

特征

- 28/56位, 50 MIPS数字音频处理器
- 2个ADC: SNR为100 dB, THD + N为-83 dB
- 4个DAC: SNR为104 dB, THD + N为-90 dB
- 完成独立操作
 - 从串行EEPROM自引导
 - 具有4输入多路复用器的辅助ADC用于模拟控制
 - GPIO用于数字控制和输出
- 使用SigmaStudio图形工具完全可编程
- 具有56位累加器的28位×28位乘法器
- 双精度处理
- 时钟振荡器, 用于从晶振产生主时钟
- 用于从 $64 \times f_s$, $256 \times f_s$ 产生主时钟的PLL, $384 \times f_s$ 或 $512 \times f_s$ 时钟
- 灵活的串行数据输入/输出端口, 兼容I2S, 左对齐, 右对齐和TDM模式
- 支持高达192 kHz的采样率
- 片上稳压器与3.3 V系统兼容
- 48引脚塑料LQFP
- 合格的汽车应用

应用

- 多媒体音箱系统
- MP3播放扬声器码头
- 汽车头部单位
- 小型立体声
- 数字电视
- Studio监视器
- 扬声器交叉
- 乐器效果处理器
- 座椅内音响系统(飞机/摩托车)

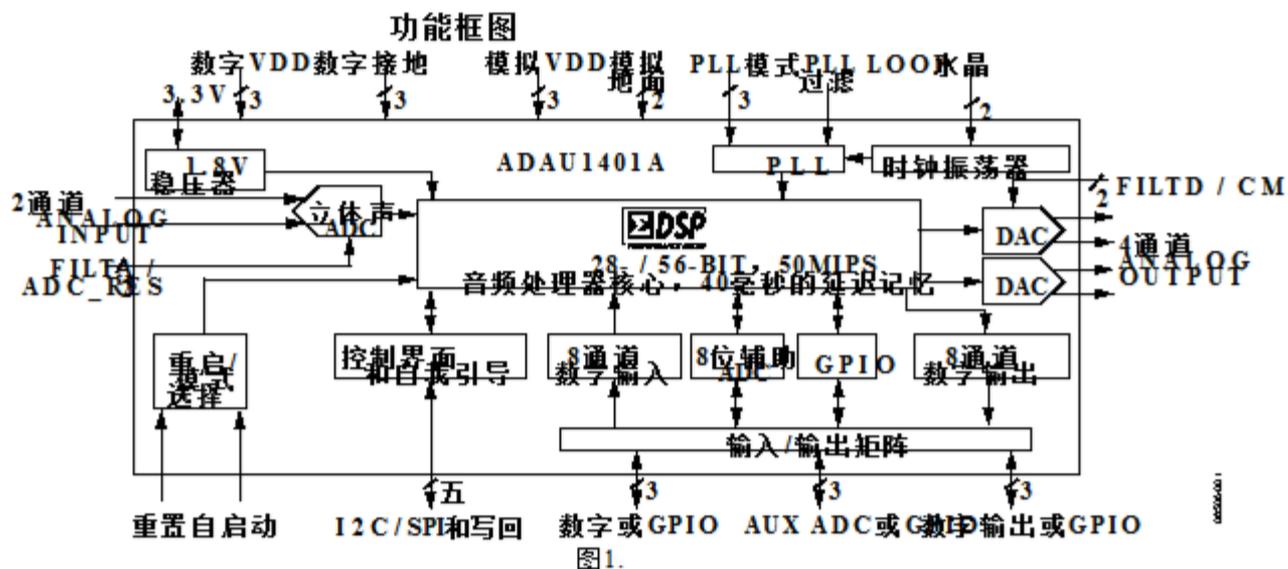
一般描述

ADAU1401A是一款完整的单芯片音频系统28/56位音频DSP, ADC, DAC和微控制器控制接口. 信号处理包括均衡, 低音增强, 多频带动态处理, 延迟补偿, 扬声器补偿和立体声图像扩大. 这个处理可以用来补偿现实世界的限制, 扬声器, 放大器和听音环境, 提供感知音频质量的显著提高.

ADAU1401A的信号处理与此相当发现在高端的演播室设备. 大多数处理都是在完整的56位, 双精度模式, 导致非常好的低级信号性能. ADAU1401A是一款完全基于程序的电路, 可编程DSP易于使用的SigmaStudio™软件允许用户以图形方式配置自定义信号处理流程. 使用诸如双二阶滤波器, 动态处理器, 电平等块控制和GPIO接口控制.

ADAU1401A程序也可以在加电时加载从串行EEPROM通过自己的自启动机制或来自外部微控制器. 在关机时, 电流参数的状态可以写回到EEPROM中. ADAU1401A将在程序下次运行时被调用.

两个 Σ - Δ 型ADC和四个 Σ - Δ 型DAC提供98.5 dB的模拟信号输入到模拟输出动态. 每个ADC都有一个THD + N -83 dB, 每个DAC的THD + N为-90 dB. 数字输入和输出端口允许无缝连接到额外的ADC和DAC. ADAU1401A通过I2C®总线进行通信或一个4线SPI®端口.



Rev. A

ADI公司提供的信息被认为是准确和可靠的, 但是没有使用模拟装置所承担的责任, 以及其他专利权的损害. rightsofthirdpartiesthatmayresultfromitsuse. Specifications subject to change without notice. ADI公司的任何专利或专利权均以暗示或其他方式授予许可. 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.
 电话: 781.329.4700
 传真: 781.461.3113
 www.analog.com
 ©2010 Analog Devices, Inc. 保留所有权利.

ADAU1401A *产品页面快速链接

上次内容更新: 02/23/2017

可比较的部件

查看类似部件的参数搜索.

评估套件

- ADAU1401评估板
- ADUSB2EBZ评估板

文档

应用笔记

- AN-1006: 使用EVAL-ADUSB2EBZ
- AN-923: 使用ADAU1701 / ADAU1702处于自启动模式
- AN-951: 使用SigmaDSP GPIO的硬件控制

数据表

- ADAU1401A: SigmaDSP 28/56位音频处理器
两个ADC和四个DAC数据表

用户指南

- UG-072: ADAU1401评估板用户指南

软件和系统要求

- ADAU1701声音音频系统Linux驱动程序
- 适用于SigmaDSP的固件加载程序

工具和模拟

- SigmaDSP处理器: 软件和工具

设计资源

- ADAU1401A材料声明
- PCN-PDN信息
- 质量和可靠性
- 符号和脚印

讨论

查看所有ADAU1401A EngineerZone讨论.

样品和购买

访问产品页面查看定价选项.

技术支持

提交技术问题或找到您的区域支持数.

文件反馈

提交此数据表的反馈.

目录

特征	1	地址地图	30
应用	1	参数RAM	30
一般说明	1	数据RAM	30
功能框图	1	读/写数据格式	30
修订记录	3	控制寄存器映射	32
规格	4	控制寄存器详细信息	34
模拟性能	4	地址2048到地址2055 (0x0800到0x0807) - 接口	
数字输入/输出	5	寄存器	34
功率	6	地址2056 (0x0808) -GPIO引脚设置寄存器	35
温度范围	6	地址2057到地址2060 (0x0809到0x080C) -	
PLL和振荡器	6	辅助ADC数据寄存器	36
稳压器	6	地址2064到地址2068 (0x0810到0x0814) -Safeload	
数字时钟规格	7	数据登记	37
绝对最大额定值	10	地址2069到地址2073 (0x0815到0x0819) -Safeload	
热阻	10	地址寄存器	37
ESD警告	10	地址2074和地址2075 (0x081A和0x081B) - 数据	
引脚配置和功能描述	11	捕获登记	38
典型性能特点	14	地址2076 (0x081C) -DSP内核控制寄存器	39
系统框图	15	地址2078 (0x081E) - 串行输出控制寄存器	
操作理论	16	地址2079 (0x081F) - 串行输入控制寄存器	
初始化	17	地址2080和地址2081 (0x0820和0x0821) -	
加电顺序	17	多用途引脚配置寄存器	42
控制寄存器设置	17	地址2082 (0x0822) - 辅助ADC和功率控制	
推荐程序/参数加载程序	17	注册	43
省电模式	17	地址2084 (0x0824) - 辅助ADC使能寄存器	43
使用振荡器	18	地址2086 (0x0826) - 振荡器关断寄存器	43
设置主时钟/ PLL模式	18	地址2087 (0x0827) -DAC设置寄存器	43
电压调节器	19	多用途引脚	44
音频ADC	20	辅助ADC	44
音频DAC	21	通用输入/输出引脚	44
控制港口	22	串行数据输入/输出端口	44
I2C端口	23	布局建议	47
SPI端口	26	零件放置	47
自开机	27	接地	47
信号处理	29	典型应用原理图	48
数字格式	29	自引导模式	48
编程	29	I2C控制	49
RAM和寄存器	30	SPI控制	50
		外形尺寸	51
		订购指南	51

修订记录**11/10-REV. 0至Rev. A**

图7和表11的变化.....	11
图37的变化.....	48
图38的变化.....	49
图39的变化.....	50

4/10修订版0: 初始版本

ADAU1401A

规格

除非另有说明，AVDD = 3.3V，DVDD = 1.8V，PVDD = 3.3V，IOVDD = 3.3V，主时钟输入 = 12.288MHz。

模拟性能

规格保证在25°C（环境）。

表格1.

参数	敏	典型	马克	单元	测试条件/评论
ADC输入					
通道数		2			立体声输入
解析度		24		位	
满量程输入		100 (283)		μArms (μApp)	2 V rms输入，20k Ω (18k Ω 外部+ 2k Ω 内部) 串联电阻
信噪比					
A加权		100		D b	
动态范围					-60 dB相对于满量程模拟输入
A加权	95	100		D b	
总谐波失真+噪声		-83		D b	-3 dB相对于满量程模拟输入
通道间增益不匹配		25	300	MDB	
相声		-82		D b	模拟通道间串扰
DC偏置	1.4	1.5	1.6	V	
增益错误	-11		+11	%	
DAC输出					
通道数		4			两个立体声输出通道
解析度		24		位	
满量程模拟输出		0.9 (2.5)		V rms (V pp)	正弦波
信噪比					
A加权		104		D b	
动态范围					相对于满量程模拟输出为-60 dB
A加权	99	104		D b	
总谐波失真+噪声		-90		D b	相对于满量程模拟输出为-1 dB
相声		-100		D b	模拟通道间串扰
通道间增益不匹配		25	250	MDB	
增益错误	-10		+10	%	
DC偏置	1.4	1.5	1.6	V	
电压参考					
绝对电压，CM引脚	1.4	1.5	1.6	V	
辅助ADC					
满量程模拟输入	2.8	2.95	3.1	V	
INL		0.5		LSB	
DNL		0.5		LSB	
抵消		15		毫伏	
输入阻抗	17.8	三十	42	千欧	

规格保证在130°C（环境）。

表2

参数	敏	典型	马克	单元	测试条件/评论
ADC输入					
通道数		2			立体声输入
解析度		24		位	
满量程输入		100 (283)		μArms (μApp)	2 V rms输入，20k Ω (18k Ω 外部+ 2k Ω 内部) 串联电阻

参数	敏	典型	马克	单元	测试条件/评论
信噪比					
A加权		100		Db	
动态范围					-60 dB相对于满量程模拟输入
A加权	92	100		Db	
总谐波失真+噪声		-83		Db	-3 dB相对于满量程模拟输入
通道间增益不匹配		25	250	MDb	
相声		-82		Db	模拟通道间串扰
DC偏置	1.4	1.5	1.6	V	
增益错误	-11		+11	%	
DAC输出					
通道数		4			两个立体声输出通道
解析度		24		位	
满量程模拟输出		0.85 (2.4)		V rms (V pp)	正弦波
信噪比					
A加权		104		Db	
动态范围					相对于满量程模拟输出为-60 dB
A加权	98	104		Db	
总谐波失真+噪声		-90		Db	相对于满量程模拟输出为-1 dB
相声		-100		Db	模拟通道间串扰
通道间增益不匹配		25	250	MDb	
增益错误	-10		+10	%	
DC偏置	1.4	1.5	1.6	V	
电压参考					
绝对电压, CM引脚	1.4	1.5	1.6	V	
辅助ADC					
满量程模拟输入	2.8	2.95	3.1	V	
INL		0.5		LSB	
DNL		0.5		LSB	
抵消		15		毫伏	
输入阻抗	17.8	三十	42	千欧	

数字输入/输出

表3.

参数	敏	典型	MAX1	单元	测试条件/评论
输入电压, 高 (V _{IH})	2.0		IOVDD	V	
输入电压, 低 (V _{IL})			0.8	V	
输入泄漏, 高 (I _{IH})			1	μA	不包括MCLKI
输入泄漏, 低 (I _{IL})			1	μA	不包括MCLKI和双向引脚
双向引脚上拉电流, 低			150	μA	
MCLKI输入泄漏, 高 (I _{IH})			3	μA	
MCLKI输入泄漏, 低 (I _{IL})			3	μA	
输出电压, 高 (V _{OH})	2.0			V	我 OH = 2毫安
输出电压, 低 (V _{OL})			0.8	V	我 OL = 2毫安
输入电容			五	pF的	
GPIO输出驱动器		2		嘛	

1 最大规格是在整个温度范围内测得的

-40°C至+130°C (case), DVDD范围为1.62V至1.98V, 以及AVDD范围为2.97V至3.63V.

ADAU1401A

功率

表4

参数	敏	典型	MAX1	单元
电源电压				
模拟电压		3.3		V
数字电压		1.8		V
PLL电压		3.3		V
IOVDD电压		3.3		V
电源电流				
模拟电流 (AVDD和PVDD)		50	85	嘛
数字电流 (DVDD)		25	40	嘛
模拟电流, 复位		35	55	嘛
数字电流, 重置		1.5	4.5	嘛
耗散				
操作 (AVDD, DVDD, PVDD) 2		259.5		毫瓦
重置, 所有的供应		118		毫瓦
电源抑制比 (PSRR)				
1 kHz, 200 mV pp信号在AVDD		50		Db

1 最大规格是在整个温度范围内测量的 -40°C至+130°C (case), DVDD范围为1.62V至1.98V, 以及AVDD范围为2.97V至3.63V.

2 功耗不包括IOVDD电源, 因为从该电源汲取的电流取决于数字输出引脚的负载.

