



# 精密模拟微控制器，12位模拟I/O， 具有增强型IRQ处理程序的ARM7TDMI MCU

## ADuC7023

### 特性

#### 模拟I/O

多通道、12位、1 MSPS ADC

多达12个ADC通道

全差分模式和单端模式

模拟输入范围：0 V至 $V_{REF}$

12位电压输出DAC

提供4个DAC输出

片内基准电压

片内温度传感器

电压比较器

#### 微控制器

16位/32位RISC架构ARM7TDMI内核

JTAG端口支持代码下载和调试

#### 时钟选项

修正的片内振荡器( $\pm 3\%$ )

外部时钟晶体

可达44 MHz的外部时钟源

具有可编程分频器的41.78 MHz锁相环

#### 存储器

62 kB Flash/EE存储器，8 kB SRAM

在线下载，基于JTAG调试

软件触发在线重新编程能力

#### 用于FIQ和IRQ的矢量中断控制器

每类中断支持8种优先级

边沿或电平中断外部引脚输入

#### 片内外设

2个完全I<sup>2</sup>C兼容通道

SPI(主模式下20 Mbps，从模式下10 Mbps)

输入级和输出级具有4字节FIFO

多达20个GPIO引脚

所有GPIO均兼容5 V电压

3个通用定时器

看门狗定时器(WDT)

可编程逻辑阵列(PLA)

16个PLA元件

16位、5通道PWM

#### 电源

额定工作电压：3 V

主动模式：11 mA(5 MHz)，28 mA(41.78 MHz)

#### 封装和温度范围

32引脚5 mm × 5 mm LFCSP封装

40引脚LFCSP

额定工作温度范围：-40°C至+125°C

#### 工具

低成本QuickStart开发系统

完全第三方支持

### 应用

光纤网络

工业控制和自动化系统

智能传感器、精密仪器

基站系统

### 概述

ADuC7023是一款完全集成的1 MSPS、12位数据采集系统，在单芯片内集成高性能多通道ADC、16位/32位MCU和Flash/EE存储器。

ADC具有多达12路单端输入。另外还有4个ADC输入通道也可以和4个DAC的输出引脚复用。ADC可以在单端模式或差分输入模式下工作，ADC输入电压范围为0 V至 $V_{REF}$ 。低漂移带隙基准电压源、温度传感器和电压比较器完善了ADC的外设设置。

通过编程可将DAC输出范围设置为两种电压范围之一。DAC输出具有一个增强特性，能够在看门狗或软件复位时序中保持其输出电压。

这些器件通过一个片内振荡器和锁相环(PLL)产生41.78 MHz的内部高频时钟信号。该时钟信号通过一个可编程时钟分频器进行中继，在其中产生MCU内核时钟工作频率。微控制器内核为ARM7TDMI<sup>®</sup>，它是一个16位/32位RISC机器，峰值性能最高可达41 MIPS。片内集成有8 KB SRAM和62 KB非易失性Flash/EE存储器。ARM7TDMI内核将所有存储器和寄存器视为一个线性阵列。

ADuC7023内置一个高级中断控制器。该矢量中断控制器(VIC)可以为每个中断分配一个优先级。它还支持嵌套中断，每个IRQ和FIQ最多允许8级嵌套。如果将IRQ和FIQ中断源合并，则可以支持总计16级嵌套中断。

片内出厂固件支持通过I<sup>2</sup>C串行接口端口进行在线下载，并且支持通过JTAG接口进行非介入仿真。这些特性都集成在支持此MicroConverter<sup>®</sup>系列的低成本QuickStart<sup>™</sup>开发系统中。该器件内置一个提供5路输出信号的16位PWM。

为便于通信，该器件内置2个I<sup>2</sup>C通道，可以将这些通道独立配置为主模式或从模式。另外还提供了支持主从两种模式的SPI接口。

ADuC7023采用2.7 V至3.6 V电源供电，额定温度范围为-40°C至+125°C工业温度范围，提供32引脚和40引脚LFCSP封装。

#### Rev. C

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.

Tel: 781.329.4700

[www.analog.com](http://www.analog.com)

Fax: 781.461.3113

©2010-2012 Analog Devices, Inc. All rights reserved.

## 目录

特性.....	1	Flash/EE控制接口 .....	37
应用.....	1	SRAM和Flash/EE执行时间 .....	40
概述.....	1	复位和重映射 .....	40
修订历史.....	3	其他模拟外设 .....	43
功能框图.....	4	DAC .....	43
技术规格.....	5	电源监控器 .....	45
时序规格.....	8	比较器 .....	45
绝对最大额定值.....	13	振荡器和锁相环—电源控制 .....	47
ESD警告.....	13	数字外设.....	50
引脚配置和功能描述.....	14	通用输入/输出 .....	50
典型性能参数.....	17	串行外设接口 .....	53
术语.....	18	I <sup>2</sup> C.....	58
ADC技术规格 .....	18	I <sup>2</sup> C功能的外部引脚配置 .....	58
DAC技术规格.....	18	串行时钟生成 .....	58
ARM7TDMI内核概览.....	19	I <sup>2</sup> C总线地址 .....	58
Thumb模式(T).....	19	I <sup>2</sup> C寄存器 .....	59
长乘(M).....	19	可编程逻辑阵列(PLA).....	66
嵌入式ICE (I) .....	19	脉宽调制器.....	70
异常 .....	19	脉宽调制器概述.....	70
ARM寄存器 .....	19	处理器相关外设.....	75
中断延迟.....	20	中断系统.....	75
存储器结构.....	21	IRQ.....	75
存储器访问 .....	21	快速中断请求(FIQ).....	76
Flash/EE存储器 .....	21	矢量中断控制器(VIC).....	77
SRAM .....	21	定时器 .....	82
存储器映射寄存器 .....	21	硬件设计考虑.....	87
ADC电路概览.....	28	电源 .....	87
传递函数.....	28	接地和电路板布局建议 .....	88
典型操作.....	29	时钟振荡器 .....	88
寄存器接口 .....	29	上电复位操作.....	89
转换器操作 .....	32	典型系统配置.....	90
驱动模拟输入.....	33	开发工具.....	91
校准 .....	33	基于PC工具 .....	91
温度传感器 .....	34	I <sup>2</sup> C在线下载器 .....	91
带隙基准电压源.....	35	外形尺寸.....	92
非易失Flash/EE存储器 .....	36	订购指南.....	93
编程 .....	36		
安全性 .....	37		

**修订历史****2012年5月—修订版B至修订版C**

将表2的SDATA更改为SDA，SCLK更改为SCL；将表3的SDATA更改为SDA，SCLK更改为SCL；将图2的SDATA更改为SDA，SCLK更改为SCL .....	8
更改图7、图8和表9 .....	14
将表17的SCLK更改为SCL .....	25
将表18的SCLK更改为SCL .....	26
更改表24的位6和表25的位4到0描述 .....	30
将REFCON寄存器部分中的参考内容从表22更改为表30 .....	35
表53增加注释1 .....	49
更改表55的注释1 .....	50
将“SPICLK(串行时钟I/O)引脚”部分更改为“SCLK(串行时钟I/O)引脚”部分 .....	53
将“串行外设接口”部分和“SCLK(串行时钟I/O)引脚”部分中的SPICLK更改为SCLK .....	53
更改表79 .....	68
更改“定时器”部分 .....	82
增加“小时、分钟、秒和1/128格式”部分和表101 .....	82
更改“T0LD寄存器”部分和“T1LD寄存器”部分 .....	83
更改“T2LD寄存器”部分 .....	85
更新“外形尺寸” .....	92
更改“订购指南” .....	93

**2010年7月—修订版A至修订版B**

更改表1中的温度传感器参数 .....	6
更改表10和表11 .....	23
更改表12和表13 .....	24

更改表16和表17 .....	25
更改表18 .....	26
更改表21和表22 .....	27
更改表24 .....	29
更改“ADCGN寄存器”和“ADCOF寄存器”部分 .....	32
更改“温度传感器”部分 .....	34
更改表29 .....	35
更改“REMAP寄存器”和“RSTCLR寄存器”部分 .....	41
更改“RSTKEY1寄存器”和“RSTKEY2寄存器”部分 .....	42
更改“振荡器和锁相环-电源控制”部分 .....	48
更改“通用输入/输出”部分 .....	51
更改“串行外设接口”部分 .....	53
更改表75 .....	67
更改表83和脉宽调制器概述部分 .....	70
更改表84 .....	71
更改表85 .....	72
更改“FIQSTAN寄存器”部分 .....	81
更改“T2CLR寄存器”部分 .....	85

**2010年6月—修订版0至修订版A**

更改表1中的温度传感器参数 .....	6
更改表24 .....	29
更改“温度传感器”部分 .....	34
更改“DACBKEY0寄存器”部分和表43 .....	47
更改“订购指南” .....	93

**2010年1月—修订版0：初始版**

# ADuC7023

## 功能框图

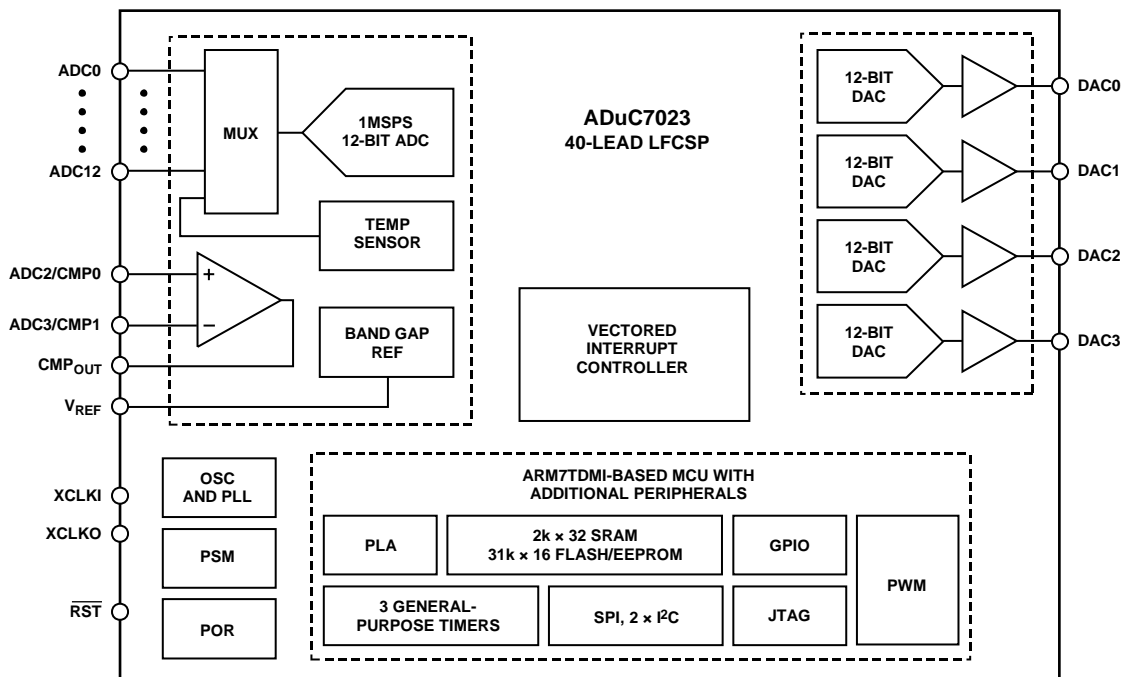


图1.

08875-001

## 技术规格

除非另有说明， $AV_{DD} = IOV_{DD} = 2.7\text{ V}$ 至 $3.6\text{ V}$ ， $V_{REF} = 2.5\text{ V}$ 内部基准电压， $f_{CORE} = 41.78\text{ MHz}$ ， $T_A = -40^\circ\text{C}$ 至 $+125^\circ\text{C}$ 。

表1.

参数	最小值	典型值	最大值	单位	测试条件/注释
ADC通道规格					8采集时钟和 $f_{ADC}/2$
ADC上电时间		5		$\mu\text{s}$	
直流精度 <sup>1,2</sup>	12			Bits	
分辨率				Bits	
积分非线性		$\pm 0.6$	$\pm 1.5$	LSB	2.5 V内部基准电压
		$\pm 1.0$		LSB	1.0 V外部基准电压
微分非线性 <sup>3,4</sup>		$\pm 0.5$	$+1/-0.9$	LSB	2.5 V内部基准电压
		$+0.7/-0.6$		LSB	1.0 V外部基准电压
直流代码分布		1		LSB	ADC输入为直流电压
端点误差 <sup>5</sup>					
失调误差		$\pm 1$	$\pm 2$	LSB	
失调误差匹配		$\pm 1$		LSB	
增益误差		$\pm 2$		LSB	
增益误差匹配		$\pm 1$		LSB	
动态性能					$f_{IN} = 10\text{ kHz}$ 正弦波， $f_{SAMPLE} = 1\text{ MSPS}$ 包括失真和噪声成分
信噪比(SNR)		69		dB	
总谐波失真(THD)		-78		dB	
峰值谐波或杂散噪声		-75		dB	
通道间串扰		-80		dB	相邻通道测量
模拟输入					
输入电压范围					
差模			$V_{CM} \pm V_{REF}/2$ <sup>6</sup>	V	
单端模式			0 to $V_{REF}$	V	
漏电流		$\pm 1$	$\pm 6$	$\mu\text{A}$	
输入电容		20		pF	在ADC采样期间
片内基准电压					在 $V_{REF}$ 和AGND之间连接 $0.47\text{ }\mu\text{F}$ 电容
输出电压		2.5		V	
精度			$\pm 4$	mV	$T_A = 25^\circ\text{C}$
基准源温度系数		$\pm 15$		ppm/ $^\circ\text{C}$	
电源抑制比		75		dB	
输出阻抗		51		$\Omega$	$T_A = 25^\circ\text{C}$
内部 $V_{REF}$ 上电时间		1		ms	
外部基准输入					
输入电压范围	0.625		$AV_{DD}$	V	
DAC通道规格					
直流精度 <sup>7</sup>					$R_L = 5\text{ k}\Omega$ , $C_L = 100\text{ pF}$
分辨率		12		Bits	
相对精度		$\pm 2$		LSB	
差分非线性			$\pm 1$	LSB	保证单调性
失调误差			$\pm 15$	mV	2.5 V内部基准电压
增益误差 <sup>8</sup>			$\pm 1$	%	
增益误差失配		0.1		%	DAC0满量程的百分比
直流精度 <sup>9</sup>					$R_L = 1\text{ k}\Omega$ , $C_L = 100\text{ pF}$
分辨率		12		Bits	
相对精度		$\pm 2.5$		LSB	
差分非线性		$\pm 1$		LSB	保证单调性
失调误差		$\pm 15$		mV	2.5 V内部基准电压
增益误差 <sup>10</sup>		$\pm 1$		%	
增益误差失配		0.1		%	DAC0满量程的百分比
模拟输出					
输出电压范围1		0 to 2.5		V	$V_{REF}$ 范围：AGND至 $AV_{DD}$
输出电压范围2		0 to $AV_{DD}$		V	
输出阻抗		2		$\Omega$	

# ADuC7023

参数	最小值	典型值	最大值	单位	测试条件/注释
运算放大器模式下DAC					
运算放大器模式下DAC输出缓冲器					
输入失调电压		±0.25		mV	
输入失调电压漂移		8		μV/°C	
输入失调电流		0.3		nA	
输入偏置电流		0.4		nA	
增益		80		dB	5 kΩ负载
单位增益频率		5		MHz	R <sub>L</sub> = 5 kΩ, C <sub>L</sub> = 100 pF
CMRR		80		dB	
建立时间		10		μs	R <sub>L</sub> = 5 kΩ, C <sub>L</sub> = 100 pF
输出压摆率		1.5		V/μs	R <sub>L</sub> = 5 kΩ, C <sub>L</sub> = 100 pF
电源抑制比(PSRR)		75		dB	
DAC交流特性					
输出电压建立时间		10		μs	
数模转换脉冲干扰		±20		nV-sec	主进位1 LSB变化(DACxDAT寄存器中同时变化的最大位数)
比较器					
输入失调电压		±10		mV	
输入偏置电流		1		μA	
输入电压范围	AGND		AV <sub>DD</sub> - 1.2	V	
输入电容		7		pF	
迟滞 <sup>4,6</sup>	2		15	mV	迟滞可以通过CMPCON寄存器中的CMPHYST位打开或关闭
响应时间		3		μs	100 mV过驱、CMPRES = 11
温度传感器					指示芯片温度
25°C时电压输出		1.369		V	
电压TC		4.42		mV/°C	
未校准时的精度		±3		°C	
利用TEMPREF寄存器内容执行单点校准后的精度		±1.5		°C	
θ <sub>JA</sub> 热阻					
40引脚LFCSP		26		°C/W	
32引脚LFCSP		32.5		°C/W	
电源监控器(PSM)					
IOV <sub>DD</sub> 跳变点选择		2.79		V	一个跳变点
电源跳变点精度		±2		%	已选跳变点标称电压
上电复位		2.41		V	
看门狗定时器(WDT)					
超时时间	0		512	秒	
FLASH/EE存储器					
耐久性 <sup>11</sup>	10,000			周期	
数据保持 <sup>12</sup>	20			年	T <sub>J</sub> = 85°C
数字输入					除XCLKI和XCLKO外的所有数字输入
逻辑1输入电流		±0.2	±1	μA	V <sub>IH</sub> = V <sub>DD</sub> 或V <sub>IH</sub> = 5 V
逻辑0输入电流		-40	-60	μA	V <sub>IL</sub> = 0 V; TDI除外
		-80	-120	μA	V <sub>IL</sub> = 0 V; TDI
输入电容		10		pF	
逻辑输入 <sup>4</sup>					除XCLKI外的所有数字输入
输入低电压V <sub>INL</sub>			0.8	V	
输入高电压V <sub>INH</sub>	2.0			V	
逻辑输出					除XCLKO外的所有数字输出
输出高电压V <sub>OH</sub>	2.4			V	I <sub>SOURCE</sub> = 1.6 mA
V <sub>OL</sub> 输出低电压 <sup>13</sup>			0.4	V	I <sub>SINK</sub> = 1.6 mA
晶体输入XCLKI和XCLKO					
逻辑输入, 仅限XCLKI					
输入低电压V <sub>INL</sub>		1.1		V	
输入高电压V <sub>INH</sub>		1.7		V	
XCLKI输入电容		20		pF	
XCLKO输出电容		20		pF	

参数	最小值	典型值	最大值	单位	测试条件/注释
内部振荡器		32.768		kHz	
			±3	%	
MCU时钟速率		326		kHz	CD = 7
采用32 kHz内部振荡器		41.78		MHz	CD = 0
采用32 kHz外部晶体	0.05		44	MHz	T <sub>A</sub> = 85°C
使用外部时钟	0.05		41.78	MHz	T <sub>A</sub> = 125°C
启动时间					内核时钟= 41.78 MHz
上电时		66		ms	
从暂停/休眠模式		24		ns	CD = 0
		3.07		µs	CD = 7
从休眠模式		1.58		ms	
从停止模式		1.7		ms	
可编程逻辑阵列(PLA)					
引脚传输延迟		12		ns	从输入引脚到输出引脚
单元传输延迟		2.5		ns	
电源要求 <sup>14, 15</sup>					
电源电压范围					
AV <sub>DD</sub> 至AGND和IOV <sub>DD</sub> 至DGND	2.7		3.6	V	
模拟电源电流		200		µA	ADC处于空闲模式
AV <sub>DD</sub> 电流					
数字电源电流					从Flash/EE执行代码
IOV <sub>DD</sub> 正常模式下电流		8.5	10	mA	CD = 7
		11	15	mA	CD = 3
		28	35	mA	CD = 0(时钟频率41.78 MHz)
IOV <sub>DD</sub> 暂停模式下电流		14	20	mA	CD = 0(时钟频率41.78 MHz)
IOV <sub>DD</sub> 休眠模式下电流		230	650	µA	T <sub>A</sub> = 125°C
附加电源电流					
ADC		1.4		mA	1 MSPS时
		0.7		mA	62.5 kSPS时
DAC		400		µA	每DAC
ESD测试					2.5 V基准电压, T <sub>A</sub> = 25°C
HBM通过			3	kV	
FICDM通过			1.0	kV	

<sup>1</sup> 在微控制器内核正常工作时，保证所有ADC通道的技术规格。

<sup>2</sup> 适用于所有ADC输入通道。

<sup>3</sup> 使用ADC失调寄存器(ADCOF)和增益系数寄存器(ADCGN)中的出厂设定默认值进行测试。

<sup>4</sup> 未经生产测试，但量产时的设计和/或特性数据可提供保证。

<sup>5</sup> 采用运算放大器AD845作为一个外部输入缓冲级，用ADCOF和ADCGN寄存器中的出厂设定默认值进行测试(如图28所示)。

当使用外部ADC系统元件时，用户需要进行系统校准来消除外部端点误差并满足规格要求(详见校准部分)。

<sup>6</sup> 输入信号可以任何直流共模电压(V<sub>CM</sub>)为中心，但该值必须位于ADC规定输入电压范围内。

<sup>7</sup> DAC的线性度是使用一个递减的数据范围(100到3995)计算出来的。

<sup>8</sup> DAC增益误差是使用一个递减的数据范围(100到内部2.5V基准电压)计算出来的。

<sup>9</sup> DAC的线性度是使用一个递减的数据范围(100到3995)计算出来的。

<sup>10</sup> DAC增益误差是使用一个递减的数据范围(100到内部2.5V基准电压)计算出来的。

<sup>11</sup> 耐久性是在分别在-40°C、+25°C、+85°C及+125°C时依据JEDEC 22标准方法A117来认定的。

<sup>12</sup> 根据JEDEC 22标准方法A117，保持期限相当于85°C结温时的寿命。保持期限会随着结温递减。

<sup>13</sup> 测试是在最多8个I/O端口输出低电平时进行的。

<sup>14</sup> 电源功耗分别在正常、暂停和休眠模式下测试的，这3种模式下的测试条件分别为：正常模式供电电压为3.6 V、暂停模式供电电压为3.6 V、休眠模式供电电压为3.6 V。

<sup>15</sup> 在一个Flash/EE擦写周期中，IOV<sub>DD</sub>电源电流通常降低2 mA。

# ADuC7023

## 时序规格

表2. 快速模式下I<sup>2</sup>C时序(400 kHz)

参数	描述	从机		主机	单位
		最小值	最大值	典型值	
$t_L$	SCL低电平脉宽	200		1360	ns
$t_H$	SCL高电平脉宽	100		1140	ns
$t_{SHD}$	起始条件保持时间	300			ns
$t_{DSU}$	数据建立时间	100		740	ns
$t_{DHD}$	数据保持时间	0		400	ns
$t_{RSU}$	重复起始建立时间	100			ns
$t_{PSU}$	停止条件的建立时间	100		800	ns
$t_{BUF}$	一个结束条件和起始条件之间的总线空闲时间	1.3			$\mu$ s
$t_R$	SCL和SDA的上升时间		300	200	ns
$t_F$	SCL和SDA的下降时间		300		ns

表3. 标准模式下I<sup>2</sup>C时序(100 kHz)

参数	描述	从机		单位
		最小值	最大值	
$t_L$	SCL低电平脉宽	4.7		$\mu$ s
$t_H$	SCL高电平脉宽	4.0		ns
$t_{SHD}$	起始条件保持时间	4.0		$\mu$ s
$t_{DSU}$	数据建立时间	250		ns
$t_{DHD}$	数据保持时间	0	3.45	$\mu$ s
$t_{RSU}$	重复起始建立时间	4.7		$\mu$ s
$t_{PSU}$	停止条件的建立时间	4.0		$\mu$ s
$t_{BUF}$	一个结束条件和起始条件之间的总线空闲时间	4.7		$\mu$ s
$t_R$	SCL和SDA的上升时间		1	$\mu$ s
$t_F$	SCL和SDA的下降时间		300	ns

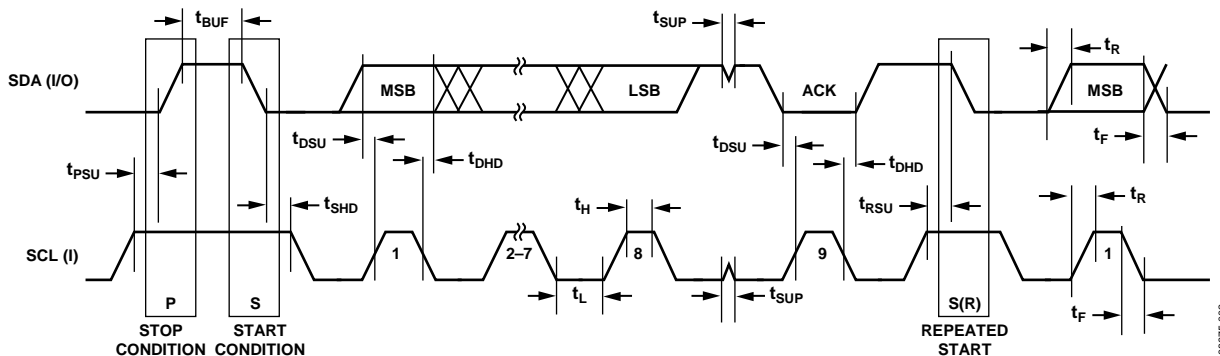


图2. I<sup>2</sup>C兼容接口时序



**表4. SPI主机定时(相位模式 = 1)**

参数	描述	最小值	典型值	最大值	单位
$t_{SL}$	SCLK低电平脉宽 <sup>1</sup>		$(SPIDIV + 1) \times t_{UCLK}$		ns
$t_{SH}$	SCLK高电平脉宽 <sup>1</sup>		$(SPIDIV + 1) \times t_{UCLK}$		ns
$t_{DAV}$	SCLK边沿之后数据输出有效时间			25	ns
$t_{DSU}$	SCLK边沿之前数据输入建立时间 <sup>1</sup>	$1 \times t_{UCLK}$			ns
$t_{DHD}$	SCLK边沿之后数据输入保持时间 <sup>1</sup>	$2 \times t_{UCLK}$			ns
$t_{DF}$	数据输出下降时间		5	12.5	ns
$t_{DR}$	数据输出上升时间		5	12.5	ns
$t_{SR}$	SCLK上升时间		5	12.5	ns
$t_{SF}$	SCLK下降时间		5	12.5	ns

<sup>1</sup> $t_{UCLK} = 23.9$  ns。其对应于在时钟分频器之前来自PLL的41.78 MHz内部时钟。

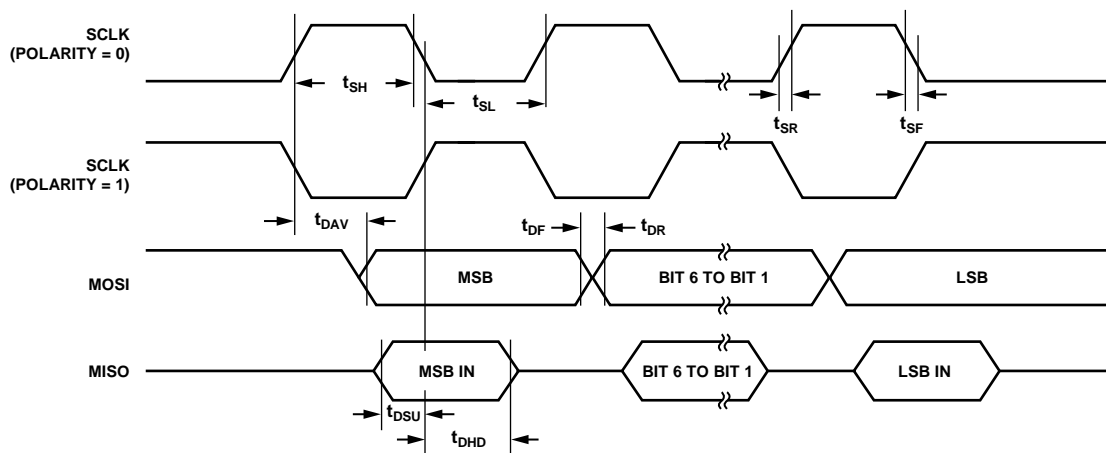


图3. SPI主机定时(相位模式 = 1)

08675-003

# ADuC7023

表5. SPI主机定时(相位模式 = 0)

参数	描述	最小值	典型值	最大值	单位
$t_{SL}$	SCLK低电平脉宽 <sup>1</sup>		$(SPIDIV + 1) \times t_{UCLK}$		ns
$t_{SH}$	SCLK高电平脉宽 <sup>1</sup>		$(SPIDIV + 1) \times t_{UCLK}$		ns
$t_{DAV}$	SCLK边沿之后数据输出有效时间			25	ns
$t_{DOSU}$	SCLK边沿之前数据输出建立时间			75	ns
$t_{DSU}$	SCLK边沿之前数据输入建立时间 <sup>1</sup>	$1 \times t_{UCLK}$			ns
$t_{DHD}$	SCLK边沿之后数据输入保持时间 <sup>1</sup>	$2 \times t_{UCLK}$			ns
$t_{DF}$	数据输出下降时间		5	12.5	ns
$t_{DR}$	数据输出上升时间		5	12.5	ns
$t_{SR}$	SCLK上升时间		5	12.5	ns
$t_{SF}$	SCLK下降时间		5	12.5	ns

<sup>1</sup> $t_{UCLK} = 23.9$  ns。其对应于在时钟分频器之前来自PLL的41.78 MHz内部时钟。

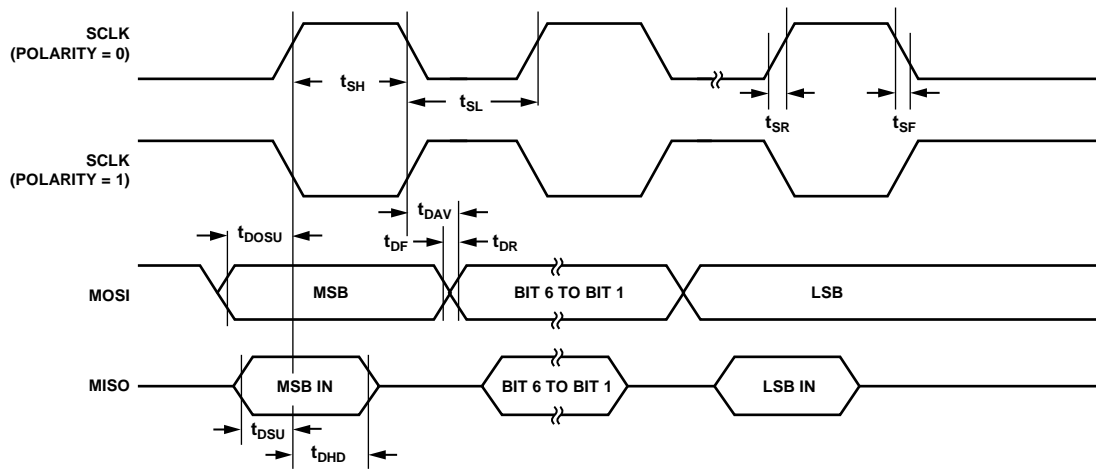


图4. SPI主机定时(相位模式 = 0)

08675-004

表6. SPI从机定时(相位模式 = 1)

参数	描述	最小值	典型值	最大值	单位
$t_{SS}$	$\overline{SS}$ 至SCLK边沿	200			ns
$t_{SL}$	SCLK低电平脉宽 <sup>1</sup>		$(SPIDIV + 1) \times t_{UCLK}$		ns
$t_{SH}$	SCLK高电平脉宽 <sup>1</sup>		$(SPIDIV + 1) \times t_{UCLK}$		ns
$t_{DAV}$	SCLK边沿之后数据输出有效时间			25	ns
$t_{DSU}$	SCLK边沿之前数据输入建立时间 <sup>1</sup>	$1 \times t_{UCLK}$			ns
$t_{DHD}$	SCLK边沿之后数据输入保持时间 <sup>1</sup>	$2 \times t_{UCLK}$			ns
$t_{DF}$	数据输出下降时间		5	12.5	ns
$t_{DR}$	数据输出上升时间		5	12.5	ns
$t_{SR}$	SCLK上升时间		5	12.5	ns
$t_{SF}$	SCLK下降时间		5	12.5	ns
$t_{SFS}$	$\overline{SS}$ 在SCLK沿后变高	0			ns

<sup>1</sup> $t_{UCLK}$  = 23.9 ns。其对应于在时钟分频器之前来自PLL的41.78 MHz内部时钟。

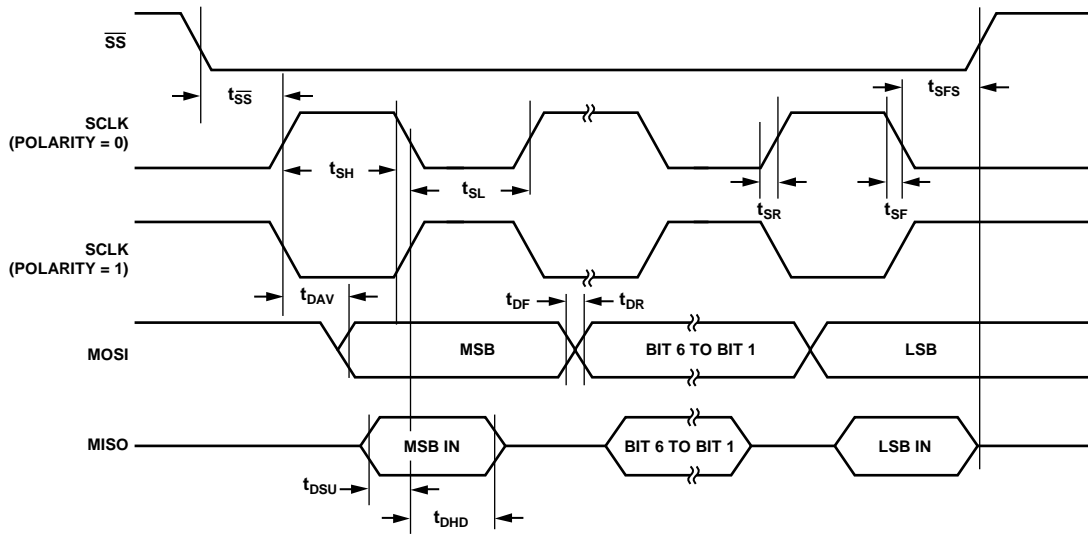


图5. SPI从机定时(相位模式 = 1)

08675-005

表7. SPI从机定时(相位模式 = 0)

参数	描述	最小值	典型值	最大值	单位
$t_{SS}$	$\overline{SS}$ 至SCLK边沿	200			ns
$t_{SL}$	SCLK低电平脉宽 <sup>1</sup>		$(SPIDIV + 1) \times t_{UCLK}$		ns
$t_{SH}$	SCLK高电平脉宽 <sup>1</sup>		$(SPIDIV + 1) \times t_{UCLK}$		ns
$t_{DAV}$	SCLK边沿之后数据输出有效时间			25	ns
$t_{DSU}$	SCLK边沿之前数据输入建立时间 <sup>1</sup>	$1 \times t_{UCLK}$			ns
$t_{DHD}$	SCLK边沿之后数据输入保持时间 <sup>1</sup>	$2 \times t_{UCLK}$			ns
$t_{DF}$	数据输出下降时间		5	12.5	ns
$t_{DR}$	数据输出上升时间		5	12.5	ns
$t_{SR}$	SCLK上升时间		5	12.5	ns
$t_{SF}$	SCLK下降时间		5	12.5	ns
$t_{DOCS}$	$\overline{SS}$ 边沿之后数据输出有效			25	ns
$t_{SFS}$	$\overline{SS}$ 在SCLK沿后变高	0			ns

<sup>1</sup> $t_{UCLK} = 23.9$  ns。其对应于在时钟分频器之前来自PLL的41.78 MHz内部时钟。

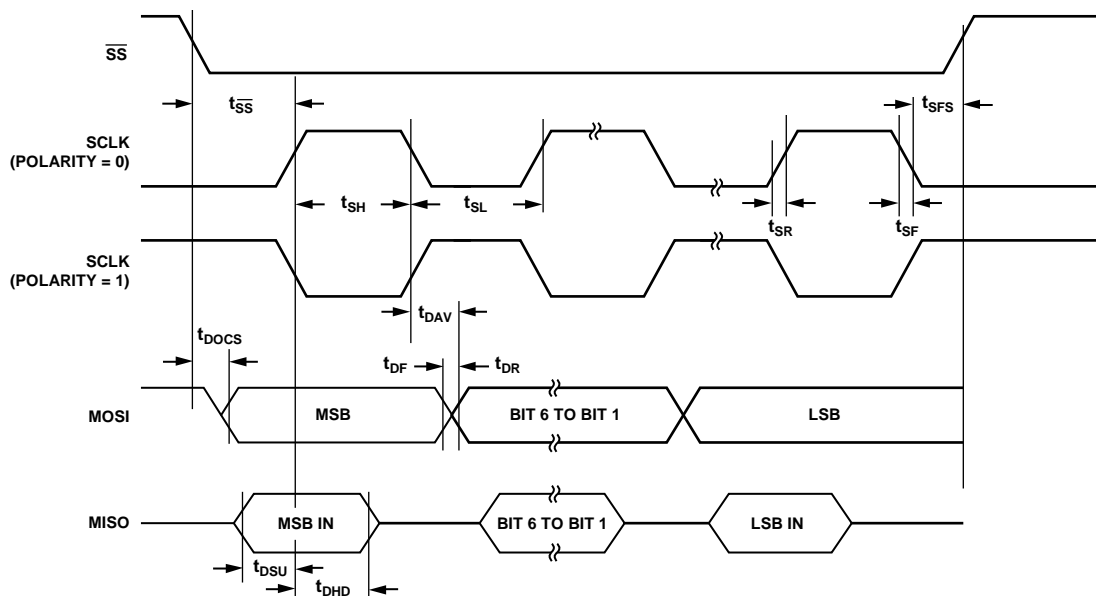


图6. SPI从机定时(相位模式 = 0)

08875-006

## 绝对最大额定值

除非另有说明，AGND = GND<sub>REF</sub>，T<sub>A</sub> = 25°C。

表8.

