# AVP32F06X 数字信号处理器

# 数据手册

编号：JXDZ3.590.018SJSC

Advancechip Electronics

## 湖南进芯电子科技有限公司

2024 年 3 月

V1.0

# 目 次

**1. 产品概述 ..... 1**

1.1. 特性 ..... 1

1.2. 市场应用 ..... 2

1.3. 产品描述 ..... 3

1.4. 功能框图 ..... 4

1.5. 系统模块图 ..... 5

1.6. 产品编码 ..... 6

**2. 器件对比表 ..... 7**

**3. 终端配置和功能 ..... 8**

3.1. 引脚图 ..... 8

3.2. 信号说明 ..... 9

**4. 技术规格 ..... 18**

4.1. 最大绝对额定值 ..... 18

4.2. 建议运行条件 ..... 18

4.3. 流耗 ..... 19

4.4. 电气特性 ..... 21

4.5. 热阻特性 ..... 22

4.6. 热设计注意事项 ..... 23

4.7. 针对 DSP 的无信号缓冲的仿真器连接 ..... 23

4.8. 参数信息 ..... 24

4.9. 测试负载电路 ..... 25

4.10. 上电时序 ..... 25
4.11. 时钟特性 ..... 27
4.12. 闪存时序 ..... 30
4.13. ESD 等级 ..... 31
**5. 详细描述 ..... 32**
5.1. 概述 ..... 32
5.2. 内存映射 ..... 40
5.3. 寄存器映射 ..... 44
5.4. 器件仿真寄存器 ..... 46
5.5. VREG, BOR, POR ..... 46
5.6. 系统控制 ..... 48
5.7. 低功耗模式 ..... 56
5.8. 中断 ..... 57
5.9. 外围设备 ..... 61
**6. 封装信息 ..... 136**
**7. 重要注意事项及声明 ..... 138**

# 1. 产品概述

## 1.1. 特性

- **高性能32位浮点处理器内核（AVP32）**
  - 主频 90MHz（单周期 11.11ns）
  - 支持 16×16 和 32×32 乘累加（MAC）运算
  - 支持 16×16 双 MAC 运算
  - 支持单精度浮点（FPU）运算
  - 哈佛（Harvard）总线架构
  - 连动运算
  - 快速中断响应和处理
  - 统一存储器编程模型
  - 高效代码（使用 C/C++和汇编语言）
- **可编程控制环路加速器（CLA）**
  - 32 位浮点数学加速器
  - 代码执行独立于主 CPU 之外
- **维特比、复杂数学、CRC单元（VCU）**
  - 以扩展指令集方式支持复杂的乘法、维特比运算和循环冗余校验（CRC）
- **嵌入式存储器**
  - 高达 256KB 的闪存
  - 高达 100KB 的随机存取存储器（RAM）
  - 2KB 一次性可编程（OTP）ROM
- **6通道直接内存存取（DMA）**
- **低器件和系统成本**
  - 3.3V 单电源供电，无需电源上电排序
  - 集成上电复位和欠压复位监控
  - 低功耗操作模式
  - 无模拟支持引脚
- **字节序：小端字节序**
- **支持JTAG边界扫描**
  - IEEE 标准 1149.1-1990 标准测试访问端口和边界扫描架构
- **时钟**
  - 内部两个零引脚振荡器
  - 片载晶体振荡器/外部时钟输入
  - 看门狗定时器模块
  - 丢失时钟检测电路
- **可支持所有外设中断的外设中断扩展(PIE)模块**
- **三个32位CPU定时器**
- **高级控制外设**
- **8个增强型脉冲宽度调制器(ePWM)模块**
  - 总共 16 个 PWM 通道（可支持 8 个 HRPWM）
  - 每个模块中的独立 16 位定时器
- **3个输入增强型捕获(eCAP) 模块**
- **多达4个高分辨率捕获(HRCAP) 模块**
- **多达2个增强型正交编码器脉冲(eQEP)模块**
- **12位模数转换器(ADC)，具有双路采样与保持(S/H)功能**
  - 高达 3.46MSPS
  - 高达 16 通道
- **片载温度传感器**
- **128 位安全密钥和锁**
  - 保护安全内存块
  - 防止固件逆向工程
- **串行端口外设**
  - 两个串行通信接口（SCI）模块
  - 两个串行外设接口（SPI）模块
  - 一个内部集成电路（I2C）总线
  - 一个多通道缓冲串行端口（McBSP）总线
  - 一个增强型控制器局域网络（eCAN)
  - 通用串行总线 USB 2.0 规范
    - 全速器件模式
    - 全速/低速主机模式
- **多达54个具有输入滤波功能的独立可编程、多路复用通用输入/输出（GPIO）引脚**
- **高级仿真特性**
  - 分析和断点功能
  - 通过硬件进行实时调试
- **封装选项**
  - 100 引脚 QP 薄型四方扁平封装（LQFP）
  - 80 引脚 QP 薄型四方扁平封装（LQFP）
- **温度选项**
  - S：-40°C 至 125°C
  - S1：-55°C 至 125°C（筛选）
  - Q：-40°C 至 125°C（通过 AEC-Q100 认证）

## 1.2. 市场应用

- 空调室外机
- 电梯门自动启闭装置驱动控制
- 逆变器和电机控制
- 车载充电器（OBC）和无线充电器
- 自动分拣设备
- CNC 控制
- 纺织机
- 焊接机
- 电动汽车充电站电源模块
- 车辆无线充电模块
- 能量存储电源转换系统（PCS）
- 中央逆变器
- 微型逆变器
- 太阳能电源优化器
- 串式逆变器
- 交流驱动器控制模块
- 交流驱动器功率级模块
- 线性电机功率级
- 伺服驱动器控制模块
- 伺服驱动器功率级模块
- 交流输入 BLDC 电机驱动器
- 直流输入 BLDC 电机驱动器
- 工业交流/直流转换器
- 三相UPS

## 1.3. 产品描述

ADP32 和 AVP32 系列数字信号处理器，在处理、传感和驱动方面进行了优化，可提高实时控制应用中的闭环处理性能，例如工业电机驱动、光伏逆变器和数字电源、电动车辆与运输、电机控制以及传感和信号处理。AVP32F069 是继 ADP32F035 后开发的又一颗 32 位数字信号处理器，在主频提升、存储容量增加的基础上，引入了单精度浮点运算单元以及控制环加速协处理器，进一步增强 32 位系列产品的处理能力，扩展了产品的应用范围。

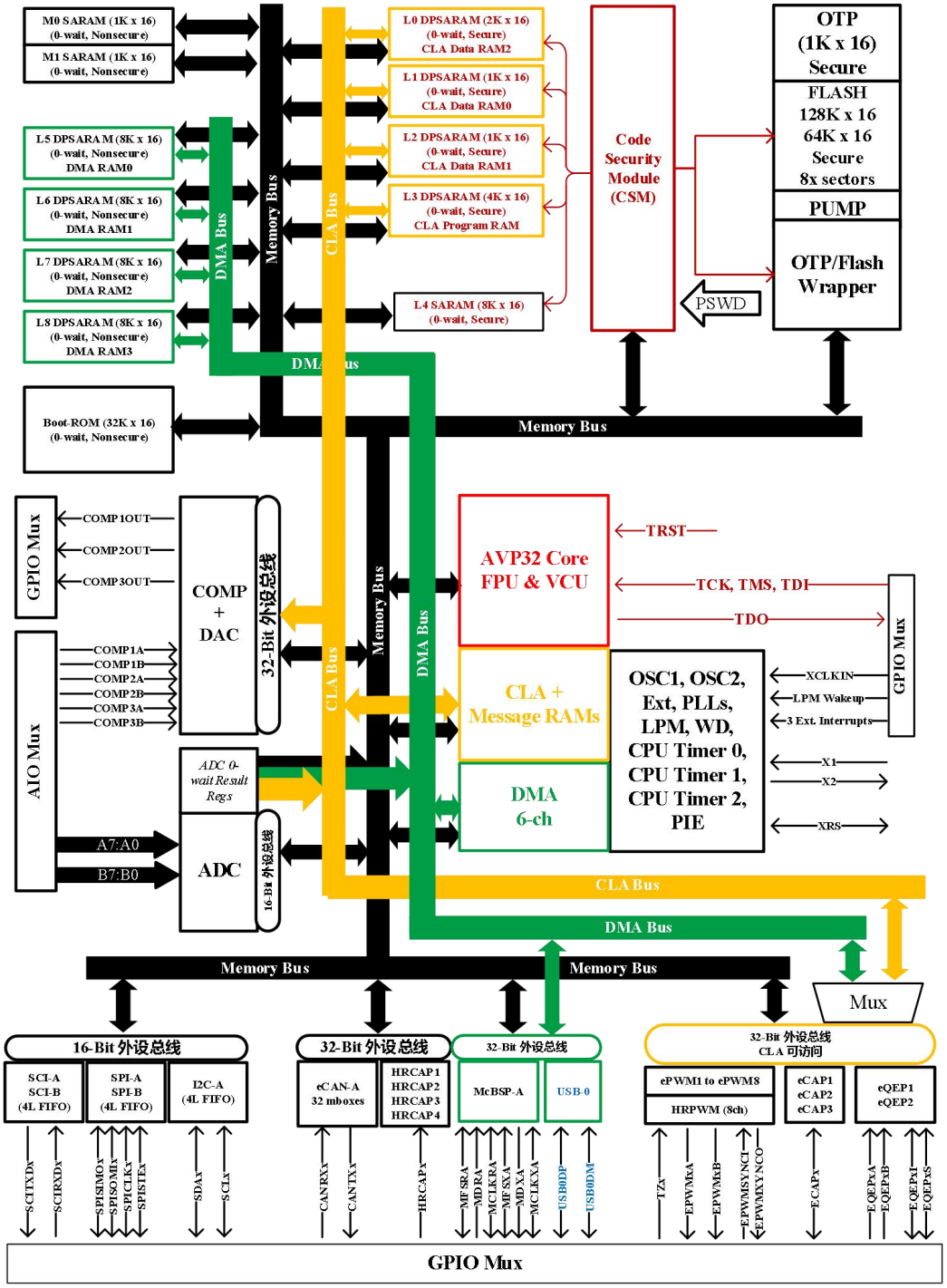
AVP32F069 数字信号处理器（DSP）提供了包含 AVP32 浮点内核以及控制环加速器（CLA）的强大处理能力。实现了与 ADP32F035 器件的代码兼容，以及性能升级需求。以较少的引脚输出，高度集成控制外设和高级别的模拟集成。两个内部线性稳压电源实现了单电源轨运行。对 HRPWM 模块实施了改进，以提供双边沿控制（调频）。器件内还新增了采用 10 位内部基准的模拟比较器，可通过与其直接相连来控制 ePWM 输出。ADC 可在 0V 至 3.3V 满量程的范围内实施转换，支持 VREFHI/VREFLO 基准的比例运算。ADC 接口已针对低开销和延迟进行了优化。

**器件信息**

| 型号 | 封装 | 尺寸 |
|---|---|---|
| **AVP32F069QP80S** | LQFP 80 | 12mm x 12mm |
| **AVP32F069QP100S** | LQFP 100 | 14mm x 14mm |
| **AVP32F062QP80S** | LQFP 80 | 12mm x 12mm |
| **AVP32F062QP100S** | LQFP 100 | 14mm x 14mm |

## 1.4. 功能框图

图 1-1 所示为器件功能框图。




注释：因引脚复用原因，并非所有功能外设可同时使用。

**图 1-1 功能框图**

## 1.5. 系统模块图

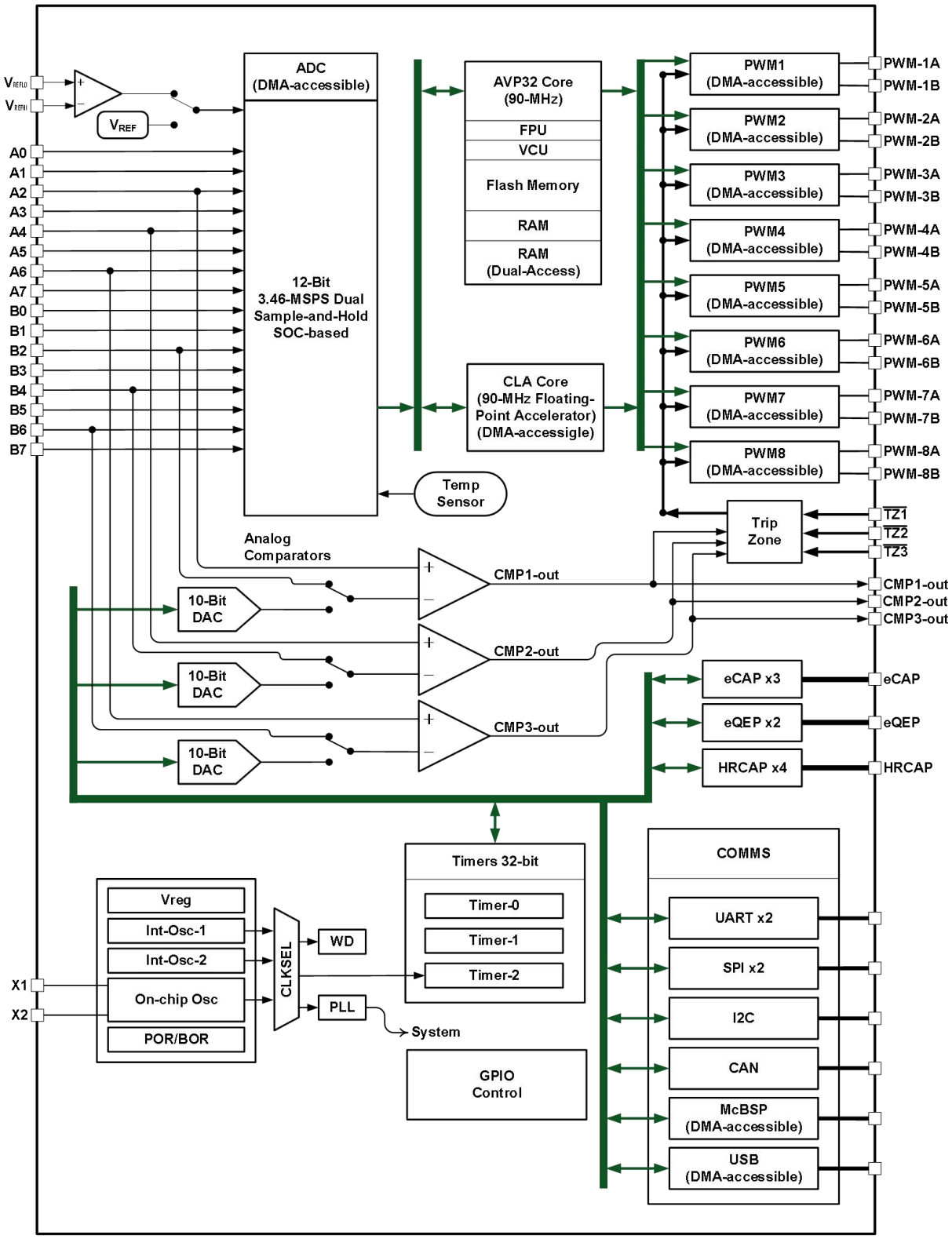
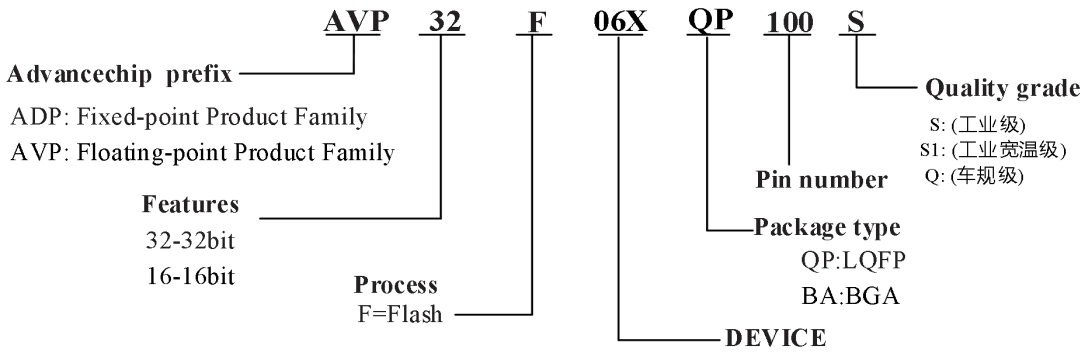



**图 1-2 外设模块**

## 1.6. 产品编码

AVP32F06X 产品代号编码规则：




器件标识：

器件正面　　器件背面

# 2. 器件对比表

**表 2-1 器件对比**

| 特性 | | AVP32F069QP100S | AVP32F069QP80S | AVP32F062QP100S | AVP32F062QP80S |
|---|---|---|---|---|---|
| 指令周期 | | 11.11sns | 11.11ns | 11.11sns | 11.11ns |
| 浮点运算(FPU) | | 单精度 | 单精度 | 单精度 | 单精度 |
| VCU | | 有 | 有 | 无 | 无 |
| CLA | | 有 | 有 | 无 | 无 |
| 6 通道 DMA | | 有 | 有 | 有 | 有 |
| 片载 Flash(16 位字) | | 128K | 128K | 64K | 64K |
| 片载 SARAM(16 位字) | | 50K | 50K | 26K | 26K |
| 加密模块(CSM) | | 有 | 有 | 有 | 有 |
| 引导 ROM(16 位字) | | 32K | 32K | 32K | 32K |
| OTP ROM(16 位字) | | 1K | 1K | 1K | 1K |
| ePWM 通道 | | 16 | 14 | 16 | 14 |
| HRPWM 通道 | | 8 | 8 | 8 | 8 |
| eCAP 输入 | | 3 | 3 | 3 | 3 |
| HRCAP | | 4 | 1 | 4 | 1 |
| eQEP 模块 | | 2 | 1 | 2 | 1 |
| 看门狗 | | 有 | 有 | 有 | 有 |
| 12 位 ADC | 转换速率 (MSPS) | 3.46 | 3.46 | 3.46 | 3.46 |
| | 转换时间 | 289ns | 289ns | 289ns | 289ns |
| | 通道数 | 16 | 12 | 16 | 12 |
| | 温度传感器 | 有 | 有 | 有 | 有 |
| | 采样保持电路 | 有 | 有 | 有 | 有 |
| 32 位 CPU 定时器 | | 3 | 3 | 3 | 3 |
| 集成 DAC 的比较器 | | 3 | 3 | 3 | 3 |
| I2C | | 1 | 1 | 1 | 1 |
| McBSP | | 1 | 1 | 1 | 1 |
| eCAN | | 1 | 1 | 1 | 1 |
| SPI | | 2 | 2 | 2 | 2 |
| SCI/UART | | 2 | 2 | 2 | 2 |
| USB | | 1 | 1 | 1 | 1 |
| 双引脚振荡器 | | 1 | 1 | 1 | 1 |
| 零引脚振荡器 | | 2 | 2 | 2 | 2 |
| I/O 引脚 (共享) | GPIO | 54 | 40 | 54 | 40 |
| | AIO | 6 | 6 | 6 | 6 |
| 外部中断 | | 3 | 3 | 3 | 3 |
| 供电电压 | | 3.3 V | 3.3 V | 3.3 V | 3.3 V |
| 温度范围(1) | | -40℃~125℃ | -40℃~125℃ | -40℃~125℃ | -40℃~125℃ |

（1）设计指标为环境温度-40℃~125℃，结温-40℃~150℃；最终产品的温度范围需参考产品的实际温度可靠性测试结果。

# 3. 终端配置和功能

## 3.1. 引脚图

图 3-1 所示为 100 引脚薄型四方扁平封装（LQFP）的引脚分布。

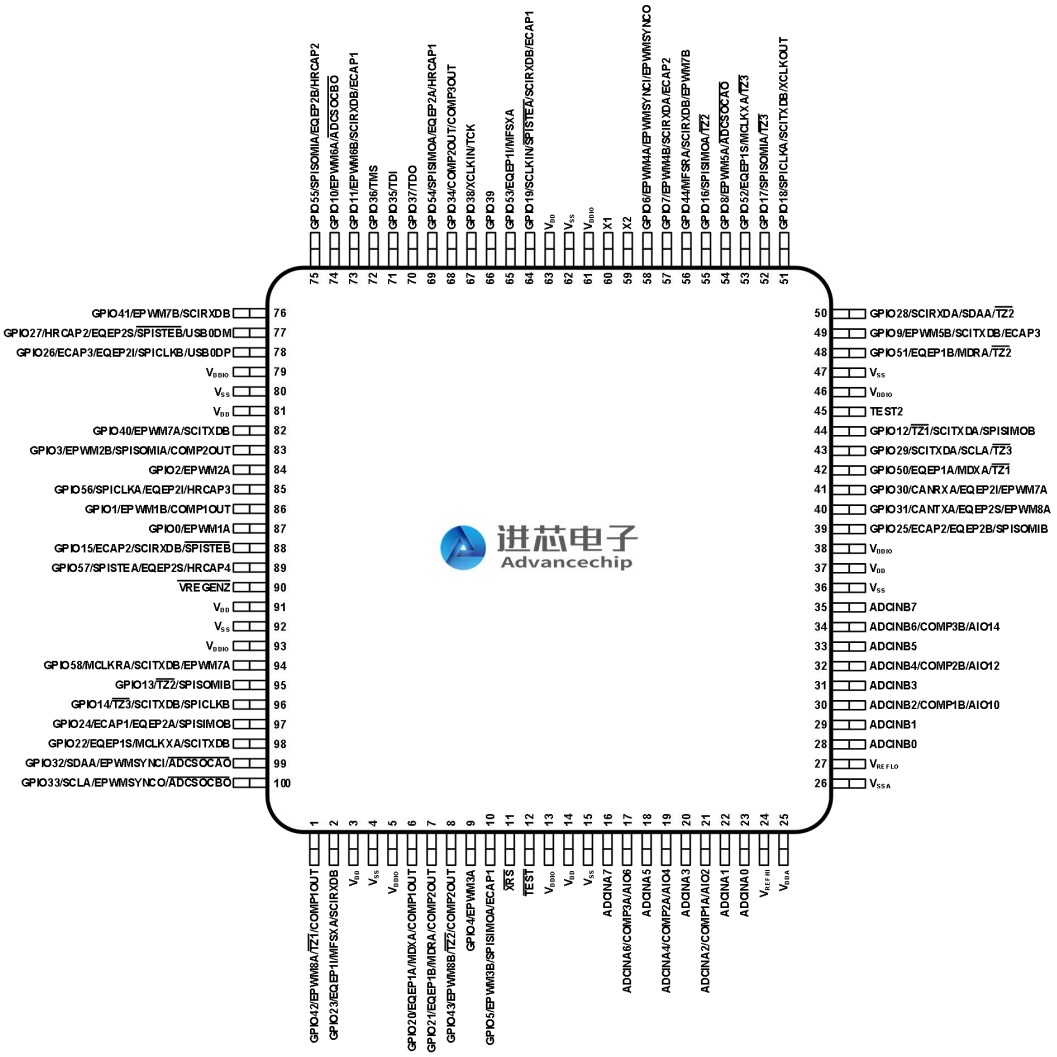



**图 3-1 100 引脚 LQFP 封装（顶视图）**

图 3-2 所示为 80 引脚薄型四方扁平封装（LQFP）的引脚分布。

**图 3-2 80 引脚 LQFP 封装（顶视图）**

## 3.2. 信号说明

第 3.2.1 节对信号进行了描述。除了 JTAG 引脚外，其他引脚复位后，默认是 GPIO 功能（除非另有说明）。模拟 IO 默认为 ADC 通道，有些外设功能并非在所有器件上提供。输入不是 5V 耐压。所有 GPIO 引脚都为 I/O/Z，且有一个内部上拉电阻，可在每个引脚上有选择性的启用/禁用该内部上拉电阻。此特性仅适用于 GPIO 引脚。器件复位时，PWM 引脚上的上拉特性默认禁用。复位时，其他 GPIO 引脚上的上拉特性则默认开启。AIO 引脚没有内部上拉电阻。

**注意**

使用片载电压调节器（VREG）时，GPIO19、GPIO26–27 和 GPIO34–38 引脚可能在上电期间出现毛刺脉冲。该潜在毛刺脉冲将在读取引导模式引脚之前结束，不会影响引导行为。如果应用中不可接受出现此毛刺脉冲，可从 1.8V 外部电源供电。或者，可以考虑添加一个与这些引脚和任何外部驱动器串联的限流电阻器（例如 470Ω），以减小引脚和/或外部电路退

化的可能性。当使用外部 1.8 V 电源时，无电源时序要求。然而，如果输入/输出引脚的电平移位输出缓冲器中的 3.3-V 晶体管在 1.8-V 晶体管之前通电，则输出缓冲器可能会开启，从而导致上电期间引脚上出现毛刺脉冲。为避免此运行状态，给 $V_{DD}$ 引脚通电应早于对 $V_{DDIO}$ 引脚供电，或与之同时进行，以确保在 $V_{DDIO}$ 引脚达到 0.7 V 之前，$V_{DD}$ 引脚已达到 0.7 V。

### 3.2.1. 信号说明

| 引脚名称 | 封装编号 | | I/O/Z[1] | 说明 |
|---|---|---|---|---|
| | QP100 | QP80 | | |
| **JTAG** | | | | |
| $\overline{TRST}$ | 12 | 10 | I | 使用内部下拉电阻进行JTAG测试复位。当$\overline{TRST}$驱动为高电平时，使扫描系统获得器件运行的控制权。如果该信号未连接或驱动至低电平，则器件将在其功能模式下运行，并将忽略测试复位信号。<br>注意：$\overline{TRST}$是一个高电平有效测试引脚，且在正常运行期间必须始终保持低电平。该引脚上需要一个外部下拉电阻器。该电阻器的值应基于适用于该设计的调试器的驱动强度。2.2kΩ电阻可提供足够的保护。 |
| TCK | 参考GPIO38 | | I | 带有内部上拉电阻的JTAG测试时钟。(↑) |
| TMS | 参考GPIO36 | | I | 带有内部上拉电阻的JTAG测试模式选择(TMS)。该串行控制输入(TMS)在TCK上升沿输入到TAP控制器中。(↑) |
| TDI | 参考GPIO35 | | I | 带有内部上拉电阻的JTAG测试数据输入(TDI)。TDI在TCK的上升沿输入到选定的寄存器（指令或数据）中。(↑) |
| TDO | 参考GPIO37 | | O/Z | JTAG扫描输出，测试数据输出(TDO)。所选寄存器（指令或数据）的内容在TCK的下降沿从TDO移出。（8mA驱动） |
| **其它** | | | | |
| TEST2 | 45 | 36 | I/O | 测试引脚，内核1.1V输出引脚，保持悬空 |
| **时钟** | | | | |
| XCLKOUT | 参考GPIO18 | | O/Z | 取自SYSCLKOUT的输出时钟。XCLKOUT频率或与SYSCLKOUT相同、或为SYSCLKOUT频率的1/2或1/4，这是由XCLK寄存器中的位[1:0](XCLKOUTDIV)控制的。复位时，XCLKOUT=SYSCLKOUT/4。通过将XCLKOUTDIV设置为3，可以关闭XCLKOUT信号。GPIO18的IO复用控制也必须设置为XCLKOUT，以使该信号输出到引脚。 |
| XCLKIN | 参考GPIO19、GPIO38 | | I | 外部振荡器输入。时钟的引脚源由XCLK寄存器中的XCLKINSEL位控制，GPIO38是默认选择。一个外部的3.3V振荡器通过此引脚提供时钟。在这种情况下，如果X1引脚可用，X1必须连接到GND，并且通过CLKCTL寄存器中的第14位(XTALOSCOFF)禁用片载的晶体振荡器。如果使用晶振或谐振器，必须通过CLKCTL寄存器中的第13位(XCLKINOFF)禁用XCLKIN路径。 |
| X1 | 60 | 48 | I | 片载1.8V晶体振荡器输入。要使用这个振荡器，石英晶体或陶瓷谐振器必须连接在X1和X2之间。在这种情况下，XCLKIN路径必须被CLKCTL寄存器中的第13位(XCLKINOFF)所禁用。如果不使用此引脚，则必须将其连接到GND。 |
| X2 | 59 | 47 | O | 片载晶体振荡器输出。石英晶体或陶瓷谐振器必须连接在X1和X2之间。如果未使用X2，则必须保持未连接状态。 |

| 引脚名称 | 封装编号 | | I/O/Z[1] | 说明 |
|---|---|---|---|---|
| | QP100 | QP80 | | |
| **复位** | | | | |
| $\overline{XRS}$ | 11 | 9 | I/OD | 设备复位（输入）和看门狗复位（输出）。这些设备具有内置的上电复位(POR)和欠压复位(BOR)电路。在上电或欠压情况下，该引脚由芯片驱动到低电平。外部电路也可以驱动该引脚使器件复位生效。当看门狗复位发生时，这个引脚也被芯片驱动到低电平。在看门狗复位期间，$\overline{XRS}$引脚被驱动到低电平持续512个OSCCLK周期。在$\overline{XRS}$和$V_{DDIO}$之间放置一个值为2.2kΩ到10kΩ的电阻。如果一个电容被放置在$\overline{XRS}$和$V_{SS}$之间进行噪声滤波，则该电容应为100nF或更小。当看门狗复位时，这些电容值需允许看门狗在512个OSCCLK周期内正确地驱动$\overline{XRS}$引脚到$V_{OL}$。无论何种原因，设备复位都会导致设备终止执行。程序计数器(PC)指向地址0x3FFFC0。停用复位后，将从程序计数器(PC)指向位置开始执行。这个引脚的输出缓冲器是一个具有内部上拉的开漏设备。(↑)如果此引脚由外部设备驱动，则应使用开漏设备进行驱动。 |
| **ADC，比较器，模拟I/O** | | | | |
| ADCINA7 | 16 | – | I | ADC A组，通道7输入 |
| ADCINA6<br>COMP3A<br>AIO6 | 17 | 14 | I<br>I<br>I/O | ADC A组，通道6输入<br>比较器输入3A<br>数字AIO 6 |
| ADCINA5 | 18 | 15 | I | ADC A组，通道5输入 |
| ADCINA4<br>COMP2A<br>AIO4 | 19 | 16 | I<br>I<br>I/O | ADC A组，通道4输入<br>比较器输入2A<br>数字AIO 4 |
| ADCINA3 | 20 | – | I | ADC A组，通道3输入 |
| ADCINA2<br>COMP1A<br>AIO2 | 21 | 17 | I<br>I<br>I/O | ADC A组，通道2输入<br>比较器输入1A<br>数字AIO 2 |
| ADCINA1 | 22 | 18 | I | ADC A组，通道1输入 |
| ADCINA0 | 23 | 19 | I | ADC A组，通道0输入<br>**注意：$V_{REFHI}$和ADCINA0在80脚封装的器件上共用同一个引脚，且它们不能同时使用。** |
| $V_{REFHI}$ | 24 | 19 | | ADC外部高电平参考，仅在ADC外部基准模式下使用。详见第5.9.2.1节。<br>**注意：$V_{REFHI}$和ADCINA0在80脚封装的器件上共用同一个引脚，且它们不能同时使用。** |
| ADCINB7 | 35 | – | I | ADC B组，通道7输入 |
| ADCINB6<br>COMP3B<br>AIO14 | 34 | 27 | I<br>I<br>I/O | ADC B组，通道6输入<br>比较器输入3B<br>数字AIO 14 |
| ADCINB5 | 33 | 26 | I | ADC B组，通道5输入 |
| ADCINB4<br>COMP2B<br>AIO12 | 32 | 25 | I<br>I<br>I/O | ADC B组，通道4输入<br>比较器输入2B<br>数字AIO 12 |
| ADCINB3 | 31 | – | I | ADC B组，通道3输入 |
| ADCINB2<br>COMP1B<br>AIO10 | 30 | 24 | I<br>I<br>I/O | ADC B组，通道2输入<br>比较器输入1B<br>数字AIO 10 |
| ADCINB1 | 29 | 23 | I | ADC B组，通道1输入 |
| ADCINB0 | 28 | 22 | I | ADC B组，通道0输入 |
| $V_{REFLO}$ | 27 | 21 | | ADC外部低电平参考<br>**注意：$V_{REFLO}$始终连接到80脚封装的设备的$V_{SSA}$引脚上。** |

| 引脚名称 | 封装编号 | | I/O/Z[1] | 说明 |
|---|---|---|---|---|
| | QP100 | QP80 | | |
| **CPU和I/O电源** | | | | |
| $V_{DDA}$ | 25 | 20 | | 模拟电源引脚。近引脚接2.2μF电容（典型）到地。 |
| $V_{SSA}$ | 26 | 21 | | 模拟接地引脚。<br>**注意：$V_{REFLO}$始终连接到80脚封装的设备的$V_{SSA}$引脚上。** |
| $V_{DD}$ | 3 | 2 | | CPU和逻辑数字电源引脚。当使用内部VREG时，在每个VDD引脚和地之间放置一个1.2μF的电容。可以选用更高容值。 |
| | 14 | 12 | | |
| | 37 | 29 | | |
| | 63 | 51 | | |
| | 81 | 65 | | |
| | 91 | 72 | | |
| $V_{DDIO}$ | 5 | 4 | | 数字I/O缓冲器的电源引脚。VREG使能时，单电源供电。在每个引脚上放置一个解耦电容，确切的容值应由系统电压调节方案来确定。 |
| | 13 | 11 | | |
| | 38 | 30 | | |
| | 61 | 49 | | |
| | 79 | 63 | | |
| | 93 | 74 | | |
| | 46 | 37 | | |
| $V_{SS}$ | 4 | 3 | | 数字地引脚 |
| | 15 | 13 | | |
| | 36 | 28 | | |
| | 47 | 38 | | |
| | 62 | 50 | | |
| | 80 | 64 | | |
| | 92 | 73 | | |
| **稳压器及控制信号** | | | | |
| $\overline{VREGENZ}$ | 90 | 71 | I | 带有内部下拉的片内稳压器使能。直接连接$V_{SS}$(低)以使能内部1.8V稳压器(VREG)。直接连接$V_{DDIO}$(高)以禁用内部稳压器(VREG)，并使用外部1.8V供电。 |
| **通用输入输出及外设信号[2]** | | | | |
| *GPIO0* | 87 | 69 | I/O/Z | 通用输入/输出0 |
| EPWM1A | | | O | 增强的PWM1输出A和HRPWM通道 |
| Reserved | | | – | 预留 |
| Reserved | | | – | 预留 |
| *GPIO1* | 86 | 68 | I/O/Z | 通用输入/输出1 |
| EPWM1B | | | O | 增强型PWM1输出B |
| Reserved | | | – | 预留 |
| COMP1OUT | | | O | 比较器1的直接输出 |
| *GPIO2* | 84 | 67 | I/O/Z | 通用输入/输出2 |
| EPWM2A | | | O | 增强的PWM2输出A和HRPWM通道 |
| Reserved | | | – | 预留 |
| Reserved | | | – | 预留 |
| *GPIO3* | 83 | 66 | I/O/Z | 通用输入/输出3 |
| EPWM2B | | | O | 增强型PWM2输出B |
| SPISOMIA | | | I/O | SPI-A从机输出，主机输入 |
| COMP2OUT | | | O | 比较器2的直接输出 |
| *GPIO4* | 9 | 7 | I/O/Z | 通用输入/输出4 |
| EPWM3A | | | O | 增强的PWM3输出A和HRPWM通道 |
| Reserved | | | – | 预留 |
| Reserved | | | – | 预留 |
| *GPIO5* | 10 | 8 | I/O/Z | 通用输入/输出5 |
| EPWM3B | | | O | 增强的PWM3输出B |
| SPISIMOA | | | I/O | SPI-A从机输入，主机输出 |
| ECAP1 | | | I/O | 增强的捕获输入/输出1 |

| 引脚名称 | 封装编号 | | I/O/Z[1] | 说明 |
|---|---|---|---|---|
| | QP100 | QP80 | | |
| *GPIO6* | 58 | 46 | I/O/Z | 通用输入/输出6 |
| EPWM4A | | | O | 增强的PWM4输出A和HRPWM通道 |
| EPWMSYNCI | | | I | ePWM外部同步脉冲输入 |
| EPWMSYNCO | | | O | ePWM外部同步脉冲输出 |
| *GPIO7* | 57 | 45 | I/O/Z | 通用输入/输出7 |
| EPWM4B | | | O | 增强的PWM4输出B |
| SCIRXDA | | | I | SCI-A接收数据 |
| ECAP2 | | | I/O | 增强的捕获输入/输出2 |
| *GPIO8* | 54 | 43 | I/O/Z | 通用输入/输出8 |
| EPWM5A | | | O | 增强的PWM5输出A和HRPWM通道 |
| Reserved | | | – | 预留 |
| $\overline{\text{ADCSOCAO}}$ | | | O | ADC转换开始A |
| *GPIO9* | 49 | 39 | I/O/Z | 通用输入/输出9 |
| EPWM5B | | | O | 增强的PWM5输出B |
| SCITXDB | | | O | SCI-B发送数据 |
| ECAP3 | | | I/O | 增强的捕获输入/输出3 |
| *GPIO10* | 74 | 60 | I/O/Z | 通用输入/输出10 |
| EPWM6A | | | O | 增强的PWM6输出A和HRPWM通道 |
| Reserved | | | – | 预留 |
| $\overline{\text{ADCSOCBO}}$ | | | O | ADC转换开始B |
| *GPIO11* | 73 | 59 | I/O/Z | 通用输入/输出11 |
| EPWM6B | | | O | 增强的PWM6输出B |
| SCIRXDB | | | I | SCI-B接收数据 |
| ECAP1 | | | I/O | 增强的捕获输入/输出1 |
| *GPIO12* | 44 | 35 | I/O/Z | 通用输入/输出12 |
| $\overline{\text{TZ1}}$ | | | I | 可编程控制故障区输入1 |
| SCITXDA | | | O | SCI-A发送数据 |
| SPISIMOB | | | I/O | SPI-B从机输入，主机输出 |
| *GPIO13* | 95 | 75 | I/O/Z | 通用输入/输出13 |
| $\overline{\text{TZ2}}$ | | | I | 可编程控制故障区输入2 |
| Reserved | | | – | 预留 |
| SPISOMIB | | | I/O | SPI-B从机输出，主机输入 |
| *GPIO14* | 96 | 76 | I/O/Z | 通用输入/输出14 |
| $\overline{\text{TZ3}}$ | | | I | 可编程控制故障区输入3 |
| SCITXDB | | | O | SCI-B发送数据 |
| SPICLKB | | | I/O | SPI-B时钟输入输出 |
| *GPIO15* | 88 | 70 | I/O/Z | 通用输入/输出15 |
| ECAP2 | | | I/O | 增强的捕获输入/输出2 |
| SCIRXDB | | | I | SCI-B接收数据 |
| $\overline{\text{SPISTEB}}$ | | | I/O | SPI-B从机发送使能输入/输出 |
| *GPIO16* | 55 | 44 | I/O/Z | 通用输入/输出16 |
| SPISIMOA | | | I/O | SPI-A从机输入，主机输出 |
| Reserved | | | – | 预留 |
| $\overline{\text{TZ2}}$ | | | I | 可编程控制故障区输入2 |

| 引脚名称 | 封装编号 | | I/O/Z[1] | 说明 |
|---|---|---|---|---|
| | QP100 | QP80 | | |
| *GPIO17* | 52 | 42 | I/O/Z | 通用输入/输出17 |
| SPISOMIA | | | I/O | SPI-A从机输出，主机输入 |
| Reserved | | | – | 预留 |
| $\overline{TZ3}$ | | | I | 可编程控制故障区输入3 |
| *GPIO18* | 51 | 41 | I/O/Z | 通用输入/输出18 |
| SPICLKA | | | I/O | SPI-A时钟输入/输出 |
| SCITXDB | | | O | SCI-B发送数据 |
| XCLKOUT | | | O/Z | 取自SYSCLKOUT的输出时钟。XCLKOUT频率或与SYSCLKOUT相同，或为SYSCLKOUT频率的一半或者1/4，这是由XCLK寄存器中的位[1:0](XCLKOUTDIV)控制的。复位时，XCLKOUT=SYSCLKOUT/4。通过将XCLKOUTDIV设置为3，可以关闭XCLKOUT信号。GPIO18的IO复用控制也必须设置为XCLKOUT，以使该信号输出到引脚。 |
| *GPIO19* | 64 | 52 | I/O/Z | 通用输入/输出19 |
| XCLKIN | | | I | 外部振荡器输入。该引脚到时钟块的路径不经过引脚的复用门控。如果用于其他外设功能，必须注意不要启用此路径做时钟。 |
| $\overline{SPISTEA}$ | | | I/O | SPI-A从机发送使能输入/输出 |
| SCIRXDB | | | I | SCI-B接收数据 |
| ECAP1 | | | I/O | 增强的捕获输入/输出1 |
| *GPIO20* | 6 | 5 | I/O/Z | 通用输入/输出20 |
| EQEP1A | | | I | 增强的QEP1输入A |
| MDXA | | | O | McBSP-A发送串行数据 |
| COMP1OUT | | | O | 比较器1的直接输出 |
| *GPIO21* | 7 | 6 | I/O/Z | 通用输入/输出21 |
| EQEP1B | | | I | 增强型QEP1输入B |
| MDRA | | | I | McBSP接收串行数据 |
| COMP2OUT | | | O | 比较器2的直接输出 |
| *GPIO22* | 98 | 78 | I/O/Z | 通用输入/输出22 |
| EQEP1S | | | I/O | 增强的QEP1选通脉冲 |
| MCLKXA | | | I/O | McBSP-A发送时钟 |
| SCITXDB | | | O | SCI-B发送数据 |
| *GPIO23* | 2 | 1 | I/O/Z | 通用输入/输出23 |
| EQEP1I | | | I/O | 增强的QEP1索引 |
| MFSXA | | | I/O | McBSP-A发送帧同步 |
| SCIRXDB | | | I | SCI-B接收数据 |
| *GPIO24* | 97 | 77 | I/O/Z | 通用输入/输出24 |
| ECAP1 | | | I/O | 增强型CAP1的输入/输出 |
| EQEP2A | | | I | 增强型QEP2的输入A<br>**注意：eQEP2仅在100引脚封装中可用。** |
| SPISIMOB | | | I/O | SPI-B从机输入，主机输出 |
| *GPIO25* | 39 | 31 | I/O/Z | 通用输入/输出25 |
| ECAP2 | | | I/O | 增强型捕获输入/输出2 |
| EQEP2B | | | I | 增强型QEP2的输入B<br>**注意：eQEP2仅在100引脚封装中可用。** |
| SPISOMIB | | | I/O | SPI-B从机输出，主机输入 |

| 引脚名称 | 封装编号 | | I/O/Z[1] | 说明 |
|---|---|---|---|---|
| | QP100 | QP80 | | |
| *GPIO26* | 78 | 62 | I/O/Z | 通用输入/输出26 |
| ECAP3 | | | I/O | 增强型CAP3输入/输出 |
| EQEP2I | | | I/O | 增强型QEP2的索引<br>**注意：eQEP2仅在100脚封装中可用。** |
| SPICLKB | | | I/O | SPI-B时钟输入/输出 |
| USB0DP[3] | | | I/O | USB正差分信号。设置GPACTRL2寄存器的USBIOEN位，使能该引脚的USB功能。 |
| *GPIO27* | 77 | 61 | I/O/Z | 通用输入/输出27 |
| HRCAP2 | | | I | 高分辨率输入捕获2 |
| EQEP2S | | | I/O | 增强型QEP2选通信号<br>**注意：eQEP2仅在100脚封装中可用。** |
| $\overline{\text{SPISTEB}}$ | | | I/O | SPI-A从机输入/输出使能 |
| USB0DM[3] | | | I/O | USB负差分信号。设置GPACTRL2寄存器的USBIOEN位，使能该引脚的USB功能。 |
| *GPIO28* | 50 | 40 | I/O/Z | 通用输入/输出28 |
| SCIRXDA | | | I | SCI-A接收数据 |
| SDAA | | | I/OD | I2C数据，双向开漏端口 |
| $\overline{\text{TZ2}}$ | | | I | 可编程控制故障区输入2 |
| *GPIO29* | 43 | 34 | I/O/Z | 通用输入/输出29 |
| SCITXDA | | | O | SCIA发送数据 |
| SCLA | | | I/OD | I2C时钟，双向开漏端口 |
| $\overline{\text{TZ3}}$ | | | I | 可编程控制故障区输入3 |
| *GPIO30* | 41 | 33 | I/O/Z | 通用输入/输出30 |
| CANRXA | | | I | CAN接收 |
| EQEP2I | | | I/O | 增强型QEP2索引<br>**注意：eQEP2仅在100脚封装中可用。** |
| EPWM7A | | | O | 增强型PWM7输出A和HRPWM通道 |
| *GPIO31* | 40 | 32 | I/O/Z | 通用输入/输出31 |
| CANTXA | | | O | CAN发送 |
| EQEP2S | | | I/O | 增强型QEP2选通信号<br>**注意：eQEP2仅在100脚封装中可用。** |
| EPWM8A | | | O | 增强型PWM8输出A和HRPWM通道 |
| *GPIO32* | 99 | 79 | I/O/Z | 通用输入/输出32 |
| SDAA | | | I/OD | I2C数据，双向开漏端口 |
| EPWMSYNCI | | | I | 增强型PWM外部同步脉冲输入 |
| $\overline{\text{ADCSOCAO}}$ | | | O | ADC转换开始A |
| *GPIO33* | 100 | 80 | I/O/Z | 通用输入/输出33 |
| SCLA | | | I/OD | I2C时钟，双向开漏端口 |
| EPWMSYNCO | | | O | 增强型PWM外部同步脉冲输出 |
| $\overline{\text{ADCSOCBO}}$ | | | O | ADC转换开始B |
| *GPIO34* | 68 | 55 | I/O/Z | 通用输入/输出34 |
| COMP2OUT | | | O | 比较器2的直接输出 |
| Reserved | | | – | 预留 |
| COMP3OUT | | | O | 比较器3的直接输出 |

| 引脚名称 | 封装编号 | | I/O/Z[1] | 说明 |
|---|---|---|---|---|
| | QP100 | QP80 | | |
| *GPIO35* | 71 | 57 | I/O/Z | 通用输入/输出35 |
| TDI | | | I | 带内部上拉的JTAG测试数据输入(TDI)。TDI在TCK上升沿输入到选定的寄存器（指令或数据）中。 |
| Reserved | | | – | 预留 |
| Reserved | | | – | 预留 |
| Reserved | | | – | 预留 |
| *GPIO36* | 72 | 58 | I/O/Z | 通用输入/输出 36 |
| TMS | | | I | 带有内部上拉功能的JTAG测试模式选择(TMS)。该串行控制输入(TMS)在TCK上升沿输入到TAP控制器中。(↑) |
| Reserved | | | – | 预留 |
| Reserved | | | – | 预留 |
| Reserved | | | – | 预留 |
| *GPIO37* | 70 | 56 | I/O/Z | 通用输入/输出37 |
| TDO | | | O/Z | JTAG扫描输出，测试数据输出(TDO)。所选寄存器（指令或数据）的内容在TCK下降沿上从TDO移出。（8mA驱动能力） |
| Reserved | | | – | 预留 |
| Reserved | | | – | 预留 |
| Reserved | | | – | 预留 |
| *GPIO38* | 67 | 54 | I/O/Z | 通用输入/输出38 |
| XCLKIN | | | I | 外部振荡器输入。这个引脚到时钟块的路径不经过引脚的复用门控。如果用于其他外设功能，必须注意不要启用此路径做时钟。 |
| TCK | | | I | 带有内部上拉电阻的JTAG测试时钟。(↑) |
| Reserved | | | – | 预留 |
| Reserved | | | – | 预留 |
| Reserved | | | – | 预留 |
| *GPIO39* | 66 | 53 | I/O/Z | 通用输入/输出39 |
| Reserved | | | – | 预留 |
| Reserved | | | – | 预留 |
| Reserved | | | – | 预留 |
| *GPIO40* | 82 | – | I/O/Z | 通用输入/输出40 |
| EPWM7A | | | O | 增强型PWM7输出A和HRPWM通道 |
| SCITXDB | | | O | SCI-B发送数据 |
| Reserved | | | – | 预留 |
| *GPIO41* | 76 | – | I/O/Z | 通用输入/输出41 |
| EPWM7B | | | O | 增强型PWM7输出B |
| SCIRXDB | | | I | SCI-B接收数据 |
| Reserved | | | – | 预留 |
| *GPIO42* | 1 | – | I/O/Z | 通用输入/输出42 |
| EPWM8A | | | O | 增强型PWM8输出A和HRPWM通道 |
| $\overline{TZ1}$ | | | I | 可编程控制故障区输入1 |
| COMP1OUT | | | O | 比较器1的直接输出 |
| *GPIO43* | 8 | – | I/O/Z | 通用输入/输出43 |
| EPWM8B | | | O | 增强型PWM8输出B |
| $\overline{TZ2}$ | | | I | 可编程控制故障区输入2 |
| COMP2OUT | | | O | 比较器2的直接输出 |

| 引脚名称 | 封装编号 | | I/O/Z[1] | 说明 |
|---|---|---|---|---|
| | QP100 | QP80 | | |
| *GPIO44* | 56 | – | I/O/Z | 通用输入/输出44 |
| MFSRA | | | I/O | McBSP-A接收帧同步 |
| SCIRXDB | | | I | SCIB接收数据 |
| EPWM7B | | | O | 增强型PWM7输出B |
| *GPIO50* | 42 | – | I/O/Z | 通用输入/输出50 |
| EQEP1A | | | I | 增强型QEP1输入A |
| MDXA | | | O | McBSP发送串行数据 |
| $\overline{TZ1}$ | | | I | 可编程控制故障区输入1 |
| *GPIO51* | 48 | – | I/O/Z | 通用输入/输出51 |
| EQEP1B | | | I | 增强型QEP1输入B |
| MDRA | | | I | McBSP-A接收串行数据 |
| $\overline{TZ2}$ | | | I | 可编程控制故障区输入2 |
| *GPIO52* | 53 | – | I/O/Z | 通用输入/输出52 |
| EQEP1S | | | I/O | 增强的QEP1选通信号 |
| MCLKXA | | | I/O | McBSP-A发送时钟 |
| $\overline{TZ3}$ | | | I | 可编程控制故障区输入3 |
| *GPIO53* | 65 | – | I/O/Z | 通用输入/输出53 |
| EQEP1I | | | I/O | 增强的QEP1索引 |
| MFSXA | | | I/O | McBSP发送帧同步 |
| Reserved | | | – | 预留 |
| *GPIO54* | 69 | – | I/O/Z | 通用输入/输出54 |
| SPISIMOA | | | I/O | SPI-A从机输入，主机输出 |
| EQEP2A | | | I | 增强的QEP2输入A |
| HRCAP1 | | | I | 高分辨率输入捕获1 |
| *GPIO55* | 75 | – | I/O/Z | 通用输入/输出55 |
| SPISOMIA | | | I/O | SPI-A从机输入，主机输出 |
| EQEP2B | | | I | 增强的QEP2输入B |
| HRCAP2 | | | I | 高分辨率输入捕获2 |
| *GPIO56* | 85 | – | I/O/Z | 通用输入/输出56 |
| SPICLKA | | | I/O | SPI-A时钟输入/输出 |
| EQEP2I | | | I/O | 增强型QEP2索引 |
| HRCAP3 | | | I | 高分辨率输入捕获3 |
| *GPIO57* | 89 | – | I/O/Z | 通用输入/输出57 |
| $\overline{SPISTEA}$ | | | I/O | SPI-A从机发送使能输入/输出 |
| EQEP2S | | | I/O | 增强型QEP2选通信号 |
| HRCAP4 | | | I | 高分辨率输入捕获4 |
| *GPIO58* | 94 | – | I/O/Z | 通用输入/输出58 |
| MCLKRA | | | I/O | McBSP接收时钟 |
| SCITXDB | | | O | SCI-B发送数据 |
| EPWM7A | | | O | 增强型PWM7输出A和HRPWM通道 |

(1) I 是输入，O 是输出， Z 是高阻抗，OD 是开漏，↑是上拉，↓是下拉；

(2) GPIO 功能（斜体）复位时是默认值。它们下面列出的外设信号是备用功能。对于具有 GPIO 功能多路复用的 JTAG 引脚，到 GPIO 块的输入路径总是有效的。GPIO 模块的输出路径到 JTAG 模块路径的引脚根据 $\overline{TRST}$信号条件启用或禁用。请参考《AVP32F06X 技术参考手册》的“系统控制与中断”章节。

(3) 根据 USB 应用程序，可能需要额外的引脚来保持符合 USB2.0 规范。更多信息请参考《AVP32F06X 技术参考手册》的“通用串行总线(USB)控制器”章节。

# 4. 技术规格

## 4.1. 最大绝对额定值

超出下列最大额定值可能导致芯片的永久性损坏，长时间在最大额定值下工作也会影响芯片的可靠性。限值仅为应力额定值，并不表明器件在这些额定值下或者任何其他超过第 4.2 节所述条件下可正常工作。除非另有说明，所有电压值均以 $V_{SS}$ 为基准。

在自然通风条件下的工作温度范围内（除非另有注明）

| | | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|
| 数字电源电压 | $V_{DDIO}$ 相对于 $V_{SS}$ | –0.3 | 4.6 | V |
| | $V_{DD}$ 相对于 $V_{SS}$ | –0.3 | 2.5 | |
| 模拟电源电压 | $V_{DDA}$ 相对于 $V_{SSA}$ | –0.3 | 4.6 | V |
| 输入电压 | $V_{IN}$ (3.3 V) | –0.3 | 4.6 | V |
| | $V_{IN}$ (X1) | –0.3 | 2.5 | |
| 输出电压 | $V_O$ | –0.3 | 4.6 | V |
| 输入钳位电流 | 数字输入(每引脚), $I_{IK}$ ($V_{IN} < V_{SS}$ 或 $V_{IN} > V_{DDIO}$)(1) | –20 | 20 | mA |
| | 模拟输入(每引脚), $I_{IKANALOG}$(VIN < VSSA 或 VIN > VDDA) | –20 | 20 | |
| | 所有输入总计, $I_{IKTOTAL}$ ($V_{IN} < V_{SS}/V_{SSA}$ 或 $V_{IN} > V_{DDIO}/V_{DDA}$) | –20 | 20 | |
| 输出钳位电流 | $I_{OK}$ ($V_O < 0$ 或 $V_O > V_{DDIO}$) | –20 | 20 | mA |
| 结温(2) | $T_J$ | –40 | 150 | °C |
| 存储温度(2) | $T_{stg}$ | –65 | 150 | °C |

(1) 单个引脚的连续钳位电流为±2mA。

(2) 长期高温储存或在最高温度条件下长时间使用，可能会导致设备的整体寿命降低。

## 4.2. 建议运行条件

| | | 最小值 | 典型值 | 最大值 | 单位 |
|---|---|---|---|---|---|
| I/O供电电源, $V_{DDIO}$ | | 2.97 | 3.3 | 3.63 | V |
| 数字电源 （CPU，时钟模块），$V_{DD}$ (内部VREG禁止，1.8 V 外供) | | 1.71 | 1.8 | 1.995 | V |
| 数字参考地, $V_{SS}$ | | | 0 | | V |
| 模拟电源, $V_{DDA}$ | | 2.97 | 3.3 | 3.63 | V |
| 模拟参考地, $V_{SSA}$ | | | 0 | | V |
| 芯片工作频率(system clock) | | 2 | | 90 | MHz |
| 高电平输入电压, $V_{IH}$ (3.3 V) | | 2 | | $V_{DDIO}$ + 0.3 | V |
| 低电平输入电压, $V_{IL}$ (3.3 V) | | $V_{SS}$ – 0.3 | | 0.8 | V |
| 输出高电平拉电流, $V_{OH} = V_{OH(MIN)}$, $I_{OH}$ | 所有GPIO/AIO引脚 | | | -4 | mA |
| | 第2组(1) | | | -8 | |
| 输出低电平灌电流, $V_{OL} = V_{OL(MAX)}$, $I_{OL}$ | 所有GPIO/AIO引脚 | | | 4 | mA |
| | 第2组(1) | | | 8 | |
| 结温, $T_J$ | | –40 | | 150 | °C |
| 环境温度, $T_A$ | | –40 | | 125 | °C |

