Fortune
Semiconductor Corporation
富晶半導體股份有限公司

FS9821

8-bit MCU with 2k program EPROM, 128-byte RAM, 1 low noise OPAMP, 6-ch 14-bit ADC, 4 × 12 LCD driver and RTC.

Data Sheet

TD-0406013

Rev. 1.2

Table of Contents

1. General Description

The FS9821 is a CMOS 8-bit single chip microcontroller(MCU) with embedded a 2kx16 bits one-time programming (OTP) ROM, a 6-channel 14-bit fully differential input analog to digital converter, low noise amplifier, and 4 x 12 LCD driver.

The FS9821 is best suited for applications such as electrical scale, meter, and sensor or transducer measurement application etc.

2. Features

- **8-bit microcontroller, 37 single word instructions.**
- **Embedded 2k x 16 bits program memory, 128-byte data memory.**
- **Voltage operation ranges from 2.2V to 3.6V.**
- **Embedded 1.0 MHz oscillator.**
- **External 32768Hz crystal oscillator (RTC).**
- **Embedded Low Voltage Reset (LVR) and Low Voltage Detector (LVD).**
- **Operation current is less than 4 mA; sleep mode current is about 3μA.**
- **6-level deep hardware stacks.**
- **5 Interrupt sources (external: 3, internal: 2).**
- **6-channel ADC with program output rate and resolution.**
- **Embedded charge pump (voltage doubler) and voltage regulator (3.6V regulated output).**
- **Embedded bandgap voltage reference (typical 1.18V±50mV, 100ppm/°C).**
- **Internal silicon temperature sensor.**
- **Low noise (1μV Vpp without chopper, 0.5μV Vpp with chopper, 0.1Hz~1Hz) OPAMP with chopper controller.**
- **Watchdog timer.**
- **16-bit bi-directional I/O port**
- **PDM (Pulse Density Modulator) output**
- **Buzzer output.**
- **I2C serial I/O port (slave mode only).**
- **4 x 12 LCD drivers.**
- **Package: dice form (57-pin), 64-pin LQFP.**

3. Applications

- **Sensor or transducer measurement applications.**
- **Electrical kitchen scale, personal scale.**
- **Digital meter.**

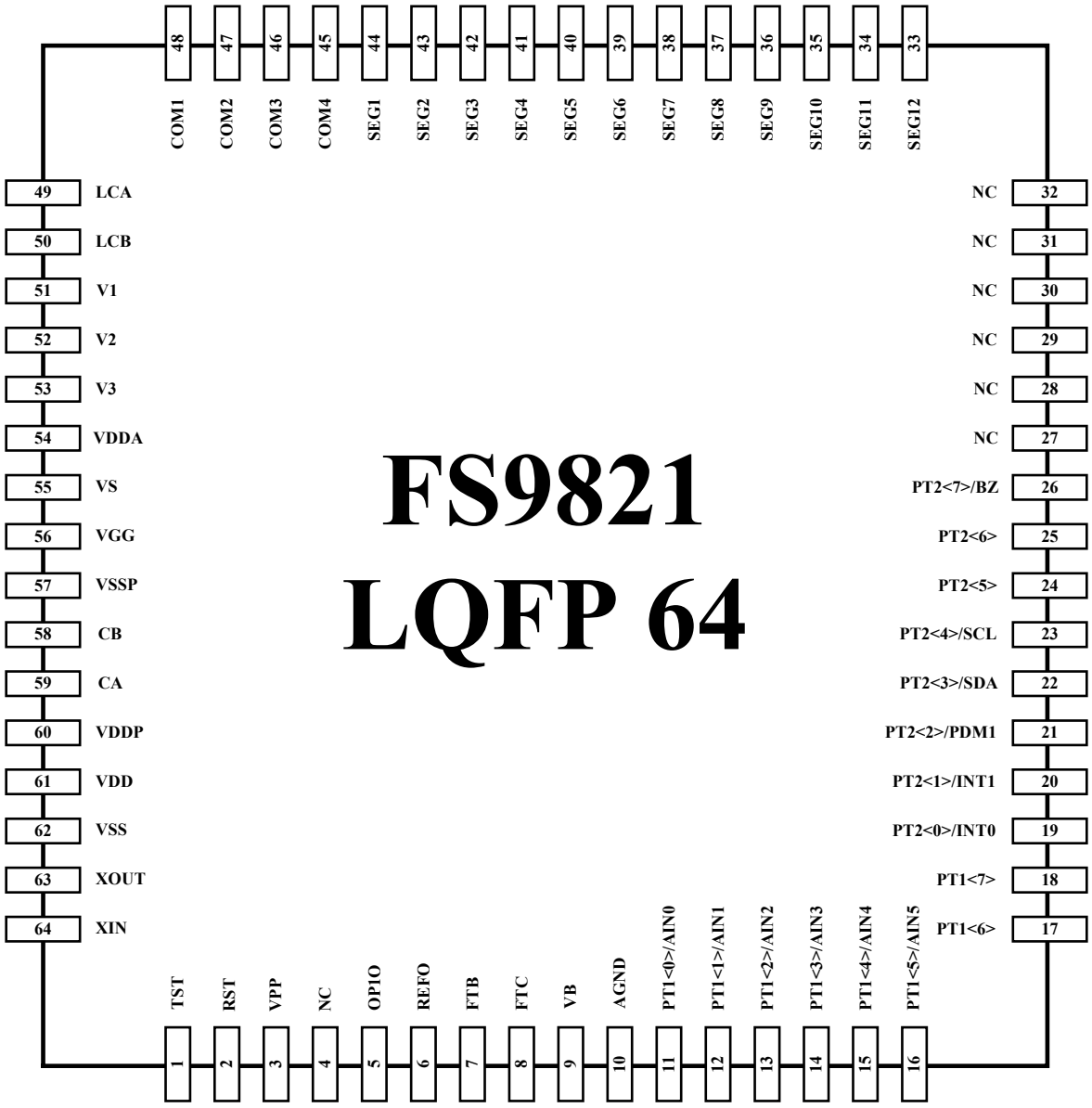
4. Ordering Information

Product Number	Package Type
FS9821-nnnV	Dice form (57-pin), 64-pin LQFP

Note1: Code number (nnnV) is assigned for customer.

Note2: Code number (nnn = 001~999); Version (V = A~Z).

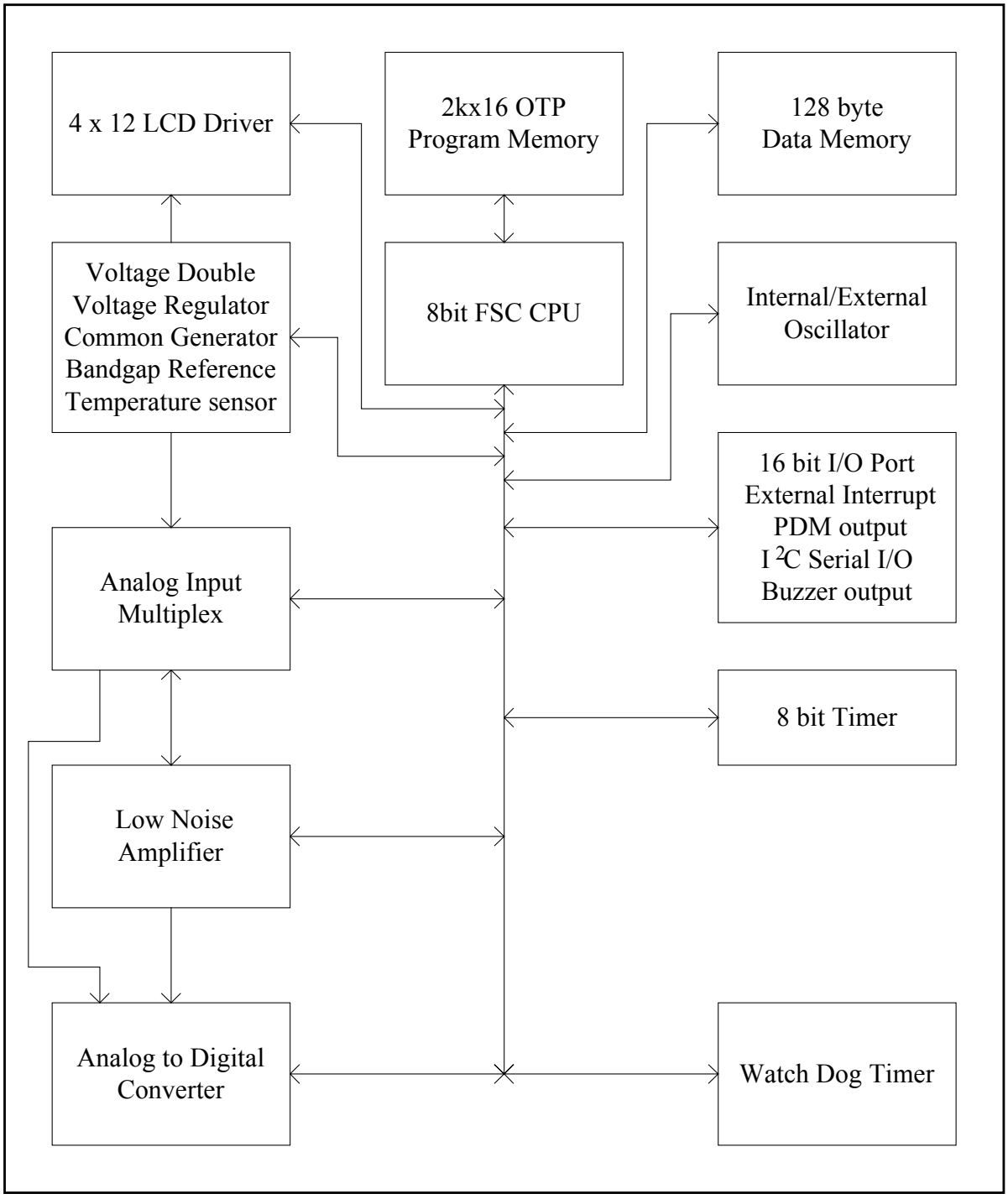
5. Pin Configuration



6. Pin Description

Name	In/Out	Pin No	Description
TST	I	1	Testing Mode
RST	I	2	CPU Reset
VPP	I	3	Programming Power Supply
OP1O	I/O	5	OPAMP 1 Output
REFO	O	6	Band gap Reference Output
FTB, FTC	I/O	7, 8	ADC Pre-Filter Capacitor Connection
VB	I	9	Analog Circuit Bias Current Input
AGND	I/O	10	Analog Ground
PT1<0~5>/AIN0~5	I/O	11~16	Digital I/O Port or Analog input channel
PT2<0~1>/INT0~1	I/O	19~20	Digital I/O Port and External Interrupt input
PT2<2>/PDM1	I/O	21	Digital I/O Port or PDM output
PT2<3>/SDA	I/O	22	Digital I/O Port or I2C serial Bi-Directional data line
PT2<4>/SCL	I/O	23	Digital I/O Port or I2C clock input
PT1<6~7>, PT2<5~6>	I/O	17~18 24~25	Digital I/O Port
PT2<7>/BZ	I/O	26	Digital I/O Port or Buzzer Output
SEG12~SEG1	O	33~44	LCD Segment Driver Output
COM4~COM1	O	45~48	LCD Common Driver Output
LCA	I/O	49	LCD Charge Pump Capacitor Positive Connection
LCB	I/O	50	LCD Charge Pump Capacitor Negative Connection
V3,V2,V1	I/O	51~53	LCD Bias
VDDA	I/O	54	Analog Power Output
VS	I/O	55	Voltage Source from VDDA
VGG	I/O	56	Charge Pump Voltage
VSSP	I	57	Charge Pump Negative Power Supply
CB	I/O	58	Charge Pump Capacitor Negative Connection
CA	I/O	59	Charge Pump Capacitor Positive Connection
VDDP	I	60	Charge Pump Positive Power Supply
VDD	I	61	Positive Power Supply
VSS	I	62	Negative Power Supply (Ground)
XOUT	O	63	32768Hz Oscillator Output
XIN	I	64	32768Hz Oscillator Input
NC	-	-	No Connection

7. Functional Block Diagram



8. Absolute Maximum Ratings

Parameter	Rating	Unit
Supply Voltage to Ground Potential	-0.3 to 5.5	V
Applied Input/Output Voltage	-0.3 to VDD+0.3	V
Ambient Operating Temperature	-10 to +85	°C
Storage Temperature	-55 to +150	°C
Soldering Temperature, Time	260°C, 10 Sec	

