CSU1221 Specification

Apr.2010

SHENZHEN Chipsea Technologies CO.,LTD.

Contents

Diagram Lists

Table Lists

1 CSU1221 DESCRIPTION

The CSU1221 is a CMOS 8-bit single chip microcontroller(MCU) with embedded a 4kx16 bits one-time programmable (OTP) ROM, a 2-channel 24-bit fully differential input analog to digital converter, low noise amplifier.

1.1 Main Features

1.1.1 High Performance RISC CPU

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

1.1.2 Peripheral Features

- 17-bit bi-directional I/O port.
- Two PDM (Pulse Density Modulator) output.
- One Buzzer output.
- One 2-channel 24-bit fully differential input analog to digital converter(ADC)
- 4-level embedded PGA
- Two external Interrupts
- Low battery detector pin
- Embedded temperature sensor

1.1.3 Analog Features

- 4-channel analog input ADC (2-channel differential input) ,24-bit resolution,20-bit effective precision (PGA=1)
- Embedded programmable gain amplifier (PGA),providing gains of 1 / 64 /128 / 256,applied to multiple signal range
- Low noise PGA

1.1.4 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 (3.0V regulated output).
- Embedded bandgap voltage reference (typical 3.0V, set 2.2/2.5 / 2.8 / 3.0V).
- 4 Interrupt sources (external: 2, internal: 2).
- Watchdog timer (WDT).
- Embedded 1.0 MHz oscillator.
- Package: 28-PIN SDIP/28-PIN SOP

1.1.5 CMOS Technology

- Voltage operation ranges from 2.4V to 3.6V.
- Operation current is less than 3 mA; sleep mode current is less than 3μA.

1.2 Application

- Sensor or transducer measurement applications.
- Electronic kitchen scale, personal scale, body fat scale, forehead/ear thermometer, low power price-computing scale
- Digital meter.

1.3 Pin Configuration

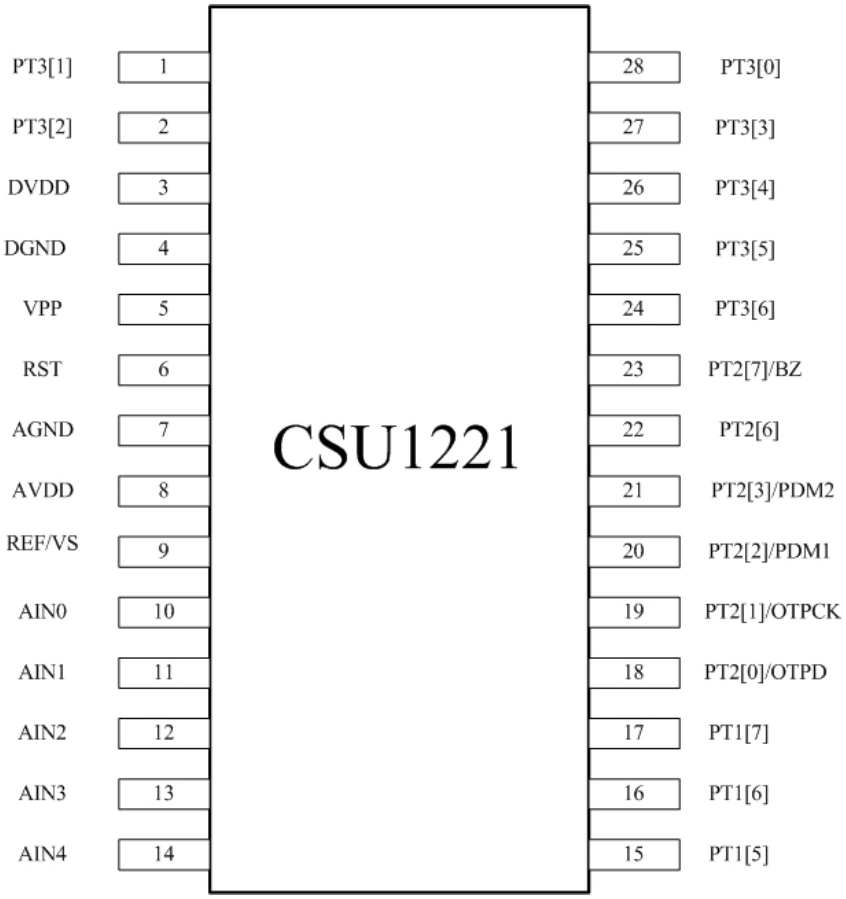


Figure 1-1 CSU1221 pin configuration

Table 1-1 CSU1221 pin description

管脚名称	输入/输出	管脚序号	描述
PT3[1]~[2]	I/O	1~2	I/O
DVDD	I	3	Digital Power Source
DGND	I	4	Digital Ground
VPP	I	5	Programming Power Supply, Connected to 6.5V (Connected to VDD on normal operation)
RST	I	6	CPU Reset, low level effective
AGND	I	7	Analog Ground
AVDD	I	8	Analog Power Source
REF/VS	I/O	9	Reference Voltage Source. Connecting phase compensation capacitor or Reference Voltage Source. Connecting phase compensation capacitor
ANA0~3	I	10~13	Analog differential input
ANA4	I	14	Low Battery detector input
PT1[5]~[7]	I/O	15~17	I/O
PT2[0]/OTPD	I/O	18	OTP Programming Data or I/O Port
PT2[1]/OTPCK	I/O	19	OTP Programming Clock or I/O Port
PT2[2]/PDM1	I/O	20	PDM output or I/O Port
PT2[3]/PDM2	I/O	21	PDM output or I/O Port
PT2[6]	I/O	22	I/O
PT2[7]/BZ	I/O	23	Buzzer output or I/O Port
PT3[6]~[3]	I/O	24~27	I/O
PT3[0]	I/O	28	I/O

1.4 Functional Block Diagram

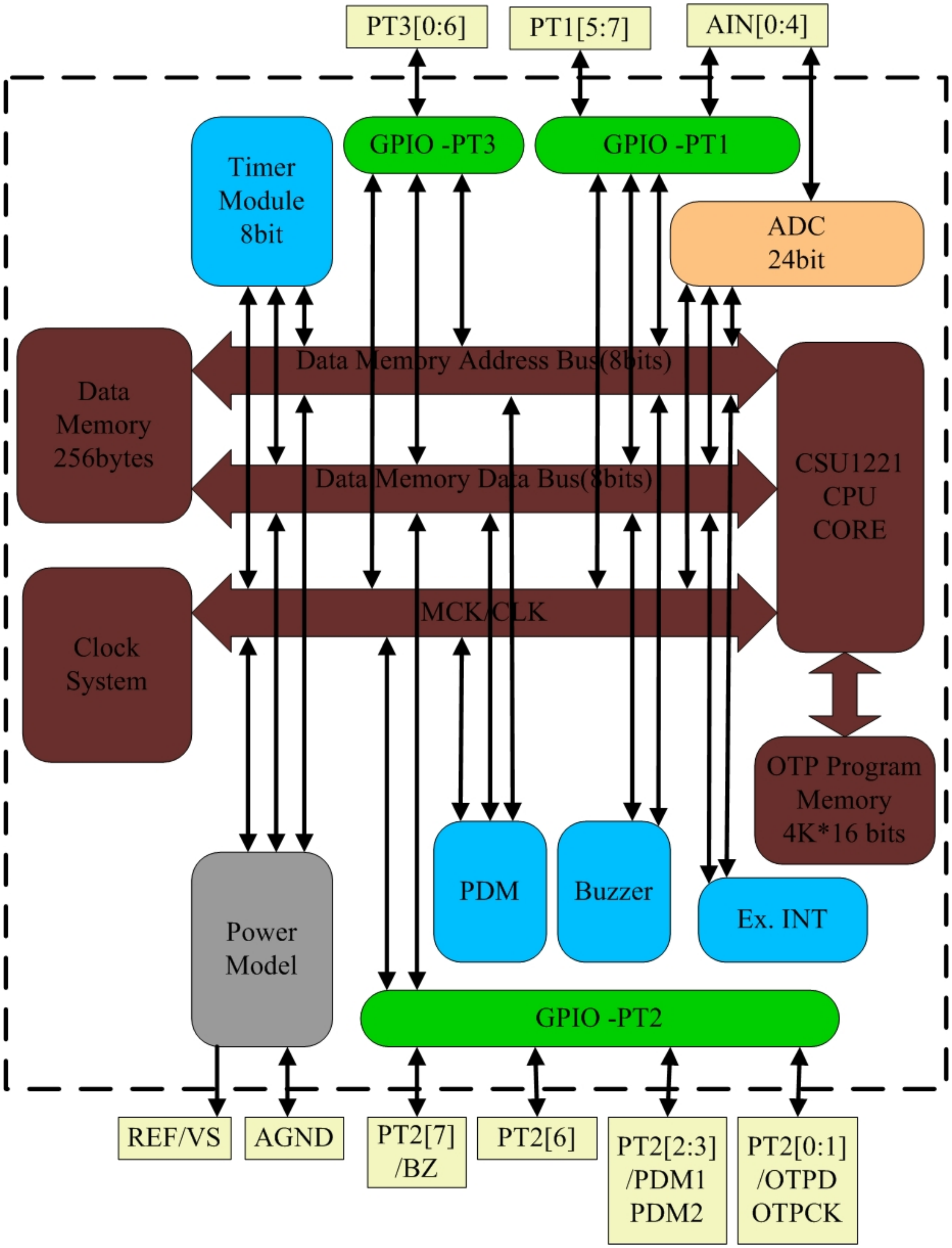


Figure 1-2 CSU1221 function block

There are 5 kinds of functional blocks in the Function Block Diagram, described as table 1-2:

Table 1-2 CSU1221 main function description table

Item	Sub Item	Description
CPU Kernel	RISC CPU Core	Please refer to Chapter 2.1 for detailed description
	OTP Program Memory	OTP: One Time Programmable 8k bytes is used for 4k line programming instructions
	Data Memory	CSU1221 has 384 bytes SRAM embedded in it. (128 bytes registers, 256 bytes general data memory)
	Clock sys	There are two clock sources in CSU1221. One is the internal clock which generates 1MHZ for CPU works, and the other is an external one which provides 32768 HZ clock signal to the chip.
Digital Function	Timer Module	Clock Counter for Time out interrupt and Watch dog Timer
	PDM	Similar to PWM function
	Buzzer	User should connect a Buzzer to the embedded buzzer port to receive the warning or reminding signal.
	Ext.INT	CSU1221 supports 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.
Power Function	Power Module	CSU1221 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 3 bits.
	PT2	The PT2 port has 8 bits. User can define these 8 bits for general purpose or some special function as External Interrupt, PDM and the Buzzer
	PT3	The PT3 port has 7 bits. User can use it after defining

1.5 Electrical Characteristics

1.5.1 Absolute Maximum Ratings

Table 1-3 CSU1221 absolute maximum rating table

Parameter	Rating	Unit
Power Supply VDD	3.6	V
Applied Input/Output Voltage	-0.3~VDD+0.3	V
Ambient Operating Temperature	-40~+85	ºC
Storage Temperature	-55~+150	ºC
Soldering Temperature, Time	220ºC，10 s	

CSU1221 has passed -40 ºC low temperature operation life test, VDD = 3.3V.

1.5.2 DC Characteristics (VDD=3.3V, TA=25℃, unless otherwise noted)

Table 1-4 CSU1221 DC characteristics

Symbol	Parameter	Test Conditions	Min.	Typ.	Max	Unit
VDD	Recommended Operation Power Voltage		2.4	3.3	3.6	V
IDD1	Supply Current 1	MCK = 1MHz CPUCLK=MCK/2 Charge Pump,ADC on		3		mA
IDD2	Supply Current 2	Internal Oscillator Off, MCK = 32768Hz		3		uA
IPO	Sleep Mode Supply Current	Sleep Instruction			3	uA
VIH	Digital Input High Voltage	PT2	VDD-0.6			V
VIL	Digital Input Low Voltage	PT2			0.3	V
IPU	Pull up Current	PT1,2,3 Vin = 0		30		uA
IOH	High Level Output Current	VOH=VDD-0.3V		3		mA
IOL	Low Level Output Current	VOL=0.3V		3		mA
VDDA	Analog Power			3		V
IREG	VDDA Regulator Output Current	VDD=3V Internal Voltage Double VDDA=2.85*V（unload）		0.8		mA
VCVDDA	VDDA Voltage Coefficient		-2		2	%V
VREF	Build in Reference Voltage To AGND			1.256		V
TCLREF	Build in Reference Voltage Temperature Coefficient	TA = -40~80ºC		50		ppm/ºC
VLBAT	Low Battery Detection Voltage	S_LB[1:0]=00		2.4		V
		S_LB[1:0]=01		3.6		
FRC	Internal RC oscillator		0.8	1.0	1.2	MHz
FWDT	Internal WDT Clock			3		KHz

1.5.3 ADC Characteristics (VDD=3.3V, TA=25℃, unless otherwise noted)

Table 1-5 CSU1221 ADC characteristics

Parameter		Test Conditions	Min.	Typ.	Max	Unit
Analog Input	Analog Input Range	Buffer Off	AGND-0.1		AGND+0.1	V
		Buffer On	AGND+0.4		AGND-1.5	V
	The Whole Range Of Input voltage				± VREF/PGA	V
	Differential input impedance	Buffer Off		16/PGA		MΩ
		Buffer On		5		GΩ
System Performance	Resolution	No missing codes		24		Bits
	Input Noise	Gain = 1		2.8		uV
		Gain = 128		120		nV
	Integral Nonlinearity	Gain = 128		±0.004		%of FS
	Offset Error	Gain = 128		14		uV
	Offect Drift	Gain = 128		0.05		uV/℃
	Gain Error	Gain = 128		0.1		%
	Gain Error drift	Gain = 128		4		ppm/℃
Reference Voltage	V_{VS}	LDOS[1:0]= 11	2	2.2	2.3	V
		LDOS[1:0]= 10	2.3	2.5	2.6	
		LDOS[1:0]= 01	2.5	2.8	2.8	
		LDOS[1:0]= 00	2.6	3	3	
	V_{VS} Temperature Coefficient			100		ppm/℃

