

特性

高精度ADC

双通道、同步采样、16位 Σ - Δ ADC
可编程ADC吞吐量: 1Hz至8kHz
片内 ± 5 ppm/ $^{\circ}$ C基准电压源

电流通道

全差分、缓冲输入
可编程增益: 1至512
ADC输入范围: -200 mV至 $+300$ mV
数字比较器, 内置电流累加器功能

电压通道

缓冲、片内衰减器, 适用于12V电池输入

温度通道

外部和片内温度传感器方案

微控制器

16位/32位RISC架构ARM7TDMI内核
20.48MHz PLL, 内置可编程分频器
PLL输入源
片内精密振荡器
片内低功耗振荡器
外部 (32.768 kHz) 时钟晶体
JTAG端口支持代码下载和调试

内存

96 kB Flash/EE存储器, 6 kB SRAM
Flash/EE耐久性: 10,000个周期, 数据保持时间: 20年
通过JTAG和LIN在线下载

片内外设

利用硬件同步通过UART兼容SAEJ2602/LIN 1.3/LIN 2.0 (从机)

灵活的唤醒I/O引脚, 主/从SPI串行I/O

9引脚GPIO端口, 3个通用定时器

唤醒和看门狗定时器

电源监控器和片内上电复位

电源

12V电池电源直接供电

功耗

正常模式: 10mA (10MHz)

低功耗监控模式

封装和温度范围

7 mm \times 7 mm, 48引脚LFCSP

工作温度范围: -40° C至 $+115^{\circ}$ C完全性能指标保证

应用

汽车系统电池检测/管理

功能框图

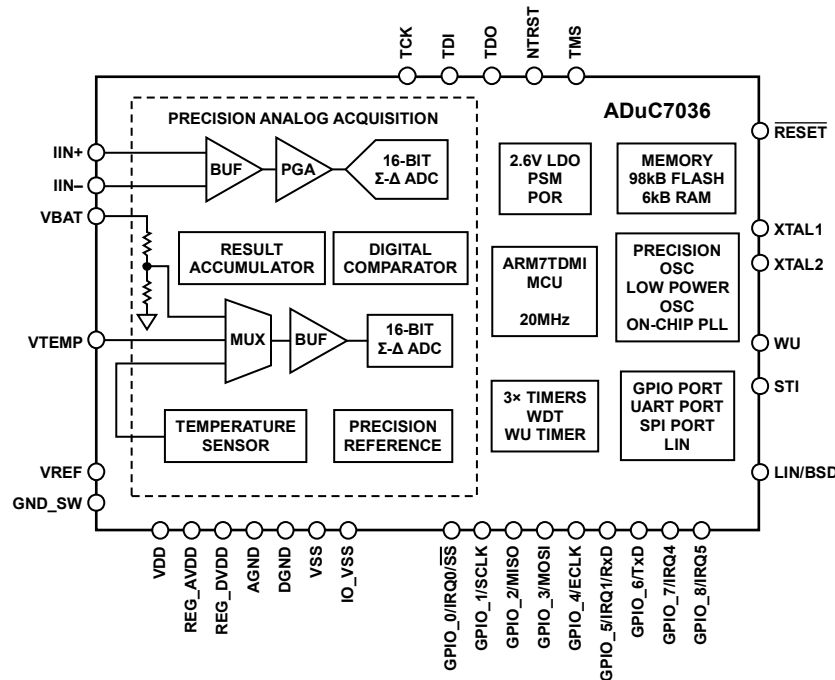


图1

Rev.

A 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 ©2008-2009 Analog Devices, Inc. All rights reserved.

ADI 中文版数据手册是英文版数据手册的译文, 敬请谅解翻译中可能存在的语言组织或翻译错误, ADI 不对翻译中存在的差异或由此产生的错误负责。如需确认任何词语的准确性, 请参考 ADI 提供的最新英文版数据手册。

目录

特性	1	中断系统	67
应用	1	定时器	70
功能框图	1	定时器 0—使用寿命定时器	71
修订历史	3	定时器 1—通用定时器	73
技术规格	4	定时器 2—唤醒定时器	75
电气规格	4	定时器 3—看门狗定时器	77
时序规格	10	定时器 4—STI定时器	79
绝对最大额定值	15	通用I/O	81
ESD警告	15	高压外设控制接口	92
引脚配置和功能描述	16	唤醒 (WU) 引脚	99
典型工作特性	18	高压外设控制接口中断处理	100
术语	19	低压标志 (LVF)	100
工作原理	20	高压诊断	100
ARM7TDMI内核概览	20	UART串行接口	101
存储器结构	22	波特率生成	101
复位	24	UART寄存器定义	102
FLASH/EE存储器	25	串行外设接口	107
Flash/EE存储器在线编程	25	MISO引脚	107
Flash/EE控制接口	25	MOSI引脚	107
FLASH/EE存储器安全性	29	SCLK引脚	107
FLASH/EE存储器可靠性	31	SS \bar 引脚	107
SRAM和Flash/EE代码执行时间	31	SPI寄存器定义	107
片上内核	32	串行测试接口	110
存储器映射寄存器	34	LIN (局域互连网络) 接口	113
完整MMR列表	35	LIN MMR描述	113
16 位 Σ - Δ ADC	41	LIN硬件接口	117
电流通道ADC (I-ADC)	41	位串行设备 (BSD) 接口	121
ADC接地开关	44	BSD通信硬件接口	121
ADC噪声性能表	44	BSD相关MMR	122
ADC MMR接口	45	BSD通信帧	123
ADC功耗模式	54	BSD数据接收	124
ADC比较器和累加器	55	BSD主机发送	124
ADC SINC3 数字滤波器响应	55	BSD接口唤醒	124
ADC校准	58	器件标识	125
ADC诊断	59	原理图	128
电源支持电路	60	外形尺寸	129
系统时钟	61	订购指南	129
系统时钟寄存器	62		
低功耗时钟校准	65		
处理器相关外设	67		

修订历史

2009年8月—修订版0至修订版A

修改“特性”部分	1
修改图1	1
修改表1	4, 8, 9
修改表3	11
修改表4	12
修改表5	13
修改图8、图9和图10	18
修改“工作原理”部分	20
修改“Flash/EE存储器可靠性”部分	31
修改表46	64
修改“正常中断（IRQ）请求”部分	68
修改“定时器0—使用寿命定时器”部分	71
修改“定时器1”部分	73
修改“定时器2—唤醒定时器”部分	75
修改“定时器3接口”部分	77
修改“定时器4—STI定时器”部分	79
修改“BSD通信帧”部分	123
修改表97	125
修改图57	128
修改“订购指南”部分	129

2008年10月—版本0:初始版

技术规格

电气规格

VDD = 3.5 V至18 V, VREF = 1.2 V内部基准电压, $f_{\text{CORE}} = 20.48 \text{ MHz}$ (除非另有说明), 采用外部32.768 kHz时钟晶体或片内精密振荡器驱动。除非另有说明, 所有规格 $T_A = -40^\circ\text{C}$ 至 $+115^\circ\text{C}$ 。

表1

参数	测试条件/注释	最小值	典型值	最大值	单位
ADC技术规格					
转换速率 ¹	斩波关闭, ADC正常工作模式	4		8000	Hz
	斩波开启, ADC正常工作模式	4		2600	Hz
	斩波开启, ADC低功耗模式	1		650	Hz
电流通道					
无失码 ¹	对所有ADC更新速率和ADC模式有效	16			位
积分非线性 (INL) ^{1,2}			±10	±60	FSR的ppm
失调误差 ^{2,3,4,5}	斩波关闭, 1 LSB = (36.6/增益) μV	-10	±3	+10	LSB
失调误差 ^{1,3,6}	斩波开启	-2	±0.5	+2	μV
失调误差 ^{1,3}	斩波开启, 低功耗或超低功耗模式, MCU关断	100	-50	-300	nV
失调误差 ^{1,3}	斩波开启, 正常模式	+0.5	-1.25	-3	μV
失调误差漂移 ⁶	斩波关闭, 对于ADC增益4至64有效, 正常模式		0.03		LSB/ $^\circ\text{C}$
失调误差漂移 ⁶	斩波关闭, 对于ADC增益128至512有效, 正常模式		30		nV/ $^\circ\text{C}$
失调误差漂移 ⁶	斩波开启		10		nV/ $^\circ\text{C}$
总增益误差 ^{1,3,7,8,9,10}	正常模式	-0.5	±0.1	+0.5	%
总增益误差 ^{1,3,7,9}	低功耗模式, 使用ADCREFL MMR	-4	±0.2	+4	%
总增益误差 ^{1,3,7,9,11}	超低功耗模式, 使用精密VREF	-1	±0.2	+1	%
增益漂移			3		ppm/ $^\circ\text{C}$
PGA增益不匹配误差			±0.1		%
输出噪声 ^{1,12}	4 Hz更新速率, 增益 = 512, ADCFLT = 0xBF1D		60	90	nV均方根值
	4 Hz更新速率, 增益 = 512, ADCFLT = 0x3F1D		75	115	nV均方根值
	10 Hz更新速率, 增益 = 512, ADCFLT = 0x961F		100	150	nV均方根值
	10 Hz更新速率, 增益 = 512, ADCFLT = 0x161F		120	180	nV均方根值
	1 kHz更新速率, 增益 ≥ 64 , ADCFLT = 0x8101		0.8	1.2	μV 均方根值
	1 kHz更新速率, 增益 ≥ 64 , ADCFLT = 0x0101		1	1.5	μV 均方根值
	1 kHz更新速率, 增益 = 512, ADCFLT = 0x0007		0.6	0.9	μV 均方根值
	1 kHz更新速率, 增益 = 32, ADCFLT = 0x0007		0.8	1.2	μV 均方根值
	1 kHz更新速率, 增益 = 8, ADCFLT = 0x8101		2.1	4.1	μV 均方根值
	1 kHz更新速率, 增益 = 8, ADCFLT = 0x0007		1.6	2.4	μV 均方根值
	1 kHz更新速率, 增益 = 8, ADCFLT = 0x0101		2.6	3.9	μV 均方根值
	1 kHz更新速率, 增益 = 4, ADCFLT = 0x0007		2.0	2.8	μV 均方根值
	8 kHz更新速率, 增益 = 32, ADCFLT = 0x0000		2.5	3.5	μV 均方根值
	8 kHz更新速率, 增益 = 4, ADCFLT = 0x0000		14	21	μV 均方根值
	ADC低功耗模式, $f_{\text{ADC}} = 10 \text{ Hz}$, 增益 = 128		1.25	1.9	μV 均方根值
	ADC低功耗模式, $f_{\text{ADC}} = 1 \text{ Hz}$, 增益 = 128		0.35	0.5	μV 均方根值
	ADC超低功耗模式, $f_{\text{ADC}} = 1 \text{ Hz}$, 增益 = 512		0.1	0.15	μV 均方根值
	ADC超低功耗模式, $f_{\text{ADC}} = 250 \text{ Hz}$, 增益 = 512		0.6	0.9	μV 均方根值

参数	测试条件/注释	最小值	典型值	最大值	单位
电压通道 ¹³					
无失码 ¹	在所有ADC更新速率下有效	16			位
积分非线性 (INL) ¹			±10	±60	FSR的ppm
失调误差 ^{3,5}	斩波关闭, 1 LSB = 439.5 μV	-10	±1	+10	LSB
失调误差 ^{1,3}	斩波开启		0.3	1	LSB
失调误差漂移	斩波关闭		0.03		LSB/°C
总增益误差 ^{1,3,7,10,14}	包括电阻不匹配	-0.25	±0.06	+0.25	%
总增益误差 ^{1,3,7,10,14}	温度范围 = -25°C至+65°C	-0.15	±0.03	+0.15	%
增益漂移	包括电阻不匹配漂移		3		ppm/°C
输出噪声 ^{1,12,15}	4 Hz更新速率, ADCFLT = 0xBF1D		60	90	μV均方根值
	10 Hz更新速率, ADCFLT = 0x961F		60	90	μV均方根值
	1 kHz更新速率, ADCFLT = 0x0007		180	270	μV均方根值
	1 kHz更新速率, ADCFLT = 0x8101		240	307	μV均方根值
	1 kHz更新速率, ADCFLT = 0x0101		270	405	μV均方根值
	8 kHz更新速率, ADCFLT = 0x0000		1600	2400	μV均方根值
温度通道					
无失码 ¹	在所有ADC更新速率下有效	16			位
积分非线性 (INL) ¹			±10	±60	FSR的ppm
失调误差 ^{3,4,5,16}	斩波关闭, 在单极性模式下1 LSB = 19.84 μV	-10	±3	+10	LSB
失调误差 ^{1,3}	斩波开启	-5	+1	+5	LSB
失调误差漂移	斩波关闭		0.03		LSB/°C
总增益误差 ^{1,3,14}	使用REG_AVDD作为基准电压	-0.2	±0.06	+0.2	%
增益漂移			3		ppm/°C
输出噪声 ¹	1 kHz更新速率		7.5	11.25	μV均方根值
ADC模拟输入规格	内部VREF = 1.2 V				
电流通道					
绝对输入电压范围	同时适用于IIN+和IIN-	-200		+300	mV
输入电压范围 ^{17,18}	增益 = 1 ¹⁹		±1.2		V
	增益 = 2 ¹⁹		±600		mV
	增益 = 4 ¹⁹		±300		mV
	增益 = 8		±150		mV
	增益 = 16		±75		mV
	增益 = 32		±37.5		mV
	增益 = 64		±18.75		mV
	增益 = 128		±9.375		mV
	增益 = 256		±4.68		mV
	增益 = 512		±2.3		mV
输入漏电流 ¹		-3		+3	nA
输入失调电流 ^{1,20}			0.5	1.5	nA
电压通道					
绝对输入电压范围		4		18	V
输入电压范围			0至28.8		V
VBAT输入电流	VBAT = 18 V	3	5.5	8	μA
温度通道	VREF = (REG_AVDD及GND_SW)/2				
绝对输入电压范围		100		1300	mV
输入电压范围			0至VREF		V
VTEMP输入电流 ¹			2.5	100	nA

ADuC7036

参数	测试条件/注释	最小值	典型值	最大值	单位
基准电压源					
ADC精密基准电压					
内部VREF			1.2		V
上电时间 ¹			0.5		ms
初始精度 ¹	在T _A = 25°C时测定	-0.15		+0.15	%
温度系数 ^{1,21}		-20	±5	+20	ppm/°C
基准电压长期稳定性 ²²			100		ppm/1000小时
外部基准电压输入范围 ²³		0.1		1.3	V
VREF二分频初始误差 ¹			0.1	0.3	%
ADC低功耗基准电压					
内部VREF			1.2		V
初始精度	在T _A = 25°C时测定	-5		+5	%
初始精度 ¹	使用ADCREF, 在T _A = 25°C时测定		0.1		%
温度系数 ^{1,21}		-300	±150	+300	ppm/°C
ADC诊断					
VREF/136 ¹	在任何增益设置时	8.5		9.4	mV
电压衰减器电流源 ¹	当电流源开启时差分电压在衰减器上递增, 温度范围 = -40°C至+85°C	3.1		3.8	V
电阻衰减器					
分频比			24		
电阻不匹配漂移			3		ppm/°C
ADC接地开关					
电阻	直接接地路径		10		Ω
电阻 ¹	选择20 kΩ电阻	10	20	30	kΩ
输入电流	允许持续电流通过具有直接接地路径的开关			6	mA
温度传感器 ²⁴	用户校准后				
精度	MCU处于省电或待机模式		±3		°C
	MCU处于省电或待机模式, 温度范围 = -25°C至+65°C		±2		°C
上电复位 (POR)					
POR触发电平	参见VDD引脚的电压	2.85	3	3.15	V
POR迟滞			300		mV
POR复位超时			20		ms
低压标志 (LVF)					
LVF电平	参见VDD引脚的电压	1.9	2.1	2.3	V
电源监控器(PSM)					
PSM触发电平	参见VDD引脚的电压		6		V
看门狗定时器(WDT)					
超时周期 ¹	32.768 kHz时钟, 256预分频	0.008		512	s
超时步幅			7.8		ms
FLASH/EE存储器 ¹					
耐久性 ²⁵		10,000			周期
数据保持 ²⁶		20			年
数字输入	除NTRST外的所有数字输入				
输入漏电流	输入高电平 = REG_DVDD		±1	±10	μA
输入上拉电流	输入低电平 = 0 V	-80	-20	-10	μA
输入电容			10		pF
输入漏电流	仅限NTRST: 输入低电平 = 0 V		±1	±10	μA
输入下拉电流	仅限NTRST: 输入高电平 = REG_DVDD	30	55	100	μA

参数	测试条件/注释	最小值	典型值	最大值	单位
逻辑输入 ¹	所有逻辑输入				
V _{INL} , 输入低电平				0.4	V
V _{INH} , 输入高电平		2			V
晶体振荡器 ¹					
逻辑输入, 仅限XTAL1					
V _{INL} , 输入低电平				0.8	V
V _{INH} , 输入高电平		1.7			V
XTAL1电容			12		pF
XTAL2电容			12		pF
片内振荡器					
低功耗振荡器			131.072		kHz
精度 ²⁷	包括1000小时使用寿命测试的漂移数据	-3		+3	%
精密振荡器			131.072		kHz
精度	包括1000小时使用寿命测试的漂移数据	-1		+1	%
MCU时钟速率	在此范围内有八种可编程内核时钟可选 (二进制除法1, 2, 4, 8, ... 64, 128)	0.16	10.24	20.48	MHz
MCU启动时间					
上电时	包括内核上电执行时间		25		ms
复位事件后	包括内核上电执行时间		5		ms
从MCU关断起					
振荡器运行					
中断唤醒			2		ms
LIN唤醒			2		ms
晶体关断					
中断唤醒			500		ms
内部PLL锁定时间			1		ms
LIN通用输入/输出					
波特率		1000		20,000	位/秒
VDD	LIN接口工作电压范围	7		18	V
输入电容			5.5		pF
输入漏电流	输入(低电平) = IO_VSS	-800		-400	μA
LIN比较器响应时间 ¹	使用22 Ω电阻		38	90	μs
I _{LIN_DOM_MAX}	LIN总线处于主动状态时驱动器电流限值, VBAT = VBAT (最大值)	40		200	mA
I _{LIN_PAS_REC}	驱动器关闭, 7 V < V _{LIN} < 18 V, VDD = V _{LIN} - 0.7 V	-20		+20	μA
I _{LIN} ¹	VBAT断开, VDD = 0 V, 0 < V _{LIN} < 18 V			10	μA
I _{LIN_PAS_DOM} ¹	输入泄漏V _{LIN} = 0 V	-1			mA
I _{LIN_NO_GND} ²⁸	控制单元接地断开, GND = VDD, 0 V < V _{LIN} < 18 V, VBAT = 12 V	-1		+1	mA
V _{LIN_DOM} ¹	LIN接收器主动状态, VDD > 7 V			0.4 VDD	V
V _{LIN_REC} ¹	LIN接收器被动状态, VDD > 7 V	0.6 VDD			V
V _{LIN_CNT} ¹	LIN接收器中心电压, VDD > 7 V	0.475 VDD	0.5 VDD	0.525 VDD	V
V _{HYS} ¹	LIN接收器迟滞电压			0.175 VDD	V
V _{LIN_DOM_DRV_LOSUP} ¹	LIN主动输出电压, VDD = 7 V			1.2	V
R _{LOAD} = 500 Ω					V
R _{LOAD} = 1000 Ω		0.6			V
V _{LIN_DOM_DRV_HISUP} ¹	LIN主动输出电压, VDD = 18 V			2	V
R _{LOAD} = 500 Ω					V
R _{LOAD} = 1000 Ω		0.8			V
V _{LIN_RECESSIVE}	LIN被动输出电压	0.8 VDD			V
VBAT偏移 ²⁸		0		0.1 VDD	V
GND偏移 ²⁸		0		0.1 VDD	V

ADuC7036

参数	测试条件/注释	最小值	典型值	最大值	单位
R_{SLAVE}	从机端接电阻	20	30	47	k Ω
$V_{SERIAL DIODE}^{28}$	串行二极管处的压降, D_{SER_INT}	0.4	0.7	1	V
发送传播延迟对称性 ¹	VDD (最小值) = 7 V	-2		+2	μ s
接收传播延迟 ¹	VDD (最小值) = 7 V			6	μ s
接收传播延迟对称性 ¹	VDD (最小值) = 7 V	-2		+2	μ s
LIN 1.3版规格	总线负载条件 ($C_{BUS} R_{BUS}$):1 nF 1 k Ω ; 6.8 nF 660 Ω ; 10 nF 500 Ω				
$\left \frac{dV}{dt}\right ^1$	压摆率 主动和被动沿, VBAT = 18 V	1	2	3	V/ μ s
$\left \frac{dV}{dt}\right ^1$	压摆率 主动和被动沿, VBAT = 7 V	0.5		3	V/ μ s
t_{SYM}^1	上升和下降沿对称性, VBAT = 18 V	-5		+5	μ s
	上升和下降沿对称性, VBAT = 7 V	-4		+4	μ s
LIN 2.0版规格	总线负载条件 ($C_{BUS} R_{BUS}$):1 nF 1 k Ω ; 6.8 nF 660 Ω ; 10 nF 500 Ω				
D1	占空比1, $TH_{REC(MAX)} = 0.744 \times V_{BAT}$, $TH_{DOM(MAX)} = 0.581 \times V_{BAT}$, $V_{SUP} = 7 V \dots 18 V$; $t_{BIT} = 50 \mu$ s, $D1 = t_{BUS_REC(MIN)}/(2 \times t_{BIT})$	0.396			
D2	占空比2, $TH_{REC(MIN)} = 0.284 \times V_{BAT}$, $TH_{DOM(MIN)} = 0.422 \times V_{BAT}$, $V_{SUP} = 7 V \dots 18 V$; $t_{BIT} = 50 \mu$ s, $D2 = t_{BUS_REC(MAX)}/(2 \times t_{BIT})$			0.581	
BSD输入/输出 ²⁹					
波特率		1164	1200	1236	位/秒
输入漏电流	输入高电平 = VDD, 或输入低电平 = IO_VSS	-50		+50	μ A
V_{OL} , 输出低电压				1.2	V
V_{OH} , 输出高电压		0.8 VDD			V
$I_{O(SC)}$ 短路输出电流	$V_{BSD} = VDD = 12 V$	50	80	120	mA
V_{INL} , 输入低电平				1.8	V
V_{INH} , 输入高电平		0.7 VDD			V
唤醒	$R_{LOAD} = 300 \Omega$, $C_{BUS} = 91 nF$, $R_{LIMIT} = 39 \Omega$				
VDD ¹	WU引脚工作电压范围	7		18	V
输入漏电流	输入高电平 = VDD 输入低电平 = IO_VSS	0.4		2.1	mA
		-50		+50	μ A
V_{OH}^{30}	输出高电平	5			V
V_{OL}^{30}	输出低电平			2	V
V_{IH}	输入高电平	4.6			V
V_{IL}	输入低电平			1.2	V
单稳态触发器超时	超时周期	0.6	1.3	2	s
$I_{O(SC)}$ 短路输出电流		100	140		mA
串行测试接口	$R_{LOAD} = 500 \Omega$, $C_{BUS} = 2.4 nF$, $R_{LIMIT} = 39 \Omega$				
波特率				40	kbps
输入漏电流	输入高电平 = VDD, 或输入低电平 = IO_VSS	-50		+70	μ A
VDD	STI工作电压范围	7		18	V
V_{OH}	输出高电平	0.6 VDD			V
V_{OL}	输出低电平			0.4 VDD	V
V_{IH}	输入高电平	0.6 VDD			V
V_{IL}	输入低电平			0.4 VDD	V

参数	测试条件/注释	最小值	典型值	最大值	单位
封装热规格 热关断 ^{1,31} 热阻(θ_{JA}) ³²	48引脚LFCSP, 堆叠式芯片	140	150	160	°C
电源要求 电源电压 VDD (电池电源) REG_DVDD, REG_AVDD ³³		3.5		18	V
功耗 I _{DD} (MCU正常模式) ³⁴	MCU时钟速率 = 10.24 MHz, ADC关闭 MCU时钟速率 = 20.48 MHz, ADC关闭 (仅对ADuC7036CCPZ有效)		10	20	mA
I _{DD} (MCU关断) ¹	ADC低功耗模式, 在T _A = -10°C至+40°C时测定, 连续ADC转换		300	400	μA
	ADC低功耗模式, 在T _A = -40°C至+85°C时测定, 连续ADC转换		300	500	μA
	ADC超低功耗模式, 在T _A = -10°C至+40°C时测定, 连续ADC转换		520	700	μA
	平均电流, 在唤醒和看门狗定时器从低功耗振荡器获取时钟信号时测定, T _A = -40°C至+85°C		120	300	μA
I _{DD} (MCU关断)	平均电流, 在T _A = -10°C至+40°C, 唤醒和看门狗定时器从低功耗振荡器获取时钟信号时测定		120	175	μA
I _{DD} (电流ADC)			1.7		mA
I _{DD} (电压/温度ADC)			0.5		mA
I _{DD} (精密振荡器)			400		μA

¹ 这些参数未经过产品测试, 但在产品发布时由设计和/或特性数据保证。

² 对于电流ADC增益设置PGA = 4至64有效。

³ 这些参数包括温度漂移。

⁴ 在增益范围= 4时测试; 自失调校准可以消除此误差。

⁵ 初始失调校准后利用内部短路测定。

⁶ 利用内部短路测定

⁷ 这些参数包括内部基准电压温度漂移。

⁸ 增益 = 1时工厂校准。

⁹ 在具体增益范围 (及温度) 内的系统校准可以消除在此范围内的误差。

¹⁰ 包括初始系统校准。

¹¹ 采用ADC正常模式基准电压源。

¹² 在低功耗模式下的典型噪声在斩波启用时测定。

¹³ 电压通道规格包括电阻衰减器输入级。

¹⁴ 系统校准可以在指定温度下消除此误差。

¹⁵ 均方根噪声折合到电压衰减器输入 (例如在f_{ADC} = 1 kHz时, ADC输入端处的典型均方根噪声为7.5 μV) 并由衰减器比例调整 (除以24) 以产生这些输入折合噪声规格/值。

¹⁶ 初始自校准后有效。

¹⁷ 在ADC低功耗模式下, 输入范围固定在±9.375 mV处。在ADC超低功耗模式下, 输入范围固定在±2.34375 mV处。

¹⁸ 通过修改增益校准寄存器的工厂设定值或利用系统校准, 可将ADC输入范围扩展10%, 也可缩小ADC输入范围 (LSB大小)。

¹⁹ 受最小/最大绝对输入电压范围限制。

²⁰ 对于低于10mV的差分输入有效。

²¹ 利用盒子方法测定。

²² 长期稳定性规格为非累积性。在后续1000小时周期内的漂移大幅低于第一个1000小时周期。

²³ 通过内部二分频, 可使用高达REG_AVDD的基准电压。

²⁴ 芯片温度。

²⁵ 耐久性是在-40°C、+25°C及+125°C时依据JEDEC 22标准方法A117认定为10,000个周期。在25°C时的典型耐久性为170,000个周期。

²⁶ 根据JEDEC 22标准方法A117, 保持期限相当于85°C结温时的寿命。保持期限会随着结温递减。

²⁷ 可针对精密振荡器或用户代码中的外部32.768 kHz晶体校准低功耗振荡器。

²⁸ 这些参数未经产品测试, 但得到LIN兼容性测试支持。

²⁹ 除高低电平外, BSD电气规格均依照LIN 2.0, 禁用上拉电阻且C_L = 10 nF最大值。

³⁰ 在39 Ω R_{LIMIT}后指定。

³¹ MCU内核不关断但被中断, 并响应热关断事件而禁用高压I/O引脚。

³² 热阻可用于计算环境到芯片温度的热梯度。

³³ 内部调节电源在REG_DVDD (I_{SOURCE} = 5 mA) 和REG_AVDD (I_{SOURCE} = 1 mA) 处提供。

³⁴ 所列指标均为典型值; 在Flash/EE存储器编程和擦除周期期间的额外电源电流消耗分别为7mA和5mA。

时序规格

SPI时序规格

表2. SPI主机定时—相位模式 = 1

参数	描述	最小值	典型值	最大值	单位
t_{SL}	SCLK低脉冲宽度 ¹		$(SPIDIV + 1) \times t_{HCLK}$		ns
t_{SH}	SCLK高脉冲宽度 ¹		$(SPIDIV + 1) \times t_{HCLK}$		ns
t_{DAV}	SCLK沿之后数据输出有效 ²			$(2 \times t_{UCLK}) + (2 \times t_{HCLK})$	ns
t_{DSU}	SCLK沿之前数据输入建立时间	0			ns
t_{DHD}	SCLK沿之后数据输入保持时间 ²	$3 \times t_{UCLK}$			ns
t_{DF}	数据输出下降时间		3.5		ns
t_{DR}	数据输出上升时间		3.5		ns
t_{SR}	SCLK上升时间		3.5		ns
t_{SF}	SCLK下降时间		3.5		ns

¹ t_{HCLK} 取决于POWCON MMR的时钟分频器 (CD) 位。 $t_{HCLK} = t_{UCLK}/2^{CD}$ 。

² $t_{UCLK} = 48.8$ ns。与时钟分频器之前从锁相环输出的20.48 MHz内部时钟相对应。

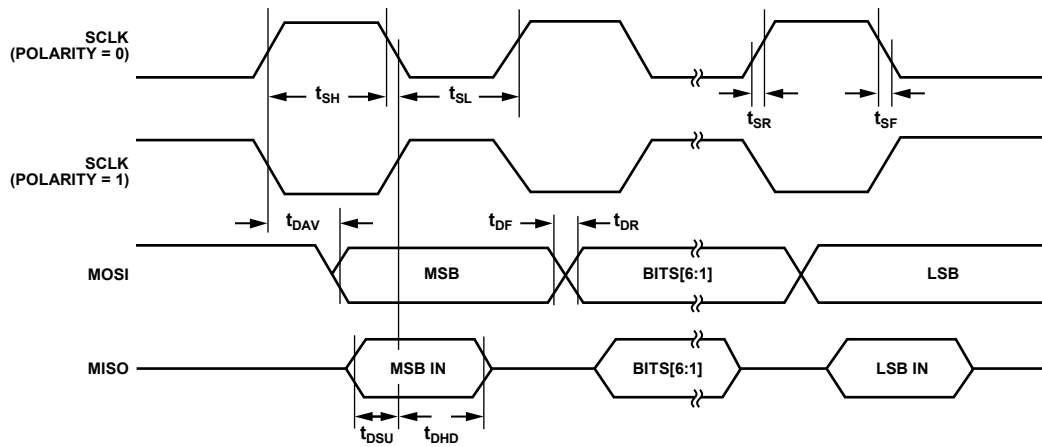


图2. SPI 主机定时—相位模式 = 1

07474-002

表3. SPI主机模式—相位模式 = 0

参数	描述	最小值	典型值	最大值	单位
t_{SL}	SCLK低脉冲宽度 ¹		$(SPIDIV + 1) \times t_{HCLK}$		ns
t_{SH}	SCLK高脉冲宽度 ¹		$(SPIDIV + 1) \times t_{HCLK}$		ns
t_{DAV}	SCLK沿之后数据输出有效 ²			$(2 \times t_{UCLK}) + (2 \times t_{HCLK})$	ns
t_{DOSU}	SCLK沿之前数据输出建立		0.5 t_{SL}		ns
t_{DSU}	SCLK沿之前数据输入建立时间	0			ns
t_{DHD}	SCLK沿之后数据输入保持时间 ²	$3 \times t_{UCLK}$			ns
t_{DF}	数据输出下降时间		3.5		ns
t_{DR}	数据输出上升时间		3.5		ns
t_{SR}	SCLK上升时间		3.5		ns
t_{SF}	SCLK下降时间		3.5		ns

¹ t_{HCLK} 取决于POWCON MMR的时钟分频器 (CD) 位。 $t_{HCLK} = t_{UCLK}/2^{CD}$ 。

² $t_{UCLK} = 48.8$ ns。与时钟分频器之前从锁相环输出的20.48 MHz内部时钟相对应。

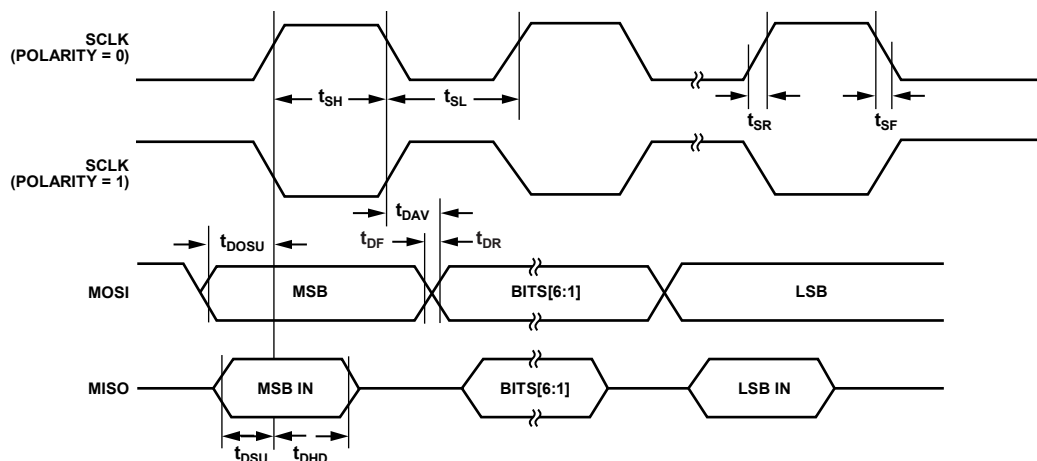


图3. SPI 主机定时—相位模式 = 0

07474-003

ADuC7036

表4. SPI从机定时—相位模式 = 1

参数	描述	最小值	典型值	最大值	单位
$t_{\overline{SS}}$	\overline{SS} 至SCLK沿		$0.5 t_{SL}$		ns
t_{SL}	SCLK低脉冲宽度 ¹		$(SPIDIV + 1) \times t_{HCLK}$		Ns
t_{SH}	SCLK高脉冲宽度 ¹		$(SPIDIV + 1) \times t_{HCLK}$		ns
t_{DAV}	SCLK沿之后数据输出有效 ²			$(3 \times t_{UCLK}) + (2 \times t_{HCLK})$	ns
t_{DSU}	SCLK沿之前数据输入建立时间	0			ns
t_{DHD}	SCLK沿之后数据输入保持时间 ²	$4 \times t_{UCLK}$			ns
t_{DF}	数据输出下降时间		3.5		ns
t_{DR}	数据输出上升时间		3.5		ns
t_{SR}	SCLK上升时间		3.5		ns
t_{SF}	SCLK下降时间		3.5		ns
t_{SFS}	\overline{SS} 在SCLK沿后变高		$0.5 t_{SL}$		ns

¹ t_{HCLK} 取决于POWCON MMR的时钟分频器 (CD) 位。 $t_{HCLK} = t_{UCLK} / 2^{CD}$ 。

² $t_{UCLK} = 48.8$ ns。与时钟分频器之前从锁相环输出的20.48 MHz内部时钟相对应。

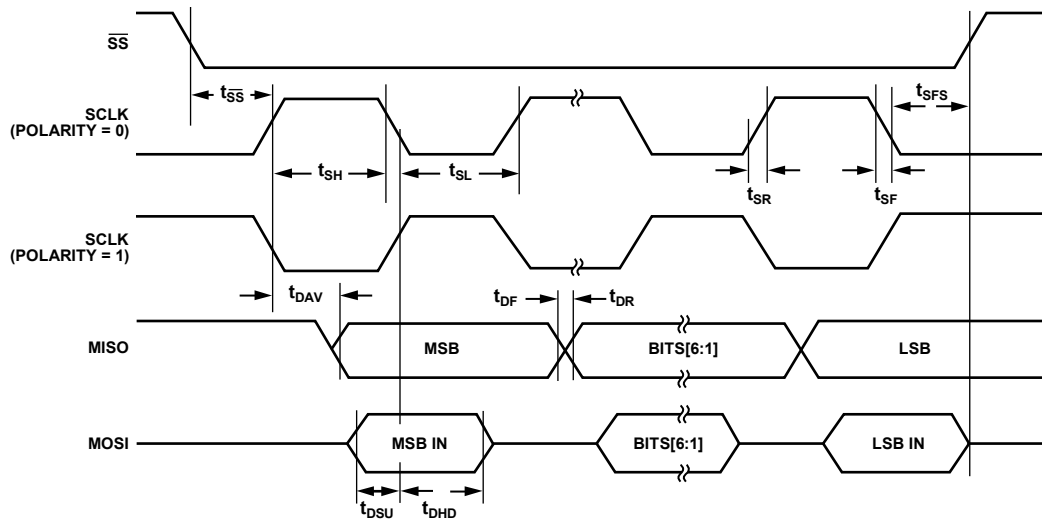


图4. SPI从机定时—相位模式 = 1

07474-004

表5. SPI从机定时(相位模式=0)

参数	描述	最小值	典型值	最大值	单位
$t_{\overline{SS}}$	\overline{SS} 至SCLK沿		$0.5 t_{SL}$		ns
t_{SL}	SCLK低脉冲宽度 ¹		$(SPIDIV + 1) \times t_{HCLK}$		ns
t_{SH}	SCLK高脉冲宽度 ¹		$(SPIDIV + 1) \times t_{HCLK}$		ns
t_{DAV}	SCLK沿之后数据输出有效 ^{1,2}			$(3 \times t_{UCLK}) + (2 \times t_{HCLK})$	ns
t_{DSU}	SCLK沿之前数据输入建立时间	0			ns
t_{DHD}	SCLK沿之后数据输入保持时间 ^{1,2}	$4 \times t_{UCLK}$			ns
t_{DF}	数据输出下降时间		3.5		ns
t_{DR}	数据输出上升时间		3.5		ns
t_{SR}	SCLK上升时间		3.5		ns
t_{SF}	SCLK下降时间		3.5		ns
t_{DOCS}	\overline{SS} 沿之后数据输出有效 ²			$(3 \times t_{UCLK}) + (2 \times t_{HCLK})$	ns
t_{SFS}	\overline{SS} 在SCLK沿后变高		$0.5 t_{SL}$		ns

¹ t_{HCLK} 取决于POWCON MMR的时钟分频器(CD)位。 $t_{HCLK} = t_{UCLK}/2^{CD}$ 。

² $t_{UCLK} = 48.8$ ns。与时钟分频器之前从锁相环输出的20.48 MHz内部时钟相对应。

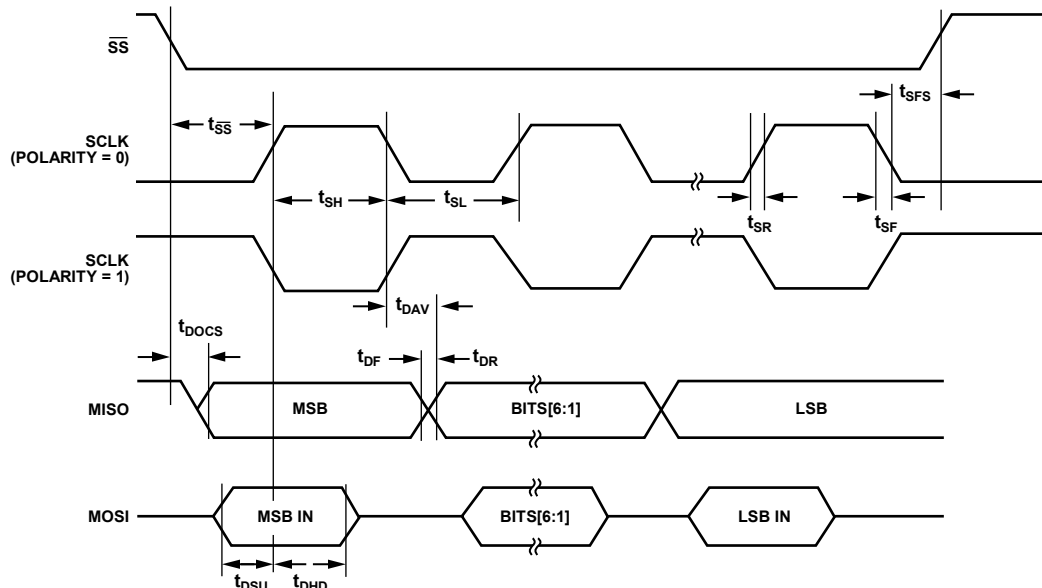


图5. SPI从机定时—相位模式=0

07474-005

LIN 定时规格

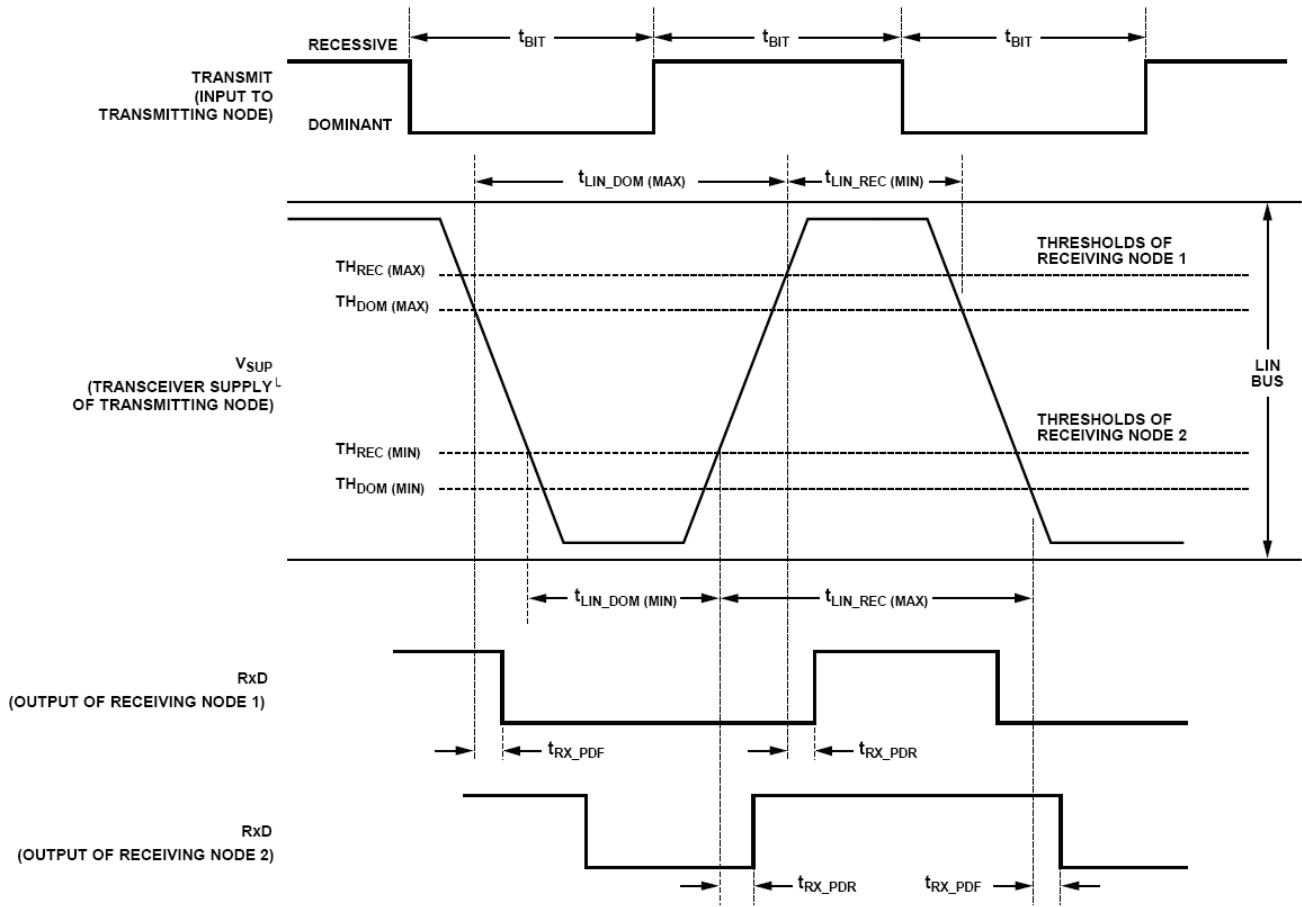


图 6. LIN 2.0 时序规格

07174-008

绝对最大额定值

除非另有说明， $T_A = -40^{\circ}\text{C}$ 至 $+115^{\circ}\text{C}$ 。

表6

参数	额定值
AGND 至 DGND 至 VSS 至 IO_VSS	-0.3 V至+0.3 V
VBAT至AGND	-22 V至+40 V
VDD至VSS	-0.3 V至+33 V
VDD至VSS达1秒	-0.3 V至+40 V
LIN至IO_VSS	-16 V至+40 V
STI和WU至IO_VSS	-3 V至+33 V
唤醒连续电流	50 mA
高压I/O引脚短路电流	100 mA
数字I/O电压至DGND	-0.3 V至REG_DVDD + 0.3 V
VREF至AGND	-0.3 V至REG_AVDD + 0.3 V
ADC输入至AGND	-0.3 V至REG_AVDD + 0.3 V
ESD额定值	
IEC 1000-4-2适用于所有引脚	1 kV
IEC 61000-4-2适用于LIN和VBAT引脚	± 5 kV
存储温度	125°C
结温	
瞬变	150°C
连续	130°C
引脚温度	
回流焊（15秒）	260°C