(1) 第二组引脚如下: GPIO16、GPIO17、GPIO18、GPIO19、GPIO28、GPIO29、GPIO36、GPIO37。

# 4.3. 流耗

## 4.3.1. AVP32F069 在 90 MHz SYSCLKOUT 运行条件下的流耗

| 模式 | 测试条件 | VREG使能 | | | | VREG禁止 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $I_{DDIO}$(1) | | $I_{DDA}$(2) | | $I_{DDIO}$(1) | | $I_{DDA}$(2) | | $I_{DD}$ | |
| | | 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 |
| Operational (Flash) | 打开外设时钟项:<br>ePWM1~8<br>eCAP1~3<br>eQEP1~2<br>eCAN<br>CLA<br>HRPWM<br>SCI-A，SCI-B<br>SPI-A，SPI-B<br>ADC<br>I2C<br>COMP1~3<br>CPU-TIMER0~2<br>McBSP<br>USB<br>所有PWM引脚在90 kHz翻转。<br>所有 I/O 引脚悬空(4) (6)<br>代码需在Flash外运行，保持为3种等待状态<br>XCLKOUT输出关闭 | 24.5mA | | 25.5mA | | 3.9mA | | 25.5mA | | 20.2mA | |
| IDLE | Flash关闭<br>XCLKOUT输出关闭<br>所有外设时钟关闭 | 9.6mA | | 64uA | | 3.5mA | | 60uA | | 5.8mA | |
| STANDBY | Flash关闭<br>所有外设时钟关闭 | 7.5mA | | 62uA | | 3.5mA | | 61.5uA | | 3.5mA | |
| HALT | Flash关闭<br>所有外设时钟关闭<br>输入时钟禁止(5) | 5.5mA | | 58uA | | 3.5mA | | 58uA | | 1.6mA (7) | |

(1) IDDIO 流耗与 I/O 引脚挂接负载电容有关。

(2) 为实现 IDDA 流耗在 IDLE、STANDBY 和 HALT 三种模式下测量，ADC 模块的时钟需要通过配置寄存器 PCLKCR0 关闭。

(3) 典型值指室温（25℃）与典型供电电源电压情况下所测流耗。

(4) 循环执行以下行为:

- SPI-A、SPI-B、SCI-A、eCAN-A、McBSP-A、I2C 端口持续输出数据。
- 运行硬件乘法。
- 看门狗工作。
- ADC 持续采样。
- COMP1 与 COMP2 持续翻转。
- GPIO17 持续翻转。

(5) 如果使用了晶体等谐振器件作为芯片时钟源，HALT 模式需关闭内部晶体振荡器电路。

(6) CLA 持续进行多项式运算。

(7) 为测量 HALT 模式下 $I_{DD}$ 流耗，需进行如下控制:

- PLL2 需通过调整 PLLCTL 寄存器 bit 2 位进行关闭。
- 地址为 0x6822 的寄存器 HRCAL 需要写 0x00FF。

注意

以上表内流耗数据代表同一 I/O 引脚上复用的外设功能，同时间只用其中之一，当同时打开多个复用外设功能时，芯片流耗将大于上表所示数据。

## 4.3.2. 降低流耗

除了采用 3 种低功耗模式，AVP32F06X 芯片可通过每个外设的时钟均可独立开启或关闭，进一步降低流耗。表 4-1 所示为通过关闭时钟所实现的流耗降低的典型值。

**表 4-1 通过关闭时钟所实现的流耗降低的典型值(系统时钟 90-MHz)**

| 外设模块(1)(3) | IDD流耗降低大小（mA） |
|---|---|
| ADC | 0.5(2) |
| I2C | 0.1 |
| ePWM | 0.4 |
| eCAP | 0.09 |
| eQEP | 0.3 |
| SCI | 0.15 |
| SPI | 0.2 |
| COMP/DAC | 0.05 |
| HRPWM | 0.3 |
| HRCAP | 0.15 |
| USB | 1.4 |
| CPU-TIMER | 0.3 |
| INTOSC1 | 0.2 |
| INTOSC2 | 0.3 |
| CAN | 0.25 |
| CLA | 0.35 |
| McBSP | 0.65 |

（1）所有外设时钟（除 CPU Timer 时钟）在复位时均关闭。对外设寄存器的读写操作需要在外设时钟打开后才能进行。

（2）此处数据仅代表 ADC 模块数字部分流耗降低，关闭 ADC 的时钟同时也会降低模拟供电电源 $I_{DDA}$ 流耗。

（3）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。比如，表中 ePWM 的流耗降低 0.4mA，仅代表关闭一个 ePWM 外设所降低的流耗大小。

进一步降低流耗的方法：

- 关闭配置为输出端口的上拉电阻可降低 VDDIO 电源轨流耗。
- 进一步关闭模拟模块的电源可进一步降低低功耗模式下 VDDA 电源轨流耗。请参考《AVP32F06X 技 术参考手册》中相应的模拟章节。
- 降低功耗可通过关闭 Flash 电源的方式，此时代码运行于 RAM 而非 Flash。

### 4.3.3. 流耗图（启用 VREG）

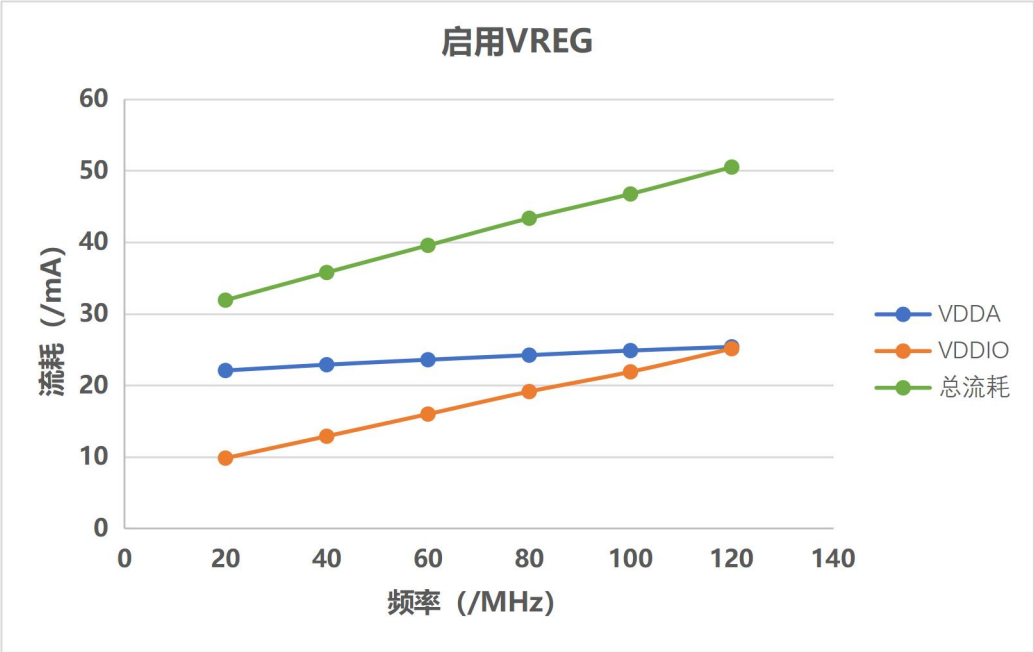



**图 4-1 典型运行电流（Flash）与频率（内部 VREG）间的关系**

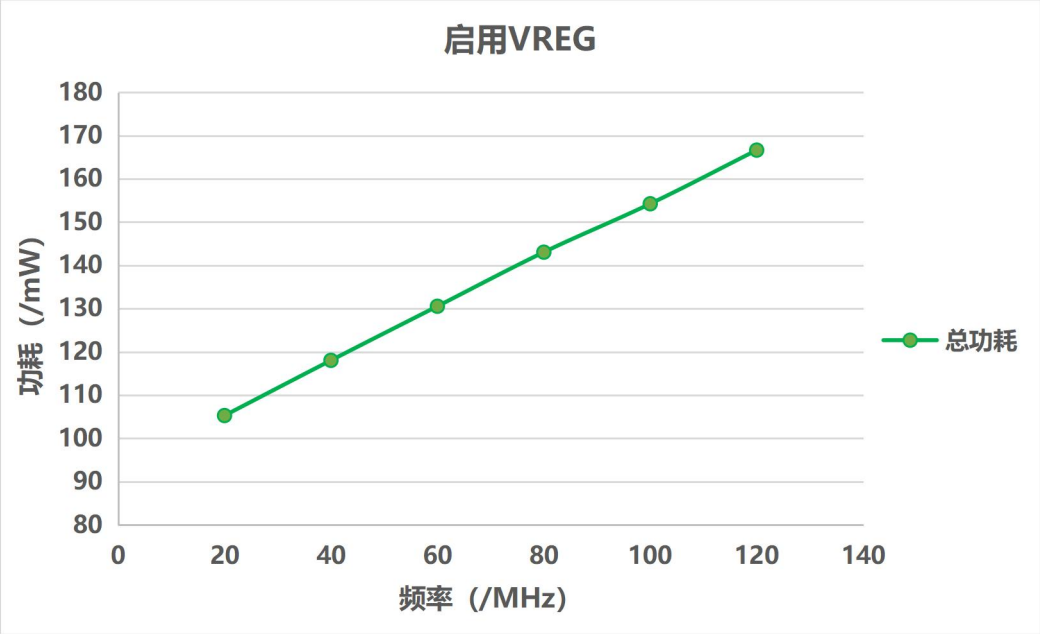



**图 4-2 典型运行功率与频率（内部 VREG）间的关系**

## 4.4. 电气特性

在建议的运行条件下（除非另有说明）(1)

| 参数项 | | 测试条件 | 最小值 | 典型值 | 最大值 | 单位 |
|---|---|---|---|---|---|---|
| $V_{OH}$ | 高电平输出电压 | $I_{OH}$ = $I_{OH}$ MAX | 2.4 | | | v |
| | | $I_{OH}$ = 50 μA | $V_{DDIO}$-0.2 | | | |
| $V_{OL}$ | 低电平输出电压 | $I_{OL}$ = $I_{OL}$ MAX | | | 0.4 | v |

| 参数项 | | 测试条件 | | 最小值 | 典型值 | 最大值 | 单位 |
|---|---|---|---|---|---|---|---|
| $I_{IL}$ 输入电流（低电平） | 上拉使能 | $V_{DDIO}$ = 3.3 V, $V_{IN}$ = 0 V | 所有GPIO | | –147 | | µA |
| | | | $\overline{XRS}$ 引脚 | | –567 | | |
| | 下拉使能 | $V_{DDIO}$ = 3.3 V, $V_{IN}$ = 0 V | | | | ±2 | µA |
| $I_{IH}$输入电流（高电平） | 上拉使能 | $V_{DDIO}$ = 3.3 V, $V_{IN}$ = $V_{DDIO}$ | | | | ±2 | µA |
| $I_{IH}$输入电流（高电平） | 下拉使能 | $V_{DDIO}$ = 3.3 V, $V_{IN}$ = $V_{DDIO}$ | | | 55 | | µA |
| $I_{OZ}$ | 输出电流，上/下拉关闭 | $V_O$ = $V_{DDIO}$ or 0 V | | | | ±2 | µA |
| | $V_{DDIO}$ BOR 触发电压 | $V_{DDIO}$下降 | | | 2.7 | | V |
| | $V_{DDIO}$ BOR 迟滞电压 | | | | 120 | | mV |
| | 电源监测释放后复位延续时间 | BOR/POR/OVR事件退出至$\overline{XRS}$释放 | | 400 | | 800 | µs |
| | VREG $V_{DD}$ 输出电压 | 内部 VREG 打开 | | | 1.9 | | V |

（1）当使用片载 VREG 时，它的输出由 POR/BOR 电路监控，当核心电压($V_{DD}$)超出范围时，该电路将复位设备。

# 4.5. 热阻特性

## 4.5.1. 100 引脚 QP 封装

| | | °C/W(1) | 气流(lfm)(2) |
|---|---|---|---|
| $R\Theta_{JC}$ | 结到壳热阻 | 14.05 | 0 |
| $R\Theta_{JB}$ | 结到板热阻 | 29.23 | 0 |
| $R\Theta_{JA}$ | 结到空气热阻 | 46.25 | 0 |
| | | 38.45 | 150 |
| | | 37.08 | 250 |
| | | 35.36 | 500 |
| $Psi_{JT}$ | 结到封装顶部 | 0.42 | 0 |
| | | 0.65 | 150 |
| | | 0.71 | 250 |
| | | 0.89 | 500 |
| $Psi_{JB}$ | 结到板 | 29.05 | 0 |
| | | 28.15 | 150 |
| | | 27.68 | 250 |
| | | 27.14 | 500 |

（1）这些值是基于 JEDEC 定义的 2S2P 系统（Theta JC [RΘJC] 值除外，该值基于 JEDEC 定义的 1S0P 系统），并将根据环境和应用而变化。有关更多信息，请参阅以下的 EIA/JEDEC 标准：

- JESD51-2，集成电路热测试方法环境条件 - 自然对流（静止空气）
- JESD51-3，导电性表面安装封装低有效热导率测试板
- JESD51-7，导电性表面安装封装高有效热导率测试板
- JESD51-9，用于区域阵列表面安装封装热测量的测试板

（2）lfm：每分钟线性英尺数

### 4.5.2. 80 引脚 QP 封装

| | | °C/W(1) | 气流(lfm)(2) |
|---|---|---|---|
| $R\Theta_{JC}$ | 结到壳热阻 | 15.30 | 0 |
| $R\Theta_{JB}$ | 结到板热阻 | 31.19 | 0 |
| $R\Theta_{JA}$ | 结到空气热阻 | 48.33 | 0 |
| | | 40.42 | 150 |
| | | 39.11 | 250 |
| | | 37.14 | 500 |
| $Psi_{JT}$ | 结到封装顶部 | 0.42 | 0 |
| | | 0.65 | 150 |
| | | 0.83 | 250 |
| | | 1.01 | 500 |
| $Psi_{JB}$ | 结到板 | 29.46 | 0 |
| | | 28.45 | 150 |
| | | 28.15 | 250 |
| | | 27.56 | 500 |

（1）这些值是基于 JEDEC 定义的 2S2P 系统（Theta JC [RΘJC] 值除外，该值基于 JEDEC 定义的 1S0P 系统），并将根据环境和应用而变化。有关更多信息，请参阅以下的 EIA/JEDEC 标准：

- JESD51-2，集成电路热测试方法环境条件 - 自然对流（静止空气）
- JESD51-3，导电性表面安装封装低有效热导率测试板
- JESD51-7，导电性表面安装封装高有效热导率测试板
- JESD51-9，用于区域阵列表面安装封装热测量的测试板

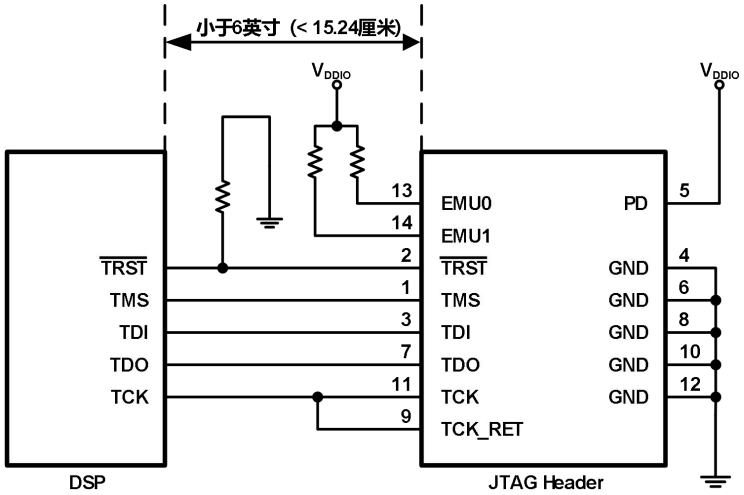
（2）lfm：每分钟线性英尺数

## 4.6. 热设计注意事项

根据终端应用程序的设计和操作配置，$I_{DD}$ 和 $I_{DDIO}$ 电流可能会发生变化。超过最终产品中推荐的最大功耗的系统可能需要额外的热增强。环境温度（$T_A$）随最终应用和产品设计而变化。影响可靠性和功能的关键因素是结温 $T_J$，而非环境温度。因此，应该注意将 $T_J$ 保持在指定的范围内。应测量 $T_{case}$ 以估计工作结温 $T_J$。通常在封装顶部表面的中心测量 $T_{case}$。

## 4.7. 针对DSP的无信号缓冲的仿真器连接

图 4-3 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 MCU 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 4-3 显示了较简单、无缓冲的情况。对于上拉/下拉电阻器的值，请见第 3.2 节。




注释：JTAG/GPIO 复用请见图 5-48。

**图 4-3 针对 DSP 的无信号缓冲的仿真器连接**

# 4.8. 参数信息

## 4.8.1. 时序参数符号

使用的时序参数符号是根据 JEDEC-100 标准而创建的。为缩短符号，某些引脚名称及相关术语缩写如下:

| 小写下标及其含义 | | 字母、符号及其含义: | |
|---|---|---|---|
| a | 存取时间 | H | 高 |
| c | 周期时间 (period) | L | 低 |
| d | 延迟时间 | V | 有效 |
| f | 下降时间 | X | 未知, 变化, 或无关紧要 |
| h | 维持时间 | Z | 高阻 |
| r | 上升时间 | | |
| su | 建立时间 | | |
| t | 转换时间 | | |
| v | 有效时间 | | |
| w | 脉冲宽度(width) | | |

## 4.8.2. 时序参数通用说明

AVP32F069 设备的所有输出信号（包括 XCLKOUT）均来自内部时钟，意味着输出信号在同一个半个周期内会以非常相近的时间点进行转换（相互之间具有最小的相位差），从而保持了良好的时序性能和一致性。

## 4.9. 测试负载电路

此测试负载电路用于测量本文档中提供的所有开关特性。

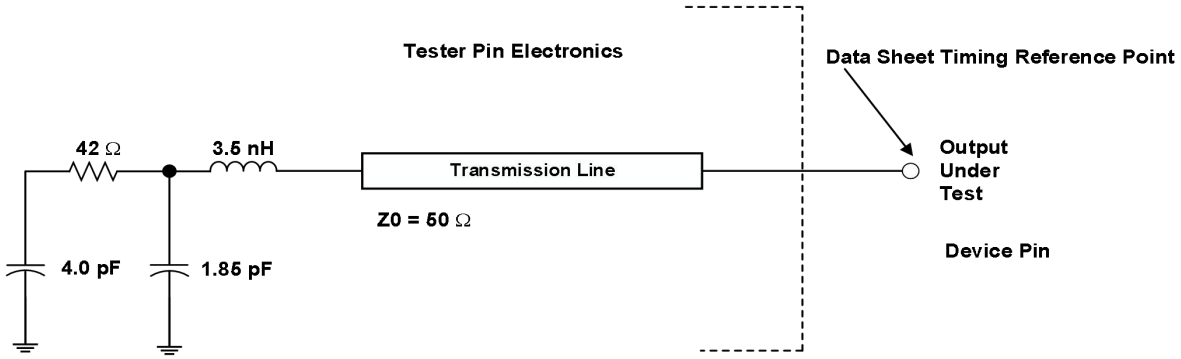
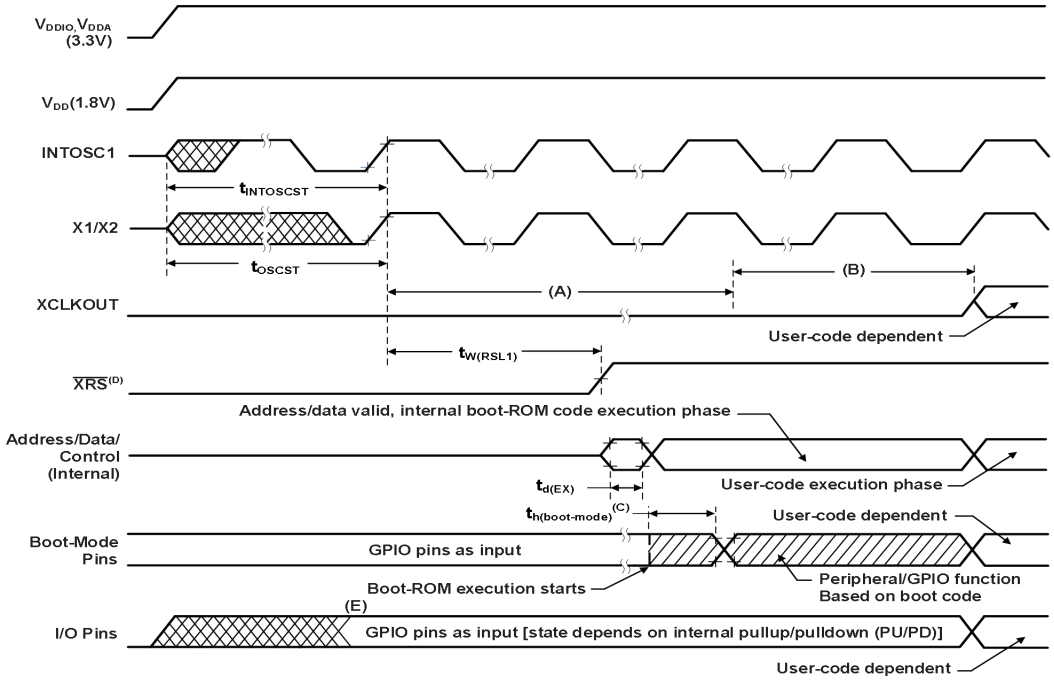



**图4-4 3.3V 测试负载电路**

## 4.10. 上电时序

在电源引脚的电源时序中，没有对其进行要求以确保所有模块的正确复位或防止 I/O 在上电或断电期间出现抖动（GPIO19、GPIO26-27、GPIO34-38 没有做上电防抖动处理）。在设备上电之前，任何数字引脚上（对于模拟引脚，该值为 $V_{DDA}$ 以上 0.7 V）都不应施加大于 $V_{DDIO}$ 以上二极管压降(0.7 V)的电压。施加在未上电器件引脚上的电压会以非法方式正偏内部 PN 节，并产生无法预料的结果。




A. 上电时，SYSCLKOUT 为 OSCCLK/4。因为 XCLK 寄存器中的 XCLKOUTDIV 位的默认状态为 0，所以 SYSCLKOUT 在 XCLKOUT 引脚输出之前还要再除以 4。此时 XCLKOUT = OSCCLK/16。

B. Boot ROM 配置 DIVSEL 位为 /1 动作。在此阶段 XCLKOUT = OSCCLK/4。在用户程序配置之前，XCLKOUT 在引脚上是不可见的。

C. 复位后，启动 ROM 代码采样启动模式引脚。基于 Boot Mode 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 代码在上电条件之后执行(在调试器环境中)，则引导代码执行时间基于当前 SYSCLKOUT 速度。SYSCLKOUT 将基于用户实际环境，可以启用或不启用 PLL。

D. 由于有片载 POR 电路，使用$\overline{XRS}$引脚是可选的。

E. 当 BOR 驱动为高电平时，内部的上拉或下拉会起作用。

**图4-5 上电复位**

## 4.10.1.复位（$\overline{XRS}$）时序要求

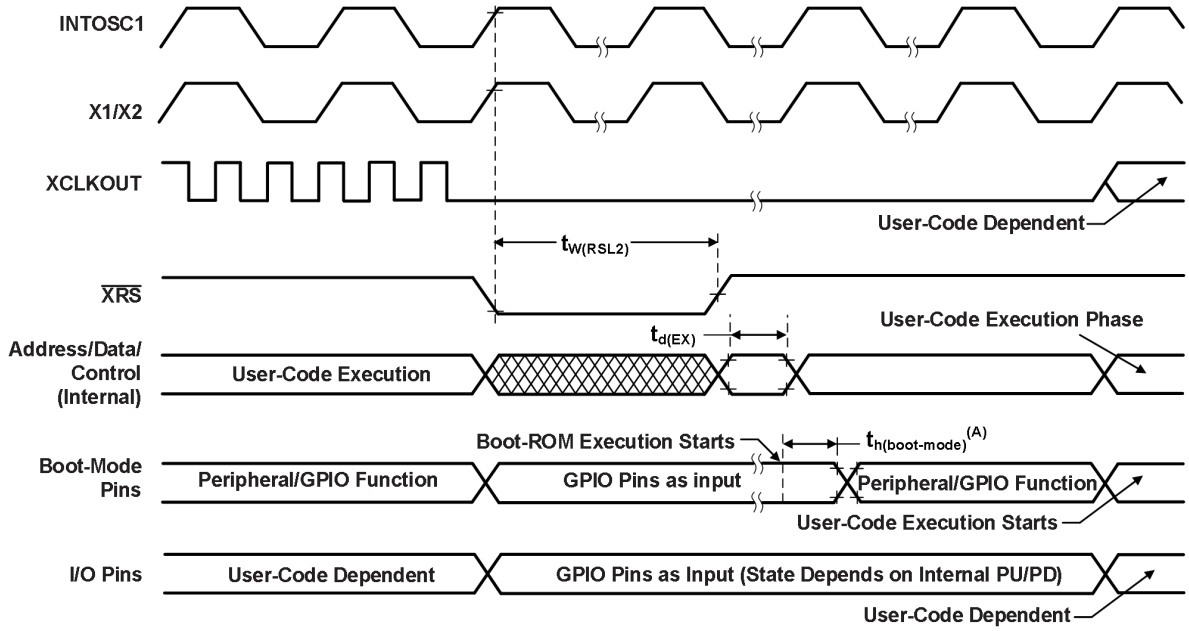
| | | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|
| $t_{h(boot-mode)}$ | 维持时间（引导模式引脚） | $1000t_{c(SCO)}$ | | cycles |
| $t_{w(RSL2)}$ | 热复位下$\overline{XRS}$低电平脉宽持续时间 | $32t_{c(OSCCLK)}$ | | cycles |

## 4.10.2.复位（$\overline{XRS}$）开关特性

在建议的运行条件下（除非另有说明）

| 参数 | | 最小值 | 典型值 | 最大值 | 单位 |
|---|---|---|---|---|---|
| $t_{w(RSL1)}$ | POR 电路驱动$\overline{XRS}$低电平脉宽持续时间 | | 600 | | μs |
| $t_{w(WDRS)}$ | 看门狗复位脉宽持续时间 | | $512t_{c(OSCCLK)}$ | | cycles |
| $t_{d(EX)}$ | $\overline{XRS}$上升沿到内部地址/数据总线有效的延迟时间 | | $32t_{c(OSCCLK)}$ | | cycles |
| $t_{INTOSCST}$ | 内部振荡器启动时间 | | 10 | | μs |
| $t_{OSCST}$ (1) | 片载晶体振荡器启动时间 | 1 | 10 | | ms |

**(1)** 取决于晶体/谐振器和电路板的设计。




A 复位后，Boot ROM 代码对 Boot Mode 引脚进行采样。基于 Boot Mode 引脚的状态，引导代码分支到目标内存或引导代码函数。如果在上电条件之后执行 Boot ROM 代码（在调试器环境中），则 Boot 代码执行时间基于当前 SYSCLKOUT 速度。SYSCLKOUT 则基于用户实际环境，可以选择启用或不启用 PLL。

**图 4-6 热复位**

图 4-7 所示为写入 PLLCR 寄存器所产生的结果示例。在第一阶段，PLLCR = 0x0004, SYSCLKOUT = OSCCLK × 2，然后将 0x0008 写入 PLLCR。就在写入 PLLCR 寄存器之后，PLL 锁定阶段开始。在此阶段，SYSCLKOUT = OSCCLK/2。锁相环锁定完成后，SYSCLKOUT 表示新的工作频率 OSCCLK × 4。

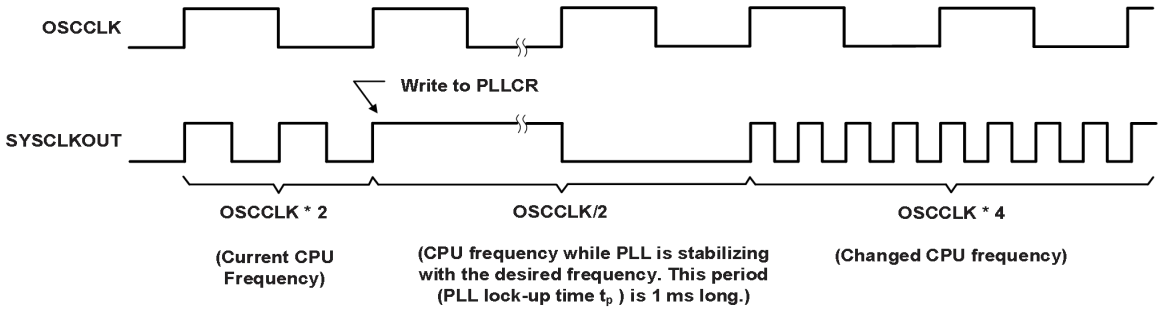



**图 4-7 写入 PLLCR 寄存器所产生的结果示例**

# 4.11. 时钟特性

## 4.11.1. 器件时钟表

本节介绍 AVP32F069 微系统控制单元上多种时钟选项的时序要求和开关特性。第 4.12.1.1 节列出了多种时钟周期。

### 4.11.1.1. AVP32F069 时钟表和术语（90-MHz 器件）

| | | 最小值 | 典型值 | 最大值 | 单位 |
|---|---|---|---|---|---|
| SYSCLKOUT | $t_{c(SCO)}$，系统周期时间 | 11.11 | | 500 | ns |
| | 频率 | 2 | | 90 | MHz |
| LSPCLK(1) | $t_{c(LCO)}$，低速时钟周期 | 11.11 | 44.4(2) | | ns |
| | 频率 | | 22.5(2) | 90 | MHz |
| ADC 时钟 | $t_{c(ADCCLK)}$，ADC时钟周期 | 22.22 | | | ns |
| | 频率 | | | 45 | MHz |

(1) 更低的时钟 LSPCLK 将降低器件功耗。
(2) 当系统时钟为 90MHz 时，为默认的复位值。

### 4.11.1.2. 器件时钟的要求/特性

| | | 最小值 | 典型值 | 最大值 | 单位 |
|---|---|---|---|---|---|
| 片载振荡器 (X1/X2 引脚) (晶振/谐振器) | $t_{c(OSC)}$，系统周期时间 | 50 | | 200 | ns |
| | 频率 | 5 | | 20 | MHz |
| 外部振荡器/时钟源 (XCLKIN 引脚) — PLL 使能 | $t_{c(CI)}$，时钟周期 (C8) | 33.3 | | 200 | ns |
| | 频率 | 5 | | 30 | MHz |
| 外部振荡器/时钟源 (XCLKIN 引脚) — PLL 禁用 | $t_{c(CI)}$，时钟周期 (C8) | 11.11 | | 250 | ns |
| | 频率 | 4 | | 90 | MHz |
| 跛行模式 SYSCLKOUT (/2 分频使能) | 频率范围 | | 1 至 5 | | MHz |
| XCLKOUT | $t_{c(XCO)}$，周期时间 (C1) | 44.44 | | 2000 | ns |

| | | | | | |
|---|---|---|---|---|---|
| | 频率 | 0.5 | | 22.5 | MHz |
| PLL 锁定时间(1) | $t_p$ | | | 1 | ms |

(1) PLLLOCKPRD 寄存器值必须是 OSCCLK 周期的整数倍。如果使用内部振荡器（10 MHz）作为时钟源，则 PLLLOCKPRD 寄存器必须写入 10，000（最小值）。

### 4.11.1.3. 内部 OSC (INTOSC1/INTOSC2) 特性

| 参数项 | | 最小值 | 典型值 | 最大值 | 单位 |
|---|---|---|---|---|---|
| 内部OSC1与OSC2，在 30°C情况下 (1) (2) | 频率 | | 10.000 | | MHz |
| 内部OSC1与OSC2，在 30°C情况下 (1) (2) | 频率 | | 10.000 | | MHz |
| 步长 (粗调) | | | 70 | | kHz |
| 步长 (微调) | | | 15 | | kHz |
| 温漂(3) | | -2 | | | kHz/°C |
| 电压（$V_{DD}$）(3) | | | 175 | | Hz/mV |

(1) 振荡器频率会随温度而变化，需补偿振荡器温漂。

(2) 仅当启用 VREG（$\overline{VREGENZ}$ = $V_{SS}$）时，才能确保频率范围。

(3) 内部振荡器的输出频率与温度和电压（$V_{DD}$）相关。例如：

- 温度升高将导致输出频率随着温度的增加而减小。
- 电压（$V_{DD}$）的降低将导致输出频率随着电压的降低而降低。

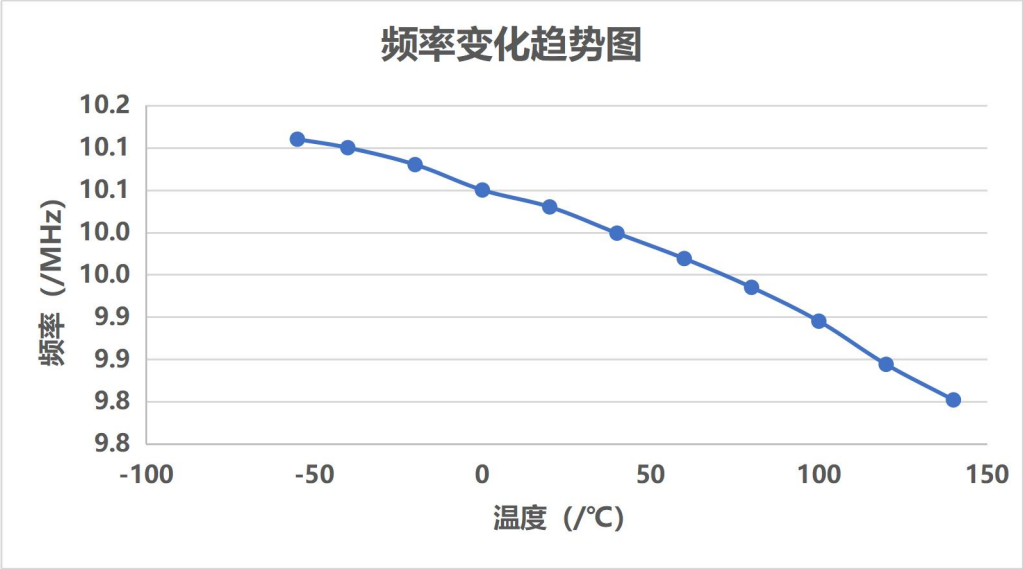



**图 4-8 内部 OSC 频率随温度的变化曲线图**

## 4.11.2. 时钟要求和特性

### 4.11.2.1. XCLKIN 时序要求 – 使能 PLL

| NO. | | | 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 | $t_{w(CIL)}$ | XCLKIN低电平脉宽占空比 $t_{c(OSCCLK)}$ | 45% | 55% | |
| C12 | $t_{w(CIH)}$ | XCLKIN高电平脉宽占空比 $t_{c(OSCCLK)}$ | 45% | 55% | |

### 4.11.2.2. XCLKIN 时序要求 – 禁用 PLL

| NO. | | | | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|---|---|
| C9 | $t_{f(CI)}$ | 下降时间，XCLKIN | 高达 20 MHz | | 6 | ns |
| | | | 20 MHz 至 90 MHz | | 2 | ns |
| C10 | $t_{r(CI)}$ | 上升时间，XCLKIN | 高达 20 MHz | | 6 | |
| | | | 20 MHz 至 90 MHz | | 2 | |
| C11 | $t_{w(CIL)}$ | XCLKIN低电平脉宽占空比 $t_{c(OSCCLK)}$ | | 45% | 55% | |
| C12 | $t_{w(CIH)}$ | XCLKIN高电平脉宽占空比 $t_{c(OSCCLK)}$ | | 45% | 55% | |

可能的配置模式如表 5-15 所示。

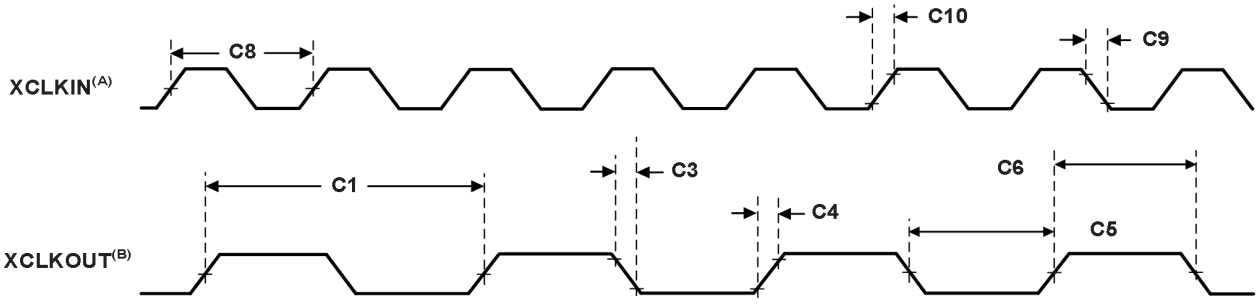
### 4.11.2.3. XCLKOUT 开关特性 (PLL 旁路或使能)

在建议运行条件下测得 (除非另有说明)(1) (2)

| NO. | PARAMETER | | MIN | MAX | UNIT |
|---|---|---|---|---|---|
| C3 | $t_{f(XCO)}$ | XCLKOUT下降时间 | | 5 | ns |
| C4 | $t_{r(XCO)}$ | XCLKOUT上升时间 | | 5 | ns |
| C5 | $t_{w(XCOL)}$ | XCLKOUT 低电平脉冲持续时间 | H – 2 | H + 2 | ns |
| C6 | $t_{w(XCOH)}$ | XCLKOUT高电平脉冲持续时间 | H – 2 | H + 2 | ns |

(1) 假定这些参数是在 40 pF 负载的情形下。

(2) $H = 0.5t_{c(XCO)}$




A. XCLKIN 与 XCLKOUT 的关系取决于所选的分频因子。所示波形关系仅用于说明时序参数，可能因实际配置而不同。.

B. XCLKOUT 可配置为 SYSCLKOUT 的比例输出。

**图 4-9 时钟时序图**

# 4.12. 闪存时序

## 4.12.1. 闪存/OTP 耐久度

| | | 擦除/编程温度(1) | 最小值 | 典型值 | 最大值 | 单位 |
|---|---|---|---|---|---|---|
| $N_f$ | 闪存对于阵列的耐久度 (写入/擦除周期) | -40°C 至125 °C (环境温度) | 100000 | | | 周期 |
| $N_{OTP}$ | OTP 对于阵列的耐久度 (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，可能会影响耐久次数。

## 4.12.2. 90-MHz SYSCLKOUT 的闪存参数

| 参数 | | 测试条件 | 最小值 | 典型值 | 最大值 | 单位 |
|---|---|---|---|---|---|---|
| 编程时间(1) | 16位字 | | | 32 | | μs |
| | 16K 扇区 | | | 360 | | ms |
| | 8K 扇区 | | | 180 | | ms |
| | 4K 扇区 | | | 90 | | ms |
| 擦除时间(2) | 16K 扇区 | | | 319 | | ms |
| $I_{DDP}$ (3) | 擦除/编程周期中的$V_{DD}$ 流耗 | 禁用VREG | | 16 | | mA |
| $I_{DDIOP}$ (3) | 擦除/编程周期中的$V_{DDIO}$流耗 | | | 16 | | mA |
| $I_{DDIOP}$ (3) | 擦除/编程周期中的$V_{DDIO}$流耗 | 使能VREG | | 41 | | mA |

(1) 编程时间为最大器件频率。此表中指示的编程时间仅适用于器件 RAM 中的所有所需代码/数据都可用并准备好进行编程的情况。编程时间包括闪存状态机的开销，但不包括将以下内容传输到 RAM 的时间：

- 使用闪存 API 对闪存进行编程的代码
- 闪存 API 本身
- 要编程的闪存数据

(2) 当器件出厂时，片载闪存处于擦除状态。因此，首次对器件进行编程时，编程之前无需擦除闪存。但是，所有后续编程操作都需要执行擦除操作。

(3) 室温下包括函数调用开销在内的典型参数，是在所有外设关闭时的参数。在整个闪存编程过程中保持稳定的电源很重要。可以设想，闪存编程期间的器件电流消耗可能高于正常工作条件下的流耗，所使用的电源应确保 $V_{MIN}$ 始终处于供电轨上，如数据手册中 “建议运行条件” 章节所述。在擦除/编程期间，任何断电或电源中断都可能损坏密码位置并永久锁定设备。不建议通过 USB 端口为目标板供电（在闪存编程期间），因为该端口可能无法响应编程过程中的电源需求。

## 4.12.3. Flash/OTP 访问时序

| 参数 | 最小值 | 最大值 | 单位 |
|---|---|---|---|
| $t_{a(fp)}$ 页面闪存访问时间 | 36 | | ns |
| $t_{a(fr)}$ 随机闪存访问时间 | 36 | | ns |
| $t_{a(OTP)}$ OTP访问时间 | 60 | | ns |

## 4.12.4. Flash 数据保持时间

| 参数 | | 测试条件 | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|---|
| $t_{retention}$ | 数据保持时间 | $T_J$ = 55°C | 10 | | 年 |

表 4-2 不同频率下所需最小的闪存/OTP 等待状态

| SYSCLKOUT (MHz) | SYSCLKOUT (ns) | 闪存页等待状态(1) | 随机等待状态(1) | OTP等待状态 |
|---|---|---|---|---|
| 90 | 11.11 | 3 | 3 | 5 |
| 80 | 12.5 | 2 | 2 | 4 |
| 70 | 14.29 | 2 | 2 | 4 |
| 60 | 16.67 | 2 | 2 | 3 |
| 55 | 18.18 | 1 | 1 | 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 |

(1) 页面和随机等待状态必须大于或等于 1。

闪存页面和随机等待状态的计算公式如下：

闪存页面等待状态 $=[(\frac{t_{a(f\cdot p)}}{t_{c(SCO)}})\text{-}1]$ 向上舍入到下一个最大整数或 1，以较大者为准。

闪存随机等待状态 $=[(\frac{t_{a(f\cdot r)}}{t_{c(SCO)}})\text{-}1]$ 向上舍入到下一个最大整数或 1，以较大者为准。

OTP 等待状态的计算公式如下：

OTP 等待状态 $=[(\frac{t_{a(OTP)}}{t_{c(SCO)}})\text{-}1]$ 向上舍入到下一个最大整数或 1，以较大者为准

## 4.13. ESD等级

| | | | 值 | 单位 |
|---|---|---|---|---|
| 100 引脚 LQFP 封装 | | | | |
| $V_{(ESD)}$ 静电放电 | 人体放电模型 (HBM)，符合 AEC Q100-002(1) | 所有引脚 | ±2000 | V |
| | 充电器件模型 (CDM)，符合 AEC Q100-011 | 所有引脚 | ±500 | |
| | | 100 引脚 LQFP 上的转角引脚：1、25、26、50、51、75、76、100 | ±750 | |
| 80 引脚 LQFP 封装 | | | | |
| $V_{(ESD)}$ 静电放电 | 人体放电模型 (HBM)，符合 AEC Q100-002(1) | 所有引脚 | ±2000 | V |
| | 充电器件模型 (CDM)，符合 AEC Q100-011 | 所有引脚 | ±500 | |
| | | 80 引脚 LQFP 上的转角引脚：1、20、21、40、41、60、61、80 | ±750 | |

(1) AEC Q100-002 指示应当按照 ANSI/ESDA/JEDEC JS-001 规范执行 HBM 应力测试。

# 5. 详细描述

## 5.1. 概述

### 5.1.1. CPU

AVP32F069 是 AVP32 浮点 DSP 平台的成员之一，采用与现有 AVP32 系列产品相同的 32 位浮点架构。每个 AVP32x 和 ADP32x 的控制器，包括 AVP32F069 产品，都是非常高效的 C/C++引擎，使用户不仅能够用高级语言开发其系统控制软件，而且还能够使用 C/C++开发数学算法。该设备在单片机的数学任务上的处理效率，与在通常由微控制器设备处理的系统控制任务上的效率一样高。该高效处理能力使得许多系统中无需第二个处理器。32×32 位 MAC 64 位处理能力使控制器能够有效地处理更高的数值分辨率问题。再加上快速中断响应和自动保存关键寄存器的上下文，从而使设备能够以最小的延迟来处理多个异步事件。该设备有一个 8 级深受保护的流水线，具有流水线式内存访问。该流水线式操作使器件能够高速执行，而无需求助于昂贵的高速内存。特别分支超前硬件显著减少了条件不连续而带来的延迟。特殊存储条件操作进一步提升了性能。

### 5.1.2. 控制环加速器（CLA）

CLA 是一个单精度（32 位）浮点协处理单元，通过并行处理方式，扩展了 AVP32 的 CPU 处理能力。CLA 是一个独立的处理器，具有自己的总线结构、取指机制和流水线。可以指定 8 个独立的 CLA 任务或例程。每个任务均可由软件或外围设备启动，如 ADC、ePWM、eCAP、eQEP 或 CPU-Timer0。CLA 每次只执行一项任务。当一个任务完成时，主 CPU 会受到 PIE 的中断通知，CLA 会自动开始下一个最高优先级的待处理任务。CLA 可以直接访问 ADC 结果寄存器、ePWM、HRPWM、eCAP 和 eQEP 寄存器。专用消息 RAM 提供在主 CPU 和 CLA 之间传递额外数据的方法。

### 5.1.3. 维特比，复杂数学，CRC 单元（VCU）

VCU 通过添加针对复杂数学、维特比解码和 CRC 计算的附加汇编指令，提升了器件的处理能力。VCU 指令提高了许多应用的处理速度：

- 用于电力线载波PRIME和G3标准的正交频分复用标准(OFDM)
- 短程雷达复杂数学计算
- 功率计算
- 存储器和数据通信数据包检查(CRC)

VCU 功能包括：

- 支持循环冗余检查（CRC）的指令，这是一个多项式代码校验和。
  - –CRC8
  - –CRC16
  - –CRC32
- 支持维特比解码器的灵活软件实现的指令
  - –对于1/2或1/3的代码率的分支度量计算

–每个蝶型结构的 5 周期内的相加-比较选择或者 Viterbi 蝶型结构

–每阶段有3周期内的追溯

–轻松支持 PRIME 和 G3 标准中的 K=7 的限制长度

- 复杂数学算术单元
  - –单个循环加法或减法
  - –2周期乘法
  - –2周期乘法累积(MAC)
  - –单周期重复MAC
- 独立寄存器空间

### 5.1.4. 内存总线（哈弗总线架构）

与许多 MCU 类型的设备一样，多总线用于在存储器和外设以及 CPU 之间传输数据。内存总线架构包含一个程序读总线、数据读总线和数据写总线。程序读取总线由 22 条地址线和 32 条数据线组成。数据读写总线分别由 32 条地址线和 32 条数据线组成。32 位宽的数据总线支持单个周期的 32 位操作。多总线架构，通常称为哈佛总线，使内核能够在一个周期内获取指令、读取数据值和写入数据值。连接到内存总线的所有外设和内存都会优先考虑内存访问。通常，内存总线优先级可以概括如下：

| | | |
|---|---|---|
| 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操作） |
| | 程序写入 | （内存总线上不能同时进行数据和程序写入操作） |
| | 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） |

### 5.1.5. 外设总线

为了使外设能够在 ADP32 和 AVP32 系列器件之间迁移，这些器件采用外设总线标准进行外设互连。外设总线桥复用了多种总线，此总线将处理器内存总线组装进一个由 16 条地址线、16 或 32 条数据线和相关控制信号组成的单总线中。支持两个版本的外设总线，一个版本仅支持 16 位访问（称为外设帧 2），另外一个版本支持 16 位和 32 位访问（称为外设帧 1）。

### 5.1.6. 实时 JTAG 和分析

该设备实现了标准的 IEEE 1149.1（IEEE 标准 1149.1-1990 标准测试访问端口和边界扫描架构）JTAG 接口进行电路调试。此外，该设备支持实时操作模式，允许在处理器运行、执行代码和服务中断时修改内存、外设和寄存器位置的内容。用户还可以通过非时间关键代码，同时在不受干扰的情况下实现对时间关键中断的服务。该设备在 CPU 内的硬件中实现了实时模式，无需软件监测器。此外，还提供了特殊的分析硬件，允许设置硬件断点或数据/地址监测点，并在发生匹配时生成各种用户可选择的中断事件。

### 5.1.7. Flash

AVP32F069 包含 128K × 16 的嵌入式 Flash 内存，分成 8 个 16K × 16 扇区。还包含一个 1K × 16 的 OTP 内存，地址范围为 0x3D7800 到 0x3D7BF9。用户可以单独删除、编程和验证 Flash 扇区，同时保留其他扇区。但是，不支持在 Flash 或 OTP 的一个扇区内执行擦除或编程其他扇区的 Flash 算法。提供了特殊的内存流水线，使 Flash 模块能够实现更高的性能。Flash/OTP 被映射到程序和数据空间；因此，它可以用于执行代码或存储数据信息。地址 0x3F7FF0 到 0x3F7FF5 是为数据变量保留的，不应包含程序代码。

**注　意**

应用程序可以配置 Flash 和 OTP 等待状态，这使得运行在较低频率上的应用能够将闪存配置为使用较少的等待状态。

可以通过在闪存选项寄存器中启用闪存流水线操作模式来提升闪存的性能。这个模式被启用时，线性代码执行的性能将远快于只由等待状态配置所表现的性能。 使用闪存流水线模式时的确切性能增益取决于应用程序。

有关闪存选项、闪存等待状态和 OTP 等待状态寄存器的更多信息，请参阅《AVP32F06X 技术参考手册》的“系统控制和中断”章节。

### 5.1.8. M0/M1 SARAM

所有设备都包含这两个单存取内存块，每个内存大小为 1K × 16。重置时，堆栈指针指向 M1 块的开始地址。M0 和 M1 块被映射到程序和数据空间。因此，用户可以使用 M0 和 M1 来执行代码或数据变量。该分区将在链接器中执行。设备向程序员提供了一个统一的内存映射，这使得在高级语言中进行编程变得更加容易。

### 5.1.9. L4 SARAM 及 L0/L1/L2/L3/L5/L7/L8 DPSARAM

器件有多达 48K × 16 单周期访问 RAM。详细内存大小见第 5.2 节中的“内存映射”图。内存块可作为程序和数据空间。L0 为 2K 大小，L1 和 L2 都是 1K 大小，L3 为 4K 大小。L4、L5、L6、L7 和 L8 都是 8K 大小。L0、L1 和 L2 被 CLA 用作数据空间，L3 可被 CLA 用作程序空间，L5、L6、L7 和 L8 可被 DMA 用作数据空间。DPSARAM 参考双端口配置。

### 5.1.10. 引导 ROM

引导 ROM 由厂家使用引导加载软件进行设定。提供的引导模式信号告诉引导加载软件在加电时使用哪种引导模式。用户能够选择正常引导或者从一个外部连接下载新软件或者选择在内部闪存/ROM 中编辑的引导软件。引导 ROM 还包含用于数学相关算法的标准表，例如 SIN/COS 波形。

**表 5-1 引导模式选择**

| 模式 | GPIO37/TDO | GPIO34/COMP2OUT / COMP3OUT | $\overline{\text{TRST}}$ | 模式 |
|---|---|---|---|---|
| 3 | 1 | 1 | 0 | GetMode引导 |
| 2 | 1 | 0 | 0 | 等待模式 (详见第5.1.11节) |
| 1 | 0 | 1 | 0 | SCI引导 |
| 0 | 0 | 0 | 0 | 并行IO引导 |
| EMU | x | x | 1 | 仿真引导 |

#### 5.1.10.1. 仿真引导

当连接 Debug 引脚后，GPIO37/TDO 引脚不能用作引导模式选择。这种情况下，引导 ROM 检测 Debug 引脚连接后利用 PIE 向量表中 SARAM 保留位置中的内容确定引导模式。如果保留位置中为无效内容，则进入等待引导。Debug 引导可以访问所有引导模式。

#### 5.1.10.2. GetMode 引导

GetMode 默认选择为引导到 Flash。可以通过编程 OTP 中的两个地址内容来选择其他引导模式。如果 OTP 中该位置内容无效，则引导到 Flash。可以选择以下引导模式之一：SCI、SPI、I2C、CAN 或 OTP。

#### 5.1.10.3. 引导程序使用的外设引脚

表 5-2 显示了外设引导程序所使用的引脚。如果这些外设引脚的使用与用户的应用有冲突，请参考 GPIO 选择列表。

**表 5-2 引导程序外设引脚**

| 引导程序 | 外设引脚 |
|---|---|
| SCI | SCIRXDA (GPIO28)<br>SCITXDA (GPIO29) |
| 并行IO引导 | 数据 (GPIO31,30,5:0)<br>DSP控制(AIO6)<br>主控制(AIO12) |
| SPI | SPISIMOA (GPIO16)<br>SPISOMIA (GPIO17)<br>SPICLKA (GPIO18)<br>$\overline{\text{SPISTEA}}$(GPIO19) |
| II2C | SDAA (GPIO28)<br>SCLA (GPIO29) |
| CAN | CANRXA (GPIO30)<br>CANTXA (GPIO31) |

### 5.1.11. 安全性

该器件支持高级别的安全性，以保护用户固件不被逆向工程。安全功能具有 128 位密码（硬编码为 16 个等待状态），用户将其编程到闪存中。一个代码安全模块（CSM）用于保护闪存/OTP 和 L0/L1SRAM 模块。安全功能可防止未经授权的用户通过 JTAG 端口检查内存内容，或尝试引导加载一些会导出安全内存内容的不良软件。要启用对安全块的访问，用户必须写入正确的 128 位 KEY 值，该值与闪存中密码位置中存储的值相匹配。

除了 CSM 之外，还实施了调试代码安全逻辑 (ECSL)，以防止未经授权的用户单步执行安全代码。连接调试探针时对 CSM 安全存储器的任何代码或数据访问都会触发 ECSL 并中断调试探针连接。为了允许调试安全代码，同时保持 CSM 对安全存储器读取的保护，用户必须将正确的值写入 KEY 寄存器 (KEY0–KEY3) 的低 64 位，该值与存储在低 64 位中的值相匹配。闪存内的密码位置。仍然必须执行对闪存中所有 128 位密码的伪读取。如果密码位置 (PWL0 – PWL3) 的低 64 位全为 1（未编程），则 KEY 值不需要匹配。在调试安全代码期间，可以进行单步操作等操作。但是，在 编辑工具窗口中无法看到安全存储器的实际内容。

当连接到 JTAG 调试探针的安全设备通电时，CPU 将开始执行并可能执行对受保护区域进行访问的指令。如果发生这种情况，ECSL 将跳闸并导致 JTAG 电路停用。在这种情况下，主机将无法与设备建立连接。

**注意**

- 编程代码安全密码时，从 0x3F7F80 到 0x3F7FF5 的所有地址都不能用作程序代码或数据。这些位置必须编程为 0x0000。
- .如果将来可能需要通过 JTAG 对安全设备进行重新编程，那么设计电路板的方式非常重要，即设备可以在上电时进入等待启动模式（当需要重新编程时）。否则，如前所述，ECSL 可能会停用 JTAG 电路并阻止与设备的连接。如果在现场将设备重新配置为等待启动模式不切实际，则必须在固件中实施某种机制来检测何时需要进行固件更新。然后代码可以分支到 bootROM 中所需的引导加载程序。它还可以分支到等待引导模式，此时可以连接 JTAG 调试探针，设备不安全并通过 JTAG 本身完成编程。
- 如果不使用代码安全功能，地址 0x3F 7F80 到 0x3F 7FEF 可用于代码或数据。地址 0x3F 7FF0 到 0x3F 7FF5 为数据保留，不应包含程序代码。
- 128 位密码（在 0x3F 7FF8 到 0x3F 7FFF）不得编程为零，这样做会永久锁定设备。

