Datasheet

FS98O25

8-bit MCU with 8k program EPROM, 256-byte RAM,
2 low noise OPAMP, 8-ch 14-bit ADC,
4 × 32 LCD driver and RTC

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

Figure List

Table List

Register List

1. Device Overview

The FS98O25 is a CMOS 8-bit single chip microcontroller(MCU) with embedded a 8kx16 bits one-time programmable (OTP) ROM, a 8-channel 14-bit fully differential input analog to digital converter, low noise amplifier, and 4 x 32 LCD driver.

The FS98O25 is best suited for applications such as electrical scale, meter, and sensor or transducer measurement application etc.

1.1. High Performance RISC CPU

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

1.2. Peripheral Features

- 20-bit bi-directional I/O port.
- Two PDM (Pulse Density Modulator) output.
- Buzzer output.
- I2C serial I/O port (slave mode only).
- 4 x 32 LCD drivers.
- One 8-channel 14-bit fully differential input analog to digital converter(ADC)
- Two low noise amplifier

1.3. Analog Features

- 8-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.

1.4. Special Microcontroller Features

- External 32768Hz crystal oscillator (RTC).
- Embedded Low Voltage Reset (LVR) and Low Voltage Detector (LVD).
- Embedded charge pump (Voltage Doubler) and voltage regulator (3.6V regulated output).
- Embedded bandgap voltage reference (typical 1.16V±50mV, 150ppm/°C).
- 8 Interrupt sources (external: 5, internal: 3).
- Internal silicon temperature sensor.
- Watchdog timer (WDT).
- Embedded 1.0 MHz oscillator.
- Package: 82-pin dice form, 100-pin LQFP.

1.5. CMOS Technology

- Voltage operation ranges from 2.2V to 3.6V.
- Operation current is less than 4 mA; sleep mode current is about 3μA.

1.6. Applications

- Sensor or transducer measurement applications.

- Electronic kitchen scale, personal scale.
- Digital meter.

1.7. Ordering Information

Table 1-1 Ordering Information

Product Number	Description	Package Type
FS98O25	MCU with OTP ROM; The customer has to program the compiled hex code into OTP ROM.	82-pin Dice form, 100-pin QFP
FS98O25-nnnV	MCU with program type; FSC programs the customer's compiled hex code into EPROM at factory before shipping.	82-pin Dice form, 100-pin QFP
FS98O251	6K ROM version of FS98O25	82-pin Dice form, 100-pin QFP
FS98O251-nnnV	6K ROM version of FS98O25 with program type	82-pin Dice form, 100-pin QFP

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

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

1.8. Pin Configuration

Figure 1-1 FS98O25 pin configuration

1.9. Pin Description

Table 1-2 FS98O25 pin description

Name	In/Out	Pin No	Description
VPP	I	1	Programming Power Supply
OP2O	I/O	4	OPAMP 2 Output
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<0~7>/AIN0~7	I/O	11~18	Digital I/O Port or Analog input channel
PT2<0~1>/INT0~1, PT3<0~1>/INT2~3	I/O	19~20 34~35	Digital I/O Port and External Interrupt input
PT2<2,5>/PDM1,2	I/O	21,24	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
PT2<7>/BZ	I/O	26	Digital I/O Port or Buzzer Output
PT3<2>/PFI	I/O	36	Digital I/O Port or Programmable Frequency Input
PT3<3>/PFO	I/O	37	Digital I/O Port or Programmable Frequency Output
PT2<6>	I/O	25	Digital I/O Port
SEG32~SEG1	O	54~74 38~48	LCD Segment Driver Output
COM4~COM1	O	75~78	LCD Common Driver Output
LCA	I/O	82	LCD Charge Pump Capacitor Positive Connection
LCB	I/O	83	LCD Charge Pump Capacitor Negative Connection
V3,V2,V1	I/O	84~86	LCD Bias
VDDA	I/O	87	Analog Power Output
VS	I/O	88	Voltage Source from VDDA
VGG	I/O	89	Charge Pump Voltage
NC	-	90	No Connection
VSSP	I	91	Charge Pump Negative Power Supply
CB	I/O	92	Charge Pump Capacitor Negative Connection
CA	I/O	93	Charge Pump Capacitor Positive Connection
VDDP	I	94	Charge Pump Positive Power Supply
VDD	I	95	Positive Power Supply
VSS	I	96	Negative Power Supply (Ground)
XOUT	O	97	32768Hz Oscillator Output
XIN	I	98	32768Hz Oscillator Input
TST	I	99	Testing Mode
RST	I	100	CPU Reset

1.10. Functional Block Diagram

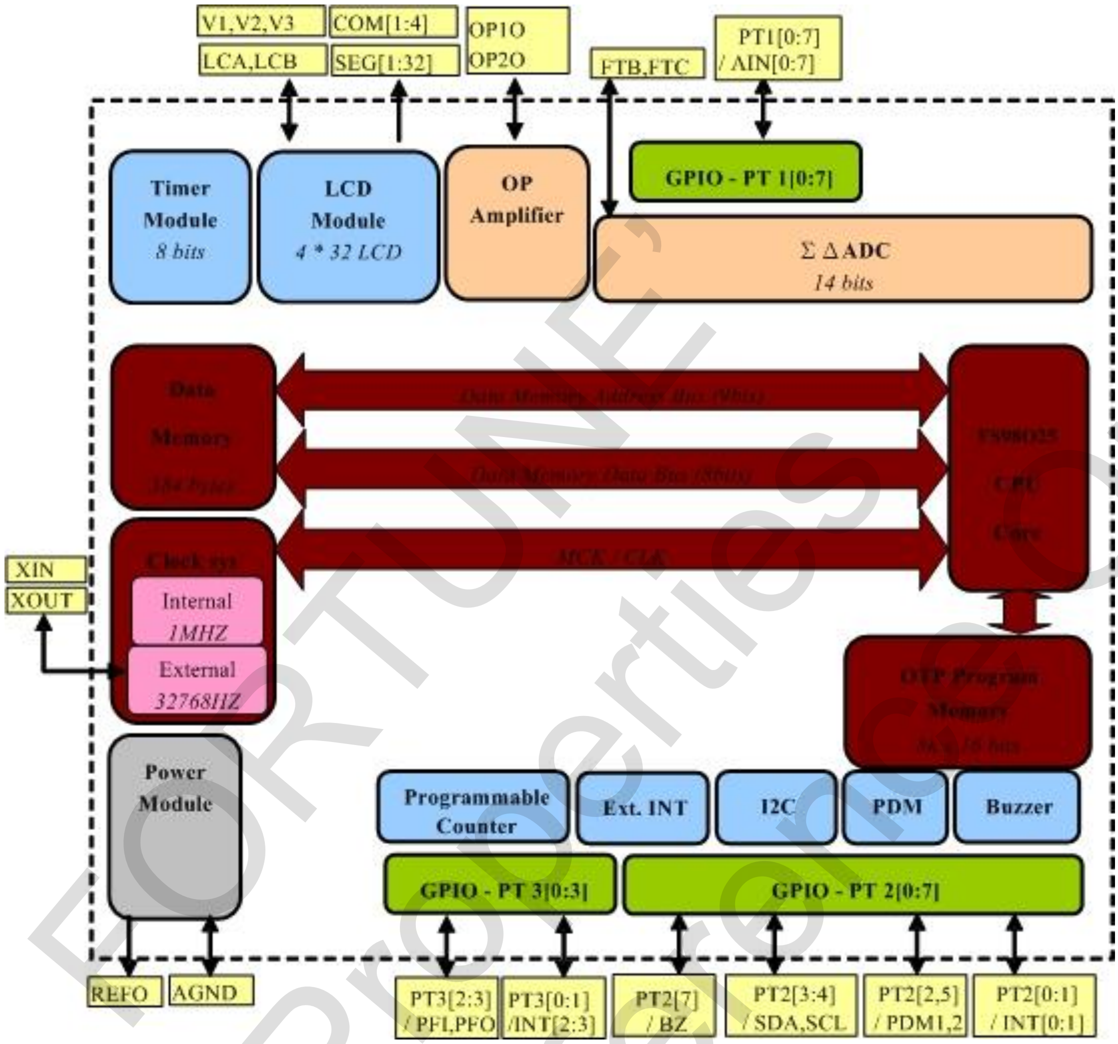


Figure 1-2 FS98O25 function block

There are 5 kinds of functional blocks in the Function Block Diagram, described as table 1-3:

Table 1-3 FS98O25 main function description table

Item	Sub Item	Description
CPU Kernel	FS98O25 CPU Core	Please refer to Chapter 1.11 for detailed description
	OTP Program Memory	OTP: One Time Programmable 16k bytes is used for 8k line programming instructions
	Data Memory	FS98O25 has 384 bytes Data Memory embedded in it. (128 bytes registers, 256 bytes general data memory)
	Clock sys	There are two clock sources in FS98O25. One is the internal clock which generates 1M HZ for CPU works, and the other is an external one which provide 32768 HZ clock signal to the chip.
Digital Function	Timer Module	Clock Counter for Time out interrupt and Watch dog Timer
	LCD Module	Embedded 4 X 32 LCD driver
	I2C	Embedded Serial Port for Communication, It support I2C protocol which is designed by Philips
	PDM	Similar to PWM function
	Buzzer	User should connect a Buzzer to the embedded buzzer port to receive the warning or reminding signal.
	Programmable Counter	FS98O25 embeds Dual 16-bit Programmable Counter which could be used to do three kinds of processes: Counter, Pulse Width Measurement and Frequency Measurement.
	Ext. INT	FS98O25 support 2 External Interrupt port
Analog Function	ADC	An embedded Sigma-Delta Analog to Digital Converter which converts the analog signal of the sensor to a digital number.
	OP Amplifier	FS98O25 has an embedded low noise OP amplifier for pre-processing the signal, which is connected to the ADC to get a better A/D resolution or amplify the signal to fit the ADC Input range.
Power Function	Power Module	FS98O25 has a special power system. The power system can supply a fixed voltage for CPU and ADC. The input voltage of the chip can be within a certain range and floating.
General Purpose I/O	PT1	The PT1 port has 8 bits. User can define these 8 bits for general purpose or special assignment as ADC input.
	PT2	The PT2 port has 8 bits. User can define these 8 bits for general purpose or some special function as External Interrupt, I2C, PDM and the Buzzer.

1.11. CPU Core

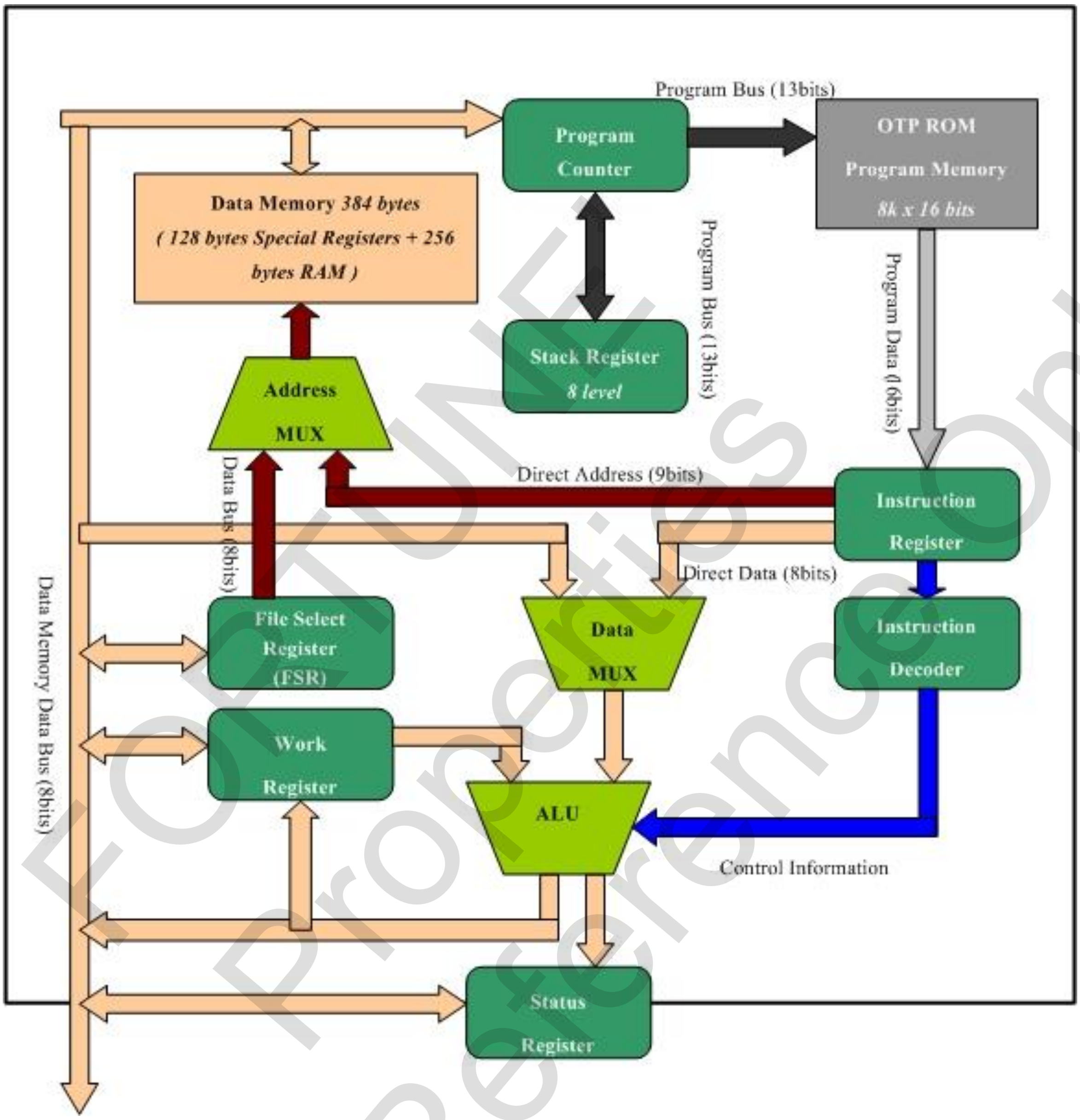


Figure 1-3 FS98O25 CPU core function block

The "CPU Core Block Diagram" shown in Section 1.11 mainly includes 7 important registers and 2 memory units. Please see the Figure 1-3 and the Table 1-4 for detailed information.

Table 1-4 FS98O25 CPU core block diagram description table

Items	Sub Items	Description
Registers	***Program Counter***	This Register plays an important role in all the CPU working cycle. It records the pointer of the instruction that the CPU processes every cycle in the ***Program Memory***. In a general CPU cycle, ***Program Counter*** pushes the *Program Memory Address* (13bits), instruction pointer, into the ***Program Memory*** and then increments for the next cycle.
	Stack Register	***Stack Register*** is used for recording the *program return instruction pointer*. When the program calls function, ***Program Counter*** will push the instruction pointer into the ***Stack Register***. After finish this function, ***Stack Register*** pushes the instruction pointer back to the ***Program Counter*** to resume the original program process.
	Instruction Register	After ***Program Counter*** pushes the instruction pointer (***Program Memory Address***) into the ***Program Memory***, ***Program Memory*** pushes the ***Program Memory Data*** (16bits), *instruction*, into ***Instruction Register*** for reference. FS98O25 instruction has 16 bits, and contains 3 kinds of information as ***Direct Address***, ***Direct Data*** and ***Control Information***. CPU could push the ***Direct Data*** into ***Work Register*** or do some process for the register stored in the ***Data Memory*** pointed by the ***Direct Address*** by ***Control Information***. ● **Direct Address** (8bits) It is the ***Data Memory Address***. CPU can use this address to process the ***Data Memory***. ● **Direct Data** (8bits) It is the value which CPU used for processing ***Work Register*** by the ***ALU*** (arithmetic and logic unit). ● **Control Information** It records the information for the ***ALU*** to process.
	Instruction Decoder	***Instruction Register*** pushes the ***Control Information*** to the ***Instruction Decoder*** to decode and then sends the decoded information to related registers.
	File Select Register	In FS98O25 Instruction Sets, ***FSR (File Select Register)*** is used for indirect data process. User could fill the ***FSR*** with the ***Data Memory Address*** of some register, and then process this register by ***IND Register***. CPU will fill the ***IND Register*** with the data address in the ***Data Memory*** as ***FSR***.
	Work Register	***Work Register*** is used for buffering the data which is stored in some memory address of ***Data Memory***.
	Status Register	While CPU processes some register data by ***ALU***, the following status may change as follows: ***PD***, ***TO***, ***DC***, ***C*** and ***Z***. Please refer to ***Section 3.3.2*** for detailed introduction.
Memory	***Program Memory***	FS98O25 has an embedded 16k bytes *OTP* (*One Time Programmable*) ROM as ***Program Memory***. Because the ***OPCODE*** of the instruction is 16 bits, user could program 8k instructions in FS98O25 at most. ***Program Memory*** Address Bus is 13 bits, and the Data Bus is 16bits.
	Data Memory	FS98O25 has an embedded 384 bytes Data Memory. The Data Memory Address Bus is 9 bits, and Data Bus is 8 bits.

1.12. Clocking Scheme/Instruction Cycle

One Instruction cycle (CPU cycle) includes 4 steps and the CPU could process 2 steps per CPU Clock. Users can setup the MCK Register to decide the step timing. Please refer to Chapter 5 for related information. For Example, if the MCK Register is filled with 0x04H (MCK = ICK, Instruction Cycle = MCK / 2, ICK = 1MHZ), the step timing is 500k HZ, and one instruction cycle needs 4us (2 x 1/500k sec) to complete. The 4 steps are described as follows. Please refer to the CPU core (Section 1.11) to understand these 4 steps.

1. **Fetch**
   Program Counter pushes the Instruction Pointer into Program Memory, and the pointed Data in the Program Memory is stored in the Instruction Register.
2. **Decode**
   The Instruction Register pushes the Direct Address to Address MUX, or pushes the Direct Data to Data MUX, and pushes the Control Information into Instruction Decoder to decode the OPCODE.
3. **Execute**
   ALU executes the process based on the decoded Control Information.
4. **Write Back**
   Push the ALU result to Work Register or Assigned Data Memory Address.

Because one OPCODE can only have either Direct Address or Direct Data, sometimes user needs 2 instructions to complete one simple job. For example, if user want to fill Data Memory address 0x55h with data 0xFF, user needs to process 【movlw 0xFFH】 to filled Work Register with 0xFFH, and then process 【movwf 0x55H】to fill Data Memory 0x55H with Work Register content. For the same reason, CPU needs 2 instruction cycles to complete some kinds of instructions such as call, goto…etc. Please see the Figure 1-4.

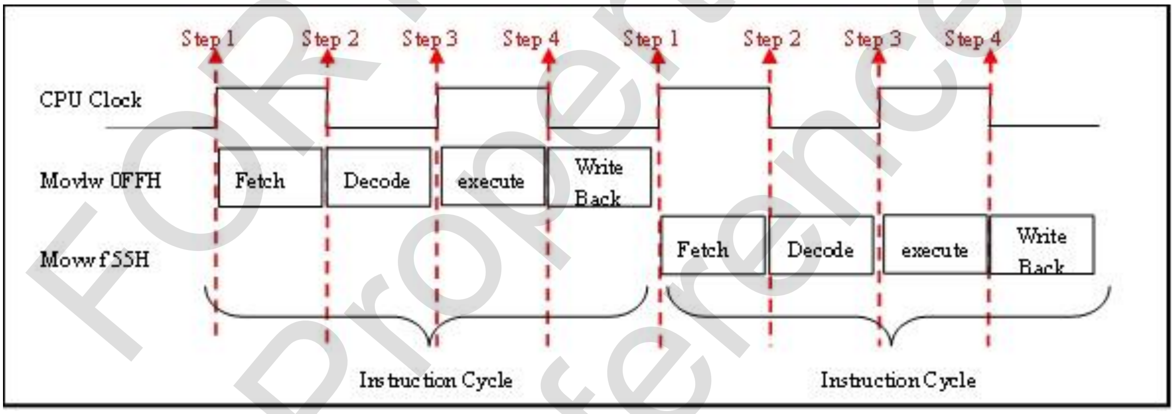


Figure 1-4 FS98O25 instruction cycle

2. Electrical Characteristics

2.1. Absolute Maximum Ratings

Table 2-1 FS98O25 absolute maximum rating table

Parameter	Rating	Unit
Supply Voltage to Ground Potential for any port	-0.3 to 5.5	V
Applied Input/Output Voltage	-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	

* FS98O25 passed -40°C LTOL (Low Temperature Operating Life) test (VDD=3V)

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

Table 2-2 FS98O25 DC characteristics

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
VDD	Recommended 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		mA
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		mA
IOH	High Level Output Current	VOH=VDD-0.3 V		7		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)		6		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 Detection Voltage	S_LB [1:0]=00		2.3		V
		S_LB [1:0]=01		3.5		
VSR	VS Switch Resistor			10		Ω
FRC	Internal RC oscillator		0.7	1.0	1.3	MHz
FWDT	Internal WDT Clock			2.1		kHz

2.3. ADC Characteristics (VDD=3V, T_A=25℃, unless otherwise noted)

Table 2-3 FS98O25 ADC characteristics

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	±31250[1]	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

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

Table 2-4 FS98O25 OPAMP characteristics

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
	Input Offset			1.5		mV
	Input Offset Voltage with Chopper	Rs<100Ω		20		μ V
	Input Reference Noise	Rs=100Ω , 0.1Hz~1Hz		1.0		μ Vpp
	Input Reference Noise with Chopper	Rs=100Ω , 0.1Hz~1Hz		0.5		μ 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

2.5. Temperature Characteristics(VDD=3V)

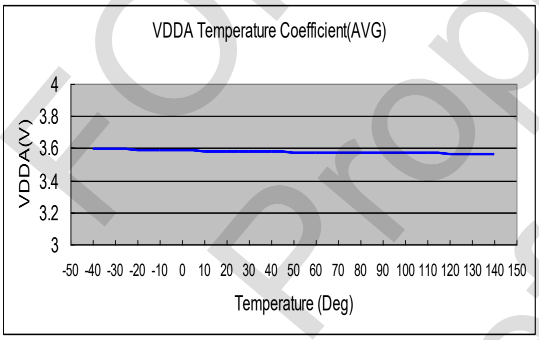


Figure 2-1 VDDA vs Temp @ VDD=3V

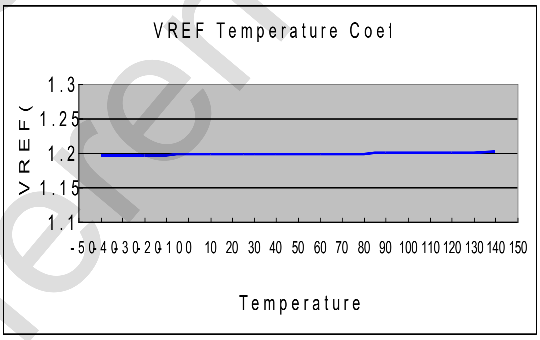


Figure 2-2 VREF vs Temp @ VDD=3V

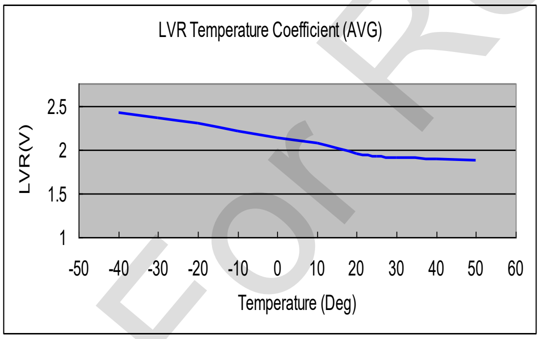


Figure 2-3 LVR vs Temp @ VDD=3V

[1] Use ADOH, ADOL and ADOLL (Extra ADC output register) three register (24 bits ADC output)

3. Memory Organization

3.1. Program Memory Structure

FS98O25 has an 13bits Program Counter which is capable of addressing a 8k x 16bits program memory space and a 8 level depth 13bits Stack Register. The Start up/Reset Vector is at 0x0000H. When FS98O25 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. Please see Figure 3-1.

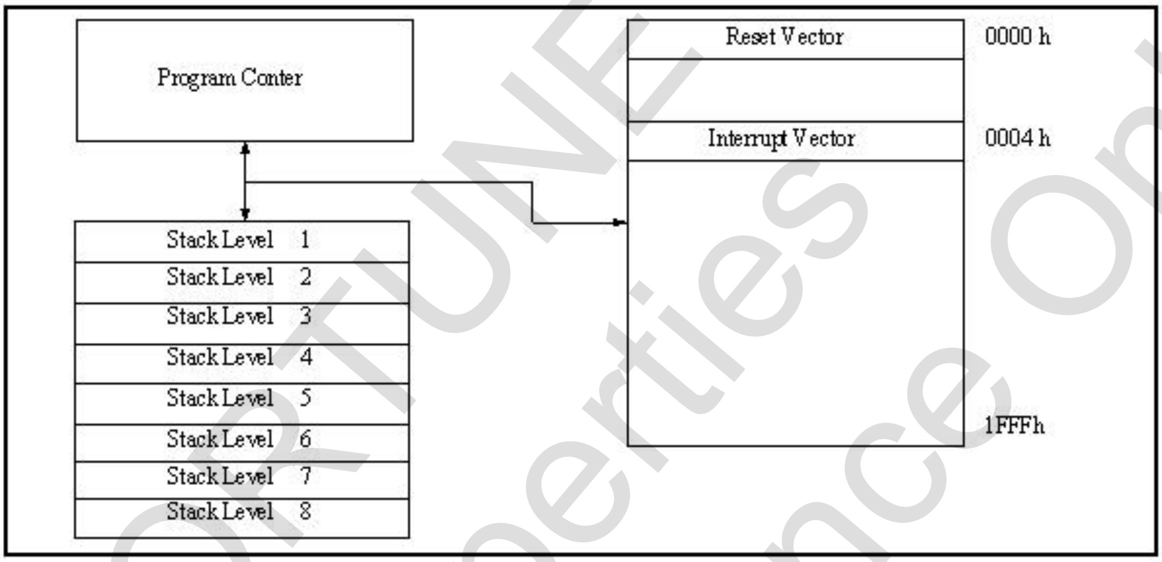


Figure 3-1 FS98O25 program memory structure

3.2. Data Memory Structure

FS98O25 has a 384-byte 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. Please see Table 3-1.

Table 3-1 FS98O25 Data memory structure

Start Address	End Address	Data Memory
0X00H	**0X07H**	***System Special Registers***
0X08H	**0X7FH**	***Peripheral Special Registers***
0X80H	**0X17FH**	***General Data Memory(256 bytes)***

3.3. 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. Please see Section 1.11 for related CPU work flow chart.

Table 3-2 system register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset[2]
00H	IND0	3.427	Use contents of FSR0 to address data memory								uuuuuuuu
01H	IND1	3.4.1	Use contents of FSR1 to address data memory								uuuuuuuu
02H	FSR0	1.11/3.4.1	Indirect data memory address pointer 0								uuuuuuuu
03H	FSR1	1.11/3.4.1	Indirect data memory address pointer 1								uuuuuuuu
04H	STATUS	1.11/3.4.2	IRP1	IRP0		PD	TO	DC	C	Z	00u00uuu
05H	WORK	1.11	WORK register								uuuuuuuu
06H	INTF	3/6/7/9/10/11				TMIF	I2CIF	ADIF	E1IF	E0IF	00000000
07H	INTE	3/6/7/9/10/11	GIE			TMIE	I2CIE	ADIE	E1IE	E0IE	00000000
16H	INTF2	6/7						CTIF	E3IF	E2IF	00000000
17H	INTE2	6/7						CTIE	E3IE	E2IE	00000000

[2] u mean unknown or unchanged

3.3.1. Special Register Contents after External Reset (Power On Reset) and WDT Reset

Table 3-3 special register reset table

Register Address	Register Name	Register Content	
		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
28H	PT3	00000000	uuuuuuuu
29H	PT3EN	00000000	uuuuuuuu
2AH	PT3PU	00000000	uuuuuuuu
2BH	PT3MR	00000000	uuuuuuuu
37H	PT2OC	uuu11uuu	uuuuuuuu
57H	I2CCON	0001uuuu	uuuuuuuu
58H	STA	uu0000u0	uuuuuuuu
59H	I2CADD	00000000	uuuuuuuu
5AH	I2CBUF	00000000	uuuuuuuu

3.3.2. IND and FSR Registers

The IND (Indirect Addressing) register is not a physical register, but indirect addressing needs the IND register. Any instruction using the IND register actually accesses the register pointed by the FSR (File Select Register). While user reads data from the IND register, the CPU gets the data from the Data Memory at the address stored in FSR. While user writes the data into IND register, CPU actually saves the data into Data Memory at the address stored in FSR. Please see Figure 3-2.

