















UCSR1C - UART1 控制和状态寄存器 C.....	109
UBRR1L - UART1 波特率寄存器低字节.....	110
UBRR1H - UART1 波特率寄存器高字节.....	110
UDR1 - UART1 数据寄存器.....	110
16. I2C 串行通信控制器.....	111
16.1. 综述.....	111
16.2. 工作模式.....	111
16.3. 操作模式.....	112
16.4. 主机发送模式.....	112
16.5. 主机接收模式.....	112
16.6. 从机发送模式.....	113
16.7. 从机接收模式.....	113
16.8. 寄存器定义.....	114
TWCRC - TWI 控制寄存器.....	114
TWSR - TWI 状态寄存器.....	114
TWR - TWI 数据寄存器.....	115
17. 同步串行外设接口 (SPI).....	115
17.1. 综述.....	115
17.2. 发送和接收.....	116
17.3. 数据模式.....	116
17.4. SPI 电路结构图.....	117
17.5. SPSS 引脚功能.....	118
17.6. SPI 初始化.....	118
17.7. SPI 主机初始化.....	119
17.8. SPI 从机初始化.....	119
17.9. SPI 中断.....	119
17.10. SPCK 和系统时钟的关系.....	119
17.11. SPI 寄存器列表.....	120
SPDR-SPI 数据寄存器.....	120
SPCR-SPI 控制寄存器.....	120
SPSR-SPI 状态寄存器.....	121
SPFR - SPI 缓冲寄存器.....	122
18. USB Type-C PD 收发器 (uTPCO/1).....	123
18.1. 概述.....	124
18.2. 收发器前端.....	124
18.3. 物理层控制器.....	125
18.4. PD 数据发送流程.....	125
18.5. PD 数据接收流程.....	125
18.6. 中断请求.....	125
18.7. 寄存器定义.....	125
TPCSR0 -TPC 状态寄存器 0.....	126
TPC1SR0 -TPC1 状态寄存器 0.....	127
TPCBDR - PD 发送波特率寄存器.....	127
KCMSK - KCode 屏蔽寄存器.....	127
TPCSIG -PD 设备模式标识寄存器.....	128
TPCON0 - TPC 控制寄存器 0.....	128
TPCON1 - TPC 控制寄存器 1.....	129
TPCON2 - TPC 控制寄存器 2.....	129



TPCTXFF - uTPC 发送 FIFO 访问寄存器 .....	130
TPCRXFF - uTPC 接收 FIFO 访问寄存器 .....	130
TPCIER -TPC 外设中断使能寄存器 .....	130
TPCIF -TPC 外设中断标志寄存器 .....	131
CCACR0 -Type-C CC 控制寄存器 0 .....	131
CCACR1 -Type-C CC 控制寄存器 1 .....	131
CCACR2 -Type-C CC 控制寄存器 2 .....	132
CCACR3 -Type-C CC 控制寄存器 3 .....	132
CCACR4 -Type-C CC 控制寄存器 4 .....	132
TPC1BDR - PD 发送波特率寄存器 .....	133
KCMSK1 -KCode 屏蔽寄存器 1 .....	133
TPC1SIG -PD 设备模式标识寄存器 .....	134
TP1CON0 - TPC1 控制寄存器 0 .....	134
TP1CON1 - TPC1 控制寄存器 1 .....	135
TP1CON2 - TPC 控制寄存器 2 .....	135
TPC1TXFF - uTPC1 发送 FIFO 访问寄存器 .....	136
TPC1RXFF - uTPC1 接收 FIFO 访问寄存器 .....	136
CC1ACR0 -Type-C1 CC 控制寄存器 0 .....	136
CC1ACR1 -Type-C1 CC 控制寄存器 1 .....	137
CC1ACR2 -Type-C1 CC 控制寄存器 2 .....	137
CC1ACR3 -Type-C1 CC 控制寄存器 3 .....	138
CC1ACR4 -Type-C1 CC 控制寄存器 4 .....	138
TPC1IER -TPC1 外设中断使能寄存器 .....	138
19. USB A 口充电模拟前端 .....	139
19.1. 综述 .....	139
TYPEAB0 电路结构图 .....	139
TYPEAB1 电路结构图 .....	139
19.2. 寄存器定义 .....	140
TPACR0 - USB A 口 DPO 通道控制寄存器 .....	140
TPACR1 - USB A 口 DMO 通道控制寄存器 .....	140
TPACR2 - USB A 口 DP1 通道控制寄存器 .....	141
DPDMTR - USB A 口延时控制寄存器 .....	141
TPACR3 USB A 口 DM1 通道控制寄存器 .....	141
TPASR0 - USB A 0 口 寄存器 .....	142
TPASR1 - USB A 1 口 寄存器 .....	142
20. 华为 HT 快充通信接口 .....	142
20.1. 综述 .....	142
20.2. 发送器 .....	143
20.3. 接收器 .....	144
20.4. 中断控制 .....	144
20.5. 寄存器定义 .....	144
HTCR - HT 传输控制寄存器 .....	144
HTSR - HT 传输状态寄存器 .....	145
HTRXD - HT 传输接收字节数据寄存器 .....	146
HTTXD - HT 传输发送字节数据寄存器 .....	146
HTIFR - HTIFR 中断寄存器 .....	146
21. 模拟比较器 0 (ACO) .....	147
21.1. 综述 .....	147

21.2.	模拟比较器的输入 .....	147
21.3.	比较器输出与 PWM 控制 .....	148
21.4.	寄存器定义 .....	148
	ACOSR - ACO 控制和状态寄存器 .....	148
	ACOXR - ACO 辅助控制器寄存器 .....	149
	DIDR2 - 数字输入禁止控制寄存器 2 .....	149
	DIDR3 - 数字输入禁止控制寄存器 3 .....	150
22.	模拟比较器 1 (AC1) .....	150
22.1.	综述 .....	150
22.2.	模拟比较器的输入 .....	151
22.3.	模拟比较器的输出 .....	151
22.4.	寄存器定义 .....	151
	AC1SR - AC1 控制和状态寄存器 .....	151
	AC1XR - AC1 辅助控制寄存器 .....	152
23.	数模转换器 (DAC0/1) .....	153
23.1.	综述 .....	153
23.2.	寄存器定义 .....	153
	DA0CON - DAC0 控制寄存器 .....	153
	DAOLR - DAC0 输出电压控制寄存器低位 .....	154
	DAOHR - DAC0 输出电压控制寄存器高位 .....	154
	DA1CON - DAC1 控制寄存器 .....	154
	DA1LR - DAC1 输出电压控制寄存器低位 .....	155
	DA1HR - DAC1 输出电压控制寄存器高位 .....	155
24.	模数转换器 (ADC) .....	155
24.1.	综述 .....	156
24.2.	ADC 的操作 .....	156
24.3.	预分频及 ADC 转换时序 .....	157
24.4.	采样通道与参考电压 .....	157
24.5.	多路输入分压电路 (VDS) .....	157
24.6.	ADC 失调校准 .....	157
24.7.	寄存器定义 .....	158
	ADCL - ADC 数据低字节寄存器 .....	158
	ADCH - ADC 数据高字节寄存器 .....	159
	ADCSRA - ADC 控制和状态寄存器 A .....	159
	ADCSRB - ADC 控制和状态寄存器 B .....	160
	ADMUX - ADC 多路选择控制寄存器 .....	160
	ADCSRC - ADC 控制和状态寄存器 C .....	161
	ADCSRD - ADC 控制和状态寄存器 D .....	162
	OFR0 - 失调补偿寄存器 0 .....	162
	OFR1 - 失调补偿寄存器 1 .....	162
25.	运算放大器 (DAP) .....	163
25.1.	寄存器定义 .....	163
	VCAL1 - 内部 1.28V 基准电压校准寄存器 .....	163
	VCAL2 - 内部 2.56V 基准电压校准寄存器 .....	163
	DAPOCR0 - DAP0 控制寄存器 0 .....	163
	DAP1CR0 - DAP1 控制寄存器 0 .....	164
	DAPOCR1 - DAP0 控制寄存器 1 .....	164
	DAP1CR1 - DAP1 控制寄存器 1 .....	165

26. 其他寄存器定义 .....	165
DIDR0 - 数字输入禁用控制寄存器 0 .....	165
DIDR1 - 数字输入禁用控制寄存器 1 .....	165
26.1. 输入失调寄存器 .....	166
DAP1TR - DAP1 输入失调校准寄存器 .....	166
DAP1TC - DAP1 输入失调校准控制寄存器 .....	166
DAP0TR - DAP0 输入失调校准寄存器 .....	166
DAP0TC - DAP0 输入失调校准控制寄存器 .....	166
26.2. 校准寄存器 .....	167
RCMTR - 24MHz HFRC 振荡器校准寄存器 .....	167
RCKTR - 32KHz RC 振荡器校准寄存器 .....	167
VCAL1 - 内部 1.28V 基准电压校准寄存器 .....	167
VCAL2 - 内部 2.56V 基准电压校准寄存器 .....	167
AC0TR - AC0 输入失调校准寄存器 .....	168
AC1TR - AC1 输入失调校准寄存器 .....	168
CCACR0 - TYPEC 控制寄存器 0 .....	168
CCTR1 - TYPEC 校准寄存器 1 .....	169
CCTR2 - TYPEC 校准寄存器 2 .....	169
CCTR3 - TYPEC 校准寄存器 3 .....	169
CCTR4 - TYPEC 校准寄存器 4 .....	169
GUID0 寄存器 .....	170
GUID1 寄存器 .....	170
GUID2 寄存器 .....	170
GUID3 寄存器 .....	170
27. 封装参数 .....	171

# 1. 产品概述

## 1.1. 概述

PDS8F208A 是一款内部自带双路 USB type-C (可做 PD 快充协议) 口 & 运放 & ADC & DAC & 高速 PWM 等外设的 8 位通用 RISC 单片机。

## 1.2. 特性

### 高级 RISC 指令构架

- 131 条指令, 80 以上为单周期执行
- 32x8 通用工作寄存器
- 24MHz 工作时最高可达 24MIPS 的执行效率
- 20Kbytes 片上可在线编程序存储器 (MTP)
- 2Kbytes 内部数据 SRAM
- 28 通用 I/O 口
- 内核直接访问程序与数据存储空间

### USB Type-C/USB PD/华为通讯控制器

- 2 组 USB A 口电源控制模拟前端 (DP/DM 偏置)
- 2 组 USB C 口 PD 控制器+PDPHY (PD3.0+)
- USB A 口华为通讯控制器
- 支持 PD2.0/3.0, QC2.0/3.0, FCP, SCP, AFC 等协议

### 串口通讯组

- 2 组 UART 异步串行通信接口
- I2C 主从通信接口
- SPI 主从通信接口

### 外设控制器

- 1 个带独立预分频器的 8 位定时器, 产生时间基准
- 1 个带独立预分频器的 16 位定时器, 支持输入俘获和 PWM 输出
- 1 个带后分频器的 12 位定时器, 支持输入捕获和 PWM 输出

- 3 组互补输出, 支持可编程死区控制
- 独立 6 路 PWM 输出
- 自动关闭控制/自动恢复控制
- 10 通道 12 位高速模数转换器 (ADC)
- 可选内部、外部参考电压
- 内部电源电压监控通道
- 2 组可编程增益 (8/16/32/128) 差分放大输入通道
- 2 路模拟比较器 (AC), 支持来自 ADC 输入通道的扩展
- 内部 1.28V/2.56V±1% 可校准参考电压源
- 2 路 10 位可编程 DAC, 支持 3mA 输出缓冲
- 可编程看门狗定时器 (WDT)
- 16 位乘法/移位加速器

### 特殊处理器功能

- SWD 双线片上调试/量产接口
- 外部中断源与 I/O 电平变化中断支持
- 内置上电复位电路 (POR) 与可编程低电压检测电路 (LVD)
- 高速定时器模式, 可通过 RC 倍频支持 48MHz 计数频率
- 内置 1 可校准 24MHz RC 振荡器, 支持倍频输出
- 内置 1 可校准 32KHz RC 振荡器
- 4x80mA 大电流推挽驱动 I/O, 支持高速 PWM 应用
- 25mA 普通 I/O 驱动支持

## 1.3. 工作环境

- 工作电压: 2.5V ~ 5.5 V 工作温度: -40C~+85C
- 最低功耗: 1uA @3.3V

- USB A/C 口电源
- 双 Type-C 口协议控制器
- PD 无线充
- DC/DC 环路控制

## 1.4. 应用场景

## 1.5. 封装形式

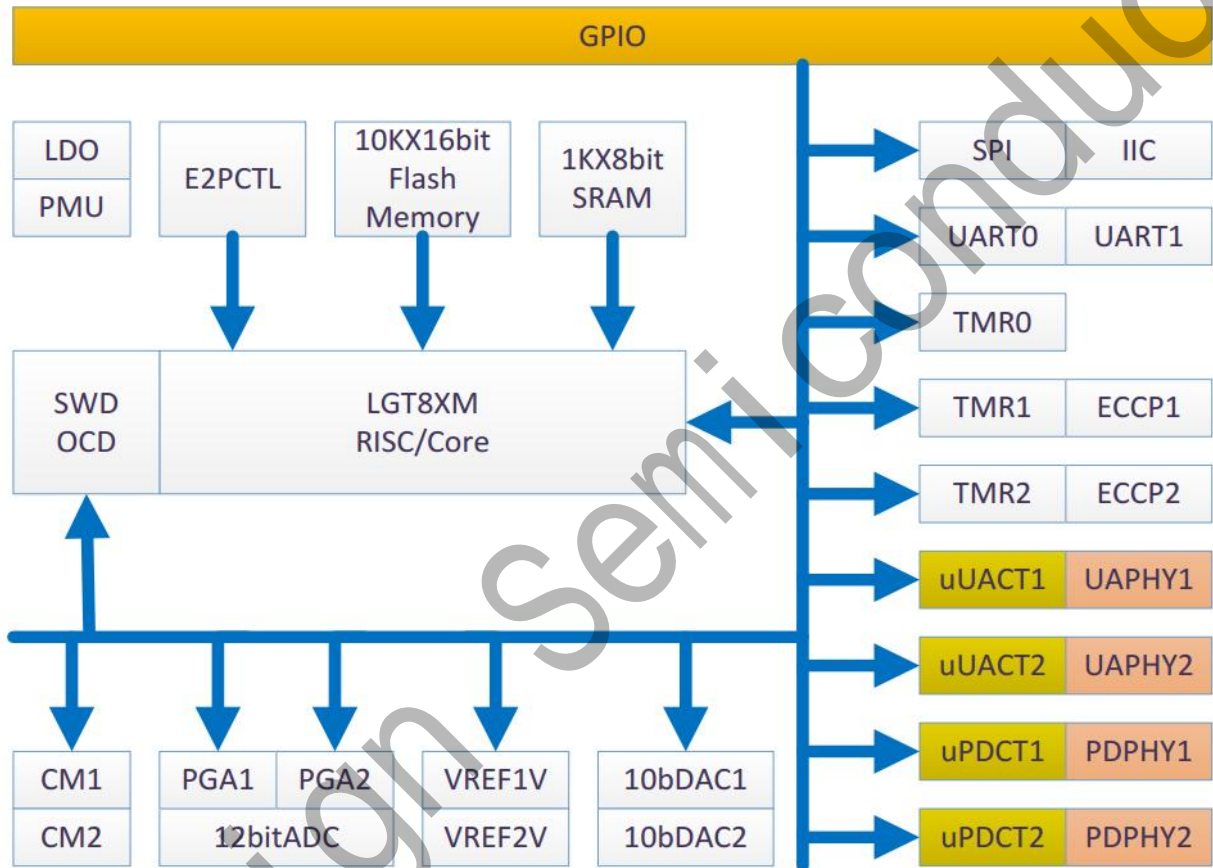
- QFN32
- SSOP24 (暂未量产)

## 1.6. 版本历史

版本号	修改纲要	修改时间
Ver1.0.0	首版	2023/03/31
Ver1.0.1	规范文档格式, 部分细节修正	2023/04/04
Ver1.0.2	部分细节修正	2023/04/11

Ver1.1.0	QFN32 的脚位定义微调	2023/04/17
Ver1.1.1	细节更新	2023/04/26
Ver1.1.2	细节更新	2023/04/28
Ver1.1.5	细节更新	2023/06/12
Ver1.2.0	加入 EEPROM 操作例程	2023/06/20
Ver1.2.1	WDT 细节更新	2023/06/25
Ver1.2.2	细节更新	2023/07/26
Ver1.2.3	细节更新	2024/01/12

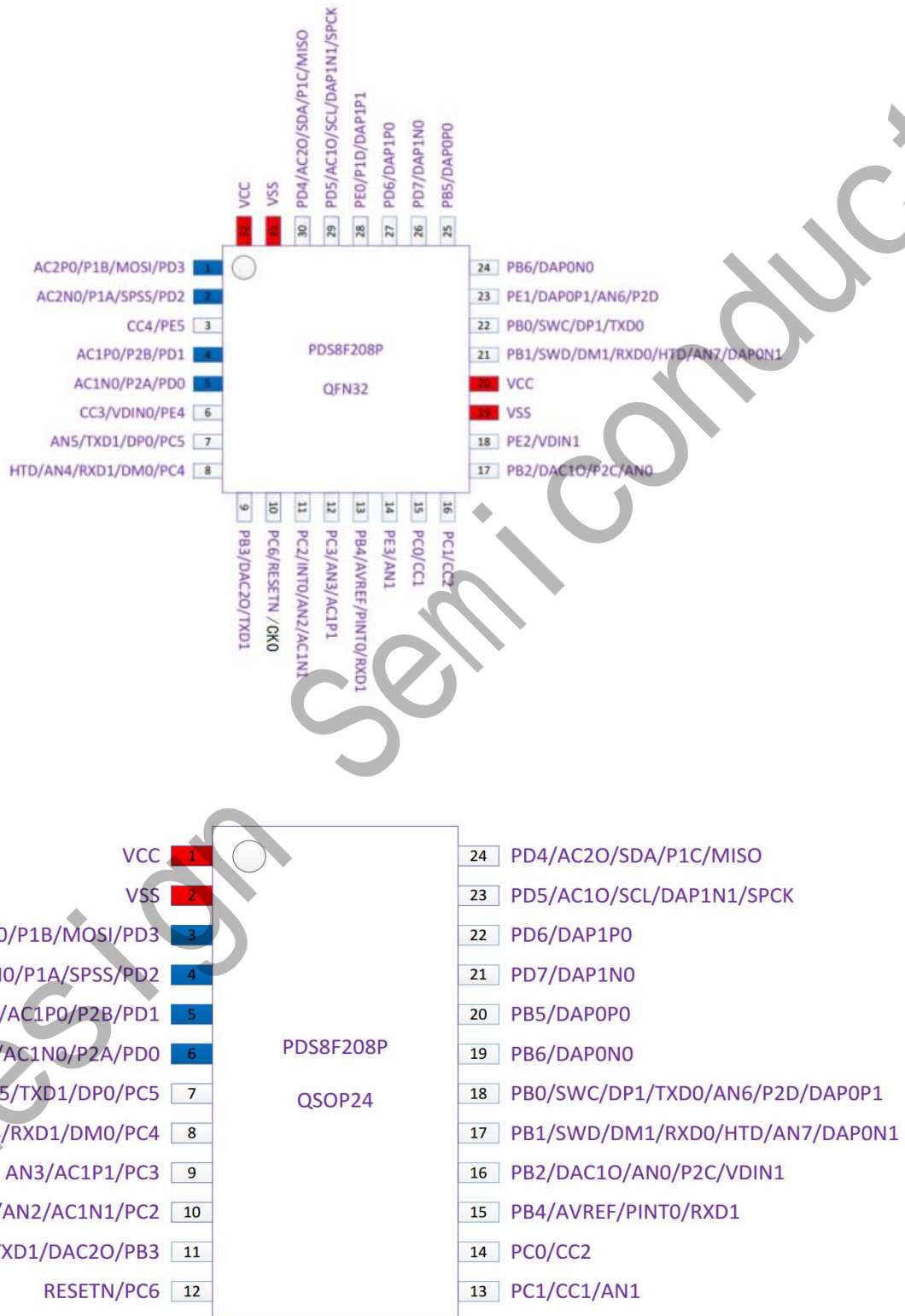
## 1.7. 系统框架



模块名称	模块功能
LGT8XM	LGT8XM 内核
SWD/OCD	SWD 片上调试器
PMU	功耗管理单元
I2C	I2C 主从收发器
SPI	SPI 主从收发器
UART0/1	异步串行收发器 0/1
TMR0/1/2	定时/计数器0/1/2
ECCP1/2	增强输入俘获/PWM 以及互补波形发生器
CM1/2	模拟比较器1/2
12bit ADC	12 位多通道模数转换器
PGA1/2	可编程增益运算放大器
DAC10/DAC8	10bit/8bit 数模转换器

VREF1V/2V	内部 1.28V/2.56V 参考电压源
10bDAC1/2	10 位数模转换器 1/2
Uuact12/UAPHY2	USB A 口电源管理协议控制器/物理前端
uPDCT12/PDPHY2	USB C 口PD 控制器/PD 物理层

## 1.8. 引脚定义



1. 红色引脚为电源引脚；
2. 蓝色引脚为大电流引脚，驱动能力为 80mA；

### 1.9. 引脚说明

引脚	名称	复用功能描述
1	PD3	PD3 - 通用GPIO端口 MOS1 - SPI 主出从入 P1B - ECP1 输出 B AC2P0 - 比较器 2 正输入 0
2	PD2	PD2 - 通用GPIO 端口 SPSS - SPI 从机选择 P1A - ECP1 输出 A AC2N0 - 比较器 2 负输入 0
3	PE5	PE5 - 通用GPIO 端口 CC3 - USB Type C 接口CC通信端口3
4	PD1	PD1 - 通用GPIO 端口 P2B - ECP2 输出 B AC1P0 - 比较器 1 正输入 0
5	PD0	PD0 - 通用GPIO 端口 P2A - ECP2 输出 A CC4 - USB Type C接口CC通信端口4
6	PE4	PE4 - 通用GPIO 端口 CC4 - USB Type C 接口CC通信端口4 VDIN0 - VDO 输入通道0
7	PC5	PC5 - 可编程 GPIO 端口 DP0 - USB A 口 DP0 端口 AN5 - ADC 外部模拟输入通道 5 TXD1 - UART1 数据发送端口
8	PC4	PC4 - 可编程 GPIO 端口 DM0 - USB A 口 DM0 端口 AN4 - ADC 外部模拟输入通道 4 RXD1 - UART1 数据接收端口 HTD - 华为快充通信数据端口
9	PB3	PB3 - 通用GPIO端口 TXD1 - UART1 数据发送端口 DAC20 - 内部 DAC2 的输出
10	PC6	PC6 - 可编程GPIO 端口 RESETN - 外部复位输入
11	PC2	PC2 - 通用 GPIO 端口 AN2 - ADC 外部模拟输入通道 2 INT0 - 外部中断输入 0 AC1N1 - 模拟比较器 1 外部负端输入 CKO-输出系统时钟
12	PC3	PC3 - 可编程GPIO 端口 AN3 - ADC 外部模拟输入通道 3 AC1P1 - 模拟比较器1 外部正端输入

13	PB4	PB4 - 通用GPIO端口 RXD1 - UART1数据接收端口 AVREF-外部参考 PINT0-
14	PE3	PE3 - 通用GPIO 端口 AN1 - ADC 外部模拟输入通道1
15	PC1	PC1 - 可编程GPIO 端口 CC1 - USB Type C 接口 CC 通信端口 1
16	PC0	PC0 - 通用GPIO 端口 CC2 - USB Type C 接口CC通信端口2
17	PB2	PB2 - 通用GPIO 端口 P2C - ECP2 输出 C DAC10 - 内部DAC1 的输出 ANO - ADC 外部模拟输入通道0
18	PE2	PE2 - 通用GPIO 端口 VDIN1 - VDO 输入通道1
19	VSS	电源-
20	VCC	电源+
21	PB1	PB1 - 通用GPIO 端口 SWD - 调试接口的数据通信端口 RXD0 - UART0数据接收端口 DM1 - USB A 口DM1 端口 AN7 - ADC 外部模拟输入通道7 HTD - 华为快充通信数据端口 DAPON1-DAPO 负端口 1
22	PB0	PB0 - 通用GPIO 端口 SWC - 调试接口的时钟信号端口 TXD0 - UART0 数据发送端口 DP1 - USB A 口DP1 端口
23	PE1	PE1 - 通用GPIO 端口 DAPOP1-DAPO 正端口 1 AN6 - ADC 外部模拟输入通道6 P2D - ECP2 输出 D
24	PB6	PB6 - 通用GPIO端口 DAPON0 - DAPO 负端口 0
25	PB5	PB5 - 通用GPIO端口 DAPOP0 - DAPO 正向输入端口 0
26	PD7	PD7 - 通用GPIO端口 DAP1N0 - DAP1 负向输入端口 0
27	PD6	PD6 - 通用GPIO端口 DAP1P0 - DAP1 正向输入端口 0
28	PE0	PE0 - 通用GPIO 端口 DAP1P1- DAP1 正向输入端口 1

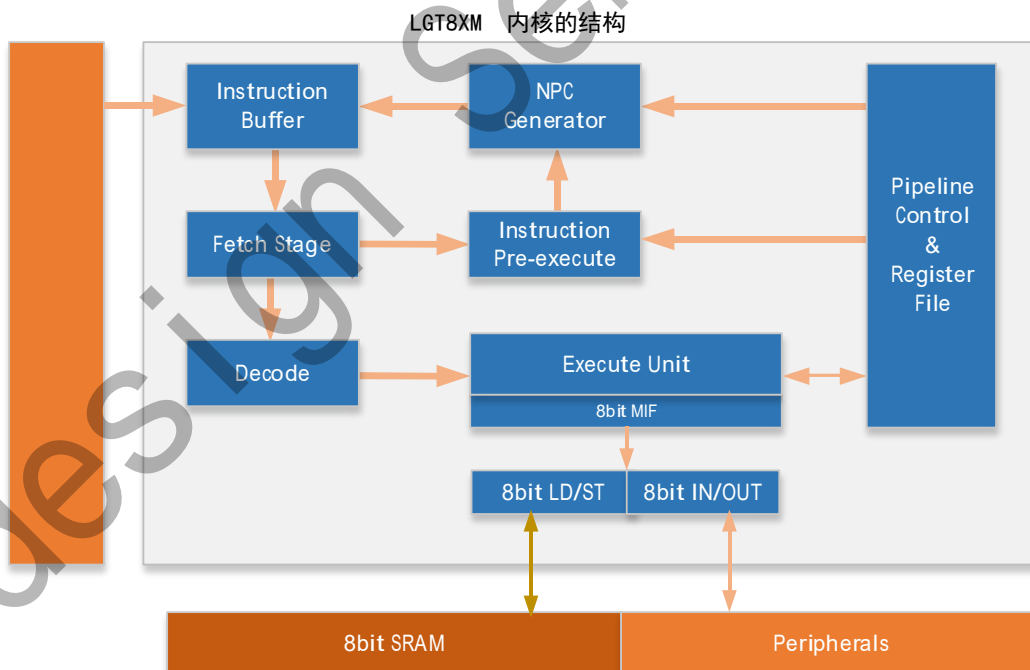
29	PD5	PD5 - 通用GPIO端口 DAP1N1 - DAP1 负向输入端口 1 AC10 - 比较器 1 输出 SCL - TWI 时钟 SPCK - SPI 时钟
30	PD4	PD4 - 通用GPIO端口 AC20 - 比较 2 输出 SDA - TWI 数据 MISO - SPI 主入从出 P1C - ECP1 输出 C
31	VCC	电源+
32	VSS	电源-

## 2. LGT8XM 内核

- 低功耗设计
- 高效率 RISC 构架
- 130 条指令，其中 80%以上为单周期
- 内嵌在线调试(OCD)支持

### 2.1. 概述

本章节主要描述LGT8XM 内核构架和功能。内核是MCU 的大脑，负责保证程序的正确 执行，因此内核必须能够准确的执行计算，控制外设以及处理各种中断。



为了实现更大的效率和并行性，LGT8XM 内核采用哈佛构架 - 独立的数据和程序总线。指令通过一个优化的两级流水线执行，两级流水线能够减少流水线中无效指令的个数，减少了对 FLASH 程序存储器的访问量，因此可以降低内核运行的功耗。同时 LGT8XM 内核在取指令的前级中增加了指令缓存（可以同时缓存 2 条指令），通过在取指令周期的预执行模块，进一步减少了对 FLASH 程序存储器的访问频率；经大量测试，LGT8XM 可以比其他同类构架的内核减少约 50%对 FLASH 的访问，大大降低了系统的运行功耗。

LGT8XM 内核具有 32 个 8 位高速访问的通用工作寄存器(Register file)，有助于实现单周期的算术逻辑运算(ALU)。一般情况下，ALU 运算的两个操作数均来自与通用工作寄存器，ALU 运算的结果也会在一个周期内写入到寄存器文件中。

32 个通过工作寄存器中的 6 个用于两两结合构成三个 16 位寄存器，可用于间接寻址地址指针，用于访问外部存储空间以及 FLASH 程序空间。LGT8XM 支持单周期的 16 位算术运算，极大的提高了间接寻址的效率。LGT8XM 内核中这三个特殊的 16 位寄存器被命名为 X, Y, Z 寄存器，将在后面详细介绍。

ALU 支持寄存器之间以及常数与寄存器之间的算术逻辑运算，单个寄存器的运算也可以在 ALU 中执行。ALU 运算完成后，运算结果对内核状态的影响更新到状态寄存器中(SREG)。

程序流程控制通过条件和无条件跳转/调用实现，可以寻址到所有的程序区域。大部分

LGT8XM 指令为 16 位。每个程序地址空间对应一个 16 位或者 32 位的 LGT8XM 指令。

内核响应中断或子程序调用后，返回地址(PC)被存储在堆栈中。堆栈被分配在系统的一般数据 SRAM 中，因此堆栈的大小仅受限于系统中 SRAM 的大小和用法。所有的支持中断或子程序调用的应用，必须首先初始化堆栈指针寄存器(SP)，SP 可以通过 I/O 空间访问。数据 SRAM 可以通过 5 种不同的寻址模式访问。LGT8XM 的内部存储空间都被线性的映射到一个统一的地址空间。具体请参考存储章节的介绍。

LGT8XM 内核包含了一个灵活的中断控制器，中断功能可以通过状态寄存器中的一个全局中断使能位控制。所有的中断都有一个独立的中断向量。中断的优先级与中断向量地址有对应关系，中断地址越小，中断的优先级就越高。

I/O 空间包含了 64 个可以通过 IN/OUT 指令直接寻址的寄存器空间。这些寄存器现实对内核控制以及状态寄存器，SPI 以及其他 I/O 外设的控制功能。这部分空间可以通过 IN/OUT 指令直接访问，也可以通过他们映射到数据存储空间的地址访问(0x20 - 0x5F)。另外，PDS8F208A 也包含扩展的 I/O 空间，他们被映射到数据存储空间 0x60 - 0xFF，这里只能使用 ST/STS/STD 以及 LD/LDS/LDD 指令访问。

## 2.2. 算术逻辑运算单元 (ALU)

LGT8XM 内部包含了一个 16 位的算术逻辑运算单元，能够在 一个周期内完成 16 为数据的算术运算。高效的 ALU 与 32 个通用工作寄存器相连。能够在 一个周期内完成两个寄存器 或者寄存器与立即数之间的算术逻辑运算。ALU 的运算分为三种：算术，逻辑以及位运算。同时 ALU 部分也包含了一个单周期的硬件乘法器，能够在 一个周期内实现两个 8 位寄存器直接的有符号或者无符号运算。请参考指令集部分的详细介绍。

## 2.3. 状态寄存器 (SREG)

状态寄存器中主要保存了因执行最近一次 ALU 运算而产生的结果信息。这些信息用于 控制程序执行流程。状态寄存器是在 ALU 操作完全结束后更新，这样就可以省去了使用单独 的比较指令，可以带来更加紧凑高效的代码实现。状态寄存器的值在响应中断和从中断中退 出时并不会自动保存和恢复，这需要软件去实现。

## 2.4. SREG 寄存器定义

SREG 系统状态寄存器								
地址: 0x3F (0x5F)				默认值: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	I	T	H	S	V	N	Z	C
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位定义								
0	C	进位标志，表示算术或逻辑操作导致了进位，具体请参考指令描述						
1	Z	零标志，表示算术或逻辑运算的结果为零，请参考指令描述部分						
2	N	负标志，表示算术或逻辑运算产生了一个负数，请参考指令描述部分						
3	V	溢出标志，表示二进制补码运算结果产生溢出，请参考指令描述部分						

4	S	符号位，等效于N 与V 的异或运算结果，具体请参考指令描述 部分
5	H	半进位标志，在 BCD 运算中 useful，表示字节运算产生的半进位
6	T	临时位，位复制(BLD)和位存储(BST)指令中使用，T 位将作为一个临时的存储位，用于临时存放通用寄存器中的某一位的值。具体请参考指令描述部分
7	I	全局中断使能位，必须设置此位为1 才能使能内核响应中断事 件。不同的中断源是由独立的控制位控制。全局中断使能位是控制中断信号进入内核的最后一道屏障。I 位在内核响应中断向量后由硬件自动清除，在执行中断返回指令(RETI)后自动置位。 I 位也可以使用SEI 和CLI 指令改变，请参考指令描述部分

## 2.5. 通用工作寄存器

通用工作寄存器根据 LGT8XM 指令集构架优化。为了达到内核执行需要的效率和灵活性，

LGT8XM 内部的通用工作寄存器支持以下几种访问模式：

- 一个 8 位的读同时一个 8 位的写操作
- 两个 8 位的读同时一个 8 位的写操作
- 两个 8 位的读同时一个 16 位的写操作
- 一个 16 位的读同时一个 16 位的写操作

LGT8XM 通用工作寄存器

寄存器单元	地址	描述
R0	0X00	
R1	0X01	
R2	0X02	
...	...	
R13	0X0D	
R14	0X0E	
R15	0X0F	
R16	0X10	
R17	0X11	
...	...	
R26	0X1A	X 寄存器低字节
R27	0X1B	X 寄存器高字节
R28	0X1C	Y 寄存器低字节
R29	0X1D	Y 寄存器高字节
R30	0X1E	Z 寄存器低字节
R31	0X1F	Z 寄存器高字节

大部分指令能够直接访问到全部的通用工作寄存器，他们大部分也都是单周期指令。如上图所示，每一个寄存器都对应一个数据存储空间的地址，这些通用工作寄存器被映射到数据存储空间。尽管他们没有真正的存在于SRAM中，但这种统一映射的存储组织给访问 他们带来了很大的灵活性。X/Y/Z 寄存器可以作为指针索引到任何通用寄存器。

## 2.6. X/Y/Z 寄存器

寄存器R26...R31 可以两两组合，构成三个16 位寄存器。这三个16 位寄存器主要用于间接 寻址访问的地址指针，X/Y/Z 寄存器结构如下：

	15	XH		XL	0
X 寄存器	7		0	7	0

	R27 (0x1B)			R26 (0x1A)		
	15	YH			YL	0
Y 寄存器	7		0	7		0
	R29 (0x1D)			R28 (0x1C)		
	15	ZH			ZL	0
Z 寄存器	7		0	7		0
	R31 (0x1F)			R30 (0x1E)		

在不同的寻址模式下，这些寄存器被用作固定偏移，自动递增以及自动递减的地址指针，具体细节请参考指令描述部分。

## 2.7. 堆栈指针

堆栈用于存储临时数据，局部变量以及中断和子程序调用的返回地址。需要特别注意的是，堆栈别设计为从高地址向低地址生长。堆栈指针寄存器 (SP) 总是指向堆栈的顶部。堆栈指针指向数据 SRAM 所在的物理空间，这里存放子程序或中断调用必须的堆栈空间。PUSH 指令将会使得堆栈指针递减。

堆栈在 SRAM 中的位置必须在子程序执行或者中断使能之前由软件正确的设置。一般情况下是将堆栈指针初始化指向 SRAM 的最高地址处。堆栈指针必须设置为高位 SRAM 开始地址。SRAM 在系统数据存储映射的地址请参考系统数据存储部分。

堆栈指针相关的指令

指令	堆栈指针	描述
PUSH	增加1	数据压入堆栈
CALL ICALL RCALL	增加2	中断或者子程序调用的返回地址压入堆栈
POP	减少1	数据从堆栈取出
RET RETI	减少2	中断或者子程序调用的返回地址从堆栈中取出

堆栈指针由分配在 I/O 空间的两个 8 位的寄存器构成。堆栈指针的实际长度与系统实现相关。在 LGT8XM 构架的有些芯片实现中，数据空间非常小，以至于仅仅 SPL 就能满足寻址需要，这种情况下，SPH 寄存器将不会出现。

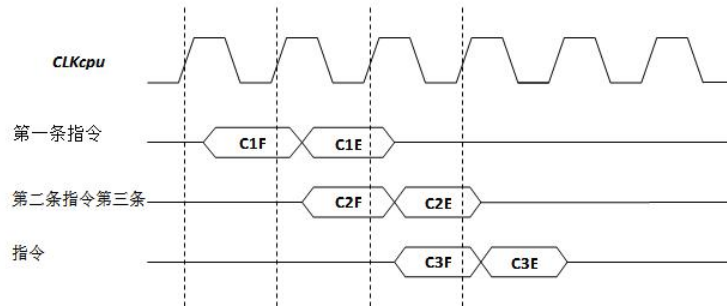
SPH/SPL 堆栈指针寄存器定义

SPH/SPL 堆栈指针寄存器		
SPH: 0x3E (0x5E)	默认值: RAMEND	
SPL: 0x3D (0x5D)		
SP	SP[15:0]	
R/W	R/W	
位定义		
[7:0]	SPL	堆栈指针低8位
[15:8]	SPH	堆栈指针高8位

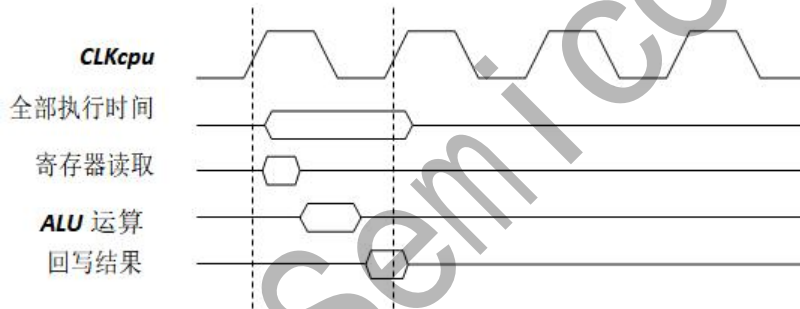
## 2.8. 指令执行时序

这一章节描述指令执行的一般时序概念。LGT8XM 内核由内核时钟 (CLKcpu) 驱动, 这个时钟直接来自与系统的时钟源选择电路。

下图展示了哈弗构架与快速访问寄存器文件概念基础上的指令流水线执行时序。这是使得内核能够获得 1MIPS/MHz 的执行效率的物理保证。从上图可以看出, 第一条指令的执行期间同时会读出第二条指令。当第二条指令进入执行期间, 同时又会读出第三条指令。这样在整个执行期间, 并不需要为读取指令花费额外的周期, 从流水线上看, 实现了每个周一执行一条指令的效率。



下图展示通用工作寄存器的访问时序, 在一个周期内, ALU 操作使用到两个寄存器作为操作数, 并在这个周期内将 ALU 执行结果写入到目标寄存器中。



## 2.9. 复位与中断处理

LGT8XM 支持多个中断源。这些中断以及复位向量在程序空间都对应一个独立的程序向量入口。一般而言, 所有的中断都有单独的控制位控制。当设置了该控制位, 并且使能了内核的全局中断使能位后, 内核才能响应这个中断。

最低的程序空间默认保留为复位以及中断向量区域。PDS8F208A 支持的完整的中断列表请参考中断章节的介绍。这个列表同时也决定了不同中断的优先级。向量地址越低的中断, 对应的中断优先级就越高。复位 (RESET) 具有最高的优先级, 然后是 INTO - 外部中断请求 0。中断向量表的起始地址 (复位向量除外) 可以被重新定义到任何 256 字节对齐的开始处, 需要通过 MCU 控制寄存器 (MCUCR) 中的 IVSEL 位以及 IVBASE 向量基地址寄存器实现。

当内核响应中断后, 全局中断使能标志为 I 会被硬件自动清除。用户可以通过将 I 位使能实现中断嵌套。这样任何随后发生的中断都会中断当前的中断服务程序。I 位在执行中断返回指令 (RETI) 后自动置位, 从而可以正常响应随后发生的中断。

有几种基本的中断类型。

第一种类型由事件触发, 中断事件发生后置位中断标志位。对于这种中断来说, 内核响应中断请求后, 当前的 PC 值被直接替换为实际的中断向量地址, 执行对应的中断服务子程序, 同时硬件自动清除掉中断标志位。中断标志位也可以通过向中断标志位的位置写 1 清除。如果在发生中断时, 中断使能位被清除, 中断标志位仍然会被设置以记录中断事件。等到中断使能后, 这个记录的中断事件会被立即响应。同样, 如果在中断发生时, 全局中断使能位 (SERG. I) 被清除, 对应的中断标志位也会被设置以记录中断事件, 等到全局中断使能位被设置后, 这些被记录的中断将会依照优先级依次执行。

第二种中断类型是当中断条件一直存在时, 中断就一直响应。这种中断不需要中断标志位。如果中断条件在中断使能之前消失, 这个中断将不会得到响应。

当 LGT8XM 内核从中断服务子程序中退出后, 执行流程会返回到主程序中。在主程序中执行一条或几条指令后, 才能响应其他等待的中断请求。

需要注意的是, 系统状态寄存器 (SREG) 在进入中断服务后并不会自动保存, 也不会从中断服务返回后自动恢复。它必须

由软件负责处理。

当使用 CLI 指令禁止中断后，中断将会被立即禁止。在 CLI 指令之后发生的所以中断都不会得到响应。即使是和 CLI 指令执行时同时发生的中断，也不会被响应。下面的例子中说明如何利用 CLI 避免中断打乱 EEPROM 的写时序：

## 2.10. 中断响应时间

LGT8XM 内核针对中断响应进行了优化，使得任何中断在 4 个系统时钟周期内一定得到响应。4 个系统时钟周期后，中断服务子程序进入执行周期。在这 4 个时钟内，中断之前的 PC 值被压入堆栈，系统执行流程跳转到中断向量对应中断服务程序。如果中断发生在一个多周期指令执行期间，内核将保证当前指令正确的执行结束。如果中断发生在系统处于休眠状态下(SLEEP)，中断响应需要额外增加 4 个时钟周期。这增加的时钟周期用于从选择的休眠模式下唤醒操作的同步周期。休眠模式的具体描述，请参考功耗管理的相关章节。

从中断服务子程序中返回需要 2 个时钟周期。在这 2 个时钟周期内，PC 从堆栈中恢复，堆栈指针加 2，并自动使能全局中断控制位。

## 3. 存储单元

- 20Kbyte MTP 在线可编程用户程序空间
- 2Kbyte 的数据 SRAM 空间

### 3.1. 概述

本章节主要描述 PDS8F208A 内部不同的存储单元。LGT8XM 构架支持两种主要的内部存储空间，分别是数据存储空间和程序存储空间。另外，PDS8F208A 系统中还包含了特殊的存储单元，用于存放系统配置信息以及芯片的全局设备号(GUID)。

下面的表格比较清楚描述了 PDS8F208A 系列芯片不同的存储空间配置：

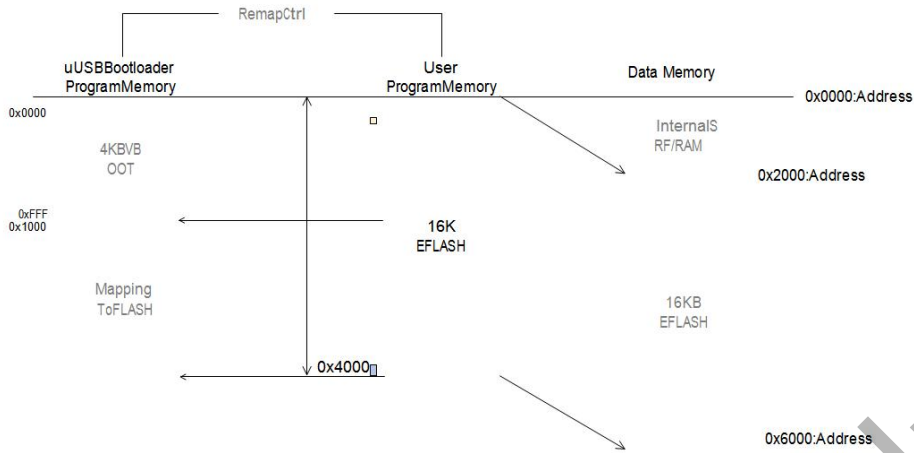
DEVICE	MTP	SRAM
PDS8HF208A	20KB	2KB

### 3.2. 在线可编程 FLASH 用户程序存储单元

PDS8F208A 系列微控制器内部包括 20K 字节的片上在线可编程 MTP 用户程序存储单元。程序 MTP 能保证至少 1,000 次以上的擦写周期。

程序空间也可以通过 LPM 指令直接访问(读取)，这个特点可以实现应用相关的常数查找表。同时 FLASH 程序空间也被映射到系统数据存储空间内，用户也可以使用 LD/LDD/LDS 实现对 FLASH 空间的访问。程序空间被映射到数据存储空间 0x2000 开始的地址范围内。

下图为 PDS8F208A 的储存空间布局和地址映射关系：



**注意：以上地址均为字节地址**

PDS8F208A 的程序空间有两种，分别为 uUSB 自启动程序空间与用户程序空间。可以通过重映射(remap)寄存器控制，将其中一个映射到程序空间的 0x0000 开始。

芯片复位重启后，硬件在执行第一条指令前，会通过检测 FLASH 空间的最后一个地址

(字节)来确定初始的重映射状态。如果 FLASH 空间最后一个字节为 0x55，并且系统配置位使能了重映射启动检测，FLASH 存储器空间将会被映射到当前程序空间。如 FLASH 最后一个字节不是 0x55，并且重映射检测位被使能，系统将 uUSB 自启动空间映射到当前程序空间；如果重映射自检位没有被使能，系统将直接从 FLASH 空间中执行指令代码，软件仍然有机会通过设置重映射寄存器重新配置程序空间。

### 3.3. SRAM 数据存储单元

PDS8F208A 系列微控制器是一种相对复杂的微控制器，它支持多种不同类型的外设，这些外设的控制器被分配在 64 个 I/O 寄存器空间内。可以直接通过 IN/OUT 指令访问。另一些外设的控制寄存器分配在 0x60 ~ 0xFF 区域内，由于这部分空间是映射到数据存储空间内，只能通过 ST/STS/STD 以及 LD/LDS/LDD 等指令访问。

PDS8F208A 的系统数据存储空间从 0 地址开始，分别映射了通用工作寄存器文件，I/O 空间，扩展 I/O 空间以及内部数据 SRAM 空间。最开始的 32 个字节地址对应 LGT8XM 内核 32 个通用工作寄存器。接下来的 64 个地址是可以通过 IN/OUT 指令直接访问的标准 I/O 空间。然后的 160 个地址是扩展 I/O 空间，在接下来就是最多 2K 字节的数据 SRAM。从 0x2000 开始到 0x5FFF 结束的这部分空间，映射了 MTP 程序存储单元。

系统支持 5 种不同的寻址模式可以覆盖到整个数据空间：直接访问，带偏移的间接访问，间接访问，访问前递减地址的间接访问，访问后递增地址的间接访问。通用工作寄存器 R26 到 R31 用于间接访问的地址指针。间接访问可以寻址整个数据存储空间。带偏移地址的间接访问能够寻址到以 Y/Z 寄存器为基地址的附近 63 个地址空间。

当使用支持地址自动递增/递减的寄存器间接访问模式，地址寄存器 X/Y/Z 会在访问发生前/后自动由硬件递减/递增。具体请参考指令集描述部分。

### 3.4. 通用 I/O 寄存器

PDS8F208A 的 I/O 空间有三个通用 I/O 寄存器 GPIOR2/I/O，这三个寄存器可以使用 IN/OUT 指令访问，用于存放用户自定义数据。

### 3.5. 外设寄存器空间

I/O 空间的详细定义，请参考 PDS8F208A 数据手册中“寄存器概述”章节。

PDS8F208A 所以的外设都被分配到 I/O 空间。所有的 I/O 空间地址都可以被 LD/LDS/LDD 以及 ST/STS/STD 指令访问。访问的数据都是通过 32 个通用工作寄存器传递。在 0x00 ~ 0x1F 之间的 I/O 寄存器可以通过位寻址指令 SBI 和 CBI 访问。在这些寄存器中，某一个位的值可以使用 SBIS 和 SBIC 指令检测，用以控制程序的执行流程。具体请参考指令集描述部分。

当使用 IN/OUT 指令访问 I/O 寄存器时，必须寻址 0x00 ~ 0x3F 之间的地址。当使用 LD 或 ST 指令访问 I/O 空间时，

必须通过 I/O 空间在系统数据存储统一映射空间的映射地址 访问(加上 0x20 的偏移)。其他一些分配在扩展 I/O 空间的外设寄存器(0x60 ~ 0xFF)，只能 够使用 ST/STS/STD 和 LD/LDS/LDD 指令访问。

为了与未来的设备兼容，保留位在写操作时必须写 0。不能在保留的 I/O 空间上执行写 操作。

一些寄存器中包括了状态标志，需要被写 1 才能清零。需要注意的是，CBI 和 SBI 指令 仅仅支持特定的位，因此 CBI/SBI 也只能工作在包含这些状态标志的寄存器上。除此之外， CBI/SBI 指令只能工作在 0x00 到 0x1F 这个地址范围内的 寄存器。

### 3.6. FLASH 控制器(E2PCTL)

PDS8F208A 内部实现集成了一个灵活可靠的 EFLASH 读写控制器，可以实现对 FLASH 空 间的在线编程操作，可以通过软件 实现在线自动升级固件的功能。通过 FLASH 控制器访问程序 FLASH 程序空间，只支持基于扇区的擦除(256 字节)以及 32 位宽度的读写访问。



E2PCTL 模拟 E2PROM 功能访问数据 FLASH 空间时，可以支持 8 位、32 位读写宽度。访 问程序 FLASH 空间时，支持页擦 除和 32 位数据读写。由于 PDS8F208A 内部 FLASH 的最小 存储单元为 32 位，因此建议用 32 位访问方式，特别是对于写操 作。32 位访问的读写操作 不仅效率高，也有利于保护 FLASH 存储单元的擦写寿命。

PDS8F208A 内部没有多余的数据 FLASH。因此，LGT8XM 内核与 E2PCTL 共享内部 16K 字 节 FLASH 存储空间。用户可以根 据需要，将 16K 字节 FLASH 空间划分为程序空间和数据空 间。通过配置 E2PCTL 控制器，可以设置模拟 E2PROM 的空间大 小。E2PCTL 使用页交换模式 实现模拟 E2PROM 逻辑，算法以页(256 字节)为单位。因此模拟 256 字节的 E2PROM 空间， 需要 占用 512 字节的 FLASH 空间。具体实现方式，请参考 E2PCTL 算法实现的描述。

### 3.7. E2PCTL 数据寄存器

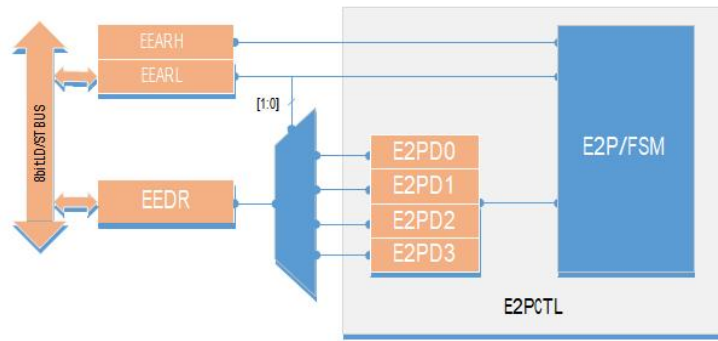
E2PCTL 控制器内部有 4 个字节的数据缓存(E2PD0~3)，此 4 字节的缓存组成最终访问

FLASH 空间的 32 位数据接口。

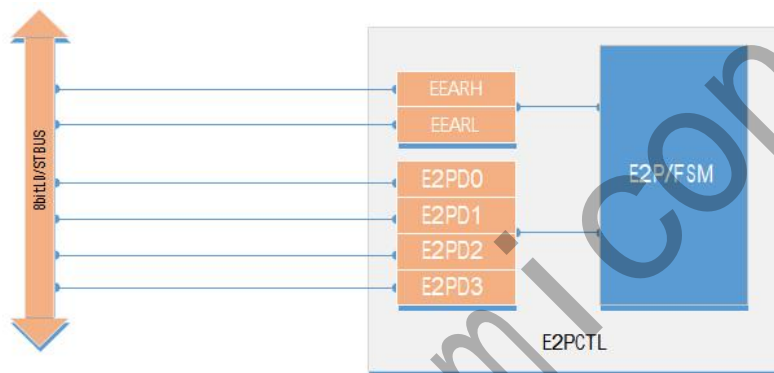
当 E2PCTL 控制器工作在字节读写模式时，EEDR 作为读写字节数据的接口， E2PCTL 更 加 EEARL[1:0]的地址信息加载数 据到正确的数据缓存中，并根据当前 FLASH 目标地址的数据 补齐另外三个字节的数 据，最终将组合的完整 32 位数据更新到 FLASH 中。

当 E2PCTL 工作在 32 位读写模式时，此时仍然可以使用 EEDR 寄存器作为一个公用的 数据接口，通过 EEARL[1:0]作为 地址寻址内部数据缓存，实现读写一个完整的 32 位数据。此外，还可以直接使用数据缓存映射到 I/O 空间的寄存器直接访 问(E0~3)。

E2PCTL 工作在 8 位字节读写模式时的数据访问示意图



E2PCTL 工作在 32 位读写模式时的数据访问示意图



PDS8F208A 的内置 FLASH 为 32 位接口宽度，使用 32 位读写模式将给读写效率和 FLASH 的擦写寿命带来极大的好处，因此建议使用 32 位读写模式。

### 3.8. E2PCTL 模拟 E2PROM 接口算法

我们知道，FLASH 存储器在写之前必须先擦除，而擦除操作是以页面为单位的。PDS8F208A 内置 FLASH 存储器一个页面的大小为 256 字节。因此为了更新页面中的一个字节数据，也需要首先擦除掉整个页面的数据，然后更新目标地址数据，并同时恢复页面中其他字节的数据，整个操作不仅仅耗时，也同时带来因电源意外丢失数据风险。

E2PCTL 内部采用页交换算法实现模拟 E2PROM。页交换算法模式可以保证在执行页擦除操作时，不会因为掉电等意外情况导致原有数据的丢失。同时也交换算法使用 2 个页面空间互为交换的方式交替使用，也增加了模拟 E2PROM 空间的使用寿命。

在效率方面，E2PCTL 控制器实现了一种连续数据更新模式，减少了因反复更新数据带来的重复擦写过程。

在实现方面，E2PCTL 对每一个页面单独管理，并占用一个页面最后 2 个字节作为页面状态的信息。因此用户在使用大于 256 的 E2PROM 模拟空间时，需要注意地址跨过 256 空间的特殊处理。因为每 256 字节空间的最后 2 个字节保留给 E2PCTL 使用，用户无法对这 2 个字节的进行正常的读写。

### 3.9. FLASH 操作的保护措施

如果 VCC 电压偏低，FLASH 的擦写操作可能会因为电压太低而发生错误。

FLASH/数据在低压下的擦写操作错误可能由两种原因。首先，正常的 FLASH 擦写操作需要一个最小工作电压，低于这个电压，操作将会失败而导致数据发生错误。第二个原因，是内核运行在某一频率下，也同样需要一个最小电压要求，当低于这个电压，将会导致指令执行出错，从而使得 FLASH 的操作发生错误。

可以通过下面简单的方法避免类似问题：在供电电压较低时，让系统进入复位状态。这可以通过配置内部的低压检测电路 (VDT) 实现。如果 VDT 检测到当前的工作电压低于设置的阈值，VDT 将会输出一个复位信号。如果 VDT 的阈值不能满足应用的需要，可以考虑在外部增加一个复位电路。

### 3. 10. EEPROM 的操作例程

```
void EEP_init(void)
{
    PRR1 &= ~(0x04); //开启 EEP
    ECCR = 0x80;
    ECCR = 0; //8 位模式
}

void EEP_Eraser(u8 saddr)
{
    if(saddr & 0x1) //擦除高 256Byte
        EEARH = 0x81;
    else //擦除低 256Byte
        EEARH = 0x80;

    EEARL = 0;

    EEER = 0x84; //使能 EEP 擦除
    EEER = 0x02; //执行编程

    //等待完成
    Delay_ms(1000);
    EEER = 0;
}

void EEP_Write(unsigned int addr, unsigned char dat0)
{
    unsigned char c0, c1;

    c0 = addr & 0xff;
    c1 = (addr >> 8) | 0x80;

    EEARH = c1;
    EEARL = c0;

    E2PD0 = dat0; //装填数据

    EEER = 0x04; //使能编程
    EEER = 0x02; //执行编程
    NOP();
    NOP();
    NOP();
    NOP(); //等待完成
}

unsigned char EEP_Read(unsigned int addr)
{
    unsigned char c0, c1;
```

```

c0 = addr;
c1 = (addr>>8)|0x80;

EEARH = c1;
EEARL = c0;

EECR = 0x01;//执行读取

NOP();
NOP();
c0 = E2PD0;//取走数据

return c0;
}

```

### 3.11. 寄存器描述

FLASH 地址寄存器- EEARH/EEARL

EEARH/EEARL		
EEARH: 0x22 (0x42)		默认值: 0x0000
EEARL: 0x21 (0x41)		
bits	EEAR[15:0]	
R/W	R/W	
位定义		
[8:0]	EEARL	EEPROM 访问地址, 最多 512 bytes。
[14:8]	EEARH	保留不用
[15][9]	EEPS	[15]=1, [9]=0: 共同选择地址对象为 EEPROM

当使用 E2PCTL 控制器访问程序 FLASH 区域时, EEAR[14:2]用作访问以 4 字节对齐的整个程序空间。EEAR[1:0]只在访问数据寄存器 EEDR 时使用。具体请参考下面关于 EEDR 数据寄存器的描述。E2PCTL 控制器支持 8/16 位模式, 无论是哪一种模式, 此处的 EEAR 都是以字节对齐寻址。

FLASH 数据寄存器- EEDR/E2PD0

EEDR/E2PD0 - E2PCTL 数据寄存器 0		
EEDR/E2PD0: 0x20 (0x40)		默认值: 0x00
bits	EEDR[7:0]	
R/W	R/W	
位定义		
[7:0]	EEDR E2PD0	E2PCTL 数据寄存器 低字节

