



芯海科技
CHIPSEA

CSU3AF10 用户手册

REV 1.0

芯海科技（深圳）股份有限公司

地 址：深圳市南山区蛇口南海大道1079号花园城数码大厦A座9楼

电 话：+(86 755)86169257 传 真：+(86 755)86169057

网 站：www.chipsea.com 邮 编：518067

微信号：芯海科技



版本历史

历史版本	修改内容	时间	修改人
REV 1.0	第一版	2021-01-13	

目 录

版本历史	2
目 录	3
1 产品概述	8
1.1 功能描述	8
1.2 主要特性	8
1.3 PIN 配置	9
1.3.1 QFN28 PIN 配置	9
1.3.2 QFN24 PIN 配置	10
1.4 引脚说明	11
2 标准功能	13
2.1 CPU 核	13
2.1.1 CPU 核概述	13
2.1.2 CPU 指令执行流程	14
2.1.3 存储器	15
2.1.4 数据存储器寻址	17
2.1.5 PC 跳转	17
2.1.6 中断向量表	18
2.1.7 堆栈	19
2.1.8 非法指令复位	19
2.1.9 CPU 核寄存器映射	19
2.1.10 CPU 核寄存器描述	20
2.2 外设地址映射	27
2.3 时钟管理单元 (CLKMU)	29
2.3.1 概述	29
2.3.2 时钟框图	30
2.3.3 基本配置	30
2.3.4 功能描述	31
2.3.5 寄存器映射	31
2.3.6 寄存器描述	32
2.4 复位管理单元 (RSTMU)	37
2.4.1 概述	37
2.4.2 功能描述	37
2.4.3 寄存器映射	40
2.4.4 寄存器描述	40
2.5 外部中断	42
2.5.1 概述	42
2.5.2 特性	42
2.5.3 框图	42
2.5.4 功能描述	42
2.5.5 寄存器映射	43
2.5.6 寄存器描述	43
2.6 定时器 0	47

2.6.1	概述	47
2.6.2	特性	47
2.6.3	框图	47
2.6.4	基本配置	47
2.6.5	功能描述	47
2.6.6	寄存器映射	48
2.6.7	寄存器描述	48
2.7	GPIO	50
2.7.1	概况	50
2.7.2	特性	51
2.7.3	框图	51
2.7.4	基本配置	51
2.7.5	功能描述	51
2.7.6	寄存器映射	52
2.7.7	寄存器描述	52
3	增强功能	60
3.1	HALT 和 SLEEP 模式	60
3.1.1	Halt 模式	60
3.1.2	Sleep 模式	60
3.2	SYSCFG	61
3.2.1	IO 复用功能	61
3.2.2	寄存器映射	61
3.2.3	寄存器描述	62
3.3	看门狗(WDT)	68
3.3.1	概况	68
3.3.2	特性	68
3.3.3	框图	68
3.3.4	基本配置	68
3.3.5	功能描述	68
3.3.6	寄存器映射	69
3.3.7	寄存器描述	69
3.4	窗看门狗(WWDT)	71
3.4.1	概况	71
3.4.2	特性	71
3.4.3	框图	71
3.4.4	基本配置	71
3.4.5	功能描述	71
3.4.6	寄存器映射	73
3.4.7	寄存器描述	73
3.5	定时器 2	75
3.5.1	概述	75
3.5.2	特性	75
3.5.3	框图	75
3.5.4	基本配置	76
3.5.5	功能描述	76
3.5.6	寄存器映射	77
3.5.7	寄存器描述	77
3.6	定时器 3	79

3.6.1	概述	79
3.6.2	特性	79
3.6.3	框图	79
3.6.4	基本配置	80
3.6.5	功能描述	80
3.6.6	寄存器映射	82
3.6.7	寄存器描述	82
3.7	定时器 4	85
3.7.1	概述	85
3.7.2	特性	85
3.7.3	框图	86
3.7.4	基本配置	86
3.7.5	功能描述	86
3.7.6	寄存器映射	87
3.7.7	寄存器描述	88
3.8	模拟比较器 (ACMP)	91
3.8.1	概述	91
3.8.2	特性	91
3.8.3	框图	91
3.8.4	基本配置	92
3.8.5	功能描述	92
3.8.6	寄存器描述	93
3.9	模数转换器 (ADC)	100
3.9.1	概述	100
3.9.2	特性	100
3.9.3	功能框图	100
3.9.4	基本配置	101
3.9.5	功能描述	101
3.9.6	寄存器映射	104
3.9.7	寄存器描述	105
3.10	串行通信接口	109
3.10.1	概述	109
3.10.2	特性	109
3.10.3	功能框图	109
3.10.4	基本配置	110
3.10.5	功能描述	110
3.10.6	寄存器映射	110
3.10.7	寄存器描述	111
3.11	FLASH 控制器	116
3.11.1	概述	116
3.11.2	特性	116
3.11.3	功能描述	116
3.11.4	寄存器映射	117
3.11.5	寄存器描述	118
3.12	在线调试功能 (ICD)	122
3.12.1	在线调试功能概述	122
3.13	烧录模块	123
3.13.1	烧录器接口	123
3.14	代码选项	124

3.14.1	OPTION0	124
3.14.2	OPTION1	125
3.14.3	ICD 功能配置选项.....	125
3.14.4	FLASH 主程序区擦写加密位	126
4	MCU 指令集.....	128
4.1	基础指令汇总	128
4.2	扩展指令汇总	130
4.3	基础指令描述	131
4.3.1	ADDLW	131
4.3.2	ADDPCW	131
4.3.3	ADDWF	131
4.3.4	ADDWFC	132
4.3.5	ANDLW	132
4.3.6	ANDWF	132
4.3.7	BCF	133
4.3.8	BSF	133
4.3.9	BTFSC	133
4.3.10	BTFSS	134
4.3.11	CALL	134
4.3.12	CLRF	134
4.3.13	CLRWDT	135
4.3.14	COMF	135
4.3.15	DAW	135
4.3.16	DECF	136
4.3.17	DECFSZ	136
4.3.18	GOTO	137
4.3.19	HALT	137
4.3.20	INCF	137
4.3.21	INCFSZ	138
4.3.22	IORLW	138
4.3.23	IORWF	138
4.3.24	MOVFW	139
4.3.25	MOVLW	139
4.3.26	MOVWF	139
4.3.27	NOP	140
4.3.28	PUSH	140
4.3.29	POP	140
4.3.30	RETFIE	140
4.3.31	RETLW	141
4.3.32	RETURN	141
4.3.33	RLF	141
4.3.34	RRF	141
4.3.35	SLEEP	142
4.3.36	SUBLW	142
4.3.37	SUBWF	143
4.3.38	SUBWFC	143
4.3.39	SWAPF	144
4.3.40	XORLW	144
4.3.41	XORWF	144
4.3.42	RL	145
4.3.43	RR	145
4.3.44	SETF	145
4.4	扩展指令描述	146

4.4.1	CALLL	146
4.4.2	GOTOL.....	146
4.4.3	ADDLFSR.....	146
4.4.4	SUBLFSR.....	146
4.4.5	MOVFWL.....	147
4.4.6	MOVWFL.....	147
5	电气特性.....	148
5.1	极限值	148
5.2	直流特性 (VDD = 3.3V, TA = 25 ℃, 如无其他说明则都是此条件)	148
5.3	ADC 特性 (VDD = 3.3V, TA = 25 ℃, 如无其他说明则都是此条件)	149
5.4	外部 24MHz XTAL 振荡器	150
5.5	内部 24MHz RC 振荡器	151
5.6	内部 10KHz RC 振荡器	151
5.7	比较器	151
5.8	内部参考电压	152
5.9	LDO 规格和 POWER 管理	152
5.10	FLASH 读写特性	152
5.11	可靠性指标	152
6	封装图	154
6.1	QFN-28 PIN	154
6.2	QFN-24 PIN	154
7	单片机产品命名规则	156
7.1	产品型号说明	156
7.2	命名举例说明	157
7.3	产品印字说明	157

1 产品概述

1.1 功能描述

CSU3AF10 是一个带 12-bit ADC 的 8 位 CMOS 单芯片 RISC MCU，内置 32K×16 位 FLASH 程序存储器。

1.2 主要特性

高性能的 RISC CPU

- 8 位单片机 MCU
- 内置 32K×16 位程序存储器 FLASH
- 2K 字节数据存储器 (SRAM)
- 只有 48 条单字指令
- 12 级 PC 存储堆栈
- 12 级 PUSH 和 POP 堆栈

振荡器

- 内带 24MHz 振荡器，精度为 $\pm 1\%$ @5V, 25°C
- 支持外部 24M 晶体振荡器
- 内部 10KHz 振荡器；精度 $\pm 10\%$ ；

外设特性

- 23 个双向 I/O 口；
- 2 个 8 位定时器，2 个 12 位定时器
- 3 路 PWM 输出，1 路互补 PWM 输出
- 15 个中断源，3 个外部中断，12 个内部中断
- 15 个具有唤醒功能的输入口
- USB-TYPEC/USB-PD
 - 两组独立 CC 口，共用 PD 模块
 - 支持 USB-PD3.0 协议
 - 支持 PPS
 - 支持 QC4.0
 - 支持上拉恒流源输出 80uA、180uA、330uA
 - 支持 5.1K 下拉
 - 支持 Dead Battery 检测
- USB-DP/DM
 - 四组独立 DP、DM
 - 支持华为 FCP、SCP
 - 支持三星 AFC
 - 支持 QC2.0、QC3.0
 - 支持 VOOC
 - 支持 Apple2.4、三星充电协议、BC1.2
- 17（外部）+3（内部）路 12-bit ADC

— 内部 1.0V、2.5V、VDD 三种参考电压

- 1.0V、2.5V 参考电压，精度 $\pm 1\%$
- 低电压检测 (LVD) 功能，内部提供 2.1V、2.4V 低电压比较
- 输入逻辑电平电压可配置
- 支持在线仿真 (ICD)
- 1 路 UART，波特率最高支持 115200
- NTC 检测功能，占用 1 个 IO 完成温度测量
- 支持 32 位 CHIP ID；
- 指令周期 12MHz/6MHz/3MHz
- 2 个模拟比较器

专用微控制器的特性

- 上电复位 (POR)
- 上电复位和硬件复位 30ms 延迟定时器
- 内带低电压检测/低电压复位 (LVD/LVR)
- 8 位定时器 0
- 8 位定时/计数器 2
- 12 定时/计数器 3
- 12 定时/计数器 4
- 扩展型看门狗定时器 (10K WDT)
- 7 位向下计数窗看门狗 WWDT

CMOS 技术

- 电压工作范围
 - 2.2V~5.5V@fcpu=12MHz(指令周期)
- 工作温度范围
 - 环境温度：-40~85 °C；结温：-40~105 °C

低功耗特性

- MCU 工作电流
 - 正常模式 3.5mA@12MHz, 3.3V
 - 休眠模式 <10uA
 - Type-C 模块工作休眠功耗 50uA

封装

- QFN28；
- QFN24

应用范围

- 快充电源、HUB、Dock

1.3 PIN 配置

1.3.1 QFN28 PIN 配置

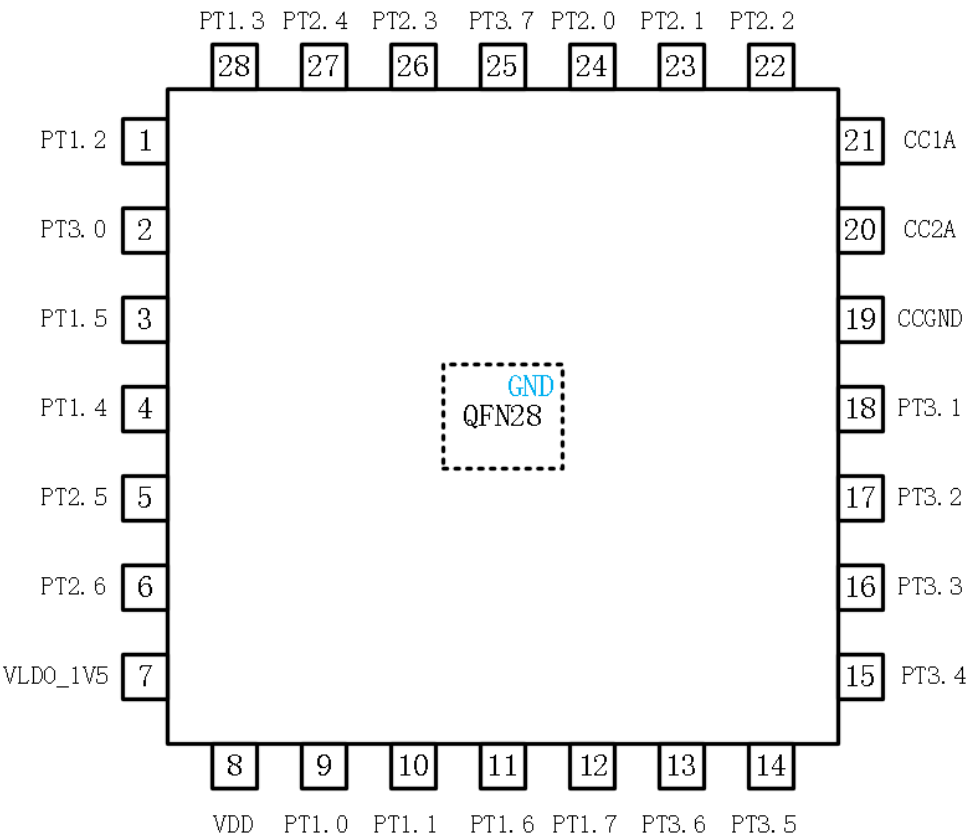


图 1-1 QFN28 管脚图

1.3.2 QFN24 PIN 配置

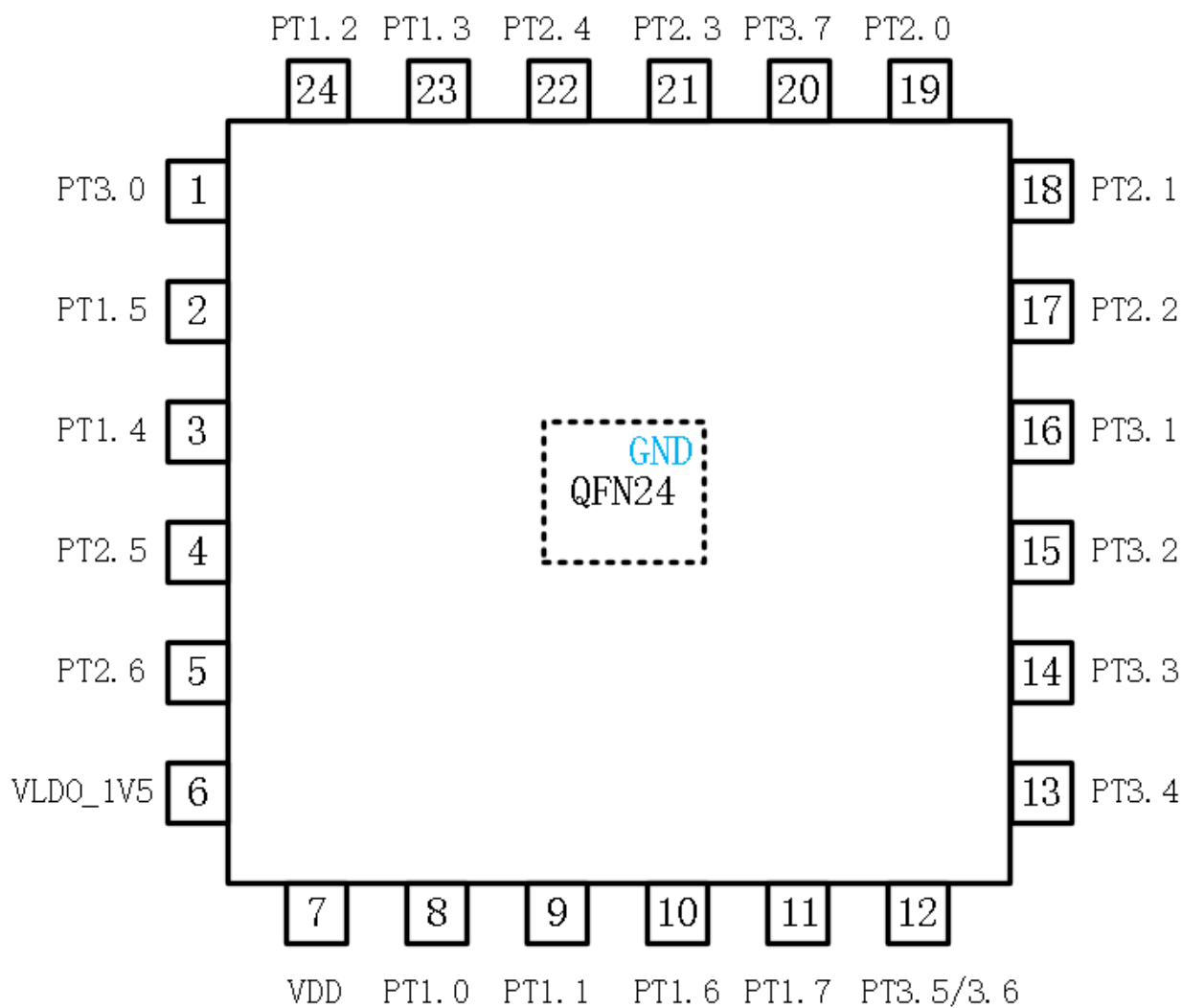


图 1-2 QFN24 管脚图

1.4 引脚说明

表 1-1 引脚说明表

管脚名称	输入/ 输出	QFN28 管 脚序号	QFN24 管 脚序号	描述
VSS	P	GND PAD	GND PAD	地
PT1.5/INT0/SWD_CLK	I/O	3	2	IO; 外部中断 0 输入, 具有唤醒功能; 烧录时钟线; 外部晶体管脚 XO
PT1.4/INT0/SWD_DAT	I/O	4	3	IO; 外部中断 0 输入, 具有唤醒功能; 烧录数据线; 外部晶体管脚 XI
PT1.3/ $\overline{RST}$ /INT0/VPP	I/O	28	23	IO, 只开漏输出; 复位输入, 低电平有效; 外部中断 0 输入, 具有唤醒功能; 烧录电压
PT1.2/INT0/PWM4/T3	I/O	1	24	IO; 外部中断 0 输入, 具有唤醒功能; PWM4 输出; 定时器 3 外部输入
PT3.0/AIN0/VREF/NTC	I/O	2	1	IO; 模拟输入通道 0; 参考电压 VREF 输出 (或者 ZTC 恒流源输出待选); 外接负温度系数电阻端口;
PT2.3/AIN11/INT2/DP_B	I/O	26	21	IO; 模拟输入通道 11, 外部中断 2 输入, 唤醒功能; USB B 口 D+端; PWM2 输出
PT2.4/AIN12/INT2/DM_B	I/O	27	22	IO; 模拟输入通道 12, 外部中断 2 输入, 唤醒功能; USB B 口 D-端; PWM3 输出
PT2.5/AIN13/INT2	I/O	5	4	IO; 模拟输入通道 13, 外部中断 2 输入, 唤醒功能; 互补 PWM3H 输出; 定时器 2 外部输入
PT2.6/AIN14/INT2	I/O	6	5	IO; 模拟输入通道 14, 外部中断 2 输入, 唤醒功能; 互补 PWM3L 输出; 定时器 4 外部输入
VLDO_1V5	I/O	7	6	内部 1.5V, LDO 外接电容管脚;
PT1.0	I/O	9	8	IO;
PT1.1	I/O	10	9	IO;
PT1.6	I/O	11	10	IO;
PT1.7	I/O	12	11	IO;
VDD	P	8	7	电源
PT3.7/AIN7	I/O	25	20	IO; 模拟输入通道 7 (IMON)
PT3.1/AIN1/TX/DP_A/INT2	I/O	18	16	IO; 模拟输入通道 1; 串口 TX; USB A 口 D+端; 外部中断 2 输入, 唤醒功能;
PT3.2/AIN2/RX/DM_A/INT2	I/O	17	15	IO; 模拟输入通道 2; 串口 RX; USB A 口 D-端; 外部中断 2 输入, 唤醒功能;
PT3.3/AIN3/TX/DP_D/CC1_B/INT2	I/O	16	14	IO; 模拟输入通道 3; 串口 TX; USB D 口 D+端; USB TYPEC(B 组)口 CC1; 外部中断 2 输入, 唤醒功能;
PT3.4/AIN4/RX/DM_D/CC2_B/INT2	I/O	15	13	IO; 模拟输入通道 4; 串口 RX; USB D 口 D-端; USB TYPEC 口(B 组)CC2; 外部中断 2 输入, 唤醒功能;
PT3.5/AIN5/TX/DP_C/INT2	I/O	14	12	IO; 模拟输入通道 5; 串口 TX; USB C 口 D+端; 外部中断 2 输入, 唤醒功能;
PT3.6/AIN6/RX/DM_C/	I/O	13	12	IO; 模拟输入通道 6; 串口 RX; USB C 口

INT2				D-端：外部中断 2 输入，唤醒功能；
AIN15/CC1_A	A	21	/	模拟输入通道 15; USB TYPEC(A 组)口 CC1
AIN16/CC2_A	A	20	/	模拟输入通道 16; USB TYPEC(A 组)口 CC2
PT2.0/AIN8	I/O	24	19	IO；模拟输入通道 8；软件模拟 I2C 口
PT2.1/AIN9	I/O	23	18	IO；模拟输入通道 9；软件模拟 I2C 口
PT2.2/AIN10/INT1	I/O	22	17	IO；模拟输入通道 10;外部中断 1 输入，唤醒功能
CCGND		19		特殊管脚，接地或浮空

2 标准功能

2.1 CPU 核

2.1.1 CPU 核概述

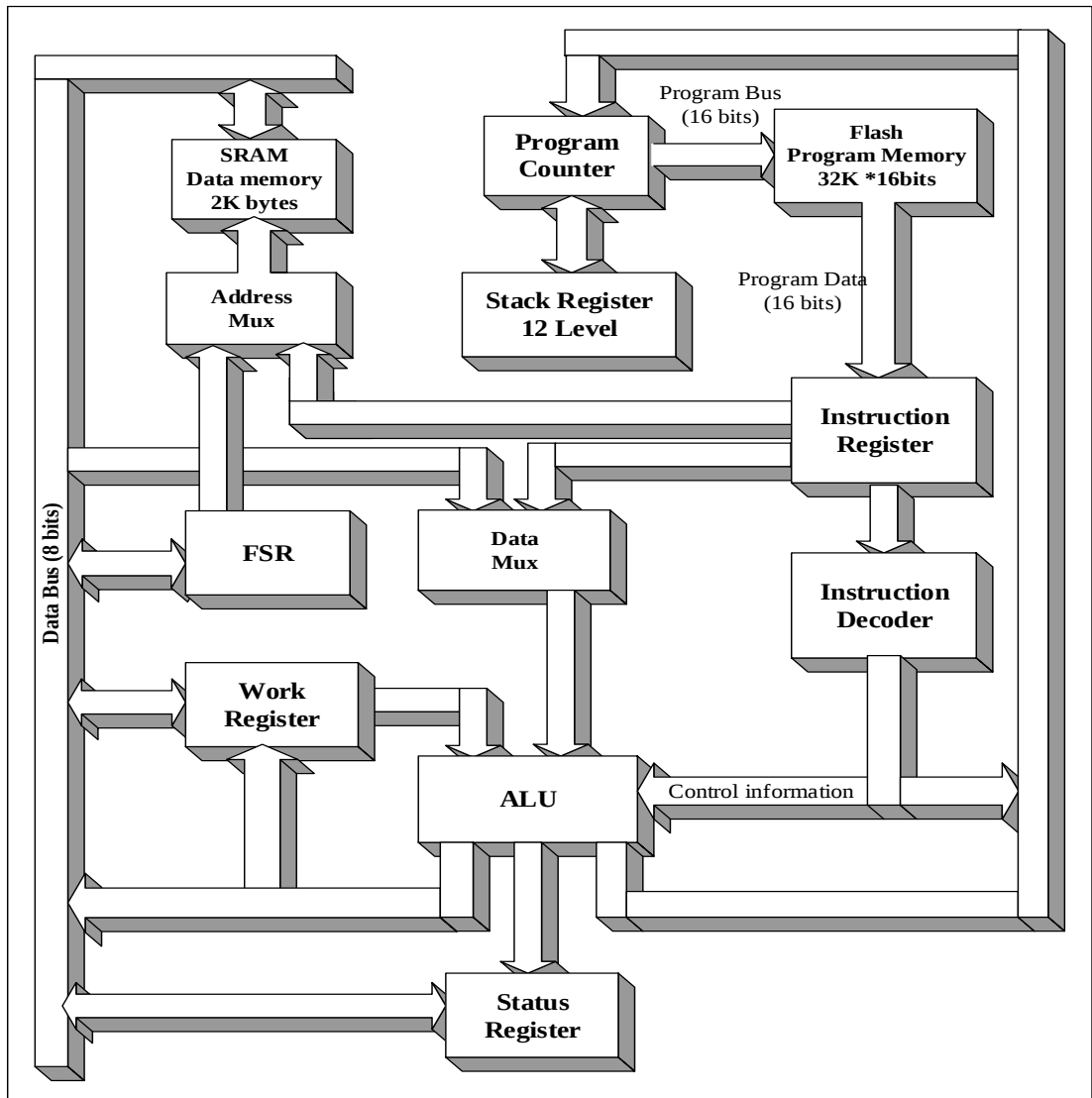


图 2-1 CSU3AF10 CPU 核的功能模块图

从 CPU 核的功能模块图中，可以看到它主要包含 7 个主要寄存器及 2 个存储器单元。

表 2-1 MCU 架构说明

模块名称	描述
程序计数器	此寄存器在 CPU 的工作周期期间起到很重要的作用，它记录 CPU 每个周期处理程序存储器中指令的指针。在一个 CPU 周期中，程序计数器将程序存储器地址（15bits），指令指针推送到程序存储器，然后自动加 1 以进行下一次周期。
堆栈寄存器	堆栈寄存器是用来记录程序返回的指令指针。当程序调用函数，程序计数器会将指令指针推送到堆栈寄存器。在函数执行结束之后，堆栈寄存器会将指令指针送

	回到程序计数器以继续原来的程序处理。
指令寄存器	程序计数器将指令指针（程序存储器地址）推送到程序存储器，程序存储器将程序存储器的数据（16bits）推送到指令寄存器。 CSU3AF10 的指令是 16bits，包括 3 种信息：直接地址，立即数及控制信息。 直接地址（9bits）：数据存储器的地址。CPU 能利用此地址来对数据存储器进行操作。 立即数（8bits）：CPU 通过 ALU 利用此数据对工作寄存器进行操作。 控制信息：它记录着 ALU 的操作信息。
指令译码器	指令寄存器将控制信息推送到指令译码器以进行译码，然后译码器将译码后的信息发送到相关的寄存器。
算术逻辑单元	算术逻辑单元不仅能完成 8 位二进制的加，减，加 1，减 1 等算术计算，还能对 8 位变量进行逻辑的与，或，异或，循环移位，求补，清零等逻辑运算。
工作寄存器	工作寄存器用来缓存数据存储器的数据和立即数。
状态寄存器	当 CPU 利用 ALU 处理寄存器数据时，如下的状态会随着如下顺序变化：PD，TO，DC，C 及 Z。
文件选择寄存器	在 CSU3AF10 的指令集中，FSR 是用于间接数据处理（即实现间接寻址）。用户可以利用 FSR 来存放数据存储器中的某个寄存器地址，然后通过间接地址寄存器（IND）对这个寄存器进行处理。
程序存储器	CSU3AF10 内带 32K×16 位的 flash 作为程序存储器。由于指令的操作码（OPCODE）是 16bits，用户最多只能编程 32K 的指令。程序存储器的地址总线是 15bits，数据总线是 16bits。
数据存储器	CSU3AF10 内带 2K bytes 的 SRAM 作为数据存储器。此数据存储器的地址总线是 9 bits，数据总线是 8 bits。其中 SRAM 最后 48 个地址为堆栈寄存器，因此不能对 SRAM 的最后 48 个地址进行写操作。

特性（Property）：

R = 可读位

W = 可写位

U = 无效位

-n = 上电复位后的值

‘1’ = 位已设置

‘0’ = 位已清零

X = 不确定位

2.1.2 CPU 指令执行流程

CPU 核指令执行分为 4 个阶段，分别是取指、译码、读取、执行/写回，内核支持流水线操作，流水线形式如下

内核流水线

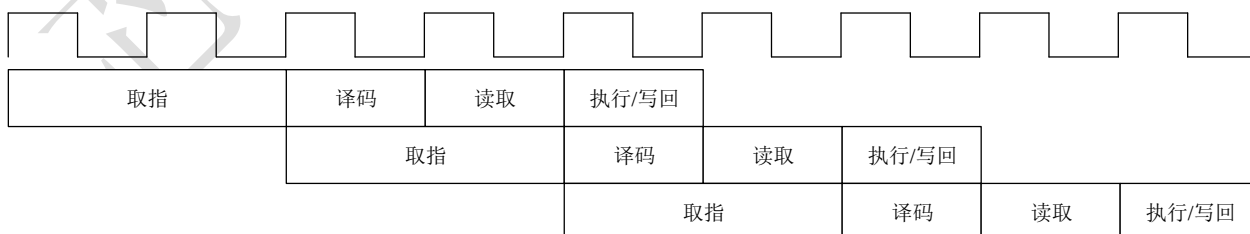


图 2-2 内核流水线

内核每 2 个时钟周期执行 1 条指令，因此指令周期为 CPUCLK 周期的 2 倍，即如果 CPUCLK 频率为 8MHz，则指令周期为 4MHz；如果 CPUCLK 频率为 24MHz，则指令周期为 12MHz。

有部分指令执行周期超过 2 个时钟周期，主要是跳转指令，如 CALL、GOTO、RETURN 等，需要 2 个指令周期才能执行一条指令，另外长寻址指令 MOVWFL、MOVWFL 也需要 2 个指令周期才能执行完成。还有 4 条指令是需要 3 个 CPUCLK 才能执行完成的，他们是 INCFSZ、DECFSZ、BTFSS、BTFSC。

2.1.3 存储器

2.1.3.1 程序存储器

程序存储器主要用于指令的存储，在 CSU3AF10 中，该程序存储器是 32K*16bit 的程序 FLASH，对于程序员来说，该存储器只读。系统的 reset 地址为 000H，中断入口地址为 004H，需要注意的是所有的中断共用同一个中断入口地址。

Area Name		Address	
Main Program Memory 32 K Words		0x0000	
		0x7FFF	
RESERVE		0x8000	
Data Flash		0xEFFF	
		0xF000	
RESERVE		0xF0FF	
		0xF100	
Option Bytes		0xF7FF	
		User Option Bytes 0.5 K Words	0xF800
		...	...
		0xF9FF	...
Program Option Bytes 0.5 K Words		0xFA00	
		...	...
Manufacture Parameters 1 K Words		0xFBFF	
		0xFC00	...
		0xFFFF	

图 2-3 程序存储器

2.1.3.2 数据存储器

数据存储器主要用于存储程序运行过程中的全局变量以及中间变量。该存储器分为三个部分。地址的 00H 至 0FH 是系统特殊功能寄存器，例如间接地址，间接地址指针，状态寄存器，工作寄存器，中断标志位，中断控制寄存器。地址 10H 至 7FH 对应外设特殊功能寄存器，例如 IO 端口，定时器。系统特殊功能寄存器和外设特殊功能寄存器是用寄存器实现，而通用数据存储器是 RAM 实现，可以读出也可以写入。

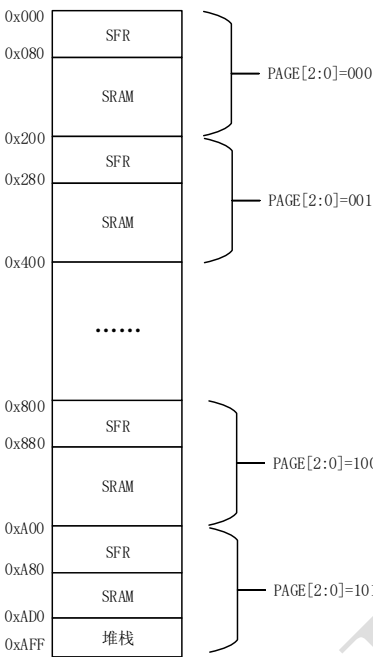


图 2-4 数据存储器地址分配图

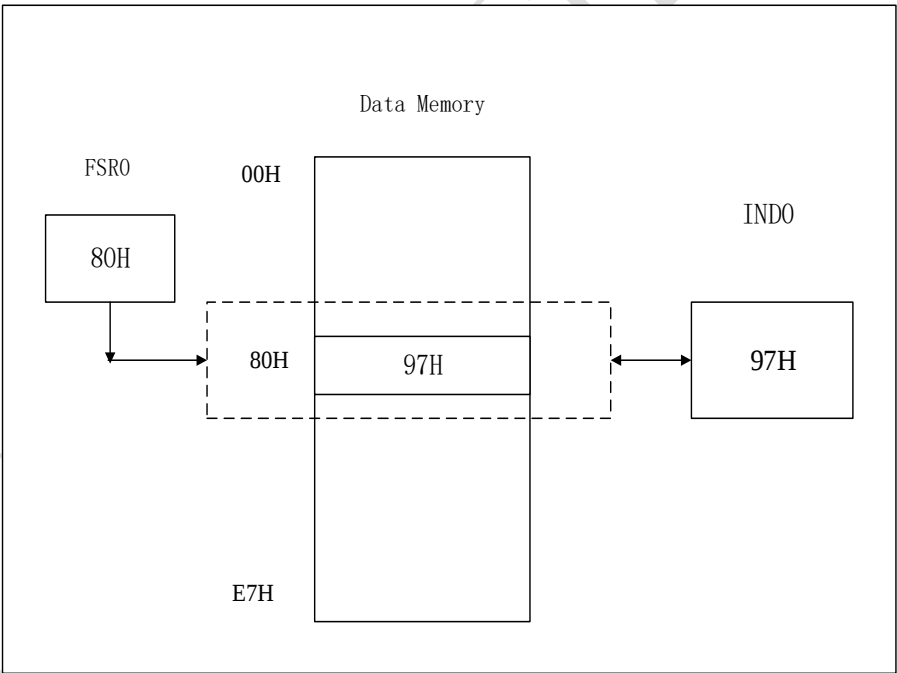


图 2-5 数据存储器地址分配图

表 2-2 数据存储器地址分配

数据存储器	起始地址	结束地址
系统特殊功能寄存器	0x00	0x0F
外设特殊功能寄存器 (PAGE0, PAGE1, PAGE2)	0x10	0x7F
通用数据存储器	0x80	0x1FF

(PAGE0, PAGE1, PAGE2, PAGE3, PAGE4)		
通用数据存储器 (PAGE5)	0x80	0xFF

2.1.4 数据存储器寻址

2.1.4.1 直接寻址

直接寻址即通过指令译码出的存储器地址，对该地址进行读写操作。直接寻址的地址范围为 000H~1FFH（2 条长寻址指令除外），即直接寻址 512 Byte，超过 512 Byte 的需要通过切换 PAGE 进行操作。

为了避免在访问 SRAM 时频繁切换 SRAM，我们增加了 2 条对数据存储器进行长寻址的指令，可以直接访问最大 4K Byte 空间的地址范围，这两条指令是 MOVFWL、MOVWFL，直接寻址地址范围 000H~FFFH。

2.1.4.2 间接寻址

通过 INDO/FSR0 和 IND1/FSR1 这两组寄存器可以对数据存储器以及特殊功能寄存器进行间接访问。当从间接地址寄存器 (IND0) 读入数据时，MCU 实际上是以 FSR0 中的值作为地址去访问数据存储器得到数据。当向间接寄存器 (IND0) 写入数据时 MCU 实际上是以 FSR0 中的值作为地址去访问数据存储器并将值存入该地址，FSR0/FSR1 最大 12 位，支持 8K 空间访问，**间接寻址不受 PAGE 影响**，其访问方式见下图。

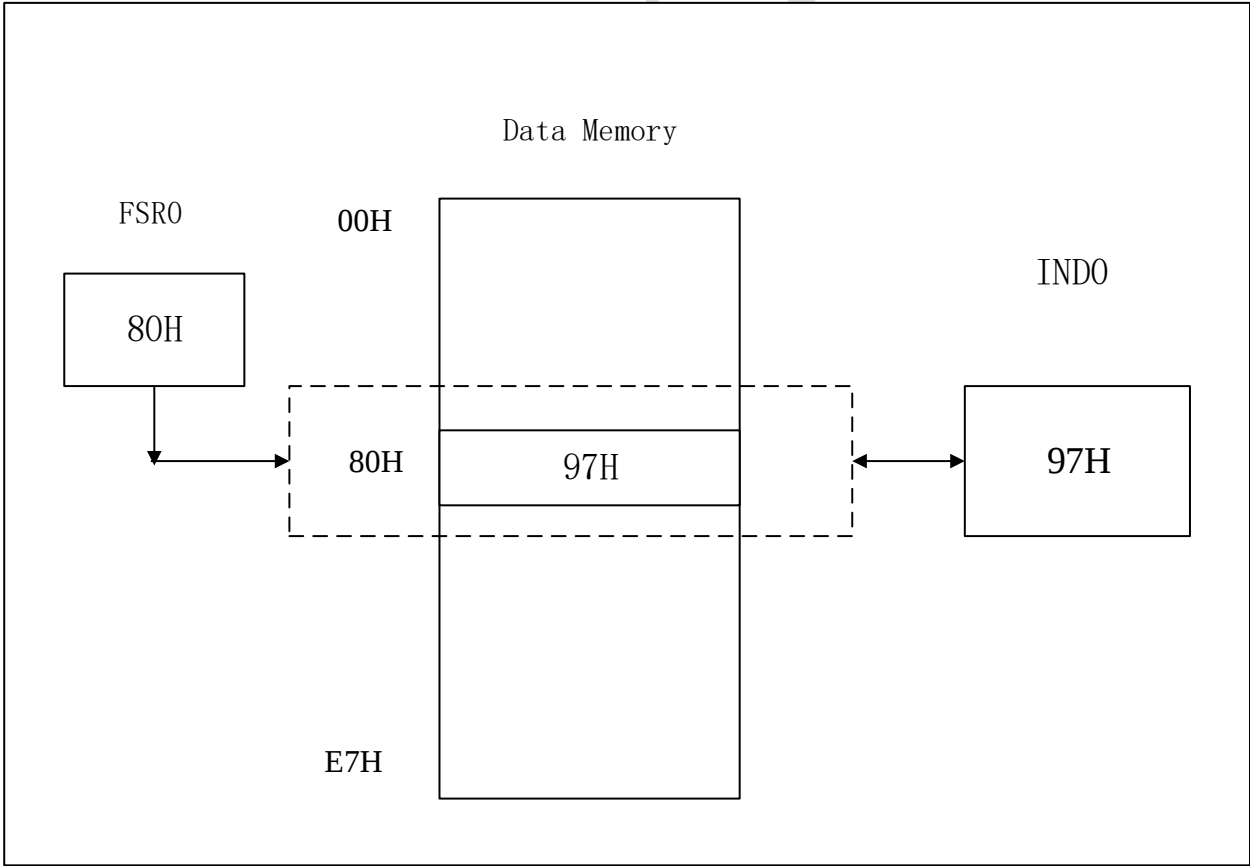


图 2-6 间接地址访问

2.1.5 PC 跳转

2.1.5.1 PC 跳转操作

通过对 PCLOADH、PCLOADL 进行赋值，可以进行程序跳转操作。每次改变 PC 时，必须先写 PCLOADH，再写 PCLOADL，写完 PCLOADL，PC 将跳转到 PCLOADH、PCLOADL 给定的值。

由于内核流水线存在，写完 PCLOADL 后，当前 PC 不会立即跳转，因此需要在写完 PCLOADL 后在加 1 条 NOP 指令，避免出现跑飞。

2.1.5.2 PC 读操作

当要获取当前 PC 值时，可以通过读取 PCLOADL 读取 PC 值低 8 位，同时，PC 高位将会被锁存在 PCLOADH 的读 BUFF 中，之后再读取 PCLOADH 即可获取完整的 PC 值，可以理解为 PCLOADH 与 PCLOADL 联动。对 PCLOADL，PCLOADH 进行读操作，读回的并不是向 PCLOADL 和 PCLOADH 写入的值，读写使用不同的 BUFF 进行数据缓存，因此读出数据与写入数据并无关联。

2.1.6 中断向量表

内核支持最多 24 个中断，所有中断共用中断入口地址 004H，不支持中断优先级。发生中断时，硬件自动跳转到中断处理函数，需要软件通过查询 INTFL/INTFM/INTFH 对应中断标志位来确定中断源，INTFL/INTFM/INTFH 只读，如需要清除中断标志位，我们需要在对应模块进行清除动作。系统中的映射向量表如下

