



ANALOG DEVICES

精密模拟微控制器，12 位模拟输入/输出，ARM7TDMI® MCU

ADuC7019/20/21/22/24/25/26/27/28/29

特性

模拟输入/输出

多通道、12位、1 MSPS模数转换器 (ADC)

最多16个ADC通道¹

全差分 and 单端模式

0V至VREF的模拟输入范围

12位电压输出数模转换器 (DAC)

最多4路DAC输出可用¹

片内基准电压

片内温度传感器 (±3°C典型值)

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

时钟选择

修正的片内振荡器 (±3%)

外部时钟晶体

可达44MHZ的外部时钟源

具有可编程分频器的41.78MHz锁相环

内存

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

片内外设

UART，2 个I²C®和SPI®串行I/O端口

最多40引脚GPIO端口¹

4个通用定时器

唤醒和看门狗定时器(WDT)

电源监控器

3相16位PWM发生器¹

可编程逻辑阵列(PLA)

可达512KB的外部存储器接口¹

电源

采用3V额定电源

主动模式：11 mA @ 5 MHz，40 mA @ 41.78 MHz

封装和温度范围

从40引脚6mmx6mm LFCSP封装到80引脚LQFP封装¹

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

工业控制和自动化系统

智能传感器，精密仪表

基站系统，光网络

功能框图

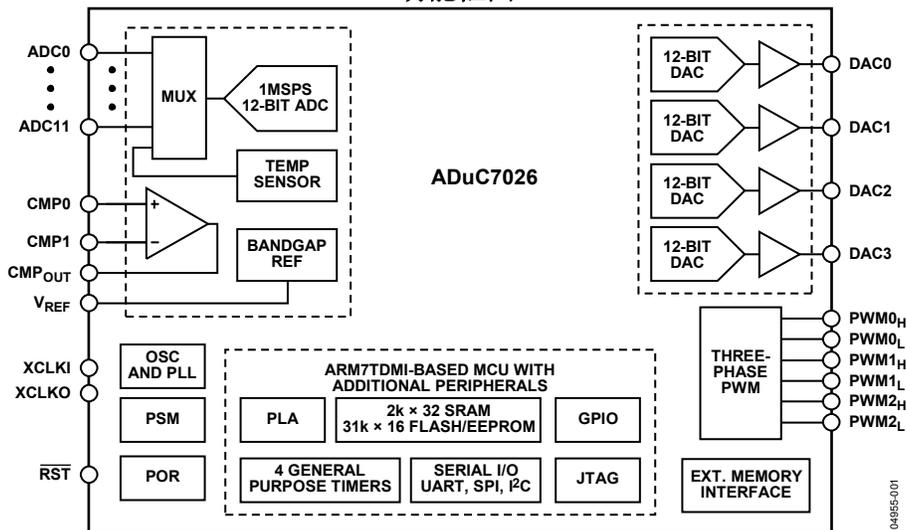


图1

¹ 取决于具体的器件型号。

更多信息请参阅订购指南。

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
Fax: 781.461.3113 ©2005–2009 Analog Devices, Inc. All rights reserved.

目录

特性	1	温度传感器.....	45
应用	1	带隙基准电压	45
功能框图	1	非易失Flash/EE存储器	46
修订历史	3	编程	46
概述	4	安全性	47
详细框图	5	Flash/EE控制接口	47
技术规格	6	SRAM和Flash/EE访问时间	49
时序规格	9	复位和重映射	49
绝对最大额定值	16	其他模拟外设	51
ESD警告	16	DAC	51
引脚配置和功能描述	17	电源监控器	52
ADuC7019/ADuC7020/ADuC7021/ADuC7022	17	比较器	52
ADuC7024/ADuC7025	20	振荡器和锁相环—电源控制	53
ADuC7026/ADuC7027	23	数字外设	56
ADuC7028	26	三相脉宽调制 (PWM)	56
典型工作特性	30	PWM模块说明	57
术语	33	通用输入/输出	62
ADC技术规格	33	串口多路复用器	64
DAC技术规格	33	UART串行接口	64
ARM7TDMI内核概览	34	串行外设接口	68
Thumb模式(T)	34	I ² C兼容接口	70
长乘 (M)	34	可编程逻辑阵列(PLA)	74
嵌入式ICE (I)	34	处理器基准外设	77
异常	34	中断系统	77
ARM寄存器	34	定时器	78
中断延迟	35	外部存储器接口	82
存储器结构	36	硬件设计考虑	86
存储器访问	36	电源	86
Flash/EE存储器	36	接地和电路板布局建议	87
SRAM	36	时钟振荡器	87
存储器映射寄存器	36	上电复位操作	88
ADC电路概览	40	典型系统配置	88
传递函数	40	开发工具	89
典型操作	41	基于PC工具	89
MMR接口	41	在线串行下载器	89
转换器操作	43	外形尺寸	90
驱动模拟输入	44	订购指南	92
校准	45		

修订历史

2009年2月—修订版B至修订版C

增加ADuC7029部分 通篇

2007年3月—修订版A至修订版B

增加ADuC7028部分 通篇

更新格式 通篇

图2更改 5

表1更改 6

ADuC7026/ADuC7027部分更改 23

图21更改 28

图32说明更改 30

表14更改 35

ADC电路概览部分更改 38

编程部分更改 44

Flash/EE控制接口部分更改 45

表24更改 47

RSTCLR寄存器部分更改 48

图52更改 49

图53更改 50

比较器部分更改 50

振荡器和锁相环-电源控制部分更改 51

数字外设部分更改 54

中断系统部分更改 75

定时器部分更改 76

外部存储器接口部分更改 80

增加IOV_{DD}电源灵敏度部分 84

订购指南更改 90

2006年1月—版本0到版本A

表1更改 6

增加Flash/EE存储器可靠性部分 43

表30更改 52

串行外设接口更改 66

订购指南更改 90

2005年10月—版本0: 初始版

概述

ADuC7019/20/21/22/24/25/26/27/28 在单芯片内集成 1 MSPS、12位、多通道高性能ADC的数据采集系统、16位/32位MCU和Flash/EE存储器。

ADC具有多达12个单端输入通道，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。4路DAC输出只是在特定型号上提供(ADuC7020和ADuC7026)。然而，在无DAC输出的情况下，这些引脚仍然可以用作ADC的输入引脚，这样ADC的输入最多可以达到16通道。ADC可以工作在单端模式或差分输入模式下。其输入电压为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外设设置。

根据不同型号，片内最多可内置4个缓冲电压输出DAC。可以通过编程将DAC输出范围设置为三种电压范围之一。

器件可以通过一个片内振荡器和一个锁相环产生一个41.78MHz的内部高频时钟信号。通过一个可编程时钟分频器进行中继，这个时钟可以用来产生MCU内核时钟工作频率。微控制器内核是ARM7TDMI，它是一个16位/32位RISC机器，其最高性能峰值高达41 MIPS。片内集成有8KB的SRAM和62KB非易失性Flash/EE存储器。ARM7TDMI内核将所有的存储器和寄存器看做是一个线性阵列。

片内出厂设置的固件支持通过UART接口或I²C接口进行在线串行下载，也可以通过JTAG接口进行非介入仿真。这些特性都集成在一个低成本的QuickStart™开发系统中支持这一MicroConverter®系列。

这些器件采用2.7 V至3.6 V电源供电，可以在-40°C至+125°C工业温度范围工作。当工作频率为41.78 MHz时，典型的功耗为120 mW。ADuC7019/20/21/22/24/25/26/27/28有多种内存类型和封装形式。

ADuC7019/20/21/22/24/25/26/27/28/29

详细框图

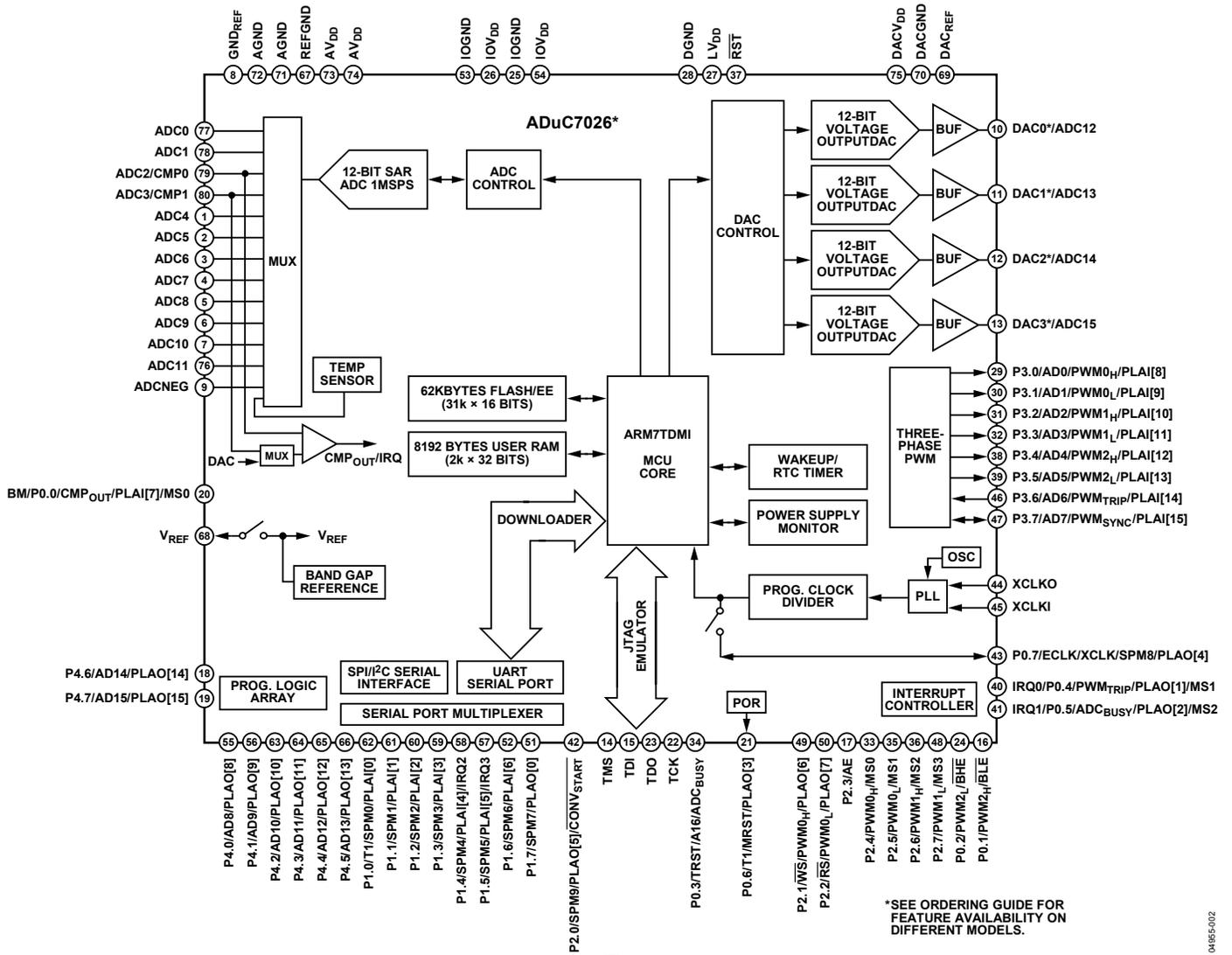


图2

*SEE ORDERING GUIDE FOR FEATURE AVAILABILITY ON DIFFERENT MODELS.

0485-002

ADuC7019/20/21/22/24/25/26/27/28/29

技术规格

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

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fADC/2
ADC上电时间		5		μs	
直流精度 ^{1,2}					
分辨率	12			位	
积分非线性		± 0.6	± 1.5	LSB	2.5 V内部基准电压
		± 1.0		LSB	1.0 V外部基准电压
微分非线性 ^{3,4}		± 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 ⁵					
偏移误差		± 1	± 2	LSB	
偏移误差匹配		± 1		LSB	
增益误差		± 2	± 5	LSB	
增益误差匹配		± 1		LSB	
动态性能					$f_{IN}=10\text{ kHz}$ 正弦波, $f_{SAMPLE}=1\text{ MSPS}$
信噪比(SNR)		69		dB	包括失真和噪声分量
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					
差分模式			$V_{CM}^6 \pm V_{REF}/2$	V	
单端模式			0 to VREF	V	
泄漏电流		± 1	± 6	μA	
输入电容		20		pF	在ADC采样期间
片内基准电压					在VREF和AGND之间接0.47 μF 电容
输出电压		2.5		V	
精度			± 5	mV	$T_A = 25^{\circ}C$
基准温度系数		± 40		ppm/ $^{\circ}C$	
电源电压抑制比		75		dB	
输出阻抗		70		Ω	$T_A = 25^{\circ}C$
内部VREF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					$R_L=5k\Omega$ $C_L=100\text{ pF}$
直流精度 ⁷					
分辨率		12		位	
相对精度		± 2		LSB	
微分非线性			± 1	LSB	保证单调性
偏移误差			± 15	mV	2.5 V内部基准电压
增益误差 ⁸			± 1	%	
增益误差失配		0.1		%	DAC0满刻度的百分比
模拟输出					
输出电压范围_0		0至DACREF		V	DACREF范围: DACGND至DACVDD
输出电压范围_1		0至2.5		V	
输出电压范围_2		0至DACVDD		V	
输出阻抗		2		Ω	

ADuC7019/20/21/22/24/25/26/27/28/29

参数	最小值	典型值	最大值	单位	测试条件/注释
DAC交流特性					
电压输出建立时间		10		μs	主进位1 LSB变化(DACxDAT寄存器中同时变化的最大位数)
数模转换毛刺能量		±20		nV-sec	
比较器					
输入偏置电压		±15		mV	迟滞可以通过CMPCON寄存器的CMPHYST位打开或关断 100 mV过驱、CMPRES = 11
输入偏置电流		1		μA	
输入电压范围	AGND		AV _{DD} -1.2	V	
输入电容		7		pF	
迟滞 ^{4,6}	2		15	mV	
响应时间		3		μs	
温度传感器					
25°C时电压输出		780		mV	
电压TC		-1.3		mV/°C	
精度		±3		°C	
电源监控器(PSM)					
IOV _{DD} 跳变点选择		2.79		V	两个可选择跳变点
		3.07		V	
电源跳变点精度		±2.5		%	已选跳变点标称电压
上电复位		2.36		V	
复位引脚干扰抑制 ³		50		μs	
看门狗定时器(WDT)					
超时周期	0		512	s	
FLASH/EE存储器					
持久性 ⁹	10,000			周期	T _J = 85°C
数据保持 ¹⁰	20			年	
数字输入					所有数字输入, 除了XCLKI和XCLKO V _{IH} =V _{DD} 或V _{IH} =5V V _{IL} =0V; ADuC7019/20/21/22/24/25的TDI除外 V _{IL} =0V; ADuC7019/20/21/22/24/25的TDI除外
逻辑1输入电流		±0.2	±1	μA	
逻辑0输入电流		-40	-60	μA	
		-80	-120	μA	
输入电容		10		pF	
逻辑输入 ³					所有数字输入, 除了XCLKI
V _{INL} , 输入低电压			0.8	V	
V _{INH} , 输入高电压	2.0			V	
逻辑输出					所有数字输出, 除了XCLKO I _{SOURCE} =1.6 mA I _{SINK} =1.6 mA
V _{OH} , 输出高电压	2.4			V	
V _{OL} , 输出低电压 ¹¹			0.4	V	
晶体输入XCLKI和XCLKO					
逻辑输入, 仅限XCLKI					
V _{INL} , 输入低电压		1.1		V	
V _{INH} , 输入高电压		1.7		V	
XCLKI输入电容		20		pF	
XCLKO输出电容		20		pF	
内部振荡器		32.768		kHz	
			±3	%	
MCU时钟速率					
采用32 kHz内部振荡器		326		kHz	CD = 7
采用32 kHz外部晶体		41.78		MHz	CD = 0
使用外部时钟	0.05		44	MHz	T _A = 85°C
	0.05		41.78	MHz	T _A = 125°C

ADuC7019/20/21/22/24/25/26/27/28/29

参数	最小值	典型值	最大值	单位	测试条件/注释
启动时间					内核时钟= 41.78 MHz
上电时		130		ms	
从暂停/休眠模式		24		ns	CD = 0
		3.06		µs	CD = 7
从睡眠模式		1.58		ms	
从停止模式		1.7		ms	
可编程逻辑阵列(PLA)					
引脚传播延迟		12		ns	从输入引脚到输出引脚
单元传播延迟		2.5		ns	
电源要求 ^{12, 13}					
电源电压范围					
AV _{DD} 至 AGND 和 IOV _{DD} 至 IOGND	2.7		3.6	V	
模拟电源电流					
AV _{DD} 电流		200		µA	ADC在空闲模式; 除ADuC7019外所有器件
		400		µA	ADC在空闲模式; 只有ADuC7019
DACV _{DD} 电流 ¹⁴		3	25	µA	
数字电源电流					
IOV _{DD} 正常模式下电流					从Flash/EE执行程序代码
		7	10	mA	CD = 7
		11	15	mA	CD = 3
		40	45	mA	CD = 0 (时钟频率41.78 MHz)
IOV _{DD} 暂停模式下电流		25	30	mA	CD = 0 (时钟频率41.78 MHz)
IOV _{DD} 睡眠模式下电流		250	400	µA	T _A =85°C
		600	1000	µA	T _A =125°C
模拟电源电流					
ADC		2		mA	@ 1 MSPS
		0.7		mA	@ 62.5 kSPS
DAC		700		µA	每DAC
ESD测试					2.5 V基准电压, T _A =25°C
最大HBM通过电压			4	kV	
最大FCIDM通过电压			0.5	kV	

¹ 在MicroConverter内核正常工作时, 保证所有ADC通道的技术规格。

² 应用于所有ADC输入通道。

³ 使用ADC偏移寄存器(ADCOF)和增益系数寄存器(ADCGN)中的出厂设定默认值进行测试。

⁴ 没有产品测试但是在产品发布时会提供一些设计和/或特性数据。

⁵ 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48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来满足规格要求(详见校准部分)。

⁶ 输入信号可以任何直流共模电压(V_{CM})为中心只要这个值位于ADC规定输入电压范围以内。

⁷ 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。

⁸ DAC增益误差是使用一个递减的数据范围100到内部2.5V V_{REF}基准电压计算出来的。

⁹ 耐用性是分别在-40°C、+25°C、+85°C和+125°C时依据JEDEC 22标准方法A117进行测试的。

¹⁰ 根据JEDEC 22标准方法A117使用寿命相当于(T_J)=85°C结温时的寿命。使用寿命会随着结温递减。

¹¹ 测试是在最多8个I/O端口输出低电平时进行的。

¹² 电源功耗分别在正常、暂停和睡眠模式下测试的, 这3种模式下的测试条件分别为: 正常模式供电电压为3.6V、暂停模式供电电压为3.6V、睡眠模式供电电压为3.6V。

¹³ 在一个Flash/EE擦写周期IOV_{DD}降低2 mA(典型值)。

¹⁴ 对于ADuC7019/20/21/22, 该电流必须加上AV_{DD}的电流。

时序规格

表2. 外部存储器写周期

参数	最小值	典型值	最大值	单位
CLK		UCLK		
$t_{MS_AFTER_CLKH}$	0		4	ns
$t_{ADDR_AFTER_CLKH}$	4		8	ns
$t_{AE_H_AFTER_MS}$		$\frac{1}{2}$ CLK		
t_{AE}		$(XMxPAR[14:12] + 1) \times CLK$		
$t_{HOLD_ADDR_AFTER_AE_L}$		$\frac{1}{2}$ CLK + $(!XMxPAR[10]) \times CLK$		
$t_{HOLD_ADDR_BEFORE_WR_L}$		$(!XMxPAR[8]) \times CLK$		
$t_{WR_L_AFTER_AE_L}$		$\frac{1}{2}$ CLK + $(!XMxPAR[10] + !XMxPAR[8]) \times CLK$		
$t_{DATA_AFTER_WR_L}$	8		12	ns
t_{WR}		$(XMxPAR[7:4] + 1) \times CLK$		
$t_{WR_H_AFTER_CLKH}$	0		4	ns
$t_{HOLD_DATA_AFTER_WR_H}$		$(!XMxPAR[8]) \times CLK$		
$t_{BEN_AFTER_AE_L}$		$\frac{1}{2}$ CLK		
$t_{RELEASE_MS_AFTER_WR_H}$		$(!XMxPAR[8]) \times CLK$		

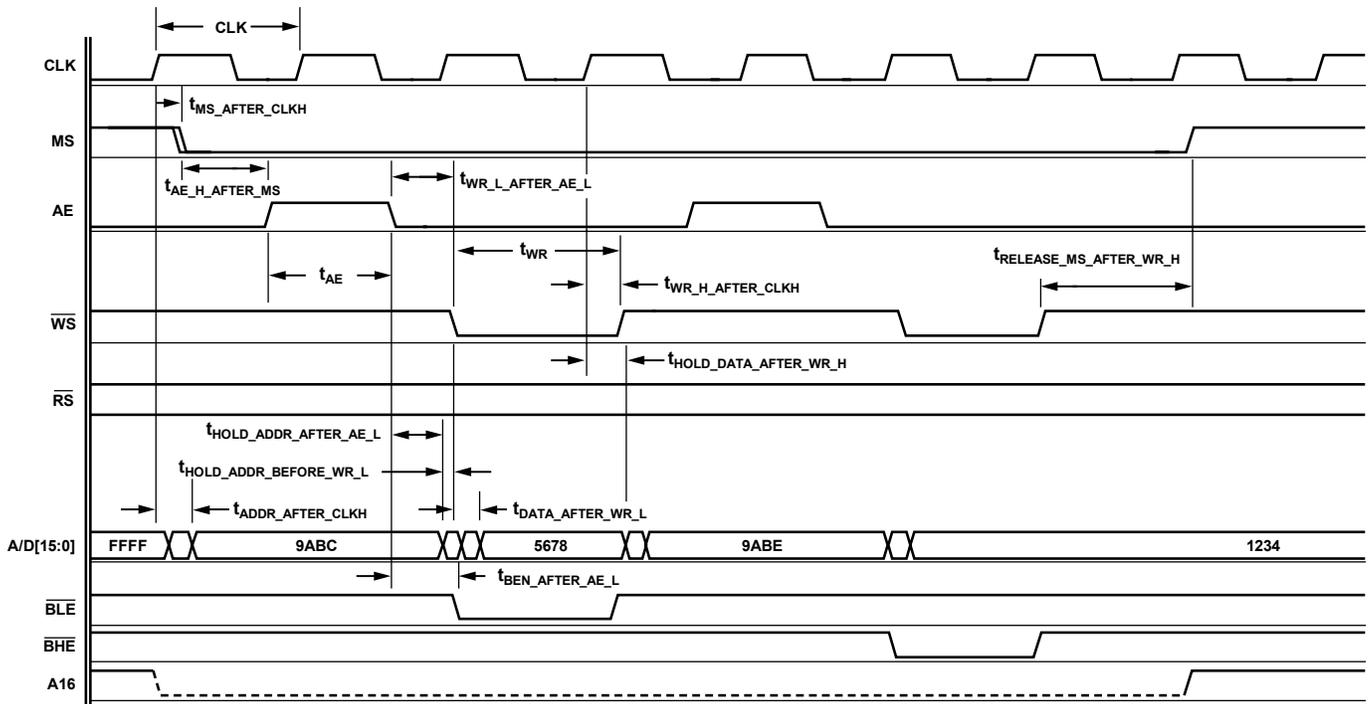


图3. 外部存储器写周期

04955-052

ADuC7019/20/21/22/24/25/26/27/28/29

表3. 外部存储器读周期

参数	最小值	典型值	最大值	单位
CLK	1/MD Clock	ns typ × (CDPOWCON[2:0] + 1)		
t _{MS_AFTER_CLKH}	4		8	ns
t _{ADDR_AFTER_CLKH}	4		16	ns
t _{AE_H_AFTER_MS}		½ CLK		
t _{AE}		(XMxPAR[14:12] + 1) × CLK		
t _{HOLD_ADDR_AFTER_AE_L}		½ CLK + (! XMxPAR[10]) × CLK		
t _{RD_L_AFTER_AE_L}		½ CLK + (! XMxPAR[10] + !XMxPAR[9]) × CLK		
t _{RD_H_AFTER_CLKH}	0		4	
t _{RD}		(XMxPAR[3:0] + 1) × CLK		
t _{DATA_BEFORE_RD_H}	16			ns
t _{DATA_AFTER_RD_H}	8	XMxPAR[9]) × CLK		
t _{RELEASE_MS_AFTER_RD_H}		1 × CLK		

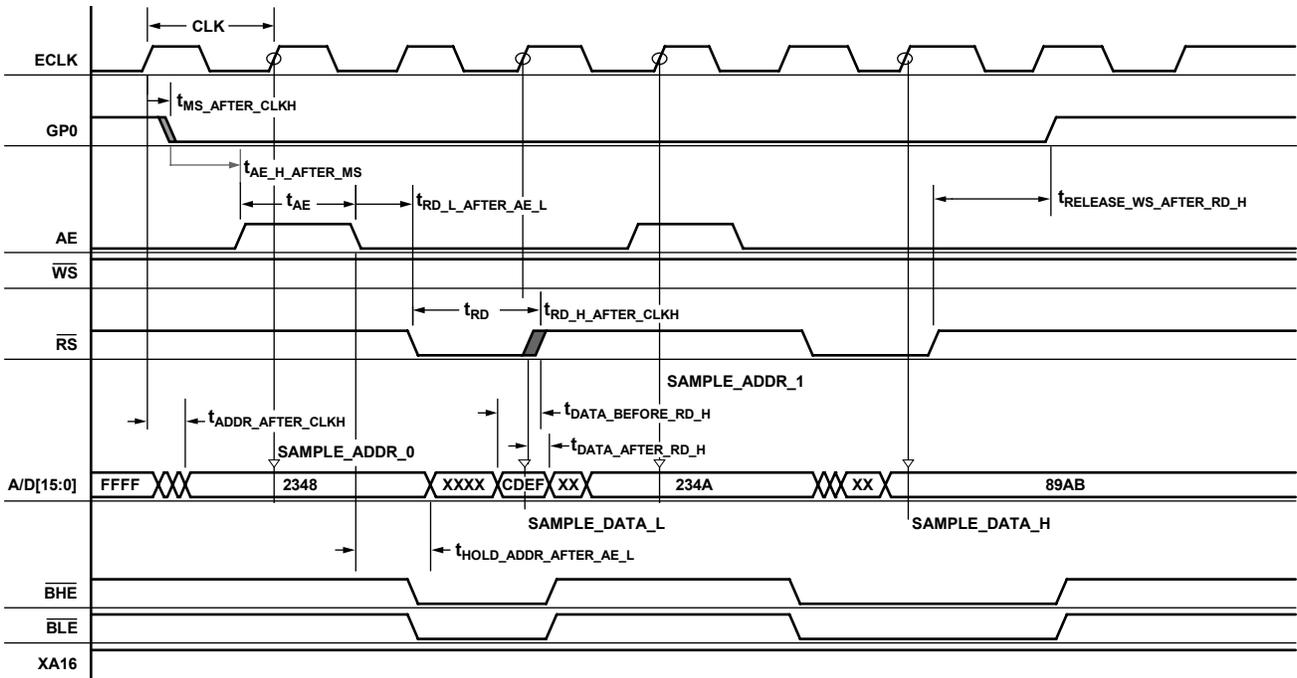


图 4. 外部存储器读周期

04955-053

ADuC7019/20/21/22/24/25/26/27/28/29

表6. SPI 主机模式时序(相位模式为1)

参数	描述	最小值	典型值	最大值	单位
t_{SL}	SCLOCK低脉冲宽度 ¹		$(SPIDIV+1) \times t_{HCLK}$		ns
t_{SH}	SCLOCK 高脉冲宽度 ¹		$(SPIDIV+1) \times t_{HCLK}$		ns
t_{DAV}	SCLOCK 边沿之后数据输出有效			25	ns
t_{DSU}	SCLOCK沿之前数据输入建立时间 ²	$1 \times t_{UCLK}$			ns
t_{DHD}	SCLOCK沿之后数据输入保持时间 ²	$2 \times t_{UCLK}$			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}	SCLOCK上升时间		5	12.5	ns
t_{SF}	SCLOCK 下降时间		5	12.5	ns

¹ t_{HCLK} 取决于时钟分频器或PLLCON寄存器的CD位。 $t_{HCLK} = t_{UCLK}/2^{CD}$ 。

² $t_{UCLK} = 23.9 \text{ ns}$.与时钟分频器之前从锁相环输出的41.78 MHz内部时钟相对应。

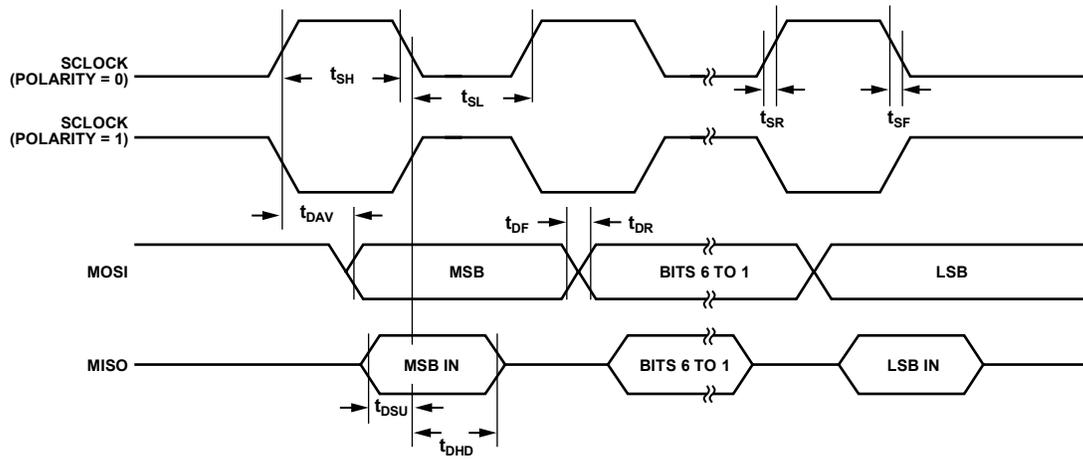


图 6. SPI 主机模式时序(相位模式为1)

04955-055

表 6. SPI 主机模式时序(相位模式为0)

参数	描述	最小值	典型值	最大值	单位
t_{SL}	SCLOCK低脉冲宽度 ¹		$(SPIDIV+1) \times t_{HCLK}$		ns
t_{SH}	SCLOCK 高脉冲宽度 ¹		$(SPIDIV+1) \times t_{HCLK}$		ns
t_{DAV}	SCLOCK 边沿之后数据输出有效			25	ns
t_{DOSU}	SCLOCK沿之前数据输出建立			75	ns
t_{DSU}	SCLOCK沿之前数据输入建立时间 ²	$1 \times t_{UCLK}$			ns
t_{DHD}	SCLOCK沿之后数据输入保持时间 ²	$2 \times t_{UCLK}$			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}	SCLOCK上升时间		5	12.5	ns
t_{SF}	SCLOCK 下降时间		5	12.5	ns

¹ t_{HCLK} 取决于时钟分频器或PLLCON寄存器的CD位。 $t_{HCLK} = t_{UCLK}/2^{CD}$ 。

² $t_{UCLK} = 23.9$ ns.与时钟分频器之前从锁相环输出的41.78 MHz 内部时钟相对应。

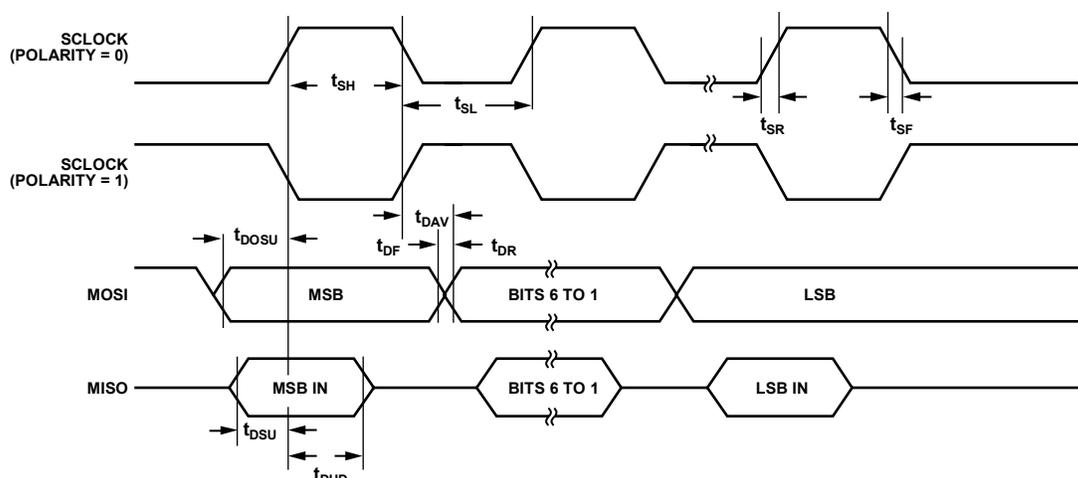


图 7. SPI 主机模式时序(相位模式为0)

0495E-056

ADuC7019/20/21/22/24/25/26/27/28/29

表7. SPI 从机模式时序(相位模式为1)

参数	描述	最小值	典型值	最大值	单位
$t_{\overline{CS}}$	\overline{CS} 到 SCLOCK 边沿 ¹	$(2 \times t_{HCLK}) + (2 \times t_{UCLK})$			ns
t_{SL}	SCLOCK 低脉冲宽度 ²		$(SPIDIV+1) \times t_{HCLK}$		ns
t_{SH}	SCLOCK 高脉冲宽度 ²		$(SPIDIV+1) \times t_{HCLK}$		ns
t_{DAV}	SCLOCK 边沿之后数据输出有效			25	ns
t_{DSU}	SCLOCK 边沿之前数据输入建立时间 ¹	$1 \times t_{UCLK}$			ns
t_{DHD}	SCLOCK 边沿之后数据输入保持时间 ¹	$2 \times t_{UCLK}$			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}	SCLOCK 上升时间		5	12.5	ns
t_{SF}	SCLOCK 下降时间		5	12.5	ns
t_{SFS}	SCLOCK 边沿后 \overline{CS} 变高	0			ns

¹ $t_{UCLK} = 23.9 \text{ ns}$, 与时钟分频器之前从锁相环输出的 41.78 MHz 内部时钟相对应。

² t_{HCLK} 取决于时钟分频器或 PLLCON 寄存器的 CD 位。 $t_{HCLK} = t_{UCLK} / 2^{CD}$ 。

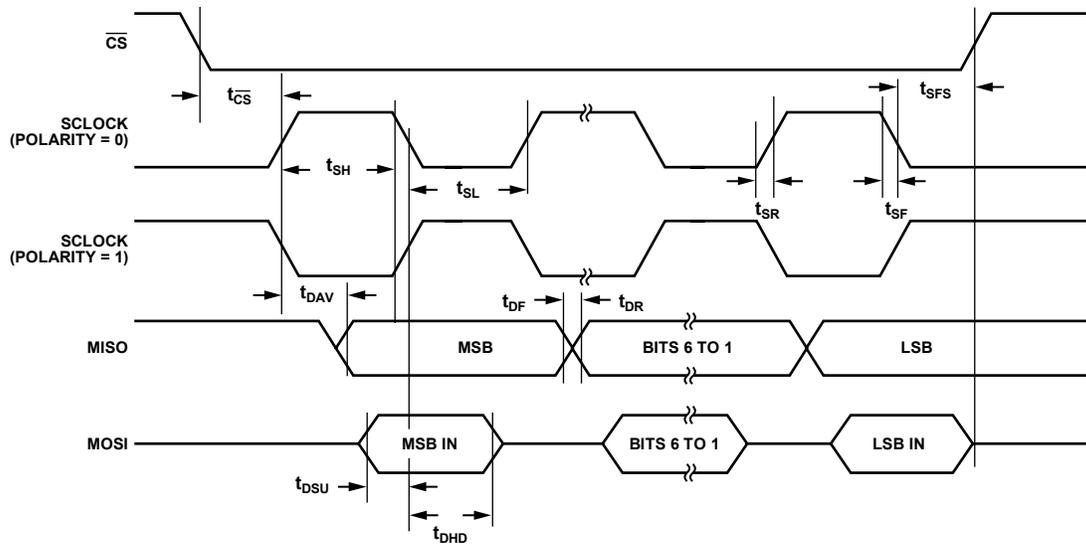


图 8. SPI 从机模式时序(相位模式为1)

表8. SPI 从机模式时序(相位模式为0)

参数	描述	最小值	典型值	最大值	单位
$t_{\overline{CS}}$	\overline{CS} 到 SCLOCK 边沿 ¹	$(2 \times t_{HCLK}) + (2 \times t_{UCLK})$			ns
t_{SL}	SCLOCK 低脉冲宽度 ²		$(SPIDIV+1) \times t_{HCLK}$		ns
t_{SH}	SCLOCK 高脉冲宽度 ²		$(SPIDIV+1) \times t_{HCLK}$		ns
t_{DAV}	SCLOCK 边沿之后数据输出有效			25	ns
t_{DSU}	SCLOCK 边沿之前数据输入建立时间 ¹	$1 \times t_{UCLK}$			ns
t_{DHD}	SCLOCK 边沿之后数据输入保持时间 ¹	$2 \times t_{UCLK}$			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}	SCLOCK 上升时间		5	12.5	ns
t_{SF}	SCLOCK 下降时间		5	12.5	ns
t_{DOCS}	\overline{CS} 边沿之后数据输出有效			25	ns
t_{SFS}	SCLOCK 边沿后 \overline{CS} 变高	0			ns

¹ $t_{UCLK} = 23.9 \text{ ns}$. 与时钟分频器之前从锁相环输出的 41.78 MHz 内部时钟相对应。

² t_{HCLK} 取决于时钟分频器或 PLLCON 寄存器的 CD 位。 $t_{HCLK} = t_{UCLK} / 2^{CD}$ 。

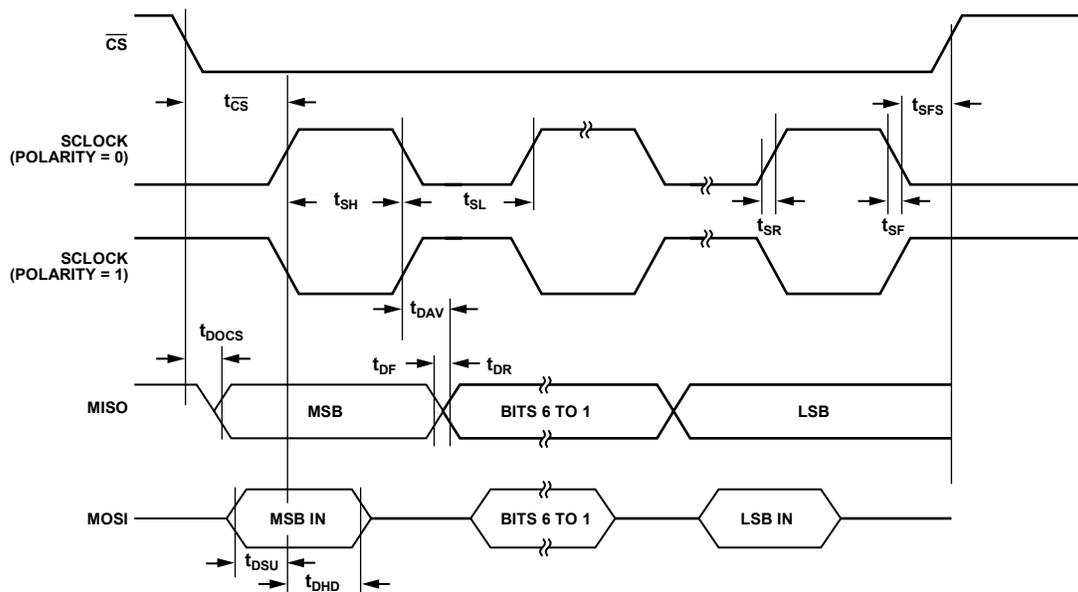


图 9. SPI 从机模式时序 (相位模式为0)