FLASH 数据寄存器- E2PD1

E2PD1 - E2PCTL 数据寄存器 1		
E2PD1: 0x5A		默认值: 0x00
bits	E2PD1[7:0]	
R/W	R/W	
位定义		

[7:0]	E2PD1	E2PCTL 数据寄存器 高字节
-------	-------	------------------

**FLASH 模式控制寄存器- ECCR**

ECCR - FLASH/E2PROM 配置寄存器								
ECCR: 0x36 (0x56)					默认值: 0x0C			
bits	WEN	-	ERN	SWM	-	-	-	-
R/W	R/W		R/W	R/W				
初始值	0		0	0				
位定义								
[7]	WEN	ECCR 写使能控制 在修改ECCR前, 必须先将WEN写1, 然后在6个系统周期内, 更新ECCR寄存器的内容						
[6]	-							
[5]	ERN	写1 将复位E2PCTL 控制器						
[4]	SWM	SWM=1: 16 位模式, SWM=0: 8 位模式						
[3]	-							
[2]	-							
[1:0]	-							

**FLASH 访问控制寄存器- EECR**

EECR - FLASH/E2PROM 控制寄存器								
EECR: 0x1F (0x3F)					默认值: 0x00			
bits	EEPEN	-	EEPM1	EEPM0	EERIE	EEMWE	EEWE	EERE
R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W
初始值	0		0	0	0	0	0	0
位定义								
[7]	EEPEN	EEPEN=0: EEPROM 写模式; EEPEN=1, EEPM1=0, EEPM0=0, EEPROM 擦模式						
[6]	-							
[5]	EEPM1	-						
[4]	EEPM0	-						
[3]	EERIE	FLASH/E2PROM 就绪中断使能控制。写1 使能, 写0 禁止。当 EEPE 被硬件自动清零后, E2PROM 就绪中断有效。在 EPROM操作过程中, 将不会产生这个中断						
[2]	EEMWE	EEMWE 用于控制EEWE 是否有效, 当同时设置EEMWE 为1, EEWE 为0后, 在之后的四个周期内, 设置EEWE 为1 将启动编程操作。否则编程操作无效。四个周期后, EEMWE 被自动清零						
[1]	EEWE	FLASH/E2PROM 编程操作使能位						
[0]	EERE	E2PROM 读使能位, 数据将在两个系统周期以后有效						



通用 I/O 寄存器- GPIOR2

GPIOR2 - 通用 I/O 寄存器 2		
GPIOR2: 0x2B (0x4B)		默认值: 0x00
Bits	GPIOR2[7:0]	
R/W	R/W	
初始值	0x00	
位定义		
[7:0]	GPIOR2	通用 I/O 寄存器2, 用于存储用户自定义数据

通用 I/O 寄存器- GPIOR1

GPIOR1 - 通用 I/O 寄存器1		
GPIOR1: 0x2A (0x4A)		默认值: 0x00
Bits	GPIOR1[7:0]	
R/W	R/W	
初始值	0x00	
位定义		
[7:0]	GPIOR1	通用 I/O 寄存器1, 用于存储用户自定义数据

通用 I/O 寄存器- GPIOR0

GPIOR0 - 通用 I/O 寄存器0		
GPIOR0: 0x1E (0x3E)		默认值: 0x00
Bits	GPIOR0[7:0]	
R/W	R/W	
初始值	0x00	
位定义		
[7:0]	GPIOR0	通用 I/O 寄存器0, 用于存储用户自定义数据

程序空间重映射控制寄存器 - RMPCR

RMPCR - 重映射控制寄存器								
RMPCR: 0x00 (0x20)					默认值: 0x08			
bits	WEN	M32	UME	UMS	-	-	RME1	RME0
R/W	R/W	R/W	R/W	R/W	-	-	R/W	R/W
初始值	0	0	0	0	-	-	0	0
位定义								
[7]	WEN	RMPCR 寄存器更新使能控制位 WEN 位写后, 在6个系统周期内完成其他位的更新						
[6]	M32	内核运行模式 1: M32 模式, 可寻址32KB 程序空间 0: M16 模式, 可寻址16KB 程序空间						
[5]	UME	此为固定位1, 不可改写						
[4]	UMS	页交换 CP1 区域使能控制						
[3:2]	-	保留未用						

[1:0]	RME	重映射模式，写此位启动重映射模式更新 01 - uUSB启动模式 10 - FLASH 程序启动模式 其他 - 无效操作
-------	-----	--

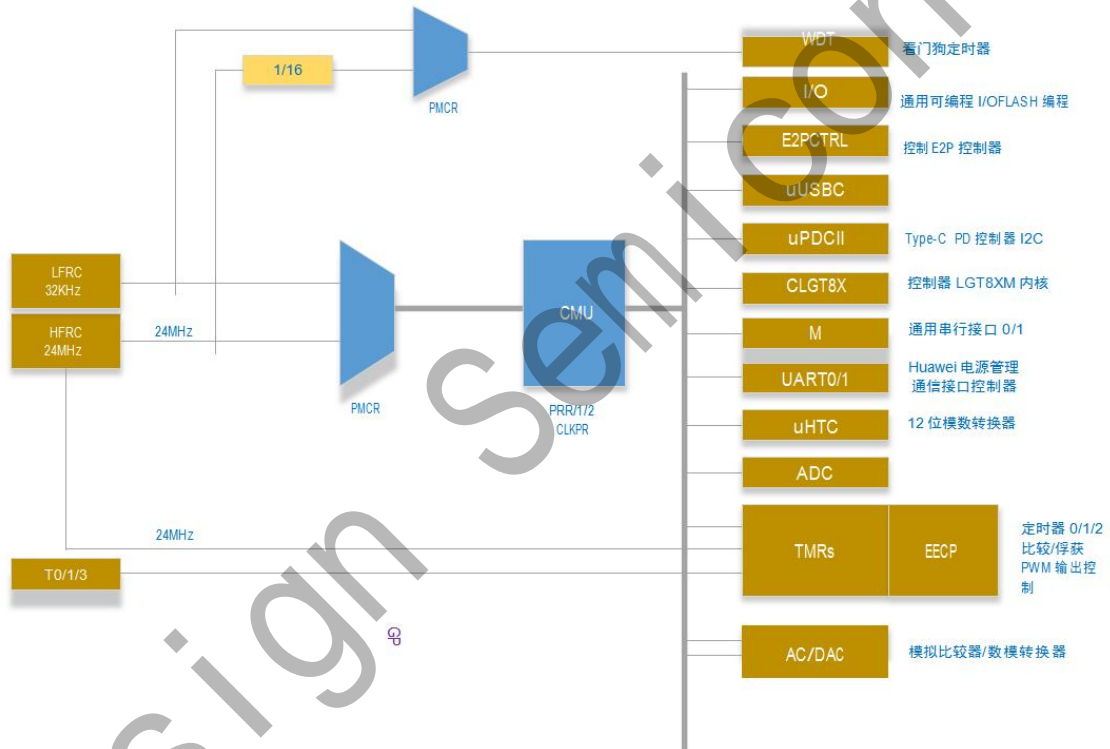
## 4. 系统时钟

### 4.1. 系统时钟分布

PDS8F208A 支持多种时钟输入。系统可以工作在两种主要的时钟源，分别是内部 32KHz 可校准 RC 振荡器，内部 24MHz 可校准 RC 振荡器。

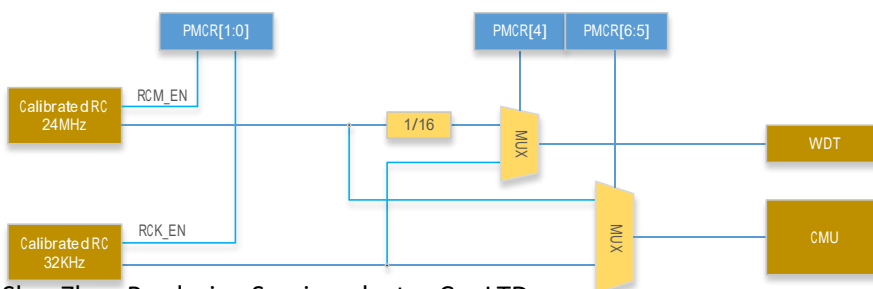
下图为 PDS8F208A 时钟系统分布，CMU 是整个时钟管理的中心，负责系统时钟的分频，为不同的模块产生独立的时钟以及对时钟进行控制等等。一般的应用中，并不不要全部的时钟同时工作，为了减小系统功耗，系统功耗管理根据不同的休眠模式，关闭没有使用的模块时钟。

具体操作细节，请参考功耗管理相关章节。



### 4.2. 时钟源选择

PDS8F208A 支持 4 种时钟源输入，用户可以通过 PMCR 寄存器实现对时钟源的使能控制以及完成主时钟的切换。下面是 PMCR 的控制结构图：

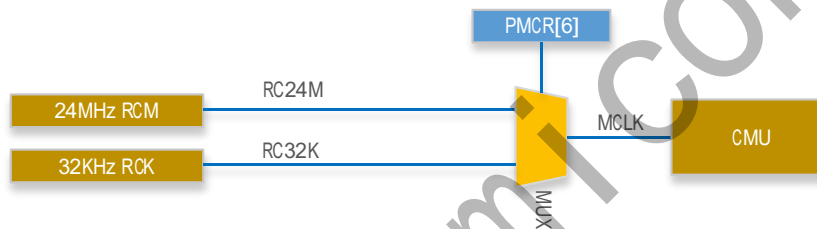


PDS8F208A 内部的 RC 振荡器分为高频和低频两种。PMCR 寄存器的最低 2 位用于控制这两种时钟源。控制关系如下：

PMCR	对应时钟源
PMCR[0]	24MHz RC 使能控制, 1 使能, 0 关闭
PMCR[1]	32KHz RC 使能控制, 1 使能, 0 关闭

PDS8F208A 系统上电后，默认使用 24MHz RC 作为系统时钟源，内核工作在时钟源的 8 分频 (3MHz)。用户可以通过设置 PMCR 寄存器以及系统预分频寄存器 (CLKPR) 改变默认配置。如果用户需要更改主时钟源配置，需要在切换时钟前保证切换后的时钟源处于稳定的工作状态。因此需要在切换主时钟源之前，通过 PMCR[1:0] 使能所需时钟源，并等待到时钟稳定后才能进行切换。

时钟源使能并等待稳定后，可以通过 PMCR[6] 切换主时钟。PMCR[6] 用于选择高速时钟源和低速时钟源。



主时钟源选择：

PMCR[6]	PMCR[5]	主时钟源
0	0	内部24MHz RC 振荡器(系统默认)
1	0	内部32KHz RC 振荡器
其他		保留未用

### 4.3. 时钟源控制时序

为保护 PMCR 寄存器被意外修改，对 PMCR 寄存器的修改需要严格按照指定的时序进行。PMCR 寄存器的最高位 (PMCR[7]) 用于实现时序控制。用户在修改 PMCR 其他位之前，必须首先要将 PMCR[7] 置 1，在置 1 操作后的 6 个周期内，更改 PMCR 其他寄存器的值。6 个周期之后，对 PMCR 的直接修改将失效。

下面以切换到外部高速晶振为例，列出建议的操作步骤：

- 切换主时钟源
- 设置 PMCR[7] = 1
- 在六个周期内，设置 PMCR[6:5]，完成系统时钟源的切换
- 执行几个 NOP 操作，提高稳定性(可选操作)

### 4.4. 系统时钟预分频控制

PDS8F208A 内部有一个系统时钟预分频器，可以通过时钟预分频寄存器 (CLKPR) 进行控制。这种功能可以用于当系统不需要非常高的处理能力时，减小系统功耗。预分频设置对系统支持的时钟源都有效。时钟预分频能够影响到内核执行时钟以及所以同步外设。

当在不同的时钟预分频设置之间切换时，系统时钟预分频确保在切换过程中不会产生毛刺，而只会保证不会有过高频的中间状态。分频切换是立即执行的，当寄存器改变生效后，最多在 2~3 个当前系统时钟周期后，系统时钟就切换到了新的分频时钟。

- 为了避免对时钟分频寄存器的误操作，对 CLKPR 的修改也必须遵循一个特殊的时序流程
- 设置时钟预分频更改使能位 (CLKPCE) 为 1，CLKPR 其他所以位为 0

- 在四个周期内，把需要的值写入 CLKPS，同时 CLKPCE 写 0

在更改时钟预分频寄存器前，需要禁止中断功能，以保证写时序能够完整的进行。关于主时钟预分频寄存器 CLKPR 的具体定义，请参考本章节寄存器描述部分。

#### 4.5. 内部 RC 振荡器校准

PDS8F208A 内部包含两个可校准 RC 振荡器，经过校准后，均可达到±1%以内的精度。其中 24MHz RC 默认用于系统工作时钟。

PDS8F208A 出产前，内部 24MHz HFRC 和 32KHz LFRC 都进行了校准，并把校准值写入系统配置信息区域。系统省电过程中，这些校准值将会被读入到内部寄存器中，通过寄存器实现对 RC 频率的重新校准。

校准寄存器位于 I0 地址空间，用户程序可以读写。对于频率有特殊需求的应用，可以通过修改校准寄存器方式调整内部振荡器的频率输出。修改校准寄存器不会改变出厂配置信息，系统重新上电或者用户启动的配置位重新加载操作，校准寄存器将会恢复到出厂设置。

#### 4.6. 寄存器定义

24MHz HFRC 振荡器校准寄存器- RCMCAL

RCMCAL - 24MHz HFRC 校准寄存器	
RCMCAL: 0x66	默认值: 出厂配置
Bits	RCCAL [7:0]
R/W	R/W
位定义	
[7:0]	RCCAL 系统上电后，寄存器的值将被系统配置信息中的 RC 校准值替换。

时钟源管理寄存器- PMCR

PMCR - 时钟源管理寄存器							
PMCR: 0xF2	默认值: 0x03						
Bits	PMCE	CLKFS/CLKSS	-	-	-	RCKEN	RCMEN
R/W	R/W	R/W	-	-	-	R/W	R/W
位定义							
[0]	RCMEN	内部 24MHz RC 振荡器使能控制，1 使能，0 禁止					
[1]	RCKEN	内部 32KHz RC 振荡器使能控制，1 使能，0 禁止					
[2]	-	保留未用					
[3]	-	保留未用					
[4]	-	保留未用					
[5]	CLKSS	主时钟源选择控制，选择时钟源类型，请参考时钟源选择部分					
[6]	CLKFS	主时钟源频率控制，选择时钟频率类型，请参考时钟源选择部分					
[7]	PMCE	PMCR 寄存器更改使能控制位。 在更改 PMCR 其他位置之前，必须首先设置此位，然后在四个周期内设置其他位的值。					

32KHz RC 振荡器校准寄存器- RCKCAL

RCKCAL - 32KHz RC 校准寄存器		
RCKCAL: 0x67		默认值: 出厂设置
Bits	RCKCAL[7:0]	
R/W	R/W	
位定义		
[7:0]	RCKCAL	将校准值写入RCKCAL 寄存器完成对32KHz RC 振荡器的校准

主时钟预分频寄存器- CLKPR

CLKPR - 主时钟预分频寄存器								
CLKPR: 0x61					默认值: 0x03			
Bits	WCE	-	CLKOE	-	PS3	PS2	PS1	PS0
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
位定义								
[3:0]	CLKPS	时钟预分频选择位						
		PS3	PS2	PS1	PS0	分频参数		
		0	0	0	0	1		
		0	0	0	1	2		
		0	0	1	0	4		
0	0	1	1	8 (默认配置)				

寄存器- T MCKR

TMCKR - 时钟源管理寄存器								
TMCKR: 0xCE					默认值: 0x0			
Bits	WCE	T2C_ENO	T1C_ENO	TOC_ENO	RC48EN	T2C_SEL	T1C_SEL	TOC_SEL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位定义								
[0]	TOC_SEL	Timer0 时钟源选择位, 设置为 0 即选择 RC1.5M 时钟, 设置为 1 即选择 RC32K 时钟						
[1]	T1C_SEL	Timer1 时钟源选择位, 设置为 0 即选择系统时钟, 设置为 1 即选择 RC48M 时钟						
[2]	T2C_SEL	Timer2 时钟源选择位, 设置为 0 即选择系统时钟, 设置为 1 即选择 RC48M 时钟						
[3]	RC48EN	RC48M 时钟使能位, 要使用 RC48M 时钟作为计数器的时钟源, 需先置位此位使能 RC48M 时钟输出						
[4]	TOC_ENO	Timer0 时钟源标志位, 0 表示当前使用 RC1.5M 时钟, 1 表示当前使用 RC32K 时钟						
[5]	T1C_ENO	Timer1 时钟源标志位, 0 表示当前使用系统时钟, 1 表示当前使用 RC48M 时钟						
[6]	T2C_ENO	Timer2 时钟源标志位, 0 表示当前使用系统时钟, 1 表示当前使用 RC48M 时钟						
[7]	WCE	寄存器更新使能位, 先置位 WCE, 然后在 6 个指令周期内完成对此寄存器的更新						

## 5. 功耗管理

### 5.1. 概述

休眠模式通过关闭系统时钟以及时钟模块, 从而减小系统功耗。PDS8F208A 提供了非常灵活多样的

休眠模式和模块控制器，用户可以根据应用，实现最理想的低功耗配置。

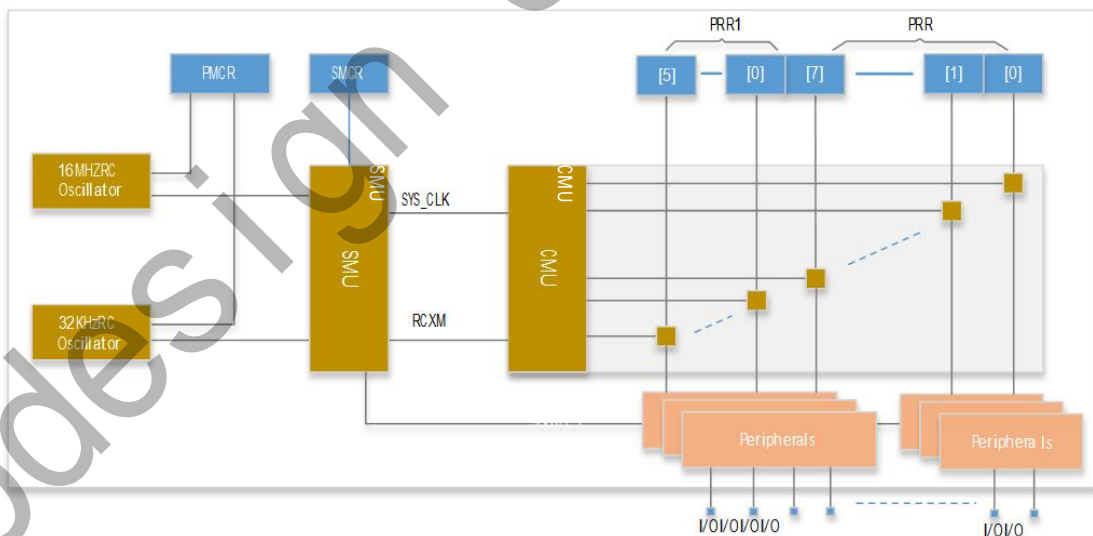
PDS8F208A 在进入休眠模式时，并不会自动关闭模拟功能模块，比如 ADC，DAC，比较器(AC)，低电压复位模块(LVD)等等，软件需根据应用要求，在进入休眠前关闭不需要的模拟功能，并在系统唤醒后恢复正确的状态。

PDS8F208A 支持多种休眠模式，其中包括 ADC 专用的噪声消除模式，用于消除 ADC 转换过程中数字部分对 ADC 电源的干扰。除此之外，其他均为功耗控制模式，共分为四种：

休眠模式	功能说明
空闲模式 (IDLE)	仅仅关闭内核时钟，其他外设模块正常工作，所有有效中断源均可以将内核唤醒
省电模式 (Save)	与DPS0 模式相同，Save 模式为与LGT8FX8D 保持兼容
掉电模式 (DPS0)	与 Save 模式相同，支持唤醒源包括： <ul style="list-style-type: none"> <li>● 所有引脚电平变化</li> <li>● 看门狗定时唤醒</li> <li>● 异步模式的 TMR2 唤醒</li> </ul>
掉电模式 (DPS1)	关闭所有内外部振荡器，支持唤醒源包括： <ul style="list-style-type: none"> <li>● 所有引脚外部电平变化</li> <li>● 外部中断 0/1</li> <li>● 工作于 32K LFRC 的看门狗定时器</li> </ul>

在休眠过程中，所有寄存器信息以及 RAM 数据均不会丢失。唤醒后，内核从休眠前的最后一条指令继续执行。

系统功耗管理示意图：



如上图所示，PDS8F208A 主要通过休眠模式控制器 (SMU) 以及时钟管理单元 (CMU) 控制整个系统的功耗。从节省功耗的级别上，我们可以把功耗分为 4 个等级：

第一级是通过 PRR 寄存器控制模块工作时钟，通过关闭没有使用模块的时钟，节省系统运行的动态功耗。一般情况下，这种级别能够节省的功耗并不明显。

第二级是通过切换主时钟源到低频时钟上，并关闭没有使用的时钟源模块以及其他模拟模块，这种模式基本上可以得到非常可观的系统运行功耗和休眠功耗。

第三级别是通过让系统进入到掉电模式 (DPS1)，DPS1 模式下 PDS8F208A 可以获得极地的待机功耗，从断电模式唤醒后，软件可以通过 MCUSR 寄存器读取复位前的状态。

## 5.2. 休眠模式与唤醒源

PDS8F208A 支持 4 种休眠模式，用户可以根据应用需求选择合适的休眠模式。SMCR 寄存器包含了休眠模式的控制设置，执行 SLEEP 指令后，内核进入休眠模式。为获得更加理想的休眠功耗，建议在内核进入休眠模式前，关闭所有没有使用的时钟以及模拟模块。但需要注意的是，某些唤醒源的产生需要工作时钟，如果需要使用这类唤醒源，请保持相关时钟源的工作状态。

休眠模式与唤醒方式：

休眠模式	有效时钟				唤醒源							
	内核时钟	外设时钟	时钟	异步时钟	引脚电平变化	外部中断	地址匹配	中断	转换结束	看门狗溢出	外设中断	电平变化
空闲模式 (IDLE)		X	X	X	X	X	X	X	X	X	X	X
ADC 噪声抑制			X	X	X	X	X	X	X	X		X
省电模式 (SAVE)				X	X	X	X	X		X		X
掉电模式 (DPS0) (With RC32K)				X	X	X		X		X		X
掉电模式 (DPS1) (Without RC32K)				X	X	X		X				X

如果需要进入以上 4 种休眠模式，SMCR 中的 SE 位必须置 1，使能休眠模式控制。然后执行一条 SLEEP 指令即可。SMCR 中的 SMO/1/2 用于选择不同的休眠模式。具体的信息请参考下面的描述。

在 MCU 处于休眠模式下，如果唤醒源有效，MCU 将会在 4 个周期后被唤醒，继续执行指令。如果中断保持有效，中断也将立即响应，进入中断服务子程序。如果在 SLEEP 模式下发生了系统复位，MCU 也将会被唤醒，并从复位向量开始执行。

当 MCU 处于 Power/Off 模式下，系统可以通过外部中断 INTO/1 唤醒，唤醒后 MCU 将从 sleep 前的位置继续执行。

## 5.3. 空闲模式 (IDLE)

当 SM2...0 设置为 000，执行 SLEEP 指令后，MCU 进入到 IDLE 模式，IDLE 模式将会关闭掉内核工作时钟，除此之外的其他外设都能正常工作。

IDLE 模式可以通过外部中断以及内部中断等唤醒。如果不需要使用比较器以及 ADC 作为唤醒源，建议将其关闭。

IDLE 模式因为仅仅关闭了内核运行的时钟，所以并不能得到明显的功耗降低。IDLE 模式下，内核也将停止执行和取指令，因此可以降低内部程序 FLASH 的运行功耗。

但 IDLE 模式拥有比较灵活的唤醒方式，用户可以通过降低系统主时钟以及关闭不需要的模块获取更加理想的运行功耗。

## 5.4. ADC 噪声抑制模式

当 SM2...0 设置为 001，执行 SLEEP 指令后，MCU 进入 ADC 噪声抑制模式。此模式下，内核以及大部分外设都将停止工作，ADC，外部中断，TWI 地址匹配，WDT 以及工作在异步时钟模式下的定时/计数器 2 都可以正常工作。

ADC 噪声一直模式主要用于为 ADC 转化提供一个良好的工作环境。降低数字模块对模拟转换的高频干扰。进入这个模式后，ADC 将自动启动采样转换，转换的数据保存到 ADC 数据寄存器后，ADC 转换结束中断将 MCU 从 ADC 噪声模式下唤醒。

## 5.5. 省电模式 (Save)

当SM2...0 设置为010, 执行SLEEP 指令后, MCU 进入到Save 模式。这种模式下, 系统 将关闭掉所有模块的工作时钟。此模式因为关闭了所有模块的工作时钟, 因此只能通过异步模式唤醒, 外部中断, TWI 地址匹配以及工作在独立时钟源模式下的WDT 都可以产生此模式下的唤醒信号。

此种模式可以关闭除主时钟源以为的所有模块。为实现更加理想的运行功耗, 建议在进入此中模式前, 将系统主时钟切换到内部 32K RC 或者外部 32KHz 低频晶振, 然后关闭掉所以没有被使用的时钟源以及模拟模块。

## 5.6. 掉电模式 DPS0

当SM[2:0]设置为110, 执行SLEEP 指令后, MCU 将进入到DPS0 模式。进入DPS0 后, 除内部32KHz RC外, 其他时钟源均被关闭。此种模式可以通过外部中断INT0/1 唤醒; 如果 使能了WDT的中断功能, 也可以通过WDT实现定时唤醒。

## 5.7. 掉电模式 DPS1

当SM[2:0]设置为011, 执行SLEEP 指令后, MCU 将进入到DPS1 模式。进入DPS1 后, 系统所有时钟源均被关闭。此种模式可以使用 I0 的电平变化, 看门狗唤醒。

## 5.8. FLASH 电源控制以及快速唤醒

当系统处于 SLEEP 模式后, 内核将不会继续执行指令, 此时可以选择关闭 FLASH 的电源, 以获得更低的待机功耗。这个功能可以通过MCUCR 寄存器的FPDEN 位控制实现;

在掉电模式下, 系统可以使用外部中断或者WDT 唤醒, 为了滤除外部信号可能的干扰, 内部唤醒电路包含了一个可配置的滤波电路, 用户可以根据需要选择合适的滤波宽度。滤波电路的配置可以通过MCUCR 寄存器的FWKPEN 实现。

MCUCR [FWKPEN] 滤波宽度控制:

FWKPEN	滤波宽度
0	260us (默认)
1	32us

## 5.9. 寄存器描述

休眠模式控制寄存器- SMCR

SMCR - 休眠模式控制寄存器					
SMCR: 0x33 (0x53)			默认值: 0x00		
Bits		SM2	SM1	SM0	SE
R/W		R/W	R/W	R/W	R/W
<b>位定义</b>					
[0]	SE	休眠模式使能控制位, 设置为 1 后, 执行 SLEEP 指令, 内核将进入休眠模式。SE位可以保护系统意外进入休眠模式。唤醒后, 建议立刻清除 SE 位。			
[3:1]	SM	休眠模式选择			
		SM2	SM1	SM0	模式说明
		0	0	0	IDLE 模式
		0	0	1	ADC 噪声抑制模式
		0	1	0	Save 模式
		0	1	1	DPS1 模式
		1	1	0	DPS0 模式
		Others			保留不用
[7:4]	-	保留不用			

## 省电控制寄存器- PRR

PRR - 省电控制寄存器								
PRR: 0x64					默认值: 0x00			
Bits	7	6	5	4	3	2	1	0
	PRTWI	PRTIM2	PRTIM0	PRUART1	PRTIM1	PRSPI	PRUART0	PRADC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位定义								
[0]	PRADC	设置为1, 关闭ADC 控制器时钟						
[1]	PRUART0	设置为1, 关闭UART0 模块的时钟						
[2]	PRSPI	设置为1, 关闭SPI 模块的时钟						
[3]	PRTIM1	设置为1, 关闭定时/计数器1 的时钟						
[4]	PRUART1	设置为1, 关闭UART1 模块的时钟						
[5]	PRTIM0	设置为1, 关闭定时/计数器0 的时钟						
[6]	PRTIM2	设置为1, 关闭定时/计数器2 的时钟						
[7]	PRTWI	设置为1, 关闭TWI 模块的时钟						

## 省电控制寄存器- PRR1

PRR1 - 省电控制寄存器1								
PRR1: 0x65					默认值: 0x00			
Bits	7	6	5	4	3	2	1	0
	PRPDC	PRHTC	PRWDT	PRPDC1	-	PREFL	PRPCI	PRACP
R/W	R/W	R/W	R/W	R/W	-	R/W	R/W	R/W
位定义								
[0]	PRACP	设置为 1, 关闭比较器模块时钟						
[1]	PRPCI	设置为1, 关闭外部引脚变化以及外部中断模块时钟						
[2]	PREFL	设置为1, 关闭FLASH 控制器接口时钟						
[3]	-	保留不用						
[4]	PRPDC1	设置为 1, 关闭 PD1 模块时钟						
[5]	PRWDT	设置为1, 关闭WDT 计数器时钟						
[6]	PRHTC	设置为 1, 关闭 HT 模块时钟						
[7]	PRPDC	设置为 1, 关闭 PDO 模块时钟						

## MCU 控制寄存器- MCUCR

MCUCR - MCU 控制寄存器								
MCUCR: 0x35 (0x55)					默认值: 0x00			
Bits	7	6	5	4	3	2	1	0
	FWKEN	FPDEN	EXRFD	PUD	IRLD	IFAIL	IVSEL	WCE
R/W	R/W	R/W	R/W	R/W	W/O	R/O	R/W	R/W
位定义								
[0]	WCE	MCUCR 更新使能位, 在更新MCUCR 之前, 需要首先设置此位, 然后在6 个周期内完成对MCUCR 寄存器的更新						
[1]	IVSEL	中断向量选择位, 此位置 1 后, 中断向量地址将根据 IVBASE 寄存器的值映射到新的地址						

[2]	IFAIL	系统配置位加载失败标志位， 0 = 配置信息校验通过 1 = 配置信息加载失败
[3]	IRLD	写1 将重新加载系统配置信息
[4]	PUD	全局上拉禁止位 0 = 使能全局上拉控制 1 = 关闭所有 I/O 的上拉电阻
[5]	EXRFD	外部复位滤波禁止位 0 = 使能外部复位的 (190us) 数字滤波器 1 = 禁用外部复位的数字滤波电路
[6]	FPDEN	Flash Power/down 使能控制 0: 系统SLEEP 后FLASH 保持上电状态 1: 系统SLEEP 后FLASH 断电
[7]	FWKEN	快速唤醒模式使能控制，仅对Power/Off 模式有效 0: 260us 滤波延时 1: 32us 滤波延时

## 5.10. 低功耗应用

LDO 在正常工作模式下，根据负载轻重不同，自身功耗大概有 10~100uA。在低功耗模式下，LDO 处于低负载模式，自身功耗降低至小于1uA。在低负载模式下，LDO 只能驱动较低的负载 (< 2mA)，因此，软件开启低 负载模式下前，需要首先降低系统运行功耗，包括关闭所有模拟功能模拟，关闭 I/O 驱动，并降低系统运行频率。等待系统功耗降低后，再将 LDO 切换到低负载模式。

同理，唤醒后，需要先恢复 LDO 的驱动能力，等待 LDO 输出稳定，然后才能恢复系统 时钟，配置开启其他外设模块。

LDOBR 寄存器

LDOBR 控制寄存器

地址: 0 x AC								默认值: 0 x 03	
Bit	7	6	5	4	3	2	1	0	
	WCE	LDO_APE	-	LDO_IM		LDO_VS			
R/W	R/W	R/W	-	R/W		R/W			
Bit	Name	描述							
7	WCE	LDOBR 寄存器更新使能控制位 在更新LDOBR 寄存器之前，需要先写WCE 位为1，在之后的6 个周期内完成对LDOBR 寄存器的更新							
6	LDO_APE	LDO 自动切换使能位，用于睡眠模式							
5	-	-							
4:3	LDO_IM	LDO 负载电流模式选择位 00: 只带低负载，适用于深度睡眠模式 01/10: 可带高负载，适用于一般睡眠模式. 此时 WDT 可以正常工作 11: 可带高负载，适用于正常工作模式							

2:0	LDO_VS	<p>LDO 输出电压选择位</p> <p>000: LDO 输出电压为1.36V</p> <p>001: LDO 输出电压为1.41V</p> <p>010: LDO 输出电压为1.47V</p> <p><b>011: LDO 输出电压为1.50V[默认]</b></p> <p>100: LDO 输出电压为1.53V</p> <p>101: LDO 输出电压为1.59V</p> <p>11x: LDO 输出电压为1.62V</p>
-----	--------	--

LDOCR - LDO 控制寄存器

LDOCR - LDO控制寄存器

地址: 0xCB				默认值:				
Bit	7	6	5	4	3	2	1	0
	WCE	RCM_IS1	RCM_ISO	HLDO_IM		LDO_VS		
R/W	R/W	R/W		R/W		R/W		
Bit	Name	描述						
7	WCE	HLDOCR 寄存器更新使能控制位 在更新HLDOCR 寄存器之前, 需要先写WCE 位为1, 在之后的6 个周期内 完成对 HLDOCR 寄存器的更新						
6:5	-	RC24M 振荡器内部电流选择位 测试用, 正常工作请保留默认 01 的配置						
4:3	LDO_IM	LDO 负载电流模式选择位 00: 只带低负载, 适用于深度睡眠模式 01/10: 可带高负载, 适用于一般睡眠模式, 此时 WDT 可以正常工作 11: 可带高负载, 适用于正常工作模式						
2:0	LDO_VS	LDO 输出电压选择位 000: LDO 输出电压为1.36V 001: LDO 输出电压为1.41V 010: LDO 输出电压为1.47V 011: LDO 输出电压为1.50V[默认] 100: LDO 输出电压为1.53V 101: LDO 输出电压为1.59V 11x: LDO 输出电压为1.62V						

## 6. 系统复位与看门狗

### 6.1. 概述

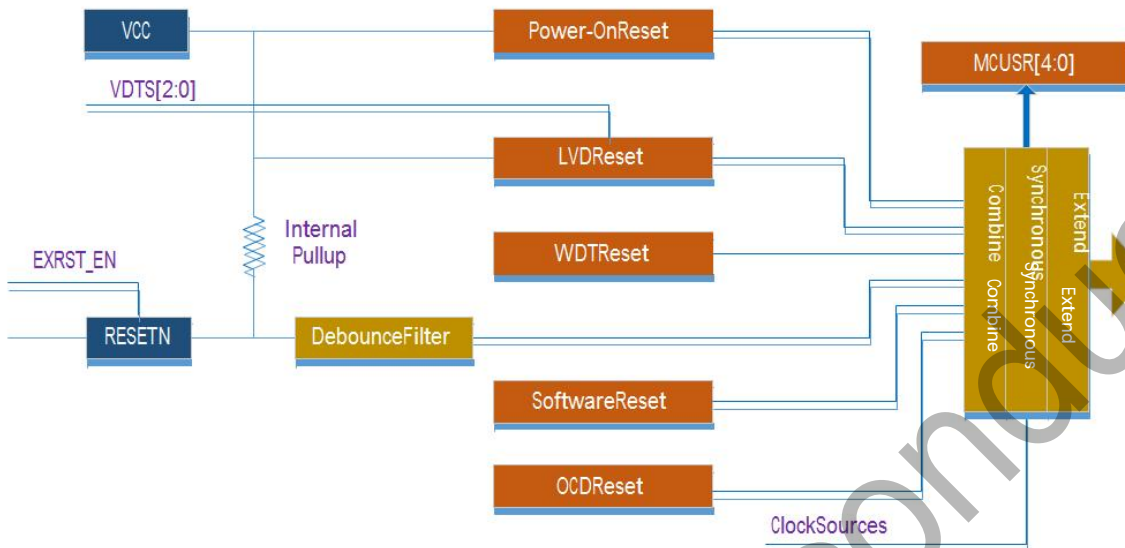
系统复位以后, 所有的 I/O 寄存器都会被设置为它们的初始值, 程序从复位向量处开始 执行。PDS8F208A 的中断向量地址上, 必须用一个RJMP - 相对跳转指令跳转到复位处理程序。如果程序没使用到中断, 没有使能中断源, 中断向量也就不会被使用, 中断向量区域就可以用来存放用户的程序代码。

复位有效后, 所有 I/O 端口立即进入它们的初始状态。大部分 I/O 的初始化状态为输入 并关闭掉内部上拉电阻。有模拟输入功能的 I/O, 也初始化为数字 I/O 功能。

当复位变为无效后, PDS8F208A 内部的定时计数器开始启动, 用于展宽复位。展宽复 位信号的宽度

用于保证系统中的电源以及时钟等模块进入到稳定的状态。

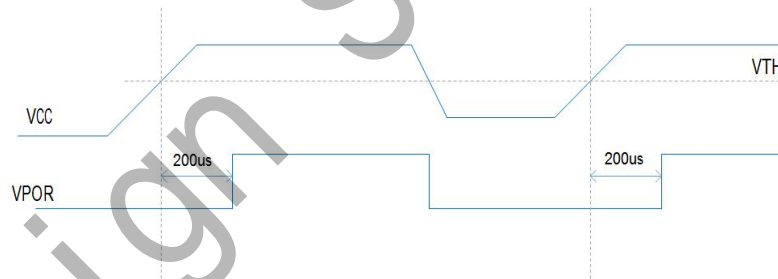
## 6.2. 复位系统结构图：



## 6.3. 上电复位

上电复位信号由内部的电压检测电路产生。当系统电源(VCC)低于检测阈值时，上电复位信号有效。上电复位的检测阈值，请参考电气参数部分。

上电复位电路能够保证芯片在上电过程中处于复位状态，芯片上电后能够从一个已知的稳定的状态开始运行。上电复位信号也会被芯片内部的计数器展宽，以保证上电后内部的各种模拟模块，比如 RC 振荡器等能够进入稳定的工作状态。

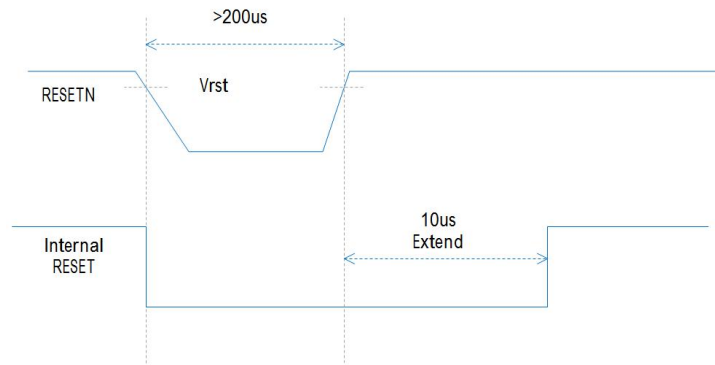


## 6.4. 外部复位

在外部复位引脚(RSTN)上施加一个低电平，外部复位立即有效。低电平的宽度要大于一个

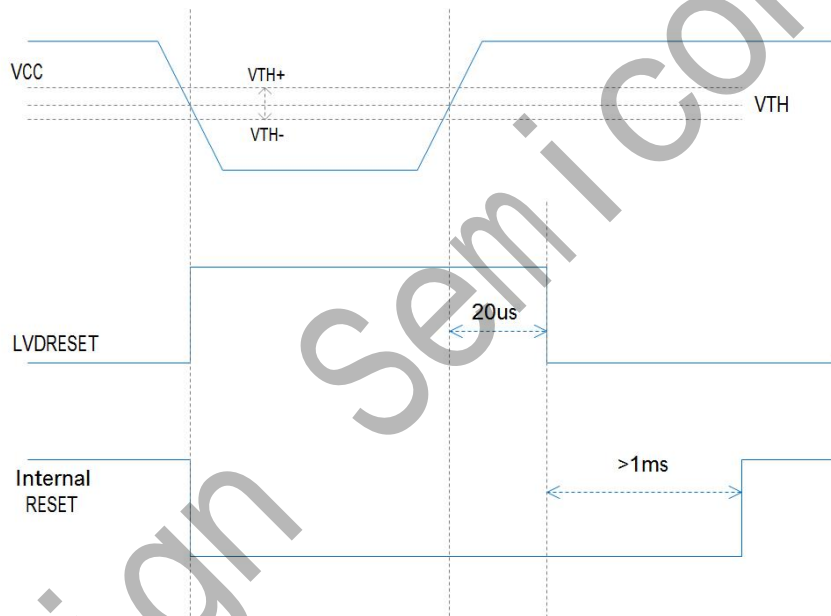
最小复位脉冲宽度要求。外部复位为异步复位，即使芯片没有时钟工作，外部复位仍然能

够对芯片进行复位。PDS8F208A 的外部复位引脚同时也可以作为通用 I/O 使用。在芯片上电以后，默认作为外部复位功能。用户可以通过寄存器配置，关闭该引脚的外部复位功能，从而可以当作普通的 I/O 使用。具体使用请参考 IOCR 寄存器的描述部分。



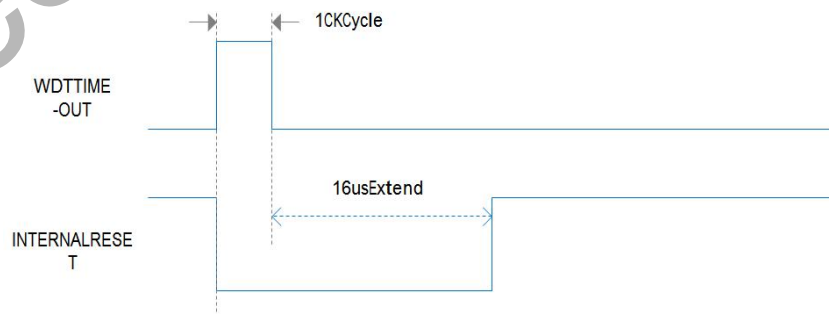
### 6.5. 低电压检测 (LVD) 复位

PDS8F208A 内部包含一个可编程低电压检测 (LVD) 电路。LVD 同样是检测 VCC 的电压变化，但与上电复位不同的是，LVD 可以选择检测电压的阈值。用户可以通过直接通过操作 VDTCR 寄存器在不同的电压阈值之间选择。LVD 的电压检测电路具有  $\pm 10mV \sim \pm 50mV$  的迟滞特性，用于滤除 VCC 电压的抖动。当 LVD 使能后，如果 VCC 的电压下降到设定的复位阈值，LVD 复位将立刻有效。当 VCC 增加到复位阈值以上后，内部的复位展开电路启动，将复位继续展宽至少 1 毫秒。



### 6.6. 看门狗复位

当看门狗定时器溢出时，如果使能了看门狗系统复位功能，将立刻产生一个周期的系统复位信号。看门狗复位信号通用也会被内部的延时光计数器展宽。看门狗控制器的详细操作，请参考下面的详细介绍部分。



### 6.7. 软件复位、OCD 复位

软件复位是用户通过操作 VDTCR 寄存器的第六位触发，软件复位的时序与看门狗复位 完全相似。内部将复位信号展宽 16us。

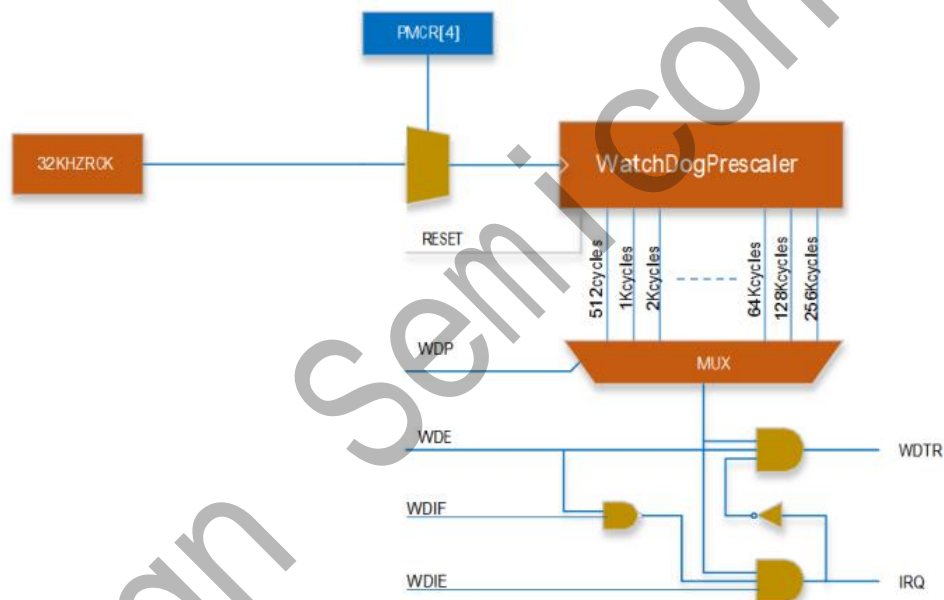
OCD 复位由芯片内部的调试器单元产生，OCD 复位一般是由调试器控制，用户软件无法 触发 OCD 复位。

### 6.8. 看门狗定时器

- WDT 时钟固定为内部 32K
- 支持中断模式，复位模式以及复位中断模式
- 定时器超时最大可到 8 秒

PDS8F208A 内部是一个 18 位的计数器。WDT 定时器的工作时钟可以是内部的 32KHz RC 振荡器，也可以是内部 24MHz RC 振荡器的 16 分频。WDT 计数器溢出后，可以输出 一个中断或者一个系统复位信号。在正常使用时，需要软件执行一个 WDR - 看门狗定时 器复位指令在溢出之前重启计数器。如果系统没有及时的执行 WDR 指令，WDT 将会产 生中断或系统复位。

看门狗定时器的结构图



在中断模式下，WDT 溢出后会产生一个中断请求信号。可以使用这个中断作为休 眠模式的唤醒信号，也可以作为一 个一般的系统定时器使用。比如可以使用这个中断限制某个操作的执行时间，在溢出中终止当前某一个任务。在系统复位模式 下，WDT 在计数器溢出后立刻产生一个系统复位信号。最典型的用途就是用于防止系统死机或跑飞。第三种模式，就是复 位中断模式，结合了中断和复位两种功能。首先系统将响应 WDT 中断功能，退出WDT 中断复位程序后，立刻切换到复位 模式。这个功能可以支持在复 位之前保存一些比较关键的参数信息。

为了防止 WDT 被意外禁止，关闭 WDT 的操作必须按照一个严格定义的时序进行。下面 的代码描述如何关闭看门狗 定时器。下面的例子假设中断已经被禁止，这样整个操作流 程就不会被中断。



看门狗使能以及关闭操作的示例代码

汇编代码

```
WDT_OFF:
; Turn off global interrupt
CLI
; Reset watchdog timer
WDR
; Clear WDRF in MCUSR
IN r16, MCUSR
ANDI r16, ~(1 << WDRF)
OUT MCUSR, r16
; Write logical one to WDCE and WDE
; Keep old Prescaler setting to prevent unintentional time-out
LDS r16, WDTCR
ORI r16, (1 << WDCE) | (1 << WDE)
STS WDTCR, r16
; Turn off WDT
LDI r16, (0 << WDE)
STS WDTCR, r16
; Turn on global interrupt
SEI
RET
```

C 语言代码

```
void WDT_OFF(void)
{
    _disable_interrupt();
    _watchdog_reset();
    /* Clear WDRF in MCUSR */
    MCUSR &= ~(1 << WDRF);
    /* Write logical one to WDCE and WDE */
    /* Keep old Prescaler setting to prevent unintentional time-out */
    WDTCR |= (1 << WDCE) | (1 << WDE);
    /* Turn off WDT */
    WDTCR = 0x00;
    _enable_interrupt();
}
```

[使用提示]

如果WDT被意外使能，比如程序跑飞，芯片会被复位，但是WDT仍然还是在使能状态。如果用 户代码里没有处理WDT，这将会导致循环复位。为避免这种情况，建议用户软件在初始化程序 中清除看门狗复位标记位(WDRF)和WDE控制位。

下面的代码描述如何改变看门狗定时器的超时值。

汇编代码
<pre> WDT_TOV_Change:     ; Turn off global interrupt     CLI     ; Reset watchdog timer     WDR     ; Start timed sequence     LDS r16, WDTCR     ORI r16, (1 &lt;&lt; WDCE)   (1 &lt;&lt; WDE) STS WDTCR,     r16     ; -- Got for cycles to set the new value from here --     ; Set new time-out value = 64k cycles     LDI r16, (1 &lt;&lt; WDE)   (1 &lt;&lt; WDP2)   (1 &lt;&lt; WDPO) STS WDTCR,     r16     ; -- Finished setting new value, used 2 cycles --     ; Turn on global interrupt     SEI RET                 </pre>
C 语言代码
<pre> void WDT_TOV_Change(void) {     _ disable_interrupt();     _ watchdog_reset();     /* Start timed sequence */     WDTCR  = (1 &lt;&lt; WDCE)   (1 &lt;&lt; WDE);     /* Set new time-out value = 64K cycles */     WDTCR  = (1 &lt;&lt; WDE)   (1 &lt;&lt; WDP2)   (1 &lt;&lt; WDPO);     _ enable_interrupt(); }                 </pre>

#### 【使用说明】

在改变WDP配置位之前，建议复位看门狗定时器。因为更改WDP位到比较小的超时周期很可能 会导致看门狗超时复位。

## 6.9. 寄存器定义

低压检测 (LVD) 控制寄存器- LVDCR

LVDCR - LVD 控制寄存器								
LVDCR: 0x62				默认值: 0x00				
Bits	WCE	SWR	-	LVDS2	LVDS1	LVDS0	LVREN	LV DEN
R/W	R/W	W/R	-	R/W	R/W	R/W	R/W	R/W
位定义								

[0]	LV DEN	低压检测模块使能控制, 1 使能, 0 禁止
[1]	LV REN	低压复位功能使能控制, 1 使能, 0 禁止
[4:2]	LV DS	低压检测阈值配置位 000 = 2.2V 001 = 2.4V 010 = 2.6V 011 = 3.0V 100 = 3.6V 101 = 4.0V 其他配置 = 关闭LVD
[5]	-	保留不用
[6]	SW R	软复位使能位, 此位清零将产生软件复位
[7]	W CE	VDT CR 值改变使能位 用户在改变 VDT CR 寄存器的值之前, 必须首先将此位写 1, 在之后的6个时钟周期内, 更改VDT CR其他位的值。四个周 期后WCE 自动清零, 对VDT CR 寄存器的更新操作无效。

## IO 功能复用寄存器- IOCR

## IOCR - IO 功能复用寄存器

IOCR: 0xF0				默认值: 0x00				
Bits	WCE	-	-	TWIPS	-	-	B4EN	C6EN
R/W	R/W	-	-	R/W	-	-	R/W	R/W
位定义								
0	C6EN	C6 引脚默认为复位功能, 设置此位为1 将禁止外部复位功 能, 复位功能禁止后, PC6 可作为一个普通的 I/O 使用						
1	B4EN	B4 引脚默认为模拟输入功能, 设置此位为1, 将关闭模拟输入功能, 这个引脚可以作为 GPIO 使用						
3:2	-	保留不用						
4	TWIPS	TWI 端口选择						
6:5	-	保留不用						
7	WCE	IOCR 值改变使能位 用户在改变IOCR 寄存器的值之前, 必须首先将此位写1, 在之后的6个时钟周期内, 更改IOCR 其他位的值。四个周期 后WCE 自动清零, 对IOCR 寄存器的更新操作无效。						

## MCU 状态寄存器- MCUSR

## MCUSR - IO 特殊功能控制寄存器

MCUSR: 0x34 (0x54)				默认值: 0x00				
Bits	SWDD	-	PDRF	OCDRF	WDRF	BORF	EXTRF	PORF
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	R/W
位定义								
[0]	PORF	上电复位标志, 写0 清零						
[1]	EXTRF	外部复位标志, 上电复位自动清零, 或写0 清零						



[3]	WDE	<p>WDT 使能控制位。</p> <p>当设置WDE 位为“1”时，WDT 被使能。当设置WDE 位为“0”时，WDT 被禁止。</p> <p>只有在 WDTOE 位置位时 WDE 才能被清零。要关闭已经使能了的 WDT，必须按照下列时序操作：</p> <ol style="list-style-type: none"> <li>1. 同时置位WDTOE 和WDE 位，即使WDE 已经被置位，在关闭 操作开始之前也必须对 WDE位写入“1”；</li> <li>2. 在接下来的 4 个时钟周期内，对 WDE 位写入“0”。这将关闭 WDT。</li> </ol> <p>当WDE 位为“1”且WDT 溢出复位系统后会置位WDT 复位系统标志WDRF（位于MCUSR 寄存器）。当WDRF 位处于置位状态时会置 位WDE 位。因此要清零WDE 位，必须先清零WDRF 位。</p>
[2:0]	WDP	<p>WDT 预分频因子选择控制。</p> <p>用来设置WDT 的溢出周期。建议在WDT 未计数时改变WDP 的 值，在计数过程中改变WDP 的值就会产生不可预期的WDT 溢出。</p>

看门狗预分频选择列表

WDP3	WDP2	WDP1	WDP0	看门狗定时器 溢出周 期数	32KHz 时钟
0	0	0	0	2K cycles	64ms
0	0	0	1	4K cycles	128ms
0	0	1	0	8K cycles	256ms
0	0	1	1	16K cycles	512ms
0	1	0	0	32K cycles	1s
0	1	0	1	64K cycles	2s
0	1	1	0	128K cycles	4s
0	1	1	1	256K cycles	8s
1	0	0	0	512K cycles	16s
1	0	0	1	1024K cycles	32s
其他				与64K Cycles 配置相同	

## 7. 中断与中断向量

- 内部/外部多路丰富中断源
- 可编程向量起始地址
- 自动中断标志位清除

PDS8F208A 的复位向量从地址 0x0000 开始执行。除复位向量外，其他向量地址都可以通过MCUCR 寄存器中的IVSEL 以及 IVBASE 寄存器重新定向到512 字节对齐的起始地址。

### 7.1. PDS8F208A 中断向量列表

编号	向量地址	中断源信号	中断源说明
----	------	-------	-------

1	0x0000	RESET	外部复位，上电复位，看门狗复位，SWD 调试复位，低电压复位
2	0x0002	INT0	外部中断请求0
3	0x0004	PCI2	引脚电平中断2
4	0x0006	PCI0	引脚电平中断0
5	0x0008	PCI1	引脚电平中断1
6	0x000A	PDC_RXIR	PD 控制器数据接收中断
7	0x000C	WDT	看门狗溢出中断
8	0x000E	ECP21R   DCM211R   DCM221R   DCM231R	ECP2 匹配，俘获，PWM 自动关闭中断
9	0x0010	PDC_TXIR	PD 控制器数据发送中断
10	0x0012	T2IR	定时器2 计数器溢出中断
11	0x0014	ECP11R   DCM111R   DCM121R   DCM131R	ECP1 匹配，俘获，PWM 自动关闭中断
12	0x0016	PCI3	引脚电平变化中断 3
13	0x0018	PDC1_RXIR	PD 控制器 1 数据接收中断
14	0x001A	T1IR	定时器1 计数溢出中断
15	0x001C	PDC1_TXIR	PD 控制器 1 数据发送中断
16	0x001E	HTCIR	华为通信接口中断
17	0x0020	T0IR	定时器0 计数溢出中断
18	0x0022	PDC_TPIR	PD 控制器中断
19	0x0024	UART0_RXC	UART0 接收结束中断
20	0x0026	UART0_UDRE	UART0 数据寄存器空中断
21	0x0028	UART0_TXC	UART0 发送结束中断
22	0x002A	ADC	ADC 转换结束中断
23	0x002C	EEP	EEP 控制器中断
24	0x002E	ACO	模拟比较器 0 中断
25	0x0030	TWI	TWI 中断
26	0x0032	AC1	模拟比较器 1 中断
27	0x0034	PDC1_TPIR	PD 控制器 1 中断
28	0x0036	SPI	SPI 中断
29	0x0038	UART1_RXC	UART1 接收结束中断
30	0x003A	UART1_UDRE	UART1 数据寄存器空中断
31	0x003C	UART1_TXC	UART1 发送结束中断

## 7.2. 外部中断源 INT0

外部中断由INT0 引脚触发。只要外部中断被使能，即使这个引脚配置为输出也能触发 中断。这可以用来产生软件中断。外部中断可以由上升沿，下降沿或低电平触发，由外部中断控制寄存器EICR来配置。当外部中断使能并且配置为电平触发（只有INT0 引脚）时，只 要引脚电平为低，中断就会一直产生。INT0 引脚的上升沿或下降沿中断触发需要10 时钟正 常工作，而INT0 引脚的低电平触发中断都是异步检测的。除了空闲模式，其它睡眠模式下 10 时钟都是停止工作的。因此，这 2 个外部中断都可用作除空闲模式外的其它睡眠模式下的唤醒源。

若电平触发中断用作省电模式下的唤醒源，改变的电平必须保持一定的时间来唤醒 MCU， 以降低 MCU 对噪声的敏感程度。要求的电平必须保持足够长的时间使 MCU 结束唤醒过程， 然后触发电平中断。

## 7.3. 引脚电平变化中断

引脚电平变化中断由 PBn, PCn, PDn, PEn 和 PFn 引脚触发。只要引脚电平变化中断被使能, 即使这些引脚配置为输出也能触发中断。这可以用来产生软件中断。

任何一个使能的 PBn 引脚翻转都会触发引脚电平中断 PCI0, 使能的 PCn 引脚翻转将触发 PCI1, 使能的 PDn 引脚翻转将触发 PCI2, 使能的 PEn 引脚翻转将触发 PCI3。各个引脚变化 中断的使能分别由 PCMSK0~4 寄存器来控制。所有的引脚电平变化中断都是异步检测的, 可 用作某些睡眠模式下的唤醒源。

#### 7.4. 寄存器定义

MCU 控制寄存器- MCUCR								
MCUCR - MCU 控制寄存器								
MCUCR: 0x35 (0x55)					默认值: 0x00			
MCUCR	FWKEN	FPDEN	EXRFD	PUD	IRLD	IFAIL	IVSEL	WCE
R/W	R/W	R/W	R/W	R/W	W/O	R/O	R/W	R/W
位定义								
[0]	WCE	MCUCR 更新使能位, 在更新MCUCR 之前, 需要首先设置此位, 然后在6 个周期内完成对MCUCR 寄存器的更新						
[1]	IVSEL	中断向量选择位, 此位置 1 后, 中断向量地址将根据 IVBASE 寄存器的值映射到新的地址						
[2]	IFAIL	系统配置位加载失败标志位, 0 = 配置信息校验通过 1 = 配置信息加载失败						
[3]	IRLD	写1 将重新加载系统配置信息						
[4]	PUD	全局上拉禁止位 0 = 使能全局上拉控制 1 = 关闭所有I0 的上拉电阻						
[5]	EXRFD	外部复位滤波禁止位 0 = 使能外部复位的(190us)数字滤波器 1 = 禁用外部复位的数字滤波电路						
[6]	FPDEN	Flash Power/down 使能控制 0: 系统SLEEP 后FLASH 保持上电状态 1: 系统SLEEP 后FLASH 断电						
[7]	FWKEN	快速唤醒模式使能控制, 仅对Power/Off 模式有效 0: 260us 滤波延时 1: 32us 滤波延时						