温度范围

表5

参数	敏	典型	马克斯	单元
功能保证	-40		+105	°C的环境

PLL和振荡器

表6

参数1	敏	典型	马克斯	单元
PLL工作范围		MCLK_Nom - 20%	MCLK_Nom + 20%	兆赫
PLL锁定时间			20	女士
晶体振荡器跨导 (gm)		78		位mmho

1 最大规格是在整个温度范围内测量的 -40°C至+130°C (case), DVDD范围为1.62V至1.98V, 以及AVDD范围为2.97V至3.63V.

稳压器

表7

参数1	敏	典型	马克斯	单元
DVDD电压	1.7	1.8	1.84	V

1 稳压器规格是使用Zetex Semiconductors FZT953晶体管在电路中计算的.

数字时钟规格

表8

参数1	限制		单元	描述
	t MIN	最大		
主时钟				
MP	36	244	NS	MCLKI周期, 512×fS 模式.
MP	48	366	NS	MCLKI周期, 384×fS 模式.
MP	73	488	NS	MCLKI周期, 256×fS 模式.
MP	291	1953年	NS	MCLKI周期, 64×fS 模式.
串行端口				
t BIL	40		NS	INPUT_BCLK低脉冲宽度.
t BIH	40		NS	INPUT_BCLK高脉冲宽度.
LIS	10		NS	INPUT_LRCLK设置;时间到INPUT_BCLK上升.
T LIH	10		NS	INPUT_LRCLK保持;时间从INPUT_BCLK上升.
SIS	10		NS	SDATA_INx设置;时间到INPUT_BCLK上升.
SIH	10		NS	SDATA_INx保持;时间从INPUT_BCLK上升.
t LOS	10		NS	从机模式下的OUTPUT_LRCLK设置.
t LOH	10		NS	OUTPUT_LRCLK保持从模式.
t TS		五	NS	OUTPUT_BCLK下降到OUTPUT_LRCLK时序歪斜.
SODS		40	NS	从模式下的SDATA_OUTx延迟;从OUTPUT_BCLK下降的时间.
SODM		40	NS	主模式下SDATA_OUTx延迟;从OUTPUT_BCLK下降的时间.
SPI端口				
f CCLK		6.25	兆赫	CCLK频率.
CCPL	80		NS	CCLK脉冲宽度低.
CCPH	80		NS	CCLK脉冲宽度高.
t CLS	0		NS	CLATCH设置;时间到CCLK上升.
t CLH	100		NS	CLATCH保持;时间从CCLK上升.
t CLPH	80		NS	CLATCH脉冲宽度高.
CDS	0		NS	CDATA设置;时间到CCLK上升.
CDH	80		NS	CDATA持有;时间从CCLK上升.
t COD		101	NS	COUT延迟;时间从CCLK下降.
我 2 C端口				
f SCL		400	千赫	SCL频率.
t SCLH	0.6		微秒	SCL高.
t SCLL	1.3		微秒	SCL低.
t SCS	0.6		微秒	SCL建立时间, 与重复启动条件相关.
SCH	0.6		微秒	SCL保持时间.在此期间之后, 生成第一个时钟.
t DS	100		NS	数据建立时间.
t SCR		300	NS	SCL上涨时间.
t SCF		300	NS	SCL下降时间.
t SDR		300	NS	SDA上升时间.
t SDF		300	NS	SDA下降时间.
t BFT	0.6			免费巴士时间;停止和开始之间的时间.
多重引脚和复位				
GRT		50	NS	GPIO上升时间.
t GFT		50	NS	GPIO下降时间.
GIL		1.5×1 / fS	微秒	GPIO输入延迟;核心读取高/低值的时间.
t RLPW	20		NS	RESET低脉冲宽度.

1 所有的时序规格都是针对 串行输入端口和串行输出端口的默认 (I2S) 状态给出的 (参见表66) .

数字时序图

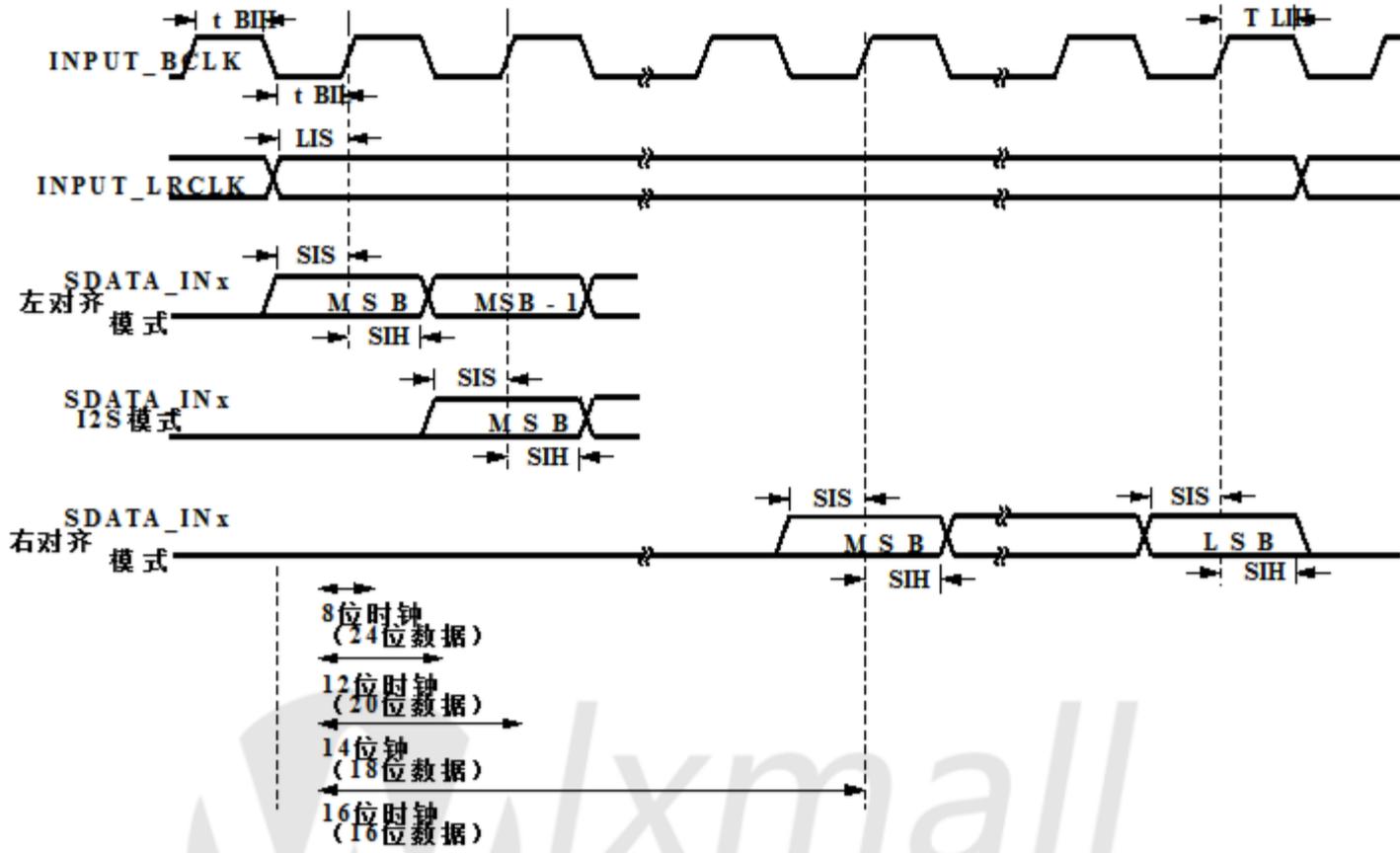


图2. 串行输入端口时序

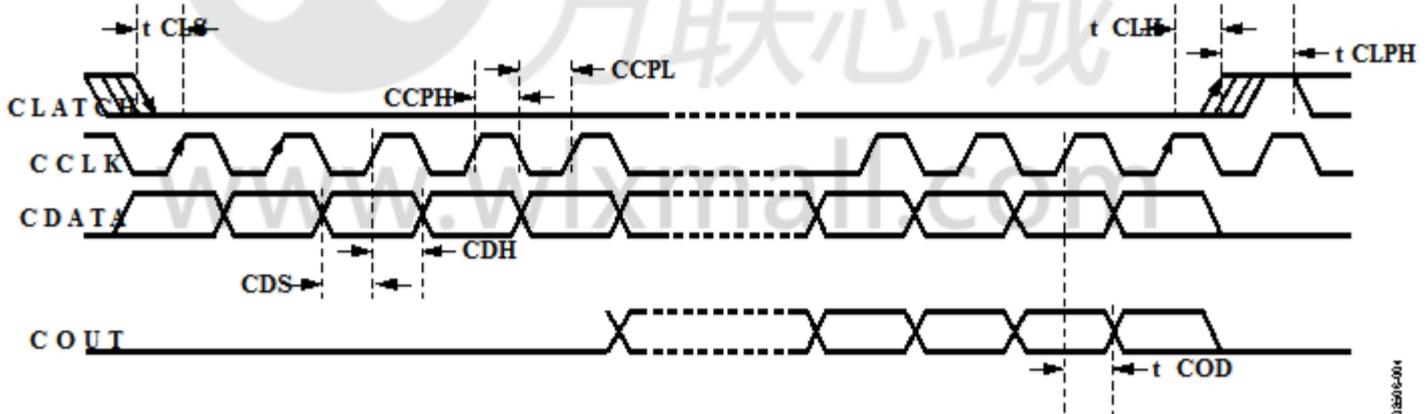


图3. SPI端口时序

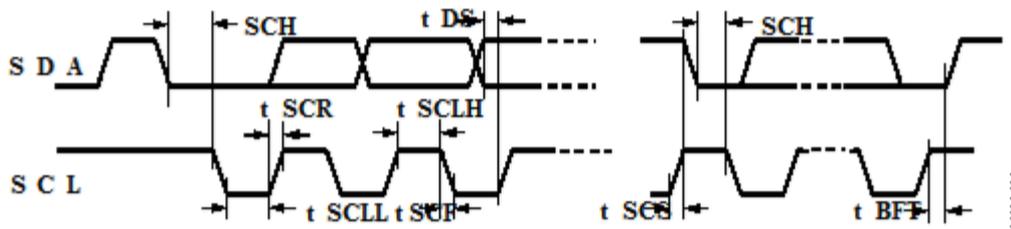


图4. I2C端口时序

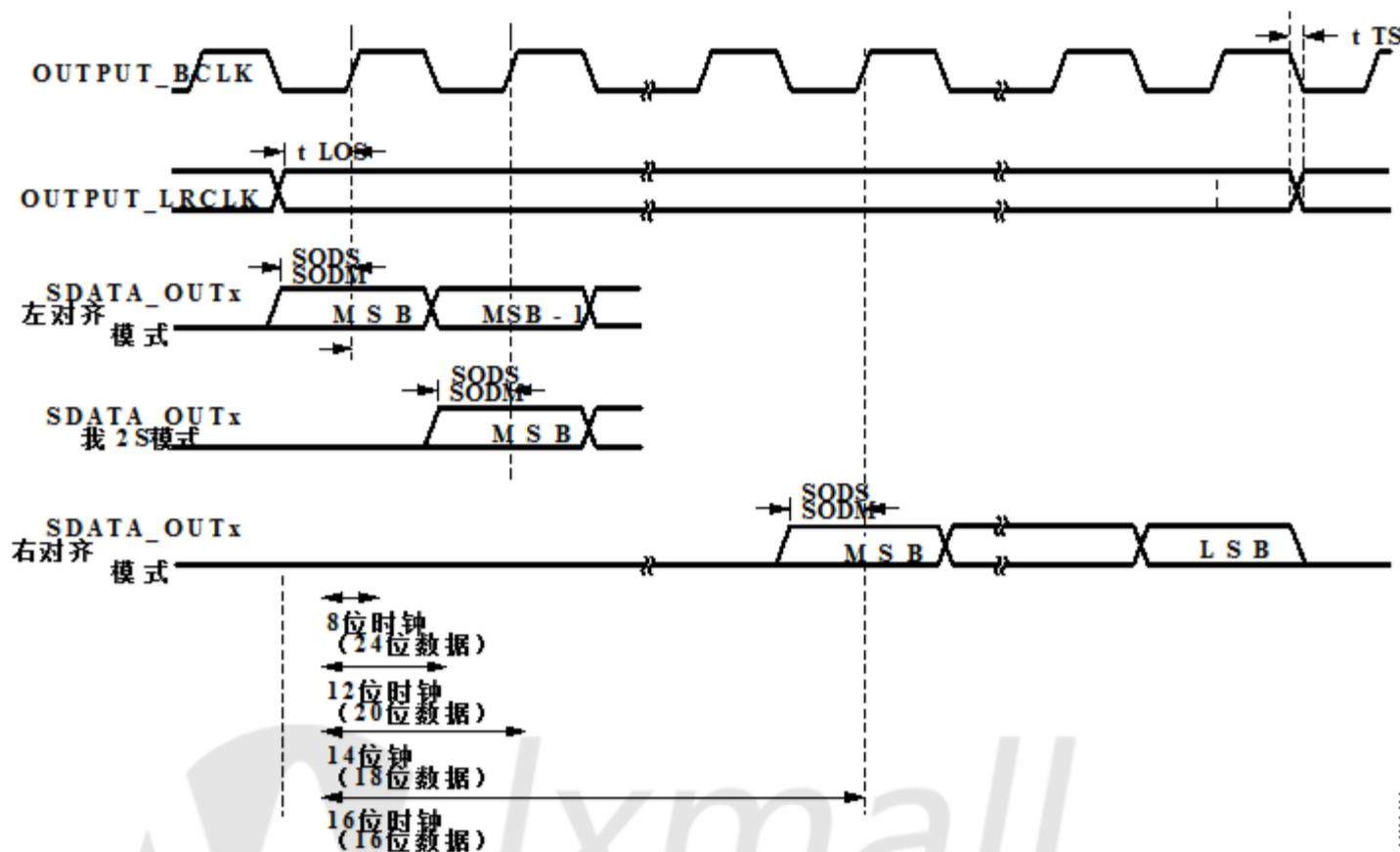


图5.串行输出端口时序

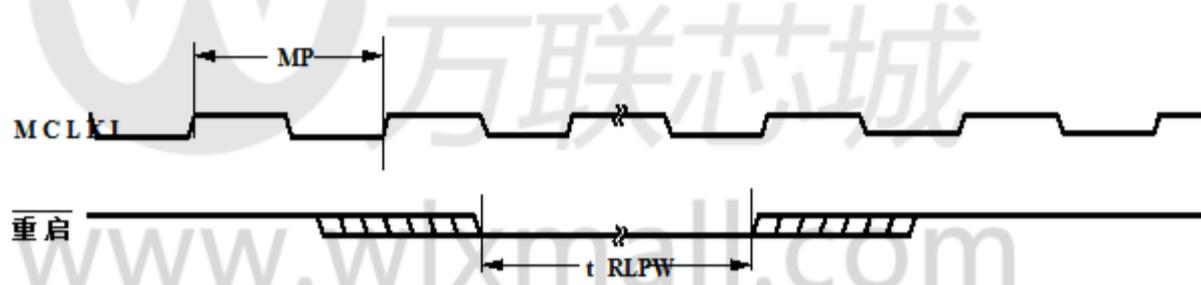


图6.主时钟和复位时序

绝对最大额定值

表9

参数	评分
DVDD到地面	0 V至2.2 V
AVDD接地	0 V至4.0 V
IOVDD接地	0 V至4.0 V
数字输入	DGND - 0.3 V, IOVDD + 0.3 V
最高结温	135°C
存储温度范围	-65°C至+ 150°C
焊接 (10秒)	300°C

强调超出绝对最大额定值列出的那些可能会导致设备永久性损坏.这是一个压力只有评级.在这些或任何设备的功能操作其他情况超出业务指标不是暗示本规范的一部分.接触绝对延长期限的最大额定条件可能会影响设备可靠性.

