Flag. Cleared by writing 0 or power-on reset. Set by sleep instruction
- TO: Watch Dog Time Out Flag. Cleared by writing 0 or power-on reset. Set by Watch Dog Time Out
- DC: Digit Carry Flag, for ADDWF(C) and SUBWF(C), this bit is set if there is a carry out from the 4th order bit of resultant.
- C: Carry Flag (~Borrow)
- Z: Zero Flag
- E0IF, E0IE: PT2.4 External Interrupt flag and enable.
- E1IF, E1IE: PT2.5 External Interrupt flag and enable.
- ADIF, ADIE: Analog to digital converter Interrupt flag and enable.
- TMIF, TMIE: 8-bit Timer Interrupt flag and enable.
- I2CIF, I2CIE: I2C Interface Interrupt flag and enable.
- GIE: Global interrupt enable.

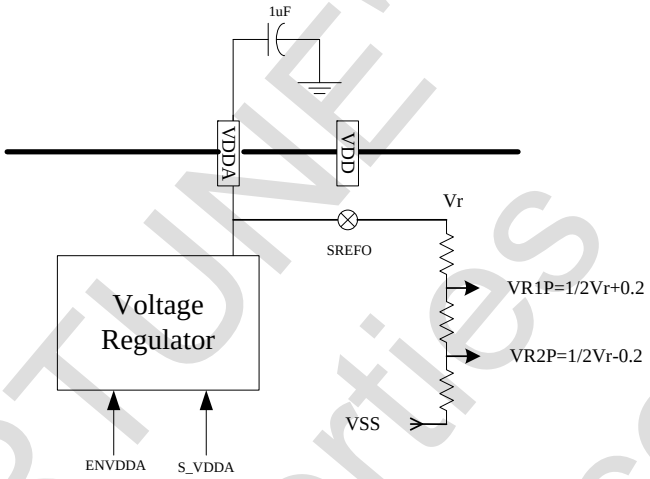
Peripheral Special Registers

Address	Name	Content (u mean unknown or unchanged)								Reset State
0DH	WDTCON	WTDTEN					WTS [2:0]			0uuuu000
0EH	TMOUT	TMOUT [7:0]								00000000
0FH	TMCON	TRST				TMEN	INS [2:0]			1uuu0000
10H	ADOH	ADO [15:8]								00000000
11H	ADOL	ADO [7:0]								00000000
13H	ADCON					ADRST	ADM [2:0]			uuuu0000
14H	MCK						M2_CK	M1_CK		00000000
15H	PCK					S_CH1CK[1:0]		S_Beep		Uuuu000u
18H	NETA	SINL[1:0]		SINH[2:0]			SFTA[2:0]			00000000
19H	NETB			SOP1N[1:0]		SVRL[1:0]		SVRH[1:0]		00000000
1AH	NETC	SREFO				ADG[1:0]		ADEN	AZ	00000000
1BH	NETD	ENCCref	ENCC	SCHAN[5:4]		OP1EN	SOP1P[2:0]			00000000
1CH	NETE	CURSEL [7:4]				SILB[1:0]		ENLB		00000000
1DH	NETF			ENVDDA	S_VDDA	S_AGND[1:0]		ENAGND	EN_VB	00000000
1FH	SVD								LBOUT	uuuuuuuu
20H	PT1	PT1 [7:0]								uuuuuuuu
21H	PT1EN	PT1EN [7:0]								00000000
22H	PT1PU	PT1PU [7:0]								00000000
24H	PT2	PT2 [7:0]								uuuuuuuu
25H	PT2EN	PT2EN [7:0]								00000000
26H	PT2PU	PT2PU [7:0]								00000000
28H	PT3	PT3 [7:0]								uuuuuuuu
29H	PT3EN	PT3EN [7:0]								00000000
2AH	PT3PU	PT3PU [7:0]								00000000
2BH	PT3MR				BZEN	E1M[1:0]		E0M[1:0]		00000000
37H	PT1OCB	PT1OCB[7:4]								10000000
57H	I2CCON	WCOL	I2COV	I2CEN	CKP					0001uuuu
58H	I2CSTA			DA	P	S	RW		BF	uu0000u0
59H	I2CADD	I2CADD [7:0]								00000000
5AH	I2CBUF	I2CBUF [7:0]								00000000

Power System

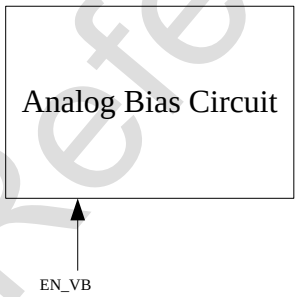
Address	Name	Content (u mean unknown or unchanged)							Reset State
1CH	NETE					SILB[1:0]	ENLB		00000000
1DH	NETF			ENVDDA	S_VDDA	S_AGND[1:0]	ENAGND	EN_VB	00000000
1FH	SVD							LBOUT	uuuuuuuu

10..1 Voltage Regulator



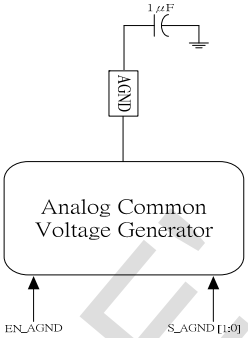
- Before enabling the VDDA, S_VDDA must be set 10mS.
- VDDA is the power supply voltage for analog circuit. When ENVDDA is set, the voltage regulator will active, and VDDA=3.3V. Otherwise VDDA can be as external regulated power supply input.
- If enabling VDDA, 3.6V~6V operating voltage is necessary for VDD voltage. SOP1P[2:0] in Analog Function Network can absolutely not set to ”000”.

10..2 Analog Bias Circuit



- Before enabling the analog block, EN_VB must be set.

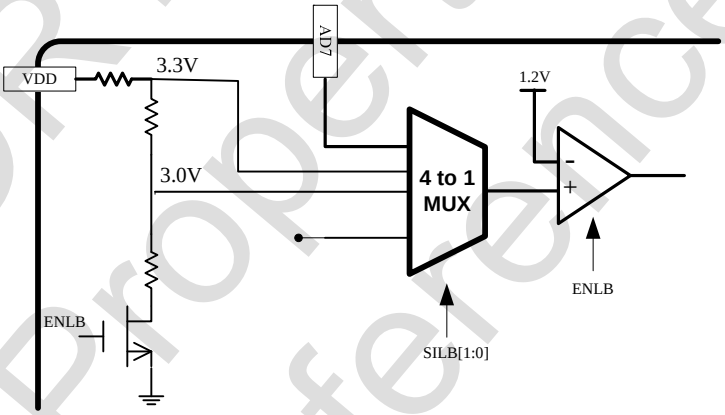
10..3 Analog Common Voltage Generator



- Analog common voltage generator is used to generate the analog common voltage, AGND.
- When EN_AGND = 1, analog common voltage generator is enabled. AGND =1/2 VDDA ,S_AGND is used to select the AGND voltage level.

S_AGND [1:0]	**AGND voltage**
01	1/2 VDDA

10..4 Low Battery Comparator



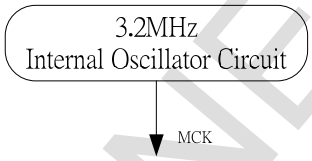
- When ENLB=1, low battery comparator will active.
- SILB [1:0]: Detect input select.

SILB [1:0]	Detect Voltage
00	Reserved
01	VDD < 3.0
10	VDD < 3.3
11	AD7 < 1.2

Clock System

The clock system offers several clocks to some important blocks in FS98O12, such as CPU clock, ADC sample frequency, beeper clock, voltage doubler operating frequency, etc. Only with the clock signals from the clock system, the FS98O12 can work normally.

Address	Name	Content (u mean unknown or unchanged)								Reset State
14H	MCK						M2_CK	M1_CK		00000000
15H	PCK		-			S_CH1CK [1:0]				00000000



Oscillator State

MCK is the heart of the clock system. Almost all clock signals are derived from the MCK. If we stop MCK, many clock signals will be stopped.

Sleep	MCK
1	Disable
0	Enable

- MCK and CLK

M1_CK	CLK
0	MCK
1	MCK/4

10..2 CPU Instruction Cycle

- When M2_CK=0, CPU has a different operation clock cycle from ADC in order to maintain a stable ADC output. In applications where a resolution of more than 13-bits is necessary, M2_CK should be set to zero.
- CPU's operation clock cycle may change as M1_CK,M2_CK change. Users must make sure that switching can be made only after the oscillator's output is stabilized. An NOP command should be added after the switching.

```
BSF   MCK, 2
NOP
….
```

M2_CK	M1_CK	Instruction Cycle (IF ADEN=1)	Instruction Cycle (IF ADEN=0)
0	0	MCK/7.14	MCK/2
0	1	MCK/14.28	MCK/4
1	0	MCK/2	MCK/2
1	1	MCK/4	MCK/4

10..3 ADC Sample Frequency

M1_CK	ADC sample Frequency (ADCF)
0	MCK/100
1	MCK/200

10..4 Chopper Operation Amplifier Input Control Signal

S_CH1CK [1]	S_CH1CK [0]	Chopper Control Signal
0	0	0
0	1	1
1	0	CLK/2000
1	1	CLK/4000

10..5 Timer Module Input Clock

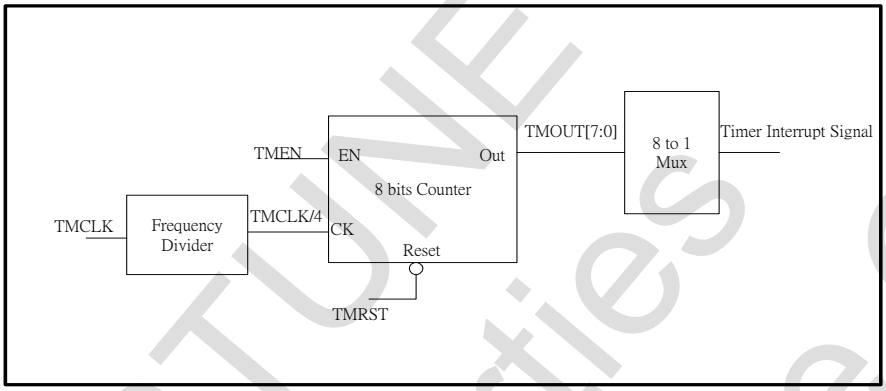
Timer Module input Clock TMCLK
MCK/1024

10..6 Beeper Clock

S_BEEP	Beeper Clock
0	MCK/512
1	MCK/1024

8-bits Timer

Address	Name	Content (u mean unknown or unchanged)								Reset State
06H	INTF				TMIF	-	-	-	-	00000000
07H	INTE	GIE			TMIE	-	-	-	-	00000000
0EH	TMOUT	TMOUT [7:0]								00000000
0FH	TMCON	TRST				TMEN	INS [2:0]			1uuu0000



