

ADM16F03A2 数字信号处理器

ADM16F03A2QN56Q

数据手册

编号: JXDZ3.590.032SJSC

Advancechip



Electronics

湖南进芯电子科技有限公司

2024 年 07 月

V1.4

目 次

1. 产品特性	1
2. 型号命名规则	2
3. 系统概述	3
4. 引脚排布与说明	4
5. 内存映射	7
5.1 存储空间映射	7
5.2 外设空间映射	8
6. 内核说明	9
6.1 内核结构原理	9
6.2 系统寄存器	11
6.2.1 系统状态寄存器	11
6.2.2 系统控制寄存器	13
7 存储系统	16
7.1 片上存储器	16
7.2 双存取 RAM (DARAM)	16
7.3 单存取 RAM (SARAM)	16
7.4 闪存 Flash	16
7.4.1 Flash 说明	16
7.4.2 唯一 ID	17
7.4.3 Cache 说明	17
8 时钟系统	20
8.1 PLL 时钟模块	20
8.2 PLL 时钟模块控制寄存器	21
8.3 片内高精度 OSC	22
9 中断系统	23
9.1 中断系统结构	23
9.2 CPU 中断寄存器	27
9.3 外设中断控制寄存器	29
9.4 外设中断控制寄存器	33
10 看门狗	35
10.1 看门狗模块结构框图	35
10.2 看门狗控制寄存器	36
11 硬件加速单元	38
11.1 硬件加速单元功能介绍	38
11.2 硬件加速单元各模块详细说明	39

11.2.1 开平方根模块	39
11.2.2 除法器模块	41
11.2.3 Park 运算模块	44
11.2.4 乘法移位器模块	46
11.2.5 Cordic 模块	49
12 数字 I/O 口与引脚复用功能	53
12.1 数字 I/O 口与引脚复用功能描述	53
12.2 数字 I/O 控制寄存器	54
12.2.1 I/O 端口复用控制寄存器 PORTA, PORTB	54
12.2.2 I/O 端口复用控制寄存器 PORTC, PORTD	57
12.2.3 I/O 端口复用控制寄存器 PORTE, PORTF	59
13 串行通信接口 SCI	63
13.1 串行通信接口 SCI 功能简介	63
13.2 SCI 寄存器	65
14 串行外设接口 SPI 模块	71
14.1 串行外设接口 (SPI) 模块简介	71
14.2 串行外设接口 (SPI) 相关寄存器	72
15 事件管理器模块 EM1	77
15.1 事件管理器模块 EM1 简介	77
15.2 事件管理器模块 EM1 中断	78
15.2.1 事件管理器模块 EM1 中断组	78
15.2.2 事件管理器模块 (EM1) 中断控制寄存器	79
15.3 事件管理器 EM1 通用定时器	81
15.3.1 事件管理器 EM1 通用定时器控制寄存器	82
15.4 事件管理器模块 EM1 全功能比较单元	87
15.4.1 事件管理器模块 EM1 全功能比较控制寄存器	87
15.5 事件管理器模块 EM1 脉宽调制电路	89
15.5.1 PWM 特性	89
15.5.2 PWM 相关寄存器	90
15.6 事件管理器模块 (EM1) 捕获单元	90
15.6.1 事件管理器模块 (EM1) 捕获单元控制寄存器	91
15.7 正交编码脉冲(QEP) 电路	93
16 模数转换器 ADC	94
16.1 ADC 的特性	94
16.2 ADC 模块概述	95
16.3 自动排序器	97
16.3.1 顺序采样模式	100

16.3.2 不中断的自动排序模式	101
16.3.3 排序器的启/停工作模式	104
16.3.4 输入触发源	106
16.3.5 排序转换期间的中断操作	106
16.4 ADC 时钟预分频	109
16.5 ADC 校准	110
16.6 片内温度传感器	110
16.6.1 温度传感器传递函数	110
16.7 ADC 寄存器概述	112
17 运放与复用数字 IO 输入	123
17.1 运放与复用数字 IO 输入说明	123
17.2 运放与复用数字 IO 输入寄存器	125
17.2.1 通道对应的配置寄存器	125
17.2.2 OPA 与 PGA 配置寄存器	126
18 电压比较器	129
18.1 电压比较器说明	129
18.1.1 内阈值电压比较器	129
18.1.2 外阈值组电压比较器	130
18.1.3 电压比较器控制寄存器	131
19 数字延时滤波	136
19.1 数字延时滤波模块结构	136
19.2 数字延时滤波模块寄存器说明	137
19.2.1 数字延时滤波控制寄存器	137
20 LIN 收发器	139
20.1 工作模式	139
20.1.1 上电模式	139
20.1.2 待机模式	139
20.1.3 正常模式	141
20.1.4 休眠模式	141
20.1.5 正常至休眠/待机模式的状态切换	142
20.2 远程唤醒	142
20.3 显性超时功能	142
20.4 过温保护功能	142
20.5 失效安全特性	143
21 供电	144
21.1 片上 LDO 电源管理	144
22 典型应用	146

23 电气参数	148
23.1 建议的运行条件	148
23.2 建议运行温度范围内的电气特性	148
23.3 减少流耗	149
23.4 信号转换电平	149
23.5 时序参数符号	150
23.6 定时参数的通用注释	150
23.7 12 位模数转换器 (ADC)	150
23.8 ADC 建议运行条件	151
23.9 ADC 工作频率设定	151
23.10 推荐工况下的 ADC 电气特性	151
23.11 预驱模块参数	152
23.11.1 最大额定值	152
23.11.2 推荐工作条件 (无特别说明情况下, 测试条件为 $T_A=25^{\circ}\text{C}$)	153
23.11.3 静态电气参数	153
23.11.4 动态电气参数 (无特别说明情况下, $V_{BIAS}=V_{DRV}=V_B=15\text{V}$, $C_L=1000\text{pF}$, $T_A=25^{\circ}\text{C}$)	154
24 机械数据	155
25 重要注意事项及声明	157

图 清 单

图 3.1	ADM16F03A2 系统框图	3
图 4.1	ADM16F03A2 芯片 QFN56 封装引脚排布	4
图 5.1	ADM16F03A2 存储空间映射	7
图 5.2	ADM16F03A2 外设空间映射	8
图 6.1	CPU 功能模块框图	9
图 7.1	Cache 配置流程图	18
图 8.1	PLL 时钟模块功能框图	20
图 8.2	PLL 时钟旁路控制信号时序	20
图 9.1	PIE 中断硬件结构框架	25
图 10.1	看门狗功能框图	36
图 12.1	I/O 复用控制框图	54
图 13.1	SCI 功能框图	65
图 14.1	SPI 模块框图	72
图 15.1	事件管理器模块 EM1 功能框图	78
图 16.1	ADC 系统结构框图	96
图 16.2	ADC 电源引脚连接示例	97
图 16.3	双排序模式下 ADC 自动排序原理框图	98
图 16.4	级联模式下 ADC 自动排序原理框图	99
图 16.5	顺序采样模式的时序	100
图 16.6	不间断的自动排序模式 ADC 转换流程图	103
图 16.7	EM1 触发源启动排序器示例	105
图 16.8	多次排序转换之间的中断操作	109
图 16.9	ADC 时钟预分频	110
图 16.10	温度传感器传递函数	111
图 17.1	运放与数字 I/O 复用连接	123
图 17.2	PGA 等效电路	124
图 18.1	比较器模块内部框架图	129
图 18.2	内阈值组电压比较器框图	130
图 18.3	外阈值组电压比较器组框图	131
图 19.1	CAP1 数字延时滤波示意图	136
图 20.1	ADM16F03A2 芯片 LIN 模块工作模式	139

图 20. 2	正常至待机/休眠跳转时序图	140
图 20. 3	远程唤醒时序图	141
图 21. 1	芯片供电连接参考图示	145
图 22. 1	驱动模块典型应用图	146

表 清 单

表 4.1	ADM16F03A2 芯片引脚功能定义	4
表 6.1	CPU 内部硬件模块说明	10
表 8.1	SCSR1 寄存器配置 PLL 倍频系数表	21
表 8.2	OSC 的指标参数	22
表 9.1	中断源优先级与中断向量表	25
表 10.1	WD 溢出时间选择	37
表 12.1	数字 I/O 模块控制寄存器	54
表 14.1	SPI 时钟配置	74
表 15.1	针对 EM1 模块和信号名称	77
表 16.1	不同工作模式 SEQ1 和 SEQ2 比较	100
表 16.2	示例 2 中 CHSELSEQn 寄存器写入值	102
表 16.3	CHSELSEQ n 设定值 (MAX_CONV1 设为 2)	105
表 16.4	第二次自动转换会话后的值	105
表 16.5	输入触发源	106
表 16.6	CALIBRATION 校准寄存器—70B8h	110
表 16.7	温度传感器特性	111
表 16.8	ADC 相关寄存器列表	112
表 17.1	PGA 内阻 R1 与 R2 数值	125
表 17.2	PGA 放大倍数配置关系	127
表 18.1	COMP 迟滞电压配置关系	135
表 21.1	ADC 建议运行条件	151
表 21.2	ADC 工作频率范围	151
表 21.3	推荐工况下的 ADC 电气特性	151

1. 产品特性

单源供电

- 4.5V ~ 40V
- 集成 LDO
(预驱 12V, 内核 1.2V, 数字 I/O 5V、模拟 3V)

低功耗休眠模式流耗

- 20uA

高性能 16 位定点 DSP 内核

- 主频最高 150MHz
- 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

15 个通用 I/O+8 个数字输入

事件管理器 EM1

- 4 个 16 位定时/计数器
- 8 路 PWM 输出
(PWM1~6 为 3 组互补输出, 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

集成三相高、低侧半桥驱动电路

- 六路 NMOS Pre-Driver
- 栅极驱动电路高侧最高浮动绝对电压达到 200V
- 驱动能力 IO+/ IO-: +1.2A/-2.0A
@VDRV=12V, VBS=12V

•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, 可组成上、下限值保护电路, 限值可通过一对连接到比较器输入端的 DAC 配置

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 组可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(半双工)、SCI

时钟

- 10M 片内振荡器
- PLL 倍频系数 1x~15x

LIN 收发器

- 兼容 “LIN2.x/ISO 17987-4:2016(12V)/SAE J2602” 标准
- 带高压 LDO 5V 稳压源输出
- 内置过温保护功能 (热关断)
- 内置显性超时功能
- 总线限流保护功能
- 5V 稳压限流保护功能
- 稳压源输出欠压检测功能
- 极低功耗的休眠模式
- 支持总线远程唤醒
- LIN 数据传输速率可达 20kbps

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

• 温度范围: -40°C~125°C (通过 AEC-Q100)

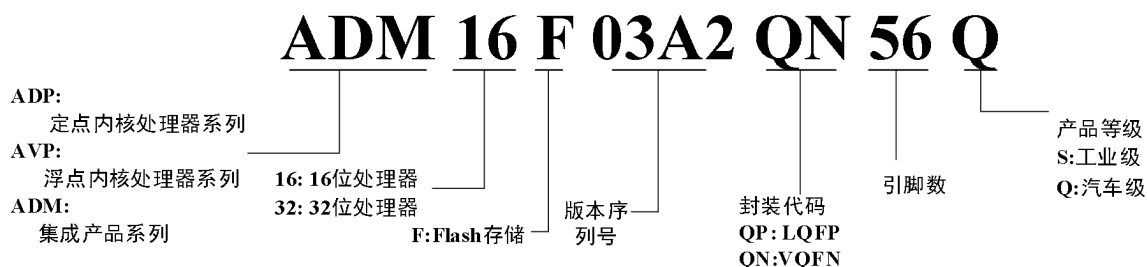
• 结温: -40°C~150°C

• 封装形式

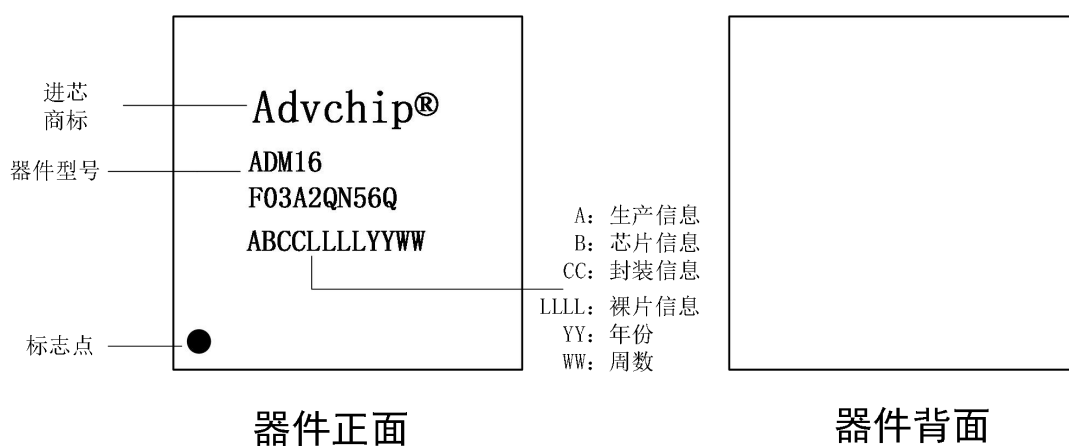
- QFN56 封装

2. 型号命名规则

ADM16F03A2QN56Q 产品命名规则:



器件标识:



3. 系统概述

ADM16F03A2 是一款面向无刷电机控制应用场景开发的 16 位增强型 DSP 器件，片上集成双 LDO，支持单电源供电模式。改进型 DSP 处理器内核与运算加速单元，同时主频可运行到 150MHz，满足多种类型电机应用算法的算力要求，并配置事件管理器、SPI 与 SCI 模块、12 位 ADC、运算放大器、电压比较器、6NMOS 预驱模块以及 LIN 收发器，有利于应用布板的极简简化。

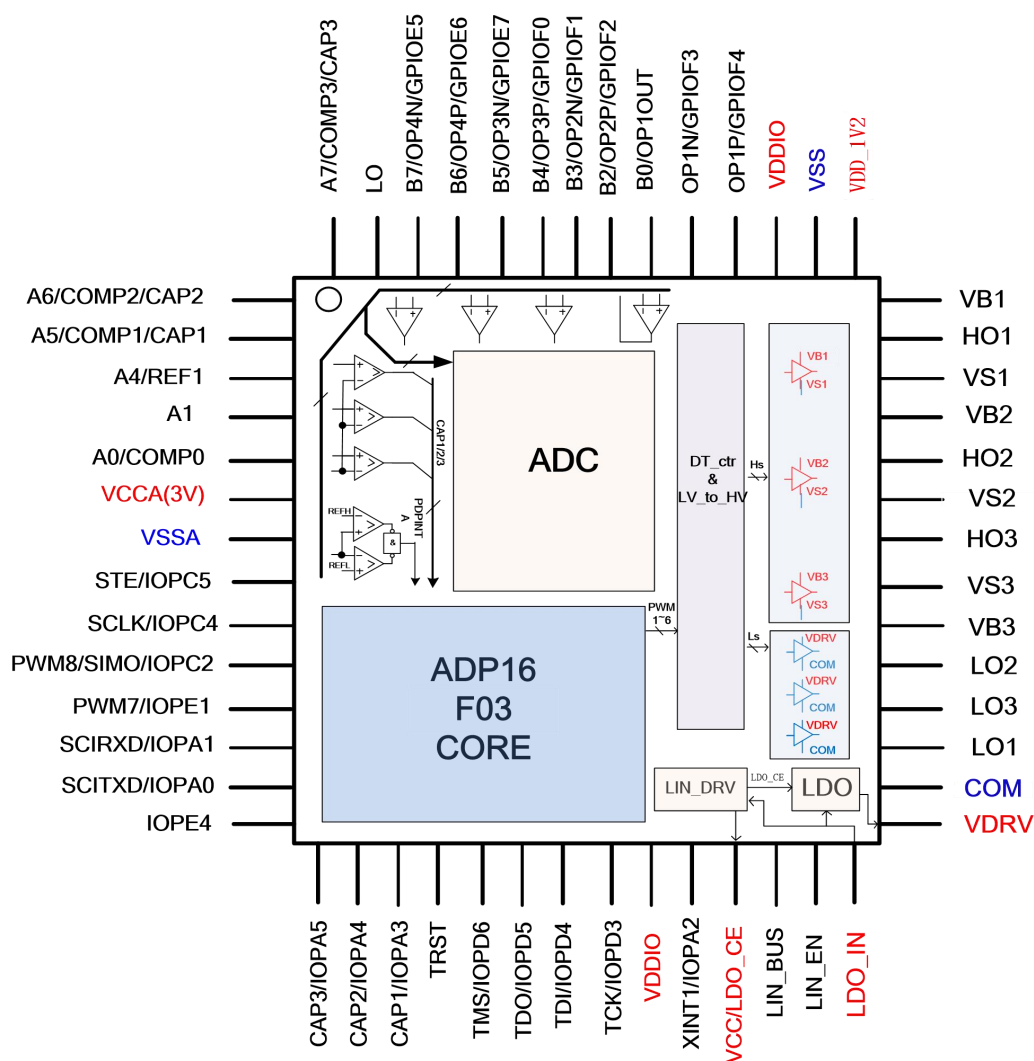
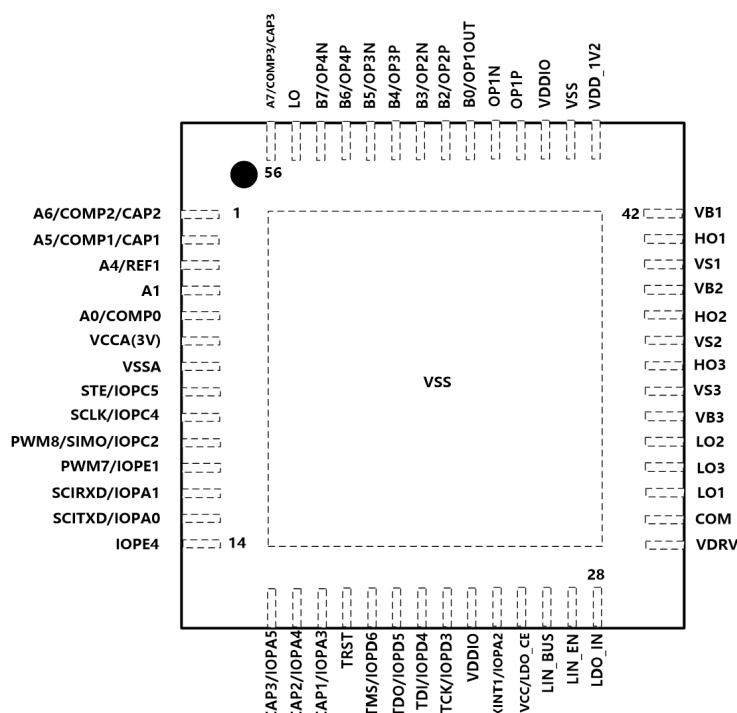


图 3.1 ADM16F03A2 系统框图

4. 引脚排布与说明



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

图 4. 1 ADM16F03A2 芯片 QFN56 封装引脚排布

表 4. 1ADM16F03A2 芯片引脚功能定义

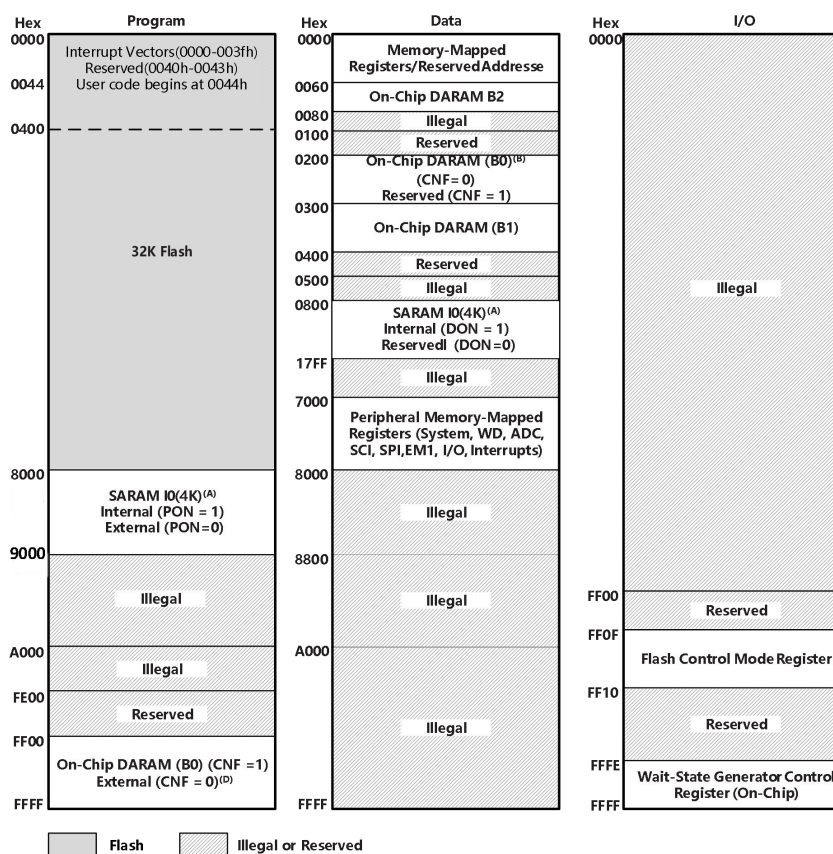
管脚	名称	类型	PU/PD/Z	功能说明
1	A6/COMP2	模拟输入	-	ADC 输入端口 A6 或比较器 2 输入端
2	A5/COMP1	模拟输入	-	ADC 输入端口 A5 或比较器 1 输入端
3	A4/ REF1	模拟输入	-	ADC 输入端口 A4 或比较器 1、2、3 公共端
4	A1	模拟输入	-	ADC 输入端口 A1
5	A0/COMP0	模拟输入	-	ADC 输入端口 A0 或比较器 0（双门限）输入端
6	VCCA	电源	-	模拟 3.0V 电源，内部 LDO 产生
7	VSSA	电源	-	模拟地
8	STE/IOPC5	数字输入/输出	PU	SPI 主机片选或 IOPC5
9	SCLK/IOPC4			SPI 时钟或 IOPC4
10	PWM8/SIMO/IOPC2			PWM8 或 SPI 主机发送或 IOPC2
11	PWM7/IOPE1	数字输入/输出	PU	PWM7 或 GPIOE1
12	SCIRXD/IOPA1	数字输入/输出	PU	SCI 接收或 GPIOA1
13	SCITXD/IOPA0			SCI 发送或 GPIOA0

管脚	名称	类型	PU/PD/Z	功能说明
14	IOPE4	数字输入/输出	PU	GPIOE4 或 Boot 引导分支控制
15	CAP3/IOPA5	数字输入/输出	PU	CAP3 或 GPIOA5
16	CAP2/IOPA4			CAP2 或 GPIOA4
17	CAP1/IOPA3			CAP1 或 GPIOA3
18	TRST	JTAG	PD	JTAG 复位
19	TMS/IOPD6		PU	JTAG 模式选择或 GPIOD6
20	TDO/IOPD5		PD	JTAG 串行输出或 GPIOD5
21	TDI/IOPD4		PU	JTAG 串行输入或 GPIOD4
22	TCK/IOPD3		PU	JTAG 时钟输入或 GPIOD3
23	VDDIO	电源	-	5.0V 数字电源
24	XINT1/IOPA2	数字输入/输出	PU	外部中断 1
25	VCC/LDO_CE	电源	-	LIN 收发器 5.0V 输出或 LDO 使能
26	LIN_BUS	12V 输入/输出	PU	LIN 总线
27	LIN_EN	数字输入	PD	LIN 收发器使能
28	LDO_IN	电源	-	LDO 输入引脚, 4.5 ~ 40V
29	VDRV	电源	-	预驱模拟电源 10 ~ 18V, 外接 10uF 或更大滤波电容, 10 ~ 18V
30	COM	电源	-	预驱模拟电源参考地, 外部需单点连接至 VSS
31	LO1	输出	-	相 1 半桥低端驱动 PWM 输出
32	LO3	输出	-	相 3 半桥低端驱动 PWM 输出
33	LO2	输出	-	相 2 半桥低端驱动 PWM 输出
34	VB3	电源	-	自举高端电源 3
35	VS3	电源	-	高端悬浮地端 3
36	HO3	输出	-	相 3 半桥高端驱动 PWM 输出
37	VS2	电源	-	相 2 半桥高端悬浮地端
38	HO2	输出	-	相 2 半桥高端驱动 PWM 输出
39	VB2	电源	-	相 2 半桥高端自举电源
40	VS1	电源	-	相 1 半桥高端悬浮地端
41	HO1	输出	-	相 1 半桥高端驱动 PWM 输出
42	VB1	电源	-	相 1 半桥高端自举电源
43	VDD_1V2	电源	-	1.2V 内核电源输出, 外接 10uF+0.1uF 电容
44	VSS	电源	-	数字地
45	VDDIO	电源	-	5.0V 数字电源
46	OP1P	模拟输入	-	运放 1 输入正端或 GPIOF4 输入
47	OP1N	模拟输入	-	运放 1 输入负端或 GPIOF3 输入
48	B0/OP1OUT	模拟输入/输出	-	ADC 输入端口 B0 或运放 1 输出端
49	B2/OP2P	模拟输入/数字输入	-	ADC 输入端口 B2 或运放 2 输入正端或 GPIOF2