热阻

θ_{JA} 是针对最差条件 (即器件) 而规定的焊接在表面贴装封装的电路板上.

表10.热阻

包装类型	θ_{JA}	θ_{JC}	单元
48引脚LQFP	72	19.5	°C/W

ESD警告



ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.



引脚配置和功能说明

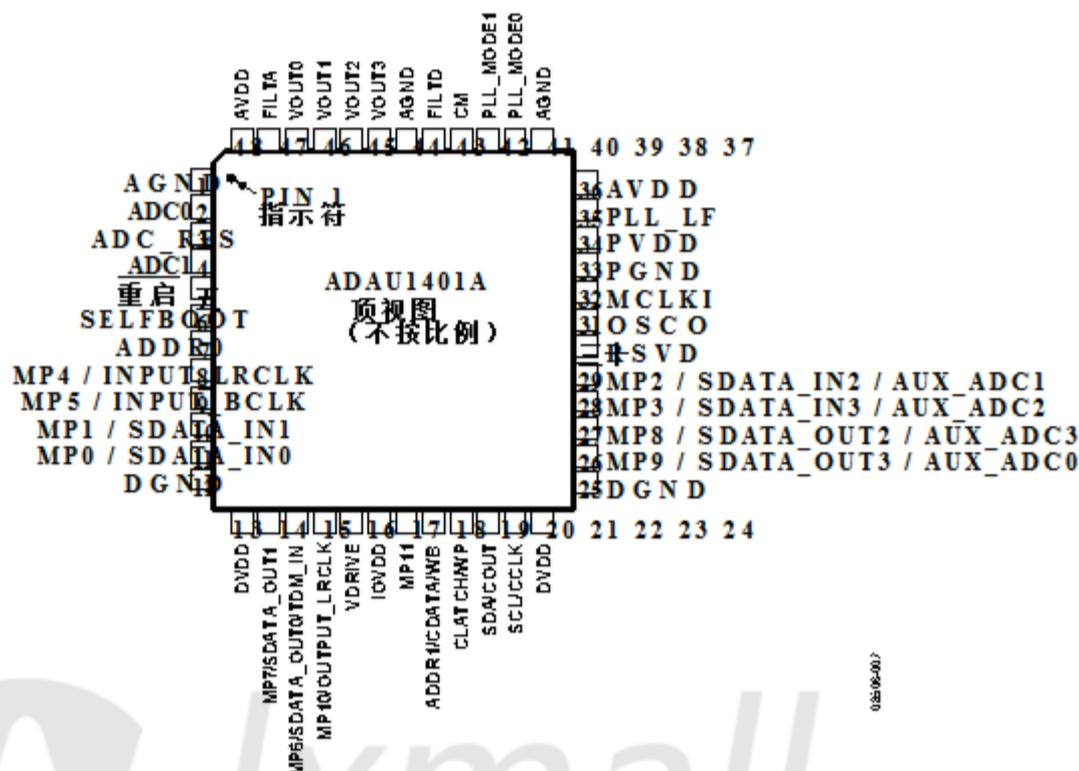


图7. 48引脚LQFP引脚配置

表11. 引脚功能描述

销号	助记符	类型1	描述
1,37,42	AGND	PWR	模拟地引脚。AGND, DGND和PGND引脚可以直接连接在一起共同的地面飞机。应使用100 nF电容将AGND解耦至AVDD引脚。
2	ADC0	A_IN	模拟音频输入0。满量程100μA均方根输入。电流输入允许输入电压电平可以通过一个外部电阻进行缩放。一个18kΩ的电阻将产生2 V rms满量程输入。有关详细信息, 请参阅音频ADCS部分。
3	ADC_RES	A_IN	ADC参考电流。ADC的满量程电流可以通过外部18kΩ来设置电阻连接在这个引脚和地面之间。有关详细信息, 请参阅音频ADCS部分。
4	ADC1	A_IN	模拟音频输入1。满量程100μA均方根输入。电流输入允许输入电压电平可以通过一个外部电阻进行缩放。一个18kΩ的电阻将产生2 V rms满量程输入。
五	重启	D_IN	低电平有效复位输入。复位在高电平至低电平时触发, ADAU1401A退出重置在低到高的边缘。有关初始化的更多信息, 请参阅加电序列部分的细节。
6	SELFBOOT	D_IN	启用/禁用自启动。SELFBOOT选择控制端口(低电平)或自启动(高电平)。设置当ADAU1401A退出复位状态时, 该引脚为高电平, 启动自引导操作。该引脚可以直接连接到控制电压, 也可以通过电阻上拉/下拉。看到自引导部分。
7	ADDR0	D_IN	I2C和SPI地址0。结合ADDR1, 该引脚最多允许四个ADAU1401A在同一个I2C总线上使用的器件, 或者最多两个与普通SPI一起使用的IC CLATCH信号。有关详细信息, 请参阅I2C端口部分。
8	MP4 / INPUT_LRCLK	D_IO	多用途GPIO / 串行输入端口LRCLK。有关更多信息, 请参阅多用途引脚部分细节。
9	MP5 / INPUT_BCLK	D_IO	多用途GPIO / 串行输入端口BCLK。有关更多详细信息, 请参阅多用途引脚部分。
10	MP1 / SDATA_IN1	D_IO	多用途GPIO / 串行输入端口数据1。有关更多信息, 请参见多用途引脚部分细节。
11	MP0 / SDATA_IN0	D_IO	多用途GPIO / 串行输入端口数据0。有关更多信息, 请参见多用途引脚部分细节。
12, 25	DGND	PWR	数字地引脚。AGND, DGND和PGND引脚可以直接连接在一起共同的地面飞机。应使用100 nF电容将DGND解耦至DVDD引脚。
13, 24	DVDD	PWR	1.8 V数字电源。该引脚的输入可以从外部提供或者生成

ADAU1401A

销号	助记符	类型1	描述
14	MP7 / SDATA_OUT1	D_IO	采用板载1.8 V稳压器的3.3 V电源供电. DVDD应解耦为DGND与一个100 nF电容器. 多用途GPIO /串行输出端口数据1.请参见多用途引脚部分更多细节.
15	MP6 / SDATA_OUT0 / TDM_IN	D_IO	多用途GPIO /串行输出端口数据0 / TDM数据输入.请参阅多用途引脚部分更多细节.
16	MP10 / OUTPUT_LRCLK	D_IO	多用途GPIO /串行输出端口LRCLK.请参阅多用途引脚部分更多细节.
17	VDRIVE	A_OUT	驱动1.8 V稳压器.稳压器外部PNP晶体管的基极是从VDRIVE驱动.有关详细信息, 请参阅电压调节器部分.
18	IOVDD	PWR	提供输入和输出引脚.该引脚上的电压设置了最高的输入电压.应该在数字输入引脚上看到.该引脚也是数字输出信号的电源.在控制端口和MPx引脚上, IOVDD应该始终设置为3.3 V.这是当前的情况.引脚是可变的, 因为它取决于数字输出的负载.
19	MP11	D_IO	多用途GPIO或串行输出端口BCLK (OUTPUT_BCLK).请参阅多用途引脚部分更多细节.
20	ADDR1 / CDATA / WBD_IN	D_IO	I2C地址1 / SPI数据输入 / EEPROM写回触发器. ADDR1与ADDR0结合使用设置IC的I2C地址, 以便在同一I2C上使用四个ADAU1401A器件总线 (详见I2C端口部分).有关此CDATA功能的更多信息.引脚, 请参阅SPI端口部分.上升 (默认) 或下降 (如果由EEPROM消息设置) 边沿在WB引脚上触发将接口寄存器写回到外部EEPROM.这个功能可用于在断电时保存参数数据 (参见“自引导”部分详情).
21	CLATCH / WP	D_IO	SPI锁存信号/自引导EEPROM写保护. CLATCH必须在一开始就低SPI交易, 在交易结束时高.每个SPI交易可以采取不同的CCLK引脚上的周期数完成, 具体取决于地址和读/写在SPI事务开始时发送 (详细信息请参见SPI端口部分).该当器件处于自引导模式时, WP引脚为开路集电极输出. ADAU1401A将WP拉低以启用对外部EEPROM的写操作.这个引脚应该拉高3.3 V (详情请参见自引导部分).
22	SDA / COUT	D_IO	I2C数据 / SPI数据输出. SDA是一个双向开路集电极.该线路连接到SDA应该有一个2.2kΩ的上拉电阻 (详见I2C端口部分). COUT用于读回寄存器和内存位置.当一个SPI读取不是时, 它是三态的活动 (详见SPI端口部分).
23	SCL / CCLK	D_IO	I2C时钟 / SPI时钟.在I2C控制模式下, SCL始终为集电极开路输入.在自启动模式下, SCL是一个集电极开路输出 (I2C主控).连接到SCL的线路应该有一个2.2kΩ的上拉电阻 (详见I2C端口部分). CCLK也可以连续运行或在SPI事务之间关闭 (详见SPI端口部分).
26	MP9 / SDATA_OUT3 / AUX_ADC0	D_IO / A_	多用途GPIO /串行输出端口数据3 /辅助ADC输入0.参见多用途引脚部分的更多细节.
27	MP8 / SDATA_OUT2 / AUX_ADC3	D_IO / A_	多用途GPIO /串行输出端口数据2 /辅助ADC输入3.参见多用途引脚部分的更多细节.
28	MP3 / SDATA_IN3 / AUX_ADC2	D_IO / A_	多用途GPIO /串行输入端口数据3 /辅助ADC输入2.参见多用途引脚部分的更多细节.
29	MP2 / SDATA_IN2 / AUX_ADC1	D_IO / A_	多用途GPIO /串行输入端口数据2 /辅助ADC输入1.参见多用途引脚部分的更多细节.
三十	RSVD		保留.将该引脚直接或通过下拉电阻接地.
31	OSCO	D_OUT	晶体振荡器电路输出.两者之间应连接一个100Ω阻尼电阻.这个引脚和晶体.这个输出不应该被用来直接驱动一个时钟到另一个.我知道了.如果不使用晶体振荡器, 则该引脚可以悬空.请参阅使用振荡器部分的细节.
32	MCLKI	D_IN	主时钟输入.该引脚可以连接到3.3 V时钟信号或作为输入来自晶体振荡器电路.有关详细信息, 请参见设置主时钟/ PLL模式部分.
33	PGND	PWR	PLL接地引脚. AGND, DGND和PGND引脚可以直接连接在一起.共同的地面飞机.应使用100 nF电容将PGND解耦到PVDD.
34	PVDD	PWR	用于PLL和辅助ADC模拟部分的3.3 V电源.这个引脚应该是一个100 nF电容去耦到PGND.
35	PLL_LF	A_OUT	PLL环路滤波器连接.这个引脚必须连接两个电容和一个电阻如图15所示.有关更多详细信息, 请参见设置主时钟/ PLL模式部分.
36, 48	AVDD	PWR	3.3 V模拟电源.该引脚应通过100 nF电容去耦至AGND.

销号	助记符	类型 ¹	描述
38, 39	PLL_MODE0, PLL_MODE1	D_IN	PLL模式设置.这些引脚设置主时钟PLL的输出频率.看到了设置主时钟/PLL模式部分了解更多详情.
40	厘米	A_OUT	1.5 V共模参考.应该连接一个47 μ F去耦电容在此引脚和地之间,以减少ADC和DAC之间的串扰.材料电容器并不重要.只要这个引脚可以用来偏置外部模拟电路.那些电路没有从引脚吸收电流(例如当CM引脚连接到运算放大器的同相输入).
41	FILTD	A_OUT	DAC滤波器去耦引脚.此引脚和引脚之间应连接一个10 F电容地面.电容器材料并不重要.该引脚上的电压为1.5 V.
43	VOUT3	A_OUT	VOUT3 DAC输出.满量程输出电压为0.9 V rms.这个输出可以使用有源或无源输出重构滤波器.有关详细信息,请参阅音频DACS部分.
44	VOUT2	A_OUT	VOUT2 DAC输出.满量程输出电压为0.9 V rms.这个输出可以使用有源或无源输出重构滤波器.有关详细信息,请参阅音频DACS部分.
45	VOUT1	A_OUT	VOUT1 DAC输出.满量程输出电压为0.9 V rms.这个输出可以使用有源或无源输出重构滤波器.有关详细信息,请参阅音频DACS部分.
46	VOUT0	A_OUT	VOUT0 DAC输出.满量程输出电压为0.9 V rms.这个输出可以使用有源或无源输出重构滤波器.有关详细信息,请参阅音频DACS部分.
47	FILTA	A_OUT	ADC滤波器去耦引脚.此引脚和引脚之间应连接一个10 F电容地面.电容器材料并不重要.该引脚上的电压为1.5 V.

¹ PWR =电源/接地, A_IN =模拟输入, D_IN =数字输入, A_OUT =模拟输出, D_IO =数字输入/输出, D_IO / A_IO =数字输入/输出或模拟输入输出.



