

ADP16F03 数字信号处理器

数据手册

编号: JXDZ7.381.015SJSC

Advancechip



Electronics

湖南进芯电子科技有限公司

2023年4月

V1.5

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	2021-08	周 乐	首次发布	吴顺锋	林钊漳
V1.1	2022-08-20	陈毅华	所有章节内容描述	吴顺锋	林钊漳
V1.2	2022-08-26	陈毅华	结构框图及部分描述	吴顺锋	林钊漳
V1.3	2022-10-27	何 龙	订正 PWM1-6 引脚排序为 PWM1、 3、5, PWM2、4、6	吴顺锋	林钊漳
V1.4	2023-03-02	何 龙	增加 LQFP32 封装及测试数据	吴顺锋	林钊漳
V1.5	2023-04-17	陈毅华	增加 QFN32 封装	吴顺锋	林钊漳

注：请关注更新版次！

目 次

1 产品特性	1
2 系统概述	2
3 引脚排布及说明	3
3.1 封装及引脚说明	3
3.2 产品编码	6
4 内存映射	7
4.1 存储空间映射	7
4.2 外设空间映射	8
5 内核说明	9
5.1 内核结构	9
5.2 系统寄存器	11
6 存储系统	15
6.1 内部存储器	15
6.2 双存取 RAM (DARAM)	15
6.3 单存取 RAM (SARAM)	15
6.4 闪存 FLASH	16
7 时钟系统	19
7.1 PLL 时钟模块	19
7.2 PLL 时钟模块控制寄存器	20
7.3 片内高精度 OSC	21
8 中断系统	21

8.1 中断系统结构.....	21
8.2 CPU 中断寄存器.....	25
8.3 外设中断控制寄存器.....	27
8.4 外部中断控制寄存器.....	32
9 看门狗.....	33
9.1 看门狗模块结构框图.....	33
9.2 看门狗控制寄存器.....	34
10 硬件加速单元.....	36
10.1 硬件加速单元功能特性.....	36
10.2 硬件加速单元各模块使用说明.....	37
11 数字 I/O 口以及复用引脚功能.....	49
11.1 数字 I/O 口以及复用引脚功能描述.....	49
11.2 数字 I / O 控制寄存器.....	50
12 串行通信接口 (SCI) 模块.....	58
12.1 串行通信接口 (SCI) 模块简介.....	58
12.2 串行通信接口 (SCI) 相关寄存器.....	60
13 串行外设接口 (SPI) 模块.....	66
13.1 串行外设接口 (SPI) 模块简介.....	66
13.2 串行外设接口 (SPI) 相关寄存器.....	68
14 事件管理器模块 (EM1)	72
14.1 事件管理器模块 (EM1) 简介.....	72
14.2 事件管理器模块 (EM1) 中断.....	74
14.3 事件管理器模块 (EM1) 通用定时器.....	76
14.4 事件管理器模块 (EM1) 全功能比较单元.....	81

14.5 事件管理器模块 (EM1) 脉冲宽度调制电路	84
14.6 事件管理器模块 (EM1) 捕获单元	85
14.6 正交编码脉冲(QEP) 电路	87
15 模数转换器 (ADC)	88
15.1 ADC 的特性	88
15.2 ADC 模块概述	89
15.3 自动排序器	91
15.4 ADC 时钟预定	103
15.5 ADC 校准	103
15.6 ADC 相关寄存器概述	103
16 运放与复用数字 IO 输入	113
16.1 运放与复用数字 IO 输入说明	113
16.2 运放与复用数字 IO 输入寄存器	115
17 电压比较器	119
17.1 电压比较器说明	119
18. 数字延时滤波	126
18.1 数字延时滤波模块结构	126
18.2 数字延时滤波模块寄存器说明	127
18.2.1 数字延时滤波控制寄存器	127
19. 供电	129
19.1 片上 LDO 电源管理	129
20.电气参数	132
20.1 建议的运行条件	132
20.2 建议运行温度范围内的电气特性	132

20.3 减少流耗.....	133
20.4 信号转换电平.....	134
20.5 时序参数符号.....	135
20.6 12 位模数转换器 (ADC)	135
20.6.1 建议运行环境.....	135
20.6.2 ADC 运行频率.....	136
20.7 在推荐运行条件范围下的 ADC 电气特性.....	136
21 机械数据.....	138
22 重要注意事项及声明.....	141

1 产品特性

- **供电电压**
 - 3.3~5V
 - (内核 1.2V, 模拟 3V, 由内部 LDO 产生)
- **高性能 16 位定点 DSP 内核**
 - 主频最高 100MHz
 - 哈佛(Harvard)总线结构
 - 快速中断响应和处理
- **片内存储器资源**
 - 4K x 16 位 SARAM
 - (256+256+32) x 16 位 DARAM
 - 32K x 16 位 Flash
- **运算加速单元**
 - 除法、开方
 - 反正切
 - 帕克变换
 - 32 位乘加及移位
- **30 个通用 I/O 引脚**
- **事件管理器 EM1**
 - 4 个 16 位定时/计数器
 - 8 路 PWM 输出
(PWM1~6 构成 3 组互补, PWM7/8 独立)
 - 1 组正交编码单元
 - 3 个捕获单元(CAP1/2/3)
 - 可配置 PWM 周期内中断产生和 ADC 采样触发时刻
- **ADC**
 - 12 位, 转换速率 1MSPS
 - 16 通道, 带温度传感器通道
 - 通道排序器, 可设置 ADC 采样通道序列
- **运算放大器**
 - 1 个 OP, 可用于母线电流检测放大
 - 3 个 PGA, 可用于相电流检测放大
- **电压比较器**
 - 3 个电压比较器, 参考端共用,
可用于 BEMF 比较检测
 - 2 个电压比较器, 带参考电压产生 DAC,
可用于保护
- **中断**
 - 2 个外部中断, 29 个由 PIE 设置的外设中断
- **数字延时滤波**
 - 3 组独立配置延时滤波单元,
分别对应外部中断、功率保护和 CAP1/2/3
- **串行通讯外设**
 - SPI、SCI
- **时钟**
 - 10M 片内振荡器
 - PLL 倍频系数 1x~10x
- **支持 WDT**
- **支持 4 线制快速程序烧录**
- **支持 JTAG 在线仿真**
 - 分析和断点功能
 - 基于硬件的实时调试
- **ESD 等级(HBM): 2000V**
- **MSL 湿敏等级: 3**
- **封装形式**
 - LQFP48 封装
 - LQFP32 封装
 - QFN32 封装

2 系统概述

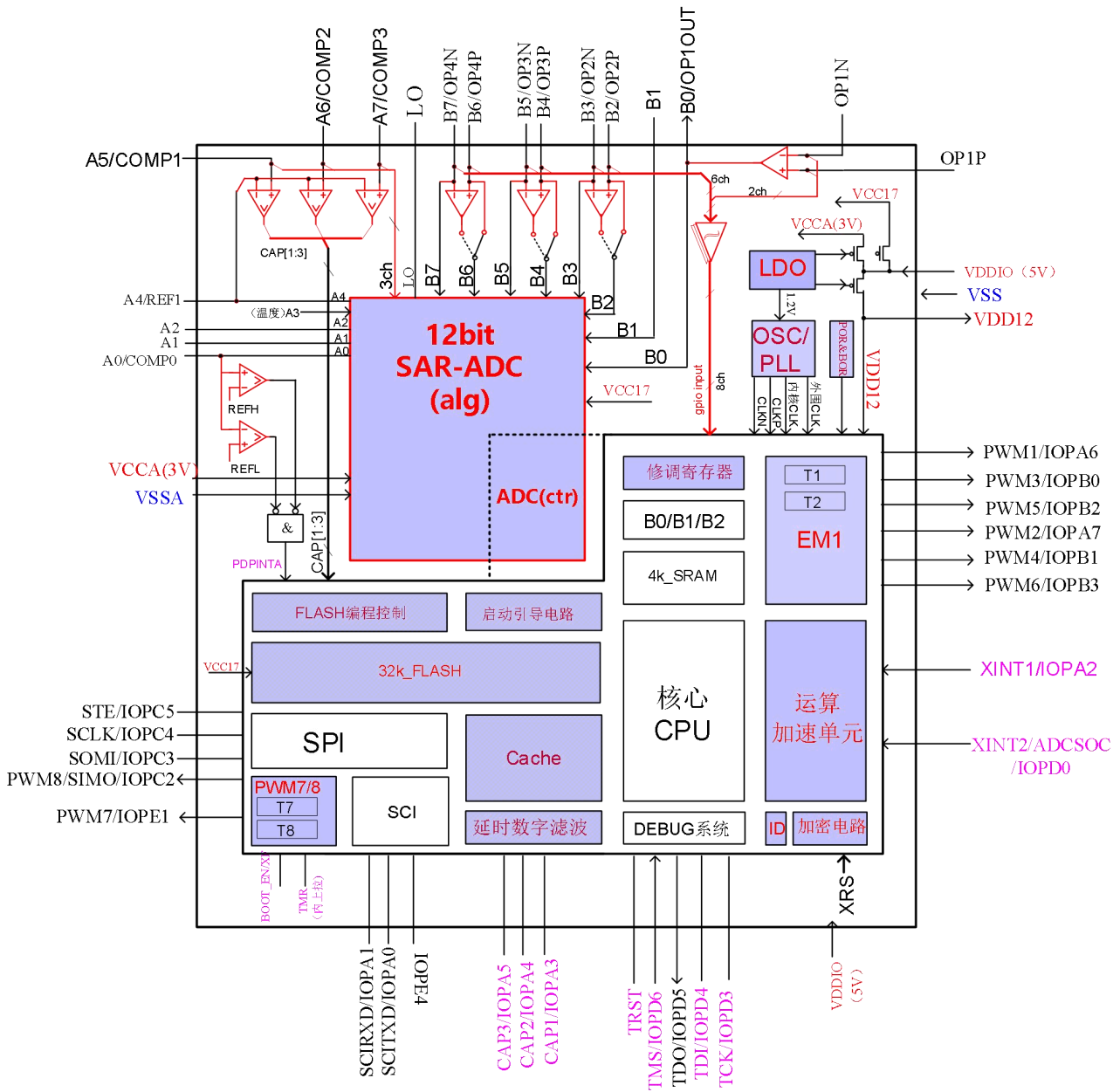


图 2-1 ADP16F03 芯片系统框图

ADP16F03 是面向无刷电机控制的 16 位 DSP 改进型，搭载 100M 主频 16 位 DSP 处理核+加速运算单元，事件管理器，SPI 和 SCI 模块，12 位 ADC，并集成运算放大器、电压比较器；可构成控制核心，支持有感、无感、方波、弦波等多模式无刷电机控制。

3 引脚排布及说明

3.1 封装及引脚说明

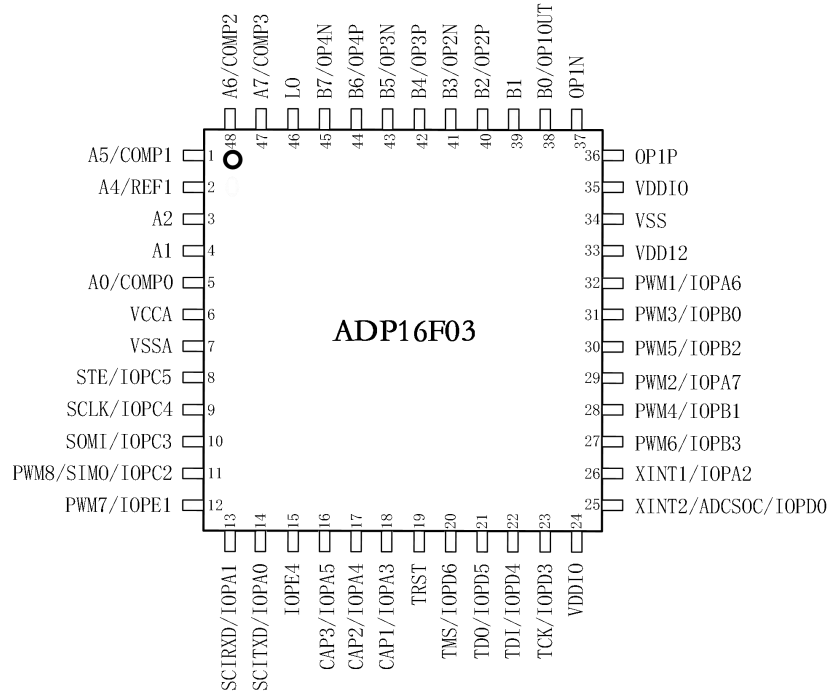


图 3-1 ADP16F03 芯片 QFP48 引排布

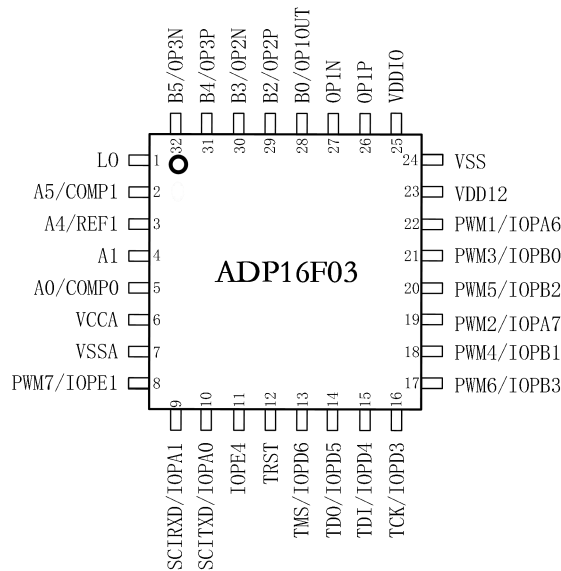


图 3-2 ADP16F03 芯片 QFP32 引排布

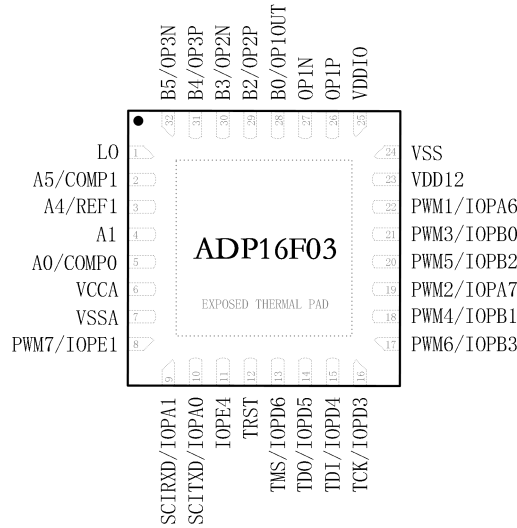


图 3-3 ADP16F03 芯片 QFN32 引排布

表 3-1 ADP16F03 芯片引脚功能定义

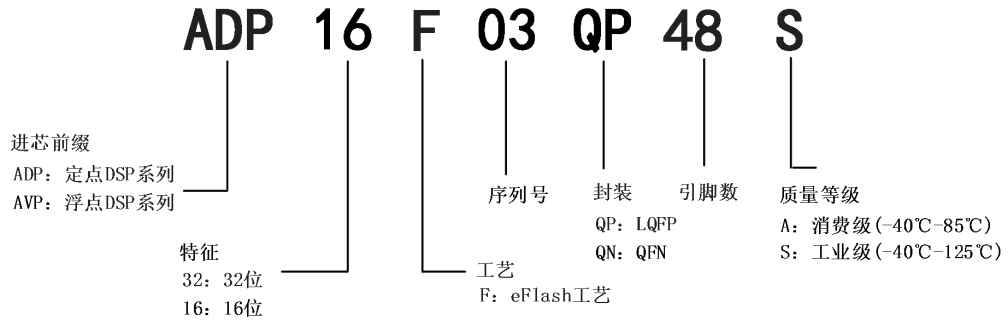
引脚名	序号			I/O/Z	PU/PD/Z	说明	
	QFP48	QFP32	QFN32				
A5/COMP1	1	2	2	I	-	ADC 输入 A5/比较器 1 输入端	
A4/ REF1	2	3	3	I	-	ADC 输入 A4/比较器 1、2、3 参考电压输入	
A2	3	-	-	I	-	ADC 输入 A2/运放 1 输入正端	
A1	4	4	4	I	-	ADC 输入 A1	
A0/COMP0	5	5	5	I	-	ADC 输入 A0/比较器 0 (双门限) 输入端	
VCCA	6	6	6	-	-	模拟电源 (3.0v, 由内部 LDO 产生)	
VSSA	7	7	7	-	-	模拟地	
STE/IOPC5	8	-	-	I/O/Z	PU	SPI SPI 从机发送使能/ GPIOC5	
SCLK/IOPC4	9	-	-	I/O/Z	PU		SPI 时钟/ GPIOC4
SOMI/IOPC3	10	-	-	I/O/Z	PU		SPI 从动输出、主控输入/ GPIOC3
PWM8/SIMO /IOPC2	11	-	-	I/O/Z	PU		PWM8 输出/SPI 从动输入、主控输出/ GPIOC2
PWM7/IOPE1	12	8	8	I/O/Z	PU	PWM7/GPIOE1	
SCIRXD/IOPA1	13	9	9	I/O/Z	PU	SCI SCI 接收/ GPIOA1	
SCITXD/IOPA0	14	10	10	I/O/Z	PU		SCI 发送/ GPIOA0
IOPE4	15	11	11	I/O/Z	PU	GP10E4	
CAP3/IOPA5	16	-	-	I/O/Z	PU	EM1 捕获 CAP3/GPIOA5	
CAP2/IOPA4	17	-	-	I/O/Z	PU		CAP2/GPIOA4
CAP1/IOPA3	18	-	-	I/O/Z	PU		CAP1/GPIOA3
TRST	19	12	12	I	PD	JTAG JTAG 复位	
TMS/IOPD6	20	13	13	I/O/Z	PU	JTAG 模式选择/GPIOD6	

引脚名	序号			I/O/Z	PU/PD/Z	说明	
	QFP48	QFP32	QFN32				
TDO/IOPD5	21	14	14	I/O/Z	PU	JTAG 串行输出/ GPIOD5	
TDI/IOPD4	22	15	15	I/OZ	PU	JTAG 串行输入/GPIOD4	
TCK/IOPD3	23	16	16	I/O/Z	PD	JTAG 时钟输入/GPIOD3	
VDDIO	24	-	-	-	-	3.3~5V 电源	
XINT2/ADCSOC /IOPD0	25	-	-	I/O/Z	PU	外部中断 2/ADC 外部触发/GPIOD0	
XINT1/IOPA2	26	-	-	I/O/Z	PU	外部中断 1/GPIOA2	
PWM6/IOPB3	27	17	17	I/O/Z	Z	EM1 PWM	PWM6/GPIOB3
PWM4/IOPB1	28	18	18	I/O/Z	Z		PWM4/GPIOB1
PWM2/IOPA7	29	19	19	I/O/Z	Z		PWM2/GPIOA7
PWM5/IOPB2	30	20	20	I/O/Z	Z		PWM5/GPIOB2
PWM3/IOPB0	31	21	21	I/O/Z	Z		PWM3/GPIOB0
PWM1/IOPA6	32	22	22	I/O/Z	Z		PWM1/GPIOA6
VDD12	33	23	23	-	-		1.2V 内核电源
VSS	34	24	24	-	-	数字地	
VDDIO	35	25	25	-	-	3.3~5V 电源	
OP1P	36	26	26	I	-	运放 1 输入正端/GPIOF4 输入	
OP1N	37	27	27	I	-	运放 1 输入负端/GPIOF3 输入	
B0/OP1OUT	38	28	28	I/O	-	ADC 输入 B0/运放 1 输出端	
B1	39	-	-	I	-	ADC 输入 B1	
B2/OP2P	40	29	29	I	-	ADC 输入 B2/运放 2 输入正端/GPIOF2 输入	
B3/OP2N	41	30	30	I	-	ADC 输入 B3 运放 2 输入负端/GPIOF1 输入	
B4/OP3P	42	31	31	I	-	ADC 输入 B4/运放 3 输入正端/GPIOF0 输入	
B5/OP3N	43	32	32	I	-	ADC 输入 B5/运放 3 输入负端/GPIOE7 输入	
B6/OP4P	44	-	-	I	-	ADC 输入 B6/运放 4 输入正端/GPIOE6 输入	
B7/OP4N	45	-	-	I	-	ADC 输入 B7/运放 4 输入负端/GPIOE5 输入	
LO	46	1	1	-	-	ADC 及运放参考地	
A7/COMP3	47	-	-	I	-	ADC 输入 A7/比较器 3 输入端	
A6/COMP2	48	-	-	I	-	ADC 输入 A6 /比较器 2 输入端	

- (1) GPIO 通用输入/输出引脚，所有 GPIO 在复位后默认为输入状态；
- (2) 建议模拟电源地与数字电源地隔离以保持指定精度；
- (3) PU 表示片内默认上拉，PD 表示片内默认下拉。

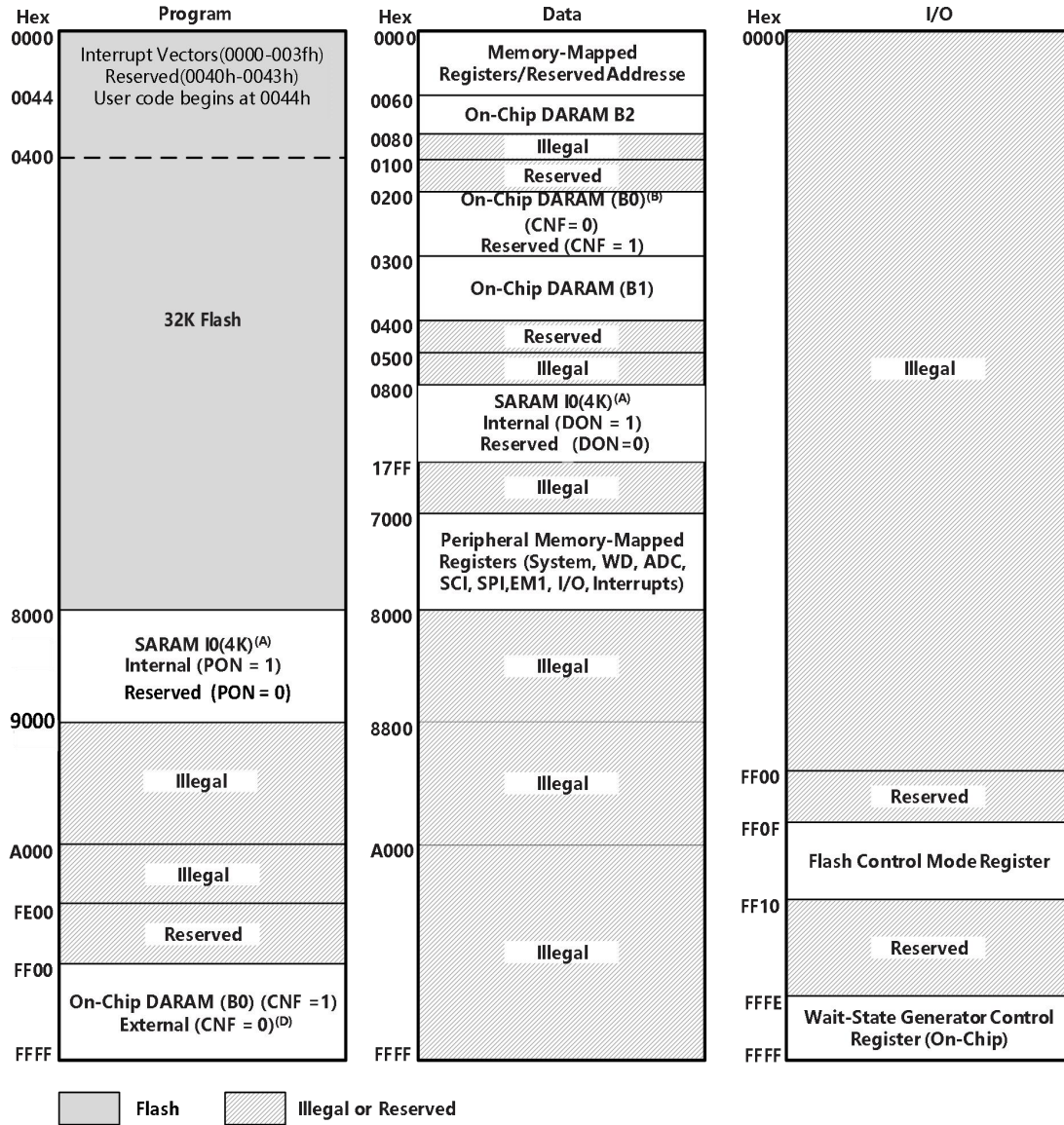
3.2 产品编码

ADP16F03 产品代号编码规则:



4 内存映射

4.1 存储空间映射



- A. 单周期访问 RAM I0, 配置 PON, DON 来选择程序或者数据空间。
- B. 双口 RAM B0, 通过配置 CNF 来选择程序或者数据空间。

图 4-1 ADP16F03 存储空间映射

4.2 外设空间映射

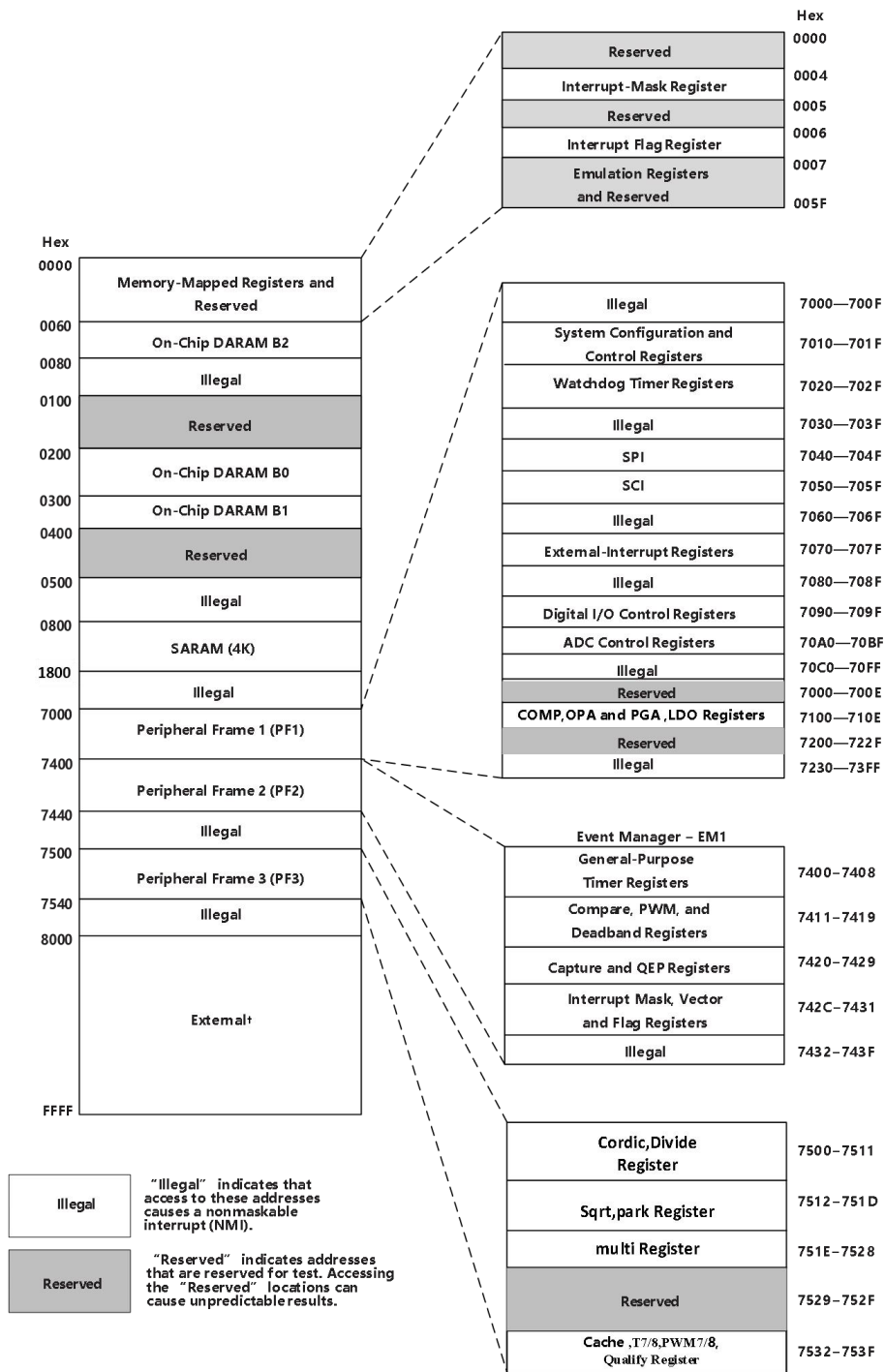


图 4-2 ADP16F03 外设空间映射

5 内核说明

5.1 内核结构

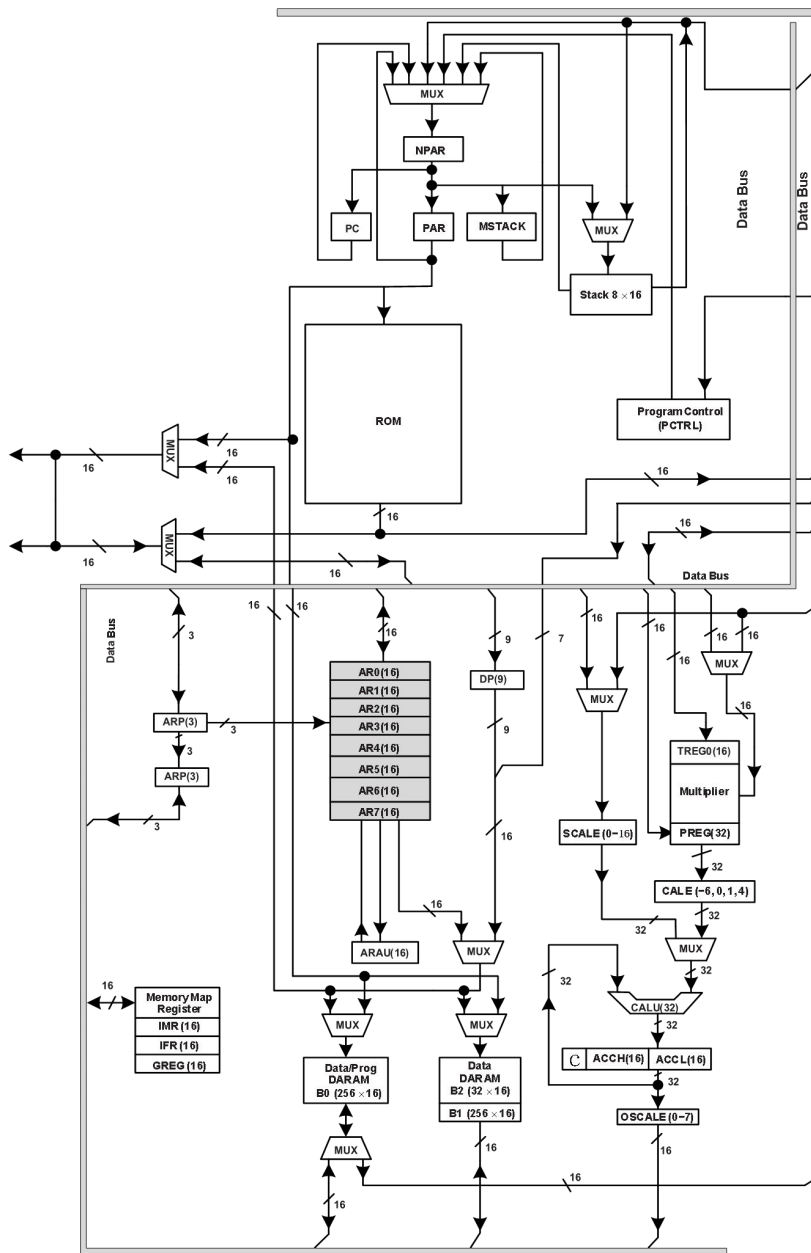


图 5-1 ADP16F03 内核结构

ADP16F03 内核结构简要说明：

中央处理单元 (CPU) 是 DSP 的核心模块，主要包括输入定标部分，乘法移位部分，中央算术逻辑部分，辅助寄存器算术单元和状态寄存器。

输入定标部分，乘法部分和中央算术逻辑部分是 CPU 的基本组成部分，是 DSP 运算处理

的核心。辅助寄存器算术单元主要对 8 个辅助寄存器进行运算。状态寄存器决定处理器的工作方式，寻址指针，工作状态和运算结果。图 5-1 为 CPU 的基本组成，表 5-1-1 是 AD16F03 的 CPU 内部硬件功能模块的作用和意义。

表 5-1-1 ADP16F03 DSP CPU 内部硬件说明

符号	名称	描述
ACC	累加器	32 位寄存器，保存 CALU 计算结果，并为下一次 CALU 操作提供输入，并具有移位和循环操作功能。
ARAU	辅助寄存器算术单元	无符号的 16 位运算单元，间接寻址时使用辅助寄存器来计算间接地址。
AUX REGS	辅助寄存器 0~7	这些 16 位寄存器可以作为地址指针指向存储器任意地址。它们由 ARAU 操作并由辅助寄存器指针 (ARP) 选定。AR0 可用作更新 ARx (x 为 1~7) 参考值，也可作为 ARx 的比较值。
C	进位标志位	从 CALU 寄存器输出。C 被反馈到 CALU 进行扩展算术运算。C 标志位位于状态寄存器 1 (ST1)，其状态可通过条件指令测试。C 也可以用于累加器移位和循环。
CALU	中央算术逻辑单元	CPU 中 32 位主算术逻辑单元，CALU 在单个机器周期中执行 32 位操作。CALU 使用来自 ISCALE 或 PSCALE 的数据以及来自 ACC 的数据操作，并将运算后结果保存在 PCTRL 单元中。
DARAM	双口 RAM	如果芯片上的 RAM 配置控制位 (CNF) 被设置为 0，B0 (DARAM) 被映射到数据空间；否则，B0 被映射到程序空间。块 B1 和 B2 被映射到地址为 0300h~03FFh 和 0060h~007Fh 的数据存储器空间。B0 和 B1 容量均为 256 个字，B2 容量为 32 个字。
DP	数据存储器页指针	9 位 DP 寄存器与指令字的 7 个最低有效位 (LSBs) 组合形成 16 位直接存储器地址。DP 可以由 LST 和 LDP 指令修改。
GREG	全局存储器配置寄存器	GREG 指定全局数据存储空间的大小。由于在 ADP16F03 器件中没有使用全局内存空间，所以这个寄存器是保留的。
IMR	中断屏蔽寄存器	CPU 级中断使能屏蔽寄存器。IMR 寄存器各位分别用来屏蔽或使能对应的中断。
IFR	中断标志寄存器	CPU 级中断标志寄存器。
INT #	中断陷阱	总共 32 个可通过硬件或软件方式产生的中断。
ISCALE	输入数据定标移位器	16 到 32 位的桶形左移位器。ISCALE 将输入的 16 位数据的 0 到 16 位在捕获周期内向左移位，以得到 32 位输出，因此输入定标移位操作不需要额外周期。
MPY	乘法器	16x16 位乘法器输出 32 位乘积。MPY 在单周期执行乘法。MPY 可进行有符号或者无符号的二进制补码乘法运算
MSTACK	微堆栈	当程序地址生成逻辑被用作在数据空间生成连续地址时，MSTACK 为一条指令的地址提供暂存
MUX	多路选择器	从多路总线选择一个通用输入
NPAR	下一个程序地址寄存器	NPAR 保存下一个指令周期出现在程序地址总线上的程序地址
OSCALE	输出数据定标寄存器	16-32 位的桶形左移寄存器。OSCALE 将 32 位的累加器输出左移 0 到 7 位，用以实现数据归一化，并将移位的 32 位数据高 16 位或者低 16 位输出到数据写总线 (DWEB)。
PAR	程序地址寄存器	PAR 保存当前在 PAB 上运行地址，用于完成当前总线周期的所有内存操作所需的

		多个循环。
PC	程序计数器	PC 计数器每次将 NPAR 的值增加 1，以便为下一次的取指操作和数据传送操作提供地址
PCTRL	程序控制器	PCTRL 解码指令，管理流水线，存储状态位，并对条件操作进行解码。
PREG	乘积寄存器	32 位寄存器保存 16x16 的乘积
PSCALE	乘积定标移位器	将乘法器乘积 0 位 1 位,4 位左移，或 6 位右移。左移可得到由二进制补码乘法运算产生的附加标志位。右移可用来将数字量按比例减少，以防止 CALU 内的乘积累加溢出。
STACK	堆栈	堆栈是用于存储子程序和中断服务例程的返回地址的内存块，或者用于存储数据。ADP16x 堆栈是 16 位位宽，8 级深度。
TREG	临时寄存器	用来保存一个乘法运算操作数中的 16 位寄存器。TREG 可以保存 LACT、ADDT 和 SUBT 指令的移位个数，TREG 也可保存 BITT 指令的测试位的位置

5.2 系统寄存器

5.2.1 状态寄存器

两个状态寄存器 ST0 和 ST1 包含各种条件和模式的状态。这些寄存器可以存储在数据存储器中，并从数据存储器加载，从而允许保存微处理器状态并恢复子程序。

加载状态寄存器 (LST) 指令用于写入 ST0 和 ST1。 存储状态寄存器 (SST) 指令用于从 ST0 和 ST1 读取 - 除了不受 LST 指令影响的 INTM 位之外。 使用 SETC 和 CLRC 指令时，可以将这些寄存器的各个位置 1 或清 0。 下面显示了状态寄存器 ST0 和 ST1 的组织，指示了每个状态寄存器中包含的所有状态位。 读状态寄存器中保留位为逻辑 1。表 5-2-1 列出了状态寄存器字段定义。

1. ST0 状态寄存器

15~13	12	11	10	9	8~0
ARP	OV	OVM	1	INTM	DP
RW_X	RW_X	RW_X	RW_1	RW_X	RW_X

注：R=可读，W=可写，C=清除，_0=复位值；

位 15~13	ARP: 辅助寄存器缓冲区的指针。当 ARP 被加载到 ST0 时, 旧的 ARP 值被复制到 ARB, 除非在一个指令的指令中。当 ARB 通过一个 LST #1 指令加载时, 同样的值也被复制到 ARP 中。
位 12	OV: 溢出标志位。当将 OVM 设置为 0 时, 溢出的结果通常在累加器中溢出。当设置为 1 时, 当遇到溢出时, 累加器将被设置为最正值或负值。SETC 和 CLRC 指令分别设置并重新设置这个位。还可以用来修改 OVM。
位 11	OVM: 值。SETC 和 CLRC 指令设置和重置此位。LST 也可以用来修改 OVM。
位 9	INTM: 中断模式。当 INTM 被设置为 0 时, 所有的非屏蔽中断都是启用的。当设置为 1 时, 所有的屏蔽中断都是禁用的。INTM 由 SETC INTM 和 CLRC INTM 指令设置并重新设置。RS 也能设置 INTM。INTM 对不可屏蔽的 RS 和 NMI 中断没有影响。注意, INTM 不受该指令的影响。这个位被重置为 1。当一个可屏蔽的中断陷阱被捕获时, 它也被设置为 1。
位 8~位 0	DP: 数据存储页指针。9 位 DP 寄存器与指令字的 7 个 LSB 指令组合以形成 16 个比特的直接存储器地址。DP 可以由 LST 和 LDP 指令修改。