管脚	名称	类型	PU/PD/Z	功能说明
				输入
50	B3/OP2N	模拟输入数字输入	-	ADC 输入端口 B3、运放 2 输入负端或 GPIOF1 输入
51	B4/OP3P	模拟输入数字输入	-	ADC 输入端口 B4、运放 3 输入正端或 GPIOF0 输入
52	B5/OP3N	模拟输入数字输入	-	ADC 输入端口 B5、运放 3 输入负端或 GPIOE7 输入
53	B6/OP4P	模拟输入数字输入	-	ADC 输入端口 B6、运放 4 输入正端或 GPIOE6 输入
54	B7/OP4N	模拟输入数字输入	-	ADC 输入端口 B7、运放 4 输入负端或 GPIOE5 输入
55	LO	电源	-	ADC 及运放参考地，连接至模拟地
56	A7/ COMP3	模拟输入	-	ADC 输入端口 A7 或比较器 3 输入端

5. 内存映射

5.1 存储空间映射



- A. 单周期访问 RAM I0, 配置 PON,DON 选择映射到程序空间或数据空间;
B. 双口 RAM B0, 通过配置 CNF 选择映射到程序空间或数据空间。

图 5. 1 ADM16F03A2 存储空间映射

5.2 外设空间映射

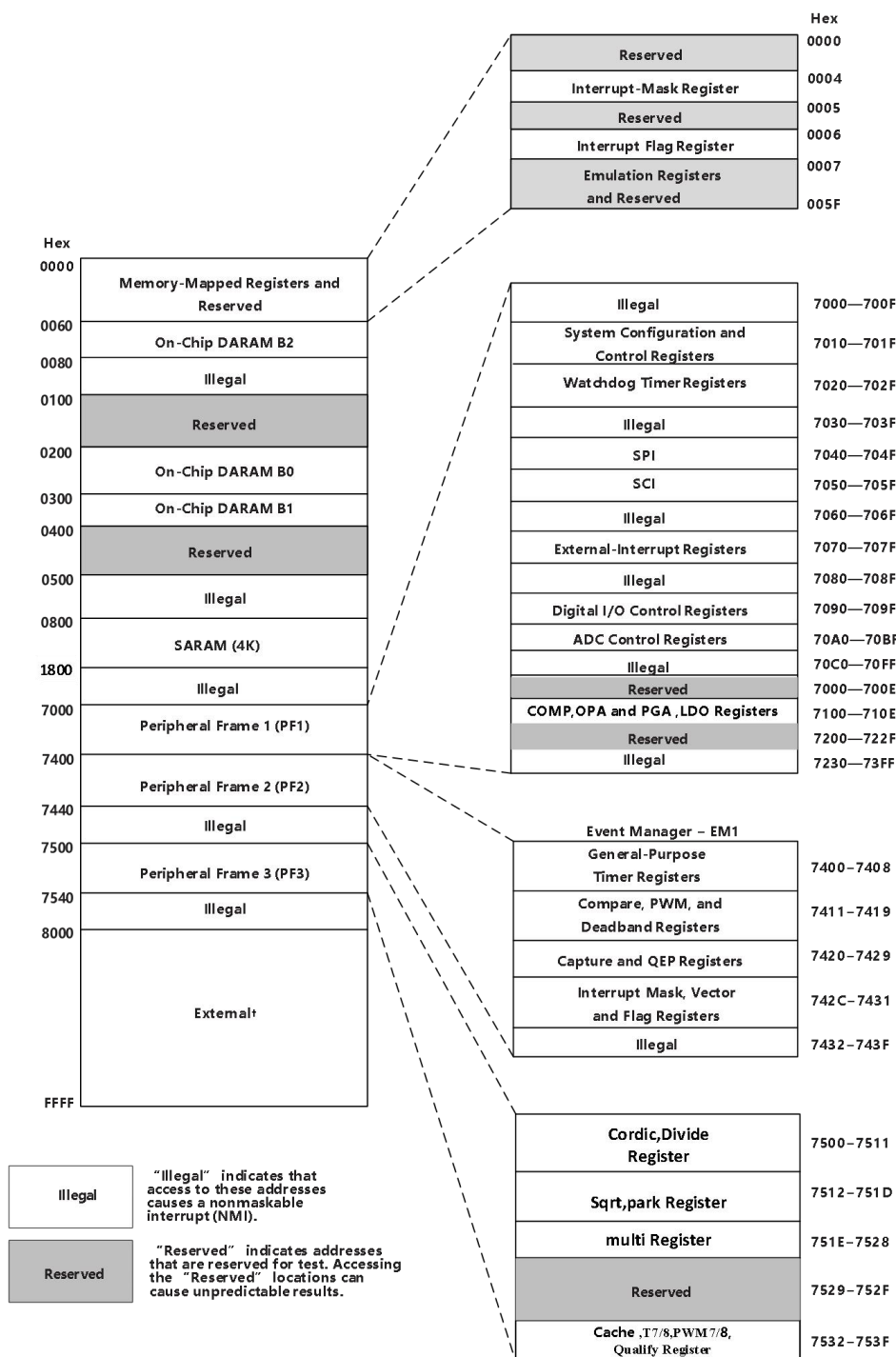


图 5.2 ADM16F03A2 外设空间映射

6. 内核说明

6.1 内核结构原理

CPU 包括一个 16 位定标移位器，一个 16 位乘 16 位并行乘法器，一个 32 位中央算术逻辑单元，一个 32 位累加器，以及用于累加器和乘法器输出的辅助移位器。CPU 的功能模块如图 6.1 所示，表 6.1 列出了 CPU 内部硬件模块的功能。

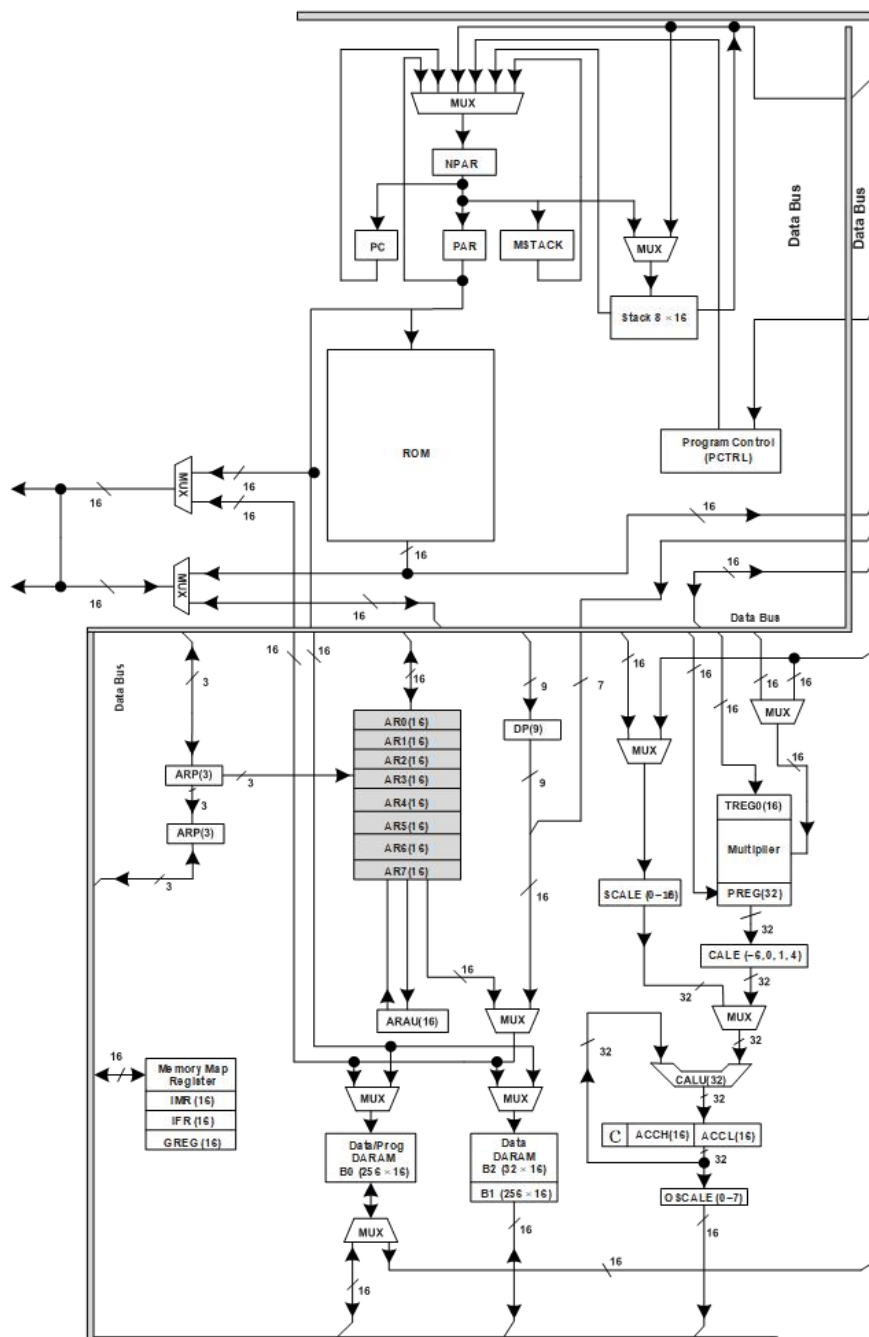


图 6.1 CPU 功能模块框图

表 6.1 CPU 内部硬件模块说明

模块符号	模块名称	模块功能描述
ACC	累加器	32 位寄存器，用于保存中央算术逻辑单元 CALU 的计算结果，并为下一次 CALU 运算提供输入，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，当使用辅助寄存器作为输入输出，用于计算间接地址。
AUX REGS	辅助寄存器 0~7	16 位寄存器用作指向任意数据空间地址的指针。这些寄存器通过 ARAU 操作，并由辅助寄存器指针 (ARP) 选定。AR0 可用作 ARx (x 为 1~7) 超过 1 个发生更新时的索引值，也可作为 ARx 的比较值。
C	进位标志位	CALU 进位输出寄存器。此进位被反馈到 CALU 进行扩展算术运算。C 进位标志位分配于状态寄存器 1 (ST1)，可通过条件指令测试。也可以用于累加器移位和循环。
CALU	中央算术逻辑单元	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 对来自 ISCALE 或 PSCALE 的数据与来自 ACC 的数据进行运算操作，并将运算后结果保存到 PCTRL 单元。
DARAM	双口 RAM	当片上 RAM 配置控制位 (CNF) 被设置为 0，B0 (DARAM) 被映射到数据空间；反之，B0 被映射到程序空间。块 B1 和 B2 被映射到数据存储器空间，地址范围为 0300h~03FFh 和 0060h~007Fh。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 设定全局数据存储空间的大小。由于在 ADM16F03A2 器件中没有使用全局内存空间，所以此寄存器功能保留。
IMR	中断屏蔽寄存器	CPU 级中断使能屏蔽寄存器。IMR 寄存器可独立配置，用于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 在单周期内执行乘法。MPY 可进行有符号或者无符号的二进制补码算术乘法。
MSTACK	微堆栈	当程序地址生成逻辑用作生成数据空间的连续地址时，MSTACK 用于临时存储下一条待取址指令的地址。
MUX	多路选择器	将多路总线复用到一个公共输入。
NPAR	下一个程序地址寄存器	NPAR 保存程序地址总线上下一个指令周期的程序地址。
OSCALE	输出数据定标寄存器	16 位到 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，并将移位后 32 位数据的高 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 在完成与当前总线周期相关的所有内存操作前期间内，保存当前在 PAB 上运行地址。

模块符号	模块名称	模块功能描述
PC	程序计数器	PC 计数器每次将 NPAR 的值增加 1，为取指操作和连续数据传送操作输出连续地址。
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，用于保存 16x16 的乘积。
PSCALE	乘积定标移位器	将乘法器乘积左移 0/1/4 位，或右移 6 位。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子程序返回地址的内存块，或者用于存储数据。ADM16x 定义堆栈为 16 位位宽，8 级深度。
TREG	临时寄存器	16 位寄存器，用于保存乘法运算中的一个操作数。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。

6.2 系统寄存器

6.2.1 系统状态寄存器

CPU 集成两个系统状态寄存器，分别为 ST0 和 ST1，包含多种条件和模式下的状态定义。这两个寄存器存储在数据地址，可从数据地址加载，并支持保存微处理器状态以及恢复子程序。

可通过状态寄存器加载指令（LST）对 ST0 和 ST1 进行写入操作。存储状态寄存器指令（SST）用于从 ST0 和 ST1 读取，（不受 LST 指令影响的 INTM 位除外）。可借助 SETC 和 CLRC 指令对其进行逐位写 1 或清 0。下文介绍了状态寄存器 ST0 和 ST1 的构成，对每个状态寄存器中包含的所有状态位进行描述。状态寄存器包含部分保留位，对保留位的读操作返回值 1。下表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

注：R=可读，W=可写，C=清除，_0=复位值；

位 15~13	ARP：辅助寄存器指针。ARP 选择用于直接寻址的 AR。ARP 加载时，ARP 中的旧值复制到 ARB 寄存器中。使用直接寻址时，ARP 可以通过存储相关指令例如 LARP，MAR，LST 指令修改。当 LST #1 指令执行时，ARP 加载与 ARB 相同的值。
---------	---

位 12	OV: 溢出标志位。当算术逻辑单元产生溢出, 溢出信号被锁存, OV 被置 1。溢出产生后, OV 将保持置位, 直到 CPU 复位、对 OV/NOV 进行 BCND/D 操作, 以及 LST 指令清除 OV 状态位。
位 11	OVM: 溢出模式位。当 OVM 设置为 0, 溢出结果在累加器上正常溢出, 当设置为 1, 溢出发生时累加器设置为最大正值或最小负值。SETC 指令和 CLRC 指令可对此位进行置位和清零操作。LST 指令也可以用来修改 OVM。
位 9	INTM: 中断模式位。当 INTM 被设置为 0 代表所有的非屏蔽中断均启用。设置为 1 代表所有的屏蔽中断均为禁用状态。INTM 由 SETC INTM 和 CLRC INTM 指令置位与复位。RS 亦可设置 INTM。INTM 对不可屏蔽的 RS 和 NMI 中断没有影响。注意, INTM 不受 LST 指令的影响。系统复位时此位被默认置 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: 数据存储器页指针。9 位 DP 寄存器与指令字的 7 个最低有效位组合成 16 个位的直接存储器地址。DP 可以通过 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_00

注: R=可读, W=可写, C=清除, _0=复位值;

位 15~位 13	ARB: 辅助寄存器指针缓存。当 ARP 被加载到 ST0 时, 旧的 ARP 值被复制到 ARB, 除非在一个 LST 指令中。当 ARB 通过一个 LST #1 指令加载时, 同样的值也将复制到 ARP 中。
位 12	CNF: 片内的 RAM 配置控制位。如果将 CNF 设置为 0, 则可重新配置双寻址 RAM 块映射到数据空间; 反之, 则映射到程序空间。CNF 可以通过 SETC CNF、CLRC CNF 和 LST #1 指令进行修改。系统复位时 CNF 为 0。
位 11	TC: 测试/控制标志位。TC 受 BIT, BITT, CMPR, LST #1 和 NORM 指令的影响。当 BIT 或 BITT 中一个测试位为 1, 或者 AR(ARP)和 AR0 之间存在 CMPR 比较测试条件, 或者当测试 NORM 指令执行 OR 函数时累加器的 2 个最高有效位为 1 时 TC 被置位为 1。条件跳转, 调用和返回指令可基于 TC 条件执行。
位 10	SXM: 符号扩展模式位。SXM=1 在数据通过定标移位器传递到累加器时产生符号扩展。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 通过 SETC SXM 指令置 1, 通过 CLRC SXM 指令复位清 0, 且可以通过 LST #1 指令加载。复位时 SXM 默认设置为 1。
位 9	C: 进位。如果加法运算结果产生一个进位, 则 C 被置为 1, 如果减法运算结果产生一个借位, 则 C 被复位为 0, 反之, 在加法后复位为 0 或减法之后被置位为 1, ADD 或带有 16 位移位的 SUB 指令除外。通常情况下, ADD 只能置位进位, SUB 只能复位进位, 但不能影响进位, 单 bit 位的移位操作循环移位指令以及 SETC, CLRC 和 LST #1 指令均会影响 C 的状态, 复位时 C 被置位为 1。
位 1~位 0	PM: 乘积移位方式 PM=00, 乘法器的 32 位乘积结果直接载入 CALU, 不移位 PM=01, PREG 输出左移 1 位, 最低位 (LSB) 以 0 填充后载入 CALU PM=10, PREG 输出左移 4 位, 最低位 (LSB) 以 0 填充后载入 CALU PM=11, PREG 输出右移 6 位, 并进行符号扩展 注意: PREG 中的内容一直保持不变。当把 PREG 中的内容传送到 ALU 单元时, 移位操作

发生，PM 可由 SPM 指令和 LST #1 指令加载，复位时 PM 清 0。

6.2.2 系统控制寄存器

1. 系统控制寄存器 SCSR1—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，写入 1 代表 bypass 控制使能，默认开启，写入 0 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]是逻辑 1，则表示 PLL 已锁定，反之则表示 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； 详见表 8-1 通过 SCSR1 (0x7018) 寄存器选择 PLL 时钟
位 7	ADCCLK_EN: ADC 模块时钟使能控制位； 0: 禁用 ADC 模块时钟（即：关断 ADC 模块，以降低功耗）； 1: 使能 ADC 模块时钟；
位 6	SCICLK_EN: SCI 模块时钟使能控制位； 0: 禁用 SCI 模块时钟（即：关断 SCI 模块，以降低功耗）； 1: 使能 SCI 模块时钟；
位 5	SPICLK_EN: SPI 模块时钟使能控制位； 0: 禁用 SPI 模块时钟（即：关断 SPI 模块，以降低功耗）； 1: 使能 SPI 模块时钟；
位 4	Auxiliary CLKEN 0: 辅助寄存器 0 时钟使能控制位，辅助寄存器 0 包括运放和电

	压比较器模块的寄存器; 0: 禁用辅助寄存器 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, Cache 和数字延时滤波模块的寄存器, 加速运算单元的寄存器; 0: 禁用辅助寄存器 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: 禁用 EM1 模块时钟 (即: 关断 EM1 模块, 以降低功耗); 1: 使能 EM1 模块时钟;
位 1	保留
位 0	ILLDR: 无效地址检测位;

2. 系统控制寄存器 SCSR2—7019h

15	14	13	12	11	10	9	8
保留位							
RW_0							
7	6	5	4	3	2	1	0
保留位	保留位	WD OVERRIDE	保留位	保留位	保留位	DON	PON
RW_0	RW_0	RC_1	RW_0	RW_1	RW_0	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, 复位时默认值为 1, 允许用户通过软件将 WDCR 寄存器中的 WDDIS 位置 1 禁止 WD 工作, 该位是一个只能清 0 位, 通过向该位写 1 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	SARAM 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 被映射到片上程序空间;
	1	0	SARAM 被映射到片上数据空间;
	1	1	SARAM 被同时映射到片上程序空间和数据空间;

7 存储系统

7.1 片上存储器

该系列器件配置如下内存模块：

- 4K x 16 位 SARAM
- (256+256+32) 544x 16 位 DARAM
- 32K x 16 位 Flash

7.2 双存取 RAM (DARAM)

该系列器件上包含 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，并且这两个块仅映隐射到数据存储空间。块 0 包含 256 个字，可以配置为数据空间或程序空间。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) 指令可用作允许通过软件动态配置存储器映射。

代码在 RAM 中运行时可达到 CPU 主频速率，无需配置等待状态。DARAM 允许在一个周期内执行两次访问。

7.3 单存取 RAM (SARAM)

该系列器件包含一块 4Kx16 位 I/O SARAM，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 位配置详细信息，可请参考见 SCSR2 寄存器描述表。复位时，PON 和 DON 为 11，片上 SARAM 同时映射到程序空间和数据空间。

7.4 闪存 Flash

7.4.1 Flash 说明

该系列器件集成一块片上闪存，大小为 32K*16bit，映射的程序地址段为 0000h ~ 7FFFh。其中，中断向量表空间 0000h ~ 003Fh，保留地址段 0040h ~ 0043h 为代码安全密码。

闪存空间被分为四个扇区，扇区 0 大小为 4K*16bit，映射到地址段 0000h ~ 0FFFh，扇区 1 大小为 12K*16bit，映射到地址段 1000h ~ 3FFFh，扇区 2 大小为 12K*16bit，映射地址段 4000h ~ 6FFFh，扇区 3 大小为 4K*16bit，映射到地址段 7000h ~ 7FFFh。

7.4.2 唯一 ID

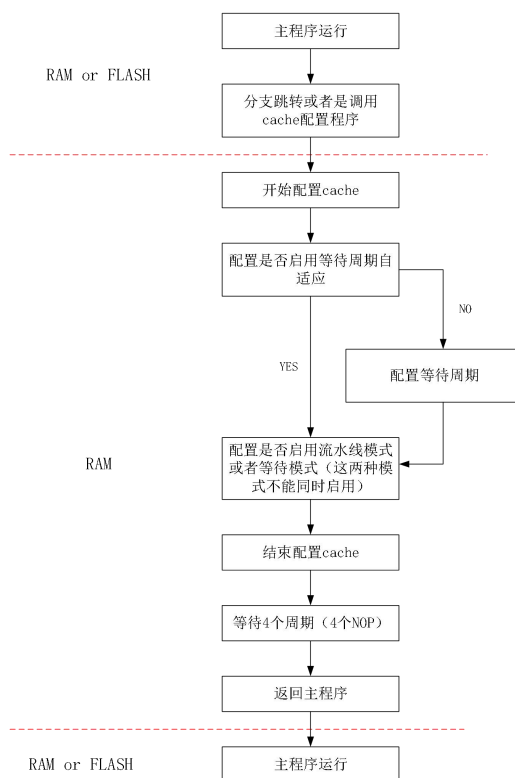
该系列器件设有唯一 ID 编码。器件完成上电完成后唯一 ID 编码被加载到寄存器 752Dh, 752Ch, 752Bh, 752Ah。这些寄存器均为只读寄存器。

7.4.3 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根据系统时钟频率自动调节
- 3) 缓存容量：64bit
- 4) Cache 功能可软件关闭
- 5) 等待模式时，每当高位地址（最高位到 bit2 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（最高位到 bit2 位）变化不连续时（地址不连续），等待相应时钟周期数（WaitNum）。

2. 配置流程



备注:

- A. 禁止在 Flash 中运行 Cache 的配置代码;
- B. 配置过程应遵循上述图示流程;
- C. 系统时钟周期高于 45ns, 无需启用 Cache;
- D. 等待周期计算方法:
WaitNum= (45ns) /系统时钟周期; 去尾法取整;
- E. 要使用 Cache 必须开启 PLL 倍频锁定之前 bypass 功能。

图 7. 1 Cache 配置流程图

3. Cache 控制寄存器—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	WaitMode En
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 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;
位 12~位 10	保留
位 9	PipeEn: 流水线模式使能位: 1: 启用流水线模式; 0: 关闭流水线模式;
位 8	WaitModeEn: 等待模式使能位: 1: 启用等待模式; 0: 关闭等待模式;
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/系统时钟周期); 去尾法取整;

- 1) Cache 控制寄存器在使用前应先使能辅助寄存器 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。

8 时钟系统

8.1 PLL 时钟模块

该系列器件集成一个片载、基于 PLL 的时钟模块。此模块为器件提供所有需要的时钟信号，以及控制器件进入低功耗模式。PLL 通过 4 个 bit 位 SCSR1[11:8]，CLK_NS3，CLK_NS2，CLK_NS1，CLK_NS0 控制和选择不同的 CPU 时钟频率。

PLL 时钟控制增加了 PLL 时钟在倍频锁定之前的旁路功能，以保证内核稳定工作，其中 SCSR1[15]在读、写具有不同的含义，写操作代表开启 PLL 倍频锁定之前 bypass 控制的使能位，写入 1 开启，默认开启，写入 0 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]是逻辑 1，则表示 PLL 已锁定，反之则表示 PLL 还未锁定。