04955-058

绝对最大额定值

AGND = REFGND = DACGND = GND_{REF},

T_A = 25°C, 除非另有说明。

表9

参数	额定值
AV _{DD} 至IOV _{DD}	-0.3V至+0.3V
AGND至DGND	-0.3V至+0.3V
IOV _{DD} 至IOGND, AV _{DD} 至AGND	-0.3V至+6V
数字输入电压至IOGND	-0.3V至+5.3V
数字输出电压至IOGND	-0.3V至IOV _{DD} +0.3V
V _{REF} 至AGND	-0.3V至AV _{DD} +0.3V
模拟输入至AGND	-0.3V至AV _{DD} +0.3V
模拟输出至AGND	-0.3V至AV _{DD} +0.3V
工作温度范围	-40°C至+125°C
储存温度范围	-65°C至+150°C
结温	150°C
θ _{JA} 热阻抗	
40引脚LFCSP封装	26°C/W
64引脚LFCSP封装	24°C/W
64引脚CSP_BGA封装	75°C/W
64引脚LQFP封装	47°C/W
80引脚LQFP封装	38°C/W
回流焊接峰值温度	
锡铅体系(10秒~ 30秒)	240°C
RoHS体系(20秒~ 40秒)	260°C

注意, 超出以上所列绝对最大额定值可能导致器件永久性损坏。这只是强调的额定值, 不涉及器件在这些或任何其它条件下超出本技术规格指标的功能性操作。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用保护电路, 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

ADuC7019/ADuC7020/ADuC7021/ADuC7022

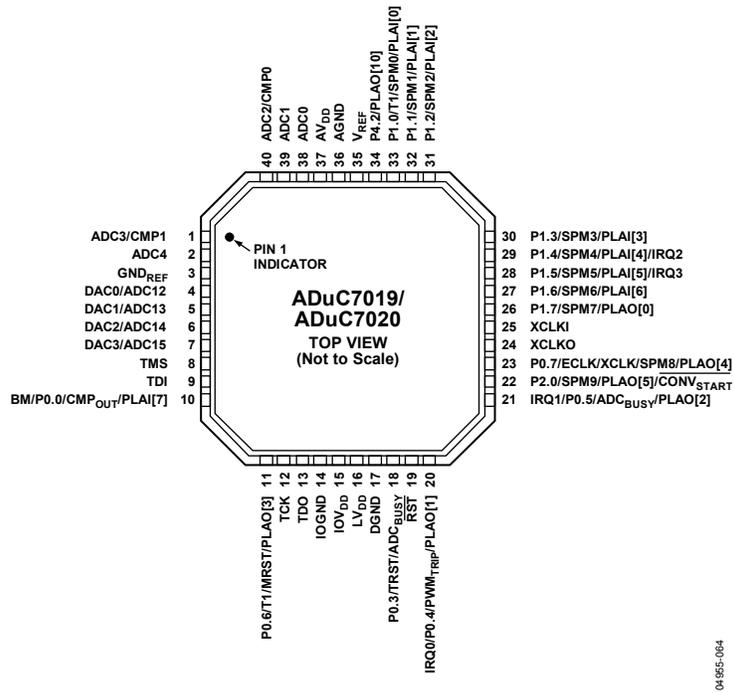


图10. 40 ADuC7019/ADuC7020 引脚FCSP_VQ封装引脚配置

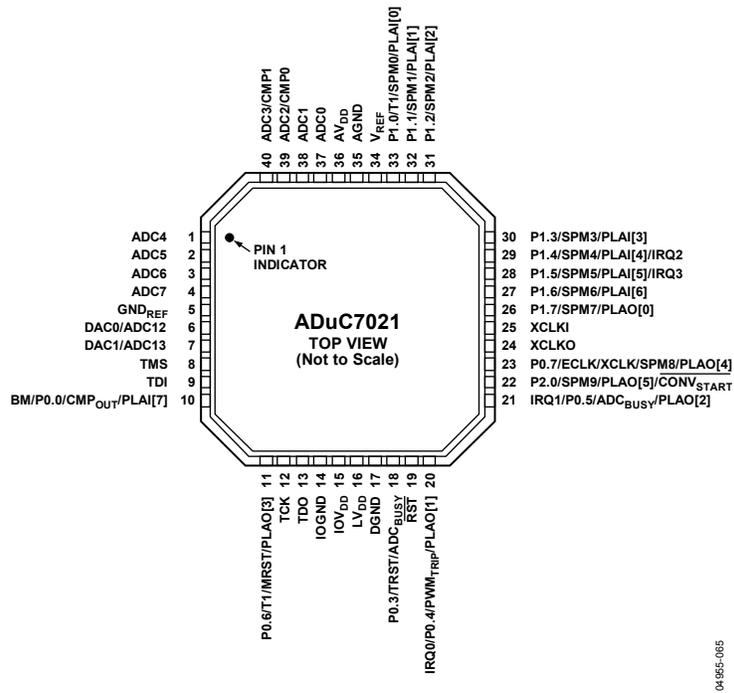


图 11. ADuC7021 40 引脚FCSP_VQ封装引脚配置

ADuC7019/20/21/22/24/25/26/27/28/29

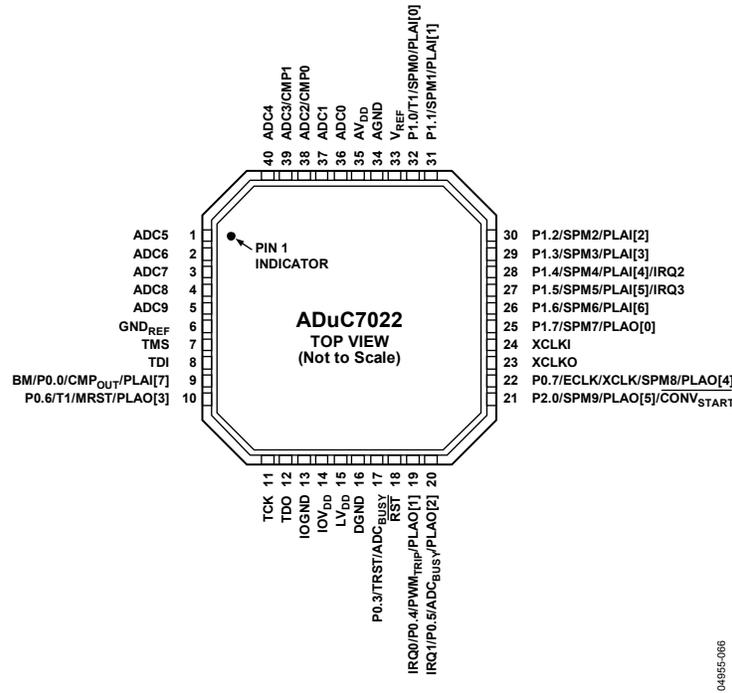


图 12. ADuC7022 40引脚FCSP_VQ封装引脚配置

表10. ADuC7019/ADuC7020/ADuC7021/ADuC7022引脚配置描述

引脚编号			引脚名称	描述
7019/7020	7021	7022		
38	37	36	ADC0	单端或差分模拟输入0。
39	38	37	ADC1	单端或差分模拟输入1。
40	39	38	ADC2/CMP0	单端或差分模拟输入2/比较器正相输入。
1	40	39	ADC3/CMP1	单端或差分模拟输入3(ADuC7019缓冲输入)/比较器反相输入端。
2	1	40	ADC4	单端或差分模拟输入4。
-	2	1	ADC5	单端或差分模拟输入5。
-	3	2	ADC6	单端或差分模拟输入6。
-	4	3	ADC7	单端或差分模拟输入7。
-	-	4	ADC8	单端或差分模拟输入8。
-	-	5	ADC9	单端或差分模拟输入9。
3	5	6	GND _{REF}	ADC地基准电压。为了优化性能，模拟电源应同IOGND和DGND分离。
4	6	-	DAC0/ADC12	DAC0电压输出/单端或差分模拟输入12。
5	7	-	DAC0/ADC13	DAC0电压输出/单端或差分模拟输入13。
6	-	-	DAC0/ADC14	DAC0电压输出/单端或差分模拟输入14。
7	-	-	DAC0/ADC15	ADuC7020。DAC3电压输出在ADuC7019芯片内，需要在该引脚和AGND/单端或差分模拟输入15之间连接一个10 nF电容。
8	8	7	TMS	测试模式选择，JTAG测试端口输入。调试和下载访问。该引脚具有一个连接至DVDD的内部上拉电阻。在有些情况下，也需要一个外部上拉电阻(~100K)，以确保器件不会进入错误状态。
9	9	8	TDI	测试数据输入，JTAG测试端口输入。调试和下载访问。
10	10	9	BM/P0.0/CMP _{OUT} /PLAI[7]	多功能输入输出引脚。引导模式(BM)。复位时BM为低电平则芯片进入串行下载模式；BM由1kΩ电阻上拉至高电平则运行程序代码/通用I/O端口P0.0/电压比较器输出/PLA输入单元7。

ADuC7019/20/21/22/24/25/26/27/28/29

引脚编号			引脚名称	描述
7019/7020	7021	7022		
11	11	10	P0.6/T1/MRST/PLAO[3]	多功能引脚。复位后拉低。通用输入输出端口P 0.6/定时器1输入/上电复位输出/可编程逻辑阵列输出单元3。
12	12	11	TCK	测试时钟, JTAG测试端口输入。调试和下载访问。
13	13	12	TDO	测试数据输出, JTAG测试端口输出。调试和下载访问。
14	14	13	IOGND	GPIO地。通常连接到DGND。
15	15	14	IOV _{DD}	3.3 V电源。用于GPIO和片内稳压器输入。
16	16	15	LV _{DD}	片内稳压器2.6V输出。该输出只能通过一个0.47 μF电容连接到DGND。
17	17	16	DGND	内核逻辑地。
18	18	17	P0.3/TRST/ADC _{BUSY}	通用输入输出端口P0.3/测试复位, JTAG测试端口输入/ADC _{BUSY} 信号输出。
19	19	18	$\overline{\text{RST}}$	复位输入, 低电平有效。
20	20	19	IRQ0/P0.4/PWM _{TRIP} /PLAO[1]	多功能输入输出引脚。外部中断请求0, 高电平有效/通用输入输出端口P 0.4/PWM触发外部输入/可编程逻辑阵列输出单元1。
21	21	20	IRQ1/P0.5/ADC _{BUSY} /PLAO[2]	多功能输入输出引脚。外部中断请求1, 高电平有效/通用输入输出端口P 0.5/ADC _{BUSY} 信号输出/可编程逻辑阵列输出单元2。
22	22	21	P2.0/SPM9/PLAO[5]/ $\overline{\text{CONV}}_{\text{START}}$	串行复用端口。通用输入输出端口P 2.0/UART/可编程逻辑阵列输出单元5/ADC起始转换输入信号。
23	23	22	P0.7/ECLK/XCLK/SPM8/PLAO[4]	串行复用端口。通用输入输出端口P0.7/外部时钟信号输出/内部时钟发生器电路输入/UART/可编程逻辑阵列输出单元4。
24	24	23	XCLKO	晶体振荡反相器输出。
25	25	24	XCLKI	晶体振荡反相器输入、内部时钟发生器电路输入。
26	26	25	P1.7/SPM7/PLAO[0]	串行复用端口。通用输入输出端口P 1.7/UART, SPI/可编程逻辑阵列输出单元0。
27	27	26	P1.6/SPM6/PLAI[6]	串行复用端口。通用输入输出端口P 1.6/UART, SPI/可编程逻辑阵列输入单元6。
28	28	27	P1.5/SPM5/PLAI[5]/IRQ3	串行复用端口。通用输入输出端口P 1.5/UART, SPI/可编程逻辑阵列输入单元5/外部中断请求3, 高电平有效。
29	29	28	P1.4/SPM4/PLAI[4]/IRQ2	串行复用端口。通用输入输出端口P 1.4/UART, SPI/可编程逻辑阵列输入单元4/外部中断请求2, 高电平有效。
30	30	29	P1.3/SPM3/PLAI[3]	串行复用端口。通用输入输出端口P 1.3/UART, I2C1/可编程逻辑阵列输入单元3。
31	31	30	P1.2/SPM2/PLAI[2]	串行复用端口。通用输入输出端口P 1.2/UART, I2C1/可编程逻辑阵列输入单元2。
32	32	31	P1.1/SPM1/PLAI[1]	串行复用端口。通用输入输出端口P 1.1/UART, I2C0/可编程逻辑阵列输入单元1。
33	33	32	P1.0/T1/SPM0/PLAI[0]	串行复用端口。通用输入输出端口P 1.0/定时器1输入/UART, I2C0/可编程逻辑阵列输入单元0。
34	-	-	P4.2/PLAO[10]	通用输入输出端口P 4.2/可编程逻辑阵列输出单元10。
35	34	33	V _{REF}	2.5 V内部基准电压。在使用内部基准时必须外部对AGND连接一个0.47 μF电容。
36	35	34	AGND	模拟地。模拟电路的地基准点。
37	36	35	AV _{DD}	3.3 V模拟电源。

ADuC7019/20/21/22/24/25/26/27/28/29

ADuC7024/ADuC7025

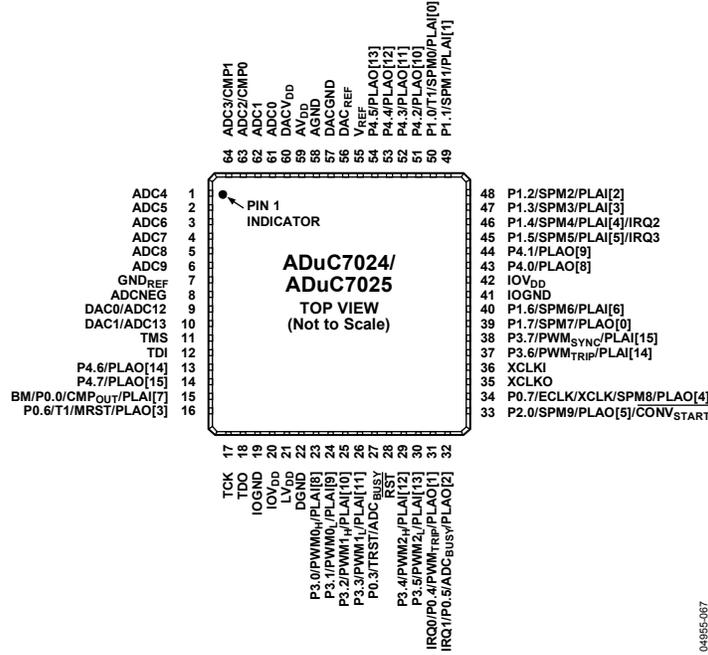


图13. ADuC7024/ADuC7025 64引脚LCSP_VQ封装引脚配置

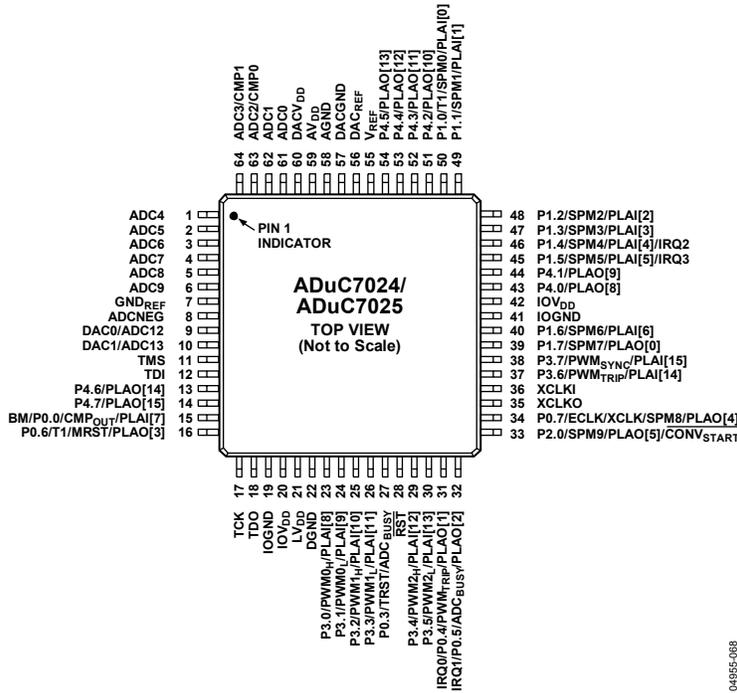


图14. ADuC7024/ADuC7025 64引脚LQFP封装引脚配置

ADuC7019/20/21/22/24/25/26/27/28/29

表11. 引脚功能描述 (ADuC7024/ADuC7025 64引脚LFCSP_VQ封装和64引脚LQFP封装)

引脚编号	引脚名称	描述
1	ADC4	单端或差分模拟输入4。
2	ADC5	单端或差分模拟输入5。
3	ADC6	单端或差分模拟输入6。
4	ADC7	单端或差分模拟输入7。
5	ADC8	单端或差分模拟输入8。
6	ADC9	单端或差分模拟输入9。
7	GND _{REF}	ADC地基准电压。为了优化性能，模拟电源应同IOGND和DGND分离。
8	ADCNEG	伪差分模式下ADC偏置点或反相模拟输入。必须连接到要转换信号的地。该偏置点必须在0 V ~ 1 V之间。
9	DAC0/ADC12	DAC0电压输出/单端或差分模拟输入12。在ADuC7025上没有DAC输出。
10	DAC1/ADC13	DAC1模拟电压输出/单端或差分输入13。在ADuC7027上没有DAC输出。
11	TMS	JTAG测试端口输入，测试模式选择。调试和下载访问。
12	TDI	JTAG测试端口输入，测试数据输入。调试和下载访问。
13	P4.6/PLAO[14]	通用输入输出端口P 4.6/可编程逻辑阵列输出单元14。
14	P4.7/PLAO[15]	通用输入输出端口P 4.7/可编程逻辑阵列输出单元15。
15	BM/P0.0/CMP _{OUT} /PLAI[7]	多功能输入输出引脚。引导模式。复位时如果BM为低电平，则ADuC7024/ADuC7025进入下载模式，如果BM通过1 kΩ电阻上拉至高电平则执行程序代码/通用输入输出端口P 0.0/电压比较器输出/可编程逻辑阵列输入单元7。
16	P0.6/T1/MRST/PLAO[3]	多功能引脚，复位后输出低电平。通用输入输出端口P 0.6/定时器1输入/上电复位输出/可编程逻辑阵列输出单元3。
17	TCK	JTAG测试端口输入，测试时钟。调试和下载访问。
18	TDO	JTAG测试端口输出，测试数据输出。调试和下载访问。
19	IOGND	GPIO地。通常连接到DGND。
20	IOV _{DD}	3.3 V电源用于GPIO和片内稳压器输入。
21	LV _{DD}	片内稳压器2.6 V输出。该输出只能通过一个0.47 μF电容连接至DGND。
22	DGND	内核逻辑地。
23	P3.0/PWM0 _H /PLAI[8]	通用输入输出端口P 3.0/PWM相位0高边输出/可编程逻辑阵列输入单元8。
24	P3.1/PWM0 _L /PLAI[9]	通用输入输出端口P 3.1/PWM相位0低边输出/可编程逻辑阵列输入单元9。
25	P3.2/PWM1 _H /PLAI[10]	通用输入输出端口P 3.2/PWM相位1高边输出/可编程逻辑阵列输入单元10。
26	P3.3/PWM1 _L /PLAI[11]	通用输入输出端口P 3.3/PWM相位1低边输出/可编程逻辑阵列输入单元11。
27	P0.3/TRST/ADC _{BUSY}	通用输入输出端口P 0.3/JTAG测试端口输入，测试复位/ADC _{BUSY} 信号输出。
28	RST	复位输入，低电平有效。
29	P3.4/PWM2 _H /PLAI[12]	通用输入输出端口P 3.4/PWM相位2高边输出/可编程逻辑阵列输入单元12。
30	P3.5/PWM2 _L /PLAI[13]	通用输入输出端口P 3.5/PWM相位2低边输出/可编程逻辑阵列输入单元13。
31	IRQ0/P0.4/PWM _{TRIP} /PLAO[1]	多功能输入输出引脚。外部中断请求0，高电平有效/通用输入输出端口P 0.4/PWM触发外部输入/可编程逻辑阵列输出单元1。
32	IRQ1/P0.5/ADC _{BUSY} /PLAO[2]	多功能输入输出引脚。外部中断请求1，高电平有效/通用输入输出端口P 0.5/ADC _{BUSY} 信号输出/可编程逻辑阵列输出单元2。
33	P2.0/SPM9/PLAO[5]/ $\overline{\text{CONV}}_{\text{START}}$	串行复用端口。通用输入输出端口P 2.0/UART/可编程逻辑阵列输出单元5/ADC起始转换输入信号。
34	P0.7/ECLK/XCLK/SPM8/PLAO[4]	串行复用端口。通用输入输出端口P0.7/外部时钟信号输出/内部时钟发生器电路输入/UART/可编程逻辑阵列输出单元4。
35	XCLKO	晶体振荡反相器输出。
36	XCLKI	晶体振荡反相器输入、内部时钟发生器电路输入。

ADuC7019/20/21/22/24/25/26/27/28/29

引脚编号	引脚名称	描述
37	P3.6/PWM _{TRIP} /PLAI[14]	通用输入输出端口P 3.6/PWM安全关断/可编程逻辑阵列输入单元14。
38	P3.7/PWM _{SYNC} /PLAI[15]	通用输入输出端口P 3.7/PWM同步输入输出/可编程逻辑阵列输入单元15。
39	P1.7/SPM7/PLAO[0]	串行复用端口。通用输入输出端口P 1.7/UART, SPI/可编程逻辑阵列输出单元0。
40	P1.6/SPM6/PLAI[6]	串行复用端口。通用输入输出端口P 1.6/UART, SPI/可编程逻辑阵列输入单元6。
41	IOGND	GPIO地。通常连接到DGND。
42	IOV _{DD}	3.3 V电源。用于GPIO和片内稳压器输入。
43	P4.0/PLAO[8]	通用输入输出端口P 4.0/可编程逻辑阵列输出单元8。
44	P4.1/PLAO[9]	通用输入输出端口P 4.1/可编程逻辑阵列输出单元9。
45	P1.5/SPM5/PLAI[5]/IRQ3	串行复用端口。通用输入输出端口P 1.5/UART, SPI/可编程逻辑阵列输入单元5/外部中断请求3, 高电平有效。
46	P1.4/SPM4/PLAI[4]/IRQ2	串行复用端口。通用输入输出端口P 1.4/UART, SPI/可编程逻辑阵列输入单元4/外部中断请求2, 高电平有效。
47	P1.3/SPM3/PLAI[3]	串行复用端口。通用输入输出端口P 1.3/UART, I2C1/可编程逻辑阵列输入单元3。
48	P1.2/SPM2/PLAI[2]	串行复用端口。通用输入输出端口P 1.2/UART, I2C1/可编程逻辑阵列输入单元2。
49	P1.1/SPM1/PLAI[1]	串行复用端口。通用输入输出端口P 1.1/UART, I2C0/可编程逻辑阵列输入单元1。
50	P1.0/T1/SPM0/PLAI[0]	串行复用端口。通用输入输出端口P 1.0/定时器1输入/UART, I2C0/可编程逻辑阵列输入单元0。
51	P4.2/PLAO[10]	通用输入输出端口P 4.2/可编程逻辑阵列输出单元10。
52	P4.3/PLAO[11]	通用输入输出端口P 4.3/可编程逻辑阵列输出单元11。
53	P4.4/PLAO[12]	通用输入输出端口P 4.4/可编程逻辑阵列输出单元12。
54	P4.5/PLAO[13]	通用输入输出端口P 4.5/可编程逻辑阵列输出单元13。
55	V _{REF}	2.5 V内部基准电压。在使用内部基准时必须外部对AGND连接一个0.47 μ F电容。
56	DAC _{REF}	DAC外部基准电压。范围: DACGND ~ DACV _{DD} 。
57	DACGND	DAC地。通常连接到AGND。
58	AGND	模拟地。模拟电路的地基准点。
59	AV _{DD}	3.3 V模拟电源。
60	DACV _{DD}	DAC3.3 V电压源。必须连接到AV _{DD} 。
61	ADC0	单端或差分模拟输入0。
62	ADC1	单端或差分模拟输入1。
63	ADC2/CMP0	单端或差分模拟输入2/比较器正相输入。
64	ADC3/CMP1	单端或差分模拟输入3/比较器反相输入端。

ADuC7019/20/21/22/24/25/26/27/28/29

ADuC7026/ADuC7027

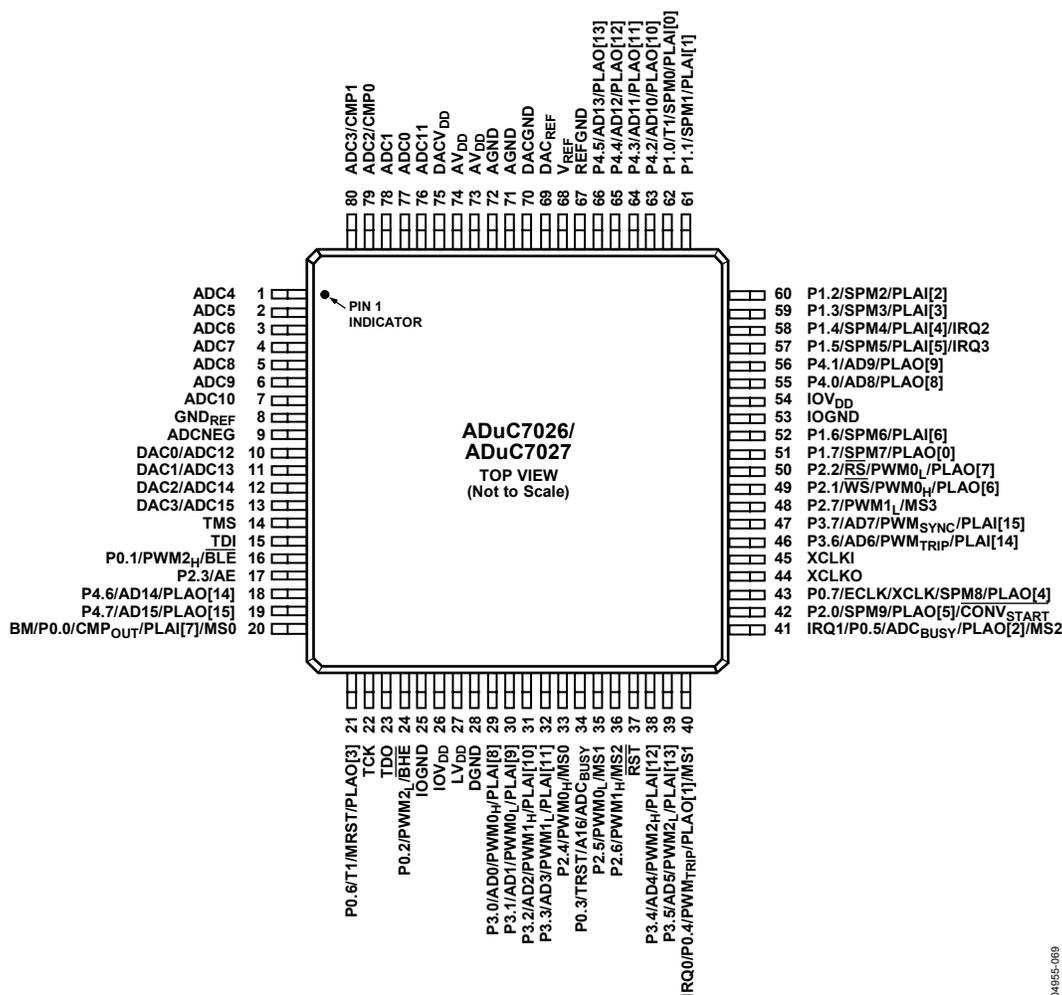


图15. 80引脚LQFP封装引脚配置(ADuC7026/ADuC7027)

表12 引脚功能描述(ADuC7026/ADuC7027)

引脚编号	引脚名称	描述
1	ADC4	单端或差分模拟输入4。
2	ADC5	单端或差分模拟输入5。
3	ADC6	单端或差分模拟输入6。
4	ADC7	单端或差分模拟输入7。
5	ADC8	单端或差分模拟输入8。
6	ADC9	单端或差分模拟输入9。
7	ADC10	单端或差分模拟输入10。
8	GND _{REF}	ADC地基准电压。为了优化性能，模拟电源应同IOGND和DGND分离。
9	ADCNEG	伪差分模式下ADC偏置点或反相模拟输入。必须连接到要转换信号的地。该偏置点必须在0 V ~ 1 V之间。
10	DAC0/ADC12	DAC0模拟电压输出/单端或差分输入12。在ADuC7027上没有DAC输出。
11	DAC1/ADC13	DAC1模拟电压输出/单端或差分输入13。在ADuC7027上没有DAC输出。
12	DAC2/ADC14	DAC2模拟电压输出/单端或差分输入14。在ADuC7027上没有DAC输出。
13	DAC3/ADC15	DAC3模拟电压输出/单端或差分输入15。在ADuC7027上没有DAC输出。
14	TMS	JTAG测试端口输入，测试模式选择。调试和下载访问。

ADuC7019/20/21/22/24/25/26/27/28/29

引脚编号	引脚名称	描述
15	TDI	JTAG测试端口输入，测试数据输入。调试和下载访问。
16	P0.1/PWM2 _H /BLE	通用输入输出端口P0.1/PWM相位2高边输出/外部存储器低字节使能。
17	P2.3/AE	通用输入输出端口P 2.3/外部存储器访问使能。
18	P4.6/AD14/PLAO[14]	通用输入输出端口P 4.6/外部存储器接口/可编程逻辑阵列输出单元14。
19	P4.7/AD15/PLAO[15]	通用输入输出端口P 4.7/外部存储器接口/可编程逻辑阵列输出单元15。
20	BM/P0.0/CMP _{OUT} /PLAI[7]/MS0	多功能输入输出引脚。引导模式。如果在复位时BM为低，ADuC7026/DuC7027进入UART下载模式，如果BM使用1 kΩ电阻上拉至高电平则执行程序代码/通用输入输出端口P 0.0/电压比较器输出/可编程逻辑阵列输入单元7/外部存储器选择0。
21	P0.6/T1/MRST/PLAO[3]	多功能引脚，复位后输出低电平。通用输入输出端口P 0.6/定时器1输入/上电复位输出/可编程逻辑阵列输出单元3。
22	TCK	JTAG测试端口输入，测试时钟。调试和下载访问。
23	TDO	JTAG测试端口输出，测试数据输出。调试和下载访问。
24	P0.2/PWM2 _L /BLE	通用输入输出端口P 0.2/PWM相位2低边输出/外部存储器高字节使能。
25	IOGND	GPIO地。通常连接到DGND。
26	IOV _{DD}	3.3 V电源。用于GPIO和片内稳压器输入。
27	LV _{DD}	片内稳压器2.6 V输出。该输出只能通过一个0.47 μF电容连接至DGND。
28	DGND	内核逻辑地。
29	P3.0/AD0/PWM0 _H /PLAI[8]	通用输入输出端口P 3.0/外部存储器接口/PWM相位0高边输出/可编程逻辑阵列输入单元8。
30	P3.1/AD1/PWM0 _L /PLAI[9]	通用输入输出端口P 3.1/外部存储器接口/PWM相位0低边输出/可编程逻辑阵列输入单元9。
31	P3.2/AD2/PWM1 _H /PLAI[10]	通用输入输出端口P 3.2/外部存储器接口/PWM相位1高边输出/可编程逻辑阵列输入单元10。
32	P3.3/AD3/PWM1 _L /PLAI[11]	通用输入输出端口P 3.3/外部存储器接口/PWM相位1低边输出/可编程逻辑阵列输入单元11。
33	P2.4/PWM0 _H /MS0	通用输入输出端口P 2.4/PWM相位0高边输出/外部存储器选择0。
34	P0.3/TRST/A16/ADC _{BUSY}	通用输入输出端口P 0.3/JTAG测试端口输入，测试复位/ADC _{BUSY} 信号输出。
35	P2.5/PWM0 _L /MS1	通用输入输出端口P 2.5/PWM相位0低边输出/外部存储器选择1。
36	P2.6/PWM1 _H /MS2	通用输入输出端口P 2.6/PWM相位1高边输出/外部存储器选择2。
37	RST	复位输入，低电平有效。
38	P3.4/AD4/PWM2 _H /PLAI[12]	通用输入输出端口P 3.4/外部存储器接口/PWM相位2高边输出/可编程逻辑阵列输入单元12。
39	P3.5/AD5/PWM2 _L /PLAI[13]	通用输入输出端口P 3.5/外部存储器接口/PWM相位2低边输出/可编程逻辑阵列输入单元13。
40	IRQ0/P0.4/PWM _{TRIP} /PLAO[1]/MS1	多功能输入输出引脚。外部中断请求0，高电平有效/通用输入输出端口P 0.4/PWM触发外部输入/可编程逻辑阵列输出单元1/外部存储器选择1。
41	IRQ1/P0.5/ADC _{BUSY} /PLAO[2]/MS2	多功能输入输出引脚。外部中断请求1，高电平有效/通用输入输出端口P 0.5/ADC _{BUSY} 信号输出/可编程逻辑阵列输出单元2/外部存储器选择2。
42	P2.0/SPM9/PLAO[5]/CONV _{START}	串行复用端口。通用输入输出端口P 2.0/UART/可编程逻辑阵列输出单元5/ADC起始转换输入信号。
43	P0.7/ECLK/XCLK/SPM8/PLAO[4]	串行复用端口。通用输入输出端口P0.7/外部时钟信号输出/内部时钟发生器电路输入/UART/可编程逻辑阵列输出单元4。
44	XCLKO	晶体振荡反相器输出。
45	XCLKI	晶体振荡反相器输入、内部时钟发生器电路输入。

ADuC7019/20/21/22/24/25/26/27/28/29

引脚编号	引脚名称	描述
46	P3.6/AD6/PWM _{TRIP} /PLAI[14]	通用输入输出端口P 3.6/外部存储器接口/PWM安全关断/可编程逻辑阵列输入单元14。
47	P3.7/AD7/PWM _{SYNC} /PLAI[15]	通用输入输出端口P 3.7/外部存储器接口/PWM同步/可编程逻辑阵列输入单元15。
48	P2.7/PWM1 _L /MS3	通用输入输出端口P 2.7/PWM相位1低边输出/外部存储器选择3。
49	P2.1/ \overline{WS} /PWM0 _H /PLAO[6]	通用输入输出端口P 2.1/外部存储器写选通/PWM相位0高边输出/可编程逻辑阵列输出单元6。
50	P2.2/ \overline{RS} /PWM0 _L /PLAO[7]	通用输入输出端口P 2.2/外部存储器读选通/PWM相位0低边输出/可编程逻辑阵列输出单元7。
51	P1.7/SPM7/PLAO[0]	串行复用端口。通用输入输出端口P 1.7/UART, SPI/可编程逻辑阵列输出单元0。
52	P1.6/SPM6/PLAI[6]	串行复用端口。通用输入输出端口P 1.6/UART, SPI/可编程逻辑阵列输入单元6。
53	IOGND	GPIO地。通常连接到DGND。
54	IOV _{DD}	3.3 V电源。用于GPIO和片内稳压器输入。
55	P4.0/AD8/PLAO[8]	通用输入输出端口P 4.0/外部存储器接口/可编程逻辑阵列输出单元8。
56	P4.1/AD9/PLAO[9]	通用输入输出端口P 4.1/外部存储器接口/可编程逻辑阵列输出单元9。
57	P1.5/SPM5/PLAI[5]/IRQ3	串行复用端口。通用输入输出端口P 1.5/UART, SPI/可编程逻辑阵列输入单元5/外部中断请求3, 高电平有效。
58	P1.4/SPM4/PLAI[4]/IRQ2	串行复用端口。通用输入输出端口P 1.4/UART, SPI/可编程逻辑阵列输入单元4/外部中断请求2, 高电平有效。
59	P1.3/SPM3/PLAI[3]	串行复用端口。通用输入输出端口P 1.3/UART, I2C1/可编程逻辑阵列输入单元3。
60	P1.2/SPM2/PLAI[2]	串行复用端口。通用输入输出端口P 1.2/UART, I2C1/可编程逻辑阵列输入单元2。
61	P1.1/SPM1/PLAI[1]	串行复用端口。通用输入输出端口P 1.1/UART, I2C0/可编程逻辑阵列输入单元1。
62	P1.0/T1/SPM0/PLAI[0]	串行复用端口。通用输入输出端口P 1.0/定时器1输入/UART, I2C0/可编程逻辑阵列输入单元0。
63	P4.2/AD10/PLAO[10]	通用输入输出端口P 4.2/外部存储器接口/可编程逻辑阵列输出单元10。
64	P4.3/AD11/PLAO[11]	通用输入输出端口P 4.3/外部存储器接口/可编程逻辑阵列输出单元11。
65	P4.4/AD12/PLAO[12]	通用输入输出端口P 4.4/外部存储器接口/可编程逻辑阵列输出单元12。
66	P4.5/AD13/PLAO[13]	通用输入输出端口P 4.5/外部存储器接口/可编程逻辑阵列输出单元13。
67	REFGND	基准地。通常连接到AGND。
68	V _{REF}	2.5 V内部基准电压。在使用内部基准时必须外部对AGND连接一个0.47 μ F电容。
69	DAC _{REF}	DAC外部基准电压。范围: DACGND ~ DACV _{DD} 。
70	DACGND	DAC地。通常连接到AGND。
71, 72	AGND	模拟地。模拟电路的地基准点。
73, 74	AV _{DD}	3.3 V模拟电源。
75	DACV _{DD}	DAC 3.3 V电压源。必须连接到AV _{DD} 。
76	ADC11	单端或差分模拟输入11。
77	ADC0	单端或差分模拟输入0。
78	ADC1	单端或差分模拟输入1。
79	ADC2/CMP0	单端或差分模拟输入2/比较器正相输入。
80	ADC3/CMP1	单端或差分模拟输入3/比较器反相输入端。

