

HS32F7D377PTI双核MCU

1 器件概述

1.1 产品特性

- 双核架构
 - 2个Arm® Cortex®-M7 CPU
 - 300MHz
 - IEEE754单精度浮点(FPU)
 - 128KB ITCM per CPU
 - 128KB DTCM per CPU
 - 16KB iCache per CPU
- 片上存储器
 - 2个1MB eFlash
 - 256KB Global Share SRAM
 - 唯一标识号
- 时钟和系统控制
 - 2个内部零引脚10MHz OSC
 - 片上晶体振荡器
 - 窗口化看门狗计时器模块
 - 丢失时钟检测电路
- 1.1V内核、3.3V I/O电压
- 系统外设
 - 支持ASRAM和SDRAM外部存储器接口EMIF
 - 2个6通道DMA
 - 97个具有输入滤波功能的独立可编程、多路复用GPIO
 - 支持多个具有外部唤醒功能的低功耗模式
- 通信外设
 - USB 2.0
 - 3个CAN-FD接口(引脚可引导)
 - 3个SPI接口(引脚可引导)
 - 4个UART接口(引脚可引导)
 - 2个I2C接口(引脚可引导)
- 模拟子系统
 - 4个ADC
 - * 16bit模式
 - 采样率3.5MSPS
 - 差分输入
 - 差分10通道
 - * 12bit模式
 - 采样率7MSPS
 - 单端输入
 - 20通道
 - * 每个ADC有1个采样保持(S/H)电路
 - * 每个ADC有4个数字后处理电路
 - 偏移值校准
 - 参考值比较计算
 - 高、低和过零比较
 - 时间延迟可捕捉
 - 8个具有12bit DAC的比较器(CMPSS)
 - 3个12bit Buffered DAC
- 控制外设
 - 24个PWM通道
 - 16个HRPWM通道
 - * 8个PWM模块的A、B通道均具有高分辨率
 - * 支持死区(普通/高分辨率PWM均支持)
 - 6个CAP模块
 - 3个QEP模块
 - 2个SDFM模块, 每个SDFM模块包含4个 $\Sigma-\Delta$ 滤波器组, 每个滤波器组包含两个并联的滤波器
 - 标准SDFM数据滤波
 - 用于快速响应超范围情况的比较器滤波器
- 可配置逻辑模块(CLB)
 - 增强现有外设功能
 - 支持位置管理器解决方案
- 封装
 - 176引脚LQFP
- 温度
 - -40°C至125°C结温

1.2 应用

- 牵引逆变器电机控制
- HVAC大型商用电机控制
- 自动分拣设备
- CNC控制
- 交流充电（桩）站
- 直流充电（桩）站
- 电动汽车充电站电源模块
- 能量存储电源转换系统(PCS)
- 中央逆变器
- 太阳能电源优化器
- 串式逆变器
- 逆变器和电机控制
- 板载充电器(OBC)和无线充电器
- 线性电机分段控制器
- 伺服驱动器控制模块
- 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 框图

如图1.1所示。

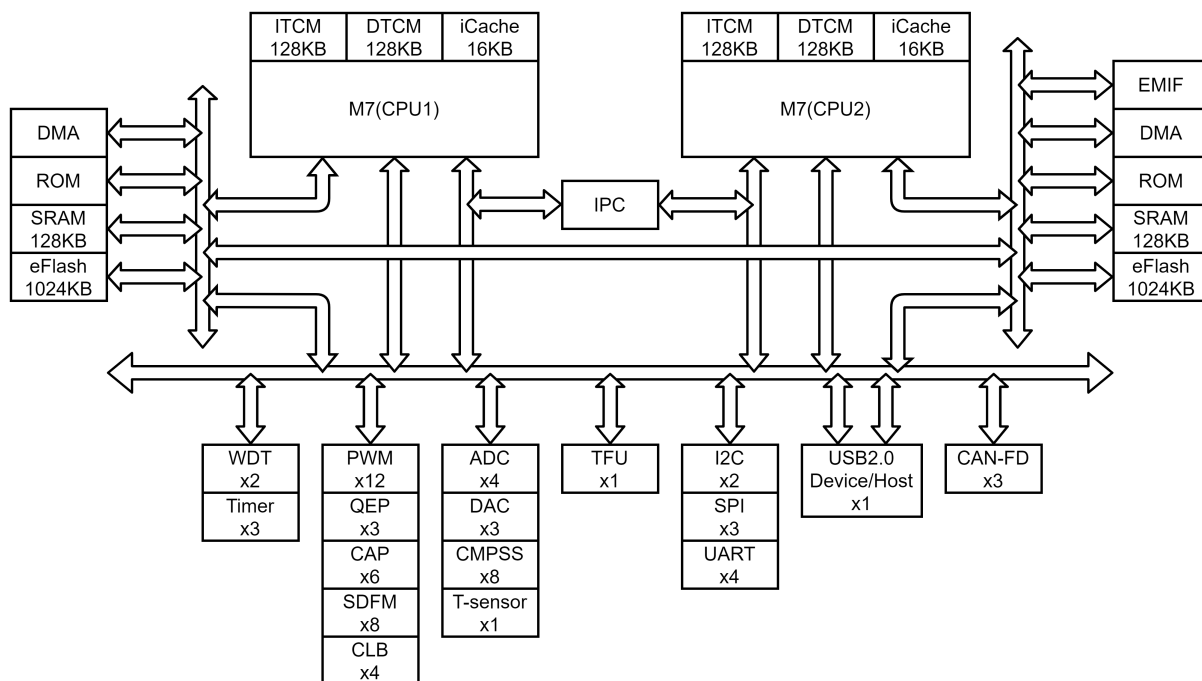


图 1.1 功能方框图

2 器件比较

器件比较如表2.1

表 2.1 器件比较

FEATURE		HS32F7D379PTI	HS32F7D377PTI	HS32F7D374PTI
Package Type (PTP is an HLQFP package.)		176-Pin PTP	176-Pin PTP	176-Pin PTP
Processor and Accelerators				
M7	Number	2	2	2
	Frequency (MHz)	300	300	300
	Floating-Point Unit (FPU)	Single precision	Single precision	Single precision
Hardware Accelerator	CRC	yes	yes	yes
	TFU	yes	yes	yes
6-Channel DMA – Type 0		2	2	2
Memory				
Flash (64-bit words)		2MB [1MB per CPU]	2MB [1MB per CPU]	2MB [1MB per CPU]
RAM (16-bit words)	ITCM	128KB	128KB	128KB
	DTCM	128KB	128KB	128KB
	ICACHE	16KB	16KB	16KB
	Global Shared RAM	256KB	256KB	192KB
	Message RAM	8KB [4KB per CPU]	8KB [4KB per CPU]	8KB [4KB per CPU]
Boot ROM		32KB for boot x1 16KB for boot x1	32KB for boot x1 16KB for boot x1	32KB for boot x1 16KB for boot x1
System				
Configurable Logic Block (CLB)		4 tiles	/	/
CPU timers		6 (3 per CPU)	6 (3 per CPU)	6 (3 per CPU)
Watchdog timers		2 (1 per CPU)	2 (1 per CPU)	2 (1 per CPU)
Nonmaskable Interrupt Watchdog (NMIWD) timers		2 (1 per CPU)	2 (1 per CPU)	2 (1 per CPU)
Crystal oscillator/External clock input		1	1	1
0-pin internal oscillator		2	2	2
I/O pins (shared)	GPIO	97	97	97
External interrupts		5	5	5
EMIF	EMIF1 (16-bit or 32-bit)	1	1	1
	EMIF2 (16-bit)	/	/	/
Analog Peripherals				
ADC 16-bit mode	MSPS	3.5	3.5	/
	Conversion Time (ns)	300	300	/
	Input pins	20	20	/
	Channels (differential)	10	10	/
ADC 12-bit mode	MSPS	7.0	7.0	7.0
	Conversion Time (ns)	150	150	150
	Input pins	20	20	20
	Channels (single-ended)	20	20	20
Number of 16-bit or 12-bit ADCs		4	4	4
Number of 12-bit only ADCs		/	/	
Temperature sensor		1	1	1
CMPSS (each CMPSS has two comparators and two internal DACs)		8	8	8
Buffered DAC		3	3	3
Control Peripherals				
eCAP inputs – Type 2		6	6	6
HRCAP		2	2	2
Enhanced Pulse Width Modulator (ePWM) channels – Type 4		24	24	24
eQEP modules – Type 2		3	3	3
High-resolution ePWM channels		16	16	16
SDFM channels – Type 2		8	8	8
Communication Peripherals				
Controller Area Network (CAN) FD		3	3	3
Inter-Integrated Circuit (I2C)		2	2	2
System Management Bus(SMBus)/ Power Management Bus(PMBus)		1	1	1
Universal Asynchronous Receiver/Transmitter (UART)		4	4	4
Serial Peripheral Interface (SPI)		3	3	3
USB 2.0 Host/Device		1	1	1
Temperature and Qualification				
Junction Temperature (Tj)	-40°C to 125°C	Yes	Yes	Yes

目录

1	器件概述	1
1.1	产品特性	1
1.2	应用	2
1.3	框图	3
2	器件比较	4
3	修订历史记录	9
4	引脚	10
4.1	引脚分配如图4.1所示	10
4.2	信号描述	11
4.2.1	模拟信号描述	11
4.2.2	GPIO和外设信号说明	14
4.2.3	其他信号说明	27
4.3	有内部和外部上拉下拉的引脚	30
4.4	引脚的多路复用	31
4.4.1	GPIO多路复用	31
4.4.2	输入X-BAR	34
4.4.3	输出X-Bar和ePWM X-Bar	36
4.4.4	USB引脚多路复用	37
4.4.5	高速SPI引脚多路复用	38
4.4.6	未使用引脚的连接	39
5	规格	40
5.1	绝对最大额定值	40
5.2	静电放电(ESD)等级	41
5.3	建议工作条件	41
5.4	功耗	42
5.4.1	模块功耗	43
5.5	电气特性	44
5.6	热阻特性	45
5.7	系统	46
5.7.1	上电时序	46
5.7.1.1	信号输入要求	46
5.7.1.2	V_{DDIO} , V_{DDA} , V_{DD3VFL} 和 V_{DDOSC} 要求	46
5.7.1.3	V_{DD} 要求	46
5.7.1.4	上电时间	46
5.7.1.5	电源监测	46
5.7.1.6	V_{DD} , V_{DD3VFL} 上下电时序要求	46
5.7.2	复位时序	46
5.7.3	复位源	47
5.7.4	时钟	47
5.7.4.1	时钟源	47

5.7.4.2	时钟频率	49
5.7.4.2.1	输入时钟频率	49
5.7.4.2.2	X1时序要求	49
5.7.4.2.3	AUXCLKIN时序要求	49
5.7.4.2.4	内部时钟频率	49
5.7.4.2.5	输出时钟频率	50
5.7.4.3	输入时钟和PLL	51
5.7.4.4	内部OSC	52
5.7.5	eFlash参数	53
5.7.6	JTAG	55
5.7.6.1	调试端口引脚	55
5.7.6.2	JTAG-DP	56
5.7.6.3	SW-DP	57
5.7.7	GPIO电气数据与时序	58
5.7.7.1	GPIO输出时序	58
5.7.7.1.1	通用输出开关特征	58
5.7.7.2	通用输入开关特征	58
5.7.7.2.1	通用输入时序要求	58
5.7.7.3	输入信号的采样窗口宽度	59
5.7.8	中断	61
5.7.8.1	特性	61
5.7.8.2	NVIC寄存器	61
5.7.8.2.1	NVIC_ISER	62
5.7.8.2.2	NVIC_ICER	62
5.7.8.2.3	NVIC_ICPR	62
5.7.8.2.4	NVIC_IABR	63
5.7.8.2.5	NVIC_IPR	63
5.7.8.2.6	NVIC_STIR	63
5.7.8.3	中断处理	63
5.7.8.4	中断和异常向量	64
5.7.9	低功耗模式	70
5.7.10	外部存储接口(EMIF)	71
5.7.10.1	异步读写时序	72
5.7.10.2	异步读写时序	74
5.8	模拟外设	76
5.8.1	模数转换器(ADC)	76
5.8.1.1	ADC可配置性	77
5.8.1.2	信号模式	78
5.8.1.3	ADC电气数据和时序	79
5.8.1.3.1	ADC输入模型	83
5.8.1.3.2	ADC时序图	86
5.8.1.4	温度传感器电气数据和时序	90
5.8.2	比较器子系统(CMPSS)	90
5.8.2.1	CMPSS电气数据和时序	91
5.8.3	缓冲数模转换器(DAC)	95

5.8.3.1	缓冲 DAC电气数据和时序	95
5.9	控制外设	99
5.9.1	增强型捕捉模块 (eCAP)	99
5.9.1.1	eCAP与HRCAP模块框图	99
5.9.1.2	eCAP的同步	100
5.9.1.3	eCAP电气数据及时序说明	100
5.9.2	高分辨率输入捕获模块 (HRCAP)	101
5.9.2.1	HRCAP模块框图	101
5.9.2.2	HRCAP时序	101
5.9.3	增强型脉宽调制器(ePWM)	102
5.9.3.1	控制外设同步(Control Peripherals Synchronization)	104
5.9.3.2	ePWM电气数据和时序	105
5.9.4	高分辨率脉宽调制器(HRPWM)	106
5.9.4.1	HRPWM电气数据和时序	106
5.9.5	增强型正交编码脉冲(eQEP)	107
5.9.5.1	eQEP电气数据和时序	108
5.9.6	Σ - Δ 滤波器模块(SDFM)	109
5.9.6.1	SDFM电气数据和时序 (使用ASYNC)	110
5.10	通信外设	112
5.10.1	控制器局域网络(CAN)	112
5.10.2	I2C (SMBus/PMBus)	114
5.10.3	UART	115
5.10.4	串行外设接口(SPI)	116
5.10.5	通用串行总线 (USB) 控制器	118
6	详细规格	119
6.1	概述	119
6.2	功能框图	119
6.3	内存	120
6.3.1	内存映射	120
6.3.2	Memory类型	123
6.3.2.1	TCM(Tightly Coupled Memory)	123
6.3.2.2	Global Share SRAM	123
6.4	芯片ID	123
6.5	总线结构	124
6.6	ARM M7	125
6.6.1	特性	125
6.6.2	接口	125
6.6.3	功能说明	127
6.6.3.1	处理器寄存器	127
6.6.3.2	数据处理单元(DPU)	128
6.6.3.3	预取单元(PFU)	128
6.6.3.4	负载存储单元(LSU)	128
6.6.3.5	浮点单元(FPU)	128
6.7	TFU	129

6.8	CRC	131
6.9	DMA	132
6.10	IPC	133
6.11	Bootrom与外设booting	134
6.11.1	CPU1 BOOT PIN配置	134
6.11.2	CPU1 BOOT外设PIN配置	135
6.11.3	CPU2 BOOT模式	136
6.11.4	Timer	136
6.11.5	NMIWD	136
6.12	watchdog	137
6.13	CLB	137
7	封装规格	138

3 修订历史记录

版本号	时间	内容描述
r1p0	2024/2/23	初始版本
r1p1	2024/3/4	添加4.4节的功耗数据 添加4.8.1.3节的ADC数据
r1p2	2024/3/5	替换了图3.1并修改了ePAD的相关描述
r1p3	2024/3/13	更新5.11节bootROM中的外设编号
r1p4	2024/3/25	删除了表3.2中McBSP和UPP的相关内容，将SCI替换成UART，删除了GPIO99，修改了表注
r1p5	2024/4/23	1.更新文档引脚图（图片上增加产品型号）2.删除了datasheet的4.9.4小节中，“HRPWM允许的最小HRPWMCLK频率为60 MHz”
r1p6	2024/5/18	提升图片清晰度，优化内容描述，优化格式
r1p7	2024/6/12	优化ADC 参数
r1p8	2024/7/4	更新芯片版本

4 引脚

4.1 引脚分配如图4.1所示

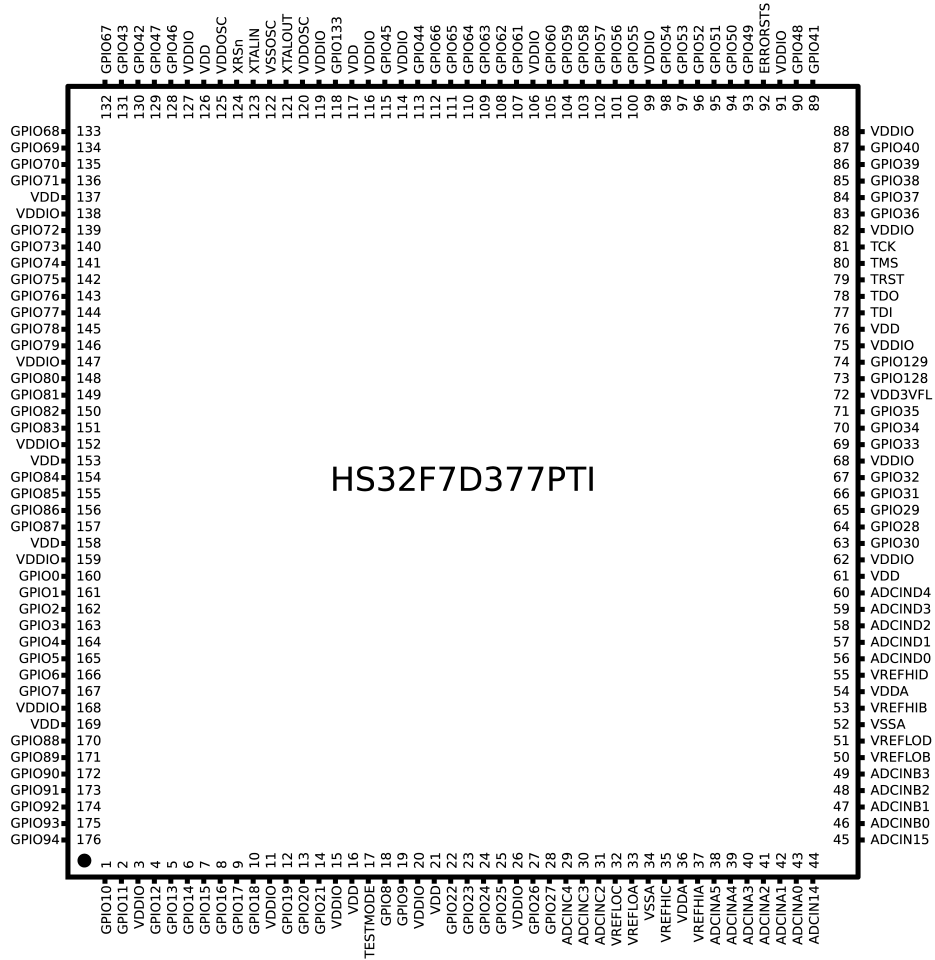


图 4.1 176Pin eQFP热增强型低剖面四通道扁平封装（顶视图）

请注意

eQFP封装的外露引线框裸片焊盘有两个功能：从芯片散热和为所有的地提供接地路径。因此，ePAD应焊接到PCB的接地（GND）平面，因为这将提供数字接地路径和良好的热传导路径。为了使eQFP封装中设计的热效率得到最佳利用，在设计PCB时必须考虑到这种技术。在ePAD主体正下方的PCB表面上需要散热焊盘。散热焊盘应焊接到ePAD封装的外露引线框裸片焊盘上；散热焊盘应尽可能大，以散发所需的热量。应使用一组散热过孔将散热焊盘与电路板的内部 GND平面连接。

4.2 信号描述

节4.2.1描述了这些信号。除非另有说明，否则复位时默认为GPIO功能。下述列出的外设信号是备用功能。所有GPIO引脚都为I/O/Z且有内部上拉下拉电阻器，可在每个引脚上有选择性地启用/禁用。这一特性只适用于GPIO引脚。复位时默认为上拉电阻未启用。

4.2.1 模拟信号描述

表 4.1 模拟信号描述

事项		I/O/Z ⁽¹⁾	描述
信号名	QFP176 Pin No.		
V_{REFHIA}	37	I	ADC-A高基准电压源。该电压必须从外部电路驱动到引脚中。对于12位模式，在此引脚上至少放置一个1 μ F电容器，对于16位模式，至少放置一个22 μ F电容器。该电容器应尽可能靠近器件放置在 V_{REFHIA} 和 V_{REFLOA} 引脚之间。注：请勿在此引脚添加外部负载。
V_{REFHIB}	53	I	ADC-B高基准电压源。该电压必须从外部电路驱动到引脚中。对于12位模式，在此引脚上至少放置一个1 μ F电容器，对于16位模式，至少放置一个22 μ F电容器。该电容器应尽可能靠近器件放置在 V_{REFHIB} 和 V_{REFLOB} 引脚之间。注：请勿在此引脚添加外部负载。
V_{REFHIC}	35	I	ADC-C高基准电压源。该电压必须从外部电路驱动到引脚中。对于12位模式，在此引脚上至少放置一个1 μ F电容器，对于16位模式，至少放置一个22 μ F电容器。该电容器应尽可能靠近器件放置在 V_{REFHIC} 和 V_{REFLOC} 引脚之间。注：请勿在此引脚添加外部负载。
V_{REFHID}	55	I	ADC-D高基准电压源。该电压必须从外部电路驱动到引脚中。对于12位模式，在此引脚上至少放置一个1 μ F电容器，对于16位模式，至少放置一个22 μ F电容器。该电容器应尽可能靠近器件放置在 V_{REFHID} 和 V_{REFLOD} 引脚之间。注：请勿在此引脚添加外部负载。
V_{REFLOA}	33	I	ADC-A 低基准电压源。
V_{REFLOB}	50	I	ADC-B 低基准电压源。
V_{REFLOC}	32	I	ADC-C 低基准电压源。
V_{REFLOD}	51	I	ADC-D 低基准电压源。
ADCIN14	44	I	所有ADC的输入14，该引脚可以作为通用ADCIN引脚，或是用于使用外部参考电源同时校准所有ADC（单端或差分）。
CMPIN4P		I	比较器4的正输入
ADCIN15	45	I	所有ADC的输入15，该引脚可以作为通用ADCIN引脚，或是用于使用外部参考电源同时校准所有ADC（单端或差分）。
CMPIN4N		I	比较器4的负输入
ADCINA0	43	I	ADC-A输入0，在ADC输入或DAC输出模式下，该引脚上有一个50k Ω 内部下拉，无法禁用。
DACOUTA		O	DAC-A输出
ADCINA1	42	I	ADC-A输入1，在ADC输入或DAC输出模式下，该引脚上有一个50k Ω 内部下拉，无法禁用。
DACOUTB		O	DAC-B输出

（续表见下页）

表3.1 模拟信号描述（续表）

事项		I/O/Z ⁽¹⁾	描述
信号名	QFP176 Pin No.		
ADCINA2	41	I	ADC-A输入2
CMPIN1P		I	比较器1的正输入
ADCINA3	40	I	ADC-A输入3
CMPIN1N		I	比较器1的负输入
ADCINA4	39	I	ADC-A输入4
CMPIN2P		I	比较器2的正输入
ADCINA5	38	I	ADC-A输入5
CMPIN2N		I	比较器2的负输入
ADCINB0	46	I	ADC-B输入0, 在ADC输入或DAC基准模式下, 此引脚通过一个100pF电容连接到VSSA, 无法禁用。如果此引脚用作片内DAC的基准电压源, 则在此引脚上至少放置一个1 μ F电容。
VDAC		I	用于片内DAC的可选外部基准电压。在ADC输入或DAC基准模式下, 此引脚通过一个100pF电容连接到VSSA, 无法禁用。如果此引脚用作片内DAC的基准电压源, 则在此引脚上至少放置一个1 μ F电容。
ADCINB1	47	I	ADC-B输入1, 在ADC输入或DAC输出模式下, 该引脚上有一个50 k Ω 内部下拉, 无法禁用。
DACOUTC		O	DAC-C输出
ADCINB2	48	I	ADC-B输入2
CMPIN3P		I	比较器3的正输入
ADCINB3	49	I	ADC-B输入3
CMPIN3N		I	比较器3的负输入
ADCINB4	-	I	ADC-B输入4
ADCINB5	-	I	ADC-B输入5
ADCINC2	31	I	ADC-C输入 2
CMPIN6P		I	比较器6的正输入
ADCINC3	30	I	ADC-C输入3
CMPIN6N		I	比较器6的负输入
ADCINC4	29	I	ADC-C输入4
CMPIN5P		I	比较器5的正输入
ADCINC5	-	I	ADC-C 输入5
CMPIN5N	-	I	比较器5的负输入
ADCIND0	56	I	ADC-D输入0
CMPIN7P		I	比较器7的正输入
ADCIND1	57	I	ADC-D输入1
CMPIN7N		I	比较器7的负输入
ADCIND2	58	I	ADC-D输入2
CMPIN8P		I	比较器8的正输入
ADCIND3	59	I	ADC-D输入3
CMPIN8N		I	比较器8的负输入

（续表见下页）

表3.1 模拟信号描述（续表）

事项		I/O/Z ⁽¹⁾	描述
信号名	QFP176 Pin No.		
ADCIND4	60	I	ADC-D输入4
ADCIND5	—	I	ADC-D输入5

(1) I=输入，O=输出，OD=漏极开路，Z=高阻抗

4.2.2 GPIO和外设信号说明

表 4.2 GPIO和外设信号说明

信号名	选择器序号	QFP176 Pin No.	I/O/Z ⁽¹⁾	描述
GPIO0	0, 4, 8, 12	160	I/O	通用输入/输出0
EPWM1A	1		O	增强型PWM1输出A（支持HRPWM）
SDAA	6		I/O	I2C-A数据漏极开路双向端口
GPIO1	0, 4, 8, 12	161	I/O	通用输入/输出1
EPWM1B	1		O	增强型PWM1输出B（支持HRPWM）
SCLA	6		I/O	I2C-A时钟漏极开路双向端口
GPIO2	0, 4, 8, 12	162	I/O	通用输入/输出2
EPWM2A	1		O	增强型PWM2输出A（支持HRPWM）
OUTPUTXBAR1	5		O	输出XBAR的输出1
SDAB	6		I/O	I2C-B数据漏极开路双向端口
GPIO3	0, 4, 8, 12	163	I/O	通用输入/输出3
EPWM2B	1		O	增强型PWM2输出B（支持HRPWM）
OUTPUTXBAR2	2		O	输出XBAR的输出2
OUTPUTXBAR2	5		O	输出XBAR的输出2
SCLB	6		I/O	I2C-B时钟漏极开路双向端口
GPIO4	0, 4, 8, 12	164	I/O	通用输入/输出4
EPWM3A	1		O	增强型PWM3输出A（支持HRPWM）
OUTPUTXBAR3	5		O	输出XBAR的输出3
CANTXA	6		O	CAN-A的TX端
CANTXC	9		O	CAN-C的TX端
GPIO5	0, 4, 8, 12	165	I/O	通用输入/输出5
EPWM3B	1		O	增强型PWM3输出B（支持HRPWM）
OUTPUTXBAR3	3		O	输出XBAR的输出3
CANRXA	6		I	CAN-A的TX端
CANRXC	9		I	CAN-C的RX端
GPIO6	0, 4, 8, 12	166	I/O	通用输入/输出6
EPWM4A	1		O	增强型PWM4输出A（支持HRPWM）
OUTPUTXBAR4	2		O	输出XBAR的输出4
EXTSYNCOUT	3		O	外部ePWM同步脉冲输出
EQEP3A	5		I	增强型QEP3输入A
CANTXB	6		O	CAN-B的TX端
GPIO7	0, 4, 8, 12	167	I/O	通用输入/输出7
EPWM4B	1		O	增强型PWM4输出B（支持HRPWM）
OUTPUTXBAR5	3		O	输出XBAR的输出5
EQEP3B	5		I	增强型QEP3输入B
CANRXB	6		I	CAN-B的RX端
GPIO8	0, 4, 8, 12	18	I/O	通用输入/输出8
EPWM5A	1		O	增强型PWM5输出A（支持HRPWM）

（续表见下页）

表4.2 GPIO和外设信号说明（续表）

信号名	选择器序号	QFP176 Pin No.	I/O/Z ⁽¹⁾	描述
CANTXB	2	18	O	CAN-B的TX端
ADCSOAO	3		O	ADC转换开始，用于外部ADC的输出A
EQEP3S	5		I/O	增强型QEP3输入Strobe
UARTTXDA	6		O	UART-A发送数据
CANTXC	9		O	CAN-C的TX端
GPIO9	0, 4, 8, 12	19	I/O	通用输入/输出9
EPWM5B	1		O	增强型PWM5输出B（支持HRPWM）
UARTTXDB	2		O	UART-B发送数据
OUTPUTXBAR6	3		O	输出XBAR的输出6
EQEP3I	5		I/O	增强型QEP3输入Index
UARTRXDA	6		I	UART-A接收数据
GPIO10	0, 4, 8, 12	1	I/O	通用输入/输出10
EPWM6A	1		O	增强型PWM6输出A（支持HRPWM）
CANRXB	2		I	CAN-B的RX端
ADCSOABO	3		O	ADC转换开始，用于外部ADC的输出B
EQEP1A	5		I	增强型QEP1输入A
UARTTXDB	6		O	UART-B发送数据
CANRXC	9		I	CAN-C的RX端
GPIO11	0, 4, 8, 12	2	I/O	通用输入/输出11
EPWM6B	1		O	增强型PWM6输出B（支持HRPWM）
UARTRXDB	6		I	UART-B接收数据
OUTPUTXBAR7	3		O	输出XBAR的输出7
EQEP1B	5		I	增强型QEP1输入B
GPIO12	0, 4, 8, 12	4	I/O	通用输入/输出12
EPWM7A	1		O	增强型PWM7输出A（支持HRPWM）
CANTXB	2		O	CAN-B的TX端
EQEP1S	5		I/O	增强型QEP1输入Strobe
UARTTXDC	6		O	UART-C发送数据
GPIO13	0, 4, 8, 12	5	I/O	通用输入/输出13
EPWM7B	1		O	增强型PWM7输出B（支持HRPWM）
CANRXB	2		I	CAN-B的RX端
EQEP1I	5		I/O	增强型QEP1输入Index
UARTRXDC	6		I	UART-C接收数据
GPIO14	0, 4, 8, 12	6	I/O	通用输入/输出14
EPWM8A	1		O	增强型PWM8输出A（支持HRPWM）
UARTTXDB	2		O	UART-B发送数据
OUTPUTXBAR3	6		O	输出XBAR的输出3
GPIO15	0, 4, 8, 12	7	I/O	通用输入/输出15
EPWM8B	1		O	增强型PWM8输出B（支持HRPWM）
UARTRXDB	2		I	UART-B接收数据

（续表见下页）

表4.2 GPIO和外设信号说明（续表）

信号名	选择器序号	QFP176 Pin No.	I/O/Z ⁽¹⁾	描述
OUTPUTXBAR4	6	7	O	输出XBAR的输出4
GPIO16	0, 4, 8, 12	8	I/O	通用输入/输出16
SPISIMOA	1		I/O	SPI-A的MOSI端
CANTXB	2		O	CAN-B的TX端
OUTPUTXBAR7	3		O	输出XBAR的输出7
EPWM9A	5		O	增强型PWM9输出A
SD1_D1	7		I	SDFM1的通道1数据输入
GPIO17	0, 4, 8, 12	9	I/O	通用输入/输出17
SPISOMIA	1		I/O	SPI-A的MISO端
CANRXB	2		I	CAN-B的RX端
OUTPUTXBAR8	3		O	输出XBAR的输出8
EPWM9B	5		O	增强型PWM9输出B
SD1_C1	7		I	SDFM1的通道1时钟输入
GPIO18	0, 4, 8, 12	10	I/O	通用输入/输出18
SPICLKA	1		I/O	SPI-A时钟
UARTTXDB	2		O	UART-B发送数据
CANRXA	3		I	CAN-A的RX端
EPWM10A	5		O	增强型PWM10输出A
SD1_D2	7		I	SDFM1的通道2数据输入
CANRXC	9	I	CAN-C的RX端	
GPIO19	0, 4, 8, 12	12	I/O	通用输入/输出19
SPISTEAn	1		I/O	SPI-A从模式的片选使能
UARTRXDB	2		I	UART-B接收数据
CANTXA	3		O	CAN-A的TX端
EPWM10B	5		O	增强型PWM10输出B
SD1_C2	7		I	SDFM1的通道2时钟输入
CANTXC	9	O	CAN-C的TX端	
GPIO20	0, 4, 8, 12	13	I/O	通用输入/输出20
EQEP1A	1		I	增强型QEP1输入A
CANTXB	3		O	CAN-B的TX端
EPWM11A	5		O	增强型PWM11输出A
SD1_D3	7		I	SDFM1的通道3数据输入
GPIO21	0, 4, 8, 12	14	I/O	通用输入/输出21
EQEP1B	1		I	增强型QEP1输入B
CANRXB	3		I	CAN-B的RX端
EPWM11B	5		O	增强型PWM11输出B
SD1_C3	7		I	SDFM1的通道1时钟输入
GPIO22	0, 4, 8, 12	22	I/O	通用输入/输出22
EQEP1S	1		I/O	增强型QEP1输入Strobe
UARTTXDB	3		O	UART-B的发送数据
EPWM12A	5		O	增强型PWM12输出A

（续表见下页）

表4.2 GPIO和外设信号说明（续表）

信号名	选择器序号	QFP176 Pin No.	I/O/Z ⁽¹⁾	描述
SPICLKB	6	22	I/O	SPI-B clock
SD1_D4	7		I	SDFM1的通道4数据输入
CANTXC	9		O	CAN-C的TX端
GPIO23	0, 4, 8, 12	23	I/O	通用输入/输出23
EQEP1I	1		I/O	增强型QEP1输入Index
UARTRXDB	3		I	UART-B接收数据
EPWM12B	5		O	增强型PWM12输出B
SPISTEBn	6		I/O	SPI-B的从模式传输使能
SD1_C4	7		I	SDFM1的通道4时钟输入
CANRXC	9		I	CAN-C的RX端
GPIO24	0, 4, 8, 12	24	I/O	通用输入/输出24
OUTPUTXBAR1	1		O	增强型输出XBAR的输出1
EQEP2A	2		I	增强型QEP2输入A
SPISIMOB	6		I/O	SPI-B的MOSI端
SD2_D1	7		I	SDFM2的通道1数据输入
PMBUSA_SCL	9		I/O	PMBus-A时钟漏极开路双向端口
GPIO25	0, 4, 8, 12	25	I/O	通用输入/输出25
OUTPUTXBAR2	1		O	增强型输出XBAR的输出2
EQEP2B	2		I	增强型QEP2输入B
SPISOMIB	6		I/O	SPI-B的MISO端
SD2_C1	7		I	SDFM2的通道1时钟输入
PMBUSA_SDA	9		I/O	PMBus-A数据漏极开路双向端口
GPIO26	0, 4, 8, 12	27	I/O	通用输入/输出26
OUTPUTXBAR3	1		O	增强型输出XBAR的输出3
EQEP2I	2		I/O	增强型QEP2输入Index
OUTPUTXBAR3	5		O	输出XBAR的输出3
SPICLKB	6		I/O	SPI-B的时钟
SD2_D2	7		I	SDFM2的通道2数据输入
PMBUSA_ALERT	9		I/O	PMBus-A漏极开路Alert双向端口
GPIO27	0, 4, 8, 12	28	I/O	通用输入/输出27
OUTPUTXBAR4	1		O	输出XBAR增强型的输出4
EQEP2S	2		I/O	增强型QEP2输入Strobe
OUTPUTXBAR4	5		O	输出XBAR的输出4
SPISTEBn	6		I/O	SPI-B的从模式传输使能
SD2_C2	7		I	SDFM2的通道2时钟输入
PMBUSA_CTL	9		I/O	PMBus-A控制信号
GPIO28	0, 4, 8, 12	64	I/O	通用输入/输出28
UARTRXDA	1		I	UART-A接收数据
EM1CS4n	2		O	外部存储器接口1片选信号4
OUTPUTXBAR5	5		O	输出XBAR增强型的输出1
EQEP3A	6		I	增强型QEP3输入A

（续表见下页）

表4.2 GPIO和外设信号说明（续表）

信号名	选择器序号	QFP176 Pin No.	I/O/Z ⁽¹⁾	描述
SD2_D3	7	64	I	SDFM2的通道3数据输入
GPIO29	0, 4, 8, 12	65	I/O	通用输入/输出29
UARTTXDA	1		O	UART-A传输数据
EM1SDCKE	2		O	外部存储器接口1 SDRAM时钟使能
OUTPUTXBAR6	5		O	输出XBAR的输出6
EQEP3B	6		I	增强型QEP3输入B
SD2_C3	7		I	SDFM2的通道3时钟输入
GPIO30	0, 4, 8, 12		63	I/O
CANRXA	1	I		CAN-A的RX端
EM1CLK	2	O		外部存储器接口1时钟信号
OUTPUTXBAR7	5	O		输出XBAR的输出7
EQEP3S	6	I/O		增强型QEP3输入Strobe
SD2_D4	7	I		SDFM2的通道4数据输入
CANRXC	9	I		CAN-C的RX端
GPIO31	0, 4, 8, 12	66	I/O	通用输入/输出31
CANTXA	1		O	CAN-A的TX端
EM1WE _n	2		O	外部存储器接口1写使能
OUTPUTXBAR8	5		O	输出XBAR的输出8
EQEP3I	6		I/O	增强型QEP3输入Index
SD2_C4	7		I	SDFM2的通道4时钟输入
CANTXC	9		O	CAN-C的TX端
GPIO32	0, 4, 8, 12	67	I/O	通用输入/输出32
SDAA	1		I/O	I2C-A数据漏极开路双向端口
EM1CS0 _n	2		O	外部存储器接口1片选信号0
CLB_OUTPUTXBAR1	7		O	CLB输出X-BAR的输出1
GPIO33	0, 4, 8, 12	69	I/O	通用输入/输出33
SCLA	1		I/O	I2C-A时钟漏极开路双向端口
EM1RNW	2		O	外部存储器接口1的读/非写信号
CLB_OUTPUTXBAR2	7		O	CLB输出X-BAR的输出2
GPIO34	0, 4, 8, 12	70	I/O	通用输入/输出34
OUTPUTXBAR1	1		O	输出XBAR的输出1
EM1CS2 _n	2		O	外部存储器接口1片选信号2
SDAB	6		I/O	I2C-B数据漏极开路双向端口
CLB_OUTPUTXBAR3	7		O	CLB输出X-BAR的输出3
GPIO35	0, 4, 8, 12	71	I/O	通用输入/输出35
UARTRXDA	1		I	UART-A接收数据
EM1CS3 _n	2		O	外部存储器接口1片选信号3
SCLB	6		I/O	I2C-B时钟漏极开路双向端口
CLB_OUTPUTXBAR4	7		O	CLB输出X-BAR的输出4
GPIO36	0, 4, 8, 12	83	I/O	通用输入/输出36
UARTTXDA	1		O	UART-A传输数据

（续表见下页）

表4.2 GPIO和外设信号说明（续表）

信号名	选择器序号	QFP176 Pin No.	I/O/Z ⁽¹⁾	描述
EM1WAIT	2	83	I	外部存储器接口1异步SRAM等待
CANRXA	6		I	CAN-A的RX端
CLB.OUTPUTXBAR5	7		O	CLB输出X-BAR的输出5
CANRXC	9		I	CAN-C的RX端
GPIO37	0, 4, 8, 12	84	I/O	通用输入/输出37
OUTPUTXBAR2	1		O	输出XBAR的输出2
EM1OEn	2		O	外部存储器接口1的输出使能
CANTXA	6		O	CAN-A的TX端
CLB.OUTPUTXBAR6	7		O	CLB输出X-BAR的输出6
CANTXC	9		O	CAN-C的TX端
GPIO38	0, 4, 8, 12	85	I/O	通用输入/输出38
EM1A0	2		O	外部存储器接口1的地址线0
UARTTXDC	5		O	UART-C传输数据
CANTXB	6		O	CAN-B的TX端
CLB.OUTPUTXBAR7	7		O	CLB输出X-BAR的输出7
GPIO39	0, 4, 8, 12	86	I/O	通用输入/输出39
EM1A1	2		O	外部存储器接口1的地址线1
UARTRXDC	5		I	UART-C接收数据
CANRXB	6		I	CAN-B的RX端
CLB.OUTPUTXBAR8	7		O	CLB输出X-BAR的输出8
GPIO40	0, 4, 8, 12	87	I/O	通用输入/输出40
EM1A2	2		O	外部存储器接口1的地址线2
SDAB	6		I/O	I2C-B数据漏极开路双向端口
GPIO41	0, 4, 8, 12	89	I/O	通用输入/输出 41, 对于使用休眠低功耗模式的应用, 该引脚用作GPIOHIBWAKE信号。有关详细信息, 请参阅HS32F7D377PTI微控制器技术参考手册中“系统控制”一章的“低功耗模式”部分。
EM1A3	2		O	外部存储器接口1的地址线3
SCLB	6		I/O	I2C-B时钟漏极开路双向端口
GPIO42	0, 4, 8, 12		130	I/O
SDAA	6	I/O		I2C-A数据漏极开路双向端口
USB0DM	7	I/O		USB PHY差分数据
UARTTXDA	15	O		UART-A 传输数据
GPIO43	0, 4, 8, 12	131	I/O	通用输入/输出43
SCLA	6		I/O	I2C-A时钟漏极开路双向端口
USB0DP	7		I/O	USB PHY差分数据
UARTRXDA	15		I	UART-A接受数据
GPIO44	0, 4, 8, 12	113	I/O	通用输入/输出44
EM1A4	2		O	外部存储器接口1的地址线4
GPIO45	0, 4, 8, 12	115	I/O	通用输入/输出45
EM1A5	2		O	外部存储器接口1的地址线5

（续表见下页）

表4.2 GPIO和外设信号说明（续表）

信号名	选择器序号	QFP176 Pin No.	I/O/Z ⁽¹⁾	描述
dft_osc2_clk	14	115	O	osc1输出时钟，用于功能测试下osc1的trim
GPIO46	0, 4, 8, 12	128	I/O	通用输入/输出46
EM1A6	2		O	外部存储器接口1的地址线6
UARTRXDD	6		I	UART-D接收数据
GPIO47	0, 4, 8, 12	129	I/O	通用输入/输出47
EM1A7	2		O	外部存储器接口1的地址线7
UARTTXDD	6		O	UART-D发送数据
GPIO48	0, 4, 8, 12	90	I/O	通用输入/输出48
OUTPUTXBAR3	1		O	输出XBAR的输出3
EM1A8	2		O	外部存储器接口1的地址线8
UARTTXDA	6		O	UART-A发送数据
SD1_D1	7		I	SDFM1的通道1数据输入
GPIO49	0, 4, 8, 12	93	I/O	通用输入/输出49
OUTPUTXBAR4	1		O	输出XBAR的输出4
EM1A9	2		O	外部存储器接口1的地址线9
UARTRXDA	6		I	UART-A接收数据
SD1_C1	7		I	SDFM1的通道1时钟输入
GPIO50	0, 4, 8, 12	94	I/O	通用输入/输出50
EQEP1A	1		I	增强型QEP3输入A
EM1A10	2		O	外部存储器接口1的地址线10
SPISIMOC	6		I/O	SPI-C的MOSI端
SD1_D2	7		I	SDFM1的通道2数据输入
GPIO51	0, 4, 8, 12	95	I/O	通用输入/输出51
EQEP1B	1		I	增强型QEP3输入B
EM1A11	2		O	外部存储器接口1的地址线11
SPISOMIC	6		I/O	SPI-C的MISO端
SD1_C2	7		I	SDFM1的通道2时钟输入
GPIO52	0, 4, 8, 12	96	I/O	通用输入/输出52
EQEP1S	1		I/O	增强型QEP1输入Strobe
EM1A12	2		O	外部存储器接口1的地址线12
SPICLKC	6		I/O	SPI-C时钟
SD1_D3	7		I	SDFM1的通道3数据输入
GPIO53	0, 4, 8, 12	97	I/O	通用输入/输出53
EQEP1I	1		I/O	增强型QEP1输入Index
EM1D31	2		I/O	外部存储器接口1的地址线31
DBGIN15	13		O	调试输出端口15
SPISTECn	6		I/O	SPI-C从模式的传输使能信号
SD1_C3	7		I	SDFM1的通道3时钟输入
GPIO54	0, 4, 8, 12	98	I/O	通用输入/输出54
SPISIMOA	1		I/O	SPI-A的MOSI端
EM1D30	2		I/O	外部存储器接口1的地址线30

（续表见下页）

表4.2 GPIO和外设信号说明（续表）

信号名	选择器序号	QFP176 Pin No.	I/O/Z ⁽¹⁾	描述
DBGIN14	13	98	O	调试输出端口14
EQEP2A	5		I	增强型QEP2输入 A
UARTTXDB	6		O	UART-B发送数据
SD1_D4	7		I	SDFM1的通道4数据输入
GPIO55	0, 4, 8, 12	100	I/O	通用输入/输出55
SPISOMIA	1		I/O	SPI-A的MISO端
EM1D29	2		I/O	外部存储器接口1的地址线29
DBGIN13	13		O	调试输出端口13
EQEP2B	5		I	增强型QEP2输入B
UARTRXDB	6		I	UART-B接收数据
SD1_C4	7		I	SDFM1的通道4时钟输入
GPIO56	0, 4, 8, 12	101	I/O	通用输入/输出56
SPICLKA	1		I/O	SPI-A时钟
EM1D28	2		I/O	外部存储器接口1的地址线28
DBGIN12	13		O	调试输出端口12
EQEP2S	5		I/O	增强型QEP2输入Strobe
UARTTXDC	6		O	UART-C发送数据
SD2_D1	7		I	SDFM2的通道1数据输入
GPIO57	0, 4, 8, 12	102	I/O	通用输入/输出57
SPISTEAn	1		I/O	SPI-A从模式的传输使能信号
EM1D27	2		I/O	外部存储器接口1的地址线27
DBGIN11	13		O	调试输出端口11
EQEP2I	5		I/O	增强型QEP2输入Index
UARTRXDC	6		I	UART-C接收数据
SD2_C1	7		I	SDFM2的通道1时钟输入
GPIO58	0, 4, 8, 12	103	I/O	通用输入/输出58
EM1D26	2		I/O	外部存储器接口1的地址线26
DBGIN10	13		O	调试输出端口10
OUTPUTXBAR1	5		O	输出XBAR的输出1
SPICLKB	6		I/O	SPI-B时钟
SD2_D2	7		I	SDFM2的通道2数据输入
SPISIMOA	15		I/O	SPI-A的MOSI端，支持SPI的高速GPIO多路复用器选项。
GPIO59	0, 4, 8, 12	104	I/O	通用输入/输出 59，该引脚的输出阻抗可低至22Ω。该输出可能具有快速边沿和振铃，具体取决于系统PCB特性。如有问题，用户应采取预防措施如添加39 Ω（10%容差）的串联端接电阻或其他方案。建议使用提供的IBIS模型进行系统级信号完整性分析。如果此引脚用于输入功能，则不需要端接。
EM1D25	2		I/O	外部存储器接口1的地址线25
DBGIN9	13		O	调试输出端口9