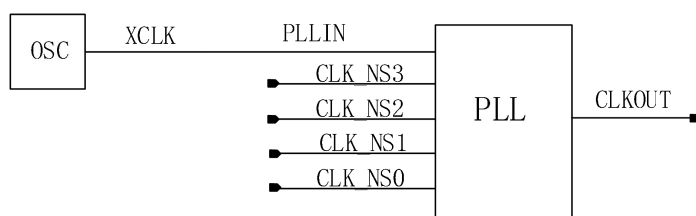


图 8.1 PLL 时钟模块功能框图

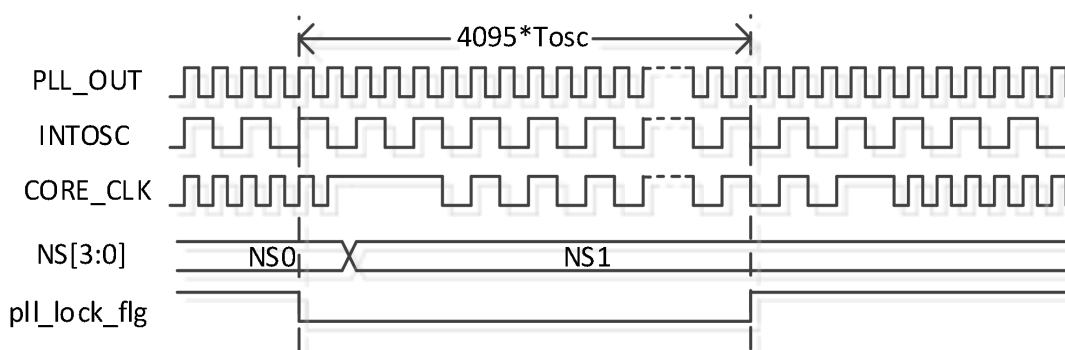


图 8.2 PLL 时钟旁路控制信号时序

表 8.1 SCSR1 寄存器配置 PLL 倍频系数表

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

8.2 PLL 时钟模块控制寄存器

1. 系统控制寄存器 SCSR1—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

注：R=可读，W=可写，C=清除，_0=复位值；

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

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，“1”有效，默认开启，若是“0”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“1”，则表示 PLL 已锁定，若是“0”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； 详细配置参考表 8-1 通过 SCSR1 (0x7018) 寄存器选择 PLL 时钟
位 7	ADCCLK_EN: ADC 模块时钟使能控制位； 0: 禁用 ADC 模块时钟（关断 ADC 模块时钟可降低功耗）； 1: 使能 ADC 模块时钟；
位 6	SCICLK_EN: SCI 模块时钟使能控制位；

	0: 禁用 SCI 模块时钟（关断 SCI 模块时钟可降低功耗）； 1: 使能 SCI 模块时钟；
位 5	SPICLK_EN: SPI 模块时钟使能控制位； 0: 禁用 SPI 模块时钟（关断 SPI 模块时钟可降低功耗）； 1: 使能 SPI 模块时钟；
位 4	Auxiliary CLKEN 0: 辅助寄存器 0 时钟使能控制位，辅助寄存器 0 包括运放相关寄存器与电压比较相关寄存器； 0: 禁用辅助寄存器 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 包含 T7、T8、PWM7、PWM8, Cache 模块、数字延时滤波以及加速运算单元寄存器； 0: 禁用辅助寄存器 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: 禁用 EM1 模块时钟（关断 EM1 模块时钟可降低功耗）； 1: 使能 EM1 模块时钟；
位 1	保留
位 0	ILLDR: 无效地址检测位；

8.3 片内高精度 OSC

该系列器件集成高精度无引脚 OSC 模块，为 PLL 模块提供时钟源，无需片外晶振。

OSC 的典型输出频率为 10MHz，其指标参数如表 8.2 所示：

表 8.2 OSC 的指标参数

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

9 中断系统

9.1 中断系统结构

该系列器件集成软件可编程中断，结构灵活且支持片上和外部中断配置，以满足高实时应用中的中断响应需求。可识别的中断源包括下述三种类型。

- 复位（由硬件或软件触发）不受 CPU 的限制，并立即优先于任意其它执行功能。所有可屏蔽的中断都被禁用，直到复位后用户程序启用。

器件复位源：一个看门狗定时器超时（复位）。

- 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：
 - 外部中断请求由 XINT1、XINT2 和电压比较器（输入到 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 内核任一可屏蔽中断线。
 - 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特定事件的使能位以及 CPU 的 IMR 来屏蔽，IMR 寄存器可以屏蔽 DSP 内核任一可屏蔽中断线。
- 该系列器件的软件产生的中断包括：
 - INTR 指令。该指令允许用软件初始化任一中断。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）。
 - 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。该系列器件未配置 NMI 硬件信号，仅支持软件触发。
 - TRAP 指令。该指令迫使 CPU 分支到指向位置 22H 的中断向量。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 TRAP 中断服务

程序时，仍可触发响应其它可屏蔽硬件中断。

- 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) 前级通过外设中断扩展(PIE)进行控制，PIE 管理来自外设级中断，并分组到六个内核中断。PIE 中断硬件结构框架如图 9-1 所示，表 9.1 列举出 PIE 与 CPU 中断向量对应关系。

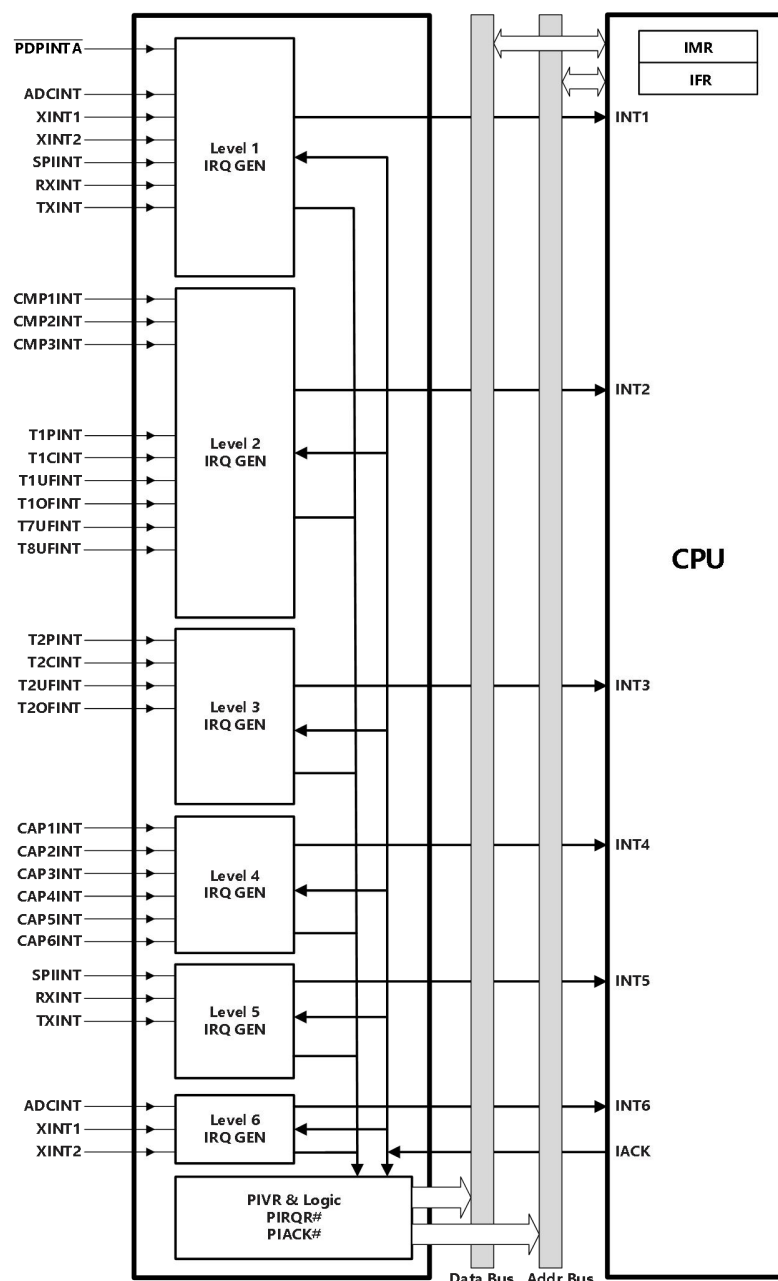


图 9.1 PIE 中断硬件结构框架

表 9.1 中断源优先级与中断向量表

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)	中断是否可屏蔽	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 下溢中断
T8UFINT	19		2.7	0032h	是	EM1	定时器 8 下溢中断
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 000Ah	1.7	0005h	是	SPI	SPI 中断 (低优先级)
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	INT6 000Ch	1.12	0004h	是	ADC	ADC 中断 (低优先级模式)
XINT1	31		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
伪中断向量	N / A	N / A	--	0000h	N / A	中央处理器	伪中断向量
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	

9.2 CPU 中断寄存器

CPU 中断寄存器包括：中断标志寄存器 (IFR) 和中断屏蔽寄存器 (IMR)

1. CPU 中断标志寄存器 IFR—0006h

15~6	5	4	3	2	1	0
保留位	INT6 flag	INT5 flag	INT4 flag	INT3 flag	INT2 flag	INT1 flag
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，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 的中断挂起；

	1	: 至少一个 INT6 的中断挂起, 对该位写 1 可清除此中断请求状态。	
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 的中断挂起; 1	: 至少一个 INT5 的中断挂起, 对该位写 1 可清除此中断请求状态。
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 的中断挂起; 1	: 至少一个 INT4 的中断挂起, 对该位写 1 可清除此中断请求状态。
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 的中断挂起; 1	: 至少一个 INT3 的中断挂起, 对该位写 1 可清除此中断请求状态。
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 的中断挂起; 1	: 至少一个 INT2 的中断悬挂, 对该位写 1 可清除此中断请求状态。
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 的中断挂起; 1	: 至少一个 INT1 的中断挂起, 对该位写 1 可清除此中断请求状态。

2. CPU 中断屏蔽寄存器 IMR—0004h

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

9.3 外设中断控制寄存器

外设中断寄存器包括:

外设中断向量寄存器;

外设中断请求寄存器 0/1/2 (PIRQR0/1/2);

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使用, 用户可忽略。

1. 外设中断向量寄存器 PIVR—701Eh

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
V15	V14	V13	V12	V11	V10	V9	V8	V7	V6	V5	V4	V3	V2	V1	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 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:

- 1) 写入 1 会发出一个中断请求到 DSP 内核, 写入 0 无影响。
- 2) 有中断请求而未被响应, 称为挂起。

位 15~0	IRQ0.0~IRQ0.15: 0 : 未挂起中断请求 1 : 挂起中断请求
--------	--

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 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:

- 1) 写入 1 会发出一个中断请求到 DSP 内核, 写入 0 无影响。
- 2) 有中断请求而未被响应, 称为挂起。

位 15~0	IRQ1.0~IRQ1.15: 0 : 未挂起中断请求 1 : 挂起中断请求
--------	--

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 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	保留
RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0	RW_0

注:

- 1) 写入 1 会发出一个中断请求到 DSP 内核, 写入 0 无影响。
- 2) 有中断请求而未被响应, 称为挂起。

位 15~0	IRQ2.0~IRQ2.15: 0 : 未挂起的中断请求 1 : 挂起中断请求
--------	---

5. 外设中断应答寄存器 PIACKR0—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 到对应的中断应答位可以清零。
--------	---

注: 对该寄存器写 1 插入中断应答并不更新 PIVR 寄存器的内容, 读此寄存器返回值是 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 PIACKR1—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 到对应的中断应答位可以清除此位。
--------	---

注：对该寄存器写 1 插入中断应答并不更新 PIVR 寄存器的内容，读此寄存器返回值是 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	T2PINT	Timer2 周期中断	INT3
IAK1.1	T2CINT	Timer2 比较中断	INT3
IAK1.2	T2UFINT	Timer2 下溢中断	INT3
IAK1.3	T2OFINT	Timer2 上溢中断	INT3
IAK1.4	CAP1INT	Capture1 中断	INT4
IAK1.5	CAP2INT	Captuer2 中断	INT4
IAK1.6	CAP3INT	Capture3 中断	INT4
IAK1.7	SPIINT	低优先级模式的 SPI 中断	INT5

IAK1.8	RXINT	低优先级模式的 SCI 接收中断	INT5
IAK1.9	TXINT	低优先级模式的 SCI 发送中断	INT5
保留			
保留			
IAK1.12	ADCINT	低优先级模式的 ADC 中断	INT6
IAK1.13	XINT1	低优先级模式的外部引脚 1 中断	INT6
IAK1.14	XINT2	低优先级模式的外部引脚 2 中断	INT6
保留			

7. 外设中断应答寄存器 PIACKR2—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	保留
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 到对应的中断应答位可以清除此位。
--------	---

注：对该寄存器写 1 插入中断应答并不更新 PIVR 寄存器的内容，读此寄存器返回值是 0。

9.4 外设中断控制寄存器

该系列器件包含两个外部中断控制寄存器 XINT1CR 和 XINT2CR，用于控制和监控 XINT1 和 XINT2 的引脚状态。

1. 外部中断寄存器 XINT1CR—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, 该位决定中断触发来自上升沿还是下降沿; 0 : 在下降沿产生中断 (由高到低跳变); 1 : 在上升沿产生中断 (由低到高跳变);
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, 可查阅表 9-1 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—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 2 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 9-1 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;

10 看门狗

10.1 看门狗模块结构框图

该系列器件集成片内看门狗（Watch Dog）定时器模块。如果器件运行期间没有定期通过软件写入正确的密钥来监视软、硬件的操作，该模块生成一个系统复位信号。看门狗定时器独立于 CPU 工作，无需 CPU 初始化。当发生系统复位时，看门狗定时器默认设置为最高速率（看门狗时钟信号配置为 CLKOUT / 512）。器件退出复位态，CPU 开始执行代码，看门狗开始递增计数。为了避免器件上电后触发看门狗超时复位，看门狗的设置应该在上电复位后尽快完成。有关看门狗模块功能结构如图 10.1 所示。看门狗功能模块主要包括以下部分：

- 看门狗定时器
- 7 种不同看门狗溢出速率
- 一个密钥寄存器（WDKEY），写入正确值清除看门狗计数器，写入错误值触发系统复位
- 检测对应的控制位，当写入一个错误值到 WDCR 寄存器，将触发系统复位
- 一旦系统复位解除，看门狗计数器自动激活
- 三个看门狗控制寄存器。

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。对高位字节写入无效。

表 10.1 列出了不同的看门狗溢出速率配置，当 SCSR2 寄存器的第 5 位 (SCSR2.5) 为 1 时，WDCR 寄存器的第 6 位 (WDCR.6) 写 “1” 可以禁用看门狗。如果 SCSR2.5 为 0，看门狗将不会被禁用。SCSR2.5 相当于器件的 WDDIS 引脚。

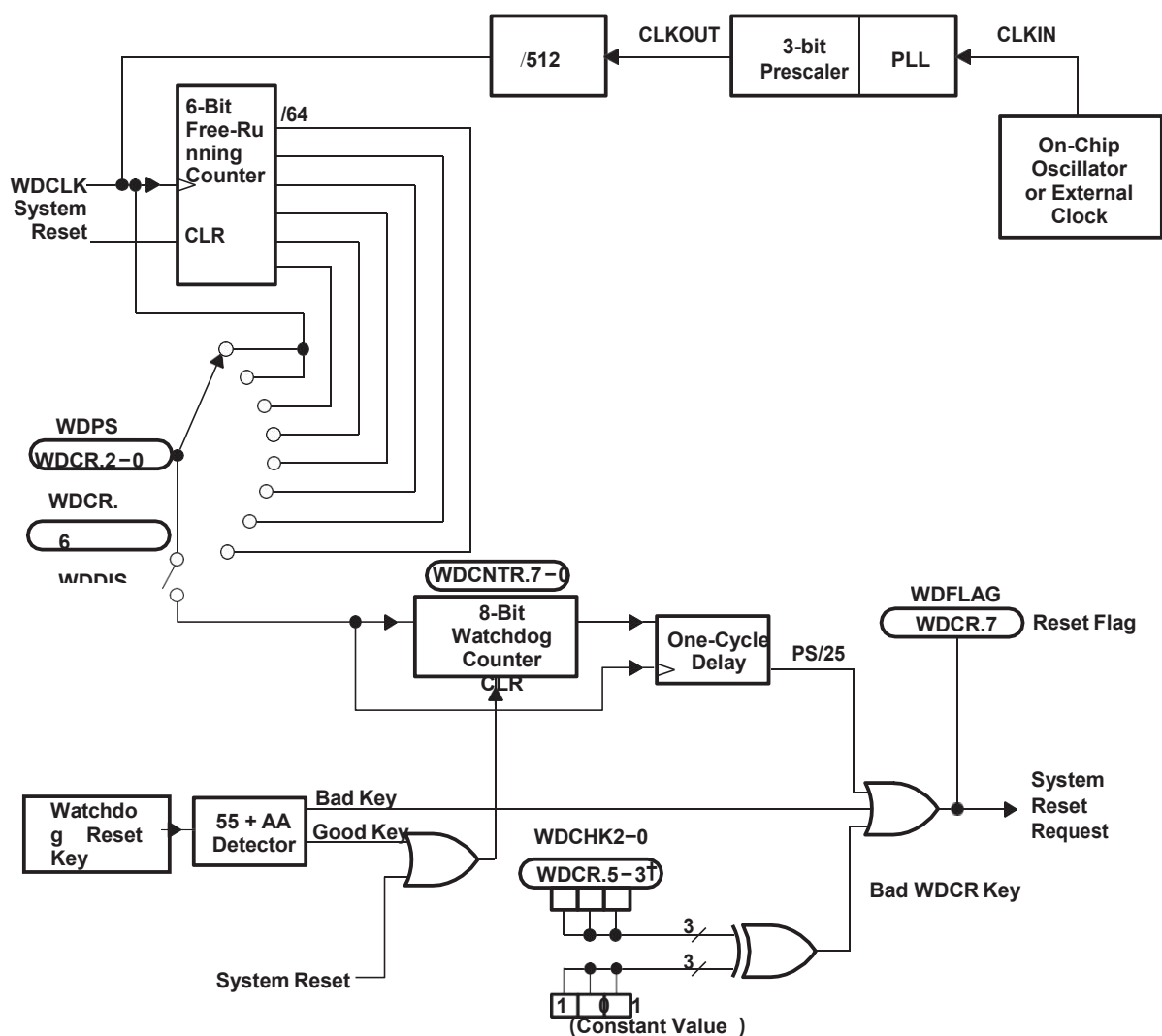


图 10.1 看门狗功能框图

10.2 看门狗控制寄存器

1. 看门狗计数寄存器 WDCNTR—7023h

8 位看门狗计数寄存器存放看门狗当前的计数值，寄存器的 D7~D0 位。WDCNTR 是一个只读寄存器，复位后为 0，对寄存器写入无效。

2. 看门狗密钥寄存器 WDKEY—7025h

顺序写入 55h+AAh（复位关键字）到 WDKEY 时，将复位看门狗计数器的值，其它写入值的组合或除 55h、AAh 之外写入值都不能清除 WDCNTR；

8 位 WDKEY 是一个可读写寄存器，复位后为 0。读取该寄存器将返回 WDCR 中内容。

3. 看门狗控制寄存器 WDCR—7029h

8 位 WDCR 用作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。该位用作判断看门狗定时器是否产生一个看门狗复位。看门狗产生的复位事件时此位置 1。 0：表明上次清除该位后，WD 定时器没有产生复位； 1：表明上次清除该位后，WD 定时器已经产生了复位；
位 6	WDDIS：禁止看门狗位。仅当 SCSR2 中的 WDOVERRIDE 位为 1 时，向该位写入有效； 0：使能看门狗； 1：禁止看门狗。
位 5~位 3	WDCHK2~WDCHK0：看门狗检查位。必须向此三位写 101h，系统才继续正常运行。否则将触发一个系统复位。读这三位返回 000h；
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 9-1 列出了 WD 溢出时间选择，WD 计数器在溢出前计数 257 个时钟，给定最小溢出时间。）

表 10.1 WD 溢出时间选择

WD 频选择 BITS			WDCLK 分频	看门狗时钟速率 f	最小溢出时间
WDPS2	WDPS1	WDPS0		频率 (Hz)	时间 (s)
0	0	X	1	WDCLK / 1	(256*1)/WDCLK
0	1	0	2	WDCLK / 2	(256*2)/WDCLK
0	1	1	4	WDCLK / 4	(256*4)/WDCLK
1	0	0	8	WDCLK / 8	(256*8)/WDCLK
1	0	1	16	WDCLK / 16	(256*16)/WDCLK
1	1	0	32	WDCLK / 32	(256*32)/WDCLK
1	1	1	64	WDCLK / 64	(256*64)/WDCLK

11 硬件加速单元

11.1 硬件加速单元功能介绍

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 的无符号整数，运算结果为 16bit 无符号整数；

硬件运算花销 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 有符号整数；

硬件运算花销 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

移位后的低 32bit 数据；

硬件运算花销 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；

计算公式：

$$d = \cos(\theta) * \alpha + \sin(\theta) * \beta$$

$$q = -\sin(\theta) * \alpha + \cos(\theta) * \beta$$

硬件运算花销 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;

硬件运算花销 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) 运算单元集成结果溢出检测、运算中断错误检测等功能;

(8) 各功能模块皆可软复位。

11.2 硬件加速单元各模块详细说明

11.2.1 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 再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—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—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: 开方运算的输入高 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—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(写入 0 复位)；
位 1	Sqrt_cal_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（写入 1 清除）；

(4) 开方模块状态寄存器 r_sqrt_s—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—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;
----------	-------------------------------------

11.2.2 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, 则为手动开启模式, 此时, 只要将 div_ctrl[1]先设置为 1, 再设置为 0, 则立即开启除法运算,

当 div_ctrl[2]设置为 0, 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, 用作保存除法器模块 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，用作保存除法器模块 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: 除数低 16bit 数据寄存器，用作保存 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，用作保存除法器模块 32 位除数的高 16 位数据；
----------	--

(5) 除法控制寄存器 r_div_ctrl—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，0 有效；
位 3	r_div_en: 除法器模块使能信号，1 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; 先写入 1, 再写入 0 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—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

15~0
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 15~位 0	r_div_ql: 除法运算商的低 16bit 数据寄存器, 用作保存除法运算 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

15~0
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 15~位 0	r_div_qh: 除法运算商的高 16bit 数据寄存器, 用作保存除法运算 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

15~0

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 15~位 0

r_div_rl: 除法运算余数低 16bit 数据寄存器，用作保存除法器模块 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

15~0

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 15~位 0

r_div_rh: 除法运算余数高 16bit 数据寄存器，用作保存除法器模块 32 位余数的高 16 位数据；

11.2.3 Park 运算模块

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—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，用作保存 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 - 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，用作保存 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 - 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，用作保存 park 模块输入数据 theta 的值；
----------	--

(4) park 变换控制寄存器 r_park_ctrl—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，写 0 有效
位 1	park_cal_en: park 运算使能信号，写 1 有效
位 0	r_st_clr: 状态寄存器清除信号，写 1 有效

(5) park 变换状态寄存器 r_park_st—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—751Ch

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, 用作保存 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—751Dh

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_mtd: park 变换结果 mtq 数据寄存器, 用作保存 park 模块运算结果 mtq 的数据。
----------	--

11.2.4 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, 操作如下:

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]写入 1 使能手动开启模式, 之后在需要运算的时候将 r_multi_ctrl[1]顺序写入 1 和 0, 触发一次运算。

(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 1, 表示读取结束。

2. 乘法移位器寄存器列表

(1) 乘法运算输入低 16 位数据寄存器 r_multi_al—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, 用作保存输入变量 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—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, 用作保存输入变量 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—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, 用作保存输入变量 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—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, 用作保存输入变量 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—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—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, 写 0 有效
位 3	r_multi_en: 乘法移位去模块使能信号, 写 1 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; 顺序写入 1, 0 将开启一次运算
位 0	r_st_clr: 状态寄存器 r_multi_st 清除信号, 写入 1 清除

(7) 乘法移位器状态寄存器 r_multi_st—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—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，用作保存运算结果 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—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，用作保存运算结果 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—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，用作保存运算结果 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—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，用作保存运算结果 64 位变量的 48~63 位数据；
----------	---

11.2.5 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，则为手动开启模式，此时，只要将 r_cor_ctrl[3] 先设置为

1, 再设置为 0, 则立即开启角度运算。

当 r_cor_ctrl[1] 设置为 0, 则为自动开启模式, 此时只要将坐标 (X, Y) 写入值, 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。

- N 模式下: 绝对误差 $< \pm 0.5^\circ$ 。
- 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—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, 用作保存输入变量 x 的数据;
----------	--

(2) 坐标 (x, y) 中的 y 输入变量寄存器 r_cor_y—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, 用作保存输入变量 y 的数据;
----------	--

(3) Cordic 运算结果 (Q&N 模式) 寄存器 r_agl—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, 用作保存运算之后得到的角度;
----------	---

(4) cordic 控制寄存器 r_cor_ctrl—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，0 有效
位 7~位 4	r_pre_amp: 表示对坐标 (x, y) 的倍乘 $2^{r_pre_amp}$
位 3	r_hand_ctrl: 手动模式下的手动开启控制信号，先置 1 再置 0 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，写入 1 清除

(5) cordic 模块状态寄存器 r_cor_st—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—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, 用作设置 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—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, 用作设置 y 预偏置大小
----------	--

12 数字 I/O 口与引脚复用功能

12.1 数字 I/O 口与引脚复用功能描述

该系列器件设计集成 15 个通用双向数字 I/O (GPIO) 引脚和 8 个数字输入引脚, 其中大部分为外设功能和通用 I/O 复用引脚。这些引脚功能通过 10 个 16 位寄存器控制。这些寄存器分为两种类型:

- 三个输出控制寄存器: MCRA、MCRB、MCRC, 用于配置选择引脚的外设功能或通用 I/O 功能。
- 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 引脚数据和输出方向。