ADuC7019/20/21/22/24/25/26/27/28/29

ADuC7028

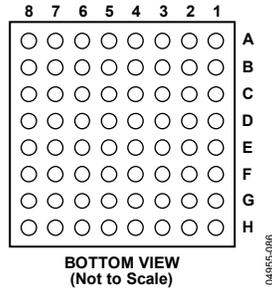


图 16. 64引脚BGA封装ADuC7028引脚配置

表13. ADuC7028引脚功能描述

引脚编号	引脚名称	描述
A1	ADC3/CMP1	单端或差分模拟输入3/比较器反相输入端。
A2	DACV _{DD}	DAC 3.3 V电压源。必须连接到AV _{DD} 。
A3	AV _{DD}	3.3 V模拟电源。
A4	AGND	模拟地。模拟电路的地基准点。
A5	DACGND	DAC地。通常连接到AGND。
A6	P4.2/PLAO[10]	通用输入输出端口P 4.2/可编程逻辑阵列输出单元10。
A7	P1.1/SPM1/PLAI[1]	串行复用端口。通用输入输出端口P 1.1/UART, I2C0/可编程逻辑阵列输入单元1。
A8	P1.2/SPM2/PLAI[2]	串行复用端口。通用输入输出端口P 1.2/UART, I2C1/可编程逻辑阵列输入单元2。
B1	ADC4	单端或差分模拟输入4。
B2	ADC2/CMP0	单端或差分模拟输入2/比较器正相输入。
B3	ADC1	单端或差分模拟输入1。
B4	DAC _{REF}	DAC外部基准电压。范围: DACGND ~ DACV _{DD} 。
B5	V _{REF}	2.5 V内部基准电压。在使用内部基准时必须连接至一个0.47 μF电容。
B6	P1.0/T1/SPM0/PLAI[0]	串行复用端口。通用输入输出端口P 1.0/定时器1输入/UART, I2C0/可编程逻辑阵列输入单元0。
B7	P1.4/SPM4/PLAI[4]/IRQ2	串行复用端口。通用输入输出端口P 1.4/UART, SPI/可编程逻辑阵列输入单元4/外部中断请求2, 高电平有效。
B8	P1.3/SPM3/PLAI[3]	串行复用端口。通用输入输出端口P 1.3/UART, I2C1/可编程逻辑阵列输入单元3。
C1	ADC6	单端或差分模拟输入6。
C2	ADC5	单端或差分模拟输入5。
C3	ADC0	单端或差分模拟输入0。
C4	P4.5/PLAO[13]	通用输入输出端口P 4.5/可编程逻辑阵列输出单元13。
C5	P4.3/PLAO[11]	通用输入输出端口P 4.3/可编程逻辑阵列输出单元11。
C6	P4.0/PLAO[8]	通用输入输出端口P 4.0/可编程逻辑阵列输出单元8。
C7	P4.1/PLAO[9]	通用输入输出端口P 4.1/可编程逻辑阵列输出单元9。
C8	IOGND	GPIO地。通常连接到DGND。
D1	ADCNEG	伪差分模式下ADC偏置点或反相模拟输入。必须连接到要转换信号的地。该偏置点必须在0 V ~ 1 V之间。
D2	GND _{REF}	ADC地基准电压。为了优化性能, 模拟电源应同IOGND和DGND分离。
D3	ADC7	单端或差分模拟输入7。
D4	P4.4/PLAO[12]	通用输入输出端口P 4.4/可编程逻辑阵列输出单元12。
D5	P3.6/PWM _{TRIP} /PLAI[14]	通用输入输出端口P 3.6/PWM安全关断/可编程逻辑阵列输入单元14。
D6	P1.7/SPM7/PLAO[0]	串行复用端口。通用输入输出端口P 1.7/UART, SPI/可编程逻辑阵列输出单元0。

ADuC7019/20/21/22/24/25/26/27/28/29

引脚编号	引脚名称	描述
D7	P1.6/SPM6/PLAI[6]	串行复用端口。通用输入输出端口P 1.6/UART, SPI/可编程逻辑阵列输入单元6。
D8	IOV _{DD}	3.3 V电源用于GPIO和片内稳压器输入。
E1	DAC3	DAC3电压输出。
E2	DAC2	DAC2电压输出。
E3	DAC1	DAC1电压输出。
E4	P3.0/PWM0 _H /PLAI[8]	通用输入输出端口P 3.0/PWM相位0高边输出/可编程逻辑阵列输入单元8。
E5	P3.2/PWM1 _H /PLAI[10]	通用输入输出端口P 3.2/PWM相位1高边输出/可编程逻辑阵列输入单元10。
E6	P1.5/SPM5/PLAI[5]/IRQ3	串行复用端口。通用输入输出端口P 1.5/UART, SPI/可编程逻辑阵列输入单元5/外部中断请求3, 高电平有效。
E7	P3.7/PWM _{SYNC} /PLAI[15]	通用输入输出端口P 3.7/PWM同步/可编程逻辑阵列输入单元15。
E8	XCLKI	晶体振荡反相器输入、内部时钟发生器电路输入。
F1	P4.6/PLAO[14]	通用输入输出端口P 4.6/可编程逻辑阵列输出单元14。
F2	TDI	JTAG测试端口输入, 测试数据输入。调试和下载访问。
F3	DAC0s	DAC0电压输出。
F4	P3.1/PWM0 _L /PLAI[9]	通用输入输出端口P 3.1/PWM相位0低边输出/可编程逻辑阵列输入单元9。
F5	P3.3/PWM1 _L /PLAI[11]	复位输入, 低电平有效。
F6	RST	通用输入输出端口P 3.3/PWM相位1低边输出/可编程逻辑阵列输入单元11。
F7	P0.7/ECLK/XCLK/SPM8/PLAO[4]	串行复用端口。通用输入输出端口P0.7/外部时钟信号输出/内部时钟发生器电路输入/UART/可编程逻辑阵列输出单元4。
F8	XCLKO	晶体振荡反相器输出。
G1	BM/P0.0/CMP _{OUT} /PLAI[7]	多功能输入输出引脚。引导模式。复位时如果BM为低电平, 则ADuC7028进入UART下载模式, 如果BM通过一个1 kΩ电阻上拉至高电平则执行程序代码/通用输入输出端口P0.0/电压比较器输出/可编程逻辑阵列输入单元7。
G2	P4.7/PLAO[15]	通用输入输出端口P 4.7/可编程逻辑阵列输出单元15。
G3	TMS	JTAG测试端口输入, 测试模式选择。调试和下载访问。
G4	TDO	JTAG测试端口输出, 测试数据输出。调试和下载访问。
G5	P0.3/TRST/ADC _{BUSY}	通用输入输出端口P 0.3/JTAG测试端口输入, 测试复位/ADC _{BUSY} 信号输出。
G6	P3.4/PWM2 _H /PLAI[12]	通用输入输出端口P 3.4/PWM相位2高边输出/可编程逻辑阵列输入单元12。
G7	P3.5/PWM2 _L /PLAI[13]	通用输入输出端口P 3.5/PWM相位2低边输出/可编程逻辑阵列输入单元13。
G8	P2.0/SPM9/PLAO[5]/CONV _{START}	串行复用端口。通用输入输出端口P 2.0/UART/可编程逻辑阵列输出单元5/ADC起始转换输入信号。
H1	P0.6/T1/MRST/PLAO[3]	多功能引脚, 复位后输出低电平。通用输入输出端口P 0.6/定时器1输入/上电复位输出/可编程逻辑阵列输出单元3。
H2	TCK	JTAG测试端口输入, 测试时钟。调试和下载访问。
H3	IOGND	GPIO地。通常连接到DGND。
H4	IOV _{DD}	3.3 V电源。用于GPIO和片内稳压器输入。
H5	LV _{DD}	片内稳压器2.6 V输出。该输出只能通过一个0.47μF电容器连接至DGND。
H6	DGND	内核逻辑地。
H7	IRQ0/P0.4/PWM _{TRIP} /PLAO[1]	多功能输入输出引脚。外部中断请求0, 高电平有效/通用输入输出端口P 0.4/PWM触发外部输入/可编程逻辑阵列输出单元1。
H8	IRQ1/P0.5/ADC _{BUSY} /PLAO[2]	多功能输入输出引脚。外部中断请求1, 高电平有效/通用输入输出端口P0.5/ADC _{BUSY} 信号输出/可编程逻辑阵列输出单元2。

ADuC7019/20/21/22/24/25/26/27/28/29

ADuC7029

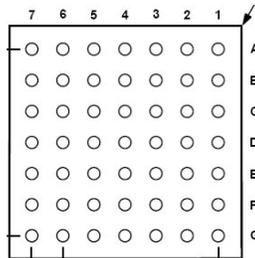


图 49. 引脚BGA封装引脚配置 (ADuC7029)

表14. ADuC7028引脚功能描述

引脚编号	引脚名称	描述
A1	ADC3/CMP1	单端或差分模拟输入3/比较器反相输入端。
A2	ADC1	单端或差分模拟输入1。
A3	ADC0	单端或差分模拟输入0。
A4	AV _{DD}	3.3 V模拟电源。
A5	V _{REF}	2.5 V内部基准电压。在使用内部基准时必须外部对AGND连接一个0.47 μF电容。
A6	P1.0/T1/SPM0/PLAI[0]	串行复用端口。通用输入输出端口P 1.0/定时器1输入/UART, I2C0/可编程逻辑阵列输入单元0。
A7	P1.1/SPM1/PLAI[1]	串行复用端口。通用输入输出端口P 1.1/UART, I2C0/可编程逻辑阵列输入单元1。
B1	ADC6	单端或差分模拟输入6。
B2	ADC5	单端或差分模拟输入5。
B3	ADC4	单端或差分模拟输入4。
B4	AGND	模拟地。模拟电路的地基准点。
B5	DAC _{REF}	DAC外部基准电压。范围: DACGND ~ DACV _{DD} 。
B6	P1.4/SPM4/PLAI[4]/IRQ2	串行复用端口。通用输入输出端口P 1.4/UART, SPI/可编程逻辑阵列输入单元4/外部中断请求2, 高电平有效。
B7	P1.3/SPM3/PLAI[3]	串行复用端口。通用输入输出端口P 1.3/UART, I2C1/可编程逻辑阵列输入单元3。
C1	GND _{REF}	ADC地基准电压。为了优化性能, 模拟电源应同IOGND和DGND分离。
C2	ADCNEG	伪差分模式下ADC偏置点或反相模拟输入。必须连接到要转换信号的地。该偏置点必须在0 V ~ 1 V之间。
C3	ADC2/CMP0	单端或差分模拟输入2/比较器正相输入。
C4	IOGND	GPIO地。通常连接到DGND。
C5	P1.2/SPM2/PLAI[2]	串行复用端口。通用输入输出端口P 1.2/UART, I2C1/可编程逻辑阵列输入单元2。
C6	P1.6/SPM6/PLAI[6]	串行复用端口。通用输入输出端口P 1.6/UART, SPI/可编程逻辑阵列输入单元6。
C7	P1.5/SPM5/PLAI[5]/IRQ3	串行复用端口。通用输入输出端口P 1.5/UART, SPI/可编程逻辑阵列输入单元5/外部中断请求3, 高电平有效。
D1	DAC0	DAC0电压输出。
D2	DAC3	DAC3电压输出。
D3	DAC1	DAC1电压输出。
D4	P3.3/PWM1 _L /PLAI[11]	通用输入输出端口P 3.3/PWM相位1低边输出/可编程逻辑阵列输入单元11。
D5	P3.4/PWM2 _H /PLAI[12]	通用输入输出端口P 3.4/PWM相位2高边输出/可编程逻辑阵列输入单元12。
D6	P3.6/PWM _{TRIP} /PLAI[14]	通用输入输出端口P 3.6/PWM安全关断/可编程逻辑阵列输入单元14。

ADuC7019/20/21/22/24/25/26/27/28/29

引脚编号	引脚名称	描述
D7	P1.7/SPM7/PLAO[0]	串行复用端口。通用输入输出端口P 1.7/UART, SPI/可编程逻辑阵列输出单元0。
E1	TMS	JTAG测试端口输入, 测试模式选择。调试和下载访问。
E2	BM/P0.0/CMP _{OUT} /PLAI[7]	多功能输入输出引脚。引导模式。复位时如果BM为低电平, 则ADuC7028进入UART下载模式, 如果BM通过一个1 kΩ电阻上拉至高电平则执行程序代码/通用输入输出端口P0.0/电压比较器输出/可编程逻辑阵列输入单元7。
E3	DAC2	DAC2电压输出。
E4	IOV _{DD}	3.3 V电源。用于GPIO和片内稳压器输入。
E5	P3.2/PWM1 _H /PLAI[10]	通用输入输出端口P 3.2/PWM相位1高边输出/可编程逻辑阵列输入单元10。
E6	P3.5/PWM2 _L /PLAI[13]	通用输入输出端口P 3.5/PWM相位2低边输出/可编程逻辑阵列输入单元13。
E7	P0.7/ECLK/XCLK/SPM8/PLAO[4]	串行复用端口。通用输入输出端口P0.7/外部时钟信号输出/内部时钟发生器电路输入/UART/可编程逻辑阵列输出单元4。
F1	TDI	JTAG测试端口输入, 测试数据输入。调试和下载访问。
F2	P0.6/T1/MRST/PLAO[3]	多功能引脚, 复位后输出低电平。通用输入输出端口P 0.6/定时器1输入/上电复位输出/可编程逻辑阵列输出单元3。
F3	IOGND	GPIO地。通常连接到DGND。
F4	P3.1/PWM0 _L /PLAI[9]	通用输入输出端口P 3.1/PWM相位0低边输出/可编程逻辑阵列输入单元9。
F5	P3.0/PWM0 _H /PLAI[8]	通用输入输出端口P 3.0/PWM相位0高边输出/可编程逻辑阵列输入单元8。
F6	RST	复位输入, 低电平有效。
F7	P2.0/SPM9/PLAO[5]/ $\overline{\text{CONV}}_{\text{START}}$	串行复用端口。通用输入输出端口P 2.0/UART/可编程逻辑阵列输出单元5/ADC起始转换输入信号。
G1	TCK	JTAG测试端口输入, 测试时钟。调试和下载访问。
G2	TDO	JTAG测试端口输出, 测试数据输出。调试和下载访问。
G3	LV _{DD}	片内稳压器2.6 V输出。该输出只能通过一个0.47μF电容器连接至DGND。
G4	DGND	内核逻辑地。
G5	P0.3/TRST/ADC _{BUSY}	通用输入输出端口P 0.3/JTAG测试端口输入, 测试复位/ADC _{BUSY} 信号输出。
G6	IRQ0/P0.4/PWM _{TRIP} /PLAO[1]	多功能输入输出引脚。外部中断请求0, 高电平有效/通用输入输出端口P 0.4/PWM触发外部输入/可编程逻辑阵列输出单元1。
G7	IRQ1/P0.5/ADC _{BUSY} /PLAO[2]	多功能输入输出引脚。外部中断请求1, 高电平有效/通用输入输出端口P 0.5/ADC _{BUSY} 信号输出/可编程逻辑阵列输出单元2。

典型工作特性

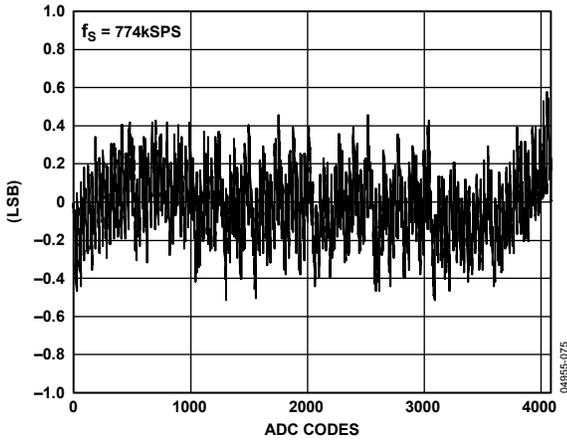


图 17. $f_s = 774$ kSPS 时典型 INL 误差

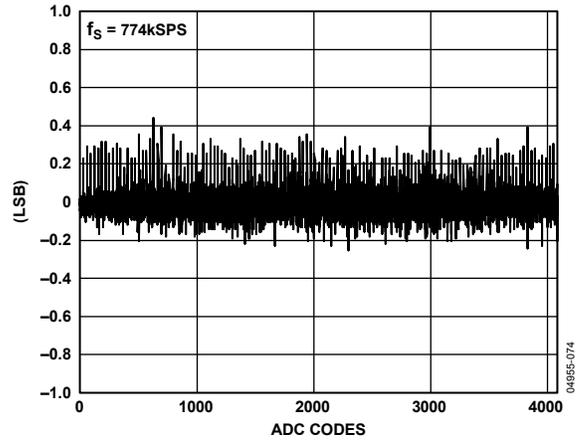


图 20. $f_s = 774$ kSPS 时典型 DNL 误差

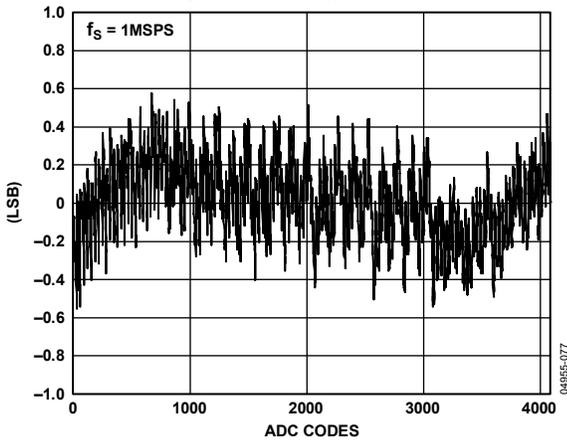


图 18. $f_s = 1$ MSPS 时典型 INL 误差

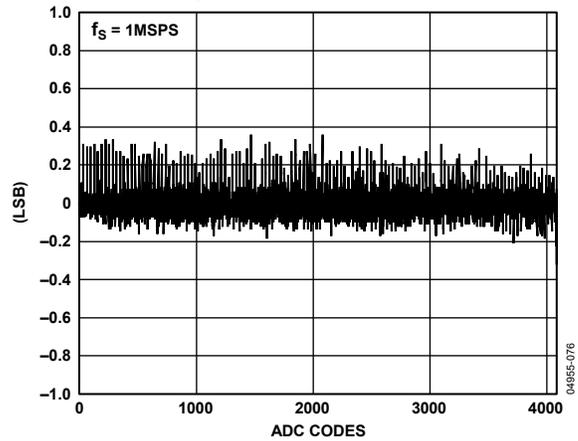


图 21. $f_s = 1$ MSPS 时典型 DNL 误差

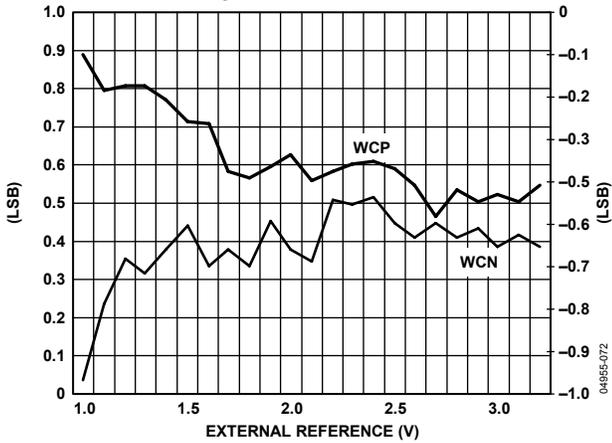


图 19. $f_s = 774$ kSPS 时典型最差 INL 误差与 V_{REF} 关系

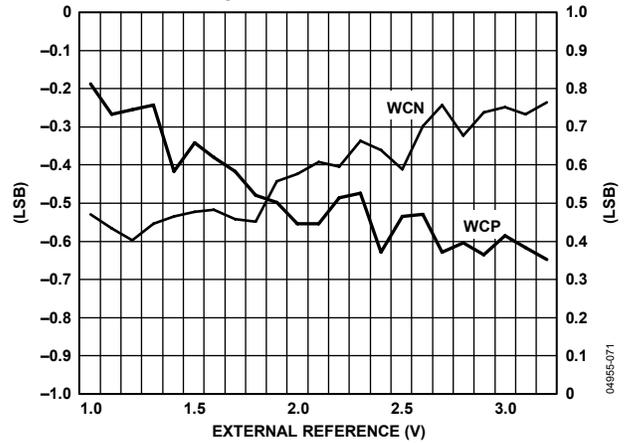


图 22. $f_s = 774$ kSPS 时典型最差 DNL 误差与 V_{REF} 关系

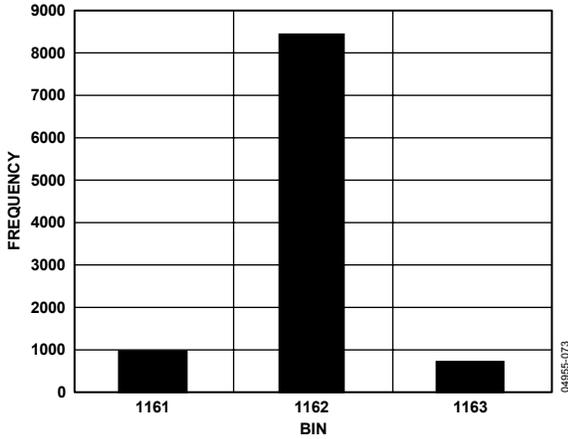


图 23. $f_s = 774$ kSPS, $V_{IN} = 0.7$ V 时代码直方图

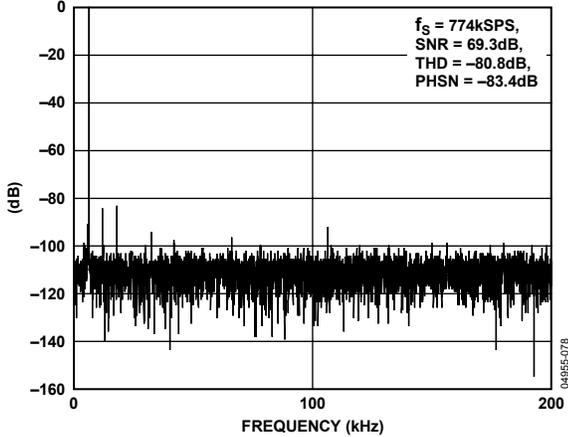


图24. $f_s = 774$ kSPS 时动态性能

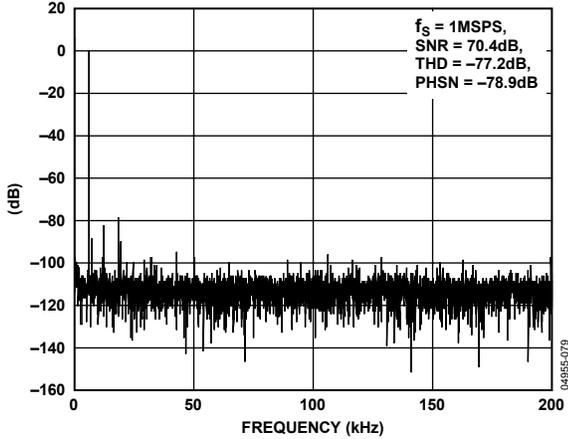


图 25. $f_s = 1$ MSPS 时动态性能

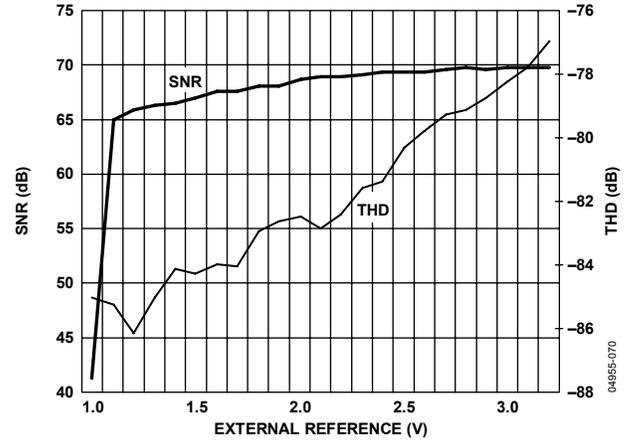


图26. 典型动态性能与 V_{REF} 关系

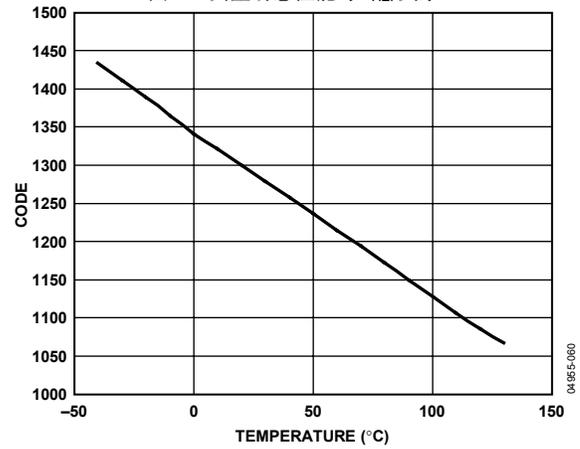


图 27. 片内温度传感器电压输出与温度的关系

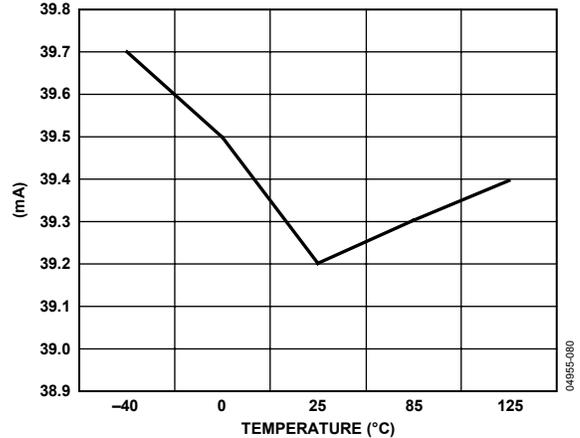


图 28. $CD = 0$ 时功耗/与温度的关系

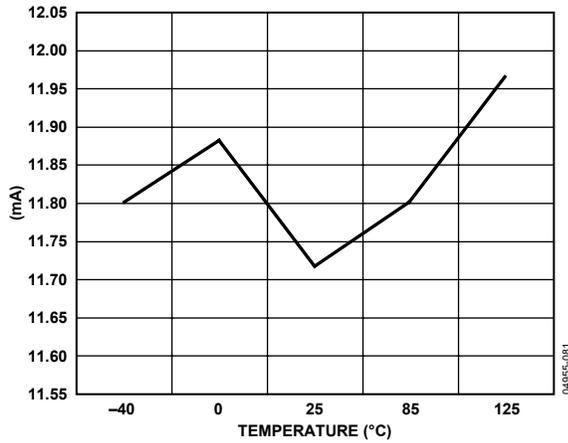


图 29. CD = 3时功耗与温度的关系

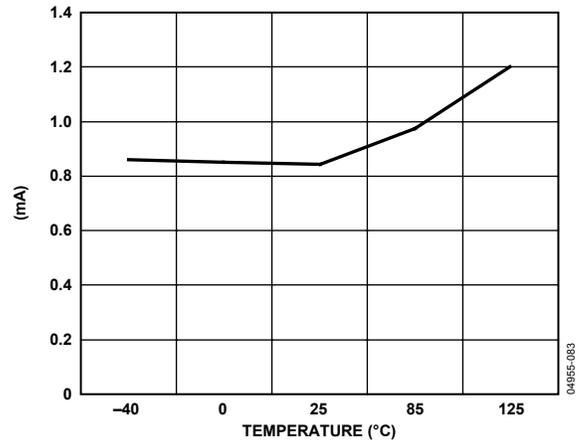


图 31. 睡眠模式下功耗与温度的关系

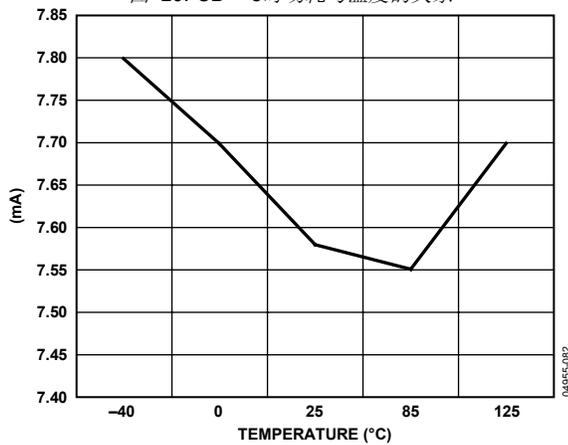


图 30. CD = 7时功耗与温度的关系

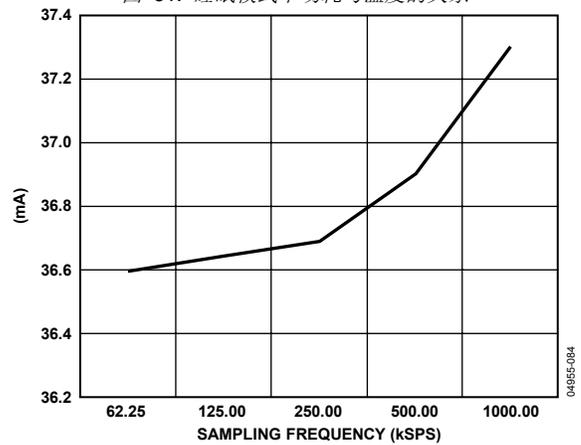


图 32. 功耗与采样频率的关系

术语

ADC技术规格

积分非线性(INL)

ADC转换结果编码偏离通过其传递函数断点的直线的最大偏差。ADC传递函数端点是指，在零点位置比第一个编码的跃变点低 $\frac{1}{2}$ LSB的点，以及在满刻度位置比最后一额编码的跃变点高 $\frac{1}{2}$ LSB的点。

微分非线性(DNL)

ADC中任意两个相邻码之间1 LSB的测量值与理想值之差。

偏移误差

第一个转换编码（从0000...000到 0000...001）的跃变点与理想点 $+\frac{1}{2}$ LSB之间的偏差。

增益误差

在偏移误差调零之后，最后一转换编码与理想AIN电压（满刻度减去1.5 LSB）之差。

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：

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

因此，对于一个12位的转换器来说，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。在零点误差和满刻度误差调零之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。

ARM7DMI内核概览

ARM7®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，并还有4个额外的特征：

- 支持16位的thumb指令集(T)
- 支持调试(D)
- 支持长乘(M)
- 包含一个支持嵌入式系统调试的EmbeddedICE模块(I)

THUMB模式(T)

一个ARM指令长度为32位。ARM7TDMI处理器支持第二个指令集，该指令集被压缩成16位，称为thumb指令集。用thumb指令集替代ARM指令集，可以更为快速的从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代码通常需要更多的指令。因此，如果更强调时效性，ARM代码更适合用来优化代码性能。
- 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位乘法(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。

嵌入式ICE(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。这是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。
- 未定义指令执行。
- 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，就是说得到更快响应时间，程序员可以定义中断为FIQ。

ARM寄存器

ARM7TDMI总共有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。

当一个异常发生时，一些标准的寄存器被替换成特定寄存器，而进入异常模式。所有的异常模式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33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。

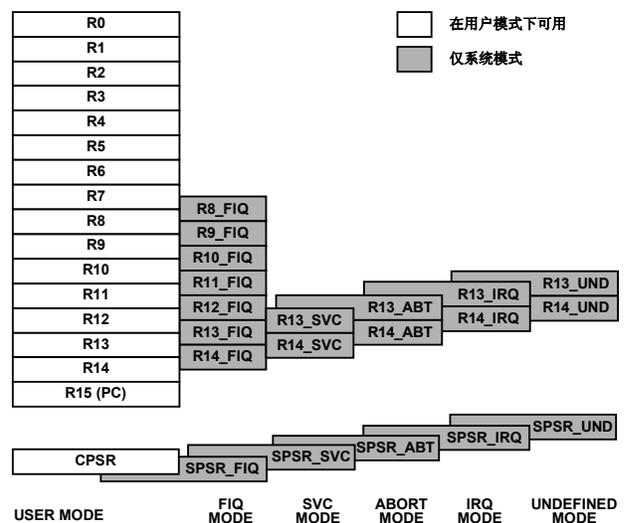


图 33. 寄存器结构图

更多关于编程模式和ARM7TDMI内核架构的信息可通过以下ARM公司的文件获得：

- DDI0029G, ARM7TDMI技术参考手册
- DDI-0100, ARM架构参考手册

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
- 最长指令执行需要的时间（最长的指令是LDM），该指令装载包括PC在内的所有寄存器
- 数据中止入口时间
- FIQ入口时间

在这个时间段的末尾，ARM7TDMI执行在0X1C（FIQ中断向量地址）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41.78MHz处理器时钟时，略微小于1.2微秒。

中断请求（IRQ）最大延迟计算也是类似的，但必须考虑到这样一个事实。即FIQ有更高优先级并且可以在任意一个时间段后延迟进入IRQ中断处理。如果没有使用LDM命令，这个时间可以减少为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模式下执行程序，在这个模式下时间可以减至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。

注意优先模式中（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

ADuC7019/20/21/22/24/25/26/27/28共有两个独立存储器模块：8KB的SRAM和64KB的片内Flash/EE存储器，其中62KB的片内Flash/EE可供用户使用，剩余2KB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34所示。

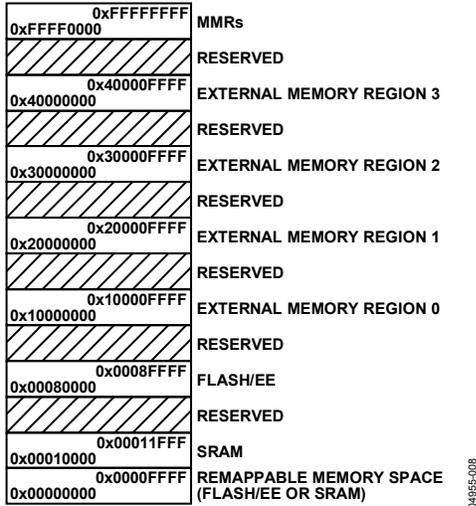


图34. 物理存储器映射图

注意默认情况下，复位之后，Flash/EE存储器被镜像到地址0x00000000。通过REMAPMMR的Bit0位置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34所示。

ADuC7019/20/21/22/24/25/26/27/28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。

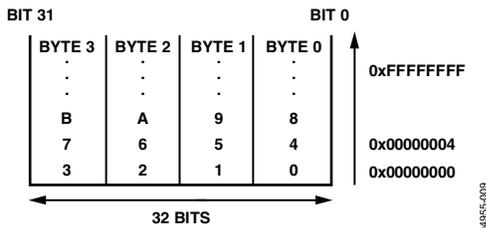


图 35. 从小到大顺序格式

FLASH/EE存储器

64KB的Flash/EE为32K×16位格式，其中31K×16位是用户空间，1K×16位用于芯片内核。Flash/EE的页面大小为512个字节。

62KB的片内Flash/EE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中执行程序时，为得到更高的存取速度，建议使用thumb代码。以thumb代码模式存取Flash/EE的最大速度为41.78MHz，而相应的以全ARM模式为20.89MHz。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2K×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41.78MHz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，并且我们可以通过间接寻址ARM7寄存器组来存取存储器映射寄存器空间。

存储器映射寄存器（MMR）空间为CPU和所有片上外设提供一个接口。除了内核寄存器，所有的寄存器都在MMR区域。图36所有的阴影区域是未占用或保留区域，不可以被用户软件访问。表15为所有的MMR存储器映射。

读取或写入一个MMR所需的存取时间取决于高级微控制器总线结构（AMBA）总线，这个总线用来访问外围设备。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、GPIO和PWM以外，其它所有ADuC7019/20/21/22/24/25/26/27/28的外围设备都连接到APB总线。

0xFFFFFFFF	
0xFFFFC3C	PWM
0xFFFFC00	
0xFFFF820	FLASH CONTROL INTERFACE
0xFFFF800	
0xFFFF46C	GPIO
0xFFFF400	
0xFFFF0B54	PLA
0xFFFF0B00	
0xFFFF0A14	SPI
0xFFFF0A00	
0xFFFF0948	I ² C1
0xFFFF0900	
0xFFFF0848	I ² C0
0xFFFF0800	
0xFFFF0730	UART
0xFFFF0700	
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	
0xFFFF0350	WAKE UP TIMER
0xFFFF0340	
0xFFFF0334	GENERAL PURPOSE TIMER
0xFFFF0320	
0xFFFF0310	TIMER 0
0xFFFF0300	
0xFFFF0238	REMAP AND SYSTEM CONTROL
0xFFFF0220	
0xFFFF0110	INTERRUPT CONTROLLER
0xFFFF0000	

图 36. 存储器映射寄存器

04985-010

表 15. 完整MMR列表

地址	名称	字节	访问类型	默认值	页
中断请求IRQ基地址 = 0xFFFF0000					
0x0000	IRQSTA	4	读	0x00000000	77
0x0004	IRQSIG ¹	4	读	0x00XXX000	77
0x0008	IRQEN	4	读/写	0x00000000	77
0x000C	IRQCLR	4	写	0x00000000	77
0x0010	SWICFG	4	写	0x00000000	78
0x0100	FIQSTA	4	读	0x00000000	78
0x0104	FIQSIG ¹	4	读	0x00XXX000	78
0x0108	FIQEN	4	读/写	0x00000000	78
0x010C	FIQCLR	4	写	0x00000000	78

¹ 取决于外部中断引脚P0.4、P0.5、P1.4和P1.5的电平。

系统控制基地址 = 0xFFFF0200

0x0220	REMAP ¹	1	读/写	0x00	50
0x0230	RSTSTA	1	读/写	0x01	50
0x0234	RSTCLR	1	写	0x00	50

¹ 取决于具体的器件型号。

定时器基地址 = 0xFFFF0300

0x0300	T0LD	2	读/写	0x0000	79
0x0304	T0VAL	2	读	0xFFFF	79
0x0308	T0CON	2	读/写	0x0000	79
0x030C	T0CLRI	1	写	0xFF	79
0x0320	T1LD	4	读/写	0x00000000	79
0x0324	T1VAL	4	读	0xFFFFFFFF	79
0x0328	T1CON	2	读/写	0x0000	79
0x032C	T1CLRI	1	写	0xFF	80
0x0330	T1CAP	4	读/写	0x00000000	80
0x0340	T2LD	4	读/写	0x00000000	80
0x0344	T2VAL	4	读	0xFFFFFFFF	80
0x0348	T2CON	2	读/写	0x0000	80
0x034C	T2CLRI	1	写	0xFF	81
0x0360	T3LD	2	读/写	0x0000	81
0x0364	T3VAL	2	读	0xFFFF	81
0x0368	T3CON	2	读/写	0x0000	81
0x036C	T3CLRI	1	写	0x00	81

锁相环PLL基地址 = 0xFFFF0400

0x0404	POWKEY1	2	写	0x0000	55
0x0408	POWCON	2	读/写	0x0003	55
0x040C	POWKEY2	2	写	0x0000	55
0x0410	PLLKEY1	2	写	0x0000	55
0x0414	PLLCON	1	读/写	0x21	55
0x0418	PLLKEY2	2	写	0x0000	55

PSM基地址 = 0xFFFF0440

0x0440	PSMCON	2	读/写	0x0008	52
0x0444	CMPCON	2	读/写	0x0000	53

ADuC7019/20/21/22/24/25/26/27/28/29

地址	名称	字节	访问类型	默认值	页
----	----	----	------	-----	---

基准电压基地址 = 0xFFFF0480

0x048C	REFCON	1	读/写	0x00	45
--------	--------	---	-----	------	----

ADC基地址 = 0xFFFF0500

0x0500	ADCCON	2	读/写	0x0600	41
0x0504	ADCCP	1	读/写	0x00	42
0x0508	ADCCN	1	读/写	0x01	42
0x050C	ADCSTA	1	读	0x00	43
0x0510	ADCDAT	4	读	0x00000000	43
0x0514	ADCRST	1	读/写	0x00	43
0x0530	ADCGN	2	读/写	0x0200	43
0x0534	ADCOF	2	读/写	0x0200	43

DAC基地址 = 0xFFFF0600

0x0600	DAC0CON	1	读/写	0x00	51
0x0604	DAC0DAT	4	读/写	0x00000000	51
0x0608	DAC1CON	1	读/写	0x00	51
0x060C	DAC1DAT	4	读/写	0x00000000	51
0x0610	DAC2CON	1	读/写	0x00	51
0x0614	DAC2DAT	4	读/写	0x00000000	51
0x0618	DAC3CON	1	读/写	0x00	51
0x061C	DAC3DAT	4	读/写	0x00000000	51

UART基地址 = 0xFFFF0700

0x0700	COMTX	1	读/写	0x00	65
	COMRX	1	读	0x00	65
	COMDIV0	1	读/写	0x00	65
0x0704	COMIEN0	1	读/写	0x00	65
	COMDIV1	1	读/写	0x00	65
0x0708	COMIID0	1	读	0x01	66
0x070C	COMCON0	1	读/写	0x00	66
0x0710	COMCON1	1	读/写	0x00	66
0x0714	COMSTA0	1	读	0x60	66
0x0718	COMSTA1	1	读	0x00	67
0x071C	COMSCR	1	读/写	0x00	67
0x0720	COMIEN1	1	读/写	0x04	67
0x0724	COMIID1	1	读	0x01	67
0x0728	COMADR	1	读/写	0xAA	68
0x072C	COMDIV2	2	读/写	0x0000	67