注意，超出上述绝对最大额定值可能会导致器件永久性损坏。这只是强调的额定值，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。

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。

ADuC7036

引脚配置和功能描述

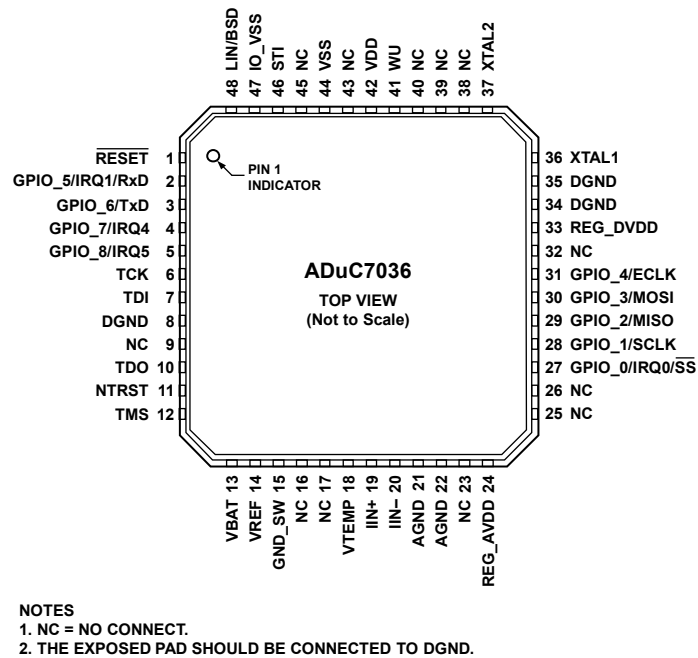


图7. 引脚配置

表7. 引脚功能描述

引脚编号	引脚名称	类型 ¹	描述
1	RESET	I	复位输入引脚。低电平有效。有连接至REG_DVDD的内部弱上拉电阻，不用时须断开。如需更高安全性和鲁棒性，建议通过电阻将此引脚绑定至REG_DVDD。
2	GPIO_5/IRQ1/RxD	I/O	通用数字I/O 5/外部中断请求 1/接收UART串行端口数据。默认情况下以及在上电复位后，配置为输入端。有内置弱上拉电阻，不用时断开。
3	GPIO_6/TxD	I/O	通用数字I/O 6/发送UART串行端口数据。默认情况下以及在上电复位后，配置为输入端。有内置弱上拉电阻，不用时断开。
4	GPIO_7/IRQ4	I/O	通用数字I/O 7/外部中断请求 4。默认情况下以及在上电复位后，配置为输入端。有内置弱上拉电阻，不用时断开。
5	GPIO_8/IRQ5	I/O	通用数字I/O 8/外部中断请求5。默认情况下以及在上电复位后，配置为输入端。有内置弱上拉电阻，不用时断开。
6	TCK	I	JTAG测试时钟。时钟输入引脚，是器件标准5针JTAG调试端口之一。只能用作输入引脚，有内置弱上拉电阻。不用时断开。
7	TDI	I	JTAG测试数据输入。数据输入引脚，是器件标准5针JTAG调试端口之一。只能用作输入引脚，有内置弱上拉电阻。不用时断开。
8, 34, 35	DGND	S	片内数字电路接地基准。
9、16、17、23、25、26、32、38至40、43、45	NC		不连接。不内接并保留待用。因此不外接。必要时接地。
10	TDO	O	JTAG测试数据输出。数据输出引脚，是器件标准5针JTAG调试端口之一。只能用作输出引脚。上电时禁用并通过内部弱上拉电阻拉高。不用时断开。

引脚编号	引脚名称	类型 ¹	描述
11	NTRST	I	JTAG测试复位。复位输入引脚，是器件标准5针JTAG调试端口之一。只能用作输入引脚，有内置弱下拉电阻。不用时断开。同时受到片上内核监控以能使LIN引导加载模式。
12	TMS	I	JTAG测试模式选择。模式选择引脚，是器件标准5针JTAG调试端口之一。只能用作输入引脚，有内置弱上拉电阻。不用时断开。
13	VBAT	I	电阻分压器的电池电压输入。
14	VREF	I	外部基准电压输入端子。不用时，直接接到AGND系统地，也可断开。
15	GND_SW	I	切换到内部模拟地基准。外部温度通道和外部基准电压的负输入端。不用时，直接接到AGND系统地。
18	VTEMP	I	NTC/PTC温度测量的外部引脚。
19	IIN+	I	电流通道的正差分输入
20	IIN-	I	电流通道的负差分输入。
21, 22	AGND	S	片内精密模拟电路的接地基准。
24	REG_AVDD	S	片内稳压器的额定2.6 V输出。
27	GPIO_0/IRQ0/ \overline{SS}	I/O	通用数字I/O 0/外部中断请求 0/SPI接口的从机选择输入。默认情况下以及在上电复位后，配置为输入端。有内置弱上拉电阻，不用时断开。
28	GPIO_1/SCLK	I/O	通用I/O 1/SPI接口的串行时钟输入。默认情况下以及在上电复位后，配置为输入端。有内置弱上拉电阻，不用时断开。
29	GPIO_2/MISO	I/O	通用数字I/O 2/SPI接口的主机输入、从机输出。默认情况下以及在上电复位后，配置为输入端。有内置弱上拉电阻，不用时断开。
30	GPIO_3/MOSI	I/O	通用数字I/O 3/SPI接口的主机输出、从机输入。默认情况下以及在上电复位后，配置为输入端。有内置弱上拉电阻，不用时断开。
31	GPIO_4/ECLK	I/O	通用数字I/O 4/2.56 MHz时钟输出。默认情况下以及在上电复位后，配置为输入端。有内置弱上拉电阻，不用时断开。
33	REG_DVDD	S	片内稳压器的额定2.6 V输出
36	XTAL1	O	晶体振荡器输出。如果不使用外部晶体，此引脚断开。
37	XTAL2	I	晶体振荡器输入。如果不使用外部晶体，将此引脚接到DGND系统地。
41	WU	I/O	高压唤醒引脚。高压I/O引脚，有内置10k Ω 下拉电阻和高端驱动器至VDD。不用时不必外接。
42	VDD	S	片内稳压器的电池电源。
44	VSS	S	接地基准。内部稳压器的接地基准。
46	STI	I/O	高压串行测试接口输出引脚。不用时将其外接到IO_VSS接地基准。
47	IO_VSS	S	高压I/O引脚的接地基准。
48	LIN/BSD	I/O	局域互连网络I/O/位串行设备I/O。高压引脚。
EPAD	裸露焊盘		裸露焊盘应接到DGND。

¹ I= 输入, O= 输出, I/O= 输入/输出, S= 电源。

典型工作特性

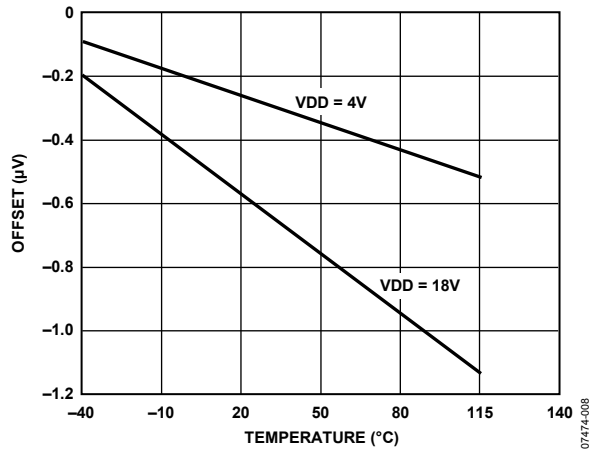


图8. ADC 电流通道失调和温度的关系, 10 MHz MCU

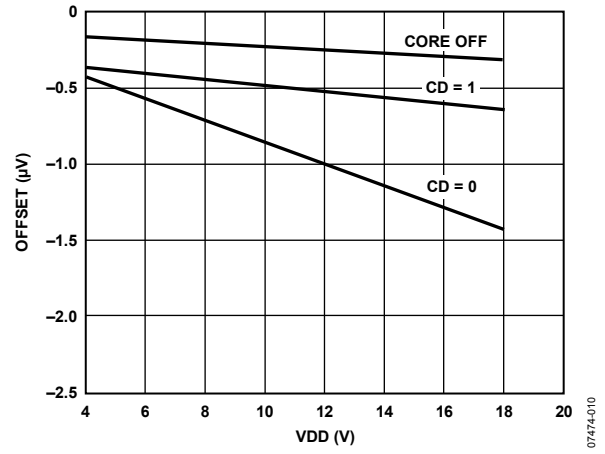


图10. ADC 电流通道失调和VDD 的关系, 25°C

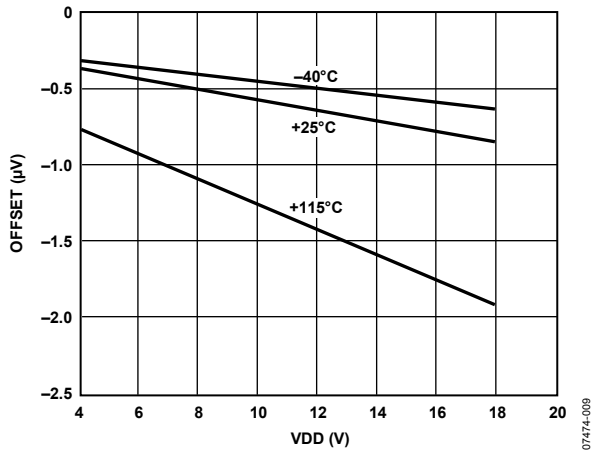


图9. ADC 电流通道失调和VDD 的关系, 10 MHz MCU

术语

转换速率

转换速率是指在ADC稳定后从ADC获得输出结果的速度。

此器件使用的是 Σ - Δ 转换技术,意味着当以相对较高的采样率对ADC前端信号进行过采样时,后接数字滤波器抽取输出,提供有效16位数据转换结果以获得1Hz至8kHz的输出速率。

注意,当软件在相同ADC上切换不同输入时,必须先清空数字滤波器,然后可以对新结果求平均。取决于ADC配置和滤波器类型,这个操作可能需要多个转换周期。

积分非线性 (INL)

转换结果编码偏离通过其传递函数端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 $1/2$ LSB的点,以及在满刻度位置比最后一个编码的跃变点高 $1/2$ LSB的点(111...110到111...111)。该误差表示为满量程的百分比。

无失码

无失码是衡量ADC微分非线性的指标。表示为位(2^N 位,其中N为无失码),指在整个ADC输入范围内保证发生的代码数目(ADC结果)。

失调误差

ADC第一个码转换输入电压和理想码跃迁的偏差即是失调误差。

失调误差漂移

失调误差漂移是指绝对失调误差的温度变化值。表示为LSB/ $^{\circ}$ C。

增益误差

增益误差是衡量ADC量程误差的指标。它考察的是传递函数上任两点的测量量程和理想值的偏差。

输出噪声

输出噪声是指在ADC直流输入信号时输出代码分布的标准偏差(即 $1 \times \Sigma$)。表示为毫伏均方根(μ V rms)。如下列等式所定义,输出噪声或均方根噪声可用来求出ADC的有效分辨率:

$$\text{有效分辨率} = \log_2 (\text{满量程}/\text{均方根噪声})$$

其中有效分辨率表示为位。

峰峰值噪声是指ADC直流输入信号时在输出代码分布 $6.6 \times \Sigma$ 范围内的代码偏差。因此峰峰值噪声是均方根噪声的6.6倍。

如下列等式所定义,峰峰值噪声可用来求出在 $6.6 \times \Sigma$ 限值内无码闪烁的ADC(无噪声代码)分辨率:

$$\text{无噪声代码分辨率} = \log_2 (\text{满量程}/\text{峰峰值噪声})$$

其中无噪声代码分辨率以位表示。

工作原理

ADuC7036是一款适用于在12V汽车电子应用中进行电池监控的完整系统解决方案，集成了所有在各种工作条件下对12V电池参数（如电池电流、电压和温度）进行精确智能监控、处理和诊断等必需的功能。

可直接从12V电池供电，从而最大程度减少了外部系统组件。片内有低压降稳压器向两个集成16位 Σ - Δ ADC供电。这些ADC精确测量电池电流、电压及温度以采集汽车电池的运行和充电状态参数。

片内还集成了一个基于Flash/EE存储器的ARM7™微控制器（MCU），用来预处理获得的电池变量并管理ADuC7036通过片内集成的局域互连网络（LIN）接口和主电子控制单元（ECU）的通信。

MCU和ADC子系统都可以单独配置采用正常工作模式或更灵活的省电工作模式。

在正常工作模式下，MCU通过锁相环（PLL）从片内振荡器间接接收时钟信号，最大时钟速率为20.48 MHz。在省电工作模式下，MCU完全断电，只被ADC转换结果就绪、数字比较器、唤醒定时器、POR或外部串行通信等事件唤醒。

ADC可配置采用正常（全功率）工作模式，在各种采样转换事件后中断MCU。电流通道支持两种低功耗模式：低功耗和超低功耗，根据较低的性能指标来产生转换结果。

片内出厂固件支持通过LIN或JTAG串行接口端口进行在线Flash/EE编程，也支持通过JTAG接口进行非介入仿真。这些功能都集成在支持ADuC7036的低成本QuickStart™开发系统中。

ADuC7036直接从12V电池供电，工作温度范围：-40°C至+115°C。在115°C至125°C温度范围内仍能工作，但性能会降低。

ARM7TDMI内核概览

ARM7内核是一种ARM公司开发的32位精简指令集计算机（RISC）。ARM7TDMI®属于冯·诺依曼架构，指令和数据只能利用32位总线。取决于内核工作模式，数据可采用8、16或32位，指令字长16位或32位。

如表8所列，ARM7TDMI是具有四个功能加强的ARM7内核。

表8. ARM7TDMI

功能	描述
T	支持Thumb®指令集（16位）
D	支持调试
M	增强乘法器。
I	包含EmbeddedICE™模块以支持嵌入式系统调试

Thumb模式 (T)

ARM指令长32位。ARM7TDMI处理器支持16位压缩的第二个指令集，即Thumb指令集。Thumb指令集具有16位存储器的更快代码执行速度和更大的代码密度，使ARM7TDMI内核特别适用于嵌入式应用。

不过，Thumb模式有三个缺点：

- 相对于ARM而言，同一个任务Thumb代码通常需要更多的指令。因此，ARM代码特别适合于在大多数应用中最大程度提升时间关键代码的性能。
- Thumb指令集不含一些异常处理指令，所以可能需要ARM代码进行异常处理。
- 发生中断时，内核矢量指向存储器内的中断地址并执行存储的代码。第一条命令要求在ARM代码内。

乘法器(M)

ARM7TDMI指令集包括一个增强乘法器，具有四条额外指令以进行32位与32位相乘或32位与32位乘加(MAC)来得到64位结果。

EmbeddedICE (I)

EmbeddedICE模块为ARM7TDMI提供集成片内调试功能，包括断点和观察点寄存器，实现非介入用户代码调试。这些寄存器可以通过JTAG测试端口加以控制。当碰到一个断点或观察点时，处理器会暂停执行，进入调试状态。一旦进入调试状态，就可以查询处理器寄存器，也可以查询Flash/EE，SRAM和存储器映射寄存器。

ARM7 异常

ARM7支持5种异常类型，每种类型都对应一种优先处理模式。这5种异常类型为：

- 正常中断或IRQ。用于内部和外部事件的通用中断处理。
- 快速中断或FIQ。用于数据传输或低延迟通信通道处理。FIQ的优先级高于IRQ。
- 存储器中止（预取和数据）
- 执行未定义指令。
- 软件中断（SWI）指令，用于调用操作系统。

一般情况下，程序员将中断定义为IRQ，但要获得优先级更高的中断，可将中断定义为FIQ。

异常优先级和矢量地址见表9。

表9. 异常优先级和矢量地址

优先级	异常	地址
1	硬件复位	0x00
2	存储器中止（数据）	0x10
3	FIQ	0x1C
4	IRQ	0x18
5	存储器中止（预取）	0x0C
6	软件中断 ¹	0x08
6	未定义指令 ¹	0x04

¹ 软件中断和未定义指令异常优先级相同且互斥。

在表9中异常表地址范围为0x00到0x1C，0x14为保留位置。此保留位置需要写入0x27011970或页0的校验和，0x14除外，否则用户代码可能无法执行而转入LIN下载模式。

ARM 寄存器

ARM7TDMI共有16个标准寄存器。R0至R12用于数据操作，R13是堆栈指针，R14是链接寄存器，而R15是程序计数器，指示当前正在执行的指令。链接寄存器存放用户转移地址（使用转移和链接命令情况）或发生异常时的命令。

堆栈指针包含堆栈的当前位置。作为ARM7TDMI的一般规则，堆栈始于可用RAM空间顶部并根据需要递减。每个异常都定义了一个独立的堆栈。用户可根据目标应用来配置每个堆栈的大小。在ADuC7036上，堆栈从0x00040FFC开始递减。利用C等高级语音进行编程时，必须确保堆栈不会溢出。这取决于所用编译器的性能。

当异常发生时，一些标准寄存器会被替换成特定寄存器，而进入异常模式。所有的异常模式都有各自的替换寄存器组，用于堆栈指针（R13）和链接寄存器（R14），如图11所示。FIQ模式具有更多寄存器（R8至R12）以获得更快的中断处理。随着非关键寄存器数目的增加，在中断处理时不必保存或恢复这些寄存器，从而缩短了中断处理过程的响应时间。

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

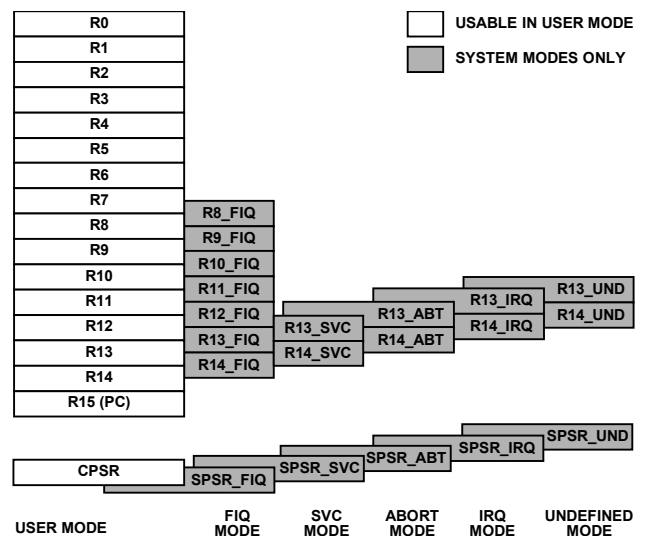


图 11. 寄存器结构图

中断延迟时间

快速中断请求(FIQ)的最大延迟时间包括请求通过同步器、最长指令(LDM)完成载入所有寄存器(包括PC)以及完成数据中止和FIQ跳转的最长可能时间总和。等待这段时间结束后，ARM7TDMI便开始执行在地址0x1C(FIQ中断矢量地址)处的指令。如果系统采用连续20.48MHz处理器时钟，则最长FIQ延迟时间为50个处理器周期，或稍稍超过2.44 μs。

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

ADuC7036

FIQ或IRQ中断的最短延迟时间为5个周期，包括请求通过同步器的最短时间加上进入异常模式的时间。

注意，发生异常时，ARM7TDMI最初(第一个指令)在ARM(32位)模式下运行。必要时，如执行中断服务例行程序时，用户可从ARM模式立即切换到Thumb模式。

存储器结构

由于采用冯·诺依曼架构，ARM7 MCU内核将存储器看作 2^{32} 个字节位置的线性阵列。如图13中所示，ADuC7036将此线性阵列映射成四个不同用户区域：可重映射的存储区域、SRAM区域、Flash/EE区域及存储器映射寄存器(MMR)区域。

- 第一块94kB用作可重新映射片内Flash/EE或SRAM的区域。
- ADuC7036在存储器映射图顶部具有第二块4 kB区域以定址MMR，用来配置并监控所有片内外设。
- ADuC7036的SRAM大小为6 kB。
- ADuC7036的片内Flash/EE存储器大小为96 kB，其中94 kB供用户使用，2KB保留供片上内核使用。

任何对存储器映射图中未定义区域的访问(读写)都会造成数据中止异常。

存储格式

ADuC7036存储被配置成从小到大顺序格式：LSB位于最低字节地址，MSB位于最高字节地址。

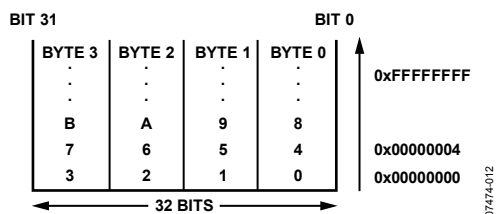


图 12. 从小到大顺序格式

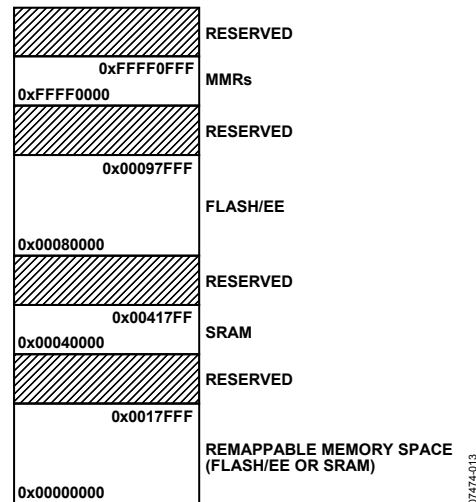


图 13. 存储器映射图

SRAM

ADuC7036的SRAM大小为6kB，从0x00040000起被组织成 1536×32 位，即1536个字。

RAM空间既可用作数据存储，也可用作易失性程序空间。

由于SRAM被配置成32位宽的存储器阵列，ARM代码可以直接从SRAM以完全时钟速度执行。SRAM可采用8、16及32位段读写。

重映射

ARM异常矢量位于存储器阵列底部地址0x00000000到0x00000020。

默认情况下，在复位后Flash/EE存储器会被逻辑映射到地址0x00000000。

如果要将SRAM逻辑重映射到地址0x00000000，需要在0xFFFF0220处设置SYSMAP0 MMR的0位。要使Flash/EE返回到0x00000000，SYSMAP0的0位需清0。

如果要优化ADuC7036的中断延迟时间，将RAM重映射到0x00000000可能比较有利，因为这样代码可以在完全32位ARM模式下以最高内核速度执行。注意，发生异常时，内核默认为ARM模式。

重映射操作

ADuC7036发生复位时，会自动执行工厂内置的程序代码。这就是所谓的隐藏内核，用户代码不能访问。如果ADuC7036处于正常模式，它就执行内核的上电配置例行程序，然后跳转到复位矢量地址0x00000000执行用户的复位异常例行程序。由于复位时Flash/EE镜像到存储器阵列底部，复位例行程序必须始终写入Flash/EE内。

重映射命令必须从绝对Flash/EE地址处执行，不得从可能被SRAM替代的存储器镜像、重映射段执行。如果在从镜像位置执行代码时执行重映射操作，可能会发生预取/数据中止或用户可能观察到异常程序操作。

任何一种复位都将Flash/EE存储器重映射到存储器阵列底部。

SYSMAP0 寄存器

名称：SYSMAP0

地址：0xFFFF0220

默认值：由内核更新

访问类型：读写

功能：此8位寄存器允许用户代码将RAM或Flash/EE空间重映射到起始地址为0x00000000的ARM存储器空间底部。

表10. SYSMAP0 MMR位分配

位	描述
7至1	保留。这些位保留，应由用户代码写入0。
0	重映射位。 由用户设置以将SRAM重映射到0x00000000。 复位后自动清0将Flash/EE存储器重映射到0x00000000。

ADuC7036

复位

有四种复位形式：外部复位、上电复位、看门狗复位及软件复位。RSTSTA寄存器指示上一次复位来源并可由用户代码写入以启动软件复位事件。此寄存器的清0可通过在0xFFFF0234处写入RSTCLR MMR来完成。RSTCLR的位分配镜像RSTSTA的位。这两种寄存器用以在复位异常服务例行程序期间识别复位来源。所有四种复位事件的含义如表12所示。

RSTSTA 寄存器

名称：RSTSTA

地址：0xFFFF0230

默认值：视复位类型而定（见表11）

访问类型：读写

功能：此8位寄存器指示上一次复位事件来源并可由用户代码写入以启动软件复位。

RSTCLR 寄存器

名称：RSTSTA

地址：0xFFFF0234

访问类型：只写

功能：此8位只写寄存器将对应RSTSTA位清0。

表11. RSTSTA/RSTCLR MMR位分配

位	描述
7至4	未使用。全部不用，始终读取0。
3	外部复位。 发生外部复位时自动置1。 通过设置对应RSTCLR位清0。
2	软件复位。 由用户代码置1以产生软件复位。 通过设置对应RSTCLR位清0。 ¹
1	看门狗超时。 发生看门狗复位时自动置1。 通过设置对应RSTCLR位清0。
0	上电复位。 发生上电复位时自动设置。 通过设置对应RSTCLR位清0。

¹ 如果RSTSTA的软件复位位已设，任何不将此位清0的RSTCLR写入操作都会产生软件复位。

表12. 器件复位含义

复位	影响							RSTSTA状态 (复位事件后)
	将外部引脚复位到默认状态	执行内核	复位所有外部MMR（不包括RSTSTA）	复位所有高压间接寄存器	复位外设	复位看门狗定时器	有效RAM ¹	
POR	是	是	是	是	是	是	是/否 ²	RSTSTA[0] = 1
看门狗	是	是	是	是	是	否	是	RSTSTA[1] = 1
软件	是	是	是	是	是	否	是	RSTSTA[2] = 1
外部引脚	是	是	是	是	是	否	是	RSTSTA[3] = 1

¹ 在LIN下载后复位的情况下RAM无效。

² LVF使能后，对RAM的影响取决于HVMON[3]内容。当利用HVCFG0[2]使能LVF时，如果LVF状态位HVMON[3]为1，则POR机制不会破坏RAM。详情见低压标识（LVF）节。

FLASH/EE存储器

ADuC7036片内集成了Flash/EE存储器技术，向用户提供非易失、在线可编程存储器空间。

类似于EEPROM，Flash存储器可在字节水平上进行系统内编程，但在编程前必须先擦除，擦除单位为页模块。因此，通常更确切地被称为Flash/EE存储器。

总之，由于具有非易失性、在线编程、高密度及低成本等特点，Flash/EE更接近于理想存储设备。通过集成于ADuC7036内的Flash/EE存储器技术，用户可以在线更新程序代码空间而不必在远程操作节点处替换一次性可编程（OTP）设备。

Flash/EE存储器位于地址0x80000。硬件复位时，Flash/EE存储器被映射到地址0x00000000。所有Flash/EE存储器位置的工厂设置默认内容为0xFF。可采用8、16及32位段读取并采用16位段写入。额定耐久性为10,000个周期。此额定值基于反复操作（即擦除和编程）每个字节循环的次数。还实现软件冗余配置，确保所有闪存位置都不会达到10,000耐用周期。

用户还可在代码执行时将数据变量写入Flash/EE存储器，例如用以存储电池诊断参数数据。

整个Flash/EE都可以用作用户的代码和非易失性数据存储。在执行ARM代码时，不区分数据和程序空间。Flash/EE存储器的实际宽度为16位，这意味着在ARM模式（32位指令）下每次取指令必须两次访问Flash/EE存储器。当工作速度低于20.48MHz时，Flash/EE存储器控制器可在一个内核时钟周期内透明读取第二个16位半字（32位ARM操作代码的一部分）。因此，速度低于20.48MHz（即CD > 0）时，建议使用ARM模式。对于20.48MHz操作（即CD = 0），建议使用Thumb模式。

Flash/EE的页大小为512个字节。一般情况下，不论CD值多少，Flash/EE控制器擦除一页都需要20 ms。CD = 0、1、2或3时写入一个16位字需要50 μs；CD = 4或5时70 μs；CD = 6时80 μs；而CD = 7时105 μs。

一个16位位置在前后两次擦除中间只能写入两次；即字节操作而非位操作。如果一个位置连续写入两次以上，Flash/EE页内容就有可能被破坏。

Flash/EE存储器在线编程

Flash/EE存储器可通过LIN接口或集成JTAG端口在串行下载模式下进行在线编程。

串行下载（在线编程）

ADuC7036方便通过LIN/BSD引脚下载代码。

JTAG访问

ADuC7036具有一个片内JTAG端口进行代码下载和调试。

ADuC7036 Flash/EE存储器

Flash/EE存储器总容量为96kB，结构为47,000 × 16位。其中，94 kB指定为用户空间，2kB保留用于引导加载程序/内核空间。

Flash/EE控制接口

ADuC7036通过片内存储器控制器来管理Flash/EE存储器的访问和控制。控制器作为两个独立模块（模块0和模块1）来管理Flash/EE存储器。

模块0为32kB，从地址0x00090000映射到地址0x00097FFF，其中模块顶部保留2kB内核空间。

模块1为64kB，从地址0x00080000映射到地址0x0008FFFF。

注意，MCU内核可在某一模块上进行擦除或运行程序的同时在另一个存储模块上继续执行代码。如果在目前执行代码相同的模块上执行一条命令，内核必须暂停操作直到命令结束。这点也同样适用于代码执行。

用户代码、LIN和JTAG编程都利用了Flash/EE控制接口，它包括下列MMR：

- FEExSTA (x = 0或1)：只读寄存器。反映Flash/EE控制接口的状态。
- FEExMOD (x = 0或1)：设置Flash/EE控制接口的工作模式。
- FEExCON (x = 0或1)：8位命令寄存器。命令含义如表13所述。
- FEExDAT (x = 0或1)：16位数据寄存器。
- FEExADR (x = 0或1)：16位地址寄存器。
- FEExSIG (x = 0或1)：由于签名命令启动，保持24位代码签名。
- FEExHID (x = 0或1)：MMR保护。Flash/EE存储器代码空间的读写保护控制。如果先前是通过FEExPRO寄存器配置的，FEExHID可能要求软件密钥来启用访问。
- FEExPRO (x = 0或1)：FEExHID寄存器的缓存。存储FEExHID值并在以后复位和上电事件时自动下载到FEExHID寄存器。

注意，用户软件必须确保Flash/EE控制器在PLL关断前完成任何擦除或写入周期。如果PLL在擦除或写入周期结束前关断，可能会破坏Flash/EE页或字节。

ADuC7036

“FEE0CON和FEE1CON寄存器”部分到“FEE0MOD和FEE1MOD”部分详细介绍每个Flash/EE控制MMR的位分配。

FEE0CON和FEE1CON寄存器

名称：FEE0CON和FEE1CON

地址：0xFFFFF0E8和0xFFFFF0E8

默认值：0x07

访问类型：读写

功能：这些8位寄存器由用户代码写入来控制Flash/EE存储器控制器的模块0（32kB）和模块1（64kB）工作模式。

表13. FEE0CON和FEE1CON中的命令代码

代码	命令	描述 ¹
0x00 ²	保留	保留。此命令不能由用户代码写入。
0x01 ²	单次读取	将FEExADR索引的16位数据载入FEExDAT。
0x02 ²	单次写入	在FEExADR指向的地址处写入FEExDAT。操作耗时50 μs。
0x03 ²	擦除写入	擦除FEExADR索引的页并在FEExADR指向的位置处写入FEExDAT。操作耗时20 ms。
0x04 ²	单次验证	将FEExADR指向的位置内容和FEExDAT内的数据进行比较。比较结果返回到FEExSTA位1或位0。
0x05 ²	单次擦除	擦除FEExADR索引的页。
0x06 ²	批量擦除	擦除用户空间的模块0（32kB）或模块1（64kB）。2kB内核被保护。操作耗时1.2 s。为防止意外执行，执行此指令要求一个命令序列（见“批量擦除命令执行序列”部分）。
0x07		默认命令。
0x08	保留	保留。此命令不能由用户代码写入。
0x09	保留	保留。此命令不能由用户代码写入。
0x0A	保留	保留。此命令不能由用户代码写入。
0x0B	签名	FEE0CON：此命令产生一个基于24位线性反馈移位寄存器（LFSR）的签名，然后载入到FEE0SIG内。如果FEE0ADR低于0x97800，此命令从FEE0ADR内的指定页起向上产生用户代码空间的一个基于24位LFSR的签名，包括内核、安全位和Flash/EE密钥。如果FEE0ADR大于0x97800，表示内核和制造数据已经签名。操作耗时120 μs。FEE1CON：此命令从FEE1ADR起到63,500模块末尾产生一个基于24位LFSR的签名，然后载入到FEE1SIG内。此模块的最后一页不包括在生成签名中。
0x0C	保护	此命令只能运行一次。保存FEExPRO值并只能利用批量擦除（0x06）或软件保护密钥加以清除。
0x0D	保留	保留。此命令不能由用户代码写入。
0x0E	保留	保留。此命令不能由用户代码写入。
0x0F	Ping	无操作，产生中断。

¹ x代表0或1，指定Flash/EE模块0或模块1。

² FEE0CON寄存器在此命令执行完毕后立即读取0x07。

批量擦除命令执行序列

在批量擦除命令签名后，必须执行下列特定代码序列以开始此操作：

FEE_xMOD的位3置1。

将0xFFC3写入FEE_xADR。

将0x3CFF写入FEE_xDAT。

运行在FEE_xCON内的批量擦除命令（代码0x06）。

此序列实例如下：

```
Int a = FEExSTA;           // Ensure FEExSTA is cleared
FEExMOD = 0x08
FEExADR = 0xFFC3
FEExDAT = 0x3CFF
FEExCON = 0x06;           // Mass erase command
while (FEExSTA & 0x04){} //Wait for command to finish
```

注意，通过FEE0CON执行批量擦除命令时，必须禁用低64kB的写保护。即，FEE1HID/FEE1PRO设为0xFFFFFFFF。此设置可通过先取消保护或擦除低64kB来实现。

FEE0STA和FEE1STA寄存器

名称：FEE0STA和FEE1STA

地址：0xFFFF0E00和0xFFFF0E80

默认值：0x20

访问类型：只读

功能：这些8位只读寄存器可由用户代码读取并反映Flash/EE存储器控制器的当前状态。

表14. FEE0STA和FEE1STA MMR位分配

位	描述 ¹
7至4	未使用。全部不用，始终读取0。
3	Flash/EE中断状态位。 发生中断时，即命令结束且FEE _x MOD寄存器的Flash/EE中断使能位置1时，自动置1。 当用户代码读取FEE _x STA寄存器时自动清0。
2	Flash/EE控制器忙指示。 当Flash/EE控制器忙时自动置1。 当控制器不忙时自动清0。
1	命令失败。 当命令写入FEE _x CON不成功时自动置1。 当用户代码读取FEE _x STA寄存器时自动清0。
0	命令成功。 当命令成功完成时由MCU自动置1。 当用户代码读取FEE0STA寄存器时自动清0。

¹ x代表0或1，指定Flash/EE模块0或模块1。

FEE0ADR和FEE1ADR寄存器

名称：FEE0ADR和FEE1ADR

地址：0xFFFF0E10和0xFFFF0E90

默认值：0x0000（FEE1ADR）。对于FEE0ADR，参见“系统标识FEE0ADR”部分。

访问类型：读写

功能：这些16位寄存器决定通过FEE_xCON执行Flash/EE命令时的操作地址。

ADuC7036

FEE0DAT和FEE1DAT寄存器

名称: FEE0DAT和FEE1DAT

地址: 0xFFFFF0E0C和0xFFFFF0E8C

默认值: 0x0000

访问类型: 读写

功能: 此16位寄存器存放Flash/EE存储器的读写数据。

FEE0MOD和FEE1MOD寄存器

名称: FEE0MOD和FEE1MOD

地址: 0xFFFFF0E04和0xFFFFF0E84

默认值: 0x00

访问类型: 读写

功能: 这些寄存器由用户代码写入以配置Flash/EE存储器控制器的工作模式。

表15.FEE0MOD和FEE1MOD MMR位分配

位	描述 ¹
15至7	未使用。全部保留日后使用并应由用户代码写入0。
6, 5	Flash/EE安全锁定位。这些位必须以[6:5] = 10写入以完成Flash/EE安全保护序列。
4	Flash/EE控制器命令完成中断使能。 由用户代码置1以使能Flash/EE控制器在Flash/EE命令完成时产生中断。 清0以在Flash/EE命令完成时禁止产生Flash/EE中断。
3	Flash/EE擦除/写入使能。 由用户代码置1以通过FEE _x CON使能Flash/EE擦除和写入访问。 由用户代码清0以通过FEE _x CON禁用Flash/EE擦除和写入访问。
2	保留。应写入0。
1	Flash/EE控制器中止使能。 由用户代码置1时, 使能Flash/EE控制器中止功能。
0	保留。应写入0。

¹ x代表0或1, 指定Flash/EE模块0或模块1。

FLASH/EE存储器安全性

Flash/EE存储器有94kB供用户使用，可利用FEE0HID和FEE1HID寄存器实现读写保护。

在模块0中，FEE0HID MMR保护30kB。此寄存器的位[0:28]写入保护页0至57。每位保护2页，即1 kB。位[29:30]分别保护页58和页59，即每位写入保护一页512字节。此寄存器的MSB（位31）保护模块0不被JTAG读取。

FEE0PRO寄存器镜像复制FEE0HID MMR的位分配。FEE0PRO MMR允许用户代码锁定Flash/EE存储器的保护或安全配置，这样在以后上电或复位时可自动载入保护配置。这种灵活性允许用户利用FEE0HID MMR临时设置并测试保护设置，随后在将保护系统运送到现场时锁定所需保护配置（利用FEE0PRO）。

在模块1（64kB）中，FEE1HID MMR保护64kB。此寄存器的位[00:29]写入保护页0至119。每位保护4页，即2 kB。位30保护页120至127，即位30写入保护512字节的八页。此寄存器的MSB（位31）保护模块1不被JTAG读取。

如同模块0，FEE1PRO寄存器镜像复制FEE1HID MMR的位分配。FEE1PRO MMR允许用户代码锁定Flash/EE存储器的保护或安全配置，这样在以后上电或复位事件时可自动载入保护配置。

有三种保护级别：临时保护、加密永久性保护和永久性保护。

临时保护

临时保护可通过直接写入FEExHID MMR来设置和取消。此寄存器为易失性，因此保护只在器件上电时才能维持。保护在通电周期后便不再重新载入。

加密永久性保护

加密永久性保护可通过FEExPRO设置以锁定保护配置。一旦保存了所需FEExPRO写入序列开头使用的软件密钥，就必须在以后任何FEExHID或FEExPRO MMR访问中提供。批量擦除可将密钥设回到0xFFFF，但同时也擦除了整个用户代码空间。

永久性保护

永久性保护可通过FEExPRO设置，设置方式类似于加密永久性保护，不同之处在于，所用软件密钥是0xDEADDEAD。FEExPRO写入序列保存后，只有批量擦除可将密钥设回到0xFFFFFFFF。同时也擦除整个用户代码空间。

写入密钥和设置永久性保护的序列

1. 写入和欲保护页相对应的FEExPRO。
2. 将新的（用户定义）32位密钥写入FEExADR位[31:16]和FEExDAT位[15:0]。
3. 将位[6:5] = 0x10写入FEExMOD。
4. 执行FEExCON内的写入密钥命令（代码0x0C）。

要取消或修改保护，可使用相同的序列，同时修改FEExPRO值。

前面写入密钥并设置永久性保护的序列实例如下，此实例写入保护Flash/EE的页4和页5。

```
Int a = FEExSTA;           //Ensure FEExSTA is cleared
FEExPRO = 0 xFFFFFFFB;    //Protect Page 4 and Page 5
FEExADR = 0x66BB;         //32-bit key value (Bits[31:16])
FEExDAT = 0xAA55;         //32-bit key value (Bits[15:0])
FEExMOD = 0x0048          // Lock security sequence
FEExCON = 0x0C;           // Write key command
while (FEExSTA & 0x04){    //Wait for command to
finish
```