9. Electrical Characteristics

9.1 DC Characteristics (VDD=3V, T_A=25℃, unless otherwise noted)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VDD	Recommend Operation Power Voltage		2.2		3.6	V
IDD1	Supply Current 1	MCK=1MHz, CPUCLK=MCK/2, Charge Pump, ADC,OPAMP ON		4		mA
IDD2	Supply Current 2	Internal Oscillator Off, MCK=32768Hz LCD ON.		8	15	µA
IPO	Sleep Mode Supply Current	Sleep Instruction		3		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	Vin=0		20		µA
IOH	High Level Output Current	VOH=VDD-0.3 V		3		mA
IOL	Low Level Output Current	VOL=0.3 V		5		mA
VDDA	Analog Power			3.6		V
IREG	VDDA Regulator Output Current	VDD=3V Internal Voltage Double VDDA=0.95*VDDA(unload)		3		mA
VCVDDA	VDDA Voltage Coefficient		-2		2	%/V
AGND	Analog Ground Voltage			VDDA/2		V
VREF	Build in Reference Voltage	To AGND		1.18		V
TCREF	Build in Reference Voltage Temperature Coefficient	Ta=0~50℃		100		ppm/℃
VLBAT	Low Battery Detector Voltage	S_LB [1:0]=00		2.3		V
		S_LB [1:0]=01		3.5		
VSR	VS Switch Resister			10		Ω
FRC	Internal RC oscillator		0.7	1.0	1.3	MHz
FWDT	Internal WDT Clock			3		kHz

9.2 ADC Characteristics (VDD=3V, 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	0	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

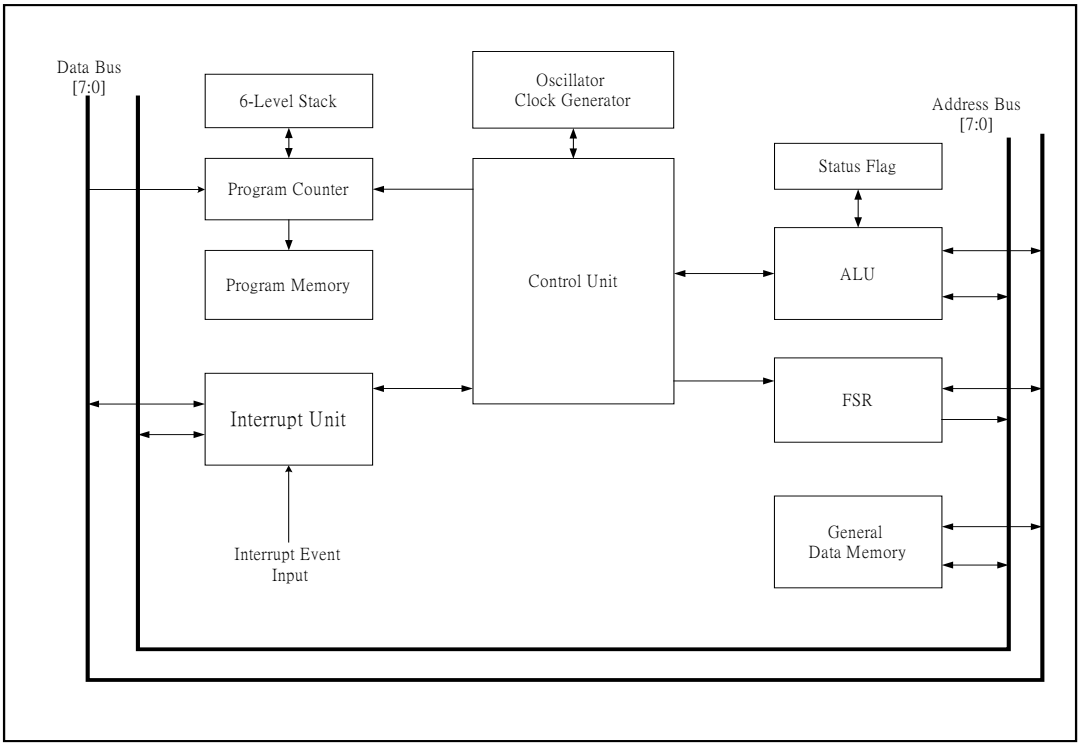
9.3 OPAMP Characteristics (VDD=3V, T_A=25℃, unless otherwise noted)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
	Input Offset			1		mV
	Input Offset Voltage with Chopper	Rs<100Ω		20		μV
	Input Reference Noise	Rs=100Ω, 0.1Hz~1Hz		1.0		μVpp
	Input Reference Noise with Chopper	Rs=100Ω, 0.1Hz~1Hz		0.5		μVpp
	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. Function Description

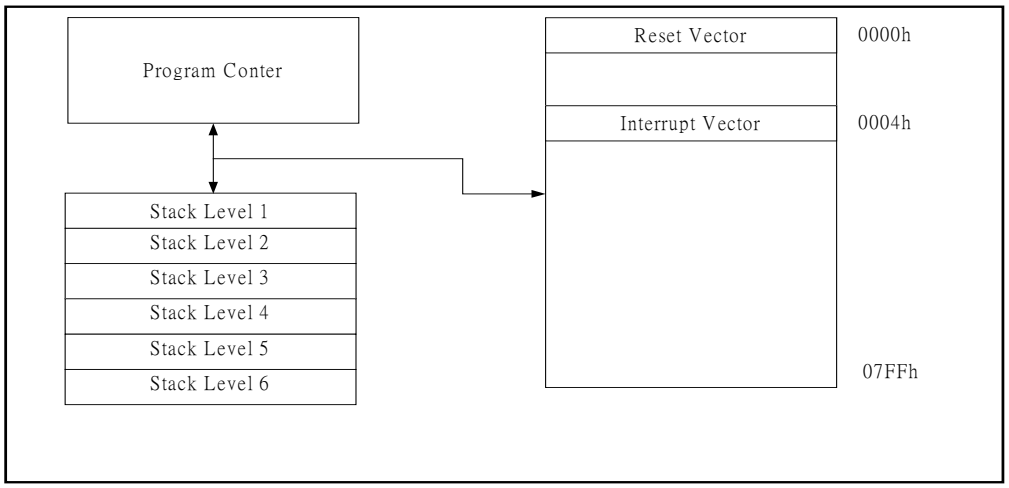
10.1 CPU Core

10.1.1 CPU Core Block Diagram



10.1.2 Program Memory Organization

CPU has an 11-bit program counter capable of address and 2k x 16 program memory space. The reset vector is at 0000h and the interrupt vector is at 0004h.



10.1.3 Data Memory Organization

The data memory is partitioned into three parts. The address 00h~07h areas are system special registers, like indirect address, indirect address pointer, status register, working register, interrupt flag, interrupt control register. The address 08h~7Fh areas are peripheral special registers, like I/O ports, timer, ADC, signal conditional network control register, LCD driver. The address 80h~FFh areas are general data memory.

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				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: PT2.0 External Interrupt flag and enable.
- E1IF, E1IE: PT2.1 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.

10.1.4 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	M7_CK	M6_CK	M5_CK		M3_CK	M2_CK	M1_CK	M0_CK	00000000
15H	PCK		ENPUMP			S_CH1CK [1:0]		S_BEEP	S_PCK	00000000
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
1CH	NETE				ENVS	SILB[1:0]		ENLB		00000000
1DH	NETF		ENBAND	ENVDDA				ENAGND	ENVB	00000000
1FH	SVD								LBOUT	uuuuuuuu
20H	PT1	PT1 [7:0]								uuuuuuuu
21H	PT1EN	PT1EN [7:0]								00000000
22H	PT1PU	PT1PU [7:0]								00000000
23H	AIENB1			AIENB[5:0]						00000000
24H	PT2	PT2 [7:0]								uuuuuuuu
25H	PT2EN	PT2EN [7:0]								00000000
26H	PT2PU	PT2PU [7:0]								00000000
27H	PT2MR	BZEN			PM1EN	E1M[1:0]		E0M[1:0]		00000000
30H	PMD1H	PMD1[15:8]								00000000
31H	PMD1L	PMD1[7:0]								00000000
36H	PMCON				PDMEN		PMCS[2:0]			00000000
37H	PT2OCB				PT2OC[4:3]					uuu11uuu
40H	LCD1	SEG2 [3:0]]				SEG1 [3:0]				uuuuuuuu
41H	LCD2	SEG4 [3:0]				SEG3 [3:0]				uuuuuuuu
42H	LCD3	SEG6 [3:0]				SEG5 [3:0]				uuuuuuuu
43H	LCD4	SEG8 [3:0]				SEG7 [3:0]				uuuuuuuu
44H	LCD5	SEG10 [3:0]				SEG9 [3:0]				uuuuuuuu
45H	LCD6	SEG12 [3:0]				SEG11 [3:0]				uuuuuuuu
54H	LCDENR	LCDCKS [1:0]		LCDEN		LEVEL	LCD_DUTY[1:0]		ENPMPL	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

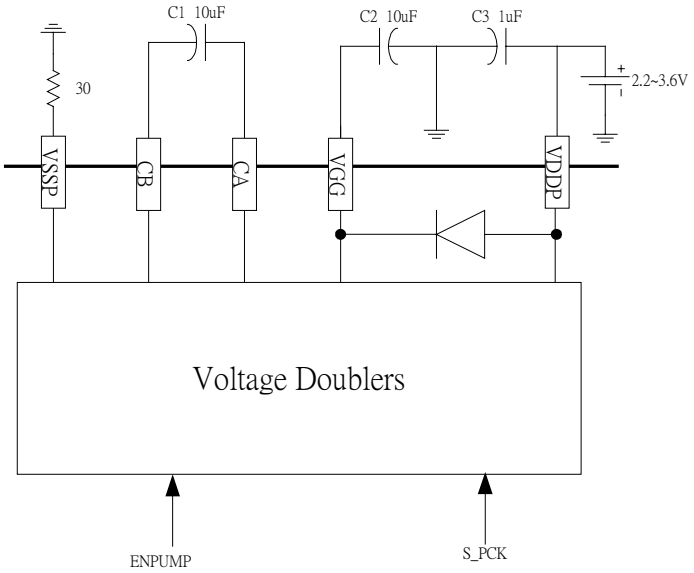
10.1.5 Special Register External Reset (Power On Reset) and WDT Reset State

Address	Name	External Reset	WDT Reset
04H	STATUS	00u00uuu	uuuu1uuu
0DH	WDTCON	00000000	uuuuuuuu
20H	PT1	00000000	uuuuuuuu
21H	PT1EN	00000000	uuuuuuuu
22H	PT1PU	00000000	uuuuuuuu
23H	AIENB1	00000000	uuuuuuuu
24H	PT2	00000000	uuuuuuuu
25H	PT2EN	00000000	uuuuuuuu
26H	PT2PU	00000000	uuuuuuuu
27H	PT2MR	00000000	uuuuuuuu
37H	PT2OC	uuu11uuu	uuuuuuuu
57H	I2CCON	0001uuuu	uuuuuuuu
58H	I2CSTA	uu0000u0	uuuuuuuu
59H	I2CADD	00000000	uuuuuuuu
5AH	I2CBUF	00000000	uuuuuuuu

10.2 Power System

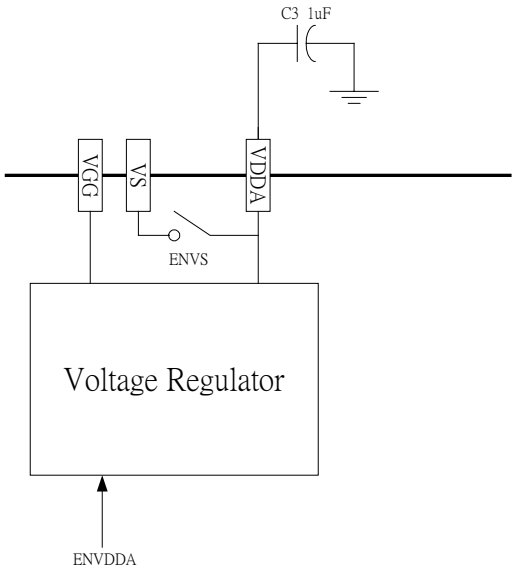
Address	Name	Content (u mean unknown or unchanged)							Reset State
15H	PCK		ENPUMP			-	-	S_PCK	00000000
1CH	NETE				ENVS	SILB[1:0]	ENLB		00000000
1DH	NETF		ENBAND	ENVDDA			ENAGND	ENVB	00000000
1FH	SVD							LBOUT	uuuuuuuu

10.2.1 Voltage Doubler



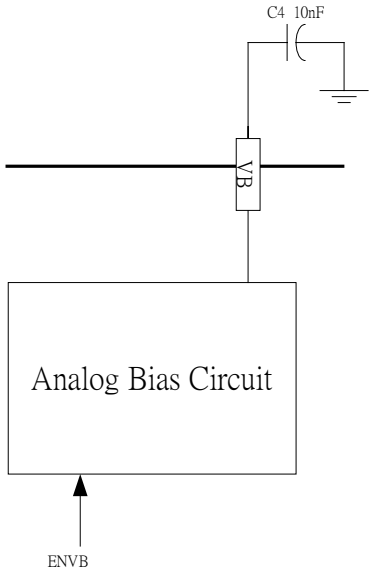
- VGG is VDDA voltage regulator input. When ENPUMP=1, voltage doubler active. The VGG voltage is about two times VDDP. When ENPUMP=0, you can input a voltage as voltage regulator power supply.
- Voltage doubler operation frequency is selected by S_PCK. The details will be described in **Clock System** section.