表 2-3 中断向量映射表

序号	中断号	中断名	中断源	中断描述	唤醒 SLEEP 模式
0	SOF	SOF	内核	堆栈溢出中断（无中断使能）	否
1	INTF0	TMOIF	Timer0	定时器 0 溢出中断	是
2	INTF1	EOIF	GPIO	GPIO 管脚上的外部中断 0	是
3	INTF2	E1IF	GPIO	GPIO 管脚上的外部中断 1	是
4	INTF3	SRADIF	SAR ADC	SAR ADC 转换中断	否
5	INTF4	RESERVE	保留	保留	否
6	INTF5	TM2IF	Timer2	定时器 2 溢出中断	是
7	INTF6	TM3IF	Timer3	定时器 3 溢出中断	否
8	INTF7	URTOIF	UART0	UART0 中断	是
9	INTF8	RESERVE	保留	保留	否
10	INTF9	RESERVE	保留	保留	否
11	INTF10	TM4IF	Timer4	定时器 4 溢出中断	否
12	INTF11	CMPOIF	CMP0	比较器 0 中断	是
13	INTF12	CMP1IF	CMP1	比较器 1 中断	是
14	INTF13	RESERVE	保留	保留	否
15	INTF14	E2IF	GPIO	GPIO 管脚上的外部中断 2	是
16	INTF15	RESERVE	保留	保留	
17	INTF16	RESERVE	保留	保留	
18	INTF17	RESERVE	保留	保留	
19	INTF18	RESERVE	保留	保留	
20	INTF19	RESERVE	保留	保留	
21	INTF20	RESERVE	保留	保留	
22	INTF21	RESERVE	保留	保留	
23	INTF22	RESERVE	保留	保留	
24	INTF23	RESERVE	保留	保留	

内核中断标志位只读，如果要清除中断标志位，需要清除中断源对应模块中的中断标志位。

当芯片响应中断请求时，会把当前的 PC 值入栈保护，并将 PC 置为 004H，同时把总使能位 GIE 清 0，执行完中断服务程序后，用户用 RETFIE 指令返回到之前的主程序，同时硬件把 GIE 置 1。

CSU3AF10 中断系统不支持中断嵌套，没有中断优先级，因此禁止在中断服务程序中将 GIE 位置 1。

中断标志位都是硬件置 1，软件清 0。在对应的中断使能位没有置 1 的情况下，中断标志位也会硬件置 1。所以建议在中断函数里先判断中断使能位是否打开，然后再判断相应的中断标志位。否则有可能造成中断函数误执行。

2.1.7 堆栈

当前内核堆栈分为两种，PC 堆栈和数据堆栈，两种堆栈独立使用，互不影响。

PC 堆栈主要用于存储 PC 指针，即程序执行函数调用（CALL 指令）和进入中断时，需要将当前 PC 压入堆栈，方便函数返回和中断退出时继续执行。

数据堆栈用于对 WORK 寄存器和当前 STATUS 寄存器的 Z，C，DC 标志位以及 PAGE[3:0] 位进行压栈，方便必要时复原以上数据。数据堆栈在执行 PUSH 指令时进行压栈，在执行 POP 指令时出栈。

2.1.7.1 堆栈地址

堆栈数据存放在数据存储器（SRAM）的最后一段地址，堆栈数据占用的地址数量是配置的堆栈级数的 4 倍，即当配置 12 级堆栈时，堆栈将占用 48 个（12*4）地址，也就是说 SRAM 的最后 48 个地址是堆栈地址。堆栈地址读写操作由内核硬件进行控制，用户不能进行写操作。具体堆栈地址参考章节 [2.1.3.2 数据存储器](#) 的图 2-4

2.1.7.2 堆栈溢出中断

当出现 PC 堆栈和数据堆栈溢出时，STATUS 寄存器的 SOF 位将会硬件置位，堆栈溢出中断没有单独的中断使能位，只要总中断使能位 GIE 打开，当发生堆栈溢出事件时，就会进入中断函数。

堆栈溢出包括上溢出和下溢出，上溢出为压栈级数超过堆栈深度（12 级），下溢出为出栈级数大于压栈级数。

2.1.7.3 PUSH 和 POP 处理

当前内核有 12 级的 PUSH 和 POP 堆栈。有中断请求被响应后，程序跳转到 004h 执行子程序。响应中断之前必须保存 WORK 和 STATUS 中的标志位（只保存 C，DC，Z），在退出中断前，需要使用 POP 指令进行出栈。芯片提供 PUSH 和 POP 指令进行入栈保存和出栈恢复功能，从而避免中断结束后程序运行错误。子程序中也可以使用 PUSH 和 POP 指令对 WORK 和 STATUS（C，DC，Z）进行保存和恢复，PUSH、POP 必须成对出现。

2.1.8 非法指令复位

为了增强芯片抗干扰能力，内核会自动检测系统非法指令，如果检测到非法指令，自动产生 MCU 复位信号，将芯片复位。

非法指令包括以下几种情况：

- 本文档指令集和伪指令以外的其他指令码。
- 访问超出 SRAM 和堆栈地址以外的地址。
- 访问超出程序空间以外的地址

非法指令复位后芯片会延时一小段时间，PC 指针从 000H 地址开始运行，内核和所有外设寄存器被复位。

2.1.9 CPU 核寄存器映射

表 2-4 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值	
00h	IND0	以 FSR0 中内容作为地址的数据存储器中的数据								xxxxxxxx	
01h	IND1	以 FSR1 中内容作为地址的数据存储器中的数据								xxxxxxxx	
02h	FSR0	间接数据存储器的地址指针 0 低 8 位								00000000	
03h	FSR1	间接数据存储器的地址指针 1 低 8 位								00000000	
04h	STATUS			SOF	PD	TO	DC	C	Z	xx000000	
05h	WORK	工作寄存器								00000000	
06h	INTE	GIE								00u00000	
07h	BSR				PAGE[3:0]					uuuu0000	
08h	FSROH				间接数据存储器的地址指针 0 高 4 位					uuuu0000	
09h	FSR1H				间接数据存储器的地址指针 1 高 4 位					uuuu0000	
0Ah	PCLOADL	PCLOAD[7:0]									00000000
0Bh	PCLOADH			PCLOAD[13:8]							uu000000
0Ch	INTFL	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	INTF0	00000000	
0Dh	INTFM	INTF15	INTF14	INTF13	INTF12	INTF11	INTF10	INTF9	INTF8	00000000	
0Eh	INTFH	INTF23	INTF22	INTF21	INTF20	INTF19	INTF18	INTF17	INTF16	00000000	

2.1.10 CPU 核寄存器描述

2.1.10.1 IND0 寄存器（地址为 00h）

表 2-5

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IND0	IND0[7:0]							

表 2-6

位地址	标识符	功能
7: 0	IND0[7:0]	以 FSR0 中内容作为地址的数据存储器中的数据

2.1.10.2 IND1 寄存器（地址为 01h）

表 2-7

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
IND1	IND1[7:0]							

表 2-8

位地址	标识符	功能
7: 0	IND1[7:0]	以 FSR1 中内容作为地址的数据存储器中的数据

2.1.10.3 FSR0 寄存器（地址为 02h）

表 2-9

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FSR0	FSR0[7:0]							

表 2-10

位地址	标识符	功能
7: 0	FSR0[7:0]	间接数据存储器的地址指针 0

2.1.10.4 FSR1 寄存器（地址为 03h）

表 2-11

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
FSR1	FSR1[7:0]							

表 2-12

位地址	标识符	功能
7: 0	FSR1[7:0]	间接数据存储器的地址指针 1

2.1.10.5 STATUS 状态寄存器（地址为 04h）

状态寄存器包含 ALU 的算术状态及复位状态。状态寄存器与其它寄存器一样，可以作为任何指令的目标寄存器。如果状态寄存器是某条指令的目标寄存器，而且影响到 Z，DC 或 C 位，那么对这三个位的写是无效的。这些位是由器件逻辑进行置位或清零。

表 2-13

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0
STATUS			SOF	PD	T0	DC	C	Z

表 2-14

位地址	标识符	功能
7: 6	RESERVE	保留
5	SOF	堆栈溢出中断标志位，该位硬件置高，软件写 1 清零。 0: 未出现中断溢出标志位 1: 出现中断溢出标志位 此处堆栈溢出包括 PC 堆栈和数据堆栈。
4	PD	掉电标志位，sleep 后置此位 1: 执行 SLEEP 指令后 0: 上电复位后或硬件复位或 CLRWDI 指令之后
3	T0	看门狗定时溢出标志，看门狗定时溢出置位此位 1: 看门狗定时溢出发生 0: 上电复位后或硬件复位或 CLRWDI 指令后或 SLEEP 指令后
2	DC	半字节进位标志/借位标志 用于进位时 1: 结果的第 4 位出现进位溢出

		0: 结果的第 4 位未出现进位溢出 用于借位时, 极性相反 0: 结果的第 4 位出现借位溢出 1: 结果的第 4 位未出现借位溢出
1	C	进位标志/借位标志 用于进位时 1: 结果的最高位 (MSB) 出现进位溢出 0: 结果的最高位 (MSB) 不出现进位溢出 用于借位时, 极性相反 0: 结果的最高位 (MSB) 出现借位溢出 1: 结果的最高位 (MSB) 不出现借位溢出
0	Z	零标志 1: 算术运算或逻辑操作结果为 0 0: 算术运算或逻辑操作结果不为 0

2.1.10.6 WORK 寄存器 (地址为 05h)

表 2-15

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
WORK	WORK[7:0]							

表 2-16

位地址	标识符	功能
7: 0	WORK[7:0]	工作寄存器 做为访问 SFR 和 SRAM 的临时数据寄存器

2.1.10.7 INTE 寄存器 (地址为 06h)

表 2-17

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
INTE	GIE							

表 2-18

位地址	标识符	功能
7	GIE	全局中断使能位, 高电平有效 1: 全局中断使能 0: 全局中断关闭 当需要使用中断函数响应中断时, 必须使能全局中断。进入中断后 GIE 硬件自动拉低, 中断函数内部对 GIE 进行写操作无效, 退出中断函数时应将自动将 GIE 拉高。
6:0	RESERVE	保留

2.1.10.8 BSR 寄存器（地址为 07h）

表 2-19

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
BSR							PAGE [2:0]	

表 2-20

位地址	标识符	功能
7:3	RESERVE	保留
2:0	PAGE[2:0]	PAGE 页选择 000 : 直接寻址和间接寻址访问 PAGE0 001 : 直接寻址和间接寻址访问 PAGE1 010 : 直接寻址和间接寻址访问 PAGE2 011 : 直接寻址和间接寻址访问 PAGE3 100 : 直接寻址和间接寻址访问 PAGE4 101 : 直接寻址和间接寻址访问 PAGE5 110 : 直接寻址和间接寻址访问 PAGE6 111 : 保留

2.1.10.9 FSR0H 寄存器（地址为 08h）

表 2-21

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
FSR0H					FSR0 [11:8]			

表 2-22

位地址	标识符	功能
7: 4	RESERVE	保留
3: 0	FSR0 [11:8]	间接数据存储器地址指针 0 高 4 位

2.1.10.10 FSR1H 寄存器（地址为 09h）

表 2-23

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
FSR1H					FSR1 [11:8]			

表 2-24

位地址	标识符	功能
7: 4	RESERVE	保留
3: 0	FSR1 [11:8]	间接数据存储器地址指针 1 高 4 位

2.1.10.11 PLOADL 寄存器（地址为 0Ah）

表 2-25

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCLOADL	PCLOAD[7:0]							

表 2-26

位地址	标识符	功能
7 : 0	PCLOAD[7:0]	PC 值低 8 位。

2.1.10.12 PLOADH 寄存器（地址为 0Bh）

表 2-27

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PLOADH	PCLOAD[13:8]							

表 2-28

位地址	标识符	功能
7 : 6	RESERVE	保留
5 : 0	PCLOAD[13:8]	PC 值高 6 位。

2.1.10.13 INTFL 寄存器（地址为 0Ch）

表 2-29

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
INTFL	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	INTF0

表 2-30

位地址	标识符	功能
7	INTF7	INTF7 中断标志位，高电平有效，只读 具体中断参考中断向量表
6	INTF6	INTF6 中断标志位，高电平有效，只读 具体中断参考中断向量表
5	INTF5	INTF5 中断标志位，高电平有效，只读 具体中断参考中断向量表
4	INTF4	INTF4 中断标志位，高电平有效，只读 具体中断参考中断向量表
3	INTF3	INTF3 中断标志位，高电平有效，只读 具体中断参考中断向量表
2	INTF2	INTF2 中断标志位，高电平有效，只读 具体中断参考中断向量表
1	INTF1	INTF1 中断标志位，高电平有效，只读 具体中断参考中断向量表

0	INTF0	INTF0 中断标志位，高电平有效，只读 具体中断参考中断向量表
---	-------	-------------------------------------

2.1.10.14 INTFM 寄存器（地址为 0Dh）

表 2-31

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
INTFM	INTF15	INTF14	INTF13	INTF12	INTF11	INTF10	INTF9	INTF8

表 2-32

位地址	标识符	功能
7	INTF15	INTF15 中断标志位，高电平有效，只读 具体中断参考中断向量表
6	INTF14	INTF14 中断标志位，高电平有效，只读 具体中断参考中断向量表
5	INTF13	INTF13 中断标志位，高电平有效，只读 具体中断参考中断向量表
4	INTF12	INTF12 中断标志位，高电平有效，只读 具体中断参考中断向量表
3	INTF11	INTF11 中断标志位，高电平有效，只读 具体中断参考中断向量表
2	INTF10	INTF10 中断标志位，高电平有效，只读 具体中断参考中断向量表
1	INTF9	INTF9 中断标志位，高电平有效，只读 具体中断参考中断向量表
0	INTF8	INTF8 中断标志位，高电平有效，只读 具体中断参考中断向量表

2.1.10.15 INTFH 寄存器（地址为 0Eh）

表 2-33

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
INTFH	INTF23	INTF22	INTF21	INTF20	INTF19	INTF18	INTF17	INTF16

表 2-34

位地址	标识符	功能
7	INTF23	INTF23 中断标志位，高电平有效，只读 具体中断参考中断向量表
6	INTF22	INTF22 中断标志位，高电平有效，只读 具体中断参考中断向量表
5	INTF21	INTF21 中断标志位，高电平有效，只读 具体中断参考中断向量表
4	INTF20	INTF20 中断标志位，高电平有效，只读 具体中断参考中断向量表
3	INTF19	INTF19 中断标志位，高电平有效，只读 具体中断参考中断向量表

2	INTF18	INTF18 中断标志位，高电平有效，只读 具体中断参考中断向量表
1	INTF17	INTF17 中断标志位，高电平有效，只读 具体中断参考中断向量表
0	INTF16	INTF16 中断标志位，高电平有效，只读 具体中断参考中断向量表

芯海科技CHIPSEA

2.2 外设地址映射

系统给每个外设分配了一个固定的地址段，地址段包括基址和长度，外设地址分配如下

表 2-35 外设地址映射表

模块	符号	绝对地址	按 PAGE 划分地址		描述
			PAGE	地址	
CORE	-	0x00~0x0F	不分 PAGE	0x00~0x0F	CS-R8V2 内核寄存器
CLKMU	CLKMU_BA	0x10~0x1F	PAGE0	0x10~0x1F	时钟管理单元
SYSCFG	SYS_BA	0x20~0x2F	PAGE0	0x20~0x2F	系统配置模块
RSTMU	RSTMU_BA	0x30~0x37	PAGE0	0x30~0x37	复位管理单元
EXTI	EXTI_BA	0x38~0x3F	PAGE0	0x38~0x3F	外部中断模块
WDT	WDT_BA	0x40~0x43	PAGE0	0x40~0x43	看门狗模块
WWDT	WWDT_BA	0x44~0x47	PAGE0	0x44~0x47	窗看门狗模块
Timer0	TM0_BA	0x48~0x4F	PAGE0	0x48~0x4F	定时器 0
GPIO0	GPIO0_BA	0x50~0x5F	PAGE0	0x50~0x5F	GPIO 控制器
SAR_ADC	SRADC_BA	0x60~0x67	PAGE0	0x60~0x67	SAR ADC
FMC	FMC_BA	0x68~0x6F	PAGE0	0x68~0x6F	Flash 存储器控制器
Timer2	TM2_BA	0x70~0x77	PAGE0	0x70~0x77	定时器 2
RESERVE	RESERVE	0x78~0x7F	PAGE0	0x78~0x7F	保留
Timer3	TM3_BA	0x210~0x21F	PAGE1	0x10~0x1F	定时器 3
UART0	UART0_BA	0x220~0x227	PAGE1	0x20~0x27	串行通信模块 0
RESERVE	RESERVE	0x228~0x22F	PAGE1	0x28~0x2F	保留
RESERVE	RESERVE	0x230~0x23F	PAGE1	0x30~0x3F	保留
RESERVE	RESERVE	0x240~0x247	PAGE1	0x40~0x47	保留
Timer4	TM4_BA	0x248~0x24F	PAGE1	0x48~0x4F	定时器 4

RESERVE	RESERVE	0x250~0x257	PAGE1	0x50~0x57	保留
CMP0	CMP0_BA	0x258~0x25B	PAGE1	0x58~0x5B	比较器 0
CMP1	CMP1_BA	0x25C~0x25F	PAGE1	0x5C~0x5F	比较器 1
RESERVE	RESERVE	0x260~0x26F	PAGE1	0x60~0x6F	保留
RESERVE	RESERVE	0x270~0x277	PAGE1	0x70~0x77	保留
TEST	TEST_BA	0x277~0x27F	PAGE1	0x77~0x7F	测试模块
RESERVE	RESERVE	0x410~0x41F	PAGE2	0x10~0x1F	保留
RESERVE	RESERVE	0x420~0x423	PAGE2	0x20~0x23	保留
RESERVE	RESERVE	0x424~0x427	PAGE2	0x24~0x27	保留
RESERVE	RESERVE	0x428~0x42F	PAGE2	0x28~0x2F	保留
RESERVE	RESERVE	0x430~0x437	PAGE2	0x30~0x37	保留
RESERVE	RESERVE	0x438~0x43F	PAGE2	0x38~0x3F	保留
RESERVE	RESERVE	0x440~0x44F	PAGE2	0x40~0x4F	保留
RESERVE	RESERVE	0x450~0x46F	PAGE2	0x50~0x6F	保留
RESERVE	RESERVE	0x470~0x477	PAGE2	0x70~0x77	保留
RESERVE	RESERVE	0x478~0x47F	PAGE2	0x78~0x7F	保留
RESERVE	RESERVE	0x610~0x67F	PAGE3	0x10~0x7F	保留
PROG	PROG_BA	0x810~0x83F	PAGE4	0x10~0x3F	烧录模块寄存器

2.3 时钟管理单元（CLKMU）

2.3.1 概述

芯片的时钟源包括内置 24MHz 的 RC 振荡时钟（HIRC）、外置高速晶振、内置低速 10 KHz 的 WDT 时钟。内置 24MHz RC 振荡时钟和外部高速晶振可以作为系统时钟源 F_{SYS} 。 F_{CPU} 是 CPU 时钟频率。

普通模式（高速时钟）： $F_{CPU}=F_{SYS}/N$ ， $N=1、2、4$

2.3.2 时钟框图

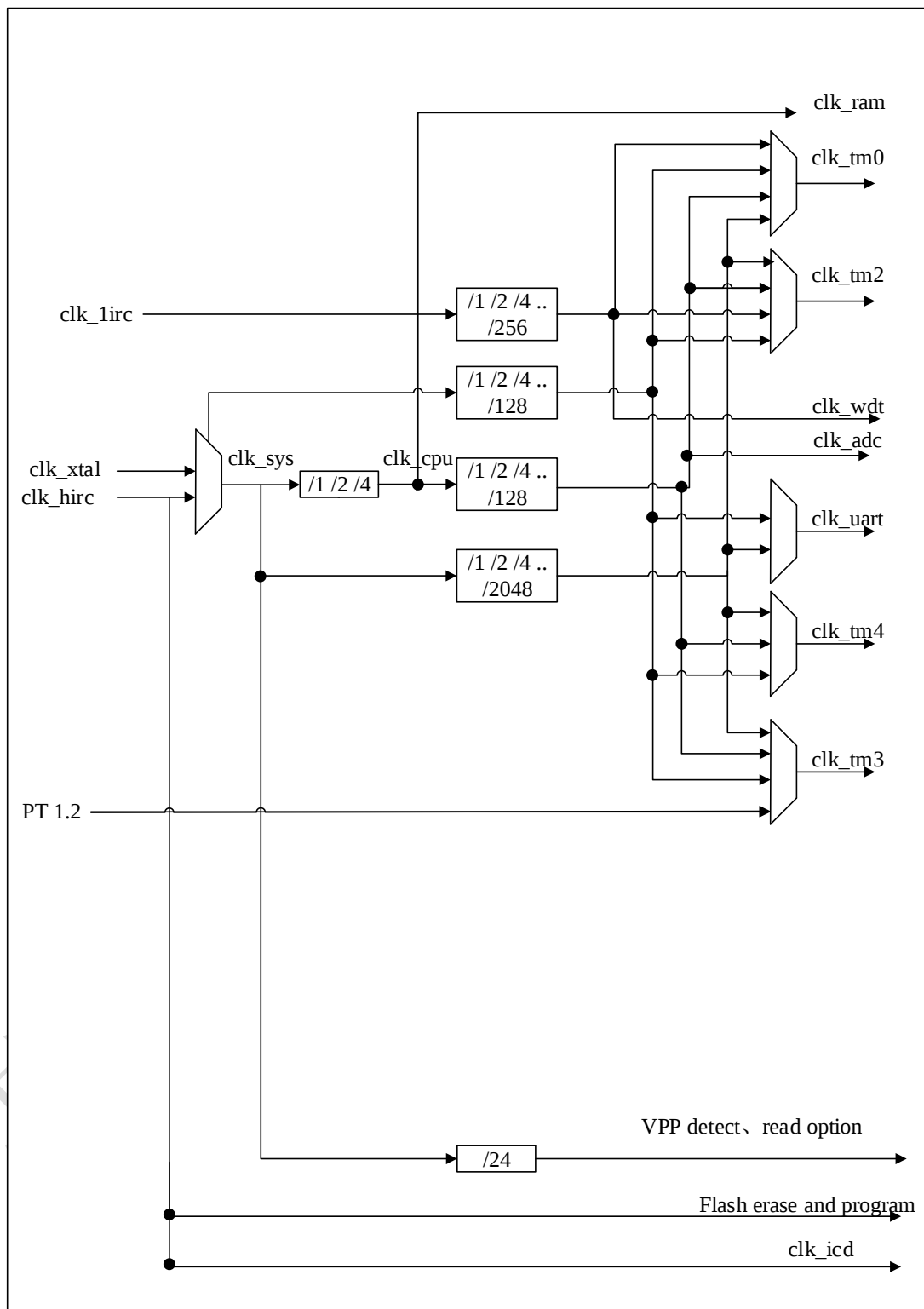


图 2-7 时钟结构

2.3.3 基本配置

CPUCLK 分频在 Flash 0xF800 地址中配置，无法在程序运行时改变。

为了保证芯片稳定性，外设时钟源切换和分频选择，软件必须先关闭模块使能。

PT14 和 PT15 口可以配置为外部高速晶振输入输出接口，默认情况下，PT14、PT15 为通用 GPIO 功能，如需要使用外部高速晶振，则需将 SYSCFG 模块 CFGR 寄存器的 PT14FUNC 和 PT15FUNC 都配置成 01。

2.3.4 功能描述

2.3.4.1 系统时钟 SYSCLK

内置 24MHz RC 振荡时钟和外部高速晶振可以做为系统时钟源 SYSCLK。为保证系统时钟的稳定性，避免死机，外部晶振使能 CST_XT 和内部 24MHz 振荡器使能 CST_IN 禁止同时失效，硬件保证芯片始终存在系统时钟。

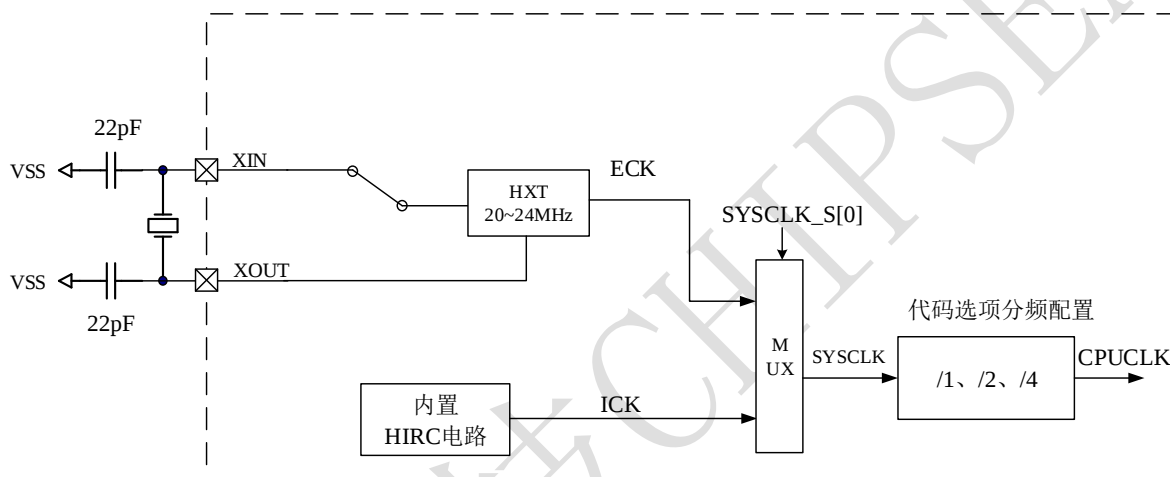


图 2-8 CSU3AF10 振荡器状态框图

2.3.4.2 内部低速 wdt 时钟

内部低速 wdt 时钟（10KHz），可以通过 WDT_CFG 配置，当 WDT_CFG 为 0 时，内部 10K 低速振荡器固定打开，软件无法关闭。当 WDT_CFG 为 1 时，通过寄存器 CST_WDT 使能开关。内部 wdt 时钟不能做为系统主时钟，只能做为 WDT 和部分外设的时钟使用。

2.3.5 寄存器映射

表 2-36 CLKMU 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
基址 PAGE0, CLKMU_BA = 0x10										
00H	MCK	CST_XT	CST_IN	CST_WDT	XT_GM[2:0]		SYSCLK_S[1:0]		01111100	
01H	TCTRIM						tc_trim[2:0]		00000100	
02H	CLKSEL0	T3CKS[1:0]		T2CKS[1:0]				T0CKS[1:0]		00000000
03H	CLKSEL1					URTCKS[1:0]				00000000
04H	CLKSEL2							T4CKS[1:0]		00000000
05H	CLKDIV0						T0DIV[2:0]		00000000	
06H	CLKDIV1		T3DIV[2:0]				T2DIV[2:0]		00000000	
07H	CLKDIV2		URTDIV[2:0]						00000000	
08H	CLKDIV3		T4DIV[2:0]					SRADCKS[1:0]		00000000

2.3.6 寄存器描述

2.3.6.1 MCK 寄存器（地址 00h）

表 2-37

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W -0	R/W -1	R/W -1	R/W -1	R/W -1	R/W -1	R/W -0	R/W -0
MCK	CST_XT	CST_IN	CST_WDT	XT_GM[2:0]			SYSCLK_S[1:0]	

表 2-38

位地址	标识符	功能
7	CST_XT	外部高速晶振使能 0: 关闭 1: 使能
6	CST_IN	内部高速 RC 振荡器使能 0: 关闭 1: 使能
5	CST_WDT	内部 10K 低速振荡器启动开关，当 WDT_CFG 配置为内部 10K 低速振荡器固定打开时，该位无效。 0: 内部 10K 低速振荡器关闭 1: 内部 10K 低速振荡器打开
4:2	XT_GM[2:0]	晶振电路偏置电流（控制跨导）选择 000: 64uA 001: 128 uA 010: 256 uA 011: 512 uA 1xx: 保留
1:0	SYSCLK_S[1:0]	系统时钟选择 00: 内部高速 RC 振荡器作为系统时钟 01: 外部高速晶振作为系统时钟 1x: 保留

2.3.6.2 TCTRIM 寄存器（地址 01h）

表 2-39

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U -0	U -0	U -0	U -0	U -0	R/W -1	R/W -0	R/W -0
TCTRIM						TC_TRIM[2:0]		

表 2-40

位地址	标识符	功能
7:3	RESERVE	保留
2:0	TC_TRIM[2:0]	内部高速时钟温度 TRIM 值

2.3.6.3 CLKSEL0 寄存器（地址 02h）

表 2-41

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W -0	R/W -0	R/W -0	R/W -0	U -0	U -0	R/W -0	R/W -0
CLKSEL0	T3CKS[1:0]		T2CKS[1:0]				T0CKS[1:0]	

表 2-42

位地址	标识符	功能
7:6	T3CKS[1:0]	定时器 3 时钟源选择 00: CPUCLK 01: SYSCLK 10: 外部高速时钟 ECK 11: PT1.2 外部事件输入
5:4	T2CKS[1:0]	定时器 2 时钟源选择 00: CPUCLK 01: SYSCLK 10: 外部高速时钟 ECK 11: 10 KHz 时钟
3:2	RESERVE	保留
1:0	T0CKS[1:0]	定时器 0 时钟源选择 00: CPUCLK 01: SYSCLK 10: 外部高速时钟 ECK 11: 10 KHz 时钟

2.3.6.4 CLKSEL1 寄存器（地址 03h）

表 2-43

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U -0	U -0	U -0	U -0	R/W -0	R/W -0	U -0	U -0
CLKSEL1					URTCKS[1:0]			

表 2-44

位地址	标识符	功能
7:4	RESERVE	
3:2	URTCKS[1:0]	串口时钟源选择 00: SYSCLK 01: 外部时钟 ECK 1x: 保留
1:0	RESERVE	保留

2.3.6.5 CLKSEL2 寄存器（地址 04h）

表 2-45

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-----	------	------	------	------	------	------	------	------

特性	U -0	U -0	U -0	U -0	U -0	U -0	R/W -0	R/W -0
CLKSEL2							T4CKS[1:0]	

表 2-46

位地址	标识符	功能
7:2	RESERVE	保留
1:0	T4CKS[1:0]	定时器 4 时钟源选择 00: CPUCLK 01: SYSCLK 10: 外部高速时钟 ECK 11: 保留

2.3.6.6 CLKDIV0 寄存器（地址 05h）

表 2-47

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U -0	U -0	U -0	U -0	U -0	R/W -0	R/W -0	R/W -0
CLKDIV0						T0DIV[2:0]		

表 2-48

位地址	标识符	功能																		
7:3	RESERVE	保留																		
2:0	T0DIV[2:0]	定时器 0 时钟分频选择，其中 CKT0 为通过 T0CKS[1:0]选择的时钟源。																		
		<table><tr><td>T0DIV [2:0]</td><td>TM0CLK</td></tr><tr><td>000</td><td>CKT0</td></tr><tr><td>001</td><td>CKT0/2</td></tr><tr><td>010</td><td>CKT0/4</td></tr><tr><td>011</td><td>CKT0/8</td></tr><tr><td>100</td><td>CKT0/16</td></tr><tr><td>101</td><td>CKT0/32</td></tr><tr><td>110</td><td>CKT0/64</td></tr><tr><td>111</td><td>CKT0/128</td></tr></table>	T0DIV [2:0]	TM0CLK	000	CKT0	001	CKT0/2	010	CKT0/4	011	CKT0/8	100	CKT0/16	101	CKT0/32	110	CKT0/64	111	CKT0/128
		T0DIV [2:0]	TM0CLK																	
		000	CKT0																	
		001	CKT0/2																	
		010	CKT0/4																	
		011	CKT0/8																	
		100	CKT0/16																	
		101	CKT0/32																	
		110	CKT0/64																	
111	CKT0/128																			

2.3.6.7 CLKDIV1 寄存器（地址 06h）

表 2-49

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U -0	R/W -0	R/W -0	R/W -0	U -0	R/W -0	R/W -0	R/W -0
CLKDIV1		T3DIV[2:0]				T2DIV[2:0]		

表 2-50

位地址	标识符	功能	
7	RESERVE	保留	
6:4	T3DIV[2:0]	定时器 3 时钟分频选择，其中 CKT3 为通过 T3CKS[1:0]选择的时钟源。	
		T3DIV [2:0]	TM3CLK
		000	CKT3

		001	CKT3/2	
		010	CKT3/4	
		011	CKT3/8	
		100	CKT3/16	
		101	CKT3/32	
		110	CKT3/64	
		111	CKT3/128	
3	RESERVE	保留		
2:0	T2DIV[2:0]	定时器 2 时钟分频选择，其中 CKT2 为通过 T2CKS[1:0]选择的时钟源。		
		T2DIV [2:0]	TM2CLK	
		000	CKT2	
		001	CKT2/2	
		010	CKT2/4	
		011	CKT2/8	
		100	CKT2/16	
		101	CKT2/32	
		110	CKT2/64	
		111	CKT2/128	

2.3.6.8 CLKDIV2 寄存器（地址 07h）

表 2-51

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U -0	R/W -0	R/W -0	R/W -0	U -0	U -0	U -0	U -0
CLKDIV2		URTDIV[2:0]						

表 2-52

位地址	标识符	功能	
7	RESERVE	保留	
6:4	URTDIV[2:0]	UART 时钟分频选择	
		URTDIV[2:0]	UART_CLK
		000	/2, 二分频
		001	/4, 4 分频
		010	/8, 8 分频
		011	/16, 16 分频
		100	/32, 32 分频
		其他	/64, 64 分频
3:0	RESERVE	保留	

2.3.6.9 CLKDIV3 寄存器（地址 08h）

表 2-53

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U -0	R/W -0	R/W -0	R/W -0	U -0	U -0	R/W -0	R/W -0
CLKDIV3		T4DIV[2:0]					SRADCKS[1:0]	

表 2-54

位地址	标识符	功能	
7	RESERVE	保留	
6:4	T4DIV[2:0]	定时器 4 时钟分频选择，其中 CKT4 为通过 T4CKS[1:0]选择的时钟源。	
		T4DIV [2:0]	TM4CLK
		000	CKT4
		001	CKT4/2
		010	CKT4/4
		011	CKT4/8
		100	CKT4/16
		101	CKT4/32
		110	CKT4/64
111	CKT4/128		
3:2	RESERVE	保留	
1:0	SRADCKS[1:0]	ADC 时钟频率选择信号	
		SRADCKS[1:0]	ADC 时钟频率选择
		00	CPUCLK/4
		01	CPUCLK/4
		10	CPUCLK/8
		11	CPUCLK/16

2.4 复位管理单元（RSTMU）

2.4.1 概述

CSU3AF10 有以下方式复位：

- 上电复位
-  硬件复位
- WDT 复位（正常操作）
- WDT 复位（从 Sleep 模式）
- 低电压复位（LVR）
- WWDT 复位
- 非法指令复位
- EMC 复位

当上电复位、 硬件复位和 EMC 复位发生，程序会在 30ms 后重新开始运行，并且重读代码选项的配置

上述任意一种复位发生时，所有系统寄存器恢复默认状态（WDT 复位时 TO、PD 标志位除外），程序停止运行，同时程序计数器 PC 清零。复位结束后，系统地址从 000H 重新开始。各种复位情况下的 TO，PD 标志位如下表所示。

表 2-55 复位信号和状态寄存器关系

条件	TO	PD	WWDTF	EMCF	ILOPF
上电复位	0	0	0	0	0
 硬件复位（正常操作）	0	0	0	0	0
WDT 复位（正常操作）	1	不变	不变	不变	不变
WDT 复位（从 Sleep 模式）	1	1	不变	不变	不变
低电压复位	不变	不变	不变	不变	不变
WWDT 复位	不变	不变	1	不变	不变
非法指令复位	不变	不变	不变	不变	1
EMC 复位	不变	不变	0	1	0

2.4.2 功能描述

复位电路原理图如下所示

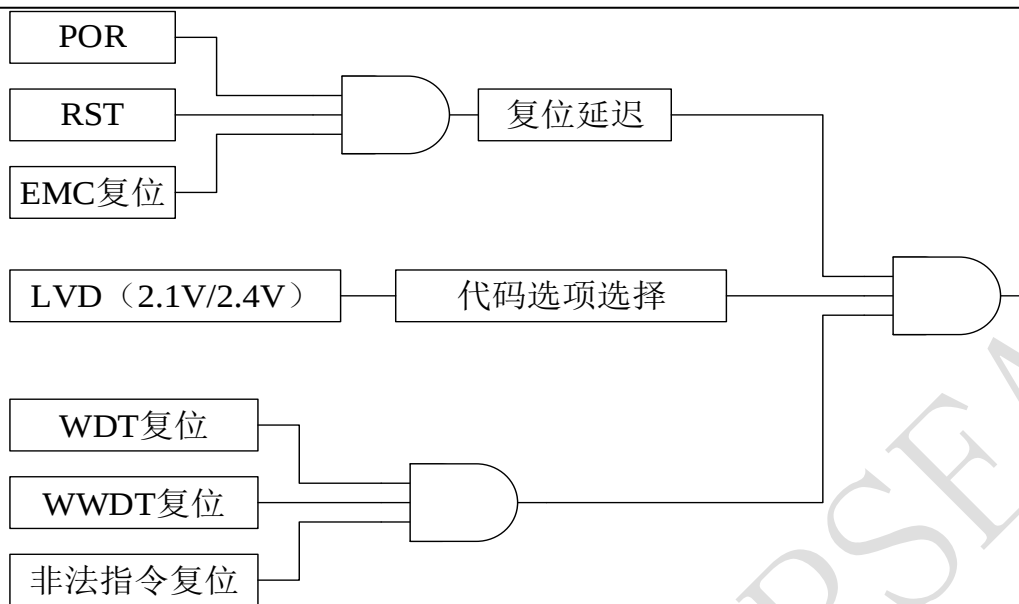


图 2-9 复位电路原理图

2.4.2.1 上电复位

系统上电电压呈现逐渐上升的曲线形式，需要一定时间才能达到正常的工作电压(对于不同的指令周期所需工作电压是不同的，指令周期越快相应所需的工作电压就越高，见 5.2 直流特性)。

用户应在上电复位后，预留一定的时间等待系统稳定。用户在终端使用过程中，应注意考虑主机对上电复位的要求。

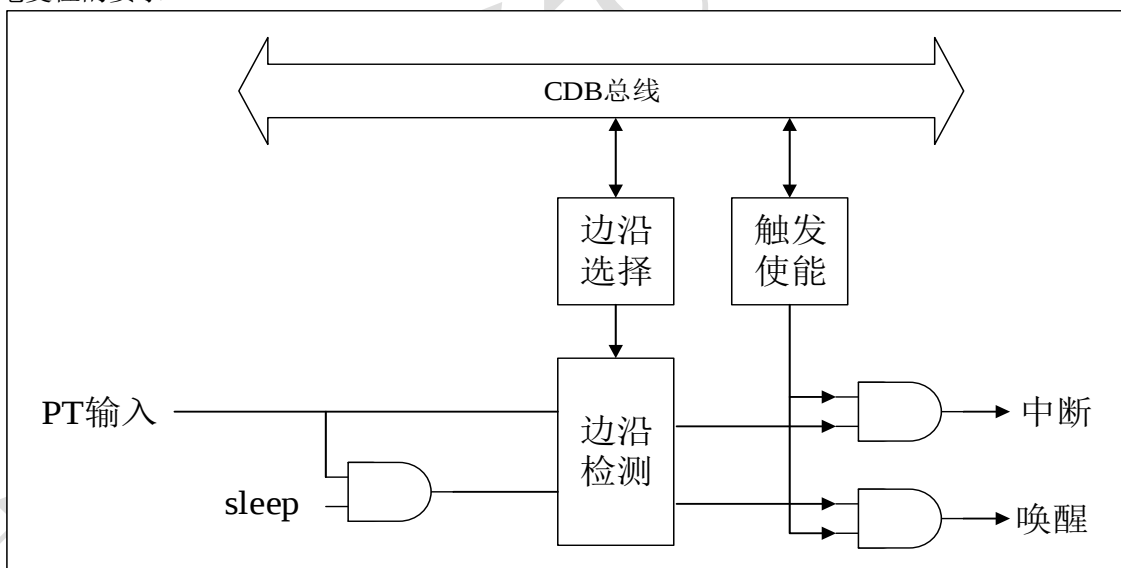


图 2-10 上电复位电路示例及上电过程

表 2-56

参数	最小值	典型值	最大值
VPOR	2.0	2.05V	2.1V
VLVR	1.9	1.93V	1.96V
twvs (测试条件: VDD=5V, T=25℃)	21ms	30ms	41.6ms

VPOR: 上电复位

VLVR: 掉电复位

T_{WVS} : 等待电压稳定时间

2.4.2.2 看门狗复位

看门狗复位是一种系统保护设置。在正常状态下，程序将看门狗定时器清零。如出错，系统处于未知状态，此时利用看门狗复位。看门狗复位后，系统重新进入正常状态。

看门狗复位可以唤醒 SLEEP 模式和 HALT 模式，芯片复位，系统重新进入正常状态。

2.4.2.3 窗看门狗复位

窗看门狗主要用来检测由外部接口或者无法预测的逻辑错误导致的软件出错。软件出错时导致应用程序跑飞，WWDT 可以产生复位来复位 MCU。

WWDT 在 SLEEP 模式下将停止计数，因此无法唤醒 SLEEP 模式。

WWDT 在 HALT 模式下的行为可以通过代码选项进行配置，可以配置为在 HALT 模式继续进行计数或者在 HALT 模式不进行计数。如果在 HALT 模式下继续进行计数，则可以通过复位唤醒芯片。