2 FUNCTIONAL BLOCK DESCRIPTION

2.1 CPU Core

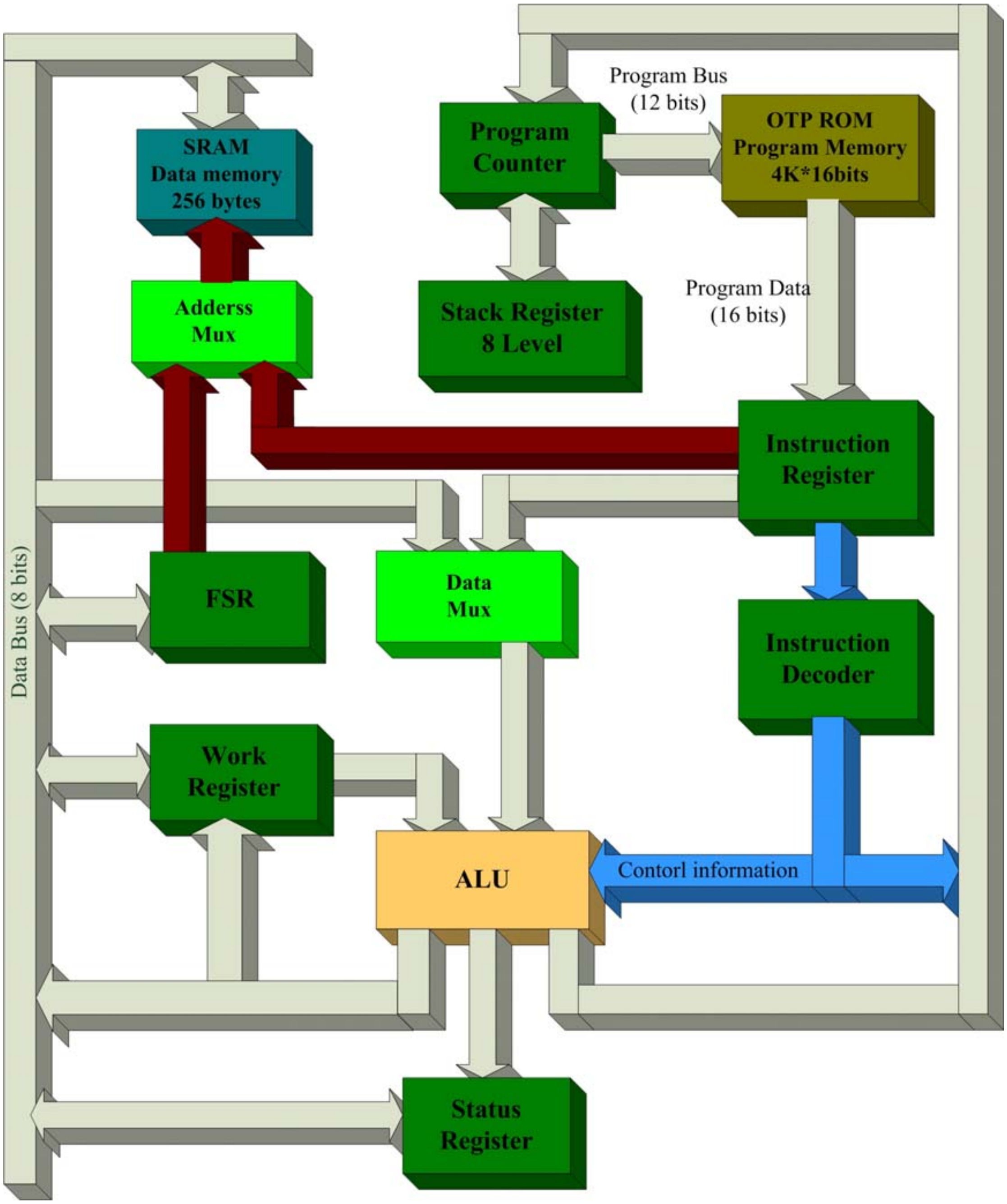


Figure 2-1 CSU1221 CPU core function block

The “CPU Core Block Diagram” shown in Section 2.1 mainly includes 7 important registers and 2 memory units. Please see the Figure 2-1 and the Table 2-1 for detailed information.

Table 2-1 CSU1221 CPU core block diagram description table

Items	Description
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 (12bits), 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. CSU1221 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 (9bits) 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。
Arithmetic and Logic Unit	Arithmetic and Logic unit not only can do arithmetic operation on 8 bit binary as addition , substruction, adding 1 and subtracting 1,but also can do logical operation on 8 bit variable as logical and, or, xor, cyclic shift, complement and clear.
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.
File Select Register	In CSU1221 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.
Program Memory	CSU1221 has an embedded 8k bytes OTP (One Time Programmable) ROM as Program Memory. Because the OPCODE of the instruction is 16 bits, user could program 4k instructions in CSU1221 at most. Program Memory Address Bus is 12 bits, and the Data Bus is 16bits.
Data Memory	CSU1221 has an embedded 256bytes SRAM as Data Memory. The Data Memory Address Bus is 9 bits, and Data Bus is 8 bits.

2.1.1 Clocking Scheme/Instruction Cycle

CPU is real RISC configuration and single clock Instruction cycle.

One Instruction cycle (CPU cycle) includes 4 steps and the CPU could process 2 steps per CPU Clock, so one instruction cycle is two CPU clock. As the CPU clock can reach 1MHZ at most, the fastest instruction cycle is 500KHZ. The 4 steps are described as follows.

Step1.Fetch

Program Counter pushes the Instruction Pointer into Program Memory, and the pointed Data in the Program Memory is stored in the Instruction Register.

Step2.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.

Step3.Execute

ALU executes the process based on the decoded Control Information while setting Program Counter to standby mode for reducing power dissipation.

Step4.Write Back

Push the ALU result to Work Register or Assigned Data Memory Address.

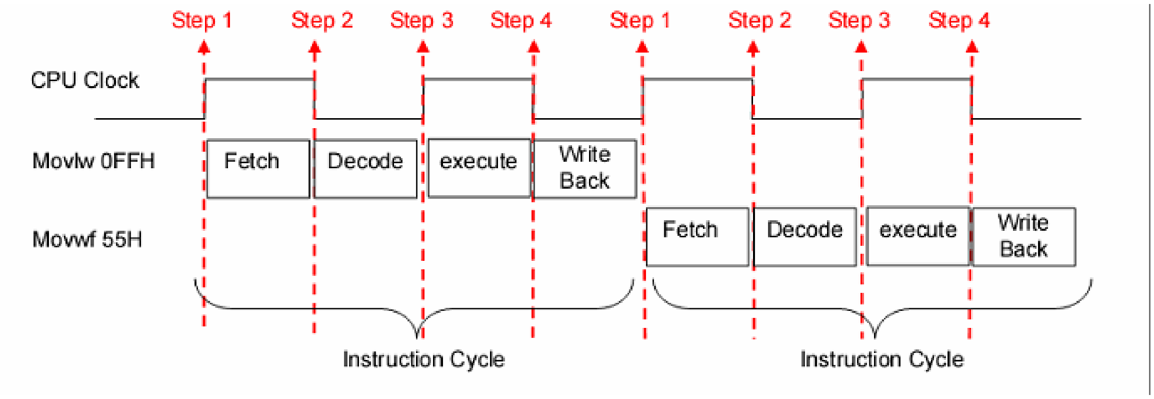


Figure 2-2 CSU1221 instruction cycle

2.1.2 Memory Organization

1. CSU1221 has a 12bits Program Counter which is capable of addressing a 4k×16bits program memory space. The Start up/Reset Vector is at 0x000. When CSU1221 is started or its program is reset, the Program Counter will point to Reset Vector. The Interrupt Vector is at 0x004. No matter what ISR is processed , the Program Counter will point to Interrupt Vector. Please see Figure 2-3.

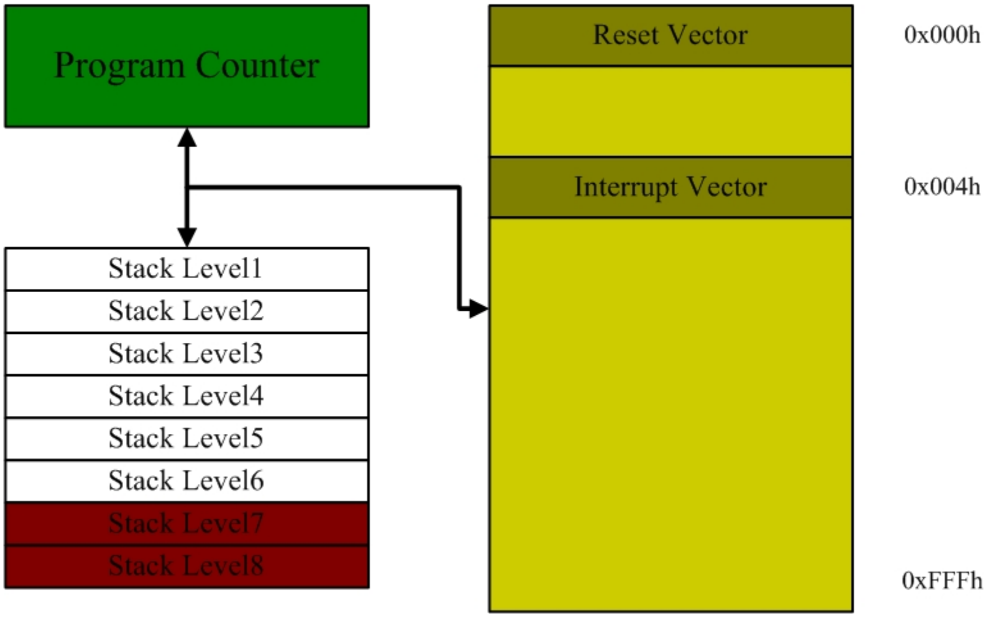


Figure 2-3 CSU1221 program memory structure

2. CSU1221 has a 384 byte SRAM for Data Memory. The data memory is partitioned into three parts. The area with address 00h~07h is reserved for system special registers, such as indirect address, indirect address pointer, status register, working register, interrupt flag, interrupt control register. The address 08h~7Fh areas are peripheral special registers, such as I/O ports, timer, ADC. The address 80h~17Fh areas are general data memory. Please see Table 2-2.

Table 2-2 CSU1221 Data memory structure

Data Memory	Start Address	End Address
System Special Registers	0x00	0x07
Peripheral Special Registers	0x08	0x7F
General Data Memory	0x80	0x17F

3. 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 2-4.

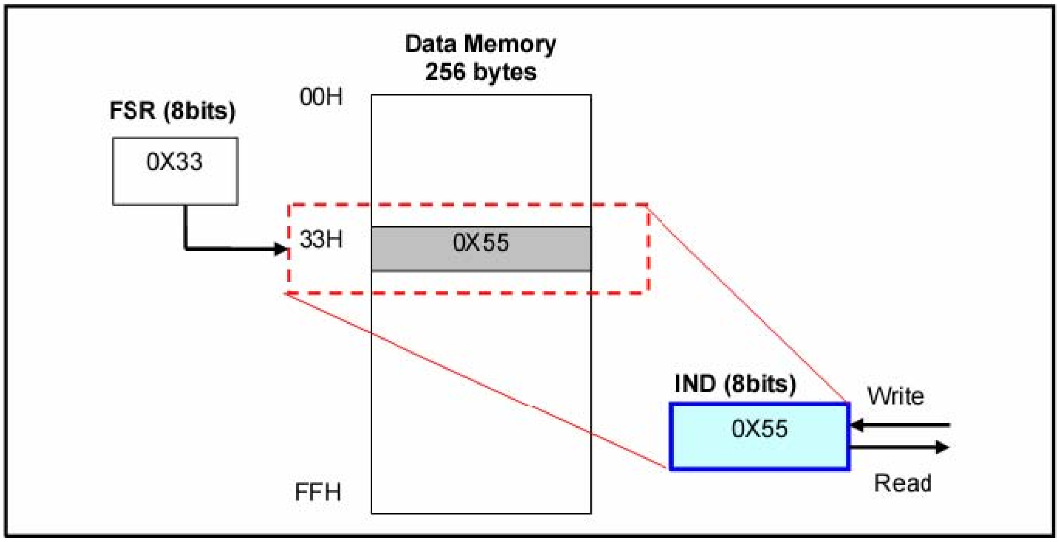


Figure 2-4 IND & FSR function description

4. Access of the last 128 bytes of Data Memory
1).Direct Access. Such as in MOVWF f, f can be 9 bit address.
2).Indirect addressing. Set IRP, then access by indirect addressing. It's important to note, the address stored in FSR is the result of subtracting 0x100 from actual address.

2.1.3 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-0	R/W-0	R/W-0
STATUS	IRP1	IRP0		PD	TO	DC	C	Z
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 IRP1: Page turning Flag of indirect addressing 1
1 =indirect addressing 1 access 0X100~0X17F
0 = indirect addressing 1 access 0X00~0XFF
Bit 6 IRP0: Page turning Flag of indirect addressing 0
1 =indirect addressing 0 access 0X100~0X17F
0 = indirect addressing 0 access 0X00~0XFF
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 ADDWF(C) and SUBWF(C)
(for borrow the polarity is reversed)
1 = If there is a carry out from the 4th bit of the result
0 = No carry out from the 4th 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

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

2.1.4 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	U-0	R/W-0	R/W-0	R/W-0
INTE	GIE			TMIE		ADIE	E1IE	E0IE
	Bit7	Bit6	Bit5	Bit4		Bit2	Bit1	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 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	U-0	R/W-0	R/W-0	R/W-0
INTF				TMIF		ADIF	E1IF	E0IF
	Bit7	Bit6	Bit5	Bit4		Bit2	Bit1	Bit0

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

2.2 Special Functional Register (SFR)

2.2.1 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.