10.2.2 Voltage Regulator



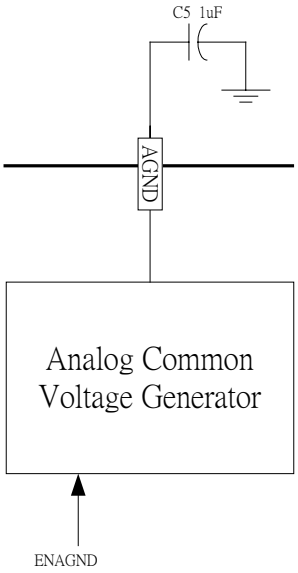
- VDDA is the power supply voltage for analog circuit and LCD driver. When ENVDDA is set, the voltage regulator will active, and VDDA=3.6V. Otherwise VDDA can be as external regulated power supply input.
- VS is the voltage source from VDDA. When ENVS = 1, the switch is short.

10.2.3 Analog Bias Circuit



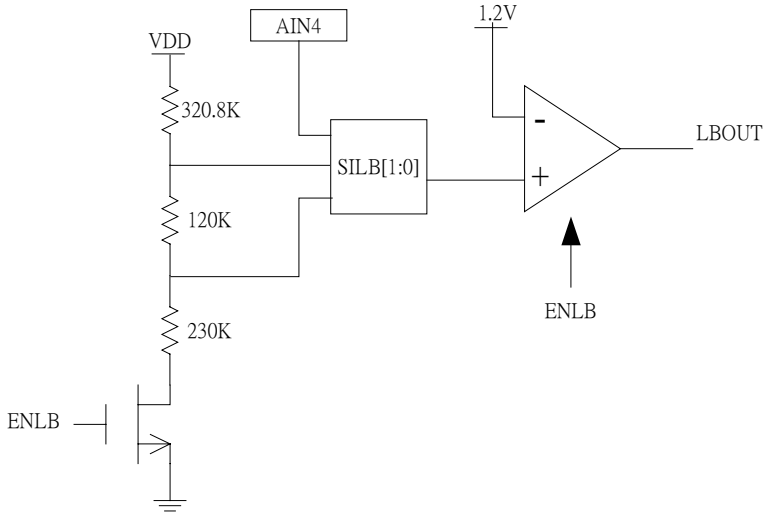
- Before you active analog block, you must set ENVB. When you use internal voltage doubler. Pin VB must be connected a 10nF capacitor to VSS for reducing voltage doubler noise.

10.2.4 Analog Common Voltage Generator



- AGND is analog common voltage. When ENAGND=1, analog common voltage generator will active. AGND = 1/2 VDDA

10.2.5 Low Battery Comparator

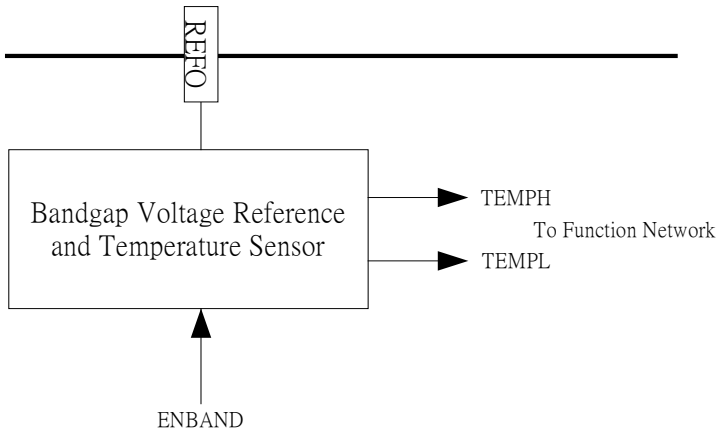


- When ENLB=1, low battery comparator will active.
- SILB [1:0]: Detect input select.

SILB [1:0]	Detect Voltage

00	VDD< 2.45
01	VDD < 3.65
10	AIN < 1.2

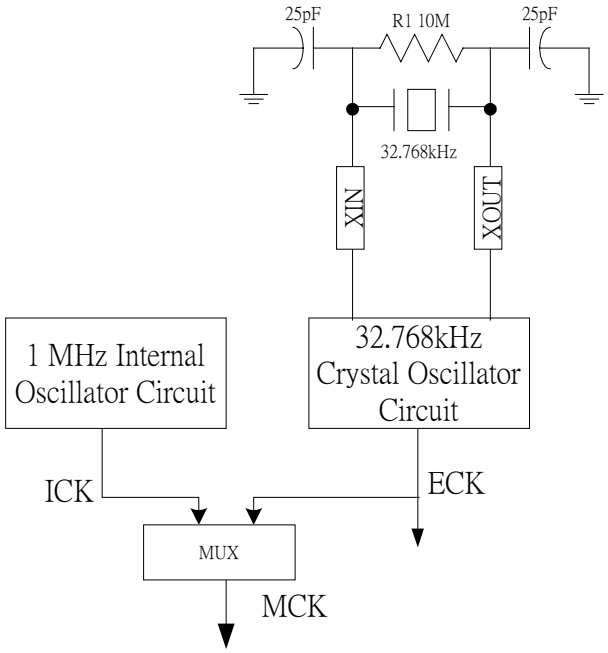
10.2.6 Bandgap Voltage and Temperature Sensor



- REFO is low temperature coefficient bandgap voltage reference output. When ENBAND=1, the circuit will active. The output voltage to AGND is about 1.18V. Typical temperature coefficient is 100ppm/°C.
- {TEMPH, TEMPL} is proportion to ambient temperature. You can select them to ADC input and transfer to digital code. (Typical 550μV±50μV/°C)

10.3 Clock System

Address	Name	Content (u mean unknown or unchanged)								Reset State
14H	MCK	M7_CK	M6_CK	M5_CK		M3_CK	M2_CK	M1_CK	M0_CK	00000000
15H	PCK		-			S_CH1CK [1:0]		S_BEEP	S_PCK	00000000



10.3.1 Oscillator State

Input			Oscillator State	
Sleep	M7_CK	M6_CK	Internal	External
1	X	X	Disable	Disable
0	0	0	Enable	Enable
0	0	1	Enable	Disable
0	1	0	Disable	Enable
0	1	1	Enable	Disable

- MCK and CLK

M3_CK	M0_CK	MCK	M1_CK	CLK
X	0	ICK	0	MCK
0	1	ECK	1	MCK/4
1	1	ECK/2		

10.3.2 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 M0_CK, M1_CK,M2_CK,M3_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
0	0	MCK/6.5

0	1	MCK/12.5
1	0	MCK/2
1	1	MCK/4

10.3.3 ADC Sample Frequency

M1_CK	ADC sample Frequency (ADCF)
0	MCK/25
1	MCK/50

10.3.4 Beeper Clock

M0_CK	S_BEEP	Beeper Clock
X	0	CLK/250
0	1	CLK/375
1	1	ECK/8

10.3.5 Voltage doubler Operation Frequency

M0_CK	S_PCK	Voltage doubler Operation Frequency
0	0	MCK/200
0	1	MCK/100
1	X	ECK/32

10.3.6 Chopper Operation Amplifier Input Control Signal

S_CH1CK [1]	S_CH1CK [0]	Chopper Control Signal
0	0	0
0	1	1
1	0	CLK/500
1	1	CLK/1000

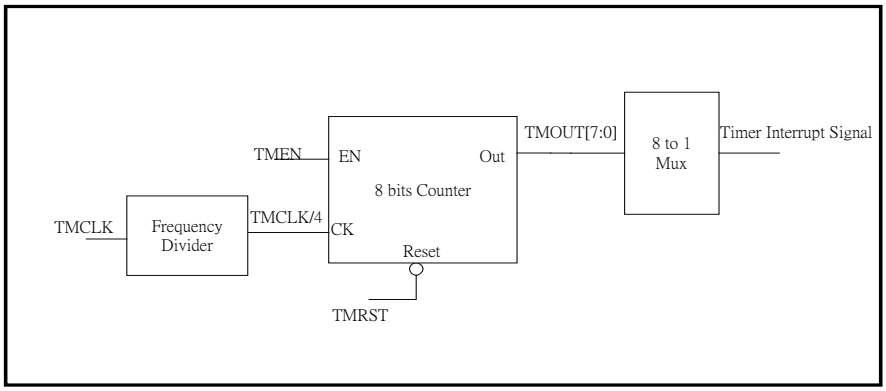
10.3.7 Timer and LCD Module Input Clock

M5_CK	Timer and LCD Module input Clock TMCLK
0	CLK/1000
1	ECK/32

10.4 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

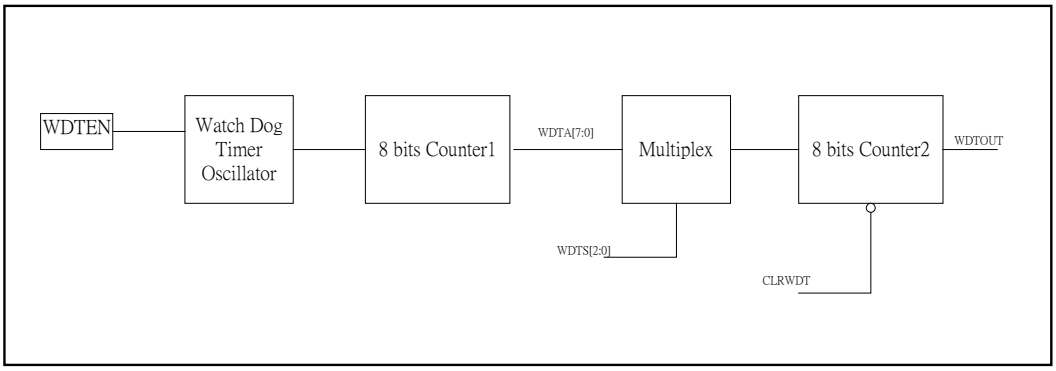


- Write a “0” to bit 7 of address 0Fh; the CPU will send a low pulse to TRST and reset the 8-bit counter. Then read bit 7 of TMCON to get “1”.
- TMEN=1, the 8-bit counter will be enabled. TMEN=0, the 8-bit counter will stop.
- TMOUT [7:0] is the output of the 8-bit counter. It is read-only.
- INS [2:0] selects timer interrupt source. The selection codes are as follows:

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

10.5 Watch Dog Timer

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



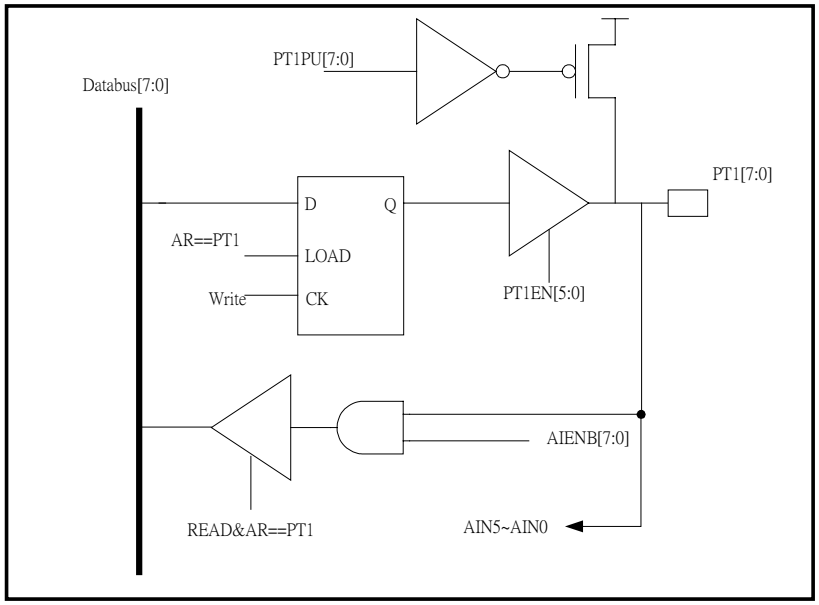
- WDTEN = “1” : enable watchdog timer oscillator. “0” : watchdog timer function will be disabled. WDTEN write “1” only.
- When WDT Counter 2 overflows, it will send WDTOUT to reset the CPU and set TO flag.
- CLRWDT instruction will reset WDT Counter 2
- WTS [2:0] selects WDT Counter 2 and the code selections are as follows, 111: WDTA [0], 110: WDTA [1], 101: WDTA [2], 100: WDTA [3], 011: WDTA [4], 010: WDTA [5], 001: WDTA [6], 000: WDTA [7].

