

CLM320F28335数字信号控制器

产品特性

- 高性能静态 CMOS 技术
 - 高达 150MHz (6.67ns 周期时间)
 - 1.9V/1.8V内核, 3.3V I/O 设计
- 高性能 32 位 CPU
 - IEEE-754 单精度浮点单元 (FPU)
 - 16 x 16和32 x 32介质访问控制 (MAC) 运算
 - 16 x 16 双 MAC
 - 哈佛 (Harvard) 总线架构
 - 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)
- 6通道DMA 处理器 (用于ADC, McBSP, ePWM, XINTF 和 SARAM)
- 16 位或 32 位外部接口 (XINTF)
 - 超过 2M x 16 地址范围
- 片载存储器
 - 256Kx16 闪存, 34Kx16 SARAM
 - 1K x 16 一次性可编程 (OTP) ROM
- 引导 ROM (8K X 16)
 - 支持软件引导模式 (通过 SCI,SPI,CAN,I2C,McBSP,XINTF和并行I/O)
 - 标准数学表
- 时钟和系统控制
 - 支持动态锁相环(PLL)比率变化
 - 片载振荡器
 - 安全装置定时器模块
- GPIO0到GPIO63引脚可以连接到八个外部内核中断其中的一个
- 可支持全部 58 个外设中断的外设中断扩展 (PIE) 块
- 128 位安全密钥/锁
 - 保护闪存 / OTP/RAM 模块
 - 防止固件逆向工程
- 低功耗模式和省电模式
 - 支持 IDLE (空闲),STANDBY (待机)、HALT (暂停) 模式
- 字节序 : 小端序
- 工作温度范围 : -40°C至125°C
- 增强型控制外设
 - 多达 18 个脉宽调制 (PWM) 输出
 - 高达 6 个支持 150ps 微边界定位 (MEP) 分辨率的高分辨率脉宽调制器 (HRPWM) 输出
 - 高达 6 个事件捕捉输入
 - 多达两个正交编码器接口
 - 高达 8 个 32 位定时器 (6 个 eCAP 以及 2 个 eQEP)
 - 高达 9 个 32 位定时器 (6 个 ePWM 以及 3 个 XINTCTR)
- 三个32位CPU定时器
- 串行端口外设
 - 2 个控制器局域网 (CAN) 模块
 - 3 个 SCI (UART) 模块
 - 2 个 McBSP 模块 (可配置为 SPI)
 - 1 个 SPI 模块
 - 1 个内部集成电路 (I2C) 总线
- 12位模数转换器(ADC), 16个通道
 - 80ns 转换速率
 - 2 x 8 通道输入多路复用器
 - 两个采样保持
 - 单个/同步转换
 - 内部或外部基准
- 多达88个具有输入滤波功能可单独编程的多路复用通用输入输出(GPIO)引脚
- JTAG 边界扫描支持
- 高级仿真特性
 - 分析和断点功能
 - 硬件实时调试
- 开发支持包括
 - ANSI C/C++ 编译器/汇编器/连接器
 - Code Composer Studio™ IDE
 - DSP/BIOS™ 和 SYS/BIOS
 - 数字电机控制和数字电源软件库
- 封装选项
 - LQFP-176

订购信息

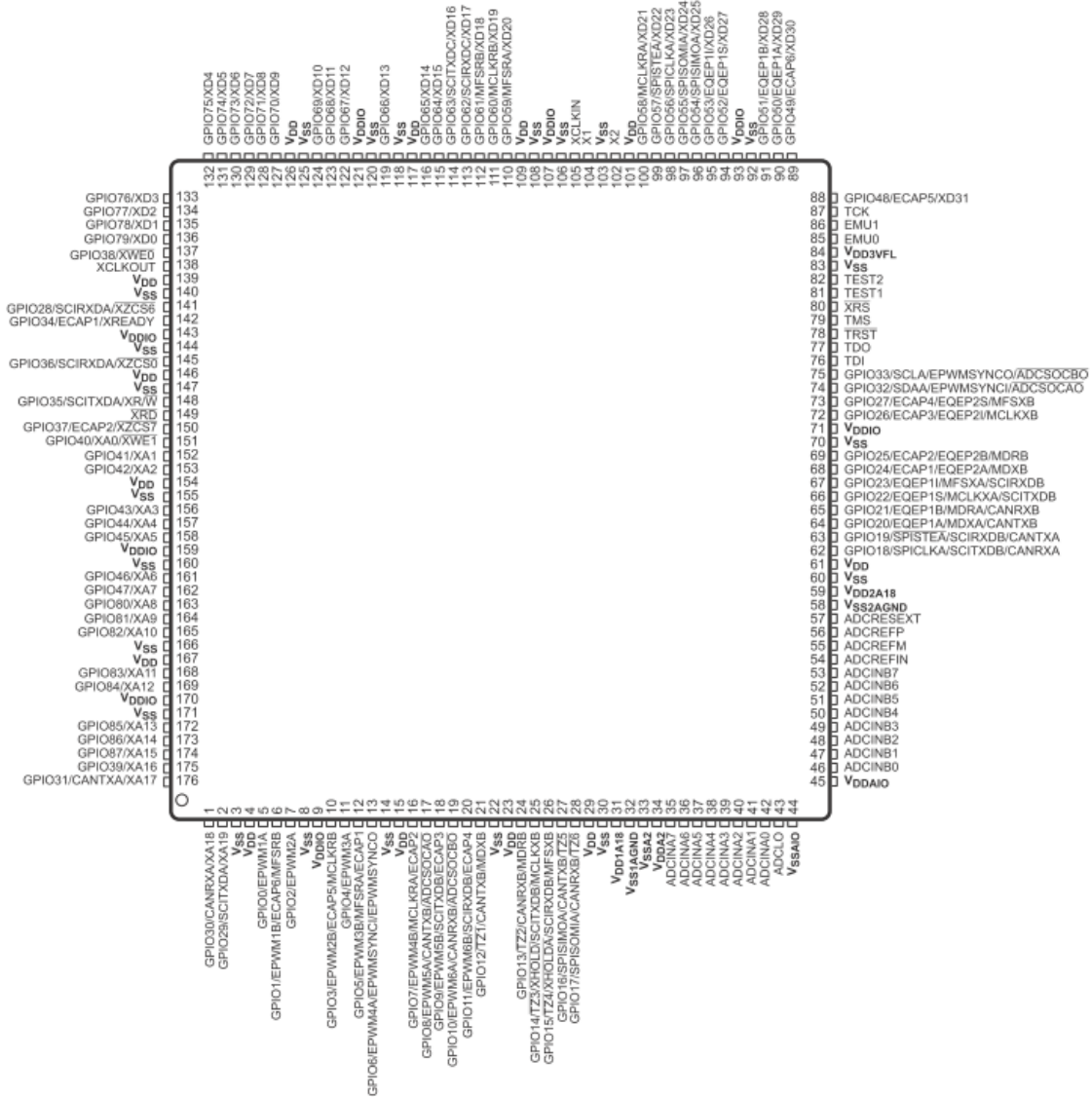
| 零件号 | 温度范围 | 封装 | 状态 |
|------------------|---------------|-----------|------|
| CLM320F28335PGFA | -40°C ~ 85°C | LQFP-176 | 正常发货 |
| CLM320F28335PTPQ | -40°C ~ 125°C | HLQFP-176 | 正常发货 |
| CLM320F28335PTPS | -40°C ~ 125°C | HLQFP-176 | 正常发货 |
| CLM320F28335ZJZA | -40°C ~ 85°C | BGA-176 | 正常发货 |
| CLM320F28335ZJZS | -40°C ~ 125°C | BGA-176 | 正常发货 |
| CLM320F28335ZJZQ | -40°C ~ 125°C | BGA-176 | 正常发货 |

器件特性

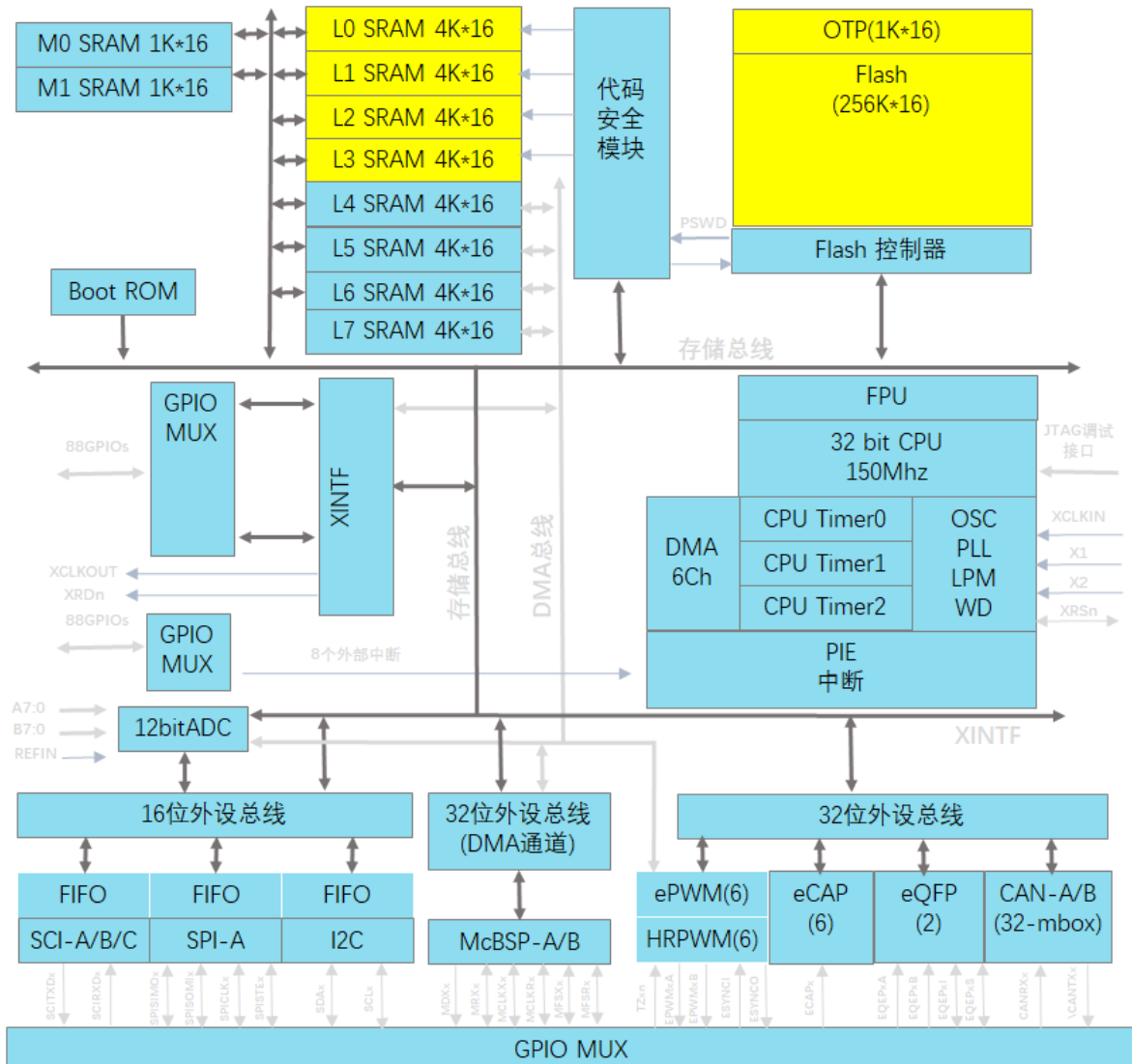
| | |
|----------------------------------|-----------------------|
| 特性 | CLM320F28335(150MHz) |
| 指令周期 | 6.67ns |
| 浮点单元 | 支持 |
| 3.3V 片载闪存(16位字) | 256K |
| 单周期访问 RAM (SARAM) (16 位字) | 34K |
| 一次性可编程 (OTP) ROM (16 位字) | 1K |
| 针对片上闪存/SARAM/OTP 块的代码安全 | 支持 |
| 引导 ROM (8K×16) | 支持 |
| 16/32 位外部接口 (XINTF) | 是 |
| 6通道直接内存存取 (DMA) | 是 |
| PWM通道 | ePWM1/2/3/4/5/6 |
| HRPWM通道 | ePWM1A/2A/3A/4A/5A/6A |
| 32位捕捉输入或辅助PWM输出 | eCAP1/2/3/4/5/6 |
| 32 位正交编码器脉冲 (QEP) 通道 (四个输入/通道) | eQEP1/2 |
| 看门狗定时器 | 支持 |
| 16通道12 位 ADC | 是 |
| 32 位 CPU 定时器 | 3 |
| 多通道缓冲串行端口 (McBSP)/ SPI | 2(A/B) |
| 串行外设接口 (SPI) | 1 |
| 串行通信接口 (SCI) (兼容 UART) | 3(A/B/C) |
| 增强型控制器局域网 (eCAN) | 2(A/B) |
| 内部集成电路 (I2C) | 1 |
| 通用 I/O 引脚 (共用) | 88 |
| 外部中断 | 8 |

引脚分配

下图显示了 176 引脚薄型四方扁平封装 (LQFP) 引脚分配。



功能框图



功能说明

内存总线（哈佛总线架构）

与很多 MCU 类型器件一样，内存、外设和 CPU 之间使用多个总线来移动数据。芯片内存总线架构包括程序读取总线、数据读取总线 and 数据写入总线。程序读取总线包含 22 条地址线和 32 条数据线。数据读取总线 and 数据写入总线各由 32 条地址线和 32 条数据线组成。32 位宽数据总线可实现单周期 32 位运行。多总线结构，通常称为哈佛总线，使得芯片能够在单周期内取一个指令、读取一个数据值和写入一个数据值。所有连接在内存总线上的外设和内存对内存访问进行优先级设定。总的来说，内存总线访问的优先级可概括如下：

| | | |
|------|------|------------------------|
| 最高级： | 数据写入 | （内存总线上不能同时进行数据和程序写入。） |
| | 程序写入 | （内存总线上不能同时进行数据和程序写入。） |
| | 数据读取 | |
| | 程序读取 | （内存总线上不能同时进行程序读取和取指令。） |
| 最低级： | 取指令 | （内存总线上不能同时进行程序读取和取指令。） |

实时 JTAG 和分析

CLM320F28335 器件使用标准的 IEEE1149.1 JTAG 接口。此外，器件支持实时运行模式，在处理器正在运行、执行代码并且处理中断时，可修改存储器内容、外设、和寄存器位置。用户也可以通过非时间关键代码进行单步操作，同时可在没有干扰的情况下启用将被处理的时间关键中断。此器件在 CPU 的硬件内执行实时模式。这是 CLM320F28335 器件所独有的特性，无需软件监控。此外，还提供了特别分析硬件以实现硬件断点或者数据/地址观察点的设置并当一个匹配发生时生成不同的用户可选中断事件。

外部接口(XINTF)

该异步接口由 20 位地址总线，32 位数据总线，和 3 个芯片选信号组成。此芯片选择线路被映射到 3 个外部区域，即区域 0、6 和 7。3 个区域中的每个区域可被设定为不同的等待状态数量、选通信号设置和保持时序，并且每个区域可被外部设定为扩展等待状态或者没有扩展等待状态。设定的等待状态、芯片选择和可编程选通时序可实现到外部存储器和外设的无缝对接。

闪存

CLM320F28335 器件包含 256K x16 的嵌入式闪存，该闪存分成 8 个 32K x16 扇区。器件还包含一个单 1K x 16 OTP 内存，其地址范围为 0x380400-0x3807FF。用户能够在不改变其它扇区的同时单独对一个闪存扇区进行擦除、编程和验证。然而，不能使用闪存的一个扇区或者这个 OTP 来执行对其它扇区进行擦除/编程的闪存算法。提供了特殊内存流水线操作以使闪存模块实现更高性能。闪存/OTP 被映射到程序和数据空间；因此，它可用于执行代码或者存储数据信息。请注意地址 0x33FFF0-0x33FFF5 为数据变量保留且不应包含程序代码。

SARAM

芯片包含两个单周期访问存储器块 M0 和 M1 SARAM，每个大小为 1K x 16。复位时，堆栈指针指向块 M1 的开始位置。M0 和 M1 块被映射到程序和数据空间。因此，用户能够使用 M0 和 M1 来执行代码或者用于数据变量。分区在连接器内执行。芯片提供了一个到编程器的统一内存映射。这使得用高级语言编程变得更加容易。

芯片包含了 32K x 16 的 13 个单周期访问 RAM (L0, L1, L2, L3, L4, L5, L6, L7, H0, H1, H2, H3, H4, H5 SARAM)，每个 RAM 分为 8 个区块 (L0-L7 且每块 4K)。每个块可被独立访问以大大减少 CPU 管线延迟。每个块被映射到程序和数据空间。L4, L5, L6, L7 可由 DMA 访问。

引导 ROM

- 引导 ROM

引导 ROM 由厂家使用引导载入软件进行设定。引导 ROM 程序在器件复位后，检查四个 GPIO 引脚状态后确认芯片进入相应的引导模式。例如，用户可以选择执行已经出现在内部闪存中的代码或者通过几个串行端口中

| 模式 | GPIO87/XA15 | GPIO86/XA14 | GPIO85/XA13 | GPIO84/XA12 | 模式描述 |
|----|-------------|-------------|-------------|-------------|-------------------|
| F | 1 | 1 | 1 | 1 | 跳转到闪存 |
| E | 1 | 1 | 1 | 0 | SCI-A boot |
| D | 1 | 1 | 0 | 1 | SPI-A 引导 |
| C | 1 | 1 | 0 | 0 | I2C-A 引导 |
| B | 1 | 0 | 1 | 1 | eCAN-A 引导 |
| A | 1 | 0 | 1 | 0 | McBSP-A 引导 |
| 9 | 1 | 0 | 0 | 1 | 跳转到 XINTF x16 |
| 8 | 1 | 0 | 0 | 0 | 跳转到 XINTF x32 |
| 7 | 0 | 1 | 1 | 1 | 跳转到 OTP |
| 6 | 0 | 1 | 1 | 0 | 并行 GPIO I/O 引导 |
| 5 | 0 | 1 | 0 | 1 | 并行XINTF引导 |
| 4 | 0 | 1 | 0 | 0 | 跳转至SARAM |
| 3 | 0 | 0 | 1 | 1 | 分支到检查引导模式 |
| 2 | 0 | 0 | 1 | 0 | 跳转到闪存,跳过ADC校准 |
| 1 | 0 | 0 | 0 | 1 | 跳转至SARAM,跳过ADC 校准 |
| 0 | 0 | 0 | 0 | 0 | 跳转至SCI,跳过ADC校准 |

的一个将全新的程序下载至内部 RAM。引导 ROM 还包含用于数学相关算法中的标准表，例如 SIN/COS 波形。

- 引导加载器使用的外设引脚

下表显示了每一个外设引导加载器所使用的 GPIO 引脚。请参阅 GPIO 多路复用器表以检查这些引脚是否与您希望在应用中使用的任一外设冲突。

| 引导加载器 | 外设加载器引脚 |
|-------|--|
| SCI-A | SCIRXDA (GPIO28) SCITXDA (GPIO29) |
| SPI-A | SPISIMOA (GPIO16) SPISOMIA (GPIO17) SPICLKA (GPIO18) SPISTEA (GPIO19) |
| I2C | SDAA (GPIO32) SCLA (GPIO33) |
| CAN | CANRXA (GPIO30) CANTXA (GPIO31) |
| McBSP | MDXA (GPIO20) MDRA (GPIO21) MCLKXA (GPIO22) MFSXA (GPIO23) MCLKRA (GPIO7) MFSRA (GPIO5) |

安全性

此芯片支持高级别安全以保护用户固件不受逆向工程损坏。该加密装置有一个 128 位密码（针对 16 个等待状态的硬编码），此密码由用户编辑写入闪存。一个代码安全模块（CSM）被用于保护闪存/OTP 和 L0/L1/L2/L3 SARAM 块。这个安全特性防止未经授权的用户通过 JTAG 端口检查内存内容，从外部内存执行代码或者试图引导加载一些将会输出安全内存内容的恶意软件。为了启用到安全块的访问，用户必须写入与存储在闪存/ROM 密码位置内的值相匹配的正确的 128 位“KEY（密钥）”值。

除了 CSM，仿真代码安全逻辑电路（ECSL）也已经被实现用来防止未经授权的用户安全代码。在仿真器连接时，任何对于闪存、用户 OTP 或者 L0, L1, L2, L3 内存的代码或者数据访问将触发 ECSL 并断开仿真连接。为了实现安全代码仿真，同时保持 CSM 安全内存读取，用户必须向 KEY 寄存器的低 64 位写入正确的值，这个值与存储在闪存密码位置的低 64 位的值相符合。请注意仍须执行闪存内所有 128 位密钥的伪读取。如果密码位置的低 64 位为全 1（未被编辑），那么无须符合 KEY 值。