中断向量基地址寄存器 - IVBASE	
IVBASE - 中断向量基地址寄存器	
IVBASE: 0x75	
默认值: 0x00	
IVBASE	IVBASE [7:0]
R/W	R/W
位定义	

[7:0]	IVBASE	如果 IVSEL 为 1, 中断向量(复位向量除外)将以 IVBASE 为基址 址在 512 字节的页面上重新映射。映射后的中断向量基址为: (IVBASE << 8) + 表 1 中对应的 向量地址
-------	--------	--

**外部中断控制寄存器 - EICR**

EICR - 外部中断控制寄存器								
地址: 0x3C				默认值: 0x00				
Bit	7	6	5	4	3	2	1	0
Name	ISC01	ISC00	-	EINTE	-	-	-	EINTF
R/W	-	-	-	-	-	-	-	R/W
Bit	Name	描述						
7	ISC01	INT0 引脚中断触发方式控制位高位。						
6	ISC00	INT0 引脚中断触发方式控制位低位。 当全局中断置位且 GICR 寄存器的相应中断屏蔽控制位被置位时, 外部中断 0 由 INT0 引脚激发。中断的触发方式见表描述。在边沿检测之前 MCU 首先采样 INT0 引脚上的电平。如果选用了边沿触发方式或电平变化触发方式, 那么持续时间大于 1 个系统时钟周期的脉冲将触发中断, 过短的脉冲则不能保证触发中断。如果选择低电平触发方式, 那么低电平必须保持到当前指令执行完成才会触发中断。						
5	-	保留未用						
4	EINTE	外部中断使能位, 写 1 使能外部中断, 写 0 禁止外部中断						
3:1	-	保留未用						
0	EINTF	外部中断标志位, 当 EINTE/EINTF 同时置位后, 外部中断源有效, 对						

**外部中断 0 触发方式控制**

ISC0[1:0]	描述
0	外部引脚 INT0 低电平触发
1	外部引脚 INT0 上升沿或下降沿触发
2	外部引脚 INT0 下降沿触发
3	外部引脚 INT0 上升沿触发

**PCIFR - 引脚改变中断标志寄存器**

PCIFR - 引脚改变中断标志寄存器								
地址: 0x3B				默认值: 0x00				
Bit	7	6	5	4	3	2	1	0
	PCIE3	PCIE2	PCIE1	PCIE0	PCIF3	PCIF2	PCIF1	PCIF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	PCIE3	引脚改变 3 中断使能控制位						
6	PCIE2	引脚改变 2 中断使能控制位 当设置 PCIE2 位为 “1” 且全局中断使能时, 引脚改变 2 中断被使能。任何一个使能的 PDn 引脚的电平变化都能产生 PCIE2 中断。PDn 引脚中断的使能可分别由 PCMSK2 寄存器来控制。 当设置 PCIE2 位为 “0” 时, 引脚改变 2 中断被禁止。						

5	PCIE1	<p>引脚改变1 中断使能控制位</p> <p>当设置PCIE1 位为“1”且全局中断使能时，引脚改变1 中断被使能。任何一个使能的PCn 引脚的电平变化都会产生PCIE1 中断。PCn 引脚中断的使能可分别由PCMSK1 寄存器来控制。</p> <p>当设置PCIE1 位为“0”时，引脚改变1 中断被禁止。</p>
4	PCIE0	<p>引脚改变0 中断使能控制位</p> <p>当设置PCIE0 位为“1”且全局中断使能时，引脚改变0 中断被使能。任何一个使能的PBn 引脚的电平变化都会产生PCIE0 中断。PBn 引脚中断的使能可分别由PCMSK0 寄存器来控制。</p> <p>当设置PCIE0 位为“0”时，引脚改变0 中断被禁止。</p>
3	PCIF3	引脚改变3 中断标志位
2	PCIF2	引脚改变2 中断标志位
1	PCIF1	<p>引脚改变1 中断标志位</p> <p>任何一个使能的PCn 引脚的电平变化都会置位PCIF1。当PCIE1 和全局中断均置位时，MCU 将会跳转至 PCIE1 中断入口地址。PCn 引脚中断的使能可分别由PCMSK1 寄存器来控制。执行中断服务程序或往PCIF1 位写“1”都会清零PCIF1 位。</p>
0	PCIF0	<p>引脚改变0 中断标志位</p> <p>任何一个使能的PBn 引脚的电平变化都会置位PCIF0。当PCIE0 和全局中断均置位时，MCU 将会跳转至 PCIE0 中断入口地址。PBn 引脚中断的使能可分别由PCMSK0 寄存器来控制。执行中断服务程序或往PCIF0 位写“1”都会清零PCIF0 位。</p>

PCMSK0 - 引脚改变中断屏蔽寄存器 0

PCMSK0 - 引脚改变屏蔽寄存器0								
地址: 0x6B					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	-	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	-	无						
6	PCINT6	引脚改变使能屏蔽位6。						
5	PCINT5	引脚改变使能屏蔽位5。						
3	PCINT3	<p>引脚改变使能屏蔽位3。</p> <p>当设置 PCINT3 位为“1”时，PB3 引脚电平改变中断被使能。PB3 引脚上的电平改变将置位 PCIF0，若 PCIE0 位和全局中断置位，将会产生PCIE0 中断。当设置 PCINT3 位为“0”时，PB3 引脚电平改变中断被禁止。</p>						
2	PCINT2	<p>引脚改变使能屏蔽位2。</p> <p>当设置 PCINT2 位为“1”时，PB2 引脚电平改变中断被使能。PB2 引脚上的电平改变将置位 PCIF0，若 PCIE0 位和全局中断置位，将会产生PCIE0 中断。当设置 PCINT2 位为“0”时，PB2 引脚电平改变中断被禁止。</p>						
1	PCINT1	<p>引脚改变使能屏蔽位1。</p> <p>当设置 PCINT1 位为“1”时，PB1 引脚电平改变中断被使能。PB1 引脚上的电平改变将置位 PCIF0，若 PCIE0 位和全局中断置位，将会产生PCIE0 中断。当设置 PCINT1 位为“0”时，PB1 引脚电平改变中断被禁止。</p>						

0	PCINT0	引脚改变使能屏蔽位0。 当设置 PCINT0 位为“1”时，PB0 引脚电平改变中断被使能。PB0 引脚上的电平改变将置位 PCIF0，若 PCIE0 位和全局中断置位，将会产生 PCIO 中断。当设置 PCINT0 位为“0”时，PB0 引脚电平改变中断被禁止。
---	--------	---

**PCMSK1 - 引脚改变中断屏蔽寄存器 1**

PCMSK1 - 引脚改变屏蔽寄存器1								
地址: 0x6C					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	-	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	-	无						
6	PCINT14	引脚改变使能屏蔽位14。 当设置 PCINT14 位为“1”时，PC6 引脚电平改变中断被使能。PC6 引脚上的电平改变将置位 PCIF1，若 PCIE1 位和全局中断置位，将会产生 PCIO 中断。当设置 PCINT14 位为“0”时，PC6 引脚电平改变中断被禁止。						
5	PCINT13	引脚改变使能屏蔽位13。 当设置 PCINT13 位为“1”时，PC5 引脚电平改变中断被使能。PC5 引脚上的电平改变将置位 PCIF1，若 PCIE1 位和全局中断置位，将会产生 PCIO 中断。当设置 PCINT13 位为“0”时，PC5 引脚电平改变中断被禁止。						
4	PCINT12	引脚改变使能屏蔽位12。 当设置 PCINT12 位为“1”时，PC4 引脚电平改变中断被使能。PC4 引脚上的电平改变将置位 PCIF1，若 PCIE1 位和全局中断置位，将会产生 PCIO 中断。当设置 PCINT12 位为“0”时，PC4 引脚电平改变中断被禁止。						
3	PCINT11	引脚改变使能屏蔽位11。 当设置 PCINT11 位为“1”时，PC3 引脚电平改变中断被使能。PC3 引脚上的电平改变将置位 PCIF1，若 PCIE1 位和全局中断置位，将会产生 PCIO 中断。当设置 PCINT11 位为“0”时，PC3 引脚电平改变中断被禁止。						
2	PCINT10	引脚改变使能屏蔽位2。 当设置 PCINT10 位为“1”时，PC2 引脚电平改变中断被使能。PC2 引脚上的电平改变将置位 PCIF1，若 PCIE1 位和全局中断置位，将会产生 PCIO 中断。当设置 PCINT10 位为“0”时，PC2 引脚电平改变中断被禁止。						
1	PCINT9	引脚改变使能屏蔽位1。 当设置 PCINT9 位为“1”时，PC1 引脚电平改变中断被使能。PC1 引脚上的电平改变将置位 PCIF1，若 PCIE1 位和全局中断置位，将会产生 PCIO 中断。当设置 PCINT9 位为“0”时，PC1 引脚电平改变中断被禁止。						
0	PCINT8	引脚改变使能屏蔽位0。 当设置 PCINT8 位为“1”时，PC0 引脚电平改变中断被使能。PC0 引脚上的电平改变将置位 PCIF1，若 PCIE1 位和全局中断置位，将会产生 PCIO 中断。当设置 PCINT8 位为“0”时，PC0 引脚电平改变中断被禁止。						

**PCMSK2 - 引脚改变中断屏蔽寄存器 2**

PCMSK2 - 引脚改变屏蔽寄存器0								
地址: 0x6D					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16

R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	PCINT23	引脚跳变中断使能位23 当设置 PCINT23 位为“1”时，PD7 引脚电平改变中断被使能，PD7 引脚上的电平改变将置位 PCIF2，若 PCIE2 位和全局中断置位，将会产生 PC12 中断。当设置 PCINT23 位为“0”时，引脚电平改变中断被禁止。						
6	PCINT22	引脚跳变中断使能位22 当设置 PCINT22 位为“1”，PD6 引脚电平改变中断被使能。PD6 引脚上的电平改变将置位 PCIF2，若 PCIE2 位和全局中断置位，将会产生 PC12 中断。当设置 PCINT22 位为“0”，PD6 引脚电平改变中断被禁止						
5	PCINT21	引脚跳变中断使能位21 当设置 PCINT21 位为“1”时，PD5 引脚电平改变中断被使能。PD5 引脚上的电平改变将置位 PCIF2，若 PCIE2 位和全局中断置位，将会产生 PC12 中断。当设置 PCINT21 位为“0”时，PD5 引脚电平改变中断被禁止。						
4	PCINT20	引脚跳变中断使能位20 当设置 PCINT20 位为“1”时，PD4 引脚电平改变中断被使能。PD4 引脚上的电平改变将置位 PCIF2，若 PCIE2 位和全局中断置位，将会产生 PC12 中断。当设置 PCINT20 位为“0”时，PD4 引脚电平改变中断被禁止。						
3	PCINT19	引脚跳变中断使能位19 当设置 PCINT19 位为“1”时，PD3 引脚电平改变中断被使能。PD3 引脚上的电平改变将置位 PCIF2，若 PCIE2 位和全局中断置位，将会产生 PC12 中断。当设置 PCINT19 位为“0”时，PD3 引脚电平改变中断被禁止。						
2	PCINT18	引脚跳变中断使能位18 当设置 PCINT18 位为“1”时，PD2 引脚电平改变中断被使能。PD2 引脚上的电平改变将置位 PCIF2，若 PCIE2 位和全局中断置位，将会产生 PC12 中断。当设置 PCINT18 位为“0”时，PD2 引脚电平改变中断被禁止。						
1	PCINT17	引脚跳变中断使能位17 当设置 PCINT17 位为“1”时，PD1 引脚电平改变中断被使能。PD1 引脚上的电平改变将置位 PCIF2，若 PCIE2 位和全局中断置位，将会产生 PC12 中断。当设置 PCINT17 位为“0”时，PD1 引脚电平改变中断被禁止。						
0	PCINT16	引脚跳变中断使能位16 当设置 PCINT16 位为“1”时，PD0 引脚电平改变中断被使能。PD0 引脚上的电平改变将置位 PCIF2，若 PCIE2 位和全局中断置位，将会产生 PC12 中断。当设置 PCINT16 位为“0”时，PD0 引脚电平改变中断被禁止。						

## 8. 通用可编程端口 (GPIO)

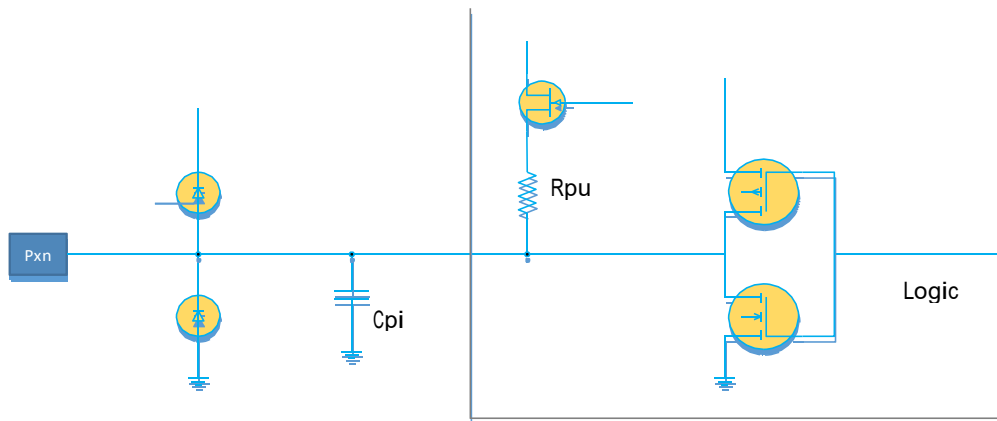
### 8.1. 概述

所有基于 LGT8XM 内核系列实现的 MCU 都具有 I/O 端口读-改-写功能。这意味着，某一个端口的状态可以使用 SBI 和 CBI 指令单独的改变，而不会影响到其他任何 I/O。同样，改变一个端口的方向或者控制它的上拉电阻也可以如此。

PDS8F208A 的大部分 I/O 拥有对称的驱动特性，能够驱动和吸收较大的电流。I/O 具有两级驱动能力，用户可以控制每组 I/O 的驱动能力。I/O 的驱动能力可以直接驱动一些 LED。

PDS8F208A 的大部分 I/O 可以驱动高达 25mA 的电流，可直接用于驱动段码 LED。所有的 I/O 的 VCC 和 GND 直接都有独立的 ESD 保护二极管，设计至少可以承受高达 4000V 的 ESD 脉冲。

I/O 等效电路图



本章下面所有寄存器采用统一描述方式，小写的“x”表示端口的字母序号名，小写的“n”表示端口中的位号。但当在程序中使用端口寄存器时，必须使用准确的寄存器名字。比如 PORTB3，它表示PORTB 的第三位，这里则统一用PORTxn 表示。I/O 相关寄存器的详细定义，请参考寄存器描述部分。

每个端口分配有三个 I/O 寄存器空间，它们为：端口数据输出寄存器 (PORTx)，端口方向寄存器 (DDRx)，端口数据输入寄存器 (PINx)。端口数据输入寄存器为只读寄存器。数据输出寄存器与端口方向寄存器可读也可以改写。MCUCR 寄存器中的 PUD 位，用于控制所有 I/O 的上拉电阻，当 PUD 位为 1 时，将禁止所有 I/O 的上拉电阻。

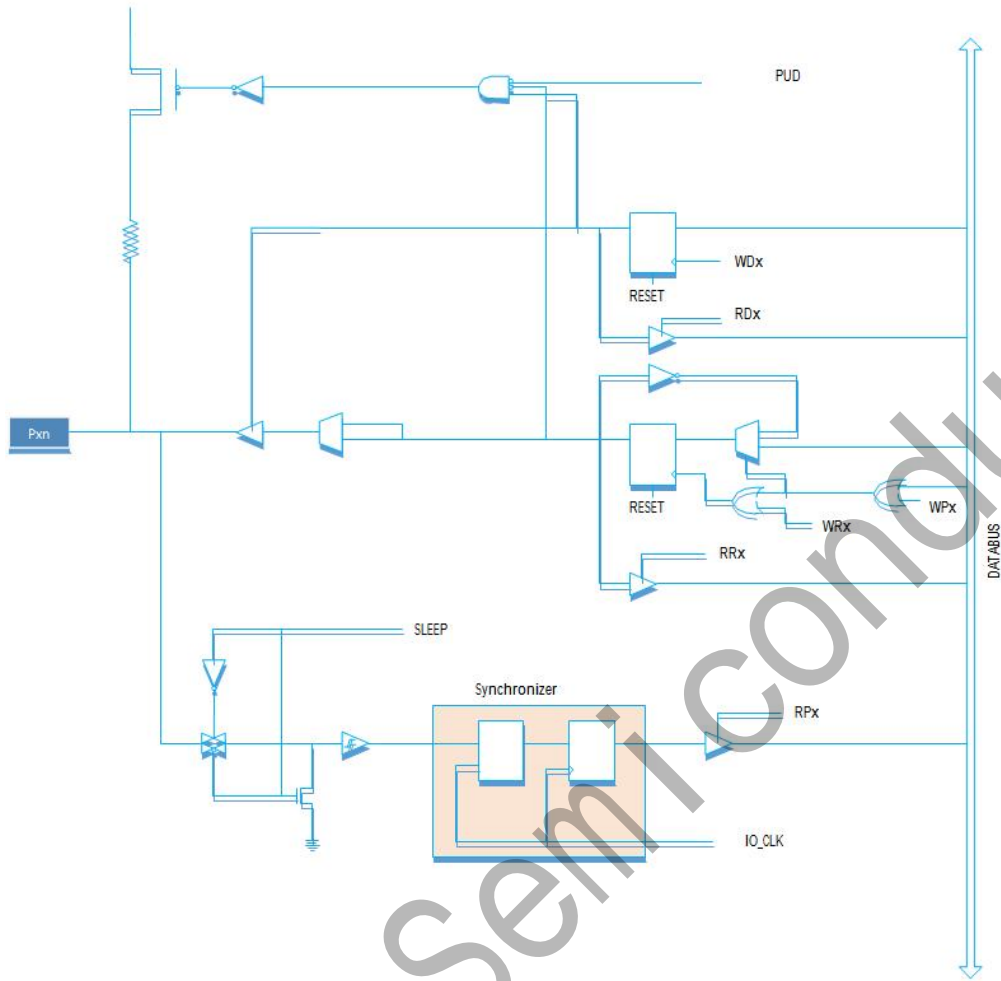
大部分 I/O 除了具有通用输入/输出功能，也会被复用为其他外设功能。具体的复用功能请参考关于端口功能复用的章节。

需要注意的是，使能某些端口的复用功能并不会影响这些端口作为数字 I/O 使用。而且某些复用功能也可能需要通过 I/O 寄存器控制端口的输入/输出方向。具体的设置将会在各个复用模块的文档的介绍。

## 8.2. 通用输入/输出端口

作为通用 I/O 时，端口为双向驱动 I/O 端口，内部可编程上拉。下图为通用 I/O 端口的等效电路图：

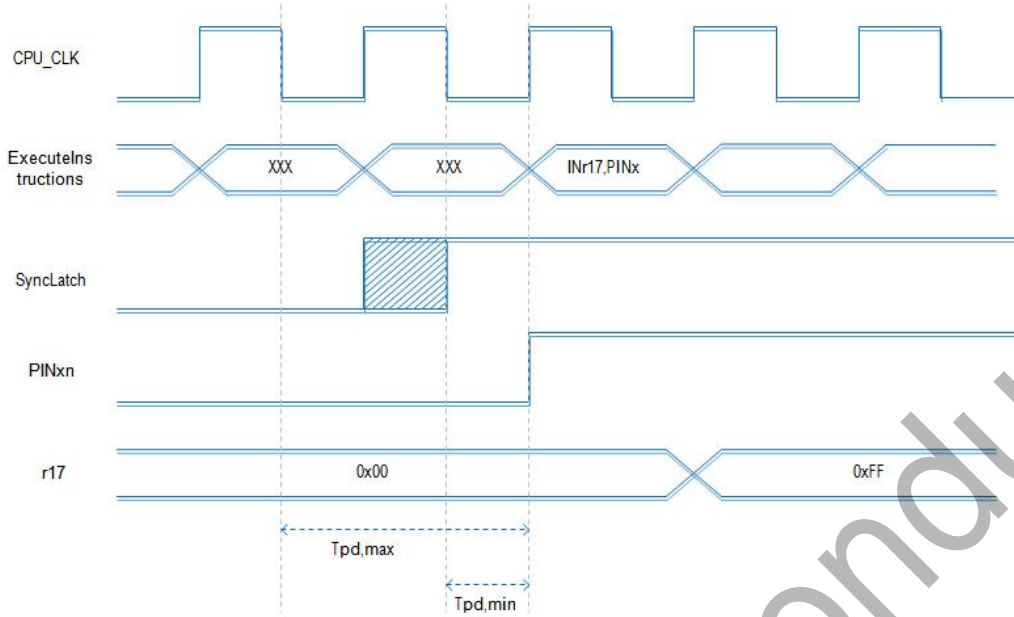
PUD: PULLUPDISABLE	WDx: WRITEDDRx
SLEEP: SLEEPCONTROL	RDx: READ DDRx
IO_CLK: I/OCLK	WRx: WRITEPORTx
	RRx: READPORTxREGISTER
	RPx: READ PORTxPIN
	WPx: WRITEPINxREGISTER



### 8.3. 端口使用配置

每个端口由三个寄存器位控制: DDxn, PORTxn 和 PINxn。其中 DDxn 用于可以通过 DDRx 寄存器访问, PORTxn 可以通过 PORTx 寄存器访问, PINxn 可以通过 PINx 寄存器访问。

DDxn 寄存器位用于设置端口的输入/输出方向。如果 DDxn 设置为 1, Pxn 端口就被配置为一个输出端口。如果 DDxn 设置为 0, Pxn 就被配置为一个输入端口。



如果 PORTxn 位被写 1，同时这个端口被配置为输入端口，这个端口的上拉电阻有效。如果想要禁止端口的上拉电阻，PORTxn 必须写为 0 或者将这个端口配置为输出端口。

端口的复位初始化状态为输入状态，上拉电阻无效。

PORTxn 设置为 1，同时这个端口被配置为输出端口，外部端口将会被驱动为高电平。如果 PORTxn 设置为 0，端口将会被驱动为低。

### 8.4. 输入/输出切换

当 I/O 状态在三态 ([DDxn, PORTxn] = 0b00) 和输出高电平 ([DDxn, PORTxn] = 0b11) 之间切换时，将会出现一个端口上拉或者输出为低的中间状态。通常，上拉电阻是可以被接受的，因为在一个高阻环境下，驱动为高和上拉之间的区别并不重要。如果不是这种情况，可以通过 MCUCR 寄存器中的 PUD 位关闭所以端口的上拉功能。

同样，在上拉使能的输入与输出低电平之间切换时，也会出现同样的问题。用户必须使用三态 ([DDxn, PORTxn] = 0b00) 或者输出高 ([DDxn, PORTxn] = 0b11) 作为中间状态。

端口驱动配置表：

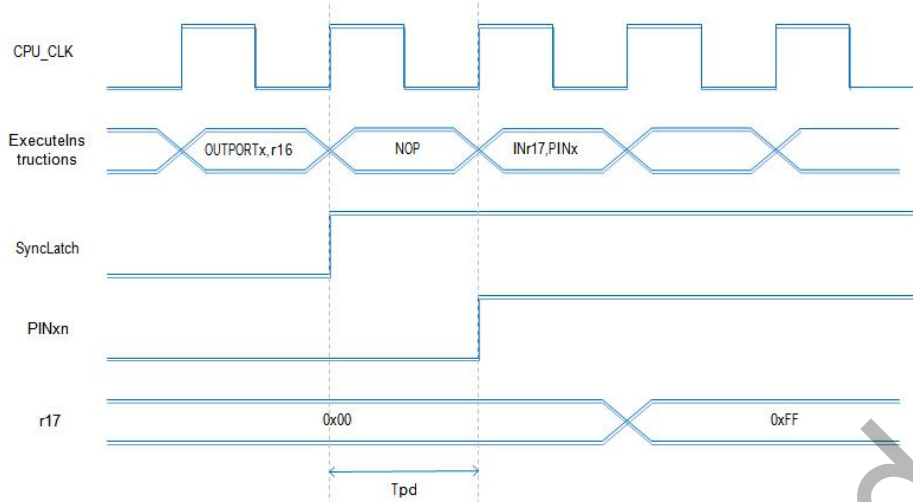
DDxn	PORTxn	PUD	端口状态	上拉	功能说明
0	0	X	输入	禁止	三态 (High-Z)
0	1	0	输入	使能	输入+内部上拉模式
0	1	1	输入	禁止	三态 (High-Z)
1	0	X	输出	禁止	输出低 (扇入)
1	1	X	输出	禁止	输出高 (扇出)

### 8.5. 读端口值

无论端口方向位 DDxn 如何设置，都可以通过 PINxn 寄存器位读取到端口的当前状态。为避免直接读取端口产生的亚稳态，PINxn 寄存器位是端口经过一个同步器的结果。同步器为一个锁存器和一个寄存器共同组成，因此 PINxn 的值与当前端口之间有一个很小的延迟。这个延迟是因为同步器存在的结果，延迟时间最多为 1 个半系统周期。

我们假设系统周期从系统时钟的第一个下降沿开始，锁存器在时钟为低的时候锁存数据，时钟为高时数据直通过锁存器，如上图阴影部分所示。在时钟为低电平时，端口数据被锁存，并且在下一个时钟的上升沿被寄存器到 PINxn 寄存器。上图中的 Tpd,max 以及 Tpd,min 为端口数据的最大和最小延迟，分为为 1.5 周期和 0.5 周期。

如果要读取到软件设置的端口值，需要在 I/O 的写和读字节支持插入一个空操作指令 (NOP)。时序如下图所示：



下面的代码说明如何设置端口 B 的引脚 0/1 为高，2/3 为低，定义引脚 4~7 为输入并且使能了引脚 6、7 的上拉电阻。然后引脚的值回读到通用工作寄存器中，按照之前的描述，在引脚的输出和输入直接插入了一个 NOP 指令。

#### 汇编代码

```
; Define Pull-ups and set outputs high
; Define directions for port pins
LDI r16, (1<<PB7) | (1<<PB6) | (1<<PB1) | (1<<PB0)
LDI r17, (1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0)
OUT PORTB, r16
OUT DDRB, r17
; Insert nop for synchronization
NOP
; Read port pins
IN r16, PINB
```

#### C 语言代码

```
unsigned char I;
/* Define pull-ups and set outputs high */
/* Define directions for port pins */
PORTB = (1<<PB7) | (1<<PB6) | (1<<PB1) | (1<<PB0); DDRB =
(1<<DDB3) | (1<<DDB2) | (1<<DDB1) | (1<<DDB0);
/* Insert nop for synchronization */
no_operation();
/* Read port pins */
I = PINB;
```

## 8.6. 输入使能与休眠控制

从 I/O 的等效电路图中我们可以看到，数字输入可以在 SLEEP 信号的控制下被钳位到地电平。SLEEP 信号由 MCU 的休眠控制器以及各种休眠模式控制。这样可以保证在进入休眠后，系统不会因为端口输入浮空而造成漏电。

端口的 SLEEP 控制作用会被外部中断功能取代。如果外部中断请求无效，SLEEP 控制仍然可以起作用。SLEEP 控制功能也会被其他一些第二功能取代，具体请参考下面关于端口第二功能的介绍。

## 8.7. 快速翻转端口状态

端口状态设置为输出的 I/O，可以通过 PORTn 寄存器改变端口状态。如果需要翻转当前端口的输出状态，通常需要首先读取当前端口状态 PINx，然后取反回写到 PORTn 寄存器完成翻转。PDS8F208A 提供另外一种更加高效的方式翻转端口状态，通过直接向 PINx 寄存器写 1 即可实现将指定的端口状态翻转。比如我们写 PINB[3] 为 1，即可实现将 PB3 的端口状态翻转。对于需要产生输出时钟的应用中，这种方式非常的实用。

### 8.8. 数字/模拟复用端口

PDS8F208A 部分端口为数字/模拟混合复用端口。除内部 DAC 的输出 PD4 外，其他混合端口的均作为模拟输入用。当端口作为模拟功能使用时，软件需要将该端口设置为输入模式，并根据需要关闭内部上拉，以免对模拟收入产生影响。DIDR0~3 寄存器用于关闭混合功能端口的数字输入通道，以避免模拟输入对数字电路造成多余功耗损失。DIDRx 不会关闭端口的数字输出功能。

### 8.9. 大电流推挽驱动端口

PDS8F208A 的 PC2/3/4/5 共四路大电流推挽驱动端口，支持最大 80mA 的推挽驱动。

PC2/3/4/5 内部与 EEC1/2 的 PWM 输出复用，可实现基于半桥/全桥的电源应用电路。

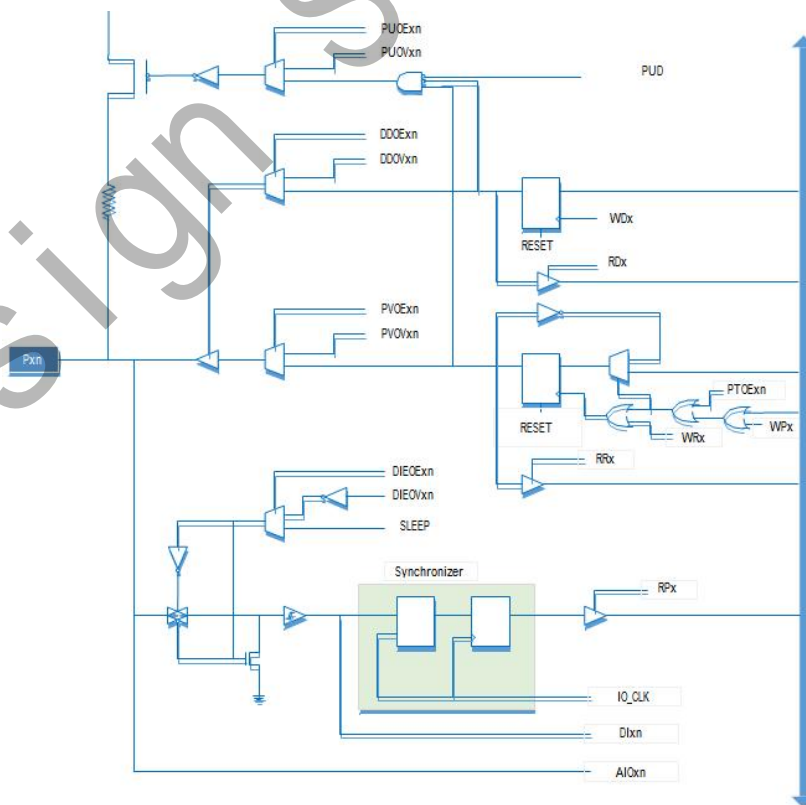
### 8.10. 空闲端口的处理

如果一些端口没有被使用，建议将他们驱动到一个固定的电平。在任何情况下，浮空的引脚都会带来更多的功耗，并且会导致系统在强干扰下变的不稳定。

给端口一个固定电平最简单的方法就是打开端口的上拉电阻。需要注意的是，上拉电阻在上电复位过程中是禁止的。上拉电阻的方式也会带来多余的漏电。因此建议使用外部的上拉或者下拉电阻连接。直接将端口与电源或地连接是不建议的，因为如果这些引脚被配置为输出，会有可能导致非常大的电流由端口经过，对芯片造成破坏性的影响。

### 8.11. 端口复用功能

大部分端口都有复用功能，下面的等效电路说明了端口复用功能对端口的控制。这些复用功能并不一定存在与所有的端口引脚。



PUOExn: Pxn PULL-UPOVERRIDEENABLE	PUD:
PULLUPDISABLEPUOVxn:	Pxn PULL-
UPOVERRIDEVALUE	WDx:
WRITEDDRxDDOExn:	Pxn
DATADIRECTIONOVERRIDEENABLE	RDx: READDRx
DDOVxn: Pxn DATADIRECTIONOVERRIDEVALUE	RRx:
READPORTxREGISTERPVOExn:	Pxn PORTVALUEOVERRIDEENABLE
WRx:	WRITEPORTx
PVOVxn: Pxn PORTVALUEOVERRIDEVALUE	RPx:
READPORTxPINDIEOExn: Pxn INPUT-ENABLEOVERRIDEENABLE	WPx:
WRITEPINxDIEOVxn: Pxn INPUT-ENABLEOVERRIDEVALUE	IO_CLK:
I/OCLOCK	
SLEEP: SLEEPCONTROL	Dlxn:
INPUTPINnONPORTxPTOExn:	Pxn PORTTOGGLEOVERRIDEENABLE
AIOxn:	ANALOGI/OPINnONPORTx

复用功能控制信号一般描述:

信号	全称	功能描述
PUOE	上拉复用使能	此位为1, 上拉使能由PVOV控制; 如果此位为 0, 上拉使能受 DDxn, PORTxn 以及PUD 共同控制
PUOV	上拉复用值	如果PUOE为1, 此位为1 将使能引脚的上拉电阻, 否则将禁止引脚上拉电阻
DDOE	端口方向复用使能	次位为 1, 引脚输出使能由 DDOE 控制, 否则由 DDxn 控制
DDOV	端口方向复用值	如果DDOE 为1, 次位为1, 将使能引脚的输出功能, 否则关闭引脚的输出
PVOE	端口数据复用使能	如果次位为 1, 并且引脚输出使能, 引脚的输出值将由PVOV 控制, 否则是由PORTxn 控制
PVOV	端口数据复用值	参考PVOE 功能描述
PTOE	端口翻转复用使能	次位为1, PORTxn 位将翻转
DIEOE	数字输入使能复用使能	如果次位为1, 端口数字输入使能将由DIEOV 控制; 否则将有 MCU 的运行状态控制
DIEOV	数字输入使能复用值	如果 DIEOE 为 1, 端口的数字输入功能将由次位控制, 与 MCU 运行状态无关
DI	数字输入	这个是输入给替代功能模块的数字输入信号。从 I/O 等下电路图中可以看到, 这个值在施密特触发器之后, 但在 I/O 输入同步器之前。这个信号连接到外设模块中, 外设模块将会根据需要进行同步处理

A10	模拟输入	模拟输入/输出信号，这个信号直接与 I/O 的 PAD 相连，可作为模拟的双向信号使用。这个信号直接与内部的 ADC、比较器等模拟模块的端口相连接
-----	------	---

## 8.12. 寄存器定义

端口 B 输出数据寄存器- PORTB

PORTB - 端口B 输出数据寄存器

PORTB: 0x05 (0x25)				默认值: 0x00				
Bits	-	PB6	PB5	PB4	PB3	PB2	PB1	PB0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位定义								
[6:0]	PORTB	B 组端口输出寄存器						

端口 B 方向寄存器- DDRB

DDRB - 端口B 方向寄存器

DDRB: 0x04 (0x24)				默认值: 0x00				
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位定义								
[6:0]	DDB	端口B 组方向控制位: 1 = 输出, 0 = 输入						

端口 B 输入数据寄存器- PINB

PINB - 端口B 输入数据寄存器

PINB: 0x03 (0x23)				默认值: 0x00				
PINB	-	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位定义								
[6:0]	PINB	B 组端口状态寄存器。读PINB 直接获得端口的当前状态; 写PINBn 位1 将翻转PORTBn 的输出状态						

端口 C 输出数据寄存器- PORTC

PORTC - 端口C 输出数据寄存器

PORTC: 0x08 (0x28)				默认值: 0x00				
PORTC	-	PC6	PC5	PC4	PC3	PC2	PC1	PC0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位定义								
[6:0]	PORTC	C 组端口输出寄存器						

端口 C 方向寄存器- DDRC

DDRC - 端口C 方向寄存器

DDRC: 0x07 (0x27)				默认值: 0x00				
DDRC	-	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位定义								

[6:0]	DDC	C 组端口方向控制位; 1 = 输出, 0 = 输入
-------	-----	----------------------------

**端口 C 输入数据寄存器- PINC**
**PINC - 端口C输入数据寄存器**

PINB: 0x06 (0x26)				默认值: 0x00				
PINC	-	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**位定义**

[6:0]	PINC	C组端口状态寄存器; 读PINC得到当前端口状态 写 PINC将翻转当前端口输出
-------	------	--

**端口 D 输出数据寄存器- PORTD**
**PORTD - 端口D输出数据寄存器**

PORTD: 0x0B (0x2B)				默认值: 0x00				
Bits	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**位定义**

[7:0]	PORTD	D组端口输出寄存器
-------	-------	-----------

**端口 D 方向寄存器- DDRD**
**DDRD - 端口D方向寄存器**

DDRD: 0x0A (0x2A)				默认值: 0x00				
Bits	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**位定义**

[7:0]	DDD	端口D组方向控制位; 1 = 输出, 0 = 输入
-------	-----	---------------------------

**端口 D 输入数据寄存器- PIND**
**PIND - 端口D输入数据寄存器**

PIND: 0x09 (0x29)				默认值: 0x00				
PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

**位定义**

[6:0]	PIND	B组端口状态寄存器。读PIND直接获得端口的当前状态; 写PINDn 位1 将翻转PORTDn 的输出状态
-------	------	---

**端口 E 输出数据寄存器- PORTE**
**PORTE - 端口E输出数据寄存器**

PORTE: 0x0E (0x2E)				默认值: 0x00				
Bits	-	-	PE5	PE4	PE3	PE2	PE1	PE0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W

**位定义**

[5:0]	PORTE	E 组端口输出寄存器
-------	-------	------------

**端口 E 方向寄存器- DDRE**



DDRE - 端口方向寄存器									
DDRE: 0x0D (0x2D)					默认值: 0x00				
Bits	-	-	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位定义									
[5:0]	DDE	端口E组方向控制位; 1 = 输出, 0 = 输入							

端口 E 输入数据寄存器- PINE									
PINE - 端口 E 输入数据寄存器									
PINE: 0x0C (0x2C)					默认值: 0x00				
PINE	-	-	PINE5	PINE4	PINE3	PINE2	PINE1	PINE0	
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位定义									
[5:0]	PINE	E 组端口状态寄存器。读PINE 直接获得端口的当前状态; 写PINEn 位1 将翻转PORTEn 的输出状态							

端口复用控制寄存器 - IOCR									
IOCR - IO 功能复用寄存器									
IOCR: 0xF0					默认值: 0x00				
Bits	WCE	-	-	TWIPS	-	-	B4EN	C6EN	
R/W	R/W	-	-	R/W	-	-	R/W	R/W	
位定义									
0	C6EN	C6 引脚默认为复位功能, 设置此位为1 将禁止外部复位功 能, 复位功能禁止后, PC6 可作为一个普通的 I/O 使用							
1	B4EN	B4 引脚默认为模拟输入功能, 设置此位为1, 将关闭模拟输入功能, 这个引脚可以作为 GPIO 使用							
3:2	-	保留不用							
4	TWIPS	TWI 端口选择							
6:5	-	保留不用							
7	WCE	IOCR 值改变使能位 用户在改变 IOCR 寄存器的值之前, 必须首先将此位写1, 在之后的6 个时钟周期内, 更改 IOCR 其他位的值。四个周期 后WCE 自动清零, 对IOCR 寄存器的更新操作无效。							

端口 B 下拉控制寄存器- PDNB									
PDNB - 端口B下拉控制寄存器									
PDNB: 0x11 (0x31)					默认值: 0x00				
Bits	-	DNB6	DNB5	DNB4	DNB3	DNB2	DNB1	DNB0	
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位定义									
[6:0]	DNB	端口B组下拉控制位							

端口 C 下拉控制寄存器- PDNC									
PDNC - 端口C下拉控制寄存器									

PDNC: 0x12 (0x32)				默认值: 0x00				
Bits	-	DNC6	DNC5	DNC4	DNC3	DNC2	DNC1	DNC0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位定义								
[6:0]	DNC	端口C组下拉控制位						

端口 D 下拉控制寄存器- PDND

PDND: 0x13 (0x33)				默认值: 0x00				
Bits	DND7	DND6	DND5	DND4	DND3	DND2	DND1	DND0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
位定义								
[7:0]	DND	端口D组下拉控制位						

端口 E 下拉控制寄存器- PDNE

PDNE: 0x14 (0x34)				默认值: 0x00				
Bits	-	-	DNE5	DNE4	DNE3	DNE2	DNE1	DNE0
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
位定义								
[5:0]	DNE	端口 E 组下拉控制位						

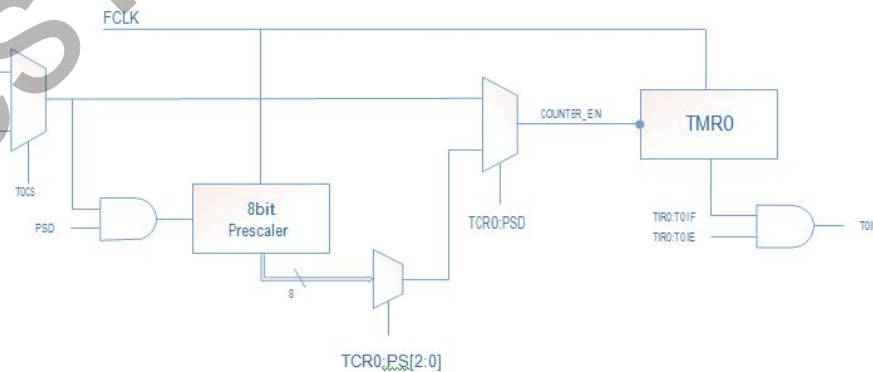
## 9. 定时/计数器 0 (TMRO)

- 8 位定时/计数寄存器 (TMRO)
- 8 位独立预分频器
- 溢出中断

### 9.1. 综述

TMRO 递增计数溢出时产生溢出中断，软件通过 TIRO 寄存器的 TOIF 位查询 TMRO 的中断标志。因为在休眠模式下 TMRO 的接口时钟被关闭，无法产生溢出中断，因此无法支持休眠唤醒。

TMRO 框架图：



### 9.2. 定时器模式

当工作于定时器模式，TMRO 的定时器在每个系统时钟周期递增(无预分频)，定时器模式通过清零 TCR0 寄存器的 TICS 位使

能。当对 TMRO 寄存器进行写操作，在写操作发生的接下来两个周期内，TMRO 寄存器被禁止递增。如果需要补充写周期后的定时误差，可以在写 TMRO 之前对将要写的值继续调整。

### 9.3. 可编程预分频器

TMRO 支持独立的 8 位可编程预分频器。

### 9.4. TMRO 中断处理

TMRO 在计数溢出 (FFh 到 00h) 时，TOIF 为在溢出中断发生时。无论系统有没有使能 TOIE，此中断标记位仍然会被置位为 1。在休眠模式下，TMRO 工作时钟被关闭，无法产生用于休眠唤醒中断的信号。

### 9.5. TMRO 寄存器定义

PSRO - TMRO 预分频寄存器

PSRO - TMRO 预分频寄存器								
地址: 0x47					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	PSRO[7:0]							
R/W	W/R							
Bit	Name	描述						
7:0	PSRO	定时器0 预分频寄存器 T0 的时钟源默认 1.5M $Ft0 = 1500 / ((PRO+1) * (PSR+1) * 2)$ kHz						

TCRO - TMRO 控制寄存器

TCRO - TMRO 控制寄存器								
地址: 0x44					默认值: 0x3F			
Bit	7	6	5	4	3	2	1	0
	TOEN	-	-	-	-	-	-	-
R/W	W/R	-	-	-	-	-	-	-
Bit	Name	描述						
7	TOEN	TMRO 使能控制位，写1 使能TMRO，写0 禁用TMRO						
6-0	-	保留未用						

TIRO - TMRO 中断控制寄存器

TIRO - TMRO 中断控制寄存器								
地址: 0x35					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	-	-	-	TOIE	-	-	-	TOIF
R/W	-	-	-	W/R	-	-	-	W/R
Bit	Name	描述						
7:5	-	保留未用						
4	TOIE	TMRO 中断使能寄存器						
3:1	-	保留未用						
0	TOIF	TMRO 中断标志位，写1 清零						

TMRO - TMRO 计数寄存器

TMRO - TMRO 计数寄存器								
地址: 0x46					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	TMRO[7:0]							
R/W	W/R							
Bit	Name	描述						
7:0	TMRO	定时器0 计数寄存器						

PRO - TMRO 周期寄存器								
地址: 0x45					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	PRO[7:0]							
R/W	W/R							
Bit	Name	描述						
7:0	PRO	定时器0 周期寄存器						

## 10. 定时/计数器 1 (TMR1)

- 16 位定时计数寄存器 TMR1 (TMR1={TMR1H:TMR1L})
- 16 位周期寄存器 PR1 (PR1={PR1H:PR1L})
- 可选内部低频时钟或系统时钟源
- 3 位独立预分频器
- 同步/异步模式
- TMR1 计数门控支持: 比较器
- 计数溢出中断
- 溢出唤醒 (内部低频时钟下的异步模式)

### 10.1. 综述

定时计数器 TC1 是一个 16 位的计数器。TC1 既可工作于同步模式，也可工作于异步模式，由 TCR1 寄存器的 T1SYNCB 位来选择。TC1 的计数时钟可来自于内部低频时钟和系统时钟，由多位寄存器位来选择。TC1 还可工作于门控模式，由内部模拟比较器的输出来控制计数。

TC1 的结构框架图如下：

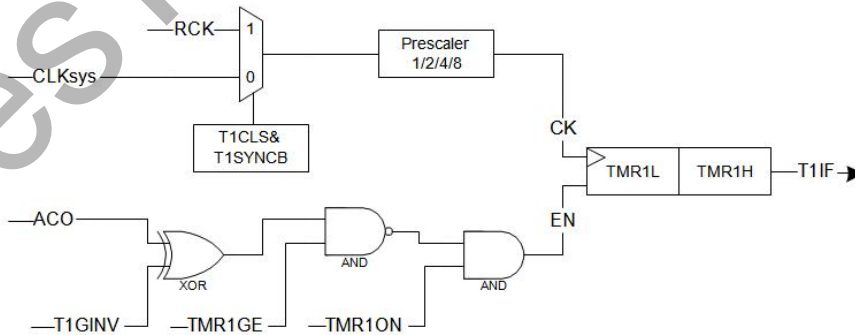


Figure 1 定时器1 结构框图

### 10.2. 时钟源

TC1 的时钟源由多位寄存器位来选择，如下表所示：

Table 1 TC1 时钟源选择表

T1SYNCB	T1CLS	时钟源
0	0	CLKsys
1	1	RCK

当设置 T1SYNCB 位为 0 时, TC1 工作于同步模式下, 时钟源为内部系统时钟。

当设置 T1SYNCB 位为 1 且 T1CLS 位为 1 时, TC1 工作于异步模式下, 时钟源为内部低频振荡器的输出时钟 RCK。

### 10.3. 预分频器

TC1 有一个 3 位的预分频器。预分频器支持 1/1, 1/2, 1/4 以及 1/8 分频, 由位于 TCR1 寄存器的 T1CKPS 位来配置。预分频器的溢出用于驱动 TC1 计数。预分频器本身不能够被直接访问。更新 TMR1 计数值寄存器 (TMR1H/TRM1L) 时会清零预分频器。

### 10.4. 同步模式

当设置 T1SYNCB 位为 0 时, TC1 工作在同步模式下, 使用内部系统时钟计数, 在内部时钟的驱动下进行累加。

### 10.5. 异步模式

当设置 T1SYNCB 位为 1 时, TC1 工作在异步模式下, 使用非系统时钟进行计数。只可选用内部振荡器产生的 RCK 时钟。此时 TC1 可以在系统休眠模式下继续运行。因此, TC1 的溢出中断可用于唤醒休眠模式。

TC1 工作于异步模式时, 在读写 TMR1 寄存器时, 需要特别注意:

首先, 异步模式下访问 TMR1 寄存器有专属的同步逻辑以保证访问的完整有效。但必须考虑到, 16 位的 TMR1 计数寄存器由两个 8 位的寄存器组合而成, 读取时必须分两次操作, 因此有可能在读取高低字节的间隔内, 计数器发生溢出, 此时读到的数据将不能完整反映当前计数器的状态。

然后, 对于写操作, 建议用户在写 TMR1 寄存器之前将 TC1 关闭。写操作与计数器的递增操作可能会导致最终写入不可预知的结果。

### 10.6. 门控模式

TC1 的门控输入可选用模拟比较器的输出。门控模式可实现用 TC1 测量外部事件。外部事件可以是来自模拟比较器反应的电平变化。可以利用这个功能实现简单的 Delta-Sigma ADC 转换以及其他多种模拟量的测试应用。

TC1 的门控输入的极性可以通过 TCR1 寄存器的 T1GINV 位来设置。用于实现测量高有效或者低有效的信号变化。

### 10.7. 溢出中断

当 TMR1 与 PR1 的值发生匹配时, TMR1 的值会回到 0x0000, 同时置位 TC1 的溢出中断标志 T1IF 位。软件可通过 T1IFR1 寄存器访问到 T1IF 位。当系统同时也使能了 T1IER1 寄存器的 T1IE 位以及全局中断使能 GIE 位时, 内核将响应 TC1 的溢出中断请求, PC 跳转至硬件中断向量, 进入中断服务程序后, 硬件会对 T1IF 位进行清零。软件也可以对 T1IF 位写“1”来进行清零。

TC1 支持在休眠模式下工作, 此时需要将 TC1 设置为异步计数模式。使能 TC1 溢出中断才能支持唤醒。当计数器发生溢出时, 将系统从休眠模式唤醒, 继续执行指令。如果使能了全局中断, 内核将响应溢出中断请求。

### 10.8. 寄存器定义

TCR1 - 定时器 1 控制寄存器

TCR1 - 定时器 1 控制寄存器

地址: 0x80		默认值: 0000_0000						
Bit	7	6	5	4	3	2	1	0
	-	-	T1CKPS1	T1CKPS0	-	-	-	TMR1ON



R/W	-	-	W/R	W/R	-	-	-	W/R
Bit	Name	描述						
7	-	保留						
6	-	保留						
5:4	T1CKPS	TMR1 计数时钟预分频器系数选择位; 00: 1/1 01: 1/2 10: 1/4 11: 1/8						
3:1	-	无						
0	TMR1ON	TMR1 使能控制位; 1: TMR1 被使能 0: TMR1 被禁止						

TMR1L - 定时器 1 计数值寄存器低字节

TMR1L - 定时器1 计数值寄存器低字节								
地址: 0x84				默认值: 0000_0000				
Bit	7	6	5	4	3	2	1	0
	TMR1L7	TMR1L6	TMR1L5	TMR1L4	TMR1L3	TMR1L2	TMR1L1	TMR1L0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	TMR1L	TC1 计数值低字节, TMR1L 与 TMR1H 组成16 位计数值TMR1						

TMR1H - 定时器 1 计数值寄存器高字节

TMR1H - 定时器1 计数值寄存器高字节								
地址: 0x85				默认值: 0000_0000				
Bit	7	6	5	4	3	2	1	0
	TMR1H7	TMR1H6	TMR1H5	TMR1H4	TMR1H3	TMR1H2	TMR1H1	TMR1H0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	TMR1H	TC1 计数值高字节, TMR1L 与 TMR1H 组成16 位计数值TMR1						

PR1L - 定时器 1 周期值寄存器低字节

PR1L - 定时器1 周期值寄存器低字节								
地址: 0x82				默认值: 1111_1111				
Bit	7	6	5	4	3	2	1	0
	PR1L7	PR1L6	PR1L5	PR1L4	PR1L3	PR1L2	PR1L1	PR1L0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R



Bit	Name	描述
7:0	PR1L	TC1 周期值低字节, PR1L 与PR1H 组成16 位周期值PR1

PR1H - 定时器 1 周期值寄存器高字节

PR1H - 定时器1 周期值寄存器高字节

地址: 0x83		默认值: 1111_1111						
Bit	7	6	5	4	3	2	1	0
	PR1H7	PR1H6	PR1H5	PR1H4	PR1H3	PR1H2	PR1H1	PR1H0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	PR1H	TC1 周期值高字节, PR1L 与PR1H 组成16 位周期值PR1						

TIER1 - 定时器 1 中断使能寄存器

TIER1 - 定时器1 中断使能寄存器

地址: 0x81		默认值: 0000_0000						
Bit	7	6	5	4	3	2	1	0
	-	-	-	DCM13IE	DCM12IE	DCM11E	ECP1IE	T1IE
R/W	-	-	-	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:5	-	保留						
4	DCM13IE	DCM3 匹配中断使能控制位; 1: DCM3 匹配中断被使能 0: DCM3 匹配中断被禁止						
3	DCM12IE	DCM2 匹配中断使能控制位; 1: DCM2 匹配中断被使能 0: DCM2 匹配中断被禁止						
2	DCM11IE	DCM1 匹配中断使能控制位; 1: DCM1 匹配中断被使能 0: DCM1 匹配中断被禁止						
1	ECP1IE	ECP1 自动开中断使能控制位; 1: ECP1 中断被使能 0: ECP1 中断被禁止						
0	T1IE	TMR1 溢出中断使能控制位; 1: TMR1 溢出中断被使能 0: TMR1 溢出中断被禁止						

TIFR1 - 定时器 1 中断标志寄存器

TIFR1 - 定时器1 中断标志寄存器

地址: 0x36		默认值: 0000_0000						
Bit	7	6	5	4	3	2	1	0
	-	-	-	DCM13IF	DCM12IF	DCM11IF	ECP1IF	T1IF
R/W	-	-	-	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:5	-	保留						

4	DCM13IF	DCM13 匹配中断标志位 定时器1 计数值TMR1 与PWM 占空比值DutyCycle1 匹配时, 会置位 DCM11F 位。若此时DCM1 中断使能位DCM1IE 为1 且全局中断使能, 会产生 DCM1 中断。执行此中断服务程序时硬件会清零 DCM11F 位。软件也可对 DCM11F 写“1”来清零该位。
3	DCM12IF	DCM12 匹配中断标志位 定时器1 计数值TMR1 与PWM 占空比值DutyCycle1 匹配时, 会置位 DCM11F 位。若此时DCM1 中断使能位DCM1IE 为1 且全局中断使能, 会产生 DCM1 中断。执行此中断服务程序时硬件会清零 DCM11F 位。软件也可对 DCM11F 写“1”来清零该位。
2	DCM11IF	DCM11 匹配中断标志位 定时器1 计数值TMR1 与PWM 占空比值DutyCycle1 匹配时, 会置位 DCM11F 位。若此时DCM1 中断使能位DCM1IE 为1 且全局中断使能, 会产生 DCM1 中断。执行此中断服务程序时硬件会清零 DCM11F 位。软件也可对 DCM11F 写“1”来清零该位。
1	ECP1IF	ECP1 自动关闭 中断标志位 发生自动关闭时, 会置位 ECP11F。若此时 ECP1 中断使能位ECP1IE 为1 且全局中断使能, 会产生ECP1 中断。执行此中断服务程序时硬件会清零ECP11F 位。软件也可对ECP11F 位写“1”来清零该位。
0	T1IF	TMR1 溢出中断标志位 当TMR1 计数器发生溢出时, 会置位溢出标志T11F 位。若此时溢出中断 使能位T1IE 为1 且全局中断使能, 则会产生溢出中断。执行此中断服 务程序时硬件会清零T11F 位。软件也可对T11F 位写1 来清零该位。

## 11. 定时/计数器 2 (TMR2)

- 12 位定时计数寄存器 TMR2
- 12 位周期寄存器 PR2
- 可编程预分频器 (1:1, 1:4, 1:16)
- 可编程后分频器 (1:1 到 1:16)
- TMR2 溢出中断

### 11.1. 综述

定时计数器 TC2 的计数时钟可来自于系统时钟。计数时钟输入到 TC2 内部的预分频器。预分频器支持 1/1, 1/4 以及 1/16 分频, 由位于 TCR2 寄存器的 T2CKPS 位来配置。预分频器的溢出用于驱动 TC2 计数。

TC2 的计数方式为单向递增计数, 即计数器从最小值 0 开始递增, 累加到最大值 PR2 后又会回到最小值 0 重新递增。

当 TMR2 与 PR2 的值发生匹配时, TC2 的后分频器递增。后分频器的分频系数从 1:1 到 1:16 可选, 由位于 TCR2 寄存器的 TOUTPS 位来配置。当 TC2 的后分频器发生溢出时, 会置位 TC2 的溢出中断标志位 T2IF 位。软件可通过 PIR1 寄存器访问到 T2IF 标志位。

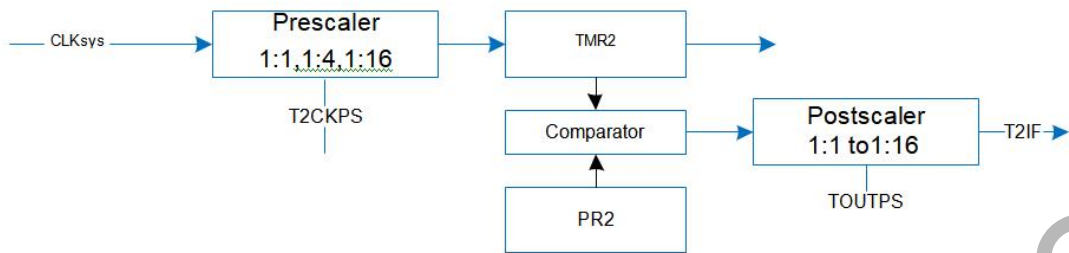
TMR2 寄存器与 PR2 寄存器支持软件读写访问。在发生任何种类的复位后, TMR2 寄存器被复位到 0, PR2 寄存器被复位到 0xFF。

预分频器和后分频器不能被软件直接访问, 在以下情况发生时, 将清零分频计数器:

- 写 TMR2 寄存器;
- 写 TCR2 寄存器;
- 任何类型的复位

通过设置 TCR2 寄存器的 TMR2ON=1 使能 TC2 模块；清零 TMR2ON 将禁止 TC2 模块的任何功能。

TC2 的结构框架图



## 11.2. 寄存器定义

TCR2 - 定时器 2 控制寄存器

TCR2 - 定时器2控制寄存器

地址: 0xB0									X0000_0000									
Bit	7	6	5	4	3	2	1	0										
	TC2DIR	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0										
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R										
Bit	Name	描述																
7	-	-																
6:3	TOUTPS	TMR2 输出后分频系数选择位; 0000: 1/1 0001: 1/2 0010: 1/3 ..... 1101: 1/14 1110: 1/15 1111: 1/16																
2	TMR2ON	TMR2 使能控制位; 1: 使能 0: 禁止																
1:0	T2CKPS	TMR2 计数时钟预分频器系数选择位; 00: 1/1 01: 1/4 1X: 1/16																

TMR2 - 定时器 2 计数值寄存器

TMR2 - 定时器2计数值寄存器低字节

地址: 0xB2									X0000_0000									
Bit	7	6	5	4	3	2	1	0										
	TMR27	TMR26	TMR25	TMR24	TMR23	TMR22	TMR21	TMR20										
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R										
Bit	Name	描述																
7:0	TMR2	TC2计数值低 8 位																

