

ADP32F036 数字信号处理器

数据手册

编号: JXDZ3.590.026SJSC

Advancechip



Electronics

湖南进芯电子科技有限公司

2024 年 11 月

V1.6

目 次

1 产品特征	1
2 ADP32F036 简介	2
2.1 产品编码	2
2.2 系统概述	3
2.3 引脚排布及说明	4
2.4 引脚定义	7
3 功能概述	11
3.1 方框图	11
3.2 内部存储映射	12
3.3 简要说明	16
3.3.1 ADP32F036 DSP	16
3.3.2 控制律加速器 (CLA)	16
3.3.3 存储总线 (哈佛总线架构)	16
3.3.4 外设总线	17
3.3.5 实时 JTAG 和分析	17
3.3.6 闪存	17
3.3.7 M0、M1 SARAM	17
3.3.8 L0 SARAM、L4 SARAM 和 L1、L2 以及 L3 DPSARAM	18
3.3.9 引导 ROM	18
3.3.10 器件加密	20
3.3.11 SRAM 诊断	21
3.3.12 外设中断扩展 (PIE) 块	22
3.3.13 外部中断 (XINT1-XINT3)	22
3.3.14 内部零引脚振荡器、振荡器和 PLL	22
3.3.15 看门狗	22
3.3.16 外设时钟	23
3.3.17 低功耗模式	23
3.3.18 外设帧 0, 1, 2, 3 (PFn)	23
3.3.19 通用输入/输出(GPIO) 复用器	24
3.3.20 32 位 CPU 定时器 (0, 1, 2)	24
3.3.21 控制外设	25
3.3.22 串行端口外设	25
3.4 寄存器映射	27
3.5 仿真寄存器	28
3.6 中断	29

3.6.1 外部中断	32
3.6.2 外部中断时序	32
3.7 LDO /BOR/POR	34
3.7.1 片载电压稳压器 (LDO)	34
3.7.1.1 使用片载 LDO	34
3.7.2 片载上电复位 (POR) 和欠压复位(BOR) 电路	34
3.8 系统控制	36
3.8.1 内部零引脚振荡器	38
3.8.2 晶体振荡器选项	39
3.8.3 PLL 时钟模块	40
3.8.4 输入时钟丢失 (NMI 看门狗功能)	42
3.8.5 CPU 看门狗	43
3.9 低功耗模式	45
4 外设	46
4.1 控制律加速器 (CLA)	46
4.2 模拟模块	50
4.2.1 ADC	51
4.2.2 ADC MUX (AIO 的输出功能被删除)	59
4.2.3 比较器模块	60
4.2.4 可编程增益放大器 (PGA)、运算放大器 (OPA)	62
4.3 同步串行通信接口 (SPI) 模块	69
4.3.1 同步串行通信接口(SPI) 主模式时序	72
4.3.2 SPI 从模式时序	76
4.4 异步串行通信接口 (SCI) 模块	79
4.5 本地互连网络 (LIN)	82
4.6 增强型控制器局域网络 (eCAN) 模块	85
4.7 可变速率控制局域网(CANFD)模块	89
4.7.1 简介	89
4.7.2 CANFD 协议说明	89
4.7.3 CANFD 模块特性说明	90
4.7.4 时钟配置	91
4.7.5 引脚配置	92
4.7.6 中断配置	93
4.7.7 传输模式	95
4.7.8 帧类型	97
4.7.9 发送帧	101
4.7.10 CIA603 时间戳	112
4.7.11 接收帧和滤波	113

4.7.12 休眠模式	114
4.7.13 错误	115
4.7.14 寄存器描述	116
4.8 串行 I2C 模块	148
4.8.1 I2C 电气特性和时序	150
4.9 增强型 PWM 模块 (ePWM1/2/3/4/5)	151
4.9.1 增强型脉宽调制器 (ePWM) 时序	156
4.9.2 可编程控制故障区输入时序	156
4.10 高分辨率 PWM (HRPWM)	157
4.10.1 高分辨率 PWM (HRPWM) 时序	157
4.11 增强型捕获模块 (eCAP1)	158
4.11.1 增强型捕获 (eCAP) 时序	159
4.12 高性能捕获 (HRCAP) 模块	160
4.12.1 高性能捕获 (HRCAP) 时序	161
4.13 JTAG 端口	162
4.14 GPIO MUX	163
4.14.1 GPIO 输出时序	169
4.14.2 GPIO 输入时序	169
4.14.3 输入信号的采样窗口宽度	171
4.14.4 低功耗唤醒时序	172
5 电气规范	176
5.1 最大绝对额定值 ⁽¹⁾⁽²⁾	176
5.2 建议的运行条件	176
5.3 电气特性 ⁽¹⁾	177
5.4 流耗	178
5.4.1 减少流耗	179
5.5 散热设计考虑	180
5.6 无信号缓冲的仿真器连接	180
5.7 时序参数符号	182
5.7.1 时序参数的通用注释	182
5.7.2 测试负载电路	182
5.7.3 器件时钟表	183
5.8 时钟要求和特性	183
5.9 电源时序	185
5.10 详细说明	188
5.11 闪存时序	189
6 热性能/机械数据	191

1 产品特征

- **高性能 32 位定点 DSP 内核**
 - 主频最高 100MHz
 - 16 x 16 和 32 x 32 乘累加 MAC 操作
 - 16 x 16 双 MAC 操作
 - 哈佛 (Harvard) 总线架构
 - 快速中断响应和处理
- **可编程控制律加速单元(CLA)**
 - 32 位浮点算术加速器
 - 加速代码与 CPU 代码并行执行
- **单电源供电**
 - IO 3.3~5V, 内核 1.2V
 - 集成 POR、BOR 电路
- **片内存储器资源**
 - 18K×16 位 SARAM (ECC 或奇偶校验保护)
 - 64K x 16 位 Flash
 - 8K x 16 位 BootROM
 - 1K x 16 位 OTP
- **诊断特性**
 - 内存开机自检
- **128 位安全密钥**
- **ADC**
 - 12bit SAR, 转换速率 4MSPS
 - 15 通道, 含 1 个温度传感器通道
 - 输入范围 0~3V, 内部基准
- **运算放大器**
 - 1 个 OP, 可用于母线电流检测放大
 - 3 个 PGA, 可用于相电流检测放大
- **电压比较器**
 - 3 个电压比较器
 - 外部或内置 8bit DAC 电压参考
 - 输出关联 TZ, 支持逐周期封波保护
- **增强型控制外设**
 - 3 个 32 位定时/计数器
 - 5 个 16 位定时/计数器
 - 10 路 PWM 输出 (5 路支持 HRPWM)
 - 2 个捕获单元 (CAP1/HRCAP1)
- **中断**
 - 最多 54 个由 PIE 设置的中断
- **串行通讯外设**
 - CAN、SPI、SCI、LIN、IIC
 - 1 通道 CANFD
- **IO**
 - 27 个通用 IO
- **时钟**
 - 10M 片内振荡器
 - 石英晶体振荡器/外部输入模式
 - PLL 倍频系数 1x~12x
- **支持 WDT**
- **片载闪存 / SARAM/OTP 块的代码安全**
- **支持 JTAG 在线仿真**
 - 分析和断点功能
 - 基于硬件的实时调测
- **LQFP48 封装**
- **QFN48 封装**
- **QFN56 封装**
- **LQFP32 封装**
- **QFN28 封装**
- **TSSOP20 封装**
- **环境温度范围 Q: -40°C~+125°C**
(T_j=150°C通过 AEC-Q100 认证)
- **环境温度范围 Q0: -40°C~+150°C**
(T_j=175°C通过 AEC-Q100 认证)
- **环境温度范围 A: -40°C~+85°C**
- **环境温度范围 S: -40°C~+125°C**

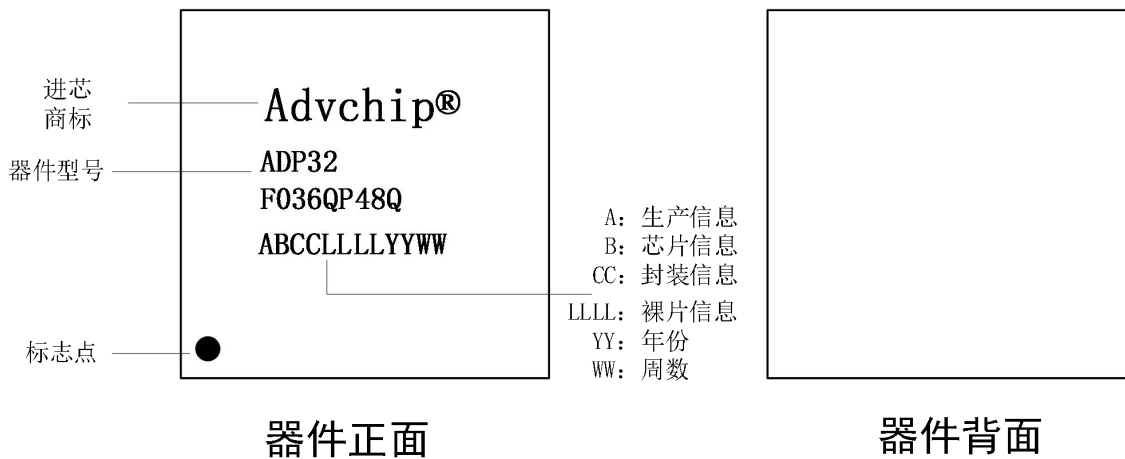
2 ADP32F036 简介

2.1 产品编码

ADP32F036 产品代号编码规则：



器件标识：



2.2 系统概述

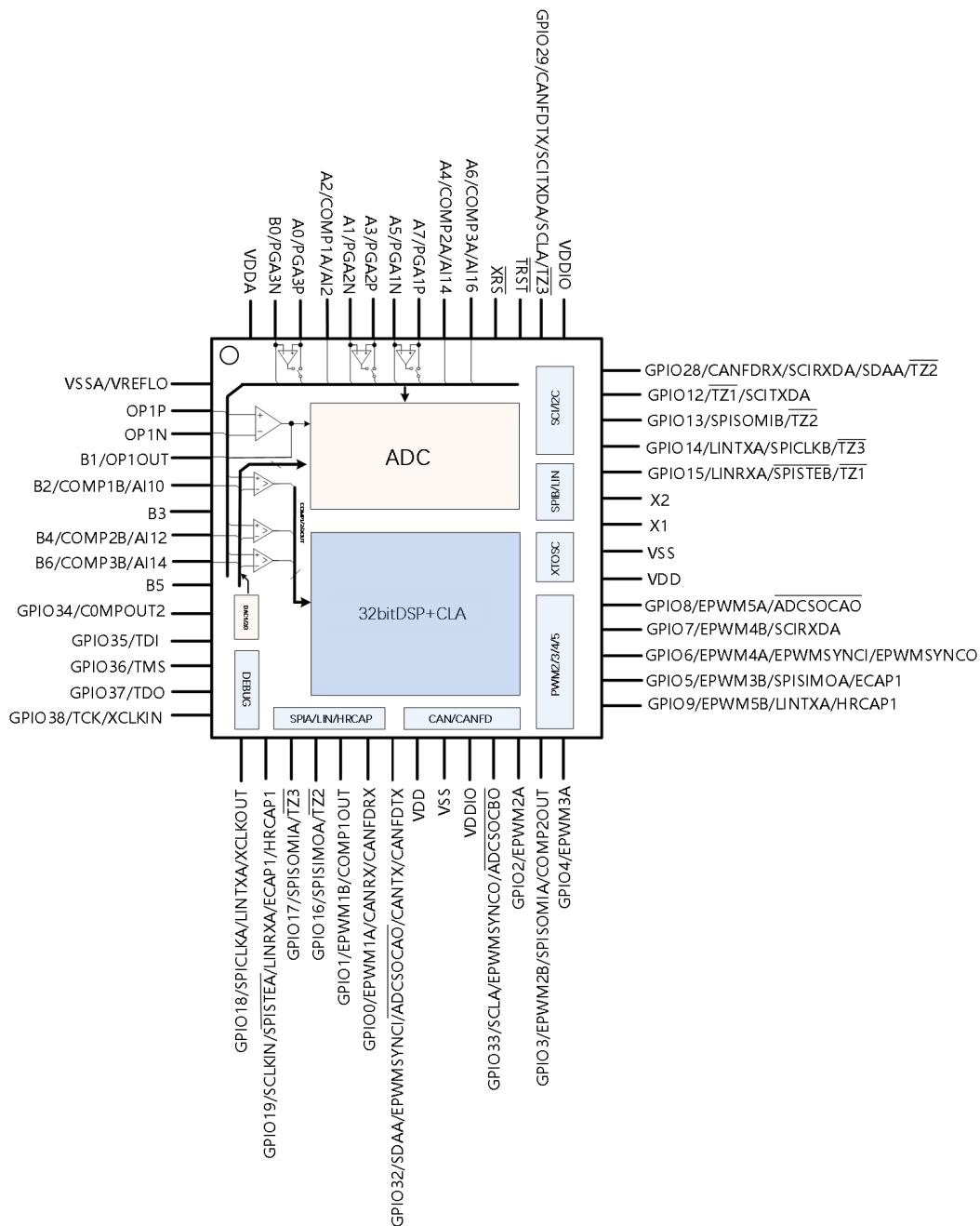


图 2-1 ADP32F036 系统框图

ADP32F036 器件包含一个 32 位主 DSP 内核和一个 32 位控制率加速器，运行频率可达 100MHz，该系列器件集成典型增强型控制外设，如 ePWM、eCAP、HRCAP、ADC、电压比较器、温度传感器，以及典型通讯模块如 CAN、CANFD、LIN 等。

Pin diagram of the ADP33203F036 showing pin numbers 1 through 28 and their corresponding functions. The chip is shown in a square package with pins on all four sides. Pin 1 is at the top left, and pin 28 is at the bottom right. The chip is labeled 'ADP33203' in the center.

Pin	Function
1	VSSA/VREFLO
2	OP1P
3	OP1N
4	B1/OP1OUT
5	I2/COMP1B/A1I0
6	B3
7	B4/COMP2B/A1I2
8	B6/COMP3B/A1I4
9	B5
10	GPIO34/COMP20U1/COMP30U1
11	GPIO35/TDI
12	GPIO36/TMS
13	GPIO37/TDO
14	GPIO38/CLK/XCLKIN
15	GPIO16/SPICLK/INTXA/CLKOUT
16	GPIO17/SPICLK/INTXA/HRCAP
17	GPIO17/SPICLK/INTXA
18	GPIO16/SPICLK/INTXA
19	GPIO17/SPICLK/INTXA
20	GPIO17/SPICLK/INTXA
21	GPIO17/SPICLK/INTXA
22	GPIO17/SPICLK/INTXA
23	GPIO17/SPICLK/INTXA
24	GPIO17/SPICLK/INTXA
25	GPIO17/SPICLK/INTXA
26	GPIO17/SPICLK/INTXA
27	GPIO17/SPICLK/INTXA
28	GPIO17/SPICLK/INTXA

Pin diagram of the ADP32F036 IC. The chip is a square package with pins numbered 1 to 32. The top pins (1-8) are labeled: 1 VSSA/VREFL0, 2 0P1P, 3 0P1N, 4 B1/0P1OUT, 5 B2/COMP1B/A110, 6 B4/COMP2B/A112, 7 GP1035/TDI, 8 GP1036/TMS. The bottom pins (9-16) are labeled: 9 GP1037/TDO, 10 GP1038/TCK/XCLKIN, 11 GP1039/XCLKIN/SP1S15/TAU/LINRMA/ECAP1/HRCP1, 12 GP1032/SDAA/EPWMSYNCI/ALCSUCM3/CANTX/CANFDTX, 13 VDD, 14 VSS, 15 GP1040/IO000, 16 GP103/EPWM2B/SP1S00LA/COMP20UT. The right pins (17-24) are labeled: 17 GP105/EPWM3B/SP1S10MA/ECAP1, 18 GP107/EPWM4B/SC1RXDA, 19 GP102/EPWM2A, 20 GP104/EPWM3A, 21 GP106/EPWM4A/EPWMSYNCI/EPWMSYNCO, 22 GP1028/CANFDRX/SC1RXDA/SDAA/TZ2, 23 GP1029/CANFDTX/SC1TXDA/SCLA/TZ3, 24 TRST. The left pins (25-32) are labeled: 25 XCS, 26 A4/COMP2A/A14, 27 A7/PGA1P, 28 A5/PGA1N, 29 A3/PGA2P, 30 A1/PGA2N, 31 A2/COMP1A/A12, 32 VDDA.

湖南进芯电子科技有限公司

图 2-6 展示了 28 引脚 QFN 引脚排布。

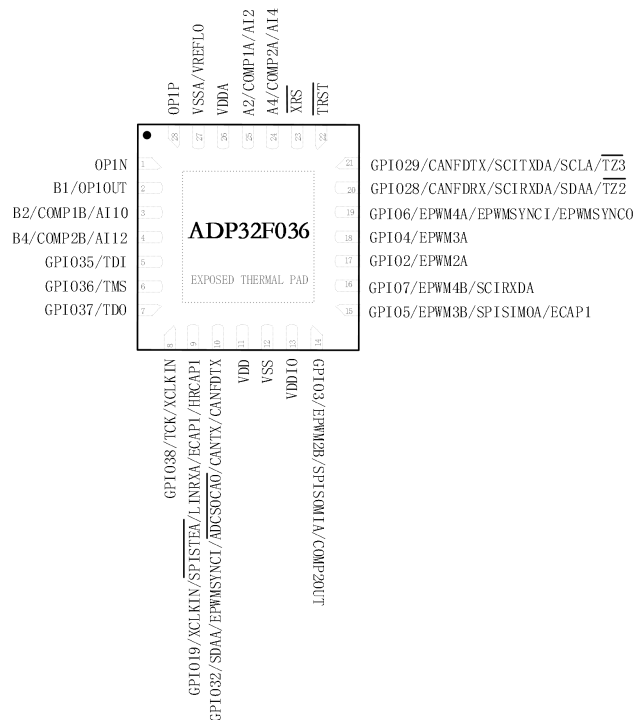


图 2-6 ADP32F036 芯片 QFN28 封装引脚排布

图 2-7 展示了 20 引脚 TSSOP 引脚排布。

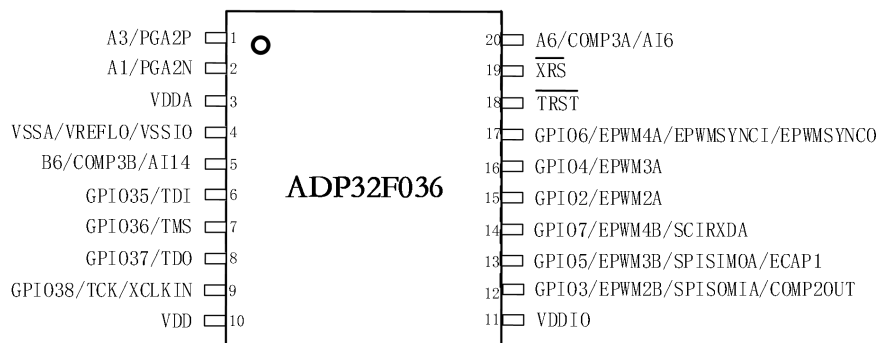


图 2-7 ADP32F036 芯片 TSSOP20 封装引脚排布

2.4 引脚定义

除 JTAG 引脚外，引脚复位时默认为 GPIO（另外注明除外）功能。所有 GPIO 引脚为 I/O/Z 且带有一个内部上拉电阻器，用户可软件配置启用/禁用此上拉电阻；复位时启用 GPIO 引脚的上拉电阻（PWM 引脚上的上拉电阻在复位时并不启用），AI 数字输入引脚无内部上拉电阻。

注意：AI 对应的数字输入引脚电平为 0~3V。

表 2-1 ADP32F036 引脚定义

名称	管脚					类型	功能说明
	QFP48/QFN48	QFN56	LQFP32	QFN28	TSSOP20		
OP1N	1	3	3	1	—	模拟输入	OPA1 输入负端
B1 OP1OUT	2	4	4	2	—	模拟输入 模拟输出	ADC 通道输入 B1 OPA1 输出
B2 COMP1B AI10	3	5	5	3	—	模拟输入 模拟输入 数字输入	ADC 通道输入 B2 比较器 1B 输入端 数字输入 AI10 (0~3V)
B3	4	6	—	—	—	模拟输入	ADC 通道输入 B3
B4 COMP2B AI12	5	7	6	4	—	模拟输入 模拟输入 数字输入	ADC 通道输入 B4 比较器 2B 输入端 数字输入 AI12 (0~3V)
B6 COMP3B AI14	6	8	—	—	5	模拟输入 模拟输入 数字输入	ADC 通道输入 B6 比较器 3B 输入端 数字输入 AI14 (0~3V)
B5	7	9	—	—	—	模拟输入	ADC 通道输入 B5
GPIO34 COMP2OUT COMP3OUT	8	10	—	—	—	数字 IO	通用 IO34 比较器 2 输出 比较器 3 输出
GPIO35 TDI	9	11	7	5	6	数字 IO	通用 IO35 JTAG 数据输入
GPIO36 TMS	10	12	8	6	7	数字 IO	通用 IO36 JTAG 模式选择
GPIO37 TDO	11	13	9	7	8	数字 IO	通用 IO37 JTAG 数据输出
GPIO38 TCK XCLKIN	12	14	10	8	9	数字 IO	通用 IO38 JTAG 时钟输入 外部时钟输入
GPIO18 SPICLK_A LINTX_A XCLKOUT	13	15	—	—	—	数字 IO	通用 IO18 SPI-A 时钟 LIN-A 发送 时钟输出
GPIO19 XCLKIN SPISTEA LINRX_A ECAP1 HRCAP1	14	16	11	9	—	数字 IO	通用 IO19 外部时钟输入 SPI-A 从发送使能 LIN-A 接收 CAP1 高精度 CAP1
GPIO17 SPISOMIA	15	17	—	—	—	数字 IO	通用 IO17

名称	管脚					类型	功能说明
	QFP48/QFN48	QFN56	LQFP32	QFN28	TSSOP20		
TZ3							SPI-A 从输出、主输入 TZ3 保护输入
GPIO16 SPISIMOA TZ2	16	18	——	——	——	数字 IO	通用 IO16 SPI-A 从输入、主输出 TZ2 保护输入
GPIO1 EPWM1B COMP1OUT	17	19	——	——	——	数字 IO	通用 IO1 PWM1B 输出 比较器 1 输出
GPIO0 EPWM1A CANRX CANFDRX	18	20	——	——	——	数字 IO	通用 IO0 PWM1A、HRPWM 输出 CAN 接收 CANFD 接收
GPIO32 SDAA EPWMSYNCI ADCSOCAO CANTX CANFDTX	19	21	12	10	——	数字 IO	通用 IO32 IIC-A 数据 PWM 外部同步脉冲输入 ADC 启动转换 A CAN 发送 CANFD 发送
VDD	20	22	13	11	10	电源	内核电源 1.2V，外接去耦电容（禁止外接电源）
X2	21	——	——	——	——	模拟输出	晶体振荡器输出
X1	22	——	——	——	——	模拟输入	晶体振荡器输入
VSS	23	23	14	12		地	IO 及数字地
VDDIO	24	24	15	13	11	电源	IO 及内部 LDO 电源 3.3V~5V，外接去耦电容
GPIO33 SCLA EPWMSYNCO ADCSOCBO	——	25	——	——	——	数字 IO	通用 IO33 I2C 时钟开漏双向端口 增强型 PWM 外部同步脉冲输入 ADC 转换开始 B
GPIO3 EPWM2B SPISOMIA COMP2OUT	25	27	16	14	12	数字 IO	通用 IO3 PWM2B 输出 SPI-A 从机输出、主机输入 比较器 2 输出
GPIO5 EPWM3B SPISIMOA ECAP1	26	30	17	15	13	数字 IO	通用 IO5 PWM3B 输出 SPI-A 从机输入、主机输出 CAP1
GPIO7 EPWM4B SCIRXDA	27	32	18	16	14	数字 IO	通用 IO7 PWM4B 输出 SCI-A 接收
GPIO9 EPWM5B LINTXA HRCAP1	28	29	——	——	——	数字 IO	通用 IO9 PWM5B 输出 LIN-A 发送 高精度 CAP1
GPIO2 EPWM2A	29	26	19	17	15	数字 IO	通用 IO2 PWM2A 输出，可配置为高精度（HRPWM）
GPIO4 EPWM3A	30	28	20	18	16	数字 IO	通用 IO4 PWM3A 输出，可配置为高精度（HRPWM）
GPIO6 EPWM4A EPWMSYNCI EPWMSYNCO	31	31	21	19	17	数字 IO	通用 IO6 PWM4A 输出，可配置为高精度（HRPWM） PWM 外部同步脉冲输入 PWM 外部同步脉冲输出

名称	管脚					类型	功能说明
	QFP48/QFN48	QFN56	LQFP32	QFN28	TSSOP20		
GPIO8 EPWM5A ADCSOCAO	32	33	—	—	—	数字 IO	通用 IO8 PWM5A 输出, 可配置为高精度 (HRPWM) ADC 启动转换 A
VDD	—	34	—	—	—	电源	内核电源 1.2V, 外接去耦电容 (禁止外接电源)
VSS	—	35	—	—	—	地	IO 及数字地
X1	—	36	—	—	—	模拟输入	晶体振荡器输入
X2	—	37	—	—	—	模拟输出	晶体振荡器输出
GPIO15 LINRXA SPISTEB TZ1	—	38	—	—	—	数字 IO	通用 IO15 LIN-A 接受 SPI-B 从使能 TZ1 保护输入
GPIO14 LINTXA SPICLKB TZ3	—	39	—	—	—	数字 IO	通用 IO14 LIN-A 发送 SPI-B 时钟 TZ3 保护输入
GPIO13 SPISOMIB TZ2	—	40	—	—	—	数字 IO	通用 IO13 SPI-B 从机输出、主机输入 TZ2 保护输入
GPIO12 SCITXDA SPISIMOB TZ1	—	41	—	—	—	数字 IO	通用 IO12 SCI-A 发送 SPI-B 从机输入、主机输出 TZ1 保护输入
GPIO28 CANFDRX SCIRXDA SDAA TZ2	33	42	22	20	—	数字 IO	通用 IO28 CANFD 接收 SCI-A 接收 IIC-A 数据 TZ2 保护输入
VDDIO	—	43	—	—	—	电源	IO 及内部 LDO 电源 3.3V~5V, 外接去耦电容
GPIO29 CANFDTX SCITXDA SCLA TZ3	34	44	23	21	—	数字 IO	通用 IO29 CANFD 发送 SCI-A 发送 IIC-A 时钟 TZ3 保护输入
TRST	35	45	24	22	18	数字输入	JTAG 复位
XRS	36	46	25	23	19	数字 IO	复位
A6 COMP3A AI6	37	47	—	—	20	模拟输入 模拟输入 数字输入	ADC 通道输入 A6 比较器输入 3A 数字输入 AI6 (0~3V)
A4 COMP2A AI4	38	48	26	24	—	模拟输入 模拟输入 数字输入	ADC 通道输入 A4 比较器输入 2A 数字输入 AI4 (0~3V)
A7 PGA1P	39	49	27	—	—	模拟输入	ADC 通道输入 A7 PGA1 输入正端
A5 PGA1N	40	50	28	—	—	模拟输入	ADC 通道输入 A5 PGA1 输入负端
A3 PGA2P	41	51	29	—	1	模拟输入	ADC 通道输入 A3 PGA2 输入正端

名称	管脚					类型	功能说明
	QFP48/QFN48	QFN56	LQFP32	QFN28	TSSOP20		
A1 PGA2N	42	52	30	——	2	模拟输入	ADC 通道输入 A1 PGA2 输入负端
A2 COMP1A AI2	43	53	31	25	——	模拟输入 模拟输入 数字输入	ADC 通道输入 A2 比较器输入 1A 数字输入 AI2 (0~3V)
A0 PGA3P	44	54	——	——	——	模拟输入	ADC 通道输入 A0 PGA3 输入正端
B0 PGA3N	45	55	——	——	——	模拟输入	ADC 通道输入 B0 PGA3 输入负端
VDDA	46	56	32	26	3	电源	模拟电源 3V，外接去耦电容（禁止外接电源）
VSSA/VREFLO	47	1	1	27	4	地	模拟地及参考地（禁止共用外部电流回路）
OP1P	48	2	2	28	——	模拟输入	OPA1 输入正端

3 功能概述

3.1 方框图

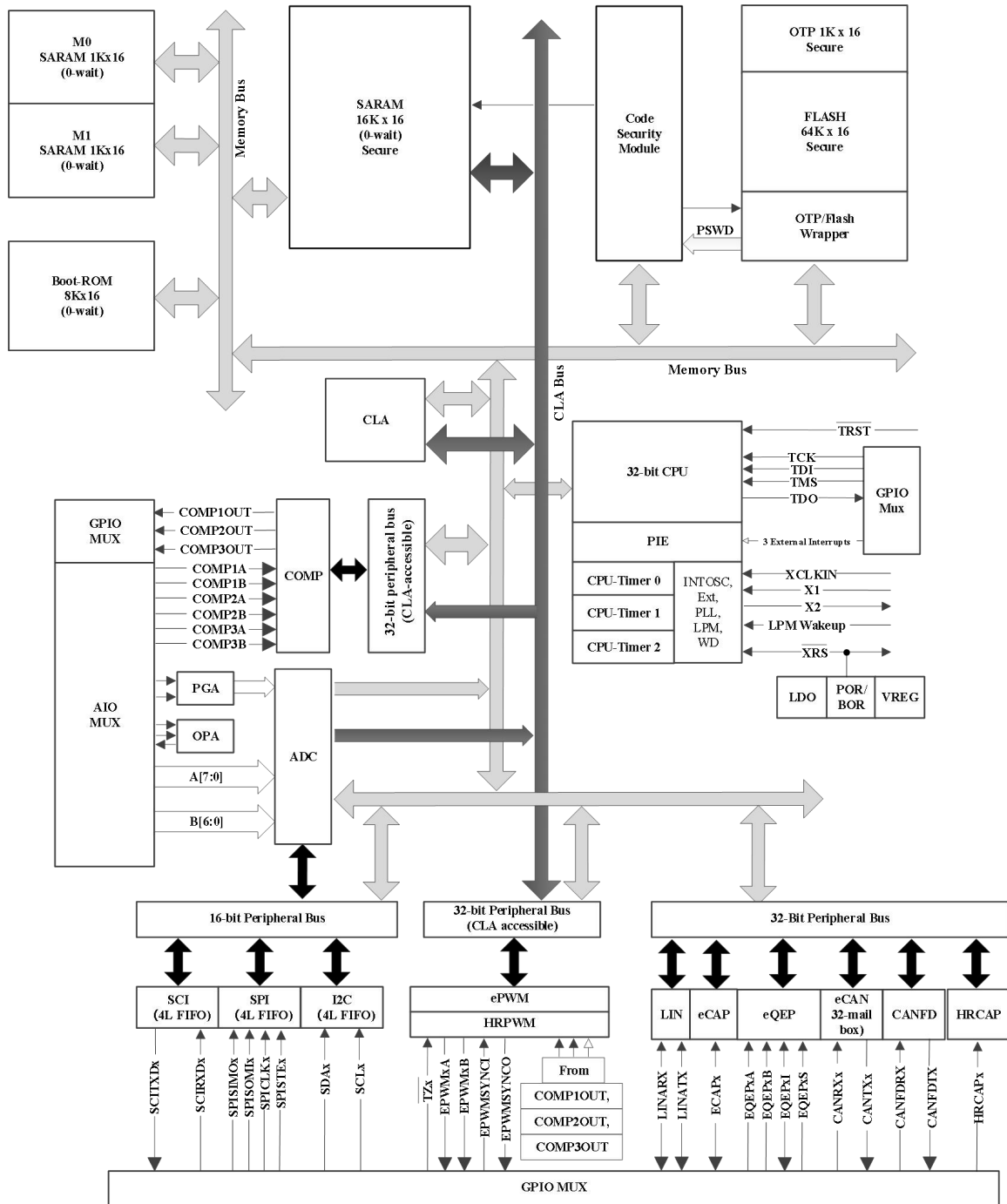


图 3-1 功能框图

3.2 内部存储映射

- 内部存储不可拓展。
- 外设帧 0、外设帧 1、外设帧 2 和外设帧 3 对应的地址空间仅限于数据存储，用户无法在程序空间访问这些地址。
- “受保护”（protected）意味着保存“写后读”操作的顺序，而非流水线顺序。
- 特定内存区域受 EALLOW 保护以防止配置之后的误操作写入。
- 地址 0x3D7C80-0x3D7CC0 包含内部振荡器和 ADC 校准例程，用户禁止对这些内存空间进行编程或者擦除。

	Data Space	Prog Space	
0x00 0000	M0 Vector RAM (Enabled if VMAP = 0)		
0x00 0040	M0 SARAM (1K x 16, 0-Wait)		
0x00 0400	M1 SARAM (1K x 16, 0-Wait)		
0x00 0800	Peripheral Frame 0	Reserved	
0x00 0D00	PIE Vector – RAM (256 x 16) (Enabled if VMAP = 1, ENPIE = 1)		
0x00 0E00	Peripheral Frame 0		
0x00 1400	CLA Registers		
0x00 1480	CLA-to-CPU Message RAM		
0x00 1500	CPU-to-CLA Message RAM		
0x00 1580	Peripheral Frame 0		
0x00 2000	Reserved		
0x00 6000	Peripheral Frame 1 (1K x 16, Protected)		Reserved
0x00 6400	Peripheral Frame 3 (1.5K x 16, Protected)		
0x00 6A00	Peripheral Frame 1 (1.5K x 16, Protected)		
0x00 7000	Peripheral Frame 2 (4K x 16, Protected)		
0x00 8000	L0 SARAM (2K x 16) (0-Wait, Secure Zone + ECSL, Dual-Mapped)		
0x00 8800	L1 DPSARAM (1K x 16) (0-Wait, Secure Zone + ECSL, CLA Data RAM0)		
0x00 8C00	L2 DPSARAM (1K x 16) (0-Wait, Secure Zone + ECSL, CLA Data RAM 1)		
0x00 9000	L3 DPSARAM (4K x 16) (0-Wait, Secure Zone + ECSL, CLA Data RAM)		
0x00 A000	L4 SARAM (8K x 16) (0-Wait, Secure Zone + ECSL, Dual-Mapped)		
0x00 C000	Reserved		
0x3D 7800	User OTP (1K x 16, Secure Zone + ECSL)		
0x3D 7C00	ONLYID		
0x3D 7C04	PARTID		
0x3D 7C80	Calibration Data		
0x3D 7CC0	Get_mode function		
0x3D 7CE0	Reserved		
0x3D 7EB0	Calibration Data		
0x3D 7EB0	Reserved		
0x3E 8000	FLASH (64K x 16, 8 Sectors, Secure Zone+ ECSL)		
0x3F 7FF8	128-Bit Password		
0x3F 8000	L0 SARAM (2K x 16) (0-Wait, Secure Zone + ECSL, Dual-Mapped)		
0x3F 8800	L4 SARAM (8K x 16) (0-Wait, Secure Zone + ECSL, Dual-Mapped)		
0x3F C000	Reserved		
0x3F E000	Boot ROM (8K x 16, 0-Wait)		
0x3F FFC0	Vector (32 Vectors, Enabled if VMAP = 1)		

图 3-2 ADP32F036 内部存储映射

表 3-1 ADP32F036 中闪存扇区的地址

地址范围	程序和数据空间
0x3E 8000-0x3E 9FFF	扇区 H (8K x 16)
0x3E A000-0x3E BFFF	扇区 G (8K x 16)
0x3E C000-0x3E DFFF	扇区 F (8K x 16)
0x3E E000-0x3E FFFF	扇区 E (8K x 16)
0x3F 0000-0x3F 1FFF	扇区 D (8K x 16)
0x3F 2000-0x3F 3FFF	扇区 C (8K x 16)
0x3F 4000-0x3F 5FFF	扇区 B (8K x 16)
0x3F 6000-0x3F 7F7F 0x3F 7F80-0x3F 7FF5 0x3F 7FF6-0x3F 7FF7 0x3F 7FF8-0x3F 7FFF	扇区 A (8K x 16) 当使用代码安全模块时，编程至 0x0000 引导至闪存进入点（程序分支指令） 安全密码（128 位）（不要设定为零）

备注

- 如果启用代码加密功能，从 0x3F 7F80 到 0x3F 7FF5 的所有地址不能存放程序代码和数据，必须编程为 0x0000。
- 如果禁用代码加密功能，则 0x3F 7F80 到 0x3F 7FEF 地址可以存放程序代码和数据。0x3F 7FF0 到 0x3F 7FF5 地址只存放数据，不应包含程序代码。

表 3-2 对比了 0x3F 7F80 到 0x3F 7FF5 地址的使用区别。

表 3-2 使用安全代码模块的区别

地址范围	内存	
	启用代码安全模块	禁用代码安全模块
0x3F 7F80-0x3F 7FEF	用 0x0000 填充	程序代码和数据。
0x3F 7FF0-0x3F 7FF5		只存放数据。

外设帧 1、2、3 被组合在一起，组成一个受读写保护的外设块。这种受保护模式确保了对这些块的所有访问都按照预期的顺序进行。然而，由于流水线的处理机制，如果在不同内存位置写入操作后紧接着进行读取操作，这些操作在 CPU 内存总线上的顺序将被颠倒。此情况可能在某些外设应用中引发问题，尤其是在用户期望写入操作超前于读取操作时。

为了解决这一问题，CPU 提供了一种块保护模式，该模式可以保护一个内存区域，确保操作顺序与预期一致（尽管这可能会增加一些额外的处理周期）。这种模式是可编程的，并且默认情况下会启用，以保护选定的区域。

针对内存映射区域内不同空间的等待状态，如表 3-3 所示。

表 3-3 等待状态

区域 (AREA)	等待状态 (CPU)	注释
M0 和 M1 SARAM	0 - 等待	固定的
外设帧 0	0 - 等待	
外设帧 1	0 - 等待 (写入) 2 - 等待 (读取)	周期可由已生成的外设扩展。 到外设帧 1 寄存器的背靠背写入操作将生成一个 1 周期停止 (1 周期延迟)。
外设帧 2	0 - 等待 (写入) 2 - 写入 (读取)	固定的 周期不可由外设扩展。
外设帧 3	0 - 等待 (写入) 2 - 等待 (读取)	假定 CPU 和 CLA 间无冲突。周期可由已生成的外设扩展。
L0 SARAM	0 - 等待数据和程序	假定没有 CPU 冲突
L1 SARAM	0 - 等待数据和程序	假定没有 CPU 冲突
L2 SARAM	0 - 等待数据和程序	假定没有 CPU 冲突
L3 SARAM	0 - 等待数据和程序	假定没有 CPU 冲突
L4 SARAM	0 - 等待数据和程序	假定没有 CPU 冲突
OTP	可编程 1 - 等待最小	由闪存寄存器设定。 1 - 等待是等待状态所允许的最小值。
闪存	可编程 0 - 页式等待最小值 1 - 随机等待最小值 随机等待 ≥ 页式等待	由闪存寄存器设定。
闪存密码	16	密码位置的等待状态是固定的。
引导 - ROM	0 - 等待	

3.3 简要说明

3.3.1 ADP32F036 DSP

ADP32F036 是一款高性能的 32 位定点 DSP，继承 ADP32F03X DSP 处理器，片上集成了控制率加速（CLA）协处理器，能快速有效的处理多个异步事件。

3.3.2 控制律加速器 (CLA)

控制律加速器是一款单精度（32 位）浮点协处理单元，具有其独立的总线结构、取指令机制和流水线。可指定 8 个独立的 CLA 任务。每个任务由软件或者一个外设（如 ADC、ePWM、eCAP 或者 CPU 定时器 0 等）触发。

CLA 一次执行一项任务直至完成。当一个任务完成时，PIE 产生一个中断到主 CPU，而 CLA 自动开始下一个优先级最高的挂起任务。

CLA 能够直接访问 ADC 的结果寄存器和 ePWM+HRPWM 寄存器。

CPU 和 CLA 之间可通过专用信息 RAM 进行数据传输。

3.3.3 存储总线（哈佛总线架构）

ADP32F036 存储总线架构为增强型哈佛总线架构，包含程序读总线、数据读总线 and 数据写总线。

程序读总线由 22 位地址总线和 32 位数据总线组成。数据读总线 and 数据写总线由 32 位地址总线和 32 位数据总线组成。多总线架构能够实现在一个单周期内取一个指令、读取一个数据值和写入一个数据值。所有连接在存储总线上的外设和内存，具有确定的总线访问优先级。存储总线访问的优先级可概括如下：

最高级：数据写入（内存总线上不能同时进行数据和程序写入）

程序写入（内存总线上不能同时进行数据和程序写入）

数据读取

程序读取（内存总线上不能同时进行程序读取和取指令）

最低级：取指令（内存总线上不能同时进行程序读取和取指令）

3.3.4 外设总线

ADP32F036 外设总线分为外设帧 1、外设帧 2 和外设帧 3。外设帧 1 支持 16 位和 32 位访问；外设帧 2 只支持 16 位访问；外设帧 3 支持 CLA、16 位和 32 位访问。

3.3.5 实时 JTAG 和分析

JTAG 口标准为 IEEE1149.1，支持实时运行模式，在处理器正在运行、执行代码并且处理中断时，可修改存储器、外设和寄存器值。

3.3.6 闪存

ADP32F036 片上闪存分为 Flash 和 OTP。Flash 存储大小为 64K x 16，被分为 8 个扇区，每个扇区大小为 8K x 16；OTP 大小为 1K x 16，地址范围为 0x3D 7800-0x3D 7BFF。

- (1) 用户能够单独擦除、编辑和验证一个扇区，但不能使用一个扇区或者 OTP 区域来执行擦除或者编辑其它扇区。
- (2) 具有特殊内存流水线操作以提升闪存中代码运行速度。
- (3) 闪存和 OTP 被映射到程序和数据空间，用于执行程序或者存储数据信息。
- (4) 地址 0x3F 7FF0-0x3F 7FF5 为数据变量保留区，不能存放程序代码。
- (5) 闪存和 OTP 等待状态可由应用程序配置。
- (6) 可启用闪存流水线操作模式来提升闪存的性能（通过配置闪存选项寄存器实现）。在流水线模式下，线性代码执行的性能将显著高于仅配置由等待状态运行的原始性能。使用闪存流水线模式的性能提升效率视应用程序而定。

3.3.7 M0、M1 SARAM

M0、M1 为单周期访问 RAM，大小为 1K x 16；复位时堆栈指针指向 M1 的起始地址；M0 和 M1 被映射到程序和数据空间，可以执行代码或者存储数据变量。

3.3.8 L0 SARAM、L4 SARAM 和 L1、L2 以及 L3 DPSARAM

L0~L4 为 16K x 16 的单周期访问 RAM，双映射至程序空间和数据空间。L0 大小为 2K x 16；L4 大小为 8K x 16；L1 和 L2 大小均为 1K x 16 并且与 CLA 共用，为 CLA 的数据空间；L3 大小为 4K x 16 并且与 CLA 共用，为 CLA 的程序空间。L1、L2 和 L3 为双端口 SRAM。

3.3.9 引导 ROM

引导 ROM 包含引导程序和用于数学相关算法中的标准表，例如 SIN/COS 波形。

表 3-4 引导模式选择

模式	GPIO37/TDO	GPIO34/ COMP2OUT/ COMP3OUT	TRST	模式
3	1	1	0	取模式(GetMode)
2	1	0	0	等待
1	0	1	0	SCI
0	0	0	0	保留
EMU	x	x	1	仿真引导

3.3.9.1 仿真引导

连接仿真器时，GPIO37/TDO 引脚不能被用于引导模式选择。在这种情况下，引导 ROM 检测一个连接的仿真器，并使用 PIE 向量表中两个保留的 SARAM 位置内的内容来确定引导模式。如果两个位置内的内容均无效，则使用等待引导选项。可在仿真引导中访问所有引导模式选项。

3.3.9.2 GetMode

GetMode 的默认运行状态选项为引导至闪存。通过在 OTP 中设定两个位置，运行状态能够更改为其它的引导选项。如果两个 OTP 位置的内容均为无效，则引导至闪存。可指定以下加载器之一：SCI、SPI、I2C 或者 OTP。

3.3.9.3 引导加载器使用的外设引脚

表 3-5 显示了每一个外设引导加载器所使用的 GPIO 引脚。

表 3-5 外设引导加载引脚

引导加载器	外设加载器引脚
SCI	SCIRXDA (GPIO28) SCITXDA (GPIO29)
SPI	SPISIMOA (GPIO16) SPISOMIA (GPIO17) SPICLKA (GPIO18) SPISTEA(GPIO19)
I2C	SDAA (GPIO32) SCLA (GPIO33)
CAN	CANRX (GPIO0) CANTX (GPIO32)

3.3.10 器件加密

ADP32F036 支持高级别加密以保护用户固件不受逆向工程损坏。该加密模块有一个 128 位密码（针对 16 个等待状态的硬编码），密码由用户写入闪存。一个代码加密模块（CSM）用于保护闪存/ROM/OTP 和 L0/L1/L4 SARAM 块。此加密特性防止未经授权的用户通过 JTAG 端口检查内存内容，或从外部内存执行代码或者试图引导加载恶意软件。为了启用到加密区域的访问，用户必须写入与闪存密码位置内存值相匹配的正确的 128 位“KEY（密钥）”值。

仿真代码安全逻辑电路(ECSL)防止未经授权情形下对保护区域访问。在仿真器连接时，任何加密区域的代码或者数据访问将触发ECSL并断开仿真连接。为了实现加密代码下仿真，同时保持CSM加密内存读取，用户必须向KEY 寄存器的低64 位写入正确的值，这个值与存储在闪存密码位置的低64 位的值相符合。请注意仍须执行闪存内所有128 位密钥的伪读取。如果密码位置的低64 位为全1（未被编辑），那么无须符合KEY 值。

当在器件加密时进行初始调试时，CPU 将开始运行并可执行一个指令来访问一个受保护的ECSL 区域。一旦情况发生，ECSL 将发生错误并使仿真器连接被断开。

注

- 编辑当代码安全密钥时，0x3F7F80 到 0x3F7FF5 间的所有地址不能被用作程序代码或者数据。这些位置必须编程为 0x0000。
 - 如果代码安全特性未使用，地址 0x3F7F80 至 0x3F7FEF 可用于代码或者数据。地址 0x3F7FF0-0x3F7FF5 为数据保留且不能包含程序代码。
 - 128 位密码（位于 0x3F7FF8-0x3F7FFF）不能写入全零。一旦写入全零将永久锁住此器件，请务必慎重。
-

代码安全模块免责声明

此器件所包含的代码安全模块（CSM）旨在保护存储在相关内存（ROM 或者闪存）中的数据，并且由原厂提供质量保证，与其标准条款和条件相一致，符合原厂发布的规范以获得适用于该期间的保修期。

但是，原厂不保证或声明CSM不会受到损坏或破坏，或无法通过其它方式访问相关存储器中存储的数据。而且，除了上述内容外，原厂也未对本器件的CSM或操作做任何保证或表示，包括任何隐含的用于特定用途的商用性或适用性保证。

在任何情况下，原厂均不对任何因使用CSM或本器件引起的任何后果性、特殊、间接、偶然或惩罚性损害负责，无论原厂是否告知此类损害。上述损害包括但不限于数据丢失、信誉损失、无法使用、业务中断或其它经济损失。

3.3.11 SRAM 诊断

ADP32F036 支持 M0、M1、CANFD_SRAM ECC 校验，能够对存储 1bit 错误纠错，2bit 错误及以上报错。对 L0SRAM、L1SRAM、L2SRAM、L3SRAM、L4SRAM、ECAN_SRAM 做奇偶校验。

注意：

ECC 能正确纠正单 bit 错误，且能对双 bit 错误报错，但不能校验多 bit 错误，当多 bit 错误时，电路不能正确给出标志和数据。

表 3-6 SRAM 诊断表

SRAM	ADDR	ECC	奇偶校验
M0SRAM	0x0-0x3FF	YES	
M1SRAM	0x400-0x7FF	YES	
PIE	0xD00-0xE00		
CLA-to-CPU	0x1480-0x14FF		YES
CPU-to-CLA	0x1500-0x157F		YES
L0SRAM	0x8000-0x87FF		YES
L1SRAM	0x8800-0x8BFF		YES
L2SRAM	0x8C00-0x8FFF		YES
L3SRAM	0x9000-0x9FFF		YES
ECANA_SRAM	0x6100-0x61FF		YES
ECANB_SRAM	0x6040-0x60FF		YES
CANFD_SRAM		YES	

系统复位后 FLASH 初始化过程中会读出配置在 FLASH 中 0x3D_7E17~0x3D_7E14 中关于 ECC 使能和 SRAM 初始化选择值。

如果高 32 位不等于 0xAA55_AA55 时，系统不能开启 ECC 和奇偶校验功能，DSP 正常进入程序；

当高 32 位等于 0xAA55_AA55，则 SRAM 打开 ECC 或奇偶校验功能；

如果低 32 位不为 0xAA55_AA55，则内部 SRAM 初始化过程写入全 0；

如果低 32 位为 0xAA55_AA55，则内部 SRAM 在程序运行前进入自检，并会输出自检结果到结果寄存器 0x6212。

ECC 双 bit 错误和奇偶校验错误产生一个中断 DUL_PAR_int（仅报错）；

ECC 单 bit 错误产生另一个中断 SING_int（可校准）。

3.3.12 外设中断扩展 (PIE) 块

PIE 模块将分散的中断源统一管理，将其集中到更少的中断输入点。PIE 能够支持多达 52 个外设中断。52 个中断被分成至 8 组，每组被提供 12 个 CPU 中断线（INT1 或者 INT12）中的 1 个。每个中断带有独立向量，并存储在专用 RAM 快，并支持用户改写。在处理中断时，CPU 会自动提取这些向量，同时保存关键寄存器，整个过程仅需 8 个 CPU 时钟周期，确保了对中断事件的快速响应。中断的优先级可以通过硬件和软件进行控制，且每个中断都可以在 PIE 模块中独立启用或禁用。

3.3.13 外部中断 (XINT1-XINT3)

ADP32F036 提供三个可屏蔽的外部中断（XINT1 至 XINT3），用户可以根据需要选择触发方式，包括负边沿触发、正边沿触发或双边沿触发。中断可以灵活启用或禁用，以适应不同的应用场景。此外，每个中断都配备了一个 16 位的自由运行增计数器，当检测到有效的中断边沿信号时计数器会清零，此功能可用于中断事件计时，从而实现中断事件的精确计时。

这些中断并不占用专用引脚，而是可以通过 GPIO0 至 GPIO31 引脚接收输入信号。这种设计使得中断资源的利用更加灵活，便于用户根据实际需求进行配置。。

3.3.14 内部零引脚振荡器、振荡器和 PLL

ADP32F036 支持多种时钟源，包括内部零引脚振荡器、外部振荡器以及连接到片载振荡器电路的晶振。并且配备了一个 12 种倍频的 PLL，用户可以通过软件在线调整 PLL 倍率，以适应不同的运行需求。此外，PLL 模块还具备旁路模式。

为了更深入地了解时钟相关的电气规范和时序要求，建议参考文档中的“电气规范”。

3.3.15 看门狗

ADP32F036 配备了两个看门狗定时器：一个是用于监控 CPU 的内部看门狗，另一个是用于检测时钟丢失的 NMI 看门狗。为了保持系统稳定，用户需要通过软件在规定的时间内重置 CPU 看门狗计数器。如果未按时重置，CPU 看门狗将触发一个复位信号，重启处理器。用户也可以选择禁用

CPU 看门狗，以适应不同的应用需求。

NMI 看门狗则专门用于监测时钟系统的状态。NMI 看门狗仅在检测到时钟故障时被激活，并可配置生成一个中断或直接触发复位。

3.3.16 外设时钟

每个外设在不需要使用时，可禁用其输入时钟以降低功耗。

3.3.17 低功耗模式

ADP32F036 支持三种低功耗模式：

IDLE：将 CPU 设置为低功耗模式。在此模式下，可以选择性地关闭不必要的外设时钟，仅保留那些在 IDLE 期间必须持续运行的关键外设。当选定外设中断被触发或看门狗定时器溢出，可将处理器从 IDLE 模式中唤醒，恢复到正常工作状态。

STANDBY：关闭到 CPU 和外设的时钟。在此模式下，振荡器和 PLL 仍然运行。一个外部中断事件将唤醒处理器和外设。唤醒操作在检测到中断事件之后的下一个时钟周期执行。

HALT：此模式基本上会关闭设备并将其置于尽可能低的功耗模式。如果将内部零引脚振荡器用作时钟源，则默认情况下，HALT 模式会将其关闭。为了防止这些振荡器关断，可以使用 CLKCTL 寄存器中的 INTOSCnHALTI 位。因此，零引脚振荡器可用于在此模式下为 CPU 看门狗提供时钟。如果片内晶体振荡器用作时钟源，则在此模式下将其关断。复位或外部信号（通过 GPIO 引脚）或 CPU 看门狗可以将设备从此模式中唤醒。

在将设备置于HALT或者STANDBY模式前，CPU时钟 (OSCCLK) 和 WDCLK应来自同一个时钟源。

3.3.18 外设帧 0, 1, 2, 3 (PFn)

此设备将外设分成四个部分。外设映射如下：

PF0： **PIE**：PIE 中断启用和控制寄存器加上 PIE 向量表

闪存：闪存控制、编程、擦除、验证寄存器

定时器：CPU 定时器 0, 1, 2 寄存器

CSM: 代码安全模块 KEY 寄存器

ADC: ADC 结果寄存器

CLA: 控制率加速器寄存器和消息 RAM

PF1: GPIO: GPIO MUX 配置和控制寄存器

eCAN: eCAN 邮箱和控制寄存器

CANFD: CANFD 缓存和控制寄存器

LIN: 本地互联网配置和控制寄存器

eCAP: 增强型捕获模块和寄存器

HRCAP: 高性能捕获模块和寄存器

PF2: SYS: 系统控制寄存器

SCI: SCI 控制和 RX/TX 寄存器

SPI: SPI 控制和 RX/TX 寄存器

ADC: ADC 状态、控制和配置寄存器

I2C: I2C 控制寄存器

XINT: 外部中断寄存器

PF3: ePWM: PWM 寄存器

HRPWM: 高精度 PWM 寄存器

比较器: 比较器模块

3.3.19 通用输入/输出(GPIO) 复用器

大多数的外设信号与 GPIO 复用，复位时所有 GPIO 引脚被配置为输入。用户能够独立设定每一个引脚作为 GPIO 模式还是外设信号模式。对于特定的输入引脚，用户可以选择输入限定系统时钟周期窗口，过滤掉毛刺噪声。GPIO 可用于将设备从特定低功耗模式唤醒。

3.3.20 32 位 CPU 定时器 (0, 1, 2)

CPU 定时器 0、1 和 2 是完全一样的 32 位定时器，可预先设定周期和 16 位时钟预分频。定时器有一个 32 位倒计时寄存器，当寄存器在计数达到 0 时生成一个中断，并自动重新载入一个 32 位的周

期值。

CPU 定时器 0 用于一般用途，并连接到 PIE 模块；CPU 定时器 1 也用于一般用途，可以连接到 CPU 的 INT13；CPU 定时器 2 是为 DSP/BIOS 保留，并连接到 CPU 的 INT14，如果未使用 DSP/BIOS，则 CPU 定时器 2 可供一般使用。

CPU 定时器 2 可由下列任何一种方式进行计时：

- SYSCLKOUT (默认)
- 内部零引脚振荡器 (INTOSC)
- 外部时钟源

3.3.21 控制外设

ePWM： 增强型 PWM 外设支持独立/互补 PWM 生成、前/后沿的可调死区生成、锁存/逐周期跳闸机制。一些 PWM 引脚支持 HRPWM 高分辨率占空比和周期特性。支持更高的死区分辨率、增强的 SOC 和中断生成，以及高级触发，包括基于比较器输出的跳闸功能。

eCAP： 增强型捕获使用 32 位时基，并在连续或者单次捕获模式下记录多达 4 个可编程事件。此外设也可配置为生成辅助 PWM 信号。

ADC： ADC 为一个 12 位转换器，有 15 个单端通道输入引脚。它包含一个用于同步采样的采样保持单元。

比较器： 每个比较器模块由一个模拟比较器和一个内部 8 位基准 (DAC) 组成。

HRCAP： 高分辨率捕获外设可以通过一个 16 位计数器 (从 HCCAPCLK 计时) 以正常捕获模式运行，或者通过利用内置校准逻辑和校准库在高分辨率捕获模式下运行。

3.3.22 串行端口外设

SPI： 可编程比特传输速率；可编程数据长度 (1 至 16 位)；包含一组 4 级深度接收和发送 FIFO。

SCI： 包含一组 4 级深度的接收和发送 FIFO。

I2C: 符合飞利浦半导体内部 IC 总线(I2C-bus) 规范版本 2.1 并由一个 I2C-bus 相连; 通过 I2C 模块, 连接在这个两线制总线上的外部组件能够发送或者接收 8 位数据; 包含一组 4 级深度的接收和发送 FIFO。

eCAN: 支持 32 个邮箱、消息时间戳并与 CAN 2.0B 兼容。

CANFD: 可变速率 CAN, 支持 CAN2.0B。

LIN: LIN 1.3 或者 2.0 兼容外设, 可配置为 SCI 端口模式使用。

3.4 寄存器映射

此器件包含 4 个外设寄存器空间，这些空间分类如下：

外设帧 0：直接映射到 CPU 内存总线的外设，请参阅表 3-7。

外设帧 1：映射到 32 位外设总线的外设，请参阅表 3-8。

外设帧 2：映射到 16 位外设总线的外设，请参阅表 3-9。

外设帧 3：映射到 32 位外设总线并可由 CLA 访问的外设，请参阅表 3-9。

表 3-7 外设帧 0 寄存器⁽¹⁾

名称	地址范围	大小 (x16)	受 EALLOW 保护 ⁽²⁾
仿真寄存器	0x00 0880-0x00 0984	261	支持
系统功耗控制寄存器	0x00 0985-0x00 0987	3	支持
闪存寄存器 ⁽³⁾	0x00 0A80-0x00 0ADF	96	支持
代码安全模块寄存器	0x00 0AE0-0x00 0AEF	16	支持
ADC 寄存器 (0 等待只读)	0x00 0B00-0x00 0B0F	16	否
CPU 定时器 0/1/2 寄存器	0x00 0C00-0x00 0C3F	64	否
PIE 寄存器	0x00 0CE0-0x00 0CFF	32	否
PIE 向量表	0x00 0D00-0x00 0DFF	256	否
CLA 寄存器	0x00 1400-0x00 147F	128	支持
CLA 到 CPU 消息 RAM (忽略 CPU 写入)	0x00 1480-0x00 14FF	128	不适用
CPU 到 CLA 消息 RAM (忽略 CLA 写入)	0x00 1500-0x00 157F	128	不适用

(1) 外设帧 0 中的寄存器支持 16 位和 32 位访问。

(2) 如果寄存器是 EALLOW 受保护的，在 EALLOW 指令被执行前写入无效。EDIS 指令禁用写入以防止杂散代码或指针破坏寄存器值。

(3) 闪存寄存器也受到代码安全模块 (CSM) 的保护。

表 3-8 外设帧 1 寄存器

名称	地址范围	大小 (x16)	受 EALLOW 保护
eCAN 寄存器	0x00 6000-0x00 61FF	512	(1)
eCAP1 寄存器	0x00 6A00-0x00 6A1F	32	否
HRCAP1 寄存器	0x00 6AC0-0x00 6ADF	32	(1)
LIN-A 寄存器	0x00 6C00-0x00 6C7F	128	(1)
CANFD 寄存器	0x00 6C80-0x00 6F7F	768	否
GPIO 寄存器	0x00 6F80-0x00 6FFF	128	(1)

(1) 部分寄存器受 EALLOW 保护。详细信息请参阅各模块寄存器描述。

表 3-9 外设帧 2 寄存器

名称	地址范围	大小 (x16)	受 EALLOW 保护
系统控制寄存器	0x00 7010-0x00 702F	32	是
SPI-A 寄存器	0x00 7040-0x00 704F	16	否
SCI-A 寄存器	0x00 7050-0x00 705F	16	否
NMI 看门狗中断寄存器	0x00 7060-0x00 706F	16	是
外部中断寄存器	0x00 7070-0x00 707F	16	是
ADC 寄存器	0x00 7100-0x00 717F	128	(1)
I2C-A 寄存器	0x00 7900-0x00 793F	64	(1)

(1) 部分寄存器受 EALLOW 保护。详细信息请参阅各模块寄存器说明。

表 3-10 外设帧 3 寄存器

名称	地址范围	大小 (x16)	受 EALLOW 保护
比较器 1 寄存器	0x00 6400-0x00 641F	32	(1)
比较器 2 寄存器	0x00 6420-0x00 643F	32	(1)
比较器 3 寄存器	0x00 6440-0x00 645F	32	(1)
ePWM1 + HRPWM1 寄存器	0x00 6800-0x00 683F	64	(1)
ePWM2 + HRPWM2 寄存器	0x00 6840-0x00 687F	64	(1)
ePWM3 + HRPWM3 寄存器	0x00 6880-0x00 68BF	64	(1)
ePWM4 + HRPWM4 寄存器	0x00 68C0-0x00 68FF	64	(1)
ePWM5 + HRPWM5 寄存器	0x00 6900-0x00 693F	64	(1)

(1) 某些寄存器受 EALLOW 保护。详细信息请参阅各模块寄存器说明。

3.5 仿真寄存器

表 3-11 仿真寄存器

名称	地址范围	大小 (x16)	说明	受 EALLOW 保护
DEVICECNF	0x00 0880 0x00 0881	2	器件配置寄存器	支持
PARTID	0x3D7C04	1	ADP32F036QP48Q ADP32F036QN48Q ADP32F036QN56Q ADP32F036QP32Q ADP32F036QN28Q ADP32F036SP20Q ADP32F036QP48S ADP32F036QN48S ADP32F036QP32A ADP32F036QN28A ADP32F036SP20A 部件 ID 寄存器	否
CLASSID	0x0882	1	类别 ID 寄存器	否
ONLYID	0x3D7C00 0x3D7C01 0x3D7C02 0x3D7C03	4	芯片唯一码 (64 位随机值)	否

3.6 中断

图 3-3 所示为外部和 PIE 中断源。

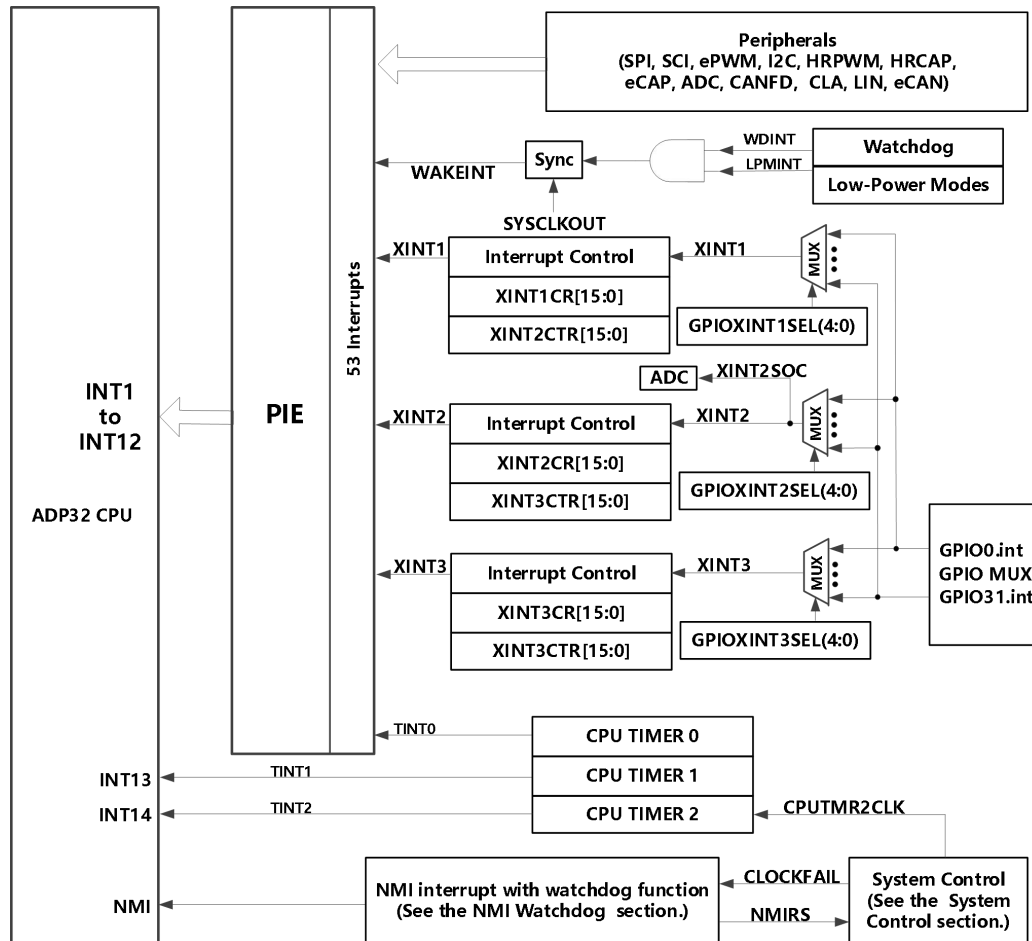


图 3-3 外部和 PIE 中断源

8 个 PIE 块中断组合成一个 CPU 中断。总计 12 个 CPU 中断组，每组有 8 个中断，但本产品只使用其中 52 个中断。

TRAP #VectorNumber 指令将程序控制权转移到与指定向量对应的中断服务程序。

TRAP #0 指令尝试将程序控制转移到复位向量所指向的地址。但是，PIE 向量表不包含复位向量。

因此，在启用 PIE 时，不应使用 TRAP #0 指令，否则将导致未知运行状态。

当 PIE 启用时，TRAP #1 到 TRAP #12 指令会将程序控制权转移到与 PIE 组中的第一个向量对应的中断服务程序。例如：TRAP #1 指令从 INT1.1 获取向量，TRAP #2 指令从 INT2.1 获取向量，依此类推。