2.4.2.4 掉电复位

掉电复位是针对外部引起的电源电压跌落情况，例如受到干扰或者负载变化。电源掉电可能会引起系统工作状态不正常或者程序执行错误。

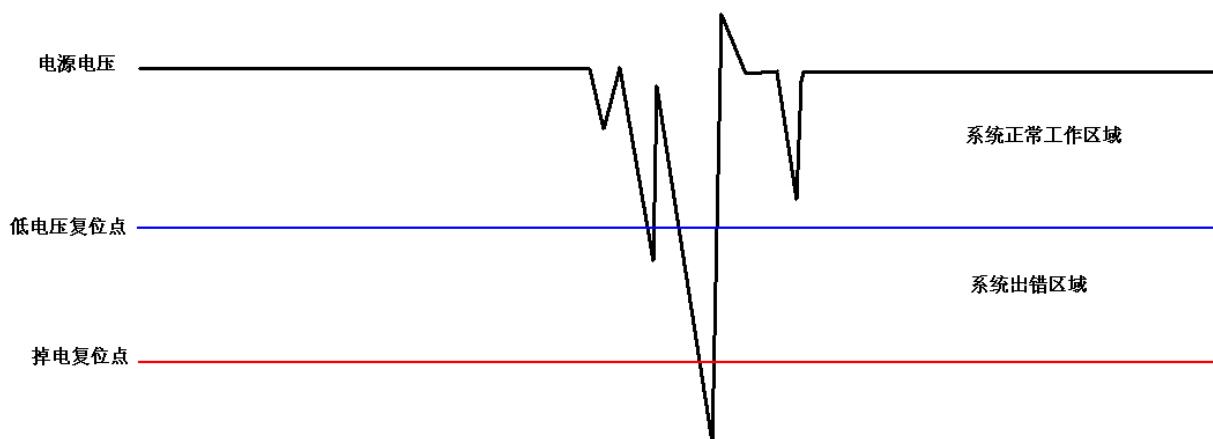


图 2-11 系统掉电复位示意图

电源电压跌落可能会导致芯片进入系统死区。系统死区，即电源电压不能满足系统的最小工作电压要求。系统掉电复位示意图如上图所示。芯片的低电压复位点可以通过软件设置成 2.1V 或 2.4V。

为避免进入系统死区，建议利用低电压复位（LVR）功能，尤其是指令周期是高速应用的情况。掉电复位性能的改善可以通过如下几点实现：

- 1) 低电压复位（LVR）
- 2) 看门狗复位
- 3) 降低系统指令周期
- 4) 采用外部复位电路（稳压二极管复位电路；电压偏移复位电路；外部 IC 复位）

2.4.2.5 低电压检测（LVD）

低电压检测（LVD）功能提供内部 2.1V、2.4V 低电压检测。

2.4.2.6 外部硬件复位

外部复位由代码选项 **RESET_PIN** 控制。通过设置该代码选项，可使能外部硬件复位功能。外部硬件复位引脚为施密特触发结构，低电平有效。硬件复位引脚为高电平时，系统正常工作；硬件复位引脚为低电平时，系统复位。

在芯片代码选项使能外部硬件复位功能后，需要注意的是：在系统上电完成后，外部复位需要输入高电平，否则，系统会一直复位，直到外部硬件复位结束。

外部硬件复位可以在上电过程中使用系统复位。良好的外部复位电路可以保护系统避免进入系统死区。

2.4.2.7 非法指令复位

为了增强芯片抗干扰能力，芯片会自动检测系统非法指令，如果检测到非法指令，自动产生 MCU 复位信号，将芯片复位。可以通过代码选项关闭非法指令使能位。

非法指令包括以下几种情况：

- 1、本文档指令集和伪指令以外的其他指令码。
- 2、访问超出 SRAM 空间的地址。

2.4.2.8 EMC 复位

为了防止由于 ESD、EFT 等强干扰导致芯片寄存器被改写，CSU3AF10 对关键寄存器增加影子寄存器，它们之间是反码关系，当系统检测到它们之间的反码关系不成立，则产生复位信号将芯片复位。可以通过代码选项关闭 EMC 使能位。

CSU3AF10 中增加了影子寄存器进行校验的寄存器包括以下几个：

- 1、WDT 模块使能位
- 2、WWDT 模块使能位
- 3、WDT 时钟使能位
- 4、代码选项
- 5、SLEEP 寄存器

影子寄存器的值用户无法读取，仅做校验使用。

2.4.3 寄存器映射

表 2-57 复位系统寄存器列表

地址	名称	Bit7	Bits6	Bit5	Bits4	Bit3	Bits2	Bit1	Bit0	上电复位值
基址 PAGE0, RSTMU_BA = 0x30										
00h	RSTSR						EMCF	ILOPF	WWDTF	00000000
01h	LVDCON				LVDE N	LVDF				00000000

2.4.4 寄存器描述

2.4.4.1 RSTSR 寄存器（地址 00h）

表 2-58

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	U-0	R-0	R-0	R-0
RSTSR						EMCF	ILOPF	WWDTF

表 2-59

位地址	标识符	功能
7:3	RESERVE	保留
2	EMCF	EMC 复位标志位 该位硬件置 1，软件写 0 清除。 0：未发生 EMC 复位 1：发生了 EMC 复位
1	ILOPF	非法指令复位标志位 该位硬件置 1，软件写 0 清除。 0：未发生非法指令复位 1：发生了非法指令复位
0	WWDTF	WWDT 复位标志位 该位硬件置 1，软件写 0 清除。 0：未发生 WWDT 复位 1：发生了 WWDT 复位

2.4.4.2 LVDCON 寄存器（地址 01h）

表 2-60

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	R/W-0	R-0	U-0	U-0	U-0
LVDCON				LV DEN	LV DF			

表 2-61

位地址	标识符	功能
7: 5	RESERVE	保留
4	LV DEN	低电压检测模块使能 0：LVD 关闭 1：LVD 使能
3	LV DF	低电压标志位： 0：VDD 电压在高于低电压检测档位 1：VDD 电压在低于低电压检测档位 注：只能读，不可写
2:0	RESERVE	保留

2.5 外部中断

2.5.1 概述

CSU3AF10 有 11 个中断源，只有 1 个中断入口地址 004H。与中断相关的 SFR：中断使能控制寄存器 EXIIE 和中断标志位寄存器 EXIIF。这 11 个中断源都各自有一个中断使能，并且共用一个中断总使能位 GIE，它们的标志位硬件置位，软件清 0。

2.5.2 特性

- 支持外部中断 0
- 支持外部中断 1
- 支持外部中断 2
- 中断唤醒 halt 模式
- 中断唤醒 sleep 模式

2.5.3 框图

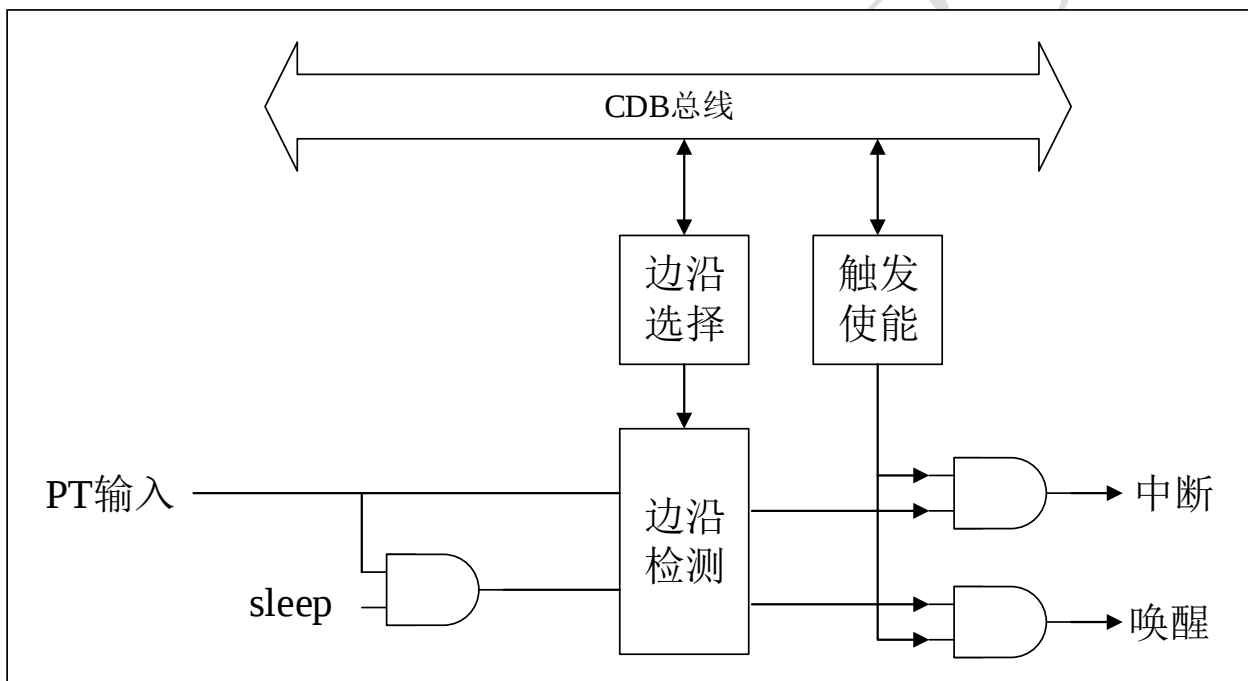


图 2-12 中断结构框图

2.5.4 功能描述

2.5.4.1 外部中断 0

PT1.2、PT1.3、PT1.4 和 PT1.5 为外部中断 0 的输入端。触发方式由 EXICON 寄存器中的 E0M[1:0] 寄存器决定。EXIIE 寄存器中的 E0IE 为外部中断 0 的使能位，EXIIF 寄存器中的 E0IF 为中断标志位，硬件置 1，软件清 0。可唤醒 sleep 或 halt 模式。只要外部中断 0 被触发，中断标志位 E0IF 就会置 1。

2.5.4.2 外部中断 1

PT2.2、PT2.3、PT2.4、PT2.5、PT2.6 可作为外部中断 1 的输入端。触发方式由 EXICON 寄存器中的 E1M 寄存器决定。EXIIE 寄存器中的 E1IE 为外部中断 1 的使能位，EXIIF 寄存器中的 E1IF 为中断标志位，硬件置 1，软件清 0。可唤醒 sleep 或 halt 模式。只要对应 PT 口作为外部中断输入端，且外部中断 1 被触发，中断标志位 E1IF 就会置 1。

2.5.4.3 外部中断 2

PT3.1、PT3.2、PT3.3、PT3.4、PT3.5、PT3.6 都可作为外部中断 2 的输入端。触发方式由 EXICON 寄存器中的 E2M 寄存器决定。EXIIE 寄存器中的 E2IE 为外部中断 2 的使能位，EXIIF 寄存器中的 E2IF 为中断标志位，硬件置 1，软件清 0。可唤醒 sleep 或 halt 模式。只要对应 PT 口作为外部中断输入端，且外部中断 2 被触发，中断标志位 E2IF 就会置 1。

2.5.4.4 外部中断配置流程

按照如下流程配置，使能外部中断：

- 1) 配置 INTCFGx 寄存器（x=1、2、3），选择中断口；
- 2) 配置 ExM[1:0]（x=0、1、2），选择中断触发方式；
- 3) 软件对 ExIF 写 0（x=0、1、2），清除清除对应标志位
- 4) 配置 ExIE（x=0、1、2），使能中断
- 5) 配置总使能 GIE

2.5.5 寄存器映射

表 2-62

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
基址 PAGE0, EXTL_BA = 0x38										
00h	EXICON			E2M[1:0]		E1M[1:0]		E0M[1:0]		00000000
01h	EXIIE						E2IE	E1IE	E0IE	00000000
02h	EXIIF						E2IF	E1IF	E0IF	00000000
03h	INTCFG1	PT1INT[7:0]								00000000
04h	INTCFG2	PT2INT[7:0]								00000000
05h	INTCFG3	PT3INT[7:0]								00000000

2.5.6 寄存器描述

2.5.6.1 EXICON 寄存器（地址为 00h）

表 2-63

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EXICON			E2M[1:0]		E1M[1:0]		E0M[1:0]	

表 2-64

位地址	标识符	功能
7:6	RESERVE	保留
5:4	E2M[1:0]	外部中断 2 触发模式 01 = 外部中断 2 为上升沿触发 00 = 外部中断 2 为下降沿触发 1x = 外部中断 2 在状态改变时触发
3:2	E1M[1:0]	外部中断 1 触发模式 01 = 外部中断 1 为上升沿触发 00 = 外部中断 1 为下降沿触发

		1x = 外部中断 1 在状态改变时触发
1:0	E0M[1:0]	外部中断 0 触发模式 01 = 外部中断 0 为上升沿触发 00 = 外部中断 0 为下降沿触发 1x = 外部中断 0 在状态改变时触发

2.5.6.2 EXIIE 寄存器（地址为 01h）

表 2-65

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
EXIIE						E2IE	E1IE	E0IE

表 2-66

位地址	标识符	功能
7:3	RESERVE	保留
2	E2IE	外部中断 2 使能 1 = 使能外部中断 2 0 = 不使能外部中断 2
1	E1IE	外部中断 1 使能 1 = 使能外部中断 1 0 = 不使能外部中断 1
0	E0IE	外部中断 0 使能 1 = 使能外部中断 0 0 = 不使能外部中断 0

2.5.6.3 EXIIF 寄存器（地址为 02h）

表 2-67

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
INTF						E2IF	E1IF	E0IF

表 2-68

位地址	标识符	功能
7:3	RESERVE	保留
2	E2IF	外部中断 2 中断标志，软件写 0 清除，硬件置高 1 = 外部中断 2 发生中断，必须软件清 0 0 = 外部中断 2 没发生中断
1	E1IF	外部中断 1 中断标志，软件写 0 清除，硬件置高 1 = 外部中断 1 发生中断，必须软件清 0 0 = 外部中断 1 没发生中断
0	E0IF	外部中断 0 中断标志，软件写 0 清除，硬件置高 1 = 外部中断 0 发生中断，必须软件清 0 0 = 外部中断 0 没发生中断

2.5.6.4 INTCFG1 寄存器（地址为 03h）

表 2-69

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
INTCFG1			PT1INT[3:0]					

表 2-70

位地址	标识符	功能
7:6	RESERVE	
5	PT1INT[3]	PT1.5 外部中断 0 使能 0 = 禁止 PT1.5 外部中断 0 1 = 使能 PT1.5 外部中断 0
4	PT1INT[2]	PT1.4 外部中断 0 使能 0 = 禁止 PT1.4 外部中断 0 1 = 使能 PT1.4 外部中断 0
3	PT1INT [1]	PT1.3 外部中断 0 使能 0 = 禁止 PT1.3 外部中断 0 1 = 使能 PT1.3 外部中断 0
2	PT1INT [0]	PT1.2 外部中断 0 使能 0 = 禁止 PT1.2 外部中断 0 1 = 使能 PT1.2 外部中断 0
1:0	RESERVE	保留

2.5.6.5 INTCFG2 寄存器（地址为 04h）

表 2-71

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	R/W -0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
INTCFG2			PT2INT[4:0]					

表 2-72

位地址	标识符	功能
7	RESERVE	保留
6	PT2INT[4]	PT2.6 外部中断 1 使能 0 = 禁止 PT2.6 外部中断 1 1 = 使能 PT2.6 外部中断 1
5	PT2INT[3]	PT2.5 外部中断 1 使能 0 = 禁止 PT2.5 外部中断 1 1 = 使能 PT2.5 外部中断 1
4	PT2INT[2]	PT2.4 外部中断 1 使能 0 = 禁止 PT2.4 外部中断 1 1 = 使能 PT2.4 外部中断 1
3	PT2INT[1]	PT2.3 外部中断 1 使能 0 = 禁止 PT2.3 外部中断 1

		1 = 使能 PT2.3 外部中断 1
2	PT2INT[0]	PT2.2 外部中断 1 使能 0 = 禁止 PT2.2 外部中断 1 1 = 使能 PT2.2 外部中断 1
1: 0	RESERVE	保留

2.5.6.6 INTCFG3 寄存器（地址为 05h）

表 2-73

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0
INTCFG3		PT3INT[5:0]						

表 2-74

位地址	标识符	功能
7	RESERVE	保留
6	PT3INT[5]	PT3.6 外部中断 2 使能 0 = 禁止 PT3.6 外部中断 2 1 = 使能 PT3.6 外部中断 2
5	PT3INT[4]	PT3.5 外部中断 2 使能 0 = 禁止 PT3.5 外部中断 2 1 = 使能 PT3.5 外部中断 2
4	PT3INT[3]	PT3.4 外部中断 2 使能 0 = 禁止 PT3.4 外部中断 2 1 = 使能 PT3.4 外部中断 2
3	PT3INT[2]	PT3.3 外部中断 2 使能 0 = 禁止 PT3.3 外部中断 2 1 = 使能 PT3.3 外部中断 2
2	PT3INT[1]	PT3.2 外部中断 3 使能 0 = 禁止 PT3.2 外部中断 2 1 = 使能 PT3.2 外部中断 2
1	PT3INT[0]	PT3.1 外部中断 2 使能 0 = 禁止 PT3.1 外部中断 2 1 = 使能 PT3.1 外部中断 2
0	RESERVE	保留

2.6 定时器 0

2.6.1 概述

定时器 0 主要实现 8bit 计数器功能，定时器 0 模块的输入时钟的选择和分频在 CLKMU 模块中配置。当选择内部 10KHz WDT 时钟进行计数时，可以唤醒 Sleep 模式。

2.6.2 特性

- 8 位可编程定时器
- Sleep 模式唤醒功能
- 中断上报功能

2.6.3 框图

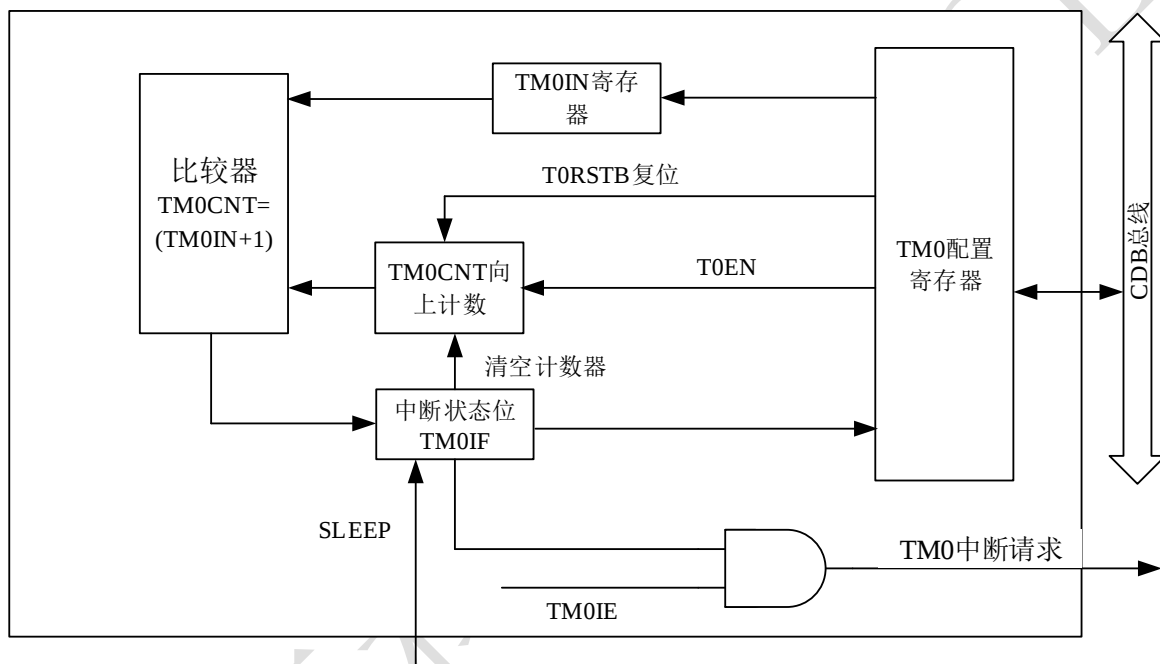


图 2-13 定时器 0 功能框图

2.6.4 基本配置

定时器 0 的时钟源通过时钟管理单元的 CLKSEL0 寄存器的 T0CKS 位进行选择，定时器 0 的时钟分频通过时钟管理单元的 CLKDIV0 寄存器的 T0DIV 位选择。

2.6.5 功能描述

2.6.5.1 定时器 0 时钟

定时器 0 的时钟在时钟管理单元进行配置，定时器 0 可选时钟源如下

- ◆ CPUCLK
- ◆ SYSCLK
- ◆ 外部高速时钟 ECK

◆ 10 KHz 内部 WDT 时钟

定时器 0 的时钟分频同样通过时钟管理单元进行配置，可对定时器 0 时钟源进行 1~128 分频，分频后作为定时器 0 的定时时钟。

2.6.5.2 定时功能

当用户设置了定时器 0 模块的使能位，8 bits 计数器将启动，计数值将会从 00H 递增到 TM0IN。用户需要设置 TM0IN（定时器 0 模块计数溢出值）以选择定时超时时间。

当定时超时发生时，定时器 0 中断标志位 TM0IF 硬件置 1，该位只能通过软件清零。如果使能定时器 0 中断使能（TM0IE=1）和中断总使能（GIE 为内核寄存器信号），程序计数器会跳转到 004H 以执行中断服务程序

定时器 0 溢出时间计算公式如下

$$\text{定时器 0 溢出时间} = (\text{TM0IN} + 1) / \text{TM0CLK}$$

2.6.5.3 定时功能配置步骤

定时功能操作步骤如下：

- 1) 设置 CLKSEL0 寄存器的 T0CKS[1:0]位，为定时器 0 模块选择时钟输入。
- 2) 设置 CLKDIV0 寄存器的 T0DIV[2:0]位，为定时器 0 模块选择时钟输入
- 3) 设置 TM0IN，选择定时器 0 溢出值。（溢出值：1~255）
- 4) 清除定时器 0 中断标志位：将 TM0CON 寄存器的 TM0IF 位清零。
- 5) 设置定时器 0 中断使能：将 TM0IE 位和 INTE 寄存器的与 GIE（GIE 为内核寄存器信号）置位，使能定时器 0 中断。
- 6) 清零定时器 0 计数值：将 TM0CON 寄存器的 T0RSTB 位配置为 0，复位定时器 0 模块的计数器。
- 7) 使能定时器 0 模块：将 TM0CON 寄存器的 T0EN 置位，使能定时器 0 模块的 8 bits 计数器。

2.6.6 寄存器映射

表 2-75

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
基址 PAGE0, TM0_BA = 0x48										
00H	TM0CON	T0EN	TM0IF	TM0IE			T0RSTB			00000100
01H	TM0IN	TM0IN[7:0]								11111111
02H	TM0CNT	TM0CNT[7:0]								00000000

2.6.7 寄存器描述

2.6.7.1 TM0CON 寄存器(地址 00H)

表 2-76

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-1	U-0	U-0
TM0CON	T0EN	TM0IF	TM0IE			T0RSTB		

表 2-77

位地址	标识符	功能
7	TOEN	定时器 0 使能位 1: 使能定时器 0 0: 禁止定时器 0
6	TM0IF	8-Bit 定时器 0 中断标志, 软件写 0 清零, 硬件置高 1 = 发生定时中断, 必须软件清 0 0 = 没发生定时中断
5	TM0IE	8-Bit 定时 0 器中断使能 1 = 使能定时器 0 中断 0 = 不使能定时器 0 中断
4:3	RESERVE	保留
2	TORSTB	定时器 0 复位, 默认值为 1 1: 禁止定时器 0 复位 0: 使能定时器 0 复位 当将该位配置为 0, 一个指令周期后定时器 0 复位完成, TORSTB 硬件置 1。
1:0	RESERVE	保留

2.6.7.2 TM0IN 寄存器(地址 01H)

表 2-78

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TM0IN	TM0IN[7:0]							

表 2-79

位地址	标识符	功能
7 : 0	TM0IN[7:0]	定时器 0 溢出值 (溢出值: 1~255)

2.6.7.3 TM0CNT 寄存器(地址 02H)

表 2-80

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TM0CNT	TM0CNT[7:0]							

表 2-81

位地址	标识符	功能
7 : 0	TM0CNT[7:0]	定时器 0 计数寄存器, 只读

2.7 GPIO

2.7.1 概况

微控制器中的通用 I/O 口（GPIO）用于通用的输入与输出功能。用户可以通过 GPIO 接收数据信号或将数据传送给其它的数字设备。CSU3AF10 的部分 GPIO 可以被定义为其它的特殊功能。在本节，只说明 GPIO 的通用 I/O 口功能，特殊功能将会在接下来的章节中说明。

CSU3AF10 的所有 GPIO 都支持内部上拉，上拉电阻阻值均为 $30K\Omega$ ，通过寄存器 PT1PU、PT2PU、PT3PU 进行配置。当将 GPIO 配置为模拟口时，芯片内部上拉电阻自动断开。而将 GPIO 配置为数字口时，上拉电阻不会自动断开。

2.7.2 特性

- 输出状态：带有上拉的推挽输出或开漏输出
- 输入状态：浮空、上拉、模拟输入
- 模拟功能
- 允许 GPIO 口和外设引脚的高灵活性复用

2.7.3 框图

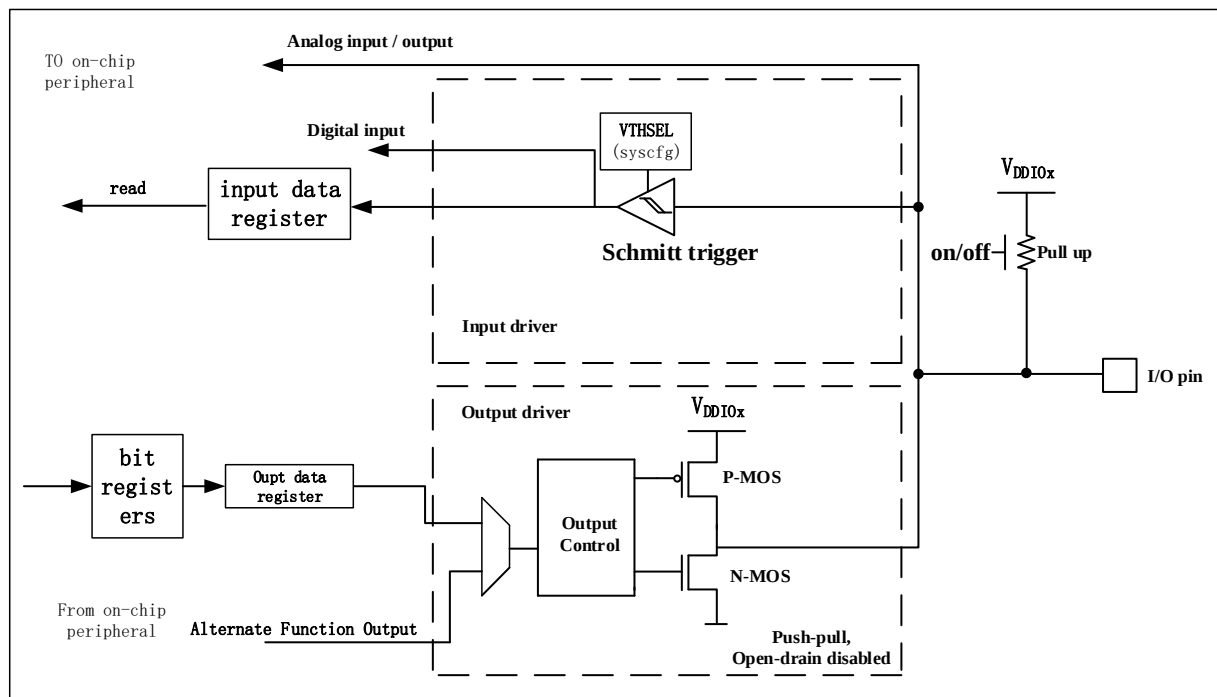


图 2-11 GPIO 整体框图

2.7.4 基本配置

gpio 的功能由 syscfg 的寄存器控制，其决定管脚具体功能。

2.7.5 功能描述

2.7.5.1 基本操作

基本操作配置如下：

- 1 在 syscfg 寄存器中将管脚设置成期望使用的功能
- 2 如果作为模拟功能需要在 PTOFFDIN 关闭相应的输入接收
- 3 如果设置为 gpio 功能根据需求配置 PTxEN, PTxPU, 和 PTx
- 4 包含开漏输出的管脚 PT2.1 PT2.0 作为 I2C 使用时通过 PT2OTYPE 寄存器配置为开漏输出
- 5 对于需要管脚短接的配置寄存器 PTSHORT (用户不可见)

2.7.5.2 模拟功能

作为模拟输入输出时，IO 的上拉电阻断开，对于模拟输入可能会导致数字逻辑功耗增大，因此可以通过配置 PTOFFDIN 寄存器断开其数字输入。

2.7.5.3 复用功能

当作为复用功能时，IO 直接由对应的使用模块控制。

2.7.5.4 通用 GPIO

当作为通用 gpio 使用时，其上拉由 PTxPU 控制，输出使能由 PTxEN 控制，输出值通过 PTx 控制。当需要软件实现 I2C 时配置 PT2OTYPE 寄存器将管脚设置为开漏模式。

2.7.5.5 关闭输入

对于模拟和一些其它情况下需要关闭管脚信号的情况可以通过 PTOFFDIN 关闭管脚输入。

2.7.5.6 管脚短接（用户不可见）

管脚 P20/23,P21/24,P22/25,P37/30 可以短接，在寄存器 PTSHORT 中配置。

2.7.6 寄存器映射

表 2-82 GPIO 寄存器表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
基址 PAGE0, GPIO0_BA = 0x50										
00h	PT1	PT1[7:0]								xxxxxxxx
01h	PT1EN	PT1EN[7:0]								00000000
02h	PT1PU	PT1PU[7:0]								00000000
05h	PT2		PT2[6:0]							0xxxxxxxx
06h	PT2EN		PT2EN[6:0]							00000000
07h	PT2PU		PT2PU[6:0]							00000000
09h	PT2OTYPE							PT2OTYPE[1:0]		00000000
0Ah	PT3	PT3[7:0]								xxxxxxxx
0Bh	PT3EN	PT3EN[7:0]								00000000
0Ch	PT3PU	PT3PU[7:0]								00000000
0Dh	PTOFFDIN	PT2OFFDIN[4:3]		PT3OFFDIN[6:1]						00000000
0Eh	PTSHORT					SHORT_20 23	SHORT_21 24	SHORT_22 25	SHORT_37 30	00000000

2.7.7 寄存器描述

2.7.7.1 PT1 寄存器（地址为 00h）

表 2-83

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
PT1	PT1[7:0]							

表 2-84

位地址	标识符	功能
7:0	PT1[7:0]	PT1 口 bit7~bit0 数据位 进行读操作时读取 GPIO 值 进行写操作配置 GPIO 口输出值。

2.7.7.2 PT1EN 寄存器（地址为 01h）

表 2-85

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W -0	R/W -0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT1	PT1EN [7:0]							

表 2-86

位地址	标识符	功能
7	PT1EN[7]	PT1.7 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
6	PT1EN[6]	PT1.6 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
5	PT1EN[5]	PT1.5 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
4	PT1EN[4]	PT1.4 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
3	PT1EN[3]	PT1.3 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
2	PT1EN[2]	PT1.2 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
1	PT1EN[1]	PT1.1 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
0	PT1EN[0]	PT1.0 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口

2.7.7.3 PT1PU 寄存器（地址为 02h）

表 2-87

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W -0	R/W -0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT1	PT1PU [7:0]							

表 2-88

位地址	标识符	功能
7	PT1PU[7]	PT1.7 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
6	PT1PU[6]	PT1.6 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻

		GPIO 口上拉电阻在配置为模拟口时自动断开
5	PT1PU[5]	PT1.5 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
4	PT1PU[4]	PT1.4 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
3	PT1PU[3]	PT1.3 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
2	PT1PU[2]	PT1.2 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
1	PT1PU[1]	PT1.1 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
0	PT1PU[0]	PT1.0 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开

2.7.7.4 PT2 寄存器（地址为 05h）

表 2-89

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
PT2	PT2[6:0]							

表 2-90

位地址	标识符	功能
7	RESERVE	
6: 0	PT2[6:0]	PT2 口 bit6~bit0 数据位 进行读操作时读取 GPIO 值 进行写操作配置 GPIO 口输出值。

2.7.7.5 PT2EN 寄存器（地址为 06h）

表 2-91

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT2EN	PT2EN[6:0]							

表 2-92

位地址	标识符	功能
7	RESERVE	
6	PT2EN[6]	PT2.6 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
5	PT2EN[5]	PT2.5 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
4	PT2EN[4]	PT2.4 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
3	PT2EN[3]	PT2.3 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
2	PT2EN[2]	PT2.2 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
1	PT2EN[1]	PT2.1 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
0	PT2EN[0]	PT2.0 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口

2.7.7.6 PT2PU 寄存器（地址为 07h）

表 2-93

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT2PU		PT2PU[6:0]						

表 2-94

位地址	标识符	功能
7	RESERVE	
6	PT2PU[6]	PT2.6 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
5	PT2PU[5]	PT2.5 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
4	PT2PU[4]	PT2.4 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
3	PT2PU[3]	PT2.3 口上拉电阻使能位 0 = 断开上拉电阻

		1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
2	PT2PU[2]	PT2.2 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
1	PT2PU[1]	PT2.1 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
0	PT2PU[0]	PT2.0 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开

2.7.7.7 PT2OTYPE 寄存器（地址为 09h）

表 2-95

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	U-0	U-0
PT2OTYPE							PT2OTYPE[1:0]	

表 2-96

位地址	标识符	功能
7:2	RESERVE	
1	PT2OTYPE[1]	PT2.1 口输出类型 0: 推挽输出 1: 开漏输出
0	PT2OTYPE[0]	PT2.0 口输出类型 0: 推挽输出 1: 开漏输出

2.7.7.8 PT3 寄存器（地址为 0Ah）

表 2-97

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X	R/W-X
PT3	PT3[7:0]							

表 2-98

位地址	标识符	功能
7: 0	PT3[7:0]	PT3 口 bit7~bit0 数据位 进行读操作时读取 GPIO 值 进行写操作配置 GPIO 口输出值。

2.7.7.9 PT3EN 寄存器（地址为 0Bh）

表 2-99

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT3EN	PT3EN[7:0]							

表 2-100

位地址	标识符	功能
7	PT3EN[7]	PT3.7 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
6	PT3EN[6]	PT3.6 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
5	PT3EN[5]	PT3.5 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
4	PT3EN[4]	PT3.4 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
3	PT3EN[3]	PT3.3 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
2	PT3EN[2]	PT3.2 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
1	PT3EN[1]	PT3.1 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口
0	PT3EN[0]	PT3.0 口输入输出控制位 0 = 定义为输入口 1 = 定义为输出口

2.7.7.10 PT3PU 寄存器（地址为 0Ch）

表 2-101

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PT3PU	PT3PU[7:0]							

表 2-102

位地址	标识符	功能
7	PT3PU[7]	PT3.7 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开

6	PT3PU[6]	PT3.6 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
5	PT3PU[5]	PT3.5 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
4	PT3PU[4]	PT3.4 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
3	PT3PU[3]	PT3.3 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
2	PT3PU[2]	PT3.2 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
1	PT3PU[1]	PT3.1 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开
0	PT3PU[0]	PT3.0 口上拉电阻使能位 0 = 断开上拉电阻 1 = 使能上拉电阻 GPIO 口上拉电阻在配置为模拟口时自动断开

2.7.7.11 PTOFFDIN 寄存器（地址为 0Dh）

表 2-103

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTOFFDIN	PT2OFFDIN[4:3]		PT3ODFFDIN[6:1]					

表 2-104

位地址	标识符	功能
7	PT2OFFDIN[4]	PT2.4 数字输入功能控制 0 = 打开数字输入 1 = 关闭数字输入
6	PT2OFFDIN[3]	PT2.3 数字输入功能控制 0 = 打开数字输入 1 = 关闭数字输入
5	PT3OFFDIN[6]	PT3.6 数字输入功能控制 0 = 打开数字输入 1 = 关闭数字输入
4	PT3OFFDIN[5]	PT3.5 数字输入功能控制 0 = 打开数字输入

		1 = 关闭数字输入
3	PT3OFFDIN[4]	PT3.4 数字输入功能控制 0 = 打开数字输入 1 = 关闭数字输入
2	PT3OFFDIN[3]	PT3.3 数字输入功能控制 0 = 打开数字输入 1 = 关闭数字输入
1	PT3OFFDIN[2]	PT3.2 数字输入功能控制 0 = 打开数字输入 1 = 关闭数字输入
0	PT3OFFDIN[1]	PT3.1 数字输入功能控制 0 = 打开数字输入 1 = 关闭数字输入

2.7.7.12 PTSHORT 寄存器（地址为 0Eh）

写该寄存器之前，需要向 TEST 寄存器连续写入 A5h, Adh，解锁写保护功能。

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
PTSHORT					SHORT_2023	SHORT_2124	SHORT_2225	SHORT_3730

位地址	标识符	功能
7:4	RESERVED	保留
3	SHORT_2023	PT2.0 旁路至 PT2.3 数字输入功能控制 0 = 关闭旁路 1 = 使能旁路
2	SHORT_2124	PT2.1 旁路至 PT2.4 数字输入功能控制 0 = 关闭旁路 1 = 使能旁路
1	SHORT_2225	PT2.2 旁路至 PT2.5 数字输入功能控制 0 = 关闭旁路 1 = 使能旁路
0	SHORT_3730	PT3.7 旁路至 PT3.0 数字输入功能控制 0 = 关闭旁路 1 = 使能旁路

3 增强功能

3.1 Halt 和 Sleep 模式

CSU3AF10 支持低功耗工作模式。目前低功耗模式有 2 种，分别是 Halt 模式或 Sleep 模式。

3.1.1 Halt 模式

芯片执行 HALT 指令后进入 HALT 模式，内核指令停止运行，程序计数器停止计数直到出现中断事件，任何中断都可以导致芯片退出 Halt 模式。

Halt 模式下，芯片内部高速振荡器和内部 10 KHz WDT 时钟仍然正常工作，内核时钟停止，定时器可以正常计数，ADC 仍然可以继续未完成转换。

Halt 模式功耗介于 SLEEP 模式和正常模式之间，由于高速时钟和外设仍然在工作，因此 Halt 模式功耗还是比较大。

3.1.2 Sleep 模式

3.1.2.1 SLEEP 模式进入

芯片执行 SLEEP 指令后，进入睡眠模式，此时

- 内核停止执行指令
- 内部高速振荡器和外部高速晶振停止工作
- 内部低速振荡器保持原有工作状态
- 内置 LDO 切换到低功耗模式
- ADC、比较器、IO 口维持原有状态，建议在进入 SLEEP 模式前配置为关闭或者固定状态

3.1.2.2 SLEEP 模式退出

在 SLEEP 模式下，出现一个中断事件或者芯片复位即可唤醒 CPU，能唤醒 SLEEP 模式的中断源在[中断向量表](#)中有描述，请参考。当芯片退出 SLEEP 模式时，会执行以下动作

- 内置 LDO 退出低功耗模式
- 内部高速振荡器和外部高速振荡器回复正常工作
- 延时 20us
- 内核恢复工作，继续执行指令

芯片在睡眠模式下的功耗大约 6uA。当系统从 sleep 模式唤醒后，系统会在延迟 20us 之后，重新开始工作。

为了保证 CPU 在睡眠模式下的功耗最小，在执行睡眠指令之前，需要保证所有的输入口是接到 VDD 或 VSS 电平。ADC 模块在 SLEEP 模式下必须关闭，并且模拟输入通道不能配置为 1/8VDD（SRADCON2 寄存器的 CHS[4:0]位不能配置为 101XXB）。

注：

芯片如果处于 sleep 状态，这时候降低电压，配置的低电压复位不会起作用，低于 POR 掉电复位点才会复位。如果 sleep 唤醒后，此时还处于低电压复位点以下，会立即复位。

3.2 SYSCFG

3.2.1 IO 复用功能

通过 CDB 总线配置相应寄存器，控制对应 IO 口的复用功能。

PTXXFUNC	00	01	10	11
PT1.0	通用 GPIO			
PT1.1	通用 GPIO			
PT1.2	通用 GPIO	模拟功能	PWM4 输出	定时器 3 外部输入 T3
PT1.3	通用 GPIO			
PT1.4	通用 GPIO	外部晶振输出 XOUT		
PT1.5	通用 GPIO	外部晶振输入 XIN		
PT1.6	通用 GPIO			
PT1.7	通用 GPIO			
PT2.0	通用 GPIO	模拟功能 AIN8		
PT2.1	通用 GPIO	模拟功能 AIN9		
PT2.2	通用 GPIO	模拟功能 AIN10		
PT2.3	通用 GPIO	模拟功能 AIN11		PWM2 输出
PT2.4	通用 GPIO	模拟功能 AIN12		PWM3 输出
PT2.5	通用 GPIO	模拟功能 AIN13	互补 PWM3H 输出	ICD 通信
PT2.6	通用 GPIO	模拟功能 AIN14	互补 PWM3L 输出	
PT3.0	通用 GPIO	模拟功能 (AIN0、VREF、NTC)		
PT3.1	通用 GPIO	模拟功能 AIN1	UART TX	
PT3.2	通用 GPIO	模拟功能 AIN2	UART RX	
PT3.3	通用 GPIO	模拟功能 AIN3	UART TX	
PT3.4	通用 GPIO	模拟功能 AIN4	UART RX	
PT3.5	通用 GPIO	模拟功能 AIN5	UART TX	
PT3.6	通用 GPIO	模拟功能 AIN6	UART RX	
PT3.7	通用 GPIO	模拟功能 AIN7		