当对闪存内被编辑的密码区（即安全的）进行最初调试时，CPU 将开始运行并可执行一个指令来访问一个受保护的 ECSL 区域。如果这一情况发生，ECSL 将发生错误并使仿真器连接被断开。这个问题有两个解决方案：

- 1) 首先是使用在等待中复位的仿真模式，该模式将保持器件在复位状态直到仿真器获得控制权。仿真器必须支持此选项的这种模式。
- 2) 第二种选择是使用“分支至检查引导模式”引导选项。这将进入一个环路，并不断轮询引导模式选择引脚。通过重新映射 PC 到另一个地址，或通过把引导模式选择引脚更改为所需的引导模式，用户可以选择此引导模式，然后在仿真器被连接时退出这种模式。

注：

- a) 当代码安全密钥被编辑时，0xFF3380 到 0x33FFF5 间的所有地址不能被用作程序代码或者数据。这些位置必须被设定为 0x0000
- b) 如果代码安全特性未被使用，地址 0x33FF80 至 0x33FFEF 可被用于代码或者数据。地址 0x33FFF0-0x33FFF5 为数据保留且不能包含程序代码。
- c) 128 位密码（位于 0x33FFF8-0x33FFFF）不能写入全零。一旦写入全零将永久锁住此器件，请务必慎重。

外设中断扩展 (PIE)

PIE 块将许多中断源复用至中断输入的较小的集合中。PIE 块能够支持多达 96 个外设中断。在 CLM320F28335 中，96 中断中的 58 个被外设使用。96 个中断被分成 8 组，每组被提供 12 个 CPU 中断线（INT1 或者 INT12）中的 1 个。96 个中断中的每一个中断由其存储在一个可被用户写覆盖的专用 RAM 块中的矢量支持。在处理这个中断时，这个矢量由 CPU 自动抽取。抽取这个矢量以及保存关键 CPU 寄存器将花费 8 个 CPU 时钟周期。因此 CPU 能够对中断事件做出快速响应。可以通过硬件和软件控制中断的优先级。每个中断都可以在 PIE 块内启用或禁用。

外部中断 (XINT1-XINT7, XNMI)

此器件支持 8 个被屏蔽的外部中断 (XINT1-XINT7, XNMI)。XNMI 可被连接至 INT13 或者 CPU 的 NMI 中断。这些中断中的每一个可被选择用于负边沿、正边沿或者正负边沿触发，并且可被启用或禁用（包括 XNMI 在内）。XINT1、XINT2 和 XNMI 还包含一个 16 位自由运行的加法计数器，当检测到有效的中断边沿时，该计数器复位为 0。这个计数器可被用于为中断精确计时。XINT1, XINT2 和 XINT 中断可接受来自 GPIO0-GPIO31 引脚的输入。XINT3-XINT7 中断可接受来自 GPIO32-GPIO63 引脚的输入。

振荡器和锁相环 (PLL)

此器件可由一个外部振荡器计时或者由一个连接到片载振荡器电路的晶振计时。提供的一个 PLL 支持高达 10 个输入时钟缩放比。PLL 比率可用软件中在器件运行时更改，这使得用户在需要低功耗运行时能够按比例降低运行频率。

看门狗

用户软件必须在特定的周期内定期复位 CPU 看门狗计数器；否则 CPU 看门狗将产生一个复位信号复位处理器。如果需要可将 CPU 看门狗禁用。

外设时钟

在外设闲置时，到每一个独立外设的时钟可被启用/禁用以减少功耗。此外，到串行端口（除了 I²C和 eCAN）和ADC的系统时钟可按照 CPU 时钟进行缩放。

低功耗模式

CLM320F28335器件是完全静态 CMOS 器件。提供三个低功耗模式：

➤ IDLE：

将 CPU 置于低功耗模式。可有选择性地关闭外设时钟并且只有那些在 IDLE 期间必须运行的外设保持运行状态。来自使能外设的已启用的中断或者看门狗将处理器从 IDLE 模式中唤醒。

➤ STANDBY：

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

➤ HALT：

关断内部振荡器的基础上，将器件置于尽可能低的功耗模式中。在此模式下可由一个复位信号或者外部信号将器件从这个模式中唤醒。

外设帧 0, 1, 2, 3 (PFn)

此器件将外设分成四个部分。外设映射如下：

| | | |
|-----|-------|------------------------|
| PF0 | PIE | PIE中断启用和控制寄存器加上PIE矢量表 |
| | 闪存 | 闪存等待状态寄存器 |
| | XINTF | 外部接口寄存器 |
| | DMA | DMA寄存器 |
| | 定时器 | CPU-定时器0, 1, 2寄存器 |
| | CSM | 代码安全模块KEY寄存器 |
| | ADC | ADC结果寄存器（双映射） |
| PF1 | eCAN | eCAN邮箱和控制寄存器 |
| | GPIO | GPIOMUX配置和控制寄存器 |
| | ePWM | 增强型脉冲宽度调制器模块和寄存器（双映射） |
| | eCAP | 增强型捕捉模块和寄存器 |
| | eQEP | 增强型正交解码器脉冲模块和寄存器 |
| PF2 | SYS | 系统控制寄存器 |
| | SCI | 串行通信接口(SCI)控制和RX/TX寄存器 |
| | SPI | 串行端口接口(SPI)和RX/TX寄存器 |
| | ADC | ADC状态、控制和结果寄存器 |
| | IC2 | 内部电路模块和寄存器 |
| | 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器模块和寄存器（双映射） |

通用输入/输出 (GPIO) 复用器

大多数的外设信号与通用输入/输出(GPIO)信号复用。这个复用使得用户能够在外设信号或者功能不使用时将一个引脚用作 GPIO。复位时所有 GPIO 引脚被配置为输入。针对 GPIO 模式或者外设信号模式，用户能够独立设定每一个引脚。对于特定的输入引脚，用户也可以选择输入限定周期的数量。这是为了过滤掉有害的噪音毛刺脉冲。GPIO 信号也可被用于使器件脱离特定低功耗模式。

32 位 CPU 定时器 (0, 1, 2)

CPU 定时器 0、1 和 2 是完全一样的 32 位定时器，这些定时器带有可预先设定的周期和 16 位时钟预分频。此定时器有一个 32 位倒计时寄存器，此寄存器在计数器达到 0 时生成一个中断。这个计数器的减量为被预分频值设置所分频的 CPU 时钟速度的值。当此计数器达到 0 时，它自动重新载入一个 32 位的周期值。CPU 定时器 2 为 DSP/BIOS 预留，并且连接到 CPU 的 INT14。如果 DSP/BIOS 未被使用，CPU 定时器 2 也可作为通用定时器使用。CPU 定时器 0 也为通用定时器并被连接至 PIE 模块。

控制外设

器件支持以下用于嵌入式控制和通信的外设：

| 外设名称 | 描述 |
|------|--|
| ePWM | 增强型PWM外设支持针对前缘和后缘边沿、被锁存的和逐周期触发机制的独立的和互补的PWM生成，可调节死区生成。某些PWM引脚支持HRPWM特性。ePWM寄存器由DMA支持，以便减少处理该外设的开销。 |
| eCAP | 增强型捕捉外设使用一个32位时基并在连续/单次捕捉模式中记录多达四个可编程事件。此外还可配置为生成辅助PWM信号。 |
| eQEP | 增强型QEP外设使用一个32位位置计数器，使用捕捉单元和一个32位单元定时器分别支持低速测量和高速测量。这个外设具有一个看门狗定时器来检测电机停转，并通过输入错误检测逻辑电路来识别QEP信号中的同步边沿转换。 |
| ADC | ADC是一个12位、单端、16通道的模数转换器。它包含两个用于同步采样的采样保持单元。ADC寄存器被DMA支持以便减少处理该外设的开销。 |

串行端口外设

| 外设名称 | 描述 |
|-------|---|
| eCAN | 该功能是CAN外设的增强型版本。它支持32个邮箱、消息时间戳并符合ISO11898-1(CAN2.0B)标准。 |
| McBSP | 多通道缓冲串行端口(McBSP)连接到E1/T1线路、调制解调器应用的语音质量编解码器，或高质量立体声音频DAC器件。McBSP接收和发送寄存器由DMA支持以大大减少处理这个外设所用的开销。如果需要，每一个McBSP模块可被配置为一个SPI。 |
| SPI | SPI是一个高速同步串行I/O端口，此端口允许已经过长度编程（1至16位）的串行位流以可编程的位传输速率移入和移出器件。通常，SPI用于MCU和外部外设或者其他处理器之间的通信。典型应用包含通过移位寄存器、显示驱动器和ADC等器件进行外部I/O或外设扩展。多器件通信由SPI的主/从操作支持。在2833x/2823x上，SPI包含一个16级接收和发送FIFO来减少中断服务开销。 |
| SCI | 串行通信接口是一种双线制异步串行端口，通常称为UART。SCI包含一个用于减少中断处理开销的16级接收和发送FIFO。 |
| I2C | 内部集成电路(I2C)模块提供一个MCU和其它器件（符合飞利浦半导体内部IC总线(I2C-bus)规范版本2.1并由一个I2C-bus相连）间的接口。通过这个I2C模块，连接在这个两线制总线上的外部组件能够发送8位数据到MCU或者从MCU接收8位数据。I2C包含一个用于减少中断处理开销的16级接收和发送FIFO。 |

外设

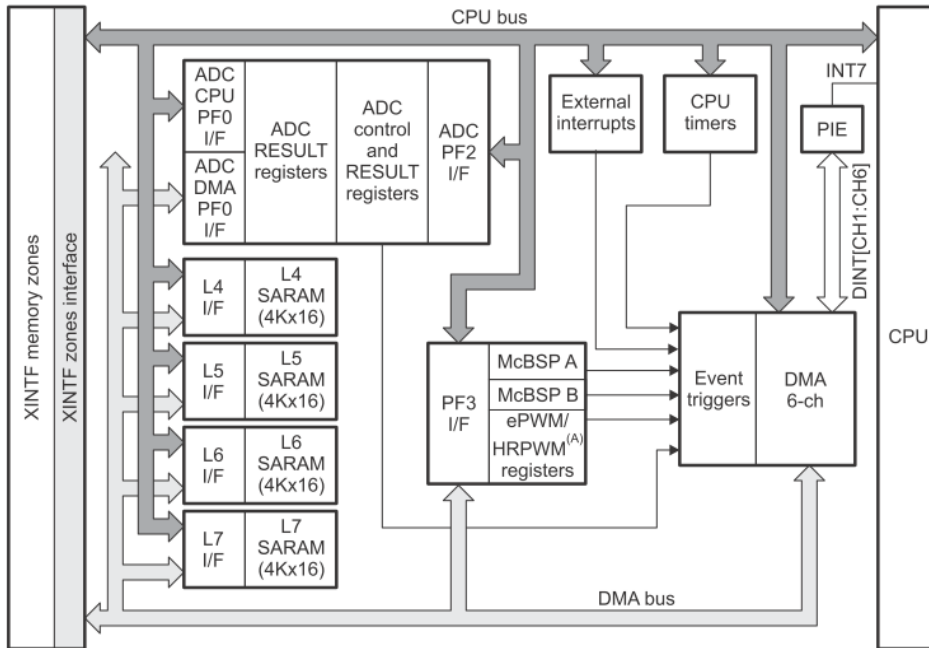
器件的集成外设有以下部分进行了说明：

- 1) 6通道直接内存存取(DMA)
- 2) 三个32位CPU定时器
- 3) 多达6个增强型PWM模块 (ePWM1、 ePWM2、 ePWM3、 ePWM4、 ePWM5、 ePWM6)
- 4) 高达6个增强型捕获模块(eCAP1 , eCAP2 , eCAP3 , eCAP4 , eCAP5 , eCAP6)
- 5) 多达2个增强型QEP模块 (eQEP1、 eQEP2)
- 6) 增强型模数转换器(ADC)模块
- 7) 多达2个增强型控制器局域网(eCAN)模块(eCAN-A , eCAN-B)
- 8) 多达3个串行通信接口模块(SCI-A , SCI-B , SCI-C)
- 9) 1个串行外设接口(SPI)模块 (SPI-A)
- 10) 内部集成电路(I2C)模块
- 11) 高达两个多通道缓冲串口(McBSP-A , McBSP-B)模块
- 12) 数字I/O和共用引脚功能
- 13) 外部接口(XINTF)

DMA概述

功能特性如下:

- 1) 6个独立的PIE中断通道
- 2) 多种中断触发源：
 - a) ePWMSOCA/SOCB
 - b) ADC序列发生器1和序列发生器2
 - c) McBSP-A和McBSP-B传输和接收逻辑
 - d) XINT1-7和XINT13
 - e) CPU定时器
 - f) 软件
- 3) 支持多种数据源和数据目标地址
- 4) 字大小：16位或32位 (McBSP限制到16位)
- 5) 吞吐量：4个周期/字 (McBSP读取时为5个周期/字)



A. ePWM 和 HRPWM 寄存器必须重新映射到 PF3 (通过 MAPCNF 寄存器的位 0) 之后才可以由 DMA 访问。

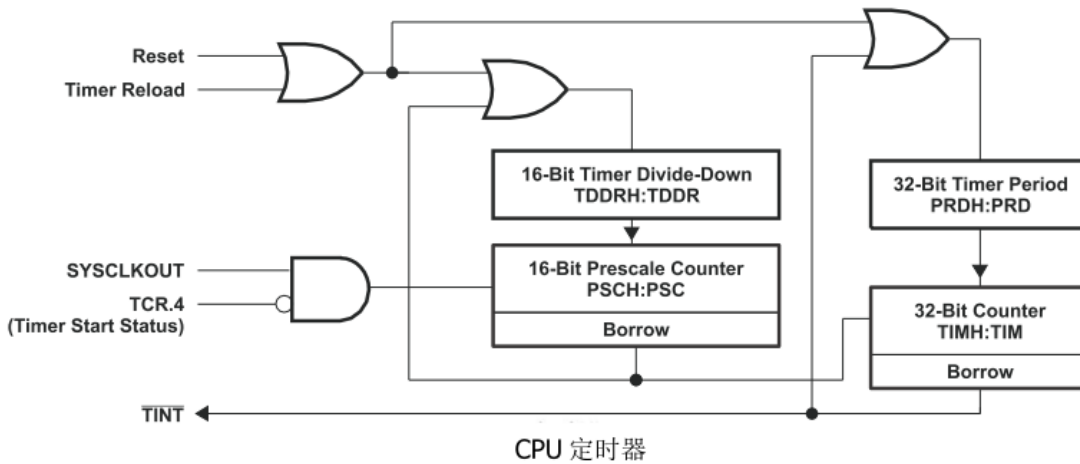
DMA 功能方框图

32位CPU定时器0，CPU定时器1，CPU定时器2

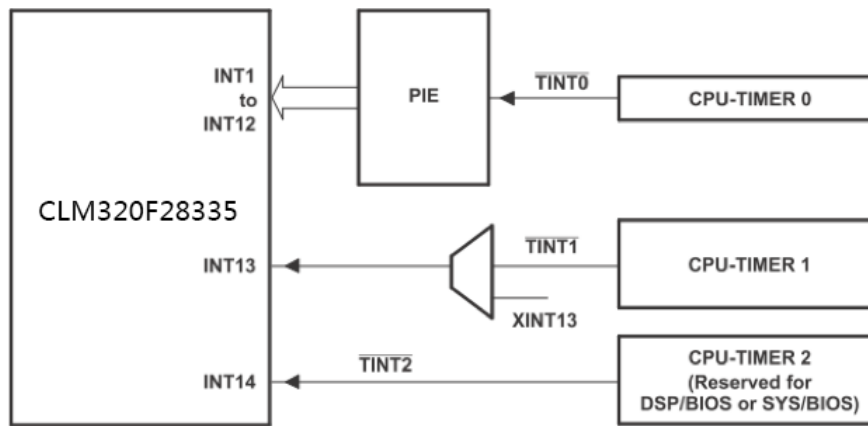
在器件上有3个32位CPU定时器(CPU定时器0，CPU定时器1，CPU定时器2)。

CPU 定时器 2 为 DSP/BIOS 或 SYS/BIOS 保留。可以在用户应用程序中使用 CPU 定时器 0 和定时器 1。这些定时器与 ePWM 模块中的定时器不同。

请注意：如果应用没有在使用 DSP/BIOS，那么 CPU 定时器 2 可被用在应用中。



定时器中断信号 (TINT0、TINT1、TINT2) 的连接如下图所示。



CPU 定时器中断信号和输出信号

定时器的通常操作如下：32位计数器寄存器“TIMH:TIM”被装入周期寄存器PRDH:PRD中的值。定时器寄存器按CPU的SYSCLKOUT速率递减。当计数器到达0时，一个定时器中断输出信号生成一个中断脉冲。

CPU 定时器 0,1,2 配置和控制寄存器如下表所示：

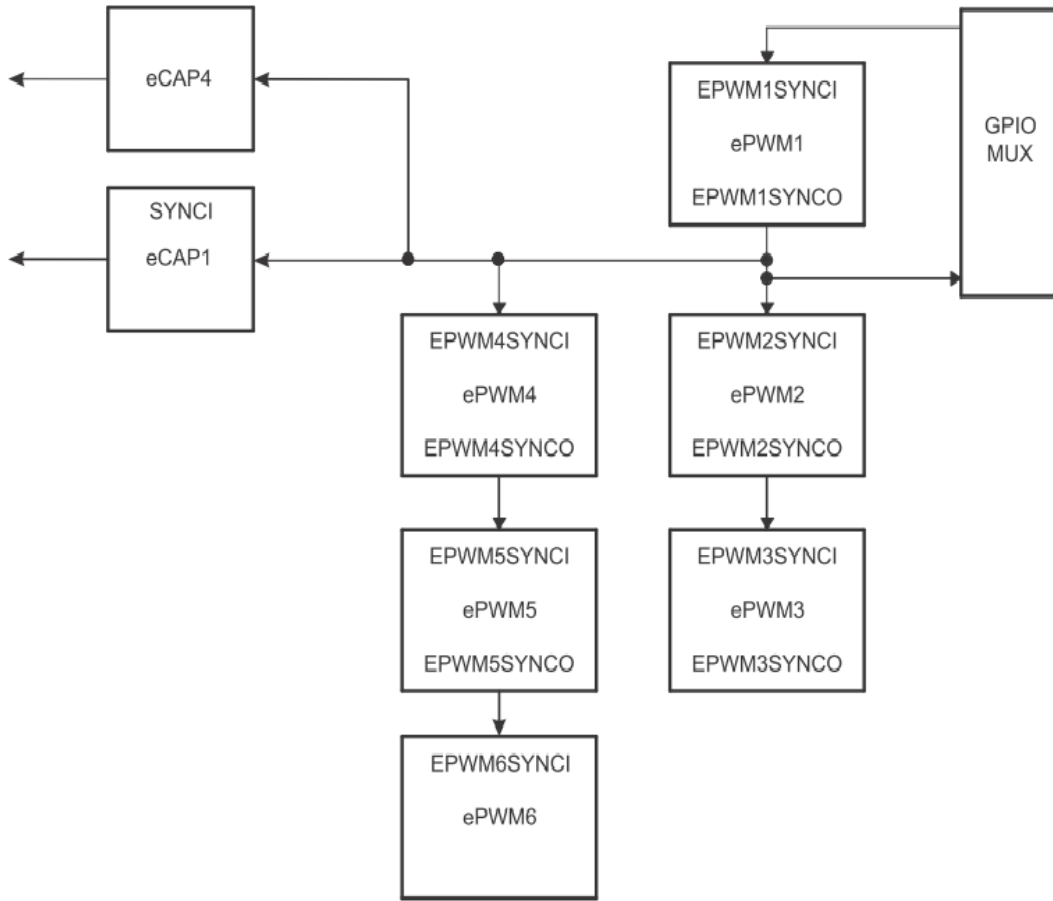
| 寄存器名称 | 地址 | 大小(x16) | 说明 |
|------------|--------|---------|--------------------|
| TIMER0TIM | 0x0C00 | 1 | CPU定时器0, 计数器寄存器 |
| TIMER0TIMH | 0x0C01 | 1 | CPU定时器0, 计数器寄存器高电平 |
| TIMER0PRD | 0x0C02 | 1 | CPU定时器0, 周期寄存器 |
| TIMER0PRDH | 0x0C03 | 1 | CPU定时器0, 周期寄存器高电平 |
| TIMER0TCR | 0x0C04 | 1 | CPU定时器0, 控制寄存器 |
| 保留 | 0x0C05 | 1 | |
| TIMER0TPR | 0x0C06 | 1 | CPU定时器0, 预分频寄存器 |
| TIMER0TPRH | 0x0C07 | 1 | CPU定时器0, 预分频寄存器高电平 |
| TIMER1TIM | 0x0C08 | 1 | CPU定时器1, 计数器寄存器 |
| TIMER1TIMH | 0x0C09 | 1 | CPU定时器1, 计数器寄存器高电平 |
| TIMER1PRD | 0x0C0A | 1 | CPU定时器1, 周期寄存器 |
| TIMER1PRDH | 0x0C0B | 1 | CPU定时器1, 周期寄存器高电平 |
| TIMER1TCR | 0x0C0C | 1 | CPU定时器1, 控制寄存器 |
| 保留 | 0x0C0D | 1 | |
| TIMER1TPR | 0x0C0E | 1 | CPU定时器1, 预分频寄存器 |
| TIMER1TPRH | 0x0C0F | 1 | CPU定时器1, 预分频寄存器高电平 |
| TIMER2TIM | 0x0C10 | 1 | CPU定时器2, 计数器寄存器 |
| TIMER2TIMH | 0x0C11 | 1 | CPU定时器2, 计数器寄存器高电平 |
| TIMER2PRD | 0x0C12 | 1 | CPU定时器2, 周期寄存器 |

| | | | |
|------------|---------------|----|--------------------|
| TIMER2PRDH | 0x0C13 | 1 | CPU定时器2, 周期寄存器高电平 |
| TIMER2TCR | 0x0C14 | 1 | CPU定时器2, 控制寄存器 |
| 保留 | 0x0C15 | 1 | |
| TIMER2TPR | 0x0C16 | 1 | CPU定时器2, 预分频寄存器 |
| TIMER2TPRH | 0x0C17 | 1 | CPU定时器2, 预分频寄存器高电平 |
| 保留 | 0x0C18-0x0C3F | 40 | |