12.1.1 复用 I/O 引脚描述

通用 I/O 引脚的控制结构如图 12.1 所示, 其中每个引脚都有三类控制位对其定义:

- Mux 控制位, 在引脚的主功能 (1) 和 I/O 功能 (0) 之间实现选择控制。
- I/O 方向位, 如果引脚选择了 I/O 功能 (多路复用控制位写入 0), 此位选择引脚模式是输入 (0) 还是输出 (1)。
- I/O 数据位, 如果引脚选择了 I/O 功能 (多路复用控制位写入 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。

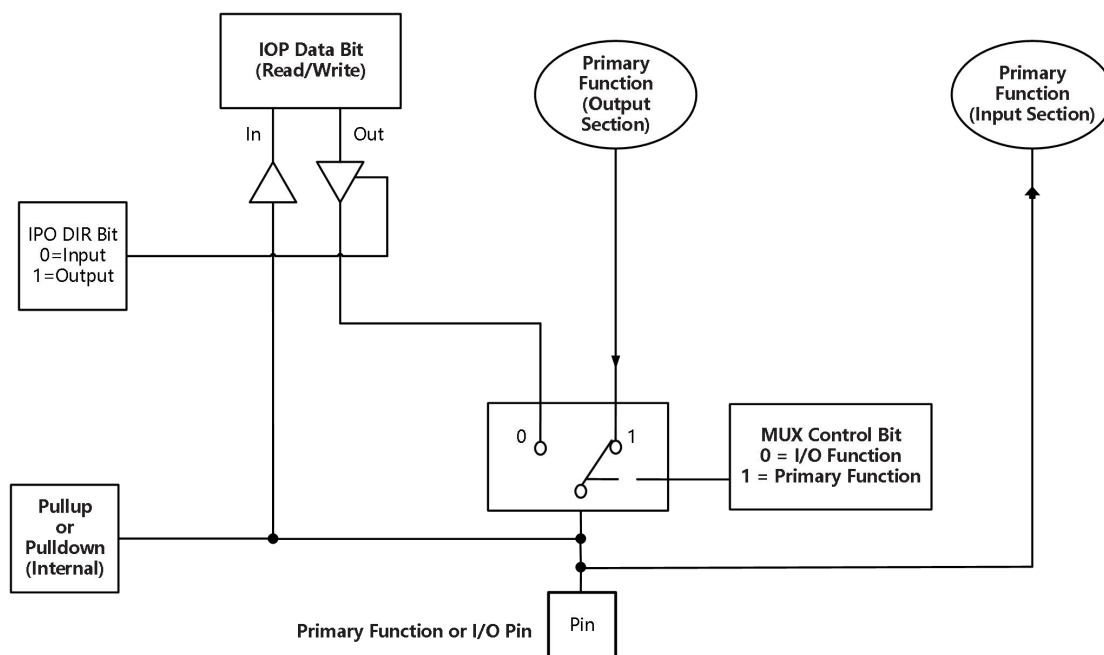


图 12.1 I/O 复用控制框图

12.2 数字 I/O 控制寄存器

表12.1列出了数字I / O模块中可用的寄存器，这些寄存器被映射到数据空间。

表 12.1 数字 I/O 模块控制寄存器

数字 I / O 控制寄存器的地址	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

12.2.1 I/O 端口复用控制寄存器 PORTA, PORTB

1) I/O 端口复用输出控制寄存器 MCRA - 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 端口复用输出控制寄存器 MCRB - 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

3) I/O 端口 A 数据和方向控制寄存器 PADATDIR - 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

4) I/O 端口 B 数据和方向控制寄存器 PBDATDIR - 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

12.2.2 I/O 端口复用控制寄存器 PORTC, PORTD

1) I/O 端口复用输出控制寄存器 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 C 数据和方向控制寄存器 PCDATDIR - 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 - 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

12.2.3 I/O 端口复用控制寄存器 PORTE, PORTF

1) I/O 端口 E 数据和方向控制寄存器 PEDATDIR - 7095h

注：GPIOE5~GPIOE7 配置输出端口无效

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

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

注: GPIOF0~GPIOF4 配置输出端口无效

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

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

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

仅支持输入功能, 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_INPU T_CT7	REG_INPU T_CT6	REG_INPU T_CT5	REG_INPU T_CT4	REG_INPU T_CT3	REG_INPU T_CT2	REG_INP UT_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 - 7096h

注: GPIOF0~GPIOF4 配置输出端口无效

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

13 串行通信接口 SCI

13.1 串行通信接口 SCI 功能简介

该系列器件集成一个串行通信接口 (SCI) 模块, 串口通信模块的寄存器的位宽为 8 位。SCI 支持 CPU 和其它使用标准 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, 具备独立使能和中断位, 两部分可独立工作, 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校验, 如帧中断校验、奇偶性、超限和帧错误校验等。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配置, 支持超过 65000 种的不同速率。SCI 模块包括:

- 两个 I/O 引脚
 - SCI 接收数据引脚
 - SCI 发送数据引脚
- 波特率可设定到 64K 不同速率
- 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 个错误检测标志: 奇偶、超载、数据帧和中断检测
- 两种唤醒多处理器模式: 空闲线唤醒和位寻址唤醒
- 半双工或全双工操作
- 双缓冲的接收和发送功能
- 发送器和接收器操作可以通过针对状态标志的中断驱动或轮询算法来完成。

- 发送器：TXRDY 标志（发送器缓冲寄存器可以接收新字符）和 TX EMPTY 标志位（发送器移位寄存器已空）
- 接收器：RXRDY 标志（接收器缓冲寄存器已经准备好接收另外字符），BRKDT 标志（故障条件发生）和 RX ERROR 标志（监视 4 个中断条件）
- 用于发送器和接收器中断的独立使能位（除了 BRKDT）
- NRZ（非归零）码格式
- SCI 模块控制寄存器位于控制寄存器帧内，开始地址为 7050h

注：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。写入高位字节无效。

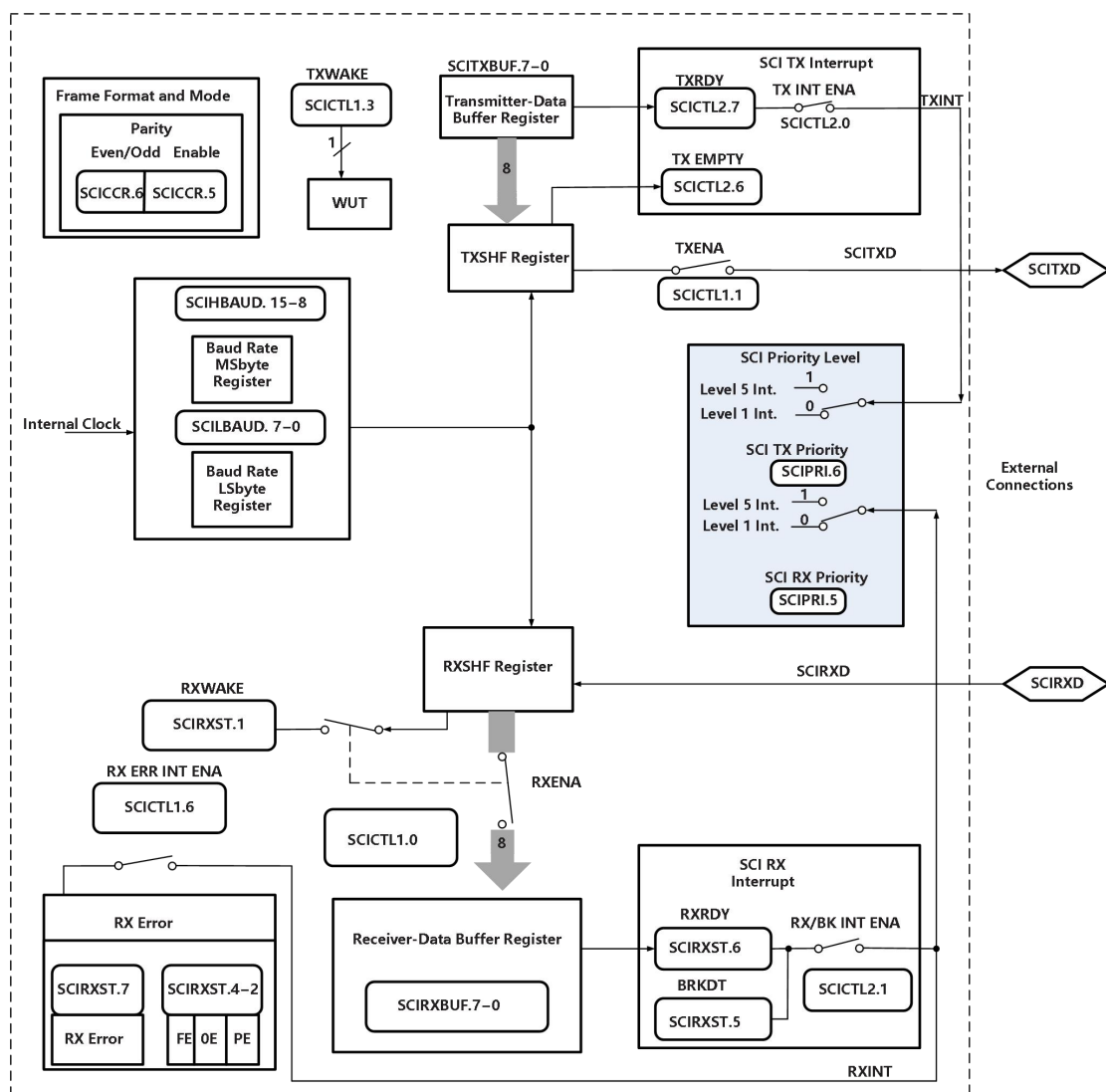


图 13. 1 SCI 功能框图

13.2 SCI 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/IDL EMODE	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: 禁用自测试模式; 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 SCICTL1 - 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 (写 0 有效) ;
位 4	保留位;
位 3	TXWAKE: SCI 发送器唤醒方式选择位。 0: 不选择发送特性; 1: 发送特性取决于工作模式, 空闲线模式或寻址位模式 空闲线模式, 对 TXWAKE 写入 1 后再对 SCITXBUF 写入传输数据, 生成一个 11 位数据的空闲周期 寻址位模式, 对 TXWAKE 写入 1 后再对 SCITXBUF 写入传输数据, 以设置给定数据帧的地址位为 1 TXWAKE 不能通过 SWRESET 清除, 但可以通过系统复位或者 TXWAKE 发送到 WUT 标识
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被送到接收移位寄存器, 再传送到接收缓冲器。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 SCIRXEMU 和 SCIRXBUF 接收缓冲寄存器; 1: 允许将接收到的数据传送到 SCIRXEMU 和 SCIRXBUF 接收缓冲;

3. 串行通信接口 (SCI) 控制寄存器 SCICTL2 - 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 未清空; 1: SCITXBUF 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/故障中断使能位。该位控制 RXRDY 或 BRKDT 标志位触发的中断请求。但 RX/BK INT ENA 并不阻止这些标志位置位;

	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 标志位触发的中断请求。但 TX INT ENA 并不阻止 TXRDY 标志位置位; 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 接收器接收到数据标志。 0: SCIRXBUF 中无新数据; 1: SCIRXBUF 中数据等待读取;
位 5	BRKDT: SCI 通讯中断检测标志; 0: 无通讯中断事件; 1: 发生通讯中断事件;
位 4	FE: SCI 帧错误标志位; 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 代表检测到接收器唤醒条件, 在地址位多处理器模式 (SCICCR.3=1), RXWAKE 反映了 SCIRXBUF 中数据地址位的值。在空闲线多处理器模式, 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 复位;
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 SCIRXBUF 中,当传送操作完成后,RXRDY 标志位置 1,表明接收到数据并等待读取操作。SCIRXBUF 和 SCIRXEMU 存放相同的数据,两者拥有独立的地址但在物理并不分离缓存,唯一区别仅在于,对 SCIRXEMU 执行读取操作时不会清除 RXRDY 标志位,而读取 SCIRXBUF 操作会同时清除 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
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；

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: 仿真挂起触发, SCI 立即停止; 01: SCI 持续运行, 不受仿真挂起影响; 10: 仿真挂起触发, 完成当前接收/发送操作后停止; 11: SCI 持续运行, 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。 $SCI\ BAUD = SYSCLK / [(BRR + 1) \times 8]$
----------	--

14 串行外设接口 SPI 模块

14.1 串行外设接口 (SPI) 模块简介

ADM16F03A2 器件集成三线串行外设接口 (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:

(1) 三个外部引脚:

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, 这三个引脚可以作为 GPIO 口使用。

(2) 两种运行模式: 主模式和从模式

(3) 波特率: 可编程 125 个不同的速率。

(4) 数据字长: 1 ~ 16 个数据位

(5) 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

7	6	5	4	3	2	1	0
SPI SW RESET	CLOCK POLARITY	保留	保留	SPI CHAR3	SPI CHAR2	SPI CHAR1	SPI CHAR0
RW_0	RW_0	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	SPI SW RESET：SPI 软件复位，当改变配置时，用户应在改变前，将该位清 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 种时钟配置方式，具体如表 14-1 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 14. 1 SPI 时钟配置

SPICLK 方式	时钟极性 (SPICCR.6)	相位极性 (SPICTL.3)
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 (受限于相关引脚负载) ; SPIBRR = 3~127 时 SPI 波特率 = SYSCLK/ (SPIRR+1) 受限于相关引脚负载) ;
---------	--

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 SUSPFREE：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；

15 事件管理器模块 EM1

15.1 事件管理器模块 EM1 简介

该系列器件集成一个事件管理器模块 EM1，此模块包括通用(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 15.1 列出了 EM1 包含的主要模块、信号名称、功能特性，图 15.1 展示了事件管理器模块（EM1）功能结构。

表 15.1 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

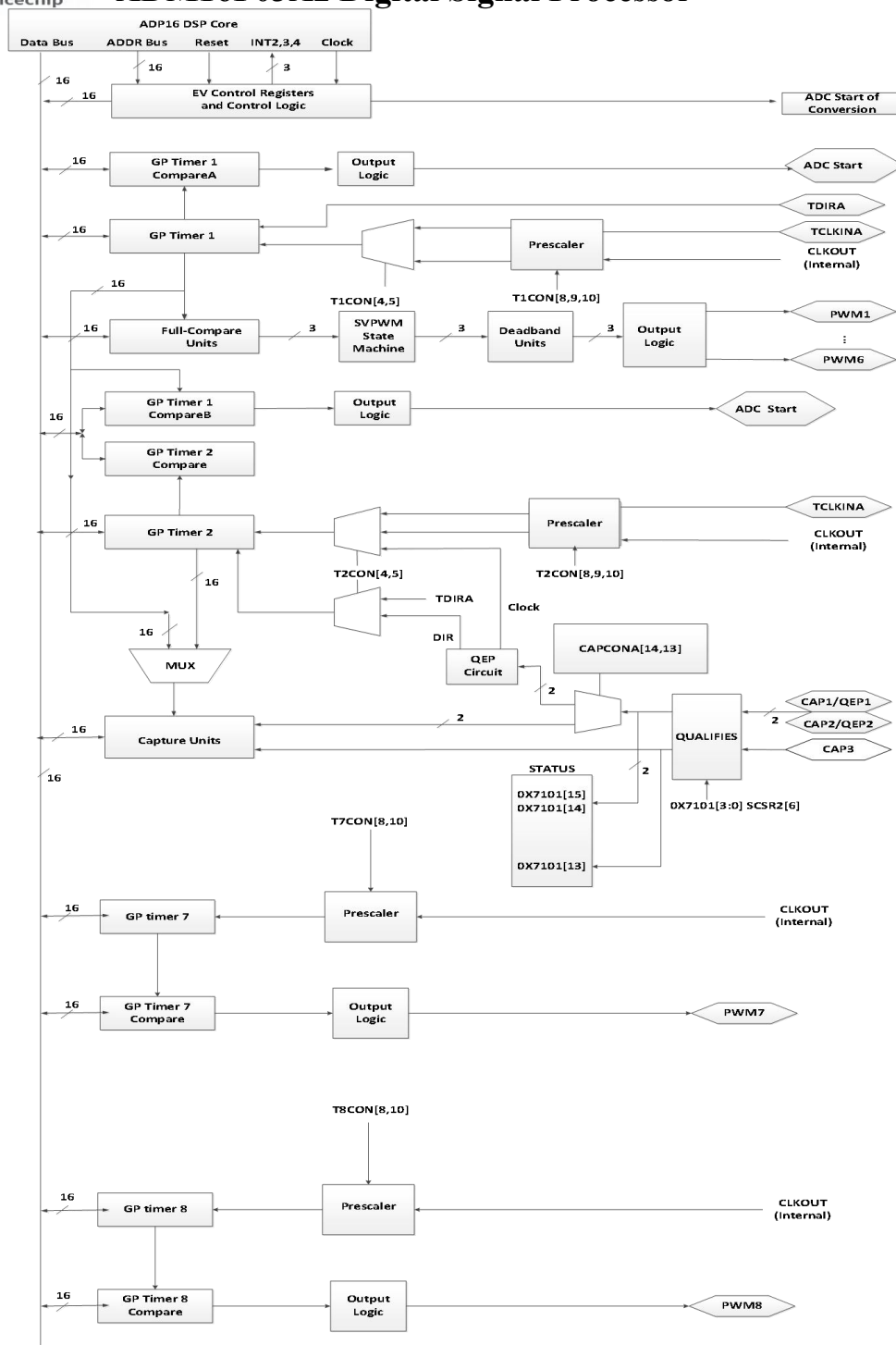


图 15.1 事件管理器模块 EM1 功能框图

15.2 事件管理器模块 EM1 中断

15.2.1 事件管理器模块 EM1 中断组

事件管理器中断事件分为三组：中断组 A，B，C。每个中断组包含各自不同的中断标志，中断使能寄存器和特定的外设事件中断请求。在第 9 章中断系统的表 9-1

ADM16F03A2 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。

15.2.2 事件管理器模块 (EM1) 中断控制寄存器

1. 中断标志寄存器 EM1IFRA—742Fh，中断屏蔽寄存器 EM1IMRA—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

注：R=可读，W=可写，C=清除，_0=复位值，_X=复位值不定，S=仅可置位，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； 1 标志被置位； 1 复位标志	比较单元 3 中断使能位： 0 禁止该中断； 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；	比较单元 2 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 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	PDPINT 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. 中断标志寄存器 EM1IFRB—7430h, 中断屏蔽寄存器 EM1IMRB—742Dh;

15~4	3	2	1	0
保留	T2OFINT FLAG	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=复位值不定, S=仅可置位, 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. 中断标志寄存器 EM1IFRC—7431 h,中断屏蔽寄存器 EM1IMRC—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=复位值不定, S=仅可置位, 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

15.3 事件管理器 EM1 通用定时器

事件管理器模块包含有四个通用定时器 Tx (x=1, 2, 7, 8)。这些通用定时器包含以下功能电路:

- 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, T7CN 地址 7533h, T8CNT 地址 7537h;
- 定时器 1 包含两个 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 地址为 7403h, T2PR 为地址 7407h, T7PR 地址为 7534h, T8PR 地址为 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，上溢，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 16.7ADC 寄存器概述），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 16.7ADC 寄存器概述），全功能比较器用于产生 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。

15.3.1 事件管理器 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) —7404h (T1CON) / 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=复位值不定，S=仅可置位，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 周期使能选择位, 在定时器 1 中保留; 0: 使用自身定时器使能位 1: 使用 T1CON 的使能或禁用, 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—只适合于定时器 2, 在定时器 1 中保留, 此操作只在 SEL T1PR = 0 时有效
位 3~位 2	TCLD1/TCLD0: 定时器比较寄存器的重新加载条件; 00: 当计数值是 0 时 01: 当计数值是 0 或等于周期寄存器时 10: 立即加载 11: 保留
位 1	TECMPR: 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—7532h (T7CON) / 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, 如果相应的中断未被屏蔽, CPU 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	TECMR	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=复位值不定，S=仅可置位，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—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 多种计数模式，以及计数方向；

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=复位值不定，S=仅可置位，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：不配置触发 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：保留；
位 8~位 7	T1TOADC：定时器 1 触发 A/D 转换事件； 00：不配置触发 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：保留；
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—753Eh;

15	14	13	12	11	10	9	8
PWM8_EN	PWM7_EN	PWM8_O UT_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

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—752Fh;

T7、T8 的上溢中断标志位产生时，将溢出中断标志置位，同时会触发相应的中断动作。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=复位值不定，S=仅可置位，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

15.4 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产生六路比较输出的 PWM 波形，并内嵌可编程的死区电路。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，以实现所需的比较/ PWM 脉冲宽度可编程调整。

完全比较单元包括：

- 3 个 16 位的比较寄存器，CMPR3 地址为 7419h，CMPR2 地址为 7418h，CMPR1 地址为 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~PWM6）；
- 控制和中断逻辑；

15.4.1 事件管理器模块 EM1 全功能比较控制寄存器

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=复位值不定，S=仅可置位，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 重新加载条件; 00: 定时器 1 下溢加载; 01: 定时器 1 下溢或周期加载; 10: 立即加载; 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 : 方式控制寄存器重新加载条件; 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—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=复位值不定, S=仅可置位, 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;

15.5 事件管理器模块 EM1 脉宽调制电路

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组独立对（六个输出 PWM1~6），PWM7/PWM8 单独输出至引脚。

15.5.1 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
- 功率驱动保护中断

- 波形发生器电路，可编程生成不对称、对称以及四象限向量的 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

15.5.2 PWM 相关寄存器

1. 死区定时器控制寄存器 DBTCONA—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=复位值不定，S=仅可置位，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 1 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；

15.6 事件管理器模块（EM1）捕获单元

捕获单元可以记录捕获输入引脚上的电平变换。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

两级深度的 FIFO 堆栈内。捕获单元由三个捕捉电路（CAP1、CAP2、CAP3）组成。

CAP 模块包括下列特性：

- 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 2 级深度的 16 位 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，输入必须在其当前的电平状态保持至少两个时钟的上升沿。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，或者双沿）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。

15.6.1 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—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=复位值不定，S=仅可置位，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 堆栈保持内容不变；

	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 堆栈保持内容不变; 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 ADC 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—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=复位值不定, S=仅可置位, 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: 已有两个值, 捕获另一个值, 丢弃首个压入的值;
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, 捕获另一个值, 丢弃首个压入的值;
位 9~位 8	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, 捕获另一个值, 丢弃首个压入的值;
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

15.7 正交编码脉冲(QEP) 电路

两个捕获输入引脚 CAP1 和 CAP2 复用片内 QEP 电路, 用于处理正交编码脉冲。

当正交脉冲序列的方向或者边沿被检测到时, 为了保证这些输入信号在芯片内部完全同步, 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。

16 模数转换器 ADC

16.1 ADC 的特性

- 12 位模数转换电路
- 2 路采样保持单元
- 模拟输入：0.0V 至 3.0V（高于 3.0V 输出满量程值，超过规格书定义端口耐压限值将对器件造成不可逆的损伤）
- 14MHz ADC 时钟，最快转换速率：1 MSPS，转换时间为 1000ns
- 16 个采样通道（其中 A3 通道未引出引脚，连接至片上温度传感器）
- 自动排序功能在单次转换序列最多可连续 16 次“自动转换”，每次转换通道可通过软件配置 16 个输入通道中的任意一个。
- 2 个的 8 态排序器模式可运行在双排序器模式（SEQ1 和 SEQ2），或级联为一个 16 态的 SEQ 排序器模式。
- 具有存储转换值的 16 个结果寄存器（RESULT0~RESULT15 可独立寻址）。

输入模拟电压对应的数字值由以下公式得出：

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

注：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：
 - 软件立即触发
 - 事件管理器 1 内部事件源

— ADCSOC 外部引脚

- 灵活的中断控制，支持每个序列结束(EOS)触发中断或每间隔一个 EOS 触发中断。
- 排序器可运行在“启/停”模式，支持多个时间排序的触发源转换同步。
- EM1 定时器 T1 相关联的两个 SOC 事件源可独立触发双排序器模式。
- ADC 偏移误差带出厂校准
- ADC 增益误差带出厂校准
- ADC 时钟频率可软件配置
- ADC 采样时间窗口可配置

ADC 模块集成一个 14MHz 的 ADC 时钟，可支持 1000ns 快速转换速度。带载 16 个输入通道，分别为 A0~A7 和 B0~B7 两组，这些通道可被配置为用于事件管理器 1 的独立 8 状态排序器或一个级联的 16 状态排序器。尽管有多个输入通道和 2 个排序器，但 ADC 模块仅存在一个转换器电路。2 个 8 状态排序器或级联的 16 状态排序器可自动对一系列转换定序，每次转换可通过模拟 MUX 选择 16 个输入通道中的任何一个。在每个排序器上，一旦转换完成，所选通道的转换结果存储在对应 RESULTx 寄存器中。系统可使用自动定序功能多次转换同一输入通道，实现过采样的算法。此算法可提供比传统的单一采样转换结果更高的分辨率。