2. ST1 状态寄存器

15~13	12	11	10	9	8	7	6	5	4	3	2	1~0
ARB	CNF	TC	SXM	C	1	1	1	1	1	1	1	PM
RW_X	RW_0	RW_X	RW_1	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_00

注：R=可读，W=可写，C=清除，_0=复位值；

位 15~位 13	ARB: 辅助寄存器缓冲区的指针。当 ARP 被加载到 ST0 时, 旧的 ARP 值被复制到 ARB, 除非在一个指令的指令中。当 ARB 通过一个 LST #1 指令加载时, 同样的值也被复制到 ARP 中。
位 12	CNF: 片内的 RAM 配置控制位。如果将 CNF 设置为 0, 则可重新配置的数据双访问 RAM 块被映射到数据空间;否则, 它们会被映射到程序空间。CNF 可以通过 SETC CNF、CLRC CNF 和 1/1 指令进行修改。RS 将 CNF 设为 0。
位 11	TC: 测试/控标志位。TC 受 BIT, BITT, CMPR, LST #1 和 NORM 指令的影响。
位 10	SXM: 符号扩展模式。SXM=1 在数据通过缩放器传递到累加器时, 在数据上产生符号扩展。SXM=0 禁止符号扩展。SXM 不影响某些指令的定义;例如 ADDS 指令, 不管 SXM 如何, 都会禁止符号扩展。SXM 由 SETC SXM 指令置 1, 并由 CLRC SXM 指令复位, 并且可以通过 LST #1 指令加载。SXM 通过复位设置为 1。
位 9	C: 进位。如果加法结果产生一个进位, 则 C 被置为 1, 如果减法的结果产生一个借位, 则 C 被复位位 0, 否则在加法或减法之后被复位, 除非指令是 ADD 或带有 16 位移位的 SUB, 在这些情况下,

	ADD 只能设置, SUB 只能重置进位, 但不能影响它, 单 bit 移位或者旋转指令也会影响 C 以及 SETC, CLRC 和 LST #1 指令, 在复位时 C 被置位为 0。
位 1~位 0	<p>PM: 乘积移位方式</p> <p>PM=00, 乘法器的 32 位乘积结果不移位, 直接载入 CALU</p> <p>PM=01, PREG 输出左移 1 位后载入 CALU, 最低位 (LSB) 以 0 填充</p> <p>PM=10, PREG 输出左移 4 位后载入 CALU, 最低位 (LSB) 以 0 填充</p> <p>PM=11, PREG 输出进行符号扩展右移 6 位</p> <p>注意: PREG 中的内容一直保持不变, 当把 PREG 中的内容传送到 CALU 单元时, 将发生移位操作, PM 可由 SPM 指令和 LST #1 指令加载, 复位时 PM 清 0</p>

5.2.2 系统控制寄存器

1. 系统控制寄存器 1 (SCSR1) --地址 7018h

15	14	13	12	11	10	9	8
W:pll_bypass_en R: pll_lock_flg	保留位	保留位	保留位	CLK_NS3	CLK_NS2	CLK_NS1	CLK_NS0
R_0	RW_0	RW_0	RW_0	RW_1	RW_1	RW_1	RW_0
7	6	5	4	3	2	1	0
ADCCLK_EN	SCICLK_EN	SPICLK_EN	Auxiliary CLKEN 0	Auxiliary CLKEN 1	EM1CLK_EN	保留位	ILLADR
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	R_0	RC_0

注: R=可读, W=可写, C=清除, _0=复位值;

位 15	PLL 时钟控制增加了 PLL 时钟在倍频锁定之前的 bypass 的功能, 以保证内核稳定工作, 其中 SCSR1[15]在读写是时分别具有不同的含义, 写入时是开启 PLL 倍频锁定之前 bypass 控制的使能位, 高电平有效, 默认开启, 若是低电平则不开启 PLL 倍频锁定之前 bypass 的功能, 即 PLL 输出直接送入内核。读 SCSR1[15]的含义是判断 PLL 是否锁定, 若读出来的 SCSR1[15]是高电平, 则表示 PLL 已锁定, 若是低电平则表示 PLL 还未锁定。
位 14	保留
位 13~12	保留
位 11~8	CLK_NS3~ CLK_NS0: PLL 倍频系数选择; (详情见系统时钟)
位 7	ADCCLK_EN: ADC 模块时钟使能控制位; 0: 禁止 ADC 模块时钟 (即: 关断 ADC 模块, 以降低功耗); 1: 使能 ADC 模块时钟, 且正常运行;
位 6	SCICLK_EN: SCI 模块时钟使能控制位; 0: 禁止 SCI 模块时钟 (即: 关断 SCI 模块, 以降低功耗); 1: 使能 SCI 模块时钟, 且正常运行;
位 5	SPICLK_EN: SPI 模块时钟使能控制位;

	0: 禁止 SPI 模块时钟 (即: 关断 SPI 模块, 以降低功耗); 1: 使能 SPI 模块时钟, 且正常运行;
位 4	Auxiliary CLKEN 0: 辅助寄存器 0 时钟使能控制位, 辅助寄存器 0 包括运放部分的寄存器与电压比较器部分的寄存器; 0: 禁止辅助寄存器 0 时钟; 1: 使能辅助寄存器 0 时钟;
位 3	Auxiliary CLKEN 1: 辅助寄存器 1 时钟使能控制位, 辅助寄存器 1 包括 T7,T8,PWM7/PWM8, cach 部分的寄存器和数字延时滤波部分的寄存器, 加速运算单元的寄存器; 0: 禁止辅助寄存器 1 时钟; 1: 使能辅助寄存器 1 时钟;
位 2	EM1CLK_EN: EM1 模块时钟使能控制位; 0: 禁止 EM1 模块时钟 (即: 关断 EM1 模块, 以降低功耗); 1: 使能 EM1 模块时钟, 且正常运行;
位 1	保留
位 0	ILLDR: 无效地址检测位;

2. 系统寄控制寄存器 (SCSR2) -- 地址 7019h

15	14	13	12	11	10	9	8
保留							
RW_0							
7	6	5	4	3	2	1	0
保留位	保留位	WDOVERRIDE	保留	保留位	保留	DON	PON
RW_0	RC_1	RW_0	RW_0	RW-BOOTEN	RC_1	RW_1	RW_1

注: R=可读, W=可写, C=清除, _0=复位值, S=只可被置位;

位 15~位 7	保留
位 6	保留
位 5	WD OVERRIDE: WD 保护位, 复位后的默认值为 1, 从而使用户可以通过用软件将 WDCR 寄存器中的 WDDIS 位置 1, 来禁止 WD 工作, 该位是一个只能清 0 位, 通过向该位写 1 来对其清 0; 0: 使用户不能通过软件来禁止 WD. 该位是一个只能清 0 位, 不能通过软件来置 1; 1: 复位时默认值, 使用户能够通过软件来禁止 WD 工作;
位 4	保留
位 3	保留
位 2	保留
位 1~位 0	SRAM 程序/数据空间选择位。 DON PON SARAM 状态; 0 0 地址空间不被映射; 0 1 SARAM 被映射到片内程序空间; 1 0 SARAM 被映射到片内数据空间; 1 1 SARAM 被映射到片内程序空间, 又被映射到片内数据空间;

6 存储系统

6.1 内部存储器

ADP16F03 器件配置如下内存模块：

- 4K x 16 位 SARAM
- (256+256+32) x 16 位 DARAM
- 32K x 16 位 Flash

6.2 双存取 RAM (DARAM)

在 ADP16F03 设备上有 544 字×16 位的 DARAM。DARAM 允许在同一个周期内对 RAM 进行写入和读取。DARAM 配置为三个块：块 0 (B0)，块 1 (B1) 和块 2 (B2)。块 1 包含 256 个字，块 2 包含 32 个字，并且这两个块仅位于数据存储空间中。块 0 包含 256 个字，可以配置为数据或程序存储器空间。SETC CNF (将 B0 配置为程序存储器) 和 CLRC CNF (将 B0 配置为数据存储器) 指令允许通过软件动态配置存储器映射。

在使用片上 RAM 时，ADP16F03 全速运行，无需等待状态。DARAM 允许在一个周期内执行两次访问，再加上 ADP16F03 架构的并行特性，使得设备可以在任何给定机器周期内执行三次并发的内存访问。

6.3 单存取 RAM (SARAM)

ADP16F03 器件中有 4K 字 x16 位 I0 SARAM,可以通过 SCSR2 寄存器 PON 和 DON 位配置为数据存储器或程序存储器或程序空间与数据空间共用，有关 SCSR2 寄存器的 PON 和 DON 位详细信息，请参见 SCSR2 寄存器描述表。复位时，PON 和 DON 为 11，片内 SARAM 映射到程序和数据空间。

6.4 闪存 FLASH

6.4.1 FLASH 说明

ADP16F03 提供闪存 Flash。ADP16F03 程序空间中集成了一个 32K x 16 位的 Flash，Flash 映射在程序空间 0000h-7FFFh,其中中断向量表空间 0000h-003Fh,保留空间 0040-0043h 为代码安全密码。

将 flash 地址分为四段,sector0(4k)地址为 0000-0FFF,sector1(12k)地址为 1000-3FFF,sector2 (12k) 地址为 4000-6FFF, sector3 (4k) 地址为 7000-7FFF。

6.4.2 唯一 ID

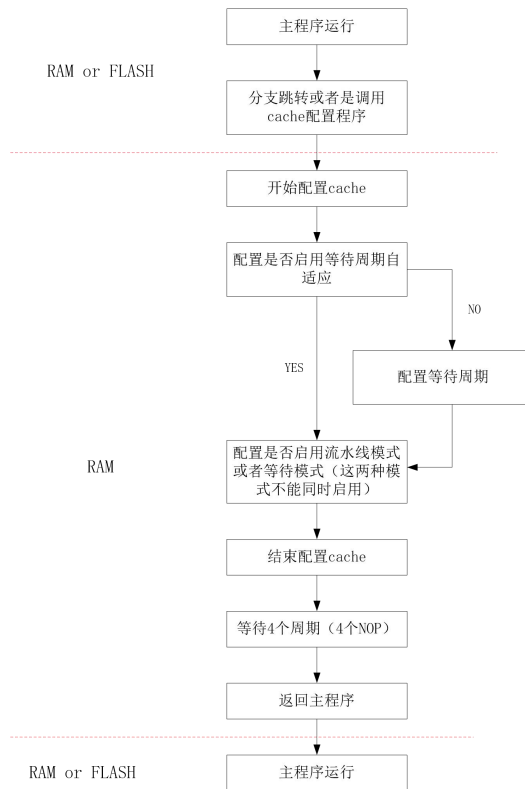
新增唯一 ID 号, 在 flash NVR_USR1 区, 写入地址为 16' h83, 16' h82, 16' h81, 16' h80。在上电完成后会将唯一 ID 号加载到寄存器 0X752D, 0X752C, 0X752B, 0X752A。该寄存器为只读寄存器。

6.4.3 cache 说明

1. cache 功能描述

- (1) 工作模式可配置为流水线模式或等待模式
- (2) 等待时钟周期支持用户配置或者根据系统时钟频率自动调节
- (3) 缓存容量: 64bit
- (4) cache 功能可关闭
- (5) 等待模式时, 每当高位地址 (高位地址为最高位到第 2 位) 变化时, 等待相应的时钟周期数 (WaitNum) ; 流水线模式时, 每当高位地址变化不连续时, 等待相应时钟周期数 (WaitNum) 。

2. 配置方法



- A. cache 的配置程序不能在 flash 中运行;
- B. 配置过程必须按照图 6-1;
- C. 当系统时钟周期大于等于 45ns 时, 不建议使用 cache;
- D. 等待周期配置方法:

$$\text{WaitNum} = (45\text{ns}) / \text{系统时钟周期}; \text{去尾法取整};$$
- E. Cache 模块只在 PLL 启动倍频锁定之前的 bypass 功能时 (pll_lock_flag=1) 可用;

图 6-1 cache 配置流程图

3. cache 控制寄存器—地址 753Eh

15	14	13	12	11	10	9	8
PWM8_EN	PWM7_ENT	PWM8_OUT_PAD	保留位	保留位	保留位	PipeEn	WaitModeEn
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
保留位	保留位	保留位	保留位	AutoF	WaitNum		
RW_0	R_1	R_1	RW_0	RW_0	RW_0	RW_0	RW_0

注: R=可读, W=可写, C=清除, _0=复位值, _1=复位值;

位 9	PipeEn: 流水线模式使能位: 1: 启用该模式; 0: 关闭该模式;
位 8	WaitModeEn: 等待模式使能位: 1: 启用该模式; 0: 关闭该模式;
位 7~位 4	保留
位 3	AutoF: 频率自适应使能位: 1: 等待周期根据频率自动调整; 0: 等待周期为 WaitNum 中的配置值;
位 2~位 0	WaitNum: 等待周期控制位, 读取 flash 中的数据需要等待的周期。 当 AutoF=1 时, 此位不用配置。 当 AutoF=0 时: $WaitNum = (45ns / \text{系统时钟周期})$; 去尾法取整;

- (1) cache 控制寄存器属于辅助寄存器 1, 使用时需要使能辅助寄存器 1 的时钟;
- (2) 流水线模式和等待模式同时启用时, 等待模式优先级更高;

7 时钟系统

7.1 PLL 时钟模块

ADP16F03 有一个片载、基于 PLL 的时钟模块。这个模块为设备提供所有需要的时钟信号,以及控制设备进入低功耗模式。PLL 通过 4 个比特位 SCSR1 寄存器中的[11:8]位, CLK_NS3, CLK_NS2, CLK_NS1, CLK_NS0 来控制 and 选择不同的 CPU 时钟频率。

PLL 时钟控制增加了 PLL 时钟在倍频锁定之前的 bypass 的功能, 以保证内核稳定工作, 其中 SCSR1[15]在读写时分别具有不同的含义, 写入时是开启 PLL 倍频锁定之前 bypass 控制的使能位, 高电平有效, 默认开启, 若是低电平则不开启 PLL 倍频锁定之前 bypass 的功能, 即 PLL 输出直接送入内核。读 SCSR1[15]的含义是判断 PLL 是否锁定, 若读出来的 SCSR1[15]是高电平, 则表示 PLL 已锁定, 若是低电平则表示 PLL 还未锁定。

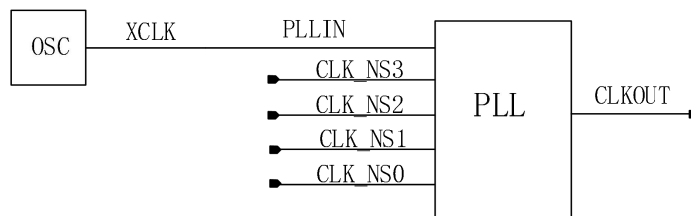


图 7-1 PLL 时钟模块框图

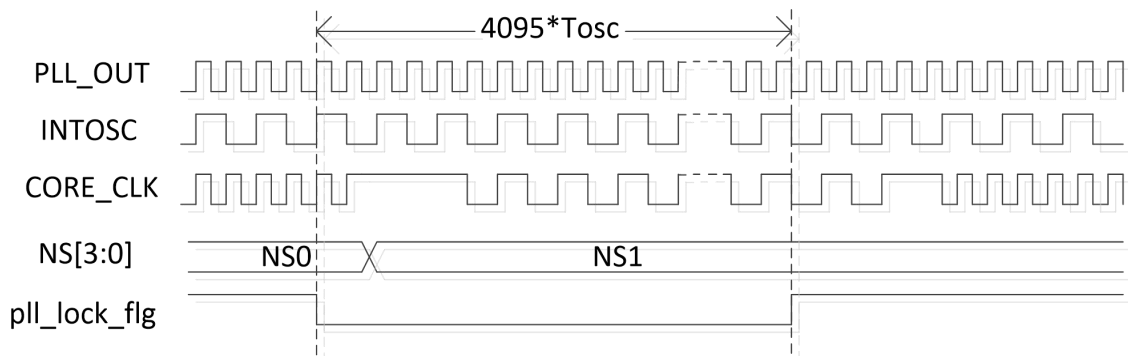


图 7-2 PLL 时钟 bypass 控制部分信号时序图

表 7-1 通过 SCSR1 (0x7018) 寄存器选择 PLL 时钟

CLK NS[3:0]	PLLOUT	CLK NS[3:0]	PLLOUT
0000	PLLIN *1	0001	PLLIN *1
0010	PLLIN *2	0011	PLLIN *3
0100	PLLIN *4	0101	PLLIN *5
0110	PLLIN *6	0111	PLLIN *7
1000	PLLIN *8	1001	PLLIN *9
1010	PLLIN *10	1011	PLLIN *11
1100	PLLIN *12	1101	PLLIN *13
1110	PLLIN *14	1111	PLLIN *15

7.2 PLL 时钟模块控制寄存器

1. 系统控制寄存器 1 (SCSR1) --地址 7018h

15	14	13	12	11	10	9	8
W:pll_bypass_en R: pll_lock_flg	保留位	保留位	保留位	CLK_NS3	CLK_NS2	CLK_NS1	CLK_NS0
R_0	RW_0	RW_0	RW_0	RW_1	RW_1	RW_1	RW_0
7	6	5	4	3	2	1	0
ADCCLK_EN	SCICLK_EN	SPICLK_EN	Auxiliary CLKEN 0	Auxiliary CLKEN 1	EM1CLK_EN	保留位	ILLADR
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	R_0	RC_0

注：R=可读，W=可写，C=清除，_0=复位值；

位 15	PLL 时钟控制增加了 PLL 时钟在倍频锁定之前的 bypass 的功能，以保证内核稳定工作，其中 SCSR1[15]在读写时分别具有不同的含义，写入时是开启 PLL 倍频锁定之前 bypass 控制的使能位，高电平有效，默认开启，若是低电平则不开启 PLL 倍频锁定之前 bypass 的功能，即 PLL 输出直接送入内核。读 SCSR1[15]的含义是判断 PLL 是否锁定，若读出来的 SCSR1[15]是高电平，则表示 PLL 已锁定，若是低电平则表示 PLL 还未锁定。
位 14	保留
位 13~12	保留
位 11~8	CLK_NS3~ CLK_NS0: PLL 倍频系数选择； (详情见系统时钟)
位 7	ADCCLK_EN: ADC 模块时钟使能控制位； 0: 禁止 ADC 模块时钟 (即: 关断 ADC 模块, 以降低功耗); 1: 使能 ADC 模块时钟, 且正常运行;
位 6	SCICLK_EN: SCI 模块时钟使能控制位； 0: 禁止 SCI 模块时钟 (即: 关断 SCI 模块, 以降低功耗);

	1: 使能 SCI 模块时钟, 且正常运行;
位 5	SPICLK_EN: SPI 模块时钟使能控制位; 0: 禁止 SPI 模块时钟 (即: 关断 SPI 模块, 以降低功耗); 1: 使能 SPI 模块时钟, 且正常运行;
位 4	Auxiliary CLKEN 0: 辅助寄存器 0 时钟使能控制位, 辅助寄存器 0 包括运放部分的寄存器与电压比较器部分的寄存器; 0: 禁止辅助寄存器 0 时钟; 1: 使能辅助寄存器 0 时钟;
位 3	Auxiliary CLKEN 1: 辅助寄存器 1 时钟使能控制位, 辅助寄存器 1 包括 T7,T8,PWM7/PWM8, cach 部分的寄存器和数字延时滤波部分的寄存器, 加速运算单元的寄存器; 0: 禁止辅助寄存器 1 时钟; 1: 使能辅助寄存器 1 时钟;
位 2	EM1CLK_EN: EM1 模块时钟使能控制位; 0: 禁止 EM1 模块时钟 (即: 关断 EM1 模块, 以降低功耗); 1: 使能 EM1 模块时钟, 且正常运行;
位 1	保留
位 0	ILLDR: 无效地址检测位;

7.3 片内高精度 OSC

ADP16F03 片内集成高精度无引脚 OSC 模块, 为 PLL 模块提供时钟源, 无需片外晶振。

典型输出频率为 10MHz, OSC 的指标参数如表 7-2 所示。

表 7-2 OSC 指标参数

指标参数	最小值	典型值	最大值	单位
OSC 在-40°C~125°C时频率	9.800	10.000	10.200	MHz

8 中断系统

8.1 中断系统结构

ADP16F03 软件可编程中断结构灵活, 支持片内和外部中断配置以满足实时中断驱动的应用程序需求。ADP16F03 可以识别三种类型的中断源。

1. 复位 (由硬件或软件启动) 不受 CPU 的限制, 并立即优先于任何其他执行功能。所有可屏蔽的中断都被禁用, 直到复位服务程序启用它们。

ADP16F03 器件复位源: 一个看门狗定时器超时 (复位) 。

2. 硬件产生的中断请求有外部引脚和片上外设请求。这两种类型如下:

- (1) 外部中断请求由 XINT1、XINT2 和电压比较器 (输出 PDPINTA 信号) 产生。这三个外部中断信号由 CPU 的中断屏蔽寄存器 (IMR) 使能或者屏蔽, IMR 寄存器可以屏蔽 DSP 内核的每个可屏蔽中断线。
- (2) 外设中断是由片上事件管理器 EM1, SPI, SCI, ADC 外设模块事件触发产生。它们可以通过每个外设中的每个事件的使能位以及 CPU 的 IMR 来屏蔽, IMR 寄存器可以屏蔽 DSP 内核中的每个可屏蔽中断线。

3. ADP16F03 器件的软件产生的中断包括:

- (1) INTR 指令。该指令允许用软件初始化任何 ADP16F03 中断。其操作数表示 CPU 分支到的中断向量位置。该指令可禁止全局可屏蔽中断 (将 INTM 位置 1)。
- (2) NMI 指令。该指令迫使分支中断向量指向位置 24H。该指令可禁止全局可屏蔽中断。ADP16F03 设备没有 NMI 硬件信号, 只支持软件触发。
- (3) TRAP 指令。该指令迫使 CPU 分支以中断向量指向位置 22H。TRAP 指令不能禁止可屏蔽中断 (INTM 未设置为 1) 因此, 当 CPU 分支到中断服务程序时, 该程序可被可屏蔽硬件中断中断。
- (4) 仿真器陷阱 该中断可以通过 INTR 指令或 TRAP 指令产生

六个内核中断 (INT1-INT6) 与外设中断(PIE)进行扩展, PIE 管理来自外设级中断, 并分组到六个内核中断。图 8-1 描述了硬件中断 PIE 图, PIE 框图(图 8-1)和中断(表 8-1)描述了 PIE 与 CPU 中断向量对应关系。

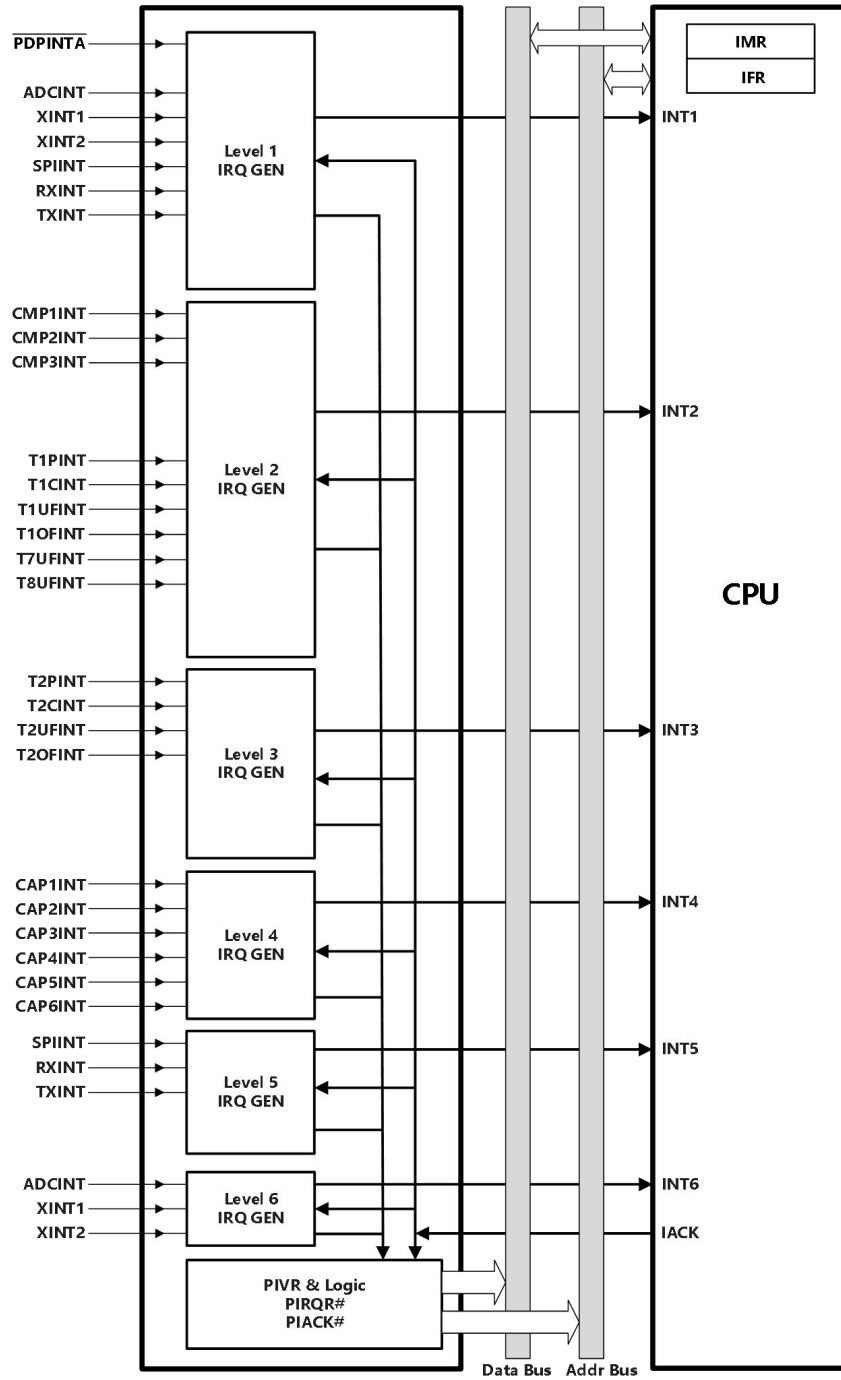


图 8-1 硬件中断 PIE 图

表 8-1 ADP16F03 中断源优先级和中断向量表

中断名称	优先级	CPU 中断向量地址	PIRQRx 和 PIACKRx 对应 bit 位	外设中断向量 (PIE)	可屏蔽中断是否	外设模块	描述	
Reset	1	RSN 0000h	--	N / A	否	RS 引脚, 看门狗	管脚复位, 看门狗超时复位	
Reserved	2	- 0026h	--	N / A	否	CPU	仿真器陷阱	
NMI	3	NMI 0024h	--	N / A	否	非屏蔽中断	不可屏蔽中断, 软件中断	
PDPINTA	4	INT1 0002h	0.0	0020h	是	EM1	电力设备保护中断引脚	
ADCINT	5		0.1	0004h	是	ADC	ADC 中断 (高优先级模式)	
XINT1	6		0.2	0001h	是	外部中断逻辑	外部中断引脚 (高优先级模式)	
XINT2	7		0.3	0011h	是	外部中断逻辑		
SPIINT	8		0.4	0005h	是	SPI	SPI 中断引脚 (高优先级模式)	
RXINT	9		0.5	0006h	是	SCI	SCI 接收中断 (高优先级模式)	
TXINT	10		0.6	0007h	是	SCI	SCI 发送中断 (高优先级模式)	
CMP1INT	11		INT2 0004h	0.9	0021h	是	EM1	比较器 1 产生中断
CMP2INT	12			0.10	0022h	是	EM1	比较器 2 产生中断
CMP3INT	13			0.11	0023h	是	EM1	比较器 3 产生中断
T1PINT	14	0.12		0027h	是	EM1	定时器 1 周期中断	
T1CINT	15	0.13		0028h	是	EM1	定时器 1 比较中断	
T1UFINT	16	0.14		0029h	是	EM1	定时器 1 下溢中断	
T1OFINT	17	0.15		002Ah	是	EM1	定时器 1 溢出中断	
T7UFINT	18	2.6		0031h	是	EM1	定时器 7 上溢中断	
T8UFINT	19	2.7		0032h	是	EM1	定时器 8 上溢中断	
T2PINT	20	INT3 0006h		1.0	002Bh	是	EM1	定时器 2 周期中断
T2CINT	21		1.1	002Ch	是	EM1	定时器 2 比较中断	
T2UFINT	22		1.2	002Dh	是	EM1	定时器 2 下溢中断	
T2OFINT	23		1.3	002Eh	是	EM1	定时器 2 溢出中断	
CAP1INT	24	INT4 0008h	1.4	0033h	是	EM1	捕获 1 中断	
CAP2INT	25		1.5	0034h	是	EM1	捕获 2 中断	
CAP3INT	26		1.6	0035h	是	EM1	捕获 3 中断	
SPIINT	27	INT5	1.7	0005h	是	SPI	SPI 中断 (低优先	

		000Ah					级)
RXINT	28		1.8	0006h	是	SCI	SCI 接收中断 (低优先级模式)
TXINT	29		1.9	0007h	是	SCI	SCI 发送中断 (低优先级模式)
ADCINT	30		1.12	0004h	是	ADC	ADC 中断 (低优先级模式)
XINT1	31	INT6 000Ch	1.13	0001h	是	外部中断逻辑	外部中断引脚 (低优先级模式)
XINT2	32		1.14	0011h	是	外部中断逻辑	
保留的		000Eh	--	N / A	是	中央处理器	保留
陷阱	N / A	0022h	--	N / A	N / A	中央处理器	TRAP 指令
假中断向量	N / A	N / A	--	0000h	N / A	中央处理器	假中断中断向量
INT8-INT16	N / A	0010h-0020h	--	N / A	N / A	中央处理器	软件中断向量
INT20-IN31	N / A	0028h-003Fh	--	N / A	N / A	中央处理器	

8.2 CPU 中断寄存器

CPU 中断寄存器包括：中断标志寄存器 (IFR) 和中断控制寄存器 (IMR)

1. CPU 中断标志寄存器 (IFR) --地址 0006h

15~6	5	4	3	2	1	0
保留位	INT6 flag	INT5 flag	INT4 flag	INT3 flag	INT2 flag	INT1 flag
_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

注：R=可读，W=可写，C=清除，_0=复位值，S=只可被置位；

位 15~6	保留位
位 5	INT6 flag; 中断 6 标志位, 该位用作连至第 6 级中断 INT6 的所有中断标志; 0 ; 无 INT6 的中断悬挂; 1 ; 至少一个 INT6 的中断悬挂, 该位写 1 可将该位清 0, 即清除中断请求。
位 4	INT5 flag; 中断 5 标志位, 该位用作连至第 5 级中断 INT5 的所有中断标志; 0 ; 无 INT5 的中断悬挂; 1 ; 至少一个 INT5 的中断悬挂, 该位写 1 可将该位清 0, 即清除中断请求。
位 3	INT4lag; 中断 4 标志位, 该位用作连至第 4 级中断 INT4 的所有中断标志; 0 ; 无 INT4 的中断悬挂; 1 ; 至少一个 INT4 的中断悬挂, 该位写 1 可将该位清 0, 即清除中断请求。

位 2	INT3 flag; 中断 3 标志位, 该位用作连至第 3 级中断 INT3 的所有中断标志; 0 ; 无 INT3 的中断悬挂; 1 ; 至少一个 INT3 的中断悬挂, 该位写 1 可将该位清 0, 即清除中断请求。
位 1	INT2 flag; 中断 2 标志位, 该位用作连至第 2 级中断 INT2 的所有中断标志; 0 ; 无 INT2 的中断悬挂; 1 ; 至少一个 INT2 的中断悬挂, 该位写 1 可将该位清 0, 即清除中断请求。
位 0	INT1 flag; 中断 1 标志位, 该位用作连至第 1 级中断 INT1 的所有中断标志; 0 ; 无 INT1 的中断悬挂; 1 ; 至少一个 INT1 的中断悬挂, 该位写 1 可将该位清 0, 即清除中断请求。

2. CPU 中断控制寄存器 (IMR) --地址 0004h

15~6	5	4	3	2	1	0
保留位	INT6 mask	INT5 mask	INT4 mask	INT3 mask	INT2 mask	INT1 mask
_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

注: R=可读, W=可写, C=清除, _0=复位值, S=只可被置位;

位 15~6	保留位
位 5	INT6 mask; 中断 6 的屏蔽位; 0 ; 屏蔽中断 INT6; 1 ; 使能中断 INT6;
位 4	INT5 mask; 中断 5 的屏蔽位; 0 ; 屏蔽中断 INT5; 1 ; 使能中断 INT5;
位 3	INT4 mask; 中断 4 的屏蔽位; 0 ; 屏蔽中断 INT4; 1 ; 使能中断 INT4;
位 2	INT3 mask; 中断 3 的屏蔽位; 0 ; 屏蔽中断 INT3; 1 ; 使能中断 INT3;
位 1	INT2 mask; 中断 2 的屏蔽位; 0 ; 屏蔽中断 INT2; 1 ; 使能中断 INT2;
位 0	INT1 mask; 中断 1 的屏蔽位; 0 ; 屏蔽中断 INT1; 1 ; 使能中断 INT1;

8.3 外设中断控制寄存器

外设中断寄存器包括：

1. 外设中断向量寄存器；
2. 外设中断请求寄存器 0/1/2 (PIRQR0/1/2) ；
3. 外设中断应答寄存器 0/1/2 (PIACKR0/1/2) ；

外设中断请求寄存器 0/1/2 和外设中断应答寄存器 0/1/2 都属于外设中断扩展模块。用来向 CPU 产生 INT1~INT6 中断请求的内部寄存器。这些寄存器用于调试的目的，而非用户应用目的，因此编程可忽略；

1. 外设中断向量寄存器 (PIVR) --地址 701Eh

	15~0	V15~V0
	RW_0	

注：R=可读，W=可写，C=清除，_0=复位值，S=只可被置位；

位 15~0	V15~V0:中断向量 V15~V0 位。该寄存器保存了最近一次应答外设中断的地址向量
--------	---------------------------------------------

2. 外设中断请求寄存器 0 (PIRQR0) --7010h

15	14	13	12	11	10	9	8
IRQ0.15	IRQ0.14	IRQ0.13	IRQ0.12	IRQ0.11	IRQ0.10	IRQ0.9	保留
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
保留	IRQ0.6	IRQ0.5	IRQ0.4	IRQ0.3	IRQ0.2	IRQ0.1	IRQ0.0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

注：*写入 1 会发出一个中断请求到 DSP 内核，写入 0 没有影响。

*有中断请求而未被响应，称为悬挂。

位 15~0	IRQ0.0~IRQ0.15: 0 :相应的中断请求为被悬挂 1 :悬挂中断请求
--------	------------------------------------------------

外设中断请求寄存器 PIRQR0 各位的定义

位的位置	中断	中断描述	中断优先级
IRQ0.0	PDPINTA	功率驱动保护引脚中断	INT1
IRQ0.1	ADCINT	高优先级模式 ADC 中断	INT1
IRQ0.2	XINT1	高优先级模式的外部引脚 1 中断	INT1
IRQ0.3	XINT2	高优先级模式的外部引脚 2 中断	INT1
IRQ0.4	SPIINT	高优先级模式的 SPI 中断	INT1
IRQ0.5	RXINT	高优先级模式的 SCI 接收中断	INT1
IRQ0.6	TXINT	高优先级模式的 SCI 发送中断	INT1
保留			
保留			
IRQ0.9	CMP1INT	Compare1 中断	INT2
IRQ0.10	CMP2INT	Compare2 中断	INT2
IRQ0.11	CMP3INT	Compare3 中断	INT2
IRQ0.12	T1PINT	Timer1 周期中断	INT2
IRQ0.13	T1CINTT	Timer1 比较中断	INT2
IRQ0.14	T1UPINT	Timer1 下溢中断	INT2
IRQ0.15	T1OFINT	Timer1 上溢中断	INT2

3. 外设中断请求寄存器 1 (PIRQR1) --7011h

15	14	13	12	11	10	9	8
保留	IRQ1.14	IRQ1.13	IRQ1.12	IRQ1.11	IRQ1.10	IRQ1.9	IRQ1.8
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
保留	IRQ1.6	IRQ1.5	IRQ1.4	IRQ1.3	IRQ1.2	IRQ1.1	IRQ1.0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

注：*写入 1 会发出一个中断请求到 DSP 内核，写入 0 没有影响。

*有中断请求而未被响应，称为悬挂。

位 15~0	IRQ1.0~IRQ1.15: 0 :相应的中断请求为被悬挂 1 :悬挂中断请求
--------	------------------------------------------------

外设中断请求寄存器 PIRQR1 各位的定义

位的位置	中断	中断描述	中断优先级
IRQ1.0	T2PINT	Timer2 周期中断	INT3
IRQ1.1	T2CINT	Timer2 比较中断	INT3
IRQ1.2	T2UFINT	Timer2 下溢中断	INT3
IRQ1.3	T2OFINT	Timer2 上溢中断	INT3
IRQ1.4	CAP1INT	Capture1 中断	INT4
IRQ1.5	CAP2INT	Captuer2 中断	INT4
IRQ1.6	CAP3INT	Capture3 中断	INT4
IRQ1.7	SPIINT	低优先级模式的 SPI 中断	INT5
IRQ1.8	RXINT	低优先级模式的 SCI 接收中断	INT5
IRQ1.9	TXINT	低优先级模式的 SCI 发送中断	INT5
保留			
保留			
IRQ1.12	ADCINT	低优先级模式的 ADC 中断	INT6
IRQ1.13	XINT1	低优先级模式的外部引脚 1 中断	INT6
IRQ1.14	XINT2	低优先级模式的外部引脚 2 中断	INT6
保留			

4. 外设中断请求寄存器 2 (PIRQR2) --7012h

15	14	13	12	11	10	9	8
保留	保留	保留	保留	保留	保留	保留	保留
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
保留	保留	保留	保留	保留	保留	保留	IRQ2.0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

注：*写入 1 会发出一个中断请求到 DSP 内核，写入 0 没有影响。

*有中断请求而未被响应，称为悬挂。

位 15~0	IRQ2.0~IRQ2.15: 0 :相应的中断请求为被悬挂 1 :悬挂中断请求
--------	------------------------------------------------

外设中断请求寄存器 PIRQR2 各位的定义

位的位置	中断	中断描述	中断优先级
IRQ2.0	PDPINTB	功率驱动保护引脚中断	INT1

5. 外设中断应答寄存器 0 (PIACKR0) --地址 7014h

15	14	13	12	11	10	9	8
IAK0.15	IAK0.14	IAK0.13	IAK0.12	IAK0.11	IAK0.10	IAK0.9	保留
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
保留	IAK0.6	IAK0.5	IAK0.4	IAK0.3	IAK0.2	IAK0.1	IAK0.0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

位 15~0	IAK0.0~IAK0.15: 外设中断应答位。写 1 引起相应的外设中断应答被插入，从而将相应的外设中断请求位清 0。
--------	-----------------------------------------------------------------