Table 2-3 system register table

Address	Name	Bit7	Bit6	Bi5	Bi4	Bit3	Bit2	Bit1	Bit0	
00h	IND0	Use contents of FSR0 to address data memory								uuuuuuuu
01h	IND1	Use contents of FSR1 to address data memory								uuuuuuuu
02h	FSR0	Indirect data memory address pointer 0								uuuuuuuu
03h	FSR1	Indirect data memory address pointer 1								uuuuuuuu
04h	STATUS	IRP1	IRP0		PD	TO	DC	C	Z	00u00uuu
05h	WORK	WORK register								uuuuuuuu
06h	INTF				TMIF		ADIF	E1IF	E0IF	00000000
07h	INTE	GIE			TMIE		ADIE	E1IE	E0IE	00000000

2.2.2 Peripheral Special Registers

The Peripheral Special Registers are designed for Peripheral functions, such as I/O ports, timer, ADC, signal conditional network control register,. Please see Table 2-4 and the following Chapters for detailed description of these peripheral functions.

Table 2-4 peripheral special registers table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
0Dh	WDTCON	WTDEN					WTS[2:0]			0uuuu000
0Eh	TMOUT	TMOUT[7:0]								00000000
0Fh	TMCON	TRST				TMEN	INS[2:0]			1uuu0000
10h	ADOH	ADO[23:16]								00000000
11h	ADOL	ADO[15:8]								00000000
12h	ADOLL	ADO[7:0]								00000000
13h	ADCON						ADM[2:0]			uuuuu000
14h	MCK	M7_CK	M6_CK	M5_CK		M3_CK	M2_CK	M1_CK	M0_CK	000u0000
15h	PCK							S_BEEP		uuuuuu0u
18h	NETA	SINL[1:0]		SENS[1:0]						0000uuuu
1Ah	NETC	CHOPM[1:0]			BUFEN	ADG[1:0]		ADEN		00u0000u
1Ch	NETE	LDOS[1:0]				SILB[1:0]		ENLB		00uu000u
1Dh	NETF			ENVDDA			BGID[1:0]		ENVB	uu0uu000
1Fh	SVD								LBOUT	uuuuuuuu
20h	PT1	PT1[7:0]								00000000
21h	PT1EN	PT1EN[7:0]								00000000
22h	PT1PU	PT1PU[7:0]								00000000
23h	AIENB1				AIENB[4:0]					uuu00000
24h	PT2	PT2[7:0]								00000000
25h	PT2EN	PT2EN[7:0]								00000000
26h	PT2PU	PT2PU[7:0]								00000000
27h	PT2MR	BZEN		PM2EN	PM1EN	E1M[1:0]		E0M[1:0]		0u000000
28h	PT3	PT3[7:0]								00000000
29h	PT3EN	PT3EN[7:0]								00000000
2Ah	PT3PU	PT2PU[7:0]								00000000
2Bh	AOENB3	AOENB3 [7:0]								00000000
30h	PMD1H	PMD1[15:8]								00000000
31h	PMD1L	PMD1[7:0]								00000000
32h	PMD2H	PMD2[15:8]								00000000
33h	PMD2L	PMD2[7:0]								00000000
36h	PMCON	PDM2EN	PM2CS[2:0]			PDM1EN	PM1CS[2:0]			00000000
40h	DATA1	DATA1[7:0]								00000000
41h	DATA 2	DATA2[7:0]								00000000
42h	DATA 3	DATA3[7:0]								00000000
43h	DATA 4	DATA4[7:0]								00000000
44h	DATA 5	DATA5[7:0]								00000000
45h	DATA 6	DATA6[7:0]								00000000
46h	DATA 7	DATA7[7:0]								00000000
47h	DATA 8	DATA8[7:0]								00000000
48h	DATA 9	DATA9[7:0]								00000000
49h	DATA 10	DATA10[7:0]								00000000
4Ah	DATA 11	DATA11[7:0]								00000000
4Bh	DATA 12	DATA12[7:0]								00000000
4Ch	DATA 13	DATA13[7:0]								00000000
4Dh	DATA 14	DATA14[7:0]								00000000
4Eh	DATA 15	DATA15[7:0]								00000000
4Fh	DATA 16	DATA16[7:0]								00000000
50h	DATA 17	DATA17[7:0]								00000000
51h	DATA 18	DATA18[7:0]								00000000
52h	DATA 19	DATA19[7:0]								00000000
53h	DATA 20	DATA20[7:0]								00000000
54h	DATA 21	DATA21[7:0]								00000000
55h	DATA 22	DATA22[7:0]								00000000

2.3 Clock System

2.3.1 Oscillator State

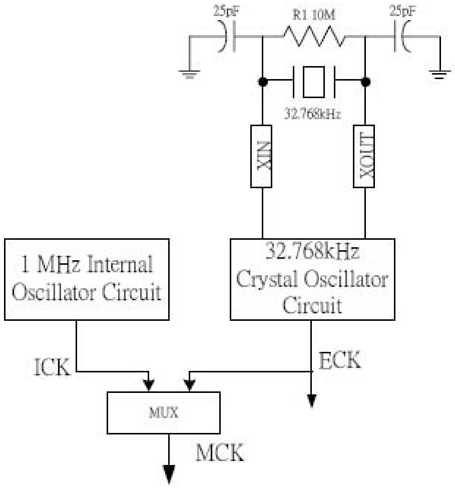

25pF
R1 10M
25pF
32.768kHz
XIN
XOUT
1 MHz Internal Oscillator Circuit
32.768kHz Crystal Oscillator Circuit
ICK
ECK
MUX
MCK

Figure 2-5 CSU1221 oscillator state block

Table 2-5 CSU1221 clock system register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
14h	MCK	M7_CK	M6_CK	M5_CK		M3_CK	M2_CK	M1_CK	M0_CK	00000000

There are two clock sources in CSU1221. 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 2-5.

There are 2 clock signals working in CSU1221: MCK and CLK. Users should use Table 2-6 and 2-7 to setup MCK and CLK based on the M0_CK, M1_CK and M3_CK.

Table 2-6 MCK selection table

M3_CK	M0_CK	MCK
X	0	ICK
0	1	ECK/4
1	1	ECK/8

To enable the internal and external oscillators, users need to set the right values for M7_CK and M6_CK as shown in Table 2-8. If users execute the sleep instruction to make CSU1221 enter the SLEEP mode, both the internal oscillators and the external oscillator will be disabled.

Table 2-7 oscillator state selection table

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

X means “don’t care”

2.3.2 CPU Instruction Cycle

Table 2-8 CSU1221 CPU instruction cycle register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value on power on restet
14h	MCK	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(In order to maintain a stable oscillator output, add a ‘NOP’ after changing the oscillator).

Table 2-9 instruction cycle selection table

M3_CK	M2_CK	M1_CK	M0_CK	MCK(KHz)		Instruction cycle(KHz)	
0	0	0	0	ICK	1000	MCK/8	125
0	0	0	1	ECK/4	8.192	MCK/8	1.024
0	0	1	0	ICK	1000	MCK/16	62.5
0	0	1	1	ECK/4	8.192	MCK/16	0.512
0	1	0	0	ICK	1000	MCK/2	500
0	1	0	1	ECK/4	8.192	MCK/2	4.096
0	1	1	0	ICK	1000	MCK/4	250
0	1	1	1	ECK/4	8.192	MCK/4	2.048
1	0	0	0	ICK	1000	MCK/8	125
1	0	0	1	ECK/8	4.096	MCK/8	0.512
1	0	1	0	ICK	1000	MCK/16	62.5
1	0	1	1	ECK/8	4.096	MCK/16	0.256
1	1	0	0	ICK	1000	MCK/2	500
1	1	0	1	ECK/8	4.096	MCK/2	2.048
1	1	1	0	ICK	1000	MCK/4	250
1	1	1	1	ECK/8	4.096	MCK/4	1.024

2.3.3 ADC Clock

Table 2-10 ADC sample frequency selection register

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
MCK					M3_CK		M1_CK	M0_CK
14h	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

IN CSU1221,the ADC sample frequency is used to select signal.user can set the sign bit of register M0_CK,M1_CK,M3_CK to change the sample clock.

Table 2-11 ADC sample frequency selection table

M3_CK	M1_CK	M0_CK	MCK(KHz)		ADCF(KHz)	
0	0	0	ICK	1000	MCK/16	62.5
0	0	1	ECK/4	8.192	MCK/16	0.512
0	1	0	ICK	1000	MCK/32	31.25
0	1	1	ECK/4	8.192	MCK/32	0.256
1	0	0	ICK	1000	MCK/16	62.5
1	0	1	ECK/8	4.096	MCK/16	0.256
1	1	0	ICK	1000	MCK/32	31.25
1	1	1	ECK/8	4.096	MCK/32	0.128

Table 2-12 ADC Ourput Rate Select Register table

Property	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
ADCON						ADM[2:0]		
13h	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Table 2-13 ADC Ourput Rate Select table

ADM[2:0]	ADC Output Rate
000	ADCF/128
001	ADCF/256
010	ADCF/512
011	ADCF/1024
100	ADCF/2048
101	ADCF/4096
110	ADCF/8192
111	ADCF/8192

2.3.4 Beeper Clock

Table 2-14 beeper clock register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
14h	MCK	M7_CK	M6_CK	M5_CK		M3_CK	M2_CK	M1_CK	M0_CK	00000000
15h	PCK							S_BEEP		uuuuuu0u

CSU1221 has a Beeper Clock which is used as the buzzer source. User could change the Beeper clock frequency by setting M0_CK, M1_CK, M3_CK and S_BEEP register flags according to Table2-13, Table 2-14 and Table 2-15.

Table 2-15 the beeper clock selection table

M3_CK	M2_CK	M1_CK	S_BEEP	MCK (KHz)		Beeper Clock (KHz)	
0	0	0	0	ICK	1000	MCK/256	3.90625
0	0	0	1	ICK	1000	MCK/512	1.953125
0	0	1	0	ECK/4	8.192	MCK/256	0.064
0	0	1	1	ECK/4	8.192	ECK/8	4.096
0	1	0	0	ICK	1000	MCK/1024	0.9765625
0	1	0	1	ICK	1000	MCK/2048	0.48828125
0	1	1	0	ECK/4	8.192	MCK/1024	0.016
0	1	1	1	ECK/4	8.192	ECK/8	4.096
1	0	0	0	ICK	1000	MCK/256	3.90625
1	0	0	1	ICK	1000	MCK/512	1.953125
1	0	1	0	ECK/8	4.096	MCK/256	0.016
1	0	1	1	ECK/8	4.096	ECK/8	4.096
1	1	0	0	ICK	1000	MCK/1024	0.9765625
1	1	0	1	ICK	1000	MCK/2048	0.48828125
1	1	1	0	ECK/8	4.096	MCK/1024	0.004
1	1	1	1	ECK/8	4.096	ECK/8	4.096

2.3.5 TMCLK -- Timer Module Input Clock

TMCLK is the clock for CSU1221 Timer Module. Users can use Table 5-20 to choose TMCLK frequency by setting the values for M5_CK.

Table 2-16 TMCLK selection table

M5_CK	M3_CK	M1_CK	M0_CK	MCK (KHz)		TMCLK(KHz)	
0	0	0	0	ICK	1000	MCK/1024	976.5625
0	0	0	1	ECK/4	8.192	MCK/1024	16
0	0	1	0	ICK	1000	MCK/4096	244.10625
0	0	1	1	ECK/4	8.192	ECK/4096	4
0	1	0	0	ICK	1000	MCK/1024	976.5625
0	1	0	1	ECK/8	4.096	MCK/1024	4
0	1	1	0	ICK	1000	MCK/4096	244.10625
0	1	1	1	ECK/8	4.096	ECK/4096	1
1	0	0	0	ICK	1000	ECK/32	1024
1	0	0	1	ECK/4	8.192	ECK/32	1024
1	0	1	0	ICK	1000	ECK/32	1024
1	0	1	1	ECK/4	8.192	ECK/32	1024
1	1	0	0	ICK	1000	ECK/32	1024
1	1	0	1	ECK/8	4.096	ECK/32	1024
1	1	1	0	ICK	1000	ECK/32	1024
1	1	1	1	ECK/8	4.096	ECK/32	1024

2.4 Timer Module

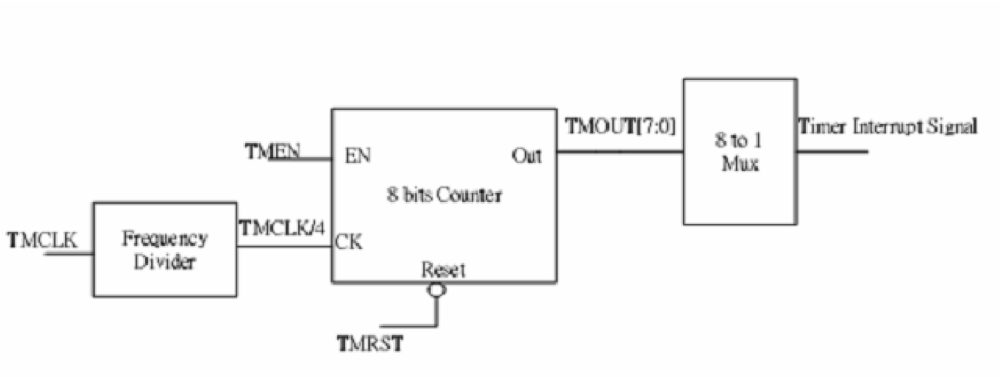


Figure 2-6 CSU1221 timer module function block

The input of Timer Module is TMCLK. CSU1221 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)

Table 2-17 timer module interrupt register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
06H	INTF				TMIF					00000000
07H	INTE	GIE			TMIE					00000000
0EH	TMOUT	TMOUT[7:0]								00000000
0FH	TMCON	TRST				TMEN	INS[3:0]			1uuu0000

Operation:

1. Setup the TMCLK for Timer module input
2. Setup the INS[2:0] to select timer interrupt source. Please see Table 2-19.
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 2-18 timer selection table

INS[2:0]	Interrupt Source	Time at TMCLK=1024(ECK/32)
000	TMOUT[0]	1/128 s
001	TMOUT[1]	1/64 s
010	TMOUT[2]	1/32 s
011	TMOUT[3]	1/16 s
100	TMOUT[4]	1/8 s
101	TMOUT[5]	1/4 s
110	TMOUT[6]	1/2 s
111	TMOUT[7]	1 s

2.5 Watch Dog Timer

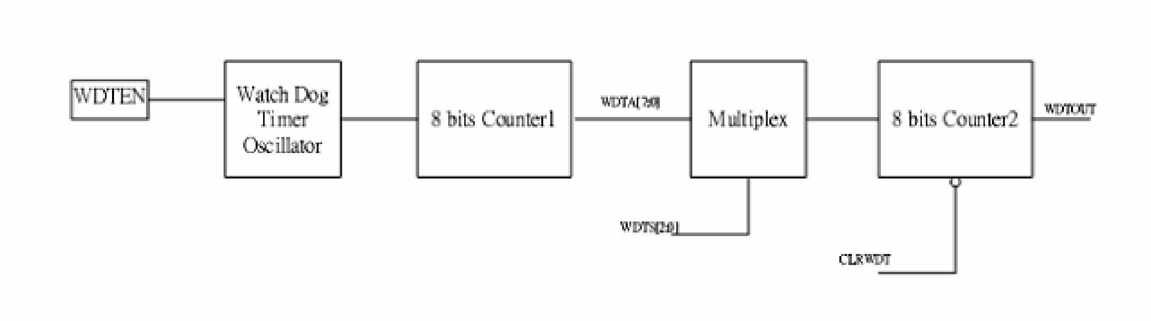


Figure 2-7 watch dog timer function block

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 CSU1221 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 2-19 watch dog timer register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
04H	STATUS					TO				00u00uuu
0DH	WDTCON	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.

