

ADP32F10, ADP32F11, ADP32F12

数字信号处理器

数据手册

编号：JXDZ7.381.002 SJSC

Advancechip



Electronics

湖南进芯电子科技有限公司

2019年7月

V1.3

历史版本记录

版本号	时间	起草/ 修改人	内容描述	审核人	批准人
V1.0	2018-4-1	刘律辑	首次发布		
V1.1	2018-11-28	刘律辑	修改 GPIOG 的 IO 命名		
V1.2	2019-3-22	刘律辑	修订 P11 主频数据		
V1.3	2019-7-11	刘律辑	修改 ADP32F10 的闪存描述		

目录

目录.....	3
1 产品特点.....	6
1.1 芯片描述.....	9
1.2 产品编码.....	12
1.3 信号说明.....	14
1.4 引脚命名简称与全称对应表.....	23
2 功能概述.....	24
2.1 内存映射.....	25
2.2 简要说明.....	30
2.2.1 ADP32F1X CPU.....	30
2.2.2 内存总线（哈佛总线架构）.....	30
2.2.3 外设总线.....	30
2.2.4 实时 JTAG 和分析.....	31
2.2.5 外部接口（XINTF）（只适用于 ADP32F12）.....	31
2.2.6 闪存.....	31
2.2.7 ROM.....	31
2.2.8 M0, M1 SARAM.....	32
2.2.9 L0, L1, H0 SARAM.....	32
2.2.10 引导 ROM.....	32
2.2.11 安全性.....	33
2.2.12 外设中断扩展(PIE)块.....	33
2.2.13 外部中断(INT1, INT2, INT13, NMI).....	34
2.2.14 振荡器和锁相环(PLL).....	34
2.2.15 看门狗定时器.....	34
2.2.16 外设时钟.....	34
2.2.17 低功耗模式.....	34
2.2.18 外设帧 0,1,2 (PFn).....	35
2.2.19 通用输入/输出(GPIO)复用器.....	35
2.2.20 32 位 CPU 定时器(0, 1, 2).....	35
2.2.21 控制外设.....	35
2.2.22 串行端口外设.....	36
2.3 寄存器映射.....	36
2.4 器件仿真寄存器.....	38
2.5 外部接口, XINTF(只适用于 ADP32F12).....	38
2.5.1 时序寄存器.....	40
2.5.2 XREVISION 寄存器.....	40
2.6 中断.....	41

ADP32FXX Digital Signal Processor

2.7 系统控制.....	43
2.8 OSC 和 PLL 块.....	46
2.8.1 输入时钟丢失.....	47
2.9 基于 PLL 的时钟模块.....	47
2.10 外部参考振荡器时钟选项.....	48
2.11 看门狗模块.....	49
2.12 低功耗模式.....	50
3 外设.....	51
3.1 32 位 CPU 定时器 0/1/2.....	51
3.2 事件管理器模块(EM1 , EM2).....	54
3.2.1 通用(GP) 定时器.....	56
3.2.2 完全比较单元.....	57
3.2.3 可编程死区发生器.....	57
3.2.4 PWM 波形生成.....	57
3.2.5 双更新 PWM 模式.....	57
3.2.6 PWM 特性.....	58
3.2.7 捕捉单元.....	58
3.2.8 正交编码器脉冲(QEP) 电路.....	58
3.2.9 外部 ADC 转换开始.....	59
3.3 增强型模数转换器(ADC) 模块.....	59
3.4 增强型控制器局域网络(ECAN) 模块.....	65
3.5 多通道缓冲串行端口(McBSP) 模块.....	69
3.6 串行通信接口(SCI) 模块.....	72
3.7 串行外设接口(SPI) 模块.....	76
3.8 GPIO MUX.....	79
3.9 片上 LDO 电源管理.....	82
4 开发支持.....	83
软件开发工具.....	83
硬件开发工具.....	83
5 电气规范.....	84
5.1 绝对最大额定值 ⁽¹⁾	84
5.2 建议的运行条件 ⁽¹⁾	84
5.3 建议运行条件下的电气特性 (除非另外注明)	85
5.4 流耗.....	86
5.5 流耗图.....	87
5.6 减少流耗.....	88
5.7 针对 DSP 的无信号缓冲的仿真器连接.....	88
5.8 电源时序要求.....	89

ADP32FXX Digital Signal Processor

5.9 信号转换电平.....	90
5.10 时序参数符号.....	91
5.11 定时参数的通用注释.....	92
5.12 测试负载电路.....	92
5.13 器件时序表.....	93
5.14 时钟要求和特性.....	94
5.14.1 输入时钟要求.....	94
5.14.2 输出时钟特性.....	95
5.15 复位时序.....	96
5.16 低功耗模式唤醒时序.....	100
5.17 事件管理器.....	103
5.17.1 PWM 时序.....	103
5.17.2 中断时序.....	105
5.18 通用输入/输出(GPIO) - 输出时序.....	106
5.19 通用输入/输出(GPIO) - 输入时序.....	106
5.20 串行外设接口(SPI) 主控模式时序.....	108
5.21 串行外设接口(SPI) 受控模式时序.....	112
5.22 外部接口(XINTF) 时序.....	114
5.23 XINTF 信号与 CLKOUT 对齐.....	117
5.24 外部接口读取时序.....	119
5.25 外部接口写入时序.....	121
5.26 带有一个外部等待状态的外部接口读取准备就绪时序.....	122
5.27 带有一个外部等待状态的外部接口写入准备就绪时序.....	126
5.28 HOLD 和 HOLDA.....	129
5.29 HOLD 和 HOLDA 时序.....	129
5.30 片载模数转换器.....	131
5.30.1 ADC 绝对最大额定值.....	131
5.30.2 在推荐运行条件下的 ADC 电气特性.....	132
5.30.3 针对不同 ADC 配置的流耗.....	133
5.30.4 ADC 加电控制位时序.....	134
5.30.5 详细说明.....	134
5.30.6 顺序采样模式 (单通道) (SMODE = 0).....	135
5.30.7 同步采样模式 (双通道) (SMODE=1).....	136
5.30.8 技术规范和术语的定义.....	137
5.31 多通道缓冲串行端口(McBSP) 模块.....	138
5.31.1 McBSP 发送和接收时序.....	138
5.31.2 McBSP 作为 SPI 主控或者受控时序.....	140
5.32 闪存时序.....	143
5.33 ROM 时序.....	144
6 机械数据.....	146

1 产品特征

•采用高性能 CMOS 工艺

- 主频 150MHz (周期 6.67ns)
- 低功耗设计 (内核 1.8V , I/O 电压 3.3V)
- 集成片内 1.8V 线性稳压电源 (LDO)

•支持 JTAG 在线仿真

•高性能 32 位 CPU

- 单周期 32 位 x32 位乘累加(MAC)运算
- 单周期 2 个 16 位 x16 位乘累加(MAC)运算
- 哈佛(Harvard)总线架构
- 原子操作
- 快速中断响应和处理
- 统一寄存器编程模式
- 4M 线性程序/数据地址
- 可使用 C/C++ 和汇编语言高效率编程

•片内存储器资源

- 闪存 : 128K x 16 位闪存
(4 个 8Kx16 位和 6 个 16Kx16 位扇区)
- ROM : 128K x 16 位 ROM (注 : C 版 , 需客户定制)
- 1K x 16 一次性可编程(OTP) ROM
- L0 和 L1 : 2 块 4Kx16 位独立寻址 SARAM
- H0 : 1 块 8K x 16 位 SARAM
- M0 和 M1 : 2 块 1Kx16 位独立寻址 SARAM

•引导 ROM (4K X 16 位)

- 支持软件引导模式
- 自带标准算术表

•外部存储扩展接口

- 超过 1M x 16 位的可扩展空间
- 可编程等待状态
- 可编程读/写时序
- 3 个独立片选信号

•时钟和系统控制

- 支持动态锁相环(PLL)分频系数调整
- 片内振荡器
- 看门狗定时器

•三个外部中断接口

•支持 45 个外设中断的外设中断扩展块(PIE)

•三个 32 位的 CPU 定时器

•128 位安全密钥/锁

- 保护闪存/ROM/OTP 和 L0/L1 SARAM
- 防止固件逆向破解

•电机控制外设 (PWM 产生电路)

- 事件管理器 1(EM1), 事件管理器 2(EM2)
- 分别包括: 2 个 16 位定时器, 3 个数值比较器, 3 个捕获单元, 1 个正交编码电路。

•串行端口外设

- 串行外设接口(SPI)
- 2 个串行通信接口(SCI), 兼容通用异步收发(UART) 标准
- 增强型控制器局域网络(eCAN)控制器, 集成片内 eCAN 驱动器
- 多通道缓冲串行端口(McBSP)

•12 位 16 通道模数转换器(ADC)

- 2 x 8 通道输入复用器
- 两个采样保持电路
- 单一/同步转换
- 快速转换速率: 80ns/12.5MSPS

- **56 个通用 I/O (GPIO) 引脚**

- **先进的仿真功能**

- 分析和断点功能
- 基于硬件的实时调试

- **开发工具**

- ANSI C/C++ 编译器/汇编语言/连接器
- 支持 Code Composer Studio™ IDE
- 支持 DSP/BIOS™
- JTAG 仿真器

- **低功耗模式和省电模式**

- 支持 IDLE (空闲)、STANDBY (待机)、HALT (暂停) 模式
- 可禁用独立外设时钟

- **强大的静电泄放(ESD)防护能力**

- ESD 人体模式(HBM): +4000V/-4000V
- ESD 机器模式(MM): +400V/-400V
- 闩锁效应(Latch-up)触发电流: 400mA

- **封装选项**

- BGA 179 (带外部存储器接口)
- LQFP176 (带外部存储器接口)
- LQFP128 (无外部存储器接口)

- **产品等级**

- S : 工业级
- A : 消费级

1.1 芯片描述

ADP32F12-BGA, 179PIN 封装引脚分配图,如图 1.2-1 所示

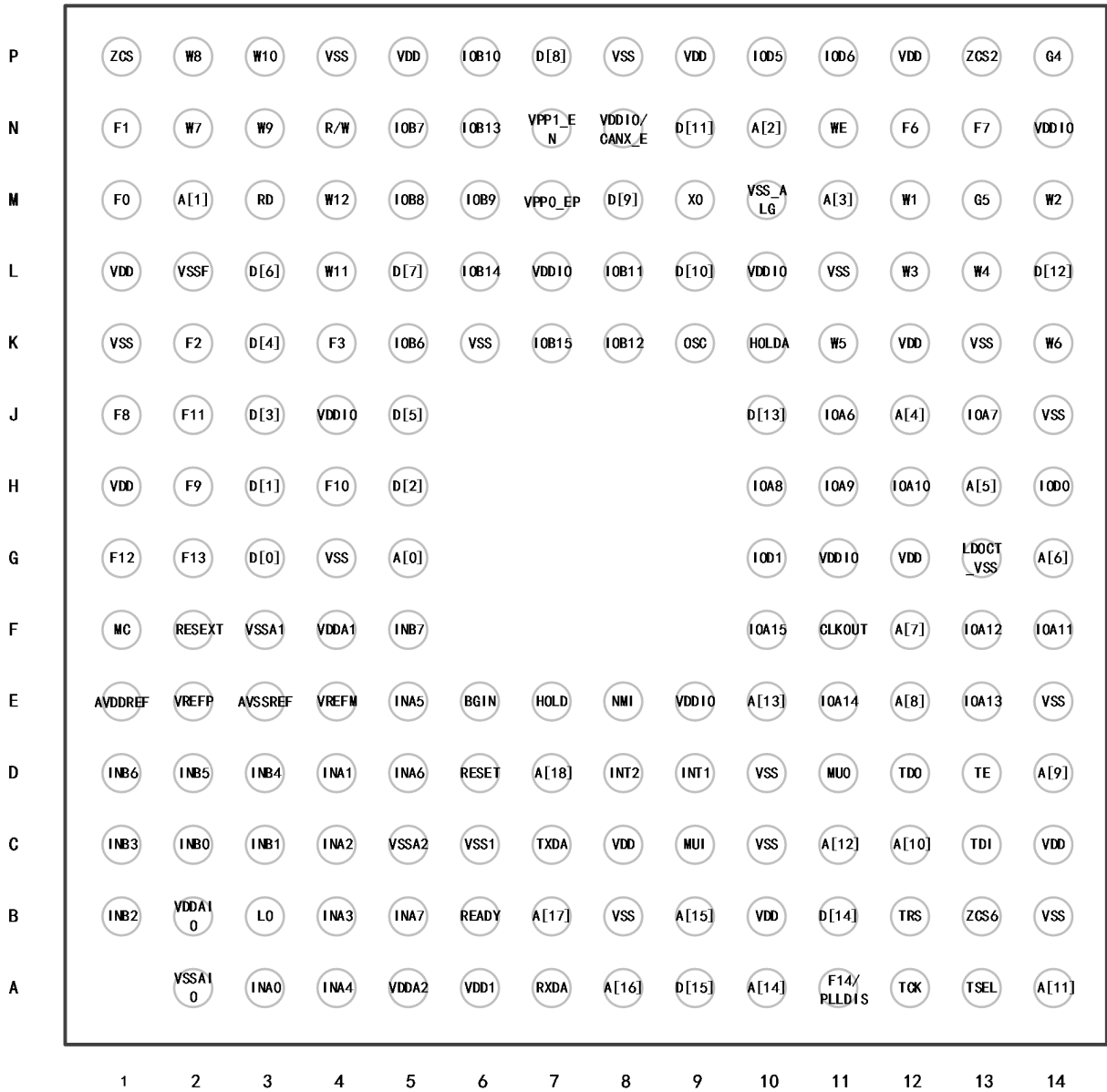


图 1.2-1 封装引脚分布图 (底视图)

ADP32FXX Digital Signal Processor

ADP32F12-LQFP,176PIN 封装引脚分配图,如图 1.2-2 所示

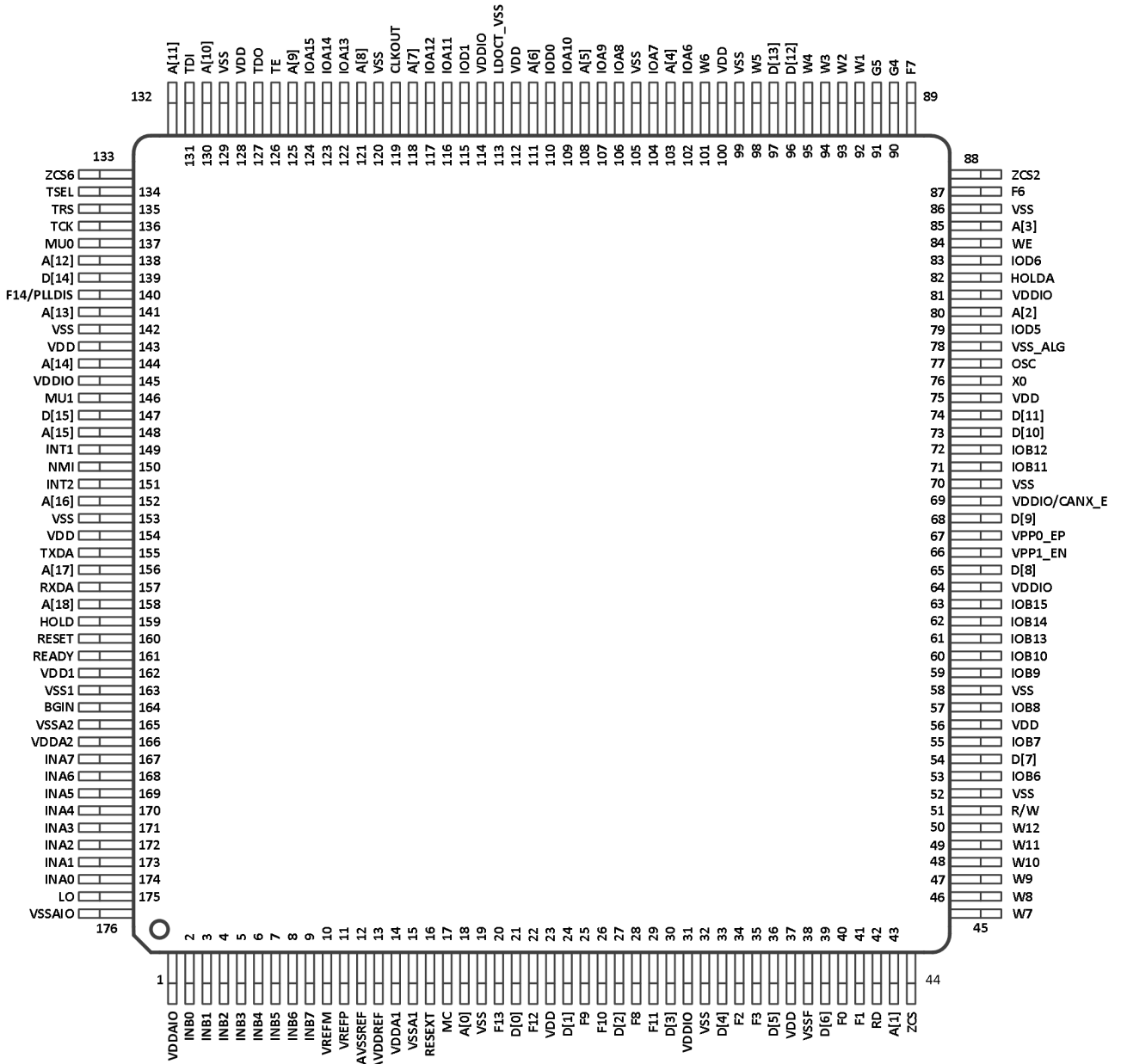


图 1.2-2 封装引脚图(顶视图)

ADP32FXX Digital Signal Processor

ADP32F10-LQFP , 128PIN 封装引脚分配图,如图 1.2-3 所示

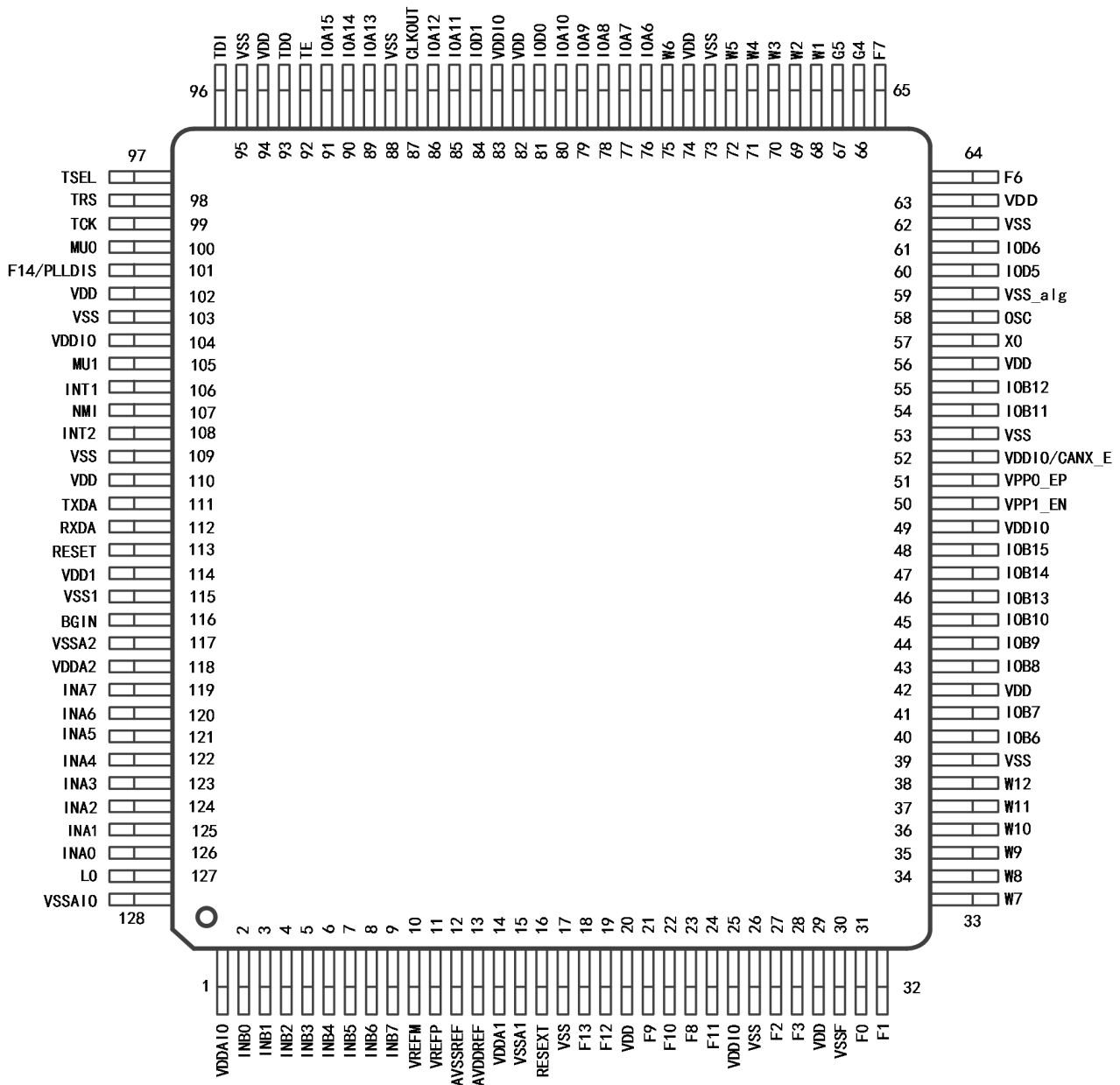
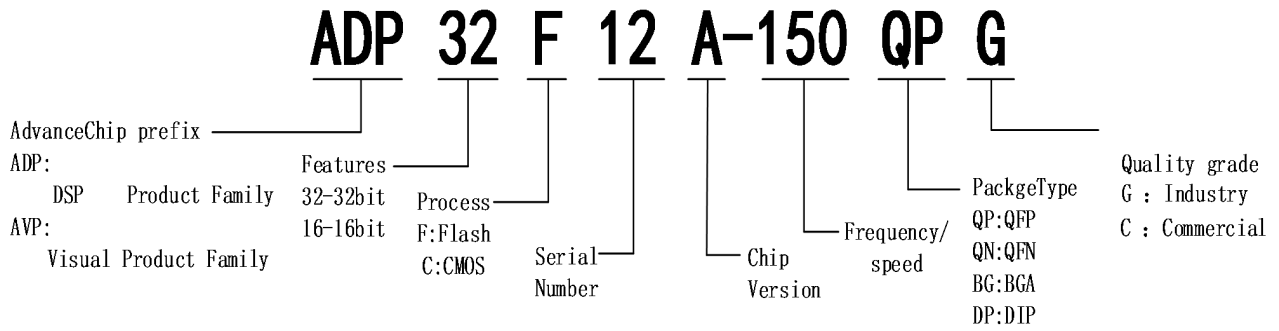


图 1.2-3 封装引脚图(顶视图)

ADP32FXX Digital Signal Processor

1.2 产品编码

ADP32FXX 产品代号编码规则:



ADP32FXX Digital Signal Processor

ADP32FXX 产品代号编码说明:

特征	ADP32F10	ADP32F11	ADP32F12
指令周期	6.67ns (@150Mhz)	6.67ns (@150Mhz)	6.67ns (@150Mhz)
片上 SRAM (16bt word)	18k	18k	18k
片上 Flash (16bt word)	128K	128K	128K
BOOT ROOM (16bt word)	4K	4K	4K
片上 OTP ROM(1K*16)	是	是	是
程序代码保护功能	是	是	是
外部存储接口	否	否	是
芯片 ID 激光熔丝	是	是	是
EM1/EM2	EM1/EM2	EM1/EM2	EM1/EM2
—GP timer	4	4	4
—比较器/PWM	16	16	16
capture/QEP 通道	6/2	6/2	6/2
GPIO(shared)	56	56	56
Watchdog timer	是	是	是
32 位 CPU Time	3	3	3
12-bit ADC	80ns	80ns	80ns
通道数	2*8 通道, 双采样保持	2*8 通道, 双采样保持	2*8 通道, 双采样保持
SPI	是	是	是
SCIA, SCIB	是	是	是
CAN host/slave	是	是	是
片上 CAN driver	是	是	是
多通道串口 McBSP	是	是	是
时钟	是	是	是
晶振驱动	1	1	1
片上振荡器	1	1	1
片上 PLL	1	1	1
供电电压	3v3 I/O, 1v8 Core		
片上 LDO	可配置, 使用片上 LDO 时, VDD 需置外接电容 (仅支持 ADP32F12)		
Package	128-pin QFP	128-pin QFP	176-pin QFP/179-pin BGA
温度选择	A : -40~85°C ; S : -40~105°C ;		

1.3 信号说明

表 1.5-1 指定了 ADP32FXX 器件的信号。所有数字输入是 TTL 兼容的。所有输出为 3.3V CMOS 电平输出。输入不是 5V 耐压，输入端口均通过 100 μ A (或者 20 μ A) 的上拉/下拉(片内电阻接电源或地)。

表 1.3-1 信号说明

名称	引脚编号			I/O/Z ⁽²⁾	PU/PD ⁽³⁾	说明
	179 焊球 BGA 封装	176 引脚 QFP 封装	128 引脚 QFP 封装			
XINTF 信号						
A[18]	D7	158	-	O/Z	-	19 位 XINTF 地址总线。
A[17]	B7	156	-	O/Z	-	
A[16]	A8	152	-	O/Z	-	
A[15]	B9	148	-	O/Z	-	
A[14]	A10	144	-	O/Z	-	
A[13]	E10	141	-	O/Z	-	
A[12]	C11	138	-	O/Z	-	
A[11]	A14	132	-	O/Z	-	
A[10]	C12	130	-	O/Z	-	
A[9]	D14	125	-	O/Z	-	
A[8]	E12	121	-	O/Z	-	
A[7]	F12	118	-	O/Z	-	
A[6]	G14	111	-	O/Z	-	
A[5]	H13	108	-	O/Z	-	
A[4]	J12	103	-	O/Z	-	
A[3]	M11	85	-	O/Z	-	
A[2]	N10	80	-	O/Z	-	
A[1]	M2	43	-	O/Z	-	
A[0]	G5	18	-	O/Z	-	
D[15]	A9	147	-	I/O/Z	PU	16 位 XINTF 数据总线。
D[14]	B11	139	-	I/O/Z	PU	
D[13]	J10	97	-	I/O/Z	PU	
D[12]	L14	96	-	I/O/Z	PU	
D[11]	N9	74	-	I/O/Z	PU	
D[10]	L9	73	-	I/O/Z	PU	
D[9]	M8	68	-	I/O/Z	PU	
D[8]	P7	65	-	I/O/Z	PU	
D[7]	L5	54	-	I/O/Z	PU	
D[6]	L3	39	-	I/O/Z	PU	
D[5]	J5	36	-	I/O/Z	PU	
D[4]	K3	33	-	I/O/Z	PU	
D[3]	J3	30	-	I/O/Z	PU	
D[2]	H5	27	-	I/O/Z	PU	
D[1]	H3	24	-	I/O/Z	PU	
D[0]	G3	21	-	I/O/Z	PU	

表 1.3-1 信号说明⁽¹⁾ (续)

名称	引脚编号			I/O/Z ⁽²⁾	PU/PU ⁽³⁾	说明
	179 焊球 BGA 封装	176 引脚 QFP 封装	128 引脚 QFP 封装			
MC	F1	17	-	I	PD	微处理器/微计算机模式选择。微处理器和微计算机模式间的切换。当处于高电平时，区域 7 外部接口上启用。当为低电平时，区域 7 被从外部接口禁用，可访问片载引导 ROM 作为替代。在复位时，这个信号被锁存在 XINTCNF2 寄存器中，并且用户可在软件中修改这个位。复位后，MC 引脚的状态被忽略。
HOLD	E7	159	-	I	PU	外部保持请求。HOLD，当有效时（低电平），请求 XINTF 释放外部总线并将所有总线和选通闸门置于一个高阻抗状态。当任一当前的访问完成并且在 XINTF 上没有等待的访问时，XINTF 将释放总线。外部保持应答。当 XINTF 已经准予一个 HOLD 请求时，HOLDA 被驱动至有效（低电平）。所有 XINTF 总线和选通闸门将处于高阻抗状态。当 HOLD 信号被释放时，HOLDA 被释放。HOLDA 为有效（低电平）时，外部器件应该只驱动外部总线。
HOLDA	K10	82	-	O/Z	-	XINTF 区域 0 和区域 1 芯片选择。当执行一个到 XINTF 区域 0 或者区域 1 的访问时，ZCS 为低电平。
ZCS	P1	44	-	O/Z	-	XINTF 区域 2 芯片选择。当执行到 XINTF 区域 2 的访问时，ZCS2 为有效（低电平）。
ZCS2	P13	88	-	O/Z	-	XINTF 区域 6 和区域 7 芯片选择。当执行一个到 XINTF 区域 6 或者区域 7 的访问时，ZCS6 为有效（低电平）。
ZCS6	B13	133	-	O/Z	-	写入启用。低电平有效写入选通脉冲。以每个区域为基础，写入选通脉冲的波形由 XTIMINGx 寄存器内的建立、激活、和跟踪周期指定。
WE	N11	84	-	O/Z	-	读取启用。低电平有效读取选通脉冲。以每个区域为基础，读取选通脉冲的波形由 XTIMINGx 寄存器内的建立、激活、和跟踪周期指定。注释：RD 和 WE 信号是相互排斥的。
RD	M3	42	-	O/Z	-	只读不写选通脉冲。通常保持高电平。当为低电平时，R/W 表明写入周期被激活；当为高电平时，R/W 表明读取周期有效。
R/W	N4	51	-	O/Z	-	就绪信号。当被置为 1 时，表明外设已经为完成访问做好准备。READY 可被配置成一个同步或者异步输入。更多细节请见时序图。
READY	B6	161	-	I	PU	

(1)对于除 TDO, CLKOUT, XF, XINTF, MU0, 和 MU1 引脚（输出缓冲器的驱动强度为 8mA 之外的所有引脚，输出缓冲器的典型驱动强度为 4mA。

(2) I= 输入, O=输出, Z=高阻抗

(3) PU= 引脚有内部上拉电阻；PD=引脚有内部下拉电阻。推荐运行条件下电气特性，中指定了上拉/下拉强度，在边界扫描模式中，上拉/下拉被启用。

表 1.3-1 信号说明⁽¹⁾ (续)

名称	引脚编号			I/O/Z ⁽²⁾	PU/PU ⁽³⁾	说明
	179 焊球 BGA 封装	176 引脚 QFP 封装	128 引脚 QFP 封装			
JTAG 和混合信号						
OSC	K9	77	58	I	-	振荡器输入，输入到内部振荡器。这个引脚也被用于提供一外部时钟。ADP32FXX 可由一个外部时钟源操作，只要适当的电压电平在 OSC 引脚上被驱动。应该注意的是，OSC 引脚以 1.8V (或者 1.9V) 内核数字电源(VDD)为基准，而不是 3.3V I/O 电源 (VDDIO)。外部时钟 0~1.8V 可直接驱动该端口；外部时钟 0~3.3V 逻辑电平也可直接驱动该端口，无需外加钳位二极管进行端口的保护。
X0	M9	76	57	O	-	振荡器输出
CLKOUT	F11	119	87	O	-	取自 SYSCLKOUT 的输出时钟被用于外部等待状态生成并作为一个通用时钟源。CLKOUT 或者与 SYSCLKOUT 的频率相同，或者为 SYSCLKOUT 频率的 1/2 或者 1/4。复位时，CLKOUT=SYSCLKOUT/4。通过将 XINTCNF2 寄存器中的位 3 (CLKOFF) 设定为 1，可将 CLKOUT 信号关闭。与其 GPIO 引脚不同，复位时，不将 CLKOUT 引脚置于一个高阻抗状态。
TSEL	A13	134	97	I	PD	测试引脚。必须接地。
RESET	D6	160	113	I/O	PU	器件复位 (输入) 和安全装置复位 (输出)。器件复位。RESET 导致器件终止执行。PC 将指向包含在位置 0x3FFFC0 中的地址。当 RESET 被置为高电平时，在 PC 指向的位置开始执行。当一个安全装置复位发生时，这个引脚被 DSP 驱动至低电平。安全装置复位期间，RESET 引脚将在 512 个 OSC 周期的安全装置复位持续时间内被驱动至低电平。这个引脚的输出缓冲器是一个有内部上拉电阻 (100μA，典型值) 的开漏器件。建议由一个开漏器件驱动这个引脚。
VPP0_EP	M7	67	51	I/O/Z	-	VPP0 测试端/片内 ECAN 收发器 EP 端(CAN_H)
VPP1_EN	N7	66	50	I/O/Z	-	VPP1 测试端/片内 ECAN 收发器 EN 端(CAN_L)

表 1.3-1 信号说明⁽¹⁾ (续)

名称	引脚编号			I/O/Z ⁽²⁾	PU/PU ⁽³⁾	说明
	179 焊球 BGA 封装	176 引脚 QFP 封装	128 引脚 QFP 封装			
						JTAG
TRS	B12	135	98	I	PD	<p>使用内部下拉进行 JTAG 测试复位。TRS 当被驱动为高电平时，使扫描系统获得器件运行的控制权。如果这个信号未连接或者被驱动至低电平，此器件在功能模式下运转，并且测试复位信号被忽略。</p> <p>注释：不要在 TRS 上使用上拉电阻器；它有一个内部下拉器件。TRS 是一个高电平有效测试引脚并且必须在正常器件运行期间一直保持低电平。在低噪声环境中，TRS 可保持悬空。在其它情况下，强烈建议使用一个外部下拉电阻器。这个电阻器的值应该基于适用于这个设计的调试器推进源代码的驱动强度。一个 2.2kΩ 电阻器一般提供足够的保护，由于这是应用专用的，建议针对调试器和应用的正确运行对每个目标板进行验证。</p>
TCK	A12	136	99	I	PU	<p>带有内部上拉电阻的 JTAG 测试时钟</p>
TE	D13	126	92	I	PU	<p>带有内部上拉电阻的 JTAG 测试模式选择(TE)。这个串行控制输入在 TCK 上升沿锁存到 TAP 控制器中</p>
TDI	C13	131	96	I	PU	<p>带有内部上拉电阻的 JTAG 测试数据输入(TDI)。TDI 在 TCK 上升沿锁存到寄存器（指令或者数据）</p>
TDO	D12	127	93	O/Z	-	<p>JTAG 扫描输出，测试数据输出(TDO)。所选寄存器（指令或者数据）的内容被在 TCK 下降沿通过 TDO 移出。</p>
MU0	D11	137	100	I/O/Z	PU	<p>仿真器引脚 0。当 TRS 被驱动至高电平时，这个引脚被用作一个到（或者来自）仿真器系统的中断并且在 JTAG 扫面过程中被定义为输入/ 输出。这个引脚也被用于将器件置于边界扫面模式中。在 MU0 引脚处于逻辑高电平状态并且 MU1 引脚处于逻辑低电平状态时，TRST 引脚的上升沿将把器件锁存在边界扫面模式。</p> <p>注释：建议在这个引脚上连接一个外部上拉电阻器。这个电阻器的值应该基于适用于这个设计的调试器推进源代码的驱动强度。通常一个 2.2kΩ 至 4.7kΩ 的电阻器已可以满足要求。由于这是应用专用的，建议针对调试器和应用正确运行对每个目标板进行验证。</p>

表 1.3-1 信号说明⁽¹⁾ (续)

名称	引脚编号			I/O/Z ⁽²⁾	PU/PU ⁽³⁾	说明
	179 焊球 BGA 封装	176 引脚 QFP 封装	128 引脚 QFP 封装			
MU1	C9	146	105	I/O/Z	PU	<p>仿真器引脚 1。当 TRS 被驱动至高电平时，这个引脚被用作一个到（或者来自）仿真器系统的中断并且在 JTAG 扫描过程中被定义为输入/输出。这个引脚也被用于将器件置于边界扫描模式中。在 MU0 引脚处于逻辑高电平状态并且 MU1 引脚处于逻辑低电平状态时，TRS 引脚的上升沿将把器件锁存在边界扫描模式。</p> <p>注释：建议在这个引脚上连接一个外部上拉电阻器。这个电阻器的值应该基于适用于这个设计的调试器推进源代码的驱动强度。通常一个 2.2kΩ 至 4.7kΩ 的电阻器已可以满足要求。由于这是应用专用的，建议针对调试器和应用的正确运行对每个目标板进行验证。</p>
ADC 模拟输入信号						
INA7	B5	167	119	I	-	<p>针对采样和保持 A 的 8 通道模拟输入。在 VDDA1, VDDA2, 和 VDDAIO 引脚被完全加电之前，不应驱动 ADC 引脚。</p>
INA6	D5	168	120	I	-	
INA5	E5	169	121	I	-	
INA4	A4	170	122	I	-	
INA3	B4	171	123	I	-	
INA2	C4	172	124	I	-	
INA1	D4	173	125	I	-	
INA0	A3	174	126	I	-	
INB7	F5	9	9	I	-	
INB6	D1	8	8	I	-	
INB5	D2	7	7	I	-	<p>针对采样和保持 B 的 8 通道模拟输入。在 VDDA1, VDDA2, 和 VDDAIO 引脚被完全加电之前，不应驱动 ADC 引脚。</p>
INB4	D3	6	6	I	-	
INB3	C1	5	5	I	-	
INB2	B1	4	4	I	-	
INB1	C3	3	3	I	-	
INB0	C2	2	2	I	-	
VREFP	E2	11	11	I/O	-	<p>ADC 电压基准输出 (2V)。要求通过一个 10μF 的低 ESR (低于 1.5Ω) 陶瓷旁路电容器接至模拟接地。[如果软件位被针对这个模式启用，可接受外部基准输入(2V)。1-10μF 低 ESR 电容器可被用在外部基准模式中。]</p> <p>注释：使用 ADC 时钟速率从系统中使用的电容器数据表中得出 ESR 技术规格。</p>

表 1.3-1 信号说明⁽¹⁾ (续)

名称	引脚编号			I/O/Z ⁽²⁾	PU/PU ⁽³⁾	说明
	179 焊球 BGA 封装	176 引脚 QFP 封装	128 引脚 QFP 封装			
VREFM	E4	10	10	I/O		ADC 电压基准输出 (1V)。要求通过一个 10 μ F 的低 ESR (低于 1.5 Ω) 陶瓷旁路电容器接至模拟接地。[如果软件位被针对这个模式启用, 可接受外部基准输入(1V)。1-10 μ F 低 ESR 电容器可被用在外部基准模式中。] 注释: 使用 ADC 时钟速率从系统中使用的电容器数据表中得出 ESR 技术规格。
RESEXT	F2	16	16	O		ADC 外部电流偏置电阻器。针对 1-18.75MHz 的 ADC 时钟范围, 使用 24.9k Ω \pm 5% 的电阻值; 针对 18.75MHz-25MHz 的 ADC 时钟范围, 使用 20k Ω \pm 5% 的电阻值
BGIN	E6	164	116	I		测试引脚。必须被保持未连接。
AVSSREF	E3	12	12	I		ADC 模拟 GND
AVDDREF	E1	13	13	I		ADC 模拟电源 (3.3V)
LO	B3	175	127	I		常用低侧模拟输入。连接到模拟接地。
VSSA1	F3	15	15	I		ADC 模拟 GND
VSSA2	C5	165	117	I		ADC 模拟 GND
VDDA1	F4	14	14	I		ADC 模拟 3.3V 电源
VDDA2	A5	166	118	I		ADC 模拟 3.3V 电源
VSS1	C6	163	115	I		ADC 数字 GND
VDD1	A6	162	114	I		ADC 数字 1.8V (或 1.9V) 电源
VDDAIO	B2	1	1			3.3V 模拟 I/O 电源引脚
VSSAIO	A2	176	128			模拟 I/O 接地引脚
电源信号						
VDD	H1	23	20			1.8V 或者 1.9V 内核数字电源引脚
VDD	L1	37	29			
VDD	P5	56	42			
VDD	P9	75	56			
VDD	P12	-	63			
VDD	K12	100	74			
VDD	G12	112	82			
VDD	C14	128	94			
VDD	B10	143	102			
VDD	C8	154	110			

表 1.3-1 信号说明⁽¹⁾(续)

名称	引脚编号			I/O/Z ⁽²⁾	PU/PU ⁽³⁾	说明	
	179 焊球 BGA 封装	176 引脚 QFP 封装	128 引脚 QFP 封装				
VSS	G4	19	17				
VSS	K1	32	26				
VSSF	L2	38	30				
VSS	P4	52	39				
VSS	K6	58	-				
VSS	P8	70	53				
VSS_alg	M10	78	59			内核与数字 I/O 接地脚。	
VSS	L11	86	62				
VSS	K13	99	73				
VSS	J14	105	-			注释：LDOCT_VSS 引脚默认接地，用户需要启动片内 LDO 功能时，需将该引脚作为控制引脚单独引出	
LDOCT_VSS	G13	113	-				
VSS	E14	120	88				
VSS	B14	129	95				
VSS	D10	142	-				
VSS	C10	-	103				
VSS	B8	153	109				
VDDIO	J4	31	25				
VDDIO	L7	64	49				
VDDIO	L10	81	-			3.3V I/O 数字电源引脚	
VDDIO	N14	-	-				
VDDIO	G11	114	83				
VDDIO	E9	145	104				
VDDIO/CANX_E	N8	69	52			该引脚默认接 VDDIO，用户需要启动片内 CAN_Driver 时，将该引脚接地。	
		GPIOE 或中断信号					
INT1	D9	149	106	I/O/Z	-	GPIO 或者 XINT1 或者 XBIO 输入	
INT2	D8	151	108	I/O/Z	-	GPIO 或者 XINT2 或者 ADC 转换开始	
NMI	E8	150	107	I/O/Z	PU	GPIO 或者 XNMI 或者 XINT13	

表 1.3-1 信号说明⁽¹⁾ (续)

名称	引脚编号			I/O/Z ⁽²⁾	PU/PU ⁽³⁾	说明
	179 焊球 BGA 封装	176 引脚 QFP 封装	128 引脚 QFP 封装			
GPIOA 或 EVA 信号						
W1	M12	92	68	I/O/Z	PU	GPIO 或者 PWM 输出引脚 #1
W2	M14	93	69	I/O/Z	PU	GPIO 或者 PWM 输出引脚#2
W3	L12	94	70	I/O/Z	PU	GPIO 或者 PWM 输出引脚 #3
W4	L13	95	71	I/O/Z	PU	GPIO 或者 PWM 输出引脚 #4
W5	K11	98	72	I/O/Z	PU	GPIO 或者 PWM 输出引脚#5
W6	K14	101	75	I/O/Z	PU	GPIO 或者 PWM 输出引脚 #6
IOA6	J11	102	76	I/O/Z	PU	GPIO 或者定时器 1 输出
IOA7	J13	104	77	I/O/Z	PU	GPIO 或者定时器 2 输出
IOA8	H10	106	78	I/O/Z	PU	GPIO 或者捕捉输入 #1
IOA9	H11	107	79	I/O/Z	PU	GPIO 或者捕捉输入#2
IOA10	H12	109	80	I/O/Z	PU	GPIO 或者捕捉输入 #3
IOA11	F14	116	85	I/O/Z	PU	GPIO 或者定时器方向
IOA12	F13	117	86	I/O/Z	PU	GPIO 或者定时器时钟输入
IOA13	E13	122	89	I/O/Z	PU	GPIO 或者比较 1 输出触发
IOA14	E11	123	90	I/O/Z	PU	GPIO 或者比较 2 输出触发
IOA15	F10	124	91	I/O/Z	PU	GPIO 或者比较 3 输出触发
GPIOB 或 EVB 信号						
W7	N2	45	33	I/O/Z	PU	GPIO 或者 PWM 输出 Pin #7
W8	P2	46	34	I/O/Z	PU	GPIO 或者 PWM 输出 Pin #8
W9	N3	47	35	I/O/Z	PU	GPIO 或者 PWM 输出 Pin #9
W10	P3	48	36	I/O/Z	PU	GPIO 或者 PWM 输出 Pin #10
W11	L4	49	37	I/O/Z	PU	GPIO 或者 PWM 输出 Pin #11
W12	M4	50	38	I/O/Z	PU	GPIO 或者 PWM 输出 Pin #12
IOB6	K5	53	40	I/O/Z	PU	GPIO 或者定时器 3 输出
IOB7	N5	55	41	I/O/Z	PU	GPIO 或者定时器 4 输出
IOB8	M5	57	43	I/O/Z	PU	GPIO 或者捕捉输入 #4
IOB9	M6	59	44	I/O/Z	PU	GPIO 或者捕捉输入#5
IOB10	P6	60	45	I/O/Z	PU	GPIO 或者捕捉输入#6
IOB11	L8	71	54	I/O/Z	PU	GPIO 或者定时器方向
IOB12	K8	72	55	I/O/Z	PU	GPIO 或者定时器时钟输入
IOB13	N6	61	46	I/O/Z	PU	GPIO 或者比较 4 输出触发
IOB14	L6	62	47	I/O/Z	PU	GPIO 或者比较 5 输出触发
IOB15	K7	63	48	I/O/Z	PU	GPIO 或者比较 6 输出触发

表 1.3-1 信号说明⁽¹⁾ (续)

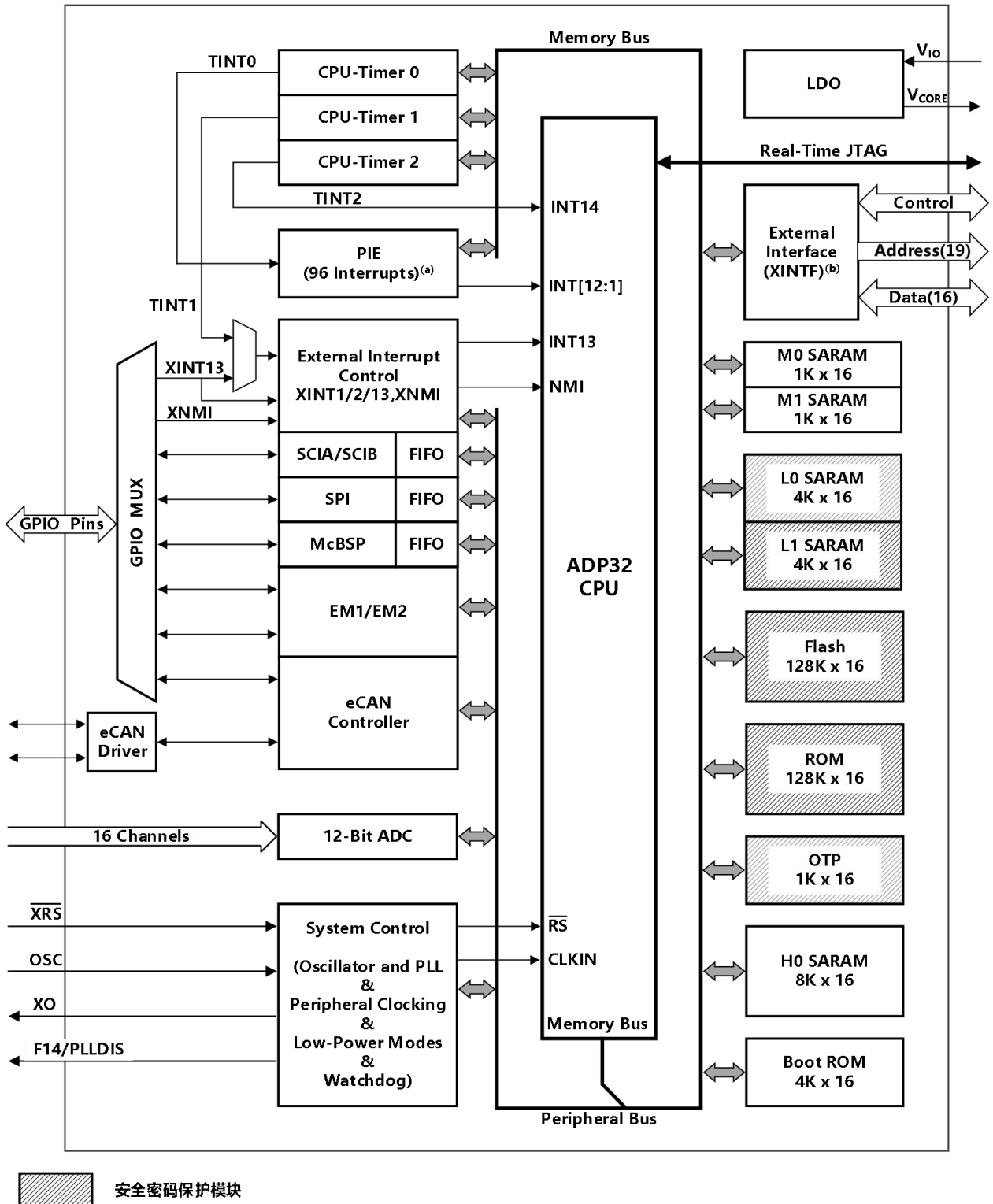
名称	引脚编号		128 引脚 QFP 封装	I/O/Z ⁽²⁾	PU/PU ⁽³⁾	说明
	179 焊球 BGA 封装	176 引脚 QFP 封装				
GPIOD 或 EM1 信号或 EM2 信号						
IOD0	H14	110	81	I/O/Z	PU	GPIO 或者定时器 1 比较输出触发
IOD1	G10	115	84	I/O/Z	PU	GPIO 或者定时器 2 比较输出触发或者外部 ADC 转换开始 EM1
IOD5	P10	79	60	I/O/Z	PU	GPIO 或者定时器 3 比较输出触发
IOD6	P11	83	61	I/O/Z	PU	GPIO 或者定时器 4 比较输出触发或者外部 ADC 转换开始 EM2
GPIOF 或 SPI 信号						
F0	M1	40	31	I/O/Z	-	GPIO 或者 SPI 从器件输入, 主器件输出
F1	N1	41	32	I/O/Z	-	GPIO 或者 SPI 从器件输出, 主器件输入
F2	K2	34	27	I/O/Z	-	GPIO 或者 SPI 时钟
F3	K4	35	28	I/O/Z	-	GPIO 或者 SPI 从器件发送使能
GPIOF 或 SCI-A 信号						
TXDA	C7	155	111	I/O/Z	PU	GPIO 或者 SCI 异步串行端口 TX 数据
RXDA	A7	157	112	I/O/Z	PU	GPIO 或者 SCI 异步串行端口 RX 数据
GPIOF 或 CAN 信号						
F6	N12	87	64	I/O/Z	PU	GPIO 或者 eCAN 发送数据
F7	N13	89	65	I/O/Z	PU	GPIO 或者 eCAN 接收数据
GPIOF 或 McBSP 信号						
F8	J1	28	23	I/O/Z	PU	GPIO 或者 McBSP 发送时钟
F9	H2	25	21	I/O/Z	PU	GPIO 或者 McBSP 接收时钟
F10	H4	26	22	I/O/Z	PU	GPIO 或者 McBSP 发送帧同步
F11	J2	29	24	I/O/Z	PU	GPIO 或者 McBSP 接收帧同步
F12	G1	22	19	I/O/Z	-	GPIO 或者 McBSP 被发送的串行数据
F13	G2	20	18	I/O/Z	PU	GPIO 或者 McBSP 接收到的串行数据
GPIOF 或者 XF CPU 输出信号						
F14/PLLDIS	A11	140	101	I/O/Z	PU	这个引脚有三个功能：1.XF-通用输出引脚。 2.XPLLDIS - 这个引脚在复位时被采样以检查 PLL 是否必须被禁用。如果这个引脚被检测到为 低电平, PLL 将被禁用。当 PLL 被禁用时, HALT 和 STANDBY 模式不能使用。3. GPIO- GPIO 功能
GPIOG 或者 SCI-B 信号						
G4	P14	90	66	I/O/Z	-	GPIO 或者 SCI 异步串行端口 TX 数据
G5	M13	91	67	I/O/Z	-	GPIO 或者 SCI 异步串行端口接收数据

注意：除电源端口外，对所有 I/O 端口的驱动必需是在电源上电完成后才开始。

1.4 引脚命名简称与全称对应表

引脚排序	引脚简称	引脚全称	引脚排序	引脚简称	引脚全称	引脚排序	引脚简称	引脚全称
1	VDDAIO	V _{DDAIO}	60	IOB10	CAP6_QEPI2	119	CLKOUT	XCLKOUT
2	INB0	ADCINB0	61	IOB13	C4TRIP	120	VSS	V _{SS}
3	INB1	ADCINB1	62	IOB14	C5TRIP	121	A[8]	XA[8]
4	INB2	ADCINB2	63	IOB15	C6TRIP	122	IOA13	C1TRIP
5	INB3	ADCINB3	64	VDDIO	V _{DDIO}	123	IOA14	C2TRIP
6	INB4	ADCINB4	65	D[8]	XD[8]	124	IOA15	C3TRIP
7	INB5	ADCINB5	66	VPP1_EN	TEST2	125	A[9]	XA[9]
8	INB6	ADCINB6	67	VPP0_EP	TEST1	126	TE	TMS
9	INB7	ADCINB7	68	D[9]	XD[9]	127	TDO	TDO
10	VREFM	ADCREFM	69	VDDIO/CANX_E	V _{DD3VFL}	128	VDD	V _{DD}
11	VREFP	ADCREFP	70	VSS	V _{SS}	129	VSS	V _{SS}
12	AVSSREF	AVSSREFBG	71	IOB11	TDIRB	130	A[10]	XA[10]
13	AVDDREF	AVDDREFBG	72	IOB12	TCLKINB	131	TDI	TDI
14	VDDA1	V _{DDA1}	73	D[10]	XD[10]	132	A[11]	XA[11]
15	VSSA1	V _{SSA1}	74	D[11]	XD[11]	133	ZCS6	XZCS6AND7
16	RESEXT	ADCRESEXT	75	VDD	V _{DD}	134	TSEL	TESTSEL
17	MC	XMP/IC	76	X0	X2	135	TRS	TRST
18	A[0]	XA[0]	77	OSC	X1/XCLKIN	136	TCK	TCK
19	VSS	V _{SS}	78	VSS_ALG	V _{SS}	137	MU0	EMU0
20	F13	MDRA	79	IOD5	T3TRIP/PDPINTB	138	A[12]	XA[12]
21	D[0]	XD[0]	80	A[2]	XA[2]	139	D[14]	XD[14]
22	F12	MDXA	81	VDDIO	V _{DDIO}	140	F14/PLLDIS	XF_XPLLDIS
23	VDD	V _{DD}	82	HOLDA	XHOLDA	141	A[13]	XA[13]
24	D[1]	XD[1]	83	IOD6	T4TRIP/EVBSOC	142	VSS	V _{SS}
25	F9	MCLKRA	84	WE	XWE	143	VDD	V _{DD}
26	F10	MFSXA	85	A[3]	XA[3]	144	A[14]	XA[14]
27	D[2]	XD[2]	86	VSS	V _{SS}	145	VDDIO	V _{DDIO}
28	F8	MCLKXA	87	F6	CANTXA	146	MU1	EMU1
29	F11	MFSRA	88	ZCS2	XZCS2	147	D[15]	XD[15]
30	D[3]	XD[3]	89	F7	CANRXA	148	A[15]	XA[15]
31	VDDIO	V _{DDIO}	90	G4	SCITXDB	149	INT1	XINT1_XBIO
32	VSS	V _{SS}	91	G5	SCIRXDB	150	NMI	XNMI_XINT13
33	D[4]	XD[4]	92	W1	PWM1	151	INT2	XINT2_ADCSOC
34	F2	SPICLKA	93	W2	PWM2	152	A[16]	XA[16]
35	F3	SPISTEPA	94	W3	PWM3	153	VSS	V _{SS}
36	D[5]	XD[5]	95	W4	PWM4	154	VDD	V _{DD}
37	VDD	V _{DD}	96	D[12]	XD[12]	155	TXDA	SCITXDA
38	VSS	V _{SS}	97	D[13]	XD[13]	156	A[17]	XA[17]
39	D[6]	XD[6]	98	W5	PWM5	157	RXDA	SCIRXDA
40	F0	SPISIMOA	99	VSS	V _{SS}	158	A[18]	XA[18]
41	F1	SPISOMIA	100	VDD	V _{DD}	159	HOLD	XHOLD
42	RD	XRD	101	W6	PWM6	160	RESET	XRS
43	A[1]	XA[1]	102	IOA6	T1PWM_T1CMP	161	READY	XREADY
44	ZCS	XZCS0AND1	103	A[4]	XA[4]	162	VDD1	V _{DD1}
45	W7	PWM7	104	IOA7	T2PWM_T2CMP	163	VSS1	V _{SS1}
46	W8	PWM8	105	VSS	V _{SS}	164	BGIN	ADCBGREFIN
47	W9	PWM9	106	IOA8	CAP1_QEP1	165	VSSA2	V _{SSA2}
48	W10	PWM10	107	IOA9	CAP2_QEP2	166	VDDA2	V _{DDA2}
49	W11	PWM11	108	A[5]	XA[5]	167	INA7	ADCINA7
50	W12	PWM12	109	IOA10	CAP3_QEP1	168	INA6	ADCINA6
51	R/W	XR/W	110	IOD0	T1TRIP_PDPINTA	169	INA5	ADCINA5
52	VSS	V _{SS}	111	A[6]	XA[6]	170	INA4	ADCINA4
53	IOB6	T3PWM_T3CMP	112	VDD	V _{DD}	171	INA3	ADCINA3
54	D[7]	XD[7]	113	LDOCT_VSS	V _{SS}	172	INA2	ADCINA2
55	IOB7	T4PWM_T4CMP	114	VDDIO	V _{DDIO}	173	INA1	ADCINA1
56	VDD	V _{DD}	115	IOD1	T2TRIP_EVASOC	174	INA0	ADCINA0
57	IOB8	CAP4_QEP3	116	IOA11	TDIR	175	LO	ADCLO
58	VSS	V _{SS}	117	IOA12	TCLKINA	176	VSSAIO	V _{SSAIO}
59	IOB9	CAP5_QEP4	118	A[7]	XA[7]			