16.2 ADC 模块概述

如图 16.1 所示，该系列器件的 ADC 系统包括 12 位 ADC 转换电路，通道选通开关以及排序器 SEQ1 和 SEQ2 或 SEQ。

其中, 排序器 SEQ1 和 SEQ2 或 SEQ 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。ADCINA3 通道固定连接内部温度传感器输出端。

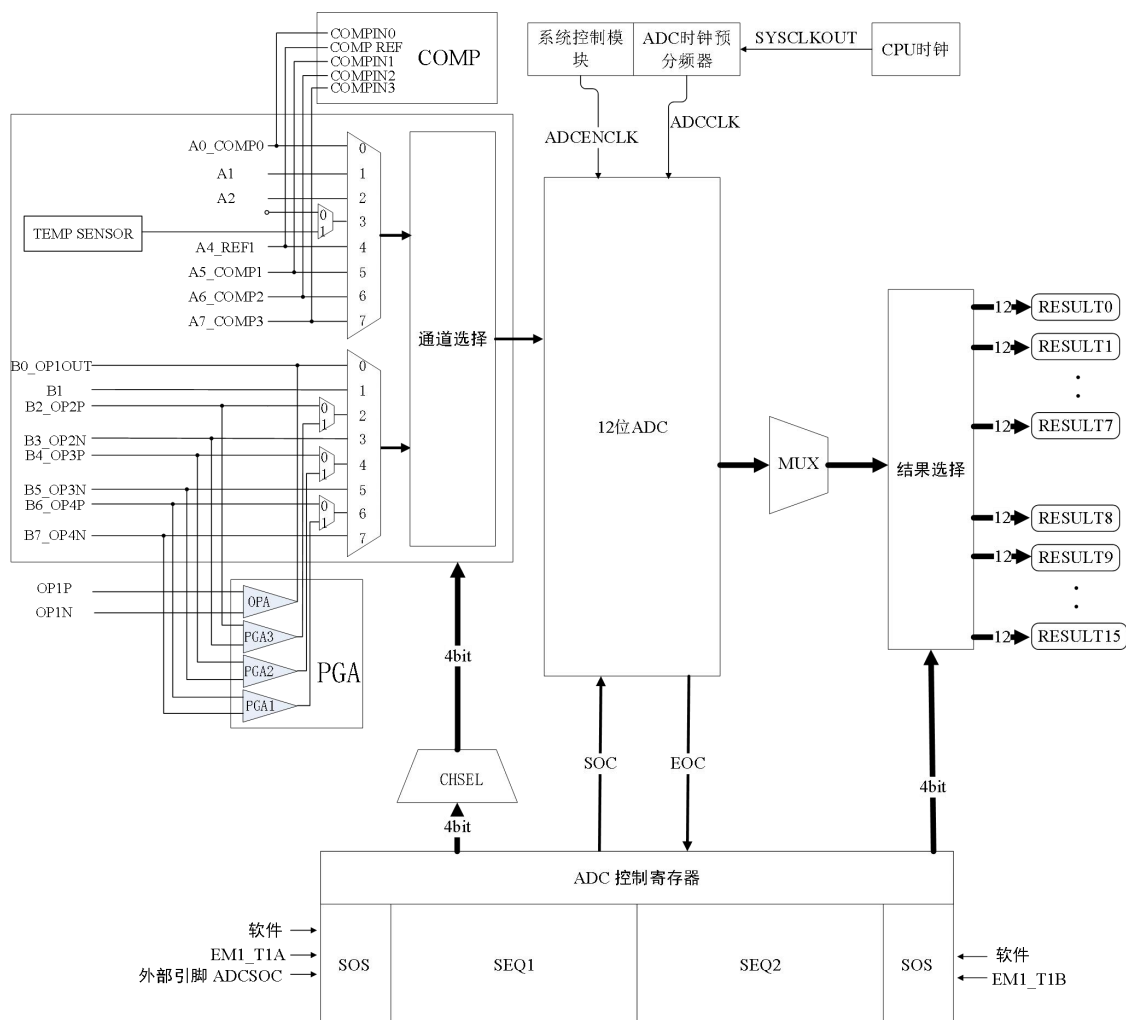


图 16. 1ADC 系统结构框图

建议在所有电源引脚上使用外部去耦合电容器。ADC 电源的引脚连线如图 16.2 所示, 设计要达到规格书定义的 ADC 精度, 电路板布局合理性非常关键。为尽可能达到最佳效果, 到 ADCINxx 引脚的走线不应与数字信号通道靠的太近。这是为了最大程度地降低耦合到 ADC 输入路径的数字线路的开关噪声。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。未使用的 ADC 输入引脚应该连接模拟地。

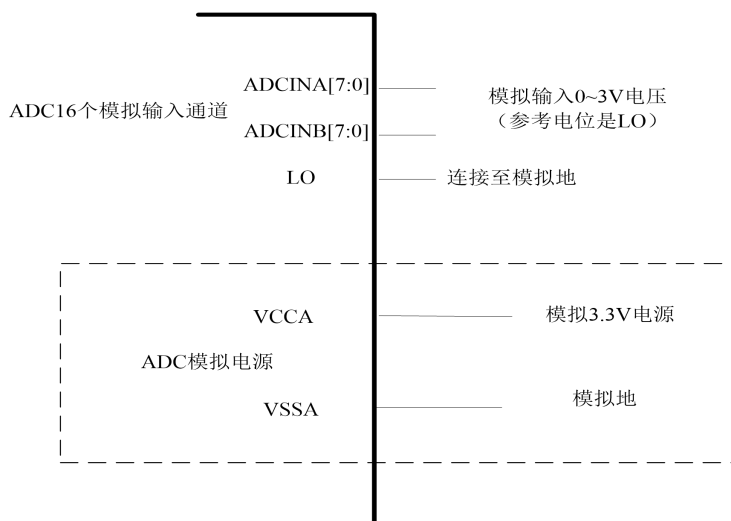


图 16. 2 ADC 电源引脚连接示例

16.3 自动排序器

图 16.3 是双排序模式下自动排序原理框图，图 16.4 是级联模式下自动排序原理框图。该 ADC 排序器由 2 个独立的 8 状态排序器 (SEQ1 和 SEQ2) 组成，SEQ1 和 SEQ2 也可以级联成一个 16 状态排序器 (SEQ)。这两种模式下，ADC 都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，它可以自动执行多个转换。对于每次需要转换通道，可通过模拟 MUX 选择 16 个可用输入通道中的任意一个。转换之后，所选通道采样转换的数值将存储在对应的结果寄存器 (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）。还可以对同一通道进行多次采样，实现“过采样”算法，此方式比传统的单采样转换结果可获得更高的分辨率。

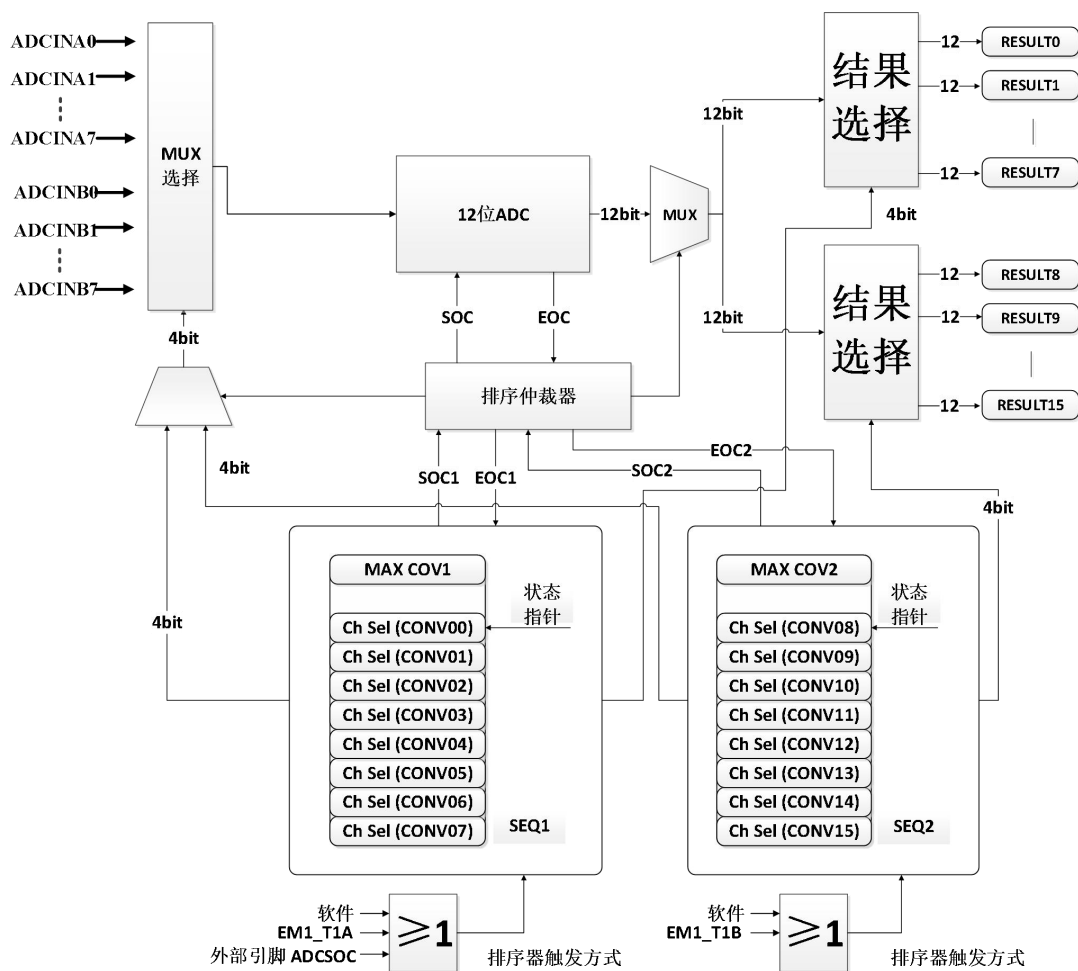


图 16. 3 双排序模式下 ADC 自动排序原理框图

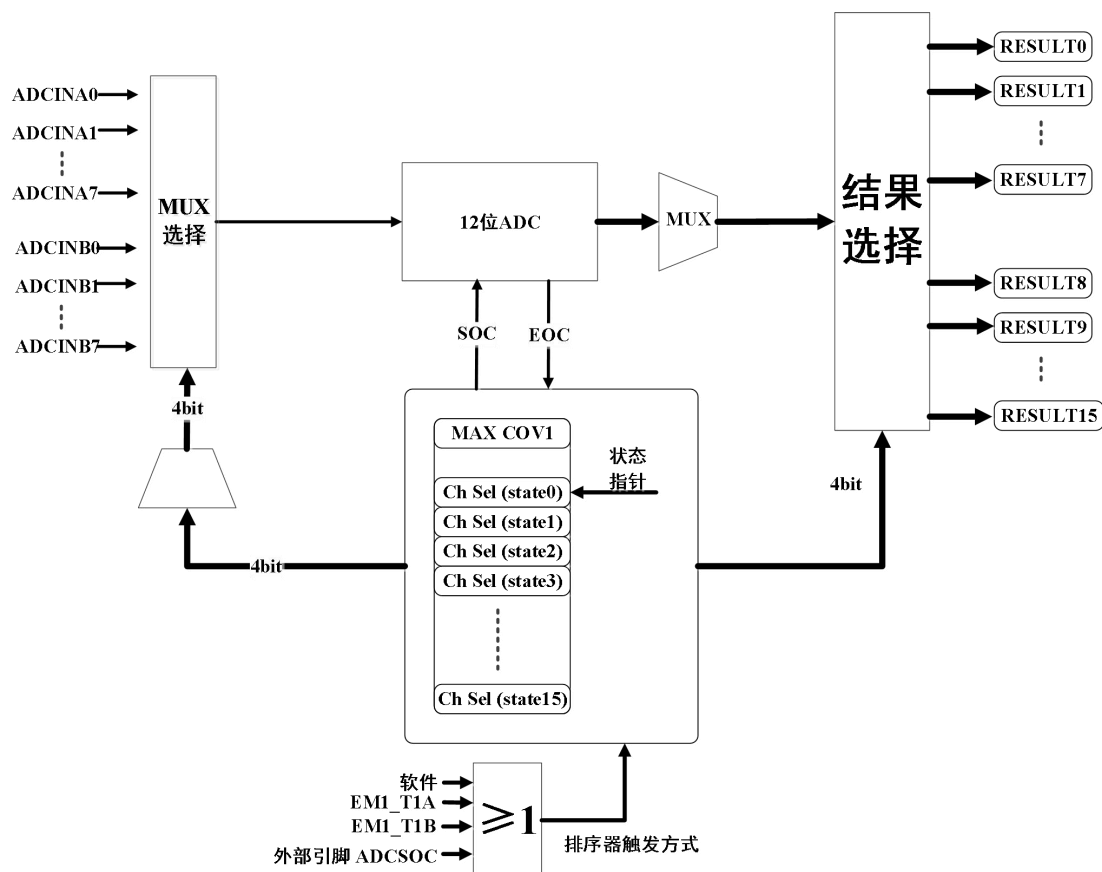


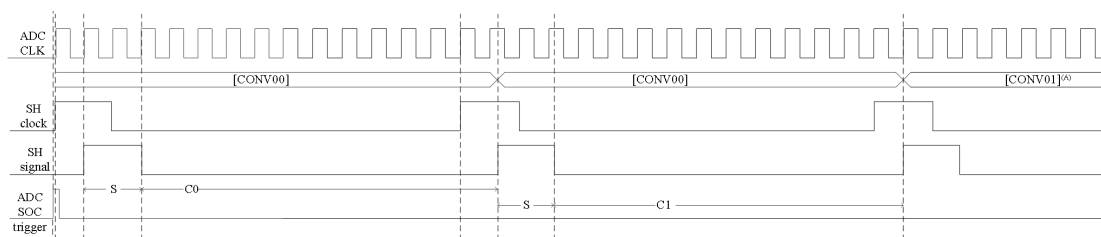
图 16. 4 级联模式下 ADC 自动排序原理框图

注:

在双排序器顺序采样模式中，一旦完成当前活动排序器启动的序列，则将执行挂起的来自其它排序器 SOC 请求。例如，假设出现来自 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 请求都为挂起状态，则 SEQ1 的 SOC 具有优先权。假设 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 请求继续保持挂起状态。

ADC 可以运行于顺序采样模式。对于每次转换，当前 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 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。

16.3.1 顺序采样模式



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 通道由[CONV00] 4 位寄存器选择；CONV00 对应 SEQ1，CONV08 对应 SEQ2。

图 16. 5 顺序采样模式的时序

表 16. 1 不同工作模式 SEQ1 和 SEQ2 比较

特性	单个 8 态 排序器 #1(SEQ1)	单个 8 态 排序器 #2 (SEQ2)	级联 16 态 排序器 (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

后续将排序器状态，定义统一如下：

- SEQ1： CONV00 - CONV07
- SEQ2： CONV08 - CONV15
- 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 位的设置值可以是 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

示例初始化:

AdcRegs.ADCTRL3.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 ADCTRL2 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

16.3.2 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意通道自动排序 8 次转换 (当级联模式或排序器先后一起使用时为 16 次)。此模式流程如图 16.5 所示。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数, 并按顺序持续 (CONV01、CONV02...以此类推) 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

假设某个转换序列 SEQ1 进行 7 次转换（定序过程为转换输入 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 16.2。

表 16.2 示例 2 中 CHSELSEQn 寄存器写入值

	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 16.6 为不间断自动排序模式下的转换流程图。

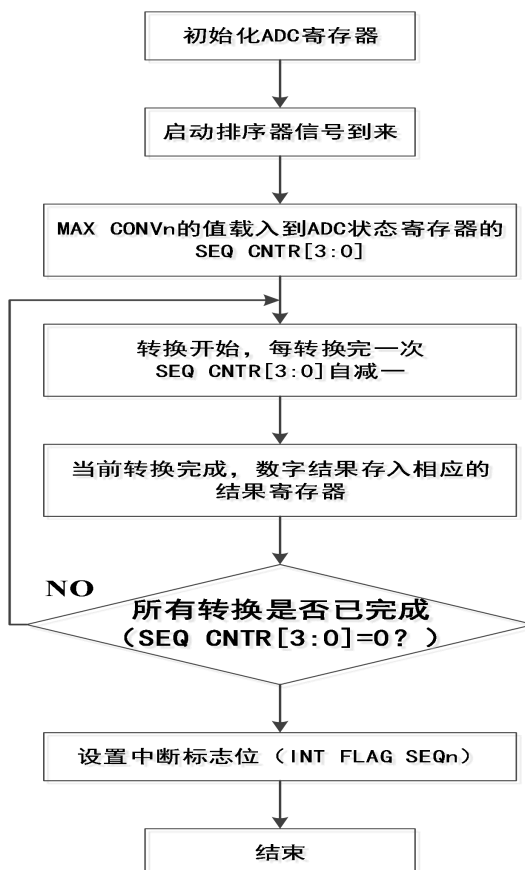


图 16. 6 不间断的自动排序模式 ADC 转换流程图

当排序器接收到转换开始 (SOC) 触发时即启动转换。SOC 触发时同时加载 SEQ_CNTR n 值。CHSELSEQ n 寄存器中设定的通道将按照预设定的顺序进行转换。每完成一个通道转换后，SEQ_CNTR n 值将自动减 1。当 SEQ_CNTR n 值减小到 0 时，可能触发两种操作，操作结果与 ADCTRL1 寄存器中连续运行位(CONT_RUN) 的状态相关。

当启用 CONT_RUN 设置，转换序列将自动再次启动（即 SEQ_CNTR n 重新加载 MAX_CONV1 的原始值且 SEQ1 状态设置为 CONV00）。此时为避免数据覆盖，应确保在下一转换序列开始之前完成对结果寄存器的读取。ADC 电路设计了仲裁机制，以确保出现争议条件触发时不会破坏结果寄存器（CPU 对结果寄存器进行读操作时，ADC 模块同时尝试对结果寄存器的写操作）。

如果未启用 CONT_RUN 设置，排序器将保持上一次转换序列的最后状态（示例中

为 CONV06) , 且 SEQ_CNTR 继续保持为 0 值。要在下一个 SOC 触发时重复序列, 必须在下一 SOC 之前使用 RST_SEQ n 复位排序器。

因每次 SEQ_CNTR 到达 0 时中断标志均产生, 用户可以在中断服务子程序程 ISR 中手动复位排序器 (对 ADCTRL2 寄存器中的 RST_SEQn 位进行写 1 操作) , 操作后下一次 SOC 触发时 SEQ_CNTR n 重新加载 MAX_CONV1 值且使得 SEQn 状态复位到初始值 (CONV00 对 SEQ1, CONV08 对 SEQ2) 。此功能可实现排序器工作在“启/停”工作模式。

16.3.3 排序器的启/停工作模式

除了不间断自动排序模式, SEQ1、SEQ2 或者 SEQ 中任意一个均可工作在停止/启动模式, 此模式可实现同步多个分时转换开始(SOC) 触发源。此模式类似于示例 2, 区别在于排序器完成第一个转换序列后, 可重新触发并且不复位到初始状态 CONV00 (即在中断服务子程序内不复位该排序器) 。因此, 当一个转换序列结束时, 排序器保持在当前转换状态。此模式下 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/停操作

要求: 使用触发源 1 (下溢) 启动 3 次自动转换 (如 I1、I2 和 I3) , 使用触发源 2 (定时周期) 启动 3 次自动转换 (如 V1、V2 和 V3) 。触发源 1 与触发源 2 在时间上相差 25 μ s , 并由事件管理器 EM1 的 T1A 和 T1B 输出。时序图如图 15.7 所示。此示例中仅使用了 SEQ1。

注:

触发源 1 和触发源 2 均可以选择 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, 以满足本例的双触发需要。同时应该注意不要因正在处理的序列而丢失多个 EM1 触发信号。详细可参考第 15.3.4 部分。

此例中将 MAX_CONV1 的值设置为 2, ADC 输入通道选择排序控制寄存器 (CHSELSEQn) 设置如表 16.3 所示。

复位完成初始化后, SEQ1 将等待触发源。首个触发信号发生时, 执行通道选择值为 CONV00(I1)、CONV01(I2) 和 CONV02(I3)的 3 次转换。之后 SEQ1 将在当前状态等待下一个触发源。25 微秒后第二个触发源

到达，SEQ1 将执行通道选择值为 CONV03 (V1)、CONV04 (V2) 和 CONV05 (V3)的 3 次转换。

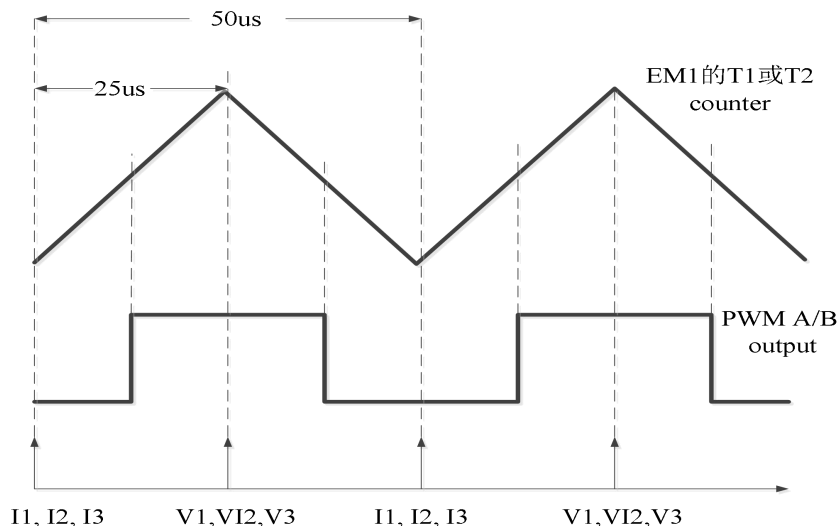


图 16. 7 EM1 触发源启动排序器示例

两次触发源到来时，MAX_CONV1 的值均自动加载到 SEQ_CNTR n 中。如第二个触发源到来时，需要修改不同的转换数，则必须通过软件（在第二次触发前恰当的时间）完成 MAX_CONV1 值的更改，否则，之前的加载值将被重新使用。可通过中断服务程序（ISR）实现在适当的时间修改 MAX_CONV1 值。第二次自动转换过程结束后，ADC 结果寄存器的值如表 16.4 所示。

此时 SEQ1 将保持在当前状态等待下一个触发源。用户可通过软件将 SEQ1 复位到状态 CONV00，并重复相同的触发源 1 和触发源 2 的转换任务。

表 16. 3 CHSELSEQ n 设定值 (MAX_CONV1 设为 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 16. 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

16.3.4 输入触发源

每个排序器可配置一组触发源输入，且可独立使能或禁用。SEQ1、SEQ2 以及级联 SEQ 的有效输入触发源选择范围可参考表 16.5。

表 16.5 输入触发源

SEQ1	SEQ2	级联 SEQ
软件触发 (软件 SOC)	软件触发 (软件 SOC)	软件触发 (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：

- SOC 触发源可在排序器处于空闲状态时启动自动转换序列。空闲状态是指排序器处于接收触发源之前的 CONV00，或者处于转换序列完成时（即 SEQ_CNTR 计数到 0 时）排序器所处的状态。
- 如果在执行当前转换序列时出现另一个 SOC 触发源，则它将对 ADCTRL2 寄存器中的 SOC_SEQ n 位写 1（此位在上一转换序列开始时已被清除）。之后再发生的 SOC 触发源将丢失（即当 SOC_SEQ n 位已被置 1（SOC 挂起），之后的触发源将被忽略）。
- 触发源到来时，排序器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，此时排序器将立即返回到空闲起始状态（SEQ1 处于 CONV00；SEQ2 处于 CONV08）。
- 当 SEQ1/2 用于级联模式时，进入 SEQ2 的触发源将被忽略，而 SEQ1 的触发源有效。此时级联模式可看作是 16 态而非 8 态的 SEQ1。

16.3.5 排序转换期间的中断操作

排序器可在两种工作模式下产生中断。中断模式可由 ADCTRL2 寄存器的中断模式使能

控制位配置。示例 3 展示了中断模式 1 和 2 在不同工作场景下的用途。

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 时发起中断请求）

1. 将 MAX_CONVn 写入 1 并完成排序器初始化，转换 I1 和 I2 通道。
2. 在"a"中断时刻，软件将 MAX_CONVn 更改为 2，以转换 V1、V2 和 V3 通道。
3. 在 "b "中断时刻，执行以下操作：
 - a. 再次将 MAX_CONVn 更改为 1，以转换 I1 和 I2 通道。
 - 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：每次 SEQ_CNTR 到 0 中断标志均被置位，且进入中断服务子程序。

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. 将 MAX_CONVn 设置为 2 并初始化排序器，以转换 I1、I2 和 I3 (或 V1、V2 和 V3)。
2. 在 "b" 和 "d" 中断时刻处，将执行下述操作：
 - 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：每次 SEQ_CNTR 到达 0 时中断标志位都被置 1，即 ADC 完成转换 I1、I2 和 I3 或完成转换 V1、V2 和 V3。而且仅当转换完 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（带有空读取操作）