Table 2-20 watch dog timer register table

WDTS[2:0]	Clock	Time
000	WDTIN[0]	21.8s
001	WDTIN[1]	10.9s
010	WDTIN[2]	5.5s
011	WDTIN[3]	2.7s
100	WDTIN[4]	1.4s
101	WDTIN[5]	0.68s
110	WDTIN[6]	0.34s
111	WDTIN[7]	0.17s

2.6 I/O Port

Table 2-21 CSU1221 I/O port register table

Ad dre ss	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
06h	INTF							E1IF	E0IF	00000000
07h	INTE	GIE						E1IE	E0IE	00000000
20h	PT1	PT1[7:5]								00000000
21h	PT1EN	PT1EN[7:5]								00000000
22h	PT1PU	PT1PU[7:5]								00000000
24h	PT2	PT2[7:6]				PT2[3:0]				00000000
25h	PT2EN	PT2EN[7:6]				PT2EN[3:0]				00000000
26h	PT2PU	PT2PU[7:6]				PT2PU[3:0]				00000000
27h	PT2MR	BZEN		PM2EN	PM1EN	E1M[1:0]		E0M[1:0]		00000000
28h	PT3		PT3[6:0]							00000000
29h	PT3EN		PT3EN[6:0]							00000000
2Ah	PT3PU		PT3PU[6:0]							00000000
2Bh	AOENB3		AOENB3 [6:0]							00000000

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 GPIO in CSU1221 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-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT1	PT1[7:5]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1[7:5]: GPIO Port 1 data flag
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

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[7:5]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1EN [7:5]: GPIO Port 1 Input / Output control flag
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

Register PT1PUat 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[7:5]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1PU [7:5]: GPIO Port 1 Pull up resistor enable flag
PT1PU[7] = GPIO Port 1 bit 7 control flag ; 0 = Pull up resistor is disconnected, 1 = with Pull up resistor
PT1PU[6] = GPIO Port 1 bit 6 control flag ; 0 = Pull up resistor is disconnected, 1 = with Pull up resistor

PT1PU[5] = GPIO Port 1 bit 5 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 PT2 at address 24H

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
PT2	PT2[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	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

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	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	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

Register PT2PU at address 26H

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
PT2PU	PT2PU[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	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

Register PT2MR at address 27H

Property	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT2MR	BZEN		PM2EN	PM1EN	E1M[1:0]		E0M[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 BZEN: Buzzer enable flag
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 5 PM2EN: PDM Module enable flag
1 = PDM Module is enabled, GPIO Port 2 bit 3 is defined as PDM output.
0 = PDM Module is disabled, GPIO Port 2 bit 3 is defined as GPIO.

Bit 4 PM1EN: PDM Module enable flag
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
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
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

Register PT3 at address 28H

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
PT3		PT3[6:0]						
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT3[6:0]: GPIO Port 3 data flag
PT3[6] = GPIO Port 3 bit 6 data flag
PT3[5] = GPIO Port 3 bit 5 data flag
PT3[4] = GPIO Port 3 bit 4 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

Register PT3EN at address 29H

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
PT3EN		PT3EN[6:0]						
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT3EN [6:0]: GPIO Port 3 Input / Output control flag
PT3EN[6] = GPIO Port 3 bit 6 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT3EN[5] = GPIO Port 3 bit 5 I/O control flag ; 0 = defined as input port, 1 = defined as output port
PT3EN[4] = GPIO Port 3 bit 4 I/O control flag ; 0 = defined as input port, 1 = defined as output port
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

Register PT3PU at address 2AH

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
PT3PU		PT3PU[6:0]						
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT3PU [6:0]: GPIO Port 3 Pull up resistor enable flag

PT3PU[6] = GPIO Port 3 bit 6 control flag ; 0 = Pull up resistor is disconnect, 1 = with Pull up resistor
PT3PU[5] = GPIO Port 3 bit 5 control flag ; 0 = Pull up resistor is disconnect, 1 = with Pull up resistor
PT3PU[4] = GPIO Port 3 bit 4 control flag ; 0 = Pull up resistor is disconnect, 1 = with Pull up resistor
PT3PU[3] = GPIO Port 3 bit 3 control flag ; 0 = Pull up resistor is disconnect, 1 = with Pull up resistor
PT3PU[2] = GPIO Port 3 bit 2 control flag ; 0 = Pull up resistor is disconnect, 1 = with Pull up resistor
PT3PU[1] = GPIO Port 3 bit 1 control flag ; 0 = Pull up resistor is disconnect, 1 = with Pull up resistor
PT3PU[0] = GPIO Port 3 bit 0 control flag ; 0 = Pull up resistor is disconnect, 1 = with Pull up resistor

Register AOENB3 at address 2BH

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
AOENB3	AOENB[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 AOENB3[6:0]: GPIO Port 3 Analog / Digital control flag

AOENB3[6] = GPIO Port 3 bit 6 D/A flag ; 0 = defined as Analog channel, 1 = defined as Digital channel
AOENB3[5] = GPIO Port 3 bit 5D/A flag ; 0 = defined as Analog channel, 1 = defined as Digital channel
AOENB3[4] = GPIO Port 3 bit 4 D/A flag ; 0 = defined as Analog channel, 1 = defined as Digital channel
AOENB3[3] = GPIO Port 3 bit 3 D/A flag ; 0 = defined as Analog channel, 1 = defined as Digital channel
AOENB3[2] = GPIO Port 3 bit 2 D/A flag ; 0 = defined as Analog channel, 1 = defined as Digital channel
AOENB3[1] = GPIO Port 3 bit 1 D/A flag ; 0 = defined as Analog channel, 1 = defined as Digital channel
AOENB3[0] = GPIO Port 3 bit 0 D/A flag ; 0 = defined as Analog channel, 1 = defined as Digital channel

2.6.1 Digital I/O Port : PT1[7:5]

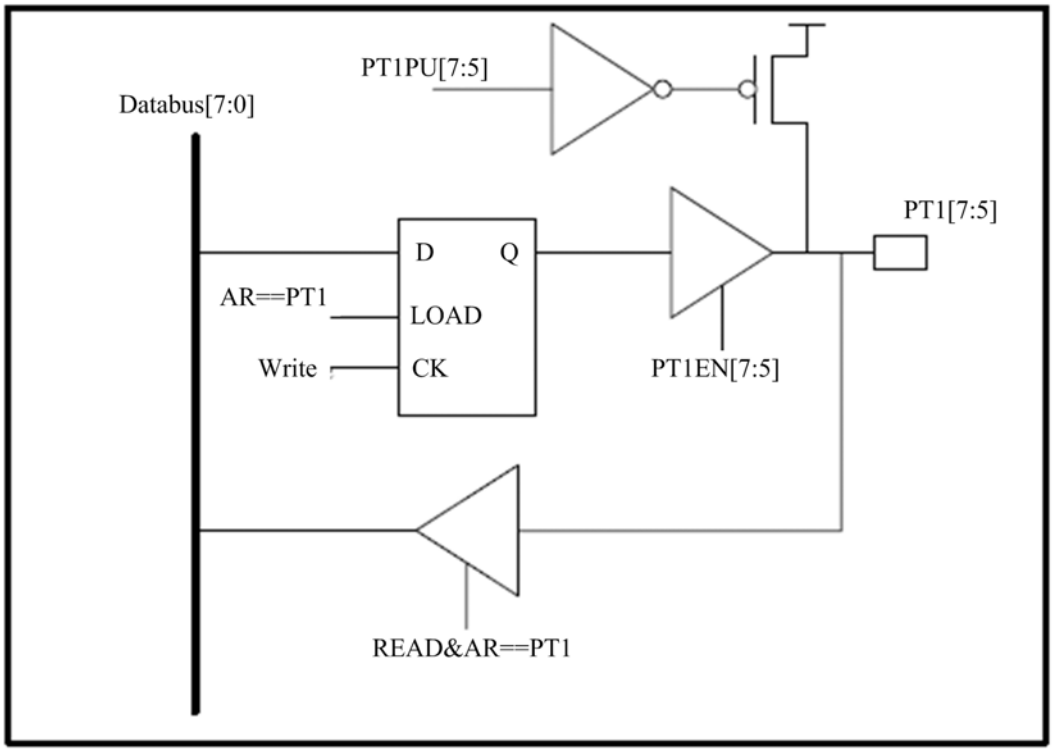


Figure 2-8 PT1[7:5] function block

GPIO Port 1 (PT1[7:5]) function block is shown in Figure 2-8. The main function of the GPIO is for data exchange between the Data bus and the ports. Users could control the PT1EN[7:5] 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 (PT1[7:5]) could only be the general I/O ports.

- Output

CSU1221 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 output while the Write signal and AR (CSU1221 internal device address pointer) is pointed to PT1.

- Pull up resistor

CSU1221 embeds an internal pull up resistor function in PT1 with about 100 KΩ(The pull up current is about 30uA). Users could control the PT1PU[7:5] 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). Remember to disable PT1PU before program falls into Sleep mode.

Table 2-22 PT1 register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
20h	PT1	PT1[7:5]								00000000
21h	PT1EN	PT1EN[7:5]								00000000
22h	PT1PU	PT1PU[7:5]								00000000

Read data Operation:

1. Clear the PT1EN[n] 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. 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[4:0] to let the PT1 be floating.
2. Parallel a small resistor (about 10 KΩ) between ports and VDD to increase the possible output current when the PT1PU[n] is set.

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

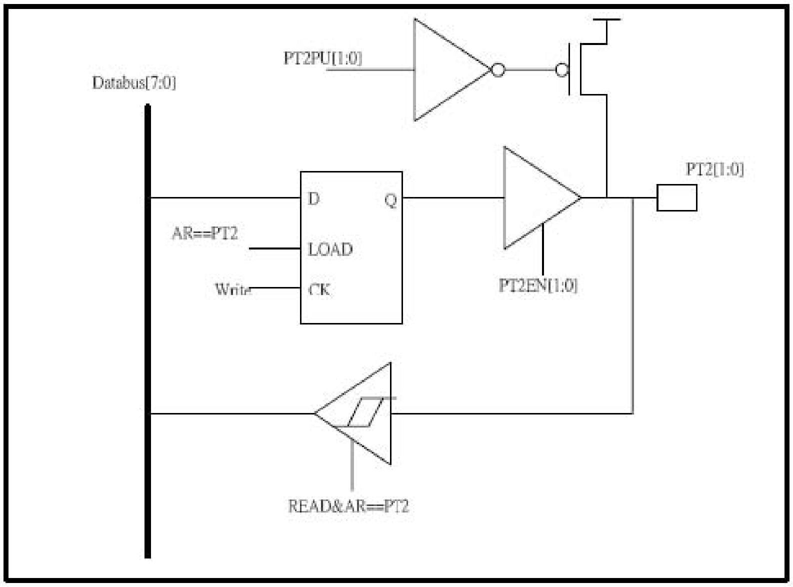


Figure 2-9 PT2[0] PT2[1] function block

GPIO Port 2 Bit1 and Bit 0 (PT2[1:0]) function block is shown in Figure 2-9. 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] 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.

- Output

CSU1221 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 (CSU1221 internal device address pointer) is pointed to PT2.

- Pull up resistor

CSU1221 embeds an internal pull up resistor function in PT2 with about 100 KΩ.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).

Table 2-23 PT2 register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
06h	INTF							E1IF	E0IF	00000000
07h	INTE	GIE						E1IE	E0IE	00000000
24h	PT2	PT2[7:0]								uuuuuuuu
25h	PT2EN	PT2EN[7:0]								00000000
26h	PT2PU	PT2PU[7:0]								00000000
27h	PT2MR					E1M[1:0]		E0M[1:0]		00000000

Read data Operation:

1. Clear the PT2EN[n] 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.

External Interrupt Operation (negative edge trigger for example):

1. Clear the PT2EN[n] register flags. The PT2[n] will be defined as an input port.
2. Set the PT2PU[n] register. The PT2[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".