（续表见下页）

表4.2 GPIO和外设信号说明（续表）

信号名	选择器序号	QFP176 Pin No.	I/O/Z ⁽¹⁾	描述
OUTPUTXBAR2	5	104	O	输出XBAR的输出2
SPISTEBn	6		I/O	SPI-B从模式传输使能
SD2_C2	7		I	SDFM2的通道2时钟输入
SPISOMIA	15		I/O	SPI-A的MISO端，支持SPI的高速GPIO多路复用器选项。
GPIO60	0, 4, 8, 12	105	I/O	通用输入/输出60
EM1D24	2		I/O	外部存储器接口1的地址线24
DBGIN8	13		O	调试输出端口8
OUTPUTXBAR3	5		O	输出XBAR的输出3
SPISIMOB	6		I/O	SPI-B的MOSI端
SD2_D3	7		I	SDFM2的通道3数据输入
SPICLKA	15		I/O	SPI-A时钟
GPIO61	0, 4, 8, 12	107	I/O	通用输入/输出61
EM1D23	2		I/O	外部存储器接口1的地址线23
DBGIN7	13		O	调试输出端口7
OUTPUTXBAR4	5		O	输出XBAR的输出4
SPISOMIB	6		I/O	SPI-B的MISO端
SD2_C3	7		I	SDFM2的通道3时钟输入
SPISTEA _n	15		I/O	SPI-A从模式的传输使能
GPIO62	0, 4, 8, 12	108	I/O	通用输入/输出62
UARTRXDC	1		I	UART-C接收数据
EM1D22	2		I/O	外部存储器接口1的地址线22
DBGIN6	13		O	调试输出端口6
EQEP3A	5		I	增强型QEP3输入A
CANRXA	6		I	CAN-A receive
SD2_D4	7		I	SDFM2的通道4数据输入
GPIO63	0, 4, 8, 12	109	I/O	通用输入/输出63
UARTTXDC	1		O	UART-C发送数据
EM1D21	2		I/O	外部存储器接口1的地址线21
DBGIN5	13		O	调试输出端口5
EQEP3B	5		I	增强型QEP3输入B
CANTXA	6		O	CAN-A的TX端
SD2_C4	7		I	SDFM2的通道4时钟输入
SPISIMOB	15	I/O	SPI-B的MOSI端	
GPIO64	0, 4, 8, 12	110	I/O	通用输入/输出64，该引脚的输出阻抗可低至22Ω。该输出可能具有快速边沿和振铃，具体取决于系统PCB特性。如有问题，用户应采取预防措施如添加39Ω（10%容差）的串联端接电阻或其他方案。建议使用提供的IBIS模型进行系统级信号完整性分析。如果此引脚用于输入功能，则不需要端接。
EM1D20	2		I/O	外部存储器接口1的地址线20

（续表见下页）

表4.2 GPIO和外设信号说明（续表）

信号名	选择器序号	QFP176 Pin No.	I/O/Z ⁽¹⁾	描述
DBGIN4	13	110	O	调试输出端口4
EQEP3S	5		I/O	增强型QEP3输入Strobe
UARTRXDA	6		I	UART-A接收数据
SPISOMIB	15		I/O	SPI-B的MISO端，支持SPI的高速GPIO多路复用器选项。
GPIO65	0, 4, 8, 12	111	I/O	通用输入/输出65
EM1D19	2		I/O	外部存储器接口1的地址线19
DBGIN3	13		O	调试输出端口3
EQEP3I	5		I/O	增强型QEP3输入Index
UARTTXDA	6		O	UART-A发送数据
SPICLKB	15		I/O	SPI-B时钟，支持SPI的高速GPIO多路复用器选项。
GPIO66	0, 4, 8, 12	112	I/O	通用输入/输出66
EM1D18	2		I/O	外部存储器接口1的地址线18
DBGIN2	13		O	调试输出端口2
SDAB	6		I/O	I2C-B数据漏极开路双向端口
SPISTEBn	15		I/O	SPI-B从模式传输使能
GPIO67	0, 4, 8, 12	132	I/O	通用输入/输出67
EM1D17	2		I/O	外部存储器接口1的地址线17
DBGIN1	13		O	调试输出端口1
GPIO68	0, 4, 8, 12	133	I/O	通用输入/输出68
EM1D16	2		I/O	外部存储器接口1的地址线16
DBGIN0	13		O	调试输出端口0
GPIO69	0, 4, 8, 12	134	I/O	通用输入/输出69
EM1D15	2		I/O	外部存储器接口1的地址线15
DBGIN15	13		O	调试输出端口15
SCLB	6		I/O	I2C-B时钟漏极开路双向端口
SPISIMOC	15		I/O	SPI-C的MOSI端
GPIO70	0, 4, 8, 12	135	I/O	通用输入/输出70
EM1D14	2		I/O	外部存储器接口1的地址线14
DBGIN14	13		O	调试输出端口14
CANRXA	5		I	CAN-A的RX端
UARTTXDB	6		O	UART-B发送数据
CANRXC	9		I	CAN-C的RX端
SPISOMIC	15		I/O	SPI-C的MISO端
GPIO71	0, 4, 8, 12	136	I/O	通用输入/输出71
EM1D13	2		I/O	外部存储器接口1的地址线13
DBGIN13	13		O	调试输出端口13
CANTXA	5		O	CAN-A的TX端
UARTRXDB	6		I	UART-B接收数据
CANTXC	9		O	CAN-C的TX端

（续表见下页）

表4.2 GPIO和外设信号说明（续表）

信号名	选择器序号	QFP176 Pin No.	I/O/Z ⁽¹⁾	描述
SPICLKC	15	136	I/O	SPI-C时钟(2)
GPIO72	0, 4, 8, 12	139	I/O	通用输入/输出72, 出厂默认引导模式选择引脚1。
EM1D12	2		I/O	外部存储器接口1的地址线12
DBGIN12	13		O	调试输出端口12
CANTXB	5		O	CAN-B的TX端
UARTTXDC	6		O	UART-C发送数据
SPISTECn	15		I/O	SPI-C从模式传输使能
GPIO73	0, 4, 8, 12		140	I/O
EM1D11	2	I/O		外部存储器接口1的地址线11
DBGIN11	13	O		调试输出端口11
XCLKOUT	3	O/Z		外部时钟输出, 通过CLKSRCCTL3选择器件内部时钟信号的分频时钟。XCLKOUTSEL位选择时钟信号, XCLKOUTDIVSEL位选择分频比。
CANRXB	5	I		CAN-B的RX端
UARTRXDC	6	I		UART-C接收数据
GPIO74	0, 4, 8, 12	141		I/O
EM1D10	2		I/O	外部存储器接口1的地址线10
DBGIN10	13		O	调试输出端口10
CANTXC	9		O	CAN-C的TX端
GPIO75	0, 4, 8, 12	142	I/O	通用输入/输出75
EM1D9	2		I/O	外部存储器接口1的地址线9
DBGIN9	13		O	调试输出端口9
CANRXC	9		I	CAN-C的RX端
GPIO76	0, 4, 8, 12	143	I/O	通用输入/输出76
EM1D8	2		I/O	外部存储器接口1的地址线8
DBGIN8	13		O	调试输出端口8
UARTTXDD	6		O	UART-D发送数据
GPIO77	0, 4, 8, 12	144	I/O	通用输入/输出77
EM1D7	2		I/O	外部存储器接口1的地址线7
DBGIN7	13		O	调试输出端口7
UARTRXDD	6		I	UART-D接收数据
GPIO78	0, 4, 8, 12	145	I/O	通用输入/输出78
EM1D6	2		I/O	外部存储器接口1的地址线6
DBGIN6	13		O	调试输出端口6
EQEP2A	6		I	增强型QEP2输入A
GPIO79	0, 4, 8, 12	146	I/O	通用输入/输出79
EM1D5	2		I/O	外部存储器接口1的地址线5
DBGIN5	13		O	调试输出端口5
EQEP2B	6		I	增强型QEP2输入B
GPIO80	0, 4, 8, 12	148	I/O	通用输入/输出80
EM1D4	2		I/O	外部存储器接口1的地址线4

(续表见下页)

表4.2 GPIO和外设信号说明（续表）

信号名	选择器序号	QFP176 Pin No.	I/O/Z ⁽¹⁾	描述
DBGIN4	13	148	O	调试输出端口4
EQEP2S	6		I/O	增强型QEP2输入Strobe
GPIO81	0, 4, 8, 12	149	I/O	通用输入/输出81
EM1D3	2		I/O	外部存储器接口1的地址线3
DBGIN3	13		O	调试输出端口3
EQEP2I	6		I/O	增强型QEP2输入Index
GPIO82	0, 4, 8, 12	150	I/O	通用输入/输出82
EM1D2	2		I/O	外部存储器接口1的地址线2
DBGIN2	13		O	调试输出端口2
GPIO83	0, 4, 8, 12	151	I/O	通用输入/输出83
EM1D1	2		I/O	外部存储器接口1的地址线1
DBGIN1	13		O	调试输出端口1
GPIO84	0, 4, 8, 12	154	I/O	通用输入/输出84，出厂默认引导模式选择引脚0。
EM1A13	2		O	外部存储器接口1的地址线13
UARTTXDA	5		O	UART-A发送数据
GPIO85	0, 4, 8, 12	155	I/O	通用输入/输出85
EM1D0	2		I/O	外部存储器接口1的地址线0
DBGIN0	13		O	调试输出端口0
UARTRXDA	5		I	UART-A接收数据
GPIO86	0, 4, 8, 12	156	I/O	通用输入/输出86
EM1A13	2		O	外部存储器接口1的地址线13
EM1CAS	3		O	外部存储器接口1的列地址选通信号
UARTTXDB	5		O	UART-B发送数据
GPIO87	0, 4, 8, 12	157	I/O	通用输入/输出87
EM1A14	2		O	外部存储器接口1的地址线14
EM1RAS	3		O	外部存储器接口1的行地址选通信号
UARTRXDB	5		I	UART-B接收数据
GPIO88	0, 4, 8, 12	170	I/O	通用输入/输出88
EM1A15	2		O	外部存储器接口1的地址线15
EM1DQM0	3		O	外部存储器接口1的字节0输入/输出掩码
GPIO89	0, 4, 8, 12	171	I/O	通用输入/输出89
EM1A16	2		O	外部存储器接口1的地址线16
EM1DQM1	3		O	外部存储器接口1的字节1输入/输出掩码
UARTTXDC	6		O	UART-C发送数据
GPIO90	0, 4, 8, 12	172	I/O	通用输入/输出90
EM1A17	2		O	外部存储器接口1的地址线17
EM1DQM2	3		O	外部存储器接口1的字节2输入/输出掩码
UARTRXDC	6		I	UART-C接收数据
GPIO91	0, 4, 8, 12	173	I/O	通用输入/输出91
EM1A18	2		O	外部存储器接口1的地址线18
EM1DQM3	3		O	外部存储器接口1的字节3输入/输出掩码

（续表见下页）

表4.2 GPIO和外设信号说明（续表）

信号名	选择器序号	QFP176 Pin No.	I/O/Z ⁽¹⁾	描述
SDAA	6	173	I/O	I2C-A数据漏极开路双向端口
PMBUSA_SCL	10		I/O	PMBus-A时钟漏极开路双向端口
CLB_OUTPUTXBAR1	14		O	CLB输出的X-BAR输出1
GPIO92	0, 4, 8, 12	174	I/O	通用输入/输出92
EM1A19	2		O	外部存储器接口1的地址线19
EM1BA1	3		O	外部存储器接口1的bank地址线1
SCLA	6		I/O	I2C-A时钟漏极开路双向端口
PMBUSA_SDA	10		I/O	PMBus-A数据漏极开路双向端口
CLB_OUTPUTXBAR2	14		O	CLB输出的X-BAR输出2
GPIO93	0, 4, 8, 12	175	I/O	通用输入/输出93
EM1BA0	3		O	外部存储器接口1的bank地址线0
UARTTXDD	6		O	UART-D发送数据
PMBUSA_ALERT	10		I/O	PMBus-A漏极开路双向Alert端口
CLB_OUTPUTXBAR3	14		O	CLB输出的X-BAR输出3
GPIO94	0, 4, 8, 12	176	I/O	通用输入/输出94
EM1A14	2		O	外部存储器接口1的地址线14
UARTRXDD	6		I	UART-D接收数据
PMBUSA_CTL	10		I/O	PMBus-A控制信号
CLB_OUTPUTXBAR4	14		O	CLB输出的X-BAR输出4
GPIO128	0, 4, 8, 12	73	I/O	通用输入/输出128 ⁽⁴⁾
UARTTXDD	3		O	UART-D发送数据
CANTXA	5		O	CAN-A的TX端
SDAA	6		I/O	I2C-A数据漏极开路双向端口
GPIO129	0, 4, 8, 12	74	I/O	通用输入/输出129 ⁽⁴⁾
UARTRXDD	3		I	UART-D接收数据
CANRXA	5		I	CAN-A的RX端
SCLA	6		I/O	I2C-A 时钟漏极开路双向端口
GPIO130	0, 4, 8, 12	119	I/O	通用输入/输出130 ⁽³⁾
dft_osc1_clk	14		O	osc1输出时钟，用于功能测试下osc1的trim
RSTN_EXTPCB	15		O	复位PCB板上的外部芯片
GPIO133	0, 4, 8, 12	118	I/O	通用输入/输出133，该GPIO引脚的AUXCLKIN功能可用于向辅助锁相环(AUXPLL)提供单端3.3V电平时钟信号，然后AUXPLL的输出用于USB模块。AUXCLKIN时钟也可用于CAN模块。
SD2_C2	7		I	SDFM2的通道2时钟输入

(1) I=输入，O=输出，OD=漏极开路，Z=高阻抗。GPIO不支持OD功能，但不影响I2C等需要OD IO的外设使用

(2) GPIO99为TESTMODE，用来做FT测试，建议接Ground或者floating。

(3) GPIO130为GPIO信号，默认接H信号。

(4) GPIO128, GPIO129是同位置的2个测试管脚

(5) QFP176封装，无EMIF2接口。

4.2.3 其他信号说明

表 4.3 其他信号说明

事项		I/O/Z ⁽¹⁾	描述
信号名	QFP176 Pin No.		
复位			
XRSn	124	I	设备复位（输入）和看门狗重置（输出）。这些器件具有内置的上电复位（POR）电路。在上电条件下，该引脚由器件驱动为低电平。外部电路也可以驱动该引脚以置位器件复位。当发生看门狗复位或NMI看门狗复位时，MCU也会将该引脚驱动为低电平。在看门狗复位期间，XRSn引脚被驱动为低电平，看门狗复位持续时间为512个OSCCLK周期。应在XRSn和VDDIO之间放置一个值为10kΩ至20kΩ的电阻。如需在XRSn和VSS间进行噪声滤波，所选的电容至少为2μF。当看门狗复位被置位时，这些值将允许看门狗在512个OSCCLK周期内将XRSn引脚正确驱动至V _{OL} 。
时钟			
XTALIN	123	I	片上晶体振荡器输入。要使用此振荡器，必须将石英晶体连接在X1（XTALIN）和X2（XTALOUT）上。如果不使用此引脚，则必须将其连接到GND。该引脚还可用于馈电单端3.3V电平时钟。在本例中，X2（XTALOUT）是无连接（NC）。
XTALOUT	121	O	片上晶体振荡器输出。石英晶体可以连接在X1（XTALIN）和X2（XTALOUT）上。如果未使用X2，则必须保持未连接状态。
JTAG			
TCK	81	I	带内部上拉的JTAG测试时钟
TDI	77	I	具有内部上拉的JTAG测试数据输入（TDI）。TDI在TCK的上升沿上时钟进入选定的寄存器（指令或数据）。
TDO	78	O/Z	JTAG扫描测试数据输出（TDO）。所选寄存器的内容（指令或数据）在TCK的下降沿移出TDO。
TMS	80	I	具有内部上拉的JTAG测试模式选择（TMS），该串行控制输入时钟进入TCK上升沿的TAP控制器。
TRSTn	79	I	具有内部下拉功能的JTAG测试复位。当驱动为高电平时，TRSTn使扫描系统能够控制设备的操作。如果该信号被驱动为低电平，则器件在其功能模式下工作，并且测试复位信号将被忽略。注意：在设备正常运行期间，TRSTn必须始终保持低电平。该引脚上需要一个外部下拉电阻。该电阻的值应基于适用于设计的调试器Pod的驱动强度。2.2kΩ或更小的电阻通常可提供足够的保护。电阻器的值因应用而异。建议验证每个目标板，以确保调试器和应用正常运行。该引脚具有一个内部50ns（标称）毛刺滤波器。
模拟，数字以及IO电源			
VDD	16		1.2V数字逻辑电源引脚。建议在每个VDD引脚附近放置一个去耦电容，最小总电容约为20μF。去耦电容的确切值应由系统电压调节解决方案确定。
	21		
	61		

（续表见下页）

表4.3 其他信号说明（续表）

事项		I/O/Z ⁽¹⁾	描述
信号名	QFP176 Pin No.		
	76		
	117		
	126		
	137		
	153		
	158		
	169		
VDD3VFL	72		3.3V闪存电源引脚，每个引脚上放置一个最小0.1 μ F去耦电容。
VDDA	36		3.3V模拟电源引脚。在每个引脚上放置一个至少2.2 μ F的去耦电容至VSSA。
	54		
VDDIO	3		3.3V数字I/O电源引脚。在每个引脚上放置一个最小0.1 μ F去耦电容。去耦电容的确切值应由系统电压调节解决方案确定。
	11		
	15		
	20		
	26		
	62		
	68		
	75		
	82		
	88		
	91		
	99		
	106		
	114		
	116		
127			
138			
147			
152			
159			
168			
VDDOSC	120		用于3.3V片内晶体振荡器（X1和X2）和两个零引脚内部振荡器的电源引脚。在每个引脚上放置一个0.1 μ F（最小值）去耦电容。
	125		
VSS	PWR PAD		设备接地。对于四方扁平封装（QFP），封装底部的ePAD必须焊接到PCB的接地层。
VSSOSC	122		晶体振荡器（XTAILIN和XTAILED）接地引脚。使用外部晶体时，请勿将此引脚连接到电路板接地。相反，应将其连接到外部晶体振荡器电路的接地参考。如果不使用外部晶体，该引脚可能连接到电路板接地。

（续表见下页）

表4.3 其他信号说明（续表）

事项		I/O/Z ⁽¹⁾	描述
信号名	QFP176 Pin No.		
VSSA	34		模拟地
	52		
特殊功能			
ERRORSTS	92	O	错误状态输出。该引脚具有内部下拉功能。
测试管脚			
TESTMODE	17	I	所有芯片的测试模式

- (1) I=输入，O=输出，OD=漏极开路，Z=高阻抗
- (2) XTALIN对应引脚X1，XTALOUT对应引脚X2
- (3) QFP176封装
- (4) TESTMODE在QFP176封装中Pin No.为17
- (5) XRSn不支持输出，是纯输入
- (6) XRSn, VDDIO和XRSn之间的上拉电阻10kΩ-20kΩ，XRSn和Ground之间的电容>10μF

4.3 有内部和外部上拉下拉的引脚

器件上的某些引脚具有内部上拉或下拉。列出了拉动方向及其活动时间。默认情况下，GPIO引脚的上拉被禁用，可以通过软件启用。为了避免任何浮动的未绑定输入，引导ROM将在特定封装中对未绑定的GPIO引脚启用内部上拉。中提到的带有上拉和下拉的其他引脚始终处于打开状态且无法被禁用。

表 4.4 带有内部上拉和下拉的引脚

引脚	复位(XRSn=0)	器件引导	应用软件
GPIOx	禁用上拉	禁用上拉 ⁽¹⁾	上拉使能由应用定义
TRSTn		下拉有效	
TCK		上拉有效	
TMS		上拉有效	
TDI		上拉有效	
XRSn		上拉有效	
VREGENZ		下拉有效	
ERRORSTS		下拉有效	
其他引脚		上拉或下拉未存在	

(1) 封装中未bond出的引脚会由Boot ROM配置内部上拉

4.4 引脚的多路复用

4.4.1 GPIO多路复用

表4.5显示了多路复用引脚。每个引脚默认具有GPIO功能，可以通过设置GPyGMUXn.GPIOz和GPyMUXn.GPIOz寄存器来选择辅助功能。GPyGMUXn寄存器应在GPyMUXn之前配置，以避免交替的多路复用选择对GPIO产生瞬时脉冲。未显示栏和空白单元格保留为GPIO多路复用器设置。

表 4.5 GPIO多路复用引脚

GPIO序号	GPIO选择器选择										
	0,4,8,12	1	2	3	5	6	7	9	10	14	15
GPyGMUXn. GPIOz=	00b,01b,10b,11b	00b			01b			10b		11b	
GPyMUXn. GPIOz=	00b	01b	10b	11b	01b	10b	11b	01b	10b	10b	11b
GPIO000(IO)	EPWM1A(O)					SDAA(IO)					
GPIO001(IO)	EPWM1B(O)					SCLA(IO)					
GPIO002(IO)	EPWM2A(O)				OUTPUT XBAR1(O)	SDAB(IO)					
GPIO003(IO)	EPWM2B(O)	OUTPUT XBAR2(O)			OUTPUT XBAR2(O)	SCLB(IO)					
GPIO004(IO)	EPWM3A(O)				OUTPUT XBAR3(O)	CANTXA(O)		CANTXC(O)			
GPIO005(IO)	EPWM3B(O)		OUTPUT TXBAR3(O)			CANRXA(I)		CANRXC(I)			
GPIO006(IO)	EPWM4A(O)	OUTPUT XBAR4(O)	EXTSYNC OUT(O)		EQEP3A(I)	CANTXB(O)					
GPIO007(IO)	EPWM4B(O)		OUTPUT XBAR5(O)		EQEP3B(I)	CANRXB(I)					
GPIO008(IO)	EPWM5A(O)	CANTXB(O)	ADCSOCAO(O)		EQEP3S(IO)	UARTTXDA(O)		CANTXC(O)			
GPIO009(IO)	EPWM5B(O)	UARTTXDB(O)	OUTPUT XBAR6(O)		EQEP3I(IO)	UARTRXDA(I)					
GPIO010(IO)	EPWM6A(O)	CANRXB(I)	ADCSOCBO(O)		EQEP1A(I)	UARTTXDB(O)		CANRXC(I)			
GPIO011(IO)	EPWM6B(O)	UARTRXDB(I)	OUTPUT XBAR7(O)		EQEP1B(I)	UARTRXDB(I)					
GPIO012(IO)	EPWM7A(O)	CANTXB(O)			EQEP1S(IO)	UARTTXDC(O)					
GPIO013(IO)	EPWM7B(O)	CANRXB(I)			EQEP1I(IO)	UARTRXDC(I)					
GPIO014(IO)	EPWM8A(O)	UARTTXDB(O)				OUTPUT XBAR3(O)					
GPIO015(IO)	EPWM8B(O)	UARTRXDB(I)				OUTPUT XBAR4(O)					
GPIO016(IO)	SPISIMOA(IO)	CANTXB(O)	OUTPUT XBAR7(O)		EPWM9A(O)			SD1_D1(I)			
GPIO017(IO)	SPISOMIA(IO)	CANRXB(I)	OUTPUT XBAR8(O)		EPWM9B(O)			SD1_C1(I)			
GPIO018(IO)	SPICLKA(IO)	UARTTXDB(O)	CANRXA(I)		EPWM10A(O)			SD1_D2(I)	CANRXC(I)		
GPIO019(IO)	SPISTEAN(IO)	UARTRXDB(I)	CANTXA(O)		EPWM10B(O)			SD1_C2(I)	CANTXC(O)		
GPIO020(IO)	EQEP1A(I)		CANTXB(O)		EPWM11A(O)			SD1_D3(I)			
GPIO021(IO)	EQEP1B(I)		CANRXB(I)		EPWM11B(O)			SD1_C3(I)			
GPIO022(IO)	EQEP1S(IO)		UARTTXDB(O)		EPWM12A(O)	SPICLKB(IO)		SD1_D4(I)	CANTXC(O)		
GPIO023(IO)	EQEP1I(IO)		UARTRXDB(I)		EPWM12B(O)	SPISTEBN(IO)		SD1_C4(I)	CANRXC(I)		
GPIO024(IO)	OUTPUT XBAR1(O)	EQEP2A(I)				SPISIMOB(IO)		SD2_D1(I)	PMBUSA _SCL(IO)		
GPIO025(IO)	OUTPUT XBAR2(O)	EQEP2B(I)				SPISOMIB(IO)		SD2_C1(I)	PMBUSA _SDA(IO)		
GPIO026(IO)	OUTPUT XBAR3(O)	EQEP2I(IO)			OUTPUT XBAR3(O)	SPICLKB(IO)		SD2_D2(I)	PMBUSA _ALERT(IO)		
GPIO027(IO)	OUTPUT XBAR4(O)	EQEP2S(IO)			OUTPUT XBAR4(O)	SPISTEBN(IO)		SD2_C2(I)	PMBUSA _CTL(IO)		

(续表见下页)

表4.5 GPIO多路复用引脚（续表）

GPIO序号	GPIO选择器选择										
	0,4,8,12	1	2	3	5	6	7	9	10	14	15
GPyGMUXn. GPIOz=	00b,01b,10b,11b	00b			01b			10b		11b	
GPyMUXn. GPIOz=	00b	01b	10b	11b	01b	10b	11b	01b	10b	10b	11b
GPIO028(IO)	UARTRXDA(I)	EM1CS4N(O)			OUTPUT XBAR5(O)	EQEP3A(I)	SD2_D3(I)				
GPIO029(IO)	UARTTXDA(O)	EM1SDCKE(O)			OUTPUT XBAR6(O)	EQEP3B(I)	SD2_C3(I)				
GPIO030(IO)	CANRXA(I)	EM1CLK(O)			OUTPUT XBAR7(O)	EQEP3S(IO)	SD2_D4(I)	CANRXC(I)			
GPIO031(IO)	CANTXA(O)	EM1WEN(O)			OUTPUT XBAR8(O)	EQEP3I(IO)	SD2_C4(I)	CANTXC(O)			
GPIO032(IO)	SDAA(IO)	EM1CS0N(O)					CLB_OUTPUT XBAR1(O)				
GPIO033(IO)	SCLA(IO)	EM1RNW(O)					CLB_OUTPUT XBAR2(O)				
GPIO034(IO)	OUTPUT XBAR1(O)	EM1CS2N(O)				SDAB(IO)	CLB_OUTPUT XBAR3(O)				
GPIO035(IO)	UARTRXDA(I)	EM1CS3N(O)				SCLB(IO)	CLB_OUTPUT XBAR4(O)				
GPIO036(IO)	UARTTXDA(O)	EM1WAIT(I)				CANRXA(I)	CLB_OUTPUT XBAR5(O)	CANRXC(I)			
GPIO037(IO)	OUTPUT XBAR2(O)	EM1OEN(O)				CANTXA(O)	CLB_OUTPUT XBAR6(O)	CANTXC(O)			
GPIO038(IO)		EM1A0(O)			UARTTXDC(O)	CANTXB(O)	CLB_OUTPUT XBAR7(O)				
GPIO039(IO)		EM1A1(O)			UARTRXDC(I)	CANRXB(I)	CLB_OUTPUT XBAR8(O)				
GPIO040(IO)		EM1A2(O)				SDAB(IO)					
GPIO041(IO)		EM1A3(O)				SCLB(IO)					
GPIO042(IO)						SDAA(IO)	USB0DM(IO)				UARTTXDA(O)
GPIO043(IO)						SCLA(IO)	USB0DP(IO)				UARTRXDA(I)
GPIO044(IO)		EM1A4(O)									
GPIO045(IO)		EM1A5(O)									
GPIO046(IO)		EM1A6(O)				UARTRXDD(I)					
GPIO047(IO)		EM1A7(O)				UARTTXDD(O)					
GPIO048(IO)	OUTPUT XBAR3(O)	EM1A8(O)				UARTTXDA(O)	SD1_D1(I)				
GPIO049(IO)	OUTPUT XBAR4(O)	EM1A9(O)				UARTRXDA(I)	SD1_C1(I)				
GPIO050(IO)	EQEP1A(I)	EM1A10(O)				SPISIMOC(IO)	SD1_D2(I)				
GPIO051(IO)	EQEP1B(I)	EM1A11(O)				SPISOMIC(IO)	SD1_C2(I)				
GPIO052(IO)	EQEP1S(IO)	EM1A12(O)				SPICLK(I)	SD1_D3(I)				
GPIO053(IO)	EQEP1I(IO)	EM1D31(IO)				SPISTECN(IO)	SD1_C3(I)				
GPIO054(IO)	SPISIMOA(IO)	EM1D30(IO)			EQEP2A(I)	UARTTXDB(O)	SD1_D4(I)				
GPIO055(IO)	SPISOMIA(IO)	EM1D29(IO)			EQEP2B(I)	UARTRXDB(I)	SD1_C4(I)				
GPIO056(IO)	SPICLKA(IO)	EM1D28(IO)			EQEP2S(IO)	UARTTXDC(O)	SD2_D1(I)				
GPIO057(IO)	SPISTEAN(IO)	EM1D27(IO)			EQEP2I(IO)	UARTRXDC(I)	SD2_C1(I)				
GPIO058(IO)		EM1D26(IO)			OUTPUT XBAR1(O)	SPICLKB(IO)	SD2_D2(I)				SPISIMOA(IO)

(续表见下页)

表4.5 GPIO多路复用引脚（续表）

GPIO序号	GPIO选择器选择											
	0,4,8,12	1	2	3	5	6	7	9	10	14	15	
GPyGMUXn. GPIOz=	00b,01b,10b,11b	00b			01b			10b		11b		
GPyMUXn. GPIOz=	00b	01b	10b	11b	01b	10b	11b	01b	10b	10b	11b	
	GPIO059(IO)		EM1D25(IO)		OUTPUT XBAR2(O)	SPISTEBN(IO)	SD2_C2(I)				SPISOMIA(IO)	
	GPIO060(IO)		EM1D24(IO)		OUTPUT XBAR3(O)	SPISIMOB(IO)	SD2_D3(I)				SPICLKA(IO)	
	GPIO061(IO)		EM1D23(IO)		OUTPUT XBAR4(O)	SPISEMIB(IO)	SD2_C3(I)				SPISTEAN(IO)	
	GPIO062(IO)	UARTRXDC(I)	EM1D22(IO)		EQEP3A(I)	CANRXA(I)	SD2_D4(I)					
	GPIO063(IO)	UARTTXDC(O)	EM1D21(IO)		EQEP3B(I)	CANTXA(O)	SD2_C4(I)				SPISIMOB(IO)	
	GPIO064(IO)		EM1D20(IO)		EQEP3S(IO)	UARTRXDA(I)					SPISOMIB(IO)	
	GPIO065(IO)		EM1D19(IO)		EQEP3I(IO)	UARTTXDA(O)					SPICLKB(IO)	
	GPIO066(IO)		EM1D18(IO)			SDAB(IO)					SPISTEBN(IO)	
	GPIO067(IO)		EM1D17(IO)									
	GPIO068(IO)		EM1D16(IO)									
	GPIO069(IO)		EM1D15(IO)			SCLB(IO)					SPISIMOC(IO)	
	GPIO070(IO)		EM1D14(IO)		CANRXA(I)	UARTTXDB(O)		CANRXC(I)			SPISOMIC(IO)	
	GPIO071(IO)		EM1D13(IO)		CANTXA(O)	UARTRXDB(I)		CANTXC(O)			SPICLKC(IO)	
	GPIO072(IO)		EM1D12(IO)		CANTXB(O)	UARTTXDC(O)					SPISTECN(IO)	
	GPIO073(IO)		EM1D11(IO)	XCLKOUT(O)	CANRXB(I)	UARTRXDC(I)						
	GPIO074(IO)		EM1D10(IO)					CANTXC(O)				
	GPIO075(IO)		EM1D9(IO)					CANRXC(I)				
	GPIO076(IO)		EM1D8(IO)			UARTTXDD(O)						
	GPIO077(IO)		EM1D7(IO)			UARTRXDD(I)						
	GPIO078(IO)		EM1D6(IO)			EQEP2A(I)						
	GPIO079(IO)		EM1D5(IO)			EQEP2B(I)						
	GPIO080(IO)		EM1D4(IO)			EQEP2S(IO)						
	GPIO081(IO)		EM1D3(IO)			EQEP2I(IO)						
	GPIO082(IO)		EM1D2(IO)									
	GPIO083(IO)		EM1D1(IO)									
	GPIO084(IO)		EM1A13(O)			UARTTXDA(O)						
	GPIO085(IO)		EM1D0(IO)			UARTRXDA(I)						
	GPIO086(IO)		EM1A13(O)	EM1CAS(O)		UARTTXDB(O)						
	GPIO087(IO)		EM1A14(O)	EMIRAS(O)		UARTRXDB(I)						
	GPIO088(IO)		EM1A15(O)	EM1DQM0(O)								
	GPIO089(IO)		EM1A16(O)	EM1DQM1(O)			UARTTXDC(O)					
	GPIO090(IO)		EM1A17(O)	EM1DQM2(O)			UARTRXDC(I)					
	GPIO091(IO)		EM1A18(O)	EM1DQM3(O)			SDAA(IO)		PMBUSA _SCL(IO)	CLB_OUTPUT XBAR1(O)		
	GPIO092(IO)		EM1A19(O)	EM1BA1(O)			SCLA(IO)		PMBUSA _SDA(IO)	CLB_OUTPUT XBAR2(O)		
	GPIO093(IO)			EM1BA0(O)			UARTTXDD(O)		PMBUSA _ALERT(IO)	CLB_OUTPUT XBAR3(O)		
	GPIO094(IO)		EM1A14(O)				UARTRXDD(I)		PMBUSA _CTL(IO)	CLB_OUTPUT XBAR4(O)		
	GPIO099(IO)		EM1A15(O)			EQEP1I(IO)						
	GPIO128(IO)			UARTTXDD(O)		CANTXA(O)	SDAA(IO)					
	GPIO129(IO)			UARTRXDD(I)		CANRXA(I)	SCLA(IO)					
	GPIO130(IO)										RSTN _EXTPCB(O)	
	GPIO133(IO)						SD2_C2(I)					

4.4.2 输入X-BAR

输入X-BAR用于将任何GPIO输入路由到ADC，eCAP和ePWM外设以及外部中断（XINT），具体如图4.2所示。表4.6显示了输入X-BAR目标。有关配置输入X-BAR的详细信息，请参阅HS32F7D377PTI TRM。

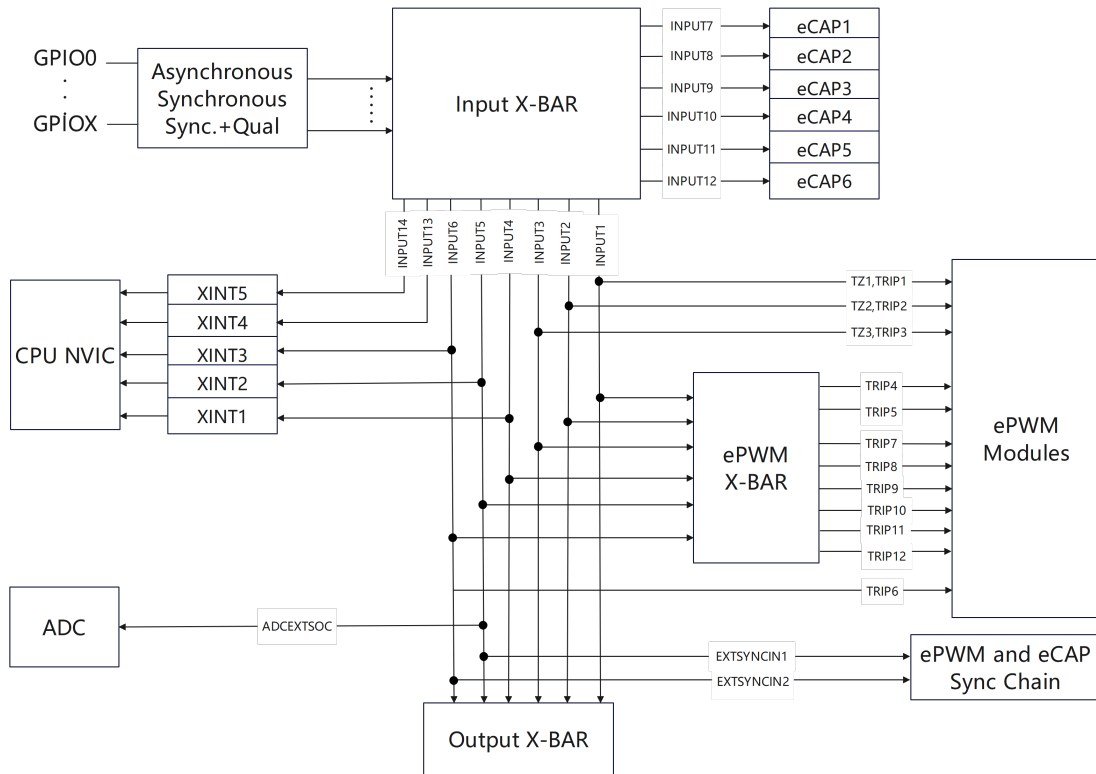


图 4.2 输入X-Bar

表 4.6 输入X-Bar目标

输入	目标
输入1	EPWM[TZ1,TRIP1]、EPWM X-BAR、输出 X-BAR
输入2	EPWM[TZ2,TRIP2]、EPWM X-BAR、输出 X-BAR
输入3	EPWM[TZ3,TRIP3]、EPWM X-BAR、输出 X-BAR
输入4	XINT1、EPWM X-BAR、输出 X-BAR
输入5	XINT2、ADCEXTSOC、EXTSYNCIN1、EPWM X-BAR、输出 X-BAR
输入6	XINT3、EPWM[TRIP6]、EXTSYNCIN2、EPWM X-BAR、输出 X-BAR
输入7	ECAP1
输入8	ECAP2
输入9	ECAP3
输入10	ECAP4
输入11	ECAP5
输入12	ECAP6
输入13	XINT4
输入14	XINT5

4.4.3 输出X-Bar和ePWM X-Bar

输出X-BAR有8个输出，可以在GPIO多路复用器上选择为OUTPUTXBARx。ePWM X-BAR有8个输出，与ePWM的TRIPx输入相连。输出X-BAR和ePWM X-BAR的源如图4.3所示。有关输出X-BAR和 ePWM X-BAR的详细信息，请参阅HS32F7D377PTI双核微控制器TRM的“交叉开关(X-BAR)”一章。

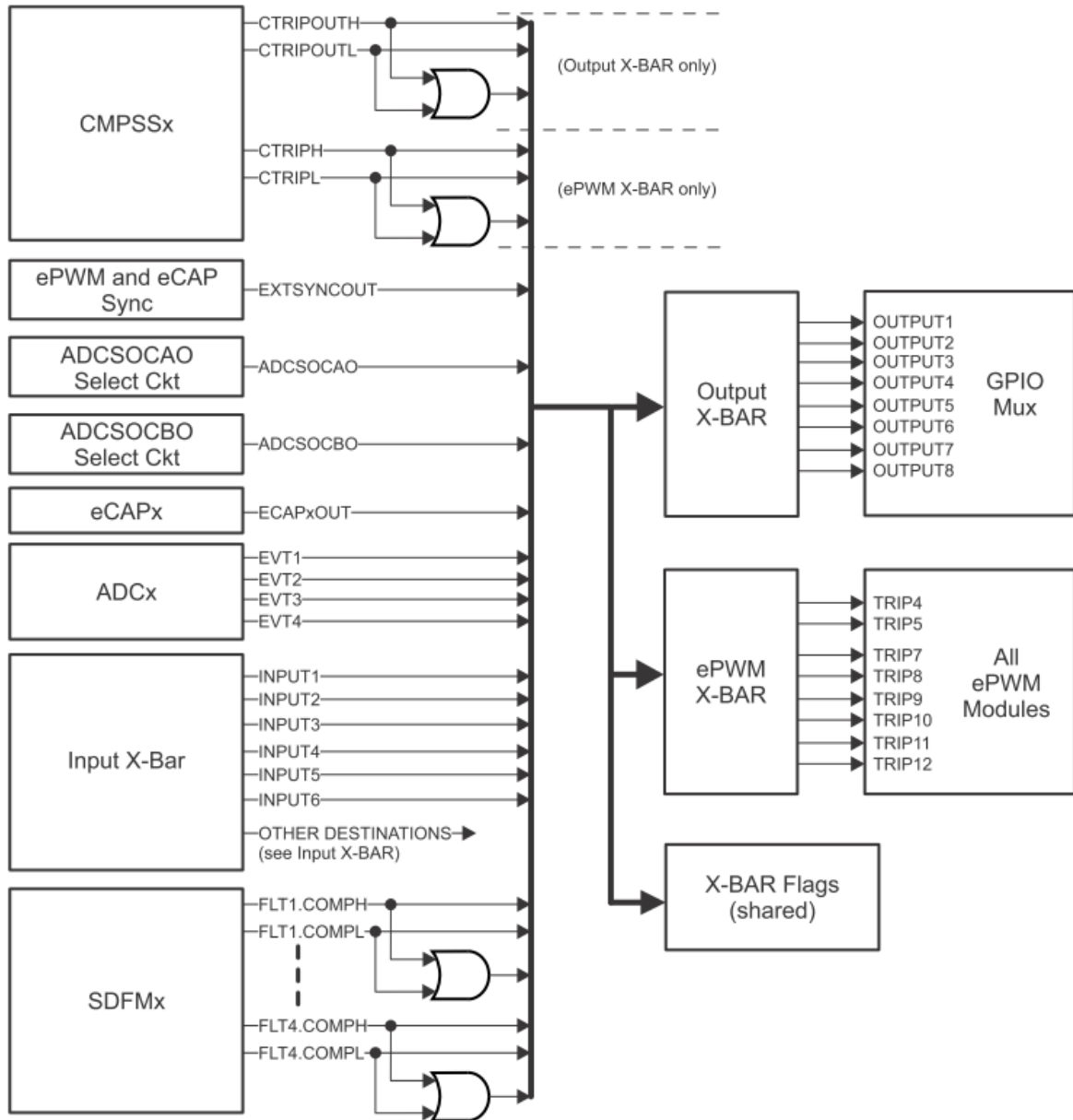


图 4.3 输出X-BAR和ePWM X-Bar

4.4.4 USB引脚多路复用

表4.7显示了备用USB功能映射的分配，可通过寄存器对他们进行配置。

USB引脚的复用，和普通GPIO一样，使用的GPIO是GPIO42和GPIO43。复用情况请参考小节“GPIO和外设说明”。

表 4.7 备用USB功能

名称	选择器位置	QFP176 Pin No.	I/O/Z ⁽¹⁾	描述
GPIO42	0,4,8,12	130	I/O	通用输入/输出42
SDAA	6		I/O	I2C-A数据漏极开路双向端口
USB0DM	7		I/O	USB PHY差分数据
UARTTXDA	15		O	UART-A发送数据
GPIO43	0,4,8,12	131	I/O	通用输入/输出43
SCLA	6		I/O	I2C-A时钟漏极开路双向端口
USB0DP	7		I/O	USB PHY差分数据
UARTRXDA	15		I	UART-A接收数据

- (1) I=输入，O=输出，OD=漏极开路，Z=高阻抗
- (2) USB的引脚复用和普通GPIO配置一样。
- (3) GPIO42, GPIO43,也可以通过MUX配置成其他功能。

4.4.5 高速SPI引脚多路复用

该器件上的SPI模块具有高速模式。为了实现尽可能高的速度，在每个SPI的单一GPIO多路复用器选项上使用了特殊的GPIO配置。

如需选择启用SPI高速模式的多路复用器选项，请配置GPyGMUX和GPyMUX寄存器，如表4.8所示。

表 4.8 高速SPI的GPIO配置

GPIO	SPI信号	多路复用器配置	
SPIA			
GPIO58	SPISIMOA	GPBGMUX2[21:20]=11b	GPBMUX2[21:20]=11b
GPIO59	SPISOMIA	GPBGMUX2[23:22]=11b	GPBMUX2[23:22]=11b
GPIO60	SPICLKA	GPBGMUX2[25:24]=11b	GPBMUX2[25:24]=11b
GPIO61	SPISTEAN	GPBGMUX2[27:26]=11b	GPBMUX2[27:26]=11b
SPIB			
GPIO63	SPISIMOB	GPBGMUX2[31:30]=11b	GPBMUX2[31:30]=11b
GPIO64	SPISOMIB	GPCGMUX1[1:0]=11b	GPCMUX1[1:0]=11b
GPIO65	SPICLKB	GPCGMUX1[3:2]=11b	GPCMUX1[3:2]=11b
GPIO66	SPISTEBN	GPCGMUX1[5:4]=11b	GPCMUX1[5:4]=11b
SPIC			
GPIO69	SPISIMOC	GPCGMUX1[11:10]=11b	GPCMUX1[11:10]=11b
GPIO70	SPISOMIC	GPCGMUX1[13:12]=11b	GPCMUX1[13:12]=11b
GPIO71	SPICLKC	GPCGMUX1[15:14]=11b	GPCMUX1[15:14]=11b
GPIO72	SPISTECN	GPCGMUX1[17:16]=11b	GPCMUX1[17:16]=11b