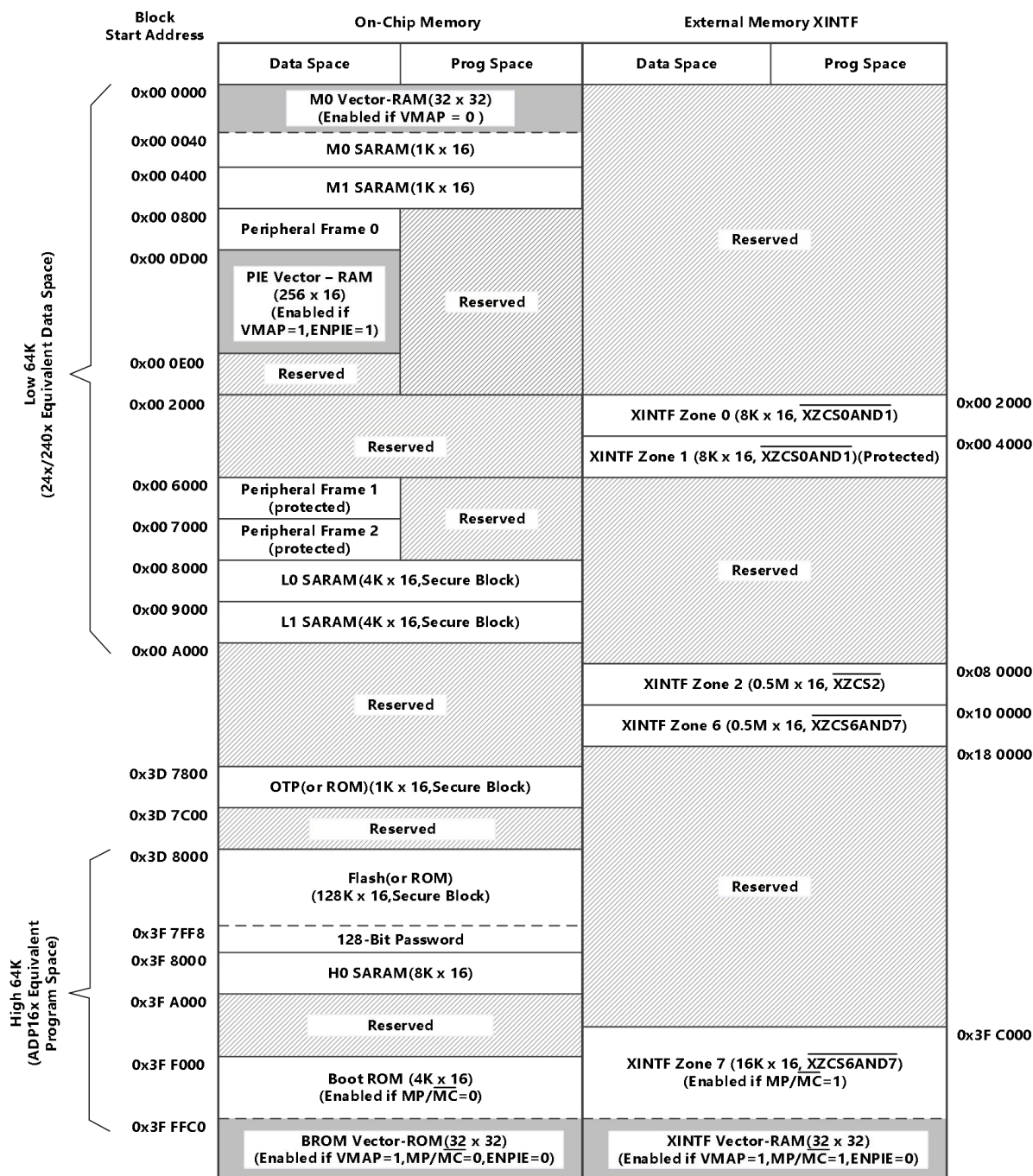
2 功能概述



- a. 96 个中断中 45 个中断已由外设使用
- b. XINTF 只在 ADP32F12 和 ADP32C12 器件上提供

图 2-1 功能方框图

2.1 内存映射

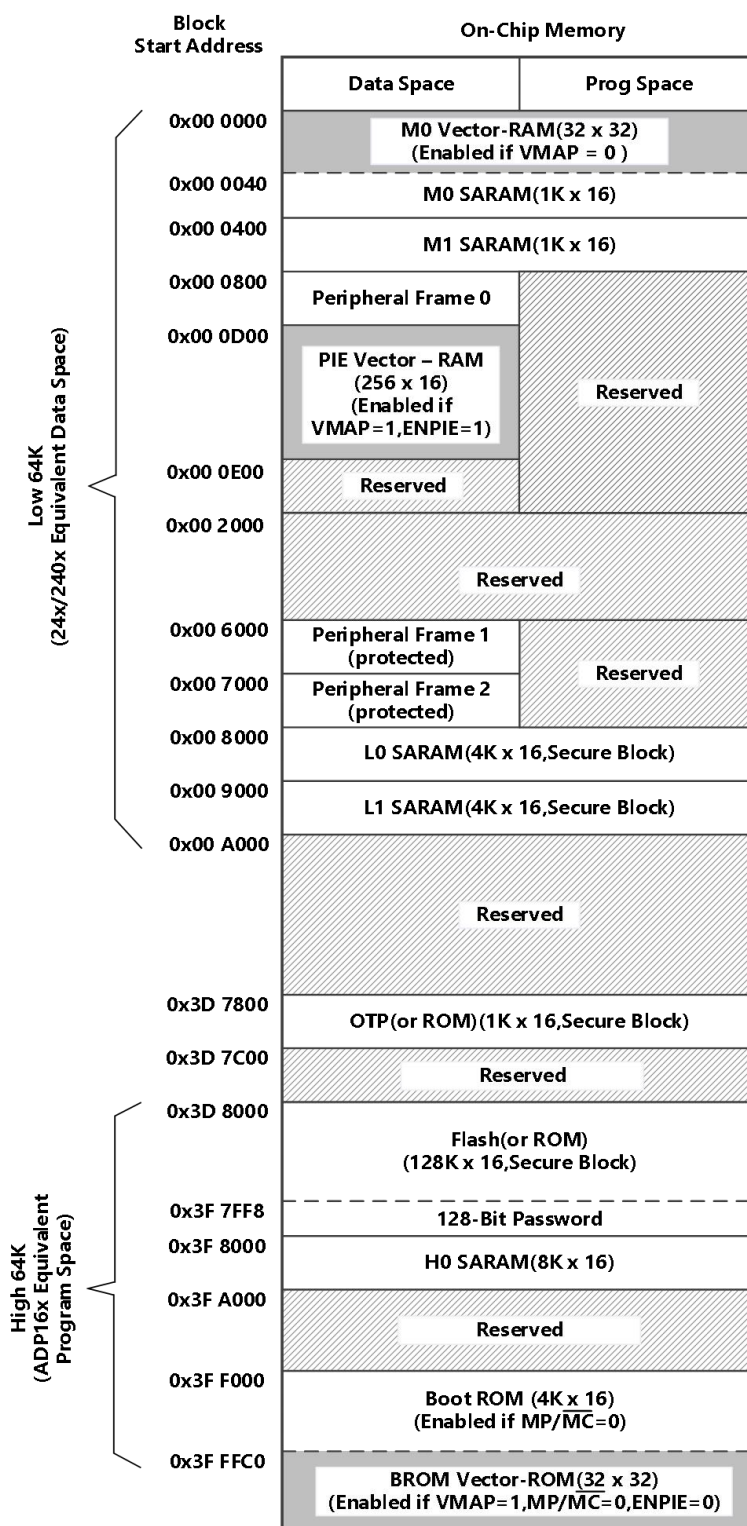


说明：在这些矢量图中每次只有M0向量，PIE向量，BROW向量，XINTF向量应该被一次性激活。

- 所有超出规定地址范围的存储区域不可使用。
- 被保留的单元用于未来的扩展。应用不应访问这些区域。
- 引导 ROM 和区域 7 内存映射可在芯片内或者 XINTF 上被激活，这取决于 \overline{MC} ，但是不能同时激活。
- 外设帧 0，外设帧 1，和外设帧 2 内存映射只限于数据内存。一个用户程序不能访问这些处于程序空间内的内存映射。
- “受保护”意味着写后读操作的顺序被保存，而不是流水线顺序。
- 特定内存区域受 EALLOW 保护以防止配置之后的假写入。
- 区域 0 和区域 6 以及区域 7 公用一样的芯片选择；因此，这些内存块有镜像单元。

图 2.1-1 ADP32F12 内存映射

ADP32FXX Digital Signal Processor

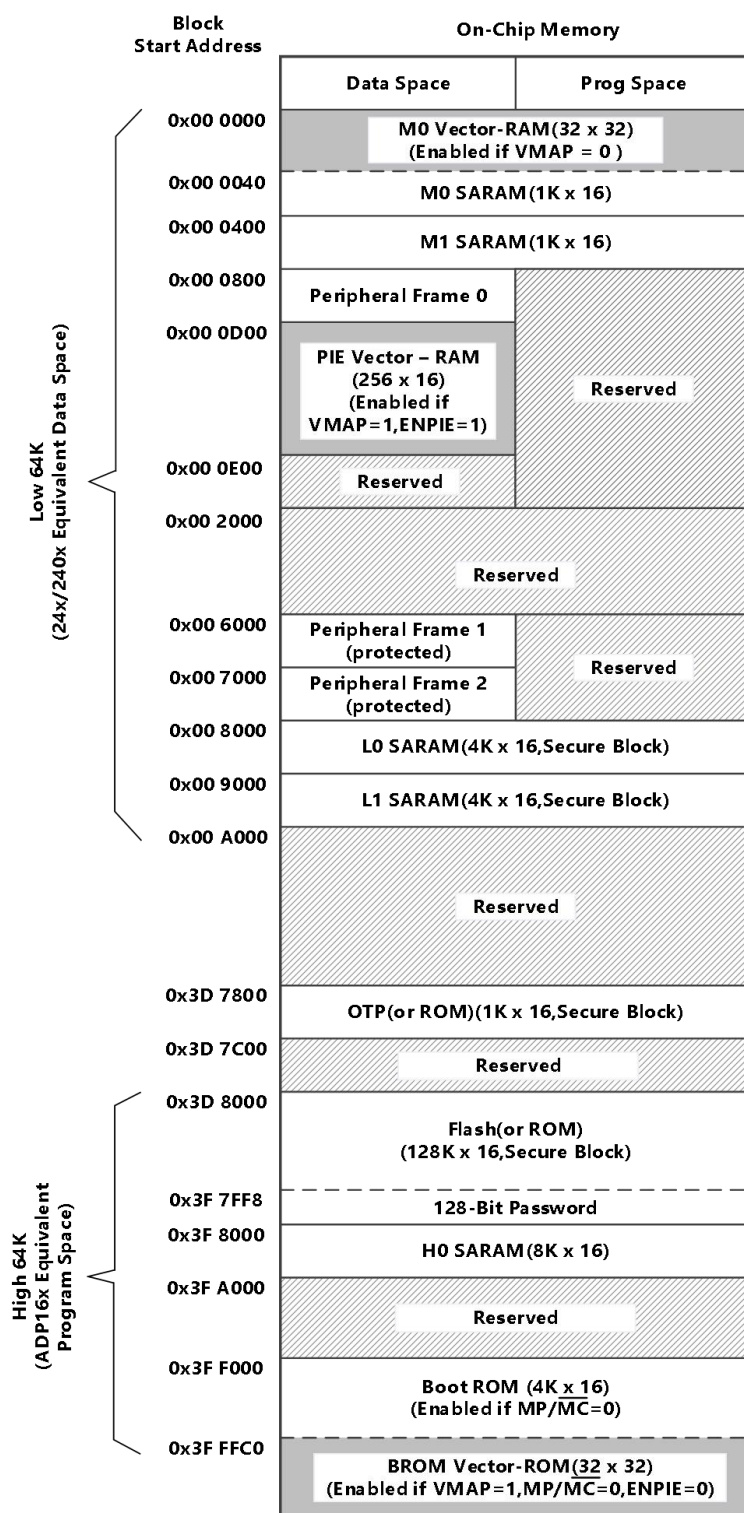


说明：在这些矢量图中每次只有M0向量，PIE向量，BROW向量，XINTF向量应该被一次性激活。

- A. 所有超出规定地址范围的存储区域不可使用。
- B. 留的单元用于未来的扩展。应用不应访问这些区域。
- C. 帧 0，外设帧 1，和外设帧 2 内存映射只限于数据内存，一个用户程序不能访问这些处于程序空间内的内存映射。
- D. 受保护”意味着写后读操作的顺序被保存，而不是流水线顺序。
- E. 定内存区域受 EALLOW 保护以防止配置之后的假写入。

图 2.1-2 ADP32F11 内存映射

ADP32FFX Digital Signal Processor



说明：在这些矢量图中每次只有M0向量，PIE向量，BROW向量，XINTF向量应该被一次性激活。

- 所有超出规定地址范围的存储区域不可使用。
- 被保留的单元用于未来的扩展。应用不应访问这些区域。
- 外设帧 0，外设帧 1，和外设帧 2 内存映射只限于数据内存，一个用户程序不能访问这些处于程序空间内的内存映射。
- “受保护”意味着写后读操作的顺序被保存，而不是流水线顺序。
- 特定内存区域受 EALLOW 保护以防止配置之后的假写入。

图 2.1-3 ADP32F10 内存映射

表 2-1.ADP32F12 和 ADP32F11 中的内存扇区地址

地址范围	程序和数据空间
0x3D8000	扇区 J, 8K x 16
0x3D9FFF	
0x3D A000	扇区 I, 8K x 16
0x3D BFFF	
0x3D C000	扇区 H, 16K x 16
0x3D FFFF	
0x3E 0000	扇区 G, 16K x 16
0x3E 3FFF	
0x3E 4000	扇区 F, 16K x 16
0x3E 7FFF	
0x3E 8000	扇区 E, 16K x 16
0x3E BFFF	
0x3E C000	扇区 D, 16K x 16
0x3E FFFF	
0x3F 0000	扇区 C, 16K x 16
0x3F 3FFF	
0x3F 4000	扇区 B, 8K x 16
0x3F 5FFF	
0x3F 6000	扇区 A, 8K x 16
0x3F 7F80	
0x3F 7FF5	当使用代码安全模块时, 编程至 0x0000。 引导至闪存 (或者 ROM) 进入点 (这里为程序分支指令) 安全密码 (128 位) (不要设定为全零)
0x3F 7FF6	
0x3F 7FF7	
0x3F 7FF8	
0x3F 7FFF	

表 2-2.ADP32F10 中内存扇区的地址

地址范围	程序和数据空间
0x3E 8000	扇区 E, 16K x 16
0x3E BFFF	
0x3E C000	扇区 D, 16K x 16
0x3E FFFF	
0x3F 0000	扇区 C, 16K x 16
0x3F 3FFF	
0x3F 4000	扇区 B, 8K x 16
0x3F 5FFF	
0x3F 6000	扇区 A, 8K x 16
0x3F 7F80	
0x3F 7FF5	当使用代码安全模块时, 编程至 0x0000。引导至闪存 (或者 ROM) 进入点 (这里为程序分支指令)
0x3F 7FF6	
0x3F 7FF7	
0x3F 7FF8	
0x3F 7FFF	
	安全密码 (128 位) (不要设定为全零)

内存地址范围的“低 64K”映射进 ADP16X 的数据空间。内存地址范围的“高 64K”映射进 ADP16X 的程序空间。ADP16X 兼容代码将只从内存区域的“高 64K”内执行。因此, 内存/ROM 和 H0 SARAM 块的前 32K 可被用于运行 ADP16X 兼容代码 (如果 MP/ \overline{MC} 模式为低电平) 或者, 在 ADP32FXX 上, 代码可以从 XINTF 区域 7 中执行 (如果 MP/ \overline{MC} 模式为高电平的话)。

ADP32FXX Digital Signal Processor

XINTF 由 5 个独立的区域组成。一个区域有其自己的芯片选择，剩余的四个区域有两个共用的芯片选择。每个区域可使用其自身的时序（等待状态）进行编程，并可被设定为采样或者忽略外部就绪信号。这样可实现与外设的简便且无缝对接。

注

XINTF 区域 0 和区域 1 的芯片选择被合并成一个单一的芯片选择(ZCS01)；并且 XINTF 区域 6 和区域 7 的芯片选择被合并成一个单一的芯片选择(ZCS67)。

外设帧 1，外设帧 2，以及 XINTF 区域 1 被编成一组以这些块成为“受保护的写入/读取外设块”。

“受保护”模式确保对这些所有的访问与文档中描述的一致。由于 ADP32FXX 的流水线，在对不同内存位置读取之前的写入操作将出现在 CPU 内存总线上相反的顺序。这会导致特定外设应用中的问题，在此类应用中，用户认为写入会首先发生（如文档所描述的那样）。ADP32FXX CPU 支持一个块保护模式，在这个模式中，可对一个内存区域进行保护，以确保操作按照本文档所描述的那样发生（代价增加了额外周期以校正运行）。可对这个模式进行编程，并且，缺省情况下，它将保护所选的区域。

在 ADP32FXX 上，复位时，如果 \overline{MC} 引脚被拉至高电平，XINTF 区域 7 被访问。这个信号选择微处理器或者微计算机模式运行。在微处理器模式中，区域 7 被映射到高位内存，这样从外部抽取矢量表。在这个模式中，引导 ROM 被禁用。在微计算机模式中，区域 7 被禁用，这样矢量取自引导 ROM。这使得用户能够从片载内存或者芯片外内存引导。 \overline{MC} 信号在复位时的状态被存储在 XINTCNF2 寄存器中的 \overline{MC} 模式位中。用户能够在软件中改变这个模式，并因此控制引导 ROM 和 XINTF 区域 7 的映射。其它的内存块不会受到 \overline{MC} 的影响。

ADP32FXX XINTF 上不支持 I/O 空间。

表 2-3 中列出了在内存映射区域内不同空间的等待状态。

表 2-3.等待状态

区域 (AREA)	等待状态	备注
M0 和 M1 SARAM	0-等待	固定的
外设帧 0	0-等待	固定的
外设帧 1	0-等待(写入) 2-等待 (读取)	固定的
外设帧 2	0-等待(写入) 2-等待 (读取)	固定的
L0 和 L1 SARAMs	0-等待	固定的
OTP(或 ROM)	可编程， 1-等待最小	由闪存寄存器设定。可在一个减少的 CPU 频率上执行 1 等待状态操作。
闪存 (或 ROM)	可编程， 1-等待最小	由闪存寄存器设定。可在一个减少的 CPU 频率上执行 0 等待状态操作。针对 16 个等待状态，CSM 密码位置为实线连接。
H0 SARAM	0-等待	固定的
引导-ROM	1-等待	固定的
XINTF	可编程， 1-等待最小	由 XINTF 寄存器设定。周期可由外部内存或者外设扩展。无法等待 0 等待操作。

2.2 简要说明

2.2.1 ADP32F1X CPU

此ADP32FXX DSP系列是进芯电子定点DSP平台上的最新产品。ADP32FXX与ADP16器件源代码向下兼容，ADP16应用于客户对价格敏感，性能要求较低的应用场合。此外，ADP32FXX是一款非常高效的C/C++引擎，此引擎不但使用户能够用高级语言开发他们的控制系统软件，还能够使用C/C++开发数学算法。ADP32FXX在处理DSP算术任务时与处理系统控制任务时同样有效，而系统控制任务通常由微控制器器件处理。这样的效率在很多系统中省却了对第二个处理器的需要。ADP32FXX的32x32位MAC功能和它的64位处理能力，使得ADP32FXX能够有效处理较高数字分辨率问题，否则的话，这些问题将需要一个更加昂贵的浮点处理器解决方案。添加了带有关键寄存器自动环境保存的快速中断响应，使得器件能够用最小的延迟处理很多异步事件。ADP32FXX有一个具有流水线式存储器访问的8级深度受保护流水线。这个流水线式操作使得ADP32FXX能够高速执行而无需求助于昂贵的高速存储器。特别分支超前硬件大大减少了条件不连续而带来的延迟。特别存储条件操作进一步提升了性能。

2.2.2 内存总线（哈弗总线架构）

与很多DSP类型器件一样，多总线被用于在内存和外设以及CPU之间移动数据。ADP32FXX内存总线架构包含一个程序读取总线、数据读取总线和数据写入总线。此程序读取总线由22条地址线路和32条数据线路组成。数据读取和写入总线由32条地址线路和32条数据线路组成。32位宽数据总线可实现单周期32位运行。多总线结构，通常称为“哈弗总线”，使得ADP32FXX能够在单周期内取一个指令、读取一个数据值和写入一个数据值。所有连接在内存总线上的外设和内存对内存访问进行优先级设定。总的来说，内存总线访问的优先级可概括如下：

最高级：数据写入（内存总线上不能同时进行数据和程序写入。）

 程序写入（内存总线上不能同时进行数据和程序写入。）

 数据读取

 程序读取（内存总线上不能同时进行程序读取和取指令。）

最低级：取指令（内存总线上不能同时进行程序读取和取指令。）

2.2.3 外设总线

ADP32FXX采用了一个针对外设互连的外设总线标准。外设总线桥复用了多种总线，此总线将处理器“内存总线”组装进一个由16条地址线路和16条或者32条数据线路和相关控制信号组成的单总线中。在ADP32FXX上支持两个版本的外设总线。一个版本只支持16位访问（被称为外设帧2）并且这个版本保留了与ADP16外设的兼容性。另外一个版本支持16位和32位访问（被称为外设帧1）。

2.2.4 实时 JTAG 和分析

ADP32FXX 执行标准 IEEE1149.1UJTAG 接口。此外，ADP32FXX 支持实时运行模式，在处理器正在运行，执行代码并且处理中断时，可修改存储器内容、外设、和寄存器位置。用户也可以通过非时间关键代码进行单步操作，同时可在没有干扰的情况下启用即将被处理的时间关键中断。ADP32FXX 在 CPU 的硬件内执行实时模式。这是 ADP32FXX 所特有的功能，无需软件监控。此外，还提供了特别分析硬件，以使用户能够设定硬件断电或者数据/地址观察点并当一个匹配发生时生成不同的用户可选中断事件。

2.2.5 外部接口 (XINTF) (只适用于 ADP32F12)

这个异步接口由 19 条地址线路，16 条数据线路，和 3 个芯片选族线路组成。此芯片选族线路被映射到 5 个外部区域，即区域 0，1，2，6，和 7。区域 0 和 1 共用一个单一芯片选择线路；区域 6 和 7 也共用一个单一芯片选择线路。5 个区域中的每个区域可被设定为不同的等待状态数量、选通信号设置和保持时序，并且每个区域可被外部设定为扩展等待状态或者没有扩展等待状态。可编程等待状态、芯片选择和可编程选通时序可实现到外部存储器和外设的无缝对接。

2.2.6 闪存

ADP32F12、ADP32F11 和 ADP32F10 包含 128Kx16 的嵌入式闪存存储器，这些存储器被分成四个 8Kx16 扇区，和六个 16Kx16 扇区。所有三个器件还包含一个单一 1Kx16 的 OTP 内存，其地址范围为 0x3D 7800-0x3D 7BFF。用户能够在不改变其它扇区的同时单独擦除、编辑、和验证一个闪存扇区。然而，不能使用闪存的一个扇区或者这个 OTP 来执行擦除/编辑其它扇区的内存算法。提供了特殊内存流水线操作以使闪存模块实现更高性能。内存/OTP 被映射到程序和数据空间；因此，它可被用于执行代码或者存储数据信息。

注

ADP32F10/F11/F12 内存和 OTP 等待状态可由应用配置。这使得运行在较低频率上的应用能够将闪存配置为使用较少的等待状态。可通过在闪存选项寄存器中启用闪存流水线操作模式来提升闪存的效能。这个模式被启用时，线性代码执行的效能将远远快于只由等待状态配置所表示的原始效能。使用闪存流水线模式的准确性能增加依应用而定。

2.2.7 ROM

客户定制 C 版本，ADP32C12 包含 128Kx16 的 ROM。ADP32C10 有 64Kx16 的 ROM。除此之外，在闪存器件中提供一个取代 OTP 内存的 1Kx16ROM 块。

2.2.8 M0 , M1 SARAM

所有 ADP32FXX 器件包含这两块单周期访问内存，每一个的大小为 1Kx16。复位时，堆栈指针指向块 M1 的开始位置。M0 与 ADP16X 器件 B0, B1, B2 RAM 重叠，因此 ADP16X 器件上数据变量的映射能够保存在 ADP32FXX 器件上的同一个物理地址。M0 和 M1 块，与所有其他 ADP32FXX 器件上的内存块一样，被映射到程序和数据空间。因此，用户能够使用 M0 和 M1 来执行代码或者用于数据变量。分区在连接器内执行。ADP32FXX 器件提供了一个到编程器的统一内存映射。这使得用高级语言编程变得更加容易。

2.2.9 L0 , L1 , H0 SARAM

ADP32FXX 包含一个附加的 16Kx16 单周期访问 RAM，此 RAM 被分成三个块（4K+4K+8K）。由于每个块可被独立访问，因此大大降低了流水线延迟。每个块被映射到程序和数据空间。

2.2.10 引导 ROM

引导 ROM 由厂家使用引导载入软件进行设定。引导 ROM 程序在器件复位并检查了几个 GPIO 引脚后执行以确定应该进入哪一个引导模式。例如，用户可以选择执行已经出现在内部闪存中的代码或者通过几个串行端口中的一个将全新的软件下载至内部 RAM。还有其它的引导模式。引导 ROM 还包含用于数学相关算法中的标准表，例如 SIN/COS 波形。表 2-4 显示了如何调用不同引导模式的细节。

表 2-4. 引导模式选^{(1) (2)}

已选择的引导模式	TXDA	F12	F3	F2
GPIO PU 状态 ⁽³⁾	PU	无 PU	无 PU	无 PU
转跳至闪存/ROM 地址 0x3F7FF6。 在复位至所需的重定向代码执行之前，必须在这里编辑一个分支指令。	1	X	X	X
调用 SPI_Boot 来从一个外部串行 SPI EEPROM 中加载	0	1	X	x
调用 SCI_Boot 来从 SCI_A 端口加载	0	0	1	1
跳转至 H0 SARAM 地址 0x3F8000	0	0	1	0
跳转至 OTP 地址 0x3D7800	0	0	0	1
调用 Parallel_Boot 来从 GPIO 端口 B 中载入	0	0	0	0

(1) 由于切换 SPICLK 来选择一个引导模式会对外部逻辑产生影响，所以应该格外小心。

(2) 如果选择的引导模式为闪存、H0、或者 OTP，那么引导载入程序不会载入外部代码。

(3) PU=引脚有一个内部上拉电阻。无 PU=引脚无内部上拉电阻。

2.2.11 安全性

ADP32FXX 支持高级别安全以保护用户固定不受逆向工程损坏。这个安全性特有一个 128 位密码（针对 16 个等待状态的硬编码），此密码由用户编辑输入闪存。一个代码安全模块（CSM）被用于保护闪存/ROM/OTP 和 L0/L1 SARAM 块。这个安全特性防止未经授权的用户通过 JTAG 端口检查内存内容，从外部内存执行代码或者试图引导加载一些将会输出安全内存内容的恶意软件。为了启用到安全块的访问，用户必须写入与存储在闪存/ROM 密码位置内的值相匹配的正确的 128 位“KEY（密钥）”值。

注

- 当代码安全密钥被编辑时，0x3F7F80 到 0x3F7FF5 间的所有地址不能被用作程序代码或者数据。这些位置必须被设定为 0x0000
- 如果代码安全特性未被使用，地址 0x3F7F80 至 0x3F7FEF 可被用于代码或者数据。
- 在 ROM 器件上，无论代码安全性被使用与否，地址 0x3F7FF2-0x3F7FF5 和 0x3D7BFC-0x3D7BFF 为原厂保留。用户应用无论如何不应使用这些位置。
- 128 位密码（位于 0x3F7FF8-0x3F7FFF）不能写入全零。一旦写入全零将永久锁住此器件，请务必慎重。

表 2-5.使用安全代码模块的影响

地址	代码安全状态	
	代码安全被启用	代码安全被禁用
0x3F7F80-0x3F7FEF	用 0x0000 填充	应用代码和数据
0x3F7FF0-0x3F7FF5		
0x3D7BFC-0x3D7BFF	应用代码和数据	

代码安全模块免责声明

这个器件所包含的代码安全模块（CSM）被设计用于对存储在相关内存（ROM 或者闪存）中的数据
进行密码保护并且由原厂提供质量保证，与其标准条款和条件相一致，符合原厂发布的规范以获得适用于这个器件的保修期。

但是，原厂不保证或表示 CSM 不会被损坏或破坏，或不能通过其它方法存取关联的存储器中存储的数据。而且，除了上述内容外，原厂也未对本器件的 CSM 或操作做任何保证或表示，包括任何隐含的用于特定用途的商用性或适用性保证。

在任何情况下，原厂对以任何方法使用 CSM 或本器件产生的任何必然、特殊、间接、偶然或严重伤害不负任何责任，无论原厂是否被告知存在这种伤害的可能性。排除的损害包括但不限于数据丢失、信誉损失、无法使用、业务中断或其它经济损失。

2.2.12 外设中断扩展(PIE)块

PIE 块将许多中断源复用到中断输入的较小的集合中。PIE 块能够支持多达 96 个外设中断。在 ADP32FXX 上，外设使用 96 个可能中断中的 45 个。96 个中断被分成 8 块，并且每组被馈入 12 个 CPU 中断线路（INT1 至 INT12）中的 1 个。96 个中断中的每一个中断由其存储在一个可被用户写覆盖的专用 RAM 块中的矢量支持。在处理这个中断时，这个矢量由 CPU 自动抽取。抽取这个矢量以及保存关键 CPU 寄存器将花费 8 个 CPU 时钟周期。因此 CPU 能够对中断事件做出快速响应。可以通过硬件和软件控制中断的优先级。每个中断都可以在 PIE 块内启用/禁用。

2.2.13 外部中断(INT1, INT2, INT13, NMI)

ADP32FXX支持三个屏蔽的外部中断(INT 1, 2, 13)。INT13 与一个非屏蔽外部中断(NMI) 组合在一起。组合的信号名称为NMI_INT13。这些中断中的每一个可被选择用于负边沿或正边沿触发, 并且可被启用或禁用(包括NMI 在内)。这些屏蔽的中断还包含一个16 位自运行上数计数器, 当一个有效中断沿被检测到的时候, 此计数器被重置为0。这个计数器可被用于为中断精确计时。

2.2.14 振荡器和锁相环(PLL)

ADP32FXX 的时钟可由一个外部振荡器或者由一个连接到片上振荡器电路的晶振计时或者片上 OSC_2 产生。经过的一个 PLL 支持高达 10 个输入时钟缩放比。PLL 比率可用软件中在器件运行时更改, 这使得用户在需要低功耗运行时能够按比例降低运行频率。时序细节, 请参考 Section 5, 电气规范。PLL 块可被设定为旁通模式。

2.2.15 看门狗定时器

ADP32FXX 支持一个看门狗定时器。用户软件必须在一个特定的时间范围内定期复位 CPU 看门狗计数器; 否则, CPU 看门狗将生产一个到处理器的复位。如果需要, 可禁用看门狗定时器。

2.2.16 外设时钟

在外设闲置时, 到每一个独立外设的时钟可被启用/禁用以减少功耗。此外, 到串行端口(除了 eCAN) 和事件管理器、CAP 和 QEP 块的系统时钟可相对于 CPU 时钟进行缩放。这样可去除外设时序到逐渐增加的 CPU 时钟速度的耦合。

2.2.17 低功耗模式

ADP32FXX 器件是完全静态 CMOS 器件。提供三个低功耗模式:

IDLE: 将 CPU 置于低功耗模式。可有选择性地关闭外设时钟并且只有那些在 IDLE 期间必须运行的外设保持运行状态。来自激活外设的已启用的中断将把处理器从 IDLE 模式中唤醒。

STANDBY: 关闭到 CPU 和外设的时钟。在这个模式下, 振荡器和 PLL 仍然运行。一个外部中断事件将唤醒处理器和外设。在检测到中断事件之后的下一个有效周期上, 执行开始。

HALT: 关断内部晶振电路与片上 OSC 与 OSC_2, 该模式会将器件置于尽可能低的功耗模式中。只有复位或者 XNMI 才能将器件从这个模式中唤醒。

2.2.18 外设帧 0,1,2 (PFn)

ADP32FXX 将外设分为三个部分。外设映射如下：

PF0 : XINTF : 外部接口配置寄存器 (只适用于 ADP32FXX)

PIE : PIE 中断启用和控制寄存器加上 PIE 矢量表

闪存 : 闪存控制、编程、擦除、验证寄存器

定时器 : CPU-定时器 0, 1, 2 寄存器

CSM : 代码安全模块 KEY 寄存器

PF1 : eCAN : eCAN 邮箱和控制寄存器

PF2 : SYS : 系统控制寄存器

GPIO : GPIO MUX 配置和控制寄存器

EV : 事件寄存器(EM1/EM2) 控制寄存器

McBSP : McBSP 控制和 TX/RX 寄存器

SCI : 串行通信接口(SCI)控制和 RX/TX 寄存器

SPI : 串行外设接口(SPI)控制和 RX/TX 寄存器

ADC : 12 位 ADC 寄存器

2.2.19 通用输入/输出(GPIO)复用器

大多数的外设信号与通用输入/输出(GPIO)信号复用。这个复用使得用户能够在外设信号或者功能不使用时将一个引脚用作 GPIO。复位时, 所有 GPIO 引脚被配置为输入。针对 GPIO 模式或者外设信号模式, 用户能够独立设定每一个引脚。对于特定的引脚, 用户还能够选择屏蔽预设时钟数后, 在合格周期内取值来滤除有害的噪声毛刺脉冲。

2.2.20 32 位 CPU 定时器(0, 1, 2)

CPU 定时器 0,1,和 2 是完全一样的 32 位定时器, 这些定时器带有可预先设定的周期和 16 位时钟预分频。此定时器有一个 32 位倒计时寄存器, 此寄存器在计数器达到 0 时生成一个中断。这个计数器的减量为被预分频值设置所分频的 CPU 时钟速度的值。当此计数器达到 0 时, 它自动重新载入一个 32 位的周期值。CPU 定时器 2 为 DSP/BIOS 实时 OS 所预留, 并且被连接至 CPU 的 INT14。如果 DSP/BIOS 未被使用, CPU 定时器 2 也可用于普通用途。CPU 定时器 1 为通用定时器并被连接至 CPU 的 INT13。CPU 定时器 0 也为通用定时器并被连接至 PIE 块。

2.2.21 控制外设

ADP32FXX 支持以下用于嵌入式控制和通信的外设：

EM : 事件管理器模块包括通用定时器、完全比较/ PWM 单元、捕捉输入(CAP)和正交编码脉冲(QEP)电路。提供的这两个事件管理器可实现驱动两个三相电机或者四个两项电机。ADP32FXX 上的事件管理器与 ADP16X 器件上的事件管理器兼容 (有一些小改进)。

ADC : ADC 块是一个 12 位、单端、16 通道转换器。它包含两个用于同步采样的采样保持单元。

2.2.22 串行端口外设

ADP32FXX 支持下列串行通信外设：

eCAN：这是 CAN 外设的增强型版本。它支持 32 个邮箱、消息时间戳、并与 CAN2.0B 兼容。

McBSP：多通道缓冲串行端口(McBSP)连接到 E1/T1 线路、语音质量编解码器以实现最新应用或者高质量立体声音频 DAC 器件。McBSP 接收和发送寄存器由一个 16 级 FIFO 支持，从而大大减少了处理这个外设所用的开销。

SPI：SPI 是一个高速、同步串行 I/O 端口，此端口可在设定的位传输速率上将一个设定长度（1 至 16 位）的串行比特流移入和移出器件。通常情况下，SPI 用于 DSP 控制器和外部外设或者其它处理器之间的通信。典型应用包括外部 I/O 或者从诸如移位寄存器、显示驱动器、和 ADC 等器件的外设扩展。多器件通信由 SPI 的主控/受控操作支持。在 ADP32FXX 上，此端口支持一个 16 级、接收发送 FIFO 来减少处理开销。

SCI：串行通信接口是一个两线制异步串行端口，通常被称为 UART。在 ADP32FXX 上，此端口支持一个 16 级、接收发送 FIFO 来减少处理开销。

2.3 寄存器映射

ADP32FXX 包含三个外设寄存器空间。这些空间分类如下：

外设帧 0：这些是直接映射到 CPU 内存总线的外设。请见表 2-6。

外设帧 1：这些是映射到 32 位外设总线的外设。请见表 2-7。

外设帧 2：这些是映射到 16 位外设总线的外设。请见表 2-8。

表 2-6. 外设帧 0 寄存器⁽¹⁾

名称	地址范围	大小 (x16)	访问类型 ⁽²⁾
器件仿真寄存器	0x00 0880-0x00 09FF	384	受 EALLOW 保护
被保留	0x00 0A00-0x00 0A7F	128	
闪存寄存器 ⁽³⁾	0x00 0A80-0x00 0ADF	96	受 EALLOW 保护 受 CSM 保护
代码安全模块寄存器	0x00 0AE0-0x00 0AEF	16	受 EALLOW 保护
被保留	0x00 0AF0-0x00 0B1F	48	
XINTF 寄存器	0x00 0B20-0x00 0B3F	32	不受 EALLOW 保护
被保留	0x00 0B40-0x00 0BFF	192	
CPU-定时器 0/1/2 寄存器	0x00 0C00-0x00 0C3F	64	不受 EALLOW 保护
被保留	0x00 0C40-0x00 0CDF	160	
PIE 寄存器	0x00 0CE0-0x00 0CFF	32	不受 EALLOW 保护
PIE 矢量表	0x00 0D00-0x00 0DFF	256	受 EALLOW 保护
被保留	0x00 0E00-0x00 0FFF	512	

(1)在帧 0 中的寄存器支持 16 位和 32 位访问。

(2)如果寄存器受 EALLOW 保护，在用户执行 EALLOW 指令之前，不能执行写入操作。EDIS 指令会禁用写入操作。这防止了杂散代码或者指针损坏寄存器内容。

(3)闪存寄存器也受到代码安全模块(CSM)的保护。

表 2-7. 外设帧 1 寄存器⁽¹⁾

ADP32FXX Digital Signal Processor

名称	地址范围	大小 (x16)	访问类型
eCAN 寄存器	0x00 6000-0x0060FF	256 (128x32)	某些 eCAN 控制寄存器 (以及在其它 eCAN 控制寄存器中选定的位) 受 EALLOW 保护。
eCAN 邮箱 RAM	0x00 6100-0x00 61FF	256 (128x32)	不受 EALLOW 保护
被保留	0x00 6200-0x00 6FFF	3584	

(1) eCAN 控制寄存器只支持 32 位读取/写入操作。所有 32 位存取与偶数地址边界对齐。

表 2-8. 外设帧 2 寄存器⁽¹⁾

名称	地址范围	大小 (x16)	访问类型
被保留	0x00 7000-0x00 700F	16	
系统控制寄存器	0x00 7010-0x00 702F	32	受 EALLOW 保护
被保留	0x00 7030-0x00 703F	16	
SPI-A 寄存器	0x00 7040-0x00 704F	16	不受 EALLOW 保护
SCI-A 寄存器	0x00 7050-0x00 705F	16	不受 EALLOW 保护
被保留	0x00 7060-0x00 706F	16	
外部中断寄存器	0x00 7070-0x00 707F	16	不受 EALLOW 保护
被保留	0x00 7080-0x00 70BF	64	
GPIO 复用寄存器	0x00 70C0-0x00 70DF	32	受 EALLOW 保护
GPIO 数据寄存器	0x00 70E0-0x00 70FF	32	不受 EALLOW 保护
ADC 寄存器	0x00 7100-0x00 711F	32	不受 EALLOW 保护
被保留	0x00 7120-0x00 73FF	736	
EM1 寄存器	0x00 7400-0x00 743F	64	不受 EALLOW 保护
被保留	0x00 7440-0x00 74FF	192	
EM2 寄存器	0x00 7500-0x00 753F	64	不受 EALLOW 保护
被保留	0x00 7540-0x00 774F	528	
SCI-B 寄存器	0x00 7750-0x00 775F	16	不受 EALLOW 保护
被保留	0x00 7760-0x00 77FF	160	
McBSP 寄存器	0x00 7800-0x00 783F	64	不受 EALLOW 保护
被保留	0x00 7840-0x00 7FFF	1984	

(1) 设帧 2 只允许 16 位访问。所有 32 位访问被忽略 (可能返回或写入无效数据)。

2.4 器件仿真寄存器

表 2-9. 器件仿真寄存器

名称	地址范围	大小 (x16)	说明
DEVICECNF	0x00 0880-0x00 0881	2	器件配置寄存器 部件 ID 寄存器 0x001 或者 0x0002
PARTID	0x00 0882	1	
REVID	0x00 0883	1	0x0001-芯片修订版本 A 0x0002-芯片修订版本 B 0x0003-芯片修订版本 C,D 修订版本 ID 寄存器 0x0004-被保留 0x0005-芯片修订版本 E
PROTSTART	0x00 0884	1	块保护起始地址寄存器
PROTRANGE	0x00 0885	1	块保护范围地址寄存器
被保留	0x00 0886-0x00 09FF	378	

这些寄存器用于控制 CPU 的保护模式和监视某些关键器件信号。表 2-9 中对这些寄存器进行了定义。

2.5 外部接口 , XINTF(只适用于 ADP32F12)

这一部分给出了在 ADP32FXX 器件上执行的外部接口 (XINTF) 的顶视图。

外部接口是一个非复用异步总线。ADP32FXX 上的外部接口被映射到图 2-5 中所示的 5 个固定区域中。

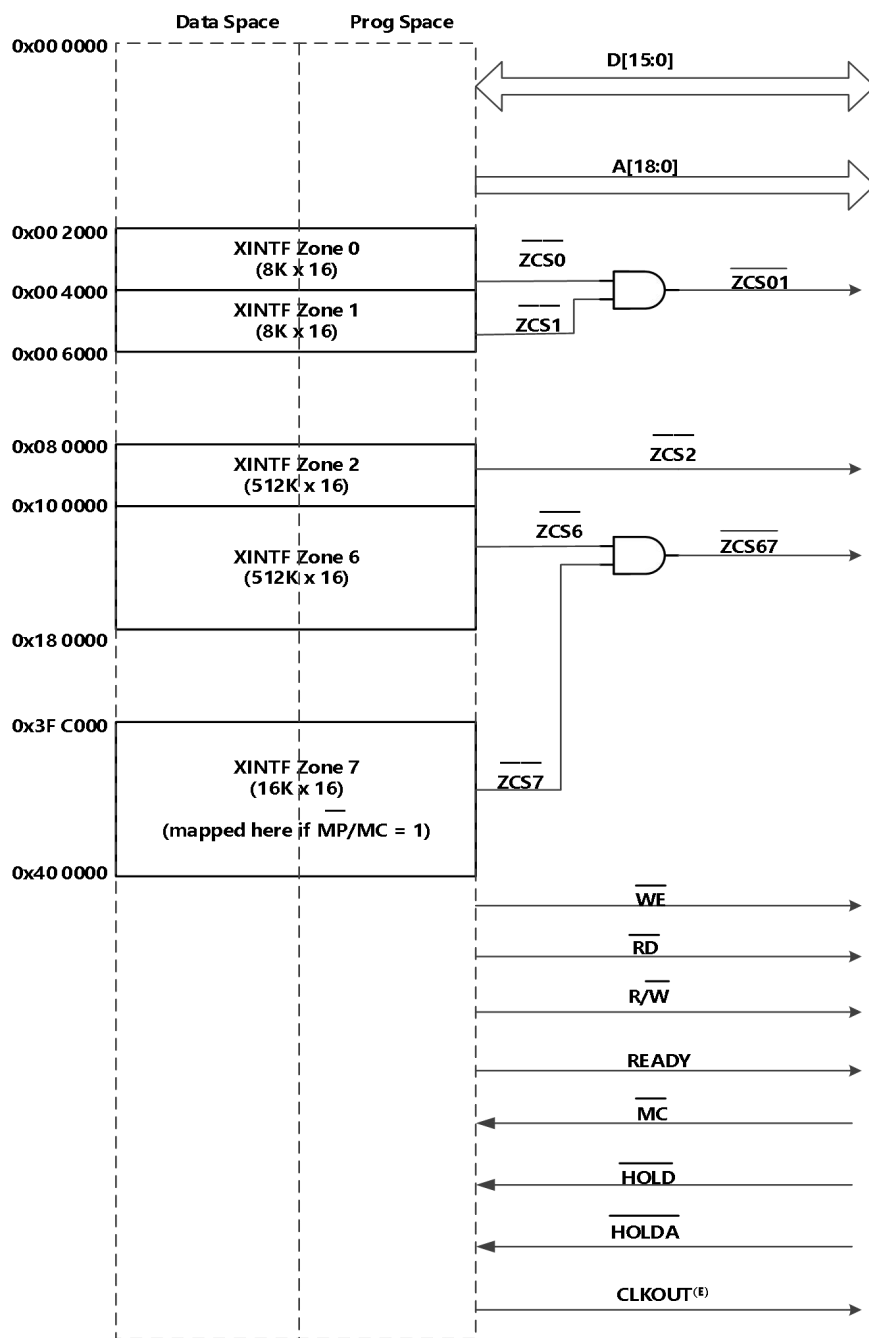


图 2-5.外部接口方框图

- A. XINTF 区域 7 的映射取决于 MC 器件输入信号和 MP/MC 模式位 (XINCNF2 寄存器的位 8)。区域 0, 1, 2, 和 6 一直被启用。
- B. 每个区域可被设定为具有不同的等待状态, 建立和保持时序, 并且由芯片选择 (\overline{ZS} , $\overline{ZCS2}$, $\overline{ZCS6}$) 支持, 当执行到一个特定区域的访问时切换。这些特性可实现到很多外部存储器和外设的无缝连接。
- C. 针对区域 0 和区域 1 的芯片选择内部组合在一起形成一个芯片 (\overline{ZCS})。被连接到 \overline{ZCS} 的任何外部存储器被双重映射到区域 0 和区域 1。
- D. 针对区域 6 和区域 7 的芯片选择内部组合在一起形成一个芯片选择 ($\overline{ZCS67}$)。被 $\overline{ZCS67}$ 的任何外部存储器被双重映射到区域 6 和区域 7。这意味着, 如果区域 7 被禁用 (通过 MP/MC 模式), 那么仍然可通过区域 6 地址空间来访问很多外部存储器。
- E. CLKOUT 也是 ADP32F10 和 ADP32F11 上的输出引脚。

外部接口的运行和时序, 可由表 2-10 中列出的寄存器控制。

表 2-10.XINTF 配置和控制寄存器映射

名称	地址	大小 (x16)	说明
XTIMING0	0x00 0B20	2	XINTF 时序寄存器, 区域 0 可作为两个 16 位寄存器或者一个 32 位寄存器进行访问
XTIMING1	0x00 0B22	2	XINTF 时序寄存器, 区域 1 可作为两个 16 位寄存器或者一个 32 位寄存器进行访问
XTIMING2	0x00 0B24	2	XINTF 时序寄存器, 区域 2 可作为两个 16 位寄存器或者一个 32 位寄存器进行访问
XTIMING6	0x00 0B2C	2	XINTF 时序寄存器, 区域 6 可作为两个 16 位寄存器或者一个 32 位寄存器进行访问
XTIMING7	0x00 0B2E	2	XINTF 时序寄存器, 区域 7 可作为两个 16 位寄存器或者一个 32 位寄存器进行访问
XINTCNF2	0x00 0B34	2	XINTF 配置寄存器可作为两个 16 位寄存器或者一个 32 位寄存器进行访问
XBANK	0x00 0B38	1	XINTF 组控制寄存器
XREVISION	0x00 0B3A	1	XINTF 修订版本寄存器

2.5.1 时序寄存器

XINTF 信号时序可被调节成特定外部器件要求相匹配, 诸如选通信号的建立和保持时间, 以实现竞争避免和最大总线效率。对于每个基于内存要求的区域或者特定区域访问的外设, XINTF 时序参数可被独立配置。这是的程序设计人员能够在预区域的基础上大大提升总线效率。所有 XINTF 时序值与 XTIMCLK 相对应, 如图 5-30 所示, 它等于 SYSCLKOUT, 或者为 SYSCLKOUT 的一半。

2.5.2 XREVISION 寄存器

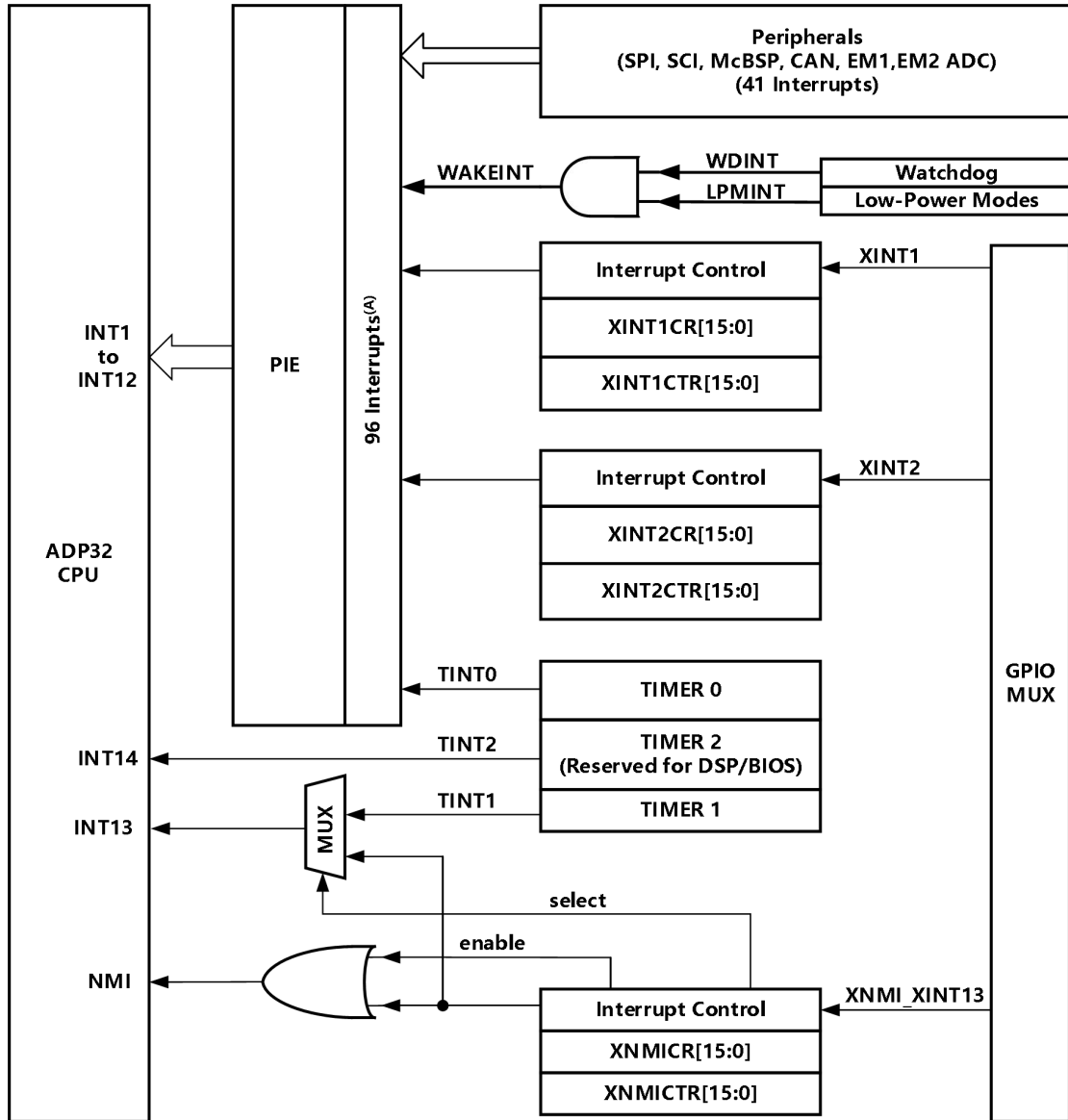
XREVISION 寄存器包含一个独特的数字用来识别产品中所使用的 XINTF 的特定版本。对于 ADP32FXX, 这个寄存器按照表 2-11 中的说明进行配置。

表 2-11.XREVISION 寄存器位定义

位	名称	类型	复位	说明
15-0	修订版本	读	0x0004	当前 XINTF 修订版本用于内部使用/参考。只用于测试用途。会有更改。

2.6 中断

图 2.6-1 显示了如何在 ADP32FXX 器件内复用不同的中断源



A. 在 96 个可能的中断中，目前有 45 个中断由外设使用

图 2.6-1 中断源

8 个 PIE 块中断被组合进一个 CPU 中断中。12 个 CPU 中断组，每个 8 个中断，相当于 96 个可能中断。在 ADP32FXX 上，这其中外设使用的 45 个中断显示在表 2-12 中。

TRAP#Vectormumber(矢量号) 指令将程序控制发送至与指定的矢量相对于用的中断处理列程。TRAP#0 尝试传送程序控制到复位矢量所指向的地址。然而，PIE 矢量表不含复位矢量。因此，当 PIE 被启用时，TRAP#0 不应被使用。这样做将导致未定义的运行状态。

当 PIE 被启用时，TRAP#1 至 TRAP#12 将传送程序控制到 PIE 组内第一个矢量相对应的中断处理列程。例如：TRAP#1 从 INT1.1 中抽取矢量，TRAP#2 从 INT2.1 中抽取矢量，以此类推。