典型的性能特征

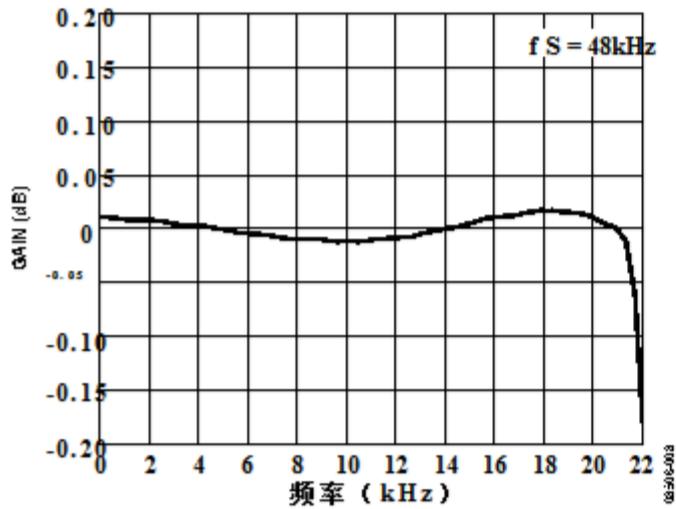


图8. ADC通带滤波器响应

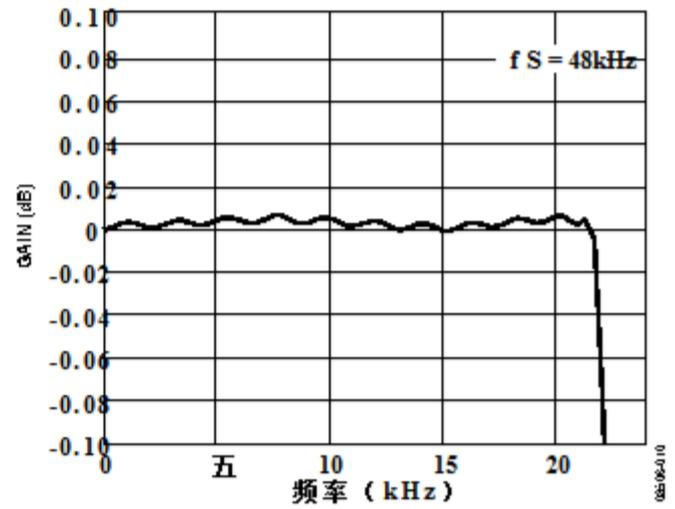


图10. DAC通带滤波器响应

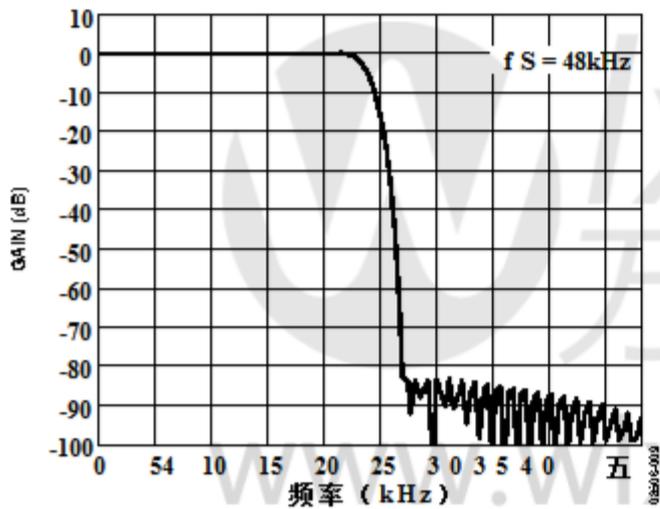


图9. ADC阻带滤波器响应

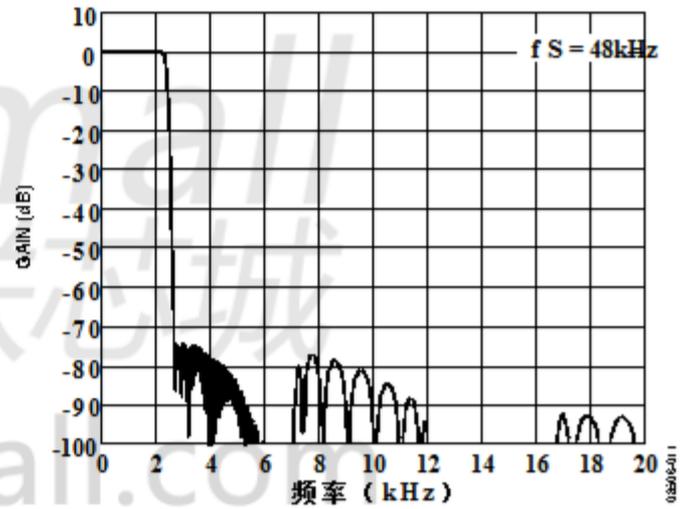


图11. DAC阻带滤波器响应

系统框图

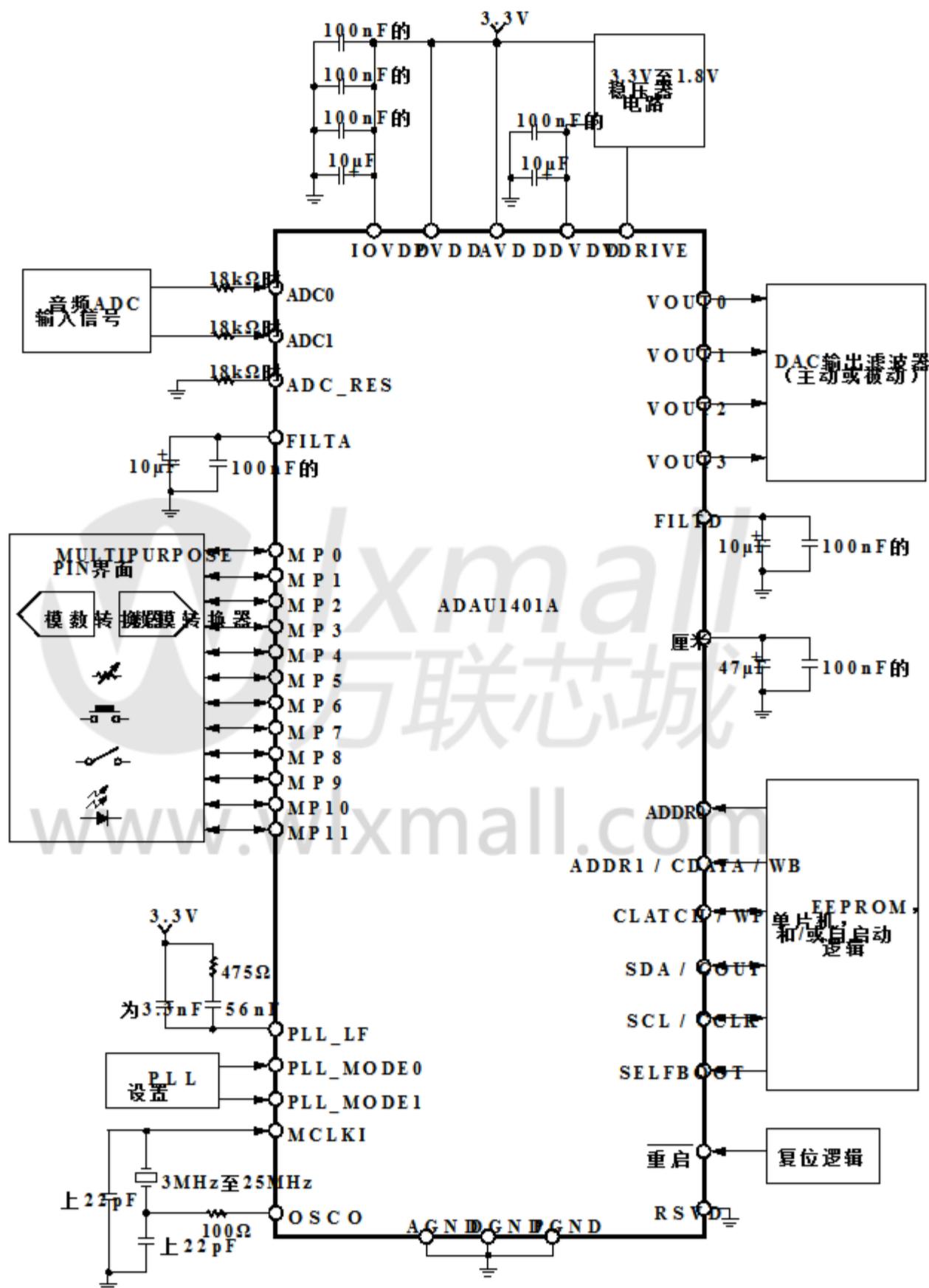


图12.系统框图

操作理论

ADAU1401A的核心是28位DSP（56位，精密处理），为音频处理而优化该程序和参数RAM可以用自定义加载音频处理信号流使用SigmaStudio图形构建ADI公司的编程软件

存储在参数RAM控制下的个人信号处理块，例如均衡滤波器，动态处理器，音频延迟和混音器级别。安全加载功能允许透明参数更新并防止输出信号中的点击。

程序RAM，参数RAM和寄存器内容可以保存在ADAU1401A的外部EEPROM中可以在启动时自行启动。在这个独立的模式下，参数可以通过板载多用途引脚进行控制。该ADAU1401A可以接受开关，电位器，旋转编码器和红外接收器。诸如音量之类的参数并可在掉电时将音调设置保存到EEPROM中并在加电时再次召回。

ADAU1401A可以使用数字或模拟输入和产出或两者的组合。立体声ADC和四个DAC具有至少+100dB的SNR和至少THD+N-83分贝。8通道，灵活的串行数据输入/输出端口允许无缝连接到各种ADC，DAC，通用DSP，S/PDIF接收器和发射器等采样率转换器。ADAU1401A的串行端口可以配置为I²S，左对齐，右对齐或TDM串行端口兼容模式。

ADAU1401A有12个多用途引脚（MP0至MP11）接收外部控制信号作为输入并输出标志或控制系统中的其他设备。MPx引脚可以配置为数字I/O，输入到4通道辅助ADC或串行数据I/O端口。作为输入，这些引脚可以连接到按钮，开关，旋转编码器，电位器，IR接收器或其他外部电路来控制内部信号处理程序。配置为输出时，这些引脚可用于驱动LED，控制其他IC或连接到应用程序中的其他外部电路。

ADAU1401A具有支持的复杂控制端口完成所有存储位置的读/写功能。控制提供寄存器来提供对配置的完全控制，芯片的配置和串行模式。ADAU1401A可以配置为SPI或I²C控制，或者可以自行启动一个外部EEPROM。

一个板上振荡器可以连接到一个外部晶体生成主时钟。另外，主时钟相位 -

锁相环（PLL）允许从ADAU1401A提供时钟各种时钟速度。PLL可以接受64×f_S，256×f_S，384×f_S或512×f_S来产生核心的内部主时钟。

SigmaStudio软件用于编程和控制SigmaDSP®通过控制端口。随着设计和调谐信号流，可以使用SigmaStudio工具进行配置所有的DSP寄存器和烧录新的程序到外部EEPROM。SigmaStudio图形界面允许任何人用数字或模拟音频处理知识来轻松设计一个DSP信号流并将其连接到目标应用程序。此外，该接口提供了足够的灵活性和可编程性为有经验的DSP程序员提供深入的控制的设计。在SigmaStudio中，用户可以连接图形块（比如双二阶滤波器，动态处理器，混频器，延迟），编译设计，并加载程序和参数文件通过控制端口插入ADAU1401A内存。提供的库中提供的信号处理块包括：

- 单和双精度双二阶滤波器
- 具有单通道峰值或均方根检测的处理器和多声道动态
- 混频器和分配器
- 音调和噪音发生器
- 固定和可变增益
- 响度
- 延迟
- 立体声增强
- 动态低音增强
- 噪音和音源
- FIR滤波器
- 液位探测器
- GPIO控制和调节

额外的处理块总是在开发中。

ADI公司还提供专有和第三方用于矩阵解码，低音等应用的算法增强和环绕虚拟器。联系模拟有关授权这些算法的信息的设备。

ADAU1401A采用1.8 V数字电源供电和3.3 V模拟电源。板载稳压器可以用于从3.3 V单电源供电。该ADAU1401A采用单片集成制造电路，并封装在一个48引脚LQFP操作通过-40°C至+105°C的温度范围。

初始化

本节介绍正确设置的步骤

ADAU1401A.以下五步序列提供了一个如何初始化IC的概述:

1. 给ADAU1401A通电.
2. 等待PLL锁定.
3. 加载SigmaDSP程序和参数.
4. 设置寄存器(包括多用途引脚和数字接口).
5. 关闭转换器的默认静音,清除数据寄存器,并初始化DAC设置寄存器(见控制寄存器设置部分的具体设置).

要仅测试模拟音频直通(ADC到DAC),跳过步骤3和步骤4,并使用默认的内部程序.

上电顺序

ADAU1401A具有初始化的内置上电序列.所有内部RAM的内容在上电或当设备时被带出一个重置.在RESET的上升沿,内容内部程序引导ROM被复制到内部程序RAM存储器,参数RAM被填充值(全0)从其相关的启动ROM,并且所有寄存器都是初始化为0.默认引导ROM程序复制音频从输入到输出而不处理(参见数字13).在这个程序中,输出SDATA_IN0和SDATA_IN1在DAC0和DAC1以及SDATA_OUT0和SDATA_OUT1. ADC0和ADC1在DAC2和DAC3上输出DAC3.数据存储器在上电时也被清零.新值不应写入控制端口,直到初始化完成.

表12.加电时间

MCLKI输入频率	在里面时间	最大程序/参数/注册启动时间(I2C)	总时间
3.072MHz (64×f _S)	85毫秒	175毫秒	260毫秒
11.2896MHz (256×f _S)	83毫秒	175毫秒	198毫秒
12.288 MHz (256×f _S)	80毫秒	175毫秒	196毫秒
18.432 MHz (384×f _S)	80毫秒	175毫秒	191毫秒
24.576MHz (512×f _S)	81毫秒	175毫秒	186毫秒

PLL启动时间持续218个周期的时钟

MCLKI引脚.这个时间范围从24.576MHz的10.7ms (512×f_S)输入时钟为85.3ms,用于3.072MHz (64×f_S)输入

时钟,并从RESET的上升沿开始测量.以下

PLL启动, ADAU1401A引导周期的持续时间是对于48kHz的f_S约为42μs.用户应该避免写入或在此启动时间内从ADAU1401A读取.为

MCLKI 12.288 MHz的输入信号,完整的初始化序列(PLL启动加启动周期)大约为21ms.作为设备

出来一个复位,时钟模式是立即设置的

PLL_MODE0和PLL_MODE1引脚.重置是同步的到内部时钟的下降沿.