参数	额定值
AV <sub>DD</sub> 至IOV <sub>DD</sub>	-0.3 V至+0.3 V
AGND至DGND	-0.3 V至+0.3 V
IOV <sub>DD</sub> 至DGND，AV <sub>DD</sub> 至AGND	-0.3 V至+6 V
数字输入电压至DGND	-0.3 V至+5.3 V
数字输出电压至DGND	-0.3 V至IOV <sub>DD</sub> + 0.3 V
V <sub>REF</sub> 至AGND	-0.3 V至AV <sub>DD</sub> + 0.3 V
模拟输入至AGND	-0.3 V至AV <sub>DD</sub> + 0.3 V
模拟输出至AGND	-0.3 V至AV <sub>DD</sub> + 0.3 V
工业温度范围	-40°C至+125°C
存储温度范围	-65°C至+150°C
结温	150°C
θ <sub>JA</sub> 热阻	
40引脚LFCSP	26°C/W
32引脚LFCSP	32.5°C/W
回流焊峰值温度	
锡铅体系(10秒至30秒)	240°C
RoHS体系(20秒至40秒)	260°C

注意，超出上述绝对最大额定值可能会导致器件永久性损坏。这只是额定最值，并不能以这些条件或者在任何其它超出本技术规范操作章节中所示规格的条件下，推断器件能否正常工作。长期在绝对最大额定值条件下工作会影响器件的可靠性。

任何时候只能使用一个绝对最大额定值。

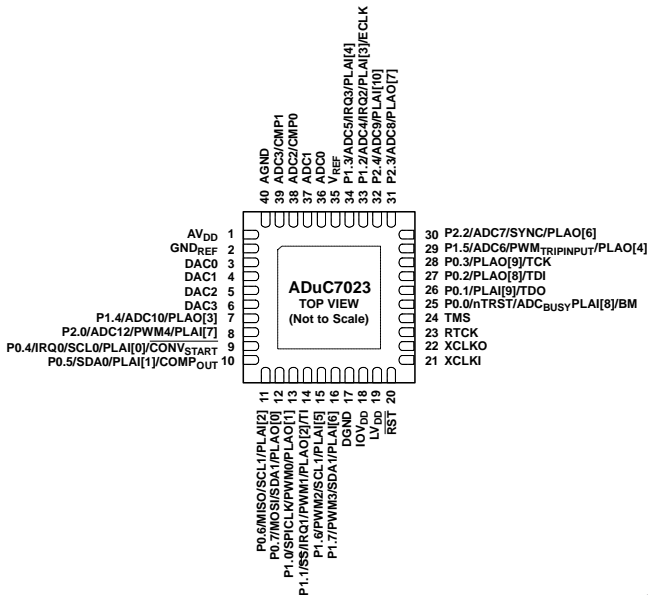
### ESD警告



#### ESD(静电放电)敏感器件。

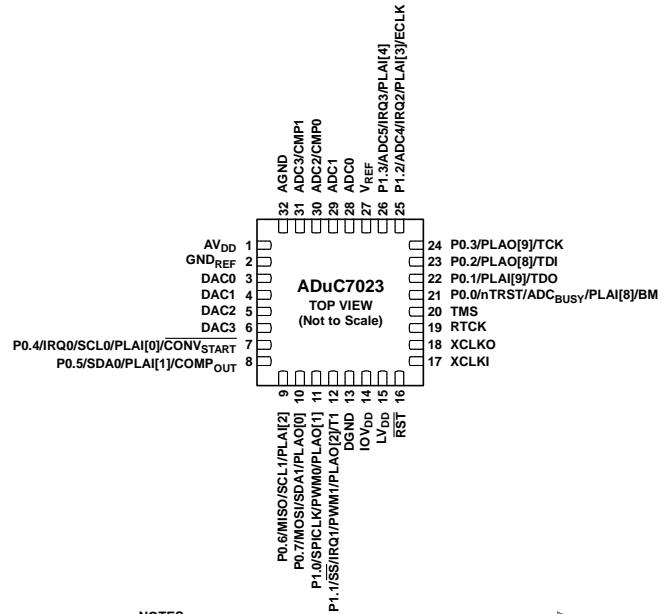
带电器件和电路板可能会在没有察觉的情况下放电。尽管本产品具有专利或专有保护电路，但在遇到高能量ESD时，器件可能会损坏。因此，应当采取适当的ESD防范措施，以避免器件性能下降或功能丧失。

## 引脚配置和功能描述



NOTES  
1. EXPOSED PAD. THE PADDLE NEEDS TO BE SOLDERED AND EITHER CONNECTED TO AGND OR LEFT FLOATING.

图7. 40引脚配置



NOTES  
1. EXPOSED PAD. THE PADDLE NEEDS TO BE SOLDERED AND EITHER CONNECTED TO AGND OR LEFT FLOATING.

图8. 32引脚配置

表9. 引脚功能描述

引脚编号		引脚名称	描述
40-LFCSP	32-LFCSP		
0	0	裸露焊盘	裸露焊盘。焊盘需要焊接，并连接到AGND或悬空。
36	28	ADC0	单端或差分模拟输入0。
37	29	ADC1	单端或差分模拟输入1。
38	30	ADC2/CMP0	单端或差分模拟输入2/比较器正相输入。
39	31	ADC3/CMP1	单端或差分模拟输入3/比较器反相输入。
32	不适用	P2.4/ADC9/PLAI[10]	通用输入和输出端口2.4/ADC单端或差分模拟输入/可编程逻辑阵列输入单元10。 默认情况下，此引脚配置为已使能弱上拉电阻的数字输入端。
31	不适用	P2.3/ADC8/PLAO[7]	通用输入和输出端口2.3/ADC单端或差分模拟输入8/可编程逻辑阵列输出单元7。 默认情况下，此引脚配置为已使能弱上拉电阻的数字输入端。用作ADC输入端时，应手动禁用上拉电阻。
30	不适用	P2.2/ADC7/SYNC/PLAO[6]	通用输入和输出端口2.2/ADC单端或差分模拟输入7/PWM同步/可编程逻辑阵列输出单元6。默认情况下，此引脚配置为已使能弱上拉电阻的数字输入端。用作ADC输入端时，应手动禁用上拉电阻。
8	不适用	P2.0/ADC12/PWM4/PLAI[7]	通用输入和输出端口2.0/ADC单端或差分模拟输入12/PWM输出4/可编程逻辑阵列输入单元7。默认情况下，此引脚配置为已使能弱上拉电阻的数字输入端。用作ADC输入端时，无法禁用内部上拉电阻。这表示此引脚的漏电流值高于其他模拟输入引脚。
2	2	GNDREF	ADC地基准电压。为了优化性能，模拟电源应同DGND分离。
3	3	DAC0	DAC0电压输出或ADC输入。
4	4	DAC1	DAC1电压输出或ADC输入。

引脚编号		引脚名称	描述
40-LFCSP	32-LFCSP		
5	5	DAC2	DAC2电压输出。
6	6	DAC3	DAC3电压输出。
24	20	TMS	测试模式选择, JTAG测试端口输入。调试和下载访问。 此引脚具有一个连接至IOV <sub>DD</sub> 的内部上拉电阻。在有些情况下, 还需要一个外部上拉电阻, 以确保器件不会进入错误状态。
25	21	P0.0/nTRST/ADC <sub>BUSY</sub> /PLAI[8]/BM	这是一个多功能引脚, 具体如下: 通用输入和输出端口0.0。默认情况下, 此引脚配置为GPIO。 JTAG复位输入。调试和下载访问。如果此引脚保持为低电平, 则无法访问JTAG, 因为JTAG接口处于复位状态并且P0.1/P0.2/P0.3均配置为GPIO引脚。 ADC繁忙信号。 可编程逻辑阵列输入单元8。 引导模式(BM)入口引脚。如果复位时BM处于低电平, 且闪存地址0x80014 = 0xFFFFFFFF, ADuC7023将进入I <sup>2</sup> C下载模式。 如果复位时BM被拉高, 或者如果复位时BM处于低电平, 但闪存地址0x80014不等于0xFFFFFFFF, 则ADuC7023将执行代码。
26	22	P0.1/PLAI[9]/TDO	此引脚的默认值取决于P0.0/BM的电平。如果P0.0/BM = 0, 此引脚默认为通用输入。如果P0.0/BM = 1, 此引脚默认为JTAG测试数据输出引脚。这是一个多功能引脚, 具体如下: 通用输入和输出端口0.1。 可编程逻辑阵列输入单元9。 测试数据输出, JTAG测试端口输出。调试和下载访问。通过JTAG调试器件时, 不得通过用户代码切换此引脚, 并且不能更改影响此引脚的GPOCON/GPODAT寄存器位。
27	23	P0.2/PLAO[8]/TDI	此引脚的默认值取决于P0.0/BM的电平。如果P0.0/BM = 0, 此引脚默认为通用输入。如果P0.0/BM = 1, 此引脚默认为JTAG测试数据输入引脚。这是一个多功能引脚, 具体如下: 通用输入和输出端口0.2。 可编程逻辑阵列输出单元8。 测试数据输入, JTAG测试端口输入。调试和下载访问。通过JTAG调试器件时, 不得通过用户代码切换此引脚, 并且不得更改影响此引脚的GPOCON/GPODAT寄存器位。
28	24	P0.3/PLAO[9]/TCK	此引脚的默认值取决于P0.0/BM的电平。如果P0.0/BM = 0, 此引脚默认为通用输入。如果P0.0/BM = 1, 此引脚默认为JTAG测试数据时钟引脚。这是一个多功能引脚, 具体如下: 通用输入和输出端口0.3。 可编程逻辑阵列输出单元9。 测试时钟, JTAG测试端口时钟输入。调试和下载访问。通过JTAG调试器件时, 不得通过用户代码切换此引脚, 并且不得更改影响此引脚的GPOCON/GPODAT寄存器位。
17	13	DGND	数字地。
18	14	IOV <sub>DD</sub>	3.3 V电源, 用于GPIO和片内稳压器输入。
19	15	LV <sub>DD</sub>	片内稳压器2.6 V输出。该输出只能通过一个0.47 μF电容器连接至DGND。
20	16	$\overline{\text{RST}}$	复位输入, 低电平有效。
23	19	RTCK	返回JTAG时钟信号。这并不是标准JTAG时钟信号, 而是来自JTAG控制器的输出信号。如果使用的是20引脚JTAG接头, 则连接到引脚11。

# ADuC7023

引脚编号		引脚名称	描述
40-LFCSP	32-LFCSP		
9	7	P0.4/IRQ0/SCL0/PLAI[0]/CONV	通用输入和输出端口0.4/外部中断请求0/I <sup>2</sup> C0时钟信号/可编程逻辑阵列输入单元0/ADC外部转换启动。默认情况下,此引脚配置为已使能弱上拉电阻的数字输入端。
10	8	P0.5/SDA0/PLAI[1]/COMP <sub>OUT</sub>	通用输入和输出端口0.5/I <sup>2</sup> C0数据信号/可编程逻辑阵列输入单元1/电压比较器输出。默认情况下,此引脚配置为已使能弱上拉电阻的数字输入端。
11	9	P0.6/MISO/SCL1/PLAI[2]	通用输入和输出端口0.6/SPI MISO信号/32引脚封装上的I <sup>2</sup> C1时钟/可编程逻辑阵列输入单元2。默认情况下,此引脚配置为已使能弱上拉电阻的数字输入端。
12	10	P0.7/MOSI/SDA1/PLAO[0]	通用输入和输出端口0.7/SPI MOSI信号/32引脚封装上的I <sup>2</sup> C1数据信号/可编程逻辑阵列输出单元0。默认情况下,此引脚配置为已使能弱上拉电阻的数字输入端。
21	17	XCLKI	晶体振荡反相器输入和内部时钟发生器电路输入。如果未使用,应连接到DGND。
22	18	XCLKO	晶体振荡反相器输出。如果未使用,应保持悬空。
16	不适用	P1.7/PWM3/SDA1/PLAI[6]	通用输入和输出端口1.7/PWM输出3/I <sup>2</sup> C数据信号/可编程逻辑阵列输入单元6。默认情况下,此引脚配置为已使能弱上拉电阻的数字输入端。
15	不适用	P1.6/PWM2/SCL1/PLAI[5]	通用输入和输出端口1.6/PWM输出2/I <sup>2</sup> C时钟信号/可编程逻辑阵列输入单元5。默认情况下,此引脚配置为已使能弱上拉电阻的数字输入端。
29	不适用	P1.5/ADC6/PWM <sub>TRIPINPUT</sub> /PLAO[4]	通用输入和输出端口1.5/ADC单端或差分模拟输入6/PWM <sub>TRIPINPUT</sub> /可编程逻辑阵列输出单元4。默认情况下,此引脚配置为已使能弱上拉电阻的数字输入端。用作ADC输入端时,应手动禁用上拉电阻。
7	不适用	P1.4/ADC10/PLAO[3]	通用输入和输出端口1.4/ADC单端或差分模拟输入10/可编程逻辑阵列输出单元3。默认情况下,此引脚配置为已使能弱上拉电阻的数字输入端。用作ADC输入端时,应手动禁用上拉电阻。
34	26	P1.3/ADC5/IRQ3/PLAI[4]	通用输入和输出端口1.3/ADC单端或差分模拟输入5/外部中断请求3/可编程逻辑阵列输入单元4。默认情况下,此引脚配置为已使能弱上拉电阻的数字输入端。用作ADC输入端时,应手动禁用上拉电阻。
33	25	P1.2/ADC4/IRQ2/PLAI[3]/ECLK/	通用输入和输出端口1.2/ADC单端或差分模拟输入4/外部中断请求2/可编程逻辑阵列输入单元3/外部时钟的输入和输出。默认情况下,此引脚配置为已使能弱上拉电阻的数字输入端。用作ADC输入端时,应手动禁用上拉电阻。
14	12	P1.1/ $\overline{SS}$ /IRQ1/PWM1/PLAO[2]/T1	通用输入和输出端口1.1/SPI接口从机选择(低电平有效)/外部中断请求1/PWM输出1/可编程逻辑阵列输出单元2/定时器1输入时钟。默认情况下,此引脚配置为已使能弱上拉电阻的数字输入端。
13	11	P1.0/SCLK/PWM0/PLAO[1]	通用输入和输出端口1.0/SPI接口时钟信号/PWM输出0/可编程逻辑阵列输出单元1。默认情况下,此引脚配置为已使能弱上拉电阻的数字输入端。
35	27	V <sub>REF</sub>	2.5 V内部基准电压。在使用内部基准电压源时必须连接至一个0.47 μF电容。
40	32	AGND	模拟地。模拟电路的地基准点。
1	1	AV <sub>DD</sub>	3.3 V模拟电源。



### 典型性能参数

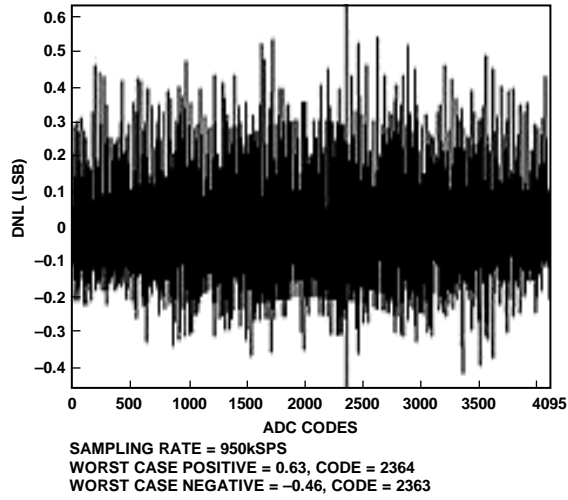


图9.  $f_{ADC} = 950$  kSPS且使用内部基准电压源时的典型微分非线性(DNL)曲线

08675-049

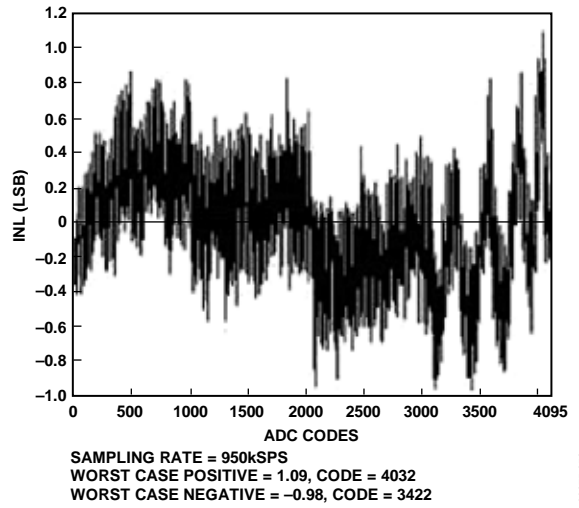


图12.  $f_{ADC} = 950$  kSPS且使用外部1.0 V基准电压源时的典型积分非线性(INL)曲线

08675-052

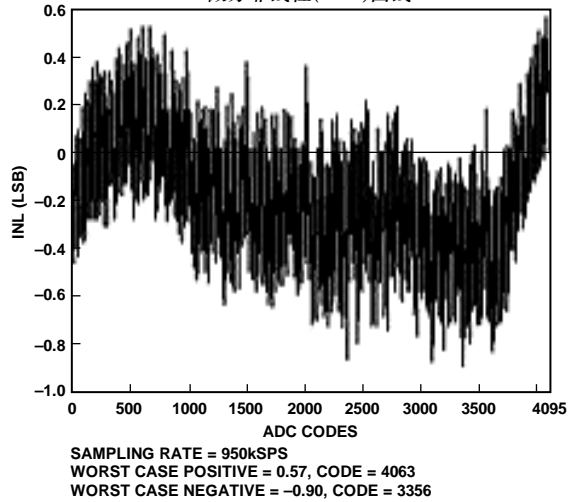


图10.  $f_{ADC} = 950$  kSPS且使用内部基准电压源时的典型积分非线性(INL)曲线

08675-050

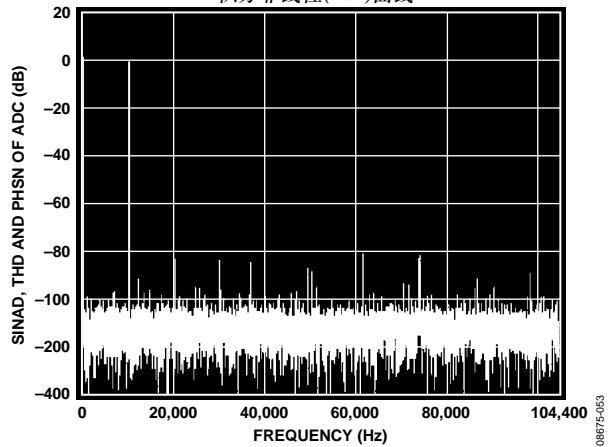


图13. 使用内部2.5 V基准电压源时ADC的SINAD、THD和PHSN

08675-053

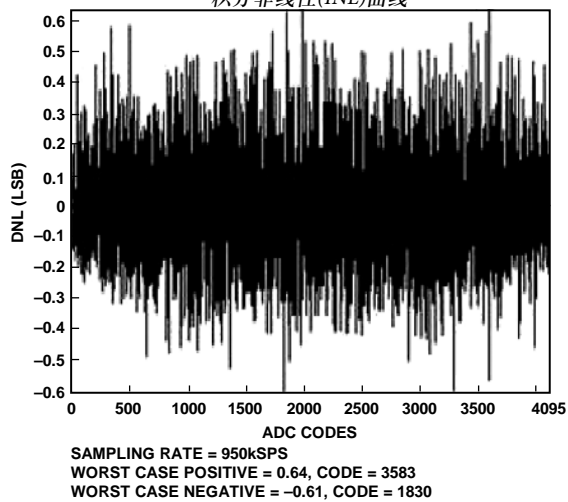


图11.  $f_{ADC} = 950$  kSPS且使用外部1.0 V基准电压源时的典型微分非线性(DNL)曲线

08675-051

## 术语

### ADC技术规格

#### 积分非线性(INL)

ADC输出与通过ADC端点的传递函数直线之间的最大偏差。传递函数端点是指，在零电平位置比第一个编码的跃变点低 $\frac{1}{2}$  LSB的点，以及在满量程位置比最后一个编码的跃变点高 $\frac{1}{2}$  LSB的点。

#### 微分非线性(DNL)

ADC中任意两个相邻码之间所测得变化值与理想的1 LSB变化值之间的差异。

#### 失调误差

第一个转换编码(从0000000到0000001)的跃变点与理想点 $\frac{1}{2}$  LSB之间的偏差。

#### 增益误差

在失调误差调零之后，最后一个转换编码的跃变点与理想AIN电压(满量程 - 1.5 LSB)的偏差。

#### 信号与(噪声+失真)比

在ADC输出端所测量到的信号与(噪声+失真)的比值。这里的信号是基波幅值的均方根值。噪声为除了直流信号以外一直到半采样频率( $f_s/2$ )的所有非基波信号均方根和。

在数字化过程中，这个比值的大小取决于量化级数，量化级数越多，量化噪声就越小。

对于一个正弦波输入的理想N位转换器，信号与噪声+失真比值的理论计算值为：

$$\text{信号与(噪声+失真)比值} = (6.02 N + 1.76) \text{ dB}$$

因此，对于12位转换器，该值为74dB。

#### 总谐波失真

所有谐波均方根和与基波均方根之比。

### DAC技术规格

#### 相对精度

也被称作端点线性度，相对精度是指DAC输出与通过DAC端点的传递函数直线之间的最大偏差。在零点误差和满量程误差调零后才可进行相对精度测量。

#### 输出电压建立时间

是指对于一个满量程输入变化，DAC输出稳定在1 LSB变化范围内所需时间。

## ARM7TDMI内核概览

ARM7<sup>®</sup>内核为32位精简指令集计算机(RISC)。指令和数据使用单32位总线。数据的长度可以是8位、16位或32位。指令字的长度为32位。

ARM7TDMI是ARM7内核，还有4个额外的特性：支持16位thumb指令集(T)、支持调试(D)、支持长乘(M)并包含一个支持嵌入式系统调试的EmbeddedICE模块(I)

### THUMB模式(T)

一条ARM指令的长度为32位。ARM7TDMI处理器支持压缩至16位的第二指令集，即Thumb<sup>®</sup>指令集。用Thumb指令集替代ARM指令集，可以更为快速地从16位存储器执行代码并且实现更高的代码密度。这就使得ARM7TDMI内核尤其适用于嵌入式系统。

然而，Thumb模式有两个缺点：对于同一工作，Thumb代码通常需要更多指令。因此，如果更强调时效性，ARM代码更适合用来优化代码性能。此外，Thumb指令集并不包含异常处理的所有指令，所以如果异常发生在Thumb状态，处理器会自动切换到ARM代码。

关于内核架构、编程模块、ARM和ARM Thumb指令集的具体内容，请参阅ARM7TDMI用户手册。

### 长乘(M)

ARM7TDMI指令集包括四个额外的指令，分别为得到64位结果的32位与32位相乘指令；得到64位结果的32位与32位乘加(MAC)指令。得到这些结果比标准的ARM7内核所需的时钟周期更少。

### EmbeddedICE (I)

EmbeddedICE支持内核片内调试。EmbeddedICE模块包含断点和观察点寄存器，在调试时这些寄存器可使代码中止执行。这些寄存器可以通过JTAG测试端口来控制。

当遇到一个断点或观察点时，处理器中断，并进入调试状态。一旦进入调试状态，就可以检查处理器寄存器、Flash/EE，SRAM和存储器映射寄存器的状态。

## 异常

ARM支持5种类型的异常，并且每一种异常模式有一种优先处理器模式。这5种异常为：

- 正常中断或IRQ。这种异常用于内部和外部事件的通用中断处理。
- 快速中断或FIQ。这是用于数据传输或低延迟时间通道处理。FIQ的优先级高于IRQ。
- 存储器中止。
- 尝试执行未定义指令。
- 软件中断指令(SWI)。它通常用于通知操作系统。

典型情况下，程序员定义中断为IRQ，但是为了得到更高优先级的中断，即得到更快响应时间，程序员可以定义中断为FIQ。

## ARM寄存器

ARM7TDMI共有37个寄存器：31个通用寄存器和6个状态寄存器。每一个工作模式都有专门的寄存器组。

当编写用户级程序时，15个通用32位寄存器(R0-R14)、程序计数器(R15)和当前程序状态寄存器(CPSR)是可用的。余下的寄存器只用于系统级编程和异常处理。

异常发生后，异常模式专用的寄存器将取代某些标准寄存器。所有的异常模式都有各自的替换寄存器组，用于堆栈指针(R13)和链接寄存器(R14)，如图14所示。快速中断模式有更多的寄存器(R8到R12)用于快速中断处理。这意味着无需先保存或者重新保存这些寄存器，就可以进行中断处理，因此在中断处理中可以节省至关重要的时间。

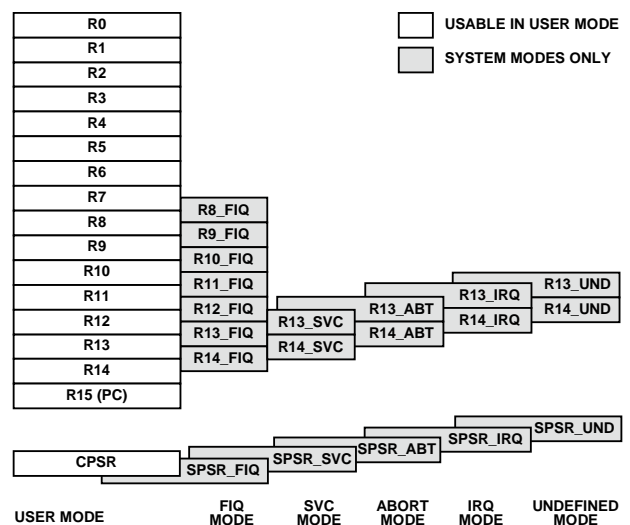


图14. 寄存器结构图

有关程序员模型和ARM7TDMI内核架构的详情，请参见ARM公司的ARM7TDMI技术和ARM架构手册。

## 中断延迟

快速中断请求(FIQ)的最大延迟时间包含：请求通过同步器的最长时间、最长指令(LDM)完成载入所有寄存器(包括PC)的时间、数据中止入口时间和FIQ入口时间。

在这个时间段的末尾，ARM7TDMI执行在0X1C(FIQ中断矢量地址)中的指令。最长总延迟时间为50个处理器周期，在系统采用连续41.78 MHz处理器时钟时，略小于1.2  $\mu$ s。

中断请求(IRQ)最大延迟时间计算方法也类似，但必须考虑到FIQ优先级更高，可能任意延长进入IRQ处理例行程序的时间。如果不使用LDM命令，这个时间可以缩短到42个周期。一些编译器可以选择不使用这个命令进行编译。另一个选择是在Thumb模式下运行器件，可以将时间缩短至22个周期。

用于FIQ或IRQ的最小中断延迟时间总共有5个周期，包括请求通过同步器的最短时间和进入异常模式的时间。

优先模式中(例如执行中断服务程序)，ARM7TDMI通常运行于32位的ARM模式。

## 存储器结构

ADuC7023共有两个独立存储器模块：8 KB的SRAM和64 KB的片内Flash/EE存储器，其中62 KB的片内Flash/EE可供用户使用，剩余2 KB被保留用于厂家配置BOOT页面。这两个存储器模块的映射如图15所示。

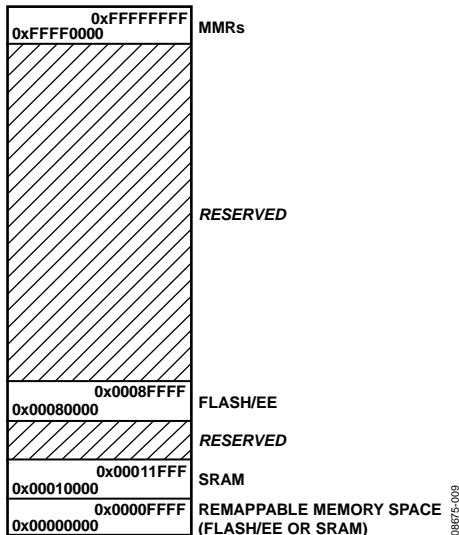


图15. 物理存储器映射图

默认情况下，复位之后，Flash/EE存储器被镜像到地址0x00000000。通过将Remap寄存器的0位清0，可以重新把SRAM映射到0x00000000。这种重映射功能在Flash/EE存储器部分有更详细描述。

### 存储器访问

ARM7内核把存储器看成是 $2^{32}$ 个字节的一个线性阵列。不同的存储器模块映射如图15所示。

ADuC7023的存储器被配置成从小到大顺序格式，这就是说，LSB位于最低字节地址，MSB位于最高字节地址。

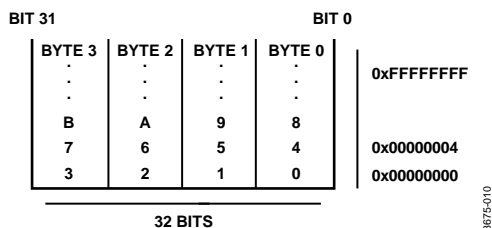


图 16. 从小到大顺序格式

### FLASH/EE存储器

64 kB的Flash/EE存储器为 $32\text{ k} \times 16$ 位格式，其中 $31\text{ k} \times 16$ 位是用户空间， $1\text{ k} \times 16$ 位保留用于芯片内核。Flash/EE的页面大小为512个字节。

62 KB的Flash/EE存储器可以存储用户代码和非易失性数据。数据和代码之间没有区别，因为ARM代码及数据共用同一空间。Flash/EE存储器的实际宽度为16位，这意味着在ARM模式下每一次执行一个指令(32位指令)，必须读取两次Flash/EE存储器。因此，当从Flash/EE存储器中执行程序时，建议使用Thumb模式来优化存取速度。以Thumb模式存取Flash/EE存储器的最大速度为41.78 MHz，而相应的以全ARM模式为20.89 MHz。更多关于Flash/EE存取时间的描述可参见SRAM和Flash/EE中的执行时间部分。

### SRAM

用户可以使用8KB的SRAM，它的组织形式为 $2\text{ k} \times 32$ 位，即两个字。如果SRAM被配置成32位宽的存储器阵列，ARM代码可以直接在SRAM中以41.78 MHz的速度执行。更多关于SRAM存取时间的描述可参见SRAM和Flash/EE中的执行时间部分。

### 存储器映射寄存器

存储器映射寄存器(MMR)空间被映射到存储器阵列的最上方两页。我们可以通过对ARM7寄存器组的间接寻址来存取存储器映射寄存器内的信息。

MMR空间为CPU和所有片内外设提供接口。除了内核寄存器，所有的寄存器都位于寄存器区域内。图17内的阴影区域为未占用区域或保留区域，不允许用户程序访问该区域。表10至表23为所有寄存器存储器映射。

读取或写入一个寄存器所需的存取时间取决于高级微控制器总线结构(AMBA)总线，该总线可用来访问外围设备。处理器有两个AMBA总线：高级性能总线(AHB)用于系统模块，高级外设总线(APB)用于低性能外围设备。访问AHB需要一个周期，访问APB需要两个周期。除了Flash/EE存储器和通用输入输出端口以外，ADuC7023中的所有外围设备均位于APB上。

0xFFFFFFFF	
0xFFFF820	FLASH CONTROL INTERFACE
0xFFFF800	
0xFFFF46C	GPIO
0xFFFF400	
0xFFFF0BF	PWM
0xFFFF0F80	
0xFFFF0B54	PLA
0xFFFF0B00	
0xFFFF0A14	SPI
0xFFFF0A00	
0xFFFF0948	I <sup>2</sup> C1
0xFFFF0900	
0xFFFF0848	I <sup>2</sup> C0
0xFFFF0800	
0xFFFF0620	DAC
0xFFFF0600	
0xFFFF0538	ADC
0xFFFF0500	
0xFFFF0490	BAND GAP REFERENCE
0xFFFF048C	
0xFFFF0448	POWER SUPPLY MONITOR
0xFFFF0440	
0xFFFF0420	PLL AND OSCILLATOR CONTROL
0xFFFF0404	
0xFFFF0370	WATCHDOG TIMER
0xFFFF0360	
0xFFFF0334	GENERAL-PURPOSE TIMER
0xFFFF0320	
0xFFFF0310	TIMER0
0xFFFF0300	
0xFFFF0238	REMAP AND SYSTEM CONTROL
0xFFFF0220	
0xFFFF0140	INTERRUPT CONTROLLER
0xFFFF0000	