Notice Operation:

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

2.6.3 Digital I/O Port or PDM Output : PT2[3:2]

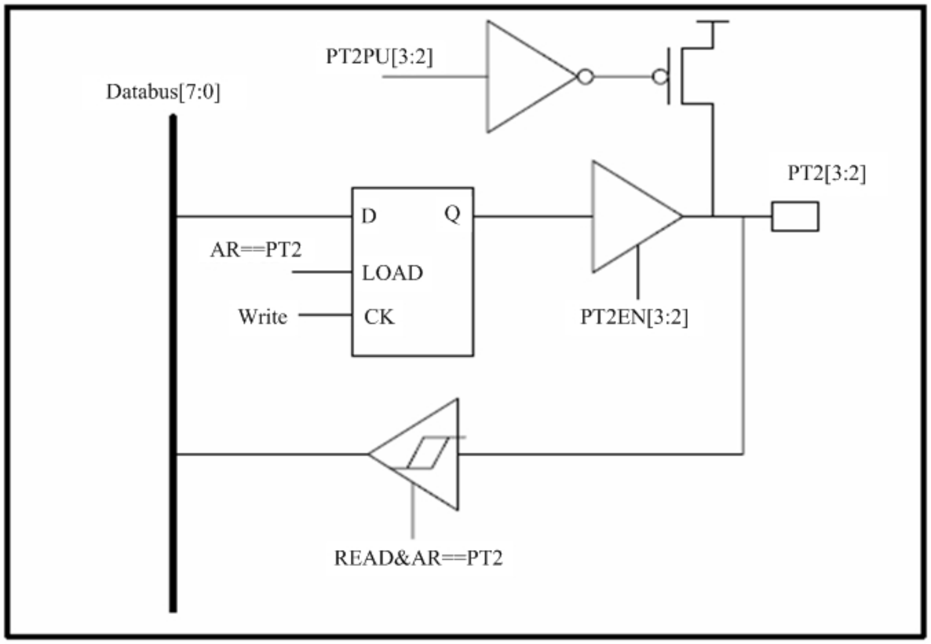


Figure 2-10 PT2[3:2] function block

GPIO Port 2 Bit2 and Bit3 (PT2[2]、PT2[3]) function block is shown in Figure 2-10. The main function of the GPIO is input and output data between the Data bus and the ports. User could control the PT2EN[3:2] 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 and Bit3 (PT2[2]、PT2[3]) could be the PDM (Pulse Density Modulator) output port or be the general I/O port. User should setup PM1EN and PM2EN register flag to decide if the PDM is enabled.

- Output

CSU1221 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 (CSU1221 internal device address pointer) is pointed to PT2.

- Pull up resistor

CSU1221 embeds an internal pull up resistor function in PT2 with about 100 KΩ.Users could control the PT2PU[3:2] 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).

Table 2-24 PT2 register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
24h	PT2	PT2[7:0]								uuuuuuuu
25h	PT2EN	PT2EN[7:0]								00000000
26h	PT2PU	PT2PU[7:0]								00000000
27h	PT2MR	BZEN		PM2EN	PM1EN					00000000

Read data Operation:

1. Clear the PT2EN[n] 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.

2.6.4 Digital I/O Port : PT2[6:4]

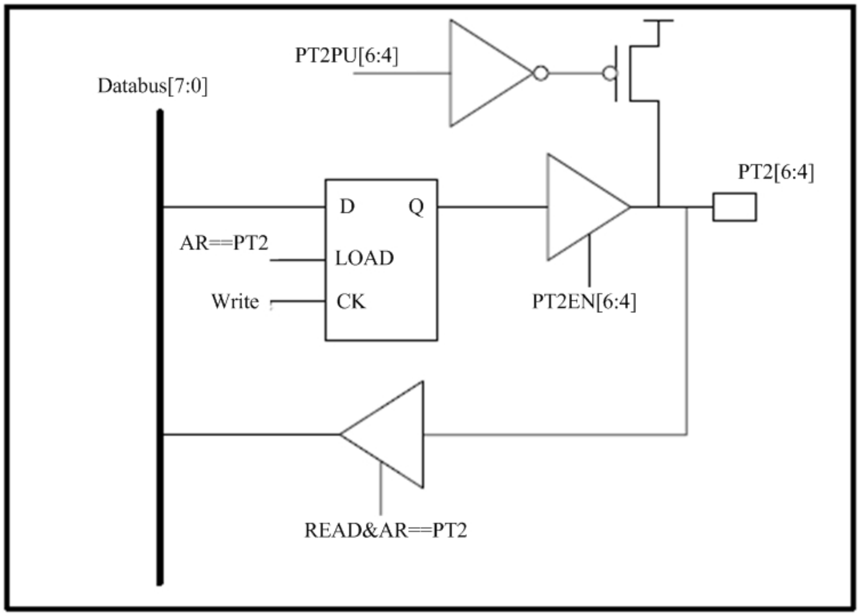


Figure 2-11 PT2[6:4] function block

GPIO Port 2 Bit 6 、Bit 5 and Bit 4 (PT2[6:4]) function block is shown in Figure 2-11. 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 are explained as follows:

- Input

GPIO Port 2 Bit 6 、Bit 5 and Bit 4 (PT2[6:4]) could only be the general I/O ports.

- Output

CSU1221 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 (CSU1221 internal device address pointer) is pointed to PT2.

- Pull up resistor

CSU1221 embeds an internal pull up resistor function in PT2 with about 100 KΩ.Users could control the PT2PU[6:4] 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).

Table 2-25 PT2 register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
24h	PT2	PT2[7:0]								uuuuuuuu
25h	PT2EN	PT2EN[7:0]								00000000
26h	PT2PU	PT2PU[7:0]								00000000
27h	PT2MR	BZEN		PM1EN	PM2EN					00000000

Read data Operation:

1. Clear the PT2EN[n] 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 10 KΩ) between ports and VDD to increase the possible output current when the PT2PU[n] is set.

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

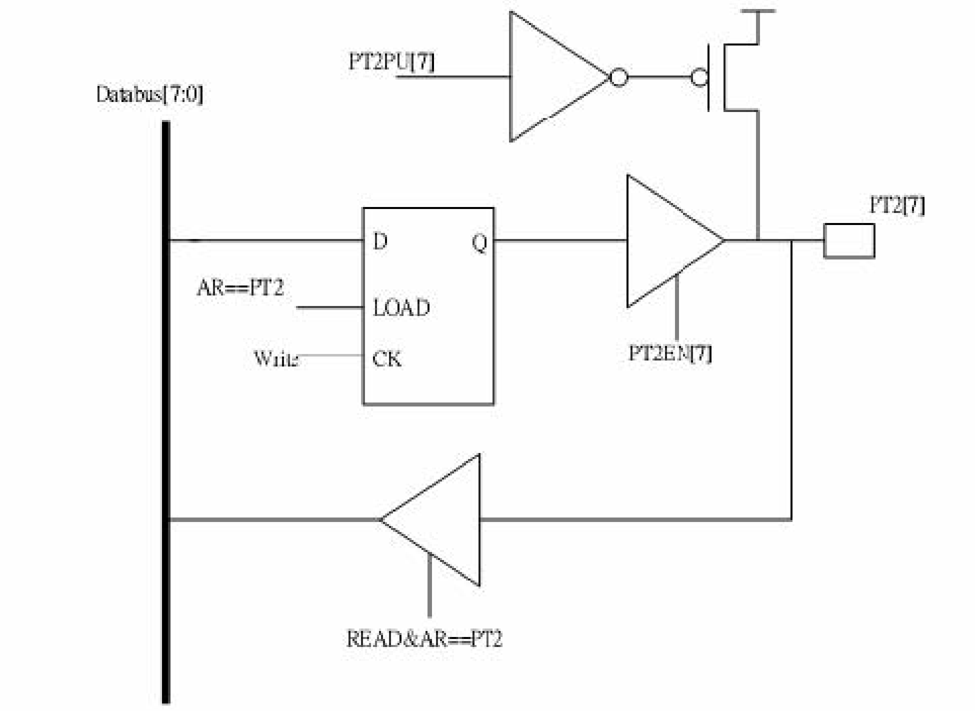


Figure 2-12 PT2[7] function block

GPIO Port 2 Bit 7 function block is shown in Figure 2-12. The main function of the GPIO is input and output data between the Data bus and the ports. Users could control the PT2EN[7] register flags to decide the input output direction. The input and output function are explained as follows:

- Input

GPIO Port 2 Bit7 (PT2[7]) 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.

- Output

CSU1221 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 (CSU1221 internal device address pointer) is pointed to PT2.

- Pull up resistor

CSU1221 embeds an internal pull up resistor function in PT2 with about 100 KΩ.Users could control the PT2PU[7] 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).

Table 2-26 PT2[7] register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
24h	PT2	PT2[7:0]								uuuuuuuu
25h	PT2EN	PT2EN[7:0]								00000000
26h	PT2PU	PT2PU[7:0]								00000000

Read data Operation

1. Clear the PT2EN[n] 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 10 KΩ) between ports and VDD to increase the possible output current when the PT2PU[n] is set.

2.6.6 Digital I/O Port: PT3[6:0]

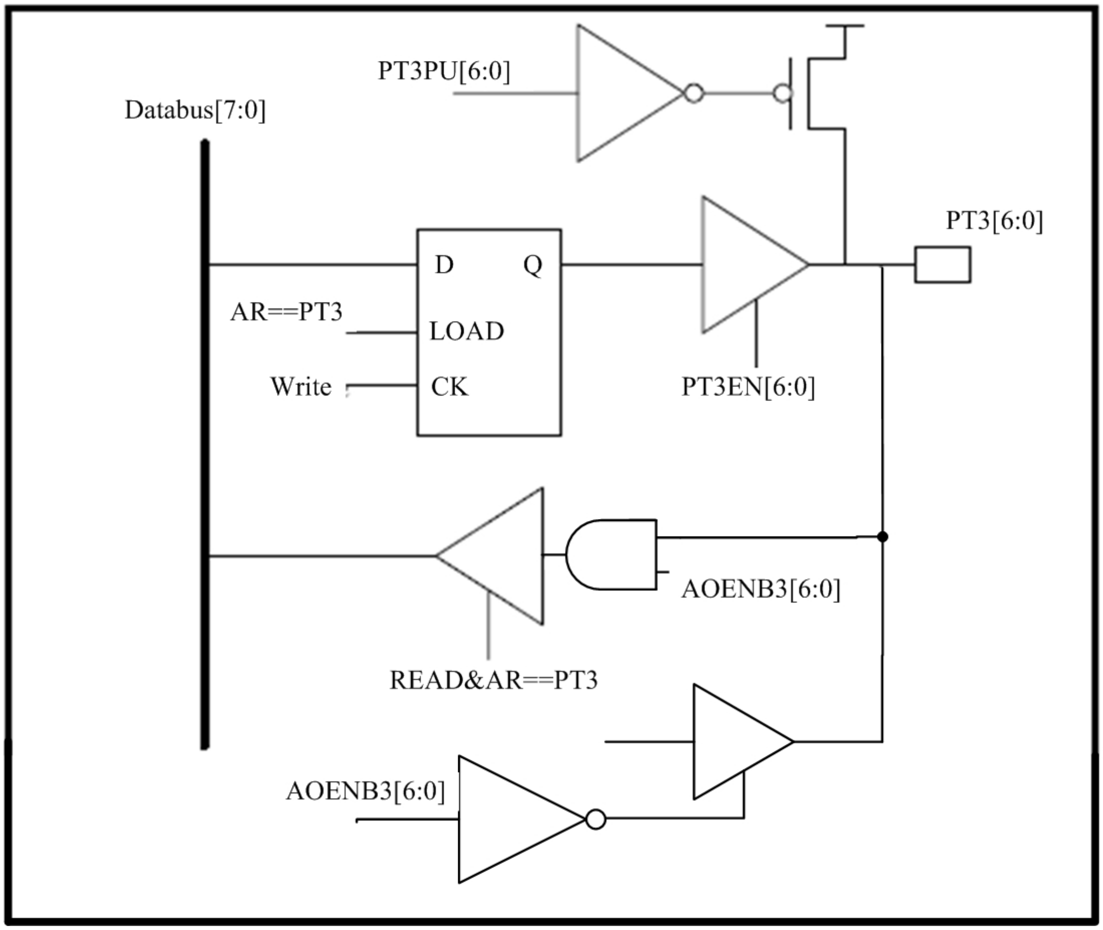


Figure 2-13 PT3[7:0] function block

GPIO Port 3 function block is shown in Figure 2-13. The main function of the GPIO is input and output data between the Data bus and the ports. Users could control the PT3EN register flags to decide the input output direction. Users could control the AOENB3 to decide the GPIO. The input and output function are explained as follows:

- Input

GPIO Port 3 can be used to input data. Users should clear the PT3EN register flags.

- Output

GPIO Port 3 could be input data or be output analog signal (only output segment signal). User should setup AOENB3 to decide the signal type. If AOENB3 is set, the AND gate embedded in the GPIO Port3 will allow the digital data to connect to the data bus. 0therwise, the Output signals will be defined as analog signals.
CSU1221 sends the digital data out by an embedded D Flip Flop. When the program sends data out through PT3, the data will be sent to data bus first, and then the D Flip Flop will latch the signal for PT3 output while the Write signal and AR (CSU1221 internal device address pointer) is pointed to PT3.

- Pull up resistor

CSU1221 embeds an internal pull up resistor function in PT3with about 100KΩ.Users could
control the PT3PUregister 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).

Table 2-27 PT3 register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
28h	PT3	PT3[6:0]								00000000
29h	PT3EN	PT3EN[6:0]								00000000
2Ah	PT3PU	PT3PU[6:0]								00000000
2Bh	AOENB3	AOENB3[6:0]								00000000

Read data Operation

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

Write data Operation:

1. Set the PT3EN[n] register flags. The PT3[n] will be defined as an output port.
2. Set the PT3PU[n] register as required. The PT3[n] will be connected to an internal pull up resistor.
3. Set the AOENB3[n] register flags if the output signals are digital signals.(n = 6 to 0)
4. Clear the AOENB3[n] register flags if the output signals are analog signals. (n =6 to 0)
5. Set the PT3[n] to output the data. The embedded D Flip Flop will latch the data till PT3[n] is changed.