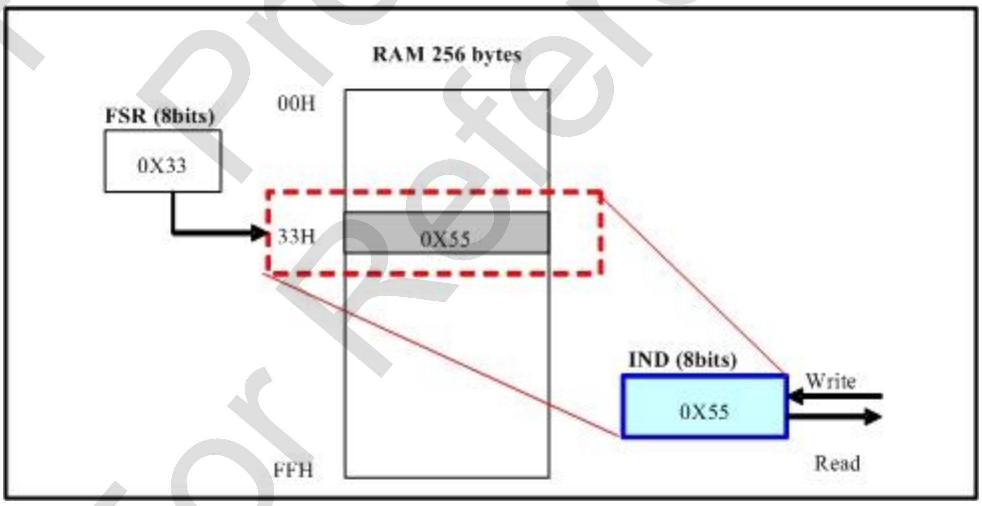


Figure 3-2 IND & FSR function description

3.3.3. STATUS Register

The STATUS register contains the arithmetic status of ALU and the RESET status. The STATUS register is similar to other registers, and can be the destination for any instruction. If the STATUS register is the destination for an instruction that affects the Z, DC or C bit, then the writing to these three bits is disabled. These bits are set or cleared according to the device logic. The TO and PD bits are not writable.

Register STATUS at address 04H

property	R/W-0	R/W-0	U-X	R-0	R-0	R/W-X	R/W-X	R/W-X
STATUS	**IRP1**	**IRP0**		**PD**	**TO**	**DC**	**C**	**Z**
	Bit7							Bit0

Bit 7 **IRP1**: Indirect address 1 page select

1 = Indirect address 1 extend memory address is set (Memory 1XXH)

0 = Indirect address 1 extend memory address is Not set (Memory 0XXH)

Bit 6 **IRP0**: Indirect address 0 page select

1 = Indirect address 0 extend memory address is set (Memory 1XXH)

0 = Indirect address 0 extend memory address is Not set (Memory 0XXH)

Bit 4 **PD**: Power down Flag.

1 = By execution of SLEEP instruction

0 = After power-on reset

Bit 3 **TO**: Watch Dog Time Out Flag. Cleared by writing 0 and Set by Watch Dog Time Out

1 = A Watch Dog Timer time-out occurred

0 = After power-on reset

Bit 2 **DC**: Digit Carry Flag/borrow Flag, for ADDWFI and SUBWFI

(for borrow the polarity is reversed)

1 = If there is a carry out from the 4^{th} bit of the result

0 = No carry out from the 4^{th} bit of the result

Bit 1 **C**: Carry Flag/borrow Flag (~Borrow)

(for borrow the polarity is reversed)

1 = If there is a carry out from the Most Significant bit of the result

0 = No carry out from the most significant bit of the result

Bit 0 **Z**: Zero Flag

1 = The result of an arithmetic or logic operation is zero

0 = The result of an arithmetic or logic operation is NOT zero

3.3.4. INTE and INTF registers

The INTE and INTF registers are readable and writable registers, and contain enable and flag bits for interrupt devices.

Register INTE at address 07H

property	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTE	GIE			TMIE	I2CIE	ADIE	E1IE	E0IE
	Bit7							Bit0

Bit 7 GIE: Global Interrupt Enable flag
1 = Enable all unmasked interrupts
0 = Disable all interrupts

Bit 4 TMIE: 8-bit Timer Interrupt Enable flag
1 = Enable Timer interrupt
0 = Disable Timer interrupt

Bit 3 I2CIE: I2C Interface Interrupt Enable flag
1 = Enable I2C interface interrupt
0 = Disable I2C interface interrupt

Bit 2 ADIE: Analog to Digital converter Interrupt Enable flag
1 = Enable analog to digital converter interrupt
0 = Disable analog to digital converter interrupt

Bit 1 E1IE: PT2.1 External Interrupt Enable flag
1 = Enable PT2.1 external interrupt
0 = Disable PT2.1 external interrupt

Bit 0 E0IE: PT2.0 External Interrupt Enable flag
1 = Enable PT2.0 external interrupt
0 = Disable PT2.0 external interrupt

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

Register INTF at address 06H

property	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
INTF				TMIF	I2CIF	ADIF	E1IF	E0IF
	Bit7							Bit0

Bit 4 TMIF: 8-bit Timer Interrupt Flag
1 = Timer interrupt occurred (must be cleared in software)
0 = No Timer interrupt

Bit 3 I2CIF: I2C Interface Interrupt Flag
1 = I2C Interface interrupt occurred (must be cleared in software)
0 = No I2C Interface interrupt

Bit 2 ADIF: Analog to digital converter Interrupt Flag
1 = Analog to digital converter Interrupt occurred (must be cleared in software)
0 = No Analog to digital converter Interrupt

Bit 1 E1IF: PT2.1 External Interrupt Flag
1 = PT2.1 External Interrupt occurred (must be cleared in software)
0 = No PT2.1 External Interrupt

Bit 0 E0IF: PT2.0 External Interrupt Flag
1 = PT2.0 External Interrupt occurred (must be cleared in software)
0 = No PT2.0 External Interrupt

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

3.4. Peripheral Special Registers

The Peripheral Special Registers are designed for Peripheral functions, such as I/O ports, timer, ADC, signal conditional network control register, LCD driver. Please see Table 3-4 and the following Chapters for detailed description of these peripheral functions.

Table 3-4 peripheral special registers table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
08H	CTAH	6.3	CTA[15:8]								uuuuuuuu
09H	CTAL	6.3	CTA[7:0]								uuuuuuuu
0AH	CTBH	6.3	CTB[15:8]								uuuuuuuu
0BH	CTBL	6.3	CTB[7:0]								uuuuuuuu
0CH	CTCON	6.3	TON	MUXSEL[2:0]			TE	FQTMB	OVAB		0000000u
0DH	WDTCON	6.2	WTDTEN					WTS [2:0]			0uuuu000
0EH	TMOUT	6.1	TMOUT [7:0]								00000000
0FH	TMCON	6.1	TRST				TMEN	INS [2:0]			1uuu0000
10H	ADOH	10/11	ADO [15:8]								00000000
11H	ADOL	10/11	ADO [7:0]								00000000
12H	ADOLL		Extra ADC output register								00000000
13H	ADCON	10/11					ADRST	ADM [2:0]			uuuu0000
14H	MCK	5	M7_CK	M6_CK	M5_CK		M3_CK	M2_CK	M1_CK	M0_CK	00000000
15H	PCK	4/5/7.5/10		ENPUMP	S_CH2CK [1:0]		S_CH1CK [1:0]		S_BEEP	S_PCK	00000000
18H	NETA	10/11	SINL[1:0]		SINH[2:0]			SFTA[2:0]			00000000
19H	NETB	10/11	SOP2N[1:0]		SOP1N[1:0]		SVRL[1:0]		SVRH[1:0]		00000000
1AH	NETC	10/11	SREFO				ADG[1:0]		ADEN	AZ	00000000
1BH	NETD	10/11	OP2EN	SOP2P[2:0]			OP1EN	SOP1P[2:0]			00000000
1CH	NETE	4/10/11				ENVS	SILB[1:0]		ENLB		00000000
1DH	NETF	4/10/11		ENBAND	ENVDDA				ENAGND	ENVB	00000000
1FH	SVD	4.5								LBOUT	uuuuuuuu
20H	PT1	7	PT1 [7:0]								uuuuuuuu
21H	PT1EN	7	PT1EN [7:0]								00000000
22H	PT1PU	7	PT1PU [7:0]								00000000
23H	AIENB1	7	AIENB[7:6]		AIENB[5:0]						00000000
24H	PT2	7	PT2 [7:0]								uuuuuuuu
25H	PT2EN	7	PT2EN [7:0]								00000000
26H	PT2PU	7	PT2PU [7:0]								00000000
27H	PT2MR	7.2/7.5/8	BZEN	PM2EN		PM1EN	E1M[1:0]		E0M[1:0]		00000000
28H	PT3	7					PT3 [3:0]				uuuuuuuu
29H	PT3EN	7					PT3EN [3:0]				00000000
2AH	PT3PU	7					PT3PU [3:0]				00000000
2BH	PT3MR	7				PFOEN	E3M[1:0]		E2M[1:0]		00000000
30H	PMD1H	8	PMD1[15:8]								00000000
31H	PMD1L	8	PMD1[7:0]								00000000
32H	PMD2H		PDMD2[15:8]								00000000
33H	PMD2L		PDMD2[7:0]								00000000
36H	PMCON	8				PDMEN		PMCS[2:0]			00000000
37H	PT2OCB	9				PT2OC[4:3]					uuu11uuu
40H	LCD1	13	SEG2 [3:0]]				SEG1 [3:0]				uuuuuuuu
41H	LCD2	13	SEG4 [3:0]				SEG3 [3:0]				uuuuuuuu
42H	LCD3	13	SEG6 [3:0]				SEG5 [3:0]				uuuuuuuu
43H	LCD4	13	SEG8 [3:0]				SEG7 [3:0]				uuuuuuuu
44H	LCD5	13	SEG10 [3:0]				SEG9 [3:0]				uuuuuuuu
45H	LCD6	13	SEG12 [3:0]				SEG11 [3:0]				uuuuuuuu
46H	LCD7	13	SEG14 [3:0]				SEG13 [3:0]				uuuuuuuu
47H	LCD8	13	SEG16 [3:0]				SEG15 [3:0]				uuuuuuuu
48H	LCD9	13	SEG18 [3:0]				SEG17 [3:0]				uuuuuuuu
49H	LCD10	13	SEG20 [3:0]				SEG19 [3:0]				uuuuuuuu
4AH	LCD1	13	SEG22 [3:0]]				SEG21 [3:0]				uuuuuuuu

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
4BH	LCD2	13	SEG24 [3:0]				SEG23 [3:0]				uuuuuuuu
4CH	LCD3	13	SEG26 [3:0]				SEG25 [3:0]				uuuuuuuu
4DH	LCD4	13	SEG28 [3:0]				SEG27 [3:0]				uuuuuuuu
4EH	LCD5	13	SEG30 [3:0]				SEG29 [3:0]				uuuuuuuu
4FH	LCD6	13	SEG32 [3:0]				SEG31 [3:0]				uuuuuuuu
54H	LCDENR	13	LCDCKS [1:0]		LCDEN		LEVEL	LCD_DUTY[1:0]		ENPMPL	00000000
57H	I2CCON	9	WCOL	I2COV	I2CEN	CKP					0001uuuu
58H	I2CSTA	9			DA	P	S	RW		BF	uu0000u0
59H	I2CADD	9	I2CADD [7:0]								00000000
5AH	I2CBUF	9	I2CBUF [7:0]								00000000

4. Power System

FS98O25 has a special power system that can supply a fixed voltage (3.6V) for CPU and ADC. FS98O25 could work when the supply voltage is within a specified range, fixed or floating. The power system has 6 function engines as ***Voltage Doubler***, ***Voltage Regulator***, ***Analog Bias Circuit***, ***Common Voltage Generator Low Battery Comparator and Band gap Voltage / Temperature Sensor***. Through the first 4 function engines, the system can generate 3 Voltage level as VGG = 2VDDP, VDDA = 3.6V, AGND = 1.8V. Please see Figure 4-1.

1. **Voltage Doubler**
   The acceptable VDD range for FS98O25 is from 2.2V to 3.6V. Voltage Doubler raises the voltage of VGG to 2 times of VDDP[3]. VGG is used as the input of Voltage Regulator. It is from 4.4V to 7.2V. Please see Section 4.1 for detailed register setting.

2. **Voltage Regulator**
   The fixed voltage is important when the Analog function is working. Voltage Regulator raises the voltage of VDDA to fixed 3.6V. Although the input voltage of Voltage Regulator, VGG, is from 4.4V to 7.2V (It depends on the voltage of VDD), the minimum possible voltage is still higher than 3.6V, so Voltage Regulator could surely supply VDDA as 3.6V. Please refer to Section 4.2 for detailed register setting.

3. **Analog Bias Circuit**
   Analog Bias Circuit is used to set VB to 3.6V. VB is used for FS98O25 Analog Function Network. The user needs to enable Analog Bias Circuit, and then the Analog Functions such as ADC or OPAMP can work correctly. Please refer to Section 4.3 for detailed register setting.

4. **Common Voltage Generator**
   FS98O25 sets the analog ground to half VDDA. Please refer to Section 4.4 for detailed register setting.

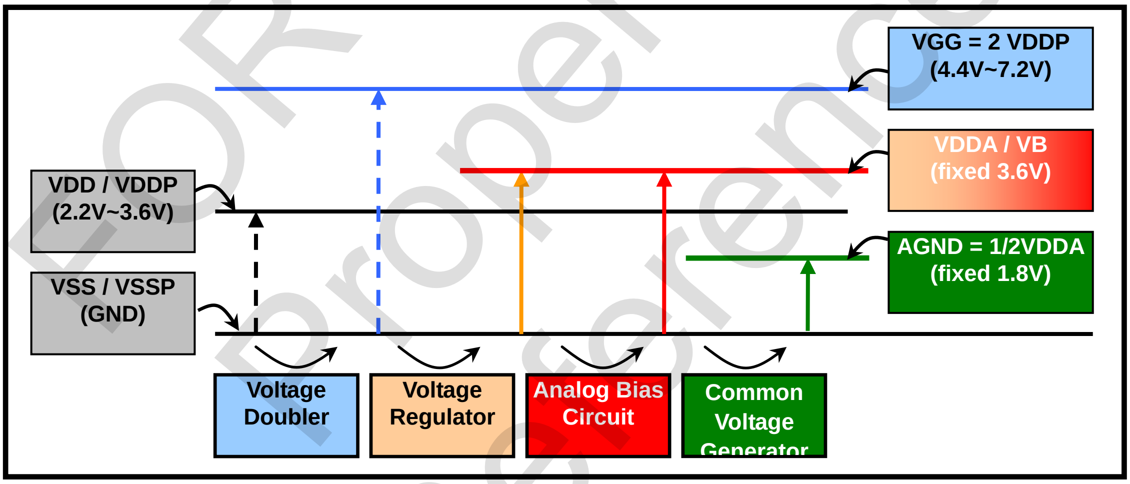


Figure 4-1 FS98O25 power system block

[3] VDDP means the VDD for Charge Pump (Voltage Doubler). User usually connects the VDDP to VDD. VSSP means the VSS for Charge Pump (Voltage Doubler). User usually connects the VSSP to VSS.

Table 4-1 FS98O25 power system register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
15H	PCK	4/5/7.5/10		ENPUMP			--		--	S_PCK	00000000
1CH	NETE	4/10/11				ENVS	SILB[1:0]		ENLB		00000000
1DH	NETF	4/10/11		ENBAND	ENVDDA				ENAGND	ENVB	00000000
1FH	SVD	4.5								LBOUT	uuuuuuuu

Register PCK at address 15H

property	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0	R/W-0
PCK		**ENPUMP**			**--**		**--**	**S_PCK**
	Bit7							Bit0

Bit 6 **ENPUMP**: Voltage Doubler enabled flag

1 = Voltage Doubler is enabled

0 = Voltage Doubler is disabled

Bit 0 **S_PCK**: Voltage Doubler operation frequency selector

1 = Voltage Doubler Operation Frequency = MCK/100 (Please see Chapter 5)

0 = Voltage Doubler Operation Frequency = MCK/200 (Please see Chapter 5)

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register NETE at address 1CH

property	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
NETE				**ENVS**	**SILB[1:0]**		**ENLB**	
	Bit7							Bit0

Bit 4 **ENVS**: VDDA Voltage Source enable flag (Please read Section 4.2 for detailed description)

1 = VDDA is connected to VS. VS could be used as a voltage source.

0 = VDDA and VS are disconnected.

Bit 3-2 **SILB[1:0]**: Low Battery Comparator Input Selector (Please refer to Section 4.5 for detailed description)

11 = No definition. The Low Battery Comparator Input is floating.

10 = Low Battery Comparator Input is selected as external analog input AIN4

01 = Low Battery Comparator Input is selected as 3.65V

00 = Low Battery Comparator Input is selected as 2.45V

Bit 1 **ENLB**: Low Battery Comparator enable flag (Please refer to Section 4.5 for detailed description)

1 = Low Battery Comparator is enabled

0 = Low Battery Comparator is disabled

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register NETF at address 1DH

property	U-0	R/W-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
NETF		ENBAND	ENVDDA				ENAGND	ENVB
	Bit7							Bit0

Bit 6 ENBAND: Band gap Voltage enable flag (Please refer to Section 4.6 for detailed description)
1 = The Band gap Voltage and Temperature Sensor are enabled, REFO to AGND is about 1.16V
0 = The Band gap Voltage and Temperature Sensor are disabled

Bit 5 ENVDDA: Voltage Regulator enable flag (Please refer to Section 4.5 for detailed description)
1 = Voltage Regulator is enabled, VDDA is 3.6V
0 = Voltage Regulator is disabled. VDDA can be from external power supply.

Bit 1 ENAGND: Analog Common Voltage Generator enable flag
(Please see Section 4.4 for detailed description)
1 = Analog Common Voltage Generator is enabled. AGND = 1/2 VDDA
0 = Analog Common Voltage Generator is disabled. AGND is floating.

Bit 0 ENVB: Analog Bias Circuit enable flag (Please see Section 4.3 for detailed description)
1 = Analog Bias Circuit is enabled. Analog system (ADC and OPAMP) can work correctly.
0 = Analog Bias Circuit is disabled. Analog system can NOT work

Register SVD at address 1FH

property	U-X	U-X	U-X	U-X	U-X	U-X	U-X	R-X
SVD								LBOUT
	Bit7							Bit0

Bit 0 LBOUT: Low Battery Comparator output (Please refer to Section 4.5 for detailed description)
1 = The Voltage selected by SILB[1:0] is higher than 1.2V.
0 = The Voltage selected by SILB[1:0] is lower than 1.2V

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

4.1. Voltage Doubler

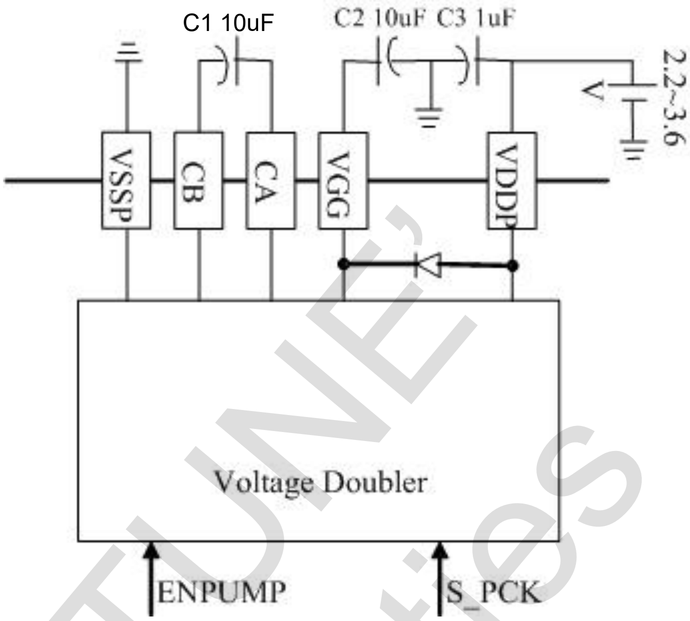


Figure 4-2 Voltage Doubler

Voltage Doubler is used for generating VGG which provide input[4] for VDDA Voltage Regulator. The inputs of Voltage Doubler are VDDP, VSSP, CA and CB. The related registers are S_PCK and ENPUMP. The Output is VGG. Please see Figure 4-2.

Table 4-2 Voltage Doubler register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
14H	MCK	5	M7_CK	M6_CK	M5_CK		M3_CK	M2_CK	M1_CK	M0_CK	00000000
15H	PCK	4/5/7.5/10		ENPUMP			--		--	S_PCK	00000000

Operations:

1. Connect the pins VDDP and VSSP to VDD (2.2V~3.6V) and VSS (system ground).
2. Put a 10Uf capacitance between CA and CB.
3. Select the Voltage Doubler Operation frequency by setting S_PCK and M0_CK[5] according to the following table
4. Set the ENPUMP flag.
5. The output, VGG, will be 2 times of VDDP.

[4] Please refer to Section 4.2 for detailed description about VDDA and Voltage regulator.
[5] M0_CK is the 1st bit of the MCK register. Please refer to Section 5.0

Table 4-3 Voltage Doubler operation frequency selection table

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

If the user doesn't want the VGG to be generated from the Voltage Doubler, then the ENPUMP should be set to disable the voltage Doubler, and input the VGG pin a voltage as voltage regulator power supply.

4.2. Voltage Regulator

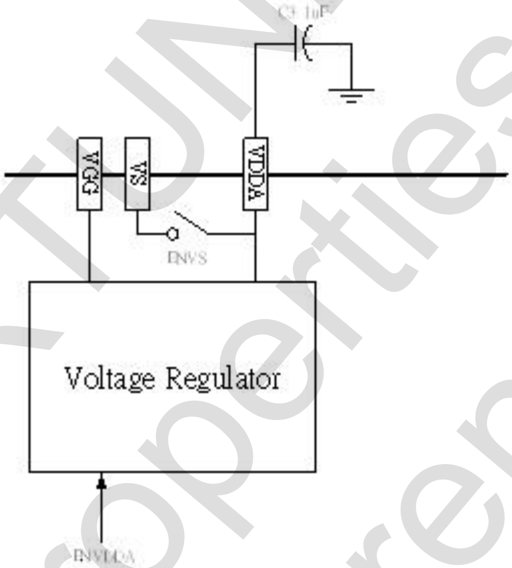


Figure 4-3 Voltage regulator

Voltage Regulator is used for generating VDDA (3.6V). The input is VGG which is generated by Voltage Doubler (please see the Section 4.1). The control Register flags are ENVDDA and ENVS. The Outputs are VDDA and VS. Please see Figure 4-3.

Table 4-4 voltage regulator register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
1CH	NETE	4/10/11				ENVS	SILB[1:0]		ENLB		00000000
1DH	NETF	4/10/11		ENBAND	ENVDDA				ENAGND	ENVB	00000000

Operations

1. Operate as Section 4.1 to get the VGG (2 times of VDD or external Power Supply).
2. Set the ENVDDA flag.
3. The output, VDDA, is 3.6V.
4. If the user wants VDDA as output voltage source, then the ENVS flag should be set. VS will be the same as VDDA.

4.3. Analog Bias Circuit

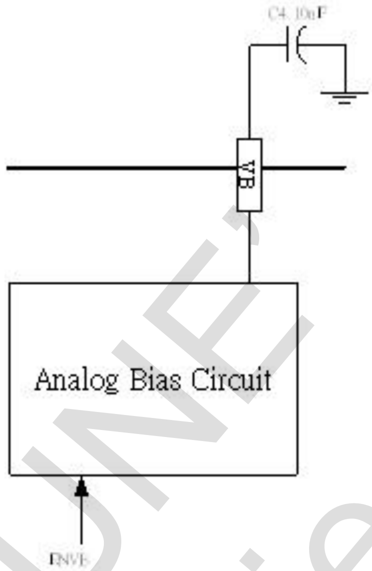


Figure 4-4 analog bias circuit

Analog Bias Circuit is used to activate VB (reference VDDA) as the power supply voltage for analog circuit (include ADC, OPAMP, Low Battery Comparator) and LCD driver. The Control register flag is ENVB. Please see Figure 4-4.

Table 4-5 analog bias circuit register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
1DH	NETF	4/10/11		ENBAND	ENVDDA				ENAGND	ENVB	00000000

Operation:

1. Operate as Section 4.1 to get the VGG (2 times of VDD or external Power Supply).
2. Operate as Section 4.2 to get the VDDA (3.6V).
3. Set the ENVB flag. The VB will be 3.6V (same as VDDA) and the analog function network and the LCD driver can be activate correctly.
4. Note that Pin VB must be connected with a 10Nf capacitor to VSS for reducing Voltage Doubler noise.

4.4. Analog Common Voltage Generator

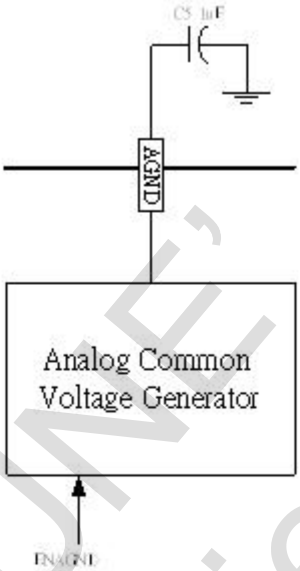


Figure 4-5 analog common voltage generator

Analog Common Voltage Generator is used to provide a voltage at the halt of AGND as 1/2 VDDA[6]. The Control register is ENAGND and the output is AGND. Please see Figure 4-5.

Table 4-6 analog common voltage generator register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
1DH	NETF	4/10/11		ENBAND	ENVDDA				ENAGND	ENVB	00000000

Operation:

1. Operate following the steps Chapter 4.1 to get the VGG (2 times VDD or external Power Supply).
2. Operate as Section 4.2 to get the VDDA (3.6V)
3. Operate as Section 4.3 to activate the Analog Bias Circuit
4. Set the ENAGND register flag.
5. The output, AGND, will be 1/2 VDDA

[6] When VDDA is 3.6V, AGND would be 1.8V

4.5. Low Battery Comparator

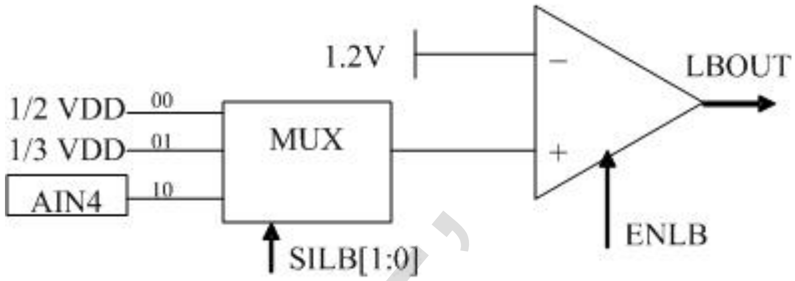


Figure 4-6 low battery comparator function block

Low Battery Comparator is used for VDD low voltage detection. FS98O25 embeds a voltage divider which can generate 1/2 VDD and the 1/3 VDD. A multiplexer is used to connect the voltage divides to component input. The multiplexer's output is compares with 1.2V. The Control register flags are SILB[1:0] and the ENLB. The Output flag is LBOUT which is for read only. Please see Figure 4-6.