注意：通过该寄存器写入 1，来插入的中断应答并不更新 PIVR 寄存器的内容，读这个寄存器得到的结果通常是 0。

外设中断应答寄存器 PIACKR0 各位的定义

位的位置	中断	中断描述	中断优先级
IAK0.0	PDPINTA	功率驱动保护引脚中断	INT1
IAK0.1	ADCINT	高优先级模式 ADC 中断	INT1
IAK0.2	XINT1	高优先级模式的外部引脚 1 中断	INT1
IAK0.3	XINT2	高优先级模式的外部引脚 2 中断	INT1
IAK0.4	SPIINT	高优先级模式的 SPI 中断	INT1
IAK0.5	RXINT	高优先级模式的 SCI 接收中断	INT1
IAK0.6	TXINT	高优先级模式的 SCI 发送中断	INT1
保留			
保留			
IAK0.9	CMP1INT	Compare1 中断	INT2
IAK0.10	CMP2INT	Compare2 中断	INT2
IAK0.11	CMP3INT	Compare3 中断	INT2
IAK0.12	T1PINT	Timer1 周期中断	INT2
IAK0.13	T1CINTT	Timer1 比较中断	INT2
IAK0.14	T1UPINT	Timer1 下溢中断	INT2
IAK0.15	T1OFINT	Timer1 上溢中断	INT2

6. 外设中断应答寄存器 1 (PIACKR1) --地址 7015h

15	14	13	12	11	10	9	8
保留	IAK1.14	IAK1.13	IAK1.12	IAK1.11	IAK1.10	IAK1.9	IAK1.8
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
保留	IAK1.6	IAK1.5	IAK1.4	IAK1.3	IAK1.2	IAK1.1	IAK1.0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

位 15~0	IAK1.0~IAK1.15: 外设中断应答位。写 1 引起相应的外设中断应答被插入, 从而将相应的外设中断请求位清 0。
--------	------------------------------------------------------------------

注意: 通过该寄存器写入 1, 来插入的中断应答并不更新 PIVR 寄存器的内容, 读这个寄存器得到的结果通常是 0。

外设中断应答寄存器 PIACKR1 各位的定义

位的位置	中断	中断描述	中断优先级
IAK1.0	PDPINTA	功率驱动保护引脚中断	INT1
IAK1.1	ADCINT	高优先级模式 ADC 中断	INT1
IAK1.2	XINT1	高优先级模式的外部引脚 1 中断	INT1
IAK1.3	XINT2	高优先级模式的外部引脚 2 中断	INT1
IAK1.4	SPIINT	高优先级模式的 SPI 中断	INT1
IAK1.5	RXINT	高优先级模式的 SCI 接收中断	INT1
IAK1.6	TXINT	高优先级模式的 SCI 发送中断	INT1
保留			
保留			
IAK1.9	CMP1INT	Compare1 中断	INT2
IAK1.10	CMP2INT	Compare2 中断	INT2
IAK1.11	CMP3INT	Compare3 中断	INT2
IAK1.12	T1PINT	Timer1 周期中断	INT2
IAK1.13	T1CINTT	Timer1 比较中断	INT2
IAK1.14	T1UPINT	Timer1 下溢中断	INT2
IAK1.15	T1OFINT	Timer1 上溢中断	INT2

7. 外设中断应答寄存器 2 (PIACKR2) --地址 7016h

15	14	13	12	11	10	9	8
保留	保留	保留	保留	保留	保留	保留	保留
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
保留	保留	保留	保留	保留	保留	保留	IAK2.0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

位 15~0	IAK2.0~IAK2.15: 外设中断应答位。写 1 引起相应的外设中断应答被插入, 从而将相应的外设中断请求位清 0。
--------	------------------------------------------------------------------

注意: 通过该寄存器写入 1, 来插入的中断应答并不更新 PIVR 寄存器的内容, 读这个寄存器得到的结果通常是 0。

外设中断应答寄存器 PIACKR1 各位的定义

位的位置	中断	中断描述	中断优先级
IRQ2.0	PDPINTB	功率驱动保护引脚中断	INT1

8.4 外部中断控制寄存器

XINT1CR 和 XINT2CR 为两个控制和监视 XINT1 和 XINT2 引脚状态的外部中断控制器。

1. 外部中断寄存器 (XINT1CR) --地址 7070h

15	14~3	2	1	0
XINT1 flag	保留位	XINT1 polarity	XINT1 priority	XINT1 enable
RC_0	R_0	RW_0	RW-0	RW_0

注：R=可读，W=可写，C=清除，_0=复位值，S=只可被置位；

位 15	XINT1 flag: 该位指示在 XINT1 引脚上是否检测到一个所选的跳变。无论外部中断 1 是否被使能，该位都可置位。当相应的中断被应答，该位被自动清 0。通过软件向该位写 1 (写 0 无效) 或者器件复位时，该位也被清 0。 0 : 未检测到跳变; 1 ; 检测到跳变;
位 14~3	保留位
位 2	XINT1 polarity; XINT1 极性，该位决定中断是上升沿产生还是下降沿产生; 0 ; 在下降沿产生中断 (由高到底跳变) 产生中断; 1 ; 在上升沿产生中断 (由低到高跳变) 产生中断;
位 1	XINT1 priority: XINT1 优先级，该读/写位决定哪一个中断优先级被请求，CPU 的优先级层次和相应的高低优先级已经被编码到外设中断扩展寄存器中，可参加表 8-1 中断源优先级和向量表; 0 ; 高优先级; 1 ; 低优先级;
位 0	XINT1 enable: XINT1 使能位，该读写位使能或禁止外部中断 XINT1; 0 ; 屏蔽中断; 1 ; 使能中断;

2. 外部中断寄存器 (XINT2CR) --地址 7071h

15	14~3	2	1	0
XINT2 flag	保留位	XINT2 polarity	XINT2 priority	XINT2 enable
RC_0	R_0	RW_0	RW-0	RW_0

注：R=可读，W=可写，C=清除，_0=复位值，S=只可被置位；

位 15	XINT2flag: 该位指示在 XINT2 引脚上是否检测到一个所选的跳变。无论外部中断 1 是否被使能，该位都可置位。当相应的中断被应答，该位被自动清 0。通过软件向该位写 1 (写 0 无效) 或者器件复位时，该位也被清 0。 0 : 未检测到跳变; 1 ; 检测到跳变;
位 14~3	保留位

位 2	XINT2 polarity; XINT2 极性, 该位决定中断是上升沿产生还是下降沿产生; 0 ; 在下降沿产生中断 (由高到低跳变) 产生中断; 1 ; 在上升沿产生中断 (由低到高跳变) 产生中断;
位 1	XINT2 priority: XINT2 优先级, 该读/写位决定哪一个中断优先级被请求, CPU 的优先级层次和相应的高低优先级已经被编码到外设中断扩展寄存器中, 可参加表 8-1 中断源优先级和向量表; 0 ; 高优先级; 1 ; 低优先级;
位 0	XINT2 enable: XINT2 使能位, 该读写位使能或禁止外部中断 XINT2; 0 ; 屏蔽中断; 1 ; 使能中断;

9 看门狗

9.1 看门狗模块结构框图

ADP16F03 片内有看门狗 (WD) 定时器模块。如果设备没有定期通过软件写入正确的密钥来监视软件和硬件的操作, 该模块的 WD 功能就会生成一个系统复位信号。WD 定时器独立于 CPU 工作, 它不需要任何 CPU 初始化功能。当发生系统复位时, WD 定时器默认设置为最高速率 (WDCLK 信号= CLKOUT / 512)。一旦内部解除复位, CPU 开始执行代码, WD 定时器开始递增。这意味着, 为了避免过早复位, WD 设置应该在上电复位的早期进行。有关 WD 模块的框图, 请参见图 9-1。WD 模块功能包括以下内容:

WD 定时器 7 种不同 WD 溢出速率一个 WD-Reset KEY (WDKEY) 寄存器, 用于在正确的值被写入时清除 WD 计数器, 并在寄存器写入错误值时产生系统复位如果 WD 控制寄存器 (WDCR) 写入不正确的值, WD 检测对应的控制位后将启动系统复位一旦系统复位被解除, WD 计时器自动激活三个 WD 控制寄存器位于控制寄存器框架中, 从地址 7020h 开始。

注: 此模块中的所有寄存器都是 8 位寄存器, 当一个寄存器被访问时, 所述寄存器的数据在低字节, 高位字节被读为零。写高位字节没有任何效果。

表 9-1 示了不同的 WD 溢出(超时)选择,当 SCCR2 寄存器 (SCSR2.5) 的第 5 位为 1 时, WDCR 寄存器 (WDCR.6) 的第 6 位写 “1” 可以禁止看门狗。如果 SCSR2.5 为 0, 看门狗

将不会被禁用。 SCSR2.5 相当于 ADP16 器件的 WDDIS 引脚。

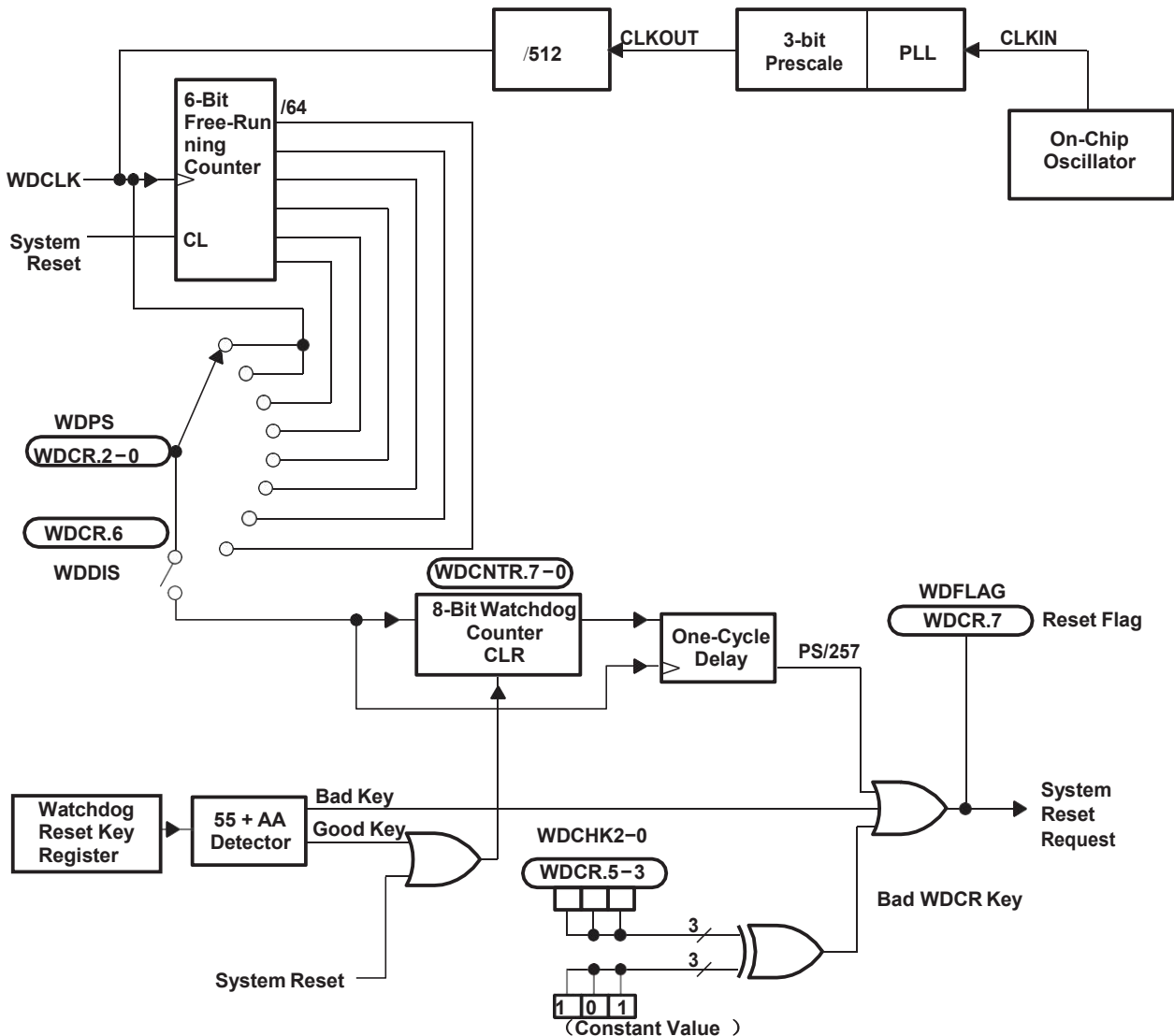


图 9-1 WD 模块的框图

9.2 看门狗控制寄存器

1. WD 计数寄存器 (WDCNTR) --7023h

8 位 WD 计数寄存器存放 WD 计数的当前值 D7~D0。WDCNTR 是一个只读寄存器，复位后为 0，写寄存器无效。

2. WD 复位关键字寄存器 (WDKEY) --7025h

当 55H 及紧接其后的 AAH (复位关键字) 写入 WDKEY 时，将清除 WDCNTR，其他任

何值的结合写入或写入任何值都不能清除 WDCNTR;

8 位 WDKEY 是一个可读写寄存器, 复位后为 0. 读该寄存器 WDKEY 并不返回最近的关键值, 而是返回 WDCR 的内容;

3. WD 定时器控制寄存器 (WDCR) --7029h

8 位 WDCR 用来存放看门狗配置的控制位。

7	6	5	4	3	2	1	0
WDFLAG	WDDIS	WDCHK2	WDCHK1	WDCHK0	WDPS2	WDPS1	WDPS0
RC_x	RWC_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

位 7	WDFLAG; 看门狗标志位。该位指出 WD 定时器是否要求了一个复位系统复位。该位由 WD 产生的复位来置 1, POR 可能引起此位混淆 0; WD 定时器没有要复位; 1; WD 定时器要求一个复位;
位 6	WDDIS; 禁止看门狗位。仅当 SCSR2 中的 WDOVERRIDE 位为 1 时, 向该位写有效; 0; 使能看门狗; 1; 禁止看门狗;
位 5~位 3	WDCHK2~ WDCHK0; 看门狗检查位。必须向着三位写 101, 系统才继续正常运行。否则将要求一个系统复位。读这三位总是 000;
位 2~位 0	WDPS1~ WDPS0; 看门狗预定标因子选择位。这三位选择产生用于 WD 计数器 CLK 的计数溢出分频 (表 9-2 列出了 WD 溢出时间选择)。由于 WD 计数器在溢出前计数 257 个时钟, 所以给出的时间是溢出的最小值。

表 9-1 WD 溢出时间选择

WD 频选择 BITS			WDCLK 除法	看门狗 时钟速率 f
WDPS2	WDPS1	WDPS0		频率 (Hz)
0	0	X#	1	WDCLK / 1
0	1	0	2	WDCLK / 2
0	1	1	4	WDCLK / 4
1	0	0	8	WDCLK / 8
1	0	1	16	WDCLK / 16
1	1	0	32	WDCLK / 32
1	1	1	64	WDCLK / 64

10 硬件加速单元

10.1 硬件加速单元功能特性

硬件加速单元的功能特性描述如下：

(1) 开平方根运算

- 输入数据为 32bit 的无符号数据，运算结果为 16bit 无符号数据；
- 硬件运算周期为 17 个周期；

(2) 除法运算

- 被除数和除数皆为 32bit 有符号整数；
- 实现有符号的整数除法，保留余数，商和余数皆为 32bit；
- 硬件运算周期为 33 个周期；

(3) 乘法移位运算

- 实现 32bit 有符号整数乘法运算；
- 运算结果在无移位操作时为 64bit 的乘法运算结果，在有移位操作时，运算结果保留为移位后的 32bit 数据；
- 硬件运算周期为 17 个周期

(4) park 变换运算

- 输入变量为 α , β , θ 三个 16bit 的有符号变量，输出为 d、q 两个 16bit 有符号数据；
- 实现公式：
$$d = \cos(\theta) * \alpha + \sin(\theta) * \beta;$$
$$q = -\sin(\theta) * \alpha + \cos(\theta) * \beta;$$
- 硬件运算周期为 74 个周期；

(5) 反正切运算

- 坐标 (x, y) 输入为 16bit 的有符号数据;
- 坐标原点偏移可调;
- 坐标预倍乘系数可调;
- 输出角度支持两种模式: Q 模式和 N 模式;
- 硬件运算周期为 15 个周期;

(6) CPU 采用轮询模式读取 FOC 硬件加速单元运算结果;

(7) 具有结果溢出检测、运算中断错误检测等功能;

(8) 各功能模块皆可软复位。

10.2 硬件加速单元各模块使用说明

10.2.1 开平方根模块

1. 开平方根模块使用说明

使用 sqrt 运算开启方法是, 在使能 sqrt 模块之后, CPU 在更新开方运算输入的 32bit 数据使能运算;

2. 开平方根模块寄存器

(1) 开方运算的输入低 16bit 数据寄存器 r_sqrt_dl—地址 7512h

	16~0
r_sqrt_dl	
WR_CAh	

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

位 16~位 0	r_sqrt_dl: 开方运算的输入低 16bit 数据寄存器, 用来保存开平方模块 32 位输入数据的低 16 位输入;
----------	---------------------------------------------------------------

(2) 开方运算的输入高 16bit 数据寄存器 r_sqrt_dh—地址 7513h

	16~0
r_sqrt_dh	
WR_0	

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

位 16~位 0	r_sqrt_dh: 开方运算的输入低 16bit 数据寄存器，用来保存开平方模块 32 位输入数据的高 16 位输入；
----------	--------------------------------------------------------------

(3) 开方运算控制寄存器 r_sqrt_ctrl—地址 7514h

15~3	2	1	0
保留	soft reset	sqrt_cal_en	r_st_clr
WR_0	WR_1	WR_1	WR_0

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

位 15~位 3	保留
位 2	soft reset: 开方模块复位信号，低电平有效；
位 1	sqrt_en: 开方运算使能位； 0: 禁止开方运算模块 1: 使能开方运算模块
位 0	r_st_clr: 状态寄存器 r_sqrt_st 清除信号，高电平有效；

(4) 开方模块状态寄存器 r_sqrt_st—地址 7515h

15~4	3	2	1	0
保留位	r_sqrt_rdy	r_sqrt_busy	r_sqrt_of	r_sqrt_err
R_0	R_0	R_0	R_0	R_0

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

位 15~4	保留
位 3	r_sqrt_rdy: 运算结束并等待 CPU 获取运算结果的标志位； 0: 标志被复位 1: 标志被置位
位 2	r_sqrt_busy: 运算处于忙状态的标志位； 0: 标志被复位 1: 标志被置位
位 1	r_sqrt_of: 运算结果溢出标志位； 0: 标志被复位 1: 标志被置位
位 0	r_sqrt_err: 运算中断错误标志位； 0: 标志被复位 1: 标志被置位

(5) 开方运算结果寄存器 r_sqrt_q—地址 7516h

16~0
r_sqrt_q
R_0

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

位 16~位 0	r_sqrt_q: 开方运算结果寄存器, 用来保存开方运算的输出结果;
----------	-------------------------------------

10.2.2 除法器模块

1. 除法器模块使用说明

除法器在开启运算方式上分为手动开启运算和自动开启运算两种方式, 由寄存器 div_ctrl 控制。

当 div_ctrl[2] 设置为 1'b1, 则为手动开启模式, 此时, 只要将 div_ctrl[1] 先设置为 1'b1, 再设置为 1'b0, 则立即开启除法运算,

当 div_ctrl[2] 设置为 1'b0, 则为自动开启模式, 此时只要将被除数和除数设置完成, 则自动开启除法运算。

2. 除法器模块寄存器列表

(1) 被除数低 16bit 数据寄存器 r_div_al—7508h

16~0	r_div_al
WR_00CAh	

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

位 16~位 0	r_div_al: 被除数低 16bit 数据寄存器, 用来保存除法器模块 32 位被除数的低 16 位数据;
----------	---------------------------------------------------------

(2) 被除数高 16bit 数据寄存器 r_div_ah—7509h

16~0	r_div_ah
WR_0900h	

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

位 16~位 0	r_div_ah: 被除数高 16bit 数据寄存器, 用来保存除法器模块 32 位被除数的高 16 位数据;
----------	---------------------------------------------------------

(3) 除数低 16bit 数据寄存器 r_div_bl—750Ah

16~0	r_div_bl
WR_357h	

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

位 16~位 0	r_div_bl: 被除数低 16bit 数据寄存器, 用来保存除法器模块 32 位除数的低 16 位数据;
----------	--------------------------------------------------------

(4) 除数高 16bit 数据寄存器 r_div_bh—750Bh

16~0	r_div_bh
WR_0	

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

位 16~位 0	r_div_bh: 除数高 16bit 数据寄存器, 用来保存除法器模块 32 位除数的高 16 位数据;
----------	-------------------------------------------------------

(5) 除法控制寄存器 r_div_ctrl—地址 750Ch

15~5	4	3	2	1	0
保留	soft reset	r_div_en	r_div_m	r_hand_div	r_st_clr
WR_0	WR_1	WR_1	WR_0	WR_0	WR_0

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

位 15~位 5	保留
位 4	soft reset: 除法器模块复位信号, 低电平有效;
位 3	r_div_en: 除法器模块使能信号, 高电平有效;
位 2	r_div_m: 运算开启模式选择位; 0: 自动模式 1: 手动模式
位 1	r_hand_div: 手动开启运算控制位; 先置高电平, 再置低电平将开启一次运算
位 0	r_st_clr: 状态寄存器清除使能位; 0: 状态寄存器保持 1: 状态寄存器清除为 0

(6) 除法器模块状态寄存器 r_div_st—地址 750Dh

15~4	3	2	1	0
保留	r_div_rdy	r_div_busy	r_div_of	r_div_err
R_0	R_0	R_0	R_0	R_0

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

位 15~位 4	保留
位 3	r_div_rdy: 运算并等待 CPU 获取结果的标志;
位 2	r_div_busy: 运算处于忙状态的标志;
位 1	r_div_of: 运算结果溢出标志;
位 0	r_div_err: 运算中断错误标志;

(7) 除法运算商的低 16bit 数据寄存器 r_div_ql—750Eh

16~0

r_div_ql

R_0

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

位 16~位 0	r_div_ql: 除法运算商的低 16bit 数据寄存器, 用来保存除法运算 32 位商的低 16 位数据;
----------	---------------------------------------------------------

(8) 除法运算商的高 16bit 数据寄存器 r_div_qh—750Fh

16~0

r_div_qh

R_0

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

位 16~位 0	r_div_qh: 除法运算商的高 16bit 数据寄存器, 用来保存除法运算 32 位商的高 16 位数据;
----------	---------------------------------------------------------

(9) 除法运算余数的低 16bit 数据寄存器 r_div_rl—7510h

16~0

r_div_rl

R_0

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

位 16~位 0	r_div_rl: 除法运算余数低 16bit 数据寄存器, 用来保存除法器模块 32 位余数的低 16 位数据;
----------	-----------------------------------------------------------

(10) 除法运算余数的高 16bit 数据寄存器 r_div_rh—7511h

16~0

r_div_rh

R_0

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

位 16~位 0	r_div_rh: 除法运算余数高 16bit 数据寄存器, 用来保存除法器模块 32 位余数的高 16 位数据;
----------	-----------------------------------------------------------

10.2.3 park 模块

1. park 模块使用说明

park 变换需要输入三个 16bit 位宽的数据变量, 当 CPU 更新这三个变量的值以后, 自动开启 park 变换。

2.park 模块寄存器列表

(1) park 变换 alph 变量输入数据寄存器 r_park_alph---地址 7517h

15~0

r_park_alph
WR_CAh

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

位 15~位 0	r_park_alph: park 变换 alph 变量输入数据寄存器，用来保存 park 模块输入数据 alph 的值；
----------	---------------------------------------------------------------

(2) park 变换 beta 变量输入数据寄存器 r_park_beta---地址 7518h

15~0

r_park_beta
WR_CAh

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

位 15~位 0	r_park_beta: park 变换 beta 变量输入数据寄存器，用来保存 park 模块输入数据 beta 的值；
----------	---------------------------------------------------------------

(3) park 变换 theta 变量输入数据寄存器 r_park_theta---地址 7519h

15~0

r_park_theta
WR_CAh

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

位 15~位 0	r_park_theta: park 变换 theta 变量输入数据寄存器，用来保存 park 模块输入数据 theta 的值；
----------	------------------------------------------------------------------

(4) park 变换控制寄存器 r_park_ctrl—地址 751Ah

15~3

2

1

0

保留	soft reset	park_cal_en	r_st_clr
RW_0	RW_1	RW_1	RW_0

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

位 15~位 3	保留
位 2	soft reset: 复位信号，低电平有效
位 1	park_cal_en: park 运算使能信号，高电平有效
位 0	r_st_clr: 状态寄存器清除信号，高电平有效

(5) park 变换状态寄存器 r_park_st—地址 751Bh

15~4

3

2

1

0

保留	r_park_rdy	r_park_busy	r_park_of	r_park_err
R_0	R_0	R_0	R_0	R_0

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

位 15~位 4	保留
位 3	r_park_rdy: 运算结束并等待 CPU 获取运算结果的标志
位 2	r_park_busy: 运算处于忙状态的标志
位 1	r_park_of: 运算结果溢出标志
位 0	r_park_err: 运算错误中断标志

(6) park 变换结果 mtd 数据寄存器 r_park_mtd—地址 751C

15~0	r_park_mtd
R_0	

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

位 15~位 0	r_park_mtd: park 变换结果 mtd 数据寄存器，用来保存 park 模块运算结果 mtd 的数据；
----------	-----------------------------------------------------------

(7) park 变换结果 mtq 数据寄存器 r_park_mtq—地址 751D

15~0	r_park_mtq
R_0	

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

位 15~位 0	r_park_mtq: park 变换结果 mtq 数据寄存器，用来保存 park 模块运算结果 mtq 的数据；
----------	-----------------------------------------------------------

10.2.4 乘法移位器模块

1. 乘法移位器模块使用说明

(1) 乘法移位器运算的开启方式

乘法运算需要 CPU 写入两个 32bit 的运算变量，采取了两种运算开启的方式，说明如下：

CPU 更新两个 32bit 的运算变量后自动开启乘法移位运算；

CPU 通过 r_multi_ctrl[1]寄存器控制位，手动开启运算，方法是先将 r_multi_ctrl[2]置高电平使能手动开启模式，之后在需要运算的时候将 r_multi_ctrl[1]置高，再置低即开启一次运算。

(2) CPU 读取运算结果结束标志的设计

由于乘法运算结果 multi_q 具有 64bit, 当存在移位操作的情况下, 只需读取 multi_q[31:0], 无移位操作时读取 multi_q 的 64bit 数据, CPU 读取结束标志 q_empty 将置高电平, 表示读取结束。

2. 乘法移位器寄存器列表

(1) 乘法移位运算输入变量 multi_a 低 16 位数据寄存器 r_multi_al—地址 751Eh

15~0

r_multi_al

RW_1CAh

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

位 15~位 0

r_multi_al: 乘法移位运算输入变量 multi_a 低 16 位数据寄存器, 用来保存输入变量 multi_a 的低 16 位数据

(2) 乘法移位运算输入变量 multi_a 高 16 位数据寄存器 r_multi_ah—地址 751Fh

15~0

r_multi_ah

RW_CAh

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

位 15~位 0

r_multi_ah: 乘法移位运算输入变量 multi_a 高 16 位数据寄存器, 用来保存输入变量 multi_a 的高 16 位数据

(3) 乘法移位运算输入变量 multi_b 低 16 位数据寄存器 r_multi_bl—地址 7520h

15~0

r_multi_bl

RW_1CAh

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

位 15~位 0

r_multi_bl: 乘法移位运算输入变量 multi_b 低 16 位数据寄存器, 用来保存输入变量 multi_b 的低 16 位数据

(4) 乘法移位运算输入变量 multi_b 高 16 位数据寄存器 r_multi_bh—地址 7521h

15~0

r_multi_bh

RW_CAh

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

位 15~位 0	r_multi_bh: 乘法移位运算输入变量 multi_b 高 16 位数据寄存器, 用来保存输入变量 multi_b 的高 16 位数据
----------	------------------------------------------------------------------------

(5) 乘法移位器的移位控制变量寄存器 r_multi_sf—地址 7522h

15~6	5~0
保留	r_multi_sf
RW_0	RW_0

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

位 15~位 6	保留
位 5~位 0	r_multi_sf: 乘法移位器的移位控制变量, 移位大小 0~32

(6) 乘法移位器控制寄存器 r_multi_ctrl—地址 7523h

15~5	4	3	2	1	0
保留	soft reset	r_multi_en	r_multi_m	r_hand_multi	r_st_clr
RW_0	RW_1	RW_1	RW_0	RW_0	RW_0

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

位 15~位 5	保留
位 4	soft reset: 复位信号, 低电平有效
位 3	r_multi_en: 乘法移位去模块使能信号, 高电平有效
位 2	r_multi_m: 运算开启模式选择位: 1: 手动模式 0: 自动模式
位 1	r_hand_multi: 手动开启运算控制位; 先置高电平, 再置低电平将开启一次运算
位 0	r_st_clr: 状态寄存器 r_multi_st 清除信号, 高电平有效

(7) 乘法移位器状态寄存器 r_multi_st—地址 7524h

15~4	3	2	1	0
保留	r_multi_rdy	r_multi_busy	r_multi_of	r_multi_err
R_0	R_0	R_0	R_0	R_0

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

位 15~位 4	保留
位 3	r_multi_rdy: 运算结束并等待 CPU 获取运算结果的标志
位 2	r_multi_busy: 运算处于忙状态的标志
位 1	r_multi_of: 运算结果溢出标志
位 0	r_multi_err: 运算中断错误标志

(8) 乘法移位运算结果 r_multi_qll 寄存器—地址 7525h

15~0
r_multi_qll

R_0

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

位 15~位 0	r_multi_qll: 乘法移位运算结果 r_multi_qll 寄存器, 用来保存运算结果 64 位变量的 0~15 位数据;
----------	-------------------------------------------------------------------

(9) 乘法移位运算结果 r_multi_qlh 寄存器—地址 7526h

15~0

r_multi_qlh

R_0

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

位 15~位 0	r_multi_qlh: 乘法移位运算结果 r_multi_qlh 寄存器, 用来保存运算结果 64 位变量的 16~31 位数据;
----------	--------------------------------------------------------------------

(10) 乘法移位运算结果 r_multi_qhl 寄存器—地址 7527h

15~0

r_multi_qhl

R_0

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

位 15~位 0	r_multi_qhl: 乘法移位运算结果 r_multi_qhl 寄存器, 用来保存运算结果 64 位变量的 32~47 位数据;
----------	--------------------------------------------------------------------

(11) 乘法移位运算结果 r_multi_qhh 寄存器—地址 7528h

15~0

r_multi_qhh

R_0

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

位 15~位 0	r_multi_qhh: 乘法移位运算结果 r_multi_qhh 寄存器, 用来保存运算结果 64 位变量的 47~63 位数据;
----------	--------------------------------------------------------------------

10.2.5 Cordic 模块

1. Cordic 模块使用说明

Cordic 模块在开启运算方式上分为手动开启运算和自动开启运算两种方式，由寄存器 r_cor_ctrl 寄存器控制。

当 r_cor_ctrl[1] 设置为 1' b1，则为手动开启模式，此时，只要将 r_cor_ctrl[3] 先设置为 1' b1，再设置为 1' b0，则立即开启角度运算，

当 r_cor_ctrl[1] 设置为 1' b0，则为自动开启模式，此时只要将 坐标 (X, Y) 设置好，则

自动开启角度运算。

Cordic 模块支持两种角度表达方式，一种是 N 模式，一种 Q 模式。

(1) N 模式下：绝对误差 $< \pm 0.5^\circ$ 。

(2) Q 模式下：最小绝对误差能达到 $< \pm 0.0055^\circ$ 。

2. Cordic 模块寄存器列表

(1) 坐标 (x, y) 中的 x 输入变量寄存器 r_cor_x—地址 7500h

15~0	r_cor_x
RW_150h	

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

位 15~位 0	r_cor_x: 坐标 (x, y) 中的 x 输入变量寄存器，用来保存输入变量 x 的数据；
----------	-------------------------------------------------

(2) 坐标 (x, y) 中的 y 输入变量寄存器 r_cor_y—地址 7501h

15~0	r_cor_y
RW_80h	

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

位 15~位 0	r_cor_y: 坐标 (x, y) 中的 y 输入变量寄存器，用来保存输入变量 y 的数据；
----------	-------------------------------------------------

(3) Cordic 运算结果 (Q&N 模式) 寄存器 r_agl—地址 7502h

15~0	r_agl
RW_0	

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

位 15~位 0	r_agl: Cordic 运算结果 (Q&N 模式) r_agl, 用来保存运算之后的到的角度；
----------	---------------------------------------------------

(4) cordic 控制寄存器 r_cor_ctrl—地址 7504h

15~9	8	7~4	3	2	1	0
保留	soft reset	r_pre_amp	r_hand_ctrl	r_angle_m	r_cal_m	r_st_clr
WR_0	WR_1	WR_Ch	WR_0	WR_1	WR_0	WR_0

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

位 15~位 9	保留
位 8	soft reset: 复位信号，低电平有效

位 7~位 4	r_pre_amp: 表示对坐标 (x, y) 的倍乘 2r_pre_amp
位 3	r_hand_ctrl: 手动模式下的手动开启控制信号, 先置高电平再置低电平操作即开启一次运算
位 2	r_angle_m: 角度模式选择位; 1: Q 模式 0: N 模式
位 1	r_cal_m: 运算开启模式选择位; 1: 手动模式 0: 自动模式
位 0	r_st_clr: 状态寄存器 r_cor_st 清除信号, 高电平有效

(5) cordic 模块状态寄存器 r_cor_st—地址 7505h

15~4	3	2	1	0
保留	r_rdy	r_busy	r_cor_of	r_cor_err
R_0	R_0	R_0	R_0	R_0

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

位 15~位 4	保留
位 3	r_rdy: 运算结束并等待 CPU 获取运算结果的标志
位 2	r_busy: 运算处于忙状态的标志
位 1	r_cor_of: 运算结果溢出标志
位 0	r_cor_err: 运算中断错误标志

(6) 坐标 (x, y) 的 x 预偏置大小设置寄存器 r_x_offset—地址 7506h

15~0
r_x_offset
RW_0

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

位 15~位 0	r_x_offset: 坐标 (x, y) 的 x 预偏置大小设置寄存器, 用来设置 x 预偏置大小
----------	----------------------------------------------------

(7) 坐标 (x, y) 的 y 预偏置大小设置寄存器 r_y_offset—地址 7507h

15~0
r_y_offset
RW_0

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

位 15~位 0	r_y_offset: 坐标 (x, y) 的 y 预偏置大小设置寄存器, 用来设置 y 预偏置大小
----------	----------------------------------------------------

11 数字 I/O 口以及复用引脚功能

11.1 数字 I/O 口以及复用引脚功能描述

ADP16F03 具有 30 个通用双向数字 I/O (GPIO) 引脚，其中大部分引脚属于功能和通用 I/O 复用硬件。所有的 I/O 的数字功能和通用 IO 功能都由 9 个 16 位寄存器控制。这些寄存器分为两种类型：

- (1) 三个输出控制寄存器：MCRA, MCRB, MCRC 用于控制引脚在的数字功能或通用 I/O 功能在多路复用器中的输出选择。
- (2) 七个数据和控制寄存器：PADATDIR, PBDATDIR, PCDATDIR, PDDATDIR, PEDATDIR, PFDATDIR0, PFDATDIR1 用于控制双向 I/O 引脚的数据和数据方向。

11.1.1 复用 I/O 引脚的描述

通用 I/O 引脚的控制结构如图 11-1 所示，其中每个引脚都有三个位来定义其操作：

- (1) Mux 控制位：该位在引脚的主功能 (1) 和 I/O 功能 (0) 之间进行选择。
- (2) I/O 方向位：如果引脚选择了 I/O 功能 (多路复用控制位设置为 0)，该位决定引脚是输入 (0) 还是输出 (1)。
- (3) I/O 数据位：如果引脚选择了 I/O 功能 (多路复用控制位设置为 0)，并且所选方向为输入，则从该位读取数据；如果选择的方向是输出，则将数据写入该位。
- (4) 多路复用控制位，I/O 方向位和 I/O 数据位在 I/O 控制寄存器中。

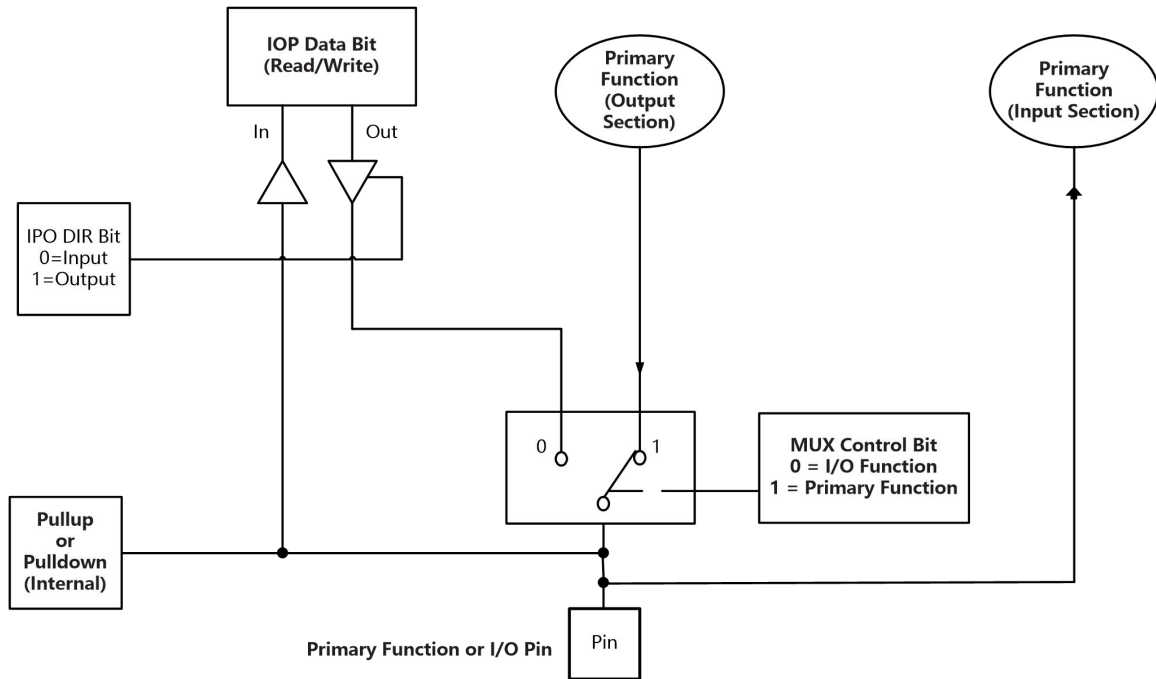


图 11-1 复用引脚配置

11.2 数字 I / O 控制寄存器

表11-2列出了数字I / O模块中可用的寄存器，这些寄存器都是内存映射到数据空间的。

表11-2

数字 I / O 控制寄存器的地址地址	寄存器	名称
7090h	MCRA	I / O 多路复用控制寄存器 A
7092h	MCRB	I / O 多路复用控制寄存器 B
7094h	MCRC	I / O 多路复用控制寄存器 C
7095h	PEDATDIR	I / O 端口 E 数据和方向寄存器
7096h	PFDATDIR0	I / O 端口 F 数据和方向寄存器 0
7103h	PFDATDIR1	I / O 端口 F 数据和方向寄存器 1
7098h	PADATDIR	I / O 端口 A 数据和方向寄存器
709Ah	PBDATDIR	I / O 端口 B 数据和方向寄存器
709Ch	PCDATDIR	I / O 端口 C 数据和方向寄存器
709Eh	PDDATDIR	I / O 端口 D 数据和方向寄存器