- 模式 2 中断操作（即每隔一个 EOS 信号产生中断请求）
 1. 将 MAX_CONVn 配置为 2 并初始化排序器，以转换 I1、I2 和 x（虚拟采样）。
 2. 在 ISR "b" 和 "d" 处，将执行下述操作：
 - a. 从 ADC 结果寄存器读取值 I1、I2、x、V1、V2 和 V3。
 - b. 排序器被复位。
 3. 重复步骤 2。

注意，第一个序列的第三采样 x 为虚拟采样，而并非必须项。然而为最大程度地降低 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的特性。

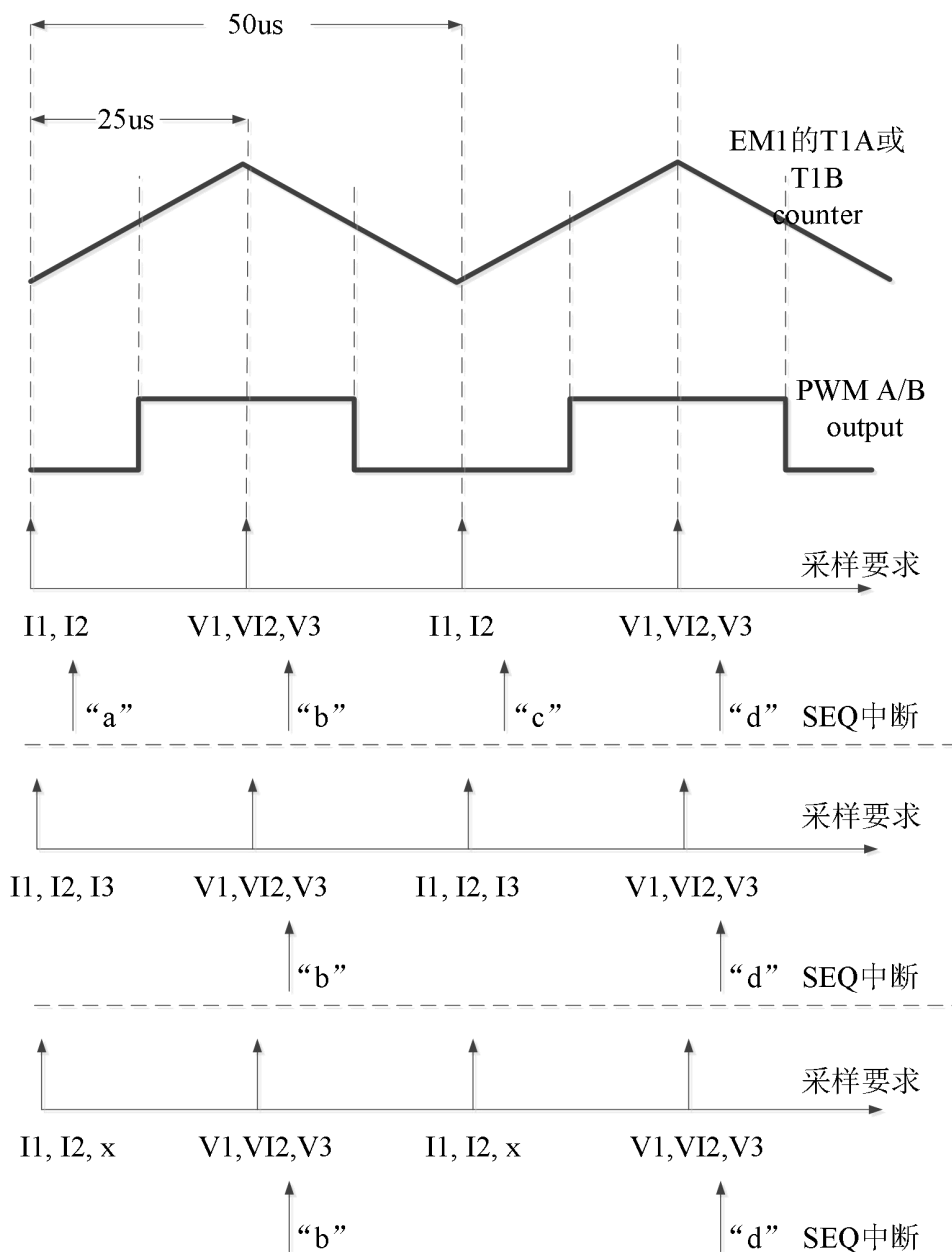


图 16. 8 多次排序转换之间的中断操作

16.4 ADC 时钟预分频

如图 16.9 所示，ADCCLK 为 ADC 模拟模块的主时钟，SH clock 为采保时钟（其中高电平区间代表 ADC 的采样窗口时长）。ADCCLKPS3~0 位为 ADC 时钟预分频控制位，ACQPS3~0 位为 ADC 采样窗口调节控制位。

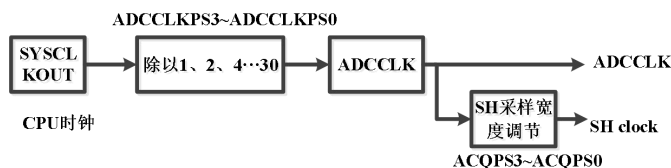


图 16.9 ADC 时钟预分频

16.5 ADC 校准

ADC 出厂校准数据会在器件上电后自动存入到 CALIBRATION 寄存器。此后，每次 ADC 正常转换的数字码值经过补偿偏移误差后送入结果寄存器中。

表 16.6 CALIBRATION 校准寄存器—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

16.6 片内温度传感器

温度传感器可用于测试器件结温，传感器输出连接到 ADCINA3 通道，ADCCTL1 的位 7 (REG_T_PWD) 控制温度传感器的上电或断电。A3 通道除与内部温度传感器连接外，并未引到芯片引脚。

16.6.1 温度传感器传递函数

温度传感器的输出经 ADCINA3 通道采样，其采样值随着结温的升高而增加。偏移误差定义为 0°C 时采样 LSB 码值，如图 16.10 所示。该信息可用于将传感器采样 ADC 码值转换为温度单位。需通过软件运算转换为温度。在对温度传感器进行采样时，ADC 必须满足“温度传感器特性”表中的采集时间要求。

调用 0x752E 寄存器值作为 Slope，温度的传递函数定义为：

$$Temp = (Sensor_{(D)} - Offset_{(D)}) \times \frac{Slope_{(D)}}{1024}$$

调用温度传感器特征值作为 Slope，温度的传递函数定义为：

$$Temp = (Sensor_{(D)} - Offset_{(D)}) \times Slope_{(D)}$$

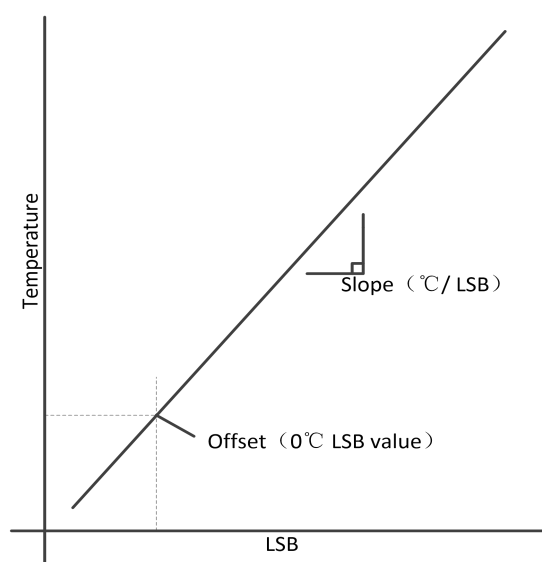


图 16.10 温度传感器传递函数

斜率和偏移详细介绍可参考 ADP16F03 数据手册中的电气特性部分，每个器件在出厂时均写入了独立校准的的斜率和偏移值，写入地址信息如下。

- 0x752E - slope 斜率 (°C/LSB，十进制乘以 1024 后，转换为二进制数据存放于此寄存器)；
- 0x752F - offset 偏移 (0°C对应的 LSB 码值)

列出的值满量程范围为 3V，使用内部 ADC 基准电压模式会自动匹配此固定电压范围。

表 16.7 温度传感器特性

参数 ⁽¹⁾		最小	典型	最大	单位
$t_{startup}$	启动时间			1	ms
t_{acq}	ADC 采集窗口时间	500			ns
T_{slope}	温度传感器测得的 ADC1 个 LSB 变化对应的温度变化值		0.128 ⁽²⁾⁽³⁾		°C/LSB

T_{offset}	在温度传感器读数为 0°C 时的 ADC 输出		2500		LSB
--------------	-------------------------	--	------	--	-----

- (1) 温度传感器斜率和偏移根据使用ADC内部基准的ADCLSB指定。
- (2) ADC温度系数已被纳入技术规范。
- (3) 温度传感器的输出（LSB结果）与温度呈现正向变化关系。温度上升将使得ADC值相对于初始值增加；温度下降将使得ADC的值相对于初始值下降。

为保证温度传感器的精度，温度传感器的使用需在其上位（ADCTRL1 寄存器内的 REG_T_PWD 位）使能后等待 $t_{startup}$ 时间。且 ADC 采集窗口时间需满足 t_{acq} 最小要求。

每颗芯片在寄存器地址 0x752E 与 0x752F 分别存有温度传感器的 Slope 值与 Offset 值，分别代表 ADC 每变化 1 个 LSB 代表的温度变化值，及 0°C 时 ADC 采样温度传感器数值。用户可根据 Slope 值与 Offset 值通过软件计算当前温度值，此时的温度传感器结果更加准确。

16.7 ADC 寄存器概述

表 16. 8 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. ADCTRL1 寄存器—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，写入无效。
14	RESET	0 1	针对 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，复位后将此位设为 0 注：ADCTRL1 寄存器复位操作示范 系统复位时 ADC 模块也被复位。如果希望在其它时刻对 ADC 进行复位，可以在此位写 1 实现。一个 NOP 指令后，就可以向 ADCTRL1 寄存器写入恰当的值。
13-12	SOFT 和 FREE 位	0 0 0 1 X 1	此两位决定仿真挂起时 ADC 模块的工作状态。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转换后停止。 仿真挂起时立刻停止 停止前完成当前转换 自由运行，忽略仿真挂起继续进行
11-8	ACQ PS [3:0]	0000 0001 0010 0011 0100 0101 0110 0111	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。SOC 脉宽为 ADCTRL1[11:8] + 1 乘以 ADCCLK 周期。 1xT _{ADCCLK} 2xT _{ADCCLK} 3xT _{ADCCLK} 4xT _{ADCCLK} 5xT _{ADCCLK} 6xT _{ADCCLK} 7xT _{ADCCLK} 8xT _{ADCCLK}

		1000 1001 1010 1011 1100 1101 1110 1111	9xT _{ADCCLK} 10xT _{ADCCLK} 11xT _{ADCCLK} 12xT _{ADCCLK} 13xT _{ADCCLK} 14xT _{ADCCLK} 15xT _{ADCCLK} 16xT _{ADCCLK}
7	REG_T_PWD	0 1	控制温度传感器上电或断电，温度传感器固定连接在 ADC 的 A3 通道（A3 未引出到芯片引脚） 温度传感器断电 温度传感器上电
6	CONT RUN	0 1	连续运行位。 此位决定排序器工作在连续转换模式还是启/停模式。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/停模式下必须复位排序器，以使转换器进入状态 CONV00。 启/停模式。到达 EOS 后排序器停止。除非复位排序器，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 SEQ_OVRD 清 0，则排序器将再次从其复位状态启动（CONV00 对应 SEQ1 和级联 SEQ，CONV08 对应 SEQ2）。如 SEQ_OVRD 置 1，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 1	排序器级联操作控制。此位决定 SEQ1 和 SEQ2 作为 2 个 8 态排序器工作还是作为单个 16 态排序器工作（SEQ）。 双排序器模式。SEQ1 和 SEQ2 作为 2 个 8 态排序器工作。 级联模式。SEQ1 和 SEQ2 作为单个 16 态排序器工作（SEQ）。
3-0	ADCCLKPS [3:0]		ADC 时钟预分频。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]，以产生 ADCCLK。
		0000 0001 0010 0011 0100 0101 0110 0111	ADCCLKPS [3:0] ADC 外设时钟除法器 ADCCLK 0 HSPCLK/1 1 HSPCLK/2 2 HSPCLK/4 3 HSPCLK/6 4 HSPCLK/8 5 HSPCLK/10 6 HSPCLK/12 7 HSPCLK/14

		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. ADCTRL2 寄存器—70A1h

15	14	13	12	11	10	9	8
保留位	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	保留位		
14	RST SEQ1	0 1	复位排序器 1，对此位写 1 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。可中止当前活动的转换序列。 无操作 立即将排序器复位到状态 CONV00
13	SOC SEQ1	0 1	排序器 1 (SEQ1) 或级联排序器 SEQ 的转换开始 (SOC) 触发源。可通过以下方式设置此位： •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) 情形 1：SEQ1 空闲且已清除 SOC 位，则 SEQ1 立即启动 (受仲裁机制控制)。允许为任何“挂起”的触发请求设置和清除此位。 情形 2：SEQ1 忙且已清除 SOC 位，则设置此位以指出触发请求正挂起。当完成当前转换后最终启动 SEQ1 时，将清除此位。 情形 3：SEQ1 忙且设置了 SOC 位，则忽略此情况下出现的任何触发信号 (丢失触发事件)。 清除挂起的 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，即不能通过写 0 来停止已启动的排序器。 软件触发器 - 从当前停止的位置启动 SEQ1（即空闲模式） 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，并在随后的指令对 SOC_SEQ1 写 1。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。
12	SEQ1 BSY	0 1	SEQ1 忙状态位。 当 ADC 自动转换排序正在进行 SEQ1 转换时此位被置 1。当转换完成时此位被清 0。 排序器处于空闲状态 排序器正在进行转换
11-10	INT ENA SEQ1	00 01 10 11	SEQ1 的中断方式使能控制位。 中断禁止 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。 保留
9	INT FLAG SEQ1	0 1	对 SEQ1 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。 无中断事件发生 中断事件发生过
8	EM1_T1A SOC SEQ1	0 1	EM1_T1A 对 SEQ1 产生 SOC 信号的屏蔽位。 禁用 EM1_T1A 的触发源启动 SEQ1 启用 EM1_T1A 的触发源启动 SEQ1
7	EXT SOC SEQ1	0 1	外部信号对 SEQ1 的转换启动位。 无操作 允许 ADCSOC 引脚上的信号触发 ADC 自动转换序列
6	RST SEQ2	0 1	排序器 2 复位 无操作 立刻复位 SEQ2，指针指向 CONV08，复位排序器时当前正在转换序列将被打断。
5	SOC SEQ2	0 1	排序器 2（SEQ2）的转换开始触发源。（仅适用于双排序器模式；级联模式时忽略。）可通过以下触发器设置此位： • S/W——通过软件将 1 写入此位 • EM1_T1B——事件管理器的 T1B 比较器 当触发源到来时，有三种工作状态可能性： 情形 1：SEQ2 空闲且已清除 SOC 位，则 SEQ2 立即启动（受仲裁机制控制），且允许为任何挂起的触发请求清除该位。 情形 2：SEQ2 正处于转换状态且已清除 SOC 位，则此位写入 1 代表存在触发请求处于挂起状态。完成当前转换后启动挂起的

			SEQ2 时将此位清 0。 情形 3：SEQ2 正处于转换状态且 SOC 同时置位，此情形将忽略此后续触发事件。 清除挂起的 SOC 触发源 注：如排序器已启动，则自动清除此位；写入 0 无效，即不能通过对此位写 0 停止已启动的排序器。 软件触发，从当前停止的位置启动 SEQ2（即空闲模式）
4	SEQ2 BSY	0 1	SEQ2 忙状态位。 当 ADC 自动排序转换时此位置 1。转换完成后此位被清 0。 排序器处于空闲状态 排序器正在进行转换
3-2	INT ENA SEQ2	00 01 10 11	SEQ2 的中断方式使能控制位。 中断禁止 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。仅当中断标志位已置 1 时，才产生中断请求，此模式支持每间隔一个转换完成产生中断请求。 保留
1	INT FLAG SEQ2	0 1	对 SEQ2 的 ADC 中断标志位 此位表示中断事件是否产生。用户通过向该位写 1 清除此位。 无中断事件发生 中断事件发生过
0	EM1_T1B SOC SEQ2	0 1	EM1_T1B 对 SEQ2 产生 SOC 信号的屏蔽位。 禁用 EM1_T1B 的触发源启动 SEQ2 启用 EM1_T1B 的触发源启动 SEQ2

3. 最大转换通道寄存器 MAXCONV—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		0000	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。 对于 SEQ1，使用 MAXCONV1_2~0 位。 对于 SEQ2，使用 MAXCONV2_2~0 位。

		0001 1110 1111 15 16	对于级联 SEQ, 使用 MAXCONV1_3~0 位。 最大通道转化数对应关系如下:
ADCMAXCONV 寄存器位编程示例: 如仅需要配置 5 次转换, 则将 MAX_CONVn 设置为 4。 情形 1: 双排序器模式 SEQ1 和级联模式 排序器从 CONV00 变化到 CONV04, 并将 5 个转换结果存储在结果寄存器 RESULT 00 至 RESULT 04 中。 情形 2: 双排序器模式 SEQ2 排序器从 CONV08 变化到 CONV12, 并将 5 个转换结果存储在结果寄存器 RESULT 08 至 RESULT 12 中。 情形 3: MAX_CONV1 值大于 7 时双排序器模式 如双排序器模式 (即 2 个独立的 8 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 AUTO_SEQ_SR—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	排序器计数状态位。自动排序转换任务开始前, SEQ CNTR n 的值被加载到 MAX CONVn 寄存器中, 可通过读取 SEQ CNTRn 减计数的值判断排序器状态, 此值与 SEQ1 和 SEQ2 Busy 状态位, 共同决定排序器工作进程。

		1001	8	
		1010	9	
		1011	10	
		1100	11	
		1101	12	
		1110	13	
		1111	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 或级联模式指针状态。

5. 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控制位介绍，如下表所示：

通道选择寄存器 CHSELSEQ1—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			

通道选择寄存器 CHSELSEQ2—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			

通道选择寄存器 CHSELSEQ3—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			

通道选择寄存器 CHSELSEQ4—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 可选择 16 个模拟输入通道中的任何一个。

CONV _{xx} 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 RESULTn—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			结果寄存器的低 4 位，读取时返回 0。

7. PWM 同步触发采样模式选择寄存器—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
R_0	R_0	R_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_C_T	COMP2_C_T	COMP3_C_T
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；

8. 温度传感器 TSLOPE—752Eh

15	14	13	12	11	10	9	8
T_SLOPE1_5	T_SLOPE1_4	T_SLOPE1_3	T_SLOPE1_2	T_SLOPE1_1	T_SLOPE1_0	T_SLOPE9	T_SLOPE8
WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0
7	6	5	4	3	2	1	0
T_SLOPE7	T_SLOPE6	T_SLOPE5	T_SLOPE4	T_SLOPE3	T_SLOPE2	T_SLOPE1	T_SLOPE0
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	名称	值	说明
15-0	T_SLOPE _x		存放温度传感器的斜率 T _{SLOPE}

9. 温度传感器 TOFFSET—752Fh

15	14	13	12	11	10	9	8
T_OFFSET1_5	T_OFFSET1_4	T_OFFSET1_3	T_OFFSET1_2	T_OFFSET1_1	T_OFFSET1_0	T_OFFSET9	T_OFFSET8
WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0
7	6	5	4	3	2	1	0
T_OFFSET7	T_OFFSET6	T_OFFSET5	T_OFFSET4	T_OFFSET3	T_OFFSET2	T_OFFSET1	T_OFFSET0
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	名称	值	说明
15-0	T_OFFSETx		存放温度传感器的偏移 T _{OFFSET}

17 运放与复用数字 IO 输入

17.1 运放与复用数字 IO 输入说明

该系列器件包括 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。放大器输入端口连接到施密特触发电路, 可作为数字逻辑输入端口使用。

运放与复用数字 IO 的连接关系如图 17.1 所示。

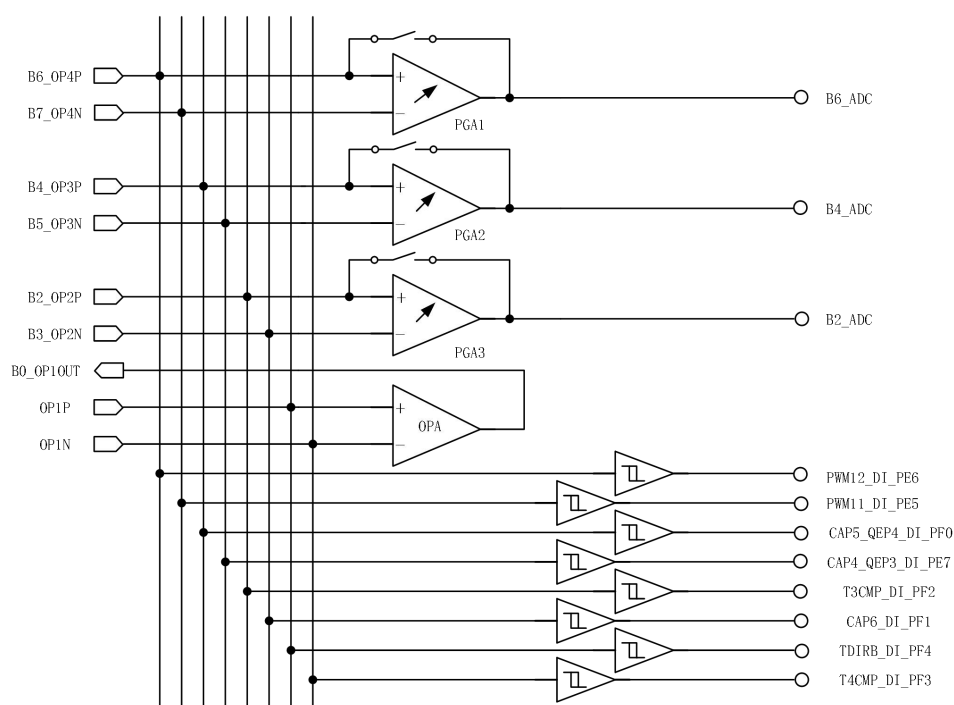


图 17.1 运放与数字 I/O 复用连接

PGA 可用作 ADC 输入的前置放大器, 通过寄存器 0x7106 配置, 3 路 PGA 输出分别连接到通道 ADCINB6、ADCINB4、ADCINB2。

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 ADCINB6 直通, 引脚 B4_OP3P 与内部 ADCINB4 直通, 引脚 B2_OP2P 与 ADCINB2 直通, 信号可直接流入引脚 B6_OP4P、B4_OP3P、B2_OP2P 后被 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 PGA1、

PGA2、PGA3 的放大器功能。引脚 B6_OP4P、B4_OP3P、B2_OP2P 与内部 ADC 通道之间开关断开，信号经过 PGA 放大后再被 ADC 采样。

PGA 等效电路如图 17.2 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，信号源需要具备一定的驱动能力。

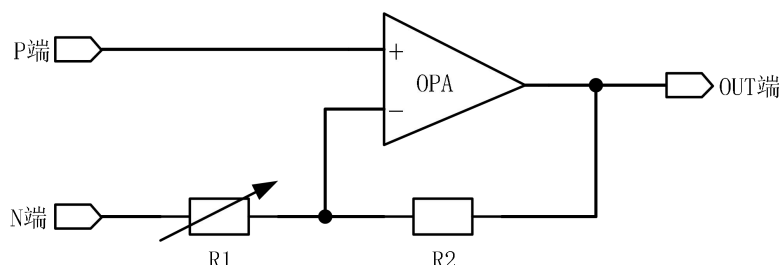


图 17.2 PGA 等效电路

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：

$$V_{CMO} = \left[\frac{(V_P - V_N)}{R_1} \right] \times R_2 + V_P$$

PGA 同相放大倍数 A_{VP} 计算公式为：

$$A_{VP} = \frac{(R_1 + R_2)}{R_1}$$

PGA 反相放大倍数 A_{VN} 计算公式为：

$$A_{VN} = -\frac{R_2}{R_1}$$

PGA 的放大倍数的调整，可通过修改寄存器 0x7106 配置调整 R_1 的阻值实现。在 PGA 不同放大倍数下， R_1 与 R_2 的近似阻值如表 17.1 所示。

表 17.1 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 (k Ω)	R2 电阻值 (k Ω)
2	-1	22	22
4	-3	22/3	22
8	-7	22/7	22
10	-9	22/9	22
16	-15	22/15	22

除了 3 个带反馈回路的可编程放大器外，该系列器件还集成了一个 3 端口均引出的独立运算放大器 OPA，如图 17.1 所示。此 OPA 通过 B0_OP1OUT 输出，输出端口同时连接到 ADCINB0 通道。默认情况下，寄存器 0x7106 为全 0 配置，OPA 功能处于关闭状态。

如图 17.1 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 位为 1 时可打开对应数字信号输入功能。

17.2 运放与复用数字 IO 输入寄存器

17.2.1 通道对应的配置寄存器

1. 通道对应的配置寄存器—7103h

15	14	13	12	11	10	9	8
COMP_H	COMP_L	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	COMP_L：比较器 COMP_L 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 ADCINB7 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 ADCINB6 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 ADCINB5 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 ADCINB4 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 ADCINB3 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 ADCINB2 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 ADCINB1 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 ADCINB0 1: 配置端口为数字 IO 输入端口 IOPF4