Table 4-7 low battery comparator register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
1CH	NETE	4/10/11				ENVS	SILB[1:0]		ENLB		00000000
1FH	SVD	4.5								LBOUT	uuuuuuuu

Operation:

1. Operate as Section 4.1 to get the VGG (2 times VDD or external Power Supply).
2. Operate as Section 4.2 to get the VDDA (3.6V)
3. Operate as Section 4.3 to active the Analog Bias Circuit
4. Set SILB to choose the Comparator input. Please see Table 4-8

Table 4-8 low battery comparator voltage detection selection table

SILB [1:0]	Detection Voltage	if LBOUT = 1
00	1/2 VDD	VDD > 2.3 volt
01	1/3 VDD	VDD > 3.5 volt
10	AIN4	AIN > 1.2 volt

5. Set the ENLB register flag, and the Low Battery Comparator is enabled.
6. The output, LBOUT, is the result of the comparator.

4.6. Bandgap Voltage and Temperature Sensor

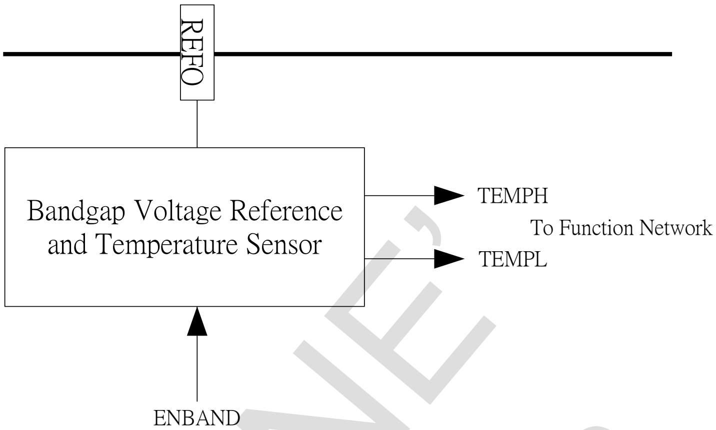


Figure 4-7 Bandgap voltage and temperature sensor function block

REFO is low temperature coefficient bandgap voltage reference output. Its voltage to AGND is 1.16V, and the typical temperature coefficient is 150ppm/°C.

FS98O25 embeds a Temperature Sensor to measure the IC temperature from the differential voltage between TEMPH and TEMPL (typically 550mV±50mV/°C). Its working range is 100 ~ 200 mV. User can connect the TEMPH and TEMPL to an ADC to get the IC temperature. Please refer to Chapter 10 and Chapter 11 for detailed instruction of ADC.

Both the bandgap Voltage Reference and the Temperature sensor are controlled by ENBAND register flag.

Please see Figure 4-7.

Table 4-9 bandgap voltage and temperature sensor register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
1DH	NETF	4/10/11		ENBAND	ENVDDA				ENAGND	ENVB	00000000

Operation:

1. Operate as Section 4.1 to get the VGG (2 times VDD or external Power Supply).
2. Operate as Section 4.2 to get the VDDA (3.6V)
3. Operate as Section 4.3 to enable the Analog Bias Circuit
4. Set the ENBAND register flag.
5. Check REFO. Its value with respect to AGND should be about 1.16V
6. The output, TEMPH and TEMPL, will show the IC temperature as the differential voltage.

5. Clock System

Table 5-1 FS98O25 clock system register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
14H	MCK	5	M7_CK	M6_CK	M5_CK		M3_CK	M2_CK	M1_CK	M0_CK	00000000
15H	PCK	4/5/7.5/10		ENPUMP			S_CH1CK [1:0]		S_BEEP	S_PCK	00000000

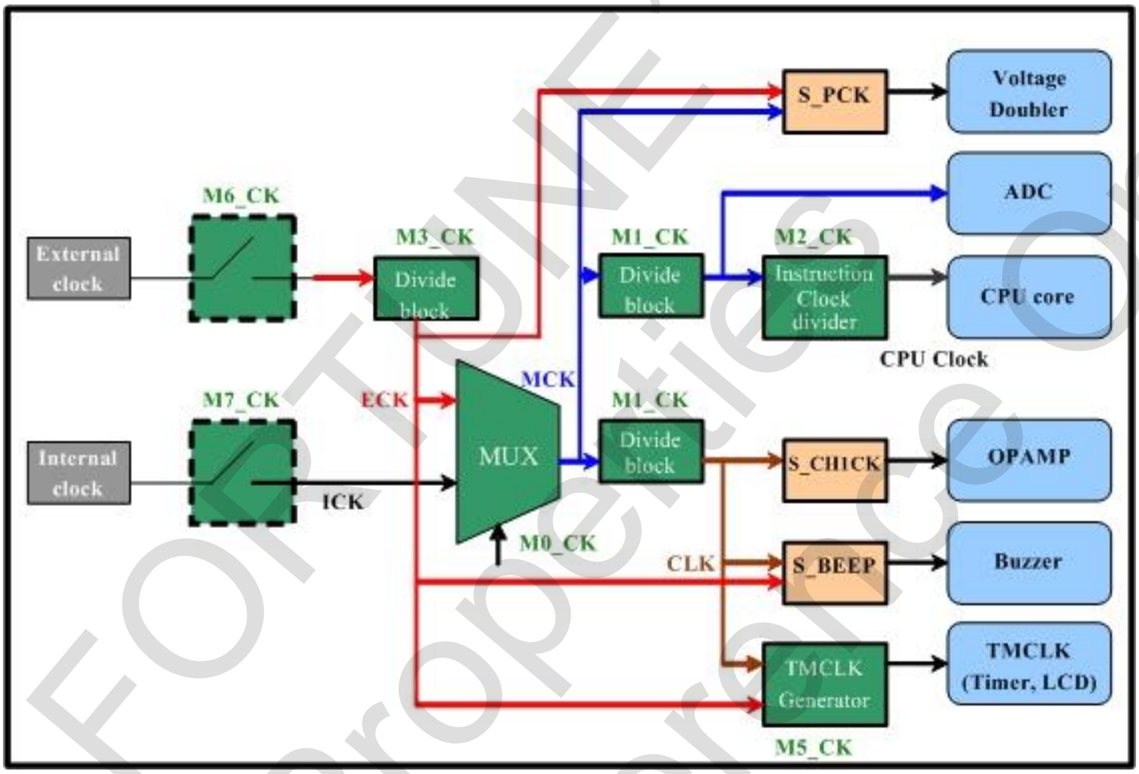


Figure 5-1 FS98O25 clock system function block

The clock system provides clock signals for the following 7 function blocks: ***Voltage Doubler***, ***ADC***, ***CPU core***, ***OPAMP***, ***Buzzer***, ***Timer module*** and ***LCD***. Users could use 10 register flags to generate all kinds of clock signals for the above 7 function blocks. These 10 register flags are M0_CK, M1_CK, M2_CK, M3_CK, M5_CK, M6_CK, M7_CK, S_PCK, S_CH1CK[1:0] and S_BEEP. The detailed setup will be illustrated in following sections. Please see Figure 5-1.

5.1. Oscillator State

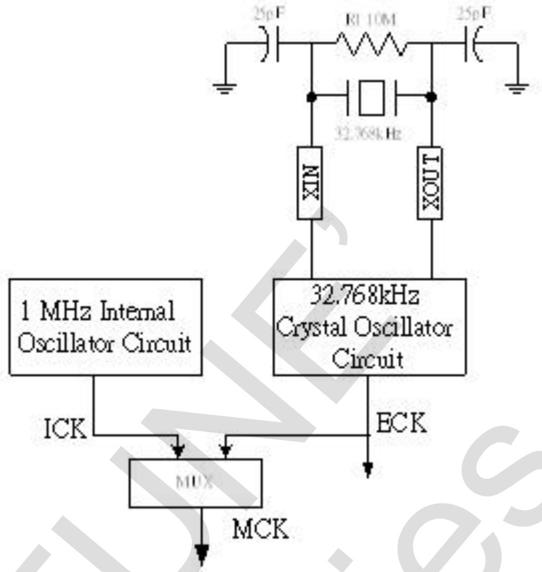


Figure 5-2 FS98O25 oscillator state block

Table 5-2 FS98O25 clock system register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
14H	MCK	5	M7_CK	M6_CK	M5_CK		M3_CK	M2_CK	M1_CK	M0_CK	00000000

There are two clock sources in FS98O25. One is the internal clock which generates 1 MHZ for CPU, and the other is an external one which provides 32768 HZ clock signal to the Chip. Users should choose one clock to use as MCK. Please see Figure 5-2.

There are 2 clock signals working in FS98O25: MCK and CLK. Users should use Table 5-2 and 5-3 to setup MCK and CLK based on the M0_CK, M1_CK and M3_CK.

Table 5-3 MCK selection table

M3_CK	M0_CK	**MCK**
X	0	ICK
0	1	ECK
1	1	ECK/2

Table 5-4 CLK selection table

M1_CK	**CLK**
0	MCK
1	MCK/4

To enable the internal and external oscillators, users need to set the right values for M7_CK and M6_CK as shown in Table 5-4. If users execute the sleep instruction to make FS98O25 enter the SLEEP mode, both the internal oscillators and the external oscillator will be disabled.

Table 5-5 oscillator state selection table

Input			Oscillator State	
Sleep instruction	**M7_CK**	**M6_CK**	**Internal**	**External**
1	X[7]	X	Disable	Disable
0	0	0	Enable	Enable
0	0	1	Enable	Disable
0	1	0	Disable	Enable
0	1	1	Enable	Disable

5.2. CPU Instruction Cycle

Table 5-6 FS98O25 CPU instruction cycle register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
14H	MCK	5	M7_CK	M6_CK	M5_CK		M3_CK	M2_CK	M1_CK	M0_CK	00000000

User can setup M0_CK, M1_CK, M2_CK and M3_CK to select the instruction cycle[8]. In order to maintain a stable ADC output, user could clear M2_CK to make CPU have a different operation clock cycle from ADC. In the applications where a resolution of ADC is more than 13 bits, M2_CK should be set to zero.

Table 5-7 MCK selection table

M3_CK	M0_CK	MCK
X	0	ICK (1MHZ)
0	1	ECK (32768 HZ)
1	1	ECK/2 (16384HZ)

Table 5-8 instruction cycle selection table

M2_CK	M1_CK	Instruction Cycle
0	0	MCK/6.5
0	1	MCK/12.5
1	0	MCK/2
1	1	MCK/4

7 X means "don't care"
8 Users must make sure that switching from one oscillator to the other can be made only after the oscillator's output is stabilized.
An NOP command should be added after the switching.

5.3. ADC Sample Frequency

FS98O25 embeds one sigma delta ADC which needs clock input to generate digital output. When users want ADC have N bits resolution digital output, ADC needs 2^N clocks cycles input. (Please refer to Chapter 10 and Chapter 11 for detailed description) User should setup the M1_CK to decide the ADC sample frequency. Please see Table 5-9.

Table 5-9 ADC sample frequency selection table

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

5.4. Beeper Clock

Table 5-10 beeper clock register table

Address	**Name**	**Referenced Section**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Value on Power on Reset**
14H	MCK	5	M7_CK	M6_CK	M5_CK		M3_CK	M2_CK	M1_CK	M0_CK	00000000
15H	PCK	4/5/7.5/10		ENPUMP			S_CH1CK [1:0]		S_BEEP	S_PCK	00000000

FS98O25 has a Beeper Clock which is used as the buzzer source. (Please refer to Section 7.5 for how to use Buzzer) User could change the Beeper clock frequency by setting M0_CK, M1_CK, M3_CK and S_BEEP register flags according to Table 5-11, Table 5-12 and Table 5-13.

Table 5-11 MCK selection table

M3_CK	M0_CK	**MCK**
X	0	ICK
0	1	ECK
1	1	ECK/2

Table 5-12 CLK selection table

M1_CK	**CLK**
0	MCK
1	MCK/4

Table 5-13 beeper clock selection table

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

Table 5-14 shows the relation between clock signals and the register flags. Please see Table 5-14)

Table 5-14 register and the beeper clock selection table

M0_CK	M1_CK	M3_CK	S_BEEP	MCK	CLK	beep clock
1	0	0	1	32768	32768	**4096**
1	0	1	1	16384	16384	**4096**
1	1	0	1	32768	8192	**4096**
1	1	1	1	16384	4096	**4096**
0	0	0	0	1000000	1000000	**4000**
0	0	1	0	1000000	1000000	**4000**
0	0	0	1	1000000	1000000	**2666.6667**
0	0	1	1	1000000	1000000	**2666.6667**
0	1	0	0	1000000	250000	**1000**
0	1	1	0	1000000	250000	**1000**
0	1	0	1	1000000	250000	**666.6667**
0	1	1	1	1000000	250000	**666.6667**
1	0	0	0	32768	32768	**131.072**
1	0	1	0	16384	16384	**65.536**
1	1	0	0	32768	8192	**32.768**
1	1	1	0	16384	4096	**16.384**

5.5. Voltage Doubler Operation Frequency

FS98O25 embeds a switching voltage regulator. Users can use M0_CK and S_PCK register flags to decide the operation frequency as in Table 5-15 and Table 5-16.

Table 5-15 MCK selection table

M3_CK	M0_CK	**MCK**
X	0	ICK
0	1	ECK
1	1	ECK/2

Table 5-16 Voltage Doubler operation frequency selection table

M0_CK	S_PCK	Voltage Doubler Operation Frequency
0	0	MCK/200
0	1	MCK/100
1	X	ECK/32 (1024 HZ)

5.6. Chopper Operation Amplifier Input Control Signal

The OPAMP embedded in FS98O25 has a chopper function to cancel the inverting and non-inverting sides voltage bias offsets. After the Chopper operation, OPAMP input voltage bias is removed. Users could setup the S_CH1CK[1:0] to choose the Chopper Control Signal. (Please see Table 5-17, Table 5-18 and Table 5-19)

Table 5-17 CLK selection table

M1_CK	**CLK**
0	MCK
1	MCK/4

Table 5-18 MCK selection table

M3_CK	M0_CK	**MCK**
X	0	ICK
0	1	ECK
1	1	ECK/2

Table 5-19 chopper control signal selection table

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

5.7. TMCLK – Timer and LCD Module Input Clock

TMCLK is the clock for FS98O25 Timer and LCD Module. Users can use Table 5-20 to choose TMCLK frequency by setting the right values for M5_CK.

Table 5-20 TMCLK selection table

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

6. Timer Module, Watch Dog Timer and Programmable Counter

Table 6-1 Timer module and watch dog timer register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
04H	STATUS	1.11/3.4.2				--	TO	--	--	--	00u00uuu
06H	INTF	3/6/7/9/10/11				TMIF	--	--	--	--	00000000
07H	INTE	3/6/7/9/10/11	GIE			TMIE	--	--	--	--	00000000
08H	CTAH	6.3	CTA[15:8]								uuuuuuuu
09H	CTAL	6.3	CTA[7:0]								uuuuuuuu
0AH	CTBH	6.3	CTB[15:8]								uuuuuuuu
0BH	CTBL	6.3	CTB[7:0]								uuuuuuuu
0CH	CTCON	6.3	TON	MUXSEL[2:0]			TE	FQTMB	OVAB		0000000u
0DH	WDTCON	6.2	WDTEN					WTS [2:0]			0uuuu000
0EH	TMOUT	6.1	TMOUT [7:0]								00000000
0FH	TMCON	6.1	TRST				TMEN	INS [2:0]			1uuu0000

The Registers are described as follows.

Register CTAH at address 08H

property	R –X	R-X	R-X	R-X	R-X	R-X	R-X	R-X
CTAH	**CTA[15:8]**							
	Bit7							Bit0

Register CTAL at address 09H

property	R-X	R-X	R-X	R-X	R-X	R-X	R-X	R-X
CTAL	**CTA[7:0]**							
	Bit7							Bit0

Bit 15-0 **CTA[15:0]**: Programmable Counter 16-bit Counter A register (Please refer to Section 6.3 for detail)

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register CTBH at address 0AH

property	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
CTBH	**CTB[15:8]**							
	Bit7							Bit0

Register CTBL at address 0BH

property	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
CTBL	**CTB[7:0]**							
	Bit7							Bit0

Bit 15-0 **CTB[15:0]**: Programmable Counter 16-bit Counter B register (Please refer to Section 6.3 for detail)

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register CTCON at address 0CH

property	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-X
CTCON	**TON**	**MUXSEL[2:0]**			**TE**	**FQTMB**	**OVAB**	**--**
	Bit7							Bit0

Bit 7 **TON**: 16-bit Counter input signal switch (Please refer to Section 6.2 for detail)
1 = The 16-bit Counter input signal switch is ON.
0 = The 16-bit Counter input signal switch is OFF.

Bit 6 **MUXSEL[2]**: Programmable Counter Counter/Pulse Width measurement mode selector.
1 = Programmable Counter is used as Pulse Width measurement.
0 = Programmable Counter is used as General Counter.

Bit 5-4 **MUXSEL[1:0]**: Counter A clock source select multiplexer 1.
11 = PFI, GPIO 3 port 2.
10 = ECK, External Clock (32768HZ)
01 = Instruction clock, please see Section 5.2
00 = ICK, Internal Clock (1MHZ)

Bit 3 **TE**: PFI signal inverting register
1 = PFI signal is inverted
0 = PFI signal is NOT inverted

Bit 2 **FQTMB**: Programmable Counter Frequency measurement mode enabled register flag.
1 = Programmable Counter is used as Frequency measurement.
0 = Programmable Counter is used as General Counter or Pulse Width measurement.

Bit 1 **OVAB**: Programmable Counter interrupt source selector
1 = Programmable Counter interrupt source is Counter A.
0 = Programmable Counter interrupt source is Counter B.

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

Register WDTCON at address 0DH

property	R/W-0	U-X	U-X	U-X	U-X	R/W-0	R/W-0	R/W-0
WDTCON	**WDTEN**					**WTS [2:0]**		
	Bit7							Bit0

Bit 7 **WDTEN**: Watch Dog Timer enable flag (Please refer to Section 6.2 for detail)
1 = Watch Dog Timer is enabled.
0 = Watch Dog Timer is disabled

Bit 2-0 **WTS [2:0]**: Watch Dog Timer counter 2 Input Selector (Please refer to Chapter 6.2 for details)
111 = Watch Dog Timer Counter 2 Input is WDTA[0]
110 = Watch Dog Timer Counter 2 Input is WDTA[1]
101 = Watch Dog Timer Counter 2 Input is WDTA[2]
100 = Watch Dog Timer Counter 2 Input is WDTA[3]
011 = Watch Dog Timer Counter 2 Input is WDTA[4]
010 = Watch Dog Timer Counter 2 Input is WDTA[5]
001 = Watch Dog Timer Counter 2 Input is WDTA[6]
000 = Watch Dog Timer Counter 2 Input is WDTA[7]

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

Register TMOUT at address 0EH

property	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TMOUT	**TMOUT [7:0]**							
	Bit7							Bit0

Bit 7-0 **TMOUT [7:0]**: Timer module 8-bit counter output (Please refer to Section 6.1 for detail)

Register TMCON at address 0FH

property	R/W-1	U-X	U-X	U-X	R/W-0	R/W-0	R/W-0	R/W-0
TMCON	**TRST**				**TMEN**	**INS [2:0]**		
	Bit7							Bit0

Bit 7 **TRST**: Timer Module reset flag (Please refer to Section 6.1 for detail)
1 = Timer Module Counter works normally.
0 = Timer Module Counter is reset.(After resetting the Counter, TRST will reset itself)

Bit 3 **TMEN:** Timer Module enable flag (Please refer to Section 6.1 for detail)
1 = Timer Module Counter will active.
0 = Timer Module Counter will be disabled.

Bit 2-0 **INS [2:0]**: Timer Module interrupt Signal Selector (Please refer to Chapter 6.1 for detail)
111 = TMOUT[7] is selected as Timer Module interrupt Signal
110 = TMOUT[6] is selected as Timer Module interrupt Signal
101 = TMOUT[5] is selected as Timer Module interrupt Signal
100 = TMOUT[4] is selected as Timer Module interrupt Signal
011 = TMOUT[3] is selected as Timer Module interrupt Signal
010 = TMOUT[2] is selected as Timer Module interrupt Signal
001 = TMOUT[1] is selected as Timer Module interrupt Signal
000 = TMOUT[0] is selected as Timer Module interrupt Signal

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

There are two timers in FS98O25: Timer Module and Watch Dog Timer. Please see the following sections for detail.

6.1. Timer Module

The Timer module has the following features:

- 8-bit Timer Counter
- Internal (1 MHZ) or External (32768HZ) clock selection
- Time out Interrupt Signal selection

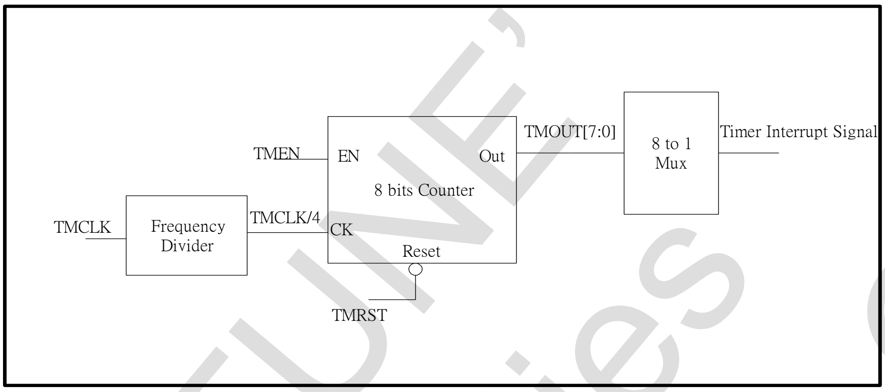


Figure 6-1 FS98O25 timer module function block

Please see Figure 6-1. The input of Timer Module is TMCLK. (Please refer to Section 5.7 for the detailed setting) FS98O25 embeds a Frequency Divider in the Timer Module to divide the TMCLK by 4, and treats the divided clock signal as 8-bit counter input clock. When a user sets the Timer Module enable flag, the 8-bit counter will activate, and the TMOUT[7:0] will increase from 0x00H to 0Xffh. User needs to setup INS (Timer Module interrupt Signal Selector) to select the time out interrupt signal. When timer out event happens, the interrupt Flag will set itself and the program counter will jump to 0x04H for ISR (Interrupt Service Routine)

6.1.1. Timer module interrupt

Table 6-2 timer module interrupt register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
06H	INTF	3/6/7/9/10/11				TMIF	--	--	--	--	00000000
07H	INTE	3/6/7/9/10/11	GIE			TMIE	--	--	--	--	00000000
0EH	TMOUT	6.1	TMOUT [7:0]								00000000
0FH	TMCON	6.1	TRST				TMEN	INS [2:0]			1uuu0000

Operation:

1. Operate as Section 5.7 to setup the TMCLK for Timer module input
2. Setup the INS[2:0] to select timer interrupt source. Please see Table 6-3.
3. Set the TMIE and GIE register flags to enable the Timer interrupt.
4. Set the TMEN register flag to enable Timer module 8-bit counter.
5. Clear the TRST register flag to reset the Timer module 8-bit counter
6. When time out event happens, TMIF register flag will reset itself, and the program counter will reset to 0x04H

Table 6-3 timer selection table

INS[2:0]	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.

6.1.2. Using Timer with External/Internal Clock

The user could see the Table 6-4, 6-5, 6-6 and 6-7 to setup related registers to decide the clock source.

Table 6-4 external timer setup register table

Address	Name	Detail on Chapter	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
14H	MCK	5	M7_CK	M6_CK	M5_CK		M3_CK	M2_CK	M1_CK	M0_CK	00000000

Table 6-5 CLK selection table

M1_CK	**CLK**
0	MCK
1	MCK/4

Table 6-6 MCK selection table

M3_CK	M0_CK	**MCK**
X	0	ICK
0	1	ECK
1	1	ECK/2

Table 6-7 TMCLK selection table

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

Users can use Table 6-8 to select TMCLK clock source based on M0_CK, M1_CK, M3_CK and M5_CK register flag.

Table 6-8 registers and timer selection table

M0_CK	M1_CK	M3_CK	M5_CK	MCK	CLK	TMCLK
0	0	0	1	1000000	1000000	**1024**
0	0	1	1	1000000	1000000	**1024**
0	1	0	1	1000000	500000	**1024**
0	1	1	1	1000000	500000	**1024**
1	0	0	1	32768	32768	**1024**
1	1	0	1	32768	16384	**1024**
1	0	1	1	16384	16384	**1024**
1	1	1	1	16384	8192	**1024**
0	0	0	0	1000000	1000000	**1000**
0	0	1	0	1000000	1000000	**1000**
0	1	0	0	1000000	500000	**500**
0	1	1	0	1000000	500000	**500**
1	0	0	0	32768	32768	**32.768**
1	1	0	0	32768	16384	**16.384**
1	0	1	0	16384	16384	**16.384**
1	1	1	0	16384	8192	**8.192**

6.2. Watch Dog Timer

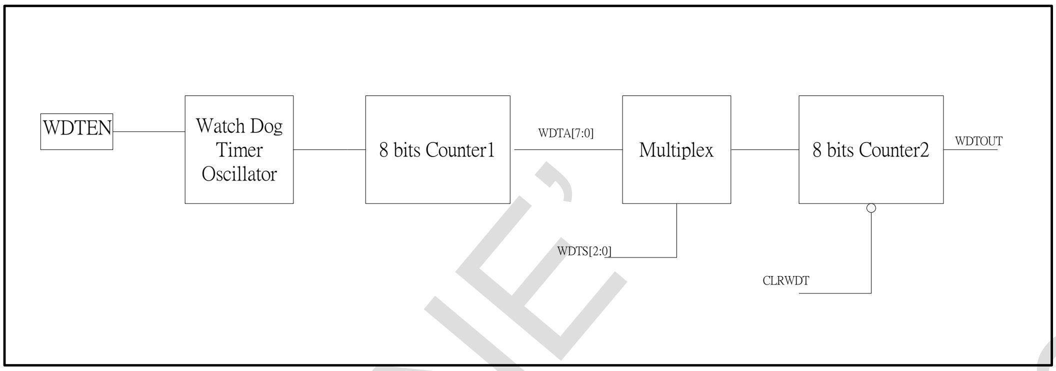


Figure 6-2 watch dog timer function block

Please see Figure 6-2. WDT (Watch Dog Timer) is used to prevent the program from being out of control by any uncertain reason. When WDT is active, it will reset the CPU when the WDT timeout. Generally, the program run in FS98O25 needs to reset the WDT before the WDT times out every time to reset the CPU. When some trouble happens, the program will be reset to the general situation by WDT and the program won't reset the WDT in that situation.

The input of Watch Dog Timer is WDTEN and WDTS[2:0] register flags. The output of Watch Dog Timer is TO register flag. When a user sets the WDTEN, the embedded Watch Dog Timer Oscillator (3 KHZ) will become active, and the generated clock will be pushed into the "8-bit counter 1" as shown in Figure 6-2. The output of the "8-bit counter 1", WDTA[7:0], is a virtual signal which is sent to one multiplexer. The multiplexer is controlled by the register flags, WDTS[2:0]. The output signal is used as the "8-bit Counter 2" clock input. When "8-bit Counter 2" overflows, it will send WDTOUT to reset the CPU (Program Counter will jump to 0x00H to reset the program) and set TO flag. Users could reset the WDT by the instruction – CLRWDT.

Table 6-9 watch dog timer register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
04H	STATUS	1.11/3.4.2				--	TO	--	--	--	00u00uuu
0DH	WDTCON	6.3	WDTEN					WDTS [2:0]			0uuuu000

Operation:

1. Setup the WDTS[2:0] to decide the WDT timeout frequency.
2. Set WDTEN register flag to enable the WDT.
3. Process the CLRWDT instruction to reset the WDT in the program.