**代码安全模块免责声明**

本器件上包含的代码安全模块（CSM）旨在密码保护存储在相关内存（ROM 或 FLASH）中的数据，并由进芯电子根据其标准条款和条件，符合适用的保修规范。

然而，进芯电子不保证或声明 CSM 不能被破坏或破坏存储在相关内存中的数据不能通过其他方式访问。此外，除上文规定外，进芯电子对 CSM 或本器件的操作不作任何保证或陈述，包括任何对适销性或适合于特定用途的隐含保证。

在任何情况下，进芯电子均不对因您使用 CSM 而产生的任何后果性、特殊性、间接性、附带性或惩罚性损害承担责任，无论您是否告知进芯电子此类损害的可能。损害包括且不限于数据损失、商誉损失、使用损失或业务中断等其他经济损失。

### 5.1.12. 外设中断扩展（PIE）模块

PIE 块用于将多个中断源多路复用成一组较小的中断输入，PIE 块最多可支持 96 个外围中断。在 AVP32F069 上，外设使用了 96 个中断中的 72 个。96 个中断被分为 8 组，每组馈入 12 条 CPU 中断线（INT1 到 INT12）中的 1 条。96 个中断有其自身向量表，存储在专用 RAM 块中，用户可以覆盖 RAM 以定位该向量表。在服务中断时，CPU 会自动获取向量。需要八个 CPU 时钟周期来获取向量并保存关键 CPU 寄存器。因此，CPU 可以快速响应中断事件，中断的优先级由硬件和软件控制，每个单独的中断都可以在 PIE 块中使能或禁用。

### 5.1.13. 外部中断（XINT1 至 XINT3）

器件支持三个可屏蔽的外部中断（XINT1–XINT3）。每个中断皆可选择负边沿、正边沿或双边沿触发，也可以启用或禁用。这些中断还包含一个 16 位自由递增计数器，当检测到有效的中断边沿时，该计数器将重置为零。此计数器可用于准确标记中断时间。外部中断没有专用引脚，XINT1、XINT2 和 XINT3 中断可以接受来自 GPIO0–GPIO31 引脚的输入。

### 5.1.14. 内部零引脚振荡器、振荡器和锁相环

此器件可选用以下任一时钟源：两个内部零引脚振荡器中的任何一个、外部振荡器或晶体振荡器。片内 PLL（锁相环）最大支持 18 倍频。PLL 倍频比例可以在软件中动态更改，使用户能够在需要较低的功率操作时降低工作频率。PLL 块可以设置为旁路模式。第二个 PLL（PLL2）提供给 HRCAP 模块。

### 5.1.15. 看门狗

器件包含两个看门狗：CPU 看门狗（用于监控内核）和 NMI 看门狗（时钟丢失检测电路）。用户程序必须在一定时间内定期复位 CPU 看门狗计数器；否则，CPU 看门狗将产生一个复位信号给处理器。如有必要，可以禁用 CPU 看门狗。NMI 看门狗仅在时钟故障时启用，并产生中断或器件复位。

### 5.1.16. 外设时钟

可以使能或禁用每个外设的时钟，以减少未使用外设时的功耗。此外，到串行端口（I2C 除外）的系统时钟可以从 CPU 时钟分频。

### 5.1.17. 低功耗模式

AVP32F069 芯片是完全静态 CMOS 器件。提供三种低功耗模式：

IDLE: 可以选择性地关闭外设时钟，只有那些在 IDLE 模式期间必须运行的外设才保持运行状态。使能来自正在工作外设或看门狗定时器的中断将使处理器从 IDLE 模式唤醒。

STANDBY: 关闭 CPU 和外设的时钟。此模式使振荡器和 PLL 正常工作。外部中断事件将唤醒处理器和外设，并在检测到中断事件后的下一个有效周期开始执行。

HALT: 此模式基本上会关闭设备并将其置于最低功耗模式。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。为防止关闭内部振荡器，需要使用 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 看门狗计时。如果将片载晶体振荡器用作时钟源，则在此模式下将其关闭。复位或外部信号（通过 GPIO 引脚）或 CPU 看门狗可以将器件从 HALT 模式唤醒。

CPU 时钟（OSCCLK）和看门狗时钟应来自同一时钟源，然后再尝试将设备置于 HALT 或 STANDBY 状态。

### 5.1.18. 外设帧 0、1、2、3 (PFn)

外设分为 4 个部分，映射如下：

| | | |
|---|---|---|
| PF0: | PIE: | PIE 中断使能和控制寄存器以及 PIE 向量表 |
| | Flash: | Flash 等待状态寄存器 |
| | Timers: | CPU 定时器 0、定时器 1、定时器 2 |
| | CSM: | 代码安全模块密钥寄存器 |
| | ADC: | ADC 结果寄存器 |
| | CLA: | 控制率加速器和消息 RAM |
| PF1: | GPIO: | GPIO 选择配置和控制寄存器 |
| | eCAN: | 增强型控制局域网络配置和控制寄存器 |

| | | |
|---|---|---|
| PF2: | SYS: | 系统控制寄存器 |
| | SCI: | SCI 控制寄存器和 SCI RX/TX 寄存器 |
| | SPI: | SPI 控制寄存器和 SPI RX/TX 寄存器 |
| | ADC: | ADC 状态、控制和配置寄存器 |
| | I2C: | I2C 模块和寄存器 |
| | XINT: | 外部中断寄存器 |
| PF3: | McBSP: | 多通道缓冲串行端口寄存器 |
| | ePWM: | 增强型脉冲宽度调制模块和寄存器 |
| | eCAP: | 增强型捕获模块和寄存器 |
| | eQEP: | 增强型正交编码脉冲模块和寄存器 |
| | Comparators: | 比较器模块 |
| | USB: | 通用串行总线模块和寄存器 |

### 5.1.19. 通用输入/输出 (GPIO) 复用器

大多数外设信号与通用输入/输出（GPIO）信号复用。这使得用户能够在不使用外设信号或者功能时将相应引脚用作 GPIO。复位时，GPIO 引脚被配置为输入。用户可独立设置每个引脚为 GPIO 模式或外设模式。对于特定的输入引脚，用户还可选择输入限定周期的数量。这是为了过滤不必要的噪声毛刺脉冲。GPIO 信号也可用于使器件退出特定低功耗模式。

### 5.1.20. 32 位 CPU 定时器 (0、1、2)

CPU 定时器 0、1 和 2 是完全相同的 32 位定时器，这些定时器带有可预先设定的周期和 16 位时钟预分频。定时器有一个 32 位递减计数寄存器，此寄存器在计数器达到 0 时生成一个中断。定时器的计数器在一个时钟频率下递减计数，此时钟频率是 CPU 时钟经过分频而来（此分频的值是通过寄存器设置的）。当此计数器达到 0 时，它将自动重新加载 32 位周期值。

CPU 定时器 0 可作为通用定时器并连接至 PIE 模块。CPU 定时器 1 可作为通用定时器并连接至 CPU 的 INT13。CPU定时器 2为 操 作 系 统 预留，并且连接到 CPU的 INT14。如果未使用 操作系统，CPU定时器 2也可作为通用定时器使用。

CPU 定时器 2 时钟源：

- SYSCLKOUT (默认)
- 内部 0 引脚振荡器 1 (INTOSC1)
- 内部 0 引脚振荡器 2 (INTSOC2)
- 外部时钟源

### 5.1.21. 控制外设

该设备支持以下用于嵌入式控制和通信的外围设备：

- ePWM： 增强型 PWM 能够输出独立/互补的 PWM 波，前沿/后沿的可调死区，锁定/逐周期翻转机制。PWM 的部分端口支持高精度周期和占空比。
- eCAP： 增强型捕获外设使用 32 位时基，并在连续/单次捕获模式下最多记录四个可编程事件。该外设也能配置产生 PWM 信号。
- eQEP： 增强型 QEP 外设使用 32 位位置计数器，支持使用捕获单元的低速测量和使用 32 位单元定时器的高速测量。该外设有一个看门狗定时器来检测电机失速和输入错误检测逻辑来识别 QEP 信号中的同时边沿转换。
- ADC： ADC 模块是一个 12 位转换器。ADC 有多达 16 个单端通道。ADC 还包含两个用于同时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用于为比较器的一个输入提供电源的内部 10 位基准组成。
- HRCAP： 高分辨率捕获外设通过一个以 HCCAPCLK 为时钟的 16 位计数器在正常捕获模式下运行，或者通过使用内置校准逻辑与校准库一起在高分辨率捕获模式下运行。

### 5.1.22. 串行接口外设

该设备支持以下串行通信外设：

- SPI： SPI 是一个高速、同步串行 I/O 端口，它允许以可配置的传输速率将具有可配置长度（1 至 16 位）的串行位流移入和移出器件。通常，SPI 用于 MCU 与外部设备或者是处理器进行数据交互。典型应用包括通过移位寄存器、显示驱动器和 ADC 等设备进行外部 I/O 或外设扩展。SPI 的主/从操作支持多设备通信。 SPI 包含一个 4 级接收和发送 FIFO，用于减少中断服务开销。
- SCI： 串行通信接口是 2 线异步串口，俗称 UART。 SCI 包含一个 4 级接收和发送 FIFO，用于减少中断服务开销。
- I2C： I2C 模块为 MCU 与其他设备提供接口，其他设备必须符合飞利浦半导体内部 IC 总线 (I2C-bus) 规范 2.1 版。连接到此 2 线串行总线的外部组件可以通过 I2C 模块向 MCU 发送/接收多达 8 位数据。I2C 包含一个 4 级接收和发送 FIFO，用于减少中断服务开销。
- eCAN： 增强版本的 CAN 外设。eCAN 支持 32 个邮箱、消息时间戳，并且符合 ISO 11898-1 (CAN 2.0B)。
- McBSP： 多通道缓冲串行端口 (McBSP) 连接到 E1/T1 线路、用于调制解调器应用的电话质量编解码器或高质量立体声音频 DAC 设备。DMA 支持 McBSP 接收和发送寄存器，

以显著减少服务该外设的开销。 每个 McBSP 模块都可以根据需要配置为 SPI。

- USB： 该 USB 外设符合 USB 2.0 协议，可用于全速（12-Mbps）设备控制器，或者全速（12-Mbps）/低速（1.5-Mbps）主机控制器。该控制器支持 6 个用户可配置端点，所有的端点能够通过 DMA 访问。另外包括一个专用端点 0。所有包的发送与接收缓存在专用 4KB 端点存储器中。该 USB 外设支持三种传输模式：控制、中断和批量传输。

## 5.2. 内存映射

- 内存块不支持缩放。
- 外设帧 0、外设帧 1、外设帧 2 和外设帧 3 内存映射仅限于数据内存。用户程序无法访问程序空间中的这些内存映射。
- “受保护”是指保留“写后读”的操作顺序，而非流水线顺序。
- 在配置后，某些内存范围可防止虚假写入。
- 地址 0x3D7C80~0x3D7CC0 包含内部振荡器和 ADC 校准例程，属出厂配置内存。用户无法对这些位置进行编程。

| | 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 | Reserved | |
| 0x00 2000 | Reserved | |
| 0x00 4000 | USB Control Registers | Reserved |
| 0x00 5000 | Peripheral Frame 3 (4K x 16,Protected) DMA-Accessible | |
| 0x00 6000 | Peripheral Frame 1 (4K x 16, Protected) | |
| 0x00 7000 | Peripheral Frame 2 (4K x 16, Protected) | |
| 0x00 8000 | L0 DPSARAM (2K x 16) (0-Wait, Secure Zone + ECSL, CLA Data RAM2) | |
| 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 RAM1) | |
| 0x00 9000 | L3 DPSARAM (4K x 16) (0-Wait, Secure Zone + ECSL, CLA Program RAM) | |
| 0x00 A000 | L4 SARAM (8K x 16) (0-Wait, Secure Zone + ECSL) | |
| 0x00 C000 | L5 DPSARAM (8K x 16) (0-Wait, DMA RAM 0) | |
| 0x00 E000 | L6 DPSARAM (8K x 16) (0-Wait, DMA RAM 1) | |
| 0x01 0000 | L7 DPSARAM (8K x 16) (0-Wait, DMA RAM 2) | |
| 0x01 2000 | L8 DPSARAM (8K x 16) (0-Wait, DMA RAM 3) | |
| 0x01 4000 | Reserved | |
| 0x3D 7800 | User OTP (1K x 16, Secure Zone + ECSL) | |
| 0x3D 7BFA | Reserved | |
| 0x3D 7C80 | Calibration Data | |
| 0x3D 7CC0 | Get_mode function | |
| 0x3D 7CD0 | Reserved | |
| 0x3D 7E80 | PARTID | |
| 0x3D 7E82 | Calibration Data | |
| 0x3D 7EB0 | Reserved | |
| 0x3D 8000 | FLASH (128K x 16) (8 Sectors, Secure Zone + ECSL) | |
| 0x3F 7FF8 | 128-Bit Password | |
| 0x3F 8000 | IQmath Libraries (16K x 16, 0-Wait State) | |
| 0x3F F3B0 | Boot ROM (16K x 16, 0-Wait State) | |
| 0x3F FFC0 | CPU Vector Table (32 Vectors, Enabled if VMAP = 1) | |

**图 5-1 AVP32F069 内存映射**

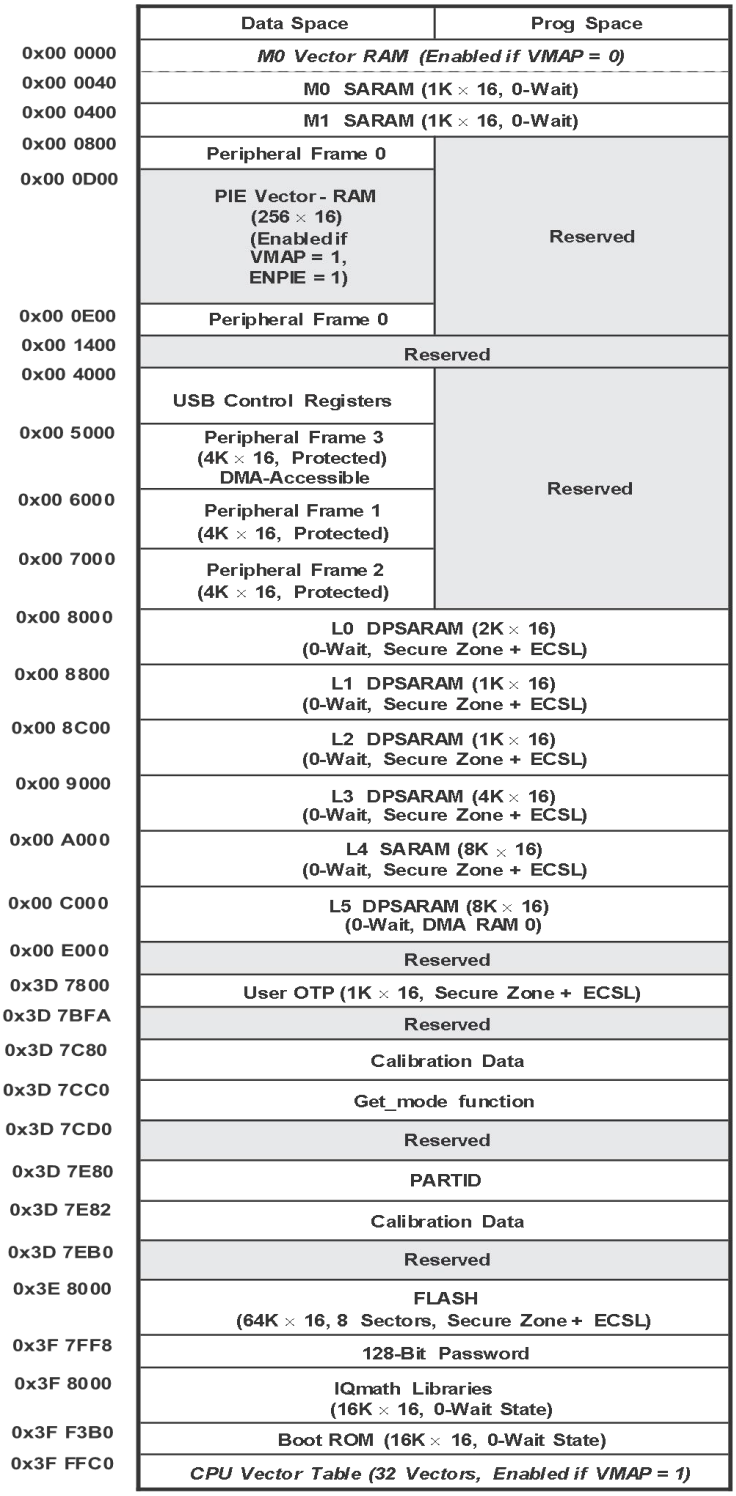



**图 5-2 AVP32F062 内存映射**

**表 5-3 AVP32F069 Flash 扇区地址**

| 地址范围 | 程序和数据空间 |
|---|---|
| 0x3D 8000 to 0x3D BFFF | 扇区H (16K × 16) |
| 0x3D C000 to 0x3D FFFF | 扇区 G (16K × 16) |
| 0x3E 0000 to 0x3E 3FFF | 扇区 F (16K × 16) |
| 0x3E 4000 to 0x3E 7FFF | 扇区 E (16K × 16) |
| 0x3E 8000 to 0x3E BFFF | 扇区 D (16K × 16) |
| 0x3E C000 to 0x3E FFFF | 扇区 C (16K × 16) |
| 0x3F 0000 to 0x3F 3FFF | 扇区 B (16K × 16) |
| 0x3F 4000 to 0x3F 7FF5<br>0x3F 7FF6 to 0x3F 7FF7<br>0x3F 7FF8 to 0x3F 7FFF | 扇区 A (16K × 16)<br>引导到Flash入口 (程序跳转指令到此)<br>安全密钥 (128-Bit) (禁止写全0) |

外设帧 1 和外设帧 2 被放到一组，以使这些块成为受保护的写/读外设块。受保护模式确保所有到这些块的访问如文档中描述的一样。由于 AVP32F069 流水线，在对不同内存位置读取之前的写入操作将出现在 CPU 内存总线上相反的顺序。这会导致特定外设应用中的问题，在此类应用中，用户认为写入会首先发生（如文档所描述的那样）。AVP32F069 支持一个块保护模式，在这个模式中，可对一个内存区域进行保护，以确保操作按照本文档所描述的那样发生（代价是增加了额外周期以校正运行）。可对该模式进行编程，并且，默认情况下，它将保护所选的区域。

针对内存映射区域内不同空间的等待状态如表 5-4 所述。

**表 5-4 等待状态**

| 区域 | 等待状态 (CPU) | 备注 |
|---|---|---|
| M0 和 M1 SARAM | 0- wait | 固定 |
| 外设帧 0 | 0-wait | |
| 外设帧 1 | 0-wait (写)<br>2-wait (读) | 通过外设生成就绪信号扩展周期<br>PF1寄存器的背靠背写入操作将导致1周期暂停（1周期延时） |
| 外设帧 2 | 0-wait (写)<br>2-wait (读) | 固定值，外设无法扩展周期 |
| 外设帧 3 | 0-wait (写)<br>2-wait (读) | 假定 CPU 和 CLA/DMA 周期之间没有冲突。等待状态可以通过外设生成的就绪来扩展。 |
| L0–L8 SARAM | 0-wait 数据和程序 | 假设没有CPU冲突 |
| OTP | 可编程<br>最小值 1-wait | 通过Flash寄存器编程<br>1-wait是等待所允许的最小数值 |
| FLASH | 可编程<br>0-wait 分页最小值<br>1-wait 随机最小值<br>随机 ≥分页 | 通过Flash寄存器编程 |
| FLASH Password | 16-wait | 密码位置的等待时间是固定的 |
| Boot-ROM | 0-wait | |

## 5.3. 寄存器映射

该器件包含四个外设寄存器空间。这些空间的分类如下：

- 外设帧 0：直接映射到 CPU 内存总线的外设，支持 16 位和 32 位访问，详见表 5-5。
- 外设帧 1：映射到 32 位外设总线的外设，详见表 5-6。
- 外设帧 2：映射到 16 位外设总线的外设，详见表 5-7。
- 外设帧 3：映射到 32 位外设总线的外设，详见表 5-8。

**表 5-5 外设帧 0 寄存器(1)**

| 名称 | 地址范围 | 大小 (×16) | 访问类型(2) |
|---|---|---|---|
| 器件仿真寄存器 | 0x00 0880至0x00 0984 | 261 | 受EALLOW保护 |
| 系统电源控制寄存器 | 0x00 0985至0x00 0987 | 3 | 受EALLOW保护 |
| FLASH寄存器(3) | 0x00 0A80至0x00 0ADF | 96 | 受EALLOW保护 |
| 代码安全模块寄存器 | 0x00 0AE0至0x00 0AEF | 16 | 受EALLOW保护 |
| ADC寄存器 (0等待只读) | 0x00 0B00至0x00 0B0F | 16 | 不受EALLOW保护 |
| CPU-TIMER0, CPU-TIMER1, CPU-TIMER2 寄存器 | 0x00 0C00至0x00 0C3F | 64 | 不受EALLOW保护 |
| PIE 寄存器 | 0x00 0CE0至0x00 0CFF | 32 | 不受EALLOW保护 |
| PIE 向量表 | 0x00 0D00至0x00 0DFF | 256 | 受EALLOW保护 |
| DMA 寄存器 | 0x00 1000至0x00 11FF | 512 | 受EALLOW保护 |
| CLA 寄存器 | 0x00 1400至0x00 147F | 128 | 受EALLOW保护 |
| CLA 到 CPU 通信 RAM (CPU写无效) | 0x00 1480至0x00 14FF | 128 | 不适用 |
| CPU 到 CLA 通信 RAM (CLA写无效) | 0x00 1500至0x00 157F | 128 | 不适用 |

(1) 帧 0 寄存器支持 16 位和 32 位访问。

(2) 如果寄存器是受 EALLOW 保护的，则在写入操作前，必须执行 EALLOW 指令。EDIS 指令禁用写入，以防止丢失的代码或指针损坏寄存器内容。

(3) 闪存寄存器也受到代码安全模块(CSM)的保护。

**表 5-6 外设帧 1 寄存器**

| 名称 | 地址范围 | 大小 (×16) | 访问类型 |
|---|---|---|---|
| eCAN-A寄存器 | 0x00 6000至0x00 61FF | 512 | (1) |
| HRCAP1寄存器 | 0x00 6AC0至0x00 6ADF | 32 | (1) |
| HRCAP2寄存器 | 0x00 6AE0至0x00 6AFF | 32 | (1) |
| HRCAP3 寄存器 | 0x00 6C80至0x00 6C9F | 32 | (1) |
| HRCAP4寄存器 | 0x00 6CA0至0x00 6CBF | 32 | (1) |
| GPIO 寄存器 | 0x00 6F80至0x00 6FFF | 128 | (1) |

(1) 有些寄存器受保护，具体见各模块寄存器说明。

**表 5-7 外设帧 2 寄存器**

| 寄存器名称 | 地址范围 | 大小 (×16) | 访问类型 |
|---|---|---|---|
| 系统控制寄存器 | 0x00 7010至0x00 702F | 32 | 受EALLOW保护 |
| SPI-A 寄存器 | 0x00 7040至0x00 704F | 16 | 不受EALLOW保护 |
| SCI-A寄存器 | 0x00 7050至0x00 705F | 16 | 不受EALLOW保护 |
| NMI 看门狗中断寄存器 | 0x00 7060至0x00 706F | 16 | 受EALLOW保护 |
| 外部中断寄存器 | 0x00 7070至0x00 707F | 16 | 受EALLOW 保护 |
| ADC 寄存器 | 0x00 7100至0x00 717F | 128 | (1) |
| SPI-B 寄存器 | 0x00 7740至0x00 774F | 16 | 不受EALLOW 保护 |
| SCI-B寄存器 | 0x00 7750至0x00 775F | 16 | 不受EALLOW 保护 |
| I2C-A寄存器 | 0x00 7900至0x00 793F | 64 | (1) |

(1) 有些寄存器受保护，具体见各模块寄存器说明。

**表 5-8 外设帧 3 寄存器**

| 名称 | 地址范围 | 大小 (×16) | 访问类型 |
|---|---|---|---|
| USB0 寄存器 | 0x00 4000 至 0x00 4FFF | 4096 | 不受EALLOW 保护 |
| McBSP-A 寄存器 | 0x00 5000 至 0x00 503F | 64 | 不受EALLOW 保护 |
| 比较器 1寄存器 | 0x00 6400 至 0x00 641F | 32 | (1) |
| 比较器 2 寄存器 | 0x00 6420 至 0x00 643F | 32 | (1) |
| 比较器 3 寄存器 | 0x00 6440 至 0x00 645F | 32 | (1) |
| ePWM1 + HRPWM1 寄存器 | 0x00 6800 至 0x00 683F | 64 | (1) |
| ePWM2 + HRPWM2 寄存器 | 0x00 6840 至 0x00 687F | 64 | (1) |
| ePWM3 + HRPWM3 寄存器 | 0x00 6880 至 0x00 68BF | 64 | (1) |
| ePWM4 + HRPWM4寄存器 | 0x00 68C0 至 0x00 68FF | 64 | (1) |
| ePWM5 + HRPWM5 寄存器 | 0x00 6900 至 0x00 693F | 64 | (1) |
| ePWM6 + HRPWM6 寄存器 | 0x00 6940 至 0x00 697F | 64 | (1) |
| ePWM7 + HRPWM7 寄存器 | 0x00 6980 至 0x00 69BF | 64 | (1) |
| ePWM8 + HRPWM8 寄存器 | 0x00 69C0 至 0x00 69FF | 64 | (1) |
| eCAP1 寄存器 | 0x00 6A00 至 0x00 6A1F | 32 | 不受EALLOW 保护 |
| eCAP2 寄存器 | 0x00 6A20 至 0x00 6A3F | 32 | 不受EALLOW 保护 |
| eCAP3 寄存器 | 0x00 6A40 至 0x00 6A57 | 32 | 不受EALLOW 保护 |
| eQEP1寄存器 | 0x00 6B00 至 0x00 6B3F | 64 | (1) |
| eQEP2 寄存器 | 0x00 6B40 至 0x00 6B7F | 64 | (1) |

(1) 某些寄存器受 ELLOW 保护，具体见各模块寄存器说明。

## 5.4. 器件仿真寄存器

这些寄存器用于控制 AVP32F069 CPU 的保护模式和监控某些关键器件信号。该寄存器定义如表 5-9 所示。

**表 5-9 器件仿真寄存器**

| 名称 | 地址范围 | 大小 (×16) | 说明 | | | 允许保护 |
|---|---|---|---|---|---|---|
| DEVICECNF | 0x0880-0x0881 | 2 | 设备配置寄存器 | | | Yes |
| PARTID | 0x3D 7E80 | 1 | PartID Register | AVP32F06X | 0x009F | No |
| CLASSID | 0x0882 | 1 | Class ID Register | | 0x009F | No |
| REVID | 0x0883 | 1 | Revision ID Register | Silicon Rev. A | 0x0001 | No |
| ALGORITHM | 0x3D 7FBE | 1 | Algorithm solidification | Normal part | 0xFFFF | No |

## 5.5. VREG, BOR, POR

因为内核和 I/O 电路在不同的电压下工作，所以芯片内部集成了两个片载稳压器 VREG18 与 VREG11（统称为 VREG），VREG18 用于从 VDDIO 产生 VDD 电压，VREG11 用于从 VDD 产生 1.1V 的内核电压 VDD11。

在芯片内部集成稳压器避免了在应用板上再使用外部稳压器所增加的成本和面积。此外，内部上电复位（POR）与欠压复位（BOR）电路在通电和运行模式下监控 VDDIO、VDD 与 VDD11 电压。

### 5.5.1. 片载 VREG

片载 VREG18 从 VDDIO 电源上产生 VDD 电压，因此，尽管每个 VDD 引脚上都需要电容器来稳定产生的电压，但在使用过程中不需要向这些引脚提供工作电压。相反，如果在应用中出现片载 VREG 产生的 VDD 电压与 VDD 引脚上外供电源电压竞争的情况，则需禁用 VREG18。VREG11 从 VDD 电压上产生内核电压 VDD11，其在芯片运行时始终工作，不能关闭。VDD11 电源电压可从 TEST2 引脚量测，TEST2 引脚在芯片工作时需始终保持悬空，否则可能造成内部 VDD11 电源不稳定。

#### 5.5.1.1. 使用片载 VREG18

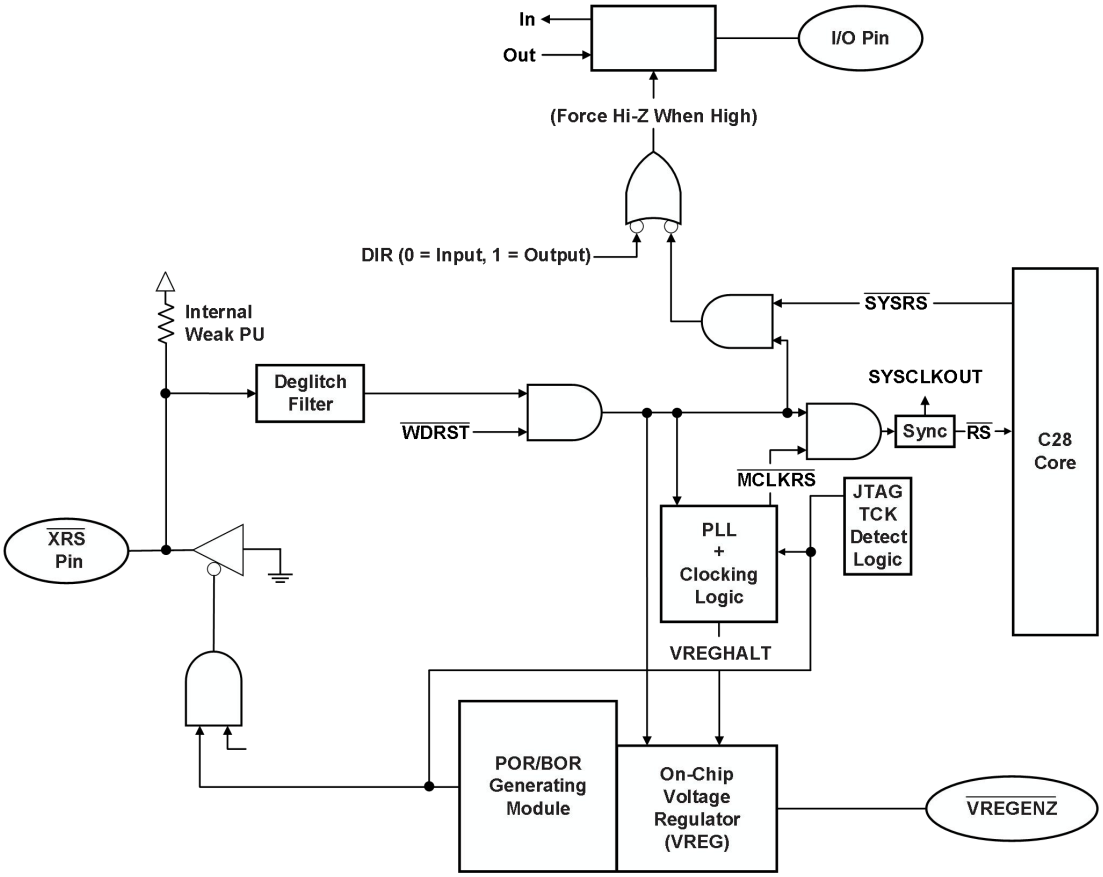
若使用片载 VREG，则$\overline{\text{VREGENZ}}$引脚需置低电平，并且应向 VDDIO 和 VDDA 引脚提供适当的推荐工作电压。在这种情况下，VDD 电压将由 VREG 产生，每个 VDD 引脚需要 1.2μF（最小）的电容才能正确调节 VREG。这些电容应尽可能靠近 VDD 引脚。使用内部 VREG 时，VDD 不支持驱动外部负载。

#### 5.5.1.2. 禁用片载 VREG18

为了节能，还可以禁用片载 VREG18。在禁用 VREG18 时，$\overline{\text{VREGENZ}}$引脚需置高电平，并需使用更高效的外部稳压器向 VDD 引脚提供 VDD 电压，以此使得内部 VREG11 正常工作，产生稳定可靠的内核逻辑工作电压 VDD11。

## 5.5.2. 片载上电复位（POR）和欠压复位（BOR）电路

上电复位（POR）和欠压复位（BOR）这两个片载监控电路，减少了在应用板上监控供电轨的负担。POR 的目的是在芯片整个上电过程中建立一个清晰的上电复位点。相对于 BOR，POR 的触发点更宽松、更低，用于在设备运行期间监控 VDD 与 VDDIO 电压轨是否发生偏离工作条件的情况，POR 功能始终存在于 VDDIO、VDD 和 VDD11 电压轨。而 BOR 功能则是在器件首次上电后，VDDIO 上始终存在 BOR 功能，在内部 VREG18 启用时，VDD 与 VDD11 上始终存在 BOR 功能（$\overline{\text{VREGENZ}}$引脚处于低位）。当电压轨中的某个电压低于各自的触发点时，这两个功能都会将$\overline{\text{XRS}}$引脚置低。VDD 的 BOR 触发点和过压触发点在推荐的正常工作电压范围之外。如果影响系统的过压或欠压条件是应用方面的问题，则无法确保设备正常运行，需添加外部电压监控器。图 5-3 所示为 VREG、POR、BOR 及复位信号连接。为了禁用 VDDIO、VDD 和 VDD11 的 BOR 功能，在 BORCFG 寄存器中提供了一个控制位。




A. $\overline{\text{WDRST}}$是来自 CPU 看门狗的复位信号。

B. $\overline{\text{PBRS}}$是来自 POR/BOR 模块的复位信号。

**图 5-3 VREG + POR + BOR +复位信号连接**

# 5.6. 系统控制

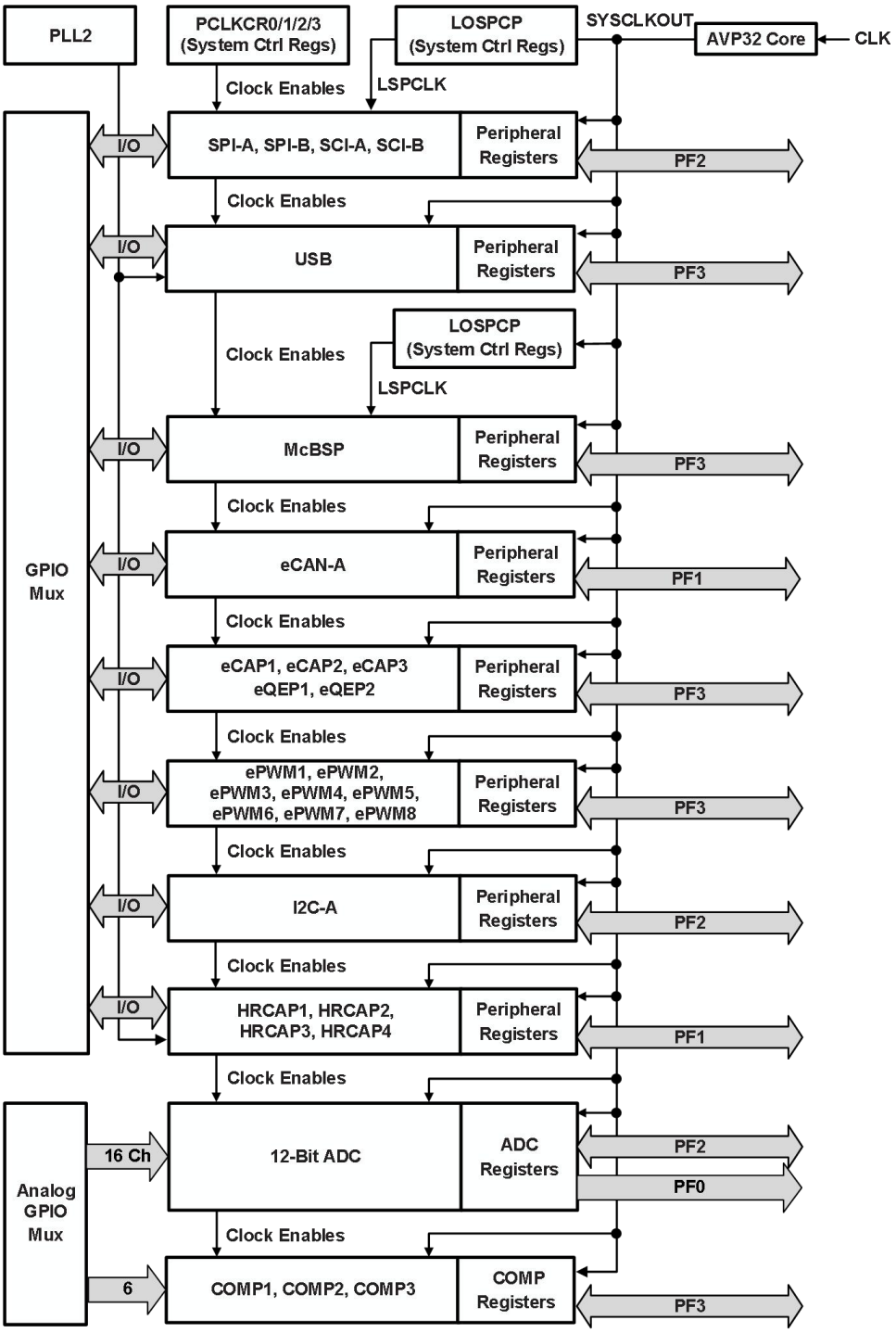
本节介绍振荡器和时钟机制、看门狗功能以及低功耗模式。

**表 5-10 PLL、时钟、看门狗和低功耗模式寄存器**

| 名称 | 地址 | 大小 (×16) | 描述(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 | 时钟寄存器 |
| PLLLOCKPRD | 0x00 7013 | 1 | PLL 锁定所需周期 |
| INTOSC1TRIM | 0x00 7014 | 1 | 内部OSC1修调寄存器 |
| INTOSC2TRIM | 0x00 7016 | 1 | 内部OSC2修调寄存器 |
| PCLKCR2 | 0x00 7019 | 1 | 外设时钟控制寄存器 2 |
| 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 | 系统控制和状态寄存器 |
| WDCNTR | 0x00 7023 | 1 | 看门狗计数寄存器 |
| 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 |
| JTAGDEBUG | 0x00 702A | 1 | JTAG 端口调试寄存器 |
| PLL2CTL | 0x00 7030 | 1 | PLL2 配置寄存器 |
| PLL2MULT | 0x00 7032 | 1 | PLL2 倍频寄存器 |
| PLL2STS | 0x00 7034 | 1 | PLL2 状态锁定寄存器 |
| SYSCLK2CNTR | 0x00 7036 | 1 | SYSCLK2 时钟计数器 |
| EPWMCFG | 0x00 703A | 1 | ePWM DMA/CLA 配置寄存器 |

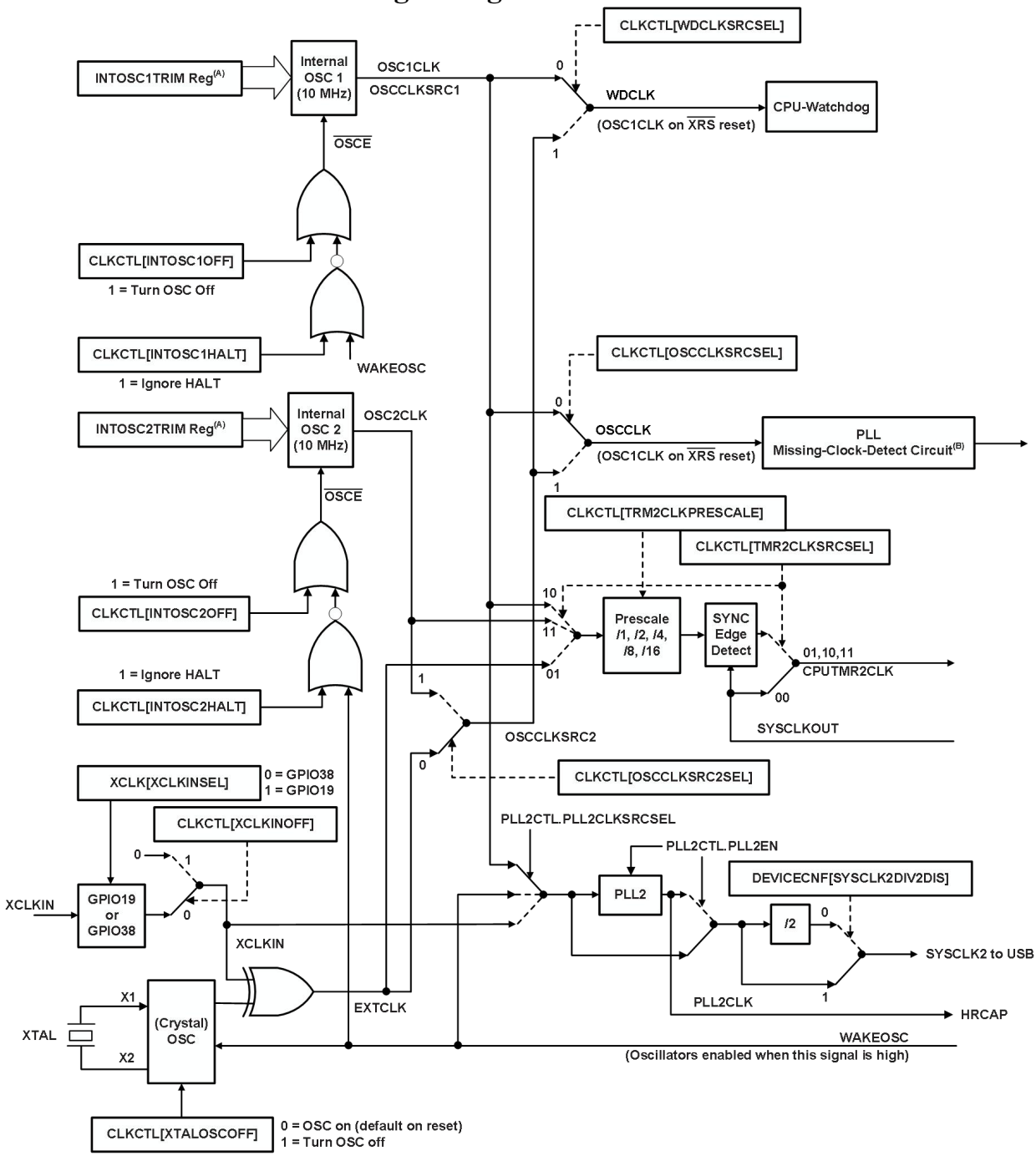
(1) 表中所有寄存器都受 EALLOW 保护。

图 5-4 所示为时钟和复位域。图 5-5 所示为供设备操作的不同的时钟源（包括内部和外部）。




A. CLKIIN 是输入 CPU 的时钟。CLKIN 通过 CPU 输出到 SYSCLKOUT（即，CLKIN 和 SYSCLKOUT 同频率）。

**图 5-4 时钟和复位域**




A. 从基于 OTP 校验函数加载寄存器。

B. 详细的时钟检测逻辑参考第 5.6.5 节。

**图 5-5 时钟树**

## 5.6.1. 内部零引脚振荡器

AVP32F069 器件包含两个独立的内部零引脚振荡器 INTOSC1 与 INTOSC2。默认情况下，两个振荡器在上电时都打开，此时内部振荡器 1 是默认的时钟源。为了降低功耗，用户可以关闭未使用的振荡器。这些振荡器的中心频率由它们各自的振荡器寄存器决定，作为引导 ROM 执行的一部分，在校准程序中写入。请参考第 5.9 节以获取有关这些振荡器的更多信息。

### 5.6.2. 晶体振荡器选项

片载晶振 X1 和 X2 引脚为 1.8V 电平信号，不能是 3.3V 电平信号。如果使用系统 3.3V 外部晶振作为时钟源，应该只连接到 XCLKIN 引脚。X1 引脚不推荐用于单端时钟输入，应与 X2 配合作为晶体振荡器。

外部石英晶体的典型规格(基本模型，并联谐振)列在表 5-11。此外，ESR 电阻范围在 30~150Ω。对于表 5-11，并联电容 Cshunt 应小于或等于 5pF。

**表 5-11 外部石英晶体的典型规格**

| 频率 (MHz) | Rd (Ω) | CL1 (pF) | CL2 (pF) |
|---|---|---|---|
| 5 | 2200 | 18 | 18 |
| 10 | 470 | 15 | 15 |
| 15 | 0 | 15 | 15 |
| 20 | 0 | 12 | 12 |

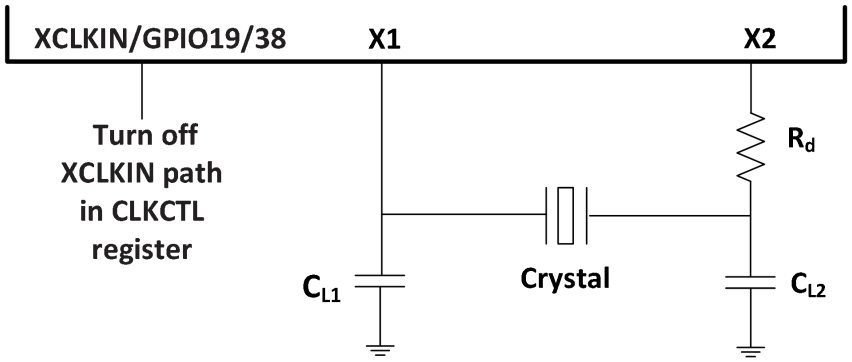



**图 5-6 片载石英晶体振荡器**

注意

1.CL1 和 CL2 为电路板和元件的总电容，不包括 IC 和晶体。这个值通常大约是晶体负载电容值的两倍。

2.晶体的负载电容在厂家的晶体规格中有说明。

3.推荐客户其设备与 MCU 芯片关于谐振器/晶体供应商的业务。谐振器/晶体供应商有设备和专业知识来调整振荡电路。供应商还可以向客户提供适当的元件值，以便在整个操作范围内产生适当的启动和稳定性。

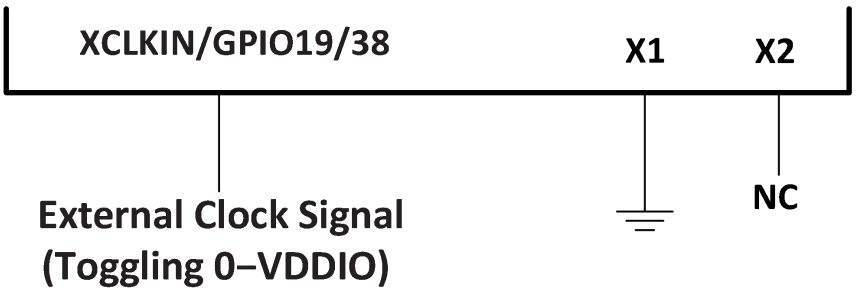



**图 5-7 使用一个 3.3v 外部振荡器**

### 5.6.3. 基于 PLL 时钟模块

这些设备集成在一个芯片上，基于 PLL 时钟模块。该模块为设备提供所有必要的时钟信号，以及对低功耗模式进入的控制。PLL 有一个 5 位寄存器控制 PLLCR[DIV]来选择不同的 CPU 时钟速率.在写入 PLLCR 寄存器之前应该禁用看门狗模块，在 PLL 模块稳定后看门狗模块可以重新启用（如果需要），这需要 1ms。输入时钟和 PLLCR[DIV]位的选择应使 PLL（VCOCLK）的输出频率至少为 50MHz。

**表 5-12 PLL 设置**

| PLLCR[DIV] VALUE(2) (3) | SYSCLKOUT (CLKIN) | | |
|---|---|---|---|
| | PLLSTS[DIVSEL] = 0 or 1(1) | PLLSTS[DIVSEL] = 2 | PLLSTS[DIVSEL] = 3 |
| 00000 (PLL bypass) | OSCCLK/4 (Default)(2) | OSCCLK/2 | OSCCLK |
| 00001 | (OSCCLK * 1)/4 | (OSCCLK * 1)/2 | (OSCCLK * 1)/1 |
| 00010 | (OSCCLK * 2)/4 | (OSCCLK * 2)/2 | (OSCCLK * 2)/1 |
| 00011 | (OSCCLK * 3)/4 | (OSCCLK * 3)/2 | (OSCCLK * 3)/1 |
| 00100 | (OSCCLK * 4)/4 | (OSCCLK * 4)/2 | (OSCCLK * 4)/1 |
| 00101 | (OSCCLK * 5)/4 | (OSCCLK * 5)/2 | (OSCCLK * 5)/1 |
| 00110 | (OSCCLK * 6)/4 | (OSCCLK * 6)/2 | (OSCCLK * 6)/1 |
| 00111 | (OSCCLK * 7)/4 | (OSCCLK * 7)/2 | (OSCCLK * 7)/1 |
| 01000 | (OSCCLK * 8)/4 | (OSCCLK * 8)/2 | (OSCCLK * 8)/1 |
| 01001 | (OSCCLK * 9)/4 | (OSCCLK * 9)/2 | (OSCCLK * 9)/1 |
| 01010 | (OSCCLK * 10)/4 | (OSCCLK * 10)/2 | (OSCCLK * 10)/1 |
| 01011 | (OSCCLK * 11)/4 | (OSCCLK * 11)/2 | (OSCCLK * 11)/1 |
| 01100 | (OSCCLK * 12)/4 | (OSCCLK * 12)/2 | (OSCCLK * 12)/1 |
| 01101 | (OSCCLK * 13)/4 | (OSCCLK * 13)/2 | (OSCCLK * 13)/1 |
| 01110 | (OSCCLK * 14)/4 | (OSCCLK * 14)/2 | (OSCCLK * 14)/1 |
| 01111 | (OSCCLK * 15)/4 | (OSCCLK * 15)/2 | (OSCCLK * 15)/1 |
| 10000 | (OSCCLK * 16)/4 | (OSCCLK * 16)/2 | (OSCCLK * 16)/1 |
| 10001 | (OSCCLK * 17)/4 | (OSCCLK * 17)/2 | (OSCCLK * 17)/1 |
| 10010 | (OSCCLK * 18)/4 | (OSCCLK * 18)/2 | (OSCCLK * 18)/1 |
| | | | |

(1) 默认情况下，PLLSTS[DIVSEL]配置为/4. (引导 ROM 会将此操作更改为/1.) PLLSTS[DIVSEL]在写入 PLLCR 之前必须为 0，并且只能在 PLLSTS[PLLLOCKS]=1 之后更改。

(2) PLL 控制寄存器(PLLCR)和 PLL 状态寄存器(PLLSTS)仅通过$\overline{XRS}$信号或监督器复位重置到其默认状态。由调试器发出的重置或丢失的时钟检测逻辑没有作用。

(3) 此寄存器受 EALLOW 保护。

**表 5-13 CLKIN 分频选择**

| PLLSTS [DIVSEL] | CLKIN DIVIDE |
|---|---|
| 0 | /4 |
| 1 | /4 |
| 2 | /2 |
| 3 | /1 |

基于 PLL 的时钟模块提供四种操作模式：

- INTOSC1 (内部零引脚振荡器 1)：这是片载内部振荡器 1，可为看门狗块、内核和 CPU 定时器 2 提供时钟。
- INTOSC2 (内部零引脚振荡器 2)：这是片载内部振荡器 2，也可为看门狗块、内核和 CPU 定

时器 2 提供时钟。INTOSC1 和 INTOSC2 均可独立选择用于看门狗模块、内核和 CPU 定时器 2。

• 晶振/谐振器：片载（晶体）振荡器能够使用连接在设备上的外部晶体/谐振器来提供时间基础。晶体/谐振器连接到 X1/X2 引脚，有些设备可能没有 X1/X2 引脚。详见第 3.2.1 节。

• 外部时钟源：如果不使用片载（晶体）振荡器，该模式允许旁路片载（晶体）振荡器。设备时钟由 XCLKIN 引脚上的外部时钟源输入产生。XCLKIN 与 GPIO19 或 GPIO38 针对进行多路复用.XCLKIN 输入可以通过 XCLK 寄存器中的 XCLKIN 位被选择为 GPIO19 或 GPIO38。CLKCTL[XCLKINOFF] 位禁用此时钟输入（强制低）。如果没有使用时钟源或各自的引脚被用作 GPIO，用户应该在引导时禁用。

在更改时钟源之前，请确保目标时钟存在。如果时钟不存在, 则在切换时钟之前必须禁用该时钟源(使用 CLKCTL 寄存器)。

**表 5-14 可能的 PLL 配置模式**

| PLL MODE | REMARKS | PLLSTS[DIVSEL] | CLKIN AND SYSCLKOUT |
|---|---|---|---|
| 关闭PLL | 由在PLLSTS寄存器中设置PLLOFF位的用户调用。PLL块在此模式下被禁用。这有助于降低系统噪声和低功耗操作.在进入此模式之前，必须首先将PLLCR寄存器设置为0x0000（PLL旁路）。CPU时钟（CLKIN）直接来自于X1/X2、X1或XCLKIN上的输入时钟。 | 0, 1<br>2<br>3 | OSCCLK/4<br>OSCCLK/2<br>OSCCLK/1 |
| 旁路PLL | PLL旁路是上电或外部复位（$\overline{XRS}$）后的默认PLL配置。当PLLCR寄存器设置为0x0000，或在修改PLLCR寄存器后将PLL锁定为一个新的频率时，将选择此模式。在此模式下，旁路PLL，但PLL没有关闭。 | 0, 1<br>2<br>3 | OSCCLK/4<br>OSCCLK/2<br>OSCCLK/1 |
| 使能PLL | 这是通过将一个非零值 n 写入PLLCR寄存器来实现的。<br>在写入PLLCR时，设备将切换到PLL旁路模式，直到PLL锁定 | 0, 1<br>2<br>3 | OSCCLK * n/4<br>OSCCLK * n/2<br>OSCCLK * n/1 |

## 5.6.4. USB 和 HRCAP PLL 模块 (PLL2)

除了主系统 PLL 之外，这些设备还包含第二个 PLL（PLL2），它可用于计时 USB 和 HRCAP 外设。PLL 支持 1 到 15 的乘数，且对其输出有一个固定的 2 分频。

通过在 PLL2CTL 寄存器中适当地修改 PLL2CLKSRCSEL 位，可以从以下三个源中时钟 PLL2：

INTOSC1 (内部无引脚振荡器 1)：这是片载内部振荡器 1，并提供了一个 10MHz 的时钟。如果用作 HRCAP 的时钟源, 则应经常调用振荡器补偿例程。由于精度的要求，INTOSC1 不能用作 USB 的时钟源。

晶振/谐振器操作：振荡器(crystal) 能够使用连接在设备上的外部晶体或谐振器来提供时钟基础。晶体或谐振器连接到 X1/X2 引脚。

外部时钟源操作：这种模式允许参考时钟从一个连接到 GPIO19 或 GPIO38 的外部单端时钟源中获得。在 XCLK 寄存器中的 XCLKINSEL 位应适当设置，以使所选 GPIO 能够驱动 XCLKIN。

注 意

为了正确运行 USB 模块，PLL2 应该配置为生成一个 120MHz 的时钟。将其除以 2 以产生 USB 外设所需的 60 MHz 频率。HRCAP 支持的最大时钟输入频率为 120 MHz。

### 5.6.5. 输入时钟丢失（NMI 看门狗功能）

AVP32F069 器件可以从内部零引脚振荡器（INTOSC1/INTOSC2）、片内晶体振荡器或外部时钟输入提供时钟。无论时钟源如何，在 PLL 使能和 PLL 旁路模式下，如果 PLL 的输入时钟消失，PLL 将在其输出端发出一个跛行模式时钟。该跛行模式时钟继续以 1–5 MHz 的典型频率为 CPU 和外设提供时钟。

当跛行模式被激活时，将生成一个时钟故障信号，该信号被作为 NMI 中断锁存。根据 NMIRESETSEL 位的配置方式，可以立即触发对设备的复位，或者 NMI 看门狗计数器可以在溢出时发出复位。除此之外，还设置了时钟丢失状态（MCLKSTS）位。应用程序可以使用 NMI 中断来检测输入时钟丢失并启动必要的纠正措施，例如切换到备用时钟源（如果可用）或启动系统的关断过程。