3.2.2 寄存器映射

表 3-1

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
基址 PAGE0, SYS_BA = 0x28										
00h	CFGR1	VTHSEL	DIG_METCH[3:2]				TMP_MEAS_EN	PT34F UNC_2	PT33F UNC_2	00000000
01h	CFGR2	PT13FUNC[1:0]		PT12FUNC[1:0]		PT11FUNC[1:0]		PT10FUNC[1:0]		00000000
02h	CFGR3	PT17FUNC[1:0]		PT16FUNC[1:0]		PT15FUNC[1:0]		PT14FUNC[1:0]		00000000
03h	CFGR4	PT23FUNC[1:0]		PT22FUNC[1:0]		PT21FUNC[1:0]		PT20FUNC[1:0]		00000000
04h	CFGR5			PT26FUNC[1:0]		PT25FUNC[1:0]		PT24FUNC[1:0]		00001100
05h	CFGR6	PT33FUNC[1:0]		PT32FUNC[1:0]		PT31FUNC[1:0]		PT30FUNC[1:0]		00000000
06h	CFGR7	PT37FUNC[1:0]		PT36FUNC[1:0]		PT35FUNC[1:0]		PT34FUNC1:0]		00000000
07h	CFGR8	CMPVR_EN	CMP1VRS[2:0]			CMP0VRS[3:0]				01001000

3.2.3 寄存器描述

3.2.3.1 CFGR1 寄存器（地址为 00h）

表 3-2

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CFGR1	VTHSEL	DIG_METCH[3:0]				TMP_MEAS_EN	PT34FUNC_2	PT33FUNC_2

表 3-3

位地址	标识符	功能
7	VTHSEL	输入逻辑电平电压控制信号
		VTHSEL 输入逻辑电平
		1
		符号 参数 最小值 典型值 最大值 单位
		VIH1 数字输入高电平 0.7VDD
		VIL1 数字输入低电平 0.3VDD
		0
		符号 参数 最小值 典型值 最大值 单位
		VIH2 数字输入高电平 0.5VDD
		VIL2 数字输入低电平 0.5VDD
6:5	DIG_METCH[3:2]	数字预留寄存器
4:3	保留	
2	TMP_MEAS_EN	测温电阻使能控制，1 使能打开，0 关闭
1	PT34FUNC_2	该位和 PT34FUNC[1:0]一起（拼接）控制 PT3.4 口的复用功能
0	PT33FUNC_2	该位和 PT33FUNC[1:0]一起（拼接）控制 PT3.3 口的复用功能

3.2.3.2 CFGR2 寄存器（地址为 01h）

表 3-4

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CFGR2	PT13FUNC[1:0]		PT12FUNC[1:0]		PT11FUNC[1:0]		PT10FUNC[1:0]	

表 3-5

位地址	标识符	功能
7:6	PT13FUNC[1:0]	PT1.3 复用功能选择 00: 通用 GPIO 01 保留 1x: 保留

5:4	PT12FUNC[1:0]	PT1.2 复用功能选择 00: 通用 GPIO 01: 模拟功能 10: PWM4 输出 11: 定时器 3 外部输入 T3
3:2	PT11FUNC[1:0]	PT1.1 复用功能选择 00: 通用 GPIO 其他: 保留
1:0	PT10FUNC[1:0]	PT1.0 复用功能选择 00: 通用 GPIO 其他: 保留

3.2.3.3 CFGR3 寄存器（地址为 02h）

表 3-6

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0
CFGR3	PT17FUNC[1:0]		PT16FUNC[1:0]		PT15FUNC[1:0]		PT14FUNC[1:0]	

表 3-7

位地址	标识符	功能
7:6	PT17FUNC[1:0]	PT1.7 复用功能选择 00: 通用 GPIO 其他: 保留
5:4	PT16FUNC[1:0]	PT1.6 复用功能选择 00: 通用 GPIO 其他: 保留
3:2	PT15FUNC[1:0]	PT1.5 复用功能选择 00: 通用 GPIO（swd_clk 由硬件控制） 01: 外部晶振输入 XIN（如果写 PT14FUNC[1:0]寄存器，本寄存器跟随变化） 10: 保留 11: 保留
1:0	PT14FUNC[1:0]	PT1.4 复用功能选择 00: 通用 GPIO（swd_data 由硬件控制） 01: 外部晶振输出 XOUT（如果写 PT15FUNC[1:0]寄存器，本寄存器跟随变化） 10: 保留 11: 保留

3.2.3.4 CFGR4 寄存器（地址为 03h）

表 3-8

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0
CFGR4	PT23FUNC[1:0]		PT22FUNC[1:0]		PT21FUNC[1:0]		PT20FUNC[1:0]	

表 3-9

位地址	标识符	功能
7:6	PT23FUNC[1:0]	PT2.3 复用功能选择 00: 通用 GPIO 01: 模拟功能 AIN11 10: 保留 11: PWM2 输出
5:4	PT22FUNC[1:0]	PT2.2 复用功能选择 00: 通用 GPIO 01: 模拟功能 AIN10 1x: 保留
3:2	PT21FUNC[1:0]	PT2.1 复用功能选择 00: 通用 GPIO 01: 模拟功能 AIN9 1x: 保留
1:0	PT20FUNC[1:0]	PT2.0 复用功能选择 00: 通用 GPIO 01: 模拟功能 AIN8 1x: 保留

3.2.3.5 CFGR5 寄存器（地址为 04h）

表 3-10

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W -0	R/W -0	R/W -0	R/W -0	R/W -1	R/W -1	R/W -0	R/W -0
CFGR5	RESERVE		PT26FUNC[1:0]		PT25FUNC[1:0]		PT24FUNC[1:0]	

表 3-11

位地址	标识符	功能
7:6	RESERVE	
5:4	PT26FUNC[1:0]	PT2.6 复用功能选择 00: 通用 GPIO 01: 模拟功能 AIN14 10: 互补 PWM3L 输出 11: 保留
3:2	PT25FUNC[1:0]	PT2.5 复用功能选择 00: 通用 GPIO 01: 模拟功能 AIN13 10: 互补 PWM3H 输出 11: ICD 通信
1:0	PT24FUNC[1:0]	PT2.4 复用功能选择 00: 通用 GPIO 01: 模拟功能 AIN12 10: 保留 11: PWM3 输出

3.2.3.6 CFGR6 寄存器（地址为 05h）

表 3-12

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0
CFGR6	PT33FUNC[1:0]		PT32FUNC[1:0]		PT31FUNC[1:0]		PT30FUNC[1:0]	

表 3-13

位地址	标识符	功能
7:6	PT33FUNC[1:0]	该位和 CFGR1 寄存器的 PT33FUNC_2 寄存器一起控制 PT3.3 口的复用功能选择 { PT33FUNC_2, PT33FUNC[1:0]} 000: 通用 GPIO 001: 模拟功能 AIN3 010: UART TX 011: 保留
5:4	PT32FUNC[1:0]	PT3.2 复用功能选择 00: 通用 GPIO 01: 模拟功能 AIN2 10: UART RX 11: 保留
3:2	PT31FUNC[1:0]	PT3.1 复用功能选择 00: 通用 GPIO 01: 模拟功能 AIN1 10: UART TX 11: 保留
1:0	PT30FUNC[1:0]	PT3.0 复用功能选择 00: 通用 GPIO 01: 模拟功能 (AIN0、VREF、NTC) 1x: 保留

3.2.3.7 CFGR7 寄存器（地址为 06h）

表 3-14

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0
CFGR7	PT37FUNC[1:0]		PT36FUNC[1:0]		PT35FUNC[1:0]		PT34FUNC[1:0]	

表 3-15

位地址	标识符	功能
7:6	PT37FUNC[1:0]	PT3.7 复用功能选择 00: 通用 GPIO 01: 模拟功能 AIN7 1x: 保留

5:4	PT36FUNC[1:0]	PT3.6 复用功能选择 00: 通用 GPIO 01: 模拟功能 AIN6 10: UART RX 11: 保留
3:2	PT35FUNC[1:0]	PT3.5 复用功能选择 00: 通用 GPIO 01: 模拟功能 AIN5 10: UART TX 11: 保留
1:0	PT34FUNC[1:0]	该位和 CFGR1 寄存器的 PT34FUNC_2 寄存器一起控制 PT3.4 口复用功能选择 { PT34FUNC_2, PT34FUNC[1:0]} 000: 通用 GPIO 001: 模拟功能 AIN4 010: UART RX 011: 保留

3.2.3.8 CFGR8 寄存器（地址为 07h）

表 3-16

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W -0	R/W -1	R/W -0	R/W -0	R/W -1	R/W -0	R/W -0	R/W -0
CFGR8	CMPVR_EN	CMP1VRS[2:0]			CMP0VRS[3:0]			

表 3-17

位地址	标识符	功能
7	CMPVR_EN	比较器 1 和比较器 0 参考电压使能信号 0: 关闭比较器 1 参考电压 CMP1_VREF 和比较器 0 参考电压 CMP0_VREF 1: 使能比较器 1 参考电压 CMP1_VREF 和比较器 0 参考电压 CMP0_VREF
6:4	CMP1VRS[2:0]	比较器 1 参考电压 CMP1_VREF 选择控制 000 = 0.2V 001 = 0.4V 010 = 0.66V 011 = 0.8V 100 = 1.23V 101 = 1.58V 110 = 2.18V 111 = 2.6V
3:0	CMP0VRS[3:0]	比较器 0 参考电压 CMP0_VREF 选择控制 0000 = 0V 0001 = 0.32 0010 = 0.6V 0011 = 1.0V 0100 = 2.0V 0101 = 2.7V 0110 = 3.0V 0111 = 3.3V

		10xx = 1.2V 11xx = 2.4V
--	--	----------------------------

芯海科技CHIPSEA

3.3 看门狗(WDT)

3.3.1 概况

看门狗定时器（WDT）用于防止程序由于某些不确定因素而失去控制。当 WDT 启动后，WDT 开始计时，当 WDT 超时后将使 CPU 复位。一般在 WDT 复位 CPU 之前正在运行的程序会先清除 WDT 的计数值。当程序因为故障没有及时清除 WDT 的计数值，WDT 计时将会在超时后将 CPU 复位。

3.3.2 特性

- 内部 10KHz 时钟源
- WDT 计数时钟有 8 种分频可配
- 8 位向上计数器
- 计数溢出值 8 位可配
- 可产生全局复位
- 可由代码选项控制开启

3.3.3 框图

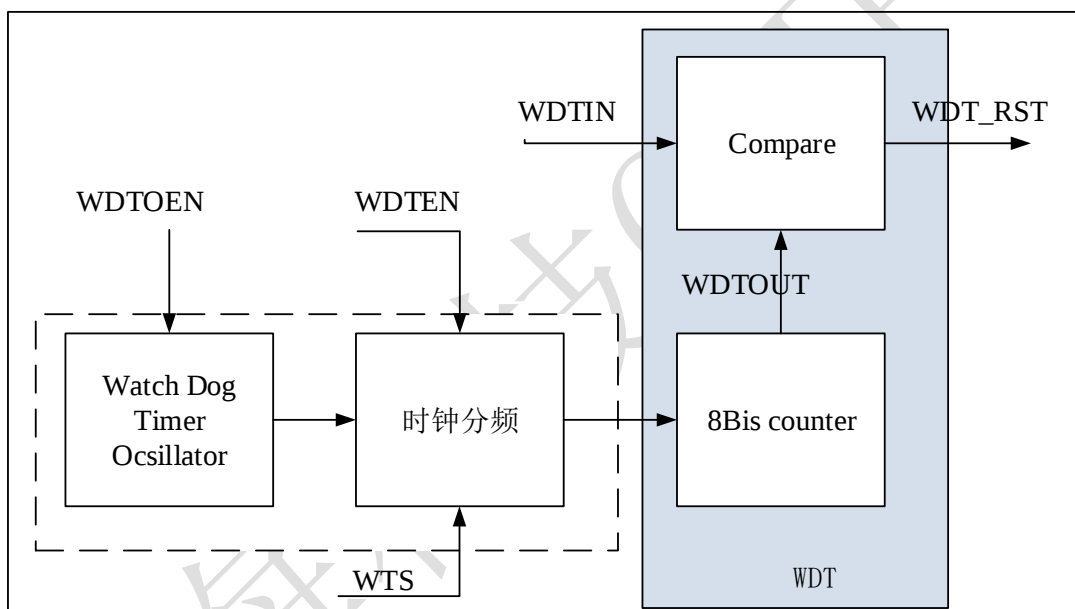


图 3-1 看门狗定时器功能框图

3.3.4 基本配置

如果代码选项 WDT_CFG 为 0，WDT 自动打开
将 MCK 寄存器的 CST_WDT 位置 1，使能内部 10K WDT 时钟

3.3.5 功能描述

3.3.5.1 基本操作

如果代码选项 WDT_CFG 为 0，WDT 自动打开否则执行以下步骤
操作步骤如下：

1. 设置 WDTCON 寄存器的 WDTS[2:0]位，选择 WDT 时钟频率。
2. 设置 WDTIN，选择溢出时间值
3. 使能 WDT 定时器：将 WDTCON 寄存器的 WDTEN 位置 1，使能 WDT。
4. 在程序中执行 CLRWDT 指令清除 WDT 计数值。

3.3.5.2 WDT 代码选项启动

WDT 模块使能和 10 KHz WDT 时钟使能受代码选项 WDT_CFG 控制，当 WDT_CFG 置 0 时，WDT 模块使能和 10 KHz WDT 时钟使能固定打开，软件无法关闭；当 WDC_CFG 置 1 时，WDT 模块使能由 WDTEN 控制，10 KHz WDT 时钟使能由 CST_WDT 控制。

3.3.5.3 WDT 时钟选择

当用户把 CST_WDT 置 1 时，内部的看门狗定时器振荡器（10 KHz）将会启动，产生的时钟被送到 8 位预分频计数器进行分频。当用户置位 WDTEN 时，8 位预分频计数器开始工作，分频后的时钟通过 WDTS[2:0]控制的多路选择器进行选择，得到 WDT 计数器时钟。当 8 位计数器计数值与 WDTIN 数值相等时判断为溢出，溢出时 WDT 会发送 WDTOUT 信号复位 CPU 以及置位 TO 标志位。用户可以使用指令 CLRWDT 复位 WDT 计数值。

3.3.5.4 WDT 计时

WDT 溢出时间计算公式：

$$\frac{2^{(8-WDTS[2:0])}}{10k} * (WDTIN[7:0] + 1)$$

WDTS[2:0]范围为 0~7，WDTIN[7:0]范围为 0~255。

表 3-18

WDTS[2:0]	计数器时钟	时间（当 WDTIN==FFH）
000	WDTA [0]	6553.6ms
001	WDTA [1]	3276.8ms
010	WDTA [2]	1638.4ms
011	WDTA [3]	819.2ms
100	WDTA [4]	409.6ms
101	WDTA [5]	204.8ms
110	WDTA [6]	102.4ms
111	WDTA [7]	51.2ms

3.3.6 寄存器映射

表 3-19

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
基址 PAGE0, WDT_BA = 0x40										
00h	WDTCON	WDTEN					WDTS[2:0]			00000000
01h	WDTIN	WDT_IN[7:0]								11111111

3.3.7 寄存器描述

3.3.7.1 WDTCON 寄存器（地址为 00h）

表 3-20

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
WDTCON	WDTEN					WDTS[2:0]		

表 3-21

位地址	标识符	功能																		
7	WDTEN	看门狗使能位，高电平有效。当 WDT_CFG 配置为 WDT 模块固定打开时，该位无效。 0：关闭 WDT 模块 1：使能 WDT 模块																		
6:3	保留	保留																		
2: 0	WDTS[2:0]	WDT 计数时钟分频																		
		<table><tr><td>WDTS [2:0]</td><td>WDT 计数时钟</td></tr><tr><td>000</td><td>WDTCLK /256</td></tr><tr><td>001</td><td>WDTCLK /128</td></tr><tr><td>010</td><td>WDTCLK /64</td></tr><tr><td>011</td><td>WDTCLK /32</td></tr><tr><td>100</td><td>WDTCLK /16</td></tr><tr><td>101</td><td>WDTCLK /8</td></tr><tr><td>110</td><td>WDTCLK /4</td></tr><tr><td>111</td><td>WDTCLK /2</td></tr></table>	WDTS [2:0]	WDT 计数时钟	000	WDTCLK /256	001	WDTCLK /128	010	WDTCLK /64	011	WDTCLK /32	100	WDTCLK /16	101	WDTCLK /8	110	WDTCLK /4	111	WDTCLK /2
		WDTS [2:0]	WDT 计数时钟																	
		000	WDTCLK /256																	
		001	WDTCLK /128																	
		010	WDTCLK /64																	
		011	WDTCLK /32																	
		100	WDTCLK /16																	
		101	WDTCLK /8																	
		110	WDTCLK /4																	
111	WDTCLK /2																			

3.3.7.2 WDTIN 寄存器（地址为 01h）

表 3-22

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
WDTIN	WDT_IN[7:0]							

表 3-23

位地址	标识符	功能
7: 0	WDT_IN[7:0]	WDT 计数输入，默认值为 FFh 该寄存器值不能写入 00h，如果写入 00h，则 WDT_IN[7:0]的值会被硬件修改为 FFh。写入除 00h 以外的其他值则不受影响。

3.4 窗看门狗（WWDT）

3.4.1 概况

WWDT 主要用来检测由外部接口或者无法预测的逻辑错误导致的软件出错。软件出错时导致应用程序跑飞，WWDT 可以产生复位请求来复位 MCU。

WWDT 是一个向下计数器，从 7Fh 向下进行计数，当计数值由 40h 跳变到 3Fh 时，WWDT 将产生 MCU 复位。当程序在计数值减少到 40h 以下（即 TR[6]变为 0）之前刷新计数器的计数值，则 WWDT 不会产生复位。当 WWDT 的 7 位向下计数值在达到窗寄存器 WWDTWR 值 WD[6:0]之前被刷新，则 WWDT 也会产生复位。

3.4.2 特性

- 7 位可编程向下计数器
- 计数器降至 3F 时产生复位
- 计数器的值在大于窗时更新产生复位

3.4.3 框图

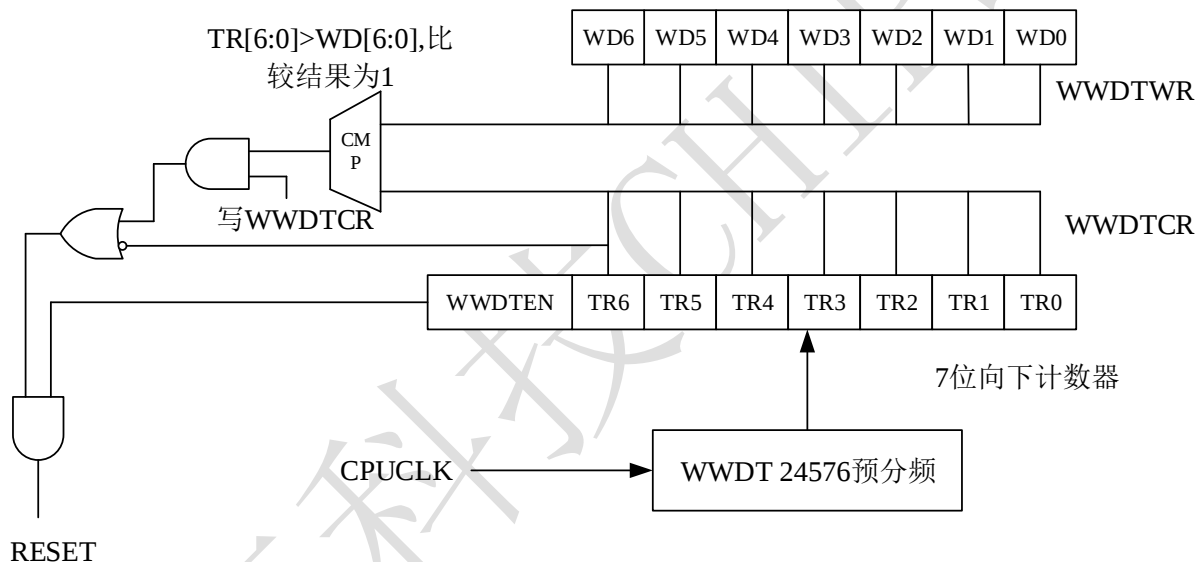


图 3-2 WWDT 框图

3.4.4 基本配置

WWDT 时钟速度与系统时钟速度一致。无需其它配置。

3.4.5 功能描述

WWDT 使能后（WWDTEN 被置位），当 7 位向下计数器值（TR[6: 0]）从 40h 变为 3Fh（TR[6]被清零）时，它将产生一个复位信号，将芯片复位。

WWDT 使能后，当软件在计数器值大于窗值（WD[6:0]）时刷新计数器的值，WWDT 也将产生复位。

应用程序必须在正常的时间间隔内更新 WWDTCR 寄存器的计数值，防止 MCU 复位。该更新操作只有在计数器值低于窗值时才能进行。写入 WWDTCR 的值必须在 FFh 和 C0h 之间，即写入 TR[6:0]的值必须在 7Fh 和 40h 之间。

3.4.5.1 基本操作

基本操作顺序如下：

1. 配置 WWDTCR 和 WWDTWR，配置窗值和计数值①
2. 配置 WWDTEN 为 1 启动 WWDT②
3. 当 TR[6:0]小于 WD 时可以通过 WWDTCR 寄存器更新 TR

注意：①对这两个寄存器的配置顺序没有要求，但是 WWDTCR 寄存器的 TR[6:0]值必须大于 40h。然后配置 WWDTEN 为 1，则 WWDT 开始正常工作。

②如果需要更新 TR 和 WD 的值，则需要先更新 TR 的值，然后再更新 WD 的值，否则有可能直接复位。

3.4.5.2 WWDT 使能

芯片复位后，WWDT 默认关闭。通过将 WWDTCR 寄存器的 WWDTEN 位置 1 来使能 WWDT，WWDT 开启后除了复位以外，没有其他方法可以关闭 WWDT。

WWDT 未使能时，WWDT 计数器仍然会进行计数，但是即使计数值从 40h 跳变到 3Fh，也不会产生复位。

在使能 WWDT 前，必须先将 WWDT 的计数值更新为大于 40h 的值，否则使能 WWDT 时，可能会立即复位。

3.4.5.3 WWDT 计时

WWDT 计时超时时间计算公式如下

$$T_{WWDT} = T_{CPU} * 24576 * (TR[6:0] - 40H + 1)$$

其中， T_{CPU} 时钟周期单位为 ms。

WWDT 计数时序图如下

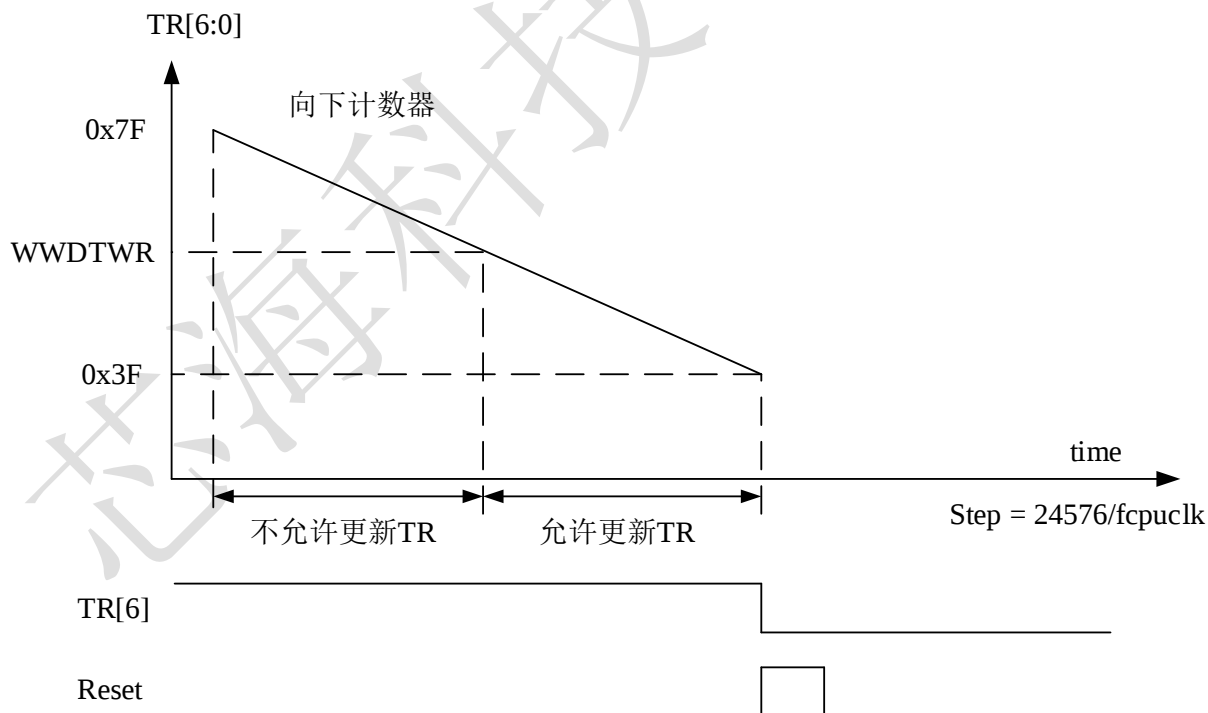


图 3-3 WWDT 时序图

不同指令周期下从当前 TR 值计数到 3Fh 对应的理论时间。

表 3-24

TR[6:0]	f _{CPU} (MHz)		
	24	12	6
40h	1.152ms	2.304ms	4.608ms
7Fh	73.755ms	147.51ms	294.12ms

3.4.5.4 软件复位功能

WWDT 可以作为软件复位功能使用，当 WWDT 计数值 TR[6:0]大于窗口值 WD[6:0]（WWDTWR 寄存器的值）时，对 WWDTCR 寄存器进行写操作将直接产生 MCU 复位信号将芯片复位。

3.4.5.5 WWDT 在 Halt 模式和 SLEEP 模式下的行为

在 SLEEP 模式下，WWDT 将自动停止计数，WWDT 也不会产生动态功耗。

WWDT 在 HALT 模式下的行为可以通过代码选项进行配置，可以配置为在 HALT 模式继续进行计数或者在 HALT 模式不进行计数。

3.4.6 寄存器映射

表 3-25 WWDT 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
基址 PAGE0, WWDT_BA = 0x44										
00h	WWDTCR	WWDTEN	TR[6:0]						01111111	
01h	WWDTWR	WD[6:0]						01111111		

3.4.7 寄存器描述

3.4.7.1 WWDTCR 寄存器（地址为 00h）

表 3-26

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
WWDTCR	WWDTEN	TR[6:0]						

表 3-27

位地址	标识符	功能
7	WWDTEN	WWDT 模块使能位 该位由软件置 1，只有在复位时硬件清零。只有在 WWDT 使能后才能产生复位。 0: WWDT 关闭 1: WWDT 使能
6:0	TR[6:0]	7 位计数值，默认值为 7Fh 该寄存器是 WWDT 计数器的计数值。它每 24576 个 CPU 时钟周期减 1。当计数值从 40H 跳变到 3Fh 时，将产生复位信号。软件可以写 WWDTCR 寄存器来更新 TR[6:0]的值，避免产生 MCU 复位。

3.4.7.2 WWDTWR 寄存器（地址为 01h）

表 3-28

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
WWDWTR		WD[6:0]						

表 3-29

位地址	标识符	功能
7	Reserved	保留
6:0	WD[6:0]	7 位窗口值，默认值为 7Fh 该寄存器值为 7 位窗口值，用来与计数器的计数值进行比较。

3.5 定时器 2

3.5.1 概述

定时器 2 主要实现 8bit 计数器功能。当选择内部 10KHz WDT 时钟进行计数时，可以唤醒 Sleep 模式。支持输出 PWM 信号。

3.5.2 特性

- 8 位可编程定时器
- Sleep 模式唤醒功能
- 中断上报功能
- PWM2 输出

3.5.3 框图

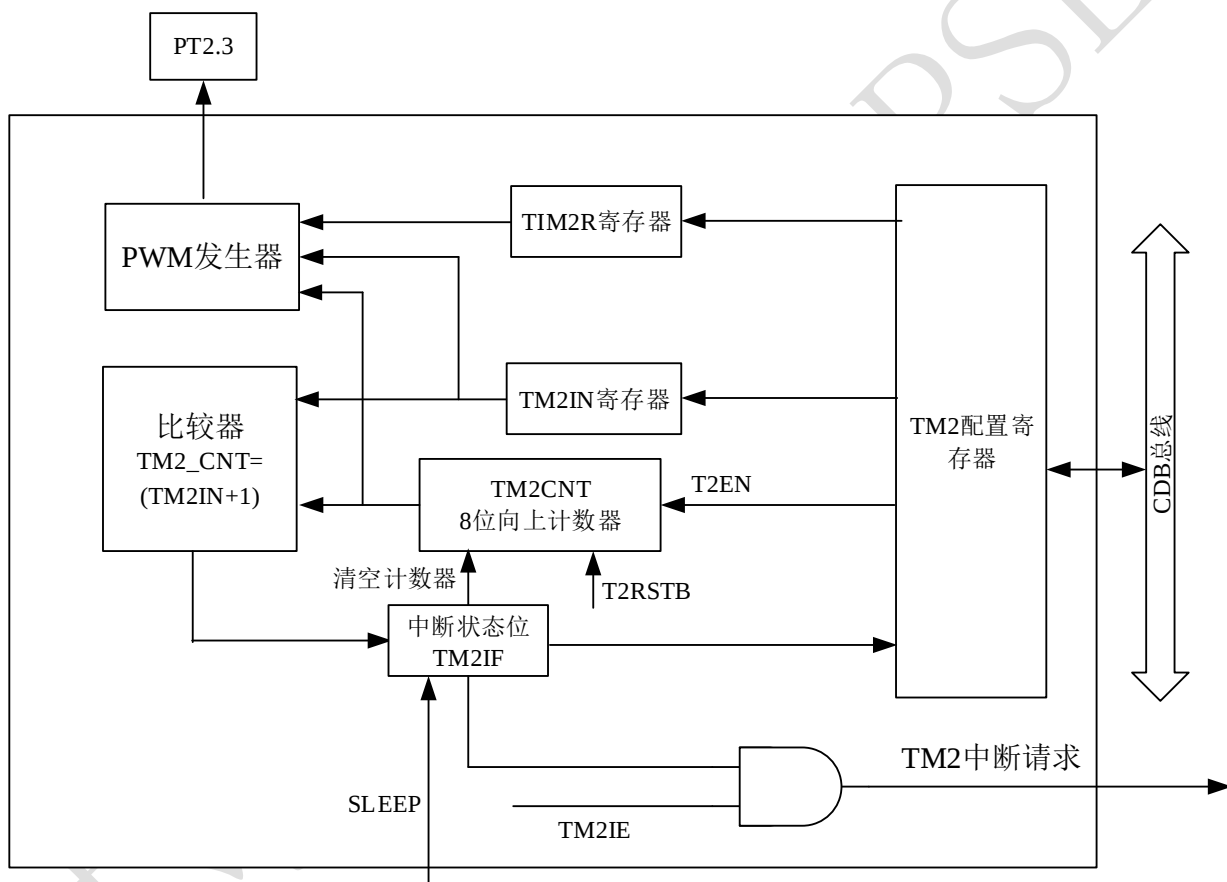


图 3-4 定时器 2 模块的功能框图

3.5.4 基本配置

通过配置 SYSCFG 模块的 CFGR4 寄存器的 PT23FUNC 位为 11，可将 PT2.3 配置为定时器 2 的 PWM 输出口。配置时钟管理单元的 CLKSEL0 寄存器的 T2CKS 信号选择定时器 2 时钟源，配置 CLKDIV1 寄存器 T2DIV 信号选择定时器 2 时钟分频。

3.5.5 功能描述

3.5.5.1 定时器 2 时钟

定时器 2 的时钟源可以选择 CPUCLK、SYSCLK、10KHz 时钟，当选择内部 10KHz WDT 时钟进行计数时，可以唤醒 Sleep 模式。

定时器 2 的时钟分频通过时钟管理单元进行配置，可对定时器 2 时钟源进行 1~128 分频，分频后作为定时器 2 的定时时钟 TM2CLK。

3.5.5.2 定时功能

定时/计数器 2 模块的输入时钟是 TM2CLK。当用户置位定时/计数器 2 模块的使能，8bits 计数器将启动，从 00h 递增到 TM2IN。用户需要设置 TM2IN（定时器模块计数溢出值）以选择定时超时时间，超时后将产生中断信号。

当定时超时发生时，定时器 2 中断标志位 TM2IF 硬件置 1，该位只能通过软件清零。如果使能了定时器 2 中断（TM2IE=1）和中断总使能（GIE 为内核寄存器信号），程序计数器会跳转到 004H 以执行中断服务程序。

定时器 2 溢出时间计算方法：

定时器 2 溢出时间 = (TM2IN[7:0] + 1) / TM2CLK. (TM2IN 不为 0)

3.5.5.3 定时功能配置步骤

配置定时功能步骤如下：

- 1) 配置 CLKMU 模块的 CLKSEL0 寄存器的 T2CKS[1:0]，为定时器模块选择时钟源。
- 2) 配置 CLKMU 模块 CLKDIV1 寄存器的 T2DIV[2:0]，为定时器 2 选择时钟分频。
- 3) 设置 TM2IN，选择定时器溢出值
- 4) 清除定时器中断标志位：将 TM2CON 寄存器的 TM2IF 位清零。
- 5) 置位寄存器位 TM2IE 与 GIE，使能定时器中断。
- 6) 清零寄存器位 T2RSTB，复位定时器模块的计数器。
- 7) 置位寄存器位 T2EN，使能定时器模块的 8bits 计数器。
- 8) 当定时超时发生时，程序计数器会跳转到 004H。

3.5.5.4 PWM

配置 PWM 输出步骤：

- 1) 配置 CLKMU 模块的 CLKSEL0 寄存器的 T2CKS[1:0]，为定时器模块选择时钟源。
- 2) 配置 CLKMU 模块 CLKDIV1 寄存器的 T2DIV[2:0]，为定时器 2 选择时钟分频。
- 3) 设置 TM2IN[7:0] 来配置 PWM2 的周期。
- 4) 设置 TM2R[7:0] 来配置 PWM2 的高电平的脉宽。
- 5) T2EN 置 1 启动定时器。

周期为 TM2IN+1，高电平脉宽为 TM2R。如 TM2IN=0x0F，TM2R=0x03 的 PWM2 波形输出如下：

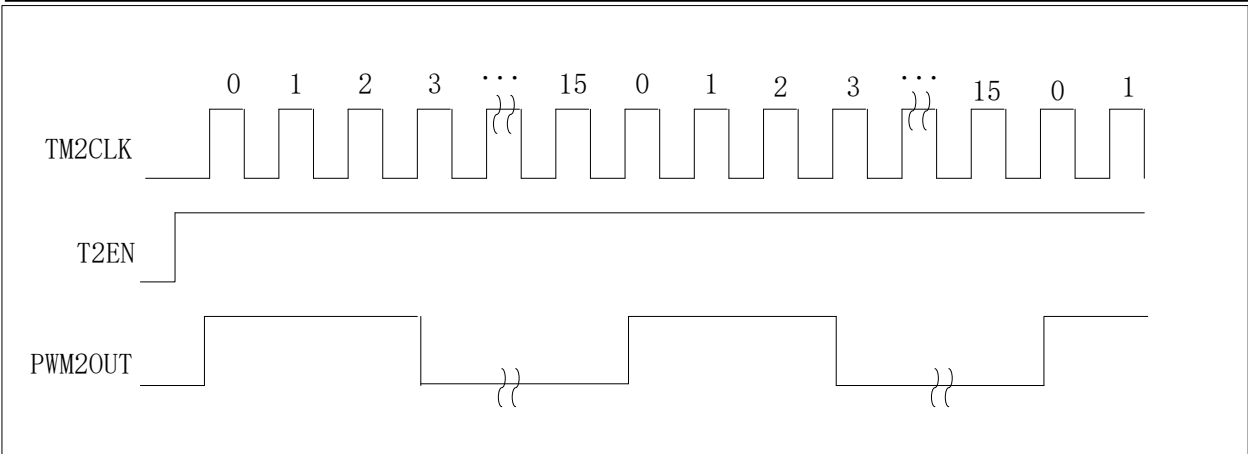


图 3-5 PWM2 输出波形图

3.5.6 寄存器映射

表 3-30 定时器寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
基址 PAGE0, TM2_BA = 0x70										
00h	TM2CON	T2EN	TM2IF	TM2IE	PWM2STALL		T2RSTB			00000100
01h	TM2IN	TM2IN[7:0]								11111111
02h	TM2CNT	TM2CNT[7:0]								00000000
03h	TM2R	TM2R[7:0]								00000000

3.5.7 寄存器描述

3.5.7.1 TM2CON 寄存器（地址 00h）

表 3-31

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-1	U-0	U-0
TM2CON	T2EN	TM2IF	TM2IE	PWM2STALL		T2RSTB		

表 3-32

位地址	标识符	功能
7	T2EN	定时/计数器 2 使能位 1: 使能定时器 2 0: 禁止定时器 2
6	TM2IF	8-Bit 定时/计数器 2 中断标志，软件写 0 清除，硬件置高 1 = 发生定时中断，必须软件清 0 0 = 没发生定时中断
5	TM2IE	8-Bit 定时/计数器 2 中断使能 1 = 使能定时/计数器 2 中断 0 = 不使能定时/计数器 2 中断
4	PWM2STALL	ICD 调试 stall 时，定时器 2 单端 PWM 输出电平控制位 1: ICD 调试 stall 时，定时器 2 单端 PWM 输出高电平

		0: ICD 调试 stall 时, 定时器 2 单端 PWM 输出低电
3	RESERVE	保留
2	T2RSTB	定时/计数器 2 复位 1: 禁止定时/计数器 2 复位 0: 使能定时/计数器 2 复位 当将该位为 0 时, 定时器 2 复位后, T2RSTB 会自动置 1
1:0	RESERVE	保留

3.5.7.2 TM2IN 寄存器 (地址 01h)

表 3-33

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TM2IN	TM2IN[7:0]							

表 3-34

位地址	标识符	功能
7: 0	TM2IN[7:0]	定时/计数器溢出值

3.5.7.3 TM2CNT 寄存器 (地址 02h)

表 3-35

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TM2CNT	TM2CNT[7:0]							

表 3-36

位地址	标识符	功能
7: 0	TM2CNT[7:0]	定时/计数器 2 计数寄存器, 只读

3.5.7.4 TM2R 寄存器 (地址 03h)

表 3-37

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TM2R	TM2R[7:0]							