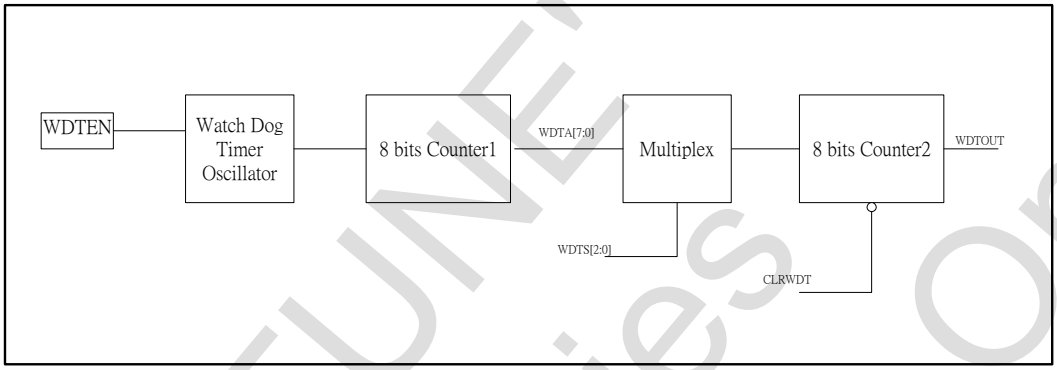
- Write a “0” to bit 7 of address 0Fh; the CPU will send a low pulse to TRST and reset the 8-bit counter. Then read bit 7 of TMCON to get “1”.
- TMEN=1, the 8-bit counter will be enabled. TMEN=0, the 8-bit counter will stop.
- TMOUT [7:0] is the output of the 8-bit counter. It is read-only.

INS [2:0] selects timer interrupt source. The selection codes are as follows:

INS	interrupt source	Time at TMCLK=3125Hz
000	TMOUT[0]	(TMCLK)/8
001	TMOUT[1]	(TMCLK)/16
010	TMOUT[2]	(TMCLK)/32
011	TMOUT[3]	(TMCLK)/64
100	TMOUT[4]	(TMCLK)/128
101	TMOUT[5]	(TMCLK)/256
110	TMOUT[6]	(TMCLK)/512
111	TMOUT[7]	(TMCLK)/1024

Watch Dog Timer

Address	Name	Content (u mean unknown or unchanged)								Reset State
04H	STATUS	-	-		-	TO	-	-	-	00u00uuu
0DH	WDTCON	WDTEN					WTS [2:0]			0uuuu000



- WDTEN = “1” : enable watchdog timer oscillator. “0” : watchdog timer function will be disabled. WDTEN write “1” only.
- When WDT Counter 2 overflows, it will send WDTOUT to reset the CPU and set TO flag.
- CLRWDT instruction will reset WDT Counter 2
- WTS [2:0] selects WDT Counter 2 and the code

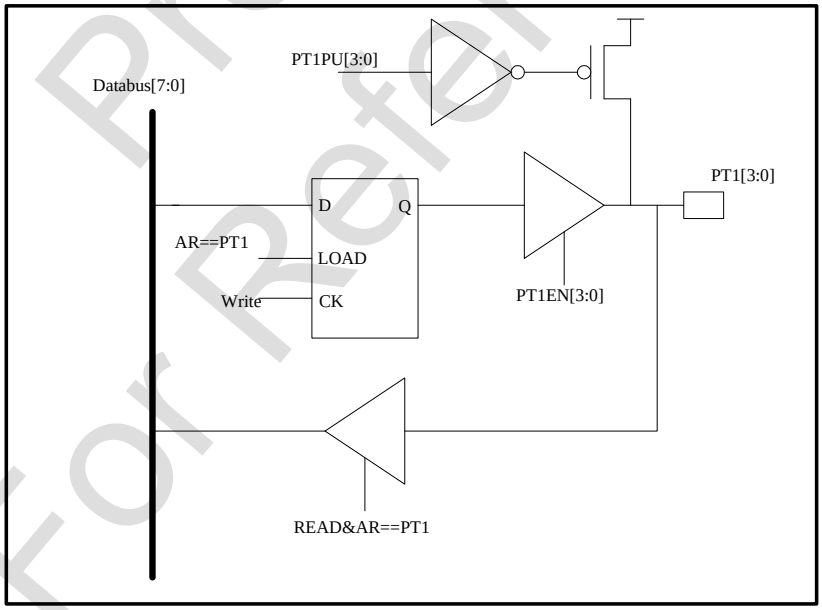
WTS[2:0]	Interrupt source	WDTOUT Time Oscilator=12kHz
000	WDTA[7]	$(12kHz/2^8)/256$
001	WDTA [6]	$(12kHz/2^7)/256$
010	WDTA [5]	$(12kHz/2^6)/256$
011	WDTA [4]	$(12kHz/2^5)/256$
100	WDTA [3]	$(12kHz/2^4)/256$
101	WDTA [2]	$(12kHz/2^3)/256$
110	WDTA [1]	$(12kHz/2^2)/256$
111	WDTA [0]	$(12kHz/2^1)/256$

I/O Port

Address	Name	Content (u mean unknown or unchanged)								Reset State
06H	INTF				-	I2CIF	-			00000000
07H	INTE	GIE			-	I2CIE	-			00000000
20H	PT1	PT1 [7:0]								uuuuuuuu
21H	PT1EN	PT1EN [7:0]								00000000
22H	PT1PU	PT1PU [7:0]								00000000
24H	PT2	PT2 [7:0]								uuuuuuuu
25H	PT2EN	PT2EN [7:0]								00000000
26H	PT2PU	PT2PU [7:0]								00000000
28H	PT3	PT3 [7:0]								uuuuuuuu
29H	PT3EN	PT3EN [7:0]								00000000
2AH	PT3PU	PT3PU [7:0]								00000000
2BH	PT3MR				BZEN	E1M[1:0]		E0M[1:0]		00000000
37H	PT1OCB	PT1OCB[7:4]								10000000

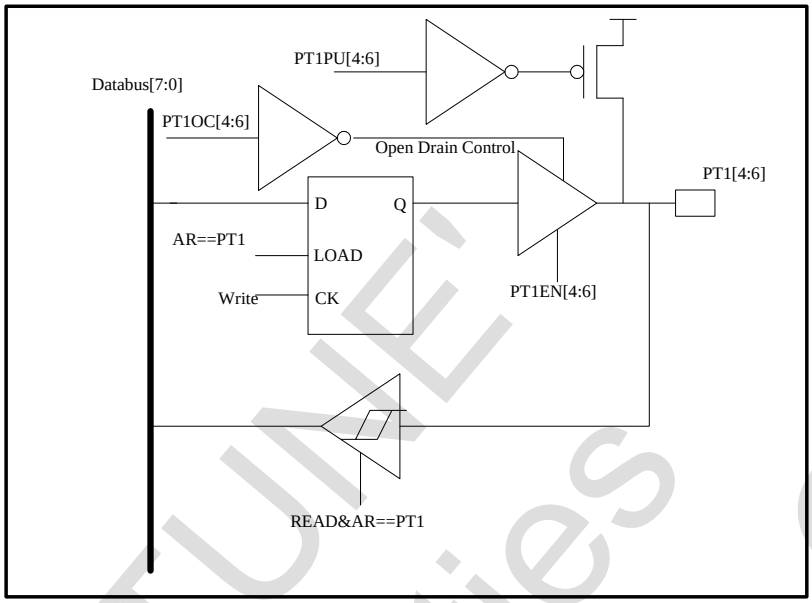
- I/O ports with pull-up resistor enable control. PT1PU [N]=”0”: PT1 [N] without pull-up resistor, “1”: PT1 [N] with pull-up resistor.
- I/O ports with pull-up resistor enable control. PT2PU [N]=”0”: PT2 [N] without pull-up resistor, “1”: PT2 [N] with pull-up resistor.
- I/O ports with pull-up resistor enable control. PT3PU [N]=”0”: PT3 [N] without pull-up resistor, “1”: PT3 [N] with pull-up resistor.
- PT1EN [N] =”0”: PT1 [N] is as input port, “1”: PT1 [N] is as output port.
- PT2EN [N] =”0”: PT2 [N] is as input port, “1”: PT2 [N] is as output port.
- PT3EN [N] =”0”: PT3 [N] is as input port, “1”: PT3 [N] is as output port.
- PT1 is the data register of I/O port.
- PT2 is the data register of I/O port.
- PT3 is the data register of I/O port.