10.6 I/O Port

Address	Name	Content (u mean unknown or unchanged)								Reset State
06H	INTF				-	I2CIF	-	E1IF	E0IF	00000000
07H	INTE	GIE			-	I2CIE	-	E1IE	E0IE	00000000
20H	PT1	PT1 [7:0]								uuuuuuuu
21H	PT1EN	PT1EN [7:0]								00000000
22H	PT1PU	PT1PU [7:0]								00000000
23H	AIENB1	AIENB[7:0]								00000000
24H	PT2	PT2 [7:0]								uuuuuuuu
25H	PT2EN	PT2EN [7:0]								00000000
26H	PT2PU	PT2PU [7:0]								00000000
27H	PT2MR	BZEN			PM1EN	E1M[1:0]		E0M[1:0]		00000000
37H	PT2OCB				PT2OC[4:3]					uuu11uuu

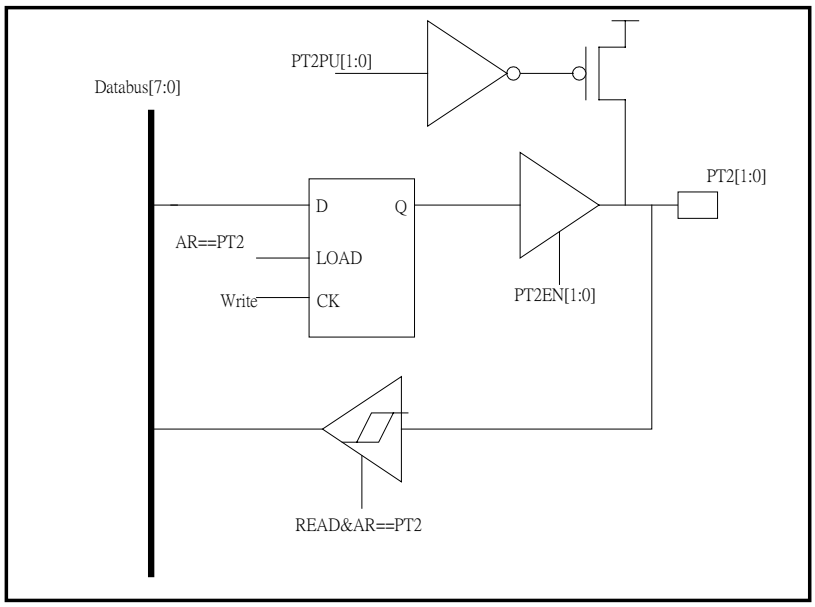
- I/O ports with pull-up resistor enable control. PT1(2)PU [N]=”0”: PT1(2) [N] without pull-up resistor, “1”: PT1(2) [N] with pull-up resistor
- PT1(2)EN [N] =”0”: PT1(2) [N] is as input port, “1”: PT1(2) [N] is as output port
- PT1(2) is the data register of I/O port.

10.6.1 Digital I/O Port with Analog Input Channel Shared: PT1<0> ~ PT1<5>



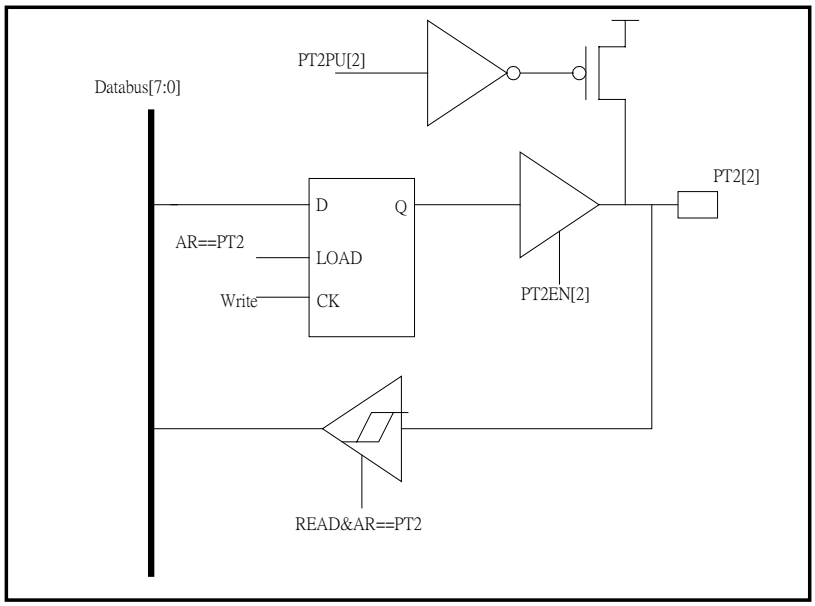
- AIENB [N] =”0”, this port is Analog input channel (AIN0 ~ AIN5), “1”: This port is Digital I/O port.
- The digital I/O port PT1 [6], PT1 [7] is active when AINENB [6], AINENB [7] set”1”,
- The VDDA Regulator must enable first then the AIN0~AIN5 will work normal, otherwise the AIN0~AIN5 work abnormal due to the parasitic diode which between AIN0~AIN5 and VDDA is active. When I/O set “1” and the leakage current will be happened.
- If we want to keep low operation current in sleep mode.
  1. AINENB=1 and PT1 is floating or pull down.
  2. PT2 is VDD or VSS state.

10.6.2 Digital I/O Port and External Interrupt Input : PT2<0>, PT2<1>



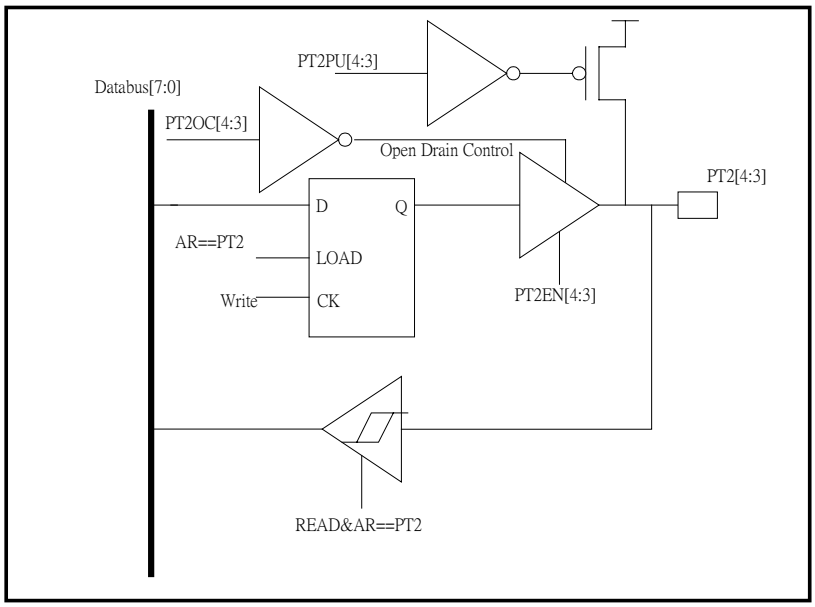
- PT2<0>/INT0, PT2<1>/INT1 can be as external interrupt sources. Interrupt mode is controlled by E0 (2) M [1:0] =”00”: negative edge, “01”: positive edge, “10”&”11”: interrupt when change.
- There has Schmitt-trigger input.

10.6.3 Digital I/O Port or PDM Output : PT2<2>



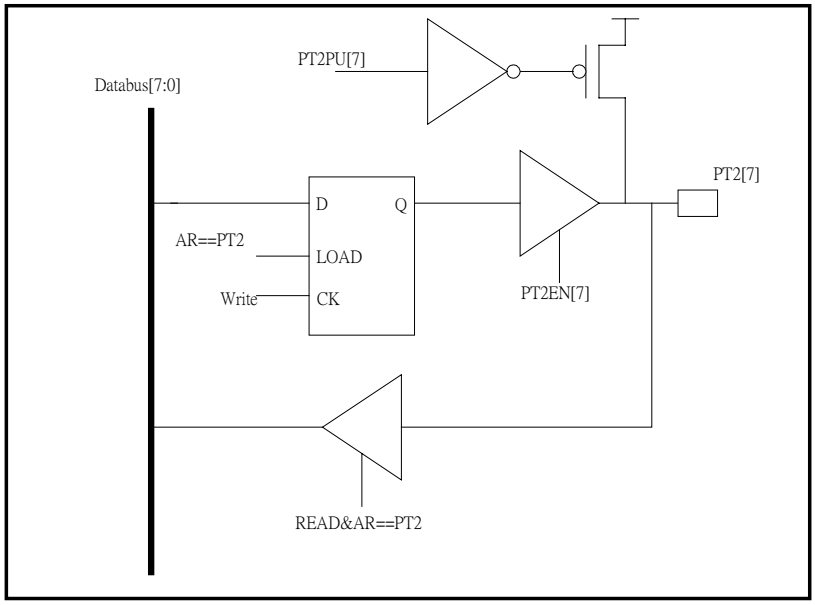
- PT2<2> has Schmitt-trigger input.
- When PM1EN=”1” and PT2EN [2] =”1”, PT2<2> is PDM Output.
- PDM details see the section PDM (Pulse Density Modulator) Module.

10.6.4 Digital I/O Port or I2C Serial Port : PT2<3>/SDA, PT2<4>/SCL



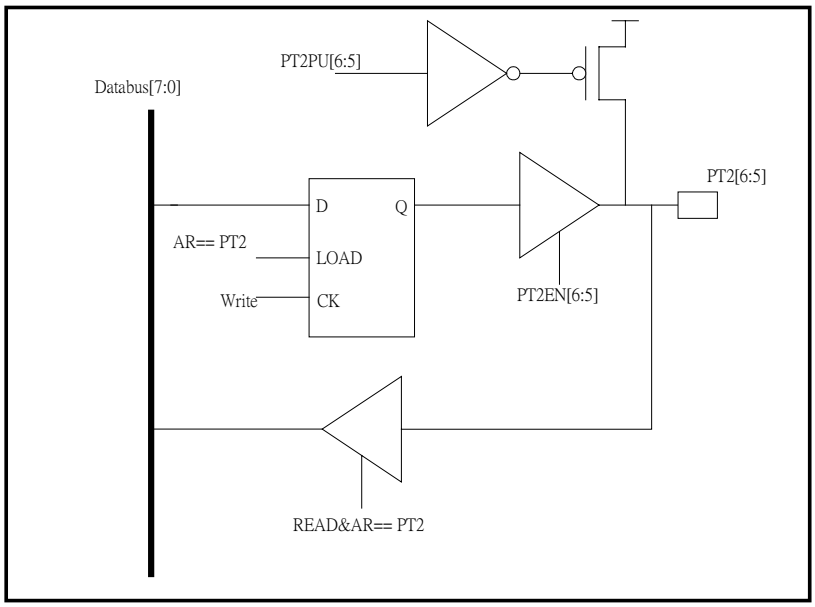
- When PT2OC [3(4)] =”1”: PT2 [3(4)] is open-drain;”0”: PT2 [3(4)] is normal digital I/O port.
- There has Schmitt-trigger input.
- I2C details see section I2C module.

10.6.5 Digital I/O Port or Buzzer Output : PT2<7>



- PT2EN [7] =”1” and BZEN=”1”, PT2 [7] as buzzer output.

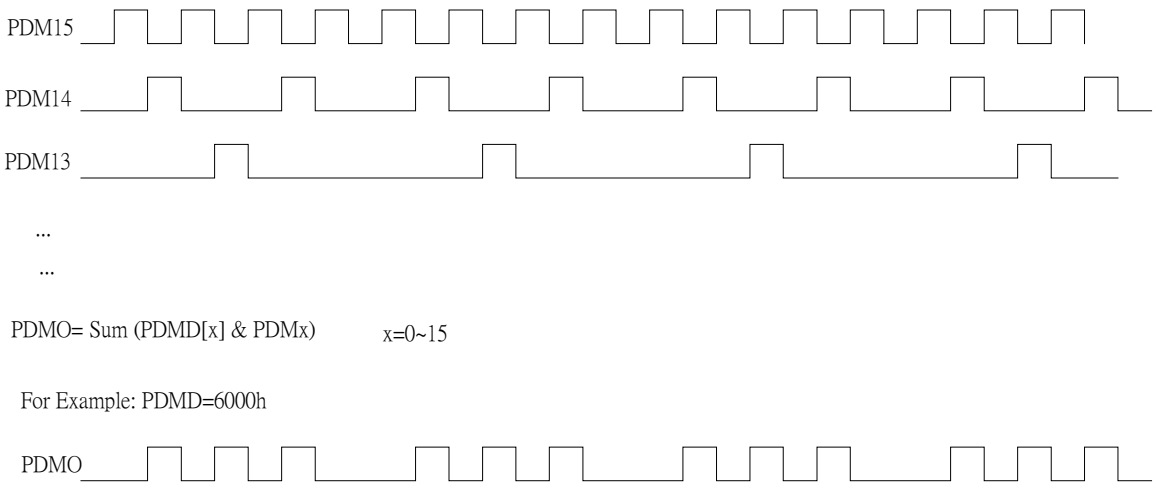
10.6.6 Digital I/O Port : PT2<5 > ~ PT2<6>



10.7 PDM (Pulse Density Modulator) Module

Address	Name	Content (u mean unknown or unchanged)						Reset State
27H	PT2MR	-			PM1EN	-	-	00000000
30H	PMD1H	PMD1[15:8]						00000000
31H	PMD1L	PMD1[7:0]						00000000
36H	PMCON				PDMEN		PMCS[2:0]	00000000

- PDM is another method to implement the function as PWM does, but offers better energy transportation in each short period that user wants within the 16-bit period of time than PWM. For example, we will demonstrate a 16-bit PWM and a 16-bit PDM with the same setting for user's better understanding.
- Below is the wave form chart of PDM definition with an example.