增强型PWM模块

器件包括多达 6 个增强型 PWM (ePWM) 模块 (ePWM1 至 ePWM6) 。

下图显示了时基计数器同步方案 3 , 以及每个模块的完整 ePWM 寄存器组和表 8-5 显示了重新映射的寄存器配置。



时基计数器同步方案 3

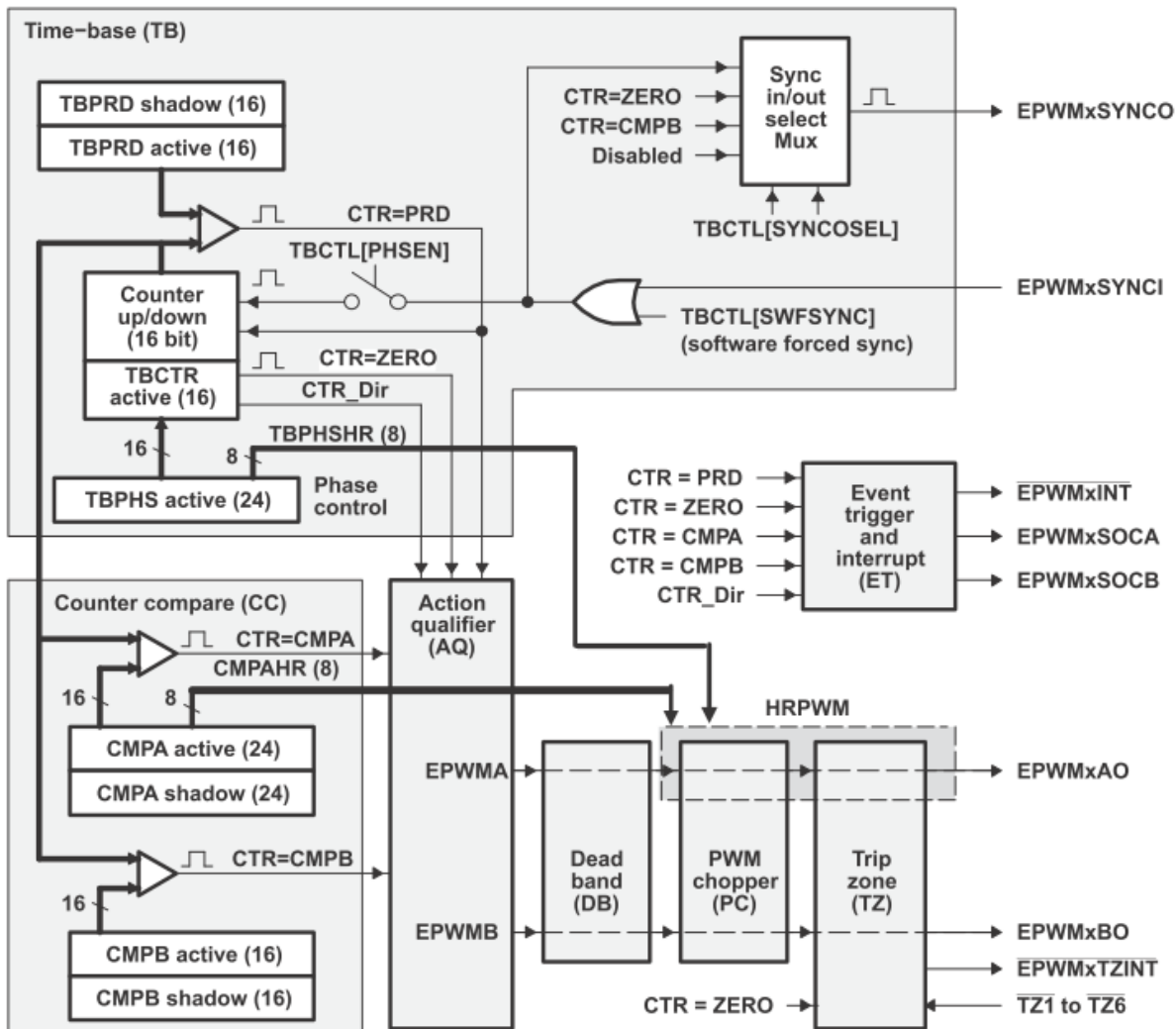
默认情况下，ePWM 和 HRPWM 寄存器会映射到外设帧 1 (PF1)。若要将寄存器重新映射至外设帧 3 (PF3) 来启用 DMA 访问，MAPCNF 寄存器（地址 0x702E）的位 0 (MAPEPWM) 必须设为 1。

ePWM 控制和状态寄存器（PF1 中的默认配置）如下表所示：

| 名称 | ePWM1 | ePWM2 | ePWM3 | ePWM4 | ePWM5 | ePWM6 | 大小(x16)/ #SHADOW | 说明 |
|---------|--------|--------|--------|--------|--------|--------|---------------------|----------------|
| TBCTL | 0x6800 | 0x6840 | 0x6880 | 0x68C0 | 0x6900 | 0x6940 | 1/0 | 时基控制寄存器 |
| TBSTS | 0x6801 | 0x6841 | 0x6881 | 0x68C1 | 0x6901 | 0x6941 | 1/0 | 时基状态寄存器 |
| TBPHSHR | 0x6802 | 0x6842 | 0x6882 | 0x68C2 | 0x6902 | 0x6942 | 1/0 | 时基相位HRPWM寄存器 |
| TBPHS | 0x6803 | 0x6843 | 0x6883 | 0x68C3 | 0x6903 | 0x6943 | 1/0 | 时基相位寄存器 |
| TBCTR | 0x6804 | 0x6844 | 0x6884 | 0x68C4 | 0x6904 | 0x6944 | 1/0 | 时基计数器寄存器 |
| TBPRD | 0x6805 | 0x6845 | 0x6885 | 0x68C5 | 0x6905 | 0x6945 | 1/1 | 时基周期寄存器设置 |
| CMPCTL | 0x6807 | 0x6847 | 0x6887 | 0x68C7 | 0x6907 | 0x6947 | 1/0 | 计数器比较控制寄存器 |
| CMPAHR | 0x6808 | 0x6848 | 0x6888 | 0x68C8 | 0x6908 | 0x6948 | 1/1 | 时基比较A HRPWM寄存器 |
| CMPA | 0x6809 | 0x6849 | 0x6889 | 0x68C9 | 0x6909 | 0x6949 | 1/1 | 计数器比较A寄存器组 |
| CMPB | 0x680A | 0x684A | 0x688A | 0x68CA | 0x690A | 0x694A | 1/1 | 计数器比较B寄存器组 |

| 名称 | ePWM1 | ePWM2 | ePWM3 | ePWM4 | ePWM5 | ePWM6 | 大小(x16)/ #SHADOW | 说明 |
|---------|--------|--------|--------|--------|--------|--------|---------------------|------------------|
| AQCTLA | 0x680B | 0x684B | 0x688B | 0x68CB | 0x690B | 0x694B | 1/0 | 用于输出A的操作限定器控制寄存器 |
| AQCTLB | 0x680C | 0x684C | 0x688C | 0x68CC | 0x690C | 0x694C | 1/0 | 用于输出B的操作限定器控制寄存器 |
| AQSFRC | 0x680D | 0x684D | 0x688D | 0x68CD | 0x690D | 0x694D | 1/0 | 操作限定器软件强制寄存器 |
| AQCSFRC | 0x680E | 0x684E | 0x688E | 0x68CE | 0x690E | 0x694E | 1/1 | 操作限定器连续S/W强制寄存器组 |
| DBCTL | 0x680F | 0x684F | 0x688F | 0x68CF | 0x690F | 0x694F | 1/1 | 死区生成器控制寄存器 |
| DBRED | 0x6810 | 0x6850 | 0x6890 | 0x68D0 | 0x6910 | 0x6950 | 1/0 | 死区生成器上升沿延迟计数寄存器 |
| DBFED | 0x6811 | 0x6851 | 0x6891 | 0x68D1 | 0x6911 | 0x6951 | 1/0 | 死区生成器下降沿延迟计数寄存器 |
| TZSEL | 0x6812 | 0x6852 | 0x6892 | 0x68D2 | 0x6912 | 0x6952 | 1/0 | 触发区选择寄存器 |
| TZCTL | 0x6814 | 0x6854 | 0x6894 | 0x68D4 | 0x6914 | 0x6954 | 1/0 | 跳匣区域控制寄存器 |
| TZEINT | 0x6815 | 0x6855 | 0x6895 | 0x68D5 | 0x6915 | 0x6955 | 1/0 | 跳匣区域启用中断寄存器 |
| TZFLG | 0x6816 | 0x6856 | 0x6896 | 0x68D6 | 0x6916 | 0x6956 | 1/0 | 触发区标志寄存器 |
| TZCLR | 0x6817 | 0x6857 | 0x6897 | 0x68D7 | 0x6917 | 0x6957 | 1/0 | 跳匣区域清除寄存器 |
| TZFRC | 0x6818 | 0x6858 | 0x6898 | 0x68D8 | 0x6918 | 0x6958 | 1/0 | 跳匣区域强制寄存器 |
| ETSEL | 0x6819 | 0x6859 | 0x6899 | 0x68D9 | 0x6919 | 0x6959 | 1/0 | 事件触发选择寄存器 |
| ETPS | 0x681A | 0x685A | 0x689A | 0x68DA | 0x691A | 0x695A | 1/0 | 事件触发预分频寄存器 |
| ETFLG | 0x681B | 0x685B | 0x689B | 0x68DB | 0x691B | 0x695B | 1/0 | 事件触发标志寄存器 |
| ETCLR | 0x681C | 0x685C | 0x689C | 0x68DC | 0x691C | 0x695C | 1/0 | 事件触发清除寄存器 |
| ETFRC | 0x681D | 0x685D | 0x689D | 0x68DD | 0x691D | 0x695D | 1/0 | 事件触发强制寄存器 |
| PCCTL | 0x681E | 0x685E | 0x689E | 0x68DE | 0x691E | 0x695E | 1/0 | PWM斩波器控制寄存器 |
| HRCNFG | 0x6820 | 0x6860 | 0x68A0 | 0x68E0 | 0x6920 | 0x6960 | 1/0 | HRPWM配置寄存器 |

ePWM 的信号互连情况如下图所示：



ePWM 子模块显示关键内部信号互连

重新映射的寄存器配置情况如下表所示：

| 名称 | ePWM1 | ePWM2 | ePWM3 | ePWM4 | ePWM5 | ePWM6 | 大小(x16)/ #SHADOW | 说明 |
|---------|--------|--------|--------|--------|--------|--------|---------------------|------------------|
| TBCTL | 0x5800 | 0x5840 | 0x5880 | 0x58C0 | 0x5900 | 0x5940 | 1/0 | 时基控制寄存器 |
| TBSTS | 0x5801 | 0x5841 | 0x5881 | 0x58C1 | 0x5901 | 0x5941 | 1/0 | 时基状态寄存器 |
| TBPHSHR | 0x5802 | 0x5842 | 0x5882 | 0x58C2 | 0x5902 | 0x5942 | 1/0 | 时基相位HRPWM寄存器 |
| TBPHS | 0x5803 | 0x5843 | 0x5883 | 0x58C3 | 0x5903 | 0x5943 | 1/0 | 时基相位寄存器 |
| TBCTR | 0x5804 | 0x5844 | 0x5884 | 0x58C4 | 0x5904 | 0x5944 | 1/0 | 时基计数器寄存器 |
| TBPRD | 0x5805 | 0x5845 | 0x5885 | 0x58C5 | 0x5905 | 0x5945 | 1/1 | 时基周期寄存器设置 |
| CMPCTL | 0x5807 | 0x5847 | 0x5887 | 0x58C7 | 0x5907 | 0x5947 | 1/0 | 计数器比较控制寄存器 |
| CMPAHR | 0x5808 | 0x5848 | 0x5888 | 0x58C8 | 0x5908 | 0x5948 | 1/1 | 时基比较A HRPWM寄存器 |
| CMPA | 0x5809 | 0x5849 | 0x5889 | 0x58C9 | 0x5909 | 0x5949 | 1/1 | 计数器比较A寄存器组 |
| CMPB | 0x580A | 0x584A | 0x588A | 0x58CA | 0x590A | 0x594A | 1/1 | 计数器比较B寄存器组 |
| AQCTLA | 0x580B | 0x584B | 0x588B | 0x58CB | 0x590B | 0x594B | 1/0 | 用于输出A的操作限定器控制寄存器 |
| AQCTLB | 0x580C | 0x584C | 0x588C | 0x58CC | 0x590C | 0x594C | 1/0 | 用于输出B的操作限定器控制寄存器 |

| 名称 | ePWM1 | ePWM2 | ePWM3 | ePWM4 | ePWM5 | ePWM6 | 大小(x16)/ #SHADOW | 说明 |
|---------|--------|--------|--------|--------|--------|--------|---------------------|------------------|
| AQSFRC | 0x580D | 0x584D | 0x588D | 0x58CD | 0x590D | 0x594D | 1/0 | 操作限定器软件强制寄存器 |
| AQCSFRC | 0x580E | 0x584E | 0x588E | 0x58CE | 0x590E | 0x594E | 1/1 | 操作限定器连续S/W强制寄存器组 |
| DBCTL | 0x580F | 0x584F | 0x588F | 0x58CF | 0x590F | 0x594F | 1/1 | 死区生成器控制寄存器 |
| DBRED | 0x5810 | 0x5850 | 0x5890 | 0x58D0 | 0x5910 | 0x5950 | 1/0 | 死区生成器上升沿延迟计数寄存器 |
| DBFED | 0x5811 | 0x5851 | 0x5891 | 0x58D1 | 0x5911 | 0x5951 | 1/0 | 死区生成器下降沿延迟计数寄存器 |
| TZSEL | 0x5812 | 0x5852 | 0x5892 | 0x58D2 | 0x5912 | 0x5952 | 1/0 | 触发区选择寄存器 |
| TZCTL | 0x5814 | 0x5854 | 0x5894 | 0x58D4 | 0x5914 | 0x5954 | 1/0 | 跳匣区域控制寄存器 |
| TZEINT | 0x5815 | 0x5855 | 0x5895 | 0x58D5 | 0x5915 | 0x5955 | 1/0 | 跳匣区域启用中断寄存器 |
| TZFLG | 0x5816 | 0x5856 | 0x5896 | 0x58D6 | 0x5916 | 0x5956 | 1/0 | 触发区标志寄存器 |
| TZCLR | 0x5817 | 0x5857 | 0x5897 | 0x58D7 | 0x5917 | 0x5957 | 1/0 | 跳匣区域清除寄存器 |
| TZFRC | 0x5818 | 0x5858 | 0x5898 | 0x58D8 | 0x5918 | 0x5958 | 1/0 | 跳匣区域强制寄存器 |
| ETSEL | 0x5819 | 0x5859 | 0x5899 | 0x58D9 | 0x5919 | 0x5959 | 1/0 | 事件触发选择寄存器 |
| ETPS | 0x581A | 0x585A | 0x589A | 0x58DA | 0x591A | 0x595A | 1/0 | 事件触发预分频寄存器 |
| ETFLG | 0x581B | 0x585B | 0x589B | 0x58DB | 0x591B | 0x595B | 1/0 | 事件触发标志寄存器 |
| ETCLR | 0x581C | 0x585C | 0x589C | 0x58DC | 0x591C | 0x595C | 1/0 | 事件触发清除寄存器 |
| ETFRC | 0x581D | 0x585D | 0x589D | 0x58DD | 0x591D | 0x595D | 1/0 | 事件触发强制寄存器 |
| PCCTL | 0x581E | 0x585E | 0x589E | 0x58DE | 0x591E | 0x595E | 1/0 | PWM斩波器控制寄存器 |
| HRCNFG | 0x5820 | 0x5860 | 0x58A0 | 0x58E0 | 0x5920 | 0x5960 | 1/0 | HRPWM配置寄存器 |

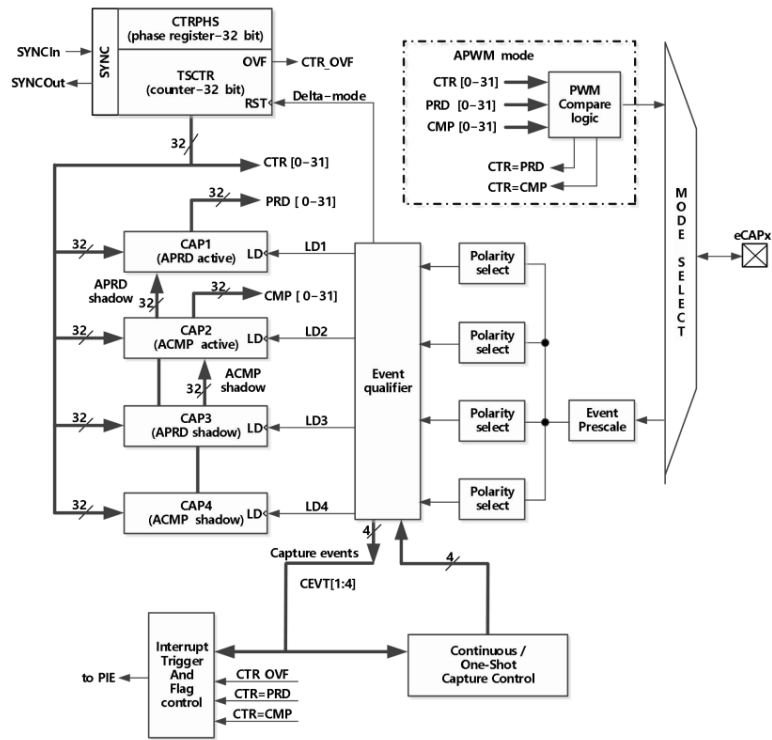
高分辨率PWM(HRPWM)

HRPWM模块提供PWM分辨率远好于使用传统数字PWM方法所能产生的分辨率。HRPWM模块的特点如下：

- 1) 大大提高了传统数字PWM的输出信号的分辨率
- 2) 通常在PWM的有效分辨率降低于大约9-10位时使用。当系统时钟为100MHz并且PWM频率大于200kHz时会发生这种情况。
- 3) 此功能可用于PWM占空比控制和相移控制。
- 4) 通过对ePWM模块的CMPA和相位寄存器的扩展来控制更加精细的时间粒度控制或者边沿定位。
- 5) 只有EPWMxA有HRPWM 功能

