芯海科技
CHIPSEA

CSU8RP1001User Manual

based onOTP ROMof8bitRISC MCUs

Rev.1.2

Mailing address: Nanhai Avenue, Nanshan District, Shenzhen1079Garden City Digital BuildingAbuilding9

Building Postal Code: 518067

Company Tel: +(86 755) 86169257

Fax: +(86 755)86169057

Company website: www.chipsea.com

芯海科技 CHIPSEA

CSU8RP1001

54pin8bitOTP ROMSingle chip product introduction

high performanceRISC CPU

- 8bit microcontrollerMCU
- built-in4K×16bit one-time programmable memory (OTP ROM)
- 256byte data memory (SRAM) only39
- single word instruction
- 8level storage stack
- Support online burning

oscillator

- inner band4MHzoscillator with an accuracy of±3%(25°C)
- external32768HzCrystal (RTC)or4MHz~8MHz crystal oscillator

Peripheral Features

- 14two-wayI/Omouth and1input port
- 10expandable output ports
- 1Buzzer output
- 4×14ofLCDdrive

— Optional internal crystal oscillator, external crystal oscillator,WDTcrystal oscillator as clock source

- Choose from two differentLCDdrive waveform

— Different bias voltage generation methods can be selected

- 2external interrupt
- Low voltage detection (LVD) pins (provided internally2.4V, 2.5V,2.6V,2.7V,2.8V,3.2V,3.6Vvoltage comparison)
- Built-in temperature sensor

Analog characteristics

- analog-to-digital converter (ADC)
  - — 2analog input
  - — twenty fourBit resolution, effective precision15bit (PGA for68 , the output rate is7.8KHz).
  - — Internally integrated programmable gain amplifier provides1, 4,68,136,272and other gains of different magnifications.
  - — ADCoutput rate30.5Hz~62.5KHz
- Built-in charge pump (2.6V 2.8V 3.0V 3.2V) built-in
- voltage regulator for sensor and modulator (2.3V, 2.5V, 2.8V, 3V)

Features of Dedicated Microcontrollers

- power-on reset (POR)
- Power-on Reset Delay Timer (39ms)
- Built-in low voltage reset (LVR)
- Timer1
  - — 8Bit programmable prescaler8bit timer counter
- Timer2
  - — 8Bit programmable prescaler8bit divider
- Extended Watchdog Timer (WDT)
  - — Programmable time range
- Voltage working range
  - — DVDD 2.4V~3.6V
  - — AVDD 2.4V~3.6V

Low Power Features

- MCUWorking current

— normal mode1mA@1MHz 2mA@4MHz (Work Voltage3.3V);

— Current in sleep mode is less than1.5μA

package

- 52-PIN die

Application

- Solar Electronic Scale
- Portable Instruments

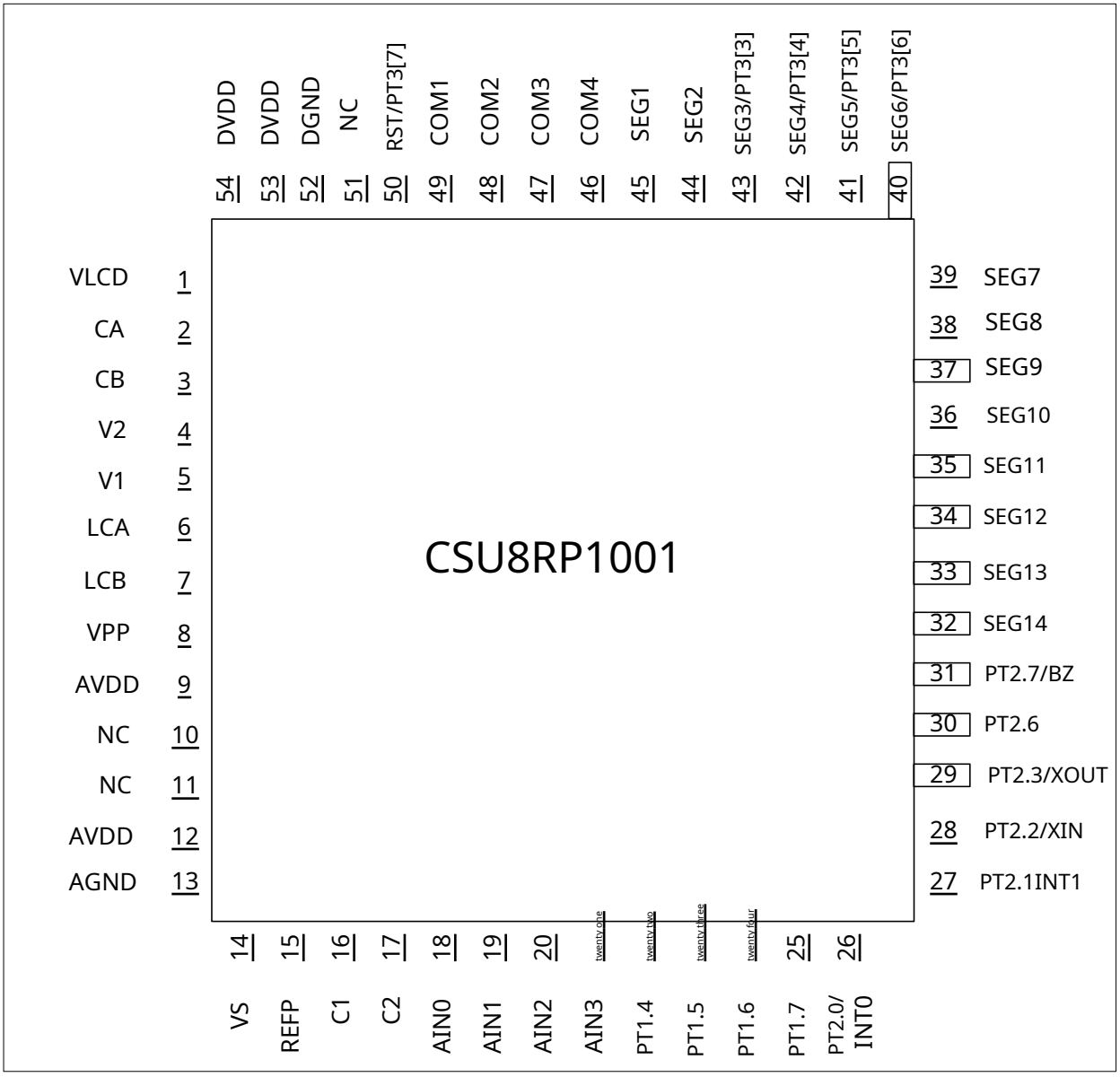
History modification record

2011year8moon15day	drafted	Version is1.0
2011year12moon9day	increase the9feet forAVDDandPT3[7]illustrate	Version is1.1
2012year12moon20	Renewedlogofirst draft completed	Version is1.2

Table of contents

1product description

Pinconfigure

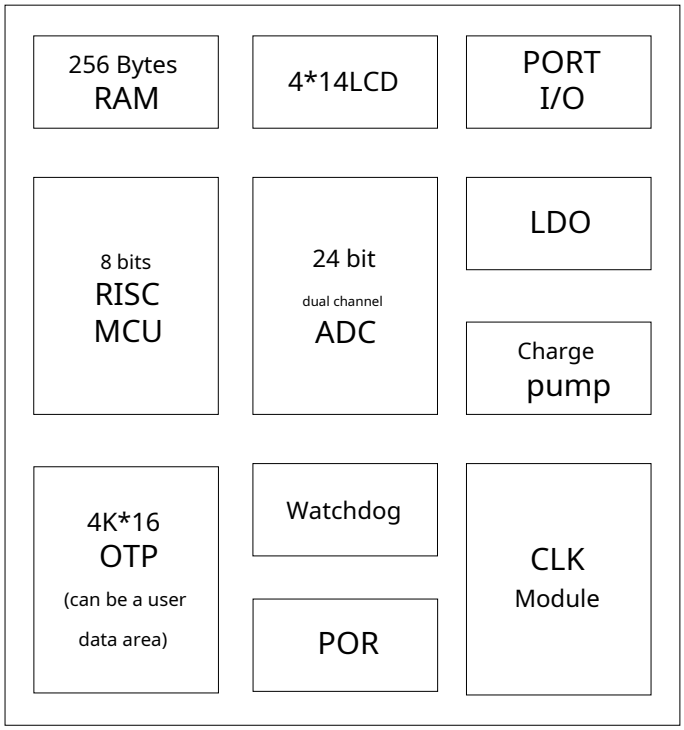


picture1-1Pin Configuration Diagram

surface1-1Pin Description Table

pin name	input Output	pin number	describe
VLCD	I/O	1	LCDpower source
CA	I/O	2	VLCDRequired Charge Exchange Capacitors
CB	I/O	3	VLCDRequired Charge Exchange Capacitors
V2	O	4	LCDDisplays the required intermediate voltage range2
V1	O	5	LCDDisplays the required intermediate voltage range1
LCA	I/O	6	Voltage divider charge pump to generate required charge exchange capacitors
LCB	I/O	7	Voltage divider charge pump to generate required charge exchange capacitors
VPP	I	8	programmingOTPTiming6.5V
AVDD	I	9	analog power
NC	- -	10	reserve
NC	- -	11	reserve
AVDD	I	12	analog power
AGND	I	13	Analogously
VS	O	14	simulationLDOoutput
REFP	I	15	ADCReference voltage input (connected toVSenter)
C1,C2	I	16~17	ADConversion capacitor
AIN0~1	I	18~19	Analog differential input
AIN2~3	I	20~21	Analog differential input
PT1[4]/LPD	I/O	twenty two	I/Oor low voltage detection input
PT1[5]	I/O	twenty three	I/O
PT1[6]	I/O	twenty four	I/O
PT1[7]	I/O	25	I/O
PT2[0]/INT0	I/O	26	OTPprogrammed data or asI/Oor external interrupt0enter
PT2[1]/INT1	I/O	27	OTPprogrammed clock or asI/Oor external interrupt1enter
PT2[2]/XIN	I/O	28	I/O;External crystal oscillator input
PT2[3]/XOUT	I/O	29	I/O;External crystal oscillator output
PT2[6]	I/O	30	I/O
PT2[7]/BZ	I/O	31	I/Oor buzzer output
SEG14~7	O	32~39	LCD Segmentoutput or as a digital output
SEG6~3/PT3[6:3]	O	40~43	LCD Segmentoutput or as a numberI/O
SEG2~1	O	44~45	LCD Segmentoutput or as a digital output
COM4~1	O	46~49	LCD Comoutput
RST/PT3[7]	I	50	Reset signal input or digital input
NC	- -	51	reserve
DGND	I	52	digitally
DVDD	I	53	digital power
DVDD	I	54	digital power

Functional Module Schematic



picture1-2 CSU8RP1001functional module

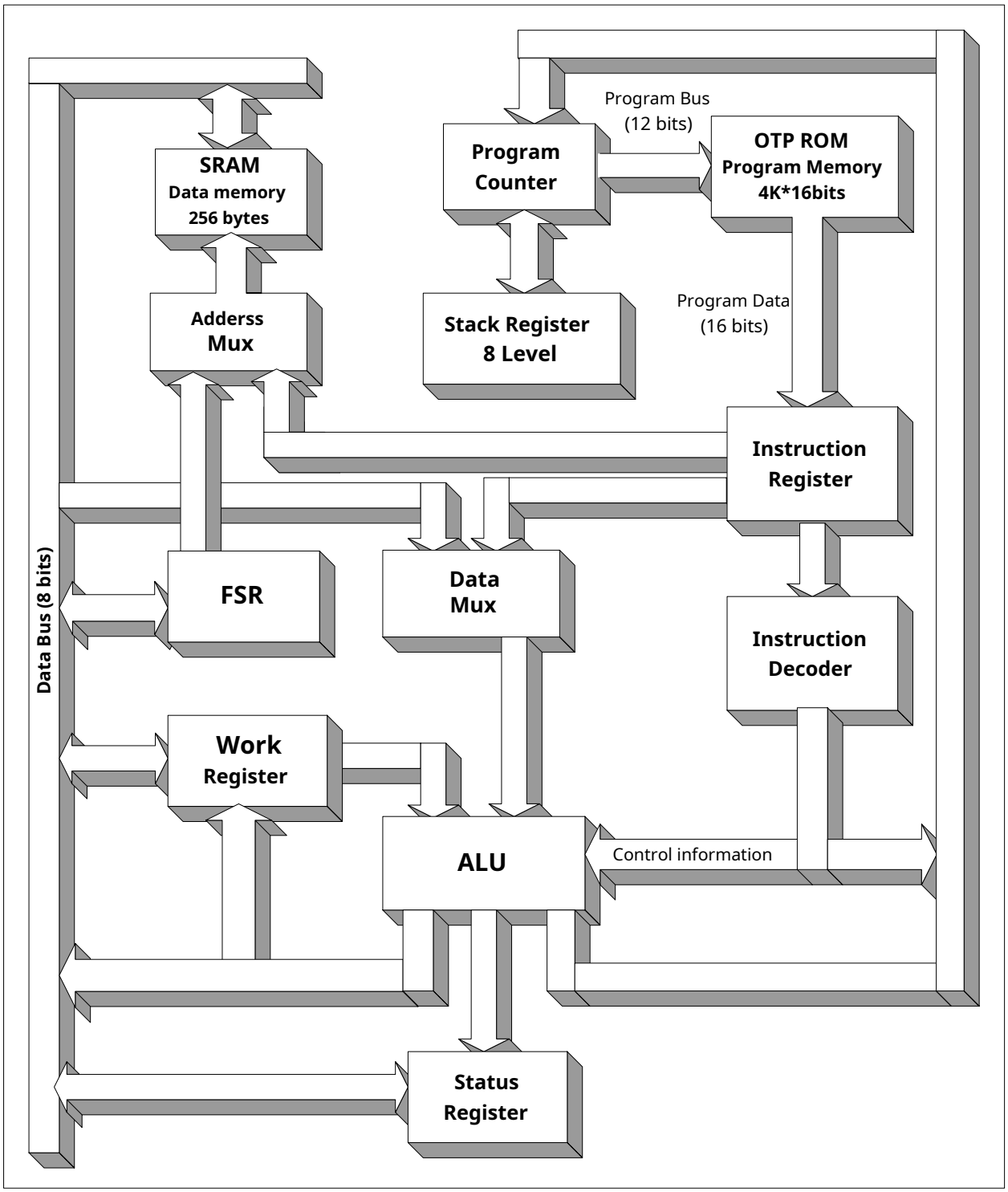
It can be seen from the functional module schematic diagram that there are5functional modules, as shown in the table1-2describe

surface1-1 CSU8RP1001Main function description

project	subproject	describe
CPUnuclear	RISC CPU Core	For a detailed description see2.1Festival
	OTPprogram memory	OTP: One-time programmable 8K Bytesused for4Kprogramming instructions
	data storage	CSU8RP1001with128 Bytesthe special function registers,256 BytesOrdinary data storage.
	clock system	CSU8RP1001There are two clock sources. one is4MHzthe internal clock forCPU work, the other is32768Hzor4MHz~8MHzexternal clock.
Data function module	timer module	Clock counter for timed interrupt and watchdog
	LCDmodule	inner band4×14ofLCDdriver
	Buzzer	User connects a buzzer to the built-in buzzer interface to receive warning or reminder signals
	Ext.INT	CSU8RP1001supply2external interrupt interface
Analog function module	ADC	inner bandSigma-DeltaofADCConvert the sensor's analog signal to a digital signal
Power function module	power module	CSU8RP1001There is a dedicated power system. This power system can beADCProvides a fixed voltage. The chip's input voltage can float within a range
UniversalI/O	PT1	PT1interface has4bit.
	PT2	PT2interface has6bit. User can define this6Bit interface for general purpose or some dedicated functions, such as external interrupt, buzzer
	PT3	PT3interface has5bit.

2Standard features

2.1 CPUnuclear



picture2-1 CSU8RP1001 CPUFunctional block diagram of the core

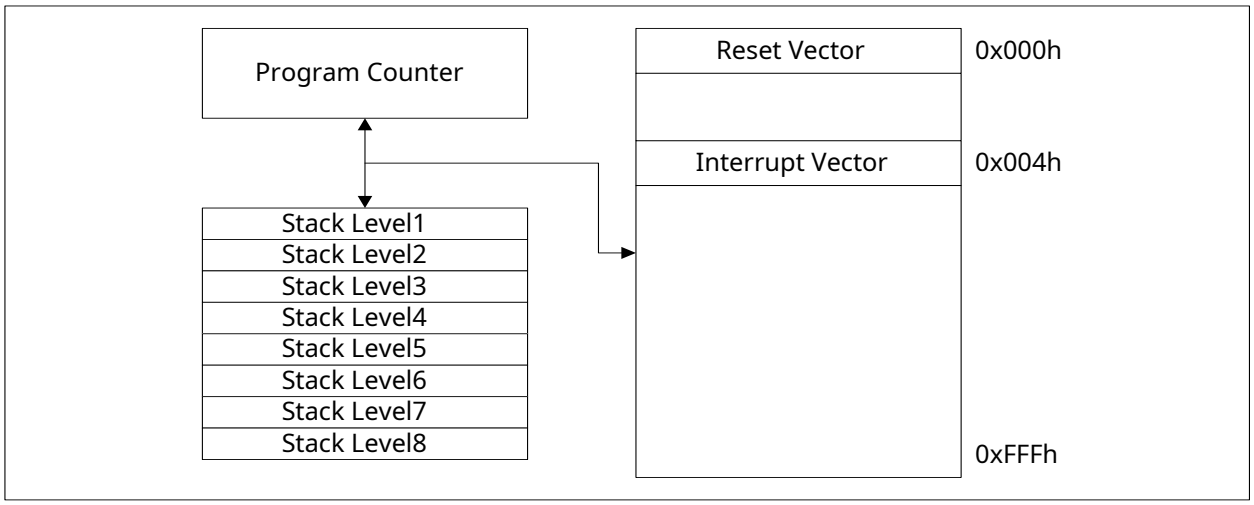
fromCPUIn the functional module diagram of the core, you can see that it mainly contains7main registers and2memory unit.

surface2-1 MCUArchitecture Description

module name	describe
program counter	This register is inCPUplays an important role during the work week, it recordsCPUA pointer to an instruction in program memory that is processed every cycle. in aCPUcycle, the program counter will update the program memory address (12bits), the instruction pointer is pushed into program memory, and then automatically incremented1for the next cycle.
stack register	The stack register is used to record the instruction pointer returned by the program. When a program calls a function, the program counter pushes the instruction pointer onto the stack register. After the function execution finishes, the stack register will return the instruction pointer to the program counter to continue the original program processing.
instruction register	The program counter pushes the instruction pointer (program memory address) to program memory, which pushes the program memory data (16bits) and the instruction is pushed to the instruction register. CSU8RP1001The command is16bits,include3Various types of information: direct address, immediate data and control information interest. CPUThe immediate data can be pushed to the working register, or after some processing, the immediate data can be stored in the data memory register pointed to by the direct address according to the control information. direct address (8bits) The address of the data memory.CPUData memory operations can be performed using this address. direct data (8bits) CPUpassALUUse this data to operate the working register. control information it recordsALUoperation information.
instruction decoder	The instruction register pushes control information to the instruction decoder for decoding, and the decoder then sends the decoded information to the relevant registers.
arithmetic logic unit	The arithmetic logic unit can not only complete8Bit binary addition, subtraction, addition1,reduce1Waiting for arithmetic calculation, you can also correct8 Bit variables perform logical operations such as logical AND, OR, XOR, cyclic shift, complement, and clearing.
working register	Working registers are used to cache data at certain memory addresses in data memory.
status register	whenCPUuseALUWhen processing register data, the following states will change in the following order:PD, TO,DC,CandZ.
file selection register	existCSU8RP1001the instruction set,FSRis used for indirect data processing (that is, to implement indirect addressing). users can take advantage ofFSRto store a register address in data memory, and then passINDThe register handles this register.
program memory	CSU8RP1001inner band8K bytesofOTP ROMas program memory. Due to the opcode of the instruction (OPCODE)Yes16bits, the user can only program at most4Kinstruction. The program memory address bus is 12bits, the data bus is16bits.
data storage	CSU8RP1001inner band256bytesofSRAMas data storage. The address bus for this data memory is 8bits, the data bus is8bits.

2.1.1memory

1.The program memory is mainly used for the storage of instructions, inCSU8RP1001, the program memory is4K*16bitofOTP, for the programmer, the memory is read-only and cannot be written to.
systematicresetaddress is0x000, the interrupt entry address is0x004, it should be noted that all interrupts share the same interrupt entry address.



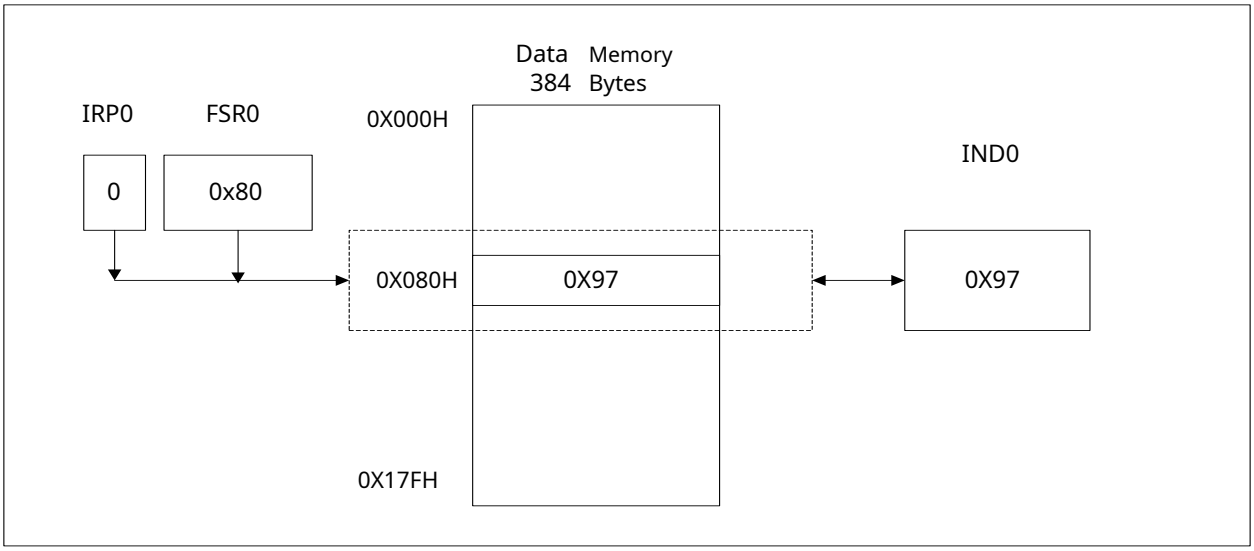
picture2-2program memory

2.The data memory is mainly used for the storage of global and intermediate variables during the running of the program. The memory is divided into three sections. address0x000to0x007It is a system special function register, such as indirect address, indirect address pointer, status register, working register, interrupt flag bit, interrupt control register. address0x008to0x07fPeripheral special function registers, such as IO port, timer,ADC,LCDDrivers, System Special Function Registers and Peripheral Special Function Registers are implemented with registers, while general purpose data memory isRAMIt can be read and written.

surface2-2Data Memory Address Assignment

data storage	initial address	end address
System Special Function Register	0x00	0x07
Peripheral Special Function Registers	0x08	0x7F
General purpose data storage	0x80	0xFF

3.passIND0as well asFSR0These two registers provide indirect access to data memory and special function registers. When the slave indirect address register (IND0)When reading data,MCUis actuallyFSR0The value in is used as the address to access the data memory to get the data. When the indirect register (IND0)When writing data,MCUis actuallyFSR0The value in is used as the address to access the data memory and store the value at that address. Its access method is shown in Fig.2-3.



picture2-3indirect address access

2.1.2status register

The status register containsALUarithmetic state and reset state. The status register is similar to other registers and can be the target register of any instruction. If the status register is the destination register of an instruction and affectsZ,DCorCbit, then writing to these three bits is disabled. These bits are set or cleared by device logic.TOandPDBits are not writable.

Status register (address is04h)

characteristic	R/W-0	R/W-0	U-0	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:IND1Indirect page addressing bits

1 =indirect addressingIND1when, after visiting128byteaddress:100H~17FH 0

=indirect addressingIND1when, before visiting256byteaddress:00H~FFH

Bit 6 IRP0:IND0Indirect page addressing bits

1 =indirect addressingIND0when, after visiting128byteaddress:100H~17FH 0

=indirect addressingIND0when, before visiting256byteaddress:00H~FFH PD:

Bit 4 Power-down flag. By writing to this bit0clear,sleepset this bit after

1 =implementSLEPafter instruction

0 =After power-on reset

Bit 3 TO: Watchdog time-out flag. By writing to this bit0Cleared, watchdog time-out sets this bit

1 =Watchdog time-out occurs

0 =After power-on reset

Bit 2 DC: nibble carry flag/borrow flag, used forADDWF(C)andSUBWF(C)

When used for borrow, the polarity is reversed

1 =the result of the4carry overflow 0 =

the result of the4no carry overflow C:

Bit 1 carry flag/borrow flag

When used for borrow, the polarity is reversed

1 =The highest bit of the result (MSB), a carry overflow occurs 0 =

The highest bit of the result (MSB) without carry overflow Z: zero

Bit 0 flag

1 =The result of an arithmetic or logical operation is0 0 =The

result of an arithmetic or logical operation is not0

characteristic(Property):

R =readable bit W =writable bit U =invalid bit

-n=Value after power-on reset '1' =bit is set '0'=bit is cleared X =indeterminate bit

2.1.3 INTEandINTFinterrupt register

The entry address of the interrupt system is0x004, There is no priority between each interrupt, and the priority of each interrupt is controlled by the program. As long as there is an interrupt flag bit, there will be an interrupt response. After responding to the interrupt, the software needs to clear the interrupt flag bit, otherwise the interrupt will continue to be responded to.

INTEandINTFThe registers are readable and writable, and include enable and flag bits for interrupting the device.