地址	名称	字节	访问类型	默认值	页
----	----	----	------	-----	---

I2C0基地址 = 0xFFFF0800

0x0800	I2C0MSTA	1	读/写	0x00	70
0x0804	I2C0SSTA	1	读	0x01	70
0x0808	I2C0SRX	1	读	0x00	71
0x080C	I2C0STX	1	写	0x00	71
0x0810	I2C0MRX	1	读	0x00	71
0x0814	I2C0MTX	1	写	0x00	71
0x0818	I2C0CNT	1	读/写	0x00	71
0x081C	I2C0ADR	1	读/写	0x00	71
0x0824	I2C0BYTE	1	读/写	0x00	71
0x0828	I2C0ALT	1	读/写	0x00	72
0x082C	I2C0CFG	1	读/写	0x00	72
0x0830	I2C0DIV	2	读/写	0x1F1F	73
0x0838	I2C0ID0	1	读/写	0x00	73
0x083C	I2C0ID1	1	读/写	0x00	73
0x0840	I2C0ID2	1	读/写	0x00	73
0x0844	I2C0ID3	1	读/写	0x00	73
0x0848	I2C0CCNT	1	读/写	0x01	73
0x084C	I2C0FSTA	2	读/写	0x0000	73

I2C1基地址 = 0xFFFF0900

0x0900	I2C1MSTA	1	读/写	0x00	70
0x0904	I2C1SSTA	1	读	0x01	70
0x0908	I2C1SRX	1	读	0x00	71
0x090C	I2C1STX	1	写	0x00	71
0x0910	I2C1MRX	1	读	0x00	71
0x0914	I2C1MTX	1	写	0x00	71
0x0918	I2C1CNT	1	读/写	0x00	71
0x091C	I2C1ADR	1	读/写	0x00	71
0x0924	I2C1BYTE	1	读/写	0x00	71
0x0928	I2C1ALT	1	读/写	0x00	72
0x092C	I2C1CFG	1	读/写	0x00	72
0x0930	I2C1DIV	2	读/写	0x1F1F	73
0x0938	I2C1ID0	1	读/写	0x00	73
0x093C	I2C1ID1	1	读/写	0x00	73
0x0940	I2C1ID2	1	读/写	0x00	73
0x0944	I2C1ID3	1	读/写	0x00	73
0x0948	I2C1CCNT	1	读/写	0x01	73
0x094C	I2C1FSTA	2	读/写	0x0000	73

SPI基地址 = 0xFFFF0A00

0x0A00	SPISTA	1	读	0x00	68
0x0A04	SPIRX	1	读	0x00	69
0x0A08	SPLITX	1	写	0x00	69
0x0A0C	SPIDIV	1	读/写	0x1B	69
0x0A10	SPICON	2	读/写	0x0000	69

ADuC7019/20/21/22/24/25/26/27/28/29

地址	名称	字节	访问类型	默认值	页
PLA基地址 = 0xFFFF0B00					
0x0B00	PLAELM0	2	读/写	0x0000	74
0x0B04	PLAELM1	2	读/写	0x0000	74
0x0B08	PLAELM2	2	读/写	0x0000	74
0x0B0C	PLAELM3	2	读/写	0x0000	74
0x0B10	PLAELM4	2	读/写	0x0000	74
0x0B14	PLAELM5	2	读/写	0x0000	74
0x0B18	PLAELM6	2	读/写	0x0000	74
0x0B1C	PLAELM7	2	读/写	0x0000	74
0x0B20	PLAELM8	2	读/写	0x0000	74
0x0B24	PLAELM9	2	读/写	0x0000	74
0x0B28	PLAELM10	2	读/写	0x0000	74
0x0B2C	PLAELM11	2	读/写	0x0000	74
0x0B30	PLAELM12	2	读/写	0x0000	74
0x0B34	PLAELM13	2	读/写	0x0000	74
0x0B38	PLAELM14	2	读/写	0x0000	74
0x0B3C	PLAELM15	2	读/写	0x0000	74
0x0B40	PLACLK	1	读/写	0x00	75
0x0B44	PLAIRQ	4	读/写	0x00000000	75
0x0B48	PLAADC	4	读/写	0x00000000	76
0x0B4C	PLADIN	4	读/写	0x00000000	76
0x0B50	PLADOUT	4	读	0x00000000	76
0x0B54	PLALCK	1	写	0x00	76

地址	名称	字节	访问类型	默认值	页
GPIO基地址 = 0xFFFFF400					
0xF400	GP0CON	4	读/写	0x00000000	63
0xF404	GP1CON	4	读/写	0x00000000	63
0xF408	GP2CON	4	读/写	0x00000000	63
0xF40C	GP3CON	4	读/写	0x00000000	63
0xF410	GP4CON	4	读/写	0x00000000	63
0xF420	GP0DAT	4	读/写	0x000000XX	64
0xF424	GP0SET	4	写	0x000000XX	64
0xF428	GP0CLR	4	写	0x000000XX	64
0xF42C	GP0PAR	4	读/写	0x20000000	63
0xF430	GP1DAT	4	读/写	0x000000XX	64
0xF434	GP1SET	4	写	0x000000XX	64
0xF438	GP1CLR	4	写	0x000000XX	64
0xF43C	GP1PAR	4	读/写	0x00000000	63
0xF440	GP2DAT	4	读/写	0x000000XX	64
0xF444	GP2SET	4	写	0x000000XX	64
0xF448	GP2CLR	4	写	0x000000XX	64
0xF450	GP3DAT	4	读/写	0x000000XX	64
0xF454	GP3SET	4	写	0x000000XX	64
0xF458	GP3CLR	4	写	0x000000XX	64
0xF460	GP4DAT	4	读/写	0x000000XX	64
0xF464	GP4SET	4	写	0x000000XX	64
0xF468	GP4CLR	4	写	0x000000XX	64

外部存储器基地址 = 0xFFFFF000					
地址	名称	字节	访问类型	默认值	页
0xF000	XMCFG	1	读/写	0x00	83
0xF010	XM0CON	1	读/写	0x00	83
0xF014	XM1CON	1	读/写	0x00	83
0xF018	XM2CON	1	读/写	0x00	83
0xF01C	XM3CON	1	读/写	0x00	83
0xF020	XM0PAR	2	读/写	0x70FF	83
0xF024	XM1PAR	2	读/写	0x70FF	83
0xF028	XM2PAR	2	读/写	0x70FF	83
0xF02C	XM3PAR	2	读/写	0x70FF	83

Flash/EE基地址 = 0xFFFFF800					
地址	名称	字节	访问类型	默认值	页
0xF800	FEESTA	1	读	0x20	47
0xF804	FEEMOD	2	读/写	0x0000	47
0xF808	FEECON	1	读/写	0x07	48
0xF80C	FEEDAT	2	读/写	0XXXXX	48
0xF810	FEEADR	2	读/写	0x0000	48
0xF818	FEESIGN	3	读	0FFFFFFF	48
0xF81C	FEEPRO	4	读/写	0x00000000	48
0xF820	FEEHIDE	4	读/写	0FFFFFFFFF	48

PWM基地址 = 0xFFFFFC00					
地址	名称	字节	访问类型	默认值	页
0xFC00	PWMCON	2	读/写	0x0000	61
0xFC04	PWMSTA	2	读/写	0x0000	61
0xFC08	PWMDAT0	2	读/写	0x0000	62
0xFC0C	PWMDAT1	2	读/写	0x0000	62
0xFC10	PWMCFG	2	读/写	0x0000	62
0xFC14	PWMCH0	2	读/写	0x0000	62
0xFC18	PWMCH1	2	读/写	0x0000	62
0xFC1C	PWMCH2	2	读/写	0x0000	62
0xFC20	PWMEN	2	读/写	0x0000	62
0xFC24	PWMDAT2	2	读/写	0x0000	62

ADC电路概览

模拟-数字转换器(ADC)集成了一个快速多通道12位ADC。该器件工作电压为2.7V至3.6V,当时钟源为41.78MHz时,其吞吐量最高可以达到1MSPS.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
- 伪差分模式,用于任何单端信号输入,可以较好的抑制共模信号

当工作在单端模式或伪差分模式下时,ADC模拟输入范围为0V至 V_{REF} 。在全差分模式下,输入信号共模电压 V_{CM} 在0V至 AV_{DD} 之间、最大幅值不超过 $2V_{REF}$ (见图37)

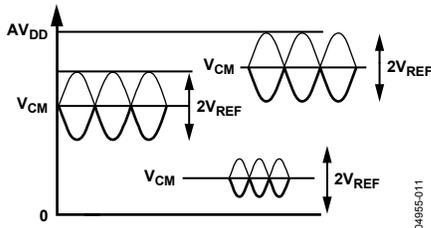


图37. 全差分模式下平衡信号

片内提供一个高精度、低漂移、工厂校准的2.5V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”一节所述。

单次或连续转换模式可由软件启动。外部 $\overline{CONV_START}$ 引脚、片内PLA的输出、定时器0或定时器1的溢出等也可用于重复触发ADC转换。

另外从片内带隙基准电压输出的与绝对温度成比例的电压也可以通过ADC前端多路复用器有连接到ADC,构成一个附加的ADC输入通道,这样就可以很方便的形成一个内部温度传感器通道,用于测量芯片温度,测量精度典型值为 $\pm 3^{\circ}C$ 。

转换功能

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0V至 V_{REF} ,输出编码为标准二进制编码

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

$$2.5V/4096 = 0.61mV \text{ 或}$$

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

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/输出转换特性如图38所示。

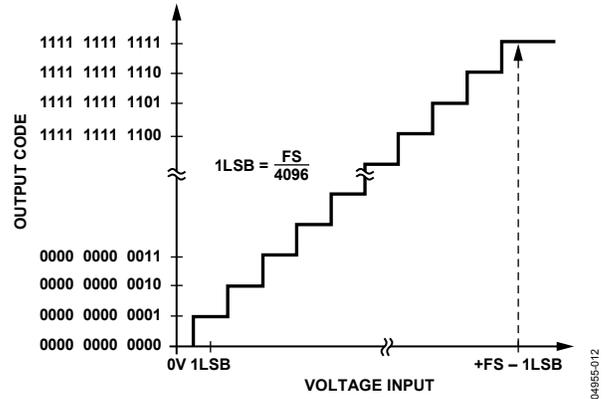


图38 伪差分或单端模式下ADC转换功能

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 $+V_{REF}$ p-p(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 $(V_{IN+} + V_{IN-})/2$ 。从而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 $V_{REF} = 2.5V$ 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 $2 \times 2.5V/4096 = 1.22mV$ 。输出结果为 ± 11 位,但是这个结果向右移了一位。当用C语言时,允许ADCDAT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/输出转换特性如图39所示。

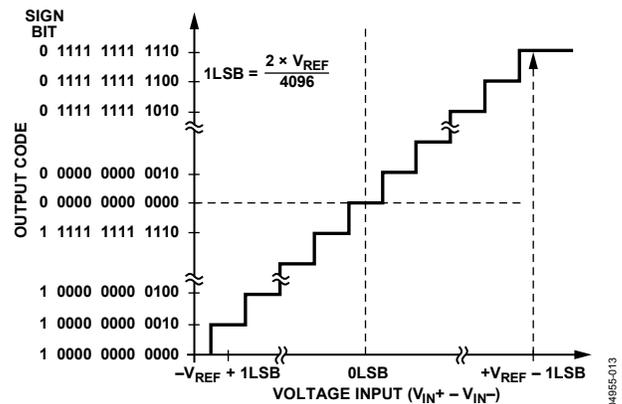


图39 差分模式下ADC转换功能

典型操作

一旦配置好ADC控制寄存器和通道选择寄存器，ADC开始转化模拟输入，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40所示寄存器中的16位至27位，同样地，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，结果是标准二进制格式表示。

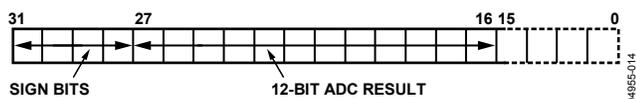


图40. ADC转换结果格式

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640μA。使用内部基准电压源电流要增加140μA。转换过程中，额外电流是0.3μA乘以采样频率（单位kHz）。图32显示了功耗与ADC采样频率的关系。

时序

ADC时序如图41所示。用户可以控制ADC时钟速度和ADCCON MMR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）个数可以设定为19，这样采样速率为774 KSPS。而温度传感器的转换，ADC采集时间被自动设定为16个时钟，并且ADC时钟设为32分频。当使用包括温度传感器的多通道转换时，在读取了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。

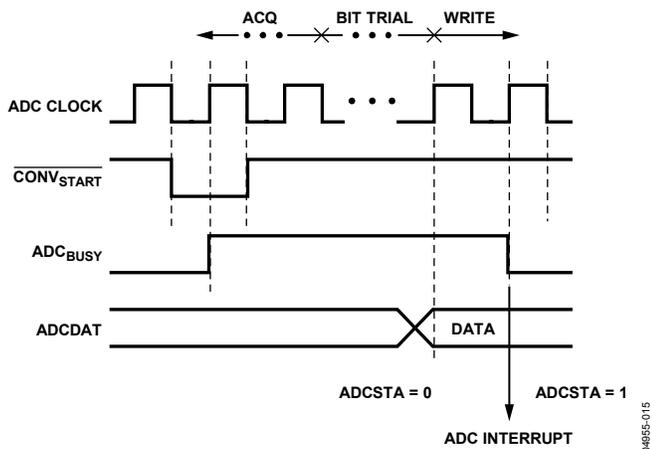


图41 ADC时序

ADuC7019

ADuC7019和ADuC7020相比只是差一个缓冲ADC通道ADC3，另外它只有三个DAC。第四个DAC的输出缓冲在内部连接到ADC3的通道，如图42所示。

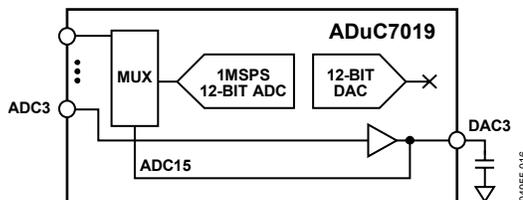


图42 ADC3缓冲输入

注意：DAC3这个输出引脚必须和AGND之间连一个10 nF的电容。并且此通道只能用来测量直流电压。对这个通道而言，ADC校准也是必要的。

MMR接口

这部分我们介绍控制和配置ADC的8个MMR。

ADCCON寄存器

名称	地址	默认值	访问
ADCCON	0xFFFF0500	0x0600	读/写

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MMR如表16所示。

ADuC7019/20/21/22/24/25/26/27/28/29

表16. ADCCON MMR位分配

位	值	描述
15:13		保留。
12:10		ADC时钟速度。
	000	fADC/1。该分频器用于在外部时钟频率小于41.78MHz时获得1MSPS ADC。
	001	fADC/2 (默认值)。
	010	fADC/4。
	011	fADC/8。
	100	fADC/16。
	101	fADC/32。
9:8		ADC采集时间。
	00	2个时钟。
	01	4个时钟。
	10	8个时钟(默认值)。
	11	16个时钟。
7		转换启动使能。
		置1, 启动任意类型的转换。清0禁用转换启动(清0该位不能使正进行的连续转换停止)。
6		使能ADC _{BUSY} 。
		置1, 使能ADC _{BUSY} 引脚。清0则禁用该引脚。
5		ADC电源控制。
		置1, 置ADC于正常模式(ADC在上电至少5微秒之后才能正确转换)。清0, ADC置于掉电模式。
4:3		转换模式。
	00	单端模式。
	01	差分模式。
	10	伪差分模式。
	11	保留。
2:0		转换类型。
	000	$\overline{\text{CONV}}_{\text{START}}$ 引脚输入作为转换使能信号。
	001	定时器1输出作为转换使能信号。
	010	定时器0输出作为转换使能信号
	011	单次软件转换。在转换后设为000(注意: 为了避免 $\overline{\text{CONV}}_{\text{START}}$ 引脚再一次触发转换, ADDCON MMR的第7位在启动单次软件转换后应清0)。
	100	连续软件转换。
	101	PLA转换。
	其他	保留。

ADCCP寄存器

名称	地址	默认值	访问
ADCCP	0xFFFF0504	0x00	读/写

ADCCP是ADC正向通道选择寄存器。该MMR如表17所示。

表17. ADCCP¹ MMR位分配

位	值	描述
7:5		保留。
4:0		正向通道选择位。
	00000	ADC0
	00001	ADC1
	00010	ADC2
	00011	ADC3
	00100	ADC4
	00101	ADC5
	00110	ADC6
	00111	ADC7
	01000	ADC8
	01001	ADC9
	01010	ADC10
	01011	ADC11
	01100	DAC0/ADC12
	01101	DAC1/ADC13
	01110	DAC2/ADC14
	01111	DAC3/ADC15
	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。
	Others	保留。

¹ ADC和DAC通道数取决于器件型号, 具体请参考订购指南。

ADCCN寄存器

名称	地址	默认值	访问
ADCCN	0xFFFF0508	0x01	读/写

ADCCN是一个ADC反向通道选择寄存器。该MMR如表18所示。

表18. ADCCN MMR位分配

位	值	描述
7:5		保留。
4:0		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	ADC11
	01100	DAC0/ADC12
	01101	DAC1/ADC13
	01110	DAC2/ADC14
	01111	DAC3/ADC15
	10000	内部基准电压（自诊断特性）
	Others	保留。

ADCSTA 寄存器

名称	地址	默认值	访问
ADCSTA	0xFFFF050C	0x00	读

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Bit0（ADCReady），表示ADC的转换状态。在一次ADC转换完成后该位置1，并且产生一个ADC中断。当读取ADCDAT MMR时，该位自动清0。在ADC进行转换时，也可以通过外部ADC_{BUSY}引脚读取ADC的工作状态。在转换期间，该引脚上为高电平；当转换结束后，ADC_{BUSY}引脚变为低电平。如果通过ADCCON寄存器使能，则可以在P0.5引脚输出ADC_{BUSY}的状态（参见“通用输入/输出”部分）。

ADCDAT 寄存器

名称	地址	默认值	访问
ADCDAT	0xFFFF0510	0x00000000	读

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40所示。

ADCRST 寄存器

名称	地址	默认值	访问
ADCRST	0xFFFF0514	0x00	读/写

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。

ADCGN 寄存器

名称	地址	默认值	访问
ADCGN	0xFFFF0530	0x0200	读/写

ADCGN是一个10位增益校准寄存器。

ADCOF 寄存器

名称	地址	默认值	访问
ADCOF	0xFFFF0534	0x0200	读/写

ADCOF是一个10位偏移校准寄存器。

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。

差分模式

ADuC7019/20/21/22/24/25/26/27/28都包含一个基于两个容性DAC的逐次逼近型ADC。图43和图44分别为ADC采样阶段和转换阶段简图。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43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。

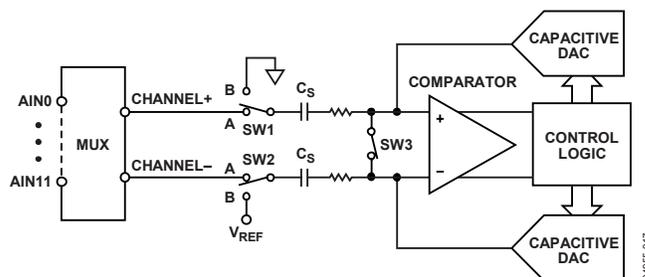


图43 ADC采样阶段

当ADC启动转换，如图44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，两个输入将会断开。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。

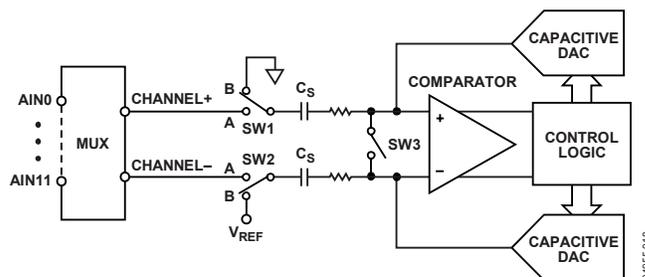


图44 ADC转换阶段

伪差分模式

在伪差分模式中，模拟输入负通道（channel-）连接在ADuC7019/20/21/22/24/25/26/27/28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} 。

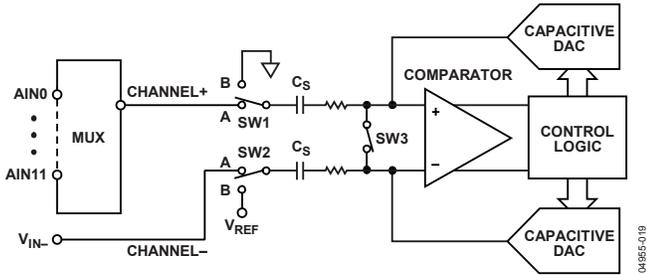


图45 伪差分模式下ADC

单端模式

单端模式下，SW2总是内部接地。 V_{IN-} 引脚可悬空。 V_{IN+} 引脚上输入信号的范围为0V至 V_{REF} 。

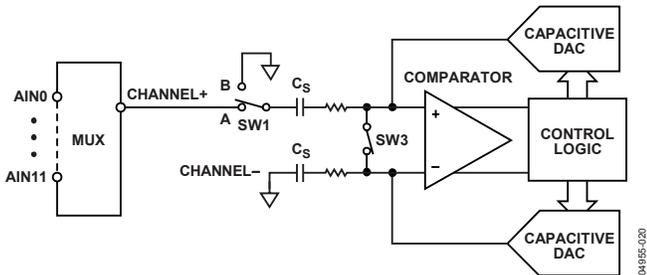


图46 单端模式ADC

模拟输入结构

ADC模拟输入结构等效电路如图47所示，图中4个二极管为模拟输入提供ESD保护；注意，这里一定要确保模拟输入信号不要超过电源电压300mV，否则将使二极管正向偏置，导通至衬底。这些二极管导通电流可达到10mA，而不会导致不可恢复的器件损坏。

图47中，电容C1典型值为4pF，可作为接地的引脚电容电阻是由开关阻抗构成的集总元件。电阻典型值为100Ω左右；电容C2为ADC采样电容，典型值为16pF。

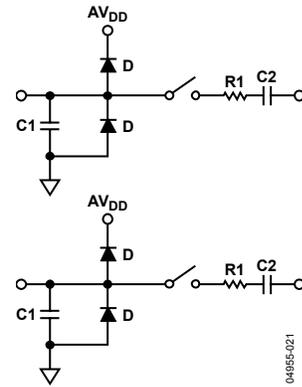


图47 等效模拟输入电路转换阶段：开关打开，采样阶段：开关闭闭

在交流应用时，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48和图49为ADC前端的示例。

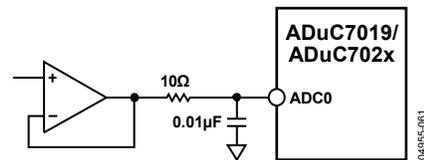


图48 带缓冲的单端模式/伪差分模式输入

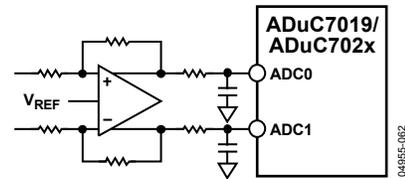


图49 带缓冲的差分模式输入

当不使用放大器来驱动模拟输入时，源阻抗应限制在1kΩ以内。最大的源阻抗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，从而使ADC性能下降。

驱动模拟输入

该ADC既可以采用内部基准电压也可以采用外部基准电压。在差分工作模式下，共模输入信号 V_{CM} 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19给出了一些计算出的 V_{CM} 的最大值和最小值。

表19 V_{CM}的范围

AV _{DD}	V _{REF}	V _{CM} 最小值	V _{CM} 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偏移寄存器(ADCOF)和增益系数寄存器(ADCGN)为出厂设定值时, 单次操作的端点误差和线性度可以达到较优的效果(参见“技术规格”部分)。如果需要进行系统校准, 可以通过修改偏移和增益系数的默认值来改善端点误差。但需注意, 任何对ADCOF和ADCGN出厂值的修改都会降低ADC的线性性能。

对于系统偏移误差校正, 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)读值从0变为1。如果ADCDAT中的值大于1, 应该降低ADCOF中的值, 直到ADCDAT读值从0变为1。这种数字化校正偏移误差的分辨率可以达到0.25 LSB, 范围可达V_{REF}的±3.125%。

对于系统增益误差校正, 必须将ADC通道输入级连接在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ADC转换结果数据(ADCDAT)读值从4094变为4095。如果ADCDAT中的值小于4095, 应该增加ADCGN中的值直到ADCDAT读值从4094变为4095。同偏移误差校正一样, 这种增益校正误差的分辨率也可以达0.25 LSB, 范围可达V_{REF}的±3%。

温度传感器

ADuC7019/20/21/22/24/25/26/27/28 提供从片内带隙基准电压输出并与绝对温度成比例的电压。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, 这样就可以很方便的形成一个内部温度传感器通道, 用于测量芯片温度, 测量精度典型值为±3°C。

以下示例显示了如何使用内部温度传感器:

```
int main(void)
{
float a = 0;
short b;
```

```
ADCCON = 0x20; // power-on the ADC
delay(2000);
ADCCP = 0x10; // Select Temperature Sensor as an
// input to the ADC
REFCON = 0x01; // connect internal 2.5V reference
// to Vref pin
ADCCON = 0xE4; // continuous conversion
while(1)
{
while (!ADCSTA){};
// wait for end of conversion
b = (ADCDAT >> 16);
// To calculate temperature in °C, use the formula:
a = 0x525 - b;
// ((Temperature = 0x525 – Sensor Voltage) / 1.3)
a /= 1.3;
b = floor(a);
printf("Temperature:%d °C\n",b);
}
return 0;
}
```

带隙基准电压源

ADuC7019/20/21/22/24/25/26/27/28均内置有一个2.5V的片内带隙基准电压, 它可用于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 μF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, 一般在作为外部电路基准时需要连接一个外部缓冲器; 在使用外部基准电压源时, 可以通过软件编程设置, 注意内部基准电压不能禁用。因此, 外部基准电压源必须有足够强的驱动能力以抑制内部基准电压源。

REFCON寄存器

名称	地址	默认值	访问
REFCON	0xFFFF048C	0x00	读/写

带隙基准电压接口包括一个8位MMR REFCON 如表20所示

表20 REFCON MMR 位分配

位	描述
7:1	保留。
0	内部基准电压输出使能。置1为内部2.5V基准电压源接在V _{REF} 上, 基准电压源可用于外部元件, 但需要增加缓冲; 清0为内部2.5V基准电压源断开与V _{REF} 管脚的连接。

非易失FLASH/EE存储器

ADuC7019/20/21/22/24/25/26/27/28片内集成Flash/EE存储器技术，为用户提供可在线重编程的非易失性存储空间。

同EEPROM类似，Flash存储器也支持在系统字节编程，尽管它在写新的数据之前需要先擦除。擦除按页进行。因而Flash存储器也常常且更准确地被称作Flash/EE存储器。

总的来讲，Flash/EE存储器是比较理想的存储器件，它具有非易失性、可在线编程、高密度性、低功耗等优点。ADuC7019/20/21/22/24/25/26/27/28的内置Flash/EE存储器技术，使得用户可在线升级程序代码空间，而不需要在远程操作节点替换一次性可编程(OTP)器件。

每个器件均含有一个64KB的Flash/EE存储器阵列，其中低62KB是提供给用户存储程序代码或非易失性数据，高2KB含有永久性嵌入式固件，支持在线串行下载。这2KB的嵌入式固件同时包括上电配置程序，用于将出厂标定参数下载到各种已校准外设，如ADC、温度传感器、带隙基准电压源等。这2KB嵌入式固件不能为用户代码访问。

Flash/EE存储器可靠性

器件上的Flash/EE存储器阵列有两项关键特征：Flash/EE周期耐久性和数据的保持力。

耐久性是对Flash/EE存储器多次编程、读取、擦除周期的次数的量化。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. 初始页面擦除时序。
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）。

在可靠性验证时，Flash/EE存储器中的三页（顶、中、底）的每半个字（16位宽）从0x0000至0xFFFF，可循环存取10,000次。正如表1所示，它的耐久性是根据JEDEC保持力寿命规范A117在-40°C至+125°C的工业温度范围内测出来的。这个结果使得对电源温度的最小耐久性系数超过10,000周期。

保持力是对Flash/EE存储器维持可编程数据时间能力的量化。同样地，这些产品是根据正式JEDEC保持力寿命规范（A117）在特定的结点温度（ $T_j = 85^\circ\text{C}$ ）评定的。作为评定程序的一部分，Flash/EE存储器循环到指定的耐力极限，正如前面所述，还拥有保存数据的特点。这意味着Flash/EE存储器保证其保留的数据完全达到规范保留其每次闪光/电擦除存储器编程。此外，需要注意的是基于一个0.6 eV激活能的保持力寿命随 T_j 递减，如图50所示。

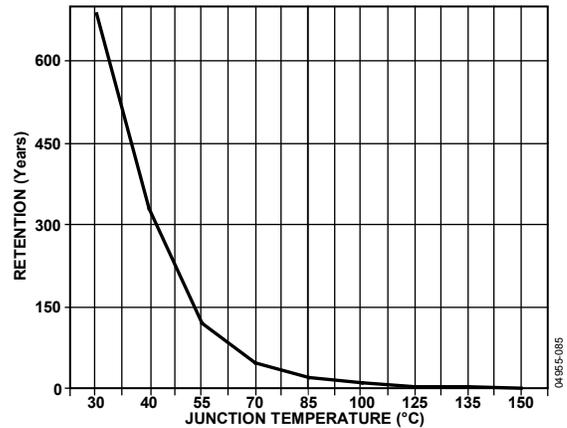


图50. Flash/EE存储器数据保持力

编程

通过串行下载模式或JTAG模式，Flash/EE存储器中的62KB区域可在线编程。

串行下载模式（在线编程）

ADuC7019/20/21/22/24/25/26/27/28通过标准的通用异步串口(UART)或I²C端口下载代码。如果BM引脚通过一个1KΩ的外部下拉电阻下拉到低电平，这样在复位或上电周期后，ADuC7026就会进入到串行下载模式。一旦进入这个模式，用户就可以下载代码到Flash/EE的全部62KB中，尽管该器件在其目标应用硬件为在线状态。可执行的PC机串行下载作为经由通用异步串口下载一部分是由开发系统提供的。The AN-806应用笔记介绍了通用异步串口(UART)和I²C端口下载代码协议。

JTAG模式

JTAG协议使用片内JTAG接口进行代码下载和调试。

安全性

用户可用的62KB区域是带有读写保护的。

通过配置FEEPRO/FEEHIDE MMR（参见表24）的第31位，可以保护这62KB，使其不能通过JTAG编程模式来读取；该寄存器的另31位可以保护Flash/EE不被写入。寄存器的每1位保护4页（即2KB），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FEEHIDE MMR设定或取消保护。在复位后，保护不再维持；
- 通过写FEEPRO MMR可设定保护。只有在一个保存保护命令(0x0C)和复位以后才生效。FEEPRO MMR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。当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 MMR和一个特定密钥值：0xDEADDEAD。可以永久保护Flash/EE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。

要去除或修改保护，同样的时序用来改变FEEPRO数值。如果密钥选择的值是0xDEAD，存储器保护则无法去除。只有批量擦除无保护的部分，但它也擦除所有用户代码。

密钥写入时序如下所示（保护flash的第4页到第7页）：

```
FEEPRO=0xFFFFFFFF; //Protect pages 4 to 7
FEEMOD=0x48; //Write key enable
FEEADR=0x1234; //16 bit key value
FEEDAT=0x5678; //16 bit key value
FEECON= 0x0C; // Write key command
```

如果要永久保护这部分内容应该用同样的时序再写一次密钥，但要注意此时FEEADR=0xDEAD，FEEDAT=0xDEAD。

FLASH/EE控制接口

串行下载和JTAG编程用到Flash/EE控制接口，其中包括这节所列的8个MMR。

FEESTA 寄存器

名称	地址	默认值	访问
FEESTA	0xFFFFF800	0x20	读

FEESTA是一个只读寄存器，它反映Flash/EE控制接口的状态。如表21所示。

表21 FEESTA MMR位分配

位	描述
15:6	保留。
5	保留。
4	保留。
3	Flash中断状态位。当中断(命令已经完成并且FEEMOD寄存器中的Flash/EE中断使能位被置1)发生时，该位自动置1；读FEESTA寄存器后该位清0。
2	Flash/EE控制器忙标志位。当控制器忙时，该位自动置1；控制器空时，该位自动清0。
1	命令失效位。当命令没有完成时，该位自动置1；读FEESTA寄存器时该位自动清0。
0	命令执行位。当命令成功执行时，该位由微转换器置1；读FEESTA寄存器时该位自动清0。

FEEMOD 寄存器

名称	地址	默认值	访问
FEEMOD	0xFFFFF804	0x0000	读/写

FEEMOD设置Flash/EE接口的工作模式。

表22 显示FEEMOD MMR位分配。

表22 FEEMOD MMR位分配

位	描述
15:9	保留。
8	保留。该位应该始终清0
7:5	保留。除了写密钥时，这些位总是清0。参见写密钥时序一节。
4	Flash/EE中断使能位。置1，使能Flash/EE中断，一个命令完成时便会产生中断；清0，禁用Flash/EE中断。
3	擦/写命令保护位。置1，使能擦/写命令；该位清0禁用擦/写命令。
2:0	保留。这些位总是清0。

FEECON 寄存器

名称	地址	默认值	访问
FEECON	0xFFFFF808	0x07	读/写

FEECON是一个8位命令寄存器，命令代码如表23所示。

表23 FEECON的命令代码

代码	命令	描述
0x00 ¹	空	空闲状态。
0x01 ¹	单次读	向FEEDAT装载16位数据。由FEEADR索引。
0x02 ¹	单次写	将FEEDAT中的数据写入FEEADR所指的地址。此项操作需要50μs。
0x03 ¹	擦/写	擦除由FEEADR索引的存储页，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 ¹	单次校验	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 ¹	单次擦除	擦除由FEEADR索引的页面。
0x06 ¹	批量擦除	擦除62KB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。对于执行批量擦除操作请参阅命令序列部分。
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32778个时钟周期。
0x0C	保护	这个命令只能执行一次。只有批量擦除命令(0x06)可以将FEEPRO中的值保存或去除。
0x0D	保留	保留。
0x0E	保留	保留。
0x0F	Ping	无操作；中断产生。

¹ 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	地址	默认值	访问
FEEDAT	0xFFFFF80C	0XXXXX	读/写

FEEDAT是一个16位数据寄存器。

FEEADR 寄存器

名称	地址	默认值	访问
FEEADR	0xFFFFF810	0x0000	读/写

FEEADR是另一个16位地址寄存器。

FEESIGN 寄存器

名称	地址	默认值	访问
FEESIGN	0xFFFFF818	0FFFFFFF	读

FEESIGN为24位编码签名。

FEEPRO 寄存器

名称	地址	默认值	访问
FEEPRO	0xFFFFF81C	0x00000000	读/写

FEEPRO MMR在MMR复位后提供保护，它需要一个软件密钥（参见表24）。

FEEHIDE 寄存器

名称	地址	默认值	访问
FEEHIDE	0xFFFFF820	0FFFFFFFFF	读/写

FEEHIDE MMR提供立即保护。它不需要任何软件密钥。但要注意FEEHIDE中的保护设置通过一个复位来清0。（参见表24）。

表24 FEEPRO和FEEHIDE MMR位分配

位	描述
31	读保护位。清0，保护所有代码。置1，可以读取所有代码。
30:0	以4页为单位，如123页到120页、119到116、0到3页。清0，页写保护。置1，允许进行页写入。