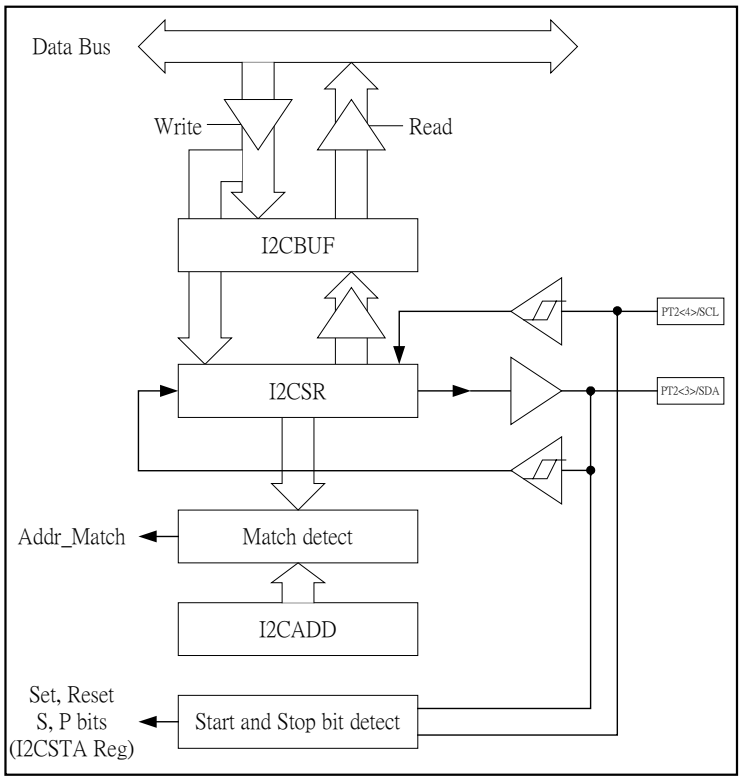
- From above definition, we may know that PDMD[15] represents the same energy weighting of PWM[15] in the 16-bit period of time. PDMD[15] can generate the same counts of positive pulse, 32,768 counts (PDMD[15] = 1 = PWM[15]) or 0 count (PDMD[15] = 0 = PWM[15]) as PWM[15] does in the 16-bit period of time. Also, PDMD[14] can generate the same counts of positive pulse, 16,384 counts (PDMD[14] = 1 = PWM[14]) or 0 count (PDMD[14] = 0 = PWM[14]) as PWM[14] does; PDMD[13] can generate the same counts of positive pulse, 8,192 counts (PDMD[13] = 1 = PWM[13]) or 0 count (PDMD[13] = 0 = PDMD[13]) as PWM[13], and so on. Then, we know that we may get the same energy weighting (or counts of positive pulse) in the 16-bit period of time if we set the same value on PDMD[15:0] and PWM[15:0].
- If we zoom in to the 8-bit period of time from the beginning within the 16-bit period with the setting of PDMD[15:0]= “1000-0000-0000-0000B” = PWM[15:0], we will see that PDM offers better energy transportation that user wants than PWM does. PDM still offers half energy (128 counts of positive pulse) within the 8-bit period of time from the beginning within the 16-bit period, but PWM offers full energy (256 counts of positive pulse) within the same period.
- PMEN: Enable PDM Module. When PM1EN=”1” and PT2EN [2] =”1”, PT2<2> is PDM Output.
- PMCS: Select Input Frequency

PWCS	PDM Pulse Width
000	1/MCK
001	2/MCK
010	4/MCK
011	8/MCK
100	16/MCK
101	32/MCK
110	64/MCK
111	128/MCK

10.8 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 PT2<4>/SCL pin, which is the clock, and the PT2<3>/SDA pin, which is the data. The user must configure these pins as open-drain through the PTOCB[4:3]. 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

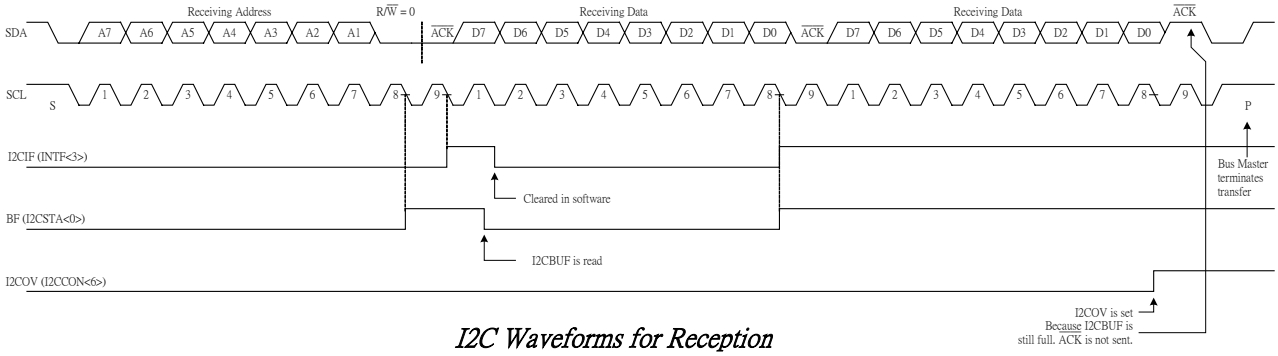
DA : Data/Address bit
1 = indicates that the last byte received was data
0 = indicates that the last byte received was address

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

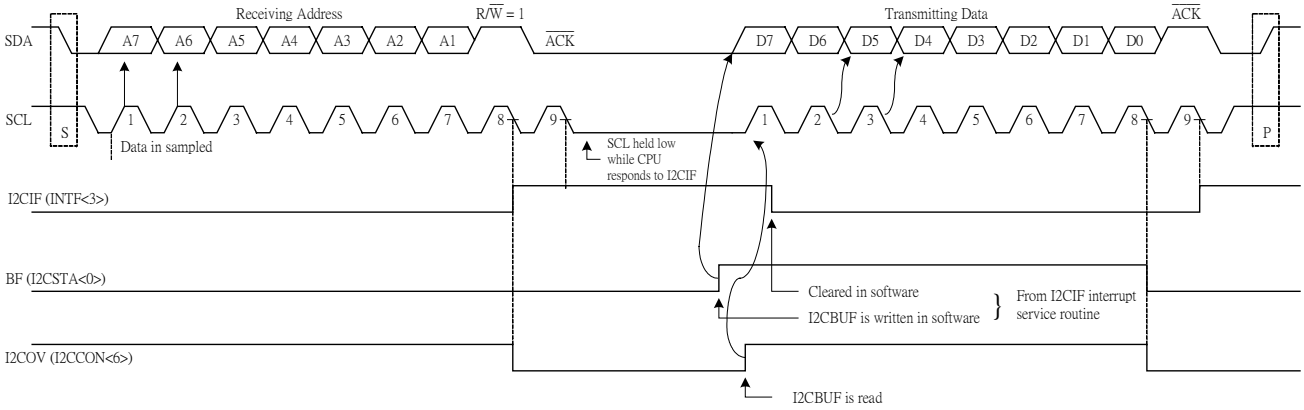
- I2CBUF is the BUFFER REGISTER of I2C module
- I2CADD is the ADDRESS REGISTER of I2C module
- Reception: When the R/W bit of the address byte is clear and an address match occurs, the R/W bit of the I2CSTA register is cleared. The received address is loaded into the I2CBUF. When the address byte overflow 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

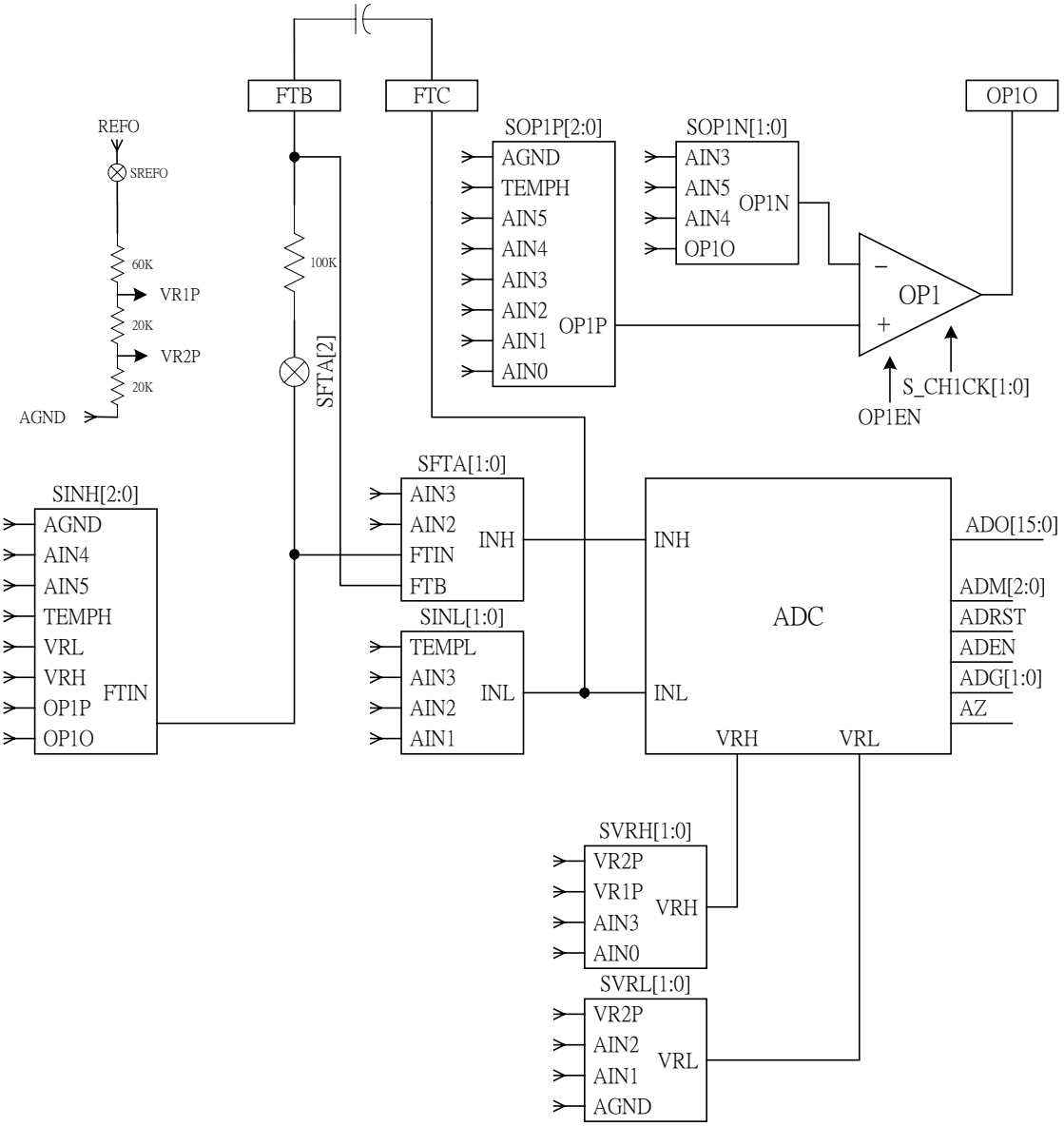
on the falling edge of the ninth clock pulse. As a slave-transmitter, the ACK pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line was high (not ACK), then the data transfer is complete. The slave then monitors for another occurrence of the I2CSTA 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

10.9 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



10.9.1 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.

10.9.2 OPAMP : OP1

- OP1EN is the OPAMP enable control signal.
- S_CH1CK [1:0] see "3.6. Chopper Operation Amplifier Input Control Signal". Can set OP1 input operation mode as follows, 00: +Offset, 01: -Offset, 10: CLK/500 chopper frequency, 11: CLK/1000 Chopper frequency.