INTEregister (address is07h)

characteristic	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	Bit3	Bit2	Bit1	Bit0

Bit 7 GIE: Global interrupt enable flag

1 =Enable all non-maskable interrupts

0 =Disable all interrupts

Bit 4 TMIE:8-BitTimer interrupt enable flag

1 =Enable timer interrupt

0 =Disable timer interrupts

Bit 2 ADIE:ADCinterrupt enable flag

1 =EnableADCinterrupt

0 =DisableADCinterrupt

Bit 1 E1IE:PT2.1External interrupt enable flag

1 =EnablePT2.1External Interrupt 0 =

DisablePT2.1External Interrupt E0IE:

Bit 0 PT2.0External interrupt enable flag

1 =EnablePT2.0External Interrupt 0 =

DisablePT2.0External Interrupt

INTFregister (address is06h)

characteristic	U-0	U-0	U-0	R-0	R/W-0	R-0	R-0	R-0
INTF				TMIF		ADIF	E1IF	E0IF
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 4 TMIF:8-BitTimed interrupt flag, cleared by software, set high by hardware

1 =A timed interrupt occurs and must be cleared by software0 0

=No timed interrupt occurred

Bit 2 ADIF:ADCInterrupt flag, cleared by software, set high by hardware

1 =occurADCinterrupt, must be cleared by software0 0 =

did not happenADCinterrupt

Bit 1 E1IF:PT2.1External interrupt flag, cleared by software, set high by hardware

1 =occurPT2.1External interrupt, must be cleared by software0 0

=did not happenPT2.1External Interrupt

Bit 0 E0IF:PT2.0External interrupt log, cleared by software, set high by hardware

1 =occurPT2.0External interrupt, must be cleared by software0 0

=did not happenPT2.0External Interrupt

characteristic(Property):

R =readable bit W =writable bit U =invalid bit

-n=Value after power-on reset '1' =bit is set '0'=bit is cleared X =indeterminate bit

2.2 SFRs

2.2.1system special register

system-specific registers are used to completeCPUThe function of the core consists of indirect address, indirect address pointer, status register, working register, interrupt flag and interrupt control register.

surface2-3System Register Table

address	name	Bit7	Bit6	Bi5	Bi4	Bit3	Bit2	Bit1	Bit0	After power-on reset the value of
00h	IND0	byFSR0The content in the data memory as the address of the data								uuuuuuuu
01h	IND1	byFSR1The content in the data memory as the address of the data								uuuuuuuu
02h	FSR0	Address pointer to indirect data memory0								00000000
03h	FSR1	Address pointer to indirect data memory1								00000000
04h	STATUS	IRP1	IRP0		PD	TO	DC	C	Z	00u00000
05h	WORK	working register								00000000
06h	INTF				TMIF		ADIF	E1IF	E0IF	uuu0u000
07h	INTE	GIE			TMIE		ADIE	E1IE	E0IE	0uu0u000

2.2.2Auxiliary special register

Auxiliary special registers are designed for auxiliary functions, such asI/Omouth, timer,ADC, the signal condition control register, LCDdrive.

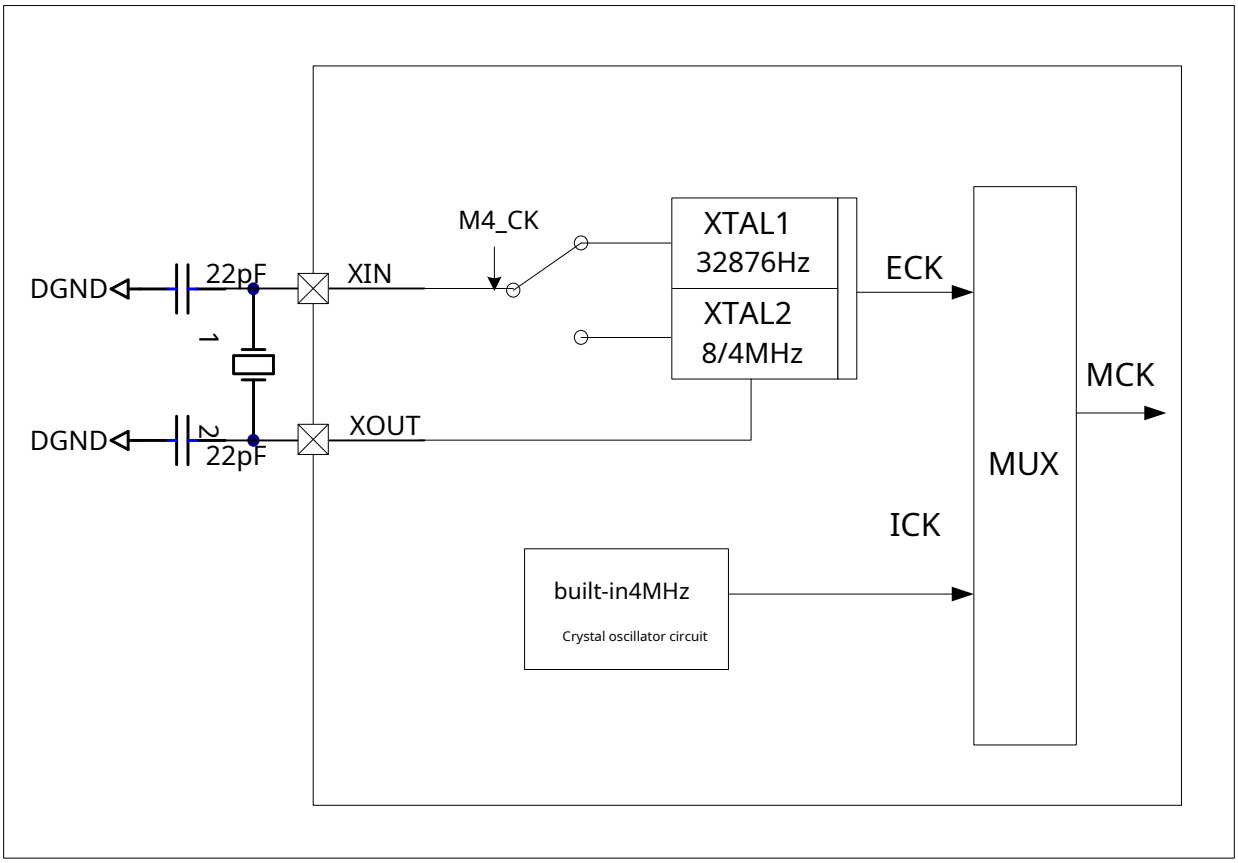
surface2-4Auxiliary Special Register List

address	name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	power-on reset value
0Ah	EADRH					PARH[3:0]				uuuu0000
0Bh	EADRL	PARL[7:0]								00000000
0Ch	EDATH	EDATH[7:0]								00000000
0Dh	WDTCON	WDTOEN	WDTEN		Wdt_lcd		WTS[2:0]			00u0u000
0Eh	TMOUT	TMOUT[7:0]								00000000
0Fh	TMCON	TRST	EO_SLP			TMEN	TMEN2			10uu00uu
10h	ADOH	ADO [23:16]								00000000
11h	ADOL	ADO[15:8]								00000000
12h	ADOLL	ADO[7:0]								00000000
13h	ADCON			ADFEN	COMBS		ADM[2:0]			uu00u000
14h	MCK	M7_CK	M6_CK	M5_CK	M4_CK	M3_CK	M2_CK	M1_CK	M0_CK	01000000
15h	PCK		LCDSCKS[3:0]			S_BEEP[1:0]				u00000uu
18h	NETA	SINL[1:0]			EN_IA	EN_CHS[1:0]		PGA_C[1:0]		00u00000
19h	NETB					ERV				uuuu0uuu
1Ah	NETC	CHS_IA	CHS_MOD	PGA_F[3:0]				ADEN		0000000u
1Bh	NETD	CHP_VPP	DIVS	LCDCH	LEVEL_S	VLCDX[1:0]		LCDREF[1:0]		00000000
1Ch	NETE	LDOS[1:0]			SILB[2:0]			ENLB		00u0000u
1Dh	NETF		LVR_EN	ENVDDA	BGIDA[1:0]		BGID[1:0]		ENVB	u0000000
1Fh	SVD								LBOUT	uuuuuuu0
20h	PT1	PT1[7:4]								0000uuuu
21h	PT1EN	PT1EN[7:4]								0000uuuu
22h	PT1PU	PT1PU[7:4]								0000uuuu
23h	AENB	AOENB3			AOENB2	AIENB1	AIENB3			0uu000uu
24h	PT2	PT2[7:6]				PT2[3:0]				00uu0000
25h	PT2EN	PT2EN[7:6]				PT2EN[3:0]				00uu0000
26h	PT2PU	PT2PU[7:6]				PT2PU[3:0]				00uu0000
27h	PT2MR	BZEN			TMOEN	E1M[1:0]		E0M[1:0]		0uu00000
28h	PT3	PT3[7:3]								00000uuu
29h	PT3EN		PT3EN[6:3]							u0000uuu
2Ah	PT3PU	PT3PU[7:3]								10000uuu
36h	TMIN	TMIN[7:0]								11111111
37h	TM2IN	TM2IN[7:0]								11111111
38h	WDTIN	WDTIN[7:0]								11111111
40h	LCD1					SEG1[3:0]				uuuu0000
41h	LCD2					SEG2[3:0]				uuuu0000
42h	LCD3					SEG3[3:0]				uuuu0000
43h	LCD4					SEG4[3:0]				uuuu0000
44h	LCD5					SEG5[3:0]				uuuu0000
45h	LCD6					SEG6[3:0]				uuuu0000
46h	LCD7					SEG7[3:0]				uuuu0000
47h	LCD8					SEG8[3:0]				uuuu0000
48h	LCD9					SEG9[3:0]				uuuu0000
49h	LCD10					SEG10[3:0]				uuuu0000
4Ah	LCD11					SEG11[3:0]				uuuu0000
4Bh	LCD12					SEG12[3:0]				uuuu0000
4Ch	LCD13					SEG13[3:0]				uuuu0000
4Dh	LCD14					SEG14[3:0]				uuuu0000
58h	LCDENR	LCDCKS[1:0]		LCDEN	LCDWS	LEVEL	LCD_DUTY[1:0]		ENPMPL	00000110
59h	COMP			TCOM						uu0uuuuu

Note: The above undefined byte part cannot be set by the user.1operate. To set it1operation, may affect the chip operation or performance.

2.3clock system

2.3.1Oscillator Status



picture2-4 CSU8RP1001Oscillator State Diagram

surface2-5 CSU8RP1001Clock System Register List

land site	name	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	power-on reset bit value
14h	MCK	M7_CK	M6_CK	M5_CK	M4_CK	M3_CK	M2_CK	M1_CK	M0_CK	01000000

CSU8RP1001There are two clock sources. One is the internal integrated clock,4MHzclock forCPUwork, the other is an external clock, viaM4_CKto choose different crystal oscillator circuits to choose32768Hzthe clock signal or4MHZ~8MHZhigh-speed crystal clock. Users can passMCKregister to select one of the clocks. Please see picture2-4.

surface2-6External crystal oscillator circuit selection list

M4_CK	Crystal oscillator circuit
0	32KHzCrystal oscillator circuit
1	4MHz~8MHzCrystal oscillator circuit

surface2-7 MCKpick list

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

User must set it correctlyM7_CKandM6_CKto enable the internal and external oscillators, as shown in the table2-8shown. If the sleep instruction is executed to enableCSU8RP1001In sleep mode, the internal and external oscillators will not be enabled.

surface2-8Oscillator State Selection List

enter					Oscillator Status	
sleep command	EO_SLP	M7_CK	M6_CK	M4_CK	Internal oscillator	External oscillator
1	0	X	X	X	Disable	Disable
1	1	X	0	0	Disable	Enable
1	1	X	0	1	Disable	Disable
1	1	X	1	X	Disable	Disable
0	x	0	0	X	Enable	Enable
0	x	0	1	X	Enable	Disable
0	x	1	0	X	Disable	Enable
0	x	1	1	X	Disable	Disable

Note:XThat is, the bit can have any value.

Note: When using an external crystal,32768HzThe crystal oscillator needs300msstart-up time.4MHzThe crystal oscillator needs10ms start-up time.

CPUinstruction cycle

surface2-1 CSU8RP1001 CPUInstruction Cycle Register List

land site	name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	power-on reset value
14h	MCK	M7_CK	M6_CK	M5_CK	M4_CK	M3_CK	M2_CK	M1_CK	M0_CK	01000000

Users can setM0_CK,M1_CK,M2_CKandM3_CKto select the instruction cycle (the user must ensure that the output of the oscillator is stable when switching to the external oscillator; generally add aNOPinstruction).

surface2-2Instruction Cycle Selection List

M4_CK	M3_CK	M2_CK	M1_CK	M0_CK	MCK(KHz)		instruction cycle (KHz)	
X	X	0	0	0	ICK	4000	MCK/8	500
X	X	0	1	0	ICK	4000	MCK/16	250
X	X	1	0	0	ICK	4000	MCK/2	2000
X	X	1	1	0	ICK	4000	MCK/4	1000
0	0	0	0	1	ECK/4	8.192	MCK/8	1.024
0	0	0	1	1	ECK/4	8.192	MCK/16	0.512
0	0	1	0	1	ECK/4	8.192	MCK/2	4.096
0	0	1	1	1	ECK/4	8.192	MCK/4	2.048
0	1	0	0	1	ECK/8	4.096	MCK/8	0.512
0	1	0	1	1	ECK/8	4.096	MCK/16	0.256
0	1	1	0	1	ECK/8	4.096	MCK/2	2.048
0	1	1	1	1	ECK/8	4.096	MCK/4	1.024
1	0	0	0	1	ECK/4	1000	MCK/8	125
1	0	0	1	1	ECK/4	1000	MCK/16	62.5
1	0	1	0	1	ECK/4	1000	MCK/2	500
1	0	1	1	1	ECK/4	1000	MCK/4	250
1	1	0	0	1	ECK/8	500	MCK/8	62.5
1	1	0	1	1	ECK/8	500	MCK/16	31.25
1	1	1	0	1	ECK/8	500	MCK/2	250
1	1	1	1	1	ECK/8	500	MCK/4	125

ADCclock

surface2-1 ADCSampling frequency selection register

address	name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	power-on reset value
1Ah	NETC			PGA_F[3:0]						0000000u

CSU8RP1001middleADCThe sampling frequency used to sample the signal. The user setsPGA_FTo change the sampling clock, the settings are shown in the table2-12.

surface2-2 ADCsSampling frequency selection (usingICKTime)

PGA_F[3:0]	ADCF
0000	4M
0101	2M
1001	1M
1010	500K
1110	250K
1111	4M

surface2-3 ADCsOutput rate selection list

ADM[2:0]	ADCoutput rate
000	ADCF/64
001	ADCF/128
010	ADCF/256
011	ADCF/512
100	ADCF/1024
101	ADCF/2048
110	ADCF/4096
111	ADCF/8192

buzzer clock

surface2-1Buzzer Clock Register List

land site	name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	power-on reset value
14h	MCK	M7_CK	M6_CK	M5_CK	M4_CK	M3_CK	M2_CK	M1_CK	M0_CK	01000000
15h	PCK							S_BEEP		u00000uu

CSU8RP1001There is a buzzer clock for the buzzer source. The user setsM0_CK,M4_CKandS_BEEPRegister flag bit to change the buzzer clock, set as shown in the table2-15. (ECKby32KHz, 4MHZexample)

surface2-2Buzzer Clock Selection List

M4_CK	M0_CK	S_BEEP		clock source (KHz)		BEEP CLOCK(KHz)	
0	0	0	0	ICK	4000	ICK/256	16
0	0	0	1	ICK	4000	ICK/512	8
0	0	1	0	ICK	4000	ICK/1024	4
0	0	1	1	ICK	4000	ICK/2048	2
0	1	0	0	ECK	32	ECK/8	4.096
0	1	0	1	ECK	32	ECK/16	2.048
0	1	1	0	ECK	32	ECK/8	4.096
0	1	1	1	ECK	32	ECK/16	2.048
1	1	0	0	ECK	4000	ECK/1024	4
1	1	0	1	ECK	4000	ECK/2048	2
1	1	1	0	ECK	4000	ECK/4096	1
1	1	1	1	ECK	4000	ECK/8192	0.5

2.3.2 TMCLK(timer module input clock)

TMCLKused forCSU8RP1001timer. According to the table2-16, the user sets theM5_CKflag to select TMCLKFrequency of. (ECKby32KHz, 4MHzexample)

surface2-3 TMCLKpick list

M5_CK	M4_CK	M1_CK	M0_CK	clock source (KHz)		TMCLK(Hz)	
0	X	0	0	ICK	4000	ICK/1024	3906
0	X	1	0	ICK	4000	ICK/4096	976
0	0	0	1	ECK	32	ECK/32	1024
0	0	1	1	ECK	32	ECK/32	1024
0	1	0	1	ECK	4000	ECK/4096	1000
0	1	1	1	ECK	4000	ECK/16000	250
1	0	X	X	ECK	32	ECK/32	1024
1	1	X	X	ECK	4000	ECK/32	125000

TM2CLKused forCSU8RP1001timer2. According to the table2-17, the user sets theM4_CKas well asM0_CKflag to selectTMCLKFrequency of. (ECKby32KHz, 4MHzexample)

surface2-4 TM2CLKpick list

M4_CK	M0_CK	clock source (KHz)		TM2CLK(KHz)	
X	0	ICK	4000	ICK/4	1000
0	1	ECK	32	ECK/8	4
1	1	ECK	4000	ECK/8	500

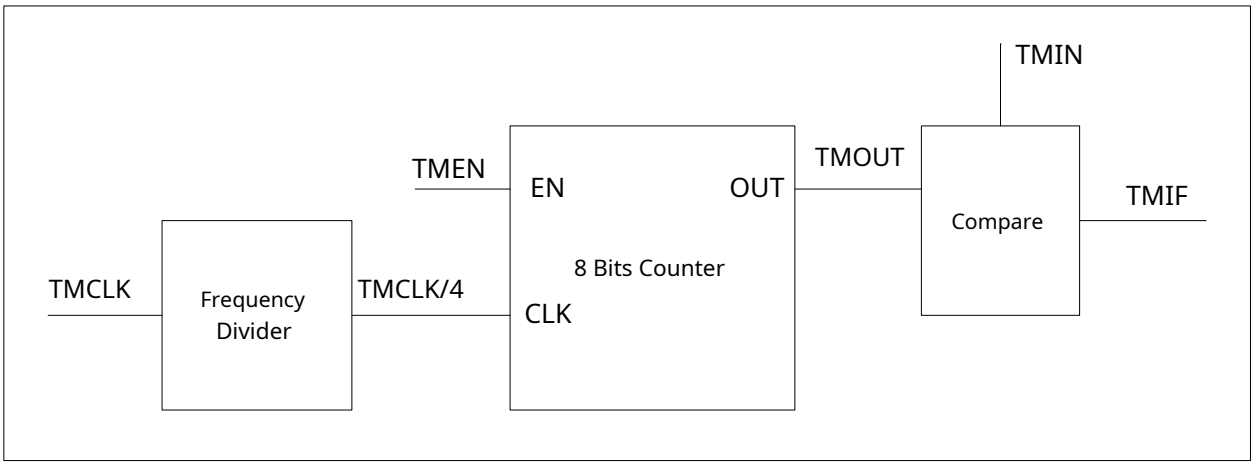
2.3.3 LCDCLK(LCDmodule input clock)

LCDCLKused forCSU8RP1001LCDmodule. According to the table2-18, the user sets theM5_CKflag to select TMCLKFrequency of. (ECKby32KHz, 4MHzexample)

surface2-5 LCDCLKpick list

M4_CK	M0_CK	M5_CK	clock source (KHz)		LCDSCK(KHz)	
X	0	0	ICK	4000	ICK/128	32
0	1	0	ECK	32	ECK	32
1	1	0	ECK	4000	ECK/512	8
0	X	1	ECK	32	ECK	32
1	X	1	ECK	4000	ECK	4000

2.4timer



picture2-1Functional block diagram of the timing module

The input to the timer module isTMCLK. The timer module integrates a divider pairTMCLKconduct4frequency division, the divided clock is used as8 bitsInput clock to the counter. When the user sets the enable flag of the timer module,8 bitsThe counter will start, TMOUT[7:0]will be from00hincrement to TMIN. User needs to setTMIN(Timer Module Interrupt Signal Selector) to select the timer timeout interrupt signal. When the time-out occurs, the interrupt flag will be set by itself, and the program counter will jump to04hto execute the interrupt service routine.

surface2-6Timer Register List

address	name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	power-on reset value
06H	INTF				TMIF					uuu0u000
07H	INTE	GIE			TMIE					0uu0u000
0EH	TMOUT	TMOUT[7:0]								00000000
0FH	TMCON	TRST				TMEN				10uu00uu
36H	TMIN	TMIN[7:0]								11111111

operate:

1. set upTMCLK, select the input for the timer module. 2.

set upTMIN, select the timer interrupt source.

3. Set register flags:TMIEandGIE, to enable the timer interrupt.

4. Set register flags:TMEN, to enable the timer module's8 bitscounter. 5. Clear

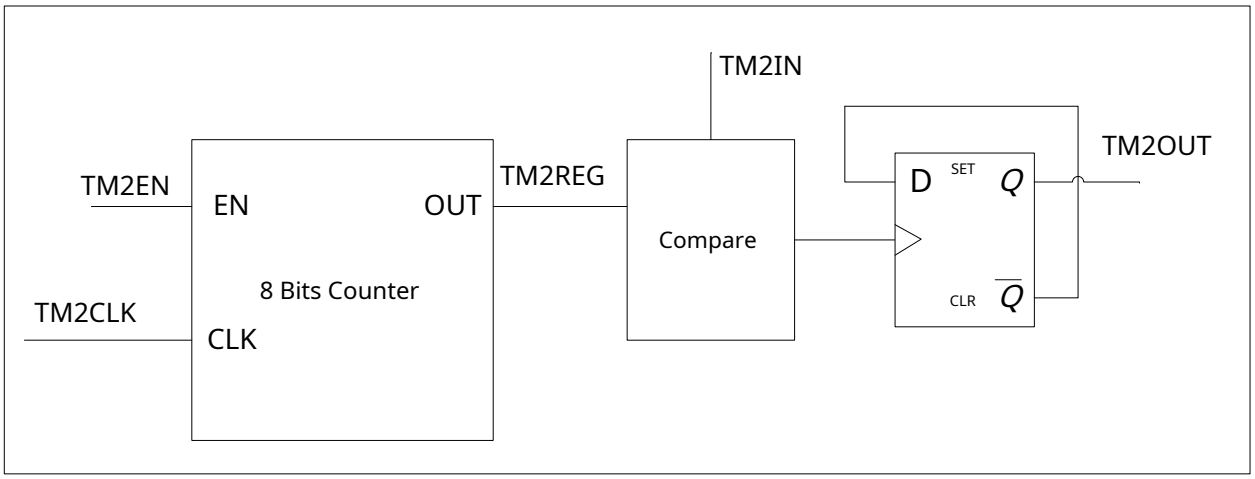
register flags:TRST, resets the counter of the timer module.

6. When the time-out occurs, the register flag bitTMIFwill self-reset, the program counter will reset to04h.

Timer time calculation method:

timer time = (TMIN+1)*4/TMCLK.

2.5timer2



picture2-2timer2Functional block diagram of the module

timer2The input to the module isTM2CLK. When the user sets a timer2module enable flag,8 bitsThe counter will start, TM2OUT[7:0]will be from00hincrement toTM2IN. User needs to setTM2IN(Timer Module Interrupt Signal Selector) to select the timer timeout interrupt signal. When a timed timeout occurs,TM2OUTThe output signal transitions.

surface2-7Timer Register List

address	name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	power-on reset value
0FH	TMCON						TM2EN			10uu00uu
27H	PT2MR				TMOEN					0uu00000
37H	TM2IN	TM2IN[7:0]								11111111

operate:

1. set upTM2CLK, for the timer2Module selection input.
2. set upTM2IN.
3. PT2.6set as the output port (PT2EN[6]set high). 4. WillTMOENset high,PT2.6output timer2clock.
5. Set register flags:TM2EN, to enable the timer module's8 bitscounter. 6. When a timed timeout occurs,TM2OUTThe output signal transitions.

timer2Clock cycle calculation method:

timer2clock cycles = (TM2IN+1)*2/TM2CLK.

2.6 I/O port

surface2-8 I/Oport register table

land site	name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	power-on reset value
06h	INTF				- -	- -		E1IF	E0IF	uuu0u000
07h	INTE	GIE			- -	- -		E1IE	E0IE	0uu0u000
20h	PT1	PT1[7:4]								0000uuuu
21h	PT1EN	PT1EN[7:4]								0000uuuu
22h	PT1PU	PT1PU[7:4]								0000uuuu
23h	AENB	AOENB3			AOENB2	AIENB1	AIENB3			0uu000uu
24h	PT2	PT2[7:6]				PT2[3:0]				00uu0000
25h	PT2EN	PT2EN[7:6]				PT2EN[3:0]				00uu0000
26h	PT2PU	PT2PU[7:6]				PT2PU[3:0]				00uu0000
27h	PT2MR	BZEN				E1M[1:0]		E0M[1:0]		u000uuuu
28h	PT3	PT3[7:3]								00000uuu
29h	PT3EN		PT3EN[6:3]							00000uuu
30h	PT3PU	PT3PU[7:3]								10000uuu

General Purpose in MicrocontrollersI/Omouth(GPIO) for general input and output functions. Users can passGPIOReceive data signals or transmit data to other digital devices.CSU8RP1001partGPIOCan be defined as other special functions. In this section, only theGPIOGeneric I/Oport function, special functions will be explained in the following chapters.