批量擦除的命令序列

```
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22ns，而访问SRAM所需的时间为2ns，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22ns，所以在Flash/EE存储器中指令执行时间无法在1个时钟周期内完成（当CD=0时，在SRAM中执行指令就能在一个周期内完成）。并且无论CD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个时钟周期来取一个指令；在Thumb模式下，指令是16位的，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则更复杂，具体如表25所示。

表25. ARM/Thumb模式下指令的执行周期

指令	取周期	死区	数据访问	死区
LD ¹	2/1	1	2	1
LDH	2/1	1	1	1
LDM/PUSH	2/1	N ²	2 x N ²	N ¹
STR ¹	2/1	1	2 x 20 ns	1
STRH	2/1	1	20 ns	1
STRM/POP	2/1	N ¹	2xNx20 ns ¹	N ¹

¹ 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。

² 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51所示。

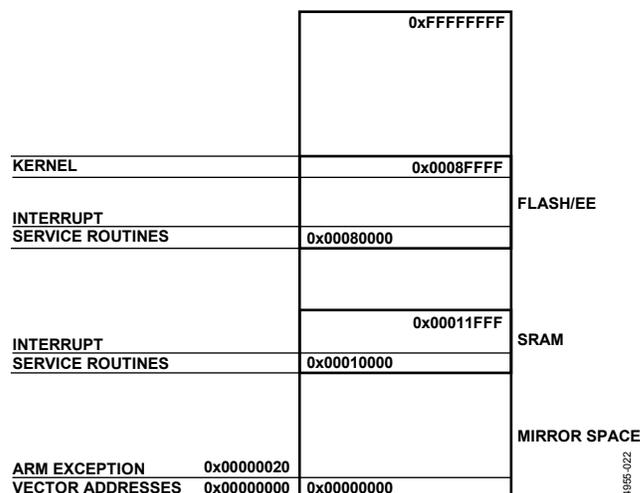


图51 异常重映射

默认情况下，复位之后，Flash/EE会被镜像到存储器阵列的底部。但使用时用户通常利用重映射功能将SRAM镜像到存储器的底部，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

当复位ADuC7019/20/21/22/24/25/26/27/28时，自动以出厂设定的形式执行内部代码。该内核是隐藏的，不能被用户代码访问。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被镜像到存储器底部，所以复位中断程序一定要写在Flash/EE中。

REMAP寄存器的第0位置1，Flash/EE将被重映射，但要注意，从Flash/EE中执行程序，是从0x00080020开始执行而不是从阵列底部开始执行，因为此时已被SRAM重映射了。

不过这个操作是可逆的，通过REMAP MMR的第0位置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映射到存储器阵列的底部。

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，清空寄存器RSTCLR可以将RSTSTA寄存器清空。在复位异常服务程序执行时，这些寄存器可以被访问。如果RSTSTA为空，则为外部复位。

REMAP 寄存器

名称	地址	默认值	访问
REMAP	0xFFFF0220	0xXX ¹	读/写

¹ 取决于具体的器件型号。

表26 REMAP MMR位分配

位	名称	描述
4	重映射	只读位。标明Flash/EE存储器可用空间的大小。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。该位置1，表示SRAM中只有4KB空间可用。
2:1		保留。
0		重映射位。该位置1，映射SRAM到地址0x00000000；重映射Flash/EE到地址0x00000000后该位自动清0。

RSTSTA 寄存器

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

表27 RSTSTA MMR位分配

位	描述
7:3	保留。
2	软件复位。该位置1，强行使用软件复位；将RSTCLR寄存器相对应的位置1后该位清0。
1	看门狗超时状态位。看门狗定时器超时该位自动置1；将RSTCLR寄存器相对应的位置1后该位清0。
0	上电复位。上电复位时该位自动置1。将RSTCLR寄存器相对应的位置1后该位清0。

RSTCLR 寄存器

名称	地址	默认值	访问
RSTCLR	0xFFFF0234	0x00	读/写

注意：清空RSTSTA寄存器时用户必须将0x07写入RSTCLR寄存器。

其他模拟外设

DAC

ADuC7019/20/21/22/24/25/26/27/28片内集成有2个、3个或4个12位DAC，取决于不同的型号。每一个DAC都有轨到轨电压输出缓冲器，驱动能力为5KΩ/100 pF。

每个DAC都有三个可选电压输出范围：0V至 V_{REF} （内部带隙2.5V基准电压源）、0V至 DAC_{REF} 、0V至 AV_{DD} 。 DAC_{REF} 为DAC的外部基准源，信号范围为0至 AV_{DD} 。

MMR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28)和DAC0DAT(参见表29)的具体功能。

DACxCON 寄存器

名称	地址	默认值	访问
DAC0CON	0xFFFF0600	0x00	读/写
DAC1CON	0xFFFF0608	0x00	读/写
DAC2CON	0xFFFF0610	0x00	读/写
DAC3CON	0xFFFF0618	0x00	读/写

表28 DAC0CON MMR位分配

位	值	名称	描述
7:6			保留。
5		DACCLK	DAC更新速率。置1，通过定时器1更新DAC。清0，使用HCLK（内核时钟）更新DAC。
4		DACCLR	DAC清0位。置1，DAC正常操作使能。清0，复位DAC数据寄存器为0。
3			保留。该位应保留为0。
2			保留。该位应保留为0。
1:0			DAC输出范围选择。
	00		掉电模式。DAC输出为三态。
	01		0至 DAC_{REF}
	10		0至 V_{REF} (2.5V)
	11		0至 AV_{DD}

DACxDAT 寄存器

名称	地址	默认值	访问
DAC0DAT	0xFFFF0604	0x00000000	读/写
DAC1DAT	0xFFFF060C	0x00000000	读/写
DAC2DAT	0xFFFF0614	0x00000000	读/写
DAC3DAT	0xFFFF061C	0x00000000	读/写

表29 DAC0DAT MMR位分配

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

DAC的使用

片内DAC由一电阻网路DAC和一个输出缓冲放大器构成，功能等效框图如图52所示。

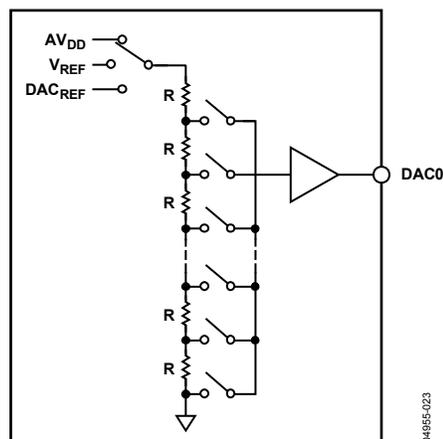


图52 DAC的结构

如图52所示，用户可在软件中选择各DAC的基准电压源。可以是 AV_{DD} 、 V_{REF} 或者 DAC_{REF} 。0至 AV_{DD} 模式时，DAC输出传递函数范围为0至 AV_{DD} 引脚电压；0至 DAC_{REF} 模式时，DAC输出传递函数范围为0至 DAC_{REF} 引脚电压；0至 V_{REF} 模式时，DAC输出传递函数范围为0至2.5V内部参考源 V_{REF} 。

每一个DAC都具有一个真正的轨到轨输出级缓冲放大器，也就是说，在输出不带负载时，DAC输出摆幅能够达到 AV_{DD} 或地电平的5mV范围以内。另外，在接有5kΩ电阻负载时，除了输出代码0至100（以及0至 AV_{DD} 模式时输出代码3995至4095）外，整个传递函数都达到线性度规范指标。

一般情况下，在DAC输出曲线的两个端点“地”和“ V_{DD} ”处会出现线性度劣化，这是由输出放大器的饱和引起的，这种效应的一般表现（忽略了偏移误差和增益误差）如图53所示。图中的虚线为理想转换函数，实线则表示可能具有端点非线性（由输出放大器饱和引起）的转换函数。注意图53仅为在输入范围在0至 AV_{DD} 模式时的传递函数。在0至 V_{REF} 模式或0至 DAC_{REF} 模式，低端具有相似的非线性，这里一般存在 $V_{REF} < AV_{DD}$ 或 $DAC_{REF} < AV_{DD}$ 。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（这里为 V_{REF} ，而不是 AV_{DD} ），没有端点线性误差。

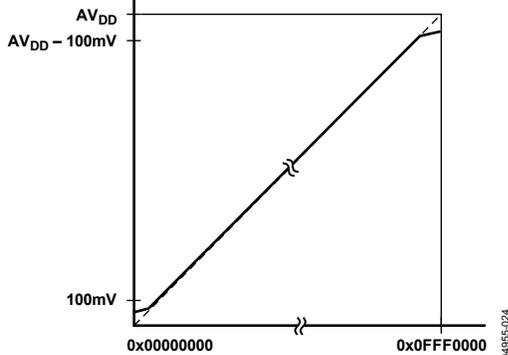


图5.3 放大器饱和引起的端点非线性

当有输出负载时，图53中的端点非线性会变得更差。ADuC7019/20/21/22/24/25/26/27/28数据手册中的绝大多数DAC输出特性指标都是在DAC输出接有一个到地的 $5K\Omega$ 负载条件下得到的。由于DAC输出被强制提供更多的驱动或灌的电流，图53中顶部或底部非线性区域将增大。而当需要大电流时，这会更明显地限制输出电压摆幅。

电源监控器

ADuC7019/20/21/22/24/25/26/27/28电源监控器主要监控片上 IOV_{DD} 电压，当 IOV_{DD} 引脚电压下降到低于两个电源触发点之一时会给出提示。监控功能是通过PSMCON寄存器来控制的。如果使能寄存器IRQEN或FIQEN，监控器使用PSMCON MMR的PSMI位向CPU发中断请求，而一旦CMP位恢复到高电平，PSMI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	地址	默认值	访问
PSMCON	0xFFFF0440	0x0008	读/写

表30. PSMCON MMR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_{DD} 电压高于已选的触发点或PSM处于掉电模式；该位为0表示 IOV_{DD} 电压低于已选触发点。在退出中断服务程序之前，该位应置1。
2	TP	触发点选择位。置1，3.07V；清0，2.79V
1	PSMEN	电源监控器使能位。该位置1，使能电源监控器；该位清0，禁用电源监控器。
0	PSMI	电源监控中断位。一旦CMP变低，该位会被CPU置1表明 IOV_{DD} 偏低，PSMI位可用来中断处理器。一旦CMP变高，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变高，PSMI位可以立即被清0。

比较器

ADuC7019/20/21/22/24/25/26/27/28集成了电压比较器。比较器的正相输入端与ADC2引脚复用，负相输入端有两个：ADC3或DAC0。通过配置，电压比较器的输出可以产生系统中断、可以作为可编程逻辑阵列PLA的输入、可以启动ADC转换或输出到外部引脚 CMP_{OUT} 上，如图54所示：

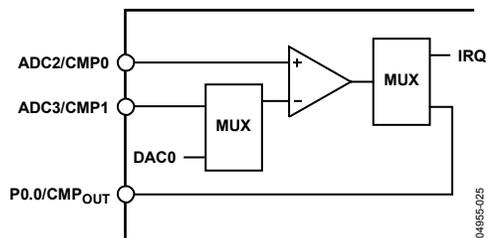


图54 比较器

注意，因为ADuC7022、ADuC7025、ADu7027不支持DAC0输出，所以把DAC0作为比较器输入这些型号的器件是不可能的。

迟滞现象

图55描述了输入偏移电压和迟滞基本原理，其中输入偏移电压(V_{OS})为迟滞范围中心对地的差，它可以是正，也可以是负；迟滞电压(V_H)为迟滞范围1/2的宽度。

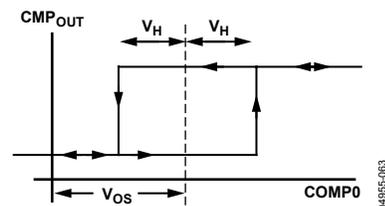


图55 比较器迟滞转换功能

比较器接口

比较器接口由一个16位寄存器CMPCON组成，如表31所示。

CMPCON 寄存器

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

表31 CMPCON MMR位功能描述

位	值	名称	描述
15:11			保留。
10		CMPEN	比较器使能位。置1，使能比较器；清0，禁用比较器
9:8		CMPIN	比较器反相输入选择位。
	00		AV _{DD} / 2
	01		ADC3输入
	10		DAC0输出
	11		保留。
7:6		CMPOC	比较器输出配置位。
	00		保留。
	01		保留。
	10		在CMP _{Out} 上输出。
	11		IRQ。
5		CMPOL	比较器输出逻辑状态位。该位清0，如果正相输入(CMP0)高于反相输入(CMP1)，比较器输出为高；该位置1，如果正相输入(CMP0)低于负相输入(CMP1)，比较器输出为高。
4:3		CMPRES	响应时间。
	00		对于大信号(2.5V差分)，响应时间典型值5μs。对于小信号(0.65V差分)，响应时间典型值17μs
	11		3μs典型值。
	01/10		保留。
2		CMPHYST	比较器迟滞位。置1，有迟滞，大约为7.5mV；清0，无迟滞
1		CMPORI	比较器输出上升沿中断。当CMP0出现上升沿时该位自动置1；对该位写1可清0。
0		CMPOFI	比较器输出下降沿中断。出现下降沿时该位自动置1；由用户清0。

振荡器和锁相环—电源控制

时钟系统

ADuC7019/20/21/22/24/25/26/27/28内部集成有一个32.768 KHz±3%的振荡器，一个时钟分频器和一个锁相环(PLL)。PLL可以锁住内部振荡器或外部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下或其二的倍数分频上，实际的内核工作频率UCLK/2^{CD}为HCLK。默认的内核时钟分频为8分频(CD=3)，或5.22 MHz，内核时钟频率也可以来自ECLK引脚的外部时钟，如图56所示。当使用内部振荡器或外部晶振时，内核时钟也可以输出到ECLK引脚上。

注意：当内核时钟在ECLK引脚输出时，输出信号不带缓冲，不适合作为无外缓冲外部器件的时钟源。

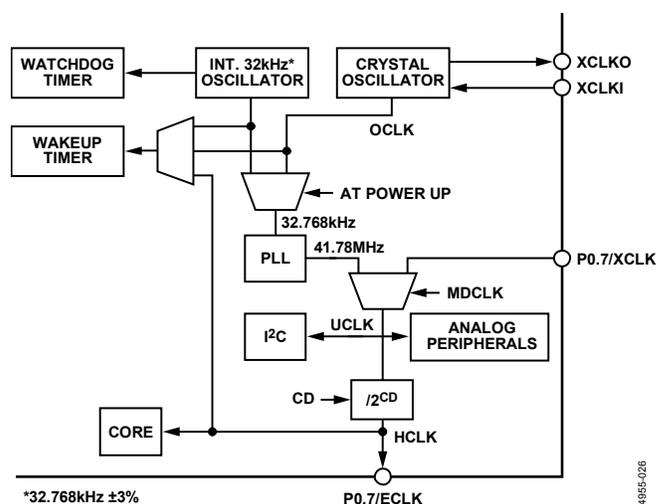


图56 时钟系统

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。

外部晶振选择

在切换使用外部晶振时，用户必须遵循如下过程：

1. 使能Timer2中断，并配置其超时周期>120μs。
2. 按照PLLCON寄存器写时序，置MDCLK位为01，清OSEL位；
3. 按照POWCON寄存器正确写时序，强制器件进入NAP模式；
4. 在NAP模式下出现Timer2中断时，时钟源已经切换到外部晶振。

示例源代码:

```
T2LD = 5;
TCON = 0x480;

while ((T2VAL == t2val_old) || (T2VAL >3)) //3))
//ensures timer value loaded
IRQEN = 0x10;
//enable T2 interrupt
PLLKEY1 = 0xAA;
PLLCON = 0x01;
PLLKEY2 = 0x55;

POWKEY1 = 0x01;
POWCON = 0x27;
// Set Core into Nap mode
POWKEY2 = 0xF4;
```

但如果此时环境中存在噪声,噪声便会从外部晶振引脚耦合进入,使PLL瞬时失锁。中断控制器就会出现PLL中断。内核时钟立即挂起,只有在锁相恢复后该中断才能被处理。为了避免晶振失振,应该使用看门狗定时器。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。

外部时钟选择

如果使用外部时钟,必须配置P0.7为模式1。一般外部时钟最高可达44MHz,容差为1%。

示例源代码:

```
T2LD = 5;
TCON = 0x480;

while ((T2VAL == t2val_old) || (T2VAL >3))
//ensures timer value loaded

IRQEN = 0x10;
//enable T2 interrupt

PLLKEY1 = 0xAA;
PLLCON = 0x03; //Select external clock
PLLKEY2 = 0x55;

POWKEY1 = 0x01;
POWCON = 0x27;
// Set Core into Nap mode
POWKEY2 = 0xF4;
```

电源控制系统

ADuC7019/20/21/22/24/25/26/27/28有多个运行模式可供选择,表32给出了不同工作模式下器件各部分的上电情况和上电时间。

表33给出了不同模式下,器件总耗电(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。注意:这些功耗值也包括测量时稳压器和测试板上的其他一些器件消耗的电流。

表32. 工作模式

模式	内核	外设	PLL	XTAL/T2/T3	IRQ0至IRQ3	启动/上电时间
活动(Active)	X	X	X	X	X	130 ms(CD = 0)
暂停(Pause)		X	X	X	X	24 ns (CD = 0); 3 μs(CD = 7)
浅休眠(Nap)			X	X	X	24 ns(CD = 0); 3μs(CD = 7)
睡眠(Sleep)				X	X	1.58 ms
停止(Stop)					X	1.7 ms

表33. 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	活动(Active)	33.1	21.2	13.8	10	8.1	7.2	6.7	6.45
001	暂停(Pause)	22.7	13.3	8.5	6.1	4.9	4.3	4	3.85
010	浅休眠(NAP)	3.8	3.8	3.8	3.8	3.8	3.8	3.8	3.8
011	睡眠(Sleep)	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
100	停止(Stop)	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4

MMR和关键字

工作模式、时钟模式和可编程时钟分频器可通过PLLCON(参见表34)和POWCON(参见表35)两个MMR控制。PLLCON控制时钟系统的工作模式，而POWCON控制内核时钟频率和掉电模式。

为了防止意外编程，写入PLLCON和POWCON寄存器时需要遵循特定的时序（参见表36）。

PLLKEYx寄存器

名称	地址	默认值	访问
PLLKEY1	0xFFFF0410	0x0000	写
PLLKEY2	0xFFFF0418	0x0000	写

PLLCON寄存器

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

表34. PLLCON MMR位分配

位	值	名称	描述
7:6			保留。
5		OSEL	32KHZ PLL 输入选择。置1,使用内部32KHz振荡器。默认值设置。清0,使用外部32KHz晶振。
4:2			保留。
1:0		MDCLK	时钟模式。
	00		保留。
	01		PLL.默认配置。
	10		保留。
	11		在P0.7引脚上的外部时钟。

POWKEYx寄存器

名称	地址	默认值	访问
POWKEY1	0xFFFF0404	0x0000	写
POWKEY2	0xFFFF040C	0x0000	写

POWCON寄存器

名称	地址	默认值	访问
POWCON	0xFFFF0408	0x0003	读/写

表35. POWCON MMR位分配

位	值	名称	描述
7			保留。
6:4		PC	工作模式。 Active模式。 Pause模式。
	000		
	001		
	010		NAP。
	011		Sleep模式。IRQ0至IRQ3和定时器2可以唤醒该器件。
	100		Stop模式IRQ0至IRQ3可以唤醒该器件。
	其他		保留。
3			保留。
2:0		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36. PLLCON和POWCON写时序

PLLCON	POWCON
PLLKEY1 = 0xAA	POWKEY1 = 0x01
PLLCON = 0x01	POWCON = 用户值
PLLKEY2 = 0x55	POWKEY2 = 0xF4

数字外设

三相脉宽调制 (PWM)

ADuC7019/20/21/22/24/25/26/27/28均内置有一个灵活的、可编程的三相脉宽调制(PWM)波形发生器。它可通过编程产生所需的开关信号来驱动三相电压源逆变器用于交流感应马达控制(ACIM)。

需要注意的是，只能产生高电平有效的波形。

PWM发生器可以在6个脉宽调制输出引脚(PWM0_H、PWM0_L、PWM1_H、PWM1_L、PWM2_H和PWM2_L)上产生三对脉宽调制信号。这六个PWM输出信号包括三个高侧驱动信号和三个低侧驱动信号。

PWM波形的开关频率和死区可以用PWMDAT0和PWMDAT1 MMR编程。除此之外，三个占空比控制寄存器(PWMCH0、PWMCH1、PWMCH2)直接控制三对PWM信号的占空比。

这六个PWM输出信号中的每一个都由PWMMEN寄存器的不同输出使能位控制使能或禁用。除此以外，PWMMEN寄存器有三个控制位允许一个PWM对的两个信号交越。在交越的模式下，设定为高侧输出的PWM信号转换为对应的互补低侧输出信号。设定为低侧输出的PWM信号转换为对应的高侧输出信号。

在很多应用中，需要为开启逆变器功率器件的栅驱动电路提供绝缘隔离。一般情况下，有两种绝缘隔离技术：采用光耦合器的光隔离和采用脉冲变压器的变压器隔离。PWM控制器允许在输出PWM信号时混合一个高频斩波信号，这样更容易与脉冲变压器接口。这种门驱动斩波模式可由PWMCFG寄存器控制。PWMCFG寄存器中有8位直接控制斩波频率。高侧输出和低侧输出也由PWMCFG寄存器的不同使能位分别控制高频的斩波率。

PWM可以工作在两种不同的模式下：单次更新模式和二次更新模式。在单次更新模式下，每一个PWM工作周期占空比的值只可以改变一次，因此产生一个关于PWM周期中点对称的PWM波形。在二次更新模式下，可以在一个PWM周期的中点再次更新占空比。

在二次更新模式下，还可以产生一个不对称的PWM波形，这个可以在三相PWM逆变器上能够产生更低次谐波失真。这种技术允许闭环控制器以更快的速率改变施加于电机绕组上的平均电压。因此，能够产生更快的闭环带宽。PWM模块的工作模式由PWMCON寄存器的一个控制位选择。在单次更新模式下，每个PWM周期都会产生一个PWMSYNC脉冲信号。在

二次更新模式下，在每个PWM周期的中点都会产生一个附加的PWMSYNC脉冲。

PWM模块也能在PWM_{SYNC}引脚上提供一个内部同步脉冲，它与PWM开关频率同步。在单次更新模式下，每个PWM周期开始都会产生一个脉冲信号。在二次更新模式下，在每个PWM周期的中点都会产生一个附加的脉冲。脉冲宽度可通过PWMDAT2寄存器编程。PWM模块也能在PWM_{SYNC}引脚上提供一个外部同步脉冲，内外同步的选择在PWMCON寄存器上。SYNC输入时间能够与内部的外设时钟同步，这可通过PWMCON选择。如果该芯片引脚的外部同步脉冲和内部的外设时钟是异步的（典型情况），则外部的PWMSYNC被认为异步，应同步。从外部脉冲到实际的PWM输出同步逻辑增加延迟和抖动。PWM_{SYNC}引脚产生的脉冲大小必须大于两个内核时钟的周期。

ADuC7019/20/21/22/24/25/26/27/28产生的PWM信号可以通过专用的异步PWM关断引脚PWM_{TRIP}停止。当接收到低电平时，PWM_{TRIP}会瞬间使六个PWM输出引脚处于关闭状态（高电平）。由于这个硬件关闭机制是异步的，相关的PWM关闭电路不会通过任何时钟逻辑，这可以保证即使在内核时钟丢失的情况下PWM也能正确关闭。

用户可通过读PWMSTA寄存器获得PWM系统的状态信息，特别地，也可提供PWM_{TRIP}引脚的状态，同时也有状态位反映PWM是运行在前半周期还是后半周期。

40 引脚封装器件

在40引脚封装器件中，PWM的输出不能被直接访问，就如在通用输入/输出一节中所描述的，如本例所示，可通过PLA在GPIO引出一个通道。

```
PWMCON = 0x1;           // enables PWM o/p
PWMDAT0 = 0x055F;      // PWM switching freq
// Configure Port Pins
GP4CON = 0x300;        // P4.2 as PLA output
GP3CON = 0x1;          // P3.0 configured as
                        // output of PWM0
                        //(internally)