如果软件不响应时钟故障情况，NMI 看门狗会在预编程的时间间隔后触发复位。图 5-8 显示了所涉及的中断机制。

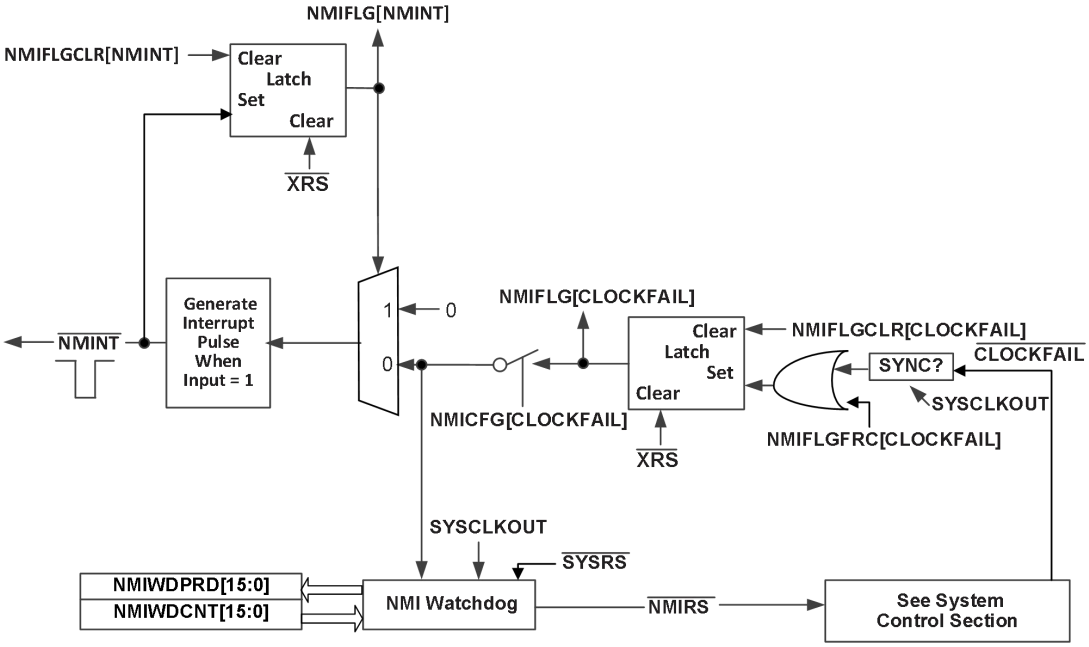
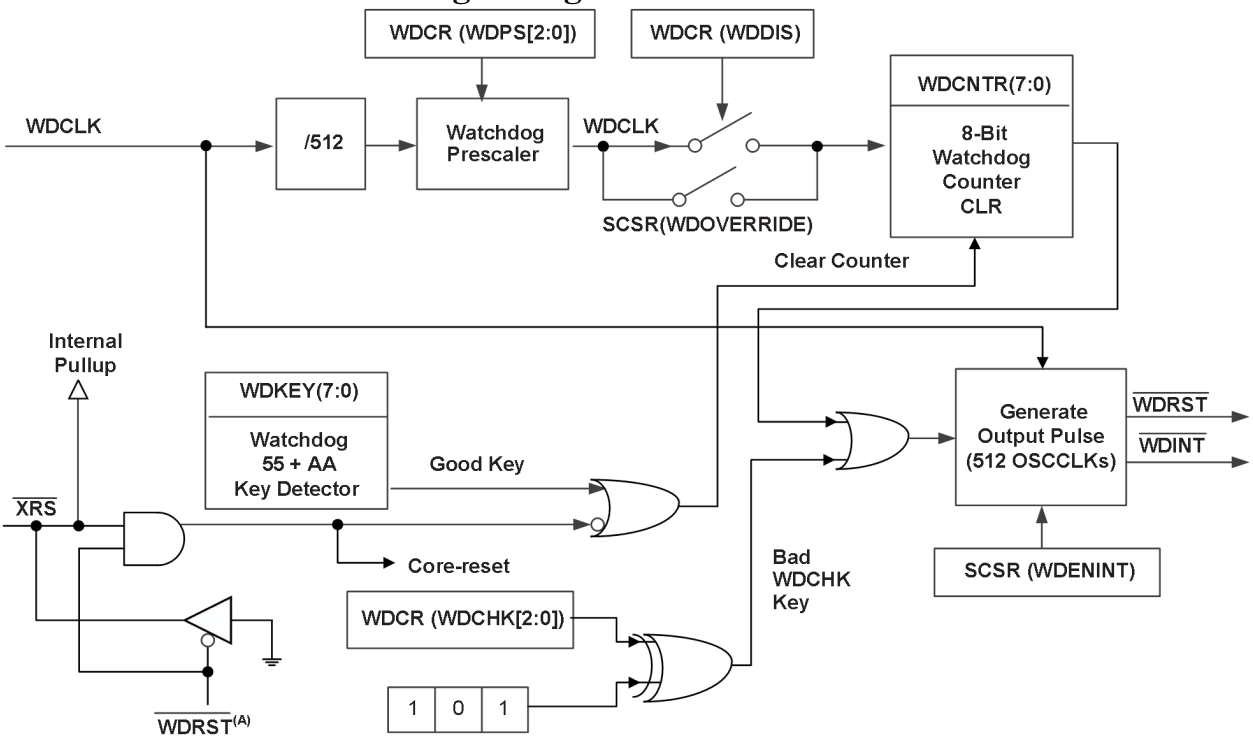



**图 5-8 NMI 看门狗**

### 5.6.6. CPU 看门狗模块

每当 8 位看门狗向上计数达到最大值时，该模块就会产生一个输出脉冲，即 512 个振荡器时钟宽（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 序列写入复位看门狗计数器的看门狗密钥寄存器。图 5-9 所示为看门狗模块内的功能块。




A WDRST信号在 512 个 OSCCLK 周期内被驱动为低电平。

**图 5-9 CPU 看门狗模块**

WDINT信号使看门狗能够用作从空闲/待机模式唤醒的激励源。

在待机模式下，设备上的所有外围设备都已关闭。唯一能够正常工作的外设是 CPU 看门狗。此模块将运行 OSCCLK。WDINT信号被馈送到 LPM 块，以便它可以将设备从待机状态唤醒（如果启用）。有关详细信息，请参阅第 5.7 节。

在 IDLE 模式下，WDINT信号可以通过 PIE 向 CPU 生成中断，以使 CPU 退出空闲模式。

在 HALT 模式下，CPU 看门狗可用于通过设备复位来唤醒设备。

通常，当输入时钟存在时，CPU 看门狗计数器会递减以启动 CPU 看门狗复位或 WDINT 中断。但是，当外部输入时钟发生故障时，CPU 看门狗计数器将停止递减（即，看门狗计数器不会随跛行模式时钟而变化）。

## 5.7. 低功耗模式

表 5-15 总结了各种低功耗模式。

**表 5-15 低功耗模式**

| MODE | LPMCR0(1:0) | OSCCLK | CLKIN | SYSCLKOUT | EXIT(1) |
|---|---|---|---|---|---|
| IDLE | 00 | 开启 | 开启 | 开启 | 复位, CPU-看门狗 中断,<br>任何使能的中断 |
| STANDBY | 01 | 开启<br>(CPU看门狗仍在运行) | 关闭 | 关闭 | 复位, CPU-看门狗 中断,<br>通用输入输出端口 A 信号，调试器(2) |
| HALT(3) | 1X | 关闭<br>(片载晶体振荡器和PLL关闭，零引脚振荡器和<br>CPU-看门狗状态取决于用户代码) | 关闭 | 关闭 | 复位, CPU-看门狗 ，<br>通用输入输出端口A信号，调试器(2) |

(1) EXIT 列出了哪些信号或在什么条件下退出低功耗模式。任何信号上的低电平信号都会退出低功耗状态。该信号必须保持足够长的低电平，以便器件能够识别中断。否则，低功耗模式将不会退出，设备将返回到指示的低功耗模式。

(2) 即使 CPU 时钟 (CLKIN) 关闭，JTAG 端口仍然可以正常工作。

(3) WDCLK 必须处于有效状态，设备才能进入 HALT 模式。

低功耗模式的工作原理如下:

| | |
|---|---|
| IDLE 模式: | 此模式由处理器识别的任何已启用中断退出。LPM 块在此模式下不执行任何任务，只要 LPMCR0 (LPM) 位设置为 0，0。 |
| STANDBY 模式: | 任何 GPIO 端口 A 信号 (GPIO[31：0]) 都可以将设备从待机模式中唤醒。用户必须在 GPIOLPMSEL 中选择一些信号将设备唤醒。所选信号在唤醒器件之前也由 OSCCLK 进行限定。OSCCLK 的数量在 LPMCR0 寄存器中指定。 |
| HALT 模式: | CPU 看门狗、$\overline{XRS}$ 和任何 GPIO 端口 A 信号 (GPIO[31：0]) 都可以将设备从 HALT 模式唤醒。用户在 GPIOLPMSEL 寄存器中选择信号。 |

**注 意**

低功耗模式不会影响输出引脚（包括 PWM 引脚）的状态。执行完 IDLE 指令后，它们将处于代码让其所处的状态。

## 5.8. 中断

图 5-10 所示为外部和 PIE 中断源。

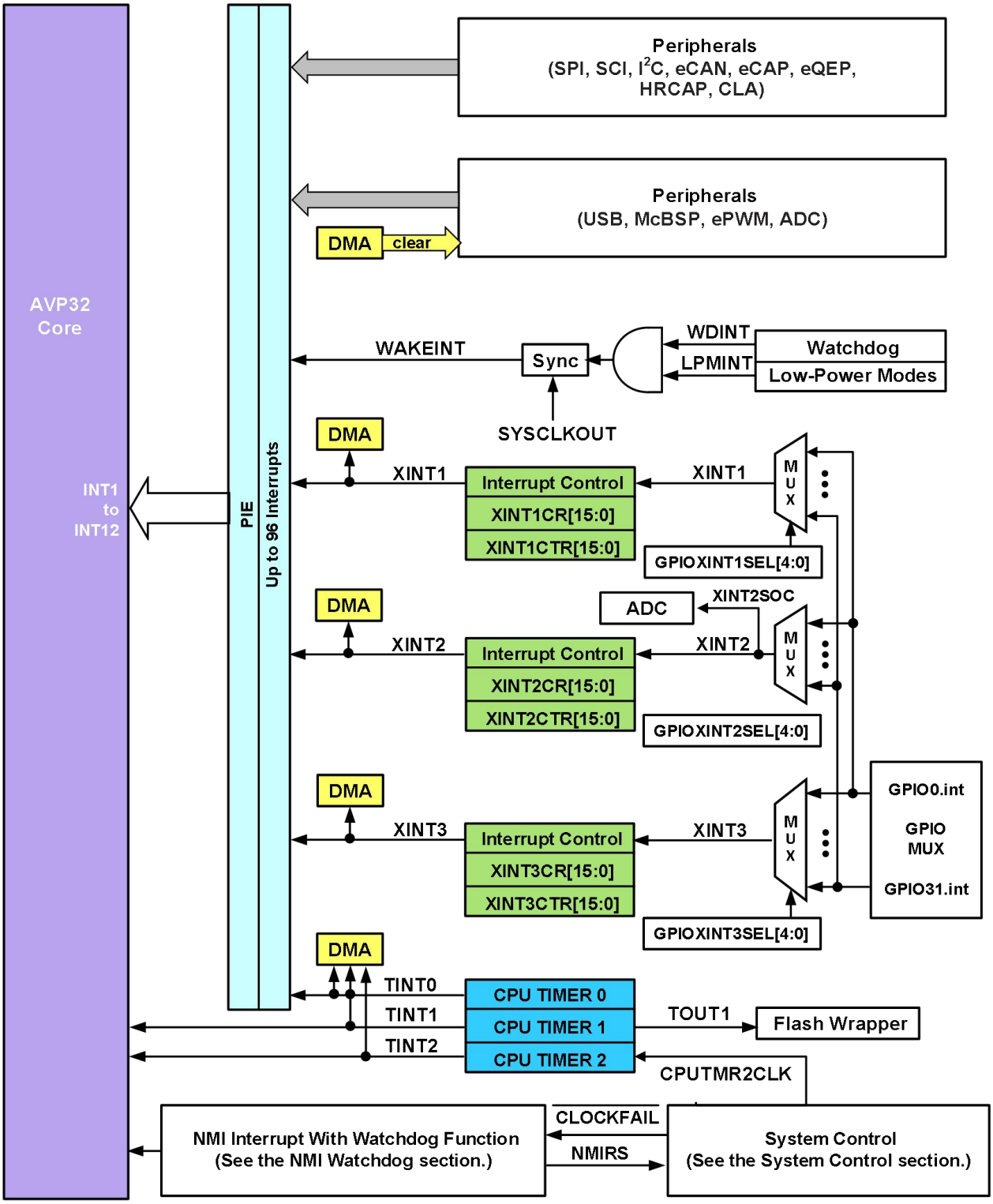



**图 5-10 外部和 PIE 中断源**

8 个 PIE 中断对应一个 CPU 中断。总共 12 个 CPU 中断组，每组包括 8 个中断，共有 96 个可用的中断。表 5-16 为 AVP32F069 器件的中断向量表。

TRAP #VectorNumber 指令发送程序控制到指定向量的中断服务程序。TRAP #0 指令尝试发送程序控制到复位向量指向的地址。然而，PIE 向量表不包含复位向量。因此，当 PIE 使能时不用 TRAP #0 指令。若使用将导致未定义的行为。

当 PIE 使能时，TRAP #1 到 TRAP #12 指令将发送程序控制到 PIE 组的对应第一个向量的 ISR。例如：TRAP #1 指令从 INT1.1 抓取向量，TRAP #2 指令从 INT2.1 抓取向量，以此类推。

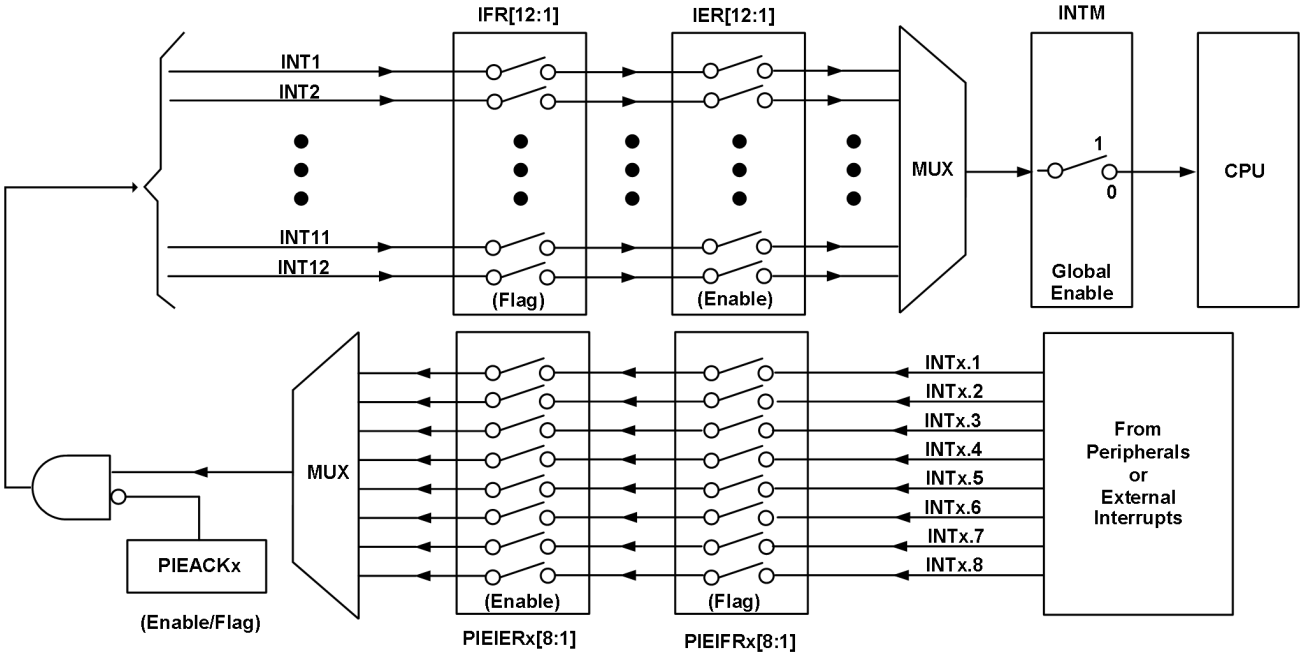



**图 5-11 使用 PIE 的中断复用**

在 96 个可用中断中，有些中断没有使用。这些中断是为将来的设备保留的。如果在 PIEIFRx 级别启用了这些中断，则可以将其用作软件中断，前提是外设未使用组内的任何中断。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，当保留的中断可用作软件中断时，有两种安全情况：

1. 组内没有外设断言中断。
2. 无外设中断被关联到组(例如，PIE 组 7)。

## 表 5-16 PIE 外设中断向量表

| | INTx.8 | INTx.7 | INTx.6 | INTx.5 | INTx.4 | INTx.3 | INTx.2 | INTx.1 |
|---|---|---|---|---|---|---|---|---|
| **INT1.y** | WAKEINT<br>(LPM/WD)<br>0xD4E | TINT0<br>(TIMER 0)<br>0xD4C | ADCINT9<br>(ADC)<br>0xD4A | XINT2<br>Ext. int. 2<br>0xD48 | XINT1<br>Ext. int. 1<br>0xD46 | Reserved<br>–<br>0xD44 | ADCINT2<br>(ADC)<br>0xD42 | ADCINT1<br>(ADC)<br>0xD40 |
| **INT2.y** | EPWM8_TZINT<br>(ePWM8)<br>0xD5E | EPWM7_TZINT<br>(ePWM7)<br>0xD5C | EPWM6_TZINT<br>(ePWM6)<br>0xD5A | EPWM5_TZINT<br>(ePWM5)<br>0xD58 | EPWM4_TZINT<br>(ePWM4)<br>0xD56 | EPWM3_TZINT<br>(ePWM3)<br>0xD54 | EPWM2_TZINT<br>(ePWM2)<br>0xD52 | EPWM1_TZINT<br>(ePWM1)<br>0xD50 |
| **INT3.y** | EPWM8_INT<br>(ePWM8)<br>0xD6E | EPWM7_INT<br>(ePWM7)<br>0xD6C | EPWM6_INT<br>(ePWM6)<br>0xD6A | EPWM5_INT<br>(ePWM5)<br>0xD68 | EPWM4_INT<br>(ePWM4)<br>0xD66 | EPWM3_INT<br>(ePWM3)<br>0xD64 | EPWM2_INT<br>(ePWM2)<br>0xD62 | EPWM1_INT<br>(ePWM1)<br>0xD60 |
| **INT4.y** | HRCAP2_INT<br>(HRCAP2)<br>0xD7E | HRCAP1_INT<br>(HRCAP1)<br>0xD7C | Reserved<br>–<br>0xD7A | Reserved<br>–<br>0xD78 | Reserved<br>–<br>0xD76 | ECAP3_INT<br>(eCAP3)<br>0xD74 | ECAP2_INT<br>(eCAP2)<br>0xD72 | ECAP1_INT<br>(eCAP1)<br>0xD70 |
| **INT5.y** | USB0_INT<br>(USB0)<br>0xD8E | Reserved<br>–<br>0xD8C | Reserved<br>–<br>0xD8A | HRCAP4_INT<br>(HRCAP4)<br>0xD88 | HRCAP3_INT<br>(HRCAP3)<br>0xD86 | Reserved<br>–<br>0xD84 | EQEP2_INT<br>(eQEP2)<br>0xD82 | EQEP1_INT<br>(eQEP1)<br>0xD80 |
| **INT6.y** | Reserved<br>–<br>0xD9E | Reserved<br>–<br>0xD9C | MXINTA<br>(McBSP-A)<br>0xD9A | MRINTA<br>(McBSP-A)<br>0xD98 | SPITXINTB<br>(SPI-B)<br>0xD96 | SPIRXINTB<br>(SPI-B)<br>0xD94 | SPITXINTA<br>(SPI-A)<br>0xD92 | SPIRXINTA<br>(SPI-A)<br>0xD90 |
| **INT7.y** | Reserved<br>–<br>0xDAE | Reserved<br>–<br>0xDAC | DINTCH6<br>(DMA)<br>0xDAA | DINTCH5<br>(DMA)<br>0xDA8 | DINTCH4<br>(DMA)<br>0xDA6 | DINTCH3<br>(DMA)<br>0xDA4 | DINTCH2<br>(DMA)<br>0xDA2 | DINTCH1<br>(DMA)<br>0xDA0 |
| **INT8.y** | Reserved<br>–<br>0xDBE | Reserved<br>–<br>0xDBC | Reserved<br>–<br>0xDBA | Reserved<br>–<br>0xDB8 | Reserved<br>–<br>0xDB6 | Reserved<br>–<br>0xDB4 | I2CINT2A<br>(I2C-A)<br>0xDB2 | I2CINT1A<br>(I2C-A)<br>0xDB0 |
| **INT9.y** | Reserved<br>–<br>0xDCE | Reserved<br>–<br>0xDCC | ECAN1_INTA<br>(CAN-A)<br>0xDCA | ECAN0_INTA<br>(CAN-A)<br>0xDC8 | SCITXINTB<br>(SCI-B)<br>0xDC6 | SCIRXINTB<br>(SCI-B)<br>0xDC4 | SCITXINTA<br>(SCI-A)<br>0xDC2 | SCIRXINTA<br>(SCI-A)<br>0xDC0 |
| **INT10.y** | ADCINT8<br>(ADC)<br>0xDDE | ADCINT7<br>(ADC)<br>0xDDC | ADCINT6<br>(ADC)<br>0xDDA | ADCINT5<br>(ADC)<br>0xDD8 | ADCINT4<br>(ADC)<br>0xDD6 | ADCINT3<br>(ADC)<br>0xDD4 | ADCINT2<br>(ADC)<br>0xDD2 | ADCINT1<br>(ADC)<br>0xDD0 |
| **INT11.y** | CLA1_INT8<br>(CLA)<br>0xDEE | CLA1_INT7<br>(CLA)<br>0xDEC | CLA1_INT6<br>(CLA)<br>0xDEA | CLA1_INT5<br>(CLA)<br>0xDE8 | CLA1_INT4<br>(CLA)<br>0xDE6 | CLA1_INT3<br>(CLA)<br>0xDE4 | CLA1_INT2<br>(CLA)<br>0xDE2 | CLA1_INT1<br>(CLA)<br>0xDE0 |
| **INT12.y** | LUF<br>(CLA)<br>0xDFE | LVF<br>(CLA)<br>0xDFC | Reserved<br>–<br>0xDFA | Reserved<br>–<br>0xDF8 | Reserved<br>–<br>0xDF6 | Reserved<br>–<br>0xDF4 | Reserved<br>–<br>0xDF2 | XINT3<br>Ext. Int. 3<br>0xDF0 |

## 表 5-17 PIE 配置和控制寄存器

| 名称 | 地址 | 大小 (×16) | 描述(1) |
|---|---|---|---|
| PIECTRL | 0x0CE0 | 1 | PIE，控制寄存器 |
| PIEACK | 0x0CE1 | 1 | PIE，响应寄存器 |
| PIEIER1 | 0x0CE2 | 1 | PIE, INT1 组使能寄存器 |
| PIEIFR1 | 0x0CE3 | 1 | PIE, INT1 组标志寄存器 |
| PIEIER2 | 0x0CE4 | 1 | PIE, INT2 组使能寄存器 |
| PIEIFR2 | 0x0CE5 | 1 | PIE, INT2 组标志寄存器 |
| PIEIER3 | 0x0CE6 | 1 | PIE, INT3 组使能寄存器 |
| PIEIFR3 | 0x0CE7 | 1 | PIE, INT3 组标志寄存器 |
| PIEIER4 | 0x0CE8 | 1 | PIE, INT4 组使能寄存器 |
| PIEIFR4 | 0x0CE9 | 1 | PIE, INT4 组标志寄存器 |
| PIEIER5 | 0x0CEA | 1 | PIE, INT5 组使能寄存器 |
| PIEIFR5 | 0x0CEB | 1 | PIE, INT5 组标志寄存器 |
| PIEIER6 | 0x0CEC | 1 | PIE, INT6 组使能寄存器 |
| PIEIFR6 | 0x0CED | 1 | PIE, INT6 组标志寄存器 |

| | | | |
|---|---|---|---|
| PIEIER7 | 0x0CEE | 1 | PIE, INT7 组使能寄存器 |
| PIEIFR7 | 0x0CEF | 1 | PIE, INT7 组标志寄存器 |
| PIEIER8 | 0x0CF0 | 1 | PIE, INT8 组使能寄存器 |
| PIEIFR8 | 0x0CF1 | 1 | PIE, INT8 组标志寄存器 |
| PIEIER9 | 0x0CF2 | 1 | PIE, INT9 组使能寄存器 |
| PIEIFR9 | 0x0CF3 | 1 | PIE, INT9 组标志寄存器 |
| PIEIER10 | 0x0CF4 | 1 | PIE, INT10 组使能寄存器 |
| PIEIFR10 | 0x0CF5 | 1 | PIE, INT10 组标志寄存器 |
| PIEIER11 | 0x0CF6 | 1 | PIE, INT11 组使能寄存器 |
| PIEIFR11 | 0x0CF7 | 1 | PIE, INT11 组标志寄存器 |
| PIEIER12 | 0x0CF8 | 1 | PIE, INT12 组使能寄存器 |
| PIEIFR12 | 0x0CF9 | 1 | PIE, INT12 组标志寄存器 |
| Reserved | 0x0CFA –0x0CFF | 6 | Reserved |

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

## 5.8.1. 外部中断

**表 5-18 外部中断寄存器**

| 名称 | 地址 | 大小 (×16) | 描述 |
|---|---|---|---|
| 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 计数寄存器 |
| XINT2CTR | 0x00 7079 | 1 | XINT2 计数寄存器 |
| XINT3CTR | 0x00 707A | 1 | XINT3 计数寄存器 |

可以使用正、负或正负边沿启用、禁用或限定每个外部中断。

### 5.8.1.1. 外部中断电气数据/时序

#### 5.8.1.1.1.外部中断时序要求

| | | 最小值(1) | 最大值 | 单位 |
|---|---|---|---|---|
| $t_{w(INT)}$ (2) 脉冲期间, INT输入低/高 | 同步 | $1t_{c(SCO)}$ | | 周期 |
| | 带限制 | $1t_{c(SCO)} + t_{w(IQSW)}$ | | 周期 |

(1) 此定时适用于为 ADCSOC 功能配置的 GPIO 引脚。

#### 5.8.1.1.2.外部中断开关特性

在建议运行条件下测得（除非另有说明）(1)

| 参数 | | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|
| $t_{d(INT)}$ | 延迟时间, 中断低/高 到中断向量提取 | | $t_{w(IQSW)} + 12t_{c(SCO)}$ | 周期 |

(1) 关于输入限定参数的解释请查阅第 5.9.14.1.3.1 节。

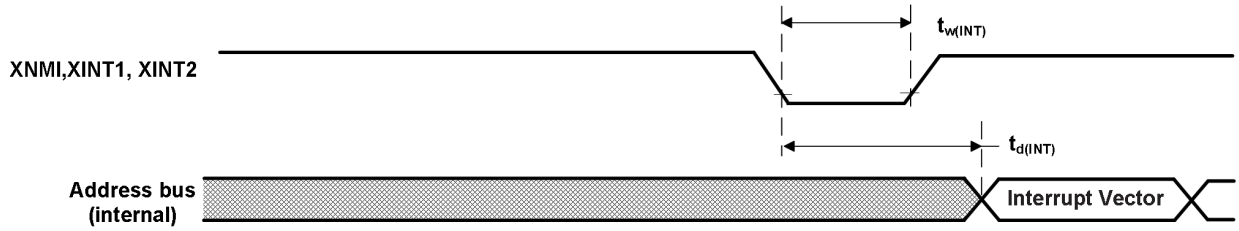



**图 5-12 外部中断时序**

# 5.9. 外围设备

## 5.9.1. CLA 模块概述

CLA 可进行并行处理的特性扩展了 CPU 的功能。使用 CLA 控制实时性强的控制环路可以降低 ADC 采样的输出延迟。因此，CLA 可实现更快的系统响应和更高频率的控制回路。通过将 CLA 用于时间关键任务的处理，那么主 CPU 则可以同时执行其他系统任务和通信功能。CLA 的主要特性列表如下。

- 与主 CPU 时钟 SYSCLKOUT 工作于同一时钟频率
- CLA 独立的架构设计使得其可以独立于主 CPU 工作

- 完整的总线架构:
  - 程序地址总线和程序读总线
  - 数据读地址总线，数据读总线，数据写地址总线，数据写总线
- 独立的 8 级流水线结构
- 12 位程序计数器 (MPC)
- 4 个 32 位的结果寄存器 (MR0 到 MR3)
- 2 个 16 位的辅助寄存器 (MAR0, MAR1)
- 状态寄存器 (MSTF)

- 指令集特性:
- 支持 IEEE 单精度（32 位）浮点算术运算
- 支持并行访存的浮点算术操作
- 加减运算可与浮点乘法操作并行执行
- 1/X 和 1/sqrt (X) 估值
- 数据类型转换
- 条件分支和调用
- 数据访存操作

- CLA 程序代码包括高达 8 个任务或者中断服务子程序.
- 每个任务的起始地址由 MVECT 寄存器指定
- 只要 CLA 程序空间可以存放的下，则不对任务的代码大小设限。
- 同一时间只能有一个任务处于执行状态，任务不可以嵌套。
- 一旦任务执行完成，则会在 PIE 产生一个与该任务相关的中断标志。
- 当前任务执行结束后，已挂起的下一个高优先级任务则会自动启动。

- 任务触发机制:
- AVP32 CPU 通过 IACK 指令
- 任务 1 到任务 7：对应的 ADC、ePWM、eQEP 或 eCAP 模块中断。例如:
  - 任务 1：ADCINT1 或 EPWM1_INT
  - 任务 2：ADCINT2 或 EPWM2_INT

- 任务 4：ADCINT4 或 EPWM4_INT 或 EQEPx_INT 或 ECAPx_INT
- 任务 7：ADCINT7 或 EPWM7_INT 或 EQEPx_INT 或 ECAPx_INT

- 任务 8：ADCINT8 或 CPU Timer 0 或 EQEPx_INT 或 ECAPx_INT。

• 内存和共享外设：

- 两个专用消息 RAM，用于 CPU 和 CLA 之间的通信。
- 主 AVP32 CPU 可以将 CLA 程序和数据存储器映射到主 CPU 空间或 CLA 空间。
- CLA 可直接访问 ADC 结果寄存器、比较寄存器、eCAP、eQEP、和 Epwm+HRPWM 的寄存器。

图 5-13 为 CLA 模块功能框图。表 5-19 所列为 CLA 控制寄存器。

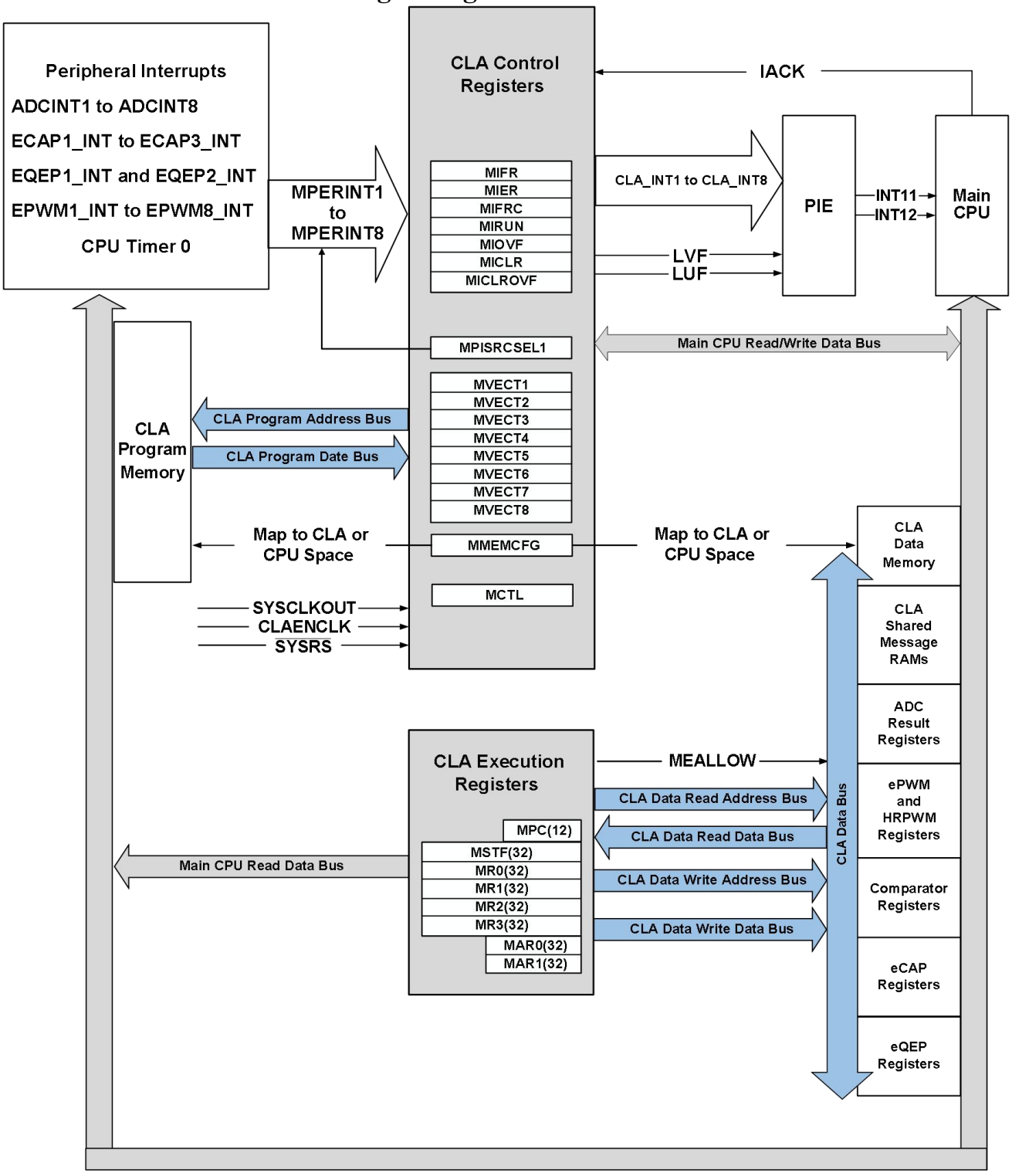

Peripheral Interrupts
ADCINT1 to ADCINT8
ECAP1_INT to ECAP3_INT
EQEP1_INT and EQEP2_INT
EPWM1_INT to EPWM8_INT
CPU Timer 0
MPERINT1 to MPERINT8
CLA Control Registers
MIFR
MIER
MIFRC
MIRUN
MIOVF
MICLR
MICLROVF
MPISRCSEL1
MVECT1
MVECT2
MVECT3
MVECT4
MVECT5
MVECT6
MVECT7
MVECT8
MMEMCFG
MCTL
IACK
CLA_INT1 to CLA_INT8
LVF
LUF
PIE
INT11
INT12
Main CPU
Main CPU Read/Write Data Bus
CLA Program Memory
CLA Program Address Bus
CLA Program Date Bus
Map to CLA or CPU Space
Map to CLA or CPU Space
SYSCLKOUT
CLAENCLK
SYSRS
CLA Data Memory
CLA Shared Message RAMs
ADC Result Registers
ePWM and HRPWM Registers
Comparator Registers
eCAP Registers
eQEP Registers
CLA Data Bus
CLA Execution Registers
MEALLOW
CLA Data Read Address Bus
CLA Data Read Data Bus
CLA Data Write Address Bus
CLA Data Write Data Bus
MPC(12)
MSTF(32)
MR0(32)
MR1(32)
MR2(32)
MR3(32)
MAR0(32)
MAR1(32)
Main CPU Read Data Bus

图 5-13 CLA 框图

**表 5-19 CLA 控制寄存器**

| 寄存器 | CLA1 地址 | SIZE (×16) | 允许保护 | 描述 |
|---|---|---|---|---|
| MVECT1 | 0x1400 | 1 | Yes | CLA Interrupt/Task 1 Start Address |
| MVECT2 | 0x1401 | 1 | Yes | CLA Interrupt/Task 2 Start Address |
| MVECT3 | 0x1402 | 1 | Yes | CLA Interrupt/Task 3 Start Address |
| MVECT4 | 0x1403 | 1 | Yes | CLA Interrupt/Task 4 Start Address |
| MVECT5 | 0x1404 | 1 | Yes | CLA Interrupt/Task 5 Start Address |
| MVECT6 | 0x1405 | 1 | Yes | CLA Interrupt/Task 6 Start Address |
| MVECT7 | 0x1406 | 1 | Yes | CLA Interrupt/Task 7 Start Address |
| MVECT8 | 0x1407 | 1 | Yes | CLA Interrupt/Task 8 Start Address |
| MCTL | 0x1410 | 1 | Yes | CLA Control Register |
| MMEMCFG | 0x1411 | 1 | Yes | CLA Memory Configure Register |
| MPISRCSEL1 | 0x1414 | 2 | Yes | Peripheral Interrupt Source Select Register 1 |
| MIFR | 0x1420 | 1 | Yes | Interrupt Flag Register |
| MIOVF | 0x1421 | 1 | Yes | Interrupt Overflow Register |
| MIFRC | 0x1422 | 1 | Yes | Interrupt Force Register |
| MICLR | 0x1423 | 1 | Yes | Interrupt Clear Register |
| MICLROVF | 0x1424 | 1 | Yes | Interrupt Overflow Clear Register |
| MIER | 0x1425 | 1 | Yes | Interrupt Enable Register |
| MIRUN | 0x1426 | 1 | Yes | Interrupt RUN Register |
| MIPCTL | 0x1427 | 1 | Yes | Interrupt Priority Control Register |
| MPC(2) | 0x1428 | 1 | – | CLA Program Counter |
| MAR0(2) | 0x142A | 1 | – | CLA Aux Register 0 |
| MAR1(2) | 0x142B | 1 | – | CLA Aux Register 1 |
| MSTF(2) | 0x142E | 2 | – | CLA STF Register |
| MR0(2) | 0x1430 | 2 | – | CLA R0H Register |
| MR1(2) | 0x1434 | 2 | – | CLA R1H Register |
| MR2(2) | 0x1438 | 2 | – | CLA R2H Register |
| MR3(2) | 0x143C | 2 | – | CLA R3H Register |

(1) 此表格所有寄存器均受 CSM 保护。

(2) 主 CPU 调试目的下仅有只读访问权限。主 CPU 不能执行 CPU 或调试器写入这些寄存器操作。

**表 5-20 CLA Message RAM**

| 地址范围 | 大小 (×16) | 描述 |
|---|---|---|
| 0x1480 – 0x14FF | 128 | CLA to CPU Message RAM |
| 0x1500 – 0x157F | 128 | CPU to CLA Message RAM |

## 5.9.2. 模拟模块

实现的 12 位 ADC 内核的时序与 32F0x 和 32F33x 器件上使用的 12 位 ADC 不同。对 ADC 的数字电路（在本文档中称为“包装器”）进行了修改，以纳入新的时序和其他增强功能来改善转换开始的控制时序。图 5-14 显示了模拟模块与 AVP32F069 系统其余部分的交互功能。

| 56-Pin 64-Pin | 80-Pin |
|---|---|
| VDDA | VDDA |
| VREFLO Tied To VSSA | VSSA |
| | VREFLO |
| VREFHI Tied To A0 | VREFHI |
| | A0 |
| A1 | A1 |
| A2 | A2 |
| A3 | A3 |
| A4 | A4 |
| | A5 |
| A6 | A6 |
| A7 | A7 |
| B0 | B0 |
| B1 | B1 |
| B2 | B2 |
| B3 | B3 |
| B4 | B4 |
| | B5 |
| B6 | B6 |
| B7 | B7 |

Signal Pinout

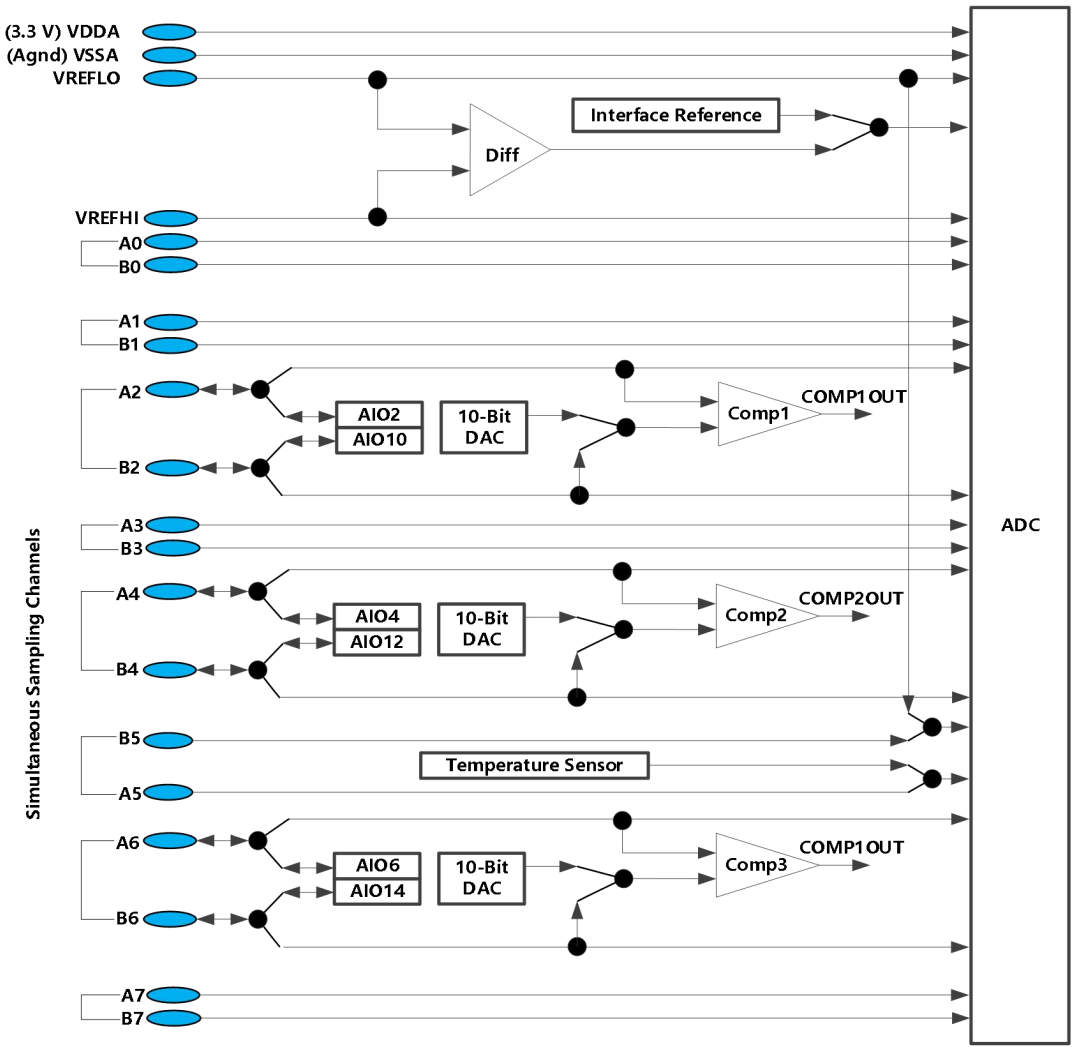



**图 5-14 模拟引脚配置**

### 5.9.2.1. 模数转换器（ADC）

#### 5.9.2.1.1.功能

该 ADC 的内核包含一个 12 位转换器，带两个采样保持电路。采样保持电路可以按顺序或同步采样的模式运行。两个采样保持电路由总共多达 16 个模拟输入通道为其提供输入信号。该转换器可配置为使用内部带隙基准电压源运行，以创建基于真实电压的转换；或可配置为与一对外部基准电压源（$V_{REFHI}$/$V_{REFLO}$）一起运行，以创建基于比率的转换。

相比于以前的 ADC 类型，该 ADC 不是基于时序控制器的。用户可以轻松地从单个触发器创建一系列转换。但是，基本操作原则以称为 SOC 或“转换开始”的单个转换的配置为中心。

ADC 模块的功能包括：

• 12 位 ADC 内核，内置双采样保持电路 （S/H）

• 同步采样或顺序采样模式

• 全范围模拟输入：基于 0 V 至 3.3 V 真实电压的固定转换公式，或基于外部电压基准电压源（$V_{REFHI}$ /$V_{REFLO}$）的比例转换公式。输入模拟电压的数字值由下式得出：

– 内部基准电压源 （$V_{REFLO}$ = $V_{SSA}$ .使用内部或外部参考模式时，$V_{REFHI}$ 不得超过 $V_{DDA}$ 。

Digital Value = 0, when input ≤ 0 V

Digital Value = $4096x\frac{\text{Input Analog Voltage}-V_{REFLO}}{3.3}$ when 0 V ≤ input ≤ 3.3 V

Digital Value = 4095, when input ≥ 3.3 V

– 外部基准电压源（$V_{REFHI}$ /$V_{REFLO}$ 连接到外部基准电压源。 使用内部或外部参考模式时，$V_{REFHI}$ 不得超过 $V_{DDA}$。

Digital Value = 0, when input ≤ 0 V

Digital Value =$4096x\frac{\text{Input Analog Voltage}-V_{REFLO}}{V_{REFHI}-V_{REFLO}}$ when 0 V <input < $V_{REFHI}$

Digital Value = 4095, when input ≥ $V_{REFHI}$

• 多达 16 个复用输入通道

• 16 个可单独配置触发源、采样窗口和通道的 SOC

• 16 个结果寄存器（可单独寻址）用于存储转化值

• 多个触发源

– S/W 软件快速启动

– ePWM 1–8

– GPIO XINT2

– CPU 定时器 0，CPU 定时器 1，CPU 定时器 2

– ADCINT1，ADCINT2

• 9 个可在任意转换后灵活配置中断请求的 PIE 中断。

**表 5-21 ADC 配置和控制寄存器**

| 寄存器名称 | 地址 | 大小（x16） | EALLOW 保护 | 说明 |
|---|---|---|---|---|
| ADCCTRL1 | 0x7100 | 1 | 支持 | ADC控制寄存器1 |
| ADCCTRL2 | 0x7101 | 1 | 支持 | ADC控制寄存器2 |
| ADCINTFLG | 0x7104 | 1 | 否 | ADC中断标志寄存器 |
| ADCINTFLGCLR | 0x7105 | 1 | 否 | ADC中断标志清除寄存器 |
| ADCINTOVF | 0x7106 | 1 | 否 | ADC中断溢出寄存器 |
| ADCINTOVFCLR | 0x7107 | 1 | 否 | ADC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（中断10保留） |
| SOCPRICTL | 0x7110 | 1 | 支持 | SOC 优先级控制寄存器 |
| ADCSAMPLEMODE | 0x7112 | 1 | 支持 | ADC采样模式寄存器 |
| ADCINTSOCSEL1 | 0x7114 | 1 | 支持 | ADC中断 SOC 选择寄存器1（8个通道） |
| ADCINTSOCSEL2 | 0x7115 | 1 | 支持 | ADC中断 SOC 选择寄存器2（8 个通道） |
| ADCSOCFLG1 | 0x7118 | 1 | 否 | ADC SOC 标志寄存器1（16 个通道） |
| ADCSOCFRC1 | 0x711A | 1 | 否 | ADC SOC 强制寄存器1（16 个通道） |
| ADCSOCOVF1 | 0x711C | 1 | 否 | ADC SOC 溢出寄存器1（16 个通道） |
| ADCSOCOVFCLR1 | 0x711E | 1 | 否 | ADC SOC 溢出清除寄存器1（16 个通道） |
| ADCSOC0CTL 至 ADCSOC15CTL | 0x7120 - 0x712F | 1 | 支持 | ADC SOC0 -- ADC SOC15 控制寄存器 |
| ADCREFTRIM | 0x7140 | 1 | 支持 | ADC参考修调寄存器 |
| ADCOFFTRIM | 0x7141 | 1 | 支持 | ADC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 | ADC修订版本寄存器 |

**表 5-22 ADC 结果寄存器（映射到 PF0）**

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

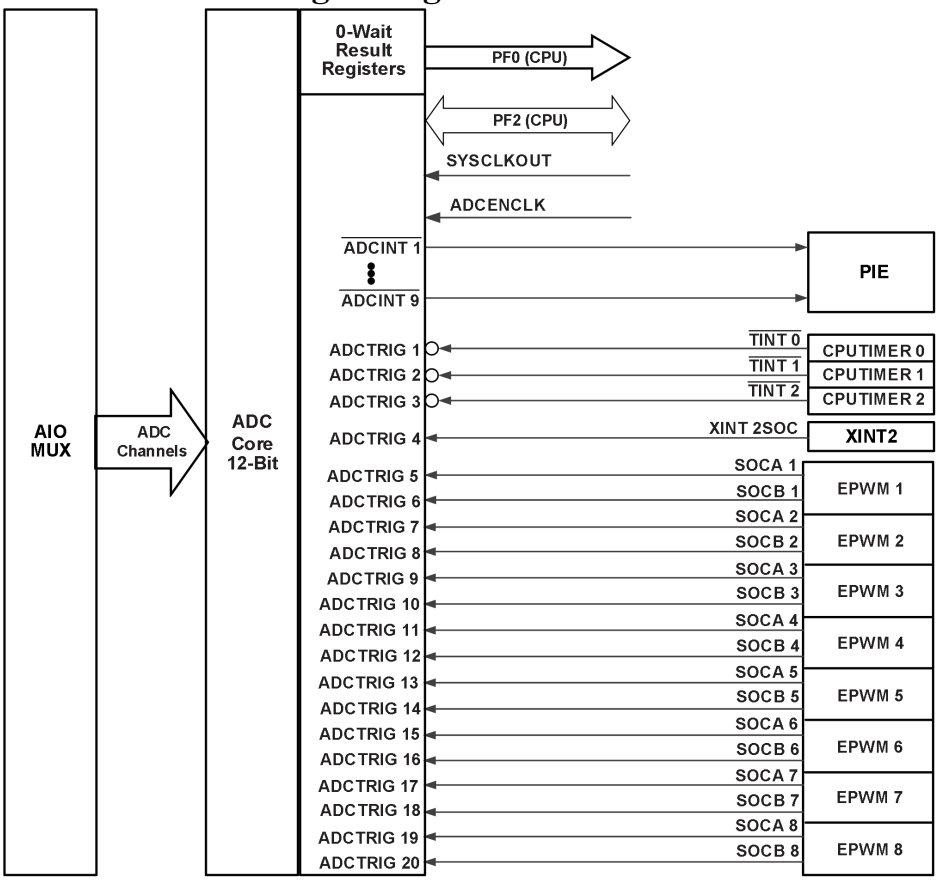



**图 5-15 模拟引脚配置**

未使用 ADC 时的 ADC 连接

我司建议保留模拟电源引脚的连接，即使未使用 ADC 也是如此。以下是 ADC 引脚的连接方式摘要（如果 ADC 未在应用中使用）：

- $V_{DDA}$ – 连接到 $V_{DDIO}$
- $V_{SSA}$ – 连接到 $V_{SS}$
- $V_{REFLO}$ – 连接到 $V_{SS}$
- ADCINAn， ADCINBn， $V_{REFHI}$ – 连接到 $V_{SSA}$

当 ADC 模块用于应用时，未使用的 ADC 输入引脚应连接到模拟地（$V_{SSA}$）。

注意

我司建议将未使用的 ADCIN 引脚（通过 AIO 功能多路复用）通过一个 1kΩ 电阻接地。此建议旨在防止任何无意中软件激活 AIO 输出逻辑高电平直接驱动至地;这种情况可能会因超过 IOH 绝对最大值而导致永久性器件损坏。

不使用 ADC 时，请确保 ADC 模块的时钟未开启，以实现节能。

#### 5.9.2.1.2. ADC开始转换的电气数据/时序

##### 5.9.2.1.2.1. 外部 ADC 转换启动的开关特性

在建议的运行条件下（除非另有说明）

| 参数 | 最小值 | 最大值 | 单位 |
|---|---|---|---|
| $t_{w(ADCSOCL)}$ 脉冲持续时间，$\overline{ADCSOCxO}$ 低电平的时间 | $32t_{c(HCO)}$ | | 周期 |

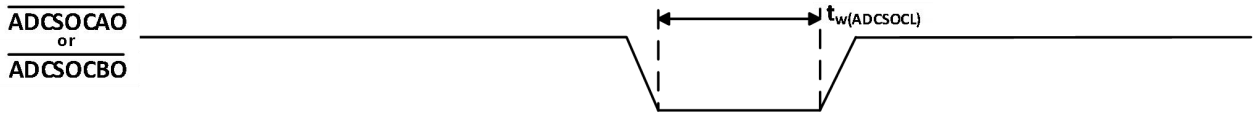



**图 5-16 $\overline{ADCSOCAO}$ 或 $\overline{ADCSOCBO}$ 时序**

#### 5.9.2.1.3.片载模数转换器（ADC）电气数据/时序

##### 5.9.2.1.3.1. *模数转换器电气特性*

| 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 | 60MHz 器件 | 0.001 | | 60 | MHz |
| 采样窗口 | AVP32F069 | 8 | | 64 | ADC时钟 |
| **精度** | | | | | |
| 在 ADC 时钟 ≤ 60MHz(1)时，INL（积分非线性） | | -7 | | 7 | 最低有效位(LSB) |
| 在 ADC 时钟 ≤ 60MHz时，DNL（微分非线性） | | -2 | | 2 | LSB |
| 偏移误差(2) | 执行单次自我校准 (3) | -20 | 0 | 20 | LSB |
| | 周期性执行自我校准(4) | -4 | 0 | 4 | |
| 通道到通道偏移变化 | | -8 | | 8 | LSB |
| 通道到通道增益变化 | | -8 | | 8 | LSB |
| **模拟输入** | | | | | |
| 带有内部基准的模拟输入电压 | | 0 | | 3.3 | V |
| 带有外部基准的模拟输入电压 | | $V_{REFLO}$ | | $V_{REFHI}$ | V |
| $V_{REFLO}$输入电压(5) | | $V_{SSA}$ | | 0.66 | V |
| $V_{REFHI}$输入电压(6) | | 2.64 | | $V_{DDA}$ | V |
| | 其中$V_{REFLO}$= $V_{SSA}$ | 1.98 | | $V_{DDA}$ | |

(1) 当 ADC 输入电压高于 VDDA 时，INL 会下降。

(2) 1 LSB 具有满量程范围（FSR）/4096 的加权值。FSR 为 3.3 V，带内部基准电压源，VREFHI - VREFLO 用于外部 基准电压源。

(3) 有关更多详细信息，请参阅《AVP32F06X 勘误表》。

(4) 定期自校准将消除系统级和温度对 ADC 零失调误差的依赖性。

(5) 在 80 引脚 PN 和 PFP 器件上，VREFLO 始终连接到 VSSA。

(6) 使用内部或外部基准电压源模式时，VREFHI 不得超过 VDDA。由于 VREFHI 与 80 引脚 PN 和 PFP 器件上的 ADCINA0 绑定，因此 ADCINA0 上的输入信号不得超过 VDDA。