增强型捕获模块 (eCAP) 模块

器件包括6个增强型捕获(eCAP)模块(eCAP1 , eCAP2 , eCAP3 , eCAP4 , eCAP5和eCAP6)。下图显示了一个模块的功能框图。



eCAP 功能方框图

eCAP 模块使用系统时钟 (SYSCLKOUT) 的频率计时。

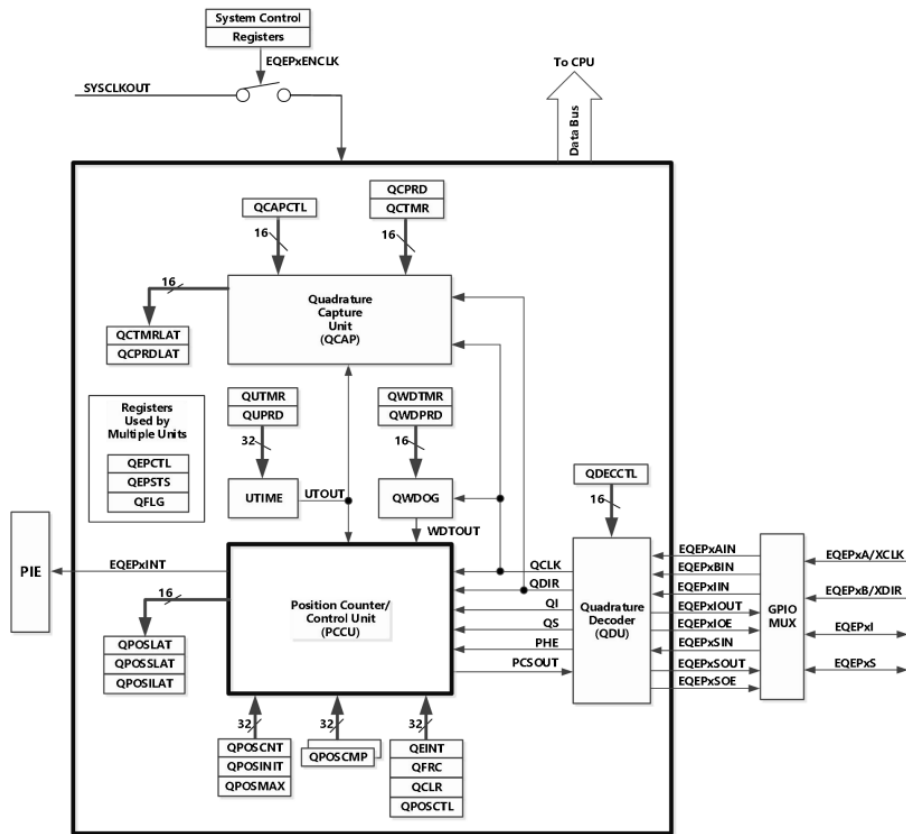
PCLKCR1 寄存器中的时钟使能位(ECAP1ENCLK,ECAP2ENCLK,ECAP3ENCLK,ECAP4EN - CLK,ECAP5ENCLK,ECAP6ENCLK)被单独用来关闭 eCAP 模块 (针对低功耗运行)。复位后, ECAP1ENCLK,ECAP2ENCLK,ECAP3ENCLK,ECAP4ENCLK,ECAP5ENCLK,ECAP6ENCLK 被设为低电平,表明外设时钟关闭。

eCAP 控制和状态寄存器配置情况如下表所示：

| 名称 | eCAP1 | eCAP2 | eCAP3 | eCAP4 | eCAP5 | eCAP6 | 大小(x16) | 说明 |
|--------|---------------|---------------|---------------|---------------|---------------|---------------|---------|-------------|
| TSCTR | 0x6A00 | 0x6A20 | 0x6A40 | 0x6A60 | 0x6A80 | 0x6AA0 | 2 | 时间戳计数器 |
| CTRPHS | 0x6A02 | 0x6A22 | 0x6A42 | 0x6A62 | 0x6A82 | 0x6AA2 | 2 | 计数器相位偏移值寄存器 |
| CAP1 | 0x6A04 | 0x6A24 | 0x6A44 | 0x6A64 | 0x6A84 | 0x6AA4 | 2 | 捕获1寄存器 |
| CAP2 | 0x6A06 | 0x6A26 | 0x6A46 | 0x6A66 | 0x6A86 | 0x6AA6 | 2 | 捕获2寄存器 |
| CAP3 | 0x6A08 | 0x6A28 | 0x6A48 | 0x6A68 | 0x6A88 | 0x6AA8 | 2 | 捕获3寄存器 |
| CAP4 | 0x6A0A | 0x6A2A | 0x6A4A | 0x6A6A | 0x6A8A | 0x6AAA | 2 | 捕获4寄存器 |
| 保留 | 0x6A0C-0x6A12 | 0x6A2C-0x6A32 | 0x6A4C-0x6A52 | 0x6A6C-0x6A72 | 0x6A8C-0x6A92 | 0x6AAC-0x6AB2 | 8 | 保留 |
| ECCTL1 | 0x6A14 | 0x6A34 | 0x6A54 | 0x6A74 | 0x6A94 | 0x6AB4 | 1 | 捕获控制寄存器1 |
| ECCTL2 | 0x6A15 | 0x6A35 | 0x6A55 | 0x6A75 | 0x6A95 | 0x6AB5 | 1 | 捕获控制寄存器2 |
| ECEINT | 0x6A16 | 0x6A36 | 0x6A56 | 0x6A76 | 0x6A96 | 0x6AB6 | 1 | 捕获中断使能寄存器 |
| ECFLG | 0x6A17 | 0x6A37 | 0x6A57 | 0x6A77 | 0x6A97 | 0x6AB7 | 1 | 捕获中断标志寄存器 |
| ECCLR | 0x6A18 | 0x6A38 | 0x6A58 | 0x6A78 | 0x6A98 | 0x6AB8 | 1 | 捕获中断清除寄存器 |
| ECFRC | 0x6A19 | 0x6A39 | 0x6A59 | 0x6A79 | 0x6A99 | 0x6AB9 | 1 | 捕获中断强制寄存器 |
| 保留 | 0x6A1A-0x6A1F | 0x6A3A-0x6A3F | 0x6A5A-0x6A5F | 0x6A7A-0x6A7F | 0x6A9A-0x6A9F | 0x6ABA-0x6ABF | 6 | 保留 |

增强型正交编码器脉冲(QEP)模块

器件有2个增强型正交编码器(eQEP)模块(eQEP1, eQEP2)。下图为eQEP模块的功能图



eQEP 功能方框图

eQEP控制和状态寄存器配置情况如下表所示：

| 名称 | eQEP1地址 | eQEP2地址 | eQEP1 大小(x16)/ #SHADOW | 说明 |
|----------|---------|---------|------------------------------|--------------|
| 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 |
| QOSLAT | 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捕捉控制寄存器 |

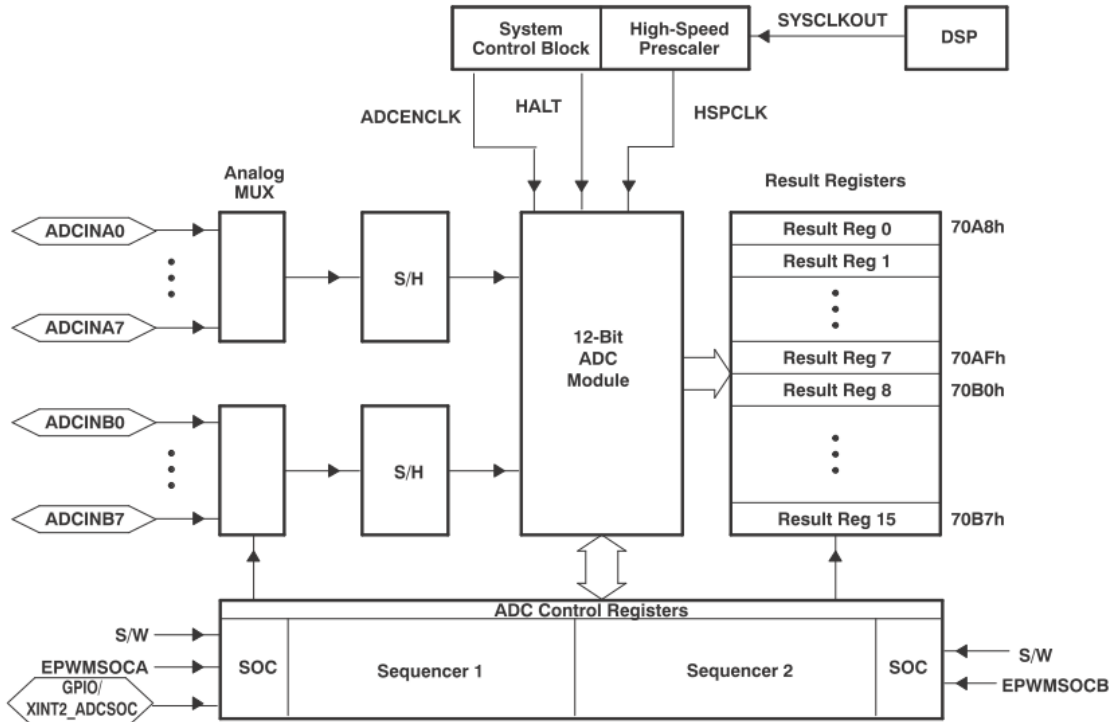
| 名称 | eQEP1地址 | eQEP2地址 | eQEP1 大小(x16)/ #SHADOW | 说明 |
|----------|---------------|-------------------|------------------------------|---------------|
| QPOSCTL | 0x6B17 | 0x6B57 | 1/0 | eQEP位置比较控制寄存器 |
| QEINT | 0x6B18 | 0x6B58 | 1/0 | eQEP中断使能寄存器 |
| QFLG | 0x6B19 | 0x6B59 | 1/0 | eQEP中断标志寄存器 |
| 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捕捉定时器 |
| QCPRD | 0x6B1E | 0x6B5E | 1/0 | eQEP捕捉周期寄存器 |
| QCTMRLAT | 0x6B1F | 0x6B5F | 1/0 | eQEP捕捉定时器锁存 |
| QCPRDLAT | 0x6B20 | 0x6B60 | 1/0 | eQEP捕捉周期锁存 |
| 保留 | 0x6B21-0x6B3F | 0x6B61-0x6B7 F | 31/0 | |

模数转换器(ADC)模块

ADC模块由一个带有内置采样保持(S/H)电路的12位ADC组成。ADC模块的功能包括：

- 1) 具有内置S/H的12位ADC内核
- 2) 模拟输入：0.0V至3.0V（高于3.0V的电压产生满量程转换结果）。
- 3) 快速转换率：在25MHzADC时钟、12.5MSPS条件下高达80ns
- 4) 16个专用ADC通道。每次采样/保持都有复用的8通道
- 5) 自动定序功能在单次会话中可提供多达16次“自动转换”。可将每次转换编程为选择16个输入信道中的任何一个。
- 6) 序列发生器可运行于2个独立的8态序列发生器，或作为1个较大的16态序列发生器（即2个级联的8态序列发生器）。
- 7) 16个用于存储转换值的结果寄存器（可分别寻址）
- 8) 作为转换开始序列(SOC)源的多触发器
 - S/W - 软件立即启动
 - ePWMM 转换开始
 - XINT2 ADC 转换开始
- 9) 灵活的中断控制允许每个序列结束(EOS)或每个其它EOS上的中断请求。
- 10) 序列发生器可运行于“启/停”模式，从而实现多个“时序触发器”同步转换。
- 11) SOCA和SOCB触发器可独立运行在双序列发生器模式中。
- 12) 采样保持(S/H)采集时间窗口具有独立的预分频控制。

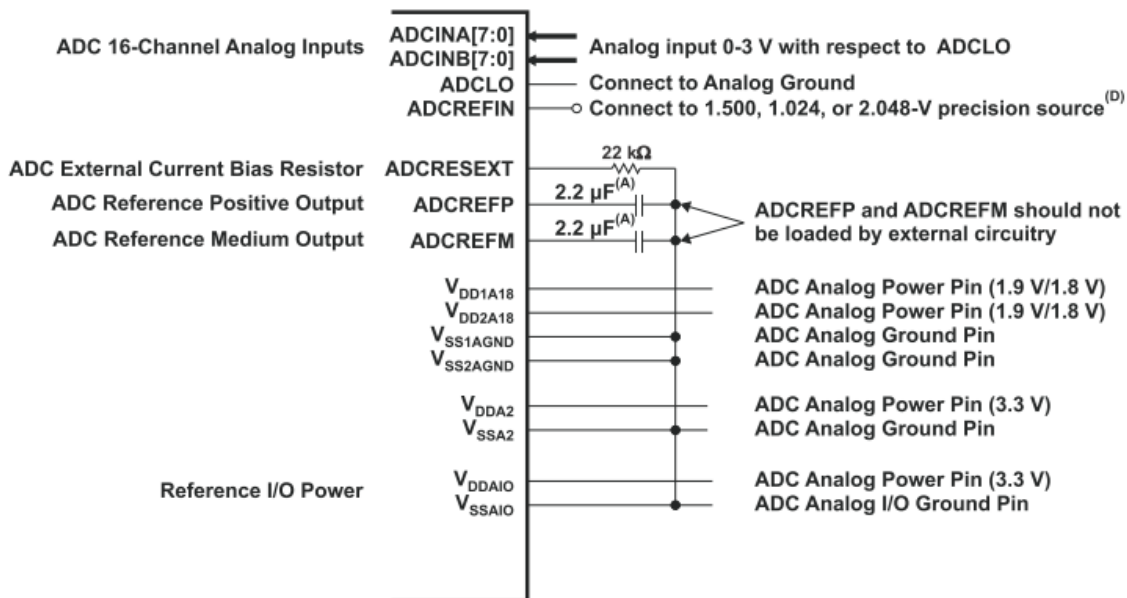
ADC简化功能框图如下图所示：



ADC 模块的方框图

ADC模块具有16个通道，并可配置为用于ePWM模块的2个独立的8通道模块。可将2个独立的8通道模块级联成1个16通道模块。尽管有多个输入通道和2个序列发生器，但在ADC模块中只有一个转换器。

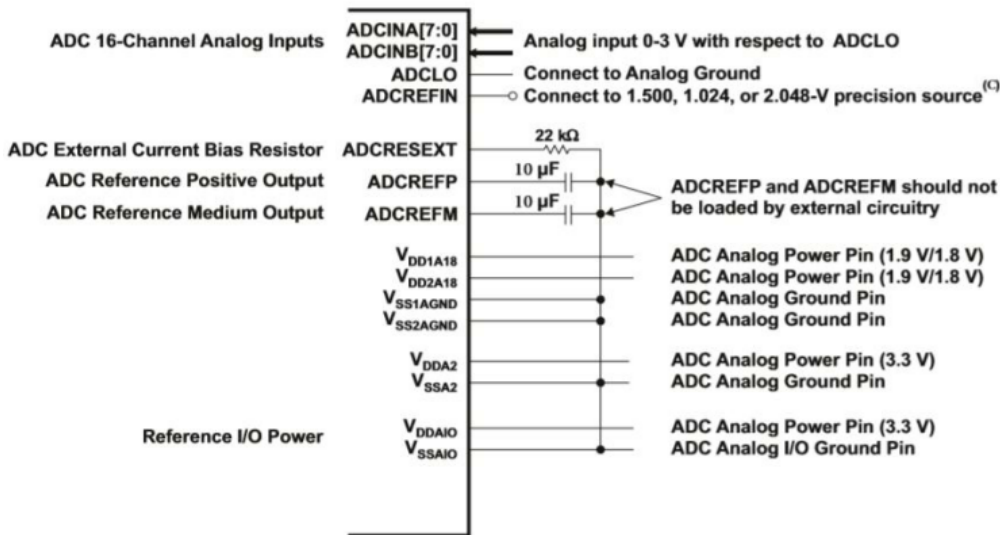
下图是ADC使用内部基准时的引脚连接方案



注：

- 1) 建议在所有电源引脚上使用外部去耦合电容器。
- 2) 为了防止ADC 性能降低，ADC模拟信号输入端必须被运算放大器驱动。

下图是ADC使用外部基准时的引脚连接方案



注：

- 1) 建议在所有电源引脚上使用外部去耦合电容器。
- 2) 为了防止ADC 性能降低，ADC模拟信号输入端必须被运算放大器驱动。

ADC未启用时ADC引脚连接建议：

- V_{DD1A18} / V_{DD2A18} - 连接至V_{DD}
- V_{DDA2} , V_{DDAIO} - 连接至V_{DDIO}
- V_{SS1AGND} / V_{SS2AGND} , V_{SSA2} , V_{SSAIO} - 连接至V_{SS}
- ADCLO - 连接至V_{SS}
- ADCREFIN - 连接至V_{SS}
- ADCREFP/ADCREFM - 连接一个100nF电容器至V_{SS}
- ADCRESEXT - 连接一个20kΩ电阻器（非常松散的耐受）至V_{SS}
- ADCINAn , ADCINBn - 连接至V_{SS}

当ADC未被使用时，为了达到节能的目的，请确保到ADC模块的时钟未被打开。当在一个应用中使用ADC模块时，未使用的ADC输入引脚应被连接至模拟接地(V_{SS1AGND}/V_{SS2AGND})

ADC的控制、状态寄存器如下表所示：

| 名称 | 地址 ⁽¹⁾ | 地址 ⁽²⁾ | 大小(x16) | 说明 |
|--------------|-------------------|-------------------|---------|-----------------|
| ADCTRL1 | 0x7100 | | 1 | ADC控制寄存器1 |
| ADCTRL2 | 0x7101 | | 1 | ADC控制寄存器2 |
| ADCMAXCONV | 0x7102 | | 1 | ADC最大转换信道数寄存器 |
| ADCCHSELSEQ1 | 0x7103 | | 1 | ADC信道选择定序控制寄存器1 |
| ADCCHSELSEQ2 | 0x7104 | | 1 | ADC信道选择定序控制寄存器2 |
| ADCCHSELSEQ3 | 0x7105 | | 1 | ADC信道选择定序控制寄存器3 |
| ADCCHSELSEQ4 | 0x7106 | | 1 | ADC信道选择定序控制寄存器4 |
| ADCASEQSR | 0x7107 | | 1 | ADC自动定序状态寄存器 |
| ADCRESULT0 | 0x7108 | 0x0B00 | 1 | ADC转换结果缓冲寄存器0 |
| ADCRESULT1 | 0x7109 | 0x0B01 | 1 | ADC转换结果缓冲寄存器1 |

| 名称 | 地址 ⁽¹⁾ | 地址 ⁽²⁾ | 大小(x16) | 说明 |
|------------|-------------------|-------------------|---------|----------------|
| ADCRESLT2 | 0x710A | 0x0B02 | 1 | ADC转换结果缓冲寄存器2 |
| ADCRESLT3 | 0x710B | 0x0B03 | 1 | ADC转换结果缓冲寄存器3 |
| ADCRESLT4 | 0x710C | 0x0B04 | 1 | ADC转换结果缓冲寄存器4 |
| ADCRESLT5 | 0x710D | 0x0B05 | 1 | ADC转换结果缓冲寄存器5 |
| ADCRESLT6 | 0x710E | 0x0B06 | 1 | ADC转换结果缓冲寄存器6 |
| ADCRESLT7 | 0x710F | 0x0B07 | 1 | ADC转换结果缓冲寄存器7 |
| ADCRESLT8 | 0x7110 | 0x0B08 | 1 | ADC转换结果缓冲寄存器 |
| ADCRESLT9 | 0x7111 | 0x0B09 | 1 | ADC转换结果缓冲寄存器9 |
| ADCRESLT10 | 0x7112 | 0x0B0A | 1 | ADC转换结果缓冲寄存器10 |
| ADCRESLT11 | 0x7113 | 0x0B0B | 1 | ADC转换结果缓冲寄存器11 |
| ADCRESLT12 | 0x7114 | 0x0B0C | 1 | ADC转换结果缓冲寄存器12 |
| ADCRESLT13 | 0x7115 | 0x0B0D | 1 | ADC转换结果缓冲寄存器13 |
| ADCRESLT14 | 0x7116 | 0x0B0E | 1 | ADC转换结果缓冲寄存器14 |
| ADCRESLT15 | 0x7117 | 0x0B0F | 1 | ADC转换结果缓冲寄存器15 |
| ADCTRL3 | 0x7118 | | 1 | ADC控制寄存器3 |
| ADCTRL4 | 0x711A | | 1 | ADC控制寄存器4 |
| ADCST | 0x7119 | | 1 | ADC状态寄存器 |
| 保留 | 0x711B | | 1 | 保留 |
| ADCREFSEL | 0x711C | | 1 | ADC基准选择寄存器 |
| ADCOFFTRIM | 0x711D | | 1 | ADC偏移调整寄存器 |
| 保留 | 0x711E | | 1 | 保留 |
| 保留 | 0x711F | | 1 | 保留 |

注：

- 1) 本列中的寄存器为外设帧2寄存器
- 2) ADC结果寄存器是双映射。外设帧2(0x7108-0x7117)中的位置为2等待状态，且为左对齐。外设帧0空间(0x0B00-0x0B0F)的位置对CPU访问是1等待状态和对于DMA访问是0等待状态，右对齐。在ADC的高速/连续转换使用期间，使用0等待状态位置进行ADC结果到用户内存的快速转换。

ADC校准

ADC_cal()例程被工厂编程到预留的OTP存储器中。引导ROM自动调用ADC_cal()例程来使用特定器件的校准数据来初始化ADCREFSEL和ADCOFFTRIM寄存器。正常运行期间，这个过程中会自动发生，无需用户进行任何操作。