11.2.1 I/O 端口复用输出控制寄存器 PORTA, PORTB

1. I/O 端口复用输出控制寄存器 A (MCRA) --地址 7090h

15	14	13	12	11	10	9	8
MCRA.15	MCRA.14	MCRA.13	MCRA.12	MCRA.11	MCRA.10	MCRA.9	MCRA.8

RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
MCRA.7	MCRA.6	MCRA.5	MCRA.4	MCRA.3	MCRA.2	MCRA.1	MCRA.0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

位	位的名称	引脚功能选择	
		基本功能 (MCRA.n=1)	基本功能 (MCRA.n=0)
15	MCRA.15	保留	保留
14	MCRA.14	保留	保留
13	MCRA.13	保留	保留
12	MCRA.12	保留	保留
11	MCRA.11	PWM6	IOPB3
10	MCRA.10	PWM5	IOPB2
9	MCRA.9	PWM4	IOPB1
8	MCRA.8	PWM3	IOPB0
7	MCRA.7	PWM2	IOPA7
6	MCRA.6	PWM1	IOPA6
5	MCRA.5	CAP3	IOPA5
4	MCRA.4	CAP2/QEP2	IOPA4
3	MCRA.3	CAP1/QEP1	IOPA3
2	MCRA.2	XINT1	IOPA2
1	MCRA.1	SCIRXD	IOPA1
0	MCRA.0	SCITXD	IOPA0

2. I/O 端口 A 数据和方向控制寄存器 (PADATDIR) --地址 7098h

15	14	13	12	11	10	9	8
A7DIR	A6DIR	A5DIR	A4DIR	A3DIR	A2DIR	A1DIR	A0DIR
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
IOPA7	IOPA6	IOPA5	IOPA4	IOPA3	IOPA2	IOPA1	IOPA0
RW_+	RW_+	RW_+	RW_+	RW_+	RW_+	RW_+	RW_+

注：R=可读，W=可写，C=清除，_0=复位值，_X=复位值不确定，_+复位后的值与相应引脚状态有关，S=只可被置位；

位 15~位 8	AnDIR: 0: 配置相应引脚为输入方式 1: 配置相应引脚为输出方式
位 7~位 0	当 AnDIR=0, 即引脚为输入方式; 0: 读相应引脚的值为低电平 1: 读相应引脚的值为高电平

	当 AnDIR=1, 即引脚为输出方式; 0: 设置相应的引脚, 使其输出为低电平 1: 设置相应的引脚, 使其输出为高电平
--	----------------------------------------------------------------------

3. I/O 端口 B 数据和方向控制寄存器 (PBDATDIR) --地址 709Ah

15	14	13	12	11	10	9	8
保留	保留	保留	保留	B3DIR	B2DIR	B1DIR	B0DIR
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
保留	保留	保留	保留	IOPB3	IOPB2	IOPB1	IOPB0
RW_+	RW_+	RW_+	RW_+	RW_+	RW_+	RW_+	RW_+

注: R=可读, W=可写, C=清除, _0=复位值, _X=复位值不确定, _+复位后的值与相应引脚状态有关, S=只可被置位;

位 15~位 8	BnDIR: 0: 配置相应引脚为输入方式 1: 配置相应引脚为输出方式
位 7~位 0	当 BnDIR=0, 即引脚为输入方式; 0: 读相应引脚的值为低电平 1: 读相应引脚的值为高电平 当 BnDIR=1, 即引脚为输出方式; 0: 设置相应的引脚, 使其输出为低电平 1: 设置相应的引脚, 使其输出为高电平

11.2.2 I/O 端口复用输出控制寄存器 PORTC, PORTD

1. I/O 端口复用输出控制寄存器 B (MCRB) --地址 7092h

15	14	13	12	11	10	9	8
MCRB.15	MCRB.14	MCRB.13	MCRB.12	MCRB.11	MCRB.10	MCRB.9	MCRB.8
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
MCRB.7	MCRB.6	MCRB.5	MCRB.4	MCRB.3	MCRB.2	MCRB.1	MCRB.0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

位	位的名称	引脚功能选择	
		基本功能 (MCRB.n=1)	基本功能 (MCRB.n=0)
15	MCRB.15	保留	保留
14	MCRB.14	TMS	IOPD6
13	MCRB.13	TDO	IOPD5

12	MCRB.12	TDI	IOPD4
11	MCRB.11	TCK	IOPD3
10	MCRB.10	保留	保留
9	MCRB.9	保留	保留
8	MCRB.8	XINT2/ADCSOC	IOPD0
7	MCRB.7	保留	保留
6	MCRB.6	保留	保留
5	MCRB.5	SPISTE	IOPC5
4	MCRB.4	SPICLK	IOPC4
3	MCRB.3	SPISOMI	IOPC3
2	MCRB.2	SPISIMO	IOPC2
1	MCRB.1	保留	保留
0	MCRB.0	保留	保留

2. I/O 端口 C 数据和方向控制寄存器 (PCDATDIR) --地址 709Ch

15	14	13	12	11	10	9	8
保留	保留	C5DIR	C4DIR	C3DIR	C2DIR	保留	保留
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
保留	保留	IOPC5	IOPC4	IOPC3	IOPC2	保留	保留
RW_+	RW_+	RW_+	RW_+	RW_+	RW_+	RW_+	RW_+

注：R=可读，W=可写，C=清除，_0=复位值，_X=复位值不确定，_+复位后的值与相应引脚状态有关，S=只可被置位；

位 15~位 8	CnDIR: 0: 配置相应引脚为输入方式 1: 配置相应引脚为输出方式
位 7~位 0	当 CnDIR=0, 即引脚为输入方式; 0: 读相应引脚的值为低电平 1: 读相应引脚的值为高电平 当 CnDIR=1, 即引脚为输出方式; 0: 设置相应的引脚, 使其输出为低电平 1: 设置相应的引脚, 使其输出为高电平

3. I/O 端口 D 数据和方向控制寄存器 (PDDATDIR) --地址 709Eh

15	14	13	12	11	10	9	8
保留	D6DIR	D5DIR	D4DIR	D3DIR	保留	保留	D0DIR
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

7	6	5	4	3	2	1	0
保留	IOPD6	IOPD5	IOPD4	IOPD3	保留	保留	IOPD0
RW_+	RW_+	RW_+	RW_+	RW_+	RW_+	RW_+	RW_+

注：R=可读，W=可写，C=清除，_0=复位值，_X=复位值不确定，_+复位后的值与相应引脚状态有关，S=只可被置位；

位 15~位 8	DnDIR: 0: 配置相应引脚为输入方式 1: 配置相应引脚为输出方式
位 7~位 0	当 DnDIR=0, 即引脚为输入方式; 0: 读相应引脚的值为低电平 1: 读相应引脚的值为高电平 当 DnDIR=1, 即引脚为输出方式; 0: 设置相应的引脚, 使其输出为低电平 1: 设置相应的引脚, 使其输出为高电平

11.2.3 I/O 端口复用输出控制寄存器 PORTE,PORTF

1. I/O 端口复用输出控制寄存器 C (MCRC) --地址 7094h

15	14	13	12	11	10	9	8
MCRC.15	MCRC.14	MCRC.13	MCRC.12	MCRC.11	MCRC.10	MCRC.9	MCRC.8
RW_0	保留	保留	保留	保留	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
MCRC.7	保留	保留	保留	保留	MCRC.2	MCRC.1	MCRC.0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

位	位的名称	引脚功能选择	
		基本功能 (MCRC.n=1)	基本功能 (MCRC.n=0)
15	MCRC.15	保留	保留
14	MCRC.14	保留	保留
13	MCRC.13	保留	保留
12	MCRC.12	OP1N	IOPF3(只做输入)
11	MCRC.11	OP1P	IOPF4(只做输入)
10	MCRC.10	B2_OP2P	IOPF2(只做输入)
9	MCRC.9	B3_OP2N	IOPF1(只做输入)
8	MCRC.8	B4_OP3P	IOPF0(只做输入)
7	MCRC.7	B5_OP3N	IOPE7(只做输入)
6	MCRC.6	B6_OP4P	IOPE6(只做输入)
5	MCRC.5	B7_OP4N	IOPE5(只做输入)
4	MCRC.4	保留	IOPE4/ BOOT 分支切换 (预留)
3	MCRC.3	保留	保留

2	MCRC.2	保留	保留
1	MCRC.1	PWM7	IOPE1
0	MCRC.0	保留	保留

2. I/O 端口 E 数据和方向控制寄存器 (PEDATDIR) --地址 7095h

15	14	13	12	11	10	9	8
保留	保留	保留	E4DIR	保留	保留	E1DIR	保留
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
IOPE7	IOPE6	IOPE5	IOPE4	保留	保留	IOPE1	保留
RW_+	RW_+	RW_+	RW_+	RW_+	RW_+	RW_+	RW_+

注：R=可读，W=可写，C=清除，_0=复位值，_X=复位值不确定，_+复位后的值与相应引脚状态有关，S=只可被置位；

位 15~位 8	EnDIR: 0: 配置相应引脚为输入方式 1: 配置相应引脚为输出方式
位 7~位 0	当 EnDIR=0, 即引脚为输入方式; 0: 读相应引脚的值为低电平 1: 读相应引脚的值为高电平 当 EnDIR=1, 即引脚为输出方式; 0: 设置相应的引脚, 使其输出为低电平 1: 设置相应的引脚, 使其输出为高电平

3. I/O 端口 F 数据和方向控制寄存器 (PFDATDIR1) --地址 7103h

OP1N, OP1P, B2_OP2P, B3_OP2N, B4_OP3P, B5_OP3N, B6_OP4P, B7_OP4N

作为基本功能只做输入使用，且需要配置寄存器PFDATDIR1。

15	14	13	12	11	10	9	8
COMPH	COMPL	PDPA	保留	保留	保留	保留	保留
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
REG_INPUT_ CT7	REG_INPUT_ CT6	REG_INPUT_ CT5	REG_INPUT_ CT4	REG_INPUT_ CT3	REG_INPUT_ CT2	REG_INPUT_ CT1	REG_INPUT_ CT0
RW_+	RW_+	RW_+	RW_+	RW_+	RW_+	RW_+	RW_+

位 15	COMPH: 比较器 COMPH 状态位; 1: A0_COMP0 输入电压超过上限电压; 0: A0_COMP0 输入电压未超过上限电压;
------	------------------------------------------------------------------------------

位 14	COMPL: 比较器 COMPH 状态位; 1: A0_COMP0 输入电压低于下限电压; 0: A0_COMP0 输入电压未低于下限电压;
位 13	PDPA: 为 PDPINTA 输入源控制切换位; 1: 配置 PDPINTA 输入源使用 COMPH/COMPL 0: 配置 PDPINTA 输入源使用 B7_OP4N
位 12	保留:
位 11	保留:
位 10~位 9	保留:
位 8	保留:
位 7	REG_INPUT_CT7: 复用 IOPE5 输入使能位; 0: 配置端口为 ADC 输入 B7 1: 配置端口为数字 IO 输入端口 IOPE5
位 6	REG_INPUT_CT6: 复用 IOPE6 输入使能位; 0: 配置端口为 ADC 输入 B6 1: 配置端口为数字 IO 输入端口 IOPE6
位 5	REG_INPUT_CT5: 复用 IOPE7 输入使能位; 0: 配置端口为 ADC 输入 B5 1: 配置端口为数字 IO 输入端口 IOPE7
位 4	REG_INPUT_CT4: 复用 IOPF0 输入使能位; 0: 配置端口为 ADC 输入 B4 1: 配置端口为数字 IO 输入端口 IOPF0
位 3	REG_INPUT_CT3: 复用 IOPF1 输入使能位; 0: 配置端口为 ADC 输入 B3 1: 配置端口为数字 IO 输入端口 IOPF1
位 2	REG_INPUT_CT2: 复用 IOPF2 输入使能位; 0: 配置端口为 ADC 输入 B2 1: 配置端口为数字 IO 输入端口 IOPF2
位 1	REG_INPUT_CT1: 复用 IOPF3 输入使能位; 0: 配置端口为 ADC 输入 B1 1: 配置端口为数字 IO 输入端口 IOPF3
位 0	REG_INPUT_CT0: 复用 IOPF4 输入使能位; 0: 配置端口为 ADC 输入 B0 1: 配置端口为数字 IO 输入端口 IOPF4

4. I/O 端口 F 数据和方向控制寄存器 (PFDATDIR0) --地址 7096h

15	14	13	12	11	10	9	8
保留	保留	保留	保留	保留	保留	保留	保留
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
保留	保留	保留	IOPF4	IOPF3	IOPF2	IOPF1	IOPF0

RW_+ RW_+ RW_+ RW_+ RW_+ RW_+ RW_+ RW_+

注：R=可读，W=可写，C=清除，_0=复位值，_X=复位值不确定，+_复位后的值与相应引脚状态有关，S=只可被置位；

位 15~位 8	FnDIR: 0: 配置相应引脚为输入方式 1: 配置相应引脚为输出方式
位 7~位 0	当 FnDIR=0, 即引脚为输入方式; 0: 读相应引脚的值为低电平 1: 读相应引脚的值为高电平 当 FnDIR=1, 即引脚为输出方式; 0: 设置相应的引脚, 使其输出为低电平 1: 设置相应的引脚, 使其输出为高电平

12 串行通信接口 (SCI) 模块

12.1 串行通信接口 (SCI) 模块简介

ADP16F03 包含一个串行通信接口 (SCI) 模块, 串口通信模块的寄存器的位宽为 8 位。SCI 支持 CPU 和其他使用标准 NRZ (非归零) 格式的异步外设之间进行异步串行数字通信。SCI 的接收器和发送器是双缓冲的, 各自有独立使能和中断位, 两者均可以独立工作, 或者在全双工模式下同时工作。为了确保数据的完整性, SCI 会对收到的数据进行测试, 如间断测试、奇偶性、超限和帧错误测试等。位速率 (波特率) 可以通过一个 16 位的波特率选择寄存器进行编程, 因此可以获得超过 65000 种的不同速率。

SCI 模块包括:

1. 两个 I/O 引脚

SCIRXD: SCI 接收数据引脚

SCITXD: SCI 发送数据引脚

2. 波特率可设定到 64K 不同速率

3. 数据字格式

1 个起始位

1~8 位的可编程数据字长度

奇偶校验位 (有/无)

一个或两个停止位

4. 4 个错误检测标志: 奇偶、超载、数据帧、和中断检测

5. 两种唤醒多处理器模式: 空闲线唤醒和位寻址唤醒

6. 半双工或全双工操作

7. 双缓冲的接收和发送功能

8. 发送器和接收器操作可以通过带有状态标志的中断驱动或轮询算法来完成。
9. 发射器：TXRDY 标志（发送器缓冲寄存器已经准备好接收另外字符）和 TX EMPTY 标志位（发射器移位寄存器已空）
10. 接收器：RXRDY 标志（接收器缓冲寄存器已经准备好接收另外字符），BRKDT 标志（间断条件发生）和 RX ERROR 标志（监视 4 个中断条件）
11. 用于发射器和接收器中断的独立使能位（除了 BRKDT）
12. NRZ（非归零）码格式
13. SCI 模块控制寄存器位于开始地址为 7050h 的控制寄存器帧内

注意：该模块中的所有寄存器都是连接到 16 位外设总线的 8 位寄存器。当访问寄存器时，寄存器数据位于低位字节（7-0），高位字节（15-8）被读为零。写入高位字节不起作用。

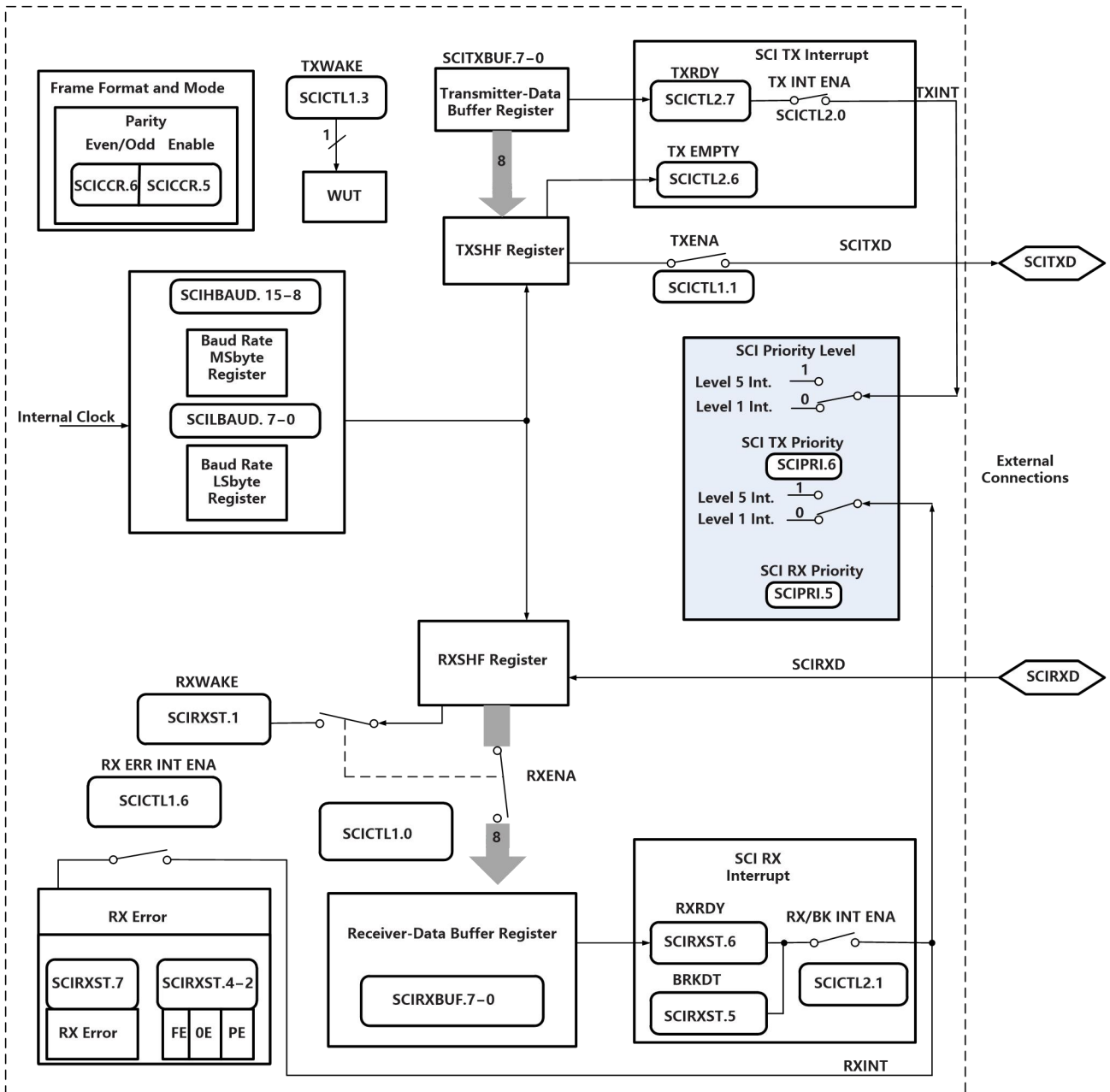


图 12-1 SCI 模块的方框图

12.2 串行通信接口 (SCI) 相关寄存器

1. 串行通信接口 (SCI) 通信控制寄存器 (SCICCR) --7050h

7	6	5	4	3	2	1	0
STOP BITS	EVE/ODD PARITY	PARITY ENABE	LOOP BACK ENA	ADDR/IDLE MODE	SCICCHAR2	SCICCHAR1	SCICCHAR0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

位 7	STOP BITS: SCI 停止位; 0: 一个停止位; 1: 两个停止位;
位 6	EVE/ODD PARITY: SCI 奇/偶校验选择位。如果 PARITY ENABLE 被置位, 则校验才有效, 即判定发送和接收的字符中的 1 的位数为计数或偶数; 0: 奇校验; 1: 偶校验;
位 5	PARITY ENABLE: SCI 奇/偶校验使能位; 0: 禁止奇偶校验; 1: 使能奇偶校验;
位 4	LOOP BACK ENA: 自测试模式使能位; 如果使能该位, 则发送引脚与接收引脚在系统内部连接在一起; 0: 禁止自测试模式; 1: 使能自测试模式;
位 3	ADDR/IDLEMODE: SCI 多处理器模式选择位; 0: 选择空闲多处理器模式; 1: 选择地址位多处理器模式;
位 2~位 0	SCICCHAR2~ SCICCHAR0: SCI 字符长度选择位; 000 1 位 001 2 位 010 3 位 011 4 位 100 5 位 101 6 位 110 7 位 111 8 位

2. 串行通信接口 (SCI) 控制寄存器 1 (SCICTL1) --地址 7051

7	6	5	4	3	2	1	0
保留位	RX ERR INT ENA	SWRESET	保留位	TXW AKE	SLEEP	TXENA	RXENA
R_0	RW_0	RW_0	R_0	RS_0	RW_0	RW_0	RW_0

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

位 7	保留位
位 6	RX ERR INT ENA: SCI 接收错误中断使能位。如果该位被置位, 当接收发生错误时, RX ERROR 位被置 1, 并且发生接收错误中断; 0: 禁止接收错误中断; 1: 使能接收错误中断;
位 5	SW RESET: SCI 软件复位位 (低有效);
位 4	保留位;
位 3	TXWAKE: SCI 发送器唤醒方法选择位。 0: 没有选定的发送特征; 1: 选定的发送特征取决于空闲模式或地址模式;
位 2	SLEEP: SCI 休眠模式位; 0: 禁止休眠模式;

	1: 使能休眠模式;
位 1	TXENA: SCI 发送使能, 仅当 TXENA 置位时, 数据才能从 SCITXD 引脚发送出去。如果复位, 则把 SCITXBUF 寄存器中的数据发送完后才停止发送; 0: 禁止发送; 1: 使能发送;
位 0	RXENA: SCI 接收使能, 从 SCIRXD 引脚上接收到的数据送到接收移位寄存器, 任何再发送到接收缓冲寄存器。该位使能或禁止接收 (传送到缓冲器); 0: 禁止将接收到的数据传送到 SCIRXBUF 和 SCIRXEMU 接收缓冲寄存器; 1: 允许将接收到的数据传送到 SCIRXBUF 和 SCIRXMUN 接收缓冲;

3. 串行通信接口 (SCI) 控制寄存器 2 (SCICTL2) --地址 7054h

7	6	5~2	1	0
TXRDY	TXEMPTY	保留	RX/BKINT ENA	TX INTENA
R_1	R_1	R_0	RW_0	RW_0

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

位 7	TXRDY: 发送缓冲寄存器准备好标志; 0: SCITXBUF 满; 1: SCITXBUF 空, 准备接收下一个字符;
位 6	TX EMPTY: 发送器空标志; 0: SCITXBUF 或者 TXSHF 寄存器或者两者都装入了数据; 1: SCITXBUF 和 TXSHF 寄存器都为空;
位 5~位 2	保留
位 1	RX/BK INT ENA: 接收缓冲器/间断中断使能位。该位控制 RXRDY 或 BRKDT 标志位置位引起的中断请求。但 RX/BK INT ENA 并不阻止这些标志位复位; 0: 禁止 RXRDY/BRKDT 中断; 1: 使能 RXRDY/BRKDT 中断;
位 0	TX INT ENA : 发送缓冲器中断使能位, 该位控制 TXRDY 标志位置位引起的中断请求。但 TX INT ENA 并不阻止 TXRDY 标志位复位; 0: 禁止 TXRDY 中断; 1: 使能 TXRDY 中断;

4. 串行通信接口 (SCI) 接收状态寄存器 (SCIRXST) --7055h

7	6	5	4	3	2	1	0
RXERROR	RXRDY	BRKDT	FE	OE	PE	RXWAKE	保留
R_0	R_0	R_0	R_0	R_0	R_0	R_0	R_0

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

位 7	RX ERROR: SCI 接收器错误标志。 0: 无错误标志被置位; 1: 有错误标志被置位;
位 6	RXRDY: SCI 接收器准备好标志。 0: SCIRXBUF 中无新数据;

	1: SCIRXBUF 中数据已经准备好, 可以读取;
位 5	BRKDT: SCI 中断检测标志; 0: 不满足中断条件; 1: 满足中断条件;
位 4	FE: SCI 错误帧标志位; 0: 未检测到帧错误; 1: 检测到帧错误;
位 3	OE: SCI 超时错误标志; 0: 未检测到超时错误; 1: 检测到超时错误;
位 2	PE: 奇偶校验错误标志; 0: 未检测到奇偶校验错误; 1: 检测到奇偶校验错误;
位 1	RXWAKE: SCI 接收器唤醒检测标志位; 该位为 1 表明检测到接收器唤醒条件, 在多处理器模式, RXWAKE 反映了保存 SCIRXBUF 寄存器中数据的地址位的值。在空闲多处理器模式, 如果检测到 SCIRXD 数据线空闲, 就置位 RXWAKE; RXWAKE 是一个只读标志位, 可通过以下方式之一清除; (1) 在地址字节送至 SCIRXBUF 后传送第一个字节; (2) 读 SCIRXBUF 寄存器; (3) 系统复位; (4) SWRESET 位有效;
位 0	保留

5. 串行通信接口 (SCI) 接收仿真数据缓冲寄存器 (SCIRXEMU) --7056h

7	6	5	4	3	2	1	0
ERXDT7	ERXDT6	ERXDT5	ERXDT4	ERXDT3	ERXDT2	ERXDT1	ERXDT0
R_0	R_0	R_0	R_0	R_0	R_0	R_0	R_0

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

6. 串行通信接口 (SCI) 接收数据缓冲寄存器 (SCIRXBUF) --7057h

7	6	5	4	3	2	1	0
RXDT7	RXDT6	RXDT5	RXDT4	RXDT3	RXDT2	RXDT1	RXDT0
R_0	R_0	R_0	R_0	R_0	R_0	R_0	R_0

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

来自接收引脚的数据从接收移位寄存器 RXSHF 传送到接收缓冲寄存器 SCITXBUF 中, 当传送操作完成后, RXRDY 标志位置 1, 表明接收到的数据已经准备好了, 可以执行读取操作。SCIRXBUF 和 SCIRXEMU 存放相同的数据, 这两个寄存器物理上是一个寄存器, 他的区别仅在于, 两者具有不同的地址, 对 SCIRXEMU 执行读取操作时, 不能清除 RXRDY 标志位, 而读取 SCIRXBUF 操作可以清除 RXRDY 标志;

7. 串行通信接口 (SCI) 发送数据缓冲寄存器 (SCITXBUF) --7059h

7	6	5	4	3	2	1	0
TXD7	TXD6	TXD5	TXD4	TXD3	TXD2	TXD1	TXD0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

8. 串行通信接口 (SCI) 通信优先级控制寄存器 (SCIPRI) --705Fh

7	6	5	4	3	2~0
保留	SCI TX PRIORITY	SCI RX PRIORITY	SCI SOFT	SCI FREE	保留
R_0	RW_0	RW_0	RW_0	RW_0	R_0

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

位 7	保留
位 6	SCI TX PRIORITY: SCI 发送中断优先级选择位; 0: 高优先级中断请求; 1: 低优先级中断请求;
位 5	SCI RX PRIORITY: SCI 接收中断优先级选择位; 0: 高优先级中断请求; 1: 低优先级中断请求;
位 4~位 3	SCI SOFT 和 SCI FREE: SCI 仿真挂起选择位; 00: 一旦仿真挂起, 立即停止; 01: 操作不受仿真挂起影响; 10: 一旦仿真挂起, 在完成当前接收/发送操作后停止; 11: 操作不受仿真挂起影响;
位 2~位 0	保留

9. 串行通信接口 (SCI) 波特率选择高字节寄存器 (SCIHBAUD) --7052h

7	6	5	4	3	2	1	0
BAUD15	BAUD14	BAUD13	BAUD12	BAUD11	BAUD10	BAUD9	BAUD8
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

10. 串行通信接口 (SCI) 波特率选择低字节寄存器 (SCILBAUD) --7053h

7	6	5	4	3	2	1	0
BAUD7	BAUD6	BAUD5	BAUD4	BAUD3	BAUD2	BAUD1	BAUD0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

位 15~位 0	BAUD15~BAUD0: 串行通信接口 16 位波特率选择位。SCIHBAUD (高字节) 和 SCILBAUD (低字节) 连接在一起形成 16 位波特率值。 SCI 异步波特率 = $\text{SYSCLK}/[(\text{BRR} + 1) \times 8]$;
----------	------------------------------------------------------------------------------------------------------------------------------------------------

13 串行外设接口 (SPI) 模块

13.1 串行外设接口 (SPI) 模块简介

ADP16F03 带有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，该设备端口传输的串行数据流的位宽 (1 至 16 位) 是可编程的。通常 SPI 用于 DSP 控制器与外部设备或其他处理器之间的通信。典型应用包括外部 I/O 或通过该设备接口外扩移位寄存器、显示驱动器以及 ADC。SPI 的主/从工作模式支持多设备通信。

SPI 模块的功能包括：

- 四个外部引脚：

SPISOMI：SPI 从机--输出/主机--输入引脚

SPISIMO：SPI 从机--输入/主机--输出引脚

SPISTE : SPI 从机发送使能引脚

SPICLK : SPI 串行时钟引脚

注：如果不使用 SPI 模块时，这四个引脚可以作为 GPIO 口使用。

- 两种运行模式：主模式和从模式
- 波特率：可编程 125 个不同的速率。
- 数据字长：1~16 个数据位
- 四个时钟模式（由时钟极性和时钟相位位控制）包括：
 - 下降沿无相位延迟：SPICLK 高电平有效。SPI 在 SPICLK 信号的下降沿传输数据，并在 SPICLK 信号的上升沿接收数据。
 - 下降沿有相位延迟：SPICLK 高电平有效。SPI 在 SPICLK 信号的下降沿前半个周期传送数据，并在 SPICLK 信号的下降沿接收数据。
 - 上升沿无相位延迟：SPICLK 无效低电平。SPI 在 SPICLK 信号的上升沿传输数据，

并在 SPICLK 信号的下降沿接收数据。

- 上升沿有相位延迟：SPICLK 无效低电平。SPI 在 SPICLK 信号的下降沿之前的一个半个周期传送数据，并在 SPICLK 信号的上升沿接收数据。
- 同时接收和发送操作（在软件中可以禁用发送功能）
- 发射器和接收器操作是通过中断或查询状态标志位的方式来完成。
- 9 个 SPI 模块控制寄存器：位于控制寄存器中，在地址 7040h 开始。

注：这个模块中的所有寄存器都是 16 位寄存器，它们连接到 16 位外围总线。当一个寄存器被访问时，寄存器数据位于下一个字节(7~ 0)，上字节(15~ 8)被读取为 0。对高字节的写入没有影响。

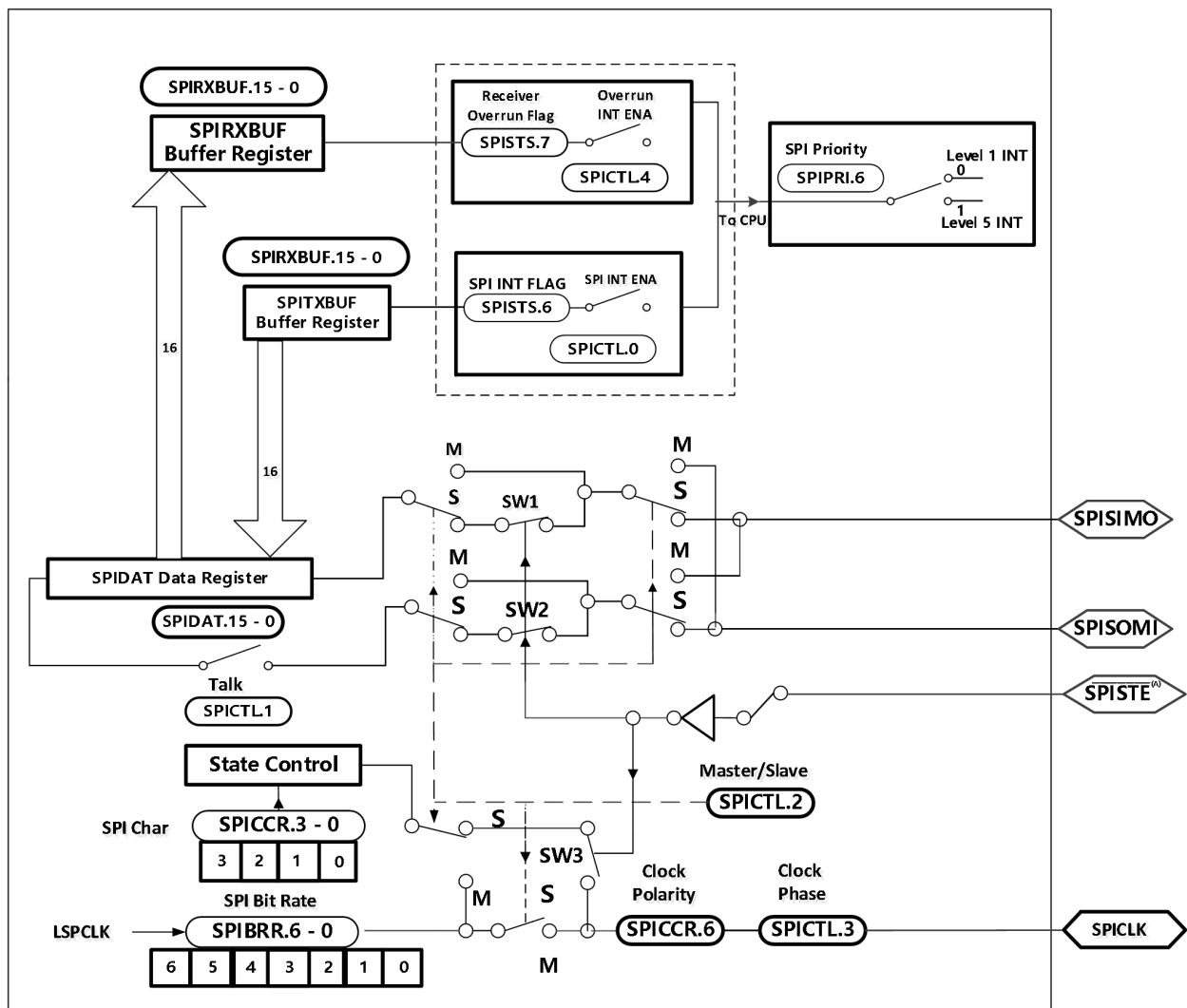


图 13-1 SPI 从模式的工作原理方框图

13.2 串行外设接口 (SPI) 相关寄存器

1. 串行外设接口 (SPI) 配置控制寄存器 (SPICCR) --7040h

7	6	5	4	3	2	1
SPI SW RESET	CLOCK POLARITY	保留	SPI CHAR3	SPI CHAR2	SPI CHAR1	SPI CHAR0
RW_0	RW_0	R_0	RW_0	RW_0	RW_0	RW_0

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

位 7	SPI SW RESET: SPI 软件复位位, 当改变配置时, 用户应在改变前, 把该位清 0, 并在恢复操作前把该位置 1; 0: 初始化 SPI 操作标志为复位标志; 1: SPI 准备发送或者接收字符;
位 6	CLOCK POLARITY: 移位时钟极性选择位; 该位和 SPICTL 寄存器中的 CLOCK PHASE (bit3) 一起控制 SPICLK 引脚上的 4 种时钟设计方式, 具体如表 13-1 所示;
位 5~位 4	保留
位 3~位 0	SPI CHAR3 ~ SPI CHAR0: 字符长度控制位; 0000~1111 : 1~16 字符长度;

2. 串行外设接口 (SPI) 操作控制寄存器 (SPICTL) --7041h

7~5	4	3	2	1	0
保留位	OVERRUN INT ENA	CLOCK PHASE	MASEWR /SLAVE	TALK	SPI INT ENA
R_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

位 7~位 5	保留位
位 4	OVERRUN INT ENA: 接收溢出中断使能位; 0: 禁止 RECEIVER OVER RUN 标志位 (SPISTS.7) 中断; 1: 使能接收溢出中断, 当 RECEIVER OVER RUN 标志位 (SPISTS.7) 被硬件置 1 时可产生一个中断;
位 3	CLOCK PHASE: SPI 时钟相位选择位; 该位和 SPICCR 寄存器中的 CLOCK POLARITY (bit6) 一起控制 SPICLK 引脚上的 4 种时钟设计方式, 具体如表 13-1 所示;
位 2	MASTWR/SLAVE: 主/从模式选择位; 0: SPI 配置为从网络; 1: SPI 配置为主网络;
位 1	TALK: 主/从发送允许位; 0: 禁止发送; 1: 允许发送;
位 0	SPI INT ENA: SPI 中断使能位: 0: 禁止中断; 1: 使能中断;

表 13-1

SPICLK 方式	时钟极性 (SPICCR.6)	相位极性 (SPICTL3)
无延时上升沿	0	0
有延时上升沿	0	1
无延时下降沿	1	0
有延时下降沿	1	1

3. 串行外设接口 (SPI) 状态寄存器 (SPISTS) --7042h

7	6	5	4~0
RECEIVER OVERRUN FLAG	SPI INT FLAG	TX BUF FULL FLAG	保留
RC_0	RC_0	RC_0	R_0

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

位 7	RECEICER OVER FLAG: SPI 接收溢出标志位, 该位是一个只读/清除标志, 当前一个数据从缓冲器中读出之前又完成了下一个数据的接收或发送操作时, 硬件将该位置 1, 表示接收到的最后一个数据被覆盖写入或丢失, 若 OVERRUN INT ENA 位 (SPICTL.4) 已被置 1, 则该位每次置 1 就会发生一次中断请求。向该位写 1, 向 SPI SW RESET 写 0, 系统复位都将清除该位;
位 6	SPI INT FLAG: 中断标志位; 当 SPI 发送或接收完最后一位数据时该位置 1, 同时收到的字符被放置在接收缓冲器中, 如果 SPI INT ENA 位已被置位, 则该标志引起中断请求, 通过读取 SPIRXBUF, 将 1 写入 SPI SW RESET (SPICCR.7), 系统复位都将清除该位; 0: 无中断请求; 1: 有中断请求;
位 5	TX BUF FULL FLAG: SPI 发送缓冲器满标志位; 0: 空; 1: 发送缓冲器有数据;
位 4~位 0	保留;

4. 串行外设接口 (SPI) 波特率设置寄存器 (SPIBRR) --7044h

7	6	5	4	3	2	1	0
保留位	BAUD6	BAUD5	BAUD4	BAUD3	BAUD2	BAUD1	BAUD0
R_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

位 6~位 0	SPIBRR = 0,1,2 时 SPI 波特率 = SYSCLK/4; SPIBRR = 3~127 时 SPI 波特率 = SYSCLK/ (SPIRR+1);
---------	---------------------------------------------------------------------------------------------

5. 串行外设接口 (SPI) 仿真接收缓冲寄存器 (SPIRXEMU) --7046h

15	14	13	12	11	10	9	8
----	----	----	----	----	----	---	---

ERXDT15	ERXDT14	ERXDT13	ERXDT12	ERXDT11	ERXDT10	ERXDT9	ERXDT8
R_0	R_0	R_0	R_0	R_0	R_0	R_0	R_0
7	6	5	4	3	2	1	0
ERXDT7	ERXDT6	ERXDT5	ERXDT4	ERXDT3	ERXDT2	ERXDT1	ERXDT0
R_0	R_0	R_0	R_0	R_0	R_0	R_0	R_0