// PWM0 onto P4.2
PLAELM8 = 0x0035;      // P3.0 (PWM output)
                        // input of element 8
PLAELM10 = 0x0059;     // PWM from element 8
```

PWM模块说明

PWM控制器功能框图如图57所示。从引脚PWM0_H到引脚PWM2_L上的六个PWM输出信号由以下四个重要模块控制：

- 三相PWM时钟单元。这是PWM控制器的核心部分，它可以产生三对互补的、死区可调的、中心基准的PWM信号。并且产生内部同步脉冲PWMSYNC，并且可以控制是否使用外部PWMSYNC引脚。
- 输出控制单元。该单元可以调整每一通道的三相时钟单元为高侧输出或为低侧输出，另外，输出控制单元可以单独控制六个PWM输出信号使能或禁用。
- 门驱动单元。该单元可以产生高频斩波以及与PWM信号混合在一起的低频波。
- PWM关闭控制器。该单元可以通过PWMT_{TRIP}引脚控制PWM的关闭并且为时钟单元提供准确的复位信号。

PWM控制器由ADuC7019/20/21/22/24/25/26/27/28的内核时钟频率驱动，可为ARM核提供两个中断。一个中断在PWMSYNC脉冲出现时产生，另一个在任何一个PWM关闭动作出现时产生。

三相时钟单元

PWM开关频率(PWMDAT0 MMR)

PWM开关频率由PWM周期寄存器PWMDAT0控制。PWM控制器的基本时钟单元为：

$$t_{CORE} = 1/f_{CORE}$$

其中 f_{CORE} 为微转换器内核频率。

因此，当 f_{CORE} 为41.78 MHz时，基本时间增量是24 ns。在半个PWM周期以内，写进PWMDAT0寄存器的数据是 f_{CORE} 时间增量的个数。PWMDAT0的值与要求PWM开关频率(f_{PWM})的函数关系为：

$$PWMDAT0 = f_{CORE} / (2 \times f_{PWM})$$

因此，PWM开关周期 t_s 可以写成：

$$t_s = 2 \times PWMDAT0 \times t_{CORE}$$

PWMDAT0 MMR可写入的最大值是0xFFFF = 65535，对应最小的PWM开关频率：

$$f_{PWM(min)} = 41.78 \times 10^6 / (2 \times 65535) = 318.75 \text{ Hz}$$

注意：PWMDAT0中的值不能为0和1。

PWM开关死区(PWMDAT1 MMR)

在PWM初始配置时第二个必须设定的重要参数是开关死区，它是在关断一个PWM信号（如0H）与打开其互补信号（0L）之间引入的一个短时延迟。这个短时延迟引入是为了在电源开关关闭（这里为0H）之后到打开互补信号之前保持电路性能。这一时间延迟可以阻止电源转换器直流链路电容引起的潜在破坏性短路。

死区由10位读/写寄存器PWMDAT1控制。只有这一个死区寄存器控制三对PWM输出信号的死区。死区 t_D 与PWMDAT1寄存器数值之间的关系为：

$$t_D = PWMDAT1 \times 2 \times t_{CORE}$$

因此，如果PWMDAT1的值为0x00A (= 10)，则会在关断一个PWM信号（例如，0H）到打开其互补信号(0L)之间插入一个426 ns的延迟。因此，死区可以2 t_{CORE} 增量进行编程(或49 ns，在内核时钟41.78 MHz时)。

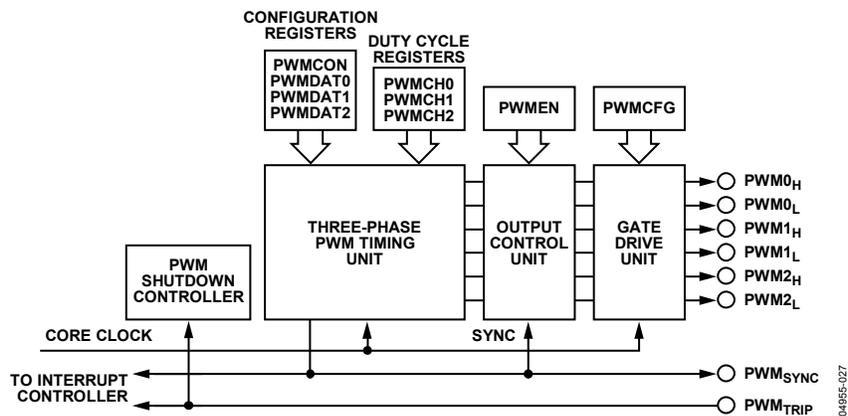


表57. PWM控制器简图

PWMDAT1寄存器是一个10位寄存器，可存储的最大值为0x3FF (= 1023)，对应的最大可编程死区为：

$$t_{D(max)} = 1023 \times 2 \times t_{CORE} = 1023 \times 2 \times 24 \times 10^{-9} = 48.97 \mu s$$

其中内核始终为41.78 MHz

显然，可通过对PWMDAT1寄存器写入0来使死区为0。

PWM工作模式(PWMCON、PWMSTA MMR)

如前所述，ADuC7019/20/21/22/24/25/26/27/28的PWM控制器可以工作在两种不同的模式下：单次更新模式和二次更新模式。PWM控制器工作在何种模式由PWMCON寄存器的第2位状态决定，如果该位为0，PWM工作在单次更新模式下；第2位置1则工作在二次更新模式，默认为单次更新模式。

在单次更新模式下，每个PWM周期都会产生一个PWMSYNC脉冲信号，该信号上升沿表示一个新的PWM周期开始，并且将PWM配置寄存器（PWMDAT0、PWMDAT1）和PWM占空比寄存器(PWMCH0、PWMCH1和PWMCH2)的新数值锁存到三相时钟单元。除此以外，在PWMSYNC脉冲的上升沿，PWMDAT1寄存器也被锁存到输出控制单元，这也就意味着在每一个PWM周期开始时PWM信号的特性和占空比值只可以被改变一次。其结果是左右对称、有关开关周期中点的PWM波形。

在二次更新模式下，在每个PWM周期的中点都会产生一个附加的PWMSYNC脉冲。这个新PWMSYNC脉冲的上升沿被用来锁存PWM配置寄存器、占空比寄存器和PWMDAT1寄存器新的数值。这样可以在每一个PWM周期中点时改变特性参数（开关频率、死区）和输出占空比。因此，还可以产生一个不再关于PWM周期中点对称的PWM开关波形（非对称PWM）。在二次更新模式下，有必要知道当前是运行在PWM的前半周期还是后半周期，这由PWMSTA寄存器的第0位反映，在每个PWM前半周期被清0（初始PWMSYNC脉冲上升沿和二次更新模式产生的新PWMSYNC脉冲上升沿之间），PWMSTA寄存器第0位在每个PWM后半周期被置1。需要时，这一状态位可用于在PWMSYNC中断服务程序中确定一个特定半周期。

二次更新模式的优点是可产生更低次谐波的电压和更快的控制带宽。尽管如此，在二次更新模式下对于一个给定的PWM

开关频率，PWMSYNC脉冲将以两倍的速度出现。因为在每一个PWMSYNC中断服务程序中都必须重新计算新的占空比值，对于ARM内核来说这是一个比较大的计算量。

PWM占空比(PWMCH0、PWMCH1、PWMCH2 MMRs)

引脚PWM0_H到PWM2_L上的六个PWM输出信号的占空比值由三个16位读写占空比寄存器PWMCH0、PWMCH1和PWMCH2控制。占空比寄存器编程为基本时钟单元 t_{CORE} 的整数倍，它决定着三相时钟单元产生的期望PWM输出高侧信号超过PWM半个周期的时间。三相时钟单元产生的开关信号也可进行调整，以配合PWMDAT1寄存器中的死区。三相时钟单元根据命令产生动态高电平有效信号来开启相关的功率器件。

图58为在单次更新模式下时钟单元产生的一对典型PWM输出（这里为0H和0L）图示的所有时间值均为相关寄存器中的整数倍，可通过乘以基本时间增量 t_{CORE} 转换为相应的时间。注意在这种模式下开关波形关于每个开关周期的中点对称，这是因为在前半周期和后半周期所用的PWMCH0、PWMDAT0和PWMDAT1值是相同的。

图58图同时也表明了如何调整已编程占空比，以便在一对PWM信号中插入要求的死区。死区的插入是通过将两个PWM信号(0H和0L)开关瞬间从PWMCH0寄存器建立瞬间移除。

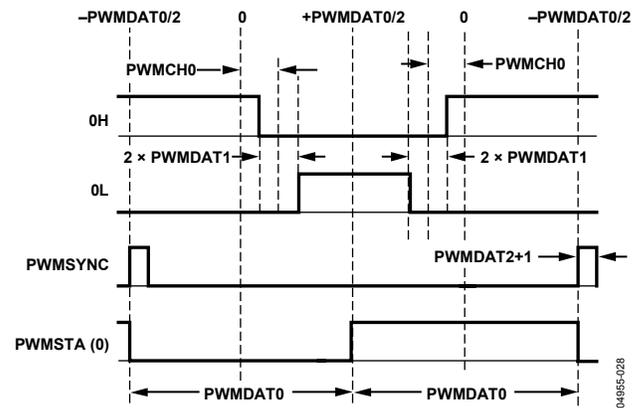


图5.8 三相时钟单元的典型PWM输出（单次更新模式）

为保证输出对称，两个开关沿移动相等的量($PWMDAT1 \times t_{CORE}$)。

图中同时还给出了PWMSYNC脉冲和PWMSTA寄存器的第0位，该位反映当前工作在前半PWM周期还是后半PWM周期。

由时钟单元产生的整个PWM周期（两个半周期）的PWM信号最终开启时间可由下式得到：

在高侧：

$$t_{OH} = PWMDAT0 + 2(PWMCH0 - PWMDAT1) \times t_{CORE}$$

$$t_{OL} = PWMDAT0 - 2(PWMCH0 - PWMDAT1) \times t_{CORE}$$

对应的占空比(d)为：

$$d_{OH} = t_{OH}/t_S = 1/2 + (PWMCH0 - PWMDAT1)/PWMDAT0$$

在低侧：

$$t_{OLH} = PWMDAT0 - 2(PWMCH0 + PWMDAT1) \times t_{CORE}$$

$$t_{OLL} = PWMDAT0 + 2(PWMCH0 + PWMDAT1) \times t_{CORE}$$

对应的占空比(d)为：

$$d_{OL} = t_{OLH}/t_S = 1/2 - (PWMCH0 + PWMDAT1)/PWMDAT0$$

可允许的 t_{OH} 和 t_{OL} 最小值为0，此时的占空比为0%。与此相似，最大值是 t_S ，对应的占空比为100%。

图59为二次更新模式下时钟单元的输出信号，显示了一个通用事例，即开关频率、死区和占空比都在PWM后半周期改变。它们其中任何一个或全部三个值既可用于PWM前半周期也可用于PWM后半周期。但这并不能保证在二次更新模式下时钟单元产生对称的PWM信号。图59 PWM信号死区的插入与单次更新模式下相同。

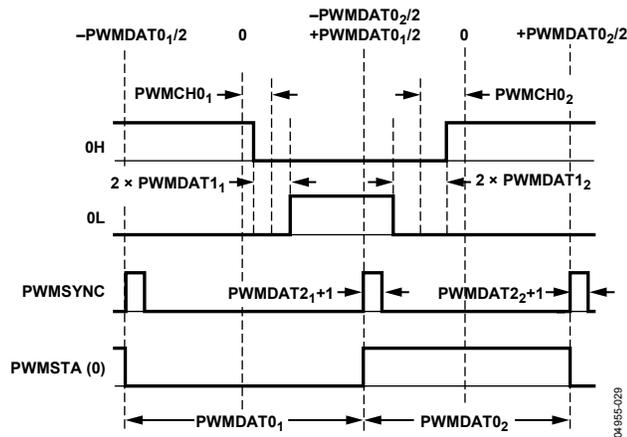


图59 三相时钟单元的典型PWM输出（二次更新模式）

总的来说，二次更新模式下PWM信号的开启时间由下式表述：

在高侧：

$$t_{OH} = (PWMDAT0_1/2 + PWMDAT0_2/2 + PWMCH0_1 + PWMCH0_2 - PWMDAT1_1 - PWMDAT1_2) \times t_{CORE}$$

$$t_{OL} = (PWMDAT0_1/2 + PWMDAT0_2/2 - PWMCH0_1 - PWMCH0_2 + PWMDAT1_1 + PWMDAT1_2) \times t_{CORE}$$

这里下角标1表示为前半周期寄存器中的值，下角标2表示为后半周期寄存器中的值。

对应的占空比(d)为：

$$d_{OH} = t_{OH}/t_S = (PWMDAT0_1/2 + PWMDAT0_2/2 + PWMCH0_1 + PWMCH0_2 - PWMDAT1_1 - PWMDAT1_2)/(PWMDAT0_1 + PWMDAT0_2)$$

在低侧：

$$t_{OLH} = (PWMDAT0_1/2 + PWMDAT0_2/2 + PWMCH0_1 + PWMCH0_2 + PWMDAT1_1 + PWMDAT1_2) \times t_{CORE}$$

$$t_{OLL} = (PWMDAT0_1/2 + PWMDAT0_2/2 - PWMCH0_1 - PWMCH0_2 - PWMDAT1_1 - PWMDAT1_2) \times t_{CORE}$$

这里下角标1表示为前半周期寄存器中的值，下角标2表示为后半周期寄存器中的值。

对应的占空比(d)为：

$$d_{OL} = t_{OLH}/t_S = (PWMDAT0_1/2 + PWMDAT0_2/2 + PWMCH0_1 + PWMCH0_2 + PWMDAT1_1 + PWMDAT1_2)/(PWMDAT0_1 + PWMDAT0_2)$$

对于二次更新模式的一个完整常见案例（见图59），开关周期表示为：

$$t_S = (PWMDAT0_1 + PWMDAT0_2) \times t_{CORE}$$

同样， t_{OH} 和 t_{OL} 值被限制在0到 t_S 之间。

通过采用与PWMCH0中介绍的一样的方法对寄存器PWMCH1和PWMCH2进行编程，可以在1H、1L、2H和2L输出上产生与图58和59类似的PWM信号。仅当所有的寄存器PWMDAT0、PWMCH0、PWMCH1和PWMCH2都被至少写入一次，PWM控制器才会产生PWM输出信号。一旦这些寄存器被写入，三相时钟单元的内部计时器即被使能。

写入PWMDAT0寄存器的同时开启PWM主计时器内部计时。如果在初始化时写入PWMCH0、PWMCH1和PWMCH2寄存器之前就先写入PWMDAT0寄存器，在单次更新模式下第一个PWMSYNC脉冲和中断（如果使能）会出现在最初写PWMDAT0寄存器后的 $1.5 \times t_{CORE} \times PWMDAT0$ 秒。在二次更新模式下，第一个PWMSYNC脉冲会在 $PWMDAT0 \times t_{CORE}$ 秒后出现。

输出控制单元

输出控制单元是由一个9位读写寄存器PWMMEN控制的，该寄存器控制着输出控制单元的两个主要特性，这些特性可以直接应用于电子计数测量(ECM)控制或二进制/十进制计数测量(BDCM)控制。PWMMEN寄存器包括有三个交越位，每位对应一对PWM输出。寄存器第8位置1使能0H/0L这对PWM信号交越，第7位置1使能1H/1L这对PWM信号交越，第6位置1使能2H/2L这对PWM信号交越，对于任何一对PWM信号一旦使能交越模式，时钟单元的高侧PWM信号(例如，0H)就会转换到相应的低侧输出，这样信号最终出现在PWM0L引脚。当然，对应的时钟单元的低侧输出也会转换到对应的互补高侧输出，这样信号最终会出现在PWM0H引脚。复位后，三个交越使能位清0，三对PWM信号全部禁用交越模式。PWMMEN寄存器有6个控制位(Bit 0至Bit 5)可用于单独控制六个PWM输出使能或禁用。如果PWMMEN寄存器相应的位被置1，那么不管相关的占空比寄存器的值如何，对应的PWM输出就会禁用。PWM输出信号会一直保持关闭状态直到对应的PWMMEN寄存器的使能/禁用位被置1。实施这一输出启用功能后实施交越功能。

复位后，PWMMEN寄存器的这六个使能位会被清空，并且所有PWM输出会恢复为默认设置。采用与占空比寄存器一样的方式，PWMMEN在PWMSYNC信号在上升沿被锁定。因此，在单次更新模式下这个寄存器中值的改变只在每一个PWM周期的开始时有效。在二次更新模式下，PWMMEN寄存器的值还可在PWM周期的中点更新。

对于控制一个ECM,在任何时候只有两个转换器引脚可供使用，并且常常是一个引脚连接高侧电平装置的同时另一个引脚连接低侧电平驱动装置。所以，通过对PWM的两个通道设置相同的占空比值(例如PWMCH0 = PWMCH1)以及对PWMMEN寄存器的第7位置1来使1H/1L这对PWM信号进行交越，这就有可能在开启A相高侧电平开关的同时开启B相低侧电平开关。在控制ECM时，对于某一些PWM周期，换向器的第三个引脚(这个例子为C相)常常禁用。通过对PWMMEN寄存器的第0位和第1位置1禁用2H和2L PWM输出可实现该功能。

情况如图60所示，在图中可以看到0H和1L信号完全相同，这是因为PWMCH0 = PWMCH1并且B相交越位被置1。

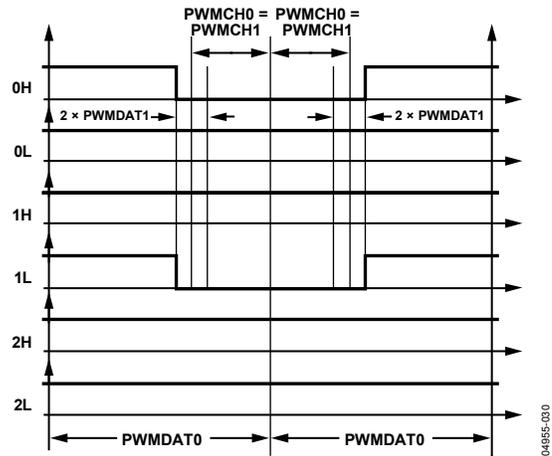


图60 单次更新模式下用于控制ECM的低电平有效PWM信号 (PWMCH0=PWMCH1, 1H和1L对交越, 0L、1H、2H和2L输出禁用)。

另外，对PWMMEN寄存器的合适的使能/禁用引脚置1可使其它四个信号(0L、1H、2H、和2L)停止工作。在图60中，对PWMMEN寄存器合适的值为0x00A7。在ECM正常操作时，需要根据转轴的位置(电机换向)来改变PWMMEN寄存器中的值，在这个过程中的一定周期内，换向器的各引脚被禁用。

门驱动单元

PWM控制器的门驱动单元增加的一些特性简化了PWM换向器的相关门驱动电路的设计。如果使用一个变压器耦合，电源装置或一个门驱动放大器，那么有效PWM信号必须在一个高频下截断。16位读写寄存器PWMCFG编程控制这个高频斩波模式。这个斩波有效PWM信号可仅用于高侧驱动，仅用于低侧驱动，或者高侧和低侧开关所以在PWMCFG寄存器上有两个不同的控制位单独控制高侧和低侧开关。

图61展示了在高侧和低侧信号下，高频斩波典型的PWM输出信号。通过对PWMCFG寄存器的第8位置1可使高侧PWM输出(0H、1H和2H)斩波。通过对PWMCFG寄存器的第9位置1可使低侧PWM输出(0L、1L和2L)斩波。这个高频斩波频率是通过向PWMCFG寄存器的第0位到第7位写入一个8位字(GDCLK)来控制的。这个高频载波的周期是：

$$t_{CHOP} = (4 \times (GDCLK + 1)) \times t_{CORE}$$

所以这个斩波频率是微处理器内核频率的一个整数分量。

$$f_{CHOP} = f_{CORE} / (4 \times (GDCLK + 1))$$

对于一个CPU频率为41.78 MHz的PWM，GDCLK的取值范围为0到255，对应的可编程斩波频率的范围为40.8 kHz到10.44 MHz。门驱动的特性必须在PWM控制器操作之前进行设置并且典型情况下在PWM控制器正常操作时不对其进行改变。接下来会有一个复位，在默认情况下，PWMCFG寄存器所有的位会被清空以便禁用高频斩波。

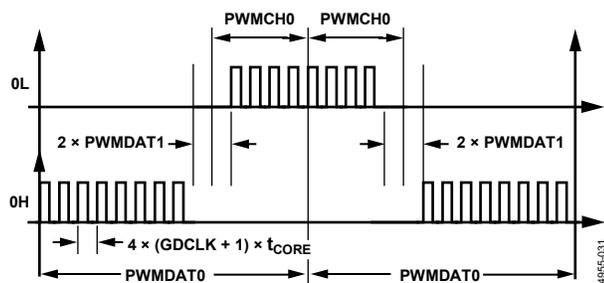


图61. 高侧和低侧开关使能的带有高频斩波的典型PWM信号

PWM关闭

当出现外部故障时，有必要以一种安全的方式迅速关闭PWM系统。PWM_{TRIP}引脚的一个低电平会使PWM控制器有一个瞬间的非同步的(独立于微转换器内核时钟)关闭动作。所有六个PWM输出会处于关闭状态，也就是低电平状态。此外，PWMSYNC脉冲禁用。PWM_{TRIP}引脚有一个内部的下拉电阻，如果这个引脚断开则这个寄存器会禁用PWM。PWM_{TRIP}引脚的状态可从PWMSTA寄存器的第3位读出。

如果一个PWM关闭命令出现，那么会产生一个PWMTRIP中断并且PWM控制器的三相时钟单元的内部时钟会停止。在PWM关闭后（例如，在一个常规PWMTRIP中断服务时）只可以通过写入所有PWMDAT0、PWMCH0、PWMCH1和PWMCH2寄存器来重启。当外部故障解除并且PWMTRIP回到高电平时，三相时钟单元的内部时钟重新开始工作并且新的占空比值会锁定在下一个PWMSYNC边界。

注意，PWMTRIP中断只出现在IRQ，PWMSYNC中断只出现在FIQ。这两个中断共用中断控制器的同一个位，所以一次只可以用一个中断。更详细的信息可参考“中断系统”部分。

PWM MMR接口

PWM模块由本节介绍的MMR进行控制。

PWMCON寄存器

名称	地址	默认值	访问
PWMCON	0xFFFFFC00	0x0000	读/写

PWMCON是一个控制寄存器，可使能PWM工作、选择数据更新速率。

表37. PWMCON MMR位功能描述

位	名称	描述
7:5	—	保留。
4	PWM_SYNCSEL	外部同步选择。置1，选择外部同步。清零，选择内部同步。
3	PWM_EXTSYNC	外部同步选择。置1，选择外部同步sync信号。清0，为异步sync信号。
2	PWMDBL	二次更新模式。置1，使能二次更新模式清零，使能单次更新模式。
1	PWM_SYNC_EN	PWM同步使能。置1，使能同步。清零，禁用同步。
0	PWMEN	PWM使能位。置1，使能PWM。清零，禁用PWM。也可由PWMTRIP自动清零。

PWMSTA寄存器

名称	地址	默认值	访问
PWMSTA	0xFFFFFC04	0x0000	读/写

PWMSTA反映PWM的状态。

表38. PWMSTA MMR位功能描述。

位	名称	描述
15:10	—	保留。
9	PWMSYNCINT	PWM Sync中断位。写入1，清除中断。
8	PWMTRIPINT	PWM触发中断位。写入1，清除中断。
3	PWMTRIP	PWMTRIP引脚的原始信号。
2:1	—	保留。
0	PWMPHASE	PWM相位。当计时器倒数时CPU置1，(前半周期)。当计时器正数时CPU清0(后半周期)。

PWMCFG 寄存器

名称	地址	默认值	访问
PWMCFG	0xFFFFFC10	0x0000	读/写

PWMCFG是一个门斩波寄存器。

表 39. PWMCFG MMR位功能描述

位	名称	描述
15:10	—	保留。
9	CHOPLO	低侧门斩波使能位。
8	CHOPHI	高侧门斩波使能位。
7:0	GDCLK	PWM门斩波周期(无符号)。

PWMEN 寄存器

名称	地址	默认值	访问
PWMEN	0xFFFFFC20	0x0000	读/写

PWMEN可使能通道输出和使能交越。其位定义如表40所示。

表 40. PWMEN MMR位功能描述

位	名称	描述
8	0H0L_XOVR	通道0输出交越使能位。置1, 可使通道0输出交越。清0, 禁用通道0输出交越。
7	1H1L_XOVR	通道1输出交越使能位。置1, 可使通道1输出交越。清0, 禁用通道1输出交越。
6	2H2L_XOVR	通道2输出交越使能位。置1, 可使通道2输出交越。清0, 禁用通道2输出交越。
5	0L_EN	0L输出使能位。置1, 禁用PWM的0L输出。清0, 使能PWM的0L输出。
4	0H_EN	0H输出使能位。置1, 禁用PWM的0H输出。清0, 使能PWM的0H输出。
3	1L_EN	1L输出使能位。置1, 禁用PWM的1L输出。清0, 使能PWM的1L输出。
2	1H_EN	1H输出使能位。置1, 禁用PWM的1H输出。清0, 使能PWM的1H输出。
1	2L_EN	2L输出使能位。置1, 禁用PWM的2L输出。清0, 使能PWM的2L输出。
0	2H_EN	2H输出使能位。置1, 禁用PWM的2H输出。清0, 使能PWM的2H输出。

PWMDAT0 寄存器

名称	地址	默认值	访问
PWMDAT0	0xFFFFFC08	0x0000	读/写

PWMDAT0是一个用于开关周期的无符号16位寄存器。

PWMDAT1 寄存器

名称	地址	默认值	访问
PWMDAT1	0xFFFFFC0C	0x0000	读/写

PWMDAT1是一个用于死区的无符号10位寄存器。

PWMCHx 寄存器

名称	地址	默认值	访问
PWMCH0	0xFFFFFC14	0x0000	读/写
PWMCH1	0xFFFFFC18	0x0000	读/写
PWMCH2	0xFFFFFC1C	0x0000	读/写

PWMCH0、PWMCH1和PWMCH2是PWM三相的通道占空比寄存器。

PWMDAT2 寄存器

名称	地址	默认值	访问
PWMDAT2	0xFFFFFC24	0x0000	读/写

PWMDAT2是一个用于PWM同步脉冲带宽的无符号10位寄存器。

通用输入/输出

ADuC7019/20/21/22/24/25/26/27/28 有 40 个通用双向 I/O (GPIO) 引脚。所有 I/O 引脚都兼容 5V 电压, 即 GPIO 支持 5V 输入电压。一般来说, GPIO 引脚都有多种功能(表 41 给出了各引脚功能说明)。默认情况下, GPIO 引脚配置为通用 I/O 引脚。

每一个 GPIO 引脚都有一个内部的上拉电阻(约为 100 k Ω), 驱动能力为 1.6mA。注意, 最多可以有 20 个驱动 1.6 mA 电流的 GPIO 引脚同时工作。可以使用 GPxPAR 寄存器控制以下端口是否使用内部上拉电阻: P0.0、P0.4、P0.5、P0.6、P0.7 及 P1 的 8 个 GPIO 口。

40 个 GPIO 引脚被分成 5 个端口, 即端口 0 到端口 4。每一个端口由 4 或 5 个 MMR 控制。

一般在复位时, CPU 会将 P0.6 默认设置改为 GPIO 功能。如果 MRST 用于外部电路, 则必须有一个外部上拉电阻来保证在 CPU 转换模式时 P0.6 的电平不降低。否则, 在复位周期时间内 P0.6 会降低。例如, 如果 MRST 需要进入掉电模式, 可在 GP0CON 寄存器重新配置。

GPIO 的输入电平在任何时间都能从 GPxDAT 寄存器中读出, 甚至在一个模式(除了 GPIO 模式)下引脚被重新配置时也可以。PLA 输入始终有效。

当 ADuC7019/20/21/22/24/25/26/27/28 进入省电模式, GPIO 引脚会保持它们的状态。

表41. GPIO引脚功能描述

端口	引脚	配置				
		00	01	10	11	
0	P0.0	GPIO	CMP	MS0	PLAI[7]	
	P0.1	GPIO	PWM2 _H	BLE		
	P0.2	GPIO	PWM2 _L	BLE		
	P0.3	GPIO	TRST	A16		ADC _{BUSY}
	P0.4	GPIO/IRQ0	PWM _{TRIP}	MS1		PLAO[1]
	P0.5	GPIO/IRQ1	ADC _{BUSY}	MS2		PLAO[2]
	P0.6	GPIO/T1	MRST			PLAO[3]
	P0.7	GPIO	ECLK/XCLK ¹	SIN		PLAO[4]
1	P1.0	GPIO/T1	SIN	SCL0	PLAI[0]	
	P1.1	GPIO	SOUT	SDA0	PLAI[1]	
	P1.2	GPIO	RTS	SCL1	PLAI[2]	
	P1.3	GPIO	CTS	SDA1	PLAI[3]	
	P1.4	GPIO/IRQ2	RI	CLK	PLAI[4]	
	P1.5	GPIO/IRQ3	DCD	MISO	PLAI[5]	
	P1.6	GPIO	DSR	MOSI	PLAI[6]	
	P1.7	GPIO	DTR	CSL	PLAO[0]	
2	P2.0	GPIO	CONV _{START} ²	SOUT	PLAO[5]	
	P2.1	GPIO	PWM0 _H	WS	PLAO[6]	
	P2.2	GPIO	PWM0 _L	RS	PLAO[7]	
	P2.3	GPIO		AE		
	P2.4	GPIO	PWM0 _H	MS0		
	P2.5	GPIO	PWM0 _L	MS1		
	P2.6	GPIO	PWM1 _H	MS2		
	P2.7	GPIO	PWM1 _L	MS3		
3	P3.0	GPIO	PWM0 _H	AD0	PLAI[8]	
	P3.1	GPIO	PWM0 _L	AD1	PLAI[9]	
	P3.2	GPIO	PWM1 _H	AD2	PLAI[10]	
	P3.3	GPIO	PWM1 _L	AD3	PLAI[11]	
	P3.4	GPIO	PWM2 _H	AD4	PLAI[12]	
	P3.5	GPIO	PWM2 _L	AD5	PLAI[13]	
	P3.6	GPIO	PWM _{TRIP}	AD6	PLAI[14]	
	P3.7	GPIO	PWM _{SYNC}	AD7	PLAI[15]	
4	P4.0	GPIO		AD8	PLAO[8]	
	P4.1	GPIO		AD9	PLAO[9]	
	P4.2	GPIO		AD10	PLAO[10]	
	P4.3	GPIO		AD11	PLAO[11]	
	P4.4	GPIO		AD12	PLAO[12]	
	P4.5	GPIO		AD13	PLAO[13]	
	P4.6	GPIO		AD14	PLAO[14]	
	P4.7	GPIO		AD15	PLAO[15]	

¹ 当配置为模式1时，默认P0.7为ECLK，或者为内核时钟输出。如果将其作为时钟输入，PLLCON的MDCLK位必须置11。

² P2.0.在所有模式下CONV_{START}信号均有效。

GPxCON 寄存器

名称	地址	默认值	访问
GP0CON	0xFFFFF400	0x00000000	读/写
GP1CON	0xFFFFF404	0x00000000	读/写
GP2CON	0xFFFFF408	0x00000000	读/写
GP3CON	0xFFFFF40C	0x00000000	读/写
GP4CON	0xFFFFF410	0x00000000	读/写

GPxCON是端口x的控制寄存器，表42为其对于端口x每一个引脚功能选择的描述。

表42. GPxCON MMR位功能描述

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

GPxPAR 寄存器

名称	地址	默认值	访问
GP0PAR	0xFFFFF42C	0x20000000	读/写
GP1PAR	0xFFFFF43C	0x00000000	读/写

GPxPAR是端口0和端口1的参数控制寄存器。注意要先写GPxPAR寄存器再写GPxDAT寄存器。

表43. GPxPAR MMR位功能描述

位	描述
31:29	保留。
28	Px.7上拉禁用。
27:25	保留。
24	Px.6上拉禁用。
23:21	保留。
20	Px.5上拉禁用。
19:17	保留。
16	Px.4上拉禁用。
15:13	保留。
12	Px.3上拉禁用。
11:9	保留。
8	Px.2上拉禁用。
7:5	保留。
4	Px.1上拉禁用。
3:1	保留。
0	Px.0上拉禁用。

GPxDAT寄存器

名称	地址	默认值	访问
GP0DAT	0xFFFFF420	0x000000XX	读/写
GP1DAT	0xFFFFF430	0x000000XX	读/写
GP2DAT	0xFFFFF440	0x000000XX	读/写
GP3DAT	0xFFFFF450	0x000000XX	读/写
GP4DAT	0xFFFFF460	0x000000XX	读/写

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44. GPxDAT MMR位功能描述

位	描述
31:24	数据传输方向位。置1，GPIO引脚作为输出。清0，GPIO引脚作为输入。
23:16	端口x数据输出。
15:8	复位时端口x引脚的状态(只读)。
7:0	端口x数据输入(只读)。

GPxSET寄存器

名称	地址	默认值	访问
GP0SET	0xFFFFF424	0x000000XX	写
GP1SET	0xFFFFF434	0x000000XX	写
GP2SET	0xFFFFF444	0x000000XX	写
GP3SET	0xFFFFF454	0x000000XX	写
GP4SET	0xFFFFF464	0x000000XX	写

GPxSET为端口x的数据置位寄存器

表45. GPxSET MMR位功能描述

位	描述
31:24	保留。
23:16	端口x置位。置1，端口x相对应的位被置位，同时GPxDAT寄存器相关的位也被置位。 清0不影响数据输出。
15:0	保留。

GPxCLR寄存器

名称	地址	默认值	访问
GP0CLR	0xFFFFF428	0x000000XX	写
GP1CLR	0xFFFFF438	0x000000XX	写
GP2CLR	0xFFFFF448	0x000000XX	写
GP3CLR	0xFFFFF458	0x000000XX	写
GP4CLR	0xFFFFF468	0x000000XX	写

GPxCLR为端口x的数据清0寄存器

表46. GPxCLR MMR位功能描述

位	描述
31:24	保留。
23:16	端口x清0位。置1，端口x相对应的位被清0，同时GPxDAT寄存器相关的位也被清0。清0不影响数据输出。
15:0	保留。

串口多路复用器

串口多路复用器为10个GPIO引脚集成了串行端口外围设备(一个SPI、UART和两个I²C)和可编程逻辑阵列(PLA)。每一个引脚都必须为其配置如表47描述的具体的I/O功能。

表47. SPM配置

SPMMUX	GPIO (00)	UART (01)	UART/I ² C/SPI (10)	PLA (11)
SPM0	P1.0	SIN	I2C0SCL	PLAI[0]
SPM1	P1.1	SOUT	I2C0SDA	PLAI[1]
SPM2	P1.2	RTS	I2C1SCL	PLAI[2]
SPM3	P1.3	CTS	I2C1SDA	PLAI[3]
SPM4	P1.4	RI	SPICLK	PLAI[4]
SPM5	P1.5	DCD	SPIMISO	PLAI[5]
SPM6	P1.6	DSR	SPIMOSI	PLAI[6]
SPM7	P1.7	DTR	SPICSL	PLAO[0]
SPM8	P0.7	ECLK/XCLK	SIN	PLAO[4]
SPM9	P2.0	CONV	SOUT	PLAO[5]

表47详细描述了每一个SPMMUX引脚的模式。这些配置需通过GP0CON、GP1CON和GP2CON寄存器来完成。默认情况下，这十个引脚被配置为通用I/O引脚。

UART串行接口

该UART外设是一个全双工的、通用异步接收/发送器。完全兼容16,450串口标准。它不但可以通过串转并的方式从一个外围器件或调制解调器接收数据，而且可以通过并转串的方式从CPU接收数据。它还内置有一个用于发生波特率的小数分频器并且支持网络寻址模式。UART的功能可通过ADuC7019/20/21/22/24/25/26/27/28的10个引脚进行设置(如表48所示)。

表48. UART信号描述

引脚	信号	描述
SPM0 (模式1)	SIN	串行数据接收。
SPM1 (模式1)	SOUT	串行数据发送。
SPM2 (模式1)	RTS	发送请求位。
SPM3 (模式1)	CTS	清除发送。
SPM4 (模式1)	RI	响铃指示。
SPM5 (模式1)	DCD	数据载波检测。
SPM6 (模式1)	DSR	数据准备就绪。
SPM7 (模式1)	DTR	数据终端准备位。
SPM8 (模式2)	SIN	串行数据接收。
SPM9 (模式2)	SOUT	串行数据发送。

在进行串行通信时，UART遵循一个非同步的协议，支持不同的字节长度、停止位和奇偶校验位。

波特率发生器

有两种方式发生UART波特率，普通的450 UART波特率发生器和小数分频器。

普通的450 UART波特率

普通456UART波特率是内核时钟的一个分频，分为两部分，低字节和高字节分别存放在COMDIV0和COMDIV1寄存器中(16位，DL)。

$$Baud\ Rate = \frac{41.78\ MHz}{2^{CD} \times 16 \times 2 \times DL}$$

表 49给出了一些常用的波特率值。

表49. 普通波特率发生器的波特率

波特率	CD	DL	实际波特率	百分比误差
9,600	0	0x88	9,600	0
19,200	0	0x44	19,200	0
115,200	0	0x0B	118,691	3
9,600	3	0x11	9600	0
19,200	3	0x08	20,400	6.25
115,200	3	0x01	163,200	41.67

小数分频器

小数分频器中集成了一个普通波特率发生器，能够产生范围更宽更精确的波特率。

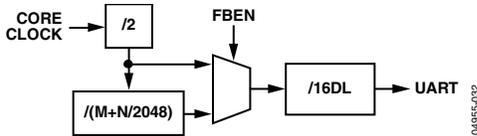


图62. 波特率生成选项

采用小数分频器的波特率计算公式如下：

$$Baud\ Rate = \frac{41.78\ MHz}{2^{CD} \times 16 \times DL \times 2 \times \left(M + \frac{N}{2048}\right)}$$

$$M + \frac{N}{2048} = \frac{41.78\ MHz}{Baud\ Rate \times 2^{CD} \times 16 \times DL \times 2}$$

例如，设置波特率为19200，CD位=3，由表49可知DL=0x08，代入公式可得：

$$M + \frac{N}{2048} = \frac{41.78\ MHz}{19200 \times 2^3 \times 16 \times 8 \times 2}$$

$$M + \frac{N}{2048} = 1.06$$

其中：

$$M = 1$$

$$N = 0.06 \times 2048 = 128$$

$$Baud\ Rate = \frac{41.78\ MHz}{2^3 \times 16 \times 8 \times 2 \times \left(\frac{128}{2048}\right)}$$

其中：

$$\text{波特率} = 19,200\ \text{bps}$$

误差 = 0%，而同样情况下，利用普通波特率发生器产生19200bps的波特率时，误差为6.25%。

UART寄存器定义

与UART接口相关的寄存器共有12个：COMTX、COMRX、COMDIV0、COMIEN0、COMDIV1、COMIID0、COMCON0、COMCON1、COMSTA0、COMSTA1、COMSCR和COMDIV2。

COMTX寄存器

名称	地址	默认值	访问
COMTX	0xFFFF0700	0x00	读/写

COMTX是一个8位发送寄存器。

COMRX寄存器

名称	地址	默认值	访问
COMRX	0xFFFF0700	0x00	读

COMRX是一个8位接收寄存器

COMDIV0寄存器

名称	地址	默认值	访问
COMDIV0	0xFFFF0700	0x00	读/写

COMDIV0是一个低字节分频锁存器。COMTX、COMRX和COMDIV0共用同一个地址。当COMCON0寄存器的第7位为0时，COMTX和COMRX可以被存取。当COMCON0寄存器的第7位为1时，存取COMDIV0。

COMIEN0寄存器

名称	地址	默认值	访问
COMIEN0	0xFFFF0704	0x00	读/写

COMIEN0是一个中断使能寄存器。

表50. COMIEN0 MMR位功能描述

位	名称	描述
7:4	-	保留。
3	EDSSI	调制解调器状态中断使能位。置1，在COMSTA1[3:1]中任何一个被置1时产生一个中断；由用户清0。
2	ELSI	Rx状态中断使能位。置1，在COMSTA0[3:0]中任何一个被置1时产生一个中断。由用户清0。
1	ETBEI	发送缓冲器空中断使能位。置1，在数据发送期间，当发送数据缓冲器空时会产生一个中断。由用户清0。
0	ERBFI	接收缓冲器满中断使能位。置1，在数据接收期间，当接收数据缓冲器满时会产生一个中断。由用户清0。

COMDIV1寄存器

名称	地址	默认值	访问
COMDIV1	0xFFFF0704	0x00	读/写

COMDIV1是一个高字节分频锁存寄存器

COMIID0寄存器

名称	地址	默认值	访问
COMIID0	0xFFFF0708	0x01	读

COMIID0为中断识别寄存器。

表51. COMIID0 MMR位功能描述

位2:1 状态位	第0位 NINT	优先级	定义	清0 操作
00	1	—	无中断	—
11	0	1	接收线状态中断	读COMSTA0。
10	0	2	接收缓冲器满中 断。	读COMRX。
01	0	3	发送缓冲器空 断。	向COMTX中 写数据或读 COMIID0。
00	0	4	调制解调器状态 中断。	读COMSTA1。

COMCON0寄存器

名称	地址	默认值	访问
COMCON0	0xFFFFF070C	0x00	读/写

COMCON0是线路控制寄存器。

表52. COMCON0 MMR位功能描述

位	名称	描述
7	DLAB	分频锁存访问。置1，可以访问COMDIV0和COMDIV1寄存器。清0，可以访问COMRX和COMTX寄存器。
6	BRK	设置断点位。置1，可以强制SOUT至零。清0，工作在正常模式。
5	SP	附加奇偶校验位。置1，则附加奇偶校验位：若EPS = 1, PEN = 1, 附加位为1；若EPS = 0, PEN = 1则附加位为0。
4	EPS	偶校验选择位。置1为偶校验。清0为奇校验。
3	PEN	奇偶校验使能位。置1，发送奇偶校验位并进行奇偶校验。清0，则不发送奇偶校验位也不进行奇偶校验。
2	STOP	停止位。置1，如果数据长度为5位则传送1.5个停止位；如果数据长度为6、7或8位则传送2个停止位。不管停止位有多少，接收器只检查第一个停止位。清0，在发送的数据中只有一个停止位。
1:0	WLS	数据长度选择位：00 = 5位、01 = 6位、10 = 7位、11 = 8位

COMCON1寄存器

名称	地址	默认值	访问
COMCON1	0xFFFFF0710	0x00	读/写

COMCON1为调制解调器控制寄存器。

表53. COMCON1 MMR位功能描述

位	名称	描述
7:5		保留。
4	LOOPBACK	回送。置1，回送模式使能。在回送模式下，SOUT强制为高电平。调制解调器的信号直接连接到状态输入，即RTS连接到CTS，DTR连接到DSR。清0，返回正常模式。
3	PEN	奇偶校验使能位。置1，发送奇偶校验位并进行奇偶校验。清0，则不发送奇偶校验位也不进行奇偶校验。
2	STOP	停止位。置1，如果数据长度为5位则传送1.5个停止位，如果数据长度为6、7或8位则传送2个停止位。不管停止位有多少，接收器只检查第一个停止位。清0，在发送的数据中只有一个停止位。
1	RTS	发送请求位。置1，RTS强制输出0。清0，RTS强制输出1。
0	DTR	数据终端准备位。置1，DTS强制输出0。清0，DTS强制输出1。

COMSTA0寄存器

名称	地址	默认值	访问
COMSTA0	0xFFFFF0714	0x60	读

COMSTA0为总线状态寄存器

表54. COMSTA0 MMR位功能描述

位	名称	描述
7		保留。
6	TEMT	COMTX空状态指示位。在COMTX空时该位自动置1。如果COMTX被写入数据，该位自动清0。
5	THRE	COMTX和COMRX空状态指示位。如果COMTX和COMRX同时空该位自动置1。当其中一个寄存器接收到数据后该位自动清0。
4	BI	中断错误。当SIN保持低电平超过最大字长时该位置1。否则该位自动清0。
3	FE	帧错误。当产生无效停止位时置1。否则该位自动清0。
2	PE	奇偶校验错误。当出现奇偶校验错误时该位置1。否则该位自动清0。
1	OE	溢出错误。如果在读之前数据被覆盖该位自动置1。否则该位自动清0。
0	DR	数据准备就绪位。COMRX满时该位自动置1。读COMRX后该位清0。

COMSTA1 寄存器

名称	地址	默认值	访问
COMSTA1	0xFFFF0718	0x00	读

COMSTA1为调制解调器状态寄存器。

表55. COMSTA1 MMR位功能描述

位	名称	描述
7	DCD	数据载波检测。
6	RI	响铃指示。
5	DSR	数据准备就绪。
4	CTS	清除发送。
3	DDCD	三角波DCD。在上一次读COMSTA1后，如果DCD状态改变了则自动置1。读COMSTA1后自动清0。
2	TERI	下降沿RI。在上一次读COMSTA1后，如果RI由0变为1则置1。读COMSTA1后自动清0。
1	DDSR	三角波DSR。在上一次读COMSTA1后，如果DSR改变状态则自动置1。读COMSTA1后自动清0。
0	DCTS	三角波CTS。在上一次读COMSTA1后，如果CTS改变状态则自动置1。读COMSTA1后自动清0。

COMSCR 寄存器

名称	地址	默认值	访问
COMSCR	0xFFFF071C	0x00	读/写

COMSCR是一个8位寄存器，用来暂时存储数据。它也可用于网络寻址UART模式。

COMDIV2 寄存器

名称	地址	默认值	访问
COMDIV2	0xFFFF072C	0x0000	读/写

COMDIV2为16位小数波特率分频寄存器

表56. COMDIV2 MMR位功能描述

位	名称	描述
15	FBEN	小数波特率使能位。置1，使能小数分频器。清0，则使能标准450 UART波特率发生器。
14:13		保留。
12:11	FBM[1:0]	M。如果FBM = 0, M = 4(具体使用请看小数分频器部分)。
10:0	FBN[10:0]	N。(具体使用请看小数分频器部分)。

网络寻址UART模式

这个模式下微转换器可连接到一个有256个节点的串行网络，或者以硬件作为单主，但可以通过软件实现多主连接。COMIEN1寄存器的第7位(ENAM)必须被置位使得UART进入网络寻址模式(参见表57)。注意：在这个模式下不能进行奇偶校验。

网络寻址UART寄存器定义

还有4个寄存器只用于网络寻址模式，它们是：COMIEN0、COMIEN1、COMIID1和COMADR。在网络寻址模式下，COMIEN1寄存器的最低位是网络地址发送控制位。如果该位置1，器件发送地址数据。清0，则发送一般数据。例如：下面就是主机发送从机地址代码的一段程序：