4.4.6 未使用引脚的连接

对于不需要使用器件所有功能的应用，表4.9列出了对任何未使用引脚的可接受条件。当表中列出了多个选项，则任何选项都可接受。表中未列的引脚必须根据节4.2.3进行连接。

表 4.9 未使用引脚的连接

信号名称	可接受的做法
模拟	
VREFH _x	连接至VDDA
VREFLO _x	连接至VSSA
ADCIN _x	无连接或连接至VSSA
数字	
GPIO _x	无连接（启用内部上拉的输入模式） 无连接（禁用内部上拉的输出模式） 上拉或下拉电阻器（任意值电阻器，输入模式，禁用内部上拉）
X1	连接至VSS
X2	无连接
TCK	无连接或上拉电阻器
TDI	无连接或上拉电阻器
TDO	无连接
TMS	无连接
TRST _n	下拉电阻器（2.2kΩ或更小）
VREGENZ	连接至VDDIO。不支持VREG。注：此Pin位置，在目前的设计中是VDDIO，所以默认连接到VDDIO。
ERRORSTS	无连接
FLT1	注：跟TI不同的地方是，FLT1的位置被GPIO替代，可以无连接，也可以当GPIO用
FLT2	注：跟TI不同的地方是，FLT1的位置被GPIO替代，可以无连接，也可以当GPIO用
电源和地	
VDD	所有VDD引脚必须按照2.2.3节所述进行连接。
VDDA	如果未使用专用模拟电源，则连接到VDDIO。
VDDIO	所有VDDIO引脚必须按照2.2.3节所述进行连接。
VDD3VFL	必须连接到VDDIO
VDDOSC	必须连接到VDDIO
VSS	所有VSS引脚必须连接到电路板接地。
VSSA	如果未使用专用模拟接地，则连接到VSS。
VSSOSC	如果未使用外部晶体，则该引脚必须连接到电路板接地。

5 规格

5.1 绝对最大额定值

表 5.1 绝对最大额定值⁽¹⁾⁽²⁾

		最小值	最大值	单位
电源电压	V_{DDIO} 相对于 V_{SS}	-0.5	4.6	V
	V_{DD3VFL} 相对于 V_{SS}	-0.5	4.6	
	V_{DDOSC} 相对于 V_{SS}	-0.5	4.6	
	V_{DD} 相对于 V_{SS}	-0.5	1.8	
模拟电压	V_{DDA} 相对于 V_{SSA}	-0.5	4.6	V
输入电压	V_{IN} (3.3 V)	-0.5	4.6	V
输出电压	V_O	-0.5	4.6	V
输出电流	数字输出 (每个管脚), I_{OUT}	-20	20	mA
工作结温	T_J	-40	125	°C
储存温度	T_{Istg}	-65	150	°C

- (1) 超出绝对最大额定值下列出的压力，可能会对设备造成永久性损坏。表中仅列出压力范围，未涉及设备在这些条件或第3.3节所述的任何其他条件下的功能特性。长时间暴露在绝对最大额定条件下可能会影响设备的可靠性。
- (2) 除非另有说明，否则所有电压值均与 V_{SS} 有关。
- (3) 每个引脚的连续钳位电流为 ± 2 mA。请勿在此条件下连续运行，因为 V_{DDIO}/V_{DDA} 电压可能会在内部上升并影响其他电气规格。
- (4) 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整体寿命缩短。

5.2 静电放电(ESD)等级

表 5.2 静电放电等级

		数值	单位
V_{ESD}	人体模型 (HBM), 符合EIA/JESD22-A114-A标准	±2000	V
	充电设备模型 (CDM), 符合JESD22-C101 或ANSI/ESDA/ JEDEC JS-002标准	±500	

5.3 建议工作条件

表 5.3 建议工作条件

		最小值	标称值	最大值	单位
电源电压	$V_{DDIO}^{(1)}$	3.13	3.3	3.47	V
	V_{DD}	1.04	1.1	1.16	
电源地	VSS		0		V
模拟电源电压	V_{DDA}	3.13	3.3	3.47	V
模拟电源地	V_{SSA}		0		V
结温	T_J	-40		125	°C

(1) V_{DDIO} , V_{DD3VFL} 和 V_{OSC} 需要保证相互之间的压差不能超过0.3V。

5.4 功耗

本节列出的电流值是在给定测试条件测试得到的，而不是可能的绝对最大值。

表 5.4 300MHz SYSCLK时的器件电流消耗和功耗

模式	测试条件	I _{DD} /mA			I _{DDIO} /mA			I _{DDA} /mA		
		典型	高温	最大	典型	高温	最大	典型	高温	最大
工作	·代码不在RAM中运行。(4) ·所有I/O引脚未连接。 ·未使能的外设时钟被禁用。 ·FLASH被读取并处于活动状态。 ·XCLKOUT使能为SYSCLK/4。	146	213	219	9	13.2	13.5	116	119	121
idle	·CPU 1和CPU 2均处于空闲模式。 ·XCLKOUT已关闭。	60	127	131	4.2	6.8	9.4	5.5	5.4	10.5
stand by(5)	·CPU 1和CPU 2均处于待机模式。 ·XCLKOUT已关闭。	50	118	121	4.0	6.2	10.2	5.5	5	10.9
Flash擦写(6)	·CPU 1从RAM运行。 ·CPU 2从Flash运行。 ·所有I/O引脚均未连接。 ·外设时钟被禁用。 ·CPU 1正在执行Flash擦写。 ·CPU 2正在访问闪存位置以保持存储在活动状态。 ·XCLKOUT关闭	78	142	147	4	9.6	12	5.9	5.8	7.7

(1) I_{DDIO}电流取决于I/O引脚上的电气负载。实际器件的电流将随具体应用的代码和引脚配置而变化。

(2) 典型: V_{nom}, 30°C

(3) 高温: V_{nom}, 125°C; 最大: V_{max}, 125°C

(4) 正常工作模式:

CPU1:

- (a) CAN a-b/UART c-d/I2C a-b: 初始化后, 进入LOOP-BACK工作
- (b) UART a: 输出打印信息
- (c) SPI/USB: Active
- (d) SDFM1-4: Active
- (e) EPWM1-12: 产生一个400K的输出到24个PIN管脚上
- (f) 所有ADC: 初始化后工作, 并持续转换数据
- (g) 所有DAC: 输出一个10KHZ的波形
- (h) CMPSS1-8: 初始化后保持活跃状态

CPU2:

- (a) URAT b: 输出打印信息
- (b) TFU: 计算一个COSINE
- (c) FPU: Active

(5) CPU 1进入STANDBY模式之前, CPU 2必须进入STANDBY模式。

(6) Flash编程时如果欠压, 可能会损坏闪存数据。使用备用电源(如USB编程器)的编程环境必须能够为器件和其他

系统组件提供额定电流，并具有足够的余量，以避免电源欠压情况。

5.4.1 模块功耗

测试关断外设时钟条件下的功耗，单一模块表示只开启该模块中的一个模组CLK，多模块表示开启该模块中的所有模组CLK。例如SPI，单模块表示只开启SPI-a，多模块表示开启SPI-a/b/c。所有测试在 V_{nom} 和常温下测得

表 5.5 模块功耗

外设模块	单一模块	多模块
ADC	1mA	4mA
CAN	5.8mA	7.7mA
CLB	6mA	23.5mA
CMPSS	0.3mA	2.8mA
CPUTIMER	0.1mA	0.9mA
DAC	0.2mA	0.5mA
ECAP	1.9mA	6.1mA
EMIF1	0.3mA	
EPWM1-EPWM4	4.2mA	10mA
EPWM5-EPWM12	4.2mA	15mA
I2C	1mA	2.8mA
UART(C/D)	0.4mA	0.9mA
SDFM	4.3mA	8.6mA
SPI	1.3mA	4mA
USB	1.3mA	

5.5 电气特性

表 5.6 电气特性

参数		测试条件	最小值	标称值	最大值	单位	
V_{OH}	输出高电平	@ $I_{OH}(\text{min})$	2.4			V	
V_{OL}	输出低电平	@ $I_{OL}(\text{min})$			0.4	V	
I_{OH}	高电平输出电流(??)	@ $V_{OH}=2.4V$	9.5		60.5	mA	
I_{OL}	低电平输出电流(??)	@ $V_{OL}=0.4V$	5.5		33.1	mA	
V_{IH}	高电平输入电压	GPIO0-GPIO99, GPIO128-GPIO133			V_{DDIO}	V	
		XTALIN, XTALOUT		$0.65 * V_{DDIO}$	$V_{DDIO} + 0.3$		
		其他管脚		2	$V_{DDIO} + 0.3$		
V_{IL}	低电平输入电压	XTALIN, XTALOUT			$0.35 * V_{DDIO}$	V	
		其他管脚		-0.3	0.8		
$I_{pull\downarrow}$	输入电流(??)	GPIO0-GPIO99, GPIO128-GPIO133	$V_{PAD}=0$	13	24	36	μA
		其他带下拉的数字输入		12	21	32	
$I_{pull\uparrow}$	输入电流(??)	GPIO0-GPIO99, GPIO128-GPIO133	$V_{PAD}=V_{DDIO}$	18	29	43	μA
		带上拉的数字输入		15	25	37	
I_L	输入漏电流				± 10	μA	
C_I	输入电容			2		pF	
$V_{DDIO-POR}$	V_{DDIO} 上电复位电压			1.75		V	

5.6 热阻特性

表 5.7 热阻特性

		°C/W ⁽¹⁾	风速/(lfm)
$R\theta_{JC}$	结壳热阻	13.745	N/A
$R\theta_{JB}$	结板热阻	6.1025	N/A
$R\theta_{JA}$ (高k PCB)	结至自由空气热阻	17.805	0
$R\theta_{JMA}$	结至流动空气热阻	15.507	150
		14.457	250
		13.519	500

- (1) 上述数据是基于JEDEC定义的2S2P系统（ $R\theta_{JC}$ 值除外，该值基于JEDEC定义的1S0P系统）测试得到的，并会随环境和具体应用而变化。
- (2) 采取的是外引脚与PCB接触位置最高温与结温之间的温差计算，HS32F7D377PTI的测试采用ePAD与PCB接触位置的最高温，因为此为散热路径经过的区域，外引脚处无散热路径，远离芯片热源区。

5.7 系统

5.7.1 上电时序

5.7.1.1 信号输入要求

在上电之前，任何digital pin上不能有比 V_{DDIO} 高0.3V的电压，任何analog pin (包括 V_{REFHI}) 上不能有比 V_{DDA} 高0.3V的电压。

5.7.1.2 V_{DDIO} , V_{DDA} , V_{DD3VFL} 和 V_{DDOSC} 要求

3.3V的供电必须一起上电，且保证相互之间的压差不能超过0.3V。

5.7.1.3 V_{DD} 要求

V_{DD} 必须在3.3V power到达1.1V之前先到达0.7V。 V_{DDOSC} 和 V_{DD} 必须同时供电断电。

5.7.1.4 上电时间

所有的供电电压的上电时间（到达满幅值）不能超过10ms。

5.7.1.5 电源监测

外部电源电压监测器（SVS）可用于监测3.3V和1.1V的电压，并且在电源电压超出工作规范时将XRSn置为低电压。

5.7.1.6 V_{DD} , V_{DD3VFL} 上下电时序要求

V_{DD} 必须在 V_{DD3VFL} 到达1.5V之前先到达0.7V； V_{DD} 必须在 V_{DD3VFL} 下降到1.5V之后开始下电。

5.7.2 复位时序

XRSn是器件的复位引脚。XRSn只作为输入。需要在XRSn和 V_{DDIO} 之间放置一个10~20k Ω 的电阻，在XRSn和 V_{SS} 之间放置一个1~2.2 μ F的电容器，用于过滤噪声。如图5.1所示。

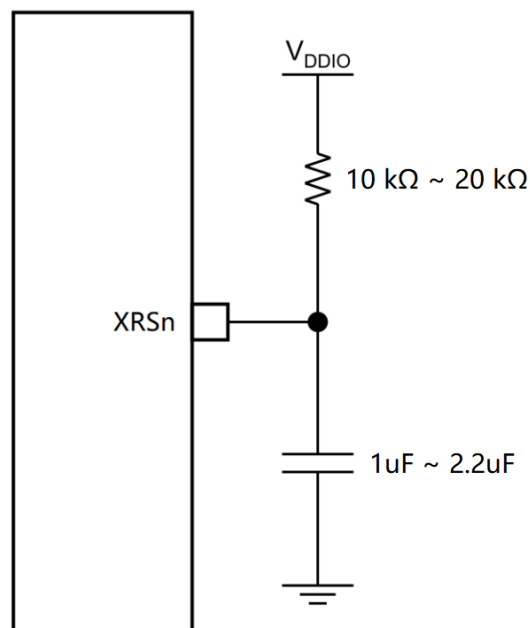


图 5.1 复位电路

5.7.3 复位源

器件复位源有：XRSn、WDRSn、NMIWDRSn以及SYSRSn。

5.7.4 时钟

5.7.4.1 时钟源

器件一共有4个可供选择的时钟源，具体如表5.8和图5.2所示：

表 5.8 可供选择的时钟源及对应模块

时钟源	模块	描述
INTOSC1	Watchdog Main PLL CPU Timer 2	Internal oscillator 1 内置的 10MHz oscillator
INTOSC2	Main PLL Auxiliary PLL CPU Timer 2	Internal oscillator 2 内置的 10MHz oscillator
XTAL	Main PLL Auxiliary PLL CPU Timer 2	通过X1和X2连接的外部crystal/resonator或者通过X1连接的单端时钟
AUXCLKIN	Auxiliary PLL CPU Timer 2	单端3.3V的时钟源GPIO133/AUXCLKIN pin应当用来提供输入时钟

(1) 在复位阶段，INTOSC2是system PLL和auxiliary PLL的默认参考时钟。

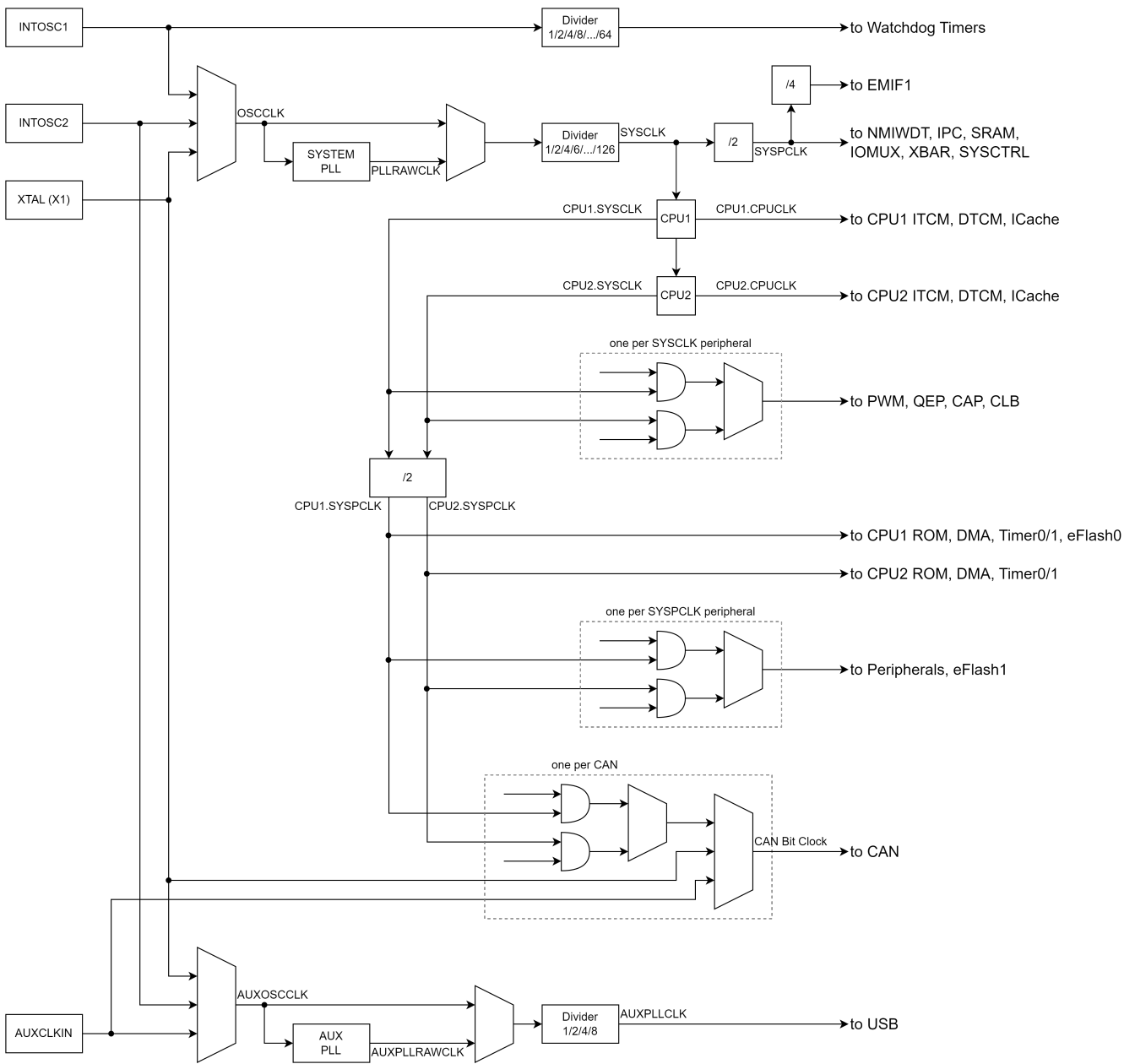


图 5.2 时钟结构图

5.7.4.2 时钟频率

5.7.4.2.1 输入时钟频率

表 5.9 输入时钟频率

名称	描述	最小	最大	单位
f_{X1}	外部oscillator	2	20	MHz
f_{AUXI}	外部oscillator	2	20	MHz

表 5.10 X1推荐工作条件

名称	描述	最小	最大	单位
X1 V_{IL}	有效的低电平输入电压	-0.3	0.8	V
X1 V_{IH}	有效的高电平输入电压	2.0	$V_{DDIO}+0.3$	V

表 5.11 PLL锁定时间

名称	描述	最小	正常	最大	单位
t(PLL)	Lock time, Main PLL		20		μs
t(USB)	Lock time, Auxilary PLL		20		μs

5.7.4.2.2 X1时序要求

表 5.12 X1时序要求

名称	最小值	最大值	单位
$t_{f(X1)}$ 下降时间,X1		15	ns
$t_{r(X1)}$ 上升时间,X1		15	ns
$t_{w(X1L)}$ 脉冲持续时间, X1低电平占 $t_{c(X1)}$ 的百分比	45%	55%	
$t_{w(X1H)}$ 脉冲持续时间, X1高电平占 $t_{c(X1)}$ 的百分比	45%	55%	

5.7.4.2.3 AUXCLKIN时序要求

表 5.13 AUXCLKIN时序要求

名称	最小值	最大值	单位
$t_{f(AUXI)}$ 下降时间, AUXCLKIN		15	ns
$t_{r(AUXI)}$ 上升时间, AUXCLKIN		15	ns
$t_{w(AUXL)}$ 脉冲持续时间, AUXCLKIN低电平占 $t_{c(XCI)}$ 的百分比	45%	55%	
$t_{w(AUXH)}$ 脉冲持续时间, AUXCLKIN高电平占 $t_{c(XCI)}$ 的百分比	45%	55%	

5.7.4.2.4 内部时钟频率

表 5.14 内部时钟频率

名称	描述	最小	正常	最大	单位
f_{SYSCLK}	频率, 器件 (系统) 时钟	2		300	MHz
$f_{PLLRAWCLK}$	频率, system PLL 输出时钟 (在SYSCLK分频之前)	160		300	MHz
$f_{AUXPLLRAWCLK}$	频率, auxiliary PLL 输出时钟 (在AUXCLK分频之前)	160		300	MHz
f_{AUXCLK}	频率, AUXPLLCLK	2	48	48	MHz
f_{OSCCLK}	频率, OSCCLK (INTOSC1 或 INTOSC2 或 XTAL 或 X1)	参考对应时钟			MHz
$f_{SYSPCLK}$	频率, SYSPCLK	1		150	MHz

5.7.4.2.5 输出时钟频率

表 5.15 输出时钟频率

名称	描述	最小	最大	单位
f_{XCO}	频率, XCLKOUT		50	MHz

5.7.4.3 输入时钟和PLL

除了内部OSC之外，器件还支持多种外部时钟源，图5.3至图5.5展示了外接无源晶振、有源晶振以及AUXCLKIN的推荐方法。

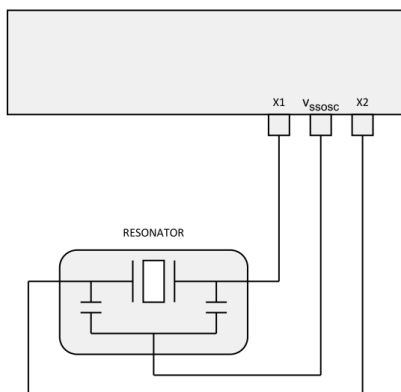


图 5.3 将谐振器连接到HS32F7D377PTI设备

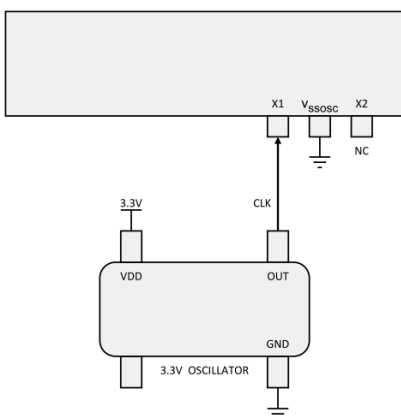


图 5.4 将振荡器连接到HS32F7D377PDI器件的引脚X1/X2

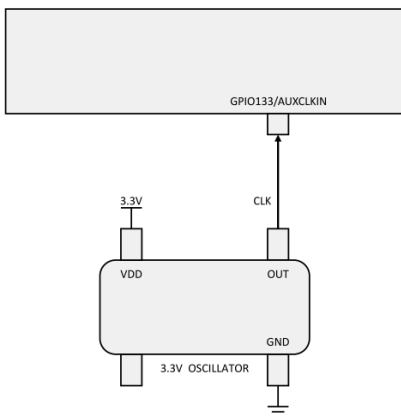


图 5.5 将振荡器连接到HS32F7D377PDI器件的引脚AUXCLKIN

5.7.4.4 内部OSC

为了减少板级制作成本和应用开发时间，器件包含2个独立的内部OSC，称为INTOSC1和INTOSC2。2个OSC默认都会上电工作。INTOSC2是OSCCLK的默认选择，INTOSC1是备份时钟源。INTOSC1也可以被手动配置成OSCCLK。表5.16展示了内部OSC的特性。

表 5.16 内部OSC特性

名称	描述	测试条件	最小	正常	最大	单位
f_{INTOSC}	频率, INTOSC1和INTOSC2		9.9	10.0	10.1	MHz
$f_{INTOSC-STABILITY}$	室温下的频率稳定性	30°C, Normal V_{DD}				
	过压下的频率稳定性	30°C				
	频率稳定性		-1.0%		1.0%	
$f_{INTOSC-ST}$	启动和稳定时间				10	μs

5.7.5 eFlash参数

eFlash是一块可多次擦除的非易失存储器，主要用于存放可执行代码和一些静态数据，允许CPU直接通过总线从eFlash中取指执行。

eFlash的特性包括：

- 特定的工作电压范围
 - 3.0V V_{DD} : 1.5V-3.6V
 - 1.1V V_{DD11} : 0.9V-1.21V
- 128K×64位闪存增强版
 - 16块NVR扇区
 - 2块冗余扇区
- 输入×64/输出×64可配置，CMOS超闪EEPROM技术ESF3-40工艺
 - 扇区耐久性：10,000次循环
 - 数据保留期超过10年
- 低功耗
 - 工作时读数据电流：
 - * 68 μ A/MHz @ V_{DD11} =0.99V-1.21V
 - * 82.5 μ A/MHz @ V_{DD11} =0.9V-1.21V
 - 工作时写数据电流：3.1mA
 - 工作时擦除数据电流：3.1mA
 - 待机电流：195 μ A
 - 深度睡眠电流：5 μ A
- 小扇区擦除能力
 - 512×64 bits/扇区
- 快速读取访问时间
 - 25ns @ V_{DD11} =0.99V-1.21V
 - 50ns @ V_{DD11} =0.9V-1.21V
- 快速擦除和编程时间
 - 扇区擦除：最长10ms
 - 芯片擦除：最长10ms
 - 编程：典型值10 μ s

表 5.17 eFlash基本数据参数

Parameter	default	$t_{clk}=2*t_{sysclk}$	t_{min}	t_{max}
t_{RT}	30	t_{min}/t_{clk}	500ns	
t_{RHR}	300	t_{min}/t_{clk}	5 μ s	
t_{DPDH}	300	t_{min}/t_{clk}	5 μ s	
t_{DPDSR}	6	t_{min}/t_{clk}	100ns	
t_{CRC_LVCTL}	1	t_{min}/t_{clk}	8ns	
t_{ACC}	3	t_{min}/t_{clk} , minimum 3	25ns	
t_{ACC_LVCTL}	3	t_{min}/t_{clk} , minimum 3	50ns	
t_{ACC_NVR}	12	t_{min}/t_{clk} , minimum 3	200ns	
t_{PROG}	45	$(t_{min}/t_{clk}, t_{max}/t_{clk})$	8 μ s	10 μ s
t_{PGS}	40	t_{min}/t_{clk}	8 μ s	
t_{ADS}	1	t_{min}/t_{clk}	15ns	
t_{ADH}	1	t_{min}/t_{clk}	15ns	
t_{PGH}	1	t_{min}/t_{clk}	15ns	
t_{NVS}	240	t_{min}/t_{clk}	4 μ s	
t_{WS}	1	t_{min}/t_{clk}	5ns	
t_{RW}	6	t_{min}/t_{clk}	100ns	
t_{RCV_PROG}	25	t_{min}/t_{clk}	5 μ s	
t_{RCV_SECER}	250	t_{min}/t_{clk}	50 μ s	
t_{RCV_CHIPER}	1000	t_{min}/t_{clk}	200 μ s	
t_{SEC_ERASE}	45000	$(t_{min}/t_{clk}, t_{max}/t_{clk})$	8ms	10ms
t_{CHIP_ERASE}	45000	$(t_{min}/t_{clk}, t_{max}/t_{clk})$	8ms	10ms
t_{MS}	300	t_{min}/t_{clk}	5 μ s	
t_{CFH}	2	t_{min}/t_{clk}	30ns	
t_{CFL}	300	t_{min}/t_{clk}	5 μ s	
t_{CONFEN}	1	t_{min}/t_{clk}	15ns	

- (1) 计算时，时间单位要一致
- (2) $(t_{CRC_LVCTL}+t_{ACC_LVCTL}+1)*t_{clk} \geq 50ns$
- (3) $(t_{CRC_LVCTL}+1)*t_{clk} \geq 8ns$
- (4) $(t_{ACC}+1)*t_{clk} \geq 25ns$

5.7.6 JTAG

调试接口（JTAG）允许软件设计人员调试和跟踪他们的嵌入式软件。调试功能可以通过JTAG/串行线调试访问端口，使用行业标准调试工具进行控制。跟踪端口允许捕获数据以进行记录和分析。

调试模块结构框图如图5.6所示。

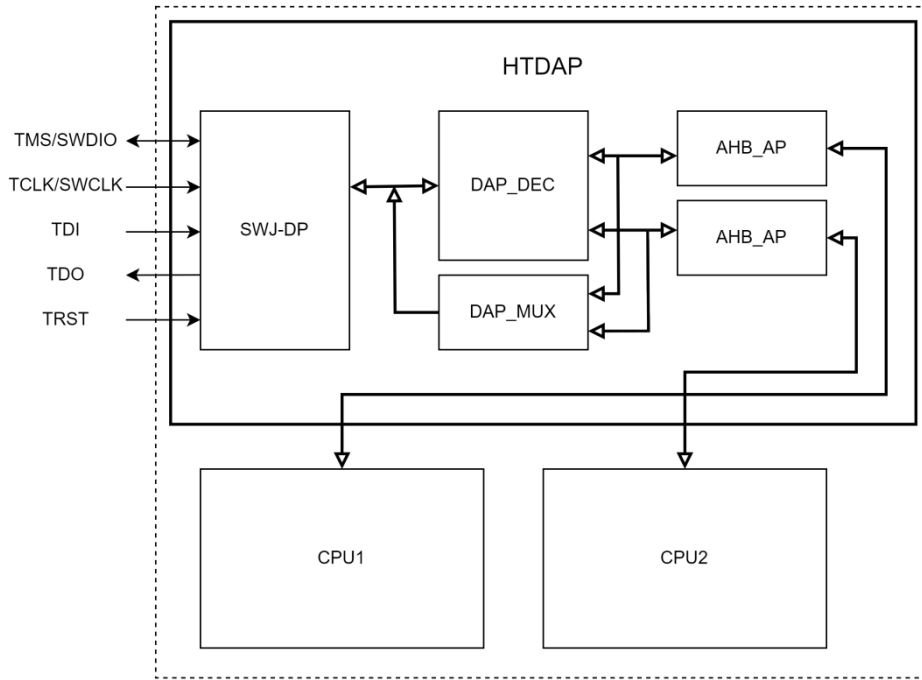


图 5.6 调试模块结构框图

5.7.6.1 调试端口引脚

表 5.18 JTAG调试端口引脚

引脚名称	JTAG调试端口		SW调试端口		引脚分配
	类型	说明	类型	说明	
JTMS/SWDIO	I	JTAG测试模式选择	IO	测试数据输入/输出	80
JTCLK/SWCLK	I	JTAG测试时钟	I	测试时钟	81
JTDI	I	JTAG测试数据输入	—	—	77
JTDO	O	JTAG测试数据输出	—	—	78
JTRST	I	JTAG测试复位	—	—	79

调试接口可以配置为：

- 一个5针标准JTAG调试端口(JTAG-DP)
- 一个2针（时钟+数据）调试端口(SW-DP)

这两种模式共用相同的IO引脚，因此两者互斥。五个IO由硬件在调试备用功能模式下配置。

5.7.6.2 JTAG-DP

JTAG-DP设计了一个基于IEEE 1149.1-1990的TAP状态机（TAPSM），状态机如图5.7所示。状态机控制两个扫描通道，一个与指令寄存器（IR）相关，一个与多个数据寄存器（DR）相关。

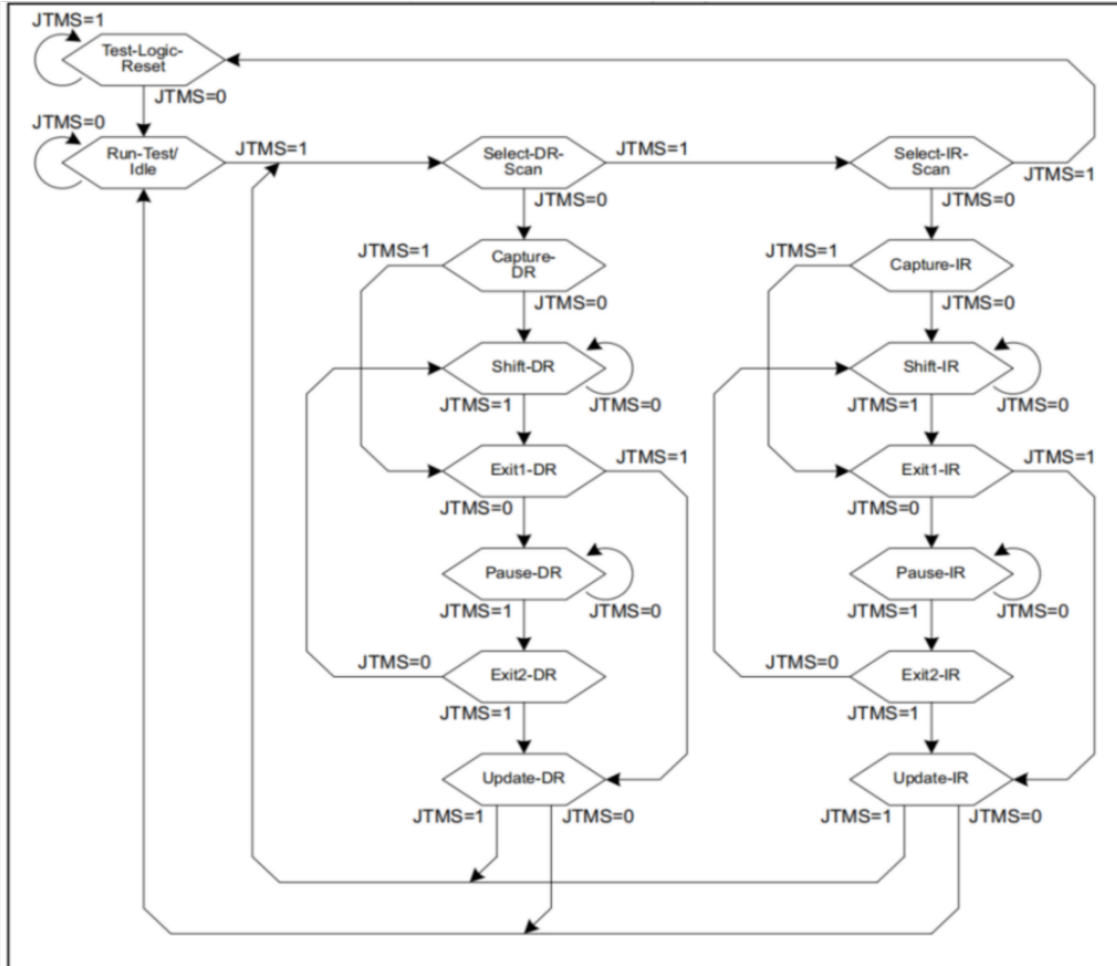


图 5.7 JTAG TAP状态机

当TAPSM通过Capture-IR状态时，0b0001被传送到指令寄存器（IR）扫描通道。IR扫描通道连接在JTDI和JTDO之间。

当TAPSM处于Shift-IR状态时，IR扫描通道在JTCK的每个上升沿移位一位，即在第一个节拍上：

- IR扫描通道的LSB是JTDO上的输出。
- IR扫描通道的位[n]被传送到位[n-1]。
- JTDI上的值会传送到IR扫描通道的MSB中。

当TAPSM通过Update-IR状态时，扫描进IR扫描通道中的值被传送到指令寄存器。

当TAPSM通过Capture-DR状态时，值从其中一个数据寄存器传送到其中一个DR扫描通道，DR扫描通道连接在JTDI和JTDO之间。

指令寄存器中保存的值决定了数据寄存器和相关的DR扫描通道的选择。

当TAPSM处于Shift-DR状态时，数据进行移位，与处于Shift-IR状态时的IR移位情形相同。

当TAPSM通过Update-DR状态时，扫描进DR扫描通道中的值被传送到所选择的数据寄存器中。

当TAPSM处于Run-Test/Idle状态时，不会发生特殊操作。IDCODE 指令加载在IR中。

JTRST信号在激活后将状态机异步重置为Test-Logic-Reset状态。

5.7.6.3 SW-DP

SW-DP是一个双引脚串行接口（SWCLK从主机到目标的时钟；SWDIO双向串行数据），它使用基于数据包的协议来读取或写入寄存器。该协议要求主机和目标之间的通信过程如下：

串行数据传输与时钟同步，首先传输LSB。如图5.8所示，传输过程包括三个阶段：

- 主机发送的数据包请求（8位）；
- 目标发送的确认响应（3位）；
- 主机（写操作时）或目标（读操作时）发送的传输数据（33位）。

只有在确认响应为“OK”时才进行数据传输。

在每一个传输阶段之间，如果数据传输方向发生改变，则插入一个单个时钟周期的周转时间。

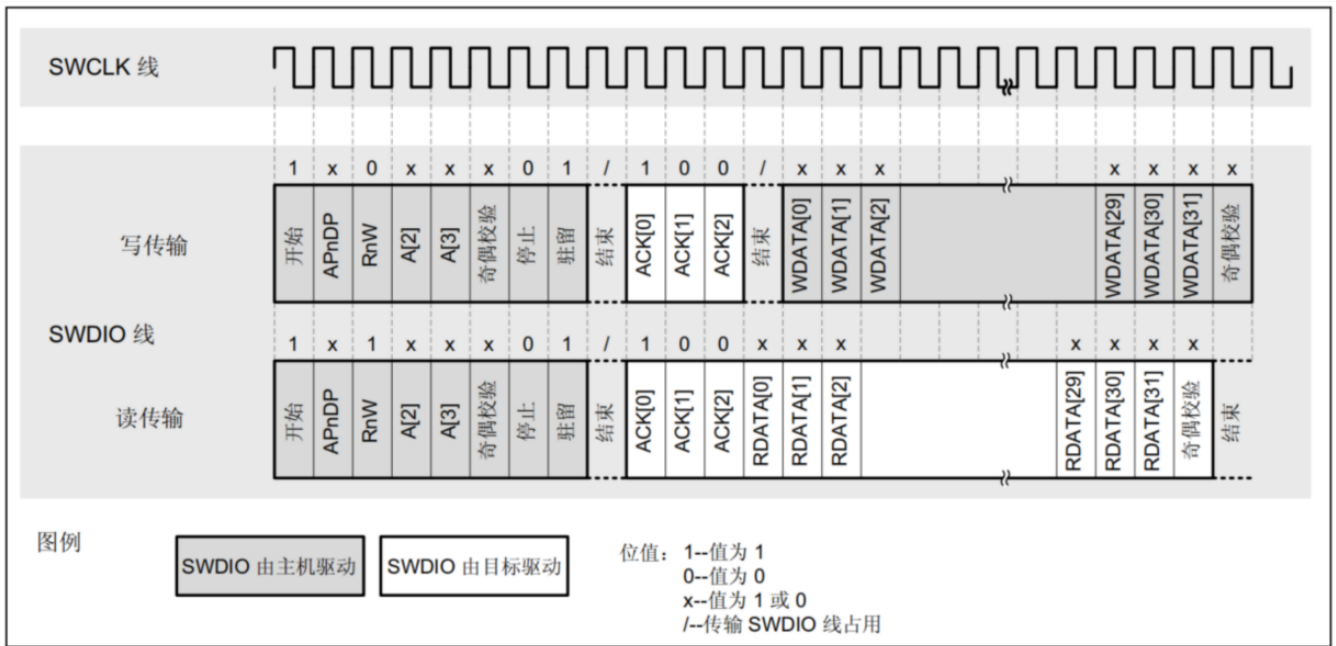


图 5.8 SWD成功的数据传输过程

如果目标响应为FAULT或WAIT ACK，则数据传输过程取消，除非使能了上溢检测，在这种情况下，数据会被目标忽略（写操作时）或不接受指令控制（读操作时）。

当引线第一次接通时，必须由主机生成引线复位信号，否则会出现协议错误。引线复位信号由50或以上个SWDIO为高电平的 SWCLK周期和紧随的两个SWDIO为低电平的SWCLK周期组成。

5.7.7 GPIO电气数据与时序

外设信号与通用输入/输出(GPIO)信号多路复用。复位时，GPIO引脚配置为输入。对于特定的输入，用户还能选择输入限定周期的数量来滤除不必要的噪声干扰。

GPIO模块包含输出X-BAR，其允许将各种内部信号路由到GPIO多路复用器位置中的GPIO上，并表示为 OUTPUTXBARx。GPIO模块还包含输入X-BAR，用于将来自任何GPIO输入的信号路由到不同的IP块，例如ADC、eCAP、ePWM和外部中断。

5.7.7.1 GPIO输出时序

节5.7.7.1.1显示了通用输出开关特征。图5.9显示了通用输出时序。

5.7.7.1.1 通用输出开关特征

表 5.19 通用输出开关特征

在推荐的工作条件下（除非另有说明）

参数	测试条件		最大值	单位
$T_{r(GPO)}$ 上升时间，GPIO从低电平切换至高电平	$F_{PWM}=10\text{MHz}$	$C_L=43\text{pf}$	14.8	ns
		$C_L=30\text{pf}$	12.4	ns
		$C_L=10\text{pf}$	7.4	ns
	$F_{PWM}=25\text{MHz}$	$C_L=43\text{pf}$	13	ns
		$C_L=30\text{pf}$	10.2	ns
		$C_L=10\text{pf}$	7.4	ns
$T_{f(GPO)}$ 下降时间，GPIO从高电平切换至低电平	$F_{PWM}=10\text{MHz}$	$C_L=43\text{pf}$	15.6	ns
		$C_L=30\text{pf}$	12.8	ns
		$C_L=10\text{pf}$	7.8	ns
	$F_{PWM}=25\text{MHz}$	$C_L=43\text{pf}$	10.2	ns
		$C_L=30\text{pf}$	9.6	ns
		$C_L=10\text{pf}$	7	ns

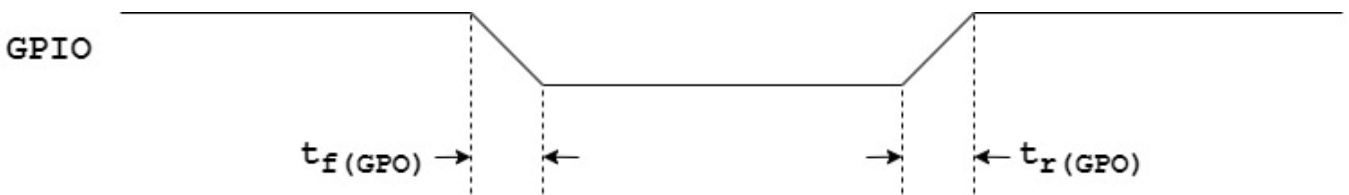


图 5.9 通用输出时序

5.7.7.2 通用输入开关特征

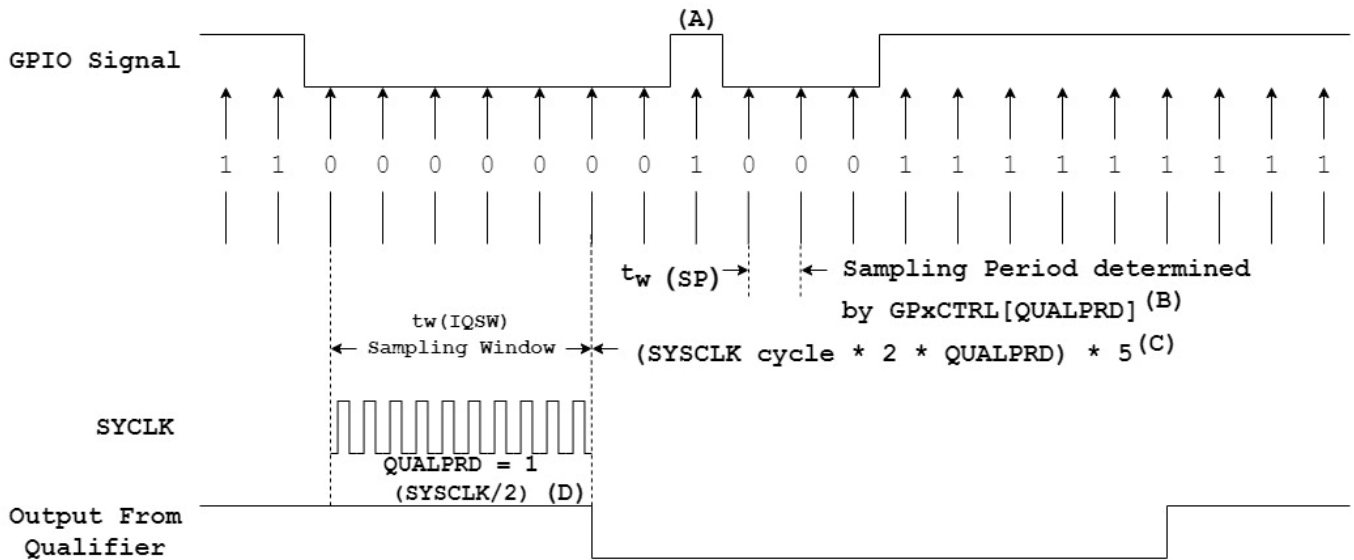
节5.7.7.2.1显示了通用输入时序要求。图5.10显示了采样模式。

5.7.7.2.1 通用输入时序要求

- (1) “n”代表由GPxQSELn寄存器定义的合格样片的数量。
- (2) 对于 $t_w(GPI)$ ，对低电平有效信号在 V_{IL} 至 V_{IL} 之间测量脉宽，而高电平有效信号，在 V_{IL} 至 V_{IH} 之间测量脉宽。

表 5.20 通用输入开关特征

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



- (A) 输入限定符将忽略此短时脉冲波干扰。QUALPRD位字段指定了限定采样周期。该位字段可以在00至0xFF之间变化。如果QUALPRD=00, 那么采样周期为1个SYSCLK周期。对于任何其他“n”值, 限定采样周期为2n SYSCLK周期(也就是说, 在每2n个SYSCLK周期上, GPIO引脚将被采样)。
- (B) 通过GPxCTRL寄存器选择的限定周期应用于8个GPIO引脚组。
- (C) 此限定块可取3个或者6个样片。GPxQSELn寄存器选择使用哪种采样模式。
- (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SYSCLK周期或者更长周期内保持稳定。换句话说, 输入应该在 $(5 \times QUALPRD \times 2)$ SYSCLK周期内保持稳定。这将确保有5个采样周期用于检测。由于外部信号是异步驱动的, 因此13个SYSCLK宽的脉冲确保了可靠的识别。

图 5.10 采样模式

5.7.7.3 输入信号的采样窗口宽度

下述小节总结了各种输入限定符配置下的输入信号的采样窗口宽度。采样频率表明相对于SYSCLK对信号进行采样的频率。

$$Sampling \ frequency = SYSCLK / (2 \times QUALPRD), \ if \ QUALPRD \neq 0 \quad (5.1)$$

$$Sampling \ frequency = SYSCLK, \ if \ QUALPRD = 0 \quad (5.2)$$

$$Sampling \ period = SYSCLK \ cycle \times 2 \times QUALPRD, \ if \ QUALPRD \neq 0 \quad (5.3)$$