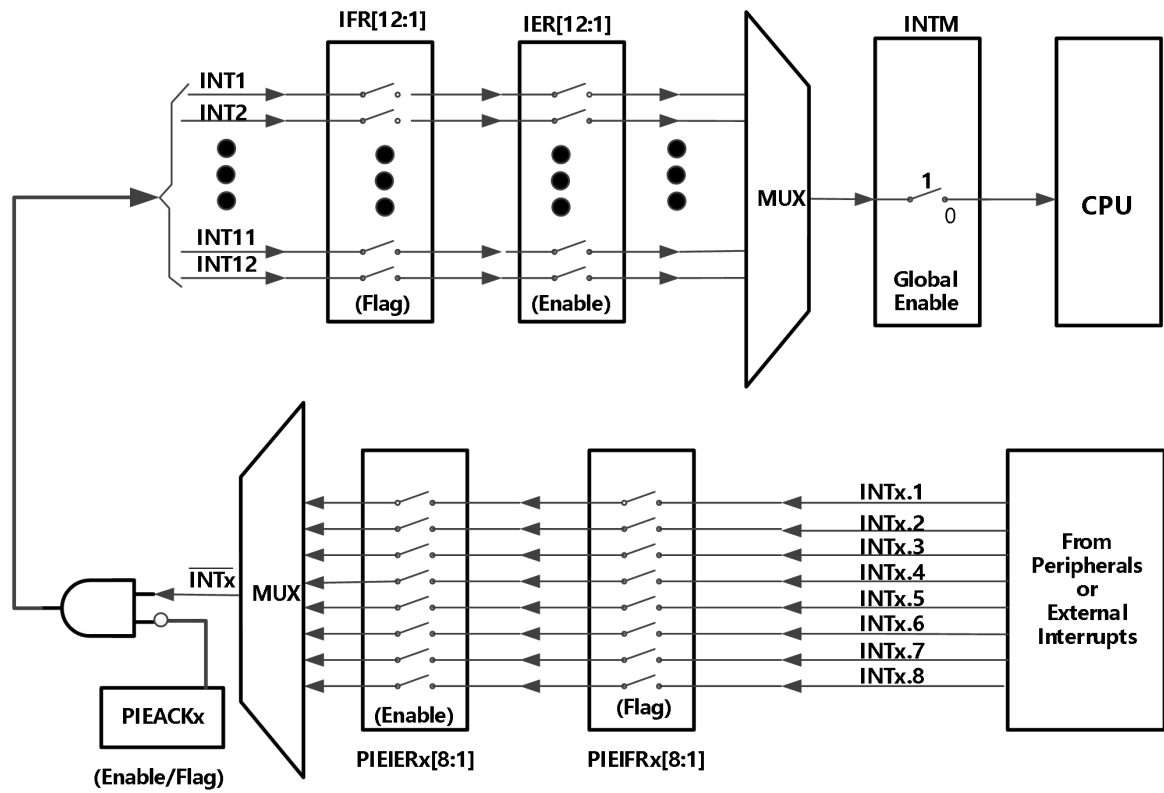


图 3-4 使用 PIE 块的中断复用

表 3-12 PIE 多路复用的外设中断向量表⁽¹⁾

	INTx.8	INTx.7	INTx.6	INTx.5	INTx.4	INTx.3	INTx.2	INTx.1
INT1.y	WAKEINT (LPM/WD) 0xD4E	TINT0 (定时器 0) 0xD4C	ADCINT9 (ADC) 0xD4A	XINT2 外部内部 2 0xD48	XINT1 外部内部 1 0xD46	保留 - 0xD44	ADCINT2 (ADC) 0xD42	ADCINT1 (ADC) 0xD40
INT2.y	保留 - 0xD5E	保留 - 0xD5C	保留 - 0xD5A	EPWM5_TZINT (ePWM5) 0xD58	EPWM4_TZINT (ePWM4) 0xD56	EPWM3_TZINT (ePWM3) 0xD54	EPWM2_TZINT (ePWM2) 0xD52	EPWM1_TZINT (ePWM1) 0xD50
INT3.y	保留 - 0xD6E	保留 - 0xD6C	保留 - 0xD6A	EPWM5_INT (ePWM5) 0xD68	EPWM4_INT (ePWM4) 0xD66	EPWM3_INT (ePWM3) 0xD64	EPWM2_INT (ePWM2) 0xD62	EPWM1_INT (ePWM1) 0xD60
INT4.y	保留 - 0xD7E	HRCAP1_INT (HRCAP1) 0xD7C	保留 - 0xD7A	保留 - 0xD78	SING_int 0xD76	DUL_PAR_int 0xD74	CANFD 0xD72	ECAP1_INT - 0xD70
INT5.y	保留 - 0xD8E	保留 - 0xD8C	保留 - 0xD8A	保留 - 0xD88	保留 - 0xD86	保留 - 0xD84	保留 - 0xD82	保留 - 0xD80
INT6.y	保留 - 0xD9E	保留 - 0xD9C	保留 - 0xD9A	保留 - 0xD98	SPITXINTB (SPI-B) 0xD96	SPIRXINTB (SPI-B) 0xD94	SPITXINTA (SPI-A) 0xD92	SPIRXINTA (SPI-A) 0xD90
INT7.y	保留 - 0xDAAE	保留 - 0xDAC	保留 - 0xDAA	保留 - 0xDA8	保留 - 0xDA6	保留 - 0xDA4	保留 - 0xDA2	保留 - 0xDA0
INT8.y	保留 - 0xDBE	保留 - 0DBC	保留 - 0DBA	保留 - 0xDB8	保留 - 0xDB6	CANFD 0xDB4	I2CINT2A (I2C-A) 0xDB2	I2CINT1A (I2C-A) 0xDB0
INT9.y	保留 - 0xDCE	保留 - 0DCC	ECAN1_INTA (CAN-A) 0xDCA	ECAN0_INTA (CAN-A) 0xDC8	LIN1_INTA (LIN-A) 0xDC6	LIN0_INTA (LIN-A) 0xDC4	SCITXINTA (SCI-A) 0xDC2	SCIRXINTA (SCI-A) 0xDC0
INT10.y	ADCINT8 (ADC) 0xDDE	ADCINT7 ADC 0DDC	ADCINT6 ADC 0DDA	ADCINT5 ADC 0DD8	ADCINT4 ADC 0DD6	ADCINT3 ADC 0DD4	ADCINT2 ADC 0DD2	ADCINT1 ADC 0DD0
INT11.y	CLA1_INT8 (CLA) 0xDEE	CLA1_INT7 (CLA) 0DEC	CLA1_INT6 (CLA) 0DEA	CLA1_INT5 (CLA) 0XDE8	CLA1_INT4 (CLA) 0xDE6	CLA1_INT3 (CLA) 0xDE4	CLA1_INT2 (CLA) 0xDE2	CLA1_INT1 (CLA) 0xDE0
INT12.y	LUF (CLA) 0DFE	LVF (CLA) 0DFC	保留 - 0DFA	保留 - 0xDF8	保留 - 0xDF6	保留 - 0xDF4	保留 - 0xDF2	XINT3 外部 内部 3 0xDF0

(1)在 96 个可能的中断中，部分中断未使用。这些中断是为将来的设备保留的。如果在 PIEIFRx 级别启用这些中断，则这些中断可以用作软件中断，前提是组内的任何中断均未被外设使用。否则，在修改 PIEIFR 时意外清除其标志，可能会丢失来自外设的中断。总之，在如下两种情况下，保留的中断可以用作软件中断：

- 组内所有外设均未触发中断请求。
- 该组没有被分配外设中断（例如，PIE 组 7）。

表 3-13 PIE 配置和控制寄存器

名称	地址	大小 (X 16)	说明 ⁽¹⁾
PIECTRL	0x 0CE0	1	PIE,控制寄存器
PIEACK	0x 0CE1	1	PIE,确认寄存器
PIEIER1	0x 0CE2	1	PIE,INT1 组启用寄存器
PIEIFR1	0x 0CE3	1	PIE,INT1 组标志寄存器
PIEIER2	0x 0CE4	1	PIE,INT2 组启用寄存器
PIEIFR2	0x 0CE5	1	PIE,INT2 组标志寄存器
PIEIER3	0x 0CE6	1	PIE,INT3 组启用寄存器
PIEIFR3	0x 0CE7	1	PIE,INT3 组标志寄存器
PIEIER4	0x 0CE8	1	PIE,INT4 组启用寄存器
PIEIFR4	0x 0CE9	1	PIE,INT4 组标志寄存器
PIEIER5	0x 0CEA	1	PIE,INT5 组启用寄存器
PIEIFR5	0x 0CEB	1	PIE,INT5 组标志寄存器
PIEIER6	0x 0CEC	1	PIE,INT6 组启用寄存器
PIEIFR6	0x 0CED	1	PIE,INT6 组标志寄存器
PIEIER7	0x 0CEE	1	PIE,INT7 组启用寄存器
PIEIFR7	0x 0CEF	1	PIE,INT7 组标志寄存器
PIEIER8	0x 0CF0	1	PIE,INT8 组启用寄存器
PIEIFR8	0x 0CF1	1	PIE,INT8 组标志寄存器
PIEIER9	0x 0CF2	1	PIE,INT9 组启用寄存器
PIEIFR9	0x 0CF3	1	PIE,INT9 组标志寄存器
PIEIER10	0x 0CF4	1	PIE,INT10 组启用寄存器
PIEIFR10	0x 0CF5	1	PIE,INT10 组标志寄存器
PIEIER11	0x 0CF6	1	PIE,INT11 组启用寄存器
PIEIFR11	0x 0CF7	1	PIE,INT11 组标志寄存器
PIEIER12	0x 0CF8	1	PIE,INT12 组启用寄存器
PIEIFR12	0x 0CF9	1	PIE,INT12 组标志寄存器
保留	0x 0CFA-0x 0CFF	6	保留

(1) PIE 配置和控制寄存器未受 EALLOW 模式保护。PIE 向量表受保护。

3.6.1 外部中断

表 3-14 外部中断寄存器

名称	地址	大小 (X 16)	说明
XINT1CR	0x00 7070	1	XINT1 配置寄存器
XINT2CR	0x00 7071	1	XINT2 配置寄存器
XINT3CR	0x00 7072	1	XINT3 配置寄存器
XINT1CTR	0x00 7078	1	XINT1 计数器寄存器
XINT2CTR	0x00 7079	1	XINT2 计数器寄存器
XINT3CTR	0x00 707A	1	XINT3 计数器寄存器

每个外部中断可由上升沿、下降沿或双沿启用、禁用或限定。

3.6.2 外部中断时序

表3-15 外部中断时序要求⁽¹⁾

参数	测试条件	最小值	最大值	单位
$t_{w(INT)}$ ⁽²⁾ 脉冲持续时间, INT 输入低电平/高电平的时间	同步	$1t_{c(SCO)}$		周期
	带输入限定器	$1t_{c(SCO)}+t_{w(IQSW)}$		周期

(1) 该时序适用于为 ADCSOC 功能性所配置的任一 GPIO 引脚。

表3-16外部中断开关特性

参数	最小值	最大值	单位
$T_{d(INT)}$ INT 低电平/高电平到中断向量提取的延迟时间		$t_{w(IQSW)} + 12t_{c(SCO)}$	周期

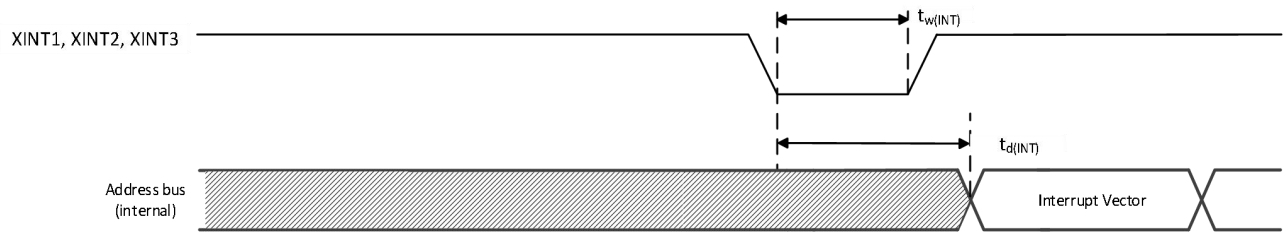


图3-5 外部中断时序

3.7 LDO /BOR/POR

数字内核、模拟外设和 I/O 电路运行在三个不同的电压上，内核电压 (V_{DD}) 和模拟电压 (V_{DDA}) 由片载线性稳压器 (LDO) 提供，这两个 LDO 的输入都来自 V_{DDIO} 电源。此外，在加电和运行模式期间，内部加电复位 (POR) 和欠压复位 (BOR) 电路监控 V_{DD} 、 V_{DDA} 和 V_{DDIO} 三个电源轨。

3.7.1 片载电压稳压器 (LDO)

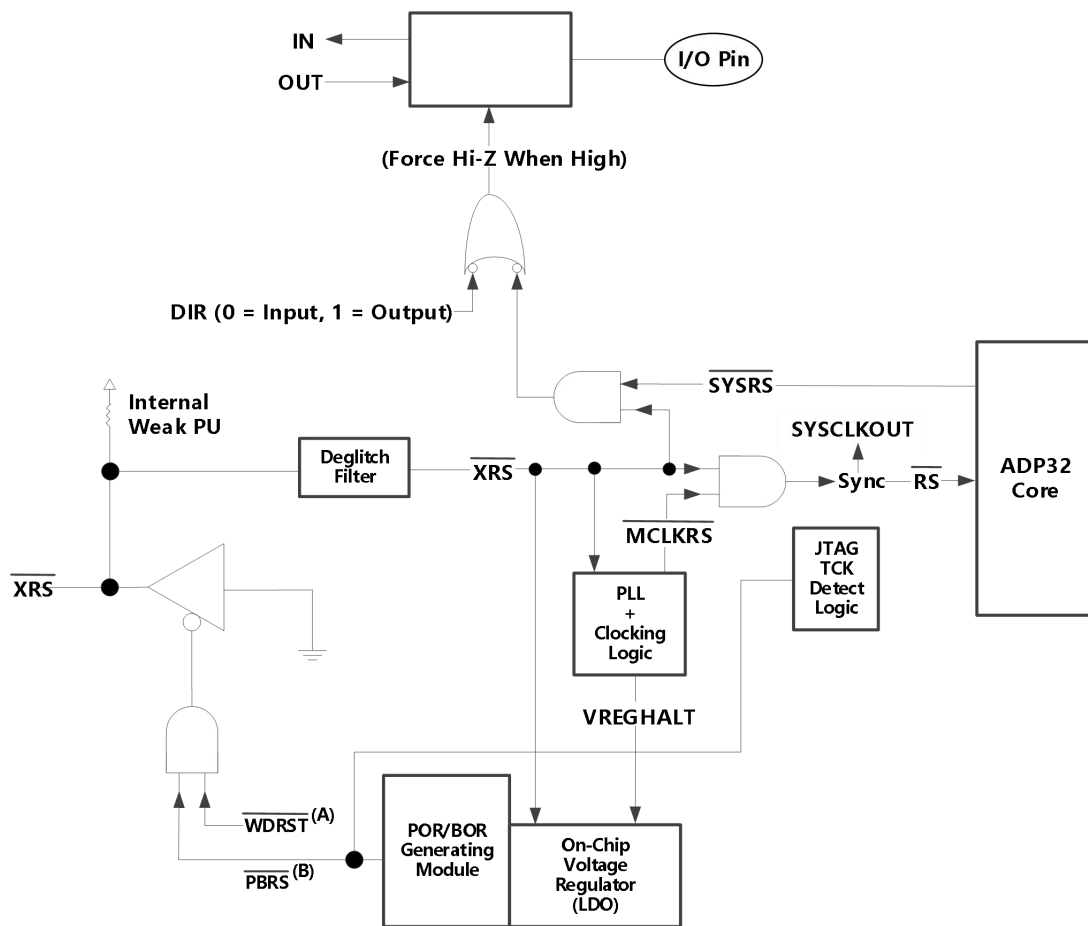
内核电压 (V_{DD}) 和模拟电压 (V_{DDA}) 由片载线性稳压器 (LDO) 提供，这两个 LDO 都由 V_{DDIO} 电源供电。因此， V_{DD} 和 V_{DDA} 引脚上都需要电容来稳定生成的电压，无需外部供电。

3.7.1.1 使用片载 LDO

片载 LDO 都是上电默认自动开启。在这种情况下，内核所需的 V_{DD} 电压和模拟电路所需的 V_{DDA} 电压都将由 LDO 生成。为 LDO 提供稳定的电压， V_{DD} 与 V_{SS} 之间应连接一组 $0.1\mu F$ 和 $2.2\mu F$ （典型值）的并联电容， V_{DDA} 与 V_{SSA} 之间脚应连接一组 $0.1\mu F$ 和 $4.7\mu F$ （典型值）的并联电容。这些电容应该被放置在尽可能接近 V_{DD} 、 V_{DDA} 引脚的位置。

3.7.2 片载上电复位 (POR) 和欠压复位 (BOR) 电路

本产品包含加电复位 (POR) 和欠压复位 (BOR) 两个电路，POR 监控 V_{DD} 、 V_{DDA} 和 V_{DDIO} 的上电过程，BOR 监控 V_{DD} 、 V_{DDA} 和 V_{DDIO} 的掉电过程。当其中任何一个电压低于各自的触发点时，都将使 XRS 引脚输出低电平。此时，整个芯片将被强制复位，GPIO 表现为高阻态。可以通过配置 BORCFG 寄存器禁用 BOR 功能。



A. WDRST是来自 CPU 看门狗的复位信号。

B. PBRS是来自 POR/BOR 模块的复位信号。

图 3-6 LDO + POR + BOR + Reset 信号连接图

3.8 系统控制

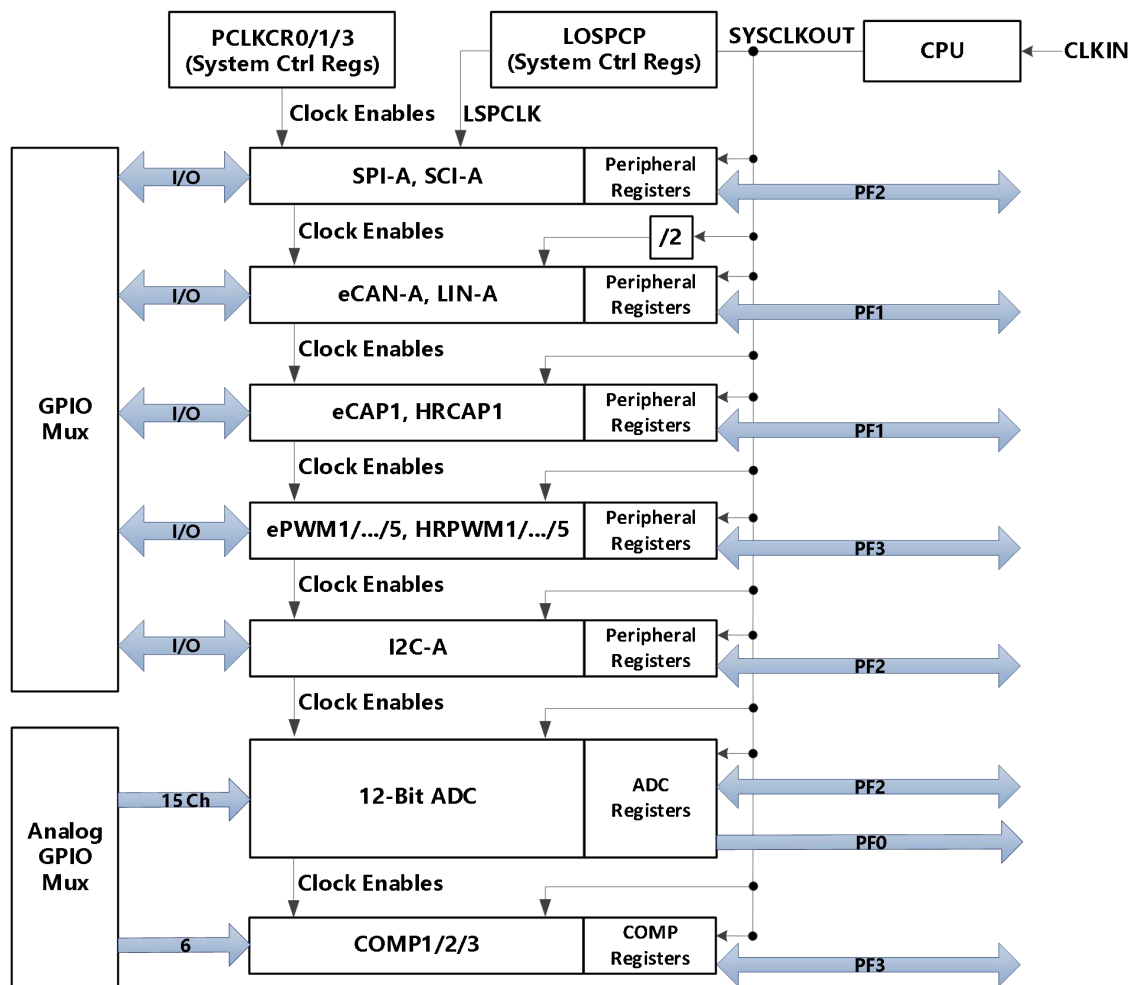
本节对振荡器和时钟机制、看门狗功能以及低功耗模式进行说明。

表 3-17 PLL、时钟、看门狗和低功耗模式寄存器

名称	地址	大小 (X 16)	说明 ⁽¹⁾
BORCFG	0x00 0985	1	BOR 配置寄存器
XCLK	0x00 7010	1	XCLKOUT 控制
PLLSTS	0x00 7011	1	PLL 状态寄存器
CLKCTL	0x00 7012	1	时钟控制寄存器
PLLLOCKPRD	0x00 7013	1	PLL 锁周期
INTOSC1TRIM	0x00 7014	1	内部振荡器 1 调整寄存器
INTOSC2TRIM	0x00 7016	1	内部振荡器 2 调整寄存器
LOSPCP	0x00 701B	1	低速外设时钟预分频寄存器
PCLKCR0	0x00 701C	1	外设时钟控制寄存器 0
PCLKCR1	0x00 701D	1	外设时钟控制寄存器 1
LPMCR0	0x00 701E	1	低功耗模式控制寄存器 0
PCLKCR3	0x00 7020	1	外设时钟控制寄存器 3
PLLCR	0x00 7021	1	PLL 控制寄存器
SCSR	0x00 7022	1	系统控制与状态寄存器
WDCNTR	0x00 7023	1	看门狗计数器寄存器
WDKEY	0x00 7025	1	看门狗复位密钥寄存器
WDCR	0x00 7029	1	看门狗控制寄存器

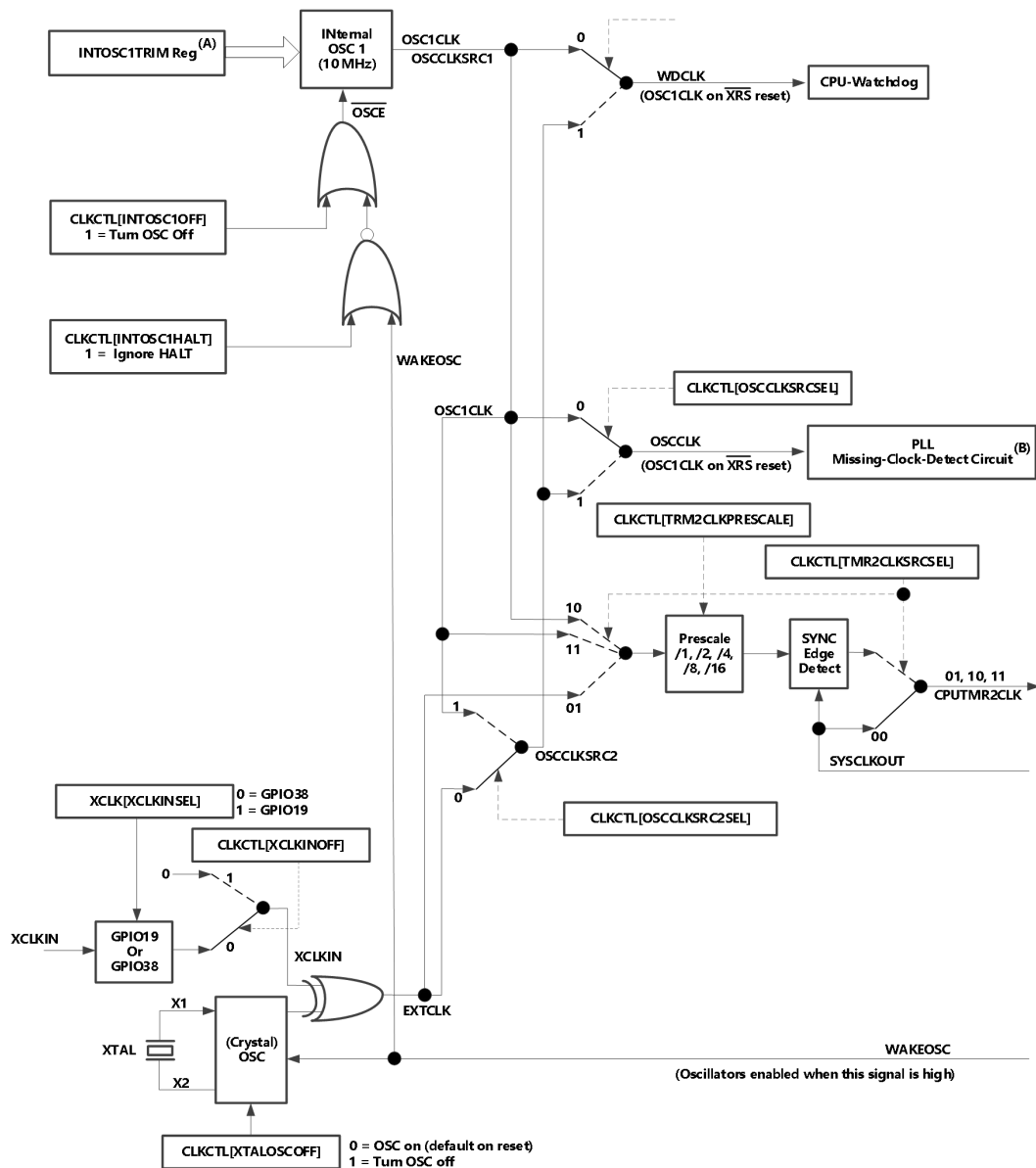
(1) 此表中的所有寄存器都是受 EALLOW 保护的。

图 3-7 所示为时钟和复位域。图 3-8 所示为供设备运行的不同的时钟源（包括内部的和外部的）。



A. CLKIN 为 CPU 输入时钟，SYSCLKOUT 为 CPU 输出时钟，与 CLKIN 频率相同。

图 3-7 时钟和复位域



A.从 OTP 的校准函数载入的寄存器。

图 3-8 时钟结构

3.8.1 内部零引脚振荡器

本产品包含一个独立的内部零引脚振荡器。默认情况下振荡器在上电时打开，此时内部振荡器 1 是默认时钟源。内部零引脚振荡器的中心频率由振荡器调整寄存器决定，并且在引导 ROM 执行过程中，会将相应的校准值写入这些寄存器，以进行振荡器的校准。有关振荡器的更多信息，请参阅第 5 章节 [“电气规范”](#)。

3.8.2 晶体振荡器选项

表 3-18 中列出了外部石英晶体（基本模式，并行共振）的典型技术规范。此外，ESR 范围 = 30 至 150Ω。

表 3-18 外部石英晶振的典型技术规范⁽¹⁾

频率 (MHz)	$R_d(\Omega)$	$C_{L1}(pF)$	$C_{L2}(pF)$
5	2200	18	18
10	470	15	15
15	0	15	15
20	0	12	12

(1) C 并联应该少于或者等于 5pF。

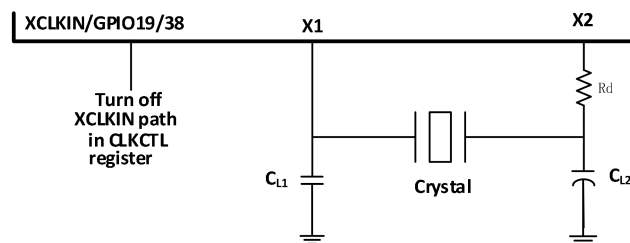


图 3-9 片载晶体振荡器的使用

注

1. C_{L1} 和 C_{L2} 是电路板和包括晶振在内的组件的总电容值。这个值通常约为晶振负载电容值的两倍。
2. 制造商的晶振技术规范中对晶振的负载电容值进行了说明。
3. 建议用户让晶振供应商对其器件与芯片的运行进行特性说明。供应商也可建议客户考虑适当的谐振组件值，这个值将在整个运行范围内实现适当的启动和稳定性。



图 3-10 3.3V/5V 外部振荡器的使用

3.8.3 PLL 时钟模块

ADP32F036 有一个片载 PLL 时钟模块。这个模块为器件提供所有需要的时钟信号，以及对进入低功耗模式的控制。PLL 有一个 4 位比率控制 PLLCR[DIV] 来选择不同的 CPU 时钟速率。在写入 PLLCR 寄存器之前，应确保看门狗处于禁止状态。在 PLL 模式稳定后，看门狗可被重新启用（如有需要），PLL 稳定过程典型延时为 1ms。输入时钟和 PLLCR[DIV] 位的配置条件应该确保 PLL (VCOCLK) 的输出频率不低于 50MHz 的时候再选择。

表 3-19 PLL 设置

PLLCR[DIV] 值 ^{(1) (2)}	SYSCLKOUT (CLKIN)		
	PLLSTS[DIVSEL]=0 或 1 ⁽³⁾	PLLSTS[DIVSEL]=2	PLLSTS[DIVSEL]=3
0000 (PLL 旁路)	OSCCLK/4 (默认) ⁽¹⁾	OSCCLK/2	OSCCLK
0001	(OSCCLK * 1)/4	(OSCCLK * 1)/2	(OSCCLK * 1)/1
0010	(OSCCLK * 2)/4	(OSCCLK * 2)/2	(OSCCLK * 2)/1
0011	(OSCCLK * 3)/4	(OSCCLK * 3)/2	(OSCCLK * 3)/1
0100	(OSCCLK * 4)/4	(OSCCLK * 4)/2	(OSCCLK * 4)/1
0101	(OSCCLK * 5)/4	(OSCCLK * 5)/2	(OSCCLK * 5)/1
0110	(OSCCLK * 6)/4	(OSCCLK * 6)/2	(OSCCLK * 6)/1
0111	(OSCCLK * 7)/4	(OSCCLK * 7)/2	(OSCCLK * 7)/1
1000	(OSCCLK * 8)/4	(OSCCLK * 8)/2	(OSCCLK * 8)/1
1001	(OSCCLK * 9)/4	(OSCCLK * 9)/2	(OSCCLK * 9)/1
1010	(OSCCLK * 10)/4	(OSCCLK * 10)/2	(OSCCLK * 10)/1
1011	(OSCCLK * 11)/4	(OSCCLK * 11)/2	(OSCCLK * 11)/1
1100	(OSCCLK * 12)/4	(OSCCLK * 12)/2	(OSCCLK * 12)/1

- (1) PLL 控制寄存器 (PLLCR) 和 PLL 状态寄存器 (PLLSTS) 只能通过 XRS 信号或者一个看门狗复位被复位为它们的默认值。调试器发出的复位或者丢失时钟检测逻辑对其没有影响。
- (2) 此寄存器受 EALLOW 保护。
- (3) 默认情况下，PLLSTS [DIVSEL] 被配置为 /4。（引导 ROM 将这个配置改为 /1。）在写入 PLLCR 前，PLLSTS[DIVSEL] 必须为 0，而只有当 PLLSTS[PLLOCKS]=1 时才应被改变。

表 3-20 CLKIN 分频选项

PLLSTS [DIVSEL]	CLKIN 分频
0	/4
1	/4
2	/2
3	/1

PLL 时钟模块有三种时钟源：

- **INTOSC（内部零引脚振荡器）：**内部片载振荡器。此振荡器可为看门狗、内核和 CPU 定时器 2 提供时钟。
- **晶体/振荡器运行：**片载（晶体）振荡器使得器件可以使用一个连接在其上的外部晶体/振荡器来提供时基。晶体/振荡器被连接至 X1/X2 引脚上。
- **外部时钟源运行：**如果片载（晶体）振荡器未被使用，此模式可实现对振荡器的旁通模式。此时器件时钟由一个外部时钟源生成并从 XCLKIN 引脚输入。请注意 XCLKIN 与 GPIO19 或者 GPIO38 引脚复用。通过 XCLK 寄存器中的 XCLKINSEL 位，XCLKIN 输入可选择为 GPIO19 或者 GPIO38 引脚。CLKCTL[XCLKINOFF] 位禁用该时钟输入（强制低电平）。
如果时钟源未被使用或者各自的引脚被用作 GPIO，用户应该在程序引导期间将其禁用。

在切换时钟源前，要确保目标切换时钟已稳定。如果时钟不存在，应在切换时钟源前禁用不存在的时钟源（使用 CLKCTL 寄存器）。

表 3-21 可能的 PLL 配置模式

PLL 模式	注释	PLLSTS[DIVSEL]	CLKIN 和 SYSCLKOUT
禁用 PLL	由 PLLSTS 寄存器中的 PLLOFF 位控制。在这个模式中，禁用 PLL 块。这对降低系统噪声和功耗非常有用。在进入此模式之前，必须先将 PLLCR 寄存器设置为 0x0000（PLL 旁路）。CPU 时钟 (CLKIN) 直接源自 X1/X2、X1 或者 XCLKIN 中任一个上的时钟输入。	0, 1	OSCCLK/4
		2	OSCCLK/2
		3	OSCCLK/1
旁路 PLL	PLL 旁路是上电或外部复位 $\overline{XRS}$ 时的默认 PLL 配置。当 PLLCR 寄存器设置为 0x0000 时或在 PLLCR 寄存器已经被修改之后，PLL 锁定至新频率时选择此模式。在此模式中 PLL 本身被旁路，但未关闭。	0, 1	OSCCLK/4
		2	OSCCLK/2
		3	OSCCLK/1
启用 PLL	通过将一个非零值“n”写入 PLLCR 寄存器实现。在写入 PLLCR 时，此器件将在 PLL 锁定之前切换至 PLL 旁路模式。	0, 1	OSCCLK * n/4
		2	OSCCLK * n/2
		3	OSCCLK * n/1

3.8.4 输入时钟丢失 (NMI 看门狗功能)

此器件可由 1 个内部零引脚振荡器 INTOSC1、片上晶体振荡器、或一个外部时钟输入提供时钟信号。无论时钟源是什么，在 PLL 启用和 PLL 旁路模式中，如果输入到 PLL 的输入时钟消失，PLL 将输出一个跛行模式时钟。该跛行模式时钟持续为 CPU 和外设提供一个典型值为 1-5MHz 的时钟。

当跛行模式激活时，生成一个被锁存为 NMI 中断的 $\overline{CLOCKFAIL}$ 信号。根据 NMIRESETSEL 位的配置方式，复位可能被立即触发或者当 NMI 看门狗计数器溢出时产生复位。除此之外，时钟丢失状态 (MCLKSTS) 位会被置位。用户程序可使用 NMI 中断来检测输入时钟故障并启动必要的纠正措施，例如切换到另一个时钟源（如果有的话）或者启动系统的关机程序。

如果软件对时钟故障情况没有响应，NMI 看门狗会在预编程的时间间隔后触发复位。

图 3-11 显示了相关的中断机制。

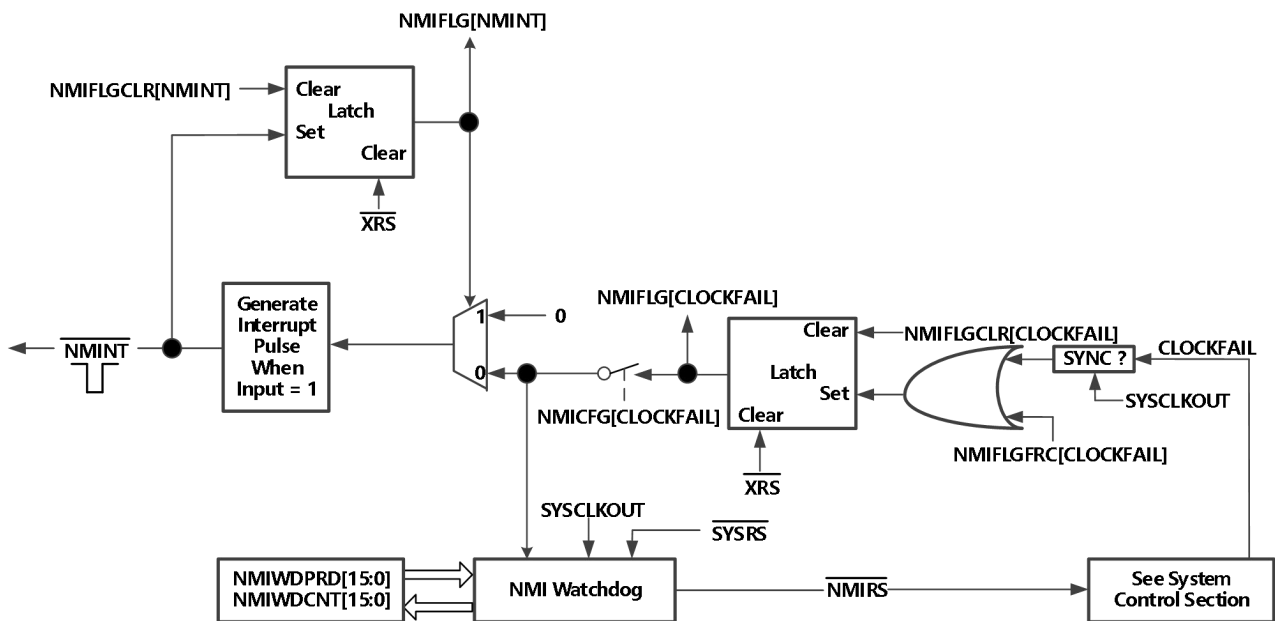


图 3-11 NMI - 看门狗

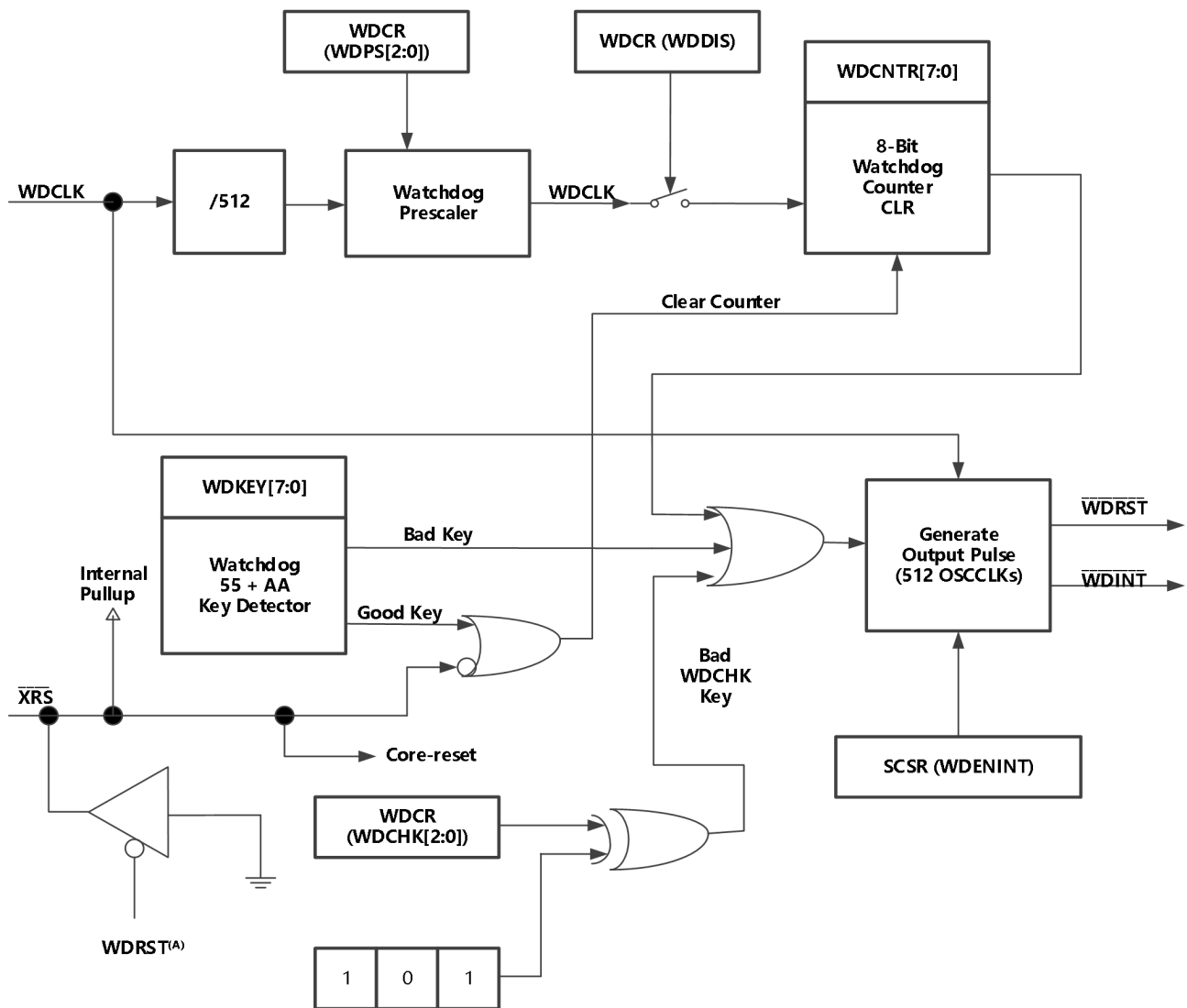
3.8.5 CPU 看门狗

当 8 位看门狗递增计数器达到其最大值时，该模块就会生成一个输出脉冲，即 512 个振荡器时钟宽度（OSCCLK）。为防止出现这种情况，用户必须禁用计数器，或者软件必须定期将 0x55 + 0xAA 序列写入看门狗 key 寄存器，以重置看门狗计数器。图 3-12 显示了看门狗模块中的各种功能块。

当外部输入时钟发生故障时，CPU 看门狗计数器停止递减（即看门狗计数器不会随跛行模式时钟而变化）。

注

为了确保 CPU 运行频率绝对正确，应该执行一个机制，通过这个机制，只要输入时钟出现故障，DSP 就被保持在复位状态。例如，如果电容器充满电，可以使用 R-C 电路来触发 DSP 的 XRS 引脚。I/O 引脚可用于定期对电容器进行放电，以防止其充满电。这种电路还有助于检测闪存的故障。



A. WDRST信号持续 512 个 OSCCLK 周期被驱动为低电平。

图 3-12 CPU 看门狗模块

WDINT信号使看门狗用作从 IDLE/STANDY 模式唤醒。

在 STANDBY 模式下，DSP 上的所有外设关闭。唯一保持正常运行的是 CPU 看门狗，该模块将在 OSCCLK 上运行。WDINT 信号输入至 LPM 模块，以便它可以将设备从 STANDBY（如果启用）中唤醒。详细信息请参阅第 3.9 节“低功耗模式”。

在 IDLE 模式下，WDINT信号可通过 PIE 向 CPU 生成一个中断，以使 CPU 退出 IDLE 模式。

在 HALT 模式下，CPU 看门狗可用于通过复位唤醒设备。

3.9 低功耗模式

表 3-22 总结了各种低功耗模式。

表 3-22 低功耗模式

模式	LPM(1:0)	OSCCLK	CLKIN	SYSCLKOUT	退出 ⁽¹⁾
IDLE	0,0	打开	打开	打开	$\overline{XRS}$, CPU 看门狗中断, 任何被启动的中断
STANDBY	0,1	打开 (CPU 看门狗仍然运行)	关闭	关闭	$\overline{XRS}$, CPU 看门狗中断, GPIO 端口 A 信号, 调试器 ⁽²⁾
HALT ⁽³⁾	1,X	关闭 (片载振荡器和 PLL 关闭, 零引脚振荡器和 CPU 看门狗状态取决于用户代码。)	关闭	关闭	$\overline{XRS}$, GPIO 端口 A 信号, 调试器 ⁽²⁾ , CPU 看门狗中断

(1) “退出”，列出哪些信号或在哪些情况下会退出低功耗模式，在这些信号中的任何一个低电平信号，将退出低功耗状态。此信号必须保持低电平足够长时间以便器件识别中断。否则，将不会从 IDLE 模式中退出，而器件将返回到已确认的低功耗模式。

(2) 内核时钟 (CLKIN) 被关闭，JTAG 仍然可以工作。

(3) WDCLK 必须处于活动状态，设备才能进入 HALT 模式。

三种不同的低功耗模式运行状态如下：

IDLE 模式 通过 CPU 识别使能的中断退出此模式。在此模式期间，在 LPMCR0 (LPM) 位被设定为 0,0 时，LPM 模块不执行任何任务。

STANDBY 模式 任意 GPIO 端口 A 信号 (GPIO[31:0]) 都可以将 CPU 从 STANDBY 模式唤醒。用户必须在 GPIOLPMSEL 寄存器中选择唤醒设备的信号源。在设备唤醒前，所选的信号源经 OSCCLK 限定，OSCCLK 的限定数量由 LPMCR0 寄存器设置。

HALT 模式 CPU 看门狗， $\overline{XRS}$ 和任意 GPIO 端口 A 信号 (GPIO[31:0])可将器件从 HALT 模式中唤醒。用户在 GPIOLPMSEL 寄存器中选择唤醒信号源。

低功耗模式并不会影响输出引脚的状态（包括 PWM 引脚在内）。当 IDLE 被指令执行时，它们将保持在代码指定的状态中。

4 外设

4.1 控制律加速器 (CLA)

CLA 通过增加并行处理能力实现对 CPU 功能的扩展。CLA 处理的时间敏感控制环路，可以缩短从 ADC 采样到输出的延迟。因此，CLA 可实现更快的系统响应和更高频率控制闭环。利用 CLA 执行时间敏感任务可以释放主 CPU 以同时执行其它任务。以下为 CLA 主要特性：

- 时钟频率与主 CPU 一致 (SYSCLKOUT)
- 独立的架构，CLA 能够独立于主 CPU 执行算法
 - 完整的总线架构：
 - ~ 程序地址总线和程序数据总线
 - ~ 数据地址总线、数据读总线和数据写总线
 - 独立的 8 级流水线
 - 12 位程序计数器 (MPC)
 - 4 个 32 位结果寄存器 (MR0-MR3)
 - 2 个 16 位辅助寄存器 (MAR0, MAR1)
 - 状态寄存器 (MSTF)
- 指令集包括：
 - IEEE 单精度 (32 位) 浮点数学运算
 - 具有并行加载或存储的浮点数学运算
 - 带有并行加法或者减法的浮点乘法
 - $1/X$ 和 $1/\sqrt{X}$ 估值
 - 数据类型转换。
 - 条件分支和调用
 - 数据加载或者存储操作
- CLA 程序代码最多包含 8 个任务或中断服务程序

- 每个任务的起始地址由 MVECT 寄存器指定;
 - 只要任务符合 CLA 程序空间, 对任务大小就有限制;
 - 每次执行一个任务直至完成, 不支持任务嵌套;
 - 任务完成后, 将在 PIE 内标识对应中断;
 - 当任务完成时, 将自动启动下一个优先级最高的待处理任务。
- CLA 任务触发机制:
 - CPU 通过 IACK 指令触发;
 - 任务 1 至 任务 7: 相应的 ADC 或 ePWM 中断。例如:
 - 任务 1: ADCINT1 或 EPWM1_INT
 - 任务 2: ADCINT2 或 EPWM2_INT
 - 任务 7: ADCINT7 或 EPWM7_INT (此器件不支持)
 - 任务 8: ADCINT8 或 CPU 定时器 0
 - 内存和共享外设:
 - CLA 和主 CPU 间的通信通过两个专用消息 RAM
 - 主 CPU 可将 CLA 程序和数据存储映射到 CPU 空间或者 CLA 空间
 - CLA 可直接访问 ADC 结果寄存器、比较器寄存器, 以及 ePWM 和 HRPWM 寄存器

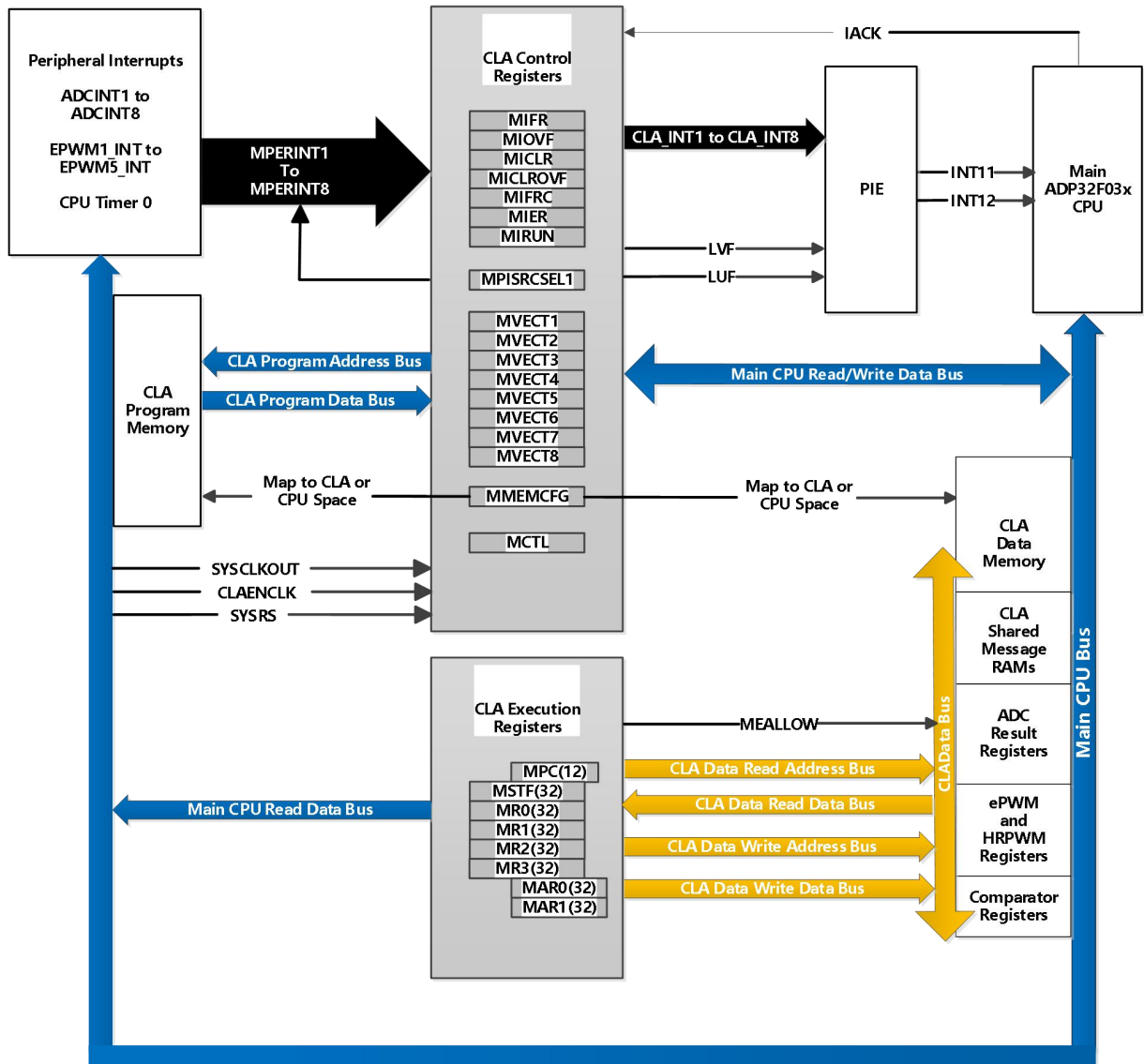


图 4-1 CLA 框图

表 4-1 CLA 控制寄存器

寄存器名	CLA1 地址	大小(x 16)	受 EALLOW 保护	说明 ⁽¹⁾
MVECT1	0x1400	1	支持	CLA 中断/任务 1 起始地址
MVECT2	0x1401	1	支持	CLA 中断/任务 2 起始地址
MVECT3	0x1402	1	支持	CLA 中断/任务 3 起始地址
MVECT4	0x1403	1	支持	CLA 中断/任务 4 起始地址
MVECT5	0x1404	1	支持	CLA 中断/任务 5 起始地址
MVECT6	0x1405	1	支持	CLA 中断/任务 6 起始地址
MVECT7	0x1406	1	支持	CLA 中断/任务 7 起始地址
MVECT8	0x1407	1	支持	CLA 中断/任务 8 起始地址
MCTL	0x1410	1	支持	CLA 控制寄存器
MMEMCFG	0x1411	1	支持	CLA 内存配置寄存器
MPISRCSEL1	0x1414	1	支持	外设中断源选择寄存器 1
MIFR	0x1420	1	支持	中断标志寄存器
MIOVF	0x1421	1	支持	中断溢出寄存器
MIFRC	0x1422	1	支持	中断强制寄存器
MICLR	0x1423	1	支持	中断清除寄存器
MICLROVF	0x1424	1	支持	中断溢出清除寄存器
MIER	0x1425	1	支持	中断使能寄存器
MIRUN	0x1426	1	支持	中断 RUN (运行) 寄存器
MIPCTL	0x1427	1	支持	中断优先级控制寄存器
MPC ⁽²⁾	0x1428	1	-	CLA 程序计数器
MAR0 ⁽²⁾	0x142A	1	-	CLA 辅助寄存器 0
MAR1 ⁽²⁾	0x142B	1	-	CLA 辅助寄存器 1
MSTF ⁽²⁾	0x142E	2	-	CLA STF 寄存器
MR0 ⁽²⁾	0x1430	2	-	CLA R0H 寄存器
MR1 ⁽²⁾	0x1434	2	-	CLA R1H 寄存器
MR2 ⁽²⁾	0x1438	2	-	CLA R2H 寄存器
MR3 ⁽²⁾	0x143C	2	-	CLA R3H 寄存器

(1) 这个表中的所有寄存器是受 CSM 保护的。

(2) CPU 对于这些寄存器只有用于调试目的的只读权限。CPU 不能执行到这些寄存器的 CPU 或者 DEBUG 写入。

表 4-2 CLA 消息 RAM

地址范围	大小(x 16)	说明
0x1480-0x14FF	128	CLA 到 CPU 消息 RAM
0x1500-0x157F	128	CPU 到 CLA 消息 RAM

4.2 模拟模块

图 4-2 显示了模拟模块的交互关系。

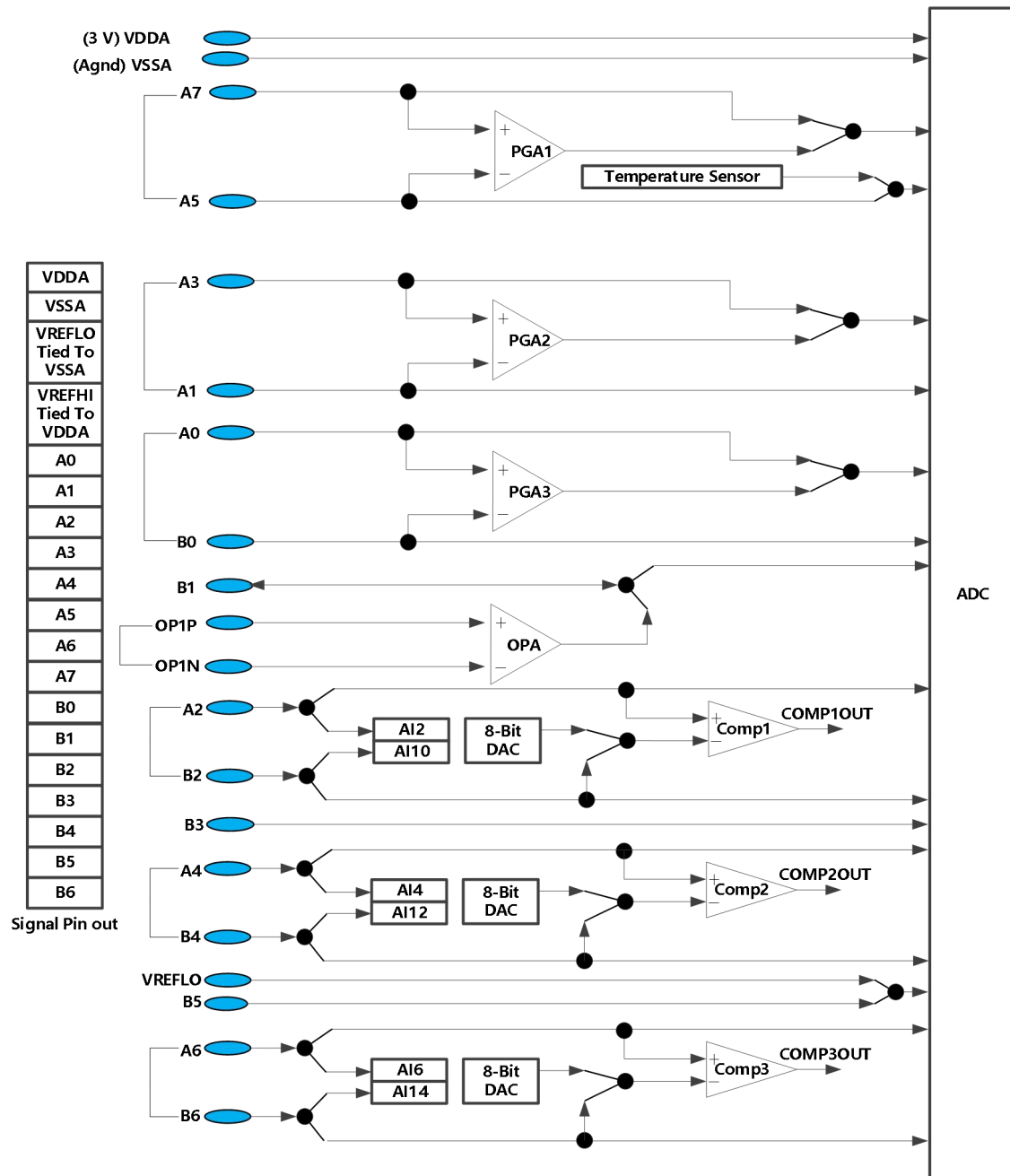


图 4-2 模拟引脚配置

4.2.1 ADC

4.2.1.1 特性

此器件内部有一个 12 位 SAR ADC 模块。ADC 内核包含一个 1 个采样保持电路和 12 位模数转换器。此 ADC 并不是基于排序器的。用户可以通过 SOC（单独转换的配置为中心）一次触发实现一系列转换。

ADC 模块的功能包括：

- 12 位 ADC 内核，包含 1 个采样保持 (S/H) 电路
- 顺序采样模式
- 全范围模拟输入：0V 至 3V 固定值。输入模拟电压的数字值由以下公式得出：

– 内部基准电压源 ($V_{REFLO}=V_{SSA}$ 。)

(1) 当 $input \leq 0V$

$$\text{Digital Value} = 0$$

(2) 当 $0V \leq input \leq 3V$

$$\text{Digital Value} = 4096 \times \frac{\text{Input Analog Voltage} - V_{REFLO}}{3}$$

(3) 当 $input \geq 3V$

$$\text{Digital Value} = 4095$$

- ADC 时钟 $\leq 60\text{MHz}$;
- **12 位或 6 位 ADC 模式可切换 (通过 ADCCTL2 的 BIT1 控制, BIT1=0 选择 6 位 ADC, BIT1=1 选择 12 位 ADC 模式, 上电复位默认选择 6 位 ADC 模式)**
- 多达 15 个复用的输入通道
- 16 个 SOC, 可针对触发、采样窗口和通道进行配置
- 用于存储转换值的 16 个结果寄存器 (可单独寻址)
- 多个触发源
 - S/W - 软件立即启动
 - ePWM 1-5

- GPIO XINT2
- CPU 定时器 0/1/2
- ADCINT1/2
- 9 个 PIE 中断，可配置在任意转换后申请中断

表 4-3 ADC 配置和控制寄存器

寄存器名称	地址	大小 (x16)	受 EALLOW 保护	说明
ADCCTRL1	0x7100	1	支持	控制寄存器 1
ADCCTRL2	0x7101	1	支持	控制寄存器 2
ADCINTFLG	0x7104	1	否	中断标志寄存器
ADCINTFLGCLR	0x7105	1	否	中断标志清除寄存器
ADCINTOVF	0x7106	1	否	中断溢出寄存器
ADCINTOVFCLR	0x7107	1	否	中断溢出清除寄存器
INTSEL1N2	0x7108	1	支持	中断 1 和 2 选择寄存器
INTSEL3N4	0x7109	1	支持	中断 3 和 4 选择寄存器
INTSEL5N6	0x710A	1	支持	中断 5 和 6 选择寄存器
INTSEL7N8	0x710B	1	支持	中断 7 和 8 选择寄存器
INTSEL9N10	0x710C	1	支持	中断 9 选择寄存器 (被保留的中断 10 选择)
SOCPRCTL	0x7110	1	支持	SOC 优先级控制寄存器
ADCSAMPLEMODE	0x7112	1	支持	采样模式寄存器
ADCINTSOCSEL1	0x7114	1	支持	中断 SOC 选择寄存器 1 (用于 8 个通道)
ADCINTSOCSEL2	0x7115	1	支持	中断 SOC 选择寄存器 2 (用于 8 个通道)
ADCSOCFLG1	0x7118	1	否	SOC 标志寄存器 1 (用于 16 个通道)
ADCSOCFRC1	0x711A	1	否	SOC 强制寄存器 1 (用于 16 个通道)
ADCSOCOVF1	0x711C	1	否	SOC 溢出寄存器 1 (用于 16 个通道)
ADCSOCOVFCLR1	0x711E	1	否	SOC 溢出清除 1 寄存器 1 (用于 16 个通道)
ADCSOC0CTL 至 DCSOC15CTL	0x7120 - 0x712F	1	支持	SOC0 控制寄存器至 SOC15 控制寄存器
ADCREFTTRIM	0x7140	1	支持	参考修调寄存器
ADCOFFTRIM	0x7141	1	支持	失调修调寄存器
COMPHYSTCTL	0x714C	1	支持	比较器迟滞控制寄存器
ADCREV	0x714F	1	否	修订版本寄存器

表 4-4 ADC 结果寄存器 (被映射至 PF0)

寄存器名称	地址	大小 (x16)	受 EALLOW 保护	说明
ADCRESULT0 至 ADCRESULT15	0xB00 - 0xB0F	1	否	ADC 结果 0 寄存器至 ADC 结果 15 寄存器

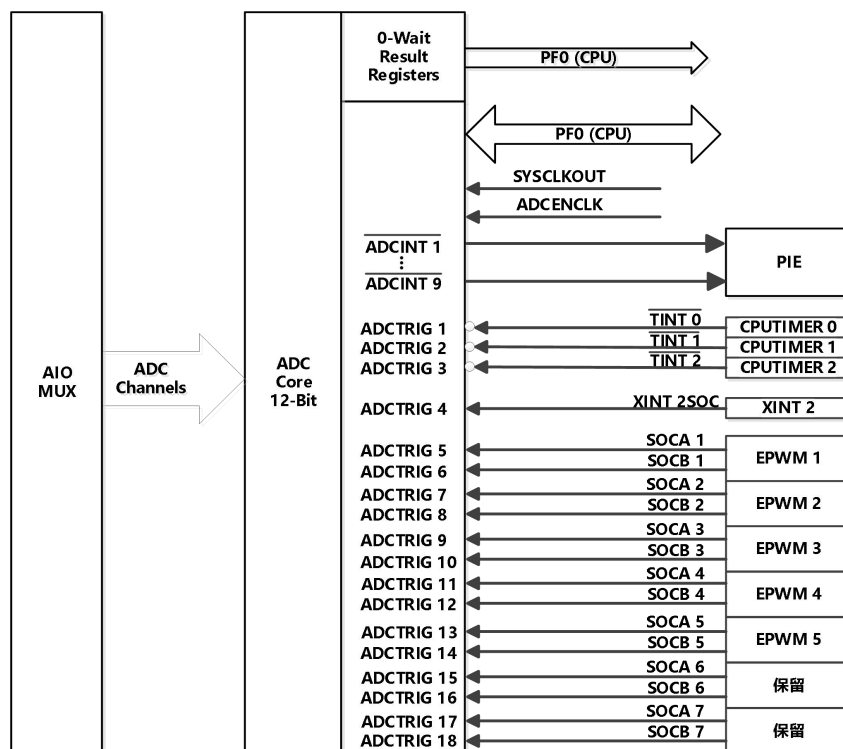


图 4-3 ADC 连接

注意：即使不使用 ADC，也应保持模拟电源引脚的连接。

ADC 未在应用中使用，ADC 引脚应按如下方式进行连接：

- V_{SSA} -连接到 V_{SS}
- A_n , B_n 连接到 V_{SSA}

在应用中使用 ADC 模块时，未使用的 ADC 输入引脚应连接到模拟地 (V_{SSA})。

注意：

与 AI 功能复用的未使用的 ADCIN 引脚不应直接连接到模拟地。它们应该通过一个 1k Ω 电阻接地。这是为了防止一个错误代码将这些引脚配置为 AI 输出并将接地的引脚驱动到逻辑高电平状态。