```
COMIEN1 = 0xE7;
//Setting ENAM,E9BT, E9BR, ETD, NABP
COMTX = 0xA0; // Slave address is 0xA0
while(!(0x020==(COMSTA0 & 0x020))){}
//wait for adr tx to finish.
COMIEN1 = 0xE6; // Clear NAB bit to indicate Data is coming
COMTX = 0x55; // Tx data to slave:0x55
```

COMIEN1 寄存器

名称	地址	默认值	访问
COMIEN1	0xFFFF0720	0x04	读/写

COMIEN1为8位网络使能寄存器。

表 57. COMIEN1 MMR位功能描述

位	名称	描述
7	ENAM	网络寻址模式使能位。该位置1，进入网络寻址模式。清0，则退出网络寻址模式。
6	E9BT	第9位发送使能位。该位置1，发送第9位，此时ENAM必须被置位。清0，则禁用发送第9位。
5	E9BR	第9位接收使能位。该位置1，接收第9位。此时ENAM必须被置位。清0，则禁用接收第9位。
4	ENI	网络中断使能位。
3	E9BD	字长。该位置1，数据长度为9位，此时E9BT必须被清0。该位清0，数据长度为8位。
2	ETD	发送引脚驱动使能位。该位置1，在从模式或多主模式下SOUT引脚作为一个输出。清0，SOUT变为三态位。
1	NABP	网络地址位。中断极性位。
0	NAB	地址数据选择位(如果NABP= 1)。该位置1，发送从机的地址。清0，发送数据

COMIID1 寄存器

名称	地址	默认值	访问
COMIID1	0xFFFF0724	0x01	读

COMIID1为8位网络中断寄存器。第7位到第4位是保留的(参见表58)。

表58. COMIID1 MMR位功能描述

位3:1 状态位	第0位 NINT	优先级	定义	清0 操作
000	1	—	无中断。	—
110	0	2	网络地址匹配	读COMRX.
101	0	3	地址发射缓冲器 清空	数据写入 COMTX或读 COMIID0。
011	0	1	接收线性状态中 断	读
010	0	2	接收缓冲器满中 断。	COMSTA0读 COMRX.
001	0	3	发射缓冲器清空 中断	数据写入 COMTX 或 读COMIID0。
000	0	4	调制解调器状态 中断。	读COMSTA1

注意，当接收一个网络地址中断时，从机必须保证COMIEN0的第0位被置1，也就是使能接收缓冲器满中断。

COMADR 寄存器

名称	地址	默认值	访问
COMADR	0xFFFF0728	0xAA	读/写

COMADR是一个8位读/写网络地址寄存器，其中保存的地址由网络寻址UART校验。一旦接收到这个地址，处理器会产生中断并且将COMIID1合适的状态位置1。

串行外设接口(SPI)

ADuC7019/20/21/22/24/25/26/27/28集成了完整的片内硬件串行外设接口(SPI)。SPI是一个工业标准同步串行接口，它允许长为8位数据同步发送和同步接收，也就是说在全双工时比特率最大可达3.48 Mb，如表59所示。SPI接口不是由内核时钟(CD)分频器的控制位操控。在主模式下POWCON[2:0] = 6或7。

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、SCL和 \overline{CS} 。

MISO(主机输入，从机输出)引脚

在主模式下，MISO引脚被配置为输入；在从模式下，配置为输出。主机的MISO数据输入线应该连接到从机的MISO数据输出线。传送的数据是以字节（8位）为单位的串行数据，起始为MSB。

MOSI(主机输出，从机输入)引脚

在主模式下，MISO引脚被配置为输出，在从模式下，配置为输入。主机的MISO数据输出线应该连接到从机的MISO数据输入线。传送的数据是以字节（8位）为单位的串行数据，起始为MSB。

SCL(串行时钟输入/输出)引脚

主机串行时钟(SCL)用于通过MOSI SCL周期同步的发送和接收数据。所以，一个字节的发送或接收需要8个SCL周期。在主模式下，SCL引脚被配置为输出，在从模式下为输入。

在主模式下，时钟的极性和相位由SPICON寄存器控制，比特率在SPIDIV寄存器中定义，公式如下：

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

而SPI时钟的最大速率取决于时钟分频位，如表59所示：

表59. 主模式下SPI速度与时钟分频位对照表

CD位	0	1	2	3	4	5
SPIDIV (16进制)	0x05	0x0B	0x17	0x2F	0x5F	0xBF
SPI速度 (MHz)	3.482	1.741	0.870	0.435	0.218	0.109

在从模式下，SPICON寄存器必须配置预期输入时钟的相位和极性。一般在CD = 0时，从机从外部主机接收数据速率可达10.4 Mb。从模式下最大速度的计算公式如下所示：

$$f_{SERIAL\ CLOCK} = \frac{f_{HCLK}}{4}$$

在主模式和从模式下，数据都是在SCL信号的一个沿发送，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。

片选(\overline{CS} 输入)引脚

在SPI从模式时，传送是否开始是通过 \overline{CS} 引脚声明，该引脚为一个低电平有效信号。SPI端口开始发送和接收8位的数据直到 \overline{CS} 声明其结束时停止。注意：在从模式下 \overline{CS} 引脚总是为输入。

SPI寄存器

以下MMR寄存器用于控制SPI接口：SPISTA、SPIRX、SPITX、SPIDIV和SPICON。

SPISTA 寄存器

名称	地址	默认值	访问
SPISTA	0xFFFF0A00	0x00	读/写

SPISTA是一个8位只读状态寄存器。只有这个寄存器的第1位或第4位产生中断。SPICON寄存器的第6位决定哪一个位产生中断。

表60. SPISTA MMR位功能描述

位	描述
7:6	保留。
5	SPIRX数据寄存器上溢状态位。如果SPIRX上溢该位置1。读SPIRX寄存器后该位清0。
4	SPIRX数据寄存器中断请求位。如果该寄存器的第3位或第5位被置1则该位自动置1。读SPIRX寄存器后该位清0。
3	SPIRX数据寄存器满状态位。如果SPIRX寄存器中存在一个有效数据则该位自动置1。读SPIRX寄存器后该位清0。
2	SPITX数据寄存器下溢状态位如果SPITX下溢则该位自动置1。向SPITX写入数据后该位清0。
1	SPITX数据寄存器中断请求。如果该寄存器的第0位被清0或第2位被置1则该位自动置1。向SPITX写入数据或发送完成SPI停止工作时该位清0。
0	SPITX数据寄存器空状态位。向SPITX寄存器写入数据后该位置1，并且在数据发送过程中这个位始终置1。当SPITX空时该位清0。

SPIRX寄存器

名称	地址	默认值	访问
SPIRX	0xFFFF0A04	0x00	读

SPIRX为8位只读接收寄存器。

SPITX寄存器

名称	地址	默认值	访问
SPITX	0xFFFF0A08	0x00	写

为8位只写发送寄存器。

SPIDIV寄存器

名称	地址	默认值	访问
SPIDIV	0xFFFF0A0C	0x1B	读/写

SPIDIV为8位串行时钟分频寄存器。

SPICON寄存器

名称	地址	默认值	访问
SPICON	0xFFFF0A10	0x0000	读/写

SPICON为16位控制寄存器。

表61. SPICON MMR位功能描述

位	描述	功能
15:13	保留。	—
12	连续发送使能位。	置1，连续发送使能。在主模式下，会连续发送数据直到TX寄存器中无有效数据时停止。在每一个8位串行发送连续工作期间会始终侦听 \overline{CS} 的状态直到TX寄存器空。清0，禁用连续发送。每一次发送都是单独的8位串行发送。如果SPITX寄存器中存在有效数据，那么在一个停转周期后会重新开始发送数据。
11	数据回送使能控制位。	置1，器件的MISO连接到MOSI，用于测试软件。清0，返回正常模式。
10	从机输出使能位。	置1，使能从机输出。清0，禁用从机输出。
9	从机输入选择使能位。	在主模式下，置1使能主机输出。清0，禁用主机输出。
8	SPIRX上溢覆盖使能	置1，新接收到的串行数据覆盖RX寄存器中的有效数据。 清0，新接收到的串行数据会被丢弃。
7	SPITX下溢模式选择位。	置1，发送0；清0，发送前一个数据。
6	发送和中断模式选择位。	置1，向SPITX寄存器写入数据时开始发送。只有当TX空时产生中断。 清0，读SPIRX寄存器开始发送。只有当RX满时产生中断。
5	LSB先发使能位。	置1，先发LSB。清0，先发MSB。
4	保留。	—
3	串行时钟极性模式选择位。	置1，串行时钟低电平有效。清0，串行时钟高电平有效。
2	串行时钟相位模式选择位。	置1，串行时钟脉冲出现在每一个串行位传输的起始位置。清0，串行时钟脉冲出现在每一个串行位传输的末尾。
1	主模式使能位。	置1，使能主模式。清0，使能从模式。
0	SPI使能位。	置1，使能SPI。清0，禁用SPI。

I²C兼容接口

ADuC7019/20/21/22/24/25/26/27/28 支持两个特许的I²C接口。I²C接口既可配置为一个硬件主机又可配置为一个完整的从从。因为这两个I²C接口是相同的，这里只对I2C0进行详细描述。注意两个主机和一个从机都拥有相互独立的中断（详见“中断系统”部分）。

注意，当被配置为一个I²C主机时，ADuC7019/20/21/22/24/25/26/27/28不能产生一个重复起始条件。

一个标准的I²C接口有两条线，一条串行数据线SDA，一条串行时钟线SCL，它们按照“线与”的方式进行配置，这允许在多器件系统下进行仲裁。这两个引脚需要接外部上拉电阻。电阻典型上拉值为10kΩ。

I²C总线系统的外设地址由用户编程设定。当发送器不工作时这个ID在任何时候都可以修正。用户可以对接口进行配置以便对四个从属地址进行应答。

一个I²C系统的发送序列包括用一个主机开启一个发送器，当总线空闲时通过产生一个起始条件来实现。在起始地址发送期间，主机发送从机的地址和数据发送器的方向。如果主机没有仲裁失效并且从机应答，那么数据开始发送。这种情况会持续直到主机发送一个停止条件，然后总线变为空闲。

一个I²C设备在任何时候只能配置成一个主机或一个从机。同一个I²C通道不能同时支持主模式和从模式。

串行时钟发生器

系统中的I²C主机会为发送器产生串行时钟。主机通道经过配置可工作在快速模式(400 kHz)或标准模式(100 kHz)。

I2C0DIV MMR的波特率的定义如下：

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

其中：

f_{UCLK} = 分频之前的时钟。

$DIVH$ = 时钟高周期。

$DIVL$ = 时钟低周期。

因而，如果希望串行时钟为100 kHz，

应该配置 $DIVH = DIVL = 0xCF$

如果希望串行时钟为400 kHz，

应该配置 $DIVH = 0x28, DIVL = 0x3C$

I2CxDIV寄存器对应于DIVH：DIVL。

从机地址

寄存器I2C0ID0、I2C0ID1、I2C0ID2和I2C0ID3包含设备的ID。设备将四个I2C0IDx寄存器中的数据与地址字节做比较。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数据的7个MSB相同。在地址识别过程中ID寄存器的LSB（发送器的方向位）忽略。

I²C寄存器

I²C外设接口包括本节讨论的18个MMR。

I2CxMSTA寄存器

名称	地址	默认值	访问
I2C0MSTA	0xFFFF0800	0x00	读/写
I2C1MSTA	0xFFFF0900	0x00	读/写

I2CxMSTA为主机通道状态寄存器。

表62. I2C0MSTA MMR位功能描述

位	访问类型	描述
7	读/写	主机发送FIFO清空位。置1，清空主机发送FIFO。一旦主机发送FIFO被清空，则该位会自动清0。该位也可以清空从机接收FIFO。
6	读	主机忙。如果主机忙，该位自动置1。否则自动清0。
5	读	仲裁失效。在多主机模式下，如果另一个主机占用总线，该位置1；如果总线空闲则自动清0。
4	读	无效应答NACK。如果从机没有地址应答，该位自动置1。读I2C0MSTA寄存器后自动清0。
3	读	主机接收中断请求。从机接收数据后该位置1。读I2C0MRX寄存器后自动清0。
2	读	主机发送中断请求。在一次发送结束时该位置1。向I2C0MTX寄存器写入数据后自动清0。
1	读	主机发送FIFO下溢。如果主机发送FIFO下溢该位自动置1。向I2C0MTX寄存器写入数据后自动清0。
0	读	主机发送FIFO空。如果从机发送FIFO空该位置1。向I2C0MTX寄存器写入两次数据后自动清0。

I2CxSSTA寄存器

名称	地址	默认值	访问
I2C0SSTA	0xFFFF0804	0x01	读
I2C1SSTA	0xFFFF0904	0x01	读

I2CxSSTA为从机通道状态寄存器。

表63. I2C0SSTA MMR位功能描述

位	值	描述
31:15		保留。这些位应该写入0。
14		起始解码位。如果器件接收到一个有效的起始条件+地址匹配，则由硬件对该位置1。当产生一个I ² C停止条件或I ² C广播复位时该位清0。
13		重复起始解码位。如果器件接收到一个有效的重复起始条件+地址匹配，则由硬件对该位置1。当产生一个I ² C停止条件、广播复位或读I2CSSTA寄存器时该位清0。
12:11		ID解码位。
	00	接收到的地址匹配ID寄存器0。
	01	接收到的地址匹配ID寄存器1。
	10	接收到的地址匹配ID寄存器2。
	11	接收到的地址匹配ID寄存器3。
10		起始和地址匹配中断后停止。在上一个I ² C起始条件+地址匹配后，如果从机接收到一个I ² C停止条件，则由硬件对该位置1。读I ² C0SSTA寄存器后该位清0。
9:8		广播ID。
	00	无广播。
	01	广播复位和编程地址。
	10	广播编程地址。
	11	广播替代ID匹配。
7		广播中断。从机接收到任何形式的广播后置1。对I2CxCFG寄存器的第8位置1后清0。如果是广播复位，所有的寄存器恢复为默认值。如果是硬件广播，接收FIFO会保存广播的第2个字节。除了重编器件地址外与I2C0ALT寄存器相似。更多资料请参阅I2C总线说明书，版本2.1，2000年1月。
6		从机忙。如果从机忙，该位置1。否则该位自动清0。
5		无效应答NACK。主机需要数据却无法获得数据时该位置1。读I2C0SSTA寄存器后该位自动清0。
4		从机接收FIFO溢出。如果从机接收FIFO溢出该位置1。读I2C0SSTA寄存器后该位自动清0。
3		从机接收中断请求。从机接收数据后该位置1。读I2C0SRX寄存器或清空FIFO后该位自动清0。
2		从机发送中断请求。从机发送数据结束后该位置1。向I2C0STX寄存器写入数据后该位自动清0。
1		从机发送FIFO下溢。如果从机发送FIFO下溢该位置1。向I2C0SSTA寄存器写入数据后该位自动清0。
0		从机发送FIFO空。如果从机发送FIFO空该位置1。向I2C0STX寄存器写入两次数据后该位自动清0。

I2CxSRX寄存器

名称	地址	默认值	访问
I2C0SRX	0xFFFF0808	0x00	读
I2C1SRX	0xFFFF0908	0x00	读

I2CxSRX为从机通道接收寄存器。

I2CxSTX寄存器

名称	地址	默认值	访问
I2C0STX	0xFFFF080C	0x00	写
I2C1STX	0xFFFF090C	0x00	写

I2CxSTX为从机通道发送寄存器。

I2CxMRX寄存器

名称	地址	默认值	访问
I2C0MRX	0xFFFF0810	0x00	读
I2C1MRX	0xFFFF0910	0x00	读

I2CxMRX为主机通道接收寄存器。

I2CxMTX寄存器

名称	地址	默认值	访问
I2C0MTX	0xFFFF0814	0x00	写
I2C1MTX	0xFFFF0914	0x00	写

I2CxMTX为主机通道发送寄存器。

I2CxCNT寄存器

名称	地址	默认值	访问
I2C0CNT	0xFFFF0818	0x00	读/写
I2C1CNT	0xFFFF0918	0x00	读/写

I2CxCNT是一个3位主机接收数据计数寄存器。该寄存器中存放的数据为主机将要将从机读取数据的个数，每接收一个字节该寄存器减1。默认时，该寄存器中的值为0，表示接收1字节数据。

I2CxADR寄存器

名称	地址	默认值	访问
I2C0ADR	0xFFFF081C	0x00	读/写
I2C1ADR	0xFFFF091C	0x00	读/写

I2CxADR是主机地址字节寄存器。其中存放的数据是主机要通信的器件地址。当主机使能位置位，如果I2CxMTX寄存器中无有效数据，在一个主机传输序列的开始自动发送。

I2CxBYTE寄存器

名称	地址	默认值	访问
I2C0BYTE	0xFFFF0824	0x00	读/写
I2C1BYTE	0xFFFF0924	0x00	读/写

I2CxBYTE是广播字节寄存器。写入广播字节寄存器的数据不会直接进入发送FIFO。该字节数据一般在地址之前传输序列的开始发送出去。一旦该字节数据被发送并且得到有效应答，I²C总线就会等待另一个数据写入I2CxBYTE或一个地址写入地址寄存器。

ADuC7019/20/21/22/24/25/26/27/28/29

I2CxALT寄存器

名称	地址	默认值	访问
I2C0ALT	0xFFFF0828	0x00	读/写
I2C1ALT	0xFFFF0928	0x00	读/写

I2CxALT是用于从模式的硬件广播ID寄存器。

I2xCFG寄存器

名称	地址	默认值	访问
I2C0CFG	0xFFFF082C	0x00	读/写
I2C1CFG	0xFFFF092C	0x00	读/写

I2xCFG为配置寄存器。

表64. I2C0CFG MMR位功能描述

位	描述
31:5	保留。这些位应该由用户写入0。
14	停止中断使能位。置1，在接收到一个停止条件之前、接收到一个有效的起始条件以及地址匹配之后产生中断。清0，在接收到一个停止条件之前不产生中断。
13	保留。
12	保留。
11	时钟延展使能位 (SCL保持低速率)。置1，使能SCL线时钟延展。清0则禁用SCL线时钟延展。
10	保留。
9	从机发送FIFO中断请求使能位。置1，禁用从机发送FIFO中断请求。清0，在读/写位的时钟负脉冲后产生一个中断请求。如果从机发送FIFO空时，用户可以向其中写入数据。在时钟速率为400 kbps并且内核时钟频率为41.78 MHz时，考虑到中断延迟时间，用户有45个时钟周期的时间操作。
8	广播状态位清0。置1，清0广播状态位。在广播状态位清0后由硬件自动清0。
7	主机串行时钟使能位。置1，在主模式使能串行时钟。清0，在主模式禁用串行时钟。
6	回送使能位。置1，内部发送端连接到内部接收端，用于测试用户软件。清0，正常工作。
5	启动延时取消位。置1，在多主模式下，如果仲裁失效，主机立即尝试再次发送；清0，启动延时，在仲裁失效后，主机在尝试再次发送数据前等待。
4	硬件广播使能位。该位置1，如果I2xCFG寄存器的该位和第3位置1并且已经接收到一个广播信号(地址为0x00)和1字节数据，器件检测比较接收寄存器和I2C0ALT中的数据，如果数据匹配，器件接收到一个广播。这个功能主要用于当器件需要紧急呼叫一个主机而又不知道呼叫哪一个时，这是呼叫“可能相关某器件”。ADuC7019/20/21/22/24/25/26/27/28寻找这些地址。故器件需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，然后其中合适的主机会同这个器件连接进行需要的操作。根据2000年1月的I ² C总线准则，I2C0ALT寄存器的LSB应该始终写入1。
3	广播使能位。该位置1，使能从机为I ² C广播发送有效应答，写地址0x00。然后器件会识别为一个数据位。如果器件接收到的数据是0x06，也就是由硬件复位和写从机地址的可编程部分，根据2000年1月的I ² C总线准则此时I ² C接口复位。这个命令可用于复位整个I ² C系统。广播中断状态位会建立一个广播，在复位后用户必须通过设置I ² C接口进行合适的操作。如果接收到的数据字节为0x04，也就是由硬件写子机地址可编程部分，广播中断状态位会建立一个广播，在复位后用户必须通过重新编程器件地址进行合适的操作。
2	保留。
1	主机使能位。该位置1，使能主机I ² C通道。该位清0，禁用主机I ² C通道。
0	从机使能位。该位置1，使能从机I ² C通道，此时会监视从机传输序列中的数据以寻找存放在I2C0ID0、I2C0ID1、I2C0ID2和I2C0ID3中的器件地址，如果识别了器件地址，就会加入从机传输序列中；该位清0，禁用从机I ² C通道。

I2CxDIV 寄存器

名称	地址	默认值	访问
I2C0DIV	0xFFFF0830	0x1F1F	读/写
I2C1DIV	0xFFFF0930	0x1F1F	读/写

I2CxDIV为时钟分频寄存器。

I2CxIDx 寄存器

名称	地址	默认值	访问
I2C0ID0	0xFFFF0838	0x00	读/写
I2C0ID1	0xFFFF083C	0x00	读/写
I2C0ID2	0xFFFF0840	0x00	读/写
I2C0ID3	0xFFFF0844	0x00	读/写
I2C1ID0	0xFFFF0938	0x00	读/写
I2C1ID1	0xFFFF093C	0x00	读/写
I2C1ID2	0xFFFF0940	0x00	读/写
I2C1ID3	0xFFFF0944	0x00	读/写

I2CxID0、I2CxID1、I2CxID2和I2CxID3是I2Cx的从机地址设备ID寄存器。

I2CxCCNT 寄存器

名称	地址	默认值	访问
I2C0CCNT	0xFFFF0848	0x01	读/写
I2C1CCNT	0xFFFF0948	0x01	读/写

I2CxCCNT是8位起始/停止计数寄存器。它可以在起始或停止条件出现时使SDA保持低电平。

I2CxFSTA 寄存器

名称	地址	默认值	访问
I2C0FSTA	0xFFFF084C	0x0000	读/写
I2C1FSTA	0xFFFF094C	0x0000	读/写

I2CxFSTA为FIFO状态寄存器。

表65. I2C0FSTA MMR位功能描述

位	访问类型	值	描述
15:10	读/写		保留。
9			主机发送FIFO清空位。置1，清空主机发送FIFO； 当主机发送FIFO已经被清空，该位自动清0；该位也可以清空从机接收FIFO。
8	读/写		从机发送FIFO清空位。置1，清空从机发送FIFO；当从机发送FIFO已经被清空，该位自动清0。
7:6			主发送FIFO状态位。
	读	00	FIFO空。
		01	向FIFO写字节数据。
		10	FIFO中有1字节数据。
	读	11	FIFO满。
		00	主机发送FIFO状态位。
		01	FIFO空。
	读	10	向FIFO写字节数据。
		11	FIFO中有1字节数据。
		00	FIFO满。
5:4	读	00	从机接收FIFO状态位。
01		FIFO空。	
10		FIFO中有1字节数据。	
	读	11	FIFO满。
		00	从机接收FIFO状态位。
		01	FIFO空。
	读	10	向FIFO写字节数据。
		11	FIFO中有1字节数据。
		00	FIFO满。
3:2	读	00	从机接收FIFO状态位。
01		FIFO空。	
10		向FIFO写字节数据。	
	读	11	FIFO中有1字节数据。
		00	FIFO满。
		01	从机接收FIFO状态位。
	读	10	FIFO空。
		11	向FIFO写字节数据。
		00	FIFO中有1字节数据。
1:0	读	01	FIFO满。
10		从机接收FIFO状态位。	
11		FIFO空。	
	读	00	向FIFO写字节数据。
		10	FIFO中有1字节数据。
		11	FIFO满。

ADuC7019/20/21/22/24/25/26/27/28/29

可编程逻辑阵列(PLA)

每一个ADuC7019/20/21/22/24/25/26/27/28都集成有一个完整的可编程逻辑阵列，它由两个相互独立但内部连接的PLA模块组成。每一个模块包括8个PLA单元，所以每种器件共有16个PLA单元。

每一个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63所示。

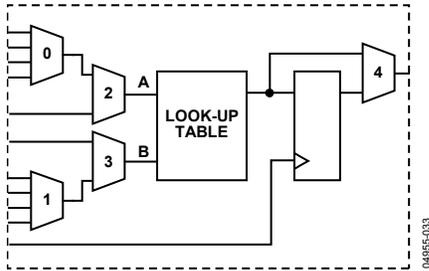


图63. PLA单元

每一个ADuC7019/20/21/ 22/24/25/26/27/28芯片上共有30个GPIO引脚可用于PLA。其中包括16个输入引脚和14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。注意，比较器输出也属于16个输入引脚之一。可使用一系列用户存储器映射寄存器（MMR）对PLA进行配置。

PLA的输出可以连接到内部中断系统、ADC的 $\overline{\text{CONV}}_{\text{START}}$ 信号、一个MMR或者16个PLA输出引脚中的任何一个。

可通过以下方式对两个模块进行互连：

- 模块1单元15的输出可以馈入到模块0单元0的Mux0
- 模块0单元7的输出可以馈入到模块1单元8的Mux0

表66. 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	P1.0	P1.7	8	P3.0	P4.0
1	P1.1	P0.4	9	P3.1	P4.1
2	P1.2	P0.5	10	P3.2	P4.2
3	P1.3	P0.6	11	P3.3	P4.3
4	P1.4	P0.7	12	P3.4	P4.4
5	P1.5	P2.0	13	P3.5	P4.5
6	P1.6	P2.1	14	P3.6	P4.6
7	P0.0	P2.2	15	P3.7	P4.7

PLA MMR接口

PLA外设接口包括22个MMR，以下是对它们的具体描述。

PLAELMx寄存器

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

PLAELMx是单元0到单元15的控制寄存器。通过它们可以配置每一个单元的输入和输出多路复用器、在查询表中选择功能并且选择旁路或使用触发器。参见表67和表70。

表67. PLAELMx MMR位功能描述

位	值	描述
31:11		保留。
10:9		Mux0控制位(见表70)。
8:7		Mux1控制位(见表70)。
6		Mux 2控制位。置1，选择Mux 0的输出。清0，从PLADIN中选择位值。
5		Mux3控制位。置1，选择特定单元的输入引脚。清0，选择Mux1的输出。
4:1		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	A。
	1101	A或B非。
	1110	或。
	1111	1。
0		Mux 4控制位。置1，旁路触发器。清0，使用触发器(默认为清0)。

PLACLK寄存器

名称	地址	默认值	访问
PLACLK	0xFFFF0B40	0x00	读/写

PLACLK是模块0和模块1触发器的时钟选择寄存器。注意当使用GPIO引脚作为PLA模块的时钟输入时，最大频率为44 MHz。

表68. PLACLK MMR位功能描述

位	值	描述
7		保留。
6:4		模块1时钟源选择。
	000	P0.5引脚连接的GPIO时钟。
	001	P0.0引脚连接的GPIO时钟。
	010	P0.7引脚连接的GPIO时钟。
	011	HCLK。
	100	OCLK (32.768 kHz)。
	101	定时器1溢出。
	其他	保留。
3		保留。
2:0		模块0时钟源选择。
	000	P0.5引脚连接的GPIO时钟。
	001	P0.0引脚连接的GPIO时钟。
	010	P0.7引脚连接的GPIO时钟。
	011	HCLK。
	100	OCLK (32.768 kHz)。
	101	定时器1溢出。
	其他	保留。

PLAIRQ寄存器

名称	地址	默认值	访问
PLAIRQ	0xFFFF0B44	0x00000000	读/写

PLAIRQ可以使能IRQ0和/或IRQ1并且选择IRQ中断源。

表69. PLAIRQ MMR位功能描述

位	值	描述
15:13		保留。
12		PLA IRQ1使能位。置1，使能PLA输出IRQ1。清0，禁用PLA输出IRQ1。
11:8		PLA IRQ1中断源。
	0000	PLA单元0。
	0001	PLA单元1。
	1111	PLA单元15。
7:5		保留。
4		PLA IRQ0使能位。置1，使能PLA输出IRQ0。清0，禁用PLA输出IRQ0。
3:0		PLA IRQ0中断源。
	0000	PLA单元0。
	0001	PLA单元1。
	1111	PLA单元15。

表70. 反馈配置

位	值	PLAELM0	PLAELM1至PLAELM7	PLAELM8	PLAELM9至PLAELM15
10:9	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
8:7	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	地址	默认值	访问
PLAADC	0xFFFF0B48	0x00000000	读/写

PLAADC是配置PLA作为ADC起始转换信号来源的寄存器。

表71. PLAADC MMR位功能描述

位	值	描述
31:5		保留。
4		ADC起始转换使能位。置1，使能从PLA启动ADC转换。清0，禁用从PLA启动ADC转换。
3:0		ADC转换起始源选择位。
	0000	PLA单元0。
	0001	PLA单元1。
	1111	PLA单元15。

PLADIN 寄存器

名称	地址	默认值	访问
PLADIN	0xFFFF0B4C	0x00000000	读/写

PLADIN是一个PLA的数据输入MMR。

表72. PLADIN MMR位功能描述

位	描述
31:16	保留。
15:0	单元15~单元0的输入位。

PLADOUT 寄存器

名称	地址	默认值	访问
PLADOUT	0xFFFF0B50	0x00000000	读

PLADOUT是一个PLA的数据输出MMR。这个寄存器是始终更新的。

表73. PLADOUT MMR位功能描述

位	描述
31:16	保留。
15:0	单元15~单元0输出位。

PLALCK 寄存器

名称	地址	默认值	访问
PLALCK	0xFFFF0B54	0x00	写

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 MMR的值都不允许修改。开发系统提供了一套PLA工具，通过它可以很容得对PLA进行配置。

处理器基准外设

中断系统

ADuC7019/20/21/22/24/25/26/27/28共有23个中断源，它们都由中断控制器控制。大多数中断都是由片内外围设备产生，例如ADC和UART。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内核只识别以下两种中断之一：普通中断请求IRQ或快速中断请求FIQ。所有中断都可以被单独屏蔽。

通过9个与中断有关的寄存器来管理中断系统的控制和配置，4个用于控制IRQ，4个用于控制FIQ。一个额外的存储器映射寄存器用于选择可编程中断源。每一个IRQ和FIQ寄存器中的控制位(除了位23)都代表相同的中断源，如表74所示。

表74. IRQ/FIQ MMR位功能描述

位	描述
0	所有中断“或逻辑”(只用于FIQ)。
1	SWI。
2	定时器0。
3	定时器1。
4	唤醒定时器(定时器2)。
5	看门狗定时器(定时器3)。
6	闪存控制。
7	ADC通道。
8	锁相环锁定。
9	I2C0从机。
10	I2C0主机。
11	I2C1主机。
12	SPI从机。
13	SPI主机。
14	UART。
15	外部IRQ0。
16	比较器。
17	PSM。
18	外部IRQ1。
19	PLA IRQ0。
20	PLA IRQ1。
21	外部IRQ2。
22	外部IRQ3。
23	PWM触发(只用于IRQ)/PWM同步(只用于FIQ)。

IRQ

中断请求(IRQ)是进入处理器IRQ模式的一个异常信号。它用于内部和外部事件的一般中断服务。

与IRQ相关的4个32位寄存器是：IRQSTA、IRQSIG、IRQEN和IRQCLR。

IRQSTA 寄存器

名称	地址	默认值	访问
IRQSTA	0xFFFF0000	0x00000000	读

IRQSTA(只读寄存器)提供当前使能的IRQ源状态。当置1时，这个源应该向ARM7TDMI内核产生一个有效IRQ请求。没有任何优先编码器或中断矢量产生。这个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位经过逻辑“或”运算后向ARM7TDMI内核发送IRQ信号。

IRQSIG 寄存器

名称	地址	默认值	访问
IRQSIG	0xFFFF0004	0x00XXX000	读

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。所有的IRQ源都可以在中断使能寄存器IRQEN MMR中屏蔽。IRQSIG是只读的。

IRQEN 寄存器

名称	地址	默认值	访问
IRQEN	0xFFFF0008	0x00000000	读/写

IRQEN提供了当前使能屏蔽值。当某一位被置1时，相应的中断源就会被使能产生一个IRQ异常。当某一位被清0时，相应的中断源就会被禁止或屏蔽，此时就不能产生一个IRQ异常。

注意：如果要想取消一个已经使能的中断源，用户必须通过对IRQCLR中相应的位置1来实现。将这个中断源在IRQEN中相应的位清0不会禁用这个中断。

IRQCLR 寄存器

名称	地址	默认值	访问
IRQCLR	0xFFFF000C	0x00000000	写

IRQCLR(只写寄存器)对IRQEN寄存器清零从而屏蔽中断源。任一位置1会将IRQEN寄存器中对应的位清零，但不影响其他位。IRQEN和IRQCLR这一对寄存器在没有自动读/写请求情况下可独立控制屏蔽使能。

FIQ

快速中断请求(FIQ)是进入处理器FIQ模式的一个异常信号。它提供一个短延时的服务于数据传输或通信通道的任务。FIQ接口与IRQ接口相同,但它会提供二级中断(最高优先级)。有4个32位的寄存器专门用于FIQ:FIQSIG、FIQEN、FIQCLR和FIQSTA。

FIQSTA 寄存器

名称	地址	默认值	访问
FIQSTA	0xFFFFF0100	0x00000000	读

FIQSIG 寄存器

名称	地址	默认值	访问
FIQSIG	0xFFFFF0104	0x00XXX000	读

FIQEN 寄存器

名称	地址	默认值	访问
FIQEN	0xFFFFF0108	0x00000000	读/写

FIQCLR 寄存器

名称	地址	默认值	访问
FIQCLR	0xFFFFF010C	0x00000000	写

FIQSTA的位31到1通过“或”逻辑运算后产生FIQ信号到内核以及FIQ和IRQ寄存器的第0位(FIQ源)。

逻辑上FIQEN和IRQEN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会使IRQEN中的同一位被清0。同样,IRQEN中的某一位被置1会使FIQEN中同一位被清0。一个中断源可以被IRQEN和FIQEN屏蔽禁止。

注意若要取消一个已经使能的FIQ中断源,用户必须将FIQCLR寄存器中相应的位置1。将一个中断在FIQEN寄存器中相应的位清0不会禁用这个中断。

可编程中断

因为可编程中断是无法屏蔽的,所以它们由另外一个寄存器SWICFG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。这个32位的寄存器SWICFG专门用于软件中断(见表75)。这个存储器映射寄存器允许对可编程中断源进行控制。

SWICFG 寄存器

名称	地址	默认值	访问
SWICFG	0xFFFFF0010	0x00000000	写

表75. SWICFG MMR位功能描述

位	描述
31:3	保留。
2	可编程中断(FIQ)。对这个位置1或清0会相应的对FIQSTA和FIQSIG寄存器的位1置1或清0。
1	可编程中断(IRQ)。对这个位置1或清0会相应的对IRQSTA和IRQSIG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都要至少等于中断延迟时间,这样才能保证被中断控制器和用户在IRQSTA/FIQSTA寄存器中检测到。

定时器

ADuC7019/20/21/22/24/25/26/27/28有四个通用定时器/计数器:

- 定时器0
- 定时器1
- 定时器2或唤醒定时器
- 定时器3或看门狗定时器

这4个定时器可以工作在两种模式:自由模式和周期模式。

当工作在自由模式时,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。)

当工作在周期模式时,定时器以装载寄存器TxLD中的值为起始值开始递减或递增计数,一直到零或满量程后再以TxLD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:

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

计数器的值可随时通过访问寄存器TxVAL读出。但要注意: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。这种情况下TxVAL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。

可通过写入相应定时器的控制寄存器TxCON来启动定时器。

在普通模式下,当递减计数时,每一次计数器的值达到0时就会产生一次IRQ请求。当递增计数时,每一次计数器的值达到满刻度值时也会产生一次IRQ请求。可以通过写任意值到TxCLR来取消一个IRQ。

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。

定时器0(实时操作系统定时器)

定时器0是一个带有可编程预分频器的通用多功能16位定时器，工作时递减计数（如图64所示）。它的时钟是内核时钟(HCLK)的一个分频，分频方式共有1、1/16和1/256三种。

定时器0可用于启动ADC转换，如框图64所示。

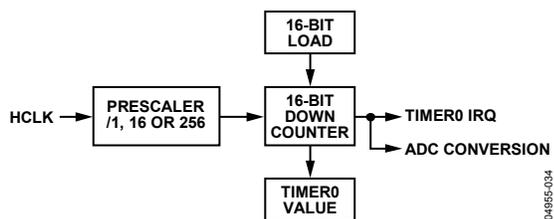


表64. 定时器0框图

定时器0接口包括4个存储器映射寄存器：TOLD、TOVAL、T0CON和T0CLRI。

TOLD 寄存器

名称	地址	默认值	访问
TOLD	0xFFFFF0300	0x0000	读/写

TOLD是一个16位装载寄存器。

TOVAL 寄存器

名称	地址	默认值	访问
TOVAL	0xFFFFF0304	0xFFFF	读

TOVAL是一个16位只读寄存器，它记录的是计数器当前的状态。

T0CON 寄存器

名称	地址	默认值	访问
T0CON	0xFFFFF0308	0x0000	读/写

T0CON是一个配置MMR，如表76所示。

Table 76. T0CON MMR位功能描述

位	值	描述
31:8		保留。
7		定时器0使能位。置1，使能定时器0。清0，禁用定时器0（默认为0）。
6		定时器0工作模式选择位。置1，选择周期模式。清0，选择自由模式。默认模式。