PR2 - 定时器 2 周期值寄存器



PR2 - 定时器2 周期值寄存器低字节								
地址: 0xB1					X1111_1111			
Bit	7	6	5	4	3	2	1	0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	PR2	TC2 周期值低 8 位						

TMR2 - 定时器2 计数值寄存器低字节								
地址: 0xB3					X1111_0000			
Bit	7	6	5	4	3	2	1	0
	PR2B	PR2A	PR29	PR28	TMR2B	TMR2A	TMR29	TMR28
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:4	PR2H	TC2 周期值高 4 位						
3:0	TMR2H	TC2 计数值高 4 位						

TIER2 - 定时器 2 中断使能寄存器								
地址: 0xB8					X0000_0000			
Bit	7	6	5	4	3	2	1	0
	-	-	-	DC23IE	DC22IE	DC21IE	ECP2IE	T2IE
R/W	-	-	-	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:5	-	保留						
4	DC23IE	DCM23 匹配中断使能控制位; 1: DCM23 中断被使能 0: DCM23 中断被禁止						
3	DC22IE	DCM22 匹配中断使能控制位; 1: DCM22 中断被使能 0: DCM22 中断被禁止						
2	DC21IE	DCM21 匹配中断使能控制位; 1: DCM21 中断被使能 0: DCM21 中断被禁止						
1	ECP2IE	ECP2 自动充电中断使能控制位; 1: ECP2 中断被使能 0: ECP2 中断被禁止						
0	T2IE	TMR2 溢出中断使能控制位; 1: TMR2 溢出中断被使能 0: TMR2 溢出中断被禁止						

TIFR2 - 定时器 2 中断标志寄存器								
地址: 0x37					X0000 0000			
Bit	7	6	5	4	3	2	1	0

R/W	-	-	-	DCM23IF	DCM22IF	DCM21IF	ECP2IF	T2IF
Bit	Name	描述						
7:5	-	保留						
4	DCM23IF	DCM23 匹配中断标志位 定时器2计数值TMR2 与PWM 占空比值DutyCycle23 匹配时, 会置位 DCM23IF 位。执行中断服务程序时硬件会清零 DCM23IF 位。软件也可对 DCM23IF 写“1”来清零该位。						
3	DCM22IF	DCM22 匹配中断标志位 定时器2计数值TMR2 与PWM 占空比值DutyCycle22 匹配时, 会置位 DCM22IF 位。执行中断服务程序时硬件会清零 DCM22IF 位。软件也可对 DCM22IF 写“1”来清零该位。						
2	DCM21IF	DCM21 匹配中断标志位 定时器2计数值TMR2 与PWM 占空比值DutyCycle21 匹配时, 会置位 DCM21IF 位。执行中断服务程序时硬件会清零 DCM21IF 位。软件也可对 DCM21IF 写“1”来清零该位。						
1	ECP2IF	ECP2 自动关闭中断标志位; 1: ECP2 中断标志位被置位 0: ECP2 中断标志位未置位 ECP2工作在PWM 模式下, PWM自动关闭事件发生时置位ECP2IF 位。若此时中断使能ECP2IE 位为1 且全局中断使能, 则会产生ECP2 中断。执行此中断服务程序时硬件会清零ECP2IF 位, 软件对ECP2IF 位写1 也会清零该位。						
0	T2IF	TMR2 溢出中断标志位; 1: TMR2 计数溢出时会置位溢出中断标志 T2IF 位 0: TMR2 溢出中断标志T2IF 位未置位 当TMR2 计数器发生溢出时, 会置位溢出标志T2IF 位。若此时溢出中断 使能T2IE 位为1 且全局中断使能, 则会产生溢出中断。执行此中断服 务程序时硬件会清零T2IF 位, 或者软件对T2IF 位写1 也会清零该位。						

## 12. 增强型比较/俘获/PWM 模块 0 (EECP1)

- 支持 Capture 模式, 捕捉外部事件, 支持 1 路输入通道
- 支持 Compare 模式, 触发外部事件
- 支持 PWM 模式, 产生 1 组带死区控制的互补 PWM 信号

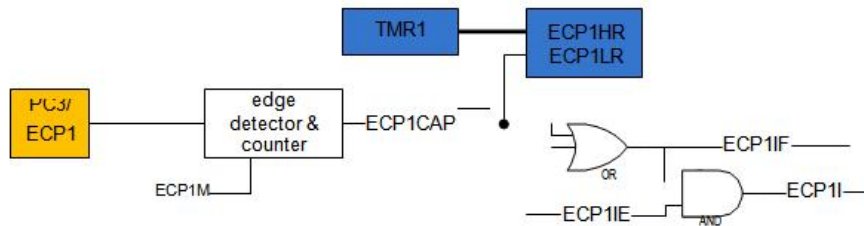
### 12.1. 综述

本章节所描述的捕获、比较及脉冲宽度调制模块 (简称为 ECP1 模块) 是一个用来计时和控制各种事件的外设。Capture 模式下, 可以对外部事件进行计数。Compare 模式下, 当预设的计时器溢出时可以触发外部事件。PWM 模式下, 可以产生单路频率和占空比可调的 PWM 信号, 也产生 1 组带死区控制的频率和占空比可调的互补 PWM 信号。

### 12.2. Capture 模式

Capture 模式下, 当捕获事件发生时, 定时器 1 的 16 位计数值 TMR1 会被捕获到 ECPR1H:ECPR1L 寄存器中去, 同时置位 ECP1 中断标志 ECP1IF 位。

Capture 模式的结构图



### Capture 使能配置

要使用 Capture 模式，须设置 ECP1CR 寄存器中的 ECP1M[3:2] 位的值为“01”，此时 Compare 模式和 PWM 模式均失效。

### Capture 模式配置

Capture 模式下，触发捕获的事件为以下四种事件中的一种，由 ECP1CR 寄存器中的 ECP1M[1:0] 位来选择，包括：

- 任一下降沿
- 任一上升沿
- 第 4 个上升沿
- 第 16 个上升沿

Capture 模式下，定时器 1 须工作在定时模式或同步计数模式。异步模式下，捕获可能工作不正常。

Capture 模式下，外部事件从 ECP1 引脚（PC3）输入，因此要配置 ECP1 引脚作为输入。

Capture 模式下，当所选捕获源的事件发生时，定时器 1 的 16 位计数值 TMR1 会被捕获到 ECP1HR:ECP1LR 中去。软件通过读取 ECP1HR:ECP1LR 寄存器中的值获得捕获值。在 ECP1HR 和 ECP1LR 寄存器的值被读走之前，如果发生新的捕获，原来的捕获值会被新的捕获值覆盖掉。

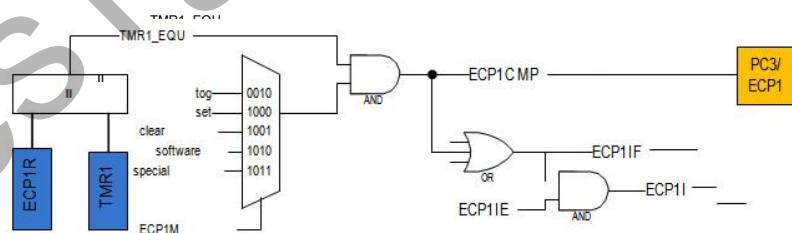
### Capture 中断

当捕获产生后，位于 TIFR1 寄存器中的 ECP1 中断标志位 ECP1IF 也会被置位。当位于 TIER1 寄存器中的 ECP1 中断使能位 ECP1IE 被置位且全局中断使能时，会执行 ECP1I 中断服务程序后，ECP1IF 位会被自动清零，软件也可以对该位写“1”来清零。

当 Capture 模式发生改变时，可能会产生一次假的捕获中断。用户需要保持 ECP1IE 位为零来避免此假中断。此外，还需要在改变 Capture 模式后清零 ECP1IF 中断标志位。

## 12.3. Compare 模式

Compare 模式结构图



在 Compare 模式下，16 位的 ECP1R 寄存器会持续和 TMR1 寄存器进行比较，当比较匹配发生时，ECP1 模块会：

- 翻转 ECP1 输出
- 置位 ECP1 输出
- 清零 ECP1 输出
- 产生一个特定的事件触发
- 产生一个软件中断 同时置位 ECP1 中断标志 ECP1IF 位。

这些操作是由位于 ECP1CR 寄存器的 ECP1M[3:0] 位来选择控制的。其中前三种模式，ECP1 模块会直接对 ECP1 引脚的进行控制，而后两种模式会失去对 ECP1 引脚的控制（见 ECP1CR 寄存器）。在比较模式下，定时器 1 须工作在定时模式或同步计数模式。异

步计数模式下，比较可能工作不正常。

### Compare 使能配置

要使能 Compare 模式，须设置 ECP1CR 寄存器中的 ECP1M[3:2]位的值为“10”或设置 ECP1M 的值为“0010”，此时 Capture 模式和 PWM 模式均失效。

### Compare 中断

当比较事件产生后，位于 TIFR1 寄存器中的 ECP1 中断标志位 ECP1IF 也会被置位。当位于 TIER1 寄存器中的 ECP1 中断使能位 ECP1IE 被置位且全局中断使能时，会执行 ECP1 中断。进入 ECP1 中断服务程序后，ECP1IF 位会被自动清零，软件也可以对该位写“1”来清零。

当选择产生软件中断模式时 (ECP1M[3:0]=1010b)，ECP1 模块失去对 ECP1 引脚的控制（见 ECP1CR 寄存器）。

当选择特殊事件触发模式时 (ECP1M[3:0]=1011b)，ECP1 模块将会：

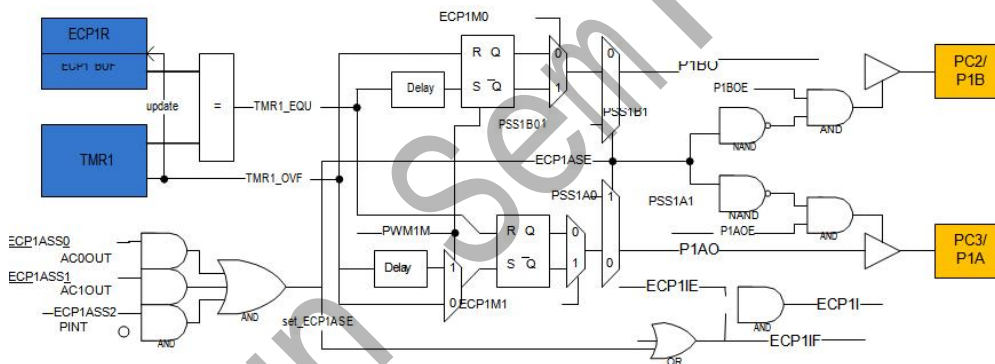
- 复位定时器 1
- ADC 使能时启动一次 ADC 转换

同时 ECP1 模块失去对 ECP1 引脚的控制（见 ECP1CR 寄存器）。

当 TMR1H, TMR1L 寄存器对与 ECPR1H, ECPR1L 寄存器对发生比较匹配时，ECP 模块的特殊事件触发输出会立即产生。TMR1H, TMR1L 寄存器对会在接下来的定时器 1 时钟上升沿发生复位。这就允许 ECPR1H, ECPR1L 寄存器对为定时器 1 有效提供 16 位可编程周期寄存器。

## 12.4. PWM 模式

PWM 模式的结构图



### PWM 使能配置

要使能 PWM 模式，须设置 ECP1CR 寄存器中的 ECP1M[3:2]位的值为“11”，此时 Capture 和 Compare 模式均失效。

### PWM 模式配置

PWM 模式下，当设置位于 ECP1CR 寄存器的 PWM1M 位为 0 时，ECP1 模块只产生单路占空比和频率可调的 PWM 信号（用 T1PWM 来表示）；当设置 PWM1M 位为 1 时，可产生 1 组带死区控制的互补 PWM 信号（分别用 T1PWMP 和 T1PWMN 来表示）。下面的描述中，用 T1PWMx 来表示 T1PWM, T1PWMP 或 T1PWMN。

PWM 信号的极性是可配置的，通过设置位于 ECP1CR 寄存器的 ECP1M[1:0]位来选择。

要使 PWM 信号输出到引脚上，须设置其相应的输出使能信号 P1A0E 和 P1B0E 位为 1 才会正常输出。其中，T1PWM 和 T1PWMP 是输出在 P1A 引脚上，T1PWMN 是输出在 P1B 引脚上。

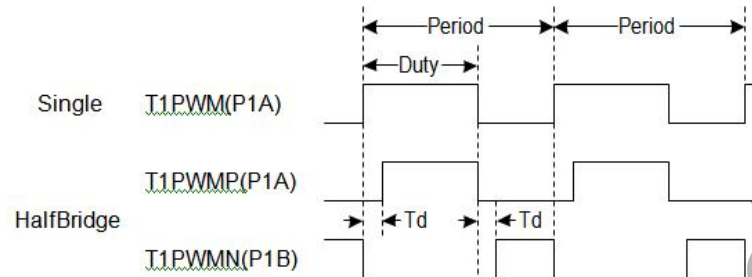
下面表格为 PWM 模式下引脚的分配情况。

PWM 输出模式	PWM1M	P1A 引脚信号	P1B 引脚信号
单路输出	0	T1PWM	-
互补输出	1	T1PWMP	T1PWMN

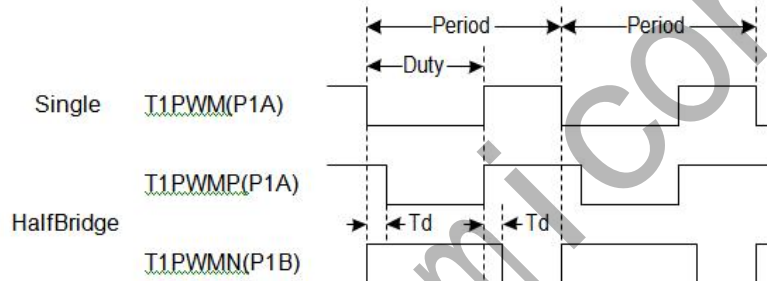
下面表格为 PWM 模式下引脚的有效电平状态

PWM 模式	PWM1M	ECP1M[1]=0	ECP1M[1]=1	ECP1M[0]=0	ECP1M[0]=1
单路输出	0	P1A 为高有效	P1A 为低有效	-	-
互补输出	1	P1A 为高有效	P1A 为低有效	P1B 为高有效	P1B 为低有效

下图为 PWM 模式下的输出波形（高有效）



下图为 PWM 增强模式下的输出波形（低有效）



上面两图中，Period 为 PWM 的周期，Duty 为 PWM 的占空比，Td 为死区时间。

### PWM 周期

PWM 周期由定时器 1 的 PR1 寄存器决定。PWM 周期由下面的公式来计算：

$$\text{PWM Period} = [\text{PR1} + 1] * \text{Tosc} * \text{TMR1PR}$$

其中，PR1 为计数最大值，Tosc 为定时器 1 的计数时钟，TRM1PR 为预分频系数。

当 TMR1 等于 PR1 时，下一个计数周期会发生以下三件事：

- TMR1 被清零
- T1PWM 被置位（例外：若 PWM 占空比为 0%，PWM 信号将不会被置位）
- PWM 占空比值由 ECPR1L 锁存到其相应的缓存器 ECPR1H 中（DC1LM=1 时）

### PWM 占空比

PWM 模式下，当设置位于 ECP1CR 寄存器的 DC1LM 位为 0 时，PWM 的占空比由 16 位寄存器 ECPR1H:ECPR1L 的值来决定，软件要更新占空比值，可直接写入此 16 位寄存器中，需要注意写入的时机；当 DC1LM 位为 1 时，PWM 的占空比由 8 位寄存器 ECPR1L 的值来决定，在定时器 1 发生计数溢出时，ECPR1L 中的值会锁存到 ECPR1H 中去，ECPR1H 中的值即为 PWM 的最新占空比值，软件要更新找空比值，可随时写入 ECPR1L 寄存器中，硬件自动更新到 ECPR1H 中去。

PWM 占空比值由下面的公式来计算：

$$\text{DutyCycle1} = \text{DC1LM} ? \text{ECPR1H} : \{\text{ECPR1H}:\text{ECPR1L}\}$$

PWM 脉冲宽度由下面的公式来计算：

$$\text{Pulse Width} = (\text{DutyCycle1}) * \text{Tosc} * (\text{TMR1PR}) \text{ PWM}$$

占空比率由下面的公式来计算：

$$\text{Duty Cycle Ratio} = (\text{DutyCycle1}) / ((\text{PR1}+1))$$

当 TRM1 和 DutyCycle1 发生匹配时, T1PWM 信号被清零。

### PWM 分辨率

对于给定的周期, 分辨率决定了可能的占空比周期数。16 位分辨率有 65536 个连续的占空比周期, 10 位分辨率有 1024 个连续的占空比周期, 8 位分辨率有 256 个占空比周期。

当 PR1 被设置为 65535 时, 拥有最大的 PWM 分辨率, 即 16 位。分辨率是 PR1 寄存器的函数, 如下面公式所示:

$$\text{Resolution} = \log[(\text{PR1} + 1)] / \log_2 \text{ bits}$$

下面表格为 PWM 频率和占空比的示例。

PWM 频率和分辨率 (Fosc = 16 MHz)

Frequency (KHz)	0.031	0.061	0.244	15.625	62.5	250
TMR1 Prescale	8	4	1	1	1	1
PR1 Value	0xFFFF	0xFFFF	0xFFFF	0x3FF	0xFF	0x3F
Max Resolution	16	16	16	10	8	6

### PWM 操作的配置

当配置 ECP1 模块工作在 PWM 模式时, 可按照以下流程来执行:

1. 清零 P1AOE 或 P1BOE 位来禁止 PWM 引脚 (P1A 或 P1B) 的输出驱动;
2. 加载 PR1 寄存器来设置 PWM 周期;
3. 加载 ECPR1H 寄存器和 ECPR1L 寄存器来设置 PWM 占空比;
4. 加载 ECP1CR 寄存器来设置 PWM 模式以及占空比的更新方式;
5. 配置和启动定时器 1, 包括:
  - 清零位于 TIFR1 寄存器的 T1IF 中断标志位
  - 加载位于 TCR1 寄存器的 T1CKPS 位来设置定时器 1 的预分频系数
  - 置位位于 TCR1 寄存器的 TMR1ON 位来使能定时器 1

一个新的 PWM 周期开始后, 使能 PWM 输出, 即:

- 等待定时器 1 溢出 (即位于 TIFR1 寄存器的 T1IF 位置位)
- 置位 P1AOE 或 P1BOE 位来使能 PWM 引脚 (P1A 或 P1B) 的输出驱动

### PWM 互补输出

当设置 PWM1M 位为 1 时, ECP1 模块会产生互补输出 PWM 信号 T1PWMP 和 T1PWMN, 分别输出到引脚 P1A 和 P1B 上, 这个模式可用在半桥应用上。

半桥应用当中, 所有的电源开关都通过 PWM 频率来调制。通常情况下, 电源开关的关闭比打开需要更多的时间。如果上下电源开关在同一时间切换 (一个打开, 另一个关闭), 两个开关可能会在短时间同时处于打开状态, 直到其中一个完全关闭。在这段短时间内, 两个开关之间会通过一段很大的电流 (直通电流), 造成半桥短路。为了避免在切换过程中产生这种潜在的破坏性直通电流, 通常在每个开关打开前加上一段延时, 以保证另一个开关已经完全关闭。

半桥模式下, 可编程的死区延用来防止在半桥器件之间产生直通电流。死区延时发生在信号由非有效状态切换至有效状态期间。位于 PWM1CR 寄存器的 DT1[6:0] 位, 用来设置输出驱动有效前的 PWM 周期数。若 DT1 的值大于占空比的周期数, 相应的输出在整个 PWM 周期内都会保持无效。

### PWM 自动关闭

PWM 模式支持自动关闭, 当关闭事件发生时, 将会禁止 PWM 输出。自动关闭模式可以放置 PWM 输出引脚至预设状态。这可以用来放置 PWM 损坏应用电路。

位于 ECP1ASR 寄存器的 ECP1ASS[2:0] 位用来选择自动关闭源。自动关闭事件可由下面来产生:

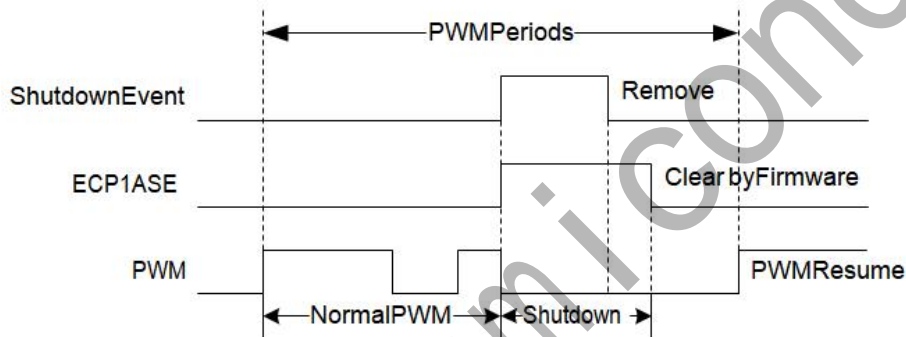
- 比较器 0 的输出

- 比较器 1 的输出
- PINT 引脚上的“0”电平
- 软件设置 ECP1ASE 位

位于 ECP1ASR 寄存器的 ECP1ASE 位用来表明自动关闭的状态。当 ECP1ASE 位为“0”时，PWM 输出为正常状态。当 ECP1ASE 位为“1”时，PWM 输出位自动关闭状态。

当自动关闭事件发生时，会发生下面两件事：

- 1) ECP1ASE 位被置位。ECP1ASE 位会保持为高，直到软件把它清零，或者发生自动重启；
- 2) ECP1IF 位被置位。ECP1 中断使能时，会产生 ECP1 中断。
- 3) PWM 引脚输出立刻被切换至自动关闭预设状态。PWM 输出引脚 P1A 和 P1B 的自动关闭状态可由位于 ECP1ASR 寄存器的 PSS1A 和 PSS1B 位来选择。每个引脚可切换至以下三种状态中的一种：
  - 驱动为逻辑高电平
  - 驱动为逻辑低电平
  - 高阻状态 下图为自动关闭操作示例图



自动关闭条件是电平驱动信号，非边沿驱动信号。当电平保持不变时，自动关闭状态可持续。当自动关闭条件保持时，软件写 ECP1ASE 位为无效操作。

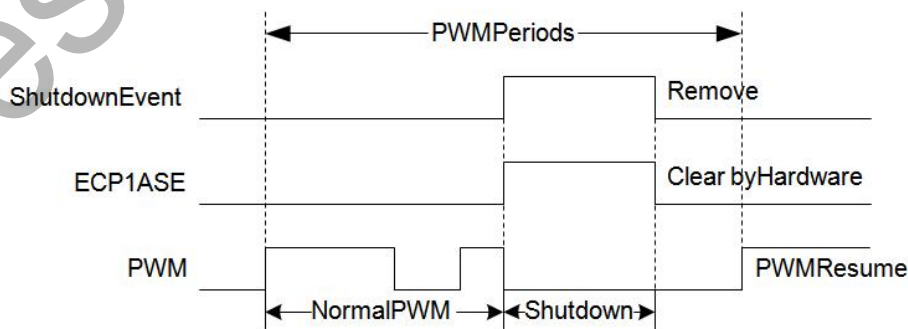
当自动关闭条件移除后，PWM 会重启（软件或自动重启），PWM 信号输出会在接下来的 PWM 周期开始重启。

#### PWM 自动重启

ECP1 模块可配置 PWM 信号自动重启，当自动关闭条件移除后，PWM 信号可自动重新启动输出。置位位于 PWM1CR 寄存器的 PRS1EN 位，即可使能自动重启功能。

自动重启被使能后，自动关闭条件仍有效时，ECP1ASE 位将会保持置位。当自动关闭条件移除时，ECP1ASE 位会被硬件清零，PWM 恢复常规操作。

下图为自动重启操作示例图



## 12.5. 寄存器定义

ECP1L - E CPR1 寄存器低字节



ECPR1L - 寄存器低字节								
地址: 0x86					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	ECPR1L7	ECPR1L6	ECPR1L5	ECPR1L4	ECPR1L3	ECPR1L2	ECPR1L1	ECPR1L0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	ECPR1L	ECPR1 寄存器低字节 Capture 模式: 捕获发生时, 所记录TMR1L 的值 Compare 模式: 发生比较时, 与TRM1L 进行比较的值 PWM 模式: 产生 PWM 时, 16 位可调占空比低八位的值						

ECPR1H - ECPR1 寄存器高字节								
地址: 0x87					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	ECPR1H7	ECPR1H6	ECPR1H5	ECPR1H4	ECPR1H3	ECPR1H2	ECPR1H1	ECPR1H0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	ECPR1H	ECPR1 寄存器高字节 Capture 模式: 捕获发生时, 所记录TMR1H 的值 Compare 模式: 发生比较时, 与TRM1H 进行比较的值 PWM 模式: 产生 PWM 时, 16 位可调占空比高八位的值						

ECP1R2L - ECPR2 寄存器低字节								
地址: 0x88					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	ECP1R2L7	ECP1R2L6	ECP1R2L5	ECP1R2L4	ECP1R2L3	ECP1R2L2	ECP1R2L1	ECP1R2L0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	ECP1R2L	ECPR2 寄存器低字节 Capture 模式: 捕获发生时, 所记录TMR1L 的值 Compare 模式: 发生比较时, 与TRM1L 进行比较的值 PWM 模式: 产生 PWM 时, 16 位可调占空比低八位的值						

ECP1R2H - ECPR2 寄存器高字节								
地址: 0x89					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	ECP1R2H7	ECP1R2H6	ECP1R2H5	ECP1R2H4	ECP1R2H3	ECP1R2H2	ECP1R2H1	ECP1R2H0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						

7:0	ECPR2H	<p>ECPR2 寄存器高字节</p> <p>Capture 模式: 捕获发生时, 所记录TMR1H 的值</p> <p>Compare 模式: 发生比较时, 与TRM1H进行比较的值</p> <p>PWM 模式: 产生 PWM 时, 16 位可调占空比高八位的值</p>
-----	--------	--

**ECP1R3L - ECP1R3 寄存器低字节**
**ECP1R3L - 控制寄存器**

地址: 0x8A					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	ECPR3L7	ECPR3L6	ECPR3L5	ECPR3L4	ECPR3L3	ECPR3L2	ECPR3L1	ECPR3L0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	ECPR3L	<p>ECPR3 寄存器低字节</p> <p>Capture 模式: 捕获发生时, 所记录TMR1L 的值</p> <p>Compare 模式: 发生比较时, 与TRM1L进行比较的值</p> <p>PWM 模式: 产生 PWM 时, 16 位可调占空比低八位的值</p>						

**ECPR3H - ECP3 寄存器高字节**
**ECP1R3H - 控制寄存器**

地址: 0x8B					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	ECPR3H7	ECPR3H6	ECPR3H5	ECPR3H4	ECPR3H3	ECPR3H2	ECPR3H1	ECPR3H0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	ECPR3H	<p>ECPR3 寄存器高字节</p> <p>Capture 模式: 捕获发生时, 所记录TMR1H 的值</p> <p>Compare 模式: 发生比较时, 与TRM1H进行比较的值</p> <p>PWM 模式: 产生 PWM 时, 16 位可调占空比高八位的值</p>						

**TIER1 - 定时器 1 中断使能寄存器**
**TIER1 - 定时器1 中断使能寄存器**

地址: 0x81					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
R/W	-	-	-	-	-	DCM1IE	ECP1IE	T1IE
						W/R	W/R	W/R
Bit	Name	描述						
7:3	-	保留						
2	DCM1IE	<p>DCM1 匹配中断使能控制位;</p> <p>1: DCM1 匹配中断被使能</p> <p>0: DCM1 匹配中断被禁止</p>						
1	ECP1IE	<p>ECP1 中断使能控制位;</p> <p>1: ECP1 中断被使能</p> <p>0: ECP1 中断被禁止</p>						



0	T1IE	TMR1 溢出中断使能控制位; 1: TMR1 溢出中断被使能 0: TMR1 溢出中断被禁止
---	------	---

ECP1DTP - ECP1 上管死区时间

ECP1DTP - 寄存器

地址: 0x66					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
R/W	W/R							
Bit	Name	描述						
7:0	-	P-Channel Dead Time Select for ECOP1						

ECP1DTN - ECP1 下管死区时间

ECP1DTN - 寄存器

地址: 0x67					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
R/W	W/R							
Bit	Name	描述						
7:0	-	N-Channel Dead Time Select for ECOP1						

TIFR1 - 定时器 1 中断标志寄存器

TIFR1 - 定时器 1 中断标志寄存器

地址: 0x36					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	DCM1IF	ECP1IF	T1IF
R/W	-	-	-	-	-	W/R	W/R	W/R
Bit	Name	描述						
7:3	-	保留						
2	DCM1IF	DCM1 中断标志位 定时器1 计数值TMR1 与PWM 占空比值DutyCycle1 匹配时, 会置位 DCM1IF 位。若此时 DCM1 中断使能位DCM1IE 为1 且全局中断使能, 会产生DCM1 中断。执行此中断服务程序时硬件会清零DCM1IF 位。软件也可对 DCM1IF 写“1”来清零该位。						
1	ECP1IF	ECP1 中断标志位 发生捕获、比较匹配或自动关闭时, 会置位 ECP1IF。若此时 ECP1 中断使能位ECP1IE 为1 且全局中断使能, 会产生ECP1 中断。执行此中断服务程序时硬件会清零 ECP1IF 位。软件也可对 ECP1IF 位写“1”来 清零该位。						

0	T1IF	<p>TMR1 溢出中断标志位</p> <p>当 TMR1 计数器发生溢出时，会置位溢出标志 T1IF 位。若此时溢出 中断使能位 T1IE 为 1 且全局中断使能，则会产生溢出中断。执行此 中断服务程序时硬件会清零 T1IF 位。软件也可对 T1IF 位写 1 来清零 该位。</p>
---	------	---

**ECP1CR0 - ECP1 控制寄存器 0**

ECP1CR0 - ECP1控制寄存器 0								
地址: 0x68					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	P1PH3	P10E3	P1PH2	P10E2	P1PH1N	P10E1N	P1PH1P	P10E1P
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	P1PH3	ECP1 通道 3 单路 PWM 输出极性选择位						
6	P10E3	ECP1 通道 3 单路 PWM 输出使能位						
5	P1PH3	ECP1 通道 2 单路 PWM 输出极性选择位						
4	P10E3	ECP1 通道 2 单路 PWM 输出使能位						
3	P1PH1N	ECP1 通道 1N 路 PWM 输出极性选择位						
2	P10E1N	ECP1 通道 1N 路 PWM 输出使能位，ECP1_0E1P 和 ECP1_0E1N 同时为 1 时，组成互补输出，否则为单路输出						
1	P1PH1P	ECP1 通道 1P 路 PWM 输出极性选择位						
0	P10E1P	ECP1 通道 1P 路 PWM 输出使能位，ECP1_0E1P 和 ECP1_0E1N 同时为 1 时，组成互补输出，否则为单路输出						

**ECP1CR1 - ECP1 控制寄存器 1**

ECP1CR1 - ECP1控制寄存器 1								
地址: 0x69					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	P1AES	P1ASS2	P1ASS1	P1ASS0	P1PSN1	P1PSN0	P1PSP1	P1PSP0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	P1AES	ECP1 通道 1PWM 输出自动关闭标志位，当自动关闭触发后，ECP1_ASE 位会被置位，PWM 输出自动关闭，引脚状态由 ECP1_PSP 和 ECP1_PSN 决定						
6	P1ASS2	ECP1 通道 1PWM 输出自动关闭触发源 2 使能位，当设置为 1 时，外部引脚 PINT 为低电平时会关闭通道 1 的 PWM 输出						
5	P1ASS1	ECP1 通道 1PWM 输出自动关闭触发源 1 使能位，当设置为 1 时，比较器 1 输出滤波后为高电平时会关闭通道 1 的 PWM 输出						
4	P1ASS0	ECP1 通道 1PWM 输出自动关闭触发源 0 使能位，当设置为 1 时，比较器 0 输出滤波后为高电平时会关闭通道 1 的 PWM 输出						
3	P1PSN1	ECP1 通道 1PWM 输出自动关闭下 1N 路的输出禁止位，当设置为 1 时，自动关闭发生后，1N 路会被禁止输出，引脚状态由 GPIO 控制						
2	P1PSN0	ECP1 通道 1PWM 输出自动关闭下 1N 路的输出状态选择位，当设置 ECP1_PSN1 为 0 时，自动关闭发生后，1N 路引脚状态由 ECP1_PSN0 来选择						
1	P1PSP1	ECP1 通道 1PWM 输出自动关闭下 1P 路的输出禁止位，当设置为 1 时，自动关闭发生后，1P 路会被禁止输出，引脚状态由 GPIO 控制						
0	P1PSP0	ECP1 通道 1PWM 输出自动关闭下 1P 路的输出状态选择位，当设置 ECP1_PSP1 为 0 时，自动关闭发生后，1P 路引脚状态由 ECP1_PSP0 来选择						

**ECP1CR2 - ECP1 控制寄存器 2**

ECP1CR2 - EEC1控制寄存器 2								
地址: 0x6A					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	P1ARE	P1DC3LM	P1DC2LM	P1DC1LM	-	-	-	-
R/W	W/R	W/R	W/R	W/R	-	-	-	-
Bit	Name	描述						
7	P1ARE	ECP1 通道 1PWM 输出自动重启使能位, 当设置为 1 时, 自动关闭触发条件去除后, ECP1_ASE 会被清零, 通道 1 的 PWM 输出恢复						
6	P1DC3LM	ECP1 通道 3PWM 占空比加载方式选择位, 当设置为 0 时, 软件写入占空比寄存器的值直接更新至占空比缓冲器中并立即生效; 当设置为 1 时, 定时器 1 溢出后占空比寄存器中的值才更新至占空比缓冲器中并生效						
5	P1DC3LM	ECP1 通道 2PWM 占空比加载方式选择位, 当设置为 0 时, 软件写入占空比寄存器的值直接更新至占空比缓冲器中并立即生效; 当设置为 1 时, 定时器 1 溢出后占空比寄存器中的值才更新至占空比缓冲器中并生效						
4	P1DC3LM	ECP1 通道 1PWM 占空比加载方式选择位, 当设置为 0 时, 软件写入占空比寄存器的值直接更新至占空比缓冲器中并立即生效; 当设置为 1 时, 定时器 1 溢出后占空比寄存器中的值才更新至占空比缓冲器中并生效						
3:0	-	保留不用						

**ECP1CR3 - ECP1 控制寄存器 3**

ECP1CR3 - EEC1控制寄存器 3								
地址: 0x63					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	P1CP3EN	P1CP2EN	P1CP3M1	P1CP3M0	P1CP2M1	P1CP2M0	P1CP3S	P1CP2S
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	P1CP3EN	ECP1 通道 3 输入捕获使能位, 当设置为 1 且 ECP1_OE3 设置为 0 时, 通道 3 输入捕获被使能, 发生捕获时 TMR1 寄存器的值会记录到 ECP1R3 中						
6	P1CP2EN	ECP1 通道 2 输入捕获使能位, 当设置为 1 且 ECP1_OE2 设置为 0 时, 通道 3 输入捕获被使能, 发生捕获时 TMR1 寄存器的值会记录到 ECP1R2 中						
5	P1CP3M1	ECP1 通道 3 输入捕获模式选择位, 00=双沿捕获; 01=正沿捕获; 10=每两个正沿捕获; 11=每四个正沿捕获						
4	P1CP3M0	ECP1 通道 3 输入捕获模式选择位, 00=双沿捕获; 01=正沿捕获; 10=每两个正沿捕获; 11=每四个正沿捕获						
3	P1CP2M1	ECP1 通道 2 输入捕获模式选择位, 00=双沿捕获; 01=正沿捕获; 10=每两个正沿捕获; 11=每四个正沿捕获						
2	P1CP2M0	ECP1 通道 2 输入捕获模式选择位, 00=双沿捕获; 01=正沿捕获; 10=每两个正沿捕获; 11=每四个正沿捕获						
1	P1CP3S	ECP1 通道 3 输入捕获源选择位, 0=外部引脚 PE0 输入; 1=外部引脚 DM0_RXD 输入						
0	P1CP3S	ECP1 通道 2 输入捕获源选择位, 0=外部引脚 PD4 输入; 1=外部引脚 DP0_RXD 输入						

### 13. 增强型比较/俘获/PWM 模块 1 (EECP2)

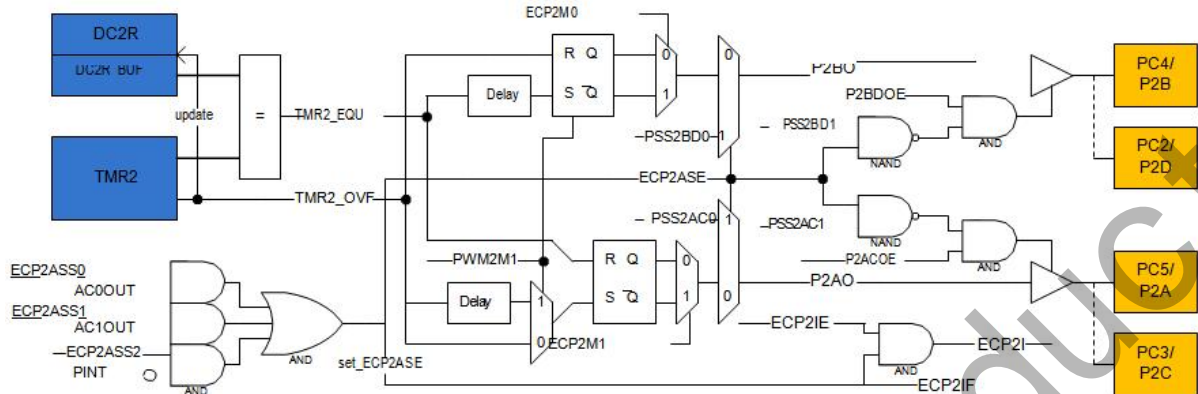
- 产生单路频率和占空比可调的 PWM 信号
- 产生 1 组带死区控制的频率和占空比可调的 PWM 半桥驱动
- 产生 1 组带死区控制的频率和占空比可调的 PWM 全桥驱动

#### 13.1. 综述

本章节所描述的增强型脉冲宽度调制模块 (简称为 ECP2 模块) 是一个用来产生 PWM 信号的外设。ECP2 模块基于定时器 2 来产生单路频率和占空比可调的 PWM 信号, 也可以产生 1 组带死区控制的频率和占空比可调的 PWM 半桥驱动和全桥驱动。

## 13.2. PWM 模式

ECP2 模块的结构图



### PWM 使能配置

ECP2 模块产生和输出 PWM 信号时，须设置 ECP2CR 寄存器中的 PWM2EN 位的值为“1”。当设置 PWM2EN 位为“0”时，ECP2 模块不会产生 PWM 信号，也不会输出 PWM 信号。

### PWM 模式配置

PWM 模式下，当设置位于 ECP2CR 寄存器的 PWM2M 位为 00 时，ECP2 模块只产生单路占空比和频率可调的 PWM 信号（用 T2PWM 来表示）；当设置 PWM2M 位为 10 时，可产生 1 组带死区控制的 PWM 半桥驱动信号（分别用 T2PWMP 和 T2PWWN 来表示）；当设置 PWM2M 位为 01 或 11 时，可产生 1 组带死区控制的 PWM 全桥驱动信号（分别用 T2PWMA, T2PWMB, T2PWMC 和 T2PWMD 来表示）。下面的描述中，用 T2PWMx 来表示所有 PWM 信号。

PWM 信号的极性是可配置的，通过设置位于 ECP2CR 寄存器的 ECP2M[1:0]位来选择。要使 PWM 信号输出到引脚上，须设置其相应的输出使能信号 P2ACOE 和 P2BDOE 位为 1 才会正常输出。

### PWM 周期

PWM 周期由定时器 2 的 PR2 寄存器决定。定时器 2 是一个八位计数器，与 TPR2 寄存器的低 4 位一起，构成一个 12 位的计数器，这就决定了 PWM 的周期由下面的公式来计算：

$$\text{PWM Period} = [(PR2) + 1] * 4 * T_{osc} * TMR2PR$$

其中，PR2 为计数最大值，Tosc 为定时器 2 的计数时钟，TRM2PR 为预分频系数。

当 TMR2 等于 PR2 时，下一个计数周期会发生以下三件事：

- TMR2 被清零
- T2PWM 被置位（例外：若 PWM 占空比为 0%，PWM 信号将不会被置位）
- PWM 占空比值由 DC2RH:DC2RL 锁存到其相应的缓存器中（DC2LM=1 时）

### PWM 占空比

PWM 的占空比由 10 位寄存器 DC2RH:DC2RL 的值来决定，当设置位于 ECP2CR 寄存器的 DC2LM 位为 0 时，软件要更新占空比值，可直接写入此 10 位寄存器中，需要注意写入的时机和次序，应先写占空比高两位的值到 DC2RH 中，再写低八位的值到 DC2RL 中；当 DC2LM 位为 1 时，软件可把将要更新的占空比值随时写入此 10 位寄存器中，在定时器 2 发生计数溢出时，DC2RH:DC2RL 的值会锁存到其相应的缓存器中，缓存器中的值即为 PWM 的最新占空比值。

PWM 占空比值由下面的公式来计算：

$$\text{DutyCycle2} = \text{DC2RH:DC2RL}$$

PWM 脉冲宽度由下面的公式来计算：

$$\text{Pulse Width} = (\text{DutyCycle2}) * T_{osc} * (\text{TMR2PR}) \text{ PWM}$$

占空比率由下面的公式来计算：

$$\text{Duty Cycle Ratio} = (\text{DutyCycle2}) / ((PR2+1)*4)$$

当 TRM2 和 DutyCycle2 发生匹配时，T2PWM 信号被清零。

### PWM 分辨率

对于给定的周期，分辨率决定了可能的占空比周期数。16 位分辨率有 65536 个连续的占空比周期，10 位分辨率有 1024 个连续的占空比周期，8 位分辨率有 256 个占空比周期。

当 PR2 被设置为 255 时，拥有最大的 PWM 分辨率，即 10 位。分辨率是 PR2 寄存器的函数，如下面公式所示：

$$\text{Resolution} = \log_2[(\text{PR2} + 1) * 4] / \log_2 \text{ bits}$$

下面两个表格为 PWM 频率和占空比的示例。

PWM 频率和分辨率 (Fosc = 20 MHz)

Frequency (KHz)	1. 22	4. 88	19. 53	78. 13	156. 25	312. 5
TMR2 Prescale	16	4	1	1	1	1
PR2 Value	0xFF	0xFF	0xFF	0x3F	0x1F	0x0F
Max Resolution	10	10	10	8	7	6

PWM 频率和分辨率 (Fosc = 8 MHz)

Frequency(KHz)	1. 23	4. 90	19. 61	76. 92	153. 85	333. 3
TMR2Prescale	16	4	1	1	1	1
PR2Value	0x65	0x65	0x65	0x19	0x0C	0x05
MaxResolution	8	8	8	6	5	4

### PWM 操作的配置

当配置 ECP2 模块工作在 PWM 模式时，可按照以下流程来执行：

1. 清零 P2ACOIE 或 P2BDOIE 位来禁止 PWM 引脚 (P2A 或 P2B 或 P2C 或 P2D) 的输出驱动；
2. 加载 PR2 寄存器来设置 PWM 周期；
3. 加载 DC2RH 寄存器和 DC2RL 寄存器来设置 PWM 占空比；
4. 加载 ECP2CR 寄存器来设置 PWM 使能，模式以及占空比的更新方式；
5. 配置和启动定时器 2，包括：
  - 清零位于 T1IFR2 寄存器的 T2IF 中断标志位
  - 加载位于 TCR2 寄存器的 T2CKPS 位来设置定时器 2 的预分频系数
  - 置位位于 TCR2 寄存器的 TMR2ON 位来使能定时器 2
6. 一个新的 PWM 周期开始后，使能 PWM 输出，即：
  - 等待定时器 2 溢出 (即位于 T1IFR2 寄存器的 T2IF 位置位)
  - 置位 P2ACOIE 或 P2BDOIE 位来使能 PWM 引脚 (P2A 或 P2B 或 P2C 或 P2D) 的输出驱动

### PWM 增强模式

PWM 增强模式可产生一路高达 10 位分辨率的 PWM 信号输出至 4 个不同的引脚上，可通过下面四种不同的 PWM 输出模式来实现：

- 单路 PWM
- 半桥 PWM
- 全桥 PWM，正向模式
- 全桥 PWM，反向模式

要选择 PWM 增强模式，位于 ECP1CR 寄存器的 PWM2M 位必须正确设置。

PWM 输出是和 IO 引脚复用的，并指定在 P2A，P2B，P2C 和 P2D 引脚上。PWM 的极性是可配置的，通过设置位于 ECP1CR 寄存器的 ECP2M 位来选择。

下面表格为 PWM 增强模式下引脚的分配情况。

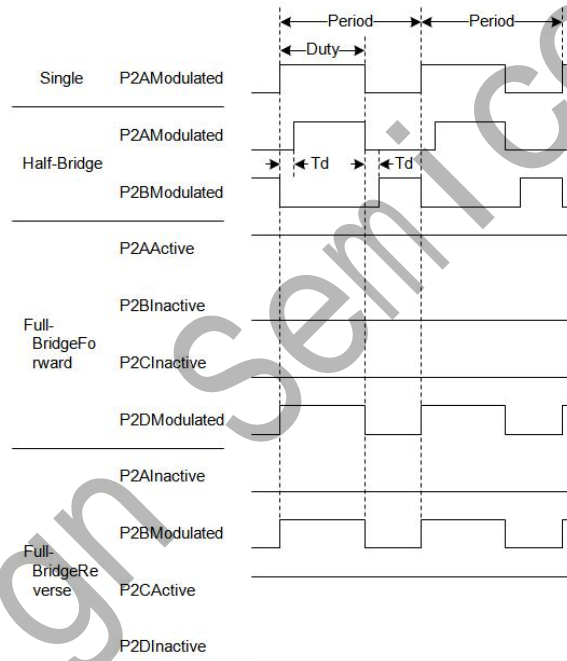
PWM 模式	PWM2M	P2A	P2B	P2C	P2D
单路	00	Yes	No	No	No

半桥	10	Yes	Yes	No	No
全桥正向	01	Yes	Yes	Yes	Yes
全桥反向	11	Yes	Yes	Yes	Yes

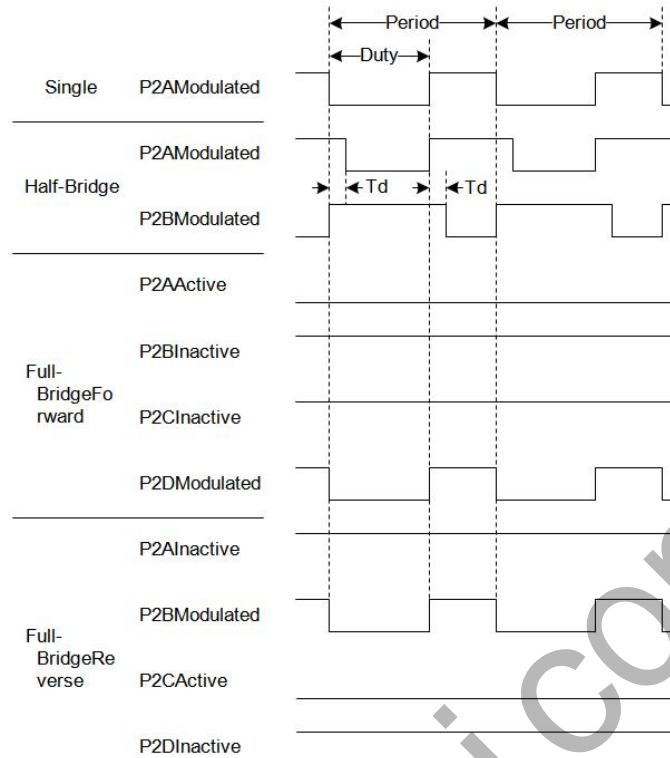
下面表格为 PWM 增强模式下引脚的有效电平状态

PWM 模式	PWM2M	ECP2M[1]=0	ECP2M[1]=1	ECP2M[0]=0	ECP2M[0]=1
单路	00	P2A 为高有效	P2A 为低有效	-	-
半桥	10	P2A 为高有效	P2A 为低有效	P2B 为高有效	P2B 为低有效
全桥正向	01	P2A/P2C 为高有效	P2A/P2C 为低有效	P2B/P2D 为高有效	P2B/P2D 为低有效
全桥反向	11	P2A/P2C 为高有效	P2A/P2C 为低有效	P2B/P2D 为高有效	P2B/P2D 为低有效

下图为 PWM 增强模式下的输出波形（高有效）



下图为 PWM 增强模式下的输出波形（低有效）



上面两图中，Period 为 PWM 的周期，Duty 为 PWM 的占空比，Td 为死区时间。

### PWM 自动关闭

PWM 模式支持自动关闭，当关闭事件发生时，将会禁止 PWM 输出。自动关闭模式可以放置 PWM 输出引脚至预设状态。这可以用来放置 PWM 损坏应用电路。

位于 ECP2ASR 寄存器的 ECP2ASS[2:0]位用来选择自动关闭源。自动关闭事件可由下面来产生：

- 比较器 0 的输出
- 比较器 1 的输出
- PINT 引脚上的“0”电平
- 软件设置 ECP2ASE 位

位于 ECP2ASR 寄存器的 ECP2ASE 位用来表明自动关闭的状态。当 ECP2ASE 位为“0”时，PWM 输出为正常状态。当 ECP2ASE 位为“1”时，PWM 输出位自动关闭状态。

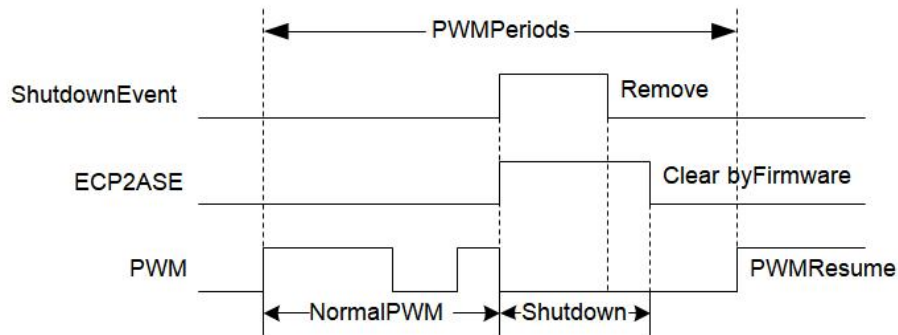
当自动关闭事件发生时，会发生下面两件事：

- 4) ECP2ASE 位被置位。ECP2ASE 位会保持为高，直到软件把它清零，或者发生自动重启；
- 5) ECP2IF 位被置位。ECP2 中断使能时，会产生 ECP2 中断。

6) PWM 引脚输出立刻被切换至自动关闭预设状态。PWM 输出引脚被分为 P2A/P2C 和 P2B/P2D 两组，每组的自动关闭状态可由位于 ECP2ASR 寄存器的 PSS2AC 和 PSS2BD 位来选择。每个引脚可切换至以下三种状态中的一种：

- 驱动为逻辑高电平
- 驱动为逻辑低电平
- 高阻状态

下图为自动关闭操作示例图



自动关闭条件是电平驱动信号，非边沿驱动信号。当电平保持不变时，自动关闭状态可持续。当自动关闭条件保持时，软件写 ECP2ASE 位为无效操作。

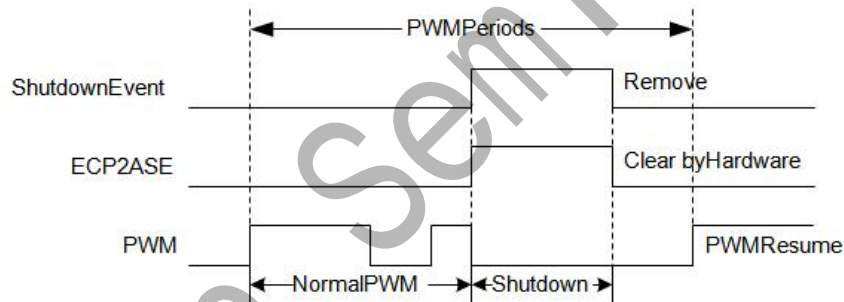
当自动关闭条件移除后，PWM 会重启（软件或自动重启），PWM 信号输出会在接下来的 PWM 周期开始重启。

#### PWM 自动重启

ECP2 模块可配置 PWM 信号自动重启，当自动关闭条件移除后，PWM 信号可自动重新启动输出。置位位于 PWM2CR 寄存器的 PRS2EN 位，即可使能自动重启功能。

自动重启被使能后，自动关闭条件仍有效时，ECP2ASE 位将会保持置位。当自动关闭条件移除时，ECP2ASE 位会被硬件清零，PWM 恢复常规操作。

下图为自动重启操作示例图



### 13.3. 半桥模式

半桥模式下，有两个输出引脚来驱动推拉负载。一路 PWM 信号由 P2A 引脚输出，另一路互补的 PWM 信号由 P2B 引脚输出。这个模式可用在半桥应用上，也可用在全桥应用上，其四个电源开关由两路 PWM 信号来调制。

半桥应用当中，所有的电源开关都通过 PWM 频率来调制。通常情况下，电源开关的关闭比打开需要更多的时间。如果上下电源开关在同一时间切换（一个打开，另一个关闭），两个开关可能会在短时间内同时处于打开状态，直到其中一个完全关闭。在这段短时间内，两个开关之间会通过一段很大的电流（直通电流），造成半桥短路。为了避免在切换过程中产生这种潜在的破坏性直通电流，通常在每个开关打开前加上一段延时，以保证另一个开关已经完全关闭。

半桥模式下，可编程的死区延时用来防止在半桥器件之间产生直通电流。死区延时发生在信号由非有效状态切换至有效状态期间。位于 PWM2CR 寄存器的 DT2[6:0] 位，用来设置输出驱动有效前的 PWM 周期数。若 DT2 的值大于占空比的周期数，相应的输出在整个 PWM 周期内都会保持无效。

### 13.4. 全桥模式

全桥模式下，4 个引脚都被用作输出。

正向模式下，P2A 引脚被驱动至有效状态，P2D 引脚被调制，而 P2B 和 P2C 会被驱动至无效状态。

反向模式下，P2C 引脚被驱动至有效状态，P2B 引脚被调制，而 P2A 和 P2D 会被驱动至无效状态。

#### 全桥模式下方向的改变

全桥模式下，位于 ECP2CR 寄存器的 PWM2M[1]位被用来控制正向和反向模式。当软件改变 这个方向控制位的值后，方向模式会在接下来的 PWM 周期改变。

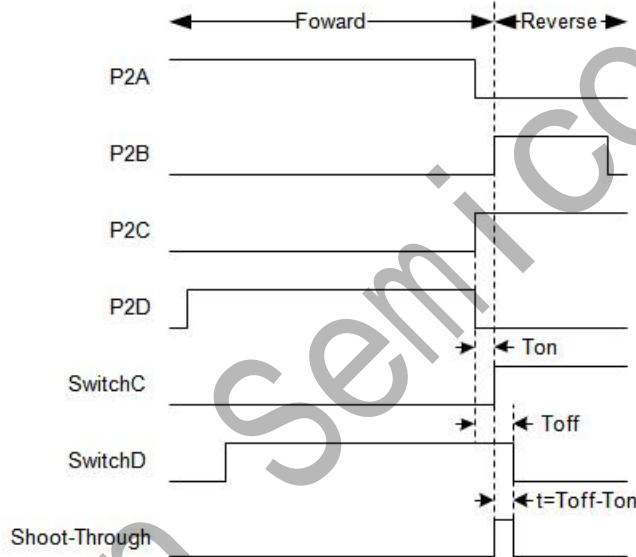
软件改变位于 ECP2CR 寄存器的 PWM2M[1]位后，方向改变就会启动。在当前 PWM 周期溢出 前的 4 个定时器 2 周期内，将会发生下面的操作：

- 调制输出（P2B 或 P2D）会变成无效状态；
- 相应的未被调制输出（P2C 或 P2A）会切换至相反状态；
- PWM 调制会在接下来的周期开始时恢复。

全桥模式下，不会产生死区延时。因在一个时刻，只有一路输出是被调制的，通常情况下死 区延时是不需要的。当下面两个条件同时存在时，需要死区延时：

- 1) 当输出的占空比为 100%或接近 100%，PWM 输出方向发生改变；
- 2) 电源开关包括电源器件和驱动电路中，关闭时间大于开启时间。

下图为占空比接近 100%，PWM 的方向由正向改变为反向时的实例。在  $t_1$  时刻，P2A 和 P2D 输出变成无效，而 P2C 输出变成有效。因关闭时间大于开启时间，电源器件 C 和 D 之间会产生直通电流，持续时间为  $t$ 。当 PWM 的方向由反向改变为正向时，会在器件 A 和 B 之间发生同样的现象。



如果实际应用当中需要在高占空比时改变 PWM 的方向，有以下两个可能的解决方案来消除 直通电流：

- 1) 在改变方向前一个 PWM 周期减小 PWM 占空比；
- 2) 使用关闭速度快于开启速度的开关驱动。或许还有其它防止直通电流的有效方法存在。

### 13.5. 启动事项

无论使用哪种 PWM 模式，硬件须在 PWM 输出引脚上使用合适的外部上拉或下拉电路。当 控制器复位被释放后，所有的 IO 引脚都处于高阻状态。外部电路须保持电源开关器件处于 OFF 状态，直到控制器驱动 IO 引脚至合适的电平状态。

位于 ECP2CR 寄存器的 ECP2M[1:0]位可用来选择 PWM 输出信号的有效驱动电平极性。在 PWM 引脚输出驱动使能之前，PWM 的输出极性须先设定好。在 PWM 引脚输出驱动使能后，不建 议再改变极性配置。

PWM 模块被初始化之后，P2A, P2B, P2C 和 P2D 输出有可能不在合适的状态。此时使能 PWM 引脚输出有可能会对应应用电路造成损坏。因此建议在一个完整的 PWM 周期后再使能 PWM 输出驱动。当位于 T1FR2 寄存器的 T21F 位被置位后，表明当前 PWM 周期结束，下一个 PWM 周期开始。