在方程式5.1、方程式5.2和方程式5.3中, SYSCLK周期表示SYSCLK的时间周期。

如果QUALPRD=0, 则采样周期 = SYSCLK周期。

在给定的采样窗口中，采取输入信号的3个或者6个样片来确定信号的有效性。这是由写入到GPxQSELn寄存器的值确定的。

情况1:

使用3个样片进行限定

如果QUALPRD \neq 0，则采样窗口宽度 = (SYSCLK周期 \times 2 \times QUALPRD) \times 2

如果QUALPRD=0，则采样窗口宽度 = (SYSCLK周期) \times 2

情况2:

使用6个样片进行限定

如果QUALPRD \neq 0，则采样窗口宽度 = (SYSCLK周期 \times 2 \times QUALPRD) \times 5

如果QUALPRD=0，则采样窗口宽度 = (SYSCLK周期) \times 5

图5.11显示了通用输入时序。

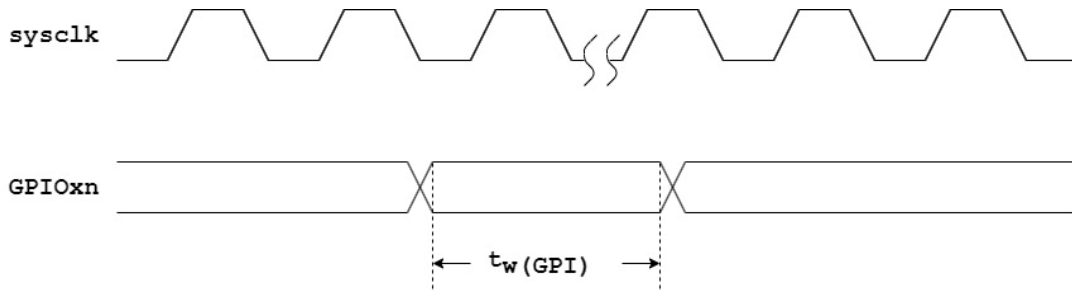


图 5.11 通用输入时序

5.7.8 中断

5.7.8.1 特性

NVIC包括以下特性：

- 具有240个可屏蔽中断。
- 每个中断的可编程优先级为0-127。较高的级别对应较低的优先级，因此级别0是最高的中断优先级。
- 支持电平中断和脉冲中断。
- 动态重新确定中断的优先级。
- 将优先级值分组为组优先级和子优先级字段。
- 中断tail-chaining（末尾连锁）。
- 一个外部不可屏蔽中断（NMI）。

嵌套向量中断控制器（NVIC）和处理器内核接口紧密配合，可以实现低延迟的中断处理和晚到中断的高效处理。包括内核异常在内的所有中断均通过NVIC进行管理。

5.7.8.2 NVIC寄存器

NVIC寄存器如表5.21所示。

表 5.21 NVIC寄存器列表

地址	名称	类型	复位值	说明
0xE000E100-0xE000E11C	NVIC_ISER0-NVIC_ISER7	RW	0x00000000	中断设置使能寄存器
0xE000E180-0xE000E19C	NVIC_ICER0-NVIC_ICER7	RW	0x00000000	中断清除使能寄存器
0xE000E200-0xE000E21C	NVIC_ISPR0-NVIC_ISPR7	RW	0x00000000	中断设置暂挂寄存器
0xE000E280-0xE000E29C	NVIC_ICPR0-NVIC_ICPR7	RW	0x00000000	中断清除暂挂寄存器
0xE000E300-0xE000E31C	NVIC_IABR0-NVIC_IABR7	RW	0x00000000	中断置位寄存器
0xE000E400-0xE000E4FC	NVIC_IPR0-NVIC_IPR63	RW	0x00000000	中断优先级寄存器，每个中断7bit
0xE000EF00	NVIC_STIR	WO	0x00000000	软件触发中断寄存器

5.7.8.2.1 NVIC_ISER

中断设置使能寄存器(NVIC_ISER0-NVIC_ISER7)。

表 5.22 ISER bit分配

Bits	名称	功能
[31:0]	SETENA	中断设置使能bits 写： 0: 不作用； 1: 使能中断； 读： 0: 中断禁止； 1: 中断使能；

如果一个暂挂的中断被使能，NVIC会根据它的优先级激活这个中断。如果一个中断未使能，其中断信号将中断状态更改为暂挂，但是NVIC永远不会激活中断，不管它的优先级如何。

5.7.8.2.2 NVIC_ICER

中断清除使能寄存器(NVIC_ICER0-NVIC_ICER7)。

表 5.23 ICER bit分配

Bits	名称	功能
[31:0]	CLRENA	中断清除使能bits 写： 0: 不作用； 1: 禁止中断； 读： 0: 中断禁止； 1: 中断使能；

ISPR bit写1后：

- 相应中断会暂挂，不起作用。
- 被禁用的中断将该中断的状态设置为暂挂。

5.7.8.2.3 NVIC_ICPR

中断清除暂挂寄存器(NVIC_ICPR0-NVIC_ICPR7)从中断中删除暂挂状态，并显示哪些中断处于暂挂状态。将1写入ICPR位不会影响相应中断的活动状态。

表 5.24 ICPR bit分配

Bits	名称	功能
[31:0]	SETPEND	中断设置暂挂bits 写： 0: 不作用； 1: 将中断移除暂挂状态； 读： 0: 中断未暂挂； 1: 中断暂挂；

5.7.8.2.4 NVIC_IABR

中断置位寄存器(NVIC_IABR0-NVIC_IABR7)。如果相应的中断状态为激活或激活暂挂状态，则该位读为1。

表 5.25 IABR bit分配

Bits	名称	功能
[31:0]	ACTIVE	中断激活指示： 0：中断未激活； 1：中断激活；

5.7.8.2.5 NVIC_IPR

中断优先级寄存器(NVIC_IPR0-NVIC_IPR63)为每个中断提供7-bit优先级设。这些寄存器是字节可访问的。

表 5.26 IPR bit分配

Bits	名称	功能
[31:25]	Priority 3	每个优先级字段包含一个优先级值，0-127。该值越低，相应中断的优先级越高。处理器只实现每个字段的[7:n]位，将[n-1:0]位读为零，忽略写。
[24]	—	
[23:17]	Priority 2	
[16]	—	
[15:9]	Priority 1	
[8]	—	
[7:1]	Priority 0	
1-2 [0]	—	

5.7.8.2.6 NVIC_STIR

软件写STIR寄存器可以产生中断。当SCR中USERSETMPEND比特设置为1时，非特权软件可以访问STIR寄存器。

表 5.27 STIR bit分配

Bits	名称	功能
[31:9]	—	保留
[8:0]	INTID	中断要触发的中断ID，范围0-239。例如0x03表示中断IRQ3。

5.7.8.3 中断处理

M7支持电平敏感中断和脉冲中断。脉冲中断也称为边缘触发中断。

电平敏感中断被保持有效直到外设解除中断信号。通常发生这种情况是因为中断服务程序(ISR)访问外设，导致它清除中断请求。脉冲中断是在处理器时钟上升沿同步采样的中断信号。为了确保NVIC检测到中断，外设必须在至少一个时钟周期内保持中断信号，在此期间NVIC检测到脉冲并锁存中断。

中断的硬件和软件控制：

- Cortex-M7锁存所有中断，由于下列原因之一，外设中断变为暂挂状态：
 - NVIC检测到中断信号为HIGH，并且中断不活动。
 - NVIC检测到中断信号上的上升沿。
- 软件写入相应的中断设置-暂挂寄存器位，或向STIR发送中断等待。
一个暂挂的中断保持暂挂状态，直到下列情况之一发生：

- 处理器为中断进入中断服务程序(ISR)。这将中断的状态从暂挂更改为活动。
 - 对于电平敏感中断，当处理器从中断服务程序(ISR)返回时，NVIC对中断信号进行采样。如果此时信号仍为有效，中断的状态变为暂挂，这可能导致处理器立即重新进入ISR。否则，中断状态变为非活动状态。
 - 对于脉冲中断，NVIC继续监视中断信号，如果是脉冲中断，则中断的状态变为暂挂和活动。在这种情况下，当处理器从ISR返回时，中断的状态变为暂挂状态，这可能导致处理器立即重新进入ISR。
 - 当处理器处于中断服务程序(ISR)中，如果中断信号此时不是脉冲的，当处理器从ISR返回时，中断状态变为非活动状态。
- 软件写入相应的中断清除暂挂寄存器bit位。

对于电平敏感中断，如果中断信号仍然被激活，则中断的状态不会改变。否则，中断状态变为非活动状态。
对于脉冲中断，中断的状态变为:

 - Inactive，如果状态为暂挂。
 - Active，如果状态是激活和暂挂。

5.7.8.4 中断和异常向量

M7处理器一共有255个异常，异常编号为1~255；其中1~15属于内核异常，大于15属于外设中断；在程序开发中，使用-14~-1表示内核中断（异常）请求编号，大于-1表示外设中断请求编号。

中断向量编号如表5.28所示。

表 5.28 中断向量编号

中断向量名称	CPU1中断序号	CPU2中断序号	地址
Initial SP value	—	—	0x0000_0000
Reset	—	—	0x0000_0004
NMI	-14	-14	0x0000_0008
Hard fault	-13	-13	0x0000_000C
Memory management fault	-12	-12	0x0000_0010
Bus fault	-11	-11	0x0000_0014
Usage fault	-10	-10	0x0000_0018
Reserved	—	—	0x0000_001C-0x0000_002B
SVCall	-5	-5	0x0000_002C
Reserved for Debug	—	—	0x0000_0030
Reserved	—	—	0x0000_0034
PendSV	-2	-2	0x0000_0038
Systick	-1	-1	0x0000_003C
CPU1 WDT intr	0	—	0x0000_0040
CPU2 WDT intr	—	0	
CPU1 timer0 intr	1	—	0x0000_0044
CPU2 timer0 intr	—	1	
CPU1 timer1 intr	2	—	0x0000_0048
CPU2 timer1 intr	—	2	
CPU1 timer2 intr	3	—	0x0000_004C
CPU2 timer2 intr	—	3	
—	4	4	0x0000_0050
—	5	5	0x0000_0054

(续表见下页)

表5.26 中断向量编号（续表）

中断向量名称	CPU1中断序号	CPU2中断序号	地址
—	6	6	0x0000_0058
—	7	7	0x0000_005C
CPU2toCPU1 IPC intr0	8	—	0x0000_0060
CPU1toCPU2 IPC intr0	—	8	
CPU2toCPU1 IPC intr1	9	—	0x0000_0064
CPU1toCPU2 IPC intr1	—	9	
CPU2toCPU1 IPC intr2	10	—	0x0000_0068
CPU1toCPU2 IPC intr2	—	10	
CPU2toCPU1 IPC intr3	11	—	0x0000_006C
CPU1toCPU2 IPC intr3	—	11	
ADC_A event intr	12	12	0x0000_0070
ADC_B event intr	13	13	0x0000_0074
ADC_C event intr	14	14	0x0000_0078
ADC_D event intr	15	15	0x0000_007C
ADC_A intr1	16	16	0x0000_0080
ADC_A intr2	17	17	0x0000_0084
ADC_A intr3	18	18	0x0000_0088
ADC_A intr4	19	19	0x0000_008C
ADC_B intr1	20	20	0x0000_0090
ADC_B intr2	21	21	0x0000_0094
ADC_B intr3	22	22	0x0000_0098
ADC_B intr4	23	23	0x0000_009C
ADC_C intr1	24	24	0x0000_00A0
ADC_C intr2	25	25	0x0000_00A4
ADC_C intr3	26	26	0x0000_00A8
ADC_C intr4	27	27	0x0000_00AC
ADC_D intr1	28	28	0x0000_00B0
ADC_D intr2	29	29	0x0000_00B4
ADC_D intr3	30	30	0x0000_00B8
ADC_D intr4	31	31	0x0000_00BC
External intr1	32	32	0x0000_00C0
External intr2	33	33	0x0000_00C4
External intr3	34	34	0x0000_00C8
External intr4	35	35	0x0000_00CC
External intr5	36	36	0x0000_00D0
—	37	37	0x0000_00D4
—	38	38	0x0000_00D8
PWM HRCAL intr	39	—	0x0000_00DC
—	—	39	
PWM1 TZ intr	40	40	0x0000_00E0
PWM2 TZ intr	41	41	0x0000_00E4

（续表见下页）

表5.26 中断向量编号（续表）

中断向量名称	CPU1中断序号	CPU2中断序号	地址
PWM3 TZ intr	42	42	0x0000_00E8
PWM4 TZ intr	43	43	0x0000_00EC
PWM5 TZ intr	44	44	0x0000_00F0
PWM6 TZ intr	45	45	0x0000_00F4
PWM7 TZ intr	46	46	0x0000_00F8
PWM8 TZ intr	47	47	0x0000_00FC
PWM9 TZ intr	48	48	0x0000_0100
PWM10 TZ intr	49	49	0x0000_0104
PWM11 TZ intr	50	50	0x0000_0108
PWM12 TZ intr	51	51	0x0000_010C
—	52	52	0x0000_0110
—	53	53	0x0000_0114
—	54	54	0x0000_0118
—	55	55	0x0000_011C
PWM1 intr	56	56	0x0000_0120
PWM2 intr	57	57	0x0000_0124
PWM3 intr	58	58	0x0000_0128
PWM4 intr	59	59	0x0000_012C
PWM5 intr	60	60	0x0000_0130
PWM6 intr	61	61	0x0000_0134
PWM7 intr	62	62	0x0000_0138
PWM8 intr	63	63	0x0000_013C
PWM9 intr	64	64	0x0000_0140
PWM10 intr	65	65	0x0000_0144
PWM11 intr	66	66	0x0000_0148
PWM12 intr	67	67	0x0000_014C
—	68	68	0x0000_0150
—	69	69	0x0000_0154
—	70	70	0x0000_0158
—	71	71	0x0000_015C
CAP1 intr	72	72	0x0000_0160
CAP2 intr	73	73	0x0000_0164
CAP3 intr	74	74	0x0000_0168
CAP4 intr	75	75	0x0000_016C
CAP5 intr	76	76	0x0000_0170
CAP6 intr	77	77	0x0000_0174
—	78	78	0x0000_0178
—	79	79	0x0000_017C
QEP1 intr	80	80	0x0000_0180
QEP2 intr	81	81	0x0000_0184
QEP3 intr	82	82	0x0000_0188

（续表见下页）

表5.26 中断向量编号 (续表)

中断向量名称	CPU1中断序号	CPU2中断序号	地址
—	83	83	0x0000_018C
CLB1 intr	84	84	0x0000_0190
CLB2 intr	85	85	0x0000_0194
CLB3 intr	86	86	0x0000_0198
CLB4 intr	87	87	0x0000_019C
CLB5 intr	88	88	0x0000_01A0
CLB6 intr	89	89	0x0000_01A4
CLB7 intr	90	90	0x0000_01A8
CLB8 intr	91	91	0x0000_01AC
—	92	92	0x0000_01B0
—	93	93	0x0000_01B4
—	94	94	0x0000_01B8
—	95	95	0x0000_01BC
SDFM1 intr	96	96	0x0000_01C0
SDFM1 DR intr1	97	97	0x0000_01C4
SDFM1 DR intr2	98	98	0x0000_01C8
SDFM1 DR intr3	99	99	0x0000_01CC
SDFM1 DR intr4	100	100	0x0000_01D0
SDFM2 intr	101	101	0x0000_01D4
SDFM2 DR intr1	102	102	0x0000_01D8
SDFM2 DR intr2	103	103	0x0000_01DC
SDFM2 DR intr3	104	104	0x0000_01E0
SDFM2 DR intr4	105	105	0x0000_01E4
—	106	106	0x0000_01E8
—	107	107	0x0000_01EC
—	108	108	0x0000_01F0
—	109	109	0x0000_01F4
—	110	110	0x0000_01F8
—	111	111	0x0000_01FC
DMA CH1 intr	112	112	0x0000_0200
DMA CH2 intr	113	113	0x0000_0204
DMA CH3 intr	114	114	0x0000_0208
DMA CH4 intr	115	115	0x0000_020C
DMA CH5 intr	116	116	0x0000_0210
DMA CH6 intr	117	117	0x0000_0214
—	118	118	0x0000_0218
—	119	119	0x0000_021C
—	120	120	0x0000_0220
—	121	121	0x0000_0224
—	122	122	0x0000_0228
—	123	123	0x0000_022C

(续表见下页)

表5.26 中断向量编号 (续表)

中断向量名称	CPU1中断序号	CPU2中断序号	地址
—	124	124	0x0000_0230
—	125	125	0x0000_0234
USB20 intr	126	126	0x0000_0238
USB20 EP MP intr	127	127	0x0000_023C
USB20 EP0 intr	128	128	0x0000_0240
USB20 EP1 intr	129	129	0x0000_0244
USB20 EP2 intr	130	130	0x0000_0248
USB20 EP3 intr	131	131	0x0000_024C
USB20 EP4 intr	132	132	0x0000_0250
USB20 EP5 intr	133	133	0x0000_0254
USB20 EP6 intr	134	134	0x0000_0258
USB20 EP7 intr	135	135	0x0000_025C
USB20 EP8 intr	136	136	0x0000_0260
USB20 EP9 intr	137	137	0x0000_0264
USB20 EP10 intr	138	138	0x0000_0268
USB20 EP11 intr	139	139	0x0000_026C
USB20 EP12 intr	140	140	0x0000_0270
USB20 EP13 intr	141	141	0x0000_0274
USB20 EP14 intr	142	142	0x0000_0278
USB20 EP15 intr	143	143	0x0000_027C
USB20 EP16 intr	144	144	0x0000_0280
USB20 EP17 intr	145	145	0x0000_0284
USB20 EP18 intr	146	146	0x0000_0288
USB20 EP19 intr	147	147	0x0000_028C
USB20 EP20 intr	148	148	0x0000_0290
USB20 EP21 intr	149	149	0x0000_0294
USB20 EP22 intr	150	150	0x0000_0298
USB20 EP23 intr	151	151	0x0000_029C
USB20 EP24 intr	152	152	0x0000_02A0
USB20 EP25 intr	153	153	0x0000_02A4
USB20 EP26 intr	154	154	0x0000_02A8
USB20 EP27 intr	155	155	0x0000_02AC
USB20 EP28 intr	156	156	0x0000_02B0
USB20 EP29 intr	157	157	0x0000_02B4
USB20 EP30 intr	158	158	0x0000_02B8
USB20 EP31 intr	159	159	0x0000_02BC
CAN_A intr0	160	160	0x0000_02C0
CAN_A intr1	161	161	0x0000_02C4
CAN_B intr0	162	162	0x0000_02C8
CAN_B intr1	163	163	0x0000_02CC
CAN_C intr0	164	164	0x0000_02D0

(续表见下页)

表5.26 中断向量编号（续表）

中断向量名称	CPU1中断序号	CPU2中断序号	地址
CAN_C intr1	165	165	0x0000_02D4
—	166	166	0x0000_02D8
—	167	167	0x0000_02DC
I2C_A intr	168	168	0x0000_02E0
I2C_B intr	169	169	0x0000_02E4
—	170	170	0x0000_02E8
—	171	171	0x0000_02EC
PMBus_A intr	172	172	0x0000_02F0
—	173	173	0x0000_02F4
—	174	174	0x0000_02F8
—	175	175	0x0000_02FC
UART_A intr	176	176	0x0000_0300
UART_B intr	177	177	0x0000_0304
UART_C intr	178	178	0x0000_0308
UART_D intr	179	179	0x0000_030C
SPI_A intr	180	180	0x0000_0310
SPI_B intr	181	181	0x0000_0314
SPI_C intr	182	182	0x0000_0318
—	183	183	0x0000_031C
SRAM correctable error intr	184	184	0x0000_0320
eFlash correctable error intr	185	185	0x0000_0324
SRAM access violation intr	186	186	0x0000_0328
EMIF error intr	187	187	0x0000_032C
WAKEUP intr	188	188	0x0000_0330
—	189	189	0x0000_0334
—	190	190	0x0000_0338
—	191	191	0x0000_033C
TFU intr	192	192	0x0000_0340
IPC ram err intr	193	193	0x0000_0344
	194	194	0x0000_0348
	195	195	0x0000_034C
	196	196	0x0000_0350
	197	197	0x0000_0354
	198	198	0x0000_0358
	199	199	0x0000_035C
	200	200	0x0000_0360
	201	201	0x0000_0364
	202	202	0x0000_0368
	203	203	0x0000_036C
	204	204	0x0000_0370
	205	205	0x0000_0374

（续表见下页）

表5.26 中断向量编号 (续表)

中断向量名称	CPU1中断序号	CPU2中断序号	地址
	206	206	0x0000_0378
	207	207	0x0000_037C
	208	208	0x0000_0380
	209	209	0x0000_0384
	210	210	0x0000_0388
	211	211	0x0000_038C
	212	212	0x0000_0390
	213	213	0x0000_0394
	214	214	0x0000_0398
	215	215	0x0000_039C
	216	216	0x0000_03A0
	217	217	0x0000_03A4
	218	218	0x0000_03A8
	219	219	0x0000_03AC
	220	220	0x0000_03B0
	221	221	0x0000_03B4
	222	222	0x0000_03B8
	223	223	0x0000_03BC
	224	224	0x0000_03C0
	225	225	0x0000_03C4
	226	226	0x0000_03C8
	227	227	0x0000_03CC
	228	228	0x0000_03D0
CPU1 exception ...	229	229	0x0000_03D4
CPU2 exception ...	230	230	0x0000_03D8
	231	231	0x0000_03DC
	232	232	0x0000_03E0
	233	233	0x0000_03E4
	234	234	0x0000_03E8
	235	235	0x0000_03EC
	236	236	0x0000_03F0
	237	237	0x0000_03F4
	238	238	0x0000_03F8
	239	239	0x0000_03FC

5.7.9 低功耗模式

器件支持2种低功耗模式。

更多的低功耗相关的细节，比如进入和退出流程，请参考TRM文档相关章节。

5.7.10 外部存储接口 (EMIF)

EMIF提供一组扩展接口，用于外挂存储器件，扩充系统存储空间，支持CPU和DMA直接对外挂存储器件进行访问。支持存储器件类型包括异步存储器件 (SRAM, NOR flash) 和同步存储器件 (SDRAM)。

XMC包括两个控制模块:

- 异步存储控制模块，支持NOR FLASH和SRAM;
- 同步存储控制模块，支持SDRAM;

异步存储控制模块支持最多挂载3块存储器件，通过3bit cs信号选择:

- 支持8bits/16bits/32bits数据位宽;
- 读写周期时序可配置;
- 读写切换间隔时间可配置;
- 支持外部wait信号，并支持配置最大等待时间;
- 支持normal mode和Select strobe mode。

同步存储控制模块支持符合JESD21-C标准的 SDR SDRAM。支持单个SDRAM芯片，通过CS信号选择。

- 支持1/2/4 bank设备;
- 8/9/10/11 column address;
- CAS延迟2/3个时钟周期;
- 16/32bit数据位宽;

此外，EMIF支持将SDRAM置于自刷新和断电模式。自刷新模式允许SDRAM处于低功耗状态，同时仍然保留SDRAM中内容，此时SDRAM处于自刷新状态，无需EMIF控制器提供时钟。断电模式实现更低的功耗，如果需保留SDRAM数据，EMIF控制器必须设置定期发出刷新指令。EMIF模块不支持移动SDRAM设备。

5.7.10.1 异步读写时序

单次异步读写分为3个阶段，setup、strobe、hold，全部寄存器可配置。支持两种模式normal mode和Select strobe mode。如图5.12、5.13、5.14、5.15所示。

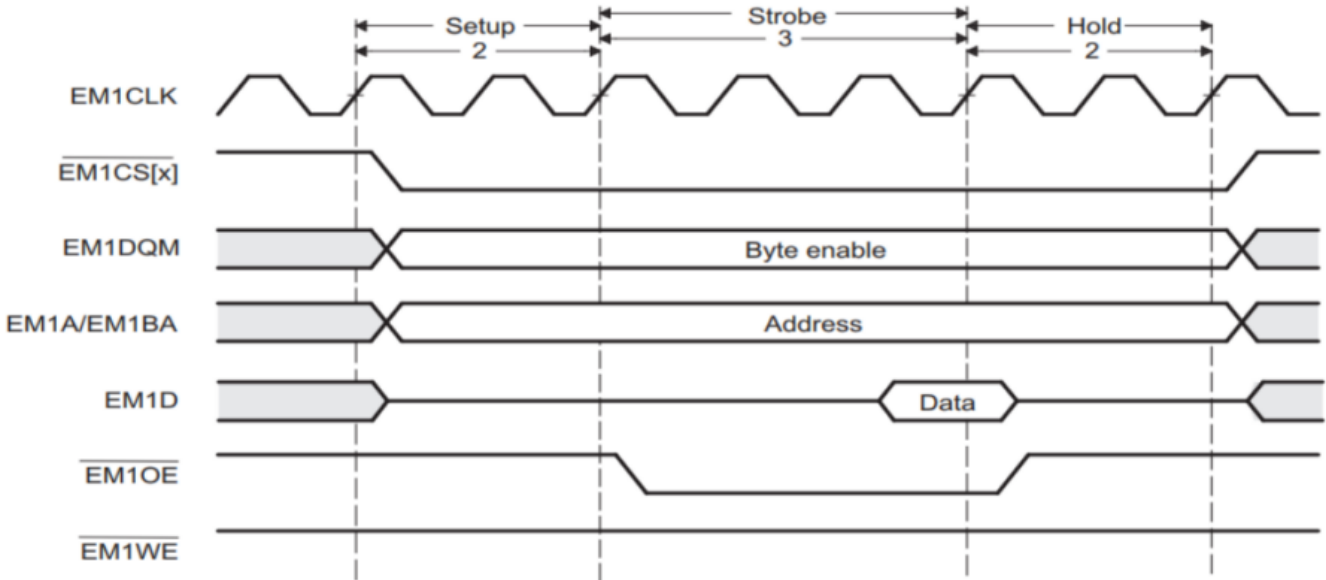


图 5.12 normal mode读时序图

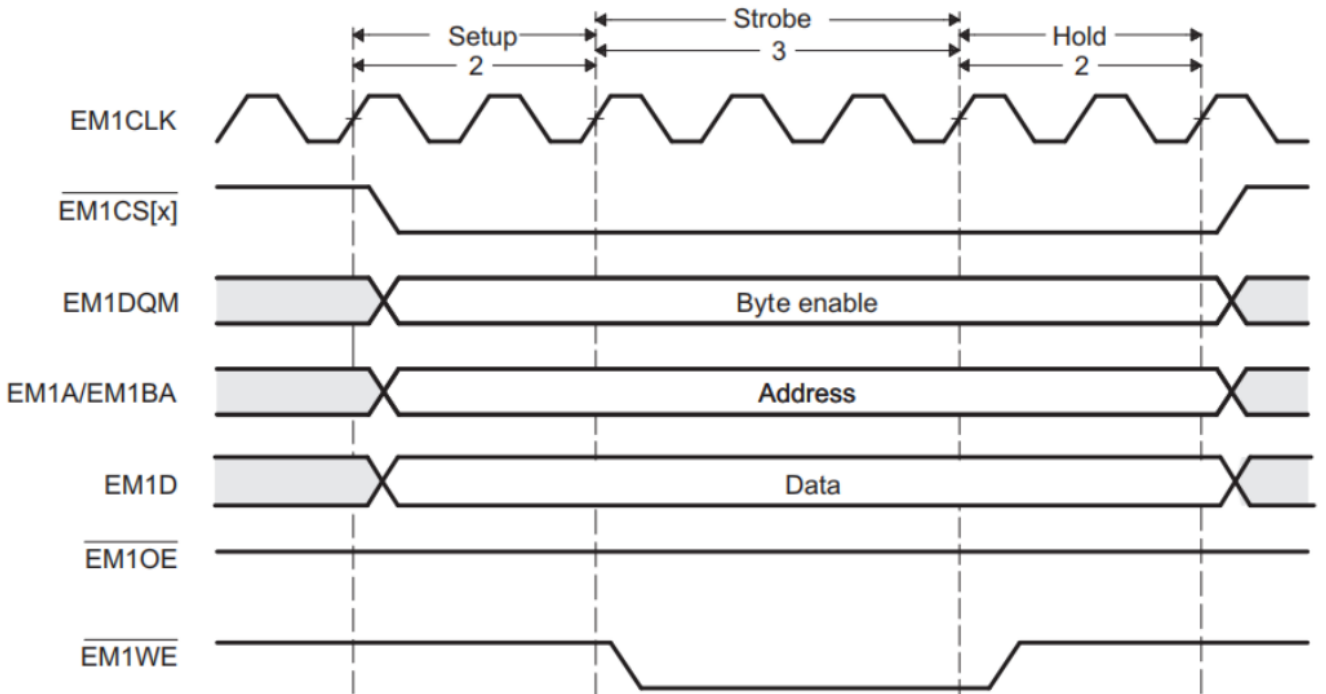


图 5.13 normal mode写时序图

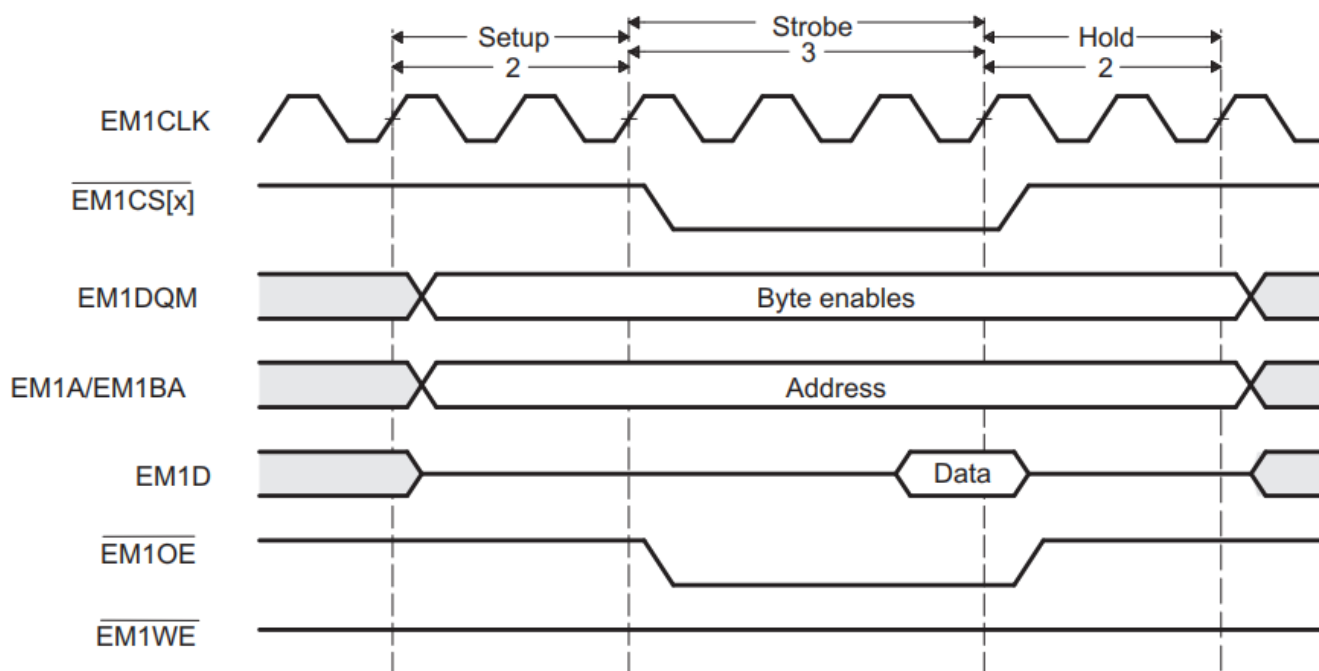


图 5.14 Select strobe mode读时序图

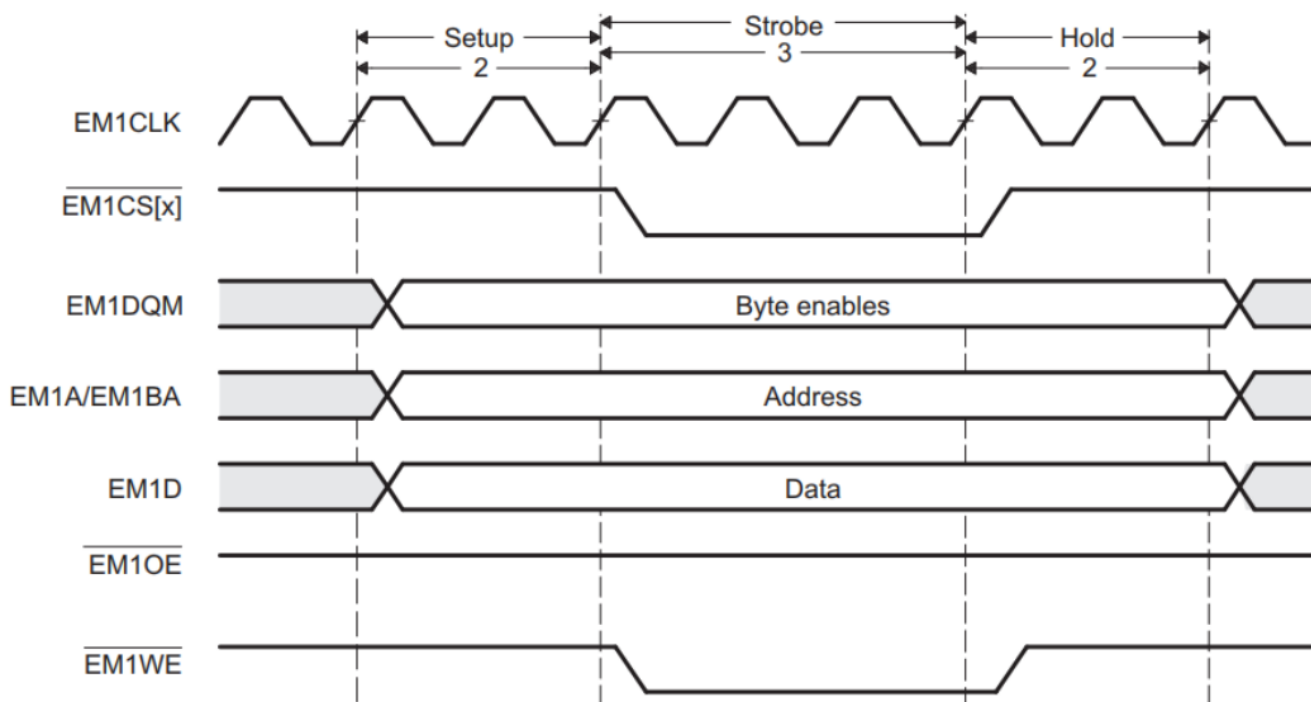


图 5.15 Select strobe mode写时序图

5.7.10.2 异步读写时序

如图5.16、5.17所示。

表 5.29 同步SDRAM读写参数

参数	描述	Min	Max	Unit
$t_c(CLK)$	Cycle time, EMIF clock EMxCLK	26.6		ns
$t_w(CLK)$	Pulse width, EMIF clock EMxCLK high or low	10		ns
$t_d(CLKH-CSV)$	Delay time, EMxCLK rising to EMxCS[y:2] valid		8	ns
$t_{oh}(CLKH-CSIV)$	Output hold time, EMxCLK rising to EMxCS[y:2] invalid	1		ns
$t_d(CLKH-DQMV)$	Delay time, EMxCLK rising to EMxDQM[y:0] valid		8	ns
$t_{oh}(CLKH-DQMIV)$	Output hold time, EMxCLK rising to EMxDQM[y:0] invalid	1		ns
$t_d(CLKH-AV)$	Delay time, EMxCLK rising to EMxA[y:0] and EMxBA[y:0] valid		8	ns
$t_{oh}(CLKH-AIV)$	Output hold time, EMxCLK rising to EMxA[y:0] and EMxBA[y:0] invalid	1		ns
$t_d(CLKH-DV)$	Delay time, EMxCLK rising to EMxD[y:0] valid		8	ns
$t_{oh}(CLKH-DIV)$	Output hold time, EMxCLK rising to EMxD[y:0] invalid	1		ns
$t_d(CLKH-RASV)$	Delay time, EMxCLK rising to EMxRAS valid		8	ns
$t_{oh}(CLKH-RASIV)$	Output hold time, EMxCLK rising to EMxRAS invalid	1		ns
$t_d(CLKH-CASV)$	Delay time, EMxCLK rising to EMxCAS valid		8	ns
$t_{oh}(CLKH-CASIV)$	Output hold time, EMxCLK rising to EMxCAS invalid	1		ns
$t_d(CLKH-WEV)$	Delay time, EMxCLK rising to EMxWE valid		8	ns
$t_{oh}(CLKH-WEIV)$	Output hold time, EMxCLK rising to EMxWE invalid	1		ns
$t_d(CLKH-DHZ)$	Delay time, EMxCLK rising to EMxD[y:0] tri-stated		8	ns
$t_{oh}(CLKH-DLZ)$	Output hold time, EMxCLK rising to EMxD[y:0] driving	1		ns
$t_{su}(EMIFDV-EMCLKH)$	Input setup time, read data valid on EMxD[y:0] before EMxCLK rising	2		ns
$t_h(CLKH-DIV)$	Input hold time, read data valid on EMxD[y:0] after EMxCLK rising	1.5		ns

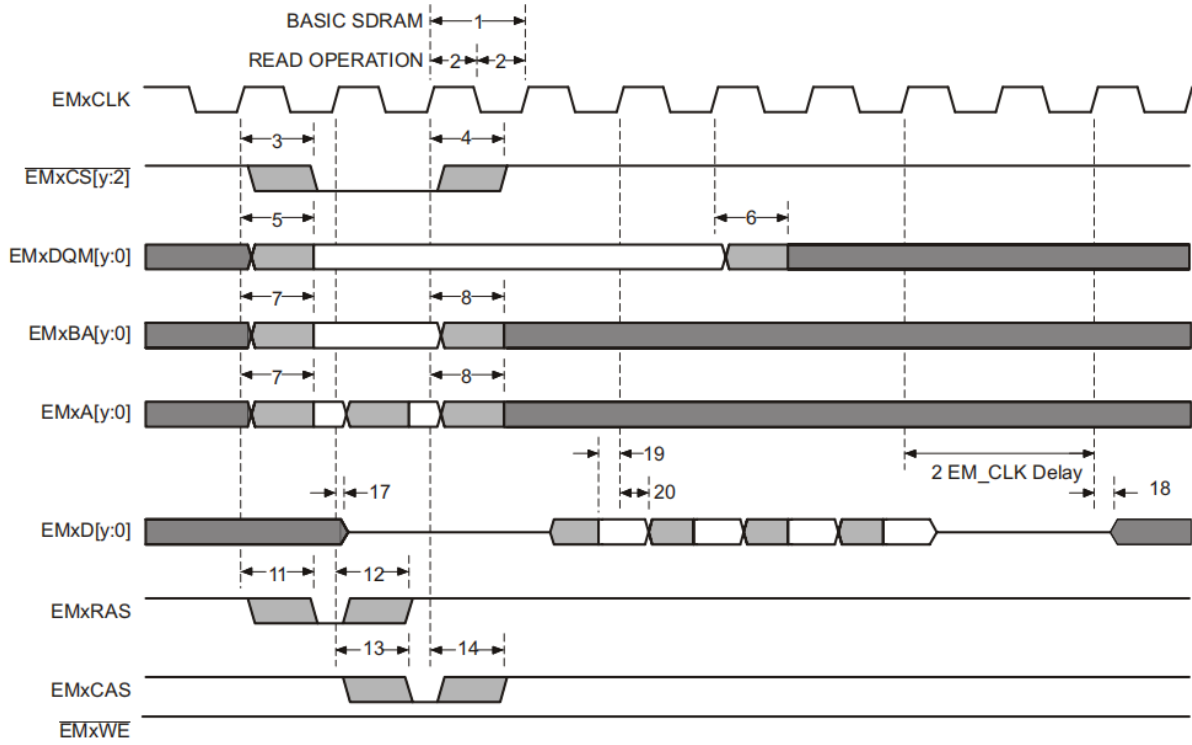


图 5.16 SDRAM读时序

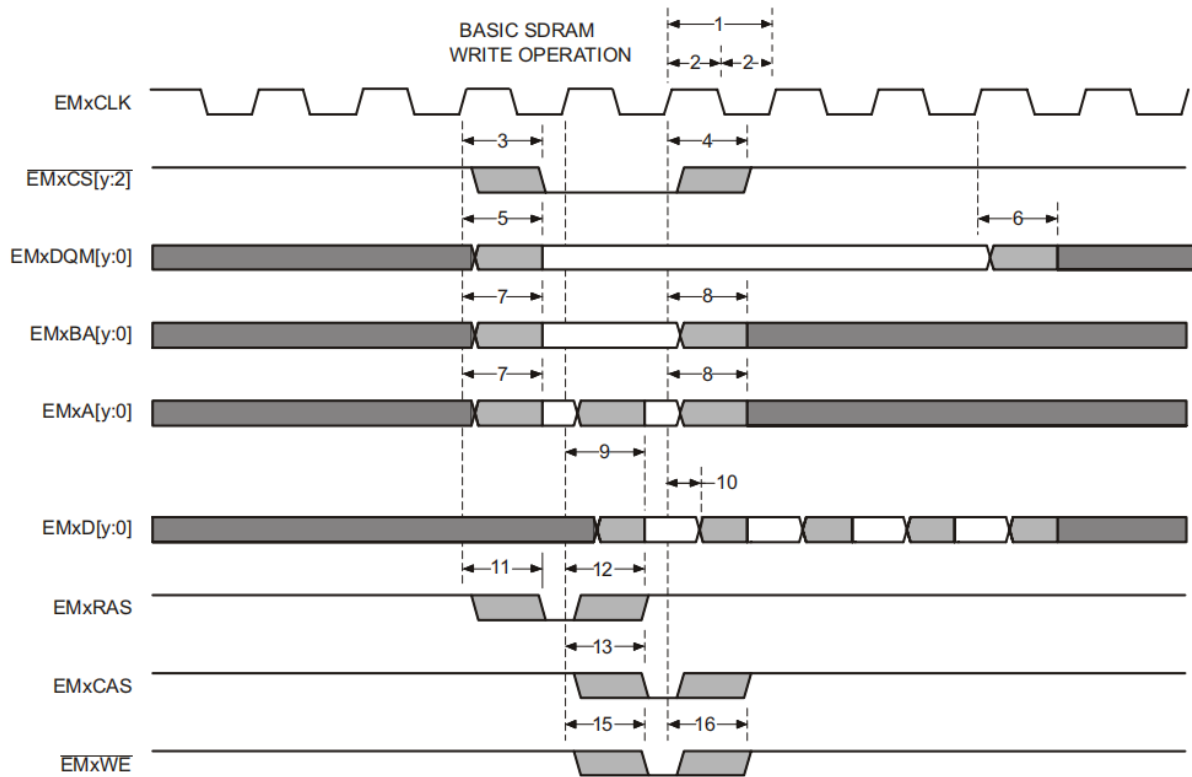


图 5.17 SDRAM写时序

5.8 模拟外设

本节介绍模拟子系统模块，具体包括ADC、温度传感器、buffered DAC和CMPSS。

5.8.1 模数转换器(ADC)

该器件上的ADC是逐次逼近（successive approximation, SAR）式ADC，具有16位或12位的可选分辨率，多个ADC模块允许同时采样。ADC wrapper是基于转换开始（start-of-conversion, SOC）触发的[参见HS32F7D377PTI双核微控制器技术参考手册的SOC工作原理部分]。ADC模型如图5.18所示。

每个ADC都有以下特点：

- 可选择的16位或12位的分辨率
- 由VREFHI和VREFLO设定的比例外部参考
- 差分信号转换（仅限16位模式）
- 单端信号转换（12位模式）
- 输入多路复用器，最多16通道（单端）或8通道（差分）
- 16个可配置的SOC
- 16个可单独寻址的结果寄存器
- 多个触发源
 - 软件立即启动
 - 所有ePWM
 - GPIO XINT2
 - CPU定时器
 - ADCINT1或2
- 四个灵活的NVIC中断
- 突发模式
- 四个后处理模块，每个具有：
 - 饱和偏移校准
 - 设定值错误计算
 - 高、低、过零比较，可产生中断和ePWM TRIP
 - 触发到采样延迟时间统计