表 3-38

位地址	标识符	功能
7: 0	TM2R[7:0]	定时/计数器 2 的 PWM 高电平占空比控制寄存器

3.6 定时器 3

3.6.1 概述

定时器 3 主要实现 12bit 计数器功能。支持输出 PWM 信号和互补式 PWM 信号。

3.6.2 特性

- 12 位可编程定时器
- 中断上报功能
- PWM 输出
- 互补式 PWM 信号输出

3.6.3 框图

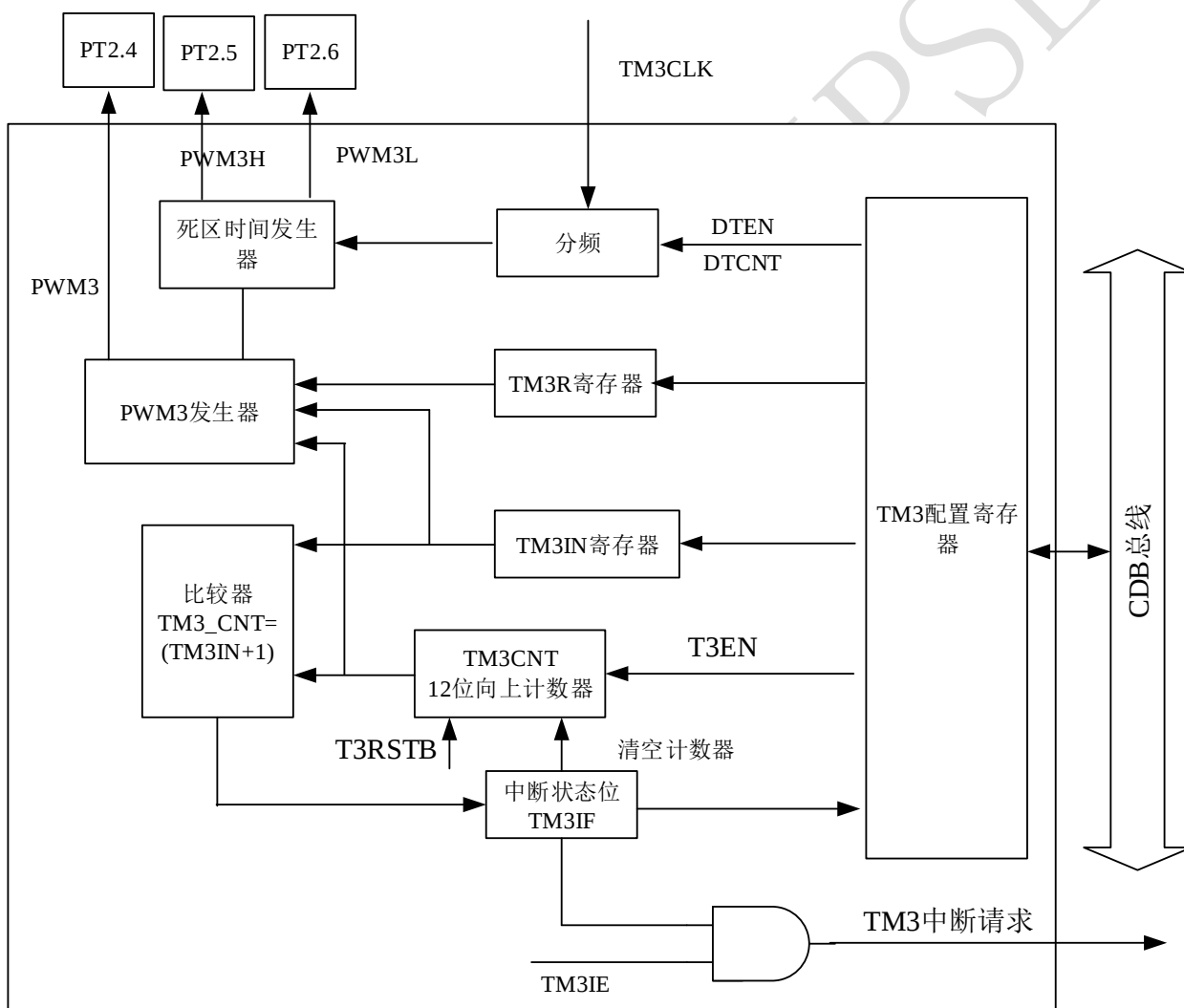


图 3-6 定时器 3 模块的功能框图

3.6.4 基本配置

通过配置 SYSCFG 模块的 CFGR5 寄存器的 PT25FUNC / PT26FUNC，可将 PT2.5、PT2.6 配置为定时器 3 的互补 PWM3H/PWM3L 输出。

通过配置 SYSCFG 模块的 CFGR5 寄存器 PT24FUNC 位，可将 PT2.4 配置为 PWM3 输出。

通过配置时钟管理单元的 CLKSEL0 寄存器的 T3CKS 位选择定时器 3 时钟源，配置 CLKDIV1 寄存器 T3DIV 信号选择定时器 3 时钟分频。

3.6.5 功能描述

3.6.5.1 定时器 3 时钟

定时器 3 的时钟源可以选择 CPUCLK、SYSCLK。

定时器 3 的时钟分频通过配置时钟管理单元 CLKDIV1 寄存器的 T3DIV 位进行配置，可对定时器 3 时钟源进行 1~128 分频，分频后作为定时器 3 的定时时钟 TM3CLK。

3.6.5.2 定时/计数器功能

定时/计数器 3 模块的输入时钟是 TM3CLK。当用户置位定时/计数器 3 模块的使能，12 bits 计数器将启动，从 000h 递增到 TM3IN。用户需要设置 TM3IN（定时器模块计数溢出值）以选择定时超时时间，超时后将产生中断信号。

当定时超时发生时，定时器 3 中断标志位 TM3IF 硬件置 1，该位只能通过软件清零。如果使能了定时器 3 中断（TM3IE=1）和中断总使能（GIE 为内核寄存器信号），程序计数器会跳转到 004H 以执行中断服务程序。

定时器 3 的时钟源可以选择 CPUCLK、SYSCLK，在 HALT 模式下定时器 3 仍然可以继续计数，因此可以唤醒 HALT 模式。SLEEP 模式下内部高速振荡器停止工作，定时器 3 停止计数，因此无法唤醒 SLEEP 模式。

定时器 3 溢出时间计算方法：

定时器 3 溢出时间 = (TM3IN[11:0]+1) / TM3CLK. (TM3IN 不为 0)

3.6.5.3 定时/计数功能配置步骤

定时/计数器操作：

- 1) 配置 CLKMU 模块的 CLKSEL0 寄存器的 T3CKS[1:0]，为定时器模块选择时钟源。
- 2) 配置 CLKMU 模块 CLKDIV1 寄存器的 T3DIV[2:0]，为定时器 3 选择时钟分频。
- 3) 设置 TM3IN[11:0]，选择定时器溢出值。
- 4) 清除定时器中断标志位：将 TM3CON 寄存器的 TM3IF 位清零
- 5) 置位寄存器位 TM3IE 与 GIE，使能定时器中断。
- 6) 清零寄存器位 T3RSTB，复位定时器模块的计数器。
- 7) 置位寄存器位 T3EN，使能定时器模块的 12 bits 计数器。
- 8) 当定时超时发生时，程序计数器会跳转到 004H。

3.6.5.4 PWM 输出

配置 PWM 输出操作：

- 1) 把 PT2.4 配置为输出口。
- 2) 配置 CLKMU 模块的 CLKSEL0 寄存器的 T3CKS[1:0]，为定时器模块选择时钟源。
- 3) 配置 CLKMU 模块 CLKDIV1 寄存器的 T3DIV[2:0]，为定时器 3 选择时钟分频。
- 4) 设置 TM3IN 来配置 PWM3 的周期。
- 5) 设置 TM3R 来配置 PWM3 的高电平的脉宽。
- 6) T3EN 置 1 启动定时器。
- 7) PWM3 从 PT2.4 输出。

周期为 $TM3IN+1$ ，高电平脉宽为 $TM3R$ 。如 $TM3IN=0x0F$ ， $TM3R=0x03$ 的 PWM3 波形输出如下：

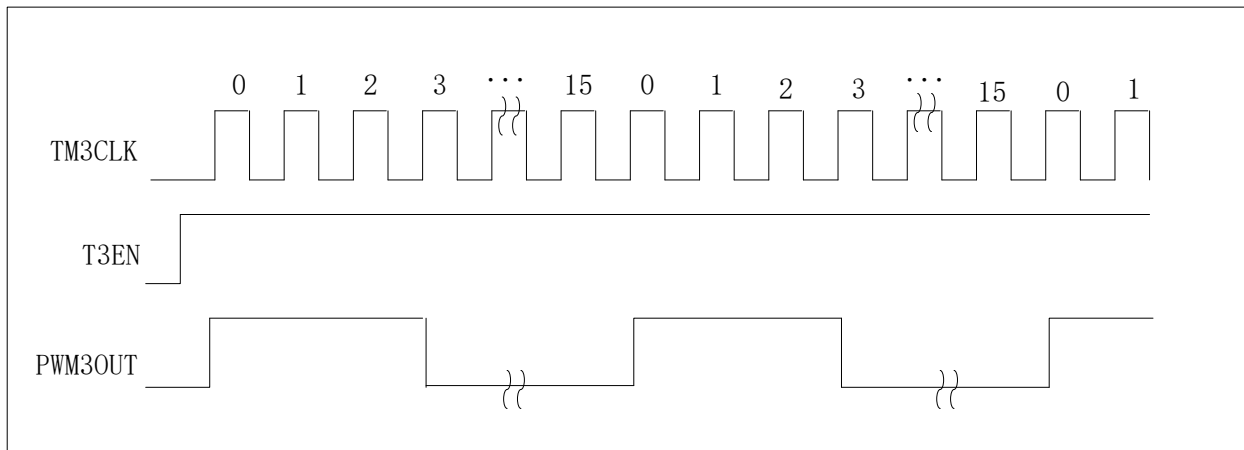


图 3-7 PWM3 输出波形图

3.6.5.5 互补式 PWM 输出

提供一对源于定时器 3 的互补式输出，可用作 PWM 驱动信号。对于 PMOS 管驱动，PWM 输出为低电平有效，而对于 NMOS 管驱动，PWM 输出为高电平有效。当这对互补式输出同时用于驱动 PMOS 和 NMOS 时，死区时间发生器插入一死区时间以防止直流电流过大，该死区时间可通过 $TM3CON2$ 寄存器的 $DT3CK[1:0]$ 和 $DT3CNT[2:0]$ 位来定义。在死区时间发生器输入信号的每个上升沿插入一个死区时间。通过死区插入电路，输出信号最终发送至外部功率晶体管。

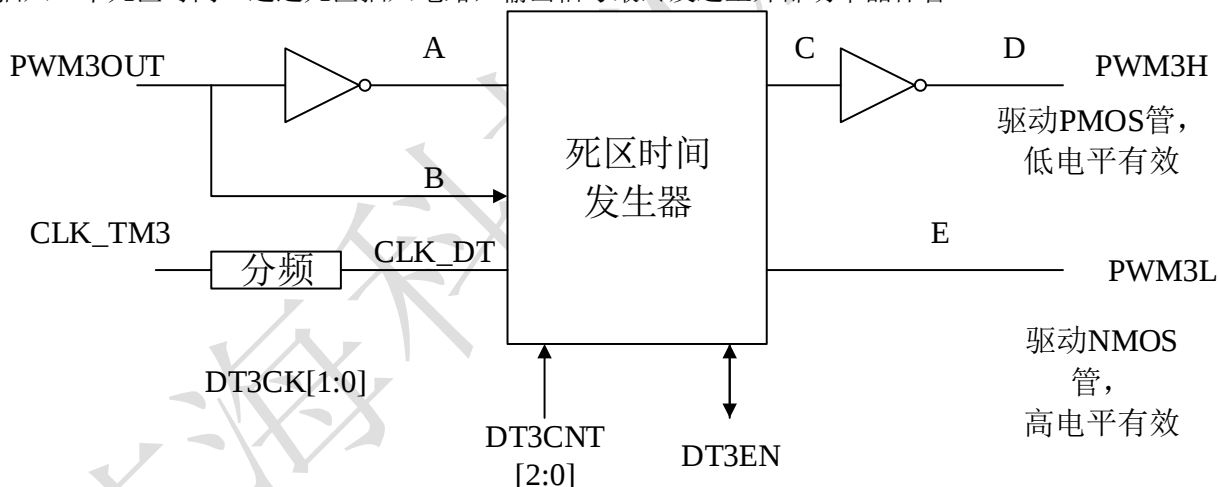


图 3-8 互补式 PWM 输出方框图

互补式 PWM 输出波形

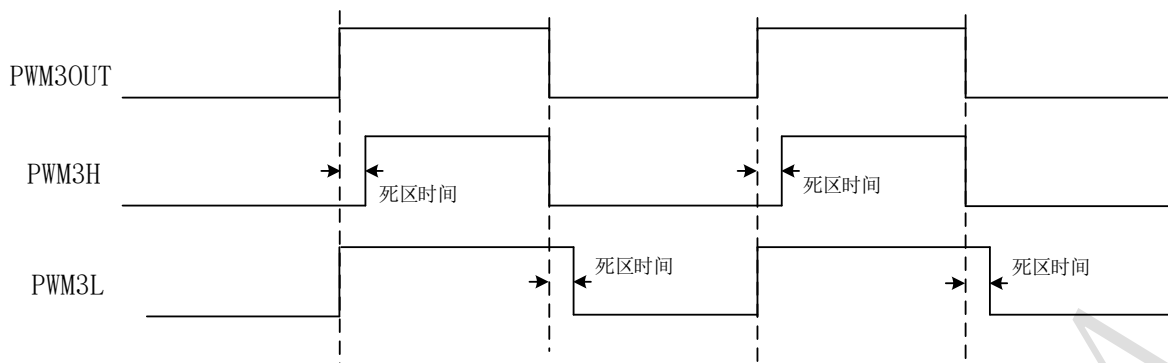


图 3-9 互补式 PWM 输出波形图

PWM 输出取反后的互补 PWM 输出

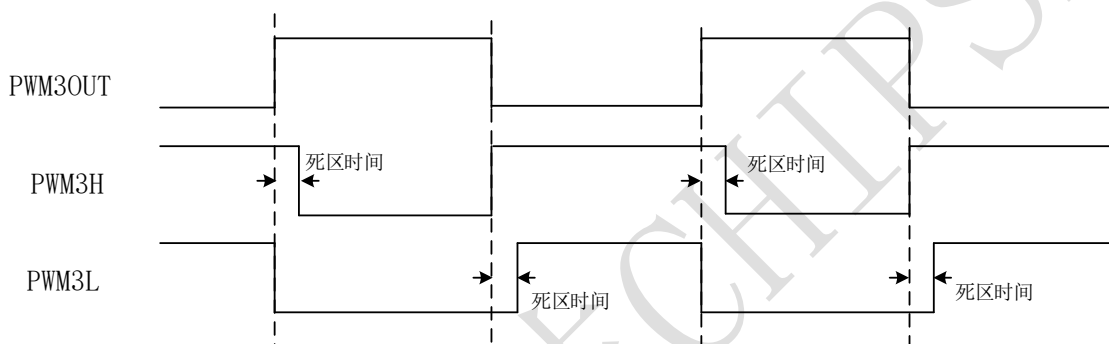


图 3-10 互补式 PWM 输出波形图（取反）

3.6.6 寄存器映射

表 3-39 定时器寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位 值
基址 PAGE1, TM3_BA = 0x10										
00h	TM3CON	T3EN	TM3IF	TM3IE			T3RSTB			00000100
01h	TM3IN	TM3IN[7:0]								11111111
02h	TM3INH					TM3IN[11:8]				00001111
03h	TM3CNT	TM3CNT[7:0]								00000000
04h	TM3CNTH					TM3CNT[11:8]				00000000
05h	TM3R	TM3R[7:0]								00000000
06h	TM3RH					TM3R[11:8]				00000000
07h	TM3CON2	PWM3STALL	DT3CNT[3:0]				DT3_EN	P3HINV	P3LINV	00000000

3.6.7 寄存器描述

3.6.7.1 TM3CON 寄存器（地址为 00H）

表 3-40

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-1	U-0	U-0
TM3CON	T3EN	TM3IF	TM3IE			T3RSTB		

表 3-41

位地址	标识符	功能
7	T3EN	定时/计数器 3 使能位 1: 使能定时器 3 0: 禁止定时器 3
6	TM3IF	12-Bit 定时/计数器 3 中断标志, 软件写 0 清除, 硬件置高 1 = 发生定时中断, 必须软件清 0 0 = 没发生定时中断
5	TM3IE	12-Bit 定时/计数器 3 中断使能 1 = 使能定时/计数器 3 中断 0 = 不使能定时/计数器 3 中断
4:3	RESERVE	保留
2	T3RSTB	定时/计数器 3 复位 1: 禁止定时/计数器 3 复位 0: 使能定时/计数器 3 复位 当将该位配置为 0, 一个指令周期后定时器 3 复位完成, T3RSTB 硬件置 1
1:0	RESERVE	保留

3.6.7.2 TM3IN 寄存器 (地址为 01H)

表 3-42

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TM3IN	TM3IN[7:0]							

表 3-43

位地址	标识符	功能
7 : 0	TM3IN[7:0]	定时/计数器溢出值低 8 位

3.6.7.3 TM3INH 寄存器 (地址为 02H)

表 3-44

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
TM3INH					TM3IN[11:8]			

表 3-45

位地址	标识符	功能
7 : 4	RESERVE	保留
3 : 0	TM3IN[11:8]	定时/计数器溢出值高 4 位

3.6.7.4 TM3CNT 寄存器 (地址为 03H)

表 3-46

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TM3CNT	TM3CNT[7:0]							

表 3-47

位地址	标识符	功能
7 : 0	TM3CNT[7:0]	定时/计数器 3 计数寄存器低 8 位，只读

3.6.7.5 TM3CNTH 寄存器（地址为 04H）

表 3-48

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
TM3CNTH	TM3CNT[11:8]							

表 3-49

位地址	标识符	功能
7 : 4	RESERVE	保留
3 : 0	TM3CNT[11:8]	定时/计数器 3 计数寄存器高 4 位，只读

3.6.7.6 TM3R 寄存器（地址为 05h）

表 3-50

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TM3R	TM3R[7:0]							

表 3-51

位地址	标识符	功能
7 : 0	TM3R[7:0]	定时/计数器 3 的 PWM 高电平占空比控制寄存器低 8 位

3.6.7.7 TM3RH 寄存器（地址为 06h）

表 3-52

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
TM3RH					TM3R[11:8]			

表 3-53

位地址	标识符	功能
7 : 4	RESERVE	保留
3 : 0	TM3R[11:8]	定时/计数器 3 的 PWM 高电平占空比控制寄存器高 4 位

3.6.7.8 TM3CON2 寄存器（地址为 07h）

表 3-54

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TM3CON2	PWM3STALL	DT3CNT[3:0]				DT3_EN	P3HINV	P3LINV

表 3-55

位地址	标识符	功能
7	PWM3STALL	ICD 调试 stall 时，定时器 3 单端 PWM 输出电平控制位 1: ICD 调试 stall 时，定时器 3 单端 PWM 输出高电平 0: ICD 调试 stall 时，定时器 3 单端 PWM 输出低电平
6:3	DT3CNT[3:0]	死区时间选择 死区时间=DT3CNT[3:0]* CLK_TM3
2	DT3_EN	死区发生器 3 使能位 0: 不使能死区发生器 3 1: 使能死区发生器 3
1	P3HINV	互补 PWM3H 输出取反控制位 0: PWM3H 不取反输出 1: PWM3H 取反输出
0	P3LINV	互补 PWM3L 输出取反控制位 0: PWM3L 不取反输出 1: PWM3L 取反输出

3.7 定时器 4

3.7.1 概述

定时器 4 主要实现 12bit 计数器功能。支持输出 PWM 信号。

3.7.2 特性

- 12 位可编程定时器
- 中断上报功能
- PWM 输出

3.7.3 框图

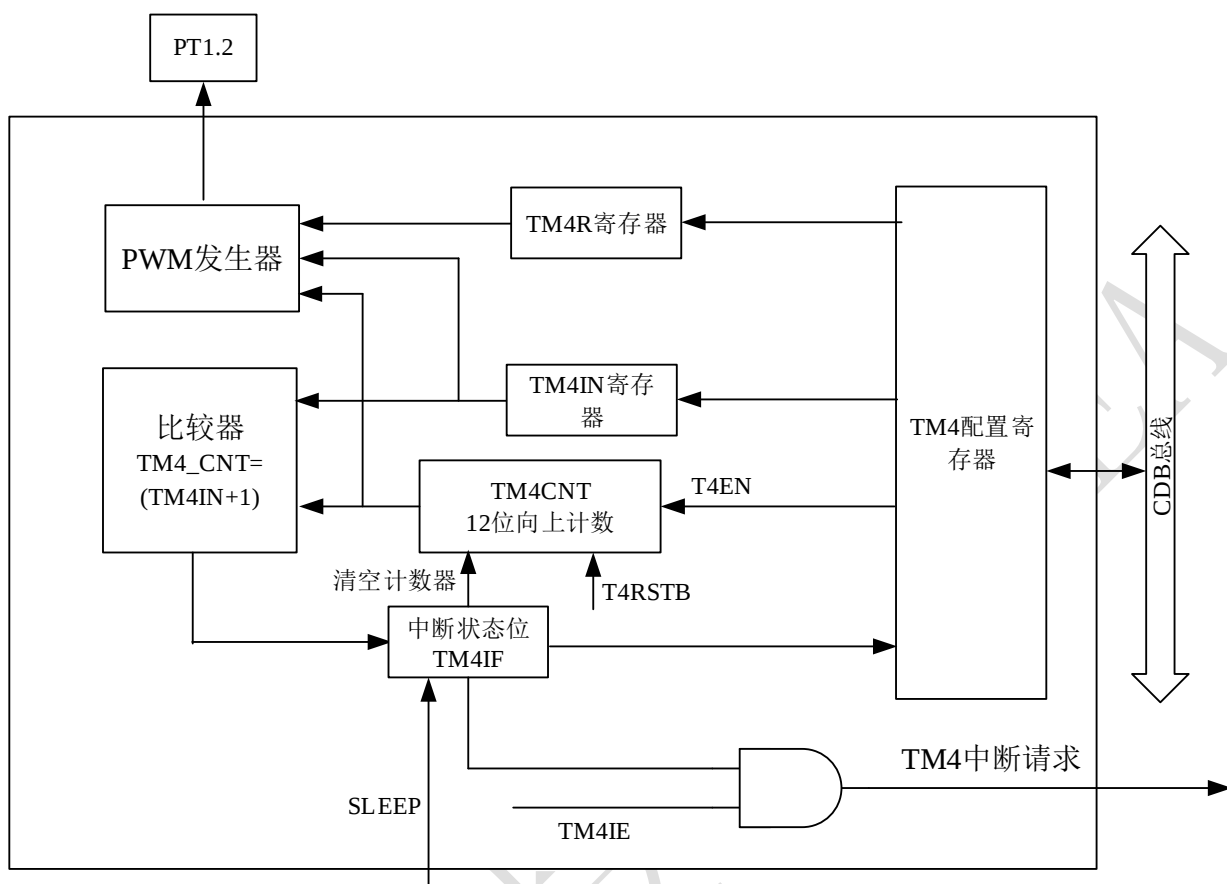


图 3-11 定时器 4 模块的功能框图

3.7.4 基本配置

通过配置时钟管理单元的 CLKSEL2 寄存器的 T4CKS 位选择定时器 4 时钟源，配置 CLKDIV3 寄存器 T4DIV 位选择定时器 4 时钟分频。

通过配置 SYSCFG 模块的 PT12FUNC[1:0] 为 10，可将 PT1.2 配置为定时器 4 的 PWM 输出口。

3.7.5 功能描述

3.7.5.1 定时器 4 时钟

定时器 4 的时钟源可以选择 CPUCLK、SYSCLK。

定时器 4 的时钟分频通过配置时钟管理单元 CLKDIV3 寄存器的 T4DIV 位进行配置，可对定时器 4 时钟源进行 1~128 分频，分频后作为定时器 4 的定时时钟 TM4CLK。

3.7.5.2 定时/计数器功能

定时器 4 的时钟源可以选择 CPUCLK、SYSCLK，在 HALT 模式下定时器 4 仍然可以继续计数，因此可以唤醒 HALT 模式。SLEEP 模式下内部高速振荡器停止工作，定时器 4 停止计数，因此无法唤醒 SLEEP 模式。

定时/计数器 4 模块的输入时钟是 TM4CLK。当用户置位定时/计数器 4 模块的使能，12 bits 计数器将启动，从 000h 递增到 TM4IN。用户需要设置 TM4IN（定时器模块计数溢出值）以选择定时超时时间，超时后将产生中断信号。

当定时超时发生时，定时器 4 中断标志位 TM4IF 硬件置 1，该位只能通过软件清零。如果使能了定时器 4 中断（TM4IE=1）和中断总使能（GIE 为内核寄存器信号），程序计数器会跳转到 004H 以执行中断服务程序。

定时器 4 溢出时间计算方法：

定时器 4 溢出时间 = (TM4IN[11:0]+1) / TM4CLK. (TM4IN 不为 0)

3.7.5.3 定时功能配置步骤

配置操作：

- 1) 配置 CLKMU 模块的 CLKSEL2 寄存器的 T4CKS[1:0]，为定时器模块选择时钟源。
- 2) 配置 CLKMU 模块 CLKDIV3 寄存器的 T4DIV[2:0]，为定时器 4 选择时钟分频。
- 3) 设置 TM4IN[11:0]，选择定时器溢出值
- 4) 清除定时器中断标志位：将 TM4CON 寄存器的 TM4IF 位清零。
- 5) 置位寄存器位 TM4IE 与 GIE，使能定时器中断。
- 6) 清零寄存器位 T4RSTB，复位定时器模块的计数器。
- 7) 置位寄存器位 T4EN，使能定时器模块的 12 bits 计数器。
- 8) 当定时超时发生时，程序计数器会跳转到 004H。

3.7.5.4 PWM

配置 PWM 输出步骤：

- 1) 配置 PT1.2 为数字输出口。
- 2) 配置 TM4CON 寄存器的低 2 位，T4OUT 设置为 0，PWM2OUT 设置为 1
- 3) 配置 CLKMU 模块的 CLKSEL2 寄存器的 T4CKS[1:0]，为定时器模块选择时钟源。
- 4) 配置 CLKMU 模块 CLKDIV3 寄存器的 T4DIV[2:0]，为定时器 4 选择时钟分频。
- 5) 设置 TM4IN[11:0] 来配置 PWM2 的周期。
- 6) 设置 TM4R[11:0] 来配置 PWM2 的高电平的脉宽。
- 7) 把 T4EN 置 1，启动定时器。
- 8) PWM 从 PT1.2 输出。

周期为 TM4IN+1，高电平脉宽为 TM4R。如 TM4IN=0x0F，TM4R=0x03 的 PWM4 波形输出如下：

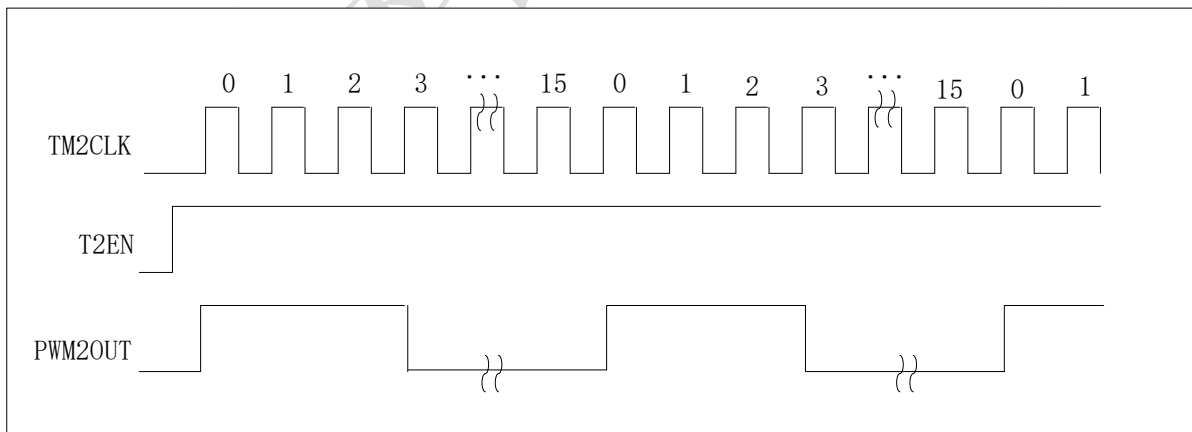


图 3-12 PWM4 输出波形图

3.7.6 寄存器映射

表 3-56 定时器寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
基址 PAGE1, TM4_BA = 0x48										
00h	TM4CON	T4EN	TM4IF	TM4IE	PWM4STALL		T4RSTB			00000100
01h	TM4IN	TM4IN[7:0]								11111111
02h	TM4INH						TM4IN[11:8]			00001111
03h	TM4CNT	TM4CNT[7:0]								00000000
04h	TM4CNTH						TM4CNT[11:8]			00000000
05h	TM4R	TM4R[7:0]								00000000
06h	TM4RH						TM4R[11:8]			00000000

3.7.7 寄存器描述

3.7.7.1 TM4CON 寄存器（地址 00h）

表 3-57

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-1	U-0	U-0
TM4CON	T4EN	TM4IF	TM4IE	PWM4STALL		T4RSTB		

表 3-58

位地址	标识符	功能
7	T4EN	定时/计数器 4 使能位 1: 使能定时器 4 0: 禁止定时器 4
6	TM4IF	12-Bit 定时/计数器 4 中断标志, 软件写 0 清除, 硬件置高 1 = 发生定时中断, 必须软件清 0 0 = 没发生定时中断
5	TM4IE	12-Bit 定时/计数器 4 中断使能 1 = 使能定时/计数器 4 中断 0 = 不使能定时/计数器 4 中断
4	PWM4STALL	ICD 调试 stall 时, 定时器 4 单端 PWM 输出电平控制位 1: ICD 调试 stall 时, 定时器 4 单端 PWM 输出高电平 0: ICD 调试 stall 时, 定时器 4 单端 PWM 输出低电平
3	RESERVE	保留
2	T4RSTB	定时/计数器 4 复位 1: 禁止定时/计数器 4 复位 0: 使能定时/计数器 4 复位 当将该位为 0 时, 定时器 4 复位后, T4RSTB 会自动置 1
1:0	RESERVE	保留

3.7.7.2 TM4IN 寄存器（地址 01h）

表 3-59

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TM4IN	TM4IN[7:0]							

表 3-60

位地址	标识符	功能
7 : 0	TM4IN[7:0]	定时/计数器溢出值低 8 位

3.7.7.3 TM4INH 寄存器（地址 02h）

表 3-61

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
TM4INH	TM4IN[11:8]							

表 3-62

位地址	标识符	功能
3 : 0	TM4IN[11:8]	定时/计数器溢出值高 4 位

3.7.7.4 TM4CNT 寄存器（地址 03h）

表 3-63

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
TM4CNT	TM4CNT[7:0]							

表 3-64

位地址	标识符	功能
7 : 0	TM4CNT[7:0]	定时/计数器 4 计数寄存器低 8 位，只读

3.7.7.5 TM4CNTH 寄存器（地址 04h）

表 3-65

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
TM4CNTH	TM4CNT[11:8]							

表 3-66

位地址	标识符	功能
3 : 0	TM4CNT[11:8]	定时/计数器 4 计数寄存器高 4 位，只读

3.7.7.6 TM4R 寄存器（地址 05h）

表 3-67

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
TM4R	TM4R[7:0]							

表 3-68

位地址	标识符	功能
7 : 0	TM4R[7:0]	定时/计数器 4 的 PWM 高电平占空比控制寄存器低 8 位。

3.7.7.7 TM4RH 寄存器（地址 06h）

表 3-69

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
TM4RH					TM4R[11:8]			

表 3-70

位地址	标识符	功能
3 : 0	TM4R[11:8]	定时/计数器 4 的 PWM 高电平占空比控制寄存器 高 4 位。

3.8 模拟比较器 (ACMP)

3.8.1 概述

CSU3AF10 有 2 个比较器，可以在不同的配置下使用。当正端输入大于负端输入时，比较器输出逻辑”1”，否则输出”0”。当比较器输出值改变，每个比较器可以配置发生中断。

3.8.2 特性

- 模拟输入电压范围: $0 \sim AV_{DD}$
- 支持迟滞功能
- 支持 8 档输出滤波功能
- 支持输出反转功能
- 每个模拟比较器正/负端可以选择输入内部参考电压
- 支持失调电压校准功能
- 每个比较器支持一个中断向量

3.8.3 框图

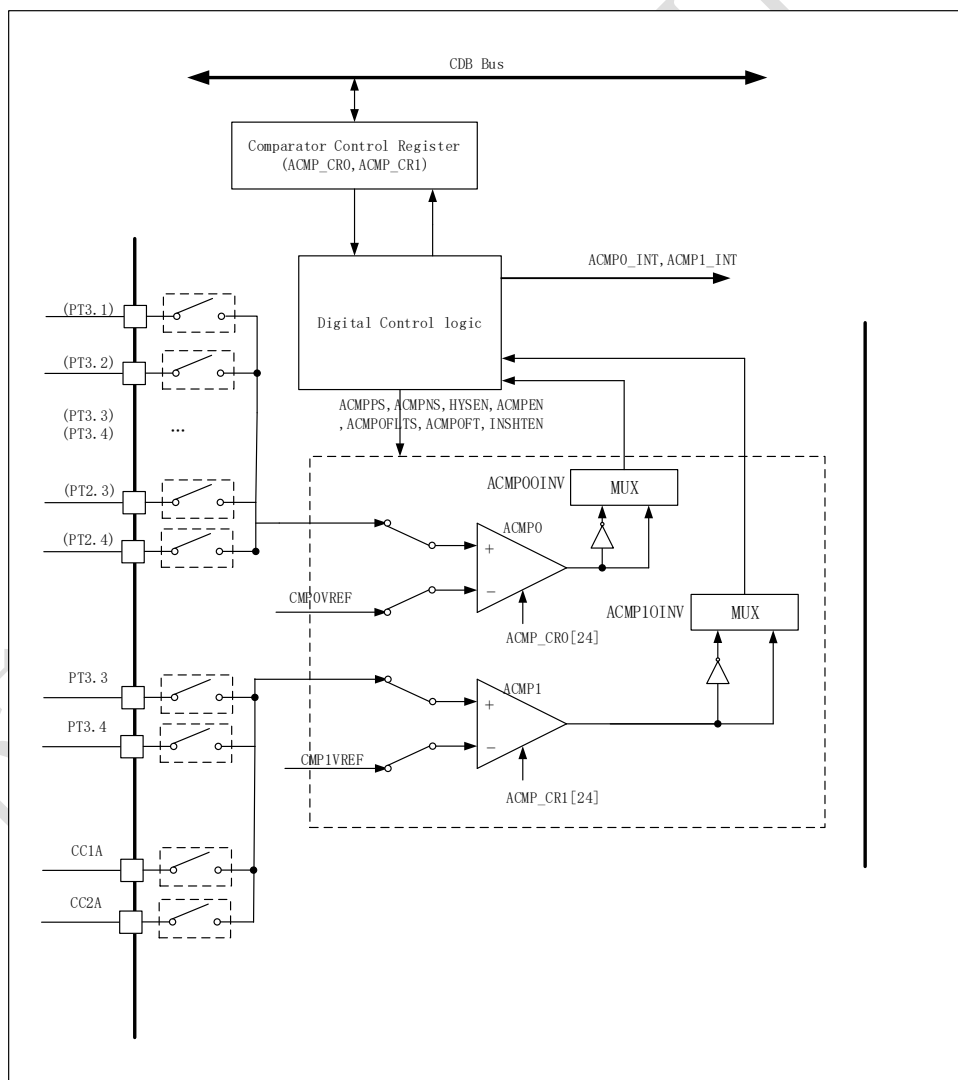


图 3-13 模拟比较器 0/1 框图

3.8.4 基本配置

ACMP 引脚功能使用寄存器 PTFUN 来配置。当作为一般比较功能时，将 PTFUN 配置为模拟功能，这样数字通路会被关闭。当引脚用于 DFP 协议检测时，数字输入通道可以和比较器同时打开，来实现协议的并行检测。

3.8.5 功能描述

3.8.5.1 中断源

比较器的输出由 CPUCLK 采样，反映在 ACMP_CR 寄存器的 ACMPxO 位(x=0,1)。如果寄存器 ACMP_CRx 中的 ACMPxIE 设为 1，当比较器输出的值改变将导致比较器标志位 ACMPFx 被设，比较器中断将发生。软件可以写 1 到 ACMPxF 来使该位清 0。

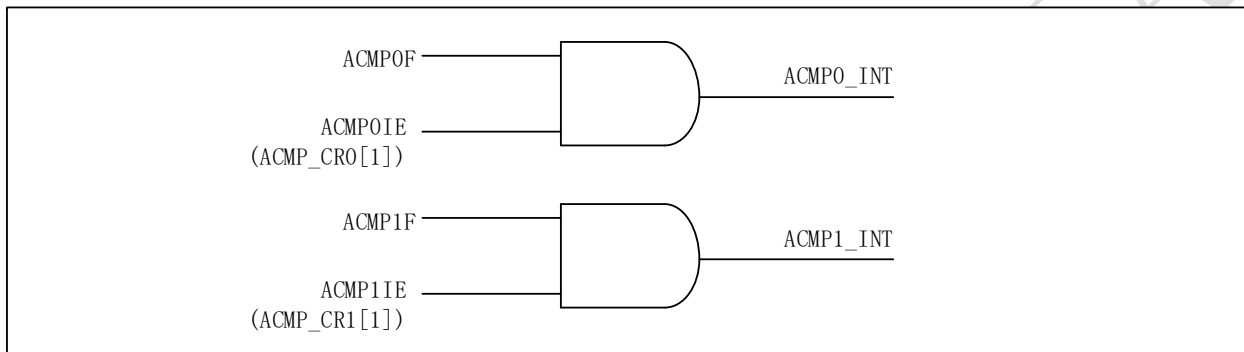


图 3-14 比较器中断源

3.8.5.2 迟滞功能

模拟比较器提供迟滞功能使比较输出转变更稳定。如果当前比较器输出为 0，正端输入电压超过负端输入电压大于迟滞电压，比较器输出才会变成 1。相似的，如果当前比较器输出为 1，负端输入电压低于正端输入电压大于迟滞电压，比较器输出才会变成 0。

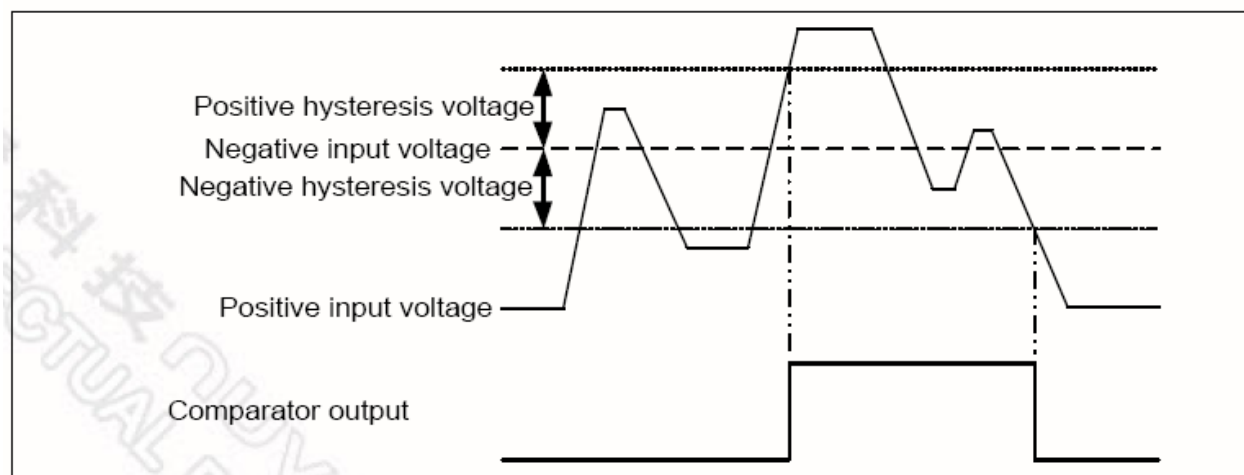


图 3-15 比较器迟滞功能

3.8.5.3 滤波功能

模拟比较器提供滤波功能使比较器输出转变更稳定。如果系统不稳定，或受到干扰，模拟比较器可能会出现异常的窄脉宽，为了对窄脉宽进行数字滤波，比较器提供 8 档的滤波选择功能，有 ACMPOFLTS[2:0]进行选择控制。当比较器输出的窄脉宽小于滤波选择的时间，则会被滤掉。

3.8.5.4 失调电压校准功能

考虑到不同工艺偏差, 和不同芯片之间的差异, 模拟比较器的失调电压具有一定的离散型或差异性。为了保证每个芯片模拟比较器失调电压的一致性, 和减少失调电压, 模拟比较器提供失调电压校准功能。

失调电压校准方式如下:

- (1) HYSEN=0, 禁止迟滞功能; ACMPOINV =0, 禁止输出反转; ACMPOFLT[2:0]=3'b010 设置滤波时间。
- (2) INSHTEN=2'b01, 使能模拟比较器内短功能。
- (3) ACMPEN=1, 使能模拟比较器。
- (4) ACMPOFT[4:0]=5'b10000。
- (5) 等待 5us, 之后判断 模拟比较器的输出结果 ACMPO。
 - a) 如果输出为 1, 则从 10000 到 11111 增加 ACMPOFT[4:0] 的值, 每次更改校准之后等待 5us; 若在校准过程中出现比较器输出为 0, 减小 ACMPOFT[4:0], 以求最接近的值。
 - b) 如果输出为 0, 则从 00000 到 01111 增加 ACMPOFT[4:0] 的值, 每次更改校准之后等待 5us; 若在校准过程中出现比较器输出为 1, 减小 ACMPOFT[4:0], 以求最接近的值。

3.8.6 寄存器描述

表 3-71 ACMP0 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
基址 PAGE1, ACMP0_BA = 0x58										
00h ₁	ACMP0CR	CAL_CNT0[5:0]					CAL_FT0	CALEN0	01111000	
01h ₂	ACMP0CR	INSHTEN0[1:0]		WEN0	ACMP0OFT[4:0]				00010000	
02h ₃	ACMP0CR	ACMP0PS[3:0]			ACMP0O	ACMP0NS[2:0]				00000000
03h ₄	ACMP0CR	ACMP0F	ACMP0OINV	ACMP0OFLT[2:0]		HYSEN0	ACMP0IE	ACMP0EN	00000000	

3.8.6.1 ACMP0CR1 寄存器 (地址 00h)

表 3-72

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0
ACMP0CR1	CAL_CNT0[5:0]						CAL_FT0	CALEN0

表 3-73

位地址	标识符	功能
7: 2	CAL_CNT0[5:0]	比较器 0 失调电压校准时间常数设置, 需要设置为大于 5us。时间计算公式如下: $CAL_CNT0[5:0] \times 4 \times T_{PCLK}$
1	CAL_FT0	比较器 0 失调电压校准失败标志位, 只读 1: 失调电压校准失败 0: 失调电压校准成功
0	CALEN0	比较器 0 失调电压校准使能控制