2.7 Data memory

Data memory Data1~Data22 can be used as data write/read register，the Value at Power on Reset is 0。

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
40h	DATA1	DATA1								00000000
41h	DATA2	DATA2								00000000
42h	DATA3	DATA3								00000000
43h	DATA4	DATA4								00000000
44h	DATA5	DATA5								00000000
45h	DATA6	DATA6								00000000
46h	DATA7	DATA7								00000000
47h	DATA8	DATA8								00000000
48h	DATA9	DATA9								00000000
49h	DATA10	DATA10								00000000
4Ah	DATA11	DATA11								00000000
4Bh	DATA12	DATA12								00000000
4Ch	DATA13	DATA13								00000000
4Dh	DATA14	DATA14								00000000
4Eh	DATA15	DATA15								00000000
4Fh	DATA16	DATA16								00000000
50h	DATA17	DATA17								00000000
51h	DATA18	DATA18								00000000
52h	DATA19	DATA19								00000000
53h	DATA20	DATA20								00000000
54h	DATA21	DATA21								00000000
55h	DATA22	DATA22								00000000

2.8 Halt and Sleep Modes

CSU1221 supports low power working mode. When the user want CSU1221 to do nothing and just stand by, CSU1221 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 less than 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        NETC        ;As Reset state
CLRF        NETE        ;As Reset state
CLRF        NETF        ;As Reset state
CLRF        PT1PU       ; Pull up resistor is disconnected
MOVLW       0FFh
MOVWF       PT1EN       ; PT1[7:0] is assigned to be output ports.
CLRF        PT1         ; Set PT1 Output Low
MOVLW       0Ih
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 CSU1221 into Sleep mode
NOP                     ; Guarantee that the program works normally when CPU wakes up.
```

2.9 Power System

2.9.1 Voltage Regulator

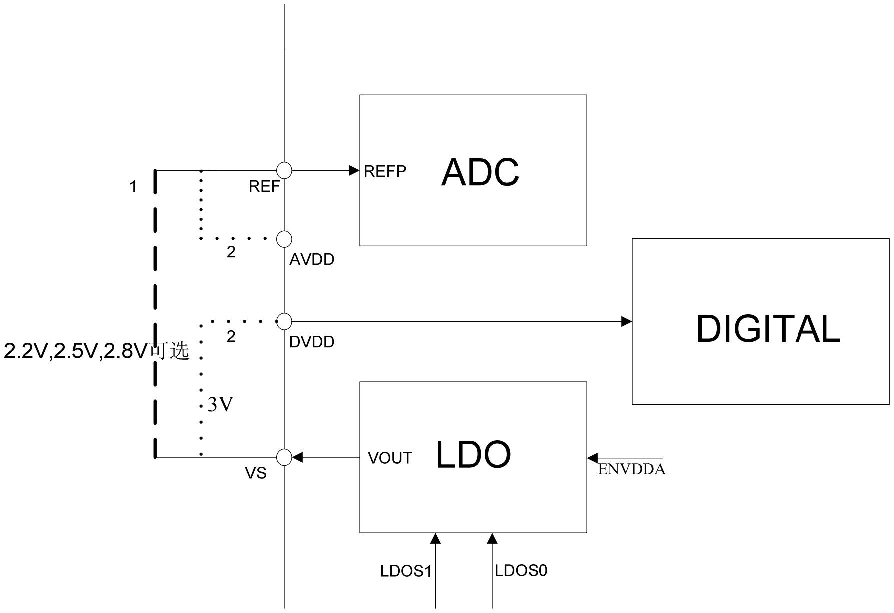


Figure 2-14 Voltage regulator

When input power supply is less than 3.3V, LDO could employ scheme 1 mainly which generates VS to supply reference voltage for the sensor and ADC. There are output voltage as 2.2V、2.5V、2.8V and 3V to choose through selecting LDOS1,LDOS0. When input power supply is 5V, LDO could employ scheme 2 mainly. LDO output voltage 3V will be the power supply of digital part on the chip while the ADC and sensor choose AVDD for the reference voltage. ENVDDA is LDO-enable signal. LDO control register flag is ENVDDA, LDOS1,LDOS0 and the output is VS.

Table 2-28 voltage regulator register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
1Dh	NETF			ENVDDA					ENVB	00000000

Bit5 ENVDDA：LDO enable flag
ENVDDA=1：LDO enable
ENVDDA=0：LDO disable

Bit0 ENVB：BandGap enable flag
ENVB=1：BandGap enable
ENVB=0：BandGap disable, analog module (such as ADC) is disable

Table 2-29 voltage source register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
1Ch	NETE	LDOS[1:0]								00000000

Bit7~6 LDOS[1:0]：VS voltage selector

LDOS[1:0] 00 VS=3
LDOS[1:0] 01 VS=2.8
LDOS[1:0] 10 VS=2.5
LDOS[1:0] 11 VS=2.2

Operation:

1. Set the ENVDDA flag. Low level effective. and high level shutoff. Low level is default.
2. Set the ENVB flag. Low level effective. and high level shutoff. Low level is default
3. Set LDOS[1:0] to select the value of VS

2.9.2 Low Battery Comparator

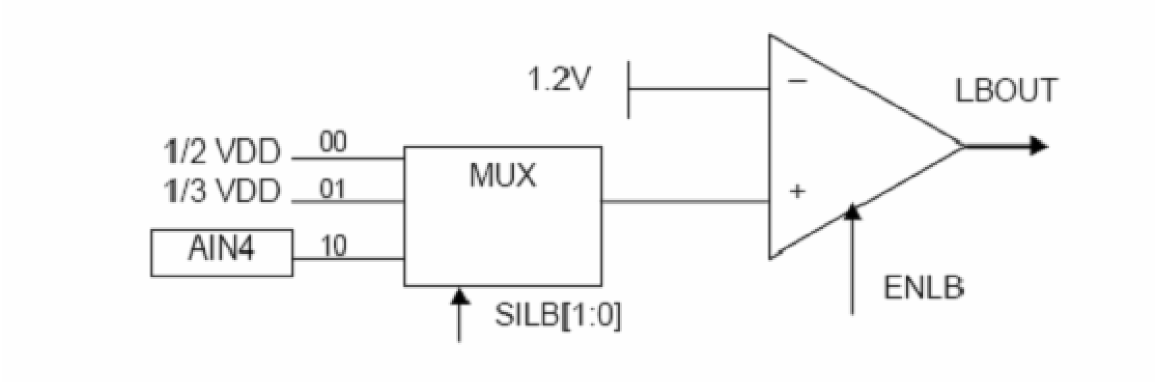


Figure 2-15 low battery comparator function block

Low Battery Comparator is used for VDD low voltage detection. CSU1221 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 2-15.

Table 2-30 low battery comparator register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
1Ch	NETE					SILB[1:0]		ENLB		00000000
1Fh	SVD								LBOUT	uuuuuuuu

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

Table 2-31 low battery comparator voltage detection selection table

SILB[1:0]	Detection Voltage	Condition	
00	1/2VDD	VDD<2.4 V	LBOUT＝0
01	1/3VDD	VDD<3.6V	LBOUT＝0
10	AIN4	AIN4<1.2V	LBOUT＝0

2.10 Analog to Digital Converter (ADC)

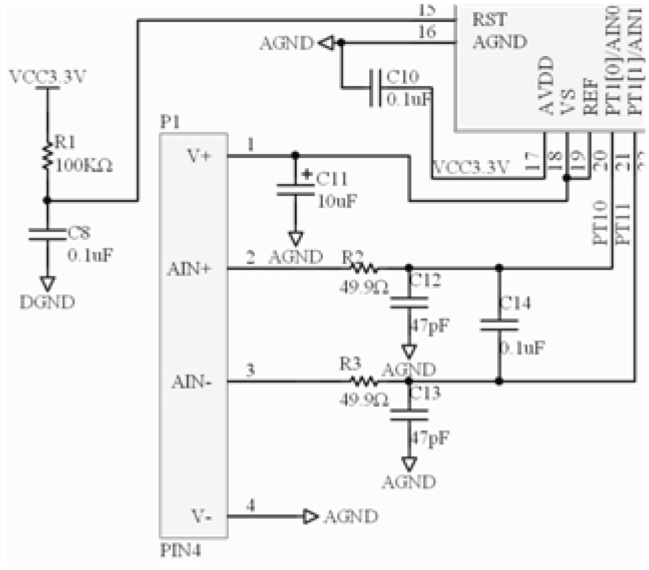


Figure 2-16 The example of using internal reference voltage

Please see Figure2-16 .Set ENVDDA (the register of NETF) to High, so that VS can be set to voltage source ,connect the VS and REF,use VS to provide power for sensor and reference.

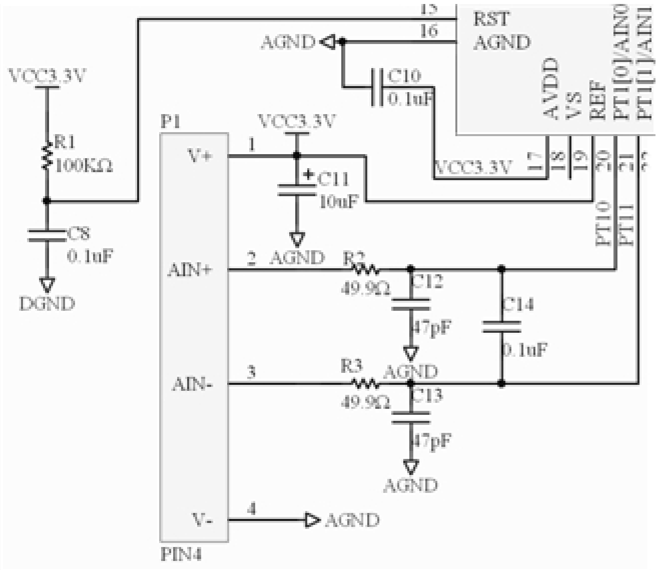


Figure 2-17 The example of using external reference voltage

Please see Figure2-17 .Set ENVDDA (the register of NETF) to Low, so that VS is closed. connect the REF and external voltage source ,use external voltage source to provide power for sensor and reference.

Table 2-32 ADC function register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
06h	INTF						ADIF			00000000
07h	INTE	GIE					ADIE			00000000
10h	ADOH	ADO[23:16]								00000000
11h	ADOL	ADO[15:8]								00000000
12h	ADOLL	ADO[7:0]								
13h	ADCON						ADM[2 :0]			uuuu0000
14h	MCK							M1_CK		00000000
18h	NETA	SINL[1:0]		SENS[1:0]						00000000
1Ah	NETC	CHOPM[1:0]				ADG[1:0]		ADEN		00000000
1Dh	NETF						BGID[1:0]			uu0uu000

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 ADOH at address 10H

Property	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
ADOH	ADO[23:16]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	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[15:8]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Register ADOLL at address 12H

Property	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
ADOLL	ADO[7:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 23-0 ADO [23:0]: ADC Digital Output
ADO[23] = ADC Digital Output sign bit. 0 = Output is positive; 1 = Output is negative.
ADO[22] = ADC Digital Output Data bit 22.
ADO[21] = ADC Digital Output Data bit 21.
ADO[20] = ADC Digital Output Data bit 20.
ADO[19] = ADC Digital Output Data bit 19.
ADO[18] = ADC Digital Output Data bit 18.
ADO[17] = ADC Digital Output Data bit 17.
ADO[16] = ADC Digital Output Data bit 16.
ADO[15] = ADC Digital Output Data bit 15.
ADO[14] = ADC Digital Output Data bit 14.
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.

Table 2-33 ADC sample frequency selection register

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
MCK							M1_CK	
14h	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Table 2-34 ADC sample frequency selection table

M1_CK	ADC Sample Frequency（ADCF）
0	MCK/16
1	MCK/32

Table 2-35 ADC output rate selection register

Property	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
ADCON						ADM[2:0]		
13h	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Table 2-36 ADC output rate selection table

ADM[2:0]	ADC Output Rate
000	ADCF/128
001	ADCF/256
010	ADCF/512
011	ADCF/1024
100	ADCF/2048
101	ADCF/4096
110	ADCF/8192
111	ADCF/8192

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]		SENS[1:0]					
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-6 SINL[1:0] ADC input port signal multiplexer
11=ADC select single-port input signal
10=ADC input port is connected to TEMP
01= ADC input port is connected to AIN2(PT1[2]) and AIN3(PT1[3])
00= ADC input port is connected to AIN0(PT1[0]) and AIN1(PT1[1])
Bit 5-4 SENS[1:0]
00=select VINP=PT1_0,VINN=PT1_4
01=select VINP=PT1_1,VINN=PT1_4
10=select VINP=PT1_2,VINN=PT1_4
11=select VINP=PT1_3,VINN=PT1_4

Input selection circuit is used to choose the channel of input signal and realize the function of Temperature Sensor, also realize the process of single-port input signal while 4 half-bridge sensor using in common. SINL[1:0] is used to choose PT1_0, PT1_1, PT1_2, PT1_3, the differential output of Temperature Sensor and single-port input signal for ADC differential input signal. When SINL[1:0] =11, users could setup SENS[1:0] to choose single-port input signal PT1_0, PT1_1, PT1_2, PT1_3 for AD converting., and PT1_4 for the reference voltage. Considering the match between temperature character and input impedance, PT1_4 should be supplied by output signal from same half-bridge sensor.