如果在开发过程中，引导ROM被CodeComposerStudio绕过，那么ADCREFSEL和ADCOFFTRIM就必须由应用进行初始化。

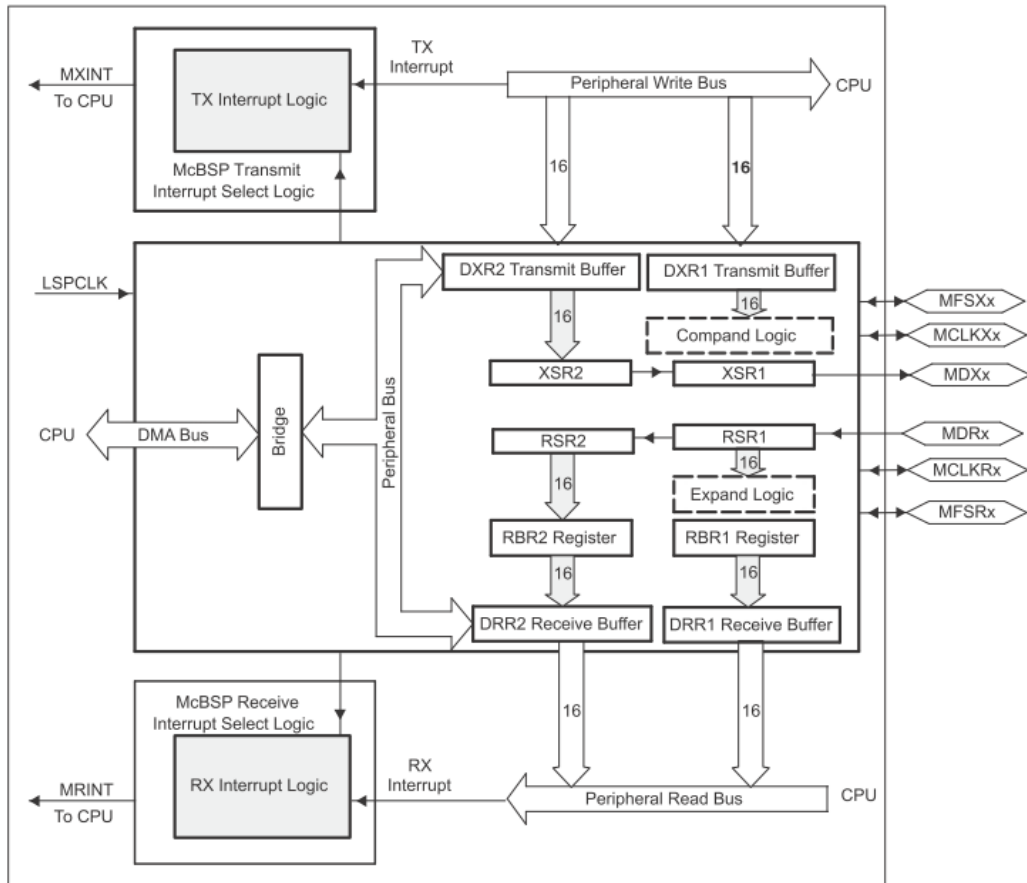
多通道缓冲串行端口(McBSP)模块

- McBSP模块有以下特性：
- 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
 - 串行外设接口(SPI)
- MCBSP 时钟速率：

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

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。

McBSP功能框图如下所示：



McBSP 模块

McBSP寄存器汇总见下表：

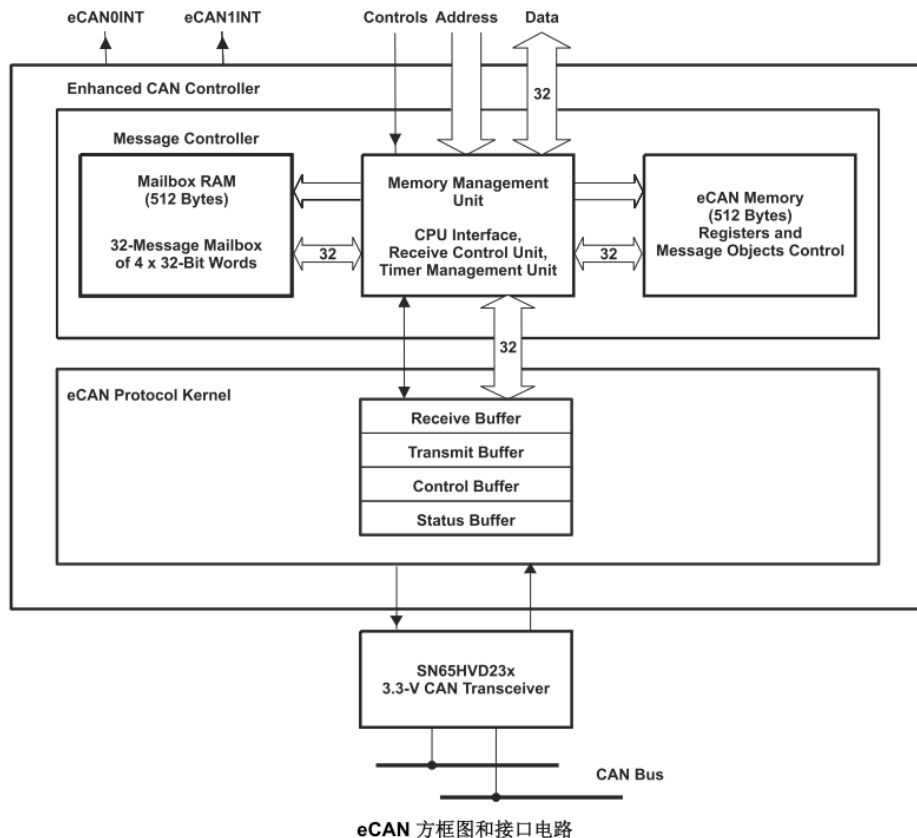
| 名称 | McBSP-A 地址 | McBSP-B 地址 | 类型 | 复位值 | 说明 |
|-------------|------------|------------|-----|--------|-----------------|
| 数据寄存器，接收、发送 | | | | | |
| DRR2 | 0x5000 | 0x5040 | 读 | 0x0000 | McBSP数据接收寄存器2 |
| DRR1 | 0x5001 | 0x5041 | 读 | 0x0000 | McBSP数据接收寄存器1 |
| DXR2 | 0x5002 | 0x5042 | 写 | 0x0000 | McBSP数据发送寄存器2 |
| DXR1 | 0x5003 | 0x5043 | 写 | 0x0000 | McBSP数据发送寄存器1 |
| McBSP 控制寄存器 | | | | | |
| SPCR2 | 0x5004 | 0x5044 | 读/写 | 0x0000 | McBSP串行端口控制寄存器2 |
| SPCR1 | 0x5005 | 0x5045 | 读/写 | 0x0000 | McBSP串行端口控制寄存器1 |
| RCR2 | 0x5006 | 0x5046 | 读/写 | 0x0000 | McBSP接收控制寄存器2 |
| RCR1 | 0x5007 | 0x5047 | 读/写 | 0x0000 | McBSP接收控制寄存器1 |
| XCR2 | 0x5008 | 0x5048 | 读/写 | 0x0000 | McBSP发送控制寄存器2 |
| XCR1 | 0x5009 | 0x5049 | 读/写 | 0x0000 | McBSP发送控制寄存器1 |
| SRGR2 | 0x500A | 0x504A | 读/写 | 0x0000 | McBSP采样率发生器寄存器2 |
| SRGR1 | 0x500B | 0x504B | 读/写 | 0x0000 | McBSP采样率发生器寄存器1 |

| 名称 | McBSP-A 地址 | McBSP-B 地址 | 类型 | 复位值 | 说明 |
|----------|------------|------------|-----|--------|-------------------|
| 多通道控制寄存器 | | | | | |
| MCR2 | 0x500C | 0x504C | 读/写 | 0x0000 | McBSP多通道寄存器2 |
| MCR1 | 0x500D | 0x504D | 读/写 | 0x0000 | McBSP多通道寄存器1 |
| RCERA | 0x500E | 0x504E | 读/写 | 0x0000 | McBSP接收通道使能寄存器分区A |
| RCERB | 0x500F | 0x504F | 读/写 | 0x0000 | McBSP接收通道使能寄存器分区B |
| XCERA | 0x5010 | 0x5050 | 读/写 | 0x0000 | McBSP发送通道使能寄存器分区A |
| XCERB | 0x5011 | 0x5051 | 读/写 | 0x0000 | McBSP发送通道使能寄存器分区B |
| PCR | 0x5012 | 0x5052 | 读/写 | 0x0000 | McBSP引脚控制寄存器 |
| RCERC | 0x5013 | 0x5053 | 读/写 | 0x0000 | McBSP接收通道使能寄存器分区C |
| RCERD | 0x5014 | 0x5054 | 读/写 | 0x0000 | McBSP接收通道使能寄存器分区D |
| XCERC | 0x5015 | 0x5055 | 读/写 | 0x0000 | McBSP发送通道使能寄存器分区C |
| XCERD | 0x5016 | 0x5056 | 读/写 | 0x0000 | McBSP发送通道使能寄存器分区D |
| RCERE | 0x5017 | 0x5057 | 读/写 | 0x0000 | McBSP接收通道使能寄存器分区E |
| RCERF | 0x5018 | 0x5058 | 读/写 | 0x0000 | McBSP接收通道使能寄存器分区F |
| XCERE | 0x5019 | 0x5059 | 读/写 | 0x0000 | McBSP发送通道使能寄存器分区E |
| XCERF | 0x501A | 0x505A | 读/写 | 0x0000 | McBSP发送通道使能寄存器分区F |
| RCERG | 0x501B | 0x505B | 读/写 | 0x0000 | McBSP接收通道使能寄存器分区G |
| RCERH | 0x501C | 0x505C | 读/写 | 0x0000 | McBSP接收通道使能寄存器分区H |
| XCERG | 0x501D | 0x505D | 读/写 | 0x0000 | McBSP发送通道使能寄存器分区G |
| XCERH | 0x501E | 0x505E | 读/写 | 0x0000 | McBSP发送通道使能寄存器分区H |
| MFFINT | 0x5023 | 0x5063 | 读/写 | 0x0000 | McBSP中断使能寄存器 |

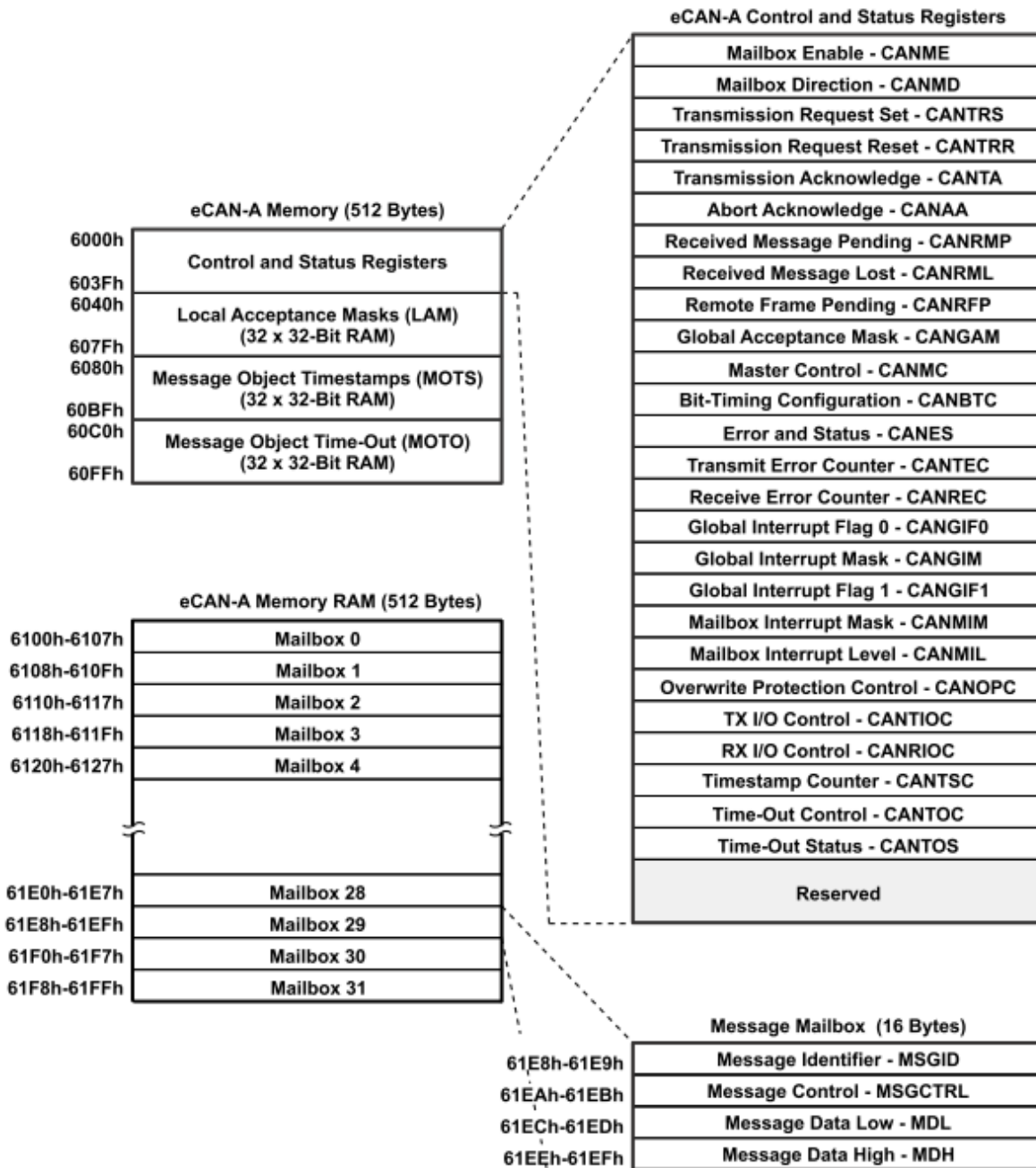
增强型控制器局域网(eCAN)模块 (eCAN-A和eCAN-B)

CAN 模块有下列特性：

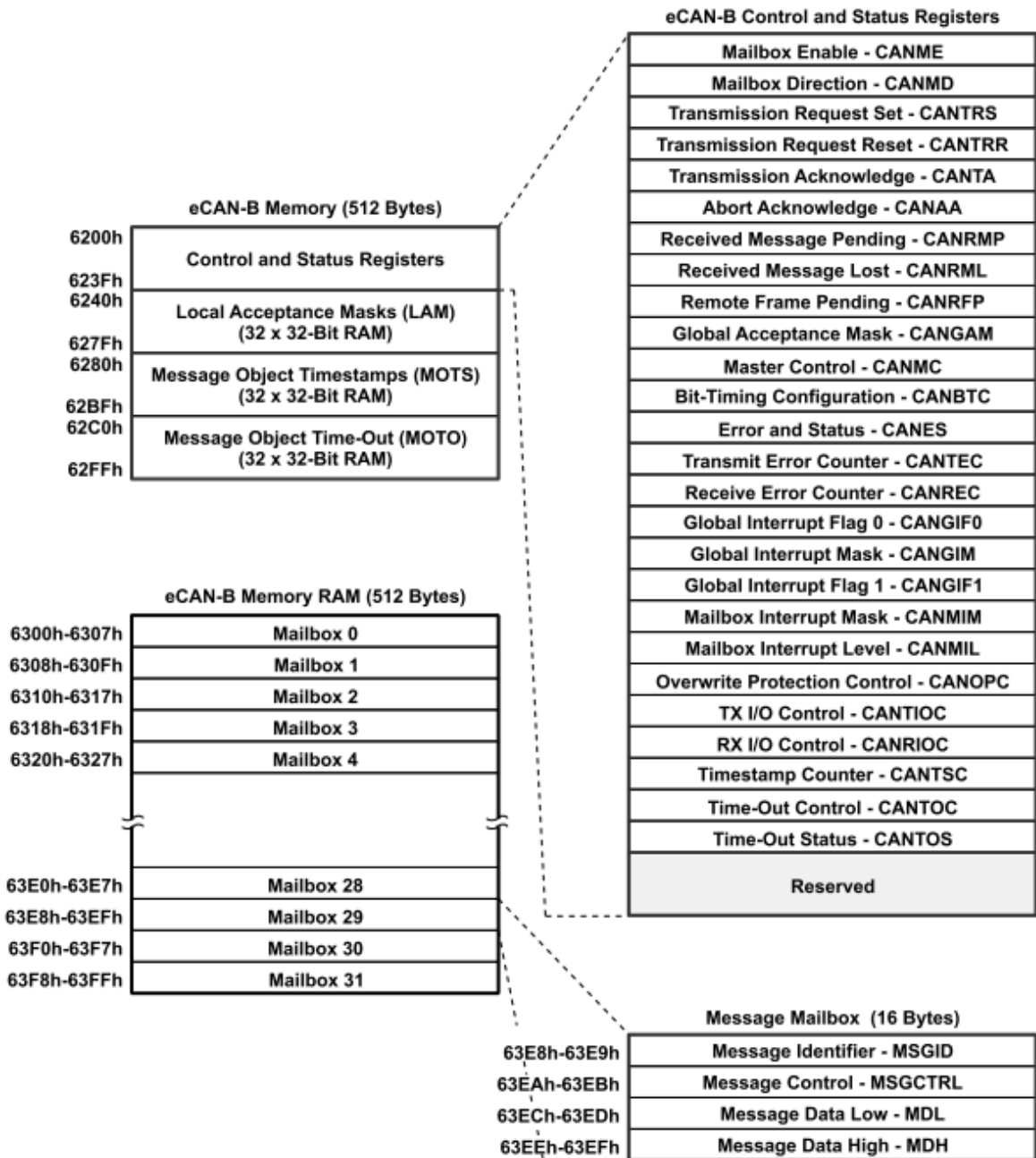
- 完全兼容ISO 11898-1 (CAN 2.0B)协议
- 支持高达 1Mbps 的数据速率
 - 可配置为接收或者发送
 - 可使用标准或者扩展标识符进行配置
 - 具有一个可编程接收掩码
 - 支持数据和远程帧
 - 由 0 至 8 字节数据组成
 - 在接收和发送消息上使用一个 32 位时间戳
 - 防止接收新消息
 - 保持发送消息的动态可编程优先级
 - 采用具有两个中断级别的可编程中断机制
 - 采用针对发送或接收超时的可编程警报
- 低功耗模式
- 总线活动上的可编程唤醒
- 对远程请求消息可自动答复
- 丢失仲裁帧或者错误情况下的帧自动重传
- 由一个特定消息同步的32位本地网络时间计数器 (与邮箱16协同通信)
- 自测模式
 - 运行在接收自身消息的回路模式。用来提供一个虚拟的确认，从而不需要由其他节点来提供确认位的需要。



eCAN 方框图和接口电路



eCAN-A 内存映射



eCAN-B 内存映射

CAN寄存器映射如下表所示：

| 名称 | eCAN-A地址 | eCAN-B地址 | 大小(x32) | 说明 |
|--------|----------|----------|---------|--------|
| CANME | 0x6000 | 0x6200 | 1 | 邮箱启用 |
| CANMD | 0x6002 | 0x6202 | 1 | 邮箱方向 |
| CANTRS | 0x6004 | 0x6204 | 1 | 发送请求设定 |
| CANTRR | 0x6006 | 0x6206 | 1 | 发送请求复位 |
| CANTA | 0x6008 | 0x6208 | 1 | 传输确认 |
| CANAA | 0x600A | 0x620A | 1 | 中止确认 |

| 名称 | eCAN-A地址 | eCAN-B地址 | 大小(x32) | 说明 |
|---------|----------|----------|---------|----------------------|
| CANRMP | 0x600C | 0x620C | 1 | 接收消息等待 |
| CANRML | 0x600E | 0x620E | 1 | 接收消息丢失 |
| CANRFP | 0x6010 | 0x6210 | 1 | 远程帧等待 |
| CANGAM | 0x6012 | 0x6212 | 1 | 全局接收屏蔽 |
| CANMC | 0x6014 | 0x6214 | 1 | 主器件控制 |
| CANBTC | 0x6016 | 0x6216 | 1 | 位时序配置 |
| CANES | 0x6018 | 0x6218 | 1 | 错误和状态 |
| CANTEC | 0x601A | 0x621A | 1 | 发送错误计数器 |
| CANREC | 0x601C | 0x621C | 1 | 接收错误计数器 |
| CANGIF0 | 0x601E | 0x621E | 1 | 全局中断标志0 |
| CANGIM | 0x6020 | 0x6220 | 1 | 全局中断屏蔽 |
| CANGIF1 | 0x6022 | 0x6222 | 1 | 全局中断标志1 |
| CANMIM | 0x6024 | 0x6224 | 1 | 邮箱中断屏蔽 |
| CANMIL | 0x6026 | 0x6226 | 1 | 邮箱中断级别 |
| CANOPC | 0x6028 | 0x6228 | 1 | 写覆盖保护控制 |
| CANTIOC | 0x602A | 0x622A | 1 | TX I/O 控制 |
| CANRIOC | 0x602C | 0x622C | 1 | RX I/O 控制 |
| CANTSC | 0x602E | 0x622E | 1 | 时间戳计数器 (保留在 SCC 模式中) |
| CANTOC | 0x6030 | 0x6230 | 1 | 超时控制 (保留在 SCC 模式中) |
| CANTOS | 0x6032 | 0x6232 | 1 | 超时状态 (保留在 SCC 模式中) |