08675-011

图 17. 存储器映射寄存器

表10. IRQ基地址 = 0xFFFF0000

地址	名称	字节	访问类型	默认值	描述
0x0000	IRQSTA	4	R	0x00000000	活动IRQ源。
0x0004	IRQSIG	4	R		所有IRQ源的当前状态(使能和禁用)。
0x0008	IRQEN	4	R/W	0x00000000	使能的IRQ源。
0x000C	IRQCLR	4	W		MMR禁用IRQ源。
0x0010	SWICFG	4	W		软件中断配置MMR。
0x0014	IRQBASE	4	R/W	0x00000000	所有矢量的基地址。指向64字节存储块的起始地址的指针；最多可以包含32个指向不同例行处理子程序的指针。
0x001C	IRQVEC	4	R	0x00000000	该寄存器内包含与当前活动IRQ源对应的例行处理子程序的地址。
0x0020	IRQP0	4	R/W	0x00000000	此寄存器包含用于设置中断源1至中断源7的中断优先级。中断优先级可被设置为0至7。
0x0024	IRQP1	4	R/W	0x00000000	此寄存器包含用于设置中断源8至中断源15的中断优先级。
0x0028	IRQP2	4	R/W	0x00000000	此寄存器包含用于设置中断源16至中断源21的中断优先级。
0x002C	RESERVED	4	R/W	0x00000000	保留。
0x0030	IRQCONN	4	R/W	0x00000000	使能IRQ和FIQ中断嵌套。
0x0034	IRQCONE	4	R/W	0x00000000	此寄存器将外部中断源设置为上升沿/下降沿/电平触发。
0x0038	IRQCLRE	4	R/W	0x00000000	清除边沿/电平触发中断源。
0x003C	IRQSTAN	4	R/W	0x00000000	此寄存器指示最新导致中断异常的中断优先级。
0x0100	FIQSTA	4	R	0x00000000	活动FIQ源。
0x0104	FIQSIG	4	R		所有FIQ源的当前状态(使能和禁用)。
0x0108	FIQEN	4	R/W	0x00000000	使能的FIQ源。
0x010C	FIQCLR	4	W		寄存器禁用FIQ源。
0x011C	FIQVEC	4	R	0x00000000	FIQ中断矢量。
0x013C	FIQSTAN	4	R/W	0x00000000	此寄存器指示最新导致FIQ异常的FIQ优先级。

表11. 系统控制基地址 = 0xFFFF0200

地址	名称	字节	访问类型	默认值 <sup>1</sup>	描述
0x0220	Remap <sup>2</sup>	1	R/W	0x00	重映射控制寄存器。
0x0230	RSTSTA	1	R/W	0x01	RSTSTA状态寄存器。
0x0234	RSTCLR	1	W	0x00	RSTCLR寄存器用于清除RSTSTA寄存器。
0x0248	RSTKEY1	1	W	0xFF	写入RSTCFG之前，0xFF应当写入此寄存器。
0x024C	RSTCFG	1	R/W	0x00	此寄存器让DAC和GPIO输出可以在看门狗或软件复位之后保持状态不变。
0x0250	RSTKEY2	1	W	0xFF	写入RSTCFG之后，0xFF应当写入此寄存器。

<sup>1</sup> N/A表示不适用。<sup>2</sup> 由内核更新。



# ADuC7023

**表12. 定时器基地址 = 0xFFFF0300**

地址	名称	字节	访问类型	默认值 <sup>1</sup>	描述
0x0300	TOLD	2	R/W	0x0000	定时器0载入寄存器。
0x0304	TOVAL	2	R	0xFFFF	定时器0值寄存器。
0x0308	TOCON	2	R/W	0x0000	定时器0控制MMR。
0x030C	TOCLRI	1	W	0xXX	定时器0中断清除寄存器。
0x0320	T1LD	4	R/W	0x00000000	定时器1载入寄存器。
0x0324	T1VAL	4	R	0xFFFFFFFF	定时器1值寄存器。
0x0328	T1CON	4	R/W	0x00000000	定时器1控制MMR。
0x032C	T1CLRI	1	W	0xXX	定时器1中断清除寄存器。
0x0330	T1CAP	4	R	0x00000000	定时器1捕捉寄存器。
0x0360	T2LD	2	R/W	0x0000	定时器2载入寄存器。
0x0364	T2VAL	2	R	0xFFFF	定时器2值寄存器。
0x0368	T2CON	2	R/W	0x0000	定时器2控制MMR。
0x036C	T2CLRI	1	W	0xXX	定时器2中断清除寄存器。

<sup>1</sup>N/A表示不适用。

**表13. PLL/PSM基地址 = 0xFFFF0400**

地址	名称	字节	访问类型	默认值 <sup>1</sup>	描述
0x0404	POWKEY1	2	W	0xFFFF	POWCON0预写密钥。
0x0408	POWCON0	1	R/W	0x00	电源控制和内核速度控制寄存器。
0x040C	POWKEY2	2	W	0xFFFF	POWCON0写后密钥。
0x0410	PLLKEY1	2	W	0xFFFF	PLLCON预写密钥。
0x0414	PLLCON	1	R/W	0x21	PLL时钟源选择MMR。
0x0418	PLLKEY2	2	W	0xFFFF	PLLCON写后密钥。
0x0434	POWKEY3	2	W	0xFFFF	POWCON1预写密钥。
0x0438	POWCON1	2	R/W	0x0004	电源控制和内核速度控制寄存器。
0x043C	POWKEY4	2	W	0xFFFF	POWCON1写后密钥。
0x0440	PSMCON	2	R/W	0x0008	电源监控器控制寄存器。
0x0444	CMPCON	2	R/W	0x0000	比较器控制寄存器。

<sup>1</sup>N/A表示不适用。

**表14. 基准电压基地址 = 0xFFFF0480**

地址： 0x048c

名称： REFCON

字节： 1

访问类型： 读/写

默认值： 0x00

描述： 基准电压控制寄存器。

**表15. ADC基地址 = 0xFFFF0500**

地址	名称	字节	访问类型	默认值	描述
0x0500	ADCCON	2	R/W	0x0600	ADC控制寄存器。
0x0504	ADCCP	1	R/W	0x00	ADC正向通道选择寄存器。
0x0508	ADCCN	1	R/W	0x01	ADC反向通道选择寄存器。
0x050C	ADCSTA	1	R	0x00	ADC状态MMR。
0x0510	ADCDAT	4	R	0x00000000	ADC数据输出寄存器。
0x0514	ADCRST	1	R/W	0x00	ADC复位寄存器。



地址	名称	字节	访问类型	默认值	描述
0x0530	ADCGN	2	R/W	厂家配置	ADC增益校准寄存器。
0x0534	ADCOF	2	R/W	厂家配置	ADC失调校准寄存器。
0x0544	TSCON	1	R/W	0x00	温度传感器斩波使能寄存器。
0x0548	TEMPREF	2	R/W	厂家配置	温度传感器参考值。

表16. DAC基地址 = 0xFFFF0600

地址	名称	字节	访问类型	默认值	描述
0x0600	DAC0CON	1	R/W	0x00	DAC0控制寄存器。
0x0604	DAC0DAT	4	R/W	0x00000000	DAC0数据寄存器。
0x0608	DAC1CON	1	R/W	0x00	DAC1控制寄存器。
0x060C	DAC1DAT	4	R/W	0x00000000	DAC1数据寄存器。
0x0610	DAC2CON	1	R/W	0x00	DAC2控制寄存器。
0x0614	DAC2DAT	4	R/W	0x00000000	DAC2数据寄存器。
0x0618	DAC3CON	1	R/W	0x00	DAC3控制寄存器。
0x061C	DAC3DAT	4	R/W	0x00000000	DAC3数据寄存器。
0x0654	DACBCFG	1	R/W	0x00	DAC配置寄存器
0x0650	DACBKEY0	2	W	0x0000	DAC密钥0寄存器
0x0658	DACBKEY1	2	W	0x0000	DAC密钥1寄存器

表17. I<sup>2</sup>C0基地址 = 0xFFFF0800

地址	名称	字节	访问类型	默认值	描述
0x0800	I2C0MCON	2	R/W	0x0000	I <sup>2</sup> C0主机控制寄存器。
0x0804	I2C0MSTA	2	R	0x0000	I <sup>2</sup> C0主机状态寄存器。
0x0808	I2C0MRX	1	R	0x00	I <sup>2</sup> C0主机接收寄存器。
0x080C	I2C0MTX	1	W	0x00	I <sup>2</sup> C0主机发送寄存器。
0x0810	I2C0MCNT0	2	R/W	0x0000	I <sup>2</sup> C0主机读取计数寄存器。在从从机设备中读取数据之前，须向该寄存器写入需要读取的字节数。
0x0814	I2C0MCNT1	1	R	0x00	I <sup>2</sup> C0主机当前读取计数寄存器。该寄存器用于记录已从从机设备中接收的字节数。
0x0818	I2C0ADR0	1	R/W	0x00	I <sup>2</sup> C0地址字节寄存器。在与从机进行通信前，向该寄存器内写入目标从机地址。
0x081C	I2C0ADR1	1	R/W	0x00	I <sup>2</sup> C0地址字节寄存器。在与从机进行通信前，向该寄存器内写入目标从机地址。仅适用于10位模式。
0x0824	I2C0DIV	2	R/W	0x1F1F	I <sup>2</sup> C0时钟控制寄存器。用于设置SCL频率。
0x0828	I2C0SCON	2	R/W	0x0000	I <sup>2</sup> C0从机控制寄存器。
0x082C	I2C0SSTA	2	R/W	0x0000	I <sup>2</sup> C0从机状态寄存器。
0x0830	I2C0SRX	1	R	0x00	I <sup>2</sup> C0从机接收寄存器。
0x0834	I2C0STX	1	W	0x00	I <sup>2</sup> C0从机发送寄存器。
0x0838	I2C0ALT	1	R/W	0x00	I <sup>2</sup> C0硬件广播识别寄存器。
0x083C	I2C0ID0	1	R/W	0x00	I <sup>2</sup> C0从机识别号ID0寄存器。从机总线识别号寄存器。
0x0840	I2C0ID1	1	R/W	0x00	I <sup>2</sup> C0从机识别号ID1寄存器。从机总线识别号寄存器。
0x0844	I2C0ID2	1	R/W	0x00	I <sup>2</sup> C0从机识别号ID2寄存器。从机总线识别号寄存器。
0x0848	I2C0ID3	1	R/W	0x00	I <sup>2</sup> C0从机识别号ID3寄存器。从机总线识别号寄存器。
0x084C	I2C0FSTA	2	R/W	0x0000	I <sup>2</sup> C0 FIFO状态寄存器。可应用于主机模式和从机模式。

表18. I<sup>2</sup>C1基地址 = 0xFFFF0900

地址	名称	字节	访问类型	默认值	描述
0x0900	I2C1MCON	2	R/W	0x0000	I <sup>2</sup> C1主机控制寄存器。
0x0904	I2C1MSTA	2	R	0x0000	I <sup>2</sup> C1主机状态寄存器。
0x0908	I2C1MRX	1	R	0x00	I <sup>2</sup> C1主机接收寄存器。
0x090C	I2C1MTX	1	W	0x00	I <sup>2</sup> C1主机发送寄存器。
0x0910	I2C1MCNT0	2	R/W	0x0000	I <sup>2</sup> C1主机读取计数寄存器。在从从机设备中读取数据之前，须向该寄存器写入需要读取的字节数。

# ADuC7023

地址	名称	字节	访问类型	默认值	描述
0x0914	I2C1MCNT1	1	R	0x00	I <sup>2</sup> C1主机当前读取计数寄存器。该寄存器用于记录已从从机设备中接收的字节数。
0x0918	I2C1ADR0	1	R/W	0x00	I <sup>2</sup> C1地址字节寄存器。在与从机进行通信前，向该寄存器内写入目标从机地址。
0x091C	I2C1ADR1	1	R/W	0x00	I <sup>2</sup> C1地址字节寄存器。在与从机进行通信前，向该寄存器内写入目标从机地址。仅适用于10位模式。
0x0924	I2C1DIV	2	R/W	0x1F1F	I <sup>2</sup> C1时钟控制寄存器。用于设置SCL频率。
0x0928	I2C1SCON	2	R/W	0x0000	I <sup>2</sup> C1从机控制寄存器。
0x092C	I2C1SSTA	2	R/W	0x0000	I <sup>2</sup> C1从机状态寄存器。
0x0930	I2C1SRX	1	R	0x00	I <sup>2</sup> C1从机接收寄存器。
0x0934	I2C1STX	1	W	0x00	I <sup>2</sup> C1从机发送寄存器。
0x0938	I2C1ALT	1	R/W	0x00	I <sup>2</sup> C1硬件广播识别寄存器。
0x093C	I2C1ID0	1	R/W	0x00	I <sup>2</sup> C1从机识别号ID0寄存器。从机总线识别号寄存器。
0x0940	I2C1ID1	1	R/W	0x00	I <sup>2</sup> C1从机识别号ID1寄存器。从机总线识别号寄存器。
0x0944	I2C1ID2	1	R/W	0x00	I <sup>2</sup> C1从机识别号ID2寄存器。从机总线识别号寄存器。
0x0948	I2C1ID3	1	R/W	0x00	I <sup>2</sup> C1从机识别号ID3寄存器。从机总线识别号寄存器。
0x094C	I2C1FSTA	2	R/W	0x0000	I <sup>2</sup> C1 FIFO状态寄存器。可应用于主机模式和从机模式。

**表19. SPI基地址 = 0xFFFF0A00**

地址	名称	字节	访问类型	默认值	描述
0x0A00	SPISTA	2	R	0x0000	SPI状态MMR。
0x0A04	SPIRX	1	R	0x00	SPI接收MMR。
0x0A08	SPLITX	1	W	0xXX	SPI发送MMR。
0x0A0C	SPIDIV	1	R/W	0x00	SPI波特率选择寄存器。
0x0A10	SPICON	2	R/W	0x0000	SPI控制MMR。

**表20. PLA基地址 = 0xFFFF0B00**

地址	名称	字节	访问类型	默认值	描述
0x0B00	PLAELM0	2	R/W	0x0000	PLA单元0控制寄存器。
0x0B04	PLAELM1	2	R/W	0x0000	PLA单元1控制寄存器。
0x0B08	PLAELM2	2	R/W	0x0000	PLA单元2控制寄存器。
0x0B0C	PLAELM3	2	R/W	0x0000	PLA单元3控制寄存器。
0x0B10	PLAELM4	2	R/W	0x0000	PLA单元4控制寄存器。
0x0B14	PLAELM5	2	R/W	0x0000	PLA单元5控制寄存器。
0x0B18	PLAELM6	2	R/W	0x0000	PLA单元6控制寄存器。
0x0B1C	PLAELM7	2	R/W	0x0000	PLA单元7控制寄存器。
0x0B20	PLAELM8	2	R/W	0x0000	PLA单元8控制寄存器。
0x0B24	PLAELM9	2	R/W	0x0000	PLA单元9控制寄存器。
0x0B28	PLAELM10	2	R/W	0x0000	PLA单元10控制寄存器。
0x0B2C	PLAELM11	2	R/W	0x0000	PLA单元11控制寄存器。
0x0B30	PLAELM12	2	R/W	0x0000	PLA单元12控制寄存器。
0x0B34	PLAELM13	2	R/W	0x0000	PLA单元13控制寄存器。
0x0B38	PLAELM14	2	R/W	0x0000	PLA单元14控制寄存器。
0x0B3C	PLAELM15	2	R/W	0x0000	PLA单元15控制寄存器。
0x0B40	PLACLK	1	R/W	0x00	PLA时钟选择寄存器。
0x0B44	PLAIRQ	4	R/W	0x00000000	PLA中断控制寄存器。
0x0B48	PLAADC	4	R/W	0x00000000	PLA ADC触发器控制寄存器。
0x0B4C	PLADIN	4	R/W	0x00000000	PLA数据输入寄存器。
0x0B50	PLADOUT	4	R	0x00000000	PLA数据输出寄存器。
0x0B54	PLALCK	1	W	0x00	PLA锁寄存器。

表21. PWM基地址 = 0xFFFF0F80

地址	名称	字节	访问类型	默认值	描述
0x0F80	PWMCON1	2	R/W	0x0012	PWM控制寄存器1。完整详情见脉宽调制器部分。
0x0F84	PWM0COM0	2	R/W	0x0000	用于比较PWM输出0和PWM输出1的比较寄存器0。
0x0F88	PWM0COM1	2	R/W	0x0000	用于比较PWM输出0和PWM输出1的比较寄存器1。
0x0F8C	PWM0COM2	2	R/W	0x0000	用于比较PWM输出0和PWM输出1的比较寄存器2。
0x0F90	PWM0LEN	2	R/W	0x0000	用于PWM输出0和PWM输出1的频率控制寄存器。
0x0F94	PWM1COM0	2	R/W	0x0000	用于比较PWM输出2和PWM输出3的比较寄存器0。
0x0F98	PWM1COM1	2	R/W	0x0000	用于比较PWM输出2和PWM输出3的比较寄存器1。
0x0F9C	PWM1COM2	2	R/W	0x0000	用于比较PWM输出2和PWM输出3的比较寄存器2。
0x0FA0	PWM1LEN	2	R/W	0x0000	用于PWM输出2和PWM输出3的频率控制寄存器。
0x0FA4	PWM2COM0	2	R/W	0x0000	用于比较PWM输出4和PWM输出5的比较寄存器0。
0x0FA8	PWM2COM1	2	R/W	0x0000	用于比较PWM输出4和PWM输出5的比较寄存器1。
0x0FB8	PWMCLRI	2	W	0x0000	PWM中断清除寄存器。向其中写入任意值就可以清除PWM中断源。

表22. GPIO基地址 = 0xFFFF400

地址	名称	字节	访问类型	默认值	描述
0xF400	GP0CON	4	R/W	0x00001111	GPIO端口0控制寄存器。
0xF404	GP1CON	4	R/W	0x00000000	GPIO端口1控制寄存器。
0xF408	GP2CON	4	R/W	0x00000000	GPIO端口2控制寄存器。
0xF420	GP0DAT	4	R/W	0x000000XX	GPIO端口0数据控制寄存器。
0xF424	GP0SET	4	W	0x000000XX	GPIO端口0数据设置寄存器。
0xF428	GP0CLR	4	W	0x000000XX	GPIO端口0数据清除寄存器。
0xF42C	GP0PAR	4	R/W	0x22220000	GPIO端口0数据清除寄存器。
0xF430	GP1DAT	4	R/W	0x000000XX	GPIO端口1数据控制寄存器。
0xF434	GP1SET	4	W	0x000000XX	GPIO端口1数据控制寄存器。
0xF438	GP1CLR	4	W	0x000000XX	GPIO端口1数据清除寄存器。
0xF43C	GP1PAR	4	R/W	0x22000022	GPIO端口1上拉禁用寄存器。
0xF440	GP2DAT	4	R/W	0x000000XX	GPIO端口2数据控制寄存器。
0xF444	GP2SET	4	W	0x000000XX	GPIO端口2数据设置寄存器。
0xF448	GP2CLR	4	W	0x000000XX	GPIO端口2数据清除寄存器。
0xF44C	GP2PAR	4	R/W	0x00000000	GPIO端口2上拉禁用寄存器。

表23. Flash/EE基地址 = 0xFFFF800

地址	名称	字节	访问类型	默认值	描述
0xF800	FEESTA	1	R	0x20	Flash/EE状态MMR。
0xF804	FEEMOD	2	R/W	0x0000	Flash/EE控制MMR。
0xF808	FEECON	1	R/W	0x07	Flash/EE控制MMR。
0xF80C	FEEDAT	2	R/W	0xFFFF	Flash/EE数据MMR。
0xF810	FEEADR	2	R/W	0x0000	Flash/EE地址MMR。
0xF818	FEESIGN	3	R	0xFFFFFFFF	Flash/EE LFSR MMR。
0xF81C	FEEPRO	4	R/W	0x00000000	Flash/EE保护MMR。
0xF820	FEEHIDE	4	R/W	0xFFFFFFFF	Flash/EE保护MMR。

## ADC电路概览

该模数转换器(ADC)集成一个快速多通道12位ADC,工作电压为2.7 V至3.6 V,当时钟源为41.78 MHz时,其吞吐量最高可以达到1 MSPS。这个模块提供给用户一个多通道多路复用器、一个差分跟踪保持电路、一个片内基准电压源和一个ADC。

该ADC为一款基于两个电容DAC的12位逐次逼近型转换器。根据对输入信号的配置不同,ADC可以在三种不同的模式下工作:全差分模式(用于小信号和平衡信号)、单端模式(用于任意单端信号)或伪差分模式(用于任何单端信号输入,可以较好地抑制共模信号)。

当工作在单端模式或伪差分模式下时,转换器模拟输入范围为0 V至 $V_{REF}$ 。在全差分模式下,输入信号共模电压 $V_{CM}$ 在0 V至 $AV_{DD}$ 之间,最大幅值不超过 $2V_{REF}$ (见图18)。

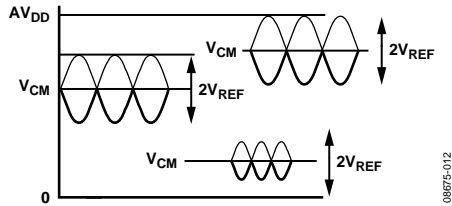


图18. 全差分模式下的平衡信号示例

片内提供一个高精度、低漂移、工厂校准的2.5 V基准电压源,也可以直接连接一个外部基准源,如后面“带隙基准电压源”一节所述。

单次或连续转换模式可由软件启动。外部 $CONV_{START}$ 引脚、片内PLA生成的输出、定时器0或定时器1溢出也可以用来生成重复性的ADC转换触发信号。

另外从片内带隙基准源输出且与绝对温度成正比的电压也可以通过ADC前端多路复用器进行中继。可选择此温度通道作为ADC输入端。这样可以很方便地形成一个内部温度传感器通道,用于测量芯片温度。

### 传递函数

#### 伪差分模式和单端模式

在伪差分或单端模式中,输入电压范围为0 V至 $V_{REF}$ ,输出编码为标准二进制编码

$$1 \text{ LSB} = FS/4096$$

$$2.5 \text{ V}/4096 = 0.61 \text{ mV或}$$

$$610 \mu\text{V}(V_{REF} = 2.5 \text{ V时})$$

理想编码转换发生在两个连续整数LSB值的中点,(即 $1/2 \text{ LSB}$ 、 $3/2 \text{ LSB}$ 、 $5/2 \text{ LSB}$ 、...、 $FS-3/2 \text{ LSB}$ )。理想输入/输出传递特性如图19所示。

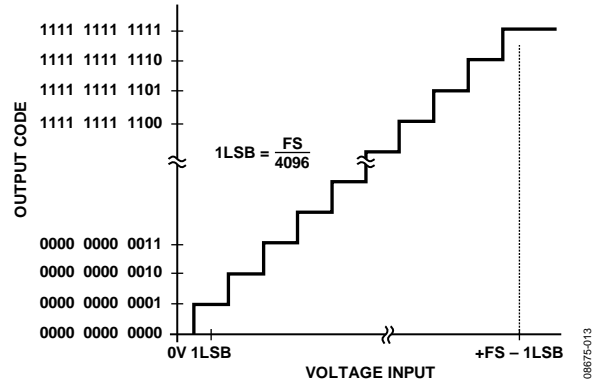


图19. 伪差分或单端模式下ADC传递函数

#### 全差分模式

差分信号的幅值为输入引脚 $V_{IN+}$ 和输入引脚 $V_{IN-}$ 的信号差值(即 $V_{IN+} - V_{IN-}$ )。因此,差分信号的最大幅值为 $-V_{REF}$ 至 $+V_{REF}$ 峰峰值(即 $2 \times V_{REF}$ )。这与共模模式(CM)无关。共模模式是两个输入信号的平均值,同时也是两个输入电压的中点,即 $(V_{IN+} + V_{IN-})/2$ 。这使得各输入的范围为 $CM \pm V_{REF}/2$ 。这一输入电压必须在外部设定,并且它的范围随着 $V_{REF}$ 而变化(参见“驱动模拟输入”部分)。

当 $V_{REF} = 2.5 \text{ V}$ 时,全差分模式下,输出编码为二进制补码,且 $1 \text{ LSB} = 2V_{REF}/4096$ ,或 $2 \times 2.5 \text{ V}/4096 = 1.22 \text{ mV}$ 。输出结果为 $\pm 11$ 位,但是这个结果向右移了一位。当用C语言时,允许ADCDAT寄存器中的结果可声明为带符号位整数。设计编码转换都发生在两个连续整数LSB值的中点(即 $1/2 \text{ LSB}$ 、 $3/2 \text{ LSB}$ 、 $5/2 \text{ LSB}$ 、...、 $FS-3/2 \text{ LSB}$ )。理想输入/输出传递特性如图20所示。

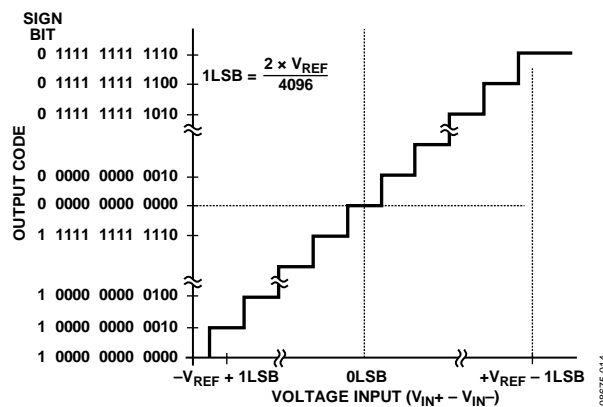


图20. 差分模式下ADC传递函数

**典型操作**

通过ADC控制寄存器和通道选择寄存器进行配置时，ADC转换器可转换模拟输入，并把一个12位的数据输出至ADC数据寄存器中。

高四位是符号位。12位转换结果存放在如图21所示寄存器中的16位至27位。需要注意的是，在全差分模式下，其结果是二进制补码格式。在伪差分模式和单端模式下，其结果是标准二进制格式。



图21. ADC转换结果格式

DAC×DAT内采用相同格式，以简化软件。

**功耗**

待机模式下，也就是上电但是没有转换情况下，ADC典型功耗为640 μA。使用内部基准电压源时，电流要增加140 μA。转换过程中，额外电流是0.3 μA乘以采样频率(单位为kHz)。

**时序**

ADC时序如图22所示。用户可以控制ADC时钟速度和ADCCON寄存器内采集时钟的数量。默认情况下，采集时间是八个时钟周期，时钟为两分频。附加时钟(如位检验或写入)个数可以设为19，这样采样速率为774 KSPS。针对温度传感器上的转换，设置ADCCON = 0x37A3。当使用包括温度传感器的多通道转换时，在读取温度传感器通道之后，定时设置就会恢复到用户自定义设定。

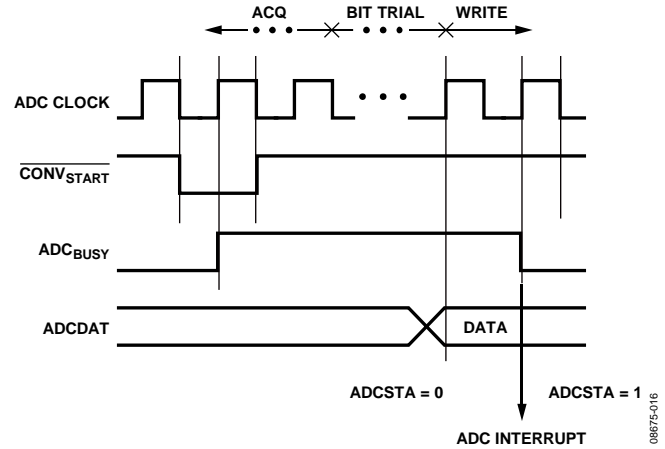


图22. ADC时序

**寄存器接口**

这部分我们介绍控制和配置ADC的8个寄存器。

**ADCCON寄存器**

名称: ADCCON

地址: 0xFFFF0500

默认值: 0x0600

访问类型: 读/写

功能: ADCCON是一个ADC控制寄存器，用户可以通过它使能ADC外设，选择ADC的工作模式(单端模式、伪差分模式、全差分模式)和转换类型等。该寄存器如表24所示。

表24. ADCCON寄存器位功能描述

位	值	描述
15至14		保留。
13		温度传感器转换使能。置1时执行温度传感器转换。置0时执行正常ADC转换。
12至10	000 001 010 011 100 101	ADC时钟速度。 f <sub>ADC</sub> /1.该分频器用于在外部时钟频率小于41.78 MHz时获得1MSPS ADC。 f <sub>ADC</sub> /2(默认值)。 f <sub>ADC</sub> /4。 f <sub>ADC</sub> /8。 f <sub>ADC</sub> /16。 f <sub>ADC</sub> /32。
9至8	00 01 10 11	ADC采集时间。 2个时钟。 4个时钟。 8个时钟(默认值)。 16个时钟。

# ADuC7023

位	值	描述
7		转换启动使能。 通过将该位置1, 用户可启动任意类型的转换命令。 通过将该位清0, 用户可禁用转换启动(清0该位不能让正在进行的连续转换停止)。
6		保留
5		ADC电源控制。 通过将该位置1, 用户可将ADC置于正常模式(ADC必须至少上电5 $\mu$ s, 然后才能正确转换)。 通过将该位清0, 用户可将ADC置于掉电模式。
4至3	00 01 10 11	转换模式。 单端模式。 差分模式。 伪差分模式。 保留。
2至0	000 001 010 011 100 101 其它	转换类型。 CONV <sub>START</sub> 引脚输入作为转换使能信号。 定时器1输出作为转换使能信号。 定时器0输出作为转换使能信号。 单次软件转换。转换后, 此位设为000(注意, 为了避免CONV <sub>START</sub> 引脚再一次触发转换, ADCCON寄存器的第13位在启动单次软件转换后应置1)。 连续软件转换。 PLA转换。 保留。

## ADCCP寄存器

名称:	ADCCP
地址:	0xFFFF0504
默认值:	0x00
访问类型:	读/写
功能:	ADCCP是ADC正向通道选择寄存器。 该寄存器如表25所示。

位	值	描述
	01100	ADC12 <sup>1</sup> .
	01101	保留
	01110	DAC0
	01111	DAC1
	10000	温度传感器。
	10001	AGND(自诊断特性)。
	10010	内部基准电压(自诊断特性)。
	10011	$AV_{DD}/2$ 。
	其它	保留。

**表25. ADCCP寄存器位功能描述**

位	值	描述
7至5		保留。
4至0		正向通道选择位。
	00000	ADC0.
	00001	ADC1.
	00010	ADC2.
	00011	ADC3.
	00100	ADC4 <sup>1</sup> .
	00101	ADC5 <sup>1</sup> .
	00110	ADC6 <sup>1</sup> .
	00111	ADC7 <sup>1</sup> .
	01000	ADC8 <sup>1</sup> .
	01001	ADC9 <sup>1</sup> .
	01010	ADC10 <sup>1</sup> .
	01011	保留。

<sup>2</sup> 如果所选ADC通道是与一个GPIO共享的, 则此引脚默认配置为使能弱上拉电阻。该上拉电阻应在相应的GPxPAR寄存器中手动禁用。请注意, 40引脚封装中P2.0/AIN12上的内置上拉电阻无法禁用。



**ADCCN寄存器**

名称: ADCCN  
 地址: 0xFFFFF0508  
 默认值: 0x01  
 访问类型: 读/写  
 功能: ADCCN是一个ADC反向通道选择寄存器。该寄存器如表26所示。

**表26. ADCCN寄存器位功能描述**

位	值	描述
7至5		保留。
4至0		反向通道选择位。
	00000	ADC0.
	00001	ADC1.
	00010	ADC2.
	00011	ADC3.
	00100	ADC4.
	00101	ADC5.
	00110	ADC6.
	00111	ADC7.
	01000	ADC8.
	01001	ADC9.
	01010	ADC10.
	01011	保留
	01100	ADC12.
	01101	保留
	01110	保留
	01111	DAC1.
	10000	温度传感器。
	10001	AGND(自诊断特性)。
	10010	内部基准电压(自诊断特性)。
	10011	保留
	其它	保留

**ADCSTA寄存器**

名称: ADCSTA  
 地址: 0xFFFFF050C  
 默认值: 0x00  
 访问类型: 读取  
 功能: ADCSTA是一个ADC状态寄存器, 指示ADC转换结果已完成。ADCSTA寄存器只有一个位, 即ADCReady(位0), 表示ADC的转换状态。在一次ADC转换完成后该位将该位置1, 并且产生一个ADC中断。当读取ADCDAT寄存器时, 该位自动清0。在ADC进行转换时, 也可以通过外部ADC<sub>BUSY</sub>引脚读取ADC的工作状态。在转换期间, 该引脚为高电平; 当转换结束后, ADC<sub>BUSY</sub>引脚变为低电平。如果通过ADCCON寄存器使能, 则可以在P0.0引脚输出ADC<sub>BUSY</sub>的状态(参见“通用输入/输出”部分)。

**ADCDAT寄存器**

名称: ADCDAT  
 地址: 0xFFFFF0510  
 默认值: 0x00000000  
 访问类型: 读取  
 功能: ADCDAT为ADC数据结果寄存器。里面存放12位ADC转换结果数据。如图21所示。

**ADCRST寄存器**

名称: ADCRST  
 地址: 0xFFFFF0514  
 默认值: 0x00  
 访问类型: 读/写  
 功能: ADCRST可以复位ADC的数字接口。通过向ADCRST中写入任意数据, 可恢复所有ADC寄存器到默认值。

# ADuC7023

## ADCGN寄存器

名称: ADCGN  
地址: 0xFFFF0530  
默认值: 厂家配置  
访问类型: 读/写  
功能: ADCGN是一个10位增益校准寄存器。

## ADCOF寄存器

名称: ADCOF  
地址: 0xFFFF0534  
默认值: 厂家配置  
访问类型: 读/写  
功能: ADCOF是一个10位失调校准寄存器。

## 转换器操作

这款ADC集成了一个包含电荷采样输入级的逐次逼近型(SAR)结构。该结构可在三种模式下工作: 差分模式、伪差分模式、单端模式。

## 差模

ADuC7023包含一个基于两个容性DAC的逐次逼近型ADC。图23和图24分别为ADC采样阶段和转换阶段的简化原理示意图。ADC由控制逻辑、一个SAR和两个容性DAC组成。在信号采样阶段,如图23所示,SW3闭合,SW1和SW2都置于A上,比较器保持在平衡状态,采样电容阵列充电,采集输入端的差分信号。

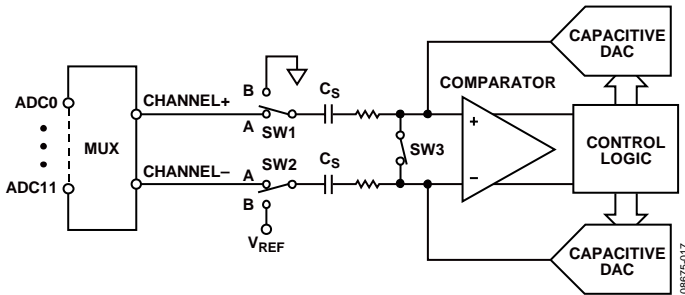


图23. ADC采样阶段

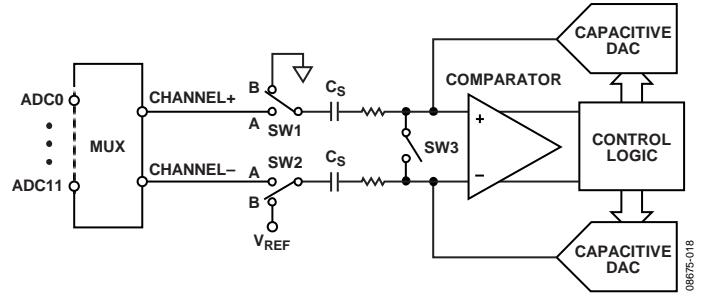


图24. ADC转换阶段

当ADC启动转换,如图24所示,SW3断开,而SW1和SW2移至位置B。这使得比较器变得不平衡。一旦转换开始,两个输入均会断开。控制逻辑和电荷再分配DAC可以加上和减去采样电容阵列中的固定电荷数量,使得比较器恢复到平衡状态。当比较器重新平衡后,转换就已经完成。控制逻辑产生ADC的输出代码。注意这里驱动 $V_{IN+}$ 和 $V_{IN-}$ 引脚的源输出阻抗一定要匹配,否则由于两个输入的建立时间不同会产生错误。