当未使用 ADC 时，确保关闭 ADC 模块的时钟降低功耗。

4.2.1.2 ADC 转换启动时序

表4-5 外部 ADC 转换启动开关特性

参数	最小值	最大值	单位
$t_{w(ADCSOCL)}$ 脉冲持续时间, $ADCSOCx0$ 低电平脉冲持续时间	32 $t_{c(HCO)}$		周期

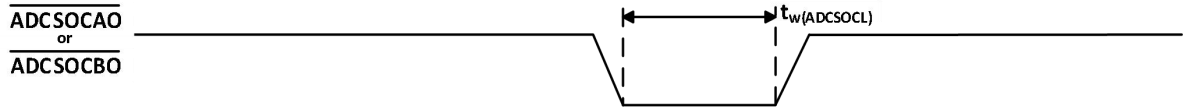


图4-4 ADCSOCAO 或ADCSOCBO 时序

4.2.1.3 片载模数转换器

表4-6 ADC电气特性⁽¹⁾

参数		最小值	典型值	最大值	单位
DC 技术规范					
分辨率		12			位
ADC 时钟		0.001		60	MHZ
采样窗口	ADP32F036	1		64	ADC 时钟
精度					
在 ADC 时钟 $\leq 60\text{MHz}^{(2)}$ 时, INL (积分非线性)		-7		7	最低有效位(LSB)
在 ADC 时钟 $\leq 60\text{MHz}$ 时, DNL (微分非线性)		-2		2	LSB
偏移误差 ⁽³⁾	执行一个单次自我校准	-20	0	20	LSB
	执行定期自我校准 ⁽⁴⁾	-8	0	8	
带有内部基准的总增益误差		-60		60	LSB
通道到通道偏移变化		-8		8	LSB
通道到通道增益变化		-8		8	LSB
带有内部基准的 ADC 温度系数			-50		ppm/°C
模拟输入					
带有内部基准的模拟输入电压		0		3	V
V _{REFLO} 输入电压 ⁽⁴⁾		-5	0	5	mV
输入电容值	模拟输入引脚的典型负载电容	不采样	5		pF
		采样	8.2		pF
输入漏电流			± 2		μA
AC 技术规范					
SINAD (28kHz) 信噪比+失真			62		dB
SNR (28kHz) 信噪比			62		dB
THD (28kHz) 总谐波失真			-68		dB
ENOB (28kHz) 有效位数			10.1		Bits
SFDR (28kHz) 无杂散动态范围			69		dB

- (1) 当 ADC 输入电压上升到高于 V_{DDA} 时, INL 将降级。
- (2) 1 LSB 为满刻度范围 (FSR)/4096 的加权值。FSR 为带有内部基准的 3V。
- (3) 定期自我校准能够消除系统级或温度对 ADC 零偏移误差的依赖, 这可在应用方案中根据需要去执行, 而不需要牺牲一个 ADC 通道。
- (4) V_{REFLO} 为输入信号的参考地, 输入信号与 V_{REFLO} 之间需增加 RC 滤波电路。

表4-7 ADC功率模式

ADC 运行模式	条件	I _{DDA}	单位
模式 A - 运行, 模式	ADC 时钟启用 ADC 上电 (ADCPWDN=1)	3.4	mA
模式 B - 关闭模式	ADC 时钟启用 ADC 断电 (ADCPWDN=0)	0.01	mA

4.2.1.3.1 内部温度传感器

内部温度传感器用于测量设备的结温。传感器的输出可以通过与 ADC 内部连接进行采样, 并通过软件转换为温度。温度传感器的传输函数和传输曲线如图 4-5 所示:

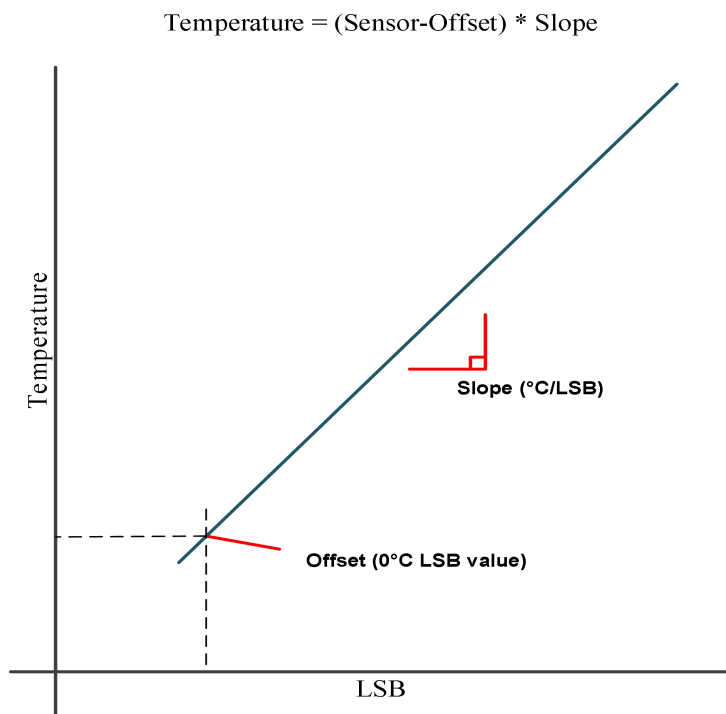


图4-5 温度传感器输入输出的传输曲线图

表4-8 温度传感器系数

参数 ⁽¹⁾	最小值	典型值	最大值	单位
T _{SLOPE} 温度按照温度传感器测得的 ADC LSB 变化而变动		0.181 ⁽²⁾⁽³⁾		°C/LSB
T _{OFFSET} 在温度传感器读数为 0°C 时的 ADC 输出		1511		LSB

- (1) 温度传感器斜坡和偏移根据使用ADC内部基准的ADC LSB指定。必须按照外部基准电压调整外部基准模式中的电压值。
- (2) ADC 温度系数被归入该技术规范。
- (3) 温度传感器的输出（以 LSB形式）与温度变化量的方向一致。上升的温度将使得ADC值相对于初始值增加，温度的下降将使得ADC的值相对于初始值下降。

4.2.1.3.2 ADC 上电控制位时序

表4-9 ADC上电延迟

参数 ⁽¹⁾	最小值	典型值	最大值	单位
t _{d(PWD)} 上电后，ADC 保持稳定的延迟时间			1	ms
VCCA 的 LDO 上电稳定所需的延迟时间			5	ms

- (1) 保持与 ADC 模块时序的兼容性。在首次转换前的 t_{d(PWD)}ms，F036 ADC 支持同时驱动所有 3 个位。

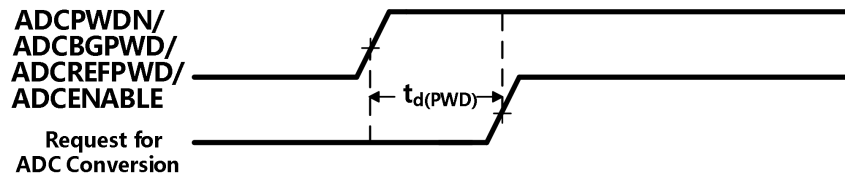
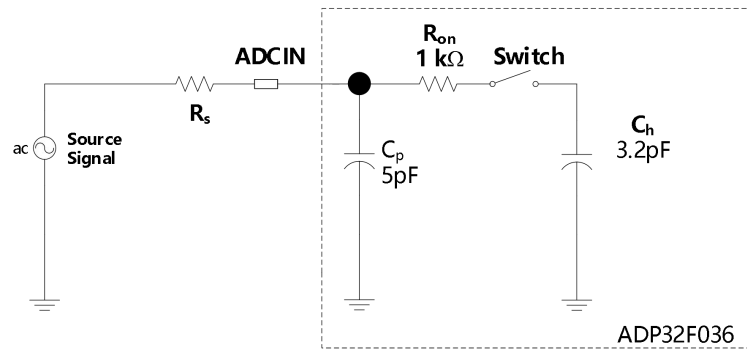


图4-6 ADC转换时序



Typical Values of the Input Circuit Components:

Switch Resistance (R_{on}): 1 kΩ

Sampling Capacitor (C_h): 3.2 pF

Parasitic Capacitance (C_p): 5 pF

Source Resistance (R_s): 50 Ω

图 4-7 ADC 输入阻抗模型

4.2.1.3.3 ADC 顺序采样模式时序

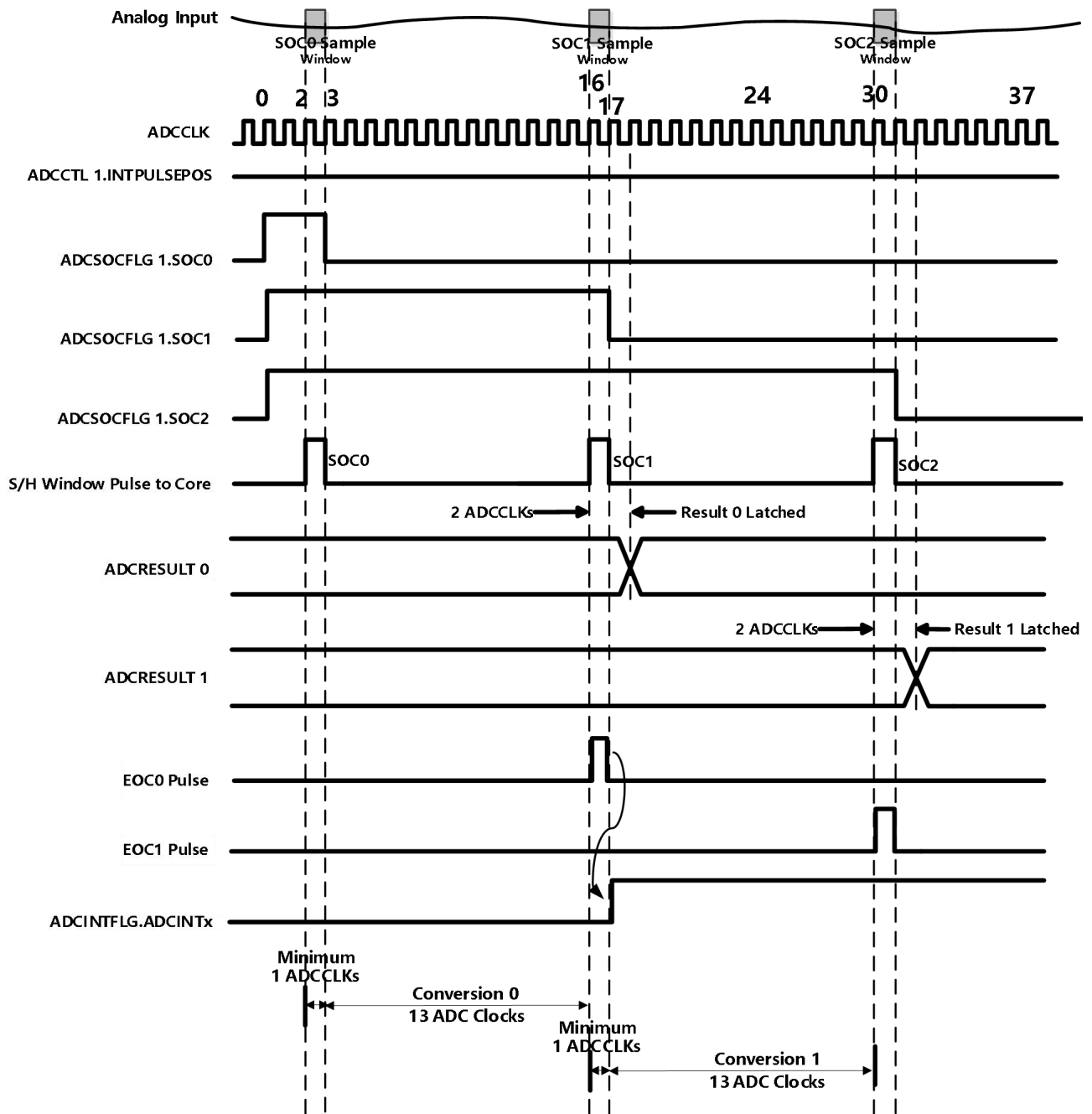


图4-8 针对顺序模式/延迟中断脉冲的时序示例

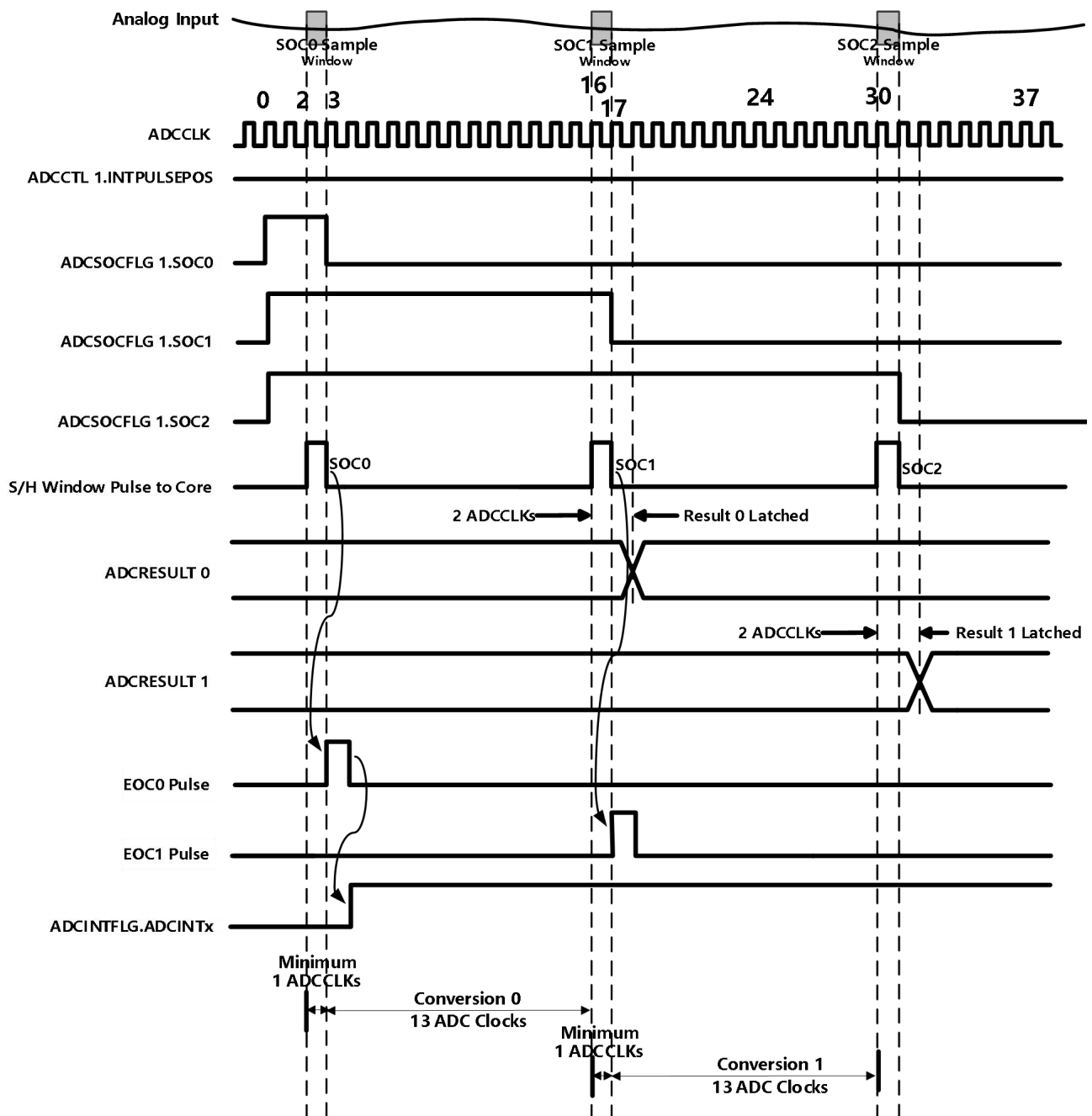


图4-9 针对顺序模式/提前中断脉冲的时序示例

4.2.3 比较器模块

图 4-11 为比较器模块系统图。

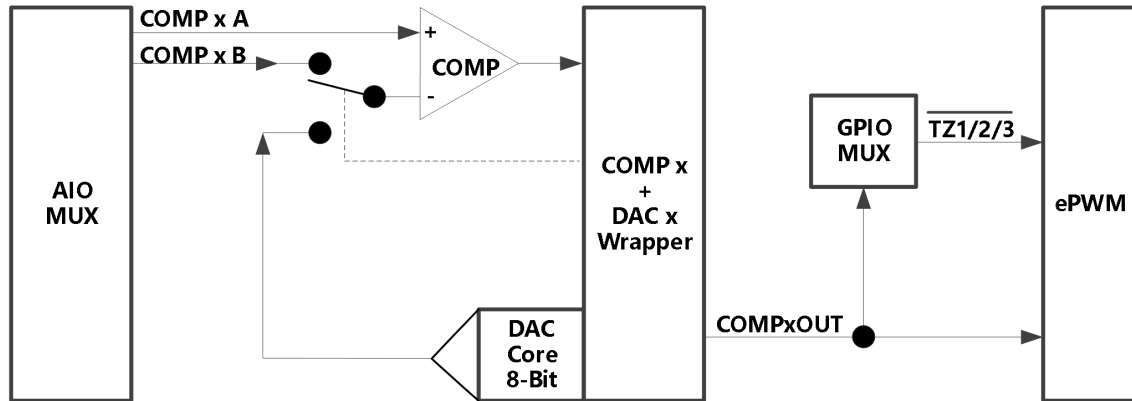
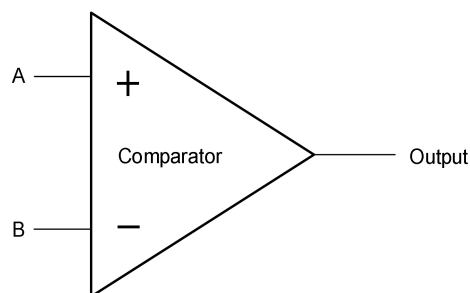


图 4-11 比较器模块图

表 4-10 比较器控制寄存器

寄存器名称	COMP1 地址	COMP2 地址	COMP3 地址	大小 (x 16)	受 EALLOW 保护	说明
COMPCTL	0x6400	0x6420	0x6440	1	支持	比较器控制寄存器
COMPSTS	0x6402	0x6422	0x6442	1	否	比较器状态寄存器
DACCTL	0x6404	0x6424	0x6444	1	支持	DAC 控制寄存器
DACVAL	0x6406	0x6426	0x6446	1	否	DAC 值寄存器
RAMPMAXREF_ACTIVE	0x6408	0x6428	0x6448	1	否	斜坡发生器最大基准（有效）寄存器
RAMPMAXREF_SHDW	0x640A	0x642A	0x644A	1	否	斜坡发生器最大基准（阴影）寄存器
RAMPDECVAL_ACTIVE	0x640C	0x642C	0x644C	1	否	斜坡发生器减量值（有效）寄存器
RAMPDECVAL_SHDW	0x640E	0x642E	0x644E	1	否	斜坡发生器减量值（阴影）寄存器
RAMPSTS	0x6410	0x6430	0x6450	1	否	斜坡发生器状态寄存器

每个 CMPSS 模块内包含 1 个比较器单元，比较器可实现功能如图 4-12 所示。可比较 A，B 两端口的模拟电压，输出对应的数字逻辑信号。



Voltages	Output
Voltage A > Voltage B	1
Voltage A < Voltage B	0

图 4-12 比较器

比较器自带 8bit 的 DAC，可替代负端输入，作为比较器参考电压。DAC 输出电压计算公式为：

$$DACOUT = \frac{DACVAL * (VDDA - VSSA)}{256} \quad (\text{式 2-1})$$

4.2.3.1 片载比较器 / DAC 电气数据和时序

表4-11 比较器 / DAC 的电气特性

特性	最小值	典型值	最大值	单位
比较器				
比较器输入范围		$V_{SSA} - V_{DDA}$		V
比较器到 PWM 可编程控制故障区的响应时间（同步）		50		ns
输入偏移		± 5		mV
输入迟滞 ⁽¹⁾		35		mV
DAC				
DAC 输出范围		$V_{SSA} - V_{DDA}$		V
DAC 分辨率		8		位
DAC 稳定时间		请参考图 4-13		
DAC 增益		-1.5%		
DAC 偏移		10		mV
单调性		支持		
INL		± 3		LSB

(1). 比较器输入端的迟滞是通过施密特触发器结构实现的。这将在比较器的输出和比较器非反向输入间实际产生一个100kΩ电阻值，有一个禁用迟滞和反馈电阻的选项；

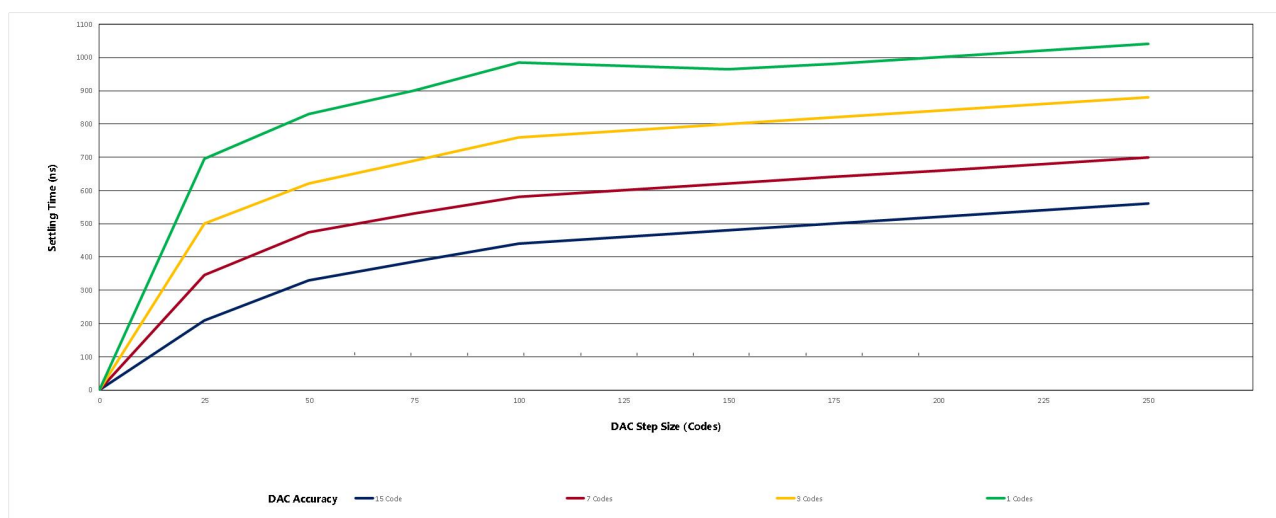


图4-13 DAC 稳定时间

4.2.4 可编程增益放大器 (PGA)、运算放大器 (OPA)

可编程增益放大器 (PGA) 和运算放大器 (OPA) 用于放大输入电压，以提高下游 ADC 模块的有效分辨率。

对于需要外部独立放大器的许多传统控制应用，集成的 PGA、OPA 有助于降低其成本和设计工作量。通过片载集成可确保 PGA、OPA 与下游 ADC 模块兼容。软件可选增益设置使 PGA 能够满足各种性能需求。OPA 可根据用户需求灵活定义放大倍数及输出共模电压。

使用 PGA 和 OPA 时需要注意，由于实际设计电阻值并非无穷大，因此需要注意板级寄生电阻对 PGA、OPA 放大倍数的影响。

该 PGA 具有以下特性：

- 四种可编程增益模式：2 倍、4 倍、8 倍、10 倍、16 倍
- 由 V_{DDA} 和 V_{SSA} 进行内部供电
- 支持使用 PGA_GND 引脚进行开尔文接地连接

PGA 中的有源器件是一个嵌入式运算放大器，通过内部反馈电阻配置为同相放大器。这些内部反馈电阻值经过配对以产生软件可选的电压增益。

器件引脚上有三个 PGA 信号：

- PGA_IN 是 PGA 运算放大器的正输入。施加到该引脚的信号将被 PGA 放大。
- PGA_GND 是 PGA_IN 信号的开尔文接地基准。理想情况下，PGA_GND 基准等于 V_{SSA} ；但是，PGA 可以容忍 V_{SSA} 的小失调电压。
- PGA_OUT 是运算放大器输出端的内部信号。它可由内部 ADC 模块进行采样和监控。图 4-14 所示为 PGA 方框图。

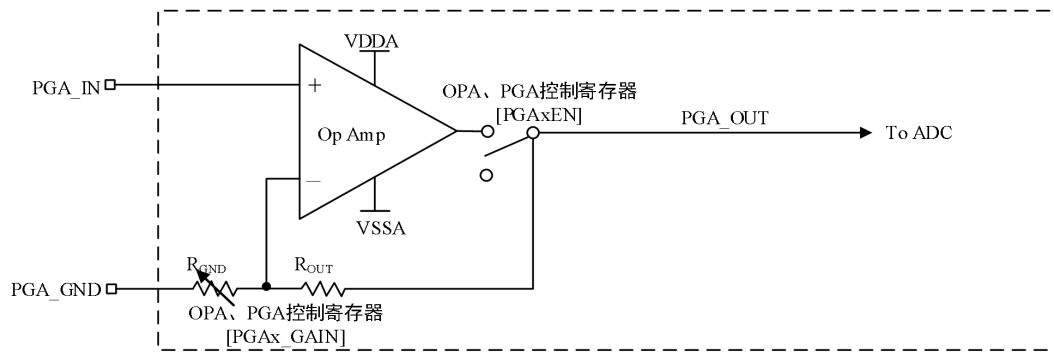


图 4-14 PGA 框图

使用 PGA 放大信号前需设置好 PGA 输出共模 V_{COM_OUT} ， V_{COM_OUT} 计算公式为，假定 VP 与 VN 分别为 PGA_IN 端与 PGA_GND 端输入共模电压值，则有：

$$V_{COM_OUT} = [(V_P - V_N)/R_{GND}] * R_{OUT} + V_P$$

PGA 同相放大倍数 AVP 计算公式为：

$$AVP = (R_{OUT} + R_{GND})/R_{GND}$$

当配置寄存器 0x7030h 改变 PGA 增益配置时，通过改变 R_{GND} 电阻值，从而改变 PGA 的放大倍数。

R_{GND} 与 R_{OUT} 在 PGA 不同放大倍数下近似电阻值如表 4-12 所示。

表4-12 PGA内阻 R_{GND} 与 R_{OUT} 数值

PGA同相放大倍数	R_{GND} 电阻值 (k Ω)	R_{OUT} 电阻值 (k Ω)
2	50	50
4	50/3	50
8	50/7	50
10	50/9	50
16	50/15	50

4.2.4.1 PGA 电气数据和时序

在建议运行条件下测得（除非另有说明）

参数	测试条件	最小值	典型值	最大值	单位
PGA 输出范围 ⁽¹⁾		VSSA + 0.2		VDDA - 0.2	V
PGA GND 范围		-50		200	mV
最小 ADC S+H (增益 = 2、4、8、10)	稳定在 ± 1 ADC LSB 精度范围内	160			ns
最小 ADC S+H (增益 = 16)	稳定在 ± 2 ADC LSB 精度范围内	200			ns

(1) 这是PGA的线性输出范围。PGA可以输出此范围以外的电压，但电压将不呈线性。

4.2.4.2 PGA 特征

在建议运行条件下测得（除非另有说明）

参数	测试条件	最小值	典型值	最大值	单位
通用					
增益设置		2、4、8、10、16			
输入偏置电流		2			nA
短路电流		10			mA
满量程阶跃响应	稳定在 ±2 ADC LSB（12 位）精度范 围内	450			ns
建立时间	增益开关	10			μs
压摆率		42		76	V/μs
上电时间		500			μs
直流特征 ⁽⁵⁾					
增益误差 ⁽¹⁾	增益 = 2、4、8、10	-1		1	%
	增益：16	-1.5		1.5	%
增益温度系数		±0.004			%/C
失调误差 ⁽²⁾	以输入为基准	-1.5		1.5	mV
失调温度系数	以输入为基准	±5.5			μV/C
直流代码扩展		2.5			12b LSB
AC 特征					
带宽 ⁽³⁾	增益 = 2、4、8、10	10			MHz
	增益 = 16	7			MHz
THD ⁽⁴⁾	直流	-78			dB
	高达 100 kHz	-70			dB
CMRR	直流	-60			dB
	高达 100 kHz	-50			dB
PSRR ⁽⁴⁾	直流	-70			dB
	高达 100 kHz	-60			dB
噪声 PSD ⁽⁴⁾	1kHz	200			nV/sqrt(Hz)
集成噪声 (以输入为基准) ⁽⁴⁾	3 Hz 至 30 MHz	100			μV

(1) 包括外部基准模式下的 ADC 增益误差。

(2) 包括外部基准模式下的 ADC 失调误差。

(3) 3dB 带宽。

(4) PGA 单独的性能。

(5) PGA 的 DNL/INL 在 ADC 的 DNL/INL 容差范围内，因此未单独显示。

4.2.4.3 PGA 典型特征图

图 4-15 所示为输入偏置电流与温度间的关系。

备注 对于图 4-15，需满足以下条件（除非另有说明）：

- $T_A = 30^\circ\text{C}$
- $V_{DDA} = 3\text{V}$
- $V_{DD} = 1.2\text{V}$

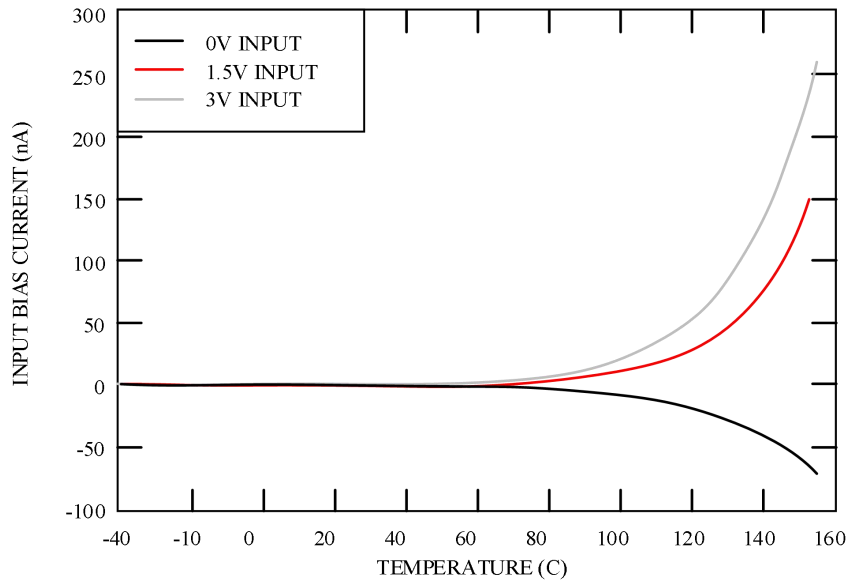


图 4-15 PGA 输入偏置电流与温度间的关系

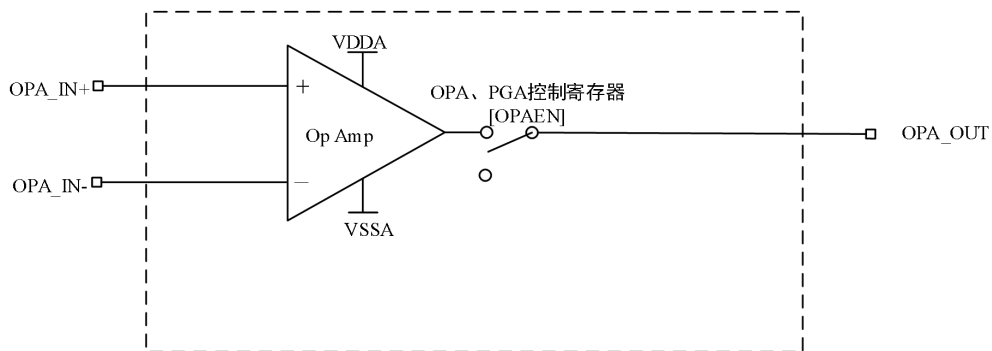


图 4-16 OPA 框图

芯片内除了3个带内部反馈回路的PGA外，还内置一个3端口均引出的独立OPA，结构图如图4-16所示，该OPA通过OP1P（接OPA_IN+ 端）、OP1N（接OPA_IN- 端）引脚输入。通过B1_OP1OUT（接OPA_OUT 端）引脚输出，该端口同时为ADC的B1通道。

4.2.4.4 OPA 电气数据和时序

在建议运行条件下测得（除非另有说明）

参数	测试条件	最小值	典型值	最大值	单位
OPA 输出范围 ⁽¹⁾		VSSA + 0.2		VDDA - 0.2	V
OPA_IN- 范围		-50		200	mV
最小 ADC S+H (外部电路实现增益 = 2、4、8、10)	稳定在 ± 1 ADC LSB 精度范围内	160			ns
最小 ADC S+H (外部电路实现增益 = 16)	稳定在 ± 2 ADC LSB 精度范围内	200			ns

(1) 这是 OPA 的线性输出范围。OPA 可以输出此范围以外的电压，但电压将不呈线性。

4.2.4.5 OPA 特征

在建议运行条件下测得（除非另有说明）

参数	测试条件	最小值	典型值	最大值	单位
通用					
输入偏置电流			2		nA
短路电流			10		mA
满量程阶跃响应	稳定在 ± 2 LSB (12 位) 精度范围内			450	ns
压摆率	无负载电阻、电容	27		36	V/ μ s
上电时间				500	μ s
直流特征⁽⁵⁾					
增益误差 ⁽¹⁾		-1.5		1.5	%
增益温度系数			± 0.004		%/C
失调误差 ⁽²⁾	以输入为基准	-1.5		1.5	mV
失调温度系数	以输入为基准		± 5.5		μ V/C
直流代码扩展			2.5		12b LSB
AC 特征					
THD ⁽⁴⁾	直流		-78		dB
	高达 100 kHz		-70		dB
CMRR	直流		-60		dB
	高达 100 kHz		-50		dB
PSRR ⁽⁴⁾	直流		-70		dB
	高达 100 kHz		-60		dB
噪声 PSD ⁽⁴⁾	1kHz		200		nV/sqrt(Hz)
集成噪声 (以输入为基准) ⁽⁴⁾	3 Hz 至 30 MHz		100		μ V

- (1) OPA 单独的增益误差。
- (2) OPA 单独的失调误差。
- (3) 3dB 带宽。
- (4) OPA 单独的性能。

4.2.4.6 OPA 典型特征图

图 4-17 所示为输入偏置电流与温度间的关系。

备注 对于图 4-17，需满足以下条件（除非另有说明）：

- $T_A = 30^\circ\text{C}$
- $V_{DDA} = 3\text{V}$
- $V_{DD} = 1.2\text{V}$

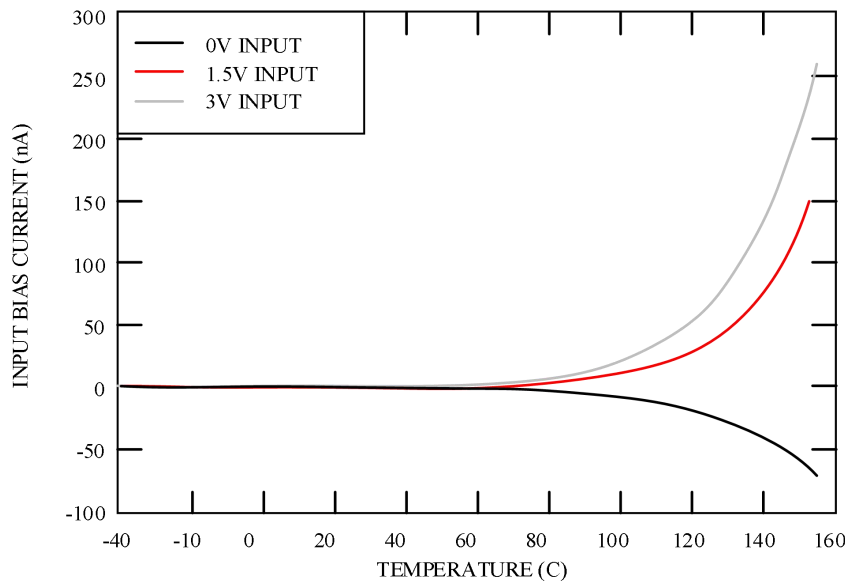


图 4-17 OPA 输入偏置电流与温度间的关系

4.2.4.7 OPA 与 PGA 配置寄存器

1. OPA 与 PGA 配置寄存器 — 0x7030h

15	14	13	12	11	10	9	8
OPAEN	PGA1EN	PGA2EN	PGA3EN	保留	PGA1_GAIN2	PGA1_GAIN1	PGA1_GAIN0
WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0
7	6	5	4	3	2	1	0
保留	PGA2_GAIN2	PGA2_GAIN1	PGA2_GAIN0	保留	PGA3_GAIN2	PGA3_GAIN1	PGA3_GAIN0
WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0

注：R=可读，W=可写，C=清除，_0=复位值，_X=复位值不确定，S=只可被置位；

位15	OPAEN：OPA使能控制位； 0：关闭OPA，释放ADC的B1通道 1：打开运放
位14	PGA1EN：PGA1使能控制位； 0：关闭PGA1，释放ADC的A7通道 1：打开增益可编程运放PGA1
位13	PGA2EN：PGA2使能控制位； 0：关闭PGA2，释放ADC的A3道 1：打开增益可编程运放PGA2
位12	PGA3EN：PGA3使能控制位； 0：关闭PGA3，释放ADC的A0通道 1：打开增益可编程运放PGA3
位11	保留

位10~位8	PGA1_GAIN2~ PGA1_GAIN0: PGA1增益控制位; 111: 保留 110: 保留 101: 保留 100: x16倍 011: x10倍 010: x8倍 001: x4倍 000: x2倍
位7	保留
位6~位4	PGA2_GAIN2~ PGA2_GAIN0: PGA2增益控制位; 111: 保留 110: 保留 101: 保留 100: x16倍 011: x10倍 010: x8倍 001: x4倍 000: x2倍
位3	保留
位2~位0	PGA3_GAIN2~ PGA3_GAIN0: PGA3增益控制位; 111: 保留 110: 保留 101: 保留 100: x16倍 011: x10倍 010: x8倍 001: x4倍 000: x2倍

表4-13 PGA放大倍数配置关系

PGA _x _GAIN2	PGA _x _GAIN1	PGA _x _GAIN0	PGA 放大倍数
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	10
1	0	0	16

 注: PGA_x_GAIN 2、PGA_x_GAIN 1、PGA_x_GAIN 0 中 PGA_x 代表 PGA3, PGA2, PGA1

4.3 同步串行通信接口 (SPI) 模块

该器件集成2个四引脚的同步串行通信接口 (SPI) 模块，SPI模块功能包括：

- 四个外部引脚：
 - SPISOMI：SPI 从输出/主输入引脚
 - SPISIMO：SPI 从输入/主输出引脚
 - $\overline{\text{SPISTE}}$ ：SPI 从发送使能引脚
 - SPICLK：SPI 时钟引脚
- 当不使用SPI模块时，这四个引脚可用作通用IO引脚。
- 两种运行模式：主模式和从模式
- 波特率：支持125种不同波特率
 - (1) 当SPIBRR=3 ~ 127
$$\text{Baud rate} = \frac{\text{LSPCLK}}{(\text{SPIBRR} + 1)}$$
 - (2) 当SPIBRR=0 ~ 2
$$\text{Baud rate} = \frac{\text{LSPCLK}}{4}$$
- 数据字长度：1~16数据位
- 4种时钟模式：
 - 无相位延迟的下降沿：SPICLK高电平有效。SPI 在SPICLK的下降沿发送数据，在SPICLK的上升沿接收数据。
 - 带相位延迟的下降沿：SPICLK高电平有效。SPI 在SPICLK下降沿的前半周期发送数据，而在SPICLK的下降沿接收数据。
 - 无相位延迟的上升沿：SPICLK低电平无效。SPI 在SPICLK的上升沿上发送数据，而在SPICLK的下降沿接收数据。
 - 带相位延迟的上升沿：SPICLK低电平无效。SPI 在SPICLK下降沿前的半周期发送数据，而在SPICLK的上升沿接收数据。

- 同时接收和发送操作（发送功能可以软件禁用）
- 同时接收和发送操作可以通过中断或轮询方式来完成。
- 9 个SPI 模块控制寄存器：位于控制寄存器内，帧开始地址7040h。
- 1-16位数据，发送数据为左对齐，接收的数据为右对齐；
- 4 级发送/接收FIFO
- 延迟的发送控制
- 支持双向 3 线 SPI 模式

SPI 端口运行由表4-14中列出的寄存器配置和控制。

表4-14 SPI-A 寄存器

名称	地址	大小 (x16)	受 EALLOW 保护	说明 ⁽¹⁾
SPICCR	0x00 7040	1	否	SPI-A 配置控制寄存器
SPICTL	0x00 7041	1	否	SPI-A 运行控制寄存器
SPISTS	0x00 7042	1	否	SPI-A 状态寄存器
SPIBRR	0x00 7044	1	否	SPI-A 波特率寄存器
SPIRXEMU	0x00 7046	1	否	SPI-A 接收仿真缓冲器寄存器
SPIRXBUF	0x00 7047	1	否	SPI-A 串行输入缓冲器寄存器
SPI TXBUF	0x00 7048	1	否	SPI-A 串行输出缓冲器寄存器
SPIDAT	0x00 7049	1	否	SPI-A 串行数据寄存器
SPIFFTX	0x00 704A	1	否	SCI-A FIFO 发送寄存器
SPIFFRX	0x00 704B	1	否	SCI-A FIFO 接收寄存器
SPIFFCT	0x00 704C	1	否	SCI-A FIFO 控制寄存器
SPIPRI	0x00 704F	1	否	SCI-A 优先级控制寄存器

(1) 此表中的寄存器映射到外设帧 2。此空间只允许 16 位访问。32 位访问会产生未知结果。

图4-18为从模式下的 SPI 框图。

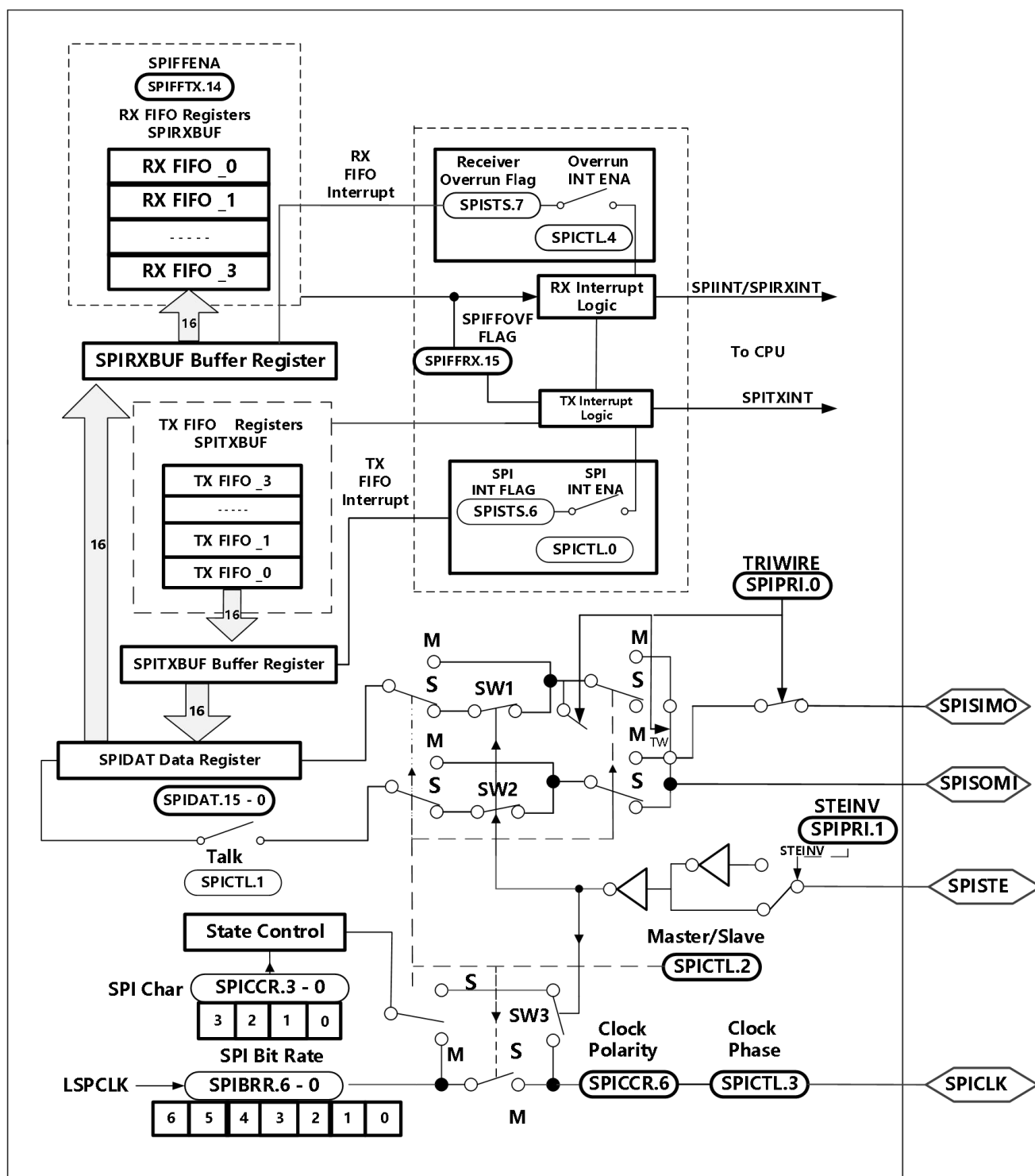


图 4-18 SPI 模块框图 (从模式)

4.3.1 同步串行通信接口(SPI) 主模式时序

表4-15列出了主模式时序（时钟相位= 0），表4-16列出了主模式时序（时钟相位=1）。图4-19和图4-20显示了时序波形。

表4-15 SPI主模式外部时序（时钟相位= 0） (1) (2) (3) (4) (5)

编号			当 (SPIBRR+1) 为偶数或者 SPIBRR=0 或者2 时的SPI		当 (SPIBRR+1) 为奇数并且SPIBRR>3 时的SPI		单位
			最小值	最大值	最小值	最大值	
1	$t_{c(SPC)M}$	周期时间, SPICLK	$4t_{c(LCO)}$	$128t_{c(LCO)}$	$5t_{c(LCO)}$	$127t_{c(LCO)}$	ns
2	$t_{w(SPCH)M}$	脉冲持续时间, SPICLK 高电平的时间 (时钟极性= 0)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}-0.5t_{c(LCO)}-10$	$0.5t_{c(SPC)M}-0.5t_{c(LCO)}$	ns
	$t_{w(SPCL)M}$	脉冲持续时间, SPICLK 低电平的时间 (时钟极性= 1)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}-0.5t_{c(LCO)}-10$	$0.5t_{c(SPC)M}-0.5t_{c(LCO)}$	
3	$t_{w(SPCL)M}$	脉冲持续时间, SPICLK 低电平的时间 (时钟极性= 0)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}+0.5t_{c(LCO)}-10$	$0.5t_{c(SPC)M}+0.5t_{c(LCO)}$	ns
	$t_{w(SPCH)M}$	脉冲持续时间, SPICLK 高电平的时间 (时钟极性= 1)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}+0.5t_{c(LCO)}-10$	$0.5t_{c(SPC)M}+0.5t_{c(LCO)}$	
4	$t_{d(SPCH-SIMO)M}$	延迟时间, SPICLK 高电平至 SPISIMO 有效的时间 (时钟极性= 0)		10		10	ns
	$t_{d(SPCL-SIMO)M}$	延迟时间, SPICLK 低电平至 SPISIMO 有效的时间 (时钟极性= 1)		10		10	
5	$t_{v(SPCL-SIMO)M}$	有效时间, SPICLK 低电平后, SPISIMO 数据有效的时间 (时钟极性= 0)	$0.5t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}+0.5t_{c(LCO)}-10$		ns
	$t_{v(SPCH-SIMO)M}$	有效时间, SPICLK 高电平之后, SPISIMO 数据有效的时间 (时钟极性= 1)	$0.5t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}+0.5t_{c(LCO)}-10$		
8	$t_{su(SOMI-SPCL)M}$	建立时间, SPISOMI 在 SPICLK 低电平之前的时间 (时钟极性= 0)	26		26		ns
	$t_{su(SOMI-SPCH)M}$	建立时间, SPISOMI 在 SPICLK 高电平之前的时间 (时钟极性= 1)	26		26		
9	$t_{v(SPCL-SOMI)M}$	有效时间, SPICLK 低电平之后 SPISOMI 数据有效的时间 (时钟极性= 0)	$0.25t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-0.5t_{c(LCO)}-10$		ns
	$t_{v(SPCH-SOMI)M}$	有效时间, SPICLK 高电平之后 SPISOMI 数据有效的时间 (时钟极性= 1)	$0.25t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-0.5t_{c(LCO)}-10$		

(1) 主模式/从模式位(SPICTL.2) 被设定, 而时钟相位的位(SPICTL.3) 被清除。

(2) $t_{c(SPC)}$ = SPI 时钟周期时间= LSPCLK/4 或者 LSPCLK/(SPIBRR + 1)

(3) 必须调整内部时钟预分频器, 使SPI时钟限制在以下 SPI 时钟速率:

主模式发送最大值 15MHz, 主模式接收最大值 10MHz

从模式发送最大值 10MHz, 从模式接收最大值 10MHz。

(4) $t_{c(LCO)}$ = LSPCLK 周期时间

(5) 作为基准的SPICLK 信号的有效边沿由CLOCK POLARITY (时钟极性) 位(SPICCR 6) 控制。

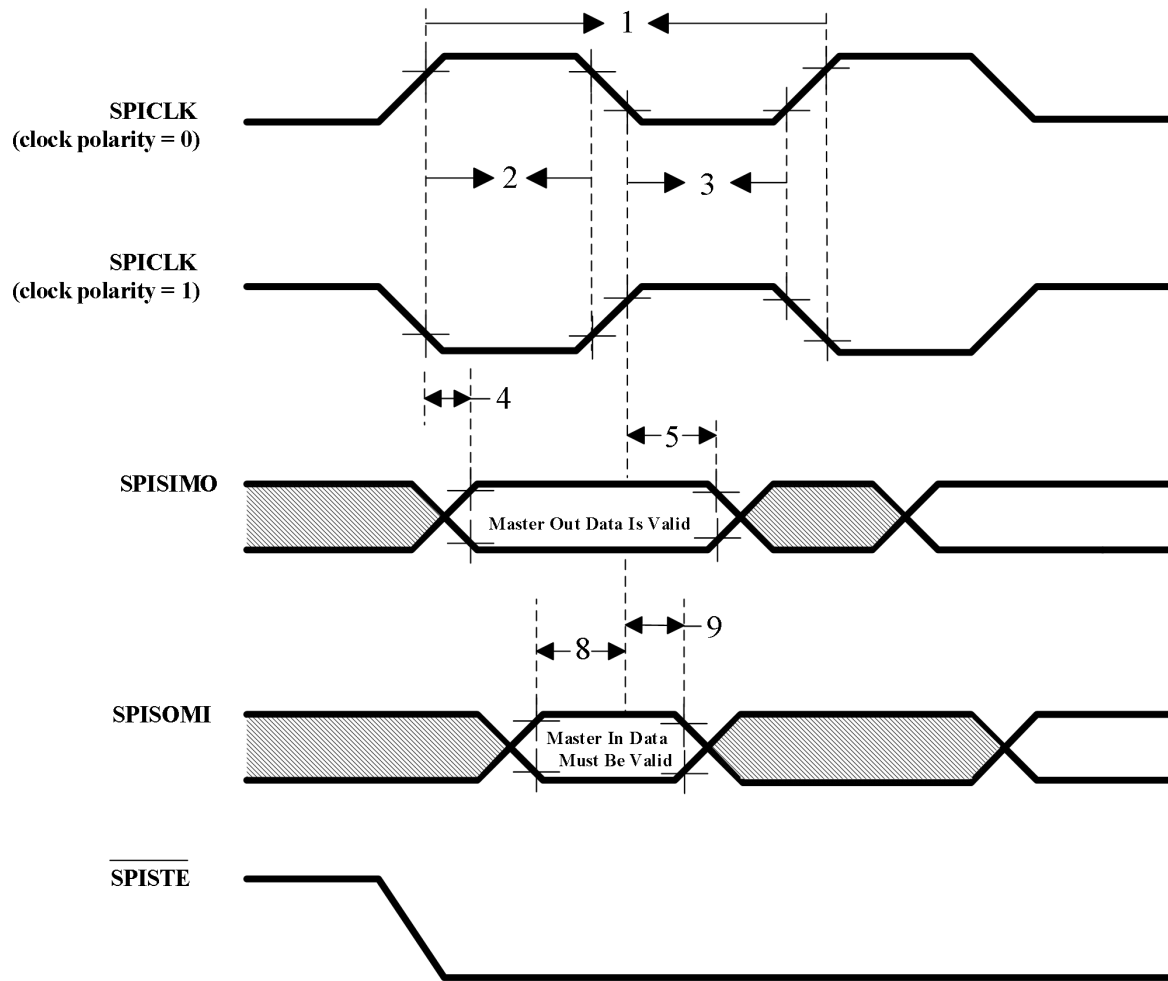


图4-19 SPI 主模式外部时序 (时钟相位= 0)

表 4-16 SPI 主模式外部时序 (时钟相位= 1) (1) (2) (3) (4) (5)

编号			当(SPIBRR+1) 为偶数或者SPIBRR=0 或者2 时的 SPI		当(SPIBRR+1) 为奇数并且 SPIBRR>3 时的 SPI		单位
			最小值	最大值	最小值	最大值	
1	$t_{c(SPC)M}$	周期时间, SPICLK	$4t_{c(LCO)}$	$128t_{c(LCO)}$	$5t_{c(LCO)}$	$127t_{c(LCO)}$	ns
2	$t_{w(SPCH)M}$	脉冲持续时间, SPICLK 高电平的时间 (时钟极性= 0)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}-0.5t_{c(LCO)}-10$	$0.5t_{c(SPC)M}-0.5t_{c(LCO)}$	ns
	$t_{w(SPCL)M}$	脉冲持续时间, SPICLK 低电平的时间 (时钟极性= 1)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}-0.5t_{c(LCO)}-10$	$0.5t_{c(SPC)M}-0.5t_{c(LCO)}$	
3	$t_{w(SPCL)M}$	脉冲持续时间, SPICLK 低电平的时间 (时钟极性= 0)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}+0.5t_{c(LCO)}-10$	$0.5t_{c(SPC)M}+0.5t_{c(LCO)}$	ns
	$t_{w(SPCH)M}$	脉冲持续时间, SPICLK 高电平的时间 (时钟极性= 1)	$0.5t_{c(SPC)M}-10$	$0.5t_{c(SPC)M}$	$0.5t_{c(SPC)M}+0.5t_{c(LCO)}-10$	$0.5t_{c(SPC)M}+0.5t_{c(LCO)}$	
6	$t_{su(SIMO-SPCH)M}$	建立时间, 在 SPICLK 高电平之前 SPISIMO 数据有效的的时间 (时钟极性= 0)	$0.5t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-10$		ns
	$t_{su(SIMO-SPCL)M}$	建立时间, 在 SPICLK 低电平之前 SPISIMO 数据有效的的时间 (时钟极性= 1)	$0.5t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-10$		
7	$t_v(SPCH-SIMO)M$	有效时间, SPICLK 高电平之后 SPISIMO 数据有效的的时间 (时钟极性= 0)	$0.5t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-10$		ns
	$t_v(SPCL-SIMO)M$	有效时间, SPICLK 低电平后, SPISIMO 数据有效的的时间 (时钟极性= 1)	$0.5t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-10$		
10	$t_{su(SOMI-SPCH)M}$	建立时间, SPISOMI 在 SPICLK 高电平之前的时间 (时钟极性= 0)	26		26		ns
	$t_{su(SOMI-SPCL)M}$	建立时间, SPISOMI 在 SPICLK 低电平之前的时间 (时钟极性= 1)	26		26		
11	$t_v(SPCH-SOMI)M$	有效时间, SPICLK 高电平之后 SPISOMI 数据有效的的时间 (时钟极性= 0)	$0.25t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-10$		ns
	$t_v(SPCL-SOMI)M$	有效时间, SPICLK 低电平之后 SPISOMI 数据有效的的时间 (时钟极性= 1)	$0.25t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-10$		

(1) 主模式/从模式(SPICTL.2) 被设定, 而时钟相位的位(SPICTL.3) 被清除。

(2) $t_{c(SPC)} = \text{SPI 时钟周期时间} = \text{LSPCLK}/4$ 或者 $\text{LSPCLK}/(\text{SPIBRR} + 1)$

(3) 必须调整内部时钟预分频器, 使SPI 时钟速度限制在下列 SPI 时钟速率上:

主模式发送最大值 15MHz, 主模式接收最大值 10MHz

从模式发送最大值 10MHz, 从模式接收最大值 10MHz。

(4) $t_{c(LCO)} = \text{LSPCLK 周期时间}$

(5) 作为基准的SPICLK 信号的有效边沿由CLOCK POLARITY (时钟极性) 位(SPICCR 6) 控制。

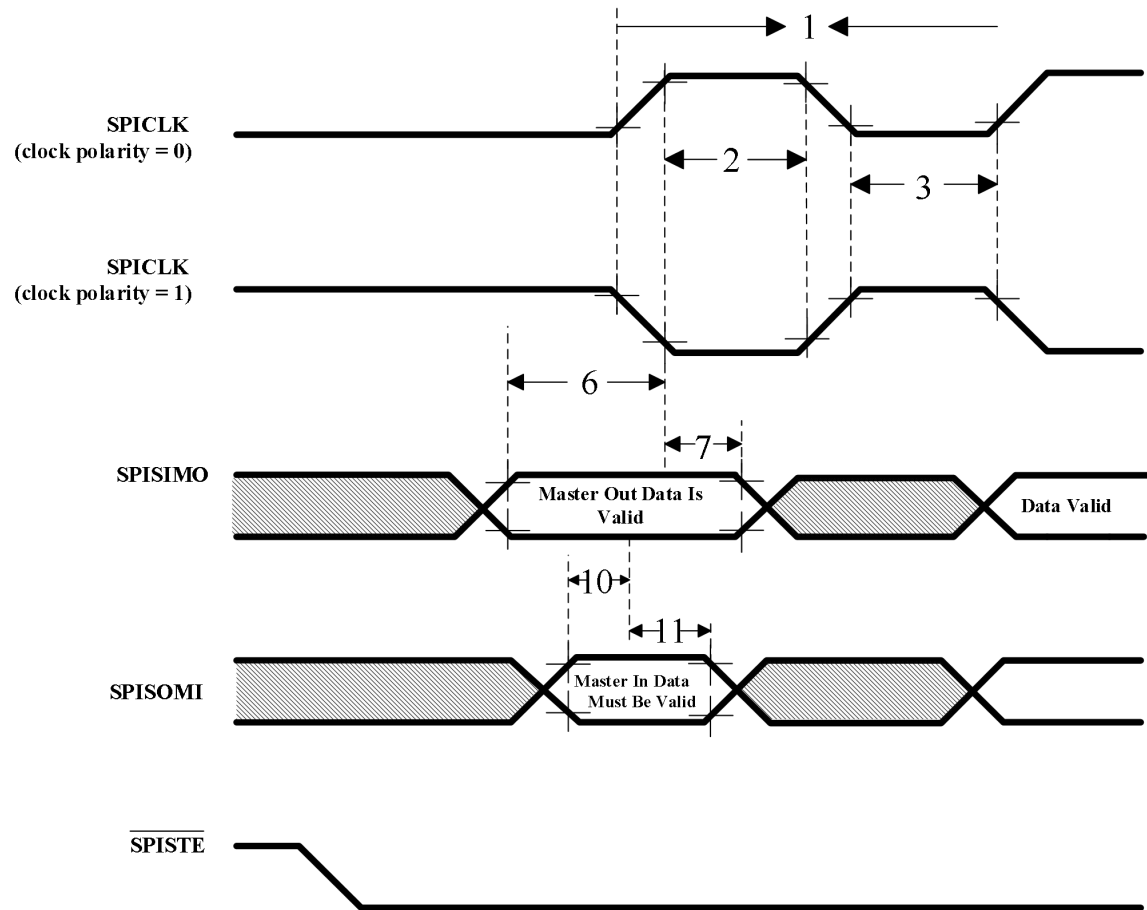


图 4-20 SPI 主模式外部时序 (时钟相位= 1)

4.3.2 SPI 从模式时序

表4-17列出了从模式外部时序（时钟相位= 0），而表4-18列出了从模式（时钟相位= 1）、图4-21和图4-22显示了时序波形。

表4-17 SPI 从模式外部时序（时钟相位= 0） (1) (2) (3) (4) (5)

编号		最小值	最大值	单位
1	$t_{c(SPC)}$ 周期时间, SPICLK	$4t_{c(LCO)}$		ns
2	$t_{w(SPCH)}$ 脉冲持续时间, SPICLK 高电平的时间 (时钟极性= 0)	$0.5t_{c(SPC)}-10$	$0.5t_{c(SPC)}$	ns
	$t_{w(SPCL)}$ 脉冲持续时间, SPICLK 低电平的时间 (时钟极性= 1)	$0.5t_{c(SPC)}-10$	$0.5t_{c(SPC)}$	
3	$t_{w(SPCL)}$ 脉冲持续时间, SPICLK 低电平的时间 (时钟极性= 0)	$0.5t_{c(SPC)}-10$	$0.5t_{c(SPC)}$	ns
	$t_{w(SPCH)}$ 脉冲持续时间, SPICLK 高电平的时间 (时钟极性= 1)	$0.5t_{c(SPC)}-10$	$0.5t_{c(SPC)}$	
4	$t_{d(SPCH-SOMI)}$ 延迟时间, SPICLK 高电平至 SPISOMI 有效的时间 (时钟极性= 0)		21	ns
	$t_{d(SPCL-SOMI)}$ 延迟时间, SPICLK 低电平至 SPISOMI 有效的时间 (时钟极性= 1)		21	
5	$t_{v(SPCL-SOMI)}$ 有效时间, SPICLK 低电平之后 SPISOMI 数据有效的时间 (时钟极性= 0)	$0.75t_{c(SPC)}$		ns
	$t_{v(SPCH-SOMI)}$ 有效时间, SPICLK 高电平之后 SPISOMI 数据有效的时间 (时钟极性= 1)	$0.75t_{c(SPC)}$		
6	$t_{su(SIMO-SPCL)}$ 建立时间, SPISIMO 在 SPICLK 低电平之前的时间 (时钟极性= 0)	26		ns
	$t_{su(SIMO-SPCH)}$ 建立时间, SPISIMO 在 SPICLK 高电平之前的时间 (时钟极性= 1)	26		
7	$t_{v(SPCL-SIMO)}$ 有效时间, SPICLK 低电平后, SPISIMO 数据有效的时间 (时钟极性= 0)	$0.5t_{c(SPC)}-10$		ns
	$t_{v(SPCH-SIMO)}$ 有效时间, SPICLK 高电平之后 SPISIMO 数据有效的时间 (时钟极性= 1)	$0.5t_{c(SPC)}-10$		

(1) 主模式/从模式位(SPICTL.2) 被设定, 而时钟相位的位(SPICTL.3) 被清除。

(2) $t_{c(SPC)}$ = SPI 时钟周期时间= LSPCLK/4 或者 LSPCLK/(SPIBRR + 1)

(3) 必须调整内部时钟预分频器, 使SPI 时钟速度限制在下列 SPI 时钟速率上:

主模式发送最大值 15MHz, 主模式接收最大值 10MHz

从模式发送最大值 10MHz, 从模式接收最大值 10MHz。

(4) $t_{c(LCO)}$ = LSPCLK 周期时间

(5) 作为基准的SPICLK 信号的有效边沿由CLOCK POLARITY (时钟极性) 位(SPICCR. 6) 控制。

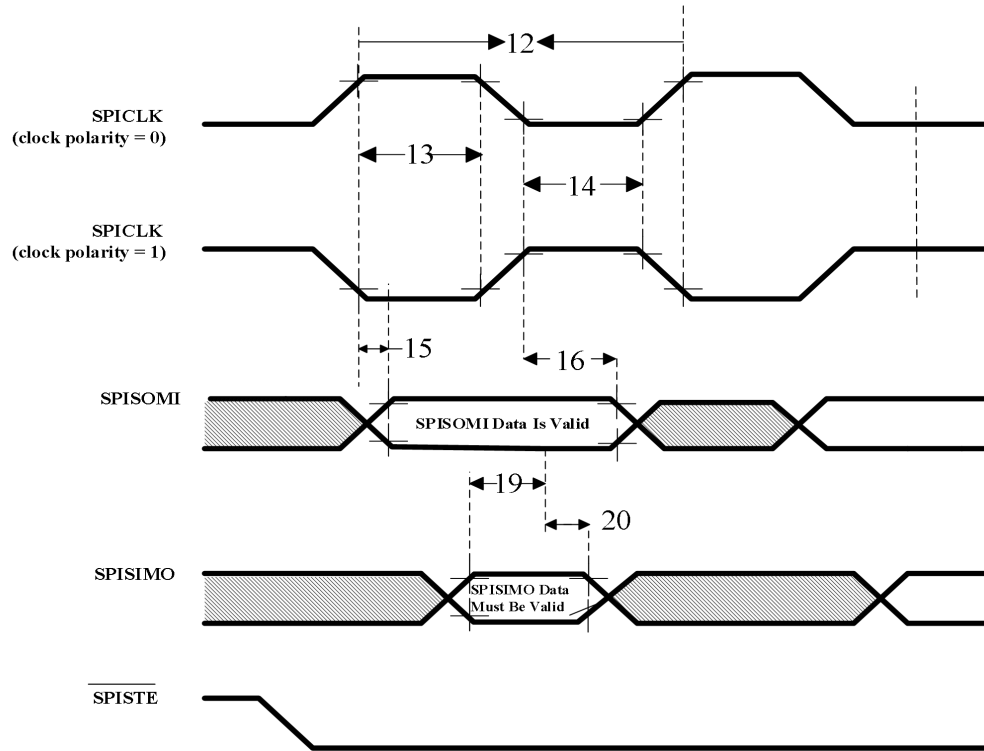


图4-21 SPI 从模式外部时序 (时钟相位= 0)

表 4-18 SPI 从模式外部时序 (时钟相位= 1) (1) (2) (3) (4)

编号		最小值	最大值	单位
1	$t_{c(SPC)}S$ 周期时间, SPICLK	$8t_{c(LCO)}$		ns
2	$t_{w(SPCH)}S$ 脉冲持续时间, SPICLK 高电平的时间 (时钟极性=0)	$0.5t_{c(SPC)}S-10$	$0.5t_{c(SPC)}S$	ns
	$t_{w(SPCL)}S$ 脉冲持续时间, SPICLK 低电平的时间 (时钟极性=1)	$0.5t_{c(SPC)}S-10$	$0.5t_{c(SPC)}S$	
3	$t_{w(SPCL)}S$ 脉冲持续时间, SPICLK 低电平的时间 (时钟极性=0)	$0.5t_{c(SPC)}S-10$	$0.5t_{c(SPC)}S$	ns
	$t_{w(SPCH)}S$ 脉冲持续时间, SPICLK 高电平的时间 (时钟极性=1)	$0.5t_{c(SPC)}S-10$	$0.5t_{c(SPC)}S$	
4	$t_{su(SOMI-SPCH)}S$ 建立时间, SPISOMI 在 SPICLK 高电平之前的时间 (时钟极性=0)	$0.125t_{c(SPC)}S$		ns
	$t_{su(SOMI-SPCL)}S$ 建立时间, SPISOMI 在 SPICLK 低电平之前的时间 (时钟极性=1)	$0.125t_{c(SPC)}S$		
5	$t_{v(SPL-SOMI)}S$ 有效时间, SPICLK 低电平后 SPISOMI 数据有效的时间 (时钟极性=1)	$0.75t_{c(SPC)}S$		ns
	$t_{v(SPCH-SOMI)}S$ 有效时间, SPICLK 高电平后 SPISOMI 数据有效时间 (时钟极性=0)	$0.75t_{c(SPC)}S$		
6	$t_{su(SIMO-SPCH)}S$ 建立时间, SPISIMO 在 SPICLK 高电平之前的时间 (时钟极性=0)	26		ns
	$t_{su(SIMO-SPCL)}S$ 建立时间, SPISIMO 在 SPICLK 低电平之前的时间 (时钟极性=1)	26		
7	$t_{v(SPCH-SIMO)}S$ 有效时间, SPICLK 高电平后 SPISIMO 数据有效的时间 (时钟极性=0)	$0.5t_{c(SPC)}S-10$		ns
	$t_{v(SPL-SIMO)}S$ 有效时间, SPICLK 低电平后, SPISIMO 数据有效的时间 (时钟极性=1)	$0.5t_{c(SPC)}S-10$		

(1) 主模式/从模式位(SPICTL.2) 被设定, 而时钟相位的位(SPICTL.3) 被清除。

(2) $t_{c(SPC)} = \text{SPI 时钟周期时间} = \text{LSPCLK}/4$ 或者 $\text{LSPCLK}/(\text{SPIBRR} + 1)$

(3) 必须调整内部时钟预分频器, 这样的话, SPI 时钟速度限制在下列 SPI 时钟速率上:

主模式发送最大值 15MHz, 主模式接收最大值 10MHz

从模式发送最大值 10MHz, 从模式接收最大值 10MHz。

(4) 作为基准的SPICLK 信号的有效边沿由CLOCK POLARITY (时钟极性) 位(SPICCR. 6) 控制。

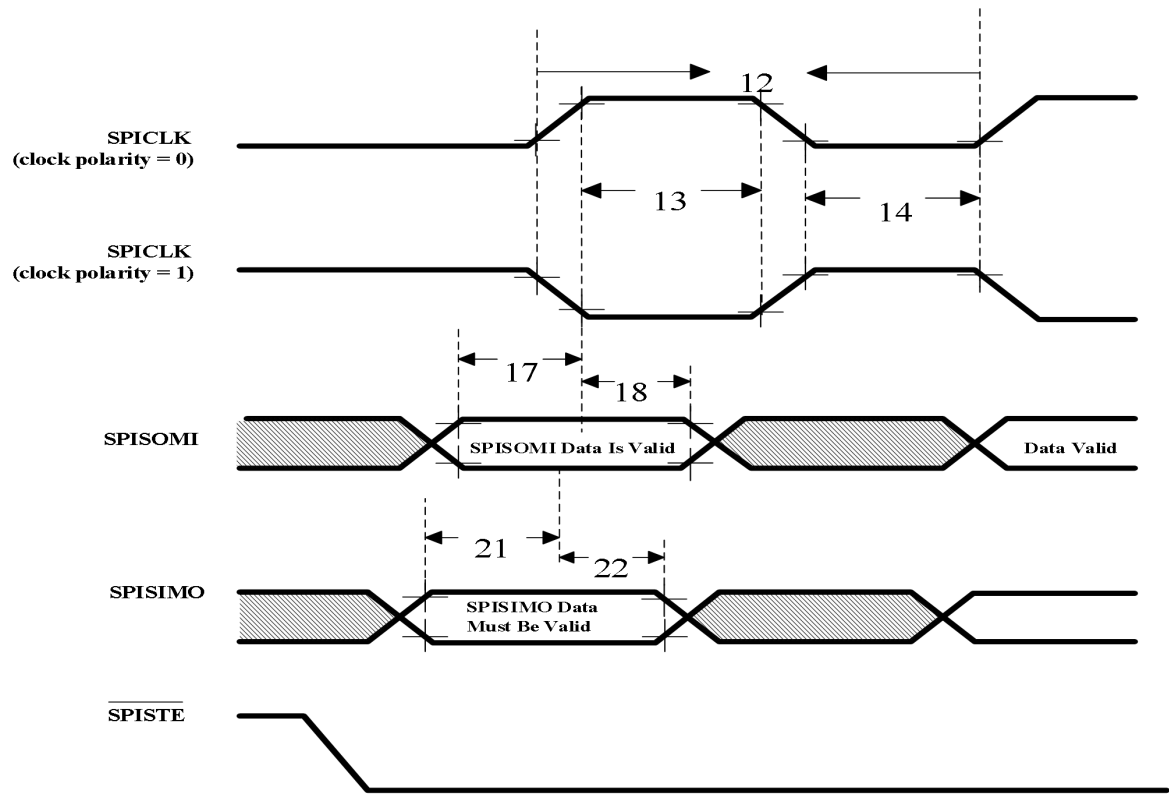


图 4-22 SPI 从模式外部时序 (时钟相位= 1)

4.4 异步串行通信接口 (SCI) 模块

该器件包括1个异步串行通信接口(SCI) 模块 (SCI-A) 。SCI 模块支持CPU 与其它异步外设之间使用标准不归零码 (NRZ) 格式的数字通信。SCI 接收器和发送器是双缓冲的，各自具有独立的使能和中断位，两者可独立运行或者在全双工模式下同时运行。为了确保数据完整性，SCI模块会对接收数据进行间断检测、奇偶校验、溢出和帧信息错误检测。通过一个16位波特率选择寄存器，可对波特率进行编程。

SCI 模块的特性包括：

- 两个外部引脚：
 - SCITXD: SCI 发送-输出引脚
 - SCIRXD: SCI 接收-输入引脚
- 当不使用SCI模块时，则四个引脚可用作通用IO引脚。
- 64000种波特率

(1) 当BRR≠0

$$\text{Baudrate} = \frac{\text{LSPCLK}}{(\text{BRR} + 1) * 8}$$

(2) 当BRR=0

$$\text{Baudrate} = \frac{\text{LSPCLK}}{16}$$

- 数据字格式
 - 一个开始位
 - 数据字长可设定为 1 至 8 位
 - 可选偶数、奇数或者无奇偶校验位
 - 一个或者两个停止位
- 四个错误检测标志：奇偶校验、溢出、帧信息错误和间断检测
- 两个唤醒多处理器模式：空闲线模式和地址位模式
- 半双工或者全双工运行
- 双缓冲接收和发送功能

•发送和接收操作可通过带有状态标志的中断驱动或者轮询来完成。

- 发送器：TXRDY 标志（发送器缓冲寄存器已准备好接收另一个字符）和 TX EMPTY 标志（发送器移位寄存器为空）
- 接收器：RXRDY 标志（接收器缓冲寄存器已准备好接收另一个字符），BRKDT 标志（产生了中断），和 RX ERROR 错误标志（监测四个中断条件）

•发送器和接收器中断的独立使能位（BRKDT除外）

•NRZ（不归零）码格式

- 自动波特率检测硬件逻辑电路
- 4 级发送/接收FIFO

注意：

此模块中的所有寄存器都是连接至外设帧 2 的 8 位寄存器。当访问寄存器时，寄存器数据位于低字节 (7~0)。高字节 (15~8) 读取为零，对高字节的写入无效。

SCI 配置和控制寄存器见表 4-19。

表4-19 SCI-A 寄存器⁽¹⁾

名称	地址	大小 (x16)	受 EALLOW 保护	说明
SCICRA	0x00 7050	1	否	SCI-A 通信控制寄存器
SCICTL1A	0x00 7051	1	否	SCI-A 控制寄存器
SCIHBAUDA	0x00 7052	1	否	SCI-A 波特率寄存器，高位
SCILBAUDA	0x00 7053	1	否	SCI-A 波特率寄存器，低位
SCICTL2A	0x00 7054	1	否	SCI-A 控制寄存器2
SCIRXSTA	0x00 7055	1	否	SCI-A 接收状态寄存器
SCIRXEMUA	0x00 7056	1	否	SCI-A 接收仿真数据缓冲寄存器
SCIRXBUFA	0x00 7057	1	否	SCI-A 接收数据缓冲寄存器
SCITXBUFA	0x00 7059	1	否	SCI-A 发送数据缓冲寄存器
SCIFFTXA ⁽²⁾	0x00 705A	1	否	SCI-A FIFO 发送寄存器
SCIFFRXA ⁽²⁾	0x00 705B	1	否	SCI-A FIFO 接收寄存器
SCIFFCTA ⁽²⁾	0x00 705C	1	否	SCI-A FIFO 控制寄存器
SCIPRIA	0x00 705F	1	否	SCI-A 优先级控制寄存器

(1) 此表中的寄存器映射到外设帧 2 空间。此空间只允许 16 位访问，32 位访问会产生未知结果。

(2) 这些寄存器是用于 FIFO 模式的寄存器。

图4-23所示为SCI模块框图。

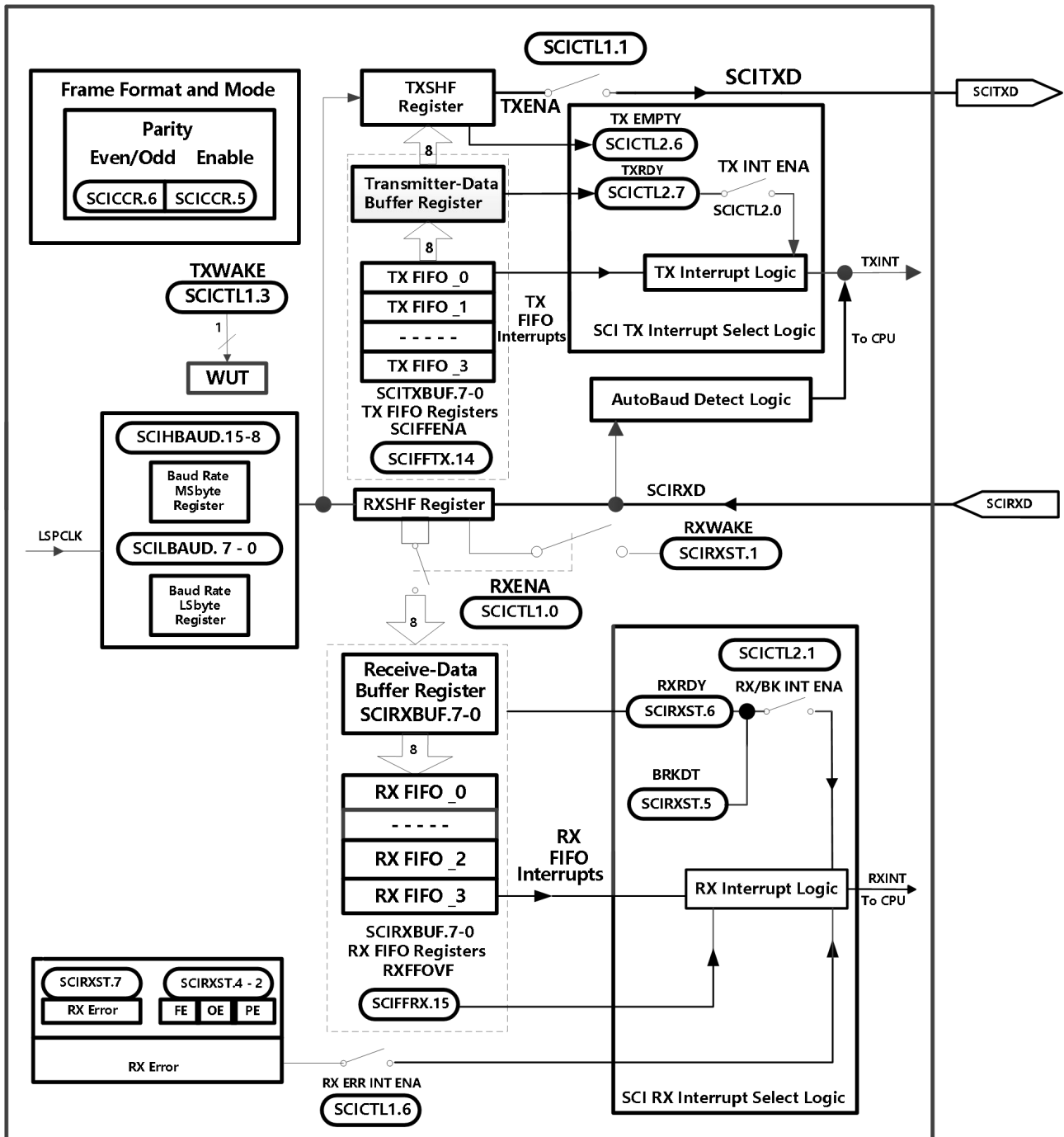


图4-23 SCI 模块框图

4.5 本地互连网络 (LIN)

该器件包含一个 LIN 控制器。LIN 标准基于 SCI (UART) 串行数据格式。LIN 模块也可配置为作为 SCI 运行。LIN 模块具有以下特性：

- LIN 1.3 或者 兼容 2.0 协议
- 两个外部引脚：LINRX 和 LINTX
- 多缓冲接收和发送单元
- 用于信息过滤的识别掩码
- 自动主器件头文件生成
 - 可编程同步中断域
 - 同步域
 - 标识符域
- 从器件自动同步
 - 同步中断检测
 - 可选波特率更新
 - 同步验证
- 带有 7 个分数位的 2^{31} 个可编程传输速率
- 从收发器 LINRX 显性电平唤醒
- 自动唤醒支持
 - 唤醒信号生成
 - 唤醒信号超期时间
- 自动总线闲置检测
- 错误检测
 - 位错误
 - 总线错误

- 无响应错误
- 校验和错误
- 同步域错误
- 奇偶校验错误
- 2 个具有优先级编码的中断：
 - 接收
 - 发送
 - ID，错误和状态

表 4-20 为 LIN 寄存器。

表 4-20 LIN-A 寄存器⁽¹⁾

名称	地址	大小 (x16)	说明
SCIGCR0	0x6C00	2	全局控制寄存器 0
SCIGCR1	0x6C02	2	全局控制寄存器 1
SCIGCR2	0x6C04	2	全局控制寄存器 2
SCISSETINT	0x6C06	2	中断使能寄存器
SCICLEARINT	0x6C08	2	中断禁用寄存器
SCISSETINTLVL	0x6C0A	2	设置中断级别寄存器
SCICLEARINTLVL	0x6C0C	2	清除中断级别寄存器
SCIFLR	0x6C0E	2	标志寄存器
SCIINTVECT0	0x6C10	2	中断向量偏移寄存器 0
SCIINTVECT1	0x6C12	2	中断向量偏移寄存器 1
SCIFORMAT	0x6C14	2	长度控制寄存器
BRSR	0x6C16	2	波特率选择寄存器
SCIED	0x6C18	2	仿真缓冲寄存器
SCIRD	0x6C1A	2	接收器数据缓冲寄存器
SCITD	0x6C1C	2	发送数据缓冲寄存器
保留	0x6C1E	4	RSVD
SIPIO2	0x6C22	2	引脚控制寄存器 2
保留	0x6C24	10	RSVD
LINCOMP	0x6C30	2	比较寄存器
LINRD0	0x6C32	2	接收数据寄存器 0
LINRD1	0x6C34	2	接收数据寄存器 1
LINMASK	0x6C36	2	接收屏蔽寄存器
LINID	0x6C38	2	寄存器包含 ID 字节，ID 从任务字节和接收到的 ID 域。
LINTD0	0x6C3A	2	发送数据寄存器 0
LINTD1	0x6C3C	2	发送数据寄存器 1
MBRSR	0x6C3E	2	波特率选择寄存器
保留	0x6C40	8	RSVD
IODFTCTRL	0x6C48	2	用于 BLIN 的 IODFT

(1) 一些寄存器和其它寄存器中的一些位是受 EALLOW 保护的。

图 4-24 所示为 LIN 模块框图。

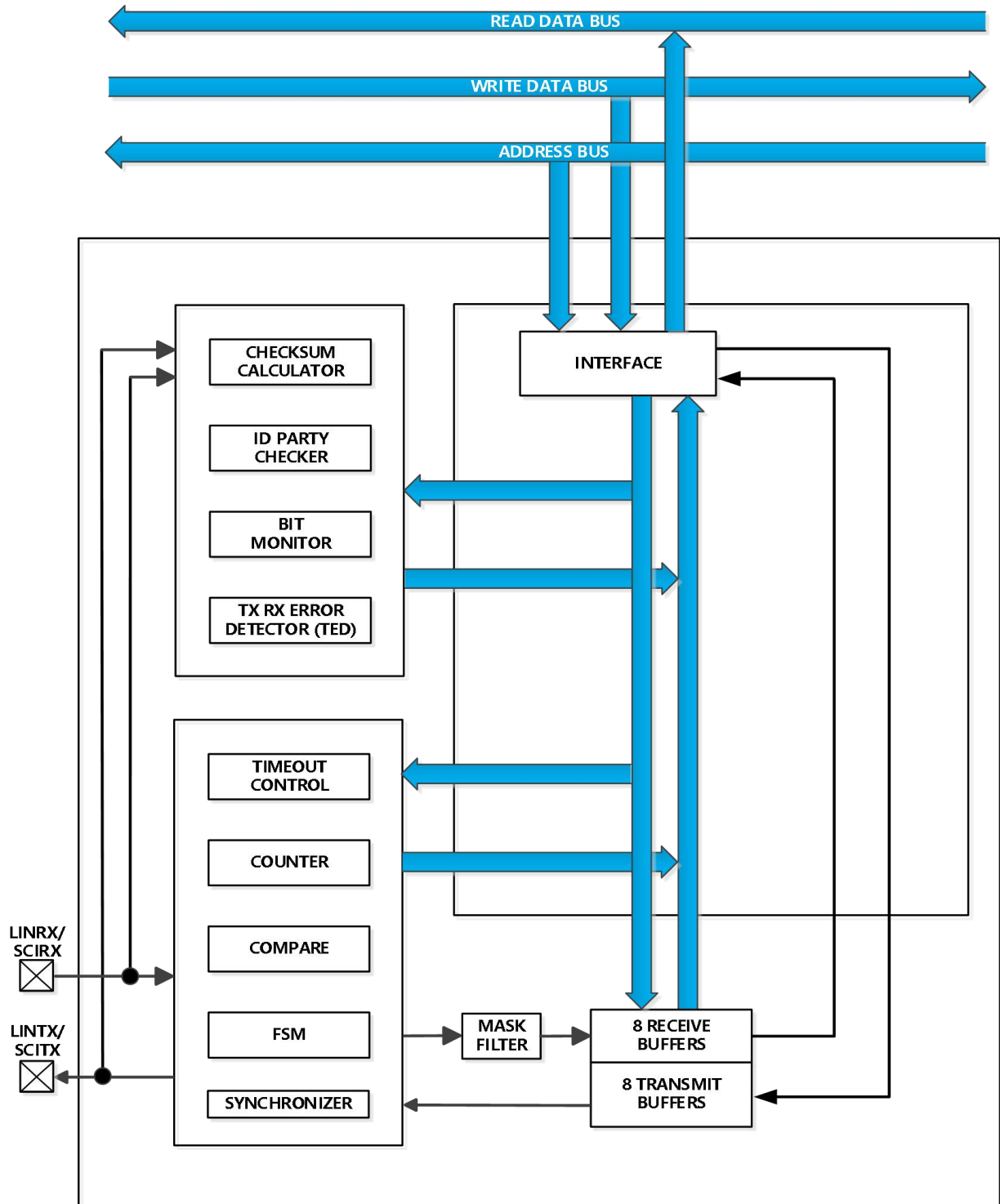


图 4-24 LIN 框图

4.6 增强型控制器局域网络 (eCAN) 模块

CAN 模块(eCAN-A)有下列特性:

- 与CAN协议 (版本2.0B) 完全兼容
- 支持高达1Mbps 的数据速率
- 32 个邮箱, 每个邮箱有下列属性:
 - 可配置为接收或者发送
 - 可使用标准或者扩展标识符进行配置
 - 有一个可编程的接收掩码
 - 支持数据和远程帧
 - 由0 至8 字节数据组成
 - 在接收和发送消息上使用一个32 位时间戳
 - 防止接收新消息
 - 保持发送消息的动态可编程优先级
 - 采用具有两个中断级别的可编程中断机制
 - 采用针对发送或接收超时的可编程警报
- 低功耗模式
- 总线活动上的可编程唤醒
- 针对远程请求消息的自动答复
- 丢失仲裁或者错误情况下的帧自动重传
- 通过一个特定消息实现同步的 32 位本地网络时间计数器 (与邮箱16 协同通信)
- 自测模式
 - 回路模式下接收自身消息, 提供一个虚拟应答, 从而无需由其它节点提供应答。

注

对于 100MHz 的 SYSCLKOUT, 最小的比特率为 7.8125 kbps;

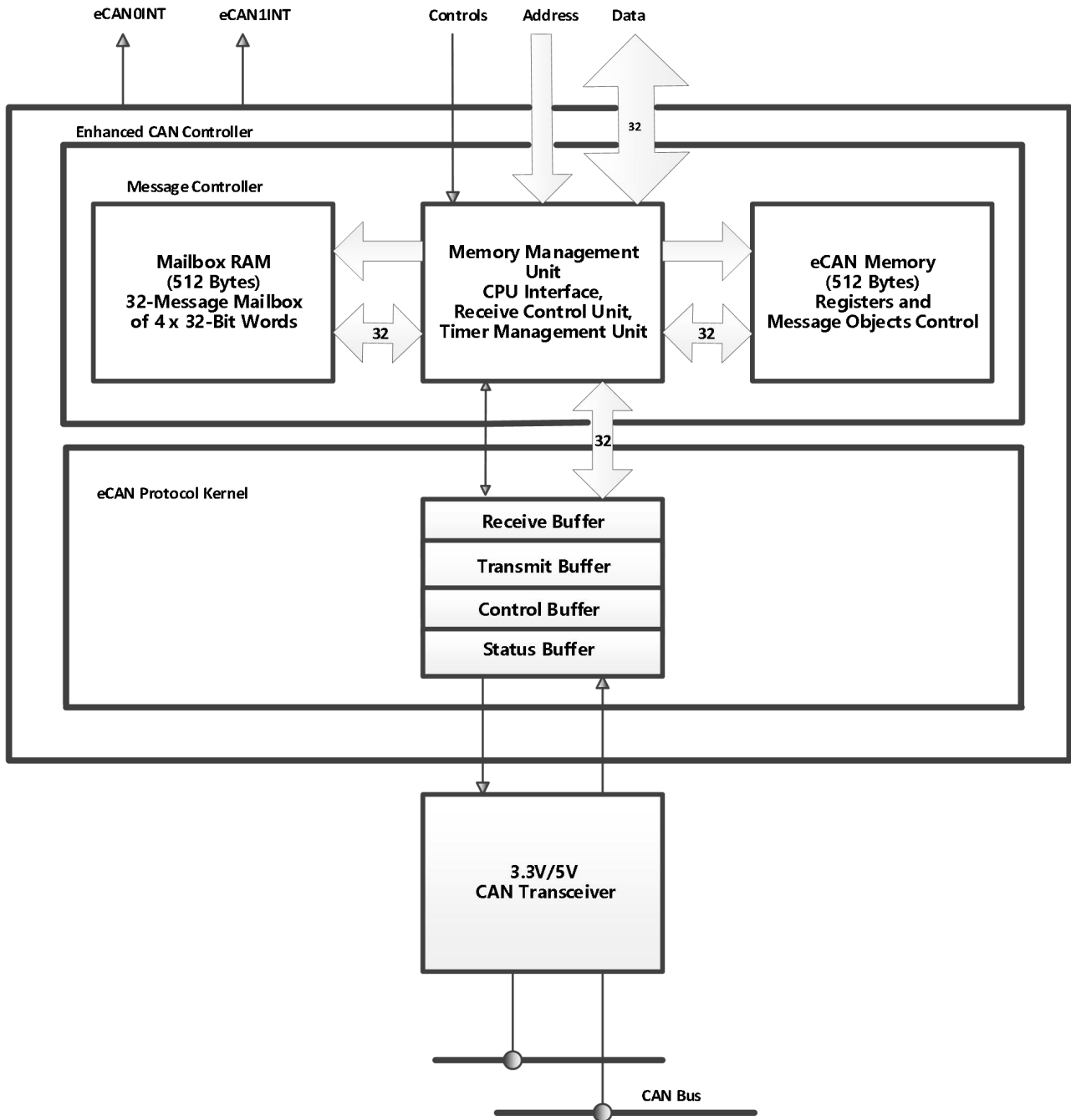


图 4-25 eCAN 框图和接口电路

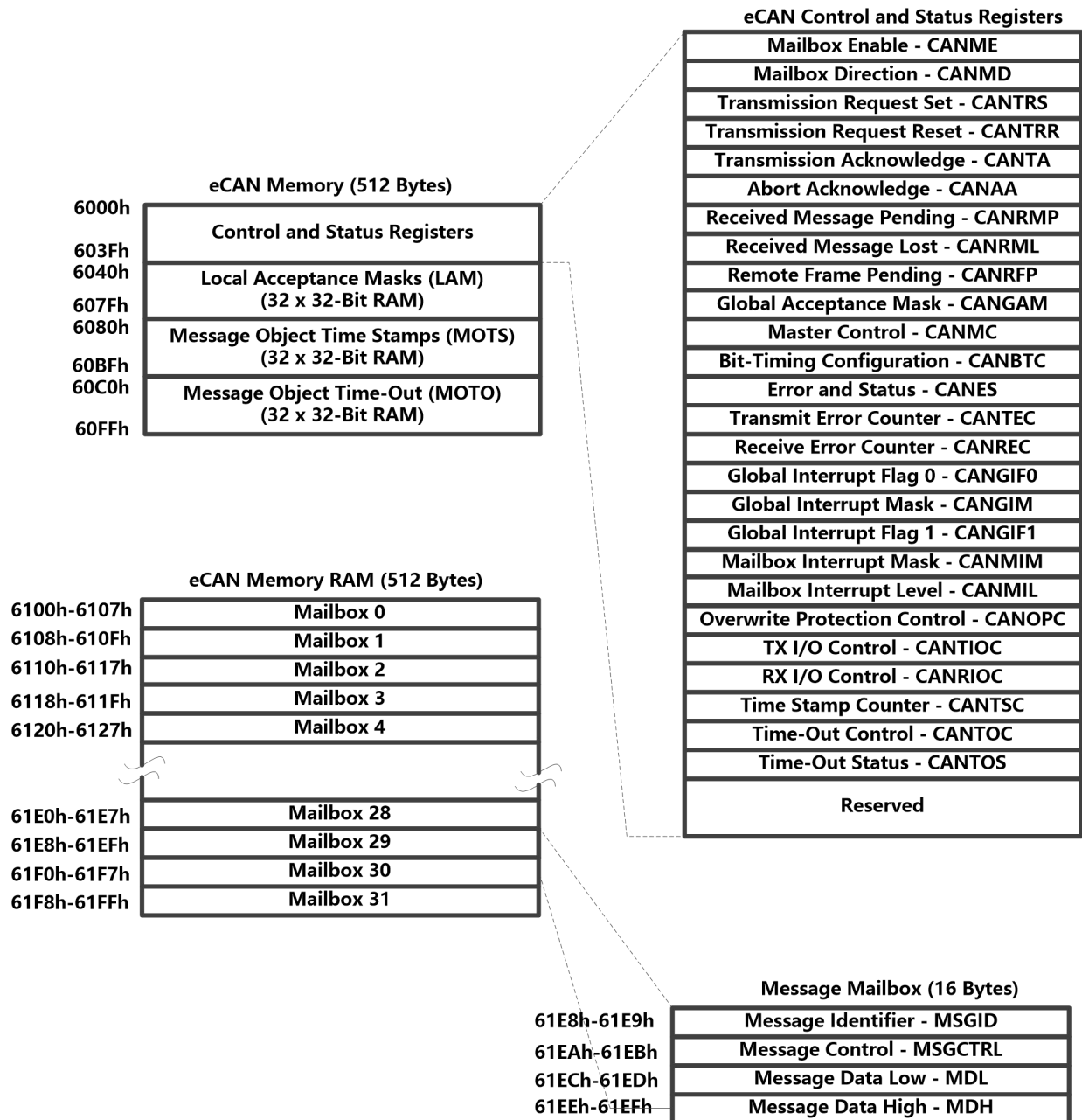


图4-26 eCAN内存映射

注

如果未在应用中使用 eCAN 模块，提供的 RAM（LAM、MOTS、MOTO 和邮箱 RAM）可用作通用 RAM。为实现这一功能，应启用 CAN 模块的时钟。

CPU 使用表 4-21 中列出的 CAN 寄存器来配置和控制 CAN 控制器及消息对象。这些寄存器映射至外设帧 1，并只支持 32 位读写操作。邮箱 RAM 支持 16 位或者 32 位访问，32 位访问采用偶边界对齐。

表4-21CAN 寄存器映射⁽¹⁾

名称	地址	大小 (x32)	说明
CANME	0x00 6000	1	邮箱启用
CANMD	0x00 6002	1	邮箱方向
CANTRS	0x00 6004	1	发送请求设定
CANTRR	0x00 6006	1	发送请求复位
CANTA	0x00 6008	1	传输确认
CANAA	0x00 600A	1	中止确认
CANRMP	0x00 600C	1	接收消息等待
CANRML	0x00 600E	1	接收消息丢失
CANRFP	0x00 6010	1	远程帧等待
CANGAM	0x00 6012	1	全局接收屏蔽
CANMC	0x00 6014	1	主器件控制
CANBTC	0x00 6016	1	位时序配置
CANES	0x00 6018	1	错误和状态
CANTEC	0x00 601A	1	发送错误计数器
CANREC	0x00 601C	1	接收错误计数器
CANGIF0	0x00 601E	1	全局中断标志0
CANGIM	0x00 6020	1	全局中断屏蔽
CANGIF1	0x00 6022	1	全局中断标志1
CANMIM	0x00 6024	1	邮箱中断屏蔽
CANMIL	0x00 6026	1	邮箱中断级别
CANOPC	0x00 6028	1	写覆盖保护控制
CANTIOC	0x00 602A	1	TX I/O 控制
CANRIOC	0x00 602C	1	RX I/O 控制
CANTSC	0x00 602E	1	时间戳计数器（保留在SCC 模式中）
CANTOC	0x00 6030	1	超时控制（保留在SCC 模式中）
CANTOS	0x00 6032	1	超时状态（保留在SCC 模式中）

4.7 可变速率控制局域网(CANFD)模块

4.7.1 简介

CANFD 模块可以配置为满足 CAN FD 和 CAN2.0B 的规范。CPU 通过 PF1 总线接口访问 CANFD 控制器，CANFD 模块通过 TX/RX 端口，在内部协议机制下控制帧的发送或接收。

CANFD 模块结构如图 4-27 所示。

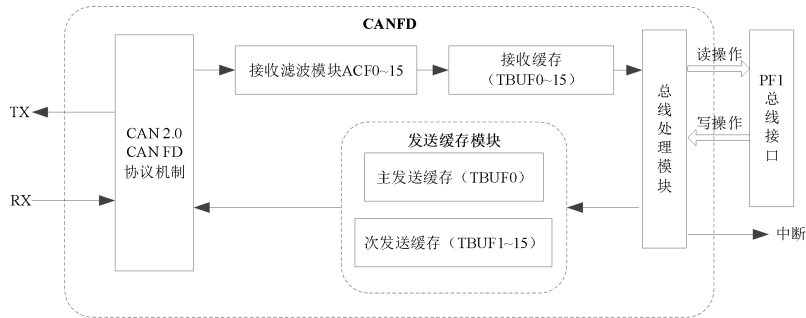


图 4-27 CANFD 控制器的结构框图

4.7.2 CANFD 协议说明

CAN FD 是 CAN 2.0 的扩展协议，二者的主要区别在于：

- (1) 数据传输负载：CAN 2.0 最高 8 字节，CAN FD 最高 64 字节；
- (2) CAN 2.0 有一个可配置的比特率，CAN FD 有两个：仲裁过程慢，数据传输快；

CAN 2.0 和 CAN FD 的所有类型的框架下图所示。

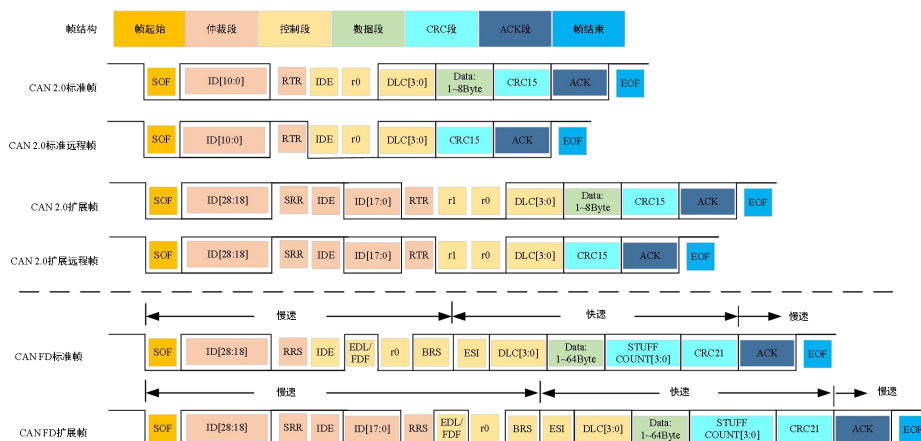


图 4-28 CAN 2.0 和 CAN FD 帧结构图

帧的位域术语描述如下表4-22。

表 4-22 帧位域术语描述

术语	描述
ID	帧 ID
RTR	标识 CAN 2.0 远程帧，高为远程帧，低为数据帧
SRR	标识 CANFD 远程帧，高为远程帧，低为数据帧
RRS	标识 CANFD 帧，保持为低
IDE	标识帧为扩展帧标识，高为扩展帧，低为标准帧
DLC	帧数据段的数据字节长度
EDL	标识 CANFD 格式，高为 CANFD 帧，低为 CAN 2.0 帧
FDF	标识 CANFD 格式，高为 CANFD 帧，低为 CAN 2.0 帧
BRS	比特率切换
ESI	错误状态标识
r0,r1	保留位，默认为低

4.7.3 CANFD 模块特性说明

- 支持 CAN 规范
 - CAN 2.0B(8 字节的有效荷载);
 - 可选支持 CAN FD (64 字节的有效荷载) ;
- 可编程数据传输速率
 - CAN 2.0B 1Mbit/s;
 - CAN FD 受收发器和 CAN 控制器时钟频率影响;
- 可编程带宽波特率分频(1~1/256);
- 主机接口和 CANFD 协议有独立时钟域;
- 可配置的接收缓冲区 (RB)
 - 通过参数选择缓冲区插槽数量;
 - 类似 FIFO 功能;
 - 接收到的“不被接收的信号”或者“不正确”的消息不会覆盖已存储的消息;
- 两个传输缓冲器
 - 一个主传输缓冲区(TBUF0);
 - 可选可配置的次要传输缓冲区 (TBUF1~15) ;
 - ✧ TBUF1~15 可选，通用参数可以选择缓冲区插槽的数量;
 - ✧ 在 FIFO 或优先级模式下的操作。

- 独立可编程的 29 位内部接收滤波器 (AMASK0~AMASK15) ;
- 扩展特征
 - 单次发送模式(发送给 TBUF);
 - 侦听模式;
 - 回环模式 (内部回环或外部回环);
 - 收发器待机模式;
- 扩展状态和错误报告
 - 捕获上次发生的错误类型和仲裁丢失的位置;
 - 可编程的错误警告限制;
- 可配置中断源;
- 用于帧缓冲器的一个双端口存储器块 (DPSRAM) ;
- CiA 603 时间戳
 - ISO 11898-4 带部分硬件支持的时间触发 CANFD;
- 具有存储保护的 ECC 校验;
- 与 AUTOSAR 兼容。

4.7.4 时钟配置

4.7.4.1 功能说明

CANFD 模块包含三种时钟和复位: can_clk 时钟、主机 host_clk 时钟、CIA 时间戳; 采用同一个复位系统, 模块内部为同步复位。

4.7.4.2 配置说明

CANFD 模块协议机的时钟源 can_clk, 对 host_clk、timer_clk 和复位无要求, 可以通过 CANFD 内部分频操作, 满足 CANFD 模块协议机要求。

host_clk 使能位: PCLKCR0[15], 0: 关闭时钟; 1: 打开时钟。

PLL 时钟的配置, 需要用到系统寄存器 PLLCR.DIV、PLLSTS.DIVSEL, 具体配置关系如下表。

表 4-23 PLL时钟配置

SYSCLKOUT (CLKIN)			
PLLCR.DIV	PLLSTS.DIVSEL=0 or 1	PLLSTS.DIVSEL=2	PLLSTS.DIVSEL=3
0000	OSCCLK/4	OSCCLK/2	OSCCLK/1
0001	(OSCCLK*1) /4	(OSCCLK*1) /2	(OSCCLK*1) /1
0010	(OSCCLK*2) /4	(OSCCLK*2) /2	(OSCCLK*2) /1
0011	(OSCCLK*3) /4	(OSCCLK*3) /2	(OSCCLK*3) /1
0100	(OSCCLK*4) /4	(OSCCLK*4) /2	(OSCCLK*4) /1
0101	(OSCCLK*5) /4	(OSCCLK*5) /2	(OSCCLK*5) /1
0110	(OSCCLK*6) /4	(OSCCLK*6) /2	(OSCCLK*6) /1
0111	(OSCCLK*7) /4	(OSCCLK*7) /2	(OSCCLK*7) /1
1000	(OSCCLK*8) /4	(OSCCLK*8) /2	(OSCCLK*8) /1
1001	(OSCCLK*9) /4	(OSCCLK*9) /2	(OSCCLK*9) /1
1010	(OSCCLK*10) /4	(OSCCLK*10) /2	(OSCCLK*10) /1
1011	(OSCCLK*11) /4	(OSCCLK*11) /2	(OSCCLK*11) /1
1100	(OSCCLK*12) /4	(OSCCLK*12) /2	(OSCCLK*12) /1
1101-1111	Reserved	Reserved	Reserved

时钟配置是系统控制寄存器具体配置流程如下。

- 配置 PLLSTS.DIVSEL 位，配置系统时钟的分频系数；
- 配置 PLLCR.DIV 位，PLL 时钟的倍频率值；
- 通过检查 PLLSTS.PLLLOCKS 位，判断 PLL 已完成锁定，时钟稳定。

4.7.5 引脚配置

4.7.5.1 配置说明

CANFD 模块输出引脚 TX、RX 给到 PAD 端的 GPIO 如下所示。

- TX 引脚与 GPIO29、GPIO32 复用；
- RX 引脚与 GPIO28、GPIO0 复用；
- 输出引脚 TX 空闲态为高电平；
- 输入引脚 RX 空闲态为高电平

具体配置如下；

- 配置 GPAPUD.GPIO29、GPAPUD.GPIO28 位，GPIO29、GPIO28、引脚上拉；
 - 配置 0x702B 地址为 0xA0，GPIO29、GPIO28 引脚 PINMUX 为 TX、RX 引脚；

- 或配置 GPAPUD.GPIO32、GPAPUD.GPIO0 位，GPIO32、GPIO0 引脚上拉；
 - 配置 0x702B 地址为 0x50，GPIO32、GPIO0 引脚 PINMUX 为 TX、RX 引脚；

4.7.6 中断配置

4.7.6.1 功能说明

CANFD 模块输出中断只有一个 CANFD_INT 中断，该中断由多种中断类型组合，中断类型分为传输状态、错误状态、仲裁丢失，具体说明如下。

- 传输状态中断类型如下；
 - 接收完的中断标志 RTINTFE.RIF；
 - 接收 RBUF 溢出的中断标志 RTINTFE.ROIF；
 - 接收 RBUF 满中的中断标志 RTINTFE.RFIF；
 - 接收 RBUF 几乎全满的中断标志 RTINTFE.RAFIF；
 - ✧ RBUF 深度为 15，通过配置 LIMIT_EINT.AFWL 确认 RBUF 几乎满的限制。
 - 完成发送主要缓存 TBUF0 区域帧的中断标志 RTINTFE.TPIF；
 - ✧ 对于主要缓存 TBUF0，如果启用使能 TPIE，则传输完成后 TPIF 被标识
 - 完成发送缓存 TBUF1~15 区域帧的中断标志 RTINTFE.TSIF；
 - ✧ 对于使用 TSONE 的缓存 TBUF1~15，如果完成一条报文传输并启用使能 TSIE，则设置中断标志 TSIF；
 - ✧ 对于使用 TSALL 的缓存 TBUF1~15，当所有报文传输完成 TBUF1~15 为空时设置 TSIE，并 TSIF 位被标识；
 - 传输过程中传输中止的中断标志 RTINTFE.AIF；
 - ✧ TBUF0 中请求但尚未启动的传输可以中止传输，未传输的报文仍保留在 TBUF0 中；
 - ✧ TBUF1~15 同样可以中止请求但尚未启动的传输，对于 TSONE 只有一帧中止，对于 TSALL 所有帧都中止；

- ◇ 只有在不向总线传输任何东西时才可以中止，总线仲裁期间、传输帧时不会被中止；

- 错误状态中断

- 传输（收发帧）过程产生错误，并且错误计数值大于设定的 LIMIT_EINT.EWL，则产生中断的中断标志 RTINTFE.EIF；
- 传输过程产生的错误是被动错误的标志 LIMIT_EINT.EPIF；
- 发送帧过程中仲裁段产生错误的标志中断 LIMIT_EINT.ALIF，并通过 ACL 指示仲裁丢失的位置；
- 总线错误中断标志 LIMIT_EINT.BEIF；
- 定时器触发中断标志 TTCFG_TBPtr.TTIF；
 - ◇ 配置的定时触发时间 TT_TRIG 寄存器等于周期时间，触发 CANFD 发送帧，则 TTIF 中断标志置 1；
- 定时触发错误中断标志 TTCFG_TBPtr.TEIF，产生触发错误的条件如下；
 - ◇ 立即模式触发，但是 TBUF 缓存为空；
 - ◇ 触发时间 TT_TRIG 设置小于周期时间；
 - ◇ 在触发类型为 2 或 3 的条件下，触发时间等于周期时，TBUF 为空；
 - ◇ 看门狗时间小于周期时间；
- 看门狗中断 TTCFG_TBPtr.WTIF；
 - ◇ 配置的 TT_WTRIG 寄存器等于周期时间；

4.7.6.2 配置说明

CANFD 模块的中断 CANFD_INT 映射到中断向量表中 INT4.2、INT8.3 中。

CANFD 中断的产生，以帧传输为例，具体如下：

- 对于主要传输

- 配置 CFG_STAT.TBSEL=0，选择主要缓存区 TBUF0，CFG_STAT.TPE=0；

- 配置 RTINTFE.TPIE=1, 设置中断使能位;
 - 将帧信息对应写入 TBUF 地址空间: 0x28~0x4b;
 - 设置 CFG_STAT.TPE=1, CFG_STAT.TSONE=0, CFG_STAT.TSALL=0, 启动发送;
 - 检测 RTINTFE.TPIF 位是否拉高, 拉高则传输完成, 并且产生传输完成中断。
- 对于次要传输
 - 配置 CFG_STAT.TBSEL=1, 选择次要缓冲区 TBUF1~15, CFG_STAT.TPE=0;
 - 设置中断使能位 RTINTFE.TSIE=1;
 - 将帧对应写入 TBUF 地址空间: 0x28~0x4b;
 - 设置 CFG_STAT.TPE=0, CFG_STAT.TSONE=0/CFG_STAT.TSALL=1,
或 CFG_STAT.TSONE=1/CFG_STAT.TSALL=0(二者不可同时为 1), 启动发送;
 - 若 TSONE=1 则完成一条报文传输后设置 TSIF, 若 TSALL=1 则 TBUF1~15 为空时设置 TSIF。

注意: CANFD中断类型是高电平有效, 如果进入中断服务程序没有全部清除, CANFD产生的新类型中断, 不会重新进入中断服务程序, 用户在进入中断服务程序时, 需要先缓存对应中断标志寄存器, 并立即写1清除所有标志位, 用户再决定响应缓存中断标志对应中断类型。

4.7.7 传输模式

4.7.7.1 功能说明

传输模式包括单次模式、回环模式、侦听模式和休眠模式。

单次模式下无论使用TBUF0还是TBUF1~15, 如果发生错误或仲裁丢失将不再执行重新传输。单次发送模式下, 如果对TBUF1~15使用TSALL全部传输, 则对TBUF1~15中所有帧进行单次传输, TBUF1~15为空则停止。

回环模式包括内部(LBMI)和外部(LBME)两种回环模式, 两种模式都能接收到自己发送的帧。LBMI模式下CANFD控制与CANFD总线断开连接, 可用于芯片内部测试, LBME与CANFD总线保持连接, 可测试收发器及其连接。

侦听模式提供了仅监测但不影响总线的功能。

待机模式下不可进行帧传输，节点可通过发送帧进行唤醒，如果所有接收节点都处于待机模式时，发送节点会接收到ACK错误。

4.7.7.2 配置说明

- 单次模式：

- 对于主要传输

- ✧ 首先配置 `CFG_STAT.TBSEL=0`，选择主要缓存区 `TBUF0`；
- ✧ 配置 `CFG_STAT.TPSS` 将 `TBUF0` 设置为单次模式；
- ✧ 将帧信息对应写入 `TBUF` 地址空间：0x28~0x4b；
- ✧ 设置 `CFG_STAT.TPE=1` 位启动发送；
- ✧ 检测 `RTINTFE.TPIF` 位是否拉高，拉高则传输完成，如果 `TPIE` 中断使能位为 1，产生传输完成中断。

- 对于次要传输：

- ✧ 配置 `CFG_STAT.TBSEL=1`，选择次要缓冲器 `TBUF1~15`；
- ✧ 配置 `CFG_STAT.TSSS` 将 `TBUF1~15` 设置为单次模式；
- ✧ 将帧信息写入对应的 `TBUF` 地址空间：0x28~0x4b；
- ✧ 设置 `CTRL.TSNEXT` 为 1，指向下一次将要传输的缓存；
- ✧ 设置 `CFG_STAT.TPE=0`，`CFG_STA.TSONE=1` 启动发送；
- ✧ 检测 `RTINTFE.TSIF` 位是否拉高，拉高则传输完成，并且产生传输完成中断；

- 回环模式：

- 对于外部回环

- ✧ 设置 `CFG_STAT.LBME=1` 进入外部回环模式，与侦听模式一起使用；

- 对于内部回环模式

- ✧ 设置 `CFG_STAT.LBMI=1` 进入内部回环模式，引脚 `TXD`（`GPIO29/GPIO32`）保

持为高;

- 侦听模式:

- 在 CFG_STAT.TPE=0、CFG_STAT.TSOME=0、CFG_STAT.TSALL=0 条件下, 设置 CFG_STAT.LOM=1, 启用侦听模式;
- ✧ CFG_STAT.LBME=0, 将禁用所有传输, 引脚 TXD (GPIO29/GPIO32) 保持为高;
- ✧ CFG_STAT.LBME=1, 为接收到的帧或错误帧禁用 ACK, 可以传输自己的帧。

- 待机模式:

- 设置 CFG_STAT.TPE=0、CFG_STAT.TSOME=0、CFG_STAT.TSALL=0 的条件下, 设置 CFG_STAT.STBY=1, 进入收发器待机模式;
- 进入待机模式后, 唤醒有两种情况, 一种休眠模式下的 CANFD 接收到其他帧, 另一种是手动清除 STBY, 一旦唤醒则需要等待 TRAN 寄存器模块的计数时间, CANFD 才能启动发送, 并且可以通过 CANFD_STBY 寄存查看是否唤醒。

4.7.8 帧类型

4.7.8.1 功能说明

对于 CAN 2.0 包括四种帧类型: 标准帧、远程帧、扩展帧和扩展远程帧。标准帧 ID 为 11 位, 扩展帧 ID 为 29 位。CAN 2.0 最大携带 8 字节数据, 远程帧不携带数据。

CAN FD 包括两种帧类型: 标准帧和扩展帧; CAN FD 是 CAN 2.0 的扩展协议, 二者主要区别在于, CAN 2.0 最高携带 8 字节, 有一个可配置的比特率; CAN FD 最高 64 字节, 有两个可配置的比特率。

4.7.8.2 配置说明

- CAN 2.0 标准帧

- 配置帧 ID;
- 配置帧 IDE 位=0, 标识当前帧非扩展帧;
- 配置帧 RTR 位=0, 标识当前帧非远程帧;

- 配置帧 FDF 位=0, 标识当前帧为 CAN 2.0 帧;
 - 配置帧 DLC, 标识当前帧数据长度;
 - 配置帧数据;
 - 将帧信息对应写入 TBUF 地址空间: 0x28~0x4b;
 - 帧信息写入 TBUF0 缓存则, 启动 TBUF0 传输:
 - ✧ 设置 CFG_STAT.TPE=1 启动发送;
 - ✧ 检测 RTINTFE.TPIF 位是否拉高, 拉高则传输完成, 并且产生传输完成中断。
 - 或启动 TBUF1~15 传输:
 - ✧ 设置 CFG_STAT.TSNEXT=1 指向下一次写入缓存的 TBUF
 - ✧ 设置 CFG_STAT.TSALL=1 启动发送
 - ✧ 检测 RTINTFE.TSIF 位是否拉高, 拉高则传输完成, 并且产生传输完成中断。
- CAN 2.0 远程帧
 - 配置帧 ID;
 - 配置帧 IDE 位=0, 标识当前帧非扩展帧;
 - 配置帧 RTR 位=1, 标识当前帧为远程帧;
 - 配置帧 FDF 位=0, 标识当前帧为 CAN 2.0 帧, CAN FD 中没有远程帧;
 - 配置帧 DLC, 远程帧不携带数据;
 - 将帧信息对应写入 TBUF 地址空间: 0x28~0x4b;
 - 帧信息写入 TBUF0 缓存则, 启动 TBUF0 传输:
 - ✧ 设置 CFG_STAT.TPE=1 启动发送;
 - ✧ 检测 RTINTFE.TPIF 位是否拉高, 拉高则传输完成, 并且产生传输完成中断。
 - 或启动 TBUF1~15 传输:
 - ✧ 设置 CFG_STAT.TSNEXT=1 指向下一次写入缓存的 TBUF
 - ✧ 设置 CFG_STAT.TSALL=1 启动发送

◇ 检测 RTINTFE.TSIF 位是否拉高，拉高则传输完成，并且产生传输完成中断。

- CAN 2.0 扩展帧

- 配置帧 ID;
- 配置帧 IDE 位=1，标识当前帧为扩展帧;
- 配置帧 RTR 位=0，标识当前帧非远程帧;
- 配置帧 FDF 位=0，标识当前帧为 CAN 2.0 帧;
- 配置帧 DLC，标识当前帧数据长度;
- 配置帧数据;
- 将帧信息对应写入 TBUF 地址空间：0x28~0x4b;
- 帧信息写入 TBUF0 缓存则，启动 TBUF0 传输：
 - ◇ 设置 CFG_STAT.TPE=1 启动发送;
 - ◇ 检测 RTINTFE.TPIF 位是否拉高，拉高则传输完成，并且产生传输完成中断。
- 或启动 TBUF1~15 传输：
 - ◇ 设置 CFG_STAT.TSNEXT=1 指向下一次写入缓存的 TBUF
 - ◇ 设置 CFG_STAT.TSALL=1 启动发送
 - ◇ 检测 RTINTFE.TSIF 位是否拉高，拉高则传输完成，并且产生传输完成中断。

- CAN 2.0 扩展远程帧

- 配置帧 ID;
- 配置帧 IDE 位=1，标识当前帧为扩展帧;
- 配置帧 RTR 位=1，标识当前帧为远程帧;
- 配置帧 FDF 位=0，标识当前帧为 CAN 2.0 帧，CAN FD 中没有远程帧;
- 配置帧 DLC，远程帧不携带数据;
- 将帧信息对应写入 TBUF 地址空间：0x28~0x4b;
- 帧信息写入 TBUF0 缓存则，启动 TBUF0 传输:

- ◇ 设置 CFG_STAT.TPE=1 启动发送;
- ◇ 检测 RTINTFE.TPIF 位是否拉高, 拉高则传输完成, 并且产生传输完成中断。
- 或启动 TBUF1~15 传输:
 - ◇ 设置 CFG_STAT.TSNEXT=1 指向下一次写入缓存的 TBUF
 - ◇ 设置 CFG_STAT.TSALL=1 启动发送
 - ◇ 检测 RTINTFE.TSIF 位是否拉高, 拉高则传输完成, 并且产生传输完成中断。
- CAN FD 标准帧
 - 配置帧 ID;
 - 配置帧 IDE 位=0, 标识当前帧非扩展帧;
 - 配置帧 RTR 位=0, 标识当前帧非远程帧, CAN FD 中没有远程帧;
 - 配置帧 DLC, 标识当前帧数据长度;
 - 将帧信息对应写入 TBUF 地址空间: 0x28~0x4b;
 - 帧信息写入 TBUF0 缓存则, 启动 TBUF0 传输:
 - ◇ 设置 CFG_STAT.TPE=1 启动发送;
 - ◇ 检测 RTINTFE.TPIF 位是否拉高, 拉高则传输完成, 并且产生传输完成中断。
 - 或启动 TBUF1~15 传输:
 - ◇ 设置 CFG_STAT.TSNEXT=1 指向下一次写入缓存的 TBUF
 - ◇ 设置 CFG_STAT.TSALL=1 启动发送
 - ◇ 检测 RTINTFE.TSIF 位是否拉高, 拉高则传输完成, 并且产生传输完成中断。
- CAN FD 扩展帧
 - 配置帧 ID;
 - 配置帧 IDE 位=1, 标识当前帧为扩展帧;
 - 配置帧 RTR 位=0, 标识当前帧非远程帧, CAN FD 中没有远程帧;
 - 配置帧 DLC, 标识当前帧数据长度;

- 将帧信息对应写入 TBUF 地址空间：0x28~0x4b;
- 帧信息写入 TBUF0 缓存则，启动 TBUF0 传输：
 - ✧ 设置 CFG_STAT.TPE=1 启动发送;
 - ✧ 检测 RTINTFE.TPIF 位是否拉高，拉高则传输完成，并且产生传输完成中断。
- 或启动 TBUF1~15 传输：
 - ✧ 设置 CFG_STAT.TSNEXT=1 指向下一次写入缓存的 TBUF
 - ✧ 设置 CFG_STAT.TSALL=1 启动发送
 - ✧ 检测 RTINTFE.TSIF 位是否拉高，拉高则传输完成，并且产生传输完成中断。

4.7.9 发送帧

4.7.9.1 功能说明

发送帧有 2 种发送方式，一种通过配置 TBUF 传输的方式发送；另一种通过 TTCAN 定时触发的方式发送。

4.7.9.1.1 配置 TBUF 发送

TBUF 缓存分为两个传输缓冲区(TB)。主要缓存(TBUF0)具有更高的优先级，但只能缓存一个帧。缓存(TBUF1~15)的优先级较低。它可以在 FIFO 或优先模式下工作。TBUF0 和 TBUF1~15 之间的优先级决定是固定的，完全独立于 CANFD 总线仲裁。总线仲裁是基于帧的 ID 优先级决定。

可以命令 TBUF1~15 传输一个或所有存储的帧。在 FIFO 模式下，每次传输时，缓冲区内先进缓存 TBUF1~15 的帧先传输。在优先级模式下，这个缓冲区内优先级最高的帧首先被传输。

不管帧 ID 是什么，位于 TBUF0 中的帧对于 CANFD 协议机来说总是比 TBUF1~15 中的帧具有更高的优先级。TBUF0 传输停止并延迟 TBUF1~15 传输。TBUF0 帧传输成功后，会自动重启 TBUF1~15 传输。高优先级的帧不会中断正在传输的帧。

使用 TBUF0 传输中断 TBUF1~15 传输可能发生在以下情况：

- 如果 TBUF1~15 输出所有存储的帧，协议机决定在 TBUF1~15 传输完成之前命令 TBUF0 传输；
- 如果 TBUF1~15 输出单帧，在 TBUF1~15 传输完成之前，协议机决定命令 TBUF0 传输；

- 如果无等待的传输帧，协议机无需决定 TBUF 的传输顺序；
- 具有相同优先级的帧写入 TBUF1~15，那么最先写入的帧先被发送。

4.7.9.1.2 传输中止

通过设置 TPA 或者 TSA 来，中止还未启动的 TBUF 传输，通过这种方式避免了高优先级消失长时间占用，而撤销高优先级消失的传输。

传输中止，只能中止还未传输的帧，并不是还未传输完成的帧，具体描述如下：

- 如果帧传输完成，或者正常传输完成，则不会中止传输；
- CANFD 节点未接收到确认的传输失败后，错误计数器收到新的错误信号，则执行中止；
- 如果 TBUF1~15 中还剩下至少一个帧，而主机已经配置发送所有帧（TSALL = 1），则向主机发送完成的帧以及中止。
- 配置 TPA，中止 TBUF0 传输，但是帧数据任然在 TBUF0 中，并且 TPE=0。
- 配置 TSA，中止 TBUF1~15 传输，对应的 TBUF1~15 的缓存会更新，导致 TBUF1~15 丢失；

4.7.9.1.3 定时触发

定时触发 TTCAN 是 ISO 11898-4 的一种操作模式，其中帧将仅在预定义的时间窗口传输，对应触发类型如下。

- TTYPE_IMME 类型，立即发送，默认采用这种类型；
- TTYPE_TIME 类型，不指定槽触发，采用 TBUF 定时触发；
- TTYPE_TSS 类型，定时触发；
- TTYPE_TSTART 类型，仲裁时间窗口(几个节点可能会发送一个帧并进行仲裁)；
- TTYPE_TSTOP 类型，用于合并仲裁时间窗口的传输停止触发器。

定时触发式在 TTCAN 模式下，TBUF 无优先级，每个 TBUF 的槽都可以由主机指定传输，每个槽都可以定义为已填充或者空。

主机可以将消息指定任意一个缓冲槽进行收发，帧将仅在预定义的时间窗口传输，触发的时间来自协议时钟，触发时间的计数时钟计算如下：

$$\frac{f_{can_clk(Mhz)}}{(2^{T_{PRSEC}}) \times (s_presc + 1) \times ((S_SEG1 + 2) + (S_SEG2 + 1))}$$

定时触发模式下，帧将在预定的时间窗口传输，每次参考信息配置 REF_MSG_0~REF_MSG_3 寄存器中 ID 和 IDE 位，与发送帧 ID 匹配时，更新窗口时间，每次窗口时间会导致周期时间变换，而触发时间与周期时间有关，具体时序描述如下图。

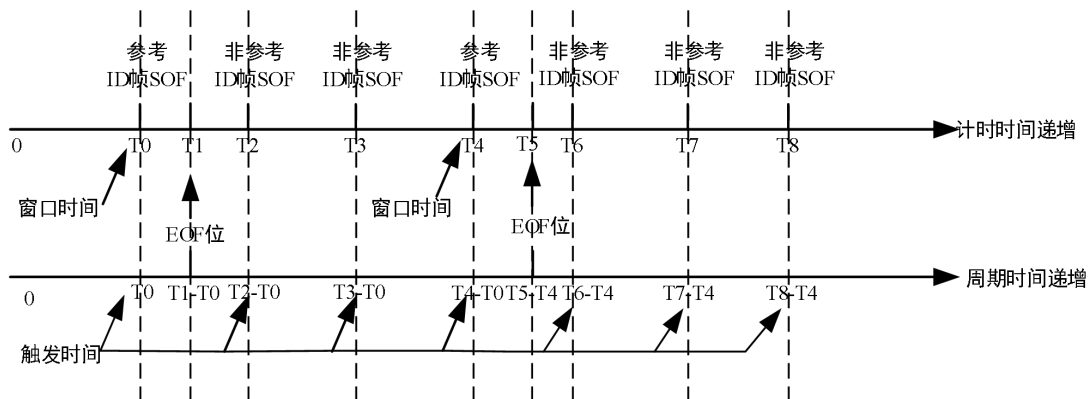


图 4-29 定时触发时序

图中每帧数据的发送前，都会与参考 ID 比较，匹配则为参考 ID 的帧，具体说明如下。

- 每次窗口时间更新，都是 ID 匹配时 SOF 所在的计数时间值；
- 每次周期时间更新，都是 ID 匹配的 SOF 计数时间值减当前周期时间值；
- 触发时间就是帧启动 SOF 位的周期时间。

对应触发类型特性如下。

- TTYPE_IMME 类型特性
 - 启动传输：发送缓存 TBUF 非空，则立即触发，无须等待配置的触发时间；
 - 产生触发错误使能情况下，标识 TEIF 位；
 - ✧ 触发时间小于 (<) 周期时间，则产生触发错误；
 - ✧ 发送缓存 TBUF 为空；
 - 在触发时间等于周期时间时，使能情况下，标识 TTIF 位；
- TTYPE_TIME 类型，空闲窗口时间；
- TTYPE_TSS 类型特性