#### 5.9.2.1.3.2. ADC 功率模式

| ADC 工作模式 | 条件 | $I_{DDA}$ | 单位 |
|---|---|---|---|
| 模式 A - 操作模式 | 启用ADC 时钟<br>带隙打开 (ADCBGPWD=1)<br>基准打开 (ADCREFPWD=1)<br>ADC上电 (ADCPWDN=1) | 20 | mA |
| 模式 B - 快速唤醒模式 | 启用ADC 时钟<br>带隙打开 (ADCBGPWD=1)<br>基准打开 (ADCREFPWD=1)<br>ADC断电 (ADCPWDN=0) | 4 | mA |
| 模式 C - 仅比较器可用模式 | 启用ADC 时钟<br>带隙打开 (ADCBGPWD=1)<br>基准关闭 (ADCREFPWD=0)<br>ADC断电 (ADCPWDN=0) | 1.5 | mA |
| 模式 D - 关闭模式 | 启用ADC 时钟<br>带隙关闭 (ADCBGPWD=0)<br>基准关闭 (ADCREFPWD=0)<br>ADC断电 (ADCPWDN=0) | 0.075 | mA |

#### 5.9.2.1.3.3. 内部温度传感器

##### 5.9.2.1.3.3.1. 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 | | 0.18(2)(3) | | ℃/LSB |
| $T_{OFFSET}$ 在温度传感器读数为 0°C 时的 ADC 输出值 | | 1750 | | LSB |

(1) 温度传感器斜率和失调以 ADC LSB 为单位，使用 ADC 的内部基准电压源给出。在外部基准电压模式下，必须根据外部基准电压相应地调整值。

(2) 温度传感器的输出（以 LSB 为单位）与温度运动的方向一致。 温度升高会使 ADC 值相对于初始值增加；温度降低会使 ADC 值相对于初始值减小。

(3) 本规范中考虑了 ADC 温度系数。

#### 5.9.2.1.3.4. ADC 上电控制位时序

##### 5.9.2.1.3.4.1. 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稳定所需延迟时间 | | | 1 | ms |

(1)时序保持与 ADC 模块的兼容性。在首次转换前的 $t_{d(PWD)}$ ms，AVP32F069 ADC 支持将 3 个位同时设置为 1。

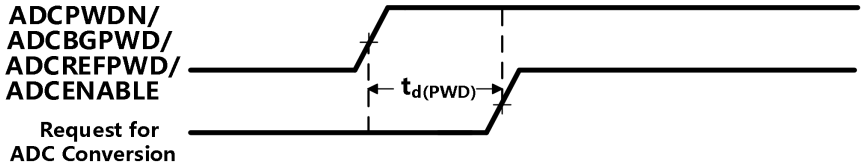
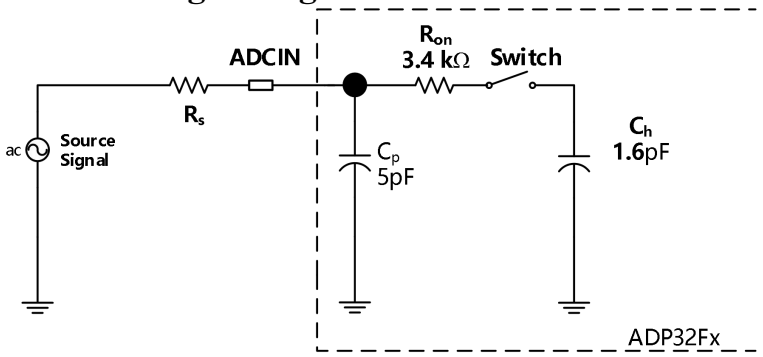



**图 5-17 ADC 转换时序**




Typical Values of the Input Circuit Components:

Switch Resistance ($R_{on}$): 3.4 kΩ

Sampling Capacitor ($C_h$ ): 1.6 pF

Parasitic Capacitance ($C_p$ ): 5 pF

Source Resistance ($R_s$ ): 50 Ω

**图 5-18 模数转换器输入阻抗模型**

### 5.9.2.1.3.5. ADC 顺序和同步时序

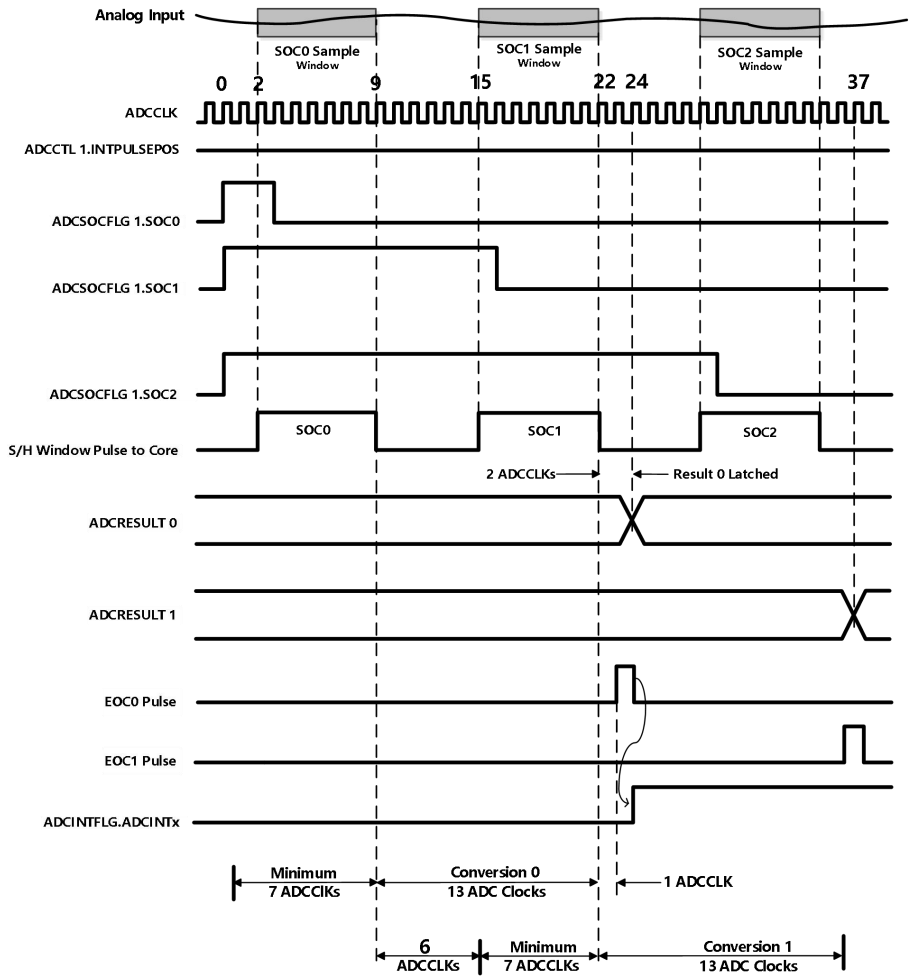



**图 5-19 顺序模式/延迟中断脉冲的时序示例**

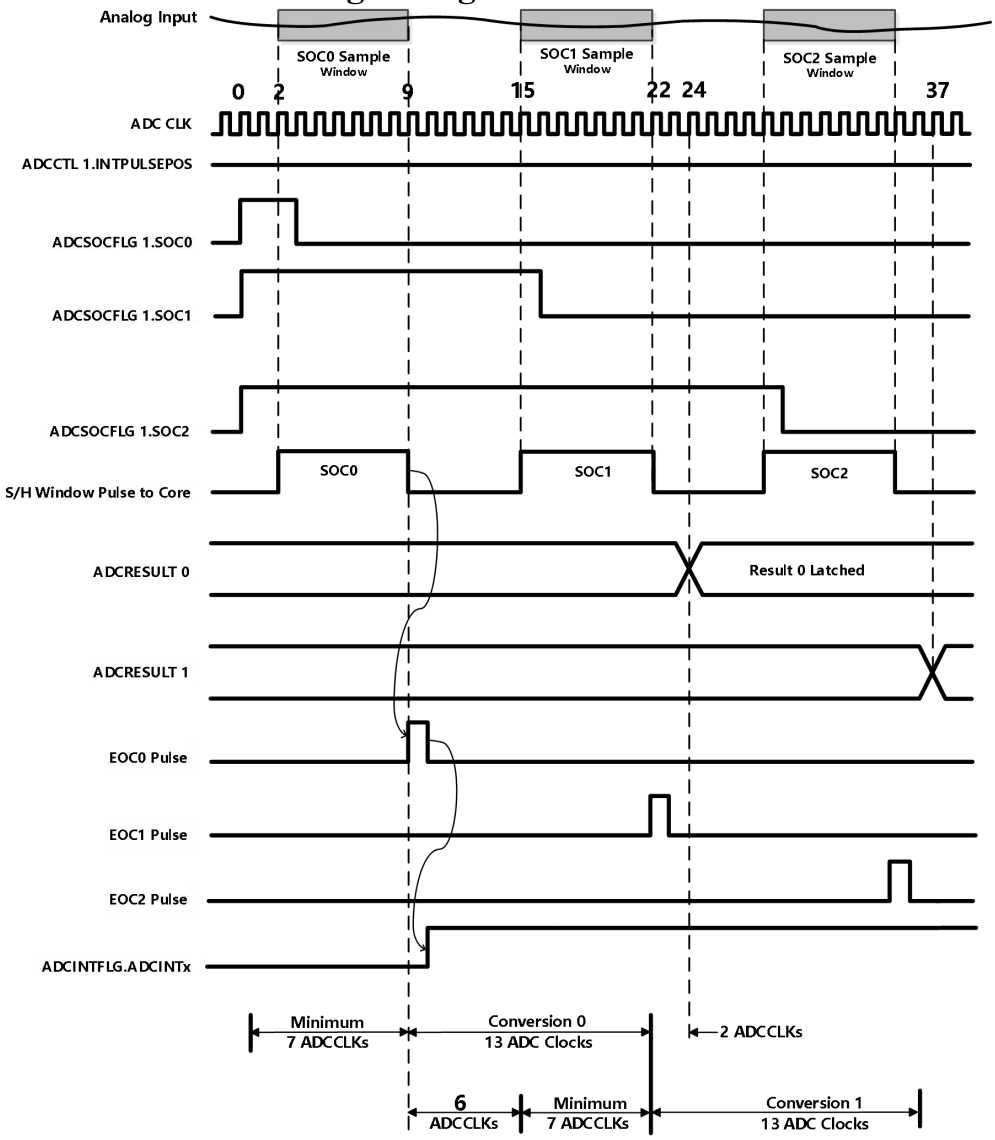



**图 5-20 顺序模式/早期中断脉冲的时序示例**

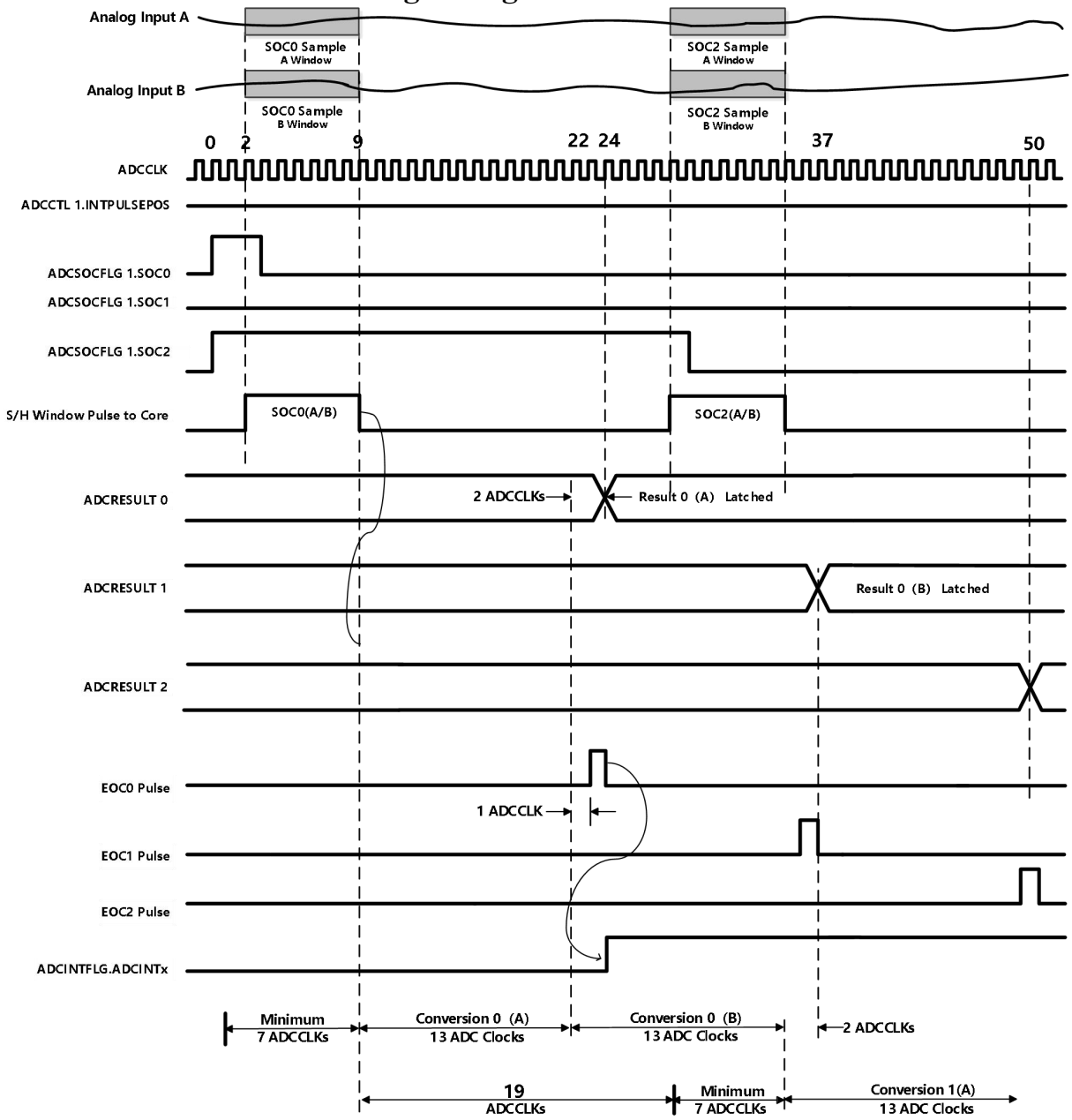



**图 5-21 同步模式/延迟中断脉冲的时序示例**

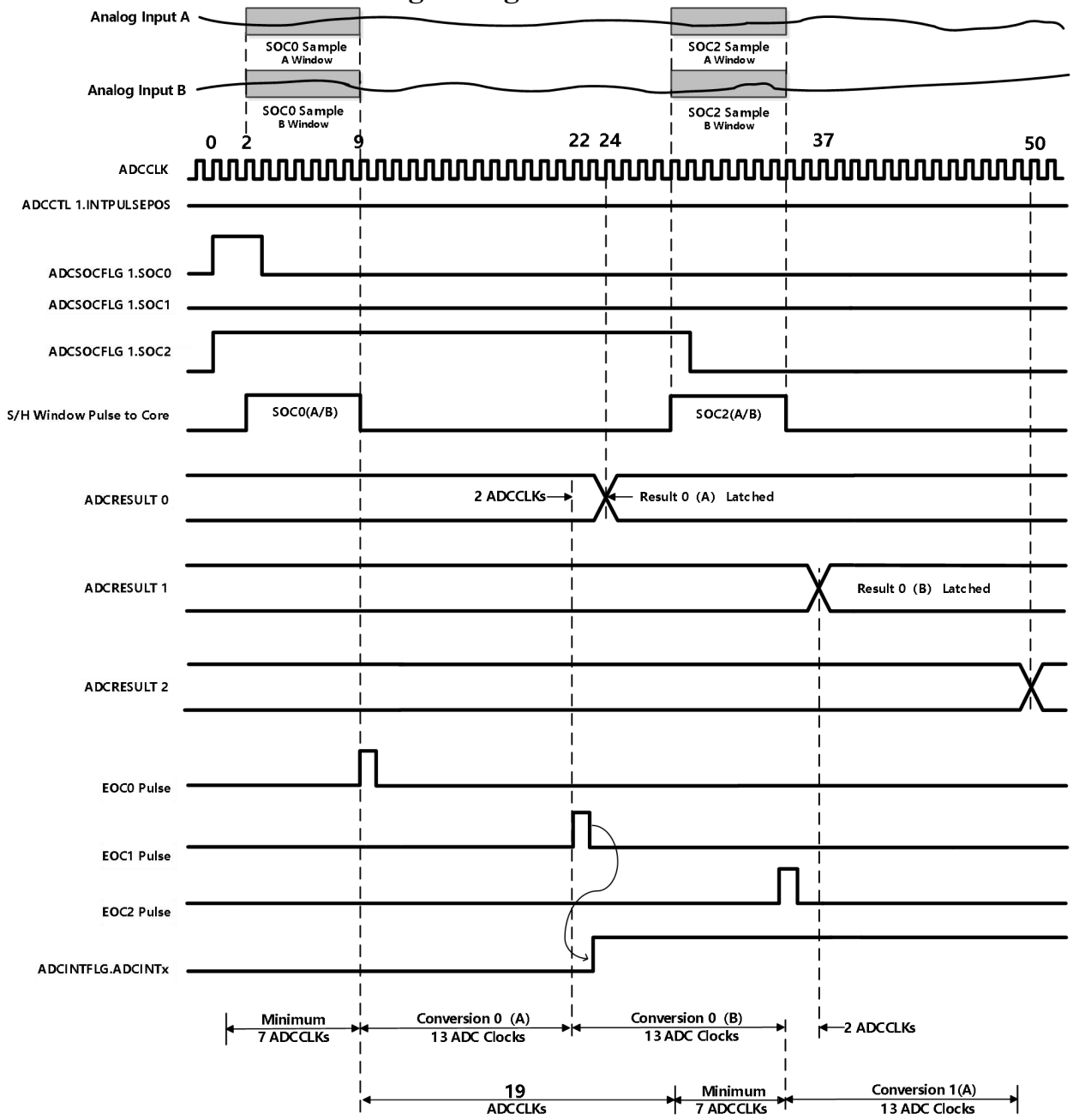



**图 5-22 同步模式/早期中断脉冲的时序示例**

### 5.9.2.2. 模数转换器多路复用器（ADC MUX）

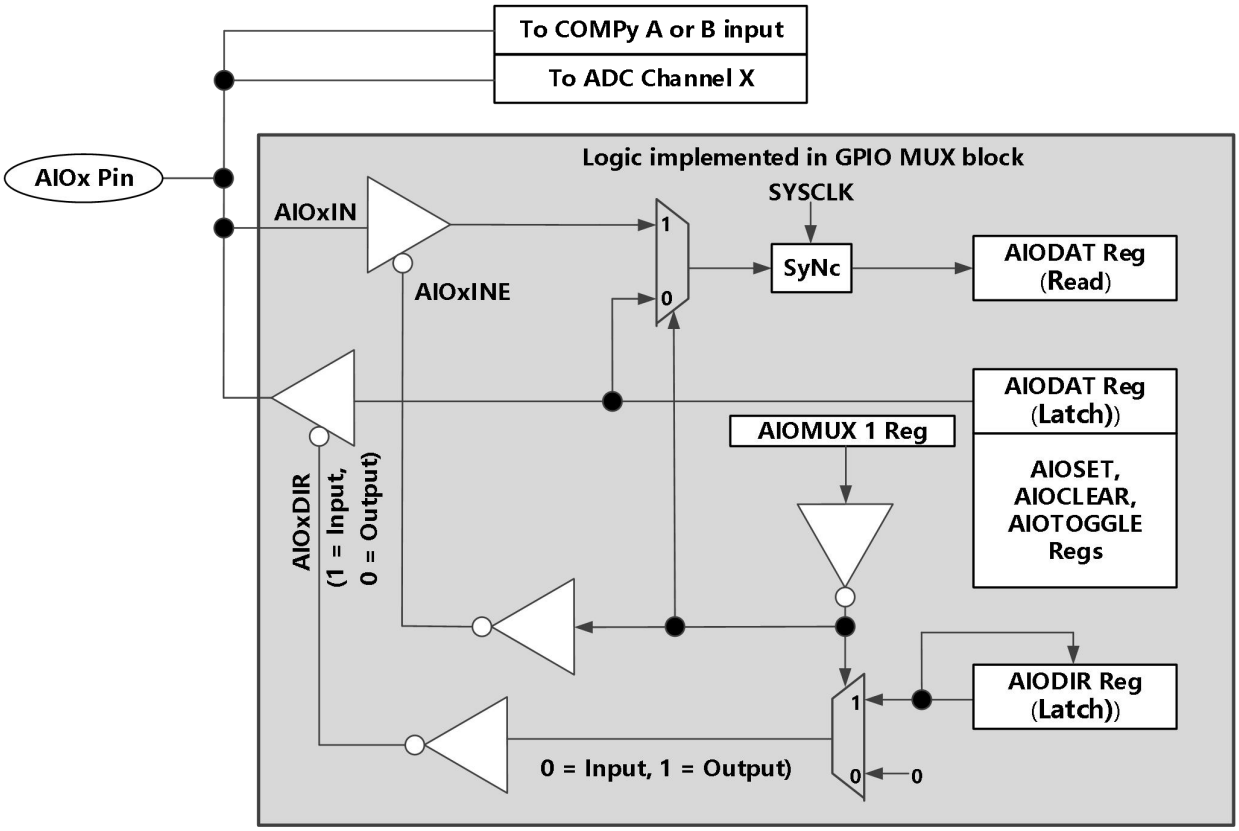



**图 5-23 AIOx 引脚多路复用**

ADC 通道和比较器功能始终可用。数字 I/O 功能仅在 AIOMUX1 寄存器中的相应位为 0 时才可用。在此模式下，读取 AIODAT 寄存器可反映实际的引脚状态。

当 AIOMUX1 寄存器中的相应位为 1 时，数字 I/O 功能被禁用。在这种模式下，读取 AIODAT 寄存器反映 AIODAT 寄存器的输出锁存器，并禁用输入数字 I/O 缓冲器， 以防止其对模拟信号产生噪声。

复位时，数字功能被禁用。如果将该引脚用作模拟输入，则用户应为该引脚禁用 AIO 功能。

### 5.9.2.3. 比较器模块

图 5-23 所示为比较器模块与系统其余部分的交互功能。

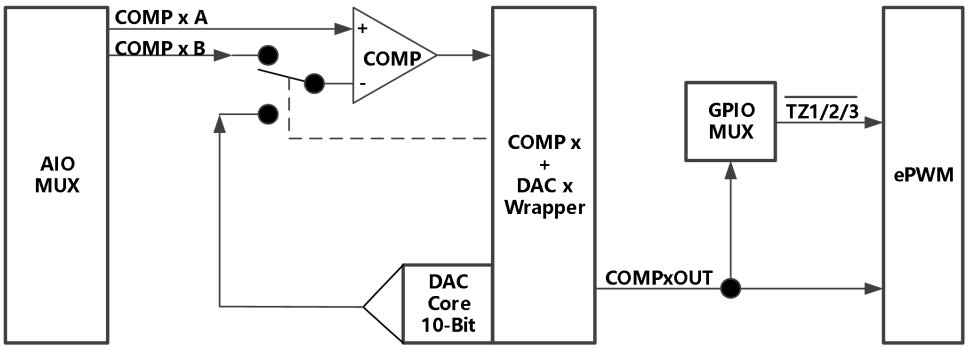



**图 5-24 比较器框图**

表 5-23 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) | 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 | 斜坡发生器最大参考（有效）寄存器 |
| RAMPMAXREF_ SHDW | 0x640A | 0x642A | 0x644A | 1 | 否 | 斜坡发生器最大参考（影子）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（影子）寄存器 |
| RAMPSTS | 0x6410 | 0x6430 | 0x6450 | 1 | 否 | 斜坡发生器状态寄存器 |

### 5.9.2.3.1.片载比较器/DAC电气数据/时序

#### 5.9.2.3.1.1. 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） | | 30 | | 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 | | 10 | | bits |
| DAC 趋稳时间 | | 请参考图5-25 | | |
| DAC 增益 | | -1.5% | | |
| DAC 偏移 | | 10 | | mV |
| 单调 | | 支持 | | |
| INL | | ±3 | | LSB |

(1) 比较器输入端的迟滞是通过施密特触发器配置实现的。这导致比较器输出和比较器非反相输入之间产生有效的 100 kΩ反馈电阻。

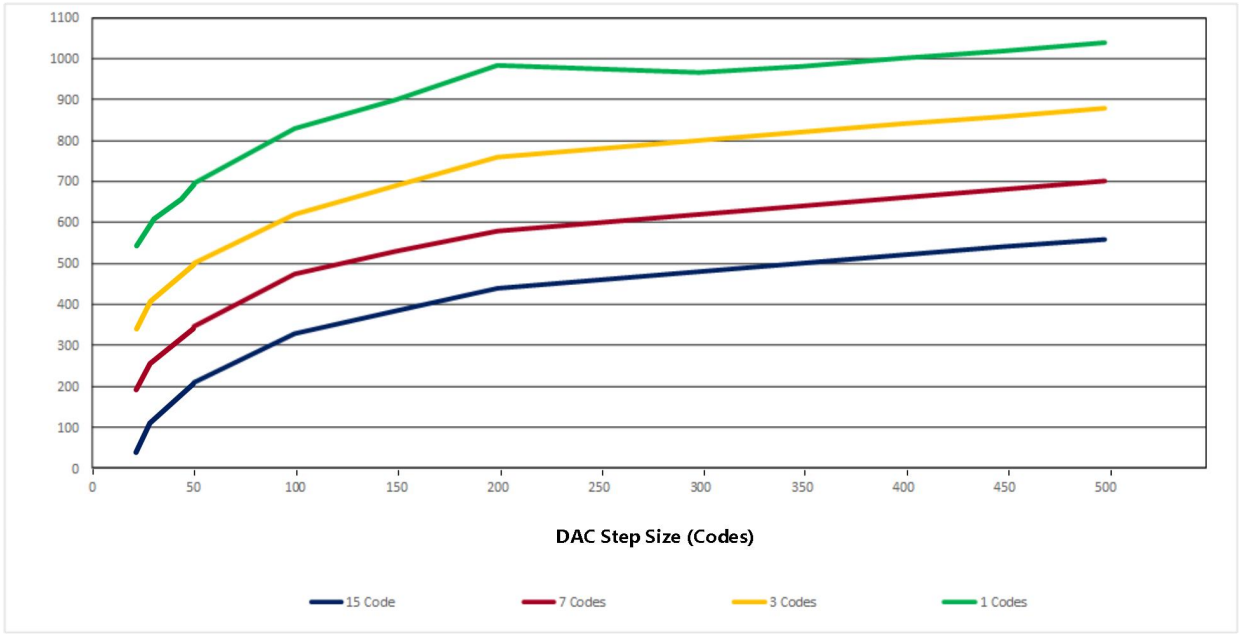



**图 5-25 DAC 趋稳时间**

#### 5.9.2.4. 详细说明

积分非线性（INL）

积分非线性是指每个单独的转换码与从零点到满量程点绘制直线的偏差。 零点出现在第一个转换码之前 1/2LSB 的位置处。满量程点定义为低于最后转换码 1/2 LSB 的位置处。偏差是指测量每个特定转换码的中心到这两个点之间的真实直线的距离。

微分非线性（DNL）

理想的 ADC 展示出转换码间的间距恰好相距 1 LSB。DNL 是与此理想值的偏差。 小于 ±1 LSB 的差分非线性误差可确保无失码。

零点偏移

当模拟输入为零伏时，应发生主要的载波转换。零误差定义为实际转换与该点的偏差。

增益误差

第一个转换码应出现在比负满量程高 1/2LSB 的模拟值处。最后一个转换码应出现在比标称满量程低 1/2 LSB 的模拟值处。增益误差是指实际第一个和最后一个转换码之间的差值与理想第一个和最后一个转换码之间的差值的偏差 。

信噪比 + 失真 （SINAD）

SINAD 是输入信号测量值的均方根值与其他所有低于奈奎斯特频率的频谱分量的均方根总和之比 ，包括谐波，但不包括直流。SINAD 的值以分贝表示。

有效位数 （ENOB）

对于正弦波，SINAD 可以用位数来表示。使用以下公式，$N = \frac{(SINAD - 1.76)}{6.02}$可以得到以 N 表示的性能度量，即有效位数。因此，在给定输入频率下，正弦波输入器件的有效位数可以直接从其测量的 SINAD 计算出来。

总谐波失真 （THD）

THD 是前九个谐波分量的均方根总和与输入信号测量值的均方根值的比值 ，以百分比或分贝表示。

无杂散动态范围（SFDR）

SFDR 是输入信号的均方根幅度与峰值杂散信号的均方根幅度之间的差值，以 dB 为单位。

### 5.9.3. 串行外设接口模块(SPI)

该器件包括四引脚串行外设接口（SPI）模块。可提供多达两个 SPI 模块。SPI 是一个高速、同步串行 I/O 端口，允许以可编程位传输速率将编程长度（1 至 16 位）的串行位流移入和移出器件。通常，SPI 用于 MCU 与外部外设或其他处理器之间的通信。典型应用包括外部 I/O 或通过移位寄存器、显示驱动器和 ADC 等器件进行外设扩展。SPI 的主/从操作支持多器件通信。

SPI 模块特性包括：

- 四个外部引脚：
  - SPISOMI：SPI 从机输出/主机输入引脚
  - SPISIMO：SPI 从机输入/主机输出引脚
  - $\overline{\text{SPISTE}}$：SPI 从机发送使能引脚
  - SPICLK：SPI 串行时钟引脚

注 意

如果未使用 SPI 模块，则所有四个引脚都可以用作 GPIO。

- 两种操作模式：主机和从机

  波特率：125 种不同的可编程速率 SPIBRR 是 3 至 127 时，波特率是 LSPCLK/(SPIBRR+1)

  当 SPIBRR 是 0，1，2 时，波特率等于 LSPCLK/4。
- 数据字长：1 至 16
- 四种时钟方案 (由时钟极性位和时钟相位位决定) 包括:
  - 不包含相位延迟的下降沿：SPICLK 没有数据发送时，SPICLK 处于高电平。SPI 在 SPICLK 信号的下降沿发送数据，在 SPICLK 的上升沿接收数据。
  - 含相位延迟的下降沿：SPICLK 没有数据发送时，SPICLK 处于高电平。SPI 在 SPICLK 信号的下降沿延时半个周期后发送数据，在随后的下降沿接收数据。
  - 上升沿且无相位延迟：SPICLK 没有数据发送时，SPICLK 处于低电平。SPI 在 SPICLK 信号的上升沿发送数据，在 SPICLK 信号的下降沿接收数据。
  - 上升沿且有相位延迟：SPICLK 没有数据发送时，SPICLK 处于低电平。SPI 在 SPICLK 信号

的上升沿延时半个周期后发送数据，在随后的上升沿接收数据。

- 同时接收和发送数据 (可以在软件中禁止发送功能)。
- 发送和接收操作是通过中断驱动或者轮询算法完成。
- 9 个 SPI 控制寄存器：控制寄存器的起始地址是 7040h。

注　意

该模块中的所有寄存器都是连接到外设帧 2 的 16 位寄存器。访问寄存器时，寄存器数据位于低位字节（7–0）中，高位字节（15–8）读取为零。写入高位字节不起作用。

增强特性：

- 4 级发送/接收 FIFO
- 带延时的发送控制
- 支持双向的 3 线 SPI 模式
- 可在两个数字音频接口的接收模式中 SPI 模块中设置$\overline{\text{SPISTE}}$反转

SPI 端口操作配置和控制由寄存器控制，寄存器如表 5-24 和表 5-25 所述。

**表 5-24 SPI-A 寄存器**

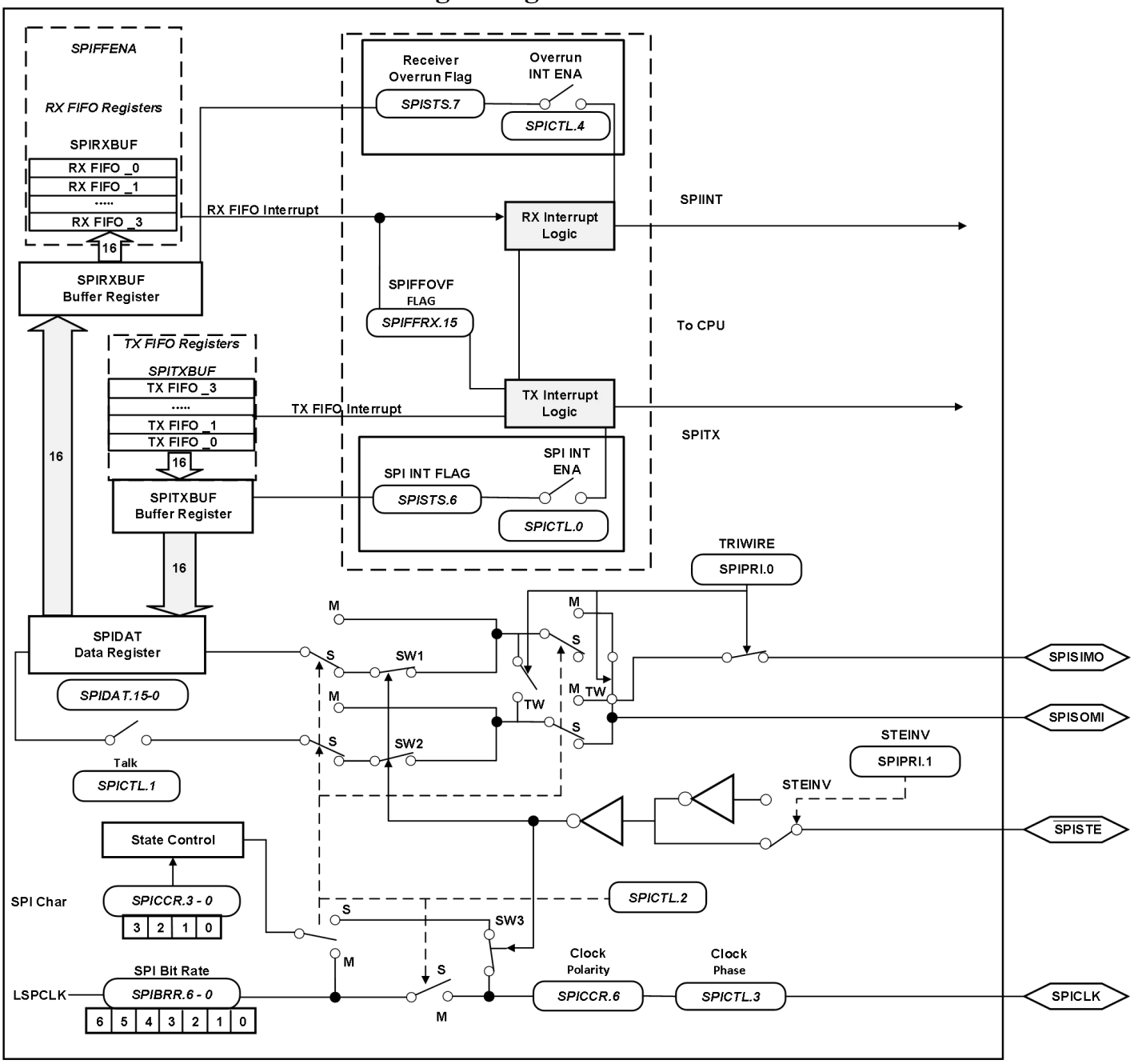
| 名称 | 地址 | 大小 (×16) | EALLOW 保护 | 描述[1] |
|---|---|---|---|---|
| SPICCR | 0x7040 | 1 | No | SPI-A 配置控制寄存器 |
| SPICTL | 0x7041 | 1 | No | SPI-A 操作控制寄存器 |
| SPISTS | 0x7042 | 1 | No | SPI-A 状态寄存器 |
| SPIBRR | 0x7044 | 1 | No | SPI-A 波特率寄存器 |
| SPIRXEMU | 0x7046 | 1 | No | SPI-A 接收调试缓冲寄存器 |
| SPIRXBUF | 0x7047 | 1 | No | SPI-A 串行输入缓冲寄存器 |
| SPITXBUF | 0x7048 | 1 | No | SPI-A 串行输出缓冲寄存器 |
| SPIDAT | 0x7049 | 1 | No | SPI-A 串行数据寄存器 |
| SPIFFTX | 0x704A | 1 | No | SPI-A FIFO 发送寄存器 |
| SPIFFRX | 0x704B | 1 | No | SPI-A FIFO 接收寄存器 |
| SPIFFCT | 0x704C | 1 | No | SPI-A FIFO 控制寄存器 |
| SPIPRI | 0x704F | 1 | No | SPI-A 优先级控制寄存器 |

(1) 此表中的寄存器映射到外设帧 2。此空间仅允许 16 位访问。32 位访问将产生未知的结果。

表 5-25 SPI-B 寄存器

| 名称 | 地址 | 大小(×16) | EALLOW 保护 | 描述(1) |
|---|---|---|---|---|
| SPICCR | 0x7740 | 1 | No | SPI-B 配置控制寄存器 |
| SPICTL | 0x7741 | 1 | No | SPI-B 操作控制寄存器 |
| SPISTS | 0x7742 | 1 | No | SPI-B 状态寄存器 |
| SPIBRR | 0x7744 | 1 | No | SPI-B 波特率寄存器 |
| SPIRXEMU | 0x7746 | 1 | No | SPI-B 接收调试缓冲寄存器 |
| SPIRXBUF | 0x7747 | 1 | No | SPI-B 串行输入缓冲寄存器 |
| SPITXBUF | 0x7748 | 1 | No | SPI-B 串行输出缓冲寄存器 |
| SPIDAT | 0x7749 | 1 | No | SPI-B 串行数据寄存器 |
| SPIFFTX | 0x774A | 1 | No | SPI-B FIFO 发送寄存器 |
| SPIFFRX | 0x774B | 1 | No | SPI-B FIFO 接收寄存器 |
| SPIFFCT | 0x774C | 1 | No | SPI-B FIFO 控制寄存器 |
| SPIPRI | 0x774F | 1 | No | SPI-B 优先级控制寄存器 |

(1) 此表中的寄存器映射到外设帧 2。此空间仅允许 16 位访问。32 位访问将产生未知的结果。




A. $\overline{\text{SPISTE}}$由主机驱动为用于从机的低电平。

**图 5-26 SPI 模块框图（从模式）**

### 5.9.3.1. SPI 主机模式电气数据/时序

第 5.9.3.1.1 节列出了主机模式的时序 (时钟相位 = 0)，第 5.9.3.1.2 节列出了主机模式时序(时钟相位 = 1)。图 5-27 和 图 5-28 展示了时序波形。

#### 5.9.3.1.1. 主机模式外部时序 (Clock Phase = 0)

| 编号 | 参数项(1) (2) (3) (4) (5) | | 波特率配置寄存器偶数 | | 波特率配置寄存器奇数 | | 单位 |
|---|---|---|---|---|---|---|---|
| | | | 最小值 | 最大值 | 最小值 | 最大值 | |
| 1 | $t_{c(SPC)M}$ | 周期时间，SPICLK | $4t_{c(LSPCLK)}$ | $128t_{c(LSPCLK)}$ | $5t_{c(LSPCLK)}$ | $127t_{c(LSPCLK)}$ | ns |
| 2 | $t_{w(SPC1)M}$ | 脉冲宽度，SPICLK 第一个脉冲 | $0.5t_{c(SPC)M} - 10$ | $0.5t_{c(SPC)M} + 10$ | $0.5t_{c(SPC)M} + 0.5t_{c(LSPCLK)} - 10$ | $0.5t_{c(SPC)M} + 0.5t_{c(LSPCLK)} + 10$ | ns |
| 3 | $t_{w(SPC2)M}$ | 脉冲宽度，SPICLK 第二个脉冲 | $0.5t_{c(SPC)M} - 10$ | $0.5t_{c(SPC)M} + 10$ | $0.5t_{c(SPC)M} - 0.5t_{c(LSPCLK)} - 10$ | $0.5t_{c(SPC)M} - 0.5t_{c(LSPCLK)} + 10$ | ns |
| 4 | $t_{d(SIMO)M}$ | 延迟时间，SPICLK 有效时钟沿至SPISIMO数据有效的时间 | | 10 | | 10 | ns |
| 5 | $t_{v(SIMO)M}$ | 有效时间，在SPICLK有效时钟沿后SPISIMO数据有效的时间 | $0.5t_{c(SPC)M} - 10$ | | $0.5t_{c(SPC)M} - 0.5t_{c(LSPCLK)} - 10$ | | ns |
| 8 | $t_{su(SOMI)M}$ | 建立时间，SPICLK时钟有效沿到来前 SPISOMI 数据有效的时间 | 26 | | 26 | | ns |
| 9 | $t_{h(SOMI)M}$ | 保持时间，SPICLK有效时钟沿后SPISOMI数据有效的时间 | 0 | | 0 | | ns |
| 23 | $t_{d(SPC)M}$ | 延迟时间，$\overline{SPISTE}$有效到SPICLK有效时钟沿的时间 | $1.5t_{c(SPC)M} - 3t_{c(SYSCLK)} - 10$ | | $1.5t_{c(SPC)M} - 3t_{c(SYSCLK)} - 10$ | | ns |
| 24 | $t_{d(STE)M}$ | 延迟时间，SPICLK 有效到$\overline{SPISTE}$无效的时间 | $0.5t_{c(SPC)M} - 10$ | | $0.5t_{c(SPC)M} - 0.5t_{c(LSPCLK)} - 10$ | | ns |

(1) MASTER / SLAVE 位 (SPICTL.2) 设置和时钟相位位 (SPICTL.3) 被清除。

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

(3) $t_{c(LCO)}$ = LSPCLK 周期时间

(4) 必须对内部时钟预分频器进行调整，以便将 SPI 时钟速度限制为以下 SPI 时钟速率:

主机模式发送最大的频率是 25MHz，主机模式接收最大频率是 12.5MHz。

从机模式发送最大的频率是 12.5MHz，从机模式接收最大频率是 12.5MHz。

(5) 参考的 SPICLK 信号的有效边沿由时钟极性位控制 (SPICCR.6)。

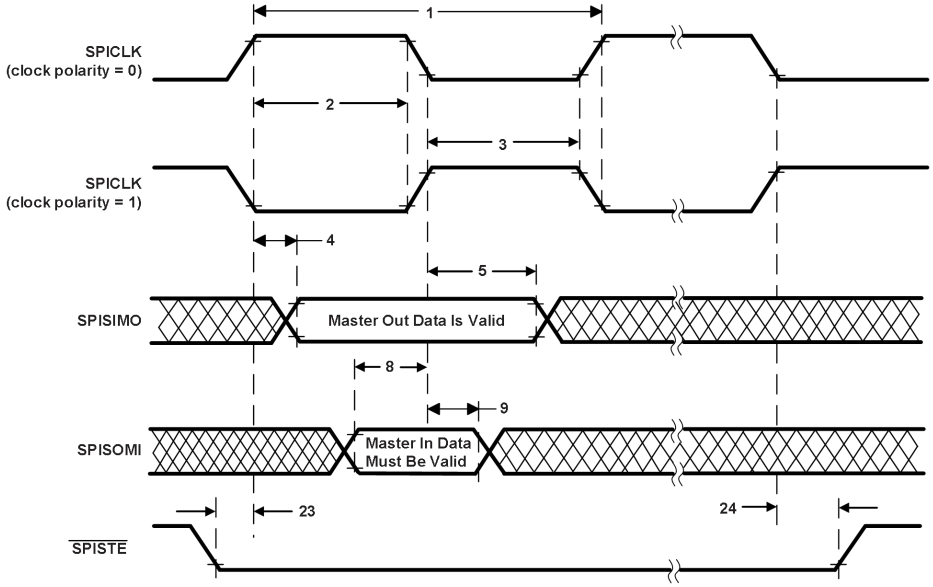



**图 5-27 SPI 主机模式外部时序(时钟相位 = 0)**

### 5.9.3.1.2. SPI 主机模式外部时序 (Clock Phase = 1)

| 编号 | 参数项(1) (2) (3) (4) (5) | | 波特率配置寄存器偶数 | | 波特率配置寄存器奇数 | | 单位 |
|---|---|---|---|---|---|---|---|
| | | | 最小值 | 最大值 | 最小值 | 最大值 | |
| 1 | $t_{c(SPC)M}$ | 周期时间, SPICLK | $4t_{c(LSPCLK)}$ | $128t_{c(LSPCLK)}$ | $5t_{c(LSPCLK)}$ | $127t_{c(LSPCLK)}$ | ns |
| 2 | $t_{w(SPC1)M}$ | 脉冲宽度, SPICLK 第一个脉冲宽度 | $0.5t_{c(SPC)M} - 10$ | $0.5t_{c(SPC)M} + 10$ | $0.5t_{c(SPC)M} - 0.5t_{c(LSPCLK)} - 10$ | $0.5t_{c(SPC)M} - 0.5t_{c(LSPCLK)} + 10$ | ns |
| 3 | $t_{w(SPC2)M}$ | 脉冲宽度, SPICLK 第二个脉冲 | $0.5t_{c(SPC)M} - 10$ | $0.5t_{c(SPC)M} + 10$ | $0.5t_{c(SPC)M} + 0.5t_{c(LSPCLK)} - 10$ | $0.5t_{c(SPC)M} + 0.5t_{c(LSPCLK)} + 10$ | ns |
| 6 | $t_{d(SIMO)M}$ | 延迟时间, SPISIMO数据有效至SPICLK有效时钟沿 | $0.5t_{c(SPC)M} - 10$ | | $0.5t_{c(SPC)M} + 0.5t_{c(LSPCLK)} - 10$ | | ns |
| 7 | $t_{v(SIMO)M}$ | 有效时间, SPICLK有效时钟沿后SPISIMO数据有效 | $0.5t_{c(SPC)M} - 10$ | | $0.5t_{c(SPC)M} - 0.5t_{c(LSPCLK)} - 10$ | | ns |
| 10 | $t_{su(SOMI)M}$ | 建立时间, SPICLK有效时钟沿到来前SPISOMI数据有效 | 26 | | 26 | | ns |
| 11 | $t_{h(SOMI)M}$ | 保持时间, SPICLK有效时钟沿至SPISOMI 数据有效 | 0 | | 0 | | ns |
| 23 | $t_{d(SPC)M}$ | 延时时间, SPISTE 有效 到 SPICLK有效时钟沿 | $2t_{c(SPC)M} - 3t_{c(SYSCLK)} - 10$ | | $2t_{c(SPC)M} - 3t_{c(SYSCLK)} - 10$ | | ns |
| 24 | $t_{d(STE)M}$ | 延时时间, SPICLK 时钟有效沿到SPISTE无效 | $0.5t_{c(SPC)} - 10$ | | $0.5t_{c(SPC)} - 0.5t_{c(LSPCLK)} - 10$ | | ns |

(1) MASTER/SLAVE 位 (SPICTL.2)设置和时钟相位位 (SPICTL.3) 设置。

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

(3) 必须对内部时钟预分频器进行调整，以便将 SPI 时钟速度限制为以下 SPI 时钟速率：
主机模式发送最大的频率是 25MHz，主机模式接收最大频率是 12.5MHz。
从机模式发送最大的频率是 12.5MHz，从机模式接收最大频率是 12.5MHz。

(4) $t_{c(LCO)}$ = LSPCLK 周期时间。

(5) 参考的 SPICLK 信号的有效边沿由时钟极性位控制 (SPICCR.6)。

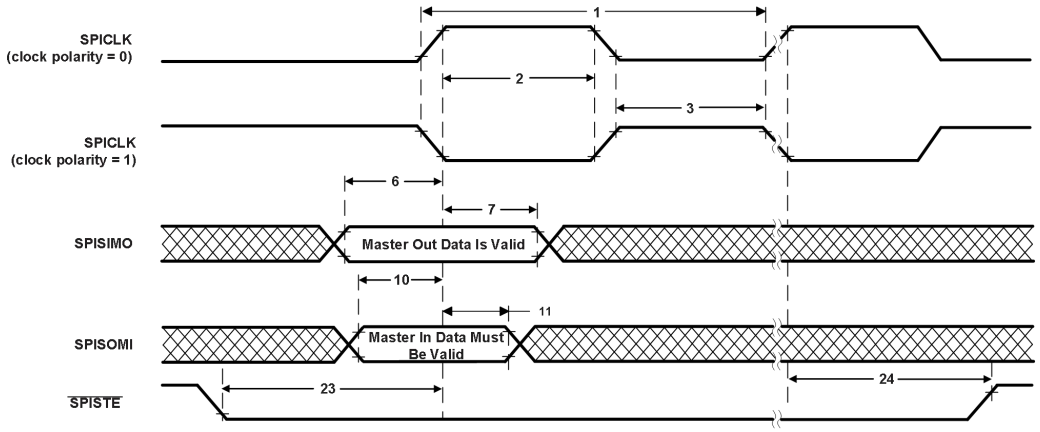



**图 5-28 SPI 主机模式外部时序 (时钟相位 = 1)**

### 5.9.3.2. SPI 从机模式电气数据/时序

第 5.9.3.2.1 节列出了从机模式时序(clock phase = 0)，第 5.9.3.2.2 节列出了从机模式时序(clock phase = 1)。图 5-29 和图 5-30 所示为时序图。

#### 5.9.3.2.1. SPI 从机模式外部时序 (Clock Phase = 0)

| NO. | 参数项(1) (2) (4) (3) (5) | | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|---|
| 12 | $t_{c(SPC)S}$ | 周期时间, SPICLK | $4t_{c(SYSCLK)}$ | | ns |
| 13 | $t_{w(SPC1)S}$ | 脉冲宽度, SPICLK 第一个脉冲 | $2t_{c(SYSCLK)} - 1$ | | ns |
| 14 | $t_{w(SPC2)S}$ | 脉冲宽度, SPICLK 第二个脉冲 | $2t_{c(SYSCLK)} - 1$ | | ns |
| 15 | $t_{d(SOMI)S}$ | 延时时间, SPICLK有效沿 到 SPISOMI 有效 | | 21 | ns |
| 16 | $t_{v(SOMI)S}$ | 有效时间, SPICLK有效时钟沿后到SPISOMI数据有效 | 0 | | ns |
| 19 | $t_{su(SIMO)S}$ | 建立时间，SPICLK时钟沿到来前SPISIMO数据有效 | $1.5t_{c(SYSCLK)}$ | | ns |
| 20 | $t_{h(SIMO)S}$ | 保持时间, SIPCLK时钟沿到来后SPISIMO数据有效 | $1.5t_{c(SYSCLK)}$ | | ns |
| 25 | $t_{su(STE)S}$ | 建立时间，在SPICLK时钟沿到来前$\overline{SPISTE}$有效 | $1.5t_{c(SYSCLK)}$ | | ns |
| 26 | $t_{h(STE)S}$ | 保持时间, SPICLK时钟沿后$\overline{SPISTE}$无效 | $1.5t_{c(SYSCLK)}$ | | ns |

(1) MASTER / SLAVE 位 (SPICTL.2) 被清零和时钟相位位(SPICTL.3)被清零。

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

(3) $t_{c(LCO)}$ = LSPCLK 周期时间

(4) 必须对内部时钟预分频器进行调整，以便将 SPI 时钟速度限制为以下 SPI 时钟速率：
主机模式发送最大的频率是 25MHz，主机模式接收最大频率是 12.5MHz。
从机模式发送最大的频率是 12.5MHz，从机模式接收最大频率是 12.5MHz。

(5) 参考的 SPICLK 信号的有效边沿由时钟极性位控制。

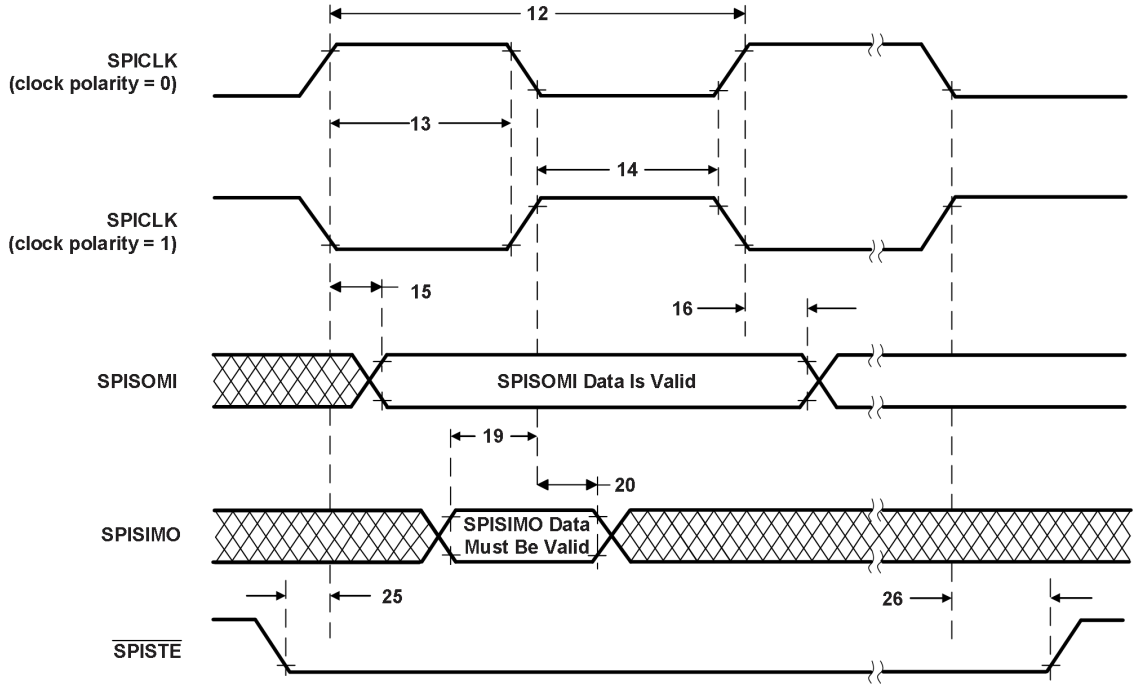



**图 5-29 SPI 从机模式外部时序 (时钟相位 = 0)**

### 5.9.3.2.2. SPI 从机外部时序 (Clock Phase = 1)

| NO. | 参数项(1) (2) (3) (4) | | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|---|
| 12 | $t_{c(SPC)S}$ | 周期时间, SPICLK | $4t_{c(SYSCLK)}$ | | ns |
| 13 | $t_{w(SPC1)S}$ | 脉冲宽度, SPICLK 第一个脉冲 | $2t_{c(SYSCLK)} - 1$ | | ns |
| 14 | $t_{w(SPC2)S}$ | 脉冲宽度, SPICLK第二个脉冲 | $2t_{c(SYSCLK)} - 1$ | | ns |
| 17 | $t_{d(SOMI)S}$ | 延迟时间, SPICLK时钟有效沿至SPISOMI数据有效 | | 21 | ns |
| 18 | $t_{v(SOMI)S}$ | 有效时间, SPICLK有效时钟沿后SPISOMI 数据有效 | 0 | | ns |
| 21 | $t_{su(SIMO)S}$ | 建立时间, SPICLK时钟有效沿到来前SPISIMO数据有效 | $1.5t_{c(SYSCLK)}$ | | ns |
| 22 | $t_{h(SIMO)S}$ | 保持时间, SPICLK有效沿后SPISIMO数据需保持时间 | $1.5t_{c(SYSCLK)}$ | | ns |
| 25 | $t_{su(STE)S}$ | 建立时间, SPICLK时钟有效沿到来前$\overline{SPISTE}$有效 | $1.5t_{c(SYSCLK)}$ | | ns |
| 26 | $t_{h(STE)S}$ | 保持时间, 在SICLK有效沿后$\overline{SPISTE}$无效 | $1.5t_{c(SYSCLK)}$ | | ns |

(1) MASTER / SLAVE 位 (SPICTL.2)被清除，且 CLOCK PHASE 位 (SPICTL.3)被清除。

(2) $t_{c(SPC)}$ = SPI 时钟周期 SPI clock cycle time = LSPCLK/4 or LSPCLK/(SPIBRR + 1)。