表12列出了将ADAU1401A引导至引脚的典型时间.假定400 kHz I2C,应用的运行状态.时钟加载完整的程序,参数集和所有寄存器(约8.5 kB).实际上,大多数应用程序不会填充RAM.因此,启动时间小于第3列中列出的值的表12.

控制寄存器设置

以下寄存器必须按照本节所述进行设置.初始化ADAU1401A.这些设置是基本的使用模拟器来操作IC所需的最小设置.输入/输出48 kHz.更多的寄存器可能需要设置,取决于应用程序.看到RAM和寄存器部分进行其他设置.

DSP内核控制寄存器(地址2076)

将位[4: 2](ADM, DAM和CR)分别设置为111.

DAC设置寄存器(地址2087)

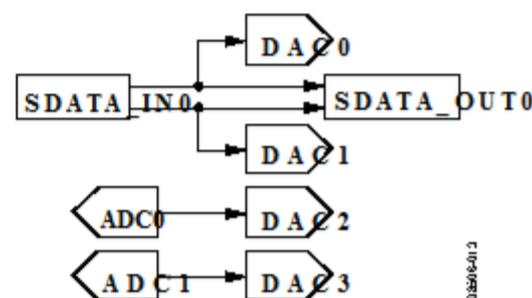
将位[1: 0](DS[1: 0])设置为01.

推荐的计划/参数

加载程序

当向程序写入大量的数据或者写入数据时,在直写模式下,内核应该是被禁用,以防止出现不愉快的噪声.音频输出.要禁用处理器核心,

1. 设置DSP内核控制寄存器的位[4: 3](低电平有效)(地址2076)设置为1,将ADC和DAC静音.这个开始一个音量减少.
2. 将DSP内核控制寄存器的位2(低电平有效)设置为1.这将SigmaDSP累加器的数据输出清零.寄存器和数据输入寄存器.
3. 使用突发模式写入填充程序RAM.
4. 使用突发模式写入填充参数RAM.
5. 将DSP内核控制寄存器的位[4: 2]设置为111.



功率降低模式

ADAU1401A芯片的部分可以打开和关闭.需要降低功耗.这些包括ADC, DAC和电压参考.

可以通过写入关闭各个模拟部分.辅助ADC和功率控制寄存器(地址2082).默认情况下,ADC, DAC和参考使能(所有位

ADAU1401A

设置为0)。每一个都可以通过写1来关闭该寄存器中的相应位。ADC掉电模式
关闭两个ADC，每个DAC都可以关闭个别。ADC的节省电流约为15 mA
断电，每个DAC供电约4 mA
下。提供给两个ADC的参考电压
和DAC只能在所有ADC和DAC都关闭的情况下进行断电。参考电压被掉电
设置辅助ADC和电源的位6和位7
控制寄存器。

使用振荡器

ADAU1401A可以使用板载振荡器来产生其信号主钟。该振荡器设计用于256×f_S
主时钟，对于48kHz的f_S是12.288MHz
对于44.1kHz的af_S为11.2896MHz。振荡器中的晶体电路应该是在其工作的AT切，并联谐振器基频。图14显示了外部电路
建议正确操作。

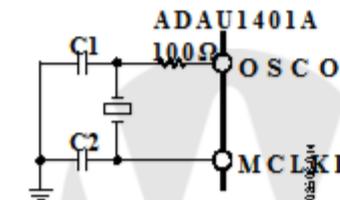


图14. 晶体振荡器电路

OSC O上的100Ω阻尼电阻给振荡器a
电压摆幅约为2.2 V。晶体并联电容
应该是7 pF。其负载电容应该是大约18 pF，
尽管该电路支持高达25 pF的值。必要
C1和C2负载电容的值可以从中计算
晶体负载电容如下：

$$C_{\text{大号}} = \frac{C1 \times C2}{C1 + C2} + C_{\text{流浪}}$$

其中C STRAY是电路中的杂散电容，通常是
假定为大约2 pF到5 pF。

不应该使用OSC O直接驱动晶振信号
另一个IC。这个信号是一个模拟正弦波，
开始使用它来驱动数字输入。有两个选项
使用ADAU1401A为其他IC提供主时钟
系统。第一种，也是较少推荐的方法是使用a
OSC O信号上的高阻抗输入数字缓冲器。如果这
的方法被使用，最小化到缓冲器输入的走线长度。
第二种方法是使用串行输出端口的时钟。
引脚19 (MP11) 可以设置为输出 (主) 时钟分频
从内部的核心时钟。如果此引脚设置为串行输出端口
(OUTPUT_BCLK) 模式在多功能引脚配置
寄存器 (地址2081)，端口被设置为串行主机
输出控制寄存器 (地址2078)，所需的输出fre-
也可以在串行输出控制寄存器中设置频率
OBF [1: 0]位 (见表49)。

如果在设计中没有使用振荡器，并且使用系统主时钟
已经在系统中可用，振荡器可以通电
以节省电力。默认情况下，振荡器开机。该
当1被写入OPD位时，振荡器掉电
振荡器掉电寄存器 (地址2086;见表60)。

设置主时钟/ PLL模式

ADAU1401A的MCLKI输入提供一个PLL
生成50 MIPS的SigmaDSP内核时钟。在正常的操作中，
MCLKI的输入必须是以下之一：64×f_S，256×f_S，
384×f_S或512×f_S，其中f_S是输入采样率。模式
如上所述通过配置PLL_MODE0和PLL_MODE1来设置
如表13所示。如果ADAU1401A设置为接收双速率信号
(通过将每个样本的程序步数减少一个因子
2使用核心控制寄存器)，主时钟频率
必须是32×f_S，128×f_S，192×f_S或256×f_S。如果ADAU1401A
被设置为接收四倍频信号 (通过减少数量)
使用DSP内核，每个采样的程序步长减少4倍
控制寄存器)，主时钟频率必须是16×f_S，
64×f_S，96×f_S或128×f_S。上电时，必须有一个时钟信号
出现在MCLKI引脚上，以便ADAU1401A可以完成
其初始化程序。

表13. PLL模式

MCLKI输入 频率	PLL_MODE0	PLL_MODE1
64×f _S	0	0
256×f _S	0	1
384×f _S	1	0
512×f _S	1	1

时钟模式不应该改变，也不能复位
ADAU1401A。如果在操作过程中模式发生变化，
点击或弹出可能会导致输出信号的状态
当RESET保持低电平时，应该改变PLL_MODEx引脚。

PLL环路滤波器应连接到PLL_LF引脚。这个
如图15所示，滤波器包括三个无源元件 -
两个电容器和一个电阻器。这些组件的值
不需要确切。宽容可以高达10%的
电阻器和高达20%的电容器。3.3 V信号显示在
图15可以连接到芯片的AVDD电源。

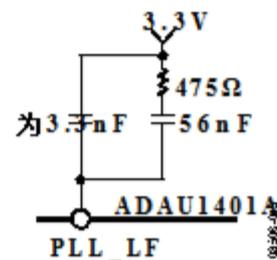


图15. PLL环路滤波器

电压调节器

ADAU1401A的数字电压必须设置为1.8 V.

芯片包括一个板载电压调节器，

器件用于没有1.8 V电源的系统

但提供3.3 V电源.唯一的外部组件

在这种情况下需要一个PNP晶体管，一个电阻和一个
很少有旁路电容.只有一个引脚，VDRIVE，是必要的
支持监管机构.

显示了电压调节器的推荐设计

见图16.图中显示了10 μ F和100 nF电容

配置建议绕过，但不是

操作所必需的.每个DVDD引脚应该有自己的

100 nF的旁路电容，但只有一个大容量电容（10 μ F到
两个DVDD引脚都需要47 μ F）.有了这个配置，

3.3 V是主系统电压; 1.8 V产生在

晶体管的集电极，它连接到DVDD引脚.

VDRIVE连接到PNP晶体管的基极.如果

调节器不用于设计，VDRIVE可以接地.

在选择规范时必须考虑两个规范，

晶体管：晶体管的电流放大系数

（ h_{FE} 或 β ）应至少为100，晶体管的集电极

必须能够消散调节时产生的热量

从3.3 V到1.8 V.从最大数字电流汲取

ADAU1401A是40 mA.等式来确定

晶体管的最小功耗如下：

$$(3.3V-1.8V) \times 60mA = 90mW$$

有许多这些规格可用的晶体管

小型SOT-23或SOT-223封装.

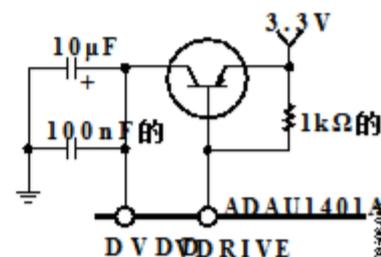


图16.电压调节器配置



音频ADC

ADAU1401A具有两个Σ-Δ型ADC.信噪比(SNR)为100 dB, THD + N为-83 dB.

立体声音频ADC是电流输入;因此,在输入端需要使用直流电阻.这意味着系统输入信号的电压电平可以设置为任何级别;只有输入电阻需要缩放来提供适当的满量程电流输入. ADC0和ADC1输入引脚以及ADC_RES引脚都有一个2kΩ的内部电阻用于ESD保护. ADC输入上直接看到的电压引脚是1.5 V共模电压.

连接到ADC_RES的外部电阻设置满量程ADC的当前输入.全部的ADC输入是ADC_RES (20kΩ) 上带有外部18 k电阻的100μArms总,因为它与内部2kΩ串联).唯一的如果改变ADC_RES电阻的原因是采样率使用48kHz以外的频率.

连接到ADC0和ADC1的电压-电流电阻已设置ADC的满量程电压输入.具有全面的电流输入100μArms,外部18kΩ的2.0 V rms信号电阻(与2kΩ的内部电阻串联)导致一个使用全范围的ADC输入.这些的匹配ADC_RES电阻器的电阻对于操作非常重要ADC.对于这三个电阻,推荐使用1%的容差.

ADC0输入引脚和/或ADC1输入引脚可以留下.如果ADC的相应通道未被使用,则不连接.

电阻值的计算假定采样率为48 kHz.推荐的输入和电流设置电阻线性缩放与采样率相关,因为ADC具有开关电容输入.总值(2kΩ内部加外部电阻)的采样率为f_{S_NEW}的ADC_RES电阻可以计算为如下:

$$R_{总} = 20 \text{ 千欧} \times \frac{48,000}{f_{小新}}$$

电阻器的值(内部加外部)串联ADC0和ADC1的引脚可以计算如下:

$$R_{INPUT总} = \text{RMS输入电压} \times 10 \text{ 千欧} \times \frac{48,000}{f_{小新}}$$

表14列出了常用的外部和总电阻值.信号输入电平为48 kHz采样率.一个全面的有效值输入电压为0.9 V,因为满量程显示在表中.此输入电平的信号等于DAC上的满量程输出.

表14. ADC输入电阻值

满量程 RMS 输入电压 (V)	ADC_RES 数值 (kΩ)	ADC0 / ADC1 电阻数值 (kΩ)	总ADC0 / ADC1 输入电阻 (外部+内部) (kΩ)
0.9	18	7	9
1.0	18	8	10
2.0	18	18	20

图17显示了ADC输入的典型配置.对于48 kHz的f_S, 2.0 V rms输入信号.47μF电容器用于交流耦合信号,使输入偏置在1.5 V.

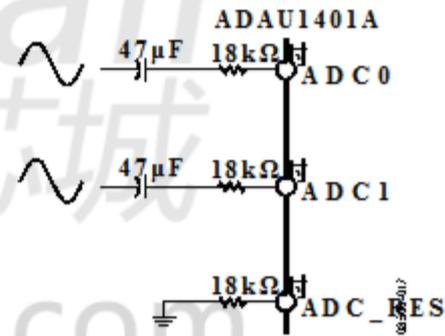


图17. 音频ADC输入配置

音频DAC

ADAU1401A包含四个 Σ - Δ DAC.信噪比DAC为104 dB, THD + N为-90 dB.全面的DAC上的输出为0.9 V rms (2.5 V pp) .

DAC处于反相配置.如果信号颠倒

从输入到输出是不可取的,也可以通过颠倒使用输出滤波器的反相配置或简单地使用反转SigmaDSP程序流程中的信号.

DAC输出可以用主动或被动滤波

重建滤波器.一个单极,被动,低通滤波器

如图18所示,50 kHz转角频率就足够了

滤除DAC的带外噪声,尽管有源滤波器也可以提供更好的音频性能.图19显示了一个三极,

积极的低通滤波器,提供更陡峭的滚降和更好的阻带衰减比无源滤波器.在这个配置中,AD8606运放的V+和V-引脚被设置为VDD和地面,分别.

要正确初始化DAC,DAC中的DS[1:0]位设置寄存器(地址2087)应设置为01.