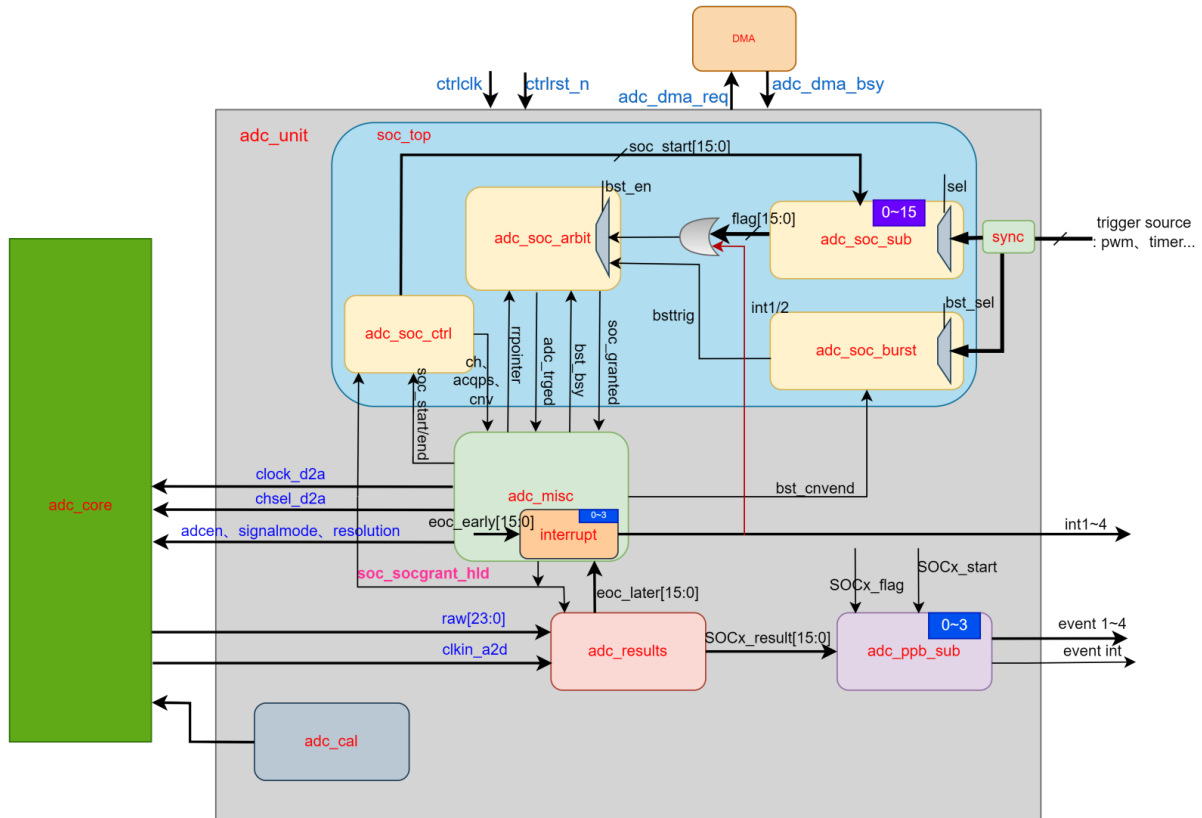


图 5.18 ADC模型框图

5.8.1.1 ADC可配置性

部分ADC配置由SOC单独控制，而其他部分则由每个ADC模块控制。表5.30总结了基本的ADC选项及其配置级别。

表 5.30 ADC选项和配置级别

选项	配置级别
时钟	模块 ⁽¹⁾
分辨率	模块 ⁽¹⁾
信号模式	模块
参考电压源	不可配置（仅外部参考源）
触发源	SOC ⁽¹⁾
转换通道	SOC
采集窗口持续时间	SOC ⁽¹⁾
EOC位置	模块
突发模式	模块 ⁽¹⁾

(1) 将这些值以不同的方式写入不同的ADC模块可能导致ADC异步操作。有关ADC同步或异步操作的指导，请参阅HS32F7D377PTI双核微控制器技术参考手册中模数转换器（ADC）章节的确保同步操作部分。

5.8.1.2 信号模式

ADC支持单端和差分两种信号模式。在单端模式下，转换器的输入电压通过一个引脚（ADCIN_x）采样，参考 V_{REFHI} 。在差分信号模式下，转换器的输入电压通过一对输入引脚采样，其中一个是正输入（ADCIN_{xP}），另一个是负输入（ADCIN_{xN}）。实际输入电压为两者之差（ADCIN_{xP}-ADCIN_{xN}）。

5.8.1.3 ADC电气数据和时序

16位差分模式的ADC工作条件如表5.31所示。16位差分模式的ADC特性如表5.32所示。12位单端模式的ADC工作条件如表5.33所示。12位单端模式的ADC特性如表5.34所示。ADCEXTSOC的时序要求如表5.35所示。

表 5.31 ADC工作条件(16位差分模式)

推荐的操作条件（除非另有说明）

	MIN	TYP	MAX	UNIT
采样窗口时间（由ACQPS和PERx.SYSCLK设置）	106.7			ns
V_{REFHI}	2.4	2.5 or 3.0	V_{DDA}	V
V_{REFLO}	V_{SSA}	0	V_{SSA}	V
$V_{REFHI}-V_{REFLO}$	2.4		V_{DDA}	V
ADC输入转换范围	V_{REFLO}		V_{REFHI}	V
ADC输入信号共模电压 ⁽¹⁾⁽²⁾	$V_{REFCM}-50$	V_{REFCM}	$V_{REFCM}+50$	mV

(1) $V_{REFCM}=(V_{REFHI}+V_{REFLO})/2$

(2) 如果ADC负输入引脚连接到 V_{SSA} 或 V_{REFLO} ，则不满足 V_{REFCM} 要求。

请注意

在工作期间，ADC输入应保持在 $V_{DDA}+0.3$ V以下。如果ADC输入超过此电平，器件内部的 V_{REF} 可能会受到干扰，这可能会影响使用相同 V_{REF} 的其他ADC或DAC输入的结果。

请注意

V_{REFHI} 引脚必须保持在 $V_{DDA}+0.3$ V以下，以确保正常的功能操作。如果 V_{REFHI} 引脚超过此电平，则阻塞电路可能被激活，并且 V_{REFHI} 的内部值可能在内部浮动到0 V，从而导致不正确的ADC转换或DAC输出。

表 5.32 ADC特性(16位差分模式)

推荐的操作条件（除非另有说明）⁽¹⁾

参数	测试条件	MIN	TYP	MAX	UNIT
ADC转换周期 ⁽²⁾		29.1		55.4	ADCCLKs
上电时间			1.5		μ s
增益误差			± 3		LSBs
偏移误差 ⁽³⁾			± 2		LSBs
DNL ⁽⁴⁾			± 0.5		LSBs
INL			± 1		LSBs
SNR ⁽⁵⁾⁽⁶⁾	$V_{REFHI}=2.5V, f_{in}=2kHz$		85.64		dB
THD ⁽⁵⁾⁽⁶⁾	$V_{REFHI}=2.5V, f_{in}=2kHz$		90.45		dB
SFDR ⁽⁵⁾⁽⁶⁾	$V_{REFHI}=2.5V, f_{in}=2kHz$		92.45		dB
SINAD ⁽⁵⁾⁽⁶⁾	$V_{REFHI}=2.5V, f_{in}=2kHz$		84.49		dB
ENOB	$V_{REFHI}=2.5V, f_{in}=2kHz$, 单个ADC		13.74		bits
V_{REFHI} 输入电流			280		μ A

- (1) 在 $V_{REFHI}=2.5V$ 和 $V_{REFLO}=0V$ 下测量典型值。最小值和最大值用 $V_{REFHI}=2.5V$ 和 $V_{REFLO}=0V$ 进行测试或表征。
- (2) 参见5.8.1.3.2节。
- (3) $ADCIN_p=ADCIN_n=V_{REFCM}$ 时与转换结果32768的差异。
- (4) No missing code。
- (5) 交流参数会受到时钟源精度和抖动的影响，所以系统选择时钟源时应考虑到这一点。测量这些参数时选用的时钟源是通过锁相环馈送的高精度外部时钟。片上内部振荡器具有比外部晶体更高的抖动，如果将其用作时钟源，这些参数将降低。
- (6) 尽量避免ADC输入和 V_{REFHI} 引脚附近的引脚I/O活动，以减少电容耦合和串扰。

表 5.33 ADC工作条件(12位单端模式)

推荐的操作条件（除非另有说明）

	MIN	TYP	MAX	UNIT
ADCCLK	10		150	MHz
采样窗口时间（由ACQPS和PERx.SYSCLK设置） ⁽¹⁾	33			ns
V_{REFHI}	2.4	2.5 or 3.0	V_{DDA}	V
V_{REFLO}	V_{SSA}	0	V_{SSA}	V
$V_{REFHI}-V_{REFLO}$	2.4		V_{VDDA}	V
ADC输入转换范围	V_{REFLO}		V_{REFHI}	V

(1) 采样窗口也必须至少为1个ADCCLK周期，以保证正确的ADC操作。

请注意

在工作期间，ADC输入应保持在 $V_{DDA}+0.3$ V以下。如果ADC输入超过此电平，器件内部的 V_{REF} 可能会受到干扰，这可能会影响使用相同 V_{REF} 的其他ADC或DAC输入的结果。

请注意

V_{REFHI} 引脚必须保持在 $V_{DDA}+0.3$ V以下，以确保正常的功能操作。如果 V_{REFHI} 引脚超过此电平，则阻塞电路可能被激活，并且 V_{REFHI} 的内部值可能在内部浮动到0 V，从而导致不正确的ADC转换或DAC输出。

表 5.34 ADC特性(12位单端模式)

推荐的操作条件（除非另有说明）⁽¹⁾

参数	测试条件	MIN	TYP	MAX	UNIT	
ADC转换时间 ⁽²⁾		16.8		32.2	ADCCLKs	
上电时间		1.5			μ s	
增益误差		± 3			LSBs	
偏移误差		± 2			LSBs	
DNL ⁽³⁾		± 0.5			LSBs	
INL		± 1			LSBs	
参数	测试条件	A	B	C	D ⁽⁶⁾	UNIT
SNR ⁽⁴⁾⁽⁵⁾	$V_{REFHI}=2.5V, f_{in}=2kHz, f_s=1.1Msps$ ⁽⁷⁾	69.96	69.99	69.93	67.76	dB
THD ⁽⁴⁾⁽⁵⁾	$V_{REFHI}=2.5V, f_{in}=2kHz, f_s=1.1Msps$ ⁽⁷⁾	81.92	79.40	76.02	77.34	dB
SFDR ⁽⁴⁾⁽⁵⁾	$V_{REFHI}=2.5V, f_{in}=2kHz, f_s=1.1Msps$ ⁽⁷⁾	84.18	81.34	81.44	81.52	dB
SINAD ⁽⁴⁾⁽⁵⁾	$V_{REFHI}=2.5V, f_{in}=2kHz, f_s=1.1Msps$ ⁽⁷⁾	69.75	69.61	69.16	67.40	dB
ENOB	$V_{REFHI}=2.5V, f_{in}=2kHz, f_s=1.1Msps$	11.29	11.27	11.20	10.90	bits
V_{REFHI} 输入电流		280			μ A	

- (1) 在 $V_{REFHI}=2.5V$ 和 $V_{REFLO}=0V$ 下测量典型值。最小值和最大值用 $V_{REFHI}=2.5V$ 和 $V_{REFLO}=0V$ 进行测试或表征。
- (2) 参见5.8.1.3.2节。
- (3) No missing code。
- (4) 交流参数会受到时钟源精度和抖动的影响，所以系统选择时钟源时应考虑到这一点。测量这些参数时选用的时钟源是通过锁相环馈送的高精度外部时钟。片上内部振荡器具有比外部晶体更高的抖动，如果将其用作时钟源，这些参数将降低。
- (5) 尽量避免ADC输入和VREFHI引脚附近的引脚I/O活动，以减少电容耦合和串扰。
- (6) 表中结果为真实应用场景测试结果。D模块容易受数字噪声影响，在管脚61（VDD）处需要做好退耦处理，亦可以通过channel 14/15测出D模块的性能，与A/B/C模块无差异。
- (7) 采样率在6Msps以上的测试条件下ENOB可能下降0.3bit。

表 5.35 ADCEXTSOC时序要求⁽¹⁾

		MIN	MAX	UNIT
$t_w(INT)$	脉冲持续时间, INT输入低/高			
	同步	$2t_c(SYCLK)$		cycles
	带滤波功能	$t_w(IQSW)+t_w(SP)+t_c(SYCLK)$		cycles

(1) 输入限定符参数说明请参见5.7.7.2.1节。

5.8.1.3.1 ADC输入模型

请注意

在工作期间, ADC输入应保持在 $V_{DDA}+0.3$ V以下。如果ADC输入超过此电平, 器件内部的 V_{REF} 可能会受到干扰, 这可能会影响使用相同 V_{REF} 的其他ADC或DAC输入的结果。

请注意

ADC通道ADCINA0, ADCINA1和ADCINB1有一个50kΩ的下拉电阻到 V_{SSA} 。

对于差分操作, ADC输入特性由图表5.36和图5.19给出。

表 5.36 差分输入模型参数

	描述	数值(16位模式)
C_p	输入寄生电容	参见表5.38
R_{on}	采样选择电阻	700Ω
C_h	采样电容	16.5pF
R_s	标称源阻抗	50Ω

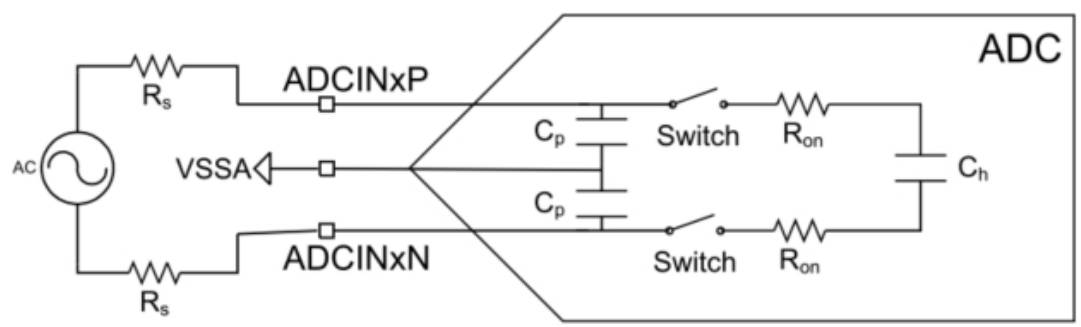


图 5.19 差分输入模型

对于单端操作，ADC输入特性由表5.37和图5.20给出。

表 5.37 单端输入模型参数

	描述	数值
C_p	输入寄生电容	参见表5.38
R_{on}	采样选择电阻	425Ω
C_h	采样电容	14.5pF
R_s	标称源阻抗	50Ω

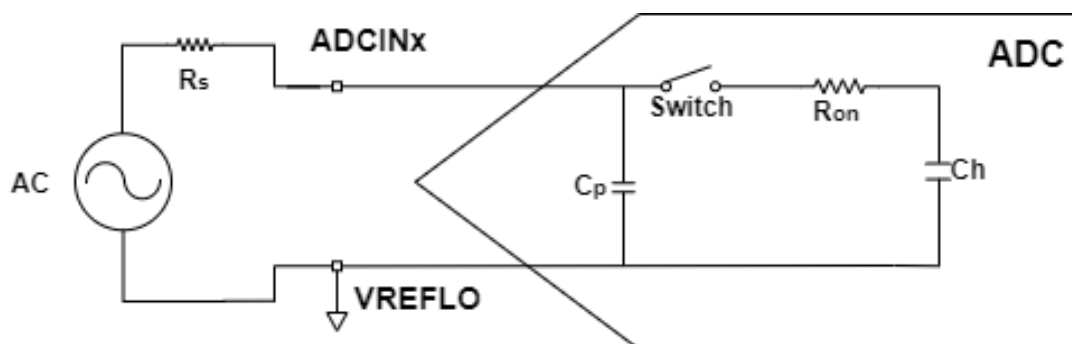


图 5.20 单端输入模型

表5.38显示了每个通道上的寄生电容。此外，启用比较器会在比较器正输入端增加约1.4 pF的电容，在比较器负输入端增加约2.5 pF的电容。

表 5.38 单通道寄生电容

ADC通道	C_p (pF)	
	比较器不使能	比较器使能
ADCINA0	12.9	N/A
ADCINA1	10.3	N/A
ADCINA2	5.9	7.3
ADCINA3	6.3	8.8
ADCINA4	5.9	7.3
ADCINA5	6.3	8.8
ADCINB0 ⁽¹⁾	117.0	N/A
ADCINB1	10.6	N/A
ADCINB2	5.9	7.3
ADCINB3	6.2	8.7
ADCINB4	5.2	N/A
ADCINB5	5.1	N/A
ADCINC2	5.5	6.9
ADCINC3	5.8	8.3
ADCINC4	5.0	6.4
ADCINC5	5.3	7.8
ADCIND0	5.3	6.7
ADCIND1	5.7	8.2
ADCIND2	5.3	6.7
ADCIND3	5.6	8.1
ADCIND4	4.3	N/A
ADCIND5	4.3	N/A
ADCIN14	8.6	10.0
ADCIN15	9.0	11.5

(1) 电容的增加是由于VDAC的功能。

这些输入模型应与实际的信号源阻抗一起使用来确定获取窗口持续时间。“选择采集窗口持续时间”的部分，请参考HS32F7D377PTI双核微控制器技术参考手册了解更多信息。

建议用户分析ADC输入电路，考虑 C_h 可能的最长充电时间，以保证在S+H窗口内， C_h 可以完全充电至 V_{REFHI} 或完全放电到 V_{REFLO} 。

5.8.1.3.2 ADC时序图

表5.40列出了12bit模式下的ADC时序。表5.41列举了16bit模式下的ADC时序。

基于以下条件，图5.21和图5.22给出了两个SOC的ADC在不同中断模式下的转换时序：

- SOC0和SOC1配置为使用相同的触发器。
- 触发器发生时，没有其他SOC正在转换或挂起。
- 轮循指针处于导致SOC0首先转换的状态。
- ADCINTSEL配置为在SOC0转换结束时设置ADCINT标志（该标志是否传播到CPU导致中断由NVIC模块中的配置决定）。

表5.39列出了图5.21和图5.22中ADC时序参数的描述。

表 5.39 ADC时序参数

参数	描述
t_{SH}	S+H窗口的持续时间。 在该窗口结束时，S+H电容上的值变为要转换为数字值的电压。持续时间由 $(ACQPS+1)$ SYSCLK周期给出。ACQPS可以为每个SOC单独配置，因此不同SOC的 t_{SH} 不一定相同。 注意:无论设备时钟设置如何，S+H电容上的值将在S+H窗口结束前大约5ns被捕获。
t_{LAT}	从S+H窗口结束到当次ADC转换结束的时间。 如果在此之前读取ADCRESULTx寄存器，则将返回先前的转换结果。
t_{EOC}	从S+H窗口结束到下一个ADC转换S+H窗口可以开始的时间。 这个时间与INTPULSEPOS=1时，ADC中断的产生的时间相关。
t_{INT}	从S+H窗口结束到设置ADCINT标志的时间(如果开启中断使能)。 如果设置了ADCCTL1寄存器中的INTPULSEPOS位， t_{INT} 将与锁存到结果寄存器中的转换结果一致。如果INTPULSEPOS位为0， t_{INT} 将与S+H窗口的末端重合。如果 t_{INT} 触发对ADC结果寄存器的读取(直接通过DMA或间接通过触发读取结果的ISR)，则必须注意确保读取发生在结果锁存之后(否则，将读取先前的结果)。
t_{CNV}	从S+H窗口结束到下一次S+H窗口产生的最短时间。 ADC允许在当前转换结果产生之前，提前产生下一次采样的S+H窗口，但是两个相邻的S+H窗口存在最小间隔时间的限制。 t_{CNV} 可通过寄存器ADCSOCxCNV配置。
t_{MUX}	通道切换到S+H窗口开始的时间。 ADC通道切换后，需等待信号稳定后再进行采样。这个信号稳定时间以SYSPCLK时钟周期为单位，可通过寄存器ADCCTL3.MUXDELAY配置。

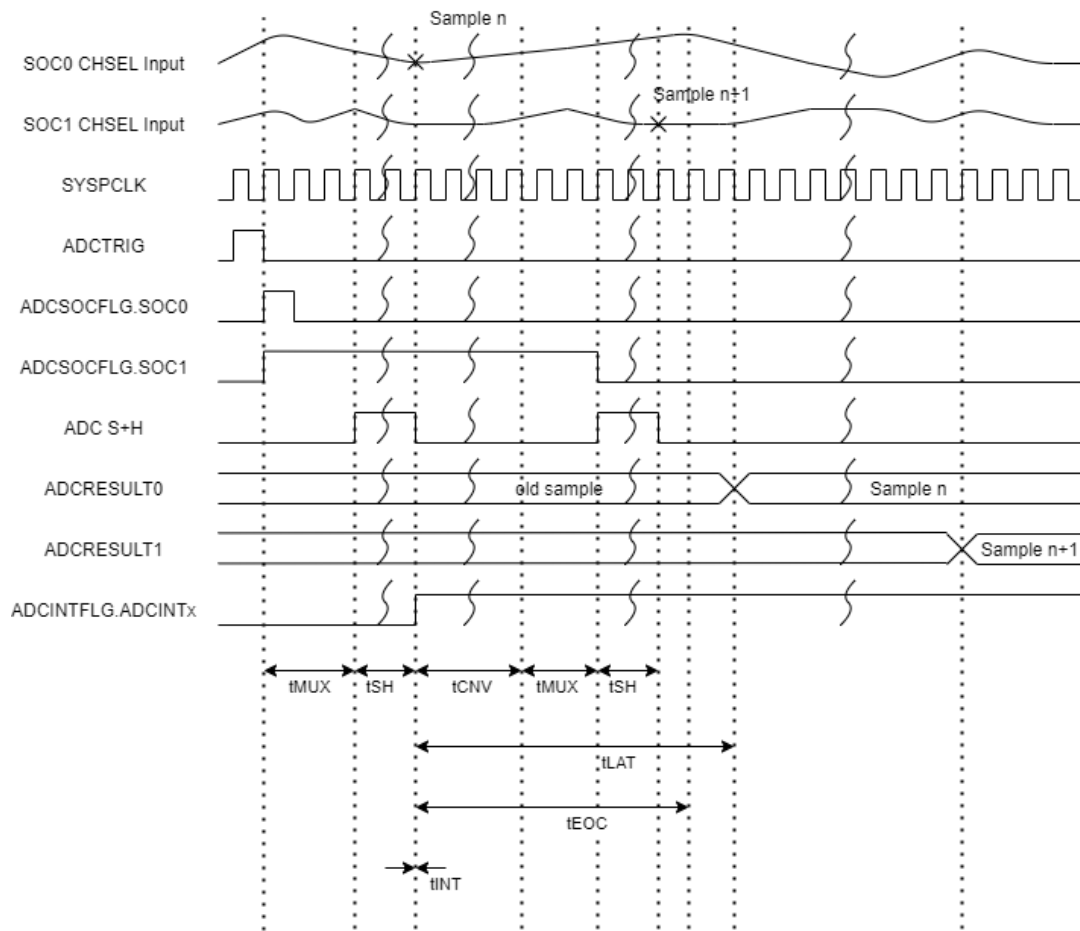


图 5.21 12bit和16bit ADC在Early Interrupt Mode下的时序

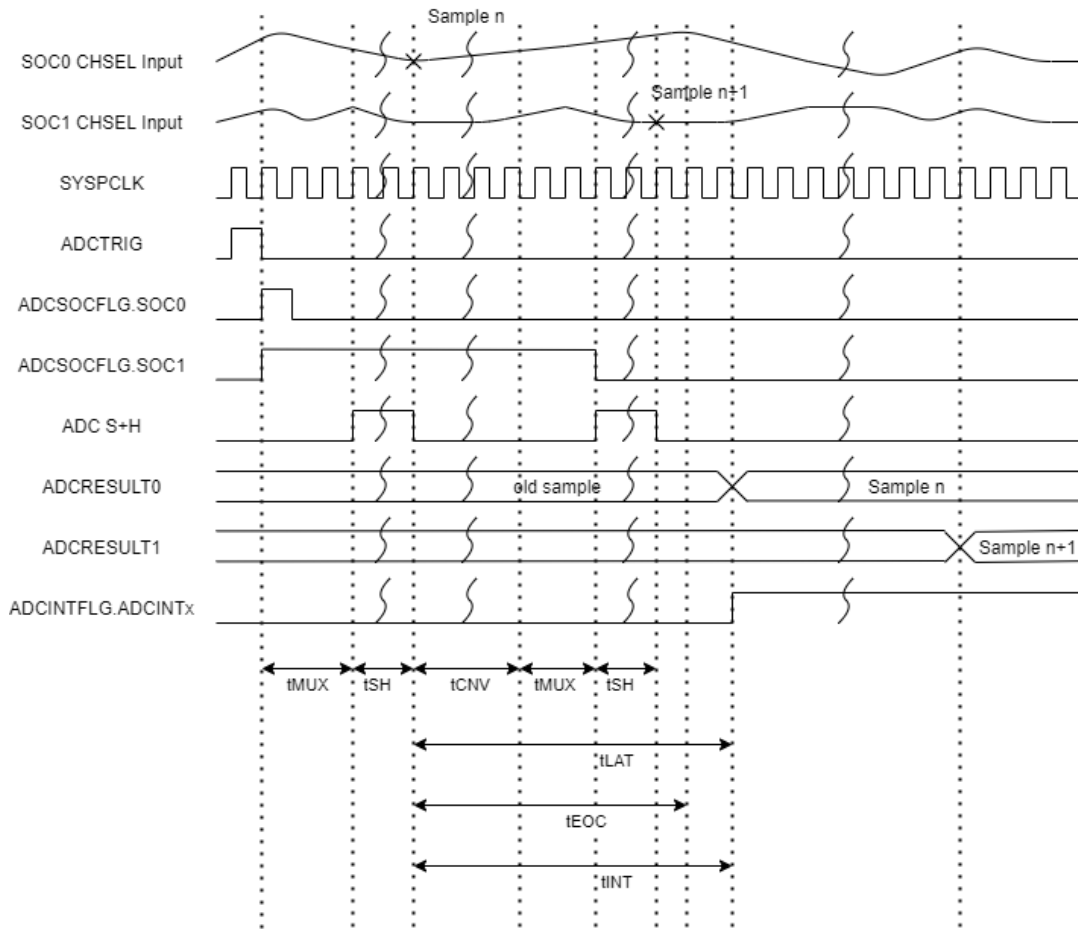


图 5.22 12bit和16bit ADC在Late Interrupt Mode下的时序

表 5.40 12bit 模式ADC时序

Parameter	Time	Comment
t_{LAT}	$t_{EOC} + 4 * T_{SYSPCLK}$	NA
t_{EOC}	245ns (at SS corner)	该转换时间根据不同的Corner、电压和温度略有变化，此处标明的是最大时间
$t_{INT}(EARLY)$	0ns	NA
$t_{INT}(LATE)$	$t_{EOC} + 4 * T_{SYSPCLK}$	NA
t_{CNV}	$ADC_{SOCx}CNV * T_{SYSPCLK}$	NA
t_{MUX}	$(ADC_{CTL3}.MUXDELAY + 3) * T_{SYSPCLK}$	除设定的MUXDELAY以外，在通道切换时设计中固有3个SYSPCLK周期的延时。

(1) 参考HS32F7D377PTI双核MCU技术参考手册中的建议。

表 5.41 16Bit 模式差分输入ADC时序

Parameter	Time	Comment
t_{LAT}	$t_{EOC} + 4 * T_{SYSPCLK}$	NA
t_{EOC}	414.5ns (at SS corner)	该转换时间根据不同的Corner、电压和温度略有变化，此处标明的是最大时间
$t_{INT}(EARLY)$	0ns	NA
$t_{INT}(LATE)$	$t_{EOC} + 4 * T_{SYSPCLK}$	NA
t_{CNV}	$ADC_{SOCx}CNV * T_{SYSPCLK}$	NA
t_{MUX}	$(ADC_{CTL3}.MUXDELAY + 3) * T_{SYSPCLK}$	除设定的MUXDELAY以外，在通道切换时设计中固有3个SYSPCLK周期的延时

(1) 参考HS32F7D377PTI双核MCU技术参考手册中的建议。

5.8.1.4 温度传感器电气数据和时序

温度传感器可用于测量器件的结温。温度传感器通过与ADC的内部连接进行采样，并通过Watech提供的软件转换为温度。对温度传感器进行采样时，ADC必须满足表5.42中的采集时间。

表 5.42 温度传感器电气特性

推荐的操作条件（除非另有说明）

参数	MIN	TYP	MAX	UNIT
温度精度		±1.5		°C
启动时间（TSNSCTL[使能]采样温度传感器）		500		μs
DC采样时间	700			ns

5.8.2 比较器子系统(CMPSS)

每个CMPSS模块包括两个比较器，两个内部电压参考DAC(CMPSS DAC)，两个数字故障滤波器和一个斜坡发生器。有CMPINxP和CMPINxN两个输入，每个输入都将在内部连接到ADCIN引脚。CMPINxP引脚总是连接到CMPSS比较器的正输入。CMPINxN可以代替DAC输出来驱动负比较器输入。有两个比较器，因此CMPSS模块的两个输出，在传递到比较器TRIP交叉条和 PWM模块或直接到GPIO引脚之前，连接到数字滤波器模块的输入。图5.23展示了HS32F7D377PTI内部的CMPSS连接。

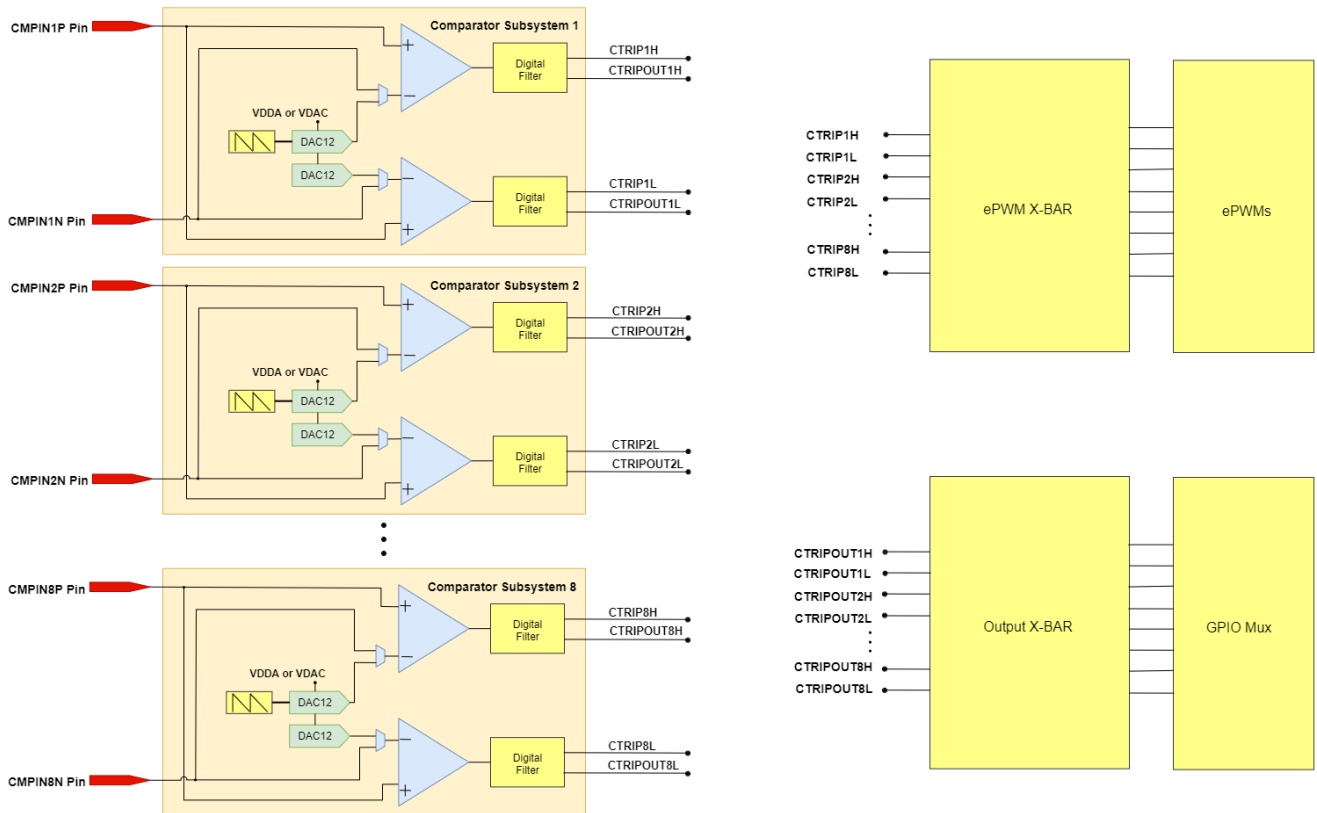


图 5.23 CMPSS连接图

5.8.2.1 CMPSS电气数据和时序

比较器电特性如表5.43所示。图5.24显示了CMPSS比较器输入参考失调量。CMPSS比较器迟滞如图5.25所示。

表 5.43 比较器电学特性

推荐的操作条件（除非另有说明）

参数	测试条件	MIN	TYP	MAX	UNIT
上电时间			3		μs
比较器输入(CMPIN _{xx})范围		0		V_{DDA}	V
输入参考偏移误差	低共模，反相输入设置为50 mV	-10		10	mV
迟滞 ⁽¹⁾	1x		12		CMPSS DAC LSB
	2x		24		
	3x		36		
	4x		48		
响应时间（从CMPIN _x 输入变化到ePWM X-BAR或X-BAR上输出的延迟）	阶跃响应		21	60	ns
	斜坡响应(1.65V/us)		26		
	斜坡响应(8.25mV/us)		30		
共模抑制比(CMRR)		40			dB

(1) 使用CMPSS DAC作为参考来确定应用多少迟滞。因此，迟滞将与CMPSS DAC参考电压成比例。迟滞可用于所有比较器输入源配置。

请注意

CMPSS输入必须保持在 $V_{DDA}+0.3$ V以下，以确保正常的功能运行。如果CMPSS输入超过这个电平，内部阻塞电路将把内部比较器与外部引脚隔离，直到外部引脚电压返回到 $V_{DDA}+0.3$ V以下。在此期间，内部比较器输入将是浮空的，并且可以在大约 $0.5\mu\text{s}$ 内衰减到 V_{DDA} 以下。在此之后，比较器可能会根据其他比较器输入的值开始输出不正确的结果。

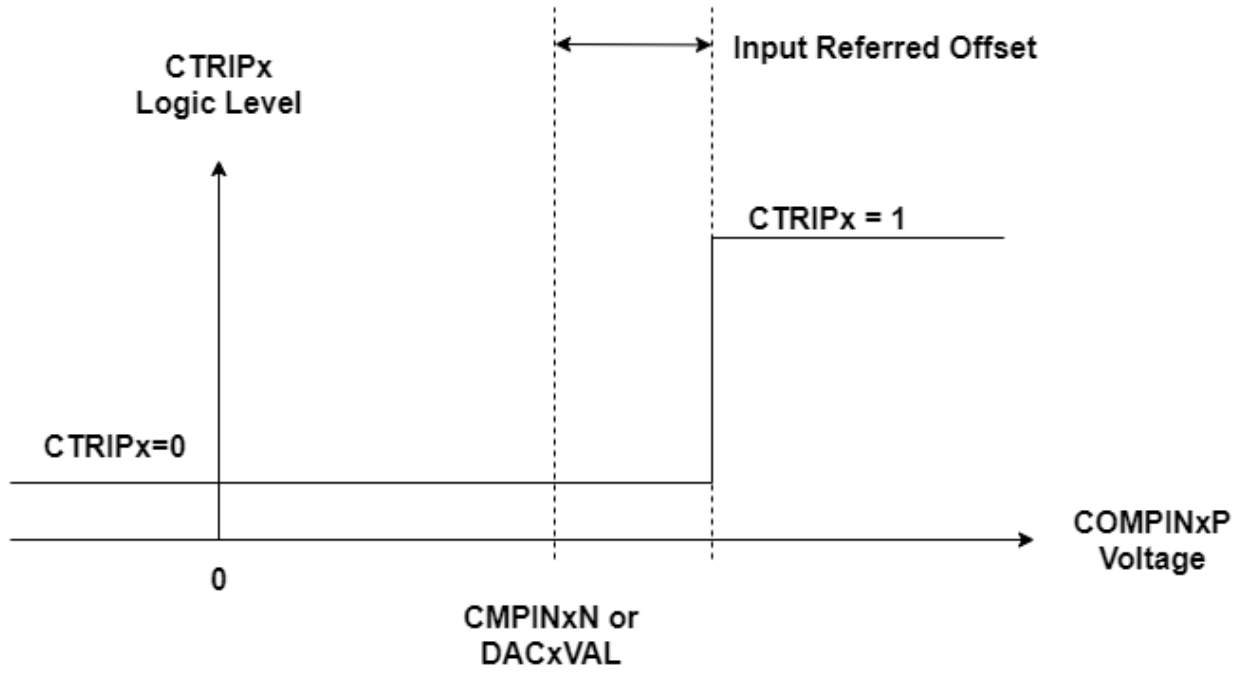


图 5.24 CMPSS比较器输入参考失调

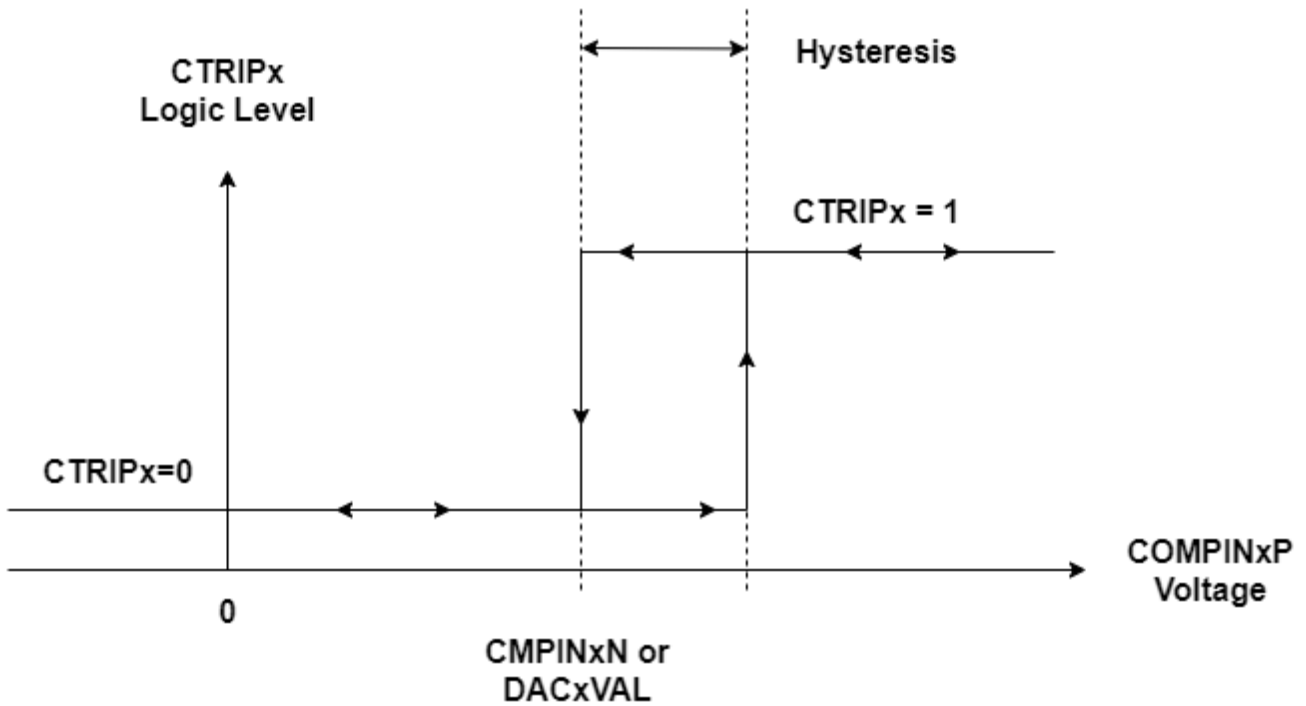


图 5.25 CMPSS比较器迟滞

CMPSS DAC的静电特性如表5.44所示。CMPSS DAC静态偏移量如图5.26所示。CMPSS DAC静态增益如图5.27所示。CMPSS DAC静态线性度如图5.28所示。

表 5.44 CMPSS DAC静态电学特性

推荐的操作条件（除非另有说明）

参数	测试条件	MIN	TYP	MAX	UNIT
CMPSS DAC输出范围	内部参考	0		$V_{DDA}^{(1)}$	V
	外部参考	0		V_{DAC}	
静态偏移误差 ⁽²⁾		-25		25	mV
静态增益误差 ⁽²⁾		-2		2	% of FSR
静态DNL	端点校正	>-1		4	LSB
静态INL	端点校正	-16		16	LSB
稳定时间	稳定到满量程输出变化1 LSB			1	μ s
分辨率			12		bits
CMPSS DAC输出扰动 ⁽³⁾	比较器跳变或同一个CMPSS模块内CMPSS DAC编码变化导致的错误	-100		100	LSB
CMPSS DAC 扰动时间 ⁽³⁾			200		ns
V_{DAC} 参考电压	以 V_{DAC} 为参考	2.4	2.5或3.0	V_{DDA}	V
V_{DAC} 负载 ⁽⁴⁾	以 V_{DAC} 为参考		6		k Ω

- (1) 当 $V_{DAC} > V_{DDA}$ 时，最大输出电压为 V_{DDA} 。
- (2) 包含比较器输入参考误差。
- (3) 在比较器跳变后的一定时间内，CMPSS DAC输出上可能存在干扰误差。
- (4) 每一个工作的CMPSS模块。

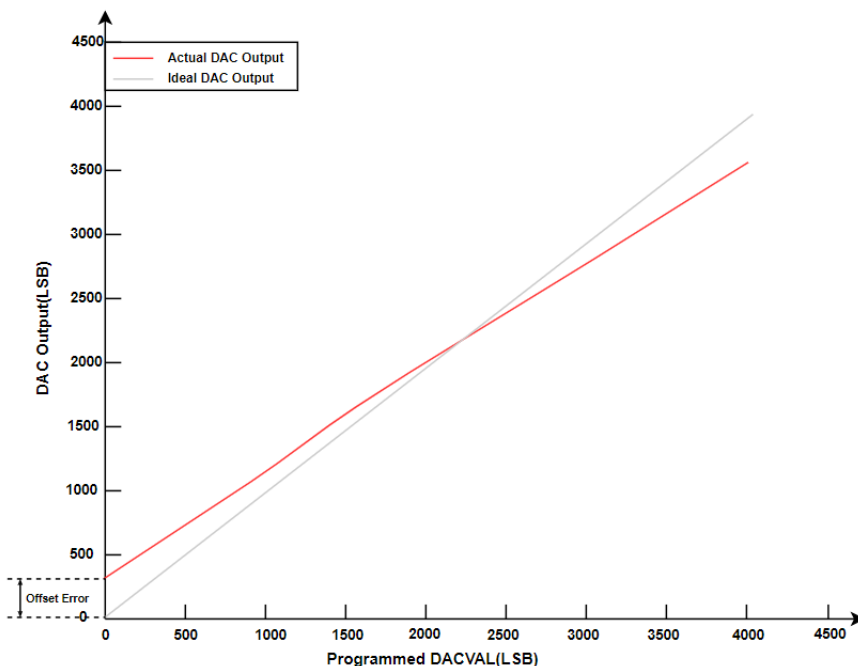


图 5.26 CMPSS DAC静态失调电压

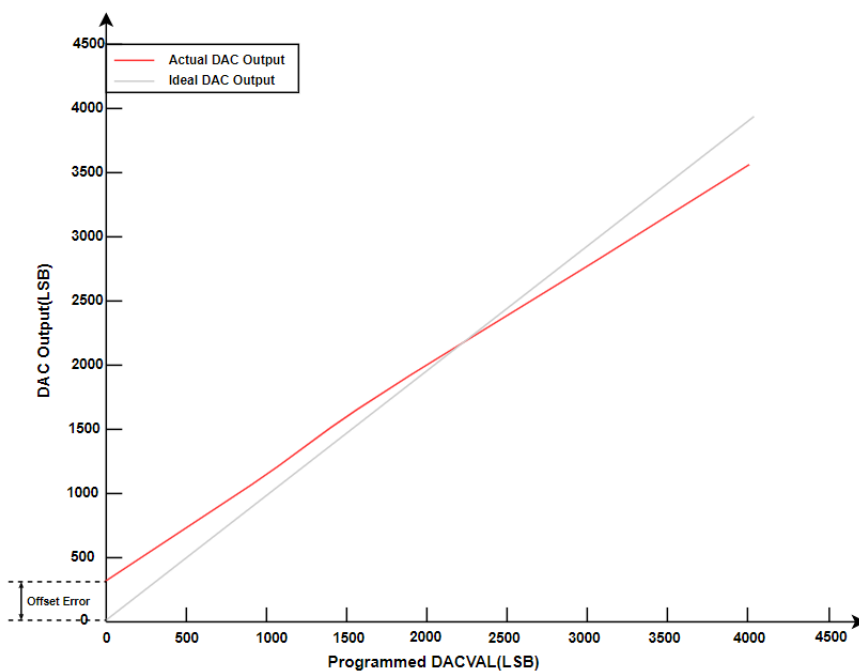


图 5.27 CMPSS DAC静态增益

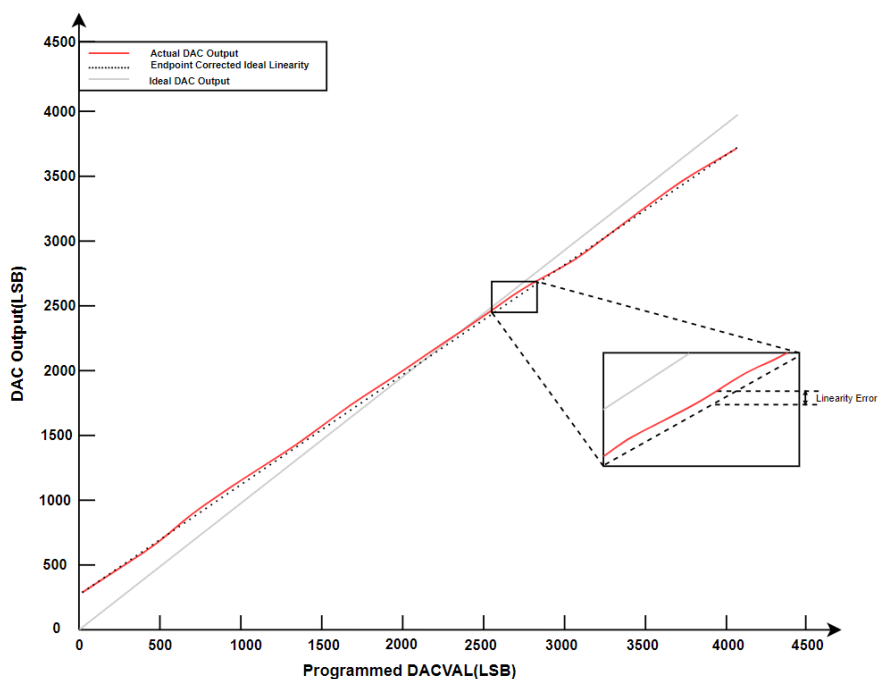


图 5.28 CMPSS DAC静态线性度

5.8.3 缓冲数模转换器(DAC)