## 伪差分模式

在伪差分模式中,负通道(Channel-)连接到ADuC7023的 $V_{IN-}$ 引脚上,SW2开关在A(Channel-)和B( $V_{REF}$ )之间进行切换。 $V_{IN-}$ 引脚必须接地或者接一低电压。 $V_{IN+}$ 上的输入信号的范围为 $V_{IN-}$ 至 $V_{REF} + V_{IN-}$ 。注意这里 $V_{IN-}$ 必须恰当选择,不要使 $V_{REF} + V_{IN-}$ 超过 $AV_{DD}$ 。

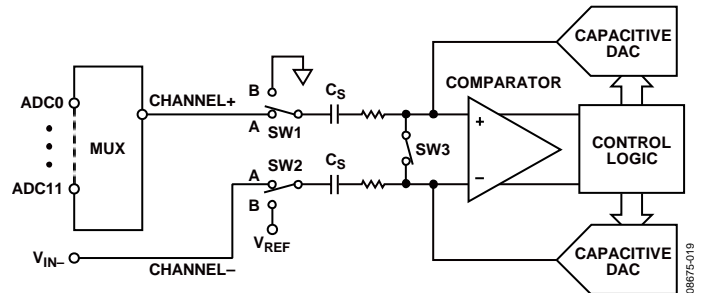


图25. 伪差分模式下ADC



## 单端模式

在单端模式下，SW2始终内部接地。V<sub>IN-</sub>引脚可悬空。V<sub>IN+</sub>引脚上的输入信号范围为0 V至V<sub>REF</sub>。

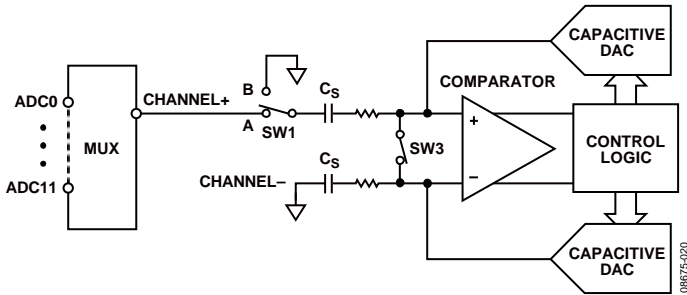


图26. 单端模式下ADC

## 模拟输入结构

ADC模拟输入结构等效电路如图27所示，图中4个二极管为模拟输入提供ESD保护。切记，模拟输入信号不得超过供电轨300 mV以上，否则会造成二极管正偏，并开始向基板内导通电流。这些二极管可以导通但不会导致器件彻底损坏的最大电流为10 mA。

图27中，电容C1典型值为4 pF，可基本上被归类为引脚寄生电容。电阻是由开关阻抗构成的集总元件。电阻典型值为100 Ω左右；电容C2为ADC采样电容，典型值为16 pF。

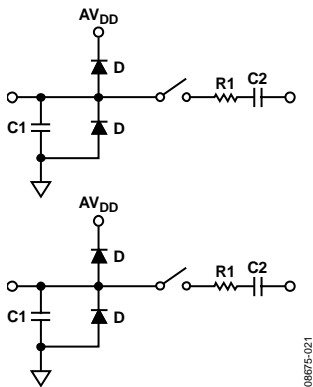


图27. 等效模拟输入电路转换阶段：开关打开，  
采样阶段：开关关闭

在交流应用中，建议在相应的模拟输入引脚用一个RC低通滤波器来滤除模拟输入信号的高频成份。在对谐波失真和信噪比要求严格的应用中，模拟输入应采用一个低阻抗源进行驱动。高源阻抗会显著影响ADC的交流特性。这种情况下有必要使用一个输入缓冲放大器。通常根据具体应用来选择运算放大器。图28和图29为ADC前端的示例。

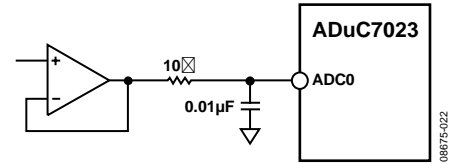


图28. 带缓冲的单端/伪差分输入

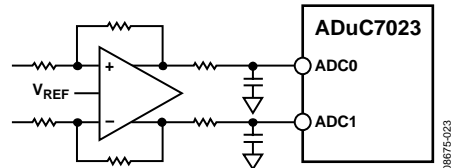


图29. 带缓冲的差分输入

不用放大器来驱动模拟输入端时，应将源阻抗的值限制在1 kΩ以下。源阻抗最大值取决于可容许的总谐波失真(THD)。总谐波失真随着输入源阻抗的增加而增大，从而导致ADC性能下降。

## 驱动模拟输入

该ADC既可以采用内部基准电压源，也可以采用外部基准电压源。在差分模式下工作时，共模输入信号(V<sub>CM</sub>)有严格的限制，它的大小取决于用来确保信号维持在供电轨之内的基准电压值和电源电压。表27给出了一些计算出的V<sub>CM</sub>最小值和最大值。

表27. V<sub>CM</sub> 范围

AV <sub>DD</sub>	V <sub>REF</sub>	V <sub>CM</sub> 最小值	V <sub>CM</sub> 最大值	信号峰峰值
3.3 V	2.5 V	1.25 V	2.05 V	2.5 V
	2.048 V	1.024 V	2.276 V	2.048 V
	1.25 V	0.75 V	2.55 V	1.25 V
3.0 V	2.5 V	1.25 V	1.75 V	2.5 V
	2.048 V	1.024 V	1.976 V	2.048 V
	1.25 V	0.75 V	2.25 V	1.25 V

## 校准

ADC失调寄存器(ADCOF)和增益系数寄存器为出厂设定值时，单次操作的端点误差和线性度可以达到最佳效果(参见“技术规格”部分)。如果需要系统进行校准，可以通过修改失调和增益系数的默认值来改善端点误差。但需要注意的是，任何对ADCOF和ADCGN出厂设定值的修改都会降低ADC的线性性能。

对于系统失调误差校正，必须将ADC通道输入级连接在AGND上。用软件控制实现连续转换环，在转换过程中不断修改ADCOF的值，直到ADC转换结果数据(ADCDAT)读数从0变为1。如果ADCDAT值大于1，应该降低ADCOF中的值，直到ADCDAT读数从0变为1。这种数字化校正失调误差的分辨率可以达到0.25 LSB，范围可达到V<sub>REF</sub>的±3.125%。

# ADuC7023

对于系统增益误差校正，必须将ADC通道输入级连接到 $V_{REF}$ 上。用软件控制实现连续转换环，在转换过程中不断修改ADCGN的值，直到ADCDAT读数从4094变为4095。如果ADCDAT值小于4094，应该降低ADCGN中的值，直到ADCDAT读数从4094变为4095。与失调误差校准一样，增益校准分辨率可以达到0.25 LSB，范围可达到 $V_{REF}$ 的 $\pm 3\%$ 。

## 温度传感器

ADuC7023提供从片内带隙基准电压源输出并与绝对温度成正比的电压。该电压可以通过前端ADC多路复用器连接到ADC模拟输入通道，这样就可以很方便地形成一个内部温度传感器通道，用于测量芯片温度。

ADC温度传感器转换与标准ADC电压不同。ADC性能规格并不适用于温度传感器。

内置放大器的斩波功能应使用TSCON寄存器来使能。要使能此模式，用户必须将TSCON的位0置1。用户还必须在此模式下取得两个连续的ADC读数并求取平均值。

ADCCON寄存器必须配置为0x37A3。

计算芯片温度的公式：

$$T - T_{REF} = (V_{ADC} - V_{TREF}) \times K$$

其中：

T为温度结果。

$T_{REF}$ 为25°C。

$V_{ADC}$ 为从两个连续转换结果得出的ADC转换结果平均值。

$V_{TREF}$ 为1369 mV，对应于 $T_{REF} = 25^\circ\text{C}$ ，如表1所述。

K为温度传感器模式下ADC的增益，具体由特性数据决定 ( $K = 0.2262^\circ\text{C}/\text{mV}$ )。它对应于表1所示的1/V TC规格。

使用表1中的默认值，并且不经任何校准，此公式变为：

$$T - 25^\circ\text{C} = (V_{ADC} - 1369) \times 0.2262$$

其中：

$V_{ADC}$ 单位为毫伏。

要提高精度，可对在受控温度值上执行单点校准。

对于未执行校准时的计算，( $T_{REF}, V_{TREF}$ ) = (25°C, 1369 mV)。单点校准的思路是使用其他已知的( $T_{REF}, V_{TREF}$ )值来取代各器件的通用值(25°C, 1369 mV)。

有些用户可能无法取得此类数值对。在这种情况下，ADuC7023会在TEMPREF寄存器中载入单点校准值。欲了解此寄存器的更多信息，请参见TEMPREF寄存器部分。

在ADuC7023的生产测试过程中，TEMPREF寄存器中载入了失调整因子。每个器件在EMPREF寄存器中拥有不同的值。借助此单点校准，可使用如下公式：

$$T - T_{REF} = (V_{ADC} - V_{TREF}) \times K$$

其中：

使用TEMPREF寄存器方法时 $T_{REF}$ 为27°C，但无法保证这一结果。

$T_{TREF}$ 可使用TEMPREF寄存器算出。

## TSCON寄存器

名称： TSCON

地址： 0xFFFF0544

默认值： 0x00

访问类型： 读/写

表28. TSCON寄存器位功能描述

位	描述
7至1	保留。
0	温度传感器斩波使能位。 通过将该位置1，可使能ADC内置放大器的斩波功能。 通过将该位清0，可禁用斩波。 该位的默认值为0。

## TEMPREF寄存器

名称： TEMPREF

地址： 0xFFFF0548

默认值： 厂家配置

访问类型： 读/写

表29. TEMPREF寄存器位功能描述

位	描述
15至9	保留
8	温度基准电压符号。
7至0	<p>温度传感器失调校准电压。要计算来自TEMPREF寄存器的<math>V_{TREF}</math>，请使用以下公式：</p> <p>如果TEMPREF符号为负，则从2292减去TEMPREF，即</p> $C_{TREF} = 2292 - TEMPREF[7:0]$ <p>其中： <math>TEMPREF[8] = 1</math>。</p> <p>或者</p> <p>如果TEMPREF符号为正，则将TEMPREF与2292相加，即</p> $C_{TREF} = TEMPREF[7:0] + 2292$ <p>其中： <math>TEMPREF[8] = 0</math>。</p> <p>那么：</p> $V_{TREF} = (C_{TREF} \times V_{REF}) / 4096 \times 1000$ <p>其中： <math>C_{TREF}</math>通过上文公式计算得出。 <math>V_{REF}</math>为2.5 V，即内置基准电压。</p> <p>将<math>V_{TREF}</math>插入以下公式：</p> $T - T_{REF} = (V_{ADC} - V_{TREF}) \times K$ <p>其中：</p> <p>当使用TEMPREF寄存器时，<math>T_{REF}</math>为27°C。 <math>V_{ADC}</math>为从两个连续转换结果得出的ADC转换结果平均值。 <math>V_{TREF}</math>通过上文公式计算得出。</p> <p>请注意，ADC码值2292是使用TEMPREF寄存器时的默认值。该值并不是精确值，仅可用于TEMPREF寄存器。</p>

### 带隙基准电压源

ADuC7023内置一个2.5 V的片内带隙基准电压源，后者可用于ADC和DAC。该内部基准电压也出现在 $V_{REF}$ 引脚上。当采用内部基准电压源时，外部 $V_{REF}$ 引脚与AGND之间必须接上一个0.47  $\mu$ F的电容，以确保ADC转换时的稳定性和快速响应。片内基准电压源也可以连接至一个外部引脚( $V_{REF}$ )，作为系统其它电路的基准电压源。

但是由于 $V_{REF}$ 输出的驱动能力较低，因此需要连接一个外部缓冲器。使用可编程选项，还可以在 $V_{REF}$ 引脚上连接外部基准电压输入。

### REFCON寄存器

名称：	REFCON
地址：	0xFFFF048C
默认值：	0x00
访问类型：	读/写
功能：	带隙基准电压接口包括一个8位REFCON寄存器，如表30所示。

表30. REFCON寄存器位功能描述

位	描述
7至2	保留
1	<p>内部基准电压关断位。</p> <p>通过将该位置1，可关断内部基准电压源。连接外部基准电压源时应将此位置1。</p> <p>通过将该位清0，可启用内部基准电压源。该位的默认值为0。</p>
0	<p>内部基准电压输出使能。</p> <p>通过将该位置1，用户可将内部2.5V基准电压源连接在<math>V_{REF}</math>上。基准电压源可用于外部元件，但需要增加缓冲。</p> <p>通过将该位清0，用户可以断开基准电压源与<math>V_{REF}</math>引脚的连接。</p>

如需给ADuC7023连接外部基准电压源，请设置REFCON = 0x00。ADC和DAC可配置为使用相同或不同的基准电压源。参见表42。

## 非易失性FLASH/EE存储器

ADuC7023片内集成了Flash/EE存储器技术，向用户提供非易失、在线可编程存储器空间。

像EEPROM一样，闪存也支持系统内字节编程(尽管必须在向存储器内写入新的数据之前将原有数据擦除)。擦除操作是以页块为单位进行的。因而，通常闪存也被准确地称作Flash/EE存储器。

由于具有非易失性、在线编程、高密度及低成本等特点，Flash/EE存储器是更理想的存储器件。利用ADuC7023集成的Flash/EE存储器，用户可以在线更新程序代码空间，而不必在远程操作节点处替换一次性可编程(OTP)设备。

每个器件均内置一个64 KB的Flash/EE存储器阵列，其中低62 kB可供用户使用，高2 KB包含永久性嵌入式固件，支持在线串行下载。另外，这2 KB嵌入式固件还包括上电配置程序，用于将出厂标定系数下载到各种已校准的外设，如ADC、温度传感器、带隙基准电压源等。用户代码无法访问这2 kB嵌入式固件。

### Flash/EE存储器可靠性

器件上的Flash/EE存储器阵列有两个关键的特性：周期耐久性和数据保持力。

耐久性用于衡量Flash/EE存储器重复多个编程、读取及擦除周期的能力。一个单耐久性周期包括4个独立的、连续的事件，定义如下：

1. 初始页面擦除序列。
2. 读/校验时序(单个Flash/EE)。
3. 存储器的字节编程时序。
4. 二次读/校验时序(耐久性周期)。

在进行可靠性验证时，Flash/EE存储器中的三页(顶、中和底)的每半个字(16位宽)可循环存取10000次(从0x0000至0xFFFF)。如表1所示，Flash/EE存储器耐久性是依据JEDEC保持期限规格A117在-40°至+125°C工业温度范围内测量出来的。这个结果满足给定温度下最小耐久性超过10,000个周期的规格。

保持期限衡量Flash/EE存储器长时间保持编程数据的能力。同样，在特定结温( $T_j = 85^\circ\text{C}$ )下根据标准JEDEC保持期限规格(A117)对器件进行测试。作为此测试程序的一部分，Flash/EE存储器循环工作的次数到达指定的耐久性限值，然后确定数据保持特性。

这表明每次对Flash/EE存储器进行重新编程时，都保证Flash/EE存储器在完全指定的保持期限内保持数据。注意：基于一个0.6 eV激活能的保持期限随 $T_j$ 递减，如图30所示。

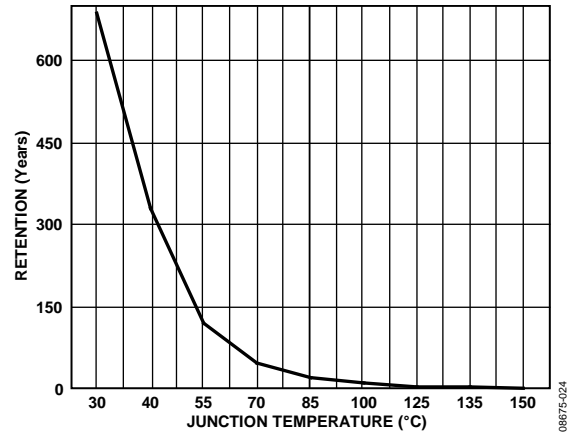


图30. Flash/EE存储器数据保持期限

### 编程

在串行下载模式或JTAG模式下，可对Flash/EE存储器中的62 kB区域进行在线编程。

#### 通过I<sup>2</sup>C执行下载(在线编程)

ADuC7023可利用I<sup>2</sup>C端口下载代码。如果BM引脚通过一个1 kΩ的外部电阻下拉到低电平且闪存地址0x80014 = 0xFFFFFFFF，ADuC7023就会在复位或上电周期后进入到下载模式。一旦进入下载模式，当器件在其目标应用硬件为在线状态时，用户就可以下载代码到Flash/EE的全部62 kB中。开发系统提供PC I<sup>2</sup>C下载功能，支持通过I<sup>2</sup>C进行串行下载。ADI公司提供USB转I<sup>2</sup>C下载适配板。此板可连接到PC的USB端口和ADuC7023的I<sup>2</sup>C端口。产品型号为USB-I2C/LIN-CONV-Z。

[AN-806](#)应用笔记详细介绍了通过I<sup>2</sup>C执行串行下载的协议。

#### JTAG访问

JTAG协议允许借助片内JTAG接口进行代码下载和调试。

如果要通过JTAG接口访问器件，必须将P0.0/BM引脚设为高电平，以使能P0.1/P0.2/P0.3作为JTAG引脚。

调试时，不应将用户代码写入P0.1/P0.2和P0.3引脚。如果用户切换这些引脚中的任意一个，JTAG调试器将无法连接到ADuC7023。如果发生这种情况，用户应确保擦除闪存地址0x80014，以允许通过I<sup>2</sup>C接口擦除器件。

## 安全性

提供给用户的62 kB Flash/EE存储区域实现了读写保护。

通过配置FEEPRO/FEEHIDE寄存器(参见表34)的第31位,可以保护这62 kB,使其不能通过JTAG编程模式来读取;该寄存器的其他31位可以保护闪存不被写入。每位保护4页,即2 kB。写保护对所有类型的访问都有效。

## 三种级别保护

可通过直接写入FEEHIDE寄存器来设定或取消保护。复位后,保护失效。

可通过写入FEEPRO寄存器来设定保护。只有在一个保存保护命令(0x0C)和复位以后才生效。FEEPRO寄存器采用一个密钥保护来避免直接访问。一旦密钥被保存,必须再次输入此密钥才可更改FEEPRO。批量擦除可以将密钥设回0xFFFF,但同时也会擦除所有的用户代码。

通过FEEPRO寄存器和一个特定密钥值(0xDEADDEAD)可以永久保护闪存中的数据。此时即使再次输入密钥也不允许更改FEEPRO寄存器。

## 密钥写入时序

1. 写入FEEPRO寄存器的某一位,相应页面会被保护;
2. 将FEEMOD寄存器的第6位置1(第5位必须为0),使能密钥保护;
3. 将一个32位密钥写入FEEADR和FEEDAT寄存器;
4. 在FEECON寄存器中运行写密钥命令0x0C,通过监测FEESTA等待读取成功;
5. 复位器件。

**表31. FEESTA寄存器位功能描述**

位	描述
7至6	保留。
5	保留。
4	保留。
3	闪存中断状态位。 中断发生后,即当一条命令执行完毕且FEEMOD寄存器中的Flash/EE中断使能位被置1时,该位自动置1。 读FEESTA寄存器时,该位清0。
2	Flash/EE控制器繁忙。 控制器繁忙时,该位自动置1。 控制器空闲时,该位自动清0。
1	命令失败。 命令未完成时,该位自动置1。 读FEESTA寄存器时,该位自动清0。
0	指令通过。 命令完成时, MicroConverter将该位置1。 读FEESTA寄存器时,该位自动清0。

要取消或修改保护,可使用相同的序列,同时修改FEEPRO值。如果密钥选择的值是0xDEAD,则无法取消存储器保护。只有批量擦除能够取消器件保护,但它也会擦除所有用户代码。

密钥写入时序如下所示(保护闪存的第4页到第7页):

```
FEEPRO=0xFFFFFFF; //Protect Page 4 to
                          Page 7
```

```
FEEMOD=0x48; //Write key enable
```

```
FEEADR=0x1234; //16 bit key value
```

```
FEEDAT=0x5678; //16 bit key value
```

```
FEECON= 0x0C; //Write key command
```

如果要永久保护器件,应该用同样的时序再写一次密钥,但要注意此时FEEADR = 0xDEAD且FEEDAT = 0xDEAD。

## Flash/EE控制接口

串行下载和JTAG编程采用了Flash/EE控制接口,其中包括本节所列的8个寄存器。

## FEESTA寄存器

名称: FEESTA

地址: 0xFFFFF800

默认值: 0x20

访问类型: 读取

功能: FEESTA是一个只读寄存器,它反映闪存控制接口的状态。如表31所示。



# ADuC7023

## FEEMOD寄存器

名称:	FEEMOD
地址:	0xFFFFF804
默认值:	0x0000
访问类型:	读/写
功能:	FEEMOD用来设置Flash/EE控制接口的工作模式。表32显示FEEMOD寄存器位功能描述。

**表32. FEEMOD寄存器位功能描述**

位	描述
15至9	保留。
8	保留。始终将该位置为0。
7至5	保留。始终将该位置为0(写密钥的情况除外)。参见“密钥写入时序”一节。
4	Flash/EE中断使能。 通过将该位置1, 用户可启用Flash/EE中断。当处理器执行完一条命令后, 将产生中断。 通过将该位清0, 用户可禁用Flash/EE中断。
3	擦/写命令保护。 通过将该位置1, 用户可启用擦/写命令。 通过将该位清0, 可禁用擦/写命令, 从而保护Flash/EE。
2至0	保留。始终将该位置为0。

## FEECON寄存器

名称:	FEECON
地址:	0xFFFFF808
默认值:	0x07
访问类型:	读/写
功能:	FEECON是一个8位命令寄存器。命令说明如表33所示。

**表33. FEECON中的命令代码**

代码	命令	描述
0x00 <sup>1</sup>	零点	空闲状态。
0x01 <sup>1</sup>	单次读取	向FEEDAT内载入16位数据。由FEEADR索引。
0x02 <sup>1</sup>	单次写入	将FEEDAT中的数据写入FEEADR所指的地址。此操作耗时50 μs。
0x03 <sup>1</sup>	擦/写	擦除由FEEADR索引的页面, 并且把FEEDAT中的数据写入FEEADR所指的存储区域。此项操作大约需要24ms。
0x04 <sup>1</sup>	单次验证	将FEEADR所指地址中的数据与FEEDAT中的数据进行比较, 比较的结果由FEESTA的第1位显示。
0x05 <sup>1</sup>	单次擦除	擦除由FEEADR索引的页面。
0x06 <sup>1</sup>	批量擦除	擦除62 kB的用户空间。而2KB的内核空间是被保护的。此项操作需要2.48 s。 为了防止误操作, 执行该命令需要一个命令序列。参见“批量擦除命令执行序列”部分。
0x07	保留	保留。
0x08	保留	保留。
0x09	保留	保留。
0x0A	保留	保留。
0x0B	签名	将64KB Flash/EE的一个签名写入24位FEESIGN寄存器中。此项操作需要32,778个时钟周期。
0x0C	保护	此命令仅可运行一次。只有批量擦除命令(0x06)或密钥(FEEADR/FEEDAT)可以保存或去除FEEPRO中的值。

代码	命令	描述
0x0D	保留	保留。
0x0E	保留	保留。
0x0F	Ping	无操作；产生中断。

<sup>1</sup>在执行完这些命令后，FEECON寄存器总是立即读取0x07。

#### FEEDAT寄存器

名称： FEEDAT  
地址： 0xFFFFF80C  
默认值： 0XXXXX  
访问类型： 读/写  
功能： FEEDAT是一个16位数据寄存器。

#### FEEADR寄存器

名称： FEEADR  
地址： 0xFFFFF810  
默认值： 0x0000  
访问类型： 读/写  
功能： FEEADR是另一个16位地址寄存器。

#### FEESIGN寄存器

名称： FEESIGN  
地址： 0xFFFFF818  
默认值： 0FFFFFFF  
访问类型： 读取  
功能： FEESIGN为24位编码签名。

#### FEEPRO寄存器

名称： FEEPRO  
地址： 0xFFFFF81C  
默认值： 0x00000000  
访问类型： 读/写  
功能： FEEPRO寄存器在寄存器复位后提供保护。它需要一个软件密钥(见表34)。

#### FEEHIDE寄存器

名称： FEEHIDE  
地址： 0xFFFFF820  
默认值： 0xFFFFFFFF  
访问类型： 读/写  
功能： FEEHIDE寄存器提供立即保护。它不需要任何软件密钥。复位后，FEEHIDE中的保护设置将清0(见表34)。

**表34. FEEPRO和FEEHIDE寄存器位功能描述**

位	描述
31	读保护位 通过将该位清0，用户可保护代码。 通过将该位置1，用户可允许读取代码。
30至0	第123页到第120页、第119页到第116页和第0页到第3页的写保护。 通过将该位清0，用户可为这些页面提供写保护。 通过将该位置1，用户可允许写入这些页面。

#### 批量擦除命令执行序列

```
FEEDAT = 0x3CFF;
FEEADR = 0xFFC3;
FEEMOD = FEEMOD|0x8;           //Erase key enable
FEECON = 0x06;                 //Mass erase
command
```

## SRAM和Flash/EE执行时间

### SRAM执行时间

因为一个最小时钟周期为22 ns，而访问SRAM所需的时间为2 ns，所以从SRAM中取指令只需一个时钟周期。尽管如此，当数据在SRAM中时，如果指令中包括读取存储器指令或写入存储器指令，则需要增加一个时钟周期，当数据在Flash/EE中时，需要增加三个时钟周期；一个周期用来执行指令，两个周期来从Flash/EE中获取32位数据。取一个控制流指令(例如分支指令)需要一个时钟周期，但同时要花两个时钟周期取新指令来填充流水线。

### Flash/EE的执行时间

因为Flash/EE为16位宽度，而读取16位字的时间为22 ns，所以在Flash/EE中执行指令无法在1个周期内完成(当CD位=0时，SRAM中执行指令就能在一个周期内完成)。并且，无论CD位为何值，在访问数据之前还存在死区时间。

在ARM模式下，指令是32位的，当CD=0时，需要2个时钟周期来读取一个指令；在Thumb模式下，指令是16位的，读取任何指令都只需要1个时钟周期。

当所执行的指令中包含使用Flash/EE进行数据存储的指令时，这两种模式下所用的时间相同。如果指令为控制流指令，除了需要1个额外的时钟周期来解码程序计数器的新地址外，还需要4个时钟周期来填充流水线；如果1个数据处理指令只在内核寄存器中，则不需要任何额外的时钟周期，但如果所执行的数据在Flash/EE中，需要用1个时钟周期来解码数据的地址，然后用2个时钟周期来从Flash/EE中读取32位的数据。在取另一个指令之前，需要外加一个时钟周期。对于数据传输指令，执行时间更复杂，具体如表35所示。

表35. ARM/Thumb模式下指令的执行周期

指令	读取周期	死区时间	数据访问	死区时间
LD <sup>1</sup>	2/1	1	2	1
LDH	2/1	1	1	1
LDM/PUSH	2/1	N <sup>2</sup>	2 × N <sup>2</sup>	N <sup>1</sup>
STR <sup>1</sup>	2/1	1	2 × 20 ns	1
STRH	2/1	1	20 ns	1
STRM/POP	2/1	N <sup>1</sup>	2 × N × 20 ns <sup>1</sup>	N <sup>1</sup>

<sup>1</sup>一条SWAP指令包含LD和STR这两条指令，而且只读取一次，共需八个时钟周期+40ns。

<sup>2</sup>N代表使用多重装载/存储指令时，装载或存储数据的个数(1 < N ≤ 16)。

## 复位和重映射

ARM异常矢量全部位于存储器阵列的底部，从地址0x00000000到地址0x00000020，如图31所示。

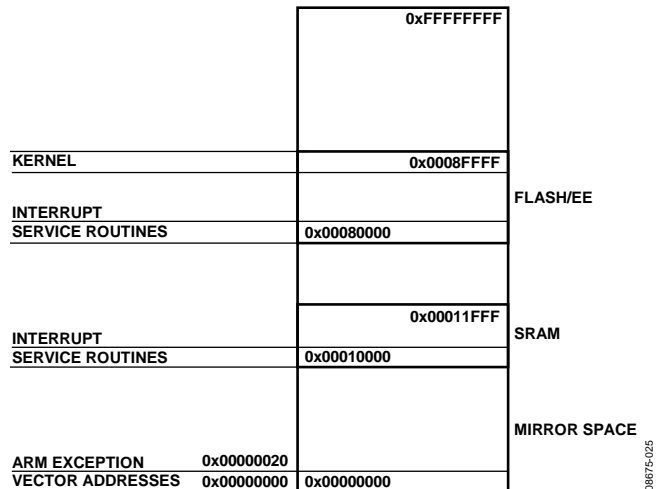


图31. 异常重映射

默认情况下，复位之后，Flash/EE会被镜像到存储器阵列的底部。凭借重映射功能，程序员可将SRAM镜像到存储器阵列的底部，便于从SRAM而不是从Flash/EE执行异常程序。这是因为在32位ARM模式下执行异常程序时，SRAM的带宽为32位，而Flash/EE带宽为16位，所以在SRAM中执行异常程序的速度要快两倍。

### 重映射操作

ADuC7023发生复位时，会自动执行工厂内置的程序代码。该内核是隐藏的，用户代码无法访问。如果器件在正常模式工作(BM引脚为高电平)，先执行内核的上电配置程序，并跳转到复位矢量地址0x00000000，然后执行用户的复位异常程序。

因为复位后，Flash/EE被镜像到存储器阵列的底部，所以复位中断程序一定要写在Flash/EE中。

通过将Remap寄存器的0位置1，将从Flash/EE执行重映射。必须注意从Flash/EE的地址0x00080020上方开始执行此命令，而不能从阵列底部开始执行，因为这部分已被SRAM所取代。

这个操作是可逆的。通过将Remap寄存器的0位清0，Flash/EE可以重映射到地址0x00000000。在镜像区域以外的地方执行映射操作时必须小心注意，任何形式的复位都会把Flash/EE存储器映射到存储器阵列的底部。



**REMAP寄存器**

名称： REMAP  
 地址： 0xFFFF0220  
 默认值： 0x00  
 访问类型： 读/写

**表36. REMAP寄存器位功能描述**

位	名称：	描述
7至5		保留。
4		只读位。标明Flash/EE存储器可用空间的大小。如果该位置1，表示Flash/EE中只有32KB空间可用。
3		只读位。标明SRAM存储器可用空间的大小。如果该位置1，表示SRAM中只有4KB空间可用。
2至1	JTAFO	只读位。详情见P0.0/BM描述。 如果等于[00]，则P0.1/P0.2/P0.3配置为JTAG引脚。 如果等于[1x]，则P0.1/P0.2/P0.3配置为GPIO引脚。 这些位均由内核在任意复位序列后进行配置，并取决于最近复位序列中的P0.0状态。
0	重映射	重映射位。 通过将该位置1，用户可将SRAM重映射至地址0x00000000。 复位后该位自动清0，以重映射Flash/EE到地址0x00000000。

**复位操作**

一共有四种类型的复位：外部复位、上电复位、看门狗复位和软件强制复位。RSTSTA寄存器会指示最近复位的源类型，RSTCLA则可将RSTSTA寄存器清空。在复位异常服务程序执行时，可以使用这两种寄存器来识别复位源。如果RSTSTA为空，则为外部复位。

RSTCFG寄存器允许不同外设在看门狗复位或软件复位后保持状态不变。

**RSTSTA寄存器**

名称： RSTSTA  
 地址： 0xFFFF0230  
 默认值： 0x01  
 访问类型： 读/写

**表37. RSTSTA寄存器位功能描述**

位	描述
7至3	保留。
2	软件复位。 通过将该位置1，用户可强制执行软件复位。 通过设定RSTCLR中的相应位，可将该位清0。
1	看门狗超时。 看门狗超时，该位自动置1。 通过设定RSTCLR中的相应位，可将该位清0。
0	上电复位。 发生上电复位时，该位自动置1。 通过设定RSTCLR中的相应位，可将该位清0。

**RSTCLR寄存器**

名称： RSTCLR  
 地址： 0xFFFF0234  
 默认值： 0x00  
 访问类型： 写入  
 功能： 请注意，清空RSTSTA寄存器时，用户必须将0x07写入RSTCLR寄存器。

**RSTCFG寄存器**

名称： RSTCFG  
 地址： 0xFFFF024C  
 默认值： 0x00  
 访问类型： 读/写

**表38. RSTCFG寄存器位功能描述**

位	描述
7至3	保留。始终置为0。
2	通过将该位置1，可配置DAC输出在看门狗复位或软件复位后保持状态不变。 通过将该位清0，DAC引脚和寄存器即会返回其默认状态。
1	保留。始终置为0。
0	通过将该位置1，可配置GPIO引脚在看门狗复位或软件复位后保持状态不变。 通过将该位清0，GPIO引脚和寄存器即会返回其默认状态。

# ADuC7023

## RSTKEY1寄存器

名称: RSTKEY1  
地址: 0xFFFF0248  
默认值: 0xXX  
访问类型: 写入

## RSTKEY2寄存器

名称: RSTKEY2  
地址: 0xFFFF0250  
默认值: 0xXX  
访问类型: 写入

表39. RSTCFG写序列

名称	代码
RSTKEY1	0x76
RSTCFG	用户设定值
RSTKEY2	0xB1

## 其他模拟外设

### DAC

ADuC7023片内集成四个12位电压输出DAC。每个DAC都有一个轨到轨电压输出缓冲器，驱动能力为5 k $\Omega$ /100 pF。

每个DAC都有两个可选电压输出范围：0 V至 $V_{REF}$ （内部带隙2.5 V基准电压源）和0 V至 $AV_{DD}$ 。

信号范围为0 V至 $AV_{DD}$ 。

通过将RSTCFG的第2位置1，DAC输出引脚可在看门狗复位或软件复位期间保持状态不变。

### 寄存器接口

每个DAC都可通过一个控制寄存器和一个数据寄存器独立配置。对于4个DAC来说，这两个寄存器是相同的。这一节我们仅介绍DAC0CON(参见表40)和DAC0DAT(参见表41)的具体功能。

### DACxCON寄存器

名称	地址	默认值	访问类型
DAC0CON	0xFFFF0600	0x00	R/W
DAC1CON	0xFFFF0608	0x00	R/W
DAC2CON	0xFFFF0610	0x00	R/W
DAC3CON	0xFFFF0618	0x00	R/W

表40. DAC0CON寄存器位功能描述

位	值	名称	描述
7			保留。
6		DACBY	通过将该位置1，可旁路DAC输出缓冲器。 通过将该位清0，可使能DAC输出缓冲器。
5		DACCLK	DAC更新速率。 通过将该位置1，用户可使用定时器1来更新DAC。 通过将该位清0，用户可使用HCLK(内核时钟)来更新DAC。
4		DACCLR	DAC清除位。 通过将该位置1，用户可使能正常DAC操作。 通过将该位清0，用户可将DAC的数据寄存器复位至0。
3			保留。该位保持为0。
2			保留。该位保持为0。
1至0			DAC范围位。 00 关断模式。DAC输出为三态。 01 保留。 10 0至 $V_{REF}$ (2.5 V)范围。 11 0至 $AV_{DD}$ 范围。