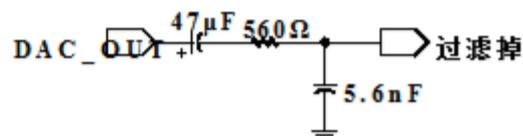


图18.无源DAC输出滤波器

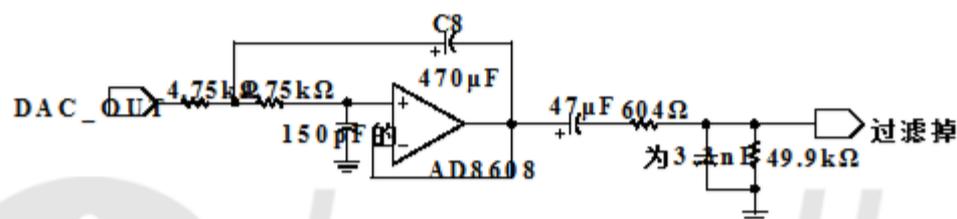


图19.有源DAC输出滤波器

www.wxmall.com

控制端口

ADAU1401A可以以三种控制模式之一运行：
I²C控制，SPI控制或自启动（无外部控制器）。

ADAU1401A具有一个4线SPI控制端口和一个2线I²C总线控制端口。两个端口都可以用来设置RAM和寄存器。当SELFBOOT引脚处于低电平时，该部分默认为I²C模式，但可以放入SPI控制模式，将CLATCH / WP引脚拉低三次。什么时候在上电时，SELFBOOT引脚设置为高电平，ADAU1401A加载它的程序，参数和寄存器设置外部EEPROM启动。

控制端口可以进行全部读/写操作。可寻址的存储器位置和寄存器。大部分信号处理通过将新值写入参数来控制参数。使用控制端口的RAM。其他功能，如静音和输入/输出模式控制，通过写入来编程寄存器。

所有的地址都可以在单一地址模式或者一个地址模式下访问。突发模式。控制端口的第一个字节（字节0）写入包含7位芯片地址和R / W位。接下来的两个字节（字节1和字节2）一起构成了子地址。存储器或ADAU1401A内的寄存器位置。这个子地址必须是两个字节，因为内存位置在ADAU1401A内可直接寻址及其大小。

超出了单字节寻址的范围。所有后续的字节（从字节3开始）包含数据，如控制端口数据，程序数据或参数数据。每个字的字节数取决于正在写入的数据的类型。确切的格式对于特定类型的写入显示在表22至表31。

ADAU1401A有多种更新信号的机制。实时处理参数而不会导致爆音或点击。如果需要下载大块数据，则输出的DSP内核可以暂停（使用DSP内核的CR位）控制寄存器（地址2076），可以加载新的数据。那么设备可以重新启动。这通常在此期间完成启动时或加载新程序时的启动顺序进入RAM。在只有少数参数需要的情况下改变了，他们可以在不停止程序的情况下加载。至避免不必要的副作用，同时加载参数，SigmaDSP提供了safeload寄存器。安全寄存器可以用来缓冲一整套参数（for例如，一个biquad的五个系数）然后转移这些参数放入一个音频内的活动节目中。帧。安全加载模式使用内部逻辑来防止DSP内核与控制端口之间的争用。

控制端口引脚是多功能的，取决于模式，其中的部分正在运行。表15详细说明了这些多种功能。

表15.控制端口引脚和SELFBOOT引脚功能

销	I ² C模式	SPI模式	自启动
SCL / CCLK	SCL-输入	CCLK输入	SCL-输出
SDA / COUT	SDA开路集电极输出和输入	COUT输出	SDA开路集电极输出和输入
ADDR1 / CDATA / W	ADDR1输入	CDATA输入	WB回写触发器
CLATCH / WP	未使用的输入连接到地或IOVDD	CLATCH输入	WP-EEPROM写保护，集电极开路输出
ADDR0	ADDR0输入	ADDR0输入	未使用的输入连接到地或IOVDD

我 I2C 端口

ADAU1401A 支持 2 线串行 (I2C 兼容) 微处理器总线驱动多个外设。两个引脚 - 串行数据 (SDA) 和串行时钟 (SCL) - 信息在 ADAU1401A 和系统 I2C 主控制器之间。在 I2C 模式下, ADAU1401A 始终是总线上的从机, 这意味着它不能启动数据传输。每个从设备是由一个独特的地址识别。地址字节格式是如表 16 所示。ADAU1401A 从地址已设置与 ADDR0 和 ADDR1 引脚。地址驻留在 I2C 写入的前七位。该字节的 LSB 设置为读或写操作。逻辑 1 级对应于读取操作, 并且逻辑电平 0 对应于写入操作。地址的位 5 和位 6 通过绑定 ADDR_x 引脚来设置 ADAU1401A 为逻辑电平 0 或逻辑电平 1。全字节地址, 包括引脚设置和读/写 (R/W) 位, 被显示在表 17。

突发模式寻址, 其中子地址是自动创建的, 在词边界上递增, 可以用于书写大量的数据到连续的内存位置。这个单字写入后增量自动发生, 除非遇到停止状况。中的寄存器和 RAM ADAU1401A 的宽度范围从 1 到 5 个字节。因此, 自动递增功能知道子地址之间的映射和目标寄存器 (或存储器的字长) 位置。数据传输始终由停止条件终止。

SDA 和 SCL 都应该有 2.2kΩ 的上拉电阻线连接到他们。这些信号线上的电压应该是不要超过 IOVDD (3.3 V)。

表 16. ADAU1401A I2C 地址字节格式

位 0	位 1	位 2	位 3	位 4	位 5	位 6	位 7
0	1	1	0	1	ADDR1	ADDR0	R / W

表 17. ADAU1401A I2C 地址

ADDR1	ADDR0	R / W	从站地址
0	0	0	0x68
0	0	1	0x69
0	1	0	的 0x6A
0	1	1	0x6B
1	0	0	0x6C
1	0	1	0x6D
1	1	0	0x6E
1	1	1	0x6F

解决

最初, I2C 总线上的每个设备都处于空闲状态监视 SDA 和 SCL 线路的起始条件和正确的地址。I2C 主机通过建立一个启动来启动数据传输条件, 由 SDA 上的高到低转换定义 SCL 保持高位。这表明一个地址或地址和一个数据流。总线上的所有设备都响应开始条件并移位下一个 8 位 (7 位地址加上 R/W 位) MSB 先。识别的设备发送地址通过拉低数据线来响应。在第九个时钟脉冲期间, 这第九位被称为一个确认位。所有其他设备退出巴士这一点, 并返回到闲置状态。R/W 位确定数据的方向。一个 LSB 上的逻辑 0 第一个字节意味着主人写信息给外设, 而逻辑 1 表示主读取。在写入子地址之后从外设发送信息并重复开始地址。数据传输直到遇到停止条件。停止条件发生时当 SCL 保持高电平时, SDA 从低电平转换到高电平。图 20 显示了 I2C 写入的时间, 图 21 显示了 I2C 阅读。

停止和启动条件可以在任何阶段检测到数据传输。如果这些条件是断言的顺序正常的读写操作, ADAU1401A 立即跳转到空闲状态。在给定的 SCL 期间高期间, 用户应该只发出一个启动条件, 一个停止状态, 或一个单一的停止条件开始条件。如果用户发出无效的子地址, ADAU1401A 不发出确认并返回闲置状态。如果用户超过最高子地址而在自动增量模式下, 则采取两个动作之一。在读模式, ADAU1401A 输出最高的子地址注册内容, 直到主设备发出不应答, 指示读取的结束。一个没有承认的条件是在第九个时钟脉冲 SDA 线没有拉低在 SCL 上。另一方面, 如果最高的子地址位置是在写入模式下达到, 无效字节的数据不是加载到任何子地址寄存器中, 将发出无应答通过 ADAU1401A, 部分返回到空闲状态。

ADAU1401A

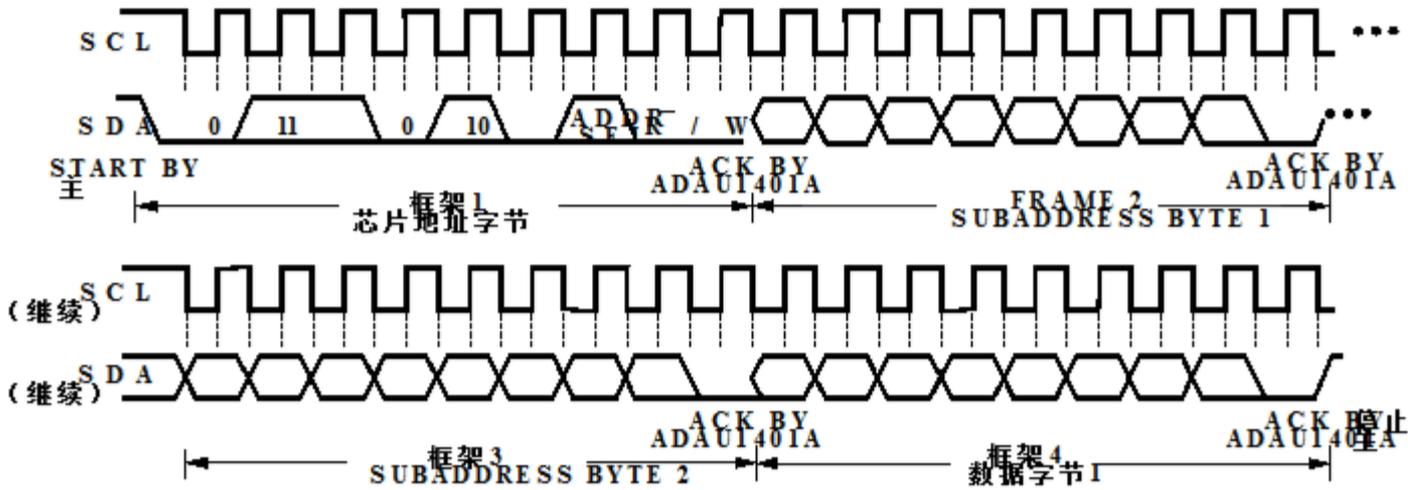


图20. I2C写入ADAU1401A时钟

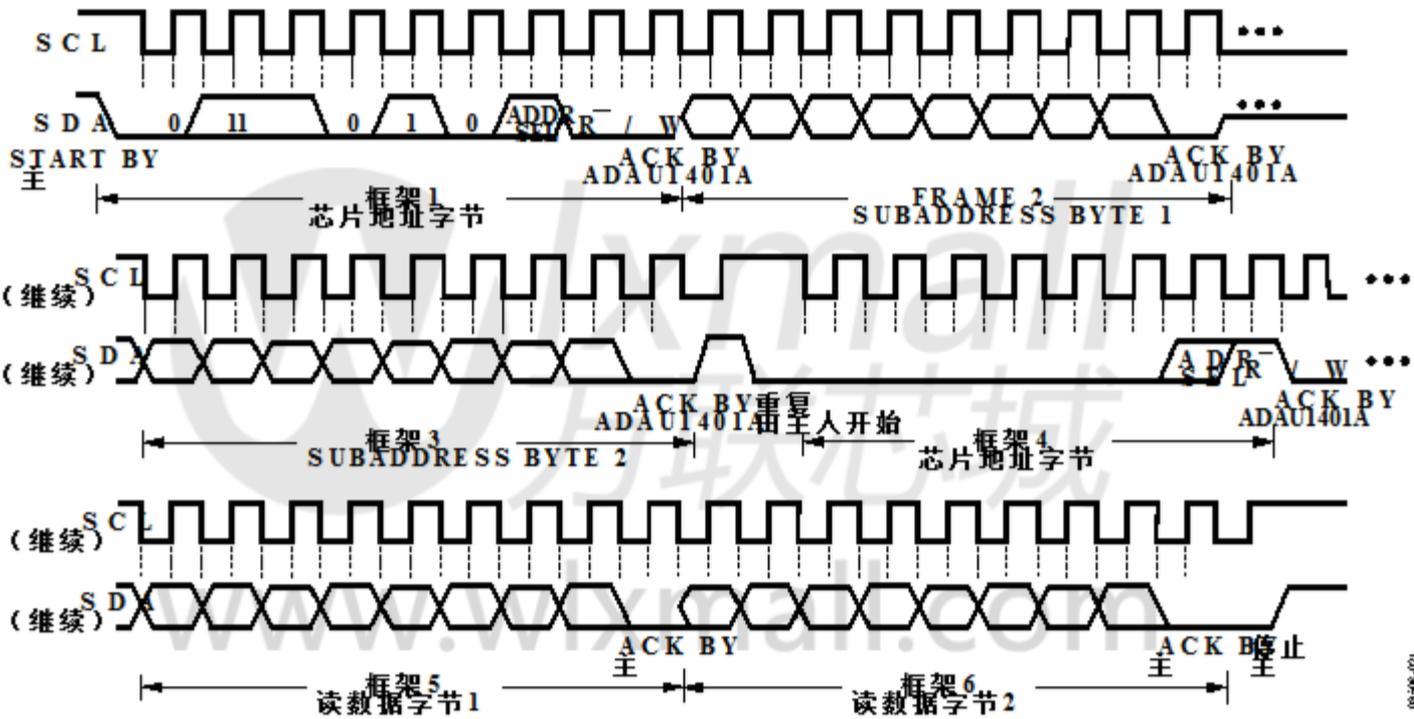


图21. I2C从ADAU1401A时钟读取

I2C读写操作

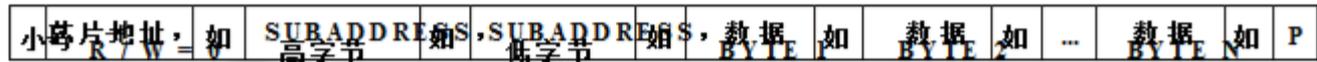
图22显示了单字写操作的时序。上每隔第九个时钟，ADAU1401A就会发出一个确认拉低SDA。

图23显示了突发模式写入序列的时序。该图显示了目标目标的示例寄存器是两个字节。ADAU1401A知道增量它的子地址寄存器每两个字节因为要求子地址对应一个寄存器或存储区域两个字节的字长。

单字读取操作的时序如图所示。图24。注意第一个R/W位是0，表示写入操作。这是因为子地址仍然需要写成设置内部地址。在ADAU1401A之后承认收到子地址，主人必须

发出一个重复的启动命令，然后是芯片地址R/W位设置为1（读）。这导致ADAU1401A SDA反转并开始将数据传回给主站。该主人然后回应每9个脉冲与承认脉冲到ADAU1401A。

图25显示了突发模式读取序列的时序。这个图中显示了一个目标读取寄存器为2的示例字节。ADAU1401A每增加一个子地址寄存器两个字节，因为请求的子地址对应于一个寄存器或存储区，字长为两个字节。其他地址可以具有从一个到五个字节的字长。ADAU1401A总是对子地址进行解码并设置自动递增电路，使地址增加后适当的字节数。



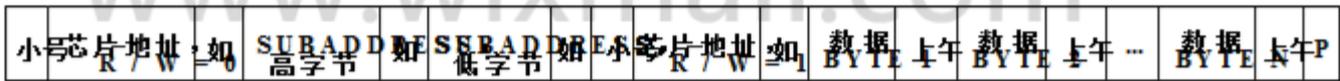
S = START位, P = 停止位, AS = 由SLAVE确认。
显示一个一个字的写，其中每个字都有N个字节。

图22.单字I2C写格式



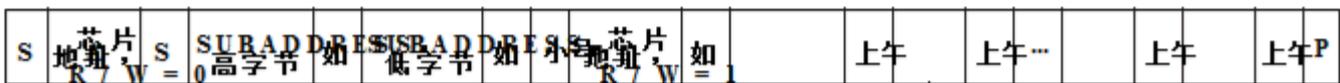
S = START位, P = 停止位, AS = 由SLAVE确认。
显示一个N-WORD写，每个字有两个字节。（其他字长是可能的，范围从1到5个字节。）

图23.突发模式I2C写入格式



S = 开始位, P = 停止位, AM = 由主器件确认, AS = 由应答器确认。
显示一个单词读，其中每个字有N个字节。

图24.单字I2C读格式



S = 开始位, P = 停止位, AM = 由主器件确认, AS = 由应答器确认。
显示一个N字读，其中每个字有两个字节。（其他字长是可能的，范围从1到5个字节。）