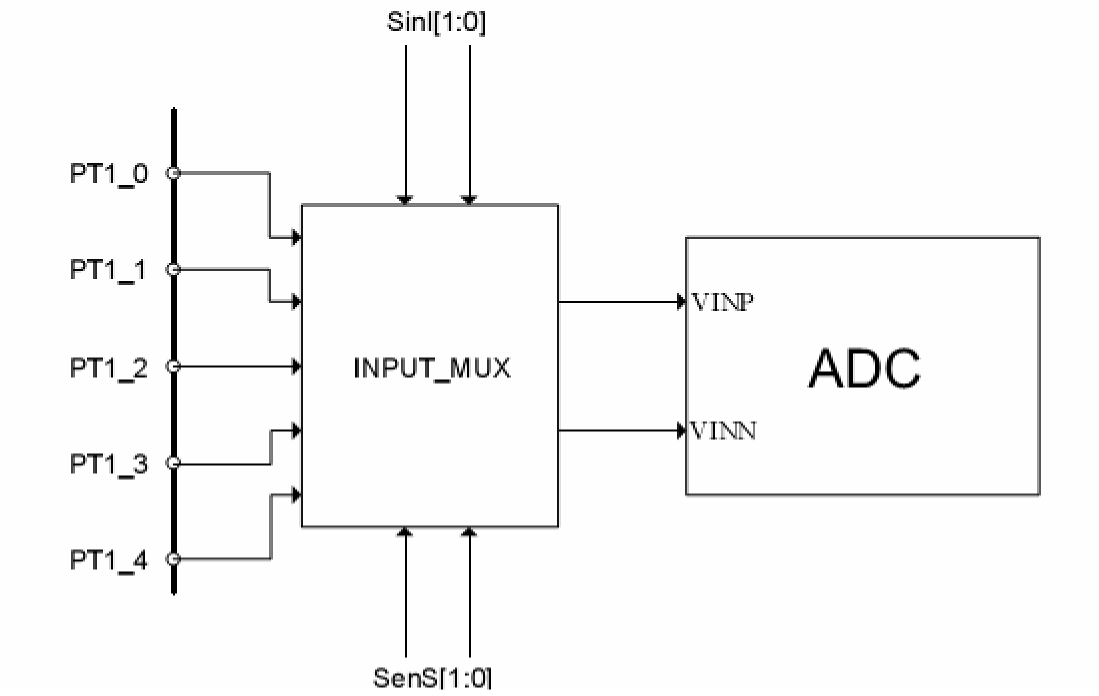


Figure 2-18 ADC selecting single-port input signal

Register NETA at address 1AH

Property	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0
NETC	CHOPM[1:0]				ADG[1:0]		ADEN	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-6 CHOPM ADC random chopper selector
11 = chopper closed
10 = random chopper
01 = 1/2 sample frequency chopper, the clock ascend edge is the middle of sample clock high level
00 = 1/2 sample frequency chopper

Bit 3-2 ADG[1:0] Internal ADC input gain.
11 = Internal ADC input gain is 256
10 = Internal ADC input gain is 128
01 = Internal ADC input gain is 64
00 = Internal ADC input gain is 1

Bit 1 ADEN: ADC enable flag.
1 = ADC is enabled.
0 = ADC is disabled.

Register NETF at address 1DH

Property	U-0	U-0	R/W -0	U-0	U-0	R/W-0	R/W-0	R/W-0
NETF						BGID[1:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 2-1 BGID[1:0]：Bias current of operational amplifier selector
11 = Bias current of internal operational amplifier increases 100%
10 = Bias current of internal operational amplifier increases 50%
01 = Bias current of internal operational amplifier increases 50%
00 = Bias current of internal operational amplifier is typical

Onboard temprature sensor

Set SINL to 10,when using Onboard temprature sensor,and then set PGA to 1.The ADC data is overflow when PGA is set the other values.After selecting Onboard temprature sensor,there is need to discard three ADC data channels. Abrupt changes in the input will require three data cycles to settle.

PGA = 1； V_{REF} = 3.3V。

Output signal of Onboard temprature sensor	400uV/℃
Error of Onboard temprature sensor	8.1%

2.11 PDM (Pulse Density Modulator) Module

The GPIO port PT2[3:2] could be defined as either PDM module output or General purpose I/O. Users could control the PDMEN registers flags to decide the definition. user needs to setup the PMCS register flag to decide the PDM CLK. 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) and PMD2 register flag to do the combination of these 16 internal clock signals.

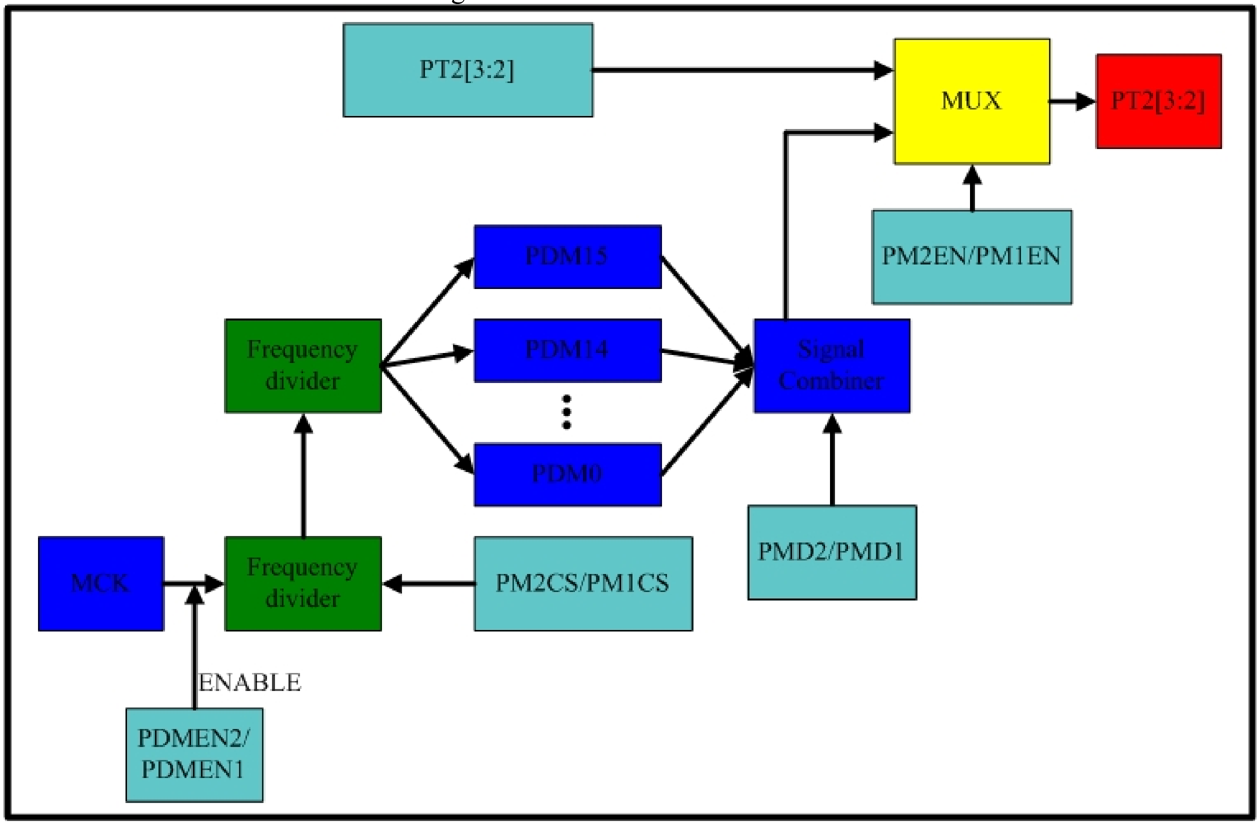


Figure 2-18 CSU1221 PDM module function block

The relation of these 16 signals is as follows. Note that any two of these 16 signal won’t hold high level at the same time. And when PMD1 is 0xFFFF, the PDM output will be constant 1.

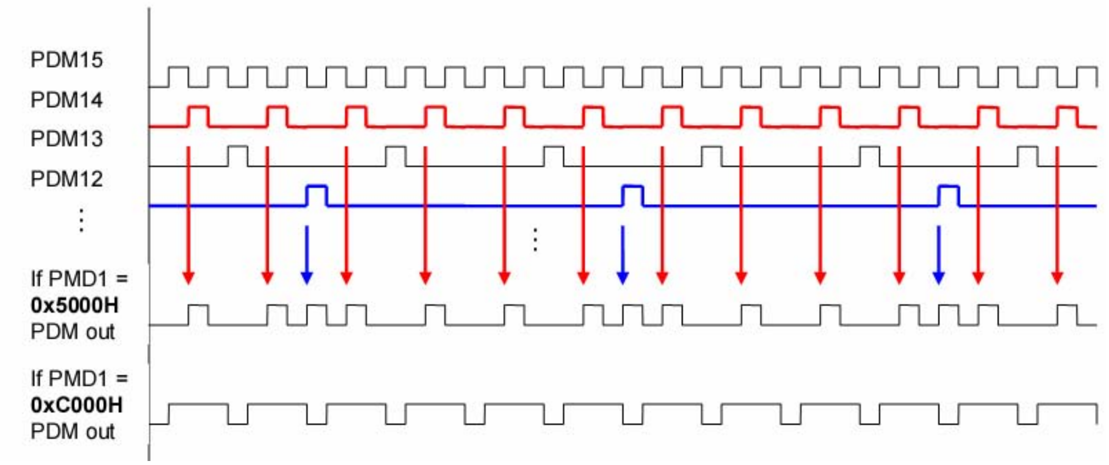


Figure 2-19 PDM module signal generation

Table 2-37 PDM module register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
27h	PT2MR			PM2EN	PM1EN					00000000
30h	PMD1H	PMD1[15:8]								00000000
31h	PMD1L	PMD1[7:0]								00000000
32h	PMD2H	PMD2[15:8]								
33h	PMD2L	PMD2[7:0]								
36h	PMCON	PDM2EN	PM2CS[2:0]			PDM1EN	PM1CS[2:0]			

Register PT2MR at address 27H

Property	U-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
PT2MR			PM2EN	PM1EN				
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 5 PM2EN: PT2[3] output multiplexer
1 = GPIO Port 2 bit 3 (PT2[3]) is defined as PDM output.
0 = GPIO Port 2 bit 3 (PT2[3]) is defined as GPIO.

Bit 4 PM1EN: PT2[2] output multiplexer
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	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	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	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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

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

PMD1[6] = PDM6 (PDM CLK/ 2^{10})Signal Combination enable flag. 1 = Enable ; 0 = Disable
PMD1[5] = PDM5 (PDM CLK/ 2^{11})Signal Combination enable flag. 1 = Enable ; 0 = Disable
PMD1[4] = PDM4 (PDM CLK/ 2^{12})Signal Combination enable flag. 1 = Enable ; 0 = Disable
PMD1[3] = PDM3 (PDM CLK/ 2^{13})Signal Combination enable flag. 1 = Enable ; 0 = Disable
PMD1[2] = PDM2 (PDM CLK/ 2^{14})Signal Combination enable flag. 1 = Enable ; 0 = Disable
PMD1[1] = PDM1 (PDM CLK/ 2^{15})Signal Combination enable flag. 1 = Enable ; 0 = Disable
PMD1[0] = PDM0 (PDM CLK/ 2^{16})Signal Combination enable flag. 1 = Enable ; 0 = Disable

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	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	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	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

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

Register PMCON at address 36H

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
PMCON	PDM2EN	PM2CS[2:0]			PDM1EN	PM1CS[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 3 PDM2EN: PDM2 Module enable flag
1 = PDM2 Module is enabled, GPIO Port 2 bit 3 could be defined as PDM output.
0 = PDM2 Module is disabled, GPIO Port 2 bit 3 could be defined as GPIO.

Bit 2-0 PMCS[2:0]: PDM CLK frequency Selector
111 = PDM2CLK frequency is as MCK/128
110 = PDM2CLK frequency is as MCK/64
101 = PDM2CLK frequency is as MCK/32
100 = PDM2CLK frequency is as MCK/16
011 = PDM2CLK frequency is as MCK/8
010 = PDM2CLK frequency is as MCK/4
001 = PDM2CLK frequency is as MCK/2
000 = PDM2CLK frequency is the same as MCK

Bit 3 PDM1EN: PDM1 Module enable flag
1 = PDM1 Module is enabled, GPIO Port 2 bit 2 could be defined as PDM output.
0 = PDM1 Module is disabled, GPIO Port 2 bit 2 could be defined as GPIO.

Bit 2-0 PMCS[2:0]: PDM CLK frequency Selector
111 = PDM1CLK frequency is as MCK/128
110 = PDM1CLK frequency is as MCK/64
101 = PDM1CLK frequency is as MCK/32
100 = PDM1CLK frequency is as MCK/16
011 = PDM1CLK frequency is as MCK/8
010 = PDM1CLK frequency is as MCK/4
001 = PDM1CLK frequency is as MCK/2
000 = PDM1CLK frequency is the same as MCK

Table 2-38 PMD register table

Address	Name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Value at Power on Reset
14h	MCK	M7_CK	M6_CK	M5_CK		M3_CK	M2_CK	M1_CK	M0_CK	00000000
25h	PT2EN	PT2EN[7:0]								00000000
27h	PT2MR				PM1EN					00000000
30h	PMD1H	PMD1[15:8]								00000000
31h	PMD1L	PMD1[7:0]								00000000
36h	PMCON				PDMEN		PMCS[2:0]			00000000

Operation:

1. Setup M0_CK, M3_CK to decide the MCK.
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 2-39 PDM CLK selection table

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

2.12 Instruction Set

Table 2-40 CSU1221 instruction set table

Instruction	Operation	Cycle	Flag
ADDLW k	[W]←[W]+k	1	C,DC,Z
ADDPCW	[PC] ←[PC]+1+[W]	1	~
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	Z
BSF f,b	[f<b>]←1	1	Z
BTFSC f,b	Jump if[f<b>]=0	1	~
BTFSS f,b	Jump if[f<b>]=1	1	~
CALL k	Push PC+1 and Goto K	1	~
CLRF f	[f]←0	1	Z
CLRWDT	Clear watch dog timer	1	~
COMF f,d	[f]←NOT([f])	1	Z
DECF f,d	[Destination] ←[f] -1	1	C,Z
DECFSZ f,d	[Destination] ←[f] -1,jump if the result is zero	1	C,Z
GOTO k	PC←k	1	~
HALT	CPU Stop	1	~
INCF f,d	[Destination] ←[f]+1	1	C,Z
INCFSZ f,d	[Destination] ←[f]+1,jump if the result is zero	1	C,Z
IORLW k	[W]←[W] OR k	1	Z
IORWF f,d	[Destination] ← [W] OR [f]	1	Z
MOVFW f	[W]←[f]	1	~
MOVLW k	[W]←k	1	~
MOVWF k	[f]←[W]	1	~
NOP	No operation	1	~
RETFIE	Pop PC and GIE =1	1	~
RETLW k	RETURN and W=k	1	~
RETURN	POP PC	1	~
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	[Destinnation] ← [f]– [W]	1	C,DC,Z
SUBWFC f,d	[Destinnation] ← [f]– [W]+C	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 ~ 17Fh).
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

Table 2-41 Instruction Description

1

ADDLW	Add Literal to work register
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

2

ADDPCW	Add W to PC
Syntax	ADDPCW
Operation	(PC)<—(PC)+1+(W) , (W)<=7Fh (PC)<—(PC)+1+(W)-100h, otherwise
Flag Affected	None
Description	The relative address PC + 1 + W are loaded into PC.
Cycle	1

Example1
ADDPCW
	Before instruction:
	W = 7Fh, PC = 0212h
	After instruction:
	PC = 0292h

3