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

6. 串行外设接口 (SPI) 接收缓冲寄存器 (SPIRXBUF) --7047h

15	14	13	12	11	10	9	8
RXDT15	RXDT14	RXDT13	RXDT12	RXDT11	RXDT10	RXDT9	RXDT8
R_0	R_0	R_0	R_0	R_0	R_0	R_0	R_0
7	6	5	4	3	2	1	0
RXDT7	RXDT6	RXDT5	RXDT4	RXDT3	RXDT2	RXDT1	RXDT0
R_0	R_0	R_0	R_0	R_0	R_0	R_0	R_0

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

SPIRXEMU 寄存器的功能与 SPIRXBUF 的基本相同，区别为读 SPIRXEMU 时不会清除 SPI INT FLAG 标志位，而读 SPIRXBUF 时将清除该标志位。一旦 SPIDAT 接收到完整的数据，就把该数据传送到 SPIRXBUF 和 SPIRXEMU 寄存器。

7. 串行外设接口 (SPI) 发送缓冲寄存器 (SPITXBUF) --7048h

15	14	13	12	11	10	9	8
TXDT15	TXDT14	TXDT13	TXDT12	TXDT11	TXDT10	TXDT9	TXDT8
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
TXDT7	TXDT6	TXDT5	TXDT4	TXDT3	TXDT2	TXDT1	TXDT0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

位 15~位 0	TXDT15~ TXDT0: 下一个要发送的数据
----------	--------------------------

8. 串行外设接口 (SPI) 发送/接收缓冲寄存器 (SPIDATA) --7049h

15	14	13	12	11	10	9	8
SDT15	SDT14	SDT13	SDT12	SDT11	SDT10	SDT9	SDT8
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0

SDT7	SDT6	SDT5	SDT4	SDT3	SDT2	SDT1	SDT0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

SDT15~SDT0，串行数据位，向 SPIDAT 中写入数据有以下两个功能：如果 TALK 位 (SPICTL.1) 被置位，则该寄存器允许将数据输出到串行输出引脚上：当 SPI 为主机时，数据开始发送；

9. 串行外设接口 (SPI) 中断优先级控制寄存器 (SPIPRI) --704Fh

7	6	5	4	3~0
保留位	SPI PRIORITY	SPI SUSP SOFT	SPI SUSP FREE	保留位
R_0	RW_0	RW_0	RW_0	R_0

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

位 7	保留；
位 6	SPI PRIORITY : SPI 中断优先级选择位； 0: 高优先级中断请求； 1: 低优先级中断请求；
位 5~位 4	SPI SUSP SOFT 和 SPI SUSP FREE: SPI 仿真挂起时的操作控制位； 00: 立即停止； 01: 在当前的接收或发送完成之后停止； 1x: SPI 操作不受仿真影响；
位 3~位 0	保留；

14 事件管理器模块 (EM1)

14.1 事件管理器模块 (EM1) 简介

事件管理器模块包括通用(GP) 定时器、完全比较/ PWM 单元、捕获单元、和正交编码器脉冲(QEP) 电路。表 14-1 显示了使用的模块、信号名称、特性、功能, 并且标出了 EM1 命名规则,图 14-1 为事件管理器模块 (EM1) 结构框图。

表 14-1 针对 EM1 模块和信号名称

事件管理器模块	EM1	
	模块	信号
通用定时器	GP 定时器 1	T1CmpareA/T1CmpareB
	GP 定时器 2	T2Cmpare
	GP 定时器 7	PWM7
	GP 定时器 8	PWM8
全功能比较单元	比较器 1	PWM1/2
	比较器 2	PWM3/4
	比较器 3	PWM5/6
捕捉单元	捕捉 1	CAP1
	捕捉 2	CAP2
	捕捉 3	CAP3
QEP	QEP1	QEP1
	QEP2	QEP2
外部输入	计数方向	TDIRA
	外部时钟	TCLKINA

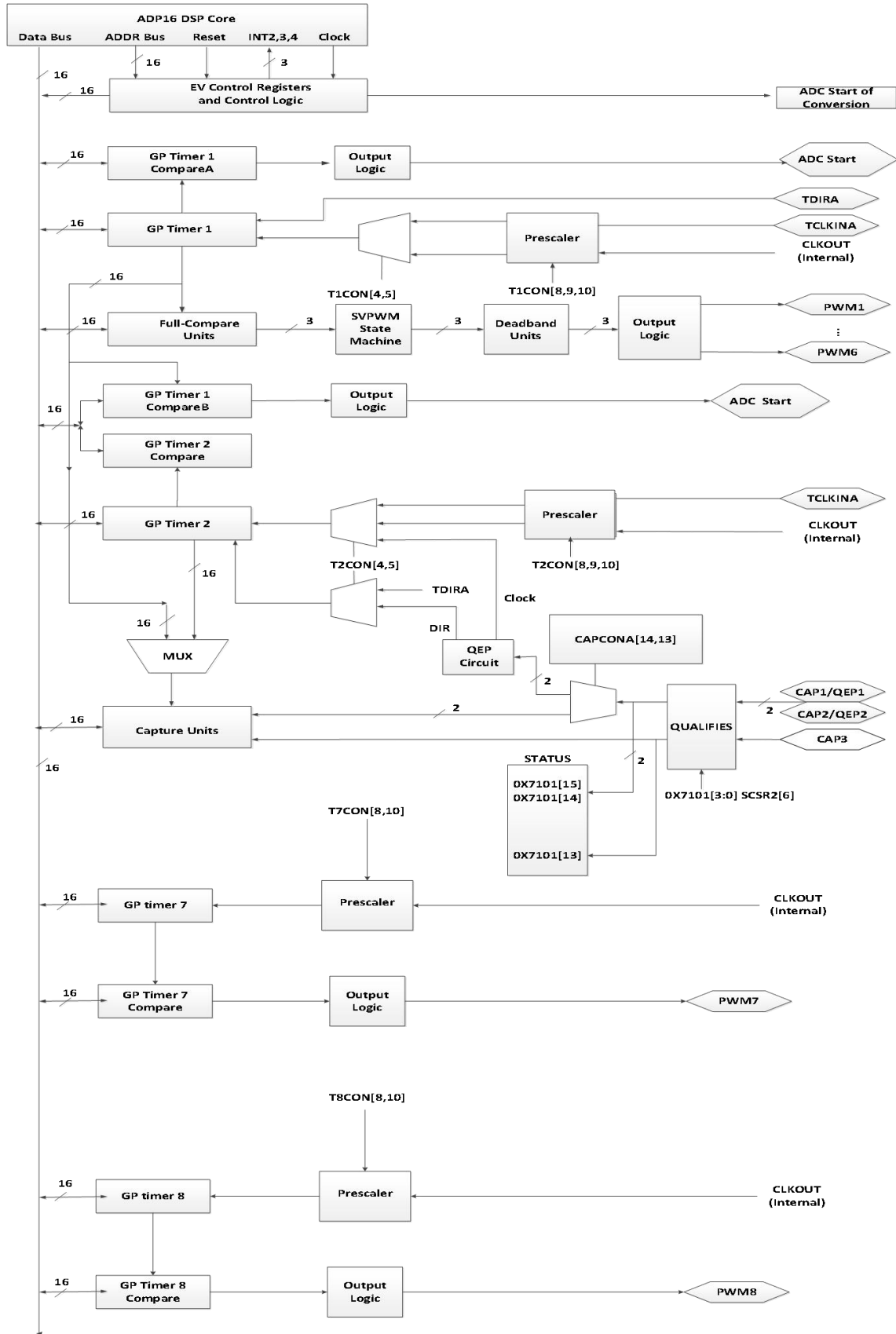


图 14-1 为事件管理器模块 (EM1) 结构框图

14.2 事件管理器模块 (EM1) 中断

14.2.1 事件管理器模块 (EM1) 中断组

事件管理器中断事件分为三组：事件管理器中断组 A,B,C。每组都有各自不同的中断标志，中断使能寄存器和一些外设事件中断请求。在第 8 章中断系统的《表 8-1 ADP16F03 中断源优先级和中断向量表》中列出了 EM1 模块相关的中断；每个 EM1 中断组都有一个中断标志寄存器和相应的中断屏蔽寄存器；

14.2.2 事件管理器模块 (EM1) 中断控制寄存器

1. 事件管理器模块 (EM1) 中断标志寄存器 A (EM1IFRA) --地址 742Fh, 与中断屏蔽寄存器 A

(EM1IMRA) --地址 742Ch;

15~11		10		9		8	
保留		T1OFINT FLAG		T1UFINT FLAG		T1CINT FLAG	
R_0		RW1C_0		RW1C_0		RW1C_0	
7	6~4	3	2	1	0		
T1PINT FLAG	保留	CMP3INT	CMP2INT	CMP1INT	PDPINT		
RW1C_0	R_0	RW1C_0	RW1C_0	RW1C_0	RW1C_0	RW1C_0	

注：R=可读，W=可写，C=清除，_0=复位值，_X=复位值不确定，S=只可被置位，W1C=写 1 清除；

	EM1IFRA	EM1IMRA
位 15~位 11	保留；	保留；
位 10	T1OFINT FLAG：通用定时器 1 上溢中断标志； 读：0 标志被复位； 写：0 无效； 1 标志被置位； 1 复位标志；	通用定时器 1 上溢中断使能位： 0 禁止该中断； 1 使能该中断；
位 9	T1UFINT FLAG：通用定时器 1 下溢中断标志； 读：0 标志被复位； 写：0 无效； 1 标志被置位； 1 复位标志	通用定时器 1 下溢中断使能位： 0 禁止该中断； 1 使能该中断；
位 8	T1CINT FLAG：通用定时器 1 比较中断标志； 读：0 标志被复位； 写：0 无效； 1 标志被置位； 1 复位标志	通用定时器 1 比较中断使能位： 0 禁止该中断； 1 使能该中断；
位 7	T1PINT FLAG：通用定时器 1 周期中断标志； 读：0 标志被复位； 写：0 无效； 1 标志被置位； 1 复位标志	通用定时器 1 周期中断使能位： 0 禁止该中断； 1 使能该中断；
位 6~位 4	保留；	保留；
位 3	CMP3INT：比较单元 3 中断标志； 读：0 标志被复位； 写：0 无效；	比较单元 3 中断使能位： 0 禁止该中断；

	1 标志被置位: 1 复位标志	1 使能该中断;
位 2	CMP2INT: 比较单元 2 中断标志; 读: 0 标志被复位; 写: 0 无效; 1 标志被置位: 1 复位标志	比较单元 2 中断使能位: 0 禁止该中断; 1 使能该中断;
位 1	CMP1INT: 比较单元 1 中断标志; 读: 0 标志被复位; 写: 0 无效; 1 标志被置位: 1 复位标志	比较单元 1 中断使能位: 0 禁止该中断; 1 使能该中断;
位 0	PDPIINT FLAG: 通用定时器 1 保护中断标志; 读: 0 标志被复位; 写: 0 无效; 1 标志被置位: 1 复位标志	通用定时器 1 保护中断使能位: 0 禁止该中断; 1 使能该中断;

2. 事件管理器模块 (EM1) 中断标志寄存器 B (EM1IFRB) --地址 7430h, 与中断屏蔽寄存器 B (EM1IMRB) --地址 742Dh;

15~4	3	2	1	0
保留	T2OFINT FLA	T2UFINT FLAG	T2CINT FLAG	T2PINT FLAG
R_0	RW1C_0	RW1C_0	RW1C_0	RW1C_0

注: R=可读, W=可写, C=清除, _0=复位值, _X=复位值不确定, S=只可被置位, W1C=写 1 清除;

	EM1IFRB	EM1IMRB
位 15~位 4	保留;	保留;
位 3	T2OFINT FLAG: 通用定时器 2 上溢中断标志; 读: 0 标志被复位; 写: 0 无效; 1 标志被置位: 1 复位标志;	通用定时器 2 上溢中断使能: 0 禁止该中断; 1 使能该中断;
位 2	T2UFINT FLAG: 通用定时器 2 下溢中断标志; 读: 0 标志被复位; 写: 0 无效; 1 标志被置位: 1 复位标志;	通用定时器 2 下溢中断使能: 0 禁止该中断; 1 使能该中断;
位 1	T2CINT FLAG: 通用定时器 2 比较中断标志; 读: 0 标志被复位; 写: 0 无效; 1 标志被置位: 1 复位标志;	通用定时器 2 比较中断使能: 0 禁止该中断; 1 使能该中断;
位 0	T2PINT FLAG: 通用定时器 2 周期中断标志; 读: 0 标志被复位; 写: 0 无效; 1 标志被置位: 1 复位标志;	通用定时器 2 周期中断使能: 0 禁止该中断; 1 使能该中断;

3. 事件管理器模块 (EM1) 中断标志寄存器 C (EM1IFRC) --地址 7431 h,与中断屏蔽寄存器 C (EM1IMRC) --地址 742Eh

15~3	2	1	0
保留	CAP1INT FLAG	CAP2INT FLAG	CAP3INT FLAG
R_0	RW1C_0	RW1C_0	RW1C_0

注：R=可读，W=可写，C=清除，_0=复位值，_x=复位值不确定，S=只可被置位，W1C=写 1 清除；

	EM1IFRC	EM1IMRC
位 15~位 3	保留；	保留；
位 2	CAP3INT FLAG: 捕获单元 3 中断标志; 读: 0 标志被复位; 写: 0 无效; 1 标志被置位: 1 复位标志;	捕获单元 3 中断使能位: 0 禁止该中断 1 使能该中断
位 1	CAP2INT FLAG: 捕获单元 2 中断标志; 读: 0 标志被复位; 写: 0 无效; 1 标志被置位: 1 复位标志;	捕获单元 2 中断使能位: 0 禁止该中断 1 使能该中断
位 0	CAP1INT FLAG: 捕获单元 1 中断标志; 读: 0 标志被复位; 写: 0 无效; 1 标志被置位: 1 复位标志;	捕获单元 1 中断使能位: 0 禁止该中断 1 使能该中断

14.3 事件管理器模块 (EM1) 通用定时器

有四个通用定时器。通用定时器 x (x=1, 2, 7, 8) 每个定时器包括:

- 一个 16 位定时增/减计数计数器, TxCNT, 用于读取或者写入, T1CNT—地址 7401h, T2CNT—地址 7405h, T7CNT--T 地址 7533h, T8CNT--地址 7537h;
- 定时器 1 有两个 16 位定时器比较寄存器, T1CMPRA,T2CMPR/T1CMPRB (使用阴影寄存器实现双缓冲), 用于读取或者写入, T1CMPRA—地址 7402h, T2CMPR/T1CMPRB—地址 7406h, 定时器 7 和定时器 8 各一个比较寄存器, T7CMPR--地址 7535h,T8CMPR--地址 7539h;
- 一个 16 位定时器周期寄存器, TxPR (使用阴影寄存器实现双缓冲), 用于读取或者写入, T1PR—地址 7403h, T2PR—地址 7407h, T7PR--地址 7534h, T8PR--7538h;
- 一个 16 位定时器控制寄存器, TxCON, 用于读取或者写入
- 一个针对内部时钟输入的可编程预分频器
- 中断控制逻辑, 用于四个可屏蔽中断: 下溢, 溢出, 定时器比较, 和周期中断。 (**定时器 7, 8 只有上溢中断**)
- 一个可选方向输入引脚(TDIRx) (当定向增/减计数模式被选中时进行增计数或者减计

数)

通用定时器 1 可为捕捉单元、正交脉冲计数、T1CompareA、T1CompareB 和全功能比较器提供时基，T1CompareA 可产生 ADC 采样触发信号 T1A SOC（可选择上升沿或者下降沿触发具体配置见 ADC 部分寄存器），T1CompareB 产生 ADC 采样触发信号 T1B SOC（可选择上升沿或者下降沿触发具体配置见 ADC 部分寄存器），全功能比较器产生 PWM1~PWM6 信号。

通用定时器 2 为捕捉单元、正交脉冲计数和 T2Compare 提供时基。

通用定时器 7 为 T7Compare 提供电路提供时基，产生 PWM7 信号，

通用定时器 8 为 T8Compare 提供电路提供时基，产生 PWM8 信号。

其中 T1CompareA 比较参考值由寄存器 T1CMPRA 提供，T1CompareB 的比较参考值由寄存器 T1CMPRB 提供，T2Compare 的比较参考值由寄存器 T2CMPR 提供，T7Compare 比较参考值由寄存器 T7CMPR 提供，T8Compare 比较参考值由寄存器 T8CMPR 提供。其中寄存器 T2CMPR 与 T1CMPRB 为同一个寄存器。

14.3.1 事件管理器模块 (EM1) 通用定时器控制寄存器

1. 定时器 x 控制控制寄存器 (TxCON:x = 1,2) --地址 7404h (T1CON) ,地址 7408h (T2CON) ;

15	14	13	12	11	10	9	8
FREE	SOFT	保留位	TMODE1	TMODE0	TPS2	TPS1	TPS0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
T2SWT1	TENABLE	TCLKS1	TCLKS0	TCLD1	TCLD0	TECMPR	SEL T1PR
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

注：R=可读，W=可写，C=清除，_0=复位值，_X=复位值不确定，S=只可被置位，W1C=写 1 清除；

位 15~位 14	FREE/SOFT: 仿真控制位; 00: 立即停止 01: 在当前定时器完成计数周期停止 10: 不受仿真影响 11: 不受仿真影响
位 13	保留位;
位 12~位 11	TMOD1/TMOD0: 计数模式选择位;

	00: 停止 01: 连续增/减 10: 连续增计数 11: 连续减计数
位 10~位 8	TPS2~TPS0: 输入时钟预定标系数; 000: x/1 001: x/2 010: x/4 011: x/8 100: x/16 101: x/32 110: x/64 111: x/128 备注: X = CPU 时钟频率;
位 7	T2SWT1: 定时器 2 周期使能位选择位, 在定时器 1 中保留; 0: 定时器 2 选择使用自身定时器使能位 1: 定时器 2 使用 T1CON 的使能, 忽略自身使能位
位 6	TENABLE: 定时器 2 使能位。 0: 禁止定时器运行 1: 使能定时器运行
位 5~位 4	TCLKS1/TCLKS0: 时钟源选择; 00: 内部 CPU 时钟 01: 保留 10: 保留 11: 正交编码脉冲电路一只适合于定时器 2, 在定时器 1 中保留, 这种操作只在 SEL T1PR = 0 时有效
位 3~位 2	TCLD1/TCLD0: 定时器比较寄存器 (如果有效) 的重装载条件; 00: 当计数值是 0 时重载 01: 当计数值是 0 或等于周期寄存器时重载 10: 立即重载 11: 保留
位 1	TECMP: 定时器比较使能; 0: 禁止定时器比较操作 1: 使能定时器比较操作 注: x=2 时, 该位同时为 T1CompareB 和 T2Compare 使能位;
位 0	SEL T1PR: 周期寄存器选择, 在定时器 2 中有效, 定时器 1 中保留; 0: 使用自身定时器周期寄存器 1: 使用 T1PR, 忽略自身周期寄存器

2. 定时器 x 控制控制寄存器 (TxCON:x = 7,8) --地址 7532h (T7CON) ,地址 7536h (T8CON) ;

当上溢事件发生后, 上溢中断标志被置位, 引起定时器 T7 或 T8 溢出中断标志置位, 产生中断请求如果相应的中断没被屏蔽, 则会响应相应的中断。当下溢事件发生时, 下溢标志会被置位。

15	14	13	12	11	10	9	8
OF FLAG	UPINT FLAG	TxSTAT	TMODE1	TMODE2	TPS2	TPS1	TPS0
RW1C_0	RW1C_0	R_1	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
保留	UPIMR	TPIN1	TPIN0	TCLD1	TCLD0	TECMPR	TENABLE
R_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

注: R=可读, W=可写, C=清除, _0=复位值, _X=复位值不确定, S=只可被置位, W1C=写 1 清除;

位 15	OF FLAG: 下溢标志位; 读: 0 标志被复位 写: 0 无效 1 标志被置位 1 复位标志
位 14	UPINT FLAG: 上溢中断标志位; 读: 0 标志被复位 写: 0 无效 1 标志被置位 1 复位标志
位 13	TxSTAT: 计数器状态位 (只读); 0: 向上计数; 1: 向下计数;
位 12~位 11	TMODE1/TMODE0: 计数模式选择; 00: 停止 01: 连续增/减 10: 连续增计数 11: 连续减计数
位 10~位 8	TPS2~TPS0: 输入时钟分频; 000: x/1 001: x/2 010: x/4 011: x/8 100: x/16 101: x/32 110: x/64 111: x/128
位 7	保留
位 6	UPIMR: 上溢中断使能; 0: 禁止上溢中断; 1: 使能上溢中断;
位 5~位 4	TPIN1/TPIN0: PWM7/PWM8 输出极性选择; 00: 强制低 01: 低有效 10: 高有效 11: 强制高
位 3~位 2	TCLD1/TCLD0: 比较寄存器重载条件; 00: 计数器为 0 时重载 01: 计数器为周期值时重载 10: 立即重载 11: 计数器为周期值和 0 时重载
位 1	TECMR: 比较使能; 0: 截止定时器比较操作 1: 使能定时器比较操作
位 0	TENABLE: 定时器使能; 0: 禁止定时器操作 1: 使能定时器操作

3. 通用定时器 1,2 全局控制寄存器 (GPTCONA) --地址 7400h;

全局通用定时器控制寄存器规定了通用定时器 1 和定时器 2 针对不同定时器事件所采取的操作, 并指明了他们的计数方向;

15	14	13	12	11	10	9	8
保留位	T2STAT	T1STAT	保留位	保留位	T2TOADC	T2TOADC	T1TOADC
RW_0	R_1	R_1	RW_0	RW_0	RW_0	RW_0	RW_0

7	6	5	4	3	2	1	0
T1TOADC	TCOMPOE	保留位	保留位	T2PIN	T2PIN	T1PIN	T1PIN
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

注：R=可读，W=可写，C=清除，_0=复位值，_x=复位值不确定，S=只可被置位，W1C=写 1 清除；

位 15	保留位；
位 14	T2STAT: 定时器 2 的状态位, 只读; 0: 减计数; 1: 增计数;
位 13	T2STAT: 定时器 2 的状态位, 只读; 0 : 减计数; 1 : 增计数;
位 12~位 11	保留位;
位 10~位 9	T2TOADC: 定时器 2 触发 A/D 转换事件; 00: 无事件触发 A/D 转换; 01: 下溢中断标志来触发 A/D 转换; 10: 周期中断标志来触发 A/D 转换; 11: T1CompareB 信号触发 A/D 转换;
位 8~位 7	T1TOADC: 定时器 1 触发 A/D 转换事件; 00: 无事件触发 A/D 转换; 01: 下溢中断标志来触发 A/D 转换; 10: 周期中断标志来触发 A/D 转换; 11: T1CompareA 信号触发 A/D 转换;
位 6	TCOMPOE: 比较输出允许; 0: 禁止所有定时器的比较输出; 1: 使能所有定时器的比较输出;
位 5~位 4	保留位;
位 3~位 2	T2PIN: 通用定时器 2 比较输出极性; 00: 强制低; 01: 低有效; 10: 高有效; 11: 强制高;
位 1~位 0	T1PIN: 通用定时器 1 比较输出极性; 00: 强制低; 01: 低有效; 10: 高有效; 11: 强制高;

4. PWM7 和 PWM8 的使能寄存器--地址 753Eh;

15	14	13	12	11	10	9	8
PWM8_EN	PWM7_EN	PWM8_OUT_PAD	保留位	保留位	保留位	PipeEn	WaitModEn
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

7	6	5	4	3	2	1	0
保留位	保留位	保留位	保留位	AutoF	WaitNum		
RW_0	R_1	R_1	RW_0	RW_0	RW_0	RW_0	RW_0

位 15	PWM8_EN: PWM8 使能位; 0: 禁止 PWM8 输出; 1: 使能 PWM8 输出;
位 14	PWM7_EN: PWM7 使能位; 0: 禁止 PWM7 输出; 1: 使能 PWM7 输出;
位 13	PWM8_OUT_PAD: 使能 PWM8 输出到 pad; 0: 禁止 PWM8 输出到 pad; 1: 使能 PWM8 输出到 pad;

5. T7 和 T8 溢出中断控制寄存器—地址 752Fh

T7、T8 的上溢中断标志置位时会使 T7、T8 的溢出中断标志置位，T7、T8 溢出中断置位会引起相应的中断动作。T7、T8 溢出中断指向 INT2。

15~11	10	9	8~0
保留	T8 INTFLAG	T7 INTFLAG	保留
R_0	RW1C_0	RW1C_0	R_0

注：R=可读，W=可写，C=清除，_0=复位值，_X=复位值不确定，S=只可被置位，W1C=写 1 清除；

位 15~位 11	保留
位 10	T8 INTFLAG: T8 溢出中断标志; 读: 0 标志被复位; 写: 0 无效; 1 标志被置位: 1 复位标志;
位 9	T7 INTFLAG: T7 溢出中断标志; 读: 0 标志被复位; 写: 0 无效; 1 标志被置位: 1 复位标志;
位 8~位 0	保留

14.4 事件管理器模块 (EM1) 全功能比较单元

事件管理器上有三个完全比较单元。这些比较单元使用通用定时器作为时基，并且产生六路比较输出，产生的 PWM 波形使用了可编程的死区电路。这六路输出中每一路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。

完全比较单元包括:

- 3 个 16 位的比较寄存器 CMPR3—地址 7419h,CMPR2—地址 7418h,CMPR1—7417h; (它们各自带有一个可读/可写的影子寄存器) ;
- 一个可读/可写的比较控制寄存器,COMCONA;
- 一个 16 位比较方式控制寄存器,ACTRA (带有影子寄存器) ;
- 6 个 PWM (三态) 输出引脚 (PWM1,PWM2,PWM3,PWM4,PWM5,PWM6) ;
- 控制和中断逻辑;

14.4.1 事件管理器模块 (EM1) 全功能比较单元控制寄存器

1. 比较控制寄存器 (COMCONA) --7411h

15	14	13	12	11	10	9	8
CENABLE	CLD1	CLD0	SVENABLE	ACTRLD1	ACTRLD0	FCOMPOE	保留
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	R_0
							7~0
							保留
							R_0

注: R=可读, W=可写, C=清除, _0=复位值, _X=复位值不确定, S=只可被置位, W1C=写 1 清除;

位 15	CENABLE: 比较使能位; 0: 禁止比较操作; 1: 使能比较操作;
位 14~位 3	CLD1/CLD0: 比较寄存器 CMPRx 重载条件; 00: 定时器 1 下溢重载; 01: 定时器 1 下溢或周期重载; 10: 立即; 11: 保留;
位 12	SVENABLE: 空间向量 PWM 模式使能位; 0: 禁止空间向量 PWM 模式; 1: 使能空间向量 PWM 模式;
位 11~位 10	ACTRLD1/ACTRLD0 : 方式控制寄存器重载条件; 00: 定时器 1 下溢; 01: 定时器 1 下溢或周期匹配; 10: 立即; 11: 保留;

位 9	FCOMPOE: 比较输出使能位; 0: PWM 输出引脚为高阻态, 即禁止; 1: PWM 输出引脚未处于高阻态, 即使能;
位 8~位 0	保留

2. 比较方式控制寄存器 (ACTRA) --地址 7413h

15	14	13	12	11	10	9	8
SVRDIR	D2	D1	D0	CMP6ACT1	CMP6ACT0	CMP5ACT1	CMP5ACT0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
CMP4ACT1	CMP4ACT0	CMP3ACT1	CMP3ACT0	CMP2ACT1	CMP2ACT0	CMP1ACT1	CMP1ACT0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

注: R=可读, W=可写, C=清除, _0=复位值, _x=复位值不确定, S=只可被置位, W1C=写 1 清除;

位 15	SVRDIR: 空间矢量 PWM 旋转方向, 仅用于空间矢量 PWM 输出的产生; 0: 正向 (CCW); 1: 反向 (CW);
位 14~位 12	D2~D0: 基本的空间矢量位, 仅用于空间矢量 PWM 输出的产生;
位 11~位 10	CMP6ACT1/0: 引脚 PWM6/IOPB3 上的比较输出方式选择位; 00 : 强制低 01 : 低有效; 10 : 高有效 11 : 强制高;
位 9~位 8	CMP5ACT1/0: 引脚 PWM5/IOPB2 上的比较输出方式选择位; 00 : 强制低 01 : 低有效; 10 : 高有效 11 : 强制高;
位 7~位 6	CMP4ACT1/0: 引脚 PWM4/IOPB1 上的比较输出方式选择位; 00 : 强制低 01 : 低有效; 10 : 高有效 11 : 强制高;
位 5~位 4	CMP3ACT1/0: 引脚 PWM6/IOPB0 上的比较输出方式选择位; 00 : 强制低 01 : 低有效; 10 : 高有效 11 : 强制高;
位 3~位 2	CMP2ACT1/0: 引脚 PWM2/IOPA7 上的比较输出方式选择位; 00 : 强制低 01 : 低有效; 10 : 高有效 11 : 强制高;
位 1~位 0	CMP1ACT1/0: 引脚 PWM1/IOPA6 上的比较输出方式选择位; 00 : 强制低 01 : 低有效; 10 : 高有效 11 : 强制高;

14.5 事件管理器模块 (EM1) 脉冲宽度调制电路

事件管理器 EM1 可同时产生最多 8 路 PWM 波形 (输出)。带有可编程死区的三个完全比较单元组成的三个独立对 (六个输出 PWM1~6)，PWM7/PWM8 单独输出至引脚。

14.5.1 PWM 特性

PWM 特性如下：

- 16 位寄存器
- 每组 PWM 输出的可编程死区计数器范围 0~255
- 最小死区宽度为 25 ns
- 按照需要通过 PWM 频率摆动来改变的 PWM 载波频率
- 可在每个 PWM 周期结束之后改变所需的 PWM 脉冲宽度
- 带有外部可屏蔽的电源和驱动保护中断
- 波形发生器电路，可编程生成不对称、对称以及四空间向量的 PWM 波形
- 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

14.5.2 PWM 相关寄存器

1. 死区定时器控制寄存器 DBTCONA—地址 7415h

15	14	13	12	11	10	9	8
DBT7	DBT6	DBT5	DBT4	DBT3	DBT2	DBT1	DBT0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
EDBT3	EDBT2	EDBT1	DBTPS2	DBTPS1	DBTPS0	保留	保留
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	R_0

注：R=可读，W=可写，C=清除，_0=复位值，_X=复位值不确定，S=只可被置位，W1C=写 1 清除；

位 15~位 8	DBT7~DBT0：死区定时器周期。这 8 位定义了 3 个 8 位死区定时器的周期值；
位 7	EDBT3：死区定时器 3 使能位 (用于比较单元 3 的 PWM5 和 PWM6 引脚)； 0：禁止； 1：使能；
位 6	EDBT2：死区定时器 2 使能位 (用于比较单元 2 的 PWM3 和 PWM4 引脚)；

	0: 禁止; 1: 使能;
位 5	EDBT1: 死区定时器 1 使能位 (用于比较单元 2 的 PWM1 和 PWM2 引脚); 0: 禁止; 1: 使能;
位 4~位 2	DBTPS2~DBTPS0: 死区时间预定标因子; 000 :X/1; 001 X/2; 010 :X/4; 011 X/8; 100 :X/16; 101 X/32; 110 :X/64; 111 X/128;
位 1~位 0	保留;

14.6 事件管理器模块 (EM1) 捕获单元

捕获单元为不同事件或者转换提供一个记录功能。当在捕获输入引脚 CAPx (x = 1, 2, 或者 3) 上检测到指定的电平变化时, 选定的通用定时器计数器的值就会被捕捉并存储在两级深 FIFO 堆栈内。比较单元由三个捕捉电路组成。

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
- 选择通用定时器 1/2 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 个捕获输入可从 CAP1/2/3 引脚引入, 也可通过电压比较器 1/2/3 产生。(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状态在两个设备时钟的上升沿。CAP1/2 也可被用作到 QEP 电路的 QEP 输入)
- 用户指定跳变 (上升边沿、下降边沿、或者这两个边沿) 检测
- 三个可屏蔽中断, 分别对应每个捕捉单元
- CAP1 和 CAP2 共用定时器 1 或 2, CAP3 单独使用定时器 1 或 2;

14.6.1 事件管理器模块 (EM1) 捕获单元控制寄存器

1. 捕获控制寄存器 (CAPCONA) --地址 7420h

15	14~13	12	11	10	9	8	
CAPRES	CAP12EPN	CAP3EN	保留	CAP3SEL	CAP12SEL	CAP3TOADC	
W-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	
7	6	5	4	3	2	1	0
CAP1DEG	CAP1DEG	CAP2DEG	CAP2DEG	CAP3DEG	CAP3DEG	保留	保留
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

注：R=可读，W=可写，C=清除，_0=复位值，_x=复位值不确定，S=只可被置位，W1C=写 1 清除；

位 15	CAPRES：捕获复位，该位只能进行写访问，读访问总位 0。使用捕获功能时，不需要向该位写 1； 0：将所有捕获单元和正交编码电路的寄存器清 0； 1：无操作；
位 14-13	CAP12EPN：表示捕获单元 1 和 2 的控制位； 00：禁止捕获 1 和 2，FIFO 内容不变； 01：使能捕获单元 1 和 2； 10：保留； 11：保留；
位 12	CAP3EN：表示捕获单元 3 的控制位； 0：禁止捕获单元 3，FIFO 内容不变； 1：使能捕获单元 3；
位 11	保留；
位 10	CAP3SEL：捕获单元 3 的通用定时器选择位。 0：通用定时器 2； 1：通用定时器 1；
位 9	CAP12SEL：捕获单元 1,2 的通用定时器选择位。 0：通用定时器 2； 1：通用定时器 1；
位 8	CAP3TOADC：捕获单元 3 事件启动模式数转换位； 0：无操作 1：当 CAP3INT 标志置位时，触发 AD 转换；
位 7~位 6	CAP1DEG：捕获单元 1 的边缘检测位： 00 : 无检测 01 : 检测上升沿； 10 : 检测下降沿 11 : 检测两个边沿；
位 5~位 4	CAP2DEG：捕获单元 2 的边缘检测位： 00 : 无检测 01 : 检测上升沿； 10 : 检测下降沿 11 : 检测两个边沿；
位 3~位 2	CAP3DEG：捕获单元 3 的边缘检测位： 00 : 无检测 01 : 检测上升沿； 10 : 检测下降沿 11 : 检测两个边沿；
位 1~位 0	保留；

2. 捕获 FIFO 状态控制寄存器 (CAPFIFO) --地址 7422h

15~14	13~12	11~10	9~8	7~0
保留	CAP3FIFO	CAP2FIFO	CAP1FIFO	保留
R_0	RW_0	RW_0	RW_0	R_0

注：R=可读，W=可写，C=清除，_0=复位值，_x=复位值不确定，S=只可被置位，W1C=写 1 清除；

位 15~位 14	保留；
位 13~位 12	表示捕获单元 3 的 FIFO 状态位； 00：空； 01：已有一个值压入堆栈； 10：已有两个值压入堆栈； 11：已有两个，丢弃先压入的值；
位 11~位 10	表示捕获单元 2 的 FIFO 状态位； 00：空； 01：已有一个值压入堆栈； 10：已有两个值压入堆栈； 11：已有两个，丢弃先压入的值；
D9~D8	表示捕获单元 1 的 FIFO 状态位； 00：空； 01：已有一个值压入堆栈； 10：已有两个值压入堆栈； 11：已有两个，丢弃先压入的值；
位 7~位 0	保留；

3. 六个 FIFO 寄存器

名称	地址	说明
CAP1FIFO	7423h	CAP1 的 FIFO 堆栈顶层寄存器
CAP2FIFO	7424h	CAP2 的 FIFO 堆栈顶层寄存器
CAP3FIFO	7425h	CAP3 的 FIFO 堆栈顶层寄存器
CAP1FBOT	7427h	CAP1 的 FIFO 堆栈底层寄存器
CAP2FBOT	7428h	CAP2 的 FIFO 堆栈底层寄存器
CAP3FBOT	7429h	CAP3 的 FIFO 堆栈底层寄存器

14.6 正交编码脉冲(QEP) 电路

两个捕获输入引脚 CAP1 和 CAP2；可被设置成片内 QEP 电路，并且与正交编码脉冲的对接。为了保证这些输入信号芯片的运行完全同步，当正交脉冲序列的方向或者边沿被检测到时，通用定时器 2 随着两个输入信号（输入脉冲的四倍频）的上升和下降沿递增或递减。

15 模数转换器 (ADC)

15.1 ADC 的特性

- 具有内置采样/保持 (S/H) 的 12 位模拟 AD 核 (ADC)。
- 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)。
- 转换速率: 14MHz ADC 采样时钟, 1 百万次采样每秒(MSPS), 转换速率为 1000ns。
- 16 通道被复用输入 (其中 A3 通道内置用于采样温度传感器)
- 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。
- 序列发生器可运行为 2 个独立的 8 态序列发生器 (SEQ1 和 SEQ2), 或作为 1 个较大的 16 态序列发生器使用 (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 (RESULT0~RESULT15 可分别寻址)。

– 输入模拟电压对应的数字值由以下公式得出:

$$\begin{aligned}
 \text{Digital Value} &= 0 && \text{when input} \leq 0 \text{ V} \\
 \text{Digital Value} &= 4096 \times \frac{\text{Input Analog Voltage} - LO}{3} && \text{when } 0 \text{ V} < \text{input} < 3 \text{ V} \\
 \text{Digital Value} &= 4095 && \text{when input} \geq 3 \text{ V}
 \end{aligned}$$

A 所有分数值均截断(小数位都舍去, 只取整数位)。

- 转换开始(SOC) 序列源的多个触发器:
 - S/W——软件立即触发
 - EM1——事件管理器 1 (EM1 内的多个事件源)
 - 外部引脚——ADCSOC
- 灵活的中断控制允许每个序列结束(EOS) 或每隔一个 EOS 上的中断请求。
- 序排序器可运行在“启动/停止”模式, 允许多个按时间排序的触发源使转换同步。

- EM1 的 T1 相关联的两个 SOC 事件源可独立运行在双序列发生器模式。
- ADC 直流失调带出厂校准。
- ADC 增益误差带出厂校准。

ADC 模块具有在 14MHz 的 ADC 时钟上有一个 1000ns 快速转换速度，有 16 个输入通道，分 A0~A7 和 B0~B7，这些通道可被配置为用于事件管理器 1 的独立 8 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。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 选择其中一个可用的 8 通道。在每个序列发生器上，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，以使用户执行采样算法。这种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。

15.2 ADC 模块概述

如图 15-2-1 所示，ADP16F03 的 ADC 系统包括 12 位主 ADC，通道选通开关以及排序器 SEQ1 和 SEQ2。

其中排序器 SEQ1 和 SEQ2 是 ADC 系统的主要控制单元，负责切换不同的通道将数据送入模拟 AD 核进行转化，结果 MUX 负责切换不同的结果寄存器将 ADC 转换的数字码送入对应的结果寄存器当中。ADC A3 通道可切换为内部温度传感器采样通道。