10.9.3 Analog Multiplex :

- Low Pass Filter Input:

SINH[2:0]	000	001	010	011	100	101	110	111
Select	OP1O	OP1P	VRH	VRL	TEMPH	AIN5	AIN4	AGND

- ADC Negative Input:

SINL[1:0]	00	01	10	11
Select	AIN1	AIN2	AIN3	TEMPL

- Low Pass Filter Output, ADC Positive Input:

SFTA[1:0]	00	01	10	11
Select	FTB	FTIN	AIN2	AIN3

- External Filter Control: SFTA [2] =1, FTIN and FTB short; SFTA [2] =0, FTIN and FTB open.
- Internal Reference Voltage Control: SREFO=1, REFO and VRP short; SREFO=0, REFO and VRP open.
- ADC Reference Voltage Negative Input:

SVRL[1:0]	00	01	10	11
Select	AGID	AIN1	AIN2	VR2P

- ADC Reference Voltage Positive Input:

SVRH[1:0]	00	01	10	11
Select	AIN0	AIN3	VR1P	VR2P

- OP1 Positive Input:

SOP1P[2:0]	000	001	010	011	100	101	110	111
Select	AIN0	AIN1	AIN2	AIN3	AIN4	AIN5	TEMPH	AGND

- OP1 Negative Input:

SOP1N[1:0]	00	01	10	11
Select	OP1O	AIN4	AIN5	AIN3

10.10 ADC Application Guide

The ADC used in FS9821 is a ∑-Δ ADC with fully differential inputs and fully differential reference voltage inputs. Its maximum output is ±15625. The conversion equation is as follows:

Dout= 15625 *G * (VIH-VIL+Vio) / (VRH-VRL+Vro)

VIH is ADC's positive input voltage, VIL is ADC's negative input voltage, and Vio is ADC's offset on the input terminals, VRH is the voltage at the positive input of Reference Voltage, VRL is the voltage at the negative input of Reference Voltage, and Vro is the offset on the input terminals of Reference Voltage. Where VRH-VRL+Vro>0. When G * (VIH-VIL+Vio) / (VRH-VRL+Vro) ≥ 1, Dout=15625. When G * (VIH-VIL+Vio) / (VRH-VRL+Vro) ≤ -1, Dout=-15625.

10.10.1 ADC Output Format

CPU can read {ADOH, ADOL} as ADC's 16-bit output. Note that the output is in 2's complement format, i.e., "1" in the most significant bit (MSB) denotes a negative number. For example, if {ADOH, ADOL} =E2F7h, then Dout= -(not (E2F7h) +1) = -7433.

10.10.2 ADC Linear Range.

ADC is close to saturation when G * (VIH-VIL+Vio) / (VRH-VRL+Vro) is close to ±1, and has good linearity in the range of ±0.95.

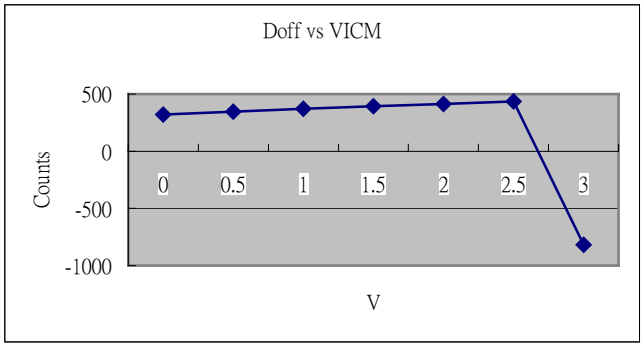
10.10.3 ADC Output Rate and Settling Time

∑Δ ADC is generally an over-sampling ADC, i.e., every ADC output is the results of sampling N times and processed by DSP. FS9821 ADC sampling frequency is decided by M1_CK. ADCM decides to send out a 16-bit output after sampling N times and an interrupt signal every time the ADC changes its output. In fact, every ADC output includes previous 2*N times sampling results. Generally speaking, if ADC inputs, reference voltage, ADG, AZ are switched, the previous two outputs are normally not stable ones, the third output and beyond are stable.

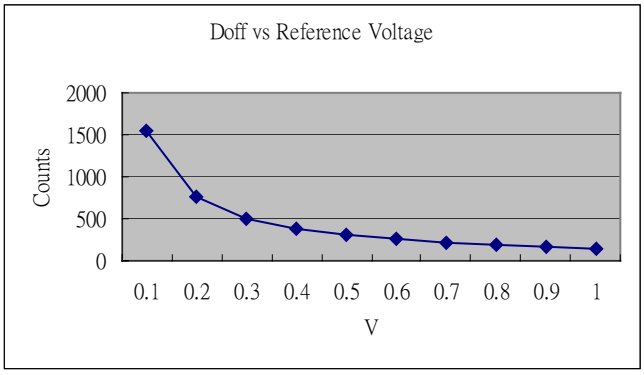
10.10.4 ADC Input Offset

ADC Input Offset Vio drifts with temperature and common mode voltage at the inputs. When the drifting is slow, set AZ bit to 1, and Doff= 15625 *G * (Vio) / (VRH-VRL+Vro). When measuring, Doff should be deducted. The relationships of Doff with voltage inputs of common mode and reference voltage are shown as follows.

- (VRH, AGND) =0.4V, VRL=AGND, VIH=VIL=VICM ADG=01



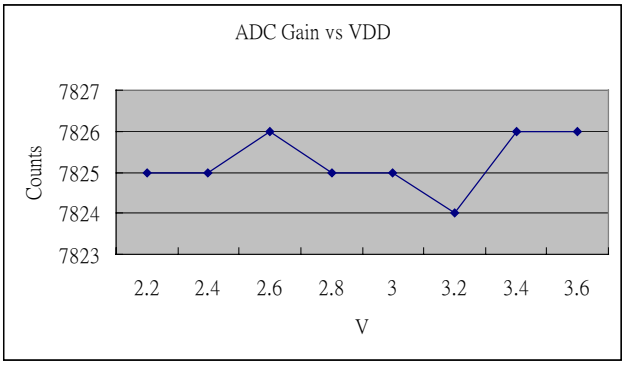
- (VRH, AGND) =0.5~1V, VRL=AGND, VIH=VIL=AGND ADG=01



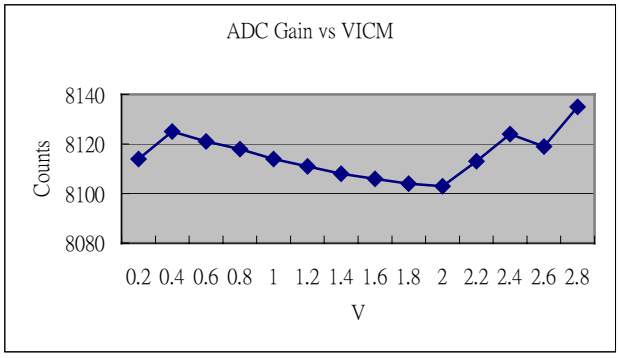
10.10.5 ADC Gain

ADC output deducted by Doff is ADC Gain. Within ADC operation range, the changes of ADC Gain are shown as follows. The results show that ADC Gain does not change as VDD changes. The suggested values for common mode voltages at ADC input and reference voltage are 1V~2V.

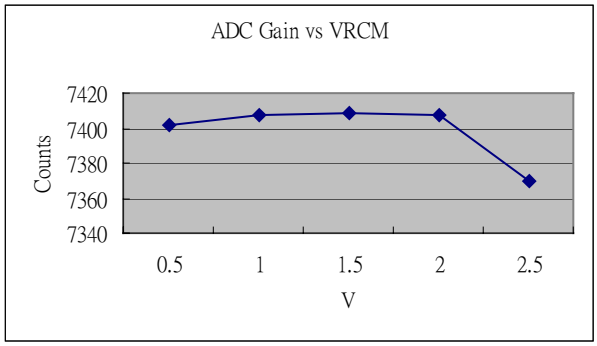
- ADC Gain vs. VDD: (VRH, VRL) = 1/3(REFO, AGND), (VIH, VIL) = 1/6(REFO, AGND), VRL=VIL=AGND ADG=01 VDD=2.2V~3.6V



- ADC Gain vs. Voltage Inputs of Common Mode: (VRH, VRL) =1/3(REFO, AGND), (VIH, VIL) =0.2V, VRL=AGND VICM=1/2(VIH+VIL) ADG=01



- ADC Gain vs. Voltage Reference of Common Mode: (VRH, VRL) =0.4, (VIH, VIL) =1/6(REFO, AGND), VRCM=1/2(VRL+VRH) VIL=AGND ADG=01



10.10.6 ADC Resolution

ADC resolution is mainly decided by ADCM (ADC out rate) and reference voltage, and the results are as follows:

- (VRH, VRL) =0.4V, (VIH, VIL) =0.2V, VRL=VIL=AGND. G=1

ADM	000	001	010	011	100	101	110
Rolling counts	10	6	4	3	3	2	1

- (VRH, VRL) =VR, (VIH, VIL) =1/2 VR, VRL=VIL=AGND. G=1 ADM=101

VR	0.05	0.1	0.2	0.3	0.4	0.6	0.8	1.0
Rolling counts	31	15	5	3	2	2	4	9

10.11 Low Noise Operation Amplifier Guide

The input noise of CMOS OPAMP is generally much larger than the one of Bipolar OPAMP. Moreover, the flick noise (1/f noise) of CMOS is a killer for low frequency small signal measurement. But the need for input bias current in Bipolar OPAMP causes that some transducers can not be used. In general, bipolar process is not good for highly integrated ICs. FS9821 used special CMOS low noise circuit design, and under normal conditions, the input noise is controlled under 1μVpp (0.1Hz~1Hz). FS9821 is good for transducer applications because there is no need to consider input bias current.

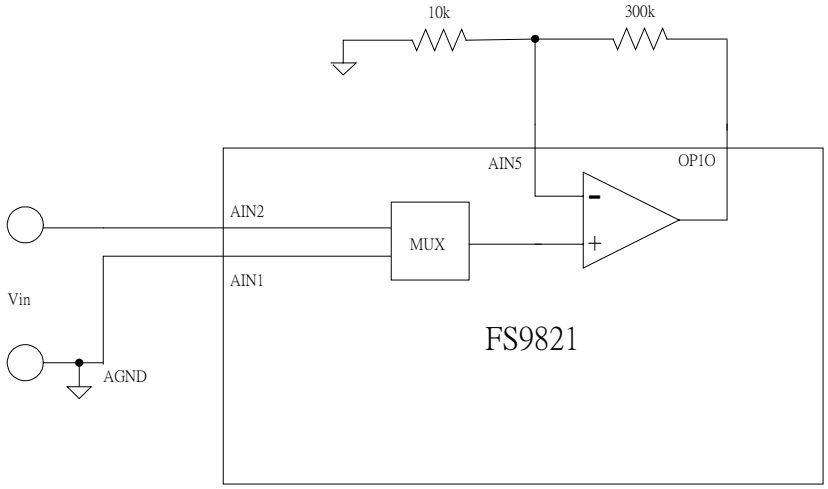
Most of the input noise in CMOS OPAMP comes from input differential amplification. S_CHCK can be set to switch the differential amplification: 00 for positive Offset Voltage, 01 for negative Offset voltage. When using one clock pulse to switch input differential amplification, that is called chopper mode. In general, chopper frequency is set between 1 kHz and 2 KHz.

Under chopper mode, the input noise peak-to-peak voltage in FS9821 is less than 0.5 μV (0.1Hz~1Hz). But an equivalent input current of less than 100pA is generated, due to the effect of switching.

10.11.1 Single End Amplifier Application

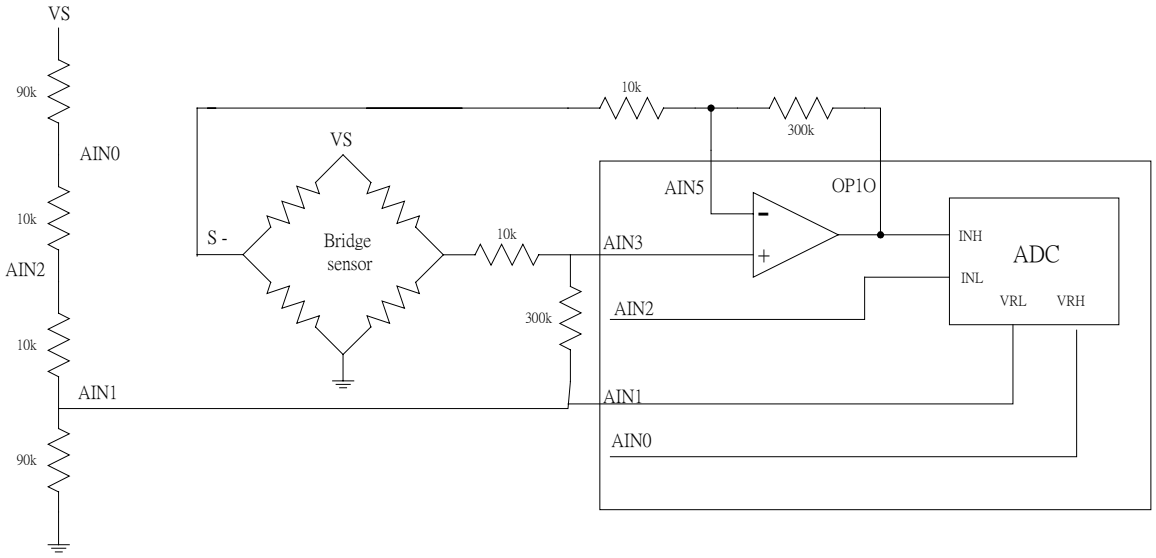
Measurement of small signal usually takes consideration of the drifting of an OPAMP offset voltage. In the Figure below, the negative input is connected to AGND. It is also possible to measure the ADC's negative input and deduct this value; in order to correct the error caused by the Amplifier's offset voltage drifting. Because AGND provides current output in applications, AIN1 is used as negative input measurement point to avoid unnecessary voltage error.