17.2.2 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_PGA1	REG_PGA2	REG_PGA3	保留	SETAG_P	SETAG_P	SETAG_PGA
	CT	CT	CT		GA1_B2	GA2_B1	3_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_PGA	SETAG_PGA	SETAG_PGA	保留	SETAG_P	SETAG_P	SETAG_PGA
	2_B2	2_B1	2_B0		GA1_B2	GA1_B1	1_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 17.2 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，不修调失调值 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

18 电压比较器

18.1 电压比较器说明

该系列器件包含 5 个迟滞比较器模块，如图 18.1 所示为比较器模块内部框架图。

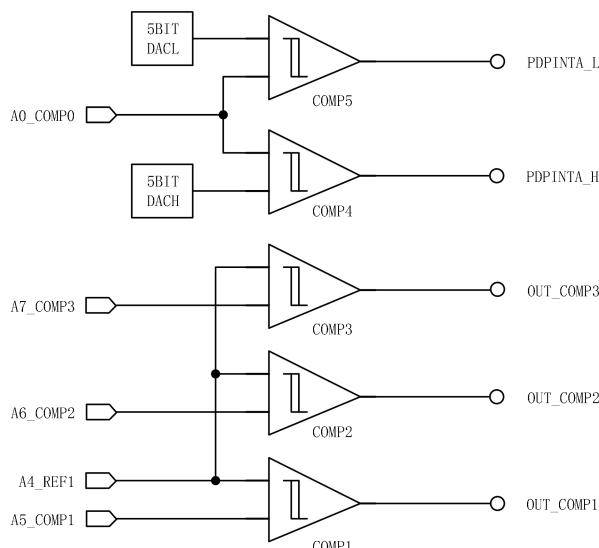


图 18.1 比较器模块内部框架图

其中 A0_COMP0、A7_COMP3、A6_COMP2、A4_REF1、A5_COMP1 为芯片外部输入引脚，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 与 COMP5 为内阈值组; 5 个比较器均为迟滞比较器, 高阻抗输入, 迟滞电压可通过寄存器 753Ch 进行配置。

18.1.1 内阈值电压比较器

COMP4、COMP5 为内阈值组电压比较器, 参考基准连接到内部 5bit DAC 输出, 5bit DAC 分别输出高、低阈值, A0_COMP0 信号与高、低阈值进行比较, 产生两组数字触发信号, 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 引

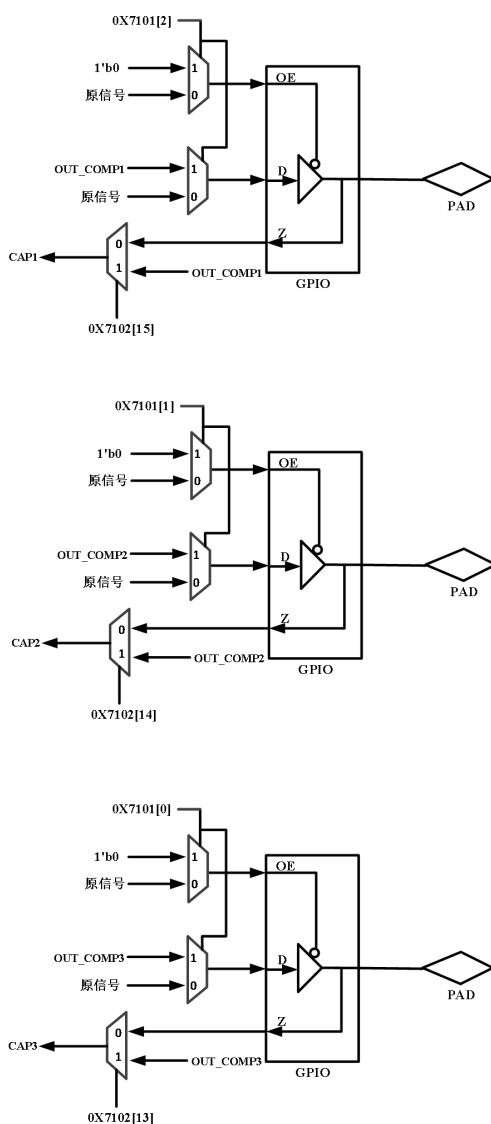


图 18.3 外阈值组电压比较器组框图

18.1.3 电压比较器控制寄存器

1. 电压比较器控制寄存器 1—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
R_0	R_0	R_0	WR_0	WR_0	WR_0	WR_0	WR_0
7	6	5	4	3	2	1	0
ADCT1BH	ADCT1BL_	保留	保留	保留	COMP1_C	COMP2_C	COMP3_C

_CTL	CTL				T	T	T
WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0	WR_0

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—7102h

15	14	13	12	11	10	9	8
REG_COM P1CT	REG_COM P2CT	REG_COM P3CT	REG_COM PHCT	REG_COM PLCT	REG_DAC CT	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 位 1: 打开比较器 COMPL 位
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; $V_{out} = \frac{V_{CCA}}{32} \times CODE5$, 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; $V_{out} = \frac{V_{CCA}}{32} \times CODE5$, VCCA 典型值 3.0V

3. COMPH/COMPL 状态寄存器—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_INPU T_CT7	REG_INPU T_CT6	REG_INPU T_CT5	REG_INPU T_CT4	REG_INPU T_CT3	REG_INPU T_CT2	REG_INPU T_CT1	REG_INPU T_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 12~位 8	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 GPIOE5
位 6	REG_INPUT_CT6: B6_OP4P 复用为数字输入端口使能 0: B6_OP4P 为模拟信号输入端口 1: B6_OP4P 为数字输入端口, 可输入数字信号至 GPIOE6
位 5	REG_INPUT_CT5: B5_OP3N 复用为数字输入端口使能 0: B5_OP3N 为模拟信号输入端口 1: B5_OP3N 为数字输入端口, 可输入数字信号至 GPIOE7
位 4	REG_INPUT_CT4: B4_OP3P 复用为数字输入端口使能 0: B4_OP3P 为模拟信号输入端口 1: B4_OP3P 为数字输入端口, 可输入数字信号至 GPIOF0
位 3	REG_INPUT_CT3: B3_OP2N 复用为数字输入端口使能 0: B3_OP2N 为模拟信号输入端口 1: B3_OP2N 为数字输入端口, 可输入数字信号至 GPIOF1
位 2	REG_INPUT_CT2: B2_OP2P 复用为数字输入端口使能 0: B2_OP2P 为模拟信号输入端口 1: B2_OP2P 为数字输入端口, 可输入数字信号至 GPIOF2
位 1	REG_INPUT_CT1: OP1N 复用为数字输入端口使能 0: OP1N 为模拟信号输入端口 1: OP1N 为数字输入端口, 可输入数字信号至 GPIOF3
位 0	REG_INPUT_CT2: B3_OP1P 复用为数字输入端口使能 0: OP1P 为模拟信号输入端口 1: OP1P 为数字输入端口, 可输入数字信号至 GPIOF4

4. COMP 配置寄存器—753Ch

15	14	13	12	11	10	9	8
REG_PLL3	REG_PLL2	REG_PLL1	REG_PLL0	保留	保留	REG_CPL_HYTU1	REG_CPL_HYTU0
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_HYTU1	REG_CPH_HYTU0	REG_CP3_HYTU1	REG_CP3_HYTU0	REG_CP2_HYTU1	REG_CP2_HYTU0	REG_CP1_HYTU1	REG_CP1_HYTU0
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 12	REG_PLL3~REG_PLL0: 时钟不交叠时间控制位, 出厂已校准, 禁止修改此值
位 11	保留
位 10	保留
位 9~位 8	REG_CPL_HYTU1~ REG_CPL_HYTU0: 比较器 COMPL 迟滞电压调整控制位
位 7~位 6	REG_CPH_HYTU1~ REG_CPH_HYTU0:

	比较器 COMPH 迟滞电压调整控制位
位 5 ~ 位 4	REG_CP3_HYTU1 ~ REG_CP3_HYTU0: 比较器 COMP3 迟滞电压调整控制位
位 3 ~ 位 2	REG_CP2_HYTU1 ~ REG_CP2_HYTU0: 比较器 COMP2 迟滞电压调整控制位
位 1 ~ 位 0	REG_CP1_HYTU1 ~ REG_CP1_HYTU0: 比较器 COMP1 迟滞电压调整控制位

表 18. 1 COMP 迟滞电压配置关系

REG_CPx_HYTU1	REG_CPx_HYTU0	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 COMPL, COMPH, COMP3, COMP2, COMP1。

19 数字延时滤波

19.1 数字延时滤波模块结构

数字延时滤波单元, 可用作 CAP1~CAP3, XINT1, XINT2/ADCSOC 以及 PDPINTA 等信号输入滤波; CAP1, CAP2, CAP3 配置控制位共用, PDPINTA 单独配置控制, XINT1 和 XINT2 配置控制位共用。

输入电平在满足跳变后持续 6 (可通过数字滤波模块寄存器设置) 个及以上时钟周期的高/低状态时, 信号才被有效捕捉。

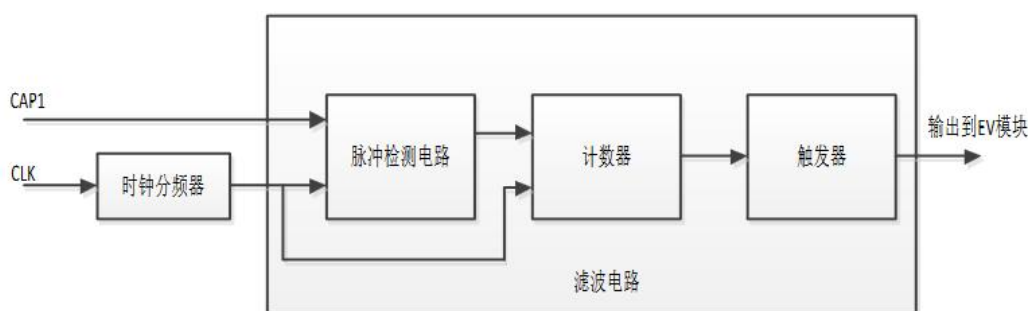


图 19.1 CAP1 数字延时滤波示意图

以 CAP1 信号为例, 图 19.1 为 CAP1 数字延时滤波示意图, 可通过 bit[4:0]配置延时电路的计数时钟, 得到不同的延时滤波时间。表 19.1 为 150M 主频下延时滤波时间参考配置表。

表 19.1 延时滤波时间参考配置表

Bit[4:0]	周期数	滤波宽度@150MHz
00000	6	40.02ns
00001	12	80.04ns
00010	20	133.4ns
00011	40	266.8ns
00100	60	400.2ns
00101	80	533.6ns
00110	100	667ns

00111	120	800.4ns
01000	140	933.8ns
01001	160	1067.2ns
01010	180	1200.6ns
01011	200	1334ns
01100	220	1467.4ns
01101	240	1600.8ns
01110	260	1734.2ns
01111	280	1867.6ns
10000	300	2001ns
10001	320	2134.4ns
10010	340	2267.8ns
10011	360	2401.2ns
10100	380	2534.6ns
10101	400	2668ns
10110	420	2801.4ns
10111	440	2934.8ns
11000	460	3068.2ns
11001	480	3201.6ns
11010	500	3335ns
其它	500	3335ns

滤波模块配置如上表所示，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 500 个时钟周期。

19.2 数字延时滤波模块寄存器说明

19.2.1 数字延时滤波控制寄存器

1. 数字滤波控制寄存器—753Fh

15	14	13	12	11	10	9	8
CfgError	XINTCTL_	XINTCTL_	XINTCTL_	XINTCTL	XINTCTL	PDPINTCT	PDPINTCTL_R
	REG4	REG3	REG2	_REG1	_REG0	L_REG4	EG3
R_0	R_X	R_X	R_X	RW_0	RW_0	RW_0	RW_0

7	6	5	4	3	2	1	0
PDPINT CTL_REG 2	PDPINTCT L_REG1	PDPINTCT L_REG0	CAPCTL_R EG4	CAPCTL_ REG3	CAPCTL_ REG2	CAPCTL_R EG1	CAPCTL_REG 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 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，详细参考表 18-1
位 9 ~ 位 5	PDPINTCTL_REG4 ~ PDPINTCTL_REG0: PDPINTA 延时电路计数时钟设置，详细配置参考表 18-1
位 4 ~ 位 0	CAPCTL_REG4 ~ CAPCTL_REG0: CAP1 ~ CAP3 延时电路计数时钟设置，详细配置参考表 18-1

20.1 工作模式

如图 20.1 所示, ADM16F03A2 LIN 模块主要有四种工作模式, 分别为上电模式、待机模式、正常模式和休眠模式。

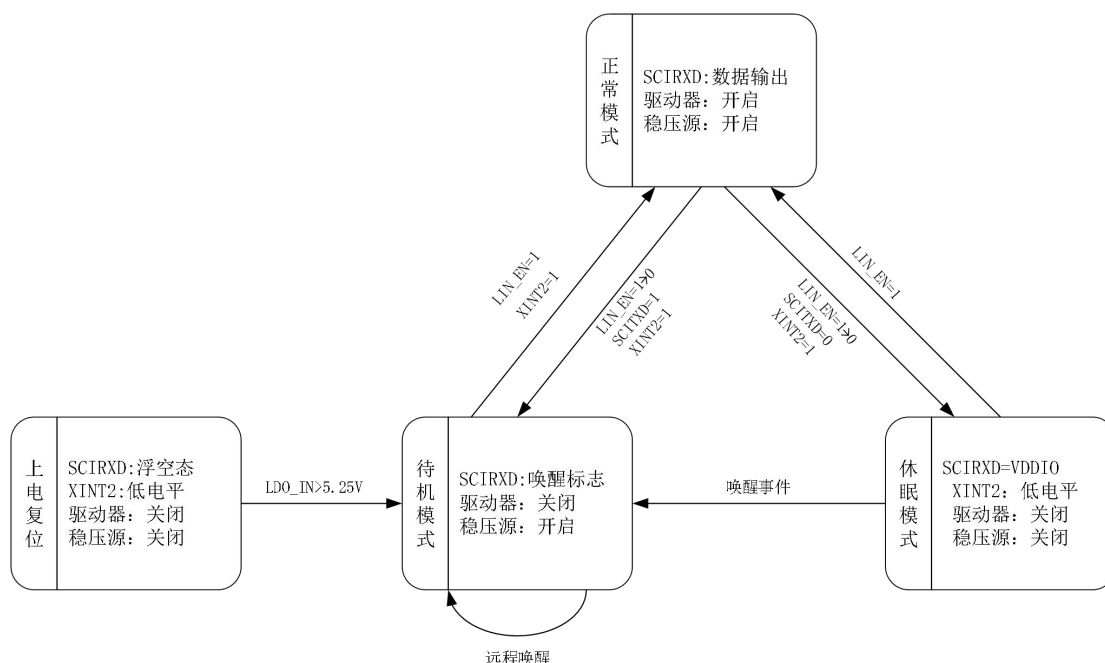


图 20. 1 ADM16F03A2 芯片 LIN 模块工作模式

20.1.1 上电模式

当 LDO_IN 电压小于掉电检测电压阈值 4.7V 或发生过温保护事件时，ADM16F03A2 将处于上电模式，稳压源和 LIN 收发器都被关闭，XINT2 = 0，可通过读取 XINT2 引脚电平值来对稳压源的供电情况进行检测，若中断使能，可在中断服务例程中处理。

20.1.2 待机模式

ADM16F03A2 可通过其他三种工作模式切换进入待机模式，在待机模式下稳压源仍保持开启状态，LIN 收发器被关断，总线远程唤醒功能被使能（以 SCIRXD = 0 作为唤醒标志）。

当 LDO_IN 电压大于上电检测电压阈值 5.25V 且未发生过温保护事件时，ADM16F03A2 将由上电模式自动跳转至待机模式。当 ADM16F03A2 处于正常模式时，在模式选择窗口期间，若 LIN_EN = 0，SCITXD = 1 且 XINT2=1，ADM16F03A2 将由正常模式跳转至待机模式（如图 20.2 所示）。

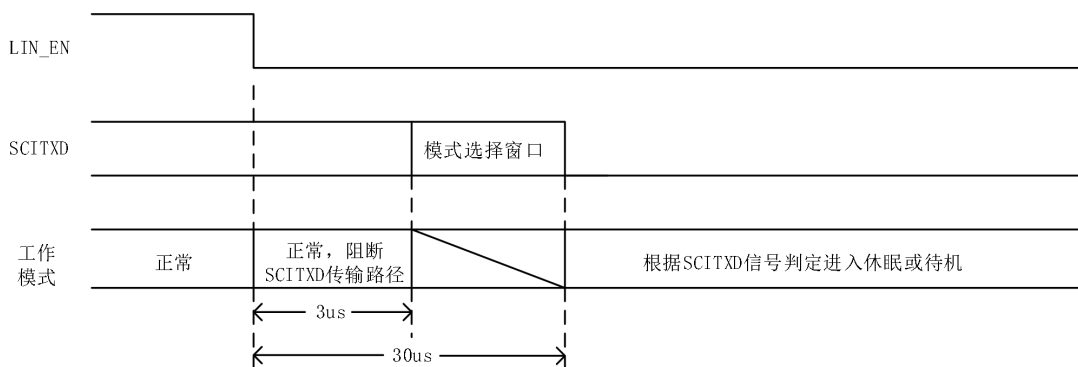


图 20.2 正常至待机/休眠跳转时序图

当 ADM16F03A2 处于休眠模式时，可通过总线远程唤醒进入待机模式，远程唤醒时序如图 20.3 所示。

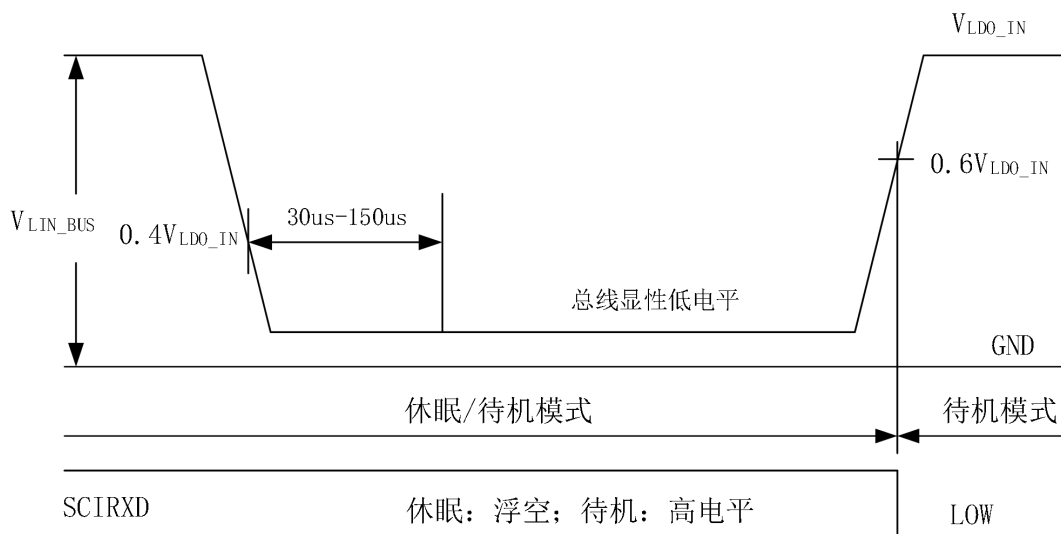


图 20.3 远程唤醒时序图

20.1.3 正常模式

正常上电启动后 ADM16F03A2 先进入待机模式, 在 $XINT2 = 1$ 的条件下将 LIN_EN 拉高则 ADM16F03A2 由待机模式切换至正常模式; 若 ADM16F03A2 处于休眠模式, 则将 LIN_EN 拉高后可切换至正常模式。在正常模式下, 稳压源开启, LIN 收发器开启, ADM16F03A2 可正常通过 LIN 总线发送和接收数据。此时微控制器可通过 SCITXD 引脚将数据发送至 LIN 总线, 实现低压逻辑电平信号转换成高压 VLDO_IN (电池) 电平信号 (高电平代表隐性, 低电平代表显性), 并进行总线波形整形以抑制电磁发射 (EME)。

同时, 总线上的信号可被 LIN 引脚接收并通过 ADM16F03A2 的 LIN 接收器输入至 SCIRXD 引脚, 转换成可供内部 CPU 处理的低压逻辑电平数据。

20.1.4 休眠模式

该模式是 ADM16F03A2 功耗最低的模式, 此时 5V 稳压源、LIN 收发器和 12V 稳压源均被关闭, XINT2 被强制拉低, 可通过 LIN 总线对其进行远程唤醒。

如图 20.2 所示，休眠模式只能在正常模式下切换进入。在模式选择窗口期间，若 $LIN_EN = 0$ ， $SCITXD = 0$ 且 $XINT2 = 1$ ，ADM16F03A2 将由正常模式跳转至休眠模式（如图 20.2 所示）。

20.1.5 正常至休眠/待机模式的状态切换

如图 20.2 所示，当 LIN_EN 拉低后 ADM16F03A2 先将 SCITXD 至 LIN 的传输路径阻断，等待 3us 时长后进入模式选择窗口，此时若 $SCITXD = 1$ 则进入待机模式，若 $SCITXD = 0$ 则进入休眠模式。 LIN_EN 拉低后总的模式选择时间为 30us。

20.2 远程唤醒

LIN_BUS 引脚上的远程唤醒：当 LIN_BUS 引脚通过一个下降沿拉低至低电平后，下一时刻出现一个上升沿，且该上升沿与前一时刻下降沿之间的低电平维持时间大于 30us~150us，则该过程被视为一个有效的远程唤醒（如图 20.3 所示）。发生远程唤醒后 SCIRXD 被置为低电平以表示唤醒标志。

20.3 显性超时功能

如果 SCITXD 引脚因硬件和（或）软件应用故障而被强制为永久低电平时，内置的 SCITXD 显性超时定时器电路可防止总线线路被驱动至永久显性状态（阻塞所有网络通信）。定时器由 SCITXD 引脚上的下降沿触发。如果 SCITXD 引脚上的低电平维持时间超过内部定时器时间 6~20ms，发送器将被禁用，驱动总线进入隐性状态。定时器通过 SCITXD 引脚上的上升沿复位。

20.4 过温保护功能

当 ADM16F03A2 处于正常或待机模式下发生过温事件时，ADM16F03A2 将被切换至上电模式，关闭稳压源和 LIN 收发器模块。当温度下降至过温保护恢复阈值时，ADM16F03A2 将从上电模式跳转至待机模式。

20.5 失效安全特性

- 总线驱动器输出级限流，以防止总线短路至 VLDO_IN 时烧毁驱动器或功能性影响。
- 掉电掉地情况对总线端口无影响，总线端口无反向电流。
- 在 LIN_EN 或 XINT2 为低电平情况下，LIN 驱动器将自动关闭。
- 切换到正常模式后，只有在检测到 SCITXD 出现高电平时 LIN 驱动器才会被使能。

21 供电

21.1 片上 LDO 电源管理

该系列器件包含 4 个电压域：12V 电压域 (VDRV)，3.2 ~ 5.5V 电压域 (VDDIO)，1.2V 电压域 (VDD_1V2)，3V 电压域 (VCCA)。芯片内部集成多个 LDO 电路，支持单电源供电，仅需对 LDO_IN 输入 4.5~40V 电源，VDRV、VOOIO、VDD_1V2、VCCA 电源引脚与地之间分别连接去耦电容，去耦电容应尽可能排布靠近器件。图 21.1 展示了芯片内各模块（除 GPIO 外）的供电关系，GPIO 驱动电源为 VDDIO。

片内集成的四个 LDO，分别：

一个为片内预驱供电的 LDO，输入 4.5~40V 输出 12V；

一个为数字 I/O 供电的 LDO，输入 12V 输出 5V；

一个为 ADC 供电的 LDO_3 ，输入 3.2 ~ 5.5V 输出 3V，为获得较好的电源纹波抑制效果， LDO_3 供电引脚应接大于一个 2.2uF 并联一个 0.1uF 的去耦电容；

一个为内核和锁相环供电 LDO_{CK} ，输入 3.2 ~ 5.5V 输出 1.2V，为获得较好的电源纹波抑制效果， LDO_{CK} 引脚应接一个大于 2.2uF 并一个 0.1uF 的去耦电容。

LDO_3 ， LDO_{CK} 参考基准均来源于内部带隙基准输出的 1.2V， LDO_{CK} 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。1.2V 电压基准由 1V 基准经过运放比例缩放后获得。