ADuC7036

模块 0、Flash/EE 存储器保护寄存器

名称：FEE0HID和FEE0PRO

地址：0xFFFF0E20（用于FEE0HID）和0xFFFF0E1C（用于FEE0PRO）

默认值：0xFFFFFFFF（用于FEE0HID）和 0x00000000（用于FEE0PRO）

访问类型：读写

功能：这些寄存器由用户代码写入以配置Flash/EE存储器保护。

表16. FEE0HID和FEE0PRO MMR位分配

位	描述 ¹
31	读取保护位。 由用户代码置1以允许通过JTAG读取访问来读取32kB Flash/EE模块代码。 由用户代码清0以通过JTAG读取访问来保护32kB Flash/EE模块代码。
30	写入保护位。 由用户代码置1以允许写入页59。 由用户代码清0以写入保护页59。
29	写入保护位。 由用户代码置1以允许写入页58。 由用户代码清0以写入保护页58。
28至0	写入保护位。 由用户代码置1以允许写入30 kB Flash/EE代码存储器的页0至57。每位写入保护2页，每页512字节。 由用户代码清0以写入保护30 kB Flash/EE代码存储器的页0至57。每位写入保护2页，每页512字节。

¹ x代表0或1，指定Flash/EE模块0或模块1。

模块 1、Flash/EE 存储器保护寄存器

名称：FEE1HID和FEE1PRO

地址：0xFFFF0EA0（用于FEE1HID）和0xFFFF0E9C（用于FEE1PRO）

默认值：0xFFFFFFFF（用于FEE1HID）和 0x00000000（用于FEE1PRO）

访问类型：读写

功能：这些寄存器由用户代码写入以配置Flash/EE存储器保护。

表17. FEE1HID和FEE1PRO MMR位分配

位	描述
31	读取保护位。 由用户代码置1以允许通过JTAG读取访问来读取64kB Flash/EE模块代码。 由用户代码清0以通过JTAG读取访问来读取保护64 kB Flash/EE模块代码。
30	写入保护位。写入保护8页。每页512字节。 由用户代码置1以允许写入64 kB Flash/EE代码存储器的页120至127。 由用户代码清0以写入保护64 kB Flash/EE代码存储器的页120至127。
29至0	写入保护位。 由用户代码置1以允许写入64 kB Flash/EE代码存储器的页0至119。每位写入保护4页，每页512字节。 由用户代码清0以写入保护64 kB Flash/EE代码存储器的页0至119。每位写入保护2页，每页512字节。

FLASH/EE存储器可靠性

此器件的Flash/EE存储器阵列完全满足两大Flash/EE存储器关键特性要求：周期耐久性和数据保持力。

耐久性衡量Flash/EE存储器循环重复多个编程、读取及擦除周期的能力。一个耐久性周期包括四个独立连续事件，分别定义为初始页擦除序列、读取/校验序列、字节编程序列和第二个读取/校验序列。

在可靠性指标测试中，在Flash/EE存储器内三页（顶部、中间及底部）的每半个字（16位宽）位置从0x0000到0xFFFF循环操作10,000次。如表1所示，此器件的Flash/EE存储器耐久性指标是根据JEDEC保存期限规格A117测试的。结果表明，随电源和温度变化，最小耐久性参数规格达10,000周期。

保持期衡量Flash/EE存储器长时间保持编程数据的能力。同样，在具体结温（ $T_j = 85^\circ\text{C}$ ）下根据标准JEDEC保持期限规格A117对器件进行测试。作为此测试程序的一部分，如前述，循环操作Flash/EE存储器到达指定的耐久性限值，然后确定数据保持特性。这表明每次对Flash/EE存储器进行重新编程时，都保证Flash/EE存储器在完全指定的保持期限内保持数据。此外，注意根据0.6 eV的活化能，保持期限随JI而递减，如图14所示。

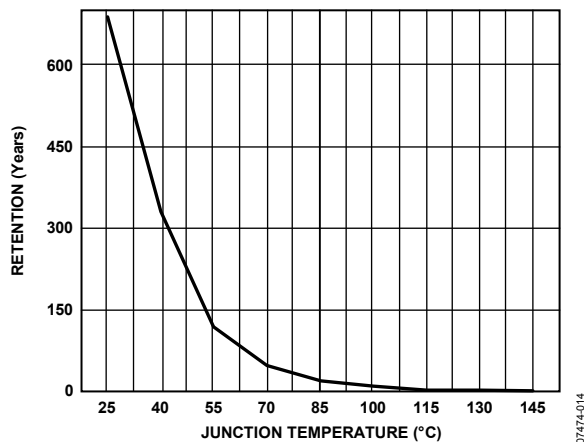


图14. Flash/EE 存储器数据保持期

SRAM和Flash/EE代码执行时间

本节介绍在执行时间关键的应用中SRAM和Flash/EE的执行访问时间。

从SRAM执行

由于SRAM访问时间为2 ns，时钟周期最低为49 ns，SRAM取指令耗费一个时钟周期。但是，当指令涉及到存储器读写时，如果数据位于SRAM内，必须增加一个周期。如果数据位于Flash/EE内，必须增加两个周期：一个用来执行指令，另外两个从Flash/EE中获取32位数据。控制流指令（如转移指令）取指令需要一个周期，用新指令填充流水线需要两个周期。

从Flash/EE执行

Thumb模式指令长16位，需要一个周期来取指令。

当CD = 0时，ARM模式需要两个周期来取32位指令。当CD > 0时，由于Flash/EE存储器仍以全速时钟速率工作，不需要任何其他取周期。此外，对于任何CD位值，访问数据前需要一些死区。

当执行指令涉及使用Flash/EE作为数据存储时，在两种模式下的时序完全相同。如果欲执行的指令为控制流指令，那么需要一个额外的周期来对程序计数器的新地址进行解码，然后在CD = 0时需要四个周期填充流水线。

数据处理指令如果只涉及内核寄存器则不需要任何额外时钟周期。数据传送指令更加复杂，见表18概述。

表18. ARM/Thumb模式的典型执行周期

指令	取周期	死区	数据访问
LD	2/1	1	2
LDH	2/1	1	1
LDM/PUSH	2/1	N	$2 \times N$
STR	2/1	1	$2 \times 50 \mu\text{s}$
STRH	2/1	1	$50 \mu\text{s}$
STRM/POP	2/1	N	$2 \times N \times 50 \mu\text{s}$

$1 < N \leq 16$, N是在多个载入/存储指令中欲载入或存储的数据数目。

默认情况下，在任何Flash/EE擦除或写入周期期间，会暂停Flash/EE代码执行。页（512字节）擦除周期需要20ms而（16位）字写入命令需要50 μs 。但是，在当前Flash/EE擦除/写入周期期间，如果ARM内核接收到有效中断，Flash/EE控制器允许中止擦除/写入周期。因此，ARM7能够立即进入中断服务，然后重新返回Flash/EE命令。典型中止操作需要10个时钟周期。如果无法执行中止操作，用户可以执行SRAM的Flash/EE程序代码和相关中断例行程序，允许内核立即进入中断服务。

片上内核

ADuC7036在顶部2 kB Flash/EE代码空间内常驻有一个片上内核。发生任何复位事件后，此内核会将工厂校准数据从制造数据空间复制到不同片内外设内。内核校准的外设如下：

- 电源监控器（PSM）
- 精密振荡器
- 低功耗振荡器
- REG_AVDD/REG_DVDD
- 低功耗基准电压源
- 正常模式基准电压源
- 电流ADC（失调和增益）
- 电压/温度ADC（失调和增益）

可以被内核修改并不同于POR默认值的用户MMR如下：

- R0至R15
- GP0CON/GP2CON
- SYSCHK
- ADCMDE/ADC0CON
- FEE0ADR/FEE0CON/FEE0SIG
- HVDAT/HVCON
- HVCFG0/HVCFG1
- T3LD

ADuC7036还附带一个片内LIN下载器。

内核执行流程图如图15所示。如表99所述，当前内核版本可从SYSSER1获得。

上电复位后，一旦内核代码激活，便会禁用看门狗定时器。在内核执行期间，看门狗定时器有效，超时周期达500ms。这确保内核发生错误时，ADuC7036会自动复位。在发生任何其他复位后，看门狗定时器维持内核周期的用户代码配置并紧接在内核退出前刷新。正常运行LIN下载器需要最低30 ms的看门狗周期。进入LIN下载模式后，会定期刷新看门狗。

正常内核执行时间（不包括LIN下载）约为5 ms。LIN下载模式的进出只能通过复位完成。

在正常内核执行期间不会修改SRAM；而在LIN下载内核执行期间会修改。

注意，即使NTRST = 0，除非地址0x14包含0x27011970或页0的校验和（不包括地址0x14），否则就不会执行用户代码。如果地址0x14不包含这些信息，就会无法执行用户代码而转入LIN下载模式。内核执行期间会禁止JTAG访问。

当NTRST = 1时，始终执行用户代码。

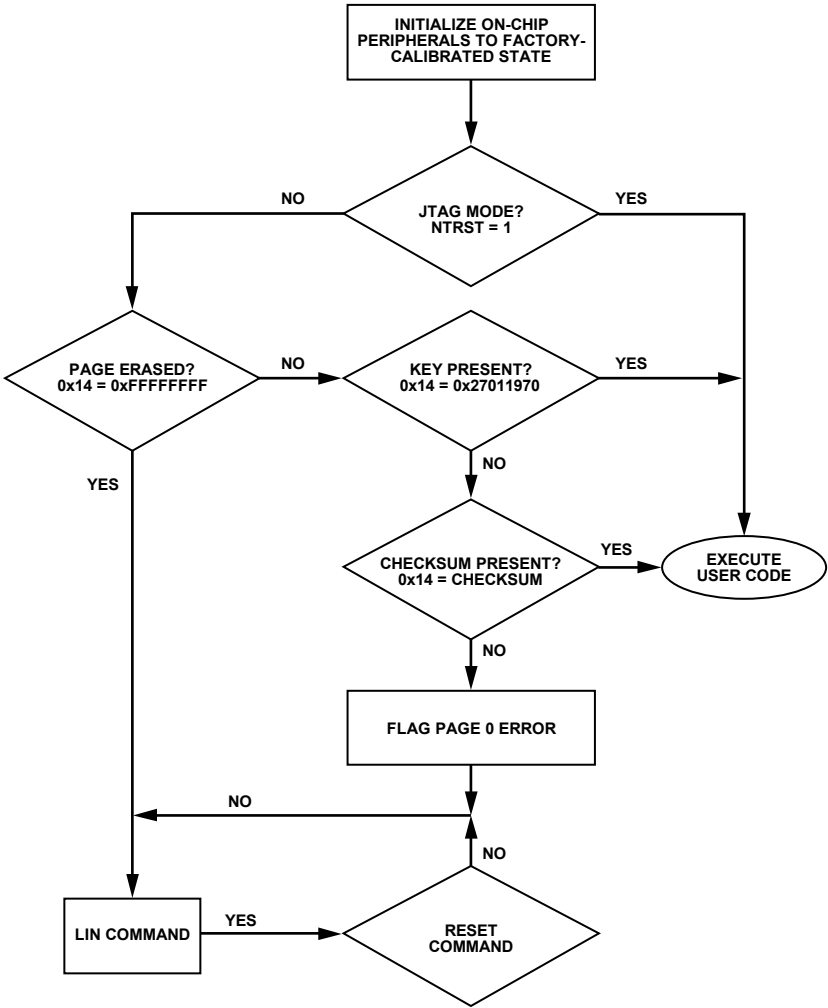


图 15. 内核流程图

07474-015

ADuC7036

存储器映射寄存器

存储器映射寄存器（MMR）空间映射到存储器阵列的顶部4 kB，并且可以通过ARM7寄存器组的间接寻址、载入和存储命令来加以访问。ADuC7036寄存器映射图如图16所示。

MMR空间为CPU和所有片内外设提供接口。除ARM7内核寄存器（参见“ARM寄存器”部分）外，所有寄存器都常驻于MMR区域内。

如“完整MMR列表”部分中表19至30所示，MMR数据宽度为1个字节（8位）到4个字节（32位）。ARM7内核可32位读写访问任何MMR（单字节或多字节宽度寄存器）。

例如，如“ARM寄存器”部分所述，读取结果采用从小到大顺序格式排列。但如果ARM7 16位访问4字节（32位）MMR，就会导致错误。在16位写入32位MMR的情况下，16个最高有效位（高16位）全部被写入0。在16位读取32位MMR的情况下，只能读取MMR位中的16位。

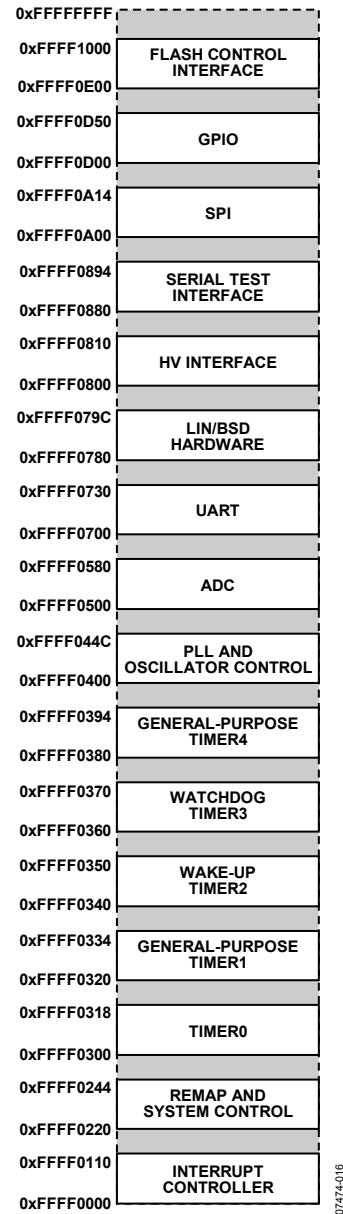


图16. 顶层MMR映射图

完整MMR列表

表19至30以十六进制代码形式列出地址。访问类型：R表示读取，W表示写入，RW表示读写。

表19. IRQ基地址 = 0xFFFF0000

地址	名称	字节	访问类型	默认值	描述
0x0000	IRQSTA	4	R	0x00000000	有效IRQ源。参见“中断系统”部分和表50。
0x0004	IRQSIG ¹	4	R	N/A	所有IRQ源的当前状态（使能和禁用）。参见“中断系统”部分和表50。
0x0008	IRQEN	4	RW	0x00000000	使能IRQ源。参见“中断系统”部分和表50。
0x000C	IRQCLR	4	W	N/A	MMR禁用IRQ源。参见“中断系统”部分和表50。
0x0010	SWICFG	4	W	N/A	软件中断配置MMR。参见“可编程中断”部分和表51。
0x0100	FIQSTA	4	R	0x00000000	有效IRQ源。参见“中断系统”部分和表50。
0x0104	FIQSIG ¹	4	R	N/A	所有IRQ源的当前状态（使能和禁用）。参见“中断系统”部分和表50。
0x0108	FIQEN	4	RW	0x00000000	使能IRQ源。参见“中断系统”部分和表50。
0x010C	FIQCLR	4	W	N/A	MMR禁用IRQ源。参见“中断系统”部分和表50。

¹ 取决于外部中断引脚（GPIO_0、GPIO_5、GPIO_7和GPIO_8）上的电平。

表20. 系统控制基地址 = 0xFFFF0200

地址	名称	字节	访问类型	默认值	描述
0x0220	SYSMAP0	1	RW	N/A	重映射控制寄存器。参见“重映射操作”部分和表10。
0x0230	RSTSTA	1	RW	因复位类型而异	复位状态MMR。参见“复位”部分和表11及12。
0x0234	RSTCLR	1	W	N/A	RSTSTA清0MMR。参见“复位”部分和表11及12。
0x0238	SYSSER0 ¹	4	RW	N/A	系统序列号0。详情参见“器件标识”部分和表98。
0x023C	SYSSER1 ¹	4	RW	N/A	系统序列号1。详情参见“器件标识”部分和表99。
0x0560	SYSALI ¹	4	R	N/A	系统组装批次ID。详情参见“器件标识”部分。
0x0240	SYSCHK ¹	4	RW	N/A	内核校验和。参见“系统内核校验和”部分。

¹ 由内核更新。

表21. 定时器基地址 = 0xFFFF0300

地址	名称	字节	访问类型	默认值	描述
0x0300	T0LD	2	RW	0x0000	定时器0载入寄存器。参见“定时器0—使用寿命定时器”和“定时器0载入寄存器”部分。
0x0304	T0VAL0	2	R	0x0000	定时器0值寄存器0。参见“定时器0—使用寿命定时器”和“定时器0值寄存器”部分。
0x0308	T0VAL1	4	R	0x00000000	定时器0值寄存器1。参见“定时器0—使用寿命定时器”和“定时器0值寄存器”部分。
0x030C	T0CON	4	RW	0x00000000	定时器0控制MMR参见“定时器0—使用寿命定时器”和“定时器0控制寄存器”部分。
0x0310	T0CLRI	1	W	N/A	定时器0中断清0寄存器。参见“定时器0—使用寿命定时器”和“定时器0载入寄存器”部分。
0x0314	T0CAP	2	R	0x0000	定时器0捕捉寄存器。参见“定时器0—使用寿命定时器”和“定时器0捕捉寄存器”部分。
0x0320	T1LD	4	RW	0x00000000	定时器1载入寄存器。参见“定时器±1”和“定时器1载入寄存器”部分。
0x0324	T1VAL	4	R	0xFFFFFFFF	定时器1值寄存器。参见“定时器±1”和“定时器1值寄存器”部分。
0x0328	T1CON	4	RW	0x01000000	定时器1控制MMR。参见“定时器±1”和“定时器1控制寄存器”部分。
0x032C	T1CLRI	1	W	N/A	定时器1中断清0寄存器。参见“定时器±1”和“定时器1清0寄存器”部分。
0x0330	T1CAP	4	R	0x00000000	定时器1捕捉寄存器。参见“定时器±1”和“定时器1捕捉寄存器”部分。

ADuC7036

地址	名称	字节	访问类型	默认值	描述
0x0340	T2LD	4	RW	0x00000000	定时器2载入寄存器。参见“定时器2—唤醒定时器”和“定时器2载入寄存器”部分。
0x0344	T2VAL	4	R	0xFFFFFFFF	定时器2值寄存器。参见“定时器2—唤醒定时器”和“定时器2值寄存器”部分。
0x0348	T2CON	2	RW	0x0000	定时器2控制MMR。参见“定时器2—唤醒定时器”和“定时器2控制寄存器”部分和表55。
0x034C	T2CLR1	1	W	N/A	定时器2中断清0寄存器。参见“定时器2—唤醒定时器”和“定时器2清0寄存器”部分。
0x0360	T3LD	2	RW	0x0040	定时器3载入寄存器。参见“定时器3—看门狗定时器”和“定时器3载入寄存器”部分。
0x0364	T3VAL	2	R	0x0040	定时器3值寄存器。参见“定时器3—看门狗定时器”和“定时器3值寄存器”部分。
0x0368	T3CON	2	RW	0x0000	定时器3控制MMR。参见“定时器3—看门狗定时器、3值寄存器”和“定时器3控制寄存器”部分及表56。
0x036C	T3CLR1 ¹	1	W	N/A	定时器3中断清0寄存器。参见“定时器3—看门狗定时器”和“定时器3清0寄存器”部分。
0x0380	T4LD	2	RW	0x0000	定时器4载入寄存器。参见“定时器4—STI定时器”和“定时器4载入寄存器”部分。
0x0384	T4VAL	2	R	0xFFFF	定时器4值寄存器。参见“定时器4—STI定时器”和“定时器4值寄存器”部分。
0x0388	T4CON	4	RW	0x00000000	定时器4控制MMR。参见“定时器4—STI定时器”和“定时器4控制寄存器”部分及表57。
0x038C	T4CLR1	1	W	N/A	定时器4中断清0寄存器。参见“定时器4—STI定时器”和“定时器4清0寄存器”部分。
0x0390	T4CAP	2	R	0x0000	定时器4捕捉寄存器。参见“定时器4—STI定时器”部分。

¹ 由内核更新。

表22. PLL基地址 = 0xFFFF0400

地址	名称	字节	访问类型	默认值	描述
0x0400	PLLSTA	1	R	N/A	PLL状态MMR。参见“PLLSTA寄存器”部分和表44。
0x0404	POWKEY0	4	W	N/A	POWCON预写密钥。参见“POWCON预写密钥”部分。
0x0408	POWCON	1	RW	0x79	功率控制和内核速度控制寄存器。参见“POWCON寄存器”部分。
0x040C	POWKEY1	4	W	N/A	POWCON写后密钥。参见“POWCON写后密钥”部分。
0x0410	PLLKEY0	4	W	N/A	PLLCON预写密钥。参见“PLLCON预写密钥”部分。
0x0414	PLLCON	1	RW	0x00	PLL时钟源选择MMR。参见“PLLCON寄存器”部分。
0x0418	PLLKEY1	4	W	N/A	PLLCON写后密钥。参见“PLLCON写后密钥”部分。
0x042C	OSC0TRM	1	RW	0xX8	低功耗振荡器调整位MMR。参见“OSC0TRM寄存器”部分。
0x0440	OSC0CON	1	RW	0x00	低功耗振荡器校准控制MMR。参见“OSC0CON寄存器”部分。
0x0444	OSC0STA	1	R	0x00	低功耗振荡器校准状态MMR。参见“OSC0STA寄存器”部分。
0x0448	OSC0VAL0	2	R	0x0000	低功耗振荡器校准计数器0 MMR。参见“OSC0VAL0寄存器”部分。
0x044C	OSC0VAL1	2	R	0x0000	低功耗振荡器校准计数器1 MMR。参见“OSC0VAL1寄存器”部分。

表23. ADC基地址 = 0xFFFF0500

地址	名称	字节	访问类型	默认值	描述
0x0500	ADCSTA	2	R	0x0000	ADC状态MMR。参见“ADC状态寄存器”部分和表35。
0x0504	ADCMSKI	1	RW	0x00	ADC中断源使能MMR。参见“ADC中断屏蔽寄存器”部分。
0x0508	ADCMDDE	1	RW	0x00	ADC模式寄存器。参见“ADC模式寄存器”部分和表36。
0x050C	ADC0CON	2	RW	0x0000	电流ADC控制MMR。参见“电流通道ADC控制寄存器”部分和表37。
0x0510	ADC1CON	2	RW	0x0000	V-/T-ADC控制MMR。参见“电压/温度通道ADC控制寄存器”部分和表38。
0x0518	ADCFLT	2	RW	0x0007	ADC滤波器控制MMR。参见“ADC滤波器寄存器”部分和表39。
0x051C	ADCCFG	1	RW	0x00	ADC配置MMR。参见“ADC配置寄存器”部分和表42。
0x0520	ADC0DAT	2	R	0x0000	电流ADC结果MMR。参见“电流通道ADC数据寄存器”部分。
0x0524	ADC1DAT	2	R	0x0000	V-ADC结果MMR。参见“电压通道ADC数据寄存器”部分。
0x0528	ADC2DAT	2	R	0x0000	T-ADC结果MMR。参见“温度通道ADC数据寄存器”部分。
0x0530	ADC0OF ¹	2	RW	N/A	电流ADC失调MMR。参见“电流通道ADC失调校准寄存器”部分。
0x0534	ADC1OF ¹	2	RW	N/A	电压ADC失调MMR。参见“电压通道ADC失调校准寄存器”部分。
0x0538	ADC2OF ¹	2	RW	N/A	温度ADC失调MMR。参见“温度通道ADC失调校准寄存器”部分。
0x053C	ADC0GN ¹	2	RW	N/A	电流ADC增益MMR。参见“电流通道ADC增益校准寄存器”部分。
0x0540	ADC1GN ¹	2	RW	N/A	电压ADC增益MMR。参见“电压通道ADC增益校准寄存器”部分。
0x0544	ADC2GN ¹	2	RW	N/A	温度ADC增益MMR。参见“温度通道ADC增益校准寄存器”部分。
0x0548	ADC0RCL	2	RW	0x0001	电流ADC结果计数限值。参见“电流通道ADC结果计数器限值寄存器”部分。
0x054C	ADC0RCV	2	R	0x0000	电流ADC结果计数值。参见“电流通道ADC结果计数寄存器”部分。
0x0550	ADC0TH	2	RW	0x0000	电流ADC结果阈值。参见“电流通道ADC阈值寄存器”部分。
0x0554	ADC0TCL	1	RW	0x01	电流ADC结果阈值计数限值。参见“电流通道ADC阈值计数限值寄存器”部分。
0x0558	ADC0THV	1	R	0x00	电流ADC结果阈值计数限值。参见“电流通道ADC阈值计数寄存器”部分。
0x055C	ADC0ACC	4	R	0x00000000	电流ADC结果累加器。参见“电流通道ADC累加器寄存器”部分。
0x057C	ADCREF ¹	2	RW	N/A	低功耗模式基准电压源比例因子。参见“低功耗基准电压源比例因子”部分。

¹ 由内核更新。

ADuC7036

表24. UART基地址 = 0xFFFF0700

地址	名称	字节	访问类型	默认值	描述
0x0700	COMTX	1	W	N/A	UART发送寄存器。参见“UART Tx寄存器”部分。
	COMRX	1	R	0x00	UART接收寄存器。参见“UART Rx寄存器”部分。
	COMDIV0	1	RW	0x00	UART标准波特率发生器除数0。参见“UART除数锁存寄存器0”部分。
0x0704	COMIEN0	1	RW	0x00	UART中断使能MMR 0。参见“UART中断使能寄存器0”部分和表84。
	COMDIV1	1	RW	0x00	UART标准波特率发生器除数1。参见“UART除数锁存寄存器1”部分。
0x0708	COMIID0	1	R	0x01	UART中断标识0。参见“UART中断标识寄存器0”部分和表85。
0x070C	COMCON0	1	RW	0x00	UART控制寄存器0。参见“UART控制寄存器0”部分和表81。
0x0710	COMCON1	1	RW	0x00	UART控制寄存器1。参见“UART控制寄存器1”部分和表82。
0x0714	COMSTA0	1	R	0x60	UART状态寄存器0。参见“UART状态寄存器0”部分和表83。
0X072C	COMDIV2	2	RW	0x0000	UART小数分频器MMR。参见“UART小数分频器寄存器”部分和表86。

表25. LIN硬件同步基地址 = 0xFFFF0780

地址	名称	字节	访问类型	默认值	描述
0x0780	LHSSTA	4	R	0x00000000	LHS状态MMR。参见“LIN硬件同步状态寄存器”部分和表92。
0x0784	LHSCON0	2	RW	0x0000	LHS控制MMR 0。参见“LIN硬件同步控制寄存器0”部分和表93。
0x0788	LHSVAL0	2	R	0x0000	LHS定时器0 MMR。参见“LIN硬件同步定时器0寄存器”部分。
0x078C	LHSCON1	1	RW	0x32	LHS控制MMR 1。参见“LIN硬件同步控制寄存器1”部分和表94。
0x0790	LHSVAL1	2	RW	0x0000	LHS定时器1 MMR。参见“LIN硬件同步中断定时器1寄存器”部分。
0x0794	LHSCAP	2	R	0x0000	LHS捕捉MMR。参见“LIN硬件同步捕捉寄存器”部分。
0x0798	LHSCMP	2	RW	0x0000	LHS比较MMR。参见“LIN硬件同步比较寄存器”部分。

表26. 高压接口基地址 = 0xFFFF0800

地址	名称	字节	访问类型	默认值	描述
0x0804	HVCON	1	RW	N/A	高压接口控制MMR。参见“高压接口控制寄存器”部分和表71及72。
0x080C	HVDAT	2	RW	N/A	高压接口数据MMR。参见“高压数据寄存器”部分和表73。

表27. STI基地址 = 0xFFFF0880

地址	名称	字节	访问类型	默认值	描述
0x0880	STIKEY0	4	W	N/A	STICON预写密钥。参见“串行测试接口密钥0寄存器”部分。
0x0884	STICON	2	RW	0x0000	串行测试接口控制MMR。参见“串行测试接口控制寄存器”部分和表91。
0x0888	STIKEY1	4	W	N/A	STICON写后密钥。参见“串行测试接口密钥1寄存器”部分。
0x088C	STIDAT0	2	RW	0x0000	STI数据MMR 0。参见“串行测试接口数据0寄存器”部分。
0x0890	STIDAT1	2	RW	0x0000	STI数据MMR 1。参见“串行测试接口数据1寄存器”部分。
0x0894	STIDAT2	2	RW	0x0000	STI数据MMR 2。参见“串行测试接口数据2寄存器”部分。

表28. SPI基地址 = 0xFFFF0A00

地址	名称	字节	访问类型	默认值	描述
0x0A00	SPISTA	1	R	0x00	SPI状态MMR。参见“SPI状态寄存器”部分和表90。
0x0A04	SPIRX	1	R	0x00	SPI接收MMR。参见“SPI接收寄存器”部分。
0x0A08	SPI TX	1	W	N/A	SPI发送MMR。参见“SPI发送寄存器”部分。
0x0A0C	SPI DIV	1	RW	0x1B	SPI波特率选择MMR。参见“SPI分频器寄存器”部分。
0x0A10	SPICON	2	RW	0x0000	SPI控制MMR。参见“SPI控制寄存器”部分和表89。

表29. GPIO基地址 = 0xFFFF0D00

地址	名称	字节	访问类型	默认值	描述
0x0D00	GP0CON	4	RW	0x11100000	GPIO端口0控制MMR。参见“GPIO端口0控制寄存器”部分和表59。
0x0D04	GP1CON	4	RW	0x10000000	GPIO端口1控制MMR。参见“GPIO端口1控制寄存器”部分和表60。
0x0D08	GP2CON	4	RW	0x01000000	GPIO端口2控制MMR。参见“GPIO端口2控制寄存器”部分和表61。
0x0D20	GP0DAT ¹	4	RW	0x000000XX	GPIO端口0数据控制MMR。参见“GPIO端口0数据寄存器”部分和表62。
0x0D24	GP0SET	4	W	N/A	GPIO端口0数据设置MMR。参见“GPIO端口0设置寄存器”部分和表65。
0x0D28	GP0CLR	4	W	N/A	GPIO端口0数据清0MMR。参见“GPIO端口0清0寄存器”部分和表68。
0x0D30	GP1DAT ¹	4	RW	0x000000XX	GPIO端口1数据控制MMR。参见“GPIO端口1数据寄存器”部分和表63。
0x0D34	GP1SET	4	W	N/A	GPIO端口1数据设置MMR。参见“GPIO端口1设置寄存器”部分和表66。
0x0D38	GP1CLR	4	W	N/A	GPIO端口1数据清0MMR。参见“GPIO端口1清0寄存器”部分和表69。
0x0D40	GP2DAT ¹	4	RW	0x000000XX	GPIO端口2数据控制MMR。参见“GPIO端口2数据寄存器”部分和表64。
0x0D44	GP2SET	4	W	N/A	GPIO端口2数据设置MMR。参见“GPIO端口2设置寄存器”部分和表67。
0x0D48	GP2CLR	4	W	N/A	GPIO端口2数据清0MMR。参见“GPIO端口2清0寄存器”部分和表70。

¹ 取决于外部GPIO引脚上的电平。

ADuC7036

表30. Flash/EE基地址 = 0xFFFF0E00

地址	名称	字节	访问类型	默认值	描述
0x0E00	FEE0STA	1	R	0x20	Flash/EE状态MMR。
0x0E04	FEE0MOD	1	RW	0x00	Flash/EE控制MMR。
0x0E08	FEE0CON	1	RW	0x07	Flash/EE控制MMR。参见表13。
0x0E0C	FEE0DAT	2	RW	0x0000	Flash/EE数据MMR。
0x0E10	FEE0ADR	2	RW	非零	Flash/EE地址MMR。
0x0E18	FEE0SIG	3	R	0xFFFFFFFF	Flash/EE LFSR MMR。
0x0E1C	FEE0PRO	4	RW	0x00000000	Flash/EE保护MMR。参见“Flash/EE存储器安全性”部分和表16。
0x0E20	FEE0HID	4	RW	0xFFFFFFFF	Flash/EE保护MMR。参见“Flash/EE存储器安全性”部分和表16。
0x0E80	FEE1STA	1	R	0x20	Flash/EE状态MMR。
0x0E84	FEE1MOD	1	RW	0x00	Flash/EE控制MMR。
0x0E88	FEE1CON	1	RW	0x07	Flash/EE控制MMR。参见表13。
0x0E8C	FEE1DAT	2	RW	0x0000	Flash/EE数据MMR。
0x0E90	FEE1ADR	2	RW	0x0000	Flash/EE地址MMR。
0x0E98	FEE1SIG	3	R	0xFFFFFFFF	Flash/EE LFSR MMR。
0x0E9C	FEE1PRO	4	RW	0x00000000	Flash/EE保护MMR。参见“Flash/EE存储器安全性”部分和表17。
0x0EA0	FEE1HID	4	RW	0xFFFFFFFF	Flash/EE保护MMR。参见“Flash/EE存储器安全性”部分和表17。

16位 Σ - Δ ADC

ADuC7036内置了两个独立的 Σ - Δ ADC：电流通道ADC（I-ADC）和电压/温度通道ADC（V-/T-ADC）。这些精确测量通道集成了片内缓冲、可编程增益放大器、16位 Σ - Δ 调制器和数字滤波以在12V汽车电池系统中精确测量电流、电压和温度变量。

电流通道ADC（I-ADC）

I-ADC转换外部100 $\mu\Omega$ 分流电阻感应的电池电流。片内可编程增益意味着，I-ADC可配置处理 ± 1 A至 ± 1500 A的电池电流量。

如图17所示，I-ADC采用 Σ - Δ 转换技术来获得16位无失码性能。

Σ - Δ 调制器将采样输入信号转换成占空比包含数字信息的数字脉冲序列。然后使用一个修正Sinc3可编程、低通滤波器来抽取调制器输出数据流以提供16位数据转换结果，在正常模式下可编程输出速率在4 Hz至8 kHz范围内而在低功耗模式下在1 Hz至2 kHz范围内。

I-ADC还内置了计数器、比较器和累加器逻辑，从而允许I-ADC结果在预定转换数目或I-ADC结果超过可编程阈值后产生中断。同时支持快速ADC超量程功能。使能此功能后，32位累加器会自动对16位I-ADC结果求和。

在电流通道上获得第一个有效（完全稳定）结果的时间为斩波模式禁用的三个ADC转换周期和斩波使能的两个ADC转换周期。

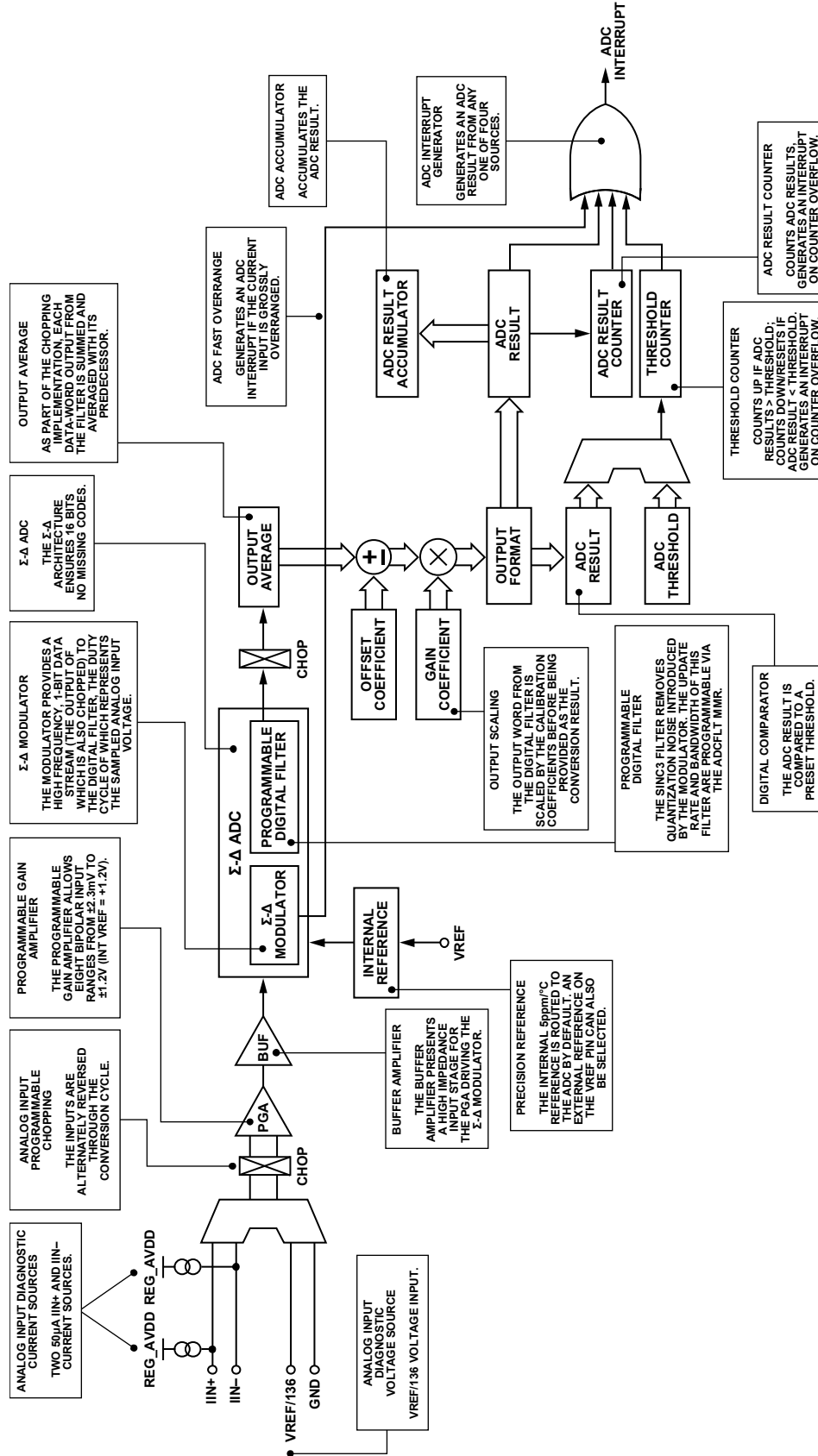


图 17. 电流 ADC 顶层概览

电压/温度通道ADC (V-/T-ADC)

电压/温度通道ADC (V-/T-ADC) 转换电压和温度等其他电池参数。此通道输入可多路复用三个输入源中的一个：外部电压、外部温度传感器电路或片内温度传感器。

和电流通道ADC (I-ADC)一样，V-/T-ADC运用了相同的 Σ - Δ 转换技术，包括一个修正Sinc3低通滤波器以在4Hz至8kHz的可编程输出速率下提供16位数据转换结果。由于已在电压通道内实现，不需要外部RC滤波器网络。

外部电池电压 (VBAT) 通过一个片内、高压 (24分频) 电阻衰减器送到ADC输入端。选择电压衰减器输入端会自动使能电压衰减器缓冲。

电池温度可通过片内温度传感器或外部温度传感器输入来导出。在电压/温度通道上切换输入通道后获得第一个有效 (完全稳定) 结果的时间为斩波模式禁用的三个ADC转换周期。

此ADC同样经过缓冲，但不同于电流通道，在VTEMP上具有0 V至 V_{REF} 和在VBAT上0 V至28.8 V的固定输入范围 (假定内部基准电压为1.2 V)。信号链顶层概览如图18所示。

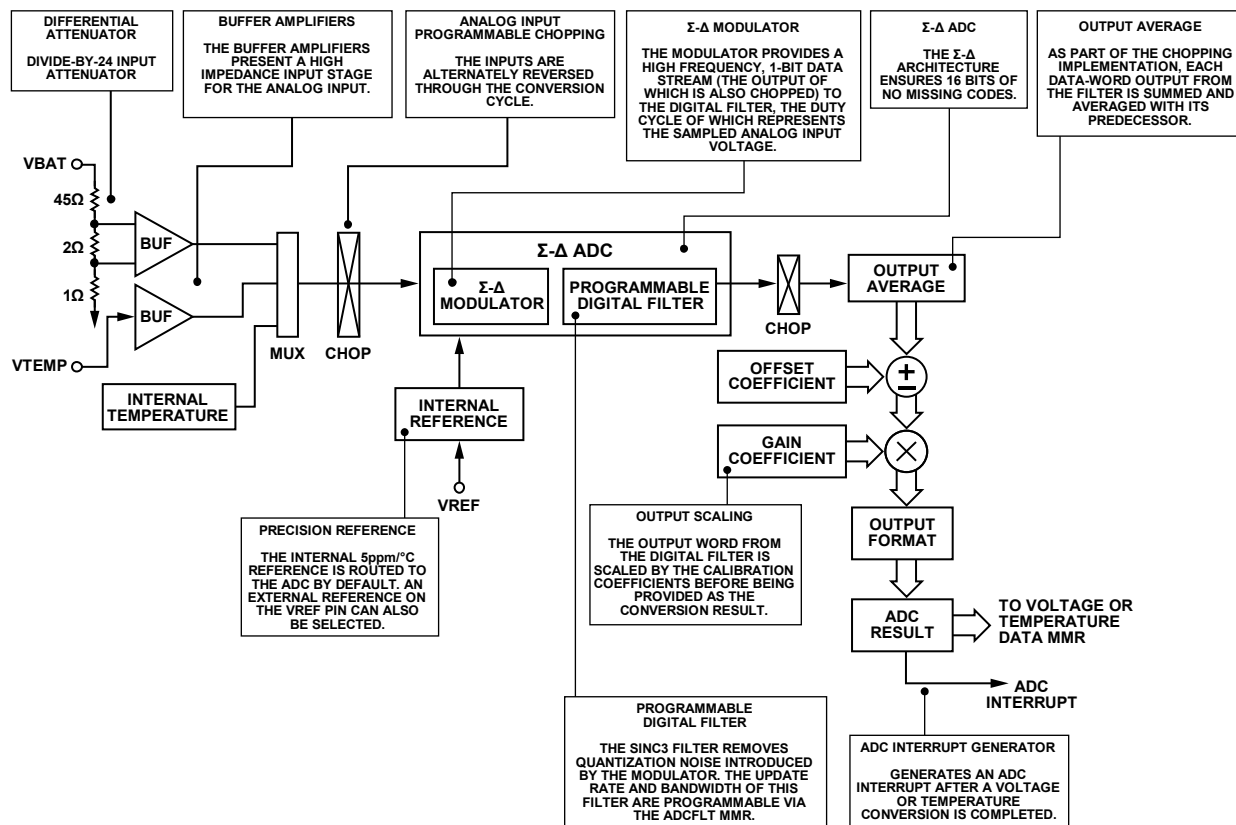


图 18. 电压/温度 ADC 顶层概览

ADuC7036

ADC接地开关

ADuC7036具有一个集成的接地开关引脚GND_SW（引脚15）。通过这个开关，用户可以将外部设备与接地动态断开，代之使用直接接地或20 kΩ电阻接地。这个附加电阻可以减少NTC电路的外部元件数目。接地开关功能有利于降低特定应用电路板的功耗。