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

Example1
ADDWF f, 0
	指令 Before instruction:
	f = C2h
	W = 17h

	After instruction:
	f = C2h
	W = D9h

Example2
ADDWF f, 1
	Before instruction:
	f = C2h
	W = 17h
	After instruction:
	f = D9h
	W = 17h

4

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 f， 1
	Before instruction:
	C=1
	OPERAND = 02h
	W = 4Dh
	After instruction:
	C=0
	OPERAND = 50h
	W = 4Dh

5

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

6

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

Example1
ANDWF f，0
	Before instruction:
	W = 0Fh, OPERAND = 88h
	After instruction:
	W = 08h, OPERAND = 88h

Example2
ANDWF f，1
	Before instruction:
	W = 0Fh, OPERAND = 88h
	After instruction:
	W = 88h, OPERAND = 08h

7

BCF	Bit Clear f
Syntax	BCF f，b 0<=f<=FFh 0<=b<=7
Operation	(f[b])<—0
Flag Affected	Z
Description	Bit b in [f] is reset to 0.
Cycle	1

Example
BCF FLAG 2
	Before instruction:
	FLAG = 8Dh
	After instruction:
	FLAG = 89h

8

BSF	Bit Set f
Syntax	BSF f，b 0<=f<=FFh 0<=b<=7
Operation	(f[b])<—1
Flag Affected	Z
Description	Bit b in [f] is reset to 1.
Cycle	1

Example
BSF FLAG 2
	Before instruction:
	FLAG = 89h
	After instruction:
	FLAG = 8Dh

9

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 of f is 0,PC add 2,the next instruction of BTFSC is skipped.
Cycle	1

Example NODE BTFSC FLAG2 OP1: OP2:	Before instruction: PC = address (Node) After instruction: If FLAG<2> = 0 PC = address(OP2) If FLAG<2> = 1 PC = address(OP1)

10

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 of f is 1,PC add 2,the next instruction of BTFSS is skipped.
Cycle	1
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)

11

CALL	Subroutine CALL
Syntax	CALL K 0<=K<=1FFFh
Operation	(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	1

12

CLRF	清除f
Syntax	CLRF f 0<=f<=17Fh
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

13

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

14

COMF	Complement f
Syntax	COMF f，d 0<=f<=17Fh d=0,1
Operation	(Destination)<—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
COMF f，0
	Before instruction:
	W = 88h, OPERAND = 23h
	After instruction:
	W = DCh, OPERAND = 23h

Example2
COMF f，1
	Before instruction:
	W = 88h, OPERAND = 23h
	After instruction:
	W = 88h, OPERAND = DCh

15

DECF	Decrement f
Syntax	DECF f，d 0<=f<=255 d=0,1
Operation	(Destination)<—(f)-1
Flag Affected	C,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

Example1
DECF f，0
	Before instruction:
	W = 88h, OPERAND = 23h
	After instruction:
	W = 22h, OPERAND = 23h

Example2
DECF f，1
	Before instruction:
	W = 88h, OPERAND = 23h
	After instruction:
	W = 88h, OPERAND = 22h

16

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	--
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 PC is added 2 and the next instruction of DECFSZ is skipped
Cycle	1
Example Node DECFSZ 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(OP)

17

GOTO	Unconditional Branch
Syntax	GOTO K 0<=K<=1FFFh
Operation	PC<—K
Flag Affected	None
Description	The immediate address is loaded into PC.
Cycle	1

18

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

19

INCF	f 加1
Syntax	INCF f，d 0<=f<=FFh d=0,1
Operation	(Destination)<—(f)+1
Flag Affected	C,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

Example1
INCF f，0
	Before instruction:
	W = 88h, f = 23h
	After instruction:
	W = 24h, f = 23h

Example2
INCF f，1
	Before instruction:
	W = 88h, f = 23h
	After instruction:
	W = 88h, f = 24h

20

INCFSZ	Increment f, skip if zero
Syntax	INCFSZ f，d 0<=f<=17Fh d=0,1
Operation	(Destination)<—(f)+1 skip if the result is zero
Flag Affected	--
Description	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 PC is added 2 and the next instuction of INCFSZ is skipped
Cycle	1
Example Node INCFSZ FLAG,1 OP1： OP2：	Before instruction: PC = address (Node) After instruction: [FLAG] = [FLAG] + 1OP2: If [FLAG] = 0 PC = address(OP2) If [FLAG] ≠ 0 PC = address(OP1)

21

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

22

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 f,1
	Before instruction:
	W = 88h, OPERAND = 23h
	After instruction:
	W = 88h, OPERAND = ABh

23

MOVFW	Move f to W
Syntax	MOVFW f 0<=f<=17Fh
Operation	(W)<—(f)
Flag Affected	None
Description	Move data from [f] to the W register.
Cycle	1

Example
MOVFW f
	Before instruction:
	W = 88h, f = 23h
	After instruction:
	W = 23h, f = 23h

24

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

25

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 f
	Before instruction:
	W = 88h, f = 23h
	After instruction:
	W = 88h, f = 88h

26

NOP	None Operation
Syntax	NOP
Operation	None Operation
Flag Affected	None
Description	None Operation
Cycle	1

27

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	1

28

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	1

29

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.

30

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 f，1
	Before instruction:
	C=0
	W = 88h, f = E6h
	After instruction:
	C=1
	W = 88h, f = CCh

31

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 f，0
	Before instruction:
	C=0
	OPERAND = 95h
	After instruction:
	C=1
	W = 4Ah, OPERAND = 95h

32

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

33

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

Example2
SUBLW 02H
	Before instruction:
	W = 02h
	After instruction:
	W = 00h
	C=1
	Z=1

Example3
SUBLW 02H
	Before instruction:
	W = 03h
	After instruction:
	W = FFh
	C=0
	Z=0

34

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

Example1
SUBWF f，1
	Before instruction:
	OPERAND = 33h, W = 01h
	After instruction:
	OPERAND = 32h
	C=1
	Z=0

Example2
SUBWF f，1
	Before instruction:
	OPERAND = 01h, W = 01h
	After instruction:
	OPERAND = 00h
	C=1
	Z=1

Example3
SUBWF f，1
	Before instruction:
	OPERAND = 04h, W = 05h
	After instruction:
	OPERAND = FFh
	C=0
	Z=0

35

SUBWFC	Subtract W and Carry from f
Syntax	SUBWFC f，d 0<=f<=FFh d=0,1
Operation	(Destination)<—(f)-(W)-1+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
SUBWFC f，1
	Before instruction:
	f = 33h, W = 01h C=1
	After instruction:
	f = 32h, C = 1, Z = 0

Example2
SUBWFC f，1
	Before instruction:
	f = 02h, W = 01h C=0
	After instruction:
	f = 00h, C = 1, Z = 1

Example3
SUBWFC f，1
	Before instruction:
	f = 04h, W = 05h C=0
	After instruction:
	f = FEh, C = 0, Z = 0

36

XORLW	Exclusive OR literal with W
Syntax	XORLW K 0<=K<=FFh
Operation	(W)<—(W)^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

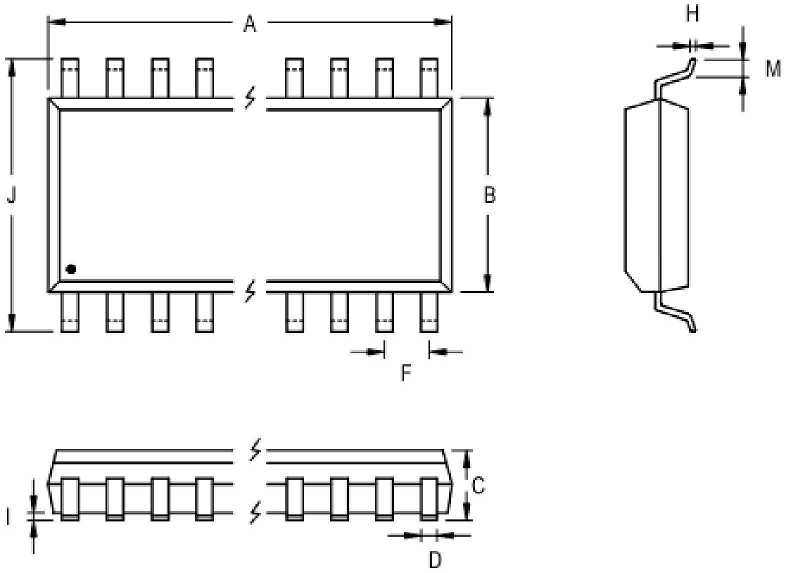
Example
XORLW 5Fh
	Before instruction:
	W = ACh
	After instruction:
	W = F3h

37

XORWF	Exclusive OR W and f
Syntax	XORWF f，d 0<=f<=17Fh d=0,1
Operation	(Destination)<—(W)^(f)
Flag Affected	Z
Description	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

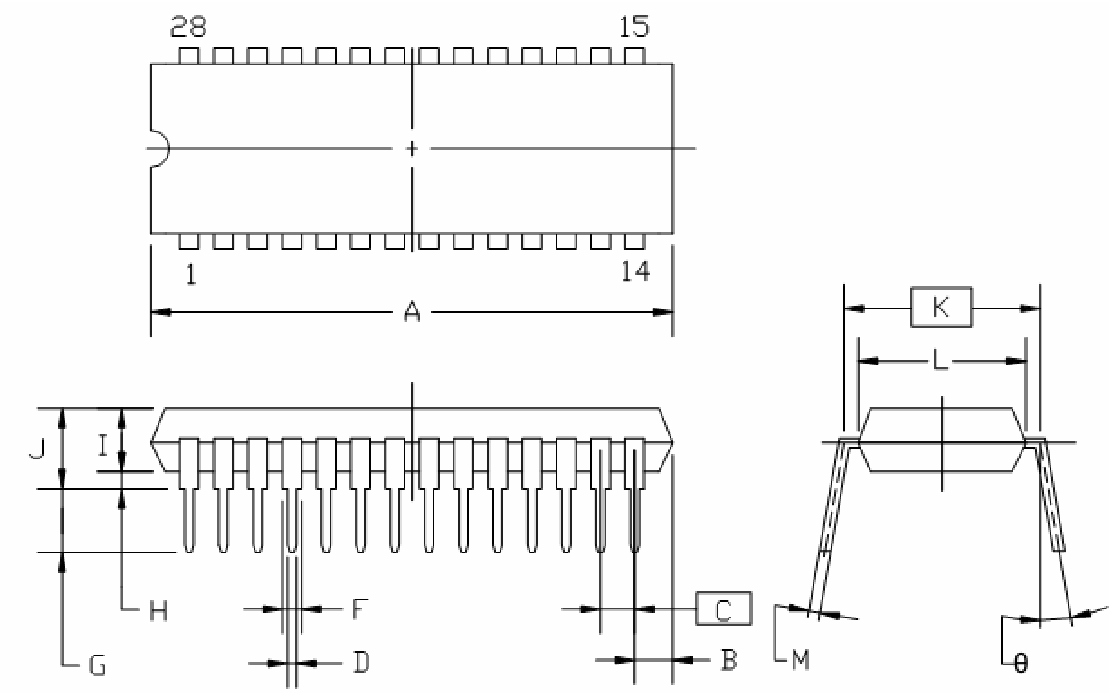
Example
XORWF f，1
	Before instruction:
	f= 5Fh, W = ACh
	After instruction:
	f = F3h

3 CSU1221 PACKAGE



Symbol	Dimensions In Millimeters		Dimensions In Inches	
	Min	Max	Min	Max
A	17.704	18.110	0.697	0.713
B	7.391	7.595	0.291	0.299
C	2.362	2.642	0.093	0.104
D	0.330	0.508	0.013	0.020
F	1.194	1.346	0.047	0.053
H	0.229	0.330	0.009	0.013
I	0.102	0.305	0.004	0.012
J	10.008	10.643	0.394	0.419
M	0.381	1.270	0.015	0.050

28-Lead SOP Plastic Package



Symbol	Dimensions In Millmeters			Dimensions in Inches		
	Min	Nom	Max	Min	Nom	Max
A	25.4	25.6	25.8	1.00	1.008	1.016
B	-	1.243	-	-	0.049	-
C	-	1.778	-	-	0.07	-
D	0.36	0.46	0.56	0.014	0.018	0.022
F	0.9	1	1.1	0.035	0.039	0.043
G	3.0	3.3	3.6	0.118	0.13	0.142
H	0.3	-	-	0.012	-	-
I	3.1	3.3	3.5	0.122	0.13	0.138
J	-	-	4.31	-	-	0.17
K	-	10.16	-	-	0.4	-
L	8.55	8.8	9.05	0.337	0.346	0.356
M	0.20	0.25	0.35	0.008	0.01	0.014
Θ	0°	-	15°	0°	-	15°

SDIP-28 Plastic Package