- 启动传输：更新触发时间小于（<）周期时间
 - ✧ 标识 TSSS 位；
 - ✧ 启动传输，但是还未触发；
 - ✧ 等待触发时间等于（=）周期时间，触发传输，使能情况下，标识 TTIF 位；
- 产生触发错误，使能情况下，标识 TEIF 位
 - ✧ 更新触发时间小于（<）周期时间；
 - ✧ 在触发时间等于周期时间的条件下，发送缓存 TBUF 为空；
- 传输停止
 - ✧ 更新触发时间+单次触发的数量 $TEW+1=$ 周期时间，并且在帧还未触发的条件下，停止帧传输。
- TTYPE_TSTART 类型特性
 - 启动传输：配置触发时间小于（<）周期时间
 - ✧ 标识 TSSS 位；
 - ✧ 启动传输，但是还未触发；
 - ✧ 等待触发时间等于（=）周期时间，触发传输，使能情况下，标识 TTIF 位；
 - 产生错误触发，使能情况下，标识 TEIF 位
 - ✧ 更新触发时间小于（<）周期时间；
 - ✧ 在触发时间等于（=）周期时间的条件下，TBUF1~15 为空；
- TTYPE_TSTOP 类型特性
 - 传输停止（tt_tsa_i）：
 - ✧ 更新触发时间等于（=）周期时间；
 - ✧ 触发时间等于周期时间；
 - 产生错误触发，使能情况下，标识 TEIF 位
 - ✧ 更新触发时间小于（<）周期时间；

4.7.9.1.4 看门狗计时

通过配置 TT_WTRIG 寄存器，具体特性说明如下。

- 更新看门狗时间小于（<）周期时间，产生错误触发 TEIF；
- 更新看门狗时间等于（=）周期时间，产生 WTIF 标识；

看门狗时间用于监控距离上次窗口时间的时间是否过长，防止设置的触发时间过短，而导致停止触发。

4.7.9.1.5 位时序

CAN 2.0B 定义了高达 1Mbit/s 的数据比特率。对于 CAN FD 没有固定的限制，可以通过位时序寄存器（SEG1、SEG2、SJW、presc）配置调整数据的比特率。CANFD 比特时间 BT 由几个时间段组成，如图 4-30。

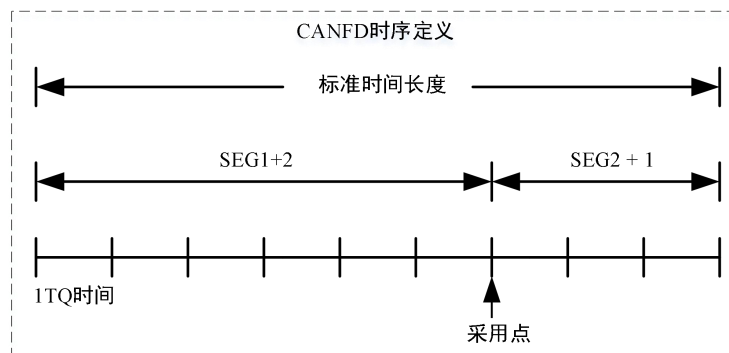


图 4-30 位采样中时间分隔

波特率 Mbit/s 计算方法：

$$\frac{f_{can_clk(Mhz)}}{(SEG1 + 2 + SEG2 + 1) \times (presc + 1)}$$

CANFD 的位时序配置范围如下表所示，并检查是否满足所有规则，但是提供比 CAN2.0 规范所定义的更广泛的配置范围。

表 4-24 CANFD 时序分隔（有效配置范围）

时间段	TQ 时间范围	描述
tSEG1	2~65TQ	CAN 2.0 比特率（慢）
	2~65TQ	CAN FD 标准比特率（慢）
	2~17TQ	CAN FD 数据比特率（快）
tSEG2	1~8TQ	tSEG1 >= tSEG2 + 2 CAN 2.0 比特率（慢）
	1~32TQ	tSEG1 >= tSEG2 + 2 CAN FD 标准比特率（慢）
	1~8TQ	tSEG1 >= tSEG2 + 1 CAN FD 数据比特率（快）
tSJW	1~16TQ	tSEG2 >= tSJW CAN 2.0 比特率（慢）

	1~16TQ	tSEG2 >= tSJW CAN FD标准比特率（慢）
	1~8TQ	tSEG2 >= tSJW CAN FD数据比特率（快）

对于慢 CAN 2.0 比特率和 CANFD 标准比特率（慢），通过设置 S_SEG1、S_SEG2、S_SJW 寄存器定义适当的长度，对于 CANFD（快）数据比特率，通过设置 F_SEG1、F_SEG2、F_SJW 寄存器都有效，具体配置如下表。

表 4-25 CANFD比特率配置

配置	慢	快
tSEG1	(S_SEG1 + 2)*TQ	(F_SEG1 + 2)*TQ
tSEG2	(S_SEG2 + 1)*TQ	(F_SEG2 + 1)*TQ
tSJW	(S_SJW + 1)*TQ	(F_SJW + 1)*TQ
nprescaler	S_PRESC+1	F_PRESC+1

CAN FD 在 BRS 位的采样点从慢标称比特率切换到快速度比特率，并在 CRC 分隔符位的采样点切回。

一个合适的比特率配置，时间量子单元 TQ， $ftq_clk=fcan_clk/presc$ ，比特时间 BT_TIME，采样点和同步跳转宽度 SJW 将都由 TQ 表示，如下图所示。

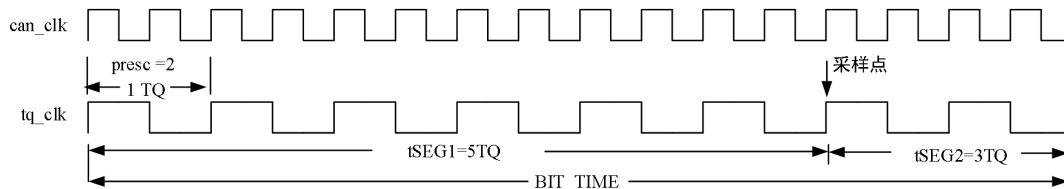


图 4-31 位采样中时间分隔

一些配置 SEG1、SEG2、SJW 的建议：

- 采样点在比特时间的中间稍微晚点，SEG1 略大于 SEG2；
- 同步跳转 SJW 宽度不能大于 SEG2，如果 SJW 太小，那么 CANFD 节点可能会太慢，而无法同步；如果 SJW 太大，那么 CANFD 节点可能会经常同步，建议 SJW 是 SEG2 的一半长。

CANFD 总线上的节点应该都设置成类似。

4.7.9.2 配置说明

- TBUF0 传输
 - 配置 CFG_STAT.TBSEL=0，选择主要缓存区 TBUF0，CFG_STATD.TPE=0；

- 配置 RTINTFE.TPIE=1, 设置中断使能位;
 - 将帧信息对应写入 TBUF 地址空间: 0x28~0x4b;
 - 设置 CFG_STAT.TPE=1, CFG_STAT.TSONE=0, CFG_STAT.TSALL=0, 启动发送;
 - 检测 RTINTFE.TPIF 位是否拉高, 拉高则传输完成, 并且产生传输完成中断。
- TBUF1~15 FIFO 模式
 - 配置 CTRL.TSMODE=0, 设置 TBUF1~15 为 FIFO 模式;
 - 配置 CFG_STAT.TBSEL=1, 选择次要缓冲区 TBUF1~15, CFG_STAT.TPE=0;
 - 设置中断使能位 RTINTFE.TSIE=1;
 - 配置 CFG_STAT.TSALL 传输所有辅助帧:
 - ✧ 将帧对应写入 TBUF 地址空间: 0x28~0x4b;
 - ✧ 设置 CFG_STAT.TSNEXT=1 指向下一次写入缓存的 TBUF1~15;
 - ✧ 可以循环多次将帧写入 TBUF 中, 并且写完成后, 配置 CFG_STAT.TSNEXT=1
 - ✧ 设置 CFG_STAT.TPE=0, CFG_STAT.TSONE=0, CFG_STAT.TSALL=1, 启动发送;
 - ✧ TBUF1~15 全部传输完成, 为空时, 设置产生传输完成中断标识 TSIF。
 - 配置 CFG_STAT.TSONE 传输辅助帧:
 - ✧ 将帧对应写入 TBUF 地址空间: 0x28~0x4b;
 - ✧ 设置 CFG_STAT.TSNEXT=1 指向下一次写入缓存的 TBUF1~15;
 - ✧ 设置 CFG_STAT.TPE=0, CFG_STAT.TSONE=1, CFG_STAT.TSALL=0, 启动发送;
 - ✧ 若 TSONE=1 则完成一条报文传输后设置 TSIF。
 - TBUF1~15 优先级模式
 - 配置 CTRL.TSMODE=1, 设置 TBUF1~15 为优先级模式;
 - 配置 CFG_STAT.TBSEL=1, 选择次要缓冲区 TBUF1~15, CFG_STAT.TPE=0;

- 设置中断使能位 RTINTFE.TSIE=1;
- 将帧对应写入 TBUF 地址空间: 0x28~0x4b;
- 设置 CFG_STAT.TSNEXT=1 指向下一次写入缓存的 TBUF1~15;
- 设置 CFG_STAT.TPE=0, CFG_STAT.TSONE=0, CFG_STAT.TSALL=1, 启动发送;
- 若 TSONE=1 则完成一条报文传输后设置 TSIF, 若 TSALL=1 则 TBUF1~15 为空时设置 TSIF。

- TBUF0 和 TBUF1~15 同时使用

- 配置 CTRL.TSMODE=0, 设置 TBUF1~15 为优先级模式;
- 配置 CFG_STAT.TBSEL=0, 选择主要缓冲区 TBUF0;
- 配置 TPE=0, TSONE=0, TSALL=0;
- 将帧对应写入 TBUF 地址空间;
- 配置 CFG_STAT.TBSEL=1, 选择次要缓冲区 TBUF1~15;
- 将帧对应写入 TBUF 地址空间;
- 配置 TPE=0, TSONE=0, TSALL=1, 对 TBUF1~15 中帧进行全部传输;
- 未传输完成, 启动 TBUF0 传输, 此时 TBUF0 优先传输。

- TTCAN 模式

定时触发模式下帧的触发类型分为:

- 类型一: 对于立即传输的立即触发器;
 - ✧ 配置 CTRL.TTTBM=1/0, 设置帧信息存储方式;
 - ✧ 配置 TTCFG_TBPTR.TTEN=1, 启动定时触发模式;
 - ✧ 配置 TTCFG_TBPTR.TTIF=1, 清除触发中断标志;
 - ✧ 配置 TTCFG_TBPTR.TEIF=1, 清除触发错误中断标志;
 - ✧ 配置 TTCFG_TBPTR.WTIF=1, 清除看门狗中断标志;
 - ✧ 配置触发类型 TRIG_CFG.TTYPE=0;

- ◇ 在配置 CTRL.TTTBM=1 的条件下:
 - ✓ 配置 TTCFG_TBPTR.TBPTR 为, 指定 TBUF0~15 中某个槽做写入槽;
 - ✓ 将帧信息写入 TBUF 地址空间: 0x28~0x4b, 将 TBPTR 指向的槽填充, 并且配置 TTCFG_TBPTR.TBF 标识该槽为填充;
 - ✓ 配置 TRIG_CFG.TTPTR, 指定 TBUF0~15 中某个槽做发送;
 - ✓ 配置 TT_TRIG, 指定发送的时间, 不过该为立即触发模式, 所以该位配置无效, 但是如果配置时间小于周期时间会产生触发错误中断;
 - ✓ 检测 RTINTFE.TSIF 位是否拉高, 拉高则传输完成, 并且产生传输完成中断。
- ◇ 在配置 CTRL.TTTBM=0 的条件下:
 - ✓ 配置触发类型 TRIG_CFG.TTYPE;
 - ✓ 配置 TT_TRIG, 指定发送的时间, 不过该为立即触发模式, 所以该位配置无效, 但是如果配置时间小于周期时间会产生触发错误中断;
 - ✓ 配置发送类型是 TBUF0 发送、或 TBUF1~15 发送, 按照对应的发送模式配置;
- 类型二: 对接收触发器的时间触发器;
 - ◇ 配置 CTRL.TTTBM=1/0, 设置帧信息存储方式;
 - ◇ 配置 TTCFG_TBPTR.TTEN=1, 启动定时触发模式;
 - ◇ 配置 TTCFG_TBPTR.TTIF=1, 清除触发中断标志;
 - ◇ 配置 TTCFG_TBPTR.TEIF=1, 清除触发错误中断标志;
 - ◇ 配置 TTCFG_TBPTR.WTIF=1, 清除看门狗中断标志;
 - ◇ 配置触发类型 TRIG_CFG.TTYPE=1;
 - ◇ 只在配置 CTRL.TTTBM=0 的条件下有效:
 - ✓ 配置触发类型 TRIG_CFG.TTYPE;
 - ✓ 配置 TT_TRIG, 指定发送的时间, 指定时间触发;
 - ✓ 配置发送类型是 TBUF0 发送、或 TBUF1~15 发送, 按照对应的发送模式配置;

- 类型三：对于用于独占时间窗口的单次发射触发器；
 - ✧ 配置 CTRL.TTTBM=1/0，设置帧信息存储方式；
 - ✧ 配置 TTCFG_TBPtr.TTEN=1，启动定时触发模式；
 - ✧ 配置 TTCFG_TBPtr.TTIF=1，清除触发中断标志；
 - ✧ 配置 TTCFG_TBPtr.TEIF=1，清除触发错误中断标志；
 - ✧ 配置 TTCFG_TBPtr.WTIF=1，清除看门狗中断标志；
 - ✧ 配置触发类型 TRIG_CFG.TTYPE=2；
 - ✧ 配置 TRIG_CFG.TEW=0~15，配置触发发送的窗口延迟时间；
 - ✧ 在配置 CTRL.TTTBM=1 的条件下：
 - ✓ 配置 TTCFG_TBPtr.TBPtr 为，指定 TBUF0~15 中某个槽做写入槽；
 - ✓ 将帧信息写入 TBUF 地址空间：0x28~0x4b，将 TBPtr 指向的槽填充，并且配置 TTCFG_TBPtr.TBF 标识该槽为填充；
 - ✓ 配置 TRIG_CFG.TTPtr，指定 TBUF0~15 中某个槽做发送；
 - ✓ 配置 TT_TRIG，指定发送的时间，如果在 TT_TRIG~ (TT_TRIG+TEW+16) 时间窗口内未发送，则停止发送，检测 RTINTFE.AIF 位是否拉高，拉高则传输停止；
 - ✓ 检测 RTINTFE.TSIF 位是否拉高，拉高则传输完成，并且产生传输完成中断。
 - ✧ 在配置 CTRL.TTTBM=0 的条件下：
 - ✓ 配置触发类型 TRIG_CFG.TTYPE；
 - ✓ 配置 TT_TRIG，指定发送的时间，不过该位配置无效，但是如果配置时间小于周期时间会产生触发错误中断；
 - ✓ 配置发送类型是 TBUF0 发送、或 TBUF1~15 发送，按照对应的发送模式配置。
- 类型四：合并仲裁时间窗口的传输启动触发器；
 - ✧ 配置 CTRL.TTTBM=1/0，设置帧信息存储方式；

- ✧ 配置 TTCFG_TBPtr.TTEN=1, 启动定时触发模式;
- ✧ 配置 TTCFG_TBPtr.TTIF=1, 清除触发中断标志;
- ✧ 配置 TTCFG_TBPtr.TEIF=1, 清除触发错误中断标志;
- ✧ 配置 TTCFG_TBPtr.WTIF=1, 清除看门狗中断标志;
- ✧ 配置触发类型 TRIG_CFG.TTYPE=3;
- ✧ 在配置 CTRL.TTTBM=1 的条件下:
 - ✓ 配置 TTCFG_TBPtr.TBPtr 为, 指定 TBUF0~15 中某个槽做写入槽;
 - ✓ 将帧信息写入 TBUF 地址空间: 0x28~0x4b, 将 TBPtr 指向的槽填充, 并且配置 TTCFG_TBPtr.TBF 标识该槽为填充;
 - ✓ 配置 TRIG_CFG.TTPtr, 指定 TBUF0~15 中某个槽做发送;
 - ✓ 配置 TT_TRIG, 指定发送的时间;
 - ✓ 检测 RTINTFE.TSIF 位是否拉高, 拉高则传输完成, 并且产生传输完成中断。
- ✧ 在配置 CTRL.TTTBM=0 的条件下:
 - ✓ 配置触发类型 TRIG_CFG.TTYPE;
 - ✓ 配置 TT_TRIG, 指定发送的时间, 不过该位配置无效, 但是如果配置时间小于周期时间会产生触发错误中断;
 - ✓ 配置发送类型是 TBUF0 发送、或 TBUF1~15 发送, 按照对应的发送模式配置;
- 类型五: 用于合并仲裁时间窗口的传输停止触发器。
 - ✧ 配置 CTRL.TTTBM=1/0, 设置帧信息存储方式;
 - ✧ 配置 TTCFG_TBPtr.TTEN=1, 启动定时触发模式;
 - ✧ 配置 TTCFG_TBPtr.TTIF=1, 清除触发中断标志;
 - ✧ 配置 TTCFG_TBPtr.TEIF=1, 清除触发错误中断标志;
 - ✧ 配置 TTCFG_TBPtr.WTIF=1, 清除看门狗中断标志;
 - ✧ 配置触发类型 TRIG_CFG.TTYPE=4;

- ◇ 在配置 CTRL.TTTBM=1 的条件下：
 - ✓ 配置 TTCFG_TBPTR.TBPTR 为，指定 TBUF0~15 中某个槽做写入槽；
 - ✓ 将帧信息写入 TBUF 地址空间：0x28~0x4b，将 TBPTR 指向的槽填充，并且配置 TTCFG_TBPTR.TBF 标识该槽为填充；
 - ✓ 配置 TRIG_CFG.TTPTR，指定 TBUF0~15 中某个槽做发送；
 - ✓ 配置 TT_TRIG，指定停止发送时间，周期时间等于触发时间 TT_TRIG，检测 TTCFG_TBPTR.TEIF 位是否拉，拉高表示则到达周期时间；再检测 RTINTFE.AIF 位是否拉高，拉高则传输停止；
- ◇ 在配置 CTRL.TTTBM=0 的条件下：
 - ✓ 配置触发类型 TRIG_CFG.TTYPE；
 - ✓ 配置 TT_TRIG，指定发送的时间，不过该位配置无效，但是如果配置时间小于周期时间会产生触发错误中断；
 - ✓ 配置发送类型是 TBUF0 发送、或 TBUF1~15 发送，按照对应的发送模式配置。

4.7.10 CIA603 时间戳

4.7.10.1 功能说明

在 603 规范中定义了 CANFD 的自动化（CIA）使用至少 16 位的时间戳方法，CIA603 的时间戳时一个外部自由运行的 64 位计时器。

时间戳产生事件请求、事件类型 SOF 或者 EOF、定时计数值，在定时器时钟、CANFD 协议时钟、主机时钟之间切换，最终产生的定时值分别存存在帧的 RTS、TTS 寄存器中，主机可以读取。

4.7.10.2 配置说明

时间戳配置类型和说明如下：

- 类型一：时间戳位置为 SOF
 - 配置 CIA_ACF_CFG.TIMEPOS=0，设置时间戳位置为 SOF；
 - 配置 CIA_ACF_CFG.TIMEEN=1，设置时间戳使能；

- 配置发送类型是 TBUF0 发送、或 TBUF1~15 发送，按照对应的发送模式配置。
- 类型二：时间戳位置为 EOF
 - 配置 CIA_ACF_CFG.TIMEPOS=1，设置时间戳位置为 EOF；
 - 配置 CIA_ACF_CFG.TIMEEN=1，设置时间戳使能；
 - 配置发送类型是 TBUF0 发送、或 TBUF1~15 发送，按照对应的发送模式配置。

时间戳是具有一个对时钟周期计数的定时器，在 SOF 或 EOF 处获取时间戳并存储在 RTS 或 TTS 中。

4.7.11 接收帧和滤波

4.7.11.1 功能说明

在 CANFD 网络中，只有一个节点可以传输具有特定标识符(ID)的报文。所有节点接收报文，节点主控制器必须决定它是否由适当的消息标识符寻址。为了减少主控制器的负载，在接收帧的过程中，通过使用接收滤波器，检查报文的 ID 是否匹配滤波 ID，每个接收滤波器的长度为 29 位，一共有 16 个滤波器。

CANFD 帧的标识符也用于 CANFD 总线仲裁。当具有较高优先级标识符的报文由另一个 CANFD 节点传输时，CANFD 协议机停止具有低优先级标识符的报文传输。CANFD 协议机器自动尝试在下一个可能的传输位置重新传输被停止的报文。

只要一条报文通过了其中一个过滤器，那么它将会被接收，并且将报文存入到 RBUF 中，如果接收中断使能 RIE，则 RIF 被置位。如果报文不被接收，则未设置 RIF，RBUF 槽的指针不会增加。未被接收的报文将被丢弃，并被下一条报文覆盖。存储的有效报文不会被任何未接收的报文覆盖。

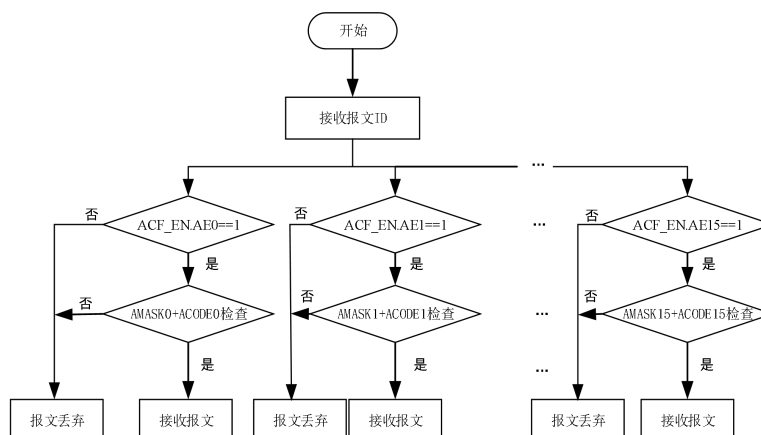


图 4-32 过滤流程图

滤波 ID (ACODE)，确认了接收特定报文，而接收掩码 (AMASK) 的每一位屏蔽对应 ACODE 位，在报文 ID 比较时该位被忽略，进一步扩展了 ACODE 滤波范围。

上电滤波器的 AE0 默认置位，AMASK0 默认全 1，也就是会接收所有报文。

4.7.11.2 配置说明

● 接收滤波

- 配置 ACF_EN，开启接收过滤器使能；
- 配置 ACF_x.AMASK，定义需要比较需要屏蔽的位；
- 配置 ACF_x.ACODE，定义需要比较位所对应的值；
- 接收到报文 ID 轮询过滤掉 ACF_x.AMASK 需要比较位的条件下，与 ACF_x.ACODE 一致，则接收缓存 RBUG0~15，则接收报文，否则不缓存报文，舍弃。

4.7.12 休眠模式

休眠模式的激活是通过 CFG_STAT.STBY 位，并且一旦激活休眠模式，则 CANFD 节点不能发送帧。

休眠模式下，通过 RX 接口监控 CANFD 总线的显性状态，显性状态下，CANFD 控制器会自动清除 CFG_STAT.STBY 位，此时 CANFD 节点从休眠模式进入唤醒模式，通过配置 TRAN 寄存器，定义的唤醒时间，在唤醒过程中 TX 无法发送 ACK 信号，唤醒成功后，会发送 ACK 信号，完成 CANFD 通信。

休眠模式启动配置说明。

- 休眠过程中，配置 TPE、TSONE、TSALL 位无效；
- CANFD 节点启动传输，则 STBY 配置无效；
- 检测 RX 端显性清除 STBY 位；
- 手动配置写 0，清除 STBY 位；

4.7.13 错误

CANFD 控制器内部有错误处理机制，不但能够自动消息重新传输和自动删除收到的错误信息，并且还能给出错误类型标识和错误计数的值。

4.7.13.1 错误的状态和种类

CANFD 控制器有 3 种错误状态信息，具体如下。

- 主动错误：节点在检测到错误时自动传输主动错误帧；
 - 接收或发送错误计数小于 128；
 - 主动错误状态是可以正常进行 CANFD 总线通信的状态；
 - 主动错误如果没发现错误，真个总线也被认为没有错误；
- 被动错误：易引起错误的状态；
 - 接收或发送错误计数大于 127；
 - 虽然进行 CANFD 总线通信，但会发送错误通知。
- 总线关闭状态
 - 发送错误计数大于 255；
 - 总线关闭，信息的接收和发送都被禁止；

上述 3 种错误状态，CANFD 控制器有两个错误计数器：发送和接收错误计数器。两者都按照 CANFD 规范的定义递增和递减，并且节点在达到 CANFD 规范定义的计数器级别时进入适当的错误状态。

如果出现错误，错误计数器将递增。可能由该节点引起的危险错误将导致计数器增加 8，可能由

其他节点引起的错误将导致计数增加 1。有效的帧传输或有效的接收导致计数器递减。

在帧的传输过程中，产生错误帧，CANFD 协议机会重新启动传输，直到该帧被无错误传输，或者主机节点进入总线关闭。

错误帧是至少 6 个连续比特位，这对于其他节点来说是违反 CANFD 的位填充操作。

4.7.13.2 错误警告

通过配置 LIMIT_EINT.EWL 寄存器作为错误计数限制，发出警告，CANFD 控制器可以通过错误计数器警告的限制，做一些预防措施。

4.7.13.3 仲裁丢失错误

在帧传输仲裁段数据时，产生错误，CANFD 控制器可以通过 DELAY_EALCAP.ALC 确认仲裁丢失错误的位置。ALC[31:0]，最大值 31 位对应扩展帧的 RTR/RRS 位。

4.7.13.4 总线关闭状态

总线关闭状态为错误计数器值大于 255，并且可以通过检测 CFG_STAT.BUSOFF，获取总线关闭状态。

从总线线关闭状态，激活总线方式如下。

- 通过系统复位；
- 接收检测到 128 次连续 11 个隐形位（高），让 CANFD 节点返回错误激活状态，进行复位。

节点在总线关闭状态，存在待发送的帧，则该帧将保持挂起状态。从总线线关闭状态，到总线激活状态，则将启动挂起的帧。

4.7.13.5 错误标识

寄存器 DELAY_EALCAP.KOER2 位标识了错误类型，CANFD 总线上的错误，每次产生错误都会更新，也就是 KOER2 是最新的错误类型。

4.7.14 寄存器描述

CAN FD 地址空间（0x6C80~0x6F7F）内，地址空间内包括接收帧地址、发送帧地址、CAN FD 控制和状态寄存器地址、TIMER64 定时器寄存器地址、休眠模式启动发送 TRAN 的 32 位计时器的地

址空间，具体寄存器偏移地址映射表如下。

表 4-26 CAN FD寄存器

地址偏移	寄存器名称	初始值	寄存器描述
0x0~0x27	RBUF	随机值	接收缓存寄存器，显示DSRAM中的随机值
0x28~0x4F	TBUF	随机值	发送缓存寄存器，显示DSRAM中的随机值
0x50	CFG_STAT	0x80	命令和状态寄存器
0x51	CTRL	0x90	控制寄存器
0x52	RTINTFE	0xFE	中断使能和标志寄存器
0x53	LIMIT_EINT	0x1B00	警告限制和错误中断寄存器
0x54	S_SEG	0x0203	慢速位定时寄存器
0x55	S_CFG	0x0102	慢速位配置寄存器
0x56	F_SEG	0x0203	快速位定时寄存器
0x57	F_CFG	0x0102	快速位配置寄存器
0x58	DELAY_EALCAP	0x0	传输延迟补偿和错误标志寄存器
0x59	ECNT	0x0	错误计数器
0x5A	CIA_ACF_CFG	0x200	CIA603和滤波配置寄存器
0x5B	ACF_EN	0x1	接收滤波使能寄存器
0x5C	ACF_0	随机值	接收代码/掩码寄存器0，显示DSRAM中的随机值，用户需要初始化配置为0x0/0x1FFF
0x5D	ACF_1	随机值	接收代码/掩码寄存器1，显示DSRAM中的随机值，用户需要初始化配置为0x0/0x1FFF
0x5E	Reserved	0x0	保留
0x5F	TTCFG_TBPTR	0x9000	时间触发配置和发送槽指针寄存器
0x60	REFID_0	0x0	参考ID寄存器0
0x61	REFID_1	0x0	参考ID寄存器1
0x62	TRIG_CFG	0x0	触发配置寄存器
0x63	TT_TRIG	0x0	触发时间寄存器
0x64	TT_WIRIG	0xFFFF	看门狗触发寄存器
0x65~0x6F	Reserved	0x00	保留
0x70	CANCFG	0x00	CAN 控制寄存器
0x71~0x7F	Reserved	0x00	保留
0x80	TIMERTIM0	0x00	定时器计数器0寄存器
0x81	TIMERTIM1	0x00	定时器计数器1寄存器
0x82	TIMERTIM2	0x00	定时器计数器2寄存器
0x83	TIMERTIM3	0x00	定时器计数器3寄存器
0x84	TIMERPRD0	0x0	定时器周期0寄存器
0x85	TIMERPRD1	0x0	定时器周期1寄存器
0x86	TIMERPRD2	0x0	定时器周期2寄存器
0x87	TIMERPRD3	0x0	定时器周期3寄存器
0x88	TIMERTCR	0x0	定时器控制寄存器
0x89	TIMERTPR0	0x0	定时器预分频寄存器
0x8A	TIMERTPR1	0x0	定时器分频寄存器
0x8B~0xFF	Reserved	0x00	保留
0x100	TRANCNT0	0x0	传输控制器计数器0寄存器
0x101	TRANCNT1	0x0	传输控制器计数器1寄存器
0x102	TRANPRD0	0xFFFF	传输控制器周期0寄存器
0x103	TRANPRD1	0x0	传输控制器周期1寄存器
0x104	TRANTPR0	0x0	传输控制器分频0寄存器
0x105	TRANTPR1	0x0	传输控制器分频1寄存器
0x106~0x2FF	Reserved	0x00	保留

4.7.14.1 接收帧寄存器 (RBUF)

RBUF 存储在 DSRAM 中，RBUF 显示的默认值都是 DSRAM 中的随机值，下面寄存器中描述的默认值 0x0，是在 DSRAM 上电后所有值都是 0x0 的情况，DSRAM 中 RBUF 缓存只有后续接收帧进

行对应位置的更新。

表 4-27 RBUF映射表

地址	名称
288~307	RBUF0+RTS[63:0]
308~327	RBUF1+RTS[63:0]
328~347	RBUF2+RTS[63:0]
348~367	RBUF3+RTS[63:0]
368~387	RBUF4+RTS[63:0]
388~407	RBUF5+RTS[63:0]
408~427	RBUF6+RTS[63:0]
428~447	RBUF7+RTS[63:0]
448~467	RBUF8+RTS[63:0]
468~487	RBUF9+RTS[63:0]
488~507	RBUF10+RTS[63:0]
508~527	RBUF11+RTS[63:0]
528~547	RBUF12+RTS[63:0]
548~567	RBUF13+RTS[63:0]
568~587	RBUF14+RTS[63:0]
588~607	RBUF15+RTS[63:0]

4.7.14.1.1 接收帧的 ID 寄存器 0

接收帧的 ID 寄存器 0 用于存储接收帧的 ID 信息，标准帧 ID 有效宽度是 11 位，扩展帧 ID 有效宽度是 29 位，该寄存器地址 (0x0)，具体说明如下。

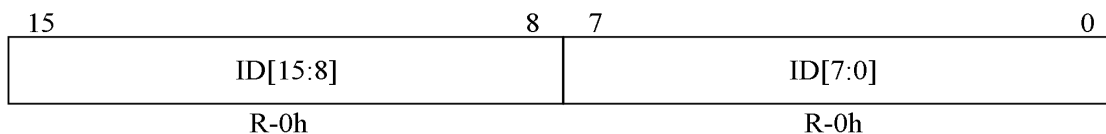


图 4-33 接收帧 ID 寄存器 0

表 4-28 接收帧ID寄存器0位域描述

位域	位名	类型	初始值	描述
15:8	ID[15:8]	R	0h	[10:8] 扩展帧和标准帧的ID有效信号位 [15:11] 扩展帧ID有效信号位、标准帧的保留位
7:0	ID[7:0]	R	0h	扩展帧和标准帧的ID有效信号位

4.7.14.1.2 接收帧的 ID 寄存器 1

接收帧的 ID 寄存器 1 用于存储接收帧的 ID 和报错类型信息，只有接收帧是扩展帧的条件下 ID 信息为有效宽度，该寄存器地址 (0x1)，具体说明如下。

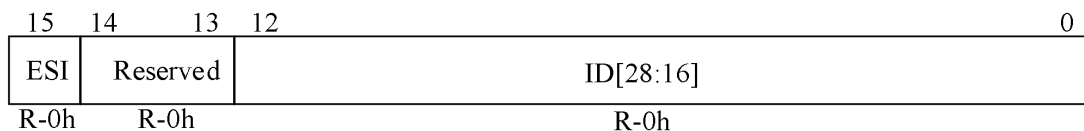


图 4-34 接收帧 ID 寄存器 1

表 4-29 接收帧ID寄存器1位域描述

位域	位名	类型	初始值	描述
15	ESI	R	0h	0: CANFD节点是错误主动 1: CANFD节点是错误被动 ESI在RBUF中对于CAN 2.0帧总是很低的。 传输的错误状态显示为寄存器ERRINT中的位EPASS。 错误状态指示符,这是RBUF的只读状态位, 在TBUF中不可用。协议机会自动将正确的ESI帧值嵌入到传输的帧中。 ESI只包含在CAN FD帧中, 而不存在于CAN 2.0帧中。
14:13	Reserved	R	0h	保留位
12:0	ID[28:16]	R	0h	扩展帧ID有效信号的[28:16]位 标准帧ID为保留位。

4.7.14.1.3 接收帧的状态寄存器

接收帧的 ID 寄存器 2 用于存储错误类型、节点总线连接标识和接收帧 DLC、BRS、FDF/EDL、RTR、IDE 状态位信息, 该寄存器地址 (0x2) , 具体说明如下。

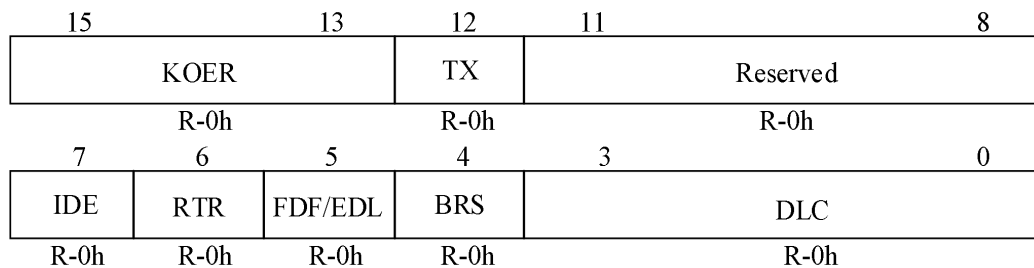


图 4-35 接收帧状态寄存器

表 4-30 接收帧状态寄存器位域描述

位域	位名	类型	初始值	描述
15:13	KOER	R	0h	错误类型 (错误代码) 000-无错误 001-位错误 010-表单错误 011-填充错误 100-确认错误 101-CRC错误 110-其他错误 (自身错误标志之后的主要位、接收到的活动错误标志太长, ACK错误之后的被动错误标志期间的主要位) 111-保留 KOER会根据每个新错误进行更新。因此, 当帧被成功发送或接收时, 它保持不变。
12	TX	R	0h	标识节点连接到总线, 传输帧。如果节点连接到总线, 那么它的TX位在其RX输入上可见。
11:8	Reserved	R	0h	保留位
7	IDE	R	0h	标识符扩展 0: 标准格式: ID (10: 0) 1: 扩展格式: ID (28: 0)
6	RTR	R	0h	远程传输请求 0: 数据帧 1: 远程帧 仅CAN 2.0帧可以是远程帧。CAN FD的没有远程帧。因此, 如果TBUF和RBUF中的FDF=1, 则RTR被强制设置为0。 如果用位RRS=1接收到CAN FD帧, 则忽略它, 期望接收数据有效载荷, RBUF中的RTR被覆盖, 但用RRS=1计算该帧的CRC。
5	FDF/EDL	R	0h	CAN FD和CAN2.0帧的标识: 0: CAN2.0帧 (最多8字节有效负载) 1: CANFD帧 (最多64字节有效负载)

4	BRS	R	0h	比特率快慢开关 0: 整个帧的标称/慢位率 1: 切换到数据有效载荷和CRC的数据/快速比特率 只有CAN FD帧可以切换比特率。因此，如果FDF=为0，BRS将被强制设置为0
3:0	DLC	R	0h	RBUF和TBUF中的数据长度段（DLC）： 一帧数据报文最多只能携带8个字节的数据内容 CAN2.0标准帧数据段字节长度： 0~8h: 数据段数据长度0~8字节； 9~Fh: 数据段数据长度8字节 CANFD FD数据段字节长度： 0~8h: 数据段数据长度0~8字节； 9h: 数据段数据长度12字节； Ah: 数据段数据长度16字节； Bh: 数据段数据长度20字节； Ch: 数据段数据长度24字节； Dh: 数据段数据长度32字节； Eh: 数据段数据长度48字节； Fh: 数据段数据长度64字节；

4.7.14.1.4 CYCLE_TIME 寄存器

TTCAN 时间戳 CYCLE_TIME 将仅在 TTCAN 模式下存储在 RBUF 中。这是在此帧的 SOF 处的循环时间。该寄存器地址（0x3），具体说明如下。

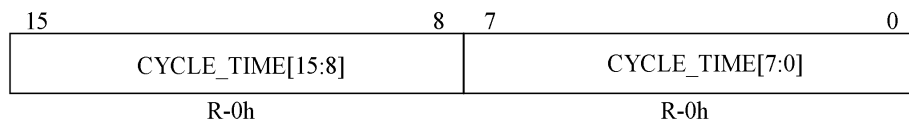


图 4-36 CYCLE_TIME 寄存器

表 4-31 CYCLE_TIME 寄存器位域描述

位域	位名	类型	初始值	描述
15:8	CYCLE_TIMER[15:8]	R	0h	SOF的循环时间寄存器的[15:8]位
7:0	CYCLE_TIMER [7:0]	R	0h	SOF的循环时间寄存器的[7:0]位

4.7.14.1.5 接收帧的数据寄存器

接收帧的数据寄存器用于存储接收帧的数据信息，CAN 2.0 帧数据最大长度是 8 字节，CAN FD 帧数据最大长度是 64 字节，该寄存器地址（0x4~0x23），具体说明如下。

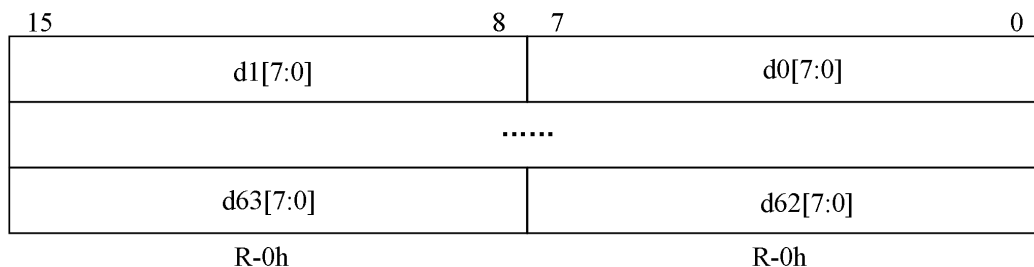


图 4-37 数据寄存器 0~3

表 4-32 数据寄存器0~3位域描述

位域	位名	类型	初始值	描述
15:8	d63[7:0]	R	0h	数据帧的63个字节数据
7:0	d62[7:0]	R	0h	数据帧的62个字节数据
... ..				
15:8	d1[7:0]	R	0h	数据帧的1个字节数据
7:0	d0[7:0]	R	0h	数据帧的0个字节数据

4.7.14.1.6 CIA603 时间戳寄存器

CiA 603 时间戳寄存器是 64 位接收时间戳（RTS）寄存器，该寄存器地址（0x24~0x27），具体说明如下。

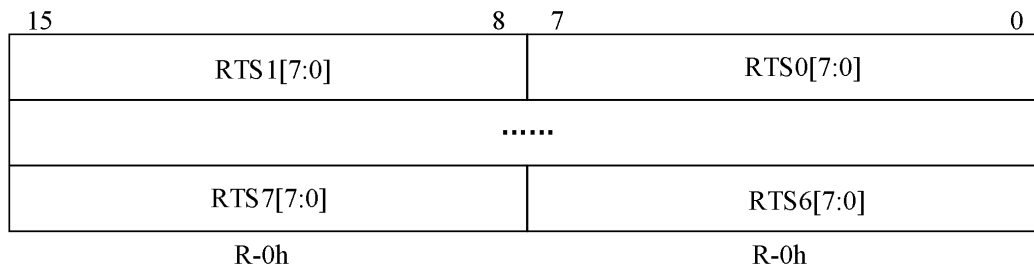


图 4-38 时间戳寄存器

表 4-33 时间戳寄存器位域描述

位域	位名	类型	初始值	描述
15:8	RTS7[7:0]	R	0h	CIA 603数据的[63:56]位域
7:0	RTS6[7:0]	R	0h	CIA 603数据的[55:48]位域
... ..				
15:8	RTS1[7:0]	R	0h	CIA 603数据的[15:8]位域
7:0	RTS0[7:0]	R	0h	CIA 603数据的[7:0]位域

4.7.14.2 发送帧寄存器（TBUF）

接收帧 TBUF 映射到 CANFD 内部的 DSRAM（640x32bit）的地址空间分布。

TBUF 存储在 DSRAM 中，TBUF 显示的默认值都是 DSRAM 中的随机值，下面寄存器中描述的默认值 0x0，是在 DSRAM 上电后所有值都是 0x0 的情况，DSRAM 中 TBUF 缓存只有后续写入发送帧，才能启动进行对应位置的更新。

表 4-34TBUF映射表

地址	名称
0~17	TBUF0
18~35	TBUF1
36~53	TBUF2
54~71	TBUF3
72~89	TBUF4
90~107	TBUF5
108~125	TBUF6
126~143	TBUF7
144~161	TBUF8
162~179	TBUF9
180~197	TBUF10

198~215	TBUF11
216~233	TBUF12
234~251	TBUF13
252~269	TBUF14
270~287	TBUF15

4.7.14.2.1 发送帧的 ID 寄存器 0

发送帧的 ID 寄存器 0 用于存储接收帧的 ID 信息，标准帧 ID 有效宽度是 11 位，扩展帧 ID 有效宽度是 29 位，该寄存器地址（0x28），具体说明如下。

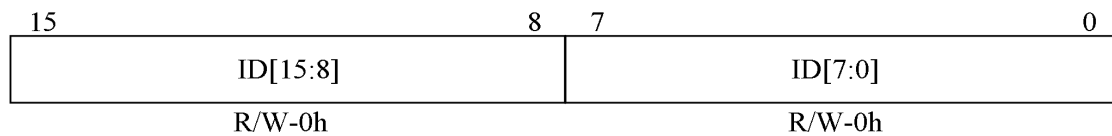


图 4-39 发送帧 ID 寄存器 0

表 4-35 发送帧ID寄存器0位域描述

位域	位名	类型	初始值	描述
15:8	ID[15:8]	R/W	0h	帧类型为扩展帧： [15:8] 扩展帧ID有效信号位 帧类型为标准帧： [10:8] 标准帧的ID有效信号位 [15:11] 标准帧的保留位
7:0	ID[7:0]	R/W	0h	扩展帧和标准帧的ID有效信号位

4.7.14.2.2 发送帧的 ID 寄存器 1

发送帧的 ID 寄存器 1 用于存储接收帧的 ID 和报错类型信息，只有发送帧是扩展帧的条件下 ID 信息为有效宽度，该寄存器地址（0x29），具体说明如下。



图 4-40 发送帧 ID 寄存器 1

表 4-36 发送帧ID寄存器1位域描述

位域	位名	类型	初始值	描述
15	ESI	R/W	0h	0: CANFD节点是错误主动 1: CANFD节点是错误被动 ESI在RBUF中对于CAN 2.0帧总是很低的。 传输的错误状态显示为寄存器ERRINT中的位EPASS。 错误状态指示符,这是RBUF的只读状态位，在TBUF中不可用。协议机会自动将正确的ESI帧值嵌入到传输的帧中。 ESI只包含在CAN FD帧中，而不存在于CAN 2.0帧中。
14:13	Reserved	R	0h	保留位
12:0	ID[28:16]	R/W	0h	扩展帧ID有效信号的[28:16]位 标准帧ID为保留位。

4.7.14.2.3 发送帧的状态寄存器

发送帧的 ID 寄存器用于存储错误类型、节点总线连接标识和接收帧 DLC、BRS、FDF/EDL、RTR、IDE 状态位信息，该寄存器地址（0x2a），具体说明如下。

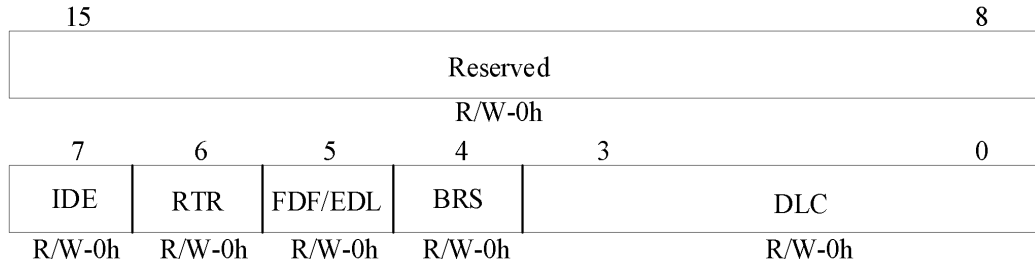


图 4-41 接收帧状态寄存器

表 4-37 接收帧状态寄存器位域描述

位域	位名	类型	初始值	描述
15:8	Reserved	R/W	0h	保留位
7	IDE	R/W	0h	标识符扩展 0: 标准格式: ID (10: 0) 1: 扩展格式: ID (28: 0)
6	RTR	R/W	0h	远程传输请求 0: 数据帧 1: 远程帧 仅CAN 2.0帧可以是远程帧。CAN FD的没有远程帧。因此，如果TBUF和RBUF中的FDF=1，则RTR被强制设置为0。 如果用位RRS=1接收到CAN FD帧，则忽略它，期望接收数据有效载荷，RBUF中的RTR被覆盖，但用RRS=1计算该帧的CRC。
5	FDF/EDL	R/W	0h	CAN FD和CAN2.0帧的标识: 0: CAN2.0帧（最多8字节有效负载） 1: CANFD帧（最多64字节有效负载）
4	BRS	R/W	0h	比特率快慢开关 0: 整个帧的标称/慢速率 1: 切换到数据有效载荷和CRC的数据/快速比特率 只有CAN FD帧可以切换比特率。因此，如果FDF=为0，BRS将被强制设置为0
3:0	DLC	R/W	0h	RBUF和TBUF中的数据长度段（DLC）： 一帧数据报文最多只能携带8个字节的数据内容 CAN2.0标准帧数据段字节长度： 0~8h: 数据段数据长度0~8字节； 9~Fh: 数据段数据长度8字节 CANFD FD数据段字节长度： 0~8h: 数据段数据长度0~8字节； 9h: 数据段数据长度12字节； Ah: 数据段数据长度16字节； Bh: 数据段数据长度20字节； Ch: 数据段数据长度24字节； Dh: 数据段数据长度32字节； Eh: 数据段数据长度48字节； Fh: 数据段数据长度64字节；

4.7.14.2.4 TBUF 保留寄存器

在 RBUF 寄存器中该寄存器地址（0x2B）为保留，可读写，具体说明如下。

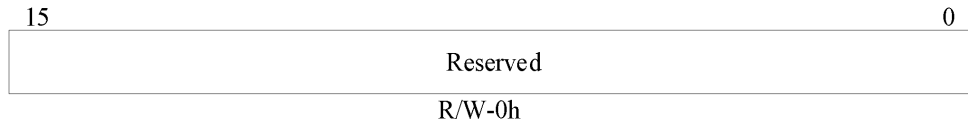


图 4-42 TBUF 保留寄存器

表 4-38 TBUF保留寄存器位域描述

位域	位名	类型	初始值	描述
15:0	Reserved	R/W	0h	保留

4.7.14.2.5 发送帧的数据寄存器

发送帧的数据寄存器用于存储发送帧的数据信息，CAN 2.0 帧数据最大长度是 8 字节，CAN FD 帧数据最大长度是 64 字节，该寄存器地址（0x2c~0x4b），具体说明如下。

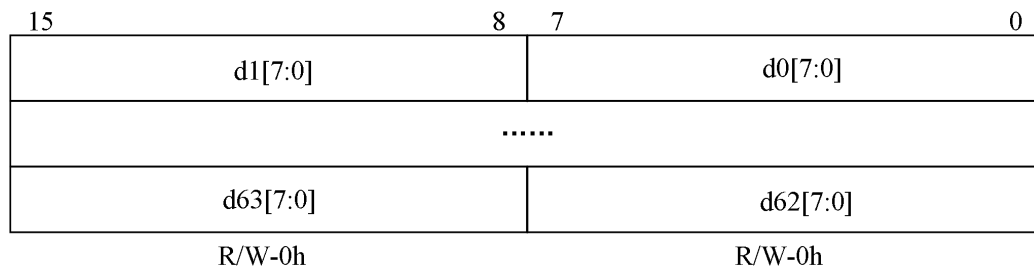


图 4-43 数据寄存器 0~3

表 4-39 数据寄存器0~3位域描述

位域	位名	类型	初始值	描述
15:8	d63[7:0]	R/W	0h	数据帧的63个字节数据
7:0	d62[7:0]	R/W	0h	数据帧的62个字节数据
.....				
15:8	d1[7:0]	R/W	0h	数据帧的1个字节数据
7:0	d0[7:0]	R/W	0h	数据帧的0个字节数据

4.7.14.2.6 CIA603 时间戳寄存器

CiA 603 时间戳寄存器是 64 位发送时间戳（TTS）寄存器，该寄存器地址（0x4c~0x4f），具体说明如下。

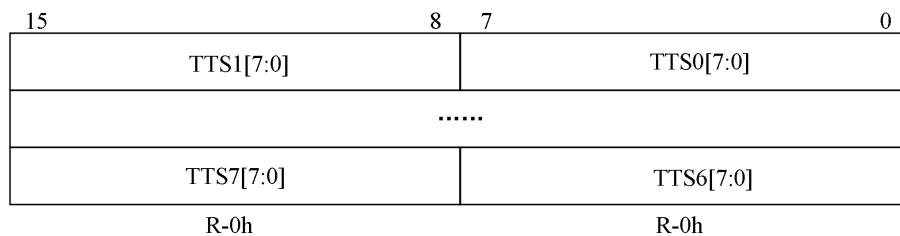


图 4-44 时间戳寄存器

表 4-40 时间戳寄存器位域描述

位域	位名	类型	初始值	描述
15:8	TTS7[7:0]	R	0h	CIA 603数据的[63:56]位域
7:0	TTS6[7:0]	R	0h	CIA 603数据的[55:48]位域
... ..				
15:8	TTS1[7:0]	R	0h	CIA 603数据的[15:8]位域
7:0	TTS0[7:0]	R	0h	CIA 603数据的[7:0]位域

4.7.14.3 命令和状态寄存器 (CFG_STAT)

命令和状态寄存器主要是发送命令、收发配置和状态信息，具体说明如下。

15	14	13	12	11	10	9	8
TBSEL	LOM	STBY	TPE	TPA	TSONE	TSALL	TSA
R/W-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESET	LBME	LBMI	TPSS	TSSS	RACTIVE	TACTIVE	BUSOFF
R/W-1h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R-0h	R-0h

图 4-45 命令和状态寄存器

表 4-41 命令和状态寄存器位域描述

位域	位名	类型	初始值	描述
15	TBSEL	R/W	0h	传输缓冲区选择 选择要加载的消息的传输缓冲区。使用TBUF寄存器进行访问。 TBSEL需要在写入TBUF寄存器和设置TSNEXT时一直保持稳定的。 0: TBUF0 (高优先级缓冲区) 1: TBUF1~15 (次要或者备选缓冲区) 如果 (TTEN=1和TTTBM=1)，该位将被重置为硬件重置值。 注意：在CFG_STAT.RESET为1时，保持为默认值。
14	LOM	R	0h	仅侦听模式 0: 禁用 1: 使能 如果设置了TPE、TSONE或TSALL，则无法设置启用LOM。如果启用了LOM并禁用了LBME，则无法启动传输。 LOM=1和LBME=0将禁用所有传输。 LOM=1和LBME=1为接收到的帧和错误帧禁用ACK，但允许传输自己的帧。 TPE=0，TSONE，TSALL=0，则LOM置1。
13	STBY	R/W	0h	收发器待机模式 0: 禁用 1: 启用 此寄存器位连接到输出信号stby，可用于控制收发器的待机模式。 如果为TPE=1、TSONE=1或TSALL=1，则不能将STBY设置为1。 如果主机将STBY设置为0，那么主机需要等待收发器需要的启动时间，然后主机请求新的传输。
12	TPE	R/W	0h	传输主启用位 1: 为高优先级TBUF0中的消息启用传输 0: 对TBUF0没有传输 如果设置了TPE，则来自TBUF0的消息将在下一个可能的传输位置进行传输。从TBUF1~15开始的传输将在之前完成，但是待定的新消息将被延迟，直到TBUF0消息已经传输完毕。TPE保持设置，直到消息传输成功或使用TPA中止。主机控制器可以将TPE设置为1，但不能重置为0。这只能使用TPA和中止消息。 如果RESET=1、STBY=1、(LOM=1和LBME=0)或(TTEN=1和TTTBM=1)，则该位将被重置为硬件复位值。 注意：在CFG_STAT.RESET为1时，保持为默认值。
11	TPA	R/W	0h	传输主中止位

				1: 中止由TPE=1请求但尚未启动的来自TBUF0的传输。(消息的数据字节仍保留在TBUF0中。) 0: 不中止 该位必须由主机控制器设置, 并将由CANFD控制器重置。设置TPA会自动取消断言TPE。 主机控制器可以将TPA设置为1, 但不能重置为0。 在控制器重置该位的短时间内, 主机无法设置该位。如果RESET=1或(TTEN=1和TTTBM=1), 位将被重置为硬件复位值。TPA不应与TPE同时设置。 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
10	TSONE	R/W	0h	次要传输启动 1: 在TBUF1~15中的一个传输启用。在FIFO模式下, 这是最早的消息, 而在优先级模式下, 这是优先级最高的消息。优先级模式下的TSONE很难处理, 因为如果同时将新消息写入TBUF1~15, 并不总是清楚会传输哪条消息。一旦总线变空, 没有TBUF0(位TPE)的请求, 控制器立即启动传输。 0: TBUF1~15没有传输。 TSONE保持设置, 直到消息被成功传输或使用TSA中止。 主机控制器可以将TSONE设置为1, 但不能重置为0。这只能使用TSA和中止消息。如果RESET=1、STBY=1、(LOM=1和LBME=0)或(TTEN=1和TTTBM=1), 则该位将被重置为硬件复位值。 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
9	TSALL	R/W	0h	次要传输所有帧 1: 传输TBUF1~15中的所有消息。一旦总线变空, 没有TBUF0(位TPE)的请求, 控制器立即启动传输。 0: TBUF1~15没有传输。 TSALL保持设置, 直到所有消息成功传输或使用TSA中止。主机控制器可以将TSALL设置为1, 但不能重置为0。这只能使用TSA和中止消息。 如果RESET=1、STBY=1、(LOM=1和LBME=0)或(TTEN=1和TTTBM=1), 则该位将被重置为硬件复位值。如果在传输期间, TBUF1~15加载了一个新的帧, 那么新的帧也将被传输。换句话说: 当TBUF1~15变空时, 由TSALL启动的传输将完成。 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
8	TSA	R/W	0h	次要传输中止 1: 中止已请求但尚未启动的TBUF1~15传输。对于TSONE传输, 只有一帧中止, 而对于TSALL传输, 所有帧都中止。一个或所有消息槽将发布更新TSSTAT。所有中止的消息都将丢失, 因为它们再也无法访问了。如果在优先模式下TSONE传输中止, 那么如果新帧同时写入TBUF1~15, 则不清楚哪个帧将中止。 0: 不中止 该位必须由主机控制器设置, 并将由CANFD控制器重置。设置TSA, 分别自动取消断言TSONE或TSALL。主机控制器可以将TSA设置为1, 但不能重置为0。如果复位=1, 则该位将被重置为硬件复位值。TSA不应与TSONE或TSALL同时设置。 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
7	RESET	R/W	1h	重置请求位 1: 主机控制器执行CANFD控制器的本地重置。 0: 没有CANFD控制器的本地重置。 对于某些寄存器(例如, 节点配置)只能在RESET=1时进行修改
6	LBME	R/W	0h	外部回环模式 0: 禁用 1: 使能 注意: 当传输被激活时, 不应启用LBME, 在CFG_STAT.RESET为1时, 保持为默认值。
5	LBMI	R/W	0h	内部回环模式 0: 禁用 1: 使能 注意: 当传输被激活时, 不应启用LBMI, 在CFG_STAT.RESET为1时, 保持为默认值。
4	TPSS	R/W	0h	针对TBUF0的传输主要单次模式 0: 禁用 1: 使能 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
3	TSSS	R/W	0h	针对TBUF1~15的传输次要单次模式

				0: 禁用 1: 使能 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
2	RACTIVE	R	0h	接收处于活动状态 (接收状态位) 1: 控制器当前正在接收一个帧。 0: 没有接收活动。
1	TACTIVE	R	0h	发送处于活动状态 (发送状态位) 1: 控制器当前正在接收一个帧。 0: 没有接收活动。
0	BUSOFF	R	0h	总线关闭, 总线状态位 1: 控制器状态为“总线关闭”。 0: 控制器状态为“总线接通”。 写一个1到BUSOFF将重置TECNT和RECN。这应该只在调试时这样做。

4.7.14.4 控制寄存器 (CTRL)

控制寄存器为收发传输控制寄存器, 具体说明如下。

15	14	13	12	11	10	9	8
SACK	ROM	ROV	RREL	RBALL	Reserved	RSTAT	
R/W-0h	R-0h	R/W1C-0h	R0/W-0h	R/W-0h	R/W-0h	R-0h	
7	6	5	4	3	2	1	0
FD_ISO	TSNEXT	TSMODE	TTTBM	Reserved		TSSTAT	
R-1h	R0/W-0h	R/W-0h	R/W-1h	R-0h		R-0h	

图 4-46 控制寄存器

表 4-42 控制寄存器位域描述

位域	位名	类型	初始值	描述
15	SACK	R/W	0h	自我应答机制 0: 无自我应答 1: 当LBME=1, 时自我应答 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
14	ROM	R/W	0h	接收缓存溢出模式 如果在收到新消息时出现完整的RBUF, 则ROM将选择以下内容: 1: 将不会存储新消息。 0: 最早的消息将被覆盖。
13	ROV	R/W1C	0h	接收缓冲区溢出 1: 溢出。至少丢失了一条消息。 0: 没有溢出。 通过设置RREL=1来设置ROV, 写1清除 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
12	RREL	R0/W	0h	接收缓存释放 主机控制器已读取了实际的RB插槽并释放了它。然后, CAN控制器指向下一个RB插槽。RSTAT更新。 1: 释放: 主机已读取了RB。 0: 没有释放 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
11	RBALL	R/W	0h	接收缓冲区存储所有数据帧 0: 正常操作 1: RB存储正确的数据帧以及有错误的帧 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
10	Reserved	R	0h	保留
9:8	RSTAT[1:0]	R	0h	接收缓冲区状态信息 00: 空 01: 非空, 小于几乎满 (AFWL) 状态

				10：大于几乎满（可编程阈值由AFWL），但不满，没有溢出 11：满（保持设置的情况下溢出-溢出信令参见ROV）
7	FD_ISO	R	1h	CAN FD ISO模式 0：Bosch CANFD（非ISO）模式 1：ISO CAN FD模式（ISO 11898-1：2015） ISO CAN FD模式具有不同的CRC初始化和额外位的计数。这两种模式都不兼容，并且不能在一个CAN网络中混合。这个位对CAN 2.0B没有影响。此位仅在RESET=1时才可写。 注意：在CFG_STAT.RESET=1时，可写。
6	TSNEXT	R0/W	0h	下一个次要传输缓冲区 0：没有激活 1：TBUF1~15插槽，选择下一个插槽。 注意：可写，读一直为0 在所有的帧字节都被写入到TBUF寄存器之后，主机控制器必须设置TSNEXT来表示该插槽已被填充。然后，CANFD控制器心将TBUF寄存器连接到下一个插槽。一旦槽标记为填充，可以使用TSONE或TSALL启动传输。 可以在一次写访问中将TSNEXT和TSONE或TSALL设置在一起。TSNEXT必须由主机控制器设置，并在设置后立即由CANFD控制器心自动重置。 如果TBSEL=0，则设置TSNEXT没有意义。在这种情况下，TSNEXT将被忽略并自动清除。它不会造成任何伤害。 如果TBUF1~15的所有插槽都已填满，TSNEXT将保持设置，直到插槽空闲 TSNEXT在TTCAN模式下没有任何意义，并且被固定为0。 （TTEN、TTTBM为1条件下为0）
5	TSMODE	R/W	0h	次要传输缓冲区操作模式 0：FIFO模式 1：优先级决策模式 在FIFO模式中，帧按照它们被写入TBUF1~15的顺序进行传输。在优先级决策模式中，优先级最高的帧是TBUF1~15中优先级最高的帧。一个帧的ID用于优先级决策。ID越低，表示帧的优先级越高。无论ID如何，TBUF0中的一个帧始终具有最高的优先级。只有当TBUF1~15为空时，才应切换TSMODE。
4	TTTBM	R/W	0h	TTCAN传输缓冲区模式如果TTEN=0，那么TTTBM被忽略，否则以下是有效的： 0：单独的TBUF0和TBUF1~15，行为定义的TSMODE 1：完整的TTCAN支持：缓冲区插槽可由TBPTR和TTPTR选择对于事件驱动的CANFD通信（TTEN=0），系统提供了TBUF0和TBUF1~15，TBUF1~15的行为由TSMODE定义。然后，TTTBM将被忽略。 对于完全支持包括时间触发传输在内的所有功能的时间触发CANFD通信（TTEN=1），需要选择TTTBM=1。然后，所有的TB插槽都可以使用TTPTR和TBPTR进行寻址。 对于仅支持接收时间戳的有时间触发的CANFD通信（TTEN=1），可以选择TTTBM=0。然后传输缓冲区作为事件驱动模式，行为可以由TSMODE选择。 只有在TBUF为空时，才能切换TTTBM。
3:2	Reserved	R	0h	保留
1:0	TSSTAT	R	0h	次要传输状态位如果TTEN=0或TTTBM=0： 00：TBUF1~15为空 01：TBUF1~15小于或等于半满 10：TBUF1~15大于半满 11：TBUF1~15已满。 如果使用STB_DISABLE禁用TBUF1~15，则使用TSSTAT=00。如果TTEN=1和TTTBM=1： 00：TBUF0和TBUF1~15为空 01：TBUF0和TBUF1~15不为空 11：TBUF0和TBUF1~15为满

4.7.14.5 中断使能和标志寄存器（RTINTFE）

中断使能和标志寄存器用于使能帧在收发状态下产生的中断使能和产生中断标志，具体描述如下。

15	14	13	12	11	10	9	8
RIF	ROIF	RFIF	RAFIF	TPIF	TSIF	EIF	AIF
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
RIE	ROIE	RFIE	RAFIE	TPIE	TSIE	EIE	TSFF
R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R/W-1h	R-0h

图 4-47 中断使能和标志寄存器

表 4-43 中断使能和标志寄存器位域描述

位域	位名	类型	初始值	描述
15	RIF	R/W1C	0h	接收中断标志 0: 没有收到任何帧 1: 数据或远程帧已被接收, 并在缓冲区可读 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
14	ROIF	R/W1C	0h	RB溢出中断标志 0: 没有覆盖RB 1: 至少有一条已收到的消息在RB中被覆盖 如果出现溢出现象, 将同时设置ROIF和RFIF。 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
13	RFIF	R/W1C	0h	RB满中断标志 0: RB缓存还未满 1: 所有RBs都满了。如果在收到下一个有效消息之前没有释放RB, 则最早的消息将丢失。 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
12	RAFIF	R/W1C	0h	RB几乎全满中断标志 0: 已填充的RB插槽数小于AFWL_i 1: 已填充的RB插槽数大于等于AFWL_i 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
11	TPIF	R/W1C	0h	传输主要中断标志 0: TBUF0的传输未完成 1: 所请求的TBUF0传输已成功完成 在TTCAN模式下, 不会设置TPIF, 只有TSIF有效 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
10	TSIF	R/W1C	0h	传输次要中断标志 0: 未完成TBUF1~15传输 1: 锁请求TBUF1~15传输完成 在TTCAN模式下, TSIF将发送所有成功传输的信号, 不管消息存放位置 注意: 在CFG_STAT.RESET为1时, 保持为默认值。
9	EIF	R/W1C	0h	错误中断标志 0: 无错误 1: 错误警告限制的边界已经两个方向, 或者两个方向改变了BUSOFF位。
8	AIF	R/W1C	0h	中止中断标志 1: 设置TPA或TSA后, 指定的消息已中止。建议不要同时设置TPA和TSA, 因为这两个源都是AIF。 0: 未执行中止行为。 AIF没有关联的启用寄存器。
7	RIE	R/W	1h	接收中断使能 1: 使能 0: 禁止
6	ROIE	R/W	1h	RB溢出中断使能 1: 使能 0: 禁止
5	RFIE	R/W	1h	RB满中断使能 1: 使能 0: 禁止
4	RAFIE	R/W	1h	RB几乎全满中断使能