### DACxDAT寄存器

名称	地址	默认值	访问类型
DAC0DAT	0xFFFF0604	0x00000000	R/W
DAC1DAT	0xFFFF060C	0x00000000	R/W
DAC2DAT	0xFFFF0614	0x00000000	R/W
DAC3DAT	0xFFFF061C	0x00000000	R/W

表41. DAC0DAT寄存器位功能描述

位	描述
31至28	保留。
27至16	DAC0的12位数据。
15至0	保留。

### DAC的使用

片内DAC架构由一电阻串DAC和一个输出缓冲放大器构成。功能等效框图如图32所示。

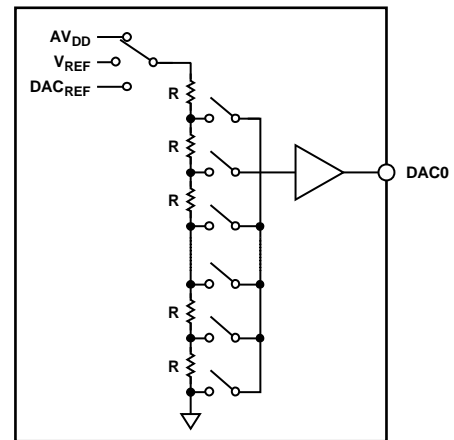


图32. DAC结构

如图32所示，用户可在软件中选择各DAC的基准电压源。基准电压源可为 $AV_{DD}$ 或 $V_{REF}$ 。在0至 $AV_{DD}$ 模式中，DAC输出传递函数范围为0 V至 $AV_{DD}$ 引脚电压；在0至 $V_{REF}$ 模式中，DAC输出传递函数范围为0至2.5 V内部基准电压 $V_{REF}$ 。

每个DAC输出缓冲放大器都有一个真轨到轨输出级。也就是说，当输出空载时，DAC输出摆幅能够达到 $AV_{DD}$ 或地电平的5 mV范围以内。此外，当驱动一个5 k $\Omega$ 阻性负载到地时，除了代码0至100(在0至 $AV_{DD}$ 模式中为代码3995至4095)外，整个传递函数都能保证符合DAC线性度规格要求。

地和 $V_{DD}$ 附近的线性度下降是由输出放大器的饱和引起的，图33反映了这种效应的一般表现(失调和增益误差忽略不计)。图33中的虚线为理想的传递函数，实线代表转换可能具有端点非线性(由输出放大器饱和引起的)的传递函数。图33仅代表0至 $AV_{DD}$ 模式下的传递函数。在0至 $V_{REF}$ 模式( $V_{REF} < AV_{DD}$ )下，下半部分的非线性度是相似的。然而，传递函数的上半部分一直到端点都表现为理想的线性(这里采用 $V_{REF}$ ，而非 $AV_{DD}$ )，这说明DAC输出没有端点线性误差。

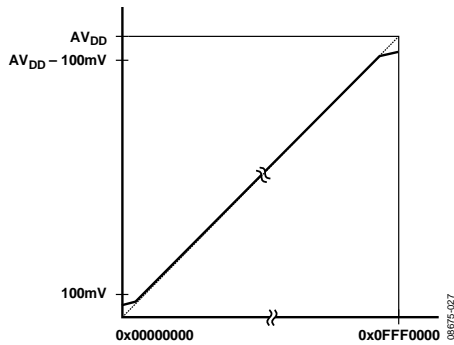


图33. 放大器饱和引起的端点非线性

当有输出负载时，图33中的端点非线性会变得更差。ADuC7023数据手册中的绝大多数技术参数都是在DAC输出端接有一个接地的5 kΩ阻性负载的条件下得到的。由于输出被强制提供更多的源电流或吸电流，图33中的顶部或底部非线性区域将增大。而当需要更大电流时，这会明显地限制输出电压摆幅。

## ADC和DAC的基准电压源

ADC和DAC均可配置为使用内部 $V_{REF}$ 或外部基准电压源作为基准源。内部 $V_{REF}$ 必须与外部0.47 μF电容搭配使用。

表42. ADC和DAC的基准电压源选择

REFCON位0	DACxCON[1:0]	描述
0	00	ADC采用外部基准电压源。DAC关断。
0	01	保留。
0	10	保留。
0	11	ADC采用外部基准电压源。DAC采用内部 $AV_{DD}$ 。
1	00	ADC采用内部 $V_{REF}$ 。DAC关断。
1	01	ADC和DAC均采用外部基准电压源。外部基准电压源必须具有足够强的驱动能力以抑制内部基准电压源。
1	10	ADC和DAC均采用内部 $V_{REF}$ 。
1	11	ADC采用内部 $V_{REF}$ 。DAC采用内部 $AV_{DD}$ 。

## 在运算放大器模式下配置DAC缓冲器

在运算放大器模式下，DAC输出缓冲器用作运算放大器，而DAC本身禁用。

如果DACBCFG位0将该位置1，则ADC0是运算放大器的正输入端，ADC1是负输入端，而DAC0是输出端。在该模式下，应通过将DAC0CON的位0和位1清0来将DAC关断。

如果DACBCFG位1将该位置1，则ADC2是运算放大器的正输入端，ADC3是负输入端，而DAC1是输出端。在该模式下，应通过将DAC1CON的位0和位1清0来将DAC关断。

如果DACBCFG位2将该位置1，则ADC4是运算放大器的正输入端，ADC5是负输入端，而DAC2是输出端。在该模式下，应通过将DAC2CON的位0和位1清0来将DAC关断。

如果DACBCFG位3将该位置1，则ADC8是运算放大器的正输入端，ADC9是负输入端，而DAC3是输出端。在该模式下，应通过将DAC3CON的位0和位1清0来将DAC关断。

## DACBCFG寄存器

名称：	DACBCFG
地址：	0xFFFF0654
默认值：	0x00
访问类型：	读/写

表43. DACBCFG寄存器位功能描述

位	描述
7至4	保留。始终置为0。
3	通过将该位置1，可配置DAC3输出缓冲器在运算放大器模式下工作。通过将该位清0，可让该DAC缓冲器在正常模式下工作。
2	通过将该位置1，可配置DAC2输出缓冲器在运算放大器模式下工作。通过将该位清0，可让该DAC缓冲器在正常模式下工作。
1	通过将该位置1，可配置DAC1输出缓冲器在运算放大器模式下工作。通过将该位清0，可让该DAC缓冲器在正常模式下工作。
0	通过将该位置1，可配置DAC0输出缓冲器在运算放大器模式下工作。通过将该位清0，可让该DAC缓冲器在正常模式下工作。

**DACBKEY0寄存器**

名称:	DACBKEY0
地址:	0xFFFF0650
默认值:	0x0000
访问类型:	写入

**DACBKEY1寄存器**

名称:	DACBKEY1
地址:	0xFFFF0658
默认值:	0x0000
访问类型:	写入

**表44. DACBCFG写序列**

名称	代码
DACBKEY0	0x9A
DACBCFG	用户设定值
DACBKEY1	0x0C

**电源监控器**

ADuC7023电源监控器负责调节片上IOV<sub>DD</sub>电压，当IOV<sub>DD</sub>引脚电压降低到一个电源跳变点时就会给出提示。监控功能是通过PSMCON寄存器来控制的。在IRQEN或FIQEN寄存器中使能后，监控器使用PSMCON寄存器的PSMI位来中断内核。而一旦CMP恢复到高电平，该位会立即被清0。

监控功能可以使用户保存当前工作寄存器中的数据，避免由于电压不足或断电造成的数据丢失；它也可以确保直到恢复安全电源时，代码正常重新执行。

**PSWCON寄存器**

名称:	PSMCON
地址:	0xFFFF0440
默认值:	0x0008
访问类型:	读/写

**表45. PSMCON寄存器位功能描述**

位	名称	描述
3	CMP	比较器位。只读位，可直接反映比较器状态。该位为1表示IOV <sub>DD</sub> 电压高于所选跳变点或PSM处于掉电模式；该位为0表示IOV <sub>DD</sub> 电压低于所选跳变点。在退出中断服务程序之前，该位应置1。
2	TP	跳变点选择位。 0 = 2.79 V. 1 = 保留。
1	PSMEN	电源监控器使能位。 通过将该位置1，可启用电源监控器电路。 通过将该位清0，可禁用电源监控器电路。
0	PSMI	电源监控器中断位。一旦CMP变为低电平，该位会被MicroConverter置1，表明I/O电源电压偏低。PSMI位可用来中断处理器。一旦CMP变为高电平，通过对该位写1来清空PSMI位。写0则不影响PSMI位；由于没有时间延迟，所以一旦CMP变为高电平，PSMI位可以立即被清0。

**比较器**

ADuC7023集成了电压比较器。比较器的正输入端与ADC2引脚复用，而负输入端则有两个：ADC3或DAC0。通过配置，电压比较器的输出可以产生系统中断、可以直接路由至可编程逻辑阵列、可以启动ADC转换或输出到外部引脚CMP<sub>OUT</sub>上，如图34所示。

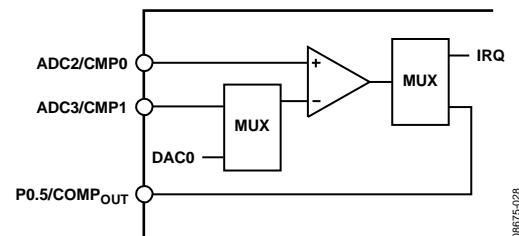


图34. 比较器

**迟滞**

图35描述了输入失调电压和迟滞的定义方式。其中，输入失调电压( $V_{OS}$ )为迟滞范围中心对地的差值。它可以是正，也可以是负；迟滞电压( $V_H$ )为迟滞范围 $\frac{1}{2}$ 的宽度。

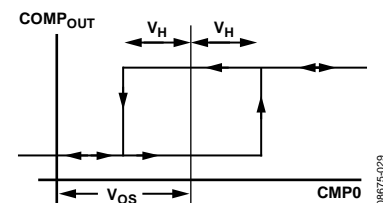


图35. 比较器迟滞传递函数

# ADuC7023

## 比较器接口

比较器接口由一个16位寄存器CMPCON组成，如表46所示。

### CMPCON寄存器

名称:	CMPCON
地址:	0xFFFF0444
默认值:	0x0000
访问类型:	读/写

表46. CMPCON寄存器位功能描述

位	值	名称	描述
15至11			保留。
10		CMPEN	比较器使能位。 通过将该位置1，用户可启用比较器。 通过将该位清0，用户可禁用比较器。
9至8	00 01 10 11	CMPIN	比较器负输入选择位。 $AV_{DD}/2$ 。 ADC3输入。 DAC0输出。 保留。
7至6	00 01 10 11	CMPOC	比较器输出配置位。 保留。 保留。 COMP <sub>OUT</sub> 上的输出。 IRQ。
5		CMPOL	比较器输出逻辑状态位。该位清0时，如果正输入(CMP0)高于负输入(CMP1)，比较器输出为高电平；该位置1时，如果正输入低于负输入，比较器输出为高电平。
4至3	00 11 01/10	CMPRES	响应时间。 对于大信号(2.5V差分)，响应时间典型值为5 $\mu$ s。 对于小信号(0.65V差分)，响应时间典型值为17 $\mu$ s。 典型值为3 $\mu$ s。 保留。
2		CMPHYST	比较器迟滞位。 通过将该位置1，用户可拥有约7.5 mV的迟滞。 通过将该位清0，用户可实现无迟滞。
1		CMPORI	比较器输出上升沿中断。 受监控的电压(CMP0)出现上升沿时，该位自动置1； 对该位写1可清0。
0		CMPOFI	比较器输出上升沿中断。 受监控的电压(CMP0)出现下降沿时，该位自动置1； 用户可将该位清0。

## 振荡器和锁相环—电源控制 时钟系统

ADuC7023集成有一个32.768 kHz  $\pm 3\%$ 振荡器、一个时钟分频器和一个锁相环。PLL锁定在内部振荡器的倍数(1275)上或外部32.768 KHz晶振，为系统产生一个稳定的41.78 MHz时钟(UCLK)。为了省电，内核可以工作在该频率或该频率的二进制约数上，实际的内核工作频率UCLK/2CD为HCLK。默认的内核时钟为PLL时钟的8分频(CD = 3)或5.22 MHz。内核时钟频率也可以来自ECLK引脚上的外部时钟，如图36所示。

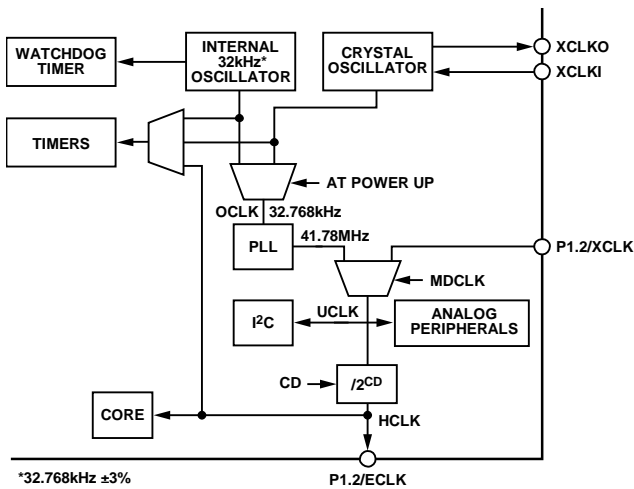


图36. 时钟系统

表47. 工作模式

模式	内核	外设	PLL	XTAL/T2/T3	IRQ0至IRQ3	启动/上电时间
有效	是	X	X	X	X	66 ms (CD = 0)
暂停		X	X	X	X	230 ns (CD = 0); 3 $\mu$ s (CD = 7)
浅休眠			X	X	X	283 ns (CD = 0); 3 $\mu$ s (CD = 7)
休眠				X	X	1.23 ms
停止					X	1.45 ms

X = 无关。

表48. 25°C时典型功耗值(单位: mA)

PC[2:0]	模式	CD = 0	CD = 1	CD = 2	CD = 3	CD = 4	CD = 5	CD = 6	CD = 7
000	有效	28	17	12	11	9.3	7.5	7.2	7
001	暂停	14	9	7.6	5.7	4.8	4.6	4.6	4.6
010	浅休眠	5	4.5	4.5	4.5	4.5	4.5	4.5	4.5
011	休眠	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23
100	停止	0.23	0.23	0.23	0.23	0.23	0.23	0.23	0.23

时钟源的选择是由PLLCON寄存器控制的，默认情况下选用内部振荡器作为PLL的输入。

但如果环境中存在噪声，噪声便会从外部晶振引脚耦合进入，使PLL快速失锁。中断控制器就会出现PLL中断。内核时钟立即挂起，只有在锁相恢复后才能处理该中断。

如果发生晶振失振，应使用看门狗定时器。一般在初始化期间，应对RSTSTA进行测试，判断有没有来自看门狗定时器的复位信号。

## 电源控制系统

ADuC7023支持多种工作模式选项。表47给出了不同模式下各器件的上电情况和上电时间。

表48给出了不同模式下的总功耗(模拟+数字电源电流)在不同时钟分频器位下的一些典型值。ADC关闭。请注意，这些值也包括测量时稳压器和测试板上其他一些器件的功耗。



# ADuC7023

## 寄存器和密钥

工作模式、时钟模式和可编程时钟分频器可通过PLLCON(参见表49)和POWCONx三个寄存器控制。PLLCON控制时钟系统的工作模式，POWCON0控制内核时钟频率和掉电模式，而POWCON1则控制I<sup>2</sup>C和SPI的时钟频率。

为了防止意外编程，写入PLLCON和POWCONx寄存器时需要遵循特定的时序。

### PLLKEY1寄存器

名称： PLLKEY1  
地址： 0xFFFF0410  
默认值： 0XXXXX  
访问类型： 写入

### PLLKEY2寄存器

名称： PLLKEY2  
地址： 0xFFFF0418  
默认值： 0XXXXX  
访问类型： 写入

### PLLCON寄存器

名称： PLLCON  
地址： 0xFFFF0414  
默认值： 0x21  
访问类型： 读/写

**表49. PLLCON MMR位分配**

位	值	名称	描述
7至6			保留。
5		OSEL	32 kHz PLL输入选择。通过将该位置1，用户可选择内部32 kHz振荡器。该位的默认值为1。通过将该位清0，用户可选择外部32 kHz晶振。
4至2			保留。
1至0	00 01 10 11	MDCLK	时钟模式。 保留。 PLL默认配置。 保留。 引脚33(40引脚LFCSP)/引脚25(32引脚LFCSP)上的外部时钟。

**表50. PLLCON写序列**

名称	代码
PLLKEY1	0xAA
PLLCON	用户设定值
PLLKEY2	0x55

### POWKEY1寄存器

名称： POWKEY1  
地址： 0xFFFF0404  
默认值： 0XXXXX  
访问类型： 写入  
功能： POWKEY1可防止POWCON0意外编程。

### POWKEY2寄存器

名称： POWKEY2  
地址： 0xFFFF040C  
默认值： 0XXXXX  
访问类型： 写入  
功能： POWKEY2可防止POWCON0意外编程。

### POWCON0寄存器

名称： POWCON0  
地址： 0xFFFF0408  
默认值： 0x00  
访问类型： 读/写

**表51. POWCON0寄存器位功能描述**

位	值	名称	描述
7			保留。
6至4	000 001 010 011 100 其它	PC	工作模式。 活动模式。 暂停模式。 浅休眠。 休眠模式。IRQ0至IRQ3可以唤醒该器件。 停止模式。IRQ0至IRQ3可以唤醒该器件。 保留。
3			保留。

位	值	名称	描述
2至0		CD	CPU时钟分频器位。
	000		41.78 MHz.
	001		20.89 MHz.
	010		10.44 MHz.
	011		5.22 MHz.
	100		2.61 MHz.
	101		1.31 MHz.
	110		653 kHz.
	111		326 kHz.

表52. POWCON0写序列

名称	代码
POWKEY1	0x01
POWCON0	用户设定值
POWKEY2	0xF4

**POWKEY3寄存器**

名称： POWKEY3

地址： 0xFFFF0434

默认值： 0xXXXX

访问类型： 写入

功能： POWKEY3可防止POWCON1意外编程。

**POWKEY4寄存器**

名称： POWKEY4

地址： 0xFFFF043C

默认值： 0xXXXX

访问类型： 写入

功能： POWKEY4可防止POWCON1意外编程。

**POWCON1寄存器**

名称： POWCON1

地址： 0xFFFF0438

默认值： 0x0004

访问类型： 读/写

表53. POWCON1寄存器位功能描述

位	值	名称	描述
15至9			保留。
8		SPIPO	通过将该位清0，可关断SPI。
7至6		SPICLKDIV	SPI模块驱动时钟分频器位。
	00		41.78 MHz.
	01		20.89 MHz.
	10		10.44 MHz.
	11		5.22 MHz.
5		I2C1PO	通过将该位清0，可关断I <sup>2</sup> C1。
4至3		I2C1CLKDIV	I <sup>2</sup> C0模块驱动时钟分频器位。
	00		41.78 MHz.
	01		10.44 MHz.
	10		5.22 MHz.
	11		1.31 MHz.
2		I2C0PO	通过将该位清0，可关断I <sup>2</sup> C0。
1至0		I2C0CLKDIV	I <sup>2</sup> C1模块驱动时钟分频器位。
	00		41.78 MHz.
	01		10.44 MHz.
	10		5.22 MHz.
	11		1.31 MHz.

<sup>1</sup> SPI/I2C0/I2C1的分频时钟必须大于或等于通过POWCON0 [2:0]选择的CPU时钟。

表54. POWCON1写序列

名称	代码
POWKEY3	0x76
POWCON1	用户设定值
POWKEY4	0XB1

## 数字外设

### 通用输入/输出

ADuC7023共有20个双向通用输入/输出(GPIO)引脚。所有I/O引脚都兼容5 V电压,即GPIO支持5 V输入电压。一般来说,GPIO引脚都有多种功能(表55给出了各引脚功能定义)。默认情况下,GPIO引脚在GPIO模式下工作。

每个GPIO引脚都有一个内部上拉电阻(约为100 kΩ),驱动能力为1.6 mA。注意,最多可以有20个驱动1.6 mA电流的GPIO引脚同时工作。通过GPxPAR寄存器,可控制是否使用内部上拉电阻。

20个GPIO引脚被分成3个端口,即端口0到端口2。每个端口由4或5个寄存器控制。

GPIO的输入电平在任何时间都能从GPxDAT寄存器中读出,甚至当引脚在除GPIO模式以外的模式下被重新配置时也可以。PLA输入始终有效。

ADuC7023器件进入省电模式后,GPIO引脚维持各自原来的状态。另外注意,通过将RSTCFG位0置1,GPIO引脚可在看门狗复位或软件复位过程中保持状态不变。

**表55. GPIO引脚功能描述**

端口	引脚	配置			
		00	01	10	11
0	P0.0	GPIO/BM	nTRST	ADC <sub>BUSY</sub>	PLAI[8]
	P0.1 <sup>1</sup>	GPIO	TDO		PLAI[9]
	P0.2 <sup>1</sup>	GPIO	TDI		PLAO[8]
	P0.3 <sup>1</sup>	GPIO	TCK		PLAO[9]
	P0.4	GPIO/IRQ0	SCL0	CONV <sub>START</sub>	PLAI[0]
	P0.5	GPIO	SDA0	COMP <sub>OUT</sub>	PLAI[1]
	P0.6	GPIO	MISO	SCL1 <sup>2</sup>	PLAI[2]
	P0.7	GPIO	MOSI	SDA1 <sup>2</sup>	PLAO[0]
1	P1.0	GPIO	SCLK	PWM0	PLAO[1]
	P1.1	GPIO/IRQ1	SS	PWM1	PLAO[2]
	P1.2 <sup>3</sup>	GPIO/IRQ2	ADC4	ECLK	PLAI[3]
	P1.3	GPIO/IRQ3	ADC5		PLAI[4]
	P1.4	GPIO	ADC10		PLAO[3]
	P1.5	GPIO	ADC6	PWM <sub>TRIPINPUT</sub>	PLAO[4]
	P1.6	GPIO	SCL1 <sup>4</sup>	PWM2	PLAI[5]
	P1.7	GPIO	SDA1 <sup>4</sup>	PWM3	PLAI[6]
2	P2.0	GPIO	ADC12	PWM4	PLAI[7]
	P2.2	GPIO	ADC7	PWM <sub>sync</sub>	PLAO[6]
	P2.3	GPIO	ADC8		PLAO[7]
	P2.4	GPIO	ADC9		PLAI[10]

<sup>1</sup>通过JTAG调试器件时,不应通过用户代码使用这些引脚。有关如何配置这些引脚在GPIO模式下工作的更多详细信息,参见表36。这些引脚的默认值取决于在最近一次复位序列中P0.0/BM引脚的电平。

<sup>2</sup>I<sup>2</sup>C1功能仅适用于32引脚封装。

<sup>3</sup>当配置为模式2时,P1.2默认为ECLK,或者为内核时钟输出。如果将其配置为时钟输入,PLLCON的MDCLK位必须置为11。

<sup>4</sup>I<sup>2</sup>C1功能仅适用于40引脚封装。

### GPxCON寄存器

名称	地址	默认值	访问类型
GP0CON	0xFFFFF400	0x00001111	R/W
GP1CON	0xFFFFF404	0x00000000	R/W
GP2CON	0xFFFFF408	0x00000000	R/W

GPxCON是端口x的控制寄存器,它可以决定端口x中每个引脚的功能。引脚功能如表56所示。

**表56. GPxCON寄存器位功能描述**

位	描述
31至30	保留。
29至28	Px.7引脚的功能选择。
27至26	保留。
25至24	Px.6引脚的功能选择。
23至22	保留。
21至20	Px.5引脚的功能选择。
19至18	保留。
17至16	Px.4引脚的功能选择。
15至14	保留。
13至12	Px.3引脚的功能选择。
11至10	保留。
9至8	Px.2引脚的功能选择。
7至6	保留。
5至4	Px.1引脚的功能选择。
3至2	保留。
1至0	Px.0引脚的功能选择。

### GP0PAR寄存器

名称: GP0PAR

地址: 0xFFFFF42C

默认值: 0x22220000

访问类型: 读/写

功能: 通过对GP0PAR编程,可分别控制端口0、端口1和端口2的参数。需要注意的是,在对GP0DAT寄存器执行写操作前,必须先对GP0PAR寄存器执行写操作。

### GP1PAR寄存器

名称: GP1PAR

地址: 0xFFFFF43C

默认值: 0x22000022

访问类型: 读/写

功能: 通过对GP1PAR编程,可分别控制端口0、端口1和端口2的参数。需要注意的是,在对GP1DAT寄存器执行写操作前,必须先对GP1PAR寄存器执行写操作。

**GP2PAR寄存器**

名称: GP2PAR

地址: 0xFFFF44C

默认值: 0x00000000

访问类型: 读/写

功能: 通过对GP2PAR编程, 可分别控制端口0、端口1和端口2的参数。需要注意的是, 在对GP2DAT寄存器执行写操作前, 必须先对GP2PAR寄存器执行写操作。

**表57. GPxPAR寄存器位功能描述**

位	描述
31	保留。
30至29	Px.7驱动强度
28	Px.7上拉禁用。
27	保留。
26至26	Px.6驱动强度
24	Px.6上拉禁用。
23	保留。
22至21	Px.5驱动强度
20	Px.5上拉禁用。
19	保留。
18至17	Px.4驱动强度
16	Px.4上拉禁用。
15	保留。
14至13	Px.3驱动强度
12	Px.3上拉禁用。
11	保留。
10至9	Px.2驱动强度
8	Px.2上拉禁用。
7	保留。
6至5	Px.1驱动强度
4	Px.1上拉禁用。
3	保留。
2至1	Px.0驱动强度
0	Px.0上拉禁用。

**表58. GPIO驱动强度控制位功能描述**

控制位值	描述
00	中等驱动强度。
01	低驱动强度。
1x	高驱动强度。

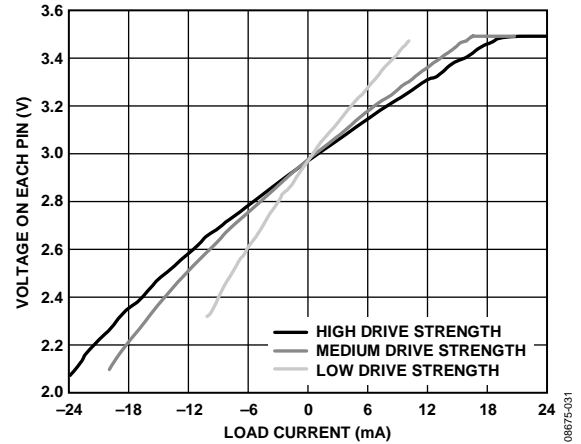


图37. 高电平的可编程强度

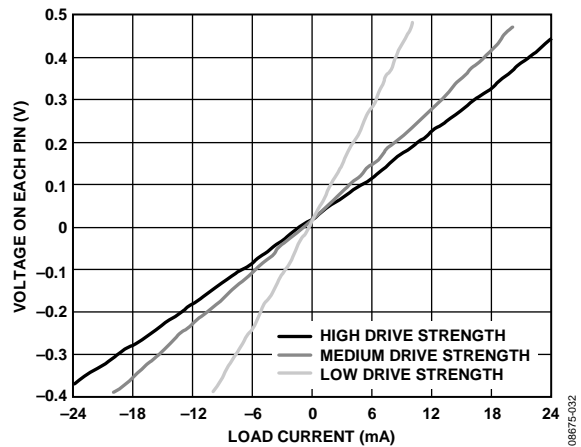


图38. 低电平的可编程强度

驱动强度位仅可在复位后写入一次。之后继续写入相关位时, 驱动强度并不会发生任何改变。对于GPIO端口, GPIO驱动强度和上拉禁用并非始终可调。有些控制位无法更改(参见表59)。

**表59. GPxPAR控制位访问描述<sup>1</sup>**

位	GP0PAR	GP1PAR	GP2PAR
31	保留	保留	保留
30至29	R/W	R/W	保留
28	R/W	R/W	保留
27	保留	保留	保留
26至26	R/W	R/W	保留
24	R/W	R/W	保留
23	保留	保留	保留
22至21	R/W	R (b00)	保留
20	R/W	R/W	保留
19	保留	保留	保留
18至17	R (b00)	R (b00)	R (b00)
16	R/W	R/W	R/W
15	保留	保留	保留
14至13	R (b00)	R (b00)	R (b00)
12	R/W	R/W	R/W
11	保留	保留	保留

# ADuC7023

位	GP0PAR	GP1PAR	GP2PAR
10至9	R (b00)	R (b00)	R (b00)
8	R/W	R/W	R/W
7	保留	保留	保留
6至5	R (b00)	R (b00)	保留
4	R/W	R/W	保留
3	保留	保留	保留
2至1	R (b00)	R (b00)	R (b00)
0	R/W	R/W	R (b0)

<sup>1</sup>当P2.0配置为AIN12时，无法禁用内部上拉电阻。

## GP0DAT寄存器

名称	地址	默认值	访问类型
GP0DAT	0xFFFFF420	0x000000XX	R/W
GP1DAT	0xFFFFF430	0x000000XX	R/W
GP2DAT	0xFFFFF440	0x000000XX	R/W

GPxDAT是端口x的配置和数据寄存器。它们用来配置端口x的GPIO引脚方向，为配置成输出的引脚设置输出值，并为配置成输入的引脚保存输入值。

**表60. GPxDAT寄存器位功能描述**

位	描述
31至24	数据传输方向。 通过将该位置1，用户可将GPIO引脚配置为输出引脚。 通过将该位清0，用户可将GPIO引脚配置为输入引脚。
23至16	端口x数据输出。
15至8	反映复位时端口x引脚的状态(只读)。
7至0	端口x数据输入(只读)。

## GP0SET寄存器

名称： GP0SET  
地址： 0xFFFFF424  
默认值： 0x000000XX  
访问类型： 写入  
功能： GP0SET是端口x的数据设置寄存器。

## GP1SET寄存器

名称： GP1SET  
地址： 0xFFFFF434  
默认值： 0x000000XX  
访问类型： 写入  
功能： GP1SET是端口x的数据设置寄存器。

## GP2SET寄存器

名称： GP2SET  
地址： 0xFFFFF444  
默认值： 0x000000XX  
访问类型： 写入  
功能： GP2SET是端口x的数据设置寄存器。

**表61. GPxSET寄存器位功能描述**

位	描述
31至24	保留。
23至16	数据端口x。 通过将该位置1，用户可将端口x的相应位置1；同时，还会将GPxDAT寄存器中的相应位置1。用户可将该位清0；该位并不影响数据输出。
15至0	保留。

## GP0CLR寄存器

名称： GP0CLR  
地址： 0xFFFFF428  
默认值： 0x000000XX  
访问类型： 写入  
功能： GP0CLR是端口x的数据清除寄存器。

## GP1CLR寄存器

名称： GP1CLR  
地址： 0xFFFFF438  
默认值： 0x000000XX  
访问类型： 写入  
功能： GP1CLR是端口x的数据清除寄存器。

## GP2CLR寄存器

名称： GP2CLR  
地址： 0xFFFFF448  
默认值： 0x000000XX  
访问类型： 写入  
功能： GP2CLR是端口x的数据清除寄存器。

表62. GPxCLR寄存器位功能描述

位	描述
31至24	保留。
23至16	数据端口x清除位。 通过将该位置1, 用户可将端口x的相应位清0; 同时, 还会将GPxDAT寄存器中的相应位清0。 用户可将该位清0; 该位并不影响数据输出。
15至0	保留。

### 串行外设接口

ADuC7023片内集成了一个完整的硬件串行外设接口(SPI)。SPI是一个工业标准同步串行接口, 允许同时双向传输8位数据(即全双工), 最大比特率可达20 Mbps。

该SPI端口可配置为主机或从机操作, 一般由4个引脚组成: MISO、MOSI、SCLK和SPIS $\overline{S}$ 。

#### MISO(主机输入, 从机输出)引脚

在主机模式下, MISO引脚被配置为输入线路; 在从机模式下, 配置为输出线路。主机上的MISO线路(数据输入)应与从机内的MISO线路(数据输出)相连。传送的数据是以字节(8位)为单位的串行数据, MSB优先。

#### MOSI(主机输出, 从机输入)引脚

在主机模式下, MOSI引脚被配置为输出线路; 在从机模式下, 配置为输入线路。主机上的MOSI线路(数据输出)应与从机内的MOSI线路(数据输入)相连。传送的数据是以字节(8位)为单位的串行数据, MSB优先。

#### SCLK(串行时钟输入/输出)引脚

主机串行时钟(SCLK)用于同步MOSI SCLK周期中发送和接收的数据。所以, 发送/接收一个字节需要8个SCLK周期。在主机模式下, SCLK引脚配置成输出端, 而在从机模式下, 配置成输入端。

在主机模式下, 时钟的极性和相位由SPICON寄存器控制, SPIDIV寄存器的值决定了比特率。比特率的计算公式如下:

$$f_{SERIAL\ CLOCK} = \frac{f_{UCLK}}{2 \times (1 + SPIDIV)}$$

其中:

$f_{UCLK}$  为POWCON1位7至位6选择的时钟。

SPI时钟的最高频率与时钟分频器位无关。

在从机模式下, 可对SPICON寄存器进行设置, 以配置预期输入时钟的相位和极性。从机可以从外部主机处接收数据(速率可达10 Mbps)。

在主机模式和从机模式下, 数据都在SCLK信号的一个沿发送, 并在另一个沿采样。所以, 从机时钟的极性和相位必须与主机配置一致。

#### SPI片选(SS输入)引脚

在SPI从机模式时, 置位 $\overline{SS}$ 引脚将启动数据传输, 该引脚为一个低电平有效输入信号。然后, SPI端口开始发送和接收8位数据, 直到发送结束时为止, 此时 $\overline{SS}$ 无效。在从机模式下,  $\overline{SS}$ 总是为输入。

在SPI主机模式下,  $\overline{SS}$ 是低电平有效输出信号。传输开始后, 它自动置位; 传输完成后, 它自动解除置位。

#### SPI功能的外部引脚配置

P1.1为从机片选引脚。在从机模式下, 该引脚作为输入引脚, 必须通过主机将其驱动为低电平。在主机模式下, 该引脚为输出引脚。当传输开始后, 该引脚的电平为低电平; 传输完成后, 该引脚的电平为高电平。