6.3. Dual 16-bit Programmable Counter

Address	Name	Content (u mean unknown or unchanged)								Reset State
07H	INTE	GIE			-	-	-	-	-	00000000
16H	INTF2						CTIF	-	-	00000000
17H	INTE2						CTIE	-	-	00000000
08H	CTAH	CTA[15:8]								uuuuuuuu
09H	CTAL	CTA[7:0]								uuuuuuuu
0AH	CTBH	CTB[15:8]								uuuuuuuu
0BH	CTBL	CTB[7:0]								uuuuuuuu
0CH	CTCON	TON	MUXSEL[2:0]			TE	FQTMB	OVAB		0000000u
2BH	PT3MR				PFOEN	-		-		00000000

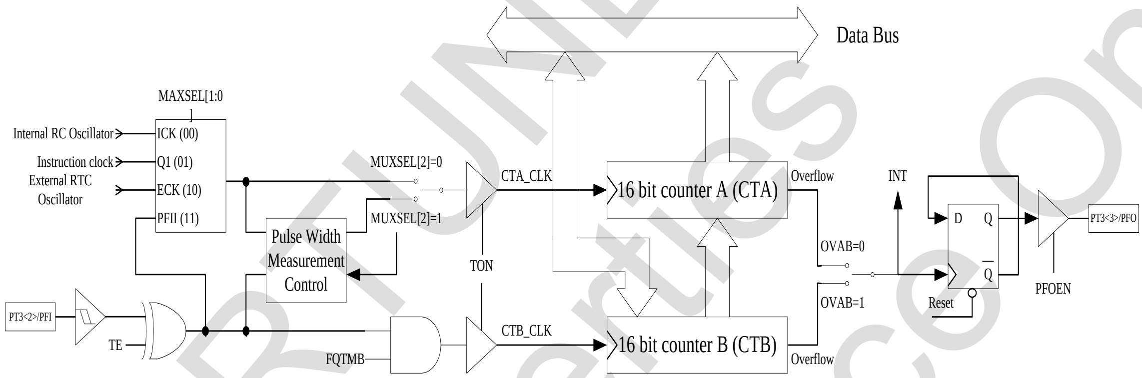


Figure 6-3 Programmable Counter Working block diagram

FS98O25 embeds Dual 16-bit Programmable Counter. It could be used under three working modes: Counter mode, Pulse Width Measurement mode and Frequency Measurement mode. Users could setup MUXSEL[2] and FQTMB register flags to decide the working mode.

Table 6-10 Programmable Counter working mode selection table

Working mode	MUXSEL[2]	FQTMB
Counter mode	0	0
Pulse Width Measurement mode	1	0
Frequency Measurement mode	0	1
NONE	1	1

- **Counter mode:**
  There are two 16-bit counters (CTA and CTB) in Programmable Counter unit.

Operation:

1. Clear FQTMB and MUXSEL[2] register flags to make the Programmable Counter work as Counter.
2. Setup MUXSEL[1:0] to decide the input clock signal.

Table 6-11 Programmable Counter Clock signal selection table

MUXSEL[1:0]	Clock signal
11	PFI
10	ECK
01	Instruction Cycle
00	ICK

3. If PFI is assigned to be the Clock signal, users could set TE to invert the PFI voltage level.
4. Clear OVAB register flag to set the CTA as the working counter. When CTA counter overflows, the interrupt (CTIE) will be triggered.
5. Clear CTIF and set the CTIE and GIE register flag to enable the Programmable Counter interrupt.
6. Setup CTB[15:0]. CTA[15:0] will be filled with the same value as CTB[15:0]. When CTA[15:0] overflows, it will be filled with the same value again. User could decide CTA timeout by setting up CTB[15:0] register.
7. Set TON to start the counter.
8. When CTA counter overflows, the interrupt will be triggered.
9. Users could clear TON register flag to stop the counting process.

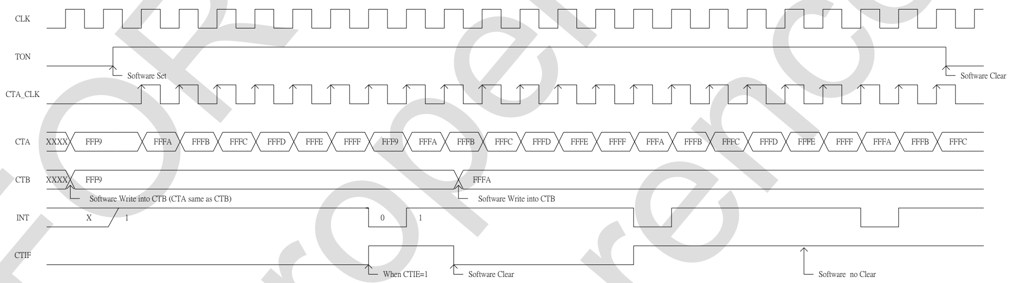


Figure 6-4 Programmable Counter Counter mode

- **Pulse Width Measurement mode:**
  Programmable Counter could be used to measure the time when a signal holds its voltage level in high or low.

Operation:

1. Clear FQTMB and clear MUXSEL[2] register flags to make the Programmable Counter work as Pulse Width Measurement.
2. Setup MUXSEL[1:0] to decide the input clock signal.
3. PFI is the signal which is ready to measure the pulse width. Users could set TE to invert the PFI voltage level.
4. Clear OVAB register flag to set the CTA as the working counter. When CTA counter overflows, the interrupt (CTIE) will be triggered.
5. Clear CTIF and set the CTIE and GIE register flag to enable the Programmable Counter interrupt.
6. Setup CTB[15:0]. CTA[15:0] will be filled with the same value as CTB[15:0]. When CTA[15:0] overflows, it will be filled with the same value again. User could decide CTA timeout by setting up CTB[15:0] register.
7. Set TON to start the Pulse Width Measurement.
8. When PFI signal is from high to low, CTA counter will stop counting and clear TON register flag. Interrupt will be triggered at the same time. Users could read the CTA counter value to know the pulse width of PFI.
9. If CTA counter overflows, and the PFI signal is still high, the interrupt will be triggered, but CTA will count again.

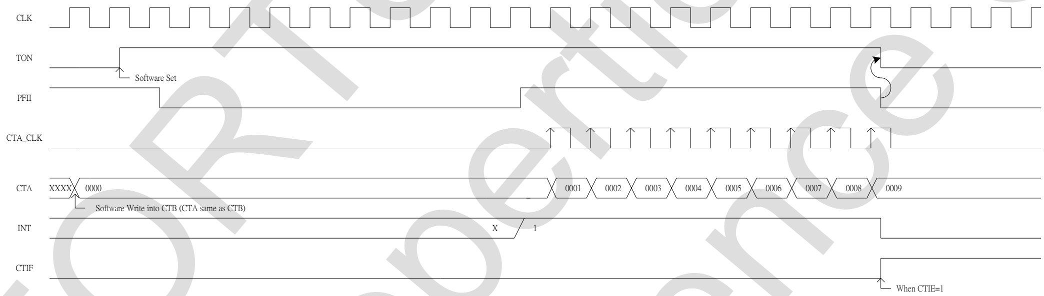


Figure 6-5 Programmable Counter Pulse Width Measurement mode

- **Frequency Measurement mode:**
  Programmable Counter could be used to measure a signal frequency.

Operation:

1. Set FQTMB and clear MUXSEL[2] register flags to make the Programmable Counter work as Frequency Measurement.
2. Setup MUXSEL[1:0] to decide the input clock signal.
3. PFI is the signal which is ready to measure the frequency. Users could set TE to invert the PFI voltage level.
4. Clear OVAB register flag to set the CTA as the working counter. When CTA counter overflows, the interrupt (CTIE) will be triggered.
5. Clear CTIF and set the CTIE and GIE register flags to enable the Programmable Counter interrupt.
6. Setup CTB[15:0]. CTA[15:0] will be filled with the same value as CTB[15:0]. When CTA[15:0] overflows, it will be filled with the same value again. User could decide CTA timeout by setting up CTB[15:0] register.
7. Set TON to start the Frequency Measurement.
8. When CTA counter overflows, the interrupt will be triggered. TON register flag will be clear automatically.
9. Users could read the CTB value to know the PFI signal frequency.

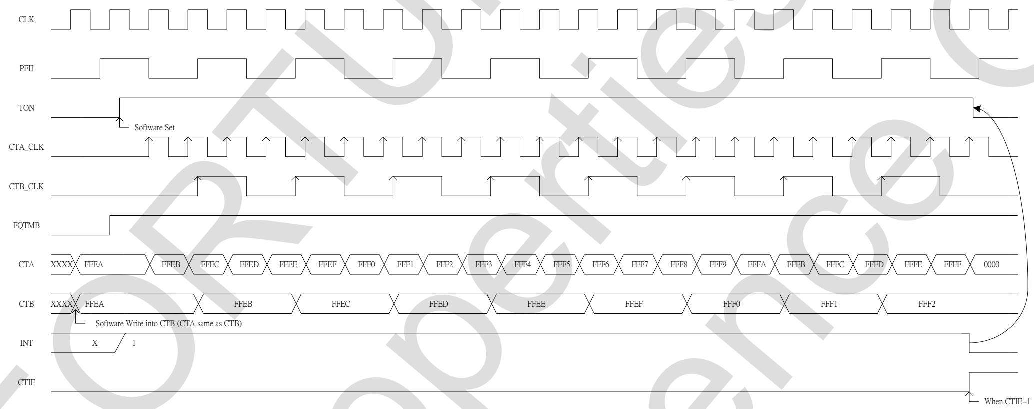


Figure 6-6 Programmable Counter Frequency Measurement mode

7. I/O Port

Table 7-1 FS98O25 I/O port register table

Address	Name	Detail on Chapter	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
06H	INTF	3/6/7/9/10/11				--	I2CIF	-	E1IF	E0IF	00000000
07H	INTE	3/6/7/9/10/11	GIE			--	I2CIE	-	E1IE	E0IE	00000000
16H	INTF2	6/7						--	E3IF	E2IF	00000000
17H	INTE2	6/7						--	E3IE	E2IE	00000000
20H	PT1	7	PT1 [7:0]								uuuuuuuu
21H	PT1EN	7	PT1EN [7:0]								00000000
22H	PT1PU	7	PT1PU [7:0]								00000000
23H	AIENB1	7	AIENB[7:6]		AIENB[5:0]						00000000
24H	PT2	7	PT2 [7:0]								uuuuuuuu
25H	PT2EN	7	PT2EN [7:0]								00000000
26H	PT2PU	7	PT2PU [7:0]								00000000
27H	PT2MR	7.2/7.5/8	BZEN			PM1EN	E1M[1:0]		E0M[1:0]		00000000
28H	PT3						PT3 [3:0]				uuuuuuuu
29H	PT3EN						PT3 [3:0]				00000000
2AH	PT3PU						PT3 [3:0]				00000000
2BH	PT3MR					PFOEN	E3M[1:0]		E2M[1:0]		00000000
37H	PT2OCB	9				PT2OC[4:3]					uuu11uuu

The GPIO (General Purpose Input Output) in a micro-controller is used for general purpose input or output function. Users could use these ports to get digital signal or transmit data to any other digital device. Some GPIOs in FS98O25 are also defined for other special functions. In this Chapter, the GPIO will be illustrated as the GPIO function. The special functions defined in the GPIO will be illustrated in the following Chapters.

Register PT1 at address 20H

property	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
PT1	**PT1 [7:0]**							
	Bit7							Bit0

Bit 7-0 **PT1[7:0]**: GPIO Port 1 data flag (Please refer to Section 7.1 for detail)

PT1[7] = GPIO Port 1 bit 7 data flag
PT1[6] = GPIO Port 1 bit 6 data flag
PT1[5] = GPIO Port 1 bit 5 data flag
PT1[4] = GPIO Port 1 bit 4 data flag
PT1[3] = GPIO Port 1 bit 3 data flag
PT1[2] = GPIO Port 1 bit 2 data flag
PT1[1] = GPIO Port 1 bit 1 data flag
PT1[0] = GPIO Port 1 bit 0 data flag

Register PT1EN at address 21H

property	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT1EN	**PT1EN [7:0]**							
	Bit7							Bit0

Bit 7-0 **PT1EN [7:0]**: GPIO Port 1 Input / Output control flag (Please refer to Section 7.1 for detail)

PT1EN[7] = GPIO Port 1 bit 7 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT1EN[6] = GPIO Port 1 bit 6 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT1EN[5] = GPIO Port 1 bit 5 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT1EN[4] = GPIO Port 1 bit 4 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT1EN[3] = GPIO Port 1 bit 3 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT1EN[2] = GPIO Port 1 bit 2 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT1EN[1] = GPIO Port 1 bit 1 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT1EN[0] = GPIO Port 1 bit 0 I/O control flag ; 0 = defined as input port, 1 = defined as output port

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register PT1PU at address 22H

property	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT1PU	**PT1PU [7:0]**							
	Bit7							Bit0

Bit 7-0 **PT1PU [7:0]**: GPIO Port 1 Pull up resistor enable flag (Please refer to Section 7.1 for detail)

PT1EN[7] = GPIO Port 1 bit 7 control flag ; 0 = Pull up resistor is disconnected, 1 = with Pull up resistor

PT1EN[6] = GPIO Port 1 bit 6 control flag ; 0 = Pull up resistor is disconnected, 1 = with Pull up resistor

PT1EN[5] = GPIO Port 1 bit 5 control flag ; 0 = Pull up resistor is disconnected, 1 = with Pull up resistor

PT1EN[4] = GPIO Port 1 bit 4 control flag ; 0 = Pull up resistor is disconnected, 1 = with Pull up resistor

PT1EN[3] = GPIO Port 1 bit 3 control flag ; 0 = Pull up resistor is disconnected, 1 = with Pull up resistor

PT1EN[2] = GPIO Port 1 bit 2 control flag ; 0 = Pull up resistor is disconnected, 1 = with Pull up resistor

PT1EN[1] = GPIO Port 1 bit 1 control flag ; 0 = Pull up resistor is disconnected, 1 = with Pull up resistor

PT1EN[0] = GPIO Port 1 bit 0 control flag ; 0 = Pull up resistor is disconnected, 1 = with Pull up resistor

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

Register AIENB1 at address 23H

property	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
AIENB1	**AIENB[7:0]**							
	Bit7							Bit0

Bit 7-0 **AIENB[7:0]**: GPIO Port 1 Analog / Digital control flag (Please refer to Section 7.1 for detail)

AIENB[7] = GPIO Port 1 bit 7 D/A flag ; 0 = defined as Analog channel, 1 = defined as Digital channel

AIENB[6] = GPIO Port 1 bit 6 D/A flag ; 0 = defined as Analog channel, 1 = defined as Digital channel

AIENB[5] = GPIO Port 1 bit 5 D/A flag ; 0 = defined as Analog channel, 1 = defined as Digital channel

AIENB[4] = GPIO Port 1 bit 4 D/A flag ; 0 = defined as Analog channel, 1 = defined as Digital channel

AIENB[3] = GPIO Port 1 bit 3 D/A flag ; 0 = defined as Analog channel, 1 = defined as Digital channel

AIENB[2] = GPIO Port 1 bit 2 D/A flag ; 0 = defined as Analog channel, 1 = defined as Digital channel

AIENB[1] = GPIO Port 1 bit 1 D/A flag ; 0 = defined as Analog channel, 1 = defined as Digital channel

AIENB[0] = GPIO Port 1 bit 0 D/A flag ; 0 = defined as Analog channel, 1 = defined as Digital channel

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

Register PT2 at address 24H

property	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
PT2	**PT2 [7:0]**							
	Bit7							Bit0

Bit 7-0 **PT2[7:0]**: GPIO Port 2 data flag

PT2[7] = GPIO Port 2 bit 7 data flag
PT2[6] = GPIO Port 2 bit 6 data flag
PT2[5] = GPIO Port 2 bit 5 data flag
PT2[4] = GPIO Port 2 bit 4 data flag
PT2[3] = GPIO Port 2 bit 3 data flag
PT2[2] = GPIO Port 2 bit 2 data flag
PT2[1] = GPIO Port 2 bit 1 data flag
PT2[0] = GPIO Port 2 bit 0 data flag

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

Register PT2EN at address 25H

property	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT2EN	**PT2EN [7:0]**							
	Bit7							Bit0

Bit 7-0 **PT2EN [7 :0]** : GPIO Port 2 Input / Output control flag

PT2EN[7] = GPIO Port 2 bit 7 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT2EN[6] = GPIO Port 2 bit 6 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT2EN[5] = GPIO Port 2 bit 5 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT2EN[4] = GPIO Port 2 bit 4 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT2EN[3] = GPIO Port 2 bit 3 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT2EN[2] = GPIO Port 2 bit 2 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT2EN[1] = GPIO Port 2 bit 1 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT2EN[0] = GPIO Port 2 bit 0 I/O control flag ; 0 = defined as input port, 1 = defined as output port

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

Register PT2PU at address 26H

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT2PU	**PT2PU [7:0]**							
	Bit7							Bit0

Bit 7-0 **PT2PU [7:0]**: GPIO Port 2 Pull up resistor enable flag

PT2PU[7] = GPIO Port 2 bit 7 control flag ; 0 = Pull up resistor is disconnect, 1 = with Pull up resistor

PT2PU[6] = GPIO Port 2 bit 6 control flag ; 0 = Pull up resistor is disconnect, 1 = with Pull up resistor

PT2PU[5] = GPIO Port 2 bit 5 control flag ; 0 = Pull up resistor is disconnect, 1 = with Pull up resistor

PT2PU[4] = GPIO Port 2 bit 4 control flag ; 0 = Pull up resistor is disconnect, 1 = with Pull up resistor

PT2PU[3] = GPIO Port 2 bit 3 control flag ; 0 = Pull up resistor is disconnect, 1 = with Pull up resistor

PT2PU[2] = GPIO Port 2 bit 2 control flag ; 0 = Pull up resistor is disconnect, 1 = with Pull up resistor

PT2PU[1] = GPIO Port 2 bit 1 control flag ; 0 = Pull up resistor is disconnect, 1 = with Pull up resistor

PT2PU[0] = GPIO Port 2 bit 0 control flag ; 0 = Pull up resistor is disconnect, 1 = with Pull up resistor

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register PT2MR at address 27H

property	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT2MR	**BZEN**			**PM1EN**	**E1M[1:0]**		**E0M[1:0]**	
	Bit7							Bit0

Bit 7 **BZEN**: Buzzer enable flag (Please refer to Section 7.5 for detail)
1 = Buzzer function is enabled, GPIO Port 2 bit 7 is defined as Buzzer output.
0 = Buzzer function is disabled, GPIO Port 2 bit 7 is defined as GPIO.

Bit 4 **PM1EN**: PDM Module enable flag (Please refer to Chapter 8 for detail)
1 = PDM Module is enabled, GPIO Port 2 bit 2 is defined as PDM output.
0 = PDM Module is disabled, GPIO Port 2 bit 2 is defined as GPIO.

Bit 3-2 **E1M[1:0]**: GPIO Port 2 bit 1 interrupt trigger mode (Please refer to Section 7.2 for detail)
11 = External Interrupt 1 (GPIO Port 2 bit 1) is triggered at state change
10 = External Interrupt 1 (GPIO Port 2 bit 1) is triggered at state change
01 = External Interrupt 1 (GPIO Port 2 bit 1) is triggered at positive edge
00 = External Interrupt 1 (GPIO Port 2 bit 1) is triggered at negative edge

Bit 1-0 **E0M[1:0]**: GPIO Port 2 bit 0 interrupt trigger mode (Please refer to Section 7.2 for detail)
11 = External Interrupt 0 (GPIO Port 2 bit 0) is triggered at state change
10 = External Interrupt 0 (GPIO Port 2 bit 0) is triggered at state change
01 = External Interrupt 0 (GPIO Port 2 bit 0) is triggered at positive edge
00 = External Interrupt 0 (GPIO Port 2 bit 0) is triggered at negative edge

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register PT3 at address 28H

property	U-X	U-X	U-X	U-X	R/W-X	R/W-X	R/W-X	R/W-X
PT3					**PT3 [3:0]**			
	Bit7							Bit0

Bit 3-0 **PT3[3:0]**: GPIO Port 3 data flag

PT3[3] = GPIO Port 3 bit 3 data flag

PT3[2] = GPIO Port 3 bit 2 data flag

PT3[1] = GPIO Port 3 bit 1 data flag

PT3[0] = GPIO Port 3 bit 0 data flag

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register PT3EN at address 29H

property	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PT3EN					**PT3EN [3:0]**			
	Bit7							Bit0

Bit 3-0 **PT3EN [3 :0]** : GPIO Port 3 Input / Output control flag

PT3EN[3] = GPIO Port 3 bit 3 I/O control flag ; 0 = defined as input port, 1 = defined as output port

PT3EN[2] = GPIO Port 3 bit 2 I/O control flag ; 0 = defined as input port, 1 = defined as output port

PT3EN[1] = GPIO Port 3 bit 1 I/O control flag ; 0 = defined as input port, 1 = defined as output port

PT3EN[0] = GPIO Port 3 bit 0 I/O control flag ; 0 = defined as input port, 1 = defined as output port

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register PT3PU at address 2AH

	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PT3PU					**PT3PU [3:0]**			
	Bit7							Bit0

Bit 3-0 **PT3PU [3:0]**: GPIO Port 3 Pull up resistor enable flag

PT3PU[3] = GPIO Port 3 bit 3 control flag ; 0 = Pull up resistor is disconnected, 1 = with Pull up resistor

PT3PU[2] = GPIO Port 3 bit 2 control flag ; 0 = Pull up resistor is disconnected, 1 = with Pull up resistor

PT3PU[1] = GPIO Port 3 bit 1 control flag ; 0 = Pull up resistor is disconnected, 1 = with Pull up resistor

PT3PU[0] = GPIO Port 3 bit 0 control flag ; 0 = Pull up resistor is disconnected, 1 = with Pull up resistor

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register PT3MR at address 2BH

property	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT3MR				**PFOEN**	**E3M[1:0]**		**E2M[1:0]**	
	Bit7							Bit0

Bit 4 **PFOEN**: Programmable Counter Enabled register flag
1 = Programmable Counter is enabled
0 = Programmable Counter is disabled

Bit 3-2 **E3M[1:0]**: GPIO Port 3 bit 1 interrupt trigger mode (Please refer to Section 7.2 for detail)
11 = External Interrupt 4 (GPIO Port 3 bit 1) is triggered at state change
10 = External Interrupt 4 (GPIO Port 3 bit 1) is triggered at state change
01 = External Interrupt 4 (GPIO Port 3 bit 1) is triggered at positive edge
00 = External Interrupt 4 (GPIO Port 3 bit 1) is triggered at negative edge

Bit 1-0 **E2M[1:0]**: GPIO Port 3 bit 0 interrupt trigger mode (Please refer to Section 7.2 for detail)
11 = External Interrupt 3 (GPIO Port 3 bit 0) is triggered at state change
10 = External Interrupt 3 (GPIO Port 3 bit 0) is triggered at state change
01 = External Interrupt 3 (GPIO Port 3 bit 0) is triggered at positive edge
00 = External Interrupt 3 (GPIO Port 3 bit 0) is triggered at negative edge

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register PT2OCB at address 37H

property	U-X	U-X	U-X	R/W-1	R/W-1	U-X	U-X	U-X
PT2OCB				**PT2OC[4:3]**				
	Bit7							Bit0

Bit 4-3 **PT2OC[4:3]**: GPIO Port 2 Open Drain control flag

PT2OC[4] = GPIO Port 2 bit 4 Open Drain control flag ; 0 = normal digital I/O, 1 = Open Drain Control

PT2OC[3] = GPIO Port 2 bit 3 Open Drain control flag ; 0 = normal digital I/O, 1 = Open Drain Control

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

7.1. Digital I/O Port with Analog Input Channel Shared: PT1[7:0]

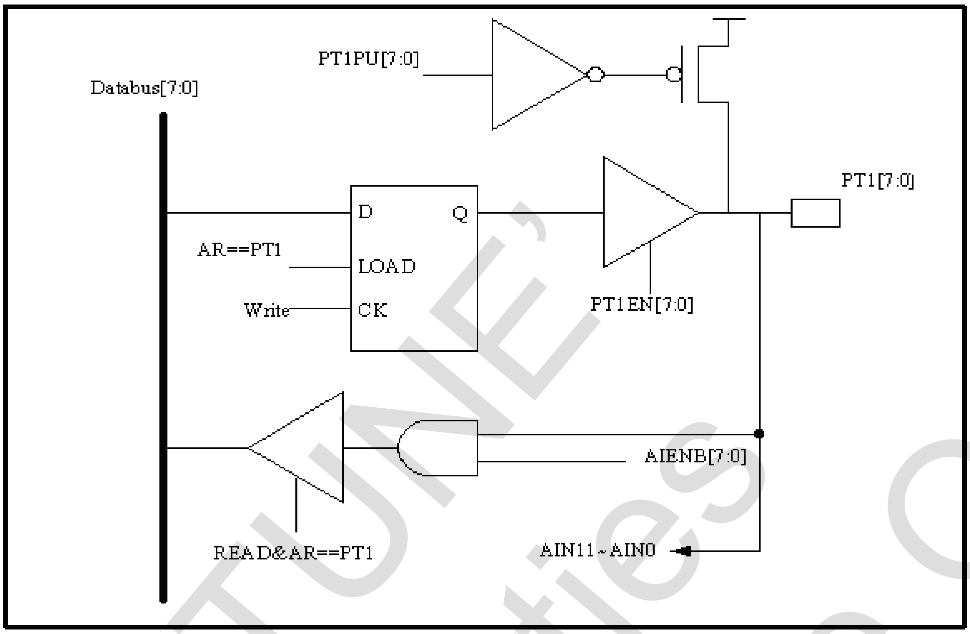


Figure 7-1 PT1[7:0] function block

GPIO Port 1 (PT1[7:0]) function block is shown in Figure 7-1. The main function of the GPIO is for data exchange between the Data bus and the ports. Users could control the PT1EN[7:0] register flags to decide the input and output direction. The input and output function and the related functions are explained as follows:

- Input:

  GPIO Port 1 Bit0 to Bit7 (PT1[7:0]) could be used to get both the digital signal and the analog signal. User should control the AIENB[11:0] register flags to decide the input type. If user sets the AIENB, the AND gate embedded in the GPIO Port1 will allow the digital data to connect to the data bus. Otherwise, the Input signals will be defined as analog signals and sent to the related function blocks (ADC, OPAMP…etc)

- Output

  FS98O25 sends the digital data out by an embedded D Flip Flop. When the program sends data out through PT1 , the data will be sent to data bus first, and then the D Flip Flop will latch the signal for PT1 and output while the Write signal and AR (FS98O25 internal device address pointer) is pointed to PT1 and .

- Pull up resistor

  FS98O25 embeds an internal pull up resistor function in PT1 with about 1000k ohm resistor[9]. Users could control the PT1PU[7:0] register flags to decide the connections to pull up resistor. When a port is connected to the pull up resistor, the input data is, by default, assigned to high (data 1).

[9] The pull up current is about 10uA. Remember to disable PT1PU before program falls into Sleep mode.

Table 7-2 PT1 register table

Address	Name	Detail on Chapter	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
20H	PT1	7	PT1 [7:0]								uuuuuuuu
21H	PT1EN	7	PT1EN [7:0]								00000000
22H	PT1PU	7	PT1PU [7:0]								00000000
23H	AIENB1	7	AIENB[7:6]		AIENB[5:0]						00000000

Read data Operation

1. Clear the PT1EN[n][10] register flags. The PT1[n] will be defined as an input port.
2. Set the PT1PU[n] register as required. The PT1[n] will be connected to an internal pull up resistor.
3. Set the AIENB[n] register flags if the input signals are analog signals.(n = 11 to 0)
4. Clear the AIENB[n] register flags if the input signals are analog signals. (n = 11 to 0[11])
5. The VDDA Regulator must be enabled first, and then the AIN0~AIN7 can work correctly. (Please refer to Chapter 4)
6. After the signal input from outside, users can get the data through PT1[n]

Write data Operation

1. Set the PT1EN[n] register flags. The PT1[n] will be defined as an output port.
2. Set the PT1PU[n] register as required. The PT1[n]] will be connected to an internal pull up resistor.
3. Set the PT1[n] to output the data. The embedded D Flip Flop will latch the data till PT1[n] is changed.

Notice Operation

1. To keep low operation current in SLEEP mode, set AIENB[11:0] to let the PT1 be floating.
2. Parallel a small resistor (about 10k ohm) between ports and VDD to increase the possible output current when the PT1PU[n] is set.