PT1register (address is20h)

characteristic	R/WX	R/WX	R/WX	R/WX	U-0	U-0	U-0	U-0
PT1	PT1[7:4]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1[7:4]:GPIO1port data flag
PT1[7] = GPIO1 bit 7data flag
PT1[6] = GPIO1 bit 6data flag
PT1[5] = GPIO1 bit 5data flag
PT1[4] = GPIO1 bit 4data flag

PT1ENregister (address is21h)

characteristic	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
PT1EN	PT1EN[7:4]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1EN[7:4]:GPIO1I/O control flags
PT1EN[7] = GPIO1 bit 7ofI/Ocontrol flag bit;0 =Defined as the input port,1 =Defined as an output port
PT1EN[6] = GPIO1 bit 6ofI/Ocontrol flag bit;0 =Defined as the input port,1 =Defined as an output port
PT1EN[5] = GPIO1 bit 5ofI/Ocontrol flag bit;0 =Defined as the input port,1 =Defined as an output port
PT1EN[4] = GPIO1 bit 4ofI/Ocontrol flag bit;0 =Defined as the input port,1 =Defined as an output port

characteristic(Property):
R =readable bit W =writable bit U =invalid bit
-n=value after power-on reset'1' =bit is set'0'=bit is cleared X =indeterminate bit

PT1PUregister (address is22h)

characteristic	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
PT1PU	PT1PU[7:4]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT1PU[7:4]:GPIO1Port pull-up resistor enable flag

PT1PU[7] = GPIO1 bit 7control flag bit;0 =Disconnect the pull-up resistor,1 =Use pull-up resistors

PT1PU[6] = GPIO1 bit 6control flag bit;0 =Disconnect the pull-up resistor,1 =Use pull-up resistors

PT1PU[5] = GPIO1 bit 5control flag bit;0 =Disconnect the pull-up resistor,1 =Use pull-up resistors

PT1PU[4] = GPIO1 bit 4control flag bit;0 =Disconnect the pull-up resistor,1 =Use pull-up resistors

AENBregister (address is23h)

characteristic	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	U-0	U-0
AENB	AOENB3			AOENB2	AIENB1	AIENB3		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 AOENB3:PT[6:3]Digital-to-analog channel selection signal 0 = PT3[6:3]All defined as analog channels 1 = PT3[6:3]All are defined as digital channels AOENB2: PT2[3],PT2[2]Digital-to-

Bit 4 analog channel selection signal 0 = PT2[3],PT2[2]Defined as an analog channel 1 = PT2[3],PT2[2]Defined as a digital channel AIENB1: PT1[4]Digital-to-analog channel selection signal 0 =

Bit 3 PT1[4]Defined as an analog channel 1 = PT1[4]Defined as a digital channel AIENB3:Reset signal selection signal

Bit 2

0= PT3[7]Defined as the reset signal

1 = PT3[7]Defined as a digital channel

PT2register (address is24h)

characteristic	R/WX	R/WX	U-0	U-0	R/WX	R/WX	R/WX	R/WX
PT2	PT2[7:6]				PT2[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT2[7:0]:GPIO2port data flag

PT2[7] = GPIO2 bit 7data flag bit

PT2[6] = GPIO2 bit 6data flag bit

PT2[3] = GPIO2 bit 3data flag bit

PT2[2] = GPIO2 bit 2data flag bit

PT2[1] = GPIO2 bit 1data flag bit

PT2[0] = GPIO2 bit 0data flag bit

characteristic(Property):

R =readable bit W =writable bit U =invalid bit

-n=Value after power-on reset '1' =bit is set '0'=bit is cleared X =indeterminate bit

PT2ENregister (address is25h)

characteristic	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PT2EN	PT2EN[7:6]				PT2EN[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT2EN[7:0]:GPIO 2I/O control flags

PT2EN[7] = GPIO2 bit 7ofI/Ocontrol flag bit;0 =Defined as the input port,1 =Defined as an output port

PT2EN[6] = GPIO2 bit 6ofI/Ocontrol flag bit;0 =Defined as the input port,1 =Defined as an output port

PT2EN[3] = GPIO2 bit 3ofI/Ocontrol flag bit;0 =Defined as the input port,1 =Defined as an output port

PT2EN[2] = GPIO2 bit 2ofI/Ocontrol flag bit;0 =Defined as the input port,1 =Defined as an output port

PT2EN[1] = GPIO2 bit 1ofI/Ocontrol flag bit;0 =Defined as the input port,1 =Defined as an output port

PT2EN[0] = GPIO2 bit 0ofI/Ocontrol flag bit;0 =Defined as the input port,1 =Defined as an output port

PT2PUregister (address is26h)

characteristic	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PT2PU	PT2PU[7:6]				PT2PU[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT2PU[7:0]:GPIO2Port pull-up resistor enable flag

PT2PU[7] = GPIO2 bit 7control flag bit;0 =Disconnect the pull-up resistor,1 =Use pull-up resistors

PT2PU[6] = GPIO2 bit 6control flag bit;0 =Disconnect the pull-up resistor,1 =Use pull-up resistors

PT2PU[3] = GPIO2 bit 3control flag bit;0 =Disconnect the pull-up resistor,1 =Use pull-up resistors

PT2PU[2] = GPIO2 bit 2control flag bit;0 =Disconnect the pull-up resistor,1 =Use pull-up resistors

PT2PU[1] = GPIO2 bit 1control flag bit;0 =Disconnect the pull-up resistor,1 =Use pull-up resistors

PT2PU[0] = GPIO2 bit 0control flag bit;0 =Disconnect the pull-up resistor,1 =Use pull-up resistors

PT2MRregister (address is27h)

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

Bit 7 BZEN: Buzzer enable flag

1 =enable the buzzer function,GPIO2mouthbit 7Defined as the buzzer output port 0 =

Do not enable the buzzer function,GPIO2mouthbit 7define genericI/Omouth E1M[1:0]

Bit 3-2 :GPIO2mouthbit 1interrupt trigger mode

11 =External Interrupt1(GPIO2mouthbit 1) fires when the state changes

10 =External Interrupt1(GPIO2mouthbit 1) fires when the state changes

01 =External Interrupt1(GPIO2mouthbit 1) is a rising edge trigger 00 =

External Interrupt1(GPIO2mouthbit 1) is a falling edge trigger E0M[1:0]:

Bit 1-0 GPIO2mouthbit 0interrupt trigger mode

11 =External Interrupt0(GPIO2mouthbit 0) fires when the state changes

10 =External Interrupt0(GPIO2mouthbit 0) fires when the state changes

01 =External Interrupt0(GPIO2mouthbit 0) is a rising edge trigger 00 =

External Interrupt0(GPIO2mouthbit 0) is a falling edge trigger

characteristic(Property):

R =readable bit W =writable bit U =invalid bit

-n=Value after power-on reset '1' =bit is set '0'=bit is cleared X =indeterminate bit

PT3register (address is28h)

characteristic	R/WX	R/WX	R/WX	R/WX	R/WX	U-0	U-0	U-0
PT3	PT3[7:3]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT3[7:3]:GPIO3port data flag
PT3[7] = GPIO3 bit 7data flag bit
PT3[6] = GPIO3 bit 6data flag bit
PT3[5] = GPIO3 bit 5data flag bit
PT3[4] = GPIO3 bit 4data flag bit
PT3[3] = GPIO3 bit 3data flag bit

PT3ENregister (address is29h)

characteristic	R-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
PT3EN		PT3EN[6:3]						
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT3EN[6:3]:GPIO 3I/O control flags
PT3EN[6] = GPIO3 bit 6ofI/Ocontrol flag bit;0 =Defined as the input port,1 =Defined as an output port
PT3EN[5] = GPIO3 bit 5ofI/Ocontrol flag bit;0 =Defined as the input port,1 =Defined as an output port
PT3EN[4] = GPIO3 bit 4ofI/Ocontrol flag bit;0 =Defined as the input port,1 =Defined as an output port
PT3EN[3] = GPIO3 bit 3ofI/Ocontrol flag bit;0 =Defined as the input port,1 =Defined as an output port

PT3PUregister (address is2Ah)

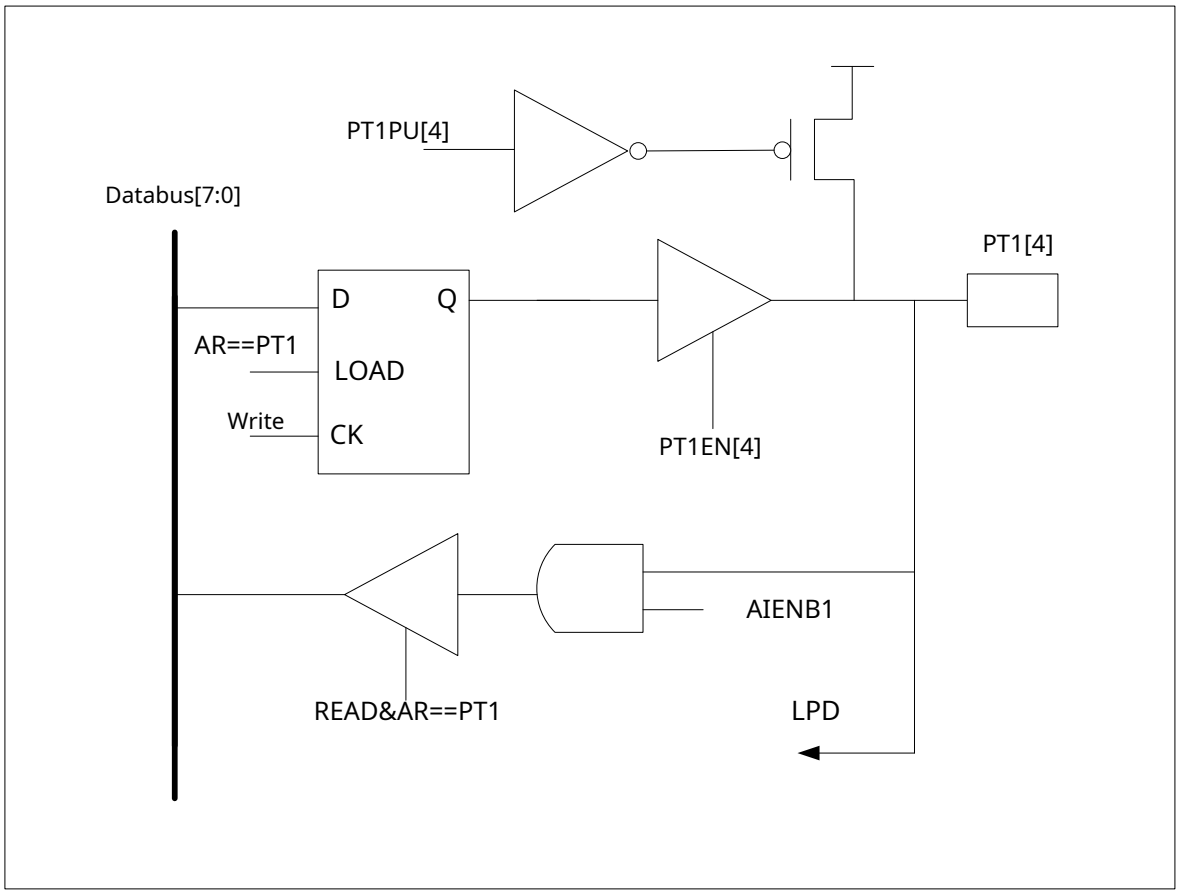
characteristic	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0
PT3PU	PT3PU[7:3]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-0 PT3PU[7:3]:GPIO3Port pull-up resistor enable flag
PT3PU[7] = GPIO3 bit 7control flag bit;0 =Disconnect the pull-up resistor,1 =Use pull-up resistors
PT3PU[6] = GPIO3 bit 6control flag bit;0 =Disconnect the pull-up resistor,1 =Use pull-up resistors
PT3PU[5] = GPIO3 bit 5control flag bit;0 =Disconnect the pull-up resistor,1 =Use pull-up resistors
PT3PU[4] = GPIO3 bit 4control flag bit;0 =Disconnect the pull-up resistor,1 =Use pull-up resistors
PT3PU[3] = GPIO3 bit 3control flag bit;0 =Disconnect the pull-up resistor,1 =Use pull-up resistors

characteristic(Property):

R =readable bit W =writable bit U =invalid bit
-n=Value after power-on reset '1' =bit is set '0'=bit is cleared X =indeterminate bit

2.6.1digital with analog input channelsI/Omouth:PT1[4]



picture2-3 PT1[7:0]Functional block diagram

GPIO1mouth(PT1[4]) functional block diagram as shown in the figure2-8shown.GPIOThe main function is for the exchange between the data bus and the interface. via control register flagsPT1EN[4]to determine whether the interface is an input or an output. Input and output functions and related functions are explained below.

-enter

GPIO1interfacebit 4(PT1[4]) can be used to input digital or analog signals. User should control register flagsAIENB1Determines the type of input signal. ifAIENB1is set (ie1),GPIO1The AND gate in the interface allows the digital signal to be connected to the data bus, otherwise, the input signal is defined as an analog signal and the analog signal is sent to the corresponding functional block

- output

CSU8RP1001through the interiorDThe flip-flop outputs a digital signal. when the program passesPT1When outputting data, the data is first sent to the data bus, when there is a write signal andAR(CSU8RP1001internal device address pointer) points toPT1when, thenDflip-flop will latch data fromPT1port output.

-Pull-up resistor

CSU8RP1001existPT1The port integrates an internal pull-up resistor function, and the pull-up resistor is approximately100KΩ(The pull-up current is approximately 30uA. Before the program is going to run into sleep mode, it must be disabledPT1PU). can be flagged by the control registerPT1PU[4]Determines whether to connect a pull-up resistor. When the interface is connected to a pull-up resistor, the input data defaults to high (ie1).

surface2-9 PT1register list

land site	name	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	power-on reset bit value
20h	PT1				PT1[4]					0000uuuu
21h	PT1EN				PT1EN[4]					0000uuuu
22h	PT1PU				PT1PU[4]					0000uuuu
23h	AENB					AIENB1				0u0000uu

Read data operation:

1. Clear register flags:PT1EN[4].PT1 [4]is defined as an input port.
2. Set register flags:PT1PU[4].PT1 [4]The interface is connected to an internal pull-up resistor. 3. If the input signal is a digital signal, set the register flag bit:AIENB1.
4. If the input signal is an analog signal, clear the register flag bit:AIENB1,At the same time willPT1EN[4]set low (set as digital input enter),PT1PU [4]Set low (no pull-up resistor).
5. need to be enabled firstVDDAvoltage regulator circuit, the analog input can work normally. 6. After the signal is input from the outside, the user canPT1[4]get data.

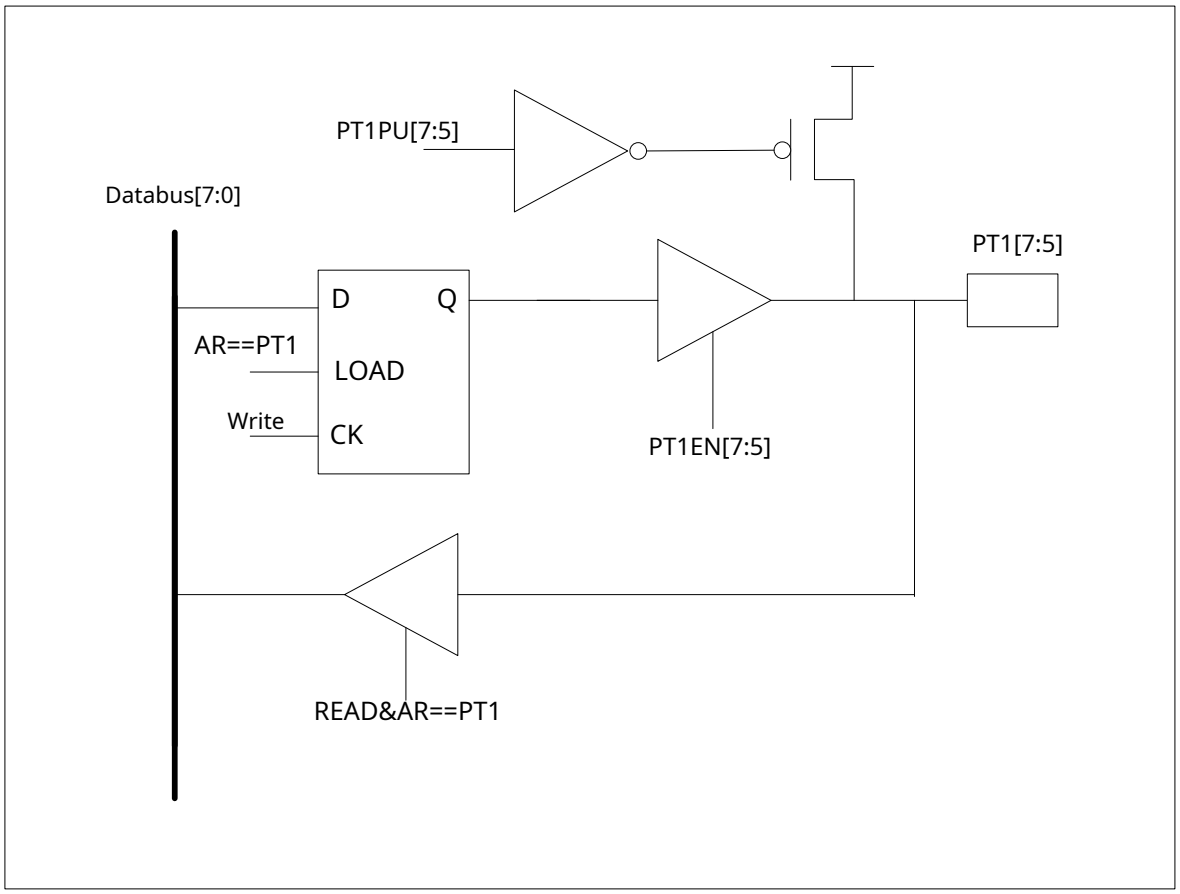
Write data operation:

1. set register flagPT1EN[4].PT1 [4]is defined as an output port.
2. Set the corresponding register flagPT1PU[4].PT1 [4]Connect to an internal pull-up resistor.
3. set upPT1[4]As data output, the internalDflip-flop will latch data untilPT1[4]data changes.

Note the operation:

1. In order to maintain low operating current in sleep mode, the situation of floating digital input should be avoided.
2. existI/Omouth withVDDA small resistor in parallel between (approximately10KΩ),whenPT1PU[4]When set, the output drive can be increased moving current,

2.6.2numberI/Omouth:PT1[7:5]



picture2-4 PT1 [7:5]Functional block diagram

GPIO1mouth(PT1[7:5]) functional block diagram as shown in the figure2-8shown.GPIOThe main function is for the exchange between the data bus and the interface. via control register flagsPT1EN[7:5]to determine whether the interface is an input or an output. Input and output functions and related functions are explained below.

-enter

GPIO1interfacebit 7~bit 5(PT1[7:5]) can be used to enter numbers. whenPT1EN[n]set to0hour,PT1[7:5]Set to digital input.

- output

CSU8RP1001through the interiorDThe flip-flop outputs a digital signal. when the program passesPT1When outputting data, the data is first sent to the data bus, when there is a write signal andAR(CSU8RP1001internal device address pointer) points toPT1when, thenDflip-flop will latch data fromPT1port output.

- Pull-up resistor

CSU8RP1001existPT1The port integrates an internal pull-up resistor function, and the pull-up resistor is approximately100KΩ(The pull-up current is approximately30uA. Before the program is going to run into sleep mode, it must be disabledPT1PU). can be flagged by the control registerPT1PU[7:5]Determines whether to connect a pull-up resistor. When the interface is connected to a pull-up resistor, the input data defaults to high (ie1).

surface2-10 PT1register list

land site	name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	power-on reset bit value
20h	PT1	PT1[7:4]								0000uuuu
21h	PT1EN	PT1EN[7:4]								0000uuuu
22h	PT1PU	PT1PU[7:4]								0000uuuu

Read data operation:

1. Clear register flags:PT1EN[n](nis controlled by the userbit).PT1 [n]is defined as an input port. 2. Set register flags:PT1PU[n].PT1 [n]The interface is connected to an internal pull-up resistor. 3. After the signal is input from the outside, the user canPT1[n]get data.

Write data operation:

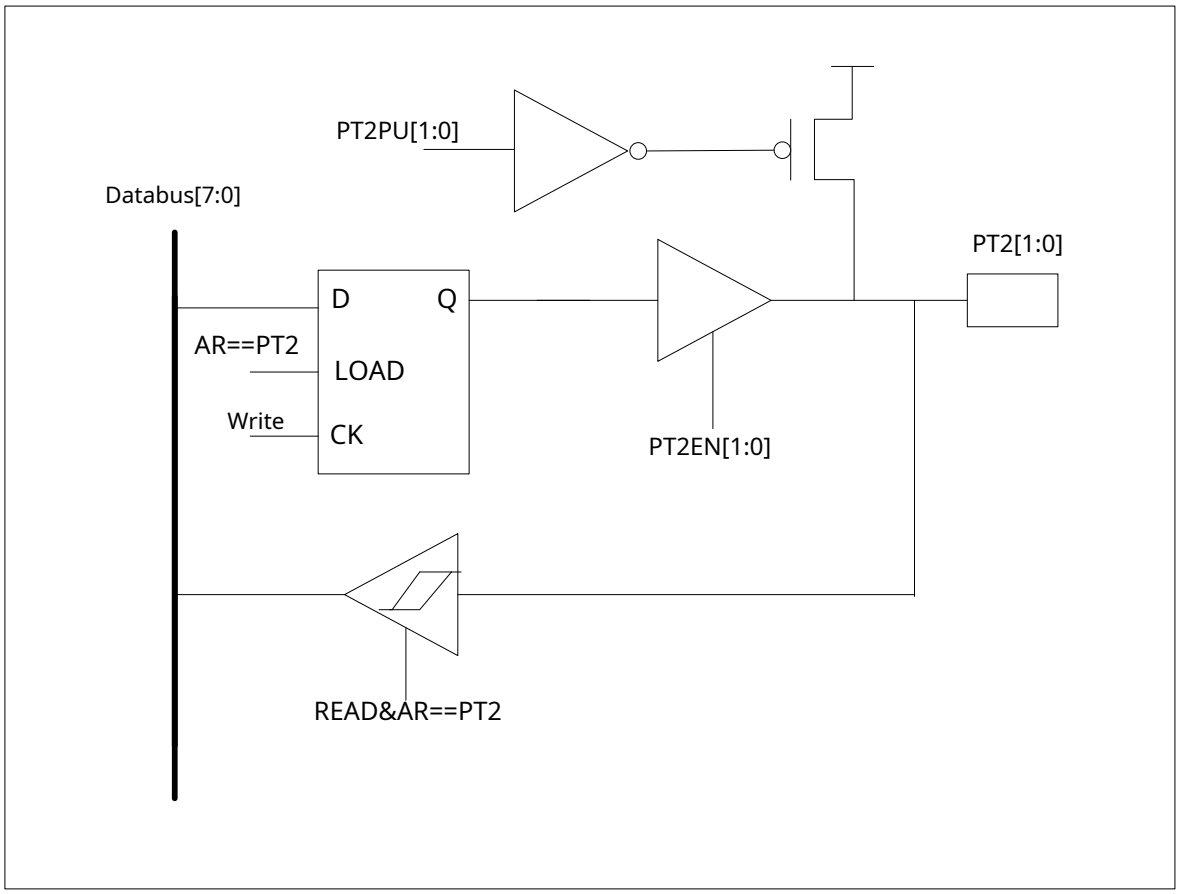
1. set register flagPT1EN[n].PT1 [n]is defined as an output port.

2. Set the corresponding register flagPT1PU[n].PT1 [n]Connect to an internal pull-up resistor.

3. set upPT1[n]As data output, the internalDflip-flop will latch data untilPT1[n]data changes. Note the operation:

1. existI/Omouth withVDDA small resistor in parallel between (approximately10KΩ),whenPT1PU[n]When set, the output drive can be increased moving current.

2.6.3numberI/OPort and external interrupt input:PT2[1:0]



picture2-5 PT2[1:0]Functional block diagram

GPIO2oralbit1~0(PT2[1:0]) functional block diagram as shown in the figure2-9shown. thisGPIOThe main function of the port is to input/output data between the data bus and the port. via control register flagsPT2EN[1:0]to determine whether the interface is an input or an output. Input and output functions and related functions are explained as follows:

-enter

GPIO2mouthbit1~0(PT2[1:0]) can be used as an external interrupt interfaceINT1andINT0, or as an ordinaryI/Omouth. by controlling INTEregister flagsE0IEandE1IEto determine whether to enable interrupts. The interrupt trigger mode is flagged by the register:E0M[1:0], E1M[1:0]Decide.

- output

CSU8RP1001through the interiorDTriggers output digital data. when the program passesPT2When outputting data, the data is first sent to the data bus, when there is a write signal andAR(CSU8RP1001internal device address pointer) points toPT2when, thenDflip-flop will latch data fromPT2port output.

-Pull-up resistor

CSU8RP1001existPT2The port integrates an internal pull-up resistor function, and the pull-up resistor is approximately100KΩ(The pull-up current is approximately 30uA. Before the program is going to run into sleep mode, it must be disabledPT2PU). can be flagged by the control registerPT2PU[1:0]Determines whether to connect a pull-up resistor. When the interface is connected to a pull-up resistor, the input data defaults to high (ie1).

surface2-11 PT2register list

land site	name	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	power-on reset bit value
06h	INTF				- -	- -	- -	E1IF	E0IF	uuu0u000
07h	INTE	GIE			- -	- -	- -	E1IE	E0IE	0uu0u000
24h	PT2	PT2[7:6]				PT2[3:0]				00uu0000
25h	PT2EN	PT2EN[7:6]				PT2EN[3:0]				00uu0000
26h	PT2PU	PT2PU[7:6]				PT2PU[3:0]				00uu0000
27h	PT2MR					E1M[1:0]		E0M[1:0]		0uu00000

Read data operation:

1. Clear register flags:PT2EN[n](nis controlled by the userbit).PT2[n]is defined as an input port. 2. Set register flags:PT2PU[n].PT2[n]The interface is connected to an internal pull-up resistor. 3. After the signal is input from the outside, the user canPT2[n]get data.

Write data operation:

1. set register flagPT2EN[n].PT2[n]is defined as an output port.
2. Set the corresponding register flagPT2PU[n].PT2[n]Connect to an internal pull-up resistor.
3. set upPT2[n]As data output, the internalDflip-flop will latch data untilPT2[n]data changes.