10..1 Digital I/O Port and LED Display : PT1<0:3>



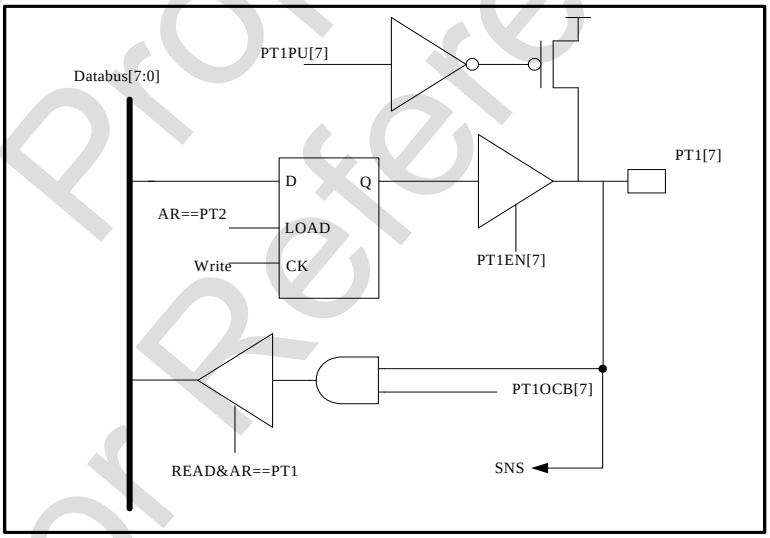
- PT1<0>/LED1 ~ PT1<3>/LED4 can be as Source or sink LED0 display.(10mA)

10..2 Digital I/O Port or I2C Serial Port and CC output: PT1<4:5>/SDA:SCL, PT1<6>/CC



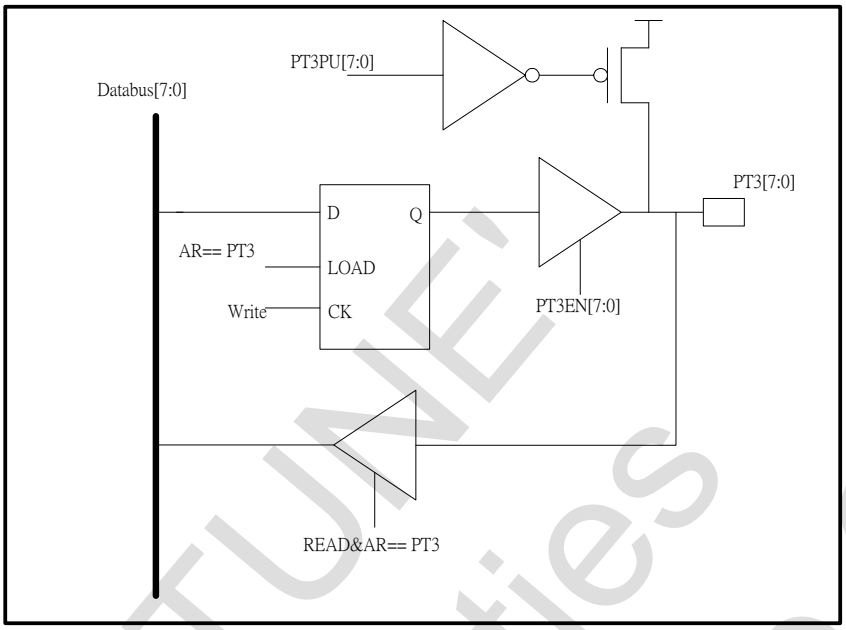
- When PT1OCB [4] =”1”: PT1[4] is open-drain;”0”: PT1 [4] is normal digital I/O port.
- When PT1OCB [5] =”1”: PT1[5] is open-drain;”0”: PT1 [5] is normal digital I/O port.
- When PT1OCB [6] =”1”: PT1[6] is open-drain;”0”: PT1 [6] is normal digital I/O port.
- CC is Charge control to Drive the pass Transistor.
- There has Schmitt-trigger input.
- I2C details see section I2C module.

10..3 Digital I/O Port with Analog Input Channel Shared : PT1<7>/SNS

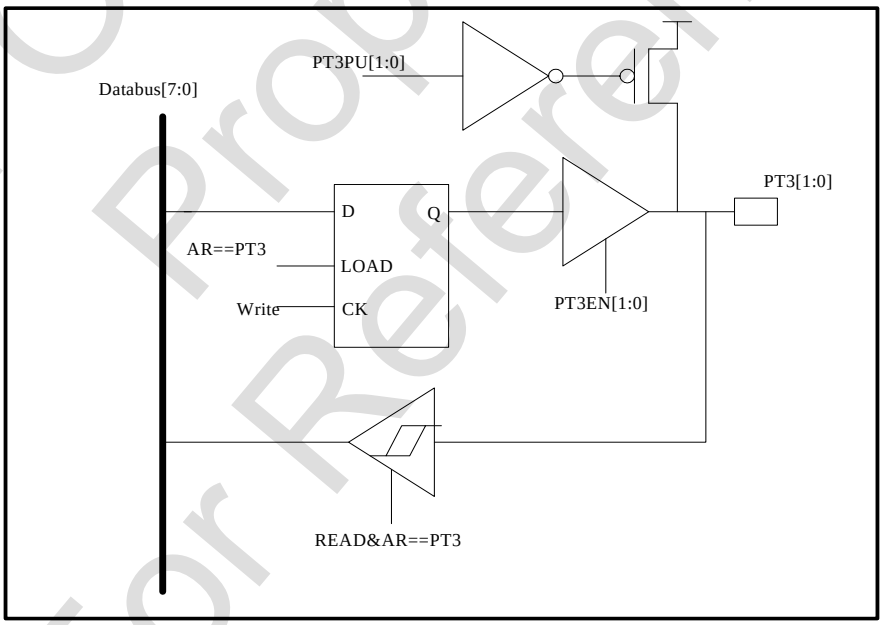


- SNS is Charge current sensing input.
- PT1OCB [7] =”0”, this port is Analog input channel (SNS), “1”: This port is Digital I/O port (PT1.7).
- If the port is used as digital I/O port, SOP1P[2:0] in Analog Function Network can absolutely not set to ”000”.

10..4 Digital I/O Port : PT2<0> ~ PT2<7>

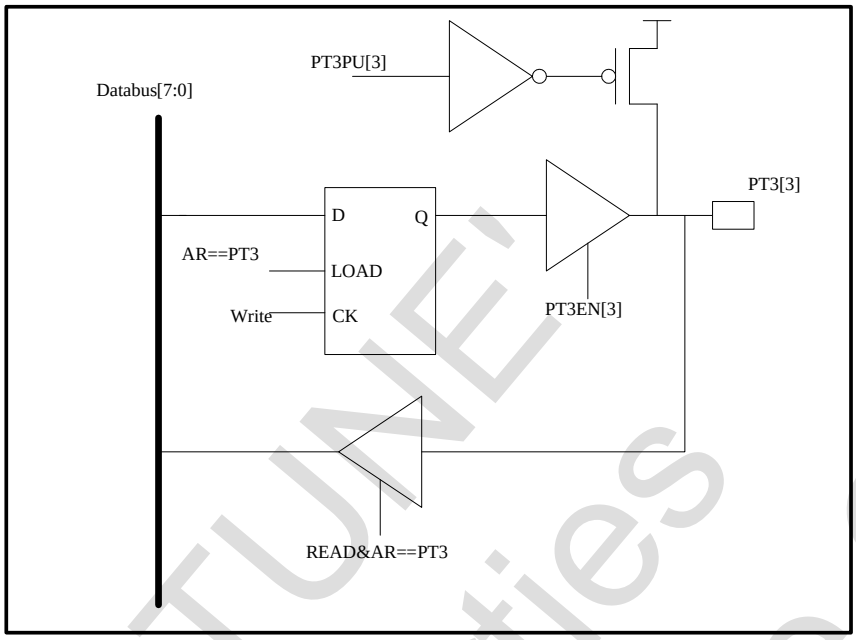


10..5 Digital I/O Port and External Interrupt Input : PT3<0>, PT3<1>



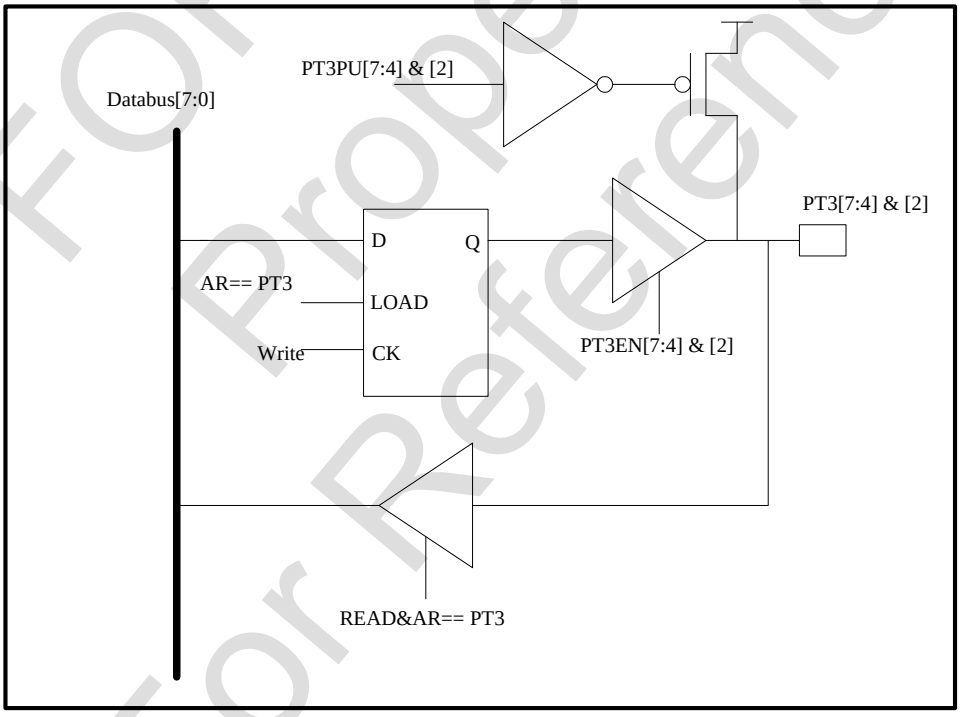
- PT3<0>/INT0, PT3<1>/INT1 can be as external interrupt sources. Interrupt mode is controlled by E0 (2) M [1:0] =”00”: negative edge, “01”: positive edge, “10”&”11”: interrupt when change.
- There has Schmitt-trigger input.