				1: 使能 0: 禁止
3	TPIE	R/W	1h	传输主要中断使能 1: 使能 0: 禁止
2	TSIE	R/W	1h	传输次要中断使能 1: 使能 0: 禁止
1	EIE	R/W	1h	错误中断使能 1: 使能 0: 禁止
0	TSFF	R	0h	如果TTEN=0或TTBM=0: 次要传输缓冲区满标志 1: TBUF1~15中填充了最大的消息数 0: TBUF1~15中没有填充最大的消息数 如果TTEN=1或TTBM=1: 传输缓冲区槽全标志 1: TBPTR选择的缓冲区槽被填充 0: TBPTR所选择的缓冲区插槽为空

4.7.14.6 警告限制和错误中断寄存器 (LIMIT_EINT)

该寄存器用于配置错误中断，以及标识错误类型，接收缓存个数的警告限制、接收或发送警告的个数配置，具体内容如下。

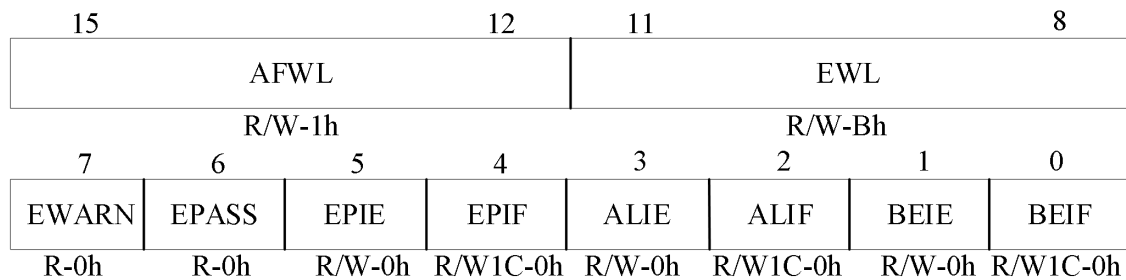


图 4-48 警告限制和错误中断寄存器

表 4-44 警告限制和错误中断寄存器位域描述

位域	位名	类型	初始值	描述
15:12	AFWL	R/W	1h	接收缓冲区几乎满警告限制 AFWL定义了内部警告限制AFWL_i，nRB是可用的RB插槽的数量。 将AFWL_i与填充的RB插槽的数量进行比较，如果相等，则触发RAFIF。 AFWL = 0毫无意义，并自动被视为0x1。（请注意，在这个规则中指的是AFWL，而不是AFWL_i。） AFWL_i > nRB是无意义的，并自动被视为nRB。 AFWL = nRB是一个有效的值，但请注意，RFIF也存在。
11:8	EWL	R/W	Bh	可编程错误警告LIMIT= (EWL+1) *8 注意：在CFG_STAT.RESET为1时，保持为默认值。
7	EWARN	R/W	1h	错误计数器警告达到限制数值 1: 其中一个错误计数器RECNT或TECNT等于或大于EWL 0: 两个计数器中的值都小于EWL。
6	EPASS	R/W	1h	错误模式 0: 非主动（节点为错误主动） 1: 主动（节点为错误被动）
5	EPIE	R/W	1h	启用被动中断错误
4	EPIF	R/W1C	1h	被动中断标志出错。如果错误状态从错误激活变为错误被动，反之，如果启用，则EPIF将被激活。 注意：在CFG_STAT.RESET为1时，保持为默认值。

3	ALIE	R/W	1h	仲裁丢失中断使能
2	ALIF	R/W1C	1h	仲裁丢失中断标志 注意：在CFG_STAT.RESET为1时，保持为默认值。
1	BEIE	R/W	1h	总线错误中断使能
0	BEIF	R/W1C	0h	总线错误中断标志 注意：在CFG_STAT.RESET为1时，保持为默认值。

4.7.14.7 慢速位定时寄存器 (S_SEG)

位时钟配置寄存器中慢速区域的位采样配置，具体说明如下。

15	14	8	7	0
Reserved	S_Seg_2		S_Seg_1	
R-0h	R/W-2h		R/W-3h	

图 4-49 慢速位定时寄存器

表 4-45 慢速位定时寄存器位域描述

位域	位名	类型	初始值	描述
15	Reserved	R	0h	保留位
14:8	S_SEG2	R/W	2h	位定时段2（慢速）采样时间开始后，采样点将设置为 $t_{SEG2} = (SEG2+1) * TQ$ 。 注意：在CFG_STAT.RESET为1时，可写。
7:0	S_SEG1	R/W	3h	位定时段1（慢速）采样时间开始后，采样点将设置为 $t_{SEG1} = (SEG1+2) * TQ$ 。 注意：在CFG_STAT.RESET为1时，可写。

4.7.14.8 慢速位配置寄存器 (S_CFG)

慢速位配置寄存器用于设置慢速时钟分频、以及跳转宽度的配置。

15	8	7	6	0
S_PRESC	Reserved		S_SJW	
R/W-1h	R-0h		R/W-2h	

图 4-50 慢速位预分频和定时器跳转宽度寄存器

表4-46 慢速位预分频和定时器跳转宽度寄存器位域描述

位域	位名	类型	初始值	描述
15:8	S_PRESC	R/W	1h	预分频（慢速率） 预分频系统时钟，得到分频时钟clock_tq_clk。有效范围按 $= [0x00, 0xff]$ ，分频值为1到256。 注意：在CFG_STAT.RESET为1时，可写。
7	Reserved	R	0h	保留位
6:0	S_SJW	R/W	2h	同步跳转宽度（慢速度）同步跳转宽度 $t_{SJW} = (SJW+1) * TQ$ 是缩短或延长重新同步的比特时间的最大时间，其中TQ是一个时间量子。 注意：在CFG_STAT.RESET为1时，可写。

4.7.14.9 快速位定时寄存器 (F_SEG)

位时钟配置寄存器中快速区域的位采样配置，具体说明如下。

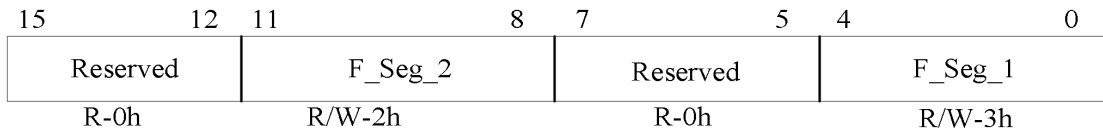


图 4-51 快速位定时寄存器

表4-47快速位定时寄存器位域描述

位域	位名	类型	初始值	描述
15:12	Reserved	R	0h	保留位
11:8	F_SEG2	R/W	2h	位定时段2（快速）采样时间开始后，采样点将设置为 $t_{SEG2} = (SEG2+1) * TQ$ 。 注意：在CFG_STAT.RESET为1时，可写。
7:5	Reserved	R	0h	保留位
4:0	F_SEG1	R/W	3h	位定时段1（快速）采样时间开始后，采样点将设置为 $t_{SEG1} = (SEG1+2) * TQ$ 。 注意：在CFG_STAT.RESET为1时，可写。

4.7.14.10 快速位配置寄存器 (F_CFG)

快速位配置寄存器用于设置慢速时钟分频、以及跳转宽度的配置。

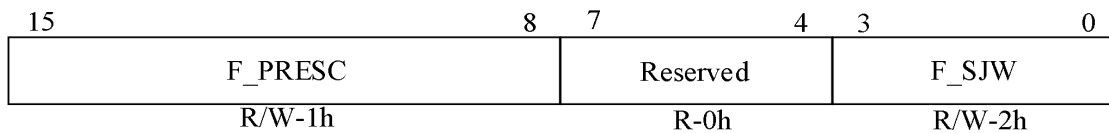


图 4-52 快速位配置寄存器

表 4-48快速位配置寄存器位域描述

位域	位名	类型	初始值	描述
15:8	F_PRESC	R/W	1h	预分频（快速率） 预分频系统时钟，得到分频时钟clock_tq_clk。有效范围按 $= [0x00, 0xff]$ ，分频值为1到256。 注意：在CFG_STAT.RESET为1时，可写。
7:4	Reserved	R	0h	保留位
3:0	F_SJW	R/W	3h	同步跳转宽度（快速度）同步跳转宽度 $t_{SJW} = (SJW+1) * TQ$ 是缩短或延长重新同步的比特时间的最大时间，其中TQ是一个时间量子。 注意：在CFG_STAT.RESET为1时，可写。

4.7.14.11 传输延迟补偿和错误标志寄存器 (DELAY_EALCAP)

传输延迟补偿和错误标志寄存器具体描述如下。

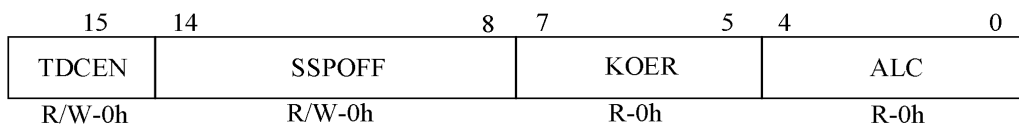


图 4-53 传输延迟补偿和错误标志寄存器

表 4-49 传输延迟补偿和错误标志寄存器位域描述

位域	位名	类型	初始值	描述
15	TDCEN	R/W	0h	延迟补偿使能 如果帧的BRS位为1， 0：延迟补偿禁用 1：在CAN FD帧的数据阶段，发射机延迟补偿被激活 注意：在CFG_STAT.RESET为1时，可写。
14:8	SSPOFF	R/W	0h	次采样点发射器延迟加SSPOFF定义了延迟补偿的次采样点的时间。SSPOFF是TQ的数量。 注意：在CFG_STAT.RESET为1时，可写。
7:5	KOER	R	3h	错误的类型（错误代码） 000：没有错误 001：位错误 010：FORM错误 011：STUFF错误 100：ACKNOWLEDGEMENT错误 100：CRC错误 110：其他错误（自己的错误标志后，收到活动错误标志过长，主导位在被动错误后ACK错误） 111：不使用 KOER更新与每个新的错误。因此，当帧成功地传输或接收帧时，它将保持不变。 注意：在CFG_STAT.RESET为1时，保持为默认值。
4:0	ALC	R	0h	仲裁丢失捕获（仲裁丢失的所在帧的位置） 注意：在CFG_STAT.RESET为1时，保持为默认值。

4.7.14.12 错误计数寄存器（ECNT）

错误计数寄存器用于标识发送错误和接收错误的个数，具体说明如下。

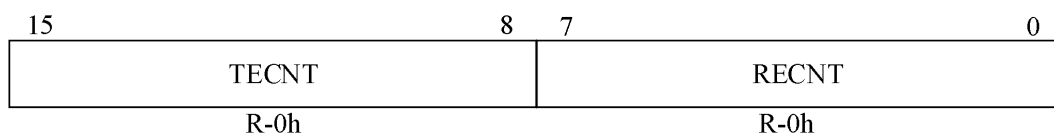


图 4-54 错误计数寄存器

表4-50 错误计数寄存器位域描述

位域	位名	类型	初始值	描述
15:8	TECNT	R	0h	传输错误计数器（传输过程中的错误数） TECNT按照CANFD规范中的定义进行增加和减少。 如果是“总线关闭状态”，TECNT可能会溢出。 如果TXB=为1，则错误计数器将被冻结。
7:0	RECNT	R	0h	接收错误计数器（接收期间的错误数） RECNT按照CANFD规范中的定义进行增加和减少。 RECNT不会溢出。如果TXB=为1，则错误计数器将被冻结。

4.7.14.13 CIA603 和滤波配置寄存器（CIA_ACF_CFG）

CIA603 和滤波配置寄存器用于 CIA603 的时间戳配置和滤波配置，具体说明如下。

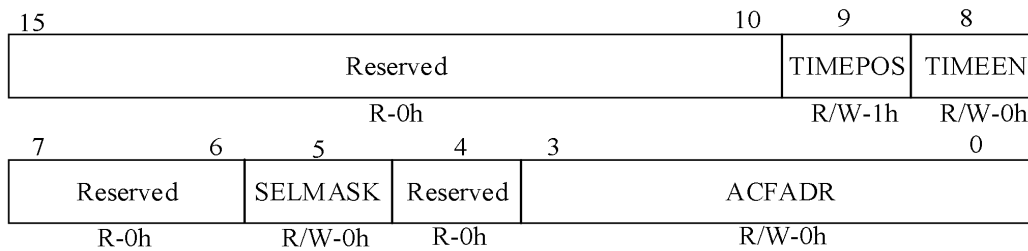


图 4-55 CIA603 和滤波配置寄存器

表 4-51 CIA603和滤波配置寄存器位域描述

位域	位名	类型	初始值	描述
15:10	Reserved	R	0h	保留
9	TIMEPOS	R/W	1h	时间戳位置 0: SOF 1: EOF TIMEPOS只能在TIMEEN=0的情况下更改，但可以使用设置TIMEEN=1的相同写入访问权限修改TIMPOS。
8	TIMEEN	R/W	0h	时间戳使能 0: 禁用 1: 使能
7:6	Reserved	R	0h	保留
5	SELMASK	RW	0h	选择接收掩码 0: 寄存器ACF_x指向接收代码 1: 寄存器ACF_x指向接收掩码。 ACFADR选择了一个特定的验收过滤器
4	Reserved	R	0h	保留
3:0	ACFADR	RW	0h	接收过滤器地址 ACFADR指向一个特定的接收过滤器。可以使用寄存器ACF_x访问所选的过滤器。位SELMASK在接收滤波器的代码和掩码之间进行选择。 ACFADR>ACF_NUMBER-1的值是无意义的，并且自动被视为值ACF_NUMBER-1。

4.7.14.14 接收滤波使能寄存器 (ACF_EN)

接收滤波使能寄存器用于使能滤波器的滤波功能，一共 16 个滤波器，每一位的使能，对应一个滤波参数。

接受代码/掩码 (ACF0~1) 在 DSRAM 中，ACF0~1 显示的默认值都是 DSRAM 中的随机值，下面寄存器中描述的默认值 0x0，是在 DSRAM 上电后所有值都是 0x0 的情况，DSRAM 中 ACF0~1 的值在 CANFD 初始化，需要按照寄存器描述的默认值进行初始化。

接收滤波使能寄存器映射到 CANFD 内部的 DPSRAM (640x32bit) 的地址空间分布。

表 4-52 滤波器映射表

地址	名称
608	AMASK1
609	ACF1
610	AMASK2
611	ACF2
612	AMASK3
613	ACF3
614	AMASK4
615	ACF4

616	AMASK5
617	ACF5
618	AMASK6
619	ACF6
620	AMASK7
621	ACF7
622	AMASK8
623	ACF8
624	AMASK9
625	ACF9
626	AMASK10
627	ACF10
628	AMASK11
629	ACF11
630	AMASK12
631	ACF12
632	AMASK13
633	ACF13
634	AMASK14
635	ACF14
636	AMASK15
637	ACF15
638	AMASK16
639	ACF16

接收滤波器使能寄存器，具体说明如下。

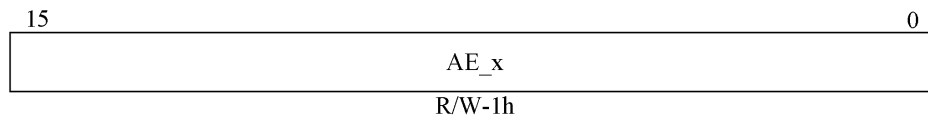


图 4-56 接收滤波器使能寄存器

表 4-53 接收滤波器使能寄存器位域描述

位域	位名	类型	初始值	描述
15:0	AE	R/W	1h	接收过滤器使能 1: 接收过滤器启用 0: 接收过滤器禁用 每个验收过滤器（AMASK / ACODE）都可以单独启用或禁用。在硬件重置后，默认只是使能滤波器0。 已禁用的筛选器将拒绝消息。如果适当的AMASK / ACODE配置匹配，则只有已启用的筛选器才能接受消息。 为了接受所有消息，必须通过设置AE_x=1、AMASK_x=0xff和ACODE_x=0x00来启用一个过滤器x。这是在禁用所有其他过滤器x=0的硬件重置后的默认配置。

4.7.14.15 接收代码/掩码寄存器 0 (ACF_0)

接收代码/掩码寄存器 0 通过 SELMASK 位配置接收代码或者掩。

该寄存器的接收代码寄存器 0 具体说明如下。

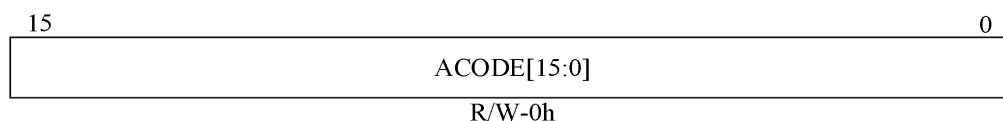


图 4-57 接收代码寄存器 0

表 4-54 接收代码寄存器0位域描述

位域	位名	类型	初始值	描述
15:0	ACODE[15:0]	R/W	0h	接受CODE 1: ACC位值与接收消息的ID位进行比较 0: ACC位值与接收消息的ID位进行比较 ACODE_x (10: 0) 将用于标准帧。 ACODE_x (15: 0) 将用于扩展帧。 只有过滤器0会受到开机复位的影响。所有其他过滤器都保持未初始化。 注意: 在CFG_STAT.RESET为1时, 可写。

该寄存器的接收掩码寄存器 0 具体说明如下。

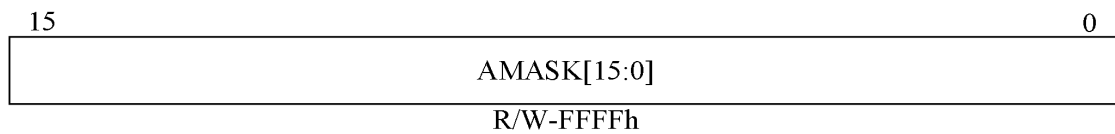


图 4-58 接收掩码寄存器 0

表 4-55 接收掩码寄存器0位域描述

位域	位名	类型	初始值	描述
15:0	AMASK[15:0]	R/W	0h	接受掩码 1: 这些接收标识符的禁用 0: 这些接收标识符的启用 AMASK_x (10: 0) 将用于标准帧。 AMASK_x (15: 0) 将用于扩展帧。 禁用的位会导致接受该消息。因此, 过滤器0重置后的默认配置接受所有消息。 只有过滤器0会受到开机复位的影响。所有其他过滤器都保持未初始化。 注意: 接收掩码, AMASK与ACODE是配套, 滤波ID与帧ID可以放过的位, 掩码对应为1, 则不用考虑滤波ID与帧ID不匹配的位 如果帧ID与滤波ID (ACODE) 异或, 存在差异有非0位, 而掩码 (AMASK) 对应非0位, 为1, 则该不匹配的帧ID, 可以被接收 注意: 在CFG_STAT.RESET为1时, 可写。

4.7.14.16 接收代码/掩码寄存器 1 (ACF_1)

接收代码/掩码寄存器 1 通过 SELMASK 位配置接收代码或者掩。

该寄存器的接收代码寄存器 1 具体说明如下。

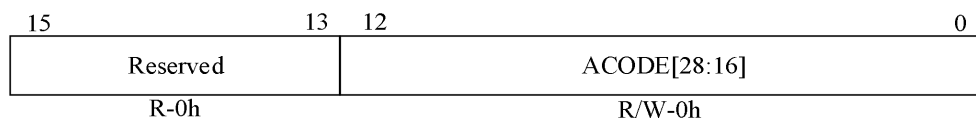


图 4-59 接收代码寄存器 1

表 4-56 接收代码寄存器1位域描述

位域	位名	类型	初始值	描述
15:13	Reserved	R	0h	保留
12:0	ACODE[28:16]	R/W	0h	接受CODE 1: ACC位值与接收消息的ID位进行比较 0: ACC位值与接收消息的ID位进行比较 ACODE_x (28:16) 将用于扩展帧。 只有过滤器0会受到开机复位的影响。所有其他过滤器都保持未初始化。 注意: 在CFG_STAT.RESET为1时, 可写。

该寄存器的接收掩码寄存器 1 具体说明如下。

15	14	13	12	0
Reserved	AIDEE	AIDE	AMASK[28:16]	
R-0h	R/W-0h	R/W-0h	R/W-1FFFh	

图 4-60 接收掩码寄存器 1

表 4-57 接收掩码寄存器1位域描述

位域	位名	类型	初始值	描述
15	Reserved	R	0h	保留
14	AIDEE	R/W	0h	接收掩码IDE位检查启用 1: 验收过滤器接受AIDE定义的标准或扩展标准 0: 验收过滤器同时接受标准帧或扩展帧 只有过滤器0会受到开机复位的影响。所有其他过滤器都保持未初始化。 注意: 在CFG_STAT.RESET为1时, 可写。
13	AIDE	R/W	0h	接受掩码IDE位值 如果AIDEE=1: 1: 接受过滤器只接受扩展帧 0: 接受过滤器只接受标准帧 只有过滤器0受到开机复位的影响。所有其他过滤器都保持未初始化。 只有AIDEE=1, 配置AIDE才有效 注意: 在CFG_STAT.RESET为1时, 可写。
12:0	AMASK[28:16]	R/W	0h	接受掩码 1: 这些接收标识符的禁用 0: 这些接收标识符的启用 AMASK_x (28:16) 将用于扩展帧。 禁用的位会导致接受该消息。因此, 过滤器0重置后的默认配置接受所有消息。 只有过滤器0会受到开机复位的影响。所有其他过滤器都保持未初始化。 注意: 接收掩码, AMASK与ACODE是配套, 滤波ID与帧ID可以放过的位, 掩码对应为1, 则不用考虑滤波ID与帧ID不匹配的位 如果帧ID与滤波ID (ACODE) 异或, 存在差异有非0位, 而掩码 (AMASK) 对应非0位, 为1, 则该不匹配的帧ID, 可以被接收 注意: 在CFG_STAT.RESET为1时, 可写。

4.7.14.17 时间触发配置和发送槽指针寄存器 (TTCFG_TBPTTR)

时间触发配置和发送槽指针寄存器用于配置时间触发, 以及 TB 插槽指针, 具体说明如下。

15	14	13	12	11	10	9	8
WITE	WITF	TTEIF	TTIE	TTIF	T_PRESC		TTEN
R/W-1h	R/W1C-0h	R/W1C-0h	R/W-1h	R/W-0h	R/W-0h		R/W-0h
7	6	5	0				
TBE	TBF	TBPTR					
R/W-0h	R/W-0h	R/W-0h					

图 4-61 时间触发配置和发送槽指针寄存器

表4-58 时间触发配置和发送槽指针寄存器位域描述

位域	位名	类型	初始值	描述
15	WITE	R/W	1h	看门狗触发器中断启用
14	WITF	R/W1C	0h	如果周期计数达到由TT_WTRIG定义的限制，并设置了WTIE，则将设置观察触发中断标志WTIF。 注意：在CFG_STAT.RESET为1时，保持默认值。
13	TEIF	R/W1C	0h	触发错误中断标志将设置TEIF时的条件，在第6.4章中定义。没有任何位来启用或禁用TEIF的处理。 注意：在CFG_STAT.RESET为1时，保持默认值。
12	TTIE	R/W	1h	时间触发中断使能，如果设置了TTIE，那么如果周期时间等于触发时间TT_TRIG，则将设置TTIF。
11	TTIF	R/W1C	0h	如果设置了TTIE，且周期时间等于触发时间TT_TRIG，则将设置时间触发中断标志TTIF。写1到TTIF就会重置它。写0没有影响如果TT_TRIG没有更新，则在下一个基本周期中将不再设置TTIF。 注意：在CFG_STAT.RESET为1时，保持默认值。
10:9	T_PRESC	R/W	0h	TTCAN定时器分频 0:1 1:2 2:4 3:8 TTCAN时间基是由S_PRESC、S_SEG1和S_SEG2定义的CANFD位时间。使用T_PRESC，额外的预调整因子定义为1、2、4或8。 T_PRESC只能在TTEN=0时进行修改，但可以通过一次写访问同时修改T_PRESC和设置TTEN。
8	TTEN	R/W	0h	时间触发器启用 1: TTCAN已启用，计时器正在运行 0: 已禁用 注意：在CFG_STAT.RESET为1时，保持默认值。
7	TBE	R/W	0h	将TB槽设置为“空” 1: TBPTR选择的槽标记为“空” 0: 无效 一旦插槽被标记为空，TSFF=0，TBE将自动重置为0。如果来自该插槽的传输是有效的，那么只要传输完成或在传输错误或仲裁丢失之后传输不再有效，TBE就会保持设置。如果TBF和TBE都被设置，那么TBE优先设置。 注意：在CFG_STAT.RESET为1时，保持默认值。
6	TBF	R/W	0h	将TB槽设置为“填充” 1: TBPTR选择的槽应标记为“填充” 0: 无动作 一旦插槽被标记为填充和TSFF=1，TBF将自动重置为0。如果TBF和TBE都被设置，那么TBE优先设置。 注意：在CFG_STAT.RESET为1时，保持默认值。
5:0	TBPTR	R/W	0h	指向TB消息插槽的指针。 0x00: 指向TBUF0 其他: 指向TBUF1~15的槽 TBPTR指向的消息槽使用TBUF寄存器可读/可写。写访问只有在TSFF=0时才能实现。将TBF设置为1将选定的槽标记为填充，将TBE设置为1将所选槽标记为空。 TBSEL和TSNEXT在TTCAN模式下未使用，没有任何意义。 TBPTR只能指向硬件中存在的缓冲区插槽。TBPTR的不可用位被固定为0。

				TBPTR仅限于TBUF0和63个TBUF1~15插槽。在TTCAN模式下不能使用更多的插槽。如果TBPTR太大，并且指向一个不可用的插槽，那么TBF和TBE将被自动重置，并且不会发生任何操作。
--	--	--	--	---

4.7.14.18 参考 ID 寄存器 0 (REFID_0)

参考 ID 寄存器 0 用于定时触发模式下，更新触发时间，该寄存器具体内容如下。

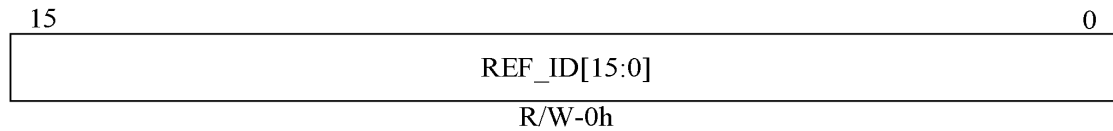


图 4-62 参考 ID 寄存器 0

表 4-59 参考ID寄存器0位域描述

位域	位名	类型	初始值	描述
15:0	REF_ID[15:0]	R/W	0h	参考消息ID标识符。如果REF_IDE为1: REF_ID (15: 0) 有效 (扩展ID) 0: REF_ID (10: 0) 有效 (标准ID) REF_ID在TTCAN模式下用于检测引用消息。这适用于时间从机 (接收) 和时间主机 (传输)。如果检测到引用消息并且没有错误，则此帧的Sync_Mark将成为Ref_Mark。 REF_ID (2: 0) 没有被测试，因此适当的寄存器位被迫为0。 CANFD控制器仅通过ID识别引用消息。附加注意：时间主控将以与普通帧相同的方式传输参考消息。REF_ID用于检测参考消息的成功传输。

4.7.14.19 参考 ID 寄存器 1 (REFID_1)

参考 ID 寄存器 1 用于定时触发模式下，更新触发时间，该寄存器具体内容如下。

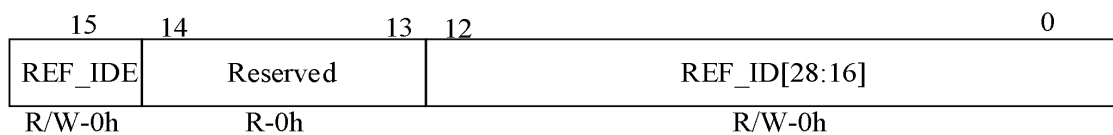


图 4-63 参考 ID 寄存器 1

表 4-60 参考ID寄存器1位域描述

位域	位名	类型	初始值	描述
15	REF_IDE	R/W	0h	参考IDE位
14:13	Reserved	R	0h	保留位
15:0	REF_ID[15:0]	R/W	0h	参考消息ID标识符。如果REF_IDE为1: REF_ID (28: 16) 有效 (扩展ID) 0: 保留位 REF_ID在TTCAN模式下用于检测引用消息。这适用于时间从机 (接收) 和时间主机 (传输)。如果检测到引用消息并且没有错误，则此帧的Sync_Mark将成为Ref_Mark。CANFD控制器仅通过ID识别引用消息。附加注意：时间主控将以与普通帧相同的方式传输参考消息。REF_ID用于检测参考消息的成功传输。

4.7.14.20 触发配置寄存器 (TRIG_CFG)

用于配置时间触发的触发类型以及触发的插槽，该寄存的具体说明如下。

15	12	11	10	8	7	6	5	0
TEW			Reserved	TTYPE		Reserved	TTPTR	
R/W-0h			R-0h	R/W-0h		R-0h	R/W-0h	

图 4-64 触发配置寄存器

表 4-61 触发配置寄存器位域描述

位域	位名	类型	初始值	描述
15:12	TEW	R/W	0h	触发发送的窗口延迟时间,在定时触发模式TTYPE=2，独占时间窗口模式的条件下，允许帧启动的周期时间有1~16个周期的延迟。TWE+1定义了延迟的周期。 TEW=0是一个有效的设置，并将传输启用窗口缩短为1。 如果定义的触发时间TT_TRING + TWE+1等于周期时间，如果还未传输，则配置TSA位，停止本次触发送。
11	Reserved	R	0h	保留位
10:8	TTYPE	R/W	0h	触发器类型 0: 对于立即传输的立即触发器 1: 对接收触发器的时间触发器 2: 对于用于独占时间窗口的单次发射触发器 3: 合并仲裁时间窗口的传输启动触发器 4: 用于合并仲裁时间窗口的传输停止触发器 其他-无效 触发器的时间由TT_TRIG定义。TTPTR为传输触发器选择TB插槽。
7:6	Reserved	R	0h	保留位
5:0	TTPTR	R/W	0h	传输触发器TB插槽指针 如果TTPTR太大并指向不可用的插槽，则设置TEIF，在TT_TRIG_1进行写访问后不能激活新的触发器。 如果TTPTR指向一个空插槽，则将在达到触发时间时设置TEIF。

4.7.14.21 触发时间寄存器 (TT_TRIG)

触发时间寄存器用于配置触发时间，具体说明如下。

15	0
TT_TRIG	
R/W-0h	

图 4-65 触发时间寄存器

表 4-62 触发时间寄存器位域描述

位域	位名	类型	初始值	描述
15:0	TT_TRIG	R/W	0h	触发器时间TT_TRIG (15: 0) 定义了触发器的周期时间。对于一个传输触发器，适当帧的SOF的最早传输点将为TT_TRIG+1。

4.7.14.22 看门狗触发寄存器 (TT_WTRIG)

看门狗触发时间寄存器用于配置看门狗触发时间，具体说明如下。



图 4-66 看门狗触发时间寄存器

表 4-63 看门狗触发时间寄存器位域描述

位域	位名	类型	初始值	描述
15:0	TT_WTRIG	R/W	0h	看门狗触发器时间 T_WTRIG (15: 0) 定义了一个监视触发器的周期时间。初始观察触发器是最大周期时间0xffff。

4.7.14.23 CANFD 控制寄存器 (CANCFG)

CANFD 控制寄存器主要用于显示 CANFD 的休眠状态，ECC 使能和 CAN_CLK 时钟选择，具体描述如下。

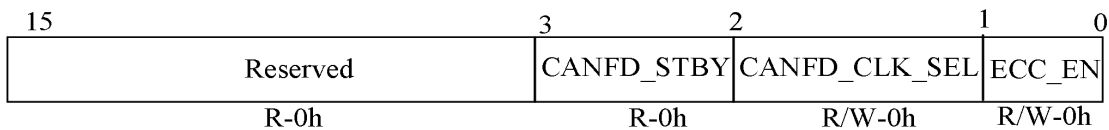


图 4-67 CANFD 控制寄存器

表 4-64 CANFD 控制寄存器位域描述

位域	位名	类型	初始值	描述
15:3	Reserved	R	0h	保留位
2	CANFD_STBY	R	0h	CANFD休眠模式状态 0: 无休眠 1: 休眠状态
1	CANFD_CLK_SEL	R/W	0h	CANFD频率选择 0: SYS时钟频率 1: PLL时钟频率 注意：在CFG_STAT.RESET为1时，可写。
0	ECC_EN	R/W	0h	DSRAM存储的ECC使能 0: 禁用ECC校验 1: 启用ECC校验 注意：如果系统启动自检，则该位配置禁用ECC无效，状态为启动ECC校验。 只有在CFG_STAT.RESET=1的条件下才启动 注意：在CFG_STAT.RESET为1时，可写。

4.7.14.24 TIMER64 定时器寄存器

TIMER64 是 64 位计数值，32 位分频的定时器，用于产生固定时间周期。

4.7.14.24.1 定时器计数器 0 寄存器 (TIMERTIM0)

定时器计数器 0 寄存器是 64 位定时的 15~0 位，配置计数器启动值，同时也显示当前的计数值，每 (PSC) 时钟计数器递增 1，其中 TDDR 是计时器 TIM 预分频值，该寄存器具体说明如下。

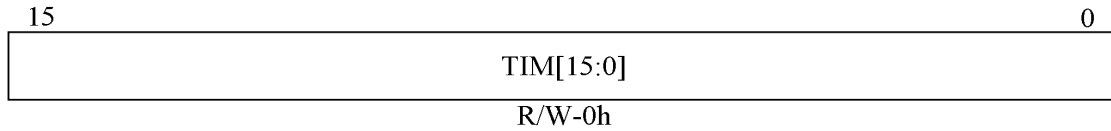


图 4-68 定时器计数器 0 寄存器

表 4-65 定时器计数器 0 寄存器位域描述

位域	位名	类型	初始值	描述
15:0	TIM[15:0]	R/W	0h	定时器计数器寄存器 TIM0寄存器持有当前计时器的64位计数的15~0位。

4.7.14.24.2 定时器计数器 1 寄存器 (TIMERTIM1)

定时器计数器 1 寄存器是 64 位定时的 31~16 位，配置计数器启动值，同时也显示当前的计数值，寄存器具体说明如下。

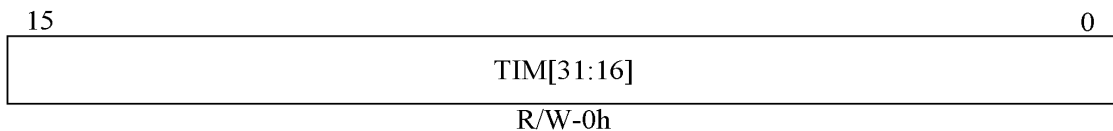


图 4-69 定时器计数器 1 寄存器

表 4-66 定时器计数器 1 寄存器位域描述

位域	位名	类型	初始值	描述
15:0	TIM[31:16]	R/W	0h	定时器计数器寄存器 计数器寄存器持有当前计时器的64位计数的低31~16位。

4.7.14.24.3 定时器计数器 2 寄存器 (TIMERTIM2)

定时器计数器 2 寄存器是 64 位定时的 47~32 位，配置计数器启动值，同时也显示当前的计数值具体说明如下。

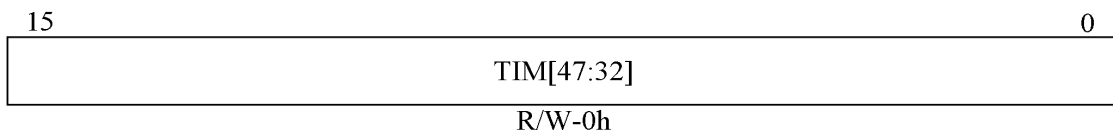


图 4-70 定时器计数器 2

表 4-67 定时器计数器 2 寄存器位域描述

位域	位名	类型	初始值	描述
15:0	TIM[47:32]	R/W	0h	定时器计数器寄存器 计数器寄存器持有当前计时器的64位计数的47~32位。

4.7.14.24.4 定时器计数器 3 寄存器 (TIMERTIM3)

定时器计数器 3 寄存器是 64 位定时的 63~48 位，配置计数器启动值，同时也显示当前的计数值具体说明如下。

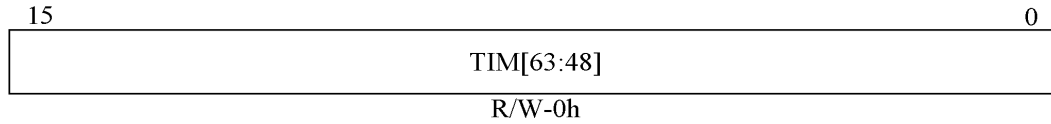


图 4-71 定时器计数器 3

表 4-68 定时器计数器 3 寄存器位域描述

位域	位名	类型	初始值	描述
15:0	TIM[63:48]	R/W	0h	定时器计数器寄存器 计数器寄存器持有当前计时器的64位计数的63~48位。

4.7.14.24.5 定时器周期 0 寄存器 (TIMERPRD0)

定时器周期寄存器，用于限定计数器的计数范围，当计数到周期值，计数器如果在自由模式下，计数会从 0 开始计数，具体说明如下。

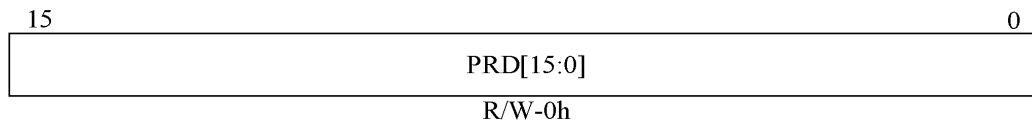


图 4-72 定时器周期 0 寄存器

表 4-69 定时器周期 0 寄存器位域描述

位域	位名	类型	初始值	描述
15:0	PRD[15:0]	R/W	0h	定时器周期器寄存器 周期寄存器持有当前计数周期的64位计数的15~0位。

4.7.14.24.6 定时器周期 1 寄存器 (TIMERPRD1)

定时器周期寄存器，用于限定计数器的计数范围，当计数到周期值，计数器如果在自由模式下，计数会从 0 开始计数，具体说明如下。

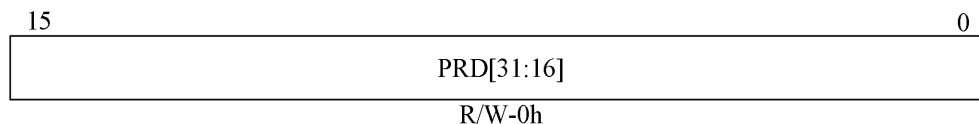


图 4-73 定时器周期 0 寄存器

表 4-70 定时器周期 0 寄存器位域描述

位域	位名	类型	初始值	描述
15:0	PRD[31:16]	R/W	0h	定时器周期器寄存器 周期寄存器持有当前计数周期的64位计数的31~16位。

4.7.14.24.7 定时器周期 2 寄存器 (TIMERPRD2)

定时器周期寄存器，用于限定计数器的计数范围，当计数到周期值，计数器如果在自由模式下，计数会从 0 开始计数，具体说明如下。

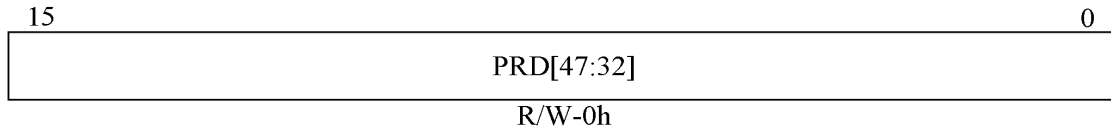


图 4-74 定时器周期 0 寄存器

表 4-71 定时器周期 0 寄存器位域描述

位域	位名	类型	初始值	描述
15:0	PRD[47:32]	R/W	0h	定时器周期器寄存器 周期寄存器持有当前计数周期的64位计数的47~32位。

4.7.14.24.8 定时器周期 3 寄存器 (TIMERPRD3)

定时器周期寄存器，用于限定计数器的计数范围，当计数到周期值，计数器如果在自由模式下，计数会从 0 开始计数，具体说明如下。

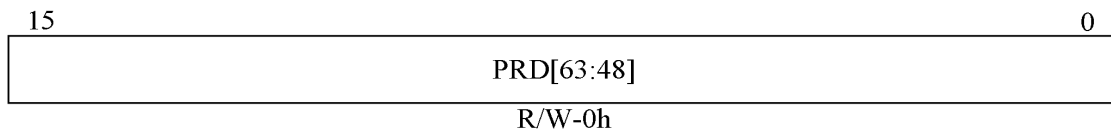


图 4-75 定时器周期 0 寄存器

表 4-72 定时器周期 0 寄存器位域描述

位域	位名	类型	初始值	描述
15:0	PRD[63:48]	R/W	0h	定时器周期器寄存器 周期寄存器持有当前计数周期的64位计数的63~48位。

4.7.14.24.9 定时器控制寄存器 (TIMERTCR)

定时器控制寄存器控制定时器工作模式，以及暂停控制，具体说明如下。

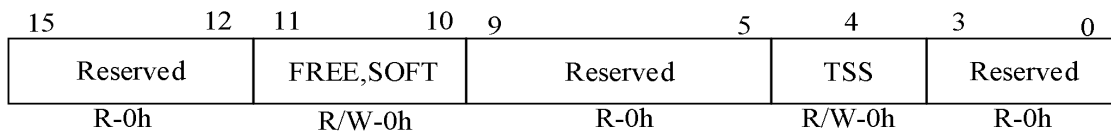


图 4-76 定时器控制寄存器

表 4-73 定时器控制寄存器位域描述

位域	位名	类型	初始值	描述
15:12	Reserved	R	0h	保留
11:10	FREE,SOFT	R/W	0h	定时器仿真停模式： 0: TIMH: TIM立即停止（硬停止） 1: TIMH: TIM递增到周期，停止（软停止）

				2: 自由运行 3: 自由运行
9:5	Reserved	R	0h	保留
4	TSS	R/W	0h	定时器停止位: TSS是一个1位的标志, 它会停止或启动定时器。 0: 表示定时器正在运行。要启动或重新启动定时器, 请将TSS 设置为0。复位后, TSS被清除为0, 定时器立即启动。 1: 停止定时器。
3:0	Reserved	R	0h	保留

4.7.14.24.10 定时器预分频寄存器 (TIMERTPR0)

定时器分频寄存器, 一共 64 位, 每个 (PSC) 计时器时钟源周期, 计时器计数器寄存器递增 1, 并且更新 PSC 的值, 也可以通过 RELOAD 位的配置使能 PSC 立即加载 TDDR 的值, 具体说明如下。

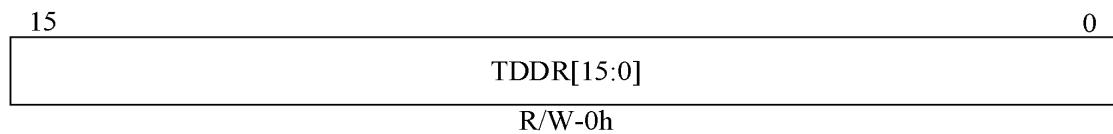


图 4-77 定时器预分频寄存器

表 4-74 定时器预分频寄存器位域描述

位域	位名	类型	初始值	描述
15:0	TDDR[15:0]	R/W	0h	定时器分频寄存器。每个 (PSC) 计时器时钟源周期, 计时器计数器寄存器递增1。 在复位时, TDDR位被清除为0。 当预计算器计数器 (PSC) 值为0时, 一个计时器时钟源周期之后, TDDR的内容将重新加载到PSC, 计时器计数器寄存器将减少1。

4.7.14.24.11 定时器分频寄存器 (TIMERTPR1)

每次更新 PSC 的值, 可以通过 RELOAD 位的配置使能 PSC 立即加载 TDDR 的值, 也可以等每个 (PSC) 计时器时钟源周期, PSC 加载 TDDR 寄存器的值, 具体说明如下。

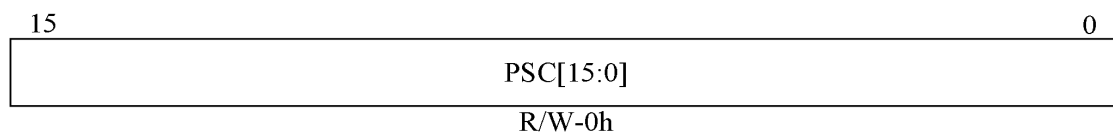


图 4-78 定时器分频寄存器

表 4-75 定时器分频寄存器位域描述

位域	位名	类型	初始值	描述
15:0	PSC[15:0]	R/W	0h	定时器分频器: PSC加载了TDDR的内容, 计时器计数器寄存器减1。 PSC可以通过读取寄存器来检查, 但不能直接设置。它必须从定时器定时器分频寄存器 (TDDR) 获取其值。 在复位时, PSC被设置为0。

4.7.14.25 TRAN 的 32 位计时器寄存器

TRAN 的 32 位计时器，用于退出休眠模式后的 CAN FD 的唤醒时间设定。

4.7.14.25.1 传输控制器计数器 0 寄存器 (TRANCNT0)

计数器 0 寄存器是 32 位定时的 15~0 位，配置计数器启动值，同时也显示当前的计数值，每 (PSC) 时钟计数器递增 1，其中 TDDR 是计时器 TIM 预分频值，该寄存器具体说明如下。

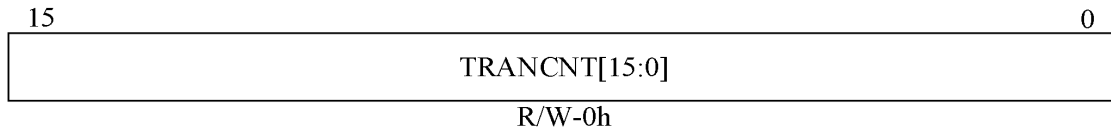


图 4-79 传输控制器计数器0寄存器

表 4-76传输控制器计数器0寄存器位域描述

位域	位名	类型	初始值	描述
15:0	TRANCNT[15:0]	R/W	0h	传输控制器待机模式的计数器寄存器 (TRAN) : TRAN每TDDR+1时钟周期递增一个，其中TDD是计时器TRAN 预分频值。 当TRAN减小为零时，TRAN寄存器将重新加载包含在PRD寄存器 中的周期值。

4.7.14.25.2 传输控制器计数器 1 寄存器 (TRANCNT1)

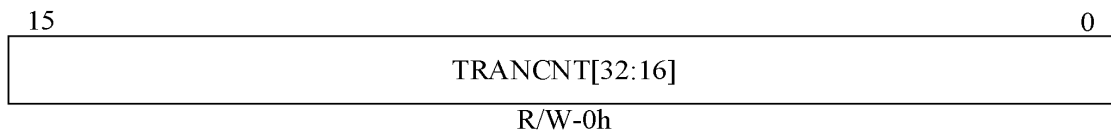


图 4-80 传输控制器计数器1寄存器

表 4-77传输控制器计数器1寄存器位域描述

位域	位名	类型	初始值	描述
15:0	TRANCNT[31:16]	R/W	0h	传输控制器待机模式的计数器寄存器 (TRAN) : TRAN每TDDR+1时钟周期递增一个，其中TDD是计时器TRAN 预分频值。 当TRAN减小为零时，TRAN寄存器将重新加载包含在PRD寄存器 中的周期值。

4.7.14.25.3 传输控制器周期 0 寄存器 (TRANPRD0)

周期寄存器，用于限定计数器的计数范围，当计数到周期值，达到设定的唤醒时间，CANFD 唤醒，具体说明如下。

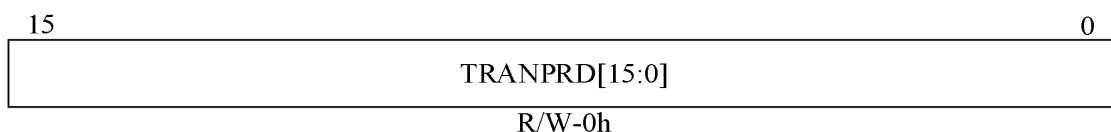


图 4-81 传输控制器周期 0 寄存器

表 4-78 传输控制器计数器 0 寄存器位域描述

位域	位名	类型	初始值	描述
15:0	TRANPRD[15:0]	R/W	0h	周期器寄存器 周期寄存器持有当前计数周期的32位计数的15~0位。

4.7.14.25.4 传输控制器周期 1 寄存器 (TRANPRD1)

周期寄存器，用于限定计数器的计数范围，当计数到周期值，达到设定的唤醒时间，CANFD 唤醒，具体说明如下。

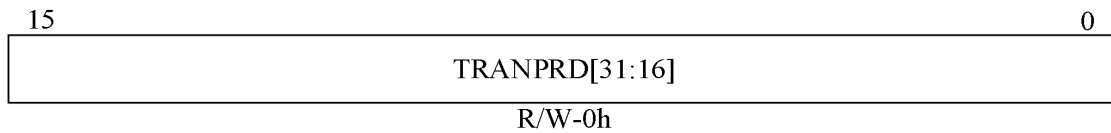


图 4-82 传输控制器周期 1 寄存器

表 4-79 传输控制器计数器 1 寄存器位域描述

位域	位名	类型	初始值	描述
15:0	TRANPRD[15:0]	R/W	0h	周期器寄存器 周期寄存器持有当前计数周期的32位计数的15~0位。

4.7.14.25.5 传输控制器预分频寄存器 (TRANTPR0)

预分频寄存器，一共 16 位，每个 (PSC) 计时器时钟源周期，计时器计数器寄存器递增 1，并且更新 PSC 的值，加载 TDDR 的值，具体说明如下。

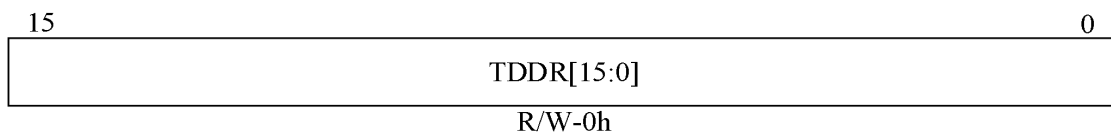


图 4-83 传输控制器预分频寄存器

表 4-80 传输控制器预分频位域描述

位域	位名	类型	初始值	描述
15:0	TDDR[15:0]	R/W	0h	定时器分频寄存器。每个 (PSC) 计时器时钟源周期，计时器计数器寄存器递增1。 在复位时，TDDR位被清除为0。 当预计算器计数器 (PSC) 值为0时，一个计时器时钟源周期之后，TDDR的内容将重新加载到PSC，TRANCNT将减少1。

4.7.14.25.6 传输控制器分频寄存器 (TRANTPR1)

每次更新 PSC 的值，等每个 (PSC) 计时器时钟源周期，PSC 加载 TDDR 寄存器的值，具体说明如下。

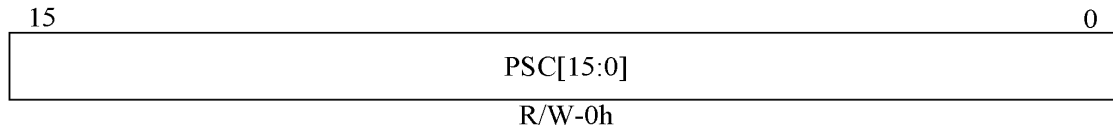


图 4-84 传输控制器预分频寄存器

表 4-81 传输控制器预分频位域描述

位域	位名	类型	初始值	描述
15:0	PSC[15:0]	R/W	0h	定时器分频器： PSC加载了TDDR的内容，计时器计数器寄存器减少了1。 PSC可以通过读取寄存器来检查，但不能直接设置。它必须从定时器定时器分频寄存器（TDDR）获取其值。 在复位时，PSC被设置为0。

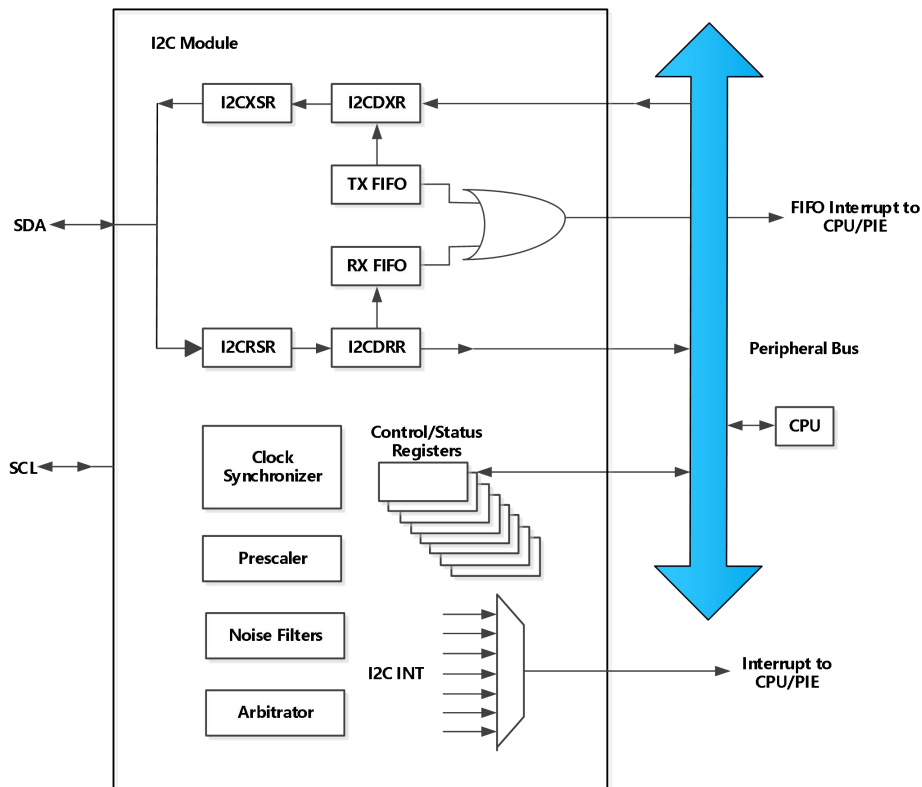
4.8 串行 I2C 模块

此器件包含一个 I2C 串行端口。I2C 结构如图 4-85 所示。

I2C 模块具有以下特性：

- 符合飞利浦 I2C 总线规范（版本 2.1）：
 - 支持 1 至 8 位格式传输
 - 7 位和 10 位寻址模式
 - 广播
 - START 字节模式
 - 支持多个主发送器和从接收器
 - 支持多个从发送器和主接收器
 - 组合主器件发送/接收和接收/发送模式
 - 数据传输速率从 10kbps 到高达 400kbps（I2C 快速模式速率）
- 一个 4 级接收 FIFO 和一个 4 级发送 FIFO
- 可以由 CPU 使用的中断。此中断可由下列条件之一生成：
 - 发送数据就绪
 - 接收数据就绪
 - 寄存器访问就绪

- 无应答
- 仲裁丢失
- 检测到停止条件
- 被寻址为从器件
- 在 FIFO 模式下，CPU 可以使用附加的中断
- 模块启用/禁用能力
- 自由数据格式模式



A.以 SYSCLKOUT 速率对 I2C 寄存器进行访问。I2C 端口的内部时序和信号波形也为 SYSCLKOUT 速率。

B.PCLKCR0 寄存器内的时钟使能位 (I2CAENCLK) 关闭到 I2C 端口的时钟以实现低功耗运行。复位时，I2CAENCLK 清除，这表明外设内部时钟被关闭。

图4-85 I2C外设模块接口

表 4-82 中的寄存器配置并且扩展 I2C 端口操作。

表 4-82 I2C-A 寄存器

名称	地址	受 EALLOW 保护	说明
I2COAR	0x7900	否	I2C 自身的地址寄存器
I2CIER	0x7901	否	I2C 中断使能寄存器
I2CSTR	0x7902	否	I2C 状态寄存器

I2CCLKL	0x7903	否	I2C 时钟低电平时间分频器寄存器
I2CCLKH	0x7904	否	I2C 时钟高电平时间分频器寄存器
I2CCNT	0x7905	否	I2C 数据计数寄存器
I2CDRR	0x7906	否	I2C 数据接收寄存器
I2CSAR	0x7907	否	I2C 从器件地址寄存器
I2CDXR	0x7908	否	I2C 数据发送寄存器
I2CMDR	0x7909	否	I2C 模式寄存器
I2CISRC	0x790A	否	I2C 中断源寄存器
I2CPSC	0x790C	否	I2C 预分频器寄存器
I2CFFTX	0x7920	否	I2C FIFO 发送寄存器
I2CFFRX	0x7921	否	I2C FIFO 接收寄存器
I2CRSR	-	否	I2C 接收移位寄存器（CPU不可访问）
I2CXSR	-	否	I2C 发送移位寄存器（CPU不可访问）

4.8.1 I2C 电气特性和时序

表4-83 I2C开关特性

	测试条件	最小值	最大值	单位
f _{SCL} SCL 时钟频率	I2C 时钟模块频率介于 7MHz 到 12MHz 之间，且对 I2C 预分频器和时钟分频器寄存器进行了适当配置。		400	kHz
V _{il} 低电平输入电压			0.3V _{DDIO}	V
V _{ih} 高电平输入电压		0.7V _{DDIO}		V
V _{hys} 输入滞后		0.05V _{DDIO}		V
V _{ol} 低电平输出电流	3mA 灌电流	0	0.4	V
t _{LOW} SCL 时钟的低周期	I2C 时钟模块频率介于 7MHz 到 12MHz 之间，且对 I2C 预分频器和时钟分频器寄存器进行了适当配置。	1.3		μs
t _{HIGH} SCL 时钟的高周期	I2C 时钟模块频率介于 7MHz 到 12MHz 之间，且对 I2C 预分频器和时钟分频器寄存器进行了适当配置	0.6		μs
I _I 输入电压介于 0.1V _{DDIO} 到 0.9V _{DDIO} （最大值）的输入电流		-10	10	μA

4.9 增强型 PWM 模块 (ePWM1/2/3/4/5)

此器件包含 5 个增强型 PWM 模块 (ePWM)。图 4-86 显示了 ePWM 模块框图。图 4-87 显示了与 ePWM 互连的信号。

表 4-84 和表 4-85 显示了每个模块的完整的 ePWM 寄存器集。

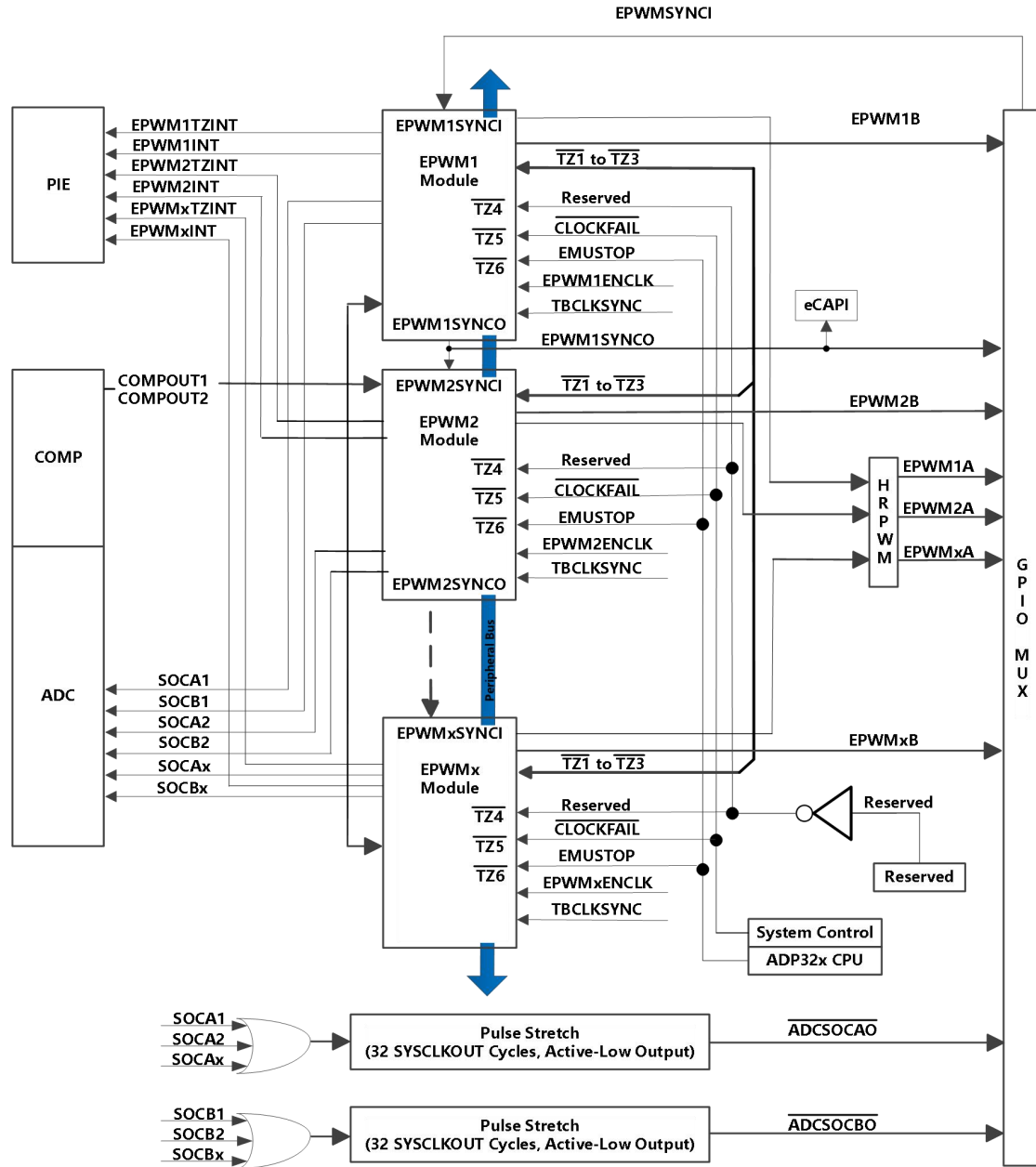


图 4-86 ePWM 框图

表 4-84 ePWM1-ePWM4 控制和状态寄存器

名称	ePWM1	ePWM2	ePWM3	ePWM4	大小(x16)/ #SHADOW	说明
TBCTL	0x6800	0x6840	0x6880	0x68C0	1/0	时基控制寄存器
TBSTS	0x6801	0x6841	0x6881	0x68C1	1/0	时基状态寄存器
TBPHSHR	0x6802	0x6842	0x6882	0x68C2	1/0	时基相位 HRPWM 寄存器
TBPHS	0x6803	0x6843	0x6883	0x68C3	1/0	时基相位寄存器
TBCTR	0x6804	0x6844	0x6884	0x68C4	1/0	时基计数器寄存器
TBPRD	0x6805	0x6845	0x6885	0x68C5	1/1	时基周期设置寄存器
TBPRDHR	0x6806	0x6846	0x6886	0x68C6	1/1	时基周期高分辨率寄存器 ⁽¹⁾
CMCTL	0x6807	0x6847	0x6887	0x68C7	1/0	计数器比较控制寄存器
CMPAHR	0x6808	0x6848	0x6888	0x68C8	1/1	时基比较 A HRPWM 寄存器
CMPA	0x6809	0x6849	0x6889	0x68C9	1/1	计数器比较 A 设置寄存器
CMPB	0x680A	0x684A	0x688A	0x68CA	1/1	计数器比较 B 设置寄存器
AQCTLA	0x680B	0x684B	0x688B	0x68CB	1/0	用于输出 A 的操作限定器控制寄存器
AQCTLB	0x680C	0x684C	0x688C	0x68CC	1/0	用于输出 B 的操作限定器控制寄存器
AQSFRC	0x680D	0x684D	0x688D	0x68CD	1/0	操作限定器软件强制寄存器
AQCSFRC	0x680E	0x684E	0x688E	0x68CE	1/1	操作限定器连续 S/W 强制设置寄存器
DBCTL	0x680F	0x684F	0x688F	0x68CF	1/1	死区生成器控制寄存器
DBRED	0x6810	0x6850	0x6890	0x68D0	1/0	死区生成器上升沿延迟计数寄存器
DBFED	0x6811	0x6851	0x6891	0x68D1	1/0	死区生成器下降沿延迟计数寄存器
TZSEL	0x6812	0x6852	0x6892	0x68D2	1/0	可编程控制故障区选择寄存器 ⁽¹⁾
TZDCSEL	0x6813	0x6853	0x6893	0x68D3	1/0	可编程控制故障区域数字比较寄存器
TZCTL	0x6814	0x6854	0x6894	0x68D4	1/0	触发区控制寄存器 ⁽¹⁾
TZEINT	0x6815	0x6855	0x6895	0x68D5	1/0	触发区启用中断寄存器 ⁽¹⁾
TZFLG	0x6816	0x6856	0x6896	0x68D6	1/0	可编程控制故障区域标志寄存器 ⁽¹⁾
TZCLR	0x6817	0x6857	0x6897	0x68D7	1/0	触发区清除寄存器 ⁽¹⁾
TZFRC	0x6818	0x6858	0x6898	0x68D8	1/0	触发区强制寄存器 ⁽¹⁾
ETSEL	0x6819	0x6859	0x6899	0x68D9	1/0	事件触发器选择寄存器
ETPS	0x681A	0x685A	0x689A	0x68DA	1/0	事件触发器预分频寄存器
ETFLG	0x681B	0x685B	0x689B	0x68DB	1/0	事件触发器标志寄存器
ETCLR	0x681C	0x685C	0x689C	0x68DC	1/0	事件触发器清除寄存器
ETFRC	0x681D	0x685D	0x689D	0x68DD	1/0	事件触发器强制寄存器
PCCTL	0x681E	0x685E	0x689E	0x68DE	1/0	PWM 斩波器控制寄存器
HRCNFG	0x6820	0x6860	0x68A0	0x68E0	1/0	HRPWM 配置寄存器 ⁽¹⁾