### 13.6. 寄存器定义

ECP2CR0 - ECP2 控制寄存器 0

ECP2CR0 - ECP2 控制寄存器 0								
地址: 0xB4					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	P2_PH3	P2_OE3	P2_PH2	P2_OE2	P2_PH1N	P2_OE1N	P2_PH1P	P2_OE1P
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	P2_PH3	ECP2 通道 3 单路 PWM 输出极性选择位						
6	P2_OE3	ECP2 通道 3 单路 PWM 输出使能位						
5	P2_PH2	ECP2 通道 2 单路 PWM 输出极性选择位						
4	P2_OE2	ECP2 通道 2 单路 PWM 输出使能位						
3	P2_PH1N	ECP2 通道 1N 路 PWM 输出极性选择位						
2	P2_OE1N	ECP2 通道 1N 路 PWM 输出使能位, ECP2_OE1P 和 ECP2_OE1N 同时为 1 时, 组成互补输出, 否则为单路输出						
1	P2_PH1P	ECP2 通道 1P 路 PWM 输出极性选择位						
0	P2_OE1P	ECP2 通道 1P 路 PWM 输出使能位, ECP2_OE1P 和 ECP2_OE1N 同时为 1 时, 组成互补输出, 否则为单路输出						

ECP2CR1 - ECP2 控制寄存器 1

ECP2CR1 - ECP2 控制寄存器 1								
地址: 0xB5					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	ASE	ASS2	ASS1	ASS0	PSN1	PSN0	PSP1	PSP0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	ASE	ECP2 通道 1PWM 输出自动关闭标志位, 当自动关闭触发后, ECP1_ASE 位会被置位, PWM 输出自动关闭, 引脚状态由 ECP1_PSP 和 ECP1_PSN 决定						
6	ASS2	ECP2 通道 1PWM 输出自动关闭触发源 2 使能位, 当设置为 1 时, 外部引脚 PINT 为低电平时会关闭通道 1 的 PWM 输出						
5	ASS1	ECP2 通道 1PWM 输出自动关闭触发源 1 使能位, 当设置为 1 时, 比较器 1 输出滤波后为高电平时会关闭通道 1 的 PWM 输出						
4	ASS0	ECP2 通道 1PWM 输出自动关闭触发源 0 使能位, 当设置为 1 时, 比较器 0 输出滤波后为高电平时会关闭通道 1 的 PWM 输出						
3	PSN1	ECP2 通道 1PWM 输出自动关闭下 1N 路的输出禁止位, 当设置为 1 时, 自动关闭发生后, 1N 路会被禁止输出, 引脚状态由 GPIO 控制						

2	PSN0	ECP2 通道 1PWM 输出自动关闭下 1N 路的输出状态选择位，当设置 ECP2_PSN1 为 0 时，自动关闭发生后，1N 路引脚状态由 ECP2_PSN0 来选择
1	PSP1	ECP2 通道 1PWM 输出自动关闭下 1P 路的输出禁止位，当设置为 1 时，自动关闭发生后，1P 路会被禁止输出，引脚状态由 GPIO 控制
0	PSP0	ECP2 通道 1PWM 输出自动关闭下 1P 路的输出状态选择位，当设置 ECP2_PSP1 为 0 时，自动关闭发生后，1P 路引脚状态由 ECP2_PSP0 来选择

**ECP2CR2 - ECP2 控制寄存器 2**

ECP2CR2 - ECP2 控制寄存器 2								
地址: 0xB6					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	ARE	DC3LM	DC2LM	DC1LM	-	-	-	-
R/W	W/R	W/R	W/R	W/R	-	-	-	-
Bit	Name	描述						
7	ARE	ECP2 通道 1PWM 输出自动重启使能位，当设置为 1 时，自动关闭触发条件去除后，ECP2_ASE 会被清零，通道 1 的 PWM 输出恢复						
6	DC3LM	ECP2 通道 3PWM 占空比加载方式选择位，当设置为 0 时，软件写入占空比寄存器的值直接更新至占空比缓存器中并立即生效；当设置为 1 时，定时器 2 溢出后占空比寄存器中的值才更新至占空比缓冲器中并生效						
5	DC2LM	ECP2 通道 2PWM 占空比加载方式选择位，当设置为 0 时，软件写入占空比寄存器的值直接更新至占空比缓存器中并立即生效；当设置为 1 时，定时器 2 溢出后占空比寄存器中的值才更新至占空比缓冲器中并生效						
4	DC1LM	ECP2 通道 1PWM 占空比加载方式选择位，当设置为 0 时，软件写入占空比寄存器的值直接更新至占空比缓存器中并立即生效；当设置为 1 时，定时器 2 溢出后占空比寄存器中的值才更新至占空比缓冲器中并生效						
3:0	-	保留						

**ECP2DTP - ECP2 上管死区时间**

ECP2DTP - 寄存器								
地址: 0x76					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
R/W	W/R							
Bit	Name	描述						
7:0	-	P-Channel Dead Time Select for EC2P2						

**ECP2CR3 - ECP2 控制寄存器 3**

ECP2CR3 - ECP2 控制寄存器 3								
地址: 0xB7					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0



Name	CP3EN	CP2EN	CP3M1	CP3M0	CP2M1	CP2M0	CP3S	CP2S
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	CP3EN	ECP2 通道 3 输入捕获使能位, 当设置为 1 且 ECP2_OE3 设置为 0 时, 通道 3 输入捕获被使能, 发生捕获时 TMR2 寄存器的值会记录到 ECP2R3 中						
6	CP2EN	ECP2 通道 2 输入捕获使能位, 当设置为 1 且 ECP2_OE2 设置为 0 时, 通道 3 输入捕获被使能, 发生捕获时 TMR2 寄存器的值会记录到 ECP2R2 中						
5	CP3M1	ECP2 通道 3 输入捕获模式选择位, 00=双沿捕获; 01=正沿捕获; 10=每两个正沿捕获; 11=每四个正沿捕获						
4	CP3M0							
3	CP2M1	ECP2 通道 2 输入捕获模式选择位, 00=双沿捕获; 01=正沿捕获; 10=每两个正沿捕获; 11=每四个正沿捕获						
2	CP2M0							
1	CP3S	ECP2 通道 3 输入捕获源选择位, 0=外部引脚 PE1 输入; 1=外部引脚 DM1_RXD 输入						
0	CP2S	ECP2 通道 2 输入捕获源选择位, 0=外部引脚 PB2 输入; 1=外部引脚 DP1_RXD 输入						

ECP2R1L - ECP2 通道 1 寄存器低字节

ECP2R1L - 寄存器								
地址: 0xA8					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
R/W	W/R							
Bit	Name	描述						
7:0	-	ECP2 通道 1 占空比寄存器低字节						

ECP2R1H - ECP2 通道 1 寄存器高字节

ECP2R1H - 寄存器								
地址: 0xA9					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	B15	B14	B13	B12	B11	B10	B9	B8
R/W	W/R							
Bit	Name	描述						
7:0	-	ECP2 通道 1 占空比寄存器高字节						

ECP2R2L - ECP2 通道 2 寄存器低字节

ECP2R2L - 寄存器								
地址: 0xAA					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
R/W	W/R							
Bit	Name	描述						
7:0	-	ECP2 通道 2 占空比寄存器低字节或通道 2 输入捕获寄存器低字节						

ECP2R2H - ECP2 通道 2 寄存器高字节

ECP2R2H - 寄存器								
地址: 0xAB					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0



Name	B15	B14	B13	B12	B11	B10	B9	B8
R/W	W/R							
Bit	Name	描述						
7:0	-	ECP2 通道 2 占空比寄存器或通道 2 输入捕获寄存器高字节						

ECP2R3L - ECP2 通道 3 寄存器低字节

ECP2R3L - 寄存器								
地址: 0xAE					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
R/W	W/R							
Bit	Name	描述						
7:0	-	ECP2 通道 3 占空比寄存器低字节或通道 3 输入捕获寄存器低字节						

ECP2R3H - ECP2 通道 3 寄存器高字节

ECP2R3H - 寄存器								
地址: 0xAF					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	B15	B14	B13	B12	B11	B10	B9	B8
R/W	W/R							
Bit	Name	描述						
7:0	-	ECP2 通道 3 占空比寄存器或通道 3 输入捕获寄存器高字节						

ECP2DTN - ECP2 下管死区时间

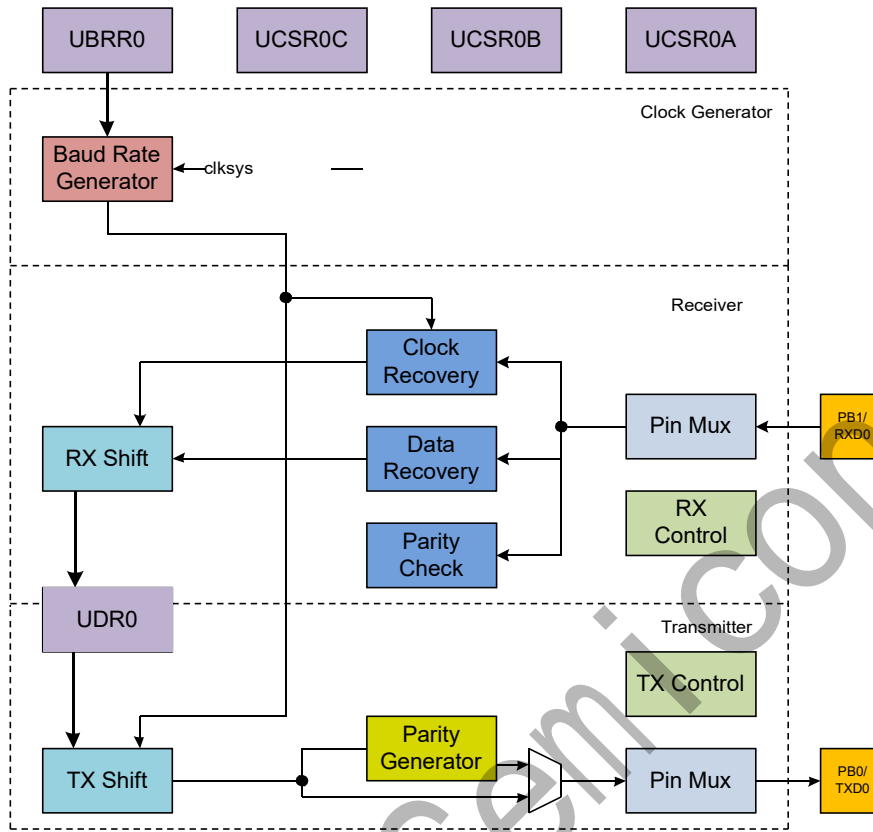
ECP2DTN - 寄存器								
地址: 0x77					默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0
Name	B7	B6	B5	B4	B3	B2	B1	B0
R/W	W/R							
Bit	Name	描述						
7:0	-	N-Channel Dead Time Select for ECPC2						

## 14. 通用异步串行收发器 0 (UART0)

- 全双工操作 (独立的串行接收和发送寄存器)
- 主机或从机操作
- 高精度的波特率发生器
- 支持 5, 6, 7, 8, 或 9 个数据位和 1, 或 2 个停止位
- 硬件支持的奇偶产生和校验机制
- 数据过速检测
- 帧错误检测
- 噪声滤波, 包括错误的起始位检测以及数字低通滤波器
- 三个独立的中断, 包括: 发送结束中断, 发送数据寄存器空中断, 以及接收结束 中断
- 多处理器通信模式
- 倍速异步通信模式

### 14.1. 综述

UART0 主要包括三个部分：时钟发生器，发送器和接收器。如下图所示：



UART0 结构图

时钟发生器根据波特率来产生发送和接收所需的时钟。发送器包括一个写数据缓冲器，串行移位寄存器，奇偶发生器以及处理不同帧格式所需的控制逻辑。写数据缓冲器允许连续发送数据而不会在数据帧之间引入延迟。接收器具有时钟和数据恢复单元，用于异步数据的接收。

除了恢复单元，接收器还包括奇偶校验，控制逻辑，串行移位寄存器和一个两级接收缓冲器 UDR0。接收器支持与发送器相同的帧格式，而且可以检测帧错误，数据过速和奇偶校验错误。

时钟产生逻辑为发送器和接收器产生基础时钟。UART0 支持 4 种模式的时钟：正常的异步模式，倍速的异步模式。USCR0A 的 U2X0 位控制异步模式下的倍速使能。

### 14.2. 波特率发生器

波特率寄存器 UBRR0 和降序计数器连接在一起作为 UART0 的可编程的预分频器或波特率发生器。降序计数器工作在系统时钟 ( $f_{sys}$ ) 下，当其计数到零或 UBRR0L 寄存器被写时，会自动加载 UBRR0 寄存器的值。当计数到零时产生一个时钟，该时钟作为波特率发生器的输出时钟，频率为  $f_{sys}/(UBRR0+1)$ 。

下表给出了各种工作模式下计算波特率（位/秒）以及 UBRR0 值的公式。

工作模式	波特率计算公式 <sup>(1)</sup>	UBRR0 值计算公式
异步正常模式	$BAUD = f_{sys}/(16*(UBRR0+1))$	$UBRR0 = f_{sys}/(16*BAUD) - 1$
异步倍速模式	$BAUD = f_{sys}/(8*(UBRR0+1))$	$UBRR0 = f_{sys}/(8*BAUD) - 1$

说明：

1. 波特率定义为每秒的位传输速度 (bps) ；

2. BUAD 为波特率,  $f_{sys}$  为系统时钟, UBRR0 为波特率寄存器 UBRR0H 和 UBRR0L 的组合值。

### 14.3. 倍速工作模式

通过设定 UCSROA 寄存器的 U2X0 位可以是传输速率加倍, 该位只在异步工作模式下有效, 同步工作模式下置该位为“0”。设置该位将会把波特率分频器的分频值减半, 有效地加倍异步通信的传输速率。在这种情况下, 接收器只使用一半的采样数来对数据进行采样及时钟恢复, 因此需要更精准的波特率设置和系统时钟。发送器则没有变化。

### 14.4. 帧格式

一个串行数据帧由数据字加上同步位(起始位和停止位)以及用于纠错的奇偶校验位构成。UART0 接受以下 30 种组合的数据帧格式:

- 1 个起始位
- 5、6、7、8 或 9 个数据位
- 无校验位、奇校验位或偶校验位
- 1 或 2 个停止位 数据帧以起始位开始, 紧接着是数据字的最低位, 接着是其它数据位, 以数据字的最高位结束, 最多成功传输 9 位数据。如果使能了校验, 校验位将紧接着数据字, 最后是停止位。当一个完整的数据帧传输后, 可以立即传输下一个新的数据帧, 或者使传输线处于空闲(高电平)状态。下图为可能的数据帧结构, 方括号中的位是可选的。



Figure3 UART0 帧结构图

说明:

- 1) IDLE 通信线 (RXD 或 TXD) 上没有数据传输, 线路空闲时必须为高电平
- 2) St 起始位, 总是为低电平
- 3) 0-8 数据位
- 4) P 校验位, 奇校验或偶校验
- 5) Sp 停止位, 总是为高电平

数据帧的结构由 UCSROB 和 UCSROC 寄存器中的 UCSZ0[2:0]、UPM0[1:0]和 USBS0 设定。接收与发送使用相同的设置。设置的任何改变都可能破坏正在进行的数据传输。其中, UCSZ0[2:0] 确定了数据帧的数据位数, UPM0[1:0]用于使能和确定校验的类型, USBS0 设置帧有一位或两位结束位。接收器会忽略第二个停止位, 因此帧错误只在第一个结束位为“0”时被检测到。

### 14.5. 校验位计算

校验位的计算是对数据的各个位进行异或运算。如果选择了奇校验, 则异或结果还需要取反。校验位与数据位的关系如下:

$$\begin{aligned} P_{even} &= d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 0 \\ P_{odd} &= d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 1 \end{aligned}$$

说明:

- 1)  $P_{even}$  偶校验结果
- 2)  $P_{odd}$  奇校验结果
- 3)  $d_n$  第 n 个数据位

### 14.6. UART0 初始化

进行通信之前首先要对 UART0 进行初始化。初始化过程通常包括波特率的设定, 帧结构的设定, 以及根据需要使能接收器或发送器。对于中断驱动的 UART0 操作, 在初始化时要清零全局中断标志并禁止 UART0 的所有中断。在进行重新初始化比如改变波特率或帧结构时, 必须确保没有数据传输。TXC0 标志位可以用来检测发送器是否完成了所有传输, RXC0 标志位可以用来检测接收缓冲器中是否还有数据未被读出。如果 TXC0 标志位用作此用途, 在每次发送数据之前(写 UDRO 寄存器之前) 必须清零 TXC0 标志位。

## 14.7. 发送器

置位 UCSROB 寄存器的 TXEN0 位将使能 UART0 的数据发送。使能后 TXD 引脚的通用 IO 功能即被 UART0 功能所取代，成为发送器的串行输出。发送数据之前要设置好波特率、工作模式与帧格式。

### 发送 5 到 8 为数据的帧

将需要发送的数据加载到发送缓冲器中来启动数据发送。CPU 通过写 UDRO 寄存器来加载数

据。当发送移位寄存器可以发送新一帧数据的时候，缓冲器中的数据将转移到移位寄存器中。当移位寄存器处于空闲状态（没有正在进行的数据传输），或者前一帧数据的最后一个停止位发送完毕，它将加载新的数据。一旦移位寄存器加载了新的数据，它将按照既定的设置传输一个完整的帧。

### 发送 9 位数据的帧

如果发送 9 位数据的帧，应先将数据的第 9 位写入寄存器 UCSROB 的 TXB08 位，然后再将低 8 位数据写入发送数据寄存器 UDRO。第 9 位数据在 multi 通信中用于表示地址帧，在同步通信中可用于协议处理。

### 发送奇偶校验位

奇偶校验产生电路为串行数据帧生成相应的校验位。当校验位使能时（UPM01 = 1），发送控制逻辑电路会在数据字的最后一位与第一个停止位之间插入奇偶校验位。

### 发送标志位与中断处理

UART0 发送器有两个标志位：UART0 数据寄存器空标志 UDRE0 和传输结束标志 TXC0，两个标志位都可以产生中断。

数据寄存器空标志 UDRE0 用来表示发送缓冲器是否可以写入一个新的数据。该位在发送缓冲器空时被置“1”，满时被置“0”。当 UDRE0 位为“1”时，CPU 可以往数据寄存器 UDRO 写入新的数据，反之则不能。

当 UCSROB 寄存器中的数据寄存器空中断使能位 UDRIE0 为“1”时，只要 UDRE0 被置位（且全局中断使能），就将产生 UART0 数据寄存器空中断请求。对寄存器 UDRO 执行写操作将清零 UDRE0。当采用中断方式传输数据时，在数据寄存器空中断服务程序中必须写入一个新的数据到 UDRO 以清零 UDRE0，或者是禁止数据寄存器空中断。否则一旦该中断服务程序结束，一个新的中断将再次产生。当整个数据帧被移出发送移位寄存器，同时发送寄存器中又没有新的数据时，发送结束标志 TXC0 将被置位。当 UCSROB 上的发送结束中断使能位 TXCIE0（且全局中断使能）置“1”时，随着 TXC0 标志位被置位，UART0 发送结束中断将被执行。一旦进入中断服务程序，TXC0 标志位即被自动清零，CPU 也可以对该位写“1”来清零。

### 禁止发送器

当 TXEN0 清零后，只有等所有的数据都发送完成以后发送器才能真正禁止，即发送移位寄存器与发送缓冲寄存器中都没有要传送的数据。发送器禁止以后，TXD 引脚恢复其通用 IO 功能。

## 14.8. 接收器

置位 UCSROB 寄存器的接收允许位 (RXEN0) 即可启动 UART0 接收器。使能后 RXD 引脚的通用 IO 功能被 UART0 功能所取代，成为接收器的串行输入口。进行数据接收之前首先要设置好波特率、操作模式及帧格式。如果使用同步接收模式，XCK 引脚上的时钟被用为传输时钟。

### 接收 5 到 8 位数据的帧

一旦接收器检测到一个有效的起始位，便开始接受数据。起始位后的每一位数据都将以所设定的波特率来进行接收，直到收到一帧数据的第一个停止位，第二个停止位会被接收器忽略。

接收到的每一位数据被送入接收移位寄存器，收到第一个停止位以后，接收器置位于 UCSROA 寄存器的接收数据完成标志 RXC0 位，并把移位寄存器中完整的数据帧转移到接收缓冲器中，CPU 通过读取 UDRO 寄存器就可以获得接收到的数据。

### 接收 9 位数据的帧

如果设定了 9 位数据的数据帧，在从 UDRO 读取低 8 位数据之前必须首先读取寄存器 UCSROB 的 RXB08 位来获得第 9 位数据。这个规则同样适用于状态标志位 FE0、DOR0 以及 PE0。读取 UDRO 存储单元会改变接收缓冲器的状态，进而改变同样存储于缓冲器中的 TXB08、FE0、DOR0 及 PE0 位。

### 接收结束标志及中断处理

UART0 接收器有一个标志位：接收结束标志 RXC0，用来表明接收缓冲器中是否有未被读出的数据。当接收缓冲器中有未被读出的数据时，此位为“1”，反之为“0”。如果接收器被禁止，接收缓冲器会被刷新，RXC0 也会被清零。

置位 UCSROB 的接收结束中断使能位 RXCIE0 后，只要 RXC0 标志被置位（且全局中断被使能），就会产生 UART0 接收结束中断。使用中断方式进行数据接收时，数据接收结束中断服务程序必须从 UDRO 读取数据来清零 RXC0 标志，否则只要中断处理程序一结束，一个

新的中断 就会产生。

### 接收错误标志

UART0 接收器有三个错误标志：帧错误 FE0、数据溢出 DOR0 及奇偶校验错误 PE0。它们都 位于 UCSR0A 寄存器。错误标志与数据帧一起保存在接收缓冲器当中。所有的错误标志都不 能产生中断。

帧错误标志 FE0 表明存储在接收缓冲器中的下一个可读帧的第一个停止位的状态。停止位正 确（值为“1”）则 FE0 标志为“0”，否则 FE0 标志为“1”。这个标志可用于检测同步丢失、传输 中断，也可用于协议处理。

数据溢出标志 DOR0 表明由于接收缓冲器满造成了数据丢失。当接收缓冲器为满，接收移位 寄存器中已有数据，若此时检测到一个新的起始位，数据溢出就产生了。DOR0 标志被置位 即表明在最近一次读取 UDRO 和下一次读取 UDRO 之间丢失了一个或多个数据帧。当数据 帧 成功地从移位寄存器转入接收缓冲器后，DOR0 标志被清零。

奇偶校验错误标志 PE0 表明接收缓冲器中的下一帧数据在接收时有奇偶错误。如果不使能奇 偶校验，UPM01 被清零。

### 禁止接收器

与发送器相比，禁止接收器即刻起作用。正在接收的数据将丢失。禁止接收器（RXEN0 清零）后，接收器将不再占用 RXD 引脚，接收缓冲器也会被刷新。

### 奇偶校验器

置位奇偶校验模式位 UPM01 将启动奇偶校验器。校验的模式（偶校验或奇校验）由 UPM00 决定。奇偶校验使能后，校验器将计算输入数据的奇偶并把结果与数据帧的奇偶位进行比较。校验结果将与数据和停止位一起存储在接收缓冲器中。CPU 通过读取 PE0 位来检查接收的帧当中是否有奇偶错误。如果下一个从接收缓冲器中读出的数据有奇偶错误，并且奇偶校验使 能，则 PE0 被置位，一直有效到接收缓冲器 UDRO 被读取。

## 14.9. 异步数据接收

UART0 有一个时钟恢复单元和数据恢复单元来处理异步数据接收。时钟恢复逻辑用于同步从 RXD 引脚输入的异步串行数据和内部的波特率时钟。数据恢复逻辑用于采集数据，并通过低 通滤波器过滤所输入的每一位数据，从而提高接收器的抗干扰性能。异步接收的工作范围依赖于内部波特率时钟的精度、帧输入的速率及一帧所包含的数据位数。

### 异步工作范围

接收器的工作范围依赖于接收到的数据速率与内部波特率之间的不匹配程度。如果发送器以 过快或过慢的比特率传输数据，或者接收器内部产生的波特率没有相同的频率，那么接收器 就无法与起始位同步。为了确保接收器不会错过下一帧起始位的采样，数据输入速率和内部 接收器波特率不能相差太大，用它们之间的比值来描述波特率的误差范围。下面两个表格分 别给出了普通模式下和倍速模式下容许的最大波特率误差范围。

Table 2 普通模式下最大接收器波特率误差范围

数据位+奇偶位长度和	最大误差范围 (%)	推荐误差范围 (%)
5	+6.7/-6.8	±3.0
6	+5.8/-5.9	±2.5
7	+5.1/-5.2	±2.0
8	+4.6/-4.5	±3.0
9	+4.1/-4.2	±1.5
10	+3.8/-3.8	±1.5

Table 3 倍速模式下最大接收器波特率误差范围

数据位+奇偶位长度和	最大误差范围 (%)	推荐误差范围 (%)
5	+5.7/-5.9	±2.5
6	+4.9/-5.1	±2.0
7	+4.4/-4.5	±1.5
8	+3.9/-4.0	±1.5
9	+3.5/-3.6	±1.0
10	+3.2/-3.3	±1.0

从表中可以看出，普通模式下波特率允许有更大的变化范围。上述推荐的波特率误差范围是假定接收器和发送器对最大总误差具有同等贡献的前提下得出的。产生接收器波特率误差的可能原因有两个。首先，接收器系统时钟的稳定性与工作电压和温度有关。使用晶振来产生

系统时钟时一般不会有此问题，但使用内部振荡器时，系统时钟可能会有偏差。第二个原因是波特率发生器不一定能通过对系统时钟的分频来得到恰好想要的波特率。此时可以调整 UBRR0 的值，使得误差低至可以接受。

#### 14.10. 波特率设置及引入误差

对于标准晶振及谐振器频率来说，异步模式下的实际通信的波特率可通过波特率计算公式来获得，它与常用通信波特率之间的误差可用如下公式来计算：

$$\text{Error} [\%] = (\text{BaUDRE0aI} / \text{Baud} - 1) * 100\%$$

其中，Baud 为常用的通信波特率，BaUDRE0aI 为通过计算公式算出来的波特率，带入波特率计算公式即可得到波特率误差与系统时钟 f<sub>sys</sub> 和波特率寄存器 UBRR0 值之间的关系如下：

普通模式：

$$\text{Error} [\%] = (\text{f}_{\text{sys}} / (16 * (\text{UBRR0} + 1)) / \text{Baud} - 1) * 100\%$$

倍速模式：

$$\text{Error} [\%] = (\text{f}_{\text{sys}} / (8 * (\text{UBRR0} + 1)) / \text{Baud} - 1) * 100\%$$

当不考虑通信两边的时钟误差，即系统时钟 f<sub>sys</sub> 为标准时钟时，即可得到波特率误差 UBRR0 值之间的关系。下表即为 16MHz 系统时钟下不同 UBRR0 值设置下的波特率误差。

Table 4 16MHz 系统时钟下设置UBRR0 值所产生的误差

波特率 (bps)	f <sub>sys</sub> = 16.000MHz			
	普通模式 (U2X0 = 0)		倍速模式 (U2X0 = 1)	
	UBRR0	误差	UBRR0	误差
2400	416	-0.1%	832	0.0%
4800	207	0.2%	416	-0.1%
9600	103	0.2%	207	0.2%
14.4K	68	0.6%	138	-0.1%
19.2K	51	0.2%	103	0.2%
28.8K	34	-0.8%	68	0.6%
38.4K	25	2.1%	34	-0.8%
57.6K	16	0.2%	51	0.2%
76.8K	12	0.2%	25	0.2%
115.2K	8	-3.5%	16	2.1%
230.4K	3	8.5%	8	-3.5%
250K	3	0%	7	0%
0.5M	1	0%	3	0%
1M	0	0%	1	0%

#### 14.11. 多处理器通信模式

置位 UCSROA 的多处理器通信模式 (MPCMO) 位可以对 UART0 接收器接收到的数据帧进行过滤。那些没有地址信息的帧将被忽略，也不会存入接收缓冲器。在一个多处理器系统中，各处理器通过相同的串行总线进行通信，这种过滤有效的减少了需要 CPU 处理的数据帧的数量。MPCMO 位的设置不影响发送器的工作，但在多处理器通信的系统中，它的使用方法会有所不同。

如果接收器所接收的数据帧长度为 5 到 8 位，那么第一个停止位会用来表示当前帧包含的

是数据还是地址信息。如果接收器所接收的数据帧长度是 9 位，那么由第 9 位来确定是数据还是地址信息。如果帧类型标志位为

“1”，那么这是地址帧，否则为数据帧。

在多处理器通信模式下，允许多个从处理器从一个主处理器接收数据。首先要通过解码地址帧来确定所寻址的是哪一个从处理器。被寻址的从处理器将正常接收后续的数据，而其他的从处理器则会忽略这些数据帧直到接收到下一个地址帧。

对于一个作为主机的处理器来说，它可以使用 9 位数据帧格式，并用第 9 位数据来标识帧

格式。在这种通信模式下，从处理器也必须工作于 9 位数据帧格式。下面即为多处理器通信模式下进行数据交换的步骤：

1. 所有从处理器都工作于多处理器通信模式（置位 MPCMO）；
2. 主处理器发送地址帧，所有从处理器都接收此帧。从处理器 UCSROA 寄存器的 RXCO 位正常置位；
3. 每个从处理器都读取 UDRO 寄存器的内容，解码地址帧来确定是否被选中。如果选中，就清零 UCSROA 寄存器的 MPCMO 位，未被选中，则保持 MPCMO 为“1”并等待下一个地址帧的到来；
4. 被寻址的从处理器接收所有的数据帧，直到收到一个新的地址帧。未被寻址的从处理器忽略这些数据帧；
5. 被寻址的从处理器收到最后一个数据帧后，置位 MPCMO 位，并等待下一个地址帧的到来。然后从第二步重复进行。

使用 5 到 8 位数据的帧格式是可以的，但是不切实际，因为接收器必须在使用 n 和 n+1 帧格式之间进行切换。由于接收器和发送器使用相同的字符长度设置，这种设置使得全双工操作变得很困难。如果使用 5 到 8 位数据的帧格式，发送器应该设置两个停止位，其中第一个停止位被用于判断帧类型。

## 14.12. UART0 寄存器

UART0 寄存器列表

寄存器	地址	默认值	描述
UCSROA	0xC0	0x20	UART0 控制和状态寄存器A
UCSROB	0xC1	0x00	UART0 控制和状态寄存器B
UCSROC	0xC2	0x06	UART0 控制和状态寄存器 C
UBRR0L	0xC4	0x0	UART0 波特率寄存器低字节
UBRR0H	0xC5	0x0	UART0 波特率寄存器高字节
UDRO	0xC6	0x0	UART0 数据寄存器

UCSROA - UART0 控制和状态寄存器 A

UCSROA - UART0 控制和状态寄存器A								
地址: 0xC0				默认值: 0x20				
Bit	7	6	5	4	3	2	1	0
	RXC0	TXC0	UDRE0	RE	DOR0	PE0	U2X0	MPCMO
R/W	R	R/W	R	R	R	R	R/W	R/W
Bit	Name	描述						
7	RXC0	接收结束标志位。 当RXC0的值为“1”时，表明接收缓冲器中有未读出的数据。当RXC0的值为“0”时，表明接收缓冲器中没有未读出的数据。接收器禁止时，接收缓冲器被刷新，导致RXC0被清零。当接收结束中断使能位RXCIE0为“1”时，RXC0可用来产生接收结束中断。						
6	TXC0	发送结束标志位。 发送移位寄存器中的数据被送出，且发送缓冲器为空时 TXC0 置位。执行发送结束中断时 TXC0 自动清零，也可以通过对 TXC0 写“1”来进行清零。当发送结束中断使能位 TXCIE0 为“1”时，TXC0 可用来产生发送结束中断。						

5	UDRE0	数据寄存器空标志位。 当UDRE0为“1”时，表明UART0发送数据缓冲器为空，可以写入数据。当UDRE0为“0”时，表明UART0发送数据缓冲器为满，不能写入数据。当数据寄存器空中断使能位UDRIE0为“1”时，UDRE0可用来产生数据寄存器空中断。
4	FE0	帧错误标志位。 当FE0为“1”时，表明接收数据缓冲器接收到的数据有帧错误，即第一个停止位为“0”。当FE0为“0”时，表明接收数据缓冲器接收到的数据没有帧错误，即第一个停止位为“1”。FE0被置位后会一直有效到UDR0被读取。对UCSROA进行写入时，FE0这一位要写“0”。
3	DOR0	数据溢出标志位。当接收缓冲器为满（包含了两个数据），接收移位寄存器中有数据，若此时检测到一个新的起始位，数据溢出产生，DOR0被置位，一直有效到UDR0被读取。对UCSROA进行写入时，DOR0这一位要写“0”。
2	PE0	奇偶校验错误标志位。 当奇偶校验使能（UPM01为“1”）时，且接收缓冲器中所接收到的数据帧有奇偶校验错误，PE0被置位，一直有效到UDR0被读取。对UCSROA进行写入时，PE0这一位要写“0”。
1	U2X0	倍速发送使能位。 当U2X0为“1”时，异步通信模式的传输速率加倍。当U2X0为“0”时，异步通信模式的传输速率为普通速率。这一位仅在异步操作模式下有效，使用同步操作模式时将此位清零。
0	MPCMO	多处理器通信模式使能位。 设置MPCMO位将启动多处理器通信模式。MPCMO置位后，UART0接收器接收到的那些不包含地址信息的输入帧都将被忽略。发送器不受MPCMO设置的影响。

UCSROB - UART0 控制和状态寄存器 B

UCSROB - UART0 控制和状态寄存器B								
地址: 0xC1					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB08	TXB08
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Bit	Name	描述						
7	RXCIE0	接收结束中断使能位。 置位后使能RXCO中断，清零后禁止RXCO中断。当RXCIE0为“1”，全局中断使能，UCSROA寄存器的RXCO为“1”时可以产生UART0接收结束中断。						
6	TXCIE0	发送结束中断使能位。 置位后使能TXCO中断，清零后禁止TXCO中断。当TXCIE0为“1”，全局中断使能，UCSROA寄存器的TXCO为“1”时可以产生UART0发送结束中断。						
5	UDRIE0	数据寄存器空中断使能位。 置位后使能UDRE0中断，清零后禁止UDRE0中断。当UDRIE0为“1”，全局中断使能，UCSROA寄存器的UDRE0为“1”时可以产生UART0数据寄存器空中断。						

4	RXEN0	接收使能位。 置位后启动UART0 接收器。RXD 引脚的通用IO 功能被UART0 接收所取代。禁止接收器将刷新接收缓冲器，并使FE0、DOR0 及PE0 标志无效。
3	TXEN0	发送使能位。 置位后启动 UART0 发送器。TXD 引脚的通用 IO 功能被 UART0 发送所取代。 TXEN0 清零后，只有等到所有的数据发送完成后才能够真正禁止 UART0 发送。
2	UCSZ02	字符长度控制第2 位。 UCSZ02 与UCSRO0 寄存器的UCSZ01:0 结合在一起设置数据帧所包含的数据位数。
1	RXB08	接收数据第8 位。 当数据帧长度为 9 位时，RXB08 是接收数据的最高位。读取 UDRO 所包含的低 8 位数据之前要先读取 RXB08。
0	TXB08	发送数据第8 位。 当数据帧长度为 9 位时，TXB08 是发送数据的最高位。写入 UDRO 所包含的低 8 位数据之前要先写入 TXB08。

UCSRO0 - UART0 控制和状态寄存器 C

UCSRO0 - UART0 控制和状态寄存器C								
地址: 0xC2					默认值: 0x86			
Bit	7	6	5	4	3	2	1	0
	-	-	UPM01	UPM00	USBS0	UCSZ01	UCSZ00	-
R/W	-	-	R/W	R/W	R/W	R/W	R/W	-
Bit	Name	描述						
7:6	-	保留						
5:4	UPM0[1:0]	奇偶校验模式选择位。 高位 UPM01 选择使能或禁止奇偶校验，低位 UPM00 选择奇校验或偶校验。 00: 禁止奇偶校验 01: 保留 10: 使能偶校验 11: 使能奇校验						
3	USBS0	停止位选择位。 0: 1 位停止位 1: 2 位停止位						
2:1	UCSZ0[1:0]	数据帧字符长度选择位。 UCSZ0[1:0]与UCSZ02 结合起来设置数据帧包含的数据位数。 000: 5 位 001: 6 位 010: 7 位 011: 8 位 100: 保留 101: 保留 110: 保留 111: 9 位						

0	-	保留
---	---	----

UBRR0L - UART0 波特率寄存器低字节

UBRR0L - UART0 波特率寄存器低字节								
地址: 0xC4					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	UBRR07	UBRR06	UBRR05	UBRR04	UBRR03	UBRR02	UBRR01	UBRR00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:0	UBRR0[7:0]	UART0 波特率寄存器的低字节部分。 UART0 波特率寄存器包含UBRR0L 和UBRR0H 两部分, 结合在一起用来设置通信的波特率。						

UBRR0H - UART0 波特率寄存器高字节

UBRR0H - UART0 波特率寄存器高字节								
地址: 0xC5					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	-	-	-	-	UBRR011	UBRR010	UBRR009	UBRR008
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:4	-	保留。						
3:0	UBRR0[11:8]	UART0 波特率寄存器的高字节部分。 UART0 波特率寄存器包含UBRR0L 和UBRR0H 两部分, 结合在一起用来设置通信的波特率。 $UBRR0 = \{UBRR0H[3:0], UBRR0L\}$						

UDRO - UART0 数据寄存器

UDRO - UART0 数据寄存器								
地址: 0xC6					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	UDR07	UDR06	UDR05	UDR04	UDR03	UDR02	UDR01	UDR00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:0	UDRO	UART0 发送和接收的数据。 UART0 发送数据缓冲器和接收数据缓冲器共享UART0 数据寄存器UDRO。将数据写入 UDRO 即写入发送数据缓冲器, 从 UDRO 读取数据即读取接收数据缓冲器。 在5 到8 位数据帧模式下, 未使用的第9 位被发送器忽略, 而接收器则将它们 设置为0。 只有当 UCSROA 寄存器的 UDRE0 标志为“1”时才能对发送缓冲器进行写操作, 否则发送器的操作会出错。当发送移位寄存器为空时, 发送器会把发送缓冲器 中的数据加载到发送移位寄存器中, 然后数据串行地从TXD 引脚输出。接收缓冲器包含一个两级 FIFO, 一旦接收缓冲器被读取, FIFO 就会改变它的状态。						

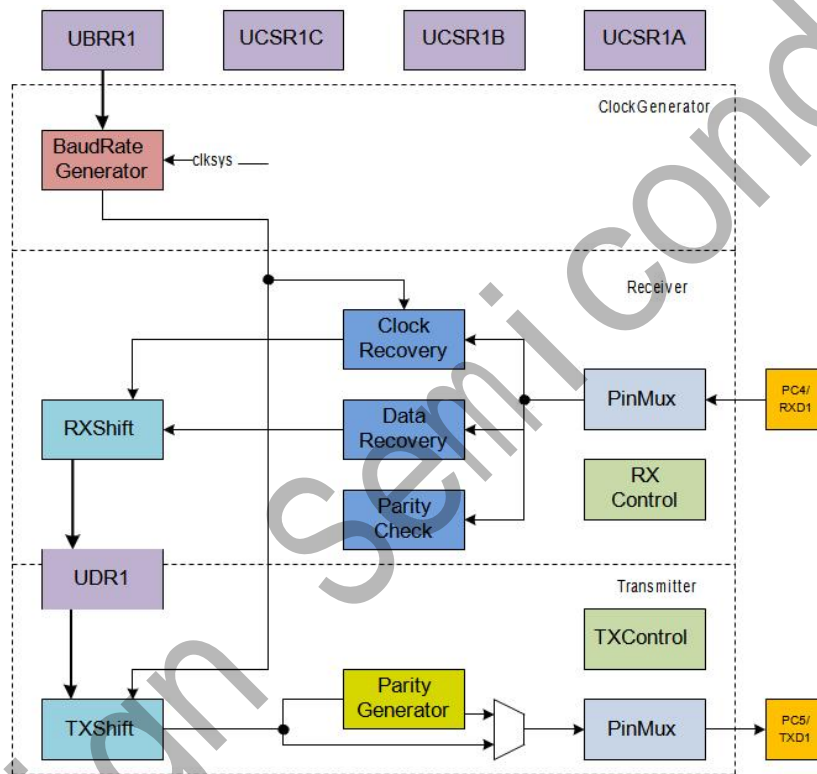
## 15. 通用异步串行收发器 1 (UART1)

- 全双工操作 (独立的串行接收和发送寄存器)
- 主机或从机操作

- 高精度的波特率发生器
- 支持 5, 6, 7, 8, 或 9 个数据位和 1, 或 2 个停止位
- 硬件支持的奇偶产生和校验机制
- 数据过速检测
- 帧错误检测
- 噪声滤波, 包括错误的起始位检测以及数字低通滤波器
- 三个独立的中断, 包括: 发送结束中断, 发送数据寄存器空中断, 以及接收结束中断
- 多处理器通信模式
- 倍速异步通信模式

## 15.1. 综述

UART1 主要包括三个部分: 时钟发生器, 发送器和接收器。如下图所示:



UART1 结构图

时钟发生器根据波特率来产生发送和接收所需的时钟。发送器包括一个写数据缓冲器, 串行移位寄存器, 奇偶发生器以及处理不同帧格式所需的控制逻辑。写数据缓冲器允许连续发送数据而不会在数据帧之间引入延迟。接收器具有时钟和数据恢复单元, 用于异步数据的接收。除了恢复单元, 接收器还包括奇偶校验, 控制逻辑, 串行移位寄存器和一个两级接收缓冲器 UDR1。接收器支持与发送器相同的帧格式, 而且可以检测帧错误, 数据过速和奇偶校验错误。

### 时钟产生

时钟产生逻辑为发送器和接收器产生基础时钟。UART1 支持 4 种模式的时钟: 正常的异步模式, 倍速的异步模式。UCSR1A 的 U2X1 位控制异步模式下的倍速使能。

## 15.2. 波特率发生器

波特率寄存器 UBRR1 和降序计数器连接在一起作为 UART1 的可编程的预分频器或波特率发生器。降序计数器工作在系统时钟 ( $f_{sys}$ ) 下, 当其计数到零或 UBRR1L 寄存器被写时, 会自动加载 UBRR1 寄存器的值。当计数到零时产生一个时钟, 该时钟作为波特率发生器的输出时钟, 频率为  $f_{sys}/(UBRR1+1)$ 。

下表给出了各种工作模式下计算波特率 (位/秒) 以及 UBRR1 值的公式。

工作模式	波特率计算公式 <sup>(1)</sup>	UBRR1 值计算公式
异步正常模式	$BAUD = f_{sys}/(16*(UBRR1+1))$	$UBRR1 = f_{sys}/(16*BAUD) - 1$
异步倍速模式	$BAUD = f_{sys}/(8*(UBRR1+1))$	$UBRR1 = f_{sys}/(8*BAUD) - 1$

说明:

1. 波特率定义为每秒的位传输速度 (bps) ;
2. BAUD 为波特率,  $f_{sys}$  为系统时钟, UBRR1 为波特率寄存器 UBRR1H 和 UBRR1L 的组合值。

### 15.3. 倍速工作模式

通过设定 UCSR1A 寄存器的 U2X1 位可以是传输速率加倍, 该位只在异步工作模式下有效, 同步工作模式下置该位为“0”。设置该位将会把波特率分频器的分频值减半, 有效地加倍异步通信的传输速率。在这种情况下, 接收器只使用一半的采样数来对数据进行采样及时钟恢复, 因此需要更精准的波特率设置和系统时钟。发送器则没有变化。

### 15.4. 帧格式

一个串行数据帧由数据字加上同步位 (起始位和停止位) 以及用于纠错的奇偶校验位构成。

UART1 接受以下 30 种组合的数据帧格式:

- 1 个起始位
- 5、6、7、8 或 9 个数据位
- 无校验位、奇校验位或偶校验位
- 1 或 2 个停止位 数据帧以起始位开始, 紧接着是数据字的最低位, 接着是其它数据位, 以数据字的最高位结束, 最多成功传输 9 位数据。如果使能了校验, 校验位将紧接着数据字, 最后是停止位。当一个完整的数据帧传输后, 可以立即传输下一个新的数据帧, 或者使传输线处于空闲 (高电平) 状态。下图为可能的数据帧结构, 方括号中的位是可选的。



Figure 4 UART 帧结构图

说明:

- 1) IDLE 通信线 (RXD 或 TXD) 上没有数据传输, 线路空闲时必须为高电平
- 2) St 起始位, 总是为低电平
- 3) 0-8 数据位
- 4) P 校验位, 奇校验或偶校验
- 5) Sp 停止位, 总是为高电平

数据帧的结构由 UCSR1B 和 UCSR1C 寄存器中的 UCSZ1[2:0]、UPM1[1:0]和 USBS1 设定。接收与发送使用相同的设置。设置的任何改变都可能破坏正在进行的数据传输。其中, UCSZ1[2:0] 确定了数据帧的数据位数, UPM1[1:0]用于使能和确定校验的类型, USBS1 设置帧有一位或两位结束位。接收器会忽略第二个停止位, 因此帧错误只在第一个结束位为“0”时被检测到。

### 15.5. 校验位计算

校验位的计算是对数据的各个位进行异或运算。如果选择了奇校验, 则异或结果还需要取反。校验位与数据位的关系如下:

$$P_{even} = d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 0$$

$$P_{odd} = d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 1$$

说明:

- 1)  $P_{even}$  偶校验结果
- 2)  $P_{odd}$  奇校验结果

3) dn 第 n 个数据位

## 15.6. UART1 初始化

进行通信之前首先要对 UART1 进行初始化。初始化过程通常包括波特率的设定，帧结构的设定，以及根据需要使能接收器或发送器。对于中断驱动的 UART1 操作，在初始化时要清零全局中断标志并禁止 UART1 的所有中断。在进行重新初始化比如改变波特率或帧结构时，必须确保没有数据传输。TXC1 标志位可以用来检测发送器是否完成了所有传输，RXC1 标志位可以用来检测接收缓冲器中是否还有数据未被读出。如果 TXC1 标志位用作此用途，在每次发送数据之前（写 UDR1 寄存器之前）必须清零 TXC1 标志位。

## 15.7. 发送器

置位 UCSR1B 寄存器的 TXEN1 位将使能 UART1 的数据发送。使能后 TXD 引脚的通用 I/O 功能即被 UART1 功能所取代，成为发送器的串行输出。发送数据之前要设置好波特率、工作模式与帧格式。

### 发送 5 到 8 为数据的帧

将需要发送的数据加载到发送缓冲器中来启动数据发送。CPU 通过写 UDR1 寄存器来加载数据。当发送移位寄存器可以发送新一帧数据的时候，缓冲器中的数据将转移到移位寄存器中。当移位寄存器处于空闲状态（没有正在进行的数据传输），或者前一帧数据的最后一个停止位发送完毕，它将加载新的数据。一旦移位寄存器加载了新的数据，它将按照既定的设置传输一个完整的帧。

### 发送 9 位数据的帧

如果发送 9 位数据的帧，应先将数据的第 9 位写入寄存器 UCSR1B 的 TXB18 位，然后再将低 8 位数据写入发送数据寄存器 UDR1。第 9 位数据在 multi-processor 通信中用于表示地址帧，在同步通信中可用于协议处理。

### 发送奇偶校验位

奇偶校验产生电路为串行数据帧生成相应的校验位。当校验位使能时（UPM1 = 1），发送控制逻辑电路会在数据字的最后一位与第一个停止位之间插入奇偶校验位。

### 发送标志位与中断处理

UART1 发送器有两个标志位：UART1 数据寄存器空标志 UDRE1 和传输结束标志 TXC1，两个标志位都可以产生中断。

数据寄存器空标志 UDRE1 用来表示发送缓冲器是否可以写入一个新的数据。该位在发送缓冲器空时被置“1”，满时被置“0”。当 UDRE1 位为“1”时，CPU 可以往数据寄存器 UDR1 写入新的数据，反之则不能。

当 UCSR1B 寄存器中的数据寄存器空中断使能位 UDRIE1 为“1”时，只要 UDRE1 被置位（且全局中断使能），就将产生 UART1 数据寄存器空中断请求。对寄存器 UDR1 执行写操作将清零 UDRE1。当采用中断方式传输数据时，在数据寄存器空中断服务程序中必须写入一个新的数据到 UDR1 以清零 UDRE1，或者是禁止数据寄存器空中断。否则一旦该中断服务程序结束，一个新的中断将再次产生。当整个数据帧被移出发送移位寄存器，同时发送寄存器中又没有新的数据时，发送结束标志 TXC1 将被置位。当 UCSR1B 上的发送结束中断使能位 TXCIE1（且全局中断使能）置“1”时，随着 TXC1 标志位被置位，UART1 发送结束中断将被执行。一旦进入中断服务程序，TXC1 标志位即被自动清零，CPU 也可以对该位写“1”来清零。

### 禁止发送器

当 TXEN1 清零后，只有等所有的数据都发送完成以后发送器才能真正禁止，即发送移位寄存器与发送缓冲寄存器中都没有要传送的数据。发送器禁止以后，TXD 引脚恢复其通用 I/O 功能。

## 15.8. 接收器

置位 UCSR1B 寄存器的接收允许位（RXEN1）即可启动 UART1 接收器。使能后 RXD 引脚的通用 I/O 功能被 UART1 功能所取代，成为接收器的串行输入。进行数据接收之前首先要设置好波特率、操作模式及帧格式。如果使用同步接收模式，XCK 引脚上的时钟被用作传输时钟。

### 接收 5 到 8 位数据的帧

一旦接收器检测到一个有效的起始位，便开始接受数据。起始位后的每一位数据都将以所设定的波特率来进行接收，直到收到一帧数据的第一个停止位，第二个停止位会被接收器忽略。接收到的每一位数据被送入接收移位寄存器，收到第一个停止位以后，接收器置位于

UCSR1A 寄存器的接收数据完成标志 RXC1 位，并把移位寄存器中完整的数据帧转移到接收缓冲器中，CPU 通过读取 UDR1 寄存器就可以获得接收到的数据。

### 接收 9 位数据的帧

如果设定了 9 位数据的数据帧，在从 UDR1 读取低 8 位数据之前必须首先读取寄存器 UCSR1B 的 RXB18 位来获得第 9 位数据。这个规则同样适用于状态标志位 FE1、DOR1 以及 PE1。读取 UDR1 存储单元会改变接收缓冲器的状态，进而改变同样存储于缓冲器中的 TXB18、FE1、DOR1 及 PE1 位。

### 接收结束标志及中断处理

UART1 接收器有一个标志位：接收结束标志 RXC1，用来表明接收缓冲器中是否有未被读出的数据。当接收缓冲器中有未被读出的数据时，此位为“1”，反之为“0”。如果接收器被禁止，接收缓冲器会被刷新，RXC1 也会被清零。

置位 UCSR1B 的接收结束中断使能位 RXCIE1 后，只要 RXC1 标志被置位（且全局中断被使能），就会产生 UART1 接收结束中断。使用中断方式进行数据接收时，数据接收结束中断服务程序必须从 UDR1 读取数据来清零 RXC1 标志，否则只要中断处理程序一结束，一个新的中断就会产生。

### 接收错误标志

UART1 接收器有三个错误标志：帧错误 FE1、数据溢出 DOR1 及奇偶校验错误 PE1。它们都位于 UCSR1A 寄存器。错误标志与数据帧一起保存在接收缓冲器当中。所有的错误标志都不能产生中断。

帧错误标志 FE1 表明存储在接收缓冲器中的下一个可读帧的第一个停止位的状态。停止位正确（值为“1”）则 FE1 标志为“0”，否则 FE1 标志为“1”。这个标志可用于检测同步丢失、传输中断，也可用于协议处理。

数据溢出标志 DOR1 表明由于接收缓冲器满造成了数据丢失。当接收缓冲器为满，接收移位寄存器中已有数据，若此时检测到一个新的起始位，数据溢出就产生了。DOR1 标志被置位即表明在最近一次读取 UDR1 和下一次读取 UDR1 之间丢失了一个或多个数据帧。当数据帧成功地从移位寄存器转入接收缓冲器后，DOR1 标志被清零。

奇偶校验错误标志 PE1 表明接收缓冲器中的下一帧数据在接收时有奇偶错误。如果不使能奇偶校验，UPM11 被清零。

### 奇偶校验器

置位奇偶校验模式位 UPM11 将启动奇偶校验器。校验的模式（偶校验或奇校验）由 UPM10 决定。奇偶校验使能后，校验器将计算输入数据的奇偶并把结果与数据帧的奇偶位进行比较。校验结果将与数据和停止位一起存储在接收缓冲器中。CPU 通过读取 PE1 位来检查接收的帧

当中是否有奇偶错误。如果下一个从接收缓冲器中读出的数据有奇偶错误，并且奇偶校验使能，则 PE1 被置位，一直有效到接收缓冲器 UDR1 被读取。

### 禁止接收器

与发送器相比，禁止接收器即刻起作用。正在接收的数据将丢失。禁止接收器（RXEN1 清零）后，接收器将不再占用 RXD 引脚，接收缓冲器也会被刷新。

## 15.9. 异步数据接收

UART1 有一个时钟恢复单元和数据恢复单元来处理异步数据接收。时钟恢复逻辑用于同步从 RXD 引脚输入的异步串行数据和内部的波特率时钟。数据恢复逻辑用于采集数据，并通过低通滤波器过滤所输入的每一位数据，从而提高接收器的抗干扰性能。异步接收的工作范围依赖于内部波特率时钟的精度、帧输入的速率及一帧所包含的数据位数。

### 异步工作范围

接收器的工作范围依赖于接收到的数据速率与内部波特率之间的不匹配程度。如果发送器以过快或过慢的比特率传输数据，或者接收器内部产生的波特率没有相同的频率，那么接收器就无法与起始位同步。为了确保接收器不会错过下一帧起始位的采样，数据输入速率和内部接收器波特率不能相差太大，用它们之间的比值来描述波特率的误差范围。下面两个表格分别给出了普通模式下和倍速模式下容许的最大波特率误差范围。

Table 5 普通模式下最大接收器波特率误差范围

数据位+奇偶位长度和	最大误差范围 (%)	推荐误差范围 (%)
5	+6.7/-6.8	±3.0
6	+5.8/-5.9	±2.5
7	+5.1/-5.2	±2.0
8	+4.6/-4.5	±3.0
9	+4.1/-4.2	±1.5

10	+3.8/-3.8	±1.5
----	-----------	------

Table 6 倍速模式下最大接收器波特率误差范围

数据位+奇偶位长度和	最大误差范围 (%)	推荐误差范围 (%)
5	+5.7/-5.9	±2.5
6	+4.9/-5.1	±2.0
7	+4.4/-4.5	±1.5
8	+3.9/-4.0	±1.5
9	+3.5/-3.6	±1.0
10	+3.2/-3.3	±1.0

从表中可以看出，普通模式下波特率允许有更大的变化范围。上述推荐的波特率误差范围是假定接收器和发送器对最大总误差具有同等贡献的前提下得出的。产生接收器波特率误差的可能原因有两个。首先，接收器系统时钟的稳定性与工作电压和温度有关。使用晶振来产生系统时钟时一般不会有此问题，但使用内部振荡器时，系统时钟可能会有偏差。第二个原因是波特率发生器不一定能通过对系统时钟的分频来得到恰好想要的波特率。此时可以调整 UBRR1 的值，使得误差低至可以接受。

### 15.10. 波特率设置及引入误差

对于标准晶振及谐振器频率来说，异步模式下的实际通信的波特率可通过波特率计算公式来获得，它与常用通信波特率之间的误差可用如下公式来计算：

$$\text{Error} [\%] = (\text{BaUDRE1a1} / \text{Baud} - 1) * 100\%$$

其中，Baud 为常用的通信波特率，BaUDRE1a1 为通过计算公式算出来的波特率，带入波特率计算公式即可得到波特率误差与系统时钟  $f_{\text{sys}}$  和波特率寄存器 UBRR1 值之间的关系如下：

普通模式：

$$\text{Error} [\%] = (f_{\text{sys}} / (16 * (\text{UBRR1} + 1)) / \text{Baud} - 1) * 100\%$$

倍速模式：

$$\text{Error} [\%] = (f_{\text{sys}} / (8 * (\text{UBRR1} + 1)) / \text{Baud} - 1) * 100\%$$

当不考虑通信两边的时钟误差，即系统时钟  $f_{\text{sys}}$  为标准时钟时，即可得到波特率误差 UBRR1 值之间的关系。下表即为 16MHz 系统时钟下不同 UBRR1 值设置下的波特率误差。

Table 7 16MHz 系统时钟下设置UBRR1 值所产生的误差

波特率 (bps)	$f_{\text{sys}} = 16.000\text{MHz}$			
	普通模式 (U2X1 = 0)		倍速模式 (U2X1 = 1)	
	UBRR1	误差	UBRR1	误差
2400	416	-0.1%	832	0.0%
4800	207	0.2%	416	-0.1%
9600	103	0.2%	207	0.2%
14.4K	68	0.6%	138	-0.1%
19.2K	51	0.2%	103	0.2%
28.8K	34	-0.8%	68	0.6%
38.4K	25	2.1%	34	-0.8%
57.6K	16	0.2%	51	0.2%
76.8K	12	0.2%	25	0.2%
115.2K	8	-3.5%	16	2.1%
230.4K	3	8.5%	8	-3.5%
250K	3	0%	7	0%
0.5M	1	0%	3	0%

1M	0	0%	1	0%
----	---	----	---	----

### 15.11. 多处理器通信模式

置位 UCSR1A 的多处理器通信模式 (MPCM1) 位可以对 UART1 接收器接收到的数据帧进行过滤。那些没有地址信息的帧将被忽略, 也不会存入接收缓冲器。在一个多处理器系统中, 各处理器通过相同的串行总线进行通信, 这种过滤有效的减少了需要 CPU 处理的数据帧的数量。MPCM1 位的设置不影响发送器的工作, 但在多处理器通信的系统中, 它的使用方法会有所不同。

如果接收器所接收的数据帧长度为 5 到 8 位, 那么第一个停止位会用来表示当前帧包含的是数据还是地址信息。如果接收器所接收的数据帧长度是 9 位, 那么由第 9 位来确定是数据还是地址信息。如果帧类型标志位为“1”, 那么这是地址帧, 否则为数据帧。

在多处理器通信模式下, 允许多个从处理器从一个主处理器接收数据。首先要通过解码地址帧来确定所寻址的是哪一个从处理器。被寻址的从处理器将正常接收后续的数据, 而其他的从处理器则会忽略这些数据帧直到接收到下一个地址帧。

对于一个作为主机的处理器来说, 它可以使用 9 位数据帧格式, 并用第 9 位数据来标识帧

格式。在这种通信模式下, 从处理器也必须工作于 9 位数据帧格式。下面即为多处理器通信模式下进行数据交换的步骤:

1. 所有从处理器都工作在多处理器通信模式 (置位 MPCM1);
2. 主处理器发送地址帧, 所有从处理器都接收此帧。从处理器 UCSR1A 寄存器的 RXC1

位正常置位;

3. 每个从处理器都读取 UDR1 寄存器的内容, 解码地址帧来确定是否被选中。如果选中, 就清零 UCSR1A 寄存器的 MPCM1 位, 未被选中, 则保持 MPCM1 为“1”并等待下一个地址帧的到来;