10..6 Digital I/O Port or Buzzer Output : PT3<3>



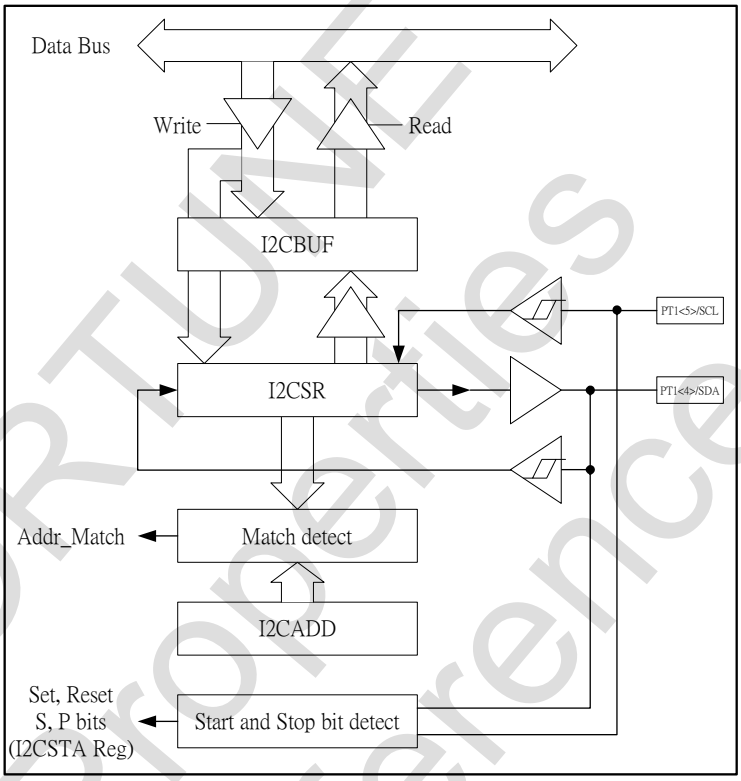
- PT3EN [3] =”1” and BZEN=”1”, PT3 [3] as buzzer output.

10..7 Digital I/O Port : PT3<7:4 > and PT3<2>



I2C (slave mode only)

Address	Name	Content (u mean unknown or unchanged)								Reset State
06H	INTF				-	I2CIF	-	-	-	00000000
07H	INTE	GIE			-	I2CIE	-	-	-	00000000
57H	I2CCON	WCOL	I2COV	I2CEN	CKP					0001uuuu
58H	I2CSTA			DA	P	S	RW		BF	uu0000u0
59H	I2CADD	I2CADD [7:0]								00000000
5AH	I2CBUF	I2CBUF [7:0]								00000000

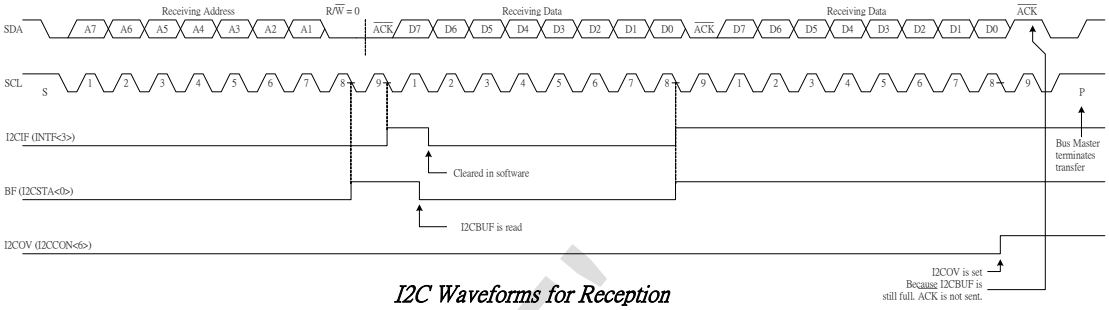


- The I2C module implements the standard specifications as well as 7-bit addressing. Two pins are used for data transfer. There are the PT1<5>/SCL pin, which is the clock, and the PT1<4>/SDA pin, which is the data. The user must configure these pins as open-drain through the PTOCB[5:4]. I2CSR: Shift Register is not directly accessible.

- I2CCON is the CONTROL REGISTER of I2C module.
  WCOL : Write collision detect.
  1 = the I2CBUF register is written while it is still transmitting the previous word.
  Must be cleared in software.
  0 = No collision.
  I2COV : Receive overflow flag.
  1 = A byte is received while the I2CBUF is still holding the previous byte.
  I2COV is a don't care in transmit mode.
  I2COV must be cleared in software in either mode.
  I2CEN : I2C functional enable.
  1 = Enables the serial port and configures SDA and SCL pins as serial port pins.
  0 = Disable serial port and configures these pins as I/O port pins.
  In both modes, when enabled, these pins must be properly configured as input or output.
  CKP : SCK release control.
  1 = Enable clock.
  0 = Holds clock low (clock stretch)
  Note : Used to ensure data setup time.

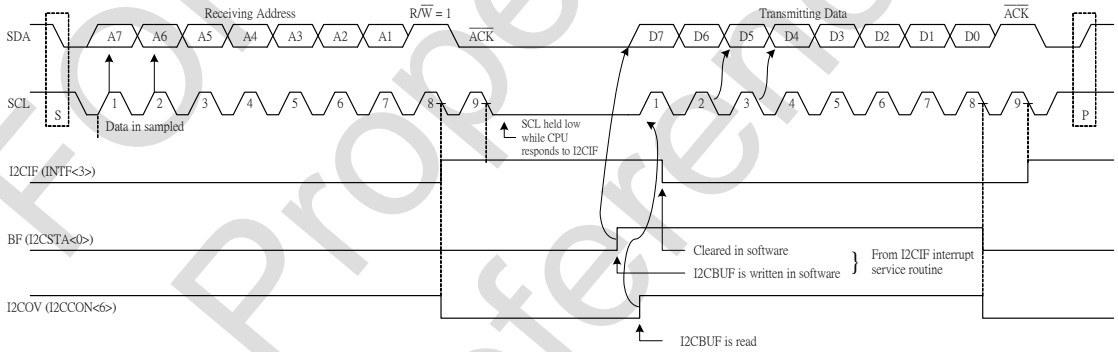
- I2CSTA is the STATUS REGISTER of I2C module
  DA : Data/Address bit
  1 = indicates that the last byte received was data
  0 = indicates that the last byte received was address
  P : Stop bit. This bit is cleared when the I2C module is disabled (I2CEN is cleared).
  1 = Indicates that a stop bit has been detected last.
  0 = Stop bit was not detected last.
  S : Start bit. This bit is cleared when the I2C module is disabled (I2CEN is cleared).
  1 = Indicates that a start bit has been detected last.
  0 = Start bit was not detected last.
  RW : Read/Write bit information. This bit holds the RW bit information received following the last address match. This bit is only valid during the transmission. The users may use this bit in software to determine whether transmission or reception is in progress. 1 = Read, 0 = Write

- I2CBUF is the BUFFER REGISTER of I2C module
- I2CADD is the ADDRESS REGISTER of I2C module
- Reception: When the R/W bit of the address byte is clear and an address match occurs, the R/W bit of the I2CSTA register is cleared. The received address is loaded into the I2CBUF. When the address byte overflow conditions exist then no acknowledge (ACK) pulse is given. An overflow condition is defined as either the BF bit (I2CSTA<0>) is set or the I2COV bit (I2CCON<6>) is set. An I2CIF interrupt is generated for each data transfer byte. The I2CIF bit must be cleared in software, and the I2CSTA register is used to determine the status of the byte.



I2C Waveforms for Reception

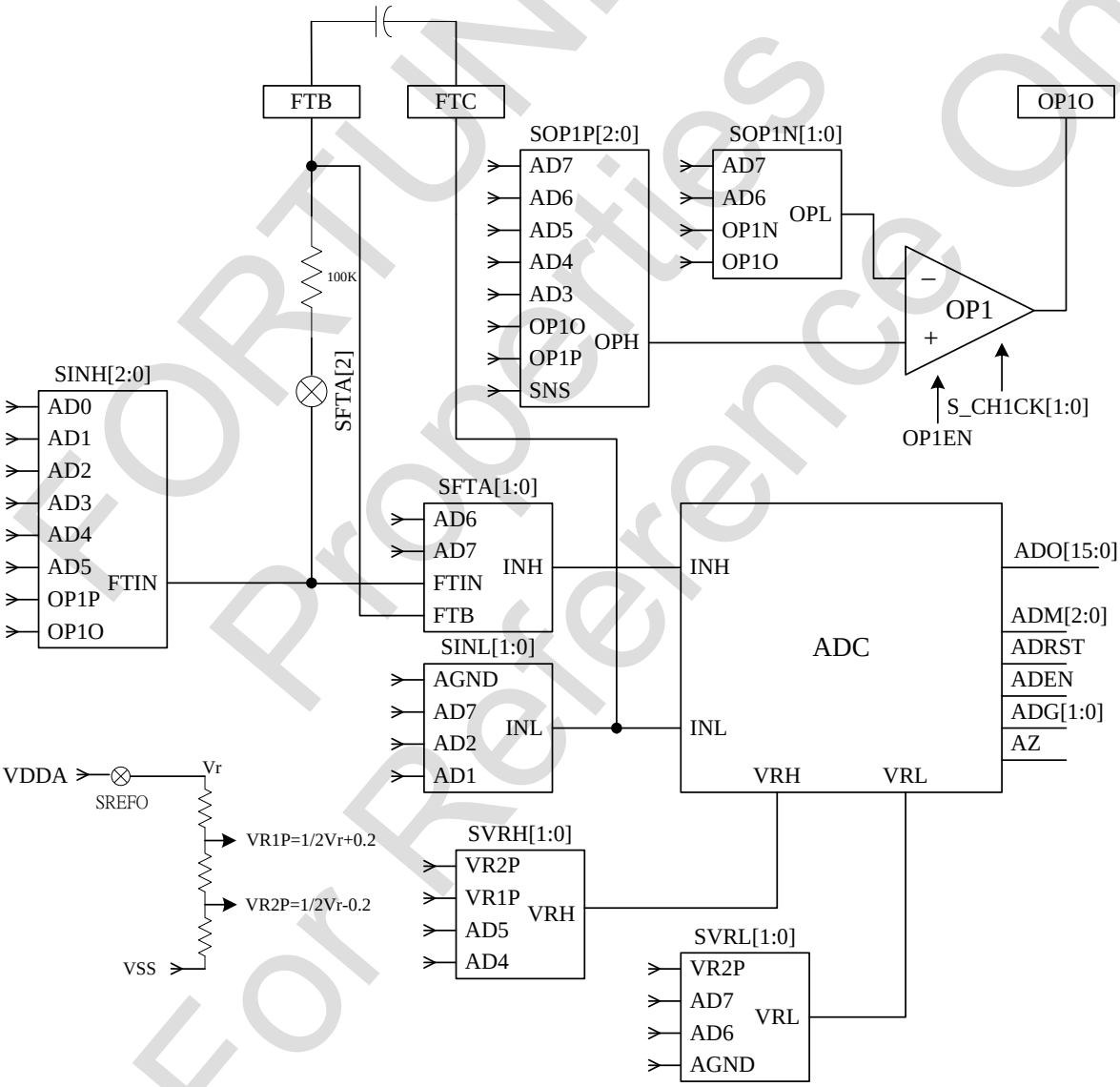
- Transmission: When the R/W bit of the address byte is set and an address match occurs, the R/W bit of the I2CSTA register is set. The received address is loaded into the I2CBUF. The ACK pulse will be sent on the ninth bit, and the SCL pin is held low. The transmit data must be loaded into the I2CBUF register, which also loads the I2CSR register. Then the SCL pin should be enabled by setting the CKP bit (I2CCON<4>). The right data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time. A I2CIF interrupt is generated for each data transfer byte. The I2CIF bit must be cleared in software, and the I2CSTA register is used to determine the status of the byte. The I2CIF bit is set on the falling edge of the ninth clock pulse. As a slave-transmitter, the ACK pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not ACK), then the data transfer is complete. The slave then monitors for another occurrence of the I2CSTA bit. If the SDA line was low (ACK), the transmit data must be loaded into the I2CBUF register, which also loads the I2CSR register. Then the SCL pin should be enabled by setting the CKP bit (I2CCON<4>).