图25.突发模式I2C读取格式

SPI端口

默认情况下，ADAU1401A处于I2C模式，但可以放入SPI控制模式通过拉低CLATCH/WP三次。该SPI端口使用4线接口，由CLATCH，CCLK，CDATA和COUT信号，始终是从端口。该CLATCH信号在交易开始时应该变低并在交易结束时高。CCLK信号锁存CDATA在从低到高的过渡期间。COUT数据被移位应在CCLK的下降沿将ADAU1401A引出被定时到一个接收设备，如微控制器上CCLK的上升沿。CDATA信号载有串行输入数据，而COUT信号是串行输出数据。COUT信号保持三态，直到请求读操作。这允许其他SPI兼容外设共享相同的回读行。所有SPI事务都具有相同的基本格式如表19所示。时序图如图3所示数据应该先写入MSB。ADAU1401A不能在完全没有复位的情况下从SPI模式中取出。

芯片地址，R / W

SPI事务的第一个字节包含7位芯片地址和一个R / W位。芯片地址由ADDR0引脚设置。这个允许两个ADAU1401A器件共享一个CLATCH信号仍然独立运作。ADDR0为低电平时，芯片地址是0000000；当ADDR0为高电平时，地址为0000001（见表18）。这个第一个字节的LSB决定了SPI事务是读（逻辑电平1）或写（逻辑电平0）。

表18. ADAU1401A SPI地址字节格式

位0	位1	位2	位3	位4	位5	位6	位7
0	0	0	0	0	0	ADDR0	R / W

子地址

12位子地址字被解码为一个位置内存区域或寄存器。这个子地址就是这个地址适当的RAM位置或寄存器。子地址的MSB是零填充的单词到一个完整的2字节长度。

数据字节

数据字节数根据寄存器或者不同而不同内存位置被访问。在突发模式写入期间，首先写入子地址，然后是连续的序列的连续内存/寄存器位置的数据。详细表中显示了连续模式操作的数据格式23和表25在读/写数据格式部分。

一个单写SPI操作的示例时序图

参数RAM如图26所示。一个采样时序图图27显示了单读SPI操作的示意图。在图27中，COUT引脚从三态变为驱动字节3开头。在这个例子中，字节0到字节2包含字节地址和R / W位，以及随后的字节传送数据。

表19. 通用控制字格式

字节0	字节1	字节2	字节3	字节41
CHIP_ADR [6: 0], R / W	0000, SUBADR [11: 8]	SUBADR [7: 0]	数据	数据

1 继续数据结束。

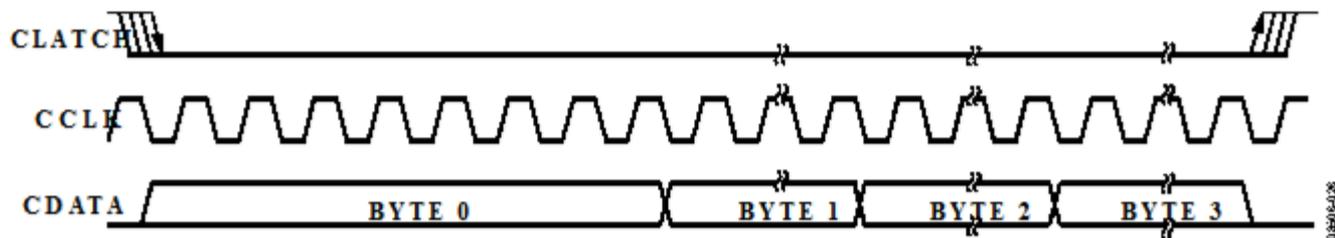


图26. SPI写入ADAU1401A时钟（单写模式）

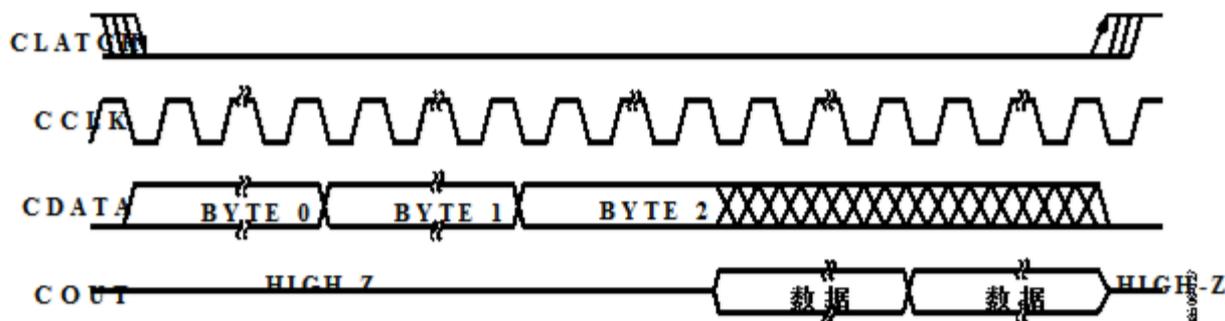


图27. 从ADAU1401A时钟读取SPI（单读模式）

SELF-BOOT

上电时, ADAU1401A可以加载一个程序和一个集合已保存在外部EEPROM中的参数.

结合辅助ADC和多用途引脚,

这消除了系统中的微控制器的需要.该

ADAU1401A作为主器件来完成自引导

在I²C总线上启动, 这发生在SELFB00T引脚上

被设置高. ADAU1401A不能在SPI模式下自启动.

程序和程序所需的最大EEPROM大小

参数是9248字节, 或者超过8.5 kB.这不是

包括寄存器设置或开销字节, 但是这样做

不添加大量的字节.这么多的记忆是

只有程序RAM (1024×5字节) 参数需要时才需要

RAM (1024×4个字节) 和接口寄存器 (8×4个字节)

完全满了.大多数应用程序不使用完整的程序

和参数RAM, 因此一个8 kB EEPROM应该足够了.

自复位操作在RESET的上升沿触发

当SELFB00T和WP引脚设置为高电平时. ADAU1401A

从中读取程序, 参数和寄存器设置

EEPROM.在ADAU1401A完成自启动后, 另外

尽管如此, 可以将消息发送到I²C总线上的ADAU1401A

这通常在自启动应用程序中不是必需的.该

I²C设备地址为0x68, 写入为0x69, 在此读取

模式. ADDR_x引脚在芯片上具有不同的功能

处于这种模式, 所以他们的设置可以被忽略.

如果WP设置为低电平, 则ADAU1401A不会自启动.保持

该引脚的低电平允许EEPROM在线编程.

WP引脚拉低 (通常有一个电阻上拉)

启用写入EEPROM, 但这反过来, 禁用

自引导功能, 直到WP引脚返回高电平.

ADAU1401A是自启动和I²C总线上的主设备

写回.虽然使用一个应用程序是不常见的

自启动也有一个微控制器连接到控制器

行, 应该小心, 没有其他设备试图写

到自启动或回写期间的I²C总线. ADAU1401A

以8×f_S产生SCL;因此, 对于48kHz的S, SCL是384kHz.

根据I²C规范, SCL的占空比为3/8.

ADAU1401A从EEPROM芯片地址0xA1读取.该

某些EEPROM地址的LSB是引脚可配置的;

在大多数情况下, 这些引脚应该连接到低电平来设置这个地址.

EEPROM格式

EEPROM数据包含一系列消息.每

离散消息是表20中定义的七种类型之一

并由一个或多个字节的序列组成.第一个字节

标识消息类型.字节先写入MSB.最

消息是块写入 (0x01) 类型, 用于写入

到ADAU1401A程序RAM, 参数RAM和控制寄存器.

消息类型之后的消息正文应该

从一个0x00字节开始;这是芯片地址.和其他所有一样

控制端口的交易, 下面的芯片地址是2个字节

寄存器/内存地址字段.

图28显示了一个应该存储在内存中的例子

EEPROM, 从EEPROM地址0x00开始.在这个例子中,

接口寄存器首先被设置为控制端口写入模式

(见图28的第1行), 然后是18无操作

(无操作) 字节 (参见图28的第2行到第4行), 以便于

接口寄存器数据出现在EEPROM的第2页.下一个

写头, 其中包括一个写, 长度和

设备地址 (见图28的第4行), 然后是32字节的

接口寄存器数据 (见图28的第5行到第8行).最后,

程序RAM数据, 从ADAU1401A地址0x04开始

写入0x00 (见图28的第9行到第11行).在这个例子中,

程序长度是70个字, 或350个字节, 所以332个字节

都包含在EEPROM中, 但未在图28中显示.

写回

当WB引脚被触发且数据正在写入时发生写回

从ADAU1401A写入EEPROM.这个功能

通常用于保存音量设置和其他参数

在将电源从电源断开之前将设置写入EEPROM

系统. WB引脚上的上升沿触发回写

器件处于自引导模式, 除非有消息设置WB引脚

在自引导中包含下降沿敏感 (0x05)

消息序列.只有一个回写发生, 除非一个

消息设置多个回写 (0x04) 包含在

自启动消息序列. a时, WP引脚被拉低

写回被触发以允许写入EEPROM.

ADAU1401A只能写回接口的内容

寄存到EEPROM.这些寄存器通常由

DSP程序, 但也可以在设置Bit 6之后直接写入

的DSP核心控制寄存器.应该的参数设置

保存在SigmaStudio中配置.

ADAU1401A

写回功能从ADAU1401A写入数据接口寄存器到自启动EEPROM的第二页，EEPROM地址0x20到EEPROM地址0x3F.开始于EEPROM地址0x1A（使接口寄存器数据从EEPROM地址0x20开始），EEPROM应该用六个字节编程 - 消息字节（0x01），两个长度字节，芯片地址（0x00）和2字节子地址用于接口寄存器（0x08，0x00）.必须有一个消息发送到DSP核心控制寄存器以启用写入接口寄存器之前的接口寄存器中的数据EEPROM.这应该存储在EEPROM地址0x00.消息之间可以使用无操作消息（0x03）确保满足这些条件.

ADAU1401A写入EEPROM芯片地址0xA0.该某些EEPROM地址的LSB是引脚可配置的.在大多数情况下，应将这些引脚连接到低电平，以将地址设置为0xA0.

从中写回的最大字节数ADAU1401A是35（八个4字节接口寄存器加三个EEPROM寻址开销字节）. SCL在384 kHz时，写回操作大约需要73μs才能完成.被触发后.确保有足够的电力可用.系统允许有足够的时间回写完成，特别是如果WB信号由降低功率触发的话.电源电压.

表20. EEPROM消息类型

消息ID	消息类型	以下字节
为0x00 0x01	结束 写	没有 两个字节表示消息的长度，后跟一个适当数量的数据字节
0x02 x03	延迟 没有操作执行	两个字节的延迟 没有
0x04	设置多个回写	没有
0x05	将WB设置为对下降沿敏感	没有
0x06	结束并等待回写	没有

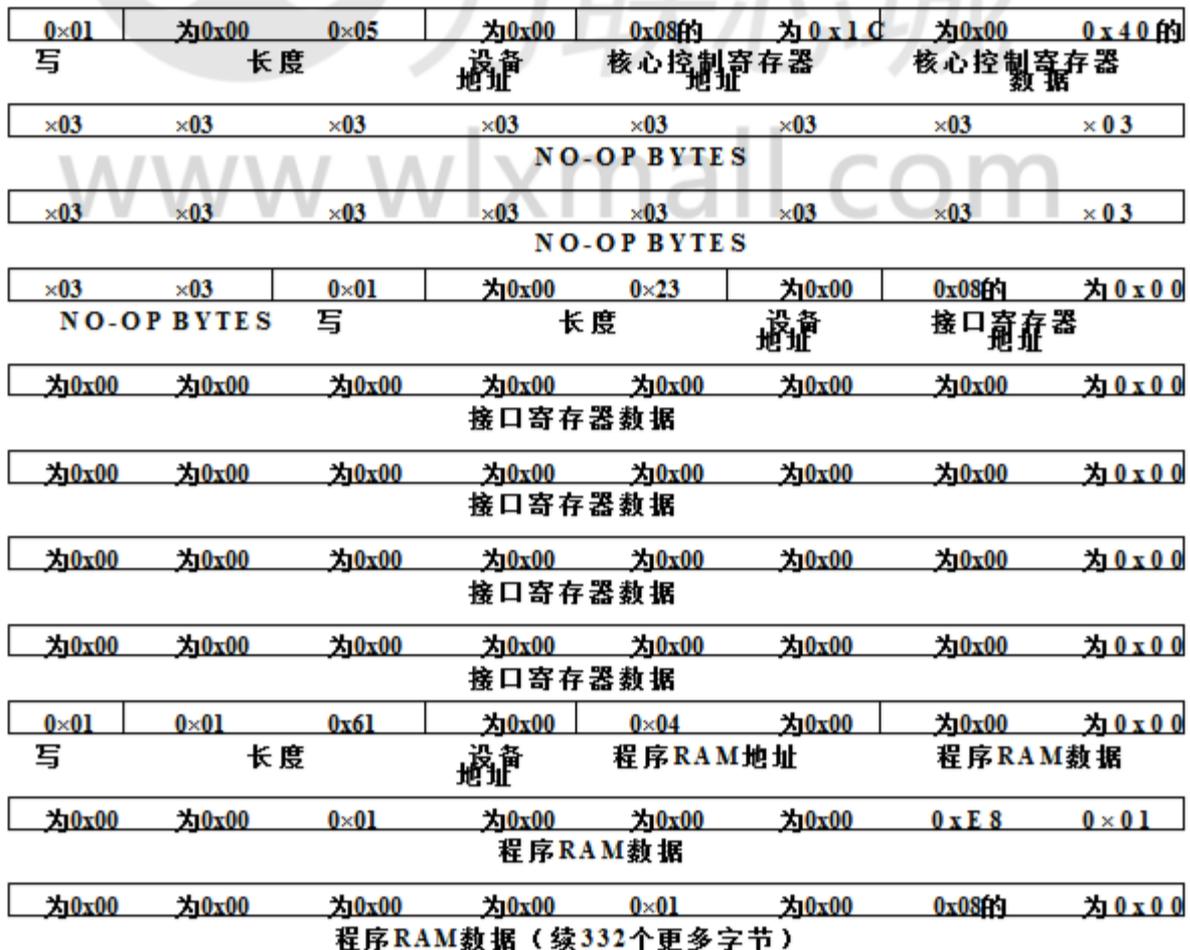


图28. EEPROM数据示例

信号处理

ADAU1401A旨在提供所有音频信号处理通常在立体声或多声道播放中使用的功能。信号处理流程是使用的SigmaStudio软件，它允许图形输入和实时操作，实时控制所有信号处理功能。