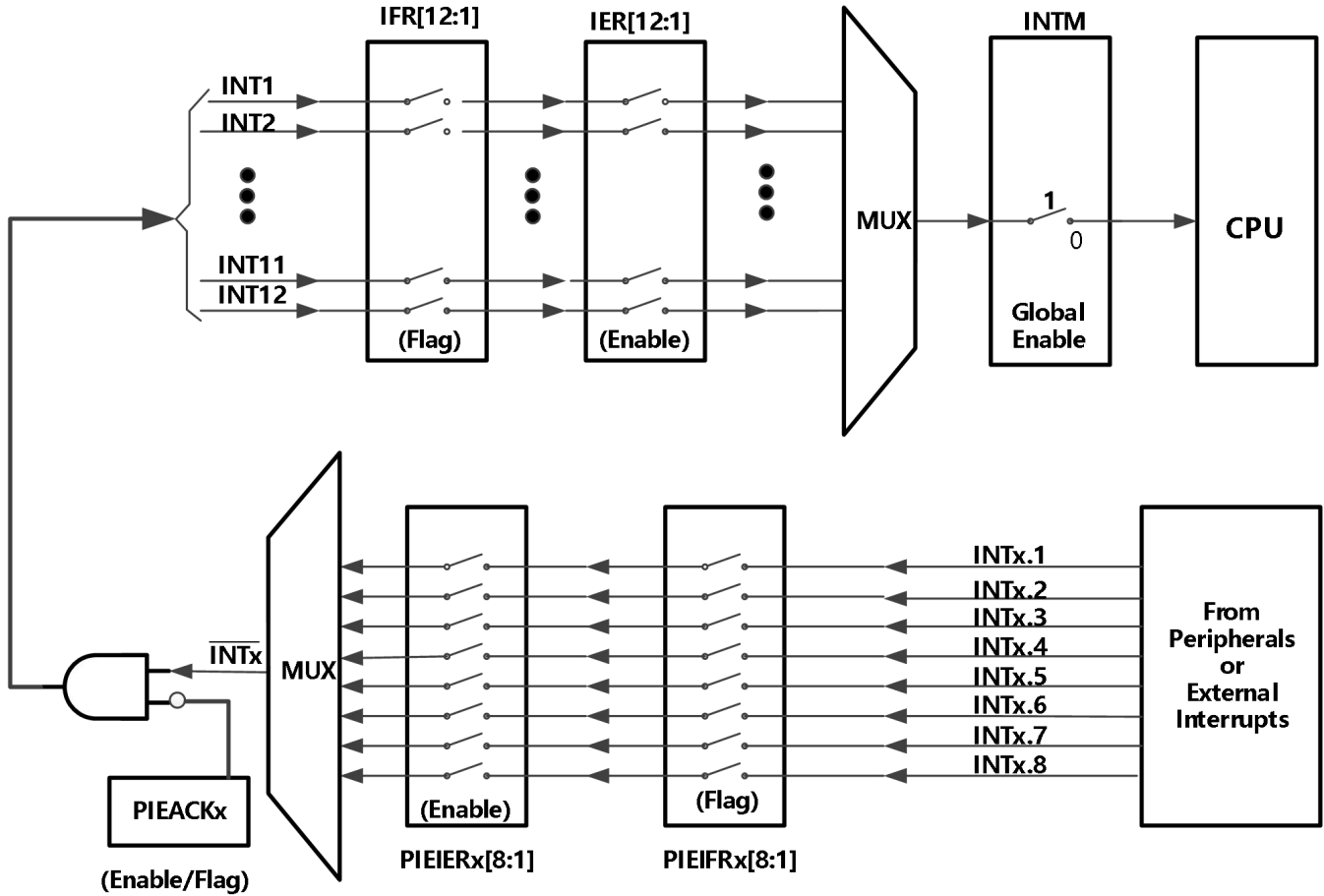


图 2.6-2 使用 PIE 块的中断复用

表 2-12 PIE 外设中断⁽¹⁾

CPU 中断	PIE 中断							
INTx.8	INTx.8	INTx.8	INTx.8	INTx.8	INTx.8	INTx.8	INTx.8	INTx.8
WAKEINT	WAKEINT	WAKEINT	WAKEINT	WAKEINT	WAKEINT	WAKEINT	WAKEINT	WAKEINT
(LPM/WD)	(LPM/WD)	(LPM/WD)	(LPM/WD)	(LPM/WD)	(LPM/WD)	(LPM/WD)	(LPM/WD)	(LPM/WD)
INT1	INT1	INT1	INT1	INT1	INT1	INT1	INT1	INT1
INT2	被保留	INT2	被保留	INT2	被保留	INT2	被保留	INT2
INT3	被保留	INT3	被保留	INT3	被保留	INT3	被保留	INT3
INT4	被保留	INT4	被保留	INT4	被保留	INT4	被保留	INT4
INT5	被保留	INT5	被保留	INT5	被保留	INT5	被保留	INT5
INT6	被保留	INT6	被保留	INT6	被保留	INT6	被保留	INT6
INT7	被保留	INT7	被保留	INT7	被保留	INT7	被保留	INT7
INT8	被保留	INT8	被保留	INT8	被保留	INT8	被保留	INT8
INT9	被保留	INT9	被保留	INT9	被保留	INT9	被保留	INT9
INT10	被保留	INT10	被保留	INT10	被保留	INT10	被保留	INT10
INT11	被保留	INT11	被保留	INT11	被保留	INT11	被保留	INT11
INT12	被保留	INT12	被保留	INT12	被保留	INT12	被保留	INT12

(1) 96 个可能中断中，目前有 45 个正在使用。其余中断保留供未来的器件使用。如果它们在 PIEIFRx 级被启用并且这个组中的中断没有一个被外设使用，这些中断可被用作软件中断。否则，在修改 PIEIFRx 的同时意外地清楚它们标志，来自外设的中断也许会丢失。总的来说，在两个安全情况下，被保留的中断可被用作软件中断

- 组内没用外设将中断置为有效。
- 没有外设中断被分配为到这个组（例如，PIE 组 12）。

表 2-13 PIE 配置和控制寄存器 (1)

名称	地址	大小 (X 16)	说明
PIECTRL	0x0000 0CE0	1	PIE,控制寄存器
PIEACK	0x0000 0CE1	1	PIE,确认寄存器
PIEIER1	0x0000 0CE2	1	PIE,INT1 组启用寄存器
PIEIFR1	0x0000 0CE3	1	PIE,INT1 组标志寄存器
PIEIER2	0x0000 0CE4	1	PIE,INT2 组启用寄存器
PIEIFR2	0x0000 0CE5	1	PIE,INT2 组标志寄存器
PIEIER3	0x0000 0CE6	1	PIE,INT3 组启用寄存器
PIEIFR3	0x0000 0CE7	1	PIE,INT3 组标志寄存器
PIEIER4	0x0000 0CE8	1	PIE,INT4 组启用寄存器
PIEIFR4	0x0000 0CE9	1	PIE,INT4 组标志寄存器
PIEIER5	0x0000 0CEA	1	PIE,INT5 组启用寄存器
PIEIFR5	0x0000 0CEB	1	PIE,INT5 组标志寄存器
PIEIER6	0x0000 0CEC	1	PIE,INT6 组启用寄存器
PIEIFR6	0x0000 0CED	1	PIE,INT6 组标志寄存器
PIEIER7	0x0000 0CEE	1	PIE,INT7 组启用寄存器
PIEIFR7	0x0000 0CEF	1	PIE,INT7 组标志寄存器
PIEIER8	0x0000 0CF0	1	PIE,INT8 组启用寄存器
PIEIFR8	0x0000 0CF1	1	PIE,INT8 组标志寄存器
PIEIER9	0x0000 0CF2	1	PIE,INT9 组启用寄存器
PIEIFR9	0x0000 0CF3	1	PIE,INT9 组标志寄存器
PIEIER10	0x0000 0CF4	1	PIE,INT10 组启用寄存器
PIEIFR10	0x0000 0CF5	1	PIE,INT10 组标志寄存器
PIEIER11	0x0000 0CF6	1	PIE,INT11 组启用寄存器
PIEIFR11	0x0000 0CF7	1	PIE,INT11 组标志寄存器
PIEIER12	0x0000 0CF8	1	PIE,INT12 组启用寄存器
PIEIFR12	0x0000 0CF9	1	PIE,INT12 组标志寄存器
被保留	0x0000 0CFA-0x0000 0CFF	6	被保留

PIE 配置和控制寄存器未受 EALLOW 模式保护。PIE 向量表受保护

2.6.1 外部中断

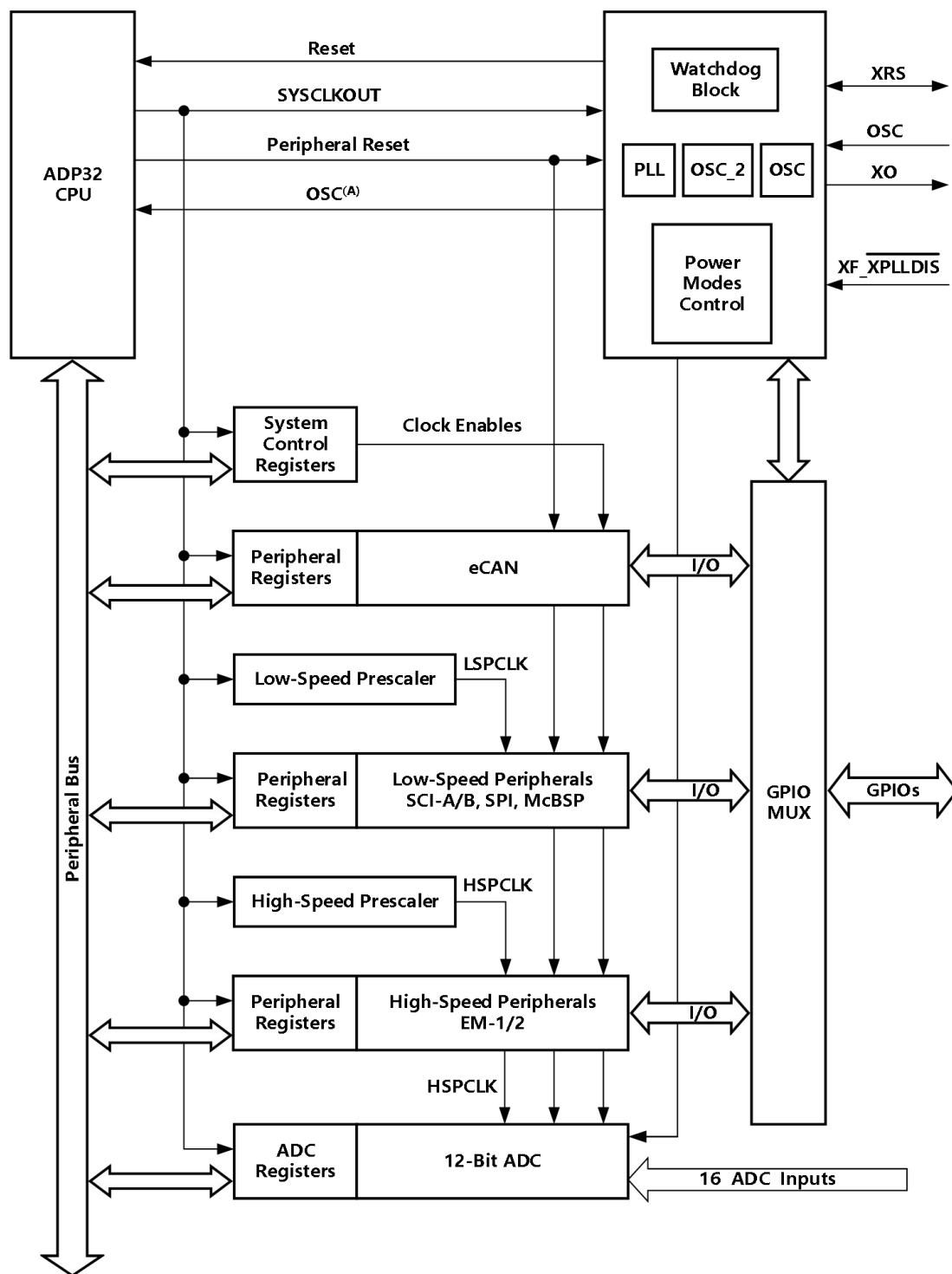
表 2-14 外部中断寄存器

名称	地址	大小 (X 16)	说明
XINT1CR	0x00 7070	1	XINT1 控制寄存器
XINT2CR	0x00 7071	1	XINT2 控制寄存器
被保留	0x00 7072-0x00 7076	5	
XNMICR	0x00 7077	1	XNMI 控制寄存器
XINT1CTR	0x00 7078	1	XINT1 计数器寄存器
XINT2CTR	0x00 7079	1	XINT2 计数器寄存器
被保留	0x00 707A-0x00 707E	5	
XNMICTR	0x00 707F	1	XNMI 计数器寄存器

每个外部中断可被启用/禁用或者使用正边沿或者负边沿来限定中断。

2.7 系统控制

这个部分描述了 ADP32FXX 振荡器、PLL 和时钟、看门狗功能和低功耗模式。图 2.7 显示了将进行讨论的 ADP32FXX 器件中的不同时钟和复位域。



A. CLKIN 是到 CPU 的时钟输入。SYSCLKOUT 是 CPU 的输出时钟。它们的频率相同

图 2.7 时钟和复位域

PLL、时钟、看门狗和低功耗模式由表 2-15 中列出的寄存器控制。

表 2-15. PLL、时钟、看门狗和低功耗模式寄存器⁽¹⁾

名称	地址	大小 (X 16)	说明
被保留	0x00 7010-0x00 7017	8	
被保留	0x00 7018	1	

ADP32FXX Digital Signal Processor

被保留	0x00 7019	1	
HISPCR	0x00 701A	1	HSPCLK 时钟的高速外设时钟预分频器寄存器
LOSPCR	0x00 701B	1	低速外设时钟预分频器寄存器用于 LSPCLK 时钟
PCLKCR	0x00 701C	1	外设时钟控制寄存器
被保留	0x00 701D	1	
LPMCR0	0x00 701E	1	低功耗模式控制寄存器 0
LPMCR1	0x00 701F	1	低功耗模式控制寄存器 1
被保留	0x00 7020	1	
PLLCR	0x00 7021	1	PLL 控制寄存器
SCSR	0x00 7022	1	系统控制与状态寄存器
WDCNTR	0x00 7023	1	安全装置计数器寄存器
被保留	0x00 7024	1	
WDKEY	0x00 7025	1	安全装置复位密钥寄存器
被保留	0x00 7026-0x00 7028	3	
WDCR	0x00 7029	1	安全装置控制寄存器
被保留	0x00 702A-0x00 702F	6	

(1) 所有上述寄存器只能通过执行 EALLOW 指令才能访问。

(2) 只用通过 XRS 才能将 PLL 控制寄存器(PLLCR)被复位至一个已知的状态。仿真复位 (通过 Code Composer Studio)将不会复位 PLLCR。

2.8 OSC 和 PLL 块

图 2.8-1 显示了 ADP32FXXX 上的 OSC 和 PLL 块。

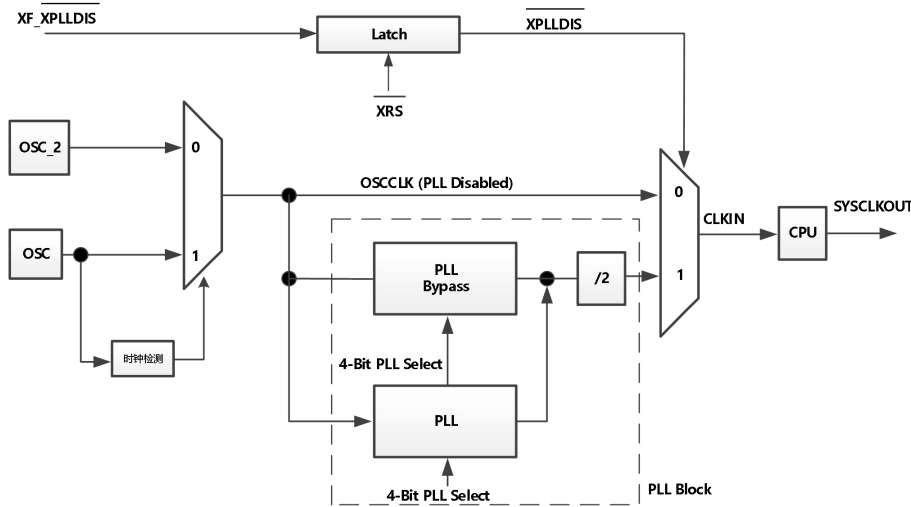


图 2.8-1 OSC 和 PLL 块

片载振荡器电路启用一个使用 OSC 和 X0 引脚连接至 ADP32FXXX 器件的晶振。如果一个晶振未被使用，那么一个外部振荡器可以直接接至 OSC 引脚，而 X0 引脚保持未连接。如果未检测到外部时钟输入或者片上 OSC 产生振荡信号，输入时钟切换到片内 OSC_2，由片内 RC 产生的 $30\text{MHz} \pm 10\%$ 的时钟驱动。PLLCR 位【3:0】设定计时比。

表 2-16 PLLCR 寄存器位定义

位	名称	类型	$\overline{\text{RESET}}$ RESET	说明																																																			
15:4	被保留	R=0	0:0																																																				
3:0	DIV	读/写	0, 0, 0, 0	<p>SYSCLKOUT = OSC * N, 在这里 n 是 PLL 倍乘因子。</p> <table border="1"> <thead> <tr> <th>位值</th> <th>n</th> <th>SYSCLKOUT</th> </tr> </thead> <tbody> <tr><td>0000</td><td>PLL 被绕过</td><td>OSC/2</td></tr> <tr><td>0001</td><td>1</td><td>OSC/2</td></tr> <tr><td>0010</td><td>2</td><td>OSC</td></tr> <tr><td>0011</td><td>3</td><td>OSC*1.5</td></tr> <tr><td>0100</td><td>4</td><td>OSC*2</td></tr> <tr><td>0101</td><td>5</td><td>OSC*2.5</td></tr> <tr><td>0110</td><td>6</td><td>OSC*3</td></tr> <tr><td>0111</td><td>7</td><td>OSC*3.5</td></tr> <tr><td>1000</td><td>8</td><td>OSC*4</td></tr> <tr><td>1001</td><td>9</td><td>OSC*4.5</td></tr> <tr><td>1010</td><td>10</td><td>OSC*5</td></tr> <tr><td>1011</td><td>11</td><td>被保留</td></tr> <tr><td>1100</td><td>12</td><td>被保留</td></tr> <tr><td>1101</td><td>13</td><td>被保留</td></tr> <tr><td>1110</td><td>14</td><td>被保留</td></tr> <tr><td>1111</td><td>15</td><td>被保留</td></tr> </tbody> </table>	位值	n	SYSCLKOUT	0000	PLL 被绕过	OSC/2	0001	1	OSC/2	0010	2	OSC	0011	3	OSC*1.5	0100	4	OSC*2	0101	5	OSC*2.5	0110	6	OSC*3	0111	7	OSC*3.5	1000	8	OSC*4	1001	9	OSC*4.5	1010	10	OSC*5	1011	11	被保留	1100	12	被保留	1101	13	被保留	1110	14	被保留	1111	15	被保留
位值	n	SYSCLKOUT																																																					
0000	PLL 被绕过	OSC/2																																																					
0001	1	OSC/2																																																					
0010	2	OSC																																																					
0011	3	OSC*1.5																																																					
0100	4	OSC*2																																																					
0101	5	OSC*2.5																																																					
0110	6	OSC*3																																																					
0111	7	OSC*3.5																																																					
1000	8	OSC*4																																																					
1001	9	OSC*4.5																																																					
1010	10	OSC*5																																																					
1011	11	被保留																																																					
1100	12	被保留																																																					
1101	13	被保留																																																					
1110	14	被保留																																																					
1111	15	被保留																																																					

(1) PLLCR 寄存器被 RESET 复位线路复位至一个已知状态，如果是由调试器发生的复位命令，PLL 计时比不发生变化。

2.8.1 输入时钟丢失

在 PLL 启用模式，如果输入时钟 OSC 或者振荡器时钟被移除或者缺失。输入时钟缺失检测电路自动将时钟源切换到片内 OSC_2 上，由 OSC_2 提供片内 30MHz 参考时钟源。这种主动式备用时钟导入模式可以保证系统在时钟丢失时能够继续“跛行”，等待外部时钟或 OSC 恢复正常时自动切换回原参考时钟继续运行尽可能避免因时钟丢失导致系统的异常。由于 OSC_2 信号源的频率和相位精度差于片外时钟源，可能导致系统工作精度下降，运行部稳定，甚至系统无法正常工作，在对时钟稳定性要求较高的场合，当采用外部时钟时，建议用户增加外围时钟丢失检测电路用来产生一个时钟丢失事件信号。该信号一般用来驱动芯片的 XRS 复位。当采用片上 OSC 晶振时，需要保证芯片的 OSC 正常工作。当系统进入 HALT 模式时，PLL 仍将发布一个“跛行模式”时钟。这个跛行模式时钟将持续为 CPU 和外设提供典型频率为 1-4MHz 的时钟。为了使这个特性正常运行，PLL 寄存器应该已经被写入一个非零值。

2.9 基于 PLL 的时钟模块

ADP32FXX 有一个片上、基于 PLL 的时钟模块。这个模块为器件提供所有需要的时钟信号，以及对低功耗模式进入的控制。PLL 具有 4 位比率控制以选择不同的 CPU 时钟频率。在写入 PLLCR 寄存器之前。看门狗模块应该被禁用。在 PLL 模式稳定后，它可被重新启用（如果需要的话），重新启用所需的时间为 131072 个 OSC 周期。

基于 PLL 的时钟模块提供两种操作模式：

- **晶振运行**：这个模式允许使用一个外部晶振/谐振器来为器件提供时基。
- **外部时钟源运行**：这个模式允许内部振荡器旁通。此器件时钟由一个外部时钟源数生成并从 OSC 引脚输入。

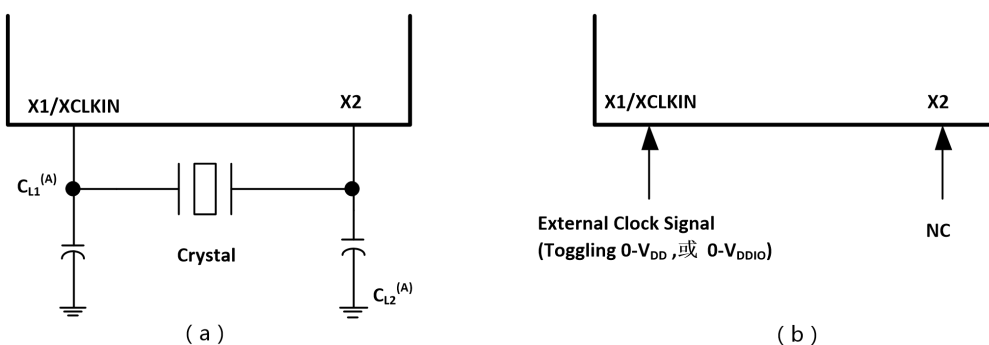


图 2.9 推荐的晶振/时钟连接

建议客户让谐振器/晶振销售商明确他们的器件与 DSP 芯片一起工作的特性。谐振器/晶体销售商具有调谐谐振电路的设备和专业技术。销售商也可以在正确谐振植选择方面为客户提出建议，这些值可在整个运行范围内确保启动和稳定性。

表 2-17.可能的 PLL 配置模式

PLL 模式	注释	SYSCLKOUT
PLL 被禁用	复位时，通过将 $\overline{\text{PPLDIS}}$ 引脚置为低电平来使得 PLL 块被完全禁用。CPU 的时钟输入 (CLKIN)直接来自 OSC 引脚上的时钟信号。	OSC
PLL 被旁路	如果 PLL 未被禁用，加电时，PLL 使用的为缺省设置。PLL 本身被绕过。然而，在馈入 CPU 之前，PLL 块内的/2 模块将这个 OSC 引脚上的时钟输入二分频。	OSC/2
PLL 被启用	通过将一个非零值“n”写入 PLLCR 寄存器实现。在馈入 CPU 之前，PLL 块内的/2 模块将 PLL 的输出二分频。	(OSC*n) /2

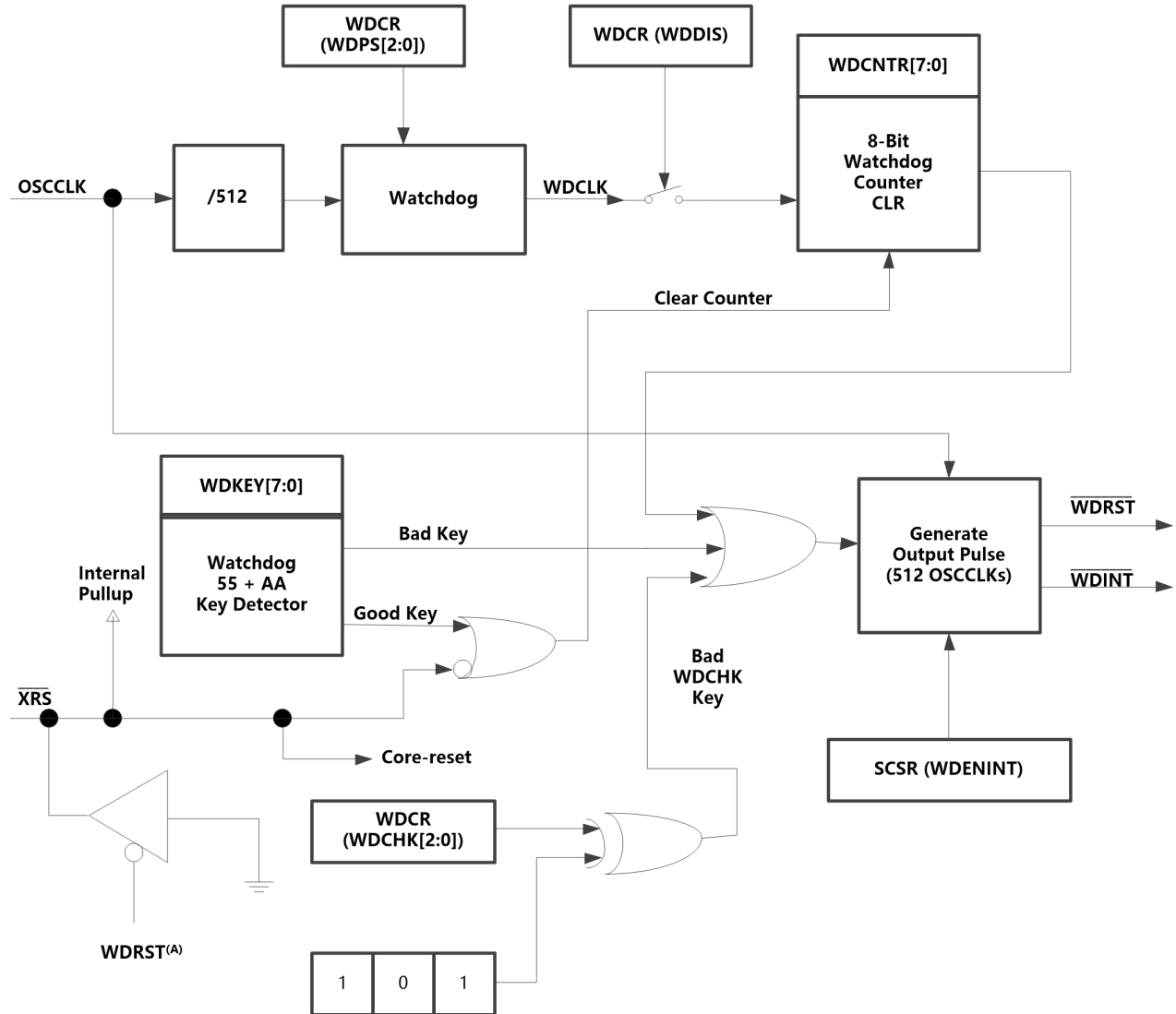
2.10 外部参考振荡器时钟选项

用于频率为 **30MHz** 的外部石英晶振的典型技术参考如下：

- .基本模式,并联谐振
- . C_L (负载电容)=12pF
- . $C_{L1} = C_{L2} = 12\text{pF}$
- . $C_{\text{并}} = 6\text{pF}$
- .ESR 范围=25 至 40 Ω

2.11 看门狗模块

ADP32FXX 上的看门狗只要 8 位安全装置上数计数器达到它的最大值，此时生成一个 512 振荡器时钟宽度 (OSCCLK)。输出脉冲要防止这种情况，用户可以禁用该计数器，或者必须通过软件定期将一个 0x55+0xAA 序列写入至安全装置密钥寄存器中，从而使看门狗计数器复位。图 2.11 显示了看门狗模块内的各种功能块。



A. \overline{WDRST} 信号在 512 个 OSCCLK 周期内被驱动为低电平。

图 2.11 看门狗模块

\overline{WDINT} 信号使能看门狗,作为 IDLE/STANDY 模式唤醒的定时器使用。

在 STANDBY 模式中，器件上的所有外设关闭。唯一继续工作的外设是看门狗。WATCHDOG 模块将关闭 PLL 块或者振荡器时钟。 \overline{WDINT} 信号被馈送到 LPM 块以便它可以将器件从 STANDBY 唤醒（如已启用）。更多细节，请见节 2.12，低功耗模块。

在 IDLE 模式中， \overline{WDINT} 信号可通过 PIE 来生成一个到 CPU 的中断来将 CPU 从 IDLE 模式中唤醒。

在 HALT 模式中，不能使用此功能，这是因为振荡器（和 PLL）被关闭，因此看门狗也被关闭。

2.12 低功耗模式

表 2-18 ADP32FXX 低功耗模式

模式	LPM(1:0)	OSCCLK	CLKIN	SYSCLKOUT	退出 ⁽¹⁾
正常	X,X	打开	打开	打开	$\overline{\text{XRSRESET}}$, $\overline{\text{WDINT}}$,
IDLE	0,0	打开	打开	打开 ⁽²⁾	任何被启动的中断, XNMI, 调试器 ⁽³⁾ $\overline{\text{RESET}}$, $\overline{\text{WDINT}}$, XINT1, XNMI,
STANDBY	0,1	打开 (看门狗仍然运行)	关闭	关闭	$\overline{\text{T1/2/3/4CTRIP}}$, $\overline{\text{C1/2/3/4/5/6TRIP}}$, SCIRXDA, SCIRXDB, CANRX, 调试器 ⁽³⁾ $\overline{\text{RESET}}$,
HALT	1,X	关闭 (振荡器和 PLL 关闭, 看门狗不工作)	关闭	关闭	XNMI, 调试器 ⁽³⁾

- (1) “退出”，列出哪些信号或在哪些情况下会退出低功耗模式，在这些信号中的任何一个低电平信号，将退出低功耗状态。此信号必须保持低电平足够长时间以便器件识别中断。否则，将不会从 **IDLE 模式** 中退出，而器件将返回到已确认的低功耗模式。
- (2) ADP32FXX 上运行状态与 ADP16X 不同，在 ADP32FXX 上来自内核 (SYSCLKOUT) 的输出时钟让将起作用,ADP16X 上时钟将被关闭
- (3) 在 ADP32FXX 上，内核时钟 (CLKIN) 被关闭，JTAG 仍然可以工作

三种不同的低功耗模式运行状态如下：

- IDLE 模式** 通过任一被启用的中断或者一个被处理器识别的 XNMI 来退出此模式。LPM 块在这个模式期间，在 LPMCR0(LPM)位被设定为 0，0 时，LPM 块不执行任何任务。
- STANDBY 模式** 被 LPMCR1 寄存器选中 (包括 XNMI 在内) 的所有信号均可将器件从 STANDBY 模式中唤醒。用户需要选择使用哪一个，或者哪几个信号将器件唤醒。所选的信号在经过 LPMCR0 寄存器中设定 OSCCLK 的数量时钟后，在合格时间内被取值。
- HALT 模式** 只有 $\overline{\text{RESET}}$ 和 XNMI 外部信号可将器件从 HALT 模式中唤醒。到内核的 XNMI 输入有一个启用/禁用位。因此，可安全使用 XNMI 信号启动该功能。

注

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

3 外设

ADP32F1X 的集成外设在下方的子部分中进行了说明：

- 三个 32 位 CPU 定时器
- 两个事件管理器模块(EM1 , EM2)
- 增强型模数转换器(ADC) 模块
- 增强型控制器局域网(eCAN) 模块
- 多通道缓冲串行端口(McBSP) 模块
- 串行通信接口模块(SCI-A , SCI-B)
- 串行外设接口(SPI) 模块
- 数字 I/O 和共用引脚功能

3.1 32 位 CPU 定时器 0/1/2

在ADP32FXX器件上有三个32 位CPU 定时器(CPU-TIMER0/1/2)

定时器2 为DSP/BIOS 预留。可以在用户应用程序中使用CPU 定时器0 和定时器1。这些定时器与出现在事件管理器模块(EM1 , EM2) 中的通用(GP) 定时器不同。

注:

如果应用不使用 DSP/BIOS , 那么 CPU 定时器 2 可被用在应用中。

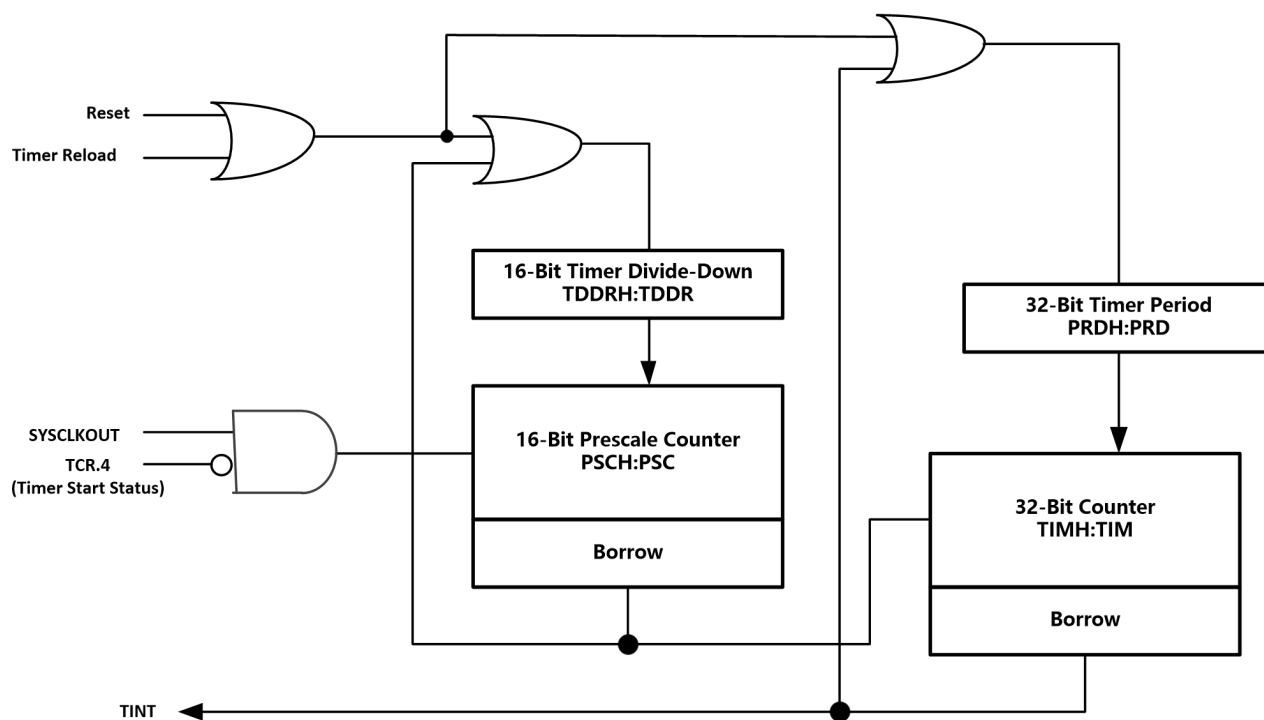


图 3-1. CPU 定时器

在ADP32FXX器件中，定时器中断信号(TINT0，TINT1，TINT2)的连接方式如图3-2中所示。

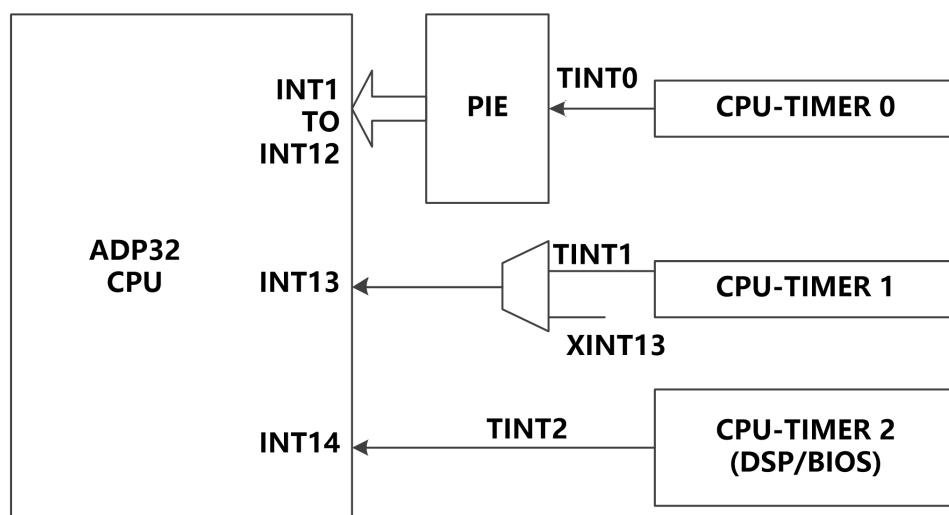


图 3-2. CPU 定时器中断信号与输出信号

A. 定时器寄存器连接到ADP32FXX处理器的存储器总线。

B. 定时器的时序与处理器时钟的 SYSCLKOUT 同步。

定时器的通常操作如下：32 位计数器寄存器“TIMH:TIM”装入周期寄存器“PRDH:PRD”中的值。计数器寄存器按ADP32FXX的SYSCLKOUT 频率递减。当计数器到达0 时，一个定时器中断输出信号生成一个中断脉冲。表3-1中列出的寄存器用于配置定时器。

表 3-1. CPU 定时器 0, 1, 2 配置和控制寄存器

名称	地址大小(x16)	大小(x16)	说明
TIMER0TIM	0x00 0C00	1	CPU 定时器 0, 计数器寄存器
TIMER0TIMH	0x00 0C01	1	CPU 定时器 0, 计数器寄存器高电平
TIMER0PRD	0x00 0C02	1	CPU 定时器 0, 周期寄存器
TIMER0PRDH	0x00 0C03	1	CPU 定时器 0, 周期寄存器高电平
TIMER0TCR	0x00 0C04	1	CPU 定时器 0, 控制寄存器
被保留	0x00 0C05	1	
TIMER0TPR	0x00 0C06	1	CPU 定时器 0, 预分频寄存器
TIMER0TPRH	0x00 0C07	1	CPU 定时器 0, 预分频寄存器高电平
TIMER1TIM	0x00 0C08	1	CPU 定时器 1, 计数器寄存器
TIMER1TIMH	0x00 0C09	1	CPU 定时器 1, 计数器寄存器高电平
TIMER1PRD	0x00 0C0A	1	CPU 定时器 1, 周期寄存器
TIMER1PRDH	0x00 0C0B	1	CPU 定时器 1, 周期寄存器高电平
TIMER1TCR	0x00 0C0C	1	CPU 定时器 1, 控制寄存器
被保留	0x00 0C0D	1	
TIMER1TPR	0x00 0C0E	1	CPU 定时器 1, 预分频寄存器
TIMER1TPRH	0x00 0C0F	1	CPU 定时器 1, 预分频寄存器高电平
TIMER2TIM	0x00 0C10	1	CPU 定时器 2, 计数器寄存器
TIMER2TIMH	0x00 0C11	1	CPU 定时器 2, 计数器寄存器高电平
TIMER2PRD	0x00 0C12	1	CPU 定时器 2, 周期寄存器
TIMER2PRDH	0x00 0C13	1	CPU 定时器 2, 周期寄存器高电平
TIMER2TCR	0x00 0C14	1	CPU 定时器 2, 控制寄存器
被保留	0x00 0C15	1	
TIMER2TPR	0x00 0C16	1	CPU 定时器 2, 预分频寄存器
TIMER2TPRH	0x00 0C17	1	CPU 定时器 2, 预分频寄存器高电平
被保留 0x00 0C18	0x00 0C18	40	

3.2 事件管理器模块(EM1 , EM2)

事件管理器模块包括通用(GP) 定时器、完全比较/ PWM 单元、捕捉单元、和正交编码器脉冲(QEP) 电路。EM1 和 EM2 定时器、比较单元、和捕捉单元功能一致。然而，EM1 和 EM2 的定时器/单元名称不同。表 3-2 显示了使用的模块和信号名称。表 3-3 显示了事件管理器模块可用的特性和功能并且标出了 EM1 命名规则。

事件管理器 A 和 B 具有完全一样的外设寄存器集，对于 EM1 外设寄存器从 7400h 开始，对于 EM2 则从 7500h 开始。这个部分中的段落描述了使用 EM1 命名规则的 GP 定时器、比较单元、捕捉单元、和 QEP 的功能。在功能方面，这些段落同样适用于 EM2 - 然而，模块/信号名称不同。表 3-3 中列出了 EM1 寄存器。

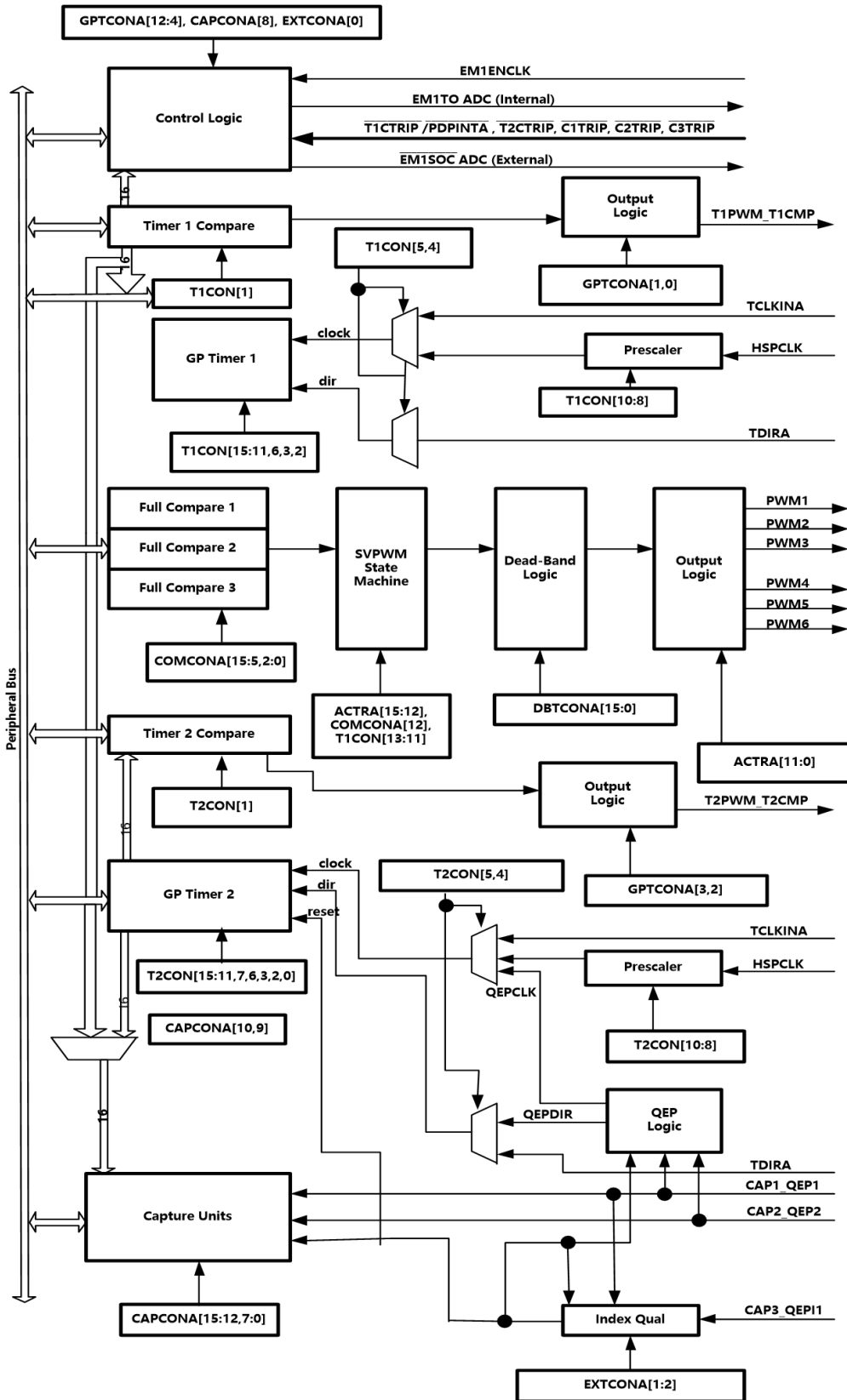
表 3-2. 针对 EM1 和 EM2 的模块和信号名称

事件管理器模块	EM1		EM2	
	模块	信号	模块	信号
GP 定时器	GP 定时器 1	T1PWM/T1CMP	GP 定时器 3	T3PWM/T3CMP
	GP 定时器 2	T2PWM/T2CMP	GP 定时器 4	T4PWM/T4CMP
比较单元	比较 1	PWM1/2	比较 4	PWM7/8
	比较 2	PWM3/4	比较 5	PWM9/10
	比较 3	PWM5/6	比较 6	PWM11/12
捕捉单元	捕捉 1	CAP1	捕捉 4	CAP4
	捕捉 2	CAP2	捕捉 5	CAP5
	捕捉 3	CAP3	捕捉 6	CAP6
QEP 通道	QEP1	QEP1	QEP3	QEP3
	QEP2	QEP2	QEP4	QEP4
	QEPI1		QEPI2	
外部时钟输入	外部方向时钟	TDIRA	外部方向时钟	TDIRB
		TCLKINA		TCLKINB
外部触发输入	比较	C1TRIP		C4TRIP
		C2TRIP		C5TRIP
		C3TRIP		C6TRIP
外部触发输入		T1CTRIP_PDPINTA		T3CTRIP_PDPINTB
		T2CTRIP/EM1SOC		T4CTRIP/EM2SOC

表 3-3. EM1 寄存器⁽¹⁾

名称	地址	大小(x16)	说明
GPTCONA	0x00 7400	1	GP 定时器控制寄存器 A
T1CNT	0x00 7401	1	GP 定时器 1 计数器寄存器
T1CMPR	0x00 7402	1	GP 定时器 1 比较寄存器
T1PR	0x00 7403	1	GP 定时器 1 周期寄存器
T1CON	0x00 7404	1	GP 定时器 1 控制寄存器
T2CNT	0x00 7405	1	GP 定时器 2 计数器寄存器
T2CMPR	0x00 7406	1	GP 定时器 2 比较寄存器
T2PR	0x00 7407	1	GP 定时器 2 周期寄存器
T2CON	0x00 7408	1	GP 定时器 2 控制寄存器
EXTCONA	0x00 7409	1	GP 扩展扩展寄存器 A
COMCONA	0x00 7411	1	比较控制寄存器 A
ACTRA	0x00 7413	1	比较操作控制寄存器 A
DBTCONA	0x00 7415	1	死区定时器控制寄存器 A
CMPR1	0x00 7417	1	比较寄存器 1
CMPR2	0x00 7418	1	比较寄存器 2
CMPR3	0x00 7419	1	比较寄存器 3
CAPCONA	0x00 7420	1	捕捉控制寄存器
CAPFIFOA	0x00 7422	1	捕捉 FIFO 状态寄存器 A
CAP1FIFO	0x00 7423	1	两级深捕捉 FIFO 堆栈 1
CAP2FIFO	0x00 7424	1	两级深捕捉 FIFO 堆栈 2
CAP3FIFO	0x00 7425	1	两级深捕捉 FIFO 堆栈 3
CAP1FBOT	0x00 7427	1	捕捉 FIFO 堆栈 1 的栈底寄存器
CAP2FBOT	0x00 7428	1	捕捉 FIFO 堆栈 2 的栈底寄存器
CAP3FBOT	0x00 7429	1	捕捉 FIFO 堆栈 3 的栈底寄存器
EM1IMRA	0x00 742C	1	中断屏蔽寄存器 A
EM1IMRB	0x00 742D	1	中断屏蔽寄存器 B
EM1IMRC	0x00 742E	1	中断屏蔽寄存器 C
EM1IFRA	0x00 742F	1	中断标志寄存器 A
EM1IFRB	0x00 7430	1	中断标志寄存器 B
EM1IFRC	0x00 7431	1	中断标志寄存器 C

(1) EM2寄存器集完全一样，除了地址范围是从0x007500 至0x00753F。上面的寄存器被映射到区域2。这个空间只允许16位访问。32位访问会生成未定义的后果。



A. EM1 模块与 EM2 模块相似。

图 3-2. 事件管理器 A 功能方框图

3.2.1 通用(GP) 定时器

有两个 GP 定时器。GP 定时器 x (对于 EM1, x=1 或者 2; 对于 EM2, x=3 或者 4) 包括:

- 一个 16 位定时器、上数/下数计数器、TxCNT，用于读取或者写入
- 一个 16 位定时器比较寄存器，TxCMPR（使用影子寄存器实现双缓冲），用于读取或者写入
- 一个 16 位定时器周期寄存器，TxPR（使用影子寄存器实现双缓冲），用于读取或者写入
- 一个 16 位定时器控制寄存器，TxCON，用于读取或者写入
- 可选内部或者外部输入时钟
- 一个针对内部或者外部时钟输入的可编程预分频器
- 控制和中断逻辑，用于四个可屏蔽中断：下溢，溢出，定时器比较，和周期中断
- 一个可选方向输入引脚(TDIRx)（当方向上数/下数计数模式被选中时进行上计数或者倒计数）

GP 定时器可独立运行或者互相之间同步。与每个 GP 定时器相关的比较寄存器可被用于比较功能和 PWM 波形生成。对于每个处在上计数或者上计数/倒计数运行中的 GP 定时器来说，有三个连续运行模式。带有可编程分频器的内部或者外部输入时钟被用于每个 GP 定时器。GP 定时器还为其它事件管理器子模块提供时基：GP 定时器 1 为所有比较和 PWM 电路提供时基，GP 定时器 2 为捕捉单元和正交脉冲计数运行提供时基。按照需要，周期和比较寄存器的双缓冲可实现定时器(PWM) 周期以及比较/ PWM 脉冲宽度的可编程变化。

3.2.2 完全比较单元

在每个事件管理器上有三个完全比较单元。这些比较单元使用 GP 定时器作为时基并为使用可编程死区电路的比较和 PWM 波形生成产生六个输出。这六个输出中每一个输出的状态可独立配置。比较单元的比较寄存器是双缓冲的，从而可实现所需的比较/ PWM 脉冲宽度可编程变化。

3.2.3 可编程死区发生器

对于每一个比较单元输出，可单独启用/禁用死区生成。死区发生器电路为每个比较单元输出信号产生两个输出（带有或者不带有死区）。通过双缓冲 ACTRx 寄存器，死区发生器的输出状态可按照需要配置和改变。

3.2.4 PWM 波形生成

每一个事件管理器可同时产生多达 8 个 PWM 波形（输出）：带有可编程死区的三个完全比较单元生成的三个独立对（六个输出），和 GP 定时器比较生成的两个独立 PWM。

3.2.5 双更新 PWM 模式

ADP32FXX 事件管理器支持“双更新 PWM 模式。”这个模式是指一个 PWM 运行模式，在这个模式中，一个 PWM 脉冲前沿和后沿的位置可在每个 PWM 周期内独立修改。为了支持这个模式，确定一个 PWM 脉冲边沿位置的比较寄存器必须允许（缓冲的）比较值在一个 PWM 周期开始时更新一次，在一个 PWM 周期中间的另外一个时间更新一次。ADP32FXX 事件管理器中的比较寄存器是完全缓冲的并且支持 3 个比较值重新载入/更新（缓冲器中的值变为有效）模式。这些模式已经在之前被证明为比较值重新载入条件。支持双更新 PWM 模式的重新载入条件在下溢（PWM 周期的开始）

或者周期（PWM 周期的中间）上被重新载入。双更新 PWM 模式可通过使用这个用于比较值重新载入的条件实现。

3.2.6 PWM 特性

PWM 特性如下：

- 16 位寄存器。
- 针对 PWM 输出引脚对的宽范围可编程死区。
- PWM 摇频所需的 PWM 载波频率的改变。
- PWM 脉冲宽度内的改变和每个 PWM 周期之后所需的改变。
- 外部可屏蔽电源和驱动保护中断。
- 脉冲波形发生器电路，用于不对称、对称、和四空间矢量 PWM 波形的可编程生成。
- 使用比较和周期寄存器的自动重新载入可大大减少 CPU 的开销。
- 在 $\overline{\text{PDPINTx}}$ 引脚被驱动至低电平并且在 $\overline{\text{PDPINTx}}$ 信号验证之后，PWM 引脚被驱动至高阻抗状态。
 $\overline{\text{PDPINTx}}$ 引脚（验证之后）被反映在 COMCONx 寄存器的位 8 上。
 - $\overline{\text{PDPINTA}}$ 引脚状态被反映在 COMCONA 寄存器的位 8 上。
 - $\overline{\text{PDPINTB}}$ 引脚状态被反映在 COMCONB 寄存器的位 8 上。
- EXTCON 寄存器位提供针对每个 PWM 信号对的独立跳闸触点控制选项。

3.2.7 捕捉单元

捕捉单元为不同事件或者转换提供一个日志记录功能。当在捕捉输入引脚上检测到所选转换，CAPx（对于 EM1，x = 1, 2, 或者 3；对于 EM2，x = 4, 5, 或者 6）时，选定的 GP 定时器计数器的值被捕捉并存储在两级深 FIFO 堆栈内。比较单元由三个捕捉成。

捕捉单元包括下列特性：

- 一个 16 位捕捉控制寄存器，CAPCONx (R/W)。
- 一个 16 位捕捉 FIFO 状态寄存器，CAPFIFOx。
- 选择 GP 定时器 1/2（对于 EM1）或者 3/4（对于 EM2）作为时基。
- 3 个 16 位 2 级深 FIFO 堆栈，每个堆栈用于一个比较单元。
- 3 个比较输入引脚（对于 EM1，CAP1/2/3，对于 EM2，CAP4/5/6）- 每个捕捉单元一个输入引脚。
[所有输入与器件(CPU) 时钟同步。为了捕捉到一个转换，输入必须保持在其当前的电平以满足输入验证电路的要求。输入引脚 CAP1/2 和 CAP4/5 也可被用 QEP 电路的 QEP 输入。]
- 用户指定转换（上升边沿、下降边沿、或者这两个边沿）检测。
- 三个可屏蔽中断标志，一个用于每个捕捉单元。
- 如果它们不被用于捕捉功能，捕捉引脚也可被用作通用中断引脚。

3.2.8 正交编码器脉冲(QEP) 电路

两个捕捉输入（对于 EM1，CAP1 和 CAP2；对于 EM2，CAP4 和 CAP5）可被用于片载 QEP 电路与正交编码器脉冲的对接。在芯片上执行这些输入的完全同步。方向或者前沿正交脉冲序列被检测到，

并且GP定时器2/4 被两个输入信号的上升和下降边沿增量或者减量（每个输入脉冲频率的四倍）。借助于EXTCONA 寄存器位，EM1 QEP 电路也可将CAP3 用作一个捕捉索引引脚。相似地，借助于EXTCONB 寄存器位，EM2 QEP 电路可将CAP6 用作一个捕捉索引引脚。

3.2.9 外部 ADC 转换开始

EM1/EM2转换开始(SOC) 可被发送到一个外部引脚($\overline{\text{EM1SOC}}$ / $\overline{\text{EM2SOC}}$) 用于外部ADC 接口。 $\overline{\text{EM1SOC}}$ 和 $\overline{\text{EM2SOC}}$ 分别与 $\overline{\text{T2CTRIP}}$ 和 $\overline{\text{T4CTRIP}}$ 复用。

3.3 增强型模数转换器(ADC) 模块

图 3.3-1 显示了一个 ADC 模块的简化功能方框图 ADC 模块由一个带有内置采样保持(S/H) 电路的 12 位 ADC 组成。

ADC 模块的功能包括：

- 具有内置 S/H 的 12 位 ADC 内核
- 模拟输入：0.0V 至 3.0V（高于 3.0V 的电压产生满量程转换结果）
- 快速转换率：25MHz ADC 时钟，12.5 每秒百万次采样(MSPS) 时为 80ns
- 16 通道，被复用输入
- 自动定序功能在单次会话中可提供多达 16 次“自动转换”。可将每次转换编程为选择 16 个输入信道中的任何一个。
- 序列发生器可运行 2 个独立的 8 态序列发生器，或作为 1 个较大的 16 态序列发生器（即 2 个级联的 8 态序列发生器）。
- 用于存储转换值的 16 个结果寄存器（可分别寻址）
 - 输入模拟电压的数值源自：

Digital Value = 0, when input ≤ 0 V

$$\text{Digital Value} = 4096 \times \frac{\text{Input Analog Voltage} - \text{ADCL0}}{3} \text{ when } 0 \text{ V} < \text{input} < 3 \text{ V}$$