I2C Waveforms for Transmission

Analog Function Network

Address	Name	Content (u mean unknown or unchanged)								Reset State
06H	INTF				-	-	ADIF	-	-	00000000
07H	INTE	GIE			-	-	ADIE	-	-	00000000
15H	PCK		-			S_CH1CK [1:0]		-	-	00000000
10H	ADOH	ADO [15:8]								00000000
11H	ADOL	ADO [7:0]								00000000
13H	ADCON					ADRST	ADM [2:0]			uuuu0000
18H	NETA	SINL[1:0]		SINH[2:0]			SFTA[2:0]			00000000
19H	NETB			SOP1N[1:0]		SVRL[1:0]		SVRH[1:0]		00000000
1AH	NETC	SREFO				ADG[1:0]		ADEN	AZ	00000000
1BH	NETD					OP1EN	SOP1P[2:0]			00000000



10..1 Analog to Digital Converter (ADC) :

- The ADC contains Σ-Δ modulator and digital comb filter. When ADRST=1, comb filter will be enabled. When ADRST=0, the comb filter will be reset. ADEN=1 starts the Σ-Δ modulator.
- The output rate is selected by ADM (N).

ADM (N)	ADC Output Rate
000	ADCF/125
001	ADCF/250
010	ADCF/500
011	ADCF/1000
100	ADCF/2000
101	ADCF/4000
110	ADCF/8000
111	ADCF/8000

- AZ=0 means that the ADC differential inputs are (INH, INL); AZ= 1 means that the ADC differential inputs are (INL, INL). We can use this mode to measure the ADC offset.
- ADG [1:0] will set ADC input gain as follows, 00: 2/3, 01: 1, 10: 2, 11: 2 1/3.

10..2 OPAMP : OP1

- OP1EN is the OPAMP enable control signal.
- S_CH1CK [1:0] Can set OP1 input operation mode as follows, 00: +Offset, 01: -Offset, 10: CLK/500 chopper frequency, 11: CLK/1000 Chopper frequency.

10..3 Analog Multiplex :

- Low Pass Filter Input:

SINH[2:0]	000	001	010	011	100	101	110	111
Select	AD0	AD1	AD2	AD3	AD4	AD5	OP1P	OP1O

- ADC Negative Input:

SINL[1:0]	00	01	10	11
Select	AD1	AD2	AD3	AGND

- Low Pass Filter Output, ADC Positive Input:

SFTA[1:0]	00	01	10	11
Select	FTB	FTIN	AD7	AD6

- Internal Reference Voltage Control: SREFO=1, to CDDA short; SREFO=0, to VDD short.
- OP1 Positive Input:

SOP1P[2:0]	000	001	010	011	100	101	110	111
Select	SNS	OP1P	OP1O	AD7	AD3	AD4	AD5	AD6

- OP1 Negative Input:

SOP1N[1:0]	00	01	10	11
Select	OP1O	OP1N	AD7	AD6

- ADC Reference Voltage Positive Input:

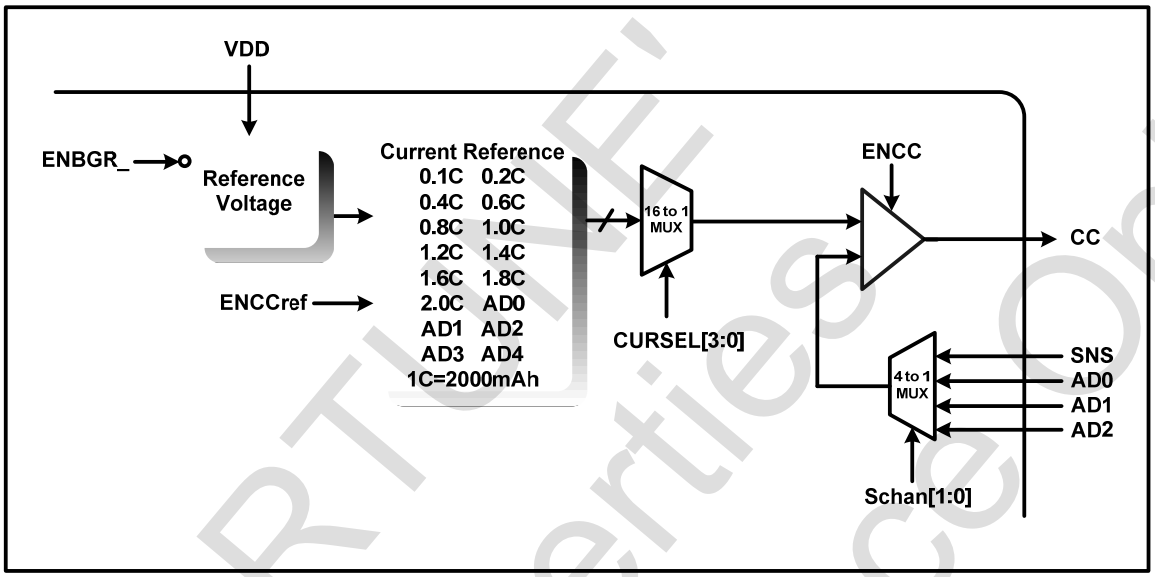
SVRH[1:0]	00	01	10	11
Select	AD4	AD5	VR1P	VR2P

- ADC Reference Voltage Negative Input:

SVRL[1:0]	00	01	10	11
Select	AGND	AD6	AD7	VR2P

Charge Current Control

Address	Name	Content (u means unknown or unchanged)								Reset State
		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	
1BH	NETD	ENCCref	ENCC	SCHAN[1]	SCHAN[0]					00000000
1CH	NETE	CURSEL[3]	CURSEL[2]	CURSEL[1]	CURSEL[0]					00000000



- Current sensing using an external sensing resistor RSNS=0.05ohm
- **ENCCref**: Enable constant current regulation reference current.0: Enable.
- **ENCC**: Enables the constant current regulation for constant current charge.
- The 3-bits select **CURSEL[3:0]** selects constant current regulation reference current. 1C=2000mAh

CURSEL[3:0]	0000	0001	0010	0011	0100	0101	0110	0111
Select	**200mA**	**400mA**	**800mA**	**1200mA**	**1600mA**	**2000mA**	**2400mA**	**2800mA**
voltage	**5mV~ 20 mV**	**15mV~ 35mV**	**35mV~ 55mV**	**50mV~ 75mV**	**70mV~ 95mV**	**90mV~ 120mV**	**105mV~ 130 mV**	**125mV~ 160 mV**
CURSEL[3:0]	1000	1001	1010	1011	1100	1101	1110	1111
Select	**3200mA**	**3600mA**	**4000mA**	**AD0**	**AD1**	**AD2**	Reserved	Reserved
voltage	**145mV~ 180 mV**	**165mV~ 210 mV**	**185mV~ 230 mV**	**Offset 10 mV**	**Offset 10 mV**	**Offset 10 mV**		

- Current sensing Input select **SCHAN[1:0]**

Schan[1:0]	00	01	10	11
Select	SNS	AD0	AD1	AD2

External Reset

The CPU has a “RST_” pin for external reset usage. When “RST_” is in logic “low” state, the CPU will go into external reset status. The external R/C circuit for reset is shown as following. When VDD changes from “low” to “high”, the CPU external reset status will be released, and the CPU will be in normal operating condition.

The signal from the “RST_” pin to CPU should remain in logic “low” state for more than 2μs to reset the CPU. If the signal from the “RST_” pin to CPU is in “low” state less than 2μs, the CPU will not be reset.

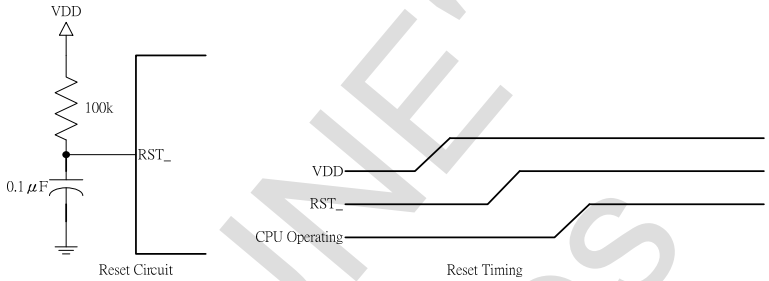


Figure 10-1: the Reset Circuit and the Reset Timing

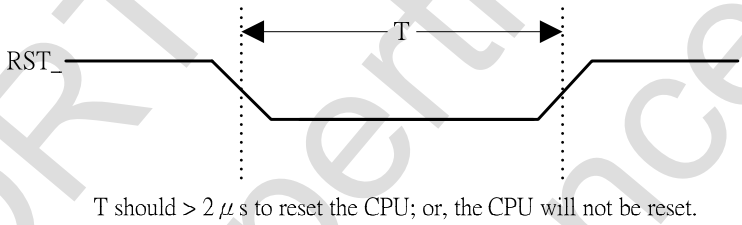


Figure 10-2: the Minimum Reset Period to Reset the CPU

Low Voltage Reset

To avoiding the CPU in an abnormal power status that makes the CPU unable to reset and causes the CPU operating abnormally, there’s a low voltage reset circuit embedded in FS98O12. When the voltage of VDD is less than LVR threshold low voltage, the CPU enters reset state; and when the voltage of VDD comes back above the LVR threshold high voltage, the CPU will be in normal operating condition.