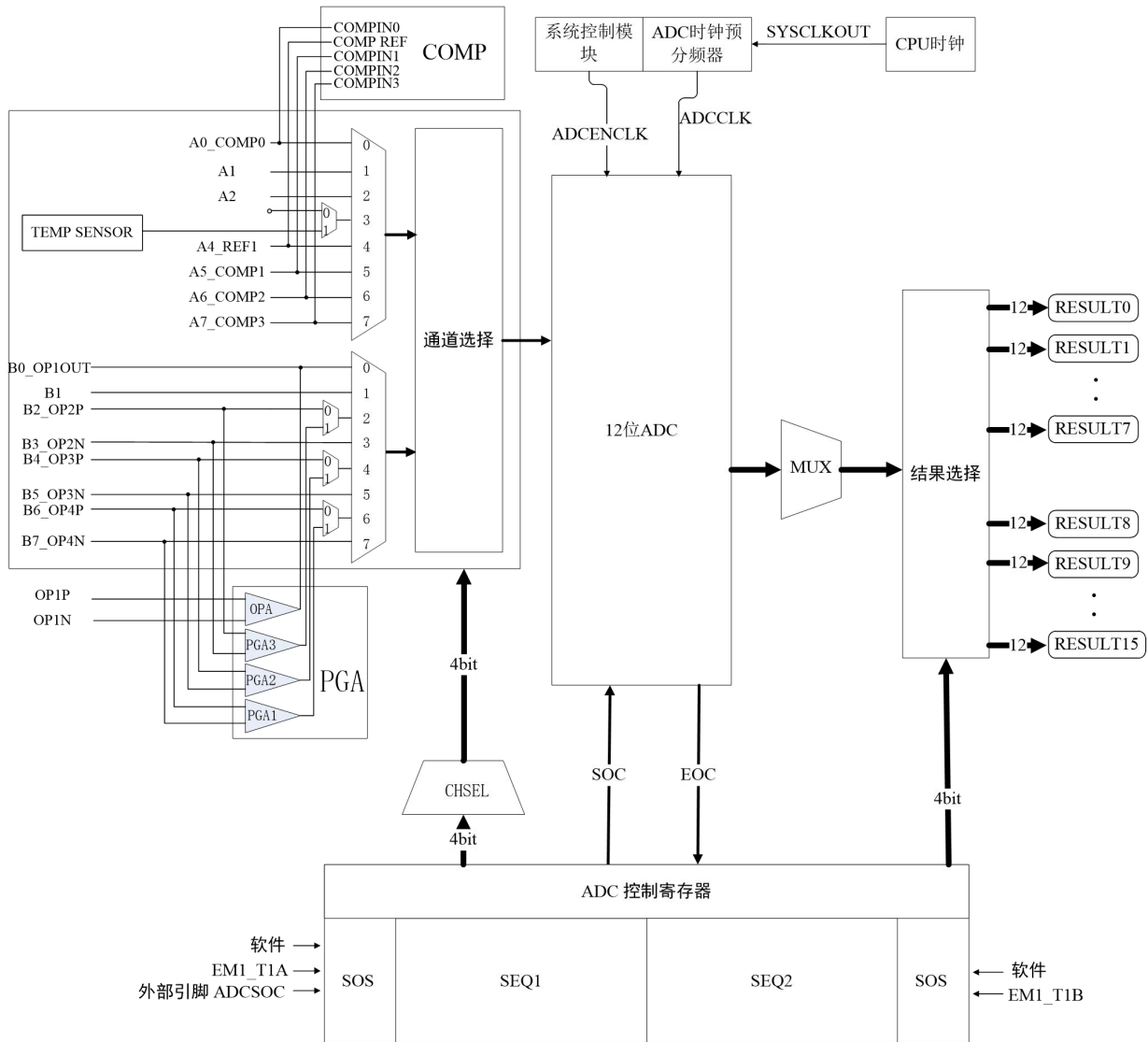


图 15-2-1 ADC 系统结构图

建议在所有电源引脚上使用外部去耦合电容器。

图 15-2-2 为 ADC 电源的引脚连线图，所示要获得指定的 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 引脚的走线不应与数字信号通道靠得太近。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 VCCA 与数字电源隔离。为了获得更佳精度和 ESD 保护，未使用的 ADC 输入引脚应该连接模拟地。

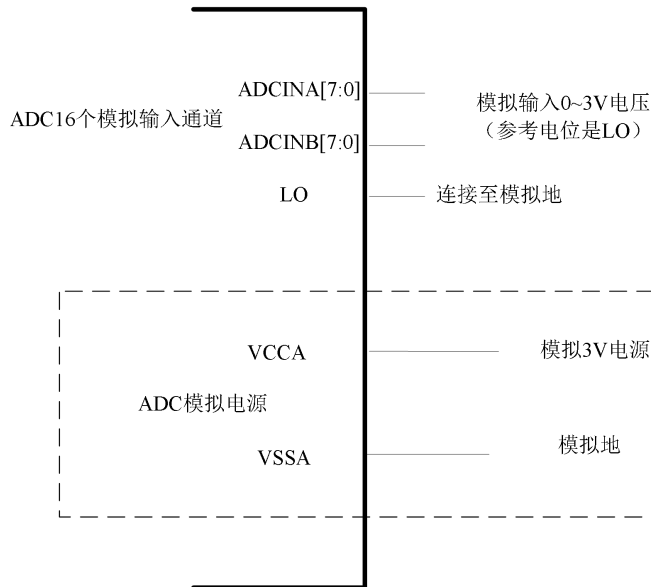


图 15-2-2 ADC 电源的引脚连线

15.3 自动排序器

图 15-3-1 是双排序模式下自动排序的 ADC 结构图，图 15-3-2 是级联模式下自动排序的 ADC 结构图。ADC 排序器由 2 个独立的 8 状态排序器 (SEQ1 和 SEQ2) 组成，也可以级联成一个 16 状态序列发生器 (SEQ)。在这种模式下，ADC 能对一系列转换进行自动定序，即每当 ADC 接收到转换开始请求时，它可以自动执行多次转换。对于每次转换，可通过模拟 MUX 选择 16 个可用输入通道中的任何一个。转换之后，所选通道的数值将存储在适当的结果寄存器 (ADCRESULTn) 中。(第一个结果存储在 RESULT0/RESULT8 中，第二个结果存储在 RESULT1/RESULT9 中，依此类推)。还可以对同一通道多次采样，以使用户执行“过采样”，从而提供比传统的单采样转换结果更高的分辨率。

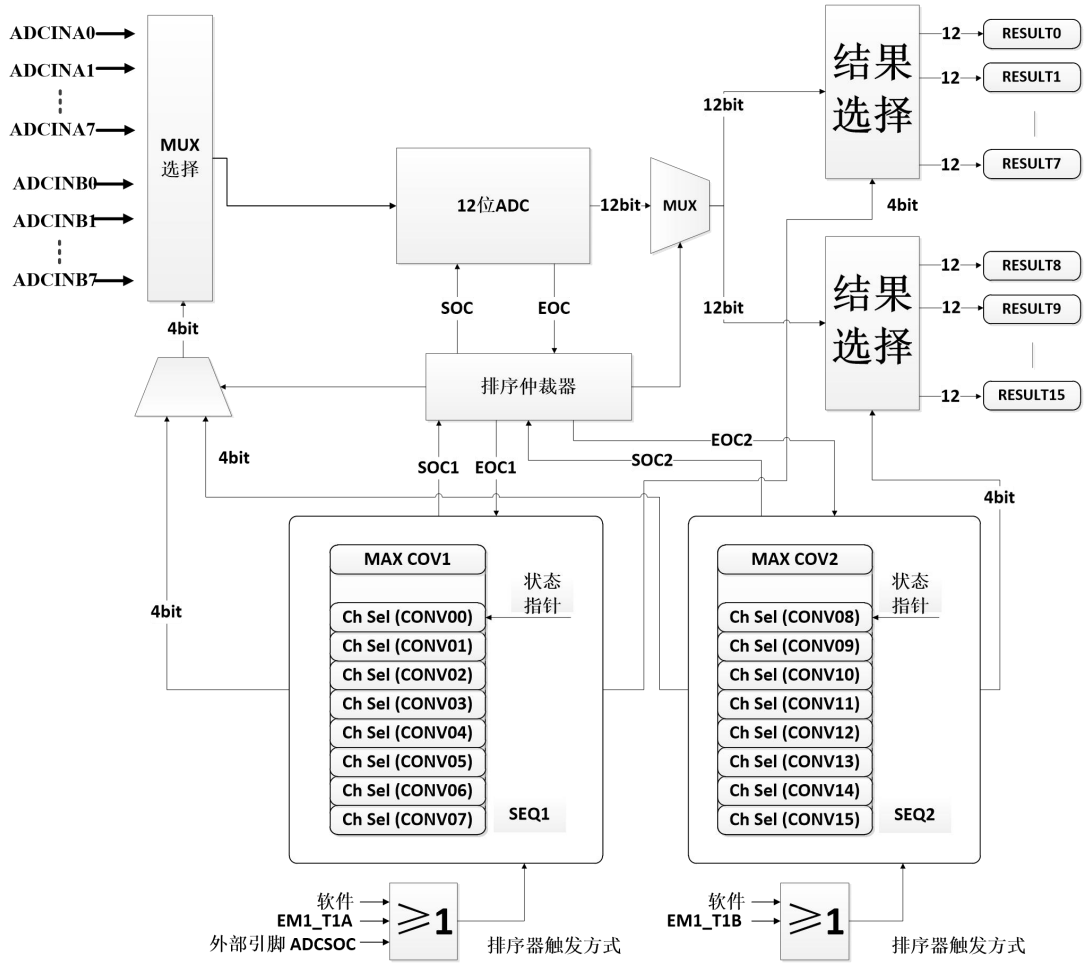


图 15-3-1 双排序模式下自动排序的 ADC 结构图

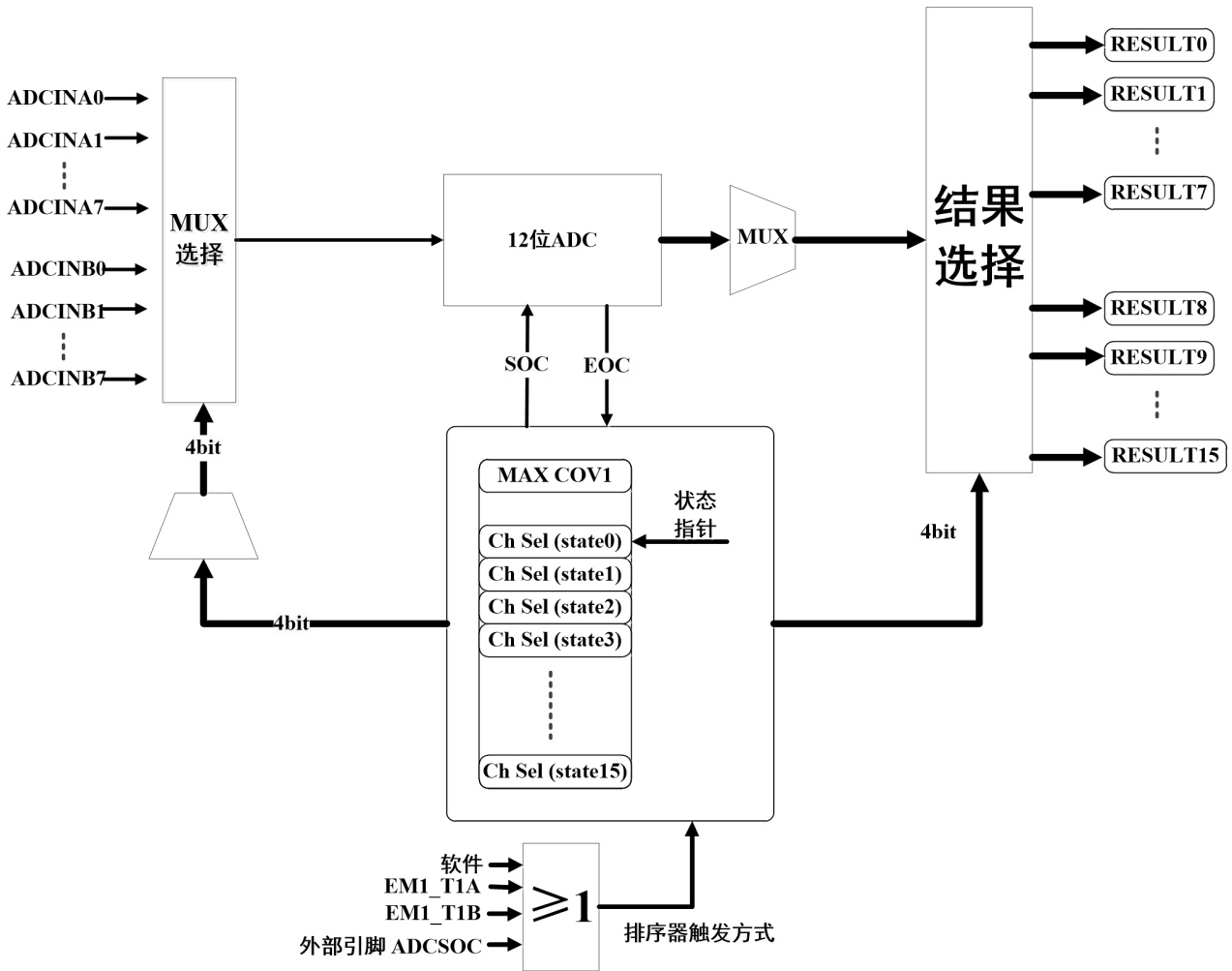


图 15-3-2 级联模式下自动排序的 ADC 结构图

注:

在顺序采样的双排序器模式中，一旦完成当前活动排序器启动的序列，则将执行暂挂的来自其它排序器的 SOC 请求。例如，假定出现来自 SEQ1 的 SOC 请求时，A/D 转换器正忙于处理 SEQ2 的请求。A/D 转换器将在完成正在处理的 SEQ2 请求之后，立即开始执行 SEQ1 的请求。如果 SEQ1 和 SEQ2 的 SOC 请求都为暂挂状态，则 SEQ1 的 SOC 具有优先权。例如，假定 A/D 转换器正忙于处理 SEQ1 的请求。在此过程中，同时出现了来自 SEQ1 和 SEQ2 的 SOC 请求。当 SEQ1 完成其活动序列时，将立即执行 SEQ1 的 SOC 请求，SEQ2 的 SOC 请求继续保持暂挂状态。

ADC 可以运行于顺序采样模式。对于每次转换，当前的 CONVxx 位字段定义了将要采样和转换的通

道。在顺序采样模式中，CONVxx 的所有 4 位用于定义输入通道。MSB 用于定义与输入通道相关联的采样保持缓冲器，三个 LSB 用于定义偏移。例如，如果 CONVxx 包含值 0101b，则 ADCINA5 为选定的输入通道。如果它包含值 1011b，则 ADCINB5 为选定的输入通道。

15.3.1 顺序采样模式

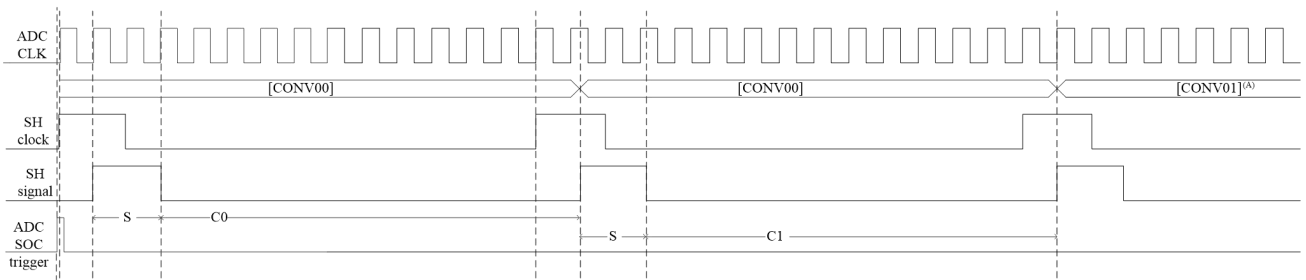


图 15-3-3 示了顺序采样模式的时序

注：

S——ADC 的真正采样窗口（宽度最少为 2 个 ADCCLK 周期）

C1——ADC 转换数据时间长度（结果寄存器刷新数据所需时长）

ADC 通道地址包含在 [CONV00] 4 位寄存器中；对 SEQ1 为 CONV00，而对 SEQ2 为 CONV08。

表15-3-1 单一工作模式SEQ1和SEQ2比较

特性	单一 8 状态 排序器 #1(SEQ1)	单一 8 状态 排序器 #2 (SEQ2)	级联 16 状态 序列发生器 (SEQ)
转换开始 (SOC) 触发方式	软件、EM1_T1A、外部引脚	软件、EM1_T1B	软件、EM1_T1A、EM1_T1B、外部引脚
最大自动转换数 (序列长度)	8	8	16
仲裁优先级	高	低	不适用
ADC 转换结果寄存器位置	0~7	8~15	0 - 15
CHSELSEQn 位字段指定	CONV00~CONV07	CONV08~CONV15	CONV00 - CONV15

为方便起见，此后将排序器状态称为：

- 对 SEQ1： CONV00 - CONV07
- 对 SEQ2： CONV08 - CONV15
- 对级联 SEQ： CONV00 - CONV15

在 ADC 输入通道选择定序控制寄存器 (CHSELSEQn) 中的 CONVxx 位字段定义了为每个定序的转换选择的模拟输入通道。CONVxx 是一个 4 位字段, 它指定 16 个用于转换的通道中的任一个。由于在使用两个排序器先后一起工作时, 序列中最多可有 16 次转换, 因此提供了 16 个此类 4 位字段(CONV00-CONV15), 且分布在 4 个 16 位寄存器 (ADCCHSELSEQ1-ADCCHSELSEQ4) 中。CONVxx 位可以是 0-15 的任何值。可按任何所需的顺序选择模拟通道, 并可多次选择同一通道。

示例 1. 顺序采样双排序器模式示例

示例初始化:

```
AdcRegs.ADCCTRL3.bit.SMODE_SEL = 0x0102; // 设置顺序采样、双排序模式, 采样窗口 2ADCCLK 周期, ADCCLK 为 CPU 时钟的 1/4
```

```
AdcRegs.MAXCONV.all = 0x0077; // 每个排序器 8 个转换 (共 16 个)
```

```
AdcRegs.CHSELSEQ1.bit.CONV00 = 0x0; // 设置从 ADCINA0 转换
```

```
AdcRegs.CHSELSEQ1.bit.CONV01 = 0x1; // 设置从 ADCINA1 转换
```

```
AdcRegs.CHSELSEQ1.bit.CONV02 = 0x2; // 设置从 ADCINA2 转换
```

```
AdcRegs.CHSELSEQ1.bit.CONV03 = 0x3; // 设置从 ADCINA3 转换
```

```
AdcRegs.CHSELSEQ2.bit.CONV04 = 0x4; // 设置从 ADCINA4 转换
```

```
AdcRegs.CHSELSEQ2.bit.CONV05 = 0x5; // 设置从 ADCINA5 转换
```

```
AdcRegs.CHSELSEQ2.bit.CONV06 = 0x6; // 设置从 ADCINA6 转换
```

```
AdcRegs.CHSELSEQ2.bit.CONV07 = 0x7; // 设置从 ADCINA7 转换
```

```
AdcRegs.CHSELSEQ3.bit.CONV08 = 0x8; // 设置从 ADCINB0 转换
```

```
AdcRegs.CHSELSEQ3.bit.CONV09 = 0x9; // 设置从 ADCINB1 转换
```

```
AdcRegs.CHSELSEQ3.bit.CONV10 = 0xA; // 设置从 ADCINB2 转换
```

```
AdcRegs.CHSELSEQ3.bit.CONV11 = 0xB; // 设置从 ADCINB3 转换
```

```
AdcRegs.CHSELSEQ4.bit.CONV12 = 0xC; // 设置从 ADCINB4 转换
```

```
AdcRegs.CHSELSEQ4.bit.CONV13 = 0xD; // 设置从 ADCINB5 转换
```

```
AdcRegs.CHSELSEQ4.bit.CONV14 = 0xE; // 设置从 ADCINB6 转换
```

```
AdcRegs.CHSELSEQ4.bit.CONV15 = 0xF; // 设置从 ADCINB7 转换
```

如果已执行 SEQ1 和 SEQ2 (先后两条语句(间隔 2 个系统周期)将 ADCCTRL2 的 Bit13 和 Bit5 置为“1”), 则结果将存入以下 RESULT 寄存器中:

```
ADCINA0 -> RESULT0
```

```
ADCINA1 -> RESULT1
```

```
ADCINA2 -> RESULT2
```

```
ADCINA3 -> RESULT3
```

```
ADCINA4 -> RESULT4
```

```
ADCINA5 -> RESULT5
```

```
ADCINA6 -> RESULT6
```

```
ADCINA7 -> RESULT7
```

```
ADCINB0 -> RESULT8
```

```
ADCINB1 -> RESULT9
ADCINB2 -> RESULT10
ADCINB3 -> RESULT11
ADCINB4 -> RESULT12
ADCINB5 -> RESULT13
ADCINB6 -> RESULT14
ADCINB7 -> RESULT15
```

15.3.2 不中断的自动排序模式

以下说明适用于 8 状态排序器 (SEQ1 或 SEQ2)。在此模式中, SEQ1/SEQ2 可在单次定序会话中对任何通道自动定序多达 8 次转换 (当级联模式或排序器先后一起使用时为 16 次)。图 15-3-4 显示了流程图。每次转换的结果存储在 8 个结果寄存器的其中一个 (对 SEQ1 为 ADCRESULT0 - ADCRESULT7, 对 SEQ2 为 ADCRESULT8 -ADCRESULT15) 中。从最低地址向最高地址填充这些寄存器。

序列中的转换数由 MAX_CONVn (ADCMAXCONV 寄存器中的 3 位字段或 4 位字段) 控制, 并在自动定序转换会话开始时自动载入自动定序状态寄存器 (ADCASEQSR) 中的定序计数器状态位 (SEQ_CNTR[3:0]) 中。MAX_CONVn 字段可为 0 - 7 范围内的值。当排序器从状态 CONV00 开始时, SEQ_CNTR 位从其载入的值开始进行倒计时, 并按顺序持续 (CONV01、CONV02...依此类推) 到 SEQ_CNTR 变为 0 为止。在自动定序会话期间完成的转换数等于 (MAX_CONVn + 1)。

示例 2. 在双排序器模式下使用 SEQ1 进行转换

假设需从 SEQ1 进行 7 次转换 (即作为自动定序会话的一部分, 必须转换输入 ADCINA2 和 ADCINA3 各两次, 接着转换 ADCINA6、ADCINA7 和 ADCINB4), 则应将 MAX_CONV1 设置为 6, ADCCHSELSEQn 寄存器设置为如表 15-3-2。

表 15-3-2 例 2 中 CHSELSEQn 寄存器填入值

	Bits 15~12	Bits 11~8	Bits 7~4	Bits 3~0	
70A3h	3	2	3	2	CHSELSEQ1
70A4h	x	12	7	6	CHSELSEQ2
70A5h	x	x	x	x	CHSELSEQ3
70A6h	x	x	x	x	CHSELSEQ4

注: 表中数值为十进制数, x 是不用考虑的数。

如图 15-3-4 为 ADC 在不中断的自动排序模式下的转换流程图。

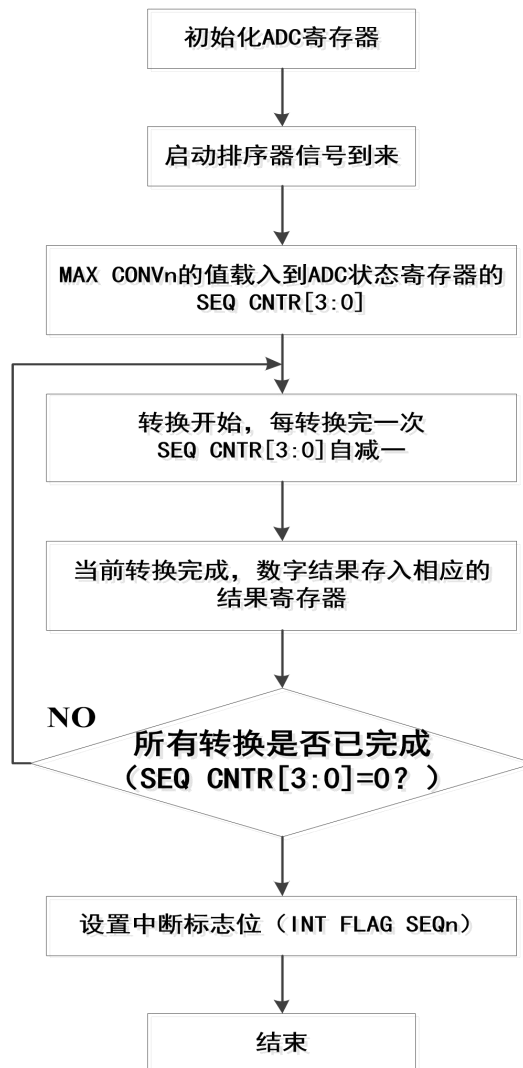


图 15-3-4 不中断的自动排序模式 ADC 转换流程图

一旦排序器接收到转换开始 (SOC) 触发信号，将开始转换。SOC 触发器也载入 SEQ_CNTR 位。将按预定顺序对 CHSELSEQn 寄存器中指定的通道进行转换。每次转换后，SEQ_CNTR 位将自动减 1。一旦 SEQ_CNTR 到 0，可能发生两类事情，这取决于 ADCTRL1 寄存器中连续运行位 (CONT_RUN) 的状态。请参阅图 15-3-4 了解流程说明。

如果设置了 CONT_RUN，转换序列将自动再次启动（即 SEQ_CNTR 重载 MAX_CONV1 的原始值且 SEQ1 设置为 CONV00）。在这种情况下，为避免覆盖数据，

必须确保在下一转换序列开始之前已读取结果寄存器。ADC 中设计了仲裁逻辑，以确保出现争用时不会破坏结果寄存器（在您尝试读取结果寄存器的同时 ADC 模块正在尝试写入这些结果寄存器）。

如果未设置 CONT_RUN，则排序器保持最后的状态（本例中为 CONV06），且 SEQ_CNTR 继续保持为 0 值。要在下一 SOC 时重复序列，必须在下一 SOC 之前使用 RST_SEQn 位复位排序器。

如果每次 SEQ_CNTR 到达 0 时设置中断标志 (INT_ENA_SEQn (Mode1) = 0 且 INT_ENA_SEQn (Mode0) = 1)，（如有需要）可以在中断服务例程 (ISR) 中手动复位排序器 (使用 ADCTRL2 寄存器中的 RST_SEQn 位)。这将使得 SEQn 状态复位成初始值 (对 SEQ1 为 CONV00，对 SEQ2 为 CONV08)。此功能在排序器的“启动/停止”操作中非常有用。

15.3.3 排序器的启动/停止模式

除了不间断自动定序模式之外，任何一个排序器 (SEQ1、SEQ2) 均可工作在停止/启动模式，在该方式下，可实现和多个转换开始(SOC) 触发器时间上同步。此模式类似于示例 2，但是排序器完成第一个转换序列后，允许重新触发而不复位为初始状态 CONV00（即在中断服务例程内不复位该排序器）。因此，当一个转换序列结束时，排序器保持在当前转换状态。ADCTRL1 寄存器中的连续运行位(CONT_RUN) 必须设置为 0（即禁用）。

示例 3.排序器的启动/停止操作

要求：使用触发器 1（下溢）启动 3 次自动转换（如 I1、I2 和 I3），使用触发器 2（定时周期）启动 3 次自动转换（如 V1、V2 和 V3）。触发信号 1 与触发信号 2 在时间上相差 25μs，并由事件管理器 EM1 的 T1A 和 T1B 提供。请参阅图 15-3-5。在这一事例中，仅使用了 SEQ1。

注：

触发信号 1 和触发信号 2 可以是来自 EM1 (EM1_T1A)、外部引脚或软件的 SOC 信号。相同的触发源可以发生 2 次，以满足本例中的双触发要求。必须注意不要因正在处理的序列而丢失多个 EM1 触发信号。请参阅第 15.3.4 部分。

此处将 MAX_CONV1 的值设置为 2，并将 ADC 输入通道选择定序控制寄存器 (CHSELSEQn) 设置为 1，如表 15-3-3 所示。

复位和初始化之后，SEQ1 将等待第一个触发信号。出现第一个触发信号时，执行通道选择值为 CONV00(I1)、CONV01 (I2)

和 CONV02 (I3) 的 3 次转换。然后, SEQ1 在当前状态等待下一个触发信号。25 微秒之后, 第二个触发信号到达, 将执行通道选择值为 CONV03 (V1)、CONV04 (V2) 和 CONV05 (V3) 的另外 3 次转换。

对于这两个触发情况, MAX_CONV1 的值都自动载入到 SEQ_CNTR1 中。如果在第二个触发信号到来时, 需要改变为不同的转换数, 则必须通过软件 (在第二次触发前的某个适当时间) 更改 MAX_CONV1 的值, 否则, 将重新使用当前 (最初载入) 的值。这可通过在适当的时间由中断服务程序 (ISR) 更改 MAX_CONV1 值来实现。

第二次自动转换会话结束时, ADC 结果寄存器的值如表 15-3-4 所示。

这时, SEQ1 保持在当前状态等待另一触发信号。现在, 用户可将 SEQ1 (通过软件) 复位成状态 CONV00, 并可以重复相同的触发信号 1 和触发信号 2 会话。

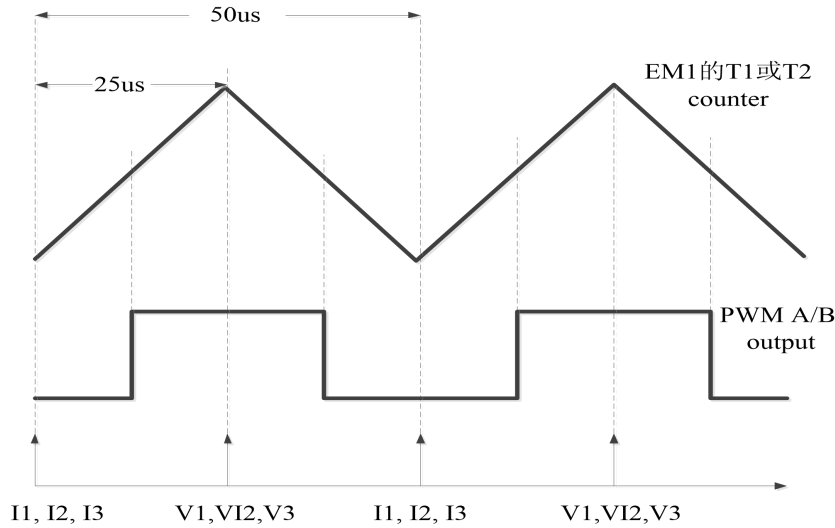


图 15-3-5 EM1_T1A/EM1_T1B 触发启动排序器的示例

表 15-3-3 CHSELSEQn 的值 (MAX_CONV1 设置为 2)

	Bits 15~12	Bits 11~8	Bits 7~4	Bits 3~0	
70A3h	V1	I3	I2	I1	CHSELSEQ1
70A4h	x	x	V3	V2	CHSELSEQ2
70A5h	x	x	x	x	CHSELSEQ3
70A6h	x	x	x	x	CHSELSEQ4

表 15-3-4 第二次自动转换会话后的值

结果寄存器	结果寄存器数据
RESULT0	I1
RESULT1	I2
RESULT2	I3
RESULT3	V1
RESULT4	V2
RESULT5	V3
RESULT6	x
RESULT7	x
RESULT8	x

RESULT9	X
RESULT10	X
RESULT11	X
RESULT12	X
RESULT13	X
RESULT14	X
RESULT15	X

15.3.4 输入触发源描述

每个排序器具有一组可以启用/禁用的触发源输入。请参阅表 15-3-5 以了解 SEQ1、SEQ2 的有效输入触发源。

表 15-3-5 输入触发源

SEQ1 (排序器 1)	SEQ2 (排序器 2)	级联 SEQ
软件触发 (软件 SOC)	软件触发 (软件 SOC)	软件触发器 (软件 SOC)
EM1_T1A (T1A SOC)	EM1_T1B (T1B SOC)	EM1_T1B (T1B SOC)、EM1_T1A (T1A SOC)
外部 SOC 引脚 (ADC SOC)		外部 SOC 引脚 (ADC SOC)

注:

- (1) SOC 触发源可在排序器处于空闲状态时启动自动转换序列。空闲状态可以是接收触发信号之前的 CONV00，也可以是转换序列完成时 (即 SEQ_CNTR 到计数 0 时) 排序器所处的状态。
- (2) 如果在执行当前转换序列时出现 SOC 触发信号，则它将设置 ADCTRL2 寄存器中的 SOC_SEQn 位为 1 (此位在上一转换序列开始时已被清除)。如果还出现另一 SOC 触发信号，则将丢失它 (即在已设置 SOC_SEQn 位 (SOC 暂挂)，将忽略随后的触发信号)。
- (3) 一旦触发成功，排序器就不能在序列中间停止或中断。程序必须等待序列结束 (EOS) 或启动排序器复位，这将使排序器立即返回到空闲起始状态 (对 SEQ1 为 CONV00; 对 SEQ2 为 CONV08)。
- (4) 当 SEQ1/2 用于级联模式时，将忽略进入 SEQ2 的触发信号，而 SEQ1 的触发信号为活动状态。可将级联模式看作 16 状态而非 8 状态的 SEQ1。

15.3.5 排序转换期间的中断操

排序器可在两种工作模式下生成中断。这些模式由 ADCTRL2 中的中断模式使能控制位 (位 11、10、3、2) 确定。**示例 3** 的变化可用于显示在不同工作条件下，中断模式 1 和模式 2 的用途。

情形 1: 第一个序列和第二个序列中的采样数不相等

- 模式 1 中断操作 (即在每次 EOS 时发出中断请求)

1. 用 MAX_CONVn = 1 初始化排序器，以转换 I1 和 I2。

2. 在 ISR "a" 处, (通过软件) 将 MAX_CONVn 更改为 2, 以转换 V1、V2 和 V3。
3. 在 ISR "b" 处, 将发生以下事件:
 - a. 再次将 MAX_CONVn 更改为 1, 以转换 I1 和 I2。
 - b. 从 ADC 结果寄存器中读取 I1、I2、V1、V2 和 V3 值。
 - c. 排序器复位。
4. 重复步骤 2 和步骤 3。

注意: 每次 SEQ_CNTR 到 0 和识别到两个中断时, 将设置中断标志。

情形 2: 第一个序列和第二个序列中的采样数相等

- 模式 2 中断操作 (即每隔一个 EOS 信号产生中断请求)
1. 用 MAX_CONVn = 2 初始化排序器, 以转换 I1、I2 和 I3 (或 V1、V2 和 V3) 。
 2. 在 ISR "b" 和 "d" 处, 将发生以下事件:
 - a. 从 ADC 结果寄存器中读取值 I1、I2、I3、V1、V2 和 V3。
 - b. 排序器被复位。
 3. 重复步骤 2。

注意: 在每次 SEQ_CNTR 到达 0 时, 中断标志位都被置 1, 这将发生在 ADC 完成转换 I1、I2、和 I3 或者 V1、V2 和 V3 之后。但是只有转换完 V1、V2 和 V3 之后才会产生 EOS, 也即产生中断。

情形 3: 第一个序列和第二个序列中的采样数相等 (使用虚假读取)

- 模式 2 中断操作 (即每隔一个 EOS 信号产生中断请求)
1. 用 MAX_CONVn = 2 初始化排序器, 以转换 I1、I2 和 x (虚假采样) 。
 2. 在 ISR "b" 和 "d" 处, 将发生以下事件:
 - a. 从 ADC 结果寄存器读取值 I1、I2、I3, V1、V2 和 V3。
 - b. 排序器被复位。
 3. 重复步骤 2。

注意, 第三个 I 采样 (x) 为虚假采样, 而实际并不需要。但是, 为最大程度地减少 ISR 开销和 CPU 干预, 可

以充分利用模式 2 的“每隔一个”中断请求特性。

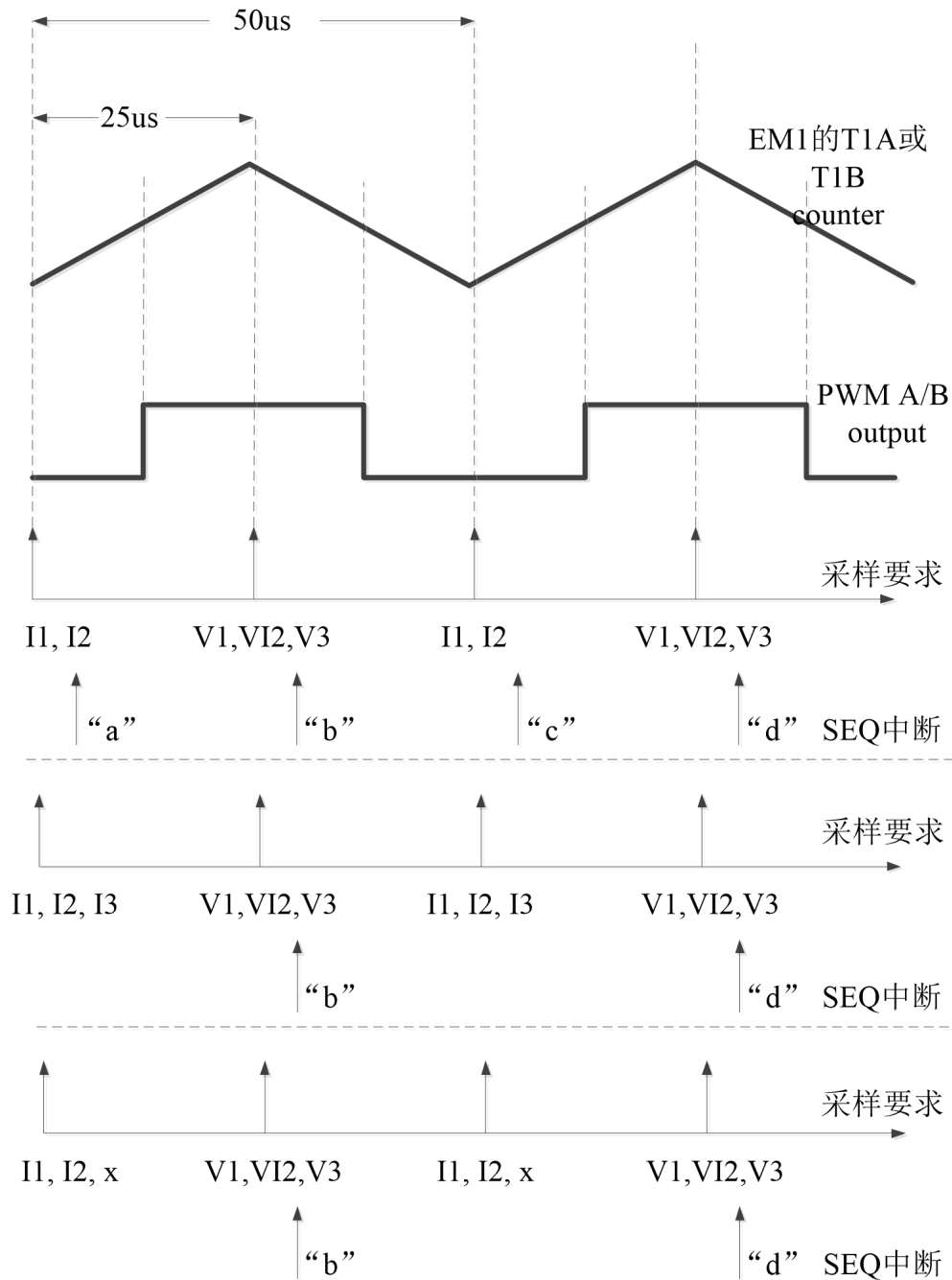


图 15-3-6 定序转换期间的中断操作

15.4 ADC 时钟预定

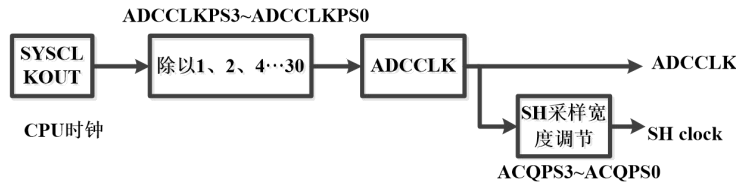


图 15-4-1 ADC 时钟预定标结构图

如图 15-4-1 所示，ADCCLK 为 ADC 模拟模块的主时钟，SH clock 为采样时钟（其高电平时长就是 ADC 的采样窗口长度）。ADCCLKPS3~0 位为 ADC 时钟预分频器控制位，ACQPS3~0 位为 ADC 采样窗口调节控制位。