7.2. Digital I/O Port and External Interrupt Input : PT2[0], PT2[1], PT3[0], PT3[1]

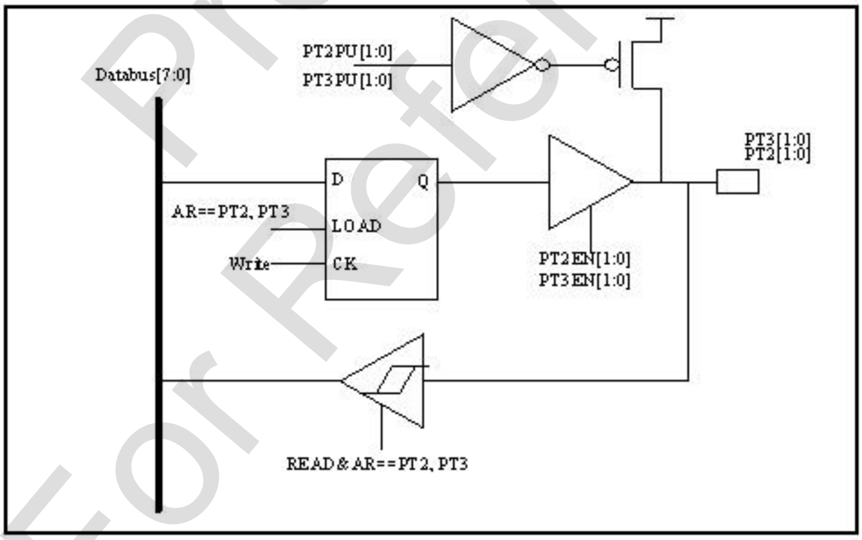


Figure 7-2 PT2[0] PT2[1] PT3[0] PT3[1] function block

[10] n means the bits indexes user want to control
[11] PT1 bit6 and bit7 could only be defined as digital signal input.

GPIO Port 2 Bit1 and Bit 0 (PT2[1:0])and Port 3 Bit1 and Bit 0 (PT3[1:0]) function block is shown in Figure 7-2. The main function of the GPIO is input and output data between the Data bus and the ports. Users could control the PT2EN[1:0] and PT3EN[1:0] register flags to decide the input output direction. The input and output function and the related functions are explained as follows:

- Input:

  GPIO Port 2 Bit1 and Bit0 (PT2[1:0]) could be the external interrupt ports as INT1 and INT0 or be the general I/O ports. User should control INTE register E0IE and E1IE flags to decide if the interrupt is enabled. The interrupt trigger mode is selected by E0M[1:0] and E1M[1:0] register flags. The input port has a Schmitt trigger in it, and the up/down trigger voltage level is 0.45VDD/0.2VDD.

  GPIO Port 3 Bit1 and Bit0 (PT3[1:0]) could be the external interrupt ports as INT3 and INT2 or be the general I/O ports. User should control INTE register E2IE and E3IE flags to decide if the interrupt is enabled. The interrupt trigger mode is selected by E2M[1:0] and E3M[1:0] register flags. The input port has a Schmitt trigger in it, and the up/down trigger voltage level is 0.45VDD/0.2VDD.

- Output

  FS98O25 sends the digital data out by an embedded D Flip Flop. When the program sends data out through PT2 or PT3, the data will be sent to data bus first, and then the D Flip Flop will latch the signal for PT2/PT3 output while the Write signal and AR (FS98O25 internal device address pointer) is pointed to PT2/PT3.

- Pull up resistor

  FS98O25 embeds an internal pull up resistor function in PT2 with about 1000k ohm resistor[12]. Users could control the PT2PU[1:0] register flags to decide the connections to pull up resistor. When a port is connected to the pull up resistor, the input data is, by default, assigned to high (data 1).

[12] The pull up current is about 10uA. Remember to disable PT1PU before program falls into Sleep mode.

Table 7-3 PT2 register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
06H	INTF	3/6/7/9/10/11				--	--	-	E1IF	E0IF	00000000
07H	INTE	3/6/7/9/10/11	GIE			--	--	-	E1IE	E0IE	00000000
24H	PT2	7	PT2 [7:0]								uuuuuuuu
25H	PT2EN	7	PT2EN [7:0]								00000000
26H	PT2PU	7	PT2PU [7:0]								00000000
27H	PT2MR	7.2/7.5/8	--			--	E1M[1:0]		E0M[1:0]		00000000
28H	PT3						PT3 [3:0]				uuuuuuuu
29H	PT3EN						PT3EN [3:0]				00000000
2AH	PT3PU						PT3PU [3:0]				00000000
2BH	PT3MR					PFOEN	E3M[1:0]		E2M[1:0]		00000000

Read data Operation

1. Clear the PT2EN[n][13]/PT3EN[n] register flags. The PT2[n]/PT3[n] will be defined as an input port.
2. Set the PT2PU[n]/PT3PU[n] register as required. The PT2[n]/PT3[n] will be connected to an internal pull up resistor.
3. After the signal input from outside, user could get the data through PT2[n]/PT3[n]

Write data Operation

1. Set the PT2EN[n]/PT3EN[n] register flags. The PT2[n]/PT3[n] will be defined as an output port.
2. Set the PT2PU[n]/PT3PU[n] register as required. The PT2[n]/PT3[n] will be connected to an internal pull up resistor.
3. Set the PT2[n]/PT3[n] to output the data. The embedded D Flip Flop will latch the data till PT2[n]/PT3[n] is changed.

External Interrupt Operation (negative edge trigger for example)

1. Clear the PT2EN[n]/PT3EN[n] register flags. The PT2[n]/PT3[n] will be defined as an input port.
2. Set the PT2PU[n]/PT3PU[n] register. The PT2[n]/PT3[n] will be connected to an internal pull up resistor.
3. Set the E0M[1:0] as 00 to define INT0 interrupt trigger mode as “negative edge trigger”.
4. Set the E1M[1:0] as 00 to define INT1 interrupt trigger mode as “negative edge trigger”.
5. Set the E2M[1:0] as 00 to define INT2 interrupt trigger mode as “negative edge trigger”.
6. Set the E3M[1:0] as 00 to define INT3 interrupt trigger mode as “negative edge trigger”.

Notice Operation

1. Parallel a small resistor (about 10k ohm) between ports and VDD to increase the possible output current when the PT2PU[n]/PT3PU[n] is set.

[13] n means the bits indexes user want to control

7.3. Digital I/O Port or PDM Output : PT2[2] and PT2[5]

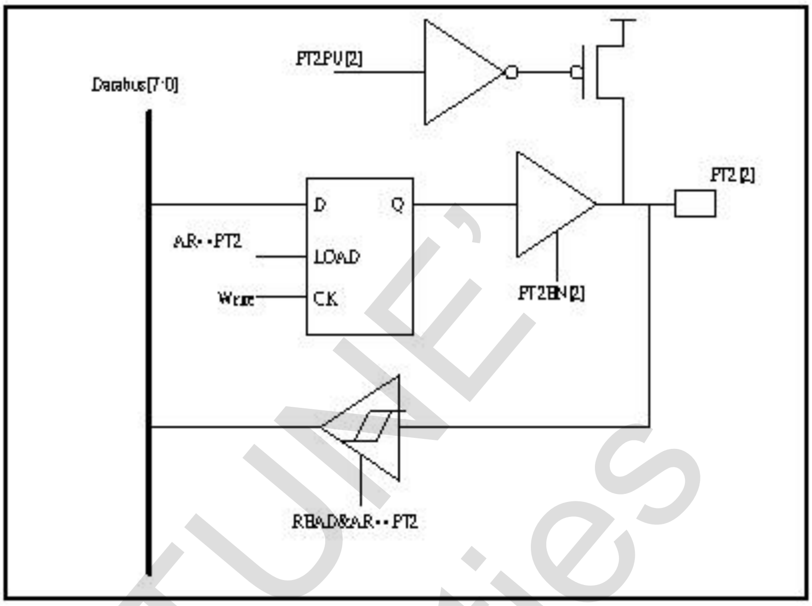


Figure 7-3 PT2[2] function block

GPIO Port 2 Bit2 (PT2[2]) and GPIO Port 2 Bit5 (PT2[5]) function block is shown in Figure 7-3. The main function of the GPIO is input and output data between the Data bus and the ports. User could control the PT2EN[2]/ PT2EN[5] register flags to decide the input output direction. The input and output function and the related functions are explained as follows:

- Input:

  GPIO Port 2 Bit2 (PT1[2]) and GPIO Port 2 Bit5 (PT1[5]) could be the PDM (Pulse Density Modulator) output port or be the general I/O port. User should setup PM1EN/ PM2EN register flag to decide if the PDM is enabled. The detailed PDM usage is described in Chapter 8.

  The input port has a Schmitt trigger in it, and the up/down trigger voltage level is 0.45VDD/0.2VDD.

- Output

  FS98O25 sends the digital data out by an embedded D Flip Flop. When the program sends data out through PT2, the data will be sent to data bus first, and then the D Flip Flop will latch the signal for PT2 output while the Write signal and AR (FS98O25 internal device address pointer) is pointed to PT2.

- Pull up resistor

  FS98O25 embeds an internal pull up resistor function in PT2 with about 1000k ohm resistor[14]. Users could control the PT2PU[2]/PT2PU[5] register flags to decide the connection to pull up resistor. When a port is connected to the pull up resistor, the input data is, by default, assigned to high (data 1).

[14] The pull up current is about 10uA. Remember to disable PT1PU before program falls into Sleep mode.

Table 7-4 PT2 register table

Address	Name	Detail on Chapter	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
24H	PT2	7	PT2 [7:0]								uuuuuuuu
25H	PT2EN	7	PT2EN [7:0]								00000000
26H	PT2PU	7	PT2PU [7:0]								00000000
27H	PT2MR	7.2/7.5/8	--	PM2EN		PM1EN	--		--		00000000

Read data Operation

1. Clear the PT2EN[n][15] register flags. The PT2[n] will be defined as an input port.
2. Set the PT2PU[n] register as required. The PT2[n] will be connected to an internal pull up resistor.
3. After the signal input from outside, user could get the data through PT2[n]

Write data Operation

1. Set the PT2EN[n] register flags. The PT2[n] will be defined as an output port.
2. Set the PT2PU[n] register as required. The PT2[n] will be connected to an internal pull up resistor.
3. Set the PT2[n] to output the data. The embedded D Flip Flop will latch the data till user change PT2[n].

Notice Operation

1. Parallel a small resistor (about 10k ohm) between ports and VDD to increase the possible output current when the PT2PU[n] is set.

7.4. Digital I/O Port or I2C Serial Port : PT2[3]/SDA, PT2[4]/SCL

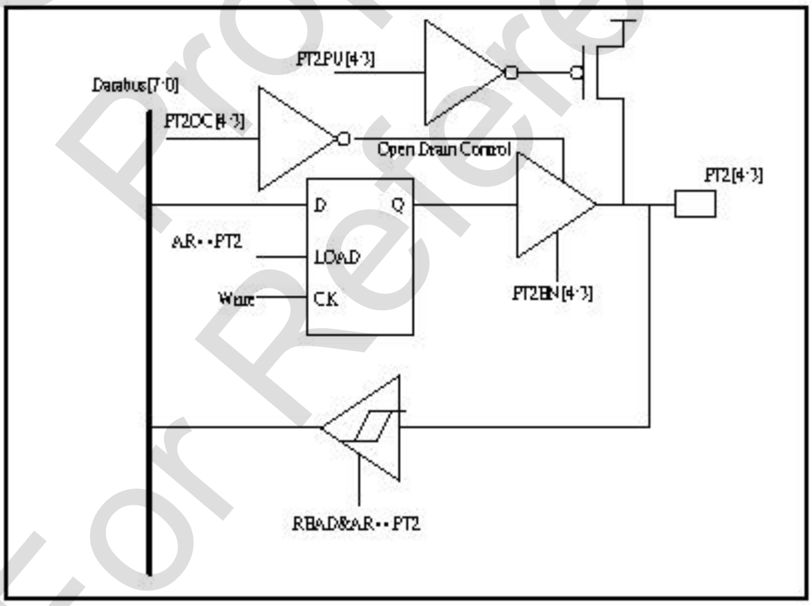


Figure 7-4 PT2[3] PT2[4] function block

[15] n means the bits indexes user want to control

GPIO Port 2 Bit4 and Bit 3 (PT2[4:3]) function block is shown in Figure 7-4. The main function of the GPIO is input and output data between the Data bus and the ports. Users could control the PT2EN[4:3] register flags to decide the input output direction. The input and output function and the related functions are explained as follows:

- Input:

  GPIO Port 2 Bit4 and Bit3 (PT2[4:3]) could be the I2C Module SCL and SDA ports or be the general I/O ports. User should setup I2CEN register flag to decide the I2C Module is enabled or not. The detailed I2C Module usage is described in Chapter 9.

  The input port has a Schmitt trigger in it, and the up/down trigger voltage level is 0.45VDD/0.2VDD.

- Output

  FS98O25 sends the digital data out by an embedded D Flip Flop. When the program sends data out through PT2, the data will be sent to data bus first, and then the D Flip Flop will latch the signal for PT2 output while the Write signal and AR (FS98O25 internal device address pointer) is pointed to PT2.

- Pull up resistor

  FS98O25 embeds an internal pull up resistor function in PT2 with about 1000k ohm resistor[16]. User could control the PT2PU[4:3] register flags to decide the connections to pull up resistor. When a port is connected to the pull up resistor, the input data is, by default, assigned to high (data 1).

- Open Drain Control

  FS98O25 embeds an internal Open Drain Control function in PT2[4:3]. Users could control the PT2OC[4:3] register flags to decide if the Open Drain Control function is enabled. When the user assigns these 2 ports to be SCL and SDA, PT2OC[4:3] should be set. Please refer to Chapter 9.

Table 7-5 PT2 register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
24H	PT2	7	PT2 [7:0]								uuuuuuuu
25H	PT2EN	7	PT2EN [7:0]								00000000
26H	PT2PU	7	PT2PU [7:0]								00000000
37H	PT2OCB	9				PT2OC[4:3]					uuu11uuu

Read data Operation

1. Clear the PT2EN[n][17] register flags. The PT2[n] will be defined as an input port.
2. Set the PT2PU[n] register as required. The PT2[n] will be connected to an internal pull up resistor.
3. Set the PT2OC[n] register as required. The PT2[n] will be connected to an internal pull low resistor.
4. After the signal input from outside, user could get the data through PT2[n]

Write data Operation

1. Set the PT2EN[n] register flags. The PT2[n] will be defined as an output port.
2. Set the PT2PU[n] register as required. The PT2[n] will be connected to an internal pull up resistor.
3. Set the PT2OC[n] register as required. The PT2[n] will be connected to an internal pull low resistor.
4. Set the PT2[n] to output the data. The embedded D Flip Flop will latch the data till PT2[n] is changed.

16 The pull up current is about 10uA. Remember to disable PT1PU before program falls into Sleep mode.
17 n means the bit index that a user want to control

Notice Operation

1. Parallel a small resistor (about 10k ohm) between ports and VDD to enlarge the possible output current when the PT2PU[n] is set.
2. The Pull up resistor function and the Open drain control function should NOT be enabled at the same time.

7.5. Digital I/O Port : PT2[6]

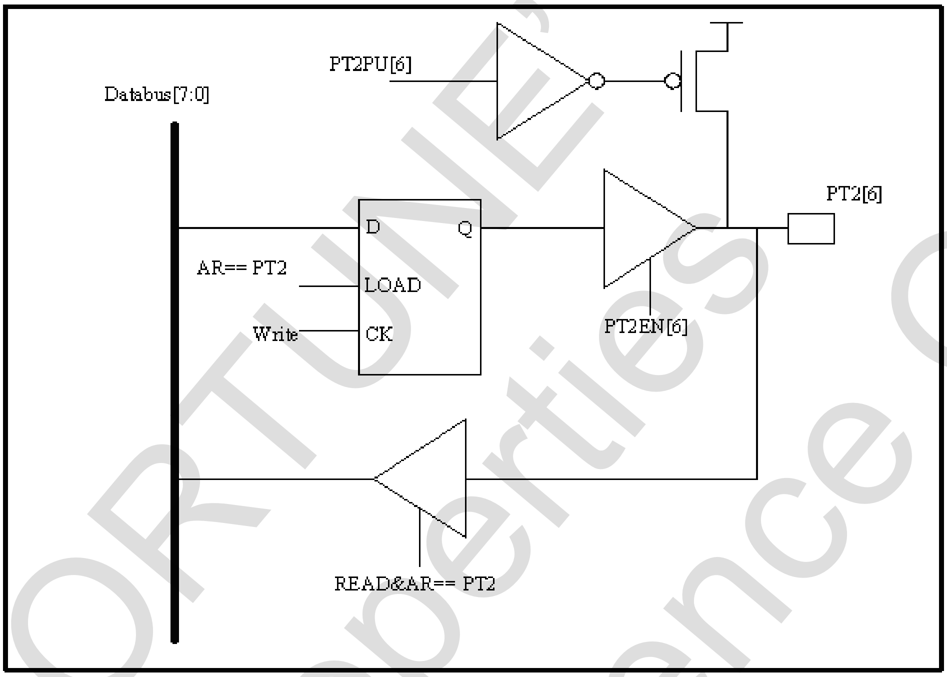


Figure 7-5 PT2[6] function block

GPIO Port 2 Bit 6 (PT2[6]) is shown in Figure 7-5. The main function of the GPIO is input and output data between the Data bus and the ports. Users could control the PT2EN[6] register flags to decide the input output direction. The input and output function are explained as follows:

- Input:

  GPIO Port 2 Bit 6 (PT2[6]) could only be the general I/O ports. The input port has a Schmitt trigger in it, and the up/down trigger voltage level is 0.45VDD/0.2VDD.

- Output

  FS98O25 sends the digital data out by an embedded D Flip Flop. When the program sends data out through PT2, the data will be sent to data bus first, and then the D Flip Flop will latch the signal for PT2 output while the Write signal and AR (FS98O25 internal device address pointer) is pointed to PT2.

- Pull up resistor

FS98O25 embeds an internal pull up resistor function in PT2 with about 1000k ohm resistor[18]. User could control the PT2PU[6] register flags to set the connections to pull up resistor. When a port is connected to the pull up resistor, the input data is, by default, assigned to high (data 1).

[18] The pull up current is about 10uA. Remember to disable PT1PU before program falls into Sleep mode.

Table 7-6 PT2 register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
24H	PT2	7	PT2 [7:0]								uuuuuuuu
25H	PT2EN	7	PT2EN [7:0]								00000000
26H	PT2PU	7	PT2PU [7:0]								00000000
27H	PT2MR	7.2/7.5/8	BZEN			--	--		--		00000000

Read data Operation

1. Clear the PT2EN[n][19] register flags. The PT2[n] will be defined as an input port.
2. Set the PT2PU[n] register as required. The PT2[n] will be connected to an internal pull up resistor.
3. After the signal input from outside, user could get the data through PT2[n]

Write data Operation

1. Set the PT2EN[n] register flags. The PT2[n] will be defined as an output port.
2. Set the PT2PU[n] register as required. The PT2[n] will be connected to an internal pull up resistor.
3. Set the PT2[n] to output the data. The embedded D Flip Flop will latch the data till PT2[n] is changed.

Notice Operation

1. Parallel a small resistor (about 10k ohm) between ports and VDD to increase the possible output current when the PT2PU[n] is set.

[19] n means the bits indexes user want to control

7.6. Digital I/O Port or Buzzer Output : PT2[7]

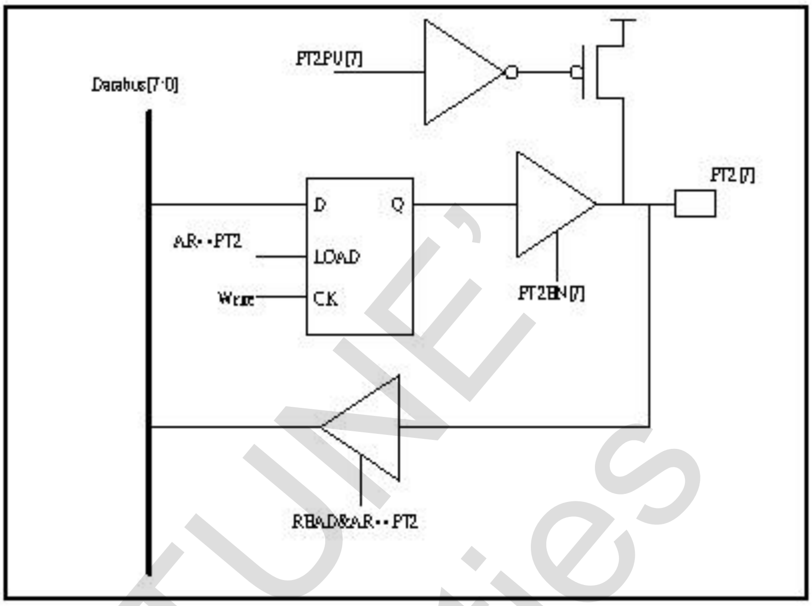


Figure 7-6 PT2[7] function block

GPIO Port 2 Bit2 (PT2[2]) function block is shown in Figure 7-6. The main function of the GPIO is input and output data between the Data bus and the ports. Users could control the PT2EN[2] register flags to decide the input output direction. The input and output function and the related functions are explained as follows:

- Input:

  GPIO Port 2 Bit2 (PT1[2]) could be the Buzzer output port or be the general I/O port. User should setup BZEN register flag to decide if the Buzzer output is enabled. The detailed Buzzer usage is described in Section 5.4.

- Output

  FS98O25 sends the digital data out by an embedded D Flip Flop. When the program sends data out through PT2, the data will be sent to data bus first, and then the D Flip Flop will latch the signal for PT2 output while the Write signal and AR (FS98O25 internal device address pointer) is pointed to PT2.

- Pull up resistor

  FS98O25 embeds an internal pull up resistor function in PT2 with about 1000k ohm resistor[20]. User could control the PT2PU[2] register flags to set the connection to pull up resistor. When a port is connected to the pull up resistor, the input data is, by default, assigned to high (data 1).

[20] The pull up current is about 10uA. Remember to disable PT1PU before program falls into Sleep mode.

Table 7-7 PT2[7] register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
24H	PT2	7	PT2 [7:0]								uuuuuuuu
25H	PT2EN	7	PT2EN [7:0]								00000000
26H	PT2PU	7	PT2PU [7:0]								00000000

Read data Operation

1. Clear the PT2EN[n][21] register flags. The PT2[n] will be defined as an input port.
2. Set the PT2PU[n] register as required. The PT2[n] will be connected to an internal pull up resistor.
3. After the signal input from outside, user could get the data through PT2[n]

Write data Operation

1. Set the PT2EN[n] register flags. The PT2[n] will be defined as an output port.
2. Set the PT2PU[n] register as required. The PT2[n] will be connected to an internal pull up resistor.
3. Set the PT2[n] to output the data. The embedded D Flip Flop will latch the data till PT2[n] is changed.

Buzzer Output Operation

1. Set the PT2EN[7] register flags. The PT2[7] will be defined as an output port.
2. Please refer to Section 5.4 for the Buzzer Clock setting.
3. Set the BZEN register flag. The PT2[7] will become the buzzer output port.
4. Connect a buzzer to PT2 bit7. The Buzzer will work correctly.

Notice Operation

1. Parallel a small resistor (about 10k ohm) between ports and VDD to increase the possible output current when the PT2PU[n] is set.

[21] n means the bits indexes user want to control

8. PDM (Pulse Density Modulator) Module

Please see Figure 8-1 and Figure 8-2. The GPIO port 2 bit 2 (PT2[2]) could be defined as either PDM module output or General purpose I/O. User could control the PDMEN register flags to decide the definition. The PDM module is the function FS98O25 uses for implementing the PWM (Pulse Width Modulation). Its working flowchart and usage will be described in this Chapter. First of all, a user needs to setup the PMCS register flag to decide the PDM CLK which is generated by a Frequency divider from the MCK[22]. Then, the PDM CLK will be divided into 16 internal clock signals named PDM15, PDM14,…, PDM0. Finally, the user should control the PMD1 (PMD1H and PMD1L) register flag to do the combination of these 16 internal clock signals. For example, if the PMD1 is set as 0x1228H, the output signal is assigned to be the combination of PDM12, PDM9, PDM5 and PDM3. If the PMD1 is set as 0x6000H, the output signal is assigned to be the combination of PDM14 and PDM13 (please refer to the following figure).The PMD1 value could be assigned from 0 to 65535, and the output signal duty cycle could be from 0 to 65535/65536[23]. For example, when user sets the PMD1 as 0x6000H (24576), the equivalent PWM duty cycle is 24576/65536.

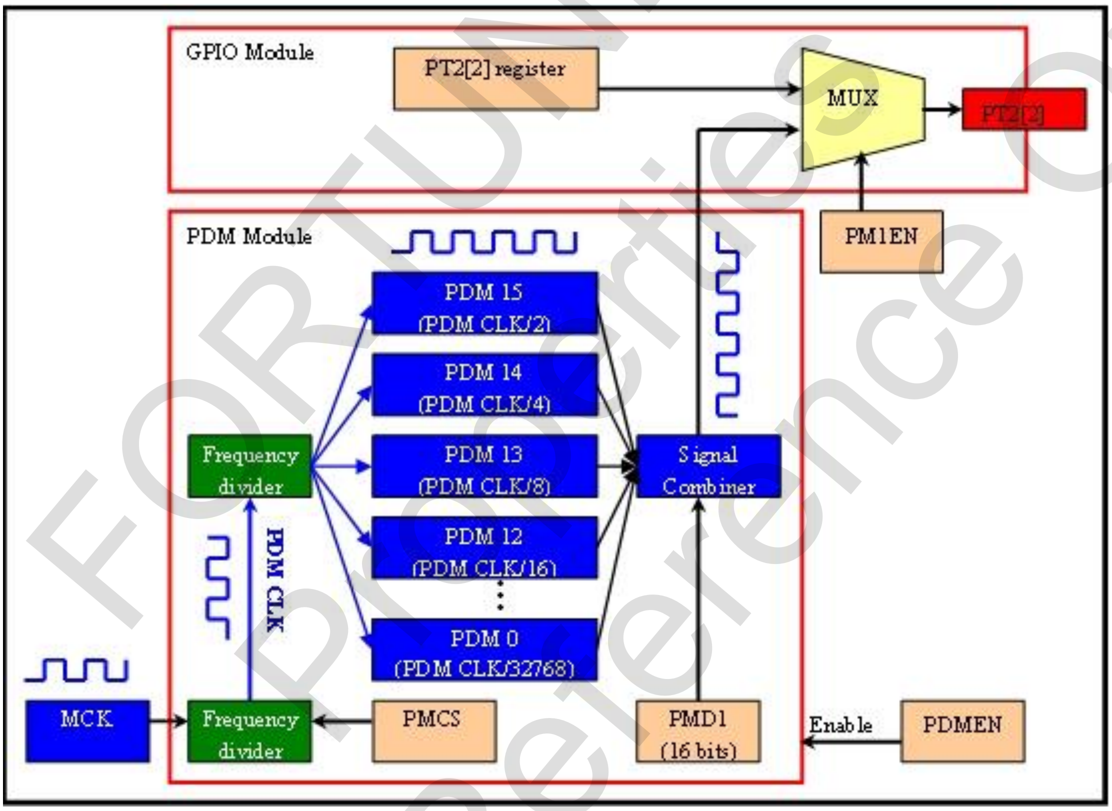


Figure 8-1 FS98O25 PDM module function block

[22] Please refer to Chapter 5 for MCK detailed information.
[23] The PDM couldn't generate signal as duty cycle 1, user needs to define the port as General purpose I/O and keep it at high voltage level (data 1) manually to represent Duty Cycle 1.