		0 = 已完成比较器 0 失调电压校准过程 1 = 使能比较器 0 失调电压校准，完成校准后该位会自动清 0
--	--	---

3.8.6.2 ACMP0CR2 寄存器（地址 01h）

表 3-74

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
ACMP0CR2	INSHTEN0[1:0]		WEN0	ACMP0OFT[4:0]				

表 3-75

位地址	标识符	功能
7: 6	INSHTEN0[1:0]	比较器 0 内短使能控制 00 = 禁止比较器 0 内短 01 = 使能比较器 0 内短，即比较器正端和负端与外部信号断开，并且比较器正端和负端短接且连接到地 10 = 使能比较器 0 内短，即比较器负端分别与外部信号断开，并且比较器正端和负端短接且连接到比较器正端 11 = 使能比较器 0 内短，即比较器正端与外部信号断开，并且比较器正端和负端短接且连接到比较器负端
5	WEN0	比较器 0 ACMPOFT[4:0]写使能控制 0 =禁止 ACMPOFT[4:0]写操作 1 =使能 ACMPOFT[4:0]写操作，只有该位置 1 后，对 ACMPOFT[4:0]写操作才有效
4: 0	ACMP0OFT[4:0]	比较器 0 失调电压校准控制 00000 =失调电压不处理 00001=失调电压正偏 0.5mV 01101 = 失调电压正偏 6.5mV 01110 =失调电压正偏 7mV 01111 =失调电压正偏 7.5mV 10000 = 失调电压不处理 10001 = 失调电压负偏 0.5mV 10010 = 失调电压负偏 1.0mV 10011 = 失调电压负偏 1.5mV 11111 = 失调电压负偏 7.5mV 失调电压每档变化 0.5mV，省略的地址依次类推。

3.8.6.3 ACMP0CR3 寄存器（地址 02h）

表 3-76

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R - 0	R/W-0	R/W-0	R/W-0
ACMP0CR3	ACMP0PS[3:0]				ACMP0O	ACMP0NS[2:0]		

表 3-77

位地址	标识符	功能
7: 4	ACMP0PS[3:0]	比较器 0 正端输入选择 0000 = PT3.1 引脚输入 0001 = PT3.2 引脚输入 0010 = PT2.3 引脚输入 0011 = PT2.4 引脚输入 0100 = 内部参考电压 Int_V _{REF} 输入 0101 = 内部比较器 0 参考电压 CMP0_VREF 输入 0110 = 接地 其他: 保留
3	ACMP0O	比较器 0 的输出 这个 bit 显示当前比较器的比较结果，用户可以通过读这个 bit 查看比较器的输出。当比较器被关闭时(ACMP_CR0[0]= 0)此 bit 将被清 0。 0 = 比较器 0 输出 0 1 = 比较器 0 输出 1 如果 ACMP0OINV 置 1，则比较器输出结果取反，比较器关闭时，此位置 1。
2: 0	ACMP0NS[2:0]	比较器 0 负端输入选择 000 = PT3.0 引脚输入 001 = 无效 01x = 无效 100 = 内部参考电压 Int_V _{REF} 输入 101 = 内部比较器 0 参考电压 CMP0_VREF 输入 11x = 无效

3.8.6.4 ACMP0CR4 寄存器（地址 03h）

表 3-78

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ACMP0CR4	ACMP0F	ACMP0OINV	ACMP0OFLTS[2:0]			HYSEN0	ACMP0IE	ACMP0EN

表 3-79

位地址	标识符	功能
7	ACMP0F	比较器 0 标志位 当比较器 0 输出状态改变时，这个 bit 将被置位。如果 ACMP_CR4[1]使能，中断将发生。 0 = 比较器 0 输出没有发生改变 1 = 从上次该位清 0 之后，比较器 0 的输出发生了改变 注：这个 bit 写 1 清 0。
6	ACMP0OINV	比较器 0 输出反转使能控制 0 = 禁止比较器输出反转 1 = 使能比较器输出反转
5: 3	ACMP0OFLTS[2:0]	比较器 0 输出滤波时间控制 000 = 不进行滤波操作 001 = 滤波时间为 4 * T _{cpuclk} 010 = 滤波时间为 8 * T _{cpuclk}

		011 = 滤波时间为 $16 * T_{cpucclk}$ 100 = 滤波时间为 $32 * T_{cpucclk}$ 101 = 滤波时间为 $64 * T_{cpucclk}$ 110 = 滤波时间为 $128 * T_{cpucclk}$ 111 = 滤波时间为 $256 * T_{cpucclk}$
2	HYSEN0	比较器 0 迟滞 使能控制 1 = 使能迟滞功能 0 = 关闭迟滞功能
1	ACMP0IE	比较器 0 中断使能控制 1 = 使能中断 0 = 禁止中断功能 如果 ACMP0IE 等于 1，比较器转换结果发生反转时，中断将发生
0	ACMP0EN	比较器 0 使能控制 1 = 使能 0 = 禁止 注：ACMP0EN 置位以后，比较器输出需要等 2 us 的稳定时间

表 3-80 ACMP1 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
基址 PAGE1, ACMP0_BA = 0x5C										
00h	ACMP1CR1	CAL_CNT1[5:0]						CAL_FT1	CALEN1	01111000
01h	ACMP1CR2	INSHTEN1[1:0]		WEN1	ACMP1OFT[4:0]					00010000
02h	ACMP1CR3	ACMP1PS[3:0]				ACMP1O	ACMP1NS[2:0]			00000000
03h	ACMP1CR4	ACMP1F	ACMP1OINV	ACMP1OFLT[2:0]		HYSEN1	ACMP1IE	ACMP1EN	00000000	

3.8.6.5 ACMP1CR1 寄存器（地址 00h）

表 3-81

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0
ACMP1CR1	CAL_CNT1[5:0]						CAL_FT1	CALEN1

表 3-82

位地址	标识符	功能
7: 2	CAL_CNT1[5:0]	比较器 1 失调电压校准时间常数设置，需要设置为大于 5us。时间计算公式如下： $CAL_CNT1[5:0] \times 4 \times T_{PCLK}$
1	CAL_FT1	比较器 1 失调电压校准失败标志位，只读 1: 失调电压校准失败 0: 失调电压校准成功
0	CALEN1	比较器 1 失调电压校准使能控制 0 = 已完成比较器 1 失调电压校准过程 1 = 使能比较器 1 失调电压校准，完成校准后该位会自动清 0

3.8.6.6 ACMP1CR2 寄存器（地址 01h）

表 3-83

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
ACMP1CR2	INSHTEN1[1:0]		WEN1	ACMP1OFT[4:0]				

表 3-84

位地址	标识符	功能
7: 6	INSHTEN1[1:0]	比较器 1 内短使能控制 00 = 禁止比较器 1 内短 01 = 使能比较器 1 内短，即比较器正端和负端与外部信号断开，并且比较器正端和负端短接且连接到地 10 = 使能比较器 1 内短，即比较器负端分别与外部信号断开，并且比较器正端和负端短接且连接到比较器正端 11 = 使能比较器 1 内短，即比较器正端与外部信号断开，并且比较器正端和负端短接且连接到比较器负端
1	WEN1	比较器 1 ACMPOFT[4:0]写使能控制 0 = 禁止 ACMPOFT[4:0]写操作 1 = 使能 ACMPOFT[4:0]写操作，只有该位置 1 后，对 ACMPOFT[4:0]写操作才有效
4: 0	ACMP1OFT[4:0]	比较器 1 失调电压校准控制 00000 = 失调电压不处理 00001 = 失调电压正偏 0.5mV 01101 = 失调电压正偏 6.5mV 01110 = 失调电压正偏 7mV 01111 = 失调电压正偏 7.5mV 10000 = 失调电压不处理 10001 = 失调电压负偏 0.5mV 10010 = 失调电压负偏 1.0mV 10011 = 失调电压负偏 1.5mV 11111 = 失调电压负偏 7.5mV 失调电压每档变化 0.5mV，省略的地址依次类推。

3.8.6.7 ACMP1CR3 寄存器（地址 02h）

表 3-85

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ACMP1CR3	ACMP1PS[3:0]				ACMP1O	ACMP1NS[2:0]		

表 3-86

位地址	标识符	功能
7: 4	ACMP1PS[3:0]	比较器 1 正端输入选择 0000 = 无效 0001 = PT3.0 引脚输入

		0010 = 无效 0011 = LDO 电压 0100 = 内部 1V 基准输入 0101 = 内部比较器 1 参考电压 CMP1_VREF 输入 0110 = 接地 100x = PT3.1 101x = PT3.2 1110 = PT3.3 1111 = PT3.4 其他: 保留
3	ACMP1O	比较器 1 的输出 这个 bit 显示当前比较器的比较结果, 用户可以通过读这个 bit 查看比较器的输出。当比较器被关闭时(ACMP_CR1[0]= 0)此 bit 将被清 0。 0 = 比较器 1 输出 0 1 = 比较器 1 输出 1 如果 ACMP1OINV 置 1, 则比较器输出结果取反, 比较器关闭时, 此位置 1。
2: 0	ACMP1NS[2:0]	比较器 1 负端输入选择 000 = PT3.0 引脚输入 001 = PT2.6 引脚输入 01x = 无效 100 = 内部参考电压 Int_VREF 输入 101 = 内部比较器 1 参考电压 CMP1_VREF 输入 11x = 无效

3.8.6.8 ACMP1CR4 寄存器 (地址 03h)

表 3-87

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ACMP1CR4	ACMP1F	ACMP1OINV	ACMP1OFLTS[2:0]			HYSEN1	ACMP1IE	ACMP1EN

表 3-88

位地址	标识符	功能
7	ACMP1F	比较器 1 标志位 当比较器 1 输出状态改变时, 这个 bit 将被置位。如果 ACMP_CR1[1]使能, 中断将发生。 0 = 比较器 1 输出没有发生改变 1 = 从上次该位清 0 之后, 比较器 1 的输出发生了改变 注: 这个 bit 写 1 清 0。
6	ACMP1OINV	比较器 1 输出反转使能控制 0 = 禁止比较器输出反转 1 = 使能比较器输出反转
5: 3	ACMP1OFLTS[2:0]	比较器 1 输出滤波时间控制 000 = 不进行滤波操作 001 = 滤波时间为 4 * Tcpuclk 010 = 滤波时间为 8 * Tcpuclk 011 = 滤波时间为 16 * Tcpuclk 100 = 滤波时间为 32 * Tcpuclk

		101 = 滤波时间为 $64 * T_{cpuclk}$ 110 = 滤波时间为 $128 * T_{cpuclk}$ 111 = 滤波时间为 $256 * T_{cpuclk}$
2	HYSEN1	比较器 1 迟滞 使能控制 1 = 使能迟滞功能 0 = 关闭迟滞功能
1	ACMP1IE	比较器 1 中断使能控制 1 = 使能中断 0 = 禁止中断功能 如果 ACMP1IE 等于 1，比较器转换结果发生反转时，中断将发生
0	ACMP1EN	比较器 1 使能控制 1 = 使能 0 = 禁止，PA.0，PA.1，PA.2 不作为比较器用途的引脚 注：ACMP1EN 置位以后，比较器输出需要等 2 us 的稳定时间

3.9 模数转换器（ADC）

3.9.1 概述

提供一个 SAR_ADC 数模转换器，支持 17 路模拟输入选择。

3.9.2 特性

- 支持 17 路 AIN 模拟输入
- 支持差分输入
- 支持中断
- 支持校正控制

3.9.3 功能框图

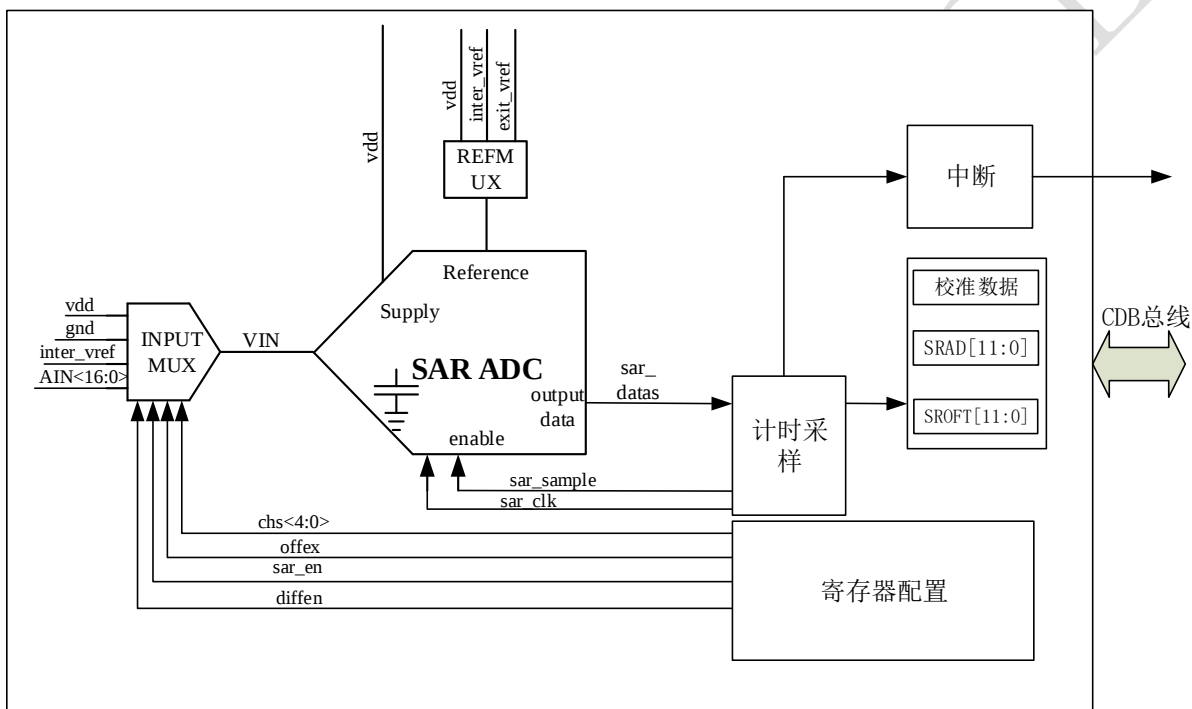


图 3-16 模数转换器 ADC 功能框图

模数转换模块共用 17 个外部通道（AIN0~AIN16）和 3 个内部通道（VDD/8，内部参考电压和 GND），可以将模拟信号转换成 12 位数字信号。

3.9.4 基本配置

- 1、配置 CLKMU 模块 CLKDIV3 寄存器的 SRADCKS[1:0]，选择 ADC 时钟频率。
- 2、配置 syscfg 模块寄存器，选择输入管脚功能

3.9.5 功能描述

3.9.5.1 ADC 转换步骤描述

- 1、进行 AD 转换时，首先要通过配置 SRADCON2 寄存器的 CHS[4:0]选择输入通道 (AIN0~AIN16)，配置 VREFS[2:0]选择参考电压。
- 2、配置 SRADCON0 寄存器的 SRADACKS[1:0]选择采样时钟个数，配置 SAR_DIFFEN 选择是否使能差分输入。
- 3、配置 SRADCON1 寄存器 SRADEN 使能 ADC
- 4、配置 SRADCON1 寄存器 SRADS 启动 AD 转换。
- 5、转换结束后，硬件自动将 SRADS 清 0，并将转换结果存入寄存器 SRADL 和 SRADH（或 SROFTL 和 SROFTH 寄存器）中。

补充转换速度限制 中断时间描述

3.9.5.1 转换速度限制

采样时间最小值为 1.33us，转换时间最小值为 5.33us，sar_clk 频率最高为 3MHz。

3.9.5.2 输入电压和 SRAD 输出数据的关系

表 3-89 输入电压和 SRAD 输出数据的关系

输入电压	SRAD[11:0]											
	11	10	9	8	7	6	5	4	3	2	1	0
0/4096*VREF	0	0	0	0	0	0	0	0	0	0	0	0
1/4096*VREF	0	0	0	0	0	0	0	0	0	0	0	1
...												
...												
4094/4096*VREF	1	1	1	1	1	1	1	1	1	1	1	0
4095/4096*VREF	1	1	1	1	1	1	1	1	1	1	1	1

如果确定参考电压值，通过 ADC 转换得到了 ADC 码值，那么可以通过简单计算得出 ADC 输入电压值。计算公式如下

$$\text{ADC 输入电压} = (\text{SRAD}[11:0]/4096) * \text{VREF}$$

例如：当 ADC 参考电压为 1.0V，ADC 转换码值为 0x200，即十进制的 512，那么输入电压值为 $(512/4096) * 1.0 = 0.125\text{V}$ 。

3.9.5.3 转换时间

$$12 \text{ 位 AD 转换时间} = (1/\text{ADC 时钟频率}) \times (12 + \text{CALIF} + \text{ADC 输入信号获取时间})$$

表 3-90 转换时间说明表⁽¹⁾

指令周期	CALIF	SRADCKS	SRADACKS	AD 转换时间 ⁽³⁾
12M 指令周期	0	10	00	$1/(\text{Fcpu} / 8) \times (12 + 0 + 16) = 9.33\mu\text{s}$
			01	$1/(\text{Fcpu} / 8) \times (12 + 0 + 8) = 6.67\mu\text{s}$
			10	$1/(\text{Fcpu} / 8) \times (12 + 0 + 4) = 5.33\mu\text{s}$
		11	00	$1/(\text{Fcpu} / 16) \times (12 + 0 + 16) = 18.66\mu\text{s}$
			01	$1/(\text{Fcpu} / 16) \times (12 + 0 + 8) = 13.33\mu\text{s}$

			10	$1/(F_{cpu}/16) \times (12+0+4) = 10.67\mu s$
			11	$1/(F_{cpu}/16) \times (12+0+2) = 9.33\mu s$
		10	00	$1/(F_{cpu}/8) \times (12+1+16) = 9.66\mu s$
			01	$1/(F_{cpu}/8) \times (12+1+8) = 7\mu s$
			10	$1/(F_{cpu}/8) \times (12+1+4) = 5.66\mu s$
		11	00	$1/(F_{cpu}/16) \times (12+1+16) = 19.33\mu s$
			01	$1/(F_{cpu}/16) \times (12+1+8) = 14\mu s$
			10	$1/(F_{cpu}/16) \times (12+1+4) = 11.33\mu s$
			11	$1/(F_{cpu}/16) \times (12+1+2) = 10\mu s$
6M 指令 周期	0	01	00	$1/(F_{cpu}/4) \times (12+0+16) = 9.33\mu s$
			01	$1/(F_{cpu}/4) \times (12+0+8) = 6.67\mu s$
		10	00	$1/(F_{cpu}/8) \times (12+0+16) = 18.67\mu s$
			01	$1/(F_{cpu}/8) \times (12+0+8) = 13.33\mu s$
			10	$1/(F_{cpu}/8) \times (12+0+4) = 10.67\mu s$
		11	00	$1/(F_{cpu}/16) \times (12+0+16) = 37.33\mu s$
			01	$1/(F_{cpu}/16) \times (12+0+8) = 26.67\mu s$
			10	$1/(F_{cpu}/16) \times (12+0+4) = 21.33\mu s$
			11	$1/(F_{cpu}/16) \times (12+0+2) = 18.67\mu s$
	1	01	00	$1/(F_{cpu}/4) \times (12+1+16) = 9.67\mu s$
			01	$1/(F_{cpu}/4) \times (12+1+8) = 7\mu s$
		10	00	$1/(F_{cpu}/8) \times (12+1+16) = 19.33\mu s$
			01	$1/(F_{cpu}/8) \times (12+1+8) = 14\mu s$
			10	$1/(F_{cpu}/8) \times (12+1+4) = 11.33\mu s$
		11	00	$1/(F_{cpu}/16) \times (12+1+16) = 38.67\mu s$
			01	$1/(F_{cpu}/16) \times (12+1+8) = 28\mu s$
			10	$1/(F_{cpu}/16) \times (12+1+4) = 22.67\mu s$
			11	$1/(F_{cpu}/16) \times (12+1+2) = 20\mu s$
3M 指令 周期	0	01	00	$1/(F_{cpu}/4) \times (12+0+16) = 18.67\mu s$
			01	$1/(F_{cpu}/4) \times (12+0+8) = 13.33\mu s$
			10	$1/(F_{cpu}/4) \times (12+0+4) = 10.67\mu s$
		10	00	$1/(F_{cpu}/8) \times (12+0+16) = 37.33\mu s$
			01	$1/(F_{cpu}/8) \times (12+0+8) = 26.67\mu s$
			10	$1/(F_{cpu}/8) \times (12+0+4) = 21.33\mu s$
			11	$1/(F_{cpu}/8) \times (12+0+2) = 16\mu s$
		11	00	$1/(F_{cpu}/16) \times (12+0+16) = 74.67\mu s$
			01	$1/(F_{cpu}/16) \times (12+0+8) = 53.33\mu s$
			10	$1/(F_{cpu}/16) \times (12+0+4) = 42.67\mu s$
			11	$1/(F_{cpu}/16) \times (12+0+2) = 32\mu s$
	1	01	00	$1/(F_{cpu}/4) \times (12+1+16) = 19.33\mu s$
			01	$1/(F_{cpu}/4) \times (12+1+8) = 14\mu s$
			10	$1/(F_{cpu}/4) \times (12+1+4) = 11.33\mu s$
		10	00	$1/(F_{cpu}/8) \times (12+1+16) = 38.67\mu s$
			01	$1/(F_{cpu}/8) \times (12+1+8) = 28\mu s$
			10	$1/(F_{cpu}/8) \times (12+1+4) = 22.67\mu s$
			11	$1/(F_{cpu}/8) \times (12+1+2) = 20\mu s$
		11	00	$1/(F_{cpu}/16) \times (12+1+16) = 77.33\mu s$
			01	$1/(F_{cpu}/16) \times (12+1+8) = 56\mu s$
			10	$1/(F_{cpu}/16) \times (12+1+4) = 45.33\mu s$
			11	$1/(F_{cpu}/16) \times (12+1+2) = 40\mu s$

(1) AD 转换时间随 Fcpu 频率的改变而改变。

3.9.5.4 ADC 采样时间

ADC 采样时间通过 SRADCON0 寄存器进行配置，通过 SRADACKS[1:0]配置采样时钟个数，通过 SRADCKS[1:0]配置 ADC 时钟频率。

ADC 采样时间 = 采样时钟个数 / ADC 时钟频率。

下面以内部 CPU 时钟为 12MHz，指令周期 6MHz 为例，对不同配置下的采样时间进行计算（其中标蓝色部分为不满足要求的配置，禁止使用）

表 3-91

SRADCKS	SRADACKS	AD 采样时间
00	00	16/(6MHz/1) = 2.67us
	01	8/(6MHz/1) = 1.33us
	10	4/(6MHz/1) = 0.67us
	11	2/(6MHz/1) = 0.33us
01	00	16/(6MHz/2) = 5.33us
	01	8/(6MHz/2) = 2.67us
	10	4/(6MHz/2) = 1.33us
	11	2/(6MHz/2) = 0.67us
10	00	16/(6MHz/4) = 10.67us
	01	8/(6MHz/4) = 5.33us
	10	4/(6MHz/4) = 2.67us
	11	2/(6MHz/4) = 1.33us
11	00	16/(6MHz/8) = 21.33us
	01	8/(6MHz/8) = 10.67us
	10	4/(6MHz/8) = 5.33us
	11	2/(6MHz/8) = 2.67us

ADC 采样时间与芯片电压、外部负载都有关系，下表列出了典型情况下的采样时间要求

表 3-92

参数	测试条件	最小值	典型值	最大值	单位
采样时间	3.3V≤VDD≤5.5V, R _{AIN} ≤10 kΩ	8/3			μs
	2.4V≤VDD<3.3V, R _{AIN} ≤10 kΩ	8/3			μs
	2.2V≤VDD<2.4V, R _{AIN} ≤10 kΩ	16/3			μs

R_{AIN} 为输入负载电阻。

3.9.5.5 参考电压输出

ADC 内部参考电压输出是将 ADC 参考电压配置为内部参考电压为 1.0V/2.5V，并通过 PT3.0 口输出，可外接电容提高参考电压精度。

配置 ADC 参考电压输出步骤如下：

- 1) 配置 SYSCFG 模块 CFGR6 寄存器的 PT30FUNC=01，将 PT3.0 口设置为模拟口
- 2) 配置 SRADCON2 的 VREFS[2:0]位为 11X，将 ADC 参考电压配置为内部参考电压为 1.0V(110)/2.5V(111)。

3.9.5.6 AD 失调电压校正

不同芯片由于离散性的原因，AD 的失调电压可能有正有负。

校正失调电压的方法：

- 1) 芯片在量产时对 ADC 失调电压进行了测试，并将失调电压对应的码值（1.0V 参考电压下）。存在信息区地址 0xFA03 中，低 8 位有效。用户需要在用户程序中用 MOVP 指令将该数据读出。
- 2) 将该数据写入 SROFTL 寄存器中，并向 SROFTH 写入 00h。
- 3) 将 SRADCON1 寄存器的 OFTEN 位置 0，CALIF 位置 1，ENOV 位置 0。
- 4) 使能 ADC 转换，那么从 SRADH/SRADL 中读取的 AD 值即为减去失调后的值。

3.9.5.7 数字比较器

ADC 模块可作为一个数字比较器。被测信号的输入频率应小于转换频率的 1/2。比较器的速率是和 AD 转换频率相关的。两个输入信号的差值必须小于 VREF/2，否则比较结果会出错。

操作：

- 1) 通过 ADC 通道选择控制位 chs[4:0]选择比较器负端的信号输入，之后把 OFTEN 置 1，CALIF 清 0，ENOV 置 0，把 SRADEN 置 1 使能 ADC，SRADS 置 1 启动转换，转换完成可把转换结果写入 SROFT 寄存器。
也可以直接把负端信号的 AD 值直接写到 SROFT 寄存器中，即人为指定负端电压值。
- 2) 通过 ADC 通道选择控制位 chs[4:0]选择比较器正端的信号输入，之后把 OFTEN 置 0，CALIF 清 1，ENOV 置 1，把 SRADEN 置 1 使能 ADC，SRADS 置 1 启动转换。
- 3) AD 数据的最高位 SRAD[11]则是比较器的结果，为 0 时表示正端电压大于负端电压，为 1 时表示正端电压小于负端电压。SRAD[11:0]为差值，带符号位的补码。

3.9.5.8 内部测量 VDD 的电压

用户可以通过使用内部参考电压或者外部参考电压输入（外部参考电压固定且不随 VDD 电压变化）两种方法来测试芯片内部 VDD 的电压。

使用外部参考电压，需额外提供参考源。

使用内部参考电压不需要额外的硬件资源。但是，使用内部参考电压会由于本身内部参考电压值的不准而影响精度。可以通过内部参考电压校正来提高测试的精度。

3.9.6 寄存器映射

表 3-93 ADC 寄存器列表

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值	
基址 PAGE0, SRADC_BA = 0x60											
00h	SRADCON ₀			SRADACKS[1:0]		SRADIF	SRADIE			SAR_DIFFEN	00000000
01h	SRADCON ₁	SRADEN	SRADS	OFTEN	CALIF	ENOV	OFFEX				00000000
02h	SRADCON ₂	CHS[4:0]					VREFS[2:0]				00000100
03h	SRADL	SRAD[7:0]									00000000
04h	SRADH					SRAD[11:8]					00000000
05h	SROFTL	SROFT[7:0]									00000000
06h	SROFTH					SROFT[11:8]					00000000

3.9.7 寄存器描述

3.9.7.1 SRADCON0 寄存器（地址为 00h）

表 3-94

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	R/W-0	R/W-0	RC/W0-0	R/W-0	U-0	R/W-0
SRADCON0			SRADACKS[1:0]		SRADIF	SRADIE		SAR_DIFFEN

表 3-95

位地址	标识符	功能
7:6	reserve	保留
5: 4	SRADACKS[1:0]	ADC 输入信号采样时钟个数选择信号
		SRADACKS[1:0]
		00
		01
		10
3	SRADIF	AD 中断标志，软件清零，硬件置高 1 = 发生 AD 中断，必须软件清 0 0 = 没发生 AD 中断
		01
		10
		11
2	SRADIE	ADC 输入信号采样时钟个数
		16 个 ADC 时钟
		8 个 ADC 时钟
		4 个 ADC 时钟
0	SAR_DIFFEN	2 个 ADC 时钟
		AD 差分输入使能
		1 = 使能 SAR_ADC 中断 0 = 不使能 SAR_ADC 中断
		1: 全差分模式(参考电压不能选择 VDD); 0: ADC 为单端模式

3.9.7.2 SRADCON1 寄存器（地址为 01h）

表 3-96

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0
SRADCON1	SRADEN	SRADS	OFTEN	CALIF	ENOV	OFFEX		

表 3-97

位地址	标识符	功能
7	SRADEN	ADC 使能位 1: 使能 0: 禁止
6	SRADS	ADC 启动位/状态控制位 1: 开始，转换过程中 0: 停止，转换结束 当置位后，启动 ADC 转换，转换完成会自动清 0

5	OFTEN	转换结果选择控制位 1: 转换结果放在 SROFT 寄存器中 0: 转换结果放在 SRAD 寄存器中
4	CALIF	校正控制位(OFTEN 为 0 时有效) 1: 使能校正, 即 AD 转换的结果是减去了 SROFT 失调电压值 0: 禁止校正, 即 AD 转换结果是没有减去 SROFT 失调电压值
3	ENOV	使能比较器溢出模式(CALIF 为 1 时有效) 1: 使能, 上溢或下溢直接是减去后的结果 0: 禁止, 下溢为 000h, 上溢为 fffh
2	OFFEX	差分 ADC 正负端内部互换 (SAR_DIFFEN=1 时, OFFEX 才有效) 1: 差分 ADC 正负端内部互换 0: 差分 ADC 正负端内部不互换

3.9.7.3 SRADCON2 寄存器 (地址为 02h)

表 3-98

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W -1	R/W -0	R/W-0
SRADCON2	CHS[4:0]					VREFS[2:0]		

表 3-99

位地址	标识符	功能
7: 3	CHS[4:0]	ADC 输入通道选择位
		CHS[3:0] 输入通道
		SAR_DIFFEN=1, 为差分 ADC 的
		00000 AIN11 / AIN12
		00001 AIN13/ AIN14
		00010 AIN15/ AIN16
		00011 AIN0/VSSA
		00100 AIN1/VSSA
		00101 AIN2/VSSA
		00110 AIN3/VSSA
		00111 AIN4/VSSA
		01000 AIN5/VSSA
		01001 AIN6/VSSA
		01010 AIN7/VSSA
		01011 AIN8/VSSA
		01100 AIN9/VSSA
		01101 AIN10/VSSA
		01110 AIN11/VSSA
		01111 AIN12/VSSA
		10000 AIN13/VSSA
		10001 AIN14/VSSA
		10010 AIN15/VSSA
		10011 AIN16/VSSA
		101XX VDD/8/VSSA
		110XX VREF/VSSA
		111XX GND/VSSA
		SAR_DIFFEN=0 为单端 ADC, 以下为单端

		00000	AIN0
		00001	AIN1
		00010	AIN2
		00011	AIN3
		00100	AIN4
		00101	AIN5
		00110	AIN6
		00111	AIN7
		01000	AIN8
		01001	AIN9
		01010	AIN10
		01011	AIN11
		01100	AIN12
		01101	AIN13
		01110	AIN14
		01111	AIN15
		100XX	AIN16
		101XX	ADC 内部 1/8VDD
		110XX	ADC 内部参考电压
		111XX	ADC 内部接地
2: 0	VREFS[2:0]	ADC 内部参考电源选择 注：不同参考电压切换，建议延迟 40uS 再做 AD 转换	
		VREFS[2:0]	AD 参考电压
		00X	AD 参考电压为 VDD
		01X	AD 参考电压为 PT3.0 口外部输入电压
		100	AD 参考电压为 1.0V 内部电压
		101	AD 参考电压为 2.5V 内部电压
		110	AD 参考电压为 1.0V 内部电压并通过 PT3.0 口输出
		111	AD 参考电压为 2.5V 内部电压并通过 PT3.0 口输出

3.9.7.4 SRADL 寄存器（地址为 03h）

表 3-100

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
SRADL	SRAD[7:0]							

表 3-101

位地址	标识符	功能
7: 0	SRAD[7:0]	ADC 数据的低 8 位，只可读

3.9.7.5 SRADH 寄存器（地址为 04h）

表 3-102

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0

SRADH		SRAD[11:8]
-------	--	------------

表 3-103

位地址	标识符	功能
3: 0	SRAD[11:8]	ADC 数据的高 4 位, 只可读

3.9.7.6 SROFTL 寄存器 (地址为 05h)

表 3-104

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SROFTL	SROFT[7:0]							

表 3-105

位地址	标识符	功能
7: 0	SROFT[7:0]	校正值数据的低 8 位

3.9.7.7 SROFTH 寄存器 (地址为 06h)

表 3-106

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
SROFTH					SROFT[11:8]			

表 3-107

位地址	标识符	功能
3: 0	SROFT[11:8]	校正值数据的高 4 位

3.10 串行通信接口

3.10.1 概述

提供 1 个独立可编程全双工异步串行通信接口。

3.10.2 特性

- 支持数据同时发送和接收
- 支持接收滤波消除毛刺
- 波特率可配，最高速率能够支持 115200bps
- 支持 8/9 位数据发送/接收
- 支持 1 级发送 FIFO 和 8 级接收 FIFO
- 支持 Sleep 模式下接收唤醒
- 支持自动波特率

3.10.3 功能框图

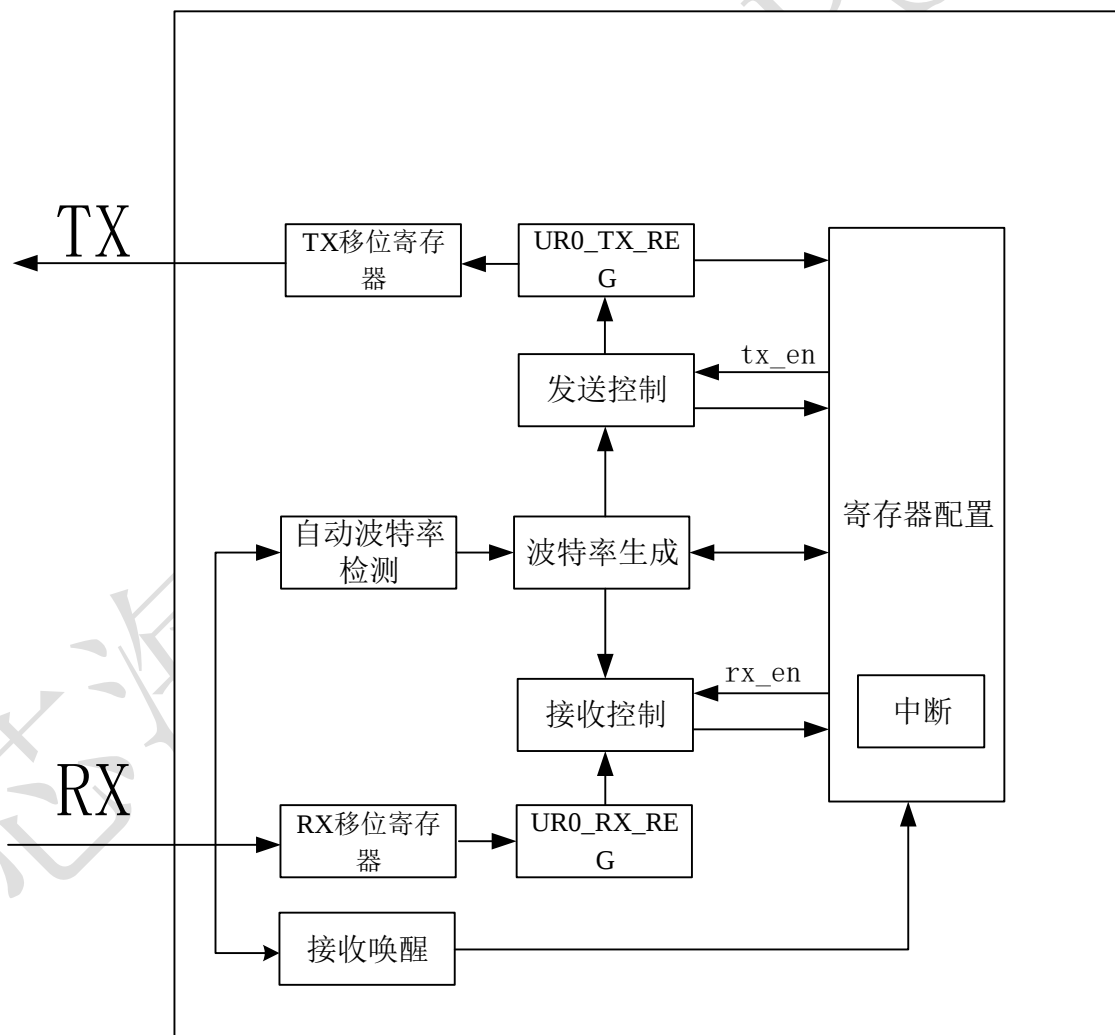


图 3-17 UART 功能框图

3.10.4 基本配置

- 1、配置 CLKMU 模块 CLKDIV2 寄存器，设置 uart 时钟
- 2、配置 CFGR6 和 CFGR7 寄存器，选择端口

3.10.5 功能描述

3.10.5.1 波特率配置

UART0 的波特率配置公式，
$$BR = \frac{UART0_CLK}{BRR0 + \frac{BRR1}{10}}$$

其中 $UART0_CLK = \frac{ICK}{2^{UART0_DIV+1}}$ ，URTDIV 的范围为 0~5；BRR1 的范围为 0~9

例如：SYSCLK=24MHz，若要以 115200bps 波特率通讯，采用 UART_CLK0=4MHz，进行第一步的计算，计算出 BRR0= 34, BRR1 = 7；如果 BRR0 的结果小于 100，可以分频系数保持不变。如果 BRR0 计算的结果大于 100，那么分频系数 URTDIV 加 1。分频系数越大，功耗就越低。

3.10.5.2 接收滤波

串口支持接收滤波，可消除接收信号毛刺，提高通信可靠性。串口可滤除毛刺的最大宽度为 $T_{UART_CLK} * 7$ 。BRR0 配置需至少为 8。

3.10.5.3 接收流程

接收的 RXFIFO 深度为 1，可以缓存一个字节的的数据。

发送流程为：

- 1) 设置 BRR0/BRR1，选择合适的波特率
- 2) 配置 URAT_EN=1，使能 UART0 串口
- 3) 如果需要接收第 9 位，将 RX9_EN 置 1
- 4) 置位 RX_EN=1
- 5) 当接收完成，硬件置位 UR0_RIF，会进入接收中断。若从 RXFIFO 读取数据后 UR0_RIF 仍为高电平，说明缓存中还有数据，知道 FIFO 中的数据全部读取之后，UR0_RIF 才会被硬件清零
- 6) 在 Sleep 模式下，当有数据接收时（RX 端口被拉低），串口唤醒中断置位，若此时串口唤醒中断使能开启的情况下，可唤醒 Sleep 模式。

3.10.5.4 发送流程

发送的 TXFIFO 深度为 1，可以缓存一个字节的的数据。

发送流程为：

- 1) 设置 BRR0/BRR1，选择合适的波特率
- 2) 配置 URAT_EN=1，使能 UART0 串口
- 3) 如果需要发送第 9 位，将 TX9_EN 置 1
- 4) 将数据写入发送缓存寄存器 UR0_TXREG
- 5) 置位 TX_EN=1
- 6) 当发送移位寄存器中的数据传输完成后，若此时 TXFIFO 有数据，硬件自动从 TXFIFO 中读取数据并发送，读取后 TXFIFO 为空，发送中断置位，此时可将下一个数据写入发送缓存寄存器中。

3.10.6 寄存器映射

表 3-108

地址	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复位值
基址 PAGE1, UART0_BA = 0x20										
00H	UR0_CR1	TX9D	RX9D	TX9_EN	RX9_EN	RX_EN N	TX_EN		UART 0_EN	00000000
01H	UR0_BR R0	BRR0								00000000
02H	UR0_BR R1					BRR1				00000000
03H	UR0_TX_ REG	TX_REG								00000000
04H	UR0_RX_ REG	RX_REG								00000000
05H	UR0_ST	UR_T INV	UR_RI NV		TXFIFO_ EMPTY	RX_B USY	TX_ BUS Y	RX_O V_ER R	STOP_ ERR	00010000
06H	UR0_INT F				UR0ERRI F	UR0_ RHIF	UR0_ _RNI F	UR0W KIF	UR0_T EIF	00000000
07H	UR0_INT E				UR0ERRI E	UR0_ RHIE	UR0_ _RNI E	UR0W K_IE	UR0_T EIE	00000000

3.10.7 寄存器描述

3.10.7.1 UR0_CR1 寄存器（地址 00H）

表 3-109

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
UR0_CR1	TX9D	RX9D	TX9_EN	RX9_EN	RX_EN	TX_EN		UART0_EN