11. Halt and Sleep Modes

FS98O12 supports low power working mode. When the user want FS98O12 to do nothing and just stand by, FS98O12 could be set to Halt mode or Sleep mode to reduce the power consumption by stopping the CPU core working. The two modes will be described below.

Halt Mode

After CPU executes a Halt command, CPU Program Counter (PC) stops counting until an interrupt command is issued. To avoid program errors caused by Interrupt Return, it is suggested to add a NOP command after Halt to guarantee the program’s normal execution when turning back.

Sleep Mode

After CPU executes Sleep command, all oscillators stop working until an external interrupt command is issued or the CPU is reset. To avoid program errors caused by Interrupt return, it is suggested to add a NOP command after Sleep to guarantee the program's normal execution. The sleep mode power consumption is about 3 uA.

To make sure that CPU consumes minimum power in Sleep mode, it is necessary to close all power blocks and analog circuits before issuing the Sleep command, and make sure that all I/O Ports are in VDD or VSS voltage levels.

It is recommended that users execute the following program before issuing the Sleep command:

```
CLRF    NETA      ; As Reset state
CLRF    NETB      ; As Reset state
CLRF    NETC      ; As Reset state
CLRF    NETD      ; As Reset state
CLRF    NETE      ; As Reset state
CLRF    NETF      ; As Reset state
CLRF    PT1PU     ; Pull up resistor is disconnected
CLRF    PT1EN     ; PT1[7:0] is assigned to be input ports.
CLRF    AINENB    ; Set PT1 as Analog Input Pin
CLRF    PT2PU     ; Pull up resistor is disconnected
CLRF    PT2EN     ; PT2[7:0] is assigned to be input ports.
MOVLW   03h
MOVWF   PT3PU     ; PT3 Pull up resistor is disconnected except port 0 & 1(external interrupt)
MOVLW   0FECh
MOVWF   PT3EN     ; PT3 ports are assigned to be output ports except port 0 & 1
CLRF    PT3       ; Set PT3 [7:2] Output Low
CLRF    INTF      ; Clear the interrupt flags
MOVLW   083h
MOVWF   INTE      ; Enable the external interrupt
SLEEP             ; Set the FS98O12 into Sleep mode
NOP               ; Guarantee that the program works normally when CPU wakes up.
```

12. Calibration

FS98O12 provides a regular resource (VDDA) with 1% accuracy and an Internal RC Oscillator (FRC), which to put the calibrating vale into the Peripheral special registers (30H & 31H) of Data Memory Structure in IC. They would be burned into ICs during producing. In order to running the calibrating value before the program is running, the Start up /Reset Vector in Program Memory of FS98O12 is set to 0FE0h. Therefore, please add up the following codes during programming:

```
ORG     0FE0h
MOVLW   0ffh
MOVWF   30h
MOVLW   0ffh
MOVWF   31h
GOTO        0000h
```

13. Instruction Set

The FS98O12 instruction set consists of 37 instructions. Each instruction could be converted to 16-bit OPCODE. The detailed descriptions are shown in the following sections.

Instruction Set Summary

Table 錯誤! 所指定的樣式的文字不存在文件中。-1 FS98O21 instruction set table

Instruction	Operation	Cycle	Flag
ADDLW k	[W] ← [W] + k	1	C, DC, Z
ADDPCW	[PC] ← [PC] + 1 + [W]	2	None
ADDWF f, d	[Destination] ← [f] + [W]	1	C, DC, Z
ADDWFC f, d	[Destination] ← [f] + [W] + C	1	C, DC, Z
ANDLW k	[W] ← [W] AND k	1	Z
ANDWF f, d	[Destination] ← [W] AND [f]	1	Z
BCF f, b	[f<b>] ← 0	1	None
BSF f, b	[f<b>] ← 1	1	None
BTFSC f, b	Skip if [f<b>] = 0	1, 2	None
BTFSS f, b	Skip if [f<b>] = 1	1, 2	None
CALL k	Push PC + 1 and GOTO k	2	None
CLRF f	[f] ← 0	1	Z
CLRWDT	Clear watch dog timer	1	None
COMF f, d	[f] ← NOT([f])	1	Z
DECF f, d	[Destination] ← [f] -1	1	Z
DECFSZ f, d	[Destination] ← [f] -1, skip if the result is zero	1, 2	None
GOTO k	PC ← k	2	None
HALT	CPU Stop	1	None
INCF f, d	[Destination] ← [f] +1	1	Z
INCFSZ f, d	[Destination] ← [f] + 1, skip if the result is zero	1, 2	None
IORLW k	[W] ← [W] | k	1	Z
IORWF f, d	[Destination] ← [W] | [f]	1	Z
MOVFW f	[W] ← [f]	1	None
MOVLW k	[W] ← k	1	None
MOVWF f	[f] ← [W]	1	None
NOP	No operation	1	None
RETFIE	Pop PC and GIE = 1	2	None
RETLW k	RETURN and W = k	2	None
RETURN	Pop PC	2	None
RLF f, d	[Destination<n+1>] ← [f<n>]	1	C,Z
RRF f, d	[Destination<n-1>] ← [f<n>]	1	C, Z
SLEEP	Stop OSC	1	PD
SUBLW k	[W] ← k – [W]	1	C, DC, Z
SUBWF f, d	[Destination] ← [f] – [W]	1	C, DC, Z
SUBWFC f, d	[Destination] ← [f] – [W] –Ċ	1	C, DC, Z
XORLW k	[W] ← [W] XOR k	1	Z
XORWF f, d	[Destination] ← [W] XOR [f]	1	Z

Note:

- f: memory address (00h ~ 7Fh).
- W: work register.
- k: literal field, constant data or label.
- d: destination select: d=0 store result in W, d=1: store result in memory address f.
- b: bit select (0~7).
- [f]: the content of memory address f.
- PC: program counter.
- C: Carry flag
- DC: Digit carry flag
- Z: Zero flag
- PD: power down flag

- TO: watchdog time out flag
- WDT: watchdog timer counter

Instruction Description

ADDLW	Add Literal to W
Syntax	ADDLW k 0 ≤ k ≤ FFh
Operation	[W] ← [W] + k
Flag Affected	C, DC, Z
Description	The content of Work register add literal “k” in Work register
Cycle	1
Example: **ADDLW 08h**	Before instruction: W = 08h After instruction: W = 10h

ADDPCW	Add W to PC
Syntax	ADDPCW
Operation	[PC] ← [PC] + 1 + [W], [W] < 79h [PC] ← [PC] + 1 + ([W] – 100h), otherwise
Flag Affected	None
Description	The relative address PC + 1 + W are loaded into PC.
Cycle	2
Example 1: **ADDPCW**	Before instruction: W = 7Fh, PC = 0212h After instruction: PC = 0292h
Example 2: **ADDPCW**	Before instruction: W = 80h, PC = 0212h After instruction: PC = 0193h
Example 3: **ADDPCW**	Before instruction: W = FEh, PC = 0212h After instruction: PC = 0211h

ADDWF	Add W to f
Syntax	ADDWF f, d 0 ≤ f ≤ FFh d ∈ [0,1]
Operation	[Destination] ← [f] + [W]
Flag Affected	C, CD, Z
Description	Add the content of the W register and [f]. If d is 0, the result is stored in the W register. If d is 1, the result is stored back in f.
Cycle	1
Example 1: **ADDWF OPERAND, 0**	Before instruction: OPERAND = C2h W = 17h After instruction: OPERAND = C2h W = D9h
Example 2: **ADDWF OPERAND, 1**	Before instruction: OPERAND = C2h W = 17h After instruction: OPERAND = D9h W = 17h

ADDWFC	Add W, f and Carry
Syntax	ADDWFC f, d 0 ≤ f ≤ FFh d ∈ [0,1]
Operation	[Destination] ← [f] + [W] + C
Flag Affected	C, DC, Z
Description	Add the content of the W register, [f] and Carry bit. If d is 0, the result is stored in the W register. If d is 1, the result is stored back in f.
Cycle	1
Example ADDWFC OPERAND,1	Before instruction: C = 1 OPERAND = 02h W = 4Dh After instruction: C = 0 OPERAND = 50h W = 4Dh

ANDLW	AND literal with W
Syntax	ANDLW k 0 ≤ k ≤ FFh
Operation	[W] ← [W] AND k
Flag Affected	Z
Description	AND the content of the W register with the eight-bit literal "k". The result is stored in the W register.
Cycle	1
Example: **ANDLW 5Fh**	Before instruction: W = A3h After instruction: W = 03h

ANDWF	AND W and f
Syntax	ANDWF f, d 0 ≤ f ≤ FFh d ∈ [0,1]
Operation	[Destination] ← [W] AND [f]
Flag Affected	Z
Description	AND the content of the W register with [f]. If d is 0, the result is stored in the W register. If d is 1, the result is stored back in f.
Cycle	1
Example 1: **ANDWF OPERAND,0**	Before instruction: W = 0Fh, OPERAND = 88h After instruction: W = 08h, OPERAND = 88h
Example 2: **ANDWF OPERAND,1**	Before instruction: W = 0Fh, OPERAND = 88h After instruction: W = 88h, OPERAND = 08h

BCF	Bit Clear f
Syntax	BCF f, b

	0 ≤ f ≤ FFh 0 ≤ b ≤ 7
Operation	[f<b>] ← 0
Flag Affected	None
Description	Bit b in [f] is reset to 0.
Cycle	1
Example: **BCF FLAG, 2**	Before instruction: FLAG = 8Dh After instruction: FLAG = 89h