15.5 ADC 校准

出厂校准数据会在上电后自动存入到 CALIBRATION 寄存器。此后，每次 ADC 正常转换时，都会自动将偏移误差值与转换数字码进行加法运算后再送入结果寄存器中。

表 15-5-1 校准 (CALIBRATION) 寄存器——地址 70B8h

15	14	13	12	11	10	9	8
D9	D8	D7	D6	D5	D4	D3	D2
7	6	5	4	3	2	1	0
D1	D0	0	0	0	0	0	0

15.6 ADC 相关寄存器概述

表 15-6-1 ADC 相关寄存器列表

名称	地址	大小 (x16)	描述
ADCTRL1	70A0h	1	ADC 控制寄存器 1
ADCTRL2	70A1h	1	ADC 控制寄存器 2
MAXCONV	70A2h	1	最大转换通道寄存器
CHSELSEQ1	70A3h	1	通道选择排序控制寄存器 1
CHSELSEQ2	70A4h	1	通道选择排序控制寄存器 2
CHSELSEQ3	70A5h	1	通道选择排序控制寄存器 3
CHSELSEQ4	70A6h	1	通道选择排序控制寄存器 4
AUTO_SEQ_SR	70A7h	1	自动排序状态寄存器
RESULT0	70A8h	1	转换结果缓冲寄存器 0
RESULT1	70A9h	1	转换结果缓冲寄存器 1

RESULT2	70AAh	1	转换结果缓冲寄存器 2
RESULT3	70ABh	1	转换结果缓冲寄存器 3
RESULT4	70ACh	1	转换结果缓冲寄存器 4
RESULT5	70ADh	1	转换结果缓冲寄存器 5
RESULT6	70AEh	1	转换结果缓冲寄存器 6
RESULT7	70AFh	1	转换结果缓冲寄存器 7
RESULT8	70B0h	1	转换结果缓冲寄存器 8
RESULT9	70B1h	1	转换结果缓冲寄存器 9
RESULT10	70B2h	1	转换结果缓冲寄存器 10
RESULT11	70B3h	1	转换结果缓冲寄存器 11
RESULT12	70B4h	1	转换结果缓冲寄存器 12
RESULT13	70B5h	1	转换结果缓冲寄存器 13
RESULT14	70B6h	1	转换结果缓冲寄存器 14
RESULT15	70B7h	1	转换结果缓冲寄存器 15
CALIBRATION	70B8h	1	校准寄存器
	7101h	1	PWM 同步触发采样模式选择寄存器

1.ADC 控制寄存器 1 (ADCTRL1) 地址 70A0h

15	14	13	12	11	10	9	8
保留	RESET	SOFT	FREE	ACQ PS3	ACQ PS2	ACQ PS1	ACQ PS0
	RS-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0
7	6	5	4	3	2	1	0
REG_T_PWD	CONT RUN	INT PRI	SEQ CASC	ADCCLK PS3	ADCCLK PS2	ADCCLK PS1	ADCCLK PS0
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

注：R=可读，W=可写，S=仅设置，C=清除，-0=复位后的值

位	名称	值	说明
15	保留		读取返回 0 值。写入无影响。
14	RESET	0 1	这一位引起整个 ADC 模块的总复位。所有的寄存器和排序器指针都复位到芯片复位引脚拉低或上电复位时的初始状态。 无影响 复位整个 ADC 模块（然后 ADC 逻辑将此位设为 0） 注：在 ADCTRL1 寄存器中用复位位。 系统复位时，ADC 模块被复位。如果想在其他时间对 ADC 进行复位，可以向这位写 1 实现。一个 NOP 指令后，就可以向 ADCTRL1 寄存器写入相应的值。
13-12	SOFT 位和 FREE 位	00 01	这两位决定仿真悬挂时，ADC 模块的工作情况。在自由运行模式下，ADC 模块可以继续它的工作。在停止模式下，ADC 模块可立刻停止或等完成当前操作后停止。 一旦仿真悬挂，立刻停止 在停止前完成当前转换

		X 1	自由运行, 不管仿真悬挂继续进行
11-8	ACQ PS [3:0]	0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111	采集窗口大小。此位字段控制 SOC 脉宽, 后者确定采样开关关闭的时间段。SOC 脉宽为 ADCTRL1[11:8] + 1 乘以 ADCCLK 周期。 1×TADCCLK 2×TADCCLK 3×TADCCLK 4×TADCCLK 5×TADCCLK 6×TADCCLK 7×TADCCLK 8×TADCCLK 9×TADCCLK 10×TADCCLK 11×TADCCLK 12×TADCCLK 13×TADCCLK 14×TADCCLK 15×TADCCLK 16×TADCCLK
7	REG_T_PWD	0 1	控制使能温度传感器输出到 ADC 的 A3 通道 (A3 未引出到芯片外面的引脚) 禁止温度传感器 使能温度传感器
6	CONT RUN	0 1	连续运行位。 此位决定排序器工作是处于连续转换模式还是启动/停止模式。可在当前转换序列有效时写入此位。此位在当前转换序列结束时生效; 在连续转换模式下, 不必复位排序器; 但是, 在启动/停止模式下必须复位排序器, 以使转换器进入状态 CONV00。启动/停止模式。到达 EOS 后排序器停止。除非执行了排序器复位, 否则排序器在遇到下一个 SOC 时将从结束时的状态启动。 连续转换模式。到达 EOS 后, 排序器的行为取决于 SEQ_OVRD 位的状态。如果清除此位, 则排序器将再次从其复位状态启动 (对 SEQ1 和级联模式为 CONV00, 对 SEQ2 为 CONV08)。如果设置了 SEQ_OVRD, 则排序器将再次从其当前位置启动, 而不会进行复位。
5	INT PRI	0 1	ADC 中断优先级位。 高优先级 低优先级
4	SEQ CASC	0	级联的序列发生器操作。此位决定 SEQ1 和 SEQ2 是作为 2 个 8 状态序列发生器工作还是作为单个 16 状态序列发生器工作 (SEQ)。 双序列发生器模式。SEQ1 和 SEQ2 作为 2 个 8 状态序列

		1	发生器工作。 级联模式。SEQ1 和 SEQ2 作为单个 16 状态序列发生器工作 (SEQ)。
3-0	ADCCLKPS [3:0]		内核时钟除法器。除了在 ADCCLKPS[3-0] 为 0000 时 HSPCLK 直通外, 将外设时钟 HSPCLK 除以 $2*ADCCLKPS[3-0]$ 。将分频后的时钟进一步除以 $ADCTRL1[7]+1$, 以产生内核时钟 ADCCLK。
			ADCCLKPS [3:0] 内核时钟除法器 ADCCLK
		0000	0 HSPCLK/(ADCTRL1[7] + 1)
		0001	1 HSPCLK/[2*(ADCTRL1[7] + 1)]
		0010	2 HSPCLK/[4*(ADCTRL1[7] + 1)]
		0011	3 HSPCLK/[6*(ADCTRL1[7] + 1)]
		0100	4 HSPCLK/[8*(ADCTRL1[7] + 1)]
		0101	5 HSPCLK/[10*(ADCTRL1[7] + 1)]
		0110	6 HSPCLK/[12*(ADCTRL1[7] + 1)]
		0111	7 HSPCLK/[14*(ADCTRL1[7] + 1)]
		1000	8 保留
		1001	9 保留
		1010	10 保留
		1011	11 保留
		1100	12 保留
		1101	13 保留
		1110	14 保留
		1111	15 保留

2.ADC 控制寄存器 2 (ADCTRL2) 地址 70A1h

15	14	13	12	11	10	9	8
EVB SOC SEQ	RST SEQ1	SOC SEQ1	SEQ1 BSY	INT ENA SEQ1 (Mode 1)	INT ENA SEQ1 (Mode 0)	INT FLAG SEQ1	EM1_T1A SOC SEQ1
RW-0	RS-0	RW-0	R-0	RW-0	RW-0	RC-0	RW-0
7	6	5	4	3	2	1	0
EXT SOC SEQ1	RST SEQ2	SOC SEQ2	SEQ2 BSY	INT ENA SEQ2 (Mode 1)	INT ENA SEQ2 (Mode 0)	INT FLAG SEQ2	EM1_T1B SOC SEQ2
RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

注: R=可读, W=可写, S=仅设置, C=清除, -0=复位后的值

位	名称	值	说明
15	EVB SOC SEQ		EVB 的 SOC 信号使能级联排序器位 此位未实际定义 (写 0 即可)

14	RST SEQ1	0 1	<p>复位排序器 1, 将 1 写入此位将使 SEQ1 或级联排序器立即复位到初始的“触发前”状态, 即在 CONV00 等待触发信号。将异常中止当前活动的转换序列。</p> <p>无操作</p> <p>立即将排序器复位到状态 CONV00</p>
13	SOC SEQ1	0 1	<p>排序器 1 (SEQ1) 或级联排序器的转换开始 (SOC) 触发器。可通过以下触发器设置此位:</p> <ul style="list-style-type: none"> •S/W——通过软件将 1 写入此位 •EM1_T1A——事件管理器的 T1A 比较器 •EXT ——外部引脚 (ADCSOC 引脚) <p>情形 1: SEQ1 空闲且已清除 SOC 位, 则 SEQ1 立即启动 (受仲裁器控制)。允许为任何“暂挂”的触发请求设置和清除此位。</p> <p>情形 2: SEQ1 忙且已清除 SOC 位, 则设置此位以指出触发请求正暂挂。当完成当前转换后最终启动 SEQ1 时, 将清除此位。</p> <p>情形 3: SEQ1 忙且设置了 SOC 位, 则忽略 (丢失) 此情况下出现的任何触发信号。</p> <p>清除暂挂的 SOC 触发器。</p> <p>注: 如果排序器已启动, 则自动清除此位; 因此, 写入 0 无影响, 即不能通过清除此位来停止已启动的排序器。</p> <p>软件触发器 - 从当前停止的位置启动 SEQ1 (即空闲模式)</p> <p>注: 不应在同一指令中设置 RST_SEQ1 (ADCTRL2.14) 和 SOC_SEQ1 (ADCTRL2.13) 位。这将使排序器复位, 但不会启动序列。正确的操作顺序为首先设置 RST_SEQ1 位, 并在随后的指令中设置 SOC_SEQ1 位。这样能够确定排序器是否已复位以及新序列是否已启动。此序列也适用于 RST_SEQ2 (ADCTRL2.6) 和 SOC_SEQ2 (ADCTRL2.5) 位。</p>
12	SEQ1 BSY	0 1	<p>SEQ1 忙状态位。</p> <p>当 ADC 自动转换排序正在进行时这位被置 1。当转换完成, 这位被清 0。</p> <p>0 排序器处于空闲状态</p> <p>1 排序器正在进行转换</p>
11-10	INT ENA SEQ1	00 01 10 11	<p>SEQ1 的中断方式使能控制位。</p> <p>00 中断禁止</p> <p>01 中断模式 1。当中断标志位置 1 时, 立刻申请中断</p> <p>10 中断模式 2。仅当中断标志位已经置 1 时, 才产生中断请求。</p> <p>11 保留</p>
9	INT FLAG SEQ1	0 1	<p>对 SEQ1 的 ADC 中断标志位</p> <p>这位表示中断事件是否已发生。用户通过向该位写 1 来清除此位。</p> <p>0 无中断事件发生</p> <p>1 中断事件发生过</p>
8	EM1_T1A SOC SEQ1	0 1	<p>EM1_T1A 对 SEQ1 产生 SOC 信号的屏蔽位。</p> <p>0 禁止用 EM1_T1A 的触发源启动 SEQ1</p> <p>1 允许用 EM1_T1A 的触发源启动 SEQ1</p>
7	EXT SOC SEQ1		<p>外部信号对 SEQ1 的转换启动位。</p>

		0	不起作用
		1	允许 ADCSOC 引脚上的信号触发 ADC 自动转换序列
6	RST SEQ2	0	排序器 2 复位位
		0	不起作用
		1	立刻复位 SEQ2, 使其指针指向 CONV08, 当前的转换序列将被中断。
5	SOC SEQ2		排序器 2 (SEQ2) 的转换开始触发器。(仅适用于双排序器模式; 在级联模式中被忽略。) 可通过以下触发器设置此位: <ul style="list-style-type: none"> • S/W——通过软件将 1 写入此位 • EM1_T1B——事件管理器的 T1B 比较器 当触发发生时, 有三种可能: 情形 1: SEQ2 空闲且已清除 SOC 位, 则 SEQ2 立即启动 (受仲裁器控制), 且允许为任何暂挂的触发请求清除该位。 情形 2: SEQ2 忙且已清除 SOC 位, 则设置此位以指出触发请求正暂挂。当完成当前转换后最终启动 SEQ2 时, 将清除此位。 情形 3: SEQ2 忙且设置了 SOC 位, 将忽略 (丢失) 此情况下出现的任何触发信号。 0 清除暂挂的 SOC 触发器 注: 如果排序器已启动, 则自动清除此位; 因此, 写入 0 无影响, 即不能通过清除此位来停止已启动的排序器。 1 从当前停止的位置启动 SEQ2 (即空闲模式)
4	SEQ2 BSY		SEQ2 忙状态位。 当 ADC 自动转换排序正在进行时这位被置 1。当转换完成, 这位被清 0。
		0	排序器处于空闲状态
		1	排序器正在进行转换
3-2	INT ENA SEQ2	00	SEQ2 的中断方式使能控制位。
		01	中断禁止
		10	中断模式 1。当中断标志位置 1 时, 立刻申请中断
			中断模式 2。仅当中断标志位已经置 1 时, 才产生中断请求。
		11	保留
1	INT FLAG SEQ2		对 SEQ2 的 ADC 中断标志位 这位表示中断事件是否已发生。用户通过向该位写 1 来清除此位。
		0	无中断事件发生
		1	中断事件发生过
0	EM1_T1B SOC SEQ2		EM1_T1B 对 SEQ2 产生 SOC 信号的屏蔽位。
		0	禁止用 EM1_T1B 的触发源启动 SEQ2
		1	允许用 EM1_T1B 的触发源启动 SEQ2

3.最大转换通道寄存器

最大转换通道寄存器 (MAXCONV) 地址 70A2h

15	14	13	12	11	10	9	8
保留							
R-x							
7	6	5	4	3	2	1	0
保留	MAX CONV2_2	MAX CONV2_1	MAX CONV2_0	MAX CONV1_3	MAX CONV1_2	MAX CONV1_1	MAX CONV1_0
R-x	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

注: R=可读, W=可写, S=仅设置, C=清除, -0=复位后的值

位	名称	值	说明
15-7			保留位
6-0			MAXCONVn 位域 MAXCONVn 位域定义在自动转换中最多转换的通道个数。该位域的值随着排序器工作模式的变化而变化。 对 SEQ1 操作, 使用 MAXCONV1_2~0 位。 对 SEQ2 操作, 使用 MAXCONV2_2~0 位。 对级联模式, 使用 MAXCONV1_3~0 位。 最大通道转化数对应关系如下:
		0000	1
		0001	2
		1110	15
		1111	16

ADCMAXCONV 寄存器位编程示例:

如果只需要 5 次转换, 则将 MAX_CONVn 设置为 4。

情形 1: 双模式 SEQ1 和级联模式排序器从 CONV00 变化到 CONV04, 并将 5 个转换结果存储在转换结果缓冲器的寄存器 RESULT 00 至 RESULT 04 中。

情形 2: 双模式 SEQ2 排序器从 CONV08 变化到 CONV12, 并将 5 个转换结果存储在转换结果缓冲器的寄存器 RESULT 08 至 RESULT 12 中。

情形 3: 双排序器模式的 MAX_CONV1 值大于 7

如果为双排序器模式 (即 2 个独立的 8 状态排序器) 选择了大于 7 的 MAX_CONV1 值, 则 SEQ_CNTR 将继续计数到超过 7, 这将导致排序器回绕到 CONV00, 并继续计数。

4.自动排序状态寄存器

自动排序状态寄存器 (AUTO_SEQ_SR) 地址 70A7h

15	14	13	12	11	10	9	8
保留				SEQ CNTR3	SEQ CNTR2	SEQ CNTR1	SEQ CNTR0
R-x				R-0	R-0	R-0	R-0
7	6	5	4	3	2	1	0
保留	SEQ2- State2	SEQ2- State1	SEQ2- State0	SEQ1- State3	SEQ1- State2	SEQ1- State1	SEQ1- State0
R-x	R-0	R-0	R-0	R-0	R-0	R-0	R-0

注：R=可读，W=可写，S=仅设置，C=清除，-0=复位后的值

位	名称	值	说明
15-12			保留位
11-8	SEQ CNTR3 ~ SEQ CNTR0	0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111	排序器计数状态位。可以表示当前排序器还有多少未转换通道个数。 有待转换个数： 1/0 (依赖于 SEQ1/2 BSY 位, BSY=0, 对应 1 个待转换) 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
7			保留位
6-4	SEQ2-State2 ~SEQ2-State0		SEQ2_STATE 位字段为 SEQ2 的指针。反映 SEQ2 指针的状态。
3-0	SEQ1-State3 ~SEQ1-State0		SEQ1_STATE 位字段为 SEQ1 或级联模式的指针。反映 SEQ1 或级联模式指针的状态。

5. ADC 输入通道选择排序控制寄存器

ADC 输入通道选择排序控制寄存器 (CHSELSEQn)

ADC 输入通道选择排序控制寄存器如下表所示：

ADC 输入通道选择排序控制寄存器 1 (CHSELSEQ1) 地址 70A3h

15	14	13	12	11	10	9	8
CONV03				CONV02			
RW-0				RW-0			
7	6	5	4	3	2	1	0
CONV01				CONV00			
RW-0				RW-0			

ADC 输入通道选择排序控制寄存器 2 (CHSELSEQ2) 地址 70A4h

15	14	13	12	11	10	9	8
CONV07				CONV06			
RW-0				RW-0			
7	6	5	4	3	2	1	0
CONV05				CONV04			
RW-0				RW-0			

ADC 输入通道选择排序控制寄存器 3 (CHSELSEQ3) 地址 70A5h

15	14	13	12	11	10	9	8
CONV11				CONV10			
RW-0				RW-0			
7	6	5	4	3	2	1	0
CONV09				CONV08			
RW-0				RW-0			

ADC 输入通道选择排序控制寄存器 4 (CHSELSEQ4) 地址 70A6h

15	14	13	12	11	10	9	8
CONV15				CONV14			
RW-0				RW-0			
7	6	5	4	3	2	1	0
CONV13				CONV12			
RW-0				RW-0			

注: R=可读, W=可写, S=仅设置, C=清除, -0=复位后的值

每一个 4 位域 CONVXX 都可以为自动排序转换选择 16 个模拟输入通道中的任何一个。

CONVXX 中 xx 的四位二进制码	ADC 模拟输入通道选择
0000	通道 A0
0001	通道 A1
0010	通道 A2
0011	通道 A3
0100	通道 A4
0101	通道 A5
0110	通道 A6
0111	通道 A7
1000	通道 B0
1001	通道 B1
1010	通道 B2
1011	通道 B3
1100	通道 B4
1101	通道 B5
1110	通道 B6
1111	通道 B7

6.ADC 转换结果缓冲寄存器

ADC 转换结果缓冲寄存器 (RESULTn) 地址 70A8h~70B7h

15	14	13	12	11	10	9	8
D11	D10	D9	D8	D7	D6	D5	D4
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
7	6	5	4	3	2	1	0
D3	D2	D1	D0	0	0	0	0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

注：R=可读，W=可写，S=仅设置，C=清除，-0=复位后的值

位	名称	值	说明
15-4			存放 ADC 的 12 转换结果 D11 对应最高位 D0 对应最低位
3-0			16 位寄存器的低 4 位，读取的结果都是 0。

7.PWM 同步触发采样模式选择寄存器—地址 7101h

15	14	13	12	11	10	9	8
CAP1_ST	CAP2_ST	CAP2_ST	保留	保留	保留	ADCT1AH_ CTL	ADCT1AL_ CTL
WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0
7	6	5	4	3	2	1	0
ADCT1BH_ CTL	ADCT1BL_ CTL	保留	保留	保留	COMP1_CT	COMP2_CT	COMP3_CT
WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0

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

位 9	ADCT1CompareAH_CTL: T1CompareA 信号上升沿触发;
位 8	ADCT1CompareAL_CTL: T1CompareA 信号下降沿触发;
位 7	ADCT1CompareBH_CTL: T1CompareB 信号上升沿触发;
位 6	ADCT1CompareBL_CTL: T1CompareB 信号下降沿触发;

16 运放与复用数字 IO 输入

16.1 运放与复用数字 IO 输入说明

ADP16F03 括 3 个可编程放大器 (PGA) 电路与 1 个可独立使用的运算放大器 (OPA) 电路。放大器输入端口通过内部施密特电路可作为逻辑信号的输入端口。如图 16-1 所示为运放与复用数字 IO 连接关系。

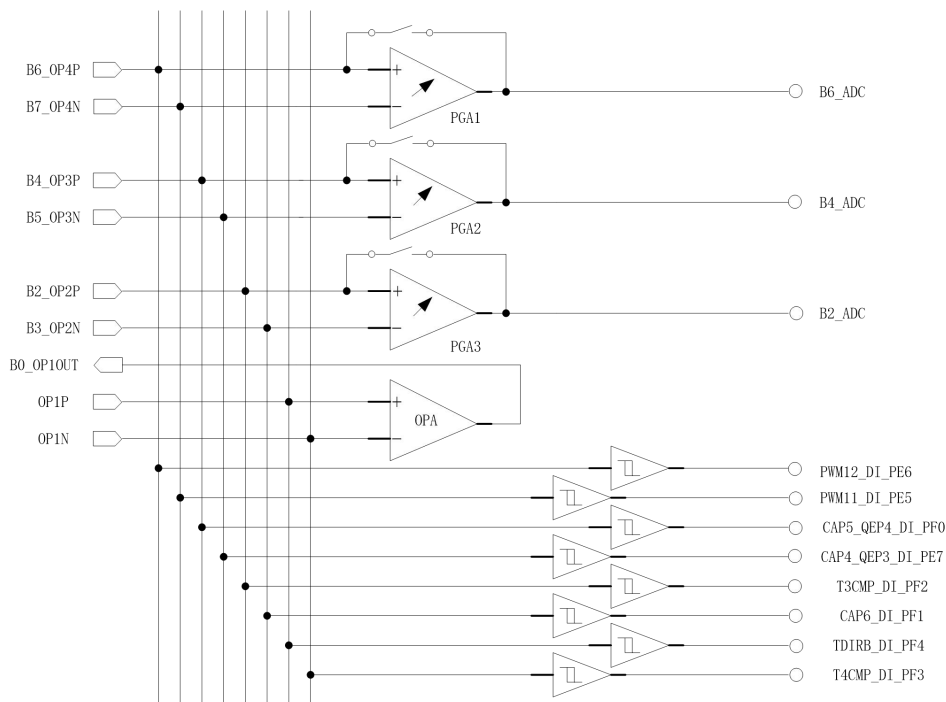


图 16-1 运放与复用数字 IO 连接关系

PGA 作为 ADC 的前置放大器，可通过寄存器 0X7106 进行配置，3 路 PGA 分别输出至 ADC 通道 B6_ADC、B4_ADC、B2_ADC。

默认情况下，寄存器 0X7106 为全 0 配置，PGA1、PGA2、PGA3 三个对应使能控制位 BIT14、BIT13、BIT12 为全 0，PGA 功能被禁止，旁路开关使引脚 B6_OP4P 与内部 ADC 采样通道 B6_ADC 直通，引脚 B4_OP3P 与内部 ADC 采样通道 B4_ADC 直通，引脚 B2_OP2P 与内部 ADC 采样通道 B2_ADC 直通，信号可直接通过引脚 B6_OP4P、B4_OP3P、B2_OP2P

被 ADC 直接采样。

当配置寄存器 0X7106 的 BIT14、BIT13、BIT12 分别为 1 时, 分别对应开启 PGA0、PGA1、PGA2 三个 ADC 前置放大器功能。引脚 B6_OP4P、B4_OP3P、B2_OP2P 与内部 ADC 通道之间开关被切断, 信号经过 PGA 放大后再被 ADC 采样。

PGA 等效电路如图 16-2 所示。该 PGA 可实现同相放大与反相放大。同相放大模式需在 PGA 的 P 端接入待放大信号, N 端接固定偏置。反相放大模式需在 PGA 的 N 端接入待放大信号, P 端接固定偏置。同相放大模式输入为高阻, 信号源内阻不影响 PGA 放大倍数; 反相输入模式输入内阻为 R1, 信号源内阻将影响 PGA 放大倍数, 为维持 PGA 增益不发生改变, 信号源需要一定驱动能力。

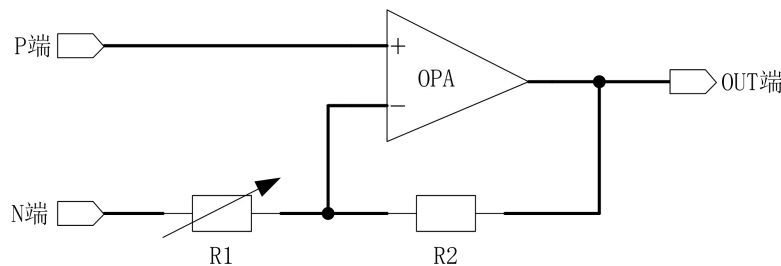


图 16-2 PGA 等效电路

使用 PGA 放大信号前需设置好 PGA 输出共模 VCMO, VCMO 计算公式为, 假定 VP 与 VN 分别为 P 端与 N 端输入共模:

$$V_{CMO} = [(V_P - V_N)/R_1] * R_2 + V_P$$

PGA 同相放大倍数 AVP 计算公式为:

$$A_{VP} = (R_2 + R_1)/R_1$$

PGA 反相放大倍数 AVN 计算公式为:

$$A_{VN} = -R_2/R_1$$

当配置寄存器 0X7106 改变 PGA 增益配置时, 将改变 R1 电阻值, 从而改变 PGA 的放大

倍数。

R1 与 R2 在 PGA 不同放大倍数下近似电阻值如表 16-1 所示。

表 16-1 PGA 内阻 R1 与 R2 数值

PGA 同相放大倍数	PGA 反相放大倍数	R1 电阻值 (kΩ)	R2 电阻值 (kΩ)
2	-1	22	22
4	-3	22/3	22
8	-7	22/7	22
10	-9	22/9	22
16	-15	22/15	22

芯片内除了 3 个带反馈回路的可编程放大器外，还内置一个 3 端口均引出的独立 OPA，如图 16-1 所示，该 OPA 通过 B0_OP1OUT 输出，该端口同时为 ADC 的 B0 通道。

默认情况下，寄存器 0X7106 为全 0 配置，OPA 使能关闭。

如图 16-1 所示 3 个 PGA 与 1 个 OPA 的输入端口可同时被复用为数字信号输入端口，默认情况下该复用端口为模拟信号输入端口，当控制寄存器 0X7103 的低 8 位时可打开相应数字信号通道。

16.2 运放与复用数字 IO 输入寄存器

16.2.1 通道对应的配置寄存器

1. 通道对应的配置寄存器—地址 7103h

15	14	13	12	11	10	9	8
COMP_H	COMPL	PDPA	保留	保留	保留	保留	保留
R_X	R_X	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
REG_INPUT_CT7	REG_INPUT_CT6	REG_INPUT_CT5	REG_INPUT_CT4	REG_INPUT_CT3	REG_INPUT_CT2	REG_INPUT_CT1	REG_INPUT_CT0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

位 15	COMP_H: 比较器 COMP_H 状态位; 1: A0_COMP0 输入电压超过上限电压;
------	----------------------------------------------------

	0: A0_COMP0 输入电压未超过上限电压;
位 14	COMPL: 比较器 COMPH 状态位; 1: A0_COMP0 输入电压低于下限电压; 0: A0_COMP0 输入电压未低于下限电压;
位 13	PDPA: 为 PDPINTA 输入源控制切换位; 1: 配置 PDPINTA 输入源使用 COMPH/COMPL 0: 配置 PDPINTA 输入源使用 B7_OP4N
位 12~位 8	保留
位 7	REG_INPUT_CT7: 复用 IOPE5 输入使能位; 0: 配置端口为 ADC 输入 B7 1: 配置端口为数字 IO 输入端口 IOPE5
位 6	REG_INPUT_CT6: 复用 IOPE6 输入使能位; 0: 配置端口为 ADC 输入 B6 1: 配置端口为数字 IO 输入端口 IOPE6
位 5	REG_INPUT_CT5: 复用 IOPE7 输入使能位; 0: 配置端口为 ADC 输入 B5 1: 配置端口为数字 IO 输入端口 IOPE7
位 4	REG_INPUT_CT4: 复用 IOPF0 输入使能位; 0: 配置端口为 ADC 输入 B4 1: 配置端口为数字 IO 输入端口 IOPF0
位 3	REG_INPUT_CT3: 复用 IOPF1 输入使能位; 0: 配置端口为 ADC 输入 B3 1: 配置端口为数字 IO 输入端口 IOPF1
位 2	REG_INPUT_CT2: 复用 IOPF2 输入使能位; 0: 配置端口为 ADC 输入 B2 1: 配置端口为数字 IO 输入端口 IOPF2
位 1	REG_INPUT_CT1: 复用 IOPF3 输入使能位; 0: 配置端口为 ADC 输入 B1 1: 配置端口为数字 IO 输入端口 IOPF2
位 0	REG_INPUT_CT0: 复用 IOPF3 输入使能位; 0: 配置端口为 ADC 输入 B0 1: 配置端口为数字 IO 输入端口 IOPF3

16.2.2 OPA 与 PGA 配置寄存器

1. OPA 与 PGA 配置寄存器—地址 7106h

15	14	13	12	11	10	9	8
REG_OPACT	REG_PGA1CT	REG_PGA2CT	REG_PGA3CT	保留	SETAG_PGA 1_B2	SETAG_PGA 1_B1	SETAG_PGA 1_B0
WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0
7	6	5	4	3	2	1	0
保留	SETAG_PGA2_B	SETAG_PGA	SETAG_PGA2_B0	保留	SETAG_PGA	SETAG_PGA	SETAG_PGA

	2	2_B1			3_B2	3_B1	3_B0
WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0

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

位 15	REG_OPACT: OPA 使能控制位; 0: 关闭运放, 释放 ADC 通道 1: 打开运放
位 14	REG_PGA1CT: PGA1 使能控制位; 0: 关闭运放, 释放 ADC 通道 1: 打开增益可编程运放 PGA1
位 13	REG_PGA2CT: PGA2 使能控制位; 0: 关闭运放, 释放 ADC 通道 1: 打开增益可编程运放 PGA2
位 12	REG_PGA3CT: PGA3 使能控制位; 0: 关闭运放, 释放 ADC 通道 1: 打开增益可编程运放 PGA3
位 11	保留
位 10~位 8	SETAG_PGA1_B2~ SETAG_PGA1_B0: PGA1 增益控制位;
位 7	保留
位 6~位 4	SETAG_PGA2_B2~ SETAG_PGA2_B0: PGA2 增益控制位;
位 3	保留
位 2~位 0	SETAG_PGA3_B2~ SETAG_PGA3_B0: PGA3 增益控制位;

表 16-2 PGA 放大倍数配置关系

SETAG_PGAX_B2	SETAG_PGAX_B1	SETAG_PGAX_B0	PGA 放大倍数
0	0	0	2/-1
0	0	1	4/-3
0	1	0	8/-7
0	1	1	10/-9
1	0	0	16/-15

注：SETAG_PGAX_B2、SETAG_PGAX_B1、SETAG_PGAX_B0 中 PGAX 代表 PGA3, PGA2, PGA1

2. 失调校准寄存器地址——0X753Bh

15	14	13	12	11	10	9	8
保留	保留	REG_OP1T_U1	REG_OP1T_U0	保留	REG_PGA1_VOFTU2	REG_PGA1_VOFTU1	REG_PGA1_VOFTU0
WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0
7	6	5	4	3	2	1	0
保留	REG_PGA2_VOFTU2	REG_PGA2_VOFTU1	REG_PGA2_VOFTU0	保留	REG_PGA3_VOFTU2	REG_PGA3_VOFTU1	REG_PGA3_VOFTU0

WR_0 WR_0 WR_0 WR_0 WR_0 WR_0 WR_0 WR_0

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

位 15~位 14	保留
位 13~位 12	OPA 失调校准控制位： 00: 默认值，不修调失调 01: 增加 3.5mV 输入失调 10: 增加-3.5mV 输入失调
位 11	保留
位 10~位 8	PGA1 失调校准控制位： 000: 默认值，不修调失调 001: 增加+2.5mV 失调 011: 增加+5mV 失调 101: 增加-2.5mV 失调 111: 增加-5mV 失调
位 7	保留
位 6~位 4	PGA2 失调校准控制位： 000: 默认值，不修调失调 001: 增加+2.5mV 失调 011: 增加+5mV 失调 101: 增加-2.5mV 失调 111: 增加-5mV 失调
位 3	保留
位 2~位 0	PGA3 失调校准控制位： 000: 默认值，不修调失调 001: 增加+2.5mV 失调 011: 增加+5mV 失调 101: 增加-2.5mV 失调 111: 增加-5mV 失调

17 电压比较器

17.1 电压比较器说明

ADP16F03 含 5 个迟滞比较器模块，如图 17-1-1 所示为比较器模块内部框架图。

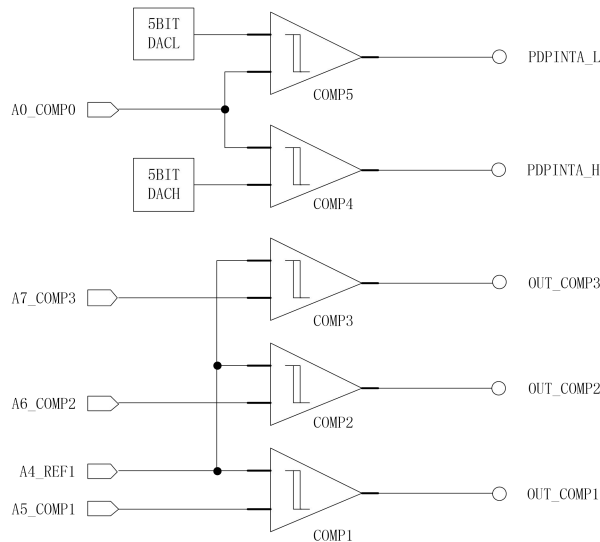


图 17-1-1 比较器模块内部框架图

其中 A0_COMP0、A7_COMP3、A6_COMP2、A4_REF1、A5_COMP1 为芯片外部输入 Pin 脚，PDPINTA_L、PDPINTA_H、OUT_COMP3、OUT_COM2、OUT_COMP1 为芯片内部信号线。

5 个比较器可大致分为 2 组，其中 COMP1，COMP2，COMP3 为外阈值组，COMP4，COMP5 为内阈值组；5 个比较器均为迟滞比较器，输入阻抗为高阻，迟滞电压可通过调整寄存器 0X753C 进行调整。

17.1.1 内阈值电压比较器

COMP4、COMP5 为内阈值组电压比较器，输入参考基准为 5BIT DAC 提供，5BIT DAC 输出高、低阈值，A0_COMP0 信号与高低阈值进行比较，产生 2BIT 的数字触发信号，用于监测 A0_COMP0 端口的输入电压大小，比较器两个输出端通过与门连接至 PDPINTA 中断触发信号（还有一路 PDPINTA 中断源信号来自复用 B7_OP4N），当发生 A0_COMP0 电压高

于 REFH 或低于 REFL，且持续时间超过延时滤波所设置时间，会引起 PAPINTA 中断。如下图 17-1-2 所示；内部 5BIT DAC 的输出电压与 5BIT 码 CODE5 的关系为：

$$V_{out} = V_{CCA}/32 * CODE5$$

2 个 5BIT DAC 共用一个使能信号，配置为 1 时使能，默认情况下为禁止，禁止时输出为高阻 0。

其中，VCCA 为参考电源电压，CODE5 为二进制配置位输入。如，CODE5 为 10000 时，Vout 为 VCCA/2；

VCCA 为模拟电源电压，通过内部 LDO 产生，典型值为 3.0V。

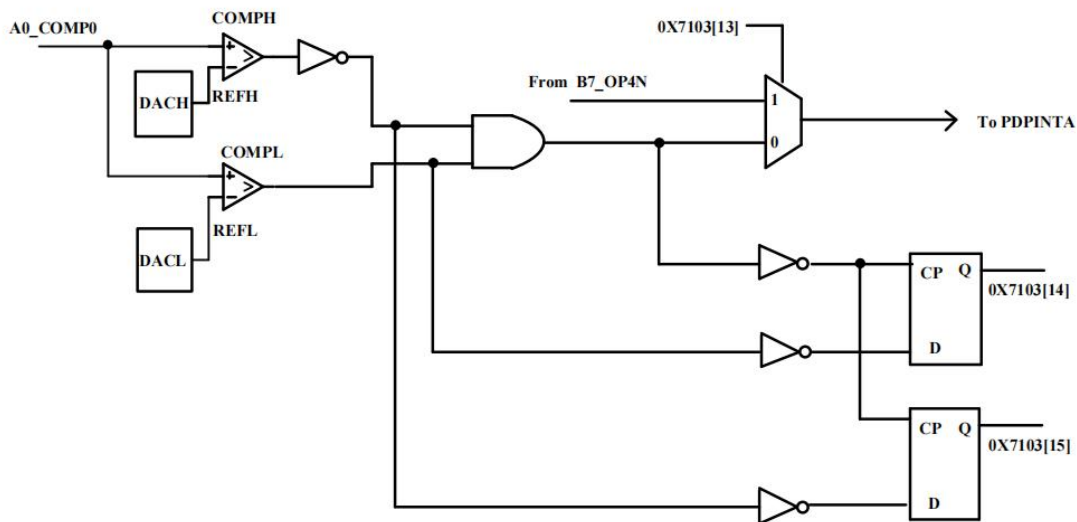


图 17-1-2 内阈值组电压比较器组框图

17.1.2 外阈值组电压比较器

COMP1~COMP3 外阈值组电压比较器，A4_REF1 作为三个比较器的外部公共参考电压输入端，比较器输出信号可用于监测 A5_COMP1、A6_COMP1、A7_COMP1 三个端口输入信号的相位，OUT_COMP1，OUT_COMP2，OUT_COMP2，可通过端口复用分别从 CAP1/IOA3，CAP2/IOA4，CAP3/IOA5 引脚输出，同时可以通过 CAP1/IOA3，CAP2/IOA4，CAP3/IOA5 的 Din 通路引入至芯片内部。也可以不通过 CAP1/IOA3，CAP2/IOA4，