应用实例如图19所示。

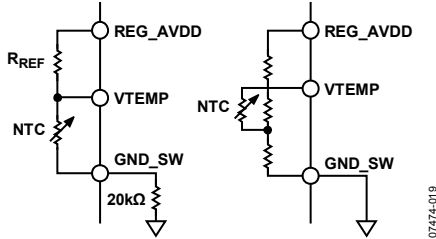


图19. 外部温度传感器电路实例

图19显示在两种模式下连接的外部NTC：使用内部20 kΩ电阻和通过GND_SW直接接地。

如图20所示，ADCCFG[7]控制接地开关的接地，而ADCMDE[6]控制GND_SW电阻。

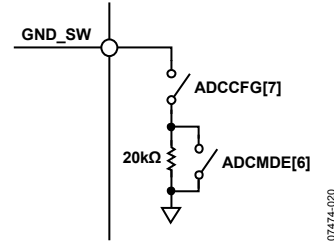


图20. 内部接地开关配置

ADCCFG[7]和ADCMDE[6]的所有可能组合如表31所示。

表31. GND_SW配置

ADCCFG[7]	ADCMDE[6]	GND_SW
0	0	悬空
0	1	悬空
1	0	直接接地
1	1	通过20 kΩ电阻接地

ADC噪声性能表

表32、33及34列出了I-ADC和V-/T-ADC在某些典型输出更新速率下的输出均方根噪声（单位毫伏）。这些数据都为典型值并在差分输入电压0V时产生。输出均方根噪声是指在ADC直流输入信号时输出代码分布的标准差（或 1σ ）。以毫伏均方根表示（ $\mu\text{V rms}$ ）。

表32. 电流通道ADC在正常功耗模式下的典型输出均方根噪声

ADCFLT	数据更新速率	ADC输入范围									
		$\pm 2.3\text{ mV}$ (512)	$\pm 4.6\text{ mV}$ (256)	$\pm 4.68\text{ mV}$ (128)	$\pm 18.75\text{ mV}$ (64)	$\pm 37.5\text{ mV}$ (32)	$\pm 75\text{ mV}$ (16)	$\pm 150\text{ mV}$ (8)	$\pm 300\text{ mV}$ (4 ¹)	$\pm 600\text{ mV}$ (2 ¹)	$\pm 1.2\text{ V}$ (1 ¹)
0xBF1D	4 Hz	0.040 μV	0.040 μV	0.043 μV	0.045 μV	0.087 μV	0.175 μV	0.35 μV	0.7 μV	1.4 μV	2.8 μV
0x961F	10 Hz	0.060 μV	0.060 μV	0.060 μV	0.065 μV	0.087 μV	0.175 μV	0.35 μV	0.7 μV	1.4 μV	2.8 μV
0x007F	50 Hz	0.142 μV	0.142 μV	0.144 μV	0.145 μV	0.170 μV	0.305 μV	0.380 μV	0.7 μV	2.3 μV	2.8 μV
0x0007	1 kHz	0.620 μV	0.620 μV	0.625 μV	0.625 μV	0.770 μV	1.310 μV	1.650 μV	2.520 μV	7.600 μV	7.600 μV
0x0000	8 kHz	2.000 μV	2.000 μV	2.000 μV	2.000 μV	2.650 μV	4.960 μV	8.020 μV	15.0 μV	55.0 μV	55.0 μV

¹ 最大绝对输入电压允许范围：对地-200 mV至+300 mV。

表33. 电压通道ADC的典型输出均方根噪声（折合到ADC电压衰减器输入）

ADCFLT	数据更新速率	ADC输入范围28.8 V
0xBF1D	4 Hz	65 μV
0x961F	10 Hz	65 μV
0x0007	1 kHz	180 μV
0x0000	8 kHz	1600 μV

表34. 温度通道ADC的典型输出均方根噪声

ADCFLT	数据更新速率	ADC输入范围0 V至1.2 V
0xBF1D	4 Hz	2.8 μV
0x961F	10 Hz	2.8 μV
0x0007	1 kHz	7.5 μV
0x0000	8 kHz	55 μV

ADC MMR接口

ADC利用多个MMR实现控制和配置，详见“ADC状态寄存器”部分至“低功耗基准电压源比例因子”部分。

ADCSTA MMR所有高八位（位[8:15]）只定义为标志位，不产生中断。MMR低八位（位[0:7]）的所有位为逻辑或向MCU内核产生一个ADC中断。为了响应ADC中断，用户代码将会查询ADCSTA MMR以确定中断源。如“ADC中断屏蔽寄存器”部分所述，每个ADC中断源都可以通过ADCMSKI MMR单独屏蔽。

所有ADC结果就绪位在ADC0DAT MMR读取后即被清0。如果电流通道ADC没有使能，那么所有ADC结果就绪位在读取ADC1DAT或ADC2DAT MMR后被清0。

为了确保I-ADC和V-/T-ADC转换数据能够同步，用户代码应先读取ADC1DAT MMR，然后再读取ADC0DAT MMR。

只有先清0对应ADC结果就绪位，才能将新ADC转换结果写入到ADCxDAT MMR。这条规则的唯一例外情况是，数据转换结果正好在ARM内核断电时更新。在此模式下，即使没有清0就绪位，ADCxDAT寄存器仍始终保存最近一次的ADC转换结果。

ADC状态寄存器

名称：ADCSTA

地址：0xFFFF0500

默认值：0x0000

访问类型：只读

功能：该只读寄存器保持ADC工作模式或电流状态相关的一般状态信息。

表35. ADCSTA MMR位分配

位	描述
15	ADC校准状态。 硬件自动置1，指示ADC校准周期已完成。ADCMDE写入后清0。
14	ADC温度转换错误。 硬件自动置1，指示发生温度转换超量程或欠量程。此时将转换结果箝位至负满量程（欠量程错误）或正满量程（超量程错误）。有效（量程内）温度转换结果写入ADC2DAT寄存器后清0。
13	ADC电压转换错误。 硬件自动置1，指示发生电压转换超量程或欠量程。此时将转换结果箝位至负满量程（欠量程错误）或正满量程（超量程错误）。将有效（量程内）电压转换结果写入ADC1DAT寄存器后清0。
12	ADC电流转换错误。 硬件自动置1，指示发生电压转换超量程或欠量程。此时将转换结果箝位至负满量程（欠量程错误）或正满量程（超量程错误）。将有效（量程内）电流转换结果写入ADC0DAT寄存器后清0。
11至5	未使用。这些位保留供未来其他功能使用，不必由用户代码监控。
4	电流通道ADC比较器阈值。仅在电流通道ADC比较器由ADCCFG MMR使能时有效。在I-ADC转换结果绝对值大于ADC0TH MMR写入值时由硬件置1。但如果使用ADC阈值计数器（ADC0TCL），仅在指定I-ADC转换次数等于ADC0THV MMR值时置1。 重新配置ADC或禁用比较器后硬件自动清0。
3	电流通道ADC超量程位。 当通过ADCCFG MMR使能超量程检测功能且I-ADC输入明显（约>30%）超量程时硬件置1。每隔125 μs更新一次。 仅当ADCCFG[2]清0以禁用此功能或通过ADC0CON MMR修改ADC增益时由软件清0。
2	温度转换结果就绪位。 当温度通道ADC使能后，一将有效温度转换结果写入温度数据寄存器（ADC2DAT MMR）便由硬件置1。每次校准结束后也会置1。读取ADC2DAT或ADC0DAT后即清0。
1	电压转换结果就绪位。 当电压通道ADC使能后，一将有效电压转换结果写入电压数据寄存器（ADC1DAT MMR）便由硬件置1。每次校准结束后也会置1。读取ADC1DAT或ADC0DAT后即清0。
0	电流转换结果就绪位。 当电流通道ADC使能后，一将有效电流转换结果写入电流数据寄存器（ADC0DAT MMR）便由硬件置1。每次校准结束后也会置1。读取ADC0DAT后即清0。

ADuC7036

ADC中断屏蔽寄存器

名称: ADCMSKI

地址: 0xFFFFF0504

默认值: 0x00

访问类型: 读写

功能: 该寄存器允许ADC中断源单独使能。此寄存器的位位置和ADCSTA MMR的低八位一样。若一位由用户代码置1, 相应中断使能。默认情况下, 所有位为0, 表示禁用所有ADC中断源。

ADC模式寄存器

名称: ADCMDE

地址: 0xFFFFF0508

默认值: 0x00

访问类型: 读写

功能: 该8位寄存器配置ADC子系统的工作模式。

表36. ADCMDE MMR位分配

位	描述
7	未使用。该位保留供未来使用, 应由用户代码写入0。
6	20 kΩ电阻选择。 置1选择20 kΩ电阻, 如图20所示。 置0选择直接接地路径, 如图20所示(默认)。
5	低功耗模式基准电压选择。 置1在低功耗模式或超低功耗模式下使能精密基准电压源, 导致功耗增加。 置0在低功耗模式或超低功耗模式下使能低功耗基准电压源(默认)。
4至3	ADC功耗模式配置。 00 = ADC正常模式。使能后ADC以正常功耗工作, 获得最佳电气性能。 01 = ADC低功耗模式。使能后I-ADC以低功耗工作。主要是通过将增益固定为128并利用片内低功耗(131 kHz)振荡器直接驱动ADC电路来实现(牺牲ADC噪声性能)来实现。 10 = ADC超低功耗模式。使能后ADC以低功耗工作。在此模式下增益固定为512且功耗约200 μA, 高于ADC低功耗模式。功耗更高同时确保了ADC噪声性能优于ADC低功耗模式。 11 = 未定义。
2至0	ADC工作模式配置。 000 = ADC断电模式。所有ADC电路(包括内部基准电压)断电。 001 = ADC连续转换模式。在此模式下任何使能后ADC连续转换。 010 = ADC单转换模式。在此模式下任何使能ADC执行一次转换。ADC在完成单次转换后进入空闲模式。取决于斩波模式, 单次转换耗费两到三个ADC时钟周期。 011 = ADC空闲模式。在此模式下ADC完全上电, 但保持复位状态。 100 = ADC自失调校准。在此模式下利用内部生成的0V对任何使能ADC进行失调校准。校准是在用户编程的ADC设置下进行的; 因此, 和正常单次ADC转换一样, 需要两到三个ADC转换周期, 校准结果才能完全建立。然后自动写入对应ADC的ADCxOF MMR。失调校准周期结束后, ADC返回空闲模式并设置校准和转换就绪状态位。 101 = ADC自增益校准。在此模式下, 参照内部基准电压校准所有使能后ADC的增益。增益校准分两阶段, 耗时是失调校准的两倍。校准结果自动写入对应ADC的ADCxGN MMR。增益校准周期结束后, ADC返回空闲模式并设置校准和转换就绪状态位。只有电流通道ADC才能进行自增益校准。电压温度测量应使用编程、工厂校准的系数(内部Flash/EE自动下载)。在使用外部NTC的情况下, 应对温度通道进行自校准。 110 = ADC系统零电平校准。在此模式下参照ADC输入引脚处驱动的外部零电平电压对使能ADC通道进行零电平校准。校准是在用户编程的ADC设置下进行的; 因此, 和正常单次ADC转换一样, 需要三个ADC转换周期, 校准结果才能完全建立。 111 = ADC系统满量程校准。在此模式下参照ADC输入引脚处驱动的外部满量程电压对使能ADC通道进行满量程校准。

电流通道ADC控制寄存器

名称: ADC0CON

地址: 0xFFFF050C

默认值: 0x0000

访问类型: 读写

功能: 该16位寄存器用于配置I-ADC。

注意, 在ADC0CON内重新配置电流ADC后, 电压和温度ADC也会复位。

表37. ADC0CON MMR位分配

位	描述
15	电流通道ADC使能。 由用户代码置1使能I-ADC。 清零0, I-ADC断电并将ADCSTA MMR中的对应ADC就绪位复位为0。
14, 13	IIN电流源使能。 00 = 电流源关闭。 01 = 在IIN+上使能50 μ A电流源。 10 = 在IIN-上使能50 μ A电流源。 11 = 同时在IIN-和IIN+上使能50 μ A电流源。
12至10	未使用。这些位保留供未来其他功能使用, 应写入0。
9	电流通道ADC输出编码。 用户代码置1将I-ADC输出编码配置为单极性。 用户代码清0将I-ADC输出编码配置为二进制补数。
8	未使用。这些位保留供未来其他功能使用, 应写入0。
7, 6	电流通道ADC输入选择。 00 = 选择IIN+、IIN-。 01 = 选择IIN-、IIN-。诊断性、内部短路配置。 10 = $V_{ref}/136$, 0 V, 诊断性、测试电压, 适用于增益设置 ≤ 128 。注意, 如果选择(REG_AVDD, AGND)二分频基准电压, 则在此模式下使用REG_AVDD作为VREF。从而导致将ADC0DAT比例缩放2。 11 = 未定义。
5, 4	电流通道ADC基准电压选择。 00 = 选择内部1.2 V精密基准电压。在ADC低功耗模式下, 基准电压选择由ADCMDE[5]控制。 01 = 选择外部基准电压输入 (VREF, GND_SW)。 10 = 选择外部基准电压输入二分频(VREF, GND_SW)/2, 从而允许高达REG_AVDD的外部基准电压。 11 = 选择(REG_AVDD, AGND)二分频。
3至0	电流通道ADC增益选择。标称I-ADC满量程输入电压 = (VREF/增益) 0000 = I-ADC增益1。 0001 = I-ADC增益2。 0010 = I-ADC增益4。 0011 = I-ADC增益8。 0100 = I-ADC增益16。 0101 = I-ADC增益32。 0110 = I-ADC增益64。 0111 = I-ADC增益128。 1000 = I-ADC增益256。 1001 = I-ADC增益512。 1xxx = I-ADC增益未定义。

ADuC7036

电压/温度通道ADC控制寄存器

名称: ADC1CON

地址: 0xFFFF0510

默认值: 0x0000

访问类型: 读写

功能: 该16位寄存器用来配置V-/T-ADC。

注意, 选择VBAT衰减器输入后, 电压衰减器或缓冲会自动使能。

表38. ADC1CON MMR位分配

位	描述
15	电压/温度通道ADC使能。 用户代码置1使能V-/T-ADC。 清0V-/T-ADC断电。
14, 13	VTEMP电流源使能。 00 = 电流源关闭。 01 = 在VTEMP上使能50 μ A电流源。 10 = 在GND_SW上使能50 μ A电流源。 11 = 在VTEMP和GND_SW上使能50 μ A电流源。
12至10	未使用。这些位保留供未来其他功能使用, 不能由用户代码修改。
9	电压/温度通道ADC输出编码。 用户代码置1将V-/T-ADC输出编码配置为单极性。 用户代码清0将V-/T-ADC输出编码配置为二进制补码。
8	未使用。该位保留供未来使用, 应由用户代码写入0。
7, 6	电压/温度通道ADC输入选择。 00 = VBAT/24, AGND。选择VBAT衰减器。该配置中高压缓冲自动使能。 01 = VTEMP, GND_SW。选择外部温度输入, 转换结果写入ADC2DAT。 10 = 内部传感器。选择内部温度传感器输入, 转换结果写入ADC2DAT。温度梯度为0.33 mV/°C; 仅适用于内部温度传感器。 11 = 内部短路。短路输入。
5, 4	电压/温度通道ADC基准电压选择。 00 = 选择内部1.2 V精密基准电压。 01 = 选择外部基准电压输入 (VREF, GND_SW)。 10 = 选择外部基准电压输入二频 (VREF, GND_SW) /2。允许高达REG_AVDD的外部基准电压。 11 = 选择 (REG_AVDD, AGND)/2用于电压通道。选择 (REG_AVDD, GND_SW)/2用于温度通道。
3至0	未使用。这些位保留供未来其他功能使用, 不能由用户代码写入0。

ADC滤波器寄存器

名称：ADCFLT

地址：0xFFFF0518

默认值：0x0007

访问类型：读写

功能：该16位寄存器控制片内ADC的速度和分辨率。

注意：若ADCFLT被修改，电流和电压/温度ADC会复位。

表39. ADCFLT MMR位分配

位	描述
15	<p>斩波使能。</p> <p>由用户置1使能所有激活的ADC的系统斩波。该位置1后，ADC失调误差和漂移很低，但在AF = 0时输出速率降为原来的三分之一（见此表中的Sinc3抽取系数位[6:0]）。AF > 0时，输出更新速率在斩波开启或关闭时均相同。斩波使能时，建立时间为两个输出周期。</p>
14	<p>移动平均。</p> <p>使能除二移动平均功能，以便降低ADC噪声。在斩波激活时，自动使能。斩波不激活时可选，且使能后（斩波不激活时）不会降低ADC输出速率，而将建立时间增加一个转换周期。清0禁用移动平均功能。</p>
13至8	<p>平均系数（AF）。这些位写入值用来实现可编程一阶Sinc3后滤波器。如此表中Sinc3抽取系数位[6:0]中所述，平均系数可以进一步降低ADC噪声，但会牺牲输出噪声。</p>
7	<p>Sinc3修正。</p> <p>修正标准Sinc3频率响应以将滤波器阻带抑制增加约5dB。通过插入第二个陷波(NOTCH2) $f_{NOTCH2} = 1.333 \times f_{NOTCH}$，即可实现，其中 f_{NOTCH} 是响应中第一个陷波所在位置。</p>
6至0	<p>Sinc3抽取系数（SF）。¹ 写入这些位的值（SF）控制Sinc3滤波器的过采样（抽取系数）。当斩波位（位15，斩波使能）= 0且平均系数（AF）= 0时，Sinc3滤波器的输出速率 $f_{ADC} = (512,000 / ([SF + 1] \times 64)) \text{ Hz}^2$，以上公式适用于所有SF值 ≤ 125。</p> <p>SF = 126时，强制 f_{ADC} 为60 Hz。</p> <p>SF = 127时，强制 f_{ADC} 为50 Hz。</p> <p>有关SF（不包括126和127）和AF值的 f_{ADC} 计算信息，见表40。</p>

¹ 由于数字滤波器内部数据路径有限，可以用来产生所需ADC输出速率的Sinc3抽取系数（SF）和平均系数（AF）的组合也有限。此约束在正常功耗模式下将最低ADC更新速率限制为4Hz或在更低功耗模式下限制为1Hz。

² 在低功耗模式和超低功耗模式下，ADC由低功耗振荡器（131kHz）直接驱动而非512kHz。计算 f_{ADC} 时都应（约）除以4。

ADuC7036

表40. ADC转换速率和建立时间

斩波使能	平均系数	移动平均	f_{ADC}	$t_{SETTLING}^1$
否	否	否	$\frac{512,000}{[SF + 1] \times 64}$	$\frac{3}{f_{ADC}}$
否	否	是	$\frac{512,000}{[SF + 1] \times 64}$	$\frac{4}{f_{ADC}}$
否	是	否	$\frac{512,000}{[SF + 1] \times 64 \times [3 + AF]}$	$\frac{1}{f_{ADC}}$
否	是	是	$\frac{512,000}{[SF + 1] \times 64 \times [3 + AF]}$	$\frac{2}{f_{ADC}}$
是	N/A	N/A	$\frac{512,000}{[SF + 1] \times 64 \times [3 + AF] + 3}$	$\frac{2}{f_{ADC}}$

¹ 在第一个ADC可以使用前每个ADC需要约60 μ s的额外时间。

表41. SF和AF的允许组合

SF	AF范围		
	0	1至7	8至63
0至31	是	是	是
32至63	是	是	否
64至127	是	否	否

ADC配置寄存器

名称：ADCCFG

地址：0xFFFF051C

默认值：0x00

访问类型：读写

功能：该8位ADC配置MMR控制片内ADC相关的扩展功能。

表42. ADCCFG MMR位分配

位	描述
7	<p>模拟接地开关使能。</p> <p>用户软件置1将外部GND_SW引脚（引脚15）接到内部模拟地基准点。在程序控制下可用于将外部电路和部件接地和断开，从而在将外部电路或元件不用时将直流功耗降到最低。该位与ADCMDE[6]配合可选择20kΩ电阻接地。</p> <p>用户代码清0断开外部GND_SW引脚。</p>
6, 5	<p>电流通道（32位）累加器使能。</p> <p>00 = 累加器禁用并复位为0。一次完整ADC转换（ADCSTA[0]设置两次）结束后必须禁用累加器，然后再使能以确保累加器复位。</p> <p>01 = 累加器激活。</p> <p>正电流值相加到累加器总数；如果允许转换次数超过65,535次，累加器可能溢出。</p> <p>负电流值从累加器总数中减去；累加器箝位到最小值0。</p> <p>10 = 累加器激活。</p> <p>正电流值相加到累加器总数；如果允许转换次数超过65,535次，累加器可能溢出。负电流的绝对值从累加器总数中减去；在此模式下累加器继续在0以下负累加。</p> <p>11 = 未定义。</p>
4, 3	<p>电流通道ADC比较器使能。</p> <p>00 = 比较器禁用。</p> <p>01 = 比较器激活，如果I-ADC转换结果绝对值$I \geq \text{ADC0TH}$，则中断断言。</p> <p>10 = 比较器计数模式激活，如果对于ADC0TCL转换次数，I-ADC转换结果绝对值$I \geq \text{ADC0TH}$，则中断断言。转换值$I < \text{ADC0TH}$，计数器阈值（ADC0THV）复位为0。</p> <p>11 = 比较器计数模式激活，如果对于ADC0TCL转换次数，I-ADC转换结果绝对值$I \geq \text{ADC0TH}$，则中断断言。转换值$I < \text{ADC0TH}$，向0递减计数器阈值（ADC0THV）。</p>
2	<p>电流通道ADC超量程使能。</p> <p>用户置1以在电流通道ADC上使能粗调比较器。如果电流读数对于有效增益设置明显（约>30%）超量程，则设置ADCSTA MMR的超量程位。</p> <p>电流必须超出此范围持续125 μs以上标志位才能置1。在ADC低功耗模式下不应使用此项功能。</p> <p>用户代码清0禁用超量程功能。</p>
1	未使用。该位保留供未来使用，应由用户代码写入0。
0	<p>电流通道ADC结果计数器使能。</p> <p>置1使能结果计数模式。在此模式下，仅当ADC0RCV = ADC0RCL时产生I-ADC中断。这样不仅允许I-ADC持续监控电流，而且在达到预定转换次数后中断MCU内核。使能后，电压/温度ADC也会继续转换，但同样，发生ADC计数器中断时只保存最后一次转换结果（不存储中间V-/T-ADC转换结果）。</p>

ADuC7036

电流通道ADC数据寄存器

名称: ADC0DAT

地址: 0xFFFF0520

默认值: 0x0000

访问类型: 只读

功能: 该ADC数据MMR保持I-ADC的16位转换结果。ADC0转换结果就绪位 (ADCSTA[0]) 置1后ADC不能更新此MMR。MCU读取该MMR后即清0所有断言就绪标志 (ADCSTA[2:0])。

电压通道ADC数据寄存器

名称: ADC0DAT

地址: 0xFFFF0524

默认值: 0x0000

访问类型: 只读

功能: 该ADC数据MMR保持V-/T-ADC的16位电压转换结果。电压转换结果就绪位 (ADCSTA[1]) 置1后ADC不能更新此MMR。如果I-ADC未激活, MCU读取此MMR即清0所有断言就绪标志 (ADCSTA[2:1])。

温度通道ADC数据寄存器

名称: ADC2DAT

地址: 0xFFFF0528

默认值: 0x0000

访问类型: 只读

功能: 该ADC数据MMR保持V-/T-ADC的16位温度转换结果。温度转换结果就绪位 (ADCSTA[2]) 置1后禁止ADC更新此MMR。如果I-ADC和V-ADC未激活, MCU读取此MMR即清0所有断言就绪标志 (ADCSTA[2])。此MMR读取后即清0ADCSTA[2]。

电流通道ADC失调校准寄存器

名称: ADC0OF

地址: 0xFFFF0530

默认值: 视器件而定, 工厂设置

访问类型: 读写

功能: 该ADC失调MMR保持I-ADC的16位失调校准系数。上电时该寄存器配置为出厂默认值。但是, 如果用户通过ADCMDE MMR位对I-ADC进行失调校准, 寄存器自动覆写。用户代码只有在ADC处于空闲模式时才能写入该校准寄存器。在写入任何失调或增益寄存器前ADC必须使能并且处于空闲模式。空闲模式必须持续至少23 μ s。

电压通道ADC失调校准寄存器

名称: ADC1OF

地址: 0xFFFF0534

默认值: 视器件而定, 工厂设置

访问类型: 读写

功能: 该ADC失调MMR保持电压通道的16位失调校准系数。上电时该寄存器配置为出厂默认值。但是, 如果用户通过ADCMDE MMR位对电压通道进行失调校准, 寄存器自动覆写。用户代码只有在ADC处于空闲模式时才能写入该校准寄存器。在写入任何失调或增益寄存器前ADC必须使能并且处于空闲模式。空闲模式必须持续至少23 μ s。

温度通道ADC失调校准寄存器

名称: ADC2OF

地址: 0xFFFF0538

默认值: 视器件而定, 工厂设置

访问类型: 读写

功能: 该ADC失调MMR保持温度通道的16位失调校准系数。上电时该寄存器配置为出厂默认值。但是, 如果用户通过ADCMDE MMR位对温度通道进行失调校准, 寄存器自动覆写。用户代码只有在ADC处于空闲模式时才能写入该校准寄存器。在写入任何失调或增益寄存器前ADC必须使能并且处于空闲模式。空闲模式必须持续至少23 μ s。

电流通道ADC增益校准寄存器

名称: ADC0GN

地址: 0xFFFF053C

默认值: 视器件而定, 工厂设置

访问类型: 读写

功能: 该增益MMR保持比例调整I-ADC转换结果的16位增益校准系数。上电时该寄存器配置为出厂默认值。但是, 如果用户通过ADCMDE MMR位对I-ADC进行增益校准, 寄存器自动覆写。用户代码只有在ADC处于空闲模式时才能写入该校准寄存器。在写入任何失调或增益寄存器前ADC必须使能并且处于空闲模式。空闲模式必须持续至少23 μ s。

电压通道ADC增益校准寄存器

名称: ADC1GN

地址: 0xFFFF0540

默认值: 视器件而定, 工厂设置

访问类型: 读写

功能: 该增益MMR可保持比例调整电压通道转换结果的16位增益校准系数。上电时该寄存器配置为出厂默认值。但是, 如果用户通过ADCMDE MMR位对电压通道进行增益校准, 寄存器自动覆盖。用户代码只有在ADC处于空闲模式时才能写入该校准寄存器。在写入任何失调或增益寄存器前ADC必须使能并且处于空闲模式。空闲模式必须持续至少23 μ s。

温度通道ADC增益校准寄存器

名称: ADC2GN

地址: 0xFFFF0544

默认值: 视器件而定, 工厂设置

访问类型: 读写

功能: 该增益MMR可保持比例调整温度通道转换结果的16位增益校准系数。上电时该寄存器配置为出厂默认值。但是, 如果用户通过ADCMDE MMR位对温度通道进行增益校准, 寄存器自动覆盖。用户代码只有在ADC处于空闲模式时才能写入该校准寄存器。在写入任何失调或增益寄存器前ADC必须使能并且处于空闲模式。空闲模式必须持续至少23 μ s。

电流通道ADC结果计数器限值寄存器

名称: ADC0RCL

地址: 0xFFFF0548

默认值: 0x0001

访问类型: 读写

功能: 该16位MMR可设置发生ADC中断所必需的转换次数。默认情况下, 置0x0001。必须先ADCCFG MMR设置ADC结果计数器使能位来使能ADC计数器功能。

电流通道ADC结果计数寄存器

名称: ADC0RCV

地址: 0xFFFF054C

默认值: 0x0000

访问类型: 只读

功能: 该16位只读MMR可保持I-ADC转换结果的当前数值。和ADC0RCL配合产生I-ADC中断屏蔽, 从而产生较低的中断速率。ADC0RCV = ADC0RCL时, ADC0RCV值复位至0并重新开始计数。还可配合累加器(ADC0ACC)一起计算电流均值。通过ADCCFG[0]使能。当重新配置I-ADC时, 即当写入ADC0CON或ADCMDE时, 复位至0。

电流通道ADC阈值寄存器

名称: ADC0TH

地址: 0xFFFF0550

默认值: 0x0000

访问类型: 读写

功能: 该16位MMR可设置阈值以和转换结果绝对值进行比较。在单极性模式下, 和ADC0TH[15:0]比较, 而在二进制补码模式下, 和ADC0TH[14:0]比较。

电流通道ADC阈值计数限值寄存器

名称: ADC0TCL

地址: 0xFFFF0554

默认值: 0x01

访问类型: 读写

功能: 该8位MMR可决定在设置ADCSTA MMR的I-ADC比较器阈值位, 产生中断前转换结果读数超过ADC0TH必须达到的累积次数(即低于递减阈值或将计数复位至0的值)。ADC0THV = ADC0TCL, I-ADC比较器阈值位即置位。

电流通道ADC阈值计数寄存器

名称: ADC0THV

地址: 0xFFFF0558

默认值: 0x00

访问类型: 只读

功能: 该8位MMR在每次转换结果绝对值 $|I| \geq \text{ADC0TH}$ 时递增。每次 $|I| < \text{ADC0TH}$ 时递减或复位至0。通过设置ADCCFG MMR的电流通道ADC比较器位进行配置。

ADuC7036

电流通道ADC累加器寄存器

名称: ADC0ACC

地址: 0xFFFF055C

默认值: 0x00000000

访问类型: 只读

功能: 该32位MMR可保持电流累加器值。应使用ADCSTA MMR的I-ADC就绪位来确保此MMR的安全读取。在ADCCFG MMR内禁用累加器或重新配置电流通道ADC时复位至0。

低功耗基准电压源比例因子寄存器

名称: ADCREF

地址: 0xFFFF057C

默认值: 视器件而定, 工厂设置

访问类型: 读写。严禁写入此寄存器。

功能: 该MMR允许用户校正低功耗模式(LPM)基准电压的初始误差。值0x8000, 表示和正常模式基准电压相比没有误差。通过将ADC结果大小乘以ADCREF值并除以0x8000, 补偿低功耗基准电压实际值。如果LPM基准电压源比1.2 V低1%, 则ADCREF值约为0x7EB9。如果比1.2 V高1%, 则值约为0x8147。

此寄存器校正25°C下的LPM基准电压有效值, 该温度是在ADI公司生产流程上测量基准电压时的温度。使用ADCREF MMR时LPM基准电压的温度系数不发生变化。

在低功耗模式下使用精密基准电压(即设置ADCMDE[5])时不应使用此寄存器。

ADC功耗模式

通过修正ADCMDE[4:3], 可配置各种低功耗或全功耗模式, ARM7 MCU也可配置成低功耗模式(POWCON[5:3])。内核功耗模式是独立控制的, 和下文所述ADC功耗模式无关。

ADC正常功耗模式

正常模式下, 电流和电压/温度通道全部使能。此时采用512 kHz调制器时钟, 使能ADC在4Hz和8 kHz之间速率提供正常转换结果(见ADC滤波器寄存器节)。两个通道都由MCU完全控制, 可随时重新配置。所有通道的默认ADC更新速率为1 kHz。

注意, I-ADC和V-/T-ADC通道可配置在返回全功耗模式前在正常功耗模式下高精度地启动单转换周期。此配置由MCU通过ADCMDE MMR完全控制, 更加灵活, 从而实现电池电流、电压和温度设置的持续周期性监控, 同时将平均直流功耗降到最低。

正常模式下, PLL不得关断。

ADC低功耗模式

此时I-ADC功耗和精度都会降低。调制器时钟直接采用片内131 kHz低功耗振荡器, 将ADC更新速率降低至1 Hz(ADCFLT)。增益固定为128。

ADC正常功耗模式节所述的所有ADC外设功能(结果计数器、数字比较器和累加器)均可在低功耗模式使能。

一般只有I-ADC配置低更新速率, 持续监控电池电流。MCU处于省电模式并在产生中断时唤醒。I-ADC检测到当前转换结果超过预设阈值、设定点或设定次数, 就会产生这类中断。

设置ADCMDE[5], 还可选择精密基准电压源或低功耗模式基准电压源。

ADC超低功耗模式

此时I-ADC通道工作模式几乎和低功耗模式相同(ADCMDE[4:3])。不过, 增益固定在512, 功耗提升(约)200 μ A, 噪声性能比低功耗模式设置更出色。

ADC正常功耗模式节所述的所有ADC外设功能(结果计数器、数字比较器和累积器)均可在超低功耗模式使能。

跟低功耗模式一样, I-ADC只配置为低更新速率, 持续监控电池电流。MCU处于省电模式并在产生中断时唤醒。I-ADC检测到电流转换结果超过预设阈值或设定点, 就会产生中断。

设置ADCMDE[5], 还可选择精密基准电压源或低功耗模式基准电压源。

ADC比较器和累加器

通过内置比较器逻辑，I-ADC可以在达到预定转换次数或超过预设阈值后产生中断。

每个转换结果都可以和ADCCFG[4:3]的预设阈值(ADC0TH)进行比较。如果结果绝对（符号无关）值大于预设比较器阈值，就会产生MCU中断。或者，作为比较器的扩展功能，用户代码可以配置一个阈值计数器（ADC0THV）监控转换结果高于或低于预设阈值的次数。阈值计数器达到ADC0TCL预设值，就会产生中断。

还可通过集成32位累加器（ADC0ACC）功能并设置ADCCFG[6:5]，相加或减去多个采样结果。用户代码可直接读取累加值（ADC0ACC），不需要任何进一步软件处理。

ADC SINC3数字滤波器响应

所有ADuC7036 ADC的总频率响应主要取决于片内Sinc3数字滤波器的低通滤波器响应。Sinc3滤波器的作用是抽取ADC Σ - Δ 调制器输出数据位流以产生有效的16位数据结果。所有ADC的数字滤波器响应都完全相同并通过16位ADC滤波器寄存器(ADCFLT)配置。这个寄存器决定ADC的总吞吐速率。这个预设吞吐速率决定ADC噪声分辨率。对于电流通道ADC，吞吐速率和所选增益决定噪声分辨率。

总频率响应和吞吐速率主要取决于Sinc3滤波器抽取系数(SF)位(ADCFLT[6:0])和平均系数(AF)位(ADCFLT[13:8])的配置。由于数字滤波器内部数据路径有限，可以用来产生所需输出速率的SF和AF组合也有限，从而在正常功耗模式下将最低更新速率限制为4Hz或在低功耗模式下限制为1Hz。吞吐速率的计算方法详见ADCFLT位分配表（见表39），而AF和SF值的允许组合限制见表41。

默认情况下ADCFLT = 0x0007，将吞吐速率配置为1 kHz，同时禁用所有其他滤波器选项（斩波、移动平均、平均系数及Sinc3修正）。此时典型滤波器响应如图21所示。

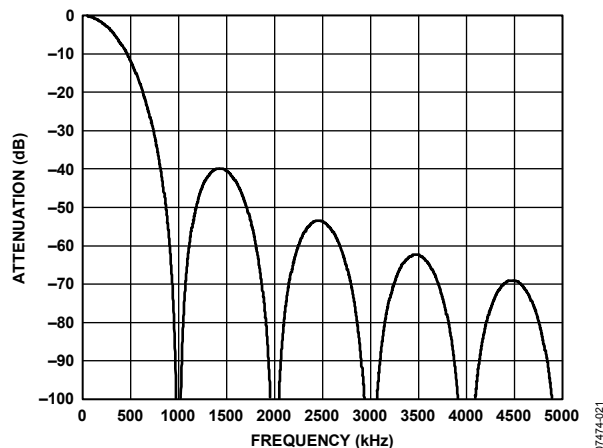


图21. $f_{ADC} = 1 \text{ kHz}$ ($ADCFLT = 0x0007$) 时的典型数字滤波器响应

此外，ADCFLT寄存器还提供了一个Sinc3修正位(ADCFLT[7])。用户置1可以修正标准Sinc3频率响应，将滤波器阻带抑制提升约5 dB。在以下等式决定的位置处插入第二个陷波即可

$$f_{NOTCH2} = 1.333 \times f_{NOTCH}$$

其中 f_{NOTCH} 是响应中第一个陷波所在位置。

此位有效，ADC噪声会略微增加。图22显示Sinc3修正位有效时修正过的1 kHz滤波器响应。和标准1 kHz响应相比，和阻带抑制提升一样，新陷波清楚地出现在1.33 kHz处。

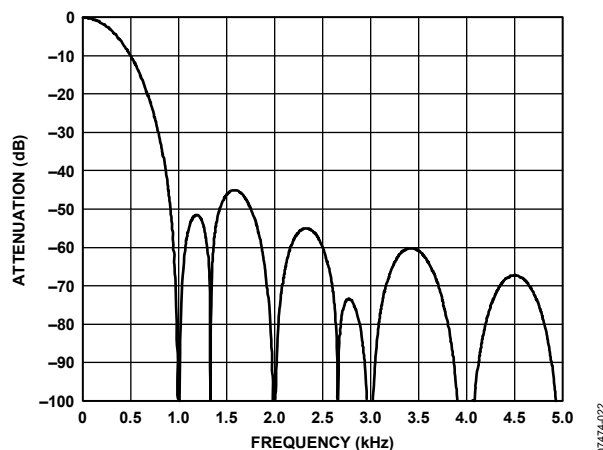


图22. $f_{ADC} = 1 \text{ kHz}$ ($ADCFLT = 0x0087$) 时的修正后Sinc3数字滤波器响应

ADuC7036

在正常功耗模式下，最大吞吐速率为8 kHz。此配置需将ADCFLT MMR内的SF和AF位置零，所有其他滤波选项禁用。因此ADCFLT写入0x0000。图23显示此时的典型8 kHz滤波器响应。

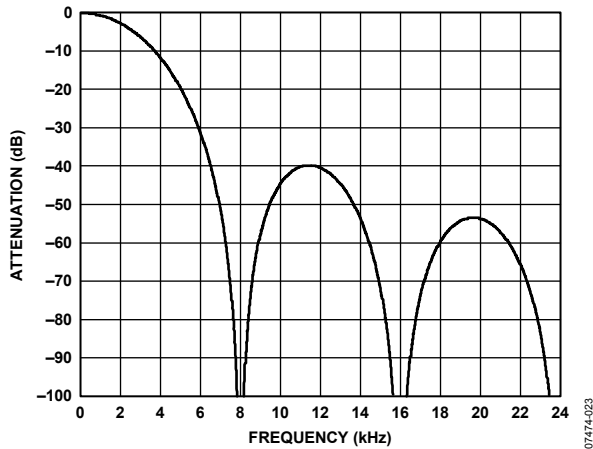


图23. $f_{ADC} = 8 \text{ kHz}$ ($ADCFLT = 0x0000$) 时的典型数字滤波器响应

8 kHz滤波器响应的改进形式可通过设置移动平均值(ADCFLT[14])加以配置。因此，在所有ADC输出采样点上引入了另外一个二分之一移动平均值滤波器，从而进一步降低了ADC输出噪声。此外，通过维持8 kHz吞吐速率，将建立时间增加一个完整转换周期。修正后的模拟频率响应如图24所示。

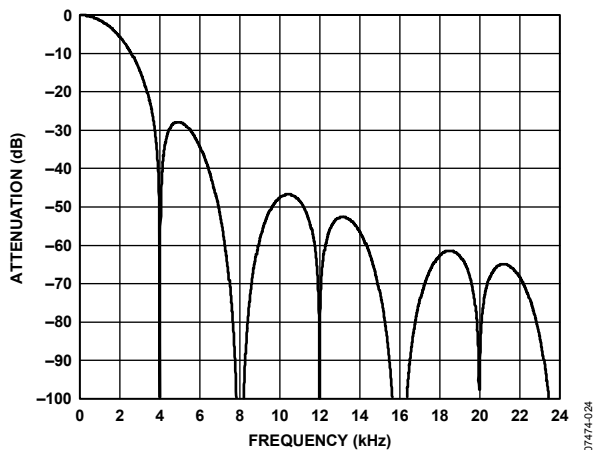


图24. $f_{ADC} = 8 \text{ kHz}$ ($ADCFLT = 0x4000$) 时的典型数字滤波器响应

吞吐速率极低时，可激活ADCFLT寄存器内的斩波使能位，最大程度降低失调误差，更重要的是最大程度降低了失调误差的温度漂移。斩波使能时，用户可使用两个主变量(Sinc3抽取系数和平均系数)来选择最佳滤波器响应，但需要权衡滤波器带宽和ADC噪声。

例如，斩波使能位(ADCFLT[15])置1，SF值(ADCFLT[6:0])递增至0x1F(十进制31)并选择AF值(ADCFLT[13:8])0x16(十进制22)，从而产生10 Hz的ADC吞吐速率。频率响应曲线如图25所示。

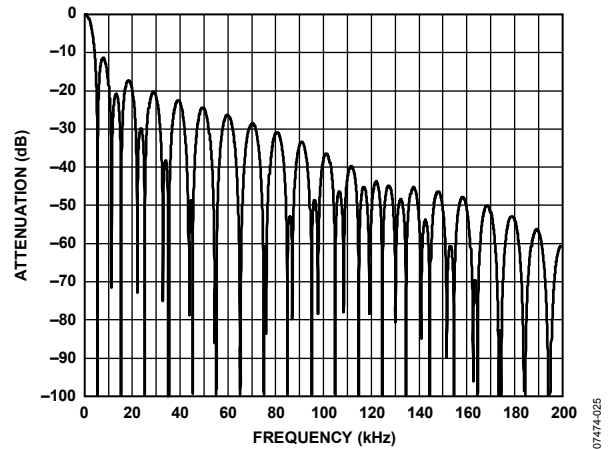


图25. $f_{ADC} = 10 \text{ kHz}$ ($ADCFLT = 0x961F$) 时的典型数字滤波器响应

SF变为0x1D且AF置0x3F，同时斩波使能位激活，可在正常模式下将最小吞吐速率配置为4 Hz。数字滤波器频率响应如图26所示。

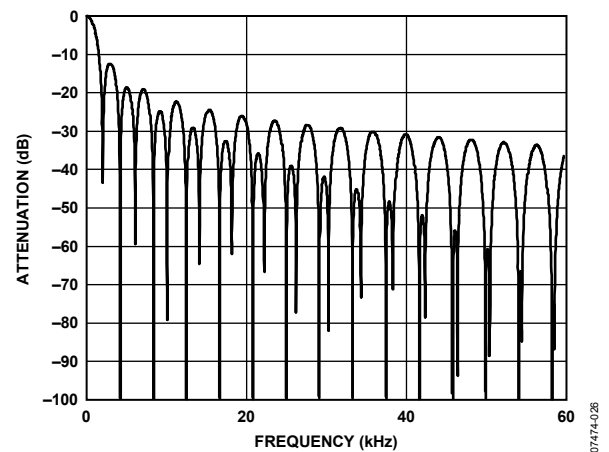


图26. $f_{ADC} = 4 \text{ kHz}$ ($ADCFLT = 0xBF1D$) 时的典型数字滤波器响应

在低功耗模式下， Σ - Δ 调制器时钟不再由512 kHz驱动而直接采用片内低功耗131 kHz振荡器。随后如果在正常模式下采用相同的ADCFLT配置，所有滤波器值将比例缩小约4倍。因此，可在低功耗模式下将吞吐速率设为1 Hz。滤波器频率响应如图27所示。