表 3-110

位地址	标识符	功能
7	TX9D	发送数据第 9 位 1: 发送的第 9 位数据为 1 0: 发送的第 9 位数据为 0
5	RX9D	接收数据第 9 位, 该位只读, 不可写 1: 接收的第 9 位数据为 1 0: 接收的第 9 位数据为 0
5	TX9_EN	发送第 9 位数据使能信号 1: 发送第 9 位数据 0: 不发送第 9 位数据
4	RX9_EN	接收第 9 位数据使能信号 1: 接收第 9 位数据 0: 不接收第 9 位数据
3	RX_EN	数据接收控制信号 1: 允许接收数据 0: 禁止接收数据
2	TX_EN	数据发送控制信号

		1: 允许发送数据 0: 禁止发送数据
1	RESERVE	
0	UART0_EN	串口 0 模块使能控制信号 1: 打开 UART0 0: 关闭 UART0

3.10.7.2 UR0_BRR0 寄存器（地址为 01H）

表 3-111

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UR0_BRR0	BRR0[7:0]							

表 3-112

位地址	标识符	功能
7: 0	BRR0[7:0]	波特率设置寄存器 0, 整数部分

3.10.7.3 UR0_BRR1 寄存器（地址为 02H）

表 3-113

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
UARTDIV					BRR1[3:0]			

表 3-114

位地址	标识符	功能
7: 4	RESERVE	
3: 0	BRR1[3:0]	波特率设置寄存器 1, 小数部分

3.10.7.4 UR0_TX_REG 寄存器（地址为 03H）

表 3-115

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UR0_TX_REG	TX_REG[7:0]							

表 3-116

位地址	标识符	功能
7: 0	TX_REG[7:0]	串口发送数据寄存器

3.10.7.5 UR0_RX_REG 寄存器（地址为 04H）

表 3-117

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
-----	------	------	------	------	------	------	------	------

特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UR0_RX_REG	RX_REG[7:0]							

表 3-118

位地址	标识符	功能
7: 0	RX_REG[7:0]	串口接收数据寄存器

3.10.7.6 UR0_ST 寄存器（地址为 05H）

表 3-119

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	U-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
UR0_ST	UR_TINV	UR_RINV		TXFIFO_EMPTY	RX_BUSY	TX_BUSY	RX_OV_ERR	STOP_ERR

表 3-120

位地址	标识符	功能
7	UR_TINV	UART 输入输出信号取反控制信号 1: 使能 UART 输出信号取反 0: 禁止 UART 输出信号取反（默认）
6	UR_RINV	UART 输入输出信号取反控制信号 1: 使能 UART 输入信号取反 0: 禁止 UART 输入信号取反（默认）
5	RESERVE	保留
4	TXFIFO_EMPTY	发送 FIFO 空标志，可在中断中清除 TX_TEIF 后判断发送 FIFO 是否为空状态 1: 发送 FIFO 为空 0: 发送 FIFO 不为空
3	RX_BUSY	接收 Busy 指示信号 1: 接收端正在接收 0: 接收端未进行接收，处于空闲状态 注：当进入 Sleep 后，系统可以采用 UART 唤醒中断进行唤醒。系统被唤醒后，可以查询该位，若 RX_BUSY=0，表示此次唤醒可能是由于干扰造成。
2	TX_BUSY	发送寄存器 TX_REG 的数据是否全部串行发送完毕 1: 还未全部发送完毕 0: 已经全部发送完毕
1	RX_OV_ERR	接收 FIFO 溢出错误标志 1: 发生了接收 FIFO 溢出的错误 0: 未发生接收 FIFO 溢出的错误
0	STOP_ERR	接收停止位错误标志 1: 发生了停止位错误 0: 未发生停止位错误

3.10.7.7 UR0_INTF 寄存器（地址为 06H）

表 3-121

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UR0_INTF				UR0ERRIF	UR0_RHIF	UR0_RNIF	UR0WK_IF	UR0_TEIF

表 3-122

位地址	标识符	功能
7: 5	RESERVE	保留
4	UR0ERRIF	串口 0 接收错误中断标志 1: 发生了串口 0 接口错误中断 0: 未发生串口 0 接收错误中断, 或软件清除 RX_OV_ERR/STOP_ERR 后 当串口 0 接收停止位错误或 FIFO 溢出错误发生时, 标志置位。需查验 UR0_ST 寄存器 RX_OV_ERR、STOP_ERR 确认错误类型, 在清除 RX_OV_ERR、STOP_ERR 后, UR0ERRIF 自动清零, 不能直接清除 UR0ERRIF 标志
3	UR0_RHIF	串口 0 接收 FIFO 半满中断标志 1: 发生了串口 0 接收 FIFO 半满中断, 接收 FIFO 中数据长度大于等于 4 0: 未发生串口 0 接收 FIFO 半满中断, 接收 FIFO 中数据长度小于 4
2	UR0_RNIF	串口 0 接收 FIFO 非空中断标志 1: 发生了串口 0 接收 FIFO 非空中断, 接收 FIFO 不是空状态 0: 未发生串口 0 接收 FIFO 非空中断, 接收 FIFO 此时为空状态
1	UR0WK_IF	串口接收唤醒中断标志 1: 使能串口接收唤醒中断 0: 不使能串口接收唤醒中断 在 Sleep 状态时, 串口接收到数据起始位时置 1; 在非 Sleep 状态下串口接收到数据起始位时不触发该中断
0	UR0_TEIF	串口 0 发送 FIFO 空中断发生标志 1: 发生了串口 0 发送 FIFO 空中断 0: 未发生串口 0 发送 FIFO 空中断 当串口 0 发送 FIFO 空中断产生时, 可向发送 FIFO 写入发送数据。需注意, 当发送 FIFO 空中断产生时, 串口仍处于发送数据状态, 必须等待 UR0_ST 寄存器中的 TX_BUSY 标志拉低时, 才表示数据发送完成, 才能进入 Sleep 状态。

3.10.7.8 UR0_INTE 寄存器 (地址为 07H)

表 3-123

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UR0_INTE				UR0ERRIE	UR0_RHIE	UR0_RNIE	UR0WK_IE	UR0_TEIE

表 3-124

位地址	标识符	功能
7: 5	RESERVE	保留
4	UR0ERRIE	串口 0 接收错误中断标志 1: 开启串口 0 接口错误中断使能

		0: 关闭串口 0 接收错误中断使能
3	UR0_RHIE	串口 0 接收 FIFO 半满中断标志 1: 开启串口 0 接收 FIFO 半满中断使能 0: 关闭串口 0 接收 FIFO 半满中断使能
2	UR0_RNIE	串口 0 接收 FIFO 非空中断标志 1: 开启串口 0 接收 FIFO 非空中断使能 0: 关闭串口 0 接收 FIFO 非空中断使能
1	UR0WK_IE	串口接收唤醒中断标志 1: 开启串口接收唤醒中断使能 0: 关闭串口接收唤醒中断使能
0	UR0_TEIE	串口 0 发送 FIFO 空中断发生标志 1: 开启串口 0 发送 FIFO 空中断使能 0: 关闭串口 0 发送 FIFO 空中断使能

3.11 FLASH 控制器

3.11.1 概述

CSU3AF10 芯片支持 64KB 闪存空间，具有擦写保护机制。Flash 控制器响应内核、烧录或 ICD 对 Flash 的读写擦访问。

3.11.2 特性

- 64KB 大容量存储
- 读选项字节
- Flash 擦写保护机制
- 支持 Flash 写入和擦除操作
- 支持低功耗模式

3.11.3 功能描述

3.11.3.1 擦写操作解锁

对 Flash 进行擦写操作时，必须解锁写保护。解锁写保护需对 WRPRT 寄存器连续写入 96H、69H、5AH，解锁成功后，UNLOCK 寄存器会被置 1；当对 ISPCR 寄存器的 START 位写 1 时，UNLOCK 寄存器会被清零，解锁自动失效。

3.11.3.2 FLASH 读操作

读 Flash 步骤如下：

- 1、读 ISPSR 寄存器的 BSY 位，确认上一次编程操作已结束；
- 2、写 CMD=00，打开写 Flash 功能；
- 3、向 EADRH 和 EADRL 寄存器写入待操作的地址；
- 4、写 START=1，开始读 Flash；
- 5、内核停止取指，等待 BSY 位为 0，从 EDATx 寄存器中获取 Flash 的值，程序继续运行。

3.11.3.3 FLASH 写操作

Flash 数据只允许从 1 改写成 0，不允许从 0 改写成 1，所以每次执行写操作前，需要先擦除对应地址所在 sector 的全部数据。在对 Flash 编程时，Flash 接口会预读待编程地址的数据是否为全 1，如果不是全 1，编程操作会自动取消，并在置位 ISPSR 寄存器中的 PGERR 标志位提示编程错误告警。如果为全 1，则正确编程。

在用户模式下，主程序区编程步骤如下：

- 1、读 ISPSR 寄存器的 BSY 位，确认上一次编程操作已结束；
- 2、写 ISPEN=1 使能擦写 Flash；写 CMD=05，打开写 Flash 功能；
- 3、向 EADRH 和 EADRL 寄存器写入待操作的地址；
- 4、向 EDATH 和 EDATL 寄存器写入待写入的数据；
- 5、向 WRPRT 寄存器连续写入 96H、69H、5AH，读 UNLOCK 寄存器确认已解锁成功；
- 6、写 START=1，内核停止取指，开始编程；
- 7、硬件自动等待 50 微秒后，擦除结束，程序继续运行。

3.11.3.4 FLASH 擦除

Flash 存储器可以按 sector 擦除，也可以整片擦除，用户模式下，不允许执行 CHIP 擦操作。

Sector 擦除步骤如下：

- 1、读 ISPSR 寄存器的 BUSY 位，确认上一次编程操作已结束；

- 2、写 ISPEN=1 使能擦写 Flash；写 CMD=0A，打开 Flash sector 擦除功能；
- 3、向 EADDRH 和 EADRL 寄存器写入待擦除的 sector 地址；
- 4、向 WRPRT 寄存器连续写入 96H、69H、5AH，读 UNLOCK 寄存器确认已解锁成功；
- 5、写 START=1，内核停止取指，开始擦除；
- 6、硬件自动等待 6 毫秒后，擦除结束，程序继续运行；

3.11.3.5 FLASH 地址映射

因为 flash 只支持 sector program，所以假如只想改写 1 个地址的数据，则应进行 sector erase，而在 erase 之前必须先把不改写的数据读出来进行保存。NVR sector0 为用户数据区，NVR sector1 为代码选项区，NVR sector2 为 BIST 校准保存区，NVR sector3 为生产信息区。

其中 NVR 区只有 sector0 和 sector1 在用户模式下可改写，其他 NVR 区在用户模式下不可改写。烧录模式和 ICD 调试模式，全部 NVR 区都可改写。

表 3-125 FLASH 地址映射表

地址分段	FLASH 地址编 EADR[15:0]	FLASH 实际物理地址
主程序	0000h	0000h
	0001h	
	0002h	0001h
	0003h	
		
		
	7FFEh	3FFFh
	7FFFh	
NVR sector0	F000h	0000h
	F001h	
		
		
	F0FEh	007Fh
	F0FFh	
NVR sector1	F800h	0080h
	F801h	
		
		
	F8FEh	00FFh
	F8FFh	
NVR sector2	FA00h	0100h
	FA01h	
		
		
	FAFEh	017Fh
	FAFFh	
NVR sector3	FC00h	0180h
	FC01h	
		
		
	FCFEh	01FFh
	FCFFh	

3.11.4 寄存器映射

表 3-126

地	名称	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	上电复
---	----	------	------	------	------	------	------	------	------	-----

址										位值
基址 PAGE0, FMC_BA = 0x68										
00h	EADRH	EDAR [15:8]								00000000
01h	EADRL	EDAR [7:0]								00000000
02h	EDATH	EDAT[15:8]								00000000
03h	EDATL	EDAT[7:0]								00000000
04h	ISPCR	READM1	READM0	ISPEN					START	00000000
05h	CMD	CMD[7:0]								00000000
06h	ISPSR					UNLOCK	PGERR	ISPOF	BSY	00000000
07h	WRPRT									00000000

3.11.5 寄存器描述

3.11.5.1 EADRH 寄存器（地址为 00h）

表 3-127

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EADRH	EADR[15:8]							

表 3-128

位地址	标识符	功能
7: 0	EADR[15:8]	读操作地址高位，该寄存器与 EADRL 组成 16 位的地址控制寄存器，可以访问 32K*16 空间。超出 32K 空间的地址读出数据为代码选项区或者无效数据。

3.11.5.2 EADRL 寄存器（地址为 01h）

表 3-129

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EADRL	EADR[7:0]							

表 3-130

位地址	标识符	功能
7: 0	EADR[7:0]	读操作地址低 8 位，该寄存器与 EADRH 组成 16 位的地址控制寄存器，可以访问 32K*16 空间。

3.11.5.3 EDATH 寄存器（地址为 02h）

表 3-131

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EDATH	EDAT [15:8]							

表 3-132

位地址	标识符	功能
7: 0	EDAT[15:8]	读写数据高 8 位，与存储在 EDATL 中的低 8 位数据一起组成 16 位数据。

3.11.5.4 EDATL 寄存器（地址为 03h）

表 3-133

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
EDATL	EDAT [7:0]							

表 3-134

位地址	标识符	功能
7: 0	EDAT[7:0]	读写数据低 8 位，与存储在 EDATH 中的高 8 位数据一起组成 16 位数据。

3.11.5.5 ISPCR 寄存器（地址为 04H）

表 3-135

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W -0	R/W -0	R/W -0	U -0	U -0	U -0	U -0	W-0
ISPCR	READM1	READM0	ISPEN					START

表 3-136

位地址	标识符	功能
7	READM1	Margin read 1 使能 0: 未打开 1: 打开
6	READM0	Margin read 0 使能 0: 未打开 1: 打开
5	ISPEN	Flash 擦写使能 在对 Flash 执行擦写操作时，需要先对该位写 1。
4:1	RESERVE	保留
0	START	软件向该位写 1，执行擦除或写入动作，仅单次有效。当 UNLOCK=1 时，该位可写。

3.11.5.6 CMD 寄存器（地址为 05H）

表 3-137

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W -0	R/W-0
CMD	CMD[7:0]							

表 3-138

位地址	标识符	功能
7:0	CMD[7:0]	Flash 操作命令 00: 读 Flash 05: 写 Flash 0A: sector 擦除 A0: chip 擦除 50: margin read 其他: 保留 当 BSY 为高时, CMD 寄存器不可改写。

3.11.5.7 ISPSR 寄存器 (地址为 06H)

表 3-139

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0
ISPSR					UNLOCK	PGERR	ISPOF	BSY

表 3-140

位地址	标识符	功能
7:4	RESERVE	保留
3	UNLOCK	Flash 序列解锁标志。向 WRPRT 写入正确的解锁序列, 该位置位; 内核发起任意写操作, 清除该位。 0: Flash 未解锁 1: Flash 已解锁
2	PGERR	编程错误 当对 Flash 中数据为非 0xFFFF 的地址写入时, 硬件置位, 软件写 0 清除。 0: 未发生编程错误 1: 发生编程错误
1	ISPOF	写/擦除操作失败标志位。(硬件置位, 软件写 0 清除, 用户模式) 1'b0: 写/擦除操作成功。 1'b1: 写/擦除操作失败, 写 0 清零 如果对 USR_W_SEC[x]=0 的区域进行擦写会置位 ISPOF。
0	BSY	该位在 Flash 操作开始时置位, 操作结束后清除。

3.11.5.8 WRPRT 寄存器 (地址为 07H)

表 3-141

位编号	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
特性	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0
WRPRT								

表 3-142

位地址	标识符	功能
7:0	WRPRT	对 Flash 擦写操作进行解锁时, 需对 WRPRT 寄存器连续写入 96H、69H、5AH。UNLOCK 表示解锁是否成功。

EADRH/EADRL 提供读操作的数据地址；
EDATH/ EDATL 提供读操作所用的数据。

芯海科技CHIPSEA

3.12 在线调试功能（ICD）

3.12.1 在线调试功能概述

CSU3AF10 支持在线调试功能，通过 PT2.5 口与芯片进行通信，配合 IDE 实现在线调试功能，SWD 为开漏输出口。

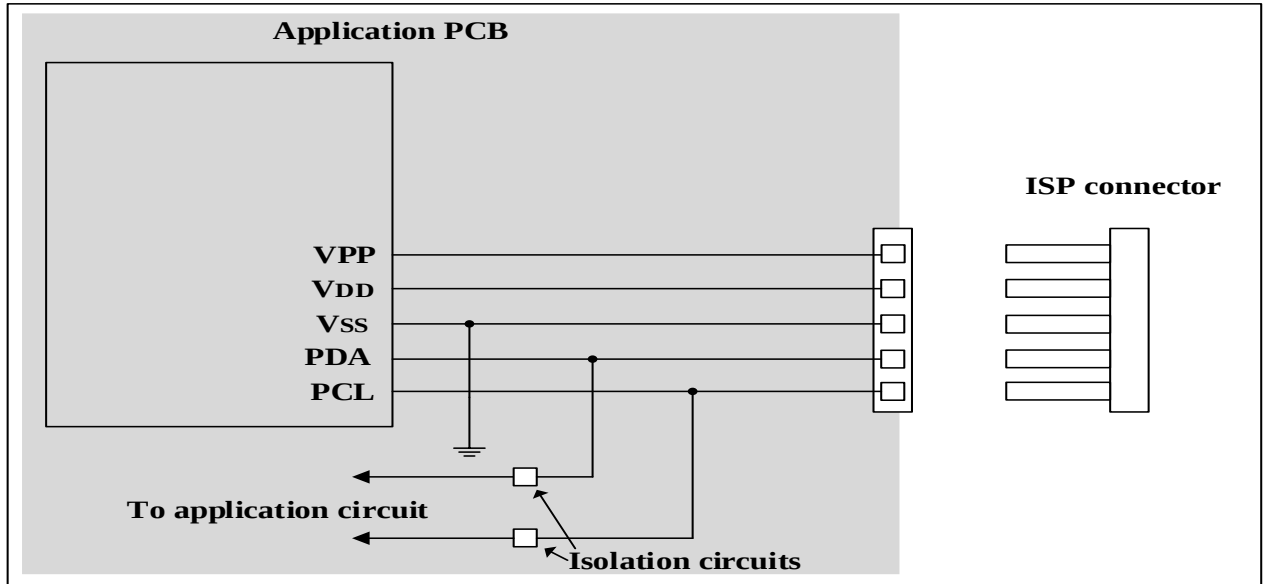


图 3-18 在线调试系统框图

3.13 烧录模块

3.13.1 烧录器接口

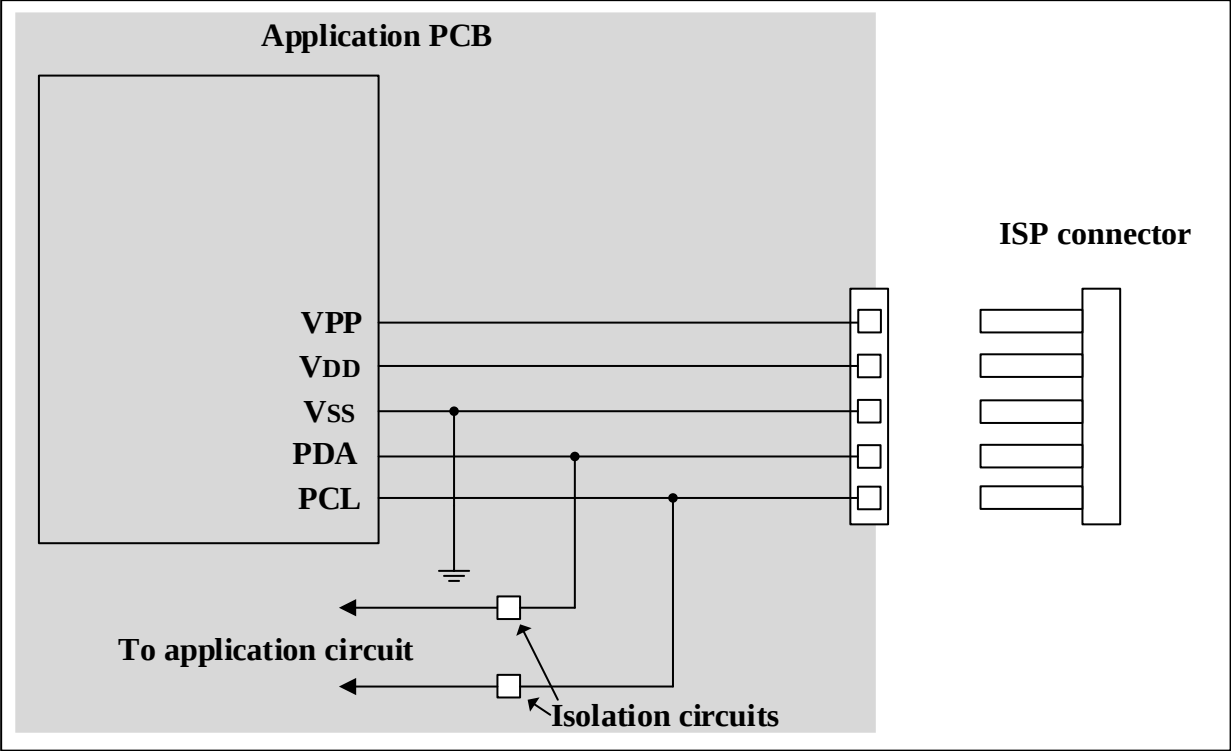


图 3-19 烧录器接口图

表 3-143 烧录接口说明

端口名称	型式	说明
VPP	输入	PT1.3 端口，烧录电源
VDD	输入	电源正端
VSS	输入	电源负端
PDA	输入/输出	PT1.4 端口，数据信号
PCL	输入	PT1.5 端口，时钟信号

3.14 代码选项

3.14.1 OPTION0

地址：0xF800。

代码选项数据的低 8 位和高 8 位必须为取反关系，否则芯片无法进入正常工作模式。

表 3-144

Bit-15	Bit-14	Bit-13	Bit-12	Bit-11	Bit-10	Bit-9	Bit-8
~CPUCLK_S[1:0]		~LVR_EN	~LVD_SEL	~RESET_PIN	~WWDT_HALT		~SECURITY
Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
CPUCLK_S[1:0]		LVR_EN	LVD_SEL	RESET_PIN	WWDT_HALT		SECURITY

表 3-145

位地址	标识符	功能
15:14	~CPUCLK_S[1:0]	必须为 CPUCLK_S[1:0]的取反值
13	~LVR_EN	必须为 LVR_EN 的取反值
12	~LVD_SEL	必须为 LVD_SEL 的取反值
11	~RESET_PIN	必须为 RESET_PIN 的取反值
10	~WWDT_HALT	必须为 WWDT_HALT 的取反值
9	RESERVE	保留
8	~SECURITY	必须为 SECURITY 的取反值
7:6	CPUCLK_S[1:0]	CPUCLK 预分频 00: 不分频 01: 2 分频（默认） 10: 4 分频 11: 保留
5	LVR_EN	低电压检测复位使能 0: 不复位（默认） 1: 复位
4	LVD_SEL	低电压检测档位选择 0: 2.1V（默认） 1: 2.4V
3	RESET_PIN	复位引脚选择 1: PT1.3 作为复位引脚 0: PT1.3 作为普通输入输出口（默认）
2	WWDT_HALT	WWDT 在 HALT 模式下行为模式配置 0: WWDT 在 HALT 模式下继续计数，计数溢出后也会产生复位。（默认） 1: WWDT 在 halt 模式下不进行计数。
1	RESERVE	保留
0	SECURITY	烧录模式主程序代码保密位 0: 使能代码加密（默认） 1: 禁止代码加密 注：当加密时，除 NVR2 和 NVR3 外，所有区域不能进行 sector 擦，只能用 chip 擦除，读主程序区和 NVR0 时，读出的数据固定为 0000H；其他 NVR 可以正常读。在加密时，必须执行 chip_erase 后，才能获得加密区域的擦写权限。

3.14.2 OPTION1

地址：0xF801。

代码选项数据的低 8 位和高 8 位必须为取反关系，否则芯片无法进入正常工作模式。

表 3-146

Bit-15	Bit-14	Bit-13	Bit-12	Bit-11	Bit-10	Bit-9	Bit-8
SAR_LPOWER	WDT_CFG					~ILOPLEN	~ EMCEN
Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
SAR_LPOWER	WDT_CFG					ILOPLEN	EMCEN

表 3-147

位地址	标识符	功能
15	~SAR_LPOWER	必须为 SAR_LPOWER 的取反值
14	~WDT_CFG	必须为 WDT_CFG 的取反值
9	~ILOPLEN	必须为 ILOPLEN 的取反值
8	~ EMCEN	必须为 EMCEN 的取反值
7	SAR_LPOWER	SAR_ADC 低功耗模式配置位 1: SAR_ADC 配置为正常功耗模式 0: SAR_ADC 配置为低功耗模式（默认）
6	WDT_CFG	WDT 模块使能和内部 10K 低速振荡器使能配置位 1: WDT 模块使能和内部 10K 低速振荡器使能由软件配置（默认） 0: WDT 模块使能和内部 10K 低速振荡器使能固定打开，软件无法修改。
5:2	RESERVE	保留
1	ILOPLEN	非法指令复位使能 0: 使能非法指令复位（默认） 1: 禁止非法指令复位
0	EMCEN	EMC 复位使能 0: 使能 EMC 复位（默认） 1: 禁止 EMC 复位

3.14.3 ICD 功能配置选项

地址：0xF802。

表 3-148

Bit-15	Bit-14	Bit-13	Bit-12	Bit-11	Bit-10	Bit-9	Bit-8
Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
				ICD_CFG[3:0]			

表 3-149

位地址	标识符	功能
15:4	RESERVE	保留
3:0	ICD_CFG[3:0]	ICD 模式使能选项

		0000: 使能 ICD 功能 其他: 禁止 ICD 功能 注: 在 ICD 功能使能并进入 sleep 时 24MHZ 时钟不关闭, 这是为了保证在 sleep 时能进入 ICD 调试模式。如果想让 sleep 时 24MHZ 关闭, 必须配置禁止 ICD 功能。
--	--	---

3.14.4 FLASH 主程序区擦写加密位

地址: 0xFA13。

表 3-150

Bit-15	Bit-14	Bit-13	Bit-12	Bit-11	Bit-10	Bit-9	Bit-8
~USR_W_SEC [7:0]							
Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
USR_W_SEC [7:0]							

表 3-151

位地址	标识符	功能
15	~USR_W_SEC[7]	表示的取反 USR_W_SEC[7]
14	~USR_W_SEC[6]	表示的取反 USR_W_SEC[6]
13	~USR_W_SEC[5]	表示的取反 USR_W_SEC[5]
12	~USR_W_SEC[4]	表示的取反 USR_W_SEC[4]
11	~USR_W_SEC[3]	表示的取反 USR_W_SEC[3]
10	~USR_W_SEC[2]	表示的取反 USR_W_SEC[2]
9	~USR_W_SEC[1]	表示的取反 USR_W_SEC[1]
8	~USR_W_SEC[0]	表示的取反 USR_W_SEC[0]
7	USR_W_SEC[7]	空间 (7000H~7FFFH) 加密位: 0 表示加密, 用户模式对应空间不能进行擦写操作 1 表示不加密, 用户模式对应空间能进行擦写操作 (默认)
6	USR_W_SEC[6]	空间 (6000H~6FFFH) 加密位: 0 表示加密, 用户模式对应空间不能进行擦写操作 1 表示不加密, 用户模式对应空间能进行擦写操作 (默认)
5	USR_W_SEC[5]	空间 (5000H~5FFFH) 加密位: 0 表示加密, 用户模式对应空间不能进行擦写操作 1 表示不加密, 用户模式对应空间能进行擦写操作 (默认)
4	USR_W_SEC[4]	空间 (4000H~4FFFH) 加密位: 0 表示加密, 用户模式对应空间不能进行擦写操作 1 表示不加密, 用户模式对应空间能进行擦写操作 (默认)
3	USR_W_SEC[3]	空间 (3000H~3FFFH) 加密位: 0 表示加密, 用户模式对应空间不能进行擦写操作 1 表示不加密, 用户模式对应空间能进行擦写操作 (默认)
2	USR_W_SEC[2]	空间 (2000H~2FFFH) 加密位: 0 表示加密, 用户模式对应空间不能进行擦写操作 1 表示不加密, 用户模式对应空间能进行擦写操作 (默认)
1	USR_W_SEC[1]	空间 (1000H~1FFFH) 加密位: 0 表示加密, 用户模式对应空间不能进行擦写操作 1 表示不加密, 用户模式对应空间能进行擦写操作 (默认)

0	USR_W_SEC[0]	空间（0000H~0FFFH）加密位： 0 表示加密，用户模式对应空间不能进行擦写操作 1 表示不加密，用户模式对应空间能进行擦写操作（默认）
---	--------------	---

当写/擦除操作区域所对应 USR_W_SEC[X]值为 1'b1 时，则可以对该地址 sector 进行擦除或写操作；当写/擦除操作区域所对应 USR_W_SEC[X]值为 1'b0，则写/擦除操作失败，并且操作标志位 ISPOF 置 1，软件清零；

4 MCU 指令集

4.1 基础指令汇总

表 4-1 MCU 指令集

指令	操作	指令周期	标志位
ADDLW k	$[W] \leftarrow [W] + k$	1	C,DC,Z
ADDPCW	$[PC] \leftarrow [PC] + 1 + [W]$	2	~
ADDWF f,d	$[Destination] \leftarrow [f] + [W]$	1	C,DC,Z
ADDWFC f,d	$[Destination] \leftarrow [f] + [W] + C$	1	C,DC,Z
ANDLW k	$[W] \leftarrow [W] \text{ AND } k$	1	Z
ANDWF f,d	$[Destination] \leftarrow [W] \text{ AND } [f]$	1	Z
BCF f,b	$[f] \leftarrow 0$	1	~
BSF f,b	$[f] \leftarrow 1$	1	~
BTFSC f,b	Jump if $[f] = 0$	1/2	~
BTFSS f,b	Jump if $[f] = 1$	1/2	~
CALL k	Push PC+1 and Goto K $0 \leq K \leq 1FFFh$	2	~
CLRF f	$[f] \leftarrow 0$	1	Z
CLRWDT	Clear watch dog timer	1	~
COMF f,d	$[f] \leftarrow \text{NOT}([f])$	1	Z
DAW	Decimal Adjust W	1	C,DC
DECF f,d	$[Destination] \leftarrow [f] - 1$	1	Z
DECFSZ f,d	$[Destination] \leftarrow [f] - 1$, jump if the result is zero	1/2	Z
GOTO k	$PC \leftarrow k$	2	~
HALT	CPU Stop	1	~
INCF f,d	$[Destination] \leftarrow [f] + 1$	1	Z
INCFSZ f,d	$[Destination] \leftarrow [f] + 1$, jump if the result is zero	1/2	Z
IORLW k	$[W] \leftarrow [W] \text{ OR } k$	1	Z
IORWF f,d	$[Destination] \leftarrow [W] \text{ OR } [f]$	1	Z
MOVFW f	$[W] \leftarrow [f]$	1	~
MOVLW k	$[W] \leftarrow k$	1	~
MOVWF f	$[f] \leftarrow [W]$	1	~
NOP	No operation	1	~
POP	Pop Work,BSR and Status	2	~
PUSH	Push Work,BSR and Status	2	~
RETFIE	Pop PC and GIE =1	2	~
RETLW k	RETURN and W=k	2	~
RETURN	POP PC	2	~
RLF f,d	$[Destination<n+1>] \leftarrow [f<n>]$	1	C,Z
RRF f,d	$[Destination<n-1>] \leftarrow [f<n>]$	1	C,Z
SLEEP	STOP OSC	1	PD
SUBLW k	$[W] \leftarrow k - [W]$	1	C,DC,Z
SUBWF f,d	$[Destinnation] \leftarrow [f] - [W]$	1	C,DC,Z
SUBWFC f,d	$[Destinnation] \leftarrow [f] - [W] - 1 + C$	1	C,DC,Z
SWAPF f,d	swap f	1	~
XORLW k	$[W] \leftarrow [W] \text{ XOR } k$	1	Z
XORWF f,d	$[Destination] \leftarrow [W] \text{ XOR } [f]$	1	Z
RL	$[Destination<n+1>] \leftarrow [f<n>]$	1	Z
RR	$[Destination<n-1>] \leftarrow [f<n>]$	1	Z

SETF	[f]←FF	1	Z
------	--------	---	---

参数说明:

f:数据存储器地址(00h~1FFh)

W:工作寄存器

k: 立即数

d:目标地址选择: d=0 结果保存在工作寄存器, d=1: 结果保存在数据存储器 f 单元

b:位选择(0~7)

[f]:f 地址的内容

PC:程序计数器

C:进位标志

DC:半加进位标志

Z:结果为零标志

PD:睡眠标志位

TO:看门狗溢出标志

WDT:看门狗计数器

4.2 扩展指令汇总

扩展指令在不同项目可选择支持或者不支持。

表 4-2 扩展指令集

指令	操作	指令周期	标志位
CALLL k	Push PC+1 and Goto K $0 \leq K \leq 7FFFh$	3	~
GOTOL K	$PC \leftarrow k$ $0 \leq K \leq 7FFFh$	3	~
ADDLFSR d,K	FSR0/FSR0H or FSR1/FSR1H + K $0 \leq K \leq 127$	1	C,Z
SUBLFSR d,K	FSR0/FSR0H or FSR1/FSR1H - K $0 \leq K \leq 127$	1	C,Z
MOVFWL f	$[W] \leftarrow [f]$ $0 \leq f \leq 3FFFh$	1	~
MOVWFL f	$[f] \leftarrow [W]$ $0 \leq f \leq 3FFFh$	1	~

4.3 基础指令描述

4.3.1 ADDLW

表 4-3

ADDLW	加立即数到工作寄存器
指令格式	ADDLW K (0<=K<=FFH)
操作	$(W) \leftarrow (W) + K$
标志位	C, DC, Z
描述	工作寄存器的内容加上立即数 K 结果保存到工作寄存器中
周期	1
例子 ADDLW 08H	在指令执行之前: W=08H 在指令执行之后: W=10H

4.3.2 ADDPCW

表 4-4

ADDPCW	将 W 的内容加到 PC 中
指令格式	ADDPCW
操作	$(PC) \leftarrow (PC) + 1 + (W)$ 当 $(W) < 7FH$ $(PC) \leftarrow (PC) + 1 + (W) - 100H$ 其余
标志位	没有
描述	将地址 $PC + 1 + W$ 加载到 PC 中
周期	2
例子 1 ADDPCW	在指令执行之前: W=7FH, PC=0212H 指令执行之后: PC=0292H
例子 2 ADDPCW	在指令执行之前: W=80H, PC=0212H 指令执行之后: PC=0193H
例子 3 ADDPCW	在指令执行之前: W=FEH, PC=0212H 指令执行之后: PC=0211H

4.3.3 ADDWF

表 4-5

ADDWF	加工作寄存器到 f
指令格式	ADDWF f,d 0<=f<=1FFH d=0,1
操作	$[目标地址] \leftarrow (f) + (W)$
标志位	C, CD, Z
描述	将 f 的内容和工作寄存器的内容加到一起。 如果 d 是 0, 结果保存到工作寄存器中。 如果 d 是 1, 结果保存到 f 中。

周期	1
例子 1 ADDWF f 0	指令执行之前: f=C2H W=17H 在指令执行之后 f=C2H W=D9H
例子 2 ADDWF f 1	指令执行之前 f=C2H W=17H 指令执行之后 f=D9H W=17H

4.3.4 ADDWFC

表 4-6

ADDWFC	将 W f 和进位位相加
指令格式	ADDWFC f, d 0<=f<=1FFH d=0,1
操作	(目标地址)<←(f)+(W)+C
标志位	C, DC, Z
描述	将工作寄存器的内容和 f 的内容以及进位位相加 当 d 为 0 时结果保存到工作寄存器 当 d 为 1 时结果保存到 f 中
周期	1
例子 ADDWFC f, 1	指令执行之前 C=1 f=02H W=4DH 指令执行之后 C=0 f=50H W=4DH

4.3.5 ANDLW

表 4-7

ANDLW	工作寄存器与立即数相与
指令格式	ANDLW K 0<=K<=FFH
操作	(W)<←(W) AND K
标志位	Z
描述	将工作寄存器的内容与 8bit 的立即数相与，结果保存到工作寄存器中。
周期	1
例子 ANDLW 5FH	在指令执行之前 W=A3H 在指令执行之后 W=03H

4.3.6 ANDWF

表 4-8

ANDWF	将工作寄存器和 f 的内容相与
指令格式	ANDWF f, d 0<=f<=1FFH d=0,1
操作	(目标地址)<←(W) AND (f)
标志位	Z
描述	将工作寄存器的内容和 f 的内容相与

	如果 d 为 0 结果保存到工作寄存器中 如果 d 为 1 结果保存到 f 中
周期	1
例子 1 ANDWF f, 0	在指令执行之前 W=0FH f=88H 在指令执行之后 W=08H f=88H
例子 2 ANDWF f, 1	在指令执行之前 W=0FH f=88H 在指令执行之后 W=0FH f=08H

4.3.7 BCF

表 4-9

BCF	清除 f 的某一位
指令格式	BCF f, b 0<=f<=1FFH 0<=b<=7
操作	(f[b])←0
标志位	无
描述	F 的第 b 位置为 0
周期	1
例子 BCF FLAG 2	指令执行之前: FLAG=8DH 指令执行之后: FLAG=89H

4.3.8 BSF

表 4-10

BSF	F 的 b 位置 1
指令格式	BSF f, b 0<=f<=1FFH 0<=b<=7
操作	(f[b])←1
标志位	无
描述	将 f 的 b 位置 1
周期	1
例子 BSF FLAG 2	在指令执行之前 FLAG=89H 在指令执行之后 FLAG=8DH

4.3.9 BTFSC

表 4-11

BTFSC	如果 bit 测试为 0 则跳转
指令格式	BTFSC f, b 0<=f<=1FFH 0<=b<=7
操作	Skip if (f[b])=0
标志位	无
描述	如果 f 的 bit 位是 0, 下一条取到的指令将被丢到, 然后执行一条空指令组成

	一个两周期的指令。
周期	无跳转则为 1 个指令周期，否则 2 个指令周期
例子 NODE BTFSC FLAG 2 OP1: OP2:	在程序执行以前 PC=address(NODE) 指令执行之后 If(FLAG[2])=0 PC=address(OP2) If(FLAG[2])=1 PC=address(OP1)

4.3.10 BTFSS

表 4-12

BTFSS	如果 bit 测试为 1，则跳转
指令格式	BTFSS f, b 0<=f<=1FFH 0<=b<=7
操作	Skip if (f[b])=1
标志位	无
描述	如果 f 的 bit 位是 1，下一条取到的指令将被丢到，然后执行一条空指令组成一个两周期的指令。
周期	无跳转则为 1 个指令周期，否则 2 个指令周期
例子 NODE BTFSS FLAG 2 OP1: OP2:	在程序执行以前 PC=address(NODE) 指令执行之后 If(FLAG[2])=0 PC=address(OP1) If(FLAG[2])=1 PC=address(OP2)

4.3.11 CALL

表 4-13

CALL	子程序调用
指令格式	CALL K 0<=K<=FFFH
操作	(top stack)<—PC+1 PC<—K
标志位	无
描述	子程序调用，先将 PC+1 压入堆栈，然后把立即数地址下载到 PC 中。
周期	2

4.3.12 CLRf

表 4-14

CLRf	清除 f
指令格式	CLRf f 0<=f<=1FFH
操作	(f)<—0
标志位	Z
描述	将 f 的内容清零

周期	1
例子 CLRF WORK	在指令执行之前 WORK=5AH 在指令执行之后 WORK=00H

*注。当 clrf 清除 status 寄存器时，标志位 Z 不会置高

4.3.13 CLRWDT

表 4-15

CLRWDT	清除看门狗定时器
指令格式	CLRWDT
操作	看门狗计数器清零
标志位	无
描述	清除看门狗定时器
周期	1
例子 CLRWDT	指令执行之后 WDT=0

4.3.14 COMF

表 4-16

COMF	f 取反
指令格式	COMF f, d 0<=f<=1FFH d=0,1
操作	(目的地址)←NOT(f)
标志位	Z
描述	将 f 的内容取反， 当 d 为 0 时，结果保存到工作寄存器中， 当 d 为 1 时，结果保存到 f 中。
周期	1
例子 COMF f, 0	在指令执行之前 W=88H, f=23H 在指令执行之后 W=DCH, f=23H
例子 2 COMF f, 1	在指令执行之前 W=88H, f=23H 在指令执行之后 W=88H, f=DCH

4.3.15 DAW

表 4-17

DAW	十进制调整 W 寄存器
指令格式	DAW
操作	十进制调整 W 寄存器
标志位	C,DC
描述	一般与加法一起使用。

	如果低半字节的值大于 9 或 DC 为 1 时，低半字节加 6； 如果高半字节的值大于 9 或 C 为 1 时，高半字节加 6
周期	1
例子 若 W=25H; ADDLW 39H DAW	在 DAW 指令执行之前 W=25H+39H=64=5EH 在指令执行之后 W= (64) BCD 25H + 39H ----- 5EH + 06H ----- 64H

4.3.16 DECF

表 4-18

DECF	f 减 1
指令格式	DECF f, d 0<=f<=1FFH d=0,1
操作	(目的地址)<-(f)-1
标志位	Z
描述	F 的内容减 1 当 d 为 0 时，结果保存到工作寄存器中 当 d 为 1 时，结果保存到 f 中。
周期	1
例子 DECF f, 0	在指令执行之前 W=88H f=23H 在指令执行之后 W=22H f=23H
例子 2 DECF f, 1	在指令执行之前 W=88H f=23H 在指令执行之后 W=88H f=22H