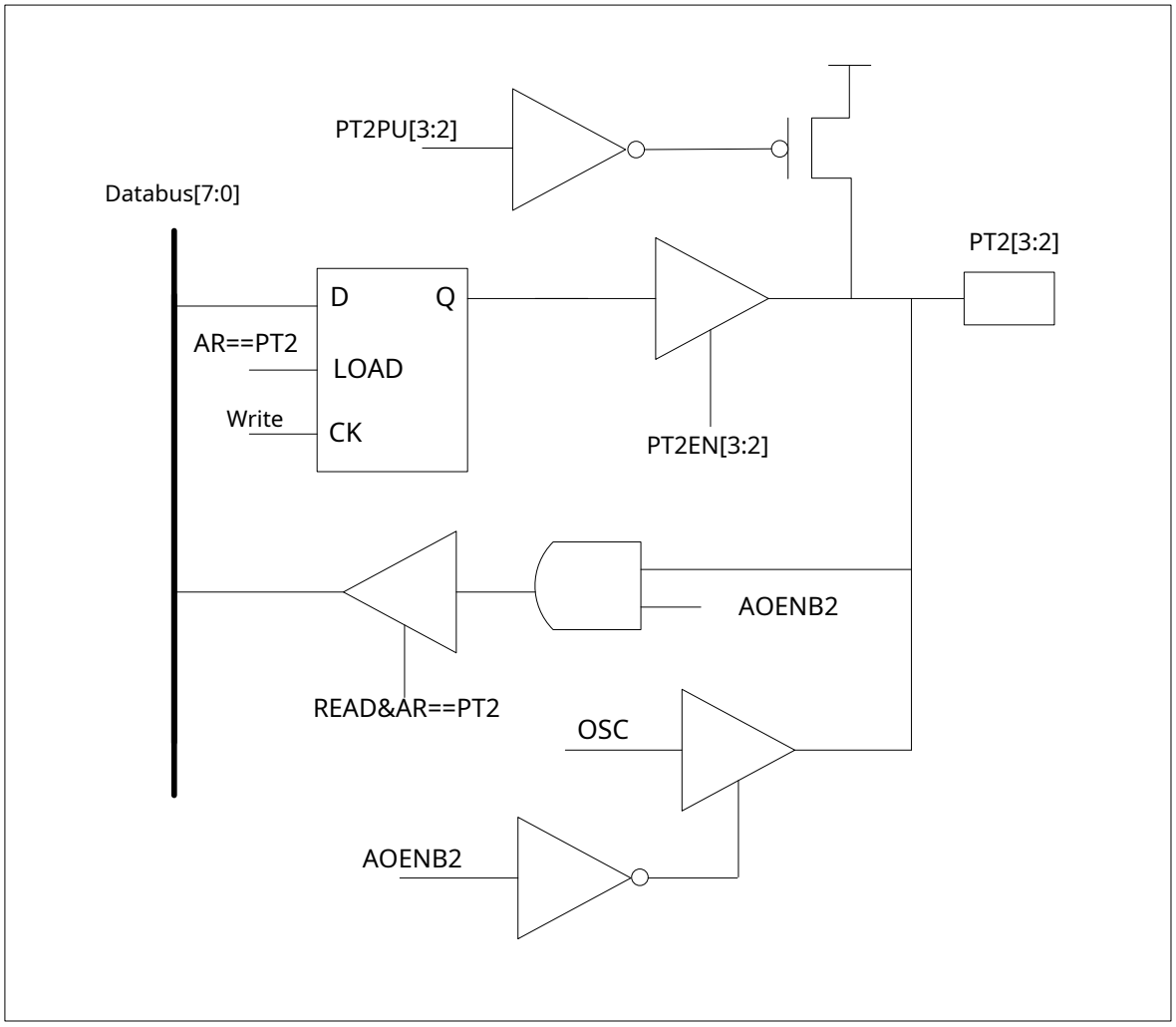
External interrupt operation (take falling edge trigger as an example)

1. clear register flagsPT2EN[n].PT2[n]is defined as an input port.
2. Set the corresponding register flagPT2PU[n].PT2[n]Connect to an internal pull-up resistor.
3. setE0M[1:0]for00,definitionINT0The interrupt trigger mode is "Falling Edge Trigger". 4. set E1M[1:0]for00,definitionINT1The interrupt trigger mode is "Falling Edge Trigger".

Note the operation:

1. existI/Omouth withVDDA small resistor in parallel between (approximately10KΩ),whenPT2PU[n]When set, the output drive can be increased moving current,

2.6.4digital with external crystalI/OmouthPT2[3:2]



picture2-6 PT2[3:2]Functional block diagram

GPIO2mouthbit 3:2(PT2[3:2]) is shown in the functional block diagram. thisGPIOThe main function of the port is to input/output data between the data bus and the port. via control register flagsPT2EN[3:2]to determine whether the interface is an input or an output. Input and output functions and related functions are explained as follows:

- enter

  GPIO2mouthbit3:2(PT2[3:2]) can be used as a genericI/Omouth.

- output

  CSU8RP1001use internalDThe latches output digital data. when the program passesPT2When outputting data, the data is first sent to the data bus, when there is a write signal andAR(CSU8RP1001internal device address pointer) points toPT2when, thenDflip-flop will latch data from PT2port output.

- Pull-up resistor

  CSU8RP1001existPT2The port integrates an internal pull-up resistor function, and the pull-up resistor is approximately100KΩ(The pull-up current is approximately30uA. Before the program is going to run into sleep mode, it must be disabledPT2PU). can be flagged by the control registerPT2PU[3:2]Determines whether to connect a pull-up resistor. When the interface is connected to a pull-up resistor, the input data defaults to high (ie1).

surface2-12 PT2register list

land site	name	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	power-on reset value
24h	PT2	PT2[7:6]				PT2[3:0]				00uu0000
25h	PT2EN	PT2EN[7:6]				PT2EN[3:0]				00uu0000
26h	PT2PU	PT2PU[7:6]				PT2PU[3:0]				00uu0000
27h	PT2MR	BZEN		PM2EN	PM1EN					0uu00000

Read data operation:

1. Clear register flags:PT2EN[n](nis controlled by the userbit).PT2[n]is defined as an input port. 2. Set register flags:PT2PU[n].PT2[n]The interface is connected to an internal pull-up resistor. 3. After the signal is input from the outside, the user canPT2[n]get data.

Write data operation:

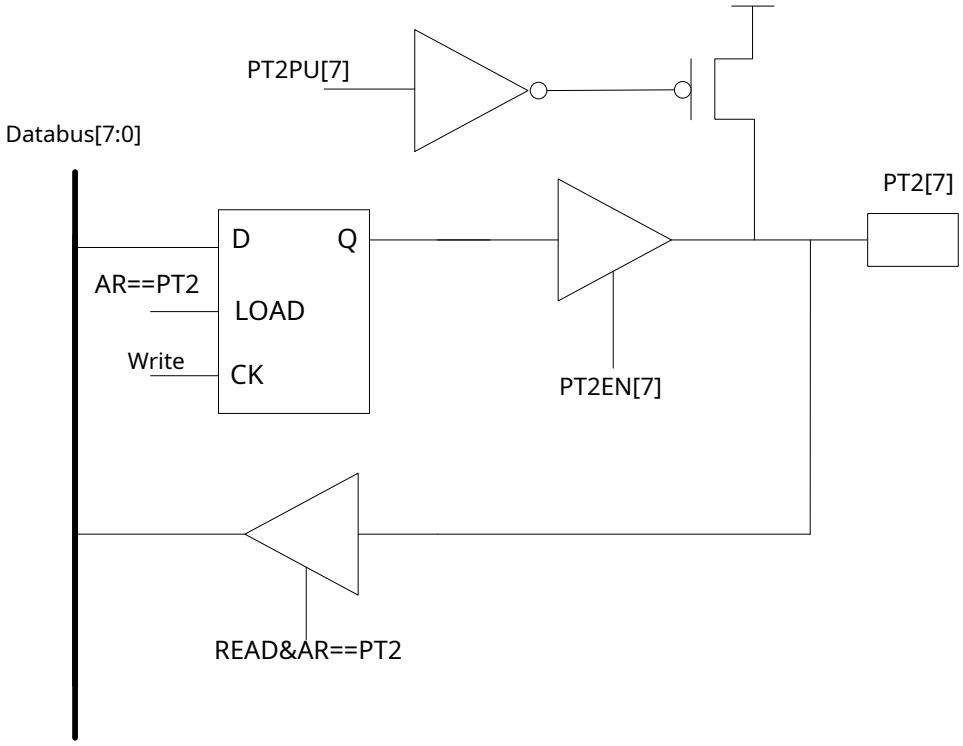
1. Set the corresponding register flag bit:PT2EN[n].PT2[n]is defined as an output port. 2. Set the corresponding register flag bit:PT2PU[n].PT2[n]Connect to an internal pull-up resistor.

3. set upPT2[n]As data output, the internalDflip-flop will latch data untilPT2[n]data changes.

Note the operation:

1. existI/Omouth withVDDA small resistor in parallel between (approximately10KΩ),whenPT2PU[n]When set, the output drive can be increased moving current.

2.6.5Digital with buzzer outputI/Ointerface:PT2[7]



picture2-7 PT2[7]Functional block diagram

GPIO2mouthbit7(PT2[7]), the functional block diagram is shown in Fig.2-11shown. thisGPIOThe main function of the port is to input/output data between the data bus and the port. via control register flagsPT2EN[7]to determine whether the interface is an input or an output. Input and output functions and related functions are explained as follows:

enter

GPIO2mouthbit 7(PT2[7]) can be used as a buzzer output port, or as a commonI/Ointerface. By setting the register flag BZEN Determines whether to enable the buzzer output.

output

CSU8RP1001use internalDThe latches output digital data. when the program passesPT2When outputting data, the data is first sent to the data bus, when there is a write signal andAR(CSU8RP1001internal device address pointer) points toPT2when, thenDflip-flop will latch data from PT2port output.

Pull-up resistor

CSU8RP1001existPT2The port integrates an internal pull-up resistor function, and the pull-up resistor is approximately100KΩ(The pull-up current is approximately30uA. Before the program is going to run into sleep mode, it must be disabledPT2PU). can be flagged by the control registerPT2PU[7]Determines whether to connect a pull-up resistor. When the interface is connected to a pull-up resistor, the input data defaults to high (ie1)

surface2-13 PT2[7]register list

land site	name	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	power-on reset value
24h	PT2	PT2[7:6]				PT2[3:0]				00uu0000
25h	PT2EN	PT2EN[7:6]				PT2EN[3:0]				00uu0000
26h	PT2PU	PT2PU[7:6]				PT2PU[3:0]				00uu0000

Read data operation:

1. Clear register flags:PT2EN[7].PT2[7]is defined as an input port.

2. Set the corresponding register flag bit:PT2PU[7].PT2[7]The interface is connected to an internal pull-up resistor.

3. After the signal is input from the outside, the user canPT2[7]get data.

Write data operation:

1. Set the corresponding register flag bit:PT2EN[7].PT2[7]is defined as an output port. 2. Set the corresponding register flag bit:PT2PU[7].PT2[7]Connect to an internal pull-up resistor.

3. set upPT2[7]As data output, the internalDflip-flop will latch data untilPT2[7]data changes.

Buzzer output operation:

1. set register flagPT2EN[7].PT2[7]Defined as an output port. 2

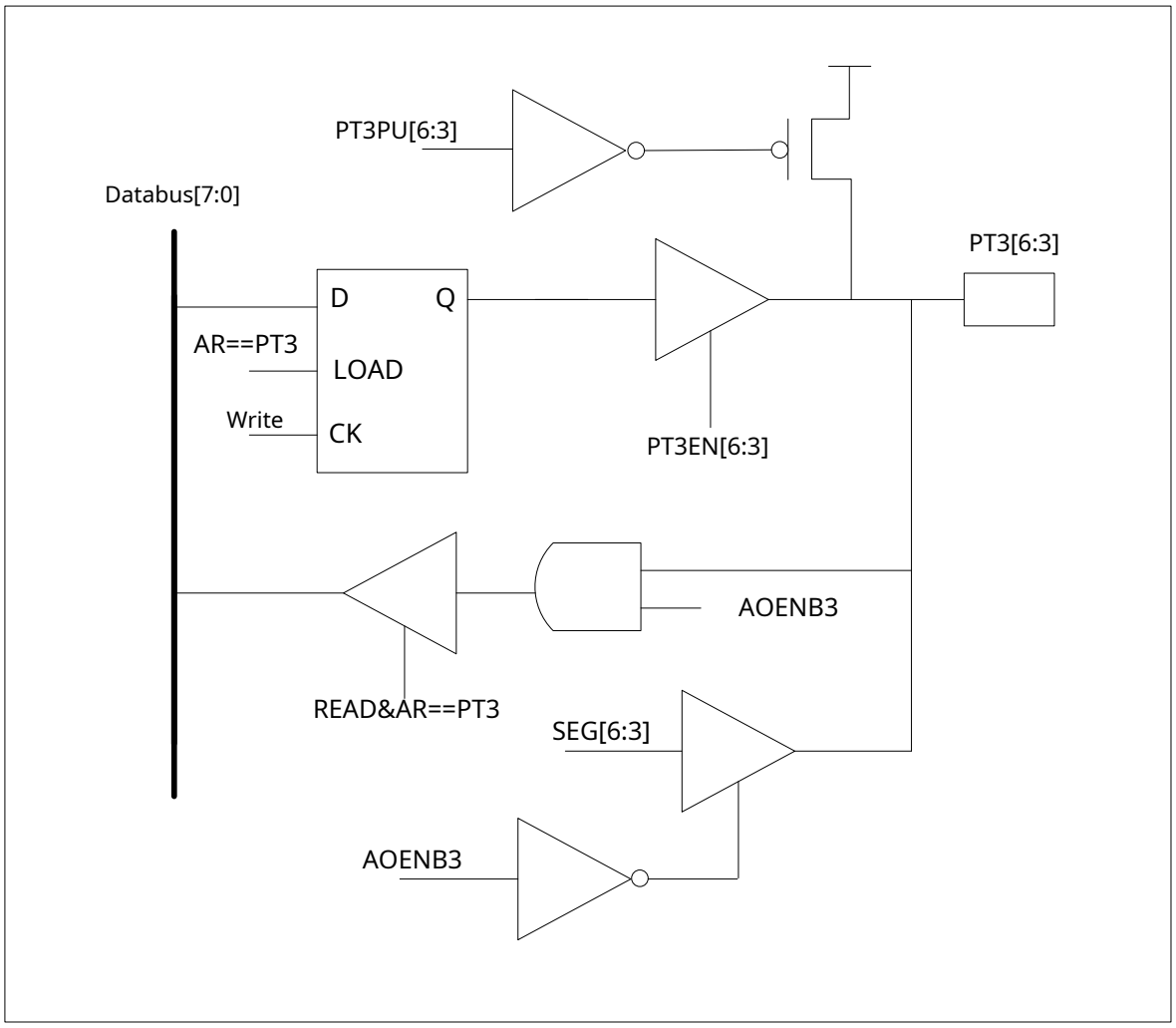
. set register flagS_BEEP, set the buzzer frequency.

3. set register flagBZEN.PT2[7]Just as the buzzer output.

Note the operation:

1. existI/Omouth withVDDA small resistor in parallel between (approximately10KΩ),whenPT2PU[7]When set, the output drive can be increased moving current.

2.6.6bringLCD Segmentdigital to drive the outputI/Omouth:PT3[6:3]



picture2-8 PT3 [6:3]Functional block diagram

GPIO3mouth(PT3[6:3]) functional block diagram as shown in the figure2-12shown.GPIOThe main function is for the exchange between the data bus and the interface. via control register flagsPT3EN[6:3]to determine whether the interface is an input or an output. can also be used asLCDofSeg[6:3] Output. Depend onAOENB3decide what to doI/Ouse or asLCDDrive output is used. The default isLCDdrive output. Input and output functions and related functions are explained below.

- enter

  GPIO3interfacebit 6~bit 3(PT3[6:3]) can be used to enter numbers. User should control register flagsPT3ENfor0.

- output

  GPIO3interfacebit 6~bit 3(PT3[6:3]) can be used to input digital or analog signal output (onlysegsignal output) the user should control the register flagsAOENB3Determines the type of output signal. ifAOENB3is set (ie1),GPIO3The AND gates in the interface allow digital signals to be connected to the data bus, otherwise, the output signals are defined as analog signals.

CSU8RP1001through the interiorDThe flip-flop outputs a digital signal. when the program passesPT3When outputting data, the data is first sent to the data bus, when there is a write signal andAR(CSU8RP1001internal device address pointer) points toPT3when, thenDflip-flop will latch data fromPT3port output.

-Pull-up resistor

CSU8RP1001existPT3The port integrates an internal pull-up resistor function, and the pull-up resistor is approximately100KΩ(The pull-up current is approximately30uA. Before the program is going to run into sleep mode, it must be disabledPT3PU). can be flagged by the control registerPT3PU[6:3]Determines whether to connect a pull-up resistor. When the interface is connected to a pull-up resistor, the input data defaults to high (ie1).

surface2-14 PT3register list

address	name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	power-on reset bit value
28h	PT3	PT3[7:3]								00000uuu
29h	PT3EN		PT3EN[6:3]							u0000uuu
2Ah	PT3PU	PT3PU[7:3]								00000uuu
23h	AENB	AOENB3								0u0000uu

Read data operation:

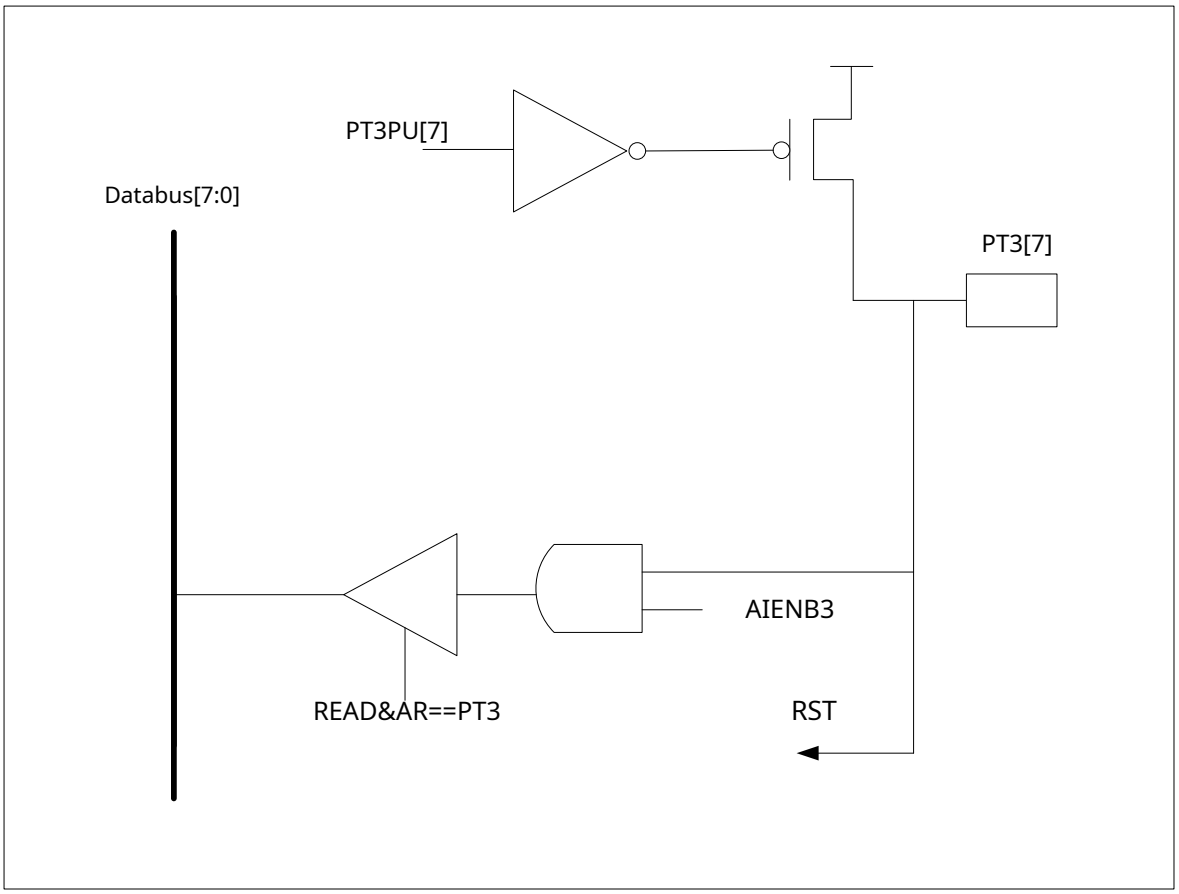
1. Clear register flags:PT3EN[n](nis controlled by the userbit).PT3 [n]is defined as an input port. 2. Set register flags:PT3PU[n].PT3 [n]The interface is connected to an internal pull-up resistor. 3. After the signal is input from the outside, the user canPT3[n]get data.

Write data operation:

1. set register flagPT3EN[n].PT3 [n]is defined as an output port.
2. Set the corresponding register flagPT3PU[n].PT3 [n]Connect to an internal pull-up resistor.
3. If the output signal is a digital signal, set the register flag bit:AOENB3;

If the output signal is an analog signal, clear the register flag bit:AOENB3. 4.
need to be enabled firstLCDcircuit,SEGto work properly.
5. set upPT3[n]As data output, the internalDflip-flop will latch data untilPT3[n]data changes.

2.6.7Digital input with reset:PT3[7]



picture2-9 PT3[7]Functional block diagram

GPIO3mouth(PT3[7]) functional block diagram as shown in the figure2-13 PT3[7]The functional block diagram is shown. Depend onAIENB3Decide whether to use it as a digital input port or as a reset signal input. Default is reset input. The related functions of the digital inputs are explained below.

- enter

  GPIO3interfacebit 7(PT3[7]) can be used to enter numbers. User should control register flagsAIENB3for1.

- Pull-up resistor

  CSU8RP1001existPT3The port integrates an internal pull-up resistor function, and the pull-up resistor is approximately100KΩ(The pull-up current is approximately30uA. Before the program is going to run into sleep mode, it must be disabledPT3PU). can be flagged by the control registerPT3PU[7]Determines whether to connect a pull-up resistor. When the interface is connected to a pull-up resistor, the input data defaults to high (ie1).

surface2-15 PT3register list

address	name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	power-on reset bit value
28h	PT3	PT3[7:3]								00000uuu
29h	PT3EN		PT3EN[6:3]							u0000uuu
2Ah	PT3PU	PT3PU[7:3]								00000uuu
23h	AENB	AOENB3								0u0000uu

Read data operation:

1. Set register flags:AIENB3.PT3 [7]Defined as a digital input port.

2. Set register flags:PT3PU[7].PT3 [7]The interface is connected to an internal pull-up resistor. 3.

After the signal is input from the outside, the user canPT3[7]get data.

2.6.8 LCD segmentas digital output

LCDofsegmentport is configuredNETDin the registerLCDCHandLEVEL_SAfter the register, it can be used as a digital IO port for output.

address	name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	power-on reset value
1Bh	NETD	CHP_VPP	DIVS	LCD_CH	LEVEL_S	VLCDX[1:0]		LCDREF[1:0]		00000000
58h	LCDENR	LCDCKS[1:0]		LCDEN	LCDWS	LEVEL	LCD_DUTY[1:0]		ENPMPL	00000110

Instructions:

1WillLCDCHset high,segport can be used as a digital output.

2set upLEVEL_SSignal

whenLEVEL_Sfor0Timelcd_xregister value0X0Foutput high level,
lcd_xregister value0x00output low level when
LEVEL_Sfor1Timelcd_xregister value0X0Foutput low level,
lcd_xregister value0x00output high level

3Enhancements

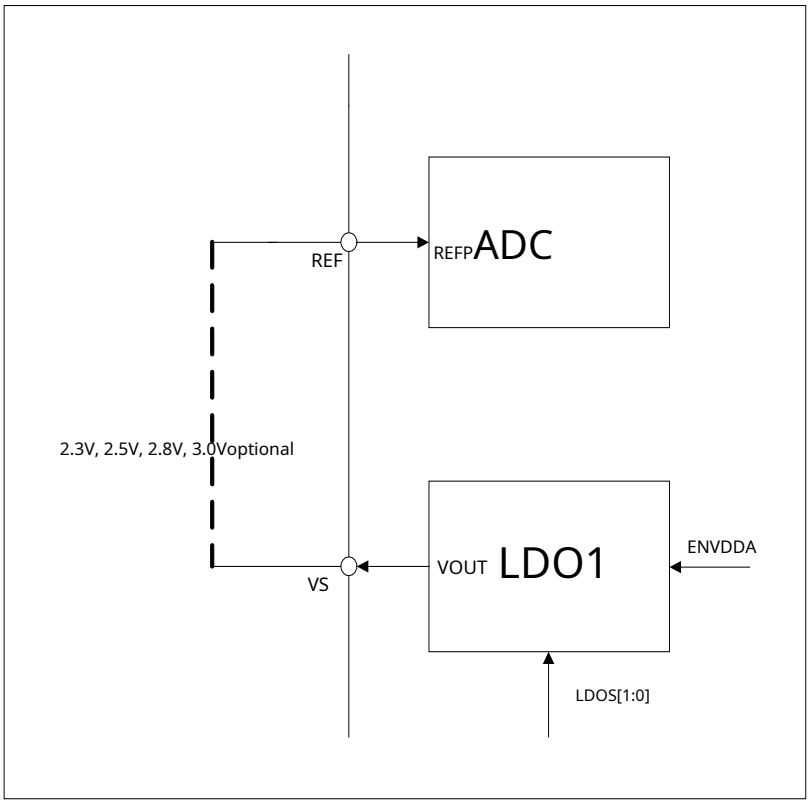
3.1Power Systems

3.1.1 Regulator

surface3-1Regulator circuit register list

land site	name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	power-on reset bit value
1Ch	NETE	LDOS[1:0]								00u0000u
1Dh	NETF			ENVDDA					ENVB	u0000000

3.1.1.1for modulatorLDO



picture3-1Analog part of the voltage regulator circuit

As shown3-1shown, used to generateVSas a sensor andADCthe reference voltage, by selectingLDOScan make the output2.3V, 2.5V,2.8V,3.0VOptional.ENVDDAasLDO1the enable signal.LDO1The control register flags areENVDDAand LDOS. The output voltage isVS.ENVBAs an enable signal for the entire analog power supply section, after shutdownADCandLCD change pump etc will not work.