P1.0为SCLK引脚。

P0.6为主机输入、从机输出(MISO)引脚。

P0.7为主机输出、从机输入(MOSI)引脚。

要配置这些引脚在SPI模式下工作, 请参见“通用输入/输出”部分。

#### SPI寄存器

下列MMR寄存器用来控制SPI接口: SPISTA、SPIRX、SPITX、SPIDIV和SPICON。

#### SPI状态寄存器

名称: SPISTA

地址: 0xFFFF0A00

默认值: 0x0000

访问类型: 读取

功能: 该32位寄存器用于存储主机、从机模式下SPI接口的状态。



# ADuC7023

**表63. SPISTA MMR位分配**

位	名称	描述
15至12		保留位。
11	SPIREX	SPI接收FIFO存在过剩字节。 接收FIFO中字节的个数超过由SPICON寄存器中SPIMDE位规定的个数后，该位被置1。 FIFO中字节的个数不超过由SPICON寄存器中SPIMDE位规定的个数时，该位被清0。
10至8	SPIRXFSTA[2:0]	SPI接收FIFO状态位。 [000] = 接收FIFO为空。 [001] = 该FIFO内有1个有效字节。 [010] = 该FIFO内有2个有效字节。 [011] = 该FIFO内有3个有效字节。 [100] = 该FIFO内有4个有效字节。
7	SPIFOF	SPI接收FIFO溢出状态位。 当接收FIFO已满时，如果再次向该FIFO内写入数据，该位被置1。除非SPICON寄存器的SPIRFLH位将该位置1，否则该位被置1将产生一个中断。 读取SPISTA寄存器的内容后，该位清0。
6	SPIRXIRQ	SPI接收IRQ状态位。 产生接收中断时，该位置1。SPICON寄存器的SPITMDE位置1时，在接收到所需字节数后，SPIRXIRQ位被置1。 读取SPISTA寄存器的内容后，该位清0。
5	SPITXIRQ	SPI发送IRQ状态位。 产生发送中断时，该位置1。SPICON寄存器的SPITMDE位置1时，在发送所需字节数后，SPITXIRQ位被置1。 读取SPISTA寄存器的内容后，该位清0。
4	SPITXUF	SPI发送FIFO下溢。 当启动一次发送操作且发送FIFO内没有有效数据时，该位置1。除非SPICON寄存器的SPITFLH位将该位置1，否则该位被置1将产生一个中断。 读取SPISTA寄存器的内容后，该位清0。
3至1	SPITXFSTA[2:0]	SPI发送FIFO状态位。 [000] = 发送FIFO为空。 [001] = 该FIFO内有1个有效字节。 [010] = 该FIFO内有2个有效字节。 [011] = 该FIFO内有3个有效字节。 [100] = 该FIFO内有4个有效字节。
0	SPIISTA	SPI中断状态位。 SPI中断发生时，该位被置1。 读取SPISTA寄存器的内容后，该位清0。

## SPIRX寄存器

名称： SPIRX  
地址： 0xFFFF0A04  
默认值： 0x00  
访问类型： 读取  
功能： 该8位寄存器是SPI接收寄存器。

## SPITX寄存器

名称： SPITX  
地址： 0xFFFF0A08  
默认值： 0xXX  
访问类型： 写入  
功能： 该8位寄存器是SPI发送寄存器。



**SPIDIV寄存器**

名称： SPIDIV  
地址： 0xFFFF0A0C  
默认值： 0x00  
访问类型： 读/写  
功能： 该8位寄存器是SPI波特率选择寄存器。

**SPI控制寄存器**

名称： SPICON  
地址： 0xFFFF0A10  
默认值： 0x0000  
访问类型： 读/写  
功能： 该16位寄存器用于在主机和从机模式下配置SPI外设。

表64. SPICON MMR位分配

位	名称	描述
15至14	SPIMDE	SPI IRQ模式位。这些位用于配置在传输过程中何时发生发送/接收中断。  [00] = 传输完1个字节后，产生发送中断。FIFO接收到一个或以上字节后，产生接收中断。 [01] = 传输完2个字节后，产生发送中断。FIFO接收到两个或以上字节后，产生接收中断。 [10] = 传输完3个字节后，产生发送中断。FIFO接收到三个或以上字节时，产生接收中断。 [11] = 传输完4个字节后，产生发送中断。当接收FIFO已满或存在四个字节时，产生接收中断。
13	SPITFLH	SPI发送FIFO清空使能位。 通过将该位置1，可清除发送FIFO。该位无法自清0；需要一个单次清空操作时，应将该位置1。 如果该位的值总保持为1，那么，发送0x00还是最后被发送的数值取决于SPIZEN位的值。 该位为1时，无法对发送FIFO进行写操作。 通过将该位清0，可禁用发送FIFO清除。
12	SPIRFLH	SPI接收FIFO清除使能位。 通过将该位置1，可清除接收FIFO。该位无法自清0；需要一个单次清空操作时，应将该位置1。 该位置1后，所有向接收FIFO写数据的操作将被忽略，且系统不产生中断。 如果该位置1且SPITMDE = 0，对接收FIFO执行读操作可以启动一次数据传输。 通过将该位清0，可禁用接收FIFO清除。
11	SPICONT	连续传输使能。 通过将该位置1，用户可使能连续传输。在主机模式下，数据传输连续进行，直到发送寄存器内无有效数据为止。SS置位，并在每一次8位串行传输期间保持置位，直到发送寄存器为空。 通过将该位清0，用户可禁用连续传输。每一次传输都是单独的8位串行传输。 如果SPITX寄存器中存在有效数据，那么经过1个串行时钟周期的停转时间后，重新开始传输数据。
10	SPILP	回送使能位。 通过将该位置1，用户可将MISO连接到MOSI并测试软件。 通过将该位清0，可返回正常模式。
9	SPIOEN	从机MISO输出使能位。 通过将该位置1，可让MISO在正常模式下工作。 通过将该位清0，可禁用MISO引脚上的输出驱动。该位被清0后，MISO引脚变为开漏极。
8	SPIROW	SPIRX上溢覆盖使能。 用户将该位置1后，新接收到的串行数据将覆盖SPIRX寄存器中的有效数据。 用户将该位清0后，新接收到的串行数据会被丢弃。
7	SPIZEN	发送FIFO为空时，SPI发送0。 将该位置1后，当发送FIFO无有效数据时，SPI发送0x00。 将该位清0后，当发送FIFO无有效数据时，SPI发送上一次发送的数值。
6	SPITMDE	SPI传输和中断模式。 通过将该位置1，用户可在向SPITX寄存器写入数据时开始传输。只有当发送寄存器为空时，才产生中断。 通过将该位清0，用户可在从SPITX寄存器读取数据时开始传输。只有当接收寄存器已满时，才产生中断。
5	SPILF	LSB优先传输使能位。 用户将该位置1后，优先发送LSB。 用户将该位清0后，优先发送MSB。
4	SPIWOM	SPI线或模式使能位。 通过将该位置1，可使能开漏数据输出。数据输出引脚需要外部上拉电阻。 通过将该位清0，可启动正常输出模式。
3	SPICPO	串行时钟极性模式位。 用户将该位置1后，串行时钟高电平空闲。 用户将该位清0后，串行时钟低电平空闲。
2	SPICPH	串行时钟相位模式位。 用户将该位置1后，串行时钟脉冲出现在每一次串行位传输的起始位置。 用户将该位清0后，串行时钟脉冲出现在每一次串行位传输的末尾。

位	名称	描述
1	SPI MEN	主机模式使能位。 通过将该位置1，用户可使能主机模式。 通过将该位清0，用户可使能从机模式。
0	SPI EN	SPI使能位。 通过将该位置1，用户可使能SPI。 通过将该位清0，用户可禁用SPI。

## I<sup>2</sup>C

ADuC7023集成两个I<sup>2</sup>C外设，用户可将这些设备配置成完全I<sup>2</sup>C兼容型I<sup>2</sup>C总线主机或者完全I<sup>2</sup>C总线兼容型从机。

引脚SDA和SCL用于数据传输，通过对这两个引脚进行“线与”配置，可以在多主机系统中进行仲裁。这两个引脚需要接外部上拉电阻。典型的上拉电阻值介于4.7 Ω和10 kΩ之间。

用户可对I<sup>2</sup>C总线系统内的I<sup>2</sup>C总线外设地址进行编程。没有进行传输时，可随时修改这个ID。用户可以对接口进行配置，以使其响应四个从机地址。

I<sup>2</sup>C系统的传输过程为：当总线处于空闲状态时，主机通过产生起始条件来启动传输；在初始地址传输期间，主机发送从机的地址和数据传输方向(读和/或写)；如果主机没有仲裁失效且从机进行了应答，那么开始向从机传输数据；传输会持续到主机发送一个停止条件为止，然后总线进入空闲状态。

在同一时刻，I<sup>2</sup>C外设无法既作为主机又作为从机。同一个I<sup>2</sup>C通道不能同时支持主机模式和从机模式。

ADuC7023上的I<sup>2</sup>C接口支持重复起始条件。在主机模式下，通过对ADuC7023进行编程，可以允许其重复启动一次。在从机模式下，ADuC7023可识别重复起始条件。在主机和从机模式下，器件可识别7位和10位总线地址。在I<sup>2</sup>C主机模式下，ADuC7023允许在一个传输时序下，从单个从机中连续读取512字节数据。在主机模式和从机模式下，均支持时钟延展。在从机模式下，通过对ADuC7023进行编程，可以允许其返回一个NACK。这样，可以保证在I<sup>2</sup>C数据传输结束时校验和字节是有效的。在主机模式下，支持总线仲裁。支持内、外部回送，用于I<sup>2</sup>C硬件测试。在回送模式下，在主机和从机模式下，发送和接收电路均有两字节的FIFO缓存。为用户提供状态位，以便控制上述FIFO缓存。

### I<sup>2</sup>C功能的外部引脚配置

ADuC7023器件的I<sup>2</sup>C引脚为用于I<sup>2</sup>C0的P0.4和P0.5，以及用于I<sup>2</sup>C1的P0.6和P0.7。

P0.4和P0.6的功能是传输I<sup>2</sup>C时钟信号，而P0.5和P0.7的功能是传输I<sup>2</sup>C数据信号。例如，要配置I<sup>2</sup>C0引脚(SCL0和SDA0)，GP0CON寄存器的位16和位20必须置为1，以启用I<sup>2</sup>C模式。而要配置I<sup>2</sup>C1引脚(SCL1和SDA1)，GP0CON寄存器的位25和位29必须置为1，以启用I<sup>2</sup>C模式，如GPIO部分所示。I<sup>2</sup>C1功能在32引脚封装上由P0.6和P0.7提供，而在40引脚封装则由P1.6和P1.7提供。

### 串行时钟生成

系统中的I<sup>2</sup>C主机生成传输串行时钟。主机通道经过配置，可以在快速模式(400 kHz)或标准模式(100 kHz)下工作。

I2CDIV寄存器中的比特率定义如下：

$$f_{SERIAL\ CLOCK} = \frac{f_{UCLK}}{(2 + DIVH) + (2 + DIVL)}$$

其中：

$f_{UCLK}$ 为分频之前的时钟和由POWCON1位4至位0选择的时钟。

DIVH是时钟高电平周期。

DIVL是时钟低电平周期。

因而，如果希望串行时钟为100 kHz，那么

$$\text{应该配置 } DIVH = DIVL = 0xCF$$

如果希望串行时钟为400 kHz，那么

$$\text{应该配置 } DIVH = 0x28, DIVL = 0x3C$$

I2CDIV寄存器与DIVH:DIVL相对应。

### I<sup>2</sup>C总线地址

#### 从机模式

在从机模式下，I2CxID0、I2CxID1、I2CxID2和I2CxID3寄存器包含器件ID。器件将4个I2CxIDx寄存器的内容与从总线主机处接收的地址字节相比较。为确保寻址准确，每一个ID寄存器的7个MSB必须与最先接收到的地址字节的7个MSB相同。在地址识别过程中，ID寄存器的LSB(传输方向位)被忽略。

ADuC7023还支持10位寻址模式。当I2CxSCON寄存器的位1(ADR10EN)置1时，在从机模式下，系统支持10位地址，且将该地址保存在I2CxID0寄存器和I2CxID1寄存器之中。10位地址的组成如下：

I2CxID0[0]：读/写位，不属于I<sup>2</sup>C地址。

I2CxID0[7:1] = 地址位[6:0]。

I2CxID1[2:0] = 地址位[9:7]。

I2CxID1[7:3]的值必须为11110b。

#### 主机模式

在主机模式下，I2CxADR0寄存器编程为存储器件的I<sup>2</sup>C地址。

在7位地址模式下，I2CxADR0[7:1]设置为存储器件地址。I2CxADR0[0]是读/写位。

在10位地址模式下，10位地址创建如下：

I2CxADR0[7:3]必须设置为11110b。

I2CxADR0[2:1] = 地址位[9:8]。

I2CxADR1[7:0] = 地址位[7:0]。

I2CxADR0[0]是读/写位。

**I<sup>2</sup>C寄存器**

比较器接口由一个16位寄存器CMPCON组成，如表46所示。

**I<sup>2</sup>C主机寄存器****I<sup>2</sup>CxMCON(I<sup>2</sup>C主机控制寄存器)**

名称： I2C0MCON, I2C1MCON

地址： 0xFFFFF0800, 0xFFFFF0900

默认值： 0x0000, 0x0000

访问类型： 读/写

功能： 这些16位寄存器用于在主机模式下配置I<sup>2</sup>C外设。

**表65. I2CxMCON寄存器位功能描述**

位	名称	描述
15至9		保留。这些位为保留位，不允许向其中写入数据。
8	I2CMCENI	I <sup>2</sup> C发送完成中断使能位。 通过将该位置1，当从I <sup>2</sup> C总线上检测到停止条件时，将产生中断。 该位可清除该中断源。
7	I2CNACKENI	I <sup>2</sup> C非应答信号接收中断使能位。 通过将该位置1，当I <sup>2</sup> C主机接收到非应答信号时，将产生中断。 该位可清除该中断源。
6	I2CALENI	I <sup>2</sup> C仲裁失效中断使能位。 通过将该位置1，当I <sup>2</sup> C主机没有获得对I <sup>2</sup> C总线的控制权时，将产生中断。 该位可清除该中断源。
5	I2CMTENI	I <sup>2</sup> C发送中断使能位。 通过将该位置1，当I <sup>2</sup> C主机完成一个字节的传输后，将产生中断。 该位可清除该中断源。
4	I2CMRENI	I <sup>2</sup> C接收中断使能位。 通过将该位置1，当I <sup>2</sup> C主机接收到数据时，将产生中断。 通过将该位清0，用户可以在I <sup>2</sup> C主机接收数据过程中禁用中断。
3	I2CMSEN	I <sup>2</sup> C主机SCL伸展使能位。 通过将该位置1，可使能时钟延展功能。当SCL处于低电平状态，将该位置1，可强制器件在I2CMSEN被清0之前让SCL保持低电平状态。当SCL处于高电平状态，通过将该位置1，可强制器件在下一个下降沿后让SCL保持低电平状态。 通过将该位清0，可禁用时钟延展功能。
2	I2CILEN	I <sup>2</sup> C内部回送使能位。 通过将该位置1，可使能回送测试模式。在该模式下，SCL和SDA信号在内部分别与各自的输入信号相连。 通过将该位清0，用户可禁用回送模式。
1	I2CBD	I <sup>2</sup> C主机撤回禁用位。 通过将该位置1，可允许器件与另一器件争夺对总线的控制权(即使另一个器件正在产生一个起始条件)。 通过将该位清0，则在I <sup>2</sup> C总线释放后执行撤回。
0	I2CMEN	I <sup>2</sup> C主机使能位。 通过将该位置1，用户可使能I <sup>2</sup> C主机模式。 通过将该位清0，可禁用I <sup>2</sup> C主机模式。

# ADuC7023

## I2CxMSTA(I<sup>2</sup>C主机状态寄存器)

名称: I2C0MSTA, I2C1MSTA

地址: 0xFFFFF0804, 0xFFFFF0904

默认值: 0x0000, 0x0000

访问类型: 读取

功能: 这些16位寄存器是主机模式下的I<sup>2</sup>C状态寄存器。

**表66. I2CxMSTA寄存器位功能描述**

位	名称	描述
15至11		保留。这些位是保留的。
10	I2CBBUSY	I <sup>2</sup> C总线忙碌状态位。 当从I <sup>2</sup> C总线上检测到起始条件后, 该位被置1。 当从I <sup>2</sup> C总线上检测到停止条件后, 该位被清0。
9	I2CMRxFO	主机接收FIFO溢出。 接收FIFO已满后, 又有一个字节的数据写入FIFO时, 该位被置1。 在其它条件下, 该位被清0。
8	I2CMTC	I <sup>2</sup> C发送完成状态位。 当主机与从机之间完成一次传输后, 该位被置1。如果I2CxMCON寄存器的I2CMCENI位被将该位置1, 当该位的值为1时, 使能中断。 该位可清除该中断源。
7	I2CMNA	I <sup>2</sup> C主机非应答数据位。 在执行数据写传输过程中, 当主机接收到一个非应答条件后, 该位被置1。如果I2CxMCON寄存器的I2CNACKENI位被将该位置1, 当该位的值为1时, 使能中断。 在其它条件下, 该位被清0。
6	I2CMBUSY	I <sup>2</sup> C主机忙碌状态位。 当主机忙于处理事务时, 该位被置1。 当主机处于就绪状态或者当另一主机取得了总线控制权时, 该位被清0。
5	I2CAL	I <sup>2</sup> C仲裁失效状态位。 当I <sup>2</sup> C主机未成功获得对I <sup>2</sup> C总线的控制权时, 该位被置1。如果I2C1MCON的I2CALENI位被置1, 当该位的值为1时, 产生中断。 在其它条件下, 该位被清0。
4	I2CMNA	I <sup>2</sup> C主机非应答地址位。 当主机接收到一个与地址对应的非应答条件后, 该位被置1。如果I2C1MCON寄存器的I2CNACKENI位被置1, 当该位的值为1时, 产生中断。 在其它条件下, 该位被清0。
3	I2CMRXQ	I <sup>2</sup> C主机接收请求位。 当数据进入接收FIFO后, 该位被置1。如果I2C1MCON的I2CMRENI位被置1, 则产生中断。 在其它条件下, 该位被清0。
2	I2CMTXQ	I <sup>2</sup> C主机发送请求位。 如果发送FIFO为空或仅包含一个字节, 且主机已经传输一个地址并执行了写操作, 则该位置1。如果I2C1MCON寄存器的I2CMTENI位被将该位置1, 当该位被置1时, 使能中断。 在其它条件下, 该位被清0。
1 to 0	I2CMTFSTA	I <sup>2</sup> C主机发送FIFO状态位。 00 = I <sup>2</sup> C主机发送FIFO为空。 01 = 在主机发送FIFO中包含1个字节的数据。 10 = 在主机发送FIFO中包含1个字节的数据。 11 = I <sup>2</sup> C主机发送FIFO已满。

**I2CxMRX(I<sup>2</sup>C主机接收寄存器)**

名称： I2C0MRX, I2C1MRX  
 地址： 0xFFFF0808, 0xFFFF0908  
 默认值： 0x00  
 访问类型： 只读  
 功能： 这些8位寄存器是I<sup>2</sup>C主机接收寄存器。

**I2CxMTX(I<sup>2</sup>C主机发送寄存器)**

名称： I2C0MTX, I2C1MTX  
 地址： 0xFFFF080C, 0xFFFF090C  
 默认值： 0x00, 0x00  
 访问类型： 只写  
 功能： 这些8位寄存器是I<sup>2</sup>C主机发送寄存器。

**I2CxMCNT0(I<sup>2</sup>C主机读取计数寄存器)**

名称： I2C0MCNT0, I2C1MCNT0  
 地址： 0xFFFF0810, 0xFFFF0910  
 默认值： 0x0000, 0x0000  
 访问类型： 读/写  
 功能： 这些16位寄存器用于保存主机需要从从机中读取的字节数。

**表67. I2CxMCNT0寄存器位功能描述：地址 = 0xFFFF0810、0xFFFF0910。默认值 = 0x0000**

位	名称	描述
15至9		保留。
8	I2CRECNT	当需要从从机处读取的字节数多于256时，该位置1。当读取的字节数为256或更少时，该位清0。
7至0	I2CRCNT	这8位所保存的数值为需要从从机处读取的字节数减1的结果。如果只需要读取1个字节，则应将这8位清0。

**I2CxMCNT1(I<sup>2</sup>C主机当前读取计数寄存器)**

名称： I2C0MCNT1, I2C1MCNT1  
 地址： 0xFFFF0814, 0xFFFF0914  
 默认值： 0x00, 0x00  
 访问类型： 读取  
 功能： 这些8位寄存器用于保存在一次读序列中主机从从机处接收到的字节数。

**I2CxADR0(I<sup>2</sup>C地址0寄存器)**

名称： I2C0ADR0, I2C1ADR0  
 地址： 0xFFFF0818, 0xFFFF0918  
 默认值： 0x00  
 访问类型： 读/写  
 功能： 当主机开始与从机进行通信后，这些8位寄存器用于保存7位从机地址和读/写位。

**表68. 7位地址模式下的I2CxADR0寄存器：地址 = 0xFFFF0818、0xFFFF0918。默认值 = 0x00**

位	名称	描述
7至1	I2CADR	这些位包含目标从机的7位地址。
0	R/W	位0为读/写位。 当该位置1时，表示需要读序列。 当该位清0时，表示需要写序列。

**表69. 10位地址模式下的I2CxADR0寄存器**

位	名称	描述
7至3		在10位地址模式下，这些位的值必须为[11110b]。
2至1	I2CMADR	在10位地址模式下，这些位包含ADDR[9:8]。
0	R/W	读/写位。 当该位置1时，表示需要读序列。 当该位清0时，表示需要写序列。



# ADuC7023

## I2CxADR1(I<sup>2</sup>C地址1寄存器)

名称: I2C0ADR1, I2C1ADR1

地址: 0xFFFF081C, 0xFFFF091C

默认值: 0x00

访问类型: 读/写

功能: 这些8位寄存器只可用于10位寻址模式。这些寄存器包含地址的最低有效字节。

**表70. 10位地址模式下的I2CxADR1寄存器**

位	名称	描述
7至0	I2CLADR	在10位地址模式下, 这些位包含ADDR[7:0]。

## I2CxDIV(I<sup>2</sup>C主机时钟控制寄存器)

名称: I2C0DIV, I2C1DIV

地址: 0xFFFF0824, 0xFFFF0924

默认值: 0x1F1F

访问类型: 读/写

功能: 这些寄存器用于控制SCL引脚上由主机产生的I<sup>2</sup>C时钟的频率。详情见I<sup>2</sup>C开头部分。

**表72. I2CxSCON寄存器位功能描述**

位	名称	描述
15至11		保留位。
10	I2CSTXENI	从机发送中断使能位。 通过将该位置1, 当从机发送一个字节后, 将产生中断。 该位可清除该中断源。
9	I2CSRXENI	从机接收中断使能位。 通过将该位置1, 当从机接收到数据后, 将产生中断。 该位可清除该中断源。
8	I2CSSENI	I <sup>2</sup> C停止条件检测中断使能位。 通过将该位置1, 当从I2C总线上检测到停止条件时, 将产生中断。 该位可清除该中断源。
7	I2CNACKEN	I <sup>2</sup> C非应答使能位。 通过将该位置1, 可对传输序列中的下一字节不作出应答。 通过将该位清0, 可允许硬件对应答/非应答序列进行控制。
6	I2CSSEN	I <sup>2</sup> C从机SCL延展使能位。 通过将该位置1, 可启用时钟延展功能。当SCL处于低电平状态, 通过将该位置1, 可强制器件在I2CMSEN被清0之前让SCL保持低电平状态。当SCL处于高电平状态, 通过将该位置1, 可强制器件在下一个下降沿后让SCL保持低电平状态。 通过将该位清0, 可禁用时钟延展功能。
5	I2CSETEN	I <sup>2</sup> C发送提前中断使能位。 通过将该位置1, 可允许在读位发送的过程中, 在SCL正向沿之后立刻使能发送请求中断。 通过将该位清0, 可允许在读位发送的过程中, 在SCL负向沿之后立刻使能发送请求中断。

**表71. I2CxDIV寄存器**

位	名称	描述
15至8	DIVH	这些位用于控制SCL高电平周期的持续时间。
7至0	DIVL	这些位用于控制SCL低电平周期的持续时间。

## I<sup>2</sup>C从机寄存器

### I2CxSCON(I<sup>2</sup>C从机控制寄存器)

名称: I2C0SCON, I2C1SCON

地址: 0xFFFF0828, 0xFFFF0928

默认值: 0x0000

访问类型: 读/写

功能: 这些16位寄存器用于在从机模式下配置I<sup>2</sup>C外设。

位	名称	描述
4	I2CGCCLR	I <sup>2</sup> C广播状态与ID清除位。 通过向该位写入1，可清除I2CxSSTA寄存器中的呼叫状态与ID位。 在其它状况下，该位被清0。
3	I2CHGCEN	I <sup>2</sup> C硬件广播使能位。硬件广播使能位。当该位和位2置1时，如果已接收到一个广播信号(地址为0x00)和1字节数据，器件将对接收寄存器中的数据和I2CxALT中的数据进行比较。如果数据匹配，表明器件接收到一个硬件广播。当器件需要紧急呼叫一个主机而又不知道呼叫哪一个时，可使用该功能。该广播消息会发送到总线上的所有主机。ADuC7023会监视这些地址。要求主机注意的器件会将自己的地址嵌入到消息中。所有的主机都会侦听这些消息，然后能够处理该器件要求的主机会与其从机通信并进行相应操作。根据2000年1月的I <sup>2</sup> C总线规范，I2CxALT寄存器的LSB应该始终写入1。 通过将该位和I2CGCEN置1，可在从机模式下使能硬件广播识别。 通过将该位清0，可禁用硬件广播命令识别。
2	I2CGCEN	I <sup>2</sup> C广播使能位。通过将该位置1，可以让从机为I <sup>2</sup> C广播发送有效应答，写地址0x00。然后器件将识别一个数据位。如果器件接收到的数据是0x06，即由硬件复位和对从机地址的可编程部分进行写操作，那么，根据2000年1月的I <sup>2</sup> C总线规范，此时I <sup>2</sup> C接口复位。这个命令可用于复位整个I <sup>2</sup> C系统。如果接收到的数据为0x04，即由硬件对从机地址可编程部分进行写操作，则当产生任一广播后，广播中断状态位被置位。在复位后，用户必须通过重新对器件地址进行编程来进行恰当的操作。 通过将该位置1，可允许从机应答I <sup>2</sup> C广播命令。 通过将该位清0，可禁用广播命令识别。
1	ADR10EN	I <sup>2</sup> C 10位地址模式。 通过将该位置1，可启用10位地址模式。 通过将该位清0，可启用正常地址模式。
0	I2CSEN	I <sup>2</sup> C从机使能位。 通过将该位置1，用户可启用I <sup>2</sup> C从机模式。 通过将该位清0，用户可I <sup>2</sup> C从机模式。

### I2CxSSTA(I<sup>2</sup>C从机状态寄存器)

名称： I2C0SSTA, I2C1SSTA

地址： 0xFFFF082C, 0xFFFF092C

默认值： 0x0000, 0x0000

访问类型： 读/写

功能： 这些16位寄存器是从机模式下的I<sup>2</sup>C状态寄存器。

# ADuC7023

**表73. I2CxSSTA寄存器位功能描述**

位	名称	描述
15		保留位。
14	I2CSTA	下列情况下，此位置1：检测到后跟匹配地址的起始条件；接收到起始字节(0x01)；已使能广播且接收到广播代码(0x00)。 接收到停止条件后，该位清0。
13	I2CREPS	检测到重复起始条件时，该位置1。 接收到停止条件后，该位清0。
12 to 11	I2CID[1:0]	I <sup>2</sup> C地址匹配寄存器。这些位用于表示哪个I2CxIDx寄存器与接收到的地址相匹配。 [00] = 接收到的地址与I2CxID0相匹配。 [01] = 接收到的地址与I2CxID1相匹配。 [10] = 接收到的地址与I2CxID2相匹配。 [11] = 接收到的地址与I2CxID3相匹配。
10	I2CSS	I <sup>2</sup> C起始后停止条件检测位。 若在起始条件和匹配地址之后检测到停止条件，则该位置1。 如果I2CxSCON的I2CSSENI位将该位置1，则产生中断。 读该寄存器时，该位清0。
9 to 8	I2CGCID[1:0]	I <sup>2</sup> C广播ID位。 [00] = 未收到广播。 [01] = 广播复位和程序地址。 [10] = 通用程序地址。 [11] = 广播匹配可供选择的ID。 无法通过广播复位命令来清除这些位。 通过向I2CxSCON的I2CGCCLR位写入1，可清除这些位。
7	I2CGC	I <sup>2</sup> C广播状态位。 从机接收到任何类型的广播命令后，该位置1。从机接收到复位命令后，寄存器返回各自的默认状态。从机接收到硬件广播命令后，接收FIFO将保存命令的第2个字节，它可以与I2CxALT寄存器的值进行比较。 通过向I2CxSCON的I2CGCCLR位写入1，可清除该位。
6	I2CSBUSY	I <sup>2</sup> C从机忙碌状态位。 从机接收到起始条件后，该位置1。 发生下列情形之一，硬件会自动清0该位：接收到的地址与I2CxIDx寄存器的内容不匹配，从机收到停止条件，或重复起始地址与I2CxIDx寄存器的内容不匹配。
5	I2CSNA	I <sup>2</sup> C从机非应答数据位。 从机发出非应答信号以响应总线地址时，该位被置1。在下列条件下，该位被断言：从机因发送FIFO内无数据而返回一非应答信号，或者I2CxSCON寄存器的I2CNACKEN位被置1。 在其它条件下，该位被清0。
4	I2CSRxFO	从机接收FIFO溢出。 接收FIFO已满后，又有一个字节的数据写入FIFO时，该位被置1。 在其它条件下，该位被清0。
3	I2CSRXQ	I <sup>2</sup> C从机接收请求位。 从机接收FIFO不为空时，该位被置1。I2CxSCON寄存器的I2CSRXENI位置1时，通过将I2CSRXQ置1可产生中断。 通过对接收FIFO执行读操作或清除操作，可将该位清0。
2	I2CSTXQ	I <sup>2</sup> C从机发送请求位。 从机接收到一个匹配的地址并执行读操作后，该位被置1。如果I2CxSCON寄存器的I2CSETEN位清0，则在读位发送的过程中，SCL负向沿之后，I2CSTXQ的值立即被置1。如果I2CxSCON寄存器的I2CSETEN位置1，则在读位发送的过程中，SCL正向沿之后，I2CSTXQ的值立即被置1。 I2CxSCON寄存器的I2CSTXENI位置1时，通过将I2CSTXQ置1可产生中断。 在其它条件下，该位被清0。

位	名称	描述
1	I2CSTFE	I <sup>2</sup> C从机FIFO下溢状态位。 主机要求从机发送数据时，如果发送FIFO为空，则该位变为高电平。在读位操作期间，在SCL上升沿对该位进行断言。 在其它条件下，该位被清0。
0	I2CETSTA	I <sup>2</sup> C从机传输前FIFO状态位。 如果I2CxSCON寄存器的I2CSETEN位清0，则当从机发送FIFO为空时，I2CETSTA变为高电平。 如果I2CxSCON寄存器的I2CSETEN位置1，则在写位发送的过程中，SCL正向沿之后，I2CETSTA的值立即被置1。在一次传输过程中，该位只能被断言一次。 对该位执行读操作后，该位被清0。

**I2CxSRX(I<sup>2</sup>C从机接收寄存器)**

名称： I2C0SRX, I2C1SRX  
地址： 0xFFFF0830, 0xFFFF0930  
默认值： 0x00  
访问类型： 读取  
功能： 这些8位寄存器是I<sup>2</sup>C从机接收寄存器。

**I2CxSTX(I<sup>2</sup>C从机发送寄存器)**

名称： I2C0STX, I2C1STX  
功能： 0xFFFF0834, 0xFFFF0934  
默认值： 0x00  
访问类型： 写入  
功能： 这些8位寄存器是I<sup>2</sup>C从机发送寄存器。

**I2CxALT(I<sup>2</sup>C硬件广播识别寄存器)**

名称： I2C0ALT, I2C1ALT  
功能： 0xFFFF0838, 0xFFFF0938  
默认值： 0x00  
访问类型： 读/写  
功能： I2CxSCON寄存器的位3置1时，这些8位寄存器可用于识别硬件广播。当主机无法为从机生成地址，从机却必须为主机生成地址时，这些寄存器便可以发挥作用了。

**I2CxIDx(I<sup>2</sup>C从机ID寄存器)**

名称： I2C0IDx, I2C1IDx  
地址： 0xFFFF093C = I2C1ID0  
0xFFFF083C = I2C0ID0  
0xFFFF0940 = I2C1ID1  
0xFFFF0840 = I2C0ID1  
0xFFFF0944 = I2C1ID2  
0xFFFF0844 = I2C0ID2  
0xFFFF0948 = I2C1ID3  
0xFFFF0848 = I2C0ID3  
默认值： 0x00  
访问类型： 读/写  
功能： 通过编程，可在这些8位寄存器内编程从机的I<sup>2</sup>C总线ID。详情见I<sup>2</sup>C总线地址部分。

**I<sup>2</sup>C公共寄存器****I2CxFSTA(I<sup>2</sup>C FIFO状态寄存器)**

名称： I2C0FSTA, I2C1FSTA  
功能： 0xFFFF084C, 0xFFFF094C  
默认值： 0x0000  
访问类型： 读/写  
功能： 这些16位寄存器用于存储在主机和从机模式下接收/发送FIFO的状态。

**表74. I2CxFSTA寄存器位功能描述**

位	名称	描述
15至10		保留位。
9	I2CFMTX	通过将该位置1，可清除主机发送FIFO。
8	I2CFSTX	通过将该位置1，可清除从机发送FIFO。
7至6	I2CMRXSTA	I <sup>2</sup> C主机接收FIFO状态位。 [00] = FIFO为空。 [01] = FIFO字节写入。 [10] = FIFO中有1字节数据。 [11] = FIFO已满。
5至4	I2CMTXSTA	I <sup>2</sup> C主机发送FIFO状态位。 [00] = FIFO为空。 [01] = FIFO字节写入。 [10] = FIFO中有1字节数据。 [11] = FIFO已满。
3至2	I2CSRXSTA	I <sup>2</sup> C从机接收FIFO状态位。 [00] = FIFO为空 [01] = FIFO字节写入 [10] = FIFO中有1字节数据 [11] = FIFO已满
1至0	I2CSTXSTA	I <sup>2</sup> C从机发送FIFO状态位。 [00] = FIFO为空。 [01] = FIFO字节写入。 [10] = FIFO中有1字节数据。 [11] = FIFO已满。

## 可编程逻辑阵列(PLA)

每个ADuC7023均集成有一个完整的可编程逻辑阵列(PLA)，它由十六个PLA单元组成。

每个PLA单元都包含有一个双输入的查询表，通过配置可以实现任何基于双输入和一个触发器的逻辑输出功能，如图39所示。

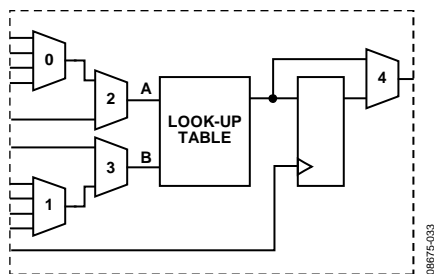


图39. PLA单元

ADuC7023上共有20个GPIO引脚可用于PLA。其中包括11个输入引脚和9个输出引脚，在使用PLA功能之前需要在GPxCON寄存器中对这些引脚进行配置。

PLA是通过一组用户寄存器进行配置的。PLA的输出可以连接到内部中断系统、ADC的CONV<sub>START</sub>信号、一个寄存器或者8个PLA输出引脚中的任何一个。

**表75. 单元输入/输出**

PLA模块0			PLA模块1		
单元	输入	输出	单元	输入	输出
0	P0.4	P0.7	8	P0.0	P0.2
1	P0.5	P1.0	9	P0.1	P0.3
2	P0.6	P1.1	10	P2.4	P2.5 <sup>1</sup>
3	P1.2	P1.4	11	NC	NC
4	P1.3	P1.5	12	NC	NC
5	P1.6	P2.1 <sup>1</sup>	13	NC	NC
6	P1.7	P2.2	14	NC	NC
7	P2.0	P2.3	15	NC	NC