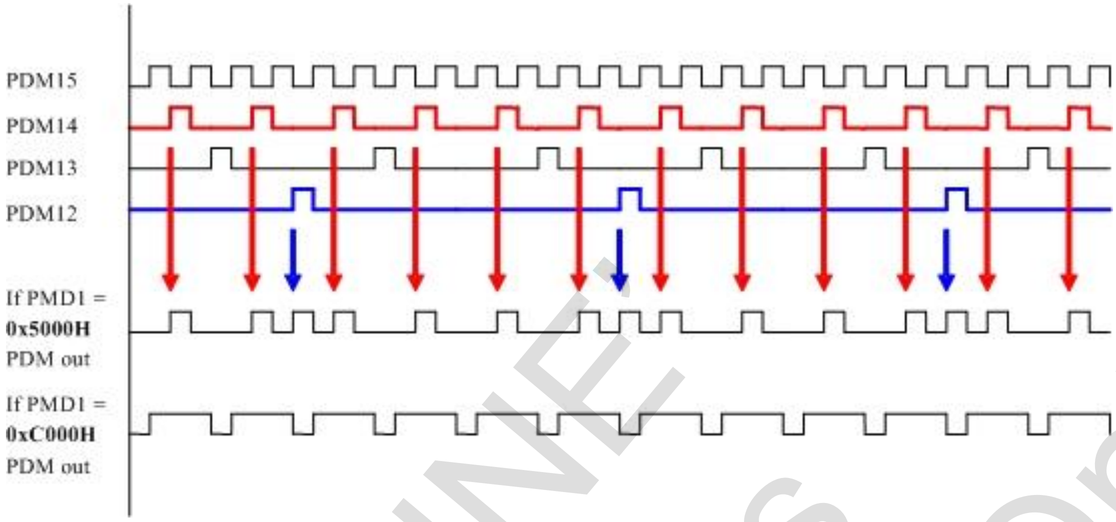


Figure 8-2 PDM module signal generation

Table 8-1 PDM module register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
27H	PT2MR	7.2/7.5/8	--	PM2EN		PM1EN	--		--		00000000
30H	PMD1H	8	PMD1[15:8]								00000000
31H	PMD1L	8	PMD1[7:0]								00000000
32H	PMD2H		PDMD2[15:8]								00000000
33H	PMD2L		PDMD2[7:0]								00000000
36H	PMCON	8				PDMEN		PMCS[2:0]			00000000

Register PT2MR at address 27H

property	U-0	R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0
PT2MR	**--**	**PM2EN**		**PM1EN**	**--**		**--**	
	Bit7							Bit0

Bit 6 **PM2EN**: PT2[5] output multiplexer (Please refer to Section 7.3 for details)

1 = GPIO Port 2 bit 5 (PT2[5]) is defined as PDM output.

0 = GPIO Port 2 bit 5 (PT2[5]) is defined as GPIO.

Bit 4 **PM1EN**: PT2[2] output multiplexer (Please refer to Section 7.3 for details)

1 = GPIO Port 2 bit 2 (PT2[2]) is defined as PDM output.

0 = GPIO Port 2 bit 2 (PT2[2]) is defined as GPIO.

Property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register PMD1H at address 30H

property	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMD1H	**PMD1[15:8]**							
	Bit7							Bit0

Register PMD1L at address 31H

property	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMD1L	**PMD1[7:0]**							
	Bit7							Bit0

Bit 15-0 **PMD1[15:0]**: PDM Module Data output Control Register

PMD1[15] = PDM15 (PDM CLK/2^{1})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD1[14] = PDM14 (PDM CLK/2^{2})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD1[13] = PDM13 (PDM CLK/2^{3})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD1[12] = PDM12 (PDM CLK/2^{4})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD1[11] = PDM11 (PDM CLK/2^{5})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD1[10] = PDM10 (PDM CLK/2^{6})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD1[9] = PDM9 (PDM CLK/2^{7})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD1[8] = PDM8 (PDM CLK/2^{8})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD1[7] = PDM7 (PDM CLK/2^{9})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD1[6] = PDM6 (PDM CLK/2^{10})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD1[5] = PDM5 (PDM CLK/2^{11})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD1[4] = PDM4 (PDM CLK/2^{12})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD1[3] = PDM3 (PDM CLK/2^{13})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD1[2] = PDM2 (PDM CLK/2^{14})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD1[1] = PDM1 (PDM CLK/2^{15})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD1[0] = PDM0 (PDM CLK/2^{16})Signal Combination enable flag. 0 = Enable ; 1 = Disable

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register PMD2H at address 32H

property	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMD2H	**PMD2[15:8]**							
	Bit7							Bit0

Register PMD2L at address 33H

property	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PMD2L	**PMD2[7:0]**							
	Bit7							Bit0

Bit 15-0 **PMD2[15:0]**: PDM Module Data output Control Register

PMD2[15] = PDM15 (PDM CLK/2^1)Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD2[14] = PDM14 (PDM CLK/2^2)Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD2[13] = PDM13 (PDM CLK/2^3)Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD2[12] = PDM12 (PDM CLK/2^4)Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD2[11] = PDM11 (PDM CLK/2^5)Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD2[10] = PDM10 (PDM CLK/2^6)Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD2[9] = PDM9 (PDM CLK/2^7)Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD2[8] = PDM8 (PDM CLK/2^8)Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD2[7] = PDM7 (PDM CLK/2^9)Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD2[6] = PDM6 (PDM CLK/2^{10})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD2[5] = PDM5 (PDM CLK/2^{11})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD2[4] = PDM4 (PDM CLK/2^{12})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD2[3] = PDM3 (PDM CLK/2^{13})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD2[2] = PDM2 (PDM CLK/2^{14})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD2[1] = PDM1 (PDM CLK/2^{15})Signal Combination enable flag. 0 = Enable ; 1 = Disable
PMD2[0] = PDM0 (PDM CLK/2^{16})Signal Combination enable flag. 0 = Enable ; 1 = Disable

Register PMCON at address 36H

property	U-0	U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
PMCON				**PDMEN**		**PMCS[2:0]**		
	Bit7							Bit0

Bit 4 **PDMEN**: PDM Module enable flag (Please refer to Chapter 8 for details)

1 = PDM Module is enabled, GPIO Port 2 bit 2 could be defined as PDM output.

0 = PDM Module is disabled, GPIO Port 2 bit 2 could be defined as GPIO.

Bit 2-0 **PMCS[2:0]**: PDM CLK frequency Selector

111 = PDM CLK frequency is as MCK/128

110 = PDM CLK frequency is as MCK/64

101 = PDM CLK frequency is as MCK/32

100 = PDM CLK frequency is as MCK/16

011 = PDM CLK frequency is as MCK/8

010 = PDM CLK frequency is as MCK/4

001 = PDM CLK frequency is as MCK/2

000 = PDM CLK frequency is the same as MCK

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

Table 8-2 PMD register table

Address	Name	Detail on Chapter	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
14H	MCK	5	M7_CK	M6_CK	M5_CK		M3_CK	M2_CK	M1_CK	M0_CK	00000000
25H	PT2EN	7	PT2EN [7:0]								00000000
27H	PT2MR	7.2/7.5/8	--			PM1EN	--		--		00000000
30H	PMD1H	8	PMD1[15:8]								00000000
31H	PMD1L	8	PMD1[7:0]								00000000
36H	PMCON	8				PDMEN		PMCS[2:0]			00000000

PDM Operation

1. Setup M0_CK, M3_CK to decide the MCK.(Please refer to Section 5.1 for detailed instruction for setup)
2. Set PDMEN to enable the PDM Module.
3. Setup PMCS[2:0] to decide the PDM CLK frequency.
4. Setup PMD1[15:0] to decide the PDM output signal.
5. Set PT2EN[2] to assign the PT2[2] to be an output port.
6. Set PM1EN to assign the PT2[2] to be PDM Module output.

Table 8-3 PDM CLK selection table

PWCS	PDM CLK frequency
000	MCK
001	MCK/2
010	MCK/4
011	MCK/8
100	MCK/16
101	MCK/32
110	MCK/64
111	MCK/128

9. I2C Module (slave mode only)

FS98O25 embeds a slave mode I2C module. The two pins, SCL and SDA, are used to perform the I2C system. The pin SCL is assigned to be the clock pin, and the pin SDA is assigned to be the data pin in the I2C module. In an I2C system, there are master side and slave side. Master side would send the clock, slave ID and the commands to slave side. One master could connect to several slave sides with different IDs. First of all, the slave side would check if the ID sent by master side is the same as itself. If the ID matched, slave side would check the following bit from master. If the bit was low, it means that master side want to transfer some data or command to slave, so slave side should sent back an acknowledgement signal and then receive the data from master side. On the other hand, if the bit was high, it means that master side want to receive the data from slave side, so slave side should sent back an acknowledgement signal and then transmit the data back.(Please see Figure 9-1)

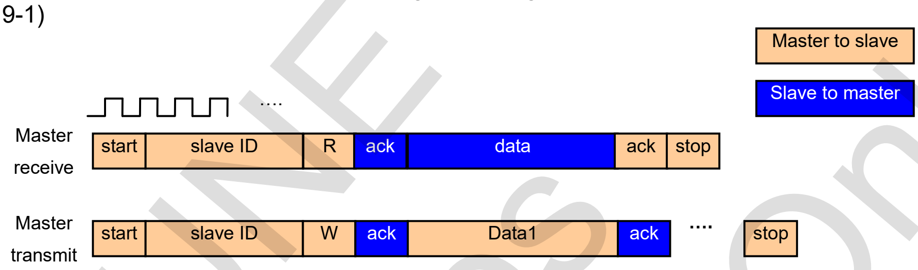


Figure 9-1 FS98O25 I2C module communication

In the I2C module embedded in FS98O25, there are 5 register flags shown in following figure. The SCL and SDA signal is connected to I2CSR and the Start and stop bit detector. The I2CSR is assigned to be the data buffer. When some signal is sent from master, the Start and stop bit detector will respond to the situation, and the Match detector will determine if the input data is matched with the slave ID. If it matches the ID, the user should send back the acknowledgement (data low) to respond to the master side. No matter whether the I2C module sends the data or receives the data, the I2CBUF is assigned to be the buffer. When the module receives the data, the data signals will be stored in the I2CSR, and send the whole data to I2CBUF after the data is sent completely.

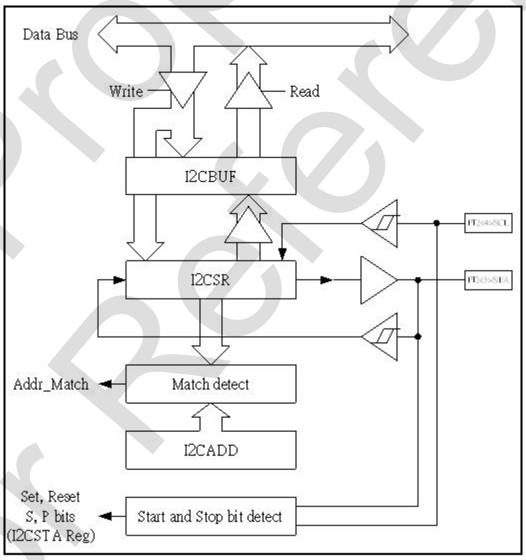


Figure 9-2 I2C module function block

Table 9-1 I2C module register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
57H	I2CCON	9	WCOL	I2COV	I2CEN	CKP					0001uuuu
58H	I2CSTA	9			DA	P	S	RW		BF	uu0000u0
59H	I2CADD	9	I2CADD [7:0]								00000000
5AH	I2CBUF	9	I2CBUF [7:0]								00000000

Register I2CCON at address 57H

property	R/W-0	R/W-0	R/W-0	R/W-1	U-X	U-X	U-X	U-X
I2CCON	**WCOL**	**I2COV**	**I2CEN**	**CKP**				
	Bit7							Bit0

Bit 7 **WCOL**: Write collision detector register flag.
1 = The I2CBUF register is written while it is still transmitting the previous data.
0 = No write collision is happened. This register should be clear in software.

Bit 6 **I2COV**: Receive overflow detector register flag
1 = A byte is received while the I2CBUF is still holding the previous data.
0 = No receive overflow is happened. This register should be clear in software

Bit 5 **I2CEN**: I2C module enable flag
1 = I2C module is enabled.
0 = I2C module is disabled.

Bit 4 **CKP**: SCK signal control register
1 = SCK pin is enabled.
0 = SCK pin is disabled and hold to low.

Property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register I2CSTA at address 58H

property	U-X	U-X	R/W-0	R/W-0	R/W-0	R/W-0	U-X	R/W-0
I2CSTA			**DA**	**P**	**S**	**RW**		**BF**
	Bit7							Bit0

Bit 5 **DA**: Data / Address bit register flag.
1 = The last received byte is data.
0 = The last received byte is address.

Bit 4 **P**: Stop bit register flag
1 = A stop bit is detected.
0 = No stop bit is detected. When the I2C module is disabled, this bit would be clear.

Bit 3 **S**: Start bit register flag
1 = A start bit is detected.
0 = No start bit is detected. When the I2C module is disabled, this bit would be clear.

Bit 2 **RW**: Read / Write register flag
1 = Read command is detected.
0 = Write command is detected.

Bit 0 **BF**: I2CBUF full register flag.
1 = I2CBUF is full. The user could get data from I2CBUF register.
0 = I2CBUF is empty.

Register I2CADD at address 59H

property	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
I2CADD	**I2CADD [7:0]**							
	Bit7							Bit0

Bit 7-0 **I2CADD[7:0]**: I2C module slave mode ID buffer register.

Property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register I2CBUF at address 5AH

property	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
I2CBUF	**I2CBUF [7:0]**							
	Bit7							Bit0

Bit 7-0 **I2CBUF[7:0]**: I2C module Data buffer register.

Property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Table 9-2 I2C register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
06H	INTF	3/6/7/9/10/11				--	I2CIF	--	--	--	00000000
07H	INTE	3/6/7/9/10/11	GIE			--	I2CIE	--	--	--	00000000
37H	PT2OCB	9				PT2OC[4:3]					uuu11uuu
57H	I2CCON	9	WCOL	I2COV	I2CEN	CKP					0001uuuu
58H	I2CSTA	9			DA	P	S	RW		BF	uu0000u0
59H	I2CADD	9	I2CADD [7:0]								00000000
5AH	I2CBUF	9	I2CBUF [7:0]								00000000

I2C data receive operation: (master to slave)

1. Configure SCL and SDA pins as open-drain through the PTOCB[4:3]
2. Set I2CEN register flag to enable the I2C module.
3. Clear I2CIF to reset the I2C interrupt.
4. Set I2CIE and GIE to enable the I2C interrupt.
5. Wait for the interrupt.
6. When the I2C master device sends data to slave side, the data (ID) transmitted from the master device will be sent to I2CBUF, and the BF register flag will be set.
7. If the RW register flag is clear,(low) the I2C module will enter the receive mode.
8. The acknowledgement signal will be sent automatically and an interrupt will occur.
9. Clear the I2CIF and reset the interrupt to wait for the interrupt happened again.
10. When an interrupt occurs, read the I2CBUF for receiving the data transmitted from master side. The acknowledgement signal will be sent automatically.
11. If the user doesn't read the data from I2CBUF, the BF register flag will be held high. When the data is sent to slave again, the I2COV register flag will be set, and the interrupt will NOT happen.

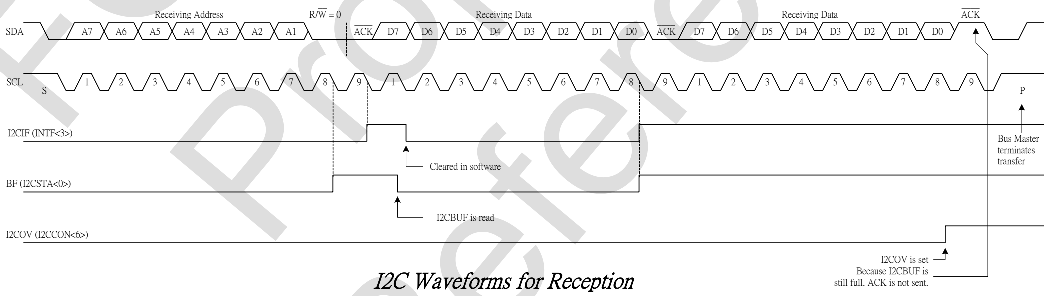


Figure 9-3 I2C waveform for reception

I2C data transmit operation: (slave to master)

1. Configure SCL and SDA pins as open-drain through the PTOCB[4:3].
2. Set I2CEN register flag to enable the I2C module.
3. Clear I2CIF to reset the I2C interrupt.
4. Set I2CIE and GIE to enable the I2C interrupt.
5. Wait for the interrupt.
6. When the I2C master device sends data to slave side, the data (ID) transmitted from the master device will be sent to I2CBUF, and the BF register flag will be set.
7. If the RW register flag is set,(high) the I2C module will enter the transmit mode.
8. The acknowledgement signal will be sent automatically and the interrupt will happen.
9. Set the CKP register flag to hold the SCK to low, and then write the data, which is ready to send to master side, to I2CBUF.
10. Clear the I2CIF and reset the interrupt to wait for the interrupt to happen again.
11. Clear the CKP register flag to enable the SCK pin. The master side will start to get the data.
12. When interrupt happen, the master side has already finished the transmission, the acknowledgement has been sent back to salve side, and the BF register flag has been clear.

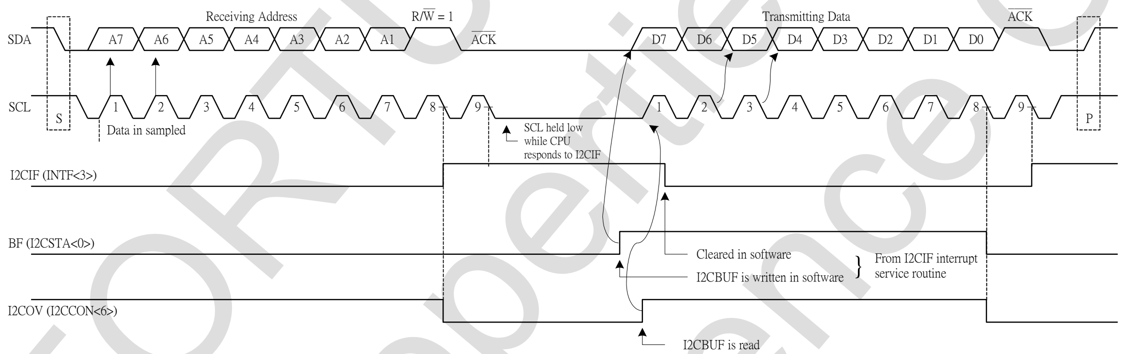


Figure 9-4 I2C waveforms for transmission

10. Analog Function Network

Please see Figure 10-1. FS98O25 Analog Function Network has 2 main functions: Low Noise OP Amplifier (OPAMP) and Sigma Delta Analog to Digital Converter (ADC). OPAMP is used to amplify the input analog signal for ADC. ADC is used to convert the analog signal to digital signal.

The OPAMP has 2 input ports as inverting side and non-inverting side. Users could setup SOP1P[2:0] and SOP1N[1:0] to choose the input signals. S_CH1CK[1:0] and OP1EN register flags are used to control OPAMP and OP1O is the OPAMP output port. The detailed operations will be described in Section 10.2.

The embedded ADC contains ***sigma delta modulator*** and ***digital comb filter***. It is a fully differential input system. User could give 2 signals for differential reference and 2 signals for differential input. ADC will convert the ratio of differential input to differential reference to 14-bit digital output. The related control instructions will be illustrated in Section 10.1.

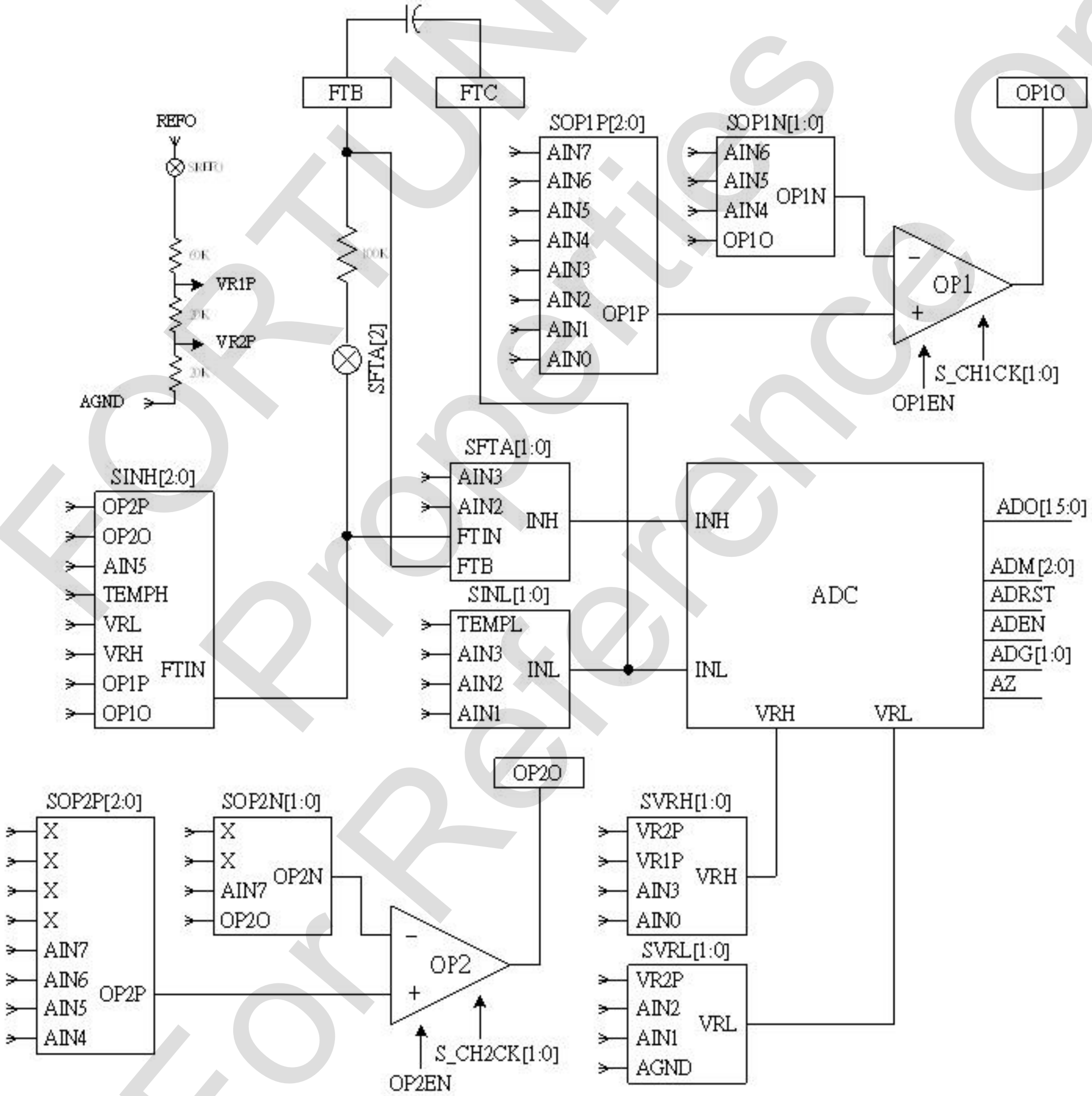


Figure 10-1 FS98O25 analog function network

Table 10-1 analog function network register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
06H	INTF	3/6/7/9/10/11				--	--	ADIF	--	--	00000000
07H	INTE	3/6/7/9/10/11	GIE			--	--	ADIE	--	--	00000000
10H	ADOH	10/11	ADO [15:8]								00000000
11H	ADOL	10/11	ADO [7:0]								00000000
12H	ADOLL	10/11	Extra ADC output register								00000000
13H	ADCON	10/11					ADRST	ADM [2:0]			uuuu0000
15H	PCK	4/5/7.5/10		--			S_CH1CK [1:0]		--	--	00000000
18H	NETA	10/11	SINL[1:0]		SINH[2:0]			SFTA[2:0]			00000000
19H	NETB	10/11	SOP2N[1:0]		SOP1N[1:0]		SVRL[1:0]		SVRH[1:0]		00000000
1AH	NETC	10/11	SREFO				ADG[1:0]		ADEN	AZ	00000000
1BH	NETD	10/11	OP2EN	SOP2P[2:0]			OP1EN	SOP1P[2:0]			00000000

Register ADOH at address 10H

Property	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
ADOH	**ADO [15:8]**							
	Bit7							Bit0

Register ADOL at address 11H

property	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
ADOL	**ADO [7:0]**							
	Bit7							Bit0

Bit 15-0 **ADO [15:0]**: ADC Digital Output

ADO[15] = ADC Digital Output sign bit. 0 = Output is positive; 1 = Output is negative.

ADO[14] = ADC Digital Output sign bit. 0 = Output is positive; 1 = Output is negative.

ADO[13] = ADC Digital Output Data bit 13.

ADO[12] = ADC Digital Output Data bit 12.

ADO[11] = ADC Digital Output Data bit 11.

ADO[10] = ADC Digital Output Data bit 10.

ADO[9] = ADC Digital Output Data bit 9.

ADO[8] = ADC Digital Output Data bit 8.

ADO[7] = ADC Digital Output Data bit 7.

ADO[6] = ADC Digital Output Data bit 6.

ADO[5] = ADC Digital Output Data bit 5.

ADO[4] = ADC Digital Output Data bit 4.

ADO[3] = ADC Digital Output Data bit 3.

ADO[2] = ADC Digital Output Data bit 2.

ADO[1] = ADC Digital Output Data bit 1.

ADO[0] = ADC Digital Output Data bit 0.

Property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

Register ADOLL at address 12H

property	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
ADOLL	**Extra ADC output register**							
	Bit7							Bit0

Users could take the value of 3 registers, ADOH, ADOL and ADOLL as 24 bits ADC output.

Register ADCON at address 13H

property	U-X	U-X	U-X	U-X	R/W-0	R/W-0	R/W-0	R/W-0
ADCON					**ADRST**	**ADM [2:0]**		
	Bit7							Bit0

Bit 3 **ADRST**: ADC comb filter enable register (Please refer to Section 10.1 for detail)

1 = ADC comb filter is enabled, ADC could work correctly.

0 = ADC comb filter is disabled, ADC digital output will be zero.

Bit 2-0 **ADM [2:0]**: ADC output rate selector

111 = ADC output rate is ADCF/8000[24]

110 = ADC output rate is ADCF/8000

101 = ADC output rate is ADCF/4000

100 = ADC output rate is ADCF/2000

011 = ADC output rate is ADCF/1000

010 = ADC output rate is ADCF/500

001 = ADC output rate is ADCF/250

000 = ADC output rate is ADCF/125

property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

[24] Please refer to Section 5.3 for ADCF information.

Register PCK at address 15H

property	U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0	U-0
PCK		**--**			**S_CH1CK [1:0]**		**--**	**--**
	Bit7							Bit0

Bit 3-2 **S_CH1CK [1:0]**: OPAMP Control Register (Please refer to Section 10.2)

11 = The OPAMP Chopper mode is enabled, and the Chopper frequency is CLK/1000

10 = The OPAMP Chopper mode is enabled, and the Chopper frequency is CLK/500

01 = The OPAMP Chopper mode is disabled. OPAMP input operation mode is set to be "-Offset".

00 = The OPAMP Chopper mode is disabled. OPAMP input operation mode is set to be "+Offset".

Property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register NETA at address 18H

property	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NETA	**SINL[1:0]**		**SINH[2:0]**			**SFTA[2:0]**		
	Bit7							Bit0

Bit 7-6 **SINL[1:0]**: ADC negative input port signal multiplexer (Please refer to Section 10.1)

11 = The ADC negative input port is connected to TEMPL. (Please refer to Section 4.6)

10 = The ADC negative input port is connected to AIN3 (PT1[3]).

01 = The ADC negative input port is connected to AIN2 (PT1[2]).

00 = The ADC negative input port is connected to AIN1 (PT1[1]).

Bit 5-3 **SINH[2:0]**: Embedded ADC Low Pass Filter input port signal multiplexer (Please refer to Section 10.1)

111 = The ADC Low Pass Filter input port is connected to OP2P. (Please refer to Section 4.4)

110 = The ADC Low Pass Filter input port is connected to OP2O (PT1[4]).

101 = The ADC Low Pass Filter input port is connected to AIN5 (PT1[5]).