CAP3/IOA5 引脚，直接引入芯片内部，如图 17-1-3 所示；

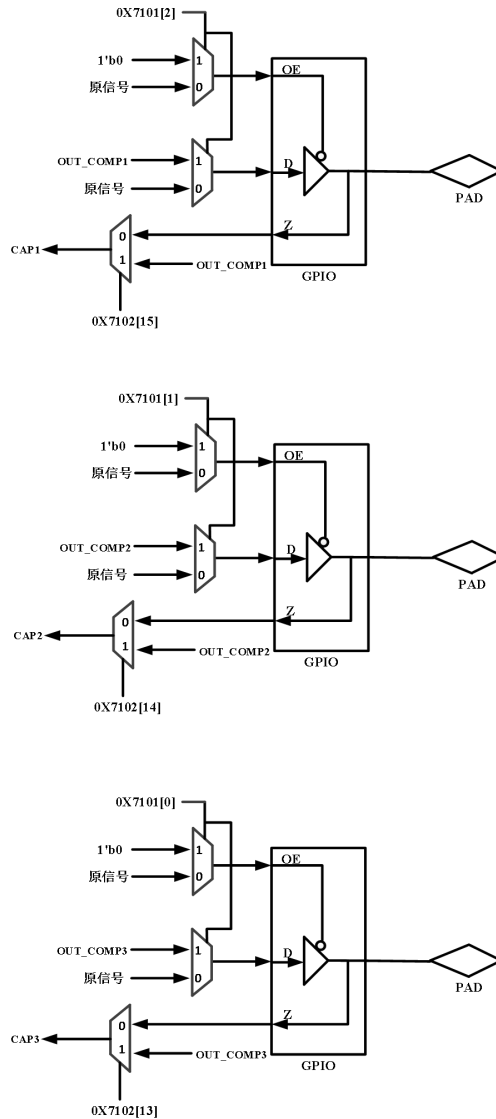


图 17-1-3 外阈值电压比较器组框图

17.1.3 电压比较器控制寄存器

1. 电压比较器控制寄存器 1—地址 7101h

15	14	13	12	11	10	9	8
CAP1_ST	CAP2_ST	CAP2_ST	保留	保留	保留	ADCT1AH_CTL	ADCT1AL_CTL
WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0

7	6	5	4	3	2	1	0
ADCT1BH_ CTL	ADCT1BL_ C TL	保留	保留	保留	COMP1_CT	COMP2_CT	COMP3_CT
WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0

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

位 15	CAP1_ST: CAP1 状态位;
位 14	CAP2_ST: CAP2 状态位;
位 13	CAP3_ST: CAP3 状态位;
位 12~位 10	保留
位 9	ADCT1CompareAH_CTL: T1CompareA 信号上升沿触发;
位 8	ADCT1CompareAL_CTL: T1CompareA 信号下降沿触发;
位 7	ADCT1CompareBH_CTL: T1CompareB 信号上升沿触发;
位 6	ADCT1CompareBL_CTL: T1CompareB 信号下降沿触发;
位 5~位 3	保留
位 2	COMP1_CT: 使能比较器 COMP1 输出到 CAP1_QEP1_IOPA3;
位 1	COMP2_CT: 使能比较器 COMP2 输出到 CAP2_QEP2_IOPA4;
位 0	COMP3_CT: 使能比较器 COMP3 输出到 CAP3_IOPA5;

2. 电压比较器控制寄存器 2—地址 7102h

15	14	13	12	11	10	9	8
REG_COMP 1CT	REG_COMP 2CT	REG_COMP 3CT	REG_COMP HCT	REG_COMP LCT	REG_DACCT	REG_REFH_ CODE4	REG_REFH_ CODE3
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
REG_REFH_ CODE2	REG_REFH_ CODE1	REG_REFH_ CODE0	REG_REFL_ CODE4	REG_REFL_ CODE3	REG_REFL_ CODE2	REG_REFL_ CODE1	REG_REFL_ CODE0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

位 15	REG_COMP1CT: 比较器 COMP1 使能位 0: 关闭比较器 COMP1 1: 打开比较器, 同时打开通往 CAP1 通道
位 14	REG_COMP2CT: 比较器 COMP2 使能位 0: 关闭比较器 COMP2 1: 打开比较器, 同时打开通往 CAP2 通道
位 13	REG_COMP3CT: 比较器 COMP3 使能位 0: 关闭比较器 COMP3 1: 打开比较器, 同时打开通往 CAP3 通道
位 12	REG_COMPHCT: 比较器 COMPH 使能位 0: 关闭比较器 COMPH

	1: 打开比较器 COMPH
位 11	REG_COMPLCT: 比较器 COMPL 使能位 0: 关闭比较器 COMPL 1: 打开比较器 COMPL
位 10	REG_DACCT: 内部两个 5bit DAC 使能位; 0: 关闭 DAC 1: 使能 DAC
位 9~位 5	REG_REFH_CODE4~ REG_REFH_CODE0: 5 位 DAC 输入数据位, 高阈值输入; (Vout = VCCA/32*CODE5), VCCA 典型值 3.0V。
位 4~位 0	REG_REFL_CODE4~ REG_REFL_CODE0: 5 位 DAC 输入数据位, 低阈值输入; (Vout = VCCA/32*CODE5), VCCA 典型值 3.0V。

3. COMPH/COMPL 状态寄存器—地址 7103h

15	14	13	12	11	10	9	8
COMPH	COMPL	PDPA	保留	保留	保留	保留	保留
R_X	R_X	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
REG_INPUT_CT7	REG_INPUT_CT6	REG_INPUT_CT5	REG_INPUT_CT4	REG_INPUT_CT3	REG_INPUT_CT2	REG_INPUT_CT1	REG_INPUT_CT0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

位 15	COMPH: 比较器 COMPH 状态位; 1: A0_COMP0 输入电压超过上限电压; 0: A0_COMP0 输入电压未超过上限电压;
位 14	COMPL: 比较器 COMPL 状态位; 1: A0_COMP0 输入电压低于下限电压; 0: A0_COMP0 输入电压未低于下限电压;
位 13	PDPA : PDPA 引脚输入选择位; 0: PDPAINTA 输入为 A0_COMP0 1: PDPAINTA 输入为 B7_OP4N
位 8~位 12	保留
位 7	REG_INPUT_CT7: B7_OP4N 复用为数字输入端口使能 0: B7_OP4N 为模拟信号输入端口 1: B7_OP4N 为数字输入端口, 可输入数字信号至 PWM11
位 6	REG_INPUT_CT6: B7_OP4N 复用为数字输入端口使能 0: B7_OP4N 为模拟信号输入端口 1: B7_OP4N 为数字输入端口, 可输入数字信号至 PWM12
位 5	REG_INPUT_CT5: B7_OP4N 复用为数字输入端口使能 0: B7_OP4N 为模拟信号输入端口 1: B7_OP4N 为数字输入端口, 可输入数字信号至 CAP4_QEP3

位 4	REG_INPUT_CT7: B7_OP4N 复用为数字输入端口使能 0: B7_OP4N 为模拟信号输入端口 1: B7_OP4N 为数字输入端口, 可输入数字信号至 CAP5_QEP4
位 3	REG_INPUT_CT7: B7_OP4N 复用为数字输入端口使能 0: B7_OP4N 为模拟信号输入端口 1: B7_OP4N 为数字输入端口, 可输入数字信号至 CAP6
位 2	REG_INPUT_CT7: B7_OP4N 复用为数字输入端口使能 0: B7_OP4N 为模拟信号输入端口 1: B7_OP4N 为数字输入端口, 可输入数字信号至 T3CMP
位 1	REG_INPUT_CT7: B7_OP4N 复用为数字输入端口使能 0: B7_OP4N 为模拟信号输入端口 1: B7_OP4N 为数字输入端口, 可输入数字信号至 T4CMP
位 0	REG_INPUT_CT7: B7_OP4N 复用为数字输入端口使能 0: B7_OP4N 为模拟信号输入端口 1: B7_OP4N 为数字输入端口, 可输入数字信号至 TDIRB

4. COMP 配置寄存器—地址 753Ch

15	14	13	12	11	10	9	8
REG_PLL3	REG_PLL2	REG_PLL1	REG_PLL0	保留	保留	REG_CPL_H YTU1	REG_CPL_H YTU0
R_X	R_X	R_X	R_X	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
REG_CPH_H YTU1	REG_CPH_H YTU0	REG_CP3_H YTU1	REG_CP3_H YTU0	REG_CP2_H YTU1	REG_CP2_H YTU0	REG_CP1_H YTU1	REG_CP1_H YTU0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

位 15~12	REG_PLL3~REG_PLL0: 时钟不交叠时间控制位, 出厂已校准, 不允许随意修改
位 11	保留
位 10	保留
位 9~8	REG_CPL_HYTU1~ REG_CPL_HYTU0: 比较器 COMPL 迟滞电压调整控制位
位 7~6	REG_CPH_HYTU1~ REG_CPH_HYTU0: 比较器 COMPH 迟滞电压调整控制位
位 5~4	REG_CP3_HYTU1~ REG_CP3_HYTU0: 比较器 COMP3 迟滞电压调整控制位
位 3~2	REG_CP2_HYTU1~ REG_CP2_HYTU0: 比较器 COMP2 迟滞电压调整控制位
位 1~0	REG_CP1_HYTU1~ REG_CP1_HYTU0: 比较器 COMP1 迟滞电压调整控制位

表 17-1-1 COMP 迟滞电压配置关系

REG_CPX_HYTU1	REG_CPX_HYTU1	迟滞电压(/mV)
0	0	50
0	1	65
1	0	35
1	1	0

注：REG_CPX_HYTU1、REG_CPX_HYTU0 中 CPX 代表 CPL, CPH, CP3, CP2, CP1

18. 数字延时滤波

18.1 数字延时滤波模块结构

数字延时滤波单元，作用于以下信号：CAP1~CAP3, XINT1、XINT2/ADCSOC 以及 PDPINTA; CAP1, CAP2, CAP3 共用一组配置控制（配置控制信号相同），PDPINTA 单独用一组配置控制，XINT1 和 XINT2 共用一组配置控制。

在满足跳变后持续超过 6（可通过数字滤波模块寄存器设置）个时钟周期的高/低电平条件时，返些信号才真正会被响应。

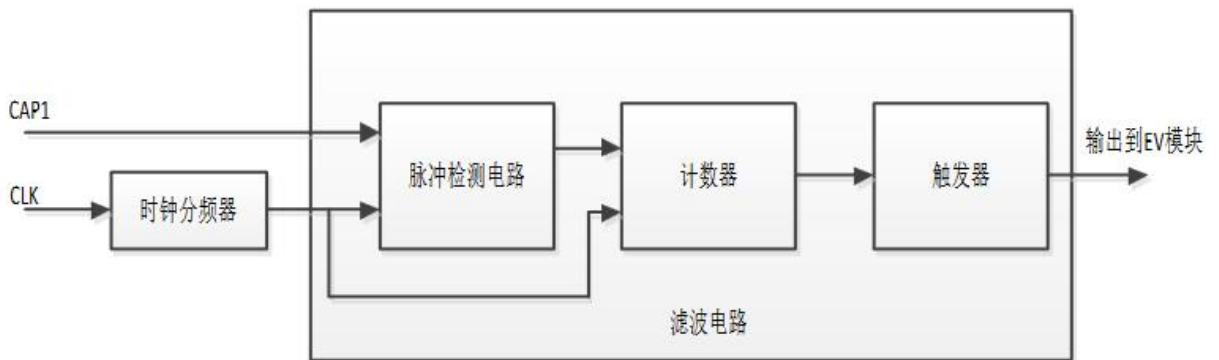


图 18-1 CAP1 数字延时滤波示意图

以 CAP1 信号为例，图 18-1 为 CAP1 数字延时滤波示意图，可通过 BIT[4:0]可以配置延时电路的计数时钟，最终得到不同的延时滤波时间。表 18-1 为 100M 主频下延时滤波时间参考配置表。

表 18-1 延时滤波时间参考配置表

BIT[4:0]	滤波宽度 100MHz (周期数)
00000	60ns (6)
00001	120ns (12)
00010	200ns (20)
00011	400ns (40)
00100	600ns (60)
00101	800ns (80)
00110	1000ns (100)
00111	1200ns (120)
01000	1400ns (140)

01001	1600ns	(160)
01010	1800ns	(180)
01011	2000ns	(200)
01100	2200ns	(220)
01101	2400ns	(240)
01110	2600ns	(260)
01111	2800ns	(280)
10000	3000ns	(300)
10001	3200ns	(320)
10010	3400ns	(340)
10011	3600ns	(360)
10100	3800ns	(380)
10101	4000ns	(400)
10110	4200ns	(420)
10111	4400ns	(440)
11000	4600ns	(460)
11001	4800ns	(480)
11010	5000ns	(500)
其余	5000ns	(500)

滤波模块配置列表如上所示，当配置为 0 时，输入脉冲小于 6 个周期将会被过滤掉。最大滤波宽度为 499 个时钟周期。

18.2 数字延时滤波模块寄存器说明

18.2.1 数字延时滤波控制寄存器

1. 数字滤波控制寄存器—地址 753Fh

15	14	13	12	11	10	9	8
CfgError	XINTCTL_re g4	XINTCTL_re g3	XINTCTL_re g2	XINTCTL_re g1	XINTCTL_re g0	PDPINTCTL _reg4	PDPINTCTL _reg3
R_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0
7	6	5	4	3	2	1	0
PDPINTCTL _reg2	PDPINTCTL _reg1	PDPINTCTL _reg0	CAPCTL_re g4	CAPCTL_re g3	CAPCTL_re g2	CAPCTL_re g1	CAPCTL_re g0
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

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

位 15	CfgError : flash 配置寄存器错误标志位; 0: 正确 1: 错误
位 14~位 10	XINTCTL_reg4~ XINTCTL_reg0: XINT1 和 XINT2/ADCSOC 延时电路计数时钟设置, 详情见表 18-1 延时滤波时间参考配置表;
位 9~位 5	PDPINTCTL_reg4~ PDPINTCTL_reg0: PDPINTA 延时电路计数时钟设置, 详情见表 18-1 延时滤波时间参考配置表;
位 4~位 0	CAPCTL_reg4~ CAPCTL_reg0: CAP1~ CAP3 延时电路计数时钟设置, 详情见表 18-1 延时滤波时间参考配置表;

19. 供电

19.1 片上 LDO 电源管理

ADP16F03 全芯片有 4 个电压域: 3.2~5.5V 电压域 (VDDIO) , 1.2V 电压域 (VDD) , 1.7V 电压域, 3V 电压域 (VCCA)。芯片内供电采用 LDO 技术, 仅需对 VDDIO 输入 3.2~5.5V 电源, VDD 与 VCCA 引脚增加去耦电容即可完成对芯片各模块的供电。图 19-1-1 所示为芯片内 (除 IO 外) 各模块的供电关系, IO 驱动电源为 VDDIO。

三个 LDO, 一个为 3.2~5.5V 转 3V 的 LDO, 为 ADC 提供电源电压, 引脚需接大于 2.2uF 的去耦电容, 为提高电源高频纹波抑制能力, 需要增加一个 0.1uF 电容并联。

一个为 3.2~5.5V 转 1.2V 的 LDO, 为内核和 PLL 提供电源电压, , 引脚需接大于 2.2uF 的去耦电容, 为提高电源高频纹波抑制能力, 需要增加一个 0.1uF 电容并联。

一个为 3.2~5.5V 转 1.7V 的 LDO, 为 FLASH 提供电源电压, 无需片外电容。

LDO3, LDO17, LDOCK 参考基准均来源于内部带隙基准输出的 1.2V, LDO12 参考基准来源于内部 1V 电压基准。该 1V 基准同时作为 Flash 的参考基准源。1V 电压基准的实现方式为 1.2V 基准通过运放比例衰减后得到。

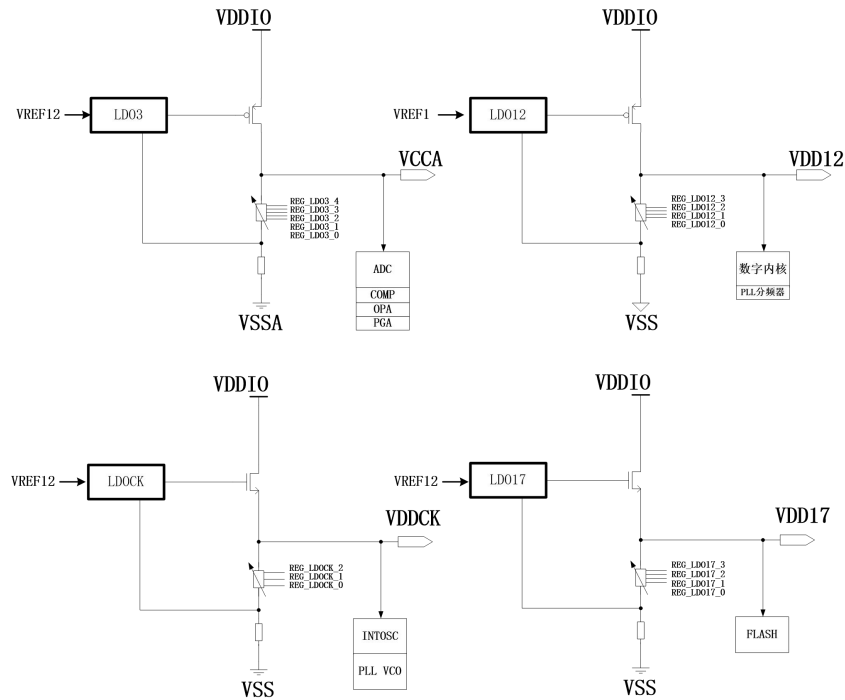


图 19-1-1 芯片各模块供电框图

ADP16F03 芯片内置 PORBOR 模块对芯片上电过程进行管控。图 19-1-2 所示为芯片上电时序图。芯片内置 PORBOR 模块，监控电源状态。

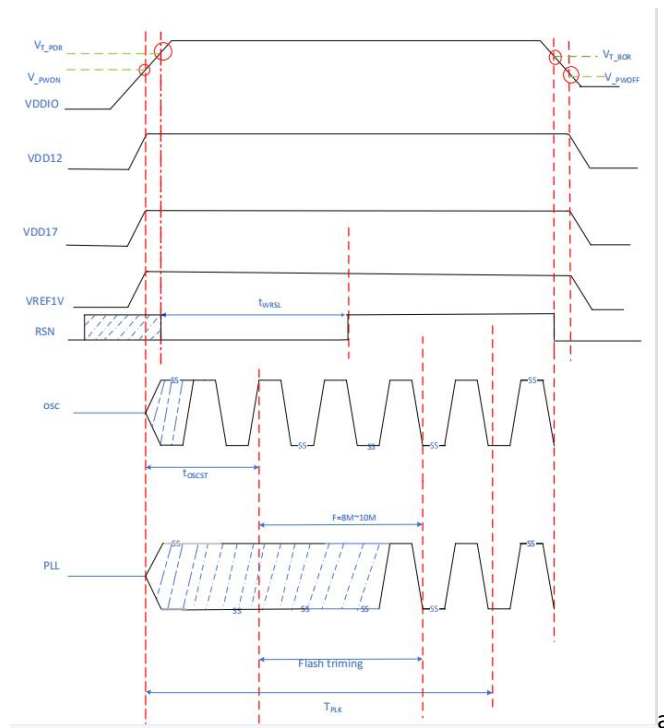


图 19-1-2 BOR、POR 上电掉电复位时序图

上、下电关键电气与延迟参数

		最小值	标称值	最大值	单位
Vt_POR	上电释放 POR 信号时 VDDIO 电压点		3.0		V
Vt_BOR	下电释放 BOR 信号时 VDDIO 电压点		2.8		V
tOSCST	10MHz 振荡器启动时间		20		us
tPLK	PLL 锁定时间		100		us
tWRSL	POR 信号释放后系统维持复位时间		40		us

20.电气参数

20.1 建议的运行条件

		最小值	标称值	最大值	单位
V _{DDIO}	电源电压, I/O	3.2	3.3	5	V
VSS	电源接地	0			V
f _{CLKOUT}	设备时钟频率 (系统时钟)				MHz
V _{IH} ^③	高电平输入电压	所有 GPIO		0.7*VDDIO	V
V _{IL}	低电平输入电压	所有 GPIO		VSS-0.3	V
I _{OH}	高电平输出源电流	所有 GPIO		4	mA
I _{OL}	低电平输出灌电流	所有 GPIO		4	mA

20.2 建议运行温度范围内的电气特性

参数		测试条件	最小值	标称值	最大值	单位
V _{OH} 高电平输出电压	I _{OH} =I _{OH} 最大值		VDDIO-0.8		V	
	I _{OH} =50uA		VDDIO-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) ⁽¹⁾	上拉使能	Vin=0V, VDDIO=3.3V	100		uA	
		Vin=0V, VDDIO=5V	240		μA	
	上拉不使能	Vin=0V, VDDIO=3.3V	±2		uA	
		Vin=0V, VDDIO=5V	±2		uA	
I _{IH} 输入电流 (高电平) ⁽¹⁾	上拉使能	Vin=VDDIO, VDDIO=3.3V 或 5V	±2		μA	
	上拉不使能		±2		uA	
I _{OZ} 输出电流		Vo=VDDIO 或 0	±2		μA	
TRSTN 下拉电阻			15		kΩ	

(1) 除 IOPB3, IOPB1, IOPA7, IOPB2, IOPB0, IOPA6 外所有 IO, IOPB3, IOPB1, IOPA7, IOPB2, IOPB0, IOPA6 内部无上拉或下拉, I_{IH}/I_{IL} 最大值为 ±2uA

OPA 电气参数:

参数	测试条件	最小值	标称值	最大值	单位
输入共模范围		0		2.0	V
线性输出范围		0.1		2.8	V
输入失调			±2		mV
压摆率	负载 20pF		10		V/us

PGA 电气参数:

参数	测试条件	最小值	标称值	最大值	单位
输入共模范围		0		2.0	V
线性输出范围		0.1		2.8	
输入失调			±2.5		mV
压摆率	负载 20pF		10		V/us

COMP 电气参数:

参数	测试条件	最小值	标称值	最大值	单位
输入共模范围		0		3	V
输入迟滞电压	REG_CP5~1_HYTU1/ REG_CP5~1_HYTU0: 00		35		mV
	REG_CP5~1_HYTU1/ REG_CP5~1_HYTU0: 01		45		mV
	REG_CP5~1_HYTU1/ REG_CP5~1_HYTU0: 10		18		mV
	REG_CP5~1_HYTU1/ REG_CP5~1_HYTU0: 11		0		mV

20.3 减少流耗

ADP16F03 DSP 采用了一种独特的方法来减少器件的电流消耗。通过关闭指定不使用的
外设模块的时钟可减少流耗。

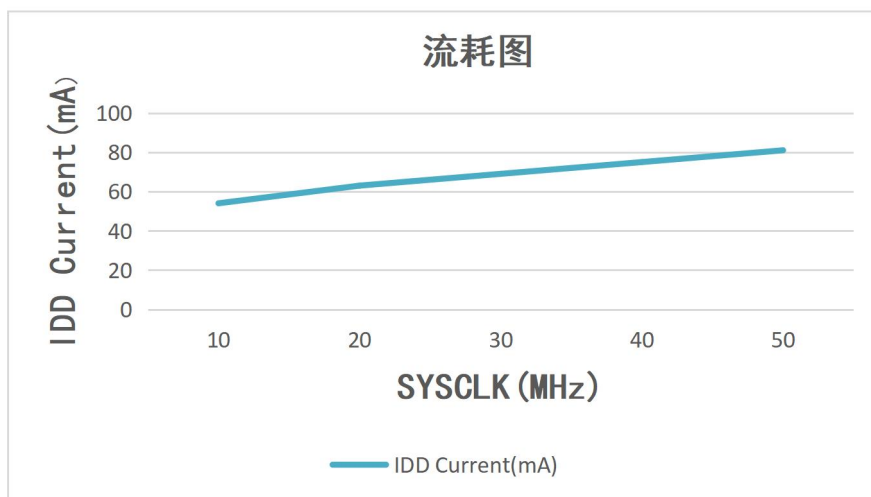


图 20-1 ADP16F03 在频率范围内的典型电流消耗 (使能外设时钟)

20.4 信号转换电平

请注意一些信号使用不同的参考电压，请参阅推荐的工作条件表。输出电平被驱动至 2.4 V 的最小逻辑高电平和 0.8 V 的最大逻辑低电平。图 20-2 显示了输出电平。

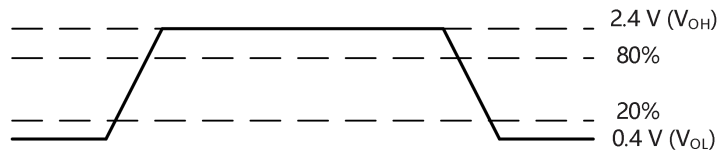


图 20-2 输出电平

输出转换时间指定如下：

1. 对于一个从高电平到低电平的转换，输出不再被认为是高电平的电平为低于 V_{OH} （最小值）的 80%，而输出不再被认为是低电平的电平为 V_{OL} （最大值）的 20% 且更低。
2. 对于一个从低电平到高电平的转换，输出不再被认为是低电平的电平为高于 V_{OL} （最大值）的 20%，而输出不再被认为是高电平的电平为 V_{OH} （最小值）的 80% 且更高。

图 20-3 显示了输入电平。

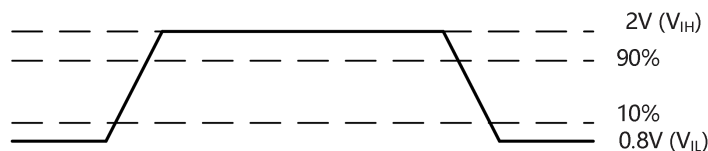


图 20-3 输入电平

输入转换时间指定如下：

1. 对于一个从高电平到低电平的转换，输入不再被认为是高电平的电平为低于 V_{IH} （最小值）的 90%，而输入不再被认为低电平的电平为 V_{IL} （最大值）的 10% 且更低。
2. 对于一个从低电平到高电平的转换，输入不再被认为是低电平的电平为高于 V_{IL} （最大值）的 10%，而输入不再被认为高电平的电平为 V_{IH} （最小值）的 90% 且更高。

20.5 时序参数符号

所用的时序参数符号根据 JEDEC 标准 100 创建。为了缩短符号，一些引脚名称和其他术语缩写如下：

CI	X1/CLKIN	CO	CLKOUT
INT	XINT1, XINT2	RS	复位引脚 RS

小写下标和它们的含意：

a	访问时间
c	周期时间 (周期)
d	延迟时间
f	下降时间
h	保持时间
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)

字母和符号和它们的含意：

H	高
L	低
V	有效
X	未知、变、或者无关电平
Z	高阻抗

20.6 12 位模数转换器 (ADC)

12 位 ADC 的模拟电路有单独的电源总线。这些引脚涉及到 VCCA 和 VSSA。这些隔离的电源总线是为了通过防止数字逻辑电路 VSS 和 VCC 产生的噪声连接到模拟模块从而增强 ADC 的能力。所以 ADC 规格详述将遵守 VSSA 给出，除非有另外说明。

20.6.1 建议运行环境

表 20-1 建议运行环境

		最小值	正常值	最大值	单位
V_{CCA}	模拟供电电压 (内设 LDO, 引脚只需外挂电容)		3		V
V_{SSA}	模拟地		0		V
V_{AI}	模拟输入电压, ADCINA0—ADCINB7	0		3	V

20.6.2 ADC 运行频率

表 20-2 ADC 运行频率

	MIN	MAX	单位
ADC 运行频率	0.1	14	MHZ

20.7 在推荐运行条件范围下的 ADC 电气特性

表 20-3 在推荐运行条件范围下的 ADC 电气特性⁽¹⁾⁽²⁾

分辨率	描述		最小值	典型值	最大值	单位
I_{CCA}	模拟供电电流	$V_{CCA}=3V$				mA
		$V_{CCA}=3V$	PLL or OSC power down			uA
I_{ADCIN}	模拟输入漏电流				±2	uA
C_{ai}	模拟输入电容	模拟输入引脚上的典型容性负载	非采样时	10		pF
			采样时	13.5		
$E_{DNL}^{(3)(4)}$	微分非线性误差	实际步长与理想值之间的误差			±2	LSB
$E_{INL}^{(4)}$	积分非线性误差	从最好的直线到模数转换特性的最大偏移量 不包括量化错误			±4	LSB
$t_d(PU)$ 后	延迟时间, ADC 上电使能后	上电后模拟电路稳定所需时间				μs
Z_{AI}	模拟输入源阻抗	转换到规格内最小 t_w 所需的模拟输入源阻抗				Ω
偏移误差			-15		15	LSB
内部基准时总增益误差			-50		50	LSB
通道到通道的偏移变化				±4		LSB

通道到通道的增益变化			±4		LSB
模拟输入					
模拟输入电压 (ADCINx 至 LO) ⁽⁶⁾		LO		3	V
LO		-5		5	mV
内部参考基准 ⁽⁵⁾					
LO			0		V
V _{CCA}			3		V
温度系数			50		PPM/°C
AC 技术规范					
SINAD (7.8kHz) 信噪比+失真			61.9		dB
SNR (7.8kHz) 信噪比			62.2		dB
THD (7.8kHz) 总谐波失真			-73		dB
ENOB (7.8kHz) 有效位数			10		位
SFDR (7.8kHz) 无杂散动态范围			74		dB

(1) 在 14 MHz ADCCLK 上测得。

(2) 这个表中的所有电压相对 V_{SSA}。

(3) Advchip 指定 ADC 将无丢码。

(4) 1 个 LSB 为 3/4096=0.732 mV 的加权值。

(5) 内部基准的精度对于总体增益十分关键。电压 V_{CCA} - LO 将确定总体精度，针对内部基准选项的温度范围内的增益误差将取决于所使用源的温度参数。

(6) 应用到一个模拟输入引脚上的高于 V_{CCA} +0.3V 或者低于 V_{SSA}-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。并且应用到一个模拟输入引脚上的高于 V_{CCA} 或者低于 V_{SSA} 的电压会影响 ADC 的各项性能参数。

21 机械数据

LQFP48

mm

单位: mm

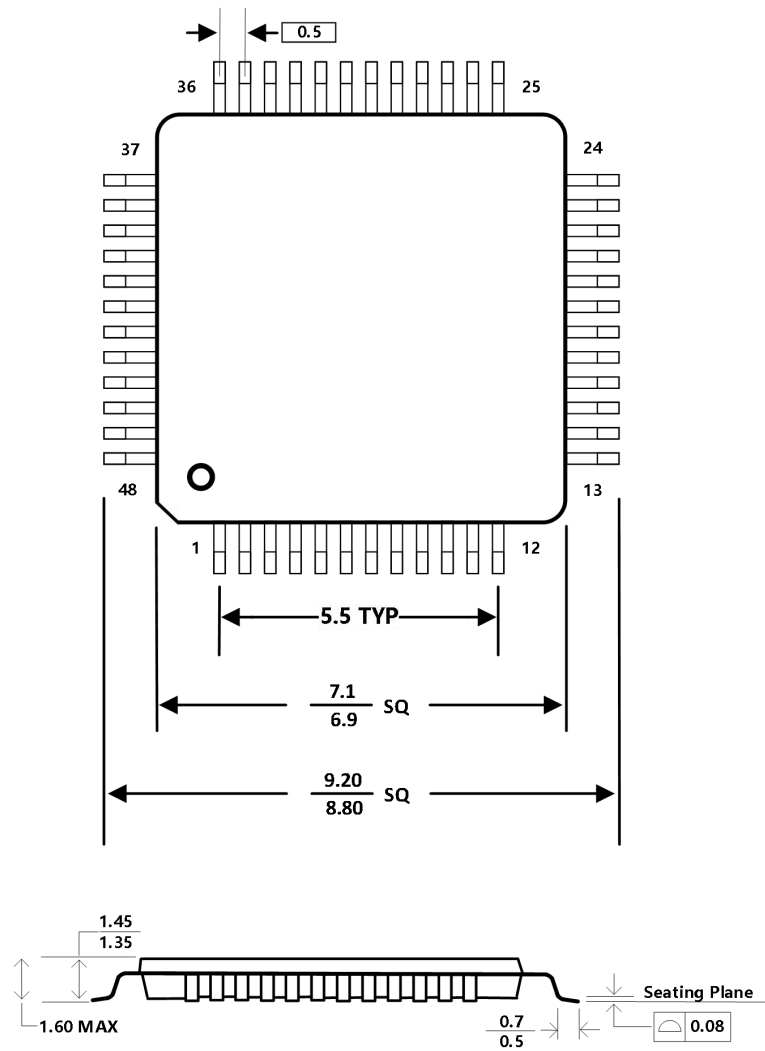


图 21-1 LQFP48 封装外形尺寸图

LQFP32

mm

单位: mm

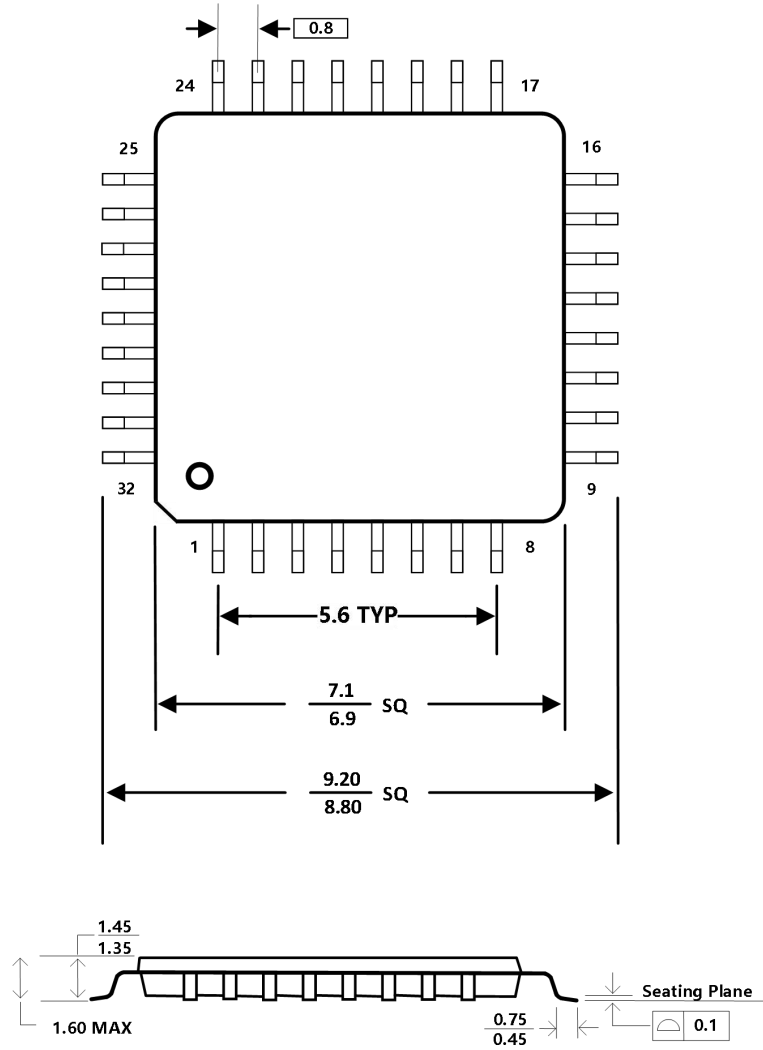


图 21-2 LQFP32 封装外形尺寸图

QFN32

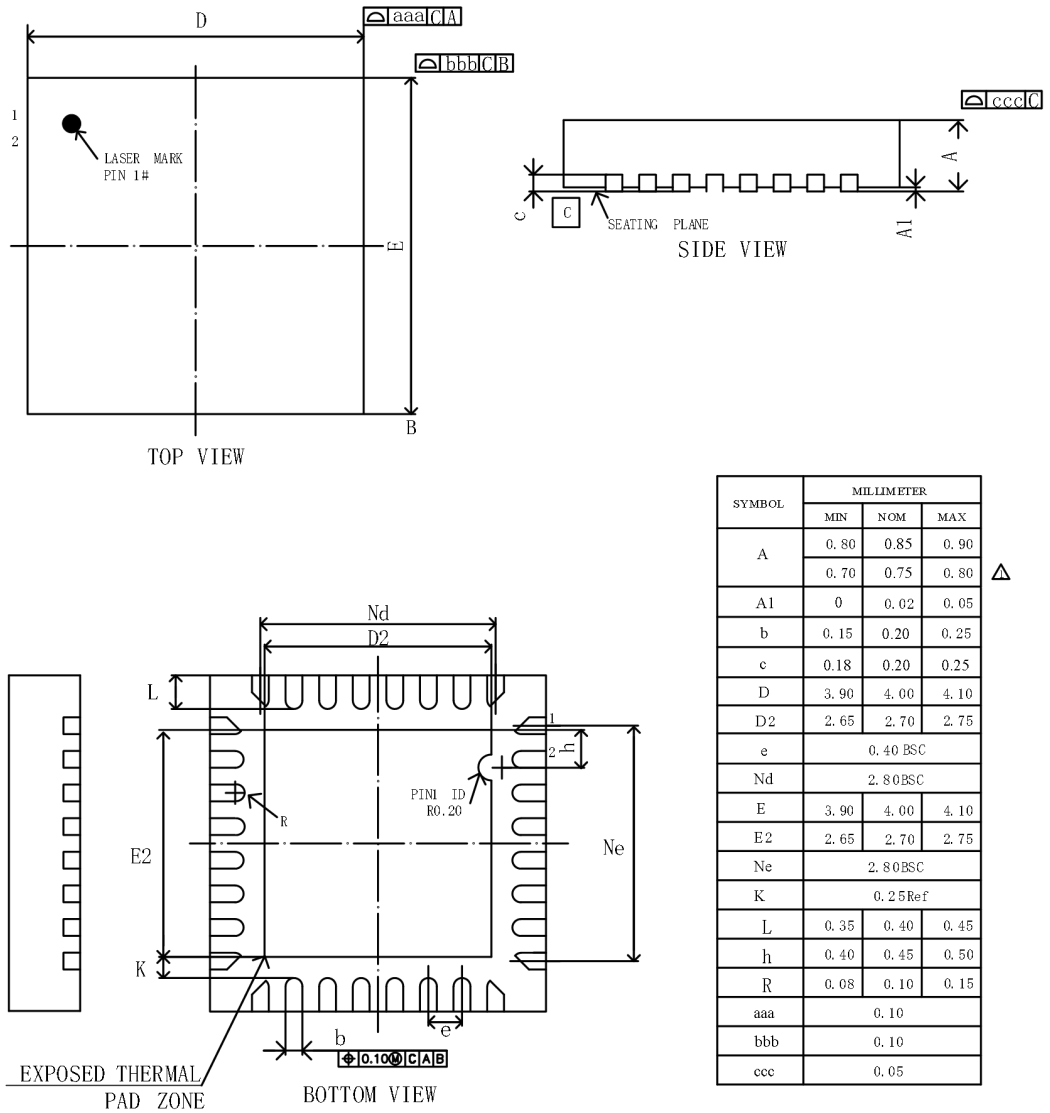


图 21-3 QFN32 封装外形尺寸图

22 重要注意事项及声明

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

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

联系方式

公司网址: www.advancechip.com

联系邮箱: sales@advancechip.com

销售联系电话: 0731-88731027 (长沙)

025-66051670 (南京)

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

南京销售中心: 南京市雨花台区软件大道 106 号 2 号楼 802 室