BSF	Bit Set f
Syntax	BSF f, b 0 ≤ f ≤ FFh 0 ≤ b ≤ 7
Operation	[f<b>] ← 1
Flag Affected	None
Description	Bit b in [f] is set to 1.
Cycle	1
Example: **BSF FLAG, 2**	Before instruction: FLAG = 89h After instruction: FLAG = 8Dh

BTFSC	Bit Test skip if Clear
Syntax	BTFSC f, b 0 ≤ f ≤ FFh 0 ≤ b ≤ 7
Operation	Skip if [f<b>] = 0
Flag Affected	None
Description	If bit 'b' in [f] is 0, the next fetched instruction is discarded and a NOP is executed instead making it a two-cycle instruction.
Cycle	1, 2
Example: Node **BTFSC FLAG, 2** OP1 : OP2 :	Before instruction: PC = address (Node) After instruction: If FLAG<2> = 0 PC = address(OP2) If FLAG<2> = 1 PC = address(OP1)

BTFSS	Bit Test skip if Set
Syntax	BTFSS f, b 0 ≤ f ≤ FFh 0 ≤ b ≤ 7
Operation	Skip if [f<b>] = 1
Flag Affected	None
Description	If bit 'b' in [f] is 1, the next fetched instruction is discarded and a NOP is executed instead making it a two-cycle instruction.
Cycle	1, 2
Example: Node **BTFSS FLAG, 2** OP1 : OP2 :	Before instruction: PC = address (Node) After instruction: If FLAG<2> = 0 PC = address(OP1) If FLAG<2> = 1 PC = address(OP2)

CALL	Subroutine CALL
Syntax	CALL k

	0 ≤ k ≤ 1FFFh
Operation	Push Stack [Top Stack] ← PC + 1 PC ← k
Flag Affected	None
Description	Subroutine Call. First, return address PC + 1 is pushed onto the stack. The immediate address is loaded into PC.
Cycle	2

CLRF	Clear f
Syntax	CLRF f 0 ≤ f ≤ 255
Operation	[f] ← 0
Flag Affected	None
Description	Reset the content of memory address f
Cycle	1
Example: **CLRF WORK**	Before instruction: WORK = 5Ah After instruction: WORK = 00h

COMF	Complement f
Syntax	COMF f, d 0 ≤ f ≤ 255 d ∈ [0,1]
Operation	[f] ← NOT([f])
Flag Affected	Z
Description	[f] is complemented. If d is 0, the result is stored in the W register. If d is 1, the result is stored back in [f]
Cycle	1
Example 1: **COMF OPERAND,0**	Before instruction: W = 88h, OPERAND = 23h After instruction: W = DCh, OPERAND = 23h
Example 2: **COMF OPERAND,1**	Before instruction: W = 88h, OPERAND = 23h After instruction: W = 88h, OPERAND = DCh

DECF	Decrement f
Syntax	DECF f, d 0 ≤ f ≤ 255 d ∈ [0,1]
Operation	[Destination] ← [f] -1
Flag Affected	Z
Description	[f] is decremented. If d is 0, the result is stored in the W register. If d is 1, the result is stored back in [f].
Cycle	1
Example 1: **DECF OPERAND,0**	Before instruction: W = 88h, OPERAND = 23h After instruction: W = 22h, OPERAND = 23h
Example 2: **DECF OPERAND,1**	Before instruction: W = 88h, OPERAND = 23h After instruction: W = 88h, OPERAND = 22h

CLRWDT	Clear watch dog timer

Syntax	CLRWDT
Operation	Watch dog timer counter will be reset
Flag Affected	None
Description	CLRWDT instruction will reset watch dog timer counter.
Cycle	1
Example: **CLRWDT**	After instruction: WDT = 0

DECFSZ	Decrement f, skip if zero
Syntax	DECFSZ f, d $0 \le f \le FFh$ $d \in [0,1]$
Operation	[Destination] ← [f] -1, skip if the result is zero
Flag Affected	None
Description	[f] is decremented. If d is 0, the result is stored in the W register. If d is 1, the result is stored back in [f]. If the result is 0, then the next fetched instruction is discarded and a NOP is executed instead making it a two-cycle instruction.
Cycle	1, 2
Example: Node **DECFSZ FLAG, 1** OP1 : OP2 :	Before instruction: PC = address (Node) After instruction: [FLAG] = [FLAG] - 1 If [FLAG] = 0 PC = address(OP1) If [FLAG] ≠ 0 PC = address(OP2)

GOTO	Unconditional Branch
Syntax	GOTO k $0 \le k \le 1FFFh$
Operation	PC ← k
Flag Affected	None
Description	The immediate address is loaded into PC.
Cycle	2

INCF	Increment f
Syntax	INCF f, d $0 \le f \le FFh$ $d \in [0,1]$
Operation	[Destination] ← [f] +1
Flag Affected	Z
Description	[f] is incremented. If d is 0, the result is stored in the W register. If d is 1, the result is stored back in [f].
Cycle	1
Example 1: **INCF OPERAND,0**	Before instruction: W = 88h, OPERAND = 23h After instruction: W = 24h, OPERAND = 23h
Example 2: **INCF OPERAND,1**	Before instruction: W = 88h, OPERAND = 23h After instruction: W = 88h, OPERAND = 24h

INCFSZ	Increment f, skip if zero

Syntax	INCFSZ f, d 0 ≤ f ≤ FFh d ∈ [0,1]
Operation	[Destination] ← [f] + 1, skip if the result is zero
Flag Affected	None
Description	[f] is incremented. If d is 0, the result is stored in the W register. If d is 1, the result is stored back in [f]. If the result is 0, then the next fetched instruction is discarded and a NOP is executed instead making it a two-cycle instruction.
Cycle	1, 2
Example: Node **INCFSZ FLAG, 1** OP1 : OP2 :	Before instruction: PC = address (Node) After instruction: [FLAG] = [FLAG] + 1 If [FLAG] = 0 PC = address(OP2) If [FLAG] ≠ 0 PC = address(OP1)

HALT	Stop CPU Core Clock
Syntax	HALT
Operation	CPU Stop
Flag Affected	None
Description	CPU clock is stopped. Oscillator is running. CPU can be waked up by internal and external interrupt sources.
Cycle	1

IORLW	Inclusive OR literal with W
Syntax	IORLW k 0 ≤ k ≤ FFh
Operation	[W] ← [W] | k
Flag Affected	Z
Description	Inclusive OR the content of the W register and the eight-bit literal "k". The result is stored in the W register.
Cycle	1
Example: **IORLW 85h**	Before instruction: W = 69h After instruction: W = EDh

IORWF	Inclusive OR W with f
Syntax	IORWF f, d 0 ≤ f ≤ FFh d ∈ [0,1]
Operation	[Destination] ← [W] | [f]
Flag Affected	Z
Description	Inclusive OR the content of the W register and [f]. If d is 0, the result is stored in the W register. If d is 1, the result is stored back in [f].
Cycle	1
Example: **IORWF OPERAND,1**	Before instruction: W = 88h, OPERAND = 23h After instruction: W = 88h, OPERAND = ABh

MOVFW	Move f to W

Syntax	MOVFW f 0 ≤ f ≤ FFh
Operation	[W] ← [f]
Flag Affected	None
Description	Move data from [f] to the W register.
Cycle	1
Example: **MOVFW OPERAND**	Before instruction: W = 88h, OPERAND = 23h After instruction: W = 23h, OPERAND = 23h

MOVLW	Move literal to W
Syntax	MOVLW k 0 ≤ k ≤ FFh
Operation	[W] ← k
Flag Affected	None
Description	Move the eight-bit literal "k" to the content of the W register.
Cycle	1
Example: **MOVLW 23h**	Before instruction: W = 88h After instruction: W = 23h

MOVWF	Move W to f
Syntax	MOVWF f 0 ≤ f ≤ FFh
Operation	[f] ← [W]
Flag Affected	None
Description	Move data from the W register to [f].
Cycle	1
Example: **MOVWF OPERAND**	Before instruction: W = 88h, OPERAND = 23h After instruction: W = 88h, OPERAND = 88h

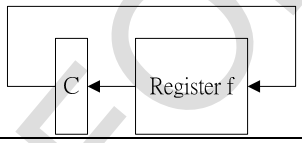
NOP	No Operation
Syntax	NOP
Operation	No Operation
Flag Affected	None
Description	No operation. NOP is used for one instruction cycle delay.
Cycle	1

RETFIE	Return from Interrupt
Syntax	RETFIE
Operation	[Top Stack] => PC Pop Stack 1 => GIE
Flag Affected	None
Description	The program counter is loaded from the top stack, then pop stack. Setting the GIE bit enables interrupts.
Cycle	2

RETLW	Return and move literal to W

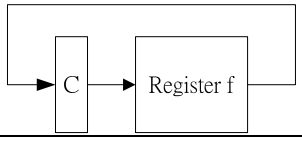
Syntax	RETLW k 0 ≤ k ≤ FFh
Operation	[W] ← k [Top Stack] => PC Pop Stack
Flag Affected	None
Description	Move the eight-bit literal "k" to the content of the W register. The program counter is loaded from the top stack, then pop stack.
Cycle	2

Return	Return from Subroutine
Syntax	RETURN
Operation	[Top Stack] => PC Pop Stack
Flag Affected	None
Description	The program counter is loaded from the top stack, then pop stack.
Cycle	2

RLF	Rotate left [f] through Carry
Syntax	RLF f, d 0 ≤ f ≤ FFh d ∈ [0,1]
Operation	[Destination<n+1>] ← [f<n>] [Destination<0>] ← C C ← [f<7>]
Flag Affected	C, Z
Description	[f] is rotated one bit to the left through the Carry bit. If d is 0, the result is stored in the W register. If d is 1, the result is stored back in [f]. 
Cycle	1
Example: **RLF OPERAND, 1**	Before instruction: C = 0 W = 88h, OPERAND = E6h After instruction: C = 1 W = 88h, OPERAND = CCh

SLEEP	Oscillator stop
Syntax	SLEEP
Operation	CPU oscillator is stopped
Flag Affected	PD
Description	CPU oscillator is stopped. CPU can be waked up by external interrupt sources.[1]
Cycle	1

Please make sure all interrupt flags are cleared before running SLEEP; "NOP" command must follow HALT and SLEEP commands.