(3) 必须对内部时钟预分频器进行调整，以便将 SPI 时钟速度限制为以下 SPI 时钟速率：

主机模式发送最大频率 25-MHz，主机接收模式最大频率 12.5-MHz。

从机模式发送最大频率 12.5-MHz，从机接收模式最大频率 12.5-MHz。

(4) 引用的 SPICLK 信号的有效边沿由时钟极性位控制(SPICCR.6)。

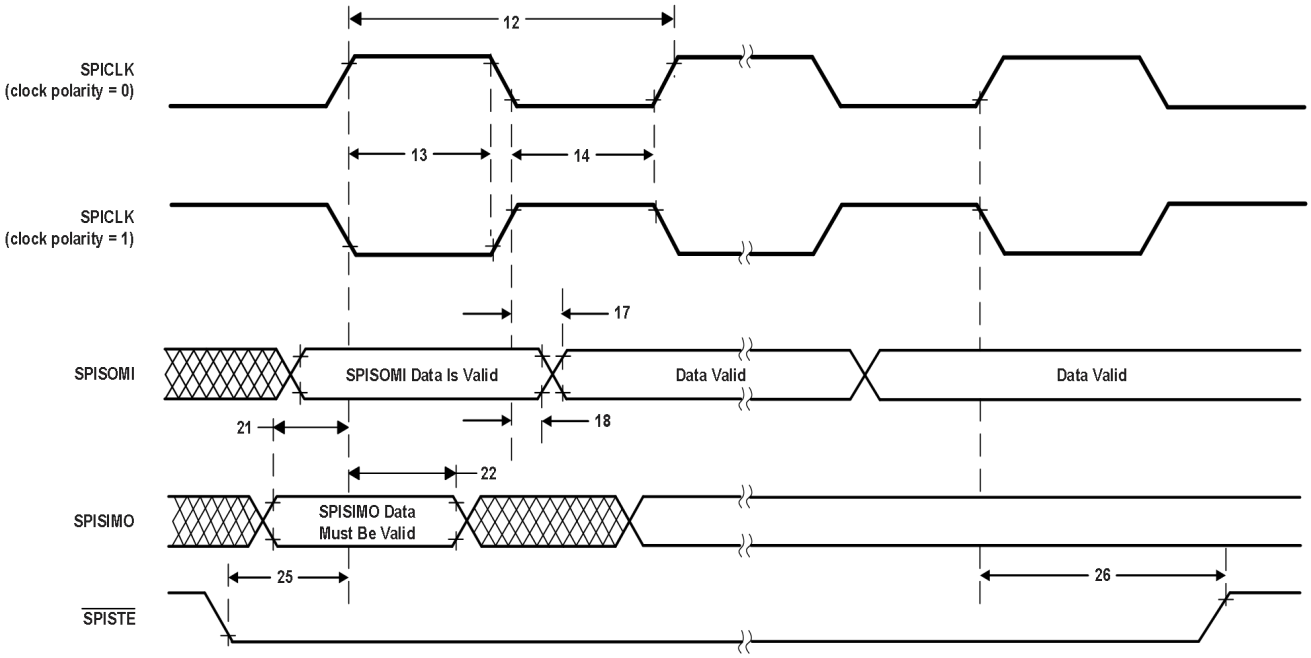



**图 5-30 SPI 从机模式的外部时序 (Clock Phase = 1)**

### 5.9.4. 异步串行通信接口（SCI）模块

AVP32F069 器件包括两个异步串行通信接口(SCI) 模块 (SCI-A, SCI-B)。SCI 模块支持 CPU 与其它异步外设之间的使用标准非归零码(NRZ) 格式的数字通信。SCI 接收器和发送器是双缓冲的，并且它们中的每一个有其自身独立的使能和中断位。两个器件都可独立或者同时地运行在全双工模式。为了确保数据完整性，SCI 在中断检测、奇偶性、超载和组帧错误方面对接收到的数据进行检查。通过一个 16 位波特率选择寄存器，可将比特率设定为超过 65000 种不同的波特率。

每个 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

| 注意 |
| --- |
| 如果不用于 SCI，两个引脚都可以用作 GPIO。 |

- 64K 种不同速率的可编程波特率:

$$\text{波特率} = \frac{\text{LSPCLK}}{(\text{BRR}+1) * 8} \quad \text{当 BRR} \neq 0$$

$$\text{波特率} = \frac{\text{LSPCLK}}{16} \quad \text{当 BRR} = 0$$

- 数据字格式
  - -一个起始位
  - -数据字长度可编程为 1 至 8 位
  - -可选偶数/奇数/无奇偶校验位
  - -1 或者 2 位停止位
- 四个错误检测标志：奇偶校验错误、溢出错误、帧错误和中断检测错误
- 两种唤醒多处理器模式：空闲线唤醒和地址位唤醒
- 半双工或全双工操作
- 双缓冲接收和发送功能
- 发送和接收操作可通过中断驱动或带有状态标志的轮询算法完成。
  - -发送：TXRDY 标志（发送寄存器准备发送另一个字符）和 TX 空标志（发送移位寄存器为空）
  - -接收：RXRDY 标志（接收寄存器准备接收另一个字符）、BRKDT 标志（发生中断条件）和 RX ERROR 标志（监测四个中断条件）
- 独立的发送和接收使能控制（BRKDT 除外）
- NRZ（不归零）格式

| 注意 |
| --- |
| 此模块中的所有寄存器都是连接到外围帧 2 的 8 位寄存器。当访问寄存器时，寄存器数据位于低位字节（7-0），高位字节（15-8）读取为零。写入高位字节无效。 |

增强的功能：

- 自动波特率检测硬件逻辑
- 4 级发送/接收 FIFO

SCI 端口操作由表 5-26 和 表 5-27 中列出的寄存器进行配置和控制。

**表 5-26 SCI-A 寄存器**

| 名称(1) | 地址 | 大小 (×16) | EALLOW 保护 | 描述 |
|---|---|---|---|---|
| SCICCRA | 0x7050 | 1 | No | SCI-A 通信控制寄存器 |
| SCICTL1A | 0x7051 | 1 | No | SCI-A 控制寄存器 1 |
| SCIHBAUDA | 0x7052 | 1 | No | SCI-A 波特率寄存器，高位 |
| SCILBAUDA | 0x7053 | 1 | No | SCI-A 波特率寄存器，低位 |
| SCICTL2A | 0x7054 | 1 | No | SCI-A 控制寄存器 2 |
| SCIRXSTA | 0x7055 | 1 | No | SCI-A 接收状态寄存器 |
| SCIRXEMUA | 0x7056 | 1 | No | SCI-A接收仿真数据缓冲寄存器 |
| SCIRXBUFA | 0x7057 | 1 | No | SCI-A 接收数据缓冲寄存器 |
| SCITXBUFA | 0x7059 | 1 | No | SCI-A 发送数据缓冲寄存器 |
| SCIFFTXA(2) | 0x705A | 1 | No | SCI-A FIFO 发送寄存器 |
| SCIFFRXA(2) | 0x705B | 1 | No | SCI-A FIFO 接收寄存器 |
| SCIFFCTA(2) | 0x705C | 1 | No | SCI-A FIFO 控制寄存器 |
| SCIPRIA | 0x705F | 1 | No | 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。此空间仅允许 16 位访问。32 位访问会产生未定义的结果。

(2) 这些寄存器是 FIFO 模式的新寄存器。

**表 5-27 SCI-B 寄存器**

| 名称(1) | 地址 | 大小 (×16) | 描述 |
|---|---|---|---|
| SCICCRB | 0x7750 | 1 | SCI-B 通信控制寄存器 |
| SCICTL1B | 0x7751 | 1 | SCI-B 控制寄存器 1 |
| SCIHBAUDB | 0x7752 | 1 | SCI-B 波特率寄存器，高位 |
| SCILBAUDB | 0x7753 | 1 | SCI-B 波特率寄存器，低位 |
| SCICTL2B | 0x7754 | 1 | SCI-B 控制寄存器 2 |
| SCIRXSTB | 0x7755 | 1 | SCI-B 接收状态寄存器 |
| SCIRXEMUB | 0x7756 | 1 | SCI-B接收仿真数据缓冲寄存器 |
| SCIRXBUFB | 0x7757 | 1 | SCI-B 接收数据缓冲寄存器 |
| SCITXBUFB | 0x7759 | 1 | SCI-B 发送数据缓冲寄存器 |
| SCIFFTXB(2) | 0x775A | 1 | SCI-B FIFO 发送寄存器 |
| SCIFFRXB(2) | 0x775B | 1 | SCI-B FIFO 接收寄存器 |
| SCIFFCTB(2) | 0x775C | 1 | SCI-B FIFO 控制寄存器 |
| SCIPRIB | 0x775F | 1 | SCI-B 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。此空间仅允许 16 位访问。32 位访问会产生未定义的结果。

(2) 这些寄存器是 FIFO 模式的新寄存器。

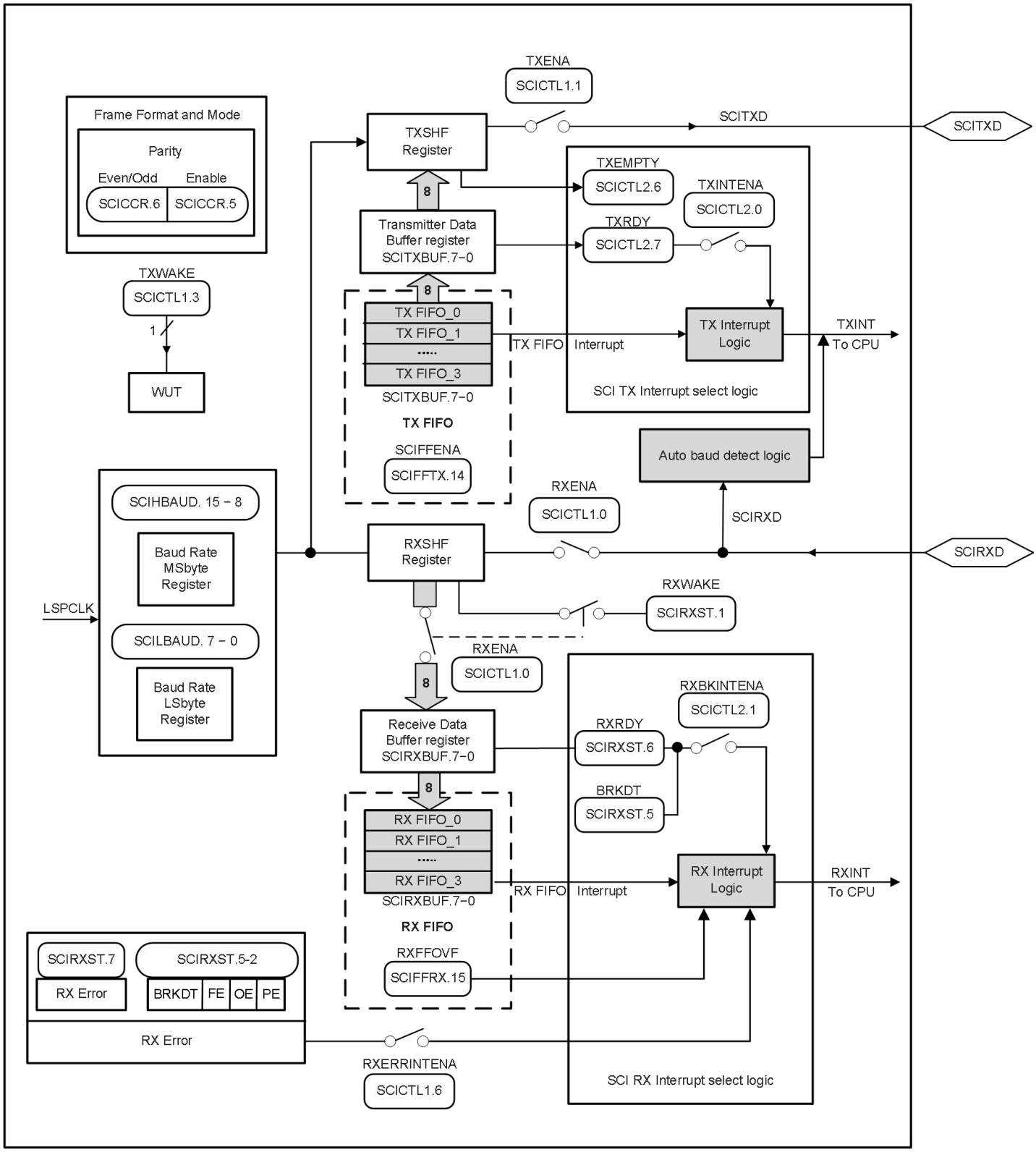



**图 5-31 串行通信接口（SCI）模块框图**

## 5.9.5. 多通道缓冲串行端口 (McBSP) 模块

### 5.9.5.1. McBSP 基本特性

McBSP 模块有以下特性:

- 与 AVP32Fx DSP 器件内的 McBSP 兼容
- 全双工通信
- 允许连续数据流的双缓冲数据寄存器
- 用于接收和发送的独立成帧和时钟
- 外部移位时钟生成或者一个内部可设定频率移位时钟
- 包括 8，12，16，20，24，或者 32 位在内的宽数据尺寸选择
- 以 LSB 或者 MSB 开头的 8 位数据传输
- 用于帧同步和数据时钟的可编程极性
- 高度可编程内部时钟和帧生成
- 到工业标准 CODEG、模拟接口芯片 (AIC)、和其它串行连接的 A/D 和 D/A 器件的直接接口
- 与 SPI 兼容器件一起工作
- McBSP 上支持下列应用接口:
  - -T1/E1 成帧器
  - -符合 IOM-2 的器件
  - -AC97 - 兼容器件 (提供所需的多相位帧同步功能。)
  - -IIS-兼容器件
  - -SPI
- McBSP 时钟速率:

$$CLKG = \frac{CLKSRG}{(1 + CLKGDV)}$$

CLKSRG 时钟源可以是 LSPCLK、CLKX 或者 CLKR。串行端口性能受到 I/O 缓冲器开关速度的影响。内部预分频器必须根据使用的时钟源进行调整，外设速度将低于 I/O 缓冲器速度。

| 注意 |
| --- |
| 参考第 5.9 节了解最大 I/O 引脚切换速度。 |

| 注意 |
| --- |
| 在 80 脚封装上，只支持 McBSP 的时钟停止模式 SPI（clock-stop mode）。 |

McBSP 模块框图如图 5-32 所示:

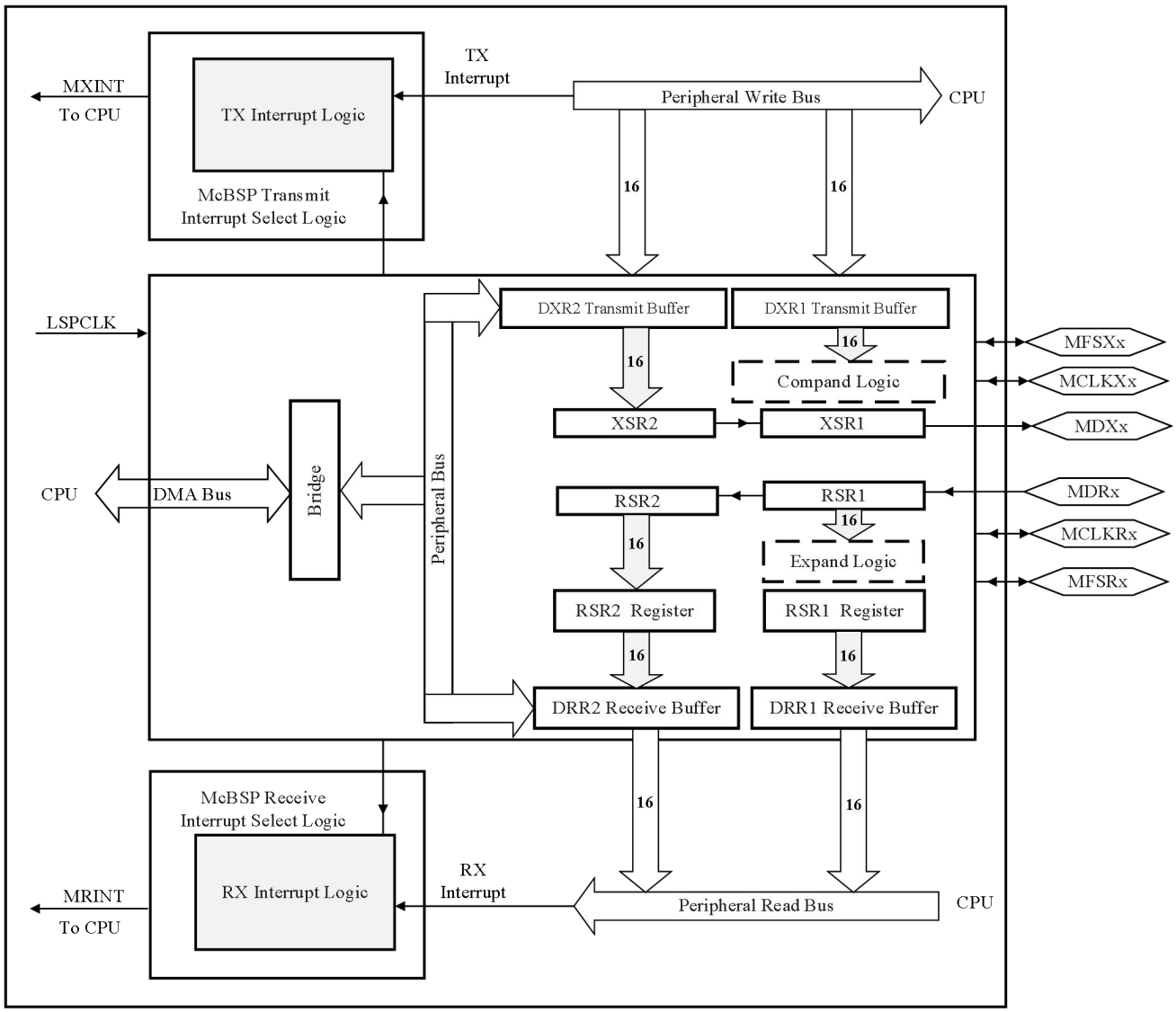



**图 5-32 McBSP 模块**

表 5-28 所示为 McBSP 寄存器的汇总表。

**表 5-28 McBSP 寄存器汇总表**

| 名称 | McBSP-A 地址 | 类型 | 复位值 | 说明 |
|---|---|---|---|---|
| **数据寄存器，接收、发送** | | | | |
| DRR2 | 0x5000 | R | 0x0000 | McBSP 数据接收寄存器 2 |
| DRR1 | 0x5001 | R | 0x0000 | McBSP 数据接收寄存器 1 |
| DXR2 | 0x5002 | W | 0x0000 | McBSP 数据发送寄存器 2 |
| DXR1 | 0x5003 | W | 0x0000 | McBSP 数据发送寄存器 1 |
| **McBSP 控制寄存器** | | | | |
| SPCR2 | 0x5004 | R/W | 0x0000 | McBSP 串行端口控制寄存器 2 |
| SPCR1 | 0x5005 | R/W | 0x0000 | McBSP 串行端口控制寄存器 1 |
| RCR2 | 0x5006 | R/W | 0x0000 | McBSP 接收控制寄存器 2 |
| RCR1 | 0x5007 | R/W | 0x0000 | McBSP 接收控制寄存器 1 |
| XCR2 | 0x5008 | R/W | 0x0000 | McBSP 发送控制寄存器 2 |
| XCR1 | 0x5009 | R/W | 0x0000 | McBSP 发送控制寄存器 1 |
| SRGR2 | 0x500A | R/W | 0x0000 | McBSP 采样率生成器寄存器 2 |
| SRGR1 | 0x500B | R/W | 0x0000 | McBSP 采样率生成器寄存器 1 |
| **多通道控制寄存器** | | | | |
| MCR2 | 0x500C | R/W | 0x0000 | McBSP 多通道寄存器 2 |
| MCR1 | 0x500D | R/W | 0x0000 | McBSP 多通道寄存器 1 |
| RCERA | 0x500E | R/W | 0x0000 | McBSP 接收通道使能寄存器分区 A |
| RCERB | 0x500F | R/W | 0x0000 | McBSP 接收通道使能寄存器分区 B |
| XCERA | 0x5010 | R/W | 0x0000 | McBSP 发送通道使能寄存器分区 A |
| XCERB | 0x5011 | R/W | 0x0000 | McBSP 发送通道使能寄存器分区 B |
| PCR | 0x5012 | R/W | 0x0000 | McBSP 引脚控制寄存器 |
| RCERC | 0x5013 | R/W | 0x0000 | McBSP 接收通道使能寄存器分区 C |
| RCERD | 0x5014 | R/W | 0x0000 | McBSP 接收通道使能寄存器分区 D |
| XCERC | 0x5015 | R/W | 0x0000 | McBSP 发送通道使能寄存器分区 C |
| XCERD | 0x5016 | R/W | 0x0000 | McBSP 发送通道使能寄存器分区 D |
| RCERE | 0x5017 | R/W | 0x0000 | McBSP 接收通道使能寄存器分区 E |
| RCERF | 0x5018 | R/W | 0x0000 | McBSP 接收通道使能寄存器分区 F |
| XCERE | 0x5019 | R/W | 0x0000 | McBSP 发送通道使能寄存器分区 E |
| XCERF | 0x501A | R/W | 0x0000 | McBSP 发送通道使能寄存器分区 F |
| RCERG | 0x501B | R/W | 0x0000 | McBSP 接收通道使能寄存器分区 G |
| RCERH | 0x501C | R/W | 0x0000 | McBSP 接收通道使能寄存器分区 H |
| XCERG | 0x501D | R/W | 0x0000 | McBSP 发送通道使能寄存器分区 G |
| XCERH | 0x501E | R/W | 0x0000 | McBSP 发送通道使能寄存器分区 H |
| MFFINT | 0x5023 | R/W | 0x0000 | McBSP 中断使能寄存器 |

## 5.9.5.2. McBSP 电气数据/时序

### 5.9.5.2.1.McBSP传输和接收时序

#### 5.9.5.2.1.1. McBSP 时序要求

| 编号[1][2] | | | | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|---|---|
| | McBSP 模块时钟 (CLKG, CLKX, CLKR) 范围 | | | 1 | | kHz |
| | | | | | 20[3][4] | MHz |
| | McBSP 模块周期时间 (CLKG, CLKX, CLKR) 范围 | | | 50[4] | | ns |
| | | | | | 1 | ms |
| M11 | $t_{c(CKRX)}$ | 周期时间， CLKR/X | CLKR/X 外部 | 2P | | ns |
| M12 | $t_{w(CKRX)}$ | 脉冲持续时间， CLKR/X 高电平或者 CLKR/X 低电平 | CLKR/X 外部 | P – 7 | | ns |
| M13 | $t_{r(CKRX)}$ | 上升时间， CLKR/X | CLKR/X 外部 | | 7 | ns |
| M14 | $t_{f(CKRX)}$ | 下降时间，CLKR/X | CLKR/X 外部 | | 7 | ns |
| M15 | $t_{su(FRH-CKRL)}$ | 建立时间，在 CLKR 低电平之前，外部 FSR 为高电平 | CLKR 内部 | 18 | | ns |
| | | | CLKR 外部 | 2 | | |
| M16 | $t_{h(CKRL-FRH)}$ | 保持时间，在CLKR 低电平之后，外部 FSR 为高电平 | CLKR 内部 | 0 | | ns |
| | | | CLKR 外部 | 6 | | |
| M17 | $t_{su(DRV-CKRL)}$ | 建立时间，在 CLKR 低电平之前，DR 有效的时间 | CLKR 内部 | 18 | | ns |
| | | | CLKR 外部 | 2 | | |
| M18 | $t_{h(CKRL-DRV)}$ | 保持时间，在 CLKR 低电平之后，DR 有效的时间 | CLKR 内部 | 0 | | ns |
| | | | CLKR 外部 | 6 | | |
| M19 | $t_{su(FXH-CKXL)}$ | 建立时间，在 CLKX 低电平之前，外部 FSX 为高电平 | CLKX 内部 | 18 | | ns |
| | | | CLKX 外部 | 2 | | |
| M20 | $t_{h(CKXL-FXH)}$ | 保持时间，在 CLKX 低电平之后，外部 FSX 为高电平 | CLKX 内部 | 0 | | ns |
| | | | CLKX 外部 | 6 | | |

(1) 极性位 CLKRP = CLKXP = FSRP = FSXP = 0。如果任一信号的极性被反转，那么该信号的时序基准也被反转。

(2) 2P = 1/CLKG，单位为 ns。CLKG 是采样率发生器复用的输出。CLKG = CLKSRG/(1 + CLKGDV)。CLKSRG 可由 LSPCLK、CLKX 和 CLKR 供源。 CLKSRG ≤ (SYSCLKOUT/2)。McBSP 的性能受到 I/O 缓冲器开关速度的限制。

(3) 内部时钟预分频器必须被调整，这样的话，McBSP 时钟 (CLKG、CLKX、CLKR) 速度不会大于 I/O 缓冲器速度限制 (20 MHz)。

(4) 用于内部 CLKR 的最大 McBSP 模块时钟基准减少为 10 MHz。

### 5.9.5.2.1.2. McBSP 开关特性

在建议运行条件下测得（除非额外注明）

| 编号 | 参数(1) (2) | | | | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|---|---|---|
| 1 | $t_{c(CKRX)}$ | 周期时间，CLKR/X | | | 2P | | ns |
| 2 | $t_{w(CKRXH)}$ | 脉冲持续时间，CLKR/X 高电平的时间 | | CLKR/X 内部 | D–5(3) | D + 5(3) | ns |
| 3 | $t_{w(CKRXL)}$ | 脉冲持续时间，CLKR/X 低电平的时间 | | CLKR/X 内部 | C–5(3) | C + 5(3) | ns |
| 4 | $t_{d(CKRH-FRV)}$ | 延迟时间，CLKR 高电平到内部 FSR 有效的时间 | | CLKR 内部 | 0 | 4 | ns |
| | | | | CLKR 外部 | 3 | 27 | |
| 5 | $t_{d(CKXH-FXV)}$ | 延迟时间，CLKX 高电平到内部 FSX 有效的时间 | | CLKX 内部 | 0 | 4 | ns |
| | | | | CLKX 外部 | 3 | 27 | |
| 6 | $t_{dis(CKXH-DXHZ)}$ | 禁用时间， CLKX 高电平到 DX 在最后一个数据位后为高阻抗的时间 | | CLKX 内部 | | 8 | ns |
| | | | | CLKX 外部 | | 14 | |
| 7 | $t_{d(CKXH-DXV)}$ | 延迟时间，CLKX 高电平到 DX 有效的时间。<br>这应用于除了第一个位之外的所有被发送的位。 | | CLKX 内部 | | 9 | ns |
| | | | | CLKX 外部 | | 28 | |
| | | 延迟时间，CLKX 高电平到 DX 有效的时间<br>当处于数据延迟1 或者2 （XDATDLY = 01b 或者 10b）模式时，只应用于发送的第一个位。 | DXENA = 0 | CLKX 内部 | | 8 | |
| | | | | CLKX 外部 | | 4 | |
| | | | DXENA = 1 | CLKX 内部 | | P + 8 | |
| | | | | CLKX 外部 | | P + 14 | |
| 8 | $t_{en(CKXH-DX)}$ | 使能时间，CLKX 高电平待 DX 被驱动的时间<br>当处于数据延迟1或者2（XDATDLY = 01b 或者 10b）模式时，只应用于发送的第一个位。 | DXENA = 0 | CLKX 内部 | 0 | | ns |
| | | | | CLKX 外部 | 6 | | |
| | | | DXENA = 1 | CLKX 内部 | P | | |
| | | | | CLKX 外部 | P + 6 | | |
| 9 | $t_{d(FXH-DXV)}$ | 延迟时间，FSX 高电平到 DX 有效的时间<br>当处于数据延迟0（XDATDLY = 00b）模式时，只应用于发送的第一个位。 | DXENA = 0 | FSX 内部 | | 8 | ns |
| | | | | FSX 外部 | | 14 | |
| | | | DXENA = 1 | FSX 内部 | | P + 8 | |
| | | | | FSX 外部 | | P + 14 | |
| 10 | $t_{en(FXH-DX)}$ | 使能时间，FSX 高电平到 DX 驱动的时间<br>当处于数据延迟0（XDATDLY = 00b）模式时，只应用于发送的第一个位。 | DXENA = 0 | FSX 内部 | 0 | | ns |
| | | | | FSX 外部 | 6 | | |
| | | | DXENA = 1 | FSX 内部 | P | | |
| | | | | FSX 外部 | P + 6 | | |

(1) 极性位 CLKRP = CLKXP = FSRP = FSXP = 0。如果任一信号的极性被反转，那么那个信号的定时基准也被反转。

(2) 2P = 1/CLKG，单位为 ns。

(3) C = CLKRX 低脉冲宽度 = P
D = CLKRX 高脉冲宽度 = P

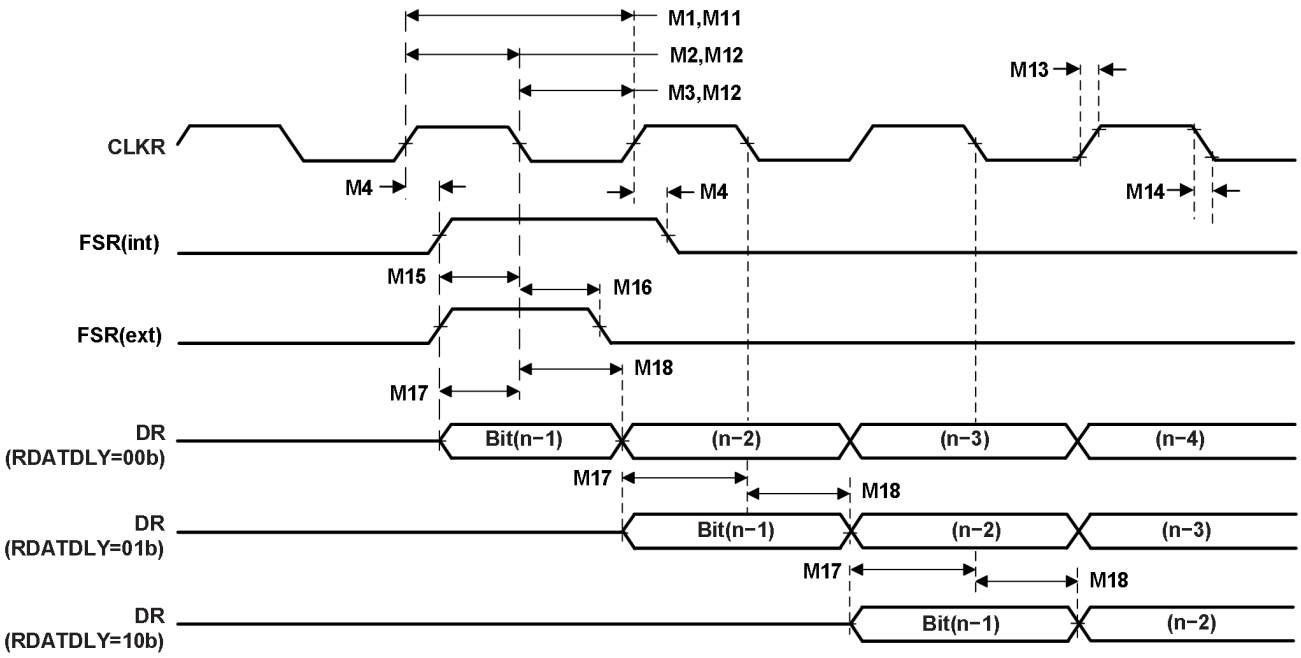



**图 5-33 McBSP 接收时序**

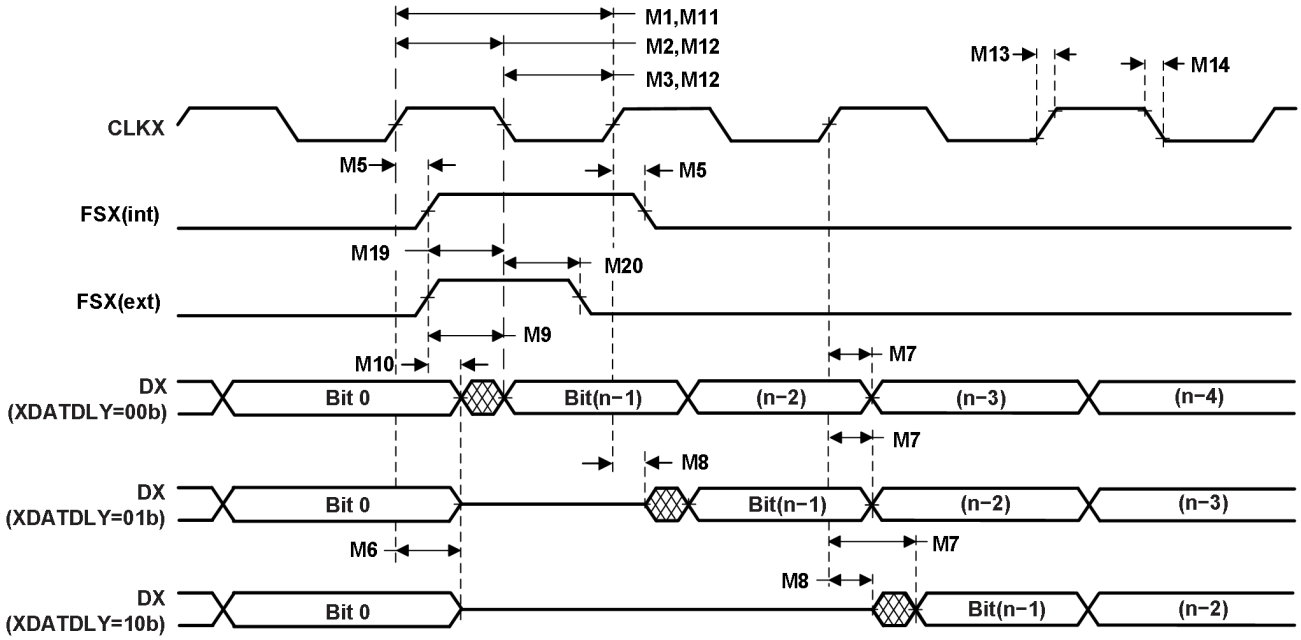



**图 5-34 McBSP 发送时序**

### 5.9.5.3. McBSP 作为 SPI 主机或从机时序

#### 5.9.5.3.1. McBSP 作为 SPI 主机或从机时序要求 (CLKSTP = 10b, CLKXP = 0)

| 编号[1] | | | 主机 | | 从机 | | 单位 |
|---|---|---|---|---|---|---|---|
| | | | 最小值 | 最大值 | 最小值 | 最大值 | |
| M30 | $t_{su(DRV-CKXL)}$ | 建立时间，在 CLKX 低电平之前，DR 有效的时间 | 30 | | 8P – 10 | | ns |
| M31 | $t_{h(CKXL-DRV)}$ | 保持时间，在 CLKX 低电平之后，DR 有效的时间 | 1 | | 8P – 10 | | ns |
| M32 | $t_{su(BFXL-CKXH)}$ | 建立时间，在 CLKX 高电平之前，FSX 为低电平的时间 | | | 8P + 10 | | ns |
| M33 | $t_{c(CKX)}$ | 周期时间，CLKX | 2P[2] | | 16P | | ns |

(1) 对于所有的 SPI 从机模式，CLKX 必须至少是 8 个 CLKG 周期。此外，通过设置 CLKSM = CLKGDV = 1，CLKG 应该为 LSPCLK/2 。

(2) 2P = 1/CLKG

#### 5.9.5.3.2. McBSP 作为 SPI 主机或从机开关特性 (CLKSTP = 10b, CLKXP = 0)

在建议运行条件下测得（除非额外注明）

| 编号 | 参数 | | 主机 | | 从机 | | 单位 |
|---|---|---|---|---|---|---|---|
| | | | 最小值 | 最大值 | 最小值 | 最大值 | |
| M24 | $t_{h(CKXL-FXL)}$ | 保持时间， CLKX 低电平之后，FSX 为低电平的时间 | 2P[1] | | | | ns |
| M25 | $t_{d(FXL-CKXH)}$ | 延迟时间，FSX 低电平到 CLKX 变为高电平的时间 | P | | | | ns |
| M26 | $t_{d(CKXH-DXV)}$ | 延迟时间，CLKX 高电平到 DX 有效的时间 | –2 | 0 | 3P + 6 | 5P + 20 | ns |
| M28 | $t_{dis(FXH-DXHZ)}$ | 禁用时间，从 FSX 高电平到最后一个数据位后 DX 高阻抗的时间 | 6 | | 6P + 6 | | ns |
| M29 | $t_{d(FXL-DXV)}$ | 延迟时间，FSX 低电平到 DX 有效的时间 | 6 | | 4P + 6 | | ns |

(1) 2P = 1/CLKG

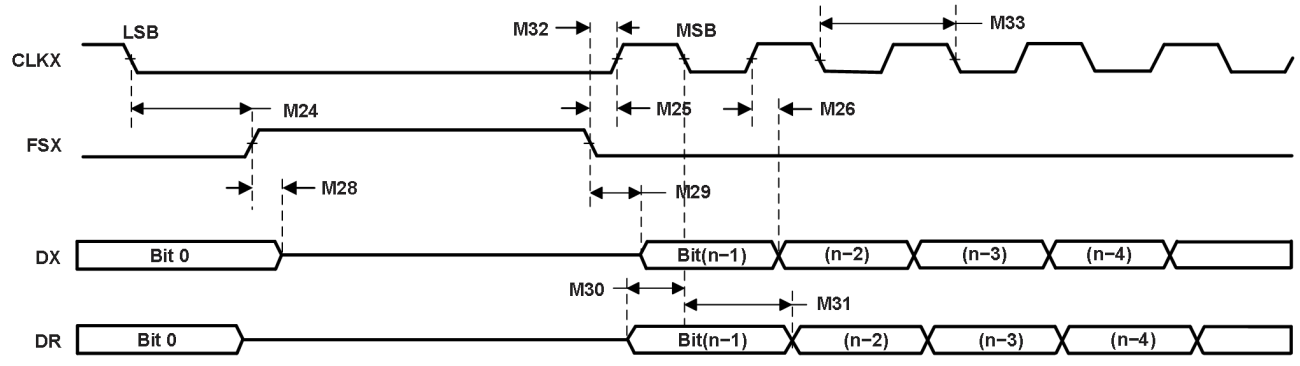



**图 5-35 McBSP 作为 SPI 主或从时序：CLKSTP=10b, CLKXP=0**

### 5.9.5.3.3. McBSP 作为 SPI 主机或从机时序要求 (CLKSTP = 11b, CLKXP = 0)

| 编号[1] | | | 主机 | | 从机 | | 单位 |
|---|---|---|---|---|---|---|---|
| | | | 最小值 | 最大值 | 最小值 | 最大值 | |
| M39 | $t_{su(DRV-CKXH)}$ | 建立时间，在 CLKX 高电平之前，DR 有效的时间 | 30 | | 8P –10 | | ns |
| M40 | $t_{h(CKXH-DRV)}$ | 保持时间，在 CLKX 高电平之后，DR 有效的时间 | 1 | | 8P –10 | | ns |
| M41 | $t_{su(FXL-CKXH)}$ | 建立时间，在 CLKX 高电平之前，FSX 为低电平的时间 | | | 16P + 10 | | ns |
| M42 | $t_{c(CKX)}$ | 周期时间，CLKX | 2P[2] | | 16P | | ns |

(1) 对于所有的 SPI 从机模式，CLKX 必须至少是 8 个 CLKG 周期。此外，通过设置 CLKSM = CLKGDV = 1，CLKG 应该为 LSPCLK/2。
(2) 2P = 1/CLKG

### 5.9.5.3.4. McBSP 作为 SPI 主机或从机开关特性 (CLKSTP = 11b, CLKXP = 0)

在建议运行条件下测得（除非额外注明）

| 编号 | 参数 | | 主机 | | 从机 | | 单位 |
|---|---|---|---|---|---|---|---|
| | | | 最小值 | 最大值 | 最小值 | 最大值 | |
| M34 | $t_{h(CKXL-FXL)}$ | 保持时间，CLKX 低电平之后，FSX 为低电平的时间 | P | | | | ns |
| M35 | $t_{d(FXL-CKXH)}$ | 延迟时间，FSX 低电平到 CLKX 变为高电平的时间 | 2P[1] | | | | ns |
| M36 | $t_{d(CKXL-DXV)}$ | 延迟时间，CLKX 低电平到 DX 有效的时间 | –2 | 0 | 3P + 6 | 5P + 20 | ns |
| M37 | $t_{dis(CKXL-DXHZ)}$ | 禁用时间，从 CLKX 低电平到最后一个数据位后 DX 高阻抗的时间 | P + 6 | | 7P + 6 | | ns |
| M38 | $t_{d(FXL-DXV)}$ | 延迟时间，FSX 低电平到 DX 有效时的时间 | 6 | | 4P + 6 | | ns |

(1) 2P = 1/CLKG

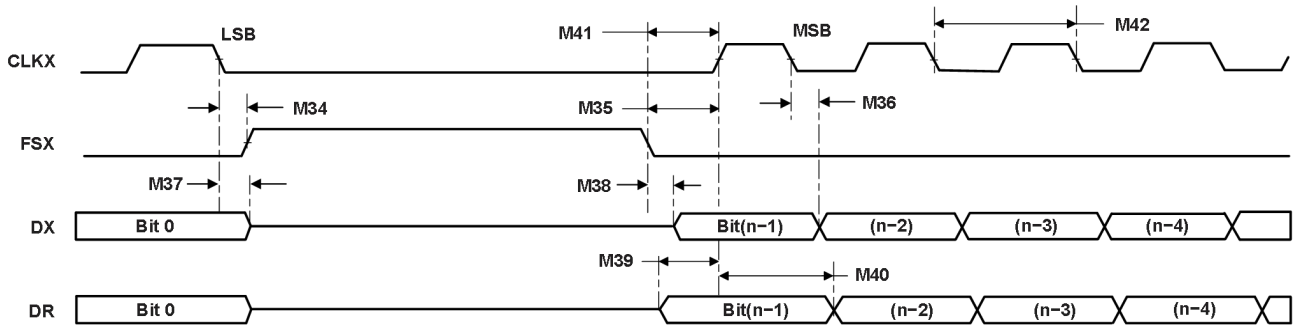



**图 5-36 McBSP 作为 SPI 主或从时序：CLKSTP=11b, CLKXP=0**

### 5.9.5.3.5. McBSP 作为 SPI 主机或从机时序要求 (CLKSTP = 10b, CLKXP = 1)

| 编号(1) | | | 主机 | | 从机 | | 单位 |
|---|---|---|---|---|---|---|---|
| | | | 最小值 | 最大值 | 最小值 | 最大值 | |
| M49 | $t_{su(DRV-CKXH)}$ | 建立时间，在 CLKX 高电平之前，DR 有效的时间 | 30 | | 8P –10 | | ns |
| M50 | $t_{h(CKXH-DRV)}$ | 保持时间，在 CLKX 高电平之后，DR 有效的时间 | 1 | | 8P –10 | | ns |
| M51 | $t_{su(FXL-CKXL)}$ | 建立时间，在 CLKX 低电平之前，FSX 为低电平的时间 | | | 8P + 10 | | ns |
| M52 | $t_{c(CKX)}$ | 周期时间， CLKX | 2P(2) | | 16P | | ns |

(1) 对于所有的 SPI 从机模式，CLKX 必须至少是 8 个 CLKG 周期。此外，通过设置 CLKSM = CLKGDV = 1，CLKG 应该为 LSPCLK/2。

(2) 2P = 1/CLKG

### 5.9.5.3.6. McBSP 作为 SPI 主机或从机开关特性 (CLKSTP = 10b, CLKXP = 1)

在建议运行条件下测得（除非额外注明）

| 编号 | 参数 | | 主机 | | 从机 | | 单位 |
|---|---|---|---|---|---|---|---|
| | | | 最小值 | 最大值 | 最小值 | 最大值 | |
| M43 | $t_{h(CKXH-FXL)}$ | 保持时间，CLKX 高电平之后，FSX 为低电平的时间 | 2P(1) | | | | ns |
| M44 | $t_{d(FXL-CKXL)}$ | 延迟时间，FSX 低电平到 CLKX 变为低电平的时间 | P | | | | ns |
| M45 | $t_{d(CKXL-DXV)}$ | 延迟时间，CLKX 低电平到 DX 有效的时间 | –2 | 0 | 3P + 6 | 5P + 20 | ns |
| M47 | $t_{dis(FXH-DXHZ)}$ | 禁用时间，从 FSX 高电平到最后一个数据位后 DX 高阻抗的时间 | 6 | | 6P + 6 | | ns |
| M48 | $t_{d(FXL-DXV)}$ | 延迟时间，FSX 低电平到 DX 有效时的时间 | 6 | | 4P + 6 | | ns |

(1) 2P = 1/CLKG

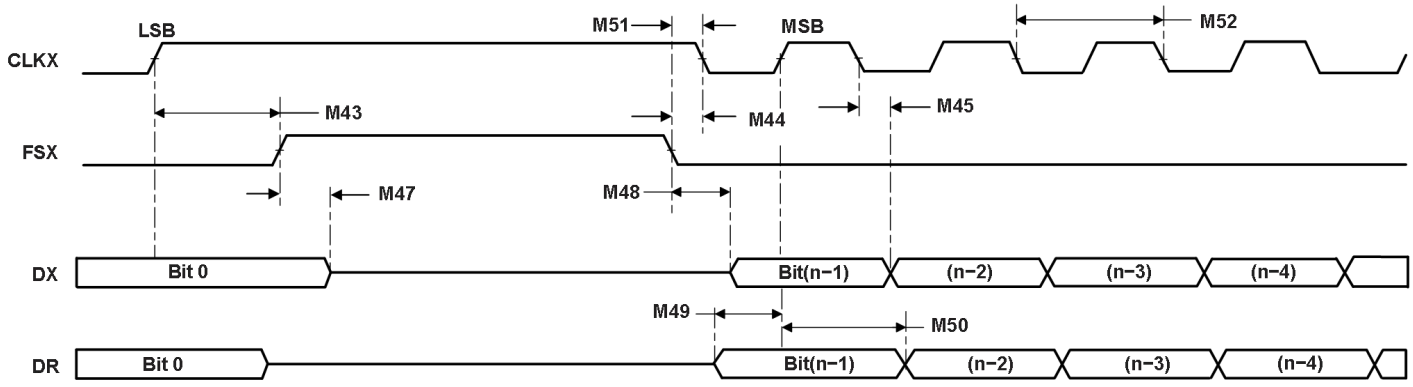



**图 5-37 McBSP 作为 SPI 主或从时序：CLKSTP=10b, CLKXP=1**

### 5.9.5.3.7. McBSP 作为 SPI 主机或从机时序要求 (CLKSTP = 11b, CLKXP = 1)

| 编号[1] | | | 主机 | | 从机 | | 单位 |
|---|---|---|---|---|---|---|---|
| | | | 最小值 | 最大值 | 最小值 | 最大值 | |
| M58 | $t_{su(DRV-CKXL)}$ | 建立时间，在 CLKX 低电平之前，DR 有效的时间 | 30 | | 8P – 10 | | ns |
| M59 | $t_{h(CKXL-DRV)}$ | 保持时间，在 CLKX 低电平之后，DR 有效的时间 | 1 | | 8P – 10 | | ns |
| M60 | $t_{su(FXL-CKXL)}$ | 建立时间，在 CLKX 低电平之前，FSX 为低电平的时间 | | | 16P + 10 | | ns |
| M61 | $t_{c(CKX)}$ | 周期时间， CLKX | 2P[2] | | 16P | | ns |

(1) 对于所有的 SPI 从机模式，CLKX 必须至少是 8 个 CLKG 周期。此外，通过设置 CLKSM = CLKGDV = 1，CLKG 应该为 LSPCLK/2 。

(2) 2P = 1/CLKG

### 5.9.5.3.8. McBSP 作为 SPI 主机或从机开关特性 (CLKSTP = 11b, CLKXP = 1)

在建议运行条件下测得（除非额外注明）(1)

| 编号 | 参数 | | 主机 | | 从机 | | 单位 |
|---|---|---|---|---|---|---|---|
| | | | 最小值 | 最大值 | 最小值 | 最大值 | |
| M53 | $t_{h(CKXH-FXL)}$ | 保持时间，CLKX 高电平之后，FSX 为低电平的时间 | P | | | | ns |
| M54 | $t_{d(FXL-CKXL)}$ | 延迟时间，FSX 低电平到 CLKX 变为低电平的时间 | 2P[1] | | | | ns |
| M55 | $t_{d(CKXH-DXV)}$ | 延迟时间，CLKX 高电平到 DX 有效的时间 | –2 | 0 | 3P + 6 | 5P + 20 | ns |
| M56 | $t_{dis(CKXH-DXHZ)}$ | 禁用时间，从 CLKX 高电平到最后一个数据位后 DX 高阻抗的时间 | P + 6 | | 7P + 6 | | ns |
| M57 | $t_{d(FXL-DXV)}$ | 延迟时间，FSX 低电平到 DX 有效时的时间 | 6 | | 4P + 6 | | ns |

(1) 2P = 1/CLKG

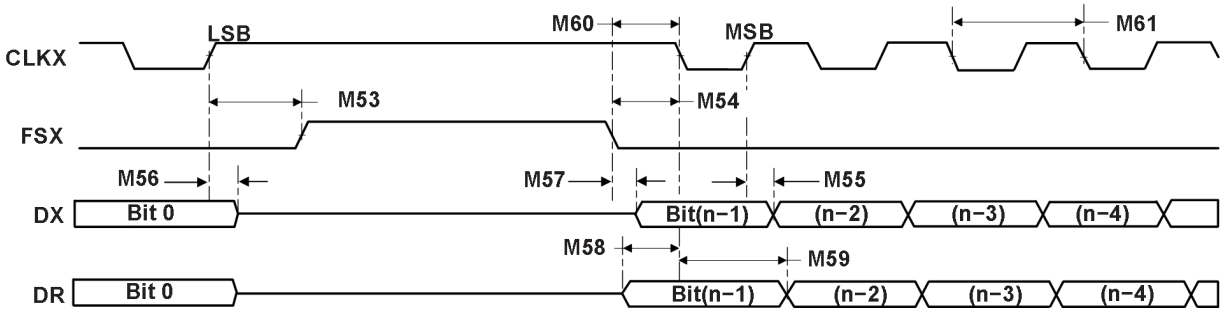



**图 5-38 McBSP 作为 SPI 主或从时序：CLKSTP=11b, CLKXP=1**

### 5.9.6. 增强型控制器局域网模块（eCAN)

CAN (eCAN-A)模块有以下特性：

- 完全符合 CAN 协议，版本 2.0B
- 支持数据传输速率最高 1Mbps
- 32 个邮箱，每个邮箱有以下特性：
  - -可配置为接收或者发送
  - -可配置为标准的或者扩展标识
  - -有可编程的接收屏蔽
  - -支持数据帧和远程帧
  - -由 0 到 8 字节的数据组成
  - -在接收和发送信息时使用了 32 位时间戳
  - -防止接收新消息
  - -保存传输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方案
  - -在传输或接收超时时采用可编程报警
- 低功耗模式
- 可编程的总线唤醒
- 自动响应远程帧需求
- 在仲裁失败或错误的情况下自动重传帧
- 由特定邮件同步的 32 位本地网络时间计数器（与邮箱 16 一起通信）
- 自测模式
  - -在环回模式下运行，接收自己的消息。提供了“虚拟”响应，从而消除了另一个节点提供响应位的需要。

注

对于 90 MHz 的 SYSCLKOUT，可能的最小比特率是 6.25 kbps。

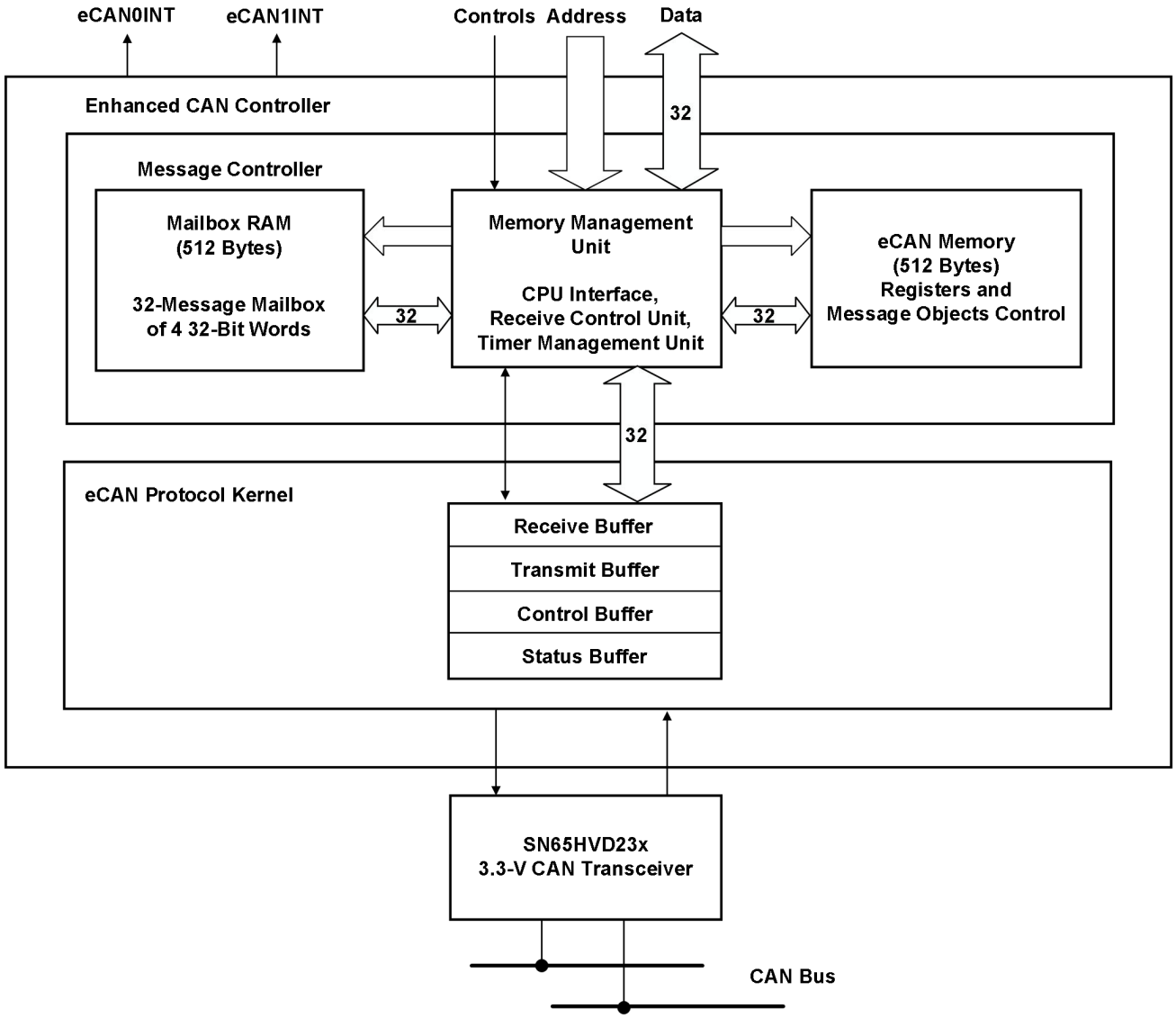



**图 5-39 eCAN 框图和接口电路**

**表 5-29 3.3V eCAN 收发器**

| 器件型号 | 电源电压 | 低功耗模式 | 斜坡控制 | VREF | 其他 | $T_A$ |
|---|---|---|---|---|---|---|
| SN65HVD230 | 3.3 V | 待机 | 可调节 | 支持 | - | - 40°C 至 85°C |
| SN65HVD230Q | 3.3 V | 待机 | 可调节 | 支持 | - | - 40°C 至 125°C |
| SN65HVD231 | 3.3 V | 休眠 | 可调节 | 支持 | - | - 40°C 至 85°C |
| SN65HVD231Q | 3.3 V | 休眠 | 可调节 | 支持 | - | - 40°C 至 125°C |
| SN65HVD232 | 3.3 V | 无 | 无 | 无 | - | - 40°C 至 85°C |
| SN65HVD232Q | 3.3 V | 无 | 无 | 无 | - | - 40°C 至 125°C |
| SN65HVD233 | 3.3 V | 待机 | 可调节 | 无 | 诊断环回 | - 40°C 至 125°C |
| SN65HVD234 | 3.3 V | 待机和休眠 | 可调节 | 无 | - | - 40°C 至 125°C |
| SN65HVD235 | 3.3 V | 待机 | 可调节 | 无 | 环回模式下的自动波特率 | - 40°C 至 125°C |
| ISO1050 | 3 - 5.5 V | 无 | 无 | 无 | 内置隔离 低 延迟 热关断 故障安全操作 主导超时 | - 55°C 至 105°C |