4.3.17 DECFSZ

表 4-19

DECFSZ	f 减 1 如果为 0 则跳转
指令格式	DECFSZ f, d 0<=f<=1FFH d=0,1
操作	(目的地址)<-(f)-1,如果结果为 0 跳转
标志位	Z
描述	f 的内容减 1。 如果 d 为 0，结果保存到工作寄存器中。 如果 d 为 1，结果保存到 f 中 如果结果为 0，下一条已经取到的指令将被丢掉，然后插入一条 NOP 指令组成一个两个周期的指令。
周期	无跳转则为 1 个指令周期，否则 2 个指令周期

例子 Node DECFSZ FLAG, 1 OP1: OP2:	在指令执行之前 PC=address(Node) 在指令执行之后 (FLAFG)=(FLAG)-1 If(FLAG)=0 PC=address(OP2) If(FLAG)!=0 PC=address(OP1)
---	---

4.3.18 GOTO

表 4-20

GOTO	无条件跳转
指令格式	GOTO K 0<=K<=FFFH
操作	PC<←K
标志位	无
描述	立即地址载入 PC
周期	2

4.3.19 HALT

表 4-21

HALT	停止 CPU 时钟
指令格式	HALT
操作	CPU 停止
标志位	无
描述	CPU 时钟停止，晶振仍然工作，CPU 能够通过内部或者外部中断重启。
周期	1

4.3.20 INCF

表 4-22

INCF	f 加 1
指令格式	INCF f, d 0<=f<=1FFH d=0,1
操作	(目的地址)<←(f)+1
标志位	Z
描述	f 加 1 如果 d 为 0，结果保存到工作寄存器中 如果 d 为 1，结果保存到 f 中。
周期	1
例子 INCF f, 0	在指令执行之前 W=88H f=23H 在指令执行之后 W=24H f=23H
例子 2 INCF f, 1	在指令执行之前 W=88H f=23H 在指令执行之后

W=88H f=24H

4.3.21 INCFSZ

表 4-23

INCFSZ	f 加 1, 如果结果为 0 跳转
指令格式	INCFSZ f, d 0<=f<=1FFH d=0,1
操作	(目的地址)<-(f)+1 如果结果为 0 就跳转
标志位	Z
描述	f 的内容加 1。 如果 d 为 0, 结果保存到工作寄存器中。 如果 d 为 1, 结果保存到 f 中 如果结果为 0, 下一条已经取到的指令将被丢掉, 然后插入一条 NOP 指令组成一个两个周期的指令。
周期	无跳转则为 1 个指令周期, 否则 2 个指令周期
例子 Node INCFSZ FLAG, 1 OP1: OP2:	在指令执行之前 PC=address(Node) 在指令执行之后 (FLAG)=(FLAG)+1 If(FLAG)=0 PC=address(OP2) If(FLAG)!=0 PC=address(OP1)

4.3.22 IORLW

表 4-24

IORLW	工作寄存器与立即数或
指令格式	IORLW K 0<=K<=FFH
操作	(W)<-(W) K
标志位	Z
描述	立即数与工作寄存器的内容或。结果保存到工作寄存器中。
周期	1
例子 IORLW 85H	在指令执行之前 W=69H 在指令执行之后 W=EDH

4.3.23 IORWF

表 4-25

IORWF	f 与工作寄存器或
指令格式	IORWF f, d 0<=f<=1FFH d=0,1
操作	(目的地址)<-(W) (f)
标志位	Z
描述	f 和工作寄存器或 当 d 为 0 时, 结果保存到工作寄存器中 当 d 为 1 时, 结果保存到 f 中
周期	1

例子 IORWF f,1	在指令执行前 W=88H f=23H 在指令执行后 W=88H f=ABH
-----------------	--

4.3.24 MOVFW

表 4-26

MOVFW	将 f 寄存器值传送到工作寄存器
指令格式	MOVFW f 0<=f<=1FFH
操作	(W)<←(f)
标志位	无
描述	将数据从 f 传送到工作寄存器
周期	1
例子 MOVFW f	在指令执行之前 W=88H f=23H 在指令执行之后 W=23H f=23H

4.3.25 MOVLW

表 4-27

MOVLW	将立即数传送到工作寄存器中
指令格式	MOVLW K 0<=K<=FFH
操作	(W)<←K
标志位	无
描述	将 8bit 的立即数传送到工作寄存器中
周期	1
例子 MOVLW 23H	在指令执行之前 W=88H 在指令执行之后 W=23H

4.3.26 MOVWF

表 4-28

MOVWF	将工作寄存器的值传送到 f 中
指令格式	MOVWF f 0<=f<=1FFH
操作	(f)<←(W)
标志位	无
描述	将工作寄存器的值传送到 f 中
周期	1
例子 MOVWF f	在指令执行之前 W=88H f=23H 在指令执行之后 W=88H f=88H

4.3.27 NOP

表 4-29

NOP	无操作
指令格式	NOP
操作	无操作
标志位	无
描述	无操作
周期	1

4.3.28 PUSH

表 4-30

PUSH	把 work 寄存器、status 寄存器的 DC、C、Z 和 BSR 寄存器的 PAGE[2:0]入栈保护
指令格式	PUSH
操作	(top stack)←work/status/BSR
标志位	无
描述	把 work 寄存器、status 寄存器的 DC、C、Z 和 BSR 寄存器的 PAGE[2:0]入栈处理，支持 12 级堆栈，不同于 PC 堆栈；其中状态寄存器不包括 LVD36，LVD24，PD 和 TO。
周期	2

4.3.29 POP

表 4-31

POP	把 work 和 status 寄存器出栈处理
指令格式	POP
操作	(Top Stack)⇒work/status Pop Stack
标志位	无
描述	把当前栈顶的值做出栈处理，分别更新 work 和 status 寄存器，支持 12 级堆栈，不同于 PC 堆栈；其中状态寄存器不包括 LVD36，LVD24，PD 和 TO。
周期	2

4.3.30 RETFIE

表 4-32

RETFIE	从中断返回
指令格式	RETFIE
操作	(Top Stack)⇒PC Pop Stack 1⇒GIE
标志位	无
描述	PC 从堆栈顶部得到，然后出栈，设置全局中断使能位为 1

周期	2
----	---

4.3.31 RETLW

表 4-33

RETLW	返回，并将立即数送到工作寄存器中
指令格式	RETLW K 0<=K<=FFH
操作	(W)<←K (Top Stack)=>PC Pop Stack
标志位	无
描述	将 8bit 的立即数送到工作寄存器中，PC 值从栈顶得到，然后出栈
周期	2

4.3.32 RETURN

表 4-34

RETURN	从子程序返回
指令格式	RETURN
操作	(Top Stack)=>PC Pop Stack
标志位	无
描述	PC 值从栈顶得到，然后出栈
周期	2

4.3.33 RLF

表 4-35

RLF	带进位左移
指令格式	RLF f, d 0<=f<=1FFH d=0,1
操作	(目标地址[n+1])<←(f[n]) (目标地址[0])<←C C<←(f[7])
标志位	C, Z
描述	F 带进位位左移一位 如果 d 为 0，结果保存到工作寄存器 如果 d 为 1，结果保存到 f 中
周期	1
例子 RLF f, 1	在指令执行之前 C=0 W=88H f=E6H 在指令执行之后 C=1 W=88H f=CCH

4.3.34 RRF

表 4-36

RRF	带进位右移
-----	-------

指令格式	RRF f, d 0<=f<=1FFH d=0,1
操作	(目标地址[n-1])<←(f[n]) (目标地址[7])<←C C<←(f[0])
标志位	C
描述	F 带进位位右移一位 如果 d 为 0, 结果保存到工作寄存器 如果 d 为 1, 结果保存到 f 中
周期	1
例子 RRF f, 0	在指令执行之前 C=0 W=88H f=95H 在指令执行之后 C=1 W=4AH f=95H

4.3.35 SLEEP

表 4-37

SLEEP	晶振停止
指令格式	SLEEP
操作	CPU 晶振停止
标志位	PD
描述	CPU 晶振停止。CPU 通过中断唤醒
周期	1

4.3.36 SUBLW

表 4-38

SUBLW	立即数减工作寄存器的值
指令格式	SUBLW K 0<=K<=FFH
操作	(W)<←K-(W)
标志位	C, DC, Z
描述	8bit 的立即数减去工作寄存器的值, 结果保存到工作寄存器中
周期	1
例子 SUBLW 02H	在指令执行之前 W=01H 在指令执行之后 W=01H C=1(代表没有借位) Z=0(代表结果非零)
例子 2 SUBLW 02H	在指令执行之前 W=02H 在指令执行之后 W=00H C=1(代表没有借位) Z=1(代表结果为零)
例子 2 SUBLW 02H	在指令执行之前 W=03H 在指令执行之后 W=FFH C=0(代表有借位) Z=0(代表结果非零)

4.3.37 SUBWF

表 4-39

SUBWF	f 的值减工作寄存器的值
指令格式	SUBWF f, d 0<=f<=1FFH d=0,1
操作	(目标地址)<-(f)-(W)
标志位	C, DC, Z
描述	f 的值减去工作寄存器的值。 如果 d 为 0, 结果保存到工作寄存器 如果 d 为 1, 结果保存到 f 中
周期	1
例子 SUBWF f, 1	在指令执行之前 f=33H W=01H 在指令执行之后 f=32H C=1 Z=0
例子 2 SUBWF f, 1	在指令执行之前 f=01H W=01H 在指令执行之后 f=00H C=1 Z=1
例子 3 SUBWF f, 1	在指令执行之前 f=04H W=05H 在指令执行之后 f=FFH C=0 Z=0

4.3.38 SUBWFC

表 4-40

SUBWFC	带借位的减法
指令格式	SUBWFC f, d 0<=f<=1FFH d=0,1
操作	(目标地址)<-(f)-(W)-1+C
标志位	C, DC, Z
描述	f 的值减去工作寄存器的值 如果 d 为 0, 结果保存到工作寄存器 如果 d 为 1, 结果保存到 f 中
周期	1
例子 SUBWFC f, 1	在指令执行之前 W=01H f=33H C=1 在指令执行之后 f=32H C=1 Z=0
例子 2 SUBWFC f, 1	在指令执行之前 W=01H f=02H C=0 在指令执行之后 f=00H C=1 Z=1
例子 3 SUBWFC f, 1	在指令执行之前 W=05H f=04H C=0 在指令执行之后 f=FEH C=0 Z=0

4.3.39 SWAPF

表 4-41

SWAPF	交换寄存器的值
指令格式	SWAPF f, d 0<= f <=1FFH d=0,1
操作	(des[3:0]) $\leftarrow$ f[7:4] (des[7:4]) $\leftarrow$ f[3:0]
标志位	无
描述	把 f 寄存器的高 4 位数据给目标寄存器的低 4 位； 把 f 寄存器的低位数据给目标寄存器的高 4 位 d 为 1 时，f 寄存器为目标寄存器；否则，w 寄存器为目标寄存器
周期	1
例子 SWAPF f,1	在指令执行之前 f=ACH 在指令执行之后 f=CAH

4.3.40 XORLW

表 4-42

XORLW	工作寄存器的值与立即数异或
指令格式	XORLW K 0<= K <=FFH
操作	(W) $\leftarrow$ (W) $\wedge$ K
标志位	Z
描述	8bit 的立即数与工作寄存器的值异或，结果保存在工作寄存器中
周期	1
例子 XORLW 5FH	在指令执行之前 W=ACH 在指令执行之后 W=F3H

4.3.41 XORWF

表 4-43

XORWF	f 的值与工作寄存器的值异或
指令格式	XORWF f, d 0<= f <=1FFH d=0,1
操作	(目标地址) $\leftarrow$ (W) $\wedge$ (f)
标志位	Z
描述	F 的值与工作寄存器的值异或， 当 d 为 0 时，结果保存到工作寄存器中 当 d 为 1 时，结果保存到 f 中
周期	1
例子 XORWF f, 1	在指令执行之前 W=ACH f=5FH 在指令执行之后 f=F3H

4.3.42 RL

表 4-44

RL	不带进位左移
指令格式	RL f, d 0<=f<=1FFH d=0,1
操作	(目标地址[n+1])<←(f[n]) (目标地址[0])<←0
标志位	Z
描述	F 不带进位位左移一位 如果 d 为 0, 结果保存到工作寄存器 如果 d 为 1, 结果保存到 f 中
周期	1
例子 RL f, 1	在指令执行之前 C=0 W=88H f=E6H 在指令执行之后 C=0 W=88H f=CCH

4.3.43 RR

表 4-45

RR	不带进位右移
指令格式	RR f, d 0<=f<=1FFH d=0,1
操作	(目标地址[n-1])<←(f[n]) (目标地址[7])<←0
标志位	Z
描述	F 不带进位位右移一位 如果 d 为 0, 结果保存到工作寄存器 如果 d 为 1, 结果保存到 f 中
周期	1
例子 RR f, 0	在指令执行之前 C=0 W=88H f=95H 在指令执行之后 C=0 W=4AH f=95H

4.3.44 SETF

表 4-46

SETF	向 f 寄存器写入 FFH
指令格式	SETF f 0<=f<=1FFH
操作	(f)<←FFH
标志位	Z
描述	向 f 寄存器写入 FFH
周期	1
例子 SETF WORK	在指令执行之前 WORK=5AH 在指令执行之后 WORK=FFH

4.4 扩展指令描述

4.4.1 CALLL

表 4-47

CALLL	长寻址子程序调用，可访问 32K word 空间
指令格式	CALLL K $0 \leq K \leq 7FFFH$
操作	$(top\ stack) \leftarrow PC+1$ $PC \leftarrow K$
标志位	无
描述	长寻址子程序调用，先将 PC+1 压入堆栈，然后把立即数地址下载到 PC 中。
周期	3
例子 CALLL function1	调用 function1 函数

4.4.2 GOTOL

表 4-48

GOTOL	长寻址无条件跳转，可访问 32K word 空间
指令格式	GOTOL K $0 \leq K \leq 7FFFH$
操作	$PC \leftarrow K$
标志位	无
描述	立即地址载入 PC
周期	3

4.4.3 ADDLFSR

表 4-49

ADDLFSR	对 FSR0/FSR1 值加 K
指令格式	ADDLFSR d,K $d=0,1$ $0 \leq k \leq 127$
操作	$(fsrx) \leftarrow (fsrx) + K$ ($x=0,1$)
标志位	C, Z
描述	对 FSR0/FSR1 寄存器值进行加法操作，其中 FSR0/FSR1 包括低位和高位地址，即 FR0 和 FSR0H，FSR1 和 FSR1H
周期	1
例子 ADDLFSR 0,11H	在指令执行之前 W=88H fsr0[12:0]=780H 在指令执行之后 W=88H fsr0[12:0]=791H

4.4.4 SUBLFSR

表 4-50

SUBLFSR	对 FSR0/FSR1 值减 K
指令格式	SUBLFSR d,K $d=0,1$ $0 \leq k \leq 127$
操作	$(fsrx) \leftarrow (fsrx) - K$ ($x=0,1$)

标志位	C, Z
描述	对 FSR0/FSR1 寄存器值进行减法操作，其中 FSR0/FSR1 包括低位和高位地址，即 FR0 和 FSR0H，FSR1 和 FSR1H
周期	1
例子 SUBLFSR 1,11H	在指令执行之前 W=88H fsr1[12:0]=780H 在指令执行之后 W=88H fsr1[12:0]=76fH

4.4.5 MOVFWL

表 4-51

MOVFWL	将 f 寄存器值传送到工作寄存器，f 寻址范围为 0~8K Bytes
指令格式	MOVFWL f 0<=f<=1FFFH
操作	(W)<-(f)
标志位	无
描述	将数据从 f 传送到工作寄存器
周期	2
例子 MOVFWL 7FFH	在指令执行之前 W=88H f=23H 在指令执行之后 W=23H f=23H

4.4.6 MOVWFL

表 4-52

MOVWFL	将工作寄存器的值传送到 f 中，f 寻址范围为 0~8K Bytes
指令格式	MOVWFL f 0<=f<=1FFFH
操作	(f)<-(W)
标志位	无
描述	将工作寄存器的值传送到 f 中
周期	2
例子 MOVWFL 7FFH	在指令执行之前 W=88H f=23H 在指令执行之后 W=88H f=88H

5 电气特性

5.1 极限值

表 5-1

参数	范围	单位
电源 VDD	-0.3~6.0	V
引脚输入电压	-0.3~VDD+0.3	V
工作温度	-40~+85(环境温度)	℃
	-40~105 (结温)	
存储温度	-55~+150	℃
焊接温度, 时间	220 ℃, 10 秒	

5.2 直流特性 (VDD = 3.3V, TA = 25 ℃, 如无其他说明则都是此条件)

表 5-2

符号	参数	测试条件	最小值	典型值	最大值	单位
VDD	工作电压	25 ℃	2.2	3.3	5.5	V
Tcpu	指令周期			83.3		ns
VIH	数字输入高电平	施密特关	0.55VDD		VDD+0.3	V
		施密特开	0.7VDD		VDD+0.3	
VIL	数字输入低电平	施密特关	-0.3		0.45VDD	V
		施密特开	-0.3		0.3VDD	
VIH(FCP)	数字输入高电平	FCP 通信信号	1.2			V
VIL(FCP)	数字输入低电平	FCP 通信信号			0.6	V
RPU	上拉电阻	VDD=3.3V		40		KΩ
	上拉电阻	VDD=2.2V		70		KΩ
ILK	输入漏电流	0<VIN<VDD 开漏模式或输入模式	-1		1	uA
IOH	高电平输出电流 (VPP 除外)	VOH=0.9VDD; VDD=3.3V	4	6	10	mA
IOL	低电平输出电流	VOL=0.1VDD; VDD=3.3V	2.5	5	9	mA
VPOR	上电复位释放电压	电源上电	1.85	2	2.2	V
VPDR	掉电复位电压	电源掉电	1.8	1.95	2.15	V
VPP_THR	烧录阈值电压		5.7	6.4	7.5	V
VLVD	低电压检测电压	2.4V 检测点;	2.3	2.4	2.45	V
		2.1V 检测点;	2.05	2.1	2.15	
fHIRC	内置 24MHz			24		MHz

	RC 时钟频率					
Ef _{HIRC}	24M 时钟频率误差	校准误差 (25℃, 3.3V)	-1		1	%
		随电压温度的变化 (-40℃~105℃, 2.2V~5.5V)	-2.5		2.5	
f _{LRC}	内置 10K RC 时钟频率			10		KHz
Ef _{LRC}	内置 10K RC 时钟频率误差	校准误差(25℃, 3.3V)	-3		3	%
		随电压温度的变化 (-40℃~105℃, 2.2V~5.5V)	-10		10	%
T _{INT0/1}	中断触发脉宽	25℃, 5V	T _{cpu}			ns
VREF	1.0V 参考电压	-40℃~105℃, 2.2V~5.5V	-2%	1.0	2%	V
	2.5V 参考电压	-40℃~105℃, 2.2V~5.5V	-2%	2.5	2%	V
	VDD			VDD		V
IDD1	sleep 模式电流	VDD=3.3V, 关掉 WDT		10		uA
		VDD=3.3V, 打开 WDT		12		uA
		VDD=3.3V, 打开 type-C 检测		50		uA
IDD2	工作电流	内部 RC 振荡器模式, F=24MHz, VDD=3.3V, f _{cpu} =fosc/1		4		mA
		内部 RC 振荡器模式, F=16MHz, VDD=3.3V, f _{cpu} =fosc/2		2		
		内部 RC 振荡器模式, F=16MHz, VDD=3.3V, f _{cpu} =fosc/4		1		

5.3 ADC 特性 (VDD = 3.3V, TA = 25℃, 如无其他说明则都是此条件)

表 5-3

符号	参数	条件	最小值	典型值	最大值	单位
VDD	工作电压	-40℃~+105℃	2.2	5	5.5	V
I _{DDA(ADC)} (1)	功耗	VDD=3.3V(VDD 作参考)		0.6		mA
V _{CMIN}	模拟输入共模电压	全差分模式:		VREF/2		V
V _{AIN}	模拟输入电压 (AINP-AINN)	全差分模式	-VREF		VREF	V
		单端模式	0		VREF	V
VREF	参考电压	全差分模式(固定三档 1V/2.5V/VDD)		1.0	VDD	V
		单端模式(固定三档 1V/2.5V/VDD)		1.0	VDD	V
R _{ADC} ⁽²⁾	采样开关电阻			3.1		kΩ
C _{ADC} ⁽²⁾	内部采样和保持电容				12	pF
R _{AIN} ⁽³⁾	外部输入阻抗			0.3	5	kΩ

$f_{\text{ADC}}^{(2)}$	ADC 时钟频率	$2.2\text{V} \leq \text{VDD} \leq 5.5\text{V}$	1.5	1	3	MHz
$f_s^{(2)}$	ADC 采样率		53.6	150	187.5	kHz
$t_s^{(3)}$	采样时间	$2.4\text{V} \leq \text{VDD} \leq 5.5\text{V}$, $R_{\text{AIN}} \leq 0.3\text{ k}\Omega$	1.33			μs
		$2.4\text{V} \leq \text{VDD} < 5.5\text{V}$, $R_{\text{AIN}} \leq 1\text{ k}\Omega$	2.67			μs
		$2.4\text{V} \leq \text{VDD} < 5.5\text{V}$, $R_{\text{AIN}} \leq 5\text{ k}\Omega$	5.33			μs
		$2.2\text{V} \leq \text{VDD} < 2.4\text{V}$, $R_{\text{AIN}} \leq 5\text{ k}\Omega$	16			μs
$t_{\text{CONV}}^{(2)}$	总转换时间	采样时间+逐次趋近时间	5.33	6.67	18.67	μs
Resolution ⁽⁴⁾	分辨率			12		Bit
Offset ⁽⁶⁾	失调电压	全差分模式下, 通过 sar_offex 两次测量降低 offset (AD 十进制, sar_offex=0)- (AD 十进制, sar_offex=1)	-3		3	mV
INL ⁽⁵⁾	积分非线性	单端模式, $f_{\text{ADC}} = 2\text{MHz}$, $t_s = 4\mu\text{s}$, $2.4\text{V} \leq \text{VDD} \leq 5.5\text{V}$, $R_{\text{AIN}} \leq 1\text{ k}\Omega$, $T_A: -40\text{ }^\circ\text{C} \sim +105\text{ }^\circ\text{C}$, $V_{\text{ref}} = 1.0\text{V}/3\text{V}$		± 4	± 8	LSB
DNL ⁽⁵⁾	微分非线性			± 2	± 4	LSB
Gain error ⁽⁷⁾	增益误差			± 2	± 4	LSB
Vref precision ⁽⁸⁾	内部参考电压精度	$2.4\text{V} \leq \text{VDD} \leq 5.5\text{V}$, $-40\text{ }^\circ\text{C} \sim +105\text{ }^\circ\text{C}$	-2%		+2%	V
Vref temp drift	内部参考电压温漂			100		ppm

注释:

- 1) 在转换期间, 应考虑模拟 I_{DDA} 上的 $100\text{ }\mu\text{A}$ 及数字 I_{DD} 上 $60\text{ }\mu\text{A}$ 的额外消耗;
- 2) 由设计保证, 未经生产测试;
- 3) 满足该公式: $(R_{\text{ADC}} + R_{\text{AIN}}) * C_{\text{ADC}} * (\ln 2^{N+2}) < t_s / f_{\text{ADC}}$, N 表示 ADC 分辨率;
- 4) 设计参数, 不表示实际性能指标;
- 5) 消除失调和增益误差后, 衡量的线性指标;
- 6) 使用 sar_offex 功能, offset 会更小;
- 7) 当参考电压不准时, 也会引起增益误差;
- 8) trim 后的精度;

5.4 外部 24Mhz XTAL 振荡器

表 5-4

符号	参数	最小值	典型值	最大值	单位	测试条件
V_{HXT}	外部振荡器工作电压	2.2	-	5.5	V	-
T_A	温度	-40	-	105	$^\circ\text{C}$	-
I_{HXT}	工作电流	1.1		2	mA	24MHz, $V_{\text{DD}} = 3.3\text{V}$
		0.85	-	1.5	mA	24MHz, $V_{\text{DD}} = 2.2\text{V}$

F _{HXT}	输出时钟频率		24		MHz	
------------------	--------	--	----	--	-----	--

外部高速晶振的典型应用：

表 5-5

晶振引脚	晶振频率	C1	C2
OSC_IN, OSC_OUT	24Mhz	16~30pF	16~30pF

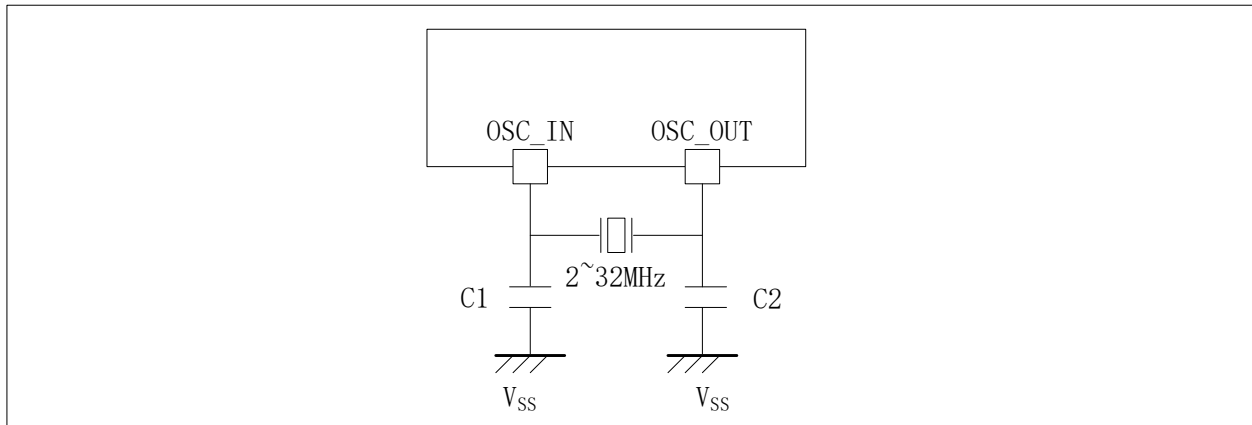


图 5-1 晶振外接图

注：当晶振频率高于 24MHz，ESR 电阻较差（>30 欧姆）的晶体不易起振。

5.5 内部 24MHz RC 振荡器

表 5-6

符号	参数	最小值	典型值	最大值	单位	测试条件
V _{HRC}	输出高电平电压	1.35	1.5	1.65	V	-
F _{HRC}	中心频率	-	24	-	MHz	T _A =25 ℃, V _{DD} =3.3V
	校准之后	-1	-	+1	%	T _A =25 ℃, V _{DD} =3.3V
		-2.5	-	+2.5	%	T _A =-40~105 ℃, V _{DD} =2.2~5.5V
I _{HRC}	工作电流	80	120	180	uA	T _A =25 ℃, V _{DD} =3.3V

5.6 内部 10KHz RC 振荡器

表 5-7

符号	参数	最小值	典型值	最大值	单位	测试条件
V _{LRC}	电压	2.2	-	5.5	V	-
F _{LRC}	中心频率	-	10	-	KHz	T _A =25 ℃, V _{DD} =3.3V
	校准之后	-3	-	+3	%	T _A =25 ℃, V _{DD} =3.3V
		-10	-	+10	%	T _A =-40~85 ℃, V _{DD} =2.28~5.5V
I _{LRC}	工作电流	-	2	-	uA	T _A =25 ℃, V _{DD} =3.3V

5.7 比较器

表 5-8

符号	参数	最小值	典型值	最大值	单位	测试条件
V _{CMP}	工作电压	2.2	5	5.5	V	-40 ℃ ~ +105 ℃
T	温度	-40	25	105	℃	-
V _{IN}	输入电压范围	0	-	A _{VDD}	V	-
I _{comp}	工作电流	45	75	100	uA	A _{VDD} =2.2V~5V
PSRR	电源电压抑制比	-	60	-	dB	-
CMRR	共模抑制比	-	60	-	dB	-
Comp Reponse	响应时间	-	1	-	uS	比较器一端接(V _{DD} -1.5)/2，另外一端从 VSS 跳变到 V _{DD} -1.5

CMP LSB	最小分辨率	-	1	-	mV	-
V _{offset}	失调电压	-2	-	2	mV	2.2V~5.5V, -40℃~+105℃

5.8 内部参考电压

表 5-9

符号	参数	最小值	典型值	最大值	单位	测试条件
V _{REF}	内部参考电压	-1%	1.0	+1%	V	AV _{DD} =3.3V, T _A =25℃
		-2%	1.0	-2%	V	AV _{DD} =2.2~5.5V, T _A =-40~105℃
		-1%	2.5	+1%	V	AV _{DD} =3.3V, T _A =25℃
		-2%	2.5	-2%	V	AV _{DD} =2.2~5.5V, T _A =-40~105℃
		-	AV _{DD}	-	V	
CMP0_VREF	比较器 0 参考电压	0.00	0.00	0.05	V	AV _{DD} >2.2V
		0.275	0.325	0.38	V	AV _{DD} >2.2V
		0.55	0.60	0.65	V	AV _{DD} >2.2V
		0.85	1.00	1.15	V	AV _{DD} >2.2V
		1.80	2.00	2.20	V	AV _{DD} >2.5V
		2.40	2.70	3.00	V	AV _{DD} >3.3V
		2.70	3.00	3.30	V	AV _{DD} >3.6V
		3.00	3.30	3.60	V	AV _{DD} >3.6V
		1.00	1.20	1.40	V	AV _{DD} >2.7V
		2.25	2.40	2.65	V	AV _{DD} >2.9V
CMP1_VREF	比较器 1 参考电压	0.15	0.20	0.25	V	AV _{DD} >2.2V
		0.35	0.40	0.45	V	AV _{DD} >2.2V
		0.66	0.66	0.72	V	AV _{DD} >2.2V
		0.73	0.80	0.87	V	AV _{DD} >2.2V
		1.13	1.23	1.33	V	AV _{DD} >2.2V
		1.45	1.58	1.71	V	AV _{DD} >2.2V
		1.98	2.18	2.38	V	AV _{DD} >2.7V
		2.40	2.60	2.80	V	AV _{DD} >3.1V

注：AV_{DD} 需比参考电压大 0.3V 才能保证参考电压的精度。

5.9 LDO 规格和 Power 管理

表 5-10

符号	参数	最小值	典型值	最大值	单位	测试条件
V _{DD}	输入电压	2.2	-	5.5	V	-
V _{LDO}	输出电压	1.35	1.5	1.65	V	-
T _A	工作温度	-40	25	105	℃	-
C _{LDO}	电容	-	4.7	-	uF	R _{ESR} <1Ω

注：为保证电源稳定，要做 LDO 和最近的 VSS 之间接一颗 4.7uF 电容。

5.10 FLASH 读写特性

FLASH 作为数据存储器件时，特性如下：

表 5-11

参数	测试条件	最小值	典型值	最大值	单位
写操作电压范围	-40℃~85℃	4.5		5.5	V
读操作电压范围	-40℃~85℃	2.35	5	5.5	V
写操作电流	5V, 25℃	-	9	-	mA
写操作次数	5V, 25℃	10000	-	-	次
写操作时间	5V, 25℃	1425	1500	1875	us
擦除操作时间	5V, 25℃	2375	2500	3125	us

5.11 可靠性指标

HBM ESD $\geq 4\text{KV}$

EFT $\geq 4.8\text{KV}$

Latchup $\geq 400\text{mA}$

芯海科技CHIPSEA

6 封装图

6.1 QFN-28 PIN

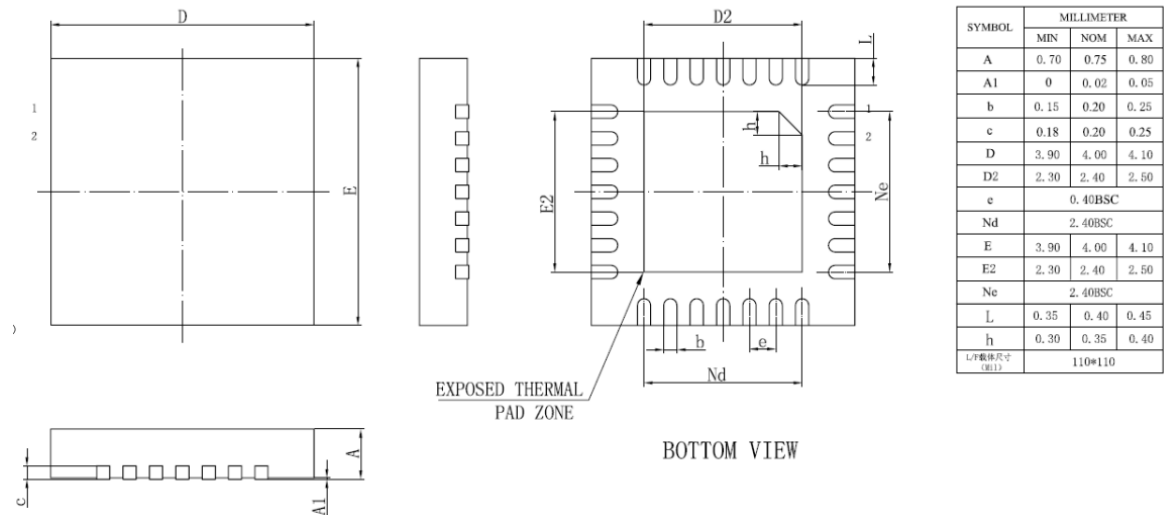


图 6-1 QFN28 封装外形图

表 6-1 QFN28 封装尺寸表

外形尺寸 Package size(mm):	QFN28
长* 宽*高(mm):	0404X0.75
外管脚间距 Lead/Pin Pitch(mm):	e=0.40
外管脚数量 Lead/Pin Count:	28

6.2 QFN-24 PIN

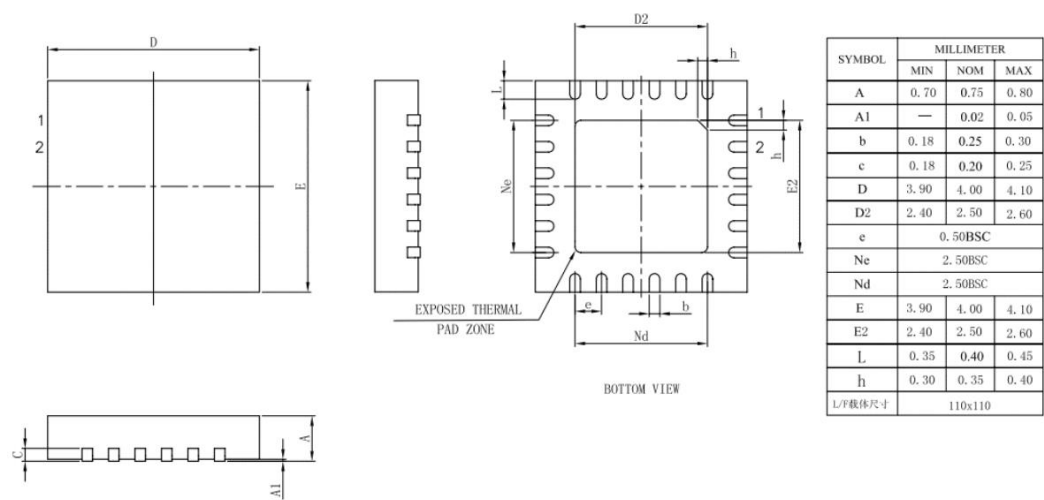


图 6-2 QFN24 封装外形图

表 6-2 QFN24 封装尺寸表

外形尺寸 Package size(mm):	QFN24L(0404X0.75-0.50)
长* 宽*高(mm):	0404X0.75
外管脚间距 Lead/Pin Pitch(mm):	0.5
外管脚数量 Lead/Pin Count:	24

7 单片机产品命名规则

7.1 产品型号说明

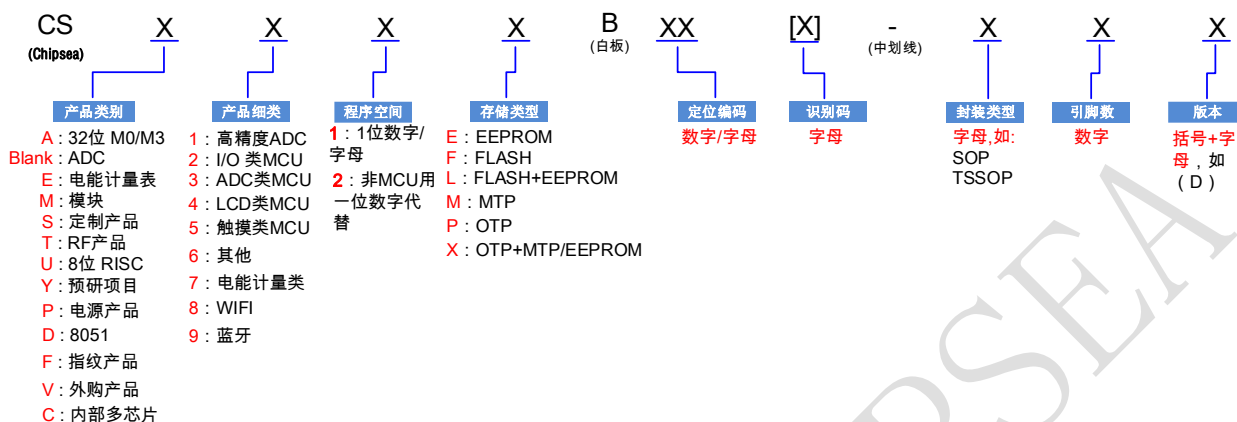


图 7-1

表 7-1 封装缩写表

标示符	封装类型
BD	Bonding
DI	DIP
SD	SDIP
SO	SOP
SS	SSOP
TS	TSSOP
QF	QFP
LQ	LQFP
TQ	TQFP
QN	QFN
MS	MSOP

7.2 命名举例说明

表 7-2

名称	内核	ROM 类型	功能分类	产品定位型号	芯片版本	封装形式	工作温度范围	封装材料
CSU32M10-SOP16	8 位 Risc MCU	MT P	ADC	20	第 1 版	SOP	-40~85 ℃	无铅封装(PB-Free 封装)

7.3 产品印字说明

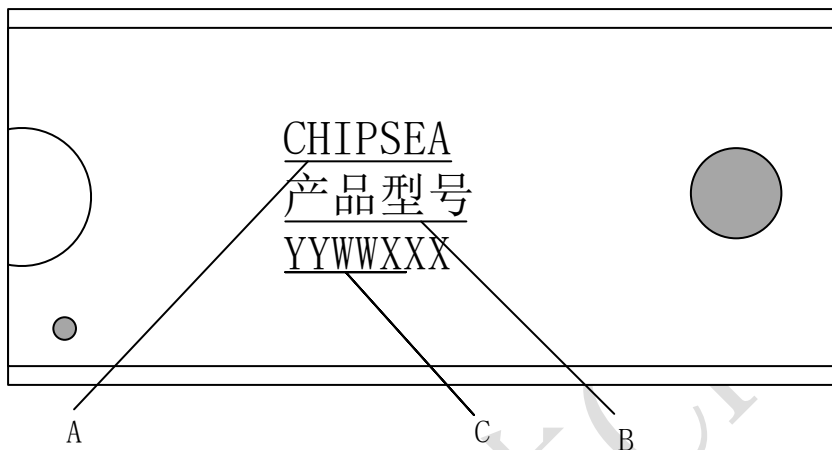


图 7-2

芯片正面印字一般有 3 行：

第一行为公司名称，为 **CHIPSEA**。

第二行为产品型号。对于一些小尺寸封装，会对产品型号进行缩减。

第三行为日期码。从左端起算，前两位为公历年号后两位；第三第四位为本年度日历周数，不足两位时左端补 0；最后三位为产品随机号。

例如，CSU32M10 的印字如下：

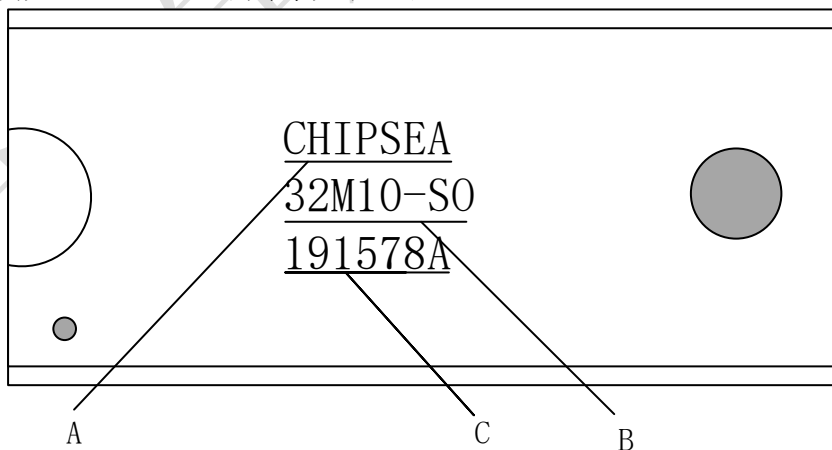


图 7-3