串行通信接口(SCI)模块(SCI-A, SCI-B, SCI-C)

器件包括三个异步串行通信接口(SCI)模块。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：如果不用于SCI，则两个引脚都可以用作GPIO

- 波特率被设定为64000种不同速率：

$$\text{Baud rate} = \frac{\text{LSPCLK}}{(\text{BRR} + 1) * 8} \quad \text{when BRR} \neq 0$$

$$\text{Baud rate} = \frac{\text{LSPCLK}}{16} \quad \text{when BRR} = 0$$

- 数据字格式
 - 一个开始位
 - 数据-字长度可被设定为1至8位
 - 可选偶/奇/无奇偶校验位
 - 一个或者两个停止位
- 四个错误检测标志：奇偶、超载、组帧和中断检测
- 两个唤醒多处理器模式：空闲线路和地址位
- 半双工或者全双工运行
- 双缓冲接收和发送功能。
- 可通过带有状态标志的中断驱动或者轮询算法来完成发射器和接收器操作。
 - 发送器：TXRDY标志（发射器缓冲寄存器已经准备好接收另外字符）和TXEMPTY（TX空）标志（发送器移位寄存器已空）
 - 接收器：RXRDY标志（接收器缓冲寄存器已经准备好接收另外的字符），BRKDT标志（发生了中断条件）和RXERROR错误标志（监控四个中断条件）
- 用于发送器和接收器中断的独立使能位（除了BRKDT）
- NRZ（非归零）码格式

增强型特性：

- 自动波特率检测硬件逻辑电路
- 16级发送/接收FIFO

SCI-A寄存器配置和控制如下表所示：

| 名称 | 地址 | 大小 (x16) | 是否受EALLOW保护 | 说明 |
|-----------|----------|-------------|-------------|----------------|
| SCICCRRA | 0x007050 | 1 | 否 | SCI-A通信控制寄存器 |
| SCICTL1A | 0x007051 | 1 | 否 | SCI-A控制寄存器 |
| SCIHBAUDA | 0x007052 | 1 | 否 | SCI-A波特率寄存器，高位 |

| | | | | |
|-----------|----------|---|---|------------------|
| SCILBAUDA | 0x007053 | 1 | 否 | SCI-A波特率寄存器，低位 |
| SCICTL2A | 0x007054 | 1 | 否 | SCI-A控制寄存器2 |
| SCIRXSTA | 0x007055 | 1 | 否 | SCI-A接收状态寄存器 |
| SCIRXEMUA | 0x007056 | 1 | 否 | SCI-A接收仿真数据缓冲寄存器 |
| SCIRXBUFA | 0x007057 | 1 | 否 | SCI-A接收数据缓冲寄存器 |
| SCITXBUFA | 0x007059 | 1 | 否 | SCI-A发送数据缓冲寄存器 |
| SCIFFTXA | 0x00705A | 1 | 否 | SCI-A FIFO发送寄存器 |
| SCIFFRXA | 0x00705B | 1 | 否 | SCI-A FIFO接收寄存器 |
| SCIFFCTA | 0x00705C | 1 | 否 | SCI-A FIFO控制寄存器 |
| SCIPRIA | 0x00705F | 1 | 否 | SCI-A优先级控制寄存器 |

注：

- 1) 这个表中的寄存器被映射到外设帧 2 空间。这空间只允许 16 位访问。32 位访问会产生未定义的后果。
- 2) 这些寄存器是用于 FIFO 模式的全新寄存器。

SCI-B寄存器配置和控制如下表所示：

| 名称 | 地址 | 大小 (x16) | 是否受EBLLOW保护 | 说明 |
|-----------|----------|----------|-------------|------------------|
| SCICCRB | 0x007750 | 1 | 否 | SCI-B通信控制寄存器 |
| SCICTL1B | 0x007751 | 1 | 否 | SCI-B控制寄存器 |
| SCIHBBUDB | 0x007752 | 1 | 否 | SCI-B波特率寄存器，高位 |
| SCILBBUDB | 0x007753 | 1 | 否 | SCI-B波特率寄存器，低位 |
| SCICTL2B | 0x007754 | 1 | 否 | SCI-B控制寄存器2 |
| SCIRXSTB | 0x007755 | 1 | 否 | SCI-B接收状态寄存器 |
| SCIRXEMUB | 0x007756 | 1 | 否 | SCI-B接收仿真数据缓冲寄存器 |
| SCIRXBUFB | 0x007757 | 1 | 否 | SCI-B接收数据缓冲寄存器 |
| SCITXBUFB | 0x007759 | 1 | 否 | SCI-B发送数据缓冲寄存器 |
| SCIFFTXB | 0x00775B | 1 | 否 | SCI-B FIFO发送寄存器 |
| SCIFFRXB | 0x00775B | 1 | 否 | SCI-B FIFO接收寄存器 |

| | | | | |
|----------|----------|---|---|-----------------|
| SCIFFCTB | 0x00775C | 1 | 否 | SCI-B FIFO控制寄存器 |
| SCIPRIB | 0x00775F | 1 | 否 | SCI-B优先级控制寄存器 |

注：

- 1) 这个表中的寄存器被映射到外设帧 2 空间。这空间只允许 16 位访问。32 位访问会产生未定义的后果。
- 2) 这些寄存器是用于 FIFO 模式的全新寄存器。

SCI-C寄存器配置和控制如下表所示：

| 名称 | 地址 | 大小 (x16) | 是否受EBLLOW保护 | 说明 |
|-----------|----------|----------|-------------|------------------|
| SCICCRC | 0x007770 | 1 | 否 | SCI-C通信控制寄存器 |
| SCICTL1C | 0x007771 | 1 | 否 | SCI-C控制寄存器 |
| SCIHBBUDC | 0x007772 | 1 | 否 | SCI-C波特率寄存器，高位 |
| SCILBBUDC | 0x007773 | 1 | 否 | SCI-C波特率寄存器，低位 |
| SCICTL2C | 0x007774 | 1 | 否 | SCI-C控制寄存器2 |
| SCIRXSTC | 0x007775 | 1 | 否 | SCI-C接收状态寄存器 |
| SCIRXEMUC | 0x007776 | 1 | 否 | SCI-C接收仿真数据缓冲寄存器 |
| SCIRXBUFC | 0x007777 | 1 | 否 | SCI-C接收数据缓冲寄存器 |
| SCITXBUFC | 0x007779 | 1 | 否 | SCI-C发送数据缓冲寄存器 |
| SCIFFTXC | 0x00777B | 1 | 否 | SCI-C FIFO发送寄存器 |
| SCIFFRXC | 0x00777B | 1 | 否 | SCI-C FIFO接收寄存器 |
| SCIFFCTC | 0x00777C | 1 | 否 | SCI-C FIFO控制寄存器 |
| SCIPRIC | 0x00777F | 1 | 否 | SCI-C优先级控制寄存器 |

注：

- 1) 这个表中的寄存器被映射到外设帧 2 空间。这空间只允许 16 位访问。32 位访问会产生未定义的后果。
- 2) 这些寄存器是用于 FIFO 模式的全新寄存器。

串行外设接口(SPI)模块 (SPI-A)

器件包括四引脚的串行通信接口(SPI) 模块。一个SPI模块 (SPI-A) 可用。SPI 是一个高速、同步串行I/O 端口，此端口可在设定的位传输速率上将一个设定长度 (1 至16 位) 的串行比特流移入和移出器件。通常，SPI 用于DSP 和外部外设或者其它处理器之间的通信。典型应用包括外部I/O 或者从诸如移位寄存器、显示驱动器和ADC 等器件的外设扩展。多器件通信由SPI 的主控/受控操作支持。

- SPI 模块的特性包括：
- 四个外部引脚：

- SPISOMI : SPI 从器件输出/主器件输入引脚
 - SPISIMO : SPI 从器件输入/主器件输出引脚
 - SPISTEn : SPI 从器件发送使能引脚
 - SPICLK : SPI 串行时钟引脚
- 两个运行模式：主控和受控
 - 支持多种波特率：125个不同的可编辑速率

$$\text{Baud rate} = \frac{\text{LSPCLK}}{(\text{SPIBRR} + 1)} \quad \text{when SPIBRR} = 3 \text{ to } 127$$

$$\text{Baud rate} = \frac{\text{LSPCLK}}{4} \quad \text{when SPIBRR} = 0, 1, 2$$

- 数据字长度：1至16数据位
- 包括4种时钟模式（由时钟极性和时钟相位的位控制）：
 - 无相位延迟的下降沿：SPICLK高电平有效。SPI在SPICLK信号的下降沿上发送数据，在SPICLK信号的上升沿上接收数据。
 - 有相位延迟的下降沿：SPICLK高电平有效。SPI在SPICLK信号下降沿提前半个周期发送数据，在SPICLK信号的下降沿上接收数据。
 - 无相位延迟的上升沿：SPICLK低电平无效。SPI在SPICLK信号的上升沿上发送数据，在SPICLK信号的下降沿上接收数据。
 - 有相位延迟的上升沿：SPICLK低电平无效。SPI在SPICLK信号上升沿的半个周期之前发送数据，而在SPICLK信号的上升沿上接收数据。
- 同时接收和发送操作（发送功能可在软件中被禁用）
- 通过中断驱动或者轮询算法来完成发射器和接收器运行
- 9个SPI模块控制寄存器：位于控制寄存器内，帧开始地址7040h
- 增强型特性：
 - 16级发送/接收FIFO
 - 延迟发送控制

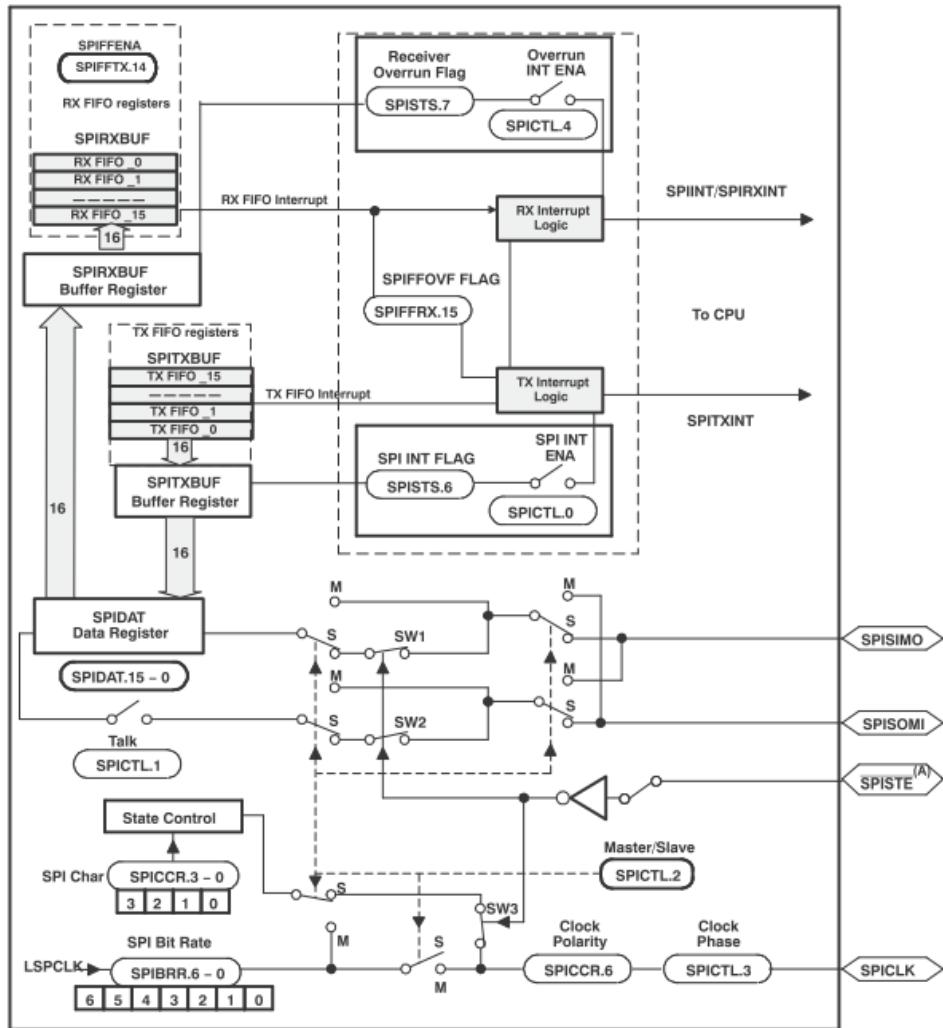
SPI 端口配置和控制寄存器如下表所示：

SPI-A 寄存器

| 名称 | 地址 | 大小 (x16) | 说明 |
|----------|--------|----------|-----------------|
| SPICCR | 0x7040 | 1 | SPI-A配置控制寄存器 |
| SPICTL | 0x7041 | 1 | SPI-A运行控制寄存器 |
| SPISTS | 0x7042 | 1 | SPI-A状态寄存器 |
| SPIBRR | 0x7044 | 1 | SPI-A波特率寄存器 |
| SPIRXEMU | 0x7046 | 1 | SPI-A接收仿真缓冲器寄存器 |
| SPIRXBUF | 0x7047 | 1 | SPI-A串行输入缓冲器寄存器 |
| SPITXBUF | 0x7048 | 1 | SPI-A串行输出缓冲器寄存器 |

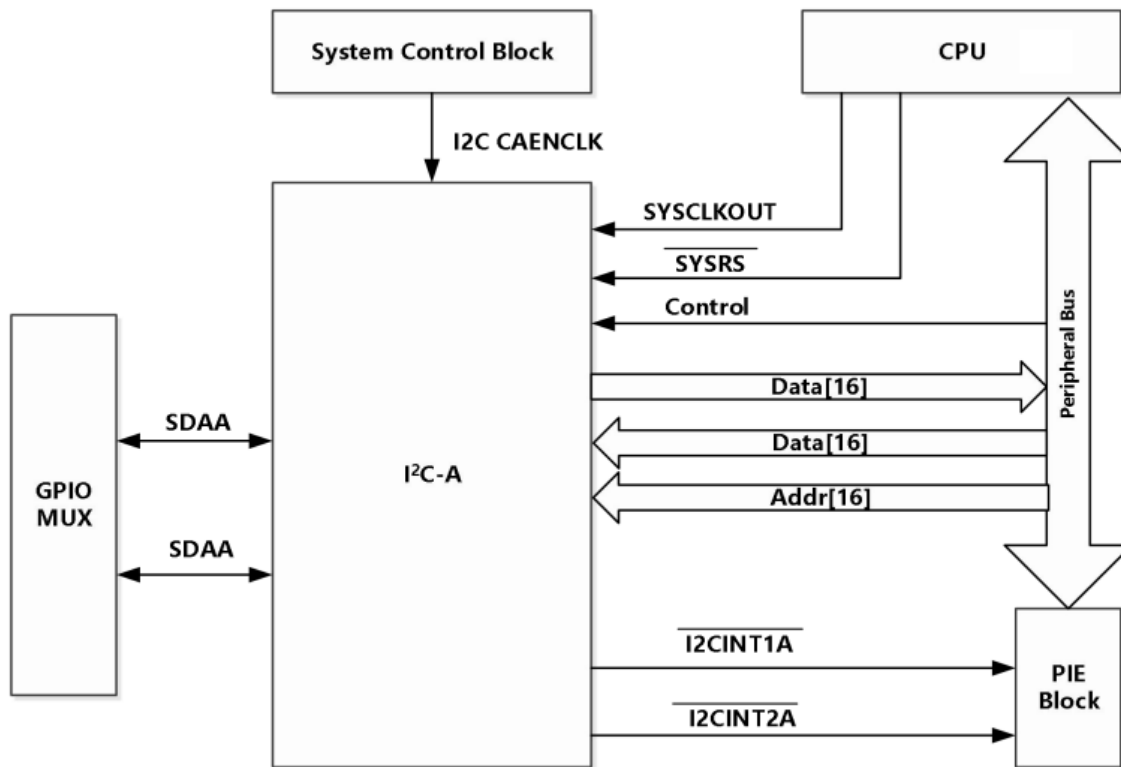
| | | | |
|---------|--------|---|-----------------|
| SPIDAT | 0x7049 | 1 | SPI-A串行数据寄存器 |
| SPIFFTX | 0x704A | 1 | SCI-A_FIFO发送寄存器 |
| SPIFFRX | 0x704B | 1 | SCI-A_FIFO接收寄存器 |
| SPIFFCT | 0x704C | 1 | SCI-A_FIFO控制寄存器 |
| SPIPRI | 0x704F | 1 | SCI-A优先级控制寄存器 |

下图为处于受控模式下的SPI的功能框图：



内部集成电路(I2C)

包含一个I2C串行端口。下图显示了此器件内的 I2C 外设模块接口。



I2C 外设模块接口

I2C 模块具有以下特性：

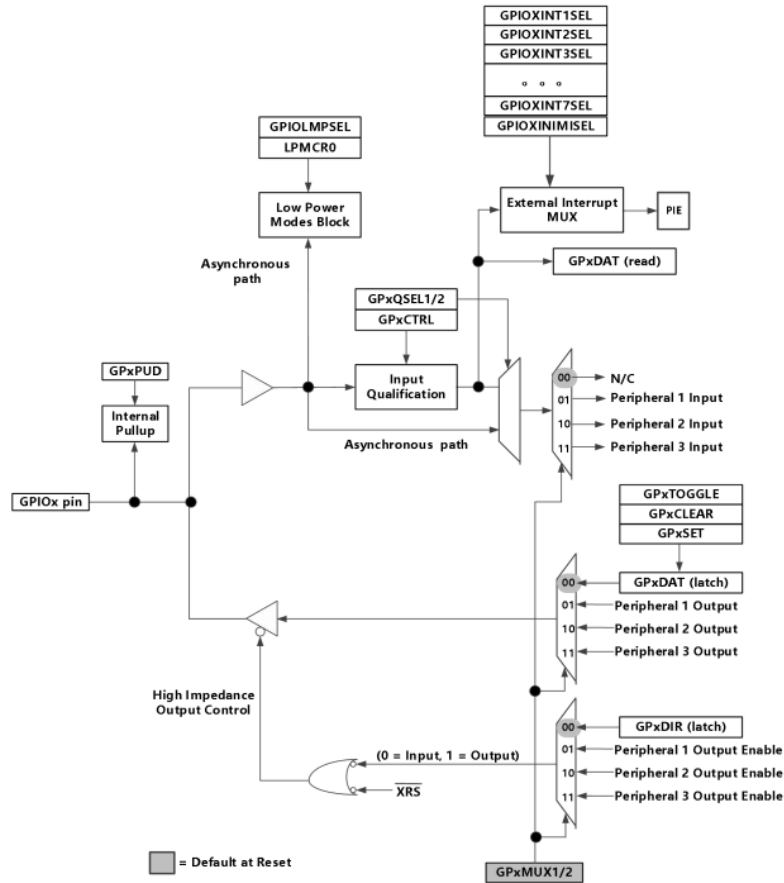
- 符合飞利浦半导体I2C总线规格（版本2.1）：
 - 支持1位至8位格式传输
 - 7 位和 10 位寻址模式
 - 常规调用
 - START 字节模式
 - 支持多个主发送器和从接收器
 - 支持多个从发送器和主接收器
 - 组合主器件发送/接收和接收/发送模式
 - 数据传输速率从 10kbps 到高达 400kbps (I2C 快速模式速率)
- 一个16字接收FIFO和一个16字发送FIFO
- 可以由CPU使用的一个中断。这个中断可由下列条件中的一个生成：
 - 发送数据准备好
 - 接收数据准备好
 - 寄存器访问准备好
 - 没有接收到确认
 - 仲裁丢失
 - 检测到停止条件
 - 被寻址为从器件
- 在FIFO模式下，CPU可以使用附加的中断
- 模块启用/禁用能力
- 自由数据格式模式

| I2C-A 寄存器 | | |
|-----------|--------|---------------------|
| 名称 | 地址 | 说明 |
| I2COAR | 0x7900 | I2C自身的地址寄存器 |
| I2CIER | 0x7901 | I2C中断使能寄存器 |
| I2CSTR | 0x7902 | I2C状态寄存器 |
| I2CCLKL | 0x7903 | I2C时钟低电平时间分频器寄存器 |
| I2CCLKH | 0x7904 | I2C时钟高电平时间分频器寄存器 |
| I2CCNT | 0x7905 | I2C数据计数寄存器 |
| I2CDRR | 0x7906 | I2C数据接收寄存器 |
| I2CSAR | 0x7907 | I2C从器件地址寄存器 |
| I2CDXR | 0x7908 | I2C数据发送寄存器 |
| I2CMDR | 0x7909 | I2C模式寄存器 |
| I2CISRC | 0x790A | I2C中断源寄存器 |
| I2CPSC | 0x790C | I2C预分频器寄存器 |
| I2CFFTX | 0x7920 | I2C_FIFO发送寄存器 |
| I2CFFRX | 0x7921 | I2C_FIFO接收寄存器 |
| I2CRSR | - | I2C接收移位寄存器（不可访问CPU） |
| I2CXSR | - | I2C发送移位寄存器（不可访问CPU） |