Digital Value = 4095, when input ≥ 3 V

- 作为转换开始(SOC) 序列源的多个触发器
 - S/W - 软件立即启动
 - EM1 - 事件管理器A（EM1内的多个事件源）
 - EM2- 事件管理器B（EM2内的多个事件源）
- 灵活的中断控制允许每个序列结束(EOS)或其它EOS上的中断请求。
- 序列发生器可运行在“启动/停止”模式，以便多个“时序触发器”进行同步转换。

ADP32FXX Digital Signal Processor

- EM1和EM2触发器可独立运行在双序列发生器模式。
- 采样保持(S/H)采集时间窗口具有独立的预分频控制。

ADP32FXX中的ADC 模块已被增强来为提供到事件管理器A 和B 的灵活接口。ADC 接口在一个快速，12位ADC模块周围建立，此模块具有在25MHz ADC 时钟上有一个80ns 的快速转换速率。

ADC 模块有16个通道这些通道可被配置为两个用于事件管理器A和B的独立8通道模块。可将两个独立的8通道模块级联以形成一个16通道模块。尽管有多个输入信道和2 个序列发生器，但在ADC 模块中只有一个转换器。图3-4显示了ADP32FXXADC模块的方框图。

2个8通道模块可自动对一系列转换定序，每个模块可以通过模拟MUX选择其中一个可用8信道。在级联模式中，自动序列发生器将作为一个单个16 通道序列发生器使用。在每个序列发生器一旦转换完成，所选的通道值将存储在各自的RESULT 寄存器中。系统可使用自动定序功能多次转换同一信道，以使用户执行过采样算法。这种过采样算法可提供比传统的单一采样转换结果更高的分辨率。

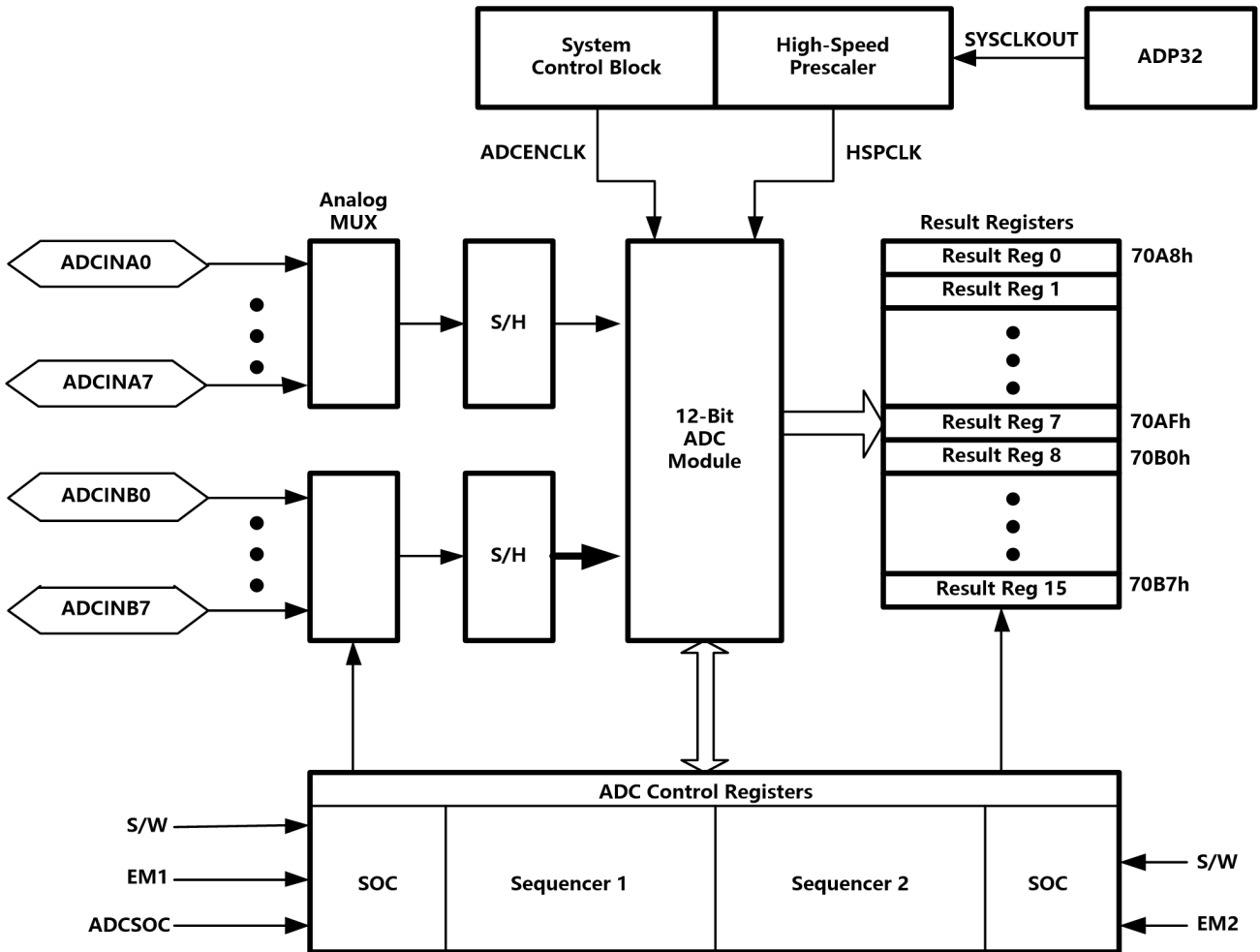


图3.3-1. ADP32FXX ADC 模块方框图

要获得指定的ADC 精度，正确的电路板布局非常关键。为尽可能达到最佳效果，引入ADCINxx 引脚的走线不应数字信号通道靠的太近。这是为了最大程度地减少数字线路上因ADC 输入耦合而产生的开关噪声。此外，必须使用适当的隔离技术，将ADC 模块电源引脚(V_{DDA1}/V_{DDA2} , $AVDDREF$)与数字电源隔离。要获得更佳的精度和ESD 保护，未使用的ADC 输入应该被接至模拟接地。

注释：

1. 用SYSCLKOUT 速率对ADC 寄存器进行访问。ADC 模块的内部时序由高速外设时钟(HSPCLK) 控制。
2. 基于ADCENCLK 和HALT 信号的ADC 模块的运行方式如下：

ADCENCLK：复位时，这个信号必须为低电平。虽然复位为低电平有效(RESET)，到寄存器的时钟将仍然起作用。有必要确保所有寄存器和模式进入它们的缺省复位状态。然而，模拟模块将处于一个低功耗非激活状态。一旦复位变为高电平，那么到寄存器的时钟将被禁用。当用户ADCENCLK 信号设定为高电平时，那么到寄存器的时钟将被启用并且模拟模块将被启用。在ADC 稳定并可被使用前，会有一个特定的时间延迟 (ms范围)。

HALT：这个信号只影响模拟模块。它不影响寄存器。如果为低电平，ADC 模块被加电。如果为高电平，ADC 模块进入低功耗模式。HALT 模式将停止到CPU 的时钟，这样将停止HSPCLK。因此，ADC 寄存器将被间接关闭。

ADP32FXX Digital Signal Processor

图3.3-2显示了针对内部基准的ADC 引脚偏置而图3.3-3显示了针对外部基准的ADC 引脚偏置。

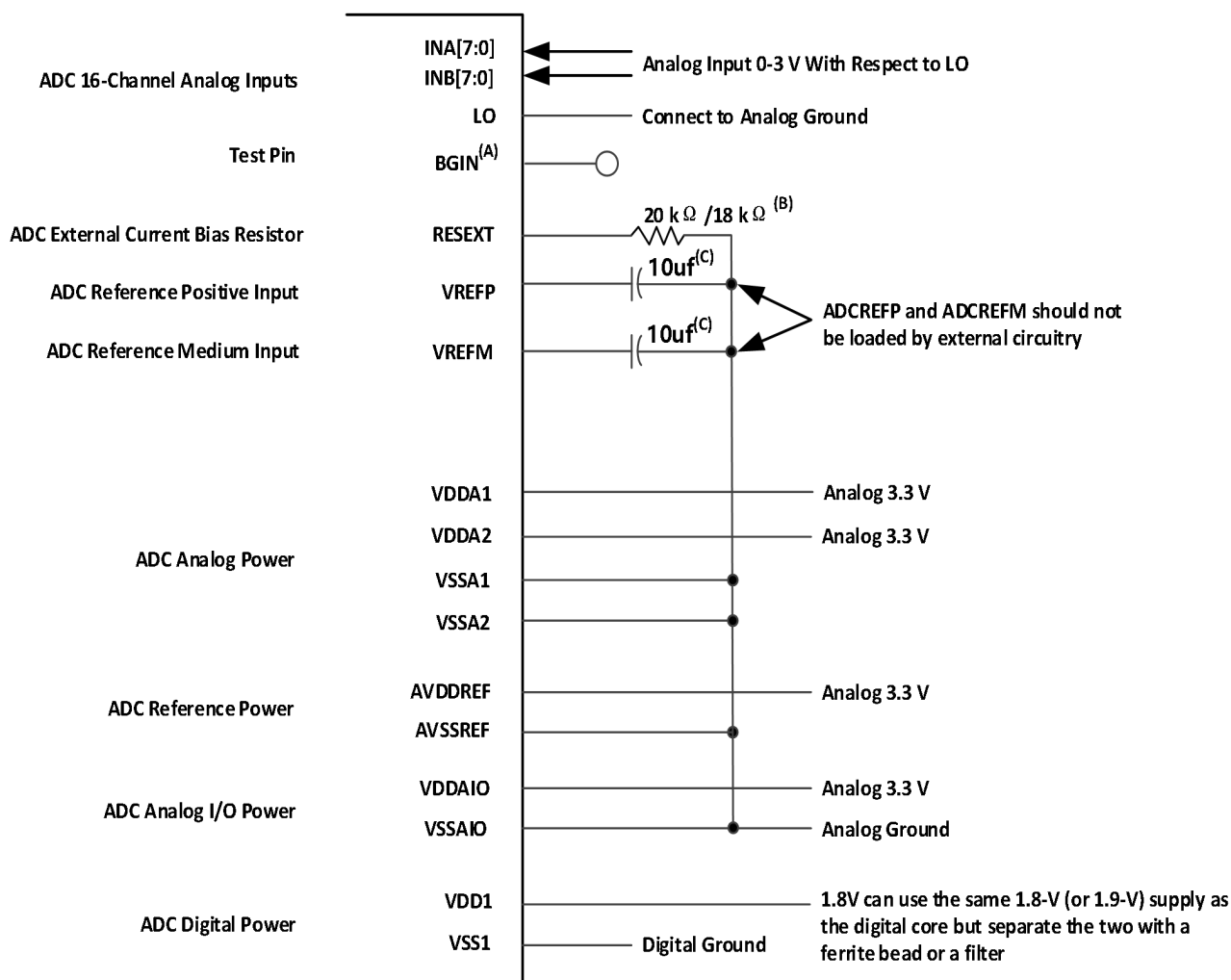


图3.3-2. 与内部基准的ADC 引脚连接

- A. 在PCB 布局中提供到这个引脚的访问。只用于测试用途。
- B. 对于1-18.75MHz 的时钟范围使用20kΩ，对于18.75-25MHz 的时钟范围使用18kΩ。
- C. TAIYO YUDEN EMK325F106ZH, EMK325BJ106MD，或者等效陶瓷电容器。
- D. 建议在所有电源引脚上使用外部去耦合电容器。
- E. 必须从不会降低ADC性能的运算放大器上驱动模拟输入。

注:

任何推荐组件的额定温度必须与最终产品的额定值相匹配。

ADP32FXX Digital Signal Processor

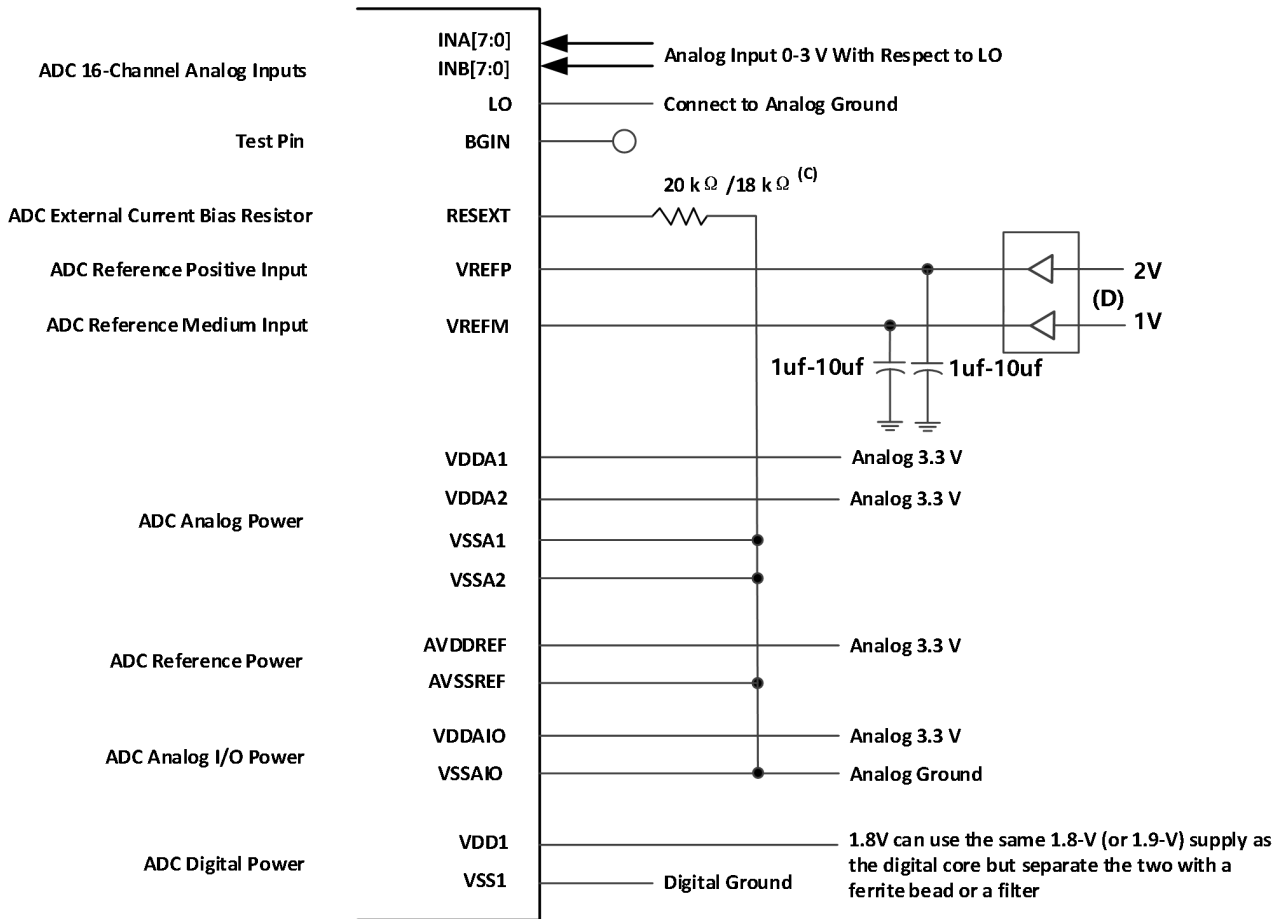


图3.3-3. 与外部基准的ADC 引脚连接

- A. 建议在所有电源引脚上使用外部去耦合电容器。
- B. 必须从不会降低ADC性能的运算放大器上驱动模拟输入。
- C. 对于1-18.75MHz 的时钟范围使用20kΩ，对于18.75-25MHz 的时钟范围使用18kΩ。
- D. 建议将经缓冲外部基准与一个 (VREFP-VREFM)= 1V±0.1% 或者更好的电压差异一起提供。

在ADC 加电时，使用ADCTRL3寄存器中的位3 可启用外部基准。在这个模式中，外部基准的精度对于总体增益十分关键。电压VREFP-VREFM将确定总体精度。当外部基准被连接至VREFP 和VREFM时，不要启用内部基准。

ADC 运行由表3-4中所列寄存器配置、控制、和监控。

表3-4. ADC 寄存器(1)

名称	地址	大小 (x16)	说明
ADCTRL1	0x00 7100	1	ADC 控制寄存器 1
ADCTRL2	0x00 7101	1	ADC 控制寄存器 2
ADCMAXCONV	0x00 7102	1	ADC 最大转换信道数寄存器
ADCCHSELSEQ1	0x00 7103	1	ADC 信道选择定序控制寄存器 1
ADCCHSELSEQ2	0x00 7104	1	ADC 信道选择定序控制寄存器 2
ADCCHSELSEQ3	0x00 7105	1	ADC 信道选择定序控制寄存器 3
ADCCHSELSEQ4	0x00 7106	1	ADC 信道选择定序控制寄存器 4
ADCASEQSR	0x00 7107	1	ADC 自动定序状态寄存器
ADCRESULT0	0x00 7108	1	ADC 转换结果缓冲寄存器 0
ADCRESULT1	0x00 7109	1	ADC 转换结果缓冲寄存器 1
ADCRESULT2	0x00 710A	1	ADC 转换结果缓冲寄存器 2
ADCRESULT3	0x00 710B	1	ADC 转换结果缓冲寄存器 3
ADCRESULT4	0x00 710C	1	ADC 转换结果缓冲寄存器 4
ADCRESULT5	0x00 710D	1	ADC 转换结果缓冲寄存器 5
ADCRESULT6	0x00 710E	1	ADC 转换结果缓冲寄存器 6
ADCRESULT7	0x00 710F	1	ADC 转换结果缓冲寄存器 7
ADCRESULT8	0x00 7110	1	ADC 转换结果缓冲寄存器 8
ADCRESULT9	0x00 7111	1	ADC 转换结果缓冲寄存器 9
ADCRESULT10	0x00 7112	1	ADC 转换结果缓冲寄存器 10
ADCRESULT11	0x00 7113	1	ADC 转换结果缓冲寄存器 11
ADCRESULT12	0x00 7114	1	ADC 转换结果缓冲寄存器 12
ADCRESULT13	0x00 7115	1	ADC 转换结果缓冲寄存器 13
ADCRESULT14	0x00 7116	1	ADC 转换结果缓冲寄存器 14
ADCRESULT15	0x00 7117	1	ADC 转换结果缓冲寄存器 15
ADCTRL3	0x00 7118	1	ADC 控制寄存器 3
ADCST	0x00 7119	1	ADC 状态寄存器
被保留	0x00 711C-0x00 711F	4	

(1) 上述寄存器是外设帧 2 寄存器。

3.4 增强型控制器局域网(eCAN) 模块

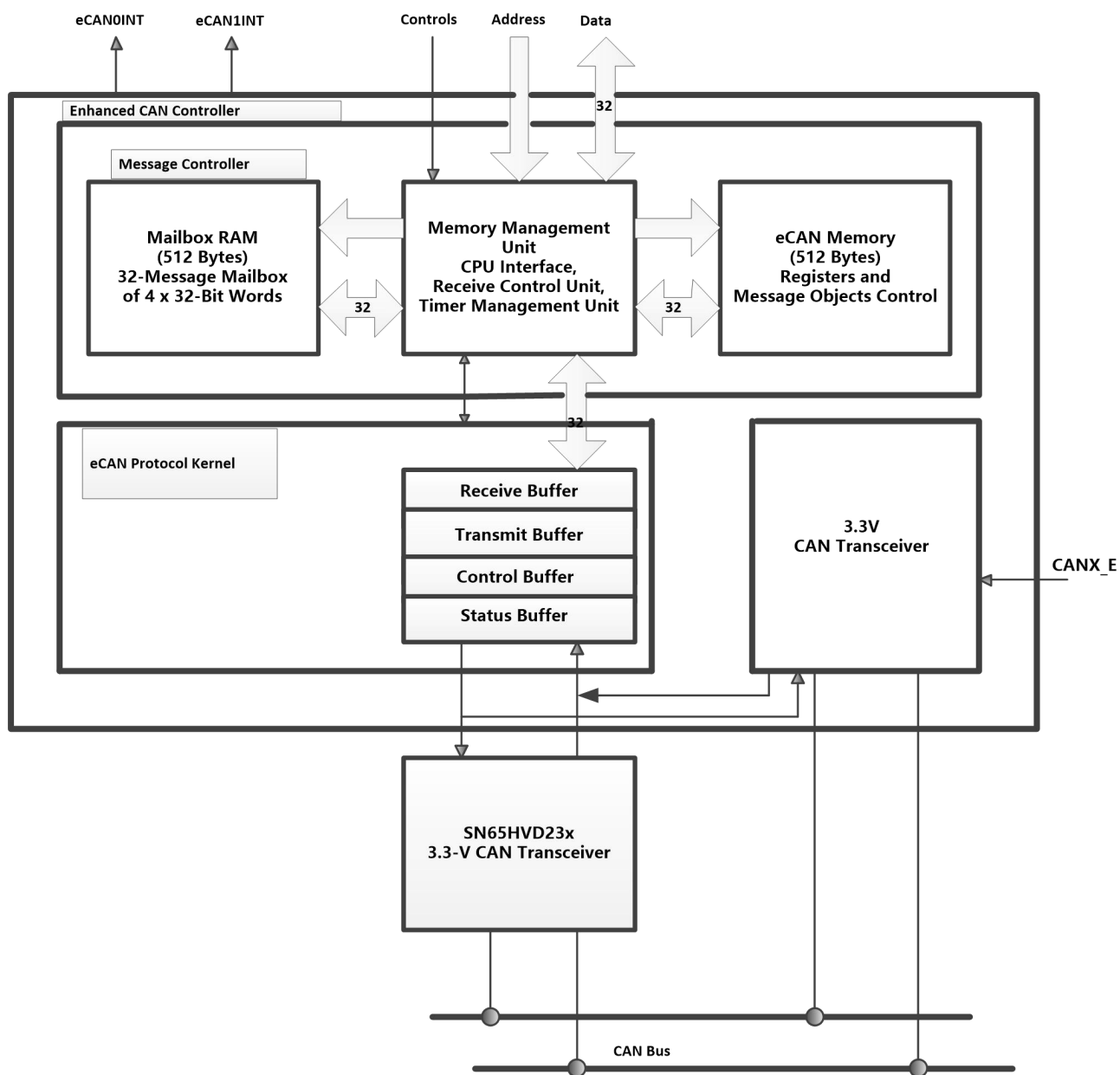


图 3.4-1. eCAN 方框图和接口电路

CAN 模块有下列特性：

- 与CAN 协议，版本2.0B 完全兼容
- 支持高达1Mbps 的数据速率
- 32 个邮箱，每一个邮箱有下列属性：
 - 可配置为接收或者发送
 - 可使用标准或者扩展标识符进行配置
 - 有一个可编辑接收屏蔽
 - 支持数据和远程帧

ADP32FXX Digital Signal Processor

- 由0 至8 字节数据组成
- 在接收和发送消息上使用一个32 位时间戳
- 防止接收新消息
- 保持发送消息的动态可编辑优先级
- 采用一个具有两个中断级别的可编辑中断机制
- 在发送或者接收超时采用一个可编辑报警

- **低功耗模式**

- **总线活动上的可编辑唤醒**

- **对远程请求消息的自动答复**

- **丢失仲裁或者错误情况下的帧自动重传**

- **由一个特定消息同步的32 位本地网络时间计数器（与邮箱6 协同通信）**

- **自测模式**

- 运行在接收自身消息的回路模式。提供一个“假”确认，从而消除了对于另外节点提供确认位的需要。

• **内部CAN收发驱动器和外部CAN收发驱动同一时间内只能有一个工作（在某些应用场合，用户可以同时使用片内收发器和片外收发器配合1个eCAN协议处理器构成伪双路eCAN链路,片内eCAN收发器的电气参数规范参见文档“ADP32FXX片内eCAN收发器功能及电参数规范”）**

注释：对于150MHz 的SYSCLKOUT，最小可能的比特率为23.4kbps，ADP32FXX CAN 已经通过了ISO/DIS 16845 的符合性测试。

表 3-5. 3.3V eCAN 收发器用于 ADP32FXX

部件型号	电源电压	低功耗模式	斜坡控制	VREF	其它	T _A
SN65HVD230	3.3V	待机	可调节	支持	-	-40°C 至 85°C
SN65HVD230Q	3.3V	待机	可调节	支持	-	-40°C 至 125°C
SN65HVD231	3.3V	睡眠	可调节	支持	-	-40°C 至 85°C
SN65HVD231Q	3.3V	睡眠	可调节	支持	-	-40°C 至 125°C
SN65HVD232	3.3V	无	无	无	-	-40°C 至 85°C
SN65HVD232Q	3.3V	无	无	无	-	-40°C 至 125°C
SN65HVD233	3.3V	待机	可调节	无	诊断回路	-40°C 至 125°C
SN65HVD234	3.3V	待机和睡眠	可调节	无	-	-40°C 至 125°C
SN65HVD235	3.3V	待机	可调节	无	自动波特率回路 内置隔离 低传播延迟	-40°C 至 125°C
ISO1050	3.3V-5.5V	无	无	无	热关断 故障安全运行 主计时	-55°C 至 105°C
ADP32FXX 片内收发器	3.3V	无	无	无	NMOS 源跟随输出，无门锁机理	-55°C 至 125°C

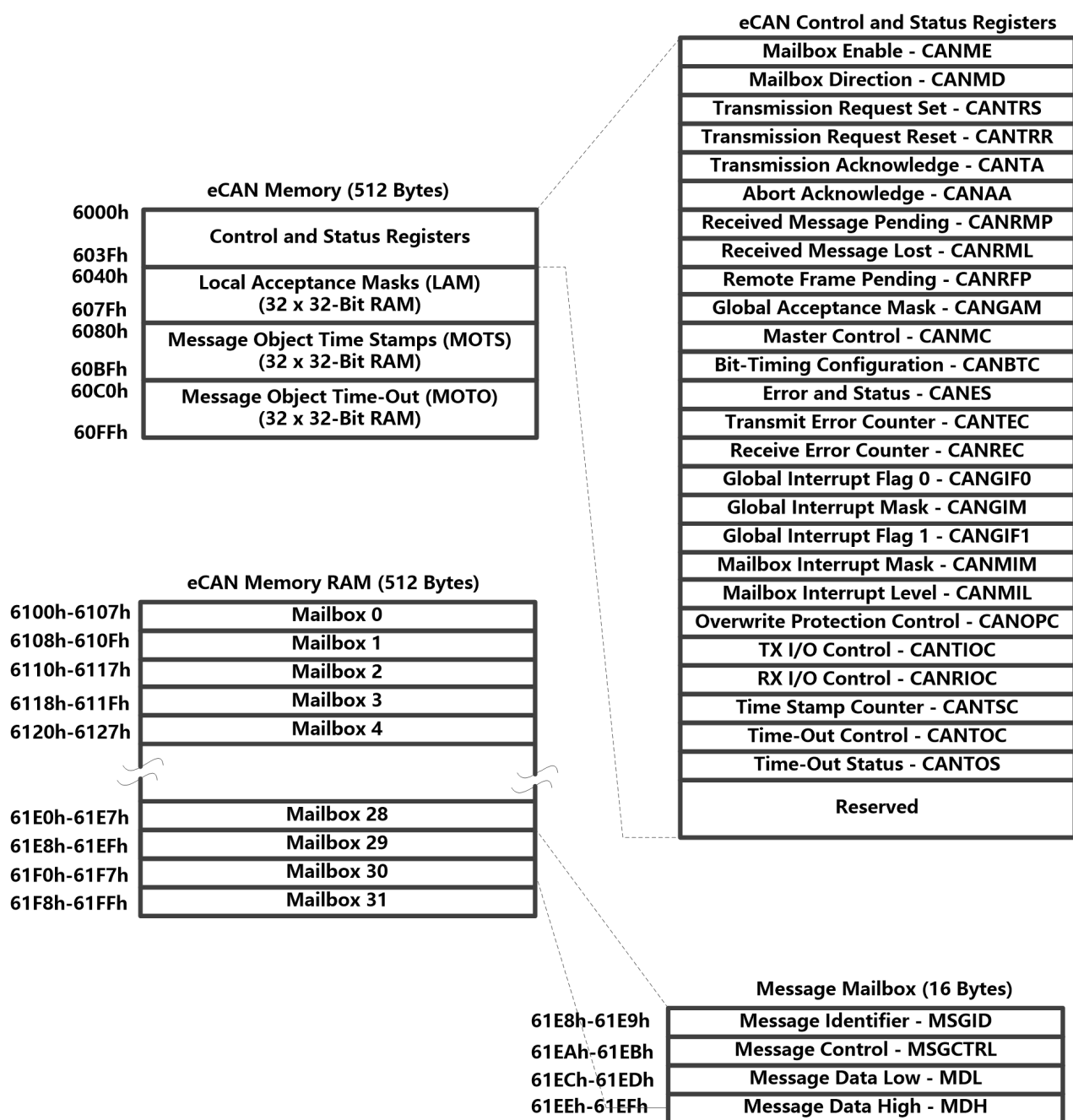


图3.4-2.eCAN内存映射

注:

如果eCAN 模块未在应用中使用，可用的RAM（LAM，MOTS，MOTO，和邮箱RAM）可被用作通用RAM。为实现这一功能CAN 模块时钟应被启用。

ADP32FXX Digital Signal Processor

表3-6中列出的CAN 寄存器由CPU 用于配置和控制CAN 控制器和消息目标。eCAN 控制寄存器只支持32位读取/写入操作。邮箱RAM 可进行16 位或者32 位访问。32 位访问与一个偶边界对齐。

表3-6. CAN 寄存器⁽¹⁾

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

(1)这些寄存器被映射至外设帧 1。

3.5 多通道缓冲串行端口(McBSP) 模块

McBSP 模块有以下特性：

- 与ADP32FXX DSP 器件内的McBSP 兼容，除DMA 功能之外
- 全双工通信
- 双缓冲数据寄存器，允许连续数据流
- 用于接收和发送的独立成帧和时钟
- 外部移位时钟生成或者一个内部可设定频率移位时钟
- 包括8，12，16，20，24，或者32 位在内的宽数据尺寸选择
- 以LSB 或者MSB 开头的8 位数据传输
- 用于帧同步和数据时钟的可编程极性
- 高度可编程内部时钟和帧生成
- 支持A-bis 模式
- 到工业标准CODEC、模拟接口芯片(AIC)、和其它串行连接的A/D 和D/A 器件的直接接口
- 与SPI 兼容器件一起工作
- 用于发送通道的两个16 x 16 级FIFO
- 用于接收通道的两个16 x 16 级FIFO

McBSP 上支持下列应用接口：

- T1/E1 成帧器
- MVIP 开关兼容和ST-BUS 兼容器件包括：
 - MVIP 成帧器
 - H.100 成帧器
 - SCSA 成帧器
 - 符合IOM-2 的器件
 - AC97 - 兼容器件（提供所需的多相位帧同步功能。）
 - IIS - 兼容器件
- McBSP 时钟速率 = $CLKG = CLKSRG / (1 + CLKGDIV)$ ，在这里CLKSRG 源应该为LSPCLK，CLKX，或者CLKR。⁽²⁾

(2) 串行端口性能受到I/O 缓冲器开关速度的影响。内部预分频器必须被调整，这样，外设速度将低于I/O 缓冲器速度限制- 最大值20MHz。

ADP32FXX Digital Signal Processor

图3-5显示了带有FIFO的McBSP模块的方框图，以及到外设帧2 ADP32FXX版本的接口。

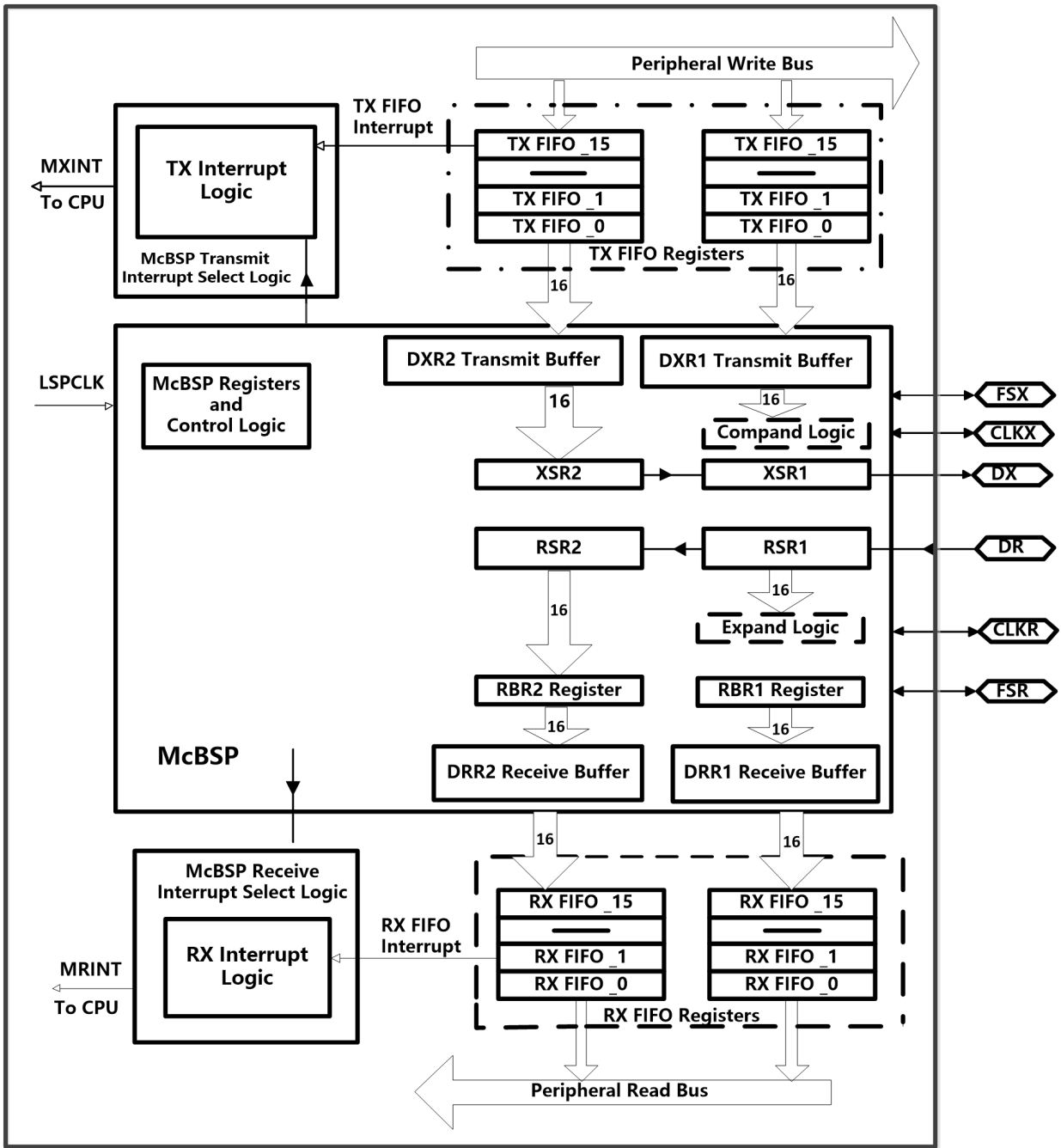


图3-5. 带有FIFO的McBSP模块

表3-7提供了McBSP 寄存器的汇总

表3-7. McBSP 寄存器

名称	地址 0x00 78xxh	类型 (R/W)	复位值 (十六进制)	说明
数据寄存器、接收、发送 ⁽¹⁾				
-	-	-	0x0000	MsBSP 接收缓冲寄存器
-	-	-	0x0000	McBSP 接收位移寄存器
-	-	-	0x0000	McBSP 发送移动寄存器
DRR2	00	R	0x0000	McBSP 数据接收寄存器2 • 如果字的大小大于16位, 读取第一个或者忽略DDR2
DRR1	01	R	0x0000	McBSP 数据接收寄存器1 • 如果字的大小大于16位, 读取第二个或者只读取DDR1
DXR2	02	W	0x0000	McBSP 数据发送寄存器2 • 如果字的大小大于16位, 写入第一个或者忽略DXR2
DXR1	03	W	0x0000	McBSP 数据发送寄存器1 • 如果字的大小大于16位, 写入第二个或者之写入DXR1
McBSP 控制寄存器				
SPCR2	04	读/写	0x0000	McBSP 串行端口控制寄存器2
SPCR1	05	读/写	0x0000	McBSP 串行端口控制寄存器1
RCR2	06	读/写	0x0000	McBSP 接收控制寄存器2
RCR1	07	读/写	0x0000	McBSP 接收控制寄存器1
XCR2	08	读/写	0x0000	McBSP 发送控制寄存器2
XCR1	09	读/写	0x0000	McBSP 发送控制寄存器1
SRGR2	0A	读/写	0x0000	McBSP 采样率发生器寄存器2
SRGR1	0B	读/写	0x0000	McBSP 采样率发生器寄存器1
多通道控制寄存器				
MCR2	0C	读/写	0x0000	McBSP 多通道寄存器2
MCR1	0D	读/写	0x0000	McBSP 多通道寄存器1
RCERA	0E	读/写	0x0000	McBSP 接收通道使能寄存器分区A
RCERB	0F	读/写	0x0000	McBSP 接收通道使能寄存器分区B
XCERA	10	读/写	0x0000	McBSP 发送通道使能寄存器分区A
XCERB	11	读/写	0x0000	McBSP 发送通道使能寄存器分区B
PCR	12	读/写	0x0000	McBSP 引脚控制寄存器
RCERC	13	读/写	0x0000	McBSP 接收通道使能寄存器分区C
RCERD	14	读/写	0x0000	McBSP 接收通道使能寄存器分区D
XCERC	15	读/写	0x0000	McBSP 发送通道使能寄存器分区C
XCERD	16	读/写	0x0000	McBSP 发送通道使能寄存器分区D
RCERE	17	读/写	0x0000	McBSP 接收通道使能寄存器分区E
RCERF	18	读/写	0x0000	McBSP 接收通道使能寄存器分区F
XCERE	19	读/写	0x0000	McBSP 发送通道使能寄存器分区E
XCERF	1A	读/写	0x0000	McBSP 发送通道使能寄存器分区F
RCERG	1B	读/写	0x0000	McBSP 接收通道使能寄存器分区G
RCERH	1C	读/写	0x0000	McBSP 接收通道使能寄存器分区H
XCERG	1D	读/写	0x0000	McBSP 发送通道使能寄存器分区G
XCERH	1E	读/写	0x0000	McBSP 发送通道使能寄存器分区H

(1) 在FIFO 模式中, DRR2/DRR1 和DXR2/DXR1 共用接收和发送FIFO 寄存器的同一地址。

表3-7. McBSP 寄存器(续)

名称	地址 0x00 78xxh	类型 (R/W)	复位值 (十六进制)	说明
FIFO 模式寄存器 (只适用于FIFO 模式)				
FIFO 数据寄存器(2)				
DRR2	00	R	0x0000	McBSP 数据接收寄存器2- 接收FIFO 的顶部 • 读取第一个FIFO 指针将不会提前
DRR1	01	R	0x0000	McBSP 数据接收寄存器1- 接收FIFO 的顶部 • 提前读取第二个FIFO 指针
DXR2	02	W	0x0000	McBSP 数据发送寄存器2- 发送FIFO 的顶部 • 写入第一个FIFO 指针将不会提前
DXR1	03	W	0x0000	McBSP 数据发送寄存器1- 发送FIFO 顶部 • 提前写入第二个FIFO 指针
FIFO 控制寄存器				
MFFTX	20	读/写	0xA000	McBSP 发送FIFO 寄存器
MFFRX	21	读/写	0x201F	McBSP 接收FIFO 寄存器
MFFCT	22	读/写	0x0000	McBSP FIFO 控制寄存器
MFFINT	23	读/写	0x0000	McBSP FIFO 中断寄存器
MFFST	24	读/写	0x0000	McBSP FIFO 状态寄存器

(2)FIFO 指针提前是基于到DDR2/DDR1 和DXR2/DXR1 寄存器的访问顺序。

3.6 串行通信接口(SCI) 模块

ADP32FXX器件包括两个串行通信接口(SCI) 模块。SCI 模块支持CPU 与其它异步外设之间的使用标准非归零码(NRZ) 格式的数字通信。SCI 接收器和发送器是双缓冲的，并且它们中的每一个有其自身独立的使能和中断位。两个器件都可独立或者同时地运行在全双工模式。为了确保数据完整性，SCI 在中断检测、奇偶性、超载、和组帧错误方面对接收到的数据进行检查。通过一个16 位波特率选择寄存器，可将比特率设定为超过65000 种不同的波特率。

每个SCI 模块的特性包括：

- 两个外部引脚：
 - SCITXD：SCI 发送-输出引脚
 - SCIRXD：SCI 接收-输入引脚

注释：两个引脚如果不被用于SCI 的话，可被用作GPIO。

- 波特率被设定为64K 个不同速率⁽³⁾

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

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

- 数据-字格式

- 一个开始位
- 数据-字长度可被设定为1 至8 位
- 可选偶/奇/无奇偶校验位
- 一个或者两个停止位

- 四个错误检测标志：奇偶、超载、组帧、和中断检测

- 两个唤醒多处理器模式：空闲线路和地址位

- 半双工或者全双工运行

- 双缓冲接收和发送功能

- 可通过带有状态标志的中断驱动或者轮询算法来完成发射器和接收器操作。

- 发送器：TXRDY 标志（发射器缓冲寄存器已经准备好接收另外字符）和TX EMPTY（TX 空）标志（发送器移位寄存器已空）

- 接收器：RXRDY 标志（接收器缓冲寄存器已经准备好接收另外的字符），BRKDT 标志（发生了中断条件），和RX ERROR 错误标志（监控四个中断条件）

- 用于发送器和接收器中断的独立使能位（除了BRKDT）

- 最大位速率= $75\text{MHz}/16 = 4.688 \times 10^6\text{b/s}$

- NRZ（非归零）码格式

- SCI 模块控制寄存器位于开始地址为7050h 的控制寄存器帧内

注释：所有处于这个模式中的寄存器是被连接至外设帧2 的8 位寄存器。当一个寄存器被访问时，低字节(7-0)，和高字节(15-8) 内的寄存器数据被读作零。对高字节的写入没有效果。

增强型特性：

- 自动波特率检测硬件逻辑电路

- 16 级发送/接收FIFO

(3) 串行端口性能受到I/O 缓冲器开关速度的影响。内部预分频器必须被调整，这样，外设速度将低于I/O 缓冲器速度限制- 20MHz 最大值。

SPI 端口运行由表3-8和表3-9中列出的寄存器配置和控制。

表3-8. SCI-A 寄存器

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

(1)这些寄存器是用于FIFO 模式的全新寄存器。

表3-9. SCI-B 寄存器⁽¹⁾

名称	地址	大小 (x16)	说明
SCICCRB	0x00 7750	1	SCI-B 通信控制寄存器
SCICTL1B	0x00 7751	1	SCI-B 控制寄存器1
SCIHBAUDB	0x00 7752	1	SCI-B 波特率寄存器, 高位
SCILBAUDB	0x00 7753	1	SCI-B 波特率寄存器, 低位
SCICTL2B	0x00 7754	1	SCI-B 控制寄存器2
SCIRXSTB	0x00 7755	1	SCI-B 接收状态寄存器
SCIRXEMUB	0x00 7756	1	SCI-B 接收仿真数据缓冲寄存器
SCIRXBUFB	0x00 7757	1	SCI-B 接收数据缓冲寄存器
SCITXBUFB	0x00 7759	1	SCI-B 发送数据缓冲寄存器
SCIFFTXB ⁽²⁾	0x00 775A	1	SCI-B FIFO 发送寄存器
SCIFFRXB ⁽²⁾	0x00 775B	1	SCI-B FIFO 接收寄存器
SCIFFCTB ⁽²⁾	0x00 775C	1	SCI-B FIFO 控制寄存器
SCIPRIB	0x00 775F	1	SCI-B 优先级控制寄存器

(1)这个表中的寄存器被映射到外设总线16 空间。这空间只允许16 位访问。32 位访问会产生未定义的后果。

(2) 这些寄存器是用于FIFO 模式的全新寄存器。

ADP32FXX Digital Signal Processor

图3-6显示了SCI 模块方框图。

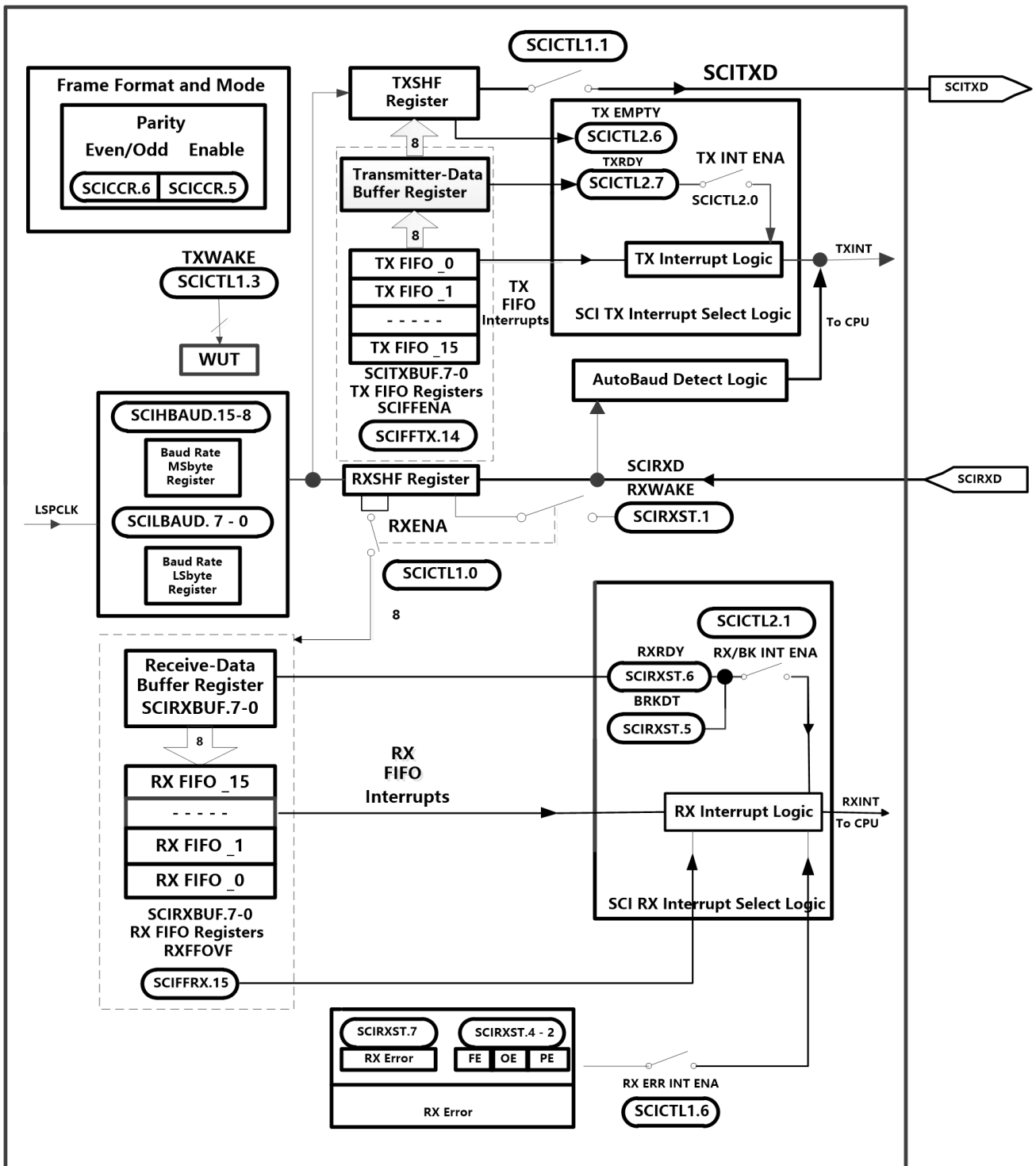


图3-6. 串行通信接口(SCI) 模块方框图

3.7 串行外设接口(SPI) 模块

ADP32FXX器件包括四引脚串行外设接口(SPI) 模块。SPI 是一个高速、同步串行I/O 端口，此端口可在设定的位传输速率上将一个设定长度（1 至16 位）的串行比特流移入和移出器件。通常，SPI 用于DSP 和外部外设或者其它处理器之间的通信。典型应用包括外部I/O 或者从诸如移位寄存器、显示驱动器、和ADC 等器件的外设扩展。多器件通信由SPI 的主控/受控操作支持。

SPI 模块的特性包括：

•四个外部引脚：

- F1：SPI 从器件输出/主器件输入引脚
- F0：SPI 从器件输入/主器件输出引脚
- F3：SPI 从器件发送使能引脚
- F2：SPI 串行时钟引脚

注释：如果SPI 模块未被使用，所有四个引脚可被用作GPIO。

•两个运行模式：主控和受控

•波特率：125 个不同的可编辑速率

$$\text{Baud rate} = \frac{\text{LSPCLK}}{(\text{SPIBRR}+1)} \text{ when SPIBRR} \neq 0$$
$$= \frac{\text{LSPCLK}}{4} \text{ when SPIBRR} = 0, 1, 2, 3$$

串行端口性能受到I/O 缓冲器开关速度的影响。内部预分频器必须被调整，这样，外设速度将低于I/O缓冲器速度限制- 20MHz 最大值。

•数据字长度：1~16数据位

•包括4个计时机制（由时钟极性和时钟相位的位控制）：

- 无相位延迟的下降沿：F2高电平有效。SPI 在F2信号的下降沿上传送数据，而在F2信号的上升沿上接收数据。
- 有相位延迟的下降沿：F2高电平有效。SPI 在F2信号下降沿的一半周期之前发送数据，而在F2信号的下降沿上接收数据。
- 无相位延迟的上升沿：F2低电平无效。SPI 在F2信号的上升沿上发送数据，而在F2信号的下降沿上接收数据。
- 有相位延迟的上升沿：F2低电平无效。SPI 在F2信号下降沿之前的半个周期发送数据，而在F2信号的上升沿上接收数据。

•同时接收和发送操作（发送功能可在软件中被禁用）

•通过中断驱动或者轮询算法来完成发射器和接收器运行。

•9 个SPI 模块控制寄存器：位于控制寄存器内，帧开始地址7040h。

注释：这个模块中的所有寄存器是被连接至外设帧2 的16 位寄存器。当一个寄存器被访问时，低字节(7-0)，和高字节(15-8) 内的寄存器数据被读作零。对高字节的写入没有效果。

增强型特性：

ADP32FXX Digital Signal Processor

- 16 级发送/接收FIFO
- 经延迟的发送控制

SPI 端口运行由表4-10和中列出的寄存器配置和控制。

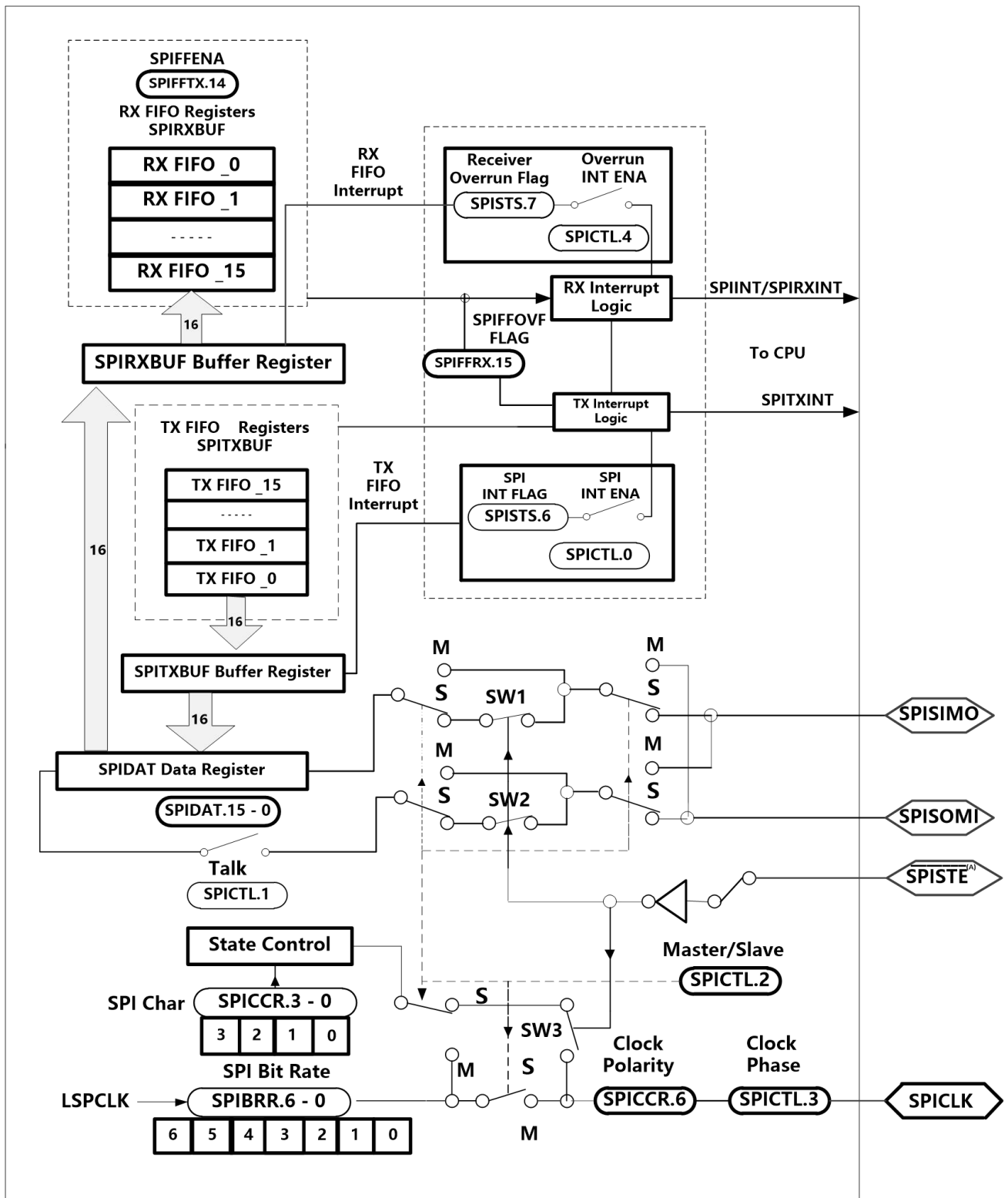
表3-10. SPI 寄存器⁽¹⁾

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

(1)这些寄存器被映射至外设帧2。这空间只允许16 位访问。32 位访问会产生未定义的后果。

ADP32FXX Digital Signal Processor

图3-7是一个处于受控模式下的SPI 的方框图。



A.F3 被主控器件驱动为用于受控器件的低电平。

图 3-7. 串行外设接口模块方框图 (受控模式)

3.8 GPIO MUX

GPIO 复用寄存器用于选择 ADP32FXX 器件上共用引脚操作。这个引脚可被独立选中运行为“数字 I/O”或者被连接至“外设 I/O”信号（通过 GPxMUX 寄存器）。如果被选择用于“数字 I/O”模式，提供的寄存器用来配置引脚方向（通过 GPxMUX 寄存器）并且限定输入信号来去除有害噪声（通过 GPxQUAL 寄存器）。表3-11列出了这些GPIO 复用寄存器。

表3-11. GPIO 复用寄存器^{(1) (2) (3)}

名称	地址	大小 (x16)	说明
GPAMUX	0x00 70C0	1	GPIO A 复用控制寄存器
GPADIR	0x00 70C1	1	GPIO A 方向控制寄存器
GPAQUAL	0x00 70C2	1	GPIO A 输入限定条件控制寄存器
被保留	0x00 70C3	1	
GPBMUX	0x00 70C4	1	GPIO B 复用控制寄存器
GPBDIR	0x00 70C5	1	GPIO B 方向控制寄存器
GPBQUAL	0x00 70C6	1	GPIO B 输入限定条件控制寄存器
被保留	0x00 70C7	1	
被保留	0x00 70C8	1	
被保留	0x00 70C9	1	
被保留	0x00 70CA	1	
被保留	0x00 70CB	1	
GPDMUX	0x00 70CC	1	GPIO 复用控制寄存器
GPDDIR	0x00 70CD	1	GPIO D 方向控制寄存器
GPDQUAL	0x00 70CE	1	GPIO D 输入限定条件控制寄存器
被保留	0x00 70CF	1	
GPEMUX	0x00 70D0	1	GPIO E 复用控制寄存器
GPEDIR	0x00 70D1	1	GPIO E 方向控制寄存器
GPEQUAL	0x00 70D2	1	GPIO E 输入限定条件控制寄存器
被保留	0x00 70D3	1	
GPFMUX	0x00 70D4	1	GPIO F 复用控制寄存器
GPFDIR	0x00 70D5	1	GPIO F 方向控制寄存器
被保留	0x00 70D6	1	
被保留	0x00 70D7	1	
GPGMUX	0x00 70D8	1	GPIO G 复用控制寄存器
GPGDIR	0x00 70D9	1	GPIO G 方向控制寄存器
被保留	0x00 70DA	1	
被保留	0x00 70DB	1	
被保留	0x00 70DC-	1	
	0x00 70DF		