NETEregister(address=1Ch)

characteristic	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0	U-0
NETE	LDOS[1:0]							
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit7~6 LDOS[1:0]:VSVoltage value selection

LDOS[1:0] = 00 VS=3.0V
LDOS[1:0] = 01 VS=2.8V
LDOS[1:0] = 10 VS=2.5V
LDOS[1:0] = 11 VS=2.3V

NETFregister(address=1Dh)

characteristic	U-0	U-0	R/W-0	U-0	U-0	U-0	U-0	R/W-0
NETF			ENVDDA					ENVB
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit5 ENVDDA:LDO1enable signal

ENVDDA=1:LDO1Enable
ENVDDA=0:LDO1Disable

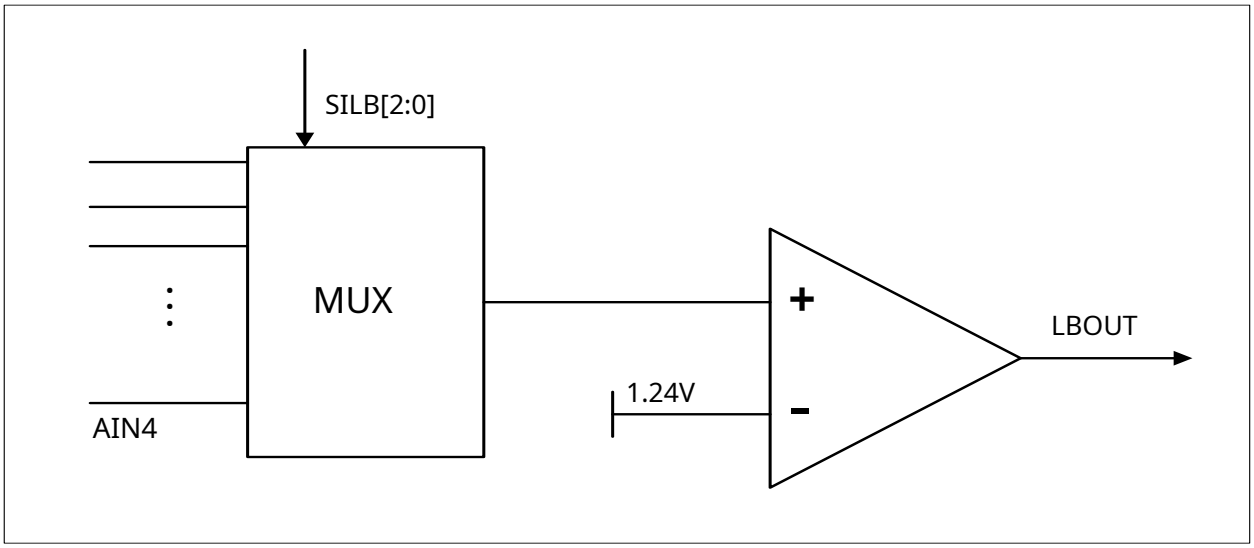
Bit0 ENVB: Analog power enable signal

ENVB=1: Analog power enable
ENVB=0: Analog power disabled

operate:

1. WillENVDDAset high
2. set upENVBset high
3.set upLDOS[1:0],chooseVSvalue.

3.1.2low voltage comparator



picture3-2Block diagram of low voltage comparison function block

Low voltage comparators are used forDVDDlow voltage detection.CSU8RP1001Integrating a can produce1/2DVDDand1/3DVDDthe voltage divider. The multiplexer is used to select different voltage dividers to connect to the input of the low voltage comparator. The output of the multiplexer and1.24V For comparison, its control register flag isSILB[2:0]andENLB, the output of the comparator isLBOUT,LBOUTis read-only. Please see picture3-2.

surface3-2Low Voltage Comparator Register List

land site	name	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	power-on reset bit value
1Ch	NETE				SILB[2:0]			ENLB		00u0000u
1Fh	SVD								LBOUT	uuuuuuu0

operate:

1. set register flagENLB, enabling the low-voltage comparator. 2.

The comparator output isLBOUT.

surface3-3Selection List of Low Voltage Comparator Detection Voltage

SILB[2:0]	Detection voltage	To meet the conditions	but
000	DVDD	DVDD >2.4V	LBOUT=1
001	DVDD	DVDD >2.5V	LBOUT=1
010	DVDD	DVDD >2.6V	LBOUT=1
011	DVDD	DVDD >2.7V	LBOUT=1
100	DVDD	DVDD >2.8V	LBOUT=1
101	DVDD	DVDD >3.6V	LBOUT=1
110	AIN4	AIN4 >1.24V	LBOUT=1
111	DVDD	DVDD >3.2V	LBOUT=1

3.2 HaltandSleepmodel

CSU8RP1001Support low voltage working mode. becauseCSU8RP1001in standby mode, allowingCPUstop working CSU8RP1001Perform stop or sleep mode to reduce power consumption. The two modes are described below: Stop Mode

CPUAfter executing the stop instruction, the program counter stops counting until an interrupt instruction occurs. To avoid returning from interrupt (Interrupt Return) caused by the program error, it is recommended to add one after the stop commandNOPinstructions to ensure that the program returns to normal operation. sleep mode

CPUAfter executing the sleep instruction, all oscillators stop working (EO_SLPfor0time) until reset by an external interrupt instruction CPU. To avoid returning from interrupt (Interrupt Return) caused by the program error, it is recommended to add one after the stop commandNOPinstructions to ensure the normal operation of the program. Power consumption in sleep mode is approximately1.5uA.

to ensure thatCPUThe power consumption in sleep mode is the smallest. Before executing the sleep command, it is necessary to turn off all power modules and analog circuits, and ensure that allI/Omouth is receivedVDDorDGNDlevel.

Before executing the sleep instruction, execute the following procedure.

```
CLRF    NETA        ; reset state
CLRF    NETC        ; reset state
CLRF    NETE        ; reset state
CLRF    NETF        ; reset state
CLRF    PT1PU       ;disconnectPT1Pull-up resistor
CLRF    AENB        ; set as analog port
MOVLW F0H
MOVWF PT1EN         ;PT1[7:4]use as output
MOVLW 01h
MOVWF PT2PU         ;disconnectPT2oral removalbit0(PT2[0]) pull-up resistors for other interfaces other than
MOVLW 0FEh
MOVWF PT2EN         ;removebit0(PT2[0])outside,PT2[7:0]used as an output
CLRF PT2            port;PT2output is low
CLRF PT3PU          ;disconnectPT3Pull-up resistor
CLRF INTF           ;Clear the interrupt flag
MOVLW 081h
MOVWF INTE          ;Enable external interrupt
SLEEP               ;MakeCSU8RP1001perform sleep mode
NOP                 ;ensureCPUAfter restarting the program works fine
```

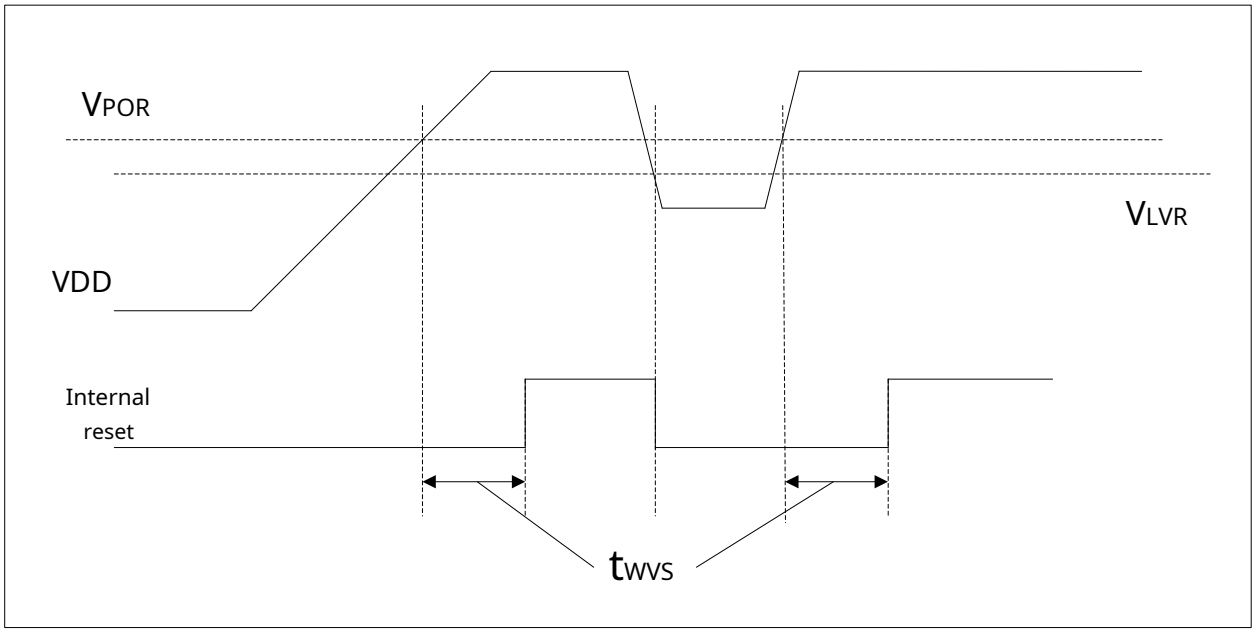
3.3reset system

CSU8RP1001Including the following reset methods:

- power-on reset
- low voltage reset
- External reset
- Watch dogreset

When power-on reset, low-voltage reset or external reset occurs, all system registers return to their default states, the program stops running, and the program counterPCclear. After the reset is complete, the system switches from the vector0000Hto restart operation.

whenwatch dogWhen a reset occurs, the system registers still keep their original values, but at this timePCPointer reset. After the reset is complete, the system switches from the vector0000Hto restart operation.



picture3-3Power-on reset circuit example and power-on process

parameter	minimum	Typical value	maximum value
VPOR		2.2V	
VLVR		2.0V	
tWVS	37.8ms	39ms	40.2ms

VPOR: Power-on reset

VLVR: Low voltage reset

tWVS: Wait for voltage stabilization time

NETDregister(address=1Bh)

characteristic	R/W-0	R/W-0	R/W-0	UX	UX	UX	UX	R/W-0
NETD		LVR_EN						

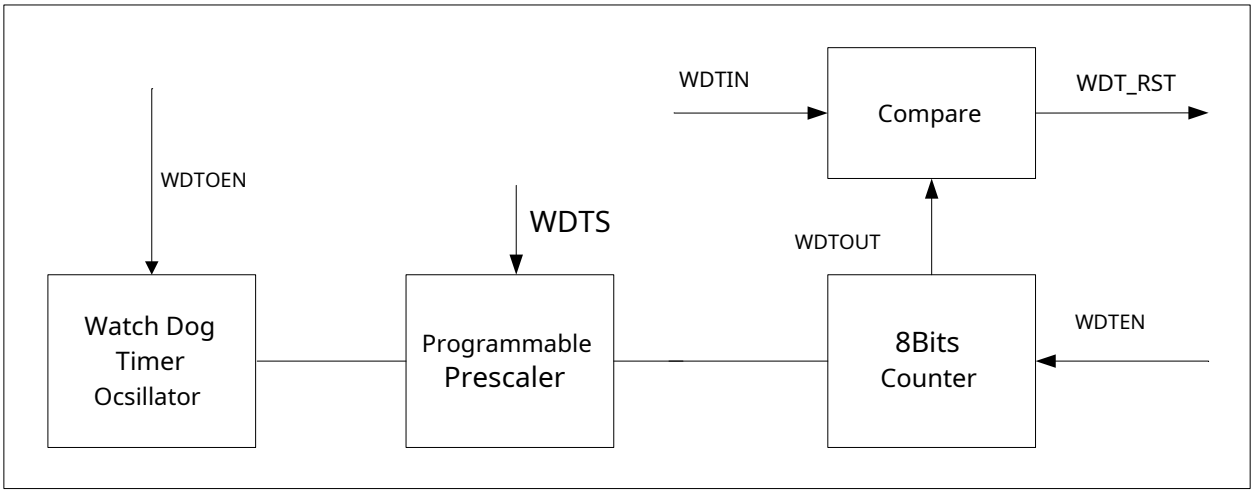
Bit 6 LVR_EN:Low voltage reset enable

0 =Low voltage reset enable on (recommended)

1 =Low Voltage Reset Enable Shutdown

Note: When the low voltage reset enable is turned off,sleepThe power consumption in mode is approximately200nA. But after turning off the low voltage reset, DVDDvoltage drops toVLVRThe following will not be reset.

3.4watchdog (WDT)



picture3-4Watchdog Timer Functional Block Diagram

watchdog timer (WDT) is used to prevent the program from getting out of control due to some uncertainty. whenWDTAt startup,WDT After the timer expires, theCPUreset. The program that is running is generally inWDTresetCPUreset beforeWDT. When certain failures occur, the program will beWDTReset to normal state, but the program will not resetWDT.

when user setWDTOEN, the internal watchdog timer oscillator (3KHz) will start, register flag bit WDTS[2:0]Controls the clock divider of the watchdog counter, setWDTENEnable counter. When the count value is equal toWDT_INoverflows when the values are equal, when it overflows it sendsWDTOUTsignal resetCPU(The program counter will jump to00hto reset the program) and setTO flag bit. User can use commandCLRWDTresetWDT.

surface3-4Watchdog Timer Register Table

address	name	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	power-on reset value
04H	STATUS					TO				00u00000
0DH	WDTCON	WDTOEN	WDTEN		Wdt_lcd		WDTS[2:0]			00u0u000
38h	WDTIN	WDT_IN[7:0]								11111111

operate:

1. set upWDTS[3:0],chooseWDTClock frequency. 2. set upWDT_IN, choose a different overflow time value 2. Set register flags:WDTEN,EnableWDT. 3. Position WDTOEN,OpenWDTcrystal oscillator. 4. execute in the programCLRWDTcommand resetWDT.

whenwdt_lcdAfter the flag position is high,LCDwill usewdtThe divided clock asLCD_CLK, please refer to the detailed configuration parameters 3.6.2LCDFrame rate selection

WDTS[2:0]	watchdog overflow timeTwdt(whenWDT_IN==FFH)
000	21.8s
001	10.9s
010	5.5s
011	2.7s
100	1.4s
101	0.68s
110	0.34s
111	0.17s

overflow timeT=Twdt*WDT_IN/255.

3.5 ADCsmodule

CSU8RP1001contains atwenty fourbitsigma deltatype of analog-to-digital converter (ADC).ADCThe reference is determined by the internal reference voltageVSProvided, but the reference voltage can also be provided externally (the internal reference voltage needs to be turned off at this time, and the external reference voltage needs to be connected REFP), both analog differential input channels can be used independently, but only one channel can be used per conversion. It also provides internal temperature detection and internal short test functions.

First put the internal powerENVBopen, then openVS(ENVDDA), then configureADCvarious configuration parameters, turn on the global interrupt enable andADCinterrupt enable, then theADCENOpen, wait for the analog part to be established, open the digital filter (ADFEN), at the beginning of work, the digital filter needs a settling time. For the second-order digital filter, it needs two data conversion cycles. For the third-order digital filter, it needs three data conversion cycles. .

3.5.1 ADCsRegister Description

surface3-5 ADCsFunction module related register list

land site	name	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	power-on reset bit value
06h	INTF						ADIF			uuu0u000
07h	INTE	GIE					ADIE			0uu0u000
10h	ADOH	ADO [23:16]								00000000
11h	ADOL	ADO[15:8]								00000000
12h	ADOLL	ADO[7:0]								00000000
13h	ADCON			ADF_EN	COMBS		ADM[2:0]			uu00u000
18h	NETA	SINL[1:0]			EN_IA	EH_CHS[1:0]		PGA_C[1:0]		00u00000
1Ah	NETC	CHS_IA	CHS_MOD	PGA_F[3:0]				ADEN		0000000u
1Dh	NETF				BGIDA[1:0]		BGID[1:0]			u0000000
59h	COMP			TCOM						uu0uuuuu

ADOHregister (address is10h)

characteristic	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

ADOLregister (address is11h)

characteristic	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

ADOLLregister (address is12h)

characteristic	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]:ADCdigital output

ADO[23] = ADCDigital output sign bit.0 =output is positive;1 =The output is negative.

ADO[22] = ADCDigital output databit 22

~

~

~

ADO[0] = ADCDigital output databit 0

characteristic(Property):

R =readable bit W =writable bit U =invalid bit

-n=Value after power-on reset '1' =bit is set '0'=bit is cleared X =indeterminate bit

COMPregister (address is59h)

characteristic	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
COMP			TCOM					
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 5 TCOM: Gain temperature compensation

0 =Positive Gain Temperature Compensation

1 =Negative Gain Temperature Compensation

ADCONregister (address is13h)

characteristic	U-0	U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0
ADCON			ADFEN	COMBS		ADM[2:0]		
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 5 ADFEN:ADCDigital filter enable signal

0 = ADCdigital filter disabled 1= ADCdigital filter enable

Bit4 COMBS: Digital filter order selection

0 =Select second-order digital filter

1 =Choose a third-order digital filter

Bit2-0 ADM[2:0]: Digital filter downsampling selection register

See the table for selection3-6

surface3-6 ADCsOutput rate selection list

ADM[2:0]	ADCoutput rate
000	ADCF/64
001	ADCF/128
010	ADCF/256
011	ADCF/512
100	ADCF/1024
101	ADCF/2048
110	ADCF/4096
111	ADCF/8192

ADCThe final output rate is given byPGA_F[3:0]andADM[2:0]come together to control whichPGA_Fselect controlADC sampling rate ADCF(see table3-9),ADMselect controlADCFoutput rate (see table3-6)

NETAregister (address is18h)

characteristic	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
NETA	SINL[1:0]			EN_IA	EN_CHS[1:0]		PGA_C[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 SINL[1:0]:ADCSignal selector at the input

00 = ADCinput is connected toAIN0 (PT1[0])andAIN1(PT1[1]) 01 = ADC input is connected toAIN0 (PT1[2])andAIN1(PT1[3]) 10 = ADCinput is connected toTEMP 11 = ADCThe input terminal is internally shorted

inTEMPThe input of the integrated temperature sensor on-chip.

Bit4 EN_IA:PGAenable register

0 = PGAnot enabled,PGAGain is1.

1 = PGAEnable,PGAWhen enabled,PGAGain is17. It is recommended to enablePGA

Bit3-2 EN_CHS: chopper enable signal

00 =Chopping off

01 =Chopping off

10 =Chopping off

11 =Chopping on

It is recommended to select chopper on

Bit1-0 PGA_C: Modulator gain option (ifPGAturn on, i.e.EN_IA=1, then in17timesPGAmultiplied by the gain).

00 = 16
01 = 8
10 = 4
11 = 1

Recommended choice4

NETCregister (address is1Ah)

characteristic	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
NETC	CHS_IA	CHS_MOD	PGA_F[3:0]				ADEN	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7 CHS_IA: reserved bit, please keep it as1

Bit 6 CHS_MOD: Modulator chopper mode selection register

0 =chopper mode1

1 =chopper mode2

Chopping mode is recommended2

Bit 5-2 PGA_F[3:0]: Sampling clock frequency selection register

selection see table3-9

Bit 0 ADEN:ADCenable flag

1 = ADCEnable

0 = ADCDisable

NETFregister (address is1Dh)

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

Bit 4-3 BGIDA[1:0]:PGABias current 00 = 1/2

times the bias current

01 = 1times the bias current

10 = 1.25times the bias current

11 = 1.75times the bias current

It is recommended to use double the bias current, and a large bias current is beneficial to the stability of the modulator.

Bit 2-1 BGID[1:0]: Modulator bias current

00 = 1/4Bias current

01 = 1/2Bias current

10 = 3/4Bias current

11 = 1times the bias current

It is recommended to use1times the bias current

3.5.2 ADCsgain

surface3-7 PGAGain Selection List

EN_IA	I_GAIN
0	1
1	17

surface3-8Capacitor Ratio Gain Selection List

PGA_C[1:0]	C_GAIN
00	16
01	8
10	4
11	1

surface3-9Sampling frequency selection list

PGA_F[3:0]	ADCF
0000	4M
0101	2M
1001	1M
1100	500K
1110	250K
1111	4M

ADCThe gain of the analog input consists of two parts,PGAgainI_GAIN(Depend onEN_IAcontrol), modulator gain C_GAIN(Depend onPGA_Ccontrol), the overall gain of the analog input $GAIN=I\_GAIN\times C\_GAIN$

Tips:Recommended configuration of solar body scale, and description:

address	name	configuration value	illustrate
13h	ADCON	33h (00110011B)	3order digital filtering, reducing512sampling
18h	NETA	1Eh(00011110)	AIN0/AIN1input, turn on chopping,Gain=17*4=68
1Ah	NETC	C2h(11000010)	Configure the chopper method and sampling frequency (4M), the final output rate is4M/512
1Ch	NETE	(110XXXX0)	VS=2.3V(inXBit indicates that the user can set it by himself)
1Dh	NETF	2Fh(00101111)	openVS,1Double the bias current, open the reference source

Comprehensive configuration:PGA=17*4=68,DataRate=4M/512=7.8KHz,1times the bias current

3.6 LCD Driver

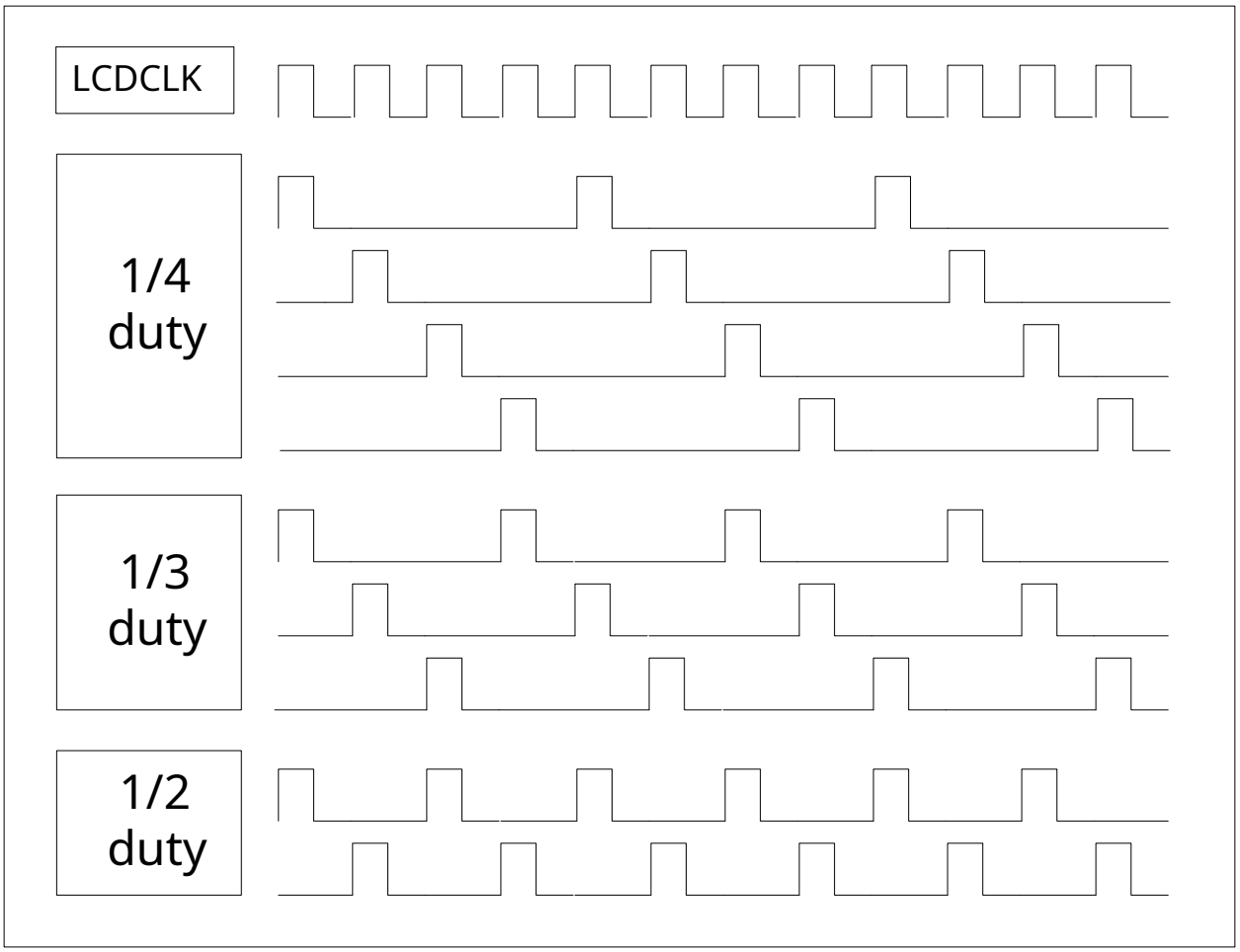
LCDdrive has14indivualsegmentoutput (SEG1-SEG14)and4indivualcommonoutput (COM1-COM4)

3.6.1 LCDcontrol mode

LCDdrive has3control modes:1/2duty,1/3dutyand1/4duty, set the register flagLCD_DUTY[1:0]Choose a mode.

surface3-10 LCDofdutypick list

LCD_DUTY[1:0]	control mode Mode	SEG1-16							
		Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
01	1/2duty					-	-	COM2	COM1
10	1/3duty					-	COM3	COM2	COM1
11	1/4duty					COM4	COM3	COM2	COM1



picture3-5 LCDofdutyMode duty cycle

3.6.2 LCDFrame rate selection

LCDThe frame frequency is determined by the registerLCDCKS[1:0]ok, yesLCDThe input clock to the module is divided to obtainLCDCK.