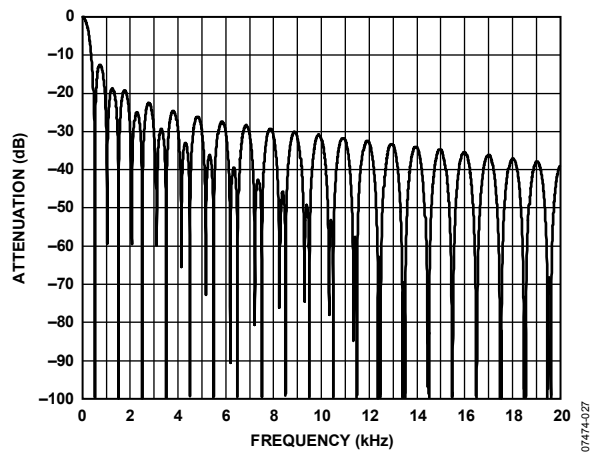


图27. $f_{\text{ADC}} = 1 \text{ kHz}$ ($\text{ADCFLT} = 0\text{xBD1F}$) 时的典型数字滤波器响应

一般情况下，相同的ADC更新速率可通过在ADCFLT寄存器内设置不同的SF和AF值来获得。但实际上，不管ADCFLT值如何，用户都要权衡频率响应和ADC噪声。要利用SF和AF组合获得最佳滤波器响应和ADC噪声，经验表明SF应选在16到40（十进制）或0x10到0x28之间，然后递增AF值直到获得所需ADC吞吐速率。表43列举了一些常见的ADCFLT配置。

表43. 常见ADCFLT配置

ADC模式	SF	AF	其他配置	ADCFLT	f_{ADC}	t_{SETTLE}
正常	0x1D	0x3F	斩波开启	0xBF1D	4 Hz	0.5 s
正常	0x1F	0x16	斩波开启	0x961F	10 Hz	0.2 s
正常	0x07	0x00	无	0x0007	1 kHz	3 ms
正常	0x07	0x00	Sinc3修正	0x0087	1 kHz	3 ms
正常	0x03	0x00	移动平均	0x4003	2 kHz	2 ms
正常	0x00	0x00	移动平均	0x4000	8 kHz	0.5 ms
低功耗	0x10	0x03	斩波开启	0x8310	20 Hz	100 ms
低功耗	0x10	0x09	斩波开启	0x8910	10 Hz	200 ms
低功耗	0x1F	0x3D	斩波开启	0xBD1F	1 Hz	2 s

ADC校准

如顶层框图（图17及18）详细所示，通过所有ADC通道的信号流如下：

1. 输入电压通过输入缓冲（在I-ADC时还通过PGA）施加到 Σ - Δ 调制器。
2. 调制器输出施加到可编程数字抽取滤波器。
3. 然后在使用斩波时对滤波器输出结果进行平均。
4. 将偏移值（ADCxOF）从上述结果中减去。
5. 根据增益值（ADCxGN）比例调节结果。
6. 然后转换成二进制补码/偏移二进制形式并舍入到16位或箝位到 \pm 满量程。

每个ADC都具有一个特定的失调和增益校正或校准系数与之相对应，它存储在MMR的失调和增益寄存器（ADCxOF和ADCxGN）内。这些寄存器的工作是消除器件内部失调和增益误差，以及器件外部系统级失调和增益误差。

上电时加载工厂预设校准值。这些工厂预设校准值因器件而异，反映了内部电路的工艺差异。但是，它们可以在空闲模式时被用户代码覆盖，也可以在用户利用ADCMDE[2:0] MMR的工作模式配置位激活失调或增益周期时自动覆盖。有两种自动校准形式可供用户选择：自校准和系统校准。

自校准

失调误差自校准时，ADC基于内部生成的0V来产生校准系数，而在增益误差自校准时，则基于满量程电压。虽然自校准可以校准内部失调和增益误差，但无法补偿外部误差，如分流电阻公差/漂移和外部失调电压。

注意，在自校准模式下，ADC0GN必须包含对应PGA = 1的值，才能开始校准配置。

系统校准

在失调误差系统校准过程中，ADC基于外部生成的零电平来产生校准系数，而在增益误差系统校准过程中，则基于满量程电压。在整个校准周期，校准系数始终施加到ADC外部输入端。

失调校准持续一个转换周期（ $3/f_{ADC}$ 斩波关闭， $2/f_{ADC}$ 斩波开启），然后ADC返回空闲模式。增益校准分两个过程，

因此是失调校准周期的两倍长。校准周期启动后，任何进行中的转换都会立即停止，校准自动以ADCFLT设置的更新速率进行，结束后ADC返回空闲模式。强烈建议，以尽可能低的更新速率来启动校准（因此要求ADCFLT中的SF值较高），这样在校准期间可以尽可能降低噪声的影响。

利用失调和增益校准

在斩波使能位ADCFLT[15]使能后，内部失调误差已降到了最低，可能不再要求任何失调校准。但如果斩波禁用，就需要进行初始失调校准，甚至可能反复校准，尤其是在发生较大温度变化之后。

视系统精度要求，可能需要在所有相关系统增益范围内进行增益校准，尤其是在I-ADC（带内部PGA）环境中。如果无法在所有增益范围内施加外部满量程电流，就施加一个较低电流，然后比例调节校准结果。例如，施加一个50%电流，然后将所产生的ADC0GN值除以2并写回到ADC0GN内。注意，由于ADC0GN只有16位，系统校准期间可以施加的输入信号是有下限的。输入范围（即系统零电平值和满量程值之间的差值）应大于额定满量程输入范围的40%（即 $>40\%$ 的VREF/增益）。

片内Flash/EE存储器可以存储多个校准系数。这些校准系数可根据系统配置由用户代码直接复制到相关校准寄存器内。一般情况下，使用校准寄存器的最简单方法是作为自动校准模式的一部分，让ADC自己计算所需值。

工厂设置或下线校准分两步：

1. 施加0 A电流。在所需PGA设置下配置ADC，并写入ADCMDE[2:0]以进行系统零电平校准，从而将新失调校准值写入ADC0OF内。
2. 针对所需PGA设置施加满量程电流。写入ADCMDE，进行系统满量程校准，从而将新失调校准值写入ADC0GN内。

了解失调和增益校准寄存器

ADC信号流（见ADC Sinc3数字滤波器响应节到利用失调和增益校准节）中的典型模块输出可视为 \pm 满量程输入范围约 ± 0.75 的小数部分。范围小于 ± 1 ，是因为调制器有衰减，使输入信号多少具有超量程能力。由于工艺容差，确切的衰减值因器件而略显不同。

失调系数读取自ADC00F校准寄存器，为16位二进制补码。根据信号链，有效范围为±1。因此，ADC00F寄存器的1 LSB和ADC0DAT寄存器的1 LSB不一样。

ADC00F正值表示从滤波器输出中减去失调，负值则相加。寄存器额定值为0x0000，表明要消除零失调。实际失调可能因器件、PGA增益不同而异。斩波模式使能（即ADCFLT[15]=1）可以将ADC内部失调降到最低。

增益系数读取自ADC0GN寄存器，为无单位的比例因子。16,384除以这个寄存器的16位值后再乘以失调校正值。寄存器标称值为0x5555，对应于倍乘因数1.3333，并比例调节标称±0.75信号产生±1满量程输出信号。在进行输出信号上溢/下溢检查后转换成二进制补码或单极性模式，然后输出到数据寄存器。

实际增益和零增益误差的所需比例因子在正常和低功耗模式下因器件、PGA设置而略有差异。上电复位时下载到ADC0GN的值对应PGA增益为1的比例因子。但如果使用不同的PGA设置，就可能会出现某种增益误差。通过用户代码或进行ADC校准覆盖校准系数即可校正这个误差。

简化后的ADC传递函数可描述为

$$ADC_{OUT} = \left[\frac{V_{IN} \times PGA}{V_{REF}} - ADC_{xOF} \right] \times \frac{ADC_{xGN}}{ADC_{xGN_{NOM}}}$$

其中等式对于电压/温度通道ADC有效。

对于电流通道ADC，

$$ADC_{OUT} = \left[\frac{V_{IN} \times PGA}{V_{REF}} - K \times ADC_{xOF} \right] \times \frac{ADC_{xGN}}{ADC_{xGN_{NOM}}}$$

其中K取决于PGA增益设置和ADC模式。

正常模式

在正常模式下，对于PGA增益1、4、8、16、32及64，K = 1；对于PGA增益2和128，K = 2；对于PGA增益256，K = 4；而对于PGA增益512，K = 8。

低功耗模式

在低功耗模式下，对于PGA增益128，K = 32。此外，如果使用REG_AVDD/2基准电压，则K系数翻倍。

超低功耗模式

在超低功耗模式下，对于PGA增益512，K = 8。此外，如果使用REG_AVDD/2基准电压，则K系数翻倍。

ADC诊断

ADuC7036在两个ADC上都具有一个诊断和开路检测功能。

电流ADC诊断

ADuC7036具有检测电流通道输入端开路状态的功能。通过ADC0CON[14:13]控制IIN+和IIN-上的两个电流源可以实现这个功能。

注意，IIN+和IIN-电流源公差为±30%。因此，电流源使能时PGA增益必须≥2（ADC0CON[3:0]≥0001）。

温度ADC诊断

ADuC7036具有检测温度通道输入端开路状态的功能。通过ADC1CON[14:13]控制VTEMP和GND_SW上的两个电流源可以实现这个功能。

电压ADC诊断

ADuC7036具有检测电压通道输入端开路状态的功能。通过高压寄存器HVCFG1[7]控制电压衰减器上的电流源可实现这项功能。

电源支持电路

ADuC7036具有两个片内低压差（LDO）稳压器，它们通过电池电压直接驱动来产生一个2.6 V内部电源，该2.6 V电源然后用作ARM7 MCU和外设的电源，包括那些片内精密模拟电路。

数字LDO在REG_DVDD上并接两个输出电容（2.2 μF 和0.1 μF ）而模拟LDO则在REG_AVDD上并接一个输出电容（0.47 μF ）。

输出电容的ESR会影响LDO控制回路的稳定性。建议在频率大于32 kHz时使用5 Ω 或以下的ESR以确保调节器的稳定性。

此外，还集成了上电复位（POR）、电压监控器（PSM）及低压标志（LVF）功能以确保MCU安全运行以及连续监控电池电源。

POR电路的设计VDD（0 V至12 V）上电时间大于100 μs 。因此，建议选择外部电源去耦元件时务必小心以确保不论VBAT上电状态如何，VDD电源上电时间始终大于100 μs 。VDD上的串联电阻和去耦电容组合要求RC时间常数至少为100 μs （例如10 Ω 和10 μF ，如图57所示）。

如图28所示，当VDD上的电源电压达到典型工作电压3 V时，POR信号将ARM内核保持在复位状态下持续20 ms。这样可以确保施加到ARM内核及相关外设上的调节电源电压（REG_DVDD）大于最小工作电压，由此保证功能完整。在RSTSTA MMR内设置POR标志，指示POR事件已发生。

ADuC7036还具有一个PSM功能。通过HVCFG0[3]使能PSM持续监控VDD引脚上的电压。如果电压低于6 V典型值，则PSM标志自动置1并在IRQEN[16]或FIQEN[16]内使能高压IRQ时产生系统中断。如图28所示实例。

电压低于POR电平时，可启用另一个低压标志（HVCFG0[2]）。这个标志可用来指示SRAM内容在复位事件后仍然有效。低压标志的实例如图28所示。HVCFG0[2]使能后，可通过HVMON[3]监控它的状态。HVCFG0[2]置1后，SRAM内容保持有效。若改位清0，可能会破坏SRAM内容。

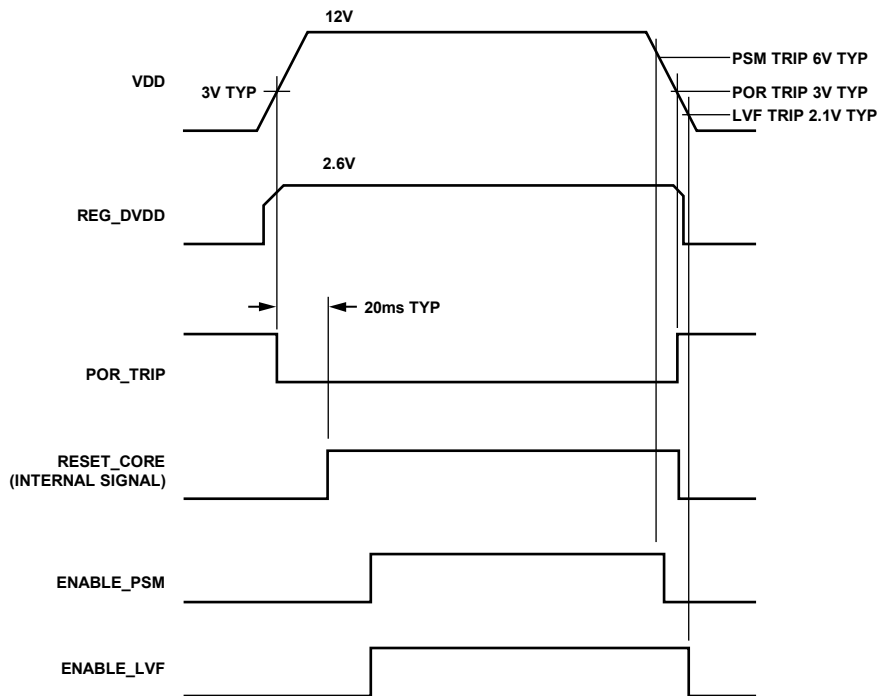


图28. 典型上电周期

07.474-028

系统时钟

ADuC7036集成一个非常灵活的时钟系统,可从下面三个时钟发生源之一产生:集成片内精密振荡器、集成片内低功耗振荡器和外部时钟晶体。三种方案如图29所示。

每个内部振荡器除以4即可产生32.768 kHz的时钟频率。内部振荡器或外部晶体提供32.768 kHz,然后PLL锁定在它的倍数(625)上,为系统提供稳定的20.48 MHz时钟。内核可以工作在该频率下或其二的倍数分频上,从而在不需峰值性能时节省功耗。

默认情况下,低功耗振荡器产生20.48 MHz时钟源,然后驱动PLL。时钟分频器(由POWCON寄存器的CD位设置)进

而驱动ARM7TDMI内核。默认情况下,CD位将PLL输出分半,产生10.24 MHz的内核时钟。用户可以设置这个除数因子,在1到128范围动态产生二进制加权除法系数。

ADC由PLL输出驱动,后者经分频后提供512 kHz的时钟源。低功耗模式时,ADC时钟源会从标准512 kHz切换到低功耗131 kHz振荡器。

注意,低功耗振荡器通过一个4分频电路来同时驱动看门狗和内核唤醒定时器。ADuC7036时钟系统的详细框图如图29所示。

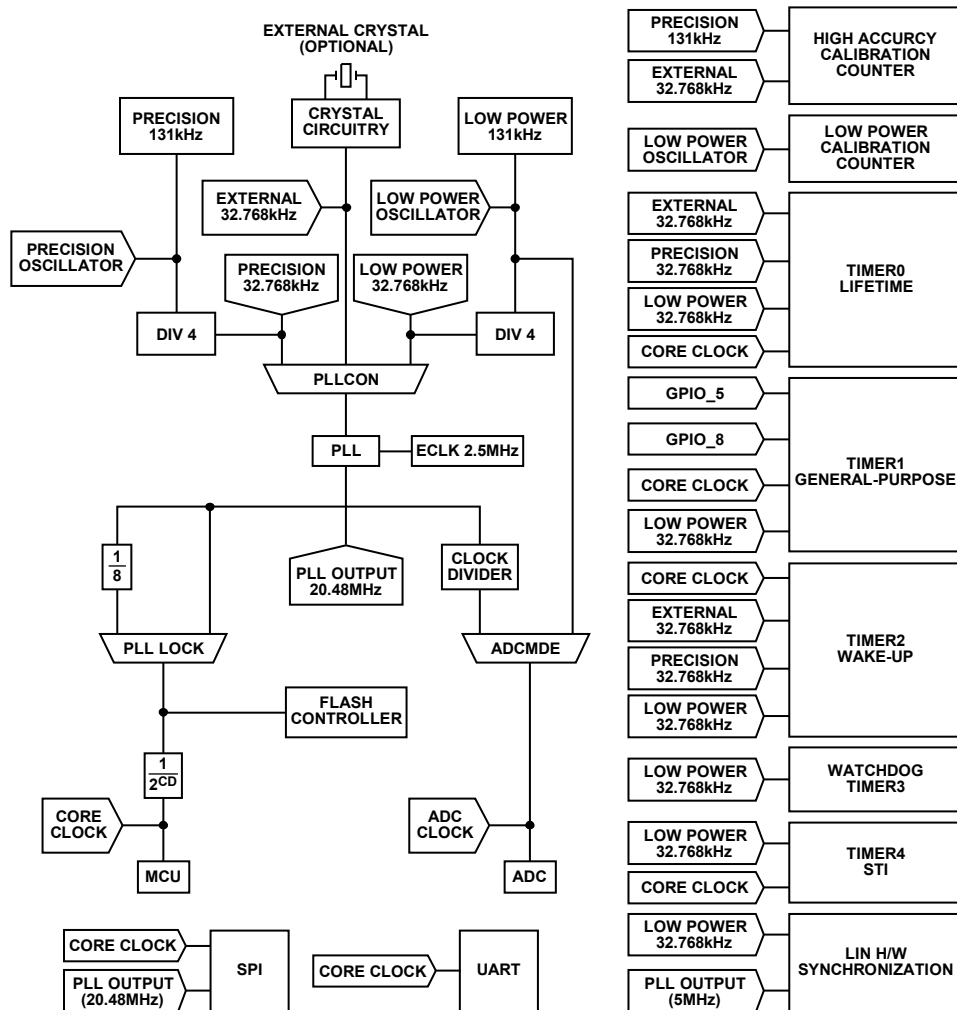


图 29. 系统时钟发生

ADuC7036

工作模式、时钟模式和可编程时钟分频器都由两个MMR PLLCON和POWCON控制，并且PLL状态由PLLSTA指示。PLLCON控制时钟系统的工作模式，而POWCON同时控制内核时钟频率和省电模式。PLLSTA指示XTAL1引脚上是否存在振荡器并提供PLL锁定状态和PLL中断的相关信息。

在ADuC7036进入省电模式前，建议将PLL时钟源切换到低功耗131 kHz振荡器以缩短唤醒时间。低功耗振荡器始终有效。

当ADuC7036从省电模式中唤醒时，PLL一开始振荡，MCU内核当即执行代码。代码执行在PLL锁定到20.48 MHz频率前进行。当PLL正在锁定时，为了确保Flash/EE存储器控制器获得有效的时钟，它通过PLL输出8分频时钟源来驱动。锁定后，PLL输出才从这个8分频切换到锁定的输出。

如果用户代码需要精确的PLL输出，用户代码必须在唤醒后恢复正常代码执行前轮询PLL锁定状态位(PLLSTA[1])。

唤醒后，如果PLL从一个有效时钟源（如低功耗131 kHz振荡器）获得时钟信号，锁定时间在2 ms内。

PLLCON为采用两个32位密钥保护的MMR。PLLKEY0（预写密钥）和PLLKEY1（写后密钥）。密钥值如下：

PLLKEY0 = 0x000000AA

PLLKEY1 = 0x00000055

PLLCON为采用两个32位密钥保护的MMR：POWKEY0（预写密钥）和POWKEY1（写后密钥）。

POWKEY0 = 0x00000001

POWKEY1 = 0x000000F4

两个MMR写入范例如下：

```
POWKEY0 = 0x01 //POWCON key
POWCON = 0x00 //Full power-down
POWKEY1 = 0xF4 //POWCON KEY
iA1*iA2 //Dummy cycle to clear the
pipeline, where iA1 and iA2 are defined as longs and are not 0
PLLKEY0 = 0xAA //PLLCON key
PLLCON = 0x0 //Switch to Low Power Osc.
PLLKEY1 = 0x55 //PLLCON key
iA1*iA2 //Dummy cycle to prevent
Flash/EE access during clock change
```

系统时钟寄存器

PLLSTA 寄存器

名称：PLLSTA

地址：0xFFFF0400

默认值：N/A

访问类型：只读

功能：该8位寄存器允许用户代码监控PLL锁定状态和外部晶体状态。

表44. PLLSTA MMR位分配

位	描述
7至3	保留。
2	XTAL时钟。只读，实时指示XTAL1上的当前逻辑电平。在32.768 kHz时钟频率下通过高低交替来指示是否存在外部时钟源。
1	PLL锁定状态位。只读。 PLL锁定并输出20.48 MHz时置1。 PLL不锁定并输出 f_{CORE} 8分频时钟源时清0。
0	PLL中断。 PLL锁定状态位信号为低电平时置1。 该位写入1时清0。

PLLCON 预写密钥

名称: PLLKEY0

地址: 0xFFFF0410

访问类型: 只写

密钥: 0x000000AA

功能: 该加密寄存器在写入PLLCON前后要求32位密钥值。
PLLKEY0为预写密钥。

PLLCON 写后密钥

名称: PLLKEY1

地址: 0xFFFF0418

访问类型: 只写

密钥: 0x00000055

功能: 该加密寄存器在写入PLLCON前后要求32位密钥值。
PLLKEY1为写后密钥。

PLLCON 寄存器

名称: PLLCON

地址: 0xFFFF0414

默认值: 0x00

访问类型: 读写

功能: 该8位寄存器允许用户代码从三种不同振荡器源中动态选择PLL时钟源。

表45.PLLCON MMR位分配

位	描述
7至2	保留。应由用户代码置0。
1至0	PLL时钟源。 ¹ 00 = 更低功耗131 kHz振荡器。 01 = 精密131 kHz振荡器。 10 = 外部32.768 kHz晶体。 11 = 保留。

¹ 如果用户代码切换MCU时钟源, 在将时钟切换写入PLLCON后应插入一个MCU空周期。

POWCON 预写密钥

名称: POWKEY0

地址: 0xFFFF0404

访问类型: 只写

密钥: 0x00000001

功能: 该加密寄存器在写入POWCON前后要求32位密钥值。
POWKEY0为预写密钥。

POWCON 写后密钥

名称: POWKEY1

地址: 0xFFFF040C

访问类型: 只写

密钥: 0x000000F4

功能: 该加密寄存器在写入POWCON前后要求32位密钥值。
POWKEY1为写后密钥。

ADuC7036

POWCON寄存器

名称: POWCON

地址: 0xFFFF0408

默认值: 0x79

访问类型: 读写

功能: 该8位寄存器允许用户代码动态进入不同低功耗模式并修改CD分频器以控制ARM7TDMI内核速度。

表46. POWCON MMR位分配

位	描述
7	精密131 kHz输入使能。 用户置1使能精密131 kHz输入。精密131 kHz振荡器还必须通过HVCFG0[6]使能。置1后功耗增加约50 μ A。不用时禁用。 用户清0关断精密131 kHz输入使能。
6	XTAL关断。 用户置1使能外部晶体电路。 用户清0关断外部晶体电路。
5	PLL关断。如果定时器外设由PLL输出时钟驱动, 则关断。如果由有效时钟源驱动, 则保持正常功耗模式。 默认置1并在唤醒事件时由硬件置1。 清0关断PLL。内核或外设使能时无法PLL关断。位3、4及5必须同时清0。
4	外设关断。可以关断的外设有: SRAM、Flash/EE存储器及GPIO接口以及SPI和UART串行端口。 默认置1和/或唤醒事件时由硬件置1。即使置1, 由低功耗振荡器驱动, 唤醒定时器(定时器2)仍可继续工作。 清0关断外设。内核使能时, 外设无法关断。位3、4必须同时清0。即使清0, LIN仍能对唤醒事件作出响应。
3	内核关断。用户代码关断MCU时, 在将关断命令写入POWCON后插入一个MCU空周期。 默认置1并在唤醒事件时由硬件置1。清0关断ARM内核。
2至0	CD内核时钟分频器位。 000 = 20.48 MHz, 48.83 ns。 001 = 10.24 MHz, 97.66 ns (上电默认设置)。 010 = 5.12 MHz, 195.31 ns。 011 = 2.56 MHz, 390.63 ns。 100 = 1.28 MHz, 781.25 ns。 101 = 640 kHz, 1.56 μ s。 110 = 320 kHz, 3.125 μ s。 111 = 160 kHz, 6.25 μ s。

低功耗时钟校准

低功耗131 kHz振荡器既可使用精密131 kHz振荡器校准，也可使用外部32.768 kHz时钟晶体校准。校准需要用到两个专用校准计数器和一个振荡器调节寄存器。

第一个计数器（计数器0）为9位，通过精确时钟振荡器（精密振荡器或外部时钟晶体）驱动。第二个计数器（计数器1）为10位，通过低功耗振荡器（直接131 kHz或4分频模块产生的32.768 kHz）驱动。每个校准计数器的时钟源频率都应相同。调节寄存器（OSC0TRM）为8位，低四位是用户可访问的调节位。OSC0TRM内的值越高，低功耗振荡器的频率越低。反之，OSC0TRM内的值越低，频率越高。根据标称频率131 kHz，典型调节范围在127 kHz和135 kHz之间。

时钟校准模式由下列MMR控制：

- OSC0CON：校准控制位。
- OSC0STA：校准状态寄存器。
- OSC0VAL0：9位计数器0。
- OSC0VAL1：10位计数器1。
- OSC0TRM：振荡器调节寄存器。

校准程序流程如图30所示。用户代码利用OSC0CON配置并使能校准序列。OSC0VAL0精密功耗振荡器校准计数器达到0x1FF后两个计数器禁用。然后用户代码回读低功耗振荡器校准计数器值。有三种可能情形：

- $OSC0VAL0 = OSC0VAL1$ 。无需进一步操作。
- $OSC0VAL0 > OSC0VAL1$ 。低功耗振荡器缓慢运行中。必须减少OSC0TRM值。
- $OSC0VAL0 < OSC0VAL1$ 。低功耗振荡器快速运行中。必须增加OSC0TRM值。

由于OSC0TRM值已发生了变化，应再次执行校准程序，同时检查新频率。

利用内部精密131 kHz振荡器执行校准程序耗时约4 ms。使用外部32.768 kHz晶体的话，增至16 ms。

开始时钟校准程序前，用户必须将PLL时钟源切换到精密131 kHz振荡器或外部32.768 kHz时钟晶体。否则，每次修改OSC0TRM，PLL可能会无法锁定，从而延长了低功耗振荡器校准时间。

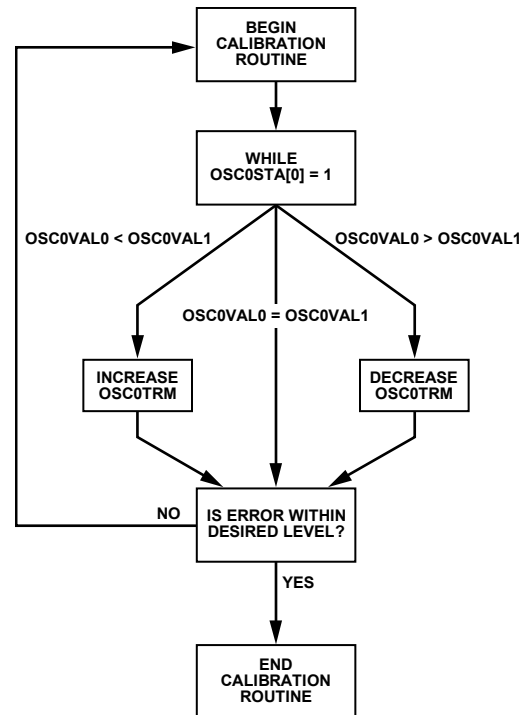


图30. OSC0TRM 校准程序

07474-030

ADuC7036

OSC0TRM 寄存器

名称: OSC0TRM

地址: 0xFFFF042C

默认值: 0xX8

访问类型: 读写

功能: 该8位寄存器控制低功耗振荡器调节。

表47. OSC0TRM MMR位分配

位	描述
7至4	保留。应写入0。
3至0	用户调节位。

OSC0CON 寄存器

名称: OSC0CON

地址: 0xFFFF0440

默认值: 0x00

访问类型: 读写

功能: 该8位寄存器控制低功耗振荡器校准程序。

表48. OSC0CON MMR位分配

位	描述
7至5	保留。应写入0。
4	校准源。 置1选择外部32.768 kHz晶体。 清0选择内部精密131 kHz振荡器。
3	校准复位。置1复位校准计数器并禁用校准逻辑。
2	置1清除OSC0VAL1。
1	置1清除OSC0VAL0。
0	校准使能。 置1开始校准。 清0中止校准。

OSC0STA 寄存器

名称: OSC0STA

地址: 0xFFFF0444

默认值: 0x00

访问类型: 只读

功能: 该8位寄存器指示低功耗振荡器校准程序状态。

表49. OSC0STA MMR位分配

位	描述
7至2	保留。
1	校准结束。 校准周期完全结束后硬件置1。 读取OSC0VAL1后清0。
0	校准中置1。校准结束清0。

OSC0VAL0 寄存器

名称: OSC0VAL0

地址: 0xFFFF0448

默认值: 0x0000

访问类型: 只读

功能: 该9位计数器可从131 kHz精密振荡器或32.768 kHz外部晶体接收时钟信号。

OSC0VAL1 寄存器

名称: OSC0VAL1

地址: 0xFFFF044C

默认值: 0x0000

访问类型: 只读

功能: 该10位计数器可从低功耗131 kHz振荡器接收时钟信号。

处理器相关外设

中断系统

ADuC7036拥有由中断控制器控制的16个中断源。大多数中断都是由片内外设产生的，例如ADC和UART。ARM7TDMI CPU内核只能识别两类中断：正常中断请求（IRQ）和快速中断请求（FIQ）。所有中断都可以被单独屏蔽。

9个与中断有关的寄存器专门管理中断系统的控制和配置，其中4个用于控制IRQ，4个用于控制FIQ。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寄存器中的控制位都代表相同的中断源，如表50所示。

进入中断服务程序（ISR）后应立即保存IRQSTA/FIQSTA以确保能够响应所有有效中断源。

ARM7TDMI内核中断产生程序如图31所示。

例如，设置定时器0每隔1 ms产生一次超时。第一次1 ms超时后，FIQSIG[2]或IRQSIG[2]置1，并且只能通过写入T0CLR1清0。如果在IRQEN或FIQEN内没有使能定时器0，则FIQSTA/IRQSTA[2]不会置1，也不会产生中断。如果在IRQEN或FIQEN内使能定时器0，则FIQSTA[2]或IRQSTA[2]置1或产生中断（FIQ或IRQ）。

注意，CPSR的IRQ和FIQ位定义仅通过ARM内核而非外设来控制中断识别。例如，如果配置定时器2通过IRQEN产生一个IRQ，则CPSR内的IRQ中断位置1（禁用）并关断ADuC7036。中断发生后，外设唤醒，但ARM内核仍处于关断状态。这相当于POWCON = 0x71。然后ARM内核只能通过复位事件来上电。

表50. IRQ/FIQ MMR位分配

位	描述	注释
0	所有中断逻辑或（限FIQ）	
1	SWI：不用于IRQEN/IRQCLR和FIQEN/FIQCLR。	
2	定时器0	参见“定时器0—使用寿命定时器”部分。
3	定时器1	参见“定时器1”部分。
4	定时器2或唤醒定时器	参见“定时器2—唤醒定时器”部分。
5	定时器3或看门狗定时器	参见“定时器3—看门狗定时器”部分。
6	定时器4或STI定时器	参见“定时器4—STI定时器”部分。
7	LIN硬件	参见“LIN（局域互连网络）接口”部分。
8	Flash/EE中断	参见“Flash/EE控制接口”部分。
9	PLL锁定	参见“系统时钟”部分。
10	ADC	参见“16位 Σ - Δ ADC”部分。
11	UART	参见“UART串行接口”部分。
12	SPI主机	参见“串行外设接口”部分。
13	XIRQ0（GPIO IRQ0）	参见“通用I/O”部分。
14	XIRQ1（GPIO IRQ1）	参见“通用I/O”部分。
15	保留	
16	IRQ3高压IRQ	高压中断；参见“高压外设控制接口”部分。
17	SPI从机	参见“串行外设接口”部分。
18	XIRQ4（GPIO IRQ4）	参见“通用I/O”部分。
19	XIRQ5（GPIO IRQ5）	参见“通用I/O”部分。
20至32	保留	保留。

正常中断 (IRQ) 请求

IRQ请求是允许在IRQ模式下进入处理器内的异常信号。它用于内部和外部事件的通用中断服务。

IRQSTA MMR的所有32位经逻辑或运算后向ARM7TDMI内核发送一个IRQ信号。IRQ专用的四个32位寄存器参见“IRQSTA寄存器到IRQCLR寄存器”部分。

IRQSTA 寄存器

名称: IRQSTA

地址: 0xFFFF0000

默认值: 0x00000000

访问类型: 只读

功能: 该寄存器指示当前的IRQ源状态(见图31)。如果某一位置1,则相应中断源向ARM7TDMI内核产生一个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。

IRQSIG 寄存器

名称: IRQSIG

地址: 0xFFFF0004

默认值: 0x00000000

访问类型: 只读

功能: 该32位寄存器反映不同IRQ源的状态。如果一个外设产生了一个IRQ信号,则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为只读寄存器。

IRQEN 寄存器

名称: IRQEN

地址: 0xFFFF0008

默认值: 0x00000000

访问类型: 读写

功能: 该寄存器指示当前使能屏蔽值。当某一位被置1时,相应的中断源就会使能产生一个IRQ异常信号。当某一位被清0时,相应的中断源就会被禁用或屏蔽,此时就不能产生一个IRQ异常信号了。IRQEN寄存器无法用来禁用中断。

IRQCLR 寄存器

名称: IRQCLR

地址: 0xFFFF000C

访问类型: 只写

功能: 该寄存器允许清除IRQEN寄存器以屏蔽中断源。某一位置1清除IRQEN寄存器的相应位而不影响其他位。当用作一对寄存器时,IRQEN和IRQCLR允许独立操纵使能屏蔽而不需要自动读取修改写入指令。

快速中断请求 (FIQ)

FIQ是允许在FIQ模式下进入处理器内的异常信号。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 MMR的所有32位通过逻辑“或”运算后产生FIQ信号到内核以及FIQ和IRQ寄存器的0位(FIQ源)。

逻辑上FIQEN和FIQCLR不允许一个中断源同时使能IRQ和FIQ屏蔽。另外,FIQEN中某一位置1会清0IRQEN中的相同位。同样,IRQEN中某一位置1也会清0FIQEN中的相同位。一个中断源可以被IRQEN和FIQEN屏蔽禁用。

可编程中断

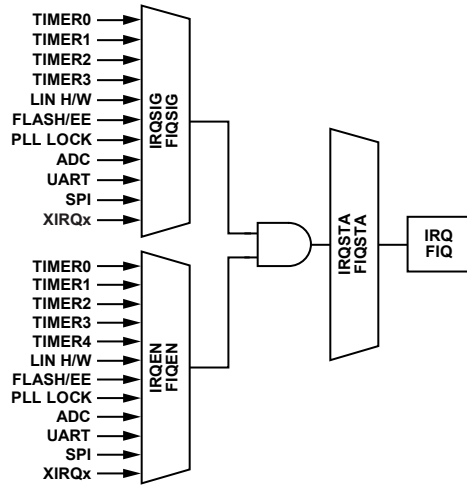
因为可编程中断是无法屏蔽的,所以它们由另外一个寄存器SWICFG控制,通过这个寄存器可以同时写入IRQSTA和IRQSIG寄存器和/或FIQSTA和FIQSIG寄存器。

这个32位的寄存器SWICFG专门用于软件中断(见表51)。这个存储器映射寄存器允许对可编程中断源进行控制。

表51. SWICFG MMR位分配

位	描述
31至3	保留。
2	可编程中断FIQ。 置1或清0, FIQSTA和FIQSIG寄存器的位1也会相应置1或清0。
1	可编程中断IRQ。 置1或清0, IRQSTA和IRQSIG寄存器的位1也会相应置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IRQSTA或FIQSTA寄存器中检测到。



07474-031

图 31. 中断结构

定时器

ADuC7036具有5个通用定时器/计数器。

- 定时器0或使用寿命定时器
- 定时器1或通用定时器
- 定时器2或唤醒定时器
- 定时器3或看门狗定时器
- 定时器4或STI定时器

这5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，计数器从最大值/最小值开始递减/递增直到0或满量程，然后从最大值/最小值重新开始。

当工作在周期模式时，计数器从装载寄存器（TxLD MMR）中的值开始递减或递增到0或满量程，然后重新开始。注意，TxLD MMR要在TxCON MMR前面配置。

计数器的值可随时通过访问寄存器TxVAL读出。定时器可通过写入相应控制寄存器TxCON来启动。

在正常模式下，每次计数器值达到0（递减计数）或满量程（递增计数）时，都会产生一个IRQ。可以通过写任意值到对应定时器（TxCLRI）的清除寄存器来取消IRQ。

此外，定时器0、定时器1及定时器4分别具有一个捕捉寄存器（分别为T0CAP、T1CAP及T4CAP）保持使能IRQ事件的捕捉值。IRQ事件见表52。

表52.定时器事件捕捉

位	描述
0	定时器0或使用寿命定时器
1	定时器1或通用定时器
2	定时器2或唤醒定时器
3	定时器3或看门狗定时器
4	定时器4或STI定时器
5	LIN硬件
6	Flash/EE中断
7	PLL锁定
8	ADC
9	UART
10	SPI主机
11	XIRQ0（GPIO_0）
12	XIRQ1（GPIO_5）
13	保留
14	IRQ3高压中断
15	SPI从机
16	XIRQ4（GPIO_7）；参见“通用I/O”部分
17	XIRQ5（GPIO_8）；参见“通用I/O”部分

定时器0—使用寿命定时器

定时器0是一个带有可编程预分频器的通用48位递增或16位递增/递减计数器定时器。它采用内核时钟或低功耗32.768 kHz振荡器工作，预分频器为1、16、256或32,768。当内核工作在20.48 MHz下，预分频器为1时，产生48.83 ns的最低分辨率。

工作在48位模式时，定时器0从0开始递增计数。当前计数器值从T0VAL0和T0VAL1读取。

工作在16位模式时，定时器0既可递增计数，也可递减计数。16位值可通过写入TOLD来载入到计数器内。当前计数器值从T0VAL0读取。定时器0有一个捕捉寄存器(TOCAP)，它可以被选定的IRQ中断源初始断言所触发。当捕捉寄存器触发时，将当前定时器值复制到TOCAP，同时定时器继续运行。此功能可用来比单独中断服务更能精确地决定事件断言。

定时器0在溢出时从TOLD重新载入值。

定时器0接口包括6个MMR：TOLD、TOCAP、T0VAL0、T0VAL1、T0CLRI和T0CON。TOLD是一个16位寄存器，保持载入计数器的16位值。只能在16位模式下使用。TOCAP是一个16位寄存器，保持使能IRQ事件所捕捉的16位值。只能在16位模式下使用。T0VAL0和T0VAL1分别是16位和32位寄存器，保持16 LSB和32 MSB。二者均为只读寄存器。16位模式使用16位T0VAL0。48位模式同时使用16位T0VAL0和32位T0VAL1。T0CLRI是一个8位寄存器。向其中写入任意值就可以清除中断。只能在16位模式下使用。T0CON是一个配置MMR，如表53所示。

定时器0 载入寄存器

名称：TOLD

地址：0xFFFF0300

默认值：0x0000

访问类型：读写

功能：TOLD0为16位寄存器，可保持载入到计数器的16位值。只能在16位模式下使用。

定时器0 清除寄存器

名称：T0CLRI

地址：0xFFFF0310

访问类型：只写

功能：该8位只写MMR通过用户代码写入（任意值）来清除中断。

定时器0 值寄存器

名称：T0VAL0、T0VAL1

地址：0xFFFF0304、0xFFFF0308

默认值：0x0000、0x00000000

访问类型：只读

功能：T0VAL0和T0VAL1分别是16位和32位寄存器，保持16 LSB和32 MSB。二者均为只读寄存器。16位模式使用16位T0VAL0。48位模式同时使用16位T0VAL0和32位T0VAL1。

定时器0 捕捉寄存器

名称：TOCAP

地址：0xFFFF0314

默认值：0x0000

访问类型：只读

功能：该16位寄存器保持使能IRQ事件所捕捉的16位值。只能在16位模式下使用。

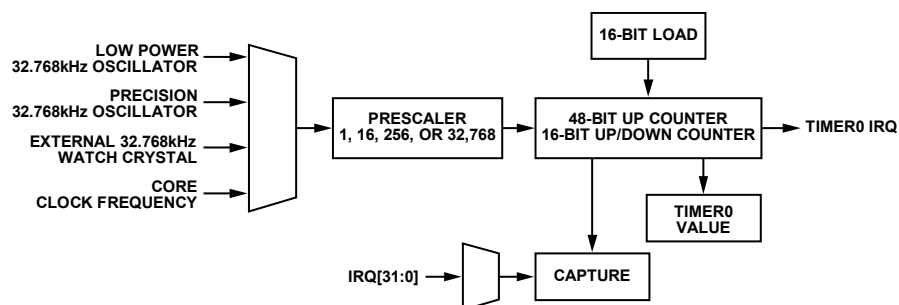


表 32. 定时器0 框图

07474-032

ADuC7036

定时器0 控制寄存器

名称: T0CON

地址: 0xFFFF030C

默认值: 0x00000000

访问类型: 读写

功能: 该32位MMR配置定时器0的工作模式。

Table 53. T0CON MMR位分配

位	描述
31至18	保留。
17	事件选择位。 用户置1, 使能定时器捕捉一个事件。 用户清0, 禁用定时器捕捉一个事件。
16至12	事件选择范围 (0至17)。事件定义参见表52。
11	保留。
10至9	时钟选择。 00 = 内核时钟 (默认)。 01 = 低功耗32.768 kHz振荡器。 10 = 外部32.768 kHz时钟晶体。 11 = 精密32.768 kHz振荡器。
8	递增计数。仅适用于16位模式。 用户置1, 定时器0递增计数。 用户清0, 定时器0递减计数 (默认)。
7	定时器0使能位。 用户置1, 使能定时器0。 用户清0, 禁用定时器0 (默认)。
6	定时器0模式。 用户置1, 选择周期模式。 用户清0, 选择自由模式 (默认)。
5	保留。
4	定时器0工作模式。 0 = 16位工作模式 (默认)。 1 = 48位工作模式。
3至0	预分频器。 0000 = 时钟源/1 (默认)。 0100 = 时钟源/16。 1000 = 时钟源/256。 1111 = 时钟源/32,768。

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。预分频器采用低功耗32.768 kHz振荡器、内核时钟或两个外部GPIO中的一个工作。时钟源分频系数共有1、16、256或32,768四种。当内核工作在20.48 MHz下且CD = 0，预分频为1（忽略外部GPIO）时，产生48.83 ns的最小分辨率。