(1) 被保留的位置返回未定义的值并且写入被忽略。

(2) 并不是所有输入都支持输入信号限定条件。

(3) 这些寄存器受EALLOW 保护。这防止假写入写覆盖内容并且损坏系统。

如果被配置为“数字 I/O”模式，提供一个附加的寄存器来设定独立 I/O 信号（通过 GPxSET 寄存器）、清除独立 I/O 信号（通过 GPxCLEAR 寄存器）、切换独立 I/O 信号（通过 GPxTOGGLE 寄存器）、或者读取/写入独立 I/O 信号（通过 GPxDAT 寄存器）。表3-12列出了GPIO 数据寄存器。

表3-12. GPIO 数据寄存器^{(1) (2)}

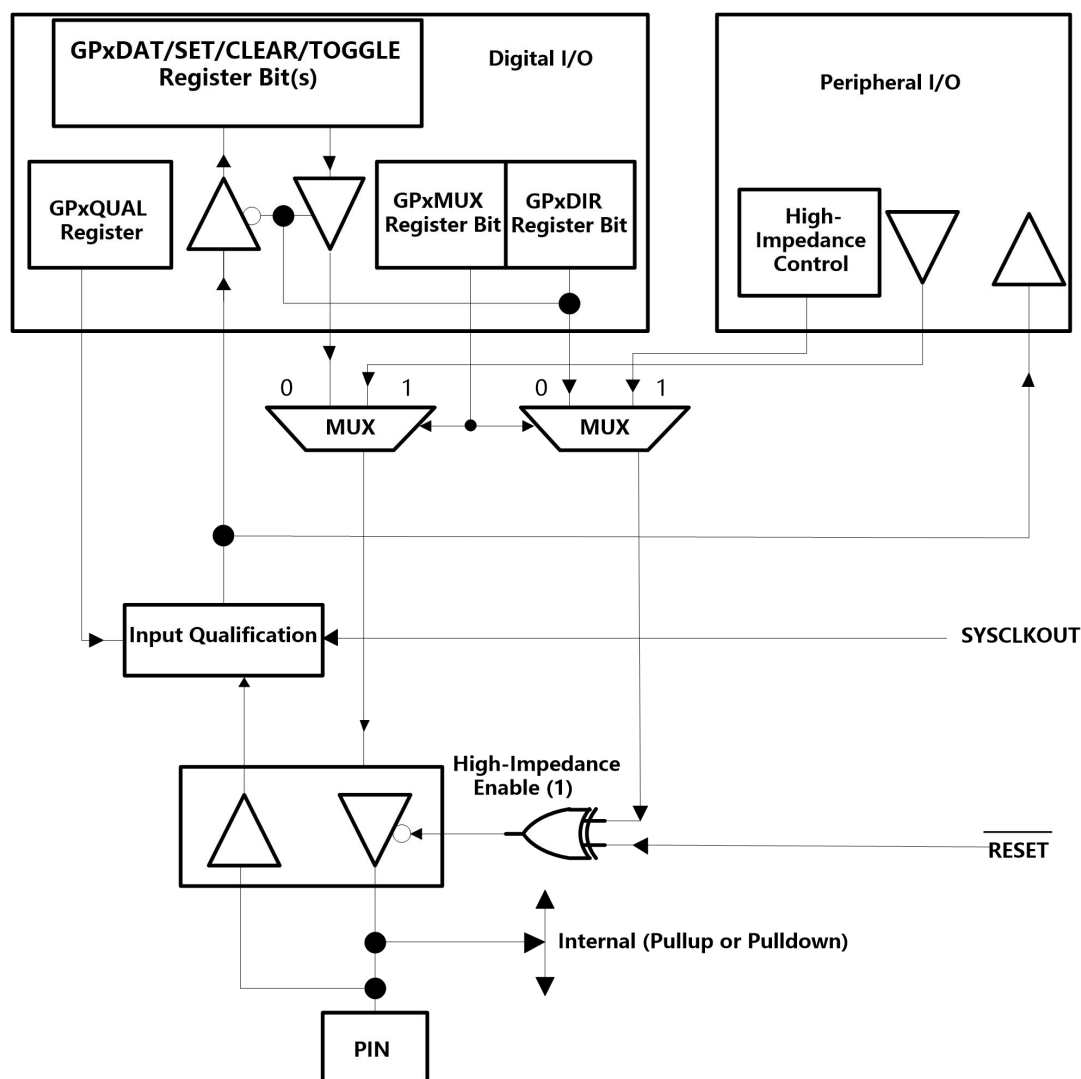
名称	地址	大小 (x16)	说明
GPADAT	0x00 70E0	1	GPIO A 数据寄存器
GPASET	0x00 70E1	1	GPIO A 设置寄存器
GPACLEAR	0x00 70E2	1	GPIO A 清零寄存器
GPATOGGLE	0x00 70E3	1	GPIO A 切换寄存器
GPBDAT	0x00 70E4	1	GPIO B 数据寄存器
GPBSET	0x00 70E5	1	GPIO B 设置寄存器
GPBCLEAR	0x00 70E6	1	GPIO B 清零寄存器
GPBTOGGLE	0x00 70E7	1	GPIO B 切换寄存器
被保留	0x00 70E8	1	
被保留	0x00 70E9	1	
被保留	0x00 70EA	1	
保留	0x00 70EB	1	
GPDDAT	0x00 70EC	1	GPIO D 数据寄存器
GPDSET	0x00 70ED	1	GPIO D 设置寄存器
GPDCLEAR	0x00 70EE	1	GPIO D 清零寄存器
GPDTOGGLE	0x00 70EF	1	GPIO D 切换寄存器
GPEDAT	0x00 70F0	1	GPIO E 数据寄存器
GPESET	0x00 70F1	1	GPIO E 设置寄存器
GPECLEAR	0x00 70F2	1	GPIO E 清零寄存器
GPETOGGLE	0x00 70F3	1	GPIO E 切换寄存器
GPFDAT	0x00 70F4	1	GPIO F 数据寄存器
GPFSET	0x00 70F5	1	GPIO F 设置寄存器
GPFCLEAR	0x00 70F6	1	GPIO F 清零寄存器
GPFTOGGLE	0x00 70F7	1	GPIO F 切换寄存器
GPGDAT	0x00 70F8	1	GPIO G 数据寄存器
GPGSET	0x00 70F9	1	GPIO G 设置寄存器
GPGCLEAR	0x00 70FA	1	GPIO G 清零寄存器
GPGTOGGLE	0x00 70FB	1	GPIO G 切换寄存器
被保留	0x00 70FC-	1	
	0x00 70FF		

(1) 被保留的位置将返回未定义的值并且写入操作将被忽略。

(2) 这些寄存器不受EALLOW 保护。用户通常定期访问上述寄存器。

ADP32FXX Digital Signal Processor

图3-8显示了不同寄存器位如何为GPIO 功能选择不同的操作模式。



- A.在GPIO模式中，当GPIO 引脚被配置为输出操作，对GPxDAT 数据寄存器的读取只能获得写入的值，而不是引脚上的值。在外设模式中，引脚的状态可通过GPxDAT 寄存器读取，前提是相应的方向位为零（输入模式）。
- B.一些选中的输入信号由SYCLKOUY限定。GPxQUAL 寄存器指定限定条件采样周期。采样窗口为6 样本宽并且只有当所有样本一致时（全0 或者全1），输出才被改变。这个特性从输入信号商去除了有害峰值。

图3-8. GPIO / 外设引脚复用

注:

GPIO 引脚的输入功能和到外设的输入路径一直启用。与主（外设）功能的输出路径复用的是GPIO 引脚的输出功能。由于一个引脚的输出缓冲器接回输入缓冲器，任何出现在此引脚上的GPIO 信号也将被传播至外设模块。因此，当一个引脚被配置为GPIO 运行时，相对应的外设功能（和中断生成功能）必须被禁用。否则，中断有可能在无意间被触发。当IOD0和IOD5引脚被用作GPIO 引脚时更是如此，这是因为GPDDAT.0 或者GPDDAT.5(PDPINTx) 的零值将会把PWM 引脚置于一个高阻抗状态。 $\overline{C_xTRIP}$ 和 T_xCTrip 引脚也会将相应的PWM 引脚置于高阻抗，前提是它们被驱动为低电平（作为GPIO 引脚）并且位EXTCONx.0=1。

3.9 片上 LDO 电源管理

ADP32FXX 片上集成 LDO 模块，引脚 LDOCTR_VSS 拉高（默认接地），使能片内 LDO，端口电压 V_{IO} 提供 LDO 输入源，通过配置 LDOCTR 寄存器，提供不同的内核电压， $V_{core_max} = 2.3V, V_{core_min} = 1.6V$;

表 3-14 LDOCTR 寄存器地址

名称	地址	大小 (x16)	说明
LDOCTR	0x000A92	1	LDO 控制寄存器寄存器

表 3-14 LDOCTR 寄存器地址与位定义

位	名称	类型	RESET	说明	
15:6	被保留	R=0	0:0		
				位值	
				000	V_{core} 1.9V
				001	1.8V
				010	2.0V
5:3	LDOCR	读/写	0,0,0	011	2.1V
				100	1.7V
				101	1.6V
				110	2.2V
				111	2.3V
2:0	被保留	R=0	0:0		

注：

片上 LDO 暂时只支持 ADP32F12，且在不同主频的工作环境下，0.02VDD 偏差。

当使用片上 LDO 时，电源 VDD 引脚需对地接滤波电容，建议每个 VDD 引脚外接 0.1uF~10uF 电容。

在对片上 Flash 进行擦除和编程操作会消耗较大 I_{DD} 流程，建议客户使用外置 V_{DD} 电源完成 Flash 的擦除与编程操作。

LDO 支持工作模式下的电压实时编程功能，根据 V_{DD} 上负载的不同，需要一定的稳定输出时间。较高的输出电压有利于获得更快的处理速度，较低的输出电压可显著改善电路的漏电流。

原厂不能保证客户的启用程序在全部电压范围内均能正常工作。客户必需仔细评价超出推荐启用电压范围带来的好处和风险。原厂不承担相应风险。

4 开发支持

软件开发工具

- Code Composer Studio™ 集成开发环境(IDE)
 - C/C++ 编译器
 - 代码生成工具
 - 汇编器/连接器
 - 周期精确模拟器
- 应用算法
- 示例应用代码

硬件开发工具

- ADP32F1XX eZdsp
- 支持基于JTAG 的仿真器- SPI515, XDS510PP, XDS510 USB, XDS100V2, XDS100V1
- 通用5V 直流电源, 推荐5V/1A以上
- 文档和线缆

5 电气规范

这个部分提供了针对 ADP32FXX 的绝对最大额定值和建议运行条件。

5.1 绝对最大额定值⁽¹⁾

电源电压范围(V_{DDIO} , V_{DDA1} , V_{DDA2} , V_{DDAIO} , 和 $AVDDREF$)	-0.3V 至 4.6V	
电源电压范围(V_{DD} , V_{DD1})	-0.5V 至 2.5V	
输入电压范围, V_{IN}	-0.3V 至 4.6V	
输出电压范围, V_O	-0.3V 至 4.6V	
输入钳制电流, $I_{IK}(V_{IN} < 0$ 或者 $V_{IN} > V_{DDIO})^{(2)}$	$\pm 20\text{mA}$	
输出钳制电流, $I_{OK}(V_O < 0$ 或者 $V_O > V_{DDIO})$	$\pm 20\text{mA}$	
运行环境温度范围, T_A	C 版本(BGA, QFP) ⁽³⁾	-40°C 至 85°C
	G 版本(BGA, QFP) ⁽³⁾	-40°C 至 125°C
	J 版本(BGA, QFP) ⁽³⁾	-55°C 至 125°C
结温范围, T_J	-60°C 至 150°C	
贮存温度范围, T_{stg} ⁽³⁾	-65°C 至 150°C	

(1) 除非另外说明, 绝对最大额定值的列表在运行温度范围内指定。在超过那些绝对最大额定值下列出的应力下运行有可能会对器件造成永久损坏。这些仅为在应力额定值下的工作情况, 对于额定值下的器件的功能性操作以及在超出Section 5.2推荐的操作条件下的任何其它操作, 在此并未说明。长时间运行在最大绝对额定条件下会影响设备的可靠性。所有电压值都是以 V_{SS} 为参考。

(2) 每个引脚上的持续钳制电流为 $\pm 2\text{mA}$ 。

(3) 长期高阻抗存储并且/或者在最大温度条件下长时间使用会使器件总体使用寿命的缩短。

5.2 建议的运行条件⁽¹⁾

		最小值	典型值	最大值	单位
V_{DDIO} 器件电源电压, I/O		3.14	3.3	3.47	V
V_{DD}, V_{DD1} 器件电源电压, CPU	1.8V (135MHz)	1.71	1.8	1.89	V
	1.9V (150MHz)	1.81	1.9	2	
V_{SS} 电源接地			0		V
V_{DDA1} , $V_{DDA2}, AVDDREF$, V_{DDAIO} , ADC 电源电压		3.14	3.3	3.47	V
$f_{SYSCLKOUT}$ 器件时钟频率 (系统时钟)	$V_{DD}=1.9\text{V} \pm 5\%$	2		150	MHz
	$V_{DD}=1.8\text{V} \pm 5\%$	2		135	
V_{IH} 高电平输入电压	除 OSC 之外的所有输入	2		V_{DDIO}	V
	OSC (@50 μA 最大值)	$0.7V_{DD}^{(3)}$		$V_{DD}^{(3)}$	V
V_{IL} 低电平输入电压	除 OSC 之外的所有输入			$V_{DDIO}^{(4)}$	V
	OSC (@50 μA 最大值)			0.8	V
					$0.8V_{DD}$
I_{OH} 高电平输出拉电流, $V_{OH}=2.4\text{V}$	除组 2 之外的所有 I/O			0.8 ⁽⁴⁾	mA
	组 2 ⁽²⁾				
I_{OL} 低电平输出灌电流 $V_{OL}=V_{OL}$ 最大值	除组 2 之外的所有 I/O			4(规范值), 5 (测量值)	mA
	组 2 ⁽²⁾			8(规范值), 10 (测量值)	
T_A 环境温度	A 版本	-40		85	°C
	S 版本	-40		125	

(1) 针对 V_{DDIO} , V_{DDAIO} , V_{DD} , $V_{DDA1}/V_{DDA2}/AVDDREF$ 。

(2) 组 2 引脚如下: XINTF 引脚, IOD0, TDO, CLKOUT, F14/PLDIS, MU0, 和 MU1。

(3) 输入信号为 0~ V_{DD} 电压域。(4) 输入信号为 0~ V_{DDIO} 电压域。

5.3 建议运行条件下的电气特性（除非另外注明）

参数		测试条件	最小值	典型值	最大值	单位	
V _{OH} 高电平输出电压		I _{OH} =I _{OH} 最大值	2.4			V	
		I _{OH} =50μA	V _{DDIO} -0.2				
V _{OL} 低电平输出电压		I _{OL} =I _{OL} 最大值			0.4	V	
I _{IL} 输入电流 (低电平)	带有上拉电阻器	V _{DDIO} =3.3V, V _{IN} =0V	-80	-140	-190	μA	
	带有下拉电阻器	V _{DDIO} =3.3V, V _{IN} =0V			±2		
I _{IL} 输入电流 (低电平)	带有上拉电阻器	V _{DDIO} =3.3V, V _{IN} =0V	除了 EM2 之外的所有 I/O ⁽¹⁾ (包括 XRS)	-80	-140	-190	μA
			GPIOB/EM2	-13	-25	-35	
	带有下拉电阻器	V _{DDIO} =3.3V, V _{IN} =0V			±2		
I _{IH} 输入电流 (高电平)	带有上拉电阻器	V _{DDIO} =3.3V, V _{IN} =V _{DD}			±2	μA	
	带有下拉电阻器 ⁽²⁾	V _{DDIO} =3.3V, V _{IN} =V _{DD}	28	50	80		
I _{OZ} 泄漏电流 (对于没有内部 PU/PD 的引脚), 高阻抗状态 (关闭状态)		V _O =V _{DDIO} 或者 0V			±2	μA	
C _i 输入电容				2		pF	
C _o 输出电容				3		pF	

(1) 以下的引脚没有内部PU/PD：INT1, INT2, F0, F1, F2, F3, F12, G4, G5。

(2) 以下引脚有一个内部下拉电阻器：MC, TESL, 和 TRS。

5.4 流耗

在 150MHz SYSCLKOUT 上的低功耗模式期间的推荐运行条件下，电源引脚的 ADP32FXX 流耗

模式	测试条件	I_{DD}		$I_{DDIO}^{(1)}$		$I_{DDA}^{(2)}$	
		典型值	最大值 ⁽³⁾	典型值	最大值 ⁽³⁾	典型值	最大值 ⁽³⁾
可用	所有外设时钟被启用。 所有PWM 引脚被切换至100MHz。数据在 SCIA, SCIB, 和CAN 端口上连续发出。硬件复用器被使用。正在闪存之外运行的代码具有 5 个等待状态。	150mA ⁽⁴⁾	200mA	10mA	20mA	35mA	45mA
IDLE	<ul style="list-style-type: none"> 闪存休眠。 CLKOUT 被关闭。 除了ADC 之外，所有外设时钟打开。 	100mA	130mA	3mA	8mA	1μA	10μA
STANDBY	<ul style="list-style-type: none"> 闪存休眠。 外设时钟被关闭。 没有一个内部 PU/PD 的引脚被接至高电平/低电平。 	5.7mA	12mA	79μA	200μA	1μA	10μA
HALT	<ul style="list-style-type: none"> 闪存休眠。 外设时钟被关闭。 没有一个内部 PU/PD 的引脚被接至高电平/低电平。 输入时钟被禁用。 	99.2μA		3.3μA	15μA	1μA	10μA

(1) I_{DDIO} 电流取决于 I/O 引脚上的电气负载。

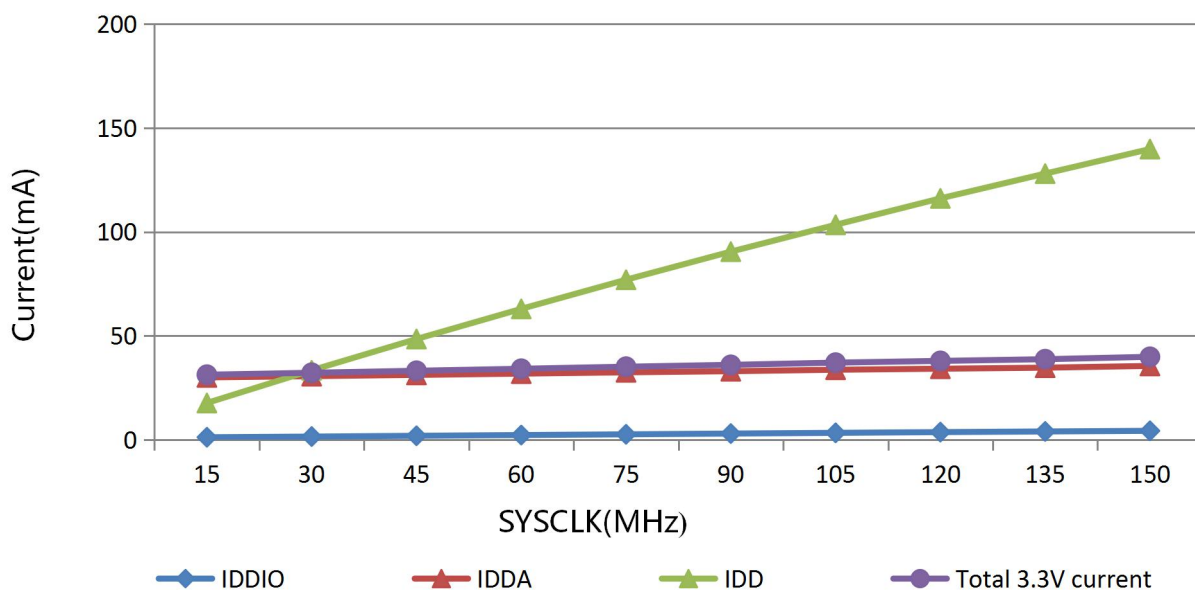
(2) I_{DDA} 包括进入 V_{DDA1} , V_{DDA2} , $AVDDREF$, 和 V_{DDAIO} 引脚的电流。

(3) 最大值条件：125°C 以及最大电压 ($V_{DD}=1.89V$; V_{DDIO} , $V_{DDA}=3.47V$)

(4) I_{DD} 代表取自 1.8V 电源轨 (V_{DD}) 的总电流。它包括少部分由 V_{DD1} 汲取的电流 (<1mA)。

注：当 PLL 被禁用时，HALT 和 STANDBY 模式不能使用。

5.5 流耗图



- A. 针对工作电流的测试条件在表5-1中进行了定义。
- B. I_{DD} 代表取自1.8V电源轨(V_{DD})的总电流。它包括少部分由 V_{DD1} 汲取的电流($<1\text{mA}$)。
- C. I_{DDA} 代表 V_{DDA1} 和 V_{DDA2} 电源轨汲取的电流。
- D. 3.3V总电流是 I_{DDIO} 和 I_{DDA} 的总和。它包括少部分由 V_{DDAIO} 汲取的电流($<1\text{mA}$)。

图 5-1. ADP32FXX 在频率范围内的典型功耗

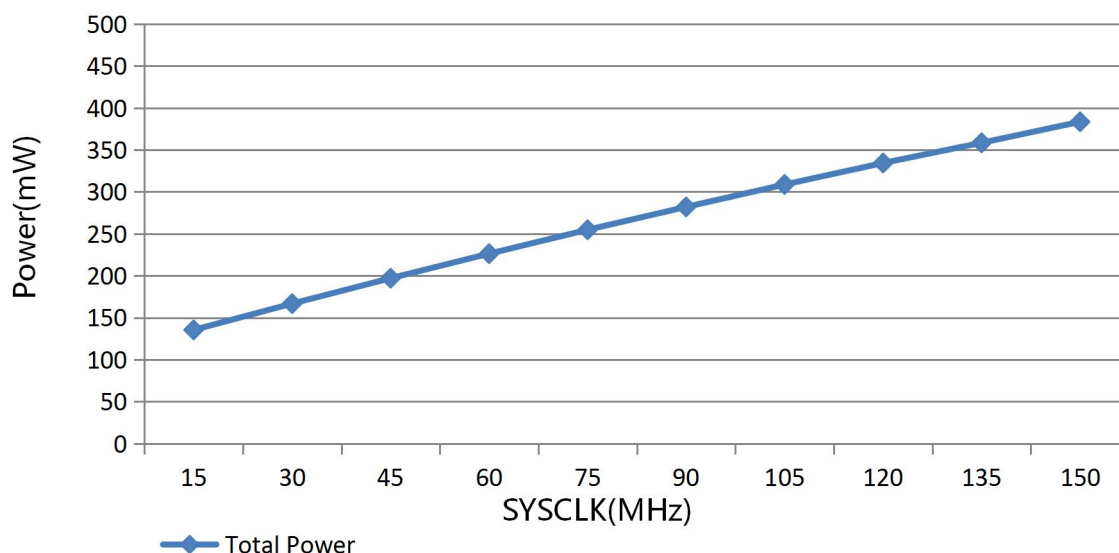


图 5-2. ADP32FXX 在频率范围内的典型流耗

5.6 减少流耗

ADP32FXXDSP 包含一个减少器件流耗的独特方法。通过关闭到任一不在一个指定应用中使用的外设模块的时钟可实现流耗减少。表5-3表明了通过关闭到不同外设的时钟所实现的流耗减少的典型值。

表 5-3. 不同外设的典型流耗 (150MHz 上时)⁽¹⁾

外设模块	I _{DD} 电流减少(mA)
eCAN	8
EM1	6
EM2	6
ADC	6 ⁽²⁾
SCIA	2
SCIB	2
SPI	5
McBSP	12

(1) 复位时，所有外设时钟被禁用。只有在外设时钟被打开后，才可进行对外设寄存器的写入/读取操作。

(2) 这个数字代表了取自ADC 模块数字部分的电流。关闭ADC 模块的时钟也将消除取自ADC (I_{DDA}) 模拟部分的电流。

5.7 针对 DSP 的无信号缓冲的仿真器连接

图5-5显示了DSP 和JTAG 接头之间针对单处理器配置的连接。如果JTAG 接头和DSP 之间的距离大于6英寸，那么仿真信号必须被缓冲。如果距离小于6英寸，通常无需缓冲。图5-5显示了较简单、无缓冲的情况。对于上拉/下拉电阻器的值，请见引脚说明部分。

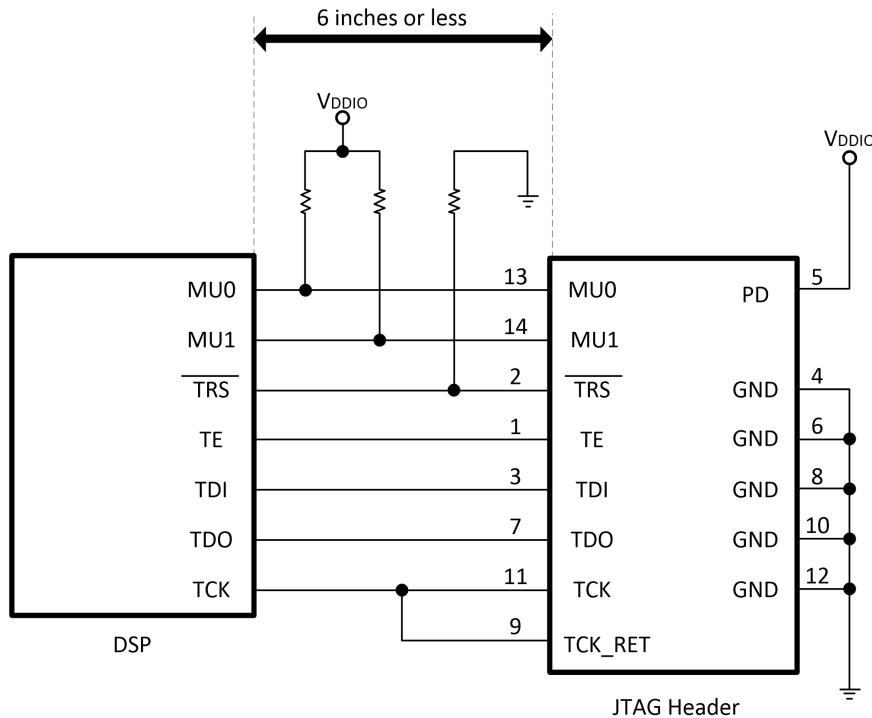


图 5-5. 针对 DSP 的无信号缓冲的仿真器连接

5.8 电源时序要求

ADP32FXX芯片要求双电压（1.8V 或1.9V 和3.3V）来为CPU，闪存，ROM，ADC，和I/O 供电。为了确保所有模块在加电期间的正常复位状态，在为器件加电/断电的同时需要满足一些要求。

ADP32FXX器件无需电源时序。换句话说，3.3V 和1.8V（或者1.9V）可以一起斜升；然而，如果1.8V（或者1.9V）电源轨落后于3.3V 电源轨，GPIO 引脚在1.8V 电源轨达到至少1V 前处于未定义状态。

- 选项1：

在这个方法中，一个外部电源排序电路首先启用 V_{DDIO} ，然后是 V_{DD} 和 V_{DD1} （1.8V 或者1.9V）。在1.8V（或者1.9V）斜升后，用于ADC($V_{DDA1}/V_{DDA2}/AVDDREF$) 模块的3.3V 被斜升。

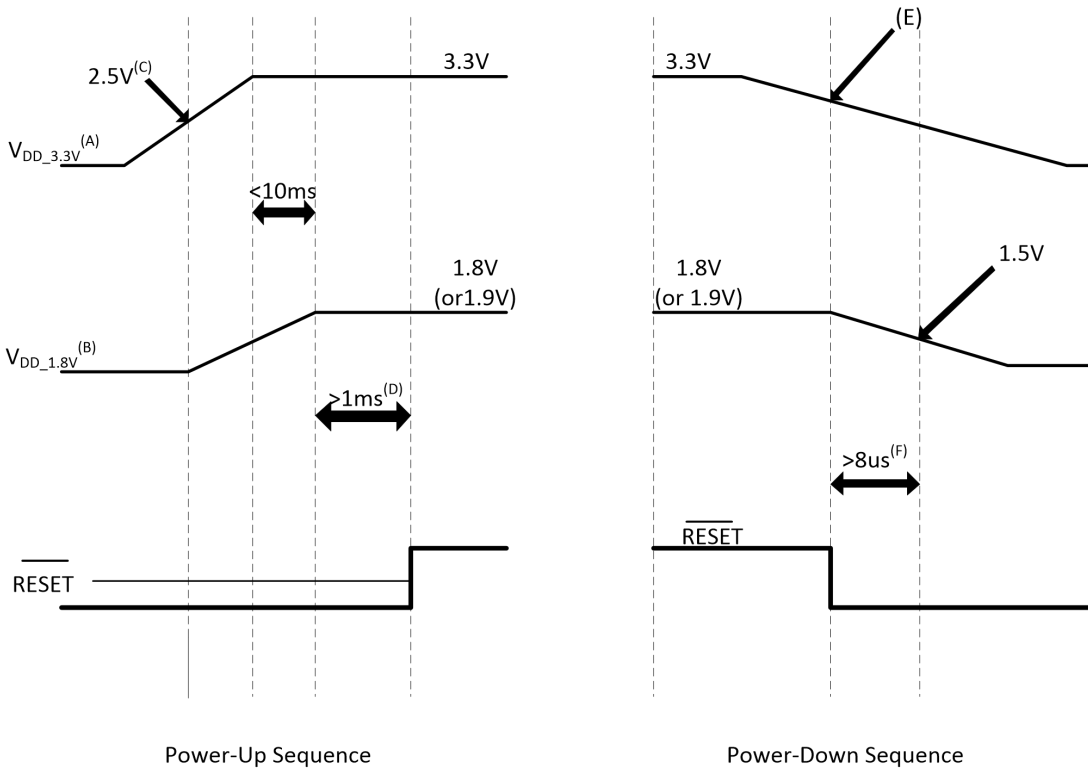
- 选项2：

启用到所有3.3V 电源引脚的电源(V_{DDIO} ， $V_{DDA1}/V_{DDA2}/V_{DDAIO}/AVDDREF$)，然后斜升1.8V（或者1.9V）(V_{DD}/V_{DD1}) 电源引脚。

在 V_{DDIO} 已经达到2.5V 之前，1.8V 或者1.9V (V_{DD}/V_{DD1}) 不应达到0.3V 这就确保了来自I/O 引脚的复位信号已经通过I/O 缓冲器传播出去，为所有器件内的模块提供加电复位。加电复位时序请见图5-11。

- 断电时序：

断电期间，在 V_{DD} 电源达到1.5V 之前，器件复位应该被置为低电平（8 μ s，最小值）。这将有助于在 V_{DDIO}/V_{DD} 电源斜降之后，将片载闪存逻辑保持在复位状态。建议使用“低压降(LDO)”稳压器或者电压监视器的器件复位控制用于满足这个约束条件。便利电源时序的LDO 稳压器（在额外外部组件的帮助下）可被用于，满足电源时序要求。



A. $V_{DD_3.3V}$ - V_{DDIO} , V_{DDAIO} , V_{DDA1} , V_{DDA2} , $AVDDREF$

B. $V_{DD_1.8V}$ - V_{DD} , V_{DD1}

C. 1.8V (或者1.9V) 电源应该在3.3V 电源至少达到2.5V 后斜升。

D. 在电源和时钟稳定前, 复位(RESET) 应该保持低电平。最小要求, 请见图5-11, 微计算机模式中的加电复位 ($\overline{MC}=0$)。

E. 当3.3V 电源稳压关闭时, 电压监控器或者LDO 复位控制将首先触发复位(RESET)。通常情况下, 这个操作在1.8V (或者1.9V) 电源至少达到1.5V 的几毫秒前发生。

F. 1.8V (或者1.9V) 电源达到1.5V 前, 在至少 $8\mu\text{s}$ 内保持复位低电平(RESET), 那么在电源斜降之前, 闪存模块将保持在完全复位。

G. 由于GPIO 引脚的状态在1.8V (或者1.9V) 电源至少达到1V 前未定义, 这个电源应该尽可能快的斜升 (在3.3V 电源至少达到2.5V 之后)。

H. 除了电源引脚, 在3.3V 电源轨已经被完全加电前, 不应驱动任何引脚。

图 5-6. ADP32FXX 典型加电和断电序列-选项 2

5.9 信号转换电平

请注意一些信号使用不同的基准电压, 请见推荐运行条件表。输出电平被驱动至一个2.4V 的最低逻辑高电平并被驱动至一个0.4V 的最大逻辑低电平。

图5-7显示了输出电平。



图 5-7. 输出电平

输出转换时间指定如下：

- 对于一个高电平到低电平的转换，输出上不再被认为是高电平的电平为低于 V_{OH} (最小值) 的电平，而输出上不再被认为是低电平的电平为 V_{OL} (最大值) 并且更低。
- 对于一个低电平到高电平转换，输出上不再被认为是低电平的电平为高于 V_{OL} (最大值) 的电平，而输出上不再被认为是高电平的电平为 V_{OH} (最小值) 并且更高。

图 5-8 显示了输入电平。



图 5-8. 输入电平

输入转换时间指定如下：

- 对于一个高电平到低电平的转换，输出上不再被认为是高电平的电平为低于 V_{IH} (最小值) 的电平，而输出上不再被认为是低电平的电平为 V_{IL} (最大值) 并且更低。
- 对于一个低电平到高电平转换，输出上不再被认为是低电平的电平为高于 V_{IL} (最大值) 的电平，而输出上不再被认为是高电平的电平为 V_{IH} (最小值) 并且更高。

NOTE

对于测试时序参数，请见单独的电平时序图。

5.10 时序参数符号

所用的时序参数符号按照 JEDEC 标准 100 创建。为了缩短符号，一些引脚的名称和其它相关的术语名已经按如下方法缩减：

小写下标和它们的含意：字母和符号和它们的含意：

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

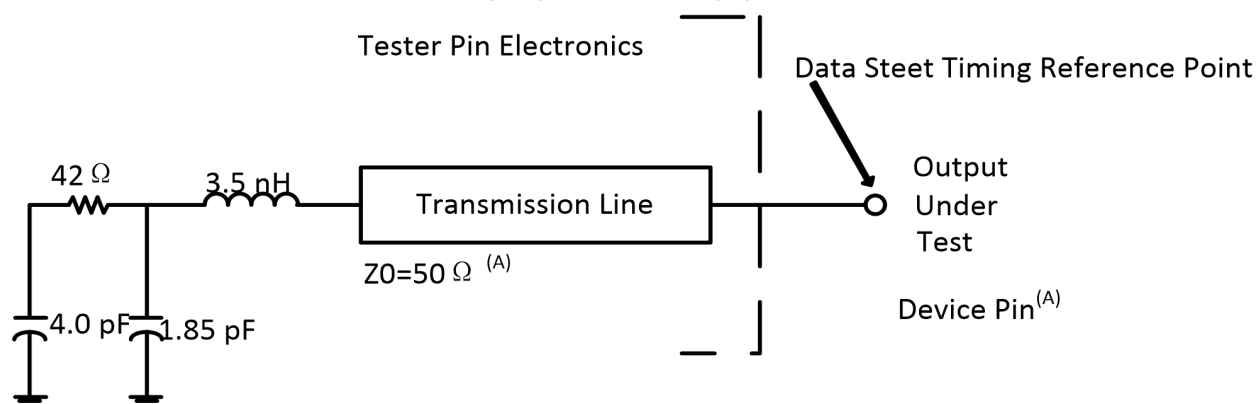
5.11 定时参数的通用注释

所有ADP32FXX器件的输出信号（包括CLKOUT）取自一个内部时钟，这样，对于一个指定半周期的所有输出转换在一个互相之间相对最小转换率时发生。

这个显示在下面时序图中的信号组合也许不一定代表真实的周期。对于真实周期范例，请参见本文档的合适周期说明部分。

5.12 测试负载电路

这个测试负载电路用于测量这个文档中提供的所有开关特性。



A. 此数据表在器件引脚上提供时序。对于输出时序分析，必须将测试器引脚电子特性和其传输线路效应考虑在内。带有2ns或者更长时间延迟的传输线路可被用于生成所需的传输线路效应。传输线路只用于一个负载。无需从数据表时序中增加或者减少传输线路延迟（2ns或者更长）。

使用一个器件引脚上 < 4伏特每纳秒(4V/ns) 的输入转换率对这个数据表中的输入要求进行测试。

图 5-9. 3.3V 测试负载电路

5.13 器件时序表

这个部分提供针对DSP 上可用的多种时钟选项的时序要求和开关特性。表5-5列出了多种时钟的周期时间。

表 5-5. ADP32FXX 时应表和命名规则

		最小值	标称值	最大值	单位
片载振荡器时钟	$t_{c(OSC)}$, 周期时间	28.6		50	ns
	频率	20		35	MHz
OSC	$t_{c(CI)}$, 周期时间	6.67		250	ns
	频率	4		150	MHz
SYSCLKOUT	$t_{c(SCO)}$, 周期时间	6.67		500	ns
	频率	2		150	MHz
CLKOUT	$t_{c(XCO)}$, 周期时间	6.67		2000	ns
	频率	0.5		150	MHz
HSPCLK	$t_{c(LCO)}$, 周期时间	6.67	13.3 ⁽¹⁾		ns
	频率		75 ⁽¹⁾	150	MHz
LSPCLK	$t_{c(LCO)}$, 周期时间	13.3	26.6 ⁽¹⁾		ns
	频率		37.5 ⁽¹⁾	75	MHz
ADC 时钟	$t_{c(ADCCLK)}$, 周期时间 ⁽²⁾	40			ns
	频率			25	MHz
SPI 时钟	$t_{c(SPC)}$, 周期时间	50			ns
	频率			20	MHz
McBSP	$t_{c(CKG)}$, 周期时间	50			ns
	频率			20	MHz
XTIMCLK	$t_{c(XTIM)}$, 周期时间	6.67			ns
	频率			150	MHz

(1) 如果SYSCLKOUT=150MHz，这个值为缺省复位值。

(2)ADCCLK 频率的最大值为25MHz。对于25MHz 或者更低的SYSCLKOUT 的值，ADCCLK 必须为SYSCLKOUT/2 或者更低。对于SYSCLKOUT 的任一值，ADCCLK=SYSCLKOUT 不是一个有效模式。

5.14 时钟要求和特性

5.14.1 输入时钟要求

OSC 引脚上提供的时钟生成内部 CPU 时钟周期。

表 5-6. 输入时钟频率

参数		最小值	典型值	最大值	单位
f _x 输入时钟频率	谐振器	20		35	MHz
	晶振	20		35	
	OSC				
		无 PLL	4	150	
		具有 PLL	5	100	
f _l 跛行模式时钟频率			2		MHz

表 5-7. OSC 时序要求- PLL 旁通或者被启用

编号		最小值	最大值	单位
C8	t _{c(CI)} 周期时间, OSC	6.67	250	ns
C9	t _{f(CI)} 下降时间, OSC		6	ns
C10	t _{r(CI)} 上升时间, OSC		6	ns
C11	t _{w(CIL)} 脉冲持续时间, OSC 低电平作为 t _{c(CI)} 的一部分的时间	40	60	%
C12	t _{w(CIH)} 脉冲持续时间, OSC 高电平作为 t _{c(CI)} 的一部分的时间	40	60	%

表 5-8. OSC 时序需求- PLL 被禁用

编号		最小值	最大值	单位
C8	t _{c(CI)} 周期时间, OSC	6.67	250	ns
C9	t _{f(CI)} 下降时间, OSC	高达 30 MHz	6	ns
		30MHz 至 150MHz	2	
C10	t _{r(CI)} 上升时间, OSC	高达 30 MHz	6	ns
		30MHz 至 150MHz	2	
C11	t _{w(CIL)} 脉冲持续时间, OSC 低电平作为 t _{c(CI)} 的一部分的时间	OSC ≤ 120MHz	40	%
		120 < OSC ≤ 150MHz	45	
C12	t _{w(CIH)} 脉冲持续时间, OSC 高电平作为 t _{c(CI)} 的一部分的时间	OSC ≤ 120MHz	40	%
		120 < OSC ≤ 150MHz	45	

表 5-9. 可能的 PLL 配置模式

PLL 模式	注释	SYSLKOUT
PLL 被禁用	通过复位时将 F14 引脚接至低电平来调用。PLL 块被完全禁用。到 CPU (CLKIN) 的时钟输入直接取自出现在 OSC 引脚的时钟信号。	OSC
PLL 被绕过	如果 PLL 未被禁用, 加电时 PLL 配置为缺省值。PLL 本身被旁通绕过。然而, 在馈入 CPU 之前, PLL 块内的/2 模块将这个 OSC 引脚上的时钟输入二分频。	OSC/2
PLL 被启用	通过将一个非零值“n”写入 PLLCR 寄存器实现。现在, 在馈入 CPU 之前, PLL 块内的/2 模块将 PLL 的输出二分频。	(OSC*n)/2

5.14.2 输出时钟特性

表 5-10. CLKOUT 开关特性 (PLL 被绕过或者被禁用)^{(1) (2)}

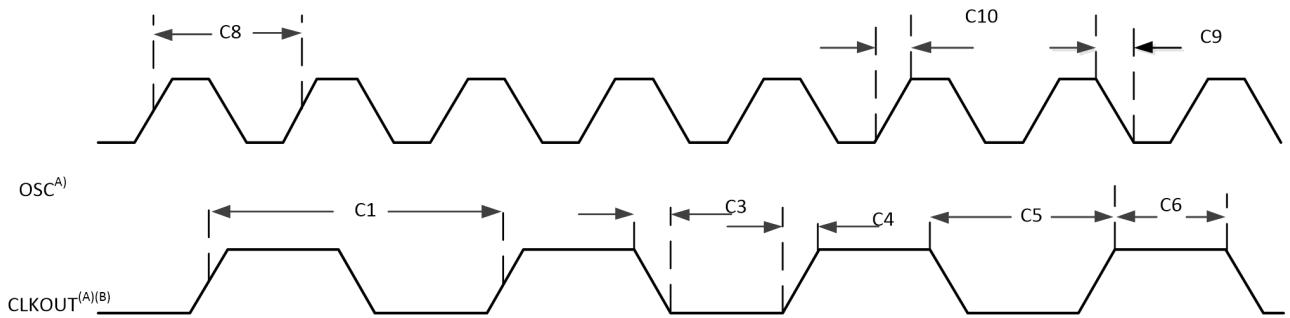
编号	参数	最小值	典型值	最大值	单位
C1	$t_{c(XCO)}$ 周期时间, CLKOUT	6.67 ⁽³⁾			ns
C3	$t_{f(XCO)}$ 下降时间, CLKOUT		2		ns
C4	$t_{r(XCO)}$ 上升时间, CLKOUT		2		ns
C5	$t_{w(XCOL)}$ 脉冲持续时间, CLKOUT 低电平的时间	H-2		H+2	ns
C6	$t_{w(XCOH)}$ 脉冲持续时间, CLKOUT 高电平的时间	H-2		H+2	ns
C7	t_{pPLL} 锁定时间 ⁽⁴⁾			131072 $t_{c(Cl)}$	ns

(1) 假定这些参数有一个40pF 的负载。

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

(3) PLL 必须被用于最大频率运行。

(4) 这个参数已经从芯片之前修订版本中的 4096OSC 周期中改变。



A. OSC与CLKOUT 的关系取决于所选择的分频因子。图5-10中显示的波形关系只用于解释时序参数并且根据实际配置会有所不同。

B. CLKOUT 被配置成反映SYSCLKOUT。

图 5-10. 时钟时序

5.15 复位时序

表 5-11. 复位 RESET 时序要求⁽¹⁾

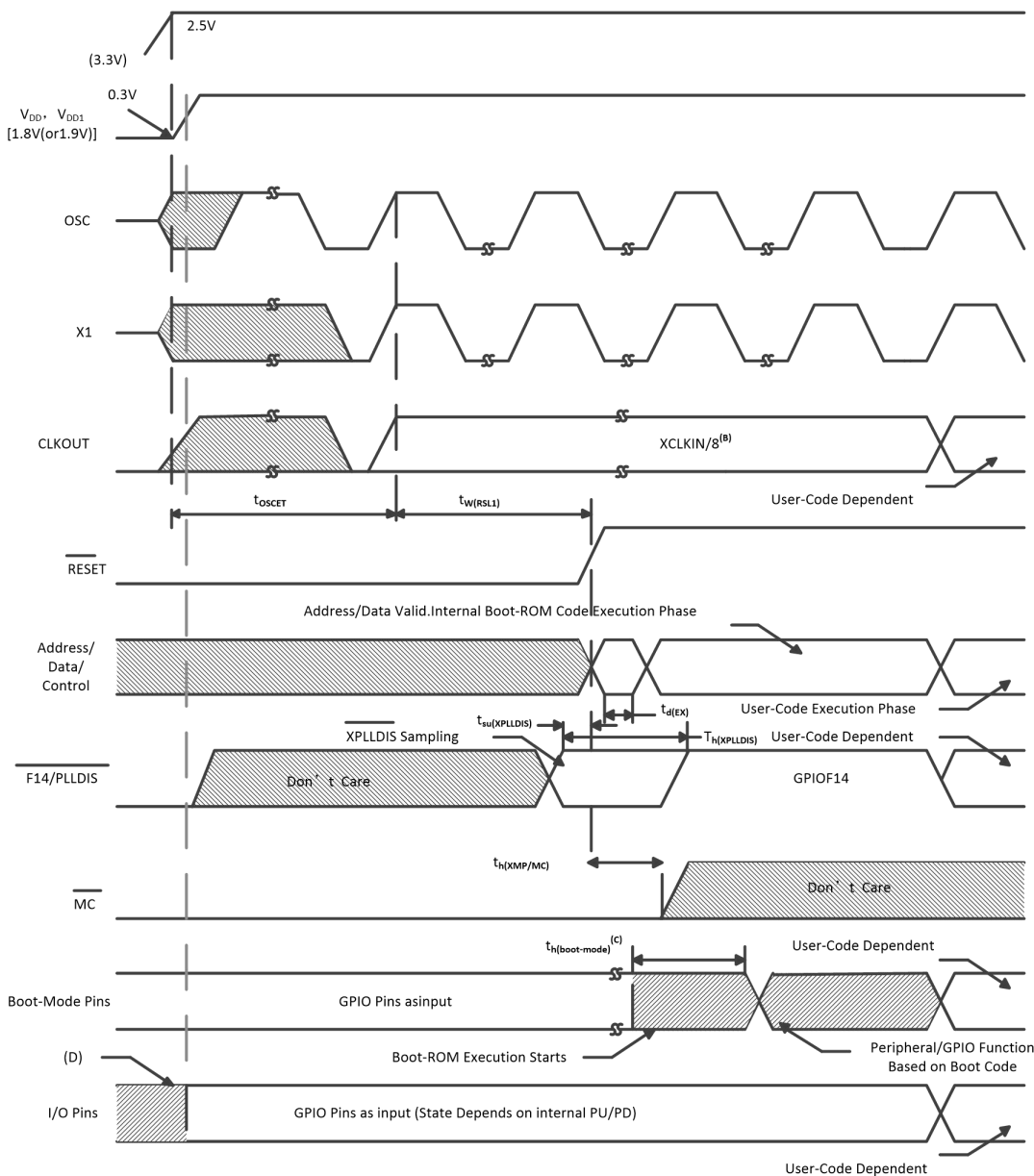
		最小值	标称值	最大值	单位
$t_{w(RSL1)}$ 脉冲持续时间, 稳定 OSC 至 RESET 高电平的时间	热复位	$8t_{c(CI)}$			周期
$t_{w(RSL2)}$ 脉冲持续时间, RESET 低电平的时间		$8t_{c(CI)}$			周期
$t_{w(WDRS)}$ 脉冲持续时间, 由安全装置生成复位脉冲的时间				$512t_{c(CI)}$	周期
$t_{d(EX)}$ 延迟时间, RESET 高电平后, 地址/数据有效的时间				$32t_{c(CI)}$	周期
$t_{OSCST}^{(2)}$ 振荡器启动时间		1	10		ms
$t_{su(XPLLDIS)}$ 针对 F14/PLLDIS 引脚的建立时间		$16t_{c(CI)}$			周期
$t_{h(XPLLDIS)}$ 针对 F14/PLLDIS 引脚的保持时间		$16t_{c(CI)}$			周期
$t_{h(MC)}$ 针对 \overline{MC} 引脚的保持时间		$16t_{c(CI)}$			周期
t_h (引导模式) 引导模式引脚的保持时间		$2520t_{c(CI)}^{(3)}$			周期

(1) 如果使用外部振荡器/时钟源, 复位时间必须在 V_{DD} 达到 1.5V 之后至少 1ms 内为低电平。

(2) 取决于晶振/谐振器和电路板设计。

(3) 引导 ROM 读取密码位置。因此, 这个时序要求包括用于闪存的唤醒时间。

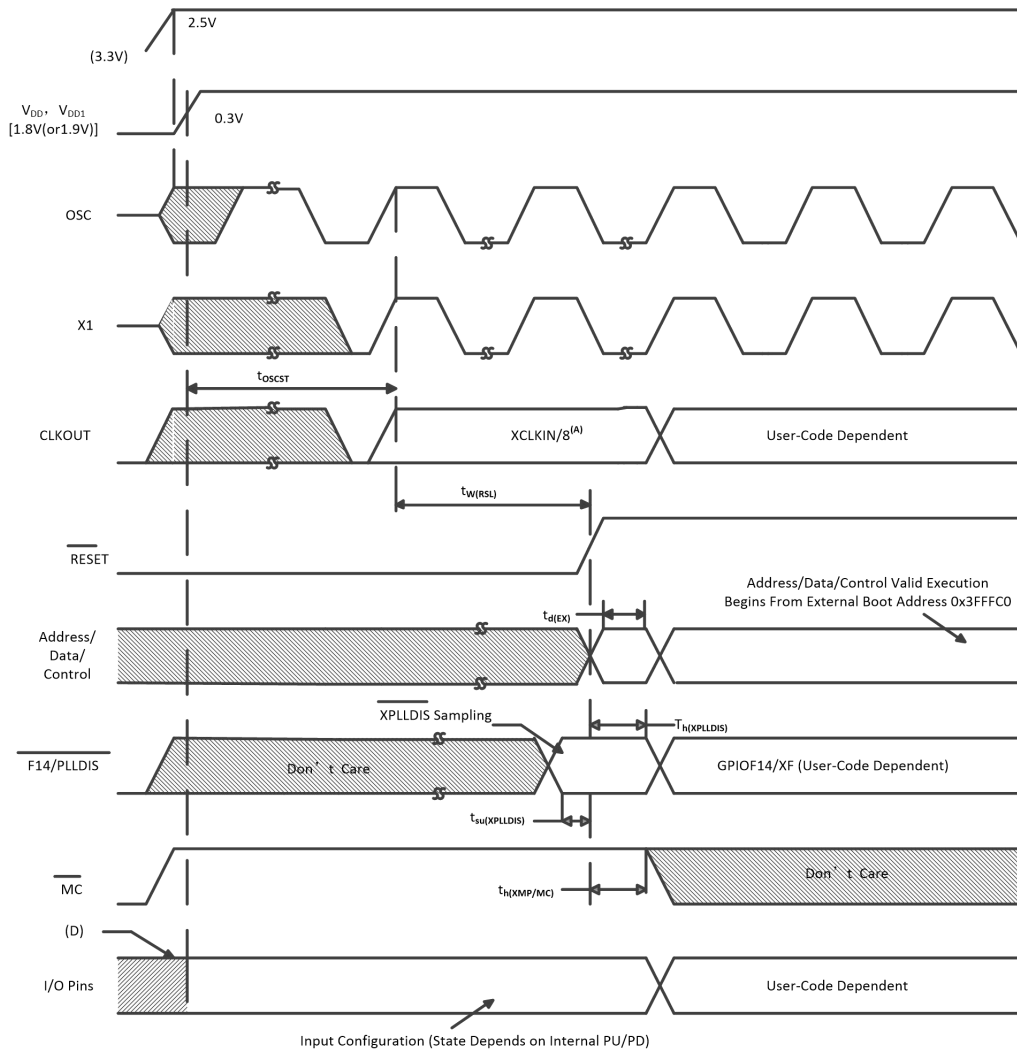
ADP32FXX Digital Signal Processor



- A. $V_{DDAn}-V_{DDA1}/V_{DDA2}$ 和AVDDREF
- B. 加电时，如果PLL被启用，SYSCLKOUT为OSC/2。由于XINTCNF2寄存器内的XTIMCLK和CLKMODE位出现时的状态为复位状态1，SYSCLKOUT在它出现在CLKOUT上之前被进一步4分频。这就解释了在这个阶段CLKOUT=OSC/8的原因。
- C. 复位后，引导ROM代码执行用于1260SYSCLKOUT周期(SYSCLKOUT=OSC/2)的指令，然后采样BOOT模式引脚。基于引导模式引脚的状态，引导代码向目的内存或者引导代码函数下达分支指令。为了适当的选择引导模式，BOOT模式引脚应该从引导ROM执行时间开始后至少2520 OSC周期内保持高电平/低电平。如果引导ROM代码在加电条件后（在调试器环境中）执行代码，引导代码执行时间由当前的SYSCLKOUT的速度而定。SYSCLKOUT将基于用户环境并可在PLL启用或者不启用时使用。
- D. 在1.8V（或者1.9V）电源至少达到1V并且3.3V电源达到2.5V之前，GPIO引脚的状态未定义（也就是说，它们应该为输入或者输出）。

图 5-11. 微计算机模式中的加电复位($\overline{MC}=0$)(请见注释 D)

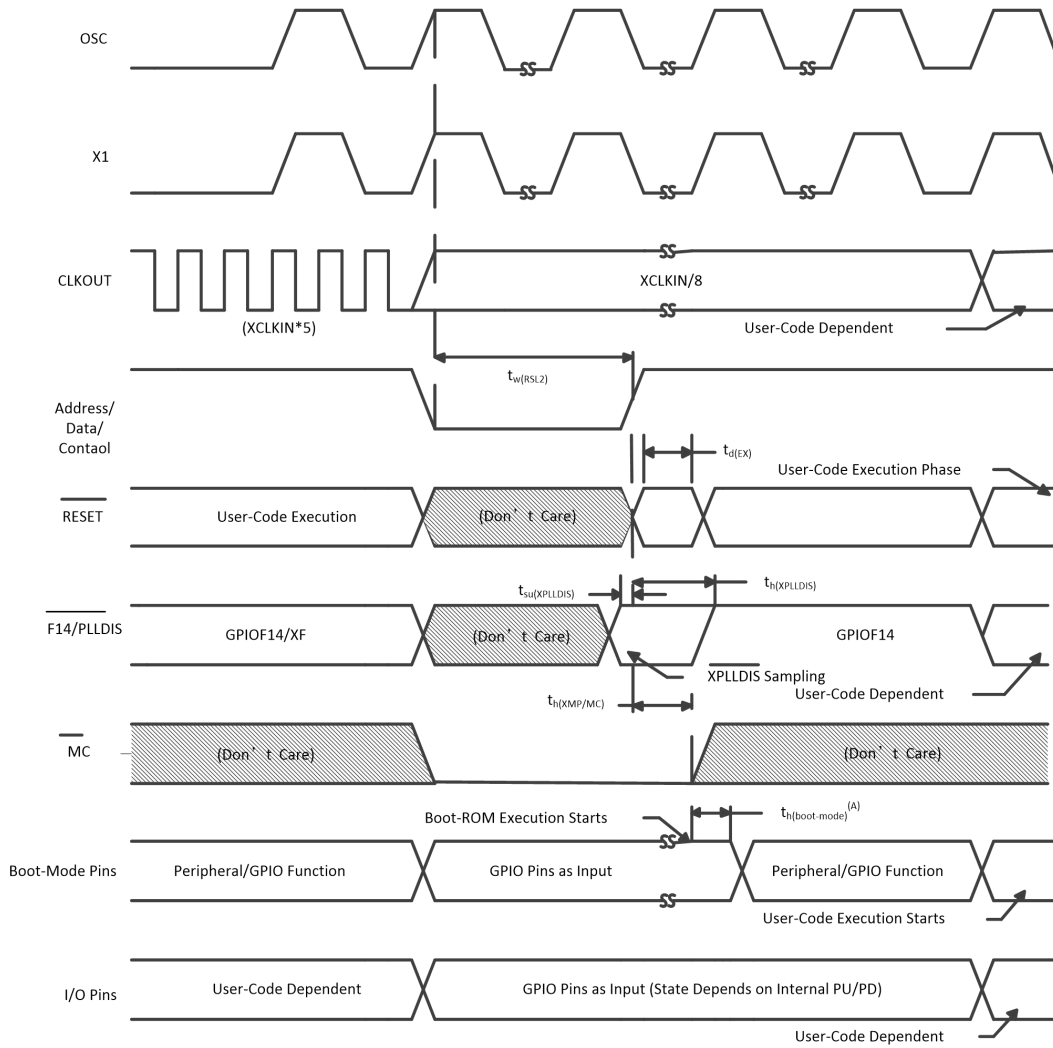
ADP32FXX Digital Signal Processor



A. 加电时，如果PLL 被启用，SYSCLKOUT 为OSC/2。由于XINTCNF2 寄存器内的XTIMCLK 和CLKMODE 位出现时的状态为复位状态1，SYSCLKOUT 在它出现在CLKOUT 上之前被进一步4分频。这就解释了在这个阶段CLKOUT=OSC/8 的原因。

该为输入或者输出)。

图 5-12. 微处理器模式中的加电复位($\overline{MC}=1$)



A. 复位后，引导ROM 代码执行用于 1260 SYSCLKOUT 周期($\text{SYSCLKOUT} = \text{OSC}/2$) 的指令，然后采样BOOT 模式引脚。基于引导模式引脚的状态，引导代码向目的内存或者引导代码函数下达分支指令。为了适当的选择引导模式，BOOT 模式引脚应该从引导ROM 执行时间开始后至少 2520 OSC 周期内保持高电平/低电平。如果引导ROM 代码在加电条件后（在调试器环境中）执行代码，引导代码执行时间由当前的 SYSCLKOUT 的速度而定。 SYSCLKOUT 将基于用户环境并可在PLL 启用或者不启用时使用。

图 5-13. 微计算机模式中的热复位

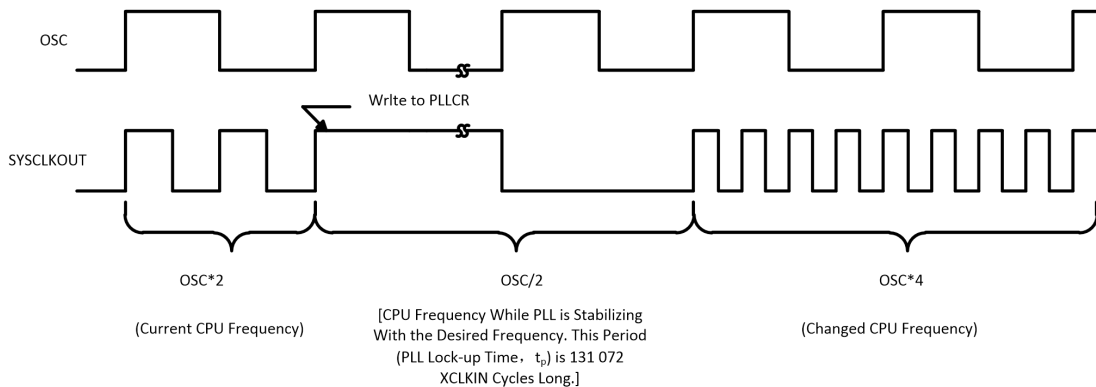


图 5-14. 写入 PLLCR 寄存器所产生的结果

5.16 低功耗模式唤醒时序

表 5-12. IDLE 模式时序要求

		最小值	标称值	最大值	单位
$t_{w(WAKE-INT)}$ 脉冲持续时间, 外部唤醒信号的时间	无输入限定器	$2t_{c(SCO)}$			周期
	带有输入限定器	$1t_{c(SCO)}+IQT^{(1)}$			

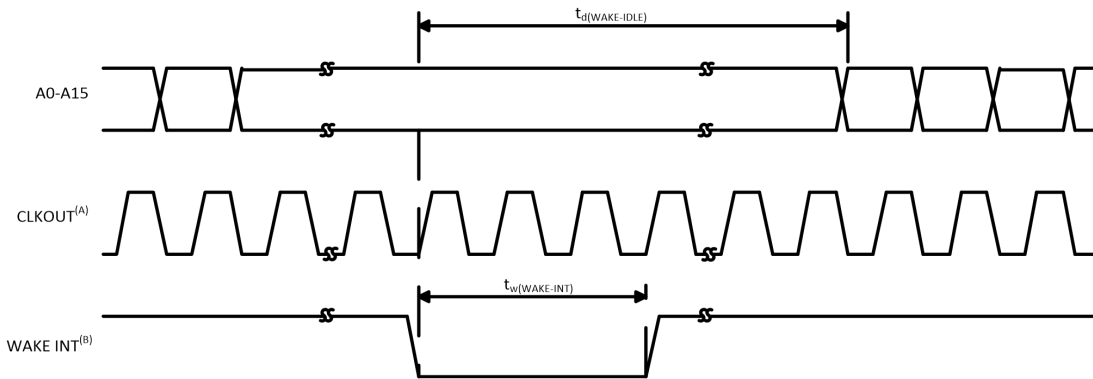
(1) 输入限定时间(IQT)=[$t_{c(SCO)} \times 2 \times QUALPRD$] $\times 5 + [t_{c(SCO)} \times 2 \times QUALPRD]$.