缓冲DAC模块由一个内部12位DAC和一个能够驱动外部负载的模拟输出缓冲区组成。当输出缓冲器被禁用时，DAC输出端的集成下拉电阻有助于提供已知的引脚电压。这个下拉电阻不能被禁用，并且仍然作为引脚上的无源元件，即使对于其他共享引脚复用功能也是如此。软件对DAC值寄存器的写入可以立即生效，也可以与EPWMSYNCPER事件同步。

每个缓冲DAC具有以下特点：

- 12位可编程内部DAC
- 可选的参考电压
- 输出端下拉电阻
- 能够与EPWMSYNCPER同步

缓冲DAC的框图如图5.29所示。

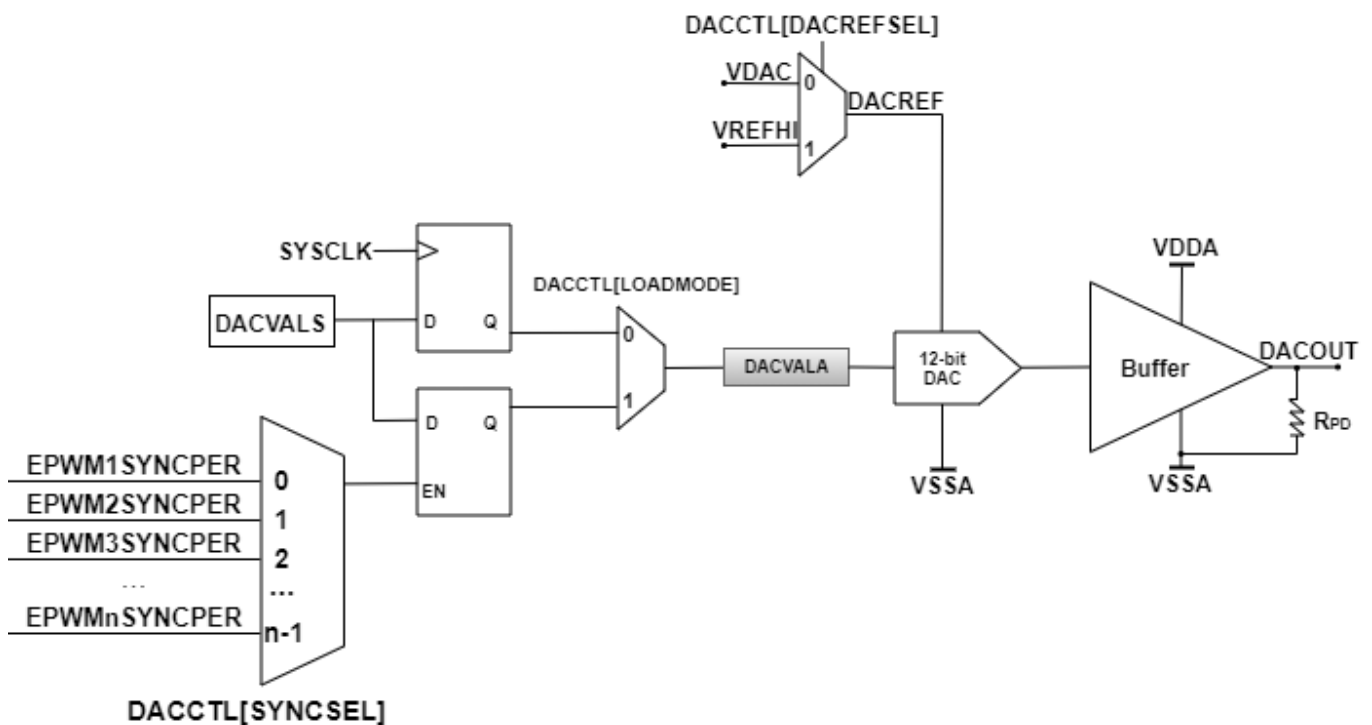


图 5.29 DAC模型框图

5.8.3.1 缓冲 DAC电气数据和时序

缓冲DAC的电气特性如表5.45所示。缓冲DAC偏移量如图5.30所示。

缓冲DAC增益如图5.31所示。缓冲DAC线性度如图5.32所示。

表 5.45 CMPSS DAC静态电学特性

推荐的操作条件（除非另有说明）⁽¹⁾

参数	测试条件	MIN	TYP	MAX	UNIT
上电时间				100	μs
偏移误差	中点	-10		10	mV
增益误差 ⁽²⁾		-2.5		2.5	% of FSR
DNL ⁽³⁾	端点校正	>-1	± 0.4	1	LSB
INL	端点校正	-5	± 2	5	LSB
DACOUTx稳定时间	稳定到0.3V到3V跳变后2 LSBs		2		μs
分辨率			12		bits
电压输出范围		0		V_{DDA}	V
负载电容	输出驱动能力			100	pF
负载电阻	输出驱动能力	5			k Ω
RPD 上拉电阻			50		k Ω
参考电压 ⁽⁴⁾	V_{DAC} 或 V_{REFHI}	2.4	2.5或3.0	V_{DDA}	V
参考输入电阻 ⁽⁵⁾	V_{DAC} 或 V_{REFHI}		170		k Ω
输出噪声	100 Hz到100 kHz的积分噪声		12.77		μV_{rms}
	10 kHz处噪声密度		47.55		$\text{nV}_{rms}/\sqrt{\text{Hz}}$
毛刺能量			1.5		V·ns
PSRR ⁽⁶⁾	DC到1kHz		70		dB
	100kHz		30		
SNR	1020Hz		67		dB
THD	1020Hz		-63		dB
SFDR	1020Hz,包括谐波和杂散		66		dBc
	1020Hz,仅杂散		104		

(1) 除另有说明外，典型值采用 $V_{REFHI}=3.3$ V测量。最小值和最大值用 $V_{REFHI}=2.5$ V测试或表征。

(2) 计算线性输出范围的增益误差。

(3) DAC输出单调。

(4) 为获得最佳PSRR性能， V_{DAC} 或 V_{REFHI} 应小于 V_{DDA} 。

(5) 每一个工作的缓冲DAC模块。

(6) $V_{REFHI}=3.2$ V, $V_{DDA}=3.3$ V DC + 100 mV Sine。

请注意

VDAC引脚必须保持在 $V_{DDA}+0.3$ V以下，以确保正常的功能操作。如果VDAC引脚超过这个电平，可能会激活阻塞电路，并且 V_{DAC} 的内部值可能会在内部浮动到0V，从而给出不正确的DAC输出。

请注意

VREFHI引脚必须保持在 $V_{DDA}+0.3$ V以下，以确保正常的功能操作。如果VREFHI引脚超过此电平，则阻塞电路可能被激活，并且VREFHI的内部值可能在内部浮动到0 V，从而导致不正确的ADC转换或DAC输出。

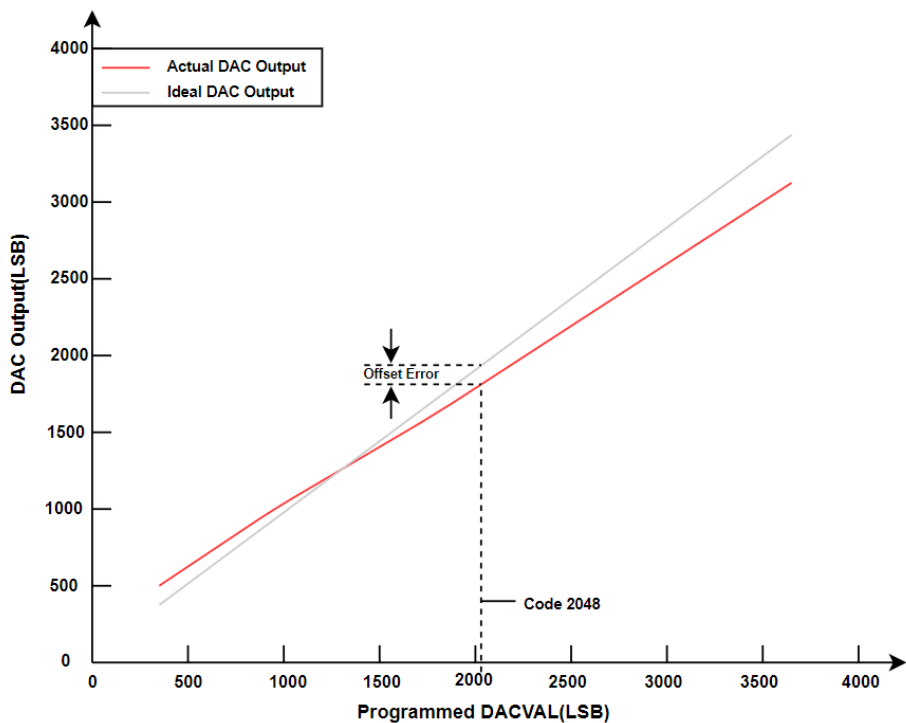


图 5.30 Buffered DAC失调电压

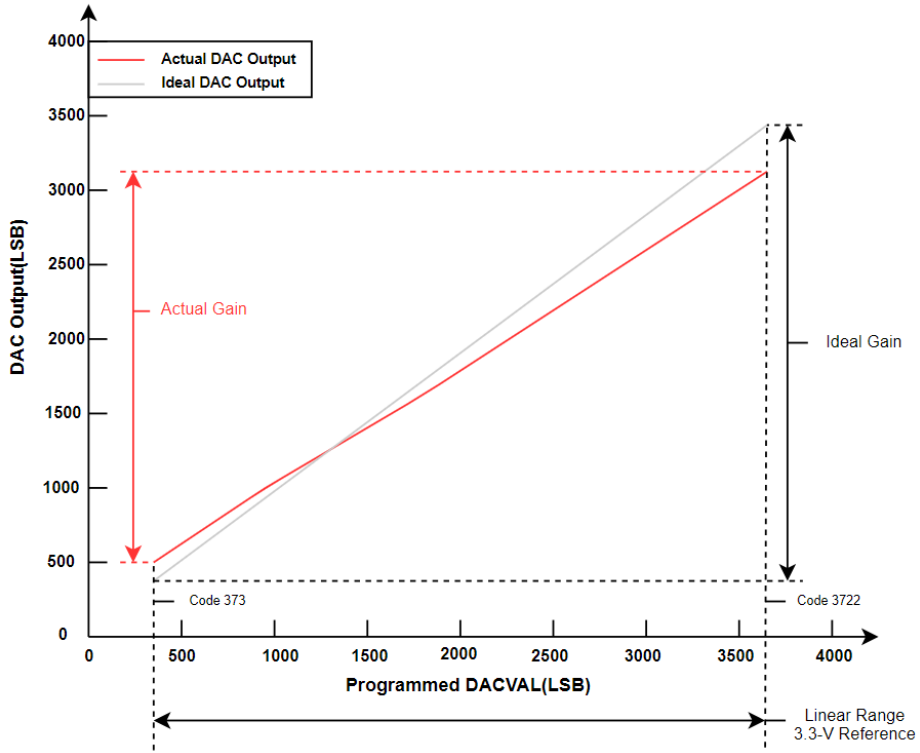


图 5.31 Buffered DAC增益

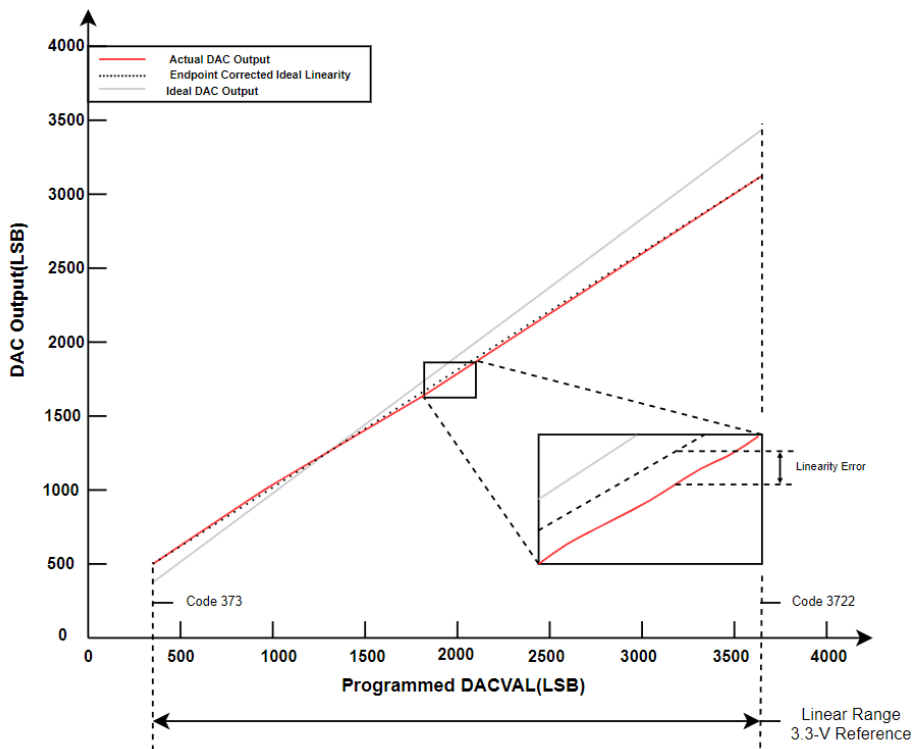


图 5.32 Buffered DAC线性度

5.9 控制外设

5.9.1 增强型捕捉模块 (eCAP)

eCAP用于精确测量两个触发事件之间的的时间。广泛应用与旋转机器的转速测量，位置传感器脉冲之间的时间测量，脉冲序列信号的周期和占空比测量等。同时eCAP支持PWM功能，可编程实现对应占空比脉冲波形。

eCAP的应用包括:

- 旋转机械的速度测量(例如，通过霍尔传感器检测齿形链轮)
- 两个位置传感器脉冲之间的时间测量
- 脉冲序列信号的周期和占空比测量
- 从占空比编码电流/电压传感器解码电流或电压幅度

eCAP模块一共集成了6组独立的通道eCAP1~6，每个eCAP通道具有以下特性:

- 每个通道具备4个时间戳寄存器;
- 触发边沿选择;
- 具有独立的中断，4次触发事件均可产生系统中断;
- 支持单次模式和连续模式，最多保存4个事件时间戳;
- 支持绝对时间捕获和增量时间捕获;
- 4个触发事件可配置为DMA触发源;
- 不使用Capture功能时，可配置为单通道APWM模式。

eCAP的输入可通过输入X-BAR连接到任何一个GPIO。APWM模式的输出可通过输出X-BAR连接到GPIO管脚。详细可见X-BAR章节。

eCAP模块由PERx.SYSCLK驱动。eCAP1~6驱动时钟可单独关闭。

5.9.1.1 eCAP与HRCAP模块框图

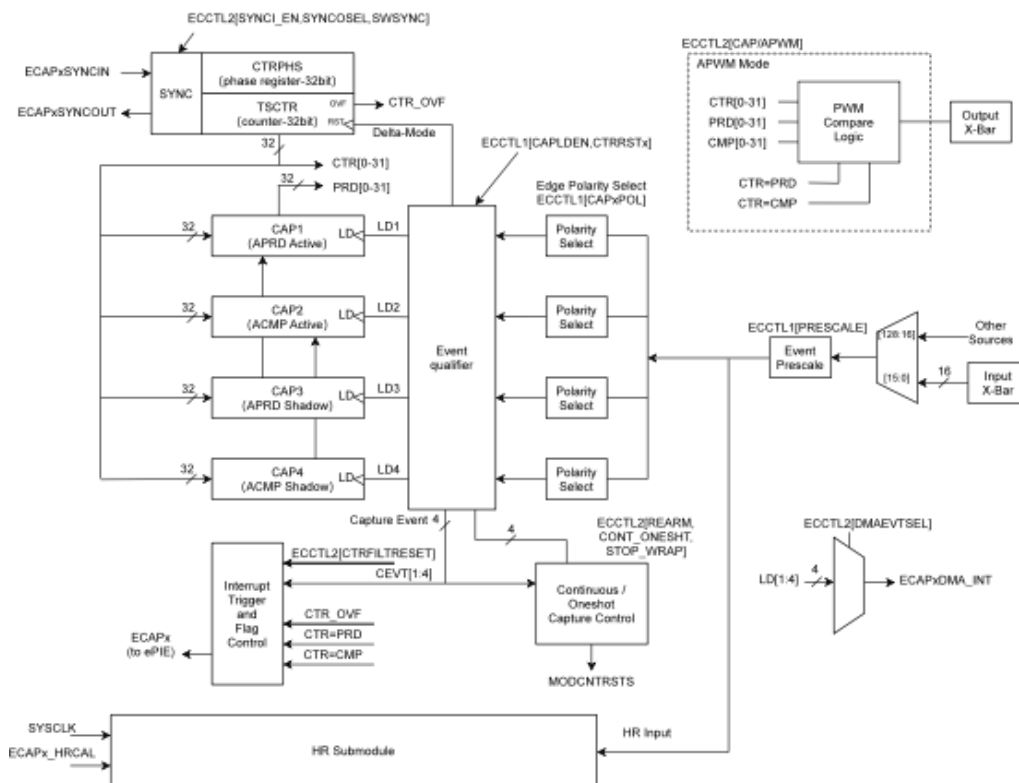


图 5.33 eCAP与HRCAP模块框图

5.9.1.2 eCAP的同步

通过选择一个通用的SYNCIN源，可以实现eCAP模块之间的同步。eCAP的SYNCIN来源可以是软件同步或外部同步。外部同步信号可以来自 EPWM或eCAP或X-Bar。同步信号由ECAPxSYNCINSEL[SEL]位中的选择定义，如图5.34所示。

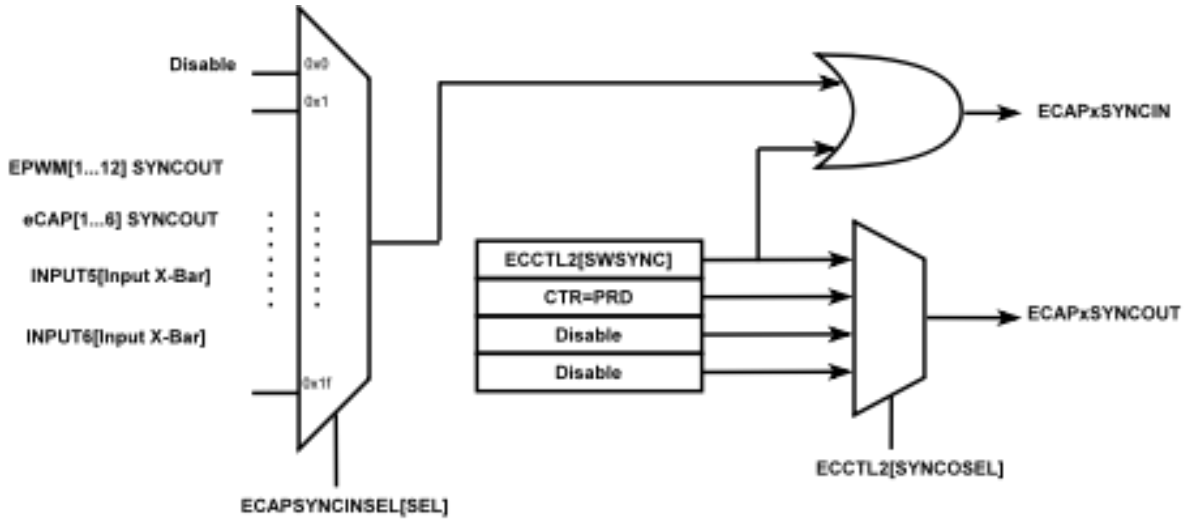


图 5.34 eCAP的同步信号

5.9.1.3 eCAP电气数据及时序说明

表 5.46 eCAP时序需求

参数	输入条件	最小值	常规值	最大值	UNIT
$t_{w(CAP)}^{(1)}$	异步输入	$2t_c(SYSCLK)$			ns
	同步输入	$2t_c(SYSCLK)$			
	使用GPIO输入整形	$1t_c(SYSCLK) + t_w(IQSW)$			

(1) $t_{w(CAP)}$ 为捕获输入脉冲宽度

表 5.47 eCAP信号电气特性

参数	最小值	常规值	最大值	UNIT
$t_w(APWM)$	20			ns

5.9.2 高分辨率输入捕获模块（HRCAP）

eCAP 0/1可配置为高分辨率捕捉模块（HRCAP）。

HRCAP的应用包括:

- 电容触摸应用
- 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
- 跨隔离边界的电压测量
- 距离/声纳测量和扫描
- 流量测量

HRCAP子模块包括以下特性:

- 绝对脉宽捕获
- 连续或单次捕捉
- 下降沿及上升沿均触发高精度捕获
- 支持硬件校准逻辑
- 支持产生校准完成中断

由于器件对温度，供电等因素非常敏感，故HRCAP内部集成了校准通路，以确保测量结果准确。HRCAP支持3种校准模式：软件校准、硬件定期校准及捕获触发校准。校准通路不影响正常的HRCAP输入捕获。

5.9.2.1 HRCAP模块框图

HRCAP集成于eCAP模块中，可参考图5.33 “eCAP与HRCAP模块框图”。

5.9.2.2 HRCAP时序

表 5.48 HRCAP时序

参数	最小值	常规值	最大值	UNIT
$t_w(HRCAP)$	300			ns

由于仅存在一组HRCAP_CAP_DATA寄存器，因此需要软件第二次捕获之前读出前一次捕获事件的HRCAP_CAP_DATA寄存器值。上表所示的是在SYSCLK配置为最高频率时，能够捕获的最小脉宽的时间。

5.9.3 增强型脉宽调制器(ePWM)

ePWM外设是控制商业和工业设备中的许多电力电子系统的关键元件。ePWM模块能够以最小的CPU开销生成复杂的脉冲宽度波形，方法是使用具有单独资源的较小模块构建外设，这些模块可以一起运行以形成一个系统。ePWM模块的一些亮点包括复杂波形生成、死区生成、灵活的同步方案、高级跳变区功能和全局寄存器重加载功能。

图5.35显示了与ePWM的信号互连。图5.36显示了ePWM跳变输入连接。

虚框中的事件由ePWM数字比较(DC)子模块根据TRIPIN输入电平生成。

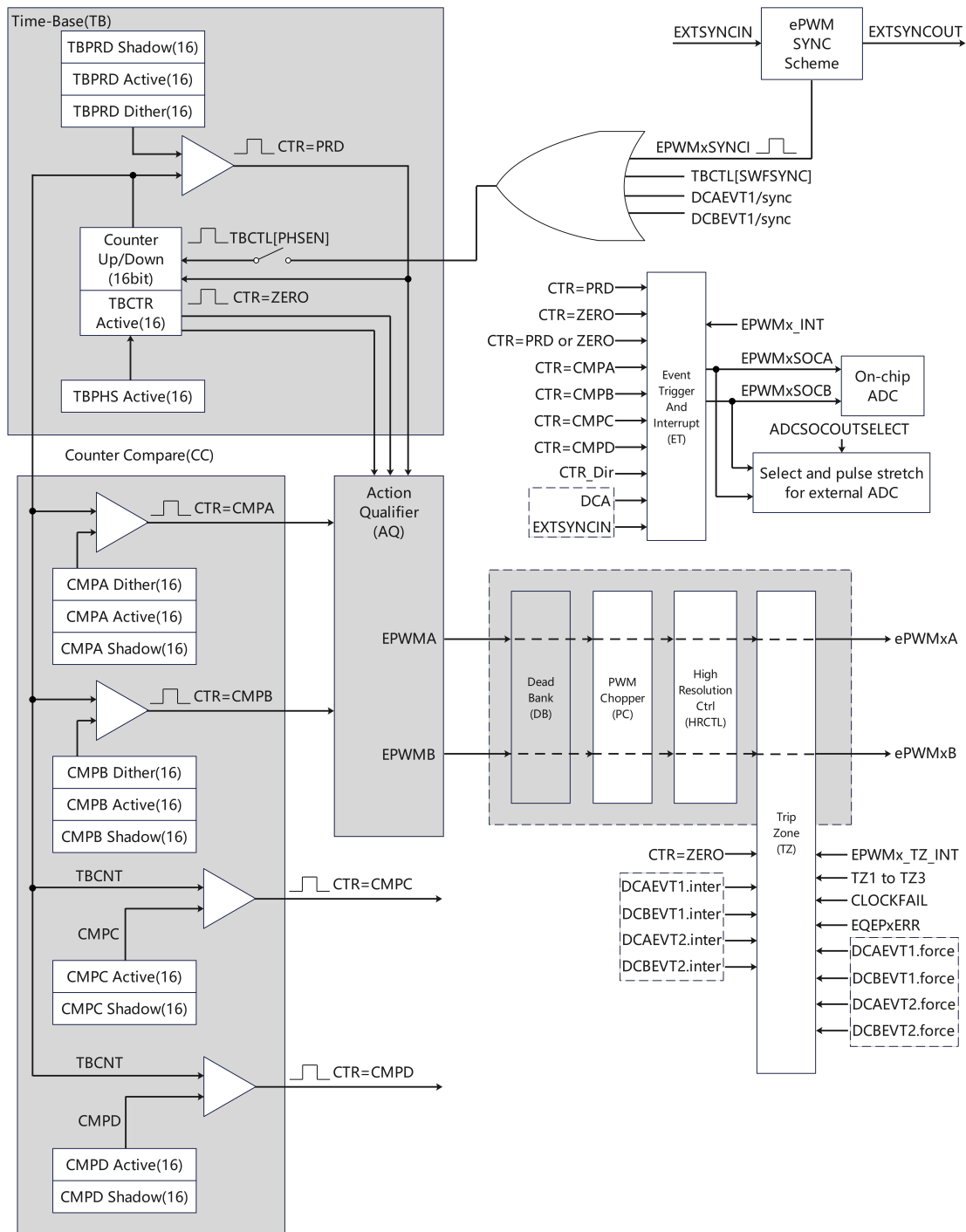


图 5.35 ePWM子模块和关键内部信号互连

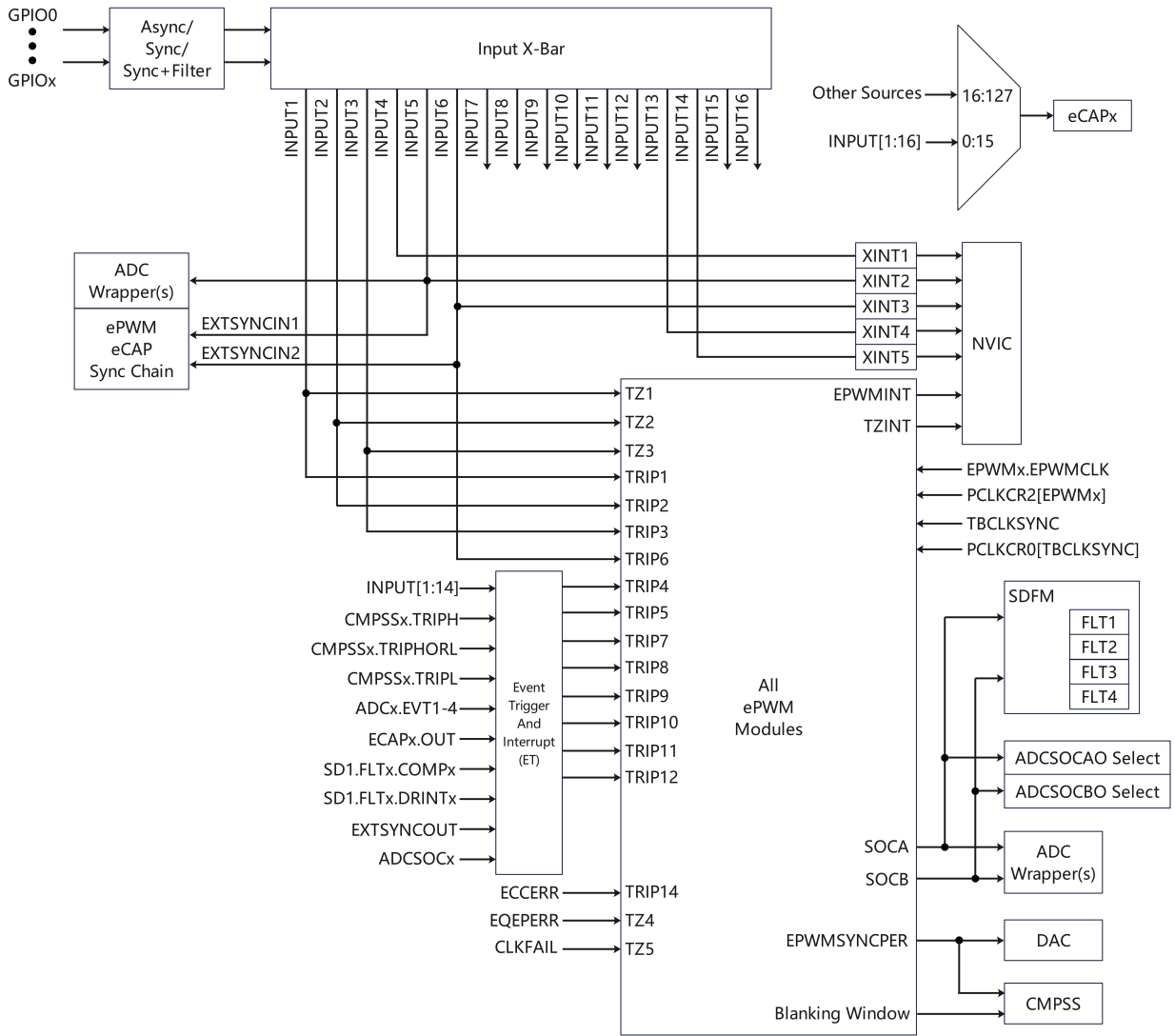


图 5.36 ePWM跳变输入连接

5.9.3.1 控制外设同步(Control Peripherals Synchronization)

器件上的ePWM和eCAP同步方案可灵活地在CPU1和CPU2之间划分ePWM和eCAP模块，并允许在属于同一CPU的模块内进行本地同步。与其他外设一样，ePWM和eCAP模块的分区需要使用CPUSELx寄存器来完成。同步方案如图5.37所示。

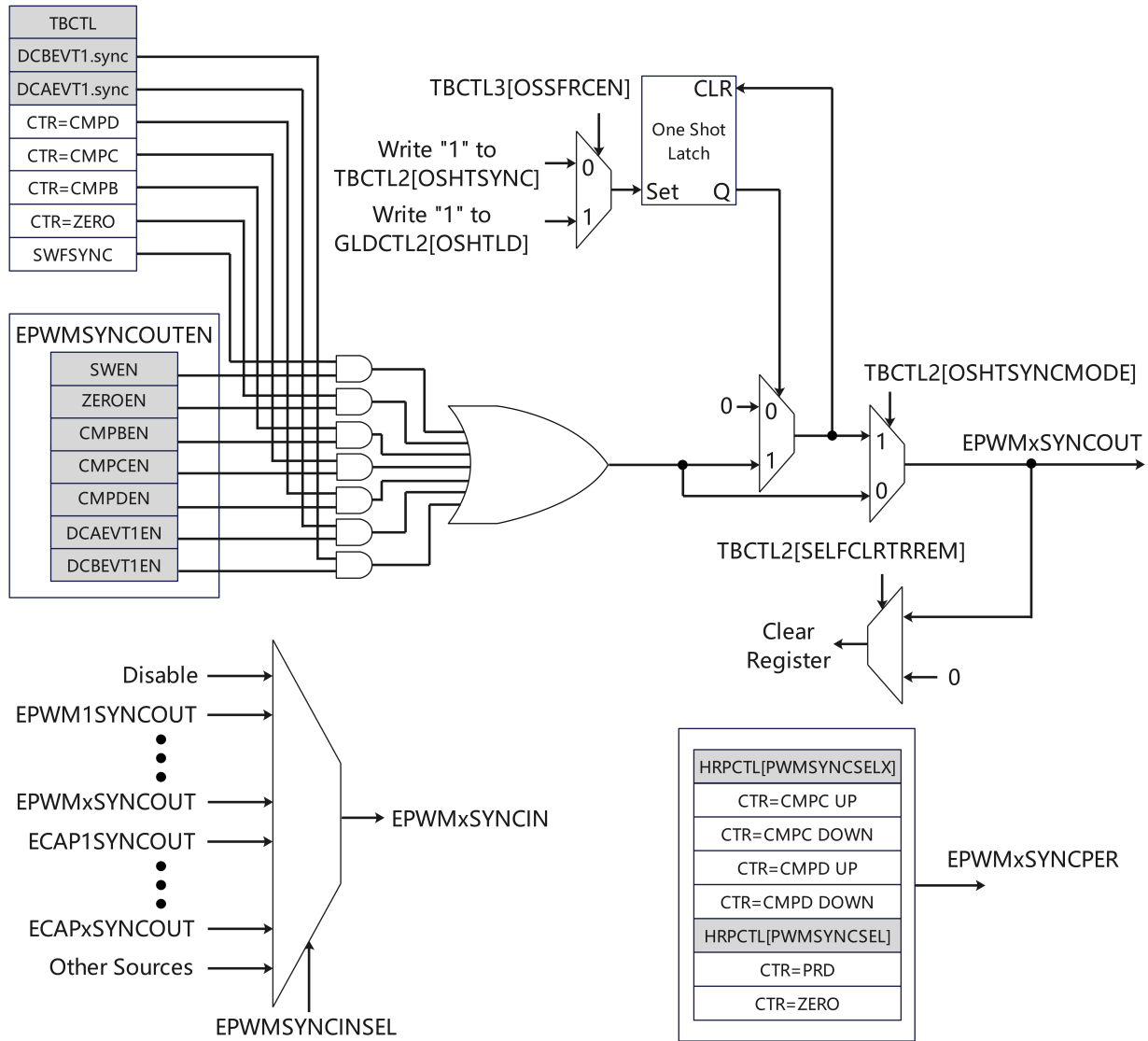


图 5.37 同步链架构

5.9.3.2 ePWM电气数据和时序

表5.49列出了PWM时序要求；表5.50列出了PWM开关特性；TZ输入请参见表5.51和图5.38；表5.52显示了外部ADC转换开始开关特性；图5.39 TZ输出时序显示了ADCSOCAO或ADCSOCBO时序。

表 5.49 ePWM时序要求

			下限	上限	单位
$f_{(EPWM)}$	EPWMCLK频率			300	MHz
$t_w(SYNCIN)$	输入sync脉冲宽度	异步	$2t_c(TBCLK)$		周期
		同步	$2t_c(TBCLK)$		周期
		带输入限定	$t_c(TBCLK) + t_w(IQSW)$		周期

表 5.50 ePWM开关特性

参数		下限	上限	单位
$t_w(PWM)$	脉冲持续时间, PWMx输出高电平/低电平	20		ns
$t_w(SYNCOU)$	同步输出脉冲宽度	$t_c(SYSCCLK)$	$65536 * t_c(SYSCCLK)$	周期
$t_d(TZ-PWM)$	延迟时间, 跳变输入有效至PWM强制高电平/低电平/高阻		30	ns
$t_{skew}(PWM)$	任意两个PWM输出之间的偏差		2.5	ns

表 5.51 TZ输入时序要求

			下限	上限	单位
$t_w(TZ)$	脉冲持续时间, TZx输入高电平	异步	$2t_c(EPWMCLK)$		周期
		同步	$2t_c(EPWMCLK)$		周期
		带输入限定	$t_c(EPWMCLK) + t_w(IQSW)$		周期

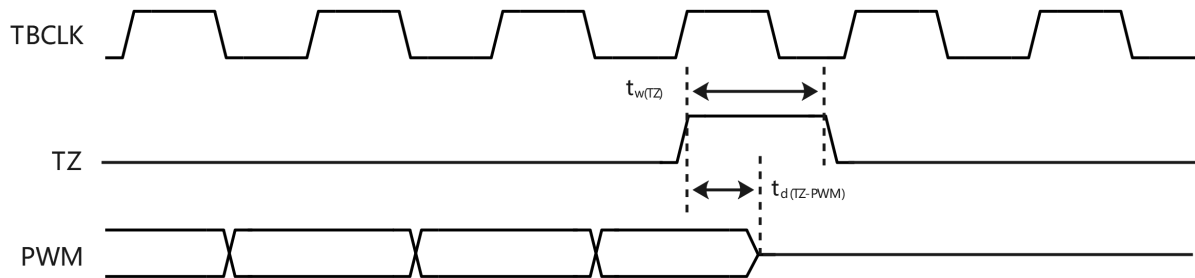


图 5.38 TZ输入时序

TZ输入包括TZ1、TZ2、TZ3, TRIP1-TRIP12。

PWM指器件中的所有PWM引脚。TZ为低电平后PWM引脚的状态取决于软件的恢复配置。

表 5.52 ADCSOCxO时序

参数		下限	上限	单位
$t_w(ADCSOCH)$	脉冲持续时间, ADCSOCxO高电平	$t_c(SYSCLK)$	$65536 * t_c(SYSCLK)$	周期

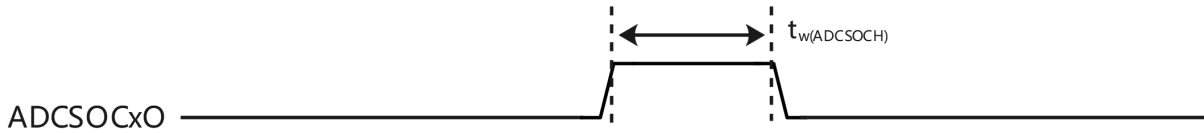


图 5.39 TZ输出时序

5.9.4 高分辨率脉宽调制器(HRPWM)

高分辨率脉宽调制器(High-Resolution Pulse Width Modulator, HRPWM)将多个延迟线组合在单个模块中, 并通过使用专用校准延迟线来简化校准系统。对于每个ePWM模块, 有两个HR输出:

- 通道A的HR占空比和死区控制。
- 通道B的HR占空比和死区控制。

HRPWM模块提供的PWM分辨率(时间粒度)明显优于传统的数字PWM方法。HRPWM模块的关键点在于:

- 极大扩展了传统数字PWM的时间分辨率能力。
- 该功能既可用于单边(占空比和移相控制), 也可用于频率/周期调制的双边控制。
- 通过扩展ePWM模块的比较A、B、相位、周期和死区寄存器, 可以控制更精细的时间粒度或边沿定位。

5.9.4.1 HRPWM电气数据和时序

表 5.53 HRPWM开关特性

参数	下限	典型	上限	单位
微边缘定位(MEP)步长 ⁽¹⁾	65	100	145	ps

(1) 在高温和VDD电压最低时, MEP步长最大。MEP步长随温度升高和电压降低而增大, 随温度降低和电压升高而减小。

5.9.5 增强型正交编码脉冲(eQEP)

eQEP与线性或旋转增量编码器连接，能够对编码器输出的多种类型（主要包括正交编码脉冲）的信号进行译码和计数，以获取旋转机械的位置、方向和速度信息，目前广泛应用于高性能运动和位置控制系统。

eQEP的具体功能包括：

- 支持6种解码模式，具体包括
 - 正交脉冲模式
 - 方向计数模式
 - 向上计数模式
 - 向下计数模式
 - PLC模式
 - 二进制码模式
- 支持位置计数脉冲的捕获功能
 - 捕获时钟频率支持模块时钟频率以及2,4,8,16,32,64,128分频。
 - 捕获事件频率支持事件频率以及2,4,8,16,32,64,128,256,512,1024,2048分频。
- 支持输出特定位置的脉冲信号到管脚
 - 支持选择输出到Index管脚或Strobe管脚
 - 支持4096档可配置的脉冲宽度，时间细度为系统时钟周期的4倍
- 支持32位的单位时间计数功能
- 支持16位的看门狗计数器功能
- 支持12种中断类型，监测模块及计数状态
 - 各个中断支持独立配置使能和清除
 - 调试模式下，支持软件配置独立触发

eQEP模块主要包括以下功能单元（如图5.40所示）：

- 每个管脚支持可编程的输入滤波功能（具体参考GPIO MUX相关章节）
- 脉冲解码单元（PDU, Pulse Decoder Unit）
- QEP控制单元（QEPC, QEP Controller）
- 脉冲捕获单元（PCU, Pulse Captor Unit）
- 自检测单元（SMU, Self Monitor Unit）

5.9.6 Σ - Δ 滤波器模块(SDFM)

Σ - Δ 滤波器模块(Sigma-Delta Filter Module, SDFM)是一款四通道数字滤波器，专为电机控制应用中的电流测量和旋转变压器位置解码而设计。每个输入通道可接收一个独立的 Σ - Δ 调制比特流。位流由四个可单独编程的数字抽取滤波器处理。滤波器组包括一个快速比较器（二级滤波器），用于过流和欠流监测的即时数字阈值比较以及过零检测。图5.41显示了SDFM的框图。

SDFM特性包括：

- 每个SDFM模块8个外部引脚：
 - 每个SDFM模块4个 Σ - Δ 数据输入引脚（SD-D_x，其中x = 1至4）。
 - 每个SDFM模块4个 Σ - Δ 时钟输入引脚（SD-C_x，其中x = 1至4）。
- 支持可配置的调制器时钟模式：
 - 调制器时钟速率等于调制器数据速率。
 - 调制器时钟速率为调制器数据速率的一半。
 - 调制器时钟速率是调制器数据速率的两倍。
- 每个SDFM模块有4个独立的、可配置的二级滤波器（比较器）单元：
 - 可选择四种不同的滤波器类型（Sinc1/Sinc2/Sincfast/Sinc3）。
 - 能够检测过值条件、欠值条件和阈值交叉条件。
 - * 两个独立的较高阈值比较器（用于检测超值情况）。
 - * 两个独立的较低阈值比较器（用于检测欠值情况）。
 - * 1个独立的阈值-交叉比较器（用于使用eCAP测量占空比/频率）。
 - 比较器滤波器单元的过采样值（COSR）可编程为1至32。
- 每个SDFM模块有4个独立可配置的初级滤波器（数据滤波器）单元：
 - 4种不同的滤波器类型选择（Sinc1/Sinc2/Sincfast/Sinc3）可供选择。
 - 数据滤波器单元的过采样值(DOSR)可编程为1至256。
 - 能够启用或禁用（或同时启用或禁用）单个滤波器模块。
 - 能够通过使用主滤波器使能(MFE)位或使用PWM信号同步SDFM模块的所有4个独立滤波器。
- 数据滤波器输出可以16位或32位表示。
- 数据滤波器单元具有可编程模式FIFO，以减少中中断开销。FIFO具有以下特点：
 - 主滤波器（数据滤波器）有一个16深x32位的FIFO。
 - FIFO可以在可编程的数据就绪事件后中断CPU。
 - FIFO等待同步功能：能够忽略数据就绪事件，直到收到PWM同步信号（SDSYNC）。一旦接收到SDSYNC事件，FIFO将在每个数据就绪事件中填满。
 - 数据滤波器输出可以16位或32位表示。
- PWMx.SOCA/SOCB可配置为每个数据滤波器通道的SDSYNC源。
- PWM可用于为 Σ - Δ 调制器生成调制器时钟。
- SD-C_x和SD-D_x均具有可配置的输入鉴定功能。
- 能够使用一个滤波器通道时钟（SD-C1）为其他滤波器时钟通道提供时钟。
- 在比较器滤波器事件上可配置数字滤波器，以消除杂散噪声引起的比较器事件。

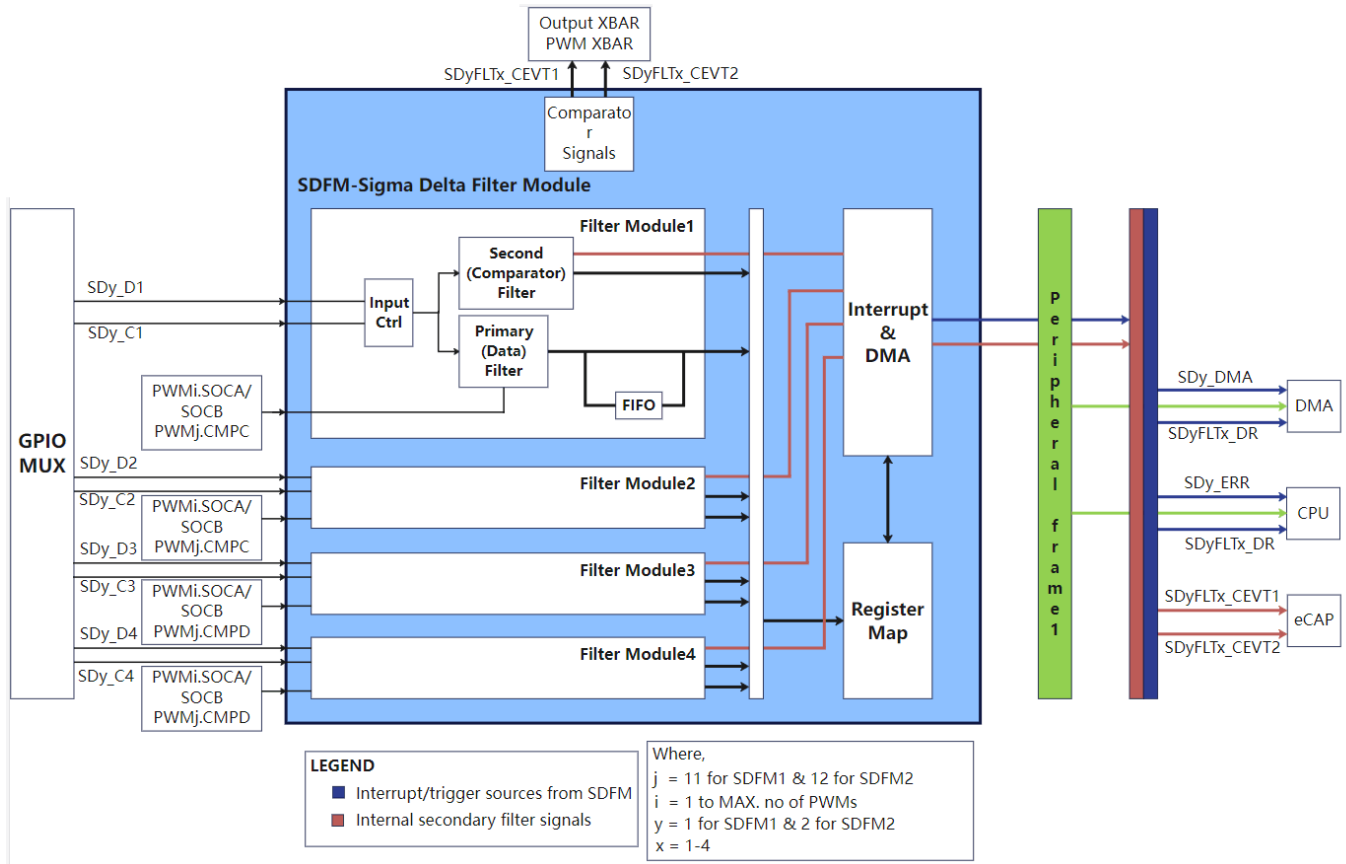


图 5.41 SDFM框图

5.9.6.1 SDFM电气数据和时序（使用ASYNC）

通过设置GPyQSELn=0b11来定义异步GPIO的SDFM操作。表5.56列出了使用异步GPIO(ASYNC)选项时的SDFM时序要求。图5.42至图5.44显示了SDFM时序图。