surface3-11 LCDSCKpick list

M4_CK	M0_CK	M5_CK	clock source (KHz)		LCDSCK(KHz)	
X	0	0	ICK	4000	ICK/128	32
0	1	0	ECK	32	ECK	32
1	1	0	ECK	4000	ECK/512	8
0	X	1	ECK	32	ECK	32
1	X	1	ECK	4000	ECK	4000

surface3-12 LCDCLKpick list

Wdt_lcd	LCDSCKS				LCDCLK(KHz)	
0	0	0	0	0	LCDSCK/32	1
0	0	0	0	1	LCDSCK/30	1.067
0	0	0	1	0	LCDSCK/28	1.143
0	0	0	1	1	LCDSCK/26	1.231
0	0	1	0	0	LCDSCK/24	1.333
0	0	1	0	1	LCDSCK/22	1.455
0	0	1	1	0	LCDSCK/20	1.6
0	0	1	1	1	LCDSCK/18	1.778
0	1	0	0	0	LCDSCK/16	2
0	1	0	0	1	LCDSCK/14	2.286
0	1	0	1	0	LCDSCK/12	2.667
0	1	0	1	1	LCDSCK/10	3.2
0	1	1	0	0	LCDSCK/8	4
0	1	1	0	1	LCDSCK/6	5.333
0	1	1	1	0	LCDSCK/4	8
0	1	1	1	1	LCDSCK/2	16
1	0	0	0	0	WTDCLK/32	0.094
1	0	0	0	1	WTDCLK/30	0.1
1	0	0	1	0	WTDCLK/28	0.107
1	0	0	1	1	WTDCLK/26	0.115
1	0	1	0	0	WTDCLK/24	0.125
1	0	1	0	1	WTDCLK/22	0.136
1	0	1	1	0	WTDCLK/20	0.15
1	0	1	1	1	WTDCLK/18	0.167
1	1	0	0	0	WTDCLK/16	0.188
1	1	0	0	1	WTDCLK/14	0.214
1	1	0	1	0	WTDCLK/12	0.25
1	1	0	1	1	WTDCLK/10	0.3
1	1	1	0	0	WTDCLK/8	0.375
1	1	1	0	1	WTDCLK/6	0.5
1	1	1	1	0	WTDCLK/4	0.75
1	1	1	1	1	WTDCLK/2	1.5

surface3-13 LCDFrame rate selection list

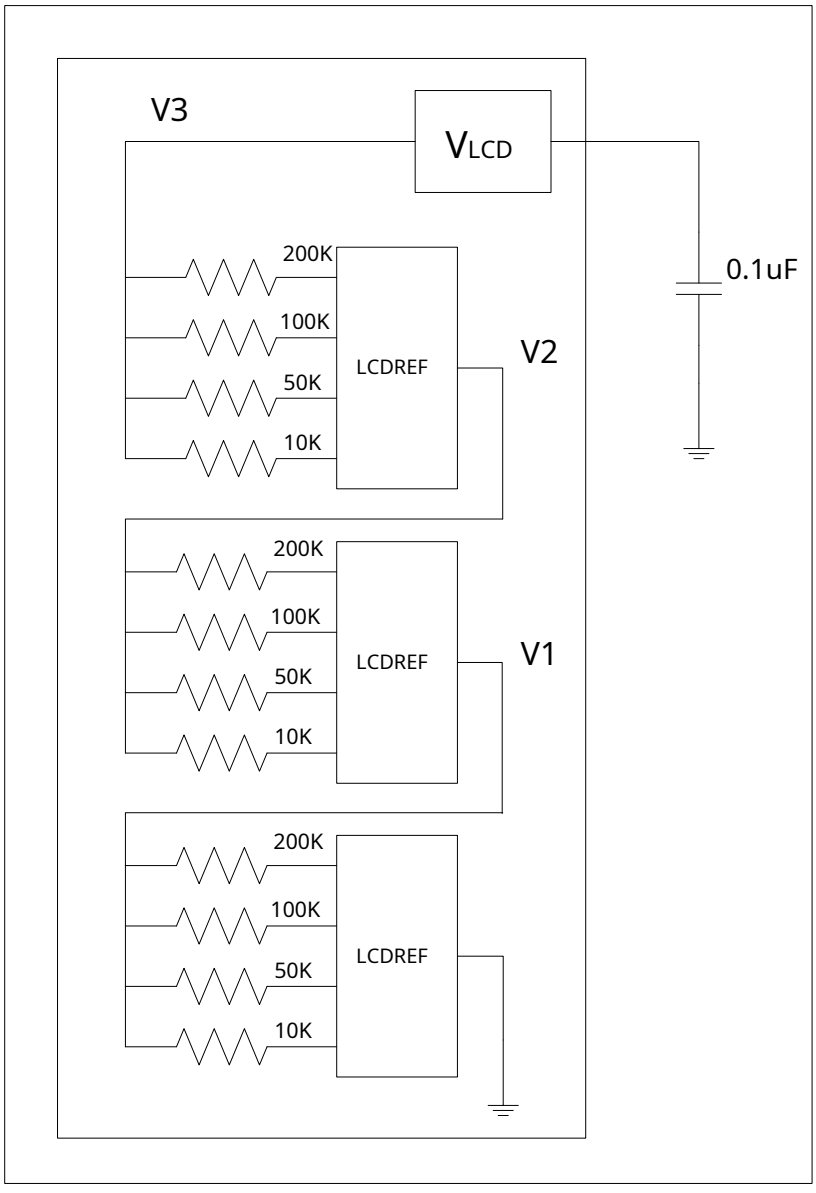
LCDCKS[1:0]	LCDframe rate (LCDCK)
00	LCDinput clock frequency/4
01	LCDinput clock frequency/8
10	LCDinput clock frequency/16
11	LCDinput clock frequency/32

3.6.3 LCDBias voltage

LCDdrive has3a bias voltage,V1,V2andV3,Have2power modes:1/3bias,1/2bias. There are two options for the generation circuit of the bias voltage, byDIVSRegister to choose, one is to use internal resistance to divide the voltage, the advantage is that it can saveV2andV1 pinof external capacitors, the disadvantage is that the turn-onLCDThere is static power consumption on the time-division voltage circuit. The size of the static power consumption is related to the resistance of the voltage-dividing resistor. The smaller the voltage-dividing resistor, the greater the static power consumption. The driving ability of the resistor-dividing method is weaker. The weaker the ability, the other uses the capacitor voltage divider. The advantage is that the driving ability is strong and there is no static power consumption. The disadvantage is that it needs to increaseV2andV1capacitance at.

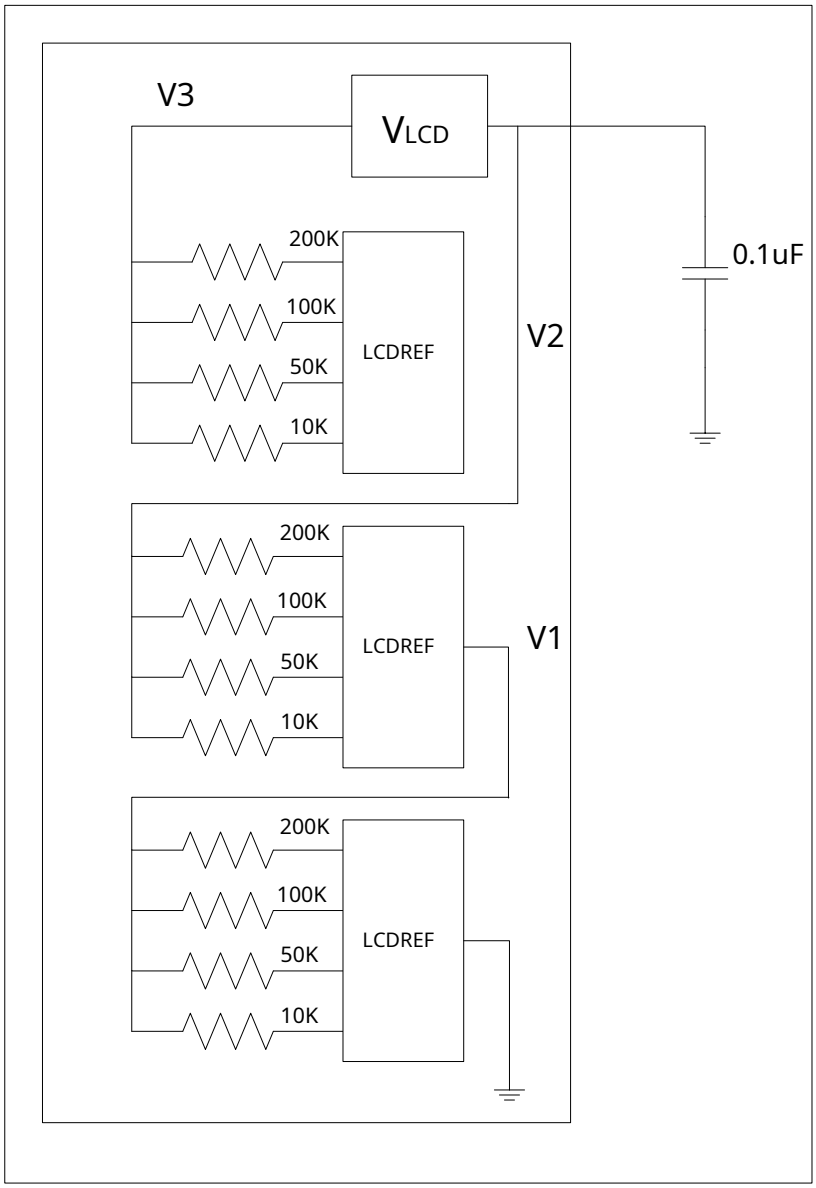
The bias voltage is generated by means of resistor divider

- 1/3biasPower Systems



picture3-6 LCDof1/3biasPower system circuit connection diagram (resistor voltage divider)

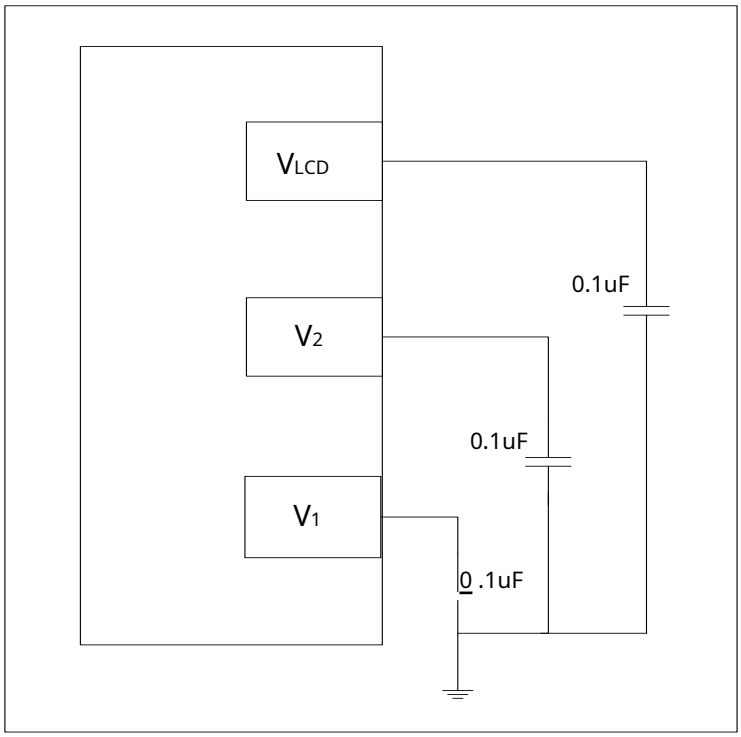
-1/2biasPower Systems



picture3-7 LCDof1/2biasPower system circuit connection diagram (resistor voltage divider)

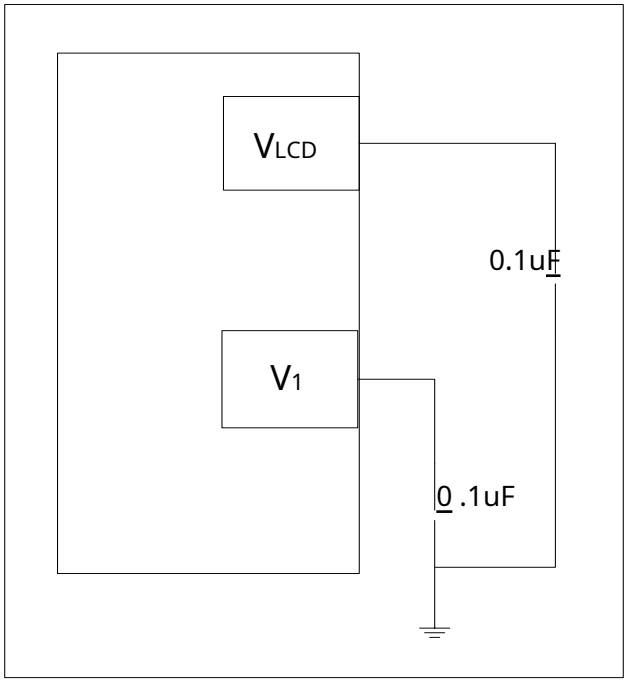
The bias voltage is generated by the way of capacitive voltage divider

- 1/3biasPower Systems



picture3-8 LCDof1/3biasPower system circuit connection diagram (capacitor voltage divider)

-1/2biasPower Systems



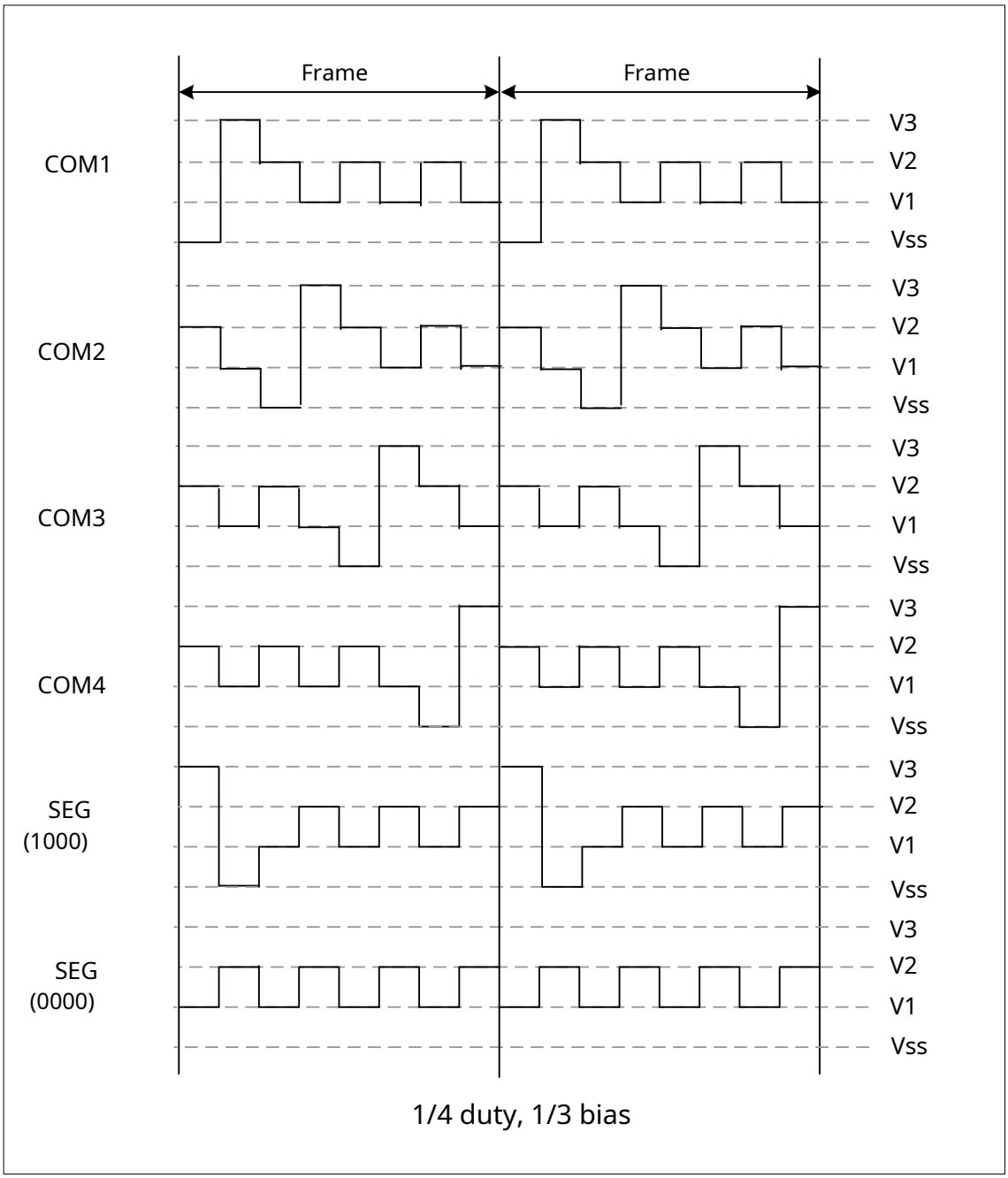
picture3-9 LCDof1/2biasPower system circuit connection diagram (capacitor voltage divider)

LCDThe power supply can be provided by an internal charge pump or an external power supply. In this case, the internal charge pump needs to be turned off, and the external power supply must be connected to the power supply.VLCD. When using the internal charge pump, the internal reference source needs to be turned on first (byENVB control).

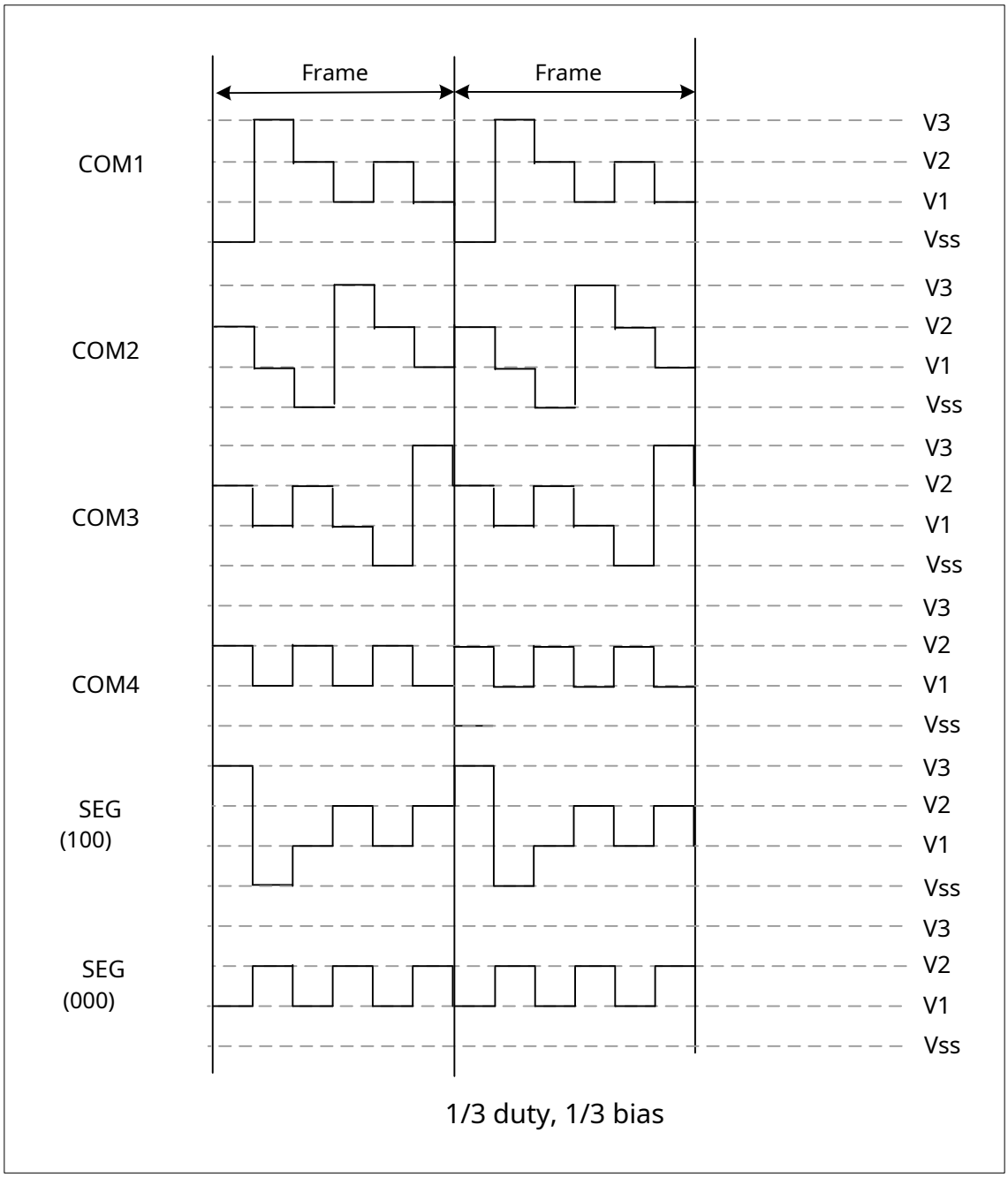
When using capacitive voltage divider, increasing the capacitance value can increase thecomandsegThe drive capability of the port.

3.6.4 LCDdrive waveform

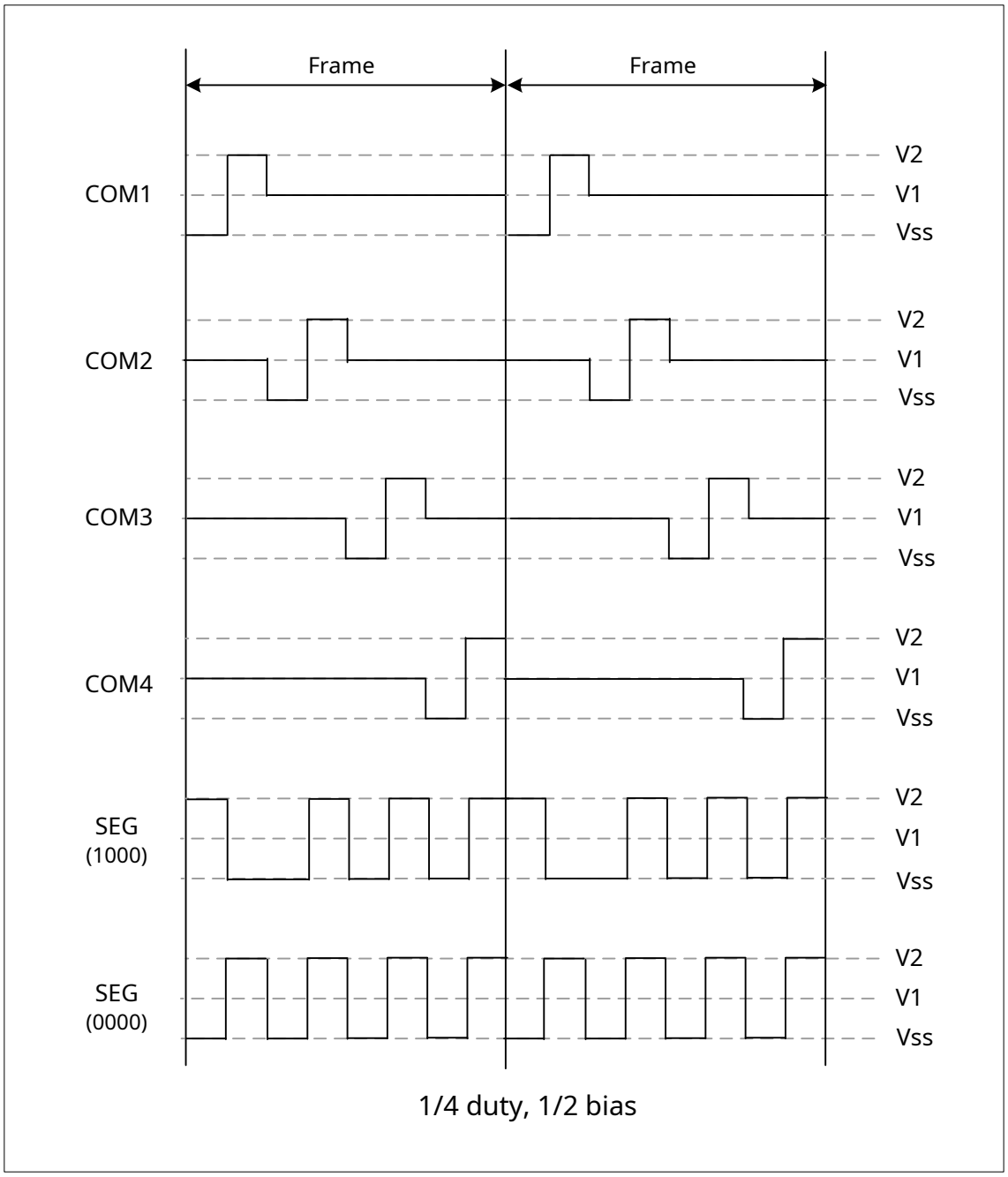
LCDThe driving waveform is divided intoAandBTwo waveforms, via registerLCDWSto choose, whichBWaveforms work better for larger displays.