OPAMP input offset is amplified by an amplifier then inputted to ADC. Too much amplification can cause OPAMP output move beyond ADC linear operation range. Hence, under normal conditions, OPAMP amplification should be less than 50 times.



10.11.2 Differential Amplifier

Measurement of differential signal is often used in bridge sensor applications. As shown in the differential amplifier below, VS Pin is used as power input for bridge sensor, ADC reference voltage is also from VS Pin after voltage division. When there is a small change in VS, ADC output does not change. Connecting AIN2 to ADC negative input can adjust the zero point of bridge sensor. When starting chopper mode, the amplification should be less than 100 times.



10.12 LCD Driver

Address	Name	Content (u mean unknown or unchanged)							Reset State
40H	LCD1	SEG2 [3:0]]				SEG1 [3:0]			uuuuuuuu
41H	LCD2	SEG4 [3:0]				SEG3 [3:0]			uuuuuuuu
42H	LCD3	SEG6 [3:0]				SEG5 [3:0]			uuuuuuuu
43H	LCD4	SEG8 [3:0]				SEG7 [3:0]			uuuuuuuu
44H	LCD5	SEG10 [3:0]				SEG9 [3:0]			uuuuuuuu
45H	LCD6	SEG12 [3:0]				SEG11 [3:0]			uuuuuuuu
54H	LCDENR	LCDCKS [1:0]	LCDEN			LEVEL	LCD_DUTY[1:0]	ENPMPL	00000000

- LCDEN =1 will start the LCD clock. LCD1~LCD6 is the LCD display data area.
- ENPMPL: enable LCD charge pump. LEVEL: select LCD bias, “0”: 1/3 bias, “1”: 1/2 bias.
- LCDCKS [1:0] select LCD frame frequency.

LCDCKS [1:0]	LCD frame frequency(1/4 duty)
00	LCD Input Frequecny/8
01	LCD Input Frequecny/16
10	LCD Input Frequecny/32
11	LCD Input Frequecny/64

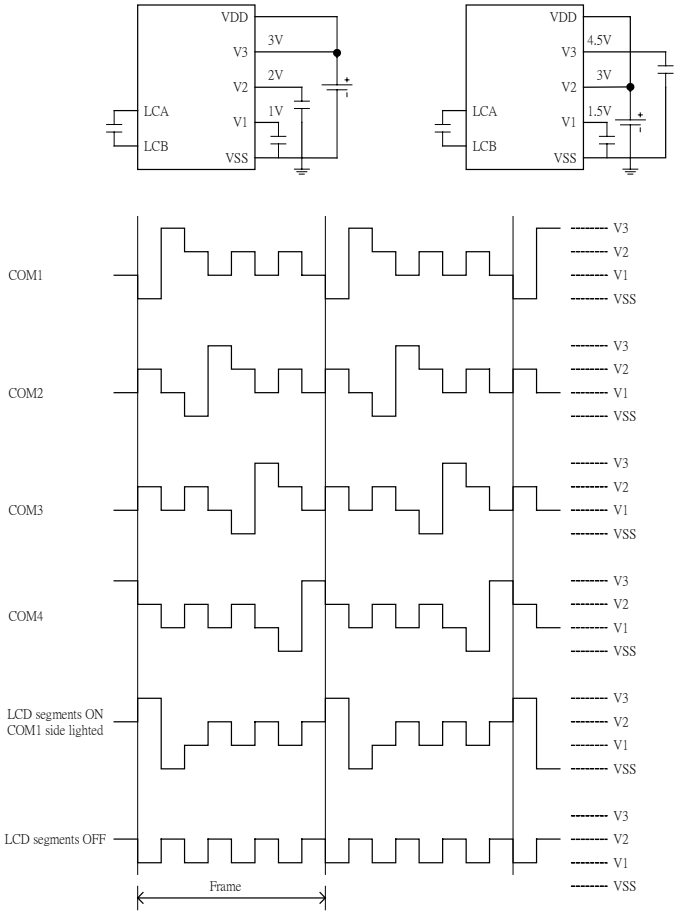
- LCD_DUTY [1:0] : select LCD segment Duty cycle.

LCD_DUTY [1:0]	General Output Port	LCD frame frequency	Driving method			
			bit3(7)	bit2(6)	bit1(5)	bit0(4)
00	static	LCDCK x 8	-	-	-	-
01	1/2	LCDCK x (4/2)	-	-	COM2	COM1
10	1/3	LCDCK x (4/3)	-	COM3	COM2	COM1
11	1/4	LCDCK x (4/4)	COM4	COM3	COM2	COM1

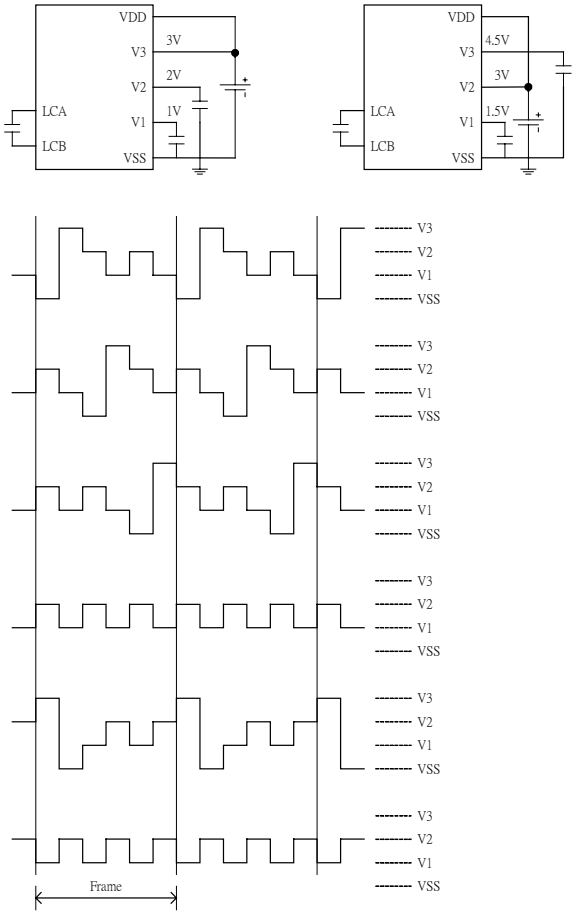
- LCD Driving Methods: There are six kinds of driving methods can be selected by LCD_DUTY [1:0] and LEVEL. The driving waveforms of LCD driver are as below:

VDD=3.0V

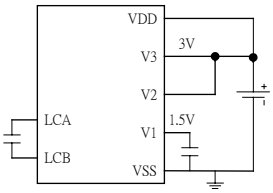
(1) 1/4 duty, 1/3 bias

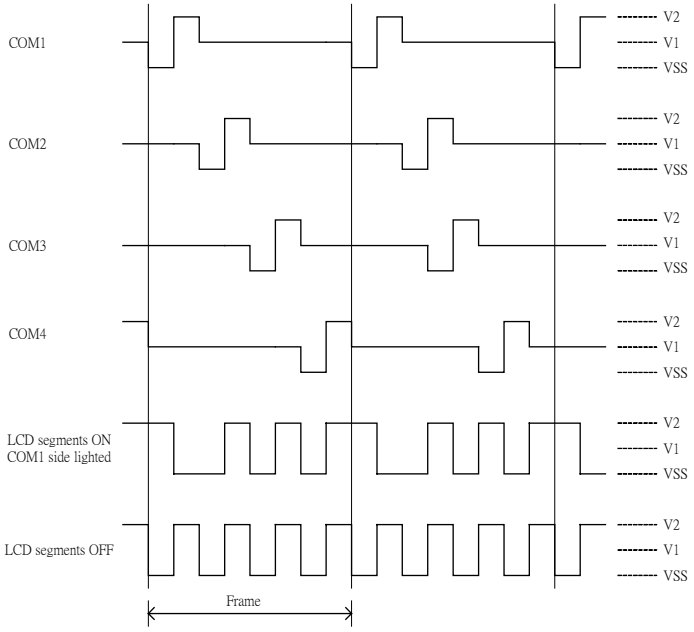


(2) 1/3 duty, 1/3 bias

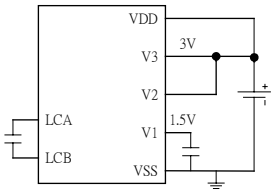


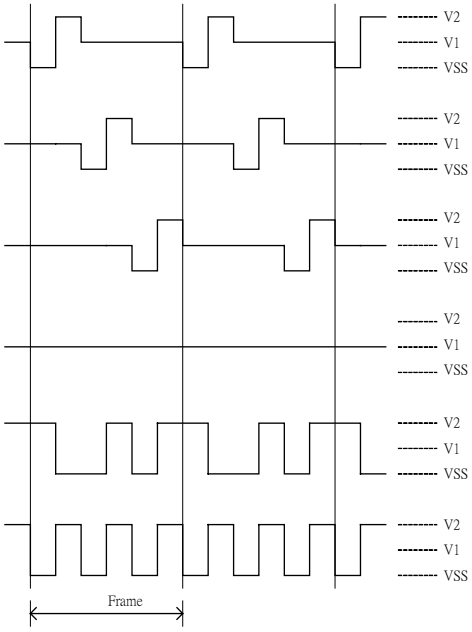
(3) 1/4 duty, 1/2 bias



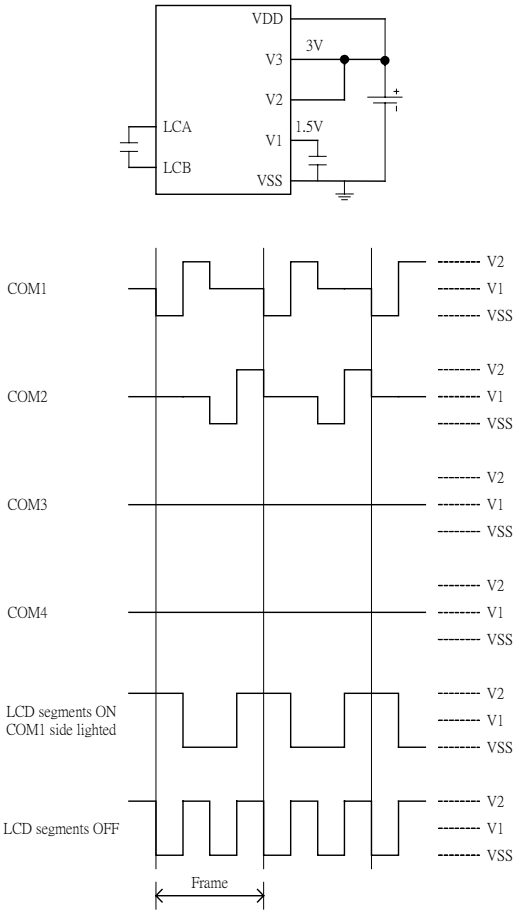


(4) 1/3 duty, 1/2 bias

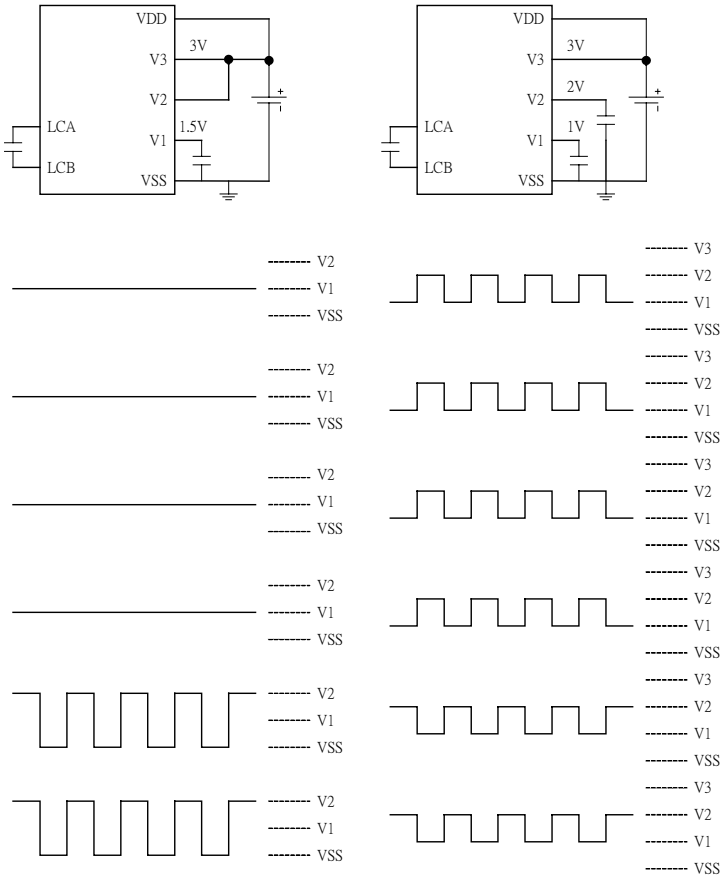




(5) 1/2 duty, 1/2 bias



(6) static



10.13 Halt and Sleep Modes

- Halt Mode

  After CPU executes an 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.