100 = The ADC Low Pass Filter input port is connected to TEMPH. (Please refer to Section 4.6)

011 = The ADC Low Pass Filter input port is connected to VRL (ADC referenced voltage negative input).

010 = The ADC Low Pass Filter input port is connected to VRH (ADC referenced voltage positive input).

001 = The ADC Low Pass Filter input port is connected to OP1P (OPAMP non-inverting input port).

000 = The ADC Low Pass Filter input port is connected to OP1O (OPAMP output port).

Bit 2 **SFTA [2]**: FTIN and FTB connector (ADC Low Pass Filter enable flag)

1 = FTIN and FTB is short. ADC Low Pass Filter is enabled.

0 = FTIN and FTB is open. ADC Low Pass Filter is disabled.

Bit 1-0 **SFTA [1:0]**: ADC positive input port signal multiplexer (Please refer to Section 10.1)

11 = The ADC positive input port is connected to AIN3 (PT1[3]).

10 = The ADC positive input port is connected to AIN2 (PT1[2]).

01 = The ADC positive input port is connected to FTIN (SINH[2:0] multiplexer output port).

00 = The ADC positive input port is connected to FTB (FTIN output signal after Low Pass filter).

Property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register NETB at address 19H

property	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NETB	SOP2N[1:0]		**SOP1N[1:0]**		**SVRL[1:0]**		**SVRH[1:0]**	
	Bit7							Bit0

Bit 7-6 **SOP2N**: OPAMP inverting input port signal multiplexer (Please refer to Section 10.2)

00 = The OPAMP inverting input port is connected to OP2O (OPAMP output port).

01 = The OPAMP inverting input port is connected to AIN7 (PT1[7]).

10 = Not available

11 = Not available

Bit 5-4 **SOP1N[1:0]**: OPAMP inverting input port signal multiplexer (Please refer to Section 10.2)

11 = The OPAMP inverting input port is connected to AIN3 (PT1[3]).

10 = The OPAMP inverting input port is connected to AIN5 (PT1[5]).

01 = The OPAMP inverting input port is connected to AIN4 (PT1[4]).

00 = The OPAMP inverting input port is connected to OP1O (OPAMP output port).

Bit 3-2 **SVRL[1:0]**: ADC reference voltage negative input port signal multiplexer (Please refer to Section 10.1)

11 = The ADC negative referenced input port is connected to VR2P (1/5 REFO[25]).

10 = The ADC negative referenced input port is connected to AIN2 (PT1[2]).

01 = The ADC negative referenced input port is connected to AIN1 (PT1[1]).

00 = The ADC negative referenced input port is connected to AGND (Please refer to Section 4.4).

Bit 1-0 **SVRH[1:0]**: ADC reference voltage positive input port signal multiplexer (Please refer to Section 10.1)

11 = The ADC negative referenced input port is connected to VR2P (1/5 REFO).

10 = The ADC negative referenced input port is connected to VR1P (2/5 REFO).

01 = The ADC negative referenced input port is connected to AIN3 (PT1[3]).

00 = The ADC negative referenced input port is connected to AIN0 (PT1[0]).

Property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

[25] Please refer to Section 4.6 for REFO detailed information

Register NETC at address 1AH

property	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
NETC	**SREFO**				**ADG[1:0]**		**ADEN**	**AZ**
	Bit7							Bit0

Bit 7 **SREFO**: Internal Reference Voltage enable flag. (Please refer to Section 10.1)

1 = Internal Reference Voltage is enabled. VR1P = 2/5 REFO, VR2P = 1/5 REFO

0 = Internal Reference Voltage is disabled. VR1P and VR2P are floating.

Bit 3-2 **ADG[1:0]**: Internal ADC input gain. (Please refer to Section 10.1)

11 = Internal ADC input gain is 7/3

10 = Internal ADC input gain is 2

01 = Internal ADC input gain is 1

00 = Internal ADC input gain is 2/3

Bit 1 **ADEN**: ADC enable flag. (Please refer to Section 10.1)

1 = ADC is enabled.

0 = ADC is disabled.

Bit 0 **AZ**: ADC differential input ports short controller. (Please refer to Section 10.1)

1 = ADC differential input ports are short and both connect to INL[26] (SINL output).

0 = ADC differential input ports are NOT short. The 2 ports connect to INH and INL.

Property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

[26] That means the ADC input differential voltage is zero. ADC output should be zero counts. User could measure ADC offset counts when the AZ register flag is set.

Register NETD at address 1BH

property	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NETD	**OP2EN**	**SOP2P[2:0]**			**OP1EN**	**SOP1P[2:0]**		
	Bit7							Bit0

Bit 7 **OP2EN**: OPAMP enable flag. (Please refer to Section 10.2)
1 = OPAMP2 is enabled.
0 = OPAMP2 is disabled.

Bit 6-4 **SOP2P[2:0]**: OPAMP non-inverting input port signal multiplexer (Please refer to Section 10.2)
111 = Not available
110 = Not available
101 = Not available
100 = Not available
011 = The OPAMP non-inverting input port is connected to AIN7 (PT1[7]).
010 = The OPAMP non-inverting input port is connected to AIN6 (PT1[6]).
001 = The OPAMP non-inverting input port is connected to AIN5 (PT1[5]).
000 = The OPAMP non-inverting input port is connected to AIN4 (PT1[4]).

Bit 3 **OP1EN**: OPAMP enable flag. (Please refer to Section 10.2)
1 = OPAMP is enabled.
0 = OPAMP is disabled.

Bit 2-0 **SOP1P[2:0]**: OPAMP non-inverting input port signal multiplexer (Please refer to Section 10.2)
111 = The OPAMP non-inverting input port is connected to AIN7 (PT1[7])
110 = The OPAMP non-inverting input port is connected to AIN6 (PT1[6])
101 = The OPAMP non-inverting input port is connected to AIN5 (PT1[5]).
100 = The OPAMP non-inverting input port is connected to AIN4 (PT1[4]).
011 = The OPAMP non-inverting input port is connected to AIN3 (PT1[3]).
010 = The OPAMP non-inverting input port is connected to AIN2 (PT1[2]).
001 = The OPAMP non-inverting input port is connected to AIN1 (PT1[1]).
000 = The OPAMP non-inverting input port is connected to AIN0 (PT1[0]).

10.1. Analog to Digital Converter (ADC) :

Please see Figure 10-2. ADC Module contains 3 main functions – Low Pass Filter, Sigma Delta Modulator and Comb Filter. Before doing the AD conversion, User could reduce the low frequency noise by the embedded Low Pass Filter. The SINH[2:0] register flags are used to choose the input signal. SFTA[2] flag is used to enable the Filter. Sigma Delta Modulator and Comb Filter are used to complete the AD Converter. First of all the Modulator will output serial bits to show the ratio of the difference between INH and INL to the difference between VRH and VRL. For example, if the ratio of VRH and VRL to INH and INL is 7/10, the output bit series will be 7 'bit1' every 10 bits in average. Comb Filter is used to increase the SNR(signal-noise ratio) and the real ADC output, ADO, will be 14-bit precision in FS98O25.

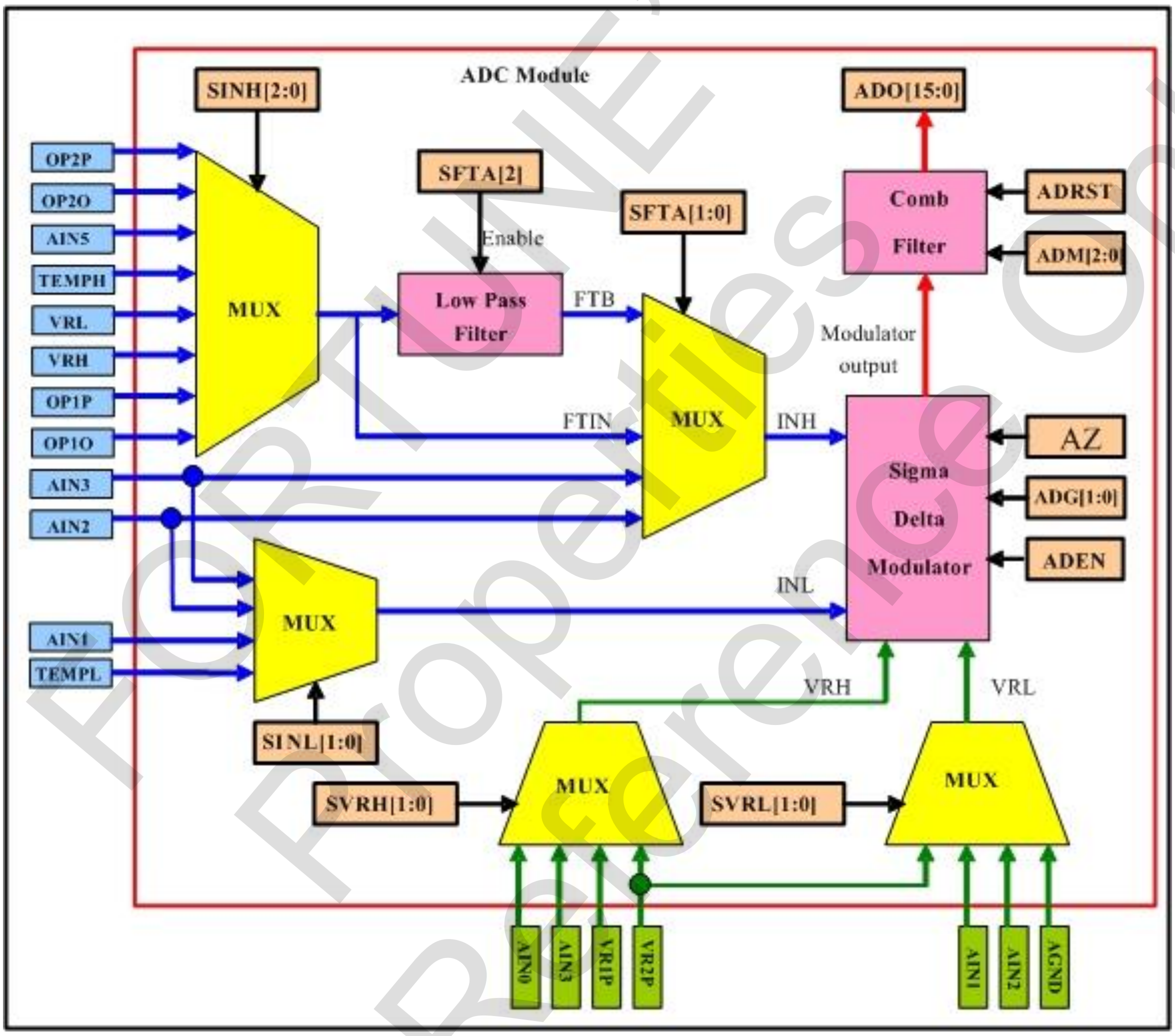


Figure 10-2 FS98O25 ADC function block

Table 10-2 ADC function register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
06H	INTF	3/6/7/9/10/11				--	--	ADIF	--	--	00000000
07H	INTE	3/6/7/9/10/11	GIE			--	--	ADIE	--	--	00000000
10H	ADOH	10/11	ADO [15:8]								00000000
11H	ADOL	10/11	ADO [7:0]								00000000
13H	ADCON	10/11					ADRST	ADM [2:0]			uuuu0000
14H	MCK	5	--	--	--	--	--	--	M1_CK	--	00000000
18H	NETA	10/11	SINL[1:0]		SINH[2:0]			SFTA[2:0]			00000000
19H	NETB	10/11	--		--		SVRL[1:0]		SVRH[1:0]		00000000
1AH	NETC	10/11	SREFO				ADG[1:0]		ADEN	AZ	00000000

ADC Operation

1. Operate as in Section 4.1 to get the VGG (2 times VDD or external Power Supply).
2. Operate as in Section 4.2 to get the VDDA (3.6V)
3. Operate as in Section 4.3 to enable the Analog Bias Circuit
4. Set SINH[2:0] and SFTA[2:0] to decide the ADC positive input port signal.(Table 10-3, 10-4 and 10-5)

Table 10-3 FTIN selection table

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

Table 10-4 FTB selection table

SFTA[2]	FTB[27]
0	ADC Low Pass Filter is disabled
1	ADC Low Pass Filter is enabled

[27] The input of ADC Low Pass Filter is FTIN, and the output is FTB

Table 10-5 INH selection table

SFTA[1:0]	INH (ADC positive input port signal)
00	FTB
01	FTIN
10	AIN2
11	AIN3

5. Set SINL[1:0] to decide the ADC negative input port signal. (Table 10-6)

Table 10-6 INL selection table

SINL[1:0]	INL (ADC negative input port signal)
00	AIN1
01	AIN2
10	AIN3
11	TEMPL

6. Set ADG[1:0] to decide the ADC input gain. (Table 10-7)

Table 10-7 ADG selection table

ADG[1:0]	ADC input gain
00	2/3
01	1
10	2
11	7/3

7. Set SREFO register flag to enable the VR1P and VR2P if needed. (VR1P = 2/5 REFO, VR2P = 1/5 REFO)
8. Set SVRH[1:0] to decide the ADC reference voltage positive input port signal. (Table 10-8)

Table 10-8 VRH selection table

SVRH[1:0]	VRH (ADC reference voltage positive input)
00	AIN0
01	AIN3
10	VR1P
11	VR2P

9. Set SVRL[1:0] to decide the ADC reference voltage negative input port signal. (Table 10-9)

Table 10-9 SVRL selection table

SVRL[1:0]	VRL (ADC reference voltage negative input)
00	AGND
01	AIN1
10	AIN2
11	VR2P

10. Set ADM[2:0] to decide the ADC output rate. (Table 10-10 and 10-11)

Table 10-10 ADC output rate selection table

ADM[2:0]	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

Table 10-11 ADC sample frequency selection table

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

11. Set ADIE and GIE register flags to enable the ADC interrupt
12. Set ADEN register flag, the embedded Σ-Δ modulator will be enabled.
13. Set ADRST register flag, the comb filter will be enabled.
14. When the ADC interrupt happen, read the ADO[15:0] to get the ADC output.(ADO[15:14] are signed bits)
15. Set AZ register flag to make the ADC positive and negative input port be internally short. Read the ADO[15:0] to get the ADC offset (The ADO should be zero if the offset is zero)
16. Clear AZ register flag to make the ADC work normally.

10.2. OPAMP : OP1 and OP2

Table 10-12 FS98O25 OPAMP register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
15H	PCK	4/5/7.5/10		--	S_CH2CK [1:0]		S_CH1CK [1:0]		--	--	00000000
19H	NETB	10/11	SOP2N[1:0]		SOP1N[1:0]		--		--		00000000
1BH	NETD	10/11	OP2EN	SOP2P[2:0]			OP1EN	SOP1P[2:0]			00000000

OPAMP1 Operation

1. Set SOP1P[2:0] to decide the OPAMP non-inverting input port signal. (Table 10-13)

Table 10-13 SOP1P selection table

SOP1P[2:0]	OP1P (OPAMP non-inverting input)
000	AIN0
001	AIN1
010	AIN2
011	AIN3
100	AIN4
101	AIN5
110	AIN6
111	AIN7

2. Set SOP1N[1:0] to decide the OPAMP inverting input port signal. (Table 10-14)

Table 10-14 SOP1N selection table

SOP1N[1:0]	OP1N (OPAMP inverting input)
00	OP1O
01	AIN4
10	AIN5
11	AIN6

3. Set S_CH1CK[1:0] to decide the OPAMP chopper mode.(Please see Section 3.6 for details)

Table 10-15 chopper mode selection table

S_CH1CK[1:0]	OPAMP chopper mode (input operation)
00	+Offset
01	-Offset
10	CLK/500 chopper frequency
11	CLK/1000 chopper frequency

4. Set OP1EN to enable the OPAMP.

Table 10-16 FS98O25 OPAMP register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
15H	PCK	4/5/7.5/10		--	S_CH2CK [1:0]		S_CH1CK [1:0]		--	--	00000000
19H	NETB	10/11	SOP2N[1:0]		SOP1N[1:0]		--		--		00000000
1BH	NETD	10/11	OP2EN	SOP2P[2:0]			OP1EN	SOP1P[2:0]			00000000

OPAMP2 Operation

1. Set SOP2P[2:0] to decide the OPAMP non-inverting input port signal. (Table 10-16)

Table 10-17 SOP2P selection table

SOP2P[2:0]	OP2P (OPAMP non-inverting input)
000	AIN4
001	AIN5
010	AIN6
011	AIN7
100	Not available
101	Not available
110	Not available
111	Not available

2. Set SOP2N[1:0] to decide the OPAMP inverting input port signal. (Table 10-17)

Table 10-18 SOP2N selection table

SOP2N[1:0]	OP2N (OPAMP inverting input)
00	OP2O
01	AIN7
10	Not available
11	Not available

3. Set S_CH2CK[1:0] to decide the OPAMP chopper mode.(Please see Section 3.6 for details)

Table 10-19 chopper mode selection table

S_CH2CK[1:0]	OPAMP chopper mode (input operation)
00	+Offset
01	-Offset
10	CLK/500 chopper frequency
11	CLK/1000 chopper frequency

4. Set OP2EN to enable the OPAMP.

11. ADC Application Guide

The ADC used in FS98O25 is a ∑-∆ ADC with fully differential inputs and fully differential reference voltage inputs. Its maximum output is ±15625. The conversion equation is as follows:

$$\text{Dout} = 15625 * G * \frac{\text{VIH - VIL} + \text{Vio}}{\text{VRH - VRL} + \text{Vro}}$$

- *G is ADC input gain. (refer to Section 10.1 ADC operation step 6)*
- *VIH is ADC's positive input voltage*
- *VIL is ADC's negative input voltage*
- *Vio is ADC's offset on the input terminals (Vio could be measured by using AZ register flag. See Section 11.4)*
- *VRH is the voltage at the positive input of Reference Voltage*
- *VRL is the voltage at the negative input of Reference Voltage*
- *Vro is the offset on the input terminals of Reference Voltage (Generally speaking, Vro could be ignored)*
- *The value (VRH-VRL+Vro) should be positive.*
- *When G * (VIH-VIL+Vio) / (VRH-VRL+Vro) ≥ 1, Dout=15625*
- *When G * (VIH-VIL+Vio) / (VRH-VRL+Vro) ≤ -1, Dout=-15625*

11.1. ADC Output Format

CPU can read ADO[14:0] as ADC's 15-bit output. Note that the output is in 2's complement format. The 14th bit of ADO[14:0] is sign bit. When the sign bit is cleared, the ADC output denotes a positive number, When the sign bit is set, the ADC output denotes a negative number.

Example:

ADO[15:0] = 0X257FH, then Dout = 9599.

ADO[15:0] = 0XE2F7H, then Dout = - (not (E2F7H) +1) = -7433.

11.2. 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.

11.3. ADC Output Rate and Settling Time

ADC output is the results of sigma delta modulator and the comb filter. The analog input signal needs to be sampled N[28] times and processed by the ADC and then the user could get one digital output. Generally speaking, the more times ADC samples the analog input signal, the more precise the digital output is.

When the user decides the sampling frequency and sampling counts, and then enables the ADC module, ADC module will send out a 15-bit signed digital output data every sampling N times and trigger the ADC interrupt.

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 ADC digital outputs are normally unstable ones, the third output and beyond are stable.

11.4. ADC Input Offset

ADC Input Offset Vio is NOT a constant. It drifts with ***temperature*** and ***common mode voltage*** at the inputs. To get a correct ADC result, Doff(ADC input offset digital output) should be deducted from the Dout. The instruction is as follows:

---

[28] 'N times' could be decided by setting ADM register flag (Please refer to Section 10.1).
FS98O25 ADC sampling frequency is decided by M1_CK(Please refer to Section 5.3).

1. Set AZ bit, and VIH and VIL will short. Dout will be 15625 *G * (Vio) / (VRH-VRL+Vro). It's called **Doff**.
2. Save Doff in memory, and then Clear AZ bit to restart the ADC module.
3. Pass the first 2 ADC interrupts for ignoring the unstable ADC result.
4. When measuring analog signal, Doff should be deducted.

11.5. ADC Digital Output

The ADC digital output deducted by Doff is **ADC Gain**. The ADC Gain doesn't change as VDD changes. The suggested values for common mode voltages at ADC input and reference voltage are 1V~2V.

ADC input gain could be set by ADG[1:0] register flag. Please see Section 10.1 for detail.

11.6. ADC Resolution

ADC resolution is mainly affected by the ADC sampling counts and the ADC reference voltage. Generally speaking, the more times ADC samples the analog input signal, the more precise the digital output is. The ADC sampling counts could be decided by ADM[2:0] register flag. The ADC digital output rolling counts versus ADM[2:0] and Reference voltage table are shown as follows:

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

Table 11-1 ADC rolling counts versus ADM

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

Table 11-2 ADC rolling counts versus VR

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

12. Low Noise Operation Amplifier Guide

The input noise of CMOS OPAMP is generally much larger than the one of a 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. FS98O25 use special CMOS low noise circuit design, and under normal conditions, the input noise is controlled under 1μVpp (0.1Hz~1Hz). FS98O25 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 FS98O25 is less than 0.5Mv (0.1Hz~1Hz). But an equivalent input current of less than 100Pa is generated, due to the effect of switching.

12.1. 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.

Please see Figure 12-1 for example.

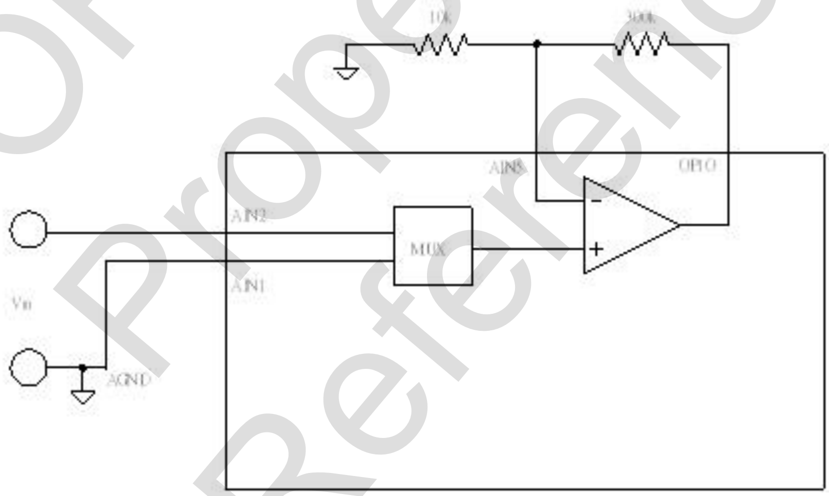


Figure 12-1 single end amplifier application example

12.2. 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.

Please see Figure 12-2 for example.

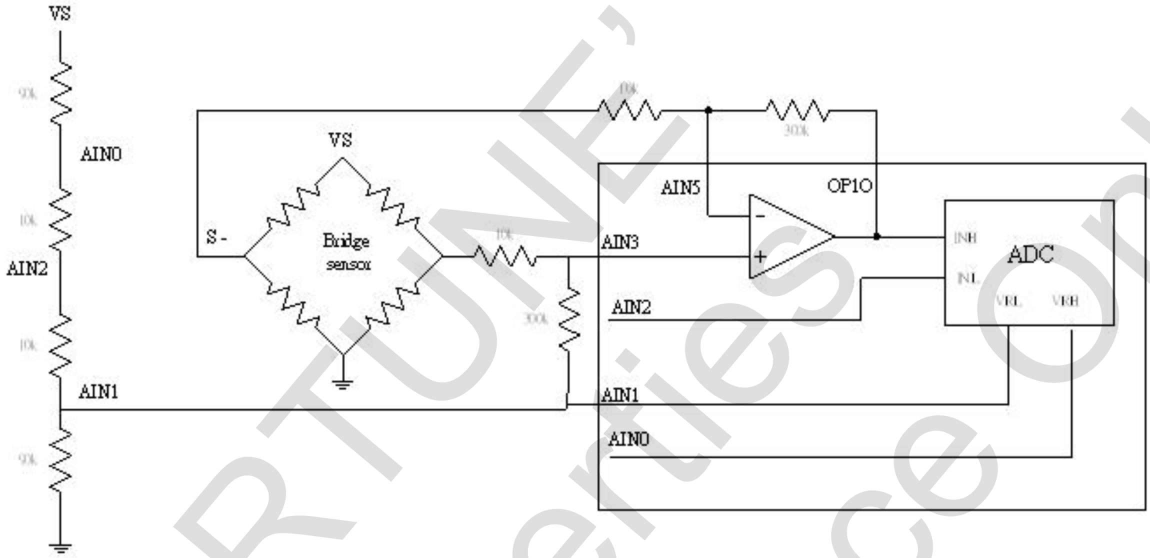


Figure 12-2 differential amplifier example

13. LCD Driver

FS98O25 embeds a LCD driver. The control signal are COM1~COM4 and SEG1~SEG32. The user could set the SEG register flags to drive a static or multiplexed LCD panel. FS98O25 LCD driver could drive up to 32 segments multiplexed with up to 4 commons. Please see Figure 13-1.

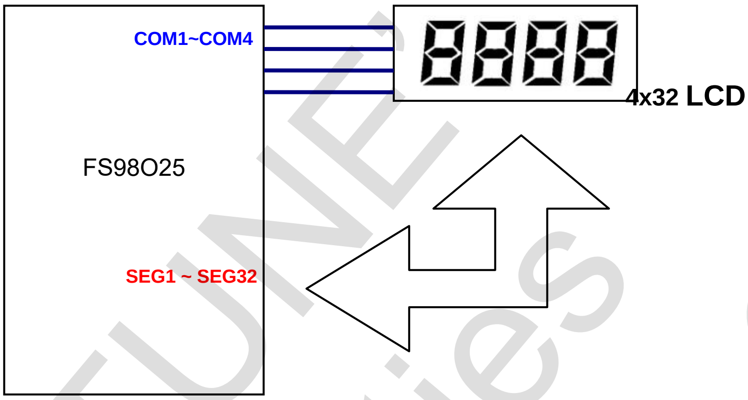


Figure 13-1 LCD driver control block

FS98O25 LCD driver has 4 kinds of control mode: static, 1/2 duty, 1/3 duty and 1/4 duty. The control mode depends on the LCD panel The user could setup LCD_DUTY[1:0] register flags to choose one. Take a 1/4 duty control mode number LCD for example, if the user wants to show number 9 in LCD, the SEG 1 includes 4 commons as [1,0,1,1] and the SEG2 include 4 commons as [1,1,1,1]. Please see Figure 13-2.