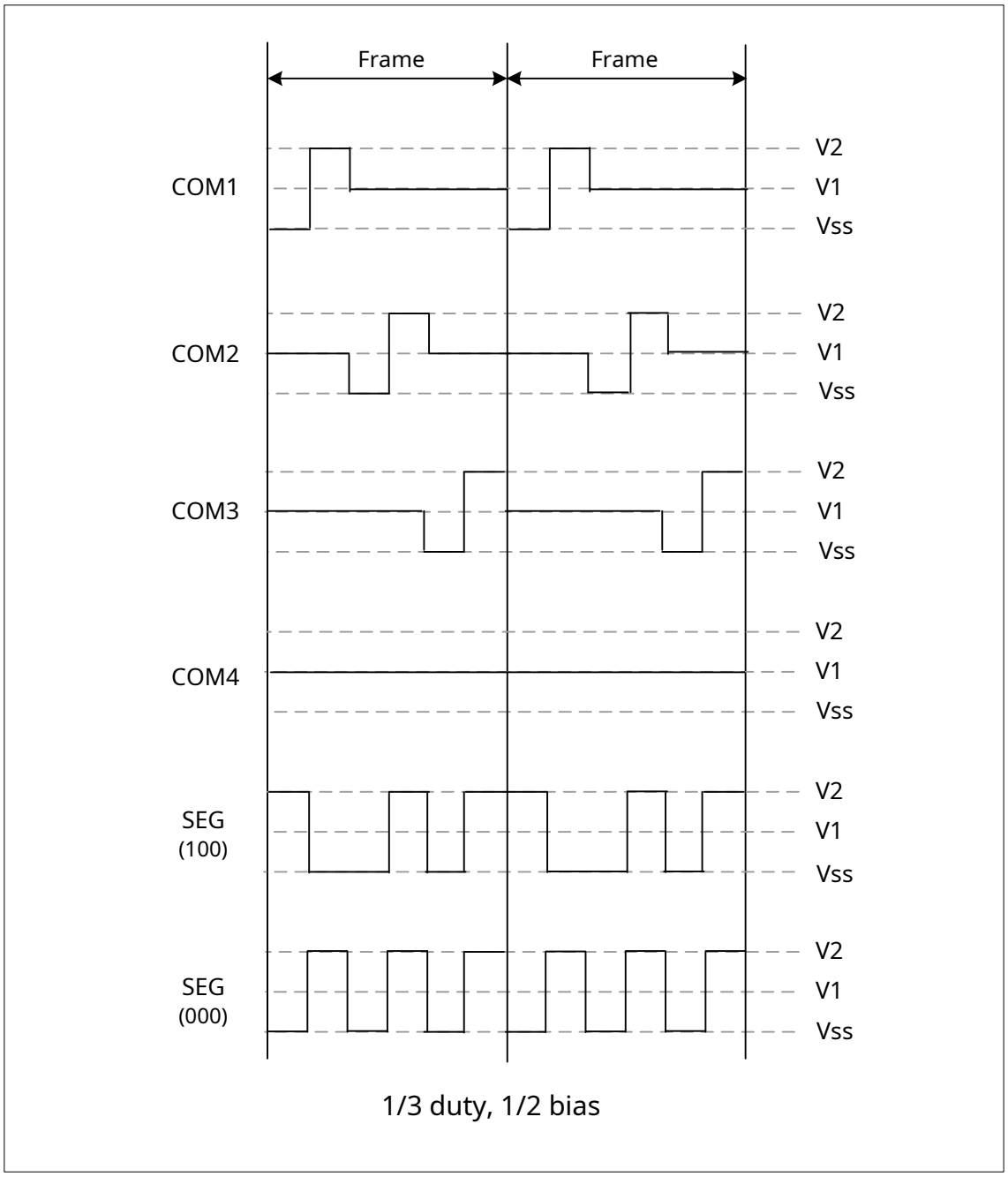
picture3-10 LCDof1/4duty 1/3biasThe clock of the power system (Awaveform)



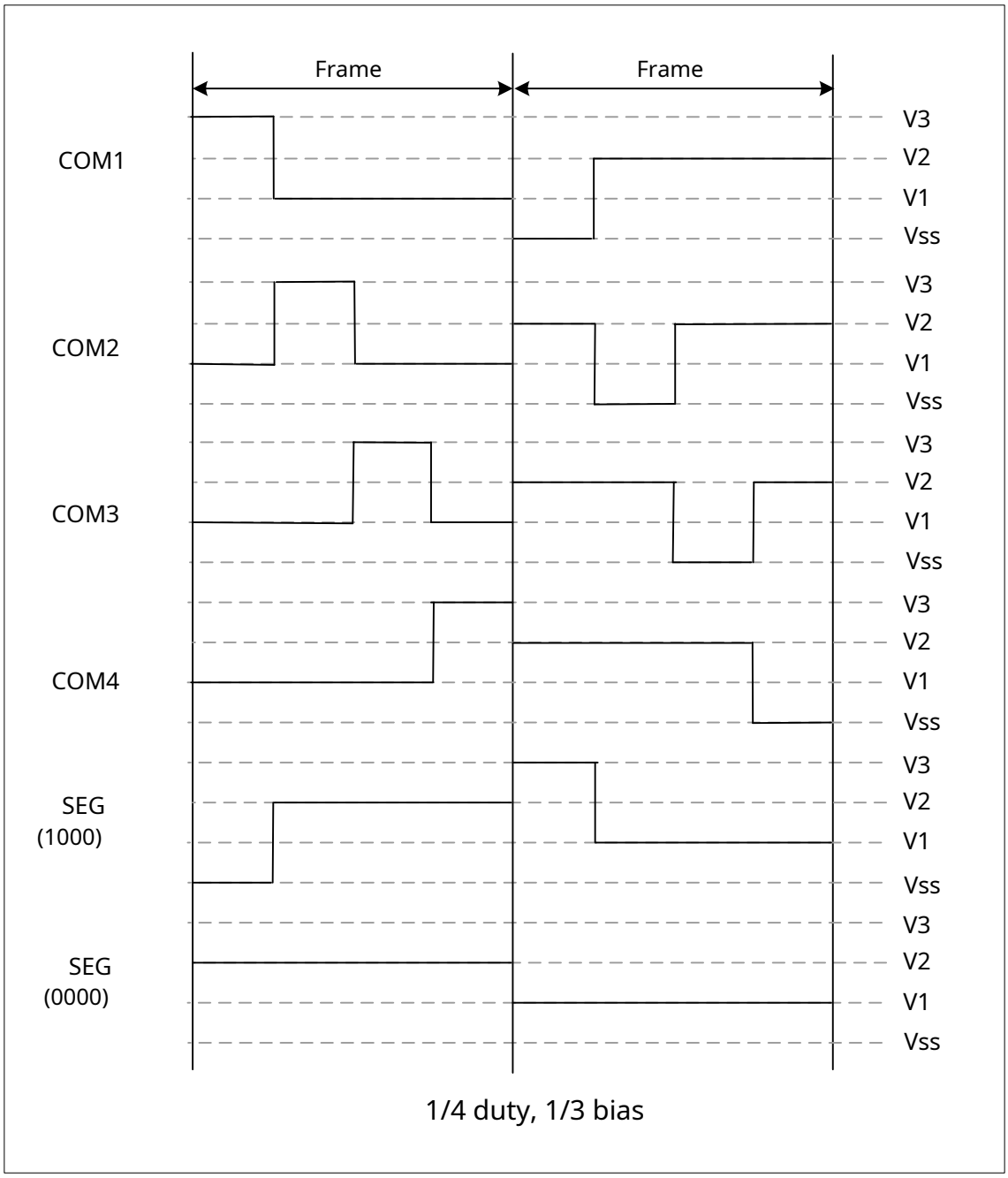
picture3-11 LCDof1/3duty 1/3biasThe clock of the power system (Awaveform)



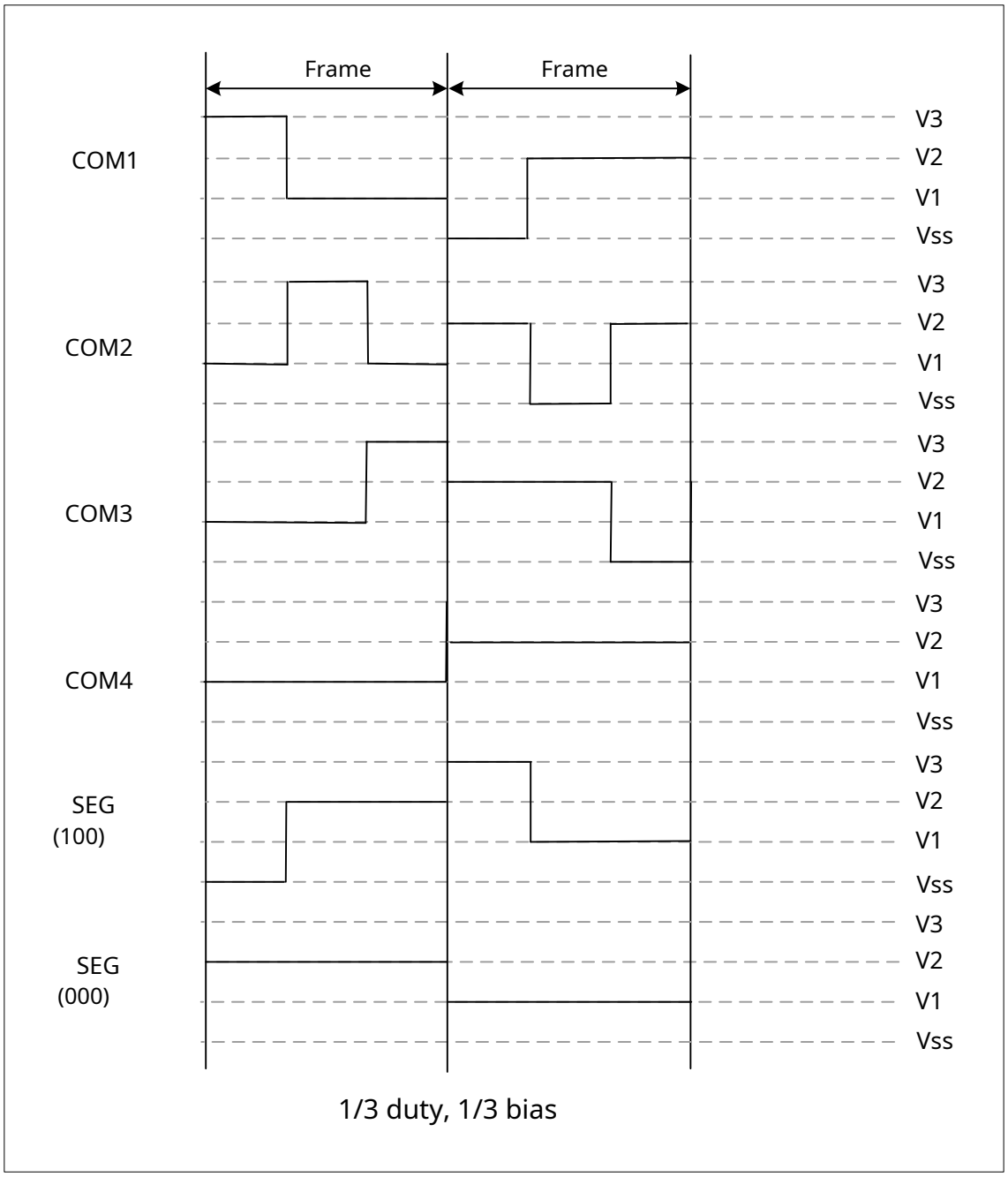
picture3-12 LCDof1/4duty 1/2biasThe clock of the power system (Awaveform)



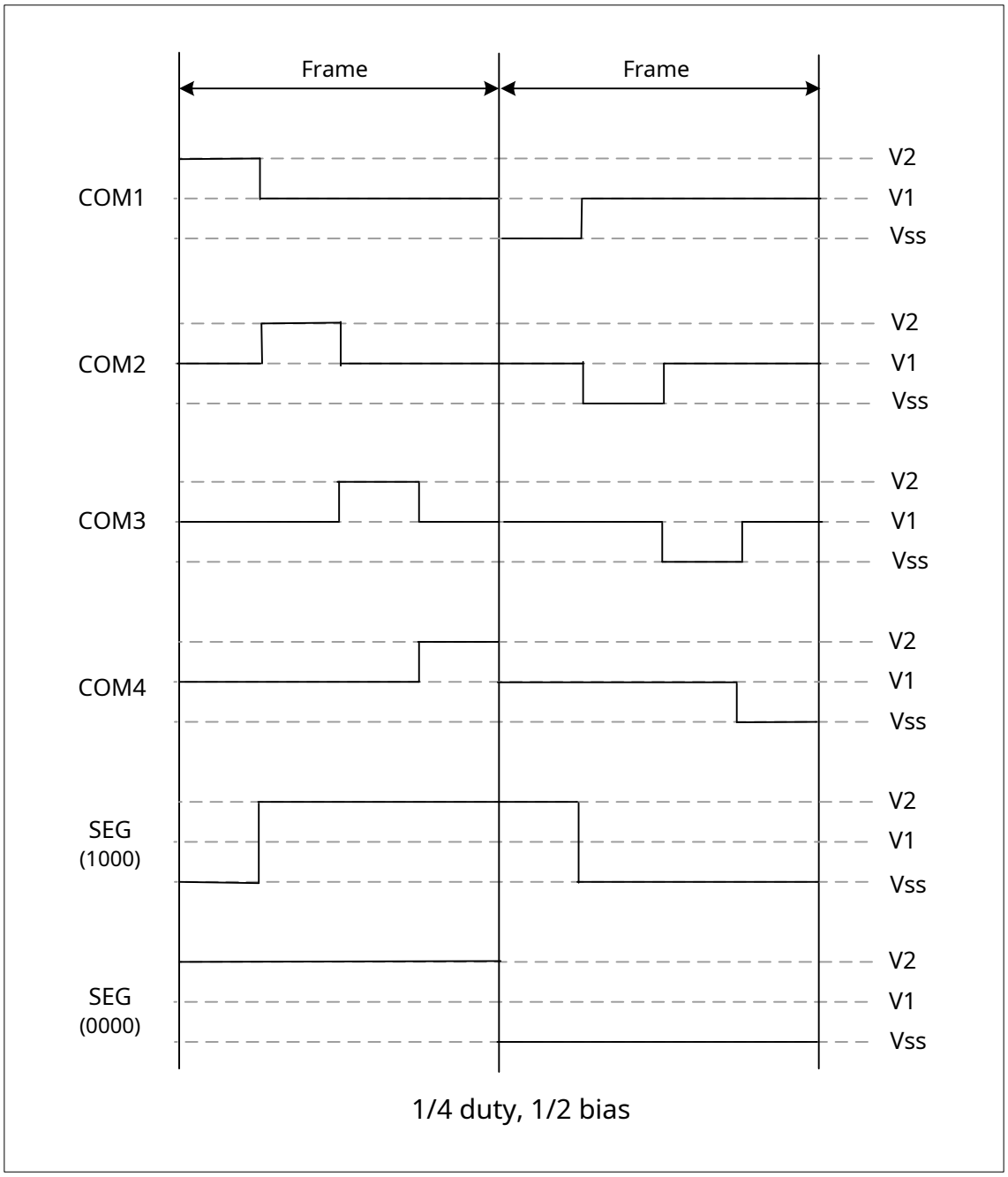
picture3-13 LCDof1/3duty 1/2biasThe clock of the power system (Awaveform)



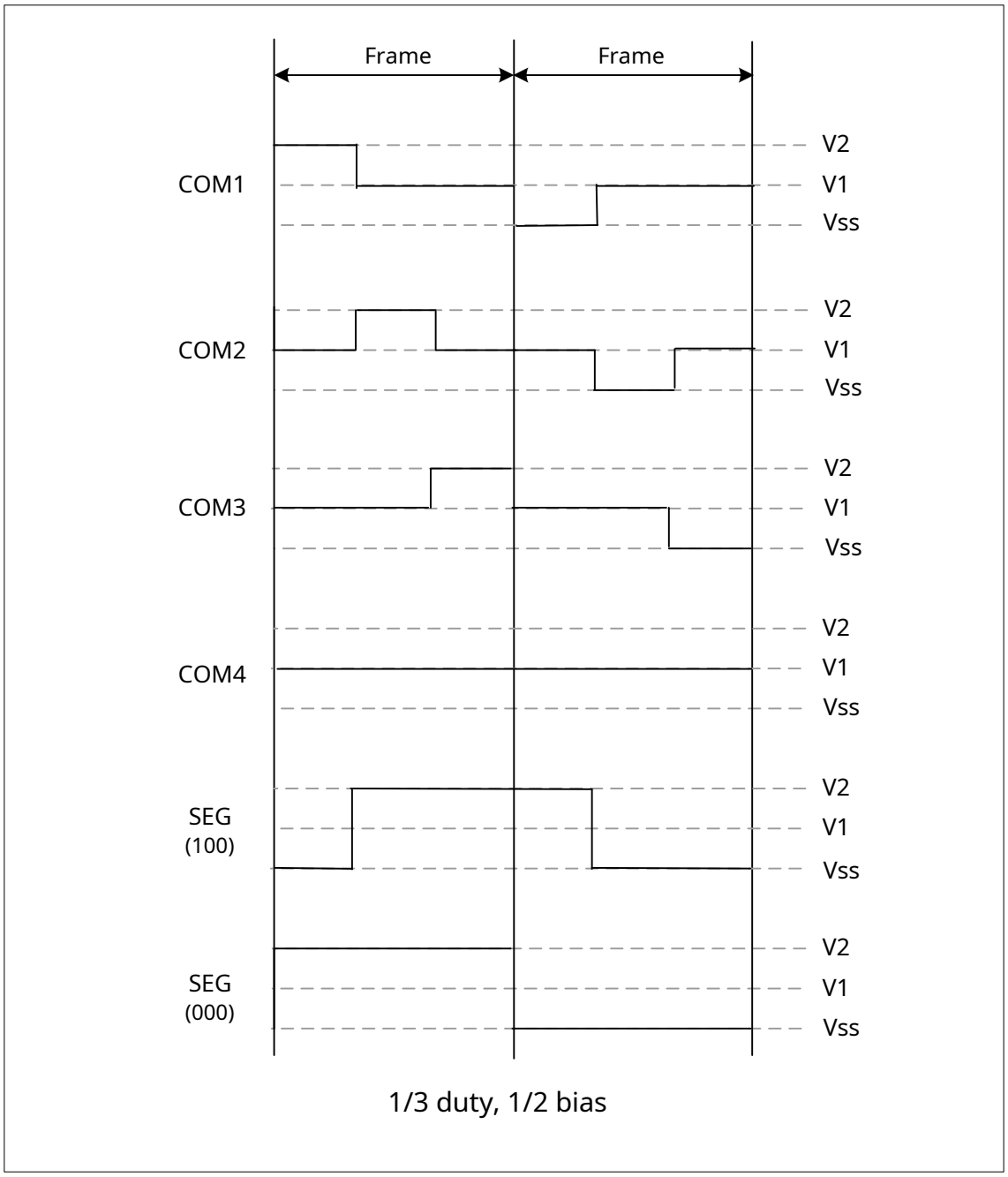
picture3-14 LCDof1/4duty 1/3biasThe clock of the power system (Bwaveform)



picture3-15 LCDof1/3duty 1/3biasThe clock of the power system (Bwaveform)



picture3-16 LCDof1/4duty 1/2biasThe clock of the power system (Bwaveform)



picture3-17 LCDof1/3duty 1/2biasThe clock of the power system (Bwaveform)

3.6.5 LCDRegister Description

surface3-14 CSU8RP1001ofLCDDriver Register List

land site	name	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	power-on reset bit value
15h	PCK					LCDSCKS[1:0]				u00000uu
1Bh	NETD		DIVS	LCDCH	LEVEL_S	VLCDX[1:0]		LCDREF[1:0]		00000000
40h	LCD1					SEG1[3:0]				uuuu0000
41h	LCD2					SEG2[3:0]				uuuu0000
42h	LCD3					SEG3[3:0]				uuuu0000
43h	LCD4					SEG4[3:0]				uuuu0000
44h	LCD5					SEG5[3:0]				uuuu0000
45h	LCD6					SEG6[3:0]				uuuu0000
46h	LCD7					SEG7[3:0]				uuuu0000
47h	LCD8					SEG8[3:0]				uuuu0000
48h	LCD9					SEG9[3:0]				uuuu0000
49h	LCD10					SEG10[3:0]				uuuu0000
4Ah	LCD11					SEG11[3:0]				uuuu0000
4Bh	LCD12					SEG12[3:0]				uuuu0000
4Ch	LCD13					SEG13[3:0]				uuuu0000
4Dh	LCD14					SEG14[3:0]				uuuu0000
58h	LCDENR	LCDCKS[1:0]		LCDEN	LCDWS	LEVEL	LCD_DUTY[1:0]		ENPMPL	00000110

NETDregister (address is1Bh)

characteristic	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
NETD		DIVS	LCDCH	LEVEL_S	VLCDX[1:0]		LCDREF[1:0]	
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 6 DIVS: Bias voltage generation circuit selection:

0 =Select charge pump divider

1 =Select resistor divider

Bit 5 LCDCH:LCDport select signal 1 = SEG port is used as a digital output 0 = SEG mouth asLCDofSEGoutput

Bit 4 LEVEL_S:SegThe output signal is inverted 1 = Segfor1The output signal is low level

Segfor0The output signal is high 0 = Segfor1The output signal is high

Segfor0The output signal is low level

Bit3-2 VLCDX:VLCDOutput voltage selection

00 = 2.6V

01 = 2.8V

10 = 3.0V

11 = 3.2V

Bit1-0 LCDREF:LCDVoltage divider resistor selection 00 = 200Kohm

01 = 100Kohm

10 = 50Kohm

11 = 10Kohm

LCD1register (address is40h)

characteristic	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
LCD1					SEG1[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 3-0 SEG1[3]:LCDDrive control signal:SEG1bringCOM4.
SEG1[2]:LCDDrive control signal:SEG1bringCOM3.
SEG1[1]:LCDDrive control signal:SEG1bringCOM2.
SEG1[0]:LCDDrive control signal:SEG1bringCOM1.

LCD2register (address is41h) ~

LCD3register (address is42h) ~

LCD4register (address is43h) ~

LCD5register (address is44h) ~

LCD14register (address is4Dh)

characteristic	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
LCD14					SEG14[3:0]			
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 3-0 SEG14[3]:LCDDrive control signal:SEG14bringCOM4.
SEG14[2]:LCDDrive control signal:SEG14bringCOM3.
SEG14[1]:LCDDrive control signal:SEG14bringCOM2.
SEG14[0]:LCDDrive control signal:SEG14bringCOM1.

LCDENRregister(58h)

characteristic	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
LCDENR	LCDCKS[1:0]		LCDEN	LCDWS	LEVEL	LCD_DUTY[1:0]		ENPMPL
	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Bit 7-6 LCDCKS[1:0]:LCDFrame rate selector
11 = LCDThe frame rate isLCDinput clock frequency1/32
10 = LCDThe frame rate isLCDinput clock frequency1/16
01 = LCDThe frame rate isLCDinput clock frequency1/8
00 = LCDThe frame rate isLCDinput clock frequency1/4

Bit 5 LCDEN:LCDdrive enable flag
1 = LCDDrive enabled.LCDthe clock is started 0 = LCDThe drive is not enabled.LCDthe clock is stopped LCDWS:LCD

Bit 4 Waveform selection 1 =waveformB

0 =waveformA

Bit 3 LEVEL:LCDBIAS VOLTAGE SELECTOR FOR DRIVE 0 = LCDThe bias voltage of the driver is1/3bias 1 = LCDThe bias voltage of the driver is1/2bias
LCD_DUTY[1:0]:LCDDrive Control Mode (SEG dutycycle) 11 = LCDThe

Bit 2-1 drive control mode is1/4dutyPeriodic mode 10 = LCDThe drive control mode is1/3dutyPeriodic mode 01 = LCDThe drive control mode is 1/2dutyPeriodic mode ENPMPL:LCD charge pumpenable flag 1 = LCD charge pumpOpen

Bit 0

0 = LCD charge pumpclosure

3.6.6 LCDoperation steps

1. Connect the segment interface toLCDpanel.

2. set register flagsLEVELchooseLCDdrive power system. (0 = 1/3bias,1 = 1/2bias) 3. set up ENPMPLEnableLCDcharge pump. (need to open firstENVB)

4. set upM0_CK,M4_CK,chooseLCDThe frequency of the input clock. (seeLCDframe rate selection) 5. set register flagsLCDCKS[1:0]chooseLCDclock frequency (seeLCDframe rate selection) 6. set register flagsLCD_DUTY[1:0]to select a control mode. (SEG dutycycle)

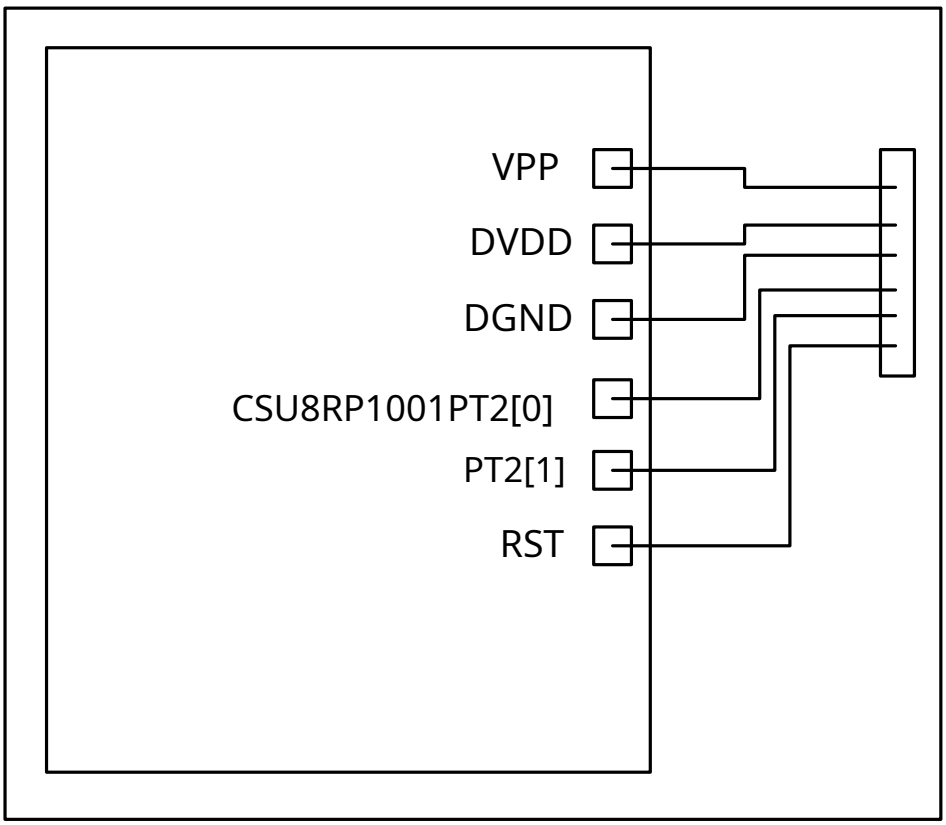
surface3-15 LCDofdutyControl mode selection list

LCD_DUTY[1:0]	control mode
00	- -
01	1/2
10	1/3
11	1/4

7. PositionLCDENto enableLCDdriver

3.7 OTPBurn interface

OTPThe interface of the programmer:



picture3-18OTPWriter interface diagram

surface3-16 OTPInterface Description

port name	Type	illustrate	Remark
VPP	enter	programming voltage. voltage range:6.25Varrive6.75V	
VDD	enter	Positive terminal of power supply	
VSS	enter	Negative end of power supply	
PT2[0]	input Output	PT2[0]port, data signal[1]	
PT2[1]	enter	PT2[1]port, clock signal[2]	
Reset	enter		

3.8 OTPonline burning

Circuit requirements: During online programming,VPP (OTPprogramming voltage pin) pin connection6.5V(voltage range:6.25V~6.75V) of the programming voltage.

Clock requirements: must use an internal crystal or4MHzand above external crystal oscillator.

surface3-17Online programming register list

land site	name	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bits0	power-on reset value
05h	WORK	working register								00000000
0Ah	EADRH					PARH[3:0]				00000000
0Bh	EADRL	PARL[7:0]								00000000
0Ch	EDATH	EDATH[7:0]								00000000
19h	NETB					ERV				00000000

EADRH:supplyOTPBurn online or read onlineOTPThe upper four bits of the address.

EADRL:supplyOTPBurn online or read onlineOTPThe lower eight bits of the address.

EDATH: Required when programmingFFh.

Work: supplyOTPBurning data during online burning or online readingOTPread data ERV :whenVPPpin voltage reaches4.5Vabove,ERVset high. Operation method:

online burningOTPTime

1.an examinationERVWhether the voltage value reaches the programming voltage.

2.Write the upper four bits of the programming addressEADRHregister.

3.Write the lower eight bits of the programming addressEADRLregister.

4.WillFFhwriteEDATHregister.

5.Write the burned dataworkregister.