(1) 寄存器受 EALLOW 保护。

表 4-84 ePWM1-ePWM4 控制和状态寄存器(续)

名称	ePWM1	ePWM2	ePWM3	ePWM4	大小(x16)/ #SHADOW	说明
HRPWR	0x6821	-	-	-	1/0	HRPWM 功率寄存器
HRMSTEP	0x6826	-	-	-	1/0	HRPWM MEP 步长寄存器
HRPCTL	0x6828	0x6868	0x68A8	0x68E8	1/0	高分辨率周期控制寄存器 ⁽¹⁾
TBPRDHRM	0x682A	0x686A	0x68AA	0x68EA	1/W ⁽²⁾	时基周期 HRPWM 镜像寄存器
TBPRDM	0x682B	0x686B	0x68AB	0x68EB	1/W ⁽²⁾	时基周期寄存器镜像
CMPAHRM	0x682C	0x686C	0x68AC	0x68EC	1/W ⁽²⁾	比较 A HRPWM 镜像寄存器
CMPAM	0x682D	0x686D	0x68AD	0x68ED	1/W ⁽²⁾	比较 A 镜像寄存器
DCTRIPSEL	0x6830	0x6870	0x68B0	0x68F0	1/0	数字比较触发选择寄存器 ⁽¹⁾
DCACTL	0x6831	0x6871	0x68B1	0x68F1	1/0	数字比较 A 控制寄存器 ⁽¹⁾
DCBCTL	0x6832	0x6872	0x68B2	0x68F2	1/0	数字比较 B 控制寄存器 ⁽¹⁾
DCFCTL	0x6833	0x6873	0x68B3	0x68F3	1/0	数字比较滤波器控制寄存器 ⁽¹⁾
DCCAPCT	0x6834	0x6874	0x68B4	0x68F4	1/0	数字比较捕获控制寄存器 ⁽¹⁾
DCFOFFSET	0x6835	0x6875	0x68B5	0x68F5	1/1	数字比较滤波偏移寄存器
DCFOFFSETCNT	0x6836	0x6876	0x68B6	0x68F6	1/0	数字比较滤波偏移计数器寄存器
DCFWINDOW	0x6837	0x6877	0x68B7	0x68F7	1/0	数字比较滤波窗口寄存器
DCFWINDOWCNT	0x6838	0x6878	0x68B8	0x68F8	1/0	数字比较滤波窗口计数器寄存器
DCCAP	0x6839	0x6879	0x68B9	0x68F9	1/1	数字比较计数器捕获寄存器

(2) W = 写入影子寄存器。

表 4-85 ePWM5 控制和状态寄存器

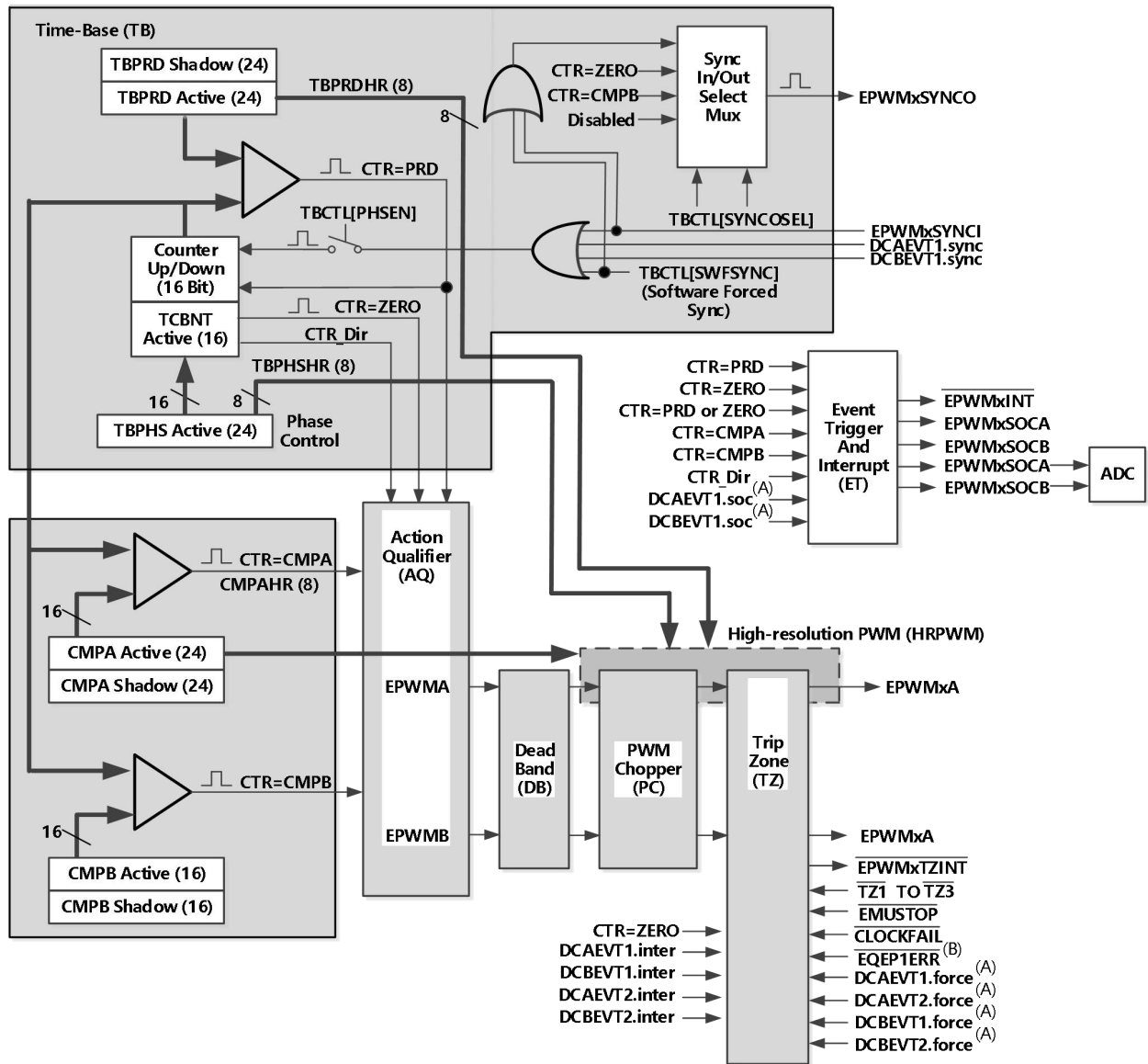
名称	ePWM5	大小(x16)/ #SHADOW	说明
TBCTL	0x6900	1/0	时基控制寄存器
TBSTS	0x6901	1/0	时基状态寄存器
TBPHSHR	0x6902	1/0	时基相位 HRPWM 寄存器
TBPHS	0x6903	1/0	时基相位寄存器
TBCTR	0x6904	1/0	时基计数器寄存器
TBPRD	0x6905	1/1	时基周期设置寄存器
TBPRDHR	0x6906	1/1	时基周期高分辨率寄存器 ⁽¹⁾
CMPCTL	0x6907	1/0	计数器比较控制寄存器
CMPAHR	0x6908	1/1	时基比较 A HRPWM 寄存器
CMPA	0x6909	1/1	计数器比较 A 设置寄存器
CMPB	0x690A	1/1	计数器比较 B 设置寄存器
AQCTLA	0x690B	1/0	用于输出 A 的操作限定器控制寄存器
AQCTLB	0x690C	1/0	用于输出 B 的操作限定器控制寄存器
AQSFR	0x690D	1/0	操作限定器软件强制寄存器
AQCSFR	0x690E	1/1	操作限定器连续 S/W 强制设置寄存器
DBCTL	0x690F	1/1	死区生成器控制寄存器

表 4-85 ePWM5 控制和状态寄存器(续)

名称	ePWM5	大小(x16)/ #SHADOW	说明
DBRED	0x6910	1/0	死区生成器上升沿延迟计数寄存器
DBFED	0x6911	1/0	死区生成器下降沿延迟计数寄存器
TZSEL	0x6912	1/0	触发区选择寄存器 ⁽¹⁾
TZDCSEL	0x6913	1/0	触发区数字比较寄存器
TZCTL	0x6914	1/0	触发区控制寄存器 ⁽¹⁾
TZEINT	0x6915	1/0	触发区启用中断寄存器 ⁽¹⁾
TZFLG	0x6916	1/0	触发区标志寄存器 ⁽¹⁾
TZCLR	0x6917	1/0	触发区清除寄存器 ⁽¹⁾
TZFRC	0x6918	1/0	触发区强制寄存器 ⁽²⁾
ETSEL	0x6919	1/0	事件触发器选择寄存器
ETPS	0x691A	1/0	事件触发器预分频寄存器
ETFLG	0x691B	1/0	事件触发器标志寄存器
ETCLR	0x691C	1/0	事件触发器清除寄存器
ETFRC	0x691D	1/0	事件触发器强制寄存器
PCCTL	0x691E	1/0	PWM 斩波器控制寄存器
HRCNFG	0x6920	1/0	HRPWM 配置寄存器 ⁽²⁾
HRPWR	-	1/0	HRPWM 功率寄存器
HRMSTEP	-	1/0	HRPWM MEP 步长寄存器
HRPCTL	0x6928	1/0	高分辨率周期控制寄存器 ⁽²⁾
TBPRDH RM	0x692A	1/W ⁽³⁾	时基周期 HRPWM 寄存器镜像
TBPRDM	0x692B	1/W ⁽³⁾	时基周期寄存器镜像
CMPAHRM	0x692C	1/W ⁽³⁾	比较 A HRPWM 寄存器镜像
CMPAM	0x692D	1/W ⁽³⁾	比较 A 寄存器镜像
DCTRISEL	0x6930	1/0	数字比较触发选择寄存器 ⁽²⁾
DCACTL	0x6931	1/0	数字比较 A 控制寄存器 ⁽²⁾
DCBCTL	0x6932	1/0	数字比较 B 控制寄存器 ⁽²⁾
DCFCTL	0x6933	1/0	数字比较滤波器控制寄存器 ⁽²⁾
DCCAPCT	0x6934	1/0	数字比较捕获控制寄存器 ⁽²⁾
DCFOFFSET	0x6935	1/1	数字比较滤波偏移寄存器
DCFOFFSETCNT	0x6936	1/0	数字比较滤波偏移计数器寄存器
DCFWINDOW	0x6937	1/0	数字比较滤波窗口寄存器
DCFWINDOWCNT	0x6938	1/0	数字比较滤波窗口计数器寄存器
DCCAP	0x6939	1/1	数字比较计数器捕获寄存器

(1)寄存器受 EALLOW 保护。

(2) W = 写入影子寄存器。



A. 这些事件由基于 COMPxOUT 和 $\overline{TZ}$ 信号电平的 Type - 1 ePWM 数字比较 (DC) 子模块生成。

B. 此信号只存在于带有 eQEP1 模块的器件。

图 4-87 ePWM 子模块和关键内部信号互连

4.9.1 增强型脉宽调制器 (ePWM) 时序

PWM是指ePWM 1-5上的 PWM 输出。表4-86所示为 ePWM 时序要求，表4-87所示为其开关特性。

表 4-86 ePWM 时序要求⁽¹⁾

参数		测试条件	最小值	最大值	单位
$t_{w(SYCIN)}$	同步输入脉冲宽度	异步	$2t_{c(SCO)}$		周期
		同步	$2t_{c(SCO)}$		周期
		带有输入限定器	$1t_{c(SCO)}+t_{w(IQSW)}$		周期

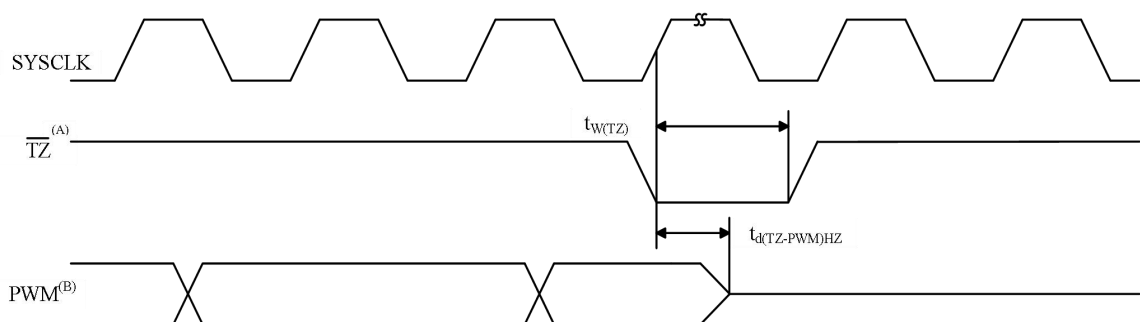
表 4-87 ePWM 开关特性

参数		测试条件	最小值	最大值	单位
$t_w(PWM)$	脉冲持续时间，PWMx 输出高电平/低电平		33.33		ns
$t_w(SYNCOUT)$	同步输出脉冲宽度		$8t_{c(SCO)}$		周期
$t_d(PWM)t_{za}$	触发输入有效到 PWM 强制高电平的延迟时间； 触发输入有效到 PWM 强制低电平的延迟时间；	无引脚负载		25	ns
$t_d(TZ-PWM)HZ$	触发输入有效至 PWM 高阻抗 (Hi-Z) 的延迟时间			20	ns

4.9.2 可编程控制故障区输入时序

表 4-88 可编程控制故障区输入时序要求⁽¹⁾

		最小值	最大值	单位
$t_{w(TZ)}$	异步	$2t_{c(TBCLK)}$		周期
	同步	$2t_{c(TBCLK)}$		周期
	有输入限定器	$2t_{c(TBCLK)}+t_{w(IQSW)}$		周期



A. $\overline{TZ}$ - $\overline{TZ1}$ 、 $\overline{TZ2}$ 、 $\overline{TZ3}$ 、 $\overline{TZ4}$ 、 $\overline{TZ5}$ 、 $\overline{TZ6}$

B. PWM 是指器件内的所有 PWM 引脚。 $\overline{TZ}$ 为高电平之后的 PWM 引脚的状态，取决于 PWM 的软件配置。

图4-88 PWM Hi-Z 特性

4.10 高分辨率 PWM (HRPWM)

HRPWM 模块为高分辨率 PWM，HRPWM 模块的关键点为：

- 大大扩展了数字 PWM 的时间分辨率
- 此功能可同时在单边沿（占空比和相移控制）以及用于频率/周期调制的双边沿控制中使用。
 - 通过对 ePWM 模块的比较 A 和相位寄存器的扩展，实现更精细的时间粒度控制或边沿定位。
 - HRPWM 功能只在 PWM 模块的 A 信号路径上提供（在 EPWMxA 输出上提供）。

EPWMxB 输出为传统 PWM 功能。

注

HRPWM 允许的最小 SYSCLKOUT 频率为 60MHz。

当启用双边沿高分辨率时（高分辨率周期模式），PWMxB 输出不可用。

4.10.1 高分辨率 PWM (HRPWM) 时序

表4-89所示为高分辨率 PWM 的开关特性。

表4-89 高分辨率 PWM开关特性 ⁽¹⁾

	最小值	典型值	最大值	单位
微边沿定位 (MEP) 步长 ⁽²⁾		180 ⁽³⁾	310	ps

(1) HRPWM 以 60MHz 的最小 SYSCLKOUT 频率工作。

(2) 最大 MEP 步长基于最差情况、最高温度。MEP 步长将随着温度的升高和电压的降低而增大，随着温度的降低和电压的升高而减小。

使用 HRPWM 特性的应用程序应使用 MEP 缩放因子优化器 (SFO) 估算软件函数。SFO 函数有助于在 HRPWM 运行时动态地估算每个 SYSCLKOUT 周期内的 MEP 步数。

(3) 典型值测试温度为（20摄氏度）、3.3V供电电压下测试值。

4.11 增强型捕获模块 (eCAP1)

此器件包含一个增强型捕获模块 (eCAP), 图 4-89 所示为 eCAP 功能框图。

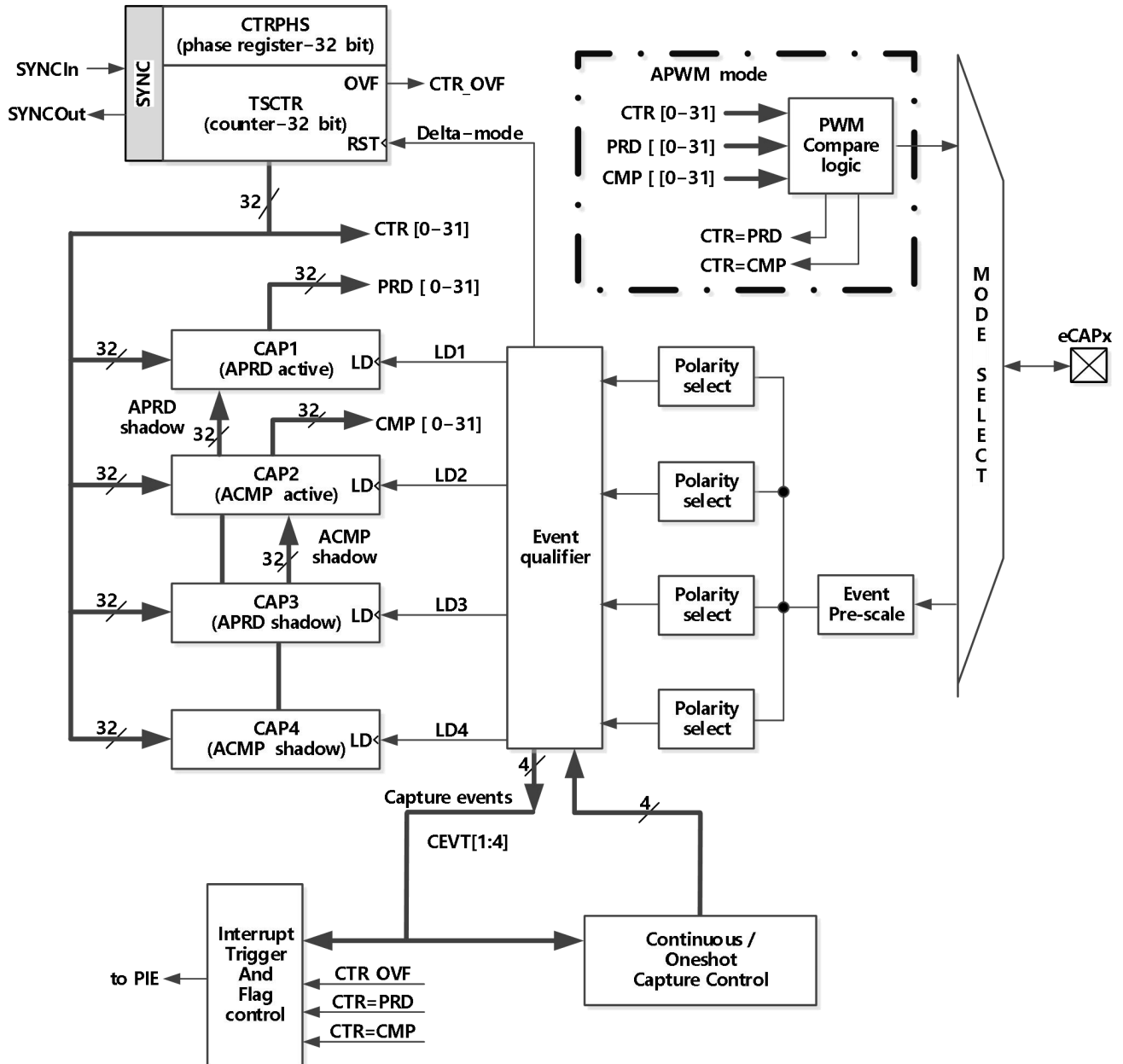


图 4-89 eCAP 功能框图

eCAP 模块基于 SYSCLKOUT 时钟。

PCLKCR1 寄存器中的时钟使能位 (ECAP1 ENCLK) 用于单独关闭 eCAP 模块（实现低功耗运行）。

复位时，ECAP1ENCLK 为低电平，表明外设时钟已关闭。

表 4-90 eCAP 控制和状态寄存器

名称	地址	大小 (X 16)	受 EALLOW 保护	说明
TSCTR	0x6A00	2		时间戳计数器
CTRPHS	0x6A02	2		计数器相位偏移值寄存器
CAP1	0x6A04	2		捕获 1 寄存器
CAP2	0x6A06	2		捕获 2 寄存器
CAP3	0x6A08	2		捕获 3 寄存器
CAP4	0x6A0A	2		捕获 4 寄存器
保留	0x6A0C-0x6A12	8		保留
ECCTL1	0x6A14	1		捕获控制寄存器 1
ECCTL2	0x6A15	1		捕获控制寄存器 2
ECEINT	0x6A16	1		捕获中断使能寄存器
ECFLG	0x6A17	1		捕获中断标志寄存器
ECCLR	0x6A18	1		捕获中断清除寄存器
ECFRC	0x6A19	1		捕获中断强制寄存器
保留	0x6A1A-0x6A1F	6		保留

4.11.1 增强型捕获 (eCAP) 时序

表4-91所示为eCAP时序要求，而表4-92所示为eCAP开关特性。

表 4-91 增强型捕获 (eCAP) 时序要求

参数		测试条件	最小值	最大值	单位
t _w (CAP)	捕获输入脉冲宽度	异步	2t _c (SCO)		周期
		同步	2t _c (SCO)		周期
		带有输入限定器	1t _c (SCO)+t _w (IQSW)		周期

表 4-92 eCAP 开关特性

参数		测试条件	最小值	最大值	单位
t _w (APWM)	脉冲持续时间，APWMx 输出高电平/低电平		20		ns

4.12 高性能捕获 (HRCAP) 模块

高性能捕获 (HRCAP) 模块以高分辨率（典型 300ps）测量外部脉冲之间的差异。

HRCAP 模块的特性包括：

- 可以在非高分辨率或者高分辨率模式下捕获脉冲宽度
- 差分 (Delta) 模式脉宽捕获
- 在每个边沿上 300ps 分辨率的典型高分辨率捕获
- 下降沿或上升沿中断
- 2 深度缓冲器中脉冲宽度的持续模式捕捉
- 用于精确高分辨率捕获的校准逻辑
- 上述所有资源都专用于单个输入引脚

除了高分辨率校准模块外，HRCAP 模块还包括一个捕获通道，在校准时，内部连接到最后一个可用的 ePWMxA HRPWM 通道。

每个 HRCAP 通道都有以下独立的关键资源：

- 专用输入捕获引脚
- 16 位 HRCAP 时钟等于 PLL 输出频率（与 SYSCLK 异步）或者等于 SYSCLK 频率（与 SYSCLK 同步）
- 在 2 级深度的缓冲器中捕获高分辨率脉冲宽度

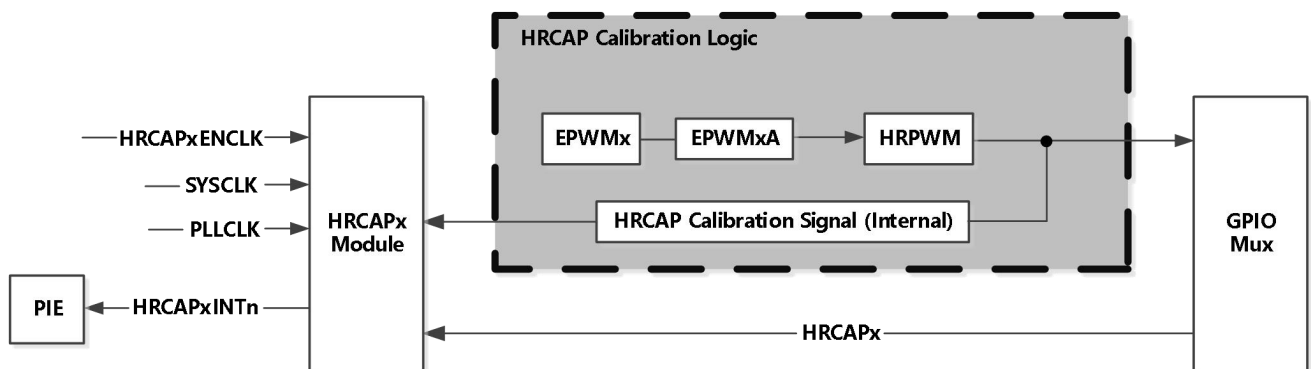


图 4-90 HRCAP 功能框图

表 4-93 HRCAP 寄存器

名称	HRCAP1	HRCAP2	大小 (X 16)	说明
HCCTL	0x6AC0	0x6AE0	1	HRCAP 控制寄存器 ⁽¹⁾
HCIFR	0x6AC1	0x6AE1	1	HRCAP 中断标志寄存器
HCICLR	0x6AC2	0x6AE2	1	HRCAP 中断清除寄存器
HCIFRC	0x6AC3	0x6AE3	1	HRCAP 中断强制寄存器
HCCOUNTER	0x6AC4	0x6AE4	1	HRCAP 16 位计数器寄存器
HCCAPCNTRISE0	0x6AD0	0x6AF0	1	在上升边沿 0 寄存器上的 HRCAP 捕获计数器
HCCAPCNTRFALL0	0x6AD2	0x6AF2	1	在下降边沿 0 寄存器上的 HRCAP 捕获计数器
HCCAPCNTRISE1	0x6AD8	0x6AF8	1	在上升边沿 1 寄存器上的 HRCAP 捕获计数器
HCCAPCNTRFALL1	0x6ADA	0x6AFA	1	在下降边沿 1 寄存器上的 HRCAP 捕获计数器

(1) 寄存器受 EALLOW 保护。

4.12.1 高性能捕获(HRCAP) 时序

表4-94 高性能捕获(HRCAP) 时序要求

	最小值	典型值	最大值	单位
$t_c(\text{HCCAPCLK})$ 周期时间, HRCAP 捕获时钟的时间	8.333		10.204	ns
$t_w(\text{HRCAP})$ 脉冲宽度, HRCAP 捕获的时间	$7t_c(\text{HCCAPCLK})^{(1)}$			ns
HRCAP 步长尺寸 ⁽²⁾		300		ps

(1) 所列出的最小脉宽并未将所有相关 HCCAP 寄存器必须被读取而且必须将脉宽内的 RISE/FALL (上升/下降) 事件标志清除以确保有效捕获数据这一限制考虑在内。

(2) HRCAP 步长将随着电压的降低和温度的升高而增大, 随着电压的升高和温度的降低而减小。在高分辨率模式中使用 HRCAP 的应用应该使用 HRCAP 的校准功能以针对变化的运行条件进行动态校准。

4.13 JTAG 端口

JTAG 端口减少到 5 个引脚 ($\overline{\text{TRST}}$, TCK, TDI, TMS, TDO), TCK、TDI、TMS 和 TDO 引脚也是 GPIO 引脚。 $\overline{\text{TRST}}$ 信号在图 4-91 中为引脚选择 JTAG 或者 GPIO 运行模式。在仿真/调试期间, 这些引脚的 GPIO 功能并不可用。如果 GPIO38/TCK/XCLKIN 引脚用于提供外部时钟, 则应使用备用时钟源在仿真/调试期间为设备提供时钟, 这是因为在仿真/调试时这个引脚将需要用于 TCK 功能。。

注

JTAG 引脚也可用作 GPIO 引脚。在电路板设计时应谨慎以确保连接到这些引脚的电路不会影响 JTAG 引脚功能的仿真能力。任一连接到这些引脚的电路不应防止仿真器驱动 JTAG 引脚 (或由 JTAG 引脚驱动) 进行成功的调试。

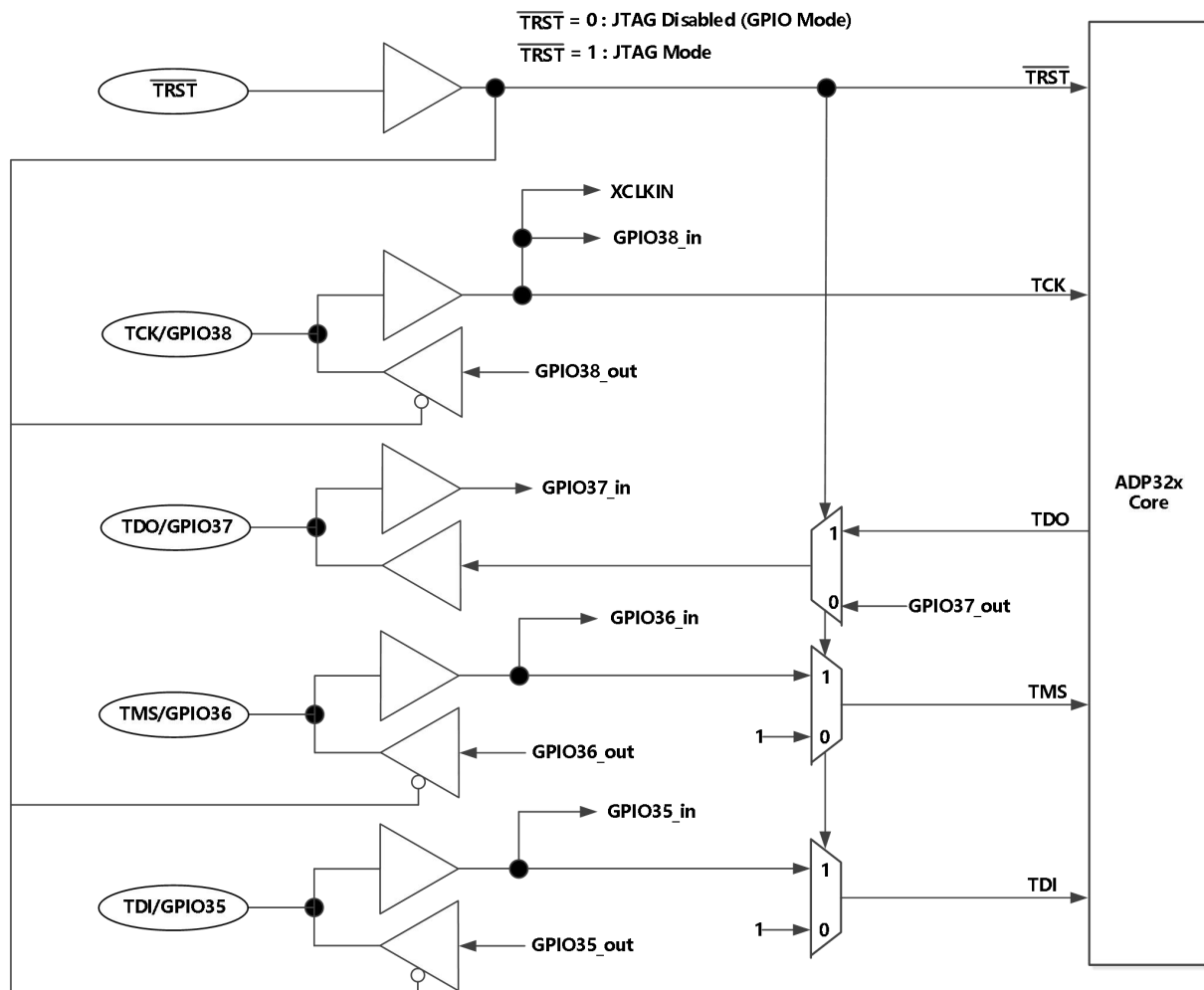


图 4-91 JTAG/GPIO 复用

4.14 GPIO MUX

本产品支持 22 个 GPIO 引脚，GPIO 控制和数据寄存器映射到外设帧 1，可以在寄存器上启用 32 位或者 16 位模式。

表 4-95 所示为 GPIO 寄存器映射。

注意：表格中寄存器包含了预留的寄存器，具体使用需根据本产品对应 GPIO 来使用。

表 4-95 GPIO 寄存器

名称	地址	大小 (x16)	说明
GPIO 控制寄存器 (受 EALLOW 保护)			
GPACTRL	0x6F80	2	GPIO A 控制寄存器 (GPIO0 至 31)
GPAQSEL1	0x6F82	2	GPIO A 限定器选择 1 寄存器 (GPIO0 至 15)
GPAQSEL2	0x6F84	2	GPIO A 限定器选择 2 寄存器 (GPIO16 至 31)
GPAMUX1	0x6F86	2	GPIO A MUX 1 寄存器 (GPIO0 至 15)
GPAMUX2	0x6F88	2	GPIO A MUX 2 寄存器 (GPIO16 至 31)
GPADIR	0x6F8A	2	GPIO A 方向寄存器 (GPIO0 至 31)
GPAPUD	0x6F8C	2	GPIO A 上拉电阻器禁用寄存器 (GPIO0 至 GPIO31)
GPBCTRL	0x6F90	2	GPIO B 控制寄存器 (GPIO32 至 44)
GPBQSEL1	0x6F92	2	GPIO B 限定器选择 1 寄存器 (GPIO32 至 44)
GPBMUX1	0x6F96	2	GPIO B MUX 1 寄存器 (GPIO32 至 44)
GPBDIR	0x6F9A	2	GPIO B 方向寄存器 (GPIO32 至 44)
GPBPUD	0x6F9C	2	GPIO B 上拉电阻器禁用寄存器 (GPIO38 至 44)
AIOMUX1	0x6FB6	2	模拟, I/O 复用 1 寄存器 (AIO0 至 AIO15)
AIODIR	0x6FBA	2	模拟, I/O 方向寄存器 (AIO0 至 AIO15)
GPIO 数据寄存器 (不受 EALLOW 保护)			
GPADAT	0x6FC0	2	GPIO A 数据寄存器 (GPIO0 至 31)
GPASET	0x6FC2	2	GPIO A 数据设定寄存器 (GPIO0 至 31)
GPACLEAR	0x6FC4	2	GPIO A 数据清除寄存器 (GPIO0 至 31)
GPATOGGLE	0x6FC6	2	GPIO A 数据取反寄存器 (GPIO0 至 31)
GPBDAT	0x6FC8	2	GPIO B 数据寄存器 (GPIO32 至 44)
GPBSET	0x6FCA	2	GPIO B 数据设定寄存器 (GPIO32 至 44)
GPBCLEAR	0x6FCC	2	GPIO B 数据清除寄存器 (GPIO32 至 44)
GPBTOGGLE	0x6FCE	2	GPIO B 数据取反寄存器 (GPIO32 至 44)
AIODAT	0x6FD8	2	模拟 I/O 数据寄存器 (AIO0 至 AIO15)
AIOSET	0x6FDA	2	模拟 I/O 数据设定寄存器 (AIO0 至 AIO15)
AIOCLEAR	0x6FDC	2	模拟 I/O 数据清除寄存器 (AIO0 至 AIO15)
AIOTOGGLE	0x6FDE	2	模拟 I/O 数据取反寄存器 (AIO0 至 AIO15)
GPIO 中断和低功耗模式选择寄存器 (受 EALLOW 保护)			
GPIOXINT1SEL	0x6FE0	1	XINT1 GPIO 输入选择寄存器 (GPIO0 至 31)
GPIOXINT2SEL	0x6FE1	1	XINT2 GPIO 输入选择寄存器 (GPIO0 至 GPIO31)
GPIOXINT3SEL	0x6FE2	1	XINT3 GPIO 输入选择寄存器 (GPIO0 至 GPIO31)
GPIO_LPMSEL	0x6FE8	2	LPM GPIO 选择寄存器 (GPIO0 至 GPIO31)

注

从写入 GPxMUXn/AIOMUXn 和 GPxQSELn 寄存器到动作有效有两个 SYSCLKOUT 周期的延迟。

表 4-96 CAN 和 CANFD MUX

702B 寄存器 (受 EALLOW 保护)							
bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
Reserved							
R-0	W/R-0	W/R-0	W/R-0	W/R-0	W/R-0	W/R-0	W/R-0
bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
canfd_gpio_mux				ecan_gpio_mux			
W/R-0	W/R-0	W/R-0	W/R-0	W/R-0	W/R-0	W/R-0	W/R-0

Bit	字段	描述
15-8	Reserved	保留位, 禁止操作
7-4	canfd_gpio_mux	canfd 引脚选择 1010: GPIO28 为 rxd, GPIO 29 为 txd 0101: GPIO 0 为 rxd, GPIO 32 为 txd 其他: 无效
3-0	ecan_gpio_mux	ecan 引脚选择 0101: GPIO 0 为 rxd, GPIO 32 为 txd 其他: 无效

注意:

(1) HRCAP

GPIO19 作为 HRCAP1 输入使用时, 配置 GPAMUX2[21:20]=01。上拉使能为 GPAPUD[19]。

(2) CANFD

CANFD 功能在 GPIO28/29 上优先级最高, 需要配置 702B 寄存器 7~4 位为 1010。上拉使能为 GPAPUD[28]和 GPAPUD[29]。

CANFD 功能在 GPIO0/32 上优先级低于 CAN 功能。需设置 702B 寄存器 7~4 位为 0101, 且 702B 寄存器 3~0 位不能为 0101。上拉使能为 GPAPUD[0]和 GPBPUD[0]。

(3) ECAN

ECAN 功能复用在 GPIO0/32 上优先级最高。需配置 702B 寄存器 3~0 位为 0101。上拉使能为 GPAPUD[0]和 GPBPUD[0]。

表 4-97 GPIOA MUX ⁽¹⁾⁽²⁾

	复位时默认值 主 I/O 功能	外设选择 1	外设选择 2	外设选择 3
GPAMUX1 寄存器位	(GPAMUX1 位 = 00)	(GPAMUX1 位 = 01)	(GPAMUX1 位 = 10)	(GPAMUX1 位 = 11)
1-0	GPIO0	EPWM1A (O)	保留	保留
3-2	GPIO1	EPWM1B (O)	保留	COMP1OUT (O)
5-4	GPIO2	EPWM2A (O)	保留	保留
7-6	GPIO3	EPWM2B (O)	SPISOMIA (I/O)	COMP2OUT (O)
9-8	GPIO4	EPWM3A (O)	保留	保留
11-10	GPIO5	EPWM3B (O)	SPISIMOA (I/O)	ECAP1 (I/O)
13-12	GPIO6	EPWM4A (O)	EPWMSYNCl (I)	EPWMSYNCO (O)
15-14	GPIO7	EPWM4B (O)	SCIRXDA (I)	保留
17-16	GPIO8	EPWM5A (O)	保留	ADCS0CA0 (O)
19-18	GPIO9	EPWM5B (O)	LINTXA (O)	HRCAP1 (I)
21-20	保留	保留	保留	保留
23-22	保留	保留	保留	保留
25-24	保留	保留	保留	保留
27-26	保留	保留	保留	保留
29-28	保留	保留	保留	保留
31-30	保留	保留	保留	保留
GPAMUX2 寄存器位	(GPAMUX2 位 = 00)	(GPAMUX2 位 = 01)	(GPAMUX2 位 = 10)	(GPAMUX2 位 = 11)
1-0	GPIO16	SPISIMOA (I/O)	保留	TZ2 (I)
3-2	GPIO17	SPISOMIA (I/O)	保留	TZ3 (I)
5-4	GPIO18	SPICLKA (I/O)	LINTXA (O)	XCLKOUT (O)
7-6	GPIO19/XCLKIN	SPISTEA (I/O)	LINRXA (I)	ECAP1 (I/O)
9-8	保留	保留	保留	保留
11-10	保留	保留	保留	保留
13-12	保留	保留	保留	保留
15-14	保留	保留	保留	保留
17-16	保留	保留	保留	保留
19-18	保留	保留	保留	保留
21-20	保留	保留	保留	保留
23-22	保留	保留	保留	保留
25-24	GPIO28	SCIRXDA (I)	SDAA (I/OD)	TZ2 (I)
27-26	GPIO29	SCITXDA (O)	SCLA (I/OD)	TZ3 (I)
29-28	保留	CANRXA (I)	保留	保留
31-30	保留	CANTXA (O)	保留	保留

(1) “保留”意味着没有为该 GPxMUX1/2 寄存器设置指定外设被。如果它被选择，那么引脚的状态将为未定义并且可驱动此引脚。这个选择是为将来进行扩展而预留的配置。

(2) I = 输入，O = 输出，OD = 开漏。

表 4-98 GPIOB MUX ⁽¹⁾

	复位时默认 主 I/O 功能	外设选择 1	外设选择 2	外设选择 3
外设选择 4	(GPBMUX1 位 = 00)	(GPBMUX1 位 = 01)	(GPBMUX1 位 = 10)	(GPBMUX1 位 = 11)
1-0	GPIO32	SDAA (I/OD)	EPWMSYNCI (I)	ADCSOAO (O)
3-2	保留	SCLA (I/OD)	EPWMSYNCO (O)	ADCSOCBO (O)
5-4	GPIO34	COMP2OUT (O)	保留	COMP3OUT (O)
7-6	GPIO35 (TDI)	保留	保留	保留
9-8	GPIO36 (TMS)	保留	保留	保留
11-10	GPIO37 (TDO)	保留	保留	保留
13-12	GPIO38/XCLKIN (TCK)	保留	保留	保留
15-14	保留	保留	被留	保留
17-16	保留	保留	被留	保留
19-18	保留	保留	被留	被保留
21-20	保留	保留	被留	保留
23-22	保留	保留	保留	保留
25-24	保留	保留	保留	保留
27-26	保留	保留	保留	保留
29-28	保留	保留	保留	保留
31-30	保留	保留	保留	保留

(1) I = 输入, O = 输出, OD = 开漏。

表 4-99 模拟 MUX ⁽¹⁾

	复位时默认
	外设选择 2 和外设选择 3
AIOMUX1 寄存器位	AIOMUX1 位 = 0,x
1-0	ADCINA0 (I)
3-2	ADCINA1(I)
5-4	AIO2 (I/O)
7-6	ADCINA3(I)
9-8	AIO4 (I/O)
11-10	ADCINA5 ⁽²⁾ (I)
13-12	AIO6 (I/O)
15-14	ADCINA7(I)
17-16	ADCINB0 (I)
19-18	ADCINB1(I)
21-20	AIO10 (I/O)
23-22	ADCINB3(I)
25-24	AIO12 (I/O)
27-26	ADCINB5 ⁽²⁾ (I)
29-28	AIO14 (I/O)
31-30	保留

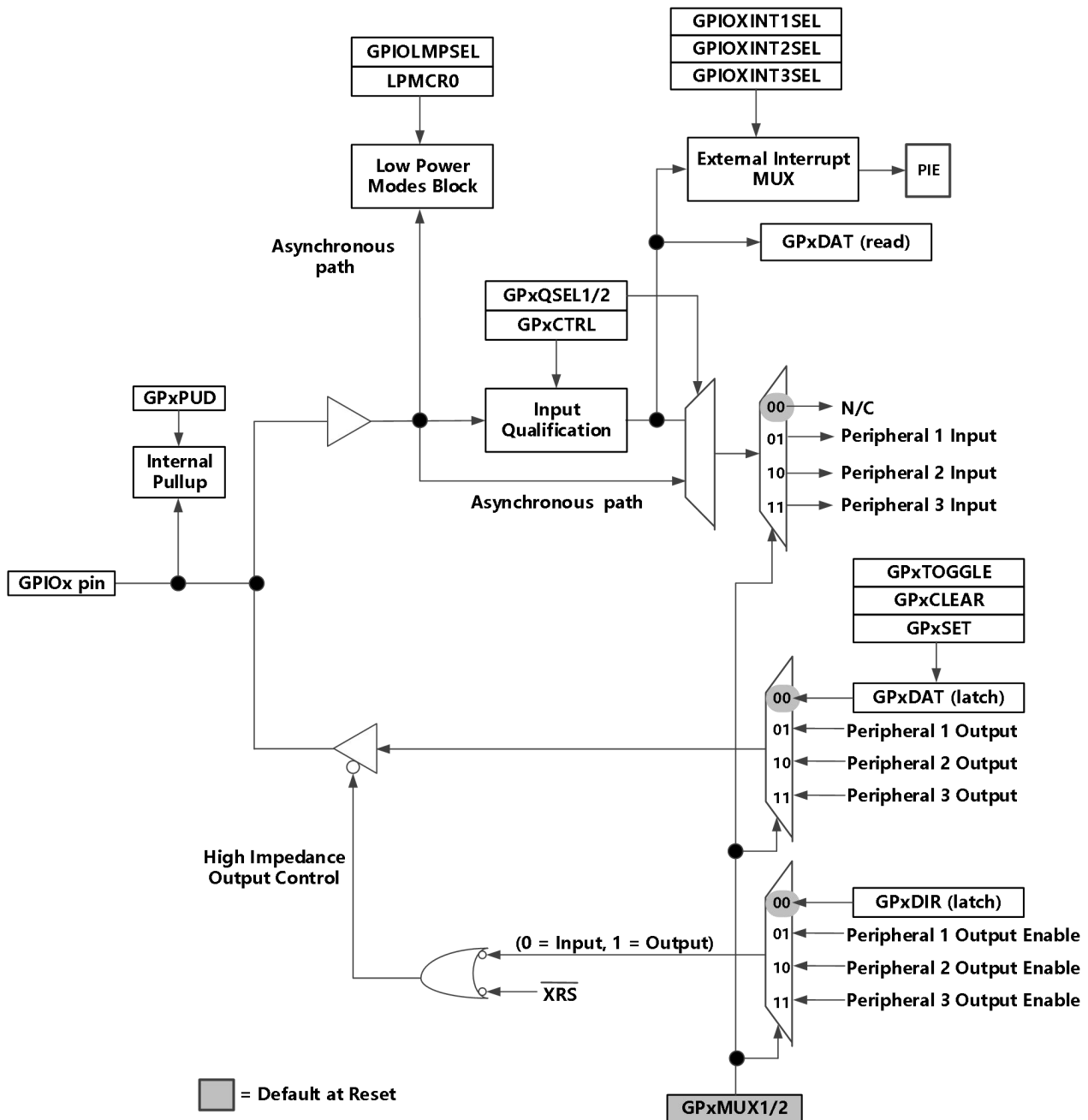
(1) I = 输入, O = 输出。

通过配置 GPxQSEL1/2 寄存器, 用户可为每个 GPIO 引脚选择输入限定的类型:

- 仅同步至 SYSCLKOUT (GPxQSEL1/2=0,0): 这是复位时所有 GPIO 引脚的默认模式, 它只会将输入信号同步至系统时钟 (SYSCLKOUT)。

- 使用采样窗口的限定条件 (GPxQSEL1/2=0, 1 和 1, 0) : 在该模式下, 输入信号在与系统时钟 (SYSCLKOUT)同步后, 通过指定的周期数进行限定, 然后才允许输入发生变化。
- 采样周期由 GPxCTRL 寄存器内的 QUALPRD 位所指定, 对一组 8 个信号进行配置。采样周期指定用于采样输入信号的 SYSCLKOUT 周期的倍数。一个采样窗口可包含 3 次或 6 次采样, 并且只有当所有采样与图 4-92 (对于 6 次采样) 中所显示的一样时 (全 0 或者全 1), 输出才会改变。
- 无同步 (GPxQSEL1/2=1, 1): 此模式用于无需同步的外设 (同步不在外设内执行) 。

由于器件上所要求的多级复用, 有可能会出现一个外设输入信号被映射到多个 GPIO 引脚的情况。此外, 当一个输入信号未被选择时, 输入信号将默认为 0 或 1 状态, 具体取决于外设。



- A. “x” 代表端口 A 或 B。例如，GPxDIR 是指 GPADIR 和 GPBDIR 寄存器，这取决于所选择的特定 GPIO 引脚。
B. GPxDAT 锁定/读取访问存储器同一位置。
C. 这是一个常用的 GPIO MUX 框图。并非所有选项都可用于所有 GPIO 引脚。

图 4-92 GPIO 复用

4.14.1 GPIO 输出时序

表4-100 通用输出开关特性

在推荐的运行条件下（除非另有说明）

参数		最小值	最大值	单位
$t_{r(GPO)}$ GPIO 从低电平切换至高电平的上升时间	所有 GPIO		13 ⁽¹⁾	ns
$t_{f(GPO)}$ GPIO 从高电平切换至低电平的下降时间	所有 GPIO		13 ⁽¹⁾	ns
$t_{f(GPO)}$ 切换频率			15	MHz

(1) 上升时间和下降时间随着 I/O 引脚上的电气负载而变化。给出的值适用于 I/O 引脚上的 40pF 负载。

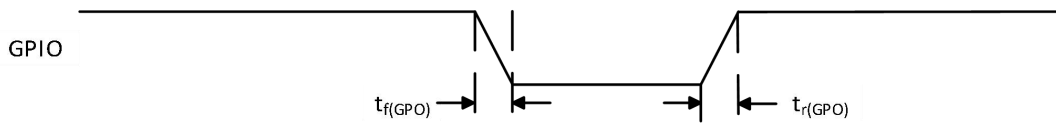


图 4-93 通用输出时序

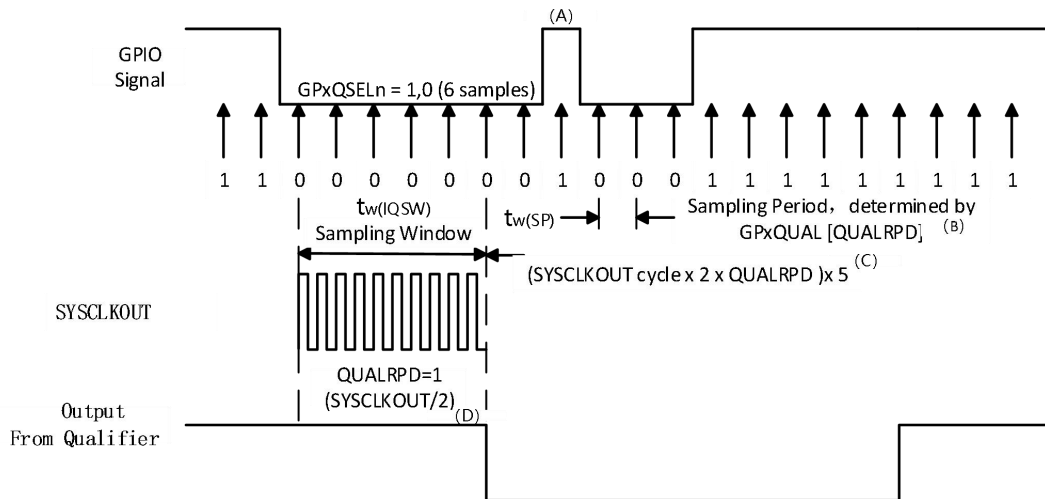
4.14.2 GPIO 输入时序

表 4-101 通用输入时序要求

		最小值	最大值	单位
$t_w(SP)$ 采样周期	QUALPRD=0	$1t_c(SCO)$		周期
	QUALPRD≠0	$2t_c(SCO)*QUALPRD$		周期
$t_w(IQSW)$ 输入限定器采样窗口		$t_w(SP)*(n^{(1)}-1)$		周期
$t_w(GPI)^{(2)}$ GPIO 低电平/高电平脉冲持续时间	同步模式	$2t_c(SCO)$		周期
	有输入限定器	$t_w(IQSW) + t_w(SP) + 1t_c(SCO)$		周期

(1) "n" 代表由 GPxQSELn 寄存器定义的限定采样数量。

(2) 对于 $t_w(GPI)$ ，低电平有效信号的脉宽在 V_{IL} 至 V_{IL} 之间进行测量，而对于高电平有效信号脉宽在 V_{IH} 至 V_{IH} 之间进行测量。



- 输入限定器将忽略该毛刺脉冲。QUALPRD 位字段指定了限定采样周期，值可在 0x00 至 0xFF 间变化。如果 QUALPRD=00，则采样周期为 1 个 SYSCLKOUT 周期。对于任何其它的 "n" 值，限定采样周期为 2n SYSCLKOUT 周期（即每 2n SYSCLKOUT 周期对 GPIO 引脚进行采样）。
- 通过 GPxCTRL 寄存器选择的限定周期应用于一组 8 个 GPIO 引脚上。
- 此限定块可取 3 个或 6 个采样。由 GPxQSELn 寄存器选择使用的采样模式。
- 在所示的示例中，为了使限定器检测到变化，输入应该在 10 个 SYSCLKOUT 周期或者更长的时间内保持稳定。换言之，输入应该在 $(5 \times QUALPRD \times 2)$ SYSCLKOUT 周期内保持稳定。这将确保出现 5 个采样周期用于检测。由于外部时钟是异步驱动的，13 个 SYSCLKOUT 宽的脉冲可确保可靠识别。

图4-94 采样模式

4.14.3 输入信号的采样窗口宽度

以下内容总结了：各种输入限定器配置下，输入信号的采样窗口宽度。

采样频率是指相对于SYSCLKOUT的信号采样频率。

如果 $QUALPRD \neq 0$ ，采样频率 = $SYSCLKOUT / (2 * QUALPRD)$

如果 $QUALPRD = 0$ ，采样频率 = $SYSCLKOUT$

如果 $QUALPRD \neq 0$ ，采样周期 = $SYSCLKOUT$ 周期 $\times 2 \times QUALPRD$

在上面的等式中， $SYSCLKOUT$ 周期是指 $SYSCLKOUT$ 的周期。

如果 $QUALPRD = 0$ ，采样周期 = $SYSCLKOUT$ 周期

在一个指定的采样窗口中，取输入信号的 3 个或者 6 个采样以确定信号的有效性，这个值由写入到 $GPxQSELn$ 寄存器的值确定。

例 1:

使用 3 个采样的限定

如果 $QUALPRD \neq 0$ ，采样窗口宽度 = $(SYSCLKOUT \text{ 周期} \times 2 \times QUALPRD) \times 2$

如果 $QUALPRD = 0$ ，采样窗口宽度 = $(SYSCLKOUT \text{ 周期}) \times 2$

情况2:

使用 6 个采样的限定

如果 $QUALPRD \neq 0$ ，采样窗口宽度 = $(SYSCLKOUT \text{ 周期} \times 2 \times QUALPRD) \times 5$

如果 $QUALPRD = 0$ ，采样窗口宽度 = $(SYSCLKOUT \text{ 周期}) \times 5$

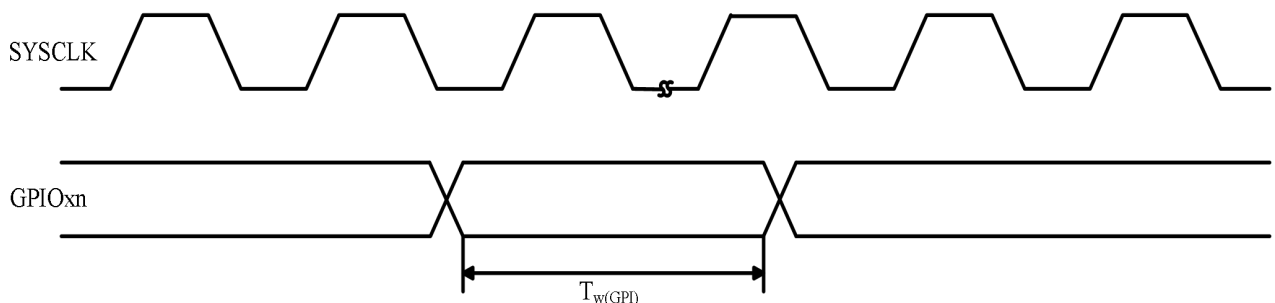


图4-95 通用输入时序

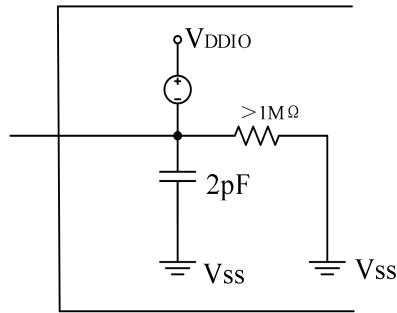


图4-96 带有内部上拉的 GPIO 引脚的输入电阻模型

4.14.4 低功耗唤醒时序

表 4-102 所示为时序要求，表 4-103 所示为开关特性，图 4-97 所示为 IDLE 模式下的时序图。

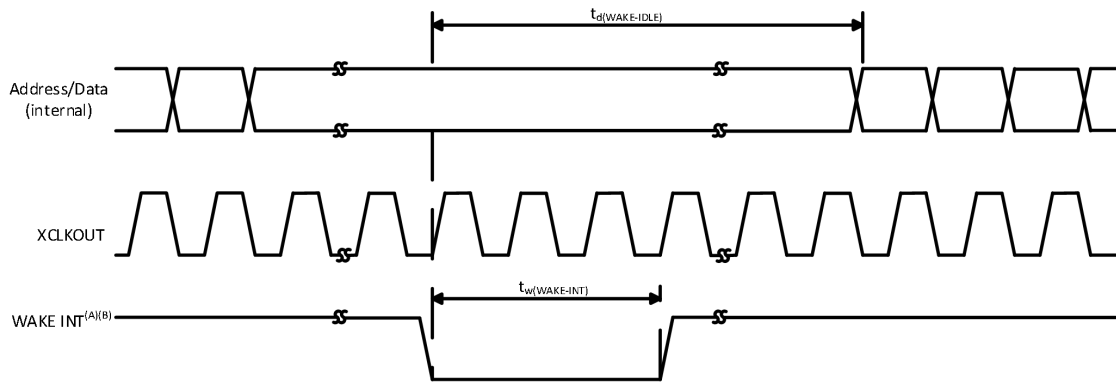
表 4-102 IDLE 模式时序要求

		最小值	典型值	最大值	单位
$t_{w(WAKE-INT)}$ 脉冲持续时间，外部唤醒信号的时间	无输入限定器	$2t_{c(SCO)}$			周期
	有输入限定器	$5t_{c(SCO)} + t_{w(IQSW)}$			

表 4-103 IDLE 模式开关特性

参数	测试条件	最小值	典型值	最大值	单位
$t_d(WAKE-IDLE)$	延迟时间，外部唤醒信号到程序执行重新开始的时间 ⁽²⁾				周期
	• 从 Flash 唤醒 – Flash 处于激活状态	无输入限定器		$20t_{c(SCO)}$	周期
		有输入限定器		$20t_{c(SCO)} + t_{w(IQSW)}$	
	• 从 Flash 唤醒 – Flash 处于休眠状态	无输入限定器		$1050t_{c(SCO)}$	周期
		有输入限定器		$1050t_{c(SCO)} + t_{w(IQSW)}$	
	• 从 SARAM 中唤醒	无输入限定器		$20t_{c(SCO)}$	周期
		有输入限定器		$20t_{c(SCO)} + t_{w(IQSW)}$	

(1) 该时间是在 IDLE 指令之后立即开始指令执行的时间。一个 ISR（由唤醒触发）信号的执行会涉及额外的延迟。



- A. WAKE INT可以是任一启用的中断， $\overline{\text{WDINT}}$ 或者 $\overline{\text{XRS}}$ 。执行IDLE 指令后，在唤醒信号生效前需要 5 个 OSCCLK 周期（最小值）的延迟。
- B. 从执行将器件置于低功耗模式 (LPM) 的 IDLE 指令开始，在至少 4 个 OSCCLK 周期之前，不应启动唤醒。

图4-97 进入和退出IDLE模式的时序

表 4-104 STANDBY 模式时序要求

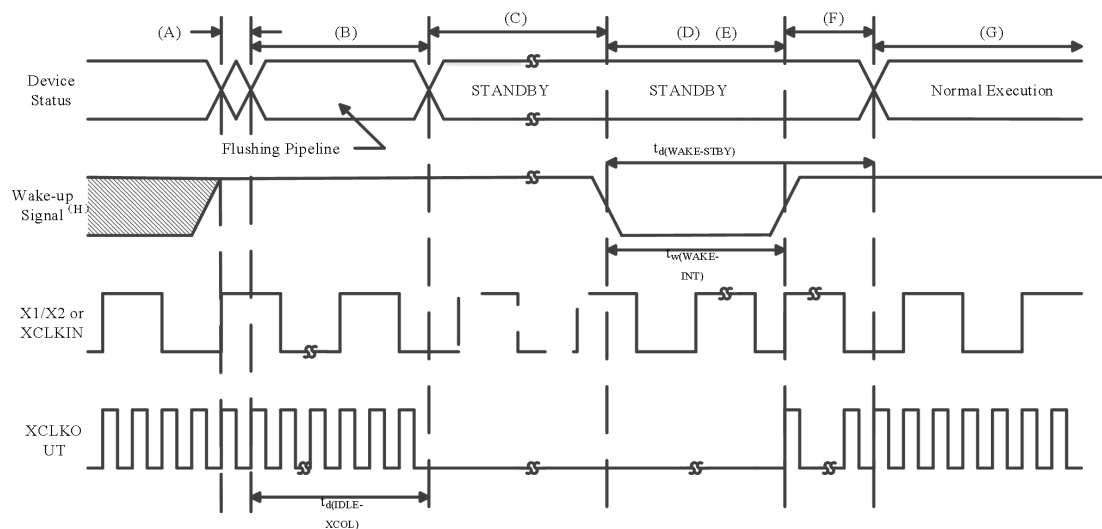
		最小值	典型值	最大值	单位
$t_w(\text{WAKE-INT})$ 脉冲持续时间，外部唤醒信号的时间	无输入限定器	$3t_c(\text{OSCCLK})$			周期
	有输入限定器 ⁽¹⁾	$(2 + \text{QUALSTDBY}) * t_c(\text{OSCCLK})$			

(1) QUALSTDBY 是 LPMCR0 寄存器内的 6 位字段。

表 4-105 STANDBY 模式开关特性

参数	测试条件	最小值	典型值	最大值	单位
$t_d(\text{IDLE-XCOL})$	延迟时间，IDLE 指令执行至 XCLKOUT 为低电平的时间	$32t_c(\text{SCO})$		$45t_c(\text{SCO})$	周期
$t_d(\text{WAKE-STBY})$	延迟时间，外部唤醒信号到程序执行重新开始的时间 ⁽¹⁾				周期
	• 从闪存唤醒 – 激活状态中的闪存模块	无输入限定器		$100t_c(\text{SCO})$	周期
		有输入限定器		$100t_c(\text{SCO}) + t_w(\text{WAKE-INT})$	
	• 从闪存唤醒 – 睡眠状态中的闪存模块	无输入限定器		$1125t_c(\text{SCO})$	周期
		有输入限定器		$1125t_c(\text{SCO}) + t_w(\text{WAKE-INT})$	
	• 从 SARAM 中唤醒	无输入限定器		$100t_c(\text{SCO})$	周期
		有输入限定器		$100t_c(\text{SCO}) + t_w(\text{WAKE-INT})$	

(1) 这个时间是在 IDLE 指令之后立即开始指令执行的时间。一个 ISR（由唤醒触发）信号的执行会涉及额外的延迟。



A. 器件执行IDLE 指令进入STANDBY模式。

B. PLL响应STANDBY 信号。SYSCLKOUT在关闭之前保持输出如下所示的周期数：

- 当DIVSEL=00 或11 时，16 个周期
- 当DIVSEL=10 时，32 个周期
- 当DIVSEL=11 时，64 个周期

此延迟使得CPU 流水线和它等待的操作被适当清空。

C. 到外设的时钟被关闭。然而，PLL 和看门狗并未关闭。此器件现在处于STANDBY 模式。IDLE 指令被执行后，在唤醒信号生效前需要 5 个 OSCCLK 周期（最小值）的延迟。

D.外部唤醒信号被驱动为有效。

E. 提供给 GPIO 引脚的唤醒信号必须满足最小脉冲宽度要求。此外，该信号必须无毛刺脉冲。如果噪声信号被送到 GPIO 引脚，器件的唤醒状态将不确定，且在随后的唤醒脉冲中，也可能不会退出低功耗模式。

F. 在一个延迟周期内，退出STANDBY 模式。

G. 恢复正常执行。此器件将响应中断（如果被启用）。

H. 从将 的 执行 IDLE 指令，器件进入低功耗模式 (LPM)后，至少 4 个 OSCCLK 周期内，不应启动唤醒。

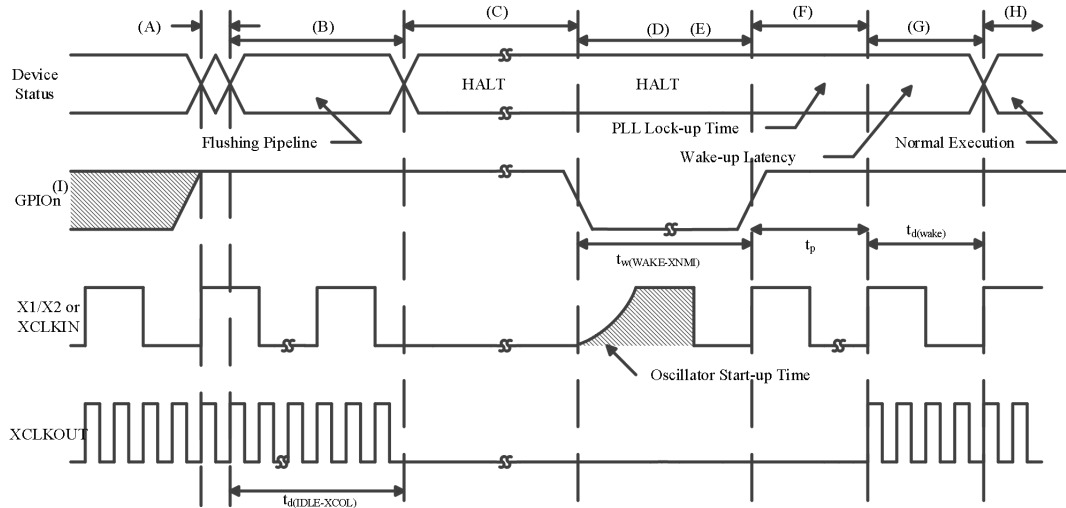
图 4-98 进入和退出 STANDBY 模式的时序

表 4-106 HALT 模式时序要求

	最小值	典型值	最大值	单位
$t_{w(WAKE- GPIO)}$ GPIO 唤醒脉冲持续时间	$t_{oscst}+2t_c(OSCCLK)$			周期
$t_{w(WAKE- XRS)}$ $\overline{XRS}$ 唤醒脉冲持续时间	$t_{oscst}+8t_c(OSCCLK)$			周期