表 5.56 使用异步GPIO(ASYNC)选项时的SDFM时序要求

模式			下限	上限	单位
模式0	$t_c(SDC)M0$	SDx_Cy周期时间	40	128*SYSCLK周期	ns
	$t_w(SDCH)M0$	脉冲持续时间, SDx_Cy 高电平时间	10	$t_c(SDC)M0-10$	ns
	$t_{su}(SDDV-SDCH)M0$	SDx_Cy变高前SDx_Dy的建立时间	5		ns
	$t_h(SDCH-SDD)M0$	SDx_Dy在变高后SDx_Dy的保持时间	5		ns
模式1	$t_c(SDC)M1$	SDx_Cy周期时间	80	256*SYSCLK周期	ns
	$t_w(SDCH)M1$	脉冲持续时间, SDx_Cy 高电平时间	10	$t_c(SDC)M1-10$	ns
	$t_{su}(SDDV-SDCL)M1$	SDx_Cy变低前SDx_Dy的建立时间	5		ns
	$t_{su}(SDDV-SDCH)M1$	SDx_Cy变高前SDx_Dy的建立时间	5		ns
	$t_h(SDCL-SDD)M1$	SDx_Cy变低后SDx_Dy的保持时间	5		ns
	$t_h(SDCH-SDD)M1$	SDx_Cy变高后SDx_Dy的保持时间	5		ns
模式3	$t_c(SDC)M2$	SDx_Cy周期时间	40	256*SYSCLK周期	ns
	$t_w(SDCH)M2$	脉冲持续时间, SDx_Cy 高电平时间	10	$t_c(SDC)M2-10$	ns
	$t_{su}(SDDV-SDCH)M2$	SDx_Cy变高前SDx_Dy的建立时间	5		ns
	$t_h(SDCH-SDD)M2$	SDx_Cy变高后SDx_Dy的保持时间	5		ns

请注意

当没有GPIO输入同步时，SDFM时钟输入（SDx_Cy引脚）直接为SDFM模块提供时钟。这些输入上的任何闪烁或振铃噪声都会破坏SDFM模块的运行。应该对这些信号采取特别的预防措施，以确保信号干净无噪声，满足SDFM的时序要求。建议采取预防措施，例如对时钟驱动器的任何阻抗不匹配而导致的振铃进行串联终止，以及将走线与其他噪声信号隔离开来。模式2时，关闭输入控制单元，不接收时钟和数据

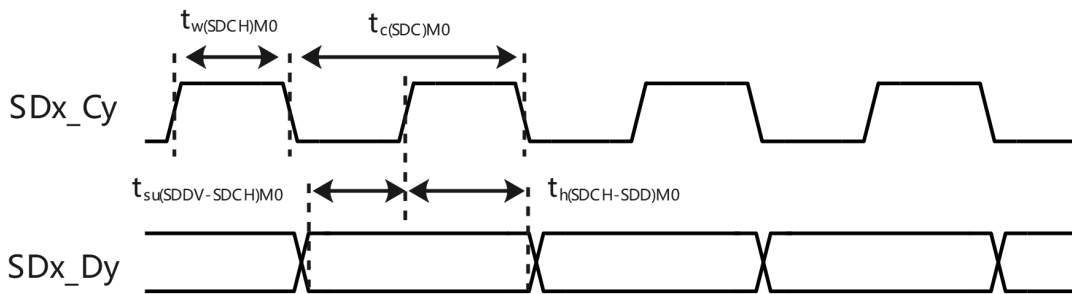


图 5.42 SDFM模式0时序图(调制器时钟速率等于调制器数据速率)

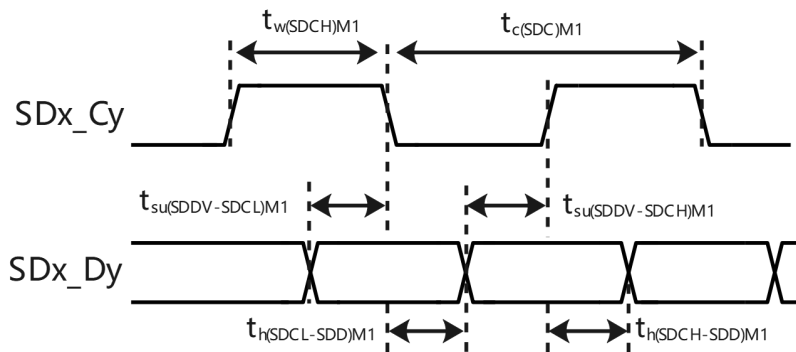


图 5.43 SDFM模式1时序图(调制器时钟速率为调制器数据速率的一半)

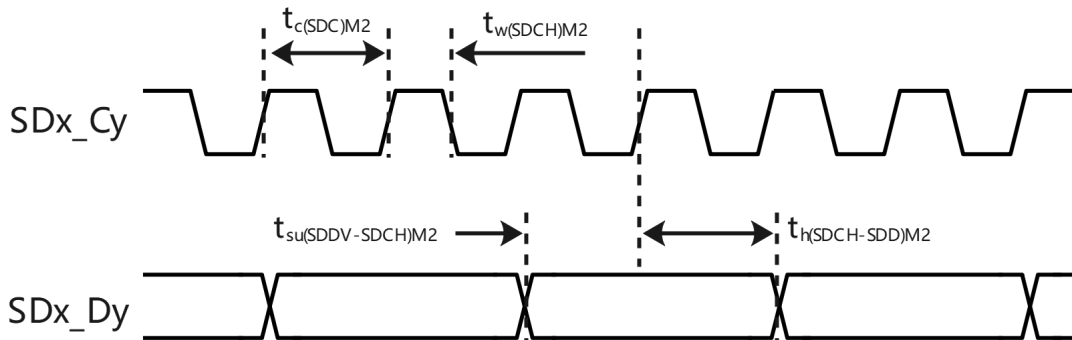


图 5.44 SDFM模式3时序图(调制器时钟速率是调制器数据速率的两倍)

5.10 通信外设

5.10.1 控制器局域网(CAN)

M_CAN遵循ISO 11898-1:2015标准（与Bosch® CAN协议标准2.0A,B一致），进行控制器局域网通信。此外，M_CAN也支持CAN FD（Flexible Data Rate），数据帧最高支持单帧64字节的数据传输，最高支持8Mbps的比特传输率。CAN模块需要配备额外的收发器硬件，才能连入实际物理层（即CAN总线网络）。

CAN网络中每一个节点的正常通信，需要专用的存储介质存储收发的各种消息帧。本设备的每个CAN模块配备了一块17KB的专属SRAM存储，支持ECC功能。

M_CAN的消息处理部分具体包括接收处理模块和发送处理模块，能够实现各种消息处理功能。其中，接收处理模块支持消息接收/过滤、数据从CAN协议逻辑单元到存储RAM的传输以及更新接收消息状态。发送处理模块支持数据从存储RAM到CAN协议逻辑单元的传输以及更新发送消息状态。

此外，CAN模块支持通过CPU直接配置寄存器，控制CAN协议逻辑、消息处理逻辑以及SRAM访问配置。

CAN模块支持的功能包括：

- 遵循ISO 11898-1:2015协议标准
- 支持CAN FD协议，单帧数据最高支持64 bytes
- CAN错误日志功能
- 升级型消息接收/过滤
- 两个可配置的RX FIFO
- 支持对于高优先级的消息进行独立采样
- 64个专用的接收缓存
- 32个专用的发送缓存
- 可配置的TX FIFO
- 可配置的TX队列
- 可配置的TX 事件FIFO
- 支持CPU直接访问消息SRAM
- 每个CAN配置专用的17KB SRAM
- 可配置的回环检测模式
- 可配置的中断掩码
- CAN协议时钟和逻辑时钟独立
- 支持CAN debug测试功能

请注意

对于200 MHz的CAN位时钟，可能的最小比特率为7.8125 kbps。

请注意

根据所使用的时序设置，片上零引脚振荡器（数据手册中指定）的精度可能无法满足CAN协议的要求。在这种情况下，必须使用外部时钟源。

图5.45 CAN模块示意图为CAN模块示意图。

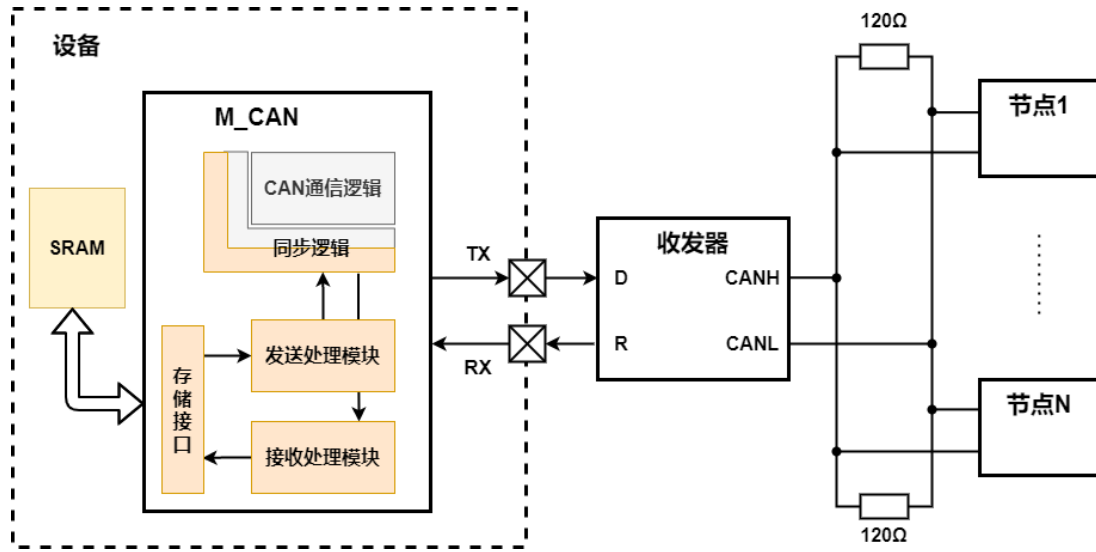


图 5.45 CAN模块示意图

5.10.2 I2C (SMBus/PMBus)

I2C(SMBus/PMBus)模块框图如图5.46所示。

I2C(SMBus/PMBus)具有以下特性：

- 双线i2c串行接口——由串行数据线（SDA）和串行时钟（SCL）组成
- 支持8位格式传输
- 两种数据传输速度：
 - 标准模式（0-100kb/s）
 - 快速模式（ $\leq 400\text{kb/s}$ ）或快速模式plus（ $\leq 1000\text{kb/s}$ ）
- 时钟同步
- General Call
- Master或Slave模式
- 支持多个master-transmitter和slave-receiver
- 7位或10位寻址
- 7位或10位组合格式传输
- 发送和接收Buffer
- 中断或轮询模式操作
- 支持可配置软件驱动程序的组件参数
- 可编程SDA保持时间（tHD;DAT）
- 支持SMBus/PMBus
- SMBus规范3.0兼容：
 - 支持ACK控制的硬件PEC（报文错误检查）生成和验证
 - 支持命令和数据确认控制
 - 支持Host和Device
 - SMBus Alert
 - SMBus SUSPEND
 - 超时和空闲状态检测

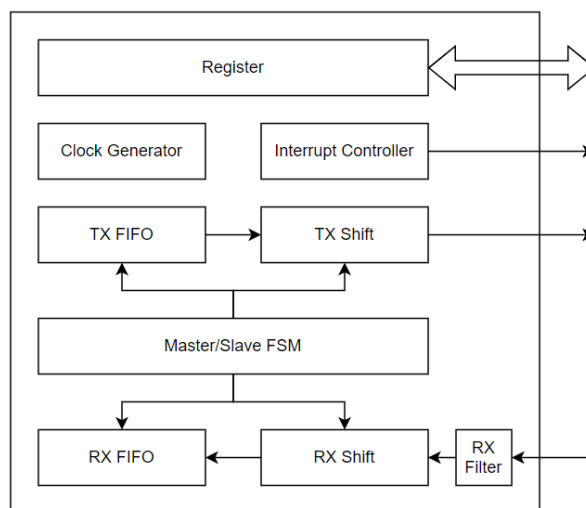


图 5.46 I2C(SMBus/PMBus)模块框图

5.10.3 UART

UART是一个可编程的通用异步接收/发送器。

UART功能特性如下：

- APB接口允许集成到SoC
- 9 位串行数据支持
- 错误起始位检测
- 可编程小数波特率支持
- 支持多分支RS485接口
- 可配置以下参数：
 - PB数据总线宽度32
 - 发送和接收FIFO深度为128
 - 内部FIFO
 - 使用pclk
 - FIFO访问模式，用于FIFO测试，使master能够写入接收FIFO，并从发送FIFO读取
 - Transmitter THRE中断模式
 - busy功能
- 能够在实例化期间设置一些配置参数
- 基于16550行业标准的功能
 - 可编程字符属性，例如：
 - * 每个字符的数据位数（5-8）
 - * 可选奇偶校验位
 - * 停止位数（1、1.5或2）
 - 优先中断识别
- 可编程FIFO启用/禁用
- 可编程串行数据波特率，计算公式如下：波特率=(串行时钟频率)/(16×除数)
- 每个时钟域单独的系统复位，以防止亚稳态

UART框图如图5.47所示：

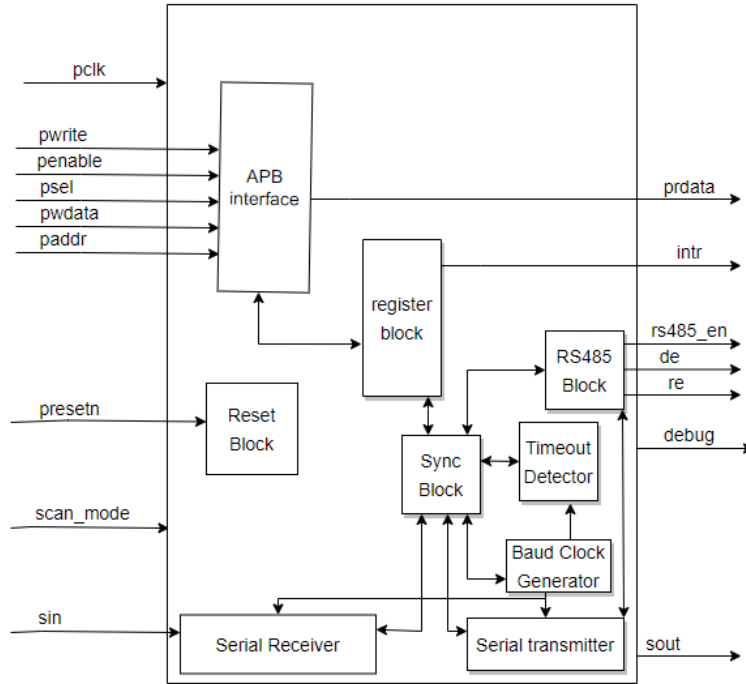


图 5.47 UART框图

5.10.4 串行外设接口(SPI)

SPI是一种高速同步串行输入/输出 (I/O) 端口，它允许编程长度 (4到32位) 的串行bit流以已编程的位传输速率移进和移出设备。SPI通常用于微控制器与外部外设或其他控制器之间的通信。SPI框图如图5.48所示。

SPI模块的功能包括：

- SPISOMI: SPI从输出/主输入引脚
- SPISIMO: SPI从输入/主输出引脚
- SPISTE: SPI从发送使能引脚
- SPICLK: SPI串行时钟引脚
- 支持master和slave两种模式，通过寄存器配置切换
- 支持SPI Standard和Dual模式
- 支持Motorola SPI和TI SSP两种帧格式
- 支持收发、只收、只发、EEPROM读模式
- 帧长度支持4-32 bit
- 深度为256的RX FIFO，深度为32的TX FIFO
- Master模式下波特率支持0-65534之间的偶数值
- Slave模式下支持采样时钟最低为通信时钟SCLK的四倍频
- 支持4种phase和polarity的组合配置，包括：
 - 下降沿无相位延迟：SPICLK有效高。SPI在SPICLK信号的下降沿传输数据，在SPICLK信号的上升沿接收数据。
 - 下降沿相位延迟：SPICLK有效高。SPI在SPICLK信号下降沿前半个周期发送数据，并在SPICLK信号下降沿接收数据。

- 上升沿无相位延迟：SPICLK无效低。SPI在SPICLK信号的上升沿传输数据，在SPICLK信号的下降沿接收数据。
- 上升沿相位延迟：SPICLK无效低。SPI在SPICLK信号上升沿前半个周期发送数据，在SPICLK信号上升沿接收数据。
- 支持RX采样延时，最高支持256
- 收发具有独立的DMA握手接口
- 单一的中断触发

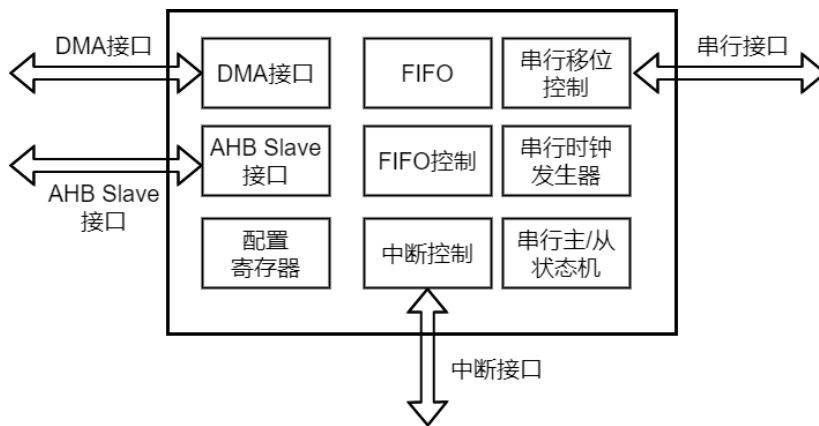


图 5.48 SPI框图

5.10.5 通用串行总线（USB）控制器

在与USB主机或从设备的点对点通信期间，USB控制器作为全速或低速控制器运行。USB框图如图5.49所示。USB模块具有以下特性：

- 通用特性
 - USB 2.0全速和低速操作
 - 支持片上全速PHY接口
 - 四种传输类型：控制，中断，批量和同步传输
 - 内嵌DMA，并可支援Scatter/Gather DMA模式。
 - 具有采用高级FIFO控制的4 KB专用 RAM：
 - * 可将RAM空间划分为不同 FIFO，以便灵活有效地使用RAM
 - * 每个FIFO可存储多个数据包
 - * 动态分配存储区
 - * FIFO大小可配置为非2的幂次方值，以便连续使用存储单元
 - 一帧之内无需应用程序干预，以达到最大USB带宽。
- 主机模式特性
 - 需要由外部电路板提供Vbus。
 - 多达16个主机通道（又称之为管道）：每个通道都可以动态实现重新配置，可支持任何类型的USB传输。
 - 内置硬件调度器可在周期性硬件队列中存储多达16个中断加同步传输请求，以及在非周期性硬件队列中存储多达16个控制加批量传输请求。
 - 管理一个共享Rx FIFO、一个周期性传输Tx FIFO和一个非周期性传输Tx FIFO，以有效使用USB数据RAM。
- 从机模式特性
 - 不支持片上速率选择硬件电路。
 - 1个双向控制端点。
 - 8个IN端点，可配置为支持批量传输、中断传输或同步传输。
 - 8个OUT 端点，可配置为支持批量传输、中断传输或同步传输。
 - 管理一个共享Rx FIFO和一个Tx-OUT FIFO，以高效使用USB数据RAM。
 - 管理多达9个专用Tx-IN FIFO（分别用于每个使能的IN EP），以降低应用程序负荷。

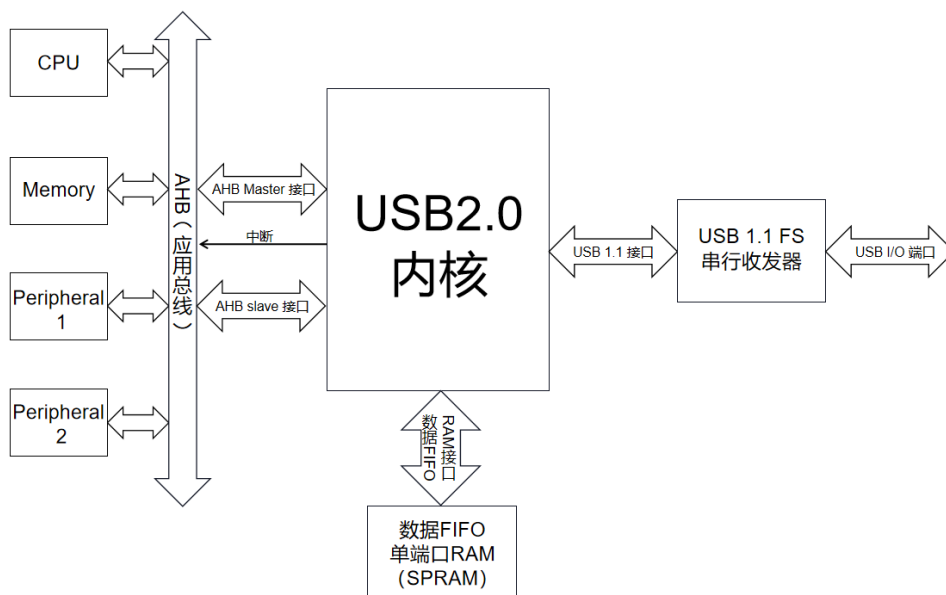


图 5.49 USB框图

6 详细规格

6.1 概述

该器件是一款功能强大的32位浮点微控制器（MCU），专为工业电机驱动器等高级闭环控制应用而设计、太阳能逆变器和数字电源、电动车辆和运输、以及传感和信号处理。提供数字电源和工业驱动器的完整开发包。该器件支持双核CPU 架构，显著提高了系统性能。集成的模拟和控制外设可以让设计者能够整合控制架构，并消除高端系统中多MCU的使用。

该32位浮点MCU可以支持双实时控制系统，每个核心可以提供300 MHz的信号处理能力。TFU加速器进一步提升了算法处理能力，使变换和转矩回路计算中常见的三角函数计算能够快速执行。

该MCU支持高达2MB的eFlash和高达1056KB的SRAM。

该MCU集成了高性能模拟和控制外设，以进一步实现系统整合。4个独立的16位ADC提供对多个模拟信号的精确高效管理，最终提高了系统吞吐量。新的 $\Sigma - \Delta$ 滤波器模块（SDFM）可以与 $\Sigma - \Delta$ 调制器协同工作，以实现隔离的电流分量测量。具有窗口比较器的比较器子系统（CMPSS）允许在超过或不满足限流条件时对功率级进行保护。该MCU还包含其他模拟和控制外设，包括DAC、PWM、CAP、QEP等。

EMIF、CAN-FD模块等外设扩展了该MCU的连接能力。USB 2.0接口让用户可以轻松地将通用串行总线（USB）连接添加到他们的应用程序中。

6.2 功能框图

功能框图如图6.1所示。

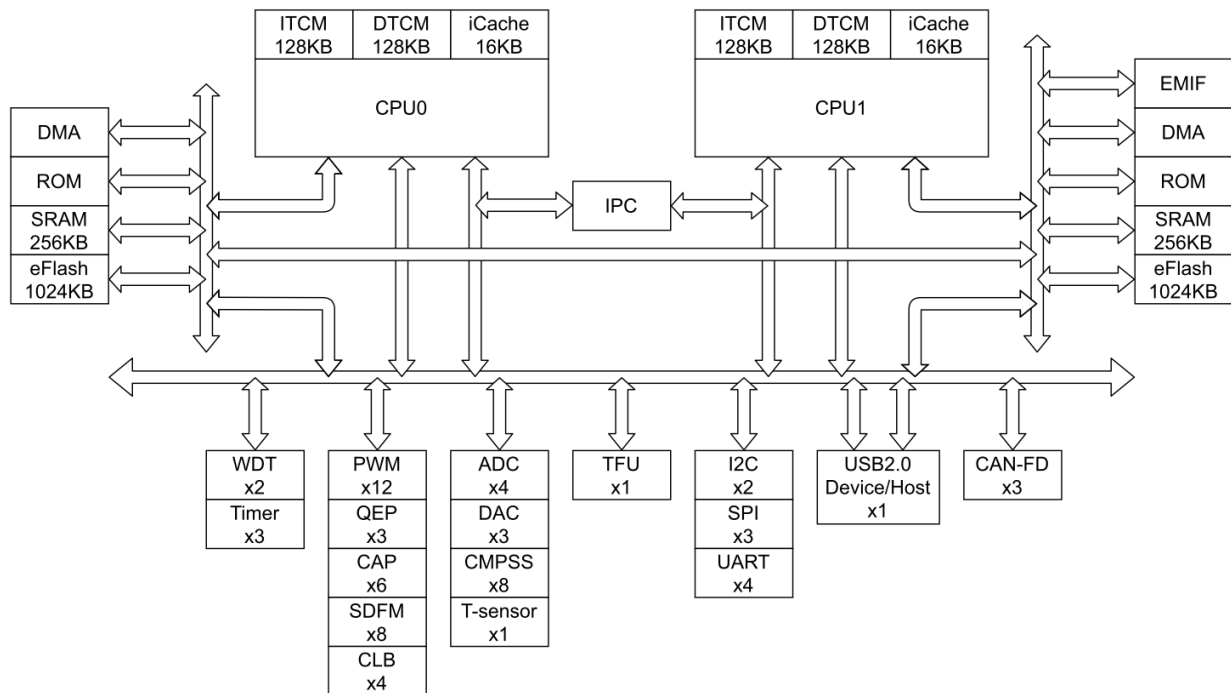


图 6.1 功能框图

6.3 内存

6.3.1 内存映射

表 6.1 内存映射表

Region	Name	Base Address	Top Address	Size	CPU1			CPU2			JTAG	USB DMAC
					AXIM	AHBP	DMAC	AXIM	AHBP	DMAC		
External Device	Reserved	0xD000_0000	0xDFFF_FFFF	256MB								
	Reserved	0xC000_0000	0xCFFF_FFFF	256MB								
	Reserved	0xB000_0000	0xBFFF_FFFF	256MB								
	SPIC Flash	0xA000_0000	0xAFFF_FFFF	256MB	✓		✓	✓		✓		
External Memory	SPIB Flash	0x9000_0000	0x9FFF_FFFF	256MB	✓		✓	✓		✓		
	SPIA Flash	0x8000_0000	0x8FFF_FFFF	256MB	✓		✓	✓		✓		
	Reserved	0x70C0_0000	0x7FFF_FFFF	244MB								
	EMIF1	0x6000_0000	0x70BF_FFFF	268MB	✓		✓	✓		✓		
Peripheral	Reserved	0x4260_0000	0x5FFF_FFFF	474MB								
	Reserved	0x4250_2000	0x425F_FFFF	1016KB								
	I2cbRegs	0x4250_1000	0x4250_1FFF	4KB		✓	✓		✓	✓	✓	
	I2caRegs	0x4250_0000	0x4250_0FFF	4KB		✓	✓		✓	✓	✓	
	Reserved	0x4240_1000	0x424F_FFFF	1020KB								
	PMBus0Regs	0x4240_0000	0x4240_0FFF	4KB		✓	✓		✓	✓	✓	
	Reserved	0x4230_3000	0x423F_FFFF	1012KB								
	SpicRegs	0x4230_2000	0x4230_2FFF	4KB		✓	✓		✓	✓	✓	
	SpibRegs	0x4230_1000	0x4230_1FFF	4KB		✓	✓		✓	✓	✓	
	SpiaRegs	0x4230_0000	0x4230_0FFF	4KB		✓	✓		✓	✓	✓	
	Reserved	0x4220_4000	0x422F_FFFF	1008KB								
	ScidRegs	0x4220_3000	0x4220_3FFF	4KB		✓	✓		✓	✓	✓	
	ScicRegs	0x4220_2000	0x4220_2FFF	4KB		✓	✓		✓	✓	✓	
	ScibRegs	0x4220_1000	0x4220_1FFF	4KB		✓	✓		✓	✓	✓	
	SciaRegs	0x4220_0000	0x4220_0FFF	4KB		✓	✓		✓	✓	✓	
	Reserved	0x4211_CC00	0x421F_FFFF	909KB								
	CANC SRAM	0x4211_8800	0x4211_CBFF	17KB		✓	✓		✓	✓	✓	
	CANB SRAM	0x4211_4400	0x4211_87FF	17KB		✓	✓		✓	✓	✓	
	CANA SRAM	0x4211_0000	0x4211_43FF	17KB		✓	✓		✓	✓	✓	
	Reserved	0x4210_3000	0x4210_FFFF	52KB								
	CancRegs	0x4210_2000	0x4210_2FFF	4KB		✓	✓		✓	✓	✓	
	CanbRegs	0x4210_1000	0x4210_1FFF	4KB		✓	✓		✓	✓	✓	
	Canaregs	0x4210_0000	0x4210_0FFF	4KB		✓	✓		✓	✓	✓	
	Reserved	0x4200_1000	0x420F_FFFF	1020KB								
	USBRegs	0x4200_0000	0x4200_0FFF	4KB		✓	✓		✓	✓	✓	
	Reserved	0x4180_1000	0x41FF_FFFF	8188KB								
	CRCRRegs	0x4180_0000	0x4180_0FFF	4KB		✓	✓		✓	✓	✓	
	Reserved	0x4170_2000	0x417F_FFFF	1016KB								
	eFlash1_ctrlregs	0x4170_1000	0x4170_1FFF	4KB		✓	✓				✓	
	eFlash1_ctrlregs	0x4170_0000	0x4170_0FFF	4KB					✓	✓		
	eFlash0_ctrlregs				✓	✓		✓				
	Reserved	0x4160_2000	0x416F_FFFF	1016KB								
	Reserved	0x4160_1000	0x4160_1FFF	4KB								
	EMIF1_ctrlregs	0x4160_0000	0x4160_0FFF	4KB		✓	✓		✓	✓	✓	
	Reserved	0x4150_2000	0x415F_FFFF	1016KB								
	Cmpss8Regs	0x4150_1C00	0x4150_1FFF	1KB		✓	✓		✓	✓	✓	
	Cmpss7Regs	0x4150_1800	0x4150_1BFF	1KB		✓	✓		✓	✓	✓	
	Cmpss6Regs	0x4150_1400	0x4150_17FF	1KB		✓	✓		✓	✓	✓	
	Cmpss5Regs	0x4150_1000	0x4150_13FF	1KB		✓	✓		✓	✓	✓	
	Cmpss4Regs	0x4150_0C00	0x4150_0FFF	1KB		✓	✓		✓	✓	✓	
	Cmpss3Regs	0x4150_0800	0x4150_0BFF	1KB		✓	✓		✓	✓	✓	
	Cmpss2Regs	0x4150_0400	0x4150_07FF	1KB		✓	✓		✓	✓	✓	
	Cmpss1Regs	0x4150_0000	0x4150_03FF	1KB		✓	✓		✓	✓	✓	
	Reserved	0x4140_0C00	0x414F_FFFF	1021KB								
	DaccRegs	0x4140_0800	0x4140_0BFF	1KB		✓	✓		✓	✓	✓	
	DacbRegs	0x4140_0400	0x4140_07FF	1KB		✓	✓		✓	✓	✓	
	DacaRegs	0x4140_0000	0x4140_03FF	1KB		✓	✓		✓	✓	✓	
	Reserved	0x4130_2000	0x413F_FFFF	1016KB								
	CPU2GpioDataRegs	0x4130_1000	0x4130_1FFF	4KB					✓			
	CPU1GpioDataRegs				✓				✓			
GpioCtrlRegs	0x4130_0000	0x4130_0FFF	4KB		✓					✓		
Reserved	0x4120_1C00	0x412F_FFFF	1017KB									
ClbOutputXbarRegs	0x4120_1800	0x4120_1BFF	1KB		✓					✓		
OutputXbarRegs	0x4120_1400	0x4120_17FF	1KB		✓					✓		
ClbXbarRegs	0x4120_1000	0x4120_13FF	1KB		✓					✓		
EPwmXbarRegs	0x4120_0C00	0x4120_0FFF	1KB		✓					✓		
ClbInputXbarRegs	0x4120_0800	0x4120_0BFF	1KB		✓					✓		
XbarRegs	0x4120_0400	0x4120_07FF	1KB		✓					✓		
InputXbarRegs	0x4120_0000	0x4120_03FF	1KB		✓					✓		
Reserved	0x4110_1000	0x411F_FFFF	1020KB									
DmaRegs	0x4110_0000	0x4110_0FFF	4KB		✓			✓		✓		
Reserved	0x4101_2000	0x410F_FFFF	952KB									
CPU2 DW_WDT_Regs	0x4101_1000	0x4101_1FFF	4KB					✓				
CPU1 DW_WDT_Regs				✓				✓				
CPU2 DW_Timer_Regs	0x4101_0000	0x4101_0FFF	4KB					✓				
CPU1 DW_Timer_Regs				✓				✓				

(续表见下页)

内存映射表 (续表)

Region	Name	Base Address	Top Address	Size	CPU1			CPU2			JTAG	USB DMAC
					AXIM	AHBP	DMAC	AXIM	AHBP	DMAC		
Peripheral	Reserved	0x4100_3C00	0x4100_FFFF	49KB								
	UID_REGS	0x4100_3800	0x4100_3BFF	1KB		✓			✓		✓	
	CPU2 CpuIdRegs	0x4100_3400	0x4100_37FF	1KB					✓			
	CPU1 CpuIdRegs					✓				✓		
	CPU2 MemoryErrorRegs	0x4100_3000	0x4100_33FF	1KB					✓			
	CPU1 MemoryErrorRegs					✓					✓	
	CPU2 AccessProtectionRegs	0x4100_2C00	0x4100_2FFF	1KB					✓			
	CPU1 AccessProtectionRegs					✓					✓	
	CPU2 MemCfgRegs	0x4100_2800	0x4100_2BFF	1KB					✓			
	CPU1 MemCfgRegs					✓						✓
	CPU2 CpuSysRegs	0x4100_2400	0x4100_27FF	1KB					✓			
	CPU1 CpuSysRegs					✓						✓
	ClkCfgRegs	0x4100_2000	0x4100_23FF	1KB		✓			✓		✓	
	AnalogSubsysRegs	0x4100_1C00	0x4100_1FFF	1KB		✓			✓		✓	
	DevCfgRegs	0x4100_1800	0x4100_1BFF	1KB		✓			✓		✓	
	CPU2 XintRegs	0x4100_1400	0x4100_17FF	1KB					✓			
	CPU1 XintRegs					✓						✓
	CPU2 NmiIntruptRegs	0x4100_1000	0x4100_13FF	1KB					✓			
	CPU1 NmiIntruptRegs					✓						✓
	CPU2 WdRegs	0x4100_0C00	0x4100_0FFF	1KB					✓			
	CPU1 WdRegs					✓						✓
	CPU2 CpuTimer2Regs	0x4100_0800	0x4100_0BFF	1KB					✓			
	CPU1 CpuTimer2Regs					✓						✓
	CPU2 CpuTimer1Regs	0x4100_0400	0x4100_07FF	1KB					✓			
	CPU1 CpuTimer1Regs					✓						✓
	CPU2 CpuTimer0Regs	0x4100_0000	0x4100_03FF	1KB					✓			
	CPU1 CpuTimer0Regs					✓						✓
	Reserved	0x4080_0800	0x40FF_FFFF	8190KB								
	SdFm2Regs	0x4080_0400	0x4080_07FF	1KB		✓	✓		✓	✓	✓	
	SdFm1Regs	0x4080_0000	0x4080_03FF	1KB		✓	✓		✓	✓	✓	
	Reserved	0x4070_0C00	0x407F_FFFF	1021KB								
	EQep3Regs	0x4070_0800	0x4070_0BFF	1KB		✓	✓		✓	✓	✓	
	EQep2Regs	0x4070_0400	0x4070_07FF	1KB		✓	✓		✓	✓	✓	
	EQep1Regs	0x4070_0000	0x4070_03FF	1KB		✓	✓		✓	✓	✓	
	Reserved	0x4060_1800	0x406F_FFFF	1018KB								
	ECap6Regs	0x4060_1400	0x4060_17FF	1KB		✓	✓		✓	✓	✓	
	ECap5Regs	0x4060_1000	0x4060_13FF	1KB		✓	✓		✓	✓	✓	
	ECap4Regs	0x4060_0C00	0x4060_0FFF	1KB		✓	✓		✓	✓	✓	
	ECap3Regs	0x4060_0800	0x4060_0BFF	1KB		✓	✓		✓	✓	✓	
	ECap2Regs	0x4060_0400	0x4060_07FF	1KB		✓	✓		✓	✓	✓	
	ECap1Regs	0x4060_0000	0x4060_03FF	1KB		✓	✓		✓	✓	✓	
	Reserved	0x4050_3400	0x405F_FFFF	1011KB								
	EPwm12Regs	0x4050_3000	0x4050_33FF	1KB		✓	✓		✓	✓	✓	
	EPwm11Regs	0x4050_2C00	0x4050_2FFF	1KB		✓	✓		✓	✓	✓	
	EPwm10Regs	0x4050_2800	0x4050_2BFF	1KB		✓	✓		✓	✓	✓	
	EPwm9Regs	0x4050_2400	0x4050_27FF	1KB		✓	✓		✓	✓	✓	
	EPwm8Regs	0x4050_2000	0x4050_23FF	1KB		✓	✓		✓	✓	✓	
	EPwm7Regs	0x4050_1C00	0x4050_1FFF	1KB		✓	✓		✓	✓	✓	
	EPwm6Regs	0x4050_1800	0x4050_1BFF	1KB		✓	✓		✓	✓	✓	
	EPwm5Regs	0x4050_1400	0x4050_17FF	1KB		✓	✓		✓	✓	✓	
	EPwm4Regs	0x4050_1000	0x4050_13FF	1KB		✓	✓		✓	✓	✓	
	EPwm3Regs	0x4050_0C00	0x4050_0FFF	1KB		✓	✓		✓	✓	✓	
	EPwm2Regs	0x4050_0800	0x4050_0BFF	1KB		✓	✓		✓	✓	✓	
	EPwm1Regs	0x4050_0400	0x4050_07FF	1KB		✓	✓		✓	✓	✓	
	SyncSocRegs	0x4050_0000	0x4050_03FF	1KB		✓	✓		×	×	✓	
	Reserved	0x4040_6000	0x404F_FFFF	1000KB								
	Clb8DataExchRegs	0x4040_5C00	0x4040_5FFF	1KB		✓	✓		✓	✓	✓	
	Clb8LogicCtrlRegs	0x4040_5800	0x4040_5BFF	1KB		✓	✓		✓	✓	✓	
	Clb8LogicCfgRegs	0x4040_5400	0x4040_57FF	1KB		✓	✓		✓	✓	✓	
	Clb7DataExchRegs	0x4040_5000	0x4040_53FF	1KB		✓	✓		✓	✓	✓	
	Clb7LogicCtrlRegs	0x4040_4C00	0x4040_4FFF	1KB		✓	✓		✓	✓	✓	
	Clb7LogicCfgRegs	0x4040_4800	0x4040_4BFF	1KB		✓	✓		✓	✓	✓	
	Clb6DataExchRegs	0x4040_4400	0x4040_47FF	1KB		✓	✓		✓	✓	✓	
	Clb6LogicCtrlRegs	0x4040_4000	0x4040_43FF	1KB		✓	✓		✓	✓	✓	
	Clb6LogicCfgRegs	0x4040_3C00	0x4040_3FFF	1KB		✓	✓		✓	✓	✓	
	Clb5DataExchRegs	0x4040_3800	0x4040_3BFF	1KB		✓	✓		✓	✓	✓	
	Clb5LogicCtrlRegs	0x4040_3400	0x4040_37FF	1KB		✓	✓		✓	✓	✓	
	Clb5LogicCfgRegs	0x4040_3000	0x4040_33FF	1KB		✓	✓		✓	✓	✓	
	Clb4DataExchRegs	0x4040_2C00	0x4040_2FFF	1KB		✓	✓		✓	✓	✓	
	Clb4LogicCtrlRegs	0x4040_2800	0x4040_2BFF	1KB		✓	✓		✓	✓	✓	
Clb4LogicCfgRegs	0x4040_2400	0x4040_27FF	1KB		✓	✓		✓	✓	✓		
Clb3DataExchRegs	0x4040_2000	0x4040_23FF	1KB		✓	✓		✓	✓	✓		

(续表见下页)

内存映射表 (续表)

Region	Name	Base Address	Top Address	Size	CPU1			CPU2			JTAG	USB DMAC	
					AXIM	AHBP	DMAC	AXIM	AHBP	DMAC			
Peripheral	C1b3LogicCtrlRegs	0x4040_1C00	0x4040_1FFF	1KB		✓	✓		✓	✓	✓		
	C1b3LogicCfgRegs	0x4040_1800	0x4040_1BFF	1KB		✓	✓		✓	✓	✓		
	C1b2DataExchRegs	0x4040_1400	0x4040_17FF	1KB		✓	✓		✓	✓	✓		
	C1b2LogicCtrlRegs	0x4040_1000	0x4040_13FF	1KB		✓	✓		✓	✓	✓		
	C1b2LogicCfgRegs	0x4040_0C00	0x4040_0FFF	1KB		✓	✓		✓	✓	✓		
	C1b1DataExchRegs	0x4040_0800	0x4040_0BFF	1KB		✓	✓		✓	✓	✓		
	C1b1LogicCtrlRegs	0x4040_0400	0x4040_07FF	1KB		✓	✓		✓	✓	✓		
	C1b1LogicCfgRegs	0x4040_0000	0x4040_03FF	1KB		✓	✓		✓	✓	✓		
	Reserved	0x4030_1000	0x403F_FFFF	1020KB									
	AdcdResultRegs	0x4030_0C00	0x4030_0FFF	1KB		✓	✓		✓	✓	✓		
	AdccResultRegs	0x4030_0800	0x4030_0BFF	1KB		✓	✓		✓	✓	✓		
	AdcbResultRegs	0x4030_0400	0x4030_07FF	1KB		✓	✓		✓	✓	✓		
	AdcaResultRegs	0x4030_0000	0x4030_03FF	1KB		✓	✓		✓	✓	✓		
	Reserved	0x4020_1000	0x402F_FFFF	1020KB									
	AdcdRegs	0x4020_0C00	0x4020_0FFF	1KB		✓	✓		✓	✓	✓		
	AdccRegs	0x4020_0800	0x4020_0BFF	1KB		✓	✓		✓	✓	✓		
	AdcbRegs	0x4020_0400	0x4020_07FF	1KB		✓	✓		✓	✓	✓		
	AdcaRegs	0x4020_0000	0x4020_03FF	1KB		✓	✓		✓	✓	✓		
	Reserved	0x4010_1000	0x401F_FFFF	1020KB									
	TFURegs	0x4010_0000	0x4010_0FFF	4KB		✓	✓		✓	✓	✓		
	Reserved	0x4000_3000	0x400F_FFFF	1012KB									
	CPU2toCPU1 MS-GRAM	0x4000_2000	0x4000_2FFF	4KB		✓			✓	✓	✓		
	CPU1toCPU2 MS-GRAM	0x4000_1000	0x4000_1FFF	4KB		✓	✓		✓		✓		
	IpcRegs	0x4000_0000	0x4000_0FFF	4KB		✓	✓		✓	✓	✓		
	SRAM	Reserved	0x2010_0000	0x3FFF_FFFF	511MB								
		CPU1 DTCM (re-mapping)	0x2100_0000	0x2101_FFFF	128KB	✓		✓				✓	
		Reserved	0x2002_0000	0x200F_FFFF	896KB								
		CPU2 DTCM	0x2000_0000	0x2001_FFFF	128KB				✓		✓	✓	
Code	Reserved	0x1040_0000	0x1FFF_FFFF	252MB									
	eFlash1	0x1030_0000	0x103F_FFFF	1024KB	✓		✓	✓		✓		×	
	Reserved	0x1022_0000	0x102F_FFFF	896KB									
	Global Share SRAM15 (close to CPU1)	0x1021_C000	0x1021_FFFF	16KB	✓		✓	✓		✓		✓	
	Global Share SRAM14 (close to CPU1)	0x1021_8000	0x1021_BFFF	16KB	✓		✓	✓		✓		✓	
	Global Share SRAM13 (close to CPU1)	0x1021_4000	0x1021_7FFF	16KB	✓		✓	✓		✓		✓	
	Global Share SRAM12 (close to CPU1)	0x1021_0000	0x1021_3FFF	16KB	✓		✓	✓		✓		✓	
	Global Share SRAM11 (close to CPU1)	0x1020_C000	0x1020_FFFF	16KB	✓		✓	✓		✓		✓	
	Global Share SRAM10 (close to CPU1)	0x1020_8000	0x1020_BFFF	16KB	✓		✓	✓		✓		✓	
	Global Share SRAM9 (close to CPU1)	0x1020_4000	0x1020_7FFF	16KB	✓		✓	✓		✓		✓	
	Global Share SRAM8 (close to CPU1)	0x1020_0000	0x1020_3FFF	16KB	✓		✓	✓		✓		✓	
	Reserved	0x1010_4000	0x101F_FFFF	1008KB									
	CPU1 ROM	0x1010_0000	0x1010_3FFF	16KB	×		×	✓		✓		×	
	Reserved	0x1000_0000	0x100F_FFFF	1024KB									
	CPU1 ITCM (re-mapping)	0x0100_0000	0x0101_FFFF	128KB	✓		✓				✓	×	
	Reserved	0x0040_0000	0x0FFF_FFFF	252MB									
	eFlash0	0x0030_0000	0x003F_FFFF	1024KB	✓		✓	×		×	✓	×	
	Reserved	0x0022_0000	0x002F_FFFF	896KB									
	Global Share SRAM7 (close to CPU1)	0x0021_C000	0x0021_FFFF	16KB	✓		✓	✓		✓		✓	
	Global Share SRAM6 (close to CPU1)	0x0021_8000	0x0021_BFFF	16KB	✓		✓	✓		✓		✓	
	Global Share SRAM5 (close to CPU1)	0x0021_4000	0x0021_7FFF	16KB	✓		✓	✓		✓		✓	
	Global Share SRAM4 (close to CPU1)	0x0021_0000	0x0021_3FFF	16KB	✓		✓	✓		✓		✓	
	Global Share SRAM3 (close to CPU1)	0x0020_C000	0x0020_FFFF	16KB	✓		✓	✓		✓		✓	
	Global Share SRAM2 (close to CPU1)	0x0020_8000	0x0020_BFFF	16KB	✓		✓	✓		✓		✓	