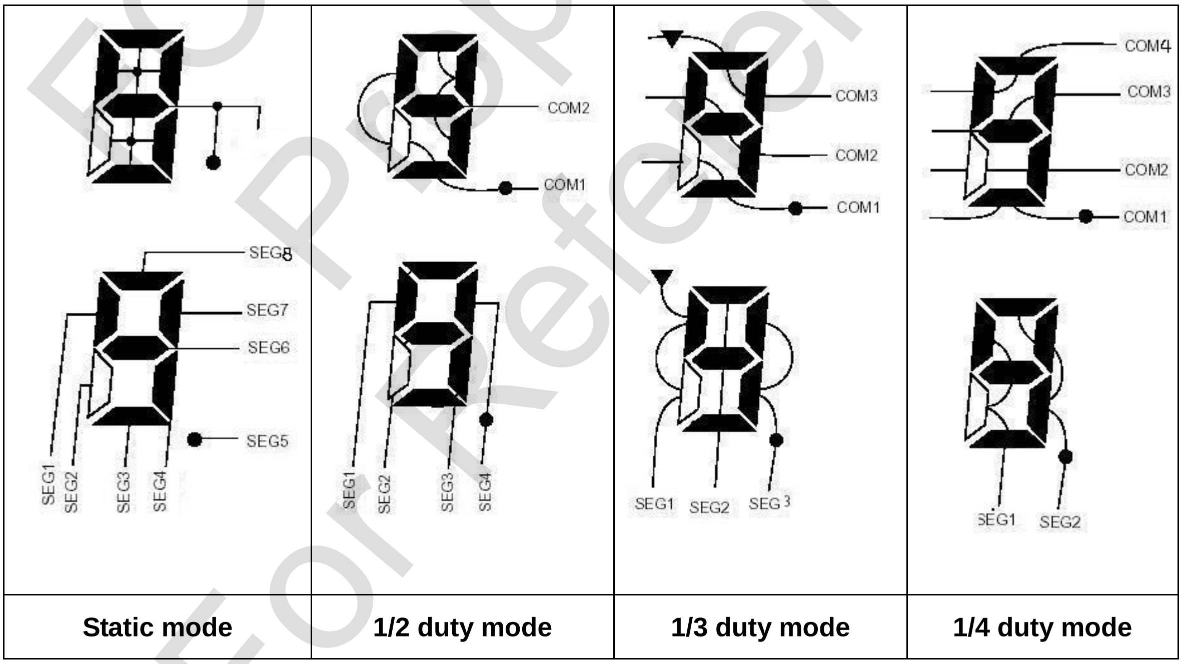


Figure 13-2 LCD control mode

The LCD frame frequency could be setup by setting the LCDCKS[1:0] register flags. FS98O25 divides the LCD Module input clock to get LCDCK. (Please see Table 13-1 and Table 13-2)

Table 13-1 LCD frame frequency selection table

LCDCKS [1:0]	LCD frame frequency (LCDCK)
00	LCD Input clock Frequency/8
01	LCD Input clock Frequency/16
10	LCD Input clock Frequency/32
11	LCD Input clock Frequency/64

Table 13-2 LCD duty selection table

LCD_DUTY [1:0]	Control mode	SEG 2, SEG 4, --- SEG 32				SEG 1, SEG 3, --- SEG 31			
		bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
00	static	-	-	-	-				
01	1/2	-	-	COM2	COM1	-	-	COM2	COM1
10	1/3	-	COM3	COM2	COM1	-	COM3	COM2	COM1
11	1/4	COM4	COM3	COM2	COM1	COM4	COM3	COM2	COM1

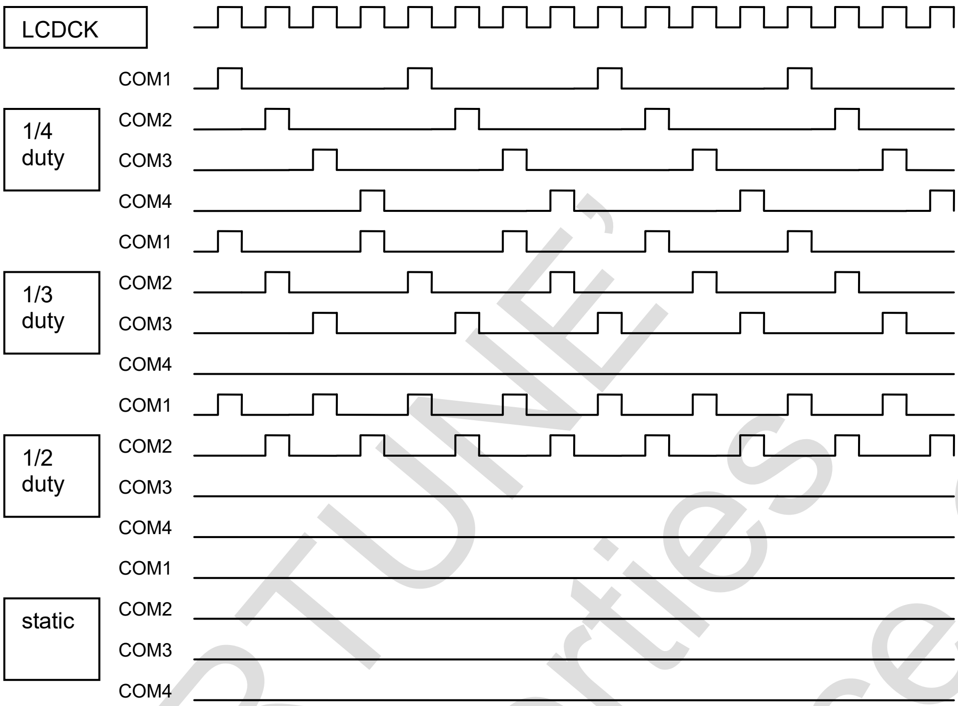


Figure 13-3 LCD duty mode working cycle

FS98O25 LCD driver has 3 voltage bias ports, such as V1, V2 and V3, and 2 kinds of power mode: 1/3 bias and 1/2 bias. Please see the following description to setup the LCD power system.

- 1/3 bias power system (Please see Figure 13-4 and 13-5)

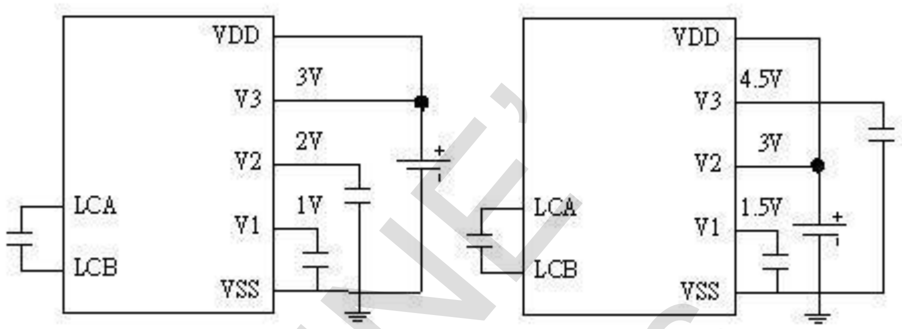


Figure 13-4 1/3 bias LCD power system circuit connection example

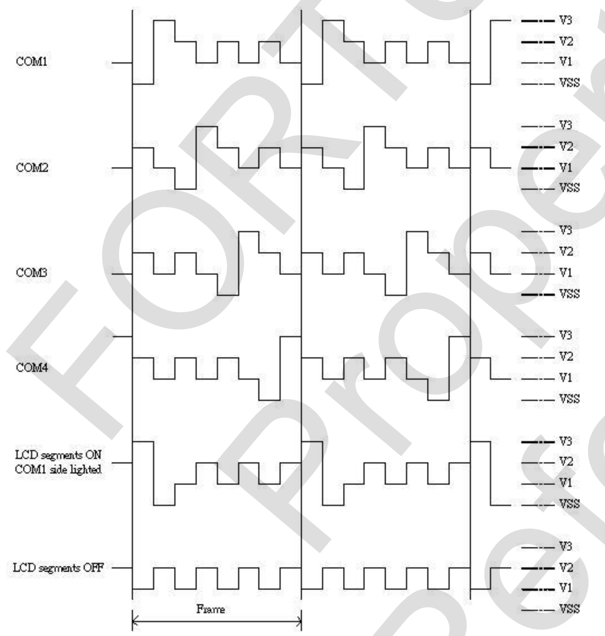
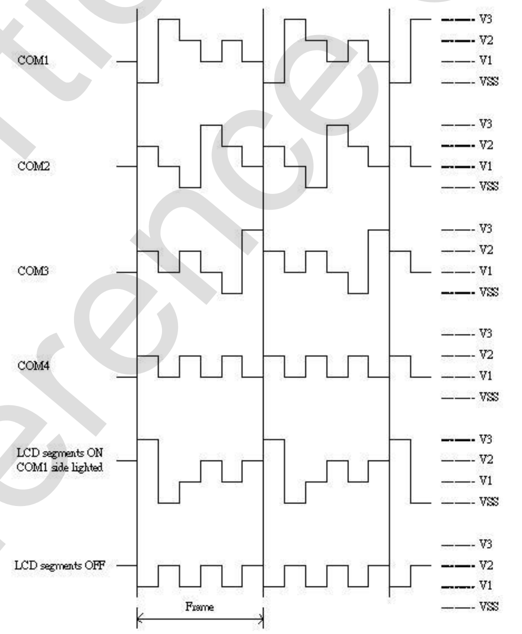
	
1/4 duty, 1/3 bias	1/3duty, 1/3bias

Figure 13-5 1/3 bias LCD power system clock

- 1/2 bias power system (Please see Figure 13-6 and 13-17)

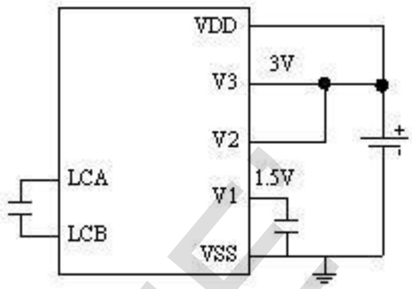
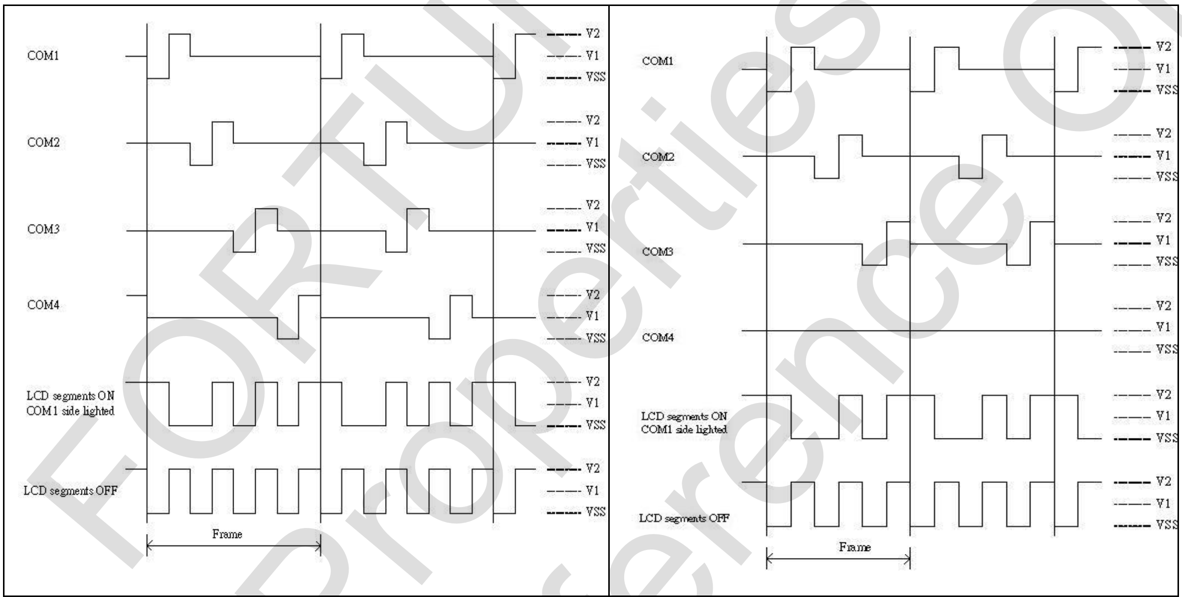


Figure 13-6 1/2 bias LCD power system circuit connection example



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

Figure 13-7 1/2 bias LCD power system clock

Table 13-3 FS98O25 LCD driver register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
40H	LCD1	13	SEG2 [3:0]				SEG1 [3:0]				uuuuuuuu
41H	LCD2	13	SEG4 [3:0]				SEG3 [3:0]				uuuuuuuu
42H	LCD3	13	SEG6 [3:0]				SEG5 [3:0]				uuuuuuuu
43H	LCD4	13	SEG8 [3:0]				SEG7 [3:0]				uuuuuuuu
44H	LCD5	13	SEG10 [3:0]				SEG9 [3:0]				uuuuuuuu
45H	LCD6	13	SEG12 [3:0]				SEG11 [3:0]				uuuuuuuu
46H	LCD7	13	SEG14 [3:0]				SEG13 [3:0]				uuuuuuuu
47H	LCD8	13	SEG16 [3:0]				SEG15 [3:0]				uuuuuuuu
48H	LCD9	13	SEG18 [3:0]				SEG17 [3:0]				uuuuuuuu
49H	LCD10	13	SEG20 [3:0]				SEG19 [3:0]				uuuuuuuu
54H	LCDENR	13	LCDCKS [1:0]		LCDEN		LEVEL	LCD_DUTY[1:0]		ENPMPL	00000000

Register LCD1 at address 40H

property	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
LCD1	**SEG2 [3:0]**				**SEG1 [3:0]**			
	Bit7							Bit0

Bit 7-4 **SEG2[3]**: LCD driver control signal: SEG2 with COM4 data.

SEG2[2]: LCD driver control signal: SEG2 with COM3 data.

SEG2[1]: LCD driver control signal: SEG2 with COM2 data.

SEG2[0]: LCD driver control signal: SEG2 with COM1 data.

Bit 3-0 **SEG1[3]**: LCD driver control signal: SEG1 with COM4 data.

SEG1[2]: LCD driver control signal: SEG1 with COM3 data.

SEG1[1]: LCD driver control signal: SEG1 with COM2 data.

SEG1[0]: LCD driver control signal: SEG1 with COM1 data.

Property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

Register LCD2 at address 41H

property	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
LCD2	**SEG4 [3:0]**				**SEG3 [3:0]**			
	Bit7							Bit0

Bit 7-4 **SEG4[3]**: LCD driver control signal: SEG4 with COM4 data.
SEG4[2]: LCD driver control signal: SEG4 with COM3 data.
SEG4[1]: LCD driver control signal: SEG4 with COM2 data.
SEG4[0]: LCD driver control signal: SEG4 with COM1 data.
Bit 3-0 **SEG3[3]**: LCD driver control signal: SEG3 with COM4 data.
SEG3[2]: LCD driver control signal: SEG3 with COM3 data.
SEG3[1]: LCD driver control signal: SEG3 with COM2 data.
SEG3[0]: LCD driver control signal: SEG3 with COM1 data.

Register LCD3 at address 42H

property	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
LCD3	**SEG6 [3:0]**				**SEG5 [3:0]**			
	Bit7							Bit0

Bit 7-4 **SEG6[3]**: LCD driver control signal: SEG6 with COM4 data.
SEG6[2]: LCD driver control signal: SEG6 with COM3 data.
SEG6[1]: LCD driver control signal: SEG6 with COM2 data.
SEG6[0]: LCD driver control signal: SEG6 with COM1 data.
Bit 3-0 **SEG5[3]**: LCD driver control signal: SEG5 with COM4 data.
SEG5[2]: LCD driver control signal: SEG5 with COM3 data.
SEG5[1]: LCD driver control signal: SEG5 with COM2 data.
SEG5[0]: LCD driver control signal: SEG5 with COM1 data.

Property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	'1' = Bit is Set	'0' = Bit is Cleared	X = Bit is unknown

Register LCD4 at address 43H

property	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
LCD4	**SEG8 [3:0]**				**SEG7 [3:0]**			
	Bit7							Bit0

Bit 7-4 **SEG8[3]**: LCD driver control signal: SEG8 with COM4 data.
SEG8[2]: LCD driver control signal: SEG8 with COM3 data.
SEG8[1]: LCD driver control signal: SEG8 with COM2 data.
SEG8[0]: LCD driver control signal: SEG8 with COM1 data.

Bit 3-0 **SEG7[3]**: LCD driver control signal: SEG7 with COM4 data.
SEG7[2]: LCD driver control signal: SEG7 with COM3 data.
SEG7[1]: LCD driver control signal: SEG7 with COM2 data.
SEG7[0]: LCD driver control signal: SEG7 with COM1 data.

Register LCD5 at address 44H

property	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
LCD5	**SEG10 [3:0]**				**SEG9 [3:0]**			
	Bit7							Bit0

Bit 7-4 **SEG10[3]**: LCD driver control signal: SEG10 with COM4 data.
SEG10[2]: LCD driver control signal: SEG10 with COM3 data.
SEG10[1]: LCD driver control signal: SEG10 with COM2 data.
SEG10[0]: LCD driver control signal: SEG10 with COM1 data.

Bit 3-0 **SEG9[3]**: LCD driver control signal: SEG9 with COM4 data.
SEG9[2]: LCD driver control signal: SEG9 with COM3 data.
SEG9[1]: LCD driver control signal: SEG9 with COM2 data.
SEG9[0]: LCD driver control signal: SEG9 with COM1 data.

Property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

Register LCD6 at address 45H

property	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
LCD6	**SEG12 [3:0]**				**SEG11 [3:0]**			
	Bit7							Bit0

Bit 7-4 **SEG12[3]**: LCD driver control signal: SEG12 with COM4 data.
SEG12[2]: LCD driver control signal: SEG12 with COM3 data.
SEG12[1]: LCD driver control signal: SEG12 with COM2 data.
SEG12[0]: LCD driver control signal: SEG12 with COM1 data.

Bit 3-0 **SEG11[3]**: LCD driver control signal: SEG11 with COM4 data.
SEG11[2]: LCD driver control signal: SEG11 with COM3 data.
SEG11[1]: LCD driver control signal: SEG11 with COM2 data.
SEG11[0]: LCD driver control signal: SEG11 with COM1 data.

Property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

Register LCDENR at address 54H

property	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
LCDENR	**LCDCKS [1:0]**		**LCDEN**		**LEVEL**	**LCD_DUTY[1:0]**		**ENPMPL**
	Bit7							Bit0

Bit 7-6 **LCDCKS[1:0]**: LCD frame frequency selector

11 = LCD frame frequency is assigned to be LCD input clock frequency/8

10 = LCD frame frequency is assigned to be LCD input clock frequency/16

01 = LCD frame frequency is assigned to be LCD input clock frequency/32

00 = LCD frame frequency is assigned to be LCD input clock frequency/64

Bit 5 **LCDEN**: LCD driver enable register flag

1 = The LCD driver is enabled. LCD clock is started

0 = The LCD driver is disabled. LCD clock is stopped

Bit 3 **LEVEL**: LCD driver voltage bias selector.

0 = LCD driver voltage bias is assigned to be 1/3 bias.

1 = LCD driver voltage bias is assigned to be 1/2 bias.

Bit 2-1 **LCD_DUTY[1:0]**: LCD driver control mode (SEG duty cycle)

11 = LCD driver control mode is assigned to be 1/4 duty cycle mode.

10 = LCD driver control mode is assigned to be 1/3 duty cycle mode.

01 = LCD driver control mode is assigned to be 1/2 duty cycle mode.

00 = LCD driver control mode is assigned to be static mode

Bit 0 **ENPMPL**: LCD driver charge pump enable register flag

1 = LCD driver charge pump is enabled.

0 = LCD driver charge pump is disabled.

Property

R = Readable bit	W = Writable bit	U = unimplemented bit	
- n = Value at Power On Reset	‘1’ = Bit is Set	‘0’ = Bit is Cleared	X = Bit is unknown

Table 13-4 LCD driver register table

Address	Name	Referenced Section	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power on Reset
14H	MCK	5	--	--	M5_CK		M3_CK	--	M1_CK	M0_CK	00000000
40H	LCD1	13	SEG2 [3:0]				SEG1 [3:0]				uuuuuuuu
41H	LCD2	13	SEG4 [3:0]				SEG3 [3:0]				uuuuuuuu
42H	LCD3	13	SEG6 [3:0]				SEG5 [3:0]				uuuuuuuu
43H	LCD4	13	SEG8 [3:0]				SEG7 [3:0]				uuuuuuuu
44H	LCD5	13	SEG10 [3:0]				SEG9 [3:0]				uuuuuuuu
45H	LCD6	13	SEG12 [3:0]				SEG11 [3:0]				uuuuuuuu
46H	LCD7	13	SEG14 [3:0]				SEG13 [3:0]				uuuuuuuu
47H	LCD8	13	SEG16 [3:0]				SEG15 [3:0]				uuuuuuuu
48H	LCD9	13	SEG18 [3:0]				SEG17 [3:0]				uuuuuuuu
49H	LCD10	13	SEG20 [3:0]				SEG19 [3:0]				uuuuuuuu
4AH	LCD11	13	SEG22 [3:0]]				SEG21 [3:0]				uuuuuuuu
4BH	LCD12	13	SEG24 [3:0]				SEG23 [3:0]				uuuuuuuu
4CH	LCD13	13	SEG26 [3:0]				SEG25 [3:0]				uuuuuuuu
4DH	LCD14	13	SEG28 [3:0]				SEG27 [3:0]				uuuuuuuu
4EH	LCD15	13	SEG30 [3:0]				SEG29 [3:0]				uuuuuuuu
4FH	LCD16	13	SEG32 [3:0]				SEG31 [3:0]				uuuuuuuu
54H	LCDENR	13	LCDCKS [1:0]		LCDEN		LEVEL	LCD_DUTY[1:0]		ENPMPL	00000000

LCD operation

1. Connect the 32 segment ports and 4 common ports to LCD panel.
2. Setup LEVEL register flag to decide the LCD driver power system. (0 = 1/3 bias, 1 = 1/2 bias)
3. Set ENPMPL to enable the LCD charge pump.
4. Setup M0_CK,M1_CK,M3_CK and M5_CK to decide the LCD input clock frequency.(Refer to Section 5.7)

Table 13-5 CLK selection table

M1_CK	**CLK**
0	MCK
1	MCK/4

Table 13-6 MCK selection table

M3_CK	M0_CK	**MCK**
X	0	ICK
0	1	ECK
1	1	ECK/2

Table 13-7 TMCLK selection table

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

5. Setup LCDCKS[1:0] register flags to decide the LCD Clock frequency.

Table 13-8 LCD frame frequency selection table

LCDCKS [1:0]	LCD frame frequency (LCDCK)
00	LCD Input clock Frequency/8
01	LCD Input clock Frequency/16
10	LCD Input clock Frequency/32
11	LCD Input clock Frequency/64

6. Setup LCD_DUTY[1:0] register flag to decide the control mode.(SEG duty cycle)

Table 13-9 LCD duty control mode selection table

LCD_DUTY [1:0]	Control mode
00	static
01	1/2
10	1/3
11	1/4

7. Set LCDEN to enable the LCD driver.

14. Halt and Sleep Modes

FS98O25 supports low power working mode. When the user want FS98O25 to do nothing and just stand by, FS98O25 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
MOVLW   01h
MOVWF   PT2PU    ; PT2 Pull up resistor is disconnected except port 0(external interrupt)
MOVLW   0Feh
MOVWF   PT2EN    ; PT2 ports are assigned to be output ports except port 0
CLRF PT2         ; Set PT2 [7:1] Output Low
CLRF INTF        ; Clear the interrupt flags
MOVLW   081h
MOVWF   INTE     ; Enable the external interrupt
SLEEP            ; Set the FS98O25 into Sleep mode
NOP              ; Guarantee that the program works normally when CPU wakes up.
```

15. Instruction Set

The FS9XXX instruction set consists of 37 instructions. Each instruction could be converted to 16-bit OPCODE. The detailed descriptions are shown in the following sections.

15.1. Instruction Set Summary

Table 15-1 FS98O25 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

15.2. 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

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

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

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

DECFSZ	Decrement f, skip if zero
Syntax	DECFSZ f, d 0 ≤ f ≤ FFh d ∈ [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

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

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

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

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

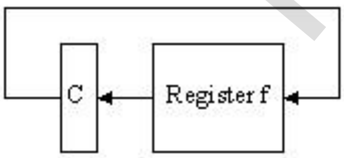
MOVWF	Move W to f
Syntax	MOVWF f 0 ≤ f ≤ FFh
Operation	[f] ← [W]
Flag Affected	None
Description	Move data from the W register to [f].
Cycle	1
Example: **MOVWF OPERAND**	Before instruction: W = 88h, OPERAND = 23h After instruction: W = 88h, OPERAND = 88h

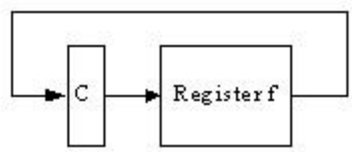
NOP	No Operation
Syntax	NOP
Operation	No Operation
Flag Affected	None
Description	No operation. NOP is used for one instruction cycle delay.
Cycle	1

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

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

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.[29]
Cycle	1

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

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

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 $0 \le k \le FFh$
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 $0 \le f \le FFh$ $d \in [0,1]$
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

16. Package Information

16.1. Package Outline

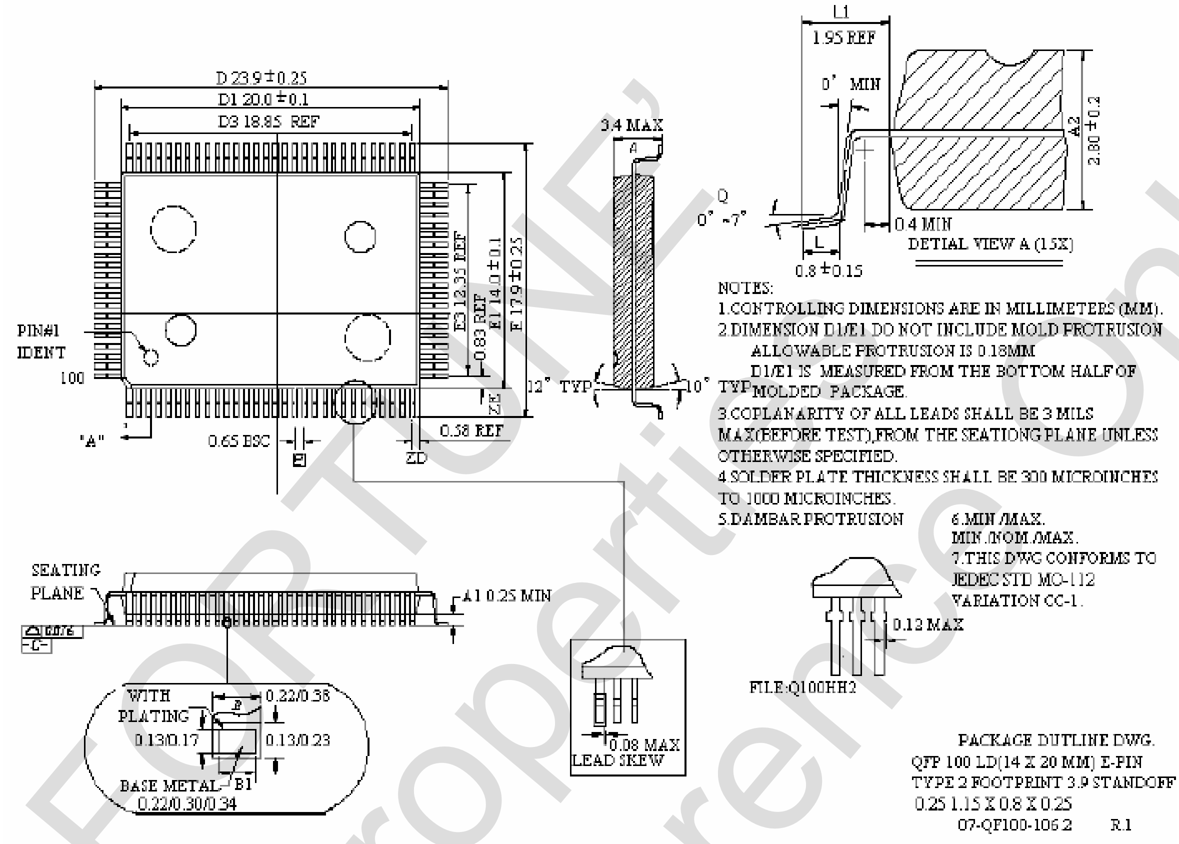


Figure 16-1 FS98O25 package outline

17. Revision History

Ver.	Date	Page	Description
1.0	2008/04/23	All	Initial release.
1.1	2008/10/9	13-14	Move VSSP~RST PIN from 90~99 to 91~100
1.2	2008/11/10	12	Add FS98O251 (6K ROM version) in ordering information
1.3	2008/12/30	34	Low Battery Comparator Input Selector Correct
1.4	2009/07/08	20	Revise Ambient Operating Temperature from -10~85 °C to 0~70 °C and add LTOL test condition description
		20	Revise Sleep Current Unit：Ma
1.5	2009/10/27	21	Revise Input Offset TYP：1.5Mv
1.6	2013/12/16	20	Revise Ambient Operating Temperature from 0~70 °C to -40~85 °C and add LTOL test condition description
1.7	2019/07/19	19 20 38	To change “Ma” to “mA” and the unit of IDD2 changes “Ma” to “µA” To Change “Mv” to “mV” To Change “Mv” to “mV”