5:4		保留。
3:2		预分频选择位。
	00	内核时钟/1。默认值。
	01	内核时钟/16。
	10	内核时钟/256。
	11	未定义。与00时相同。
1:0		保留。

T0CLRI 寄存器

名称	地址	默认值	访问
T0CLRI	0xFFFFF030C	0xFF	写

T0CLRI是一个8位寄存器。向其中写入任意一个值就可以取消定时器0的中断请求。

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。它的时钟源可以是32 KHz的外部晶振、内核时钟频率或者是一个外部GPIO、P1.0或P0.6引脚（最高频率44MHz）。可以1、1/16、1/256或1/32768对该源时钟分频。

计数器可以是标准的32位数模式或下面的形式：时:分: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中断源初始声明所触发。这一特点可以被用来判断一个事件的声明，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65所示。

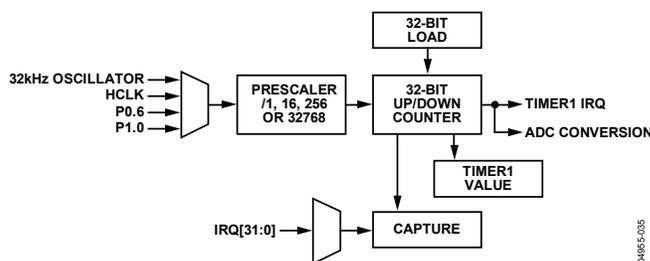


图65. 定时器1框图

定时器1接口包括5个存储器映射寄存器：T1LD、T1VAL、T1CON、T1CLRI和T1CAP。

T1LD 寄存器

名称	地址	默认值	访问
T1LD	0xFFFFF0320	0x00000000	读/写

T1LD是一个32位装载寄存器。

T1VAL 寄存器

名称	地址	默认值	访问
T1VAL	0xFFFFF0324	0xFFFFFFFF	读

T1VAL是一个32位只读寄存器，它记录的是计数器当前的状态。

T1CON 寄存器

名称	地址	默认值	访问
T1CON	0xFFFFF0328	0x0000	读/写

T1CON是一个配置MMR，如表77所示。

表77. T1CON MMR位功能描述

位	值	描述
31:18		保留。
17		事件选择位。置1, 使能定时器1捕捉一个事件。户清0, 禁用定时器1捕捉一个事件(默认为0)。
16:12		事件选择范围0至31。这些事件如表74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74中的事件2对于定时器1来说就是事件0。
11:9		时钟选择位。
	000	内核时钟(HCLK)。
	001	外部32.768 kHz晶振。
	010	P1.0上升沿触发。
	011	P0.6上升沿触发。
8		递增计数。置1, 定时器1递增计数。清0, 定时器1递减计数(默认)。
7		定时器1使能位。置1, 使能定时器1。清0, 禁用定时器1(默认)。
6		定时器1工作模式选择位。户置1, 选择周期模式。清0, 选择自由模式。默认模式。
5:4		数据格式。
	00	二进制。
	01	保留。
	10	时:分:秒:百分之一秒(23小时至0小时)。
	11	时:分:秒:百分之一秒(255小时至0小时)。
3:0		预分频选择位。
	0000	时钟源/1。
	0100	时钟源/16。
	1000	时钟源/256。
	1111	时钟源/32,768。

T1CLR1 寄存器

名称	地址	默认值	访问
T1CLR1	0xFFFF032C	0xFF	写

T1CLR1是一个8位寄存器。向其中写入任意一个值就可以清除定时器1的中断请求。

T1CAP 寄存器

名称	地址	默认值	访问
T1CAP	0xFFFF0330	0x00000000	读

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

定时器2(唤醒定时器)

定时器2是一个带有可编程预分频器的32位唤醒定时器, 工作时可递增计数或递减计数。它的时钟源可以是32KHz的外部晶振、内核时钟频率或内部32KHz的振荡器。定时器2的时钟是其所选时钟源的一个分频, 分频方式共有1、1/16、1/256、1/32768四种。当内核时钟被禁用时, 定时器2仍会继续运行。计数器可以是标准的32位数模式或下面的格式: 时:分:秒:百分之一秒。

定时器2可用于启动ADC转换, 如框图66所示。

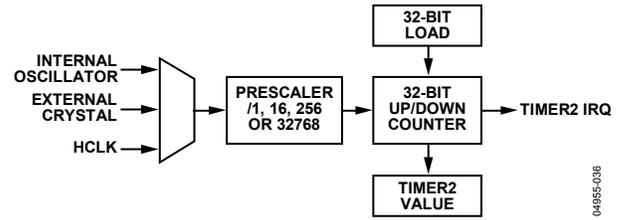


图66. 定时器2框图

定时器2接口包括4个存储器映射寄存器: T2LD、T2VAL、T2CON和T2CLR1。

T2LD 寄存器

名称	地址	默认值	访问
T2LD	0xFFFF0340	0x00000000	读/写

T2LD是一个32位装载寄存器。

T2VAL 寄存器

名称	地址	默认值	访问
T2VAL	0xFFFF0344	0xFFFFFFFF	读

T2VAL是一个32位只读寄存器, 它记录的是计数器当前的状态。

T2CON 寄存器

名称	地址	默认值	访问
T2CON	0xFFFF0348	0x0000	读/写

T2CON是一个配置MMR, 如表78所示。

表78. T2CON MMR位功能描述

位	值	描述
31:11		保留。
10:9		时钟源。
	00	外部晶振。
	01	外部晶振。
	10	内部振荡器。
	11	内核时钟(41 MHz/2 ^{CD})。
8		递增计数。置1, 定时器2递增计数。清0, 定时器2递减计数(默认)。
7		定时器2使能位。置1, 使能定时器2。清0, 禁用定时器2(默认)。
6		定时器2工作模式。置1, 选择周期模式。清0, 选择自由模式。默认模式。
5:4		数据格式。
	00	二进制。
	01	保留。
	10	时:分:秒:百分之一秒(23小时至0小时)。
	11	时:分:秒:百分之一秒(255小时至0小时)。
3:0		预分频选择位。
	0000	时钟源/1(默认)。
	0100	时钟源/16。
	1000	时钟源/256(预计用于数据格式2和3)。
	1111	时钟源/32,768。

T2CLR1 寄存器

名称	地址	默认值	访问
T2CLR1	0xFFFF034C	0xFF	写

T2CLR1是一个8位寄存器。向其中写入任意一个值就可以清除定时器2的中断请求。

定时器3(看门狗定时器)

定时器3有两种工作模式：普通模式和看门狗模式。看门狗定时器用于使处理器在进入非法软件状态后的恢复。一旦看门狗定时器被使能，它需要周期服务来阻止它强迫处理器复位。

普通模式

在普通模式下，除了时钟源和递增计数功能，定时器3和定时器0的功能相同。时钟源来自于锁相环（32KHz），其时钟分频方式共有1、1/16、1/256三种（如图67所示）。

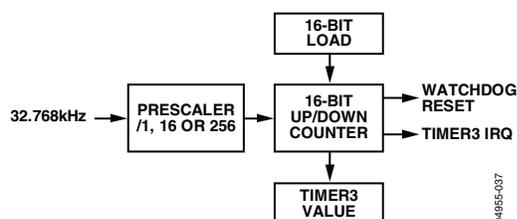


图67. 定时器3框图

看门狗模式

将T3CON MMR的第5位置1便可以进入看门狗模式。定时器3以T3LD寄存器中的数为起始值开始递减计数，一直到0为止，T3LD作为超时定时器。当使用1/256预分频时，最大的超时时间为512秒，且为T3LD的满量程。当工作在看门狗模式下时，定时器3的时钟源为内部的32KHz晶振。注意：为了成功进入看门狗模式，必须在写入T3LD MMR以后再对T3CON MMR的第5位置1。

如果定时器计数到0，就会产生一个复位或中断，这取决于T3CON寄存器的第1位的配置。如果不想产生复位或中断，必须在定时周期结束之前向T3CLR1中写入任意一个值。T3LD的计数器重载以后会开始一个新的超时周期。

一旦进入看门狗模式，T3LD和T3CON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3退出看门狗模式。

定时器3接口包括4个存储器映像寄存器：T3LD、T3VAL、T3CON和T3CLR1。

T3LD 寄存器

名称	地址	默认值	访问
T3LD	0xFFFF0360	0x0000	读/写

T3LD是一个16位装载寄存器。

T3VAL 寄存器

名称	地址	默认值	访问
T3VAL	0xFFFF0364	0xFFFF	读

T3VAL是一个16位只读寄存器，它记录的是计数器当前的状态。

T3CON 寄存器

名称	地址	默认值	访问
T3CON	0xFFFF0368	0x0000	读/写

T3CON是一个配置MMR，如表79所示。

表79. T3CON MMR位功能描述

位	值	描述
31:9		保留。
8		递增计数。置1，定时器3递增计数。清0，定时器3递减计数（默认）。
7		定时器3使能位。置1，使能定时器3。清0，禁用定时器3（默认）。
6		定时器3工作模式。置1，选择周期模式。清0，选择自由模式。默认模式。
5		看门狗模式使能位。置1，使能看门狗模式。清0，禁用看门狗模式（默认）。
4		安全清除位。置1，使能安全清除。该位清0，禁用安全清除（默认）。
3:2		预分频选择位。
	00	时钟源/1（默认）。
	01	时钟源/16。
	10	时钟源/256。
	11	未定义。与00时相同。
1		看门狗IRQ选项位。置1，当看门狗定时器递减到0时产生一个IRQ而不是复位。用户清0，禁用IRQ选项。
0		保留。

T3CLR1 寄存器

名称	地址	默认值	访问
T3CLR1	0xFFFF036C	0x00	写

T3CLR1是一个8位寄存器。普通模式下，在连续工作时向这个寄存器写入任意一个值会清除定时器3的中断请求；在看门狗模式下，向这个寄存器写入任意一个值就会重新开始一个超时周期。

注意一定要确保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)

安全清除位用于更高层次的保护。当它被置1时，一个特殊的数值序列就必须写入T3CLRI中来避免看门狗复位。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68所示。

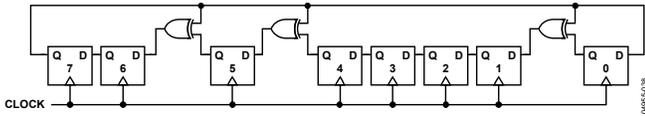


图68. 8位LFSR

在进入看门狗模式之前必须先向T3CLRI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T3CLRI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就会进入下一个状态。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，它必须在软件中产生和跟踪。

一个数值序列的示例如下所示：

1. 设定定时器3为看门狗模式之前，在T3CLRI中写入初始种子0xAA。
2. 在T3CLRI中写入0xAA，定时器3被重载。
3. 在T3CLRI中写入0x37，定时器3被重载。
4. 在T3CLRI中写入0x6E，定时器3被重载。
5. 写入0x66。0xDC是期望值，看门狗将芯片复位。

外部存储器接口

ADuC7026和ADuC7027是这一系列芯片中唯一拥有外部存储器接口的两个型号。外部存储器接口需要大量的引脚。所以这种接口只能存在于引脚数多的封装形式的芯片上。当使用外部端口时，XMCFG存储器映射寄存器必须被置1。

尽管内部支持32位的地址，但外部引脚上只有低16位地址。

存储器接口可以寻址多达4个128 kB的异步存储器(SRAM或/和EEPROM)。

一个外部存储器接口所需的引脚如表80所示。

表80. 外部存储器接口引脚

引脚	功能
AD[15:0]	地址/数据总线。
A16	仅用于8位存储器的扩展地址。
MS[3:0]	存储器选择。
\overline{WS}	写选通。
\overline{RS}	读选通。
AE	地址锁存使能。
\overline{BHE} , \overline{BLE}	字节写入容量。

共有4个可用的外部存储区域，如表81所示。与每一个区域相关的引脚是MS[3:0]。这些信号允许访问外部存储器的特定区域。每一个存储区域最大为128kB, $64\text{ k} \times 16$ 或者 $128\text{ k} \times 8$ 。如果访问一个128 k的8位存储器，需要一条扩展地址线(A16)。(参见图69所示示例)。这4个区域可单独配置。

表81. 存储区域

起始地址	结束地址	目录
0x10000000	0x1000FFFF	外部存储区域0
0x20000000	0x2000FFFF	外部存储区域1
0x30000000	0x3000FFFF	外部存储区域2
0x40000000	0x4000FFFF	外部存储区域3

每一个外部存储区域可通过3个存储器映射寄存器控制：XMCFG、XMxCON和XMxPAR。

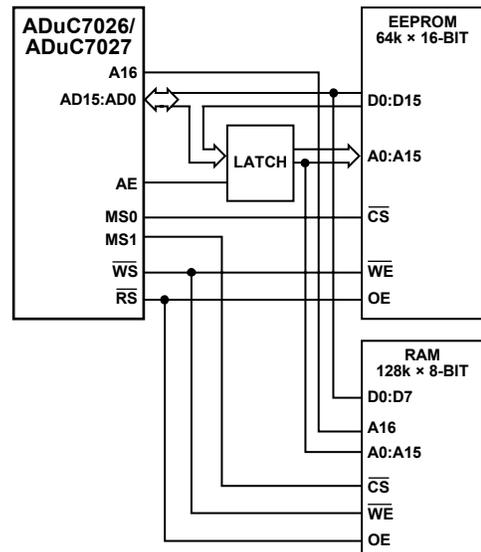


图69. 外部EEPROM/RAM接口

XMCFG 寄存器

名称	地址	默认值	访问
XMCFG	0xFFFFF000	0x00	读/写

将XMCFG寄存器置1可以使能访问外部存储器。必须在将任何端口引脚配置为外部存储器访问引脚之前对这个寄存器置1。端口引脚也必须通过GPxCON存储器映射寄存器单独使能。

XMxCON 寄存器

名称	地址	默认值	访问
XM0CON	0xFFFFF010	0x00	读/写
XM1CON	0xFFFFF014	0x00	读/写
XM2CON	0xFFFFF018	0x00	读/写
XM3CON	0xFFFFF01C	0x00	读/写

XMxCON是每一个存储区域的控制寄存器。它们用于允许使能/禁止一个存储区域并且控制存储区域的数据总线宽度。

表82. XMxCON MMR位功能描述

位	描述
1	选择数据总线宽度。置1, 选择16位数据总线。清0, 选择8位数据总线。
0	使能存储区域。置1, 使能存储区域。清0, 禁用存储区域。

XMxPAR 寄存器

名称	地址	默认值	访问
XM0PAR	0xFFFFF020	0x70FF	读/写
XM1PAR	0xFFFFF024	0x70FF	读/写
XM2PAR	0xFFFFF028	0x70FF	读/写
XM3PAR	0xFFFFF02C	0x70FF	读/写

XMxPAR存储器用来定义访问外部存储器的协议。

表83. XMxPAR MMR位功能描述

位	描述
15	使能字节写选通。这个位仅用于两个8位存储器共享同一个存储区域时。置1, 使A0作为/ \overline{WS} 信号输出。这样在不使用/ \overline{BHE} 和/ \overline{BLE} 信号时也允许字节写入能力。用户清零使用/ \overline{BHE} 和/ \overline{BLE} 信号。
14:12	地址锁存使能选通的等待状态个数。
11	保留。
10	额外地址保持时间。置1, 禁用额外保持时间。清0, 在读和写时使能1个周期的地址保持时间。
9	读数据时额外总线转换时间。置1, 禁用额外总线转换时间。清0, 在读选通(\overline{RS})之前和之后使能1个额外时钟。
8	写数据时额外总线转换时间。置1, 禁用额外总线转换时间。清0, 在写选通(\overline{WS})之前和之后使能1个额外时钟。
7:4	写等待状态个数。选择添加到/ \overline{WS} 脉冲长度的等待状态的个数。0x0: 1个时钟; 0xF: 16个时钟周期(默认值)。
3:0	读等待状态个数。选择添加到/ \overline{RS} 脉冲长度的等待状态的个数。0x0: 1个时钟; 0xF: 16个时钟周期(默认值)。

图70、图71、图72和图73分别给出了1个读周期时序图、1个带有地址保持和总线转换周期的读周期时序图、1个带有地址和写保持周期的写周期时序图和1个带有等待状态的写周期时序图。

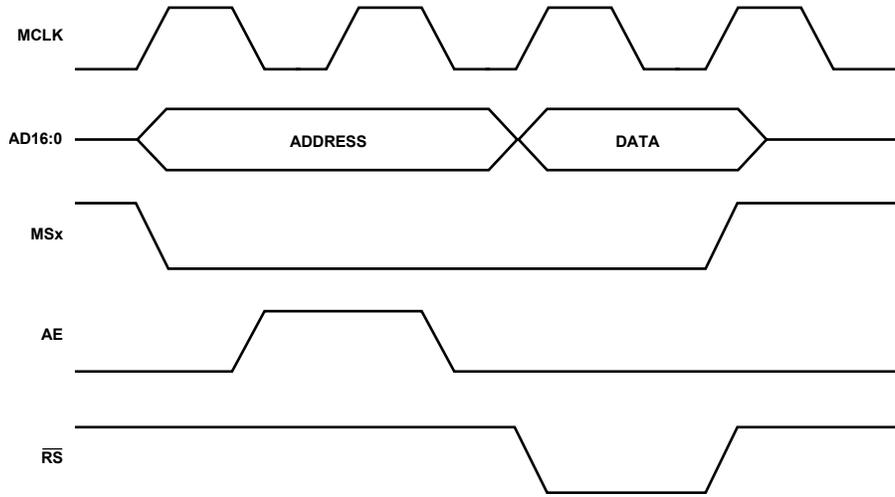


图70. 外部存储器读周期

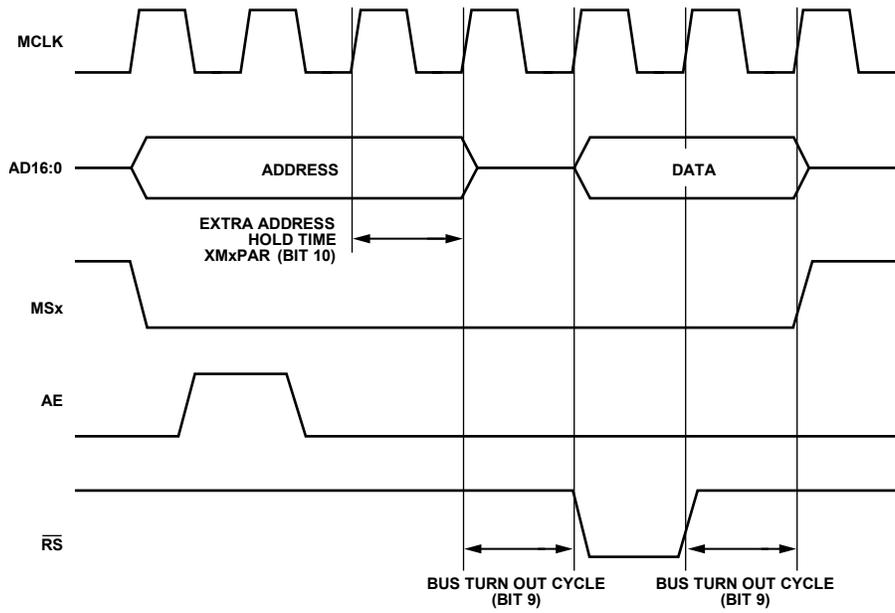


图71. 带有地址保持和总线转换周期的外部存储器读周期

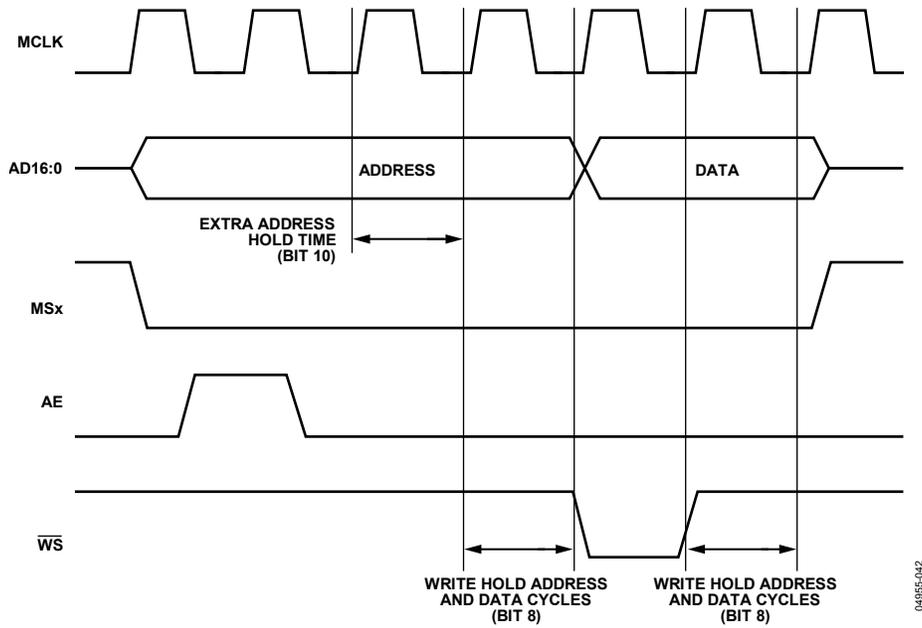


图72. 带有地址和写保持周期的外部存储器写周期

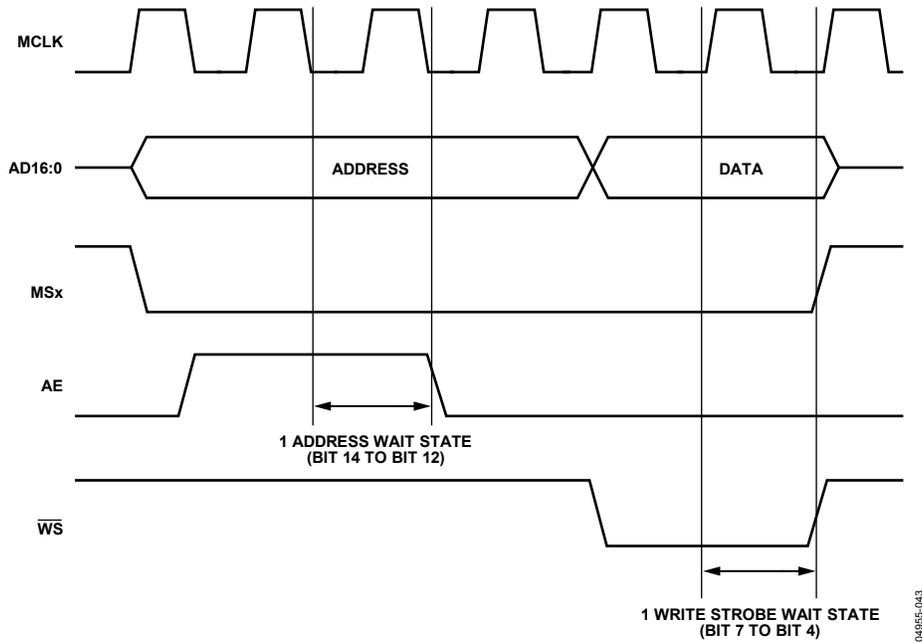


图73. 带有等待状态的外部存储器写周期

硬件设计考虑

电源

ADuC7019/20/21/22/24/25/26/27/28工作电压范围为2.7 V至3.6 V。分离的模拟和数字电源引脚(分别为AV_{DD}和IOV_{DD})使得AV_{DD}不受IOV_{DD}上数字信号干扰的影响。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,系统可以设计为IOV_{DD}工作电压为3.3 V,而AV_{DD}电压为3 V,反之亦然。一个典型的分离电源设计如图74所示。

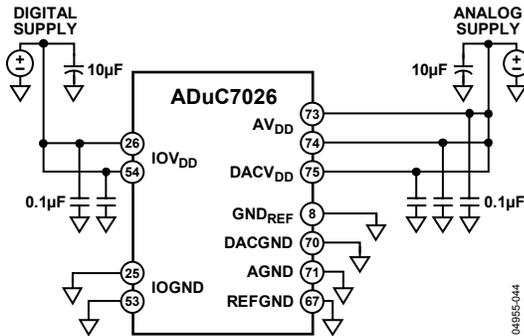


图74. 外部双电源连接

除了使用两个分离的电源以外还有一个替代的方法,用户可以通过在AV_{DD}和IOV_{DD}之间串联一个小电阻和/或磁珠来降低AV_{DD}上的噪声,然后将AV_{DD}对地单独去耦。图75所示就是用这种方法进行设计的一个示例。使用这种方法,其它模拟电路(诸如:运算放大器、基准电压源以及其它模拟电路)也可以通过AV_{DD}供电。

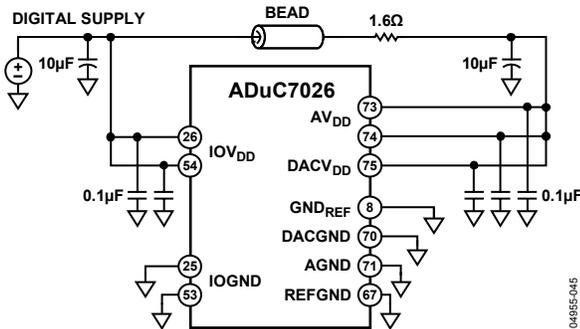


图75. 外部单电源连接

注意在图74和图75中,在IOV_{DD}处有一个大容值(10μF)的储能电容,以及在AV_{DD}处单独有一个10μF的电容。此外,在芯片的每一个AV_{DD}和IOV_{DD}引脚处都连接有一个小容值(0.1μF)的电容。按照实际设计时的标准,必须确保包括所有这些电容并且更小的电容应该尽可能地接近每一个AV_{DD}引脚,布线长度也应尽量越短越好。这些电容连接“地”的一端直接连接到地平面即可。

最后,应注意在任何时候ADuC7019/20/21/22/24/25/26/27/28的模拟地和数字地引脚必须参考同一个系统地参考点。

IOV_{DD}电源敏感度

IOV_{DD}电源对于高频噪声是很敏感的,因为片内振荡器和锁相环电路也是由IOV_{DD}供电。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。

典型情况下,电源上频率高于50 kHz并且峰峰值为50mV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50mV,那么就需要一个图76所推荐的滤波电路。

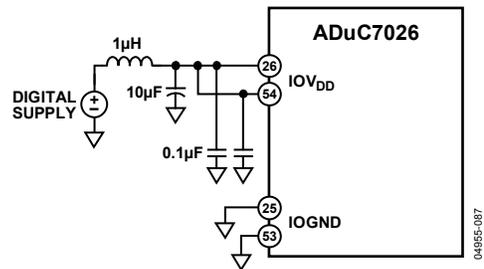


图76. 推荐的IOV_{DD}电源滤波电路

线性稳压器

各ADuC7019/20/21/22/24/25/26/27/28都需要一个3.3 V单电源,但是内核逻辑需要一个2.6 V的电源。片内有一个线性稳压器可以将来自IOV_{DD}的电源稳压到2.6 V为内核逻辑供电。LV_{DD}引脚的2.6 V电源就用来给内核逻辑供电。在LV_{DD}和DGND之间必须连接一个0.47μF的补偿电容(应尽量靠近这些引脚)作为电荷槽,如图77所示。

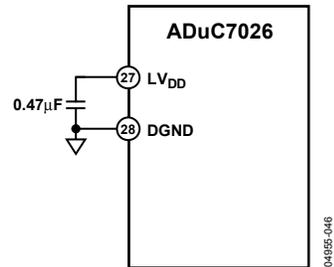


图77. 稳压器连接

LV_{DD}引脚不能用来给任何其它芯片供电。同样推荐在IOV_{DD}引脚使用良好的电源去耦装置以便提高片内稳压器的线性稳压性能。

接地和电路板布局

建议

对于所有的高分辨率数据转换器，为了从ADC和DAC获得最优的性能应特别注意基于ADuC7019/20/21/22/24/25/26/27/28的接地和PCB布局。

尽管这些器件已经将模拟地和数字地引脚分开(AGND和IOGND)，使用者一定不能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78a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(例如：在系统电源)，那么这个平面不能再连接到靠近芯片的地方，因为这样会导致一个地环路。在这些例子中，ADuC7019/20/21/22/24/25/26/27/28的所有AGND和IOGND引脚都连接到模拟地平面，如图78b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分属两个半平面，这样数字回路电流不会流经模拟电路附近(反之亦然)。

ADuC7019/20/21/22/24/25/26/27/28可以放置在数字和模拟部分之间，如图78c所示。

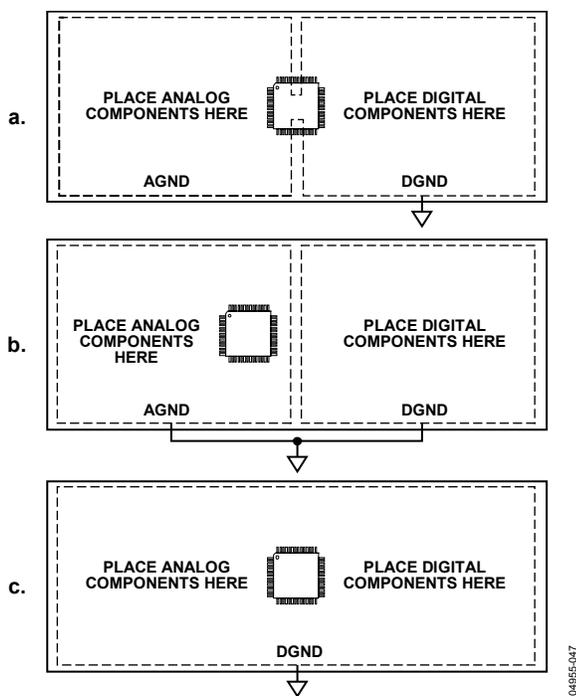


图78. 系统地框图

在所有这些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要尽量靠近电流所要到达的终点的路径。

例如，不要用IOV_{DD}给模拟部分的器件供电(如图78b所示)，因为这会导致IOV_{DD}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78c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的部分，例如：那些需要通过很长路径连接在一起的器件在同一层上，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，尽量少一些或不要有支路把引脚将其过孔与地分离。

当ADuC7019/20/21/22/24/25/26/27/28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5ns)时，应该在每一条相关的线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5ns。通常，阻值为100Ω或200Ω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

ADuC7019/20/21/22/24/25/26/27/28的时钟源可以由内部锁相环或者一个外部时钟输入产生。当使用内部锁相环时，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79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41.78MHz±3%的典型频率。

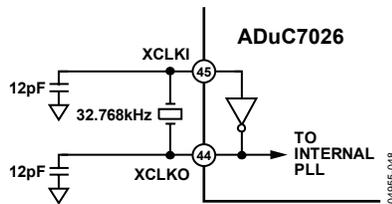


图79. 外部并行谐振晶体连接

如果使用一个外部时钟源输入代替锁相环(如图80所示)，PLLCON寄存器的位1和位0都需要修改，外部时钟从P0.7和XCLK引脚输入。

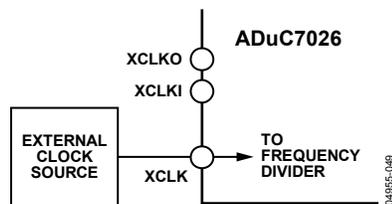


图80. 连接一个外部时钟源

当使用外部时钟源时，ADuC7019/20/21/22/24/25/26/27/28的额定时钟频率范围为50 kHz到44 MHz ±1%，这可以确保模拟外设和Flash/EE正常工作。

ADuC7019/20/21/22/24/25/26/27/28/29

上电复位操作

ADuC7019/20/21/22/24/25/26/27/28 有一个内部上电复位 (POR) 电路, 典型情况下当 LV_{DD} 低于 2.35 V 时, 内部POR会保持器件处于复位状态。当 LV_{DD} 上升超过 2.35 V 时, 一个内部定时器会在 128 ms 后溢出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。同样地, 在 LV_{DD} 下降到 2.35V 以下之前, 内部POR会保持器件处于复位状态。图81为内部POR工作的详细图解。

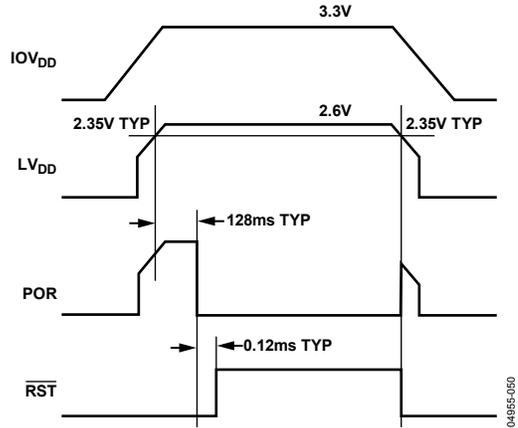


图81. 内部上电复位操作

典型系统配置

一个典型的ADuC7020配置如图82所示, 其中概括了一些前面部分探讨的硬件设计时应该考虑的地方。出于机械原因, CSP封装的器件其底层有一个裸露的焊盘需要焊接在电路板的金属片上。电路板上的金属片可以连接到地。

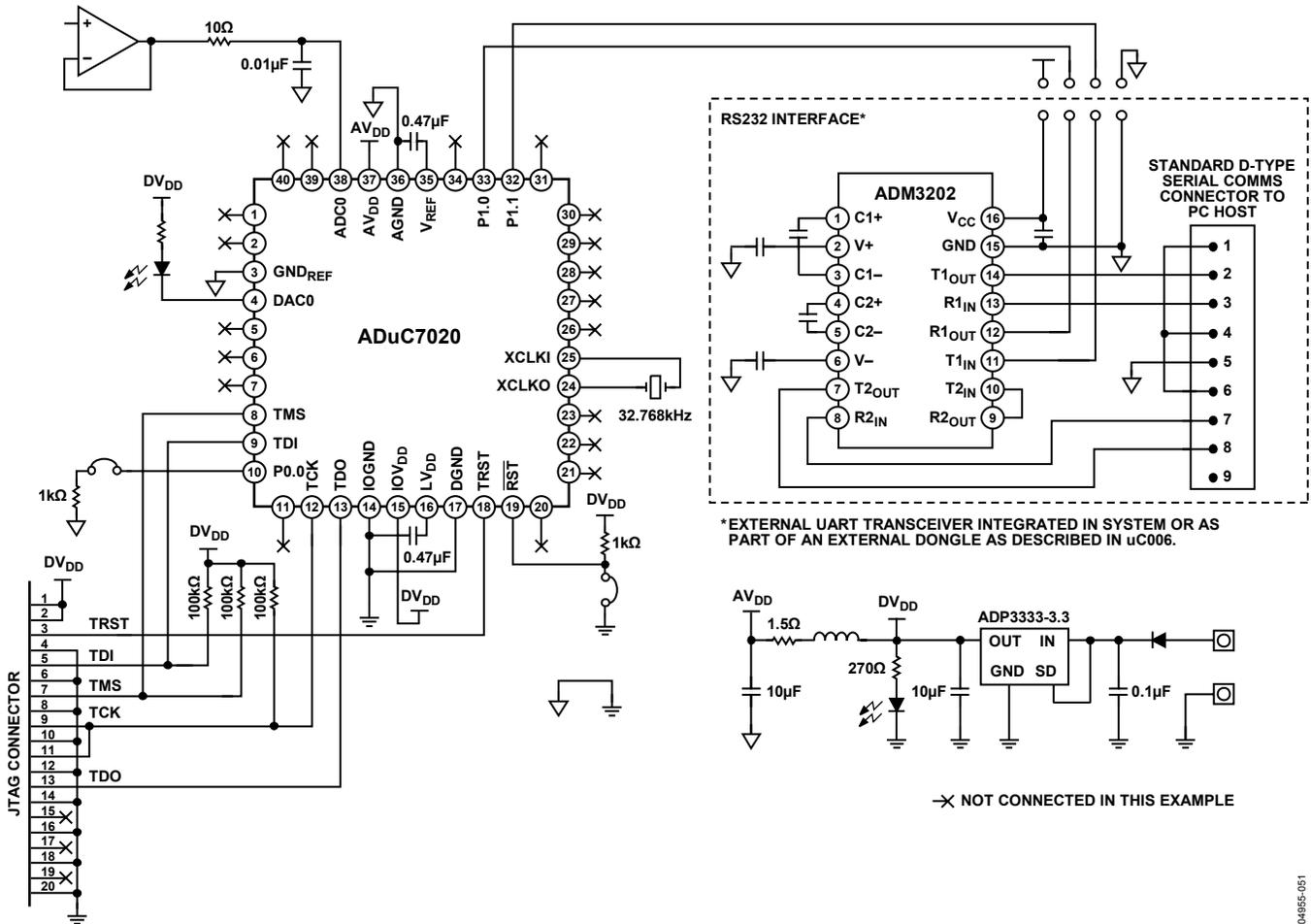


图82. 典型系统配置

开发工具

基于PC工具

ADuC7019/20/21/22/24/25/26/27/28系列有四种开发系统：

- ADuC7026 QuickStart Plus就是为那些想拥有一个全面硬件开发环境的新用户准备的。因为ADuC7026包括的功能是ADuC7019/20/21/22/24/25/26/27/28系列中最全的，它适合于那些希望在这一系列器件中的任何一个上进行开发的用户。所有的器件都是代码兼容的。
- ADuC7020、ADuC7024和ADuC7026 QuickStart适合于那些已经拥有一个仿真器的用户。

这些系统由以下基于个人电脑(PC)的(Windows®兼容)硬件和软件开发工具组成：

硬件

- ADuC7019/20/21/22/24/25/26/27/28评估板
- 串口编程电缆线
- RDI兼容的JTAG仿真器(仅包括在ADuC7026 QuickStart Plus中)

软件

- 集成开发环境，包括汇编器、编译器和基于JTAG的非介入式调试器

- 串行下载软件
- 示例代码

杂项条款

- CD-ROM文档

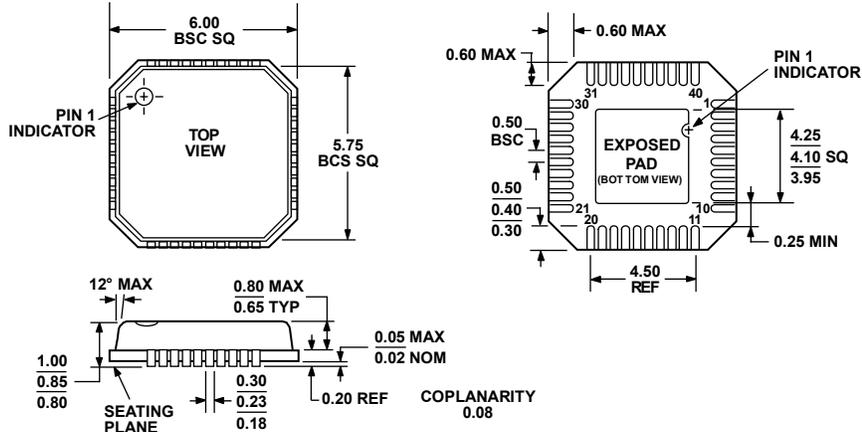
在线串行下载器

该串行下载器是一个Windows应用程序，用户可通过标准个人电脑(PC)的串口将已汇编程序串行下载到片内编程Flash/EE存储器。

所有开发系统都包含了基于UART的串行下载器，适用于订购指南中不含“1”后缀的ADuC7019/20/21/22/24/25/26/27/28器件。

WWW.ANALOG.COM还提供一种基于I²C的串行下载器。这款软件需要使用一个USB-I²C适配板（可从HOCHSCHULE PFORZHEIM获得）。该基于I²C的串行下载器只适用于订购指南中含有“1”后缀的器件。

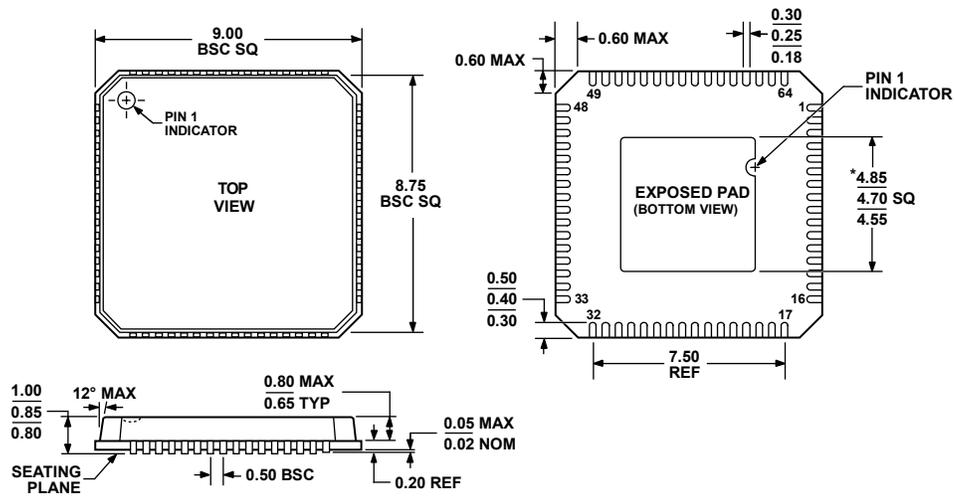
外形尺寸



COMPLIANT TO JEDEC STANDARDS MO-220-VJJD-2

图83. 40引脚LFCSP_VQ封装, 6 mm x 6 mm, 超薄体(CP-40-1) 尺寸单位: mm

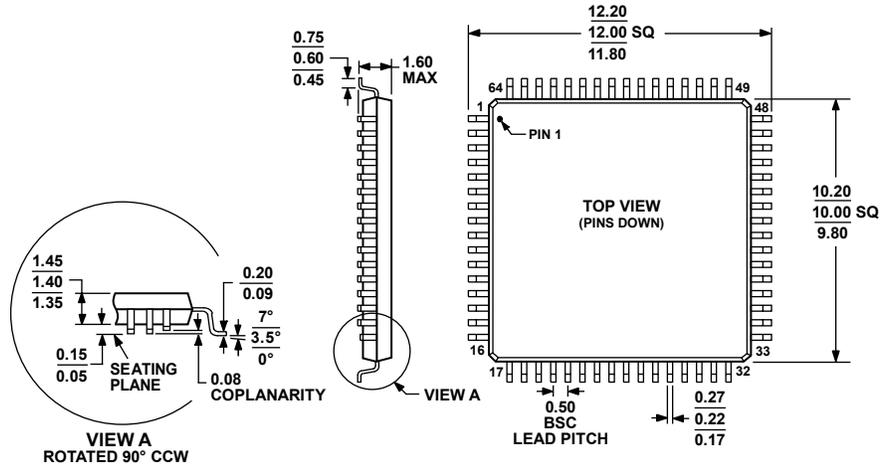
101306-A



*COMPLIANT TO JEDEC STANDARDS MO-220-VMMD-4 EXCEPT FOR EXPOSED PAD DIMENSION

图84. 64引脚LFCSP_VQ封装, 9 mm x 9 mm, 超薄体(CP-64-1) 尺寸单位: mm

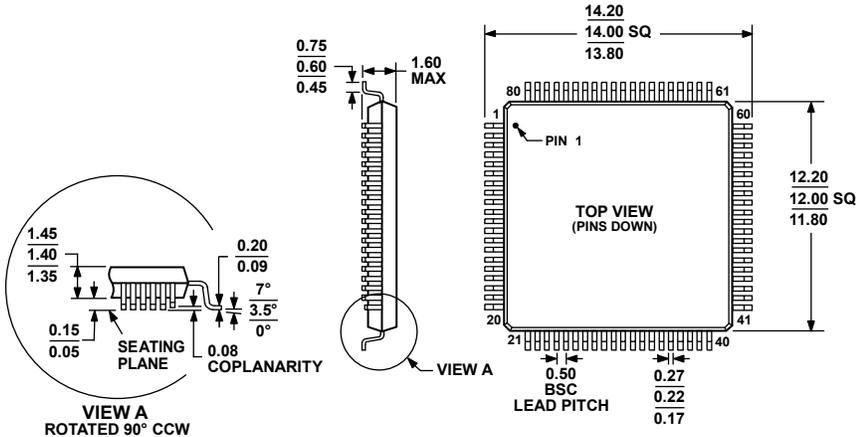
063006-B



COMPLIANT TO JEDEC STANDARDS MS-026-BCD

051706-A

图85. 64引脚LQFP封装(ST-64-2) 尺寸单位: mm

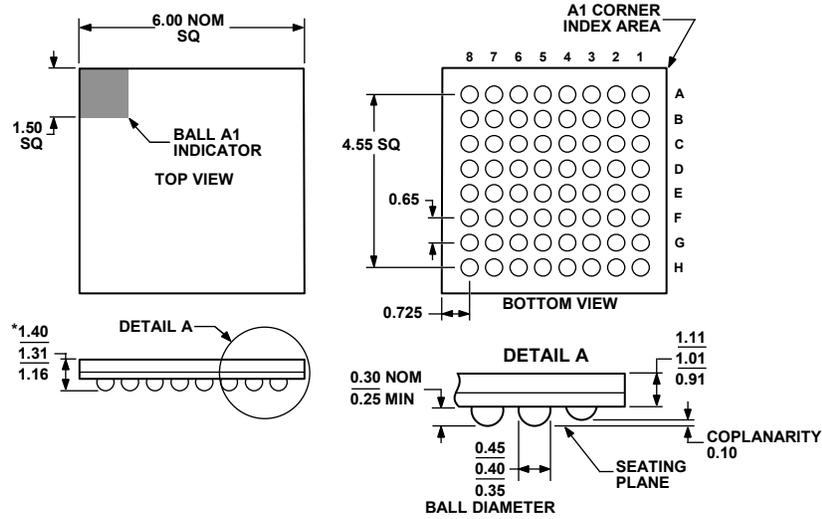


COMPLIANT TO JEDEC STANDARDS MS-026-BDD

051706-A

图86. 80引脚LQFP封装(ST-80-1) 尺寸单位: mm

ADuC7019/20/21/22/24/25/26/27/28/29



*COMPLIANT TO JEDEC STANDARDS MO-225
WITH THE EXCEPTION TO PACKAGE HEIGHT.

09086-A

图 87. 64引脚CSP_BGA封装, (BC-64-4) 尺寸单位: mm



49-Lead Chip Scale Package Ball Grid Array [CSP_BGA] (BC-49-1) Dimensions shown in millimeters

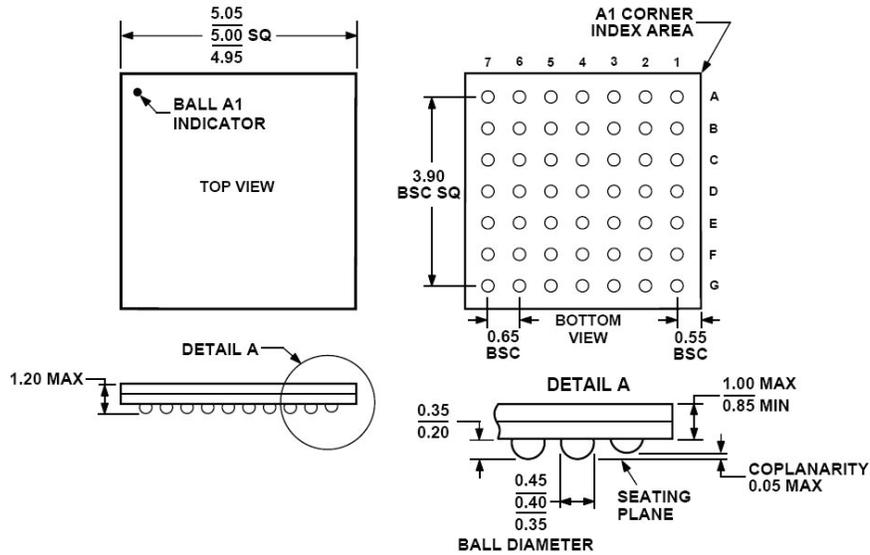


图 49. 64引脚CSP_BGA封装, (BC-49-1) 尺寸单位: mm

ORDERING GUIDE

型号	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
ADuC7019BCPZ621 ¹	5 ²	3	62 kB/8 kB	14	I ² C	-40°C至+125°C	40-引脚	CP-40-1	

ADuC7019/20/21/22/24/25/26/27/28/29

型号	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
ADuC7019BCPZ62I-RL ¹	5 ²	3	62 kB/8 kB	14	I ² C	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	2,500
ADuC7019BCPZ62IRL7 ¹	5 ²	3	62 kB/8 kB	14	I ² C	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	750
ADuC7020BCPZ62 ¹	5	4	62 kB/8 kB	14	UART	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	
ADuC7020BCPZ62-RL ¹	5	4	62 kB/8 kB	14	UART	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	2,500
ADuC7020BCPZ62-RL7 ¹	5	4	62 kB/8 kB	14	UART	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	750
ADuC7020BCPZ62I ¹	5	4	62 kB/8 kB	14	I ² C	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	
ADuC7020BCPZ62I-RL ¹	5	4	62 kB/8 kB	14	I ² C	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	2,500
ADuC7020BCPZ62IRL7 ¹	5	4	62 kB/8 kB	14	I ² C	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	750
ADuC7021BCPZ62 ¹	8	2	62 kB/8 kB	13	UART	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	
ADuC7021BCPZ62-RL ¹	8	2	62 kB/8 kB	13	UART	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	2,500
ADuC7021BCPZ62-RL7 ¹	8	2	62 kB/8 kB	13	UART	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	750
ADuC7021BCPZ62I ¹	8	2	62 kB/8 kB	13	I ² C	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	
ADuC7021BCPZ62I-RL ¹	8	2	62 kB/8 kB	13	I ² C	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	2,500
ADuC7021BCPZ62IRL7 ¹	8	2	62 kB/8 kB	13	I ² C	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	750
ADuC7021BCPZ32 ¹	8	2	32 kB/4 kB	13	UART	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	
ADuC7021BCPZ32-RL ¹	8	2	32 kB/4 kB	13	UART	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	2,500
ADuC7021BCPZ32-RL7 ¹	8	2	32 kB/4 kB	13	UART	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	750
ADuC7022BCPZ62 ¹	10		62 kB/8 kB	13	UART	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	
ADuC7022BCPZ62-RL ¹	10		62 kB/8 kB	13	UART	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	2,500
ADuC7022BCPZ62-RL7 ¹	10		62 kB/8 kB	13	UART	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	750
ADuC7022BCPZ32 ¹	10		32 kB/4 kB	13	UART	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	
ADuC7022BCPZ32-RL ¹	10		32 kB/4 kB	13	UART	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	2,500
ADuC7022BCPZ32-RL7 ¹	10		32 kB/4 kB	13	UART	-40°C至+125°C	40引脚LFCSP_VQ	CP-40-1	750
ADuC7024BCPZ62 ¹	10	2	62 kB/8 kB	30	UART	-40°C至+125°C	64引脚LFCSP_VQ	CP-64-1	
ADuC7024BCPZ62-RL ¹	10	2	62 kB/8 kB	30	UART	-40°C至+125°C	64引脚LFCSP_VQ	CP-64-1	2,500
ADuC7024BCPZ62-RL7 ¹	10	2	62 kB/8 kB	30	UART	-40°C至+125°C	64引脚LFCSP_VQ	CP-64-1	750
ADuC7024BSTZ62 ¹	10	2	62 kB/8 kB	30	UART	-40°C至+125°C	64引脚LQFP	ST-64-2	
ADuC7024BSTZ62-RL ¹	10	2	62 kB/8 kB	30	UART	-40°C至+125°C	64引脚LQFP	ST-64-2	1,000
ADuC7025BCPZ62 ¹	12		62 kB/8 kB	30	UART	-40°C至+125°C	64引脚LFCSP_VQ	CP-64-1	
ADuC7025BCPZ62-RL ¹	12		62 kB/8 kB	30	UART	-40°C至+125°C	64引脚LFCSP_VQ	CP-64-1	2,500
ADuC7025BCPZ62-RL7 ¹	12		62 kB/8 kB	30	UART	-40°C至+125°C	64引脚LFCSP_VQ	CP-64-1	750
ADuC7025BCPZ32 ¹	12		32 kB/4 kB	30	UART	-40°C至+125°C	64引脚LFCSP_VQ	CP-64-1	
ADuC7025BCPZ32-RL ¹	12		32 kB/4 kB	30	UART	-40°C至+125°C	64引脚LFCSP_VQ	CP-64-1	2,500
ADuC7025BCPZ32-RL7 ¹	12		32 kB/4 kB	30	UART	-40°C至+125°C	64引脚LFCSP_VQ	CP-64-1	750
ADuC7025BSTZ62 ¹	12		62 kB/8 kB	30	UART	-40°C至+125°C	64引脚LQFP	ST-64-2	
ADuC7025BSTZ62-RL ¹	12		62 kB/8 kB	30	UART	-40°C至+125°C	64引脚LQFP	ST-64-2	1,000
ADuC7026BSTZ62 ^{1,3}	12	4	62 kB/8 kB	40	UART	-40°C至+125°C	80引脚LQFP	ST-80-1	
ADuC7026BSTZ62-RL ^{1,3}	12	4	62 kB/8 kB	40	UART	-40°C至+125°C	80引脚LQFP	ST-80-1	1,000
ADuC7026BSTZ62I ^{1,3}	12	4	62 kB/8 kB	40	I ² C	-40°C至+125°C	80引脚LQFP	ST-80-1	
ADuC7026BSTZ62I-RL ^{1,3}	12	4	62 kB/8 kB	40	I ² C	-40°C至+125°C	80引脚LQFP	ST-80-1	1,000

ADuC7019/20/21/22/24/25/26/27/28/29

型号	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
ADuC7027BSTZ62 ^{1,3}	16		62 kB/8 kB	40	UART	-40°C至+125°C	80引脚LQFP	ST-80-1	
ADuC7027BSTZ62-RL ^{1,3}	16		62 kB/8 kB	40	UART	-40°C至+125°C	80引脚LQFP	ST-80-1	1,000
ADuC7027BSTZ62I ^{1,3}	16		62 kB/8 kB	40	UART	-40°C至+125°C	80引脚LQFP	ST-80-1	
ADuC7027BSTZ62I-RL ^{1,3}	16		62 kB/8 kB	40	UART	-40°C至+125°C	80引脚LQFP	ST-80-1	2500
ADuC7028BBCZ62 ¹	8	4	62 kB/8 kB	30	UART	-40°C至+125°C	64引脚 CSP_BGA	BC-64-4	
ADuC7028BBCZ62-RL ¹	8	4	62 kB/8 kB	30	UART	-40°C至+125°C	64引脚 CSP_BGA	BC-64-4	
ADuC7029BBCZ62 ¹	7	4	62 kB/8 kB	22	UART	-40°C至+125°C	49引脚CSP_BGA	BC-49-1	
ADuC7028BBCZ62-RL ¹	7	4	62 kB/8 kB	22	UART	-40°C至+125°C	49引脚 CSP_BGA	BC-49-1	
EVAL-ADuC7020MKZ ¹							ADuC7020迷你套件		
EVAL-ADuC7020QSZ ¹							ADuC7020 QuickStart 开发系统		
EVAL-ADuC7024QSZ ¹							ADuC7024 QuickStart开发系统		
EVAL-ADuC7026QSZ ¹							ADuC7026 QuickStart开发系统		
EVAL-ADuC7026QSPZ ¹							ADuC7026 QuickStart开发系统		
EVAL-ADuC7028QSZ ¹							ADuC7028 QuickStart开发系统		

¹ Z = RoHS兼容器件。

² 其中一个ADC通道是内部缓冲的。

³ 包括外部存储器接口。

如果系统符合Philips公司定义的I²C标准规范，则用户在购买ADI公司或其下属机构拥有Philips公司许可的I²C器件时，可以获得Philips公司I²C专利权之下的许可，以便在I²C系统中使用这些器件。