表 5-13. IDLE 模式开关特性

参数		测试条件	最小值	典型值	最大值	单位
$t_d(WAKE-IDLE)$	延迟时间, 外部唤醒信号到程序执行重新开始的时间 ⁽¹⁾					
	• 从闪存唤醒 - 激活状态中的闪存模块	无输入限定器			$8t_{c(SCO)}$	周期
		带有输入限定器			$8t_{c(SCO)}+IQT^{(2)}$	
	• 从闪存唤醒 - 睡眠状态中的闪存模块	无输入限定器			$1050t_{c(SCO)}$	周期
		带有输入限定器			$1050t_{c(SCO)}+IQT^{(2)}$	
	• 从 SARAM 中唤醒	无输入限定器			$8t_{c(SCO)}$	周期
带有输入限定器				$8t_{c(SCO)}+IQT^{(2)}$		

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

(2) 输入限定时间(IQT)=[$t_{c(SCO)} \times 2 \times QUALPRD$] $\times 5 + [t_{c(SCO)} \times 2 \times QUALPRD]$.



A. CLKOUT=SYSCLKOUT

B. WAKE INT 可以是任一被启用的中断, \overline{WDINT} , 或者 RESET.

图5-15. IDLE 进入和退出时序

表5-14. STANDBY 模式时序要求

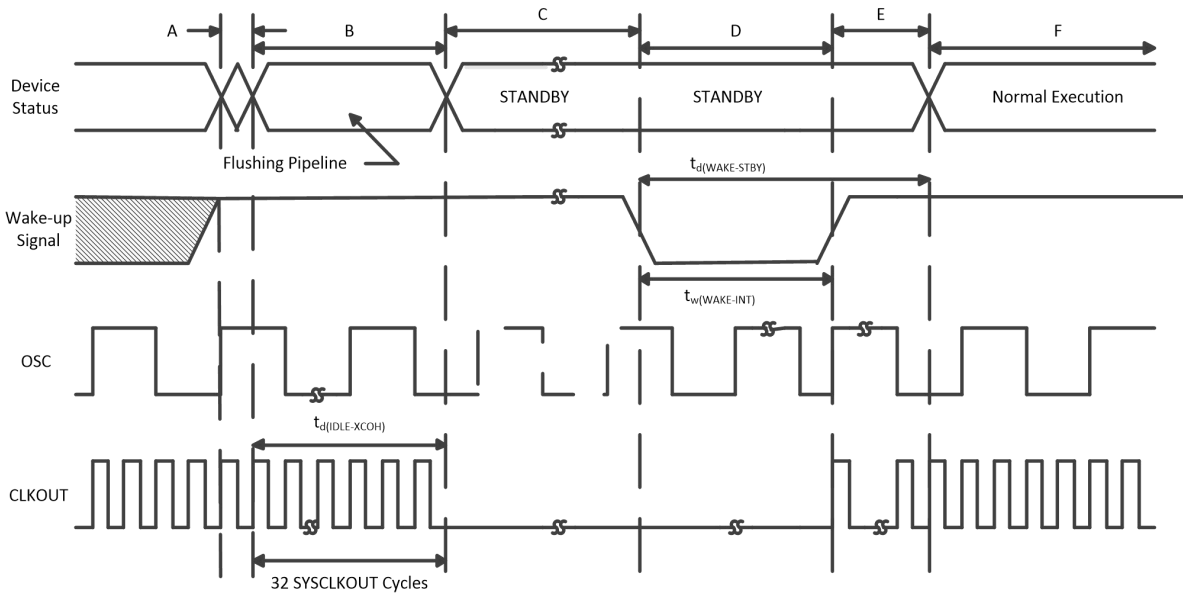
		最小值	标称值	最大值	单位
$t_{w(WAKE-INT)}$ 脉冲持续时间, 外部唤醒信号的时间	无输入限定器	$12t_{c(CI)}$			周期
	带有输入限定器	$(2 + QUALSTDBY) * t_{c(CI)}^{(1)}$			

(1) QUALSTDBY 是一个 LPMCR0 寄存器内的 6 位字段。

表 5-15. STANDBY 模式开关特性

参数		测试条件	最小值	典型值	最大值	单位
$t_{d(\text{IDLE-XCOH})}$	延迟时间，IDLE 指令执行至 CLKOUT 为高电平的时间		$32t_{c(\text{SCO})}$		$45t_{c(\text{SCO})}$	周期
	延迟时间，外部唤醒信号到程序执行重新开始的时间 ⁽¹⁾				$12t_{c(\text{CI})}$	
$t_{d(\text{WAKE-STBY})}$	• 从闪存唤醒 – 激活状态中的闪存模块	无输入限定器				周期
		带有输入限定器			$12t_{c(\text{CI})} + t_{w(\text{WAKE-INT})}$	
	• 从闪存唤醒 – 睡眠状态中的闪存模块	无输入限定器			$1125t_{c(\text{SCO})}$	周期
		带有输入限定器			$1125t_{c(\text{SCO})} + t_{w(\text{WAKE-INT})}$	
	• 从 SARAM 中唤醒	无输入限定器			$12t_{c(\text{CI})}$	周期
		带有输入限定器			$12t_{c(\text{CI})} + t_{w(\text{WAKE-INT})}$	

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



A. 被执行的IDLE 指令将器件置于STANDBY 模式。

B. PLL 块响应STANDBY 信号。在被关闭前，SYSCLKOUT 在下面标明的一定数量的周期内被保持：

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

这个延迟使得CPU 管线和其它等待的操作被适当清空。如果一个到XINTF 的访问正在进行中并且它的访问时间大于这个值，那么这个访问将发生故障。建议在没有XINTF 访问进行时从SARAM 进入STANDBY 模式。

C.到外设的时钟被关闭。然而，PLL 和安全装置并未关闭。此器件现在处于STANDBY 模式。

D.外部唤醒信号被驱动为有效。

E. 在一个延迟周期内，退出STANDBY 模式。

F. 正常执行重新开始。此器件将响应中断 (如果被启用的话)。

图 5-16. STANDBY 进入和退出时序

表 5-16. HALT 模式时序要求

	最小值	标称值	最大值	单位
$t_{w(WAKE-NMI)}$ 脉冲持续时间, NMI 唤醒信号的时间	$2t_{c(CI)}$			周期
$t_{w(WAKE-RESET)}$ 脉冲持续时间, RESET 唤醒信号的时间	$8t_{c(CI)}$			周期

表 5-17. HALT 模式开关特性

参数	最小值	典型值	最大值	单位
$t_{d(IDLE-XCOH)}$ 延迟时间, IDLE 指令被执行至 CLKOUT 为高电平的时间	$32t_{c(SCO)}$	$45t_{c(SCO)}$		周期
t_{pPLL} 锁存时间			$131072t_{c(CI)}$	周期
$t_{d(WAKE)}$ 延迟时间, PLL 锁存到程序执行重新开始的时间			$1125t_{c(SCO)}$	周期
	<ul style="list-style-type: none"> 从闪存唤醒 —处于睡眠状态的闪存模块 从 SARAM 中唤醒 		$35t_{c(SCO)}$	周期

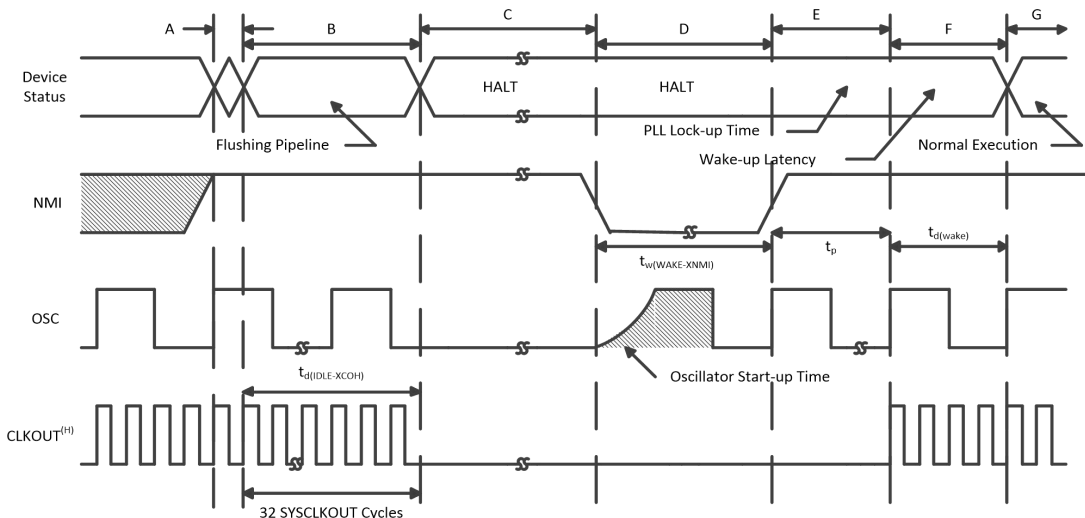


图 5-17. 使用 NMI 的 HALT 唤醒

- IDLE 指令被执行以将器件置于 HALT 模式。
- PLL 块响应 HALT 信号。在振荡器被关闭并且到内核的 CLKIN 被停止前, SYSCLKOUT 被保持另外 32 个周期。这个 32 周期延迟使得 CPU 管线和其它等待的操作被适当清空。
- 到外设的时钟被关闭并且内部振荡器和 PLL 被关断。器件现在处于 HALT 模式, 消耗绝对最小功率。
- 当 NMI 被驱动有效, 振荡器被关闭; 但是 PLL 未被激活。当一个外部振荡器被使用时, $2t_{c(CI)}$ 的脉冲持续时间适用。如果内部振荡器被使用, 振荡器唤醒时间应该被添加到这个参数。
- 当 NMI 被置为无效时, 它启动 PLL 锁序列, 这将花费 131,072 OSC 周期。
- 当到内核的 CLKIN 被启用时, 在一个延迟后, 此器件响应此中断 (如果被启用)。现在退出 HALT 模式。
- 正常运行重新开始。
- CLKOUT = SYSCLKOUT

5.17 事件管理器

5.17.1 PWM 时序

PWM 是指所有 EM1 和 EM2 上的 PWM 输出。

表 5-18. PWM 开关特性^{(1) (2)}

参数	测试条件	最小值	最大值	单位
$t_w(\text{PWM})^{(3)}$	脉冲持续时间, PWMx 输出 高电平/低电平的时间	25		ns
$t_d(\text{PWM})_{\text{XCO}}$	延迟时间, CLKOUT 高电平 到 PWMx 输出开关		10	ns

(1) 请见针对 PWM 引脚的下降/上升时间的GPIO 输出时序。

(2) PWM 引脚切换频率由GPIO 输出缓冲器开关频(20MHz) 限制。

(3) 相对于 PWM 周期, PWM 输出可以为 100%, 0%, 或者 $t_c(\text{HCO})$ 的增量。

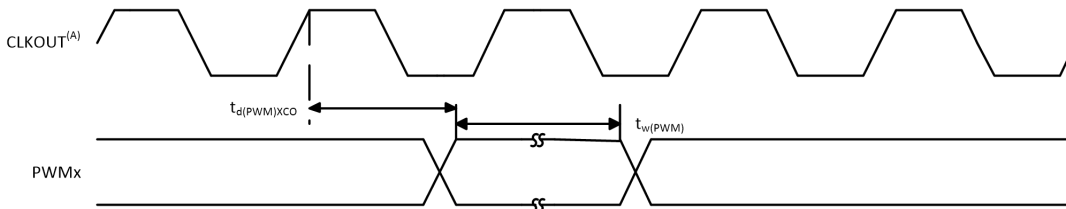
表 5-19. 定时器和捕捉单元时序要求^{(1) (2)}

参数	测试条件	最小值	最大值	单位
$t_w(\text{TDIR})$ 脉冲持续时间, TDIRx 低电平/高电平的时间	无输入限定器	$2t_c(\text{SCO})$		周期
	带有输入限定器	$1t_c(\text{SCO}) + \text{IQT}(3)$		
$t_w(\text{CAP})$ 脉冲持续时间, CAPx 输入低电平/高电平的时间	无输入限定器	$2t_c(\text{SCO})$		周期
	带有输入限定器	$1t_c(\text{SCO}) + \text{IQT}(3)$		
$t_w(\text{TCLKINL})$ 脉冲持续时间, TCLKINx 低电平作为 TCLKINx 周期时间一部分的时间		40	60	%
$t_w(\text{TCLKINH})$ 脉冲持续时间, TCLKINx 高电平作为 TCLKINx 周期时间一部分的时间		40	60	%
$t_c(\text{TCLKIN})$ 周期时间, TCLKINx		$4t_c(\text{HCO})$		ns

(1) QUALPRD 位字段值的范围从0 (无限定条件) 到0xFF (510SYSCLKOUT 周期)。限定采样周期为 $2n$ SYSCLKOUT 周期, 在这里“n”为存储在QUALPRD 位字段中的值。作为一个示例, 当QUALPRD=1, 限定采样周期为 $1 \times 2 = 2$ SYSCLKOUT 周期 (也就是说, 输入为每2个SYSCLKOUT 周期被采样)。六个这样的样本将取自五个采样窗口, 每个窗口为 $2n$ SYSCLKOUT 周期。对于QUALPRD=1, 所需的最小宽度为 $5 \times 2 = 10$ 个SYSCLKOUT 周期。然而, 由于外部信号被异步驱动, 一个11 SYSCLKOUT 宽的脉冲可确保可靠识别。

(2) 到QEP= $\min[\text{HSPCLK}/2, 20\text{MHz}]$ 的最大输入频率。

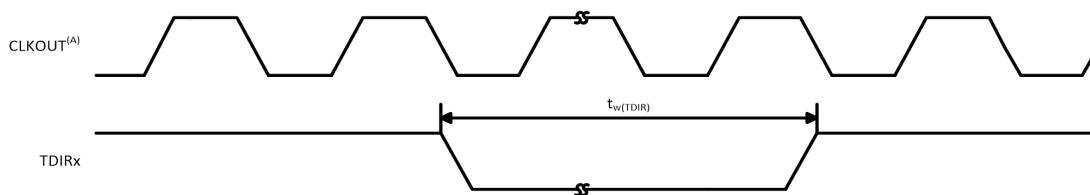
(3) 输入限定时间(IQT)=[$t_c(\text{SCO}) \times 2 \times \text{QUALPRD}$] $\times 5 + [t_c(\text{SCO}) \times 2 \times \text{QUALPRD}]$ 。



A. CLKOUT=SYSCLKOUT

图 5-18.PWM 时序

ADP32FXX Digital Signal Processor



A. CLKOUT=SYSCLKOUT

图 5-19.TDIRx 时序

表 5-20. 外部 ADC 转换开始- EM1 - 开关特性⁽¹⁾

参数	最小值	最大值	单位
$t_d(XCOH-EM1SOCL)$		$1t_{c(SCO)}$	周期
$t_w(EM1SOCL)$	$32t_{c(HCO)}$		ns

(1) CLKOUT=SYSCLKOUT

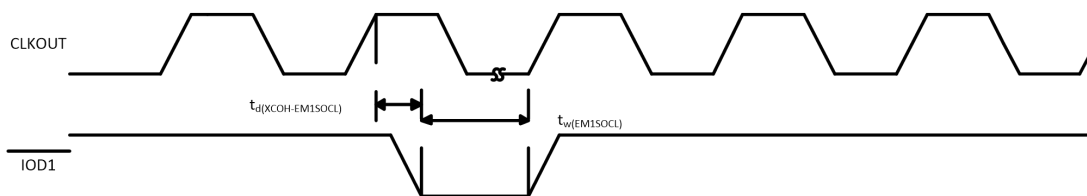


图 5-20. IOD1时序

表 5-21. 外部 ADC 转换开始- EM2 - 开关特性⁽¹⁾

参数	最小值	最大值	单位
$t_d(XCOH-EM2SOCL)$		$1t_{c(SCO)}$	周期
$t_w(EM2SOCL)$	$32t_{c(HCO)}$		ns

(1) CLKOUT=SYSCLKOUT

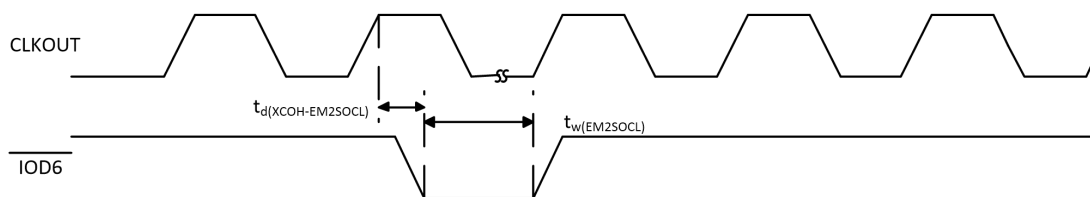


图 5-21. IOD6时序

5.17.2 中断时序

表 5-22. 中断开关特性

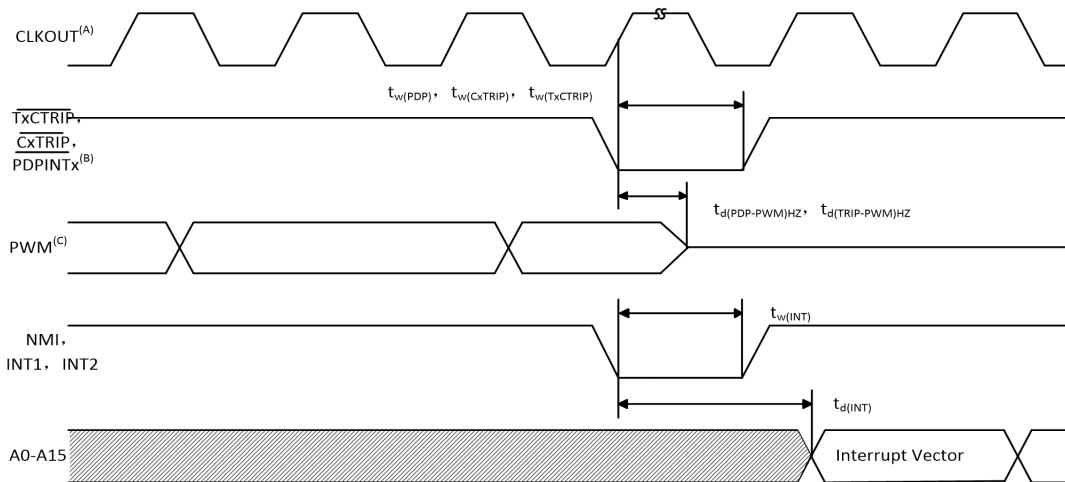
参数	最小值	最大值	单位
$t_{d(PDP-PWM)HZ}$ 延迟时间, $\overline{PDPINTx}$ 低电平至 PWM 高阻抗状态的时间	无输入限定器	12	ns
	带有输入限定器	$1t_{c(SCO)} + IQT + 12^{(1)}$	
$t_{d(TRIP-PWM)HZ}$ 延迟时间, $\overline{CxTRIP}/\overline{TxCTRIp}$ 信号低电平到 PWM 高阻抗状态的时间	无输入限定器	$3 * t_{c(SCO)}$	ns
	带有输入限定器	$2t_{c(SCO)} + IQT^{(1)}$	
$t_{d(INT)}$ 延迟时间, INT 低电平/高电平到中断矢量提取的实际那		$IQT + 12t_{c(SCO)}^{(1)}$	ns

(1) 输入限定时间(IQT)=[$t_{c(SCO)} \times 2 \times QUALPRD$] $\times 5 + [t_{c(SCO)} \times 2 \times QUALPRD]$.

表 5-23. 中断时序要求

参数	最小值	最大值	单位
$t_{w(INT)}$ 脉冲持续时间, INT 输入低电平/高电平	无限定符	$2t_{c(SCO)}$	周期
	带有限定符	$1t_{c(SCO)} + IQT^{(1)}$	
$t_{w(PDP)}$ 脉冲持续时间, $\overline{PDPINTx}$ 输入低电平的时间	无限定符	$2t_{c(SCO)}$	周期
	带有限定符	$1t_{c(SCO)} + IQT^{(1)}$	
$t_{w(CxTRIP)}$ 脉冲持续时间, \overline{CxTRIP} 输入低电平的时间	无限定符	$2t_{c(SCO)}$	周期
	带有限定符	$1t_{c(SCO)} + IQT^{(1)}$	
$t_{w(TxCTRIp)}$ 脉冲持续时间, $\overline{TxCTRIp}$ 输入低电平的时间	无限定符	$2t_{c(SCO)}$	周期
	带有限定符	$1t_{c(SCO)} + IQT^{(1)}$	

(1) 输入限定时间(IQT)=[$t_{c(SCO)} \times 2 \times QUALPRD$] $\times 5 + [t_{c(SCO)} \times 2 \times QUALPRD]$.



A. CLKOUT=SYSCLKOUT

B. TxCTRIp-IOD0, IOD1, IOD5, IOD6, CxTRIP-IOA13, IOA14, IOA15, IOB13, IOB14, 或者IOB15, PDPINTx-IOD0或者IOD5

C. PWM 是指所有器件内的PWM 引脚 (即, PWMn 和TnPWM 引脚或者与每个CxTRIP引脚相关的PWM 引脚对)。

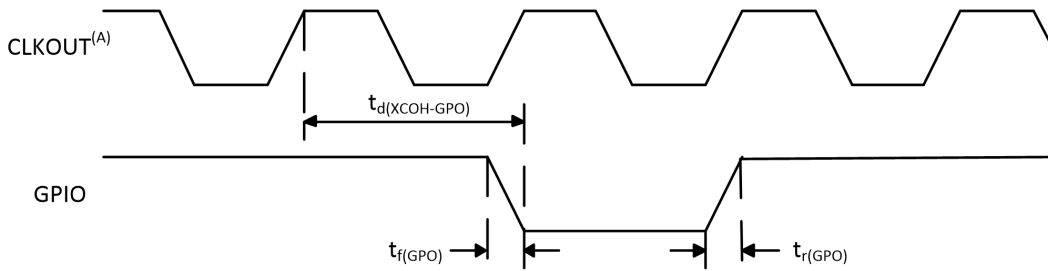
$\overline{PDPINTx}$ 被置为高电平之后 PWM 引脚的状态取决于 FCOMPOE 位的状态。

图 5-22. 外部中断时序

5.18 通用输入/输出(GPIO) - 输出时序

表 5-24. 通用输出开关特性

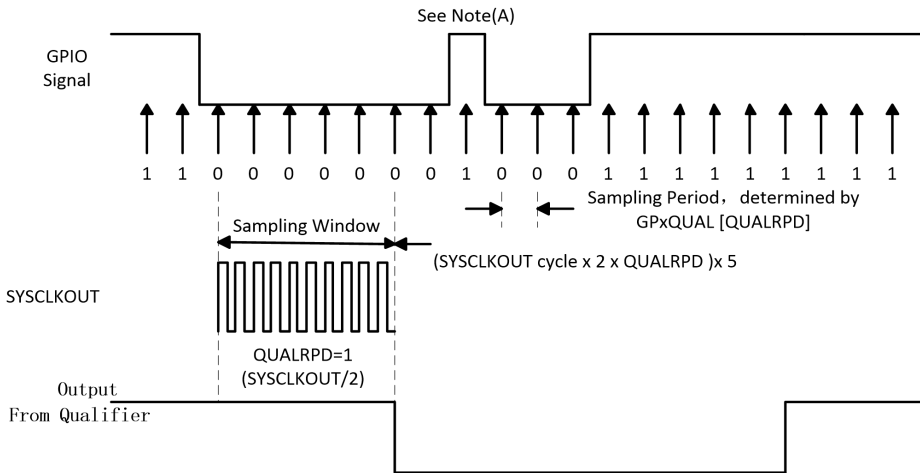
参数	最小值	最大值	单位
$t_{d(XCOH-GPO)}$ 延迟时间, CLKOUT 高电平到 GPIO 低电平/高电平的时间	所有 GPIO	$1t_{c(SCO)}$	周期
$t_{r(GPO)}$ 上升时间, GPIO 从低电平切换至高电平的时间	所有 GPIO	10	ns
$t_{f(GPO)}$ 下降时间, GPIO 从高电平切换至低电平的时间	所有 GPIO	10	ns
f_{GPO} 切换频率, GPO 引脚		20	MHz



A. CLKOUT=SYSCLKOUT

图 5-23. 通用输出时序

5.19 通用输入/输出(GPIO) - 输入时序



- A. 这个毛刺脉冲将被输入限定器所忽略。QUALPRD 位字段指定了限定采样周期。它可在00 至0xFF 间变化。当QUALPRD=00 时，输入限定不使用。对于任何其它的“n” 值，限定采样周期为2n SYSCLKOUT 周期（也就是说，在每一个SYSCLKOUT 周期上，GPIO 引脚将被采样）。为了识别一个给定的输入，六个连续的样本必须为同样的值。
- B. 为了使限定符检测到变化，输入必须在10 个SYSCLKOUT 周期或者更长的时间内保持稳定。换句话说，输入应该在(5 x QUALPRD x 2) SYSCLKOUT 周期内保持稳定。这将为检测发生启用所需的5 个采样周期。由于外部时钟被异步驱动，一个13 SYSCLKOUT 宽的脉冲将确保可靠识别。

图 5-24. GPIO 输入限定符- 针对 QUALPRD=1 的示例图

表 5-25. 通用输入时序要求

		最小值	最大值	单位
$t_{w(GPI)}$ 脉冲持续时间, GPIO 低电平/ 高电平的时间	所有 GPIO	无限定符	$2t_{c(SCO)}$	周期
		带有限定符	$1t_{c(SCO)}+IQT^{(1)}$	

(1) 输入限定时间(IQT)=[$t_{c(SCO)} \times 2 \times QUALPRD$] \times 5 + [$t_{c(SCO)} \times 2 \times QUALPRD$].

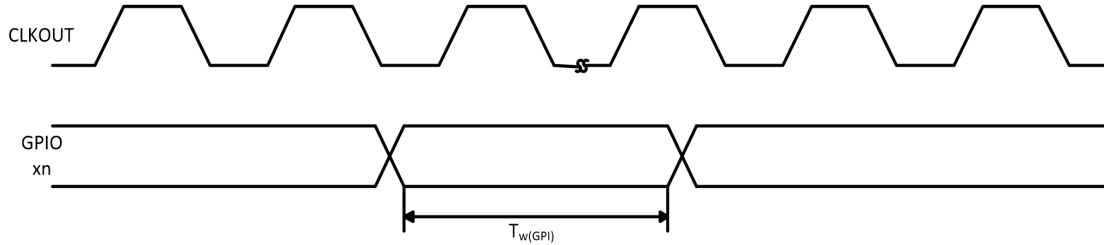


图 5-25 . 通用输入时序

NOTE:

对于通用输入的脉宽要求也同样适用于 \overline{XBIO} 和 ADCSOC 引脚。

5.20 串行外设接口(SPI) 主控模式时序

表5-26列出了主控模式时序（时钟相位= 0）而表5-27列出了时序（时钟相位=1）。图5-26和图5-27显示了时序波形。

表5-26. SPI 主控模式外部时序（时钟相位= 0）^{(1) (2)}

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

(1) 主控/受控位(SPICTL.2) 被设定, 而时钟相位的位(SPICTL.3) 被清除。

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

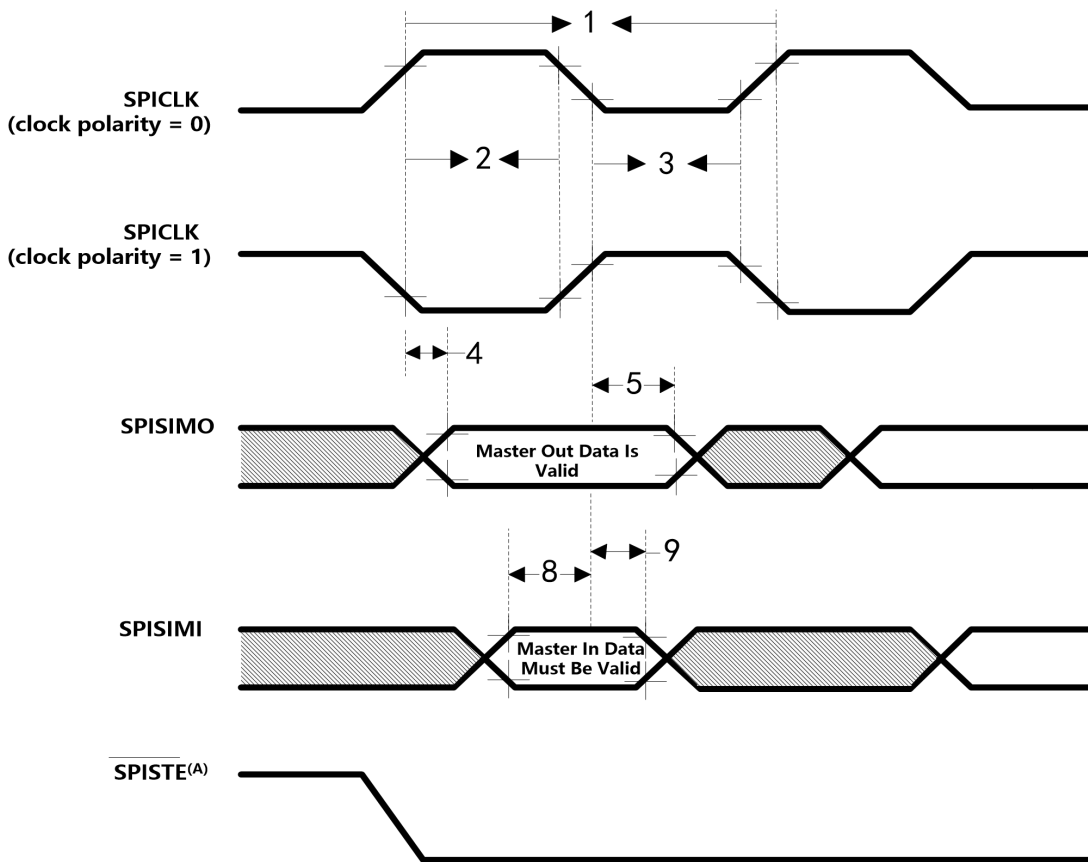
$t_{c(LCO)}$ = LSPCLK 周期时间

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

ADP32FXX Digital Signal Processor

注释：内部时钟预分频器必须被调节，这样，SPI 时钟速度被限定在以下SPI 时钟速率上：

- 主控模式发送：最大值20MHz 主控模式接收：最大值12.5MHz
- 受控模式发送：最大值 12.5MHz 受控模式接收：最大值 12.5MHz



A. 在 主控模式下，在有效的 SPI 时钟边沿之前 $0.5t_{c(SPC)}$ ， \overline{SPISTE} 变为有效。在字的尾端， \overline{SPISTE} 在接收到最后一个数据位的边沿 (SPICLK) 之后 $0.5t_{c(SPC)}$ 将变为无效，除非 \overline{SPISTE} 在 FIFO 和非 FIFO 模式中的背靠背传送字间保持有效。

图5-26. SPI 主控模式外部时序 (时钟相位 = 0)

表 5-27. SPI 主控模式外部时序 (时钟相位 = 1) (1) (2)

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

(1) 主控/受控位(SPICTL.2) 被设定并且时钟相位的位(SPICTL.3) 被设定。

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

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

注释: 内部时钟预分频器必须被调节, 这样, SPI 时钟速度被限定在以下SPI 时钟速率上:

- 主控模式发送: 最大值20MHz 主控模式接收: 最大值12.5MHz
- 受控模式发送: 最大值 12.5MHz 受控模式接收: 最大值 12.5MHz

表 5-27. SPI 主控模式外部时序 (时钟相位= 1) ⁽¹⁾⁽²⁾(continued)

编号			当(SPIBRR+1) 为偶数或者 SPIBRR=0 或者2 时的SPI		当(SPIBRR+1) 为奇数并且 SPIBRR>3 时的 SPI		单位
			最小值	最大值	最小值	最大值	
11 ⁽³⁾	$t_{v(SPCH-SOMI)M}$	有效时间, SPICLK 高电平之后 SPISOMI 数据有效的的时间 (时钟极性 = 0)	$0.25t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-10$		ns
	$t_{v(SPCL-SOMI)M}$	有效时间, SPICLK 低电平之后 SPISOMI 数据有效的的时间 (时钟极性 = 1)	$0.25t_{c(SPC)M}-10$		$0.5t_{c(SPC)M}-10$		

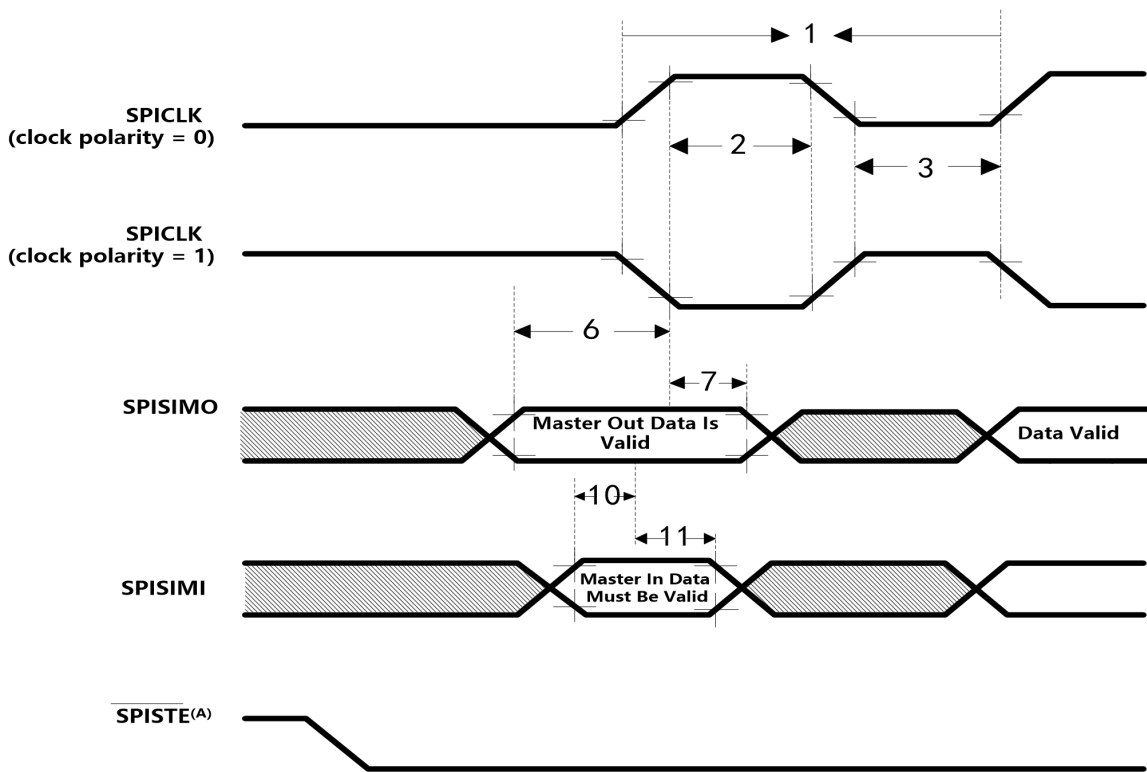


图 5-27. SPI 主控外部时序 (时钟相位= 1)

A. 在主控模式下, 在有效的SPI 时钟边沿之前 $0.5t_{c(SPC)}$, \overline{SPISTE} 变为有效。在字的尾端, \overline{SPISTE} 在接收到最后一个数据位的边沿 (SPICLK) 之后 $0.5t_{c(SPC)}$ 将变为无效, 除非 \overline{SPISTE} 在FIFO 和非FIFO 模式中的背靠背传送字间保持有效。

5.21 串行外设接口(SPI) 受控模式时序

表5-28列出了主控模式时序（时钟相位= 0）而表5-29列出了时序（时钟相位= 1）。图5-28和图5-29显示了时序波形。

表5-28. SPI 受控模式外部时序（时钟相位= 0）^{(1) (2)}

编号		最小值	最大值	单位
12	$t_{c(SPC)}$ 周期时间, SPICLK	$4t_{c(LCO)}$		ns
13 ⁽³⁾	$t_w(SPCH)$ 脉冲持续时间, SPICLK 高电平的时间 (时钟极性= 0)	$0.5t_{c(SPC)}-10$	$0.5t_{c(SPC)}$	ns
	$t_w(SPCL)$ 脉冲持续时间, SPICLK 低电平的时间 (时钟极性= 1)	$0.5t_{c(SPC)}-10$	$0.5t_{c(SPC)}$	
14 ⁽³⁾	$t_w(SPCL)$ 脉冲持续时间, SPICLK 低电平的时间 (时钟极性= 0)	$0.5t_{c(SPC)}-10$	$0.5t_{c(SPC)}$	ns
	$t_w(SPCH)$ 脉冲持续时间, SPICLK 高电平的时间 (时钟极性= 1)	$0.5t_{c(SPC)}-10$	$0.5t_{c(SPC)}$	
15 ⁽³⁾	$t_d(SPCH-SOMI)$ 延迟时间, SPICLK 高电平至 SPISOMI 有效的时间 (时钟极性= 0)	$0.375t_{c(SPC)}-10$		ns
	$t_d(SPCL-SOMI)$ 延迟时间, SPICLK 低电平至 SPISOMI 有效的时间 (时钟极性= 1)	$0.375t_{c(SPC)}-10$		
16 ⁽³⁾	$t_v(SPCL-SOMI)$ 有效时间, SPICLK 低电平之后 SPISOMI 数据有效的时间 (时钟极性= 0)	$0.75t_{c(SPC)}$		ns
	$t_v(SPCH-SOMI)$ 有效时间, SPICLK 高电平之后 SPISOMI 数据有效的时间 (时钟极性= 1)	$0.75t_{c(SPC)}$		
19 ⁽³⁾	$t_{su}(SIMO-SPCL)$ 建立时间, SPISIMO 在 SPICLK 低电平之前的时间 (时钟极性= 0)	0		ns
	$t_{su}(SIMO-SPCH)$ 建立时间, SPISIMO 在 SPICLK 高电平之前的时间 (时钟极性= 1)	0		
20 ⁽³⁾	$t_v(SPCL-SIMO)$ 有效时间, SPICLK 低电平后, SPISIMO 数据有效的时间 (时钟极性= 0)	$0.5t_{c(SPC)}$		ns
	$t_v(SPCH-SIMO)$ 有效时间, SPICLK 高电平之后 SPISIMO 数据有效的时间 (时钟极性= 1)	$0.5t_{c(SPC)}$		

(1) MASTER/SLAVE 位(SPICTL 2) 位被清除并且CLOSE PHASE (时钟相位) 位(SPICTL 3) 被清除。

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

$t_{c(LCO)}$ = LSPCLK 周期时间

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

注释：内部时钟预分频器必须被调节，这样，SPI 时钟速度被限定在以下SPI 时钟速率上：

- 主控模式发送：最大值20MHz 主控模式接收：最大值12.5MHz
- 受控模式发送：最大值 12.5MHz 受控模式接收：最大值 12.5MHz

ADP32FXX Digital Signal Processor

注释：内部时钟预分频器必须被调节，这样，SPI 时钟速度被限定在以下SPI 时钟速率上：

- 主控模式发送：最大值20MHz 主控模式接收：最大值12.5MHz
- 受控模式发送：最大值12.5MHz 受控模式接收：最大值12.5MHz

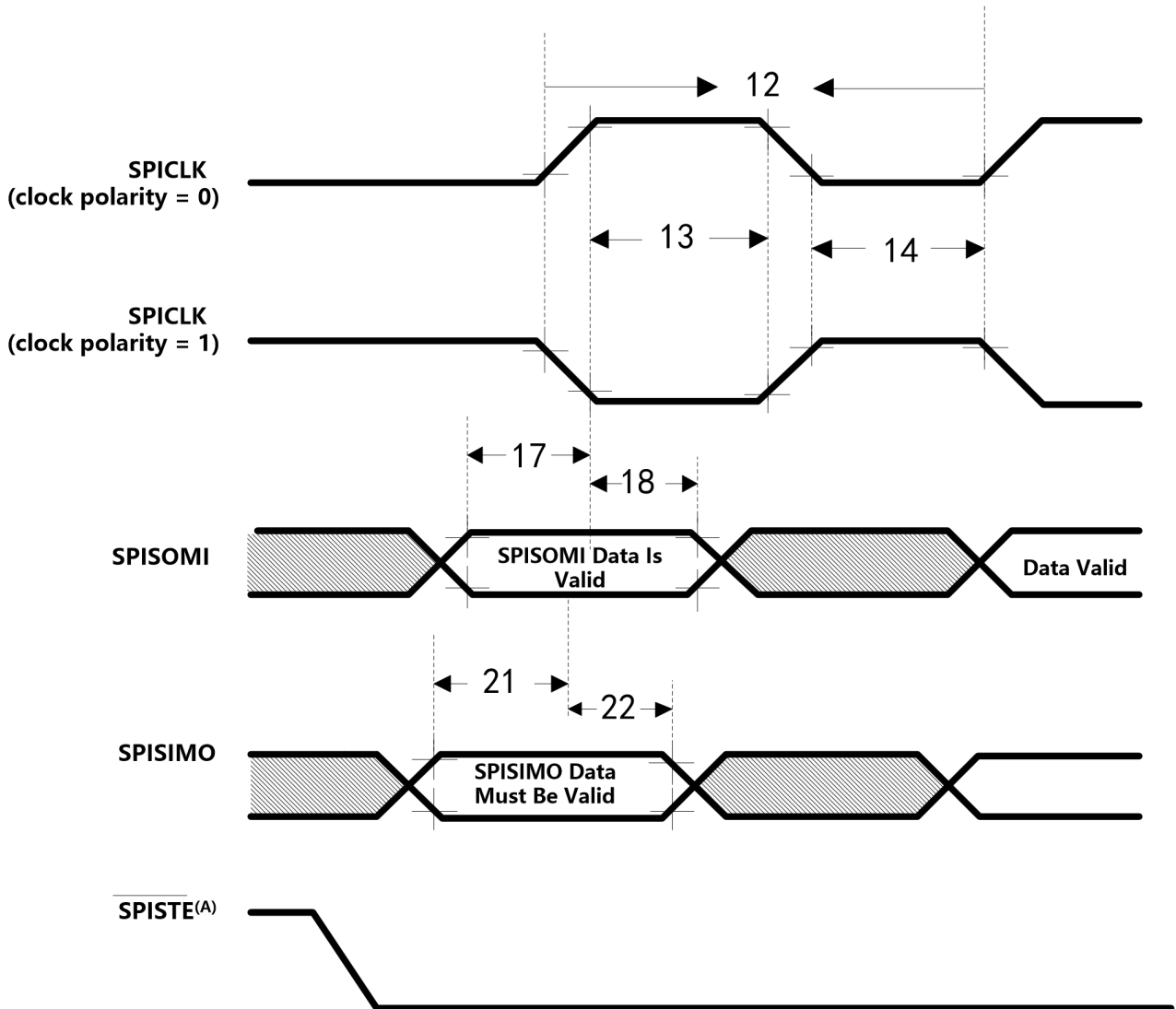


图5-29. SPI 受控模式外部时序 (时钟相位= 1)

- A. 在受控模式下， $\overline{\text{SPISTE}}$ 信号至少应该在有效SPI 时钟边沿前 $0.5t_{\text{c(SPC)}}$ 被置为低电平有效并且在接收到最后一个数据位的边沿(SPICLK)之后保持至少 $0.5t_{\text{c(SPC)}}$ 。

5.22 外部接口(XINTF) 时序

每个XINTF 访问由三部分组成：建立、有效、和跟踪。用户在XTIMING 寄存器中配置建立/有效/跟踪等待状态。有一个XTIMING 寄存器用于每个XINTF 区域。表5-30显示了XTIMING 寄存器中配置的参数和以XTIMING 周期为单位的脉冲持续时间之间的关系。

表 5-30. XTIMING 中配置的参数和脉冲持续时间之间的关系^{(1) (2)}

说明	持续时间(ns)	
	X2TIMING=0	X2TIMING=1
LR 建立周期, 读取访问	$XRDLEAD \times t_{C(XTIM)}$	$(XRDLEAD \times 2) \times t_{C(XTIM)}$
AR 激活周期, 读取访问	$(XRDACTIVE + WS + 1) \times t_{C(XTIM)}$	$(XRDACTIVE \times 2 + WS + 1) \times t_{C(XTIM)}$
TR 跟踪周期, 读取访问	$XRDTRAIL \times t_{C(XTIM)}$	$(XRDTRAIL \times 2) \times t_{C(XTIM)}$
LW 建立周期, 写入访问	$XWRLEAD \times t_{C(XTIM)}$	$(XWRLEAD \times 2) \times t_{C(XTIM)}$
AW 激活周期, 写入访问	$(XWRACTIVE + WS + 1) \times t_{C(XTIM)}$	$(XWRACTIVE \times 2 + WS + 1) \times t_{C(XTIM)}$
TW 跟踪周期, 写入访问	$XWRTRAIL \times t_{C(XTIM)}$	$(XWRTRAIL \times 2) \times t_{C(XTIM)}$

(1) $t_{C(XTIM)}$ – Cycle time, XTIMCLK

(2) WS 是指当使用 READY 时, 由硬件插入的等待状态的数量。如果此区域被配置成忽略 READY(USEREADY=0), 那么 WS=0。

当配置每个区域的XTIMING 寄存器时, 必须满足最小等待状态要求。这些要求是器件数据表中指定的任一时序要求之外的要求, 没有任何内部器件硬件来检测非法设置。

- 如果 READY 信号被忽略(USEREADY=0), 那么:

1. 建立:

$$LR \geq t_{C(XTIM)}$$

$$LW \geq t_{C(XTIM)}$$

这些配置引起以下这些 XTIMING 寄存器配置限制 (没有硬件检测非法 XTIMING 配置):

XRDLEAD	XRDACTIVE	XRDTRAIL	XWRLEAD	XWRACTIVE	XWRTRAIL	X2TIMING
≥ 1	≥ 0	≥ 0	≥ 1	≥ 0	≥ 0	0, 1

没有采样 READY 时, 有效和无效时序示例 (没有硬件检测非法 XTIMING 配置):

	XRDLEAD	XRDACTIVE	XRDTRAIL	XWRLEAD	XWRACTIVE	XWRTRAIL	X2TIMING
无效	0	0	0	0	0	0	0, 1
有效	1	0	0	1	0	0	0, 1

- 如果READY 信号在同步模式中被采样 (USEREADY=1, READYMODE=0), 那么:

1. 建立:

$$LR \geq t_{C(XTIM)}$$

$$LW \geq t_{C(XTIM)}$$

2. 有效:

$$AR \geq 2 \times t_{C(XTIM)}$$

$$AW \geq 2 \times t_{C(XTIM)}$$

注释: 限制条件不包括外部硬件等待状态。

这些配置引起以下这些 XTIMING 寄存器配置限制 (没有硬件检测非法 XTIMING 配置):

XRDLEAD	XRDACTIVE	XRDTRAIL	XWRLEAD	XWRACTIVE	XWRTRAIL	X2TIMING
≥ 1	≥ 1	≥ 0	≥ 1	≥ 1	≥ 0	0, 1

当使用同步 READY 时, 有效和无效时序示例 (没有硬件检测非法 XTIMING 配置):

ADP32FXX Digital Signal Processor

	XRDLEAD	XRDACTIVE	XRDTRAIL	XWRLEAD	XWRACTIVE	XWRTRAIL	X2TIMING
无效	0	0	0	0	0	0	0, 1
无效	1	0	0	1	0	0	0, 1
有效	1	1	0	1	1	0	0, 1

● 如果READY 信号在同步模式中被采样(USEREADY=1, READYMODE1), 那么:

1. 建立: $LR \geq t_{C(XTIM)}$

$LW \geq t_{C(XTIM)}$

2. 有效: $AR \geq 2 \times t_{C(XTIM)}$

$AW \geq 2 \times t_{C(XTIM)}$

注释: 限制条件不包括外部硬件等待状态。

3. 建立+ 有效: $LR + AR \geq 4 \times t_{C(XTIM)}$

$LW + AW \geq 4 \times t_{C(XTIM)}$

注释: 限制条件不包括外部硬件等待状态。

这些配置引起以下这些 XTIMING 寄存器配置限制 (没有硬件检测非法 XTIMING 配置):

XRDLEAD	XRDACTIVE	XRDTRAIL	XWRLEAD	XWRACTIVE	XWRTRAIL	X2TIMING
≥ 1	≥ 2	0	≥ 1	≥ 2	0	0, 1

或者 (没有硬件检测非法 XTIMING 配置)

XRDLEAD	XRDACTIVE	XRDTRAIL	XWRLEAD	XWRACTIVE	XWRTRAIL	X2TIMING
≥ 2	≥ 1	0	≥ 2	≥ 1	0	0, 1

当使用同步 READY 时, 有效和无效时序示例 (没有硬件检测非法 XTIMING 配置):

	XRDLEAD	XRDACTIVE	XRDTRAIL	XWRLEAD	XWRACTIVE	XWRTRAIL	X2TIMING
无效	0	0	0	0	0	0	0, 1
无效	1	0	0	1	0	0	0, 1
无效	1	1	0	1	1	0	0
有效	1	1	0	1	1	0	1
有效	1	2	0	1	2	0	0, 1
有效	2	1	0	2	1	0	0, 1

除非另外注明, 否则所有 XINTF 时序适用于表 5-31 中显示的时钟配置。

表 5-31. XINTF 时钟配置

模式	SYSCLKOUT	XTIMCLK	CLKOUT
1 示例 :	150MHz	SYSCLKOUT 150MHz	SYSCLKOUT 150MHz
2 示例 :	150MHz	SYSCLKOUT 150MHz	1/2SYSCLKOUT 75MHz
3 示例 :	150MHz	1/2 SYSCLKOUT 75MHz	1/2SYSCLKOUT 75MHz
4 示例 :	150MHz	1/2 SYSCLKOUT 75MHz	1/4 SYSCLKOUT 37.5MHz