图 21.1 中二极管 D1 降低 5V 供电电压到 4V 以上 5V 以下，确保 SPI 和 3.3V 的设备正常通信。

V_{bus} 供电范围为 4.5V~40V。布板时，建议数字地、模拟地和预驱地分开单点连接。

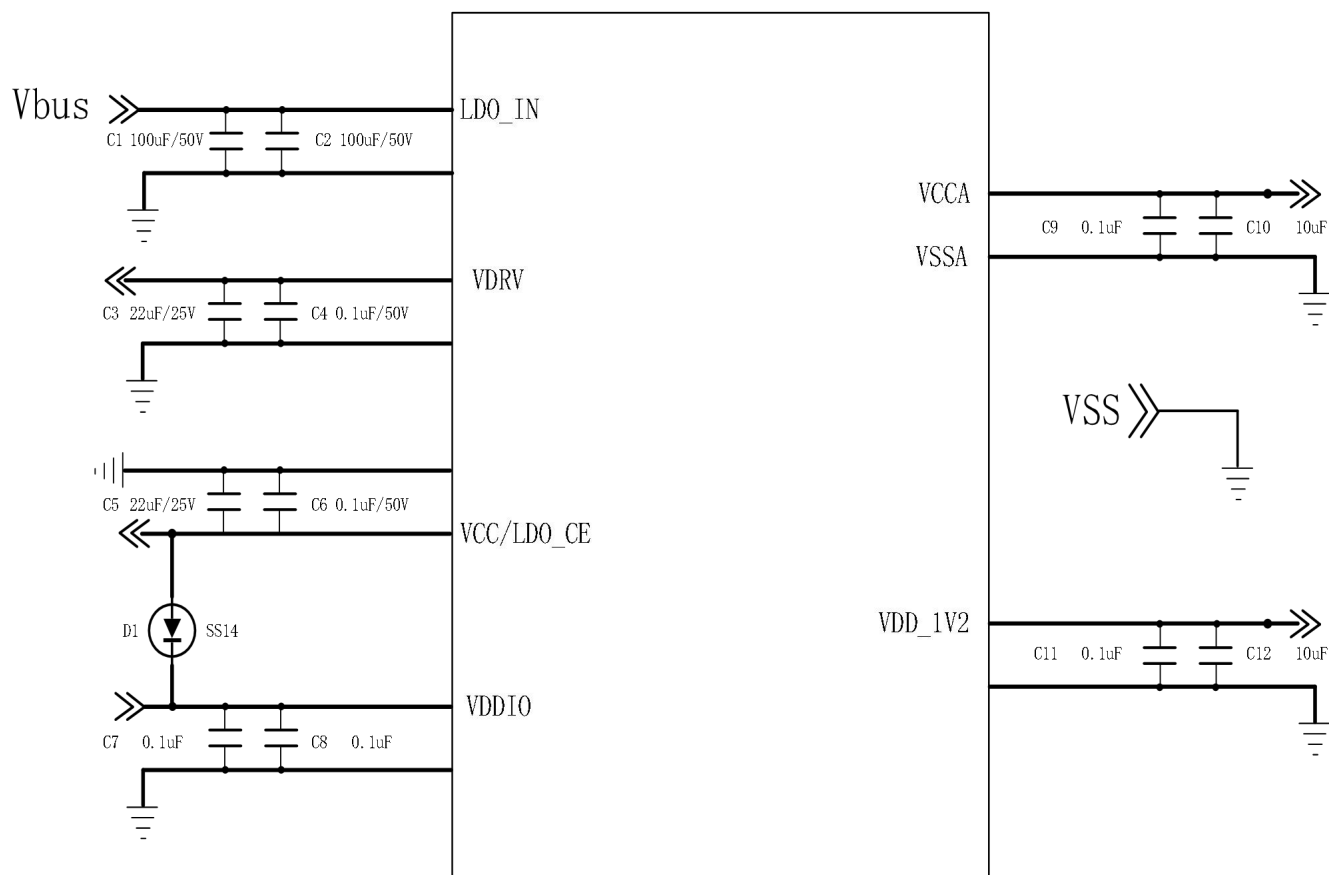


图 21.1 芯片供电连接参考图示

22 典型应用

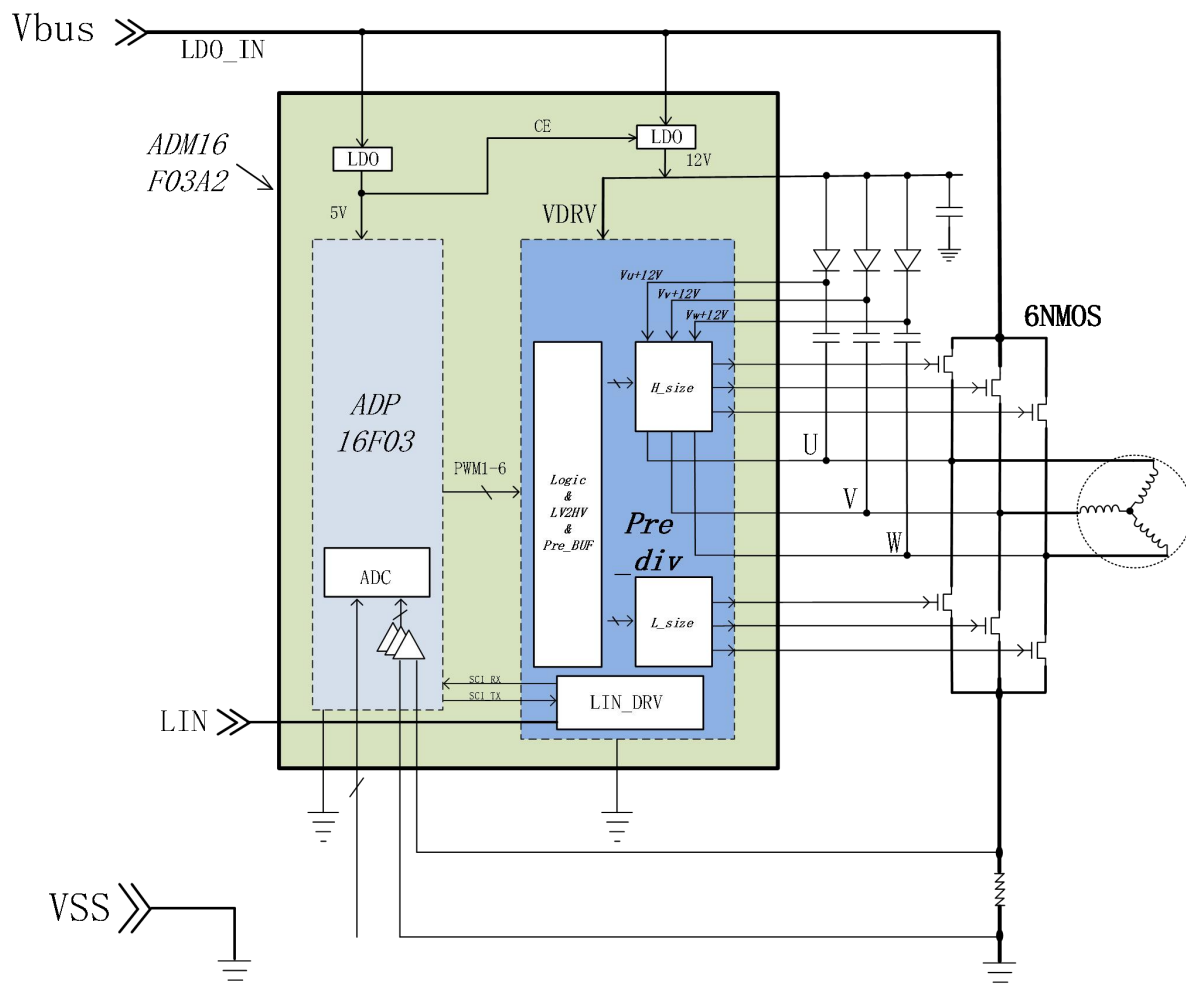


图 22.1 驱动模块典型应用图

典型应用图中，自举电容的值根据具体应用设置，MOS 管 G 级的限流电阻用于调节驱动信号的上升时间。驱动模块的输出引脚信号 HO1/LO1 对应 DSP 的 PWM1/PWM2 功能输出, HO2/LO2 对应 DSP 的 PWM3/PWM4 功能输出, HO3/LO3 对应 DSP 的 PWM5/PWM6 功能输出。

需注意的是，控制驱动模块的 LOx/HOx 的各 PWM 信号，均为高电平有效。驱动模块自带 0.5μs 硬件死区，可防止 H 桥直通。

只需要添加 DC/DC、LDO 辅助电源和少量的外围器件, 即可实现 BLDC/PMSM 等电机的有感/无感、方波/FOC 控制。

23 电气参数

23.1 建议的运行条件

		最小值	标称值	最大值	单位
V_{LDO_IN}	母线输入电压	4.5	12	40	V
V_{DDIO}	控制器 IO 电源电压, I/O	3.2	3.3	5.5	V
V_{DD_1V2}	控制器内核源电压, CPU		1.2		V
VSS	电源接地		0		V
V_{CCA}	ADC 电源电压/ADC 参考基准		3		V
f_{CLKOUT}	设备时钟频率 (系统时钟)			150	MHz
$V_{IH}^{③}$	高电平输入电压	所有输入	$0.7 \cdot V_{DDIO}$	$V_{DDIO} + 0.3$	V
V_{IL}	低电平输入电压	所有输入	$VSS - 0.3$	$0.3 \cdot V_{DDIO}$	V
I_{OH}	高电平输出源电流	所有 GPIO		4	mA
		所有 GPIO		4	
		所有 GPIO		4	
I_{OL}	低电平输出灌电流	所有 GPIO		4	mA
		所有 GPIO		4	
		所有 GPIO		4	
N_f	阵列的闪存耐久性 (写/擦除周期)	-40°C至 85°C	100000		周期

23.2 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		IOH=IOH 最大值	VDDIO-0.8			V
		IOH=50uA	VDDIO-0.2			
V _{OL} 低电平输出电压		IOL=IOL 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	±2			μA
	上拉不使能	或 5V	±2			uA
I _{OZ} 输出电流, 高阻抗状态 (关闭状态)		Vo=VDDIO 或 0	±2			μA
TRSTN 下拉电阻			15			kΩ
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		IOH=IOH 最大值	VDDIO-0.8			V
		IOH=50uA	VDDIO-0.2			
V _{OL} 低电平输出电压		IOL=IOL 最大值	0.4			V

I_{OZ} 输出电流, 高阻抗状态 (关闭状态)	$V_O = V_{DDIO}$ 或 0	± 2	μA
-----------------------------	----------------------	---------	---------

(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 $\pm 2\mu A$ 。

23.3 减少流耗

该系列器件可关闭不使用的外设模块, 达到节省器件流耗的目的。

23.4 信号转换电平

器件应用过程中应主意一些信号使用不同的参考电压, 详细可参考建议的工作条件。

输出逻辑高电平的下限值为 2.4 V, 而输出逻辑低电平的上限值为 0.8 V。图 23.1 展示了输出电平有效范围。

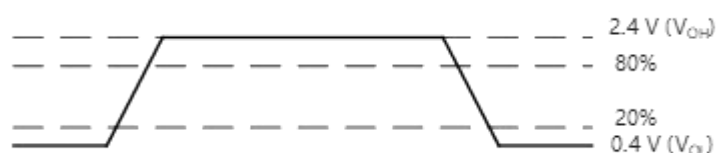


图 23.1 GPIO 输出电平有效范围

输出转换时间定义如下:

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高于 V_{OL} (最大值) 的 20%。

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低于 V_{OH} (最小值) 的 80%。

图 23.2 展示了输入电平的有效范围。

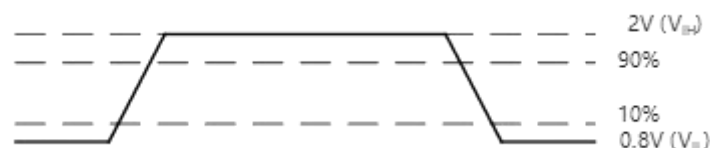


图 23.2 GPIO 输入电平有效范围

输入转换时间指定如下:

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高于 V_{IL} （最大值）的 10%。

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低于 V_{IH} （最小值）的 90%。

23.5 时序参数符号

器件规格书中所使用的时序参数符号均参考 JEDEC-100 标准创建。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：		大写字母和符号和它们的含意：	
a	访问时间	H	高
c	周期时间（周期）	L	低
d	延迟时间	V	有效
f	下降时间	X	未知变化或无关电平
h	保持时间	Z	高阻抗
r	上升时间		
su	建立时间		
t	转换时间		
v	有效时间		
w	脉冲持续时间（宽度）		

23.6 定时参数的通用注释

该系列器件的输出信号（包括 CLKOUT）取自一个内部时钟，所以对于一个指定半周期的所有输出转换在一个相对最小时间偏移下发生。下面时序图中的信号组合也许不一定代表真实的周期。对于真实周期范例，可参考本文档中有关周期的补充说明。

23.7 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。这些隔离的电源设计是为了防止数字逻辑电路 VSS 和 VCC 产生的噪声耦合到模拟模块。在无补充说明，ADC 电平参数以 VSSA 为参考。

23.8 ADC 建议运行条件

表 23.1 ADC 建议运行条件

		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_{SSA2}	模拟地		0		V
V_{AI}	模拟输入电压，ADCINA0—ADCINB7	0		3	V

23.9 ADC 工作频率设定

表 23.2 ADC 工作频率范围

	最小值	最大值	单位
ADC 运行频率	0.1	14	MHZ

23.10 推荐工况下的 ADC 电气特性

表 23.3 推荐工况下的 ADC 电气特性

参数	测试条件	最小值	典型值	最大值	单位
I_{CCA} 模拟供电电流	$V_{CCA} = 3V$				
I_{ADCIN} 模拟输入电流				± 2	μA
C_{ai} 模拟输入电容	非采样时		10		pF
	采样时		13.5		pF
E_{DNL} 微分非线性误差（3）（4）	实际步长与理想值之间的误差			± 2	LSB
E_{INL} 积分非线性误差（4）	数模转换特性的最大偏移误差，不包含量化误差			± 4	LSB
$t_{d(PU)}$ ADC 上电使能后延迟时间	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 所需模拟输入源阻抗				Ω
Offset 偏移误差		-15		15	LSB
片内基准时总增益误差		-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(ADCIN _x 对 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 信噪比+失真(7.8kHz)			61.9		dB
SNR 信噪比(7.8kHz)			62.2		dB
THD 总谐波失真(7.8kHz)			-73		dB
ENOB 有效位数(7.8kHz)			10		Bit
SFDR 无杂散动态范围(7.8kHz)			74		dB

- (1) 在 14MHz ADCCLK 上测得。
- (2) 这个表中的所有电压相对 V_{SSA} 。
- (3) ADC 无丢码。
- (4) 1 个 LSB 代表 $3/4096=0.732$ 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} 或低于 V_{SSA} 的电压会影响 ADC 的性能参数。

23.11 预驱模块参数

23.11.1 最大额定值

参数名称	符号	推荐值			单位
		最小值	典型值	最大值	
上桥臂自举电源	V_B	$V_S-0.3$		V_S+20	V
上桥臂悬浮端	V_S	-0.3		200	V
上桥臂驱动输出电压	V_{HO}	$V_B-0.3$		$V_B+0.3$	V
电源	V_{DRV}	-0.3		20	V
下桥臂驱动输出电压	V_{LO}	-0.3		$V_{CC}+0.3$	
最大功耗	P_D			800	mW
工作结温范围	T_j			150	

储存温度范围	T_{stg}	-50		150	
ESD_HBM			2		KV

注:

- (1) 器件运行条件超过上述各项最大额定值时可能造成永久性损伤。上述参数仅是运行条件的极大值，禁止将器件使用在超出该规范下运行。如器件长时间工作在最大极限条件下，将影响器件运行稳定性。
- (2) 无特殊说明，所有的电压均以 GND 作为参考。

23.11.2 推荐工作条件（无特别说明情况下，测试条件为 $T_A=25^{\circ}\text{C}$ ）

参数名称	符号	推荐值			单位
		最小值	典型值	最大值	
电源	V_{DRV}	10	15	18	V
上桥臂悬浮端	V_S	24		180	V
上桥臂自举电源	V_B	V_S+10	V_S+15	V_S+18	V
上、下桥臂负载电容	C_L			22	nF
上、下桥臂输入电平	V_{IN}	0	3.3	5.0	V
工作温度	T_A	-40		125	$^{\circ}\text{C}$
结温	T_J	-40		150	$^{\circ}\text{C}$

23.11.3 静态电气参数

静态电气参数（无特别说明，测试条件为 $V_{BIAS}=V_{DRV}=V_B=15\text{V}$ ， $T_A=25^{\circ}\text{C}$ ）

参数名称	符号	推荐值			单位
		最小值	典型值	最大值	
电源电压	V_{DRV}	10	15	20	V
静态电流 $V_{IN}=0\text{V}/5\text{V}$	I_{QCC}		210	300	μA
上桥臂自举静态电流 $V_{IN}=0\text{V}/5\text{V}$	I_{QBS}		320	400	μA
输入高电平	V_{IH}	2.5			nF
输入低电平	V_{IL}			0.8	V
上桥臂输出高电平 $I_{O+}=10\text{mA}$	V_{HOH}			100	mV
上桥臂输出低电平 $I_{O-}=10\text{mA}$	V_{HOL}			100	mV
下桥臂输出高电平 $I_{O+}=10\text{mA}$	V_{LOH}			100	mV
下桥臂输出低电平 $I_{O-}=10\text{mA}$	V_{LOL}			100	mV
逻辑 1 的输入电流 $HIN=5\text{V}, LIN=0\text{V}$	I_{N+}		85		μA

逻辑 0 的输入电流 $HIN = 0V, LIN = -5V$	I_{N-}		85		uA
欠压保护上点	VCC_{UV+}		4.5		V
欠压保护下点	VCC_{UV-}		4.2		V
输出灌电流, 驱动高输出对地短路 $P_W \leq 10\mu S$	I_{O+}	0.8	1		A
输出拉电流, 驱动低输出对地短路 $P_W \leq 10\mu S$	I_{O-}	1.3	1.5		A

23.11.4 动态电气参数 (无特别说明情况下, $V_{BIAS}=V_{DRV}=V_B=15V$, $C_L=1000pF$, $T_A=25^\circ C$)

参数名称	符号	推荐值			单位
		最小值	典型值	最大值	
高端输出 HO 开关时间特性					
上升延时	T_{on}		300		ns
下降延时	T_{off}		100		ns
上升时间	T_r		40		ns
下降时间	T_f		20		ns
低端输出 LO 开关时间特性					
上升延时	T_{on}		160		ns
下降延时	T_{off}		55		ns
上升时间	T_r		35		ns
下降时间	T_f		20		ns
死区时间特性					
死区时间	DT		200		ns
上升和下降死区时间差值	MT		10		ns

24 机械数据

VQFN56

单位: mm

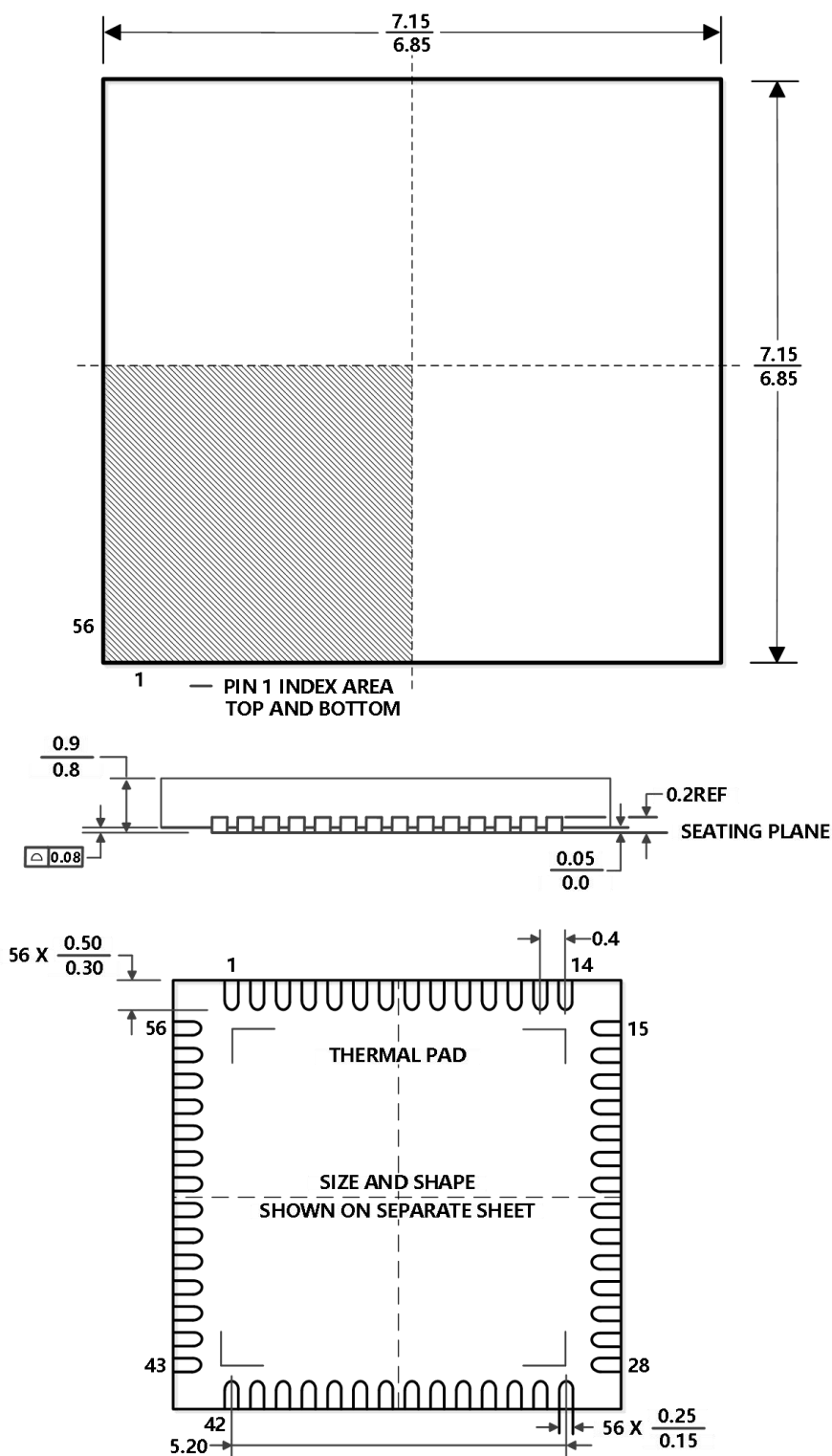
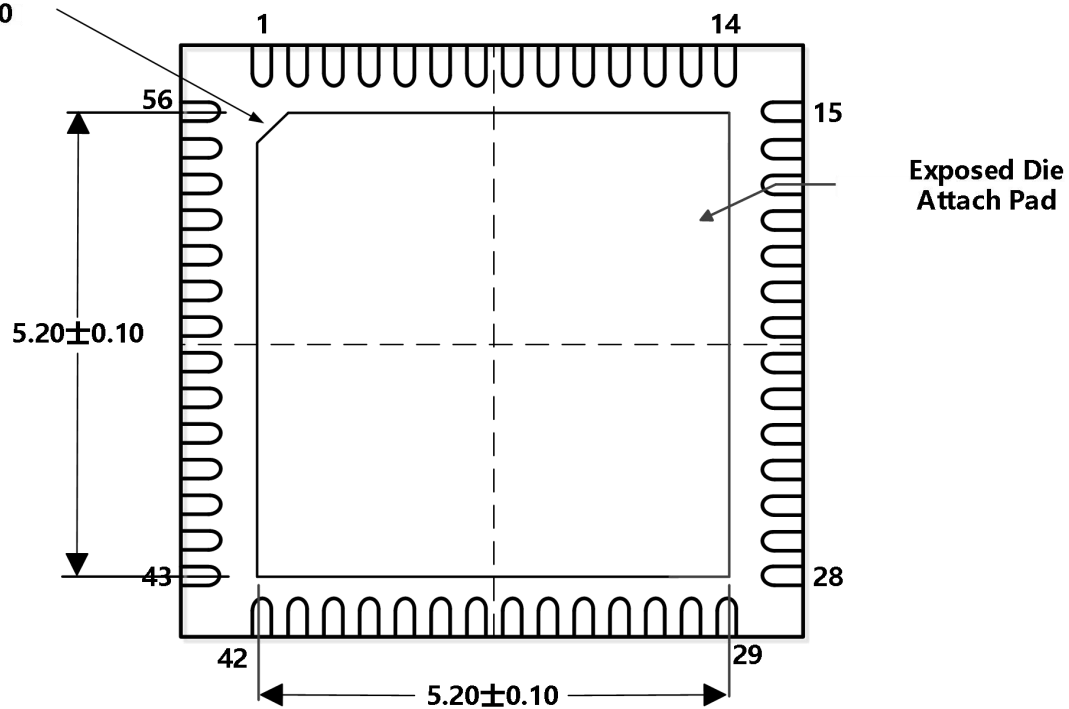


图 24.1 VQFN56 封装外形尺寸图

PIN 1 INDICATOR
0.30



Bottom View

Exposed Die Attach Pad Dimensions

图 24. 2 VQFN56 封装外形尺寸图

25 重要注意事项及声明

Advancechip 均以“原样”提供技术性 & 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

联系电话: 0731-88731027 (长沙)

总部地址: 长沙高新开发区东方红北路 601 号湖南媒体艺术产业园 A5 栋

销售中心: 南京市秦淮区卡子门大街 19 号 6 号楼 15 层