GPIOMUX

器件上除了提供独立的引脚位撕裂I/O功能外，GPIOMUX还可以将最多3个独立的外设信号复用在单一的一个GPIO引脚上。每个引脚的GPIOMUX功能框图显示在下图中。由于I2C引脚的开漏功能，这些引脚的GPIOMUX功能框图是不同的。

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



GPIO 复用

器件支持88个GPIO引脚。GPIO控制和数据寄存器被映射到外设帧1以在寄存器上启用32位运行（连同16位运行）。

GPIO寄存器映射如下表所示:

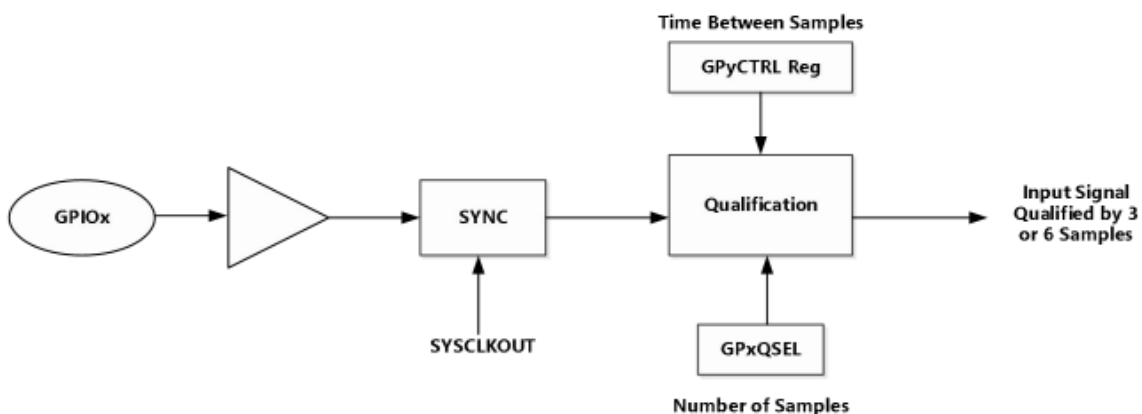
| 名称 | 地址 | 大小 (x16) | 说明 |
|-----------------------|---------------|----------|-------------------------------|
| GPIO控制寄存器 (受EALLOW保护) | | | |
| GPACTRL | 0x6F80 | 2 | GPIOA控制寄存器(GPIO0至31) |
| GPAQSEL1 | 0x6F82 | 2 | GPIOA限定器选择1寄存器(GPIO0至15) |
| GPAQSEL2 | 0x6F84 | 2 | GPIOA限定器选择2寄存器(GPIO16至31) |
| GPAMUX1 | 0x6F86 | 2 | GPIOAMUX1寄存器 (GPIO0至15) |
| GPAMUX2 | 0x6F88 | 2 | GPIOAMUX2寄存器 (GPIO16至31) |
| GPADIR | 0x6F8A | 2 | GPIOA方向寄存器(GPIO0至31) |
| GPAPUD | 0x6F8C | 2 | GPIOA上拉电阻器禁用寄存器(GPIO0至GPIO31) |
| 被保留 | 0x6F8E-0x6F8F | 2 | |
| GPBCTRL | 0x6F90 | 2 | GPIOB控制寄存器(GPIO32至63) |
| GPBQSEL1 | 0x6F92 | 2 | GPIOB限定器选择1寄存器(GPIO32至47) |
| GPBQSEL2 | 0x6F94 | 2 | GPIOB限定器选择2寄存器(GPIO48至63) |

| 名称 | 地址 | 大小 (x16) | 说明 |
|-------------------------------|-------------------|----------|--------------------------------|
| GPBMUX1 | 0x6F96 | 2 | GPIOBMUX1寄存器(GPIO32至47) |
| GPBMUX2 | 0x6F98 | 2 | GPIOBMUX2寄存器(GPIO48至63) |
| GPBDIR | 0x6F9A | 2 | GPIOB方向寄存器(GPIO32至44) |
| GPBPUD | 0x6F9C | 2 | GPIOB上拉电阻器禁用寄存器(GPIO38至44) |
| 被保留 | 0x6F9E-0x6FA5 | 8 | |
| GPCMUX1 | 0x6FA6 | 2 | GPIOCMUX1寄存器(GPIO64至79) |
| GPCMUX2 | 0x6FA8 | 2 | GPIOCMUX2寄存器(GPIO80至87) |
| GPCDIR | 0x6FAA | 2 | GPIOC方向寄存器(GPIO64至87) |
| GPCPUD | 0x6FAC | 2 | GPIOC上拉电阻器禁用寄存器(GPIO64至87) |
| 被保留 | 0x6FAE-0x6FBF | 18 | |
| GPIO数据寄存器 (不受EALLOW保护) | | | |
| GPADAT | 0x6FC0 | 2 | GPIOA数据寄存器 (GPIO0至31) |
| GPASET | 0x6FC2 | 2 | GPIOA数据设定寄存器 (GPIO0至31) |
| GPACLEAR | 0x6FC4 | 2 | GPIOA数据清除寄存器 (GPIO0至31) |
| GPATOGGLE | 0x6FC6 | 2 | GPIOA数据取反寄存器 (GPIO0至31) |
| GPBDAT | 0x6FC8 | 2 | GPIOB数据寄存器(GPIO32至63) |
| GPBSET | 0x6FCA | 2 | GPIOB数据设定寄存器 (GPIO32至63) |
| GPBCLEAR | 0x6FCC | 2 | GPIOB数据清除寄存器 (GPIO32至63) |
| GPBTOGGLE | 0x6FCE | 2 | GPIOB数据取反寄存器 (GPIO32至63) |
| GPCDAT | 0x6FD0 | 2 | GPIOC数据寄存器(GPIO64至87) |
| GPCSET | 0x6FD2 | 2 | GPIOC数据设定寄存器 (GPIO64至87) |
| GPCCLEAR | 0x6FD4 | 2 | GPIOC数据清除寄存器 (GPIO64至87) |
| GPCTOGGLE | 0x6FD6 | 2 | GPIOC数据取反寄存器 (GPIO64至87) |
| 被保留 | 0x6FD8-0x6FD F | 8 | |
| GPIO中断和低功耗模式选择寄存器 (受EALLOW保护) | | | |
| GPIOXINT1SE L | 0x6FE0 | 1 | XINT1GPIO输入选择寄存器(GPIO0至31) |
| GPIOXINT2SE L | 0x6FE1 | 1 | XINT2GPIO输入选择寄存器(GPIO0至GPIO31) |

| 名称 | 地址 | 大小 (x16) | 说明 |
|------------------|---------------|----------|---------------------------------|
| GPIOXNIMISE L | 0x6FE2 | 1 | XNMIGPIO输入选择寄存器(GPIO0至GPIO31) |
| GPIOXINT3SE L | 0x6FE3 | 1 | XINT3GPIO输入选择寄存器(GPIO32至GPIO63) |
| GPIOXINT4SE L | 0x6FE4 | 1 | XINT4GPIO输入选择寄存器(GPIO32至GPIO63) |
| GPIOXINT5SE L | 0x6FE5 | 1 | XINT5GPIO输入选择寄存器(GPIO32至GPIO63) |
| GPIOXINT6SE L | 0x6FE6 | 1 | XINT6GPIO输入选择寄存器(GPIO32至GPIO63) |
| GPIOXINT7SE L | 0x6FE7 | 1 | XINT7GPIO输入选择寄存器(GPIO32至GPIO63) |
| GPIOLPMSEL | 0x6FE8 | 2 | LPMGPIO选择寄存器(GPIO0至GPIO32) |
| 被保留 | 0x6FEA-0x6FFF | 22 | |

通过配置 GPxQSEL1/2 寄存器，用户可为每一个 GPIO 引脚选择输入限定的类型：

- 只同步至 SYSCLKOUT (GPxQSEL1/2=0,0)：这是复位时所有 GPIO 引脚的默认模式并且它只是将输入信号同步至系统时钟 (SYSCLKOUT)。
- 使用采样窗口的限定条件 (GPxQSEL1/2=0, 1 和 1, 0)：这个模式中，在与系统时钟(SYSCLKOUT)同步后，输入信号在输入被允许改变前，被一定数量的周期所限定。
- 采样周期由 GPxCTRL 寄存器内的 QUALPRD 位所指定并且可在每组 8 个信号中进行配置。它为输入信号采样指定了多个 SYSCLKOUT 周期。一个采样窗口为可包含 3 次或 6 次采样，并且只有当所有采样与下图（对于 6 次采样）中所显示的一样时（全 0 或者全 1），输出才会改变。
- 无同步 (GPxQSEL1/2=1, 1)：这个模式用于无需同步的外设（同步不在外设内执行）。由于器件上所要求的多级复用，有可能会有一个外设输入信号被映射到多于一个 GPIO 引脚的情况。此外，当一个输入信号未被选择时，输入信号将默认为一个 0 或者 1 状态，这由外设而定。

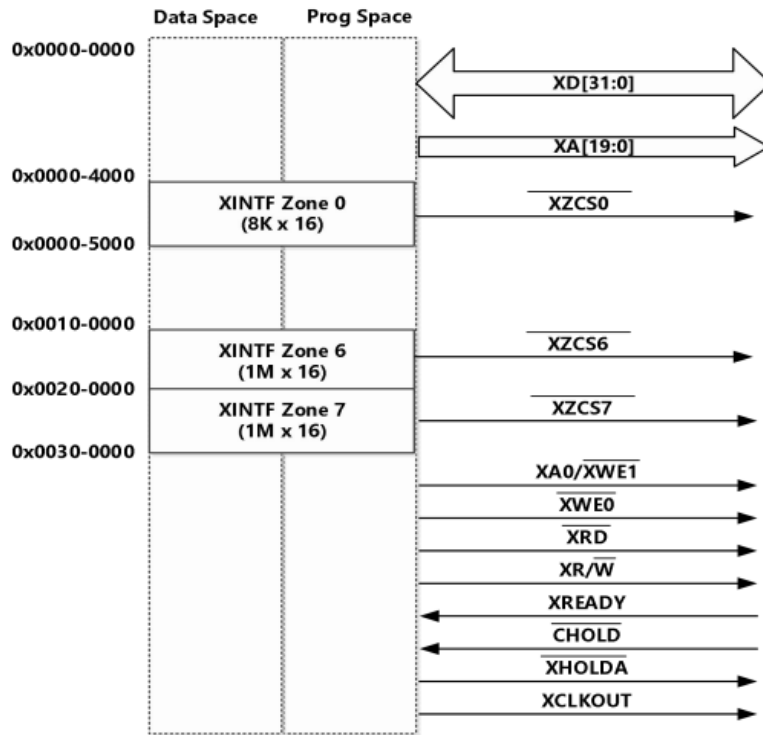


使用采样窗口的限定

外部接口(XINTF)

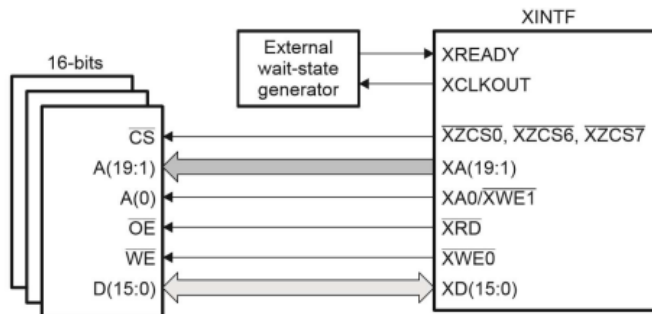
下图为器件上执行的外部接口 (XINTF) 的功能框图。

XINTF是一个非复用的异步总线，XINTF被映射到下图所示的三个固定区域。每个区域可被设定为具有不同的等待状态、建立和保持时序，并且当执行到一个特定区域的访问时，由区域芯片选择切换支持。这些特性可实现到很多外部存储器 and 外设的无缝连接。

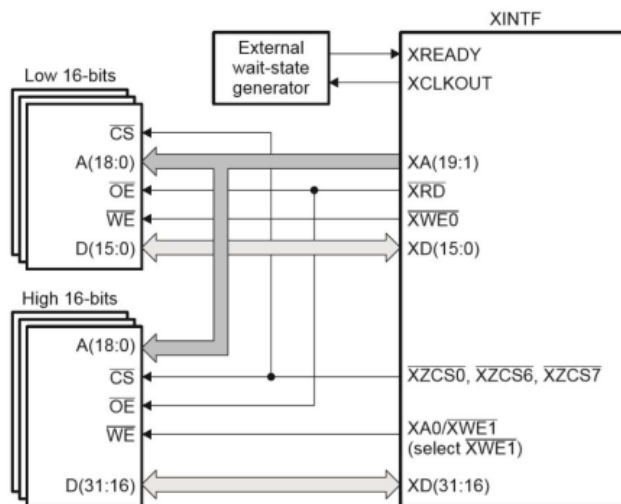


外部接口方框图

典型16位和32位数据总线XINTF连接如下图所示，下图说明了XA0和XWE1的信号功能是如何根据具体的配置变化的。



典型的 16 位数据总线 XINTF 连接



典型的 32 位数据总线 XINTF 连接

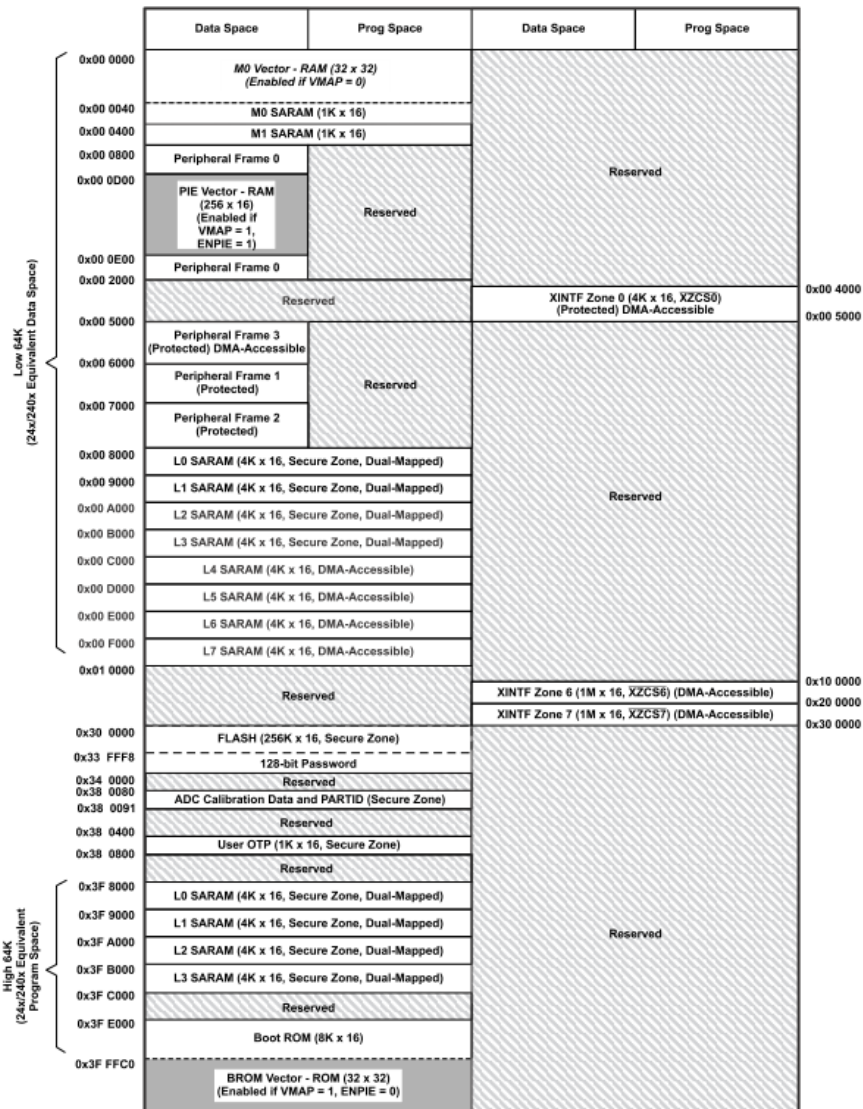
下表定义了 XINTF 配置和控制寄存器：

| XINTF 配置和控制寄存器映射 | | | |
|------------------|-----------|----------|-----------------|
| 名称 | 地址 | 大小 (x16) | 说明 |
| XTIMING 0 | 0x00-0B20 | 2 | XINTF定时寄存器, 区域0 |
| XTIMING 6 | 0x00-0B2C | 2 | XINTF定时寄存器, 区域6 |
| XTIMING 7 | 0x00-0B2E | 2 | XINTF定时寄存器, 区域7 |
| XINTCNF 2 | 0x00-0B34 | 2 | XINTF配置寄存器 |
| XBANK | 0x00-0B38 | 1 | XINTF组控制寄存器 |
| XREVISI ON | 0x00-0B3A | 1 | XINTF修订版本寄存器 |
| XRESET | 0x00-0B3D | 1 | XINTF复位寄存器 |

内存映射

器件内存情况如下：

- 内存块不可扩展
- 外设帧0，外设帧1，外设帧2和外设帧3内存映射只限于数据内存。用户程序不能访问这些处于程序空间内的内存映射。
- 受保护意味着“写后读操作”的顺序被保存，而不是保存流水线顺序。
- 特定内存区域受EALLOW保护以防止配置之后的假写入。
- 位置0x380080-0x38008F包含ADC校准程序。它不由用户编程。
- 如果eCAN模块未在应用中使用，提供的RAM（LAM，MOTS，MOTO和邮箱RAM）可被用作通用RAM。为实现这一功能CAN模块时钟应被启用。



注：一次只能启用这些向量映射中的一个：M0向量、PIE向量、BROM向量

闪存扇区的地址情况如下所示：

| 地址范围 | 程序和数据空间 |
|-------------------|-------------|
| 0x300000-0x307FFF | 扇区H(32Kx16) |

| | |
|-------------------|----------------------|
| 0x308000-0x30FFFF | 扇区G(32Kx16) |
| 0x310000-0x317FFF | 扇区F(32Kx16) |
| 0x318000-0x31FFFF | 扇区E(32Kx16) |
| 0x320000-0x327FFF | 扇区D(32Kx16) |
| 0x328000-0x32FFFF | 扇区C(32Kx16) |
| 0x330000-0x337FFF | 扇区B(32Kx16) |
| 0x338000-0x33FF7F | 扇区A(32Kx16) |
| 0x33FF80-0x33FFF5 | 当使用代码安全模块时，编程至0x0000 |
| 0x33FFF6-0x33FFF7 | 引导至闪存进入点（程序分支指令） |
| 0x33FFF8-0x33FFFF | 安全密码（128位）（不要设定为全零） |

注意：

- 当代码安全密码被编辑时，0x33FF80到0x33FFF5间的所有地址不能被用作程序代码或者数据。这些位置必须被设定为0x0000。
- 如果代码安全特性未被使用，地址0x33FF80至0x33FFEF可被用于代码或者数据。地址0x33FF0-0x33FF5为数据保留且不能包含程序代码。

处理安全代码位置如下所示：

| 地址 | 闪存 | |
|-------------------|-------------|---------|
| | 代码安全启用 | 代码安全禁用 |
| 0x33FF80-0x33FFE0 | 用 0x0000 填充 | 应用代码和数据 |
| 0x33FFF0-0x33FFF5 | | 只为数据保留 |

外设帧1，外设帧2以及XINTF区域1为一组，成为“受保护的写入/读取外设块”。“受保护”模式确保对这些所有的访问与文档中描述的一致。由于器件的流水线，在对不同内存位置读取之前的写入操作将以倒序的方式出现在CPU内存总线上。这会导致特定外设应用中用户认为写入会首先发生（如文档所描述的那样）。CPU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进行（代价是会增加额外的周期以校正运行）。可对这个模式进行编程，并且默认情况下，它将保护所选的区域。

寄存器映射

此器件包含四个外设寄存器空间。这些空间分类如下：

外设帧0：此类寄存器是直接映射到 CPU 内存总线的外设。

外设帧1：此类寄存器是映射到 32 位外设总线的外设。

外设帧2：此类寄存器是映射到 16 位外设总线的外设。

外设帧3：此类寄存器是映射到 32 位外设总线并可由 DMA 访问的外设。

| 外设帧0寄存器 | | | |
|---------|------|----------|------|
| 名称 | 地址范围 | 大小 (X16) | 访问类型 |
| | | | |

| | | | |
|--|-------------------|-----|------------|
| 器件仿真寄存器 | 0x000880-0x0009FF | 384 | 受EALLOW保护 |
| 闪存寄存器 | 0x000A80-0x000ADF | 96 | 受EALLOW保护 |
| 代码安全模块寄存器 | 0x000AE0-0x000AEF | 16 | 受EALLOW保护 |
| ADC寄存器 (双映射) 0等待(DMA), 1等待(CPU), 只读 | 0x000B00-0x000B0F | 16 | 不受EALLOW保护 |
| XINTF寄存器 | 0x000B20-0x000B3F | 32 | 受EALLOW保护 |
| CPU定时器0, CPU定时器1, CPU定时器2寄存器 | 0x000C00-0x000C3F | 64 | 不受EALLOW保护 |
| PIE寄存器 | 0x000CE0-0x000CFF | 32 | 不受EALLOW保护 |
| PIE矢量表 | 0x000D00-0x000DFF | 256 | 受EALLOW保护 |
| DMA寄存器 | 0x001000-0x0011FF | 512 | 受EALLOW保护 |