6.Use online programming instructions (TBLP) to burn. The time selection in the programming command (k) see table3-18Online programming time selection register

surface3-18Online programming time selection register

M4_CK	M2_CK	M1_CK	M0_CK	clock source (KHz)		k(decimal)
X	0	0	0	ICK	4000	255
X	0	1	0	ICK	4000	130
X	1	0	0	ICK	4000	-- [1]
X	1	1	0	ICK	4000	--
0	X	X	1	ECK	32	--
1	0	0	1	ECK	4000	50
1	0	1	1	ECK	4000	26
1	1	0	1	ECK	4000	200
1	1	1	1	ECK	4000	100

[1] "--"Indicates that the clock configuration cannot be programmed online.

read onlineOTPdata time

1 .will readOTPThe upper four bits of the address are writtenEADRHregister.

2.will readOTPThe lower eight bits of the address are writtenEADRLregister.

3.read onlineOTPinstruction(MOVP)read outOTPdata, after executing this instruction, the read data is stored inworkregister.

There are two main commands:

TBLP k
MOVP

inTBLP kis the registerworkThe data in is written toEADRH/EADRLcontent asOTPIn the write address of , the burning time isKinstruction cycles

MOVPwillEADRH/EADRLcontent asOTPThe read address, the read data is placed in the registerworkmiddle.

Using internal programming voltage

NETDregister(address=1Bh)

characteristic	R/W-0	R/W-0	R/W-0	UX	UX	UX	UX	R/W-0
NETD	chp_vpp							

Bit 7 chp_vpp:LCD charge pumpboost enable flag 1 = charge pumpboost to doubleDVDDVoltage 0 = charge pumpThe boost value is given byVLCD_Xcontrol

Operation when using internal programming voltage:

1. open firstENVB
2. set upENPMPLEnableLCDcharge pump.
3. WillLCD_ENset0,closureLCDmodule 4.

WillCHP_VPPEnable

Notice:1.When using internal programming voltage, it is requiredDVDDVoltage greater than or equal to3.25Vand less than3.5V,

2.SuggestVLCDCapacitors are used2.2uF

3.After each burning data, delay time5ms.

4 MCUsInstruction Set

surface4-1 MCUInstruction Set

instruction	operate	instruction cycle	flag bit
ADDLWk	[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	~
BSF f,b	[f<b>]←1	1	~
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	~
CLRFf	[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	Z
DECFSZ f,d	[Destination] ←[f] -1,jump if the result is zero	1	~
GOTO k	PC←k	1	~
HALT	CPU Stop	1	~
INCF f,d	[Destination] ←[f]+1	1	Z
INCFSZ f,d	[Destination] ←[f]+1,jump if the result is zero	1	~
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

Parameter Description:

f:data memory address (00h ~17Fh)

W:working register

k:immediate

d:Destination address selection:d=0The result is stored in the working register,d=1:Results are stored in data memoryfunit b:bit select (0~7)

[f]:fthe content of the address

PC:program counter

C:carry flag

DC:Half plus carry flag

Z:result zero flag
PD:sleep flag
TO:watchdog overflow flag
WDT:watchdog counter

surface4-2 MCUsInstruction set description

1

ADDLW	Add immediate data to working register
Instruction format	ADDLW K (0<=K<=FFh) 8 8
opcode	44h
operate	(W)<—(W)+K
flag bit	C,DC,Z
describe	The contents of the working register plus the immediate valueKThe result is saved to the working register
cycle	1
example ADDLW 08h	Before the instruction is executed: W=08h After the instruction is executed: W=10h

2

ADDPCW	WillWcontent added toPCmiddle
Instruction format	ADDPCW 16
opcode	0008h
operate	(PC)<—(PC)+1+(W) when(W)<=7Fh (PC)<—(PC)+1+(W)-100hthe remaining
flag bit	No
describe	will addressPC+1+Wloaded intoPCmiddle
cycle	1
example1 ADDPCW	Before the instruction is executed: W=7Fh,PC=0212h After the command is executed: PC=0292h
example2 ADDPCW	Before the instruction is executed: W=80h,PC=0212h After the command is executed: PC=0193h
example3 ADDPCW	Before the instruction is executed: W=FEh,PC=0212h After the command is executed: PC=0211h

3

ADDWF	add working register tof
Instruction format	ADDWF f,d 0<=f<=FFh d=0,1 8 8
opcode	whend=0Time04h whend=1Time06h
operate	[target address] <—(f)+(W)
flag bit	C,CD,Z
describe	Willfis added to the contents of the working register. ifdYes0, the result is stored in the working register. if dYes1, the result is saved tofmiddle.
cycle	1
example1 ADDWF f 0	Before the command is executed: f=C2h W=17h after the instruction is executed f=C2h W=D9h
example2 ADDWF f 1	before the command is executed f=C2h W=17h After the instruction is executed f=D9h W=17h

4

ADDWFC	WillW fadd the carry bit
Instruction format	ADDWFCf,d 0<=f<=FFh d=0,1 8 8
opcode	whendfor0when, fortwenty four whendfor1when, for26
operate	(destination address) <—(f)+(W)+C
flag bit	C,DC,Z
describe	Combine the contents of the working register andfThe contents of and the carry bits are added whendfor0When the result is saved to the working register whendfor1When the result is saved tofmiddle
cycle	1
example ADDWFCf,1	before the command is executed C=1 f=02h W=4Dh After the instruction is executed C=0 f=50h W=4Dh

5

ANDLW	Working register and immediate value
Instruction format	ANDLW K 0<=K<=FFh 8 8
opcode	68h
operate	(W)<—(W) AND K
flag bit	Z
describe	Compare the contents of the working register with8bitThe immediate values are ANDed, and the result is stored in the working register.
cycle	1
example ANDLW 5Fh	before the instruction is executed W=A3h after the instruction is executed W=03h

6

ANDWF	the working register andfcontent with
Instruction format	ANDWF f,d 0<=f<=FFh d=0,1 8 8
opcode	whendfor0When the opcode is28h whendfor1When the opcode is2Ah
operate	(destination address) <—(W) AND (f)
flag bit	Z
describe	Combine the contents of the working register andfcontent with ifdfor0The result is saved to the working register if dfor1results are saved tofmiddle
cycle	1
example1 ANDWF f,0	before the instruction is executed W=0Fh f=88h after the instruction is executed W=08h f=88h
example2 ANDWF f,1	before the instruction is executed W=0Fh f=88h after the instruction is executed W=0Fh f=08h

7

BCF	clearfone of
Instruction format	BCF f,b 0<=f<=FFh 0<=b<=7 BCF bf 4 4 8
opcode	ChinbThe value of the field is2*b
operate	(f[b])<—0
flag bit	none
describe	FFirstblocation is0
cycle	1
example BCF FLAG 2	Before the command is executed: FLAG=8Dh After the command is executed: FLAG=89h

8

BSF	FofbLocation1
Instruction format	BSF f,b 0<=f<=FFh 0<=b<=7 BSF bf 4 4 8
opcode	DhinbThe value of the field is2*b
operate	(f[b])<—1
flag bit	none
describe	WillfofbLocation1
cycle	1
example BSF FLAG 2	before the instruction is executed FLAG=89h after the instruction is executed FLAG=8Dh

9

BTFSC	ifbittest as0then jump
Instruction format	BTFSC f,b 0<=f<=FFh 0<=b<=7 BTFSC bf 4 4 8
opcode	EhinbThe value of the field is2*b
operate	Skip if (f[b])=0
flag bit	none
describe	iffofbitbit is0, the next fetched instruction will be dropped, and then a dummy instruction will be executed to form a two-cycle instruction.
cycle	1

example
NODE BTFSC FLAG 2
OP1:
OP2:
	before program execution
	PC=address(NODE)
	After the instruction is executed
	If(FLAG[2])=0
	PC=address(OP2)
	If(FLAG[2])=1
	PC=address(OP1)

10

BTFSS	ifbittest as1, then jump
Instruction format	BTFSS f,b 0<=f<=FFh 0<=b<=7 BTFSS bf 4 4 8
opcode	Fh,inbThe value of the field is2*b
operate	Skip if (f[b])=1
flag bit	none
describe	iffofbitbit is1, the next fetched instruction will be dropped, and then a dummy instruction will be executed to form a two-cycle instruction.
cycle	1

example
NODE BTFSS FLAG 2
OP1:
OP2:
	before program execution
	PC=address(NODE)
	After the instruction is executed
	If(FLAG[2])=0
	PC=address(OP1)
	If(FLAG[2])=1
	PC=address(OP2)

11

CALL	subroutine call
Instruction format	CALL K 0<=K<=1FFFh 3 13
opcode	100
operate	(top stack) <—PC+1 PC<—K
flag bit	none
describe	Subroutine call, firstPC+1Push onto the stack, then download the immediate address toPCmiddle.
cycle	1

12

CLRF	clearf
Instruction format	CLRF f 0<=f<=255 8 8
opcode	02h
operate	(f)<—0
flag bit	Z
describe	WillfThe content is cleared
cycle	1
example CLRF WORK	before the instruction is executed WORK=5Ah after the instruction is executed WORK=00h

* Note. whenclrf statusregister, the flag bitZwill not be set high

13

CLRWDT	clear watchdog timer
Instruction format	CLRWDT 16
opcode	0006h
operate	Watchdog counter clear
flag bit	none
describe	clear watchdog timer
cycle	1
example CLRWDT	After the instruction is executed WDT=0

14

COMF	fNegate
Instruction format	COMF f,d 0<=f<=255 d=0,1 8 8
opcode	whendfor0When the opcode is1Ch whendfor1When the opcode is1Eh
operate	(destination address) <—NOT(f)
flag bit	Z
describe	WillfThe content is reversed, whendfor0, the result is stored in the working register, whendfor1, the result is saved tofmiddle.
cycle	1
example COMF f,0	before the instruction is executed W=88h,f=23h after the instruction is executed W=DCh,f=23h
example2 COMF f,1	before the instruction is executed W=88h,f=23h after the instruction is executed W=88h,f=DCh

15

DECF	freduce1
Instruction format	DECF f,d 0<=f<=255 d=0,1 8 8
opcode	whendfor0When the opcode is10h whendfor1When the opcode is12h
operate	(destination address) <—(f)-1
flag bit	Z
describe	Fcontent less1 whendfor0, the result is stored in the working register whendfor1, the result is saved tofmiddle.
cycle	1
example DECF f,0	before the instruction is executed W=88h f=23h after the instruction is executed W=22h f=23h
example2 DECF f,1	before the instruction is executed W=88h f=23h after the instruction is executed W=88h f=22h

16

DECFSZ	freduce1if0then jump
Instruction format	DECFSZ f,d 0<=f<=FFh d=0,1 8 8
opcode	whendfor0When the opcode is14h whendfor1When the opcode is16h
operate	(destination address) <—(f)-1,if the result is0jump
flag bit	none
describe	fcontent less1. ifdfor0, the result is stored in the working register. if dfor1, the result is saved tofmiddle if the result is0, the next instruction that has been fetched will be discarded, and anNOP Instructions form a two-cycle instruction.
cycle	1

example
Node DECFSZ FLAG,1
OP1:
OP2:
	before the instruction is executed
	PC=address(Node)
	after the instruction is executed
	(FLAFG)=(FLAG)-1
	If(FLAG)=0
	PC=address(OP2)
	If(FLAG)!=0
	PC=address(OP1)

17

GOTO	unconditional jump
Instruction format	GOTO K 0<=K<=1FFFh 3 13
opcode	101
operate	PC<—K
flag bit	none

describe	immediate address loadPC
cycle	1

18

HALT	stopCPUclock
Instruction format	HALT 16
opcode	0005h
operate	CPUstop
flag bit	none
describe	CPUThe clock stops, the crystal still works,CPUCan be restarted by internal or external interrupt.
cycle	1

19

INCF	fadd1
Instruction format	INCF f,d 0<=f<=FFh d=0,1 8 8
opcode	whendfor0When the opcode is08h whendfor1When the opcode is0Ah
operate	(destination address) <—(f)+1
flag bit	Z
describe	fadd1 ifdfor0, the result is stored in the working register ifdfor1, the result is saved tofmiddle.
cycle	1
example INCF f,0	before the instruction is executed W=88h f=23h after the instruction is executed W=24h f=23h
example2 INCF f,1	before the instruction is executed W=88h f=23h after the instruction is executed W=88h f=24h

20

INCFSZ	fadd1, if the result is0jump
Instruction format	INCFSZ f,d 0<=f<=FFh d=0,1 8 8
opcode	whendfor0When the opcode is0Ch whendfor1When the opcode is0Eh
operate	(destination address) <—(f)+1if the result is0just jump
flag bit	none
describe	fcontent plus1. ifdfor0, the result is stored in the working register. if dfor1, the result is saved tofmiddle if the result is0, the next instruction that has been fetched will be discarded, and anNOP Instructions form a two-cycle instruction.
cycle	1
example Node INCFSZ FLAG,1	before the instruction is executed PC=address(Node)

OP1:
OP2:
	after the instruction is executed
	(FLAFG)=(FLAG)+1
	If(FLAG)=0
	PC=address(OP2)
	If(FLAG)!=0
	PC=address(OP1)

twenty one

IORLW	working register and immediate data or
Instruction format	IORLW K 0<=K<=FFh 8 8
opcode	6Ch
operate	(W)<—(W)|K
flag bit	Z
describe	The immediate value is ORed with the contents of the working register. The result is saved to the working register.
cycle	1
example IORLW 85H	before the instruction is executed W=69h after the instruction is executed W=EDh

twenty two

IORWF	fwith the working register or
Instruction format	IORWF f,d 0<=f<=FFh d=0,1 8 8
opcode	whendfor0, the opcode is2Ch whendfor1, the opcode is2Eh
operate	(destination address) <—(W)|(f)
flag bit	Z
describe	fand working register or whendfor0, the result is stored in the working register whendfor1, the result is saved tofmiddle
cycle	1
example IORWF f,1	before the command is executed W=88h f=23h after the command is executed W=88h f=ABh

twenty three

MOVFW	transfer to working register
Instruction format	MOVFW f 0<=f<=255 8 8
opcode	20h
operate	(W)<—(f)
flag bit	none
describe	transfer data fromftransfer to working register
cycle	1
example MOVFW f	before the instruction is executed W=88h f=23h after the instruction is executed W=23h f=23h

twenty four

MOVLW	Transfer immediate data to working register
Instruction format	MOVLW K 0<=K<=FFh 8 8
opcode	40h
operate	(W)<—K
flag bit	none
describe	Will8bitThe immediate data is transferred to the working register
cycle	1
example MOVLW 23H	before the instruction is executed W=88h after the instruction is executed W=23h

25

MOVWF	transfer the value of the working register tofmiddle
Instruction format	MOVWF f 0<=f<=FFh 8 8
opcode	22h
operate	(f)<—(W)
flag bit	none
describe	transfer the value of the working register tofmiddle
cycle	1
example MOVWF f	before the instruction is executed W=88h f=23h after the instruction is executed W=88h f=88h

26

NOP	no action
Instruction format	NOP 16
opcode	0000h
operate	no action
flag bit	none
describe	no action
cycle	1

27

RETFIE	return from interrupt
Instruction format	RETFIE 16
opcode	0002h
operate	(Top Stack)=>PC Pop Stack 1=>GIE
flag bit	none
describe	PCGet from the top of the stack, then pop the stack, set the global interrupt enable bit to1
cycle	1

28

RETLW	Return and send the immediate value to the working register
Instruction format	RETLW K 0<=K<=FFh 8 8
opcode	48h
operate	(W)<—K (Top Stack)=>PC Pop Stack
flag bit	none
describe	Will8bitThe immediate data is sent to the working register,PCThe value is obtained from the top of the stack, and then popped from the stack
cycle	1

29

RETURN	return from subroutine
Instruction format	RETURN 16
opcode	0003h
operate	(Top Stack)=>PC Pop Stack
flag bit	none
describe	PCThe value is obtained from the top of the stack, and then popped from the stack
cycle	1

30

RLF	left shift with carry
Instruction format	RLF f,d 0<=f<=FFh d=0,1 8 8
opcode	whendfor0, the opcode is34h whendfor1, the opcode is36h
operate	(target address[n+1])<—(f[n]) (target address[0]) <—C C<—(f[7])
flag bit	C,Z
describe	FShift left one bit with carry ifdfor0, the result is saved to the working register ifdfor1, the result is saved tofmiddle
cycle	1
example RLF f,1	before the instruction is executed C=0 W=88h f=E6h after the instruction is executed C=1 W=88h f=CCh

31

RRF	shift right with carry
Instruction format	RRF f,d 0<=f<=FFh d=0,1 8 8
opcode	whendfor0, the opcode is3Ch whendfor1, the opcode is3Eh
operate	(target address[n-1])<—(f[n]) (target address[7]) <—C C<—(f[7])
flag bit	C
describe	FShift right by one bit with carry ifdfor0, the result is saved to the working register ifdfor1, the result is saved tofmiddle
cycle	1
example RRF f,0	before the instruction is executed C=0 W=88h f=95h after the instruction is executed C=1 W=4Ah f=95h

32

SLEEP	Crystal stopped
Instruction format	SLEEP 16
opcode	0004h
operate	CPUCrystal stopped
flag bit	PD
describe	CPUThe crystal oscillator stops.CPURestart via external interrupt source
cycle	1

33

SUBLW	Subtract the value of the working register from the immediate value
Instruction format	SUBLW K 0<=K<=FFh 8 8
opcode	58h
operate	(W)<—K-(W)
flag bit	C,DC,Z
describe	8bitThe immediate value is subtracted from the value of the working register, and the result is stored in the working register
cycle	1
example SUBLW 02H	before the instruction is executed W=01h after the instruction is executed W=01h C=1(Represents no borrowing)Z=0(represents a non-zero result)
example2 SUBLW 02H	before the instruction is executed W=02h after the instruction is executed W=00h C=1(Represents no borrowing)Z=1(means the result is zero)
example2 SUBLW 02H	before the instruction is executed W=03h after the instruction is executed W=FFh C=0(Represents a borrow)Z=0(represents a non-zero result)

34

SUBWF	fThe value of minus the value of the working register
Instruction format	SUBWF f,d 0<=f<=FFh d=0,1 8 8
opcode	whendfor0, the opcode is18h whendfor1, the opcode is1Ah
operate	(destination address) <—(f)-(W)
flag bit	C,DC,Z
describe	fminus the value of the working register. ifdfor0, the result is saved to the working register ifdfor1, the result is saved tofmiddle
cycle	1
example SUBWF f,1	before the instruction is executed f=33h W=01h after the instruction is executed f=32h C=1 Z=0
example2 SUBWF f,1	before the instruction is executed f=01h W=01h after the instruction is executed f=00h C=1 Z=1
example3 SUBWF f,1	before the instruction is executed f=04h W=05h after the instruction is executed f=FFh C=0 Z=0

35

SUBWFC	subtraction with borrow
Instruction format	SUBWFC f,d 0<=f<=FFh d=0,1 8 8
opcode	whendfor0, the opcode is38h whendfor1, the opcode is3Ah
operate	(destination address) <—(f)-(W)-1+C
flag bit	C,DC,Z
describe	fThe value of minus the value of the working register ifdfor0, the result is saved to the working register ifdfor1, the result is saved tofmiddle
cycle	1
example SUBWFC f,1	before the instruction is executed W=01h f=33h C=1 after the instruction is executed f=32h C=1 Z=0
example2 SUBWFC f,1	before the instruction is executed W=01h f=02h C=0 after the instruction is executed f=00h C=1 Z=1
example3 SUBWFC f,1	before the instruction is executed W=05h f=04h C=0 after the instruction is executed f=FEh C=0 Z=0

36

XORLW	XOR the value of the working register with the immediate value
Instruction format	XORLW K 0<=K<=FFh 8 8
opcode	70h
operate	(W)<—(W)ˆK
flag bit	Z
describe	8bitThe immediate value is XORed with the value of the working register, and the result is stored in the working register
cycle	1
example XORLW 5Fh	before the instruction is executed W=Ach after the instruction is executed W=F3h

37

XORWF	fis XORed with the value of the working register
Instruction format	XORWF f,d 0<=f<=FFh d=0,1 8 8
opcode	whendfor0, the opcode is30h whendfor1, the opcode is32h
operate	(destination address) <—(W)ˆ(f)
flag bit	Z
describe	FThe value of is XORed with the value of the working register, whendfor0, the result is stored in the working register whendfor1, the result is saved tofmiddle
cycle	1
example XORWF f,1	before the instruction is executed W=ACh f=5Fh after the instruction is executed f=F3h

5Electrical Characteristics

5.1limit value

surface5-1 CSU8RP1001limit value

parameter	scope	unit
DVDD, AVDD	2.4~3.6	V
pin input voltage	- 0.3~DVDD+0.3	V
Operating temperature	0~+85 (Operating Voltage:2.4~3.6V)	° C
	- 10~+85(Operating Voltage:2.8~3.6V)	
storage temperature	- 55~+150	° C
Soldering temperature, time	220°C,10second	

5.2DC characteristics (DVDD = 2.8V,T_A= 25°C, unless otherwise stated, all of this condition)

surface5-2 CSU8RP1001DC characteristics

symbol	parameter	Test Conditions	minimum	Typical value	maximum value	unit
DVDD	digital power		2.4	2.8	3.6	V
AVDD	analog power		2.4	2.8	3.6	
IDD1	supply current1	MCK = 4MHz CPUCLK=MCK/2 charge pump,ADCOpen		3.5		mA
IDD2	supply current2	Internal oscillator off,MCK = 32768Hz, LCDopen		8	15	uA
IPO	Supply current in sleep mode	sleep command		1.5		uA
VIH	digital input high	PT2	DVDD-0.6			V
VIL	digital input low	PT2			0.6	V
IPU	pull-up current	PT1,2,3 Vin = 0		28		uA
IOH	High level output current	VOH=DVDD-0.3V		3	10	mA
IOL	Low level output current	VOL=0.3V		3	10	mA
VS	Modulator Power	LDOS[1:0]= 11		2.3		V
	Reference Voltage Temperature Coefficient			50		ppm/°C
VLREF	Internal for low voltage detection External reference voltage			1.2		V
TCLREF	for low voltage detection Internal reference voltage temperature system number	T_A= -10~85°C		100		ppm/°C
VLBAT	Low battery detection voltage	S_LB[2:0]=000		2.4		V
		S_LB[2:0]=001		2.5		
		S_LB[2:0]=010		2.6		
		S_LB[2:0]=011		2.7		
		S_LB[2:0]=100		2.8		
		S_LB[2:0]=101		3.6		
		S_LB[2:0]=110		1.24		
		S_LB[2:0]=111		3.2		
FRC	built-inRCoscillator		3.88	4.0	4.12	MHz
FWDT	Built-in watchdog clock		2.7	3	3.3	KHz

5.3 ADCscharacteristics (VS = 2.3V,T_A= 25°C, unless otherwise stated, all of this condition)

surface5-3 CSU8RP1001 ADCcharacteristics

parameter		condition	minimum	Typical value	maximum value	unit
simulation enter	Analog input range		AGND-0.1		AVDD+0.1	V
	Full-scale input voltage (AIN+) - (AIN-)				±VREF/PGA	V
	Differential input impedance	Bufferclosure		Ts/Cs		Ω
		BufferOpen		10		MΩ
system performance	Resolution	No missing codes		twenty four		Bits
	Input noise (rms)	gain=68 DataRate=7.8K		1.1		μV
	Integral linearity	gain=68 DataRate=7.8K		±0.02		% of FS
	offset error	gain=68 DataRate=7.8K		10		μV
	offset error drift	gain=68 DataRate=7.8K		0.05		μV/°C
	gain error	gain=68 DataRate=7.8K		10		**%**
	Gain Error Drift	gain=68 DataRate=7.8K TCOM=0		50		ppm/°C
		gain=68 DataRate=7.8K TCOM=1		- 50		

5.4DC characteristics (DVDD = 3.3V,T_A= 25ºC, unless otherwise stated, all of this condition)

surface5-4 CSU8RP1001DC characteristics

symbol	parameter	Test Conditions	minimum	Typical value	maximum value	unit
IDD1	supply current1	MCK = 4MHz CPUCLK=MCK/2 charge pump,ADCbeat open		4		mA
IDD2	supply current2	Internal oscillator off MCK = 32768Hz LCDopen		8	15	uA
IPO	Supply current in sleep mode	sleep command		1.5		uA
VIH	digital input high	PT2	DVDD-0.7			V
VIL	digital input low	PT2			0.7	V
IPU	pull-up current	PT1,2,3 Vin = 0		30		uA
IOH	High level output current	VOH=DVDD-0.3V		3	10	mA
IOL	Low level output current	VOL=0.3V		3	10	mA
VS	VSoutput	LDOS[1:0]= 11	2	2.3	2.4	V
		LDOS[1:0]= 10	2.4	2.5	2.6	
		LDOS[1:0]= 01	2.5	2.8	2.8	
		LDOS[1:0] = 00	2.6	3	3	
	Reference Voltage Temperature Coefficient			100		ppm/°C
VLREF	Internal for low voltage detection reference voltage			1.2		V
TCLREF	for low voltage detection Internal reference voltage temperature coefficient	T_A= -10~85ºC		30		ppm/ºC
VLBAT	Low battery detection voltage	S_LB[2:0]=000		2.4		V
		S_LB[2:0]=001		2.5		
		S_LB[2:0]=010		2.6		
		S_LB[2:0]=011		2.7		
		S_LB[2:0]=100		2.8		
		S_LB[2:0]=101		3.6		
		S_LB[2:0]=110		1.24		
		S_LB[2:0]=111		3.2		
FRC	built-inRCoscillator		3.88	4.0	4.12	MHz
FWDT	Built-in watchdog clock		2.7	3	3.3	KHz

5.5 ADCscharacteristics (VS = 3.0V,T_A= 25°C, unless otherwise stated, all of this condition)

surface5-5 CSU8RP1001 ADCcharacteristics

parameter		condition	minimum	Typical value	maximum value	unit
simulation enter	Analog input range		AGND-0.1		AVDD+0.1	V
	Full-scale input voltage (AIN+) - (AIN-)				±VREF/PGA	V
	Differential input impedance	Bufferclosure		Ts/Cs		Ω
		BufferOpen		10		MΩ
system performance	Resolution	No missing codes		twenty four		Bits
	Input noise (rms)	gain=136 DataRate=30		80		nV
	Integral linearity	gain=136 DataRate=30		±0.01		% of FS
	offset error	gain=136 DataRate=30		10		μV
	offset error drift	gain=136 DataRate=30		0.05		μV/°C
	gain error	gain=136 DataRate=30		10		%
	Gain Error Drift	gain=136 DataRate=30 TCOM=0		50		ppm/°C
		gain=136 DataRate=30 TCOM=1		- 50		