  HALT

  NOP

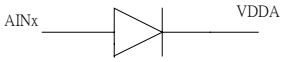
- Sleep Mode

  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.

  Sleep

  NOP

  To make sure that CPU consumes minimum power in Sleep mode, it is necessary to open all power blocks and analog circuits before issuing the Sleep command, and make sure all I/O Ports are in VDD or VSS voltage levels. There exist parasitic diodes between VDDA and analog input ports (see below Figure) When VDDA is turned off and VDDA is low, it is necessary to keep AIN0~AIN5 in Floating and AINENB [7:0] = 00h



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

```
CLRF    NETA
CLRF    NETB
CLRF    NETC
CLRF    NETD
CLRF    NETE
CLRF    NETF
CLRF    PT1PU
CLRF    PT1EN
CLRF    AINENB              ; Set PT1 as Analog Input Pin
MOVLW   01h
MOVWF   PT2PU
MOVLW   0FEh
MOVWF   PT2EN
CLRF    PT2                 ; Set PT2 [7:1] Output Low, PT2 [0] Input /Pull up
CLRF    INTF
MOVLW   081h                ; External Interrupt Enable
MOVWF   INTE
SLEEP
NOP
```

11. Instruction Set

The FS9XXX instruction set consists of 37 instructions. Each instruction is a 16-bit word with an OPCODE and one or more operands. The detail descriptions are below.

11.1 Instruction Set Summary

Table : FS9XXX Instruction Set

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

11.2 Instruction Description

(By alphabetically)

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

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

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

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

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

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

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)

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

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)

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

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 TO = 1 PD = 1

DECF	Decrement f
Syntax	DECF f, d $0 \le f \le 255$ $d \in [0,1]$
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

COMF	Complement f
Syntax	COMF f, d $0 \le f \le 255$ $d \in [0,1]$
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

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 ≤ k ≤ 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 ≤ f ≤ FFh d ∈ [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

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

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)

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

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

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

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

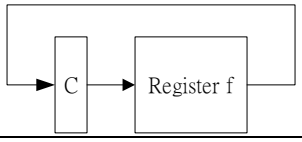
MOVWF	Move W to f
Syntax	MOVWF f 0 ≤ f ≤ FFh
Operation	[f] ← [W]
Flag Affected	None
Description	Move data from the W register to [f].
Cycle	1
Example: **MOVWF OPERAND**	Before instruction: W = 88h, OPERAND = 23h After instruction: W = 88h, OPERAND = 88h

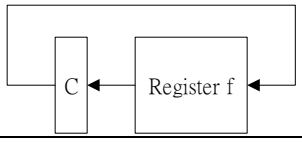
RETFIE	Return from Interrupt
Syntax	RETFIE
Operation	[Top Stack] => PC Pop Stack 1 => GIE
Flag Affected	None
Description	The program counter is loaded from the top stack, then pop stack. Setting the GIE bit enables interrupts.
Cycle	2

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

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

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

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.
Cycle	1

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

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

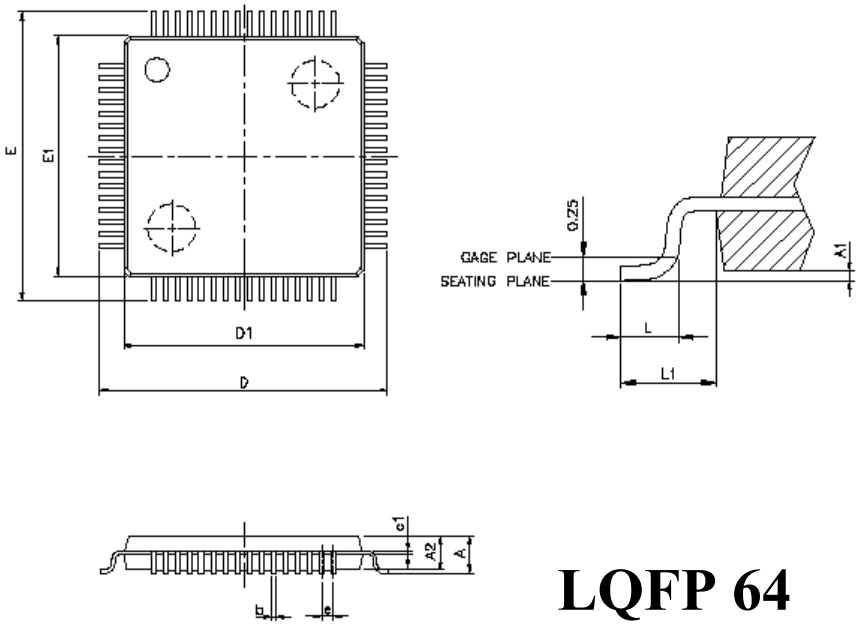
SUBWFC	Subtract W and Carry from f
Syntax	SUBWFC f, d 0 ≤ f ≤ FFh d ∈ [0,1]
Operation	[Destination] ← [f] – [W] –Ċ
Flag Affected	C, DC, Z
Description	Subtract the content of the W register from [f]. If d is 0, the result is stored in the W register. If d is 1, the result is stored back in [f].
Cycle	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

12. Package Information

12.1 Package Outline



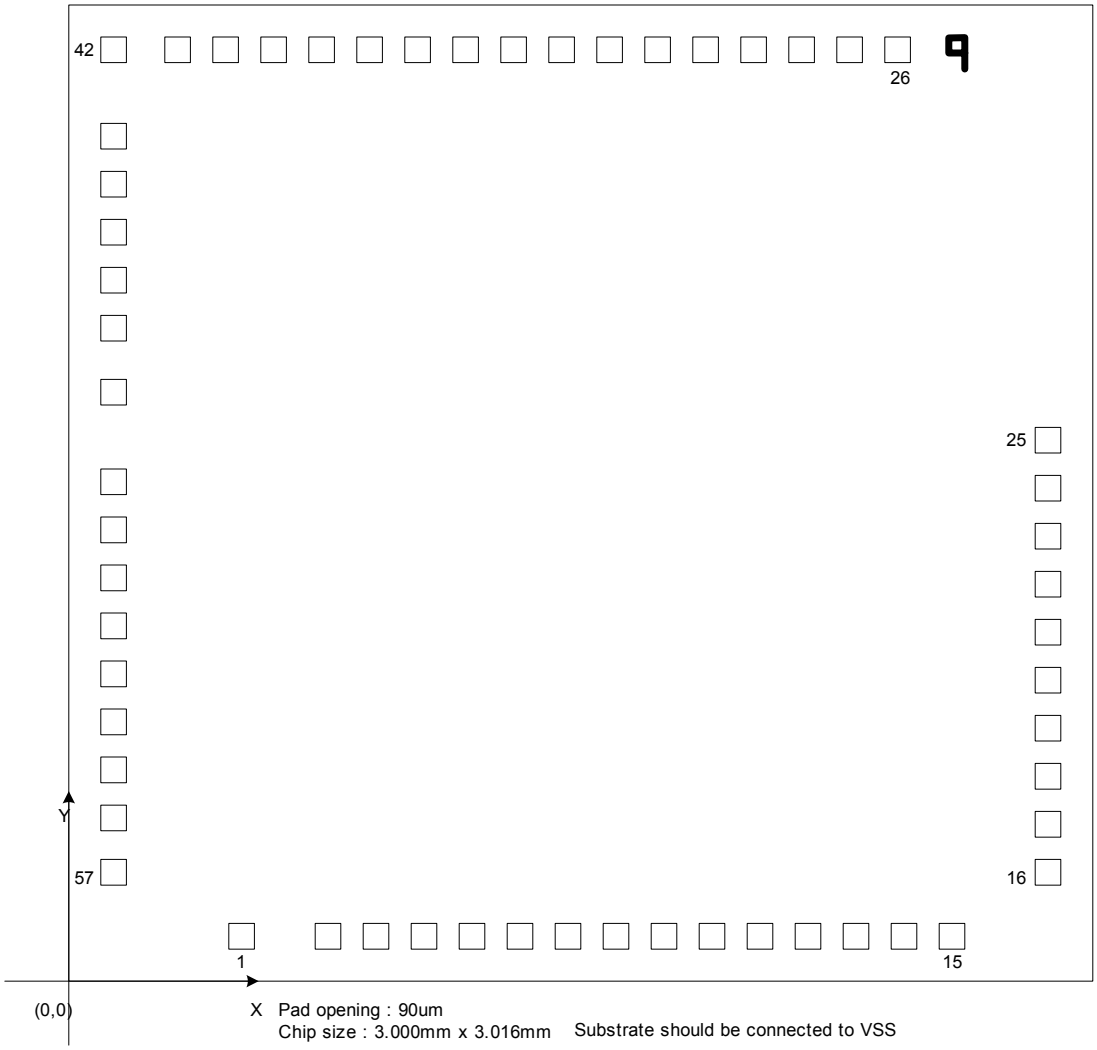
VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

SYMBOLS	MIN.	MAX.
A	--	1.60
A1	0.05	0.15
A2	1.35	1.45
b	0.17	0.27
c1	0.09	0.16
D	12.00 BSC	
D1	10.00 BSC	
E	12.00 BSC	
E1	10.00 BSC	
e	0.50 BSC	
L	0.45	0.75
L1	1.00 REF	

NOTES:
1. JEDEC OUTLINE:MS-026 BCD
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25mm PER SIDE. D1 AND E1 ARE MAXIMUM PLASTIC BODY SIZE DIMENSIONS 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.

LQFP 64

12.2 PAD Assignment



Pad opening : 90um
Chip size : 3.000mm x 3.016mm
Substrate should be connected to VSS

12.3 BONDING PAD LOCATION

BONDING PAD LOCATION							
※FSC P/N: FS9821							
※PAD NO: 57PADS				※Die Size:3000x 3016 um2			
Pad No.	Name	X[um]	Y[um]	Pad No.	Name	X[um]	Y[um]
1	TST	485.000	145.000	30	SEG<8>	1799.000	2870.000
2	RST	770.000	145.000	31	SEG<7>	1666.000	2870.000

3	VPP	906.000	145.000	32	SEG<6>	1534.000	2870.000
4	OP1O	1041.000	145.000	33	SEG<5>	1401.000	2870.000
5	REFO	1173.000	145.000	34	SEG<4>	1268.000	2870.000
6	FTB	1306.000	145.000	35	SEG<3>	1136.000	2870.000
7	FTC	1440.000	145.000	36	SEG<2>	1003.000	2870.000
8	VB	1565.000	145.000	37	SEG<1>	871.000	2870.000
9	AGND	1695.000	145.000	38	COM4	804.000	2870.000
10	PT1<0>	1828.000	145.000	39	COM3	616.000	2870.000
11	PT1<1>	1968.000	145.000	40	COM2	489.000	2870.000
12	PT1<2>	2107.000	145.000	41	COM1	362.000	2870.000
13	PT1<3>	2243.000	145.000	42	LCA	145.000	2870.000
14	PT1<4>	2377.000	145.000	43	LCB	145.000	2552.000
15	PT1<5>	2509.000	145.000	44	V1	145.000	2425.000
16	PT1<6>	2855.000	454.000	45	V2	145.000	2298.000
17	PT1<7>	2855.000	577.000	46	V3	145.000	2171.000
18	PT2<0>	2855.000	700.000	47	VDDA	145.000	2025.000
19	PT2<1>	2855.000	821.000	48	VS	145.000	1790.000
20	PT2<2>	2855.000	942.000	49	VGG	145.000	1452.000
21	PT2<3>	2855.000	1065.000	50	VSSP	145.000	1330.000
22	PT2<4>	2855.000	1187.000	51	CB	145.000	1208.000
23	PT2<5>	2855.000	1308.000	52	CA	145.000	1085.000
24	PT2<6>	2855.000	1432.000	53	VDDP	145.000	965.000
25	PT2<7>	2855.000	1556.000	54	VDD	145.000	842.000
26	SEG<12>	2329.000	2870.000	55	VSS	145.000	719.000
27	SEG<11>	2197.000	2870.000	56	XOUT	145.000	596.000
28	SEG<10>	2064.000	2870.000	57	XIN	145.000	428.000
29	SEG<9>	1931.000	2870.000				