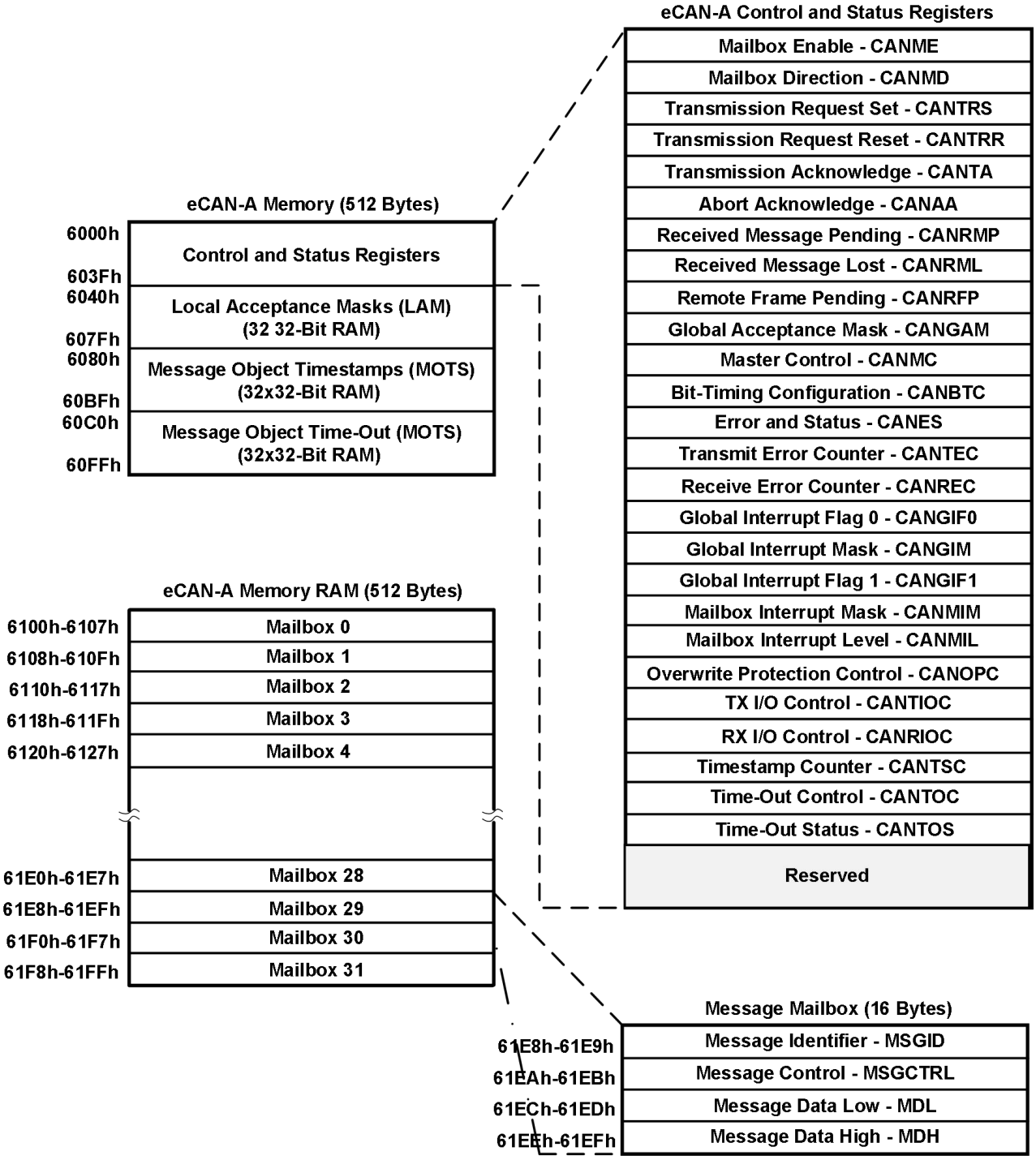



**图 5-40 eCAN-A 内存映射**

注　意

如果未在应用程序中使用 eCAN 模块，则可用的 RAM（LAM、MOTS、MOTO 和邮箱 RAM）可用作通用 RAM。为此，应使能 CAN 模块时钟。

使用表 5-30 中列出的 CAN 寄存器来配置和控制 CAN 控制器和消息对象。eCAN 控制寄存器仅支持 32 位读/写操作。邮箱 RAM 可以作为 16 位或 32 位访问。所有 32 位访问都与边界对齐。

**表 5-30 CAN 寄存器**

| 寄存器名称(1) | eCAN-A 地址 | 大小 (×32) | 描述 |
|---|---|---|---|
| CANME | 0x6000 | 1 | 邮箱使能 |
| CANMD | 0x6002 | 1 | 邮箱方向 |
| CANTRS | 0x6004 | 1 | 发送需求设置 |
| CANTRR | 0x6006 | 1 | 发送需求复位 |
| CANTA | 0x6008 | 1 | 传输确认 |
| CANAA | 0x600A | 1 | 中止确认 |
| CANRMP | 0x600C | 1 | 接收消息挂起 |
| CANRML | 0x600E | 1 | 接收消息丢失 |
| CANRFP | 0x6010 | 1 | 远程帧挂起 |
| CANGAM | 0x6012 | 1 | 全局接收屏蔽 |
| CANMC | 0x6014 | 1 | 主控 |
| CANBTC | 0x6016 | 1 | 位时序配置 |
| CANES | 0x6018 | 1 | 错误和状态 |
| CANTEC | 0x601A | 1 | 发送错误计数器 |
| CANREC | 0x601C | 1 | 接收错误计数器 |
| CANGIF0 | 0x601E | 1 | 全局中断标志0 |
| CANGIM | 0x6020 | 1 | 全局中断屏蔽 |
| CANGIF1 | 0x6022 | 1 | 全局中断标志1 |
| CANMIM | 0x6024 | 1 | 邮箱中断屏蔽 |
| CANMIL | 0x6026 | 1 | 邮箱中断级别 |
| CANOPC | 0x6028 | 1 | 覆盖保护控制 |
| CANTIOC | 0x602A | 1 | TX I/O 控制 |
| CANRIOC | 0x602C | 1 | RX I/O 控制 |
| CANTSC | 0x602E | 1 | 时间戳计数器 (SCC模式的保留位) |
| CANTOC | 0x6030 | 1 | 超时控制 (SCC模式的保留位) |
| CANTOS | 0x6032 | 1 | 超时状态 (SCC模式的保留位) |

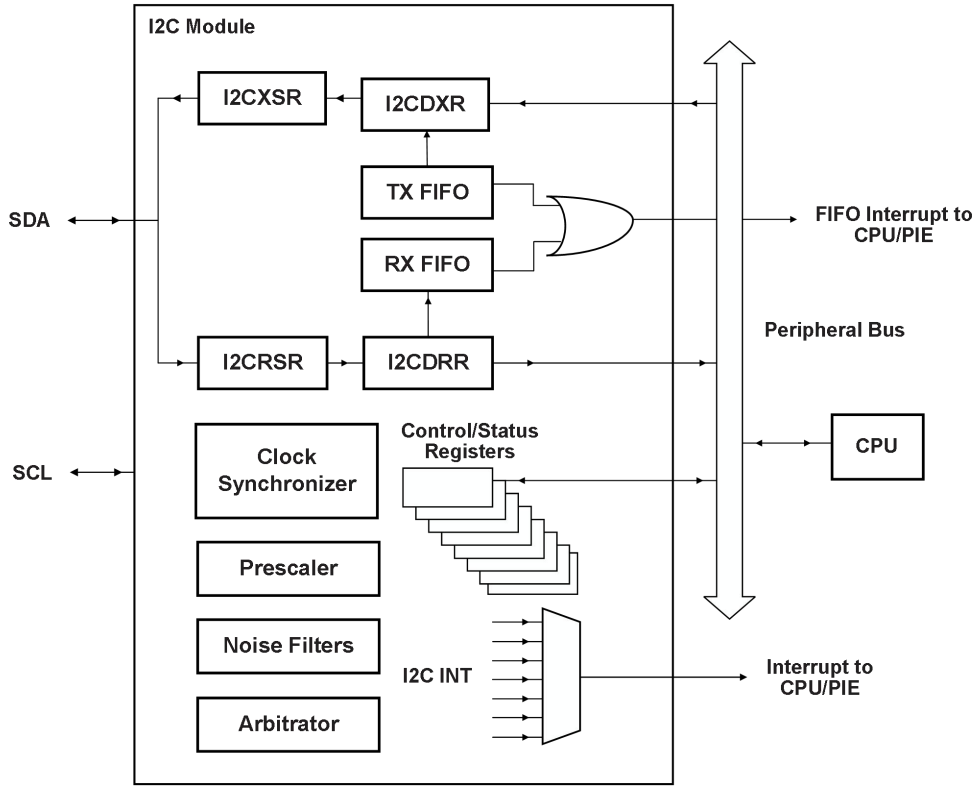
(1) 这些寄存器映射到外设帧 1。

### 5.9.7. 内部集成电路 (I2C)

器件包含一个 I2C 串行端口，图 5-41 显示了此器件内的 I2C 外设模块接口。

I2C 模块具有以下特性:

- 符合飞利浦半导体 I2C 总线规格（版本 2.1）：
  -支持 1 位至 8 位格式传输
  -7 位和 10 位寻址模式
  -广播呼叫功能
  -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 字接收 FIFO 和 一个 4 字发送 FIFO
- 可以由 CPU 使用的一个中断。这个中断可由下列条件中的一个生成:
  -发送数据准备就绪
  -接收数据准备就绪
  -寄存器访问准备就绪
  -没有接收到响应
  -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
- 全数据格式模式




A. I2C 寄存器以 SYSCLKOUT 速率访问。I2C 端口的内部时序和信号波形也是以 SYSCLKOUT 速率。

B. PCLKCRO 寄存器中的时钟使能位（I2CAENCLK）关闭 I2C 端口的时钟，用于低功耗操作。复位时 I2CAENCLK 为清除状态，表示外部内部时钟关闭。

**图 5-41 I2C 外设模块接口**

寄存器配置和 I2C 端口的控制操作如表 5-31 示。

**表 5-31 I2C-A 寄存器**

| 名称 | 地址 | EALLOW 保护 | 描述 |
|---|---|---|---|
| I2COAR | 0x7900 | No | I2C 自身的地址寄存器 |
| I2CIER | 0x7901 | No | I2C中断使能寄存器 |
| I2CSTR | 0x7902 | No | I2C 状态寄存器 |
| I2CCLKL | 0x7903 | No | I2C 时钟低电平时间分频器寄存器 |
| I2CCLKH | 0x7904 | No | I2C 时钟高电平时间分频器寄存器 |
| I2CCNT | 0x7905 | No | I2C 数据计数寄存器 |
| I2CDRR | 0x7906 | No | I2C 数据接收寄存器 |
| I2CSAR | 0x7907 | No | I2C 从器件地址寄存器 |
| I2CDXR | 0x7908 | No | I2C 数据发送寄存器 |
| I2CMDR | 0x7909 | No | I2C 模式寄存器 |
| I2CISRC | 0x790A | No | I2C 中断源寄存器 |
| I2CPSC | 0x790C | No | I2C 预分频器寄存器 |
| I2CFFTX | 0x7920 | No | I2C FIFO 发送寄存器 |
| I2CFFRX | 0x7921 | No | I2C FIFO 接收寄存器 |
| I2CRSR | – | No | I2C 接收移位寄存器（不可访问 CPU） |
| I2CXSR | – | No | I2C发送移位寄存器（不可访问 CPU） |

### 5.9.7.1. I2C 电气数据/时序

第 5.9.7.1.1 节说明了 I2C 的时序要求。第 5.9.7.1.2 节说明了 I2C 的开关特性。

#### 5.9.7.1.1. I2C时序要求

| 参数 | | | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|---|
| $t_{h(SDA-SCL)START}$ | 保持时间, START 条件, SCL 下降延迟在 SDA 下降后 | | 0.6 | | μs |
| $t_{su(SCL-SDA)START}$ | 建立时间, 重复 START, SCL 上升在 SDA 下降延迟之前 | | 0.6 | | μs |
| $t_{h(SCL-DAT)}$ | 保持时间, 数据在 SCL 下降之后 | | 0 | | μs |
| $t_{su(DAT-SCL)}$ | 建立时间, 数据在 SCL 上升之前 | | 100 | | ns |
| $t_{r(SDA)}$ | 上升时间, SDA | 输入容限 | 20 | 300 | ns |
| $t_{r(SCL)}$ | 上升时间, SCL | 输入容限 | 20 | 300 | ns |
| $t_{f(SDA)}$ | 下降时间, SDA | 输入容限 | 11.4 | 300 | ns |
| $t_{f(SCL)}$ | 下降时间 SCL | 输入容限 | 11.4 | 300 | ns |
| $t_{su(SCL-SDA)STOP}$ | 建立时间, STOP 条件, SCL 上升在 SDA 上升延迟之前 | | 0.6 | | μs |

#### 5.9.7.1.2. I2C开关特性

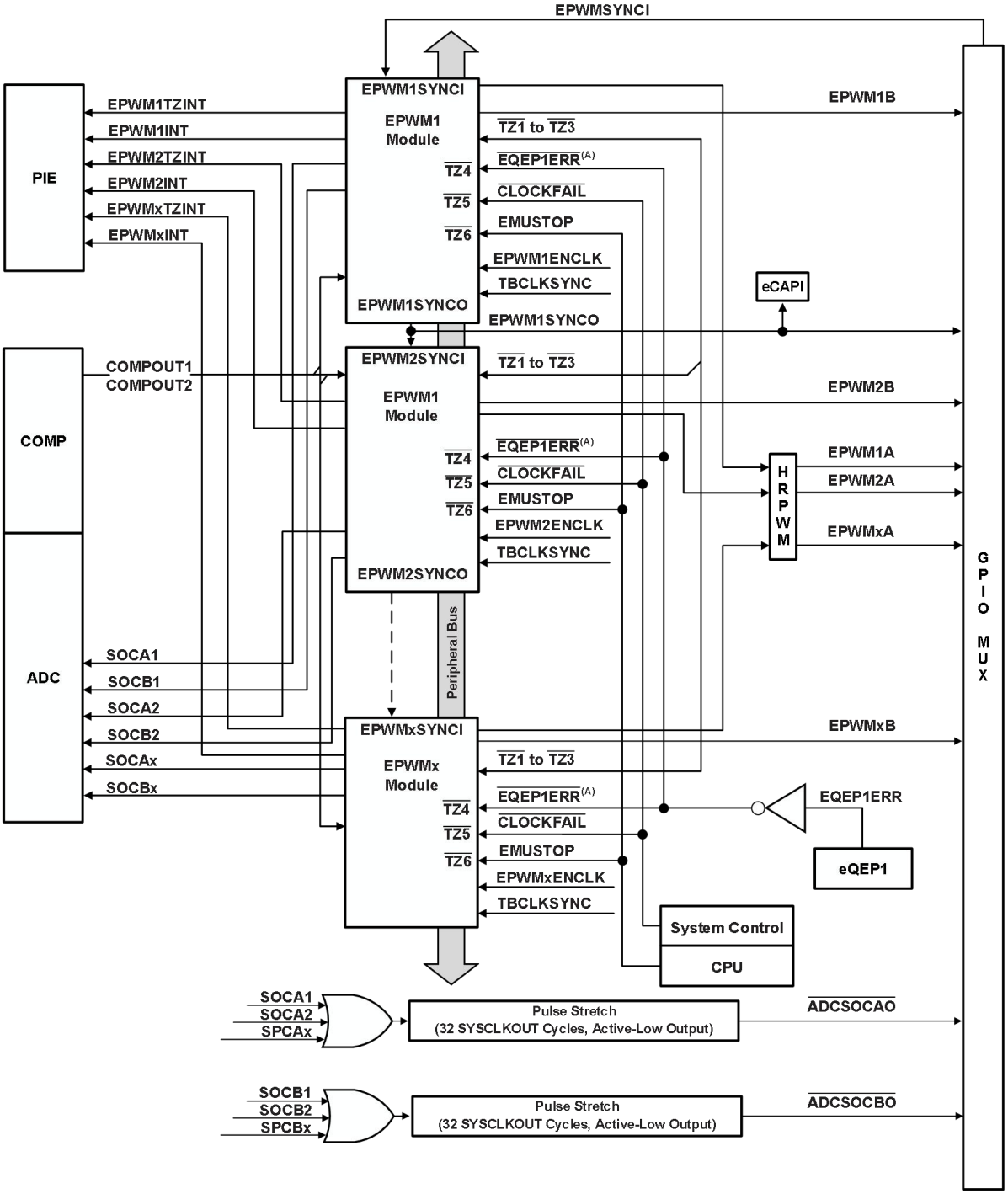
超出推荐的操作条件 (除非另有说明)

| 参数 | | 测试条件 | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|---|
| $f_{SCL}$ | SCL 时钟频率 | I2C 模块时钟频率为7-12MHz，其预分频器和时钟驱动寄存器被配置恰当。 | | 400 | kHz |
| $V_{il}$ | 低电平输入电压 | | | 0.3 $V_{DDIO}$ | V |
| $V_{ih}$ | 高电平输入电压 | | 0.7 $V_{DDIO}$ | | V |
| $V_{hys}$ | 输入滞后 | | 0.05 $V_{DDIO}$ | | V |
| $V_{ol}$ | 输出低压 | 3-mA灌电流 | 0 | 0.4 | V |
| $t_{LOW}$ | SCL时钟周期低电平 | I2C 模块时钟频率为7-12MHz，其预分频器和时钟驱动寄存器被配置恰当。 | 1.3 | | μs |
| $t_{HIGH}$ | SCL时钟周期高电平 | I2C 模块时钟频率为7-12MHz，其预分频器和时钟驱动寄存器配置适当。 | 0.6 | | μs |
| $I_I$ | 在0.1 VDDIO至0.9 VDDIO最大输入电压下的输入电流 | | –10 | 10 | μA |

### 5.9.8. 增强型 PWM 模块 (ePWM1 至 ePWM8)

这些器件包含多达 8 个增强型 PWM (ePWM) 模块。图 5-42 所示为多个 ePWM 模块的框图。图 5-43 所示为 ePWM 的内部子模块及信号互连。

表 5-31 和 表 5-32 列出了每个增强型 PWM 模块完整的寄存器设置。




A 该信号仅存在于带有 eQEP1 模块的器件上。

**图 5-42 ePWM**

**表 5-32 ePWM1–ePWM4 控制和状态寄存器**

| 名称 | PWM1 | ePWM2 | ePWM3 | ePWM4 | 大小 (×16)/ #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 | 时基计数寄存器 |
| TBPRD | 0x6805 | 0x6845 | 0x6885 | 0x68C5 | 1/1 | 时基周期寄存器 |
| TBPRDHR | 0x6806 | 0x6846 | 0x6886 | 0x68C6 | 1/1 | 时基周期HRPWM寄存器(1) |
| CMPCTL | 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 寄存器设置 |
| CMPB | 0x680A | 0x684A | 0x688A | 0x68CA | 1/1 | 计数器比较 B 寄存器设置 |
| 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 强制寄存器设置 |
| 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 | 可编程控制故障区数字比较寄存器 |
| TZCTL | 0x6814 | 0x6854 | 0x6894 | 0x68D4 | 1/0 | 可编程控制故障区控制寄存器(1) |
| TZEINT | 0x6815 | 0x6855 | 0x6895 | 0x68D5 | 1/0 | 可编程控制故障区中断使能寄存器(1) |
| TZFLG | 0x6816 | 0x6856 | 0x6896 | 0x68D6 | 1/0 | 可编程控制故障区标志寄存器 (1) |
| TZCLR | 0x6817 | 0x6857 | 0x6897 | 0x68D7 | 1/0 | 可编程控制故障区清除寄存器(1) |
| TZFRC | 0x6818 | 0x6858 | 0x6898 | 0x68D8 | 1/0 | 可编程控制故障区强制寄存器(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) |
| HRMSTEP | 0x6826 | - | - | - | 1/0 | HRPWM MEP 步进寄存器 |
| 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 寄存器镜像 |
| 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 寄存器镜像 |
| CMPAM | 0x682D | 0x686D | 0x68AD | 0x68ED | 1/W(2) | 比较 A 寄存器镜像 |
| 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) |

(1) 寄存器受 EALLOW 保护

(2) W = 写入影子寄存器

**表 5-33 ePWM5–ePWM8 控制和状态寄存器**

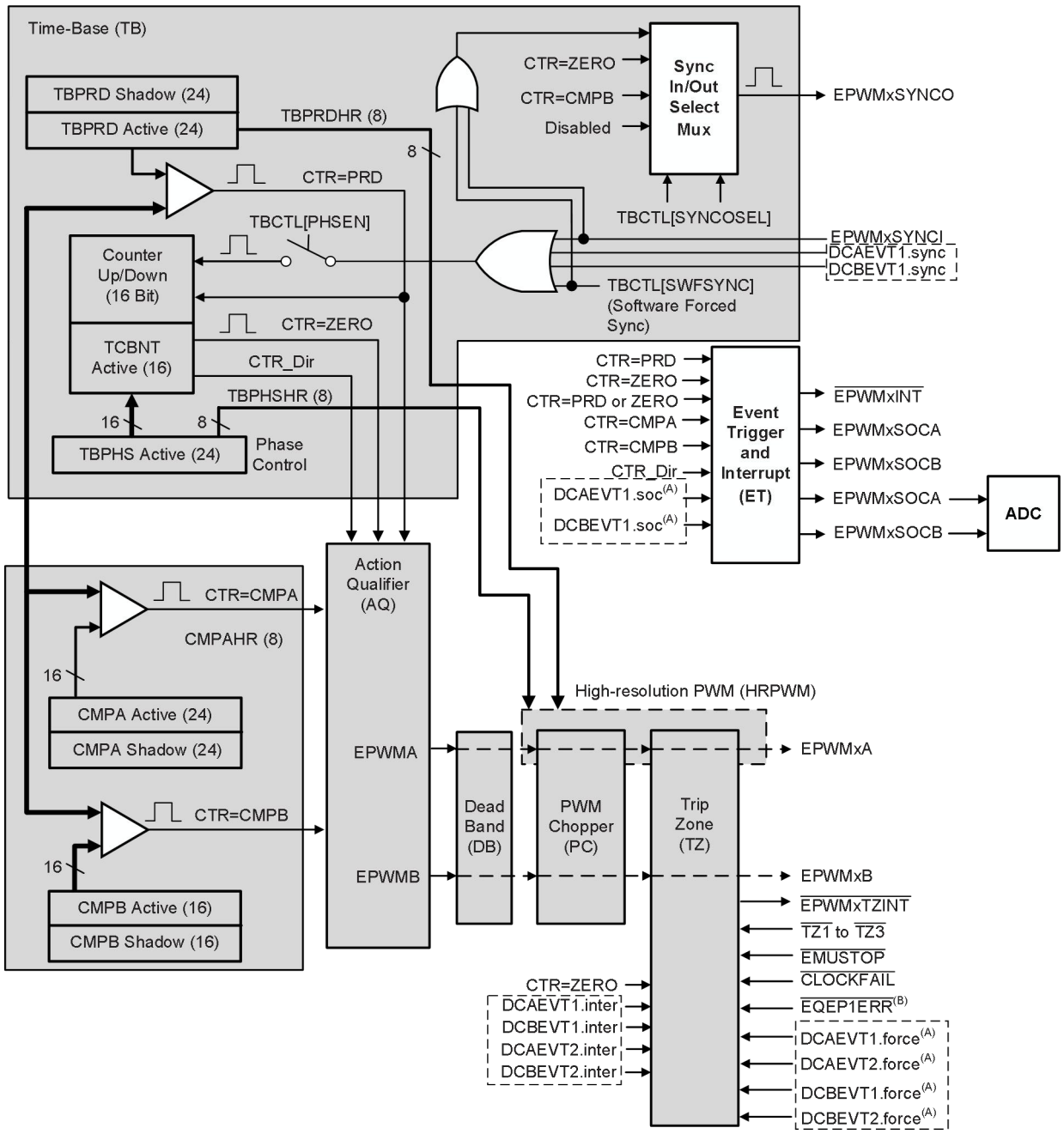
| 名称 | ePWM5 | ePWM6 | ePWM7 | ePWM8 | 大小 (×16)/ #SHADOW | 说明 |
|---|---|---|---|---|---|---|
| TBCTL | 0x6900 | 0x6940 | 0x6980 | 0x69C0 | 1/0 | 时基控制寄存器 |
| TBSTS | 0x6901 | 0x6941 | 0x6981 | 0x69C1 | 1/0 | 时基状态寄存器 |
| TBPHSHR | 0x6902 | 0x6942 | 0x6982 | 0x69C2 | 1/0 | 时基相位HRPWM寄存器 |
| TBPHS | 0x6903 | 0x6943 | 0x6983 | 0x69C3 | 1/0 | 时基相位寄存器 |
| TBCTR | 0x6904 | 0x6944 | 0x6984 | 0x69C4 | 1/0 | 时基计数寄存器 |
| TBPRD | 0x6905 | 0x6945 | 0x6985 | 0x69C5 | 1/1 | 时基周期寄存器 |
| TBPRDHR | 0x6906 | 0x6946 | 0x6986 | 0x69C6 | 1/1 | 时基周期HRPWM寄存器(1) |
| CMPCTL | 0x6907 | 0x6947 | 0x6987 | 0x69C7 | 1/0 | 计数器比较控制寄存器 |
| CMPAHR | 0x6908 | 0x6948 | 0x6988 | 0x69C8 | 1/1 | 时基比较A HRPWM寄存器 |
| CMPA | 0x6909 | 0x6949 | 0x6989 | 0x69C9 | 1/1 | 计数器比较 A 寄存器设置 |
| CMPB | 0x690A | 0x694A | 0x698A | 0x69CA | 1/1 | 计数器比较 B 寄存器设置 |
| AQCTLA | 0x690B | 0x694B | 0x698B | 0x69CB | 1/0 | 用于输出A 的操作限定器控制寄存器 |
| AQCTLB | 0x690C | 0x694C | 0x698C | 0x69CC | 1/0 | 用于输出B 的操作限定器控制寄存器 |
| AQSFRC | 0x690D | 0x694D | 0x698D | 0x69CD | 1/0 | 操作限定器软件强制寄存器 |
| AQCSFRC | 0x690E | 0x694E | 0x698E | 0x69CE | 1/1 | 操作限定器连续 S/W 强制寄存器设置 |
| DBCTL | 0x690F | 0x694F | 0x698F | 0x69CF | 1/1 | 死区生成器控制寄存器 |
| DBRED | 0x6910 | 0x6950 | 0x6990 | 0x69D0 | 1/0 | 死区生成器上升沿延迟计数寄存器 |
| DBFED | 0x6911 | 0x6951 | 0x6991 | 0x69D1 | 1/0 | 死区生成器下降沿延迟计数寄存器 |
| TZSEL | 0x6912 | 0x6952 | 0x6992 | 0x69D2 | 1/0 | 可编程控制故障区选择寄存器(1) |
| TZDCSEL | 0x6913 | 0x6953 | 0x6993 | 0x69D3 | 1/0 | 可编程控制故障区 数字比较寄存器 |
| TZCTL | 0x6914 | 0x6954 | 0x6994 | 0x69D4 | 1/0 | 可编程控制故障区控制寄存器(1) |
| TZEINT | 0x6915 | 0x6955 | 0x6995 | 0x69D5 | 1/0 | 可编程控制故障区中断使能寄存器(1) |
| TZFLG | 0x6916 | 0x6956 | 0x6996 | 0x69D6 | 1/0 | 可编程控制故障区标志寄存器 (1) |
| TZCLR | 0x6917 | 0x6957 | 0x6997 | 0x69D7 | 1/0 | 可编程控制故障区清除寄存器(1) |
| TZFRC | 0x6918 | 0x6958 | 0x6998 | 0x69D8 | 1/0 | 可编程控制故障区强制寄存器(1) |
| ETSEL | 0x6919 | 0x6959 | 0x6999 | 0x69D9 | 1/0 | 事件触发器选择寄存器 |
| ETPS | 0x691A | 0x695A | 0x699A | 0x69DA | 1/0 | 事件触发器预分频寄存器 |
| ETFLG | 0x691B | 0x695B | 0x699B | 0x69DB | 1/0 | 事件触发器标志寄存器 |
| ETCLR | 0x691C | 0x695C | 0x699C | 0x69DC | 1/0 | 事件触发器清除寄存器 |
| ETFRC | 0x691D | 0x695D | 0x699D | 0x69DD | 1/0 | 事件触发器强制寄存器 |
| PCCTL | 0x691E | 0x695E | 0x699E | 0x69DE | 1/0 | PWM 斩波器控制寄存器 |
| HRCNFG | 0x6920 | 0x6960 | 0x69A0 | 0x69E0 | 1/0 | HRPWM 配置寄存器(1) |
| HRMSTEP | - | - | - | - | 1/0 | HRPWM MEP 步进寄存器 |
| HRPCTL | 0x6928 | 0x6968 | 0x69A8 | 0x69E8 | 1/0 | 高分辨率周期控制寄存器(1) |
| TBPRDHRM | 0x692A | 0x696A | 0x69AA | 0x69EA | 1/W(2) | 时基周期 HRPWM 寄存器镜像 |
| TBPRDM | 0x692B | 0x696B | 0x69AB | 0x69EB | 1/W(2) | 时基周期寄存器镜像 |
| CMPAHRM | 0x692C | 0x696C | 0x69AC | 0x69EC | 1/W(2) | 比较 A HRPWM 寄存器镜像 |
| CMPAM | 0x692D | 0x696D | 0x69AD | 0x69ED | 1/W(2) | 比较 A 寄存器镜像 |
| DCTRIPSEL | 0x6930 | 0x6970 | 0x69B0 | 0x69F0 | 1/0 | 数字比较触发选择寄存器 (1) |
| DCACTL | 0x6931 | 0x6971 | 0x69B1 | 0x69F1 | 1/0 | 数字比较 A 控制寄存器(1) |

**表 5-33 ePWM5 – ePWM8 控制和状态寄存器 (续)**

| 名称 | ePWM5 | ePWM6 | ePWM7 | ePWM8 | 大小(×16)/ #SHADOW | 描述 |
|---|---|---|---|---|---|---|
| DCBCTL | 0x6932 | 0x6972 | 0x69B2 | 0x69F2 | 1/0 | 数字比较 B 控制寄存器(1) |
| DCFCTL | 0x6933 | 0x6973 | 0x69B3 | 0x69F3 | 1/0 | 数字比较滤波器控制寄存器(1) |
| DCCAPCT | 0x6934 | 0x6974 | 0x69B4 | 0x69F4 | 1/0 | 数字比较捕获控制寄存器(1) |
| DCFOFFSET | 0x6935 | 0x6975 | 0x69B5 | 0x69F5 | 1/1 | 数字比较滤波器偏置寄存器 |
| DCFOFFSETCNT | 0x6936 | 0x6976 | 0x69B6 | 0x69F6 | 1/0 | 数字比较滤波器偏置计数器寄存器 |
| DCFWINDOW | 0x6937 | 0x6977 | 0x69B7 | 0x69F7 | 1/0 | 数字比较滤波器窗口寄存器 |
| DCFWINDOWCNT | 0x6938 | 0x6978 | 0x69B8 | 0x69F8 | 1/0 | 数字比较滤波器窗口计数器寄存器 |
| DCCAP | 0x6939 | 0x6979 | 0x69B9 | 0x69F9 | 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 这些事件是由 Type-1 ePWM 数字比较（DC）子模块基于 COMPxOUT 和TZ信号的电平生成的。

B 此信号仅存在于带有 eQEP1 模块的器件上。

**图 5-43 关键内部信号互连的 ePWM 子模块**

### 5.9.8.1. 增强型 PWM 电气数据/时序

PWM 是指 ePWM1–8 上的 PWM 输出。第 5.9.8.1.1 节显示了 PWM 时序要求，第 5.9.8.1.2 节显示了开关特性。

#### 5.9.8.1.1. 增强型PWM时序要求

| 参数项 | | 最小值(1) | 最大值 | 单位 |
|---|---|---|---|---|
| $t_{w(SYCIN)}$ 同步输入脉宽 | 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 |

(1) 有关输入限定器参数的说明，请参考第 5.9.14.1.3.1 节。

#### 5.9.8.1.2. 增强型PWM 开关特性

在建议运行条件下测得（除非另有说明）

| 参数项 | | 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/低 | | 33.33 | | ns |
| $t_{w(SYNCOUT)}$ | 同步输出脉冲宽度 | | $8t_{c(SCO)}$ | | 周期 |
| $t_{d(PWM)tza}$ | 延迟时间，可编程控制故障区输入有效到 PWM 强制高电平的时间<br>延迟时间，可编程控制故障区输入有效到 PWM 强制低电平的时间 | 无引脚负载 | | 25 | ns |
| $t_{d(TZ-PWM)HZ}$ | 延迟时间，可编程控制故障区输入有效到PWM高阻态的时间 | | | 20 | ns |

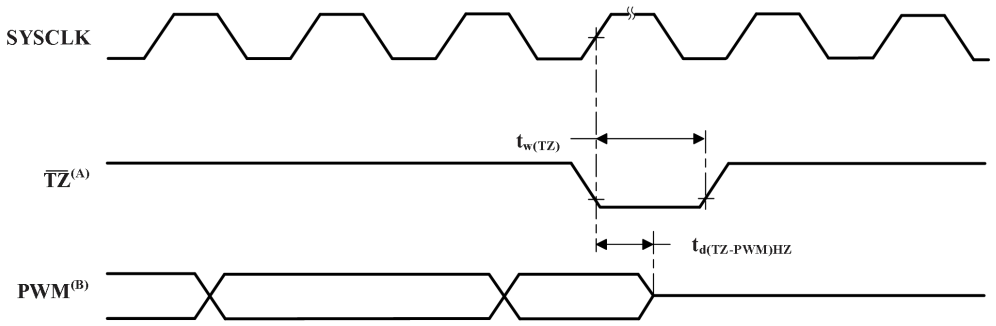
### 5.9.8.2. 可编程控制故障区输入时序

第 5.9.8.2.1 节列出了可编程控制故障区输入时序要求。图 5-44 显示了 PWM Hi-Z 特性。

#### 5.9.8.2.1. 可编程控制故障区输入时序要求

| | | 最小值(1) | 最大值 | 单位 |
|---|---|---|---|---|
| $t_{w(TZ)}$ 脉冲持续时间，$\overline{TZx}$ 输入低电平的时间 | 异步 | $2t_{c(TBCLK)}$ | | 周期 |
| | 同步 | $2t_{c(TBCLK)}$ | | 周期 |
| | 带有输入限定器 | $2t_{c(TBCLK)} + t_{w(IQSW)}$ | | 周期 |

(1) 有关输入限定器参数的说明，请参考第 8.9.15.1.2.1 节。




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

B. PWM 是指设备中所有 PWM 引脚。$\overline{TZ}$取高后的 PWM 引脚状态取决于 PWM 模块寄存器的配置。

**图 5-44 PWM 高阻态特性**

## 5.9.9. 高分辨率 PWM (HRPWM)

该模块将多个延迟线组合在一个模块中，并通过使用专用校准延迟链简化了校准系统。每个 ePWM 模块都有一条 HR 延迟链。

HRPWM 模块提供的 PWM 分辨率（时间粒度）明显优于使用传统派生数字 PWM 方法所能达到的分辨率。HRPWM 模块的关键点是：

显著扩展了传统派生数字 PWM 的时间分辨率能力

此功能可用于单边沿（占空比和相移控制）以及频率/周期调制的双边沿控制。

通过扩展 ePWM 模块的比较 A 和相位寄存器来实现更精细的时间粒度控制或边沿定位。

在特定器件上，HRPWM 的功能仅在 ePWM 模块的 A 信号路径（即 EPWMxA 输出）上提供。EPWMxB 输出具有传统的 PWM 功能。

注 意

HRPWM 最小的 SYSCLKOUT 允许频率是 60 MHz。

注 意

当双边沿高分辨率开启时（高分辨率周期模式），PWMxB 通道的输出将有 ±1–2 TBCLK 周期抖动

### 5.9.9.1. HRPWM 电气数据/时序

第 5.9.9.1.1 节列出了高分辨率 PWM 的开关特性。

#### 5.9.9.1.1. 高分辨率PWM特性

| 参数(1) | 最小值 | 典型值 | 最大值 | 单位 |
|---|---|---|---|---|
| 微边沿定位 (MEP) 步长(2) | | 150 | 310 | ps |

(1) HRPWM 允许的最小 SYSCLKOUT 运行频率为 60 MHz。

(2) 最大 MEP 步长基于最差情况、最高温度。MEP 步长会随着温度升高和电压降低而增加，随着温度降低和电压升高而减小。

(3) 使用 HRPWM 功能的应用程序应使用 MEP 比例因子优化器 (SFO) 估计软件功能。SFO 功能有助于在 HRPWM 运行时动态估计每个 SYSCLKOUT 周期的 MEP 步数。

## 5.9.10. 增强型 CAP 模块（eCAP1）

该器件包含一个增强型 CAP(eCAP)模块。图 5-45 所示为该模块功能框图。

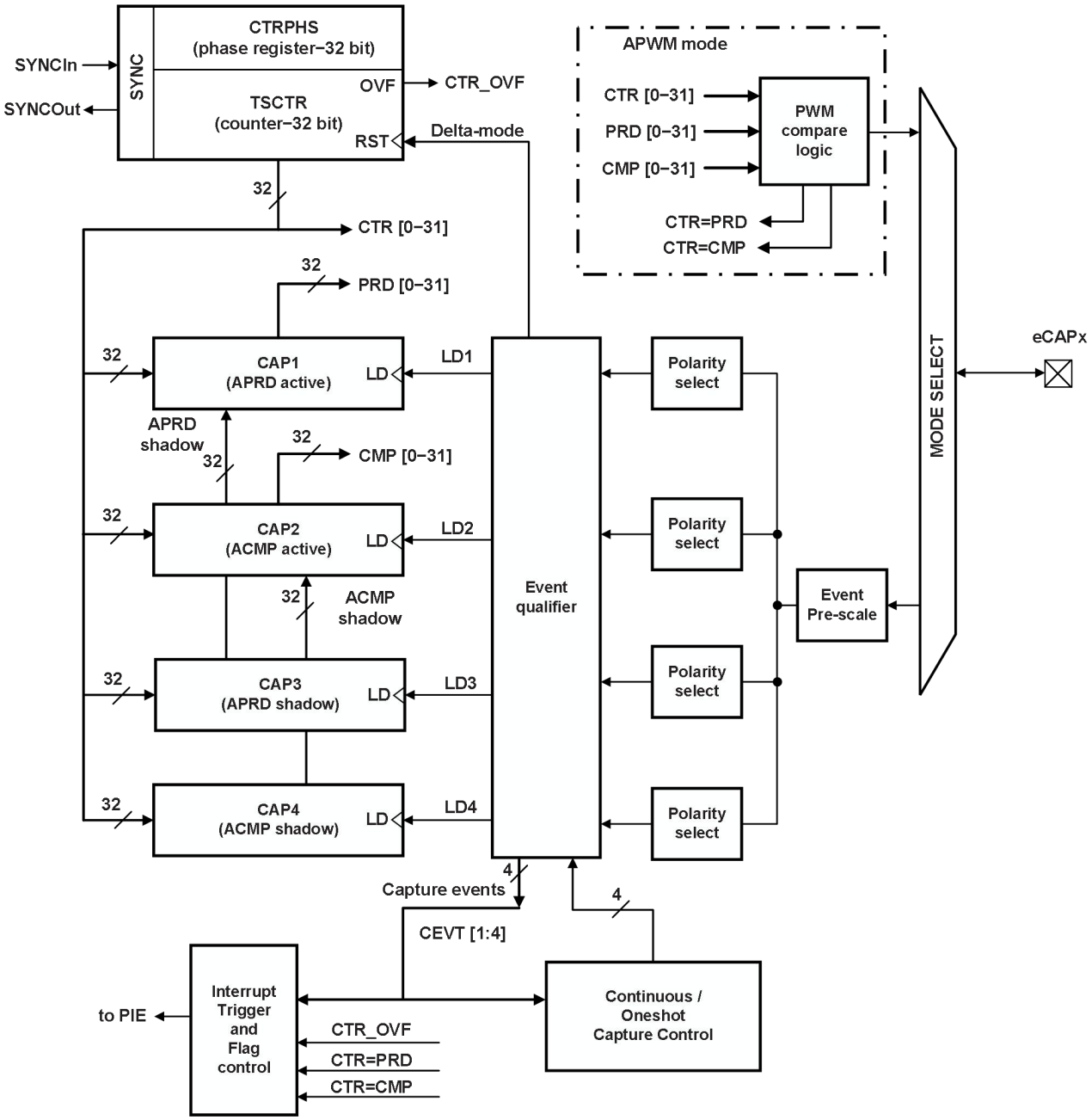



**图 5-45 eCAP 功能框图**

eCAP 模块以 SYSCLKOUT 速率计时。PCLKCR1 寄存器中的时钟使能位(ECAP1ENCLK) 用于独立关闭 eCAP 模块（为实现低功耗运行）。复位时，ECAP1ENCLK 被设为低电平，表明外设时钟关闭。

**表 5-34 eCAP 控制和状态寄存器**

| 名称 | eCAP1 | eCAP2 | eCAP3 | 大小 (×16) | 访问类型 | 描述 |
|---|---|---|---|---|---|---|
| TSCTR | 0x6A00 | 0x6A20 | 0x6A40 | 2 | 不被保护 | 时间戳计数器 |
| CTRPHS | 0x6A02 | 0x6A22 | 0x6A42 | 2 | 不被保护 | 计数器相位偏移值寄存器 |
| CAP1 | 0x6A04 | 0x6A24 | 0x6A44 | 2 | 不被保护 | 捕获1 寄存器 |
| CAP2 | 0x6A06 | 0x6A26 | 0x6A46 | 2 | 不被保护 | 捕获2 寄存器 |
| CAP3 | 0x6A08 | 0x6A28 | 0x6A48 | 2 | 不被保护 | 捕获3 寄存器 |
| CAP4 | 0x6A0A | 0x6A2A | 0x6A4A | 2 | 不被保护 | 捕获4 寄存器 |
| Reserved | 0x6A0C to 0x6A12 | 0x6A2C to 0x6A32 | 0x6A4C to 0x6A52 | 8 | 不被保护 | 保留 |
| ECCTL1 | 0x6A14 | 0x6A34 | 0x6A54 | 1 | 不被保护 | 捕获控制寄存器1 |
| ECCTL2 | 0x6A15 | 0x6A35 | 0x6A55 | 1 | 不被保护 | 捕获控制寄存器2 |
| ECEINT | 0x6A16 | 0x6A36 | 0x6A56 | 1 | 不被保护 | 捕获中断使能寄存器 |
| ECFLG | 0x6A17 | 0x6A37 | 0x6A57 | 1 | 不被保护 | 捕获中断标志寄存器 |
| ECCLR | 0x6A18 | 0x6A38 | 0x6A58 | 1 | 不被保护 | 捕获中断清除寄存器 |
| ECFRC | 0x6A19 | 0x6A39 | 0x6A59 | 1 | 不被保护 | 捕获中断强制寄存器 |
| Reserved | 0x6A1A to 0x6A1F | 0x6A3A to 0x6A3F | 0x6A5A to 0x6A5F | 6 | 不被保护 | 保留 |

## 5.9.10.1. eCAP 电气参数/时序

第 5.9.10.1.1 节说明了 eCAP 时序要求，第 5.9.10.1.2 节说明了 eCAP 开关特性。

### 5.9.10.1.1. 增强型捕获 (eCAP) 时序要求

| | | | 最小值(1) | 最大值 | 单位 |
|---|---|---|---|---|---|
| $t_{w(CAP)}$ | 捕获脉冲输入宽度 | 异步 | $2t_{c(SCO)}$ | | 周期 |
| | | 同步 | $2t_{c(SCO)}$ | | |
| | | 有输入限定器 | $1t_{c(SCO)} + t_{w(IQSW)}$ | | |

(1) 有关输入限定参数，详见第 5.9.14.1.3.1 节。

### 5.9.10.1.2. eCAP开关特性

在建议运行条件下测得（除非另有说明）

| 参数 | | 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 |

## 5.9.11. 高精度捕获模块 (HRCAP1 至 HRCAP4)

该器件有四个高精度捕获(HRCAP)模块。高精度捕获(HRCAP)模块测量外部脉冲典型精度为 300ps。

HRCAP 用途包括：

- 电容式触摸应用
- 脉冲序列周期的高分辨率周期和占空比测量
- 瞬时速度测量
- 瞬时频率测量
- 跨越隔离边界的电压测量
- 距离测量（声呐）和扫描

HRCAP 模块的特征包括：

- 在非高分辨率或高分辨率模式中的脉冲宽度捕获
- 差值（Delta）模式脉冲宽度捕获
- 每个边沿的典型高分辨率捕获分辨率约为 300 ps
- 下降沿或上升沿中断
- 在 2 级深缓冲器中连续捕获脉冲宽度的模式
- 针对精确高分辨率捕获的校准逻辑
- 上述所有资源只用于单个输入引脚
- HRCAP 校准软件库用于校准和计算微小脉冲宽度

HRCAP 模块包括一个捕获通道和一个高分辨率校准块，该模块在校准时内部连接到最后一个可用的 ePWMxA HRPWM 通道 (即，如果有八个具有 HRPWM 功能的 ePWM，则连接到 HRPWM8A)。

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 PLL2 输出频率（与 SYSCLK2 异步）或 SYSCLKOUT
- 2 级深缓冲器中捕获高分辨率脉冲宽度

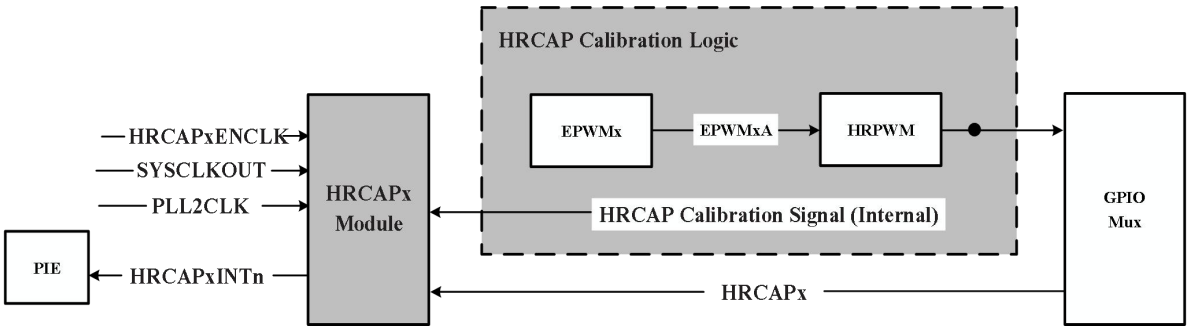



**图 5-46 HRCAP 功能框图**

**表 5-35 HRCAP 寄存器**

| 名称 | HRCAP1 | HRCAP2 | HRCAP3 | HRCAP4 | 大小 (×16) | 描述 |
|---|---|---|---|---|---|---|
| HCCTL | 0x6AC0 | 0x6AE0 | 0x6C80 | 0x6CA0 | 1 | HRCAP 控制寄存器(1) |
| HCIFR | 0x6AC1 | 0x6AE1 | 0x6C81 | 0x6CA1 | 1 | HRCAP中断标志寄存器 |
| HCICLR | 0x6AC2 | 0x6AE2 | 0x6C82 | 0x6CA2 | 1 | HRCAP中断清除寄存器(1) |
| HCIFRC | 0x6AC3 | 0x6AE3 | 0x6C83 | 0x6CA3 | 1 | HRCAP中断强制寄存器(1) |
| HCCOUNTER | 0x6AC4 | 0x6AE4 | 0x6C84 | 0x6CA4 | 1 | HRCAP 16位计数器寄存器 |
| HCCAPCNTRISE0 | 0x6AD0 | 0x6AF0 | 0x6C90 | 0x6CB0 | 1 | HRCAP 上升沿0捕获计数器寄存器 |
| HCCAPCNTFALL0 | 0x6AD2 | 0x6AF2 | 0x6C92 | 0x6CB2 | 1 | HRCAP 下降沿0捕获计数器寄存器 |
| HCCAPCNTRISE1 | 0x6AD8 | 0x6AF8 | 0x6C98 | 0x6CB8 | 1 | HRCAP 上升沿1捕获计数器寄存器 |
| HCCAPCNTFALL1 | 0x6ADA | 0x6AFA | 0x6C9A | 0x6CBA | 1 | HRCAP 下降沿1捕获计数器寄存器 |

(1) 所有寄存器都受保护。

## 5.9.11.1. HRCAP 电气参数/时序

### 5.9.11.1.1. 高精度捕获 (HRCAP) 时序要求

| | | 最小值 | 典型值 | 最大值 | 单位 |
|---|---|---|---|---|---|
| $t_{c(HCCAPCLK)}$ | 周期时间，HRCAP捕获时钟 | 8.333 | | 10.204 | ns |
| $t_{w(HRCAP)}$ | 脉冲宽度，HRCAP捕获 | $7t_{c(HCCAPCLK)}$ (1) | | | ns |
| | HRCAP 步长(2) | | 300 | | ps |

(1) 列出的最小脉冲宽度没有考虑到必须读取所有相关 HCCAP 寄存器的限制，并在脉冲宽度内清除 RISE/FALL 事件标志，以确保有效的捕获数据。

(2) HRCAP 步长会随着低压和高温而增大，随着高压和低温而减小。在高分辨率模式下使用 HRCAP 的应用，应使用 HRCAP 校准功能针对不同的工作条件进行动态校准。

## 5.9.12. 增强型正交编码器模块（eQEP1, eQEP2)

该设备最多包含两个增强型正交编码器（eQEP）模块。表 5-36 所示为 eQEP 寄存器汇总表。

**表 5-36 eQEP 控制和状态寄存器**

| 名称 | eQEP1 地址 | eQEP2 地址 | eQEP1 大小(×16)/ #SHADOW | 寄存器描述 |
|---|---|---|---|---|
| QPOSCNT | 0x6B00 | 0x6B40 | 2/0 | eQEP 位置计数器 |
| QPOSINIT | 0x6B02 | 0x6B42 | 2/0 | eQEP 初始化位置计数 |
| QPOSMAX | 0x6B04 | 0x6B44 | 2/0 | eQEP 最大位置计数 |
| QPOSCMP | 0x6B06 | 0x6B46 | 2/1 | eQEP 位置比较 |
| QPOSILAT | 0x6B08 | 0x6B48 | 2/0 | eQEP 索引位置锁存 |
| QPOSSLAT | 0x6B0A | 0x6B4A | 2/0 | eQEP 选通脉冲位置锁存 |
| QPOSLAT | 0x6B0C | 0x6B4C | 2/0 | eQEP位置锁存 |
| QUTMR | 0x6B0E | 0x6B4E | 2/0 | eQEP 单位定时器 |
| QUPRD | 0x6B10 | 0x6B50 | 2/0 | eQEP 单位周期寄存器 |
| QWDTMR | 0x6B12 | 0x6B52 | 1/0 | eQEP 看门狗定时器 |
| QWDPRD | 0x6B13 | 0x6B53 | 1/0 | eQEP 看门狗周期寄存器 |
| QDECCTL | 0x6B14 | 0x6B54 | 1/0 | eQEP 解码器控制寄存器 |
| QEPCTL | 0x6B15 | 0x6B55 | 1/0 | eQEP 控制寄存器 |
| QCAPCTL | 0x6B16 | 0x6B56 | 1/0 | eQEP 捕获控制寄存器 |
| QPOSCTL | 0x6B17 | 0x6B57 | 1/0 | eQEP 位置比较控制寄存器 |
| QEINT | 0x6B18 | 0x6B58 | 1/0 | eQEP 中断使能寄存器 |
| QFLG | 0x6B19 | 0x6B59 | 1/0 | eQEP 中断标志寄存器 |
| QCLR | 0x6B1A | 0x6B5A | 1/0 | eQEP 中断清除寄存器 |
| QFRC | 0x6B1B | 0x6B5B | 1/0 | eQEP 中断强制寄存器 |
| QEPSTS | 0x6B1C | 0x6B5C | 1/0 | eQEP 状态寄存器 |
| QCTMR | 0x6B1D | 0x6B5D | 1/0 | eQEP 捕获定时器 |
| QCPRD | 0x6B1E | 0x6B5E | 1/0 | eQEP 捕获周期寄存器 |
| QCTMRLAT | 0x6B1F | 0x6B5F | 1/0 | eQEP 捕获定时器锁存 |
| QCPRDLAT | 0x6B20 | 0x6B60 | 1/0 | eQEP 捕获周期锁存 |
| Reserved | 0x6B21 to 0x6B3F | 0x6B61 to 0x6B7F | 31/0 | |

图 5-47 所示为 eQEP 模块框图。

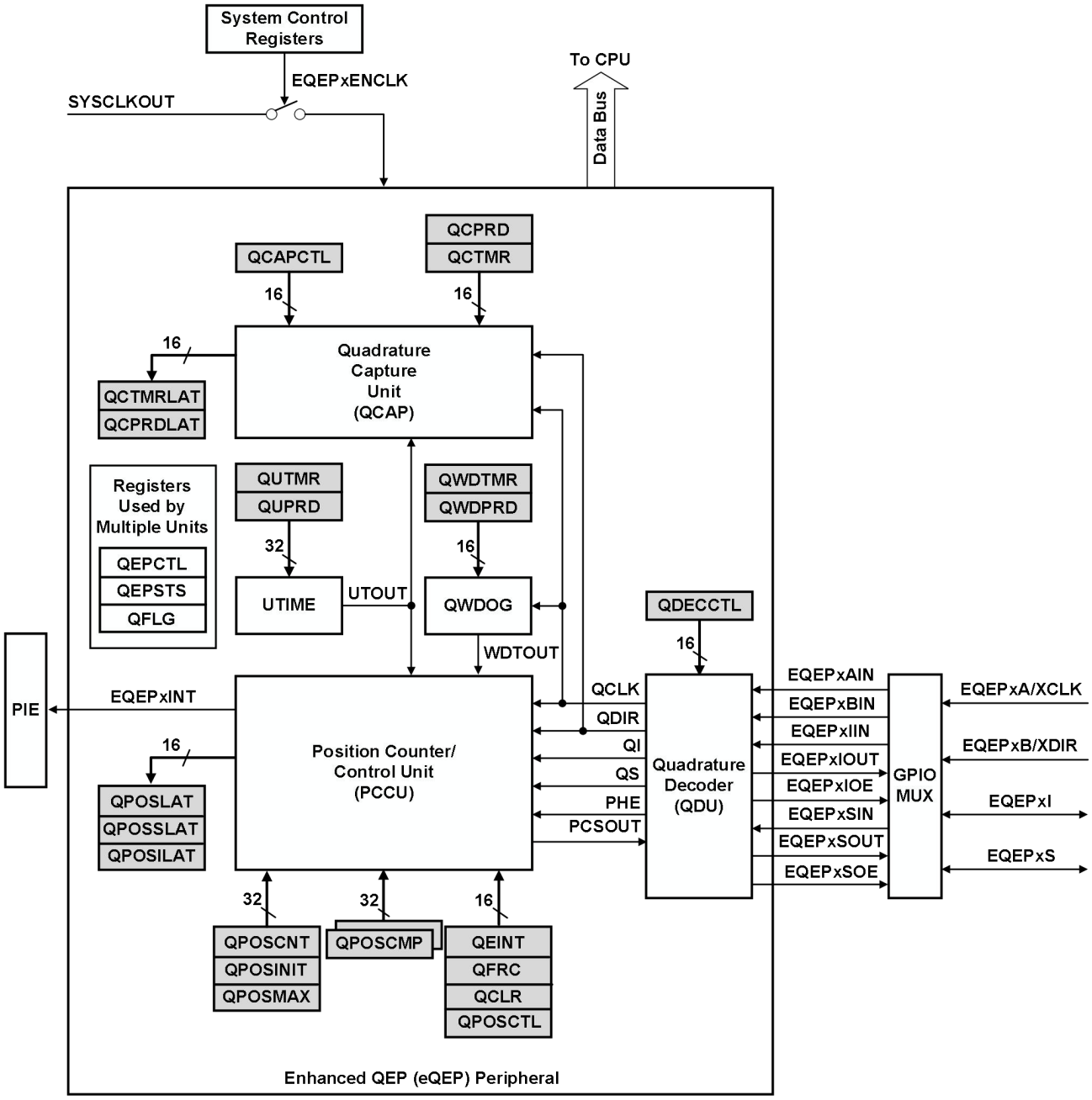



**图 5-47 eQEP 功能框图**

### 5.9.12.1. 电气数据/时序

第 5.9.12.1.1 节说明了 eQEP 的时序要求，第 5.9.12.1.2 节说明了 eQEP 的开关特性。

#### 5.9.12.1.1. 增强型正交编码脉冲（eQEP）时序要求

| | | | 最小值(1) | 最大值 | 单位 |
|---|---|---|---|---|---|
| $t_{w(QEPP)}$ | QEP 输入周期 | 异步(2)/同步 | $2t_{c(SCO)}$ | | 周期 |
| | | 带有输入限定器 | $2[1t_{c(SCO)} + t_{w(IQSW)}]$ | | |
| $t_{w(INDEXH)}$ | QEP索引输入高电平时间 | 异步(2)/同步 | $2t_{c(SCO)}$ | | 周期 |
| | | 带有输入限定器 | $2t_{c(SCO)} + t_{w(IQSW)}$ | | |
| $t_{w(INDEXL)}$ | QEP索引输入低电平时间 | 异步(2)/同步 | 2tc(SCO) | | 周期 |
| | | 带有输入限定器 | $2t_{c(SCO)} + t_{w(IQSW)}$ | | |
| $t_{w(STROBH)}$ | QEP选通脉冲高电平时间 | 异步(2)/同步 | $2t_{c(SCO)}$ | | 周期 |
| | | 带有输入限定器 | $2t_{c(SCO)} + t_{w(IQSW)}$ | | |
| $t_{w(STROBL)}$ | QEP选通脉冲输入低电平时间 | 异步(2)/同步 | $2t_{c(SCO)}$ | | 周期 |
| | | 带有输入限定器 | $2t_{c(SCO)} + t_{w(IQSW)}$ | | |