4. 被寻址的从处理器接收所有的数据帧, 直到收到一个新的地址帧。未被寻址的从处理器忽略这些数据帧;

5. 被寻址的从处理器收到最后一个数据帧后, 置位 MPCM1 位, 并等待下一个地址帧的到来。然后从第二步骤重复进行。

使用 5 到 8 位数据的帧格式是可以的, 但是不切实际, 因为接收器必须在使用 n 和 n+1 帧格式之间进行切换。由于接收器和发送器使用相同的字符长度设置, 这种设置使得全双工操作变得很困难。如果使用 5 到 8 位数据的帧格式, 发送器应该设置两个停止位, 其中第一个停止位被用于判断帧类型。

### 15.12. UART1 寄存器列表

UART1 寄存器列表

寄存器	地址	默认值	描述
UCSR1A	0XC8	0x20	UART1 控制和状态寄存器A
UCSR1B	0xC9	0x00	UART1 控制和状态寄存器B
UCSR1C	0xCA	0x06	UART1 控制和状态寄存器 C
UBRR1L	0xCB	0x0	UART1 波特率寄存器低字节
UBRR1H	0xCC	0x0	UART1 波特率寄存器高字节
UDR1	0xCD	0x0	UART1 数据寄存器

#### UART1 寄存器描述

UCSR1A - UART1 控制和状态寄存器 A

UCSR1A - UART1 控制和状态寄存器A								
地址: 0XC8				默认值: 0x20				
Bit	7	6	5	4	3	2	1	0
RXC1	TXC1	UDRE1	RE	DOR1	PE1	U2X1	MPCM1	
R/W	R	R/W	R	R	R	R	R/W	R/W
Bit	Name	描述						

7	RXC1	接收结束标志位。 当RXC1的值为“1”时，表明接收缓冲器中有未读出的数据。当RXC1的值为“0”时，表明接收缓冲器中没有未读出的数据。接收器禁止时，接收缓冲器被刷新，导致RXC1被清零。当接收结束中断使能位RXCIE1为“1”时，RXC1可用来产生接收结束中断。
6	TXC1	发送结束标志位。 发送移位寄存器中的数据被送出，且发送缓冲器为空时 TXC1 置位。执行发送结束中断时 TXC1 自动清零，也可以通过对 TXC1 写“1”来进行清零。当发送结束中断使能位 TXCIE1 为“1”时，TXC1 可用来产生发送结束中断。
5	UDRE1	数据寄存器空标志位。 当UDRE1为“1”时，表明UART1发送数据缓冲器为空，可以写入数据。当UDRE1为“0”时，表明UART1发送数据缓冲器为满，不能写入数据。当数据寄存器空中断使能位 UDRIE1 为“1”时，UDRE1 可用来产生数据寄存器空中断。
4	FE1	帧错误标志位。 当FE1为“1”时，表明接收数据缓冲器接收到的数据有帧错误，即第一个停止位为“0”。当FE1为“0”时，表明接收数据缓冲器接收到的数据没有帧错误，即第一个停止位为“1”。FE1被置位后会一直有效到UDR1被读取。对UCSR1A进行写入时，FE1这一位要写“0”。
3	DOR1	数据溢出标志位。当接收缓冲器为满（包含了两个数据），接收移位寄存器中有数据，若此时检测到一个新的起始位，数据溢出产生，DOR1被置位，一直有效到UDR1被读取。对UCSR1A进行写入时，DOR1这一位要写“0”。
2	PE1	奇偶校验错误标志位。 当奇偶校验使能（UPM11为“1”）时，且接收缓冲器中所接收到的数据帧有奇偶校验错误，PE1被置位，一直有效到UDR1被读取。对UCSR1A进行写入时，PE1这一位要写“0”。
1	U2X1	倍速发送使能位。 当U2X1为“1”时，异步通信模式的传输速率加倍。当U2X1为“0”时，异步通信模式的传输速率为普通速率。这一位仅在异步操作模式下有效，使用同步操作模式时将此位清零。
0	MPCM1	多处理器通信模式使能位。 设置MPCM1位将启动多处理器通信模式。MPCM1置位后，UART1接收器接收到的那些不包含地址信息的输入帧都将被忽略。发送器不受MPCM1设置的影响。

UCSR1B - UART1 控制和状态寄存器 B

UCSR1B - UART1 控制和状态寄存器B								
地址: 0xC9					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	RXCIE1	TXCIE1	UDRIE1	RXEN1	TXEN1	UCSZ12	RXB18	TXB18
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Bit	Name	描述						

7	RXCIE1	接收结束中断使能位。 置位后使能RXC1 中断，清零后禁止RXC1 中断。当RXCIE1 为“1”，全局中断使能，UCSR1A 寄存器的RXC1 为“1”时可以产生UART1 接收结束中断。
6	TXCIE1	发送结束中断使能位。 置位后使能TXC1 中断，清零后禁止TXC1 中断。当TXCIE1 为“1”，全局中断使能，UCSR1A 寄存器的TXC1 为“1”时可以产生UART1 发送结束中断。
5	UDRIE1	数据寄存器空中断使能位。 置位后使能UDRE1 中断，清零后禁止UDRE1 中断。当UDRIE1 为“1”，全局中断使能，UCSR1A 寄存器的UDRE1 为“1”时可以产生UART1 数据寄存器空中断。
4	RXEN1	接收使能位。 置位后启动UART1 接收器。RXD 引脚的通用 I/O 功能被UART1 接收所取代。禁止接收器将刷新接收缓冲器，并使FE1、DOR1 及PE1 标志无效。
3	TXEN1	发送使能位。 置位后启动 UART1 发送器。TXD 引脚的通用 I/O 功能被 UART1 发送所取代。 TXEN1 清零后，只有等到所有的数据发送完成后才能够真正禁止 UART1 发送。
2	UCSZ12	字符长度控制第2 位。 UCSZ12 与UCSR1C 寄存器的UCSZ11:0 结合在一起设置数据帧所包含的数据位数。
1	RXB18	接收数据第8 位。 当数据帧长度为 9 位时，RXB18 是接收数据的最高位。读取 UDR1 所包含的低 8 位数据之前要先读取 RXB18。
0	TXB18	发送数据第8 位。 当数据帧长度为 9 位时，TXB18 是发送数据的最高位。写入 UDR1 所包含的低 8 位数据之前要先写入 TXB18。

**UCSR1C - UART1 控制和状态寄存器 C**

UCSR1C - UART1 控制和状态寄存器C								
地址: 0xCA					默认值: 0x86			
Bit	7	6	5	4	3	2	1	0
	U110S	-	UPM11	UPM10	USBS1	UCSZ11	UCSZ10	-
R/W	R/W	-	R/W	R/W	R/W	R/W	R/W	-
Bit	Name	描述						
7	U110S	UART1 端口选择位： 1 - 选择PC4/5 作为串行通讯端口 0 - 选择PC0/1 作为串行通讯端口						
6	-	保留位用						
5:4	UPM1 [1:0]	奇偶校验模式选择位。 高位 UPM11 选择使能或禁止奇偶校验，低位 UPM10 选择奇校验或偶校验。 00: 禁止奇偶校验 01: 保留 10: 使能偶校验 11: 使能奇校验						



3	USBS1	停止位选择位。 0: 1 位停止位 1: 2 位停止位
2:1	UCSZ1[1:0]	数据帧字符长度选择位。 UCSZ1[1:0]与UCSZ12 结合起来设置数据帧包含的数据位数。 000: 5 位 001: 6 位 010: 7 位 011: 8 位 100: 保留 101: 保留 110: 保留 111: 9 位
0	-	保留

UBRR1L - UART1 波特率寄存器低字节

UBRR1L - UART1 波特率寄存器低字节								
地址: 0xCB					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	UBRR17	UBRR16	UBRR15	UBRR14	UBRR13	UBRR12	UBRR11	UBRR10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:0	UBRR1[7:0]	UART1 波特率寄存器的低字节部分。 UART1 波特率寄存器包含UBRR1L 和UBRR1H 两部分, 结合在一起用来设置通信的波特率。						

UBRR1H - UART1 波特率寄存器高字节

UBRR1H - UART1 波特率寄存器高字节								
地址: 0xCC					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	-	-	-	-	UBRR111	UBRR110	UBRR19	UBRR18
R/W	-	-	-	-	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:4	-	保留。						
3:0	UBRR1[11:8]	UART1 波特率寄存器的高字节部分。 UART1 波特率寄存器包含UBRR1L 和UBRR1H 两部分, 结合在一起用来设置通信的波特率。 UBRR1 = {UBRR1H[3:0], UBRR1L}						

UDR1 - UART1 数据寄存器

UDR1 - UART1 数据寄存器								
地址: 0xCD					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0

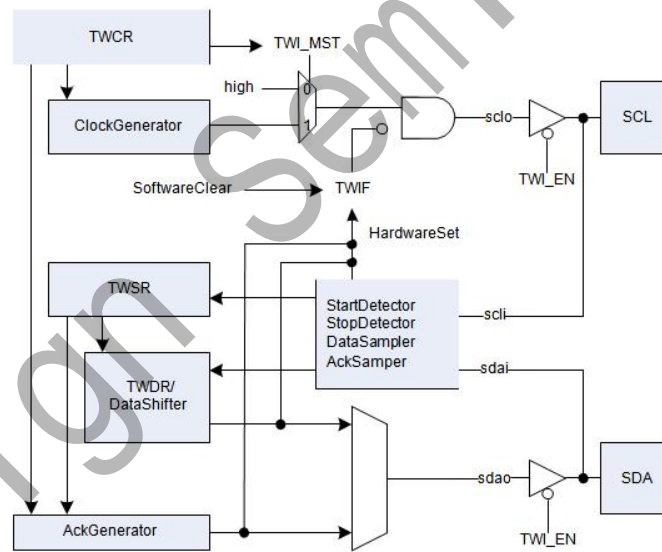
	UDR17	UDR16	UDR15	UDR14	UDR13	UDR12	UDR11	UDR10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:0	UDR1	UART1 发送和接收的数据。 UART1 发送数据缓冲器和接收数据缓冲器共享 UART1 数据寄存器 UDR1。将数据写入 UDR1 即写入发送数据缓冲器，从 UDR1 读取数据即读取接收数据缓冲器。 在 5 到 8 位数据帧模式下，未使用的第 9 位被发送器忽略，而接收器则将它们设置为 0。 只有当 UCSR1A 寄存器的 UDRE1 标志为“1”时才能对发送缓冲器进行写操作，否则发送器的操作会出错。当发送移位寄存器为空时，发送器会把发送缓冲器中的数据加载到发送移位寄存器中，然后数据串行地从 TXD 引脚输出。接收缓冲器包含一个两级 FIFO，一旦接收缓冲器被读取，FIFO 就会改变它的状态。						

## 16. I2C 串行通信控制器

- 支持主机/从机工作模式
- 支持传输速率设置

### 16.1. 综述

TWI 控制器是一种双线通讯接口控制器。协议实现兼容 I2C 总线。可用于与其他使用 I2C 接口的外设通讯



TWI 结构图

### 16.2. 工作模式

如需要 TWI 控制器工作为从机模式，需要置位 TWCR 寄存器的 TWEN。如需要工作为 I2C 主机模式，需要同时设置 TWEN 位与 TWMST 位，然后通过 TWCR 寄存器的 CKPS 位设置产生通讯所需的 SCL 时钟信号。

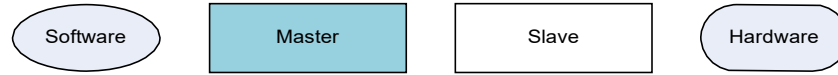
当接收到一个开始条件，重启位，一个停止位，8 位数据或一个响应位，控制器将会置位 TWSR 寄存器的对应状态信息位。软件可以通过读取 TWSR 寄存器获得接收信息，更新状态以及相关操作，然后通过写 0 清除相关的标记位。

当需要发生一个 8 位数据，响应位或者主机发生一个开始或者停止条件，软件需要设置 TWSR 寄存器中相应的标志位，之后控制器会做出相应的动作并清除标志位。当主机发送一个开始位或者停止位时，控制位也会同时接收到这个开始或者停止位。

当接收到一个开始位或一个停止位或 8 位数据或 1 位响应或者是发送一个响应位，控制器会同时置位 TWCR 寄存器中的 TWIF 中断标志位。TWIF 被置位后，控制器将 SCL 保持为低电平直到 TWIF 会被清零。

### 16.3. 操作模式

TWI 控制器可工作于 4 种主要模式。这些模式包括：主机发送(MT)，主机接收(MR)，从机发送(ST)，从机接收(SR)。下面的章节将会分别描述这些模式。在下面的图示中，“Software”表示软件相关操作；“Hardware”表示硬件相关操作；“Master”表示主机操作；“Slave”表示从机操作；“SLA”表示从机地址；“W”表示写操作(SDA 低电平)；“R”表示读操作(SDA 高电平)。“ACK”表示响应位；“NACK”表示无响应。



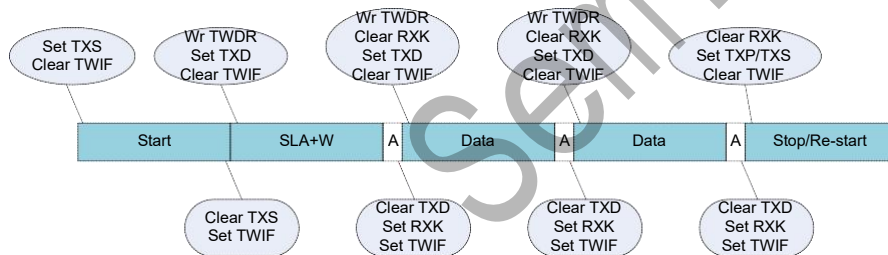
### 16.4. 主机发送模式

如需工作在主机发送模式，首先，TWI 需要配置为主机工作模式。通过置位 TWSR 寄存器的 TXS 位并清零 TWIF 中断状态，控制器将会产生在总线空闲时产生一个传输开始条件(START)。开始条件传输完成后，TWIF 状态位被置位，同时 TXS 位被清零。

接下来，需要发送一个 SLA+W。这个操作可以通过向 TWDR 寄存器中写入 SLA+W 完成，设置 TWSR 寄存器的 TXD 位，清零 TWIF。当 SLA+W 发送完成并接收到一个响应位后，TWIF 位再次被置位，TXD 位清零，RXK 位设置以表明当前已收到正确的响应信息。

SLA+W 成功传输后，接下来可以发送数据。将需要发送的数据写入 TWDR 寄存器，清零 RXK 位，置位 TXD 并清零 TWIF。当数据发送完成并接收到正确的响应信息后，TWIF 位被重新置位，TXD 被清零，RXK 被置位。

接下来重复数据发送流程，直到最后一个字节的数据发送完成。然后发送一个停止位停止数据传输或者重复一个开始位，继续传输其他的数据。停止位通过写 TWSR 寄存器的 TXP 位，清零 RXK 位与 TWIF 位实现。重复开始条件可以通过置位 TXS 位，清零 RXK 以及 TWIF 位实现。停止位发送结束后，TXP 位被清零，但 TWIF 此时不会被置位。当重复开始发送结束后，TXS 被清零，同时 TWIF 标志位被置位。



主机发送模式数据流图

### 16.5. 主机接收模式

首先，需要将 TWI 器配置为 I2C 主机模式。

通过置位 TWSR 寄存器的 TXS 位，清零 TWIF 标志位，启动传输流程。TWI 控制器将会在总线空闲后发起一个开始位。开始位发送完成后，TWIF 标志位置位，同时 TXS 位清零。

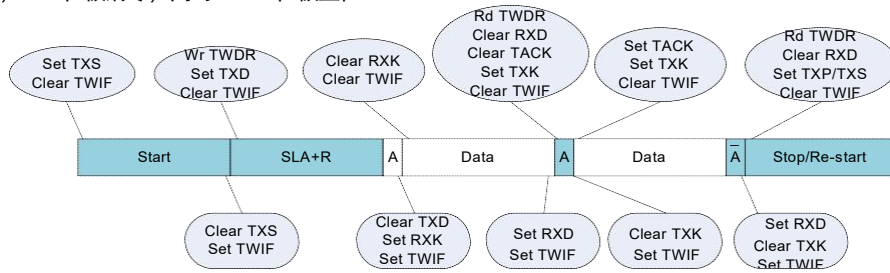
进入 MR 模式前，需要先发送 SLA+R。将 SLA+R 写入 TWDR 寄存器，置位 TXD，清零 TWIF 将会启动 SLA+R 传输。在 SLA+R 传输完成并成功接收到响应信号后，TWIF 被再次置位，TXD 位被清零，同时 RXK 位被置位。

SLA+R 成功传输后，控制器可以进入数据接收状态。清零 RXK 以及 TWIF 标志位，控制器将会在 SCL 产生 8 个时钟周期脉冲，并接收完成一个自己的数据。数据接收完成后，TWSR 寄存器的 RXD 被置位。接收的数据可以通过读取 TWDR 寄存器获得。读完数据后需清零 RXD 标志位。数据接收后，需发送一个响应给从机。置位 TXK 标志位并清零 TWIF 标志位，控制器发送的是响应(ACK)还是非响应(NACK)取决于 TWCR 寄存器的 TACK 位。如果 TACK 为 0，主机将返回一个 ACK，否则返回一个 NACK。响应信号发送完成后，TXK 位被清零，同时 TWIF 位被再次置位。

数据接收过程将会重复直到接收到最后一个字节。数据接收的过程也可以被进一步简化。简化接收流程的操作通过预先设定响应位并置位 TXK 位实现。TXK 置位后，控制器在接收到一个字节的的数据后，并不会触发 TWIF 标志位。待到响应位成功发送后，TWIF 置位。接收的数据可以从 TWDR 中读取。

传输由一个停止位或者一个重复开始位结束。通过置位 TXP 位并清零 RXK 位以及 TWIF 标志位，可以向总线发送一个停止位。重复开始条件可以通过置位 TXS 位，清零 RXK 位以及 TWIF 标志位实现。停止位发送后，TXP 位清零，TWIF 标志位将不会被置位。当重复开

启位发送后, TXS 位被清零, 同时 TWIF 位被置位。



主机接收数据流程图

## 16.6. 从机发送模式

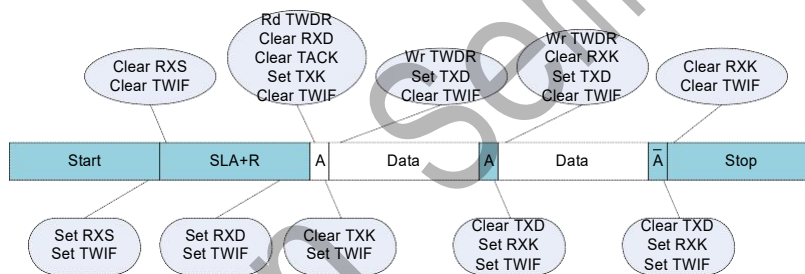
首先, 需要设置 TWI 控制器配置为 I2C 从机工作模式。

当控制器在总线上检测到一个传输开始位, TWSR 寄存器的 RXS 位被置位, 同时 TWIF 位也被置位。这种状态说明总线此时已被其他的主机占用。软件需要清零 RXS 位以及 TWIF 位准备接收接下来主机发送的地址以及读写控制位。

当接收到 SLA+W/R 字节后, RXD 标志位被置位, 同时 TWIF 位也被置位。软件响应 TWIF 中断, 读取 SLA+W/R 数据并根据其中的地址信息判断是否被寻址。如果本机是访问目标, 清零 TACK 标志位, 否则置位 TACK 标志位。如果本机被寻址, 软件需要更加 W/R 控制位确定接下来的操作。如果是主机发起的读操作, 控制器将会进入到从机发送模式, 软件可以通过清零 RXD, 设置 TXK 标志位以及清零 TWIF 标志位发送响应信息。

响应位发送后, TXK 位被清零, TWIF 位被置位。此后, 可以发送数据部分。将待发送的数据写入 TWDR 寄存器, 置位 TXD 标志位并清零 TWIF 标志位, 启动数据发送。当数据被成功发送并接收到来自主机的响应位后, TXD 位被清零, RXK 以及 TWIF 位再次被置位。

数据传输流程将会反复循环直到发送完成最后一个数据字节或接收到 NACK, 接收到一个停止位, 或者接收到一个重复开始条件。



从机发送模式数据流程图

## 16.7. 从机接收模式

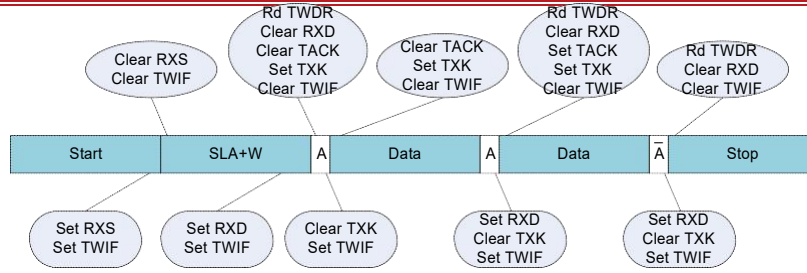
进入到从机接收模式前, 首先需要将 TWI 控制器配置为 I2C 从机模式。

当控制器在总线上检测到开始传输条件, RXS 位置位, 同时 TWIF 也被置位。此时的状态说明总线已被其他主机占用。软件需清零 RXS 以及 TWIF 位以接收从机地址控制数据。

SLA+W/R 字节接收完成后, RXD 位以及 TWIF 位同时被置位。软件通过 TWDR 寄存器获得接收的数据并根据接收的地址信息判断是否被寻址。如果被主机寻址, 软件需要清零 TACK 位, 否则置位 TACK。被寻址的控制器还需要根据读写控制位确定接下来需要进入的工作模式。如果是接收到一个写操作, 控制器进入从机接收模式。控制器通过清零 RXD 位, 置位 TXK 位以及清零 TWIF 位向主机发送响应信息。

响应信息发送完成后, TXK 被清零, TWIF 再次置位。此时可以接收数据。与主机接收模式相同, 从机的数据接收流程也可以被简化。通过预定义响应位(TACK 位), 设置 TXK 位以及清零 TWIF 位, 可使控制器工作于简化数据接收模式。控制器接收完一个字节的数并发送完响应信息后, TXK 位清零, TWIF 位被置位。接收的数据被保存到 TWDR 寄存器。

同样, 数据接收流程将会被循环直到接收完最后一个字节并完成 NACK 的发送。



从机接收模式流程图

## 16.8. 寄存器定义

TWCR - TWI 控制寄存器

TWCR - TWI 控制寄存器								
地址: 0xBC					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	TWEN	TWMST	TWIE	TWIF	TACK	RACK	CKPS1	CKPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name		描述					
7	TWEN		TWI 使能控制位, 高有效					
6	TWMST		TWI 主机模式控制位 0: 从机模式 1: 主机模式					
5	TWIE		TWI 中断使能控制位, 高有效					
4	TWIF		TWI 中断标志位 当控制器发送完或接收到起始位, 发送完数据, 发送完或接收到响应位后, 会置位 TWIF 中断标志位, 同时会一直拉低 SCL 信号, 直到 TWIF 位被清零。TWIF 位只能通过往该位写 1 的方式来清零, 即使执行中断服务程序, 硬件也不会自动 清零该位。同时要注意, 清零该位将立即继续 TWI 的操作。因此, 在清零 TWIF 位之前, 软件要首先完成对 TWSR 和 TWDR 等寄存器的访问					
3	TACK		发送总线响应控制位 0: ACK 1: NACK					
2	RACK		接收总线响应控制位 0: ACK 1: NACK					
1:0	CKPS		TWI 总线时钟分频选择位 00: FSCL = Fsys/8 01: FSCL = Fsys/15 10: FSCL = Fsys/30 11: FSCL = Fsys/60					

TWSR - TWI 状态寄存器

TWSR - TWI 状态寄存器								
地址: 0xB9					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0

	TXP	RXK	TXD	TXS	RXP	TXK	RXD	RXS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name		描述					
7	TXP		软件写1 发送停止位 硬件清零，表示停止位已发送完成					
6	RXK		接收到总线响应。硬件置位，软件写0 清零					
5	TXD		软件写1 发送数据 硬件清零，表示数据发送完成					
4	TXS		软件写1 发送开始位 硬件清零，表示开始位已发送完成					
3	RXP		接收到停止位。硬件置位，软件写0 清零					
2	TXK		软件写1 发送响应位 硬件清零，表示响应位已发送完成					
1	RXD		数据接收完成。硬件置位，软件写0 清零					
0	RXS		接收到开始位。硬件置位，软件写0 清零					

TWDR - TWI 数据寄存器

TWDR - TWI 数据寄存器								
地址: 0xBB					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name		描述					
7:0	TWDR		TWI 发送/接收数据寄存器					

## 17. 同步串行外设接口 (SPI)

- 全双工，三线同步数据传输
- 主机或从机操作
- 最低位或最高位优先传输
- 7 种可编程的比特率
- 发送结束中断标志
- 写入冲突标志保护机制
- 可从闲置模式唤醒
- 主机操作时具有倍速模式
- 支持主机双线输入模式

### 17.1. 综述

SPI 主要包括三个部分：时钟检测器，从机选择检测器，发送器和接收器。

控制和状态寄存器由这三个部分共享。时钟预分频器只工作在主机操作模式下，由比特率控制位来选择分频系数，从而产生相应的分频时钟，输出到 SPCK 引脚上。时钟检测器只工作在从机操作模式下，检测从 SPCK 引脚上输入的时钟沿，根据 SPI 的数据传输模式对发送和接收移位寄存器进行移位操作。从机选择检测器对从机选择信号 SPSS 进行检测，得到传输的状态来控制发送器和接收器的操作。发送器由一个移位寄存器和发送控制逻辑组成。接收器由一个移位寄存器，四个接收缓冲器和接收控制逻辑组成。

#### 时钟产生

时钟产生逻辑分为主机时钟预分频器和从机时钟检测器，分别工作在主机操作和从机操作模式下。时钟预分频器由比特率控制位和倍速控制位来选择分频系数，产生相应的分频时钟（共有 7 种可选的分频系数，详细信息见寄存器描述），输出到 SPCK 引脚为通信提供时钟，同时为内部发送和接收移位寄存器提供移位时钟。时钟检测器对输入时钟 SPCK 进行边沿检测，根据 SPI 的数据传输模式对发送器和接收器进行移位操作。为保证对时钟信号的正确采样，SPCK 时钟的高电平和低电平的宽度均须大于 2 个系统时钟周期。

### 17.2. 发送和接收

SPI 模块在单线模式下支持同时发送和接收，在双线模式下只支持主机双线接收。

SPI 的主机将需要通信的从机选择信号 SPSS 拉低，即可启动一次传输过程。主机和从机将需要传输的数据准备好，主机在时钟信号 SPCK 上产生时钟脉冲以交换数据，主机的数据从 MOSI 移出，从 MISO 移入，从机的数据从 MISO 移出，从 MOSI 移入，交换完数据后主机拉高 SPSS 信号即可完成通信。

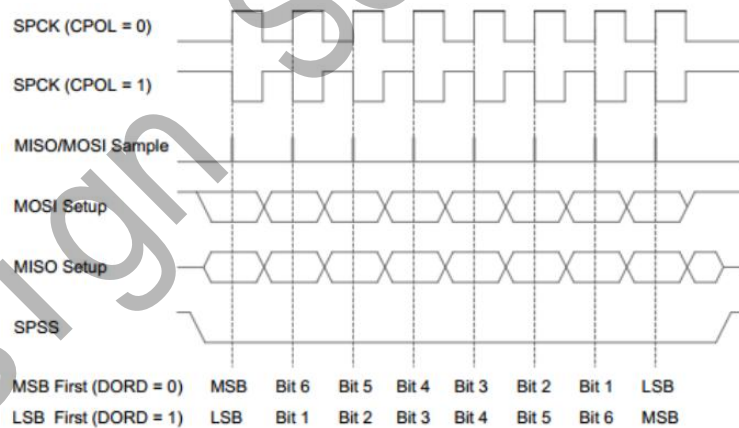
SPI 模块在发送方向只有四个缓冲器，在接收方向也有四个缓冲器。在发送数据时，当发送缓冲器处于非满状态（即发送缓冲器满标志位 WRFULL 位为低）时，可对 SPDR 寄存器进行写操作。而在接收数据时，当接收缓冲器属于非空状态（即接收缓冲器空标志位 RDEMP 位为低）时，可通过访问 SPDR 寄存器读取已经接收到的字符。

### 17.3. 数据模式

单线模式下，相对于串行数据，SPI 有 4 种 SPCK 相位和极性的组合方式，由 CPHA 和 CPOL 来控制，如下表所示。  
CPHA 和 CPOL 选择数据传输模式

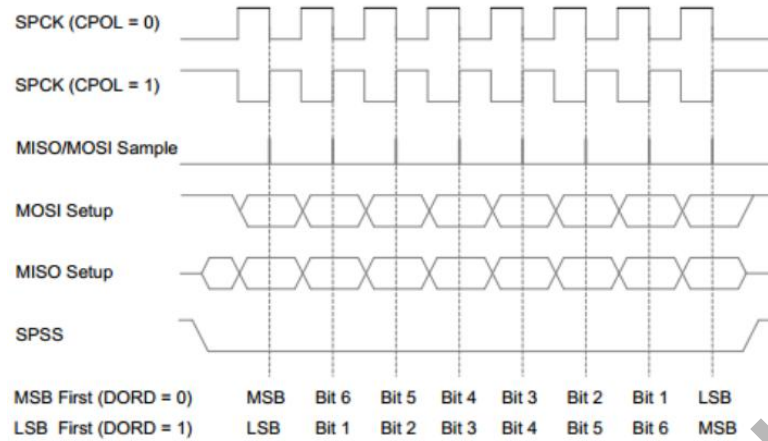
CPOL	CPHA	起始沿	结束沿	SPI 模式
0	0	采样 (上升沿)	设置 (下降沿)	0
0	1	设置 (上升沿)	采样 (下降沿)	1
1	0	采样 (下降沿)	设置 (上升沿)	2
1	1	设置 (下降沿)	采样 (上升沿)	3

当 CPHA=0 时，数据采样和设置的时钟沿如下图所示：



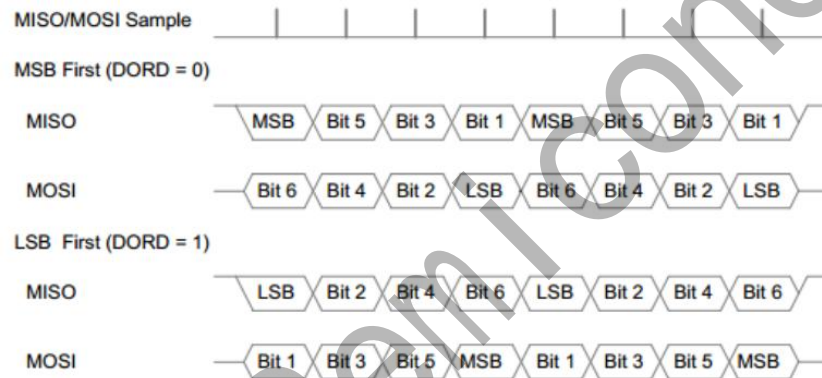
CPHA 为“0”时 SPI 数据传输模式

当 CPHA=1 时，数据采样和设置的时钟沿如下图所示：



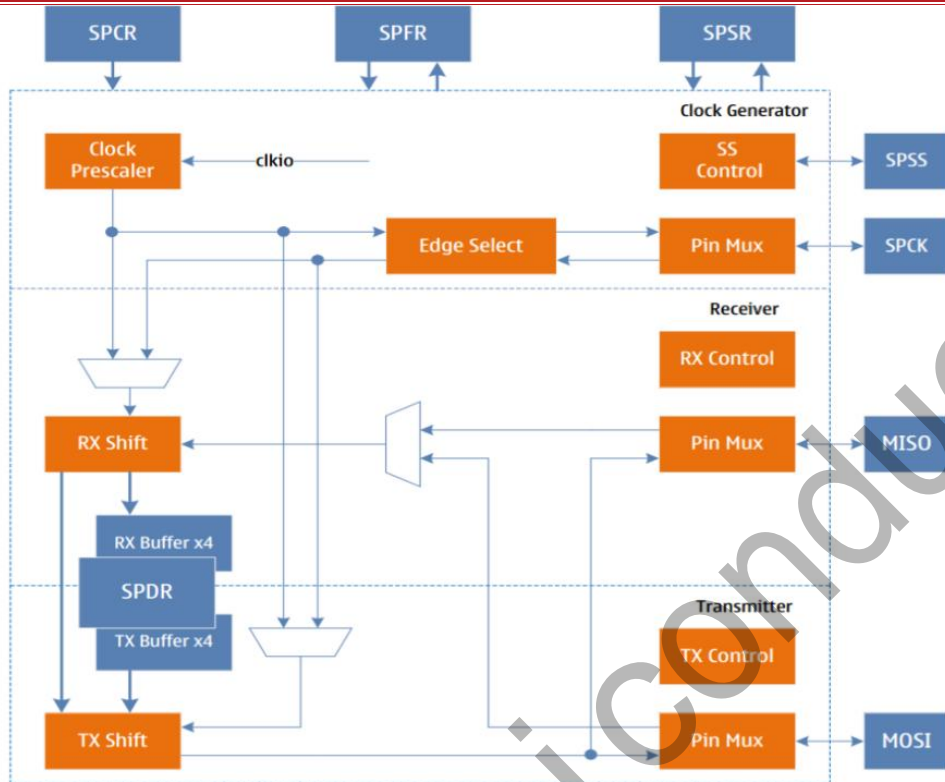
CPHA 为“1”时 SPI 数据传输模式

双线模式下，MISO 和 MISO 均用做主机的输入，数据采样的时刻仍由数据传输模式决定，采样的方式如下图所示：



主机模式下 DUAL 为“1”时 SPI 数据采样模式

#### 17.4. SPI 电路结构图



SPI 结构图

### 17.5. SPSS 引脚功能

当配置为从机时，从机选择信号 SPSS 引脚总是作为输入。当 SPSS 引脚保持为低时，SPI 接口被激活，MISO 引脚成为输出引脚（软件进行相应的端口配置），其它引脚均为输入。当 SPSS 引脚保持为高时，SPI 模块被复位，且不再接收数据。SPSS 引脚对于数据包/字节的同步非常有用，可以使从机的位计数器和主机的时钟发生器同步。当 SPSS 拉高时，SPI 从机立即复位接收和发送逻辑，并丢弃移位寄存器里不完整的数据。

当配置为主机时，用户软件可以决定 SPSS 引脚的方向。

若 SPSS 配置为输出，则它可以用来驱动从机的 SPSS 引脚。若 SPSS 配置为输入，必须保持为高以保证主机的正常工作。当配置为主机且 SPSS 引脚为输入，外部电路拉低 SPSS 引脚时，SPI 模块会认为是另外一个主机选择自己作为从机并开始传输数据。为了防止总线冲突，SPI 模块将进行如下动作：

1. 清零位于 SPCR 寄存器的 MSTR 位，转换为从机，从而 MOSI 和 SPCK 变为输入；
2. 置位位于 SPSR 寄存器的 SPIF 位，若中断使能则产生 SPI 中断。

因此，使用中断方式处理 SPI 主机的数据传输，并且存在 SPSS 被拉低的可能性时，中断服务程序应该检查 MSTR 位是否为“1”。若被清零，软件须将其置位，以重新使能 SPI 主机模式。

### 17.6. SPI 初始化

进行通信之前首先要对 SPI 进行初始化。初始化过程通常包括主机从机操作的选择，数据传输模式的设定，比特率的选择，以及各个引脚的方向控制等。其中主机和从机操作下引脚方向的控制各不相同，如下表所示：

引脚方向控制

引脚	主机模式下的方向	从机模式下的方向
MOSI	用户软件定义	输入
MISO	输入	用户软件定义
SPCK	用户软件定义	输入

SPSS	用户软件定义	输入
------	--------	----

### 17.7. SPI 主机初始化

SPI 主机模式的初始化过程如下：

1. 置位 MSTR 位，设置比特率选择控制位，数据传输模式，数据传输次序，中断使能与否，以及双线使能与否；
2. 设置 MOSI 和 SPCK 引脚为输出；
3. 置位 SPE 位。

主机模式下，当不希望 SPI 模块被别的主机选择作为从机使用时，可设置 SPSS 引脚为输出。

### 17.8. SPI 从机初始化

SPI 从机模式初始化过程如下：

1. 清零 MSTR 位，设置数据传输模式，数据传输次序，中断使能与否；
2. 设置 MISO 引脚为输出；
3. 置位 SPE 位。

### 17.9. SPI 中断

当发生下列事件之一或多个时，SPI 的中断标志位 SPIF 将会被置位：

1. 当配置为主机且 SPSS 引脚为输入，外部电路拉低 SPSS 引脚；
2. 当发送缓冲器状态为满，软件继续往 SPDR 寄存器写入数据；
3. 当接收缓冲器状态为满；
4. 当写入发送缓冲器中的数据均已发送出去，发送缓冲器状态为空。

当 SPIF 位被置位，且 SPI 中断使能位 SPIE 和全局中断使能位都为高时，会产生 SPI 中断。进入中断服务程序后，硬件会对 SPIF 进行清零。若 SPIF 位是由上述事件中的 1 和 2 来置位的，SPIF 会被清零；若 SPIF 位是由上述事件中的 3 和 4 来置位的，SPIF 并不会被清零，因为接收或发送缓冲器状态未发生改变时，仍会置位 SPIF 位，此时需要通过软件操作来清零。

SPI 中断服务程序中，软件清零 SPIF 位的操作顺序如下：

- 1) 读取 SPIF 位的状态，若为低，表明 SPIF 位已被硬件清零，无需软件再次清零；若为高，继续一下操作；
- 2) 读取 SPFR 寄存器，若 RDFULL 位为高，表明当前接收缓冲器状态为满，读取 SPDR 寄存器获得接收数据，RDFULL 位会变为低，软件可继续读取 SPDR 寄存器获得接收数据，直到 RDEMPT 位为高；
- 3) 读取 SPFR 寄存器，若 RDFULL 位为低，而 WREMP 位为高，表明当前接收缓冲器状态为 非满，而发送缓冲器状态为空，软件可读取 SPDR 寄存器获得接收数据，直到 RDEMPT 位 为高；
- 4) 软件获取所接收到的数据后，再执行清零 SPIF 位。因 SPIF 位为只读位，不能直接对 SPIF 位进行清零，而需要先读取 SPSR 寄存器，再访问 SPDR（读或写 SPDR 寄存器）的方式 来清零 SPIF 位。

### 17.10. SPCK 和系统时钟的关系

SPI2X	SPR1	SPR0	SPCK 的频率
0	0	0	$f_{sys}/4$
0	0	1	$f_{sys}/16$
0	1	0	$f_{ss}/64$
0	1	1	$f_{ys}/128$
1	0	0	$f_{sys}/2$

1	0	1	fss/8
1	1	0	fsys/32
1	1	1	fss/64

### 17.11. SPI 寄存器列表

寄存器	地址	默认值	描述
SPCR	0x4C	0x00	SPI 控制寄存器
SPSR	0x4D	0x00	SPI 状态寄存器
SPDR	0x4E	0x00	SPI 数据寄存器
SDFR	0x4F	0x00	SPI 缓冲寄存器

SPDR-SPI 数据寄存器

SPDR-SPI 数据寄存器								
地址: 0x4E					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	SPDR7	SPDR6	SPDR5	SPDR4	SPDR3	SPDR2	SPDR1	SPDR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
bit	Name	描述						
7:0	SPDR	SPI 发送和接收的数据。 SPI 发送数据和接收数据共享 SPI 数据寄存器 SPDR。将数据写入 SPDR 即写入发送数据移位寄存器，从 SPDR 读取数据即读取接收数据缓冲器。						

SPCR-SPI 控制寄存器

SPCR-SPI 控制寄存器								
地址: 0x4C					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	SPIE	SPI 中断使能位。 当设置 SPIE 位为“1”时，SPI 中断被使能。当位于 SPSR 寄存器中的 SPIIF 位被置位且全局中断使能时，产生 SPI 中断。 当设置 SPIE 位为“0”时，SPI 中断被禁止。						
6	SPE	SPI 使能位。 当设置 SPE 位为“1”时，SPI 模块被使能。进行任何 SPI 操作之前必须置位 SPE。 当设置 SPE 位为“0”时，SPI 模块被禁止。						

5	DORD	数据次序控制位。 当设置 DORD 位为“1”时，数据的 LSB 首先发送。 当设置 DORD 位为“0”时，数据的 MSB 首先发送。									
4	MSTR	主机从机选择控制位。 当设置 MSTR 位为“1”时，选择为主机模式。 当设置 MSTR 位为“0”时，选择为从机模式。 主机模式下，SPSS 引脚配置为输入且被拉低时，MSTR 位将被清零，位于 SPSR 寄存器的 SPIF 被置位，用户必须重新设置 MSTR 进入主机模式。									
3	CPOL	时钟极性控制位。 当设置 CPOL 位为“1”时，空闲状态下 SPCK 为高电平。 当设置 CPOL 位为“0”时，空闲状态下 SPCK 为低电平。									
		<table border="1"> <thead> <tr> <th>CPOL</th> <th>起始沿</th> <th>结束沿</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>上升沿</td> <td>下降沿</td> </tr> <tr> <td>1</td> <td>下降沿</td> <td>上升沿</td> </tr> </tbody> </table>	CPOL	起始沿	结束沿	0	上升沿	下降沿	1	下降沿	上升沿
		CPOL	起始沿	结束沿							
0	上升沿	下降沿									
1	下降沿	上升沿									
2	A	时钟相位控制位。 当设置 CPHA 位为“1”时，起始沿设置数据，结束沿采样数据。当设置 CPHA 位为“0”时，起始沿采样数据，结束沿设置数据。									
		<table border="1"> <thead> <tr> <th>CPHA</th> <th>起始沿</th> <th>结束沿</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>采样</td> <td>设置</td> </tr> <tr> <td>1</td> <td>设置</td> <td>采样</td> </tr> </tbody> </table>	CPHA	起始沿	结束沿	0	采样	设置	1	设置	采样
		CPHA	起始沿	结束沿							
0	采样	设置									
1	设置	采样									
1	SPR1	时钟速率选择位 1。 SPR1 和 SPR0 用来选择 SPI 传输的时钟速率。具体控制方式见 SPCK 和 系统时钟的关系表格。									
0	SPR0	时钟速率选择位 0。 SPR1 和 SPR0 用来选择 SPI 传输的时钟速率。具体控制方式见 SPCK 和 系统时钟的关系表格。									

SPSR-SPI 状态寄存器

SPSR-SPI 状态寄存器								
地址: 0x4D					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	SPIF	WCOL	-	-	-	DUAL	-	SPI2X
R/W	R	R	R	R	R	R/W	R	R/W
Bit								
Bit	Name	描述						
7	SPIF	SPI 中断标志位。 串行传输结束后置位 SPIF 标志，主机模式下，配置 SPSS 引脚为输入且被拉低时，SPIF 也将被置位。若此时 SPDR 寄存器的 SPIE 位和全局中断使能位都被置位，SPI 中断产生。进入中断服务程序后 SPIF 位自动清零，或者通过先读取 SPSR 寄存器再访问 SPDR 寄存器来清零 SPIF 位。						
6	WCOL	写冲突标志位。 在数据传输的过程中写 SPDR 寄存器将置位 WCOL 位。WCOL 位可以通过先读取 SPSR 寄存器再访问 SPDR 寄存器来清零。						



5	-	保留。
4	-	保留。
3	-	保留。
2	DUAL	<p>双线模式控制位。</p> <p>当设置 DUAL 位为“1”时，使能 SPI 双线传输模式。</p> <p>当设置 DUAL 位为“0”时，禁止 SPI 双线传输模式。</p> <p>双线传输模式只在 SPI 主机模式下有效，MISO 和 MOSI 均用作主机数据输入，数据的传输方式见主机双线接收和数据模式章节描述。</p>
1	-	保留。
0	SPI2X	<p>SPI 倍速控制位。</p> <p>当设置 SPI2X 位为“1”时，SPI 的传输速度加倍。</p> <p>当设置 SPI2X 位为“0”时，SPI 的传输速度不加倍。</p> <p>具体控制方式见 SPCK 和系统时钟的关系表格。</p>

SPFR - SPI 缓冲寄存器

TWCR - TWI 控制寄存器

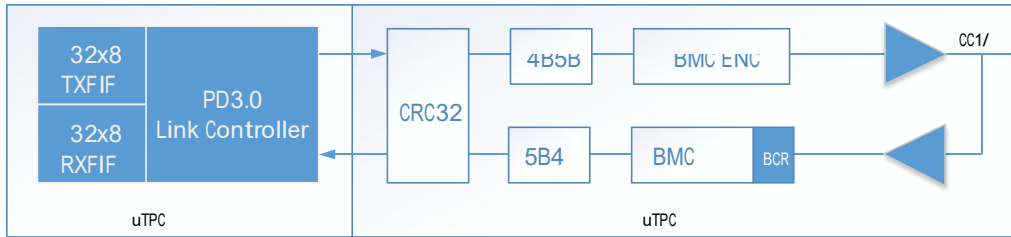
地址: 0x39				默认值: 0x00				
Bit	7	6	5	4	3	2	1	0
	RDFUL I	RDEMPT	RDPTR1	RDPTRO	WRFULL	WREMPT	WRPTR1	WRPTRO
R/W	R	R/W	R	R	R	R/W	R	R
Bit	Name		描述					
7	RDFUL I		<p>接收缓冲器满标志位</p> <p>当接收缓冲器中的数据达到四个字节时，RDFULL 位为高，表明接收缓冲器为满，同时会置位中断标志位。若软件未及时读走接收缓冲器中的数据，再次接收到数据时，接收缓冲器发生溢出，之前的数据会被新数据覆盖。</p> <p>当接收缓冲器中的数据少于四个字节时，RDFULL 位为低，表明接收缓冲器为非满，还可以接收数据。</p> <p>当同时对 RDEMPT 位和 WREMPT 位进行置位操作时，接收和发送缓冲器地址以及 SPI 移位寄存器指针都将归零，RDFULL 位为低</p>					
6	RDEMPT		<p>接收缓冲器空标志位</p> <p>当未接收到数据时，RDEMPT 位为高，表明接收缓冲器空。</p> <p>当有接收到数据时，会存入接收缓冲器，RDEMPT 位为低，表明接收缓冲器为非空，此时 MCU 可通过访问 SPDR 寄存器来读取接收缓冲器中的数据。为确保所接收到的数据不会丢失，软件可在接收缓冲器为非空状态即 RDEMPT 位为低时读走接收缓冲器中的数据。</p> <p>当对 RDEMPT 位进行置位操作（写 1）时，接收缓冲器地址将归零。</p>					

5	RDPTR1	接收缓冲器地址高位。
4	RDPTRO	接收缓冲器地址低位。  当对 SPDR 寄存器进行读操作时，MCU 将会从接收缓冲器中读到所接收的数据，同时接收缓冲器地址会进行累加。  当对 RDEMP1 位进行置位操作（写 1）时，接收缓冲器地址将归零。
3	WRFULL	发送缓冲器满标志位。 当发送缓冲器中的数据达到四个字节时，WRFULL 位为高，表明发送缓冲器为满。 当发送缓冲器中的数据少于四个字节时，WRFULL 位为低，表明发送缓冲器为非满。若想提高传输速度，软件可在发送缓冲器为非满状态即 WRFULL 位为低时写入数据，SPI 控制器会依次把数据发送出去。
2	WREMP1	发送缓冲器空标志位。 当写入发送缓冲器的数据均已发送完毕时，WREMP1 位为高，表明发送缓冲器为空，同时会置位中断标志位 SPIF。 当对 SPDR 寄存器进行写操作后，发送缓冲器地址会累加，写入发送缓冲器的数据未被全部发送时，接收缓冲器中至少有一个字节的数据，WREMP1 位为低，表明发送缓冲器非空。 当对 WREMP1 位进行置位操作（写 1）时，发送缓冲器地址将归零。 当同时对 RDEMP1 位和 WREMP1 位进行置位操作时，接收和发送缓冲器地址以及 SPI 移位寄存器指针都将归零，WREMP1 位为高。
1	WRPTR1	发送缓冲器地址高位。
0	WRPTRO	发送缓冲器地址低位。  当对 SPDR 寄存器进行写操作时，SPDR 中的数据将会被写入发送缓冲器，  同时发送缓冲器地址会进行累加。  当对 WREMP1 位进行置位操作（写 1）时，发送缓冲器地址将归零。

## 18. USB Type-C PD 收发器 (uTPC0/1)

- BMC/4b5b 编解码
- CRC32 收发校验
- 自动 GoodCRC 响应
- 支持自动生成 MessageID
- 支持自动重发机制
- 支持 PD3.0 所有传输类型
- 收发器支持独立 32 字节 FIFO
- 支持 FIFO 中断以及传输中断
- USB PD3.0 物理层模拟前端
- 支持 Ip/Rd
- 内置 VCONN 开关，限流 200mA
- 支持死电池 (dead battery)
- 支持 CC1/CC2 唤醒

## 18.1. 概述



uTPC 控制器框架图

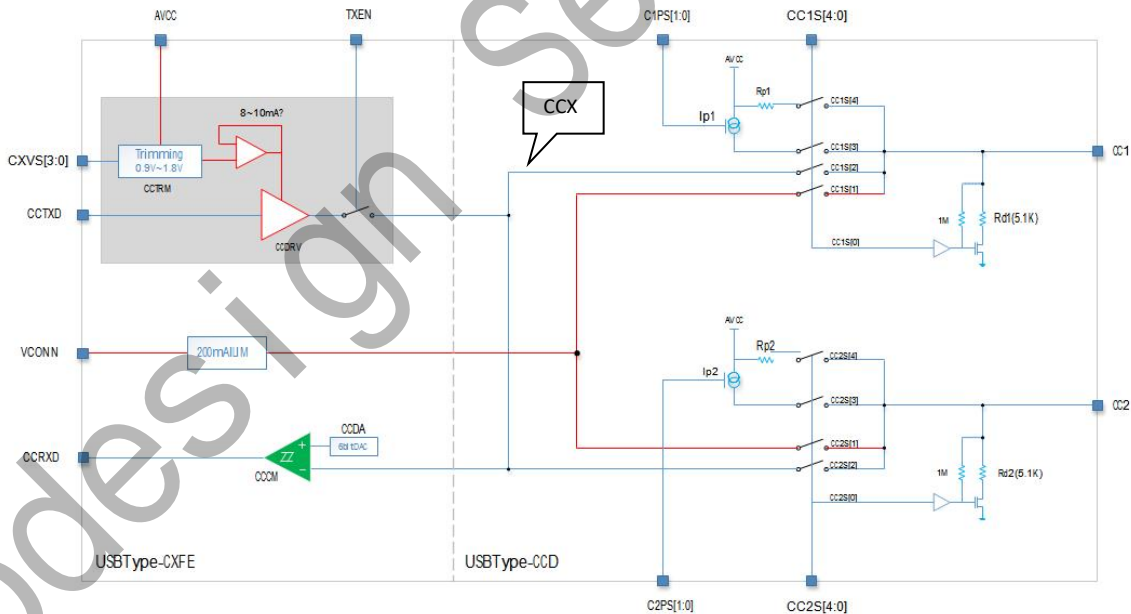
如图所示，uTPC 包括 PD3.0 链路层控制器(Link)以及 PD3.0 物理收发器(PHY)两个部分。其中链路控制器部分提供了最基本的 PD3.0 协议层通讯框架。链路层控制器主要实现协议封包，messageID 生成，自动 goodCRC 回应以及异常管理部分。链路层通过 FIFO 接口以及异常中断将复杂且不具备实时性要求的协议逻辑交由处理器处理。

物理层部分包括传输相关的编解码以及 CC 收发驱动。在数据发送模式下，从链路层封包的数据首先进行必要的 CRC32 运算，产生 CRC 字段；然后经过 4B5B 编码，之后是 BMC 编码，产生最终用于线路传输的 BMC 码流。BMC 驱动 CC 发送器，将协议数据传输到 CC 线路。

在数据接收模式下，来自 CC1/2 的数据首先经过物理层的接收器，将有效的 CC 数据解调。然后物理层控制器时钟生成部分(BCR)从 CC 码流的同步头部分恢复时钟信息，BMC 解码逻辑使用 BCR 产生的同步时钟解码，之后将数据进行 4B5B 的解码处理，产生原始的数据。数据最后经过 CRC 运算，判断数据包是否完整正确的接收。如果 CRC 校验通过，数据交由链路控制器处理，触发 GoodCRC 响应，并将数据包写入接收 FIFO。如果数据接收的任意过程中 (BCR, BMC, 4B5B, CRC) 发送异常，uTPC 将根据异常种类产生自动响应，并及时通过中断机制报告处理器。

## 18.2. 收发器前端

uTPC 收发器前端是物理层控制器(PHY)的传输驱动部分，此部分实现 Type C 接口协议对 CC 传输的定义，也为 PD 协议提供了必要的电气特性支持。



uTPC 收发器前端的功能框图

CC 收发器主要有两个功能部分，分别为 CC 收发驱动和 Type C 的线缆检测电路。

CCDRV 电路包括一个 16 级可调输出电压选择和一个驱动器。CCDRV 输出电压可以从 0.9V 调整到最高 1.8V，用于支持 PD3.0 协议对数据传输的电平定义。

Type-C 的线缆检测电路部分包括两路功能完全相同的电路。这部分电路主要实现 Type C 协议中对建立供电线路的协议支持。其中主要是包括一个可控制的电流源  $I_p$  以及一个可控制的 5.1K 的下载电阻  $R_d$ 。软件通过设置  $I_p$  以及  $R_d$  实现 Type-C 供电或者受电设备

之间的切换以及接入设备检测。除了  $I_p$  和  $R_d$  外，这部分电路还包括用于  $V_{CONN}$  供电电源开关，用于支持死电池的  $R_d$  控制器以及用于低功耗唤醒的弱上拉 ( $R_p$ )。

### 18.3. 物理层控制器

uTPC 物理层控制器实现 PD 协议对物理层和协议层的定义。这部分相关的功能请参考 USB PD R3.0 协议中第 5, 6 章节。这里我们简述物理层控制器所实现的功能。

物理层控制器分为收发独立的两条链路。这两条逻辑链路除了共享同一个 CRC32 运算模块以外，其他功能模块各自独立。

### 18.4. PD 数据发送流程

PD 数据发送由软件控制下的 uTPC 链路控制器启动。发起传输之前，软件需要配置传输数据帧相关的消息头定义 (Message Header)。发起传输后，物理层控制器首先检测 CC 线路状态，等待空闲时发起传输。物理层控制器首先发送固定长度的前导字符 (Preamble)，然后是预先设定的  $SOP/SOP' /SOP''$ ，紧接着物理层从链路控制器收集消息头 (Message Header)，完成消息头的封包。消息头之后，物理层根据消息头中的定义，决定接下来的数据是扩展消息头 (Extended Message Header) 还是数据包 (Payload)。传输数据时，物理层控制器与链路层协作，从发送 FIFO 中依次读出发送的数据，由物理层控制器完成 CRC 计算。最终数据由物理层完成 4B5B 编码，驱动收发器前端电路发送数据。

数据发送完成后，链路层启动 GoodCRC 定时器，等待接收端返回 GoodCRC 信息。如果在定时器超时之前收到有效的 GoodCRC 反馈，完成数据传输，链路层通过更新发送结束状态信息，如果使能了相应的中断，则触发中断请求。如果 GoodCRC 定时器超时，链路层将立刻启动重试，物理层开始一次新的发送流程，同时链路层中断通知系统，软件需要重新填充发送 FIFO。由于首先需要发送一定长度的前导字符，处理器有足够的时间完成填充 FIFO 数据。

如果数据发送重试失败，链路层将会结束本次数据传输，并更新传输标记信息。如果使能了中断，控制器将会同时发起中断请求。

### 18.5. PD 数据接收流程

当 uTPC 处于接收模式时，物理层控制器监控 CC 线路，当发现有效的开始位后，启动数据接收。物理层控制的时钟恢复模块 (BCR) 开始工作。BCR 检测 CC 数据，从前导数据中建立接收同步时钟，并监控前导字节。前导数据接收完毕后，后续的数据接收模块将使用 BCR 产生的波特率时钟继续接收接下来的数据。接下来的数据接收过程是由物理层控制器和链路层共同完成。物理层完成数据的串并转换，4B5B 解码并同时计算 CRC。在传输过程中，链路层控制器负责解析理解数据包信息。根据数据包信息更新相关寄存器状态，并将有效的数据信息写入接收 FIFO。整包数据接收完成后，链路层更加最终的 CRC 结果确定数据接收是否正确。如果数据包通过 CRC 检测，链路层立刻通知物理层控制器回应一个 GoodCRC 传输。

如果在数据接收的过程中出现错误，物理层或者链路层都可能会随时中断数据接收，并通过标记位或者中断通知系统。

### 18.6. 中断请求

uTPC 可根据当前的工作状态，产生 3 种中断请求：

传输控制器中断

当 uTPC 完成一次有效的数据发送或者接收，将会更新 uTPC 传输完成中断标志位。如果系统使能了 uTPC 传输完成中断，将会同时想处理器发出中断请求。软件需要通过 TPCON2 寄存器中的 TXDN 或者 RXDN 位，确定发出中断请求的传输类型。

发送 FIFO 将空中断

发送 FIFO 剩下 4 个字节数据时，产生发送 FIFO 将空的中断请求。当一次发送的数据长度大于发送 FIFO 的长度时，我们需要分多次填充发送 FIFO。这个发送 FIFO 将空的中断请求，用于及时告知软件 FIFO 将空，需要及时填充后续的数据。

接收 FIFO 半满中断

接收 FIFO 半满时中断请求。在接收数据时，当接收到的数据填到 FIFO 半满时，uTPC 向处理器发出接收 FIFO 半满中断，软件响应此中断，及时读出接收的数据，保证接收 FIFO 时刻有充足的空间接收接下来的数据。

### 18.7. 寄存器定义

名称	I/O 地址	功能描述
TPCSIG	0x92	TPC 设备标识寄存器
CCACR0	0x93	TPC 辅助控制寄存器0
CCACR1	0x94	TPC 辅助控制寄存器1