注：

- 1) 帧0中的寄存器支持16位和32位访问。
- 2) 如果寄存器是受EALLOW保护的，那么在EALLOW指令被执行前写入不能被执行。EDIS指令禁用写入以防止杂散代码或指针破坏寄存器内容。
- 3) 闪存寄存器也受到代码安全模块(CSM)的保护

| 外设帧 1 寄存器 | | |
|-----------------|-------------------|----------|
| 名称 | 地址范围 | 大小 (X16) |
| eCAN-A寄存器 | 0x006000-0x0061FF | 512 |
| eCAN-B寄存器 | 0x006200-0x0063FF | 512 |
| ePWM1+HRPWM1寄存器 | 0x006800-0x00683F | 64 |
| ePWM2+HRPWM2寄存器 | 0x006840-0x00687F | 64 |
| ePWM3+HRPWM3寄存器 | 0x006880-0x0068BF | 64 |
| ePWM4+HRPWM4寄存器 | 0x0068C0-0x0068FF | 64 |
| ePWM5+HRPWM5寄存器 | 0x006900-0x00693F | 64 |
| ePWM6+HRPWM6寄存器 | 0x006940-0x00697F | 64 |
| eCAP1寄存器 | 0x006A00-0x006A1F | 32 |
| eCAP2寄存器 | 0x006A20-0x006A3F | 32 |
| eCAP3寄存器 | 0x006A40-0x006A5F | 32 |
| eCAP4寄存器 | 0x006A60-0x006A7F | 32 |
| eCAP5寄存器 | 0x006A80-0x006A9F | 32 |
| eCAP6寄存器 | 0x006AA0-0x006ABF | 32 |
| eQEP1寄存器 | 0x006B00-0x006B3F | 64 |
| eQEP2寄存器 | 0x006B40-0x006B7F | 64 |
| GPIO寄存器 | 0x006F80-0x006FFF | 128 |

外设帧 2 寄存器

| 名称 | 地址范围 | 大小 (X16) |
|----------|-------------------|----------|
| 系统控制寄存器 | 0x007010-0x00702F | 32 |
| SPI-A寄存器 | 0x007040-0x00704F | 16 |
| SCI-A寄存器 | 0x007050-0x00705F | 16 |
| 外部中断寄存器 | 0x007070-0x00707F | 16 |
| ADC寄存器 | 0x007100-0x00711F | 32 |
| SCI-B寄存器 | 0x007750-0x00775F | 16 |
| SCI-C寄存器 | 0x007770-0x00777F | 16 |
| I2C-A寄存器 | 0x007900-0x00793F | 64 |

外设帧 3 寄存器

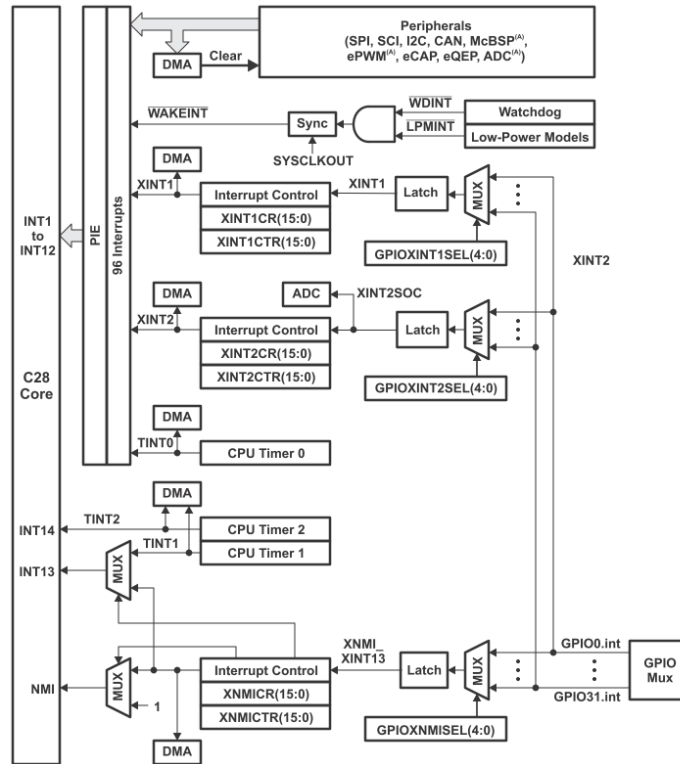
| 名称 | 地址范围 | 大小 (X16) |
|-------------------|---------------|----------|
| McBSP-A寄存器(DMA) | 0x5000-0x503F | 64 |
| McBSP-B寄存器(DMA) | 0x5040-0x507F | 64 |
| ePWM1+HRPWM1(DMA) | 0x5800-0x583F | 64 |
| ePWM2+HRPWM2(DMA) | 0x5840-0x587F | 64 |
| ePWM3+HRPWM3(DMA) | 0x5880-0x58BF | 64 |
| ePWM4+HRPWM4(DMA) | 0x58C0-0x58FF | 64 |
| ePWM5+HRPWM5(DMA) | 0x5900-0x593F | 64 |
| ePWM6+HRPWM6(DMA) | 0x5940-0x597F | 64 |

注：

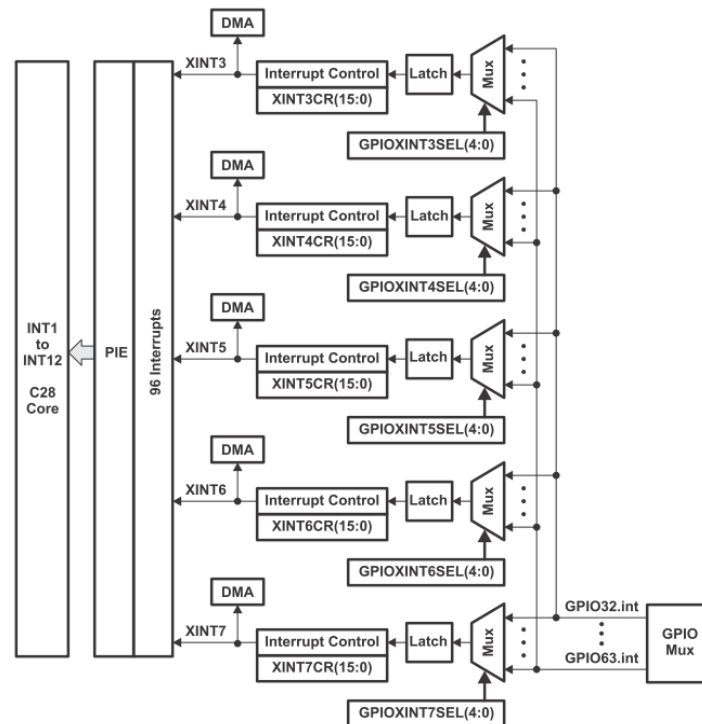
EPWM 和 HRPWM 模块可以被重新映射到可以被 DMA 模块访问的外设帧 3。要做到这点，MAPCNF 寄存器 (地址 0x702E) 的位 0 (MAPEPWM) 必须被设置为 1。此寄存器受 EALLOW 保护。当此位为 0 时，ePWM 和 HRPWM 模块被映射到外设帧 1。

中断

器件中断复用情况如下图所示：



外部和 PIE 中断源



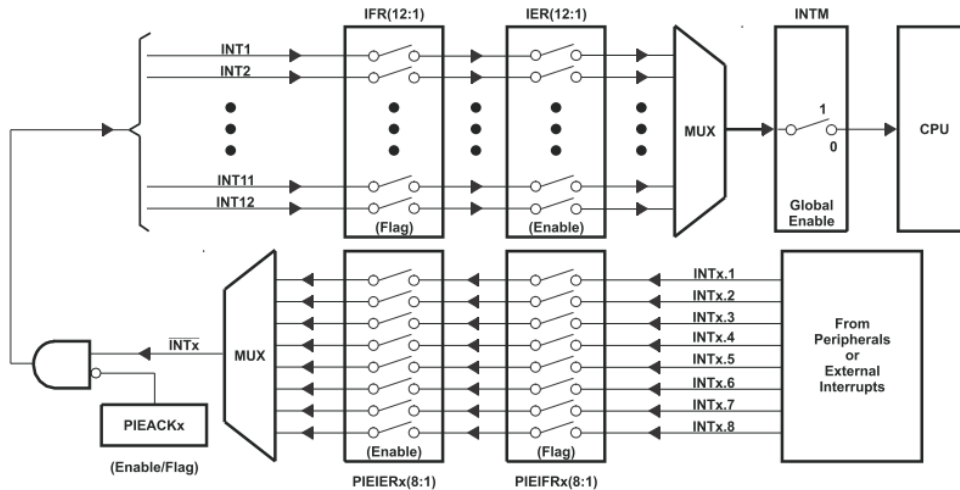
外部中断

8个PIE块中断为一个CPU中断组，总共12个CPU中断组，每组8个中断，共96个中断。

TRAP#Vectormumber(中断向量号)指令可控制器件触发指定中断向量，以及该中断向量相对应的中断处理例程。TRAP#0触发复位中断向量所指向的地址。然而，PIE矢量表不含复位向量。因此，当PIE被启用时，TRAP#0指令不应被使用。

当PIE启用时，TRAP#1至TRAP#12触发PIE中断向量表中对应中断向量。例如：TRAP#1触发INT1.1，

TRAP#2触发INT2.1，以此类推。



PIE块的中断复用

PIE 外设中断向量表

| CPU 中断 | PIE 中断 | | | | | | | |
|--------|--------------------|--------------------|---------------------|---------------------|---------------------|---------------------|---------------------|---------------------|
| | INTx.8 | INTx.7 | INTx.6 | INTx.5 | INTx.4 | INTx.3 | INTx.2 | INTx.1 |
| INT1 | WAKEINT (LPM/WD) | TINT0 (计时器 0) | ADCINT (ADC) | XINT2 | XINT1 | 保留 | SEQ2INT (ADC) | SEQ1INT (ADC) |
| INT2 | 保留 | 保留 | EPWM6_TZINT (ePWM6) | EPWM5_TZINT (ePWM5) | EPWM4_TZINT (ePWM4) | EPWM3_TZINT (ePWM3) | EPWM2_TZINT (ePWM2) | EPWM1_TZINT (ePWM1) |
| INT3 | 保留 | 保留 | EPWM6_INT (ePWM6) | EPWM5_INT (ePWM5) | EPWM4_INT (ePWM4) | EPWM3_INT (ePWM3) | EPWM2_INT (ePWM2) | EPWM1_INT (ePWM1) |
| INT4 | 保留 | 保留 | ECAP6_INT (eCAP6) | ECAP5_INT (eCAP5) | ECAP4_INT (eCAP4) | ECAP3_INT (eCAP3) | ECAP2_INT (eCAP2) | ECAP1_INT (eCAP1) |
| INT5 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | EQEP2_INT (eQEP2) | EQEP1_INT (eQEP1) |
| INT6 | 保留 | 保留 | MXINTA (McBSP-A) | MRINTA (McBSP-A) | MXINTB (McBSP-B) | MRINTB (McBSP-B) | SPITXINTA (SPI-A) | SPIRXINTA (SPI-A) |
| INT7 | 保留 | 保留 | DINTCH6 (DMA) | DINTCH5 (DMA) | DINTCH4 (DMA) | DINTCH3 (DMA) | DINTCH2 (DMA) | DINTCH1 (DMA) |
| INT8 | 保留 | 保留 | SCITXINTC (SCI-C) | SCIRXINTC (SCI-C) | 保留 | 保留 | I2CINT2A (I2C-A) | I2CINT1A (I2C-A) |
| INT9 | ECAN1_INTB (CAN-B) | ECAN0_INTB (CAN-B) | ECAN1_INTA (CAN-A) | ECAN0_INTA (CAN-A) | SCITXINTB (SCI-B) | SCIRXINTB (SCI-B) | SCITXINTA (SCI-A) | SCIRXINTA (SCI-A) |
| INT10 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 |
| INT11 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 | 保留 |
| INT12 | LUF (FPU) | LVF (FPU) | 保留 | XINT7 | XINT6 | XINT5 | XINT4 | XINT3 |

PIE 配置和控制寄存器

| 名称 | 地址 | 大小 (X16) | 说明 |
|----|----|----------|----|
|----|----|----------|----|

| | | | |
|----------|---------------|---|-----------------|
| PIECTRL | 0x0CE0 | 1 | PIE,控制寄存器 |
| PIEACK | 0x0CE1 | 1 | PIE,确认寄存器 |
| PIEIER1 | 0x0CE2 | 1 | PIE,INT1组启用寄存器 |
| PIEIFR1 | 0x0CE3 | 1 | PIE,INT1组标志寄存器 |
| PIEIER2 | 0x0CE4 | 1 | PIE,INT2组启用寄存器 |
| PIEIFR2 | 0x0CE5 | 1 | PIE,INT2组标志寄存器 |
| PIEIER3 | 0x0CE6 | 1 | PIE,INT3组启用寄存器 |
| PIEIFR3 | 0x0CE7 | 1 | PIE,INT3组标志寄存器 |
| PIEIER4 | 0x0CE8 | 1 | PIE,INT4组启用寄存器 |
| PIEIFR4 | 0x0CE9 | 1 | PIE,INT4组标志寄存器 |
| PIEIER5 | 0x0CEA | 1 | PIE,INT5组启用寄存器 |
| PIEIFR5 | 0x0CEB | 1 | PIE,INT5组标志寄存器 |
| PIEIER6 | 0x0CEC | 1 | PIE,INT6组启用寄存器 |
| PIEIFR6 | 0x0CED | 1 | PIE,INT6组标志寄存器 |
| PIEIER7 | 0x0CEE | 1 | PIE,INT7组启用寄存器 |
| PIEIFR7 | 0x0CEF | 1 | PIE,INT7组标志寄存器 |
| PIEIER8 | 0x0CF0 | 1 | PIE,INT8组启用寄存器 |
| PIEIFR8 | 0x0CF1 | 1 | PIE,INT8组标志寄存器 |
| PIEIER9 | 0x0CF2 | 1 | PIE,INT9组启用寄存器 |
| PIEIFR9 | 0x0CF3 | 1 | PIE,INT9组标志寄存器 |
| PIEIER10 | 0x0CF4 | 1 | PIE,INT10组启用寄存器 |
| PIEIFR10 | 0x0CF5 | 1 | PIE,INT10组标志寄存器 |
| PIEIER11 | 0x0CF6 | 1 | PIE,INT11组启用寄存器 |
| PIEIFR11 | 0x0CF7 | 1 | PIE,INT11组标志寄存器 |
| PIEIER12 | 0x0CF8 | 1 | PIE,INT12组启用寄存器 |
| PIEIFR12 | 0x0CF9 | 1 | PIE,INT12组标志寄存器 |
| 保留 | 0x0CFA-0x0CFF | 6 | 保留 |

外部中断寄存器

| 名称 | 地址 | 大小 (X16) | 说明 |
|----------|---------------|-------------|-------------|
| XINT1CR | 0x007070 | 1 | XINT1配置寄存器 |
| XINT2CR | 0x007071 | 1 | XINT2配置寄存器 |
| XINT3CR | 0x007072 | 1 | XINT3配置寄存器 |
| XINT4CR | 0x007073 | 1 | XINT4配置寄存器 |
| XINT5CR | 0x007074 | 1 | XINT5配置寄存器 |
| XINT6CR | 0x007075 | 1 | XINT6配置寄存器 |
| XINT7CR | 0x007076 | 1 | XINT7配置寄存器 |
| XNMICR | 0x007077 | 1 | XNMI配置寄存器 |
| XINT1CTR | 0x007078 | 1 | XINT1计数器寄存器 |
| XINT2CTR | 0x007079 | 1 | XINT2计数器寄存器 |
| 保留 | 0x707A-0x707E | 5 | 保留 |
| XNMICTR | 0x00707A | 1 | XNMI计数器寄存器 |

兼容性差异说明

HRPWM 差异测试说明

HRPWM 是基于微边缘定位器(MEP)技术，MEP 逻辑能够通过细分常规 PWM 发生器的一个粗系统时钟来非常精细地定位边缘。

TMS320F28335的步长精度约为 150ps，CLM320F28335的步长精度约为120ps。

MEP 步长根据最坏情况工艺参数，工作温度和电压而变化，为了保证HRPWM输出的准确性，可使用 SFO函数产生的缩放因子进行校正（MEP Scale Factor），动态确定每个 EPWMCLK 周期的 MEP 步数。最终实现的效果是MEP步长*MEP缩放因子为定值。

MEP 比例因子测试对比结果如下所示：

| 测试大项 | 测试小项 | MEP缩放因子/分辨率 | | MEP缩放因子 * 分辨率(ps) | |
|----------|---------|--------------|--------------|-------------------|--------------|
| | | CLM320F28335 | TMS320F28335 | CLM320F28335 | TMS320F28335 |
| MEP 因子测试 | HRPWM 1 | 56/119ps | 48/138ps | 6664 | 6624 |
| | HRPWM 2 | 52/128ps | 49/136ps | 6656 | 6664 |
| | HRPWM 3 | 56/119ps | 49/136ps | 6664 | 6664 |
| | HRPWM 4 | 56/119ps | 49/136ps | 6664 | 6664 |

由上表可知目标板CLM320F28335 HRPWM2和TMS320F28335 HRPWM1相比其他通道存在偏差，目标板SFO校正结果偏差为9ps，对照板校正结果为40ps。

从SFO校正结果上看，CLM320F28335和TMS320F28335均存在通道偏差，CLM320F28335的校正结果偏差小于TMS320F28335的校正结果偏差。

X-ON Electronics

Largest Supplier of Electrical and Electronic Components

Click to view similar products for [Digital Signal Processors & Controllers - DSP, DSC category:](#)

Click to view products by [CHIPLON manufacturer:](#)

Other Similar products are found below :

[0W888-002-XTP](#) [DSPIC33EP512MU810T-E/BG](#) [AD21489WBCPZ402](#) [ISL68221IRAZ](#) [dsPIC33CDVL64MC106-E/M8](#) [ISL69224IRAZ](#)
[DSPIC33CK64MP102-E/M6](#) [ADSP-21565WCSWZ10](#) [CLM320VC5402PGE100](#) [CLM320F28335PGFA](#) [TMS320DM8168SCYGA2](#)
[OMAPL132EZWTA2R](#) [TMS320F28335ZJZQR](#) [dsPIC33FJ16GS504-E/ML](#) [DSPIC33CK1024MP705-E/PT](#) [DSPIC33CK1024MP708-E/PT](#)
[DSPIC33CK1024MP710-E/PT](#) [MC56F8027VLHR](#) [TYCOF2812PGFA](#) [DSPIC33EP128GP502-E/SS](#) [dsPIC33EP16GS502-EMM](#)
[DSPIC33EP512MC504-EML](#) [DSPIC33EP64MC502-E/MM](#) [dsPIC33EV256GM002-ESO](#) [ADSP-2187NBSTZ-320](#) [ADSP-2185MKSTZ-300](#)
[TMS320DM6433ZDU7](#) [MC56F8245VLD](#) [SM32C6713BGDPA20EP](#) [DSPIC33FJ64GP710-I/PF](#) [DSPIC33FJ32MC204-H/PT](#)
[DSPIC33FJ256MC510-I/PF](#) [DSPIC33FJ256GP510A-I/PT](#) [DSPIC33FJ06GS102-I/MM](#) [DSPIC33EP512MU810-I/BG](#) [DSPIC33EP32MC502-](#)
[I/SP](#) [DSPIC33EP32GS202-I/M6](#) [DSPIC33EP16GS506-I/PT](#) [DSPIC33EP16GS202-I/M6](#) [DSPIC33FJ64GS610-50I/PT](#) [DSPIC33FJ64GS610-](#)
[50I/PF](#) [DSPIC33FJ64GS606-50I/PT](#) [DSPIC33FJ32MC202-I/SP](#) [DSPIC33FJ32GP202-E/MM](#) [DSPIC33FJ256MC710-I/PF](#)
[DSPIC33FJ256GP710A-I/PF](#) [DSPIC33FJ256GP510A-I/PF](#) [DSPIC33FJ128GP802-I/MM](#) [DSPIC33EV32GM006-I/PT](#) [DSPIC33EP16GS504-](#)
[I/PT](#)