(1) 有关输入限定符参数的说明，请参阅第 5.9.14.1.3.1 节。

(2) 有关异步模式的限制，请参阅《AVP32F06X 勘误表》。

#### 5.9.12.1.2. eQEP 开关特性

在建议运行条件下测得（除非另有说明）

| 参数 | | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|
| $t_{d(CNTR)xin}$ | 延迟时间，外部时钟到计数器增量的时间 | | $4t_{c(SCO)}$ | 周期 |
| $t_{d(PCS-OUT)QEP}$ | 延迟时间，QEP 输入边沿到位置比较同步输出的时间 | | $6t_{c(SCO)}$ | 周期 |

## 5.9.13. JTAG 端口

在该器件上，JTAG 端口被减少到 5 个引脚($\overline{TRST}$、TCK、TDI、TMS、TDO)。TCK、TDI、TMS 和 TDO 引脚也是 GPIO 引脚。$\overline{TRST}$信号在图 5-48 中为引脚选择 JTAG 或者 GPIO 运行模式。在调试期间，这些引脚的 GPIO 功能并不可用。如果 GPIO38/TCK/XCLKIN 引脚被用于一个外部时钟，一个替代的内部时钟源应该被用于在调试期间为器件计时，这是因为 TCK 功能需要该引脚。

注　意

在该器件中，JTAG 引脚也可被用作 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 引脚功能的调试能力。任一连接到这些引脚的电路不应防止调试探针驱动 JTAG 引脚（或者被 JTAG 引脚驱动）进行成功的调试。

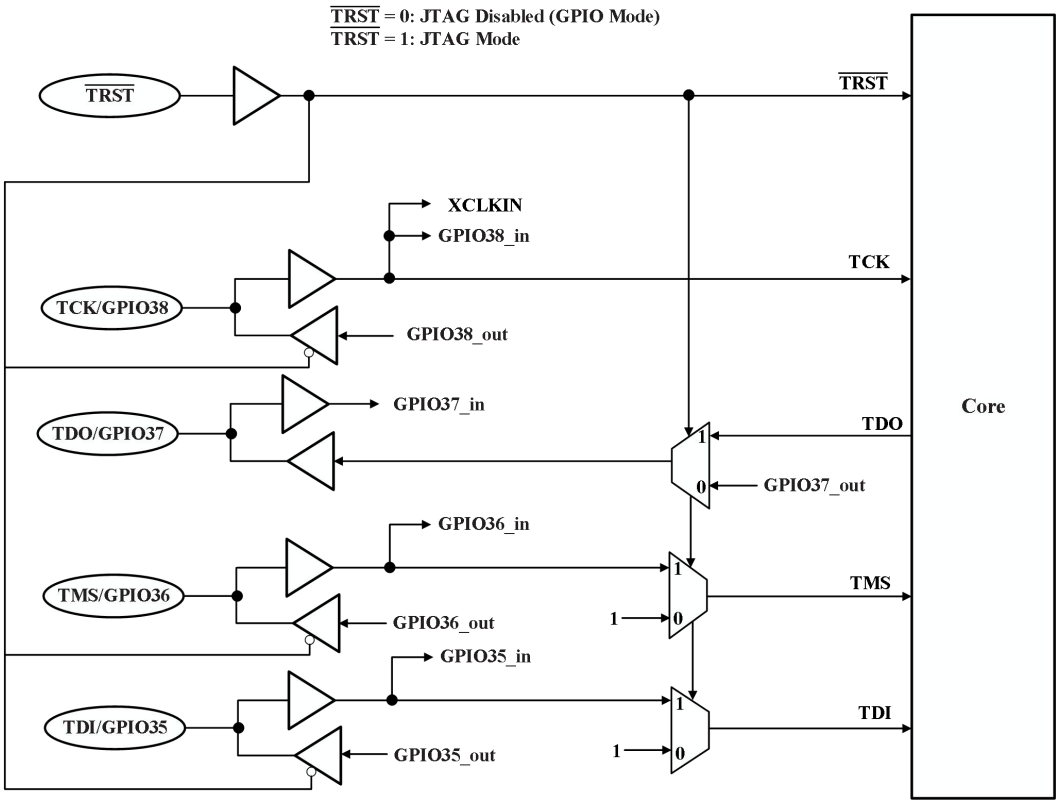



**图 5-48 JTAG/GPIO 复用**

### 5.9.14. 通用输入/输出（GPIO）复用器

GPIO 复用器除了提供独立的引脚位拆裂 I/O 功能外，还可以在单个 GPIO 引脚上多路复用多达三个独立的外设信号。

该器件支持 45 个 GPIO 引脚。GPIO 控制和数据寄存器被映射到外设帧 1 以启用对寄存器的 32 位操作（以及 16 位操作）。表 5-37 为 GPIO 寄存器映射。

**表 5-37 GPIO 寄存器**

| 名称 | 地址 | 大小（×16） | 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 上拉禁用寄存器（GPIO0 至 31） |
| 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） |
| GPBQSEL2 | 0x6F94 | 2 | GPIO B 限定器选择 2 寄存器 |
| GPBMUX1 | 0x6F96 | 2 | GPIO B MUX 1 寄存器(GPIO32 至 44) |
| GPBMUX2 | 0x6F98 | 2 | GPIO B MUX 2 寄存器(GPIO50 至 58) |
| GPBDIR | 0x6F9A | 2 | GPIO B 方向寄存器（GPIO32 至 44） |
| GPBPUD | 0x6F9C | 2 | GPIO B 上拉禁用寄存器（GPIO32 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 数据切换寄存器（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 数据切换寄存器（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 数据切换寄存器（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 31） |
| GPIOXINT3SEL | 0x6FE2 | 1 | XINT3 GPIO 输入选择寄存器（GPIO0 至 31） |
| GPIOLPMSEL | 0x6FE8 | 2 | LPM GPIO 选择寄存器（GPIO0 至 31） |

**注　意**

从写入 GPxMUXn/AIOMUXn 和 GPxQSELn 寄存器发生到动作有效有两个 SYSCLKOUT 周期延迟。

**表 5-38 GPIOA MUX**

| | 复位时默认主I/O功能(1)(2) | 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) | EPWMSYNCI (I) | EPWMSYNCO (O) |
| 15-14 | GPIO7 | EPWM4B (O) | SCIRXDA (I) | ECAP2 (I/O) |
| 17-16 | GPIO8 | EPWM5A (O) | 保留 | $\overline{\text{ADCSOCAO}}$ (O) |
| 19-18 | GPIO9 | EPWM5B (O) | SCITXDB (O) | ECAP3 (I/O) |
| 21-20 | GPIO10 | EPWM6A (O) | 保留 | $\overline{\text{ADCSOCBO}}$ (O) |
| 23-22 | GPIO11 | EPWM6B (O) | SCIRXDB (I) | ECAP1 (I/O) |
| 25-24 | GPIO12 | $\overline{\text{TZ1}}$ (I) | SCITXDA (O) | SPISIMOB (I/O) |
| 27-26 | GPIO13 | $\overline{\text{TZ2}}$ (I) | 保留 | SPISOMIB (I/O) |
| 29-28 | GPIO14 | $\overline{\text{TZ3}}$ (I) | SCITXDB (O) | SPICLKB (I/O) |
| 31-30 | GPIO15 | ECAP2 (I/O) | SCIRXDB (I) | $\overline{\text{SPISTEB}}$ (I/O) |
| **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 | $\overline{\text{TZ2}}$ (I) |
| 3-2 | GPIO17 | SPISOMIA (I/O) | 保留 | $\overline{\text{TZ3}}$ (I) |
| 5-4 | GPIO18 | SPICLKA (I/O) | SCITXDB (O) | XCLKOUT (O) |
| 7-6 | GPIO19/XCLKIN | $\overline{\text{SPISTEA}}$ (I/O) | SCIRXDB (I) | ECAP1 (I/O) |
| 9-8 | GPIO20 | EQEP1A (I) | MDXA (O) | COMP1OUT (O) |
| 11-10 | GPIO21 | EQEP1B (I) | MDRA (I) | COMP2OUT (O) |
| 13-12 | GPIO22 | EQEP1S (I/O) | MCLKXA (I/O) | SCITXDB (O) |
| 15-14 | GPIO23 | EQEP1I (I/O) | MFSXA (I/O) | SCIRXDB (I) |
| 17-16 | GPIO24 | ECAP1 (I/O) | EQEP2A(3) (I) | SPISIMOB (I/O) |
| 19-18 | GPIO25 | ECAP2 (I/O) | EQEP2B(3) (I) | SPISOMIB (I/O) |
| 21-20 | GPIO26(4) | ECAP3 (I/O) | EQEP2I(3) (I/O) | SPICLKB (I/O) |
| 23-22 | GPIO27(4) | HRCAP2 (I) | EQEP2S(3) (I/O) | $\overline{\text{SPISTEB}}$(I/O) |
| 25-24 | GPIO28 | SCIRXDA (I) | SDAA (I/OD) | $\overline{\text{TZ2}}$ (I) |
| 27-26 | GPIO29 | SCITXDA (O) | SCLA (I/OD) | $\overline{\text{TZ3}}$ (I) |
| 29-28 | GPIO30 | CANRXA (I) | EQEP2I(3) (I/O) | EPWM7A (O) |
| 31-30 | GPIO31 | CANTXA (O) | EQEP2S(3) (I/O) | EPWM8A (O) |

(1) “保留”意味着没有指定外设到该 GPxMUX1/2 寄存器设置。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。该选择是为未来扩展预留的保留配置。

(2) I = 输入， O = 输出，OD = 开漏

(3) 在 80 引脚 PN/PFP 封装内，eQEP2 外设不可用。

(4) 要使能 GPIO26（USB0DP，USB 信号的正差分一半）和 GPIO27（USB0DM，USB 信号的负差分一半）上的 USB 功能，设置 GPACTRL2 寄存器中的 USBIOEN 位。根据用户的 USB 应用，可能需要额外的引脚来保持符合 USB 2.0 规范。

表 5-39 GPIOB MUX

| | 复位时默认主 I/O 功能[1][2] | 外设选择 1 | 外设选择 2 | 外设选择 3 |
|---|---|---|---|---|
| **GPBMUX1 寄存器位** | **(GPBMUX1 位 = 00)** | **(GPBMUX1 位 = 01)** | **(GPBMUX1 位 = 10)** | **(GPBMUX1 位 = 11)** |
| 1-0 | GPIO32 | SDAA (I/OD) | EPWMSYNCI (I) | $\overline{\text{ADCSOCAO}}$ (O) |
| 3-2 | GPIO33 | SCLA (I/OD) | EPWMSYNCO (O) | $\overline{\text{ADCSOCBO}}$ (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 | GPIO39 | 保留 | 保留 | 保留 |
| 17-16 | GPIO40[3] | EPWM7A (O) | SCITXDB (O) | 保留 |
| 19-18 | GPIO41[3] | EPWM7B (O) | SCIRXDB (I) | 保留 |
| 21-20 | GPIO42[3] | EPWM8A (O) | $\overline{\text{TZ1}}$ (I) | COMP1OUT (O) |
| 23-22 | GPIO43[3] | EPWM8B (O) | $\overline{\text{TZ2}}$ (I) | COMP2OUT (O) |
| 25-24 | GPIO44[3] | MFSRA (I/O) | SCIRXDB (I) | EPWM7B (O) |
| 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 |
| **GPBMUX2 寄存器位** | **(GPBMUX2 位 = 00)** | **(GPBMUX2 位 = 01)** | **(GPBMUX2 位 = 10)** | **(GPBMUX2 位 = 11)** |
| 1-0 | 保留 | 保留 | 保留 | 保留 |
| 3-2 | 保留 | 保留 | 保留 | 保留 |
| 5-4 | GPIO50[3] | EQEP1A (I) | MDXA (O) | $\overline{\text{TZ1}}$ (I) |
| 7-6 | GPIO51[3] | EQEP1B (I) | MDRA (I) | $\overline{\text{TZ2}}$ (I) |
| 9-8 | GPIO52[3] | EQEP1S (I/O) | MCLKXA (I/O) | $\overline{\text{TZ3}}$ (I) |
| 11-10 | GPIO53[3] | EQEP1I (I/O) | MFSXA (I/O) | 保留 |
| 13-12 | GPIO54[3] | SPISIMOA (I/O) | EQEP2A (I) | HRCAP1 (I) |
| 15-14 | GPIO55[3] | SPISOMIA (I/O) | EQEP2B (I) | HRCAP2 (I) |
| 17-16 | GPIO56[3] | SPICLKA (I/O) | EQEP2I (I/O) | HRCAP3 (I) |
| 19-18 | GPIO57[3] | $\overline{\text{SPISTEA}}$ (I/O) | EQEP2S (I/O) | HRCAP4 (I) |
| 21-20 | GPIO58[3] | MCLKRA (I/O) | SCITXDB (O) | EPWM7A (O) |
| 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) “保留”意味着没有外设被指定到这个 GPxMUX1/2 寄存器设置。如果它被选择，那么引脚的状态将为未定义并且此引脚可被驱动。这个选择是为以后扩展预留的保留配置。

(2) I = 输入，O = 输出，OD = 开漏

(3) 在 80 引脚 PN/PFP 封装内，该引脚不可用。

**表 5-40 用于 100 引脚 LQFP 封装的模拟 MUX**

| | | 复位时默认 |
|---|---|---|
| | AIOx 和外设选择 1 | 外设选择 2 和外设选择 3 |
| AIOMUX1 寄存器位 | AIOMUX1 位 = 0,x | AIOMUX1 位 = 1,x |
| 1-0 | ADCINA0 (I) | ADCINA0 (I) |
| 3-2 | ADCINA1 (I) | ADCINA1 (I) |
| 5-4 | AIO2 (I/O) | ADCINA2 (I), COMP1A (I) |
| 7-6 | ADCINA3 (I) | ADCINA3 (I) |
| 9-8 | AIO4 (I/O) | ADCINA4 (I), COMP2A (I) |
| 11-10 | ADCINA5 (I) | ADCINA5 (I) |
| 13-12 | AIO6 (I/O) | ADCINA6 (I), COMP3A (I) |
| 15-14 | ADCINA7 (I) | ADCINA7 (I) |
| 17-16 | ADCINB0 (I) | ADCINB0 (I) |
| 19-18 | ADCINB1 (I) | ADCINB1 (I) |
| 21-20 | AIO10 (I/O) | ADCINB2 (I), COMP1B (I) |
| 23-22 | ADCINB3 (I) | ADCINB3 (I) |
| 25-24 | AIO12 (I/O) | ADCINB4 (I), COMP2B (I) |
| 27-26 | ADCINB5 (I) | ADCINB5 (I) |
| 29-28 | AIO14 (I/O) | ADCINB6 (I), COMP3B (I) |
| 31-30 | ADCINB7 (I) | ADCINB7 (I) |

**表 5-41 用于 80 引脚 LQFP 封装的模拟 MUX**

| | | 复位时默认(1) |
|---|---|---|
| | AIOx 和外设选择 1 | 外设选择 2 和外设选择 3 |
| AIOMUX1 寄存器位 | AIOMUX1 位 = 0,x | AIOMUX1 位 = 1,x |
| 1-0 | ADCINA0 (I), $V_{REFHI}$ (I) | ADCINA0 (I), $V_{REFHI}$ (I) |
| 3-2 | ADCINA1 (I) | ADCINA1 (I) |
| 5-4 | AIO2 (I/O) | ADCINA2 (I), COMP1A (I) |
| 7-6 | – | – |
| 9-8 | AIO4 (I/O) | ADCINA4 (I), COMP2A (I) |
| 11-10 | ADCINA5 (I) | ADCINA5 (I) |
| 13-12 | AIO6 (I/O) | ADCINA6 (I), COMP3A (I) |
| 15-14 | – | – |
| 17-16 | ADCINB0 (I) | ADCINB0 (I) |
| 19-18 | ADCINB1 (I) | ADCINB1 (I) |
| 21-20 | AIO10 (I/O) | ADCINB2 (I), COMP1B (I) |
| 23-22 | – | – |
| 25-24 | AIO12 (I/O) | ADCINB4 (I), COMP2B (I) |
| 27-26 | ADCINB5 (I) | ADCINB5 (I) |
| 29-28 | AIO14 (I/O) | ADCINB6 (I), COMP3B (I) |
| 31-30 | – | – |

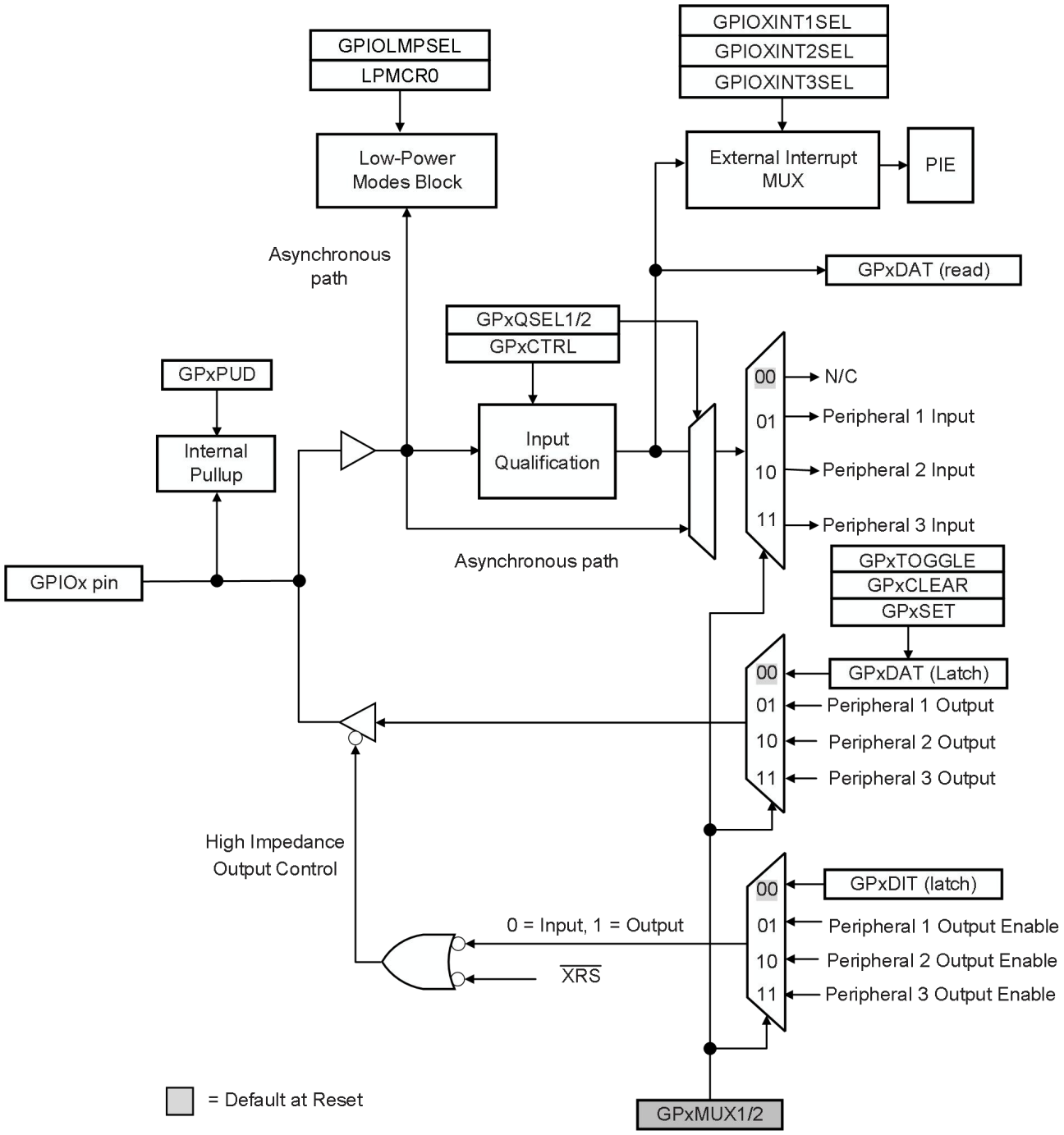
(1) I = 输入， O = 输出

用户可以通过 GPxQSEL1/2 寄存器从四个选项中选择每个 GPIO 引脚的输入限定类型：

1. 仅同步到 SYSCLKOUT (GPxQSEL1/2 = 0, 0)：这是所有 GPIO 引脚在复位时的默认模式，它只

是将输入信号同步到系统时钟 (SYSCLKOUT)。

2. 使用采样窗口的限定条件 (GPxQSEL1/2 = 0, 1 和 1, 0)：在此模式下，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，然后才允许输入改变。
3. 采样周期由 GPxCTRL 寄存器中的 QUALPRD 位指定， 对 8 个信号一组进行配置。采样周期指定用于采样输入信号的 SYSCLKOUT 周期的倍数。采样窗口是 3 次采样或 6 次采样宽，只有当所有采样都相同（全 0 或全 1）时，输出才会改变。
4. 无同步 (GPxQSEL1/2 = 1,1)：该模式用于不需要同步的外设（在外设内执行同步）。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。




(1) 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

(2) GPxDAT 锁存/读取在相同存储器位置访问。

(3) 这是一个常用的 GPIO MUX 方框图。并非所有选项都可用于所有 GPIO 引脚。详细信息请参考《AVP32F06X 技术参考手册》的“系统控制和中断”一章。

**图 5-49 GPIO 复用(1)(2)(3)**

### 5.9.14.1. GPIO 电气数据/时序

#### 5.9.14.1.1. GPIO 输出时序

##### 5.9.14.1.1.1. 通用输出开关特性

在建议运行条件下测得（除非额外注明）

| 参数 | | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|
| $t_{r(GPO)}$ 上升时间，GPIO 从低电平切换至高电平的时间 | 所有 GPIO | | 13(1) | ns |
| $t_{f(GPO)}$ 下降时间，GPIO 从高电平切换至低电平的时间 | 所有 GPIO | | 13(1) | ns |
| $F_{GPO}$ 切换频率 | | | 22.5 | MHz |

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

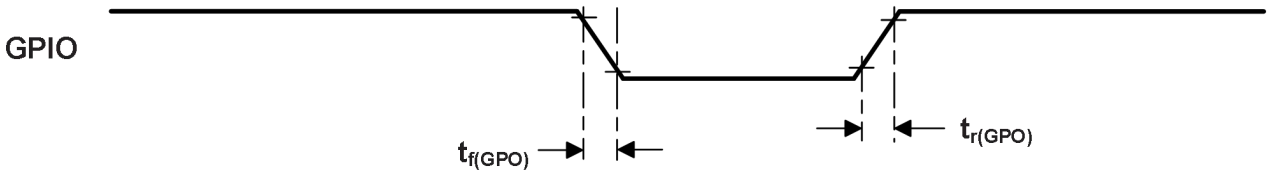



**图 5-50 通用输出时序**

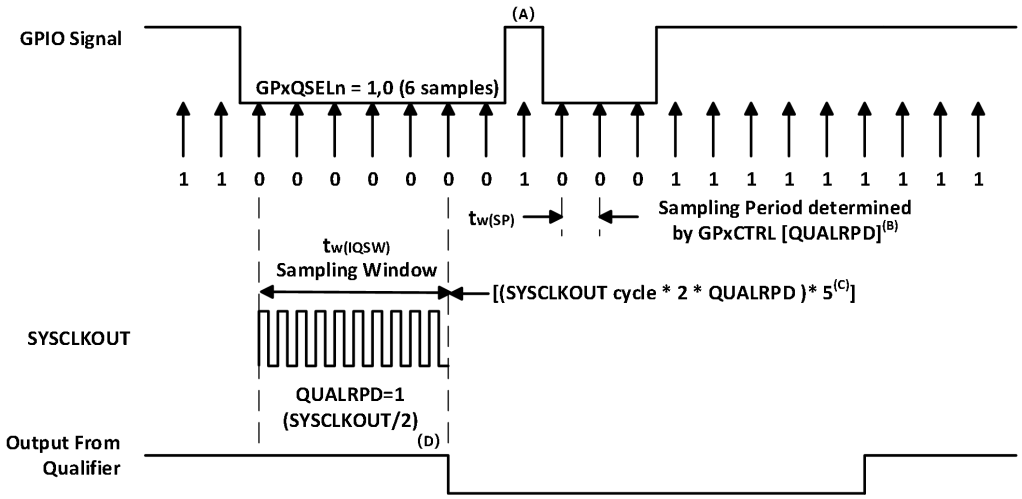
#### 5.9.14.1.2. GPIO 输入时序

##### 5.9.14.1.2.1. 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 | | |
| $t_{w(IQSW)}$ | 输入限定器采样窗口 | | $t_{w(SP)}$ * (n(1) –1) | | 周期 |
| $t_{w(GPI)}$(2) | 脉冲持续时间，GPIO低电平/高电平 | 同步模式 | $2t_{c(SCO)}$ | | 周期 |
| | | 带有输入限定器 | $t_{w(IQSW)} + t_{w(SP)} + 1t_{c(SCO)}$ | | |

(1) "n" 代表由 GPxQSELn 寄存器定义的限定采样的数量。

(2) 对于 $t_{w(GPI)}$，对于一个低电平有效信号脉宽为 $V_{IL}$ 至 $V_{IL}$ ，对于一个高电平有效信号脉宽为 $V_{IH}$ 至 $V_{IH}$。




A. 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，那么采样周期为 1 个 SYSCLKOUT 周期。对于任何其它的"n"值，限定采样周期为 2n 个 SYSCLKOUT 周期（即每 2n 个 SYSCLKOUT 周期对 GPIO 引脚进行采样）。

B. 通过 GPxCTRL 寄存器选择的限定周期应用于一组 8 个 GPIO 引脚上。

C. 此限定块可取 3 个或 6 个采样。GPxQSELn 寄存器选择使用的采样模式。

D. 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。换句话说，输入应该在 (5 x QUALPRD x 2) SYSCLKOUT 周期内保持稳定。这将确保发生 5 个用于检测的采样周期。由于外部时钟被异步驱动，一个 13 SYSCLKOUT 宽的脉冲将确保可靠识别。

**图 5-51 采样模式**

#### 5.9.14.1.3. 输入信号的采样窗口宽度

以下部分总结了不同的输入限定器配置下，用于输入信号的采样窗口宽度。采样频率表示相对于 SYSCLKOUT 的信号采样频率。

如果 QUALPRD ≠0，采样频率 = SYSCLKOUT/(2 × QUALPRD)

如果 QUALPRD = 0，采样频率 = SYSCLKOUT

如果 QUALPRD ≠0，采样频率 = SYSCLKOUT 周期 × 2 × QUALPRD

在上面的等式中，SYSCLKOUT 周期表示 SYSCLKOUT 的时间周期 。

如果 QUALPRD = 0，采样频率 = SYSCLKOUT 周期

在一个指定的采样窗口中，对输入信号进行 3 次采样或 6 次采样，以确定信号的有效性，这由写入到 GPxQSELn 寄存器的值确定。

**例 1:**

使用 3 个采样的限定

如果 QUALPRD ≠ 0，采样窗口宽度 = (SYSCLKOUT 周期 × 2 × QUALPRD) × 2

如果 QUALPRD = 0，采样窗口宽度 = (SYSCLKOUT 周期 ) × 2

**例 2:**

使用 6 个采样的限定

如果 QUALPRD ≠ 0，采样窗口宽度 = (SYSCLKOUT 周期 × 2 × QUALPRD) × 5

如果 QUALPRD = 0，采样窗口宽度 = (SYSCLKOUT 周期 ) × 5

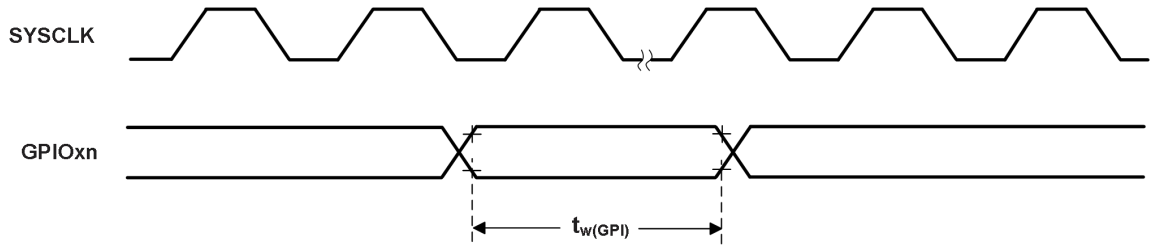



**图 5-52 通用输入时序**

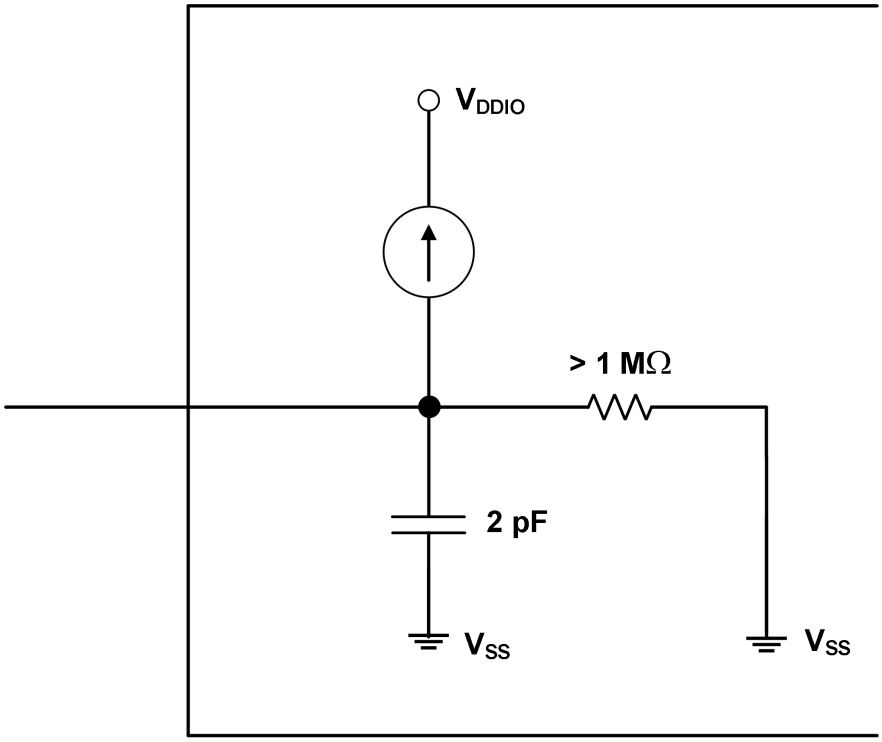



**图 5-53 带有内部上拉的 GPIO 引脚的输入电阻模型**

#### 5.9.14.1.4. 低功耗唤醒时序

第 5.9.14.1.3.1 节显示了时序要求，第 5.9.14.1.3.3 节显示了开关特性，而图 5-54 所示为 IDLE 模式下的时序图。

##### 5.9.14.1.4.1. IDLE 模式时序要求

| | | 最小值(1) | 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 |
| | 带有输入限定器 | $5t_{c(SCO)} + t_{w(IQSW)}$ | | |

(1) 对于输入限定器参数的说明，请见第 5.9.14.1.3.1 节。

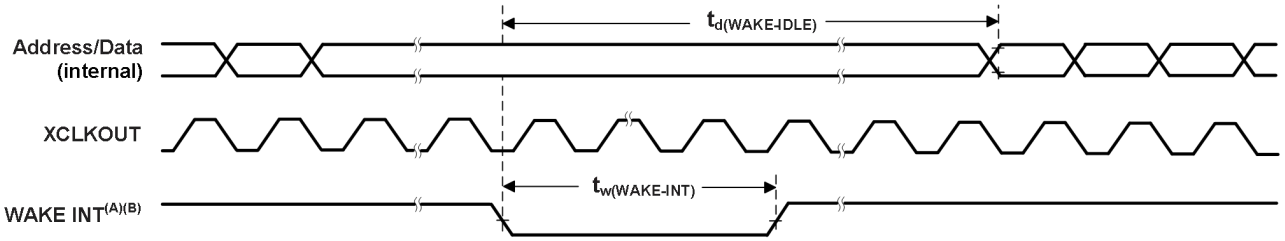
##### 5.9.14.1.4.2. IDLE 模式开关特性

在建议运行条件下测得（除非额外注明）(1)

| 参数 | | 测试条件 | 最小值 | 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 |
| | 从闪存唤醒<br>– 激活状态中的闪存模块 | 无输入限定器 | | $20t_{c(SCO)}$ | 周期 |
| | | 带有输入限定器 | | $20t_{c(SCO)} + t_{w(IQSW)}$ | |
| | 从闪存唤醒<br>– 休眠状态中的闪存模块 | 无输入限定器 | | $1050t_{c(SCO)}$ | 周期 |
| | | 带有输入限定器 | | $1050t_{c(SCO)} + t_{w(IQSW)}$ | |
| | 从SARAM 中唤醒 | 无输入限定器 | | $20t_{c(SCO)}$ | 周期 |
| | | 带有输入限定器 | | $20t_{c(SCO)} + t_{w(IQSW)}$ | |

(1) 对于输入限定器参数的说明，请见第 5.9.14.1.3.1 节。

（2）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. WAKE INT 可以是任一启用的中断，WDINT 或 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. 从将器件置于低功耗模式 (LPM) 的 IDLE 指令被执行开始，在至少 4 个 OSCCLK 周期之前，不应启动唤醒。

**图 5-54 进入和退出 IDLE 模式的时序**

### 5.9.14.1.4.3. STANDBY 模式时序要求

| | | | 最小值 | 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 | $3t_{c(OSCCLK)}$ | | 周期 |
| | | 带有输入限定器(1) | (2 + QUALSTDBY) * $t_{c(OSCCLK)}$ | | |

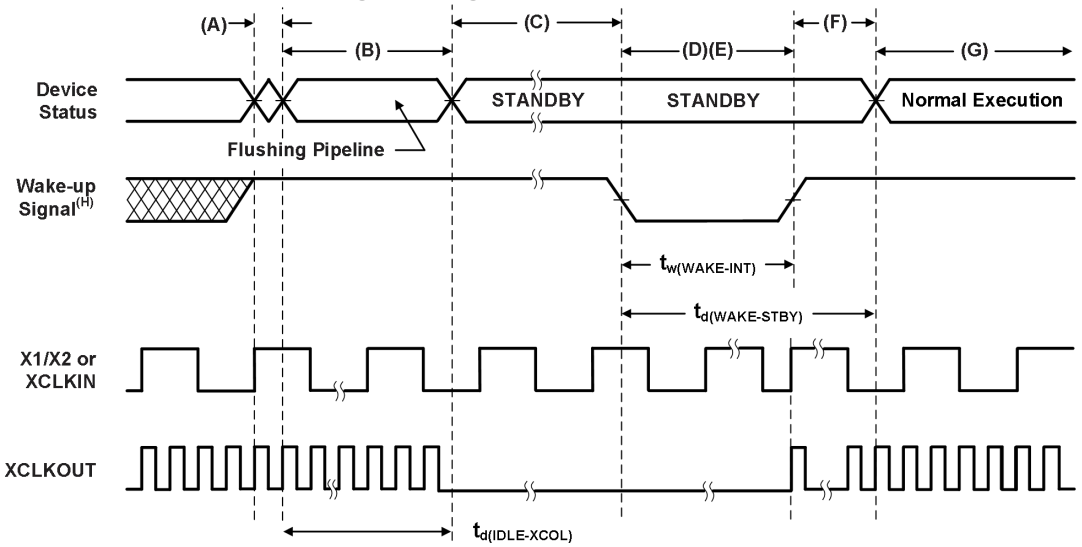
(1) QUALSTDBY 是一个 LPMCR0 寄存器内的 6 位字段。

### 5.9.14.1.4.4. STANDBY 模式开关特性

在建议运行条件下测得（除非额外注明）

| 参数 | | 测试条件 | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|---|
| $t_{d(IDLE-XCOL)}$ | 延迟时间，IDLE指令执行至 XCLKOUT 为低电平 | | $32t_{c(SCO)}$ | $45t_{c(SCO)}$ | 周期 |
| $t_{d(WAKE-STBY)}$ | 延迟时间，外部唤醒信号到程序恢复执行(1) | | | | 周期 |
| | 从闪存唤醒 – 激活状态中的闪存模块 | 无输入限定器 | | $100t_{c(SCO)}$ | 周期 |
| | | 带有输入限定器 | | $100t_{c(SCO)} + t_{w(WAKE-INT)}$ | |
| | 从闪存唤醒 – 睡眠状态中的闪存模块 | 无输入限定器 | | $1125t_{c(SCO)}$ | 周期 |
| | | 带有输入限定器 | | $1125t_{c(SCO)} + t_{w(WAKE-INT)}$ | |
| | 从SARAM 中唤醒 | 无输入限定器 | | $100t_{c(SCO)}$ | 周期 |
| | | 带有输入限定器 | | $100t_{c(SCO)} + t_{w(WAKE-INT)}$ | |

(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. 器件执行的 IDLE 指令进入 STANDBY 模式。

B. PLL 块响应 STANDBY 信号。SYSCLKOUT 在关闭之前保持如下所示的周期数：
- 当 DIVSEL = 00 或 01 时，16 个周期
- 当 DIVSEL = 10 时，32 个周期
- 当 DIVSEL = 11 时，64 个周期

这个延迟使得 CPU 流水线和其它等待的操作被适当清空。

C. 到外设的时钟被关闭。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 引脚的用于唤醒器件的唤醒信号必须符合最小脉冲宽度的要求。此外，该信号一定不能有毛刺脉冲。如果一个噪声信号被提供给 GPIO 引脚，器件的唤醒状态将是不确定的并且此器件也许不能在随后的唤醒脉冲中退出低功耗模式。

F. 在一个延迟周期内，退出 STANDBY 模式。

G. 正常执行重新开始。此器件将响应中断（如果被启用的话）。

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

**图 5-55 进入和退出 STANDBY 模式的时序图**

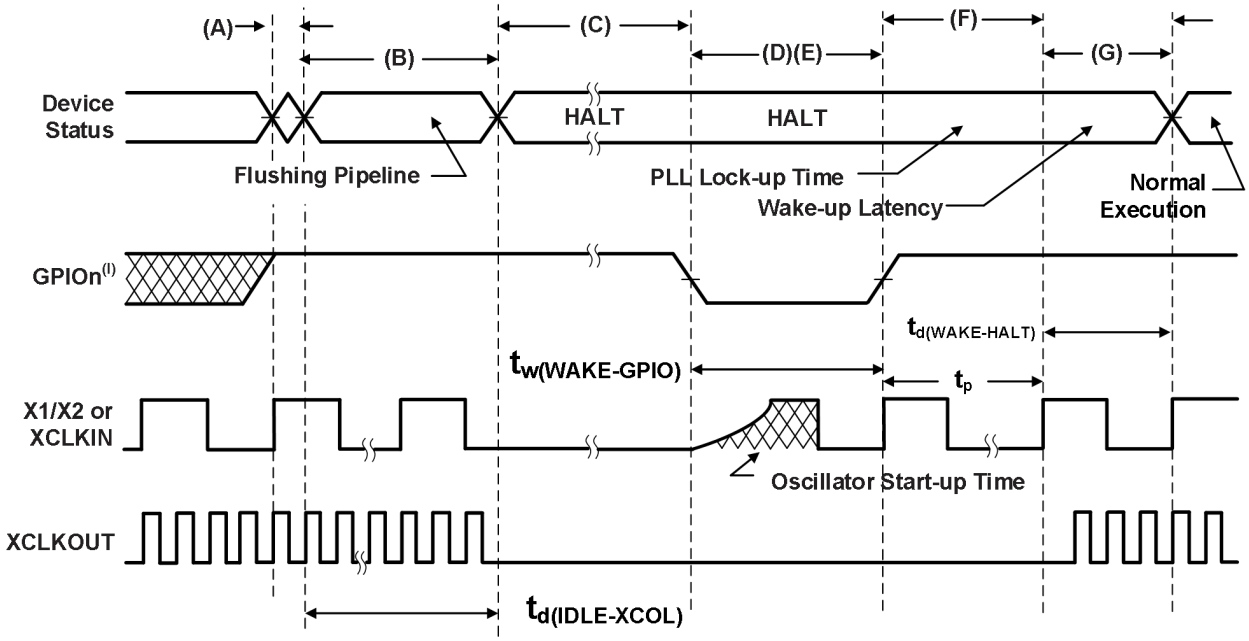
### 5.9.14.1.4.5. HALT 模式时序要求

| | 最小值 | 最大值 | 单位 |
|---|---|---|---|
| $t_{w(WAKE-GPIO)}$ 脉冲持续时间，GPIO唤醒信号的时间 | $t_{oscst} + 2t_{c(OSCCLK)}$ | | 周期 |
| $t_{w(WAKE-XRS)}$ 脉冲持续时间，$\overline{XRS}$唤醒信号的时间 | $t_{oscst} + 8t_{c(OSCCLK)}$ | | 周期 |

### 5.9.14.1.4.6. HALT 模式开关特性

在建议运行条件下测得（除非额外注明）

| 参数 | | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|
| $t_{d(IDLE-XCOL)}$ | 延迟时间，IDLE 指令被执行至 XCLKOUT 为低电平的时间 | $32t_{c(SCO)}$ | $45t_{c(SCO)}$ | 周期 |
| $t_p$ | PLL 锁存时间 | | 1 | ms |
| $t_{d(WAKE-HALT)}$ | 延迟时间，PLL 锁存到程序执行重新开始的时间<br>从闪存唤醒<br>– 睡眠状态中的闪存模块 | | $1125t_{c(SCO)}$ | 周期 |
| | 从SARAM 中唤醒 | | $35t_{c(SCO)}$ | 周期 |




A.IDLE 指令被执行以将器件置于 HALT 模式。

B.PLL 块响应 HALT 信号。 在振荡器被关闭并且到内核的 CLKIN 被停止前 SYSCLKOUT 会保持如下所示的周期数:

- 当 DIVSEL = 00 或 01 时，16 个周期
- 当 DIVSEL = 10 时，32 个周期
- 当 DIVSEL = 11 时，64 个周期

这个延迟使得 CPU 流水线和其它等待的操作被适当清空。

C. 到外设的时钟被关闭并且 PLL 被关断。如果一个石英晶振或者陶瓷谐振器被用作时钟源，内部振荡器也被关断。器件现在处于 HALT 模式，消耗绝对最小功率。可在 HALT 模式中保持零引脚内部振荡器（INTOSC1 和 INTOSC2）以及安全装置可用。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 GPIOn 引脚（用于将器件带出 HALT）被驱动为低电平时，振荡器打开且振荡器唤醒序列启动。只有当振荡器稳定时，GPIO 才应被驱动为高电平。这样可在 PLL 锁定序列期间提供干净的时钟信号。由于 GPIO 引脚的下降边沿异步启动唤醒序列，因此在进入 HALT 模式之前和期间应注意保持低噪声环境。

E. 提供给 GPIO 引脚的用于唤醒器件的唤醒信号必须符合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 锁定序列启动需耗时 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 5-56 使用 GPIOn 唤醒 HALT 模式**

# 5.9.15. 通用串行总线（USB）

## 5.9.15.1. USB 电器数据/时序

### 5.9.15.1.1. USB输入端口 DP和DM时序要求

| 参数 | | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|
| V(CM) | 差分输入共模范围 | 0.8 | 2.5 | V |
| Z(IN) | 输入阻抗 | 300 | | kΩ |
| VCRS | 交叉电压 | 1.3 | 2.0 | V |
| $V_{IL}$ | 静态SE输入逻辑低电平 | 0.8 | | V |
| $V_{IH}$ | 静态SE输入逻辑高电平 | | 2.0 | V |
| VDI | 差分输入电压 | | 0.2 | V |

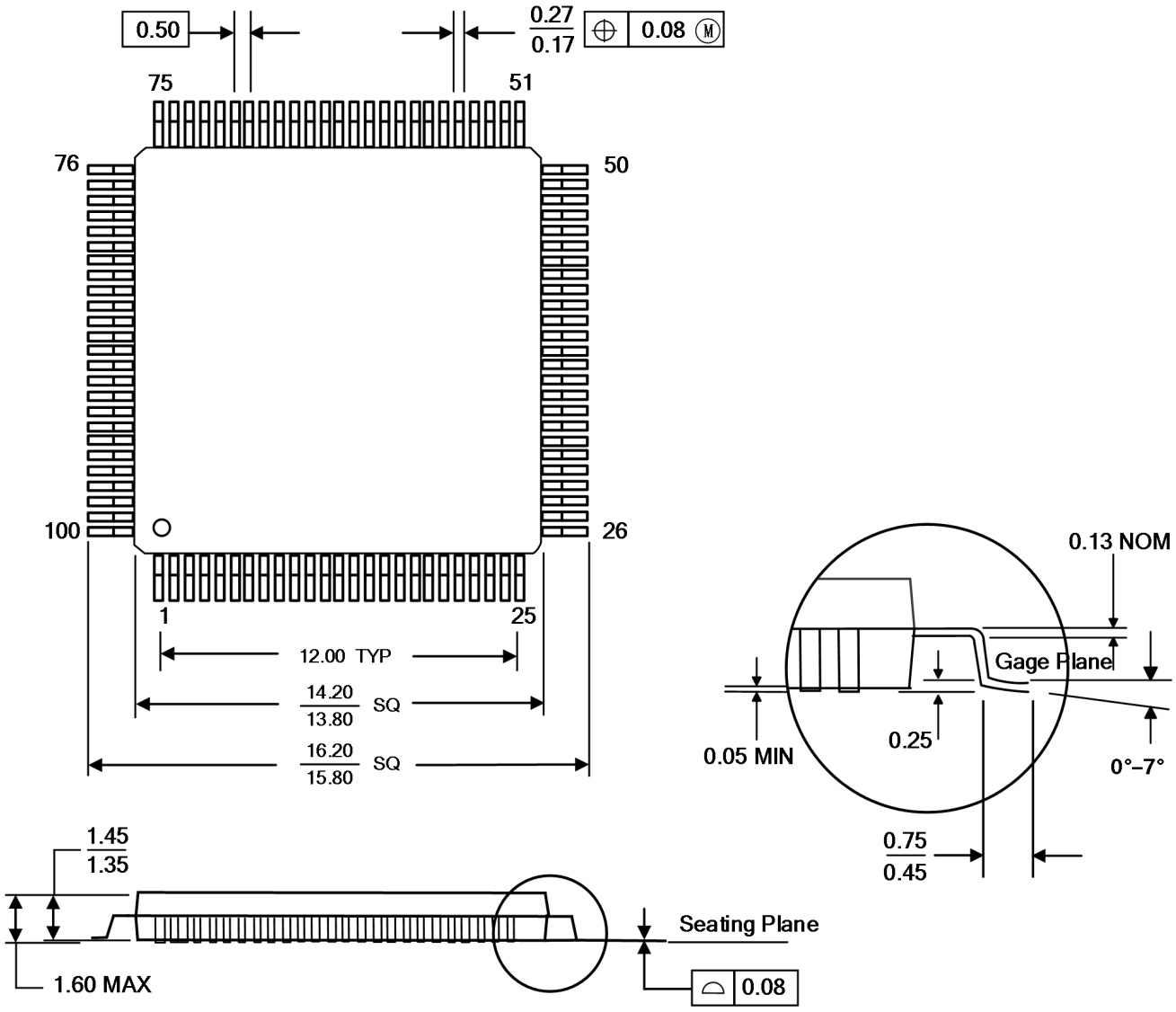
### 5.9.15.1.2. USB 输出端口 DP 和 DM 开关特性

在建议运行条件下测得（除非另行说明）

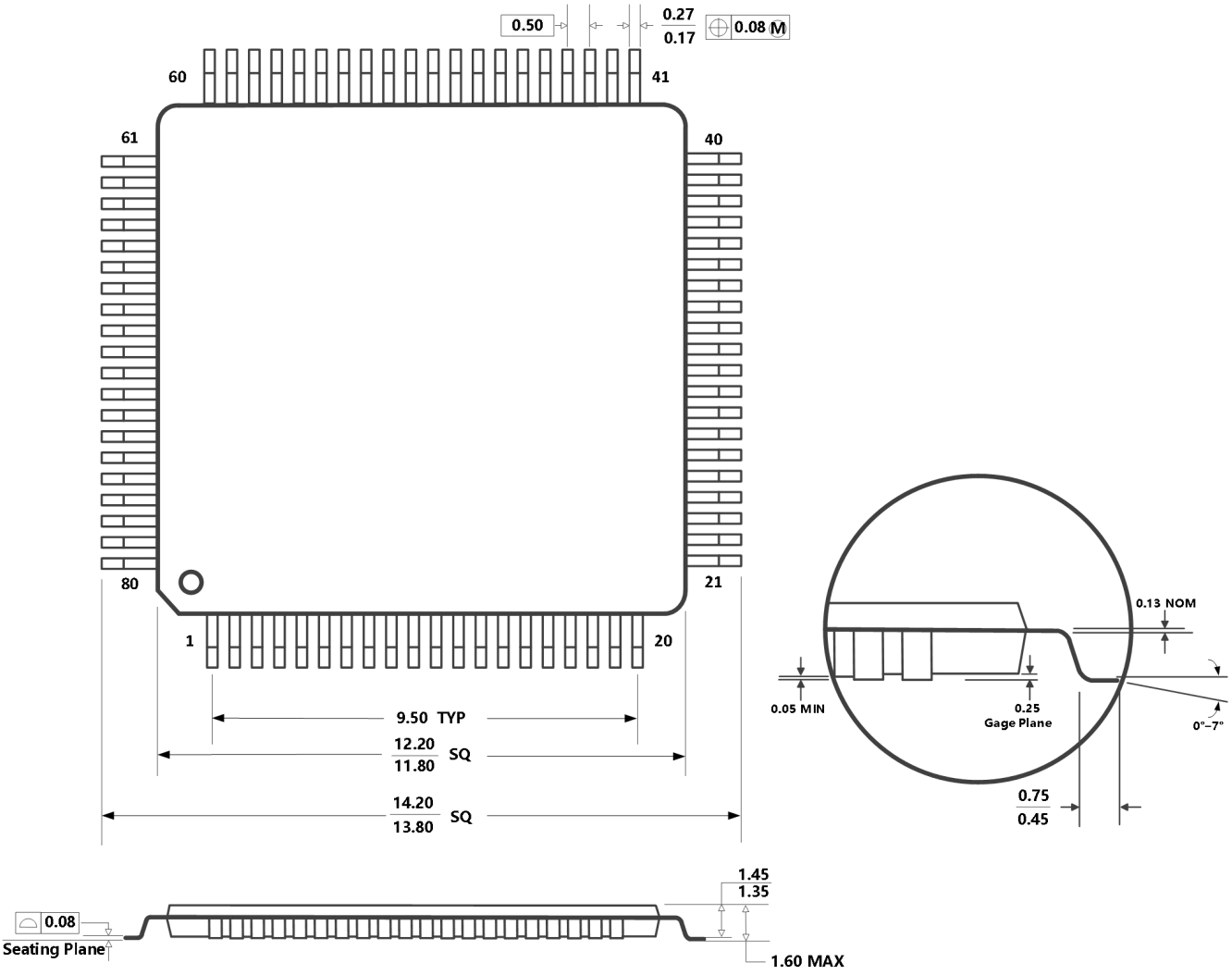
| 参数 | | 测试条件 | 最小值 | 最大值 | 单位 |
|---|---|---|---|---|---|
| $V_{OH}$ | D+,D–单端 | USB2.0 负载条件 | 2.8 | 3.6 | V |
| $V_{OL}$ | D+,D–单端 | USB2.0 负载条件 | 0 | 0.3 | V |
| Z(DRV) | D+,D–阻抗 | | 28 | 50 | Ω |
| $t_r$ | 上升时间 | 全速，差分，$C_L$=50pF，10%/90%，Rpu 处于D+上 | 4 | 20 | ns |
| $t_f$ | 下降时间 | 全速，差分，$C_L$=50pF，10%/90%，Rpu 处于D+上 | 4 | 20 | ns |

# 6. 封装信息

**LQFP100** **mm**



# LQFP80 mm


0.50
0.27
0.17
0.08 M
60
41
61
40
80
21
1
20
9.50 TYP
12.20
11.80
SQ
14.20
13.80
SQ
0.13 NOM
0.05 MIN
0.25
Gage Plane
0°–7°
0.75
0.45
1.45
1.35
0.08
Seating Plane
1.60 MAX

# 7. 重要注意事项及声明

Advancechip 均以“原样”提供技术性及可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证其中不含任何瑕疵，且不做任何明示或暗示的担保，包括但不限于对适销性、适合某特定用途或不侵犯任何第三方知识产权的暗示担保。

所述资源可供专业开发人员应用 Advancechip 产品进行设计使用。您将对以下行为独自承担全部责任：(1)针对您的应用选择合适的 Advancechip 产品；(2)设计、验证并测试您的应用；(3)确保您的应用满足相应标准以及任何其他安全、安保或其他要求。所述资源如有变更，恕不另行通知。进芯电子对您使用所述资源的授权仅限于开发资源所涉及 Advancechip 产品的相关应用。除此之外不得复制或展示所述资源，也不提供其它进芯电子或任何第三方的知识产权授权许可。如因使用所述资源而产生任何索赔、赔偿、成本、损失及债务等，进芯电子对此概不负责，并且您须赔偿由此对进芯电子及其代表造成的损害。

# 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：** 长沙高新开发区东方红街道东方红北路 601 号

湖南媒体艺术产业园 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