计数器可以是标准的32位数或时间格式：小时：分钟：秒：百分之一秒。

定时器1有一个捕捉寄存器(TICAP)，它可以被选定的IRQ中断源初始断言所触发。触发时，将当前定时器值复制到TICAP，同时定时器继续运行。此功能可用于更精确地决定事件断言。

定时器1接口包括5个MMR：TILD、TIVAL、TICAP、TICLRI和TICON。TILD、TIVAL和TICAP都是32位寄存器，保持32位无符号整数。TIVAL和TICAP是只读的。TICLRI是一个8位寄存器。向其中写入任意值就可以清除定时器1中断。TICON是一个配置MMR，如表54所示。

定时器1具有一个后分频器，允许用户在1和256之间进行定时器1超时计数。要激活后分频器，用户需将位23置1并在TICON的位[24:31]内写入所需计数值。当达到超时次数时，如果TICON[18]置1，则定时器1就会产生一个中断。

注意，当器件处于低功耗模式且定时器1从GPIO或低功耗振荡器源接收时钟信号时，定时器1仍继续工作。

定时器1在溢出时从TILD重新载入值。

定时器1 载入寄存器

名称：TILD

地址：0xFFFF0320

默认值：复位后包含组装批次ID (0x00000000) 的高16位。

访问类型：读写

功能：该32位寄存器保持计数器的32位载入值。

定时器1 清除寄存器

名称：TICLRI

地址：0xFFFF032C

访问类型：只写

功能：该8位只写MMR通过用户代码写入（任意值）来清除中断。

定时器1 值寄存器

名称：TIVAL

地址：0xFFFF0324

默认值：0xFFFFFFFF

访问类型：只读

功能：该32位寄存器保持定时器1的当前值。

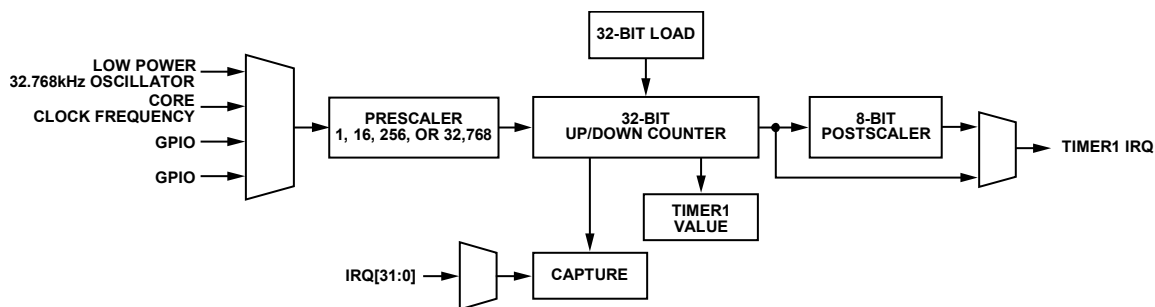


图 33. 定时器1 框图

07474-033

ADuC7036

定时器1 捕捉寄存器

名称: T1CAP

地址: 0xFFFF0330

默认值: 0x00000000

访问类型: 只读

功能: 该32位寄存器保持使能IRQ事件所捕捉的32位值。

定时器1 控制寄存器

名称: T1CON

地址: 0xFFFF0328

默认值: 0x01000000

访问类型: 读写

功能: 该32位MMR配置定时器1的工作模式。

表54. T1CON MMR位分配

位	描述
31至24	8位后分频器。 向后分频器写入8位二进制值。写入0视为1。 可从中读取8位计数器当前值。
23	定时器1使能后分频器。 置1使能定时器1后分频器。使能后, 在T1ILD所定义的T1CON[31:24]循环次数结束后, 产生中断。 清0禁用定时器1后分频器。
22至20	保留。这些位保留, 应由用户代码写入0。
19	后分频器比较标志。只读。 如果定时器1溢出次数等于后分频器写入值, 则置1。
18	定时器1中断源。 置1, 选择从后分频器计数器产生中断。 清0, 选择直接从定时器1产生中断。
17	事件选择位。 用户置1, 使能定时器捕捉一个事件。用户清0, 禁用定时器捕捉一个事件。
16至12	事件选择范围(0至17)。事件见表52。
11至9	时钟选择。 000 = 内核时钟(默认)。 001 = 低功耗32.768 kHz振荡器。 010 = GPIO_8。 011 = GPIO_5。
8	递增计数。 用户置1, 定时器1递增计数。 用户清0, 定时器1递减计数(默认)。
7	定时器1使能位。 用户置1, 使能定时器1。 用户清0, 禁用定时器1(默认)。
6	定时器1模式。 用户置1, 选择周期模式。 用户清0, 选择自由模式(默认)。
5至4	数据格式。 00 = 二进制(默认)。 01 = 保留。 10 = 小时: 分钟: 秒: 百分之一秒(23至0小时)。 11 = 小时: 分钟: 秒: 百分之一秒(255至0小时)。
3至0	预分频器。 0000 = 时钟源/1(默认)。 0100 = 时钟源/16。 1000 = 时钟源/256。 1111 = 时钟源/32,768。

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。预分频器直接从四个时钟源中的一个接收时钟信号：内核时钟（默认选择）、低功耗32.768 kHz振荡器、外部32.768 kHz时钟晶体或精密32.768 kHz振荡器。所选时钟源分频系数共有1、16、256或32,768四种。当内核时钟被禁用时，唤醒定时器仍继续运行。当内核工作在20.48 MHz下且CD = 0，预分频为1时，产生48.83 ns的最低分辨率。

计数器可以是标准的32位数模式或时间格式：小时：分钟：秒：百分之一秒。

定时器2在溢出时从T2LD重新载入。

定时器2接口包括4个MMR：T2LD、T2VAL、T2CLRI和T2CON。T2LD和T2VAL都是32位寄存器，保持32位无符号整数。T2VAL是一个只读寄存器。T2CLRI是一个8位寄存器。向其中写入任意值就可以清除定时器2中断。T2CON是一个配置MMR，如表55所示。

定时器2 载入寄存器

名称：T2LD

地址：0xFFFF0340

默认值：0x00000000

访问类型：读写

功能：该32位寄存器保持计数器的32位载入值。

定时器2 清除寄存器

名称：T2CLRI

地址：0xFFFF034C

访问类型：只写

功能：该8位只写MMR通过用户代码写入（任意值）来清除中断。

定时器2 值寄存器

名称：T2VAL

地址：0xFFFF0344

默认值：0xFFFFFFFF

访问类型：只读

功能：该32位寄存器保持定时器2的当前值。

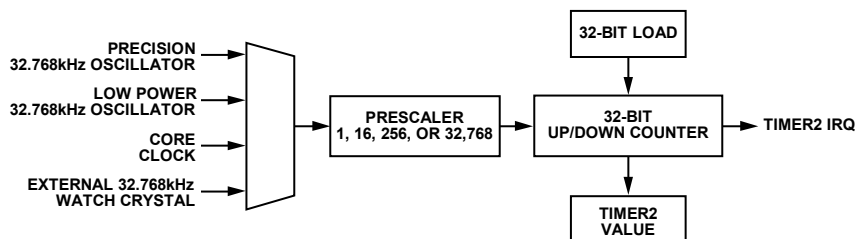


图34. 定时器2 框图

ADuC7036

定时器2 控制寄存器

名称: T2CON

地址: 0xFFFF0348

默认值: 0x0000

访问类型: 读写

功能: 该16位MMR配置定时器2的工作模式。

表55. T2CON MMR位分配

位	描述
15至11	保留。
10至9	时钟源选择。 00 = 内核时钟（默认）。 01 = 低功耗（32.768 kHz）振荡器。 10 = 外部32.768 kHz时钟晶体。 11 = 精密32.768 kHz振荡器。
8	递增计数。 用户置1，定时器2递增计数。 用户清0，定时器2递减计数（默认）。
7	定时器2使能位。 用户置1，使能定时器2。 用户清0，禁用定时器2（默认）。
6	定时器2模式。 用户置1，选择周期模式。 用户清0，选择自由模式（默认）。
5至4	数据格式。 00 = 二进制（默认）。 01 = 保留。 10 = 小时：分钟：秒：百分之一秒（23至0小时）。只在采用32 kHz时钟时有效。 11 = 小时：分钟：秒：百分之一秒（255至0小时）。只在采用32 kHz时钟时有效。
3至0	预分频器。 0000 = 时钟源/1（默认）。 0100 = 时钟源/16。 1000 = 时钟源/256。此设置应采用小时：分钟：秒：百分之一秒格式和定时器2配合使用。见此表中格式位（[5:4]）的10和11置1。 1111 = 时钟源/32,768。

定时器3—看门狗定时器

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

定时器3在溢出时从T3LD重新载入。

正常模式

在正常模式下的定时器3和在16位工作模式下的定时器0相同，但时钟源除外。时钟源采用低功耗32.768 kHz振荡器，分频系数共有1、16或256三种。

看门狗模式

看门狗模式通过设置T3CON[5]进入。定时器3从T3LD寄存器的超时值开始递减计数直到0。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。

Flash/EE存储器完成一次页擦除周期和内核执行需要20 ms，为避免与此发生任何冲突，用户软件不得将超时周期设为30 ms以下。

T3VAL到达0后，取决于T3CON[1]状态，就会产生复位或中断。要避免这种情况，就必须在T3VAL达到0之前将任意值写入T3CLRI。将T3LD值重新载入计数器内后就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。这两个寄存器无法被修改，除非上电复位看门狗定时器。任何其他复位事件，看门狗仍会继续计数。看门狗定时器应在用户代码的最初行内配置以免陷入看门狗复位的无限循环内。用户软件只能配置最低30 ms的超时周期。

定时器3在JTAG调试访问期间自动暂停并只在JTAG放弃内核控制权后才会重新开始计数。默认情况下，定时器3在断电期间仍会继续计数。但可通过将T3CON位0置1来禁用。建议采用默认值，即在断电期间仍允许看门狗定时器继续计数。

定时器3 接口

定时器3接口包括4个MMR：T3LD、T3VAL、T3CLRI和T3CON。T3LD和T3VAL都是16位寄存器（0至15位），保持16位无符号整数。T3VAL是一个只读寄存器。T3CLRI是一个8位寄存器。正常模式下向这个寄存器写入任意值会清除定时器3中断；在看门狗模式下向这个寄存器写入任意值就会重新开始一个超时周期。T3CON是配置MMR，如表56所示。

定时器3 载入寄存器

名称：T3LD

地址：0xFFFF0360

默认值：0x0040

访问类型：读写

功能：该16位MMR保持定时器3的重新载入值。

定时器3 清除寄存器

名称：T3CLRI

地址：0xFFFF036C

访问类型：只写

功能：在看门狗模式下该8位只写MMR通过用户代码写入（任意值）刷新（重新载入）定时器3以防止看门狗定时器复位事件发生。

定时器3 值寄存器

名称：T3VAL

地址：0xFFFF0364

默认值：0x0040

访问类型：只读

功能：该16位只读MMR保持定时器3的当前计数值。

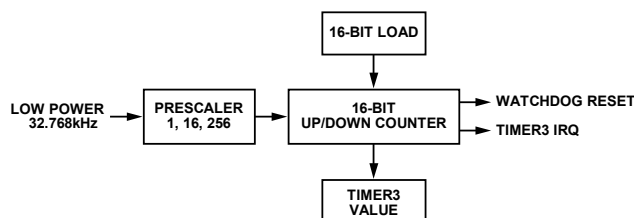


图 35. 定时器3 框图

ADuC7036

定时器3 控制寄存器

名称: T3CON

地址: 0xFFFF0368

默认值: 0x0000

访问类型: 读写

功能: 该16位只读MMR配置定时器3的工作模式, 如表56所示。

表56. T3CON MMR位分配

位	描述
15至9	保留。这些位保留, 应由用户代码写入0。
8	递增/递减计数使能。 用户代码置1, 递增计数。 用户代码清0, 递减计数。
7	定时器3使能。 用户代码置1, 使能定时器3。 用户代码清0, 禁用定时器3。
6	定时器3工作模式。 用户代码置1, 定时器3采用周期模式。 用户代码清0, 定时器3采用自由模式。
5	看门狗定时器模式使能。 用户代码置1, 使能看门狗模式。 用户代码清0, 禁用看门狗模式。
4	保留。这些位保留, 应由用户代码写入0。
3至2	定时器3时钟 (32.768 kHz) 预分频器。 00 = 时钟源/1 (默认)。 01 = 时钟源/16。 10 = 时钟源/256。 11 = 保留。
1	看门狗定时器IRQ使能。 用户代码置1, 在看门狗到达0时产生IRQ中断而非复位。 用户代码清0禁用IRQ选项。
0	PD_OFF。 用户代码置1, 在利用POWCON MMR位4使外设断电时停止定时器3。 用户代码清0, 在利用POWCON MMR位4使外设断电时使能定时器3。

定时器4—STI定时器

定时器4是一个带有可编程预分频器的16位通用定时器，工作时可递增计数或递减计数。定时器4采用内核时钟或低功耗32.768 kHz振荡器工作，预分频系数共有1、16、256或32,768四种。

定时器4有一个捕捉寄存器(T4CAP)，它可以被选定的IRQ中断源初始断言所触发。触发时，将当前定时器值复制到T4CAP，同时定时器继续运行。此功能可用来更精确地决定事件断言。

定时器4还可用来驱动串行测试接口（STI）外设。

定时器4接口包括5个MMR：T4LD、T4VAL、T4CAP、T4CLRI和T4CON。T4LD、T4VAL和T4CAP都是16位寄存器，保持16位无符号整数。T4VAL和T4CAP是只读的。T4CLRI是一个8位寄存器。向其中写入任意值就可以清除中断。T4CON是一个配置MMR，如表57所示。

定时器4 载入寄存器

名称：T4LD

地址：0xFFFF0380

默认值：0x0000

访问类型：读写

功能：该16位寄存器保持计数器的16位载入值。

定时器4 清除寄存器

名称：T4CLRI

地址：0xFFFF038C

访问类型：只写

功能：该8位只写MMR通过用户代码写入（任意值）来清除中断。

定时器4 值寄存器

名称：T4VAL

地址：0xFFFF0384

默认值：0xFFFF

访问类型：只读

功能：该16位寄存器保持定时器4的当前值。

定时器4 捕捉寄存器

名称：T4CAP

地址：0xFFFF0390

默认值：0x0000

访问类型：只读

功能：该16位寄存器保持被使能IRQ事件所捕捉的32位值。

定时器4 控制寄存器

名称：T4CON

地址：0xFFFF0388

默认值：0x00000000

访问类型：读写

功能：该32位MMR配置定时器4的工作模式。

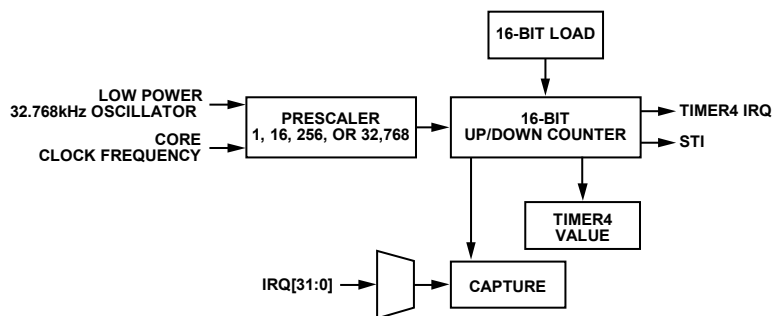


图 36. 定时器4 框图

07474-036

ADuC7036

表57. T4CON MMR位分配

位	描述
31至18	保留。
17	事件选择位。 用户置1, 使能定时器捕捉一个事件。 用户清0, 禁用定时器捕捉一个事件。
16至12	事件选择范围 (0至17)。事件见表52。
11至10	保留。
9	时钟选择。 0 = 内核时钟 (默认)。 1 = 低功耗32.768 kHz振荡器。
8	递增计数。 用户置1, 定时器4递增计数。 用户清0, 定时器4递减计数 (默认)。
7	定时器4使能位。 用户置1, 使能定时器0。 用户清0, 禁用定时器0 (默认)。
6	定时器4模式。 用户置1, 选择周期模式。 用户清0, 选择自由模式 (默认)。
5至4	保留。
3至0	预分频器。 0000 = 时钟源/1 (默认)。 0100 = 时钟源/16。 1000 = 时钟源/256。 1111 = 时钟源/32,768。

通用I/O

ADuC7036共有9个通用双向输入/输出（GPIO）引脚。一般情况下，通过用户代码设置，许多引脚拥有多种功能。引脚默认采用GPIO模式。所有GPIO引脚都具有一个内部上拉电阻，吸电流能力为0.8 mA，源电流能力为0.1 mA。

9个GPIO引脚分成三个端口：端口0、端口1和端口2。端口0为5位；端口1和端口2分别为2位。每个端口的引脚定义详见表58。典型GPIO结构如图37所示。

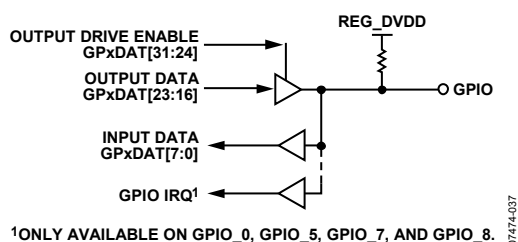


图37. 典型GPIO结构

外部中断只存在于GPIO_0、GPIO_5、GPIO_7和GPIO_8上，中断为电平触发且高电平有效。这些中断不能被锁存；因此中断源必须一直存在直到查询IRQSTA或FIQSTA。要确保正确识别，中断源必须有效持续至少一个CD分频内核时钟。

所有端口引脚通过三组（每个端口一组）四个端口专用MMR配置和控制如下：

- GPxCON: Portx控制寄存器
- GPxDAT: Portx配置和数据寄存器
- GPxSET: Portx数据置1
- GPxCLR: Portx数据清0

其中x为端口编号（0、1或2）。

在正常操作期间，用户代码可利用这些通用寄存器来控制外部GPIO引脚的功能和状态。所有引脚在断电(POWCON)模式期间保持其外部电平（高或低）。

ADuC7036

表58. 外部GPIO引脚到内部端口信号分配

端口	GPIO引脚	端口信号	功能（由GPxCON定义）
端口0	GPIO_0	P0.0	通用I/O。
		IRQ0	外部中断请求0。
		\overline{SS}	SPI从机选择I/O。
	GPIOJ	P0.1	通用I/O。
		SCLK	SPI串行时钟I/O。
GPIO_2	P0.2	通用I/O。	
GPIO_3	MISO	SPI主机输入、从机输出。	
	P0.3	通用I/O。	
	MOSI	SPI主机输出、从机输入。	
	GPIO_4	P0.4	通用I/O。
ECLK		2.56 MHz时钟输出。	
P0.5 ¹		高压串行接口。	
P0.6 ¹	高压串行接口。		
端口1	GPIO_5	P1.0	通用I/O。
		IRQ1	外部中断请求1。
RxD		UART引脚。	
GPIO_6	P1.1	通用I/O。	
TxD	UART引脚。		
端口2	GPIO_7	Port 2.0	通用I/O。
		IRQ4	外部中断请求4。
		LIN输出引脚 ²	用来从LIN引脚直接读取以进行一致性测试。
	GPIO_8	P2.1	通用I/O。
		IRQ5	外部中断请求5。
LIN高压输入引脚 ²	用来直接驱动LIN引脚以进行一致性测试。		
GPIO_11 ²	P2.4 ²	通用I/O。	
LINRX ²	LIN输入引脚。		
GPIO_12 ²	P2.5 ²	通用I/O。	
LINTX ²	LIN输出引脚。		
GPIO_13 ¹	P2.6 ¹	通用I/O; STI数据输出。	

¹ 这些信号只是内部信号，不能出现在外部引脚上。这些引脚和HVCON一起用作高压接口电路的两线式接口。

² 这些引脚/信号只是内部信号，不能出现在外部引脚上。用来提供外部引脚诊断写入（GPIO_12）和回读（GPIO_11）能力。

GPIO端口0 控制寄存器

名称: GP0CON

地址: 0xFFFF0D00

默认值: 0x11100000

访问类型: 读写

功能: 该32位寄存器选择每个端口0引脚的功能。

表59. GPOCON MMR位分配

位	描述
31至29	保留。这些位保留，应由用户代码写入0。
28	保留。这些位保留，应由用户代码写入1。
27至25	保留。这些位保留，应由用户代码写入0。
24	内部P0.6使能位。必须由用户软件置1，在使用HVCON和HVDAT寄存高压接口前使能高压串行接口。
23至21	保留。这些位保留，应由用户代码写入0。
20	内部P0.5使能位。必须由用户软件置1，在使用HVCON和HVDAT寄存高压接口前使能高压串行接口。
19至17	保留。这些位保留，应由用户代码写入0。
16	GPIO_4功能选择位。 用户代码置1，将GPIO_4引脚配置成ECLK，从而在此引脚上使能2.56 MHz时钟输出。用户代码清0，将GPIO_4引脚配置成通用I/O（GPIO）引脚。
15至13	保留。这些位保留，应由用户代码写入0。
12	GPIO_3功能选择位。 用户代码置1，将GPIO_3引脚配置成SPI端口的MOSI（主机输出从机输入）。用户代码清0，将GPIO_3引脚配置成通用I/O（GPIO）引脚。
11至9	保留。这些位保留，应由用户代码写入0。
8	GPIO_2功能选择位。 用户代码置1，将GPIO_2引脚配置成SPI端口的MISO（主机输入从机输出）。用户代码清0，将GPIO_2引脚配置成通用I/O（GPIO）引脚。
7至5	保留。这些位保留，应由用户代码写入0。
4	GPIO_1功能选择位。 用户代码置1，将GPIO_1引脚配置成SPI端口的SCLK（串行时钟I/O）。 用户代码清0，将GPIO_1引脚配置成通用I/O（GPIO）引脚。
3至1	保留。这些位保留，应由用户代码写入0。
0	GPIO_0功能选择位。 用户代码置1，将GPIO_0引脚配置成SPI端口的 \overline{SS} （串行时钟I/O）。 用户代码清0，将GPIO_0引脚配置成通用I/O（GPIO）引脚。

ADuC7036

GPIO端口1 控制寄存器

名称: GP1CON

地址: 0xFFFF0D04

默认值: 0x10000000

访问类型: 读写

功能: 该32位MMR选择每个端口1引脚的功能。

表60. GP1CON MMR位分配

位	描述
31至5	保留。这些位保留，应由用户代码写入0。
4	GPIO_6功能选择位。 用户代码置1，将GPIO_6引脚配置为TxD，为UART串行端口发送数据。 用户代码清0，将GPIO_6引脚配置为通用I/O（GPIO）引脚。
3至1	保留。这些位保留，应由用户代码写入0。
0	GPIO_5功能选择位。 用户代码置1，将GPIO_5引脚配置为RxD。为UART串行端口接收数据。 用户代码清0，将GPIO_5引脚配置为通用I/O（GPIO）引脚。

GPIO端口2 控制寄存器

名称: GP2CON

地址: 0xFFFF0D08

默认值: 0x01000000

访问类型: 读写

功能: 该32位MMR选择每个端口2引脚的功能。

表61. GP2CON MMR位分配

位	描述
31至25	保留。这些位保留，应由用户代码写入0。
24	GPIO_13功能选择位。 用户代码置1，将STI数据输出送到STI引脚。 若该位由用户代码清0，即使STI接口正常使能，也不会将STI数据送到外部STI引脚。
23至21	保留。这些位保留，应由用户代码写入0。
20	GPIO_12功能选择位。 用户代码置1，将UART TxD（发送数据）送到LIN/BSD数据引脚。此配置在LIN模式下使用。用户代码清0将LIN/BSD发送数据送到一个内部通用I/O（GPIO_12）焊盘，然后通过GP2DAT MMR写入。此配置在BSD模式下用来允许用户代码将输出数据写入到BSD接口，还可用来支持写入高压I/O引脚的诊断功能（见表75中的HVCFG1[2:0]）。
19至17	保留。这些位保留，应由用户代码写入0。
16	GPIO_11功能选择位。 用户代码置1，将输入数据从LIN/BSD接口同时送到LIN/BSD硬件时序/同步逻辑和UART RxD（接收数据）。此模式必须在使用LIN或BSD模式下通过用户代码置1。用户代码清0，内部禁用LIN/BSD输入数据路径。此时GPIO_11用来支持对所有外部高压I/O引脚的诊断回读（见表75中的HVCFG1[2:0]）。
15至5	保留。这些位保留，应由用户代码写入0。
4	GPIO_8功能选择位。 用户代码置1，将LIN/BSD输入数据送到GPIO_8引脚。此模式可将LIN收发器作为单独元件来驱动而不需要和MCU或UART进行任何交互。用户代码清0，将GPIO_8引脚配置成通用I/O（GPIO）引脚。
3至1	保留。这些位保留，应由用户代码写入0。
0	GPIO_7功能选择位。 用户代码置1，将片内LIN收发器驱动到GPIO_7引脚的数据送到LIN/BSD引脚加以输出。此模式可将LIN收发器作为单独元件来驱动而不需要和MCU或UART进行任何交互。 用户代码清0，将GPIO_7引脚配置成通用I/O（GPIO）引脚。

ADuC7036

GPIO端口0 数据寄存器

名称: GP0DAT

地址: 0xFFFF0D20

默认值: 0x000000XX

访问类型: 读写

功能: 该32位MMR配置分配给端口0的GPIO引脚的方向(见表58)。还设置配置为输出端的GPIO引脚的输出值并读取配置为输入端的GPIO引脚的状态。

表62. GPODAT MMR位分配

位	描述
31至29	保留。这些位保留, 应由用户代码写入0。
28	端口0.4方向选择位。 用户代码置1, 将分配给端口0.4的GPIO引脚配置为输出端。 用户代码清0, 将分配给端口0.4的GPIO引脚配置为输入端。
27	端口0.3方向选择位。 用户代码置1, 将分配给端口0.3的GPIO引脚配置为输出端。 用户代码清0, 将分配给端口0.3的GPIO引脚配置为输入端。
26	端口0.2方向选择位。 用户代码置1, 将分配给端口0.2的GPIO引脚配置为输出端。 用户代码清0, 将分配给端口0.2的GPIO引脚配置为输入端。
25	端口0.1方向选择位。 用户代码置1, 将分配给端口0.1的GPIO引脚配置为输出端。 用户代码清0, 将分配给端口0.1的GPIO引脚配置为输入端。
24	端口0.0方向选择位。 用户代码置1, 将分配给端口0.0的GPIO引脚配置为输出端。 用户代码清0, 将分配给端口0.0的GPIO引脚配置为输入端。
23至21	保留。这些位保留, 应由用户代码写入0。
20	端口0.4数据输出。写入到该位的值直接出现在分配给端口0.4的GPIO引脚上。
19	端口0.3数据输出。写入到该位的值直接出现在分配给端口0.3的GPIO引脚上。
18	端口0.2数据输出。写入到该位的值直接出现在分配给端口0.2的GPIO引脚上。
17	端口0.1数据输出。写入到该位的值直接出现在分配给端口0.1的GPIO引脚上。
16	端口0.0数据输出。写入到该位的值直接出现在分配给端口0.0的GPIO引脚上。
15至5	保留。这些位保留, 应由用户代码写入0。
4	端口0.4数据输入。只读, 反映分配给端口0.4的GPIO引脚的当前状态。应由用户代码写入0。
3	端口0.3数据输入。只读, 反映分配给端口0.3的GPIO引脚的当前状态。应由用户代码写入0。
2	端口0.2数据输入。只读, 反映分配给端口0.2的GPIO引脚的当前状态。应由用户代码写入0。
1	端口0.1数据输入。只读, 反映分配给端口0.1的GPIO引脚的当前状态。应由用户代码写入0。
0	端口0.0数据输入。只读, 反映分配给端口0.0的GPIO引脚的当前状态。应由用户代码写入0。

GPIO端口1 数据寄存器

名称: GP1DAT

地址: 0xFFFF0D30

默认值: 0x000000XX

访问类型: 读写

功能: 该32位MMR配置分配给端口1的GPIO引脚的方向(见表58)。还设置配置为输出端的GPIO引脚的输出值并读取配置为输入端的GPIO引脚的状态。

表63. GP1DAT MMR位分配

位	描述
31至26	保留。这些位保留, 应由用户代码写入0。
25	端口1.1方向选择位。 用户代码置1, 将分配给端口1.1的GPIO引脚配置为输出端。 用户代码清0, 将分配给端口1.1的GPIO引脚配置为输入端。
24	端口1.0方向选择位。 用户代码置1, 将分配给端口1.0的GPIO引脚配置为输出端。 用户代码清0, 将分配给端口1.0的GPIO引脚配置为输入端。
23至18	保留。这些位保留, 应由用户代码写入0。
17	端口1.1数据输出。写入到该位的值直接出现在分配给端口1.1的GPIO引脚上。
16	端口1.0数据输出。写入到该位的值直接出现在分配给端口1.0的GPIO引脚上。
15至2	保留。这些位保留, 应由用户代码写入0。
1	端口1.1数据输入。只读, 反映分配给端口1.1的GPIO引脚的当前状态。应由用户代码写入0。
0	端口1.0数据输入。只读, 反映分配给端口1.0的GPIO引脚的当前状态。应由用户代码写入0。

ADuC7036

GPIO 端口 2 数据寄存器

名称: GP2DAT

地址: 0xFFFF0D40

默认值: 0x000000XX

访问类型: 读写

功能: 该32位MMR配置分配给端口2的GPIO引脚的方向(见表58)。还设置配置为输出端的GPIO引脚的输出值并读取配置为输入端的GPIO引脚的状态。

表64. GP2DAT MMR位分配

位	描述
31	保留。这些位保留, 应由用户代码写入0。
30	端口2.6方向选择位。 用户代码置1, 将分配给端口2.6的GPIO引脚配置为输出端。 用户代码清0, 将分配给端口2.6的GPIO引脚配置为输入端。
29	端口2.5方向选择位。 用户代码置1, 将分配给端口2.5的GPIO引脚配置为输出端。此配置用来支持写入高压I/O引脚的诊断功能。 用户代码清0, 将分配给端口2.5的GPIO引脚配置为输入端。
28	端口2.4方向选择位。 用户代码置1, 将分配给端口2.4的GPIO引脚配置为输出端。 用户代码清0, 将分配给端口2.4的GPIO引脚配置为输入端。此配置用来支持回读高压I/O引脚的诊断功能(见HVCFG1[2:0])。
27至26	保留。这些位保留, 应由用户代码写入0。
25	端口2.1方向选择位。 用户代码置1, 将分配给端口2.1的GPIO引脚配置为输出端。 用户代码清0, 将分配给端口2.1的GPIO引脚配置为输入端。
24	端口2.0方向选择位。 用户代码置1, 将分配给端口2.0的GPIO引脚配置为输出端。 用户代码清0, 将分配给端口2.0的GPIO引脚配置为输入端。
23	保留。这些位保留, 应由用户代码写入0。
22	端口2.6数据输出。写入到该位的值直接出现在分配给端口2.6的GPIO引脚上。
21	端口2.5数据输出。写入到该位的值直接出现在分配给端口2.5的GPIO引脚上。
20至18	保留。这些位保留, 应由用户代码写入0。
17	端口2.1数据输出。写入到该位的值直接出现在分配给端口2.1的GPIO引脚上。
16	端口2.0数据输出。写入到该位的值直接出现在分配给端口2.0的GPIO引脚上。
15至7	保留。这些位保留, 应由用户代码写入0。
6	端口2.6数据输入。只读, 反映分配给端口2.6的GPIO引脚的当前状态。应由用户代码写入0。
5	端口2.5数据输入。只读, 反映分配给端口2.5的GPIO引脚的当前状态。应由用户代码写入0。
4	端口2.4数据输入。只读, 反映分配给端口2.4的GPIO引脚的当前状态。应由用户代码写入0。
3至2	保留。这些位保留, 应由用户代码写入0。
1	端口2.1数据输入。只读, 反映分配给端口2.1的GPIO引脚的当前状态。应由用户代码写入0。
0	端口2.0数据输入。只读, 反映分配给端口2.0的GPIO引脚的当前状态。应由用户代码写入0。

GPIO端口0 设置寄存器

名称: GP0SET

地址: 0xFFFF0D24

访问类型: 只写

功能: 该32位MMR允许用户代码对外部GPIO引脚分别进行位寻址以仅将其单独拉高。用户代码可利用GP0SET MMR来实现, 不必修改或维持GPIO引脚状态(使用GP0DAT时用户代码要求)。

表65. GP0SET MMR位分配

位	描述
31至21	保留。这些位保留, 应由用户代码写入0。
20	端口0.4设置位。 用户代码置1, 将外部GPIO_4引脚拉高。 用户软件清0不会影响外部GPIO_4引脚。
19	端口0.3设置位。 用户代码置1, 将外部GPIO_3引脚拉高。 用户软件清0不会影响外部GPIO_3引脚。
18	端口0.2设置位。 用户代码置1, 将外部GPIO_2引脚拉高。 用户软件清0不会影响外部GPIO_2引脚。
17	端口0.1设置位。 用户代码置1, 将外部GPIO_1引脚拉高。 用户软件清0不会影响外部GPIO_1引脚。
16	端口0.0设置位。 用户代码置1, 将外部GPIO_0引脚拉高。 用户软件清0不会影响外部GPIO_0引脚。
15至0	保留。这些位保留, 应由用户代码写入0。

GPIO端口1 设置寄存器

名称: GP1SET

地址: 0xFFFF0D34

访问类型: 只写

功能: 该32位MMR允许用户代码对外部GPIO引脚分别进行位寻址以仅将其单独拉高。用户代码可利用GP1SET MMR来实现, 不必修改或维持GPIO引脚状态(使用GP1DAT时用户代码要求)。

表66. GP1SET MMR位分配

位	描述
31至18	保留。这些位保留, 应由用户代码写入0。
17	端口1.1设置位。 用户代码置1, 将外部GPIO_6引脚拉高。 用户软件清0不会影响外部GPIO_6引脚。
16	端口1.0设置位。 用户代码置1, 将外部GPIO_5引脚拉高。 用户软件清0不会影响外部GPIO_5引脚。
15至0	保留。这些位保留, 应由用户代码写入0。

ADuC7036

GPIO端口2 设置寄存器

名称: GP2SET

地址: 0xFFFF0D44

访问类型: 只写

功能: 该32位MMR允许用户代码对外部GPIO引脚分别进行位寻址以仅将其单独拉高。用户代码可利用GP2SET MMR来实现, 不必修改或维持GPIO引脚状态(使用GP2DAT时用户代码要求)。

表67. GP2SET MMR位分配

位	描述
31至23	保留。这些位保留, 应由用户代码写入0。
22	端口2.6设置位。 用户代码置1, 将外部GPIO_13引脚拉高。 用户软件清0不会影响外部GPIO_13引脚。
21	端口2.5设置位。 用户代码置1, 将外部GPIO_12引脚拉高。 用户软件清0不会影响外部GPIO_12引脚。
20至18	保留。这些位保留, 应由用户代码写入0。
17	端口2.1设置位。 用户代码置1, 将外部GPIO_8引脚拉高。 用户软件清0不会影响外部GPIO_8引脚。
16	端口2.0设置位。 用户代码置1, 将外部GPIO_7引脚拉高。 用户软件清0不会影响外部GPIO_7引脚。
15至0	保留。这些位保留, 应由用户代码写入0。

GPIO端口0 清除寄存器

名称: GP0CLR

地址: 0xFFFF0D28

访问类型: 只写

功能: 该32位MMR允许用户代码对外部GPIO引脚分别进行位寻址以仅将其单独清除拉低。用户代码可利用GP0SET MMR来实现, 不必修改或维持GPIO引脚状态(使用GP0DAT时用户代码要求)。

表68. GPOCLR MMR位分配

位	描述
31至21	保留。这些位保留, 应由用户代码写入0。
20	端口0.4清除位。 用户代码置1, 将外部GPIO_4引脚清除拉低。 用户软件清0不会影响外部GPIO_4引脚。
19	端口0.3清除位。 用户代码置1, 将外部GPIO_3引脚清除拉低。 用户软件清0不会影响外部GPIO_3引脚。
18	端口0.2清除位。 用户代码置1, 将外部GPIO_2引脚清除拉低。 用户软件清0不会影响外部GPIO_2引脚。
17	端口0.1清除位。 用户代码置1, 将外部GPIO_1引脚清除拉低。 用户软件清0不会影响外部GPIO_1引脚。
16	端口0.0清除位。 用户代码置1, 将外部GPIO_0引脚清除拉低。 用户软件清0不会影响外部GPIO_0引脚。
15至0	保留。这些位保留, 应由用户代码写入0。

GPIO端口1 清除寄存器

名称: GP1CLR

地址: 0xFFFF0D38

访问类型: 只写

功能: 该32位MMR允许用户代码对外部GPIO引脚分别进行位寻址以仅将其单独清除拉低。用户代码可利用GP1CLR MMR来实现, 不必修改或维持GPIO引脚状态(使用GP1DAT时用户代码要求)。

表69. GP1CLR MMR位分配

位	描述
31至18	保留。这些位保留, 应由用户代码写入0。
17	端口1.1清除位。 用户代码置1, 将外部GPIO_6引脚清除拉低。 用户软件清0不会影响外部GPIO_6引脚。
16	端口1.0清除位。 用户代码置1, 将外部GPIO_5引脚清除拉低。 用户软件清0不会影响外部GPIO_5引脚。
15至0	保留。这些位保留, 应由用户代码写入0。

GPIO端口2 清除寄存器

名称: GP2CLR

地址: 0xFFFF0D48

访问类型: 只写

功能: 该32位MMR允许用户代码对外部GPIO引脚分别进行位寻址以仅将其单独清除拉低。用户代码可利用GP2CLR MMR来实现, 不必修改或维持GPIO引脚状态(采用GP2DAT时用户代码要求)。

表70. GP2CLR MMR位分配

位	描述
31至23	保留。这些位保留, 应由用户代码写入0。
22	端口2.6清除位。 用户代码置1, 将外部GPIO_13引脚清除拉低。 用户软件清0不会影响外部GPIO_8引脚。
21	端口2.5清除位。 用户代码置1, 将外部GPIO_12引脚清除拉低。 用户软件清0不会影响外部GPIO_7引脚。
20至18	保留。这些位保留, 应由用户代码写入0。
17	端口2.1清除位。 用户代码置1, 将外部GPIO_8引脚清除拉低。 用户软件清0不会影响外部GPIO_8引脚。
16	端口2.0清除位。 用户代码置1, 将外部GPIO_7引脚清除拉低。 用户软件清0不会影响外部GPIO_7引脚。
15至0	保留。这些位保留, 应由用户代码写入0。

高压外设控制接口

ADuC7036集成了多个高压电路功能，它们通过包括两个MMR（HVCON和HVDAT）的寄存接口来加以监控。HVCON寄存器充当命令字节解释器，允许微控制器对四个高压状态或配置寄存器中的一个进行8位数据读写操作（数值存放在HVDAT内）。这些高压寄存器不是MMR，但通常被称为间接寄存器；即只能通过HVCON和HVDAT MMR间接访问（如名称所暗示）。

HVCON寄存器和间接高压寄存器的物理接口采用基于2.56 MHz串行时钟的两线（数据和时钟）式串行接口。因此，从MCU内核将命令写入HVCON到该命令或数据到达间接高压寄存器内需要有限的10 μs（最大）延时。MCU内核将命令写入HVCON到间接寄存器数据被回读到HVDAT寄存器内也需要有限的10 μs延时。MCU可轮询忙碌位（如在MCU读取时HVCON位0）以确认读/写命令何时结束。

下列高压电路功能通过此接口监控。图38显示高压接口和下列相关电路的顶层架构：

- 精密振荡器
- 唤醒（WU）引脚功能
- 电源监控器（PSM）
- 低压标志（LVF）
- LIN工作模式
- STI诊断
- 高压诊断
- 高压衰减器缓冲电路
- 高压（HV）温度监控器

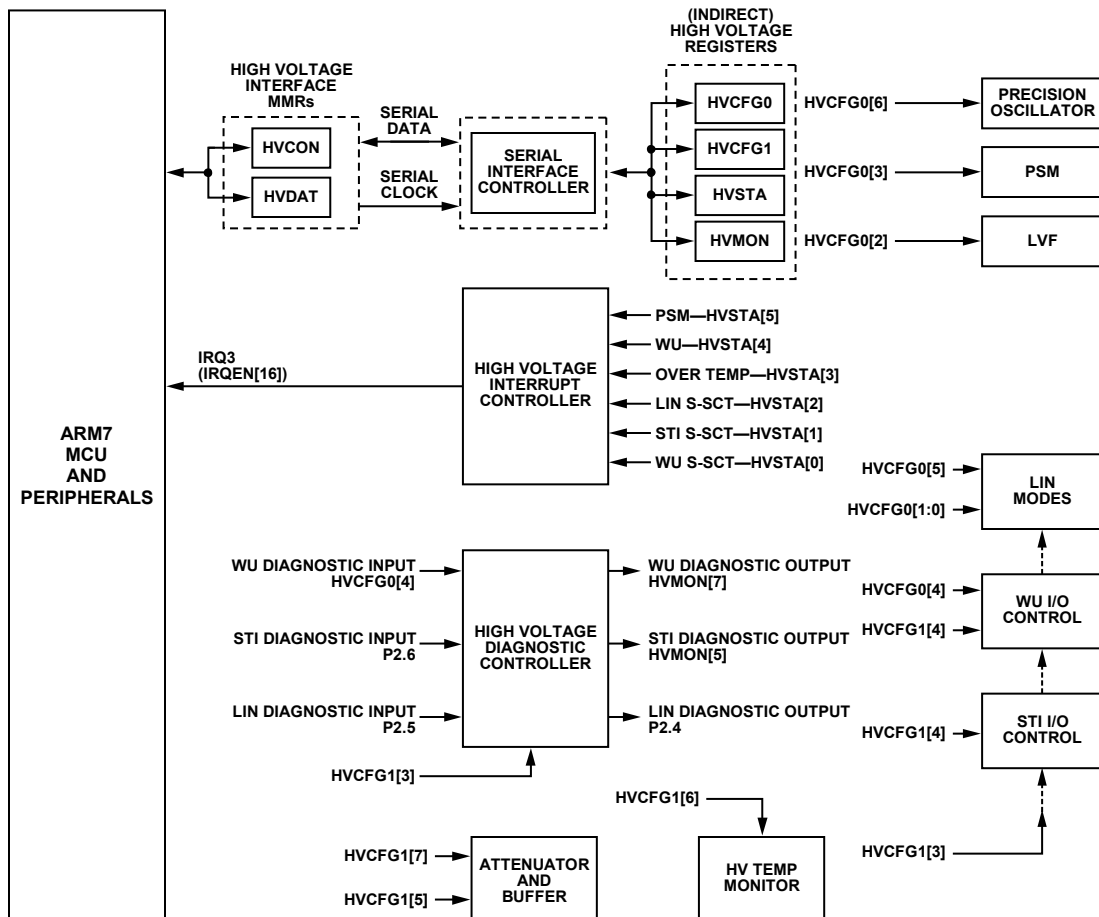


图38. 高压接口顶层框图

高压接口控制接口

名称: HVCON

地址: 0xFFFF0804

默认值: 由内核更新

访问类型: 读写

功能: 该8位寄存器用作高压控制接口的命令字节解释器。将高压电路相关的一组四个间接寄存器的写入字节解释为读或写命令。HVDAT寄存器用来存放间接寄存器的写入或回读数据。