CCACR2	0x95	TPC 辅助控制寄存器2
CCACR4	0x97	TPC 辅助控制寄存器4
TPCTXFF	0x98	TPC 发送FIFO 写数据接口
TPCRXFF	0x99	TPC 接收FIFO 读数据接口
TPCON0	0x9A	TPC 控制寄存器0
TPCON1	0x9B	TPC 控制寄存器1
TPCON2	0x9C	TPC 控制寄存器2
KCMSK	0x9D	K-code 控制寄存器
TPCBDR	0x9E	TPC 发送波特率控制寄存器
TPCSR0	0x9F	TPC 状态寄存器0
TPCIF	0x38	TPC 中断标志寄存器
TPCIER	0x91	TPC 中断使能寄存器
TPC1IER	0xD1	TPC1 中断使能寄存器
TPC1SIG	0xD2	TPC1 设备标识寄存器
GC1ACR0	0xD3	TPC1 辅助控制寄存器0
GC1ACR1	0xD4	TPC1 辅助控制寄存器1
GC1ACR2	0xD5	TPC1 辅助控制寄存器2
GC1ACR3	0xD6	TPC1 辅助控制寄存器 3
GC1ACR4	0xD7	TPC1 辅助控制寄存器4
TPC1TXFF	0xD8	TPC1 发送FIFO 写数据接口
TPC1RXFF	0xD9	TPC1 接收FIFO 读数据接口
TP1CON0	0xDA	TPC1 控制寄存器0
TP1CON1	0xDB	TPC1 控制寄存器1
TP1CON2	0xDC	TPC1 控制寄存器2
KCMSK1	0xDD	K-code 控制寄存器 1
TPC1BDR	0xDE	TPC1 发送波特率控制寄存器
TPC1SR0	0xDF	TPC1 状态寄存器0

TPCSR0 -TPC 状态寄存器 0

TPCSR0 - uTPC 状态寄存器

地址: 0x9F		默认值: 00000000						
Bit	7	6	5	4	3	2	1	0
Name	-	TRYD	GCE	RFFE	CRCE	KCE	BMCE	BCRE
R/W	-	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Bit	Name	描述						
7	-	保留未用						
6	TRYD	发送重试结束						
5	GCE	GoodCRC 定时器超时						
4	RXFF	接收FIFO 数据已满						
3	CRCE	接收数据CRC 错误						
2	KCE	接收数据Kcode 编码出错						
1	BMCE	接收数据BMC 编码出错						



0	BCRE	接收数据波特率恢复出错
---	------	-------------

TPC1SR0 - TPC1 状态寄存器 0

TPC1SR0 - uTPC 状态寄存器

地址: 0xDF						默认值: 00000000		
Bit	7	6	5	4	3	2	1	0
Name	-	TRYD	GCE	RFFE	CRCE	KCE	BMCE	BCRE
R/W	-	R/O	R/O	R/O	R/O	R/O	R/O	R/O
Bit	Name	描述						
7	-	保留未用						
6	TRYD	发送重试结束						
5	GCE	GoodCRC 定时器超时						
4	RXFF	接收FIFO 数据已满						
3	CRCE	接收数据CRC 错误						
2	KCE	接收数据Kcode 编码出错						
1	BMCE	接收数据BMC 编码出错						
0	BCRE	接收数据波特率恢复出错						

TPCBDR - PD 发送波特率寄存器

TPCBDR - PD 数据发送波特率寄存器

地址: 0x9E						默认值: 0x7B		
Bit	7	6	5	4	3	2	1	0
Name	CCIDLC				CCBDR[4:0]			
R/W	R/W				R/W			
Bit	Name	描述						
7:5	CCIDLC	CC 总线空闲周期设置, 以 PD 数据发送的波特率为时钟的计时设置 001 = 4 个CC总线周期 010 = 8 个CC总线周期 011 = 12 个CC总线周期 (默认) ... 111 = 28 个CC总线周期						
4:0	CCBDR	PD 数据发送波特率寄存器。 TPCBDR 用于调整 PD 发送数据的波特率。PD 数据发送波特率发生器工作与内部 24MHz RC 的4分频, 也就是6MHz, 默认波特率为: $6000\text{KHz}/27 = 222\text{KHz}$ , 建议设置为20以产生300KHz 的波特率						

KCMSK - KCode 屏蔽寄存器

KCMSK - KCODE 屏蔽寄存器

地址: 0x9D						默认值: X111_1111		
Bit	7	6	5	4	3	2	1	0
Name	FFR	CRE	HRE	SOP2DE	SOP1DE	SOP2E	SOP1E	SOPE
R/W	W/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	FFR	写1 将复位收发FIFO 状态						



6	CRE	写1 使能线复位请求数据帧
5	HRE	写1 使能硬复位请求数据帧
4	SOP2DE	写1 使能SOP' ' _Debug 数据帧
3	SOP1DE	写1 使能SOP' _Debug 数据帧
2	SOP2E	写1 使能SOP' ' 数据帧
1	SOP1E	写1 使能SOP' 数据帧
0	SOPE	写1 使能SOP 数据帧

TPCSIG - PD 设备模式标识寄存器

TPCSIG - PD 设备模式标识寄存器

地址: 0x92						默认值: 0XXX_1101		
Bit	7	6	5	4	3	2	1	0
Name	CCDLS	-	-	-	PPS	SR1	SR0	PDS
R/W	R/W	-	-	-	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	CCDLS	CC/IDLE 状态源选择位 0 = IDLE 状态来自CC 数据比较器 1 = IDLE 状态来自CV 电平比较器						
6:4	-	保留未用						
3	PPR	端口电源角色 (Port Power Role). 1 = Sink, 0 = Source (仅针对SOP 数据包) 1 = 信息来自DFP/UFP, 0 = 消息来自线缆插头 (仅针对Cable Plug)						
2:1	SR	兼容标准版本, 默认10 = Revision 3.0						
0	PDR	端口数据角色 (Power Data Role), 仅对SOP 数据包有效 1 = DFP, 0 = UFP						

TPCON0 - TPC 控制寄存器 0

TPCON0 - TPC 控制寄存器0

地址: 0x9A						默认值: 0000_0111		
Bit	7	6	5	4	3	2	1	0
Name	TPCEN	TXEN	TXST	RXEN	AIDEN	GIDEN	RXM	AGCEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	TPCEN	uTPC 模块使能控制。 写1 使能TPC 模块, 清0 将关闭uTPC 模块的工作时钟						
6	TXEN	写1 使能uTPC 发送功能模块						
5	TXST	写1 将启动一次数据发送						
4	RXEN	写1 使能uTPC 数据接收模块						
3	AIDEN	写1 使能自动消息 ID 硬件产生。 使能后, 有uTPC 的发送器自动产生和更新Message ID						
2	GIDEN	写1 使能对接收GoodCRC 数据包的ID 确认。 当接收到的GoodCRC 的ID 与发送的数据ID 相同时, GoodCRC 为有效。						



1	RXM	接收数据是否包含CRC 部分。 写1, 将CRC 数据一同写入接收FIFO 清0, CRC数据将不会被写入FIFO
0	AGCEN	写1 使能自动GoodCRC 响应。 当接收到正确的CRC 数据后, 硬件自动回应GoodCRC 响应数据包

TPCON1 - TPC 控制寄存器 1

TPCON1 - TPC 控制寄存器1

地址: 0x9B						默认值: 0000_0111		
Bit	7	6	5	4	3	2	1	0
Name	TXREN	TXKC2	TXKC1	TXKC0	RSYNC	RXKC2	RXKC1	RXKC0
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/O
Bit	Name	描述						
7	TXREN	发送出错重试使能。发送出错后, 将会重试发送2 次						
6:4	TXKC[2:0]	PD 数据发送KCODE 设置 000 = SOP 001 = SOP' 010 = SOP' ' 011 = SOP' _DEBUG 100 = SOP' ' _DEBUG 101 = 硬件复位请求 110 = 线缆复位请求						
3	RSYNC	写 1 使能接收同步模式。uTPC 将在检测到 CC 空闲状态时进入接收状态						
2:0	RXKC[2:0]	PD当前接收到数据包KCODE 类型 000 = SOP 001 = SOP' 010 = SOP' ' 011 = SOP' _DEBUG 100 = SOP' ' _DEBUG 101 = 硬件复位请求 110 = 线缆复位请求						

TPCON2 - TPC 控制寄存器 2

TPCON2 - TPC 控制寄存器2

地址: 0x9C						默认值: 0100_0000		
Bit	7	6	5	4	3	2	1	0
Name	CCRN	GCVEN	RXFM	CCIDL	TXFF	RXFE	TXDN	RXDN
R/W	R/W	R/W	R/W	R/O	R/O	R/O	R/W	R/W
Bit	Name	描述						
7	CCRN	写1 将执行CC 软件复位 将会复位 uTPC 物理层控制器和链路控制器的所有状态						
6	GCVEN	写1 使能GoodCRC 对链路层可见。 清 0 后, GoodCRC 仅在物理层控制器内完成, 链路层控制器不可见						



5	RXFM	接收FIFO 中断模式。 0 = 半满中断, 1 = 非空中断
4	CCIDL	CC 空闲状态标志位
3	TXFF	发送FIFO 满标记位
2	RXFE	接收FIFO 空标记位
1	TXDN	发送结束标记位。启动发送时自动清零, 软件可以通过写0 清零
0	RXDN	接收结束标记位。启动接收时自动清零, 软件可以通过写0 清零

TPCTXFF - uTPC 发送 FIFO 访问寄存器

TPCTXFF - uTPC 数据发送FIFO 访问寄存器

地址: 0x98						默认值: XX			
Bit	7	6	5	4	3	2	1	0	
Name	TPCTXFF								
R/W	W/O								
Bit	Name	描述							
7:0	TPCTXFF	访问uTPC 内部数据发送FIFO 的接口。 软件通过写TPCTXFF 寄存器, 实现向FIFO 中写数据。通过查询TPCON2 寄存器 中的TXFF 为判断发送FIFO 是否已经写满。							

TPCRXFF - uTPC 接收 FIFO 访问寄存器

TPCRXFF - uTPC 数据接收FIFO 访问寄存器

地址: 0x99						默认值: XX			
Bit	7	6	5	4	3	2	1	0	
Name	TPCRXFF								
R/W	R/O								
Bit	Name	描述							
7:0	TPCRXFF	访问uTPC 内部数据接收FIFO 的接口。 软件通过读TPCRXFF 寄存器, 从接收FIFO 中读出接收数据。通过查询TPCON2 寄存器中的RXFE 为判断发送FIFO 是否为空。							

TPCIER - TPC 外设中断使能寄存器

TPCIE - TPC 外设中断使能寄存器

地址: 0x91						默认值: 0000_0000			
Bit	7	6	5	4	3	2	1	0	
Name	-	-	-	-	-	TPIE	TXIE	RXIE	
R/W	-	-	-	-	-	W/R	W/R	W/R	
Bit	Name	描述							
7:3	-	保留未用							
2	TPCIE	uTPC 传输完成中断使能控制位							
1	TXFIE	uTPC 发送FIFO 中断使能控制位							
0	RXFIE	uTPC 接收FIFO 中断使能控制位							



TPCIF - TPC 外设中断标志寄存器

TPCIF - TPC 外设标志寄存器								
地址: 0x38						默认值: 0000_0000		
Bit	7	6	5	4	3	2	1	0
Name	-	TP1IF	TX1IF	RX1IF	-	TPCIF	TXFIF	RXFIF
R/W	-	WZ/R	WZ/R	WZ/R	-	WZ/R	WZ/R	WZ/R
Bit	Name	描述						
7	-	保留未用						
6	TP1IF	TPC1 传输完成中断标志位, 写0 清零						
5	TX1IF	TPC1 发送FIFO 中断标志位, 写0 清零						
4	RX1IF	TPC1 接收FIFO 中断标志位, 写0 清零						
3	-	保留未用						
2	TPCIF	TPC 传输完成中断标志位, 写0 清零						
1	TXFIF	发送FIFO 中断标志位, 写0 清零						
0	RXFIF	TPC 接收FIFO 中断标志位, 写0 清零						

CCACR0 -Type-C CC 控制寄存器 0

CCACR0 - Type-C CC 控制寄存器0								
地址: 0x93						默认值: 00001101		
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	RXTR1	RXTR0	-	-
R/W	-	-	-	-	R/W	R/W	-	-
Bit	Name	描述						
7	-	保留未用						
6	IPEN	内部恒流源上拉使能						
5	CCEN	CC 收发数据使能						
4	-	保留未用						
3:2	RXTR	输入延时校准位 00: 采用15 根电阻串延时 01: 采用6 根电阻串延时 10: 采用4 根电阻串延时 11: 采用2 根电阻串延时						
1:0	-	保留未用						

CCACR1 -Type-C CC 控制寄存器 1

CCACR1 - Type-C CC 控制寄存器1								
地址: 0x94						默认值: 00000000		
Bit	7	6	5	4	3	2	1	0
Name	IP1S1	IP1S0	-	C1CH4	C1CH3	C1CH2	C1CH1	C1CH0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						



7:6	IP1S	CC1 上拉电流源选择控制位 00 = 80uA 01 = 180uA 1x = 330uA
5	-	保留未用
4	C1CH4	CC1 接内部上拉电阻, 用于低功耗休眠的电平变化唤醒
3	C1CH3	CC1 接内部Ip 上拉电流源
2	C1CH2	CC1 接内部CC 数据检测比较器
1	C1CH1	CC1 接VCONN, 接内部LDO 输出VDD5
0	C1CH0	CC1 接5.1K 下拉电阻

CCACR2 -Type-C CC 控制寄存器 2

CCACR2 - Type-C CC 控制寄存器2

地址: 0x95						默认值: 00000000		
Bit	7	6	5	4	3	2	1	0
Name	IP2S1	IP2S0	-	C2CH4	C2CH3	C2CH2	C2CH1	C2CH0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:6	IP2S	CC2 上拉电流源选择控制位 00 = 80uA 01 = 180uA 1x = 330uA						
5	-	保留未用						
4	C2CH4	CC2 接内部上拉电阻, 用于低功耗休眠的电平变化唤醒						
3	C2CH3	CC2 接内部Ip 上拉电流源						
2	C2CH2	CC2 接内部CC 数据检测比较器						
1	C2CH1	CC2 接VCONN, 接内部LDO 输出VDD5						
0	C2CH0	CC2 接5.1K 下拉电阻						

CCACR3 -Type-C CC 控制寄存器 3

CCACR3 - Type-C CC 控制寄存器3

地址: 0x96						默认值: 00000000		
Bit	7	6	5	4	3	2	1	0
Name	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
0								
0								

CCACR4 -Type-C CC 控制寄存器 4

CCACR4 - Type-C CC 控制寄存器4

地址: 0x97						默认值: 00000000		
Bit	7	6	5	4	3	2	1	0
Name	ECPL	CVTE	CSS1	CSS0	CCVS3	CCVS2	CCVS1	CCVS0



R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	ECPL	CC数据极性控制						
6	CVTE	CVAC 校准使能控制						
5:4	CSS	CC接收信号源选择 00/11 = CC接收来自PDC数据比较器(CGAC) 01 = 接GND低电平(测试用) 10 = CC接收来自通用比较器GPCM1的输出						
3:0	CCVS	PD通讯发送数据的CC驱动电平控制 0000 = CC驱动电压为0.93V 0001 = CC驱动电压为1.024V 0010 = CC驱动电压为1.12V 0011 = CC驱动电压为1.22V 0100 = CC驱动电压为1.315V 0101 = CC驱动电压为1.412V 0110 = CC驱动电压为1.51V 0111 = CC驱动电压为1.61V 1000 = CC驱动电压为1.705V 其他 = CC驱动电压为1.8V						

TPC1BDR - PD发送波特率寄存器

TPC1BDR - PD数据发送波特率寄存器								
地址: 0xDE						默认值: 0x7B		
Bit	7	6	5	4	3	2	1	0
Name	CCIDLC			CCBDR[4:0]				
R/W	R/W							
Bit	Name	描述						
7:5	CCIDLC	CC总线空闲周期设置,以PD数据发送的波特率为时钟的计时设置 001 = 4个CC总线周期 010 = 8个CC总线周期 011 = 12个CC总线周期(默认) ... 111 = 28个CC总线周期						
4:0	CCBDR	PD数据发送波特率寄存器。 TPCBDR用于调整PD发送数据的波特率。PD数据发送波特率发生器工作与内部24MHz RC的4分频,也就是6MHz,默认波特率为: $6000\text{KHz}/27 = 222\text{KHz}$ ,建议设置为20以产生300KHz的波特率						

KCMSK1 -KCode屏蔽寄存器1

KCMSK1 - KCODE屏蔽寄存器1								
地址: 0xDD						默认值: X111_1111		
Bit	7	6	5	4	3	2	1	0
Name	FFR	CRE	HRE	SOP2DE	SOP1DE	SOP2E	SOP1E	SOPE
R/W	W/O	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	FFR	写1将复位收发FIFO状态						



6	CRE	写1 使能线复位请求数据帧
5	HRE	写1 使能硬复位请求数据帧
4	SOP2DE	写1 使能SOP' ' _Debug 数据帧
3	SOP1DE	写1 使能SOP' _Debug 数据帧
2	SOP2E	写1 使能SOP' ' 数据帧
1	SOP1E	写1 使能SOP' 数据帧
0	SOPE	写1 使能SOP 数据帧

TPC1SIG -PD 设备模式标识寄存器

TPC1SIG - PD 设备模式标识寄存器

地址: 0xD2						默认值: 0XXX_1101		
Bit	7	6	5	4	3	2	1	0
Name	CCDLS	-	-	-	PPS	SR1	SR0	PDS
R/W	R/W	-	-	-	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	CCDLS	CC/IDLE 状态源选择位 0 = IDLE 状态来自CC 数据比较器 1 = IDLE 状态来自CV 电平比较器						
6:4	-	保留未用						
3	PPR	端口电源角色 (Port Power Role). 1 = Sink, 0 = Source (仅针对SOP 数据包) 1 = 信息来自DFP/UFP, 0 = 消息来自线缆插头 (仅针对Gable Plug)						
2:1	SR	兼容标准版本, 默认10 = Revision 3.0						
0	PDR	端口数据角色 (Power Data Role), 仅对SOP 数据包有效 1 = DFP, 0 = UFP						

TP1CON0 - TPC1 控制寄存器 0

TP1CON0 - TPC1 控制寄存器0

地址: 0xDA						默认值: 0000_0111		
Bit	7	6	5	4	3	2	1	0
Name	TPCEN	TXEN	TXST	RXEN	AIDEN	GIDEN	RXM	AGCEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	TPCEN	uTPC 模块使能控制。 写1 使能TPC 模块, 清0 将关闭uTPC 模块的工作时钟						
6	TXEN	写1 使能uTPC 发送功能模块						
5	TXST	写1 将启动一次数据发送						
4	RXEN	写1 使能uTPC 数据接收模块						
3	AIDEN	写1 使能自动消息ID 硬件产生。 使能后, 有uTPC 的发送器自动产生和更新Message ID						
2	GIDEN	写1 使能对接收GoodCRC 数据包的ID 确认。 当接收到的GoodCRC 的ID 与发送的数据ID 相同时, GoodCRC 为有效。						



1	RXM	接收数据是否包含CRC 部分。 写1, 将CRC 数据一同写入接收FIFO 清0, CRC数据将不会被写入FIFO
0	AGCEN	写1 使能自动GoodCRC 响应。 当接收到正确的CRC 数据后, 硬件自动回应GoodCRC 响应数据包

TP1CON1 - TPC1 控制寄存器 1

TP1CON1 - TPC1 控制寄存器1

地址: 0xDB						默认值: 0000_0111		
Bit	7	6	5	4	3	2	1	0
Name	TXREN	TXKC2	TXKC1	TXKC0	RSYNC	RXKC2	RXKC1	RXKC0
R/W	R/W	R/W	R/W	R/W	R/W	R/O	R/O	R/O
Bit	Name	描述						
7	TXREN	发送出错重试使能。发送出错后, 将会重试发送2 次						
6:4	TXKC[2:0]	PD 数据发送KCODE 设置 000 = SOP 001 = SOP' 010 = SOP' ' 011 = SOP' _DEBUG 100 = SOP' ' _DEBUG 101 = 硬件复位请求 110 = 线缆复位请求						
3	RSYNC	写 1 使能接收同步模式。uTPC 将在检测到 CC 空闲状态时进入接收状态						
2:0	RXKC[2:0]	PD 当前接收到数据包KCODE 类型 000 = SOP 001 = SOP' 010 = SOP' ' 011 = SOP' _DEBUG 100 = SOP' ' _DEBUG 101 = 硬件复位请求 110 = 线缆复位请求						

TP1CON2 - TPC 控制寄存器 2

TP1CON2 - TPC 控制寄存器2

地址: 0xDC						默认值: 0100_0000		
Bit	7	6	5	4	3	2	1	0
Name	CCRN	GCVEN	RXFM	CCIDL	TXFF	RXFE	TXDN	RXDN
R/W	R/W	R/W	R/W	R/O	R/O	R/O	R/W	R/W
Bit	Name	描述						
7	CCRN	写1 将执行CC 软件复位 将会复位 uTPC 物理层控制器和链路控制器的所有状态						
6	GCVEN	写1 使能GoodCRC 对链路层可见。 清 0 后, GoodCRC 仅在物理层控制器内完成, 链路层控制器不可见						



5	RXFM	接收FIFO 中断模式。 0 = 半满中断, 1 = 非空中断
4	CCIDL	CC 空闲状态标志位
3	TXFF	发送FIFO 满标记位
2	RXFE	接收FIFO 空标记位
1	TXDN	发送结束标记位。启动发送时自动清零, 软件可以通过写0 清零
0	RXDN	接收结束标记位。启动接收时自动清零, 软件可以通过写0 清零

TPC1TXFF - uTPC1 发送 FIFO 访问寄存器

TPC1TXFF - uTPC1 数据发送FIFO 访问寄存器

地址: 0xD8						默认值: XX			
Bit	7	6	5	4	3	2	1	0	
Name	TPCTXFF								
R/W	W/O								
Bit	Name	描述							
7:0	TPCTXFF	访问uTPC 内部数据发送FIFO 的接口。 软件通过写TPCTXFF 寄存器, 实现向FIFO 中写数据。通过查询TPCON2 寄存器 中的TXFF 为判断发送FIFO 是否已经写满。							

TPC1RXFF - uTPC1 接收 FIFO 访问寄存器

TPC1RXFF - uTPC 数据接收FIFO 访问寄存器

地址: 0xD9						默认值: XX			
Bit	7	6	5	4	3	2	1	0	
Name	TPCRXFF								
R/W	R/O								
Bit	Name	描述							
7:0	TPCRXFF	访问uTPC 内部数据接收FIFO 的接口。 软件通过读TPCRXFF 寄存器, 从接收FIFO 中读出接收数据。通过查询TPCON2 寄存器中的RXFE 为判断发送FIFO 是否为空。							

CC1ACR0 -Type-C1 CC 控制寄存器 0

CC1ACR0 - Type-C1 CC 控制寄存器0

地址: 0xD3						默认值: 00001101			
Bit	7	6	5	4	3	2	1	0	
Name	-	-	-	-	RXTR1	RXTR0	-	-	
R/W	-	-	-	-	R/W	R/W	-	-	
Bit	Name	描述							
7	-	保留未用							
6	-	保留未用							
5	-	保留未用							
4	-	保留未用							



3:2	RXTR	输入延时校准位 00: 采用15 根电阻串延时 01: 采用6 根电阻串延时 10: 采用4 根电阻串延时 11: 采用2 根电阻串延时
1:0	-	保留未用

CC1ACR1 -Type-C1 CC 控制寄存器 1

CC1ACR1 - Type-C1 CC 控制寄存器1

地址: 0xD4						默认值: 00000000		
Bit	7	6	5	4	3	2	1	0
Name	IP1S1	IP1S0	-	C1CH4	C1CH3	C1CH2	C1CH1	C1CH0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:6	IP1S	CC1 上拉电流源选择控制位 00 = 80uA 01 = 180uA 1x = 330uA						
5	-	保留未用						
4	C1CH4	CC1 接内部上拉电阻, 用于低功耗休眠的电平变化唤醒						
3	C1CH3	CC1 接内部Ip 上拉电流源						
2	C1CH2	CC1 接内部CC 数据检测比较器						
1	C1CH1	CC1 接VCONN, 接内部LDO 输出VDD5						
0	C1CH0	CC1 接5.1K 下拉电阻						

CC1ACR2 -Type-C1 CC 控制寄存器 2

CC1ACR2 - Type-C1 CC 控制寄存器2

地址: 0xD5						默认值: 00000000		
Bit	7	6	5	4	3	2	1	0
Name	IP2S1	IP2S0	-	C2CH4	C2CH3	C2CH2	C2CH1	C2CH0
R/W	R/W	R/W	-	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:6	IP2S	CC2 上拉电流源选择控制位 00 = 80uA 01 = 180uA 1x = 330uA						
5	-	保留未用						
4	C2CH4	CC2 接内部上拉电阻, 用于低功耗休眠的电平变化唤醒						
3	C2CH3	CC2 接内部Ip 上拉电流源						
2	C2CH2	CC2 接内部CC 数据检测比较器						
1	C2CH1	CC2 接VCONN, 接内部LDO 输出VDD5						
0	C2CH0	CC2 接5.1K 下拉电阻						

**CC1ACR3 -Type-C1 CC 控制寄存器 3**
**CC1ACR3 - Type-C1 CC 控制寄存器3**

地址: 0xD6						默认值: 00000000		
Bit	7	6	5	4	3	2	1	0
Name	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
0								
0								

**CC1ACR4 -Type-C1 CC 控制寄存器 4**
**CC1ACR4 - Type-C1 CC 控制寄存器4**

地址: 0xD7						默认值: 00000000		
Bit	7	6	5	4	3	2	1	0
Name	ECPL	CVTE	CSS1	CSS0	CCVS3	CCVS2	CCVS1	CCVS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	ECPL	CC 数据极性控制						
6	CVTE	CVAC 校准使能控制						
5:4	CSS	CC 接收信号源选择 00/11 = CC 接收来自PDC 数据比较器 (CCAC) 01 = 接GND 低电平 (测试用) 10 = CC 接收来自通用比较器GPCM1 的输出						
3:0	CCVS	PD 通讯发送数据的CC 驱动电平控制 0000 = CC 驱动电压为0.93V 0001 = CC 驱动电压为1.024V 0010 = CC 驱动电压为1.12V 0011 = CC 驱动电压为1.22V 0100 = CC 驱动电压为1.315V 0101 = CC 驱动电压为1.412V 0110 = CC 驱动电压为1.51V 0111 = CC 驱动电压为1.61V 1000 = CC 驱动电压为1.705V 其他 = CC 驱动电压为1.8V						

**TPC1IER -TPC1 外设中断使能寄存器**
**TPC1IE - TPC1 外设中断使能寄存器**

地址: 0xD1						默认值: 0000_0000		
Bit	7	6	5	4	3	2	1	0
Name	-	TP1IE	TX1IE	RX1IE	-	TPIE	TXIE	RXIE
R/W	-	W/R	W/R	W/R	-	W/R	W/R	W/R
Bit	Name	描述						
7	-	保留未用						
6	-	保留未用						
5	-	保留未用						

4	-	保留未用
3	-	保留未用
2	TPIE	uTPC1 传输完成中断使能控制位
1	TXIE	uTPC1 发送FIFO 中断使能控制位
0	RXIE	uTPC1 接收FIFO 中断使能控制位

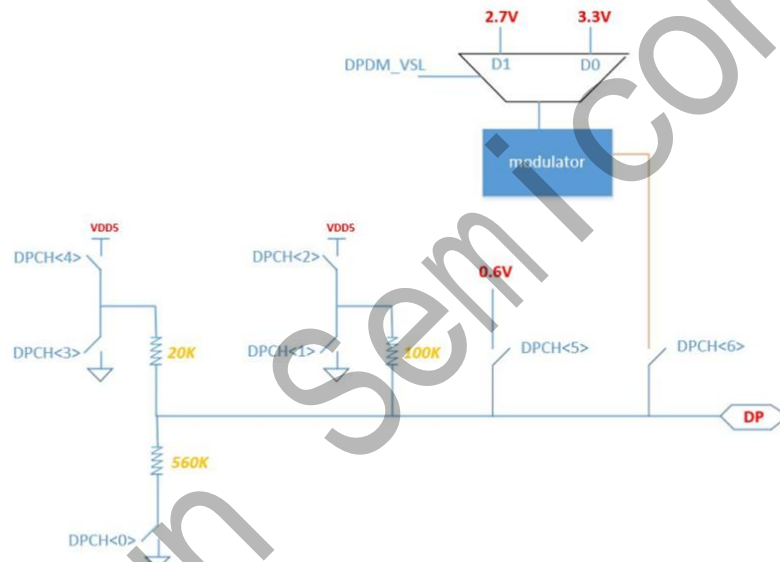
## 19. USB A 口充电模拟前端

- 两组独立 USB A 口支持

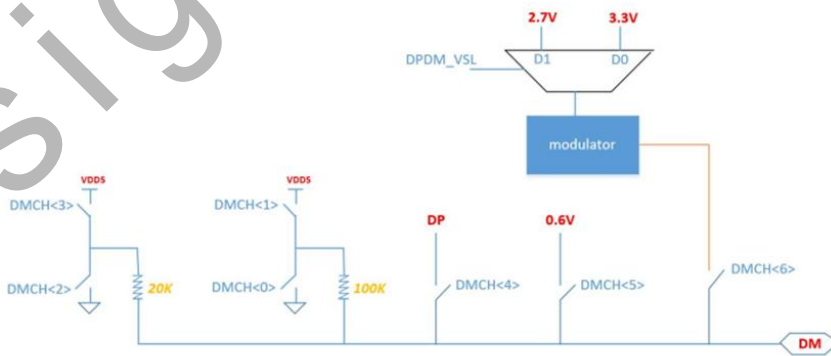
### 19.1. 综述

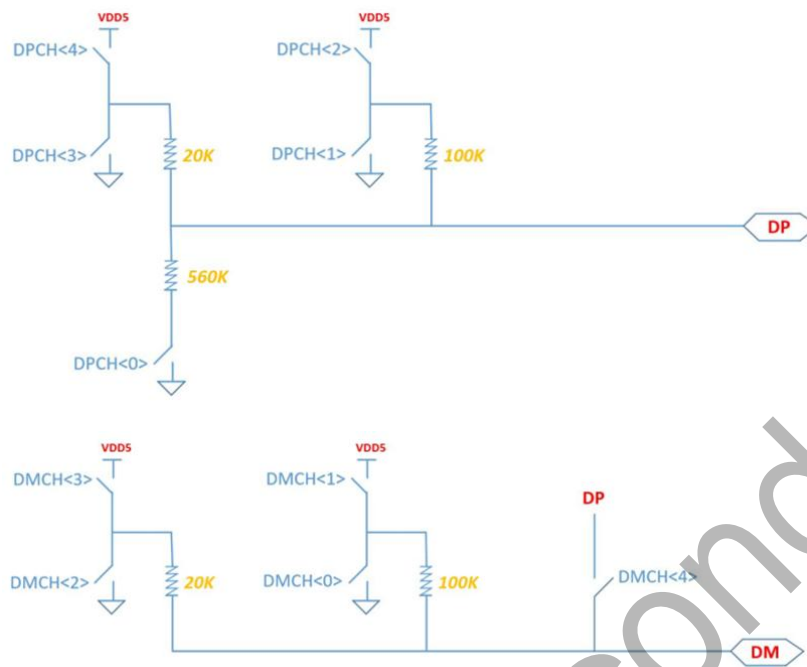
USB A 口充电模拟前端电路如下图，PDS8F208A 实现两组独立的 USB A 口充电模拟前端，可以通过开关控制，在 USB-A 口的 DP/DM 上开启上/下拉电阻。通过两个通道配合，可以实现电阻分压电路。除可编程分压电阻外，也可以使用内部的 0.6V/2.7V/3.3V 电压源输出驱动到 DP/DM 上，可以实现更精确和更高分辨率的模拟电压输出。

TYPEAB0 电路结构图



TYPEAB1 电路结构图





## 19.2. 寄存器定义

TPACR0 - USB A 口 DPO 通道控制寄存器

TPACR0 - USB A 口 DPO 通道控制寄存器								
地址: 0x6E				0000_0000				
Bit	7	6	5	4	3	2	1	0
	-	DPOCH6	DPOCH5	DPOCH4	DPOCH3	DPOCH2	DPOCH1	DPOCH0
R/W	-	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	-	保留未用						
6	DPCH06	选择 DPCH6						
5	DPCH05	选择 0.6V						
4	DPCH04	20K 选择到 VDD5						
3	DPCH03	20K 选择到地						
2	DPCH02	100K 选择到 VDD5						
1	DPCH01	100K 选择到地						
0	DPCH00	560K 选择到地						

TPACR1 - USB A 口 DMO 通道控制寄存器

TPACR1 - USB A 口 DMO 通道控制寄存器								
地址: 0x6F				0000_0000				
Bit	7	6	5	4	3	2	1	0
	-	DMCH06	DMCH00	5DMCH04	DMCH03	DMCH02	DMCH01	DMCH00
R/W	-	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	-	保留未用						
6	DMCH06	选择 DPCH6						



5	DMCH05	选择 0.6V
4	DMCH04	短接到 DP
3	DMCH03	20K 选择到 VDD5
2	DMCH02	20K 选择到地
1	DMCH01	100K 选择到 VDD5
0	DMCH00	100K 选择到地

TPACR2 - USB A 口 DP1 通道控制寄存器

TPACR2 - USB A 口 DP1 通道控制寄存器								
地址: 0x70					0000_0000			
Bit	7	6	5	4	3	2	1	0
	-	-	-	DP1CH4	DP1CH3	DP1CH2	DP1CH1	DP1CH0
R/W	-	-	-	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:5	-	保留未用						
4	DP1CH4	20K 选择到 VDD5						
3	DP1CH3	20K 选择到地						
2	DP1CH2	100K 选择到 VDD5						
1	DP1CH1	100K 选择到地						
0	DP1CH0	560K 选择到地						

DPDMTR - USB A 口延时控制寄存器

DPDMTR - USB A 口延时控制寄存器								
地址: 0xC7					0000_0000			
Bit	7	6	5	4	3	2	1	0
	DM1TR		DP1TR		DM0TR		DP0TR	
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:6	DM1TR	DM1 延时: 0=15 根电阻串, 1=6 根电阻串, 2=4 根电阻串, 3=2 根电阻串,						
5:4	DP1TR	DP1 延时: 0=15 根电阻串, 1=6 根电阻串, 2=4 根电阻串, 3=2 根电阻串,						
3:2	DM0TR	DM0 延时: 0=15 根电阻串, 1=6 根电阻串, 2=4 根电阻串, 3=2 根电阻串,						
1:0	DP0TR	DP0 延时: 0=15 根电阻串, 1=6 根电阻串, 2=4 根电阻串, 3=2 根电阻串,						

TPACR3 USB A 口 DM1 通道控制寄存器

TPACR1 - USB A 口 DM1 通道控制寄存器								
地址: 0x71					0000_0000			
Bit	7	6	5	4	3	2	1	0
	-	-	-	DM1CH4	DM1CH3	DM1CH2	DM1CH1	DM1CH0
R/W	-	-	-	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:5	-	保留未用						
4	DM1CH4	短接到 DP						
3	DM1CH3	20K 选择到 VDD5						
2	DM1CH2	20K 选择到地						
1	DM1CH1	100K 选择到 VDD5						

0	DM1CH0	100K 选择到地
---	--------	-----------

TPASR0 - USB A 0 口 寄存器

TPASR0 - USB A0 口寄存器								
地址: 0x48					0000_0000			
Bit	7	6	5	4	3	2	1	0
	-	DO_VSEL	DO_TXEN	DO_RXEN	DMO_TXD	DPO_TXD	DMO_RXD	DPO_RXD
R/W	-	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	-	保留未用						
6	DO_VSEL	TX 模块驱动电压选择: 0=3.3V, 1=2.7V						
5	DO_TXEN	1=TX 模块使能, 0=TX 模块关闭						
4	DO_RXEN	1=RX 模块使能, 0=RX 模块关闭						
3	DMO_TXD	发送到 DMO 的数据						
2	DPO_TXD	发送到 DPO 的数据						
1	DMO_RXD	DMO 收到的数据						
0	DPO_RXD	DPO 收到的数据						

TPASR1 - USB A 1 口 寄存器

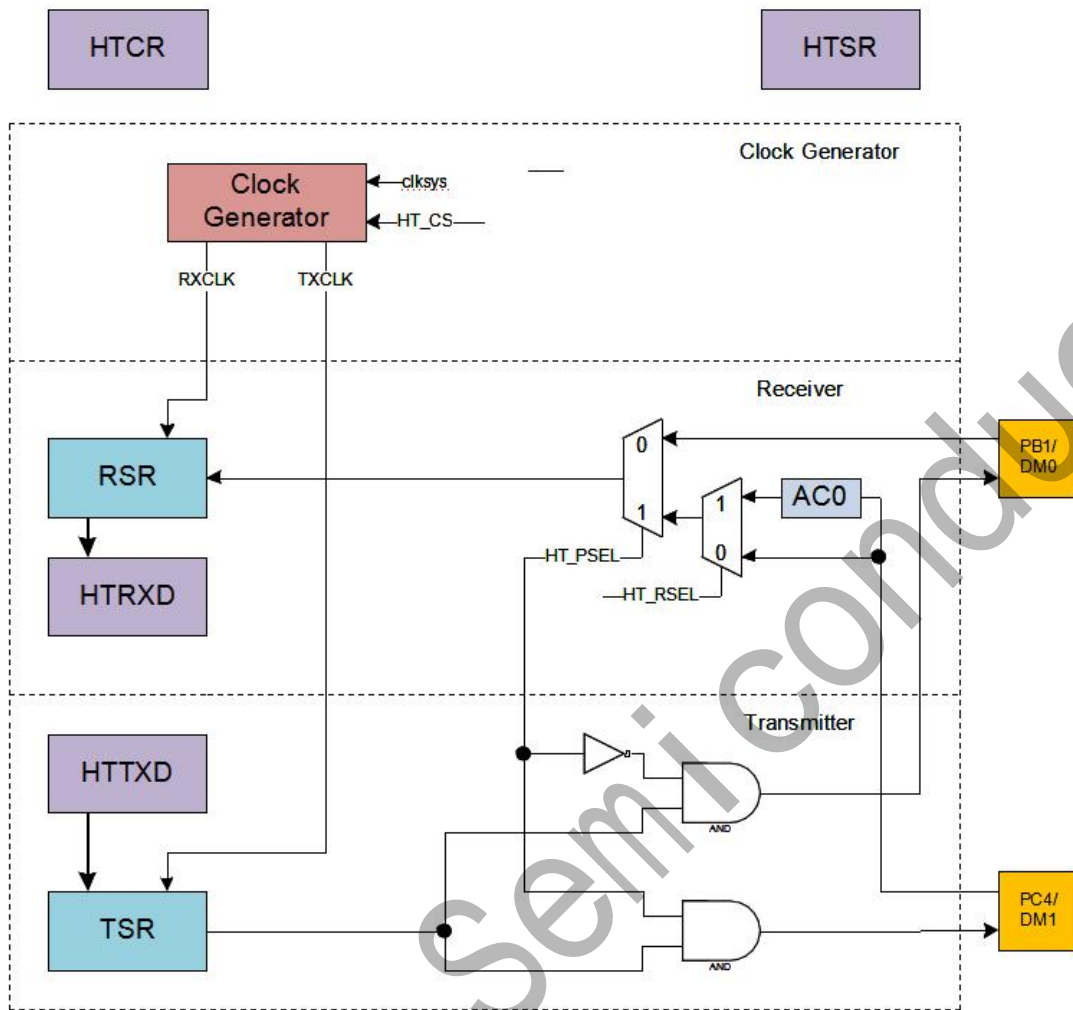
TPASR1 - USB A01 口寄存器								
地址: 0x49					0000_0000			
Bit	7	6	5	4	3	2	1	0
	-	D0_VSEL	D0_TXEN	D0_RXEN	DMO_TXD	DPO_TXD	DMO_RXD	DPO_RXD
R/W	-	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	-	保留未用						
6	D1_VSEL	TX 模块驱动电压选择: 0=3.3V, 1=2.7V						
5	D1_TXEN	1=TX 模块使能, 0=TX 模块关闭						
4	D1_RXEN	1=RX 模块使能, 0=RX 模块关闭						
3	DM1_TXD	发送到 DM1 的数据						
2	DP1_TXD	发送到 DP1 的数据						
1	DM1_RXD	DM1 收到的数据						
0	DP1_RXD	DP1 收到的数据						

## 20. 华为 HT 快充通信接口

- 支持 HT 传输
- 发送和接收均有一个 FIFO
- 发送和接收均产生 HT 传输中断

### 20.1. 综述

HT 传输控制器是一个支持华为快充协议通信传输物理接口的控制器, 它分为时钟产生器, 发送器和接收器三部分, 结构框架图如下:



HT 传输控制器结构框图

### 时钟发生器

时钟发生器首先产生一个基准时钟，它的周期为 10us。然后再用基准时钟来产生发送和接收时钟，用来发送和接收数据。

基准时钟的周期固定为 10us，要根据系统时钟的频率来设置位于 HTCR 寄存器的 HT\_CS 位来选择合适的分频系数。需要注意的是，系统时钟的频率可能会有误差，因此基准时钟的频率也会有误差。发送时钟分为分隔符时钟，数据位时钟和 PIN 高电平时钟，周期分别为 40us，160us 和 2.56ms。接收时钟与基准时钟频率相同，周期为 10us。

## 20.2. 发送器

发送器负责把要发送的数据或 PIN 头发送出去，它包含一个 9 位的数据寄存器和一个 9 位的移位寄存器，以及发送的控制状态机。

软件需要设置位于 HTCR 寄存器的 HT\_TXEN 位来使能发送器，还需要设置 HT\_CS 位来选择合适的分频系数，设置 HT\_PSEL 位来选择传输要使用的引脚，设置 HT\_PM 位来选择校验模式。

当只需要发送 PIN 头时，软件设置位于 HTCR 寄存器的 HT\_TXPIN 位为 1，硬件会在所选择的引脚上发送 2.56ms 的高电平和 160us 的低电平。

当需要发送一个或多个数据时，软件先检测位于 HTSR 寄存器的 HT\_TXEMPT 位的状态，当 HT\_TXEMPT 位为 1 时，软件可往 HTTXD 寄存器写入要发送的数据。硬件会把 HTTXD 中的数据以及相应的校验位加载到移位寄存器中，同时清零 HT\_TXEMPT 位，并按照多个分隔符+字节数据+校验位的格式开始发送到所选择的引脚上。

若需要发送多个数据，软件可再次检测 HT\_TXEMPT 的状态，当 HT\_TXEMPT 位为 1 时，软件可继续往 HTTXD 寄存器写入要发送的数据。当硬件把分隔符和移位寄存器中的数据位都发送完毕时，会检测 HT\_TXEMPT 位的状态，当 HT\_TXEMPT 位为 0 时，表明软件再此写入了新的数据，硬件会加载新的数据以及相应的校验位到移位寄存器中，启动新的发送；当 HT\_TXEMPT 位为 1 时，表明软件没有写入新的数据，即当前数据传输完成，硬件会发送多个分隔符+PIN 头作为传输的结尾。

当 PIN 头或所有的数据+PIN 头都发送完毕后，硬件会置位位于 HTSR 寄存器的 HT\_TXC 位，同时置位位于 HTIFR 寄存器的 HTIF 位。

### 20.3. 接收器

接收器负责接收数据或 PIN 头，它包含一个 9 位的数据寄存器和一个 9 位的移位寄存器，以及接收的控制状态机。

软件需要设置位于 HTCR 寄存器的 HT\_RXEN 位来使能接收器，还需要设置 HT\_CS 位来选择合适的分频系数，设置 HT\_PSEL 位来选择传输要使用的引脚，设置 HT\_RSEL 位来选择是否经过比较器解码（当选择 DM1 引脚作为输入信号源时），设置 HT\_PM 位来选择校验模式。

当接收器检测到多个分隔符后，会采用多次采样后滤波的方式接收数据，按次序保存到移位寄存器当中，当 8 位数据和 1 位校验位都接收完后，硬件把移位寄存器中的数据保存到 HTRXD 寄存器和 HT\_RXPAR 位中，同时置位位于 HTSR 寄存器的 HT\_RXBYT 位和位于 HTIFR 寄存器的 HTIF 位。软件检测到 HT\_RXBYT 位为 1 后，须尽快读取 HTRXD 和 HT\_RXPAR 获取所接收到的数据和校验位，避免被新的接收所覆盖。当 HT\_RXBYT 位为 1 后再次接收到数据和校验位，

硬件会置位 HT\_RXOER 位，表明发生接收超限事件。软件也可以读取 HT\_RXPER 位来获取校验位是否有错误。

当接收器检测到 PIN 头后，会置位位于 HTSR 寄存器的 HT\_RXPIN 位和位于 HTIFR 寄存器的 HTIF 位。

### 20.4. 中断控制

当发生以下三种情况之一时，硬件会置位位于 HTIFR 寄存器的 HTIF 位：

- 发送完 PIN 头或所有的数据+PIN 头
- 接收到 8 位数据和 1 位校验位
- 接收到 PIN 头

若此时 HT 传输中断使能位 HTIE 为 1 且全局中断使能，则会产生 HT 传输中断。执行此中断服务程序时硬件会清零 HTIF 位。软件也可对 HTIF 位写 1 来清零该位。

### 20.5. 寄存器定义

HTCR - HT 传输控制寄存器

HTCR - HT 传输控制寄存器								
地址：0x8C					X0000_0000			
Bit	7	6	5	4	3	2	1	0
	HT_RXEN	HT_TXEN	HT_PM	HT_TXPIN	HT_RSEL	HT_PSEL	HT_CS1	HT_CS0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	HT_RXEN	HT 传输接收使能控制位 1: 使能HT传输接收 0: 禁止HT传输接收						
6	HT_TXEN	HT 传输发送使能控制位 1: 使能HT传输发送 0: 禁止HT传输发送						
5	HT_PM	HT 传输校验模式选择控制位 1: 偶校验 0: 奇校验						

4	HT_TXPIN	HT 传输发送PIN使能控制位 1: 使能发送PIN 头 0: 禁止发送PIN 头 软件置位HT_TXPIN 位后, 硬件开始发送PIN 头, 并清零HT_TXPIN 位。此位用作单独发送 PIN 头。多字节数据发送完毕后, 硬件会自动发送 PIN 头作为结尾, 无须置位此位。
3	HT_RSEL	HT 传输接收源选择位 当HT_PSEL 位为0时, 选择DM0引脚作为接收输入信号 当HT_PSEL 位为1 时, 再根据HT_RSEL 位的值如下选择: 1: 选择比较器0 的输出作为接收输入信号 0: 选择DM1 引脚作为接收输入信号
2	HT_PSEL	HT 传输引脚选择位; 1: 选择DM1 引脚进行HT 传输 0: 选择DM0 引脚进行HT 传输
1:0	HT_CS	HT 传输时钟分频系数选择位; 11: 240 10: 120 01: 60 00: 30

HTSR - HT 传输状态寄存器

HTSR - HT 传输状态寄存器

地址: 0x8D		X0000_0101						
Bit	HT_RXPIN	HT_RXBYT	HT_RXPAR	HT_RXPER	HT_RXOER	HT_TXC	HT_TXBUSY	HT_TXEMPT
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7	HT_RXPIN	HT 传输接收PIN 头标志位 当HT_RXPIN 位为1 时, 表示已接收到PIN 头, 软件可对该位写1 来清 零该位。清零 HT_RXEN 位也会清零该位。						
6	HT_RXBYT	HT 传输接收字节标志位 当 HT_RXBYT 位为 1 时, 表示已接收到一个字节数据和校验位, 软件读取HTRXD 寄存器或对该位写1 都可以清零该位。清零HT_RXEN 位也会 清零该位。						
5	HT_RXPAR	HT 传输接收校验位 HT_RXPAR 表示所接收到的校验位的值						
4	HT_RXPER	HT 传输接收校验位有误标志位 当HT_RXPER 位为1 时, 表示已接收到的校验位有错误。						
3	HT_RXOER	HT 传输接收超限标志位 当HT_RXOER 位为1 时, 表示HT_RXBYT 位为1 且又接收到了一个新的字 节和校验位数据。旧的字节和校验位数据会被覆盖掉。软件可对该位写1 来清零该位。清零HT_RXEN 位也会 清零该位。						

2	HT_TXC	HT 传输发送完毕标志位 当HT_TXC位为1时，表示所有的数据和PIN头都发送完毕。开启新的发送会清零该位。清零HT_TXEN 位会置位该位。
1	HT_TXBUSY	HT 传输发送忙碌标志位 当HT_TXBUSY 位为1时，表示发送数据移位寄存器中的数据正在发送中。当移位寄存器中的数据和分隔符发送完毕且HT_TXEMPT 位为1时，硬件会清零 HT_TXBUSY 位。清零HT_TXEN 位也会清零该位。
0	HT_TXEMPT	HT 传输发送标志位 当HT_TXEMPT 位为1时，表示发送数据寄存器的状态为空，软件可往 HT_TXD 中写入要发送的数据。当发送数据移位寄存器中的数据发送完毕时，硬件会把 HT_TXD 中的数据加载到发送数据移位寄存器中进行发送，同时会置位 HT_TXEMPT 位。清零 HT_TXEN 位也会置位该位。

**HTRXD - HT 传输接收字节数据寄存器**

HTRXD - HT 传输接收字节数据寄存器								
地址: 0x8E					X1111_1111			
Bit	7	6	5	4	3	2	1	0
	HTRXD7	HTRXD6	HTRXD5	HTRXD4	HTRXD3	HTRXD2	HTRXD1	HTRXD0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	HTRXD	HT 传输接收字节数据寄存器。 当位于HTSR 寄存器的HT_RXBYT 位为1时，软件须尽快读取HTRXD 和 HT_RXPAR 来获得所接收到的字节和校验位数据，避免被新的数据覆盖。						

**HTTXD - HT 传输发送字节数据寄存器**

HTTXD - HT 传输发送字节数据寄存器								
地址: 0x8F					X0000_0000			
Bit	7	6	5	4	3	2	1	0
	HTTXD7	HTTXD6	HTTXD5	HTTXD4	HTTXD3	HTTXD2	HTTXD1	HTTXD0
R/W	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Bit	Name	描述						
7:0	HTTXD	HT 传输发送字节数据寄存器。 当HT_TXEN 位为1且HT_TXTMPT 位为1时，软件可以往HTTXD 写入要发送的数据。						

**HTIFR - HTIFR 中断寄存器**

HTIFR - HTIFR 中断寄存器								
地址: 0x39					X0000_0000			
Bit	7	6	5	4	3	2	1	0
	-	-	-	HTIE	-	-	-	HTIF
R/W	-	-	-	W/R	-	-	-	W/R
Bit	Name	描述						
7:5	-	保留						

4	HTIE	HT 传输中断使能控制位； 1: HT 传输中断被使能 0: HT 传输中断被禁止
3:1	-	保留
0	HTIF	HT 传输中断标志位； 当接收到一个字节数据或接收到 PIN 头或者发送完所有数据和 PIN 头 时，硬件会置位HT传输中断标志HTIF位。若此时HT传输中断使能位 HTIE为1且全局中断使能，则会产生HT传输中断。执行此中断服务程 序时硬件会清零HTIF位。软件也可对HTIF位写1来清零该位。

## 21. 模拟比较器 0 (AC0)

- 10mV 的比较精度
- 出厂失调校准
- 支持 3 路片外模拟输入
- 支持 ADC 的多路复用输入 (ADMUX)
- 支持 CCCV\_VFB 电压反馈输入
- 支持内部 8 位 DAC 输入 (DAO)
- 可编程输出数字滤波控制
- 比较输出可触发中断请求

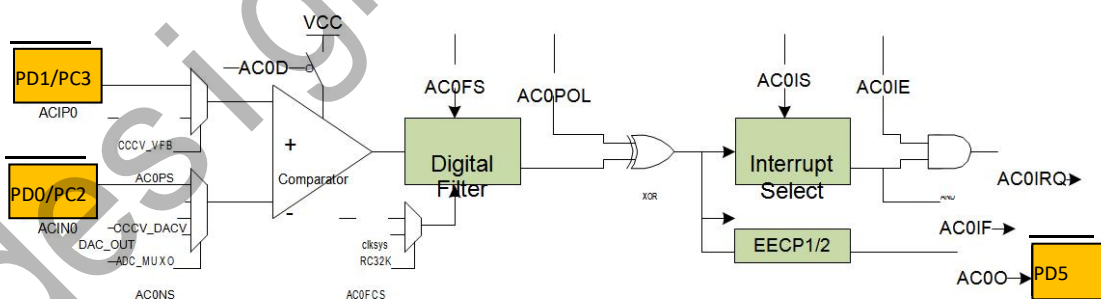
### 21.1. 综述

模拟比较器对正端的值与负端的值进行比较，当正端上的电压比负端上的电压高时，模拟比较器的输出 AC00 被置位。当 AC00 的电平发生变化时，信号的边沿可用来触发中断。

模拟比较器支持多路输入，正端和负端输入源可以选择来自外部端口或者来自多种内部产生的参考源。模拟比较器本身支持失调校准，可以保证比较器工作的一致性。比较器输出端 集成了一个硬件可编程数字滤波器，可以根据应用需求，选择合适的滤波设置，获得更加稳定的比较输出。

比较器的输出状态可以通过寄存器获取，也可以产生中断请求，实现更高效的实时事件俘 获功能。比较器的输出还可以直接输出到外部引脚。比较器的输出还会连接到 EECF 模块，用以关闭该模块所产生 PWM 信号输出，实现对外部驱动电路的闭环控制。

模拟比较器的结构图：



模拟比较器 0 功能示意图

### 21.2. 模拟比较器的输入

模拟比较器的两个输入端都有不同的输入源。

正端的输入有外部引脚 AC1P0 和 AC1P1，输入源的选择由位于 AC0XR 中的 AC0PS 位来控制。

负端的输入有外部引脚 AC1N0 和 AC1N1，内部 DAC 模块的输 DAC\_OUT 出，以及内部 ADC 多路器的 ADC\_MUX0 输出。输入源的选择由位于 ACOSR 的 ACONS1 位和 ACONSO 来控制。

### 21.3. 比较器输出与 PWM 控制

模拟比较器的输出端集成了一个可编程的数字滤波器，可以滤除瞬时干扰对比较器输出产生的影响。数字滤波器可以根据应用需求，选择合适的滤波时间宽度，只有当比较器的输出稳定持续满足滤波时间的限制，滤波电路才更新比较器的输出，从而达到一个更加稳定的输出结果。

数字滤波器的时钟通过位于 ACOXR 寄存器的 ACOFCS 位来选择，可以选择系统时钟或内部低频时钟 RC32K。滤波的时间长度由位于 ACOXR 寄存器的 ACOFS 位来选择，可以选择 1 个、2 个或三个滤波器时钟周期长度，或者关闭数字滤波功能。

模拟比较器的输出信号经过数字滤波后，还可以进行极性的变化。当设置位于 ACOXR 寄存器的 ACOPOL 位为 1 时，输出信号会被反向。当设置位于 ACOXR 寄存器的 ACOPOL 位为 0 时，输出信号不会发生改变。

经过极性控制后，模拟比较器的输出可以产生中断请求，当比较器的输出发生变化时，即输出信号产生上升沿或下降沿时，会置位于 ACOXR 寄存器的 ACOIF 位。置位的触发模式由位于 ACOXR 寄存器的 ACOIS 位来选择。当 ACOIF 位被置位，中断使能位 ACOIE 为“1”且全局中断置位时，中断产生。执行模拟比较器中断服务程序时，ACOIF 将自动清零，或对 ACOIF 位写“1”也可清零该位。

经过极性控制后，模拟比较器的输出还可以直接输出到外部引脚上，此功能是通过置位于 ACOXR 寄存器的 ACOOE 位来实现的。

经过极性控制后，模拟比较器的输出还可以输出到 EEC1 模块用作关闭 PWM 信号输出，详见 EEC1 和 EEC2 模块描述。

### 21.4. 寄存器定义

ACOSR - ACO 控制和状态寄存器								
ACOSR - ACO 控制和状态寄存器								
地址: 0x50					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	ACOD	ACONS1	ACOO	ACOIF	ACOIE	ACONS0	ACOIS1	ACOIS0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	ACOD	AC0 禁止位。 当设置ACOD 位为“1”时，模拟比较器被关闭。当设置ACOD 位为“0”时，模拟比较器被开启。						
6	ACONS1	AC0 负端选择选择控制高位。 ACONS1 和ACONS0 组成ACONS 位用来选择AC0 的负端输入 00: 选择 PD0 作为负端输入 01: 选择 PC2 作为负端输入 10: 选择内部 DA10OUTBUF 作为负端输入 11: 选择 TPC0 的内部 GCX 作为负端输入						
5	ACOO	AC0 输出状态位。 模拟比较器的输出经过同步之后直接连到 ACOO 位。软件可读取 ACOO 位的值来获取模拟比较器的输出值。						
4	ACOIF	AC0 中断标志位。 当模拟比较器的输出事件触发了由 ACOIS 位定义的中断模式时，ACOIF 位被置位。当中断使能位ACOIE 为“1”且全局中断置位时，中断产生。执行模拟比较器中断服务程序时，ACOIF 将自动清零，或对 ACOIF 位写“1”也可清零该位。						
3	ACOIE	AC0 中断使能位。 当设置ACOIE 位为“1”，且全局中断置位时，模拟比较器的中断被使能。当设置ACOIE 位为“0”时，模拟比较器的中断被禁止。						

2	ACONS0	ACO 负端选择控制低位。 ACONS1 和ACONS0 组成ACONS 位用来选择ACO 的负端输入
1:0	ACOIS	ACO 中断模式选择控制位。 ACOIS 用来选择模拟比较器的中断触发方式。 00: ACO0 的上升沿或下降沿触发 01: 保留 10: ACO0 的下降沿触发 11: ACO0 的上升沿触发

**ACOXR - ACO 辅助控制器寄存器**
**ACOXR - ACO 辅助控制寄存器**

地址: 0x51				默认值: 0x00				
Bit	7	6	5	4	3	2	1	0
	ACOTREN	ACOOE	ACOPS1	ACOPSO	ACOPOL	ACOFCS	ACOFs1	ACOFs0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	ACOTREN	ACO 失调校准使能位。 0: 禁止ACO 失调校准 1: 使能ACO 失调校准						
6	ACOOE	ACO 输出到引脚使能位。 0: 禁止ACO0 输出到外部引脚PD5 1: 使能ACO0 输出到外部引脚PD5						
5:4	ACOPS	ACO 正端选择位。 0: 选择 PD1 作为正端输入 1: 选择 PC3 作为正端输入 2: 选择 DA10UTBUF 作为正端输入 3: 选择 DAP10UT 作为正端输入						
3	ACOPOL	ACO 输出极性选择位。 0: 输出极性不变 1: 输出反向						
2	ACOFCS	ACO 输出数字滤波时钟选择位。 0: 选择系统时钟 1: 选择内部RC32K 时钟						
1:0	ACOFs	ACO 输出数字滤波时长选择位。 00: 禁止数字滤波 01: 1 个周期滤波 10: 2 个周期滤波 11: 3 个周期滤波						

**DIDR2 - 数字输入禁止控制寄存器 2**
**DIDR2 - 数字输入禁止控制寄存器2**

地址: 0x72				默认值: 0x00				
	7	6	5	4	3	2	1	0

	PD7D	PD6D	PD5D	PD4D	PD3D	PD2D	PD1D	PD0D
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name		描述					
7:0	PDxD		PD5:0 引脚数字输入禁止控制位。 当设置 PDxD 位为“1”时，相应引脚的数字输入端被禁止，并一直为零。当使能 PDxD 引脚的模拟功能时，PDxD 引脚的数字输入端功能要关闭，因此须置位 PDxD。 当设置 PDxD 位为“0”时，引脚 PDxD 的数字输入端被使能，引脚上的信号可输入到内部数字逻辑，PDxD 引脚上的模拟功能将不可用。					