<sup>1</sup> 仅限内部引脚。通过GPxDAT寄存器读取。

## PLA寄存器接口

PLA外设接口包括22个寄存器，以下是对它们的具体描述。

## PLAELMx寄存器

PLAELMx是单元0到单元15的控制寄存器。通过这些寄存器，可以配置每个单元的输入和输出多路复用器、在查询表中选择功能和选择旁路或使用触发器(参见表77)。

**表76. PLAELMx寄存器**

名称	地址	默认值	访问类型
PLAELM0	0xFFFF0B00	0x0000	R/W
PLAELM1	0xFFFF0B04	0x0000	R/W
PLAELM2	0xFFFF0B08	0x0000	R/W
PLAELM3	0xFFFF0B0C	0x0000	R/W
PLAELM4	0xFFFF0B10	0x0000	R/W
PLAELM5	0xFFFF0B14	0x0000	R/W
PLAELM6	0xFFFF0B18	0x0000	R/W
PLAELM7	0xFFFF0B1C	0x0000	R/W
PLAELM8	0xFFFF0B20	0x0000	R/W
PLAELM9	0xFFFF0B24	0x0000	R/W
PLAELM10	0xFFFF0B28	0x0000	R/W
PLAELM11	0xFFFF0B2C	0x0000	R/W
PLAELM12	0xFFFF0B30	0x0000	R/W
PLAELM13	0xFFFF0B34	0x0000	R/W
PLAELM14	0xFFFF0B38	0x0000	R/W
PLAELM15	0xFFFF0B3C	0x0000	R/W

表 77. PLAELMx寄存器位功能描述

位	值	描述
31至11		保留。
10至9		Mux0控制位(见表81)。
8至7		Mux1控制位(见表81)。
6		Mux2控制位。 通过将该位置1, 用户可选择Mux0的输出。 通过将该位清0, 用户可从PLADIN寄存器选择位值。
5		Mux3控制位。 通过将该位置1, 用户可选择特定单元的输入引脚。 通过将该位清0, 用户可选择Mux1的输出。
4至1		查询表控制位。
	0000	0。
	0001	或非。
	0010	B与A非。
	0011	A非。
	0100	A与B非。
	0101	B非。
	0110	异或。
	0111	与非。
	1000	与。
	1001	同或。
	1010	B。
	1011	A非或B。
	1100	答:
	1101	A或B非。
	1110	或。
	1111	1。
0		Mux4控制位。 通过将该位置1, 用户可旁路触发器。 通过将该位清0, 用户可选择触发器(默认值为0)。

**PLACLK寄存器**

名称:	PLACLK
地址:	0xFFFF0B40
默认值:	0x00
访问类型:	读/写
功能:	PLACLK是触发器的时钟选择寄存器。当使用GPIO引脚作为PLA模块的时钟输入时, 最大频率为41.78 MHz。

表 78. PLACLK寄存器位功能描述

位	值	描述
31至7		保留。
6至4		时钟源选择。
	000	P0.5引脚连接的GPIO时钟。
	001	P1.1引脚连接的GPIO时钟。
	010	P1.6引脚连接的GPIO时钟。
	011	HCLK。
	100	外部32.768 kHz晶振。
	101	定时器1溢出。
	110	UCLK
	111	内部32.768振荡器。
3		保留。
2至0		时钟源选择。
	000	P0.5引脚连接的GPIO时钟。
	001	P1.1引脚连接的GPIO时钟。
	010	P1.6引脚连接的GPIO时钟。
	011	HCLK。
	100	外部32.768 kHz晶振。
	101	定时器1溢出。
	110	UCLK
	111	内部32.768振荡器。

**PLAIRQ寄存器**

名称:	PLAIRQ
地址:	0xFFFF0B44
默认值:	0x00000000
访问类型:	读/写
功能:	PLAIRQ可以使能IRQ0和/或IRQ1和选择IRQ中断源。

表 79. PLAIRQ寄存器位功能描述

位	值	描述
31至13		保留。
12		PLA IRQ1使能位。
11至8		PLA单元0。
	0001	PLA单元1。
	0010	PLA单元2。
	0011	PLA单元3。
	0100	PLA单元4。
	0101	PLA单元5。
	0110	PLA单元6。
	0111	PLA单元7。
	1000	PLA单元8。
	1001	PLA单元9。
	1010	PLA单元10。
	1011	PLA单元11。
	1100	PLA单元12。
	1101	PLA单元13。
	1110	PLA单元14。
	1111	PLA单元15。
7至5		保留。

# ADuC7023

位	值	描述
4		PLA IRQ0使能位。 通过将该位置1，用户可启用从PLA输出IRQ0。 通过将该位清0，用户可禁用从PLA输出IRQ0。
3至0		PLA IRQ0中断源。
	0000	PLA单元0。
	0001	PLA单元1。
	0010	PLA单元2。
	0011	PLA单元3。
	0100	PLA单元4。
	0101	PLA单元5。
	0110	PLA单元6。
	0111	PLA单元7。
	1xxx	保留。



表80. 反馈配置

位	值	PLAELM0	PLAELM1至PLAELM7	PLAELM8	PLAELM9至PLAELM15
10至9	00	单元15	单元0	单元7	单元8
	01	单元2	单元2	单元10	单元10
	10	单元4	单元4	单元12	单元12
	11	单元6	单元6	单元14	单元14
8至7	00	单元1	单元1	单元9	单元9
	01	单元3	单元3	单元11	单元11
	10	单元5	单元5	单元13	单元13
	11	单元7	单元7	单元15	单元15

**PLAADC寄存器**

名称: PLAADC  
地址: 0xFFFF0B48  
默认值: 0x00000000  
访问类型: 读/写  
功能: PLAADC是配置PLA作为ADC起始转换信号来源的寄存器。

表81. PLAADC寄存器位功能描述

位	值	描述
31至5		保留。
4		ADC转换启动使能位。 通过将该位置1, 用户可使能从PLA启动ADC转换。 通过将该位清0, 用户可禁用从PLA启动ADC转换。
3至0		ADC转换起始源选择位。
	0000	PLA单元0.
	0001	PLA单元1.
	0010	PLA单元2.
	0011	PLA单元3.
	0100	PLA单元4.
	0101	PLA单元5.
	0110	PLA单元6.
	0111	PLA单元7.
	1000	PLA单元8.
	1001	PLA单元9.
	1010	PLA单元10.
	1011	PLA单元11.
	1100	PLA单元12.
	1101	PLA单元13.
	1110	PLA单元14.
	1111	PLA单元15.

**PLADIN寄存器**

名称: PLADIN  
地址: 0xFFFF0B4C  
默认值: 0x00000000  
访问类型: 读/写  
功能: PLADIN是一个PLA的数据输入寄存器。

表82. PLADIN寄存器位功能描述

位	描述
31至16	保留。
15至0	单元15至单元0的输入位。

**PLADOUT寄存器**

名称: PLADOUT  
地址: 0xFFFF0B50  
默认值: 0x00000000  
访问类型: 读取  
功能: PLADOUT是一个PLA的数据输出寄存器。  
这个寄存器是始终更新的。

表83. PLADOUT寄存器位功能描述

位	描述
31至16	保留。
15至0	单元15至单元0的输出位。

**PLALCK寄存器**

名称: PLALCK  
地址: 0xFFFF0B54  
默认值: 0x00  
访问类型: 写入  
功能: PLALCK是一个PLA锁定选择寄存器。位0只可以写入一次。该位一旦被置1, 除了PLADIN寄存器外其它任何PLA寄存器的值都不允许修改。开发系统提供了一套PLA工具, 通过它可以很容易地对PLA进行配置。

## 脉宽调制器

### 脉宽调制器概述

ADuC7023 集成了一个5通道脉宽调制器(PWM)接口。PWM输出即可用于驱动H桥或也可作为标准PWM输出。上电后，PWM输出的默认为H桥。这可以确保在默认状态下电机是关闭的。在标准PWM模式下，从三对PWM引脚内输出信号。用户既可以控制每一对输出引脚的工作周期，又可以单独控制每一个输出端的占空比。

**表84. PWM寄存器**

寄存器名称	描述
PWMCON1	PWM控制寄存器1。
PWM0COM0	用于比较PWM输出0和PWM输出1的比较寄存器0。
PWM0COM1	用于比较PWM输出0和PWM输出1的比较寄存器1。
PWM0COM2	用于比较PWM输出0和PWM输出1的比较寄存器2。
PWM0LEN	用于PWM输出0和PWM输出1的频率控制寄存器。
PWM1COM0	用于比较PWM输出2和PWM输出3的比较寄存器0。
PWM1COM1	用于比较PWM输出2和PWM输出3的比较寄存器1。
PWM1COM2	用于比较PWM输出2和PWM输出3的比较寄存器2。
PWM1LEN	用于PWM输出2和PWM输出3的频率控制寄存器。
PWM2COM0	PWM输出4的比较寄存器0。
PWM2COM1	PWM输出4的比较寄存器1。
PWMCLRI	PWM中断清除寄存器。

PWMxCOMx寄存器用于控制在各种模式下改变PWM输出状态的时间点。第一对PWM输出示例(PWM0和PWM1)如图40所示。

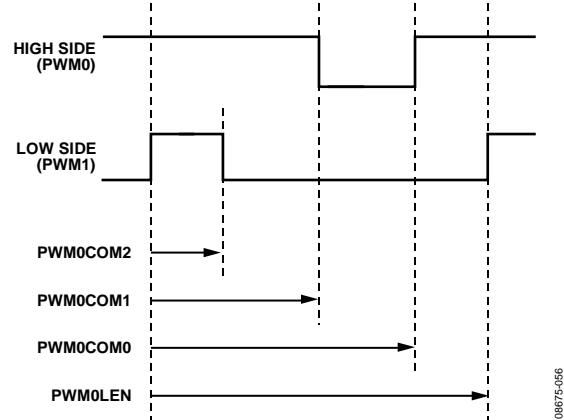


图40. PWM时序

通过PWMCON1，可将PWM时钟频率设定为以下值之一：UCLK除以2/4/8/16/32/64/128/256。PWMxLEN用于设定PWM周期长度。

PWM波形由16位定时器的计数值和比较寄存器的内容来决定，如图40所示的PWM0和PWM1的波形。

当定时器计数值达到PWM0LEN时，低端波形(PWM1)变为高电平；当定时器计数值达到PWM0COM2内所保存的数值或者当高端波形(PWM0)变为低电平时，PWM1变为低电平。

当定时器计数值达到PWM0COM0内所保存的数值时，高端波形(PWM0)变为高电平；当定时器计数值达到PWM0COM1内所保存的数值时，PWM0变为低电平。

### PWMCON1控制寄存器

名称： PWMCON1

地址： 0xFFFF0F80

默认值： 0x0012

访问类型： 读/写

功能： 该16位寄存器用于配置PWM输出。

表85. PWMCON1寄存器位功能描述

位	名称	描述
14	SYNC	使能PWM同步功能。 如果用户将该位置1, 当检测到P2.2/SYNC引脚上的由高到低跃迁后, 所有PWM计数器将在下一个时钟沿复位。 如果用户将该位清0, 则可以忽略在P2.2/SYNC引脚上发生的高低跃迁。
13	保留	用户置0。
12	PWM3INV	通过将该位置1, 用户可将PWM3反相。 通过将该位清0, 用户可将PWM3置于正常模式下。
11	PWM1INV	通过将该位置1, 用户可将PWM1反相。 通过将该位清0, 用户可将PWM1置于正常模式下。
10	PWMTRIP	通过将该位置1, 用户可使能PWM触发中断。当PWM触发中断(P1.5/PWM <sub>TRIPINPUT</sub> 引脚)处于低电平时, PWMEN位被清0, 且产生中断。 通过将该位清0, 用户可禁用PWMTRIP中断。
9	ENA	当HOFF = 0且HMODE = 1时, 该位可用。请注意, 该位仅在H桥模式下有效。 通过将该位置1, 用户可使能PWM输出。 通过将该位清0, 用户可禁用PWM输出。 如果HOFF = 1且HMODE = 1, 请参见表86。
8至6	PWMCP[2:0]	PWM时钟预分频器位。设置UCLK分频数。 [000] = UCLK/2. [001] = UCLK/4. [010] = UCLK/8. [011] = UCLK/16. [100] = UCLK/32. [101] = UCLK/64. [110] = UCLK/128. [111] = UCLK/256.
5	POINV	通过将该位置1, 用户可反相所有PWM输出。 如果用户将该位清0, 则允许正常的PWM输出。
4	HOFF	屏蔽高端。 通过将该位置1, 用户可强制PWM0和PWM2输出高电平信号。这样做还可强制PWM1和PWM3输出低电平信号。 如果用户将该位清0, 则允许正常的PWM输出。
3	LCOMP	载入比较寄存器。 用户置1后, 在PWM定时器进行从0x00到0x01的下一跃迁时, PWMxCOMx的值将被载入内部比较寄存器。 如果用户将该位清0, 则允许使用保存在内部比较寄存器中的数值。
2	DIR	方向控制。 用户置1后, 当PWM2和PWM3处于低电平状态时, 使能PWM0和PWM1作为输出信号。 用户清0后, 当PWM0和PWM1处于低电平状态时, 使能PWM2和PWM3作为输出信号。
1	HMODE	启用H桥模式 <sup>1</sup> 。 通过将该位置1, 用户可启用H桥模式。 通过将该位清0, 用户可让PWM在标准模式下工作。
0	PWMEN	通过将该位置1, 用户可使能全部PWM输出。 通过将该位清0, 用户可禁用全部PWM输出。

<sup>1</sup>在H桥模式下, HMODE = 1。关于PWM输出选择的说明, 参见表86。

# ADuC7023

上电后，PWMCON1寄存器的默认值为0x0012(HOFF = 1且HMODE = 1)。在默认状态下，PWM相关的所有GPIO引脚均被设定为在PWM模式下工作(参见表86)。通过向PWMCLRI

寄存器写入任意值，可以清除PWM触发中断。请注意，使用PWM触发中断时，在退出中断服务程序前，将清除PWM中断。这样可以防止同时产生多个中断。

**表86. PWM输出选择**

PWMCON1寄存器 <sup>1</sup>				PWM输出 <sup>2</sup>			
ENA	HOFF	POINV	DIR	PWM0	PWM1	PWM2	PWM3
0	0	X	X	1	1	1	1
X	1	X	X	1	0	1	0
1	0	0	0	0	0	HS1	LS1
1	0	0	1	HS1	LS1	0	0
1	0	1	0	HS1	LS1	1	1
1	0	1	1	1	1	HS1	LS1

<sup>1</sup> X表示无关。

<sup>2</sup> HS = 高端；LS = 低端。

**表87. 比较寄存器**

名称	地址	默认值	访问类型
PWM0COM0	0xFFFF0F84	0x0000	R/W
PWM0COM1	0xFFFF0F88	0x0000	R/W
PWM0COM2	0xFFFF0F8C	0x0000	R/W
PWM1COM0	0xFFFF0F94	0x0000	R/W
PWM1COM1	0xFFFF0F98	0x0000	R/W
PWM1COM2	0xFFFF0F9C	0x0000	R/W
PWM2COM0	0xFFFF0FA4	0x0000	R/W
PWM2COM1	0xFFFF0FA8	0x0000	R/W

**PWM0COM0比较寄存器**

名称： PWM0COM0  
 地址： 0xFFFF0F84  
 默认值： 0x0000  
 访问类型： 读/写  
 功能： 当PWM定时器计数达到该寄存器所保存的计数值时，PWM0输出引脚变为高电平。

**PWM0COM1比较寄存器**

名称： PWM0COM1  
 地址： 0xFFFF0F88  
 默认值： 0x0000  
 访问类型： 读/写  
 功能： 当PWM定时器计数达到该寄存器所保存的计数值时，PWM0输出引脚变为低电平。

**PWM0COM2比较寄存器**

名称： PWM0COM2  
 地址： 0xFFFF0F8C  
 默认值： 0x0000  
 访问类型： 读/写  
 功能： 当PWM定时器计数达到该寄存器所保存的计数值时，PWM1输出引脚变为低电平。

**PWM0LEN寄存器**

名称： PWM0LEN  
 地址： 0xFFFF0F90  
 默认值： 0x0000  
 访问类型： 读/写  
 功能： 当PWM定时器计数达到该寄存器所保存的计数值时，PWM1输出引脚变为高电平。

**PWM1COM0比较寄存器**

名称： PWM1COM0  
 地址： 0xFFFF0F94  
 默认值： 0x0000  
 访问类型： 读/写  
 功能： 当PWM定时器计数达到该寄存器所保存的计数值时，PWM2输出引脚变为高电平。

**PWM1COM1比较寄存器**

名称： PWM1COM1  
 地址： 0xFFFF0F98  
 默认值： 0x0000  
 访问类型： 读/写  
 功能： 当PWM定时器计数达到该寄存器所保存的计数值时，PWM2输出引脚变为低电平。

**PWM1COM2比较寄存器**

名称： PWM1COM2  
 地址： 0xFFFF0F9C  
 默认值： 0x0000  
 访问类型： 读/写  
 功能： 当PWM定时器计数达到该寄存器所保存的计数值时，PWM3输出引脚变为低电平。

**PWM1LEN寄存器**

名称： PWM1LEN  
 地址： 0xFFFF0FA0  
 默认值： 0x0000  
 访问类型： 读/写  
 功能： 当PWM定时器计数达到该寄存器所保存的计数值时，PWM3输出引脚变为高电平。

# ADuC7023

## PWM2COM0比较寄存器

名称: PWM2COM0  
地址: 0xFFFF0FA4  
默认值: 0x0000  
访问类型: 读/写  
功能: 当PWM定时器计数达到该寄存器所保存的计数值时, PWM4输出引脚变为高电平。

## PWM2COM1比较寄存器

名称: PWM2COM1  
地址: 0xFFFF0FA8  
默认值: 0x0000  
访问类型: 读/写  
功能: 当PWM定时器计数达到该寄存器所保存的计数值时, PWM4输出引脚变为低电平。

## PWMCLRI寄存器

名称: PWMCLRI  
地址: 0xFFFF0FB8  
默认值: 0x0000  
访问类型: 写入  
功能: 向其中写入任意值就可以清除PWM中断源。在退出PWM中断服务程序前, 必须对该寄存器进行写操作; 否则, 将同时产生多个中断。

## 处理器相关外设

### 中断系统

ADuC7023拥有由中断控制器控制的22个中断源。大多数中断都是由片内外设产生，如ADC。四个额外的中断源由外部中断请求引脚IRQ0、IRQ1、IRQ2和IRQ3产生。ARM7TDMI CPU内核只能识别以下两种中断：正常中断请求IRQ或快速中断请求FIQ。所有中断都可以被单独屏蔽。

通过9个与中断相关的寄存器来管理中断系统的控制和配置，其中4个用于控制IRQ，4个用于控制FIQ。还有一个MMR用于选择编程中断源。每个IRQ和FIQ寄存器中的控制位都代表相同的中断源，如表88所示。

ADuC7023提供一个矢量中断控制器(VIC)，用于支持中断嵌套(最高可达8级嵌套)。此外，VIC还可以让编程人员为所有中断源指定优先级。通过设置IRQCONN寄存器的ENIRQN位，可以使能中断嵌套功能。当整个矢量中断控制器被使能后，需额外用到大量寄存器。

内核开始执行中断服务程序(ISR)后，应立即保存IRQSTA/FIQSTA，以确保能够响应所有有效中断源。

**表88. IRQ/FIQ寄存器位功能描述**

位	描述
0	所有中断的逻辑或(限FIQ)。
1	SWI.
2	定时器0。
3	定时器1。
4	看门狗定时器(定时器2)。
5	闪存控制。
6	ADC通道。
7	PLL锁定。
8	I <sup>2</sup> C0主机。
9	I <sup>2</sup> C0从机。
10	I <sup>2</sup> C1主机。
11	I <sup>2</sup> C1从机。
12	SPI
13	外部IRQ0。
14	比较器。
15	PSM.
16	外部IRQ1。
17	PLA IRQ
18	外部IRQ2。
19	外部IRQ3。
20	PLA IRQ
21	PWM.

### IRQ

中断请求(IRQ)是进入处理器IRQ模式的一个异常信号。它用于内、外部事件的通用中断服务。

器件内有4个32位寄存器专门用于IRQ，包括：IRQSTA、IRQSIG、IRQEN和IRQCLR。

#### IRQSTA寄存器

名称： IRQSTA

地址： 0xFFFF0000

默认值： 0x00000000

访问类型： 读取

功能： IRQSTA(只读寄存器)提供当前使能的IRQ源状态。当置1时，这个源将向ARM7TDMI内核发出一个有效的IRQ中断请求。没有优先级编码器和中断矢量产生。该功能可以在软件中通过一个普通的中断处理程序实现。所有的32个位经过逻辑“或”运算后，形成要发送给ARM7TDMI内核的IRQ信号。

#### IRQSIG寄存器

名称： IRQSIG

地址： 0xFFFF0004

默认值： 0x00XXX000

访问类型： 读取

功能： IRQSIG反映不同IRQ源的状态。如果一个外设产生了一个IRQ信号，IRQSIG中相应的位就会被置1；否则就会被清0。当特定外设的中断请求取消时，IRQSIG的位就会被清0。通过设置IRQEN寄存器，可屏蔽所有IRQ中断源。IRQSIG为只读寄存器。



## IRQEN寄存器

名称:	IRQEN
地址:	0xFFFF0008
默认值:	0x00000000
访问类型:	读/写
功能:	<p>IRQEN提供当前使能屏蔽的值。将该寄存器的某一位置1, 可使能相应的中断请求, 此时将产生IRQ异常。将某一位清0, 可禁用或屏蔽相应的中断请求, 此时将无法产生IRQ异常。</p> <p>若要取消一个已经使能的中断源, 用户必须将IRQCLR中相应的位置1。将一个中断在IRQEN中相应的位置0不会禁用这个中断。</p>

## IRQCLR寄存器

名称:	IRQCLR
地址:	0xFFFF000C
默认值:	0x00000000
访问类型:	写入
功能:	<p>IRQCLR(只写寄存器)用于清除IRQEN寄存器的相应位, 以屏蔽中断源。将该寄存器的某一位置1, 会清除IRQEN寄存器的相应位(但不影响其他位)。寄存器IRQEN和IRQCLR配合使用, 可以实现独立的使能屏蔽功能, 而无需执行原子性读-改-写操作。</p>

## 快速中断请求(FIQ)

快速中断请求(IRQ)是进入处理器FIQ模式的一个异常信号。提供此信号的目的是以低延迟处理数据传输或通信通道任务。FIQ接口与IRQ接口相同, 但它提供二级中断(最高优先级)。器件内有4个32位寄存器专门用于FIQ, 包括: FIQSIG、FIQEN、FIQCLR和FIQSTA。

FIQSTA的位31至位1通过逻辑“或”运算产生FIQ信号到内核以及FIQ和IRQ寄存器的位0(FIQ源)。

逻辑上FIQEN和FIQCLR不允许一个中断源同时使能IRQ和FIQ屏蔽。FIQEN中的某一位被置1会导致IRQEN中的同一位被清0。同样, IRQEN中的某一位被置1会导致FIQEN中的同一位被清0。一个中断源可以被IRQEN屏蔽和FIQEN屏蔽禁用。

## FIQSIG

FIQSIG反映不同FIQ源的状态。如果一个外设产生了一个FIQ信号, FIQSIG中相应的位就会被置1; 否则就会被清0。当特定外设的中断请求取消时, FIQSIG的位就会被清0。通过设置FIQEN 寄存器, 可屏蔽所有FIQ中断源。FIQSIG为只读寄存器。

## FIQSIG寄存器

名称:	FIQSIG
地址:	0xFFFF0104
默认值:	0x00000000
访问类型:	只读

## FIQEN

FIQEN提供了当前使能屏蔽值。将该寄存器的某一位置1, 可使能相应的中断请求, 此时将产生FIQ异常。当某一位被清0时, 相应的中断源就会被禁止或屏蔽, 此时将无法产生FIQ异常。FIQEN寄存器无法用来禁用中断。

## FIQEN寄存器

名称:	FIQEN
地址:	0xFFFF0108
默认值:	0x00000000
访问类型:	读/写

## FIQCLR

FIQCLR是一个只写寄存器, 可清除FIQEN寄存器的相应位, 从而屏蔽相应的中断源。如将该寄存器的某一位置1, 会清除FIQEN寄存器的相应位(但不影响其它位)。寄存器FIQEN与FIQCLR配合使用, 可以实现独立的使能屏蔽功能, 而无需执行原子性读-改-写操作。

只有当中断源位于中断服务程序中, 或者外设被其自身控制寄存器临时禁用时, 才应使用该寄存器来禁用该中断源。

如果IRQ源具有未决中断或可能具有未决中断, 则不应使用该寄存器来禁用该IRQ源。

## FIQCLR寄存器

名称:	FIQCLR
地址:	0xFFFF010C
默认值:	0x00000000
访问类型:	只写

## FIQSTA

FIQSTA是一个只读寄存器，提供当前使能的FIQ源的状态(FIQSIG和FIQEN对应位进行逻辑“与”操作)。当置1时，这个源将向ARM7TDMI内核发出一个有效的FIQ中断请求。没有优先级编码器和中断矢量产生。该功能可以在软件中通过一个普通的中断处理程序实现。

### FIQSTA寄存器

名称：	FIQSTA
地址：	0xFFFF0100
默认值：	0x00000000
访问类型：	只读

### 可编程中断

由于可编程中断是无法屏蔽的，因此，它们由另外一个寄存器(SWICFG)来控制，通过这个寄存器可以同时写入IRQSTA和IRQSIG寄存器和/或FIQSTA和FIQSIG寄存器。

专用于设置软件中断的32位寄存器为SWICFG，见表89。该寄存器允许控制可编程源中断。

表89. SWICFG MMR位分配

位	描述
31至3	保留。
2	可编程中断FIQ。通过将该位置1或清0，可将FIQSTA和FIQSIG寄存器的位1置1或清0。
1	可编程中断FIQ。通过将该位置1或清0，可将FIQSTA和FIQSIG寄存器的位1置1或清0。
0	保留。

注意，任何中断信号的有效时间不得少于中断延迟时间，这样才能保证中断信号能够被中断控制器检测到或者被用户在IRQSTA和FIQSTA寄存器中检测到。

## 矢量中断控制器(VIC)

ADuC7023集成一个增强的中断控制系统或矢量中断控制器。通过设置IRQCONN寄存器的位0，可以启用针对IRQ中断源的矢量中断控制器。同样，通过设置IRQCONN寄存器的位1，可以启用针对FIQ中断源的矢量中断控制器。矢量中断控制器在以下几个方面增强了标准IRQ/FIQ中断：

- 矢量中断允许用户为每个中断源单独定义中断服务程序地址。这可以通过IRQBASE和IRQVEC寄存器来完成。
- IRQ/FIQ中断可根据优先级进行嵌套，最多允许8级嵌套。FIQ中断的优先级高于IRQ中断。因此，当使能FIQ和IRQ的矢量中断控制器且优先级为最高时，将有可能形成16个不同的中断级。
- 通过设置IRQP0至IRQP2寄存器，可以分配可编程中断优先级，优先级可为第0至7级。

### VIC寄存器

#### IRQBASE寄存器

IRQBASE(矢量基址寄存器)用于指向存储32位指针地址的存储器的起始地址。这些指针地址是各个终端服务程序的地址。

名称：	IRQBASE
地址：	0xFFFF0014
默认值：	0x00000000
访问类型：	读/写

表90. IRQBASE寄存器位功能描述

位	类型	初始值	描述
31:16	只读	保留	读数始终为0
15:0	R/W	0	矢量基地址。

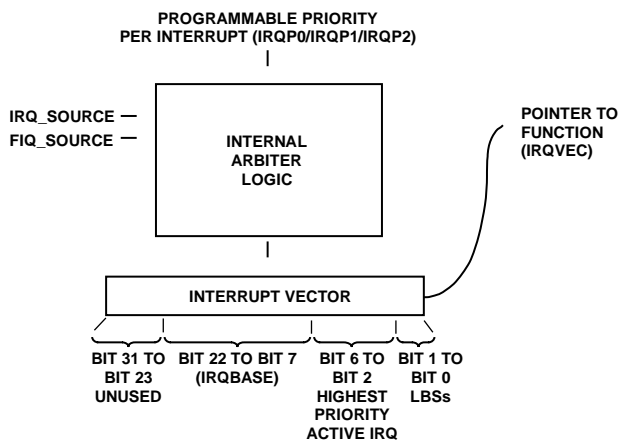


图41. 中断结构

# ADuC7023

## IRQVEC寄存器

IRQVEC(IRQ中断矢量寄存器)指向包含指向一个存储地址，该地址包含当前活动IRQ的中断服务程序的指针。当产生IRQ中断，且已通过设置IRQCONN的位0启用IRQ中断嵌套功能后，该寄存器为只读的。

## IRQVEC寄存器

名称： IRQVEC  
地址： 0xFFFF001C  
默认值： 0x00000000  
访问类型： 只读

表91. IRQVEC寄存器位功能描述

位	类型	初始值	描述
31至23	只读	0	读数始终为0。
22至7	R/W	0	IRQBASE寄存器值。
6至2	只读	0	最高优先级中断源。这些位存储的数据对应0至21其中的一个数字，代表可能的中断源。例如，如果当前活动的最高优先级IRQ是定时器2，那么这些位的值为[00100]。
1至0	保留	0	保留位。

## 优先级寄存器

IRQVEC(IRQ中断矢量寄存器)指向包含指向一个存储地址，该地址包含当前活动IRQ的中断服务程序的指针。当产生IRQ中断，且已通过设置IRQCONN的位0启用IRQ中断嵌套功能后，该寄存器为只读的。

**IRQP0寄存器**

名称： IRQP0  
地址： 0xFFFF0020  
默认值： 0x00000000  
访问类型： 读/写

**表92. IRQP0寄存器位功能描述**

位	名称	描述
31	保留	保留位。
30至28	PLLPI	为PLL锁定中断设置中断优先级(0至7)。
27	保留	保留位。
26至24	ADCPI	为ADC中断源设置中断优先级(0至7)。
23	保留	保留位。
22至20	FlashPI	为闪存控制器中断源设置中断优先级(0至7)。
19	保留	保留位。
18至16	T2PI	为定时器2设置中断优先级(0至7)。
15	保留	保留位。
14至12	T1PI	为定时器1设置中断优先级(0至7)。
11	保留	保留位。
10至8	TOPI	为定时器0设置中断优先级(0至7)。
7	保留	保留位。
6至4	SWINTP	为软件中断源设置中断优先级(0至7)。
3至0	保留	中断0不能设置优先级。

**IRQP1寄存器**

名称： IRQP1  
地址： 0xFFFF0024  
默认值： 0x00000000  
访问类型： 读/写

**表93. IRQP1寄存器位功能描述**

位	名称	描述
31	保留	保留位。
30至28	PSMPI	为电源监控器中断源设置中断优先级(0至7)。
27	保留	保留位。
26至24	COMPI	为比较器设置中断优先级(0至7)。
23	保留	保留位。
22至20	IRQ0PI	为IRQ0设置中断优先级(0至7)。
19	保留	保留位。

位	名称	描述
18至16	SPIPI	为SPI设置中断优先级(0至7)。
15	保留	保留位。
14至12	I2C1SPI	为I <sup>2</sup> C1从机设置中断优先级(0至7)。
11	保留	保留位。
10至8	I2C1MPI	为I <sup>2</sup> C1主机设置中断优先级(0至7)。
7	保留	保留位。
6至4	I2C0SPI	为I <sup>2</sup> C0从机设置中断优先级(0至7)。
3	保留	保留位。
2至0	I2C0MPI	为I <sup>2</sup> C0主机设置中断优先级(0至7)。

**IRQP2寄存器**

名称： IRQP2  
地址： 0xFFFF0028  
默认值： 0x00000000  
访问类型： 读/写

**表94. IRQP2寄存器位功能描述**

位	名称	描述
31至23	保留	保留位。
22至20	PWMPI	为PWM设置中断优先级(0至7)。
19	保留	保留位。
18至16	PLA1PI	为PLA IRQ1设置中断优先级(0至7)。
15	保留	保留位。
14至12	IRQ3PI	为IRQ3设置中断优先级(0至7)。
11	保留	保留位。
10至8	IRQ2PI	为IRQ2设置中断优先级(0至7)。
7	保留	保留位。
6至4	PLA0PI	为PLA IRQ0设置中断优先级(0至7)。
3	保留	保留位。
2至0	IRQ1PI	为IRQ1设置中断优先级(0至7)。

# ADuC7023

## IRQCONN寄存器

IRQCONN寄存器为IRQ和FIQ控制寄存器，它包含两个有效位。第一位用于使能IRQ中断的嵌套和优先级设置，而第二位用于使能FIQ中断的嵌套和优先级设置。

如果将这两个位清0，则FIQ和IRQ可能仍然可用，但无法嵌套IRQ或FIQ。此外，也无法设置中断源的中断优先级。在此默认状态下，FIQ的优先级高于IRQ的优先级。

名称： IRQCONN

地址： 0xFFFF0030

默认值： 0x00000000

访问类型： 读/写

**表95. IRQCONN寄存器位功能描述**

位	名称	描述
31至2	保留	这些位为保留位，不允许向其中写入数据。
1	ENFIQN	通过将该位置1，可使能FIQ中断嵌套。若将该位清0，则不能设置FIQ中断嵌套及优先级。
0	ENIRQN	通过将该位置1，可使能IRQ中断嵌套。若将该位清0，则不能设置IRQ中断嵌套及优先级。

## IRQSTAN寄存器

如果IRQCONN位0被断言且IRQVEC被读取，则这些位中的一位将被断言。IRQ的优先级决定了第几位将被置位。如果IRQ的中断优先级为0，则位0被断言；如果IRQ的中断优先级为1，则位1被断言；以此类推。将该寄存器中的某一位置1，可屏蔽所有与该位所对应的中断具有相同或更低优先级的中断。

如需清0该寄存器中的某一位，必须先将其它与更高优先级对应的位清0。一次只能清0一位。例如，如果该寄存器的值为0x09，第一次写0xFF可以将该寄存器的值改为0x08，再一次写0xFF可以将该寄存器的值改为0x00。

名称： IRQSTAN

地址： 0xFFFF003C

默认值： 0x00000000

访问类型： 读/写

**表96. IRQSTAN寄存器位功能描述**

位	名称	描述
31至8	保留	这些位为保留位，不允许向其中写入数据。
7至0		通过将该位置1，可使能FIQ中断嵌套。若将该位清0，则不能设置FIQ中断嵌套及优先级。

## FIQVEC寄存器

FIQVEC(FIQ中断矢量寄存器)指向包含指向一个存储地址，该地址包含当前活动FIQ的中断服务程序的指针。当产生FIQ中断，且已通过设置IRQCONN的位1启用FIQ中断嵌套功能后，该寄存器为只读的。

名称： FIQVEC

地址： 0xFFFF011C

默认值： 0x00000000

访问类型： 只读

**表97. FIQVEC寄存器位功能描述**

位	类型	初始值	描述
31至23	只读	0	读数始终为0。
22至7	R/W	0	IRQBASE寄存器值。
6至2		0	最高优先级中断源。这些位存储的数据对应0至27其中的一个数字，代表可能的中断源。例如，如果当前活动的最高优先级FIQ是定时器2，那么这些位的值为[00100]。
1至0	保留	0	保留位。

## FIQSTAN寄存器

如果IRQCONN位1被断言且FIQVEC被读取，则这些位中的一位将被断言。FIQ的优先级决定了第几位将被置位。如果FIQ的中断优先级为0，则位0被断言；如果FIQ的中断优先级为1，则位1被断言；以此类推。