RRF	Rotate right [f] through Carry
Syntax	RRF f, d 0 ≤ f ≤ FFh d ∈ [0,1]
Operation	[Destination<n-1>] ← [f<n>] [Destination<7>] ← C C ← [f<7>]
Flag Affected	C
Description	[f] is rotated one bit to the right through the Carry bit. If d is 0, the result is stored in the W register. If d is 1, the result is stored back in [f]. 
Cycle	1
Example: **RRF OPERAND, 0**	Before instruction: C = 0 OPERAND = 95h After instruction: C = 1 W = 4Ah, OPERAND = 95h

SUBLW	Subtract W from literal
Syntax	SUBLW k 0 ≤ k ≤ FFh
Operation	[W] ← k – [W]
Flag Affected	C, DC, Z
Description	Subtract the content of the W register from the eight-bit literal "k". The result is stored in the W register.
Cycle	1
Example 1: **SUBLW 02h**	Before instruction: W = 01h After instruction: W = 01h C = 1 Z = 0
Example 2: **SUBLW 02h**	Before instruction: W = 02h After instruction: W = 00h C = 1 Z = 1
Example 3: **SUBLW 02h**	Before instruction: W = 03h After instruction: W = FFh C = 0 Z = 0

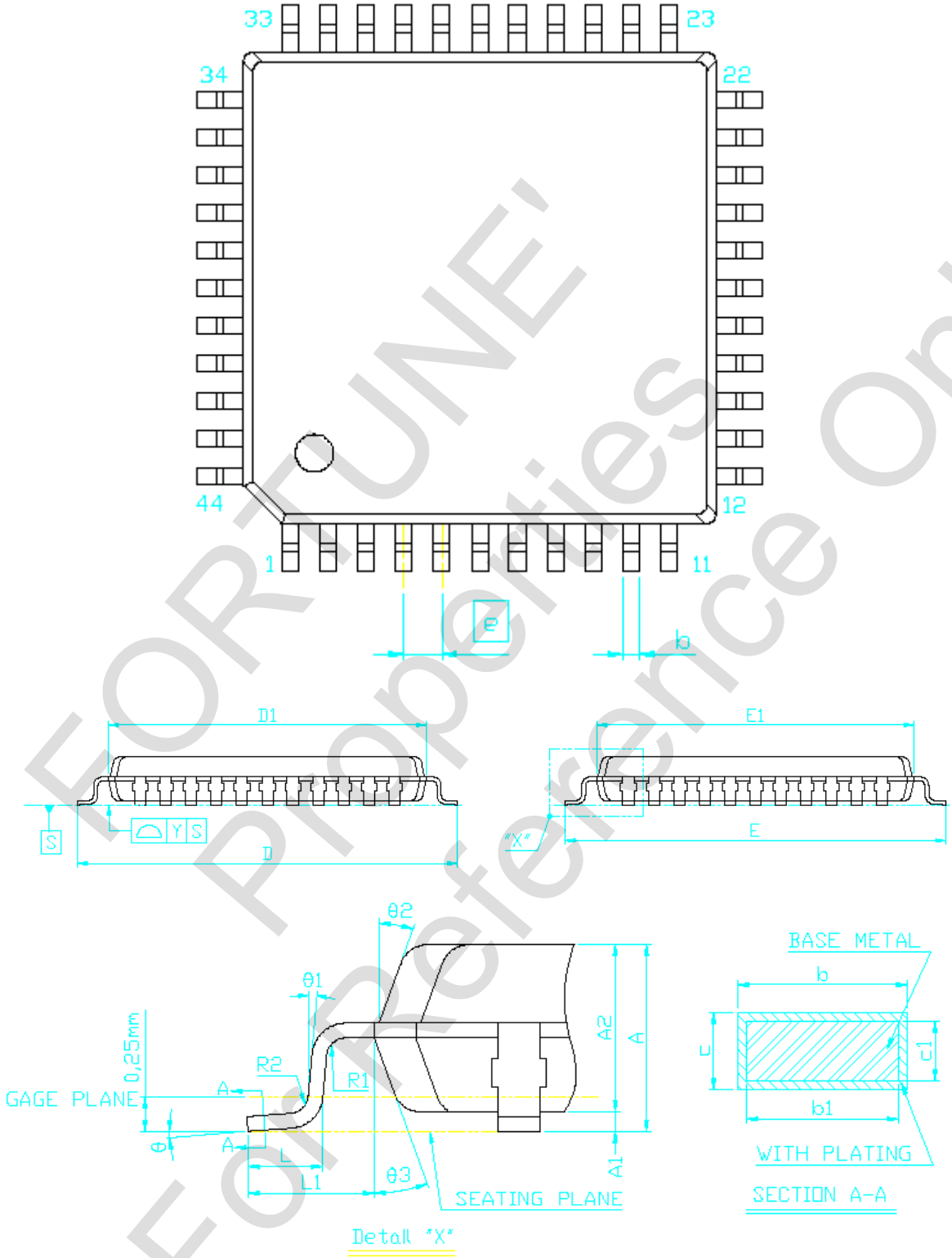
SUBWF	Subtract W from f
Syntax	SUBWF f, d 0 ≤ f ≤ FFh d ∈ [0,1]
Operation	[Destination] ← [f] – [W]
Flag Affected	C, DC, Z
Description	Subtract the content of the W register from [f]. If d is 0, the result is stored in the W register. If d is 1, the result is stored back in [f],
Cycle	1
Example 1: **SUBWF OPERAND, 1**	Before instruction: OPERAND = 33h, W = 01h After instruction: OPERAND = 32h C = 1 Z = 0
Example 2: **SUBWF OPERAND, 1**	Before instruction: OPERAND = 01h, W = 01h After instruction: OPERAND = 00h C = 1 Z = 1
Example 3: **SUBWF OPERAND, 1**	Before instruction: OPERAND = 04h, W = 05h After instruction: OPERAND = FFh C = 0 Z = 0

SUBWFC	Subtract W and Carry from f
Syntax	SUBWFC f, d 0 ≤ f ≤ FFh d ∈ [0,1]
Operation	[Destination] ← [f] – [W] –$\dot{C}$
Flag Affected	C, DC, Z
Description	Subtract the content of the W register from [f]. If d is 0, the result is stored in the W register. If d is 1, the result is stored back in [f].
Cycle	1
Example 1: **SUBWFC OPERAND, 1**	Before instruction: OPERAND = 33h, W = 01h C = 1 After instruction: OPERAND = 32h, C = 1, Z = 0
Example 2: **SUBWFC OPERAND, 1**	Before instruction: OPERAND = 02h, W = 01h C = 0 After instruction: OPERAND = 00h, C = 1, Z = 1
Example 3: **SUBWFC OPERAND, 1**	Before instruction: OPERAND = 04h, W = 05h C = 0 After instruction: OPERAND = FEh, C = 0, Z = 0

XORLW	Exclusive OR literal with W
Syntax	XORLW k 0 ≤ k ≤ FFh
Operation	[W] ← [W] XOR k
Flag Affected	Z
Description	Exclusive OR the content of the W register and the eight-bit literal "k". The result is stored in the W register.
Cycle	1
Example: **XORLW 5Fh**	Before instruction: W = ACh After instruction: W = F3h

XORWF	Exclusive OR W and f
Syntax	XORWF f, d 0 ≤ f ≤ FFh d ∈ [0,1]
Operation	[Destination] ← [W] XOR [f]
Flag Affected	Z
Description	Exclusive OR the content of the W register and [f]. If d is 0, the result is stored in the W register. If d is 1, the result is stored back in [f].
Cycle	1
Example: **XORWF OPERAND, 1**	Before instruction: OPERAND = 5Fh, W = ACh After instruction: OPERAND = F3h

14. Package Information



SYMBOL	DIMENSION (MM)			DIMENSION (MIL)		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A			1,60			63
A1	0,05		0,15	2		6
A2	1,35	1,40	1,45	53	55	57
b	0,30	0,37	0,45	12	15	18
b1	0,30	0,35	0,40	12	14	16
c	0,09		0,20	4		8
c1	0,09		0,16	4		6
D	12,00 BSC			472 BSC		
D1	10,00 BSC			394 BSC		
E	12,00 BSC			472 BSC		
E1	10,00 BSC			394 BSC		
e	0,80 BSC			31,5 BSC		
L	0,45	0,60	0,75	18	24	30
L1	1,00 REF			39 REF		
R1	0,08			3		
R2	0,08		0,20	3		8
Y			0,075			3
θ	0°	3.5°	7°	0°	3.5°	7°
θ1	0°			0°		
θ2	11°	12°	13°	11°	12°	13°
θ3	11°	12°	13°	11°	12°	13°

NOTES:
1.REFER TO JEDEC MS-026/BCB
2.DIMENSION D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
ALLOWABLE PROTRUSION IS 0.25mm PER SIDE D1 AND E1 ARE
MAXIMUM PLASTIC BODY SIZE DIMENSION INCLUDING MOLD MISMATCH.
3.DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE
DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED
THE MAXIMUM b DIMENSION BY MORE THAN 0.08mm.
4.ALL DIMENSIONS IN MILLIMETERS.

15. Revision History

Version	Date	Page	Description
0.1	2005/10	-	Preliminary version.
1.0	2006/11	-	Official release
1.1	2007/02	5. 10 29	Revise Ordering Information Revise Electrical Characteristics Revise Charge Current Control Spec
1.2	2007/03	30	Add 10-13 and 10-14
1.3	2007/04	11 14 21 30 ~ 45	1.Start up/Reset Vector 2. Revise Power System 3. Revise I/O Port 4.Add 11,12,13,14 chapters
1.4	2010/07/06	10	1.WDT 的頻率 typical spec 改成 21KHz 2.IPO、IPU 電流單位是 uA
1.5	2010/11/10	10	VDD=3.6V, SPEC 範圍為 20~32 uA,VDD=5V, SPEC 範圍為 48~68 uA
1.6	2014/05/22	2	Revised company address