DIDR3 - 数字输入禁止控制寄存器 3

DIDR3 - 数字输入禁止控制寄存器 3								
地址: 0x74				默认值: 0x00				
Bit	7	6	5	4	3	2	1	0
	-	-	PE5D	PE4D	PE3D	PE2D	PE1D	PE0D
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name		描述					
7:6	-		保留不用					
5:0	PExD		PE5:0 引脚数字输入禁止控制位。 当设置 PExD 位为“1”时，相应引脚的数字输入端被禁止，并一直为零。当使能 PExD 引脚的模拟功能时，PExD 引脚的数字输入端功能要关闭，因此须置位 PExD。 当设置 PExD 位为“0”时，引脚 PExD 的数字输入端被使能，引脚上的信号可输入到内部数字逻辑，PExD 引脚上的模拟功能将不可用。					

## 22. 模拟比较器 1 (AC1)

- 低于 10mV 的比较精度
- 多路内部/外部比较器输入
- 比较器输出直接到端口
- 可编程输出数字滤波
- 可编程输出极性控制
- 输出发生变化可产生中断

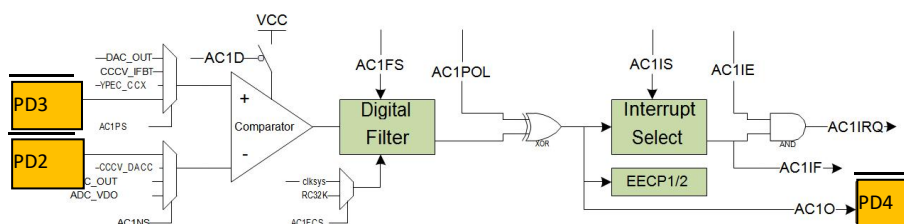
### 22.1. 综述

模拟比较器对正端的值与负端的值进行比较，当正端上的电压比负端上的电压高时，模拟比较器的输出 AC1O 被置位。当 AC1O 的电平发生变化时，信号的边沿可用来触发中断。输出信号 AC1O 还可用来触发定时计数器 1 的输入捕捉功能。

模拟比较器支持多路输入，正端和负端输入源可以选择来自外部端口或者来自多种内部产生的参考源。模拟比较器本身支持失调校准，可以保证比较器工作的一致性。比较器输出端集成了一个硬件可编程数字滤波器，可以根据应用需求，选择合适的滤波设置，获得更加稳定的比较输出。

比较器的输出状态可以通过寄存器获取，也可以产生中断请求，实现更高效的实时事件俘获功能。比较器的输出还可以直接输出到外部引脚。比较器的输出还会连接到 EECF 模块，用以关闭该模块所产生 PWM 信号输出，实现对外部驱动电路的闭环控制。

模拟比较器的结构图：



模拟比较器 1 结构图

## 22.2. 模拟比较器的输入

模拟比较器的两个输入端都有不同的输入源。

正端的输入有外部引脚 AC2P0，以及内部 TYPEC 模块的 TYPEC\_CCX 信号，输入源的选择由位于 AC1XR 中的 AC1PS 位来控制。

负端的输入有外部引脚 AC2N0，内部 DAC 模块的 DAC\_OUT 输出，以及内部 ADC 分压电路的 ADC\_VDO 输出。输入源的选择由位于 AC1SR 的 AC1NS1 位和 AC1NS0 来控制。

## 22.3. 模拟比较器的输出

模拟比较器的输出端集成了一个可编程的数字滤波器，可以滤除瞬时干扰对比较器输出产生的影响。数字滤波器可以根据应用需求，选择合适的滤波时间宽度，只有当比较器的输出稳定持续满足滤波时间的限制，滤波电路才更新比较器的输出，从而达到一个更加稳定的输出结果。

数字滤波器的时钟通过位于 AC1XR 寄存器的 AC1FCS 位来选择，可以选择系统时钟或内部低频时钟 RC32K。滤波的时间长度由位于 AC1XR 寄存器的 AC1FS 位来选择，可以选择 1 个，2 个或三个滤波器时钟周期长度，或者关闭数字滤波功能。

模拟比较器的输出信号经过数字滤波后，还可以进行极性的变化。当设置位于 AC1XR 寄存器的 AC1POL 位为 1 时，输出信号会被反向。当设置位于 AC1XR 寄存器的 AC1POL 位为 0 时，输出信号不会发生改变。

经过极性控制后，模拟比较器的输出可以产生中断请求，当比较器的输出发生变化时，即输出信号产生上升沿或下降沿时，会置位于 AC1SR 寄存器的 AC1IF 位。位置的触发模式由位于 AC1SR 寄存器的 AC1IS 位来选择。当 AC1IF 位被置位，中断使能位 AC1IE 为“1”且全局中断置位时，中断产生。执行模拟比较器中断服务程序时，AC1IF 将自动清零，或对 AC1IF 位写“1”也可清零该位。

经过极性控制后，模拟比较器的输出还可以直接输出到外部引脚上，此功能是通过置位于 AC1XR 寄存器的 AC10E 位来实现的。

经过极性控制后，模拟比较器的输出还可以输出到 EECF 模块用作关闭 PWM 信号输出，详见 EECF1 和 EECF2 模块描述。

## 22.4. 寄存器定义

AC1SR - AC1 控制和状态寄存器

AC1SR - AC1 控制和状态寄存器

地址: 0x2F		默认值: 0x00						
Bit	7	6	5	4	3	2	1	0
	AC1D	AC1NS1	AC10	AC1IF	AC1IE	AC1NS0	AC1IS1	AC1IS0
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	AC1D	AC1 禁止位。 当设置 AC1D 位为“1”时，模拟比较器被关闭。当设置 AC1D 位为“0”时，模拟比较器被开启。						
6	AC1NS1	AC1 负端选择选择控制高位。 AC1NS1 和 AC1NS0 组成 AC1NS 位用来选择 AC1 的负端输入 00: 选择 PD2 作为负端输入 01: 选择 DAOUTBUF 作为负端输入 10: 选择 ADC_MUX 作为负端输入 11: 选 TPC1 的内部 CCX 作为负端输入						
5	AC10	AC1 输出状态位。 模拟比较器的输出经过同步之后直接连到 AC10 位。软件可读取 AC10 位的值来获取模拟比较器的输出值。						
4	AC1IF	AC1 中断标志位。						

		当模拟比较器的输出事件触发了由 AC1IS 位定义的中断模式时，AC1IF 位被置位。当中断使能位AC1IE 为“1”且全局中断置位时，中断产生。执行模拟比较器中断服务程序时，AC1IF 将自动清零，或对 AC1IF 位写“1”也可清零该位。
3	AC1IE	AC1 中断使能位。 当设置AC1IE 位为“1”，且全局中断置位时，模拟比较器的中断被使能。当设置 AC1IE 位为“0”时，模拟比较器的中断被禁止。
2	AC1NS0	AC1 负端选择控制低位。 AC1NS1 和AC1NS0 组成AC1NS 位用来选择AC1 的负端输入
1:0	AC1IS	AC1 中断模式选择控制位。 AC1IS 用来选择模拟比较器的中断触发方式。 00: AC10 的上升沿或下降沿触发 01: 保留 10: AC10 的下降沿触发 11: AC10 的上升沿触发

**AC1XR - AC1 辅助控制寄存器**

AC1XR - AC1 辅助控制寄存器								
地址: 0x3A				默认值: 0x00				
Bit	7	6	5	4	3	2	1	0
	AC1TREN	AC10E	AC1PS1	AC1PS0	AC1POL	AC1FCS	AC1FS1	AC1FS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	AC1TREN	AC1 失调校准使能位。 0: 禁止AC1 失调校准 1: 使能AC1 失调校准						
6	AC10E	AC1 输出到引脚使能位。 0: 禁止AC10 输出到外部引脚PD4 1: 使能AC10 输出到外部引脚PD4						
5:4	AC1PS	AC1 正端选择位。 00: 选择 PD3 作为正端输入 01: 选择 DA00OUTBUF 作为正端输入 10: 选择 DAP0_OUT 作为正端输入 11: 选择 VD0 输出作为正端输入						
3	AC1POL	AC1 输出极性选择位。 0: 输出极性不变 1: 输出反向						
2	AC1FCS	AC1 输出数字滤波时钟选择位。 0: 选择系统时钟						

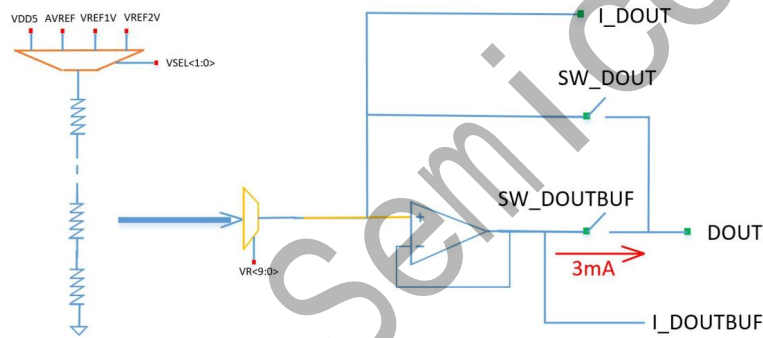
		1: 选择内部RC32K 时钟
1:0	AC1FS	AC1 输出数字滤波时长选择位。 00: 禁止数字滤波 01: 1 个周期滤波 10: 2 个周期滤波 11: 3 个周期滤波

## 23. 数模转换器 (DAC0/1)

- 10 位数模转换输出
- DAC 输出可作为模拟比较器的参考输入
- 支持 DAC 输出直接到端口
- 可选多个分压电源

### 23.1. 综述

DAC 是一个 10 位可编程的数模转换器。DAC 的输入参考可选择系统工作电源，内部基准电压源或外部引脚 AVREF/PB4 输入。DAC 的输出可选择作为内部比较器的输入参考，也可以直接输出到外部引脚 DA80/PB4 上作为外部参考使用。当 DAC 输出到外部引脚时，可以选择通过电压跟随器后再输出，直接用于驱动负载。



DAC 结构图

### 23.2. 寄存器定义

DAOCON - DAC0 控制寄存器

DAOCON - DAC0 控制寄存器

地址: 0xA0		默认值: 0x00						
Bit	7	6	5	4	3	2	1	0
	-	-	DAC_BUFEN	DAC_DBOE	DAC_EN	DAC_DOE	DAC_VSEL1	DAC_VSELO
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name		描述					
7:6	-		保留					
5	DAC_BUFEN		DAC 跟随器使能控制位。 0: 禁止DAC 跟随器 1: 使能DAC 跟随器					
4	DAC_DBOE		DAC 带跟随器输出到引脚使能控制位。 0: 禁止DAC 带跟随器的输出到引脚PB2 上 1: 使能DAC 带跟随器的输出到引脚PB2 上, 同时需置位DAC_BUFEN					
3	DAC_EN		DAC 使能控制位。					

		0: 禁止DAC 模块 1: 使能DAC 模块
2	DAC_DOE	DAC 直接输出到引脚使能控制位 0: 禁止DAC 直接输出到引脚PB2 上 1: 使能DAC 直接输出到引脚PB2 上
1:0	DAC_VSEL	DAC 输入参考电压选择控制位。 00: 系统工作电源VDD5 01: 外部参考输入引脚AVREF/PB4 10: 内部 1.28V 基准电压源 11: 内部 2.56V 基准电压源

**DAOLR - DAC0 输出电压控制寄存器低位**
**DAOLR - DAC 0输出电压控制寄存器**

地址: 0xA1				默认值: 0x00				
Bit	7	6	5	4	3	2	1	0
	DALR[7:0]							
R/W	R/W							
Bit	Name		描述					
7:0	DALR		DAC 输出电压控制低位 DAC 输出电压计算公式位: $DAC\_OUT = DAC\_VIN * (DALR + 1) / 1024$ , DAC_VIN 为通过 DAC_VSEL 选择的输入参考电压					

**DAOHR - DAC0 输出电压控制寄存器高位**
**DAOHR - DAC0 输出电压控制寄存器**

地址: 0xA1				默认值: 0x00				
Bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	DALR9	DALR8
R/W	R/W							
Bit	Name		描述					
1:0	DAHR		DAC 输出电压控制高位 DAC 输出电压计算公式位: $DAC\_OUT = DAC\_VIN * (DAHR + 1) / 1024$ DAC_VIN 为通过 DAC_VSEL 选择的输入参考电压					

**DA1CON - DAC1 控制寄存器**
**DA1CON - DAC 1控制寄存器**

地址: 0xA5				默认值: 0x00				
Bit	7	6	5	4	3	2	1	0
	-	-	DAC_BUFEN	DAC_DBOE	DAC_EN	DAC_DOE	DAC_VSEL1	DAC_VSELO
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name		描述					
5	DAC_BUFEN		DAC 跟随器使能控制位。 0: 禁止DAC 跟随器 1: 使能DAC 跟随器					

4	DAC_DBOE	DAC 带跟随器输出到引脚使能控制位。 0: 禁止DAC 带跟随器的输出到引脚PB3 上 1: 使能DAC 带跟随器的输出到引脚PB3 上, 同时需置位DAC_BUFEN
3	DAC_EN	DAC 使能控制位。 0: 禁止DAC 模块 1: 使能DAC 模块
2	DAC_DOE	DAC 直接输出到引脚使能控制位 0: 禁止DAC 直接输出到引脚PB3 上 1: 使能DAC 直接输出到引脚PB3 上
1:0	DAC_VSEL	DAC 输入参考电压选择控制位。 00: 系统工作电源VDD5 01: 外部参考输入引脚AVREF/PB4 10: 内部 1.28V 基准电压源 11: 内部 2.56V 基准电压源

DA1LR - DAC1 输出电压控制寄存器低位

DA1LR - DAC 1输出电压控制寄存器

地址: 0xA6				默认值: 0x00				
Bit	7	6	5	4	3	2	1	0
	DALR[7:0]							
R/W	R/W							
Bit	Name	描述						
7:0	DALR	DAC 输出电压控制低位 DAC 输出电压计算公式位: $DAC\_OUT = DAC\_VIN * (DALR + 1) / 1024$ , $DAC\_VIN$ 为通过 $DAC\_VSEL$ 选择的输入参考电压						

DA1HR - DAC1 输出电压控制寄存器高位

DA1HR - DAC 1输出电压控制寄存器

地址: 0xA7				默认值: 0x00				
Bit	7	6	5	4	3	2	1	0
	-	-	-	-	-	-	DALR9	DALR8
R/W	R/W							
Bit	Name	描述						
1:0	DAHR	DAC 输出电压控制高位 DAC 输出电压计算公式位: $DAC\_OUT = DAC\_VIN * (DAHR + 1) / 1024$ $DAC\_VIN$ 为通过 $DAC\_VSEL$ 选择的输入参考电压						

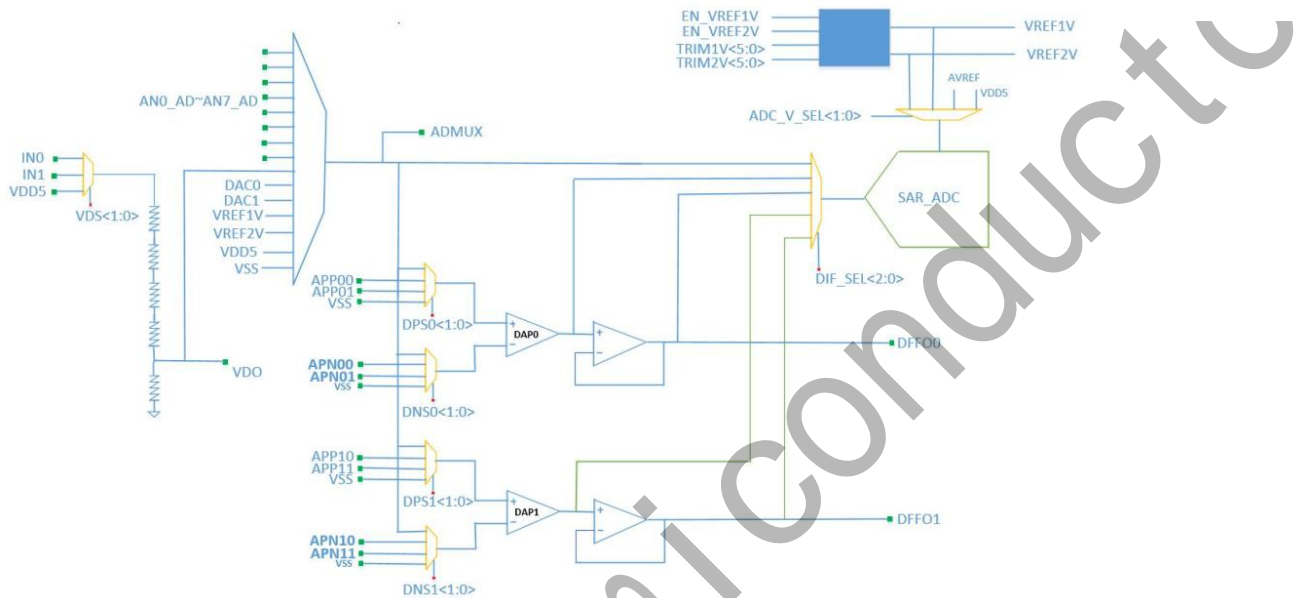
## 24. 模数转换器 (ADC)

- 2 位分辨率, DNL 为  $\pm 1LSB$ , INL 为  $\pm 1.5LSB$
- 最高分辨率时采样率高达 150KSPS
- 8 路复用的单端输入通道
- 自带两路运算放大器
- 输入电压范围为 0-VCC
- 内部 1.28V/2.56V 参考电压
- 支持 AVCC 以及外部参考电压输入
- 支持正负方向的失调校准
- 基于中断源的自动开始转换触发模式

- 转换结果支持可选左对齐模式
- 转换结束中断请求

## 24.1. 综述

模数转换器为一个 12 位的逐次逼近型 ADC，内部结构如下图所示：



Analog to Digital Converter 结构图

ADC 与一个 16 通道的模拟多路器连接，能对来自芯片外部端口的 8 路单端输入以及 8 路内部电压源进行采样转换。ADC 的内部模拟输入源包括来自 ADC 内部的多路输入分压器；内部基准电压源；内部模拟参考地；内部温度探测源；内部 DAC 输出电压源；内部 CCCV 模块的分压输出和差分放大输出信号等。

ADC 支持失调校准。失调校准的流程由软件控制。失调校准包括正、反两个方向的校准量。失调校准使能后，ADC 控制器将会自动使用正反两个校准值对 ADC 采样结果进行校准。失调校准的方法请参考本章节相关部分。

## 24.2. ADC 的操作

ADC 通过逐次逼近的方法将输入的模拟电压转换成一个 12 位的数字量。最小值代表 GND，最大值代表基准电压减去 1LSB。基准电压源可以为 ADC 的电源电压 AVCC，外部基准电压 AREF 或内部 1.024V/2.048 的参考电压，通过写 ADMUX 寄存器的 REFS 位来选择。

模拟输入通道可以通过写 ADMUX 寄存器的 MUX 位来选择。任何 ADC 的输入引脚，内部基准电压源，内部模拟参考地，内部温度探测源，内部 DAC 输出电压源，内部 CCCV 模块的分压输出和差分放大输出信号等可作为 ADC 的单端输入。

通过设置 ADCSRA 寄存器的 ADEN 位即可启动 ADC，ADEN 清零时 ADC 并不耗电，因此建议在进入睡眠模式之前关闭 ADC。

ADC 转换结果为 12 位，存放与 ADC 数据寄存器 ADCH 及 ADCL 中。默认情况下转换结果为右对齐，但可通过设置 ADMUX 寄存器的 ADLAR 位变为左对齐。

如果设置为转换结果左对齐，且最高只需要 8 位的转换精度，那么只要读取 ADCH 就足够了。否则要先读取 ADCL，再读取 ADCH，以保证数据寄存器中的内容是同一次转换的结果。一旦读取 ADCL 后，数据寄存器 ADCL 和 ADCH 被锁存，读取 ADCH 后转换结果即可再更新到数据寄存器 ADCL 和 ADCH。

ADC 转换结束可以触发中断。即使转换结束发生在读取 ADCL 与 ADCH 之间，中断仍将触发。

### 启动一次转换

向 ADC 启动转换位 ADSC 位写“1”可以启动单次转换。在转换过程中此位保持为高，直到转换结束后被硬件清零。如果在转换过程中改变了通道，那么 ADC 会在改变通道前完成这一次转换。

ADC 转换有不同的触发源。设置 ADCSRA 寄存器的 ADC 自动触发允许位 ADATE 可以使能自动触发。设置 ADCSRB 寄存器的 ADC 触发选择位 ADTS 可以选择触发源。当所选的触发信号产生上升沿时，ADC 预分频器复位并开始转换。这提供了一个在固定时间间隔下启动转换的方法。转换结束后即使触发信号仍然存在，也不会启动一次新的转换。如果在转换过程中触发信号又产生了一个上升沿，这个上

升沿也将被忽略。即使特定的中断被禁止或全局中断使能位为“0”，其中断标志仍将置位。这样可以在不产生中断的情况下触发一次转换。但是为了在下次中断事件发生时触发新的转换，必须将中断标志清零。

使用 ADC 中断标志作为触发源，可以在当前进行的转换结束后即开始下一次 ADC 转换。之后 ADC 便工作于连续转换模式，持续地进行采样并对 ADC 数据寄存器进行更新。第一次转换是通过往 ADCSRA 寄存器的 ADSC 位写“1”来启动。在此模式下，后续的 ADC 转换不依赖于 ADC 中断标志 ADIF 是否置位。

如果使能了自动触发，置位 ADCSRA 寄存器的 ADSC 将启动单次转换。ADSC 标志还可用来检测转换是否在进行之中。不论转换是如何启动，在转换过程中 ADSC 一直为“1”。

### 24.3. 预分频及 ADC 转换时序

在默认条件下，逐次逼近电路需要一个从 300kHz 到 3MHz 的输入时钟以获得最大精度。如果所需的转换精度低于 8 位，那么输入时钟的频率可以高于 3MHz，以达到更高的采样率。

ADC 模块包括一个预分频器，它可以由系统时钟来产生可接受的 ADC 输入时钟。预分频器通过 ADCSRA 寄存器的 ADPS 位进行设置。置位 ADCSRA 寄存器的 ADEN 将使能 ADC，预分频器开始计数。只要 ADEN 位为“1”，预分频器就持续计数，直到 ADEN 被清零。

ADCSRA 寄存器的 ADSC 被置位后，单端转换在下一个 ADC 时钟周期的上升沿开始启动。正常转换需要 15 个 ADC 时钟周期。ADC 使能（ADCSRA 寄存器的 ADEN 置位）后需要 50 个 ADC 输入时钟周期初始化模拟电路，之后才能有效进行第一次转换。

在 ADC 转换过程中，采样保持在转换启动之后的 1.5 个 ADC 输入时钟开始，而第一次 ADC 转换的结果输出则发生在启动之后的 14.5 个 ADC 输入时钟。转换结束后，ADC 结果被送入 ADC 数据寄存器，且 ADIF 标志位被置位。ADSC 同时被清零。之后软件可以再次置位 ADSC 标志或自动触发，从而启动一次新的转换。

### 24.4. 采样通道与参考电压

ADMUX 寄存器中的 MUX 及 REFS 通过临时寄存器实现了单缓冲。CPU 可对临时寄存器进行随机访问。在转换启动之前，CPU 可随时对通道及基准源的选择进行配置。为了保证 ADC 有充足的采样时间，一旦转换开始后，就不允许通道及基准源选择的配置。在转换完成（ADCSRA 寄存器的 ADIF 置位）之后，通道及基准源的选择才会被更新。转换的开始时刻为 ADSC 置位后的下一个 ADC 输入时钟的上升沿。因此，建议用户在置位 ADSC 之后的一个 ADC 输入时钟周期内，不要操作 ADMUX 以选择新的通道及基准源。

使用自动触发时，触发事件发生的时间是不确定的。为了控制新设置对转换的影响，在更新 ADMUX 寄存器时要特别小心。若 ADATE 及 ADEN 都置位，则中断时间可以在任意时刻发生，从而自动触发，启动 ADC 的转换。如果在此期间改变 ADMUX 寄存器的内容，那么用户就无法辨别下一次转换是基于旧的配置还是新的配置。建议用户在以下安全时刻对 ADMUX 进行更新：

- 1) ADATE 或 ADEN 位为“0”；
- 2) 在转换过程中，但是在触发事件发生后至少一个 ADC 输入时钟周期；
- 3) 转换结束之后，但是在触发源的中断标志清零之前。如果在上面所提到的任何一种情况下更新 ADMUX，那么新配置将在下一次转换前生效。

选择 ADC 输入通道时须注意，在启动转换之前先选定通道，在 ADSC 置位后的一个 ADC 输入时钟周期之后就可以选择新的模拟输入通道，但最简单的办法是等到转换结束之后再改变通道。

ADC 的参考电压源  $V_{ref}$  反映了 ADC 的转换范围。若单端通道电平超过了  $V_{ref}$ ，其转换结果将接近最大值 0x3FF。 $V_{ref}$  可以是 AVCC，外接 AREF 引脚的电压，内部 1.28V 或 2.56V 基准电压源。

### 24.5. 多路输入分压电路 (VDS)

ADC 内部包含一个多路输入的分压模块。分压输入电压源可选来自外部 ADC 输入通道 (ADC4/ADC5) 或者模拟工作电源。分压电路的输出为所选输入电压的 1/5，连接到 ADC 的内部输入通道 9，还可以输出到模拟比较器 1 的负端输入通道 3。分压电路的输出可用于内部失调校准，还可用于电源电压检测等类似应用。分压电路相关功能主要由 ADCSRD 寄存器控制实现。

### 24.6. ADC 失调校准

由于制造工艺的偏差以及电路结构的固有特性，会造成 ADC 内部比较器电路产生不同程度失调误差。因此对失调电压进行补偿，对于产生高精度的 ADC 转换结构非常关键。LGT8FX8P 芯片内部的 ADC 支持失调电压测试相关接口，可以在软件的配合下完成失调的测量和校准。

失调校准的原理：失调校准主要是通过改变内部比较器的输入极性，在正、反两个方向测试 ADC 转换结果。由于正反两个方向失调电压也是表现为两种极性，通过这两次转换结果相减，可以得到一个中间的失调误差值。正常应用时，将转换结果根据这个失调电压进行相应的调整即可。失调校准流程：

1. 配置 VDS 模块，将 VDS 输入源选择为模拟电源 (AVCC)
2. ADC 的参考电压选择为模拟电源 (AVCC)。ADCSRC[SPN] = 0, ADC 读取 4/5VDD 通道，转换值记录为 PVAL；ADCSRC[SPN] = 1, ADC 读取 1/5VDD 通道，转换值记录为 NVAL
5. 将值 (NVAL - PVAL) >>1 存储到 OFR0 寄存器
6. ADCSRC[SPN] = 1, ADC 读取 1/5VDD 通道，转换结果记录为 NVAL
7. ADCSRC[SPN] = 0, ADC 读取 4/5VDD 通道，转换结果记录为 PVAL
8. 将值 (NVAL - PVAL) >> 1 存储到 OFR1 寄存器
9. 设置 ADCSRC[OFEN]=1 使能失调补偿功能 特别注意：由于失调误差有正负方向，以上数据以及运算都为有符号操作。

失调校准过程中需要改变 ADC 相关配置，因此建议失调校准在正常使用的配置之前完成。为了提高校准精度，建议 ADC 读取通道转换时采样多次滤波。

失调校准 OFR0/1 配置完成后，通过 OFEN 位使能自动失调补偿。以后的正常转换后，ADC 控制将根据 ADC 转化结果，自动使用 OFR0/1 进行补偿。

ADC 动态校准 上面介绍的失调校准方法，基于在一个调试环境和测试输入下的失调。当系统环境改变后，ADC 的失调也会随之变化。因此如果能够实现实时的校准补偿，对于克服器件随工作环境变化而导致的性能差异，提高 ADC 测量精度，非常重要。这里提供一种建议使用的算法，基于失调校准算法的原理，可以实现动态补偿工作环境带来的失调误差，获得一致准确的测试结果。这种方法无需计算失调电压，也不用使能失调补偿 (OFEN)。算法只需要通过 SPN 控制 ADC 转换的极性，在不同 SPN 下采样两个测量结果，两个结果中由于失调引入的误差表现为正 负两种方向，因此我们可以简单的通过相加求平均的方法抵消失调产生的误差。我们假设当在 ADC 转换时，失调引入的测试误差为 VOFS，因此控制 SPN 进行连续两次 ADC 转换，所得到的 ADC 转换结果可以表示为：

$$\text{SPN}=1 \text{ 时, } \text{VADC1} = \text{VREL} + \text{VOFS1} \quad \text{SPN}=0 \text{ 时, } \text{VADC0} = \text{VREL} - \text{VOFS0}$$

我们将两次测量结果相加，即可消除掉 VOFS 对实际采样输入 VREL 产生的影响。由于电路的匹配特性，VOFS1 和 VOFS0 可能不会完全相同，但总体上仍然可以实现补偿失调误差的效果。动态失调补偿算法流程：

1. 根据应用需要初始化 ADC 转换参数
2. 设置 SPN=1，启动 ADC 采样，记录 ADC 采样结果为 VADC1
3. 设置 SPN=0，启动 ADC 采样，记录 ADC 采样结果为 VADC2
4. (VADC1 + VADC2) >>1 即为本次 ADC 的转换结果 实际应用中，可以将这种算法与取样平均算法结合，可以得到更加理想的效果。

## 24.7. 寄存器定义

ADCL - ADC 数据低字节寄存器

ADCL - ADC 数据低字节寄存器								
地址: 0x78					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name0	ADC7	ADC6	ADC5	ADC4	ADC3	ADC2	ADC1	ADC0
Name1	ADC3	ADC2	ADC1	ADC0	-	-	-	-
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name		描述					
7:0	ADC[7:0] / ADC[3:0]		ADC 数据低字节寄存器。 当 ADLAR 位为“0”时，ADC 输出数据在寄存器中的存放按低位对齐，即 ADCL 为 ADC[7:0]，如 Name0 所示；当 ADLAR 位为“1”时，ADC 输出数据在寄存器中的存放按高位对齐，即 ADCL 的高两位为 ADC[3:0]，低 4 位无意义，如 Name1 所示。					



ADCH - ADC 数据高字节寄存器

ADCH - ADC 数据高字节寄存器								
地址: 0x79					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name0	-	-	-	-	ADC11	ADC10	ADC9	ADC8
Name1	ADC11	ADC10	ADC9	ADC8	ADC7	ADC6	ADC5	ADC4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:0	ADC[11:8]/ ADC[11:4]	ADC 数据低字节寄存器。 当 ADLAR 位为“0”时, ADC 输出数据在寄存器中的存放按低位对齐, 即ADCH的低两位为ADC[11:8], 高六位无意义, 如Name0 所示; 当ADLAR 位为“1”时, ADC 输出数据在寄存器中的存放按高位对齐, 即ADCH为ADC[11:4], 如Name1 所示。						

ADCSRA - ADC 控制和状态寄存器 A

ADCSRA - ADC 控制和状态寄存器A								
地址: 0x7A					默认值: 0x05			
Bit	7	6	5	4	3	2	1	0
Name	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	ADEN	ADC 使能控制位。 当设置ADEN 位为“1”时, ADC 被使能。 当设置ADEN 位为“0”时, ADC 被禁止。						
6	ADSC	ADC开始转换。 在单次转换模式下, ADSC 置位将启动一次转换。在连续转换模式下, ADSC 置位将启动首次转换。						
5	ADATE	ADC 自动触发使能控制位。 当设置ADATE 位为“1”时, 自动触发功能被使能。所选中触发信号的 上升沿开启一次转换。触发源的选择由ADCSRB 寄存器的ADTS 来控制。 当设置ADATE位为“0”时, 自动触发功能被禁止。						
4	ADIF	ADC 中断标志位。 当ADC 完成一次转换并更新数据寄存器后置位ADIF。若ADC 中断使 能位 ADIE 为“1”且全局中断置位, ADC 中断产生。执行 ADC 中断会 清零ADIF 位, 也可对该位写“1”来清零。						
3	ADIE	ADC 中断使能控制位。 当设置 ADIE 位为“1”且全局中断置位时, ADC 中断被使能。当设置ADIE 位为“0”时, ADC 中断被禁止。						
2:0	ADPS[2:0]	ADC 预分频器选择控制位。 ADPS 选择系统时钟产生ADC 时钟的预分频因子。						
					ADPS[2:0]	预分频因子		
					0	2		
					1	2		

		2	4
		3	8
		4	16
		5	32 (default)
		6	64
		7	128

**ADCSRB - ADC 控制和状态寄存器 B**

ADCSRB - ADC 控制和状态寄存器B																										
地址: 0x7B					默认值: 0x00																					
Bit	7	6	5	4	3	2	1	0																		
Name	-	-	-	-	-	ADTS2	ADTS1	ADTS0																		
R/W	-	-	-	-	-	R/W	R/W	R/W																		
Bit	Name		描述																							
7:3	-		保留																							
2:0	ADTS[2:0]		<p>ADC 自动触发源选择控制位。</p> <p>当设置ADATE位为“1”时，自动触发功能被使能，触发源的选择由ADTS 来控制。当设置ADATE位为“0”时，ADTS的设置无效。所选中触发信号 中断标志的上升沿开启一次转换。当从一个中断标志清零的触发源切换到中断标志置位的触发源会使触发信号产生一个上升沿，如果此时ADEN 置位，ADC 也会开启一次转换。当切换到连续转换模式 (ADTS=0) 时，自动触发功能被禁止。</p> <table border="1"> <thead> <tr> <th>ADTS[2:0]</th> <th>触发源</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>连续转换模式</td> </tr> <tr> <td>1</td> <td>模拟比较器0 中断标志位置位</td> </tr> <tr> <td>2</td> <td>外部中断0 中断标志位置位</td> </tr> <tr> <td>3</td> <td>模拟比较器1 中断标志位置位</td> </tr> <tr> <td>4</td> <td>定时计数器0 溢出中断标志位置位</td> </tr> <tr> <td>5</td> <td>定时计数器1 比较匹配中断标志位置位</td> </tr> <tr> <td>6</td> <td>定时计数器1 溢出中断标志位置位</td> </tr> <tr> <td>7</td> <td>ECP1 中断标志位置位</td> </tr> </tbody> </table>						ADTS[2:0]	触发源	0	连续转换模式	1	模拟比较器0 中断标志位置位	2	外部中断0 中断标志位置位	3	模拟比较器1 中断标志位置位	4	定时计数器0 溢出中断标志位置位	5	定时计数器1 比较匹配中断标志位置位	6	定时计数器1 溢出中断标志位置位	7	ECP1 中断标志位置位
ADTS[2:0]	触发源																									
0	连续转换模式																									
1	模拟比较器0 中断标志位置位																									
2	外部中断0 中断标志位置位																									
3	模拟比较器1 中断标志位置位																									
4	定时计数器0 溢出中断标志位置位																									
5	定时计数器1 比较匹配中断标志位置位																									
6	定时计数器1 溢出中断标志位置位																									
7	ECP1 中断标志位置位																									

**ADMUX - ADC 多路选择控制寄存器**

ADMUX - ADC多路选择控制寄存器												
地址: 0x7C					默认值: 0x00							
Bit	7	6	5	4	3	2	1	0				
Name	REFS1	REFS0	ADLAR	-	CHMUX3	CHMUX2	CHMUX1	CHMUX0				
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W				
Bit	Name		描述									
7:6	REFS[1:0]		<p>参考电压选择控制位。</p> <p>通过设置REFS 控制位来选择参考电压，若在转换过程中改变REFS 的设置，只有等到当前的转换结束之后改变才会起作用。</p> <table border="1"> <thead> <tr> <th>REFS[1:0]</th> <th>参考电压选择</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>AREF</td> </tr> </tbody> </table>						REFS[1:0]	参考电压选择	0	AREF
REFS[1:0]	参考电压选择											
0	AREF											

		1	AVCC	
		2	片内2.56V基准电压源	
		3	片内1.28V基准电压源	
5	ADLAR	转换结果左对齐使能控制位。 当设置ADLAR 位为“1”时，转换结果在ADC 数据寄存器中为左对齐。当设置 ADLAR 位为“0”时，转换结果在 ADC 数据寄存器中为右对齐。		
4	-	保留		
3:0	CHMUX[3:0]	输入通道源选择控制位。		
		CHMUX[3:0]	单端输入源	描述
		0	AN0	外部端口输入
		1	AN1	
		2	AN2	
		3	AN3	
		4	AN4	
		5	AN5	
		6	AN6	
		7	AN7	
		8	VD0	1/5 分压电路输出
		9	DAC0	内部DAC0输出
		10	DAC1	内部DAC1输出
		11	IVREF1V	内部1.28V基准电压源
		12	IVREF2V	内部2.56V基准电压源
13	VDD5			
14	GND	模拟地		

**ADCSRC - ADC 控制和状态寄存器 C**

ADCSRC - ADC 控制和状态寄存器C								
地址: 0x7D						默认值: 0x00		
Bit	7	6	5	4	3	2	1	0
Name	OFEN	OFSF	SPN	-	-	SPD	-	ADTM
R/W	R/W	R/W	R/W	-	-	R/W	-	R/W
Bit	Name	描述						
7	OFEN	1=使能失调补偿; 0=关闭失调补偿						
6	OFSF	ADC 失调电压标志位 ??						
5	SPN	ADC 转换输入极性控制, 仅用于失调校准过程。正常时必须清零						
4:3,1	-	保留不用						
2	SPD	0=ADC 低速转换模式 1=ADC 高速转换模式, 仅用于低阻抗模拟输入						
0	ADTM	测试模式位, 正常工作时必须清零 0=模拟采样信号与AVREF 引脚断开 1=模拟采样信号与AVREF 引脚相连						

## ADCSRD - ADC 控制和状态寄存器 D

ADCSR D - ADC 控制和状态寄存器D								
地址: 0xAD					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	AD_CH01S	ADC_CHD	BGEN1	BGEN0	-	-	VDS1	VDS0
R/W	-	R/W	R/W	-	R/W	R/W	-	R/W
Bit	Name	描述						
7		保留						
6	ADC_CHD	ADC 通道关闭控制位 0=通道开启, 由CHUMX 位选择采样通道 1=通道关闭, 不选择任何采样通道						
5	BGEN1	内部2.56V 基准电压源全局使能 1=使能内部 2.56V 基准电压源 0=关闭内部 2.56V 基准电压源						
4	BGEN0	内部1.28V 基准电压源全局使能 1=使能内部 1.28V 基准电压源 0=关闭内部 1.28V 基准电压源						
3:2	-	保留						
1:0	VDS	分压电路输入源选择 00=VSS 01=IN0 10=IN1 11=VDD5						

## OFR0 - 失调补偿寄存器 0

OFR0 - 失调补偿寄存器0								
地址: 0xA3					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name0	OFR07	OFR06	OFR05	OFR04	OFR03	OFR02	OFR01	OFR00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:0	OFR0	失调补偿寄存器 0; OFR0 为有符号数。以二进制补码格式存储						

## OFR1 - 失调补偿寄存器 1

OFR1 - 失调补偿寄存器1								
地址: 0xA4					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name0	OFR17	OFR16	OFR15	OFR14	OFR13	OFR12	OFR11	OFR10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:0	OFR1	失调补偿寄存器 1; OFR1 为有符号数。以二进制补码格式存储						

## 25. 运算放大器 (DAP)

### 25.1. 寄存器定义

VCAL1 - 内部 1.28V 基准电压校准寄存器

VCAL1 - 内部 1.28V 基准电压校准寄存器								
地址: 0xE8					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name0	-	-	VCAL15	VCAL14	VCAL13	VCAL12	VCAL11	VCAL10
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:6	-	保留						
5:0	VCAL1	内部1.28V 基准电压校准系数						

VCAL2 - 内部 2.56V 基准电压校准寄存器

VCAL2 - 内部 2.56V 基准电压校准寄存器								
地址: 0xE9					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	-	-	VCAL25	VCAL24	VCAL23	VCAL22	VCAL21	VCAL20
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:6	-	保留						
5:0	VCAL2	内部2.56V 基准电压校准系数						

DAPOCRO - DAP0 控制寄存器 0

DAPOCRO - DAP0 控制寄存器 0								
地址: 0xBA					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	Dap_en	Dap_ga1	Dap_ga0	-	Dap_dns1	Dap_dns0	Dap_dps1	Dap_dps0
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	Dap_en	DAP0 放大器使能位						
6	Dap_ga1	DAP 放大倍数控制位						
		00: 8 倍						
5	Dap_ga0	01: 16 倍						
		10: 32 倍						
4	-	11: 128 倍						
		保留						
3	Dap_dns1	DAP0 负端信号选择位						
2	Dap_dns0	00: 来自多路开关						
		01: 来自 APN0						
1	Dap_dps1	10: 来自 APN1						
		11: 来自 VSS						
1	Dap_dps1	DAP0 正端信号选择位						
		00: 来自多路开关						

0	Dap_dps0	01:来自 APPO 10:来自 APP1 11:来自 VSS
---	----------	---------------------------------------

## DAP1CR0 - DAP1 控制寄存器 0

## DAP1CR0 - DAP1 控制寄存器 0

地址: 0xBE					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
	Dap_en	Dap_ga1	Dap_ga0	-	Dap_dns1	Dap_dns0	Dap_dps1	Dap_dps0
R/W	R/W	R/W	R/W	-	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	Dap_en	DAP1 放大器使能位						
6	Dap_ga1	DAP 放大倍数控制位 00:8 倍						
5	Dap_ga0	01:16 倍 10:32 倍 11:128 倍						
4	-	保留						
3	Dap_dns1	DAP1 负端信号选择位 00:来自多路开关 01:来自 APN0						
2	Dap_dns0	10:来自 APN1 11:来自 VSS						
1	Dap_dps1	DAP1 正端信号选择位 00:来自多路开关 01:来自 APPO						
0	Dap_dps0	10:来自 APP1 11:来自 VSS						

## DAPOCR1 - DAPO 控制寄存器 1

## DAPOCR1 - DAPO 控制寄存器 1

地址: 0xBF					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	-	-	dap_fcks1	dap_fcks0	dap_bufen	dap_difs	-	dap_vcms
R/W	-	-	R/W	R/W	R/W	R/W	-	R/W
Bit	Name	描述						
7:6	-	-						
5	dap_fcks1	CHOPPER 时钟源选择, 保留不用						
4	dap_fcks0	CHOPPER 时钟源选择, 保留不用						
3	dap_bufen	Dap_buf=0, dap0_difs=0, ADC 采 ADC 多路开关						
2	Dap0_difs	Dap_buf=0, dap0_difs=1, ADC 采 DAPO 输出 Dap_buf=1, dap0_difs=1, ADC 采 DAPO_BUF 输出						
1	-							
0	dap_vcm	为 0 时共模点来自 VSS, 为 1 时来自 Ivcm						

**DAP1CR1 - DAP1 控制寄存器 1**

DAP1CR1 - DAP1 控制寄存器 1								
地址: 0xBF					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	-	-	dap_fcks1	dap_fcks0	dap_bufen	dap_difs	-	dap_vcms
R/W	-	-	R/W	R/W	R/W	R/W	-	R/W
Bit	Name	描述						
7:6	-	-						
5	dap_fcks1	CHOPPER 时钟源选择, 保留不用						
4	dap_fcks0	CHOPPER 时钟源选择, 保留不用						
3	-	Dap_buf=0, dap1_difs=0, ADC 采 ADC 多路开关						
2	Dap1_difs	Dap_buf=0, dap1_difs=1, ADC 采 DAP1 输出 Dap_buf=1, dap1_difs=1, ADC 采 DAP1_BUF 输出						
1	-							
0	dap_vcm	为 0 时共模点来自 VSS, 为 1 时来自 Ivcm						

## 26. 其他寄存器定义

**DIDR0 - 数字输入禁用控制寄存器 0**

DIDR0 - 数字输入禁止控制寄存器0								
地址: 0x7E					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	-	PB6D	PB5D	PB4D	PB3D	PB2D	PB1D	PB0D
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
6:0	PBxD[6:0]	PB6:0 引脚数字输入禁止控制位。 当设置 PBxD 位为“1”时, 相应引脚的数字输入端被禁止, 并一直为零。当使用 PBxD 引脚的模拟功能时, PBxD 引脚的数字输入端功能要关闭, 因此须置位 PBxD。 当设置 PBxD 位为“0”时, 引脚 PBxD 的数字输入端被使能, 引脚上的信号可输入到内部数字逻辑, PBxD 引脚上的模拟功能将不可用。						

**DIDR1 - 数字输入禁用控制寄存器 1**

DIDR1 - 数字输入禁止控制寄存器1								
地址: 0x7F					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	-	PC6D	PC5D	PC4D	PC3D	PC2D	PC1D	PC0D
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						

6:0	PCxD[6:0]	<p>PC6:0 引脚数字输入禁止控制位。</p> <p>当设置 PCxD 位为“1”时，相应引脚的数字输入端被禁止，并一直为零。当使 PCxD 引脚的模拟功能时，PCxD 引脚的数字输入端功能要关闭，因此须置位 PCxD。</p> <p>当设置 PCxD 位为“0”时，引脚 PCxD 的数字输入端被使能，引脚上的信号可输入到内部数字逻辑，PCxD 引脚上的模拟功能将不可用。</p>
-----	-----------	---

## 26.1. 输入失调寄存器

DAP1TR - DAP1 输入失调校准寄存器

DAP1TR - DAP1 输入失调校准寄存器								
地址: 0xE0					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	DAPTRIM7	DAPTRIM6	DAPTRIM5	DAPTRIM4	DAPTRIM3	DAPTRIM2	DAPTRIM1	DAPTRIM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:0	DAPTRIM	DAP 输入失调校准系数						

DAP1TC - DAP1 输入失调校准控制寄存器

DAP1TC - DAP1 输入失调校准控制寄存器								
地址: 0xE1					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	BCEN	DAPTREN	DAPTRPN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:3	-	保留						
2	BCEN	系统启动控制位，仅启动过程有效						
1	DAPTREN	DAP 输入失调校准使能位。写1使能						
0	DAPTRPN	DAP 输入失调校准方向位						

DAP0TR - DAPO 输入失调校准寄存器

DAP0TR - DAPO 输入失调校准寄存器								
地址: 0xE2					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	DAPTRIM7	DAPTRIM6	DAPTRIM5	DAPTRIM4	DAPTRIM3	DAPTRIM2	DAPTRIM1	DAPTRIM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:0	DAPTRIM	DAP 输入失调校准系数						

DAP0TC - DAPO 输入失调校准控制寄存器

DAP0TC - DAPO 输入失调校准控制寄存器								
地址: 0xE3					默认值: 0x00			

Bit	7	6	5	4	3	2	1	0
Name	-	-	-	-	-	BCEN	DAPTREN	DAPTRPN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:3	-	保留						
2	BCEN	系统启动控制位，仅启动过程有效						
1	DAPTREN	DAP 输入失调校准使能位。写1使能						
0	DAPTRPN	DAP 输入失调校准方向位						

## 26.2. 校准寄存器

PDS8F208A 内部重要模拟电路支持校准功能，以实现产品量产时模拟特性的精度与一致性。正常情况下，每一个芯片在出厂前都经过了校准，校准数据写入到芯片内部的一个专用区域。当芯片上电时，这些预存的校准信息就会被硬件自动加载到相对应的校准寄存器中，实现对模拟电路的校准调整。

所以，一般应用情况下，我们不需要操作这些校准寄存器，也不建议去改变这些校准寄存器。但某些特性应用情况，需要我们调整校准值，以适应应用需要。

RCMTR - 24MHz HFRC 振荡器校准寄存器

RCMTR - 24MHz HFRC 校准寄存器		
RCMCAL: 0xEA		默认值: 出厂配置
Bits	RCCAL [7:0]	
R/W	R/W	
位定义		
[7:0]	RCCAL	系统上电后，寄存器的值将被系统配置信息中的 RC 校准值替换。

RCKTR - 32KHz RC 振荡器校准寄存器

RCKTR - 32KHz RC 校准寄存器		
RCKCAL: 0xEB		默认值: 出厂设置
Bits	RCKCAL [7:0]	
R/W	R/W	
位定义		
[7:0]	RCKCAL	将校准值写入RCKCAL 寄存器完成对32KHz RC 振荡器的校准

VCAL1 - 内部 1.28V 基准电压校准寄存器

VCAL1 - 内部 1.28V 基准电压校准寄存器								
地址: 0xE8					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name0	-	-	VCAL15	VCAL14	VCAL13	VCAL12	VCAL11	VCAL10
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:6	-	保留						
5:0	VCAL1	内部1.28V 基准电压校准系数						

VCAL2 - 内部 2.56V 基准电压校准寄存器



VCAL2 - 内部 2.56V 基准电压校准寄存器								
地址: 0xE9					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	-	-	VCAL25	VCAL24	VCAL23	VCAL22	VCAL21	VCAL20
R/W	-	-	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7:6	-	保留						
5:0	VCAL2	内部2.56V 基准电压校准系数						

ACOTR - ACO 输入失调校准寄存器								
ACOTR - ACO输入失调校准寄存器								
地址: 0xE6					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	COTRPN	COTRIM6	COTRIM5	COTRIM4	COTRIM3	COTRIM2	COTRIM1	COTRIM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	COTRPN	ACO 输入失调校准方向位						
6:0	COTRIM	ACO 输入失调校准校准系数						

AC1TR - AC1 输入失调校准寄存器								
AC1TR - AC1输入失调校准寄存器								
地址: 0xE7					默认值: 0x00			
Bit	7	6	5	4	3	2	1	0
Name	C1TRPN	C1TRIM6	C1TRIM5	C1TRIM4	C1TRIM3	C1TRIM2	C1TRIM1	C1TRIM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name	描述						
7	C1TRPN	AC1 输入失调校准方向位						
6:0	C1TRIM	AC1 输入失调校准校准系数						

CCACR0 - TYPEC 控制寄存器 0								
CCACR0 - TYPEC 控制寄存器0								
地址: 0x93					默认值: 0x04			
Bit	7	6	5	4	3	2	1	0
	-	IPEN	CCEN	CVEN	RXTR1	RXTR0	-	-
R/W	-	R/W	R/W	R/W	R/W	R/W	-	-
Bit	Name	描述						
7	-	保留						
6	IPEN	CC 上拉电流(Ip)使能						
5	CCEN	CC数据检测比较器使能 (用于PD通讯)						
4	CVEN	CC 电平检测比较器使能 (用于Type-C 协议)						



3:2	RXTR	输入延时校准位 00: 采用15 根电阻串延时 01: 采用6 根电阻串延时 10: 采用4 根电阻串延时 11: 采用2 根电阻串延时
1:0	-	保留

CCTR1 - TYPEC 校准寄存器 1

CCTR1 - TYPEC 校准寄存器1

地址: 0xE4				默认值: 0x2E				
Bit	7	6	5	4	3	2	1	0
	-	RCC1_TRIM1	RCC1_TRIM0	ICC1_TRIM4	ICC1_TRIM3	ICC1_TRIM2	ICC1_TRIM1	ICC1_TRIM0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name		描述					
7	-		保留					
6:5	RCC1_TRIM		CC1 内部电阻校准系数					
4:0	ICC1_TRIM		CC1 内部电流校准系数					

CCTR2 - TYPEC 校准寄存器 2

CCTR2 - TYPEC 校准寄存器2

地址: 0xE5				默认值: 0x2E				
Bit	7	6	5	4	3	2	1	0
	-	RCC2_TRIM1	RCC2TR0	ICC2TR4	ICC2TR3	ICC2TR2	ICC2TR1	ICC2TR0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name		描述					
7	-		保留					
6:5	RCC2_TRIM		CC2 内部电阻校准系数					
4:0	ICC2_TRIM		CC2 内部电流校准系数					

CCTR3 - TYPEC 校准寄存器 3

CCTR3 - TYPEC 校准寄存器3

地址: 0xEC				默认值: 0x2E				
Bit	7	6	5	4	3	2	1	0
	-	RCC2_TRIM1	RCC2TR0	ICC2TR4	ICC2TR3	ICC2TR2	ICC2TR1	ICC2TR0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name		描述					
7	-		保留					
6:5	RCC3_TRIM		CC3 内部电阻校准系数					
4:0	ICC3_TRIM		CC3 内部电流校准系数					

CCTR4 - TYPEC 校准寄存器 4

CCTR4 - TYPEC 校准寄存器4

地址: 0xED				默认值: 0x2E			
----------	--	--	--	-----------	--	--	--



Bit	7	6	5	4	3	2	1	0
	-	RCC2_TRIM1	RCC2TR0	ICC2TR4	ICC2TR3	ICC2TR2	ICC2TR1	ICC2TR0
R/W	-	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Bit	Name		描述					
7	-		保留					
6:5	RCC4_TRIM		CC4 内部电阻校准系数					
4:0	ICC4_TRIM		CC4 内部电流校准系数					

GUID0 寄存器

GUID 寄存器0

地址: 0xF3				默认值:				
Bit	7	6	5	4	3	2	1	0
	ID0							
R/W	R							
Bit	Name		描述					
7:0	ID0							

GUID1 寄存器

GUID 寄存器1

地址: 0xF4				默认值:				
Bit	7	6	5	4	3	2	1	0
	ID1							
R/W	R							
Bit	Name		描述					
7:0	ID1							

GUID2 寄存器

GUID 寄存器2

地址: 0xF5				默认值:				
Bit	7	6	5	4	3	2	1	0
	ID2							
R/W	R							
Bit	Name		描述					
7:0	ID2							

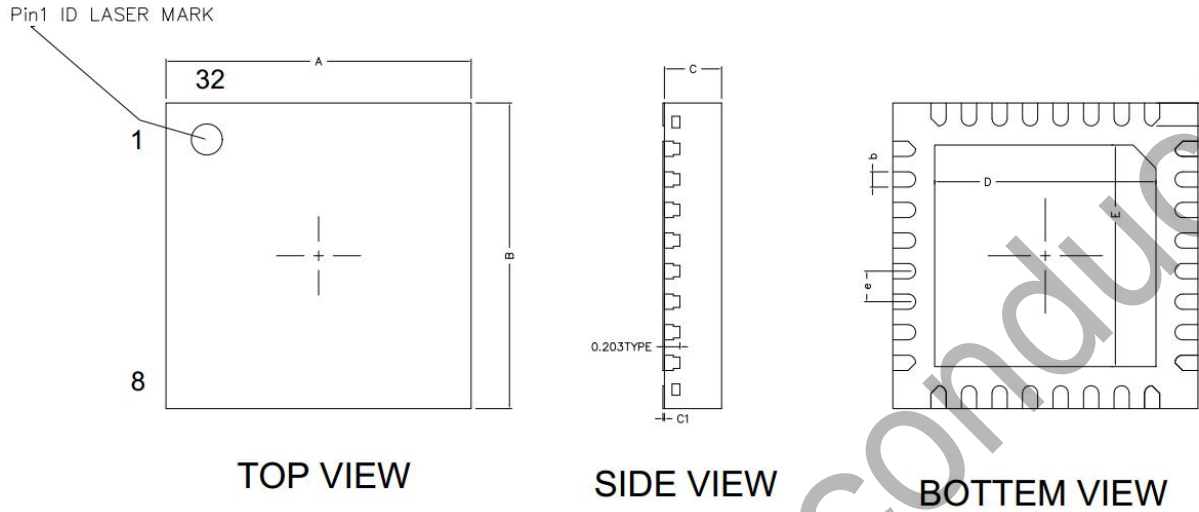
GUID3 寄存器

GUID 寄存器3

地址: 0xF6				默认值:				
Bit	7	6	5	4	3	2	1	0
	ID3							
R/W	R							
Bit	Name		描述					
7:0	ID3							

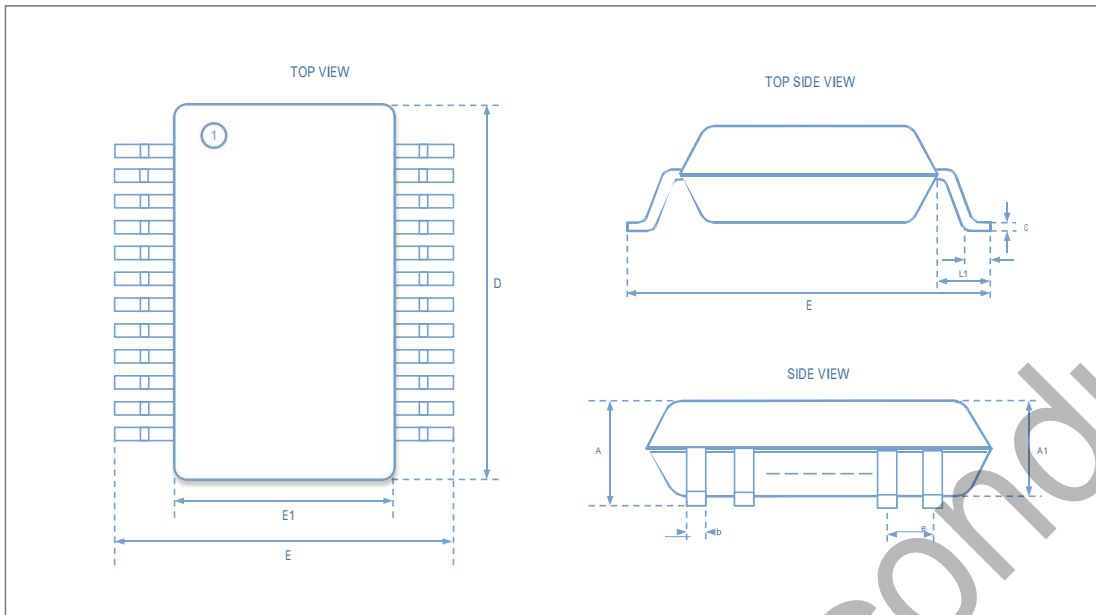
## 27. 封装参数

PITCH\_0.4mm\_4\*4mm\_QFN32



Item	Min.	Typ.	Max.	Unit
A	3.900	4.000	4.100	mm
B	3.900	4.000	4.100	mm
C	0.700	0.750	0.800	mm
C1	0.005	-	0.020	mm
D	2.850	2.900	2.950	mm
E	2.850	2.900	2.950	mm
L	0.250	0.300	0.350	mm
b	0.150	0.200	0.250	mm
e	-	0.400	-	mm

0.635mm-SSOP24 (QSOP24)



Item	Min.	Typ.	Max.	Unit
D	–	8.65	–	mm
A	1.35	–	1.75	mm
b	0.20	–	0.30	mm
e	–	0.635	–	mm
E	–	6.00	–	mm
E1	–	3.90	–	mm
L1	0.55	–	1.27	mm
L	–	–	–	mm
c	0.19	–	0.25	mm

Prodesign Semiconductor