将该寄存器中的某一位置1，可屏蔽所有与该位所对应的中断具有相同或更低优先级的中断。

如需清0该寄存器中的某一位，必须先将其它与更高优先级对应的位清0。一次只能清0一位。例如，如果该寄存器的值为0x09，第一次写0xFF可以将该寄存器的值改为0x08，再一次写0xFF可以将该寄存器的值改为0x00。

名称： FIQSTAN  
地址： 0xFFFFF013C  
默认值： 0x00000000  
访问类型： 读/写

**表98. FIQSTAN寄存器位功能描述**

位	名称	描述
31至8	保留	这些位为保留位，不允许向其中写入数据。
7至0		通过将该位置1，可使能FIQ中断嵌套。若将该位清0，则不能设置FIQ中断嵌套及优先级。

#### 外部中断和PLA中断

ADuC7023最多可支持4个外部中断源和2个PLA中断源。这些外部中断源可单独配置为电平触发、上升沿/下降沿触发。

要使能外部中断源或PLA中断源，首先必须将FIQEN或IRQEN寄存器的适当位置1。而要选择所需的边沿或电平触发，则必须正确配置IRQCONE寄存器。

通过设置IRQCLRE中的适当位，可以清除边沿触发的外部IRQ中断或边沿触发的PLA中断。

#### IRQCONE寄存器

名称： IRQCONE  
地址： 0xFFFFF0034  
默认值： 0x00000000  
访问类型： 读/写

**表99. IRQCONE寄存器位功能描述**

位	值	名称	描述
31至12		保留	这些位为保留位，不允许向其中写入数据。
11至10	11 10 01 00	PLA1SRC[1:0]	PLA IRQ1在下降沿触发。 PLA IRQ1在上升沿触发。 PLA IRQ1在低电平时触发。 PLA IRQ1在高电平时触发。

位	值	名称	描述
9至8	11 10 01 00	IRQ3SRC[1:0]	外部IRQ3在下降沿触发。 外部IRQ3在上升沿触发。 外部IRQ3在低电平时触发。 外部IRQ3在高电平时触发。
7至6	11 10 01 00	IRQ2SRC[1:0]	外部IRQ2在下降沿触发。 外部IRQ2在上升沿触发。 外部IRQ2在低电平时触发。 外部IRQ2在高电平时触发。
5至4	11 10 01 00	PLA0SRC[1:0]	PLA IRQ0在下降沿触发。 PLA IRQ0在上升沿触发。 PLA IRQ0在低电平时触发。 PLA IRQ0在高电平时触发。
3至2	11 10 01 00	IRQ1SRC[1:0]	外部IRQ1在下降沿触发。 外部IRQ1在上升沿触发。 外部IRQ1在低电平时触发。 外部IRQ1在高电平时触发。
1至0	11 10 01 00	IRQ0SRC[1:0]	外部IRQ0在下降沿触发。 外部IRQ0在上升沿触发。 外部IRQ0在低电平时触发。 外部IRQ0在高电平时触发。

#### IRQCLRE寄存器

名称： IRQCLRE  
地址： 0xFFFFF0038  
默认值： 0x00000000  
访问类型： 读/写



**表100. IRQCLRE寄存器位功能描述**

位	名称	描述
31至21	保留	这些位为保留位，不允许向其中写入数据。
20	PLA1CLRI	必须在PLA IRQ1中断服务程序中在该位置1，以清除边沿触发的PLA IRQ1中断。
19	IRQ3CLRI	必须在外部IRQ3中断服务程序中在该位置1，以清除边沿触发的IRQ3中断。
18	IRQ2CLRI	必须在外部IRQ2中断服务程序中在该位置1，以清除边沿触发的IRQ2中断。
17	PLA0CLRI	必须在PLA IRQ0中断服务程序中在该位置1，以清除边沿触发的PLA IRQ0中断。
16	IRQ1CLRI	必须在外部IRQ1中断服务程序中在该位置1，以清除边沿触发的IRQ1中断。
15至14	保留	这些位为保留位，不允许向其中写入数据。
13	IRQ0CLRI	必须在外部IRQ0中断服务程序中在该位置1，以清除边沿触发的IRQ0中断。
12至0	保留	这些位为保留位，不允许向其中写入数据。

## 定时器

ADuC7023具有3个通用定时器/计数器：定时器0、定时器1和定时器2(或看门狗定时器)。

这3个定时器可以在自由模式或周期模式下工作。

在自由模式下，计数器将从最大值递减计数一直到零，并且在达到最小值后重新递增计数。(它也可以从最小值开始递增计数直到满量程并且在达到最大值后重新开始递减计数。)

在周期模式下，计数器以装载寄存器(TxLD寄存器)中的值为起始值，开始递减/递增计数至0或满量程，然后再以该寄存器中的值为起始值，重新开始计数。

定时器间隔可通过下式计算：

如果将定时器设置为递减计数，那么

$$Interval = \frac{(TxLD) \times Prescaler}{Source\ Clock}$$

如果将定时器设置为递增计数，那么

$$Interval = \frac{(FullScale - TxLD) \times Prescaler}{Source\ Clock}$$

通过访问计数器的值寄存器(TxVAL)，可以随时读出计数器的值。当定时器的时钟源是内核时钟以外的时钟时，可能由于时钟系统不同步而读出错误的值。在这种情况下，TxVAL始终应该读两次。如果两次读到的结果不同，应该再读一次以得到正确的值。

通过对某一定时器的控制寄存器(TxCON)进行写操作，可以启动相应的定时器。

在正常模式下，当递减计数时，每一次计数器的值达到0时就会产生一个IRQ中断。当递增计数时，每一次计数器的值达到满量程值时也会产生一个IRQ中断。向某一定时器(TxCLRI)的清除寄存器内写入任一数据，可以清除IRQ中断。

当使用异步的时钟定时器时，取消定时器模块内的中断所占用的时间要比执行中断程序中的代码所占用的时间长。在退出中断服务程序之前，要保证中断信号已经被取消。这可以通过检查IRQSTA寄存器来实现。

## 小时、分钟、秒和1/128格式

若要使用小时、分钟、秒和百分数格式，应选择32768 kHz时钟和预分频值256。百位数字段不代表毫秒，而是一秒的1/128 (256/32,768)。代表时、分、秒的位在寄存器中不是连续的。使用T1CON[5:4]设置的“时:分:秒:百位数”格式时，这一配置适用于T1LD和T1VAL。详情参见表101。

**表101. 小时、分钟、秒和百分数格式**

位	值	描述
31:24	0至23或0至255	小时
23:22	0	保留
21:16	0至59	分钟
15:14	0	保留
13:8	0至59	秒
7	0	保留
6:0	0至127	1/128秒

## 定时器0(RTOS定时器)

定时器0是一个带有可编程预分频器的16位通用定时器，工作时递减计数(如图42所示)。预分频器源采用内核时钟频率(HCLK)，分频系数共有1、16和256三种。

定时器0可用于启动ADC转换，如框图42所示。

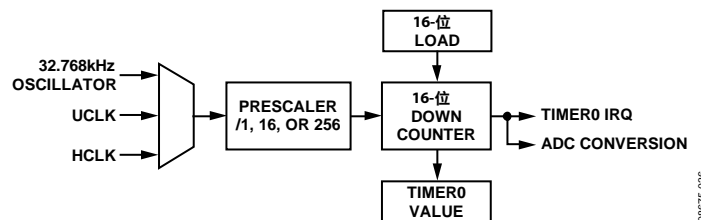


表42. 定时器0框图



定时器0接口包括4个寄存器：T0LD、T0VAL、T0CON和T0CLRI。

### T0LD寄存器

名称： T0LD  
地址： 0xFFFF0300  
默认值： 0x0000  
访问类型： 读/写

T0LD是一个16位加载寄存器，用于保存载入计数器的16位值。

### T0VAL寄存器

名称： T0VAL  
地址： 0xFFFF0304  
默认值： 0xFFFF  
访问类型： 读取

T0VAL是一个16位只读寄存器，用于表示计数器的当前状态。

### T0CON寄存器

名称： T0CON  
地址： 0xFFFF0308  
默认值： 0x0000  
访问类型： R/W

T0CON是一个配置寄存器，如表102所示。

表102. T0CON寄存器位功能描述

位	值	描述
15至8		保留。
7		定时器0使能位。 通过将该位置1，用户可使能定时器0。 通过将该位清0，用户可禁用定时器0。
6		定时器0模式。 通过将该位置1，用户可选择周期模式。 通过将该位清0，用户可选择自由模式。 默认模式。
5至4	00 01 10 11	时钟选择位。 HCLK。 UCLK。 内部32,768 Hz振荡器。 保留。
3至2	00 01 10 11	时钟源/1。默认值。 时钟源/16。 时钟源/256。 未定义。与00时相同。
1至0		保留。

### T0CLRI寄存器

名称： T0CLRI  
地址： 0xFFFF030C  
默认值： 0xXX  
访问类型： 写入

T0CLRI是一个8位寄存器。向其中写入任意值就可以清除中断。

### 定时器1(通用定时器)

定时器1是一个带有可编程预分频器的32位通用定时器，工作时可递增计数或递减计数。时钟源可以是32 kHz外部晶振、未分频系统、内核时钟或P1.1引脚(最高频率44 MHz)。时钟源分频系数共有1、16、256或32,768四种。

计数器可以采用标准的32位数格式或者时间格式(小时：分钟：秒：百分之一秒)。

定时器1有一个事件捕获寄存器(T1CAP)，它可以被选定的IRQ中断源初始置位所触发。这一特性可用来判断一个事件的置位，当用于IRQ中断请求服务时，这种方法比RTOS定时器所允许的精度更高。

定时器1可用于启动ADC转换，如框图43所示。

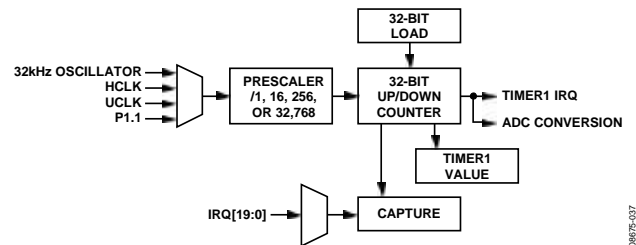


图43. 定时器1框图

定时器1接口有5个寄存器：T1LD、T1VAL、T1CON、T1CLRI和T1CAP。

### T1LD寄存器

名称： T1LD  
地址： 0xFFFF0320  
默认值： 0x00000000  
访问类型： 读/写

T1LD是一个32位加载寄存器，用于保存载入计数器的32位值。

# ADuC7023

## T1VAL寄存器

名称: T1VAL  
地址: 0xFFFF0324  
默认值: 0xFFFFFFFF  
访问类型: 读取

T1VAL是一个32位只读寄存器，用于表示计数器的当前状态。

## T1CON寄存器

名称: T1CON  
地址: 0xFFFF0328  
默认值: 0x00000000  
访问类型: 读/写

T1CON是一个配置寄存器，如表103所示。

**表103. T1CON寄存器位功能描述**

位	值	描述
31至18		保留。
17		事件选择位。通过将该位置1，用户可使能对事件的实时捕捉。通过将该位清0，用户可禁用对事件的实时捕捉。
16至12		事件选择范围0至31。这些事件如表88所示，其中所有事件的序号对于定时器1都相应减少两位。例如，表88中的事件2对于定时器1来说就是事件0。
11至9	000 001 010 011	时钟选择。 内核时钟(HCLK)。 内部32.768 kHz晶振。 UCLK P1.1上升沿触发。
8		递增计数。通过将该位置1，用户可让定时器1递增计数。通过将该位清0，用户可让定时器1递减计数(默认)。
7		定时器1使能位。通过将该位置1，用户可使能定时器1。通过将该位清0，用户可禁用定时器1。
6		定时器1模式。通过将该位置1，用户可选择周期模式。通过将该位清0，用户可选择自由模式。默认模式。
5至4	00 01 10 11	格式。 二进制。 保留。 小时: 分钟: 秒: 百分之一秒 (23小时至0小时)。 小时: 分钟: 秒: 百分之一秒 (255小时至0小时)。

位	值	描述
3至0	0000 0100 1000 1111	预分频选择位。 时钟源/1。 时钟源/16。 时钟源/256。 时钟源/32,768。

## T1CLRI寄存器

名称: T1CLRI  
地址: 0xFFFF032C  
默认值: 0xXX  
访问类型: 写入

T1CLRI是一个8位寄存器。向其中写入任意值，可以清除定时器1中断。

## T1CAP寄存器

名称: T1CAP  
地址: 0xFFFF0330  
默认值: 0x00000000  
访问类型: 读取

T1CAP是一个32位寄存器。当一个特殊事件发生时，它会保持T1VAL寄存器中当时的数值。这个事件必须是在T1CON中已经选择的。

## 定时器2(看门狗定时器)

定时器2共有两种工作模式：正常模式和看门狗模式。看门狗定时器用于强制处理器从非法软件状态恢复到正常工作状态。看门狗定时器使能后，需要周期性服务来阻止自身强制处理器执行复位操作。

### 正常模式

在正常模式下，除了时钟源和递增计数功能，定时器2和定时器0是完全相同的。时钟源来自于锁相环(32KHz)，其时钟分频系数共有1、16、256三种(如图44所示)。

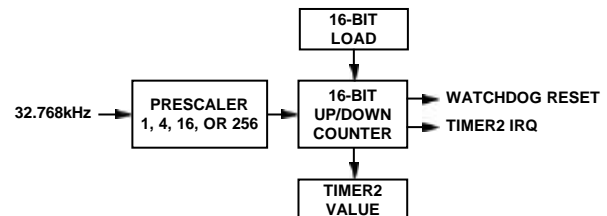


图44. 定时器2框图

08675-038

### 看门狗模式

通过将T2CON寄存器的位5置1，可进入看门狗模式。定时器2以T2LD寄存器中的数值为起始值开始递减计数，一直到0为止。T2LD用作超时定时器。当使用1/256预分频时，最大的超时时间为512秒，且为T2LD的满量程。在看门狗模式下工作时，定时器3的时钟源为内部的32KHz晶振。为了成功进入看门狗模式，必须在写入T2LD寄存器以后再将T2CON寄存器的位5置1。

如果定时器计数到0，就会产生一个复位或中断，具体取决于T2CON寄存器中位1的值。为避免产生复位或中断，必须在定时周期结束之前向T2CLR1中写入任意一个值。此操作可将T2LD的值重新载入计数器内，且重新开始一个新的超时周期。

一旦进入看门狗模式，T2LD和T2CON就会被写保护。此时这两个寄存器不能被修改，直到有一个复位信号清除了看门狗使能位，这将使定时器2退出看门狗模式。

定时器2接口有4个寄存器：T2LD、T2VAL、T2CON和T2CLR1。

### T2LD寄存器

名称： T2LD

名称： 0xFFFF0360

默认值： 0x0000

访问类型：读/写

T2LD是一个16位加载寄存器，用于保存载入计数器的16位值。

### T2VAL寄存器

名称： T2VAL

名称： 0xFFFF0364

默认值： 0xFFFF

访问类型：读取

T2VAL是一个16位只读寄存器，用于表示计数器的当前状态。

### T2CON寄存器

名称： T2CON

名称： 0xFFFF0368

默认值： 0x0000

访问类型：读/写

T2CON是一个配置MMR，如表104所示。

表104. T2CON寄存器位功能描述

位	值	描述
15至9		保留。
8		递增计数。 通过将该位置1，用户可让定时器2递增计数。 通过将该位清0，用户可让定时器2递减计数(默认)。
7		定时器2使能位。 通过将该位置1，用户可启用定时器2。 通过将该位清0，用户可禁用定时器2(默认)。
6		定时器2模式。 通过将该位置1，用户可选择周期模式。 通过将该位清0，用户可选择自由模式。 默认模式。
5		看门狗模式使能位。 通过将该位置1，用户可启用看门狗模式。 通过将该位清0，用户可禁用看门狗模式(默认)。
4		安全清除位。 通过将该位置1，用户可使用安全清除选项。 通过将该位清0，用户可禁用安全清除选项(默认)。
3至2	00 01 10 11	预分频选择位。 时钟源/1(默认)。 时钟源/16。 时钟源/256。 未定义。与00时相同。
1		看门狗IRQ选项位。 通过将该位置1，可在看门狗计数值为0时产生IRQ中断而非复位。 通过将该位清0，用户可禁用IRQ选项。
0		保留。

**T2CLRI寄存器**

名称:	T2CLRI
地址:	0xFFFF036C
默认值:	0xXX
访问类型:	写入

T2CLRI是一个8位寄存器。在正常模式下，向这个寄存器连续写入任意值，将清除定时器2中断；而在看门狗模式下向这个寄存器写入任意值，则重新开始一个超时周期。

用户必须向该寄存器执行连续写入，才能开始一个新的超时周期。

**安全清除位(仅用于看门狗模式)**

安全清除位用于实现更高层次的保护。该位置1时，为避免看门狗复位，必须将一个特殊的数值序列写入T2CLRI。这个特殊的数值序列是由一个8位的线性反馈移位寄存器(LFSR)多项式  $= X^8 + X^6 + X^5 + X + 1$  产生的，如图45所示。

在进入看门狗模式之前，必须先向T2CLRI中写入一个初始值或种子。在进入看门狗模式以后，再次写入到T2CLRI的值必须与期望值相匹配。如果匹配，当计数器被重新载入时，LFSR就会进入下一状态。如果不匹配，即使计数器没有计满，也将立即产生复位。

根据这个多项式的性质，0x00不应该作为初始值种子，因为0x00会一直迫使系统快速复位。此外，LFSR的值不能被访问，且必须在软件中产生和跟踪。

数值序列的示例如下：

1. 将定时器2设为看门狗模式之前，在T2CLRI中写入初始种子0xAA。
2. 在T2CLRI中写入0xAA，定时器2被重载。
3. 在T2CLRI中写入0x37，定时器2被重载。
4. 在T2CLRI中写入0x6E，定时器2被重载。
5. 写入0x66。0xDC是期望值，看门狗将芯片复位。

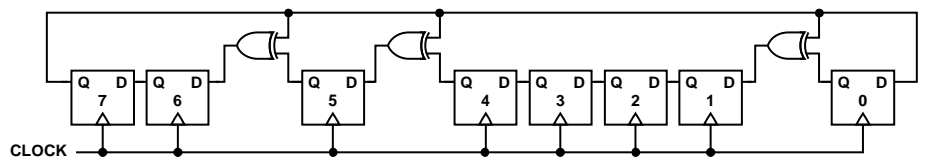


图45. 8位LFSR

## 硬件设计考虑

### 电源

ADuC7023工作电压范围为2.7 V至3.6 V。模拟电源引脚和数字电源引脚(分别对应 $AV_{DD}$ 和 $IOV_{DD}$ )是分离的,因此 $AV_{DD}$ 不受系统 $IOV_{DD}$ 线路上经常出现的干扰数字信号影响。在这种模式下,器件可以在分离电源下工作;也就是说,每个电源可以使用不同的电压。例如,系统的 $IOV_{DD}$ 工作电压为3.3 V,而 $AV_{DD}$ 电压为3 V;反之亦然。图46是一个典型的分离电源配置。

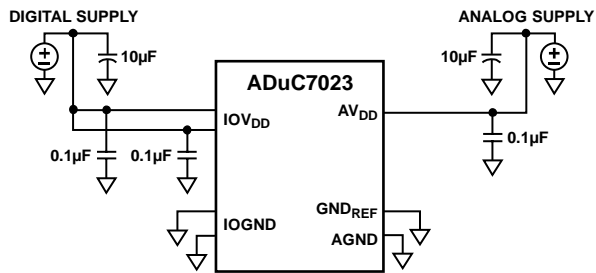


图46. 外部双电源连接

除了使用两个分离的电源外,用户还可以通过在 $AV_{DD}$ 和 $IOV_{DD}$ 之间串联一个小电阻和/或磁珠来降低 $AV_{DD}$ 的噪声,然后将 $AV_{DD}$ 对地单独去耦。图47所示的就是用这种方法进行设计的一个示例。使用这种方法时,其它模拟电路(如运算放大器和基准电压源等)也可以通过 $AV_{DD}$ 供电。

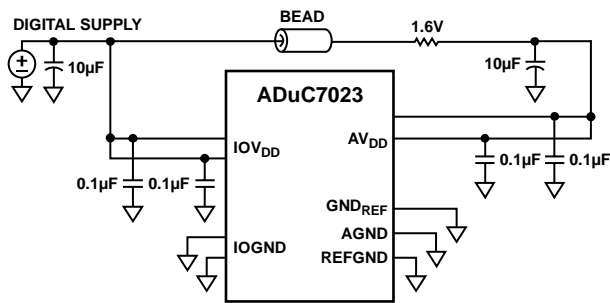


图47. 外部单电源连接

在图46和图47中,在 $IOV_{DD}$ 处有一个大容量值(10  $\mu$ F)的储能电容,而在 $AV_{DD}$ 处则单独有一个10  $\mu$ F的电容。此外,在芯片的每一个 $AV_{DD}$ 和 $IOV_{DD}$ 引脚都连接了一个小容值(0.1  $\mu$ F)电容。根据标准设计规范,设计中应包括所有这些电容并且确保较小的电容接近各 $AV_{DD}$ 引脚,布线长度也应尽量越短越好。这些电容的接地线端直接连接到地平面即可。

最后,无论在任何时候,ADuC7023的模拟和数字接地引脚必须以同一系统接地基准点为基准。

### $IOV_{DD}$ 电源灵敏度

$IOV_{DD}$ 电源对高频噪声很敏感,因为片内振荡器和锁相环电路也是由 $IOV_{DD}$ 供电的。当内部锁相环失锁时,一个门电路会将时钟源与CPU隔离开,并且ARM7TDMI内核会停止执行代码,直到锁相环重新锁定。这个特性可以确保闪存接口时序或ARM7TDMI时序不受干扰。

通常,电源上频率高于50 kHz并且峰峰值为50 mV的噪声会导致内核停止工作。

如果在电源部分推荐的去耦电容不足以保证 $IOV_{DD}$ 上的所有噪声低于50 mV,那么就需要一个图48所推荐的滤波电路。

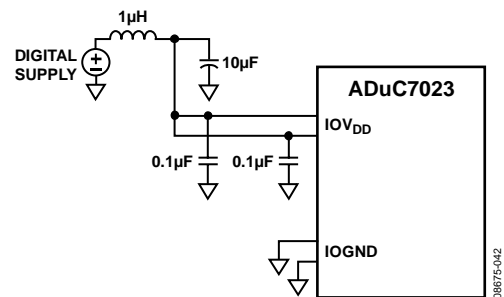


图48. 推荐的 $IOV_{DD}$ 电源滤波电路

### 线性稳压器

ADuC7023要求采用3.3 V单电源供电,而内核逻辑要求采用2.6 V电源供电。片内有一个线性稳压器,可以将来自 $IOV_{DD}$ 的电源调节到2.6 V,从而为内核逻辑供电。 $LV_{DD}$ 引脚的2.6 V电源用来给内核逻辑供电。在 $LV_{DD}$ 和DGND之间必须连接一个0.47  $\mu$ F的补偿电容(应尽量靠近这些引脚)作为电荷槽,如图49所示。

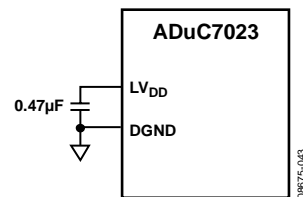


图49. 稳压器连接

$LV_{DD}$ 引脚不能用来给任何其它芯片供电。此外,推荐在 $IOV_{DD}$ 引脚使用性能优良的电源去耦装置,以便提高片内稳压器的线性稳压性能。

# ADuC7023

## 接地和电路板布局建议

和所有高分辨率数据转换器一样，为了实现ADC和DAC的最优性能，对于基于ADuC7023的设计，应特别注意接地和PC电路板布局。

尽管这些器件已经将模拟地和数字地引脚分开(AGND和DGND)，但用户切勿将这些引脚连接到两个分开的地平面，除非这两个地平面非常靠近器件。图50a是一个简化的示意图。在系统中，如果数字地和模拟地平面在某处连接在一起(如系统电源处)，那么这个平面不能再连接到靠近器件的地方，因为这样会形成一个地环路。在这种情况下，ADuC7023的所有AGND和DGND引脚都应连接到模拟地平面，如图50b所示。如果系统中只有一个地平面，必须确保数字和模拟器件在板上是物理分开且分属两个半平面，这样数字回路电流就不会流经模拟电路附近(反之亦然)。

ADuC7023可以放置在数字和模拟部分之间，如图50c所示。

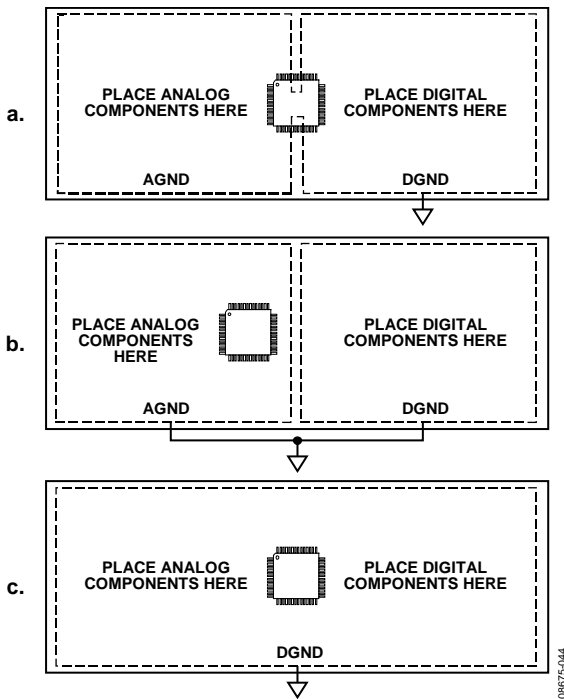


图50. 系统地框图

在上述方案以及更为复杂的实际应用中，用户应该特别注意来自电源的电流和返回地的电流。要确保所有电流的回路路径均尽可能靠近电流到达目的地所经过的路径。

例如，不要用IOV<sub>DD</sub>给模拟部分的器件供电(如图50 b所示)，因为这会导致IOV<sub>DD</sub>回路电流强行通过AGND。如果一个带有噪声的数字芯片被放置在板的左半平面(如图50 c所示)，那么应该避免可能出现的数字电流流经模拟电路。如果可能的话，尽量避免在地平面上出现长的不连续部分，例如由同一层上的一条长迹线构成的不连续部分，因为它们会强迫回路信号通过一个长的路径。此外，所有需要连接到地的引脚应该直接连接到地平面，尽量少用一些或不要用迹线将引脚通过其过孔与地分离。

当ADuC7023的任何数字输入引脚连接高速逻辑信号(上升/下降时间小于5 ns)时，应该在每一条相关的线路上串联一个电阻以确保器件输入引脚上信号上升和下降时间大于5 ns。通常，阻值为100 Ω或200 Ω的电阻足以阻止高速信号从容性器件耦合进入器件并影响ADC的转换精度。

## 时钟振荡器

ADuC7023的时钟源可以由内部锁相环或一个外部时钟输入产生。使用内部锁相环时，应该在XCLKI和XCLKO引脚之间连接一个32.768 kHz的并行谐振晶体并且这两个引脚与地之间应连接一个电容，如图51所示。这个晶体使得锁相环可以正确锁相，进而产生41.78 MHz频率的时钟信号。如果不使用外部晶体，内部振荡器会产生一个41.78 MHz ± 3%的典型频率。

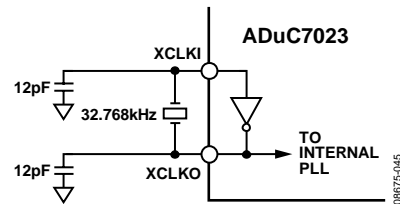


图51. 外部并行谐振晶体连接

若要使用一个外部时钟源输入来代替锁相环(如图52所示)，必须修改PLLCON寄存器的位1和位0。外部时钟从P1.1和XCLK引脚输入。

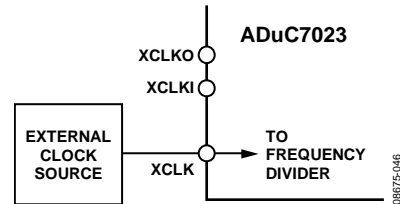


图52. 连接一个外部时钟源

使用外部时钟源时，ADuC7023的额定时钟频率范围为50 kHz到44 MHz ±1%，以确保模拟外设和Flash/EE正常工作。



### 上电复位操作

ADuC7023具有一个内部上电复位(POR)电路，当 $LV_{DD}$ 低于2.40 V典型值时，内部POR会使器件保持在复位状态。当 $LV_{DD}$ 上升超过2.40 V时，一个内部定时器通常会在64 ms后溢出，从而使芯片脱离复位状态。用户此时必须保证给 $IOV_{DD}$ 供电的电源电压至少稳定在2.7 V。当器件关断时，内部POR同样会使器件保持在复位状态，直到 $LV_{DD}$ 降至2.40 V以下。

图53详细说明了内部POR的工作原理。

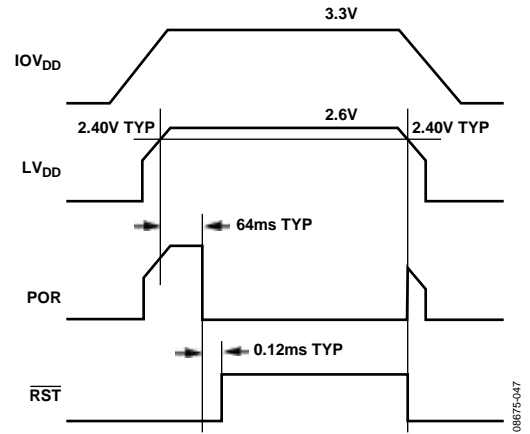


图53. 内部上电复位操作





## 开发工具

### 基于PC工具

ADuC7023系列有四种开发系统。ADuC7023 QuickStart Plus 适用于希望拥有一个全面的硬件开发环境的新用户。

这些系统由以下基于PC(兼容Windows®)的硬件和软件开发工具组成。

### 硬件

硬件系统采用ADuC7023评估板、串口编程电缆和RDI兼容型JTAG仿真器(仅包括在ADuC7023 QuickStart Plus中)。

### 软件

软件系统中集成开发环境，并包含汇编器、编译器和基于JTAG的非介入式调试器。软件系统采用串行下载软件和示例代码。

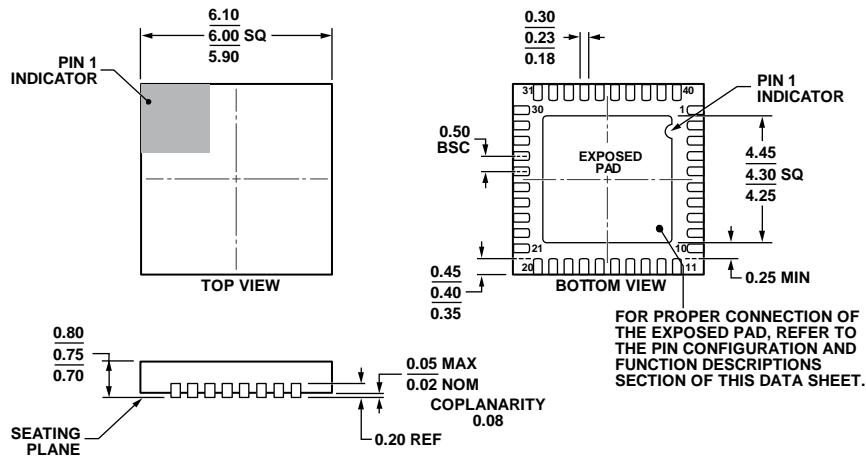
### 其它

其它系统采用CD-ROM文档。

### 在线I<sup>2</sup>C下载器

[www.analog.com](http://www.analog.com)上提供基于I<sup>2</sup>C的串行下载器。该软件要求使用ADI公司提供的USB转I<sup>2</sup>C适配板，后者产品型号为USB-I2C/LIN-CONV-Z。

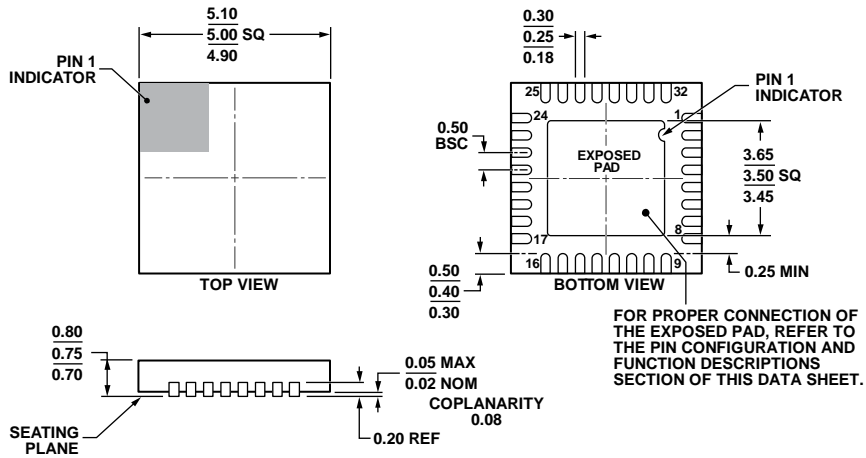
## 外形尺寸



COMPLIANT TO JEDEC STANDARDS MO-220-WJJD.

图55. 40引脚架构芯片级封装 [LFCSP\_WQ]  
6 mm x 6 mm, 超薄体 (CP-40-10)  
图示尺寸单位: mm

111808-A



COMPLIANT TO JEDEC STANDARDS MO-220-WHHD.

图56. 32引脚架构芯片级封装 [LFCSP\_WQ]  
5 mm x 5 mm, 超薄体  
(CP-32-11)  
尺寸单位: mm

112408-A

## 订购指南

型号 <sup>1</sup>	ADC通道	DAC通道	FLASH/RAM	GPIO	下载器	温度范围	封装描述	封装选项	订购数量
ADuC7023BCP6Z62I	12	4	62 kB/8 kB	20	I <sup>2</sup> C	-40°C至+125°C	40引脚 LFCSP_WQ	CP-40-10	490
ADuC7023BCP6Z62IRL	12	4	62 kB/8 kB	20	I <sup>2</sup> C	-40°C至+125°C	40引脚 LFCSP_WQ	CP-40-10	2,500
ADuC7023BCP6Z62IR7	12	4	62 kB/8 kB	20	I <sup>2</sup> C	-40°C至+125°C	40引脚 LFCSP_WQ	CP-40-10	750
ADuC7023BCPZ62I	8	4	62 kB/8 kB	12	I <sup>2</sup> C	-40°C至+125°C	32引脚 LFCSP_WQ	CP-32-11	490
ADuC7023BCPZ62I-RL	8	4	62 kB/8 kB	12	I <sup>2</sup> C	-40°C至+125°C	32引脚 LFCSP_WQ	CP-32-11	5,000
ADuC7023BCPZ62I-R7	8	4	62 kB/8 kB	12	I <sup>2</sup> C	-40°C至+125°C	32引脚 LFCSP_WQ	CP-32-11	1,500
EVAL-ADuC7023QSPZ							采用32引脚 ADuC7023 QuickStart Plus 开发系统		
EVAL-ADuC7023QSPZ1							采用40引脚 ADuC7023 QuickStart Plus 开发系统		

<sup>1</sup> Z = 符合RoHS标准的器件。

**注释**

**注释**

**注释**

I<sup>2</sup>C指最初由Philips Semiconductors(现为NXP Semiconductors)开发的一种通信协议。