表71. HVCON MMR写入位分配

位	描述
7至0	命令字节。含义为 0x00 = 将高压寄存器HVCFG0回读到HVDAT。 0x01 = 将高压寄存器HVCFG1回读到HVDAT。 0x02 = 将高压状态寄存器HVSTA回读到HVDAT。 0x03 = 将高压状态寄存器HVMON回读到HVDAT。 0x08 = 将HVDAT内的值写入高压寄存器HVCFG0。 0x09 = 将HVDAT内的值写入高压寄存器HVCFG1。

表72. HVCON MMR读取位分配

位	描述
7至3	保留。
2	发送到高压芯片命令状态。 1 = 命令成功完成。 0 = 命令失败。
1	高压芯片读取命令状态。 1 = 命令成功完成。 0 = 命令失败。
0	忙碌位（只读）。当用户代码读取此寄存器时，位0应视为高压接口的忙碌位。可用来决定读取请求是否已完成。除非忙碌=0，否则此表中所述高压（读/写）命令不能写入HVCON。 忙碌=1，高压接口忙碌且未完成前一个HVCON写入命令。位1、2无效。 忙碌=0，高压接口空闲且已完成HVCON写入命令。位1、2有效。

ADuC7036

高压数据寄存器

名称: HV DAT

地址: 0xFFFF080C

默认值: 由内核更新

访问类型: 读写

功能: 该12位寄存器保持下列高压接口寄存器的间接读写数据。

表73. HV DAT MMR位分配

位	描述
11至8	高压数据HV DAT[7:0]相关的命令。这些位为只读且应写入0 0x00 = 将高压寄存器HV CFG0回读到HV DAT。 0x01 = 将高压寄存器HV CFG1回读到HV DAT。 0x02 = 将高压状态寄存器HV STA回读到HV DAT。 0x03 = 将高压状态寄存器HV MON回读到HV DAT。 0x08 = 将HV DAT内的值写入高压寄存器HV CFG0。 0x09 = 将HV DAT内的值写入高压寄存器HV CFG1。
7至0	欲读/写的高压数据

高压配置0 寄存器

名称: HVCFG0

地址: 通过HVCON高压接口间接寻址

默认值: 0x00

访问类型: 读写

功能: 该8位寄存器控制ADuC7036上的高压电路功能。由于不是MMR, 不会出现在MMR存储器映射内。通过HVCON寄存器接口访问。欲写入该寄存器的数据通过HVDAT MMR来装载, 从该寄存器回读数据也是通过HVDAT MMR。

表74. HVCFG0位分配

位	描述
7	唤醒/STI热关断禁用。 置1, 在发生过热事件时禁用唤醒/STI驱动器的自动关断。 清0, 在发生过热事件时使能唤醒/STI驱动器的自动关断。
6	精密振荡器使能位。 置1, 使能精密131 kHz振荡器。振荡器启动时间一般持续70 μs (包括10 μs的高压接口延时)。 清0, 关断精密131 kHz振荡器。
5	位串行设备 (BSD) 模式使能位。 置1, 禁用内部 (LIN) 上拉电阻并将LIN/BSD引脚配置用于BSD操作。 清0, 使能LIN/BSD引脚上的内部 (LIN) 上拉电阻。
4	唤醒 (WU) 断言位。 置1, 置位外部WU引脚为高电平。 清0, 通过内部10 kΩ下拉电阻将外部WU引脚拉低。
3	电源监控器 (PSM) 使能位。 置1, 使能电源 (VDD引脚上的电压) 监控器。如果IRQ3 (IRQEN[16]) 使能, 一旦VDD引脚上的电压下降到6 V以下, PSM就会产生中断。 清0, 禁用电源 (VDD引脚上的电压) 监控器。
2	低压标志 (LVF) 使能位。 置1, 使能LVF功能。上电后可通过HVMON[3]查询低压标志以决定REG_DVDD电压是否先前下降到2.1 V以下。 清0, 禁用LVF功能。
1至0	LIN工作模式。使能/禁用LIN驱动器。 00 = LIN禁用。 01 = 保留 (LIN V2.0不兼容)。 10 = LIN使能。 11 = 保留, 未使用。

ADuC7036

高压配置1 寄存器

名称: HVCFG1

地址: 通过HVCON高压接口间接寻址

默认值: 0x00

访问类型: 读写

功能: 该8位寄存器控制ADuC7036上的高压电路功能。由于不是MMR, 不会出现在MMR存储器映射内。通过HVCON寄存器接口访问。欲写入该寄存器的数据通过HVDAT MMR来装载, 从该寄存器回读数据也是通过HVDAT MMR。

表75. HVCFG1位分配

位	描述
7	电压衰减器诊断使能位。 置1, 接通1.29 μ A电流源, 将170 mV差分电压加到电压通道测量上。 清0, 禁用电压衰减器诊断功能。
6	高压温度监控器。它是一个靠近高压电路的未校准片内温度监控器, 和片内精密温度传感器(通过ADCICON[7:6]控制)完全分开, 允许用户代码监控ADuC7036芯片最热部分附近的温度变化。在25°C时的典型输出电压为600 mV且负温度系数一般为-2.1 mV/°C。 置1, 使能片内高压温度监控器。此时它的电压输出直接送到电压通道ADC。 清0, 禁用片内高压温度监控器。
5	电压通道短路使能位。 置1, 使电压通道ADC内部短路(在衰减器处, 在ADC输入缓冲前)并允许噪声的自诊断测试。 清0, 禁止电压通道内部短路。
4	WU和STI回读使能位。 置1, 使能外部WU和STI引脚输入功能。此时WU和STI引脚上的上升或下降沿跳变会产生一个高压中断。置位后WU和STI引脚状态可通过HVMON寄存器(HVMON[7]和HVMON[5])来监控。 清0, 禁用外部WU和STI引脚输入功能。
3	高压I/O驱动器使能位。 置1, 再使能高压I/O引脚(LIN/BSD、STI和WU), 这些引脚由于短路电流事件而已被禁用(该事件对于LIN/BSD必须持续20 μ s以上而对于WU引脚必须持续400 μ s以上)。一旦WU和STI引脚由于热关断事件而被禁用, 也必须将此位置1以将它们再使能。注意, 此位必须置1以清除短路事件造成的任何未决中断(即使事件已过去)并再使能高压I/O引脚。 自动清0。
2	使能/禁用短路保护(LIN/BSD和STI)。 置1, 使能LIN引脚被动短路保护。此时LIN/BSD引脚上的短路事件会产生高压中断IRQ3(前提是IRQEN[16]使能), 并置位HVSTA内的相应状态位, 但不会禁用短路引脚。 清0, 使能LIN/BSD引脚主动短路保护。此时在短路事件期间LIN/BSD引脚产生高压中断(IRQ3), 置位HVSTA[16], 并自动禁用短路引脚。禁用后, 引脚只能通过写入HVCFG1[3]来再使能。
1	WU引脚超时(单稳态触发器)计数器使能/禁用。 置1, 禁用WU I/O超时计数器。 清0, 使能超时计数器, 在用户代码通过HVCFG0[4]置位WU引脚1.3秒后自动拉高WU引脚。
0	WU开路诊断使能。 置1, 使能VDD引脚内部WU I/O诊断上拉电阻, 从而允许检测WU引脚开路状况。 清0, 禁用内部WU I/O诊断上拉电阻。

高压监控器寄存器

名称：HVMON

地址：通过HVCON高压接口间接寻址

默认值：0x00

访问类型：只读

功能：该8位只读寄存器反映使能ADuC7036内高压相关电路及功能的当前状态。由于不是MMR，不会出现在MMR存储器映射内。通过HVCON寄存器接口访问并通过HVDAT回读数据。

表76. HVMON位分配

位	描述
7	WU引脚诊断回读。通过HVCFG1[4]使能后，这个只读位反映外部WU引脚状态。
6	过热。 0 = 未发生热关断事件。 1 = 已发生热关断事件。
5	STI引脚诊断回读。通过HVCFG1[4]使能后，这个只读位仅反映外部STI引脚状态。
4	缓冲使能。 0 = 电压通道ADC输入缓冲禁用。 1 = 电压通道ADC输入缓冲使能。
3	低压标志状态位。只在通过HVCFG0[2]使能时有效。 0（上电时） = REG_DVDD已下降至2.1 V以下。此时，RAM内容视为已被破坏。 1（上电时） = REG_DVDD未下降至2.1 V以下。此时RAM内容视为有效。只能通过再使能HVCFG0[2]内的低压标志来清0。
2	LIN/BSD短路状态标志。 0 = LIN/BSD驱动器工作正常。 1 = LIN/BSD驱动器已发生过短路状况并通过写入HVCFG1[3]自动清0。
1	STI短路状态标志。 0 = STI驱动器工作正常。 1 = STI驱动器已发生过短路状况并通过写入HVCFG1[3]自动清0。
0	唤醒短路状态标志。 0 = 唤醒驱动器工作正常。 1 = 唤醒驱动器已发生过短路状况。

ADuC7036

高压状态寄存器

名称: HVSTA

地址: 通过HVCON高压接口间接寻址

默认值: 0x00

访问类型: 只读, 应只在高压中断时读取。

功能: 该8位只读寄存器反映HVMON寄存器内所有相应位的状态变化。由于不是MMR, 不会出现在MMR存储器映射内。通过HVCON寄存器接口访问并通过HVDAT回读数据。高压中断控制器响应高压中断事件, 将高压状态寄存器 (HVSTA) 的当前值同时自动载入HVDAT寄存器内。

表77. HVSTA位分配

位	描述
7至6	保留。这些位不能使用, 保留供未来使用。
5	PSM状态位。只在通过HVCFG0[3]使能时有效。不锁存且需要使能IRQ来加以检测。 0 = VDD引脚上的电压保持在6 V以上。 1 = VDD引脚上的电压降到6 V以下。
4	WU请求状态位。只在通过HVCFG1[4]使能时有效。 0 = WU引脚未曾产生高压中断。 1 = WU引脚上的上升或下降沿跳变产生一个高压中断 (当通过HVCFG1[4]使能时)。
3	过热。该位始终使能。 0 = 未发生热关断事件。 1 = 已发生热关断事件。一旦发生热关断, 所有高压 (LIN/BSD、WU和STI) 引脚驱动器都自动禁用。
2	LIN/BSD短路状态标志。 0 = LIN/BSD正常工作。通过读取HVSTA寄存器自动清0。 1 = 检测到LIN/BSD短路。此时LIN驱动器自动禁用。
1	STI短路状态标志。 0 = STI驱动器正常工作。通过读取HVSTA寄存器自动清0。 1 = STI驱动器已发生过短路状况。
0	唤醒短路状态标志。 0 = 唤醒正常工作。 1 = 检测到唤醒短路。

唤醒 (WU) 引脚

唤醒 (WU) 引脚是一个通过HVCON和HVDAT控制的高压GPIO。

唤醒 (WU) 引脚电路描述

默认情况下，WU引脚被配置成具有内部10 k Ω 下拉电阻和高端FET驱动器的输出端。在默认工作模式下，通过强制外部系统WU总线为高电平，WU引脚指定用于产生高电平有效系统唤醒请求。用户代码可通过直接写入HVCFG0[4]来置位WU输出。

注意，在10 μ s延时后输出才响应；此延时是HVCON或HVDAT MMR和高压接口的串行通信中固有的（参见“高压外设控制接口”部分）。

内部FET驱动器能够供应相当大的电流；因此如果它长时间置位，就可能出现严重的片内自发热。为此包括了一个单稳态触发器（即1.3秒超时定时器）。

默认情况下，单稳态触发器在1.3秒后使能和禁用唤醒驱动器，可通过HVCFG1[1]来禁用单稳态触发器。如果唤醒单稳态触发器被禁用，唤醒驱动器在1.3秒后将被禁用。

WU引脚还带一个短路检测功能。当唤醒引脚供应100 mA以上电流一般持续400 μ s时，就会产生一个高压中断，并将HVMON[0]置1。

热关断事件会禁用WU驱动器。此后必须利用HVCFG1[3]将其手动再使能。

HVCFG1[4]写入1可将WU引脚配置成I/O模式。在此模式下，上升或下降沿会立即产生一个高压中断。HVMON[7]直接反映外部WU引脚的状态并指示外部唤醒总线（包括 $R_{LOAD} = 1\text{ k}\Omega$ ， $C_{LOAD} = 91\text{ nF}$ 且 $R_{LIMIT} = 39\ \Omega$ ）高于还是低于3 V典型电压。

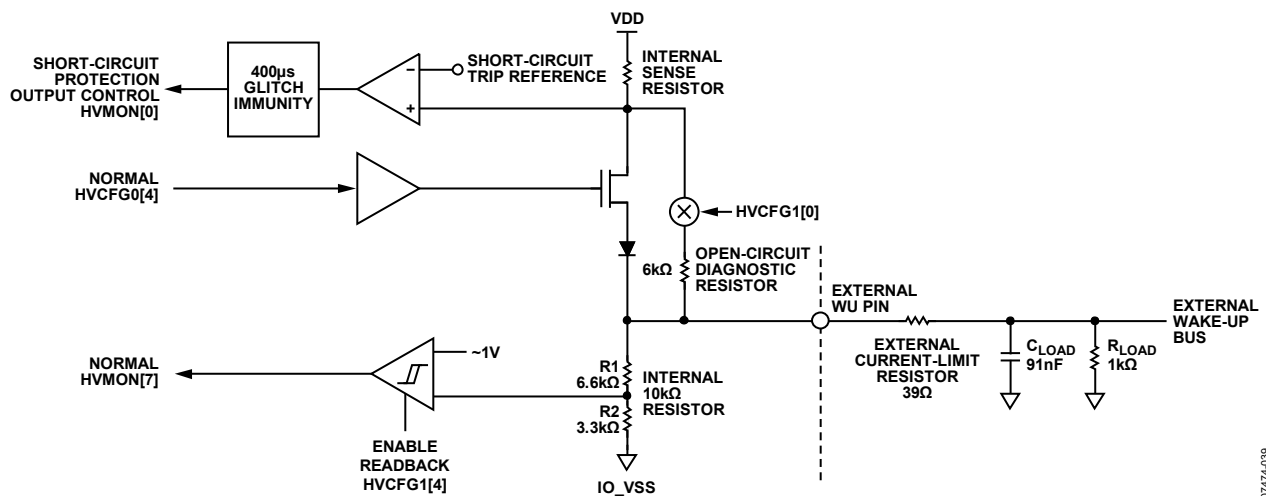


图 39. WU 电路框图

07474-039

高压外设控制接口中断处理

高压电路集成了一个中断控制器。它通过IRQEN[16]使能，然后6个高压源其中的一个可置位高压中断（IRQ3）信号并中断MCU内核。

尽管正常情况下MCU通过跳转到IRQ或FIQ矢量地址来响应此中断事件，但高压中断控制器仍同时自动地将高压状态寄存器（HVSTA）的当前值载入HVDAT寄存器内。此时HVCON[0]内的忙碌位被置1，指示传送正在进行中并在10 μs后清0以指示可从HVDAT获取HVSTA内容。

因此中断处理程序可一直轮询HVCON内的忙碌位，直到其置位。在忙碌位清0后，必须检查HVCON[1]以确保正确读取数据。接下来，可读取HVDAT寄存器。此时它保持着HVSTA寄存器的值。然后可查询状态标志以决定高压中断的确切来源，并采取适当操作。

低压标志（LVF）

ADuC7036有一个低压标志（LVF），使用户可以监控REG_DVDD。当通过HVCFG0[2]使能时，可通过HVMON[3]来监控低压标志。如果REG_DVDD降至2.1 V以下，则HVMON[3]清0且RAM内容已被破坏。低压标志在使能后只能通过REG_DVDD降至2.1 V以下或通过利用HVCFG0[2]禁用LVF功能来复位。

高压诊断

如表78所示，可在WU、LIN和STI引脚上诊断故障状况。

表78.高压诊断

高压引脚	故障状况	方法	结果
LIN或STI	LIN或STI和BAT短路	驱动LIN或STI至低电平	如果持续吸取100 mA以上电流，20μs后就会产生LIN或STI短路中断。
	LIN或STI和GND短路	驱动LIN或STI至高电平	LIN或STI回读操作回读低电平。
WU	唤醒和VBAT短路	驱动WU至低电平	在HVMON[7]内回读高电平。
	唤醒和GND短路	驱动WU至高电平	如果产生100 mA以上电流，400 μs后就会产生WU短路中断。
	开路	使能OC诊断电阻，同时禁用WU。	如果接有负载，则HVMON[7]清0；如果WU开路，则置1。

UART串行接口

ADuC7036有一个16,450兼容UART。它是一个全双工、通用异步接收器/发送器。UART的作用是对从外设接收的数据字符进行串并转换并对从ARM7TDMI接收的数据字符进行并串转换。该UART有一个小数分频器以促进高精度波特率生成，并还支持一种网络可寻址模式。UART功能通过ADuC7036的GPIO_5/IRQ1/RxD和GPIO_6/ TxD引脚提供。

串行通信采用异步协议，支持各种字长、停止位以及配置寄存器可选的奇偶校验生成形式。

波特率生成

ADuC7036有两种UART波特率生成模式：正常450 UART波特率生成和ADuC7036小数分频器波特率生成。

正常450 UART波特率生成

该波特率是利用COMDIV0和COMDIV1 MMR内值（均为16位，DL）划分内核时钟的分频形式。标准波特率生成器公式：

$$\text{波特率} = \frac{20.48\text{MHz}}{2^{CD} \times 16 \times 2 \times DL} \quad (1)$$

表79列举了几种常见波特率值。

表79. 利用标准波特率发生器的波特率

波特率 (bps)	CD	DL	实际波特率	百分比误差
9600	0	0x43	9552	0.50%
19,200	0	0x21	19,394	1.01%
115,200	0	0x6	106,667	7.41%
9600	3	0x8	10,000	4.17%
19,200	3	0x4	20,000	4.17%
115,200	3	0x1	80,000	30.56%

小数分频器

小数分频器配合正常波特率发生器可以生成精确高速的波特率。

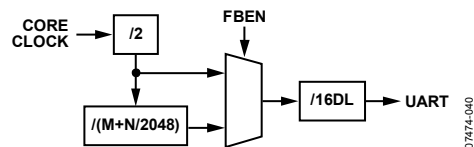


图40. 小数分频器波特率生成

如下可利用小数分频器计算波特率：

$$\text{波特率} = \frac{20.48\text{MHz}}{2^{CD} \times 16 \times DL \times 2 \times \left(M + \frac{N}{2048} \right)} \quad (2)$$

$$M + \frac{N}{2048} = \frac{20.48\text{MHz}}{\text{Baud Rate} \times 2^{CD} \times 16 \times DL \times 2}$$

其中：

CD是时钟分频器。

DL是分频锁存器。

M是除数的整数部分；小数分频器将输入除以非整数M.N。

N是除数的小数部分；小数分频器将输入除以非整数M.N。

表80列举了几种常见波特率值。

表80. 利用小数波特率发生器的波特率

波特率 (bps)	CD	DL	M	N	实际波特率	百分比误差
9600	0	0x42	1	21	9598.55	0.015%
19,200	0	0x21	1	21	19,197.09	0.015%
115,200	0	0x5	1	228	115,177.51	0.0195%

UART寄存器定义

此UART接口包括下列9个寄存器：

- COMTX：8位发送寄存器
- COMRX：8位接收寄存器
- COMDIV0：分频锁存器（低字节）
- COMDIV1：分频锁存器（高字节）
- COMCON0：线路控制寄存器
- COMCON1：线路控制寄存器
- COMSTA0：线路状态寄存器
- COMIEN0：中断使能寄存器
- COMIID0：中断识别寄存器
- COMDIV2：16位小数波特率分频寄存器

COMTX、COMRX和COMDIV0共用相同的地址。要访问COMTX、COMIEN0和COMRX，需将COMCON0寄存器内的位7清0；要访问COMDIV0和COMDIV1，则将其置1。

UART Tx 寄存器

名称：COMTX

地址：0xFFFF0700

访问类型：只写

功能：写入该8位寄存器后允许利用UART发送数据。

UART Rx 寄存器

名称：COMRX

地址：0xFFFF0700

默认值：0x00

访问类型：只读

功能：该8位寄存器接收利用UART发送的数据以供读取。

UART分频锁存器寄存器0

名称：COMDIV0

地址：0xFFFF0700

默认值：0x00

访问类型：读写

功能：该8位寄存器保持控制UART波特率的分频锁存器的最低有效字节。

UART分频锁存器寄存器1

名称：COMDIV1

地址：0xFFFF0704

默认值：0x00

访问类型：读写

功能：该8位寄存器保持控制UART波特率的分频锁存器的最高有效字节。

UART控制寄存器 0

名称: COMCON0

地址: 0xFFFF070C

默认值: 0x00

访问类型: 读写

功能: 该8位寄存器配合COMCON1控制UART操作。

表81. COMCON0 MMR位分配

位	名称	描述
7	DLAB	分频锁存器访问。 用户置1, 使能COMDIV0和COMDIV1访问功能。 用户清0, 禁用COMDIV0和COMDIV1访问功能并使能COMRX、COMTX和COMIEN0访问功能。
6	BRK	设置断开。 用户置1, 强制TxD为0。 用户清0, 采用正常模式工作。
5	SP	强制奇偶校验。用户置1, 强制奇偶校验为定义值。 如果EPS = 1且PEN = 1, 则置1。 如果EPS = 0且PEN = 1, 则置0。
4	EPS	偶校验选择位。 置1选择偶校验。 清0选择奇校验。
3	PEN	奇偶校验使能位。 用户置1发送并检查奇偶校验位。 用户清0, 无奇偶校验发送或检查。
2	停止	停止位。 用户置1, 在字长为5位时发送1.5个停止位, 或在字长为6、7或8位时发送2个停止位。不论所需停止位数目多少, 接收器只检查第一个停止位。 用户清0, 在发送数据内产生一个停止位。
1至0	WLS	字长选择。 00 = 5位。 01 = 6位。 10 = 7位。 11 = 8位。

UART控制寄存器 1

名称: COMCON1

地址: 0xFFFF0710

默认值: 0x00

访问类型: 读写

功能: 该8位寄存器配合COMCON0控制UART操作。

表82. COMCON1 MMR位分配

位	名称	描述
7至6	SMS	UART输入多路复用器。 00 = RxD由LIN输入驱动; 适用于利用LIN引脚进行LIN通信。 01 = 保留。 10 = RxD由GP5驱动; 适用于利用GPIO_5/IRQ1/RxD引脚 (RxD) 进行串行通信。 11 = 保留。
5		保留。未使用。
4	LOOPBACK	回送。用户置1, 使能回送模式。此时强制TxD为高电平。
3至0		保留。未使用。

ADuC7036

UART状态寄存器0

名称: COMSTA0

地址: 0xFFFF0714

默认值: 0x60

访问类型: 只读

功能: 该8位只读寄存器反映UART的当前状态。

表83. COMSTA0 MMR位分配

位	名称	描述
7		保留。
6	TEMT	COMTX和移位寄存器空状态位。 COMTX和移位寄存器为空时自动置1。该位指示数据已发送; 即移位寄存器不再保持数据。 通过写入COMTX自动清0。
5	THRE	COMTX空状态位。 COMTX为空时自动置1。THRE位置1便可写入COMTX, 但前面数据可能还未发送并仍可能存在于移位寄存器内。 通过写入COMTX自动清0。
4	BI	断开指示符。 当SIN保持低电平超过最大字长时置1。 自动清0。
3	FE	帧错误。 当停止位无效时置1。 自动清0。
2	PE	奇偶校验错误。 当发生奇偶校验错误时置1。 自动清0。
1	OE	溢出错误。 当前读取数据被覆盖, 自动置1。 自动清0。
0	DR	数据就绪。 COMRX满时自动置1。 COMRX读取后清0。

UART中断使能寄存器0

名称: COMIEN0

地址: 0xFFFF0704

默认值: 0x00

访问类型: 读写

功能: 该8位寄存器使能和禁用个别UART中断源。

表84. COMIEN0 MMR位分配

位	名称	描述
7至4		保留。未使用。
3	EDSSI	保留。该位应写入0。
2	ELSI	RxD状态中断使能位。 用户置1, 在COMSTA0[3:1]寄存器任一位置1时使能中断产生。 由用户清0。
1	ETBEI	使能发送缓冲空中断。 用户置1, 当发送期间缓冲为空时, 即当COMSTA0[5]置1时使能中断。 由用户清0。
0	ERBFI	使能接收缓冲满中断。 用户置1, 当接收期间缓冲满时使能中断。 由用户清0。

UART中断识别寄存器0

名称: COMIID0

地址: 0xFFFF0708

默认值: 0x01

访问类型: 只读

功能: 该8位寄存器反映UART中断源。

表85. COMIID0 MMR位分配

状态位[2:1]	位0 NINT	优先级	定义	清除操作
00	1		无中断	
11	0	1	接收线路状态中断	读取COMSTA0
10	0	2	接收缓冲满中断	读取COMRX
01	0	3	发送缓冲空中断	将数据写入COMTX或读取COMIID0
00	0	4	保留	保留

ADuC7036

UART小数分频器寄存器

名称: COMDIV2

地址: 0xFFFF072C

默认值: 0x0000

访问类型: 读写

功能: 该16位寄存器控制ADuC7036小数分频器操作。

表86. COMDIV2 MMR位分配

位	名称	描述
15	FBEN	小数波特率发生器使能位。 用户置1, 使能小数波特率发生器。 用户清0, 利用标准450 UART波特率发生器产生波特率。
14至13		保留。
12至11	FBM[1:0]	小数分频器M。如果FBM = 0, 则M = 4。有关使用M小数分频器波特率计算见公式2, 有关常见波特率值见表80。
10至0	FBN[10:0]	小数分频器N。有关使用小数分频器波特率计算见公式2, 有关常见波特率值见表80。

串行外设接口

ADuC7036片内集成了一个完整硬件串行外设接口（SPI）。SPI是一个工业标准同步串行接口，允许8位数据的同步发送和同时接收，即全双工。

该SPI接口只由内核时钟分频器位（POWCON[2:0] = 0000或001）控制。

该SPI端口可配置为主机或从机操作由四个多路复用的GPIO引脚。这四个SPI引脚分别为MISO、MOSI、SCLK和 \overline{SS} 。这些信号对应的引脚见表87。

表87. SPI输出引脚

引脚	SPI 引脚功能	描述
GP0（GPIO模式1）	\overline{SS}	片选。
GP1（GPIO模式1）	SCLK	串行时钟
GP2（GPIO模式1）	MISO	主机输入从机输出
GP3（GPIO模式1）	MOSI	主机输出从机输入

MISO引脚

MISO（主机输入从机输出）引脚在主模式下配置成输入线路而在从模式下配置成输出线路。主机上的MISO线路（数据输入）应连接到从机内的MISO线路（数据输出）。传送的数据是以字节（8位）为单位的串行数据，MSB优先。

MOSI引脚

MOSI（主机输出从机输入）引脚在主模式下配置成输出线路而在从模式下配置成输入线路。主机上的MOSI线路（数据输出）应连接到从机内的MOSI线路。传送的数据是以字节（8位）为单位的串行数据，MSB优先。

SCLK引脚

SCLK（主机串行时钟）引脚用来同步在MOSI SCLK周期内发送接收的数据。所以每8个SCLK周期发送/接收一个字节。SCLK引脚在主模式下配置成输出端而在从模式下配置成输入端。

在主模式下，时钟的极性和相位由SPICON寄存器控制，比特率利用SPI波特率计算在SPIDIV寄存器中定义如下：

$$f_{CLOCKSERIA} = \frac{20.48MHz}{2 \times (1 + SPIDIV)} \quad (3)$$

SPI时钟的最大速度取决于时钟分频位并归纳于表88内。

表88. 主模式下SPI速度和时钟分频位对照表

CD位设置	SPIDIV	最大SCLK（MHz）
0	0x05	1.667
1	0x0B	0.833

在从模式下，必须在SPICON寄存器内配置预期输入时钟的相位和极性。当CD = 0时，从机从外部主机接收数据速率高达5.12 Mb。最大速度计算公式如下：

$$f_{SERIALCLOCK} = \frac{f_{HCLK}}{4} \quad (4)$$

同时主模式和从模式下，数据都在SCL信号的一个沿发送并另一个沿采样。所以主机和从机的极性和相位配置必须一致。

\overline{SS} 引脚

在SPI从模式时，发送是否开始是通过 \overline{SS} （低电平有效输入信号）引脚声明。SPI端口发送和接收8位数据，直到 \overline{SS} 声明其结束时停止。在从模式下， \overline{SS} 总是为输入。

SPI寄存器定义

下列MMR寄存器用来控制SPI接口：

- SPICON: 16位控制寄存器
- SPISTA: 8位只读状态寄存器
- SPIDIV: 8位串行时钟分频器寄存器
- SPITX: 8位只写发送寄存器
- SPIRX: 8位只读接收寄存器

ADuC7036

SPI控制寄存器

名称: SPICON

地址: 0xFFFF0A10

默认值: 0x0000

访问类型: 读写

功能: 该16位MMR配置串行外设接口。

表89. SPICON MMR位分配

位	描述
15至13	保留。
12	连续发送使能。 用户置1, 连续发送使能。在主模式下, 会连续发送数据直到SPITX寄存器内无有效数据时停止。在每一次8位连续发送期间SS保持有效直到SPITX空。 用户清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	回送使能。 用户置1, MISO连接到MOSI, 用于测试软件。 用户清0, 返回正常模式。
10	从机输出使能。 用户置1, 使能从机输出。 用户清0, 禁用从机输出。
9	从机选择输入使能。 在主模式下, 用户置1使能主机输出。 用户清0, 禁用主机输出。
8	SPIRX上溢覆盖使能。 用户置1, 新接收到的串行数据覆盖SPIRX寄存器中的有效数据。 用户清0, 新接收到的串行数据会被丢弃。
7	SPITX下溢模式。 用户置1, 发送前一个数据。 用户清0, 发送0。
6	发送和中断模式(主模式)。 用户置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。 用户清0, 读取SPITX寄存器时开始发送。只有当SPIRX满时产生中断。
5	LSB先发使能位。 用户置1, LSB先发。 用户清0, MSB先发。
4	保留。
3	串行时钟极性模式位。 用户置1, 串行时钟高电平空闲。 用户清0, 串行时钟低电平空闲。
2	串行时钟相位模式位。 用户置1, 串行时钟脉冲出现在每一次串行位传输的起始位置。 用户清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。 用户置1, 主模式使能。 用户清0, 从模式使能。
0	SPI使能位。 用户置1, SPI使能。 用户清0, SPI禁用。

SPI状态寄存器

名称: SPISTA

地址: 0xFFFF0A00

默认值: 0x00

访问类型: 只读

功能: 该8位MMR指示串行外设接口的当前状态。

表90. SPISTA MMR位分配

位	描述
7至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寄存器后清0。
4	SPIRX数据寄存器IRQ。 位3或位5置1时自动置1。 读SPIRX寄存器后清0。
3	SPIRX数据寄存器满状态位。 如果SPIRX寄存器中存在有效数据则自动置1。 读SPIRX寄存器后清0。
2	SPITX数据寄存器下溢状态位。 SPITX下溢时自动置1。 写入SPITX寄存器后清0。
1	SPITX数据寄存器IRQ。 位0清0或位2置1时自动置1。 写入SPITX寄存器或发送完成SPI停止工作后清0。
0	SPITX数据寄存器空状态位。 向SPITX寄存器写入数据后置1。在数据发送过程中这个位始终置1。 当SPITX空时清0。

SPI接收寄存器

名称: SPIRX

地址: 0xFFFF0A04

默认值: 0x00

访问类型: 只读

功能: 该8位MMR存放串行外设接口接收的数据。

SPI发送寄存器

名称: SPITX

地址: 0xFFFF0A08

访问类型: 只写

功能: 写入该8位MMR采用串行外设接口发送数据。

SPI分频器寄存器

名称: SPIDIV

地址: 0xFFFF0A0C

默认值: 0x1B

访问类型: 读写

功能: 该8位MMR指示串行外设接口的工作频率。有关波特率计算的详情, 参见公式3。

串行测试接口

ADuC7036集成了单引脚串行测试接口（STI）端口，利用它终端用户可对成品进行评估或诊断。

STI端口以12位分组形式发送1到6个字节数据。如图41所示，每个发送分组包括一个起始位、发送字节（8位）、一个偶校验位和两个停止位。STI数据在STI引脚上发送，而波特率由定时器4的上溢速率决定。

STI端口通过6个MMR来配置和控制。

- STIKEY0: 串行测试接口密钥0
- STIKEY1: 串行测试接口密钥1
- STIDAT0: 数据0（16位）保持两个字节
- STIDAT1: 数据1（16位）保持两个字节
- STIDAT2: 数据2（16位）保持两个字节
- STICON: 控制串行测试接口

串行测试接口密钥0 寄存器

名称: STIKEY0

地址: 0xFFFF0880

访问类型: 只写

功能: 该STIKEY0 MMR配合STIKEY1 MMR保护STICON MMR。任何尝试写入STICON前必须先先在STIKEY0内写入0x0007。写入STICON后必须立即在STIKEY1内写入0x00B9以确保STICON写入序列成功完成。如果STIKEY0未写入、失序写入或不正确写入，就会忽略STICON MMR的任何后续写入。

串行测试接口密钥1 寄存器

名称: STIKEY1

地址: 0xFFFF0888

访问类型: 只写

功能: STIKEY1 MMR配合STIKEY0 MMR保护STICON MMR。在任何尝试写入STICON后必须立即在STIKEY1内写入0x00B9。写入STICON前必须在STIKEY0内写入

0x0007以确保STICON写入序列成功完成。如果STIKEY1未写入、失序写入或不正确写入，就会忽略STICON MMR的任何后续写入。

串行测试接口数据0 寄存器

名称: STIDAT0

地址: 0xFFFF088C

默认值: 0x0000

访问类型: 读写

功能: STIDAT0 MMR是一个16位寄存器，用来保持STI端口使能后STI引脚准备发送的第一个和第二个数据字节。发送的第一个字节占据位[0:7]，而第二字节占据位[8:15]。

串行测试接口数据1 寄存器

名称: STIDAT1

地址: 0xFFFF0890

默认值: 0x0000

访问类型: 读写

功能: STIDAT1 MMR是一个16位寄存器，用来保持STI端口使能后STI引脚准备发送的第三个和第四个数据字节。发送的第三个字节占据位[0:7]，而第四字节占据位[8:15]。

串行测试接口数据2 寄存器

名称: STIDAT2

地址: 0xFFFF0894

默认值: 0x0000

访问类型: 读写

功能: STIDAT2 MMR是一个16位寄存器，用来保持STI端口使能后STI引脚准备发送的第五个和第六个数据字节。发送的第五个字节占据位[0:7]，而第六字节占据位[8:15]。

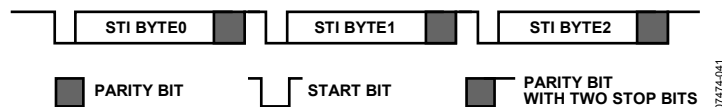


图41. 串行ADC测试接口实例（3字节发送）

串行测试接口控制寄存器

名称: STICON

地址: 0xFFFF0884

默认值: 0x0000

访问类型: 读/写, 通过两个密钥寄存器 (STIKEY0和STIKEY1) 写入保护。只有遵照下列三步写入序列才能正确完成STICON写入操作:

1. 在STIKEY0 MMR内写入0x0007。
2. 写入STICON。
3. 将0x00B9写入STIKEY1, 序列结束。

功能: STI控制MMR是一个16位寄存器, 配置串行测试接口的工作模式。

注意, 必须在GP2CON内将GPIO_13配置用于STI操作以进行STI通信。

表91. STICON MMR位分配

位	描述
15至9	保留。这些位保留供未来使用, 应由用户代码写入0。
8至5	状态位, 只读。如果接口正在发送数据中, 这些位非0。
4至2	欲发送字节数。选择欲发送字节数。随后用户代码必须将欲发送字节写入STIDAT0、STIDAT1和STIDAT2 MMR内。 000 = 1个字节发送。 001 = 2个字节发送。 010 = 3个字节发送。 011 = 4个字节发送。 100 = 5个字节发送。 101 = 6个字节发送。
1	复位串行测试接口。 1 = 复位串行测试接口。后续读取STICON返回全0。 0 = 正常工作模式 (默认)。
0	串行测试接口使能。用户代码置1。 1 = 使能串行测试接口。 0 = 禁用串行测试接口。

ADuC7036

串行测试接口输出结构

串行测试接口是一种集成了低端驱动器、短路保护和诊断引脚回读功能的高压输出端口。输出驱动器电路配置如图42所示。

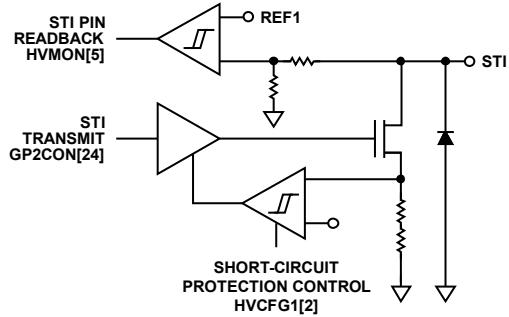


图 42. STI 输出结构

利用串行测试接口

只有进行下列配置，STI端口才能开始数据发送：

配置定时器4为波特率生成。

利用安全访问密钥 STIKEY0 和 STIKEY1 正确使能 STICON。

将欲发送的字节写入STIDAT0、STIDAT1和STIDAT2。

代码实例

下列代码段实例显示配置STI端口发送5个字节紧接着发送2个字节：

```
T4LD = 267; // Timer4 reload value
T4CON = 0xC0; // Enable T4, selecting core clock in periodic mode

STIKEY0 = 07; // STICON start write sequence
STICON = 0x11; // Enable and transmit five bytes
STIKEY1 = 0xb9; // STICON complete write

STIDAT0 = 0xAABB; // Five bytes for
STIDAT1 = 0xCCDD; // transmission
STIDAT2 = 0xFF;

while(STICON != 0x09) // Wait for transmission to complete
{}

STIKEY0 = 07; // STICON start write sequence
STICON = 0x05; // Enable and transmit two bytes
STIKEY1 = 0xb9; // STICON complete write

STIDAT0 = 0xEEFF; // Two bytes for transmission

while(STICON != 0x09) // Wait for transmission to complete
{}


```

在定时器4内载入适当值以在所需波特率下产生上溢。如果STI端口正用来发送ADC转换结果，则波特率必须在下一个ADC转换结果可用前足以输出每个ADC结果（16位）。

例如，当ADC进行1 kHz采样时，波特率必须足以输出下面36位：（3 × 8位（例如16位ADC结果和一个校验和字节）） + （3 × 1个起始位） + （3 × 1个奇偶校验位） + （3 × 2个停止位） = 36位。

所以串行测试接口的数据传输率必须超过36 kbps。最接近的标准波特率为38.4 kbps；所以定时器4载入MMR（T4LD）的重载值为0x0106（十进制267）。这个值基于预分频器1，并利用10.24 MHz内核时钟计算如下：

$$T4LD = \frac{\text{Core Clock Frequency}}{\text{Desired Baud Rate}} = \frac{10.24 \text{ Mhz}}{38.4 \text{ Kbps}} = 267$$

在写入定时器4加载值并利用T4CON MMR配置和使能定时器后，还必须配置STI端口。如前面部分所述，利用STIKEY0和STIKEY1 MMR按照一定序列写入STICON MMR，完成配置。

最后，将所需发送字节数写入STIDATx MMR，STI端口才能开始发送。开始发送后，STICON MMR内的值便会不同于初始写入值。用户代码可通过不断轮询STICON MMR直到恢复初始写入值以确保所有数据均发送完毕。要禁用串行接口，用户代码必须将STICON[0]写入0。

LIN（局域互连网络）接口

ADuC7036在ARM7 MCU内核和外部LIN总线之间具有若干高压物理接口。该LIN接口只能用作从机接口，支持1至20 kBaud并兼容LIN 2.0标准。从节点内置所需上拉电阻，从而省却了外部电路。如图43所示，LIN协议利用片内UART、一个IRQ、一个专用LIN定时器和高压收发器（也片内集成）来仿真。该LIN从断开定时器的低功耗振荡器接收时钟信号，并利用PLL的5 MHz输出用于同步字节定时。

LIN MMR描述

LIN硬件同步（LHS）功能由5个MMR控制。每个MMR的功能如下：

- LHSSTA：LHS状态寄存器。此MMR保持信息标志，指示接口的当前状态。
- LHSCON0：LHS控制寄存器0。此MMR控制LHS定时器的配置。
- LHSCON1：LHS起始和停止沿控制寄存器。此MMR决定LHS开始/停止计数的LIN同步字节边沿。
- LHSVAL0：LHS同步16位定时器。此MMR由LHSCON0控制。
- LHSVAL1：LHS断开定时器寄存器。