表 4-107 HALT 模式开关特性

参数		最小值	典型值	最大值	单位
$t_d(\text{IDLE-XCOL})$	延迟时间，执行 IDLE 指令至 XCLKOUT 为低电平	$32t_{c(\text{SCO})}$		$45t_{c(\text{SCO})}$	周期
t_p	PLL 锁定时间			1	ms
$t_d(\text{WAKE-HALT})$	延迟时间，PLL 锁存到程序恢复执行 • 从闪存唤醒 – 处于睡眠状态的闪存模块			$1125t_{c(\text{SCO})}$	周期
	• 从 SARAM 中唤醒			$35t_{c(\text{SCO})}$	周期



A. 执行 IDLE 指令进入 HALT 模式。

B. PLL 响应 HALT 信号。在关闭振荡器且停止内核输入时钟 CLKIN 之前，SYSCLKOUT 保持输出如下所示的周期数：

- 当 DIVSEL=00 或 11 时，16 周期
- 当 DIVSEL=10 时，32 个周期
- 当 DIVSEL=11 时，64 个周期

该延迟使得 CPU 流水线和它等待的操作被适当清空。

C. 关闭外设时钟和 PLL。如果使用石英晶体或者陶瓷谐振器作为时钟源，内部振荡器也会关闭。器件现处于 HALT 模式，绝对是最小功耗。可在 HALT 模式下保持零引脚内部振荡器 (INTOSC1 和 INTOSC2) 以及看门狗处于活动状态。可通过对 CLKCTL 寄存器中的适当位进行写入操作来实现此功能。执行 IDLE 指令后，在唤醒信号生效前需要 5 个 OSCCLK 周期 (最小值) 的延迟。

D. 当 GPIO_n 引脚 (用于使器件退出 HALT 模式) 驱动为低电平时，振荡器打开且振荡器唤醒序列启动。只有当振荡器稳定后，GPIO 引脚才应被驱动为高电平。这使得可在 PLL 锁存期间提供一个干净的时钟信号。由于 GPIO 引脚的下降沿异步开始唤醒序列，请注意在进入和处于 HALT 模式期间保持低噪声环境。

E. 提供给 GPIO 引脚的唤醒信号必须满足最小脉冲宽度的要求。此外，该信号必须无毛刺脉冲。如果噪声信号被送到 GPIO 引脚，器件的唤醒行为将不确定，并且在随后的唤醒脉冲中，也可能不会退出低功耗模式。

F. 一旦振荡器稳定，PLL 锁定序列启动耗时 1ms。

G. 当输入内核的 CLKIN 启用时，设备将在延迟后响应中断 (如果启用)，现在退出 HALT 模式。

H. 恢复正常运行。

I. 执行 IDLE 指令，器件进入低功耗模式 (LPM) 后，至少 4 个 OSCCLK 周期内，不应启动唤醒。

图 4-99 使用 GPIO_n 唤醒 HALT 模式

5 电气规范

5.1 最大绝对额定值⁽¹⁾⁽²⁾

电源电压范围 V_{DDIO} , (I/O)	-0.3V 至 5.5V
电源电压范围(V_{DD})	-0.3 V 至 1.32V
模拟电压范围, V_{DDA}	-0.3V 至 3.63V
输入电压范围, $V_{IN}(5V)$	-0.3V 至 5.5V
输出电压范围, V_O	-0.3V 至 4.6V
输入钳位电流, $I_{IK}(V_{IN}<0$ 或者 $V_{IN}>V_{DDIO})^{(3)}$	$\pm 20mA$
输出钳位电流, $I_{OK}(V_O<0$ 或者 $V_O>V_{DDIO})$	$\pm 20mA$
结温范围, $T_J^{(4)}$	Q: -40°C 至 150°C Q0: -40°C 至 175°C A: -40°C 至 125°C S: -40°C 至 150°C

- (1) 除非另外说明, 在超过“绝对最大额定值”列出的条件下运行有可能会对器件造成永久损坏。这些仅为在应力额定值下的工作情况, 对于额定值下的器件的功能性操作以及在超出推荐的运行条件下的任何其它操作, 在此并未说明。长时间运行在最大绝对额定条件下会影响设备的可靠性。所有电压值都是以 V_{SS} 为参考。
- (2) 所有电压值都是相对于 V_{SS} 的值, 除非额外注明。
- (3) 每个引脚上的持续钳位电流为 $\pm 2mA$ 。
- (4) 长期高温存储或在最大温度条件下长时间使用, 可能会使器件总体使用寿命缩短。

5.2 建议的运行条件

		最小值	典型值	最大值	单位
器件电源电压, I/O, $V_{DDIO}^{(1)}$		3.2	3.3 或 5	5.5	V
器件电源电压, CPU, V_{DD} (内部 LDO 供电)			1.23		V
电源接地, V_{SS}			0		V
模拟电源电压, $V_{DDA}^{(1)}$ (内部 LDO 供电)			3		V
模拟接地, V_{SSA}			0		V
$f_{SYSCLKOUT}$ 器件时钟频率 (系统时钟)				100	MHz
高电平输入电压, V_{IH}		$0.6 * V_{DDIO}$		$V_{DDIO} + 0.3$	V
低电平输入电压, V_{IL}		$V_{SS} - 0.3$		0.8	V
高电平输出拉电流, $V_{OH} = V_{OH(最小值)}$, I_{OH}	所有 GPIO/AIO 引脚			-4	mA
	组 2 ⁽²⁾			-8	
低电平输出灌电流 $V_{OL} = V_{OL(最大值)}$, I_{OL}	所有 GPIO/AIO 引脚			4	mA
	组 2 ⁽²⁾			8	
结温, T_J	Q 版本	-40		150	°C
	Q0 版本	-40		175	
	A 版本	-40		125	
	S 版本	-40		150	
环境温度, T_A	Q 版本	-40		125	°C
	Q0 版本	-40		150	
	A 版本	-40		85	
	S 版本	-40		125	

(1) V_{DDIO} 和 V_{DDA} 之间的差距应保持在大约 0.2 V 以上, V_{DDA} 是由 V_{DDIO} 供电的 LDO 产生的。

(2) 第 2 组引脚包括: GPIO16、GPIO17、GPIO18、GPIO19、GPIO28、GPIO29、GPIO36、GPIO37。

5.3 电气特性⁽¹⁾

在建议的运行条件下（除非额外注明）

参数		测试条件		最小值	典型值	最大值	单位
V _{OH} 高电平输出电压		I _{OH} =I _{OH} 最大值		V _{DDIO} - 0.9			V
		I _{OH} =50μA		V _{DDIO} -0.2			
V _{OL} 低电平输出电压		I _{OL} =I _{OL} 最大值				0.4	V
I _{IL} 输入电流 (低电平)	带有上拉电阻器	V _{DDIO} =3.3V, V _{IN} =0V	所有 GPIO		-900		μA
			XRS 引脚		-1800		
	带有下拉电阻器	V _{DDIO} =3.3V, V _{IN} =0V				±2	
I _{IH} 输入电流 (高电平)	上拉电阻器被启用	V _{DDIO} =3.3V, V _{IN} = V _{DDIO}				±2	μA
	下拉电阻器被启用	V _{DDIO} =3.3V, V _{IN} = V _{DDIO}			900		
I _{OZ} 输出电流, 高阻抗状态 (关闭状态)		V _O =V _{DDIO} 或者 0V				±2	μA
C _i 输入电容					2		pF
V _{DDIOBOR} 触发点		下降的 V _{DDIO}			2.65		V
监视器复位延迟时间		延迟时间过后, BOR/POR 事件被移除以释放 XRS (V _{DDIO} =3.3V)		280		850	μs
		延迟时间过后, BOR/POR 事件被移除以释放 XRS (V _{DDIO} =5V)		160		600	
LDO V _{DD} 输出		内部 LDO 打开			1.23		V

- (1) 当片载 LDO 被使用时, 它的输出由 POR 电路监控。
(2) 片载 LDO 输出内核电压受 POR/BOR 电路监控, 如果内核电压超出范围, 将复位器件。

5.4 流耗

表 5-1 ADP32F036 在 100MHz SYSCLKOUT 运行条件下的流耗

模式	测试条件	LDO 被启用				LDO 被禁用 (此功能被禁止)					
		IDDIO ⁽¹⁾		IDDA ⁽²⁾		IDD		IDDIO ⁽¹⁾		IDDA ⁽²⁾	
		典型值 ⁽³⁾	最大值	典型值 ⁽³⁾	最大值	典型值 ⁽³⁾	最大值	典型值 ⁽³⁾	最大值	典型值 ⁽³⁾	最大值
可用 (闪存)	下列的外设时钟被启用： • epwm1/2/3/4/5 • ecap1 • ecan • lin • cla • hrpwm • sci-a • spi-a/b • adc • iic • comp 1/2/3 • cpu-timer0/1/2 • hrcap1/2 • canfd • opa • pgal1/2/3 • t-senser 所有 PWM 引脚被切换至 150kHz。所有 I/O 引脚保持未连接状态。 ⁽⁴⁾⁽⁵⁾ 正在闪存之外运行的代码具有 2 个等待状态。 XCLKOUT 被关闭。	50mA ⁽⁶⁾	/	14mA	/						
IDLE	• 闪存休眠。 • XCLKOUT 被关闭。 • 所有外设时钟被关闭。	13.5mA	/	950μA	/						
STANDBY	• 闪存休眠。 • 外设时钟被关闭。	10.1mA	/	950μA	/						
HALT	• 闪存休眠。 • 外设时钟关闭。 • 输入时钟禁用。⁽⁷⁾	8.4mA		950μA	/						

(1) IDDIO 电流取决于 I/O 引脚上的电气负载。

(2) 为了实现所显示的用于 IDLE, STANDBY, 和 HALT 的 IDDA 电流, 必须通过写入 PCLKCR0 寄存器来明确关闭到 ADC 模块的时钟。

(3) TYP 数适用于常温和典型电压。

(4) 下面的操作在环路内完成：

- 数据持续地从 SPI-A/B, SCI-A, eCAN, LIN, 和 I2C 端口中发出。
- 硬件复用器被使用。
- 看门狗被复位。
- ADC 正在执行持续转换。
- COMP1/2 是持续开关电压。
- GPIO17 接通。

(5) CLA 正在持续执行多项式计算。

(6) 对于没有 CLA 的 ADP32F036 器件, 从表 5-1 中用于运行模式的 IDD (LDO 被禁用) / IDDIO (LDO 被启用) 电流数中减去用于 CLA 的 IDD 电流数 (请参阅表 5-2)。

(7) 如果石英晶振或者陶瓷谐振器作为时钟源, HALT 模式将关闭片载晶体振荡器。

注

外设 - 器件中执行的 I/O 复用可防止同时使用所有可用外设。这是因为多个外设功能可能共用一个 I/O 引脚。然而，可同时打开到所有外设的时钟，虽然这一配置并无实际用途。如果这一操作完成，器件消耗的电流将大于流耗表中的额定值。

5.4.1 减少流耗

ADP32F036包含一个减少器件流耗的方法。通过关闭未在指定应用中使用的外设模块的时钟来降低流耗。此外，可利用这三个低功耗模式的任意一个来进一步降低流耗。表5-2表明了通过关闭时钟所实现的流耗降低的典型值。

表 5-2 不同外设的典型流耗 (100MHz 时) ⁽¹⁾

外设模块 ⁽²⁾	I _{DD} 电流减少(mA)
ADC	0.8 ⁽³⁾
I2C	0.99
ePWM	1.40
eCAP	1.53
SCI	1.86
SPI	1.88
COMP/DAC	1.95
HRPWM	1.98
HRCAP	2.15
CPU - 定时器	2.20
内部零引脚振荡器	2.45
CAN	2.98
LIN	3.30
CLA	3.75
CANFD	3.00
OPA	0.002
PGA	0.002
T-SENSOR	0.002

(1) 复位时，所有外设时钟均禁用(除了 CPU 定时器时钟)。只有在外设时钟打开后，才可对外设寄存器进行写入/读取操作。

(2) 对于有多个重复单元的外设模块，此处数据仅表示关闭其中1个单元的流耗降低大小。例如，比如，表中ePWM的流耗降低1.4mA，仅代表关闭一个ePWM外设所降低的流耗大小。

(3) 此数据仅代表取自ADC模块数字部分的流耗降低，关闭ADC的时钟同时也会降低模拟供电电源I_{DDA}流耗。

注

当 XCLKOUT 关闭时，I_{DDIO} 流耗减少了 15mA（典型值）。

注

基线 I_{DD} 电流（在无外设使能情况下，内核执行空操作循环）典型值为 40mA。对于特定应用下 I_{DD} 的基线电流，可在此基线上增加打开对应外设所产生的流耗即可。

进一步降低流耗的方法：

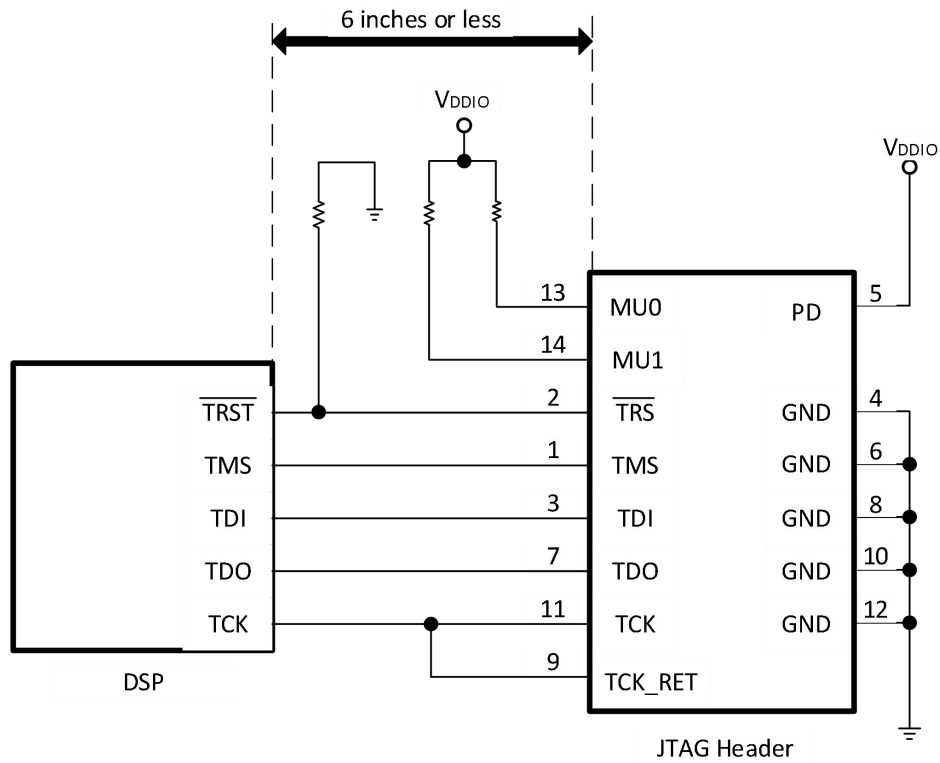
- 如果在 SARAM 上运行代码，闪存模块的电源可关闭。这将使 V_{DD} 电源轨的流耗降低 18mA（典型值）， V_{DDIO} 电源轨的流耗降低 13mA（典型值）。
- 通过禁用负责输出功能的引脚上的上拉电阻，可降低 V_{DDIO} 电源流耗。

5.5 散热设计考虑

根据最终应用设计和运行情况， I_{DD} 和 I_{DDIO} 电流会不同。最终应用中超过建议最大功率耗散的系统可能需要额外的散热增强。环境温度 (T_A) 随着最终应用和产品设计而变化。影响可靠性和功能性的关键参数是 T_J ，结温，而非环境温度。因此，应该注意将 T_J 保持在额定限值内。应该测量 $T_{\text{外壳温度}}$ 以估算工作结温 T_J 。

5.6 无信号缓冲的仿真器连接

图5-4显示了DSP 和JTAG 接头之间针对单处理器配置的连接。如果JTAG接头和DSP之间的距离大于6英寸，那么仿真信号必须加缓冲器。如果距离小于6英寸，通常无需加缓冲器。图5-4显示了较简单、无缓冲的情况。



A. JTAG/GPIO 复用请参阅图 4-17。

图 5-4 无信号缓冲情况下 MCU 与 JTAG 仿真器连接

注

ADP32F036 器件无 EMU0/EMU1 引脚。对于应用方案上 JTAG 接口的设计，接口上的 EMU0/EMU1 引脚必须通过一个 4.7kΩ（典型值）电阻器连接至 VDDIO。

5.7 时序参数符号

所用的时序参数符号按照 JEDEC 标准 100 创建。为了缩短符号，某些引脚名称和其它相关术语的缩写如下：

小写下标及其含意：字母和符号及其含意：

a	访问时间	H	高
c	周期时间 (周期)	L	低
d	延迟时间	V	有效
f	下降时间	X	未知、改变、或者无关电平
h	保持时间	Z	高阻抗
r	上升时间		
su	建立时间		
t	转换时间		
v	有效时间		
w	脉冲持续时间 (宽度)		

5.7.1 时序参数的通用注释

所有输出信号 (包括 XCLKOUT) 由内部时钟进行同步。

5.7.2 测试负载电路

该测试负载电路用于测量本文档中提供的所有开关特性。

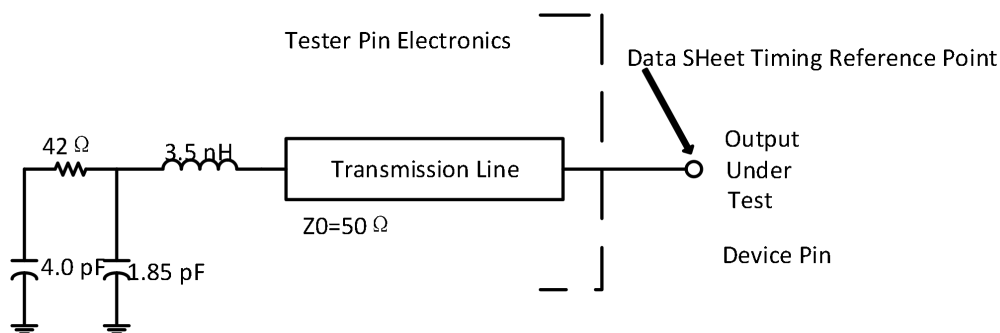


图 5-5 3.3V 测试负载电路

5.7.3 器件时钟表

本小节介绍DSP上可用的不同时钟选项的时序要求和开关特性。表5-3列出了不同时钟的周期时间。

表 5-3 ADP32F036 时钟表和命名规则 (100MHz)

		最小值	典型值	最大值	单位
SYSCLKOUT	$t_{c(SCO)}$, 周期时间	10		500	ns
	频率	2		100	MHz
LSPCLK (1)	$t_{c(LCO)}$, 周期时间	10	66.67 ⁽²⁾		ns
	频率		15 ⁽²⁾	100	MHz
ADC 时钟	$t_{c(ADCCLK)}$, 周期时间	16.67			ns
	频率			60	MHz

(1) 更低的 LSPCLK 将降低器件功耗。

(2) 如果 SYSCLKOUT=60MHz, 则该值为默认复位值。

表 5-4 器件计时要求/特性

		最小值	典型值	最大值	单位
片载振荡器时钟 (X1/X2 引脚) (晶振/谐振器)	$t_{c(OSC)}$, 周期时间	50		200	ns
	频率	5		20	MHz
外部振荡器/时钟源 (XCLKIN 引脚) — PLL 启用	$t_{c(CI)}$, 周期时间 (C8)	33.3		200	ns
	频率	5		30	MHz
外部振荡器/时钟源 (XCLKIN 引脚) — PLL 禁用	$t_{c(CI)}$, (C8) 周期时间	33.3		250	ns
	频率	4		30	MHz
跛行模式 SYSCLKOUT (1/2 启用)	频率范围	1 至 5			MHz
XCLKOUT	$T_{c(XCO)}$, 周期时间(C1)	66.67		2000	ns
	频率	0.5		25	MHz
PLL 锁定时间 ⁽¹⁾	t_p			1	ms

(1) PLLLOCKPRD 寄存器必须按照 OSCCLK 周期的数量进行更新。如果零引脚振荡器 (10MHz) 作为时钟源, 则必须将一个为10000 (最小值) 的值写入 PLLLOCKPRD 寄存器。

5.8 时钟要求和特性

表 5-5 内部零引脚振荡器 (INTOSC1) 特性

参数		最小值	典型值	最大值	单位
30°C 时的内部零引脚振荡器 1 (INTOSC1) ⁽¹⁾	频率		10.000		MHz
步进 (粗调)			132		kHz
步进 (微调)			32		kHz
温漂 ⁽²⁾					kHz/°C

(1) 只有当 LDO 启用时, 才能确保频率范围。

(2) 内部振荡器的输出频率由温度梯度确定。例如:

- 温度的上升将引起输出频率按照温度系数增加。

表 5-6 XCLKIN 时序要求- 启用 PLL

编号		最小值	最大值	单位
C9	$t_{f(CI)}$ 下降时间, XCLKIN		6	ns
C10	$t_{r(CI)}$ 上升时间, XCLKIN		6	ns
C11	$t_{w(CIL)}$ 脉冲持续时间, XCLKIN 低电平脉宽占空比 ($t_{c(OSCCLK)}$)	45	55	%
C12	$t_{w(CIH)}$ 脉冲持续时间, XCLKIN 高电平作为 $t_{c(OSCCLK)}$ 的一部分的时间	45	55	%

表 5-7 XCLKIN 时序要求-- 禁用 PLL

编号		最小值	最大值	单位
C9	$t_{f(CI)}$ 下降时间, XCLKIN	高达 20 MHz	6	ns
		20MHz 至 30MHz	2	
C10	$t_{r(CI)}$ 上升时间, XCLKIN	高达 20 MHz	6	ns
		20MHz 至 30MHz	2	
C11	$t_{w(CIL)}$ 脉冲持续时间, XCLKIN 低电平作为 $t_{c(OSCCLK)}$ 的一部分的时间	45	55	%
C12	$t_{w(CIH)}$ 脉冲持续时间, XCLKIN 高电平作为 $t_{c(OSCCLK)}$ 的一部分的时间	45	55	%

表 5-7 中显示了可能的配置模式。

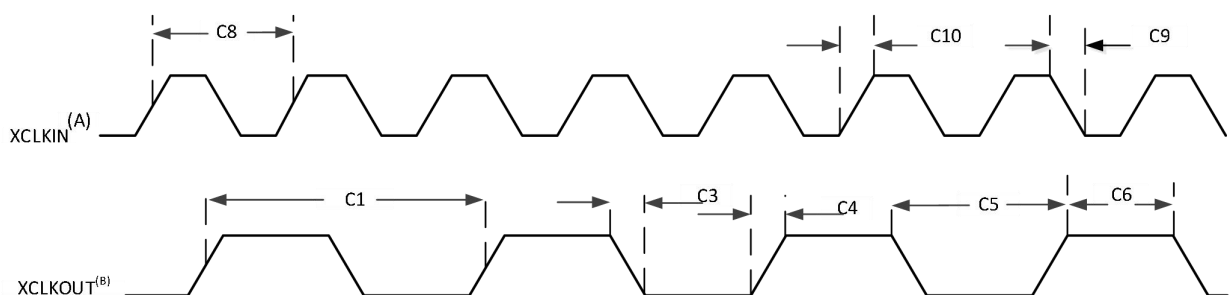
表 5-8 XCLKOUT 开关特性 (PLL 旁路或者被禁用) ^{(1) (2)}

在推荐的运行条件下 (除非额外注明)

编号	参数	最小值	典型值	最大值	单位
C3	$t_{f(XCO)}$ 下降时间, XCLKOUT			5	ns
C4	$t_{r(XCO)}$ 上升时间, XCLKOUT			5	ns
C5	$t_{w(XCOL)}$ 脉冲持续时间, XCLKOUT 低电平的时间	H-2		H+2	ns
C6	$t_{w(XCOH)}$ 脉冲持续时间, XCLKOUT 高电平的时间	H-2		H+2	ns

(1) 假定这些参数有一个40pF 的负载。

(2) $H=0.5t_{c(XCO)}$ 。



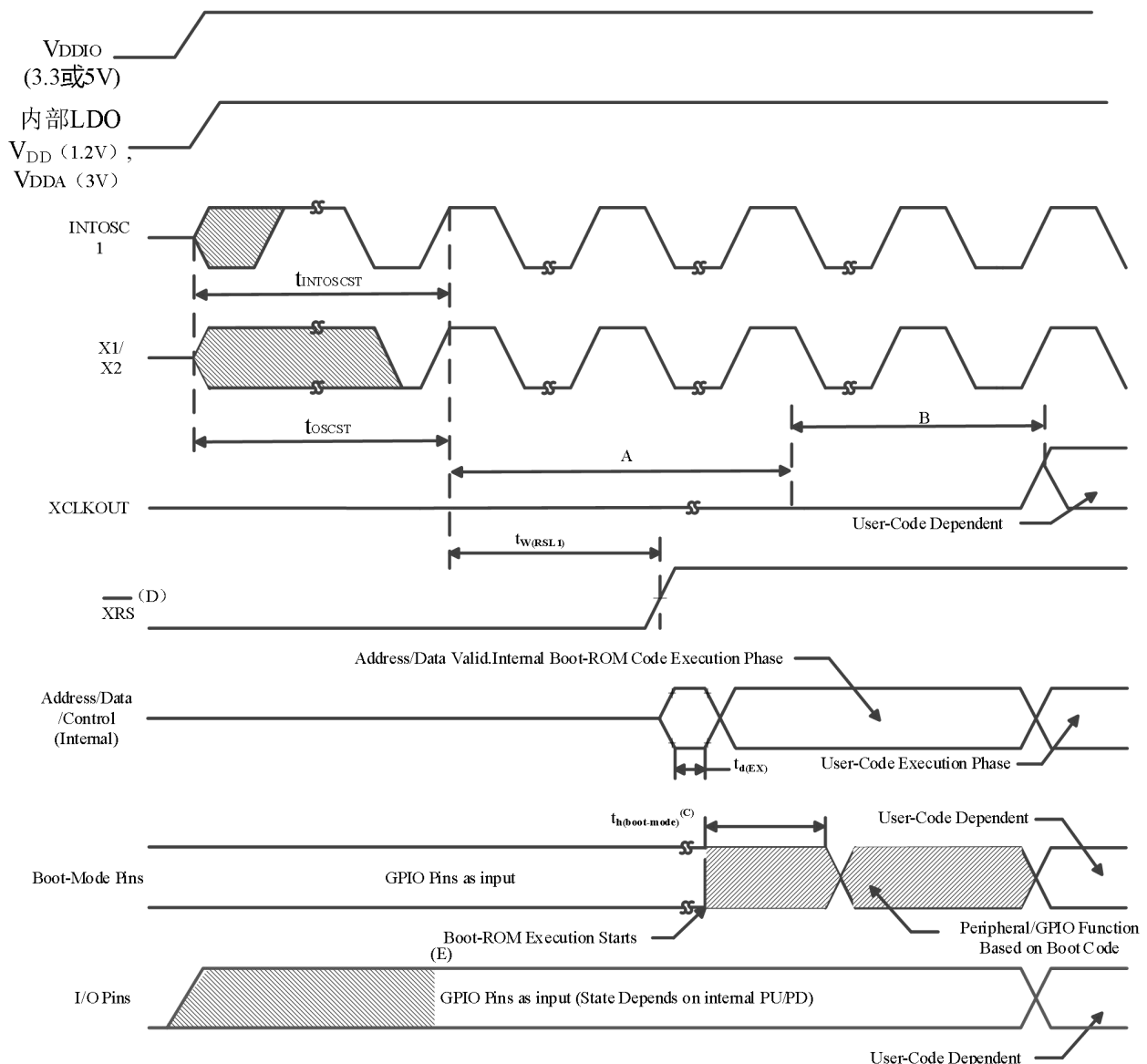
A. XCLKIN 与 XCLKOUT 的关系取决于所选择的分频系数。图示波形关系仅用于说明时序参数, 可能因实际配置而异。

B. 配置XCLKOUT以反映SYSCLKOUT。

图5-6 时钟时序

5.9 电源时序

复位后，无需电源时序来确保器件处于正常状态，或者防止上电/断电期间 I/O 上的毛刺脉冲（GPIO19，GPIO34–38 上无毛刺脉冲 I/O）。在器件上电之前，不得将高于 V_{DDIO} 以上的二极管压降 (0.7V) 的电压应用于任何数字引脚上（对于模拟引脚，该值是比 V_{DDA} 高 0.7V 的电压值）。此外， V_{DDIO} 和 V_{DDA} 之间的压差应一直在 0.2V 以上，保证 V_{DDA} 对应的 LDO 能够正常运行。施加在未上电器件引脚上的电压会以非法方式正偏内部 p-n 结，并产生无法预料的结果。



- A. 上电时，SYSCLKOUT 为 OSCCLK/4。由于 XCLK 寄存器内的 XCLKOUTDIV 位的复位状态是 0，SYSCLKOUT 在 XCLKOUT 输出之前会进一步进行 4 分频。在该状态期间， $XCLKOUT = OSCCLK/16$ 。
- B. 引导 ROM 将 DIVSEL 位配置为 /1 运行。在该状态期间， $XCLKOUT = OSCCLK/4$ 。请注意：XCLKOUT 只有通过用户代码明确配置，才会显示在引脚上。
- C. 复位后，引导 ROM 代码采样 Boot Mode（引导模式）引脚。基于引导模式引脚的状态，引导代码向目的内存或者引导代码函数下达分

支指令。如果引导 ROM 代码在上电条件后（在调试器环境中）执行代码，引导代码执行时间由当前SYSCLKOUT的速度而定。

SYSCLKOUT 将基于用户环境并可在PLL启用或者不启用时使用。如果引导ROM 代码在上电条件后（在调试器环境中）执行代码，引导代码执行时间由当前的SYSCLKOUT的速度而定。SYSCLKOUT将基于用户环境并可在PLL 启用或者不启用时使用。

D. 由于片载上电复位 (POR) 电路，使用 $\overline{\text{XRS}}$ 引脚是可选的。

E. 当触发BOR、POR时，所有GPIO都会变为高阻状态，POR、BOR释放后，内部上拉/下拉将起作用。

图5-7 上电复位

表 5-9 复位 $\overline{\text{XRS}}$ 时序要求

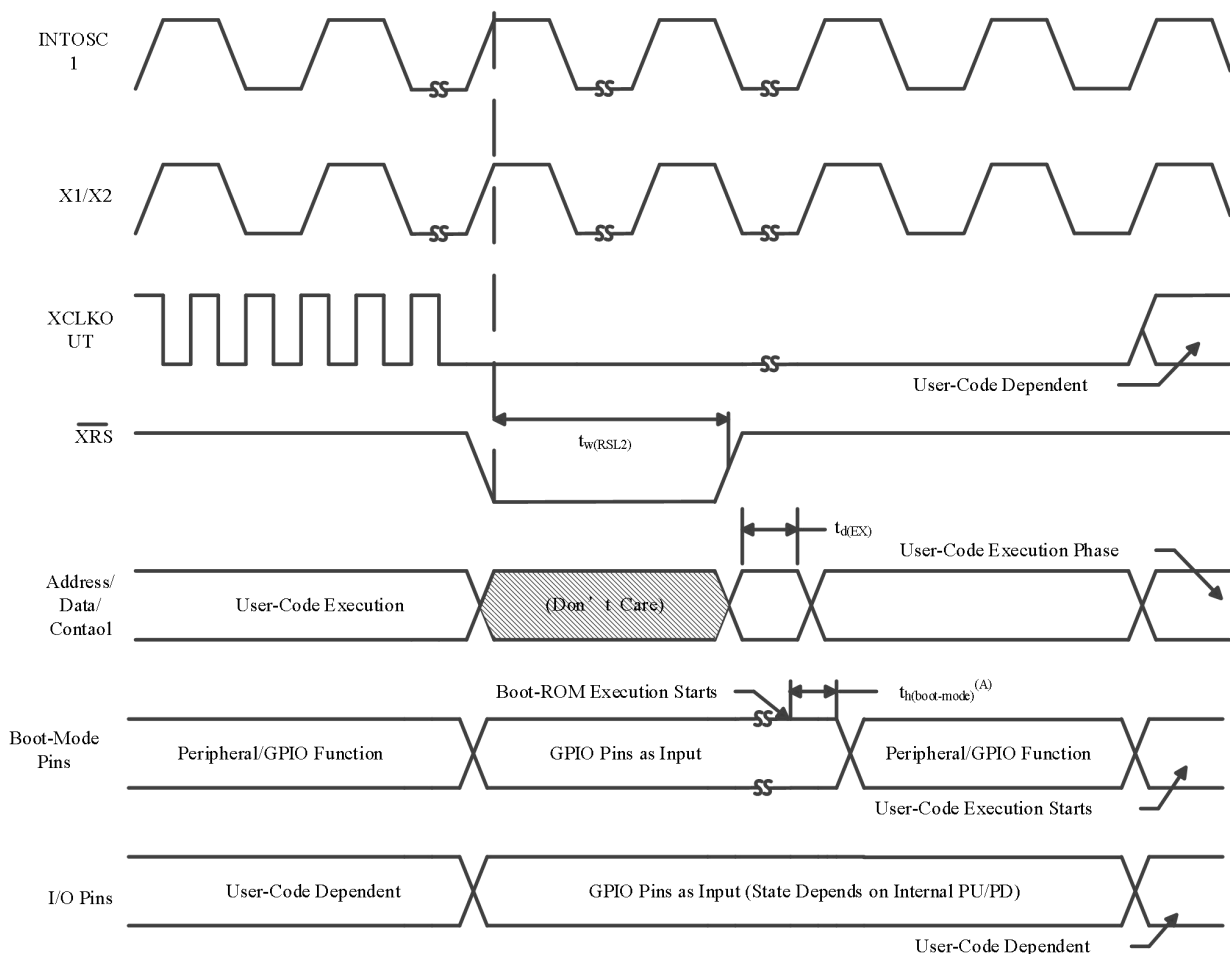
	最小值	典型值	最大值	单位
t_h (引导模式) 引导模式引脚的保持时间	$1000t_{c(SCO)}$			周期
$t_w(RSL2)$ 脉冲持续时间, $\overline{\text{XRS}}$ 低电平的时间	$32t_{c(OSCCLK)}$			周期

表 5-10 复位 $\overline{\text{XRS}}$ 开关特性

在推荐运行条件下（除非另行说明）

	测试条件	最小值	典型值	最大值	单位
$t_w(RSL1)$ 脉冲持续时间, $\overline{\text{XRS}}$ 由器件驱动			600		us
$t_w(WDRS)$ 脉冲持续时间, 由看门狗生成复位脉冲			$512t_{c(OSCCLK)}$		周期
$t_d(EX)$ 延迟时间, $\overline{\text{XRS}}$ 高电平后, 地址/数据有效			$32t_{c(OSCCLK)}$		周期
$t_{INTOSCST}$ 启动时间, 内部零引脚振荡器			100		us
$t_{OSCST}^{(1)}$ 晶体振荡器启动时间		1	10		ms

(1) 取决于晶振/谐振器和电路板设计。



A. 复位后，引导 ROM 代码对 BOOT 模式引脚进行采样。基于引导模式引脚的状态，引导代码分支到目标内存或引导代码函数。如果引导 ROM 代码在上电条件后（在调试器环境中）执行代码，则引导代码执行时间基于当前 SYSCLKOUT 速度。SYSCLKOUT 将基于用户环境，可以启用或不启用 PLL。

图5-8 热复位

图5-9所示为写入PLLCR寄存器所产生的结果示例。在第一个阶段， $PLLCR=0x0004$ 且 $SYSCLKOUT=OSCCLK \times 2$ 。然后用 $0x0008$ 写入PLLCR。就在写入PLLCR寄存器后，PLL锁存阶段开始。在此阶段， $SYSCLKOUT=OSCCLK/2$ 。在PLL锁存完成后，SYSCLKOUT表示新的频率 $OSCCLK \times 4$ 。

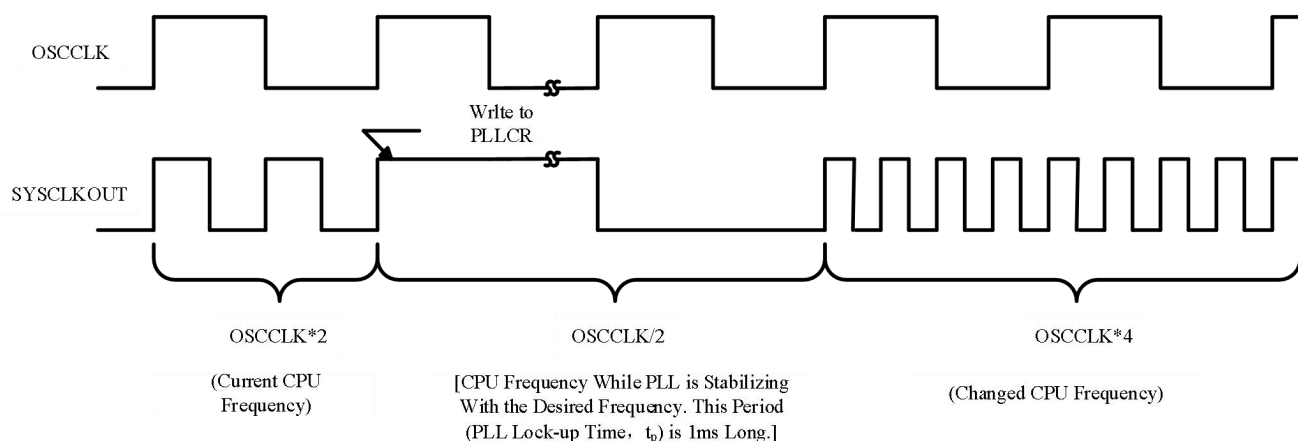


图5-9 写入PLLCR寄存器所产生的结果示例

5.10 详细说明

积分非线性

积分非线性是指每个独立代码从零至满刻度所画的一条直线上的偏离。在首次代码转换前，作为零点的点出现1/2 LSB。满刻度点被定义为超过最后一次代码转换的级别1/2 LSB。这个偏离为每一个特定代码的中心到这两个点之间的精确直线的距离。

微分非线性

一个理想ADC 显示分开距离恰好为1 个LSB 的代码转换。DNL 是与该理想值的偏差。小于±1 LSB 的微分非线性误差可确保无丢码。

零偏移

当模拟输入为零伏时，应当发生主进位转换。零误差被定义为实际转换与该点的偏差。

增益误差

第一个代码转换应该出现在比负满量程高1/2 LSB的模拟值 处。最后一次转换应该出现在低于标称满刻度的一个模拟值1.5 LSB 上。增益误差是首次和末次代码转换间的实际差异以及它们之间的理想差异。

信噪比+失真(SINAD)

SINAD 是测得的输入信号的均方根值与所有其它低于那奎斯特频率的频谱分量（包括谐波但不包括dc）的均方根总和的比。SINAD 的值用分贝表示。

有效位数(ENOB)

对于一个正弦波，SINAD 可用位的数量表示。使用下面的公式， $N = \frac{(SINAD - 1.76)}{6.02}$ 有可能获得以 N（有效位数）表达的性能测量值。因此，对于在给定输入频率上用于正弦波输入的器件的有效位数可从这个测得的SINAD 直接计算。

总谐波失真(THD)

THD 是前9个谐波分量的均方根总和与测得的输入信号的均方根值的比并表达为百分比或者分贝值。

无杂散动态范围(SFDR)

SFDR 是输入信号均方根振幅与峰值寄生信号间以分贝为单位的差值。

5.11 闪存时序

表 5-11 闪存/ OTP 对于温度材料的耐受度⁽¹⁾

	擦除/编程温度	最小值	典型值	最大值	单位
N _f 闪存对于阵列的耐受度（写入/擦除周期）	-40°C 至 125°C（环境温度）				周期
N _{OTP} OTP 对于阵列的耐受度（写入周期）	-40°C 至 125°C（环境温度）			1	写入

(1) 所示温度范围之外的写入/擦除操作并未说明，有可能影响耐受数。

表 5-12 100MHz SYSCLKOUT 上的闪存参数

参数	最小值	典型值	最大值	单位
编程时间 ⁽¹⁾ 1 个字（16bit）		240		us
擦除时间 ⁽¹⁾ 1 个扇区（8K*16bit）		333.9		ms
I _{DDREAD} ⁽²⁾ 读操作时 V _{DD} 流耗				mA
I _{DDPROG} ⁽²⁾ 编程期间的 V _{DD} 流耗				mA
I _{DDERASE} ⁽²⁾ 擦除期间的 V _{DD} 流耗				mA

(1) 当器件出货时，片载闪存存储器处于擦除状态。当首次编辑器件时，在编程前无需擦除闪存存储器。然而，对于所有随后的编程操作，需要执行擦除操作。

(2) 室温下包括函数调用开销在内的典型参数，是单独测试 Flash 时的参数。

表 5-13 闪存/ OTP 访问时序

参数	最小值	最大值	单位
t _a 闪存数据读出时间	40		ns
t _a (OTP) OTP 数据读出时间	60		ns

表 5-14 闪存数据保持持续时间

参数	测试条件	典型值	单位
t _{保持} 数据保持持续时间	T=25°C	15	年

表 5-15 不同频率上所需最小的闪存/一次性可编程 (OTP) 等待状态

SYSCLKOUT (MHz)	SYSCLKOUT (ns)	页等待状态 ⁽¹⁾	随机等待状态 ⁽¹⁾	OTP 等待状态
100	10	3	3	5
80	12.5	3	3	4
60	16.67	2	2	3
55	18.18	2	2	3
50	20	1	1	2
45	22.22	1	1	2
40	25	1	1	2
35	28.57	1	1	2
30	33.33	1	1	1
25	40	0	1	1

(1) 随机等待状态必须大于或者等于1, 用于计算表5-15中页等待状态和随机等待状态的公式如下:

页等待状态 = $\left\lceil \left(\frac{t_a}{t_{c(SCO)}} \right) - 1 \right\rceil$ (四舍五入到下一个最大的整数)

随机等待状态 = $\left\lceil \left(\frac{t_a}{t_{c(SCO)}} \right) - 1 \right\rceil$ (四舍五入到下一个最大的整数, 或 1, 以最大的数为准)

计算表5-15中OTP等待状态的公式如下:

OTP 等待状态 = $\left\lceil \left(\frac{t_a (OTP)}{t_{c(SCO)}} \right) - 1 \right\rceil$ (四舍五入到下一个最大的整数, 或 1, 以最大的数为准)

6 热性能/机械数据

表 6-1 显示了散热数据，表格之后的机械封装图反映了针对指定器件最新发布的机械数据。

表 6-1 散热模型 48 引脚 QP 结果

参数	0lfm	150lfm	250lfm	500lfm
θ_{JA} [°C/W] 高 k PCB	47.9	41.62	38.51	38.91
Ψ_{JT} [°C/W]	0.42			
Ψ_{JB}	28.98			
θ_{JC}	10.36			
θ_{JB}	28.16			

LQFP48

单位: mm

单位: mm

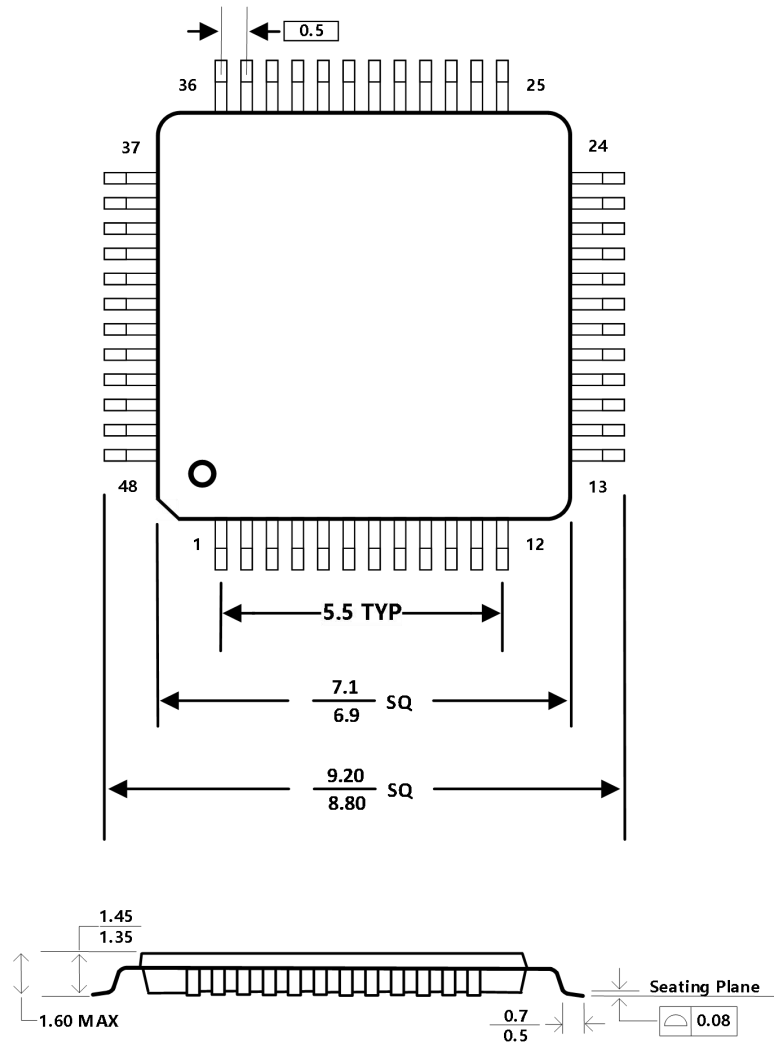
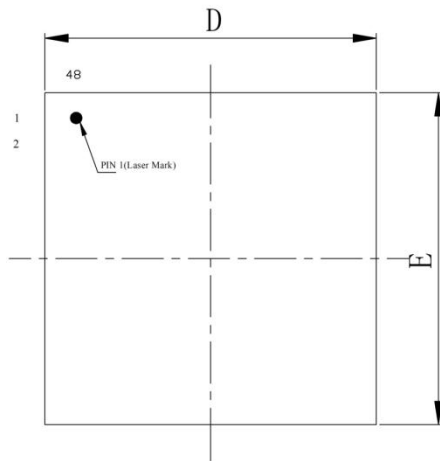


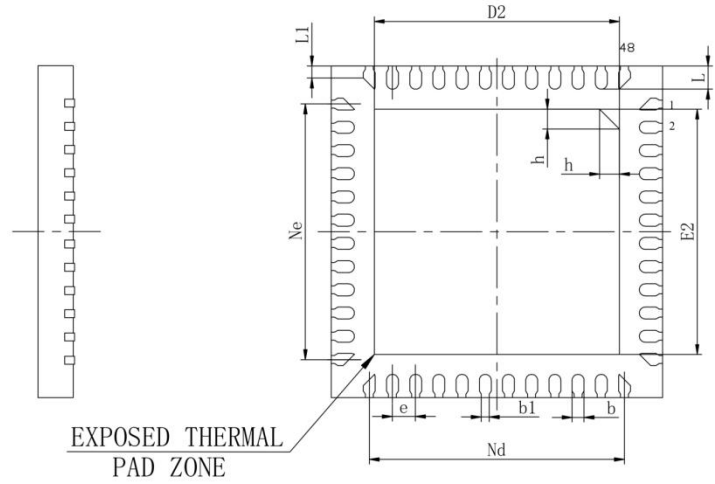
图 6-1 LQFP48 封装外形尺寸图

QFN48

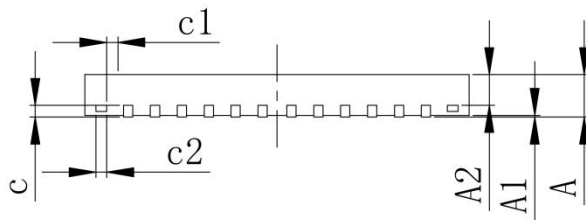
单位: mm



TOP VIEW



BOTTOM VIEW



SIDE VIEW

SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.80	0.85	0.90
A1	0	0.02	0.05
A2	0.40REF		
b	0.13	0.18	0.23
b1	0.12REF		
c	0.10	0.15	0.20
c1	0.145REF		
c2	0.140REF		
D	4.90	5.00	5.10
D2	3.60	3.70	3.80
e	0.35BSC		
Ne	3.85BSC		
Nd	3.85BSC		
E	4.90	5.00	5.10
E2	3.60	3.70	3.80
L	0.30	0.35	0.40
L1	0.13	0.18	0.23
h	0.25	0.30	0.35
载体尺寸 (mil)	154X154		

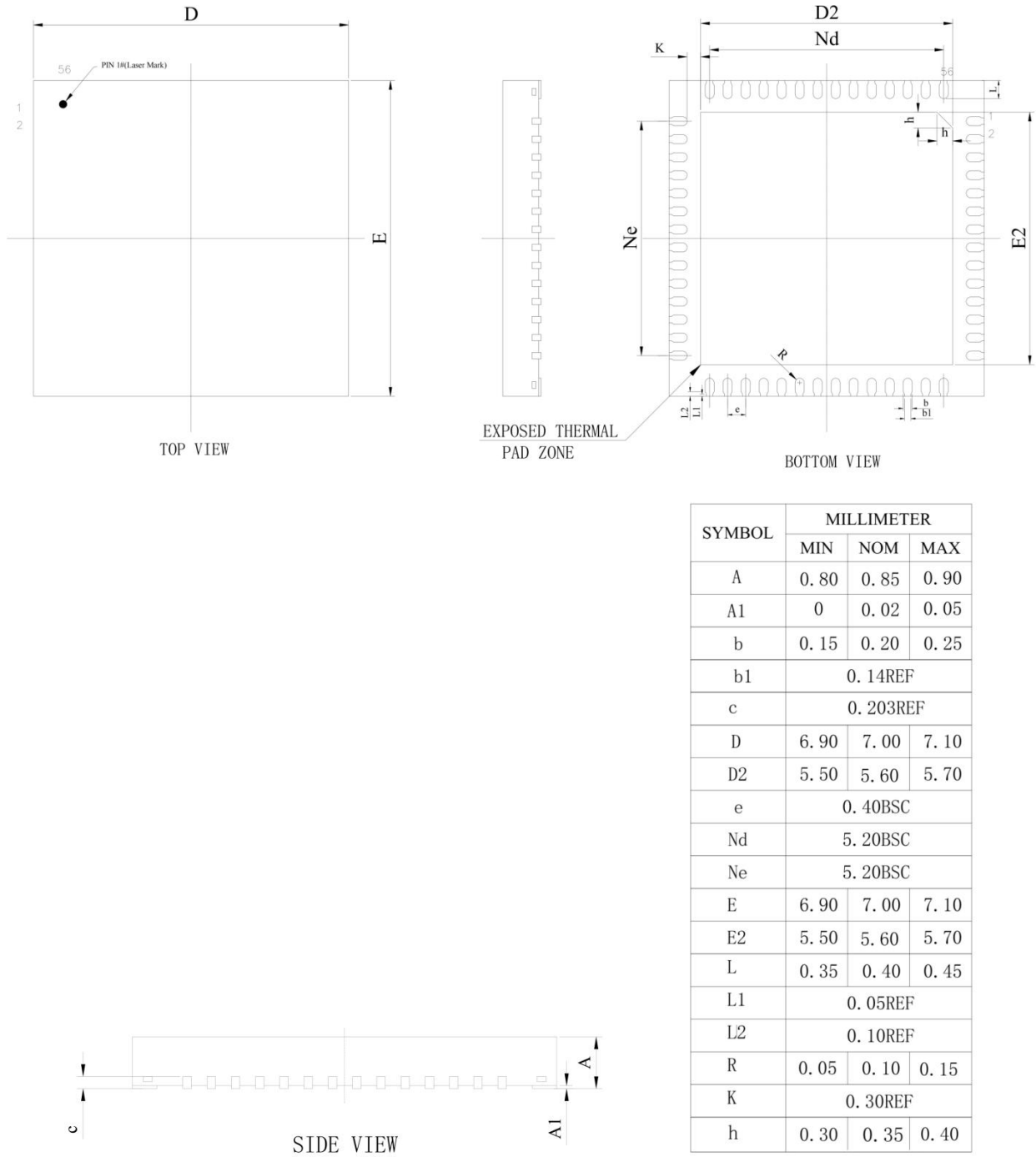


特殊设计: 框架厚度0.152

图 6-2 QFN48 封装外形尺寸图

QFN56

单位: mm



*** 特殊设计: 无

图 6-3 QFN56 封装外形尺寸图

LQFP32

单位: mm

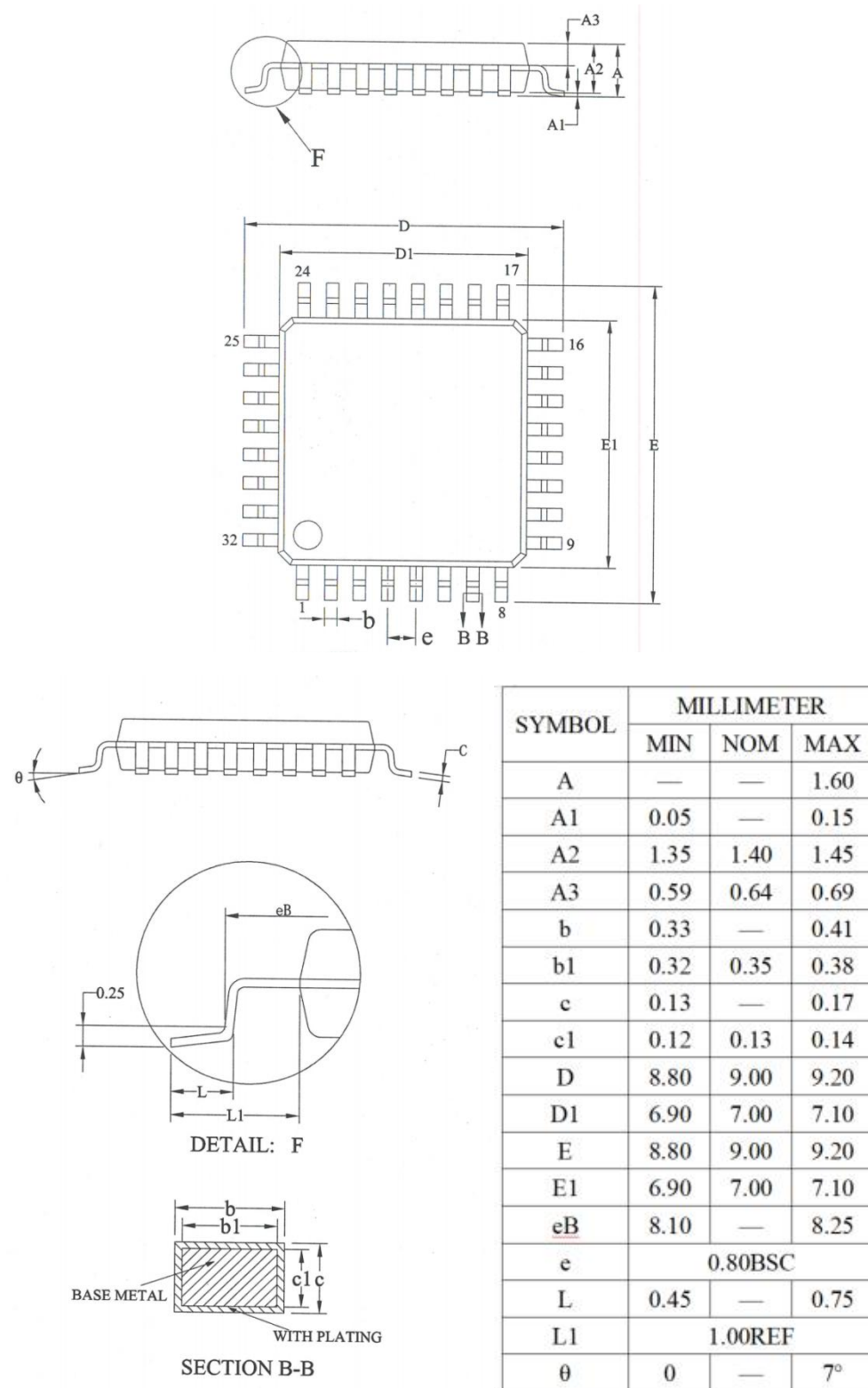
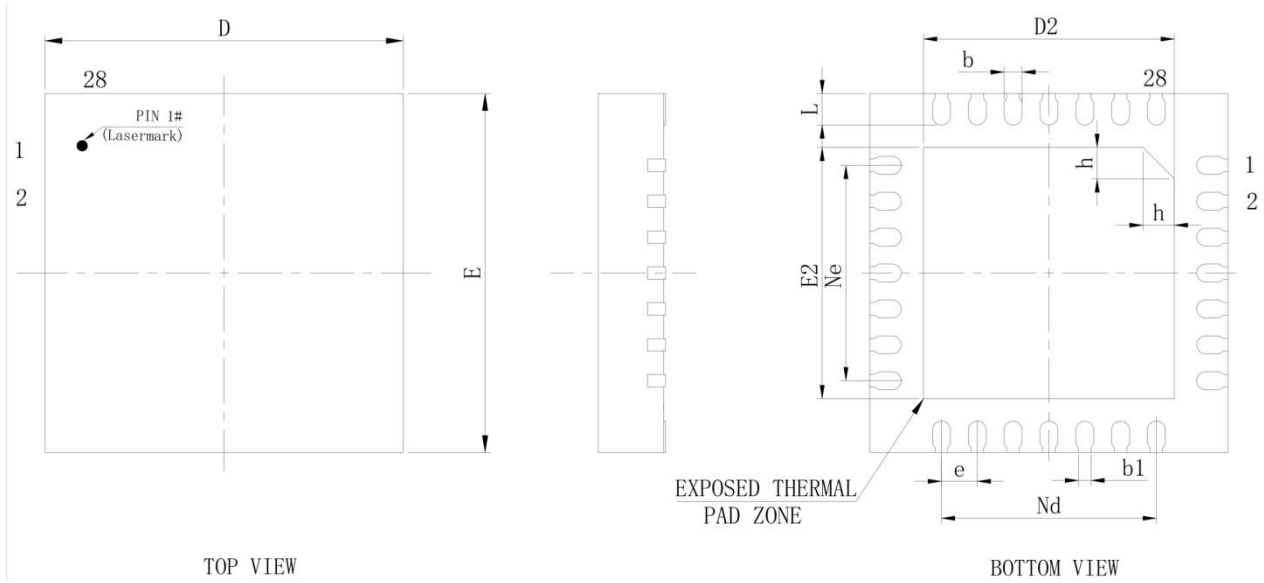


图 6-4 LQFP32 封装外形尺寸图

QFN28

单位: mm



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
A1	0	0.02	0.05
b	0.15	0.20	0.25
b1	0.14REF		
c	0.18	0.20	0.25
D	3.90	4.00	4.10
D2	2.70	2.80	2.90
e	0.40BSC		
Ne	2.40BSC		
Nd	2.40BSC		
E	3.90	4.00	4.10
E2	2.70	2.80	2.90
L	0.30	0.35	0.40
h	0.30	0.35	0.40

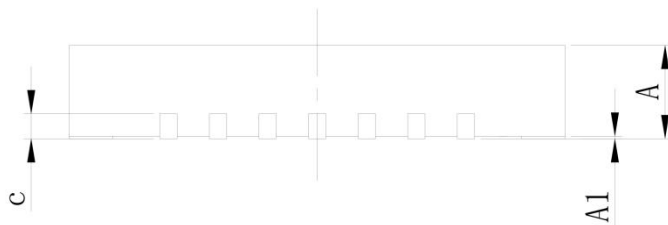


图 6-5 QFN28 封装外形尺寸图

TSSOP20

单位: mm

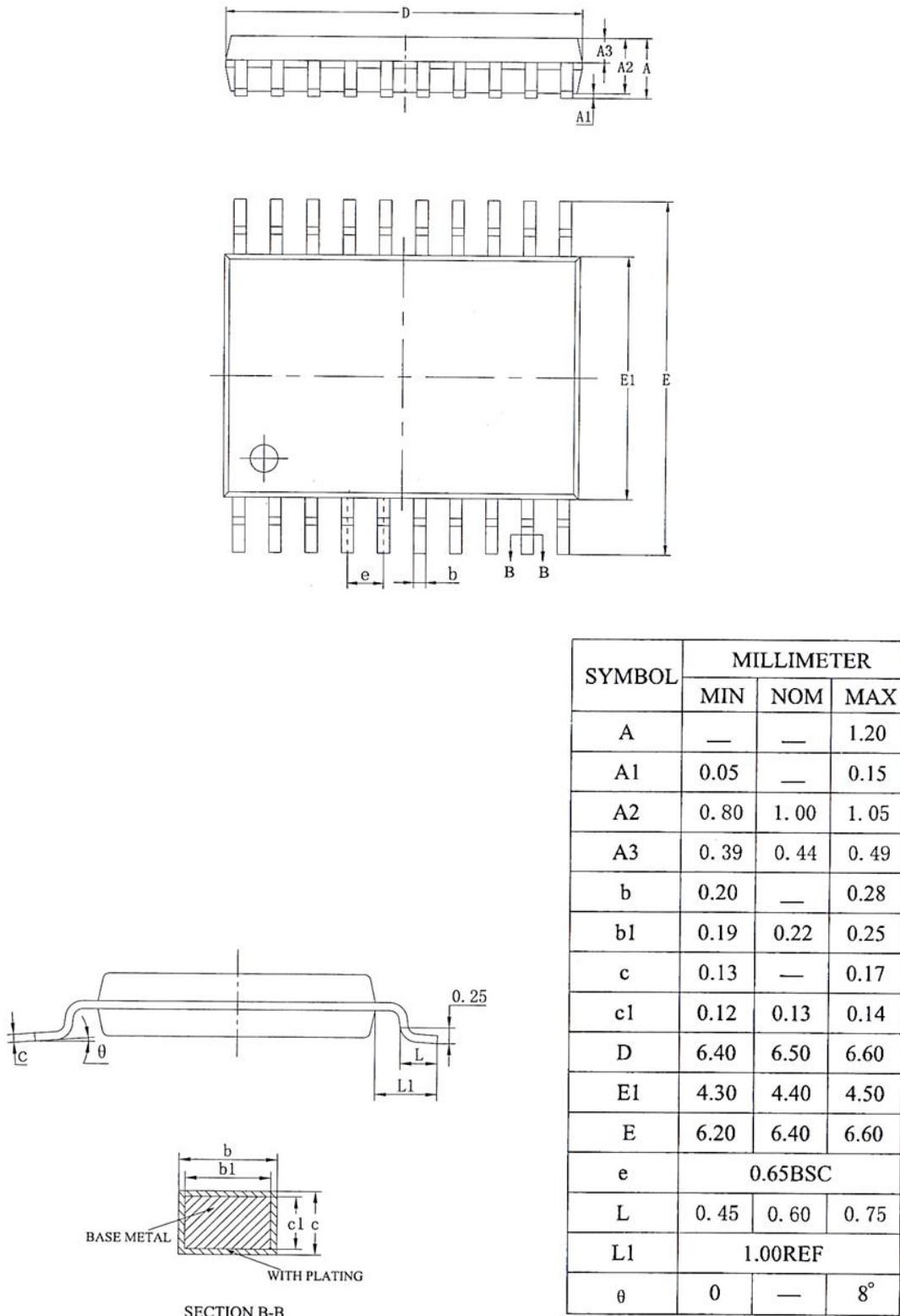


图 6-6 TSSOP20 封装外形尺寸图

联系方式

公司网址: www.advancechip.com

联系邮箱: sales@advancechip.com

销售联系电话: 0731-88731027 (长沙)

公司总部地址: 长沙市湘江新区东方红街道北斗产业园·黄金园 A5 栋

南京销售中心: 南京市秦淮区卡子门大街 19 号紫云智慧广场 6 号楼 15 层