(续表见下页)

内存映射表 (续表)

Region	Name	Base Address	Top Address	Size	CPU1			CPU2			JTAG	USB DMAC
					AXIM	AHBP	DMAC	AXIM	AHBP	DMAC		
Code	Global Share SRAM1 (close to CPU1)	0x0020_4000	0x0020_7FFF	16KB	✓		✓	✓		✓	✓	
	Global Share SRAM0 (close to CPU1)	0x0020_0000	0x0020_3FFF	16KB	✓		✓	✓		✓	✓	
	Reserved	0x0010_8000	0x001F_FFFF	992KB								
	CPU1 ROM	0x0010_0000	0x0010_7FFF	32KB	✓		✓	×		✓	×	
	Reserved	0x0002_0000	0x000F_FFFF	896KB						✓	✓	
	CPU2 ITCM CPU1 ITCM	0x0000_0000	0x0001_FFFF	128KB			✓			✓	✓	

6.3.2 Memory类型

6.3.2.1 TCM(Tightly Coupled Memory)

每个CPU都有128KB带ECC保护的ITCM和128KB带Parity校验的DTCM。

6.3.2.2 Global Share SRAM

CPU和DMA都可以访问的SRAM。

通过GSxMSEL寄存器配置，可以将各个Global Share SRAM分配给任一CPU子系统。

所有GSx RAM都有ECC保护。

当一个GSx RAM归CPU子系统所有时，CPU_x和CPU_x.DMA将完全访问该RAM块，而另一个CPU_y和CPU_y.DMA将仅具有读访问权限（没有写访问权限）。

6.4 芯片ID

芯片ID如图6.2所示。

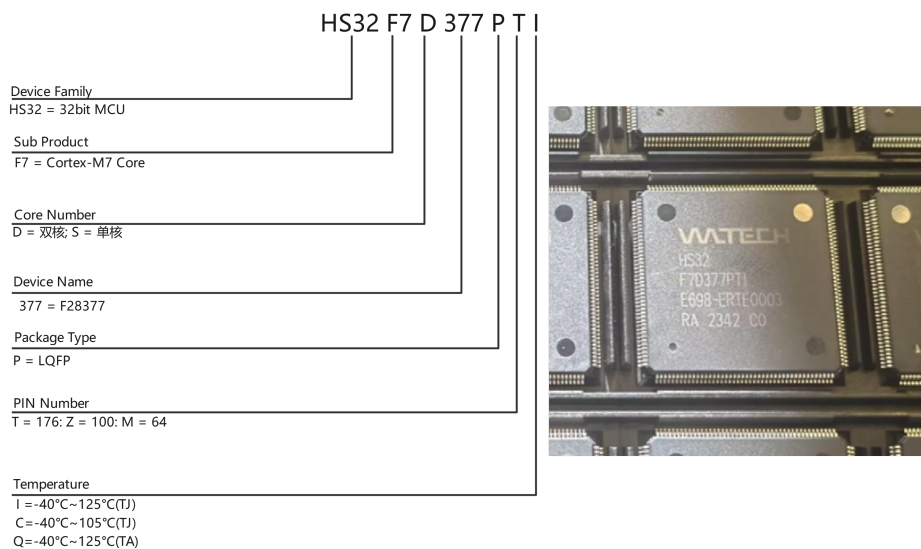


图 6.2 命名规则

6.5 总线结构

本节介绍了图6.3中所示的全芯片的bus matrix结构。

其中，cpu1_axi_matrix和cpu2_axi_matrix为64bits的axi总线，hsp_ahb_matrix和lsp_ahb_matrix为32bits的ahb bus。因为dma为32bits，接入到64bits axi bus 需要经过x2x（32bitsto64bits）转接；bootrom为32bits ahb接口，因此通过axi matrix去访问bootrom需要经过x2h（64bitto32bits）转接。通过cpu1/2 axi matrix向下访问 adc以及hsp/lsp需要做先x2h（64bitsto32bits）转接。

为了提高cpu1/2 以及dma1/2 对adc的访问效率，adc直接通过ahb_demux挂载。此外高速ahb slave直接挂载在hsp_ahb_matrix下，而低速ahb设备直接挂载在lsp_ahb_matrix下，各低速apb slave通过h2p（ahb2apb）转接也独立挂载在lsp_ahb_matrix下。另外usb ahb master需要作为主机对memory进行访问，而memory挂载在axi matrix下，因此需要通过一个ahb2axi的转接接入到cpu1_axi_matrix。此外，dap模块用于调试cpu1、cpu2以及whole system，其中对于whole system的调试通过h2x（64bitsto64bits）接入到cpu1_axi_matrix。

系统各master对各slave的地址映射和访问权限参考表6.1。

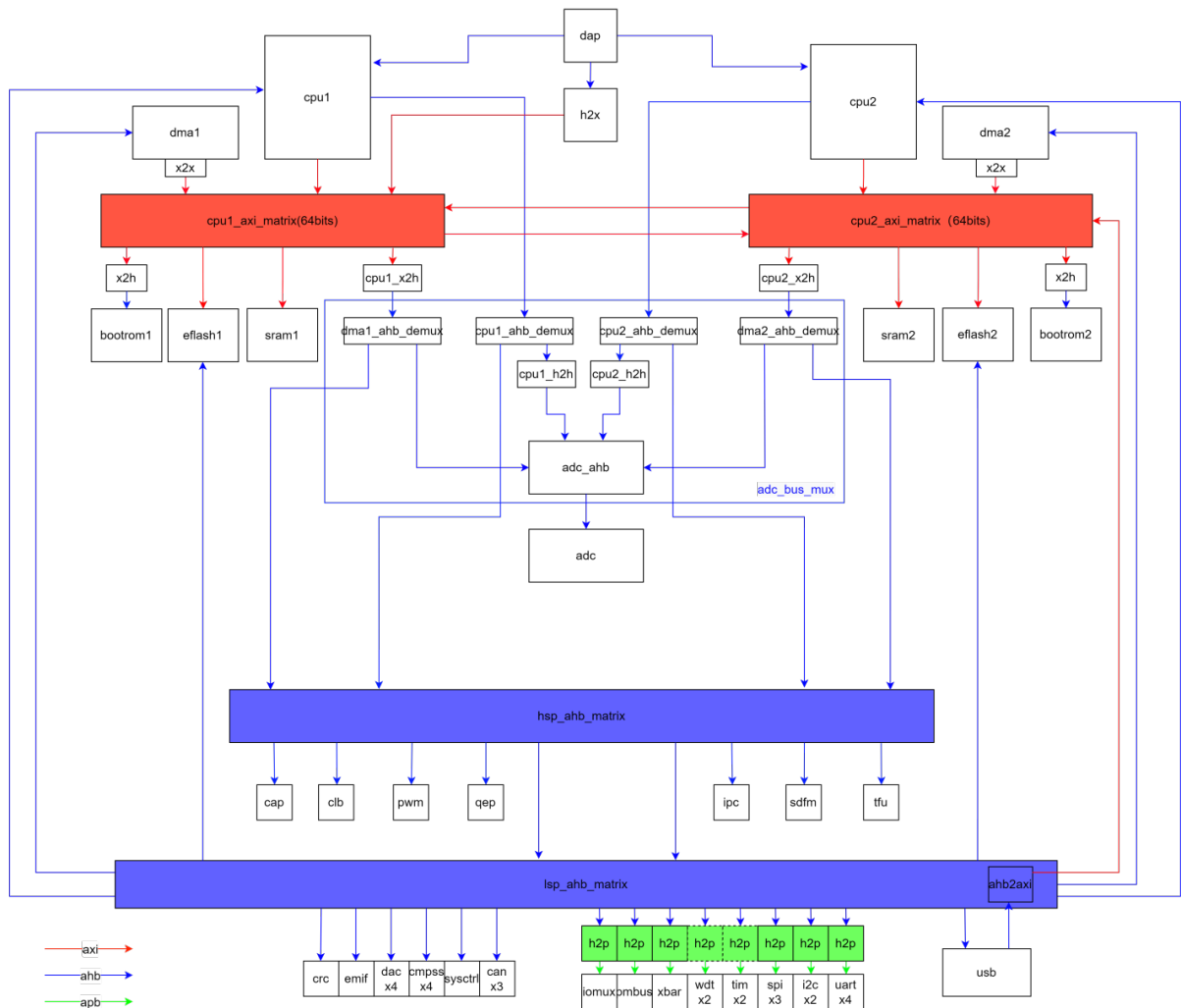


图 6.3 全芯片的总线结构

6.6 ARM M7

Arm® Cortex®-M7是一款高效、高性能嵌入式处理器，具有低中断延迟，低成本调试，并与现有 Cortex-M配置文件处理器向后兼容。M7有一个有序的超标量管道，只吃多指令，可以双重并发，包括加载/加载、加载/存储指令对，因为有多多个内存接口。

处理器支持的存储接口包括：

- 紧耦合内存（Tightly-Coupled Memory, Memory（TCM）接口
- 哈佛指令和数据缓存
- AXI主设备（AXIM）接口
- 专用的低延迟AHB-Lite外设（AHBP）接口
- AHB-Lite从（AHBS）接口提供DMA访问TCMs

6.6.1 特性

Cortex-M7具有以下特性：

- 支持顺序，具有动态分支预测的超标量流水
- DSP扩展
- ARMv7-M Thumb指令集
- 堆栈指针（SP）
- 硬件整数除法指令，SDIV和UDIV
- 处理器模式（handler mode）和线程模式（thread mode）
- Thumb和调试状态
- 自动处理器状态保存和恢复，低延迟中断服务程序（ISR）的进入和退出
- 支持ARMv7-M大端字节不变访问或小端访问
- 支持ARMv7-M不对齐访问
- 实现低延迟中断处理，通过：
 - 与处理器紧密集成的嵌套矢量中断控制器（NVIC）
 - 支持异常持续指令，如LDM, LDMDb, STM, STMDb, PUSH, POP和VLDM, VSTM, VPUSH, VPOP
- 低成本的调试解决方案，具有功能：
 - 实现断点
 - 实现观察点、跟踪和系统分析
 - 通过指令跟踪单元（ITM）支持printf()风格的调试
 - 跟踪端口接口单元（TPIU）
 - 调试访问端口（DAP）
- 支持嵌入式跟踪单元ETM
- 存储系统，包括MPU和哈佛数据和指令缓存ECC
- 浮点单元（FPU）
- 低功耗特性包括时钟门控、睡眠模式和唤醒中断控制器（WIC）

6.6.2 接口

Cortex-M7如图6.4所示。

AHBP接口提供了适合低延迟系统的访问外围设备。它支持对未对齐内存的访问，对写入缓冲区进行缓冲多处理器系统的数据写入和独占访问传输。

AHBS接口提供系统访问TCMs。

AHBD接口用来调试访问M7处理器和完整的内存映射。

EPPB接口用来访问CoreSight兼容的调试和跟踪组件，在系统中连接到处理器。

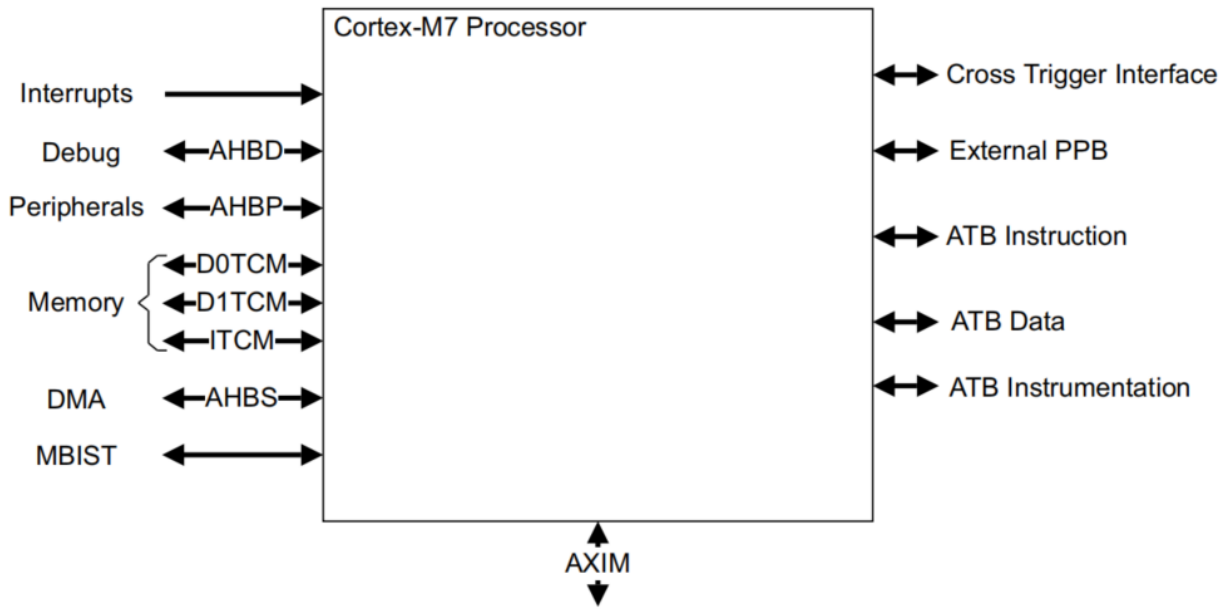


图 6.4 Cortex-M7处理器接口

ATB接口输出用于调试的跟踪信息。ATB接口兼容CoreSight架构。

TCM接口包括指令TCM和数据TCM接口，每个都有双字数据宽度。通过ITCM 64位宽接口访问ITCM。通过D1TCM 32位宽接口和D0TCM 32位宽接口访问DTCM。DTCM的访问被分割，使得低32bit总是访问D0TCM，高32bit总是访问D1TCM。

Cross Trigger接口，处理器包括一个可选的交叉触发接口单元，其中包括一个适合使用交叉触发矩阵（Matrix）连接到外部CoreSight组件的接口。

MBIST接口用于生产测试时对RAMs进行测试。Cortex-M7处理器还支持在线MBIST，允许RAM在正常执行期间使用MBIST接口进行测试。

AXIM接口提供对外部存储器系统的高性能访问。AXIM接口支持使用ARM CoreLink L2C-310 2级缓存控制器，不支持 L2C-310独占缓存配置。

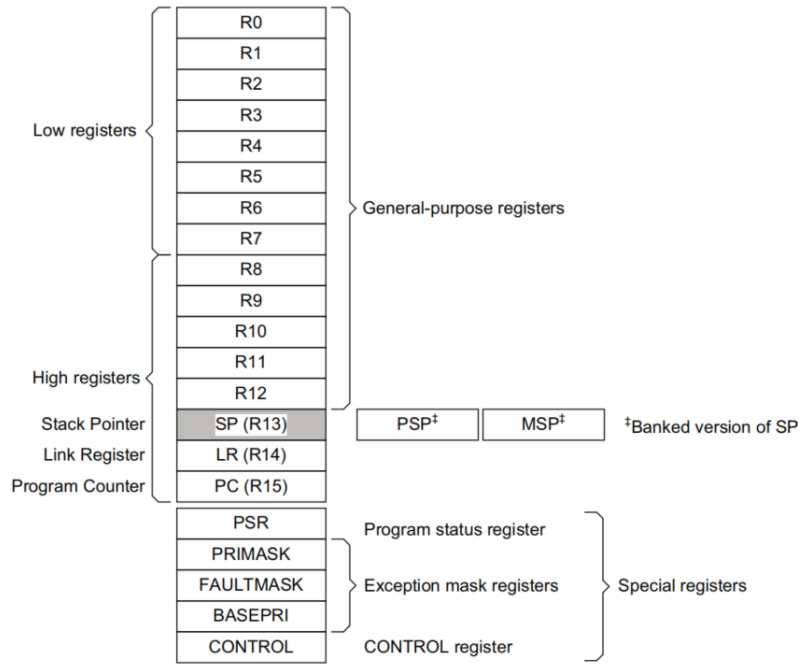


图 6.6 处理器内核寄存器

6.6.3.2 数据处理单元(DPU)

- 具有六个读端口和四个写端口的并行整数寄存器文件，用于大规模双发射；
- 广泛的转发逻辑，以尽量减少联锁（interlocks）；
- 两个ALU，其中一个ALU能够执行SIMD操作；
- 单个MAC流水能力 32×32 位 + 64位 → 64位，具有两个周期的结果延迟和一个MAC每周期吞吐量；
- 支持依赖于操作数提前终止的单除法单元。

6.6.3.3 预取单元(PFU)

- 64位指令获取带宽；
- 4x64位预取队列，从DPU流水操作中解耦指令预取；
- 分支目标地址缓存（BTAC）用于分支预测器状态和目标地址的单周期转换；
- 未指定BTAC时的静态分支预测器；
- 在解码器和处理器管道的第一个执行阶段转发直接分支的早期解析标志。

6.6.3.4 负载存储单元(LSU)

- 双32位负载通道通往TCM、数据缓存和AXIM接口，提供64位负载带宽和双32位负载能力
- 到AHB接口的单个32位加载通道
- 单64位存储通道
- 存储缓冲以提高存储吞吐量，并最大限度地减少数据和指令读取的RAM争用
- 为TCM、AHBP和AXIM提供单独的存储缓冲，以实现服务质量(QoS)和特定接口的优化

6.6.3.5 浮点单元(FPU)

- 延迟浮点上下文保存。浮点状态的自动堆叠被延迟到ISR尝试执行浮点指令。这减少了进入ISR的延迟，并删除浮点上下文（不使用浮点的ISR除外）。

- 单精度（C语言浮点型）数据处理操作说明
- 可选双精度（C双类型）数据处理操作指令
- 结合乘法和累加指令，提高精度（融合MAC）
- 硬件支持转换，加法，减法，乘法可选累加，除法和平方根
- 硬件支持标准和所有IEEE标准754-2008舍入模式
- 32个32位单精度寄存器或16个64位双精度寄存器

6.7 TFU

超越函数单元（Transcendental Functions Unit, TFU）提供数学函数的硬件加速（主要是三角函数），在电机控制、计量、信号处理等场景中均有应用。与软件实现相比，TFU加快了特定函数的计算速度，从而可以使用较低的工作频率，或者释放处理器周期以执行其他任务。

TFU的结构如图6.7所示：

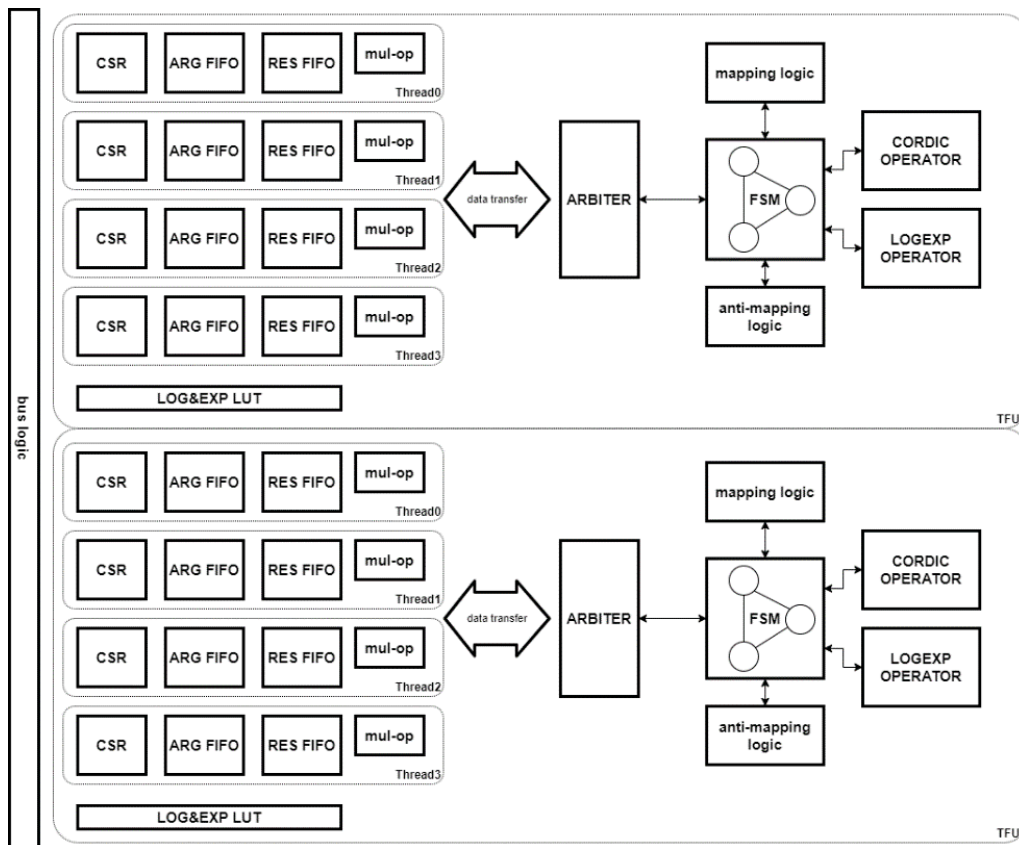


图 6.7 TFU的功能框图

TFU的功能点具体包括：

- 24位CORDIC旋转模式/向量加速器
- 支持函数: sine, cosine, atan, atan2, modulus, logarithm, exponent
- 计算精度可配置
- 低延时AHB接口
- 结果可以在准备好后立即读取，而不需要轮询或中断
- 多线程读写，最高支持8线程（2个模块）的并行计算

TFU以定点带符号整数格式操作，输入和输出值格式可以配置为q1.31或q1.15。

- 在q1.31格式中，数字由1个符号位和31个小数位(二进制小数位)表示。因此，数字范围是-1(0x80000000) 到 $1-2^{-31}$ (0x7FFFFFFF)。
- 在q1.15格式中，数字范围是1(0x8000)到 $1-2^{-15}$ (0x7FFF)。这种格式要求寄存器四字节低位对齐。

TFU每个时钟周期可以执行2次迭代。对于每个函数，每4次迭代后剩余的最大误差如表6.3所示，表中还标明了达到该精度所需的运算时钟周期数。可以在TFU_CSR寄存器的tfu_precision字段中确定和编程所需的周期数。

一旦配置的迭代次数完成，TFU就会停止，此时可以立即读取对应线程的计算结果。

表 6.3 TFU计算功能及具体参数

计算功能	迭代次数	计算周期数	最大残余误差	
			q1.31	q1.15
Sin, Cos, Arctan, Arctan2, Mod, Log, Exp	4	2	2^{-3}	2^{-3}
	8	4	2^{-7}	2^{-7}
	12	6	2^{-11}	2^{-11}
	16	8	2^{-15}	2^{-15}
	20	10	2^{-15}	2^{-18}
	24	12	2^{-15}	2^{-19}

6.8 CRC

CRC(循环冗余校验)计算单元用于从7位、8位、16位、24位或32位数据字和生成多项式中获得CRC码。在其他应用中，基于CRC的技术用于验证数据传输或存储的完整性。在功能安全标准的范围内，它们提供了一种验证Flash存储完整性的方法。

CRC计算单元的主要功能点如下：

- 默认状态下使用CRC-32（以太网）多项式：0x4C11DB7
- 或者，使用可配置大小的完全可编程多项式（7,8,24,16,32位）
- 处理8位、16位、32位数据
- 可编程CRC初始值
- 2级FIFO单输入32位数据寄存器，以避免在计算过程中总线卡顿
- 单输出32位数据寄存器
- 不同数据的CRC计算时间
 - 32位数据，4个AHB时钟周期（HCLK）中完成
 - 16位数据，2个AHB时钟周期（HCLK）中完成
 - 8位数据，1个AHB时钟周期（HCLK）中完成
- 输入/输出数据的大小端配置
- 寄存器通过AHB外设仅支持4字节对齐访问，除了可以通过字、右对齐半字和右对齐字节访问的CRC_DR寄存器。

图6.8是CRC计算单元的结构框图。

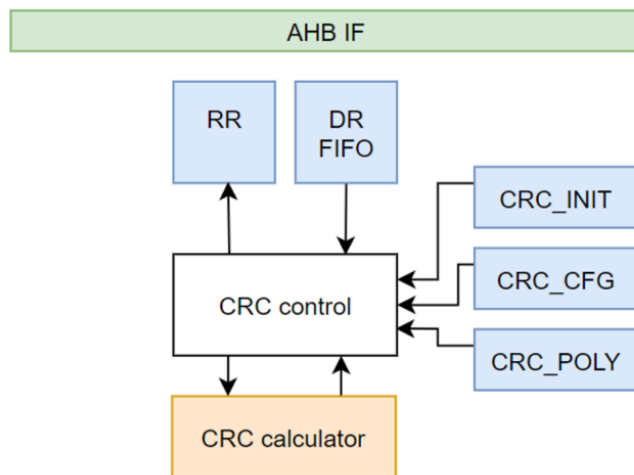


图 6.8 CRC计算单元的结构框图

6.9 DMA

每个CPU都有自己的六通道DMA模块。DMA模块提供了一种无需CPU的干预，在外设和存储器、存储器和存储器之间传输数据的硬件方法，可为其它系统功能释放带宽。DMA框图如图6.9所示。

DMA基于事件触发，模块本身没有触发机制来周期性的启动数据传输，需要一个外设或软件触发器来启动DMA传输，六个通道分别拥有独立的触发源选择逻辑，可选择当前通道的触发源。可以分别配置六个DMA通道中的每个通道的中断触发源，并且每个通道包含自己独立的NVIC中断，以便让CPU知道DMA传输何时开始或完成。六个通道中有五个完全相同，而通道1可以配置为比其他通道更高的优先级。

DMA特性包括：

- 6条通道，每条通道有独立的中断；
- 触发源
 - ADC
 - XBAR
 - CPU1 timer
 - CAP
 - SDFM
 - SPI
 - CLB
 - CAN
 - TFU
 - CPU1 dwtim
- 数据来源和目的
 - peripheral
 - eFlash
 - RAM
 - EMIF
- 支持数据位宽8/16/32
- 支持源/目的地址和数据位宽非对齐

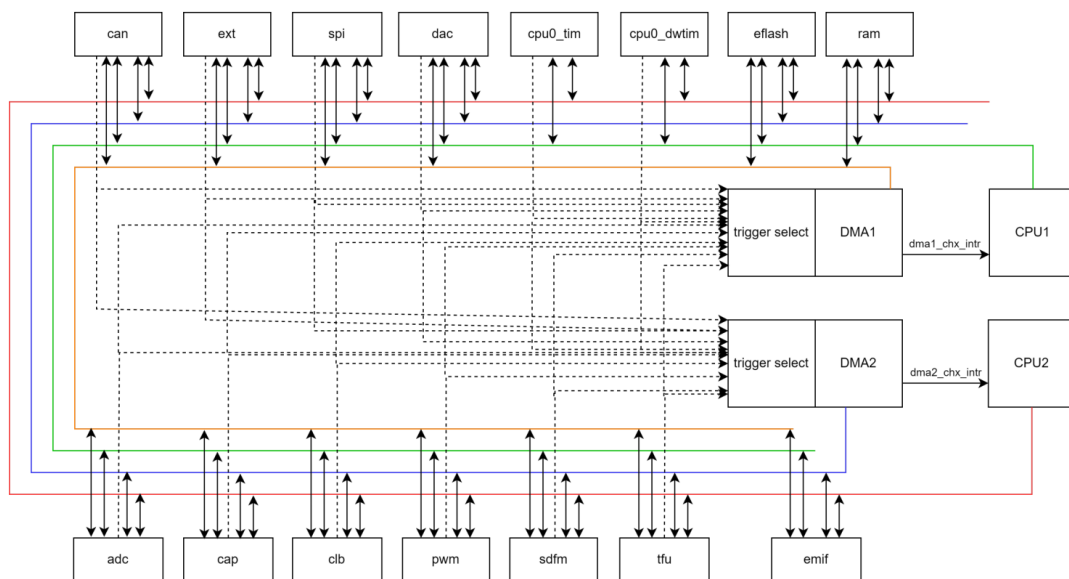


图 6.9 DMA框图

6.10 IPC

核间通信（Inter processor communication, IPC）模块支持多种处理器间通信方法：

- 每个CPU有32个IPC标志，可用于通过软件轮询发出事件信号或指示状态。每个CPU有四个标志可以生成中断。
- 共享数据寄存器，可用于在CPU之间发送命令或其他小段信息。尽管选择寄存器名称是为了支持命令/响应系统，但它们可以用于软件定义的任何用途。
- 引导模式和状态寄存器，允许CPU1控制CPU2的引导过程。
- 一个通用的自由运行的64位计数器。
- 两个共享的消息RAM，可用于传输批量数据。每个RAM都可以由两个CPU读取。CPU1可以写入一个RAM，CPU2可以写入另一个RAM。IPC架构如图6.10所示。

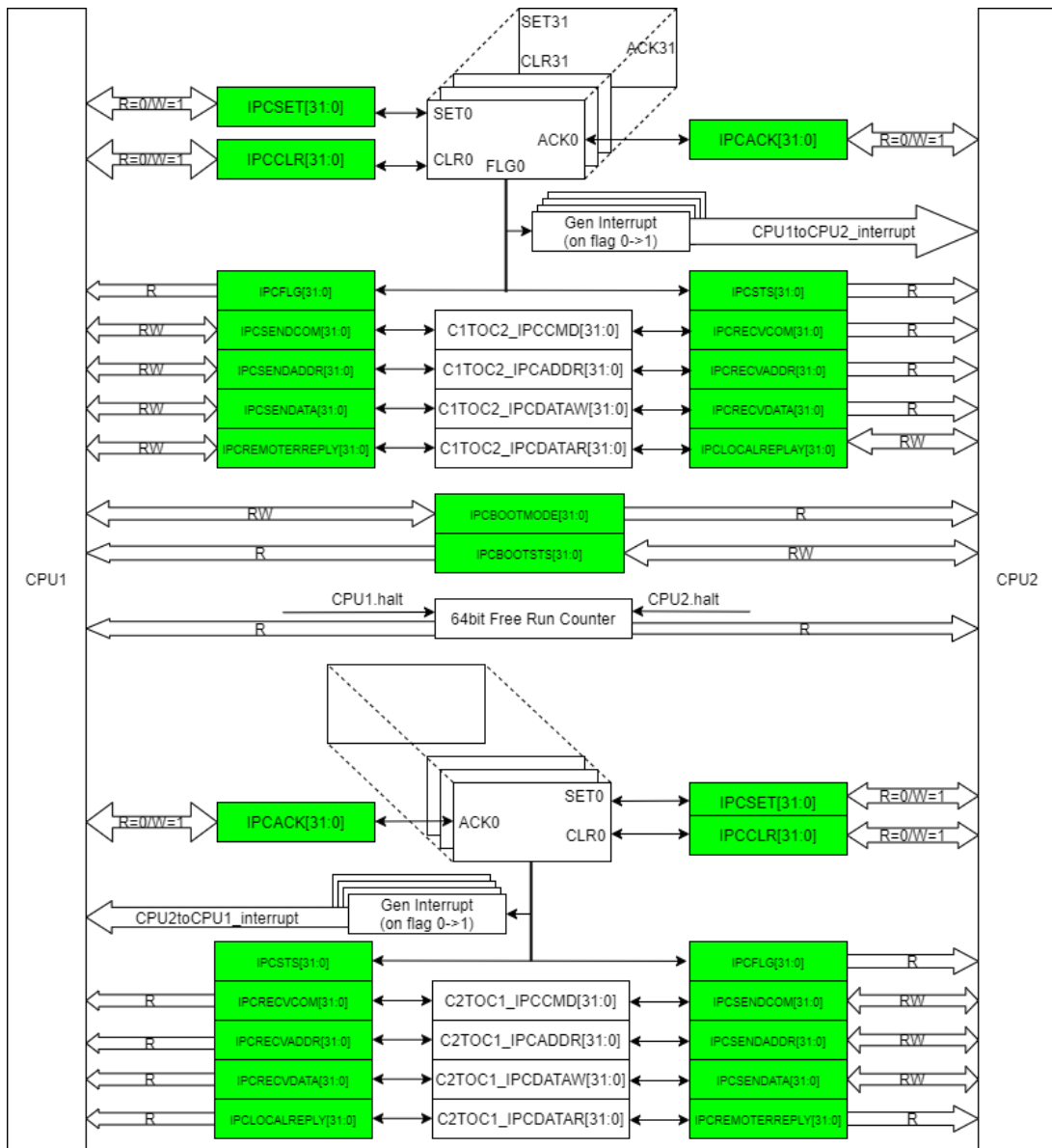


图 6.10 IPC架构

6.11 Bootrom与外设booting

芯片的每个内核的BOOTROM都固化了芯片启动软件代码，其中CPU1的BOOTROM负责整个系统的初始化工作，时钟校准，并将CPU2从复位中释放。芯片每次上电启动或者复位都会执行BOOTROM代码。

CPU1 BOOTROM是芯片的主控方，除了负责芯片的配置，还负责CPU1和CPU2的APP程序下载功能。CPU1可以配置从 EFLASH中直接启动，也选择通过BOOT外设接口，来下载程序。用户可以在CPU1的EFLASH NVR中配置BOOTMODE PIN（可配置任意GPIO口），通过BOOTPIN的状态值来决定由何种方式BOOT。

CPU2 BOOTROM启动后，会进入wait boot模式，等待CPU1 IPC发送BOOT MODE，CPU2会根据IPC boot mode，可以选择从EFLASH, ITCM或者SRAM中启动。

6.11.1 CPU1 BOOT PIN配置

CPU1的EFLASH NVR中烧录了BOOTMODE.bin文件，CPU1根据BOOTMODE.bin来选择BOOT启动模式，BOOTMODE.bin中配置了BOOT GPIO的管脚配置，以及BOOT外设管脚配置。

CPU1的BOOT PIN GPIO信息包含了两个BOOT GPIO接口（可配置任意GPIO口），用户可以对BOOT GPIO进行上拉或者下拉，来配置选择4种BOOT模式，其中3种最常用的BOOT模式，分别是EFLASH加载，UART0下载和SPI0 下载模式。如表6.4 BOOT模式配置所示：

表 6.4 BOOT模式配置

BOOT MODE	BOOT GPIO 1	BOOT GPIO 0
eFlash ADDR0 加载模式	0	0
UART-A	0	1
SPI-A	1	0
NVR BOOT	1	1

如果BOOT GPIO 0/1值均为1，则代表选择NVR配置文件中的启动模式，其包含了14种启动模式，如表6.5 所示：

表 6.5 NVR配置文件中的启动模式

BOOT MODE	描述	BOOT MODE	描述
无效值	eFLASH BOOT addr0	0x07	CAN-B
0x00	eFLASH addr0	0x08	CAN-C
0x01	eFLASH addr1	0x09	SPI-A
0x02	eFLASH addr2	0x0a	SPI-B
0x03	UART-A	0x0b	SPI-C
0x04	UART-B	0x0c	I2C-A
0x05	UART-C	0x0d	I2C-B
0x06	CAN-A		

系统默认启动模式为EFLASH ADDR0模式，CPU1支持BOOTROM从三个EFLASH地址启动，配置如下：

- EFLASH ADDR0 0x00300000
- EFLASH ADDR1 0x00340000
- EFLASH ADDR2 0x00380000

6.11.2 CPU1 BOOT外设PIN配置

BOOTMODE.bin文件中同时定义了该芯片支持的所有外设接口的PINMUX，通过定义不同的BOOTMODE.bin，可以方便客户自主的定义自己可以支持的IO口，方便用户设计。

芯片BOOTROM支持UART，CAN，SPI，I2C四种接口作为bootloader外设接口，支持将打包的APP代码download到指定地址并运行。表6.6显示了所有可被配置为外设bootloader的GPIO管脚。

表 6.6 NVR配置文件中的启动模式

外设接口	GPIO PIN设置	备注
UART-A Boot	TX: GPIO-8/29/36/42/48/65/84 RX: GPIO-9/28/35/43/49/64/85	目前波特率只支持38400
UART-B Boot	TX: GPIO-9/10/14/18/22/54/70/86 RX: GPIO-11/15/19/23/55/71/87	目前波特率只支持38400
UART-C Boot	TX:GPIO-12/38/56/63/72/89 RX:GPIO-13/39/57/62/73/90	目前波特率只支持38400
SPI-A Boot	MOSI: GPIO-16/54/58 MISO: GPIO-17/55/59 CLK: GPIO-18/56/60 CS: GPIO-19/57/61	
SPI-B Boot	MOSI: GPIO-24/60/63 MISO: GPIO-25/61/64 CLK: GPIO-22/26/58/65 CS: GPIO-23/27/59/66	
SPI-C Boot	MOSI: GPIO-50/69 MISO: GPIO-51/70 CLK: GPIO-52/71 CS: GPIO-53/72	
I2C-A Boot	SCL: GPIO-1/33/43/92 SDA: GPIO-0/32/42/91	支持配置I2C DEVICE的ADDR，以及数据模式
I2C-B Boot	SCL: GPIO-3/35/41/69 SDA: GPIO-2/34/40/66	支持配置I2C DEVICE的ADDR，以及数据模式
CAN-A Boot	TX: GPIO-4/19/31/37/63/71 RX: GPIO-5/18/30/36/62/70	支持500K，250K，125K波特率
CAN-B Boot	TX: GPIO-6/8/12/16/20/38/72 RX: GPIO-7/10/13/17/32/39/73	支持500K，250K，125K波特率
CAN-C Boot	TX: GPIO-4/8/19/22/31/37/71/74 RX: GPIO-5/10/18/23/30/36/70/75	支持500K，250K，125K波特率

6.11.3 CPU2 BOOT模式

CPU1在启动后，会release CPU2，CPU2在被唤起后，会默认进入wait模式，等待CPU1的IPC命令，CPU1的APP通过发送IPC BOOT MODE给CPU2，使CPU2进入对应的启动模式。IPC BOOTMODE配置如表6.7所示：

表 6.7 IPC BOOTMODE模式

IPC BOOT MODE	VALUE
BOOTMODE_WAIT	0x5A00
BOOTMODE_EFLASH0	0x5A01
BOOTMODE_ITCM0	0x5A02
BOOTMODE_SRAM0	0x5A03
BOOTMODE_WAIT_IPCCMD	0x5A04
BOOTMODE_ERR_RESET	0x5A05
BOOTMODE_ERR_NOHANDLE	0x5A06

6.11.4 Timer

本节介绍了图6.11中所示的三个64位CPU定时器（定时器0/1/2）。CPU定时器2保留为实时操作系统使用（例如RTOS），但如果它没有被实时操作系统使用，那么CPU定时器2也可以用于其他应用程序。CPU定时器0和CPU定时器1运行于系统时钟CPU_x.SYSPCLK。CPU定时器2默认运行于CPU_x.SYSPCLK，但也可以使用INTOSC1、INTOSC2、XTAL和AUXPLLCLK。CPU定时器0/1/2分别产生TINT0、TINT1、TINT2三个中断信号连接给CPU。

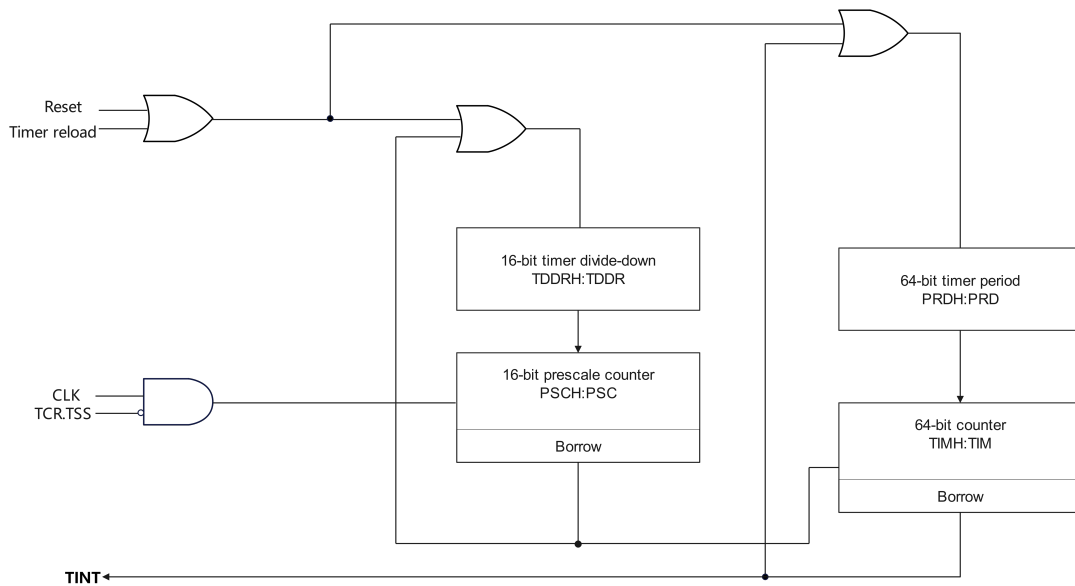


图 6.11 64位CPU定时器

6.11.5 NMIWD

每个CPU都有一个用户可编程的不可屏蔽中断看门狗模块（NMIWD），用户可以按照需求设置一个阈值，当事件发生后，阈值计数器开始按照SYSPCLK周期递减。如果在阈值计数器减为0之前用户没有处理事件，则NMIWD模块将发起复位。

6.12 watchdog

看门狗（watchdog）模块支持以下功能：

- 32位看门狗计数器
- 超时范围可编程
- 当发生超时，可以选择：
 - 直接产生复位
 - 第一次超时只产生中断，第二次超时产生复位
- 复位脉冲宽度可编程

6.13 CLB

可配置逻辑块（CLB）是可以使用软件互连的模块集合，实现自定义数字逻辑功能或增强现有的片上外设。CLB能够通过一组交叉互连来增强现有的外围设备，为现有的控制外设提供高性能的连接，如增强型脉宽调制器（ePWM）、增强型捕获模块（eCAP）和增强型正交编码器脉冲模块（eQEP）。CLB还可以通过XBAR连接到外部GPIO引脚。通过这种方式，可以将CLB配置为与设备外设交互，以执行小型逻辑功能，如比较器，或实现自定义串行数据交换协议等。需要使用外部逻辑器件完成的功能现在通过CLB在MCU内部实现。

CLB可通过CLB tool进行配置。CLB模块及其内部互连图如图6.12所示。

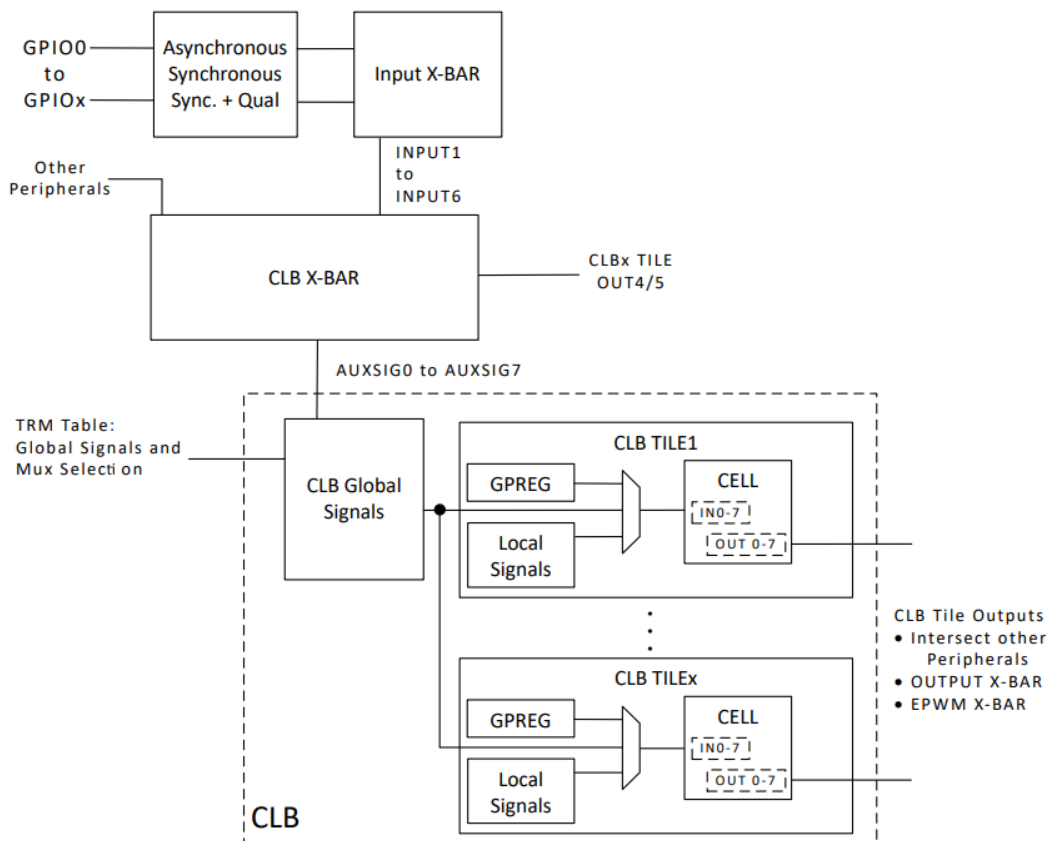


图 6.12 64位CPU定时器

7 封装规格