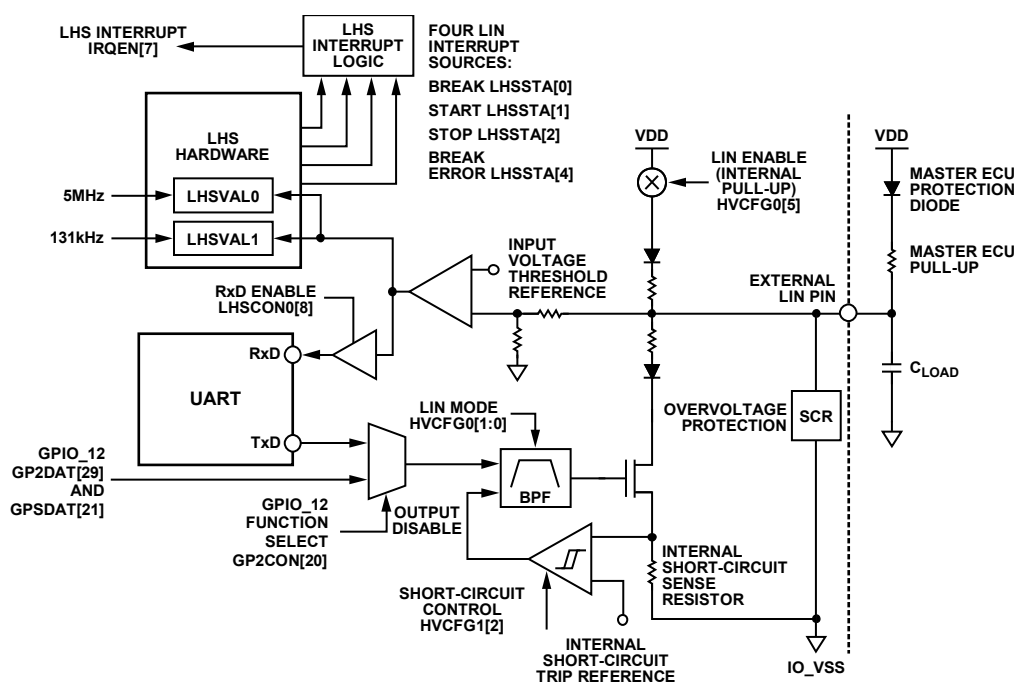


图43. LIN I/O 框图

07474-043

ADuC7036

LIN 硬件同步状态寄存器

名称: LHSSTA

地址: 0xFFFF0780

默认值: 0x00000000

访问类型: 只读

功能: 该LHS状态寄存器是一个32位寄存器, 反映LIN接口的当前工作状态。

表92. LHSSTA MMR位分配

位	描述
31至7	保留。这些只读位保留供未来使用。
6	上升沿检测 (限BSD模式)。 硬件置1, 指示已在BSD总线上检测到一个上升沿。 用户代码读取LHSSTA MMR后清0。
5	LHS复位完成标志。 硬件置1, 指示LHS复位命令已成功完成。 用户代码读取LHSSTA MMR后清0。
4	断开字段错误。 当12位断开定时器 (LHSVAL1) 寄存器上溢时硬件置1并产生LHS中断 (IRQEN[7]) 以指示LIN总线保持低电平时间过长, 从而指示可能发生的LIN总线错误。 用户代码读取LHSSTA MMR后清0。
3	LHS比较中断。 当LHSVAL0 (LIN同步位定时器) 值等于LHSCMP寄存器值时硬件置1。 用户代码读取LHSSTA MMR后清0。
2	停止条件中断。 当检测到停止条件时硬件置1。 用户代码读取LHSSTA MMR后清0。
1	起始条件中断。 当检测到起始条件时硬件置1。 用户代码读取LHSSTA MMR后清0。
0	断开定时器比较中断。 当检测到有效LIN断开条件时硬件置1。当LIN断开定时器值到达断开定时器比较值时产生LIN断开条件 (详情参见“LIN硬件同步断开定时器1寄存器”部分内的LHSVAL1)。 用户代码读取LHSSTA MMR后清0。

LIN 硬件同步控制寄存器 0

名称: LHSCON0

地址: 0xFFFF0784

默认值: 0x00000000

访问类型: 读写

功能: 该16位LHS控制寄存器配合LHSCON1寄存器, 用来配置LIN工作模式。

表93. LHSCON0 MMR位分配

位	描述	
15至13	保留。这些位保留供未来使用, 应由用户软件写入0。	
12	上升沿检测中断禁用。	
	模式	描述
	BSD模式	置1, 禁用上升沿检测中断。 清0, 使能断开上升沿检测中断。
	LIN模式	置1, 使能上升沿检测中断。 清0, 禁用断开上升沿检测中断。
11	断开定时器比较中断禁用。 置1, 禁用断开定时器比较中断。 清0, 使能断开定时器比较中断。	
10	断开定时器错误中断禁用。 置1, 禁用断开定时器错误中断。 清0, 使能断开定时器错误中断。	
9	LIN收发器独立测试模式。 用户代码置1, 使能外部GPIO_7和GPIO_8引脚以独立于UART来分别驱动LIN收发器TxD和RxD。但前提是用户代码必须先通过GP2CON寄存器内的GPIO_7功能选择位0和GPIO_8功能选择位4来配置GPIO_7和GPIO_8功能。 用户代码清0, LIN在正常模式下工作, 直接从片内UART驱动。	
8	UART/BSD R/ \overline{W} 选通位。	
	模式	描述
	UART模式	用户代码置1, 通过选通高电平禁用内部UART RxD (接收数据), 直到检测到断开字段及后续LIN同步字节, 从而确保在断开或同步字段周期期间UART不会接收任何伪串行数据, 否则UART必须在接收有效数据字段前完全清除这些伪串行数据。 用户代码清0, 在检测到断开字段及后续LIN同步字节后使能内部UART RxD (接收数据), 这样UART才可以接收后续LIN数据字段。
	BSD读取模式 ¹	用户代码置1, 在BSD总线上升沿时使能断开条件中断 (LHSSTA[0]) 产生。在产生中断后, 断开定时器 (LHSVAL1) 在下降沿时开始计数并在上升沿时停止计数, 使用户代码能够决定是否已接收到0、1或同步脉冲宽度。注意, 如果LIN断开定时器值(LHSVAL1读取值)等于断开定时器比较值(LHSVAL1写入值)且断开定时器上溢, 则断开定时器产生一个中断。此配置可在BSD读取模式下用来检测BSD总线的故障状况。
BSD写入模式 ¹	用户代码清0, 在BSD总线上升沿时禁用断开条件中断产生。LHS比较中断位 (LHSSTA[3]) 用来决定数据发送期间MCU释放BSD总线的时间。如果断开条件中断仍然使能, 一旦解除BSD总线声明, 就会产生一个不必要的中断。和BSD读取模式一样, 断开定时器在上升沿时停止计数; 所以断开定时器还可在此模式下用来允许用户代码确认发送数据位中的脉冲宽度。	
7	同步定时器停止沿类型位。 用户代码置1, 停止LHSCON1[7:4]寄存器配置的同步定时器上升沿计数。 用户代码置0, 停止LHSCON1[7:4]寄存器配置的同步定时器下降沿计数。	
6	工作模式位。 用户代码置1, 选择BSD工作模式。 用户代码清0, 选择LIN工作模式。	

ADuC7036

位	描述
5	<p>使能比较中断位。</p> <p>用户代码置1，当LHSVAL0（LIN同步位定时器）值等于LHSCMP寄存器值时产生LHS中断（IRQEN[7]）。此中断产生后，LHS比较中断位LHSSTA[3]置1。此配置在BSD写入模式下允许用户代码对欲发送BSD位的输出脉冲宽度正确定时。</p> <p>用户代码清0，禁用比较中断。</p>
4	<p>使能停止中断。</p> <p>用户代码置1，在出现停止条件时产生中断。</p> <p>用户代码清0，在出现停止条件时禁止中断。</p>
3	<p>使能起始中断。</p> <p>用户代码置1，在出现起始条件时产生中断。</p> <p>用户代码清0，在出现起始条件时禁止中断。</p>
2	<p>LIN同步使能位。</p> <p>用户代码置1，使能LHS功能。</p> <p>用户代码清0，禁用LHS功能。</p>
1	<p>边沿计数器清0位。</p> <p>用户代码置1，清0LHS外设内部边沿计数器。</p> <p>15 μs延迟后自动清0。</p>
0	<p>LHS复位位。</p> <p>用户代码置1，将所有LHS逻辑复位至默认状态。</p> <p>15 μs延迟后自动清0。</p>

¹ 在BSD模式下，LHSCON0[6]置1。因为BSD发送（从MCU到外部引脚）和接收（从外部引脚到MCU）路径的有限传播延迟，用户代码必须等到MCU确认解除外部BSD引脚声明后才能切换BSD写入和读取模式。不遵守这条规则，用户代码从BSD写入模式切换到BSD模式可能会造成意外断开条件中断。停止条件中断可用来确保避免此类情形。

LIN硬件同步控制寄存器 1

名称：LHSCON1

地址：0xFFFF078C

默认值：0x00000032

访问类型：读写

功能：该32位LHS控制寄存器配合LHSCON0寄存器用来配置LIN工作模式。

表94. LHSCON1 MMR位分配

位	描述
31至8	保留。这些位保留供未来使用，应由用户软件写入0。
7至4	LIN停止沿计数。用户代码设置停止内部LIN同步计数器所需的下降或上升沿数。用户代码可利用LHSVAL0读取此计数器的停止值。沿类型（上升或下降）由LHSCON0[7]配置。这些位的默认值为0x3，配置硬件在第三个下降沿停止计数。注意，第一个下降沿视为在LIN断开脉冲起始位置处的下降沿。
3至0	LIN起始沿计数。这4位由用户代码设置，内部LIN同步定时器开始计数所必需的下降沿数。用户代码可利用LHSVAL0读取此计数器的停止值。这些位的默认值为0x2，配置硬件在第二个下降沿开始计数。注意，第一个下降沿视为在LIN断开脉冲起始位置处的下降沿。

LIN 硬件同步定时器 0 寄存器

名称: LHSVAL0

地址: 0xFFFF0788

默认值: 0x0000

访问类型: 只读

功能: 该16位只读寄存器保持内部LIN同步定时器的值。LIN同步定时器从内部5 MHz时钟接收时钟信号并与内核时钟和波特率频率无关。在LIN模式下, 用户代码可利用从LHSVAL0寄存器读取的值来计算主机LIN波特率。然后使用此计算结果来配置内部UART波特率以确保ADuC7036从节点和LIN主节点通过UART顺利通信。

LIN 硬件同步断开定时器 1 寄存器

名称: LHSVAL1

地址: 0xFFFF0790

默认值: 0x0000

访问类型: 读写

功能: 当用户代码读取此位置时, 就会返回内部LIN断开定时器的12位值, 该定时器直接从片内低功耗131 kHz振荡器接收时钟信号并对LIN断开脉冲进行定时。LIN总线上出现负沿或用户代码读取LHSVAL1都会造成将该定时器及寄存器内容复位至0。

当用户代码写入此位置时, 12位值不会写入LIN断开定时器, 而是写入LIN断开比较寄存器。在LIN工作模式下, 持续将比较器寄存器值和断开定时器值进行比较。当定时器值到达比较值时, 就会产生一个LIN断开中断 (IRQEN[7]和LHSSTA[0])。在发生断开条件中断后, LIN断开定时器仍会继续计数直到出现断开信号上升沿。如果未检测到上升沿且12位定时器上溢 ($4096 \times 1/131 \text{ kHz} = 31 \text{ ms}$), 就会产生一个断开字段错误中断 (IRQEN[7]和LHSSTA[4])。默认情况下, 比较寄存器的值为0x0047, 对应于11位周期 (即在20 kbps时LIN断开脉冲的最小脉冲宽度)。对于不同波特率, 可通过写入LHSVAL1来修改此值。注意, 如果没有接收到有效断开中断, 通过LHSVAL0定时的后续同步脉冲就不会出现。

LIN 硬件接口**LIN 帧协议**

LIN帧协议主要分为四类: 断开符号、同步字节、保护识别符和数据字节。

帧报头、断开符号、同步字节和保护识别符的格式如图44所示。基本而言, 嵌入式UART、LIN硬件同步逻辑和高压收发器接口全部片内组合以支持并管理基于LIN的发送和接收。

LIN 帧断开符号

如图45所示, LIN断开符号持续至少13位周期, 用作一个新帧开始的通知信号。从机在预期或接收数据前必须能够检测一个断开符号。如“LIN硬件同步断开定时器1寄存器”部分所述, ADuC7036利用LHSVAL1断开条件和断开错误检测功能来实现。断开周期不必精确测量, 但如果出现总线故障条件 (总线保持低电平), 就必须加以标记。

LIN 帧同步字节

如图46所示, 可利用同步字节来计算LIN通信的波特率。测量同步字段的第一个下降沿到第五个下降沿的时间并除以8, 就可以决定欲发送数据的波特率。ADuC7036硬件实现此同步字段的时序。有关此功能的详情, 参见“LIN硬件同步状态寄存器”部分。

LIN 帧保护识别符

接收LIN同步字段后, 计算所需UART波特率。如图47所示, 然后配置UART, 从而允许ADuC7036接收保护识别符。保护识别符包括两个子字段: 识别符和识别符奇偶校验。6位识别符包括帧目标的识别符。识别符表示欲收发数据字节数。字节数可由用户在进行系统级别设计时配置。奇偶校验根据识别符计算并取决于系统设计的LIN版本。

LIN 帧数据字节

数据字节帧传送1到8个字节数据。帧所包含的字节数取决于LIN主机。如图48所示, 数据字节帧分割成若干数据字节。

ADuC7036

LIN 帧数据发送和接收

在正确接收断开符号和同步字节并配置所需UART波特率后，可通过COMTX和COMRX MMR来收发数据。配置UART以配合LIN使用要用到下列几个UART MMR：

- COMDIV0：分频锁存器（低字节）。
- COMDIV1：分频锁存器（高字节）。
- COMDIV2：16位小数波特率分频寄存器。COMDIV0、COMDIV1和COMDIV2的所需值从LHSVAL0获得以产生所需波特率。
- COMCON0：线路控制寄存器。正确配置UART后，LIN数据收发协议便和UART规格相同。

管理LIN总线上的数据要用到下列几个UART MMR：

- COMTX：8位发送寄存器
- COMRX：8位接收寄存器
- COMCON0：线路控制寄存器
- COMSTA0：线路状态寄存器

此外，在LIN总线上发送数据需要在COMTX内放置相关数据，且读取LIN总线上接收的数据需要对COMRX进行监控。要确保数据顺利收发，应监控COMSTA0。有关详情，参见“UART串行接口”和“UART寄存器定义”部分。

通过软件控制，可多路复用UART数据线路（TxD和Rx）至外部GPIO_7/IRQ4和GPIO_8/IRQ5引脚。有关详情，参见“GPIO端口1控制寄存器（GP1CON）”部分。

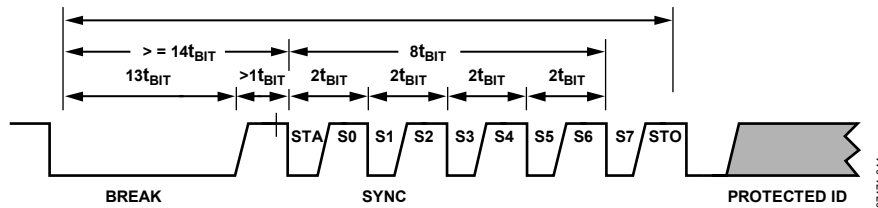


图 44. LIN 接口时序

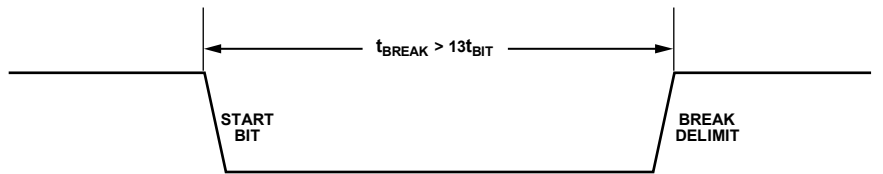


图 45. LIN 断开字段

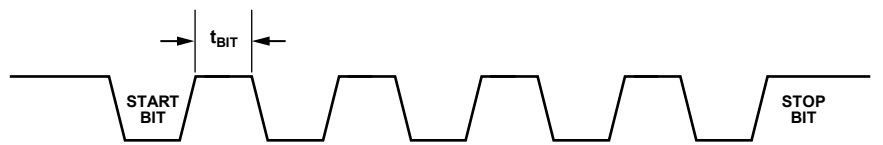


图 46. LIN 同步字节字段

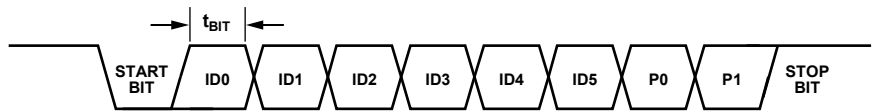


图 47. LIN 识别符字节字段



图 48. LIN 数据字节字段

LIN 硬件同步程序实例

利用下列C源代码LIN初始化程序，LHSVAl1在LIN总线上接收的第一个下降沿开始计数。如果LHSVAl1超过LHSVAl1写入值（在此情况下为0x3F），就会产生一个断开比较中断。

在下一个下降沿，LHSVAl0开始计数。LHSVAl0监控下降沿数并将其和LHSCON1[7:4]写入值进行比较。在此实例

中，欲监控的沿数是LIN帧的6个下降沿或同步字节的5个下降沿。当接收到此数目的下降沿时，就会产生一个停止条件中断。此时配置UART以接收保护识别符。

在LIN总线恢复高电平前必须通过LHSCON0[8]选通UART。如果LIN总线恢复高电平，但UART未被选通，就可能会出现UART通信错误。这个过程的细节如图49所示。代码实例遵循图49以确保此过程成功。

```
void LIN_INIT(void)
{
    char HVstatus;
    GP2CON = 0x110000;    // Enable LHS on GPIO pins

    LHSCON0 = 0x1;        // Reset LHS interface

    do{
        HVDAT = 0x02;    // Enable normal LIN Tx mode
        HVCON = 0x08;    // Write to Config0
        do{
            HVstatus = HVCON;
        }
        while(HVstatus & 0x1); // Wait until command is finished
    }
    while (!(HVstatus & 0x4)); // Transmit command is correct

    while((LHSSTA & 0x20) == 0)
    {
        // Wait until the LHS hardware is reset
    }

    LHSCON1 = 0x062; // Sets stop edge as the fifth falling edge
                    // and the start edge as the first falling
                    // edge in the sync byte
    LHSCON0 = 0x0114; // Gates UART Rx line, ensuring no interference
                    // from the LIN into the UART
                    // Selects the stop condition as a falling edge
                    // Enables generation of an interrupt on the
                    // stop condition
                    // Enables the interface
    LHSVAl1 = 0x03F; // Sets number of 131 kHz periods to generate a break interrupt
                    // 0x3F / 131 kHz ~ 480 μs, which is just over 9.5 Tbits
```

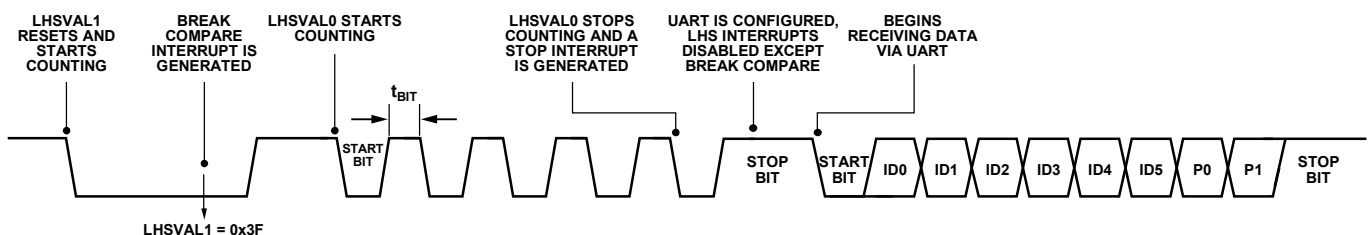


图49. LIN 配置示例

```
while((GP2DAT & 0x10) == 0)
{
    // Wait until LIN Bus returns high
    LHSCON0 = 0x4; // Enable LHS to detect Break Condition Ungate RX Line
                  // Disable all Interrupts except Break Compare Interrupt
    IRQEN = 0x800; // Enable UART Interrupt
                  // The UART is now configured and ready to be used for LIN
```

ADuC7036

LIN 诊断

ADuC7036具有非介入监控LIN/BSD引脚当前状态的功能。此回读功能利用GPIO_11来实现。LIN/BSD引脚当前状态包括在GP2DAT[4]内。

还可通过用户软件将LIN/BSD引脚驱动至高电平和低电平，从而允许用户检测开路状况。此功能通过GPIO_12来实现。要使能此功能，必须通过GP2CON[20]将GPIO_12配置成GPIO。完成配置后可利用GP2DAT将LIN/BSD引脚拉高或拉低。

ADuC7036在LIN/BSD引脚上还具有短路保护功能。如果在LIN/BSD引脚上检测到短路状况，HVSTA[2]就会置1。利用HVCFG1[3]再使能LIN驱动器可将此位清0。可通过HVCFG1[2]来禁用此功能。

在热关断期间的LIN操作

当发生过热事件时，即当HVSTA[3]置1时，LIN通信仍会继续，不被打断。

位串行设备（BSD）接口

BSD是一种具有三种可能状态（同步、0和1）的脉冲宽度调制信号。这些状态及相关容差详细列于表95内。帧长度为19位，且通信发生在 $1200 \text{ bps} \pm 3\%$ 。

BSD通信硬件接口

通过软件控制，ADuC7036利用一个GPIO、一个IRQ和LIN同步硬件来仿真BSD通信协议。

表95.BSD位水平描述

参数	最小值	典型值	最大值	单位
TxD速率	1164	1200	1236	bps
位编码				
t_{SYNC}	1/16	2/16	3/16	t_{PERIOD}
t_0	5/16	6/16	8/16	t_{PERIOD}
t_1	10/16	12/16	14/16	t_{PERIOD}

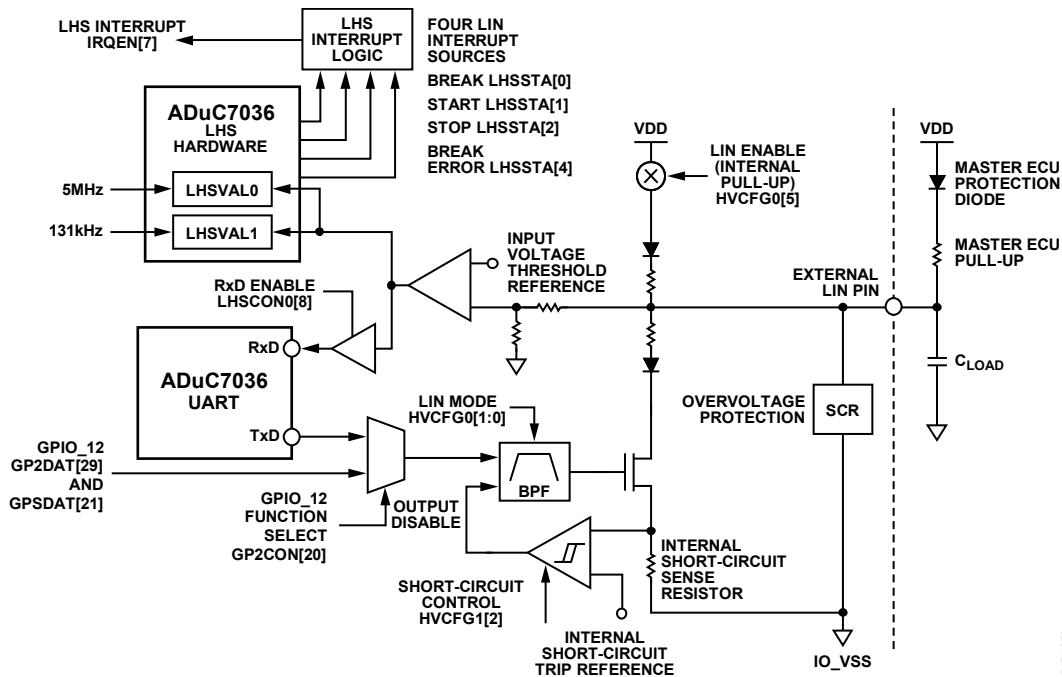


图 50. BSD I/O 硬件接口

07474-050

ADuC7036

BSD相关MMR

ADuC7036利用一个软件（位拆裂）接口及其一些硬件辅助形式LIN硬件同步逻辑来仿真BSD通信协议。事实上，ADuC7036 BSD接口利用下列协议：

- 一个内部GPIO信号（GPIO_12），它被路由到外部LIN/BSD引脚并直接由软件控制以产生0和1。
- 当读取多位时，LIN同步硬件利用LHSVAL1对输入脉冲的宽度进行计数，这样用户代码可以将这些位解读为同步、0或1。
- 当写入多位时，用户代码触发一个GPIO引脚并利用LHSCAP和LHSCMP给脉冲宽度定时并在BSD输出脉冲宽度到达所需宽度时产生一个中断。

下列是BSD通信所需的ADuC7036 MMR：

- LHSSTA：LIN硬件同步状态寄存器
- LHSCON0：LIN硬件同步控制寄存器
- LHSVAL0：LIN硬件同步定时器0（16位定时器）
- LHSCON1：LIN硬件同步沿设置寄存器
- LHSVAL1：LIN硬件同步断开定时器
- LHSCAP：LIN硬件同步捕捉寄存器
- LHSCMP：LIN硬件同步比较寄存器
- IRQEN/IRQCLR：使能中断寄存器
- FIQEN/FIQCLR：使能快速中断寄存器
- GP2DAT：GPIO端口2数据寄存器
- GP2SET：GPIO端口2设置寄存器
- GP2CLR：GPIO端口2清0寄存器

前面已列出这些MMR中大部分的详细位定义。除“LIN MMR 描述”部分中所述的寄存器外，LHSCAP和LHSCMP也是BSD接口操作所必需的寄存器。这些寄存器的详情如下。

LIN硬件同步捕捉寄存器

名称：LHSCAP

地址：0xFFFF0794

默认值：0x0000

访问类型：只读

功能：该16位只读寄存器保持内部LIN同步定时器（LHSVAL0）的最后一个捕捉值。在BSD模式下，LHSVAL0直接从一个内部5 MHz时钟接收时钟信号，且每次出现BSD总线下沿时将其值加载到该捕捉寄存器内。

LIN硬件同步比较寄存器

名称：LHSCMP

地址：0xFFFF0798

默认值：0x0000

访问类型：读写

功能：用来对BSD输出脉冲宽度进行定时。通过LHSCON0[5]使能后，在LHSCAP值等于LHSCMP写入值时就会产生一个LIN中断。此功能允许用户代码决定在总线上应声明一个BSD发送位（同步、0或1）的时间。

BSD通信帧

要在主机和从机之间传输数据，必需先构造一个BSD帧。BSD帧包括7个关键部分：暂停/同步、一个方向（DIR）位、从机地址、寄存器地址、数据、奇偶校验位（P1和P2）以及来自从机的应答位。

如果从机正在发送数据，则除应答位外所有位都由主机发送。

如果主机正向从机请求数据，则主机发送暂停/同步、方向位、从机地址、寄存器地址和P1。从机然后按照下列序列发送数据字节、P2位和应答位：

1. 暂停：≥ 三个同步脉冲
2. DIR：表示数据传输方向
 - 如果主机发送请求，则DIR = 0
 - 如果从机发送请求，则DIR = 1
3. 从机地址
4. 寄存器地址：定义欲读写的寄存器
5. 位3置1为写入，清0为读取。
6. 数据：8位只读接收寄存器
7. P1和P2
 - 如果8个前面位中有偶数个1，则P1 = 0
 - 如果8个前面位中有奇数个1，则P1 = 1
 - 如果数据字中有偶数个1，则P2 = 0
 - 如果数据字中有奇数个1，则P2 = 1
8. 应答位
 - 如果发送成功，则ACK = 0

应答位总是由从机发送以指示是否已接收或发送信息。

表96. BSD协议描述

暂停	DIR	从机地址	寄存器地址	P1	数据	P2	ACK
3位	1位	3位	4位	1位	8位	1位	1位

BSD脉冲宽度实例

不同脉冲宽度的实例如图51所示。对于每一位，总线保持低电平的周期定义了位类型。如果是同步位，则脉冲低电平持续1位。如果是0，则脉冲低电平持续3位。如果是1，则脉冲低电平持续6位。

如果主机正在发送数据，则主机在整个信号发送期间将信号保持低电平。主机正在发送一个0的实例如图52所示。如果从机正在发送数据，则主机将总线拉低以开始通信。从机必须在 t_{SYNC} 完成之前将总线拉低，然后总线保持低电平直到经过 t_0 或 t_1 ，此后从机释放总线。从机正在发送一个0的实例如图53所示。

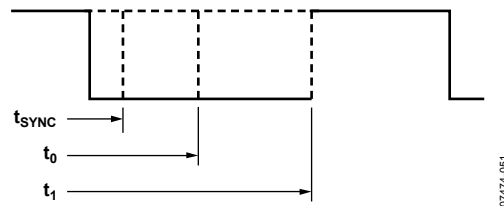


图51. BSD 位发送

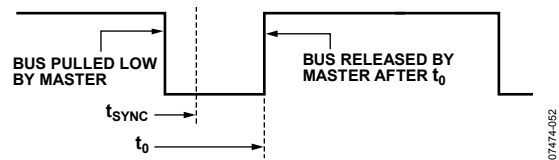


图52. BSD 主机正在发送一个0

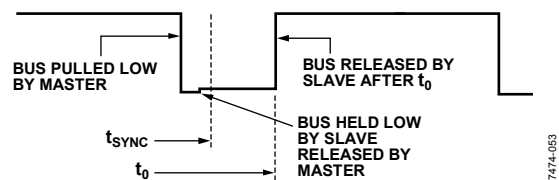


图53. BSD 从机正在发送一个0

典型BSD程序流程

因为BSD是一个由软件控制的PWM通信协议，所以用户必须逐位构造所需数据。例如，在构造从机地址过程中，从机节点接收三位，然后用户构造相关地址。

当由主机发起BSD通信时，从机发送并接收数据。此过程如流程图54所示。

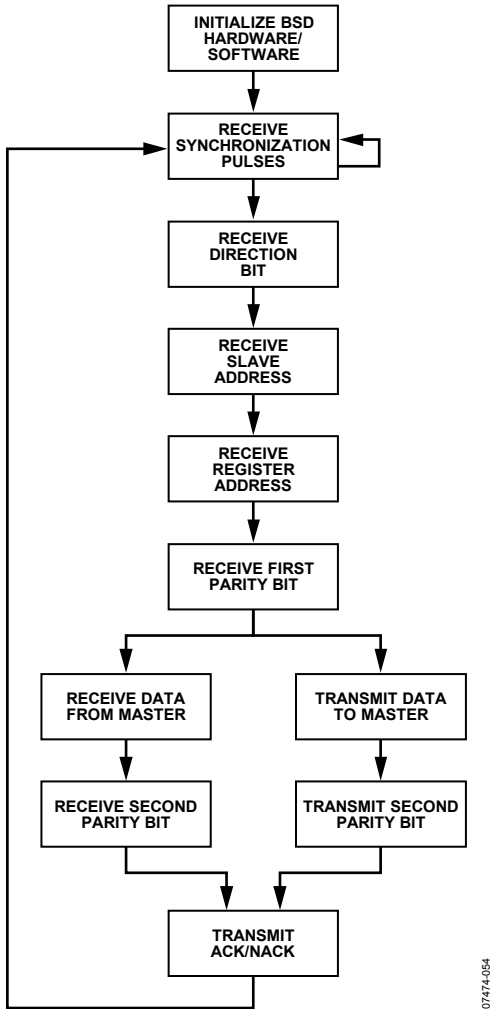


图 54. BSD 从机节点状态机

07474-054

BSD数据接收

要接收数据，必须先在BSD模式（LHSCON0[6]=1）下配置LIN/BSD外设。在此模式下，LHSCON0[8]应置1以确保LHS断开定时器（参见“LIN硬件同步断开定时器1寄存器”部分）在BSD总线的上升沿会产生一个中断。

LHS断开定时器清0并在BSD总线的下降沿开始计数；随后停止并在BSD总线的上升沿产生一个中断。假设LHS断开定时器从低功耗131 kHz振荡器接收时钟信号，则用户代码可解读LHSVAL1值以决定所接收数据位是一个BSD同步脉冲、0还是1。

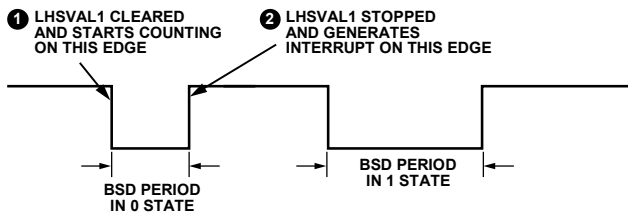


图 55. 主机发送从机读取

07474-055

BSD主机发送

用户代码强制GPIO_12信号呈低电平持续一定时间以在BSD模式下发送数据。此外，用户代码还利用同步定时器（LHSVAL0）、LHS同步捕捉寄存器（LHSCAP）和LHS同步比较寄存器（LHSCMP）来决定BSD总线应保持低电平的时间以在0或1状态下发送数据位。

如“BSD脉冲宽度实例”部分所述，即使从机正在发送数据，主机仍始终通过将BSD总线拉低来开始位发送周期。如果选择BSD模式（LHSCON0[6]=1），则在BSD总线的每个下降沿都会在LHSCAP内捕捉LIN同步定时器值。该LIN同步定时器在BSD模式下持续运行。

然后用户代码可立即强制GPIO_12呈低电平并从LHSCAP读取所捕捉的定时器值。接下来，用户可针对在0或1状态下的脉冲宽度计算在将GPIO_12驱动至高电平前应经过的时钟周期（使用5 MHz时钟）。计算后的数字可相加到LHSCAP值并写入LHSCMP寄存器内。如果LHSCON0[5]置1，则同步计时器继续计数（采用5 MHz时钟提供时钟信号），最终等于LHSCMP值并产生一个LHS比较中断（LHSSTA[3]）。

此中断的响应应迫使GPIO_12信号（并因此BSD总线）为高电平。如图56所示，通过软件控制GPIO_12信号并配合正确使用LIN同步定时器，可确保ADuC7036发送0和1状态的有效脉冲宽度。再次提醒，如表93中所示（参见LHSCON0[8]位），从BSD写入模式切换到BSD读取模式时务必小心。

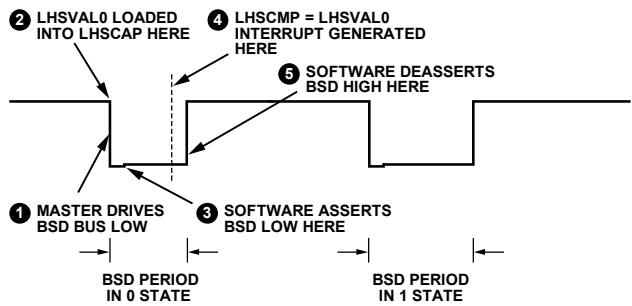


图 56. 主机读取从机发送

07474-056

BSD接口唤醒

MCU内核可通过BSD物理接口从省电状态唤醒。进入省电模式前，用户代码应先使能起始条件中断（LHSCON0[3]）。使能后在LIN/BSD引脚上的一个高至低跳变就会产生一个中断事件并唤醒MCU内核。

器件标识

有两个寄存器映射到MMR空间，允许用户代码识别并追溯制造批次ID信息、器件ID号、硅掩膜版本和内核版本。这些信息包含在SYSSER0和SYSSER1 MMR内（详情见表98和99）。

此外，FEE0ADR MMR还包含上电信息，可识别ADuC7036系列产品。

为了直接可追溯性，还提供组装批次ID，它可以是64位长。上电时，SYSALI MMR包含该组装批次ID的32位低半部分，而高半部分则包含在T1LD MMR内。

SYSSER0、SYSSER1、SYSALI和T1LD内包含的信息实现

了每个器件的完全可追溯。

如表97所示，批次号是封装标识的一部分。

表97. 标识实例

行	LFCSP
行1	ADuC7036
行2	BCPZ
行3	A40 # 日期代码
行4	组装批次号

系统串行ID寄存器0

名称：SYSSER0

地址：0xFFFF0238

默认值：0x00000000（上电时由内核更新）

访问类型：读写

功能：上电时此32位寄存器保持该具体ADuC7036单元的原始制造批次号值（限底层芯片）。配合SYSSER1，此批次号可追溯此器件的整个制造历史（限底层芯片）。

表98. SYSSER0 MMR Bit位分配

位	描述
31至27	晶圆号。从此位置读取的5位值从晶圆制造批次ID（此器件出处）提供晶圆号（1至24）。配合SYSSER0[26:0]，它提供个别晶圆可追溯性。
26至22	晶圆批次制造厂。从此位置读取的5位值反映与此晶圆批次相关的制造厂。配合SYSSER0[21:0]，它提供晶圆批次可追溯性。
21至16	晶圆批次制造ID。从此位置读取的6位值是晶圆批次制造ID的一部分且配合SYSSER0[26:22]和SYSSER0[15:0]，提供晶圆批次可追溯性。
15至0	晶圆批次制造ID。这些16位LSB被解读为晶圆制造批次ID号。配合SYSSER1值（即制造批次ID），此编号是器件的唯一标识。

ADuC7036

系统串行ID寄存器1

名称: SYSSER1

地址: 0xFFFF023C

默认值: 0x00000000 (上电时由内核更新)

访问类型: 读写

功能: 上电时, 该32位寄存器保持器件ID号、硅片掩膜版本及内核版本号(限底层芯片)的值, 详情见表99。

表99. SYSSER1 MMR位分配

位	描述
31至28	硅片掩膜版本ID。此半字节反映硅片掩膜ID号。具体来说, 这半字节的16进制值应被解码成低位半字节, 表示ASCII字符A至O。例如, 如果位[19:16] = 0001 = 0x1, 此值应解释为41, 对应ASCII字符A, 表示硅片掩膜版本A。如果位[19:16] = 1011 = 0xB, 则解释为4B, 对应ASCII字符K, 表示硅片掩膜版本K。这个值的范围为1至15, 解释为41至4F或ASCII字符A至O。
27至20	内核版本ID。此字节的16进制数应解释为一个ASCII字符, 表示片内Flash/EE存储器的嵌入内核固件版本。例如, 如果从此字节读取0x41, 则解释为A, 表示片上内核版本A。
19至16	保留。对于预发行样片, 这些位表示器件内核的局部修订版本号。
15至0	器件ID。这些16位LSB被解释为器件ID号。配合SYSSER0值(即制造批次ID), 此编号是器件的唯一标识。

系统组装批次ID

名称: SYSALI

地址: 0xFFFF0560

默认值: 0x00000000 (上电时由内核更新)

访问类型: 读写

功能: 上电时, 该32位寄存器保持组装批次ID的低半部分。例如, 如果组装批次ID为01308640, 则上电时SYSALI保持0x38363430, 而T1LD保持0x30313330。

系统内核校验和

名称: SYSCHK

地址: 0xFFFF0240

默认值: 0x00000000 (上电时由内核更新)

访问类型: 读写

功能: 上电时, 该32位寄存器保持内核校验和。

系统标识FEE0ADR

名称：FEE0ADR

地址：0xFFFF0E10

默认值：非零

访问类型：读写

功能：该16位寄存器决定通过FEE0CON执行任何Flash/EE命令的操作地址。

注意，该MMR还用以识别ADuC7036系列产品和预发布芯片版本。

表100. FEE0ADR系列标识MMR位分配

位	描述
15至4	保留
3至0	ADuC703x系列ID 0x2 = ADuC7032 0x3 = ADuC7033 0x4 = ADuC7034 0x6 = ADuC7036 其他 = 保留供将来使用

ADuC7036

原理图

此原理图实例仅表示基本功能电路实现。要确保系统满足任何EMC和其他过压/过流兼容性要求，还需添加其他元件。

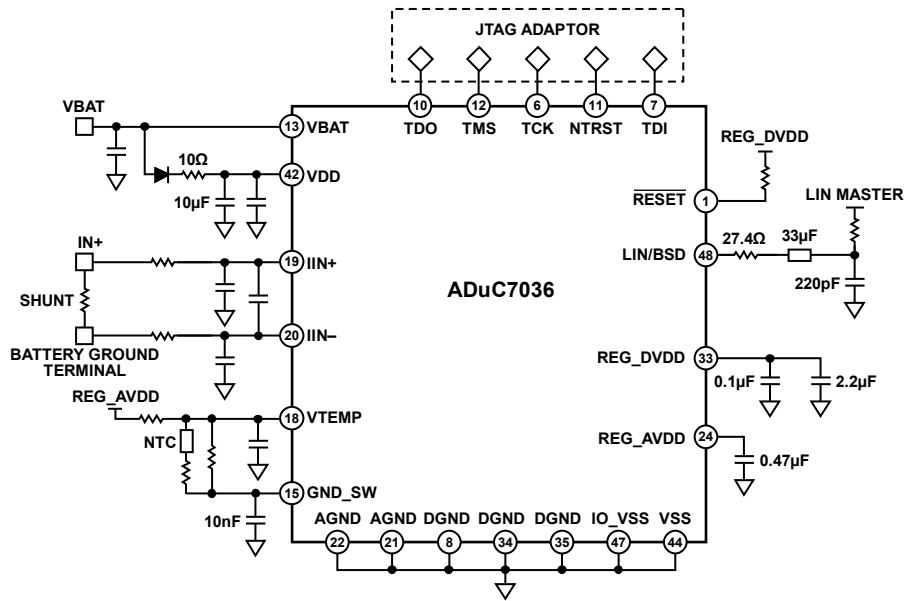
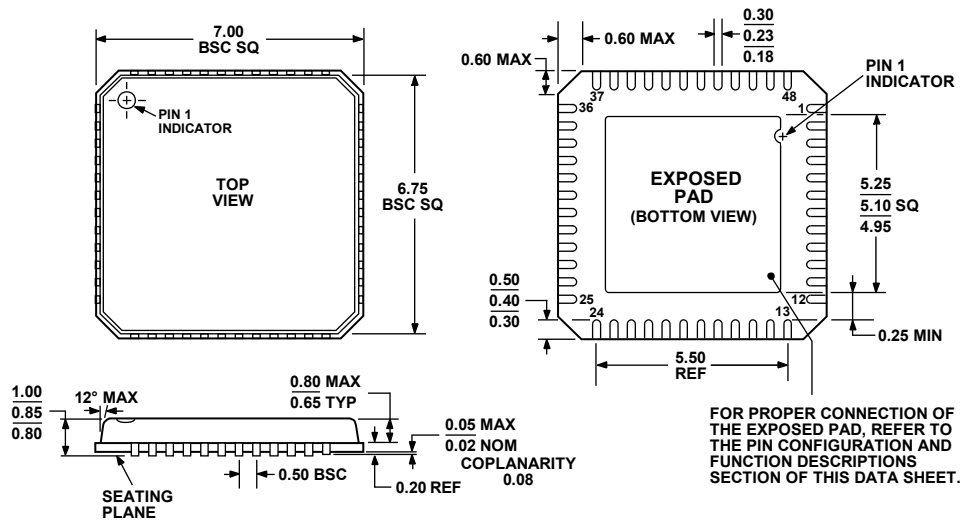


图 57. 简化原理图

07474-057

外形尺寸



COMPLIANT TO JEDEC STANDARDS MO-220-VKGD-2

图 58. 48 引脚 LFCSP_VQ 封装, 7 mm x 7 mm, 超薄体 (CP-48-1)
尺寸单位: mm

订购指南

型号	温度范围	封装描述	型号信息	封装选项
ADUC7036BCPZ ¹	-40°C至+115°C	48引脚LFCSP_VQ	10 MHz	CP-48-1
ADUC7036BCPZ-RL ¹	-40°C至+115°C	48引脚LFCSP_VQ	10 MHz	CP-48-1
ADUC7036CCPZ ¹	-40°C至+115°C	48引脚LFCSP_VQ	20 MHz	CP-48-1
ADuC7036CCPZ-RL ¹	-40°C至+115°C	48引脚LFCSP_VQ	20 MHz	CP-48-1

¹ Z = 符合RoHS标准的兼容器件。

注释

注释