SYSCLKOUT 和 XTIMCLK 之间的关系显示在图 5-30 中。

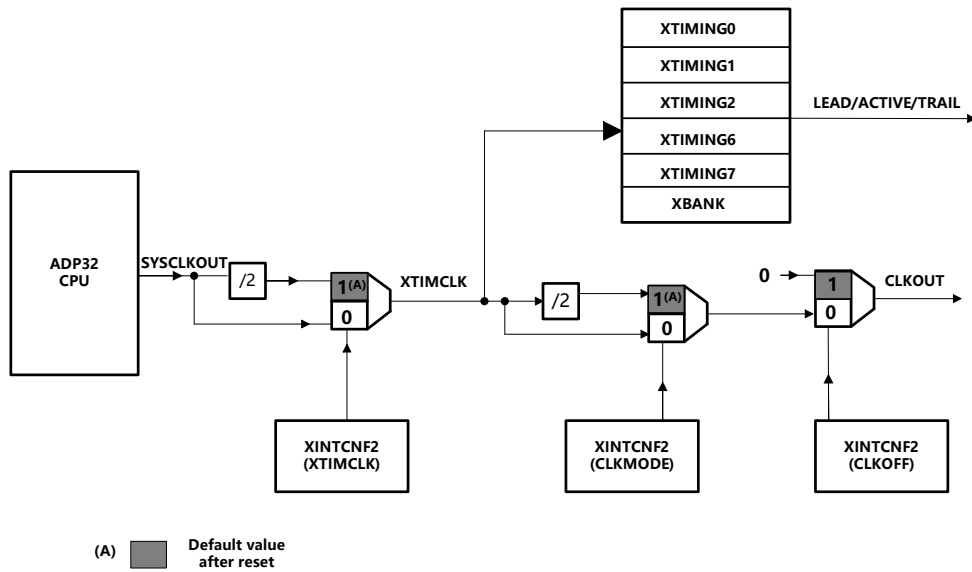


图 5-30. XTIMCLK 和 SYSCLKOUT 之间的关系

5.23 XINTF 信号与 CLKOUT 对齐

对于每个XINTF 访问，建立、有效、跟踪周期的数量基于内部时钟XTIMCLK。诸如 \overline{RD} ， \overline{WE} ，和区域芯片选择(\overline{ZCS}) 的选通脉冲的状态改变与XTIMCLK 的上升边沿有关。外部时钟，CLKOUT，可被配置成等于XTIMCLK 周期或者为XTIMCLK 周期的一半。

对于CLKOUT=XTIMCLK 的情况，所有XINTF 选通时钟将相对于CLKOUT 的上升边沿改变状态。对于CLKOUT=XTIMCLK 一半的情况，一些选通脉冲将在CLKOUT 的上升边沿或者CLKOUT 的下降边沿上改变状态。在XINTF 时序表中，符号COHL 被用于表示相对于任一种情况的参数；CLKOUT 上升边沿（高电平）或者CLKOUT 下降边沿（低电平）。如果参数一直相对于CLKOUT 的上升边沿的话，符号XCOH 被使用。

对于CLKOUT=XTIMCLK 一半的情况，基于从访问开始到信号变化发生点的XTIMCLK 周期的数量，

与变化对齐的CLKOUT 边沿可被确定。如果这个XTIMCLK 周期的数量为偶数，对齐将相对于CLKOUT 的上升边沿。如果这个XTIMCLK 周期的数量为奇数，那么信号将相对于CLKOUT 的下降边沿发生变化。示例包括如下：

- 在一个访问开始时发生变化的选通脉冲一直与CLKOUT 的上升边沿对齐。这是因为所有XINTF 方位相对于CLKOUT 的上升边沿开始。

示例： ZCSL区域芯片选择低电平有效

$\overline{\text{RNWLR}}/\overline{\text{W}}$ 低电平有效

- 如果用于访问的建立XTIMCLK 周期为偶数，在一个有效周期开始时发生变化的选通脉冲将与CLKOUT 的上升边沿对齐。如果建立XTIMCLK 周期的数量为偶数，那么对齐将相对于CLKOUT 的下降边沿。

示例： $\overline{\text{RDL}} \overline{\text{RD}}$ 低电平有效

$\overline{\text{WEL}} \overline{\text{WE}}$ 低电平有效

- 如果用于访问的建立和有效XTIMCLK 周期总数（包括硬件等待状态）为偶数，在一个跟踪周期开始时发生变化的选通脉冲将与CLKOUT 的上升边沿对齐。如果建立和有效XTIMCLK 周期的数量（包括硬件等待状态）为奇数，那么对齐将相对于CLKOUT 的下降边沿。

示例： $\overline{\text{RDHRD}}$ 高电平无效

$\overline{\text{WEH}} \overline{\text{WE}}$ 高电平无效

- 如果建立和有效加上跟踪XTIMCLK 周期总数（包括硬件等待状态）为偶数，在一个访问末尾发生变化的选通脉冲将与CLKOUT 的上升边沿对齐。如果建立和有效加上跟踪XTIMCLK 周期的数量（包括硬件等待状态）为奇数，那么对齐将相对于CLKOUT 的下降边沿。

示例： ZCSH区域芯片选择高电平无效

$\overline{\text{RNWH}} \overline{\text{R}}/\overline{\text{W}}$ 高电平无效

5.24 外部接口读取时序

表 5-32 . 外部内存接口读取开关特性

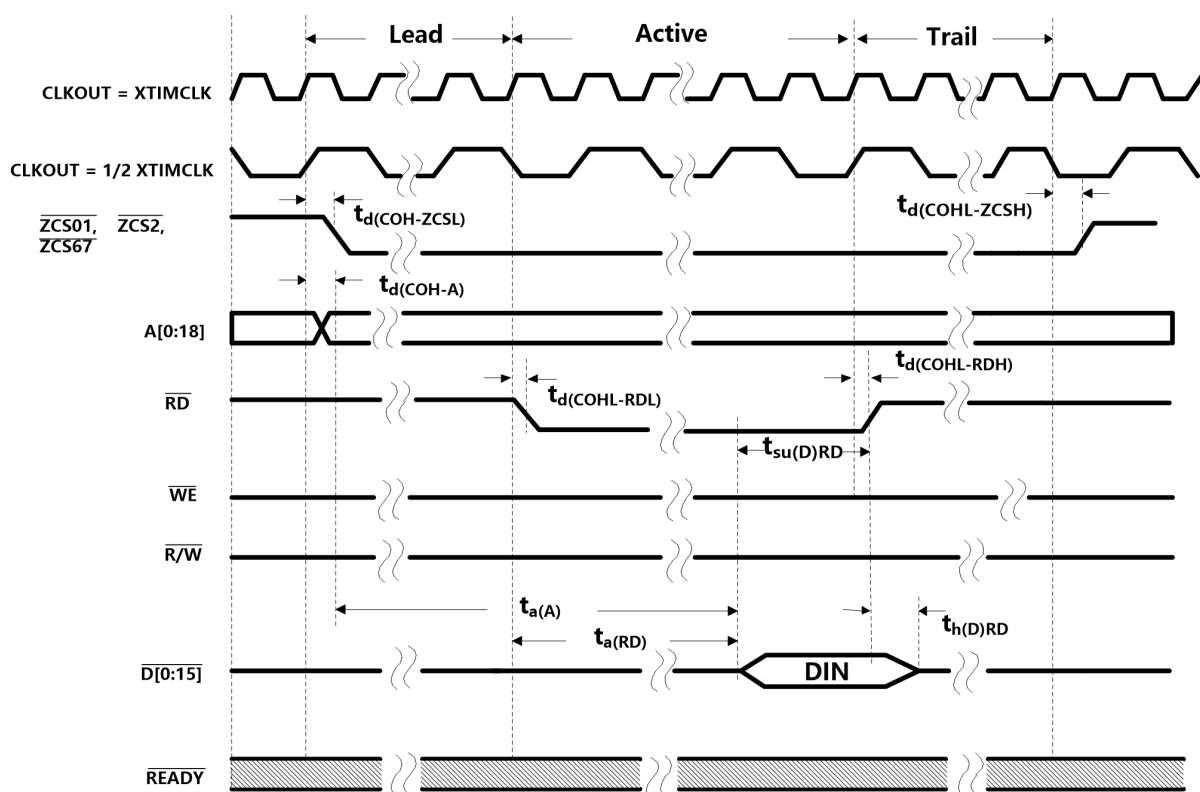
参数	最小值	最大值	单位
$t_{d(COH-ZCS)}$ 延迟时间, CLKOUT 高电平到区域芯片选择低电平有效的时间		1	ns
$t_{d(COHL-ZCSH)}$ 延迟时间, CLKOUT 高电平/低电平到芯片选择高电平无效的时间	-2	3	ns
$t_{d(COH-A)}$ 延迟时间, CLKOUT 高电平到地址有效的时间		2	ns
$t_{d(COHL-RDL)}$ 延迟时间, CLKOUT 高电平/低电平到 \overline{RD} 低电平有效的时间		1	ns
$t_{d(COHL-RDH)}$ 延迟时间, CLKOUT 高电平/低电平到 \overline{RD} 高电平无效的时间	-2	1	ns
$t_{h(A)ZCSH}$ 保持时间, 区域芯片选择高电平无效之后地址有效的时间	(1)		ns
$t_{h(A)RD}$ 保持时间, \overline{RD} 高电平无效之后地址有效的时间	(1)		ns

(1) 在无效周期期间, XINTF 地址总线将一直保持总线输出的最后一个地址。这个包括对齐周期。

表 5-33. 外部存储器接口读取时序要求

参数	最小值	最大值	单位
$t_{a(A)}$ 访问时间, 从有效地址读取数据的时间		$(LR+AR)-14^{(1)}$	ns
$t_{a(RD)}$ 访问时间, 从 \overline{RD} 低电平有效读取有效数据的时间		$AR-12^{(1)}$	ns
$t_{su(D)RD}$ 建立时间, 在 \overline{RD} 选通脉冲高电平无效之前读取有效数据的时间	12		ns
$t_{h(D)RD}$ 保持时间, 在 \overline{RD} 高电平无效之后读取有效数据的时间	0		ns

(1) LR = 建立周期, 读取访问。AR = 有效周期, 读取访问。请参考表 5-30。



- A. 所有XINTF 访问（建立周期）在CLKOUT 的上升沿上开始。当需要时，器件将在一个满足这个要求的访问之前插入一个对准周期。
- B. 在对准周期期间，所有信号将被转换为它们的未激活状态。
- C. 因为USEREADY=0，外部READY 输入信号被忽略。
- D. 在包括对准周期在内的无效周期期间，A [0:18] 将保持出现在总线上的最后一个地址。

图 5-31. 示例读取访问

XRDLEAD	XRDACTIVE	XRDTRAIL	USEREADY	X2TIMING	XWRLEAD	XWRACTIVE	XWRTRAIL	READY MODE
≥ 1	≥ 0	≥ 0	0	0	N/A ⁽¹⁾	N/A ⁽¹⁾	N/A ⁽¹⁾	N/A ⁽¹⁾

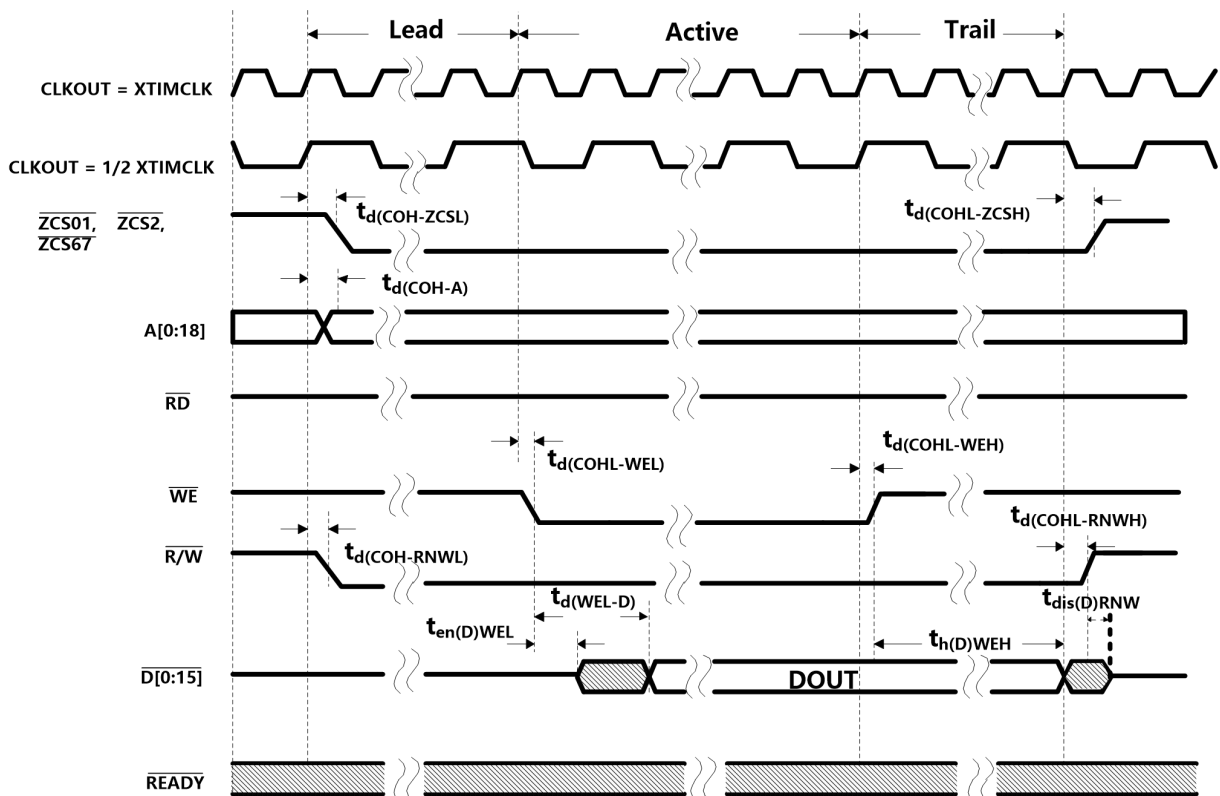
(1) 对于这个示例，N/A = “无关”

5.25 外部接口写入时序

表 5-34. 外部存储器接口写入开关特性

参数	最小值	最大值	单位
$t_{d(COH-ZCSL)}$ 延迟时间, CLKOUT 高电平到区域芯片选择低电平有效的时间		1	ns
$t_{d(COHL-ZCSH)}$ 延迟时间, CLKOUT 高电平或者低电平到区域芯片选择高电平无效的时间	-2	3	ns
$t_{d(COH-A)}$ 延迟时间, CLKOUT 高电平到地址有效的时间		2	ns
$t_{d(COHL-WEL)}$ 延迟时间, CLKOUT 高电平/低电平到 WE 低电平的时间		2	ns
$t_{d(COHL-WEH)}$ 延迟时间, CLKOUT 高电平/低电平到 WE 高电平的时间		2	ns
$t_{d(COH-RNWL)}$ 延迟时间, CLKOUT 高电平到 R/W 低电平的时间		1	ns
$t_{d(COHL-RNWH)}$ 延迟时间, CLKOUT 高电平/低电平到 R/W 高电平的时间	-2	1	ns
$t_{en(D)WEL}$ 使能时间, 从 \overline{WE} 低电平驱动数据总线的时间	0		ns
$t_{d(WEL-D)}$ 延迟时间, \overline{WE} 低电平有效之后数据有效的时间		4	ns
$t_{h(A)ZCSH}$ 保持时间, 区域芯片选择高电平无效之后地址有效时间	(1)		ns
$t_{h(D)WE}$ 保持时间, \overline{WE} 高电平无效之后写入数据有效的时间	TW-2 ⁽²⁾		ns
$t_{dis(D)RNW}$ R/W 高电平无效之后 DSP 释放数据总线的最长时间		4	ns

(1) 无效周期期间, XINTF 地址总线将一直保持总线上产生的最后地址。这包括对准周期。
 (2) TW = 跟踪周期, 写入访问。请参考表5-30。



ADP32FXX Digital Signal Processor

- A. 所有 XINTF 访问（建立周期）在 CLKOUT 的上升沿上开始。当需要时，器件将在一个满足这个要求的访问之前插入一个对准周期。
- B. 在对准周期期间，所有信号将被转换为它们的未激活状态。
- C. 因为 USEREADY=0，外部 READY 输入信号被忽略。
- D. 在包括对准周期在内的无效周期期间，A [0:18] 将保持出现在总线上的最后一个地址。

图 5-32. 示例写入访问

用于这个示例的 XTIMING 寄存器参数：

XRDLEAD	XRDACTIVE	XRDTRAIL	USEREADY	X2TIMING	XWRLEAD	XWRACTIVE	XWRTRAIL	READY MODE
N/A ⁽¹⁾	N/A ⁽¹⁾	N/A ⁽¹⁾	0	0	≥ 1	≥ 0	≥ 0	N/A ⁽¹⁾

(1) 对于这个示例，N/A = “无关”

5.26 带有一个外部等待状态的外部接口读取准备就绪时序

表 5-35. 外部存储器接口读取开关特性（读取准备就绪，1 个等待状态）

参数	最小值	最大值	单位
$t_{d(COHL-ZCSL)}$ 延迟时间，CLKOUT 高电平到区域芯片选择低电平有效的时间		1	ns
$t_{d(COHL-ZCSH)}$ 延迟时间，CLKOUT 高电平/低电平到区域芯片选择高电平无效的时间	-2	3	ns
$t_{d(COHL-A)}$ 延迟时间，CLKOUT 高电平到地址有效的时间		2	ns
$t_{d(COHL-RDL)}$ 延迟时间，CLKOUT 高电平/低电平到 \overline{RD} 低电平有效的时间		1	ns
$t_{d(COHL-RDH)}$ 延迟时间，CLKOUT 高电平/低电平到 \overline{RD} 高电平无效的时间	-2	1	ns
$t_{h(A)ZCSH}$ 保持时间，区域芯片选择高电平无效之后的地址有效时间	(1)		ns
$t_{h(A)RD}$ 保持时间， \overline{RD} 高电平无效之后的地址有效时间	(1)		ns

(1) 无效周期期间，XINTF 地址总线将一直保持总线上产生的最后一个地址。这包括对准周期。

表 5-36. 外部存储器接口读取时序要求（读取就绪，1 个等待状态）

	最小值	最大值	单位
$t_{a(A)}$ 访问时间，从有效地址读取数据的时间		$(LR + AR) - 14^{(1)}$	ns
$t_{a(RD)}$ 访问时间，从 \overline{RD} 低电平有效读取有效数据的时间		(1)	ns
$t_{su(D)RD}$ 建立时间， \overline{RD} 选通脉冲高电平无效前的读取数据有效的时间	12		ns
$t_{h(D)RD}$ 保持时间， \overline{RD} 无效高电平之后读取数据有效的时间	0		ns

(1) LR = 建立周期，读取访问。AR = 有效周期，读取访问。请参考表 5-30。

表 5-37. 同步 READY 时序要求 (读取准备就绪, 1 个等待状态)⁽¹⁾

	最小值	最大值	单位
$t_{su}(RDYsynchronL)COHL$ 建立时间, CLKOUT 高电平/低电平之前 READY (同步) 低电平的时间	15		ns
$t_h(RDYsynchronL)$ 保持时间, READY (同步) 低电平的时间	12		ns
$t_e(RDYsynchronH)$ 采样 CLKOUT 边沿之前 READY (同步) 能够变为高电平的 earliest 时间		3	ns
$t_{su}(RDYsynchronH)COHL$ 建立时间, CLKOUT 高电平/低电平之前 READY (同步) 高电平的时间	15		ns
$t_h(RDYsynchronH)ZCSH$ 保持时间, 区域芯片选择高电平之后 READY (同步) 保持高电平的时间	0		ns

(1) 第一个 READY (同步) 样本相对于图 5-33 中的 E 发生:

$$E = (XRDLEAD + XRDACTIVE) t_{c(XTIM)}$$

当首次采样时, 如果 READY (同步) 被发现为高电平, 那么访问将完成。如果发现 READY (同步) 为低电平, 它将在每个 $t_{c(XTIM)}$ 内被重新采样直到它为高电平。对于每个样本, 相对于访问开始的建立时间(D) 可计算为:

$$D = (XRDLEAD + XRDACTIVE + n - 1) t_{c(XTIM)} - t_{su}(RDYsynchronL)COHL, \text{ 在这里, } n \text{ 为样本数量 } (n = 1, 2, 3, \text{ 以此类推})。$$

表 5-38. 异步 READY 时序要求 (读取准备就绪, 1 个等待状态)⁽¹⁾

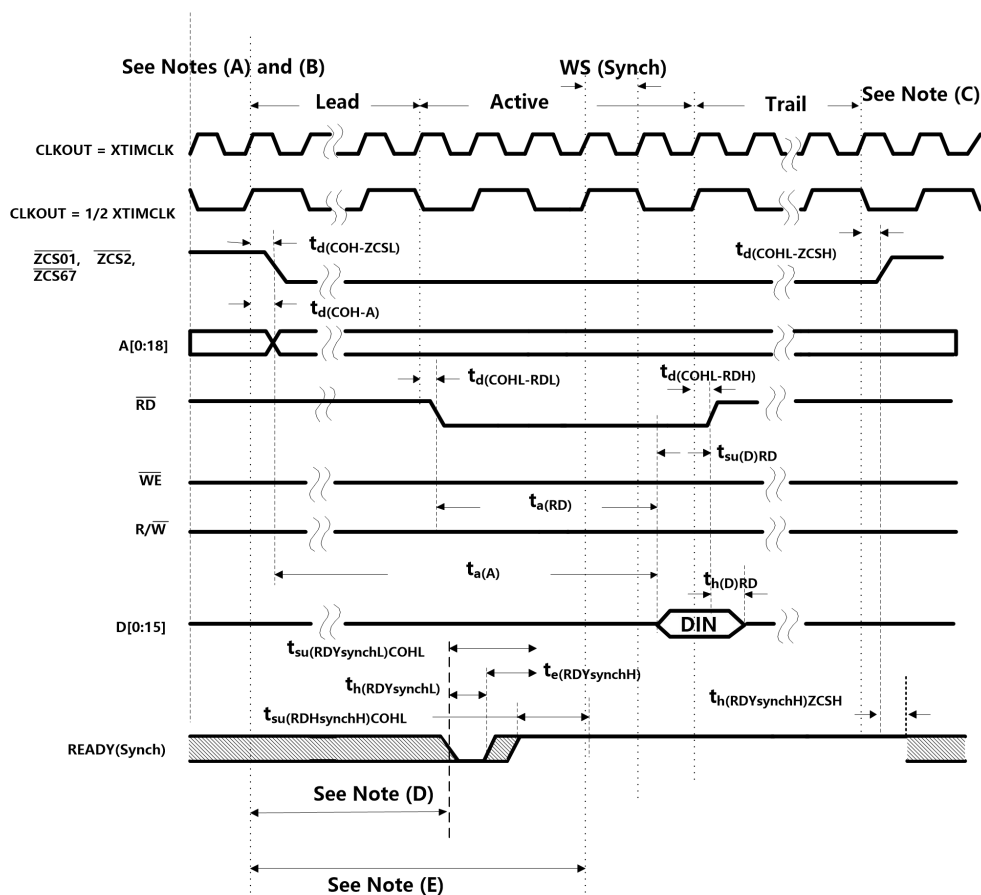
	最小值	最大值	单位
$t_{su}(RDYAsynchL)COHL$ 建立时间, 在 CLKOUT 高电平/低电平之前 READY (异步) 低电平的时间	11		ns
$t_h(RDYAsynchL)$ 保持时间, READY (异步) 低电平的时间	8		ns
$t_e(RDYAsynchH)$ 采样 CLKOUT 边沿之前, READY (异步) 能够变为高电平的 earliest 时间		3	ns
$t_{su}(RDYAsynchH)COHL$ 建立时间, 在 CLKOUT 高电平/低电平之前 READY (异步) 高电平的时间	11		ns
$t_h(RDYAsynchH)ZCSH$ 保持时间, 区域芯片选择高电平之后 READY (异步) 保持高电平的时间	0		ns

(1) 第一个 READY (异步) 样本相对于图 5-34 中的 E 发生:

$$E = (RDLEAD + RDACTIVE - t_{c(XTIM)})$$

当首次采样时, 如果 READY (异步) 被发现为高电平, 那么访问将完成。如果发现 READY (异步) 为低电平, 它将在每个 $t_{c(XTIM)}$ 内被重新采样直到它为高电平。对于每个样本, 从访问开始的建立时间可计算为:

$$D = (XRDLEAD + XRDACTIVE - 3 + n) t_{c(XTIM)} - t_{su}(RDYAsynchL)COHL, \text{ 在这里, } n \text{ 为样本数量 } (n = 1, 2, 3, \text{ 以此类推})。$$



Legend:

= Don't care. Signal can be high or low during this time.

- A. 所有XINTF 访问（建立周期）在CLKOUT 的上升沿上开始。当需要时，器件将在一个满足这个要求的访问之前插入一个对准周期。
- B. 在对准周期期间，所有信号将被转换为它们的未激活状态。
- C. 在未激活周期期间，XINTF 地址总线将一直保持总线上产生的最后一个地址。这包括对准周期。
- D. 对于每一个样本，从访问(D) 开始的建立时间可计算如下： $D = (XRDLEAD + XRDACTIVE + n - 1)t_c(XTIM) - t_{su}(RDYsynchL)COHL$
- E. 关于相对于这个点的第一个样本

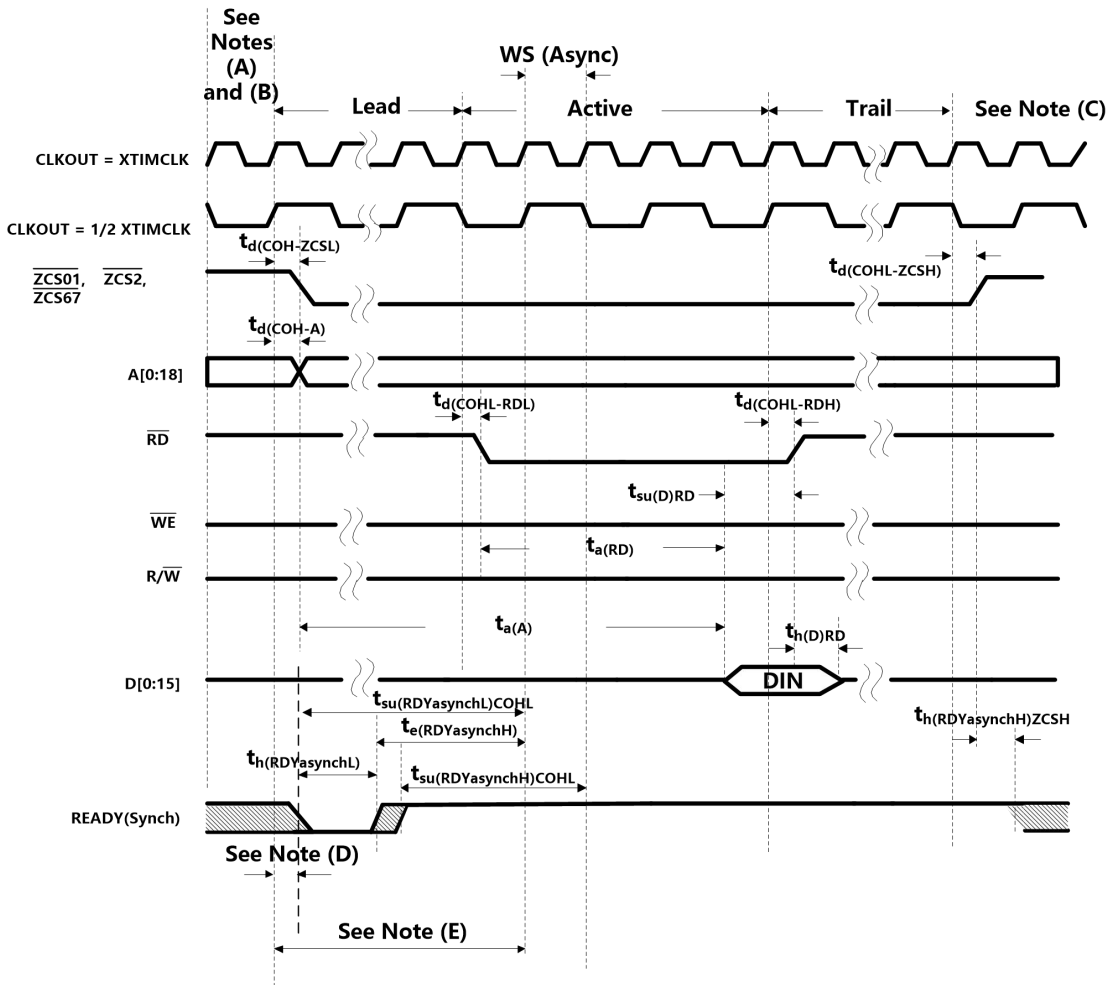
$$E = (XRDLEAD + XRDACTIVE)t_c(XTIM), \text{ 在这里, } n \text{ 为样本数量 } (n = 1, 2, 3, \text{ 以此类推})。$$

图 5-33. 使用同步 READY 访问的样本读取

用于这个示例的 XTIMING 寄存器参数：

XRDLEAD	XRDACTIVE	XRDTRAIL	USEREADY	X2TIMING	XWRLEAD	XWRACTIVE	XWRTRAIL	READYMODE
≥ 1	3	≥ 1	1	0	N/A ⁽¹⁾	N/A ⁽¹⁾	N/A ⁽¹⁾	0 = READY (同步)

(1) 在这个示例中，N/A = “无关”



Legend:

= Don't care. Signal can be high or low during this time.

- A. 所有XINTF 访问（建立周期）在CLKOUT 的上升沿上开始。当需要时，器件将在一个满足这个要求的访问之前插入一个对准周期。
- B. 在对准周期期间，所有信号将被转换为它们的未激活状态。
- C. 在未激活周期期间，XINTF 地址总线将一直保持总线上产生的最后一个地址。这包括对准周期。
- D. 对于每个示例，相对于访问开始的建立时间可计算为： $D=(XRDLEAD+XRDACTIVE-3+n) t_{c(XTIM)}-t_{su(RDYasynchL)COHL}$ ，在这里，n 为样本数量（n = 1, 2, 3，以此类推）。
- E. 关于相对于这个点的第一个样本： $E=(XRDLEAD+XRDACTIVE-2) t_{c(XTIM)}$

图 5-34. 使用异步 READY 访问的样本读取

用于这个示例的 XTIMING 寄存器参数：

XRDLEAD	XRDACTIVE	XRDTRAIL	USEREADY	X2TIMING	XWRLEAD	XWRACTIVE	XWRTRAIL	READY MODE
≥ 1	3	≥ 1	1	0	N/A ⁽¹⁾	N/A ⁽¹⁾	N/A ⁽¹⁾	1 =READY (异步)

(1) 在这个示例中，N/A = “无关”

5.27 带有一个外部等待状态的外部接口写入准备就绪时序

表 5-39. 外部存储器接口写入开关特性（写入准备就绪，1 个等待状态）

参数	最小值	最大值	单位
$t_{d(COH-ZCSL)}$ 延迟时间, CLKOUT 高电平到区域芯片选择低电平有效的时间		1	ns
$t_{d(COHL-ZCSH)}$ 延迟时间, CLKOUT 高电平或低电平到区域芯片选择高电平无效的时间	-2	3	ns
$t_{d(COH-A)}$ 延迟时间, CLKOUT 高电平到地址有效的时间		2	ns
$t_{d(COHL-WEL)}$ 延迟时间, CLKOUT 高电平/低电平到 \overline{WE} 低电平的时间		2	ns
$t_{d(COHL-WEH)}$ 延迟时间, CLKOUT 高电平/低电平到 \overline{WE} 高电平的时间		2	ns
$t_{d(COH-RNWL)}$ 延迟时间, CLKOUT 高电平到 R/ \overline{W} 低电平的时间		1	ns
$t_{d(COHL-RNWH)}$ 延迟时间, CLKOUT 高电平/低电平到 R/W 高电平的时间	-2	1	ns
$t_{en(D)WEL}$ 使能时间, 从 \overline{WE} 低电平驱动数据总线的时间	0		ns
$t_{d(WEL-D)}$ 延迟时间, \overline{WE} 低电平有效之后数据有效的时间		4	ns
$t_{h(A)ZCSH}$ 保持时间, 区域芯片选择高电平无效之后地址有效的时间	(1)		ns
$t_{h(D)WE}$ 保持时间, \overline{WE} 高电平无效之后写入数据有效的时间	TW-2(2)		ns
$t_{dis(D)RNW}$ DSP 在 \overline{WE} 高电平无效之后释放数据总线的最长时间		4	ns

(1) 在未激活周期期间, XINTF 地址总线将一直保持总线上产生的最后一个地址。这包括对准周期。

(2) TW = 跟踪周期, 写入访问。请参考表 5-30。

表 5-40. 同步 READY 时序要求（写入准备就绪，1 个等待状态）⁽¹⁾

	最小值	最大值	单位
$t_{su(RDYsynchL)COHL}$ 建立时间, 在 CLKOUT 高电平/低电平之前 READY (同步) 低电平的时间	15		ns
$t_{h(RDYsynchL)}$ 保持时间, READY (同步) 低电平的时间	12		ns
$t_{e(RDYsynchH)}$ 采样 CLKOUT 边沿之前, READY (同步) 能够变为高电平的 earliest 时间		3	ns
$t_{su(RDYsynchH)COHL}$ 建立时间, 在 CLKOUT 高电平/低电平之前 READY (同步) 高电平的时间	15		ns
$t_{h(RDYsynchH)ZCSH}$ 保持时间, 区域芯片选择高电平后 READY (同步) 保持高电平的时间	0		ns

(1) 第一个 READY (同步) 采样相对于图 5-35 中的 E 发生:

$E = (XWRLEAD + XWRACTIVE) t_{c(XTIM)}$ 当首次采样时, 如果 READY (同步) 被发现为高电平, 那么访问将完成。如果发现 READY (同步) 为低电平, 它将在每个 $t_{c(XTIM)}$ 内被重新采样直到它为高电平。

对于每个示例, 从访问开始的建立时间可计算为:

$D = (XWRLEAD + XWRACTIVE + n - 1) t_{c(XTIM)} - t_{su(RDYsynchL)COHL}$, 在这里, n 为样本数量 (n = 1, 2, 3, 以此类推)。

表 5-41. 异步 READY 时序要求（写入准备就绪，1 个等待状态）⁽¹⁾

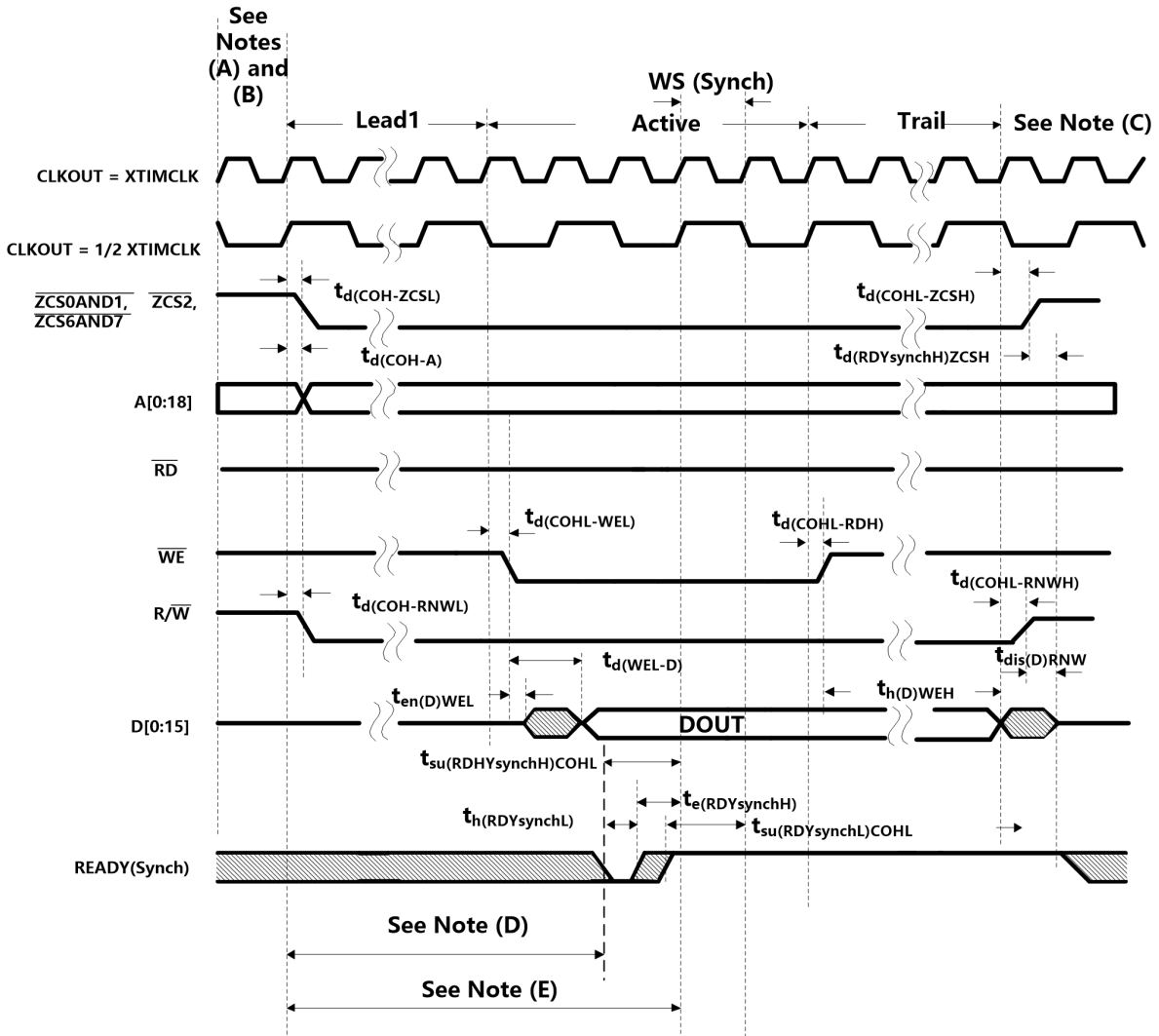
	最小值	最大值	单位
$t_{su(RDYasynchL)COHL}$ 建立时间, 在 CLKOUT 高电平/低电平之前 READY (异步) 低电平的时间	11		ns
$t_{h(RDYasynchL)}$ 保持时间, READY (异步) 低电平的时间	8		ns
$t_{e(RDYasynchH)}$ 采样 CLKOUT 边沿之前, READY (异步) 能够变为高电平的 earliest 时间		3	ns
$t_{su(RDYasynchH)COHL}$ 建立时间, 在 CLKOUT 高电平/低电平之前 READY (异步) 高电平的时间	11		ns
$t_{h(RDYasynchH)ZCSH}$ 保持时间, 区域芯片选择高电平之后 READY (异步) 保持高电平的时间	0		ns

(1) 第一个 READY (同步) 采样相对于图 5-36 中的 E 发生:

$E = (XWRLEAD + XWRACTIVE - 2) t_{c(XTIM)}$ 当首次采样时, 如果 READY (异步) 被发现为高电平, 那么访问将完成。如果发现 READY (异步) 为低电平, 它将在每个 $t_{c(XTIM)}$ 内被重新采样直到它为高电平。

对于每个样本, 访问开始的建立时间可计算为:

$D = (XWRLEAD + XWRACTIVE - 3 + n) t_{c(XTIM)} - t_{su(RDYsynchL)COHL}$, 在这里, n 为样本数量 (n = 1, 2, 3, 以此类推)。



Legend:
 = Don't care. Signal can be high or low during this time.

图 5-35. 使用同步 READY 访问的写入

- A. 所有XINTF 访问（建立周期）在CLKOUT 的上升沿上开始。当需要时，器件将在一个满足这个要求的访问之前插入一个对准周期。
- B. 在对准周期期间，所有信号将被转换为它们的未激活状态。
- C. 在未激活周期期间，XINTF 地址总线将一直保持总线上产生的最后一个地址。这包括对准周期。
- D. 对于每个样本，访问开始的建立时间可计算为：

$$D = (XWRLEAD + XWRACTIVE + n - 1) t_{c(XTIM)} - t_{su(RDYsynchL)COHL}$$

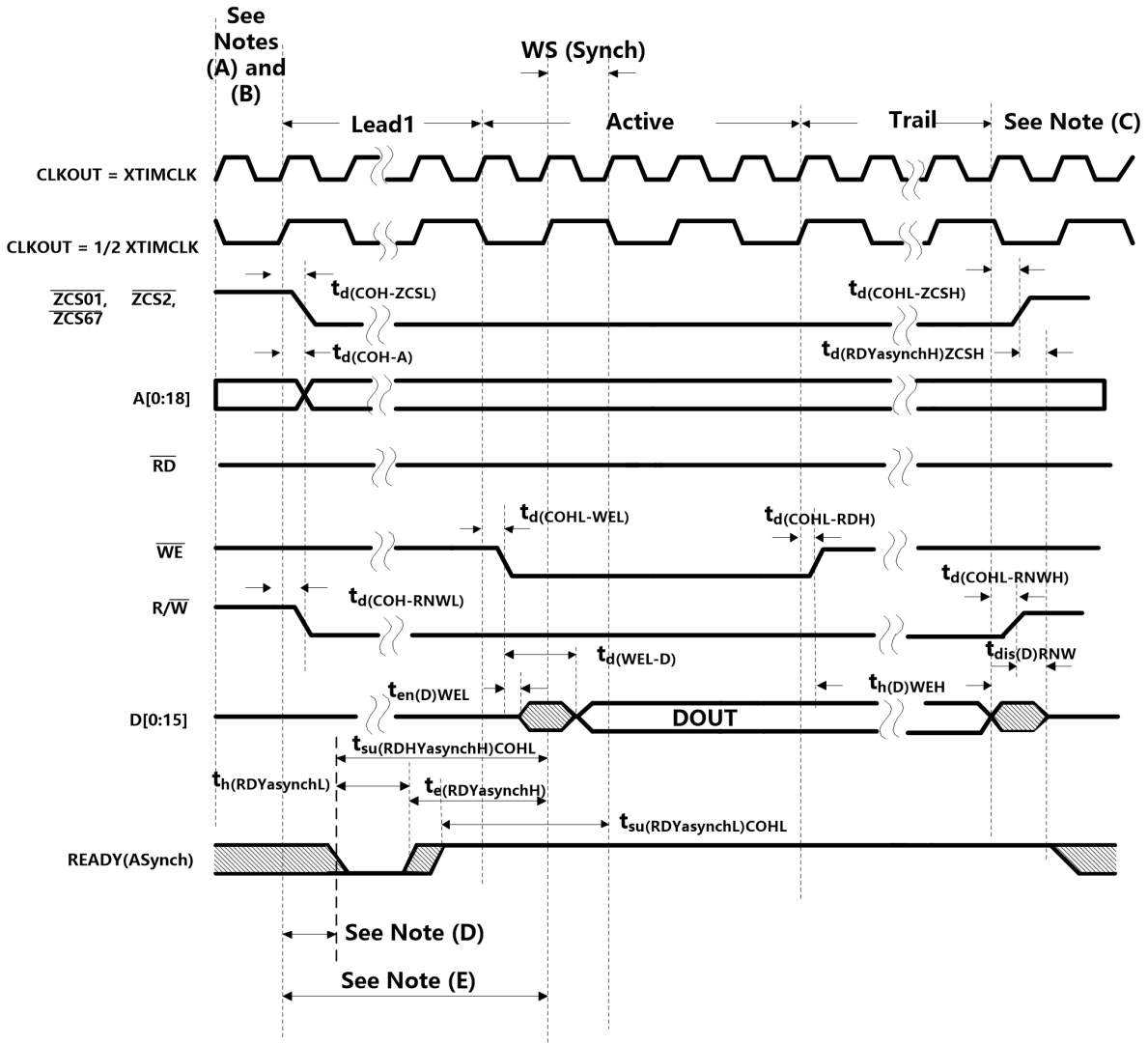
在这里，n 为样本数量（n = 1, 2, 3, 以此类推）。

- E. 关于相对于这个点的第一个样本，E = (XWRLEAD + XWRACTIVE) t_{c(XTIM)}

用于这个示例的 XTIMING 寄存器参数：

XRDLEAD	XRDACTIVE	XRDTRAIL	USEREADY	X2TIMING	XWRLEAD	XWRACTIVE	XWRTRAIL	READYMODE
N/A(1)	N/A(1)	N/A(1)	1	0	≥ 1	3	≥ 1	0 = READY (同步)

(1) 对于这个样本，N/A = “无关”



Legend:
 = Don't care. Signal can be high or low during this time.

图 5-36. 使用异步 READY 访问的写入

- A. 所有XINTF 访问（建立周期）在CLKOUT 的上升沿上开始。当需要时，器件将在一个满足这个要求的访问之前插入一个对准周期。
- B. 在对准周期期间，所有信号将被转换为它们的未激活状态。
- C. 在未激活周期期间，XINTF 地址总线将一直保持总线上产生的最后一个地址。这包括对准周期。
- D. 对于每个样本，访问开始的建立时间可计算为：

$$D = (XWRLEAD + XWRACTIVE - 3 + n) t_c(XTIM) - t_{su}(XRDYasynchL)XCOHL, \text{ 在这里, } n \text{ 为样本数量 } (n = 1, 2, 3, \text{ 以此类推})。$$

- E. 关于相对于这个点的第一个样本， $E = (XWRLEAD + XWRACTIVE - 2) t_c(XTIM)$ 。

用于这个示例的 XTIMING 寄存器参数：

XRDLEAD	XRDACTIVE	XRDTRAIL	USEREADY	X2TIMING	XWRLEAD	XWRACTIVE	XWRTRAIL	READYMODE
N/A(1)	N/A(1)	N/A(1)	1	0	≥ 1	3	≥ 1	0 = READY (异步)

(1) 对于这个样本，N/A = “无关”

5.28 HOLD 和 HOLDA

如果在 $\overline{\text{HOLD}}$ 和 $\overline{\text{HOLDA}}$ 同时为低电平时（授权外部总线访问）HOLD模式位被设定， $\overline{\text{HOLDA}}$ 信号被强制为高电平（在当前周期的末尾）并且外部接口不再为高阻抗模式。

复位时($\overline{\text{RS}}$)，HOLD 模式位被设定为0。如果 $\overline{\text{HOLD}}$ 信号在系统复位时为低电平有效，总线和所有信号选通必须为高阻抗模式，并且 $\overline{\text{HOLDA}}$ 信号也被驱动为低电平有效。

当HOLD 模式被启用并且 $\overline{\text{HOLDA}}$ 为低电平有效（外部总线置为有效），CPU 仍然可以从内部存储器执行代码。如果对外部接口进行访问，在 $\overline{\text{HOLD}}$ 信号被去除前，CPU 暂停。一个内部DMA 请求，当被批准时，将以下信号置为高阻抗模式：

$\overline{\text{A[18:0]ZCS01}}$

$\overline{\text{D[15:0]ZCS2}}$

$\overline{\text{WE}}$ ， $\overline{\text{RDZCS67}}$

$\overline{\text{R/W}}$

在这些信号事件期间，所有在这个组中未列出的其它信号保持在它们的缺省值或者功能运行模式。

5.29 HOLD 和 HOLDA 时序

表 5-42. $\overline{\text{HOLD}}/\overline{\text{HOLDA}}$ 时序要求(CLKOUT=XTIMCLK)^{(1) (2)}

	最小值	最大值	单位
$t_{d(\text{HL-HiZ})}$ 延迟时间， $\overline{\text{HOLD}}$ 低电平到所有地址、数据、和控制上的低阻抗到高阻抗的时间		$4t_{c(\text{XTIM})}$	ns
$t_{d(\text{HL-HAL})}$ 延迟时间， $\overline{\text{HOLD}}$ 低电平至 $\overline{\text{HOLDA}}$ 低电平的时间		$5t_{c(\text{XTIM})}$	ns
$t_{d(\text{HH-HAH})}$ 延迟时间， $\overline{\text{HOLD}}$ 高电平至 $\overline{\text{HOLDA}}$ 高电平的时间		$3t_{c(\text{XTIM})}$	ns
$t_{d(\text{HH-BV})}$ 延迟时间， $\overline{\text{HOLD}}$ 高电平到总线有效的的时间		$4t_{c(\text{XTIM})}$	ns

(1) 当在 $\overline{\text{HOLD}}$ 上检测到一个低电平信号时，所有等待的XINTF 访问将在总线被置为一个高阻抗状态前完成。

(2) $\overline{\text{HOLD}}$ 的状态被锁存在 XTIMCLK 的上升边沿上。

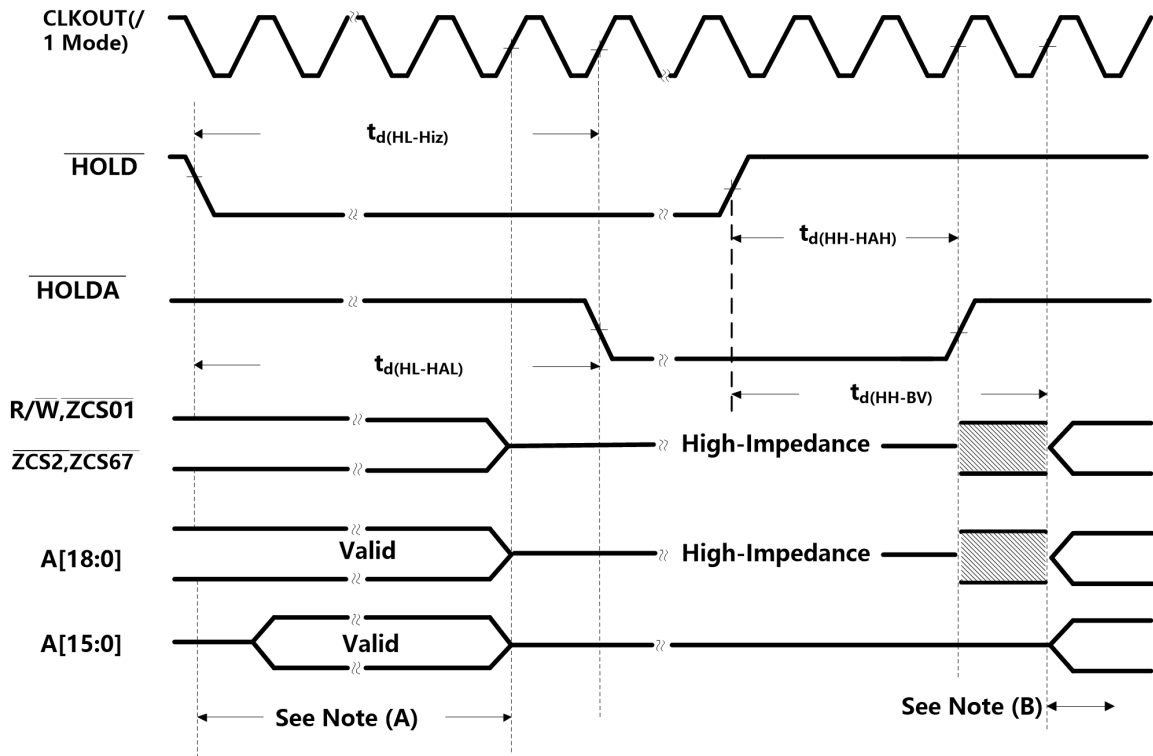


图 5-37. 外部接口保持波形
A. 所有等待中的

XINTF 访问被完成。
B. 正常 XINTF 运行重新开始。

表 5-43. XHOLD/XHOLDA 时序要求($XCLKOUT = 1/2 XTIMCLK$)^{(1) (2) (3)}

	最小值	最大值
$t_{d(HL-Hiz)}$ 延迟时间, HOLD 低电平到所有地址、数据、和控制上的低阻抗到高阻抗的时间		$4t_{c(XTIM)} + t_{c(XCO)}$
$t_{d(HL-HAL)}$ 延迟时间, HOLD 低电平至 HOLDA 低电平的时间		$4t_{c(XTIM)} + 2t_{c(XCO)}$
$t_{d(HH-HAH)}$ 延迟时间, HOLD 高电平至 HOLDA 高电平的时间		$4t_{c(XTIM)}$
$t_{d(HH-BV)}$ 延迟时间, HOLD 高电平到总线有效的的时间		$6t_{c(XTIM)}$

- 当在HOLD上检测到一个低电平信号时, 所有等待的XINTF 访问将在总线被置为一个高阻抗状态前完成。
- HOLD的状态被锁存在XTIMCLK 的上升边沿上。
- 在HOLD被检测为高电平或者低电平时, 所有总线转换和HOLDA 转换将相对于CLKOUT 的上升边沿发生。因此, 对于这个 $CLKOUT=1/2 XTIMCLK$ 的模式, 转换最多可以早于最大额定值1 XTIMCLK 周期发生

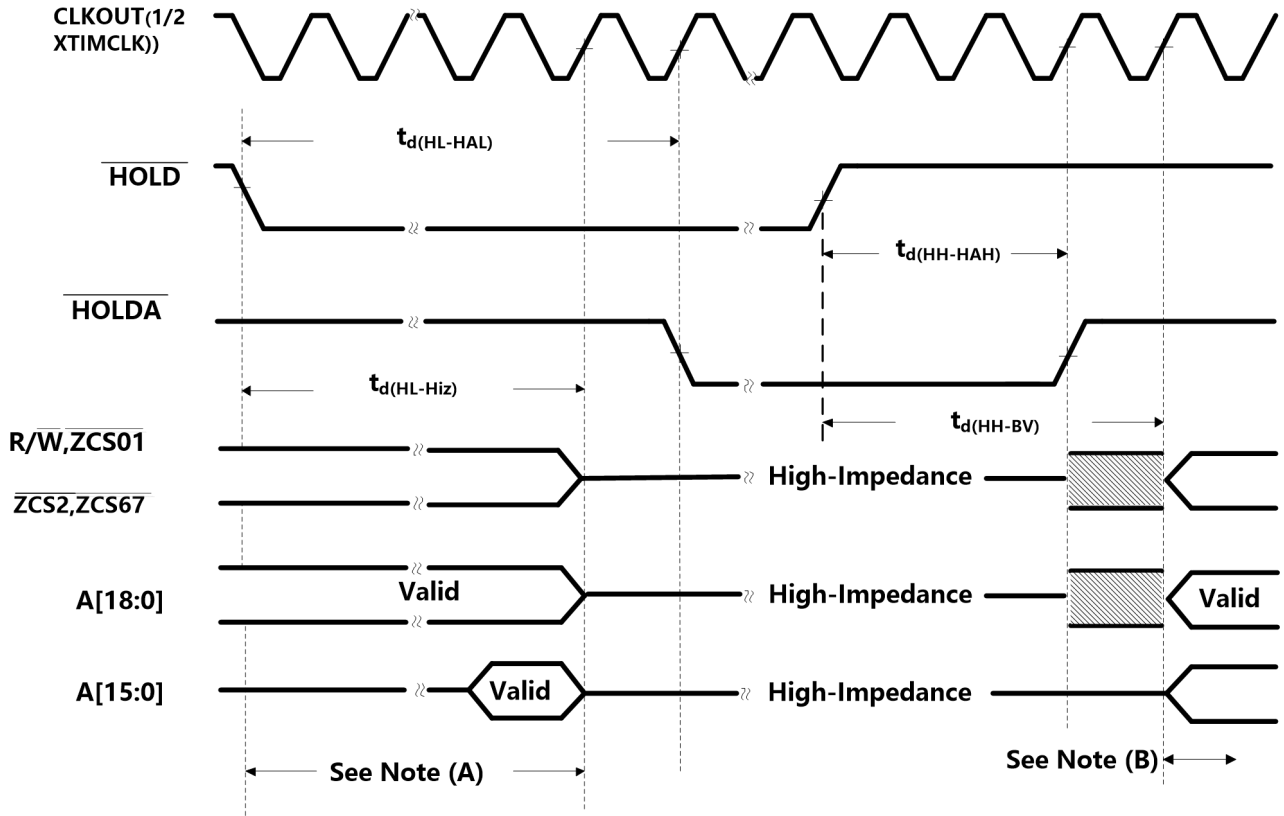


图5-38. HOLD/HOLDA时序要求(CLKOUT = 1/2 XTIMCLK)

- A. 所有等待中的XINTF 访问被完成。
- B. 正常XINTF 运行重新开始。

5.30 片载模数转换器

5.30.1 ADC 绝对最大额定值

除非另外注明，绝对最大额定值列表中的值在运行条件下指定。在超过那些绝对最大额定值下列出的应力下运行有可能会对器件造成永久损坏。它们只是应力额定值。长时间运行在最大绝对额定条件下会影响设备的可靠性。

电源电压范围 (V_{SSA1}/V_{SSA2} 至 $V_{DDA1}/V_{DDA2}/AVDDREFBG$)

-0.3V 至 4.6V

电源电压范围 (V_{SS1} 至 V_{DD1})

-0.3V 至 2.5V

模拟输入(ADCIN) 钳制电流，总数 (最大值)

$\pm 20\text{mA}^{(1)}$

- (1) 这个模拟输入有一个内部钳制电路，此电路能够将电压固定在一个高于 V_{DDA} 或者低于 V_{SS} 的二极管压降。每个引脚上的持续钳制电流为 $\pm 2\text{mA}$ 。

5.30.2 在推荐运行条件下的 ADC 电气特性

表 5-44. DC 技术规范⁽¹⁾

参数	最小值	典型值	最大值	单位
分辨率	12			位
ADC 时钟 ⁽²⁾	1		25	KHZ MHZ
精度				
INL (积分非线性) ⁽³⁾	1-18.75 MHz ADC 时钟		±1.5	最低有效位(LSB)
DNL (微分非线性) ⁽³⁾	1-18.75 MHz ADC 时钟		±1	LSB
偏移误差 ⁽⁴⁾			-80 80	LSB
带有内部基准的总增益误差 ⁽⁵⁾			-200 200	LSB
带有内部基准的总增益误差 ⁽⁶⁾	如果 VREFP- VREFM=1V±0.1%		-50 50	LSB
通道到通道偏移变化		±8		LSB
通道到通道增益变化		±8		LSB
模拟输入				
模拟输入电压 (ADCINx 至 ADCLO) ⁽⁷⁾	0		3	V
ADCLO	-5	0	5	mV
输入电容		10		pF
输入漏电流		3	±5	µA
内部电压基准 ⁽⁵⁾				
精度, VREFP	1.90	2	2.10	V
精度, VREFM	0.90	1	1.10	V
电压差异, VREFP-VREFM		1		V
温度系数		50		PPM/°C
基准噪声		100		µV
外部电压基准 ⁽⁶⁾				
精度, VREFP	1.90	2	2.10	V
精度, VREFM	0.90	1	1.10	V
输入电压差异, VREFP-VREFM	0.99	1	1.01	V

(1) 在12.5MHz ADCCLK 上测试。

(2) 如果SYSCLKOUT ≤ 25MHz, ADC 时钟 ≤ SYSCLKOUT/2。

(3) 对于高于18.75 MHz–25 MHz 的频率, INL 降级。要求这些采样速率的应用应该使用一个20K 电阻器作为RESEXT 引脚上的偏置电阻器。这提升了总体线性并且取自ADC 的电流将比24.9kΩ 偏置电流多几个mA。对于1-25MHz 的全范围, ADP32FXX器件中的ADC 模块能够运行在RESEXT 引脚上的20kΩ 偏置电流上。

(4) 1 LSB 有3.0/4096=0.732mV 的加权值。

(5) 一个单一内部带隙基准 (±5% 精度) 为VREFP 和VREFM 信号供源, 因此, 这些电压可一起跟踪。ADC 转换器使用这两个之间的差异作为它的基准。总体增益误差将是这里显示的增益误差与电压基准精度(VREFP-VREFM) 的组合。

(6) 在这个模式下, 外部基准的精度对于总体增益很关键。电压差异 (VREFP-VREFM) 将确定总体精度。

(7) 应用到一个模拟输入引脚上的高于V_{DDA}+0.3V 或者低于V_{SS}-0.3V 的电压有可能暂时影响另外引脚的转换。为了避免这种情况, 模拟输入应该被保持在这些限值内。

表 5-45. AC 技术规范

ADP32FXX Digital Signal Processor

参数	最小值	典型值	最大值	单位
SINAD 信噪比+ 失真		62		dB
SNR 信噪比		62		dB
总谐波失真(THD) (100kHz) 总谐波失真		-68		dB
ENOB (SNR) 有效位数		10.1		位
SFDR 无杂散动态范围		69		dB

5.30.3 针对不同 ADC 配置的流耗

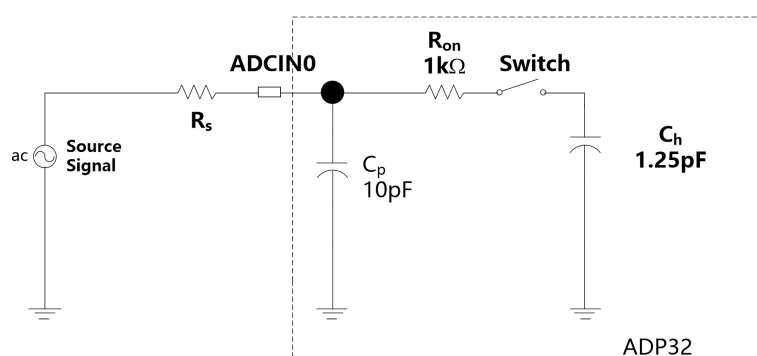
表 5-46. 针对不同 ADC 配置的流耗 (在 25MHz ADCCLK 上)

IDDA (典型值(2))	IDDAIO (典型值)	IDD1 (典型值)	ADC 运行模式/条件
35mA	1 μ A	0.5mA	模式 A (运行模式) • BG REF 被启用 • PWD 被禁用
7mA	0	5 μ A	模式 B: • ADC 时钟被启用 • BG 和 REF 被启用 • PWD 被启用
1 μ A	0	5 μ A	模式 C: • ADC 时钟被启用 • BG 和 REF 被禁用 • PWD 被启用
1 μ A	0	0	模式 D: • ADC 时钟被禁用 • BG 和 REF 被禁用 • PWD 被启用

(1) 测试条件:

- SYSCLKOUT = 150MHz
- ADC 模块时钟 = 25MHz
- ADC 在模式A 中执行全部16 个通道的连续转换

IDDA-包括进入 V_{DDA1}/V_{DDA2} 和 $AVDDREFBG$ 的电流



Typical Values of the Input Circuit Components:

Switch Resistance (R_{on}): 1k Ω

Sampling Capacitor (C_h): 1.25 pF

Parasitic Capacitance (C_p): 10 pF

Source Resistance (R_s): 50 Ω

图 5-39. ADC 模拟输入阻抗模型

5.30.4 ADC 加电控制位时序

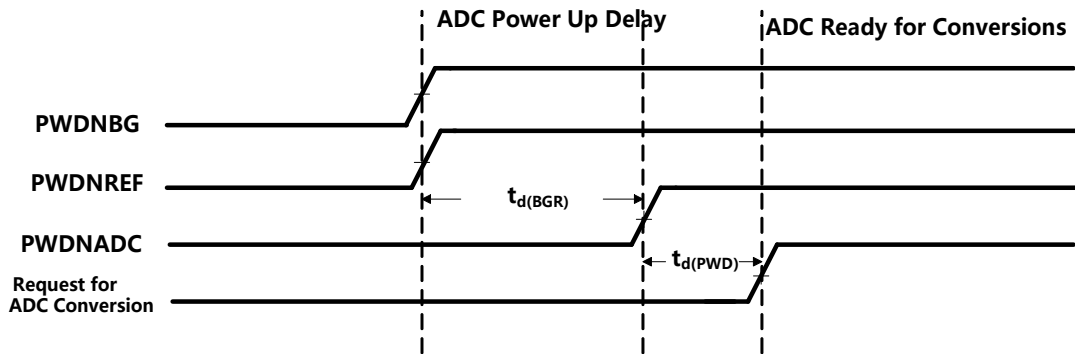


图 5-40. ADC 加电控制位时序

表 5-47. ADC 加电延迟⁽¹⁾

	最小值	典型值	最大值	单位
$t_{d(BGR)}$ 带隙基准稳定所需的延迟时间。ADCTRL3 寄存器的位 7 和 6 (ADCBGRFDN1/0) 在 ADCPWDN 位被启用前被设定为 1。	23	25	28	ms
$t_{d(PWD)}$ 断电控制稳定所需的延迟时间。ADCTRL3 寄存器的位 5 (ADCPWDN) 在任何 ADC 转换启动前被设定为 1。	20	50		μ s
		1		ms

(1) 这些延迟是必须的并且建议在转换启动之前使ADC 模拟基准电路稳定。如果转换在没有这些延迟的情况下启动，那么ADC 的结果将显示出一个更改的增益。断电时，所有三个位被同时清零。

5.30.5 详细说明

5.30.5.1 基准电压

片载ADC 有一个内置基准，这个基准为ADC 提供了基准电压。VREFP 被设定为2.0V，而VREFM 被设定为1.0V。

5.30.5.2 模拟输入

片载ADC 由16 个模拟输入组成，这些通道或者同时采样，或者每次两个通道采样。这些输入为软件可选。

5.30.5.3 转换器

片载ADC 使用一个12 位四级流水线架构，此架构可在低功耗时实现高采样率。

5.30.5.4 转换时间

转换可以在两个不同的转换模式中执行：

- 顺序采样模式(SMODE = 0)
- 同步采样模式(SMODE= 1)

5.30.6 顺序采样模式 (单通道) (SMODE = 0)

在顺序采样模式下，ADC 能够持续在任一通道 (Ax 至Bx) 上转换输入信号。ADC 能够在来自事件管理器(EM1/EM2)、软件触发器、或者从一个外部ADCSOC 信号的事件触发上启动转换。如果 SMODE 位为0，ADC 将在每个采样/保持脉冲上的所选通道上进行转换。下面对转换时间和结果寄存器更新的延迟进行解释说明。ADC 中断标志在结果寄存器更新之后的几个SYSCLKOUT 周期内被设定。所选通道将在采样/保持脉冲的每个下降边沿上被采样。采样/保持脉冲宽度可被设定为1 个ADC 时钟宽 (最小值) 或者16个ADC 时钟宽 (最大值)

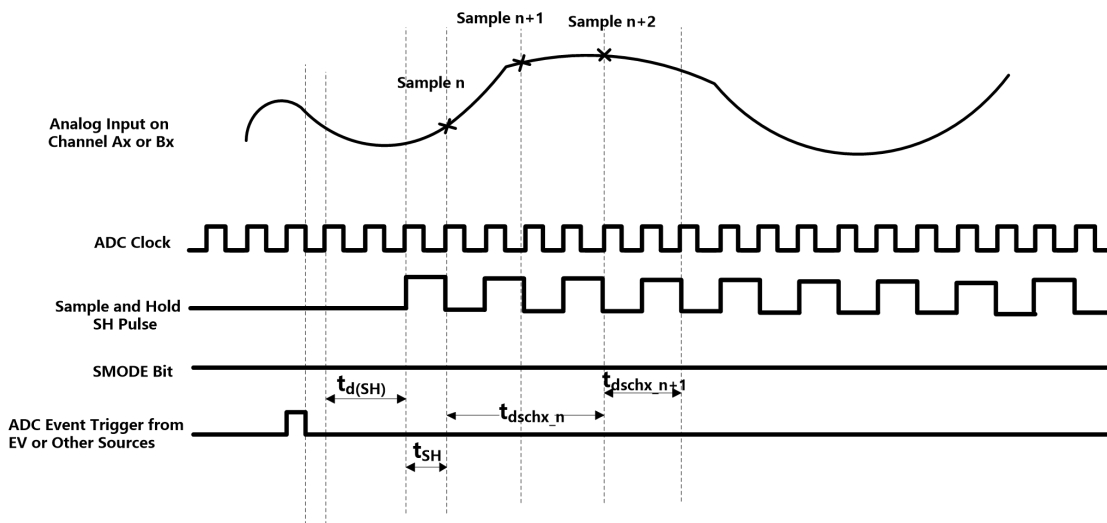


图5-41. 顺序采样模式 (单通道) 时序

表5-48. 顺序采样模式时序

		SAMPLE (样本) n	SAMPLE n + 1	在 25MHz ADC 时钟上， $t_{c(ADCCLK)}=40ns$	注释
$t_{d(SH)}$	从事件触发器到采样的 延迟时间	$2.5t_{c(ADCCLK)}$			
t_{SH}	采样 保持宽度 采集宽度	$(1 + Acqps) * t_{c(ADCCLK)}$		Acqps=0 时为 40ns	Acqps 值=0- 15ADCTRL1[8:11]
$t_{d(schx_n)}$	结果寄存器出现第一个 结果的延迟时间	$4t_{c(ADCCLK)}$		160ns	
$t_{d(schx_{n+1})}$	结果寄存器中出现连 续结果的延迟时间		$(2 + Acqps) * t_{c(ADCCLK)}$	80ns	

5.30.7 同步采样模式 (双通道) (SMODE=1)

在同步模式中，ADC 可在任何一对通道 (A0/B0 至A7/B7) 持续转换输入信号。ADC 能够在来自事件管理器(EM1/EM2)、软件触发器、或者从一个外部ADCSOC 信号的事件触发上启动转换。如果 SMODE 位为1，ADC 将在每个采样/保持脉冲上的两个所选通道上进行转换。下面对转换时间和结果寄存器更新的延迟进行解释说明。ADC 中断标志在结果寄存器更新之后的几个SYSCLKOUT 周期内被设定。所选通道将在采样/保持脉冲的下降边沿上被同时采样。采样/保持脉冲宽度可被设定为1个ADC 时钟宽 (最小值) 或者16 个ADC 时钟宽 (最大值。)

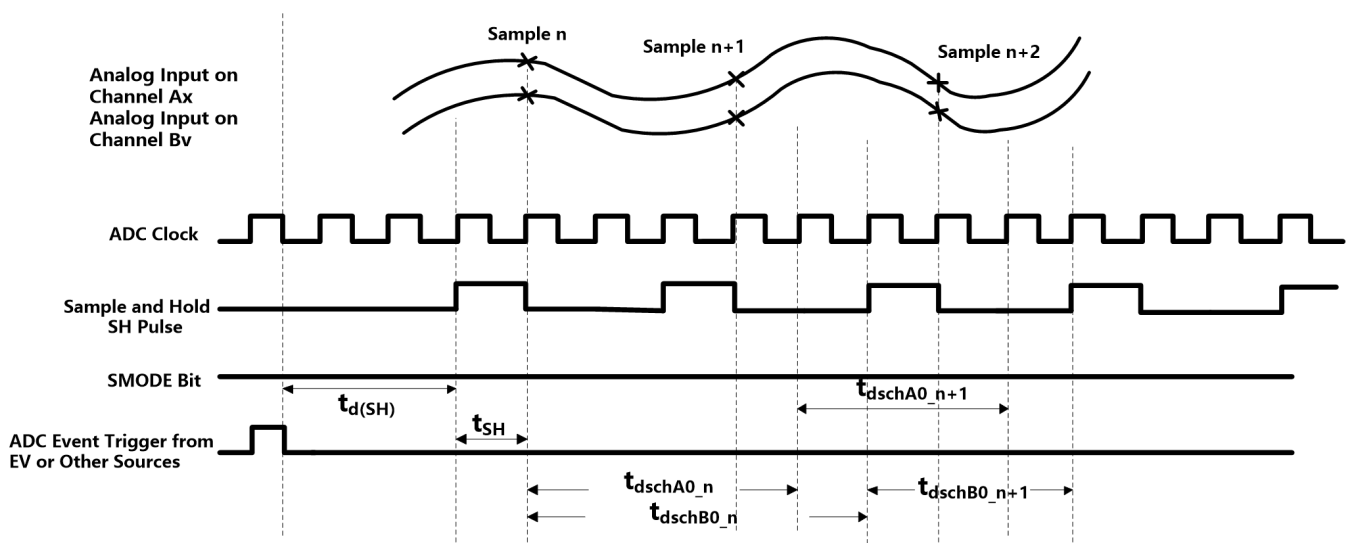


图5-42. 同步采样模式时序

表 5-49. 同步采样模式时序

		SAMPLE n	SAMPLEn+1	在 25MHz 时 ADC 时钟， $t_{c(ADCCLK)}=40ns$	注释
$t_{d(SH)}$	从事件触发到采样的延迟时间	$2.5t_{c(ADCCLK)}$			
t_{SH}	采样/保持宽度/采集宽度	$(1 + Acqps) * t_{c(ADCCLK)}$		当 $Acqps=0$ 时为 40ns	$Acqps$ 值= 0-15ADCTRL1[8:11]
$t_{d(schA0_n)}$	结果寄存器出现第一个结果的延迟时间	$4t_{c(ADCCLK)}$		160ns	
$t_{d(schB0_n)}$	结果寄存器出现第一个结果的延迟时间	$5t_{c(ADCCLK)}$		200ns	

$t_{d(schA0_n+1)}$	结果寄存器中出现连续结果的延迟时间	$(3 + Acqps) * t_{c(ADCCLK)}$	120ns
$t_{d(schB0_n+1)}$	结果寄存器中出现连续结果的延迟时间	$(3 + Acqps) * t_{c(ADCCLK)}$	120ns

5.30.8 技术规范和术语的定义

积分非线性

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

微分非线性

一个理想ADC 显示分开距离恰好为1 个LSB 的代码转换。DNL 是从这个理想值的偏离。少于±1 LSB 的微分非线性误差可确保无丢码。

零偏移

当模拟输入为零伏时，应当发生主进位转换。零误差被定义为实际转换到那个点的偏离。

增益误差

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

信噪比+失真(SINAD)

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

有效位数(ENOB)

对于一个正弦波，SINAD 可用位的数量表示。使用下面的公式，

$$N = \frac{(SINAD - 1.76)}{6.02}$$

有可能获得一个用N（位的有效数）表达的性能测量值。因此，对于在给定输入频率上用于正弦波输入的器件的有效位数量可从这个测得的SINAD 直接计算。

总谐波失真(THD)

THD 是头流个谐波分量的均方根总和与测得的输入信号的均方根值的比并表达为一个百分比或者分贝值。

无杂散动态范围(SFDR)

SFDR 是输入信号均方根振幅与峰值寄生信号间以分贝为单位的差异。

5.31 多通道缓冲串行端口(McBSP) 模块

5.31.1 McBSP 发送和接收时序

表5-50. McBSP时序时要求^{(1) (2)}

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

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

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

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

表 5-51. McBSP 开关特性^{(1) (2)}

编号	参数	位置	最小值	最大值	单位
M1	$t_{c(CKRX)}$ 周期时间, CLKR/X	CLKR/X 内部	2P		ns
M2	$t_{w(CKRXH)}$ 脉冲持续时间, CLKR/X 高电平的时间	CLKR/X 内部	D-5 ⁽³⁾	D+5 ⁽³⁾	ns
M3	$t_{w(CKRXL)}$ 脉冲持续时间, CLKR/X 低电平的时间	CLKR/X 内部	C-5 ⁽³⁾	C+5 ⁽³⁾	ns
M4	$t_{d(CKRH-FRV)}$ 延迟时间, CLKR 高电平到内部 FSR 有效的的时间	CLKR 内部	0	4	ns
		CLKR 外部	3	27	
M5	$t_{d(CKXH-FXV)}$ 延迟时间, CLKX 高电平到内部 FSX 有效的的时间	CLKX 内部	0	4	ns
		CLKX 外部	3	27	
M6	$t_{dis(CKXH-DXHZ)}$ 禁用时间, CLKX 高电平到 DX 在最后一个数据位后为高阻抗的时间	CLKX 内部		8	ns
		CLKX 外部		14	
M7	$t_{d(CKXH-DXV)}$ 延迟时间, CLKX 高电平到 DX 有效的的时间。这应用于除了第一个位之外的所有被发送的位。	CLKX 内部		9	ns
		CLKX 外部		28	

ADP32FXX Digital Signal Processor

		延迟时间, CLKX 高电平到 DX 有效的 时间。当处于数据延迟 1 或者 2 (XDATDLY= 01b 或者 10b) 模式时, 只应用于发送的第一个位。	DXENA=0	CLKX 内部		8	
			DXENA=1	CLKX 外部		14	
			DXENA=0	CLKX 内部		P+8	
			DXENA=1	CLKX 外部		P+14	
M8	$t_{en}(CKXH-DX)$	使能时间, CLKX 高电平待 DX 被驱动的 时间。当处于数据延迟 1 或者 2 (XDATDLY= 01b 或者 10b) 模式时, 只应用于发送的第一个位。	DXENA=0	CLKX 内部	0		ns
			DXENA=1	CLKX 外部	6		
			DXENA=0	CLKX 内部	P		
			DXENA=1	CLKX 外部	P+6		
M9	$t_d(FXH-DXV)$	延迟时间, FSX 高电平到 DX 有效的时 间。当处于数据延迟 0 (XDATDLY=00b) 模式时, 只应用于 发送的第一个位。	DXENA=0	FSX 内部		8	ns
			DXENA=1	FSX 外部		14	
			DXENA=0	FSX 内部		P+8	
			DXENA=1	FSX 外部		P+14	
M10	$t_{en}(FXH-DX)$	使能时间, FSX 高电平到 DX 被驱动的时 间。当处于数据延迟 0 (XDATDLY=00b) 模式时, 只应用于 发送的第一个位。	DXENA=0	FSX 内部	0		ns
			DXENA=1	FSX 外部	6		
			DXENA=0	FSX 内部	P		
			DXENA=1	FSX 外部	P+6		

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

(2) $2P=1/CLKG$, 单位为 ns。

(3) $C=CLKRX$ 低脉冲宽度= P

$D = CLKRX$ 高脉冲宽度= P

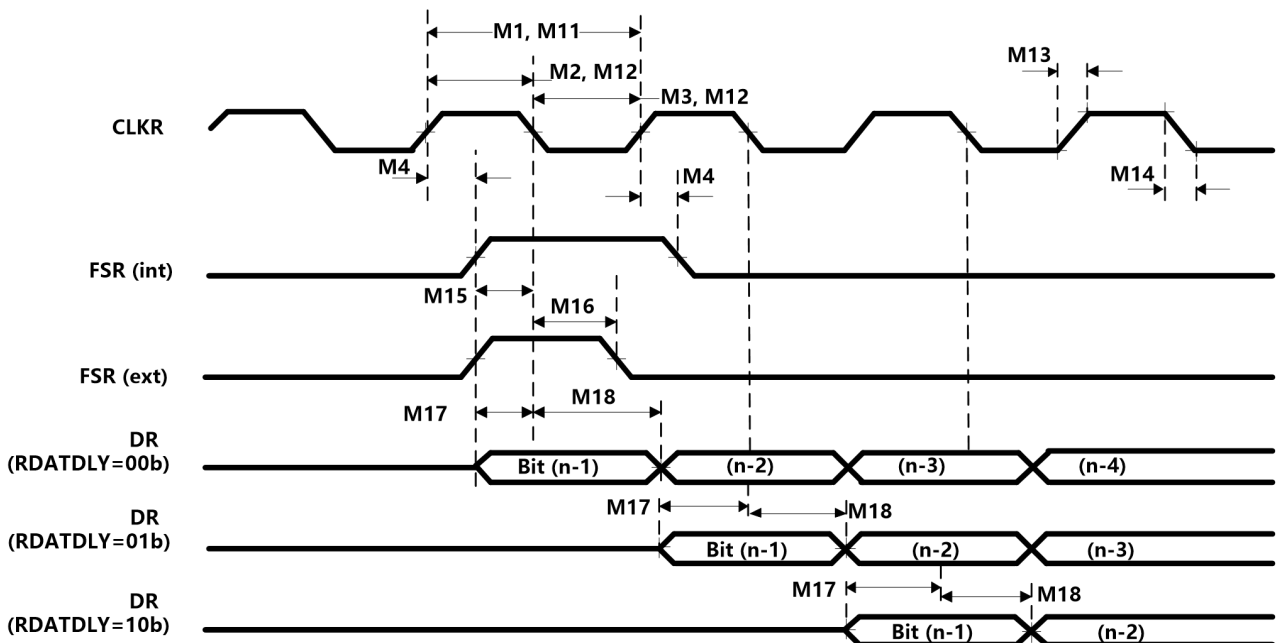


图 5-43. McBSP 接收时序

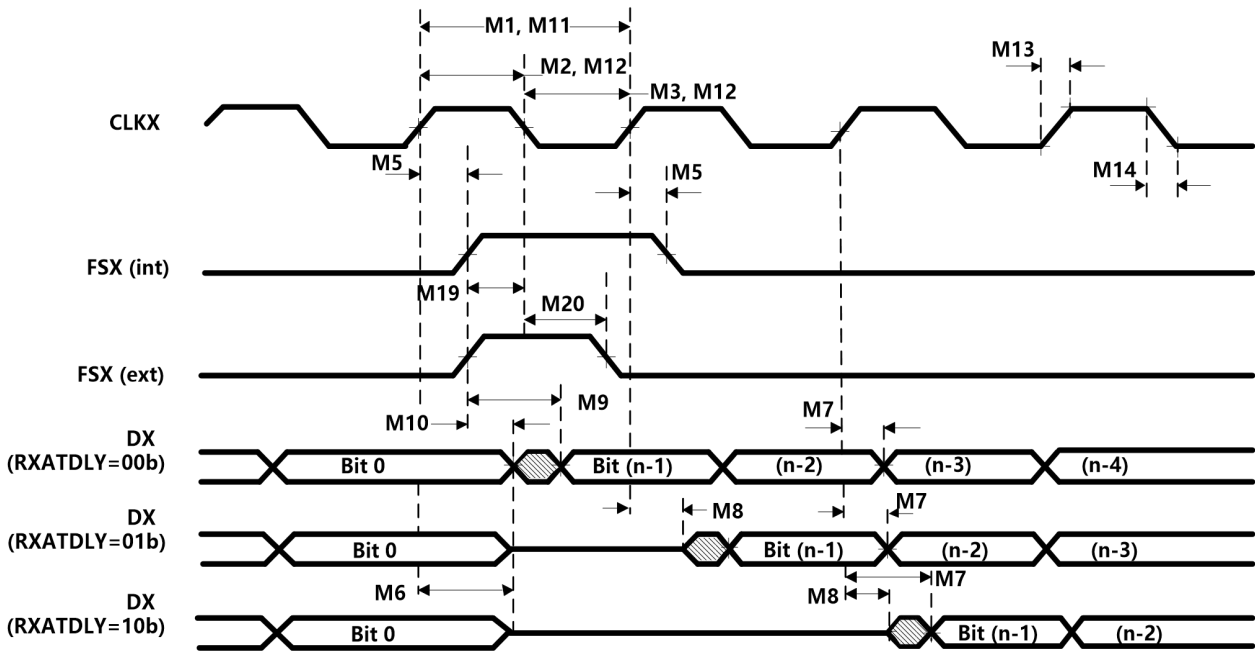


图 5-44. McBSP 发送时序

5.31.2 McBSP 作为 SPI 主控或者受控时序

表 5-52. McBSP 作为 SPI 主控或者受控时序要求(CLKSTP=10b, CLKXP=0)⁽¹⁾

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

(1) 2P=1/CLKG。对于所有SPI 受控模式, CLKX 必须至少为8个CLKG周期。此外, 通过设置CLKSM=CLKGDV=1, CLKG 应该为 LSPCLK//2。借助于75MHz 的最大LSPCLK 速度, CLKX 最大频率为LSPCLK/16, 即4.6875MHz 且P=13.3ns。

表 5-53. McBSP 作为 SPI 主控或者受控开关特性(CLKSTP= 10b, CLKXP= 0)⁽¹⁾

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

(1) 2P=1/CLKG。对于所有SPI 受控模式, CLKX 必须至少为8 个CLKG 周期。此外, 通过设置CLKSM=CLKGDV=1, CLKG 应该为 LSPCLK//2。借助于75MHz 的最大LSPCLK 速度, CLKX 最大频率为LSPCLK/16, 即4.6875MHz 且P=13.3ns。

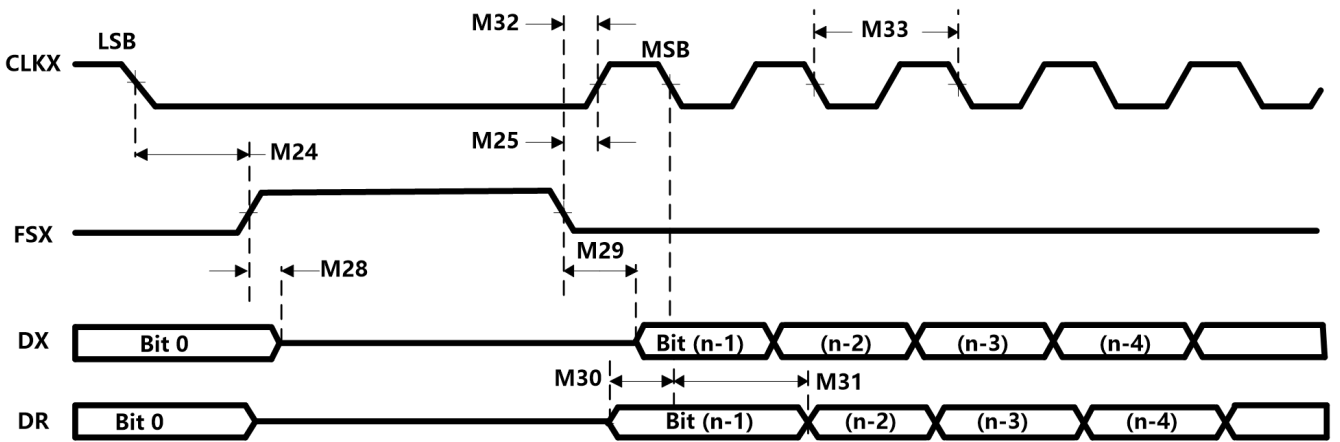


图 5-45. 作为 SPI 主控或者受控时的 McBSP 时序 : CLKSTP=10b, CLKXP=0

表 5-54. 作为 SPI 主控或者受控时的 McBSP 时序要求(CLKSTP=11b, CLKXP=0)⁽¹⁾

编号	参数	主控		受控		单位
		最小值	最大值	最小值	最大值	
M39	$t_{su(DRV-CKXH)}$ 建立时间, CLKX 高电平前, DR 有效时间	30		8P-10		ns
M40	$t_{h(CKXH-DRV)}$ 保持时间, CLKX 高电平后, DR 有效时间	1		8P-10		ns
M41	$t_{su(FXL-CKXH)}$ 建立时间, CLKX 高电平前, FSX 为低电平的时间			16P+10		ns
M42	$t_c(CKX)$ 周期时间, CLKX	2P		16P		ns

(1) 2P=1/CLKG。对于所有SPI 受控模式, CLKX 必须至少为8 个CLKG 周期。此外, 通过设置CLKSM=CLKGDV=1, CLKG 应该为 LSPCLK/2。借助于75MHz 的最大LSPCLK 速度, CLKX 最大频率为LSPCLK/16, 即4.6875MHz 且P=13.3ns。

表 5-55. McBSP 作为 SPI 主控或者受控时的开关特性(CLKSTP= 11b, CLKXP= 0)⁽¹⁾

编号	参数	主控		受控		单位
		最小值	最大值	最小值	最大值	
M34	$t_{h(CKXL-FXL)}$ 保持时间, CLKX 低电平后, FSX 为高电平的时间	P				ns
M35	$t_{d(FXL-CKXH)}$ 延迟时间, FSX 低电平到 CLKX 为高电平的时间	2P				ns
M37	$t_{dis(CKXL-DXHZ)}$ 禁用时间, 从 CLKX 低电平到最后一个数据位后的 DX	P+6		7P+6		ns

ADP32FXX Digital Signal Processor

高阻抗的时间				
M38	$t_{d(FXL-DXV)}$ 延迟时间, FSX 低电平到 DX 有效时的时间	6	4P+6	ns

(1) $2P=1/CLKG$ 。对于所有SPI 受控模式, CLKX 必须至少为8 个CLKG 周期。此外, 通过设置CLKSM=CLKGDV=1, CLKG 应该为 LSPCLK/2。借助于75MHz 的最大LSPCLK 速度, CLKX 最大频率为LSPCLK/16, 即4.6875MHz 且P=13.3ns。

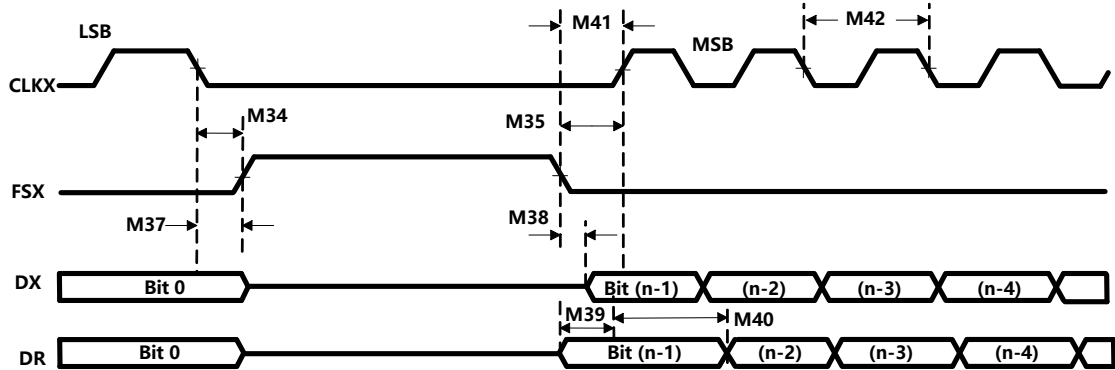


图5-46. 作为SPI 主控或者受控时的McBSP 时序: CLKSTP= 11b, CLKXP= 0

表 5-56. McBSP 作为 SPI 主控或者受控时的时序要求(CLKSTP= 10b, CLKXP= 1)⁽¹⁾

编号	参数	主控		受控		单位
		最小值	最大值	最小值	最大值	
M49	$t_{su(DRV-CKXH)}$ 建立时间, CLKX 高电平前, DR 的有效时间	30		8P-10		ns
M50	$t_{h(CKXH-DRV)}$ 保持时间, CLKX 高电平后, DR 的有效时间	1		8P-10		ns
M51	$t_{su(FXL-CKXL)}$ 建立时间, CLKX 低电平前, FSX 为低电平的时间			8P+10		ns
M52	$t_{c(CKX)}$ 周期时间, CLKX	2P		16P		ns

(1) $2P=1/CLKG$ 。对于所有SPI 受控模式, CLKX 必须至少为8 个CLKG 周期。此外, 通过设置CLKSM=CLKGDV=1, CLKG 应该为 LSPCLK/2。借助于75MHz 的最大LSPCLK 速度, CLKX 最大频率为LSPCLK/16, 即4.6875MHz 且P=13.3ns。

表 5-57. McBSP 作为 SPI 主控或者受控时的开关特性(CLKSTP= 10b, CLKXP= 1)⁽¹⁾

编号	参数	主控		受控		单位
		最小值	最大值	最小值	最大值	
M43	$t_{h(CKXH-FXL)}$ 保持时间, CLKX 高电平后, FSX 为低电平的时间	2P				ns
M44	$t_{d(FXL-CKXL)}$ 延迟时间, FSX 低电平时间到 CLKX 低电平的时间	P				ns
M47	$t_{dis(FXH-DXHZ)}$ 禁用时间, 从 FSX 高电平到最后一个数据位后 DX 高阻抗的时间	6		6P+6		ns
M48	$t_{d(FXL-DXV)}$ 延迟时间, FSX 低电平到 DX 有效时的时间	6		4P+6		ns

(1) $2P=1/CLKG$ 。对于所有SPI 受控模式, CLKX 必须至少为8 个CLKG 周期。此外, 通过设置CLKSM=CLKGDV=1, CLKG 应该为 LSPCLK/2。借助于75MHz 的最大LSPCLK 速度, CLKX 最大频率为LSPCLK/16, 即4.6875MHz 且P=13.3ns。

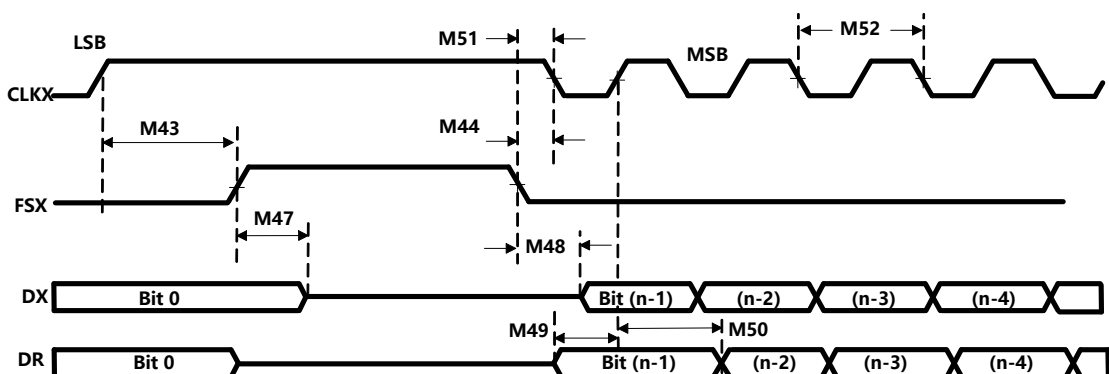


图5-47. 作为SPI 主控或者受控时的McBSP 时序：CLKSTP= 10b，CLKXP= 1

表 5-58. McBSP 作为 SPI 主控或者受控时的时序要求(CLKSTP= 11b，CLKXP= 1)⁽¹⁾

编号	参数	主控		受控		单位
		最小值	最大值	最小值	最大值	
M58	$t_{su(DRV-CKXL)}$ 建立时间，在 CLKX 低电平之前，DR 的有效时间	30		8P-10		ns
M59	$t_h(CKXL-DRV)$ 保持时间，在 CLKX 低电平之后，DR 的有效时间	1		8P-10		ns
M60	$t_{su(FXL-CKXL)}$ 建立时间，CLKX 低电平前，FSX 为低电平的时间			16P+10		ns
M61	$t_c(CKX)$ 周期时间，CLKX	2P		16P		ns

(1) $2P=1/CLKG$ 。对于所有SPI 受控模式，CLKX 必须至少为8个CLKG周期。此外，通过设置CLKSM=CLKGDV=1，CLKG 应该为LSPCLK/2。借助于75MHz 的最大LSPCLK 速度，CLKX 最大频率为LSPCLK/16，即4.6875MHz 且P=13.3ns。

表 5-59. McBSP 作为 SPI 主控或者受控时的开关特性(CLKSTP= 11b，CLKXP= 1)⁽¹⁾

编号	参数	主控		受控		单位
		最小值	最大值	最小值	最大值	
M53	$t_h(CKXH-FXL)$ 保持时间，CLKX 高电平后，FSX 为低电平的时间	P				ns
M54	$t_d(FXL-CKXL)$ 延迟时间，FSX 低电平时间到 CLKX 低电平的时间	2P				ns
M55	$t_d(CLKXH-DXV)$ 延迟时间，CLKX 高电平到 DX 有效的时间	-2	0	3P+6	5P+20	ns
M56	$t_{dis}(CKXH-DXHZ)$ 禁用时间，从 CLKX 高电平到最后一个数据位后的DX 高阻抗的时间	P+6		7P+6		ns
M57	$t_d(FXL-DXV)$ 延迟时间，FSX 低电平到 DX 有效的时间	6		4P+6		ns

(1) $2P=1/CLKG$ 。对于所有SPI 受控模式，CLKX 必须至少为8个CLKG周期。此外，通过设置CLKSM=CLKGDV=1，CLKG 应该为LSPCLK/2。借助于75MHz 的最大LSPCLK 速度，CLKX 最大频率为LSPCLK/16，即4.6875MHz 且P=13.3ns。

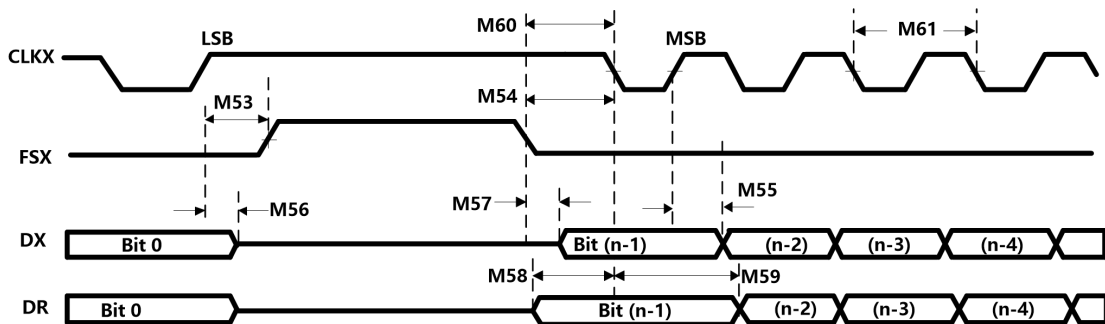


图5-48. 作为SPI 主控或者受控时的McBSP 时序：CLKSTP= 11b，CLKXP= 1

5.32 闪存时序

表 5-60. 闪存耐受度⁽¹⁾

ADP32FXX Digital Signal Processor

	擦除/编程温度	最小值	典型值	最大值	单位
N _f 闪存对于阵列的耐受度 (写入/擦除周期)	0°C 至 85°C (环境温度)	30000	60000		周期
N _{OTP} OTP 对于阵列的耐受度 (写入周期)	0°C 至 85°C (环境温度)	1			写入

(1) 所示温度范围之外的写入/擦除操作并未说明, 有可能影响耐受数。

表 5-61. 150MHz SYSCLKOUT 上的闪存参数 : (1)

参数		最小值	典型值	最大值	单位
编程时间	16 位字				μs
		使用闪存 API v2.10	50		
	8K 扇区				ms
		使用闪存 API v2.10	250		
16K 扇区				ms	
	使用闪存 API v2.10	500			
擦除时间 ⁽²⁾	8K 扇区		10		S
	16K 扇区		11		
I _{DDP} 擦除/编程周期期间的 V _{DD} 功耗			105		mA
I _{DDIOP} 擦除/编程周期期间的 V _{DDIO} 功耗			5		mA

(1) 室温下包括函数调用开销在内的典型参数, 是在所有外设关闭时的参数。

(2) 当器件出货时, 片载闪存存储器处于一个被擦除状态。这样, 当首次编辑器件时, 在编程前无需擦除闪存存储器。然而, 对于所有随后的编程操作, 需要执行擦除操作。

表 5-62. 闪存/ OTP 访问时序

参数	最小值	最大值	单位
t _{a(fp)} 页式闪存访问时间	36		ns
t _{a(fr)} 随机闪存访问时间	36		ns
t _{a(OTP)} OTP 访问时间	60		ns

(1) 用于计算页等待状态和随机等待状态的公式 :

$$\text{FlashPage Wait State} = \left[\left(\frac{t_{a(fp)}}{t_{c(SCO)}} \right) - 1 \right] (\text{round up to the next highest integer, or 0, whichever is larger})$$

$$\text{FlashRandom Wait State} = \left[\left(\frac{t_{a(fr)}}{t_{c(SCO)}} \right) - 1 \right] (\text{round up to the next highest integer, or 1, whichever is larger})$$

$$\text{OTP Wait State} = \left[\left(\frac{t_{a(OTP)}}{t_{c(SCO)}} \right) - 1 \right] (\text{round up to the next highest integer, or 1, whichever is larger})$$

(2) 随机等待状态必须大于或者等于1。

5.33 ROM 时序

表 5-63. ROM 访问时序

参数	最小值	最大值	单位
t _{a(rp)} 页式 ROM 访问时间	23		ns
t _{a(rr)} 随机 ROM 访问时间	23		ns
t _{a(ROM)} ROM (OTP 区域) 访问时间(1)	60		ns

ADP32FXX Digital Signal Processor

(1) 在 ADP32FXX 器件中，一个 1K x 16 ROM 取代了闪存器件中的 OTP 块。

表 5-64. 不同频率上所要求的最小 ROM 等待状态 (ADP32FXX 器件)

SYCLKOUT (MHz)	SYCLKOUT (ns)	页等待状态 ⁽¹⁾	随机等待状态 ⁽¹⁾⁽²⁾	OTP
150	6.67	5	5	8
120	8.33	4	4	7
100	10	3	3	5
75	13.33	2	2	4
50	20	1	1	2
30	33.33	1	1	1
25	40	0	1	1
15	66.67	0	1	1
4	250	0	1	1

(1) 用于计算页等待状态和随机等待状态的公式：

$$\text{FlashPage Wait State} = \left\lceil \left(\frac{t_{a(fp)}}{t_{c(SCO)}} \right) - 1 \right\rceil \text{(round up to the next highest integer, or 0, whichever is larger)}$$

$$\text{ROMRandom Wait State} = \left\lceil \left(\frac{t_{a(fp)}}{t_{c(SCO)}} \right) - 1 \right\rceil \text{(round up to the next highest integer, or 1, whichever is larger)}$$

(2) 随机等待状态必须大于或者等于 1。

6 机械数据

表 6-1 到 6-3 提供了针对不同封装的热阻特性

表 6-1. 针对 179 焊球 BGA 封装的热阻特性

参数	179-BGA 封装	单位
P_{sijt}	0.659	°C/W
θ_{JA}	42.58	°C/W
θ_{JC}	16.09	°C/W

表 6-2. 针对 176 引脚 QFP 封装的热阻特性

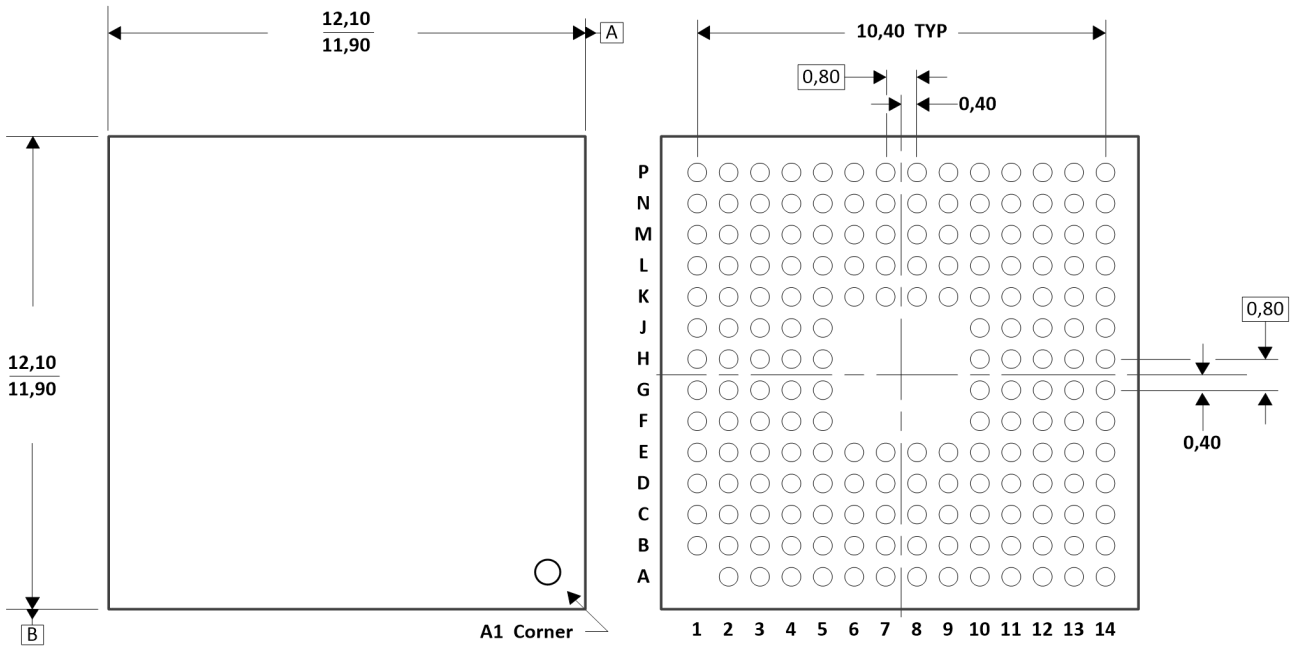
参数	176 -QFP 封装	单位
P_{sijt}	0.248	°C/W
θ_{JA}	41.91	°C/W
θ_{JC}	9.75	°C/W

表 6-3. 针对 128 引脚 QFP 封装的热阻特性

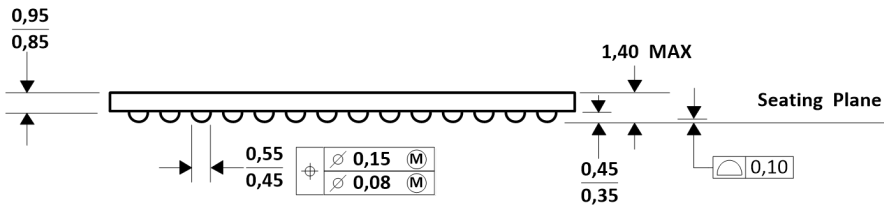
参数	128-QFP 封装	单位
P_{sijt}	0.275	°C/W
θ_{JA}	41.67	°C/W
θ_{JC}	10.81	°C/W

ADP32FXX Digital Signal Processor

BGA179



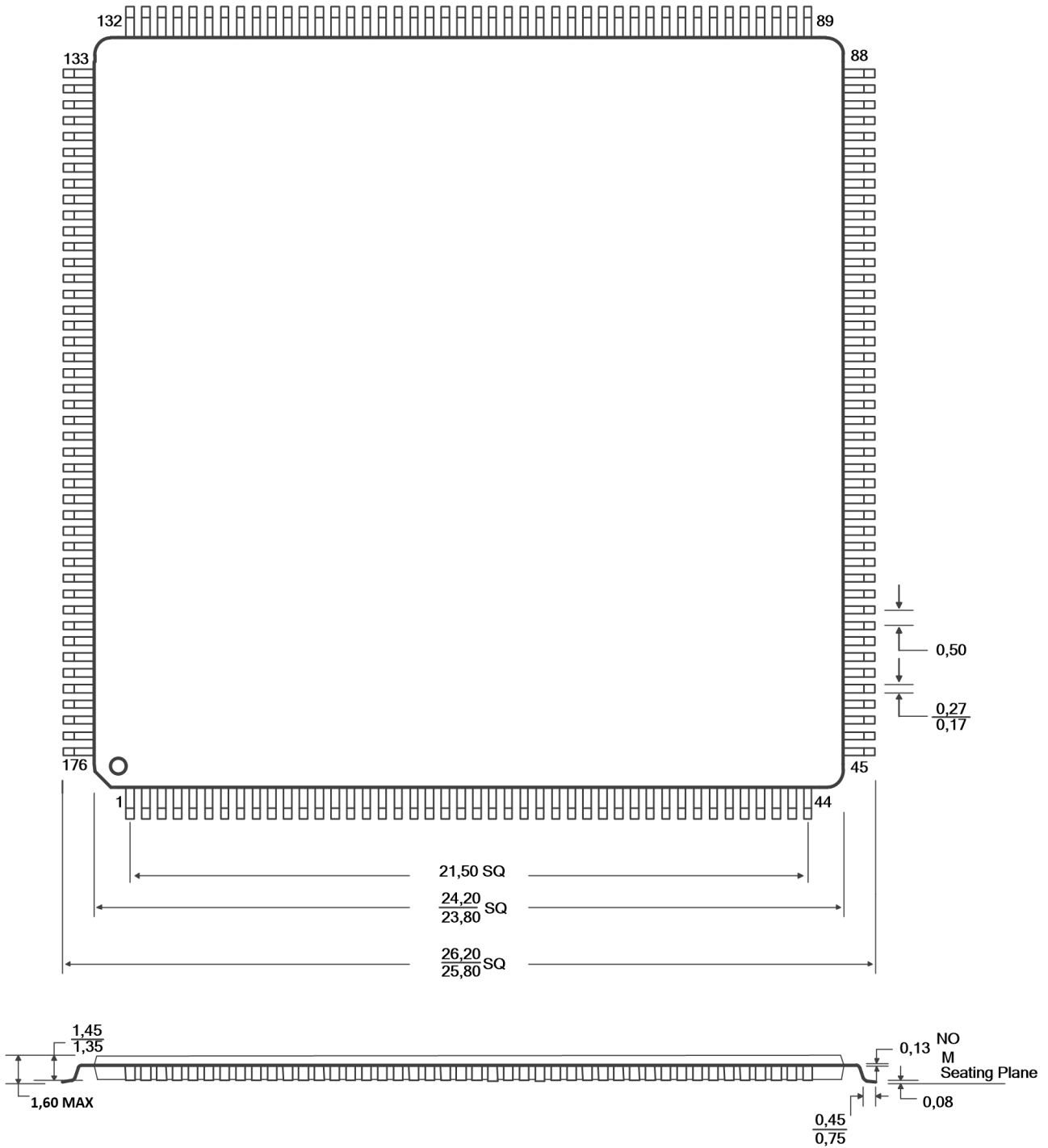
Bottom View



注释：单位 mm

ADP32FXX Digital Signal Processor

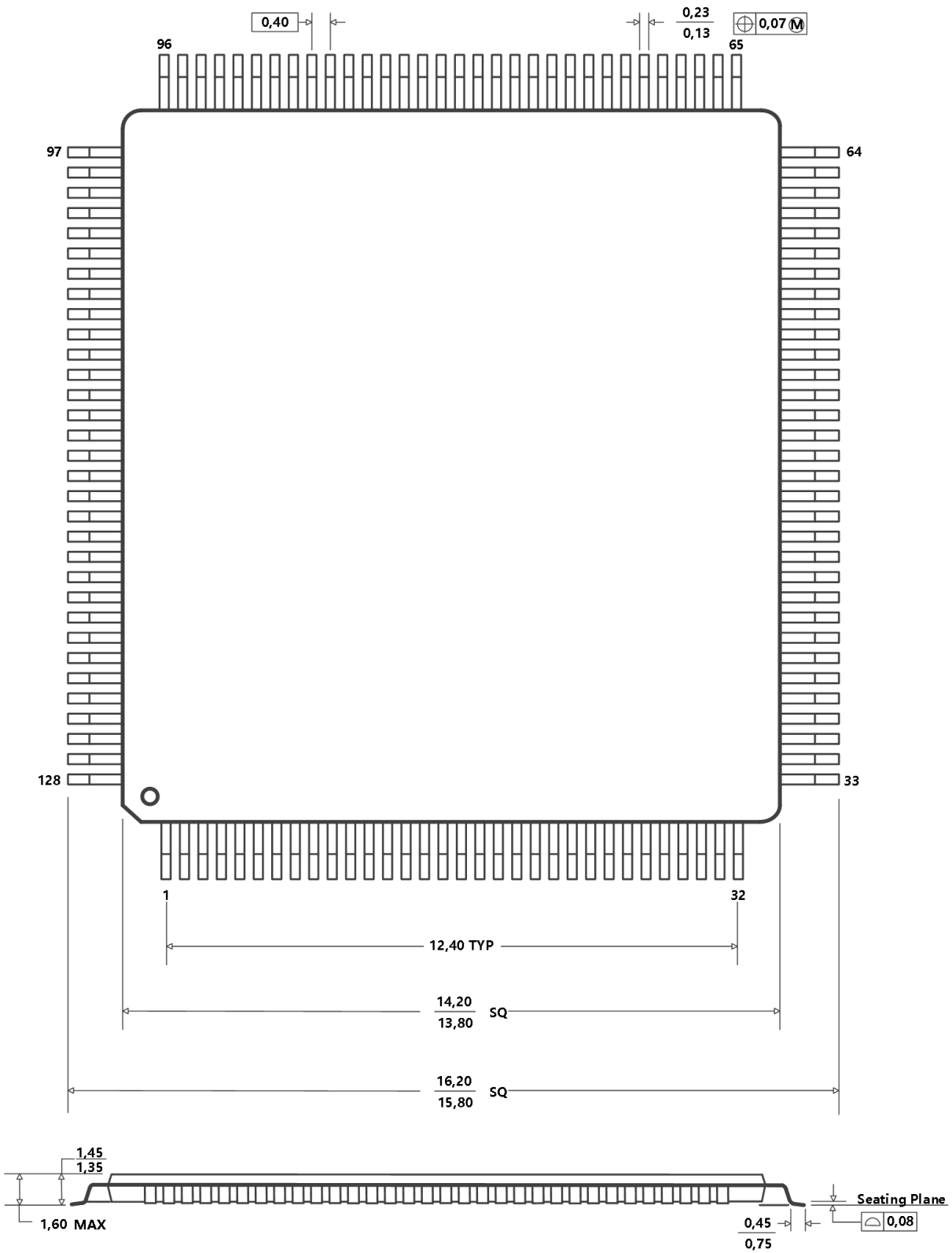
LQFP176



注释：单位 mm

ADP32FXX Digital Signal Processor

LQFP128



注释：单位 mm

联系方式

公司网址 : www.advancechip.com

联系邮箱:sales@advancechip.com

销售联系电话 : 0731-88731027

公司总部地址 : 长沙市高新开发区尖山路 39 号中电软件园总部大楼 10 楼