许多信号处理功能都使用完整的编码，56位双精度算术数据。输入和输出DSP内核的字长是24位。四个额外的空间在处理器中使用位来允许内部增益高达24 dB没有削波。额外的收益可以通过最初缩小DSP信号流中的输入信号。

数字格式

DSP系统通常使用标准的数字格式。分数系统由AB格式指定，其中A是小数点左边的位数，B是小数点右边的位数。

ADAU1401A使用相同的数字格式参数和数据值。格式如下所述。数值格式：5.23节。

数字格式：5.23

线性范围：-16.0到(+16.0 - 1 LSB)

例子：

1000 0000 0000 0000 0000 0000 0000 = -16.0
 1110 0000 0000 0000 0000 0000 0000 = -4.0
 1111 1000 0000 0000 0000 0000 0000 = -1.0
 1111 1110 0000 0000 0000 0000 0000 = -0.25
 1111 1111 0011 0011 0011 0011 0011 = -0.1
 1111 1111 1111 1111 1111 1111 1111 = (低于0.0 1 LSB)
 0000 0000 0000 0000 0000 0000 0000 = 0.0
 0000 0000 1100 1100 1100 1100 1101 = 0.1
 0000 0010 0000 0000 0000 0000 0000 = 0.25
 0000 1000 0000 0000 0000 0000 0000 = 1.0
 0010 0000 0000 0000 0000 0000 0000 = 4.0
 0111 1111 1111 1111 1111 1111 1111 = (16.0-1 LSB)。

串口最多可接受24位输入，扩展到DSP内核的全部28位。这允许内部没有内部削波的高达24 dB的增益。

在DSP的输出之间使用数字削波器电路内核和DAC或串口输出（见图29）。这个

剪辑信号的前四位以产生一个24位输出范围为1.0（减1 LSB）至-1.0。图29表示两个数据流中每个点的最大信号电平二进制和分贝水平。

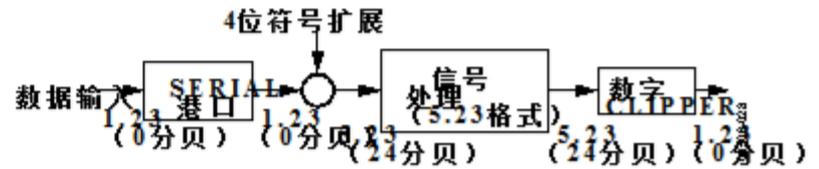


图29. 数字精度和裁剪结构

编程

上电时，ADAU1401A默认程序通过未经处理的输入信号（如图13所示），但是默认情况下输出是静音的（参见上电序列部分）。每个音频采样有1024个指令周期，导致约50 MIPS可用。SigmaDSP运行以面向流的方式，意味着全部1024条指令每个采样周期执行一次。ADAU1401A也可以设置通过减少数量来接受双速或四速输入。DSP内核控制寄存器中设置的每个采样的指令。

该部分可以使用SigmaStudio轻松编程（请参阅图30），这是ADI公司提供的图形工具。没有需要编写线路级DSP代码的知识。更多

有关SigmaStudio的信息可以在www.analog.com上找到。

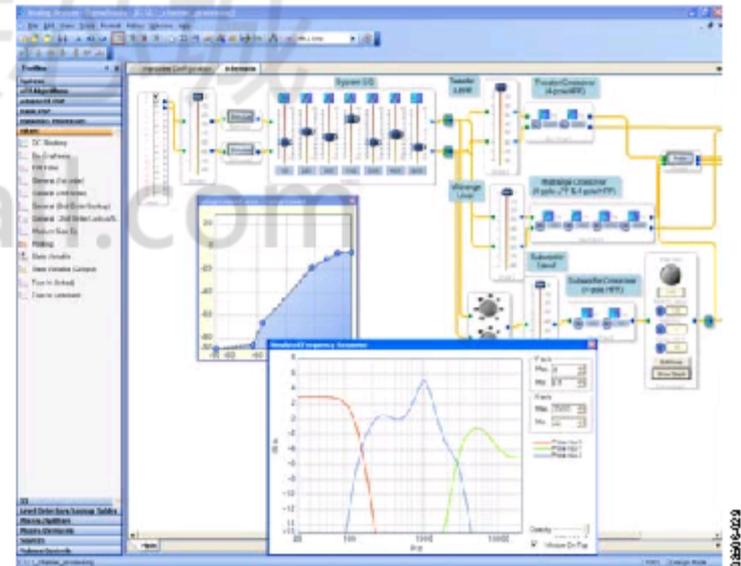


图30. SigmaStudio屏幕截图

RAMS和寄存器

表21. RAM映射和读/写模式

记忆	尺寸	地址范围	读	写	写模式
参数RAM	1024×32	0到1023 (0x0000到0x03FF)	是	是	直接写, 1安全写
程序RAM	1024×40	1024到2047 (0x0400到0x07FF)	是	是	直接写1

1 内部寄存器应该先清除, 以防止点击和弹出.

地址图

表21显示了RAM映射, 而表32显示了该映射 ADAU1401A寄存器映射. 地址空间包含一组寄存器和两个RAM: 一个RAM保存信号处理参数, 另一个RAM保存程序指令. 程序RAM和参数RAM在上电时被初始化, 从板载启动ROM (请参阅启动顺序部分).

所有的RAM和寄存器的默认值都是0, 除了程序RAM, 它是用默认程序加载的 (请参阅初始化部分).

参数RAM

参数RAM是32位宽, 占用地址0地址1023. 每个参数填充四个0之前MSB将28位字扩展为完整的4字节宽度. 该参数RAM在上电时被初始化为全0. 数据的参数RAM是二进制补码, 5.23格式. 这意味着系数的范围可以从+16.0 (减去1 LSB) 到-16.0, 用二进制字表示1.0 0000 1000 0000 0000 0000 0000 0000或十六进制字0x00 0x80 0x00 0x00.

参数RAM可以使用其中之一写入以下方法: 直接读/写或安全写入.

直接读/写

直接读/写方法允许直接访问程序RAM和参数RAM. 这种操作模式通常是使用突发模式寻址加载新的RAM时使用. 该DSP内核控制寄存器中的清零寄存器位应设置为0. 当使用这种模式来防止点击和弹出输出. 请注意, 这种模式可以在正在执行的程序中使用, 但是因为核心和控制之间没有握手端口, DSP核心期间参数RAM不可用. 控制写入, 导致音频流中的点击和弹出.

Safeload写

最多可以使用参数加载5个安全加载寄存器RAM地址和数据. 数据然后传输到当RAM不忙时请求的地址. 这个方法可以用于动态更新, 而直播节目素材通过ADAU1401A播放. 例如一个完整的一个双二阶段的更新可以发生在一个音频帧中而RAM不忙. 此方法不适用于写入程序RAM或控制寄存器.

数据RAM

ADAU1401A数据RAM用于存储音频数据字进行处理. 大部分情况下, 这个过程是透明的. 用户无法寻址RAM空间, 其中有一个大小2k字, 直接从控制端口.

实施时应考虑数据RAM的利用率. 需要大量数据RAM空间的块, 例如延迟. SigmaDSP内核处理一个样本的延迟时间增量. 因此, 用户可用的总延迟时间等于2048乘以采样周期. 对于48kHz的af S, 可用延迟池最多约43毫秒. 在实践中, 这么多的数据存储对用户是不可用的. 因为设计中的每个块都使用一些数据存储位置进行处理. 在大多数DSP程序中, 这并不重要, 影响总延迟时间. SigmaStudio编译器管理数据RAM并指示地址的数量. 在设计中需要超过可用的最大数量.

读/写数据格式

控制端口的读/写格式设计为字节为导向. 这样可以方便地编程常用的微处理器控制器芯片. 为了适应面向字节的格式, 附加0到MSB之前的数据字段扩展数据字八位. 例如, 写入参数的28位字RAM附加了四个前导0到32位 (四个字节), 而写入程序RAM的40位字是不附加0, 因为它们已经是一个完整的五个字节. 这些零填充的数据字段被附加到一个3字节的字段. 由一个7位芯片地址, 一个读/写位和一个11位组成RAM/寄存器地址. 控制端口知道有多少数据字节. 字节预计基于前三个字节中给出的地址.

单个位置写入命令的总字节数可以从四个字节 (用于控制寄存器写入) 变为八个字节 (用于程序RAM写入). 突发模式可用于填充连续寄存器或RAM位置. 突发模式写入从写入开始第一个RAM或寄存器位置的地址和数据. 书面. 而不是结束控制港交易 (通过发行I2C模式下的停止命令或CLATCH信号在数据字之后的SPI模式下高电平), 就像在a中所做的那样. 单地址写入, 下一个数据字可以立即写入时不指定地址. ADAU1401A控制端口会自动递增每个写入的地址, 甚至跨越每个写入地址不同的RAM和寄存器的边界. 表23和表25显示突发模式写入的例子.

表22.参数RAM读/写格式（单个地址）

字节0	字节1	字节2	字节3	字节[4: 6]
CHIP_ADR [6: 0], W / R	000000, PARAM_ADR [9: 8]	PARAM_ADR [7: 0]	0000, PARAM [27:24]	PARAM [23: 0]

表23.参数RAM块读/写格式（突发模式）

字节0	字节1	字节2	字节3	字节[4: 6]	字节[7:10]	字节[11:14]
CHIP_ADR [6: 0], W00R00, PARAM_ADR [9: 8]	PARAM_ADR [7: 0]	0000, PARAM [27:24]	PARAM [23: 0]
<PARAM_ADR->				PARAM_ADR + PARAM_ADR + 2		

表24.程序RAM读/写格式（单个地址）

字节0	字节1	字节2	字节[3: 7]
CHIP_ADR [6: 0], W / R	00000, PROG_ADR [10: 8]	PROG_ADR [7: 0]	PROG [39: 0]

表25.程序RAM块读/写格式（突发模式）

字节0	字节1	字节2	字节[3: 7]	字节[8:12]	字节[13:17]
CHIP_ADR [6: 0], W00R00, PROG_ADR [10: 8]	PROG_ADR [7: 0]	PROG [39: 0]
<PROG_ADR->			PROG_ADR + 1 PROG_ADR + 2		

表26.控制寄存器读/写格式（核心，串行输出0，串行输出1）

字节0	字节1	字节2	字节3	字节4
CHIP_ADR [6: 0], W / R	0000, REG_ADR [11: 8]	REG_ADR [7: 0]	数据[15: 8]	数据[7: 0]

表27.控制寄存器读/写格式（RAM配置，串行输入）

字节0	字节1	字节2	字节3
CHIP_ADR [6: 0], W / R	0000, REG_ADR [11: 8]	REG_ADR [7: 0]	数据[7: 0]

表28.数据捕获寄存器写入格式

字节0	字节1	字节2	字节3	字节4
CHIP_ADR [6: 0], W / R	00000, DATA_CAPTURE_ADR [11: 8]	DATA_CAPTURE_ADR [7: 0]	PROG_COUNT [10: 0]	PROG_COUNT [5: 0], 1 REGSEL [1: 0]

1 PROG_COUNT [10: 0]是数据捕捉发生时程序计数器的值（值由SigmaStudio编译器生成）。
 2 REGSEL [1: 0]选择四个寄存器中的一个（见地址2074和地址2075（0X081A和0X081B）-数据捕捉寄存器部分）。

表29.数据捕获（控制端口回读）寄存器读取格式

字节0	字节1	字节2	字节[3: 5]
CHIP_ADR [6: 0], W / R	0000, DATA_CAPTURE_ADR [11: 8]	DATA_CAPTURE_ADR [7: 0]	数据[23: 0]

表30. Safeload地址寄存器写入格式

字节0	字节1	字节2	字节3	字节4
CHIP_ADR [6: 0], W / R	00000, SAFELOAD_ADR [11: 8]	SAFELOAD_ADR [7: 0]	000000, PARAM_ADR [9: 8]	PARAM_ADR [7: 0]

表31. Safeload数据寄存器写入格式

字节0	字节1	字节2	字节3	字节4	字节[5: 7]
CHIP_ADR [6: 0], W00R, SAFELOAD_ADR [11: 8]	SAFELOAD_ADR [7: 0]	00000000	0000, 数据[27:24]	数据[23: 0]	...

控制寄存器映射

表32中的空白单元表示控制位在相应位置不存在。

表32.寄存器映射

寄存器地址 十六进制	没有的字节 序号	名称	MSB														LSB		默认																								
			D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16		D0																							
0x0800	2048	4 界面0 [31:16] 接口0 [15: 0]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	为0x0000 为0x0000																								
0x0801	2049	4 接口1 [31:16] 接口1 [15: 0]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	为0x0000 为0x0000																								
0x0802	2050	4 界面2 [31:16] 接口2 [15: 0]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	为0x0000 为0x0000																								
0x0803	2051	4 界面3 [31:16] 接口3 [15: 0]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	为0x0000 为0x0000																								
0x0804	2052	4 界面4 [31:16] 接口4 [15: 0]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	为0x0000 为0x0000																								
0x0805	2053	4 界面5 [31:16] 接口5 [15: 0]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	为0x0000 为0x0000																								
0x0806	2054	4 界面6 [31:16] 界面6 [15: 0]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	为0x0000 为0x0000																								
0x0807	2055	4 界面7 [31:16] 界面7 [15: 0]	0	0	0	0	IF27	IF26	IF25	IF24	IF23	IF22	IF21	IF20	IF19	IF18	IF17	IF16	为0x0000 为0x0000																								
0x0808	2056	2 GPIO引脚设置	0	0	0	0	MP11	MP10	MP09	MP08	MP07	MP06	MP05	MP04	MP03	MP02	MP01	MP00	0x00																								
0x0809	2057	2 辅助ADC数据0	0	0	0	0	AA11	AA10	AA09	AA08	AA07	AA06	AA05	AA04	AA03	AA02	AA01	AA00	0x0000																								
0x080A	2058	2 辅助ADC数据1	0	0	0	0	AA11	AA10	AA09	AA08	AA07	AA06	AA05	AA04	AA03	AA02	AA01	AA00	0x0000																								
0x080B	2059	2 辅助ADC数据2	0	0	0	0	AA11	AA10	AA09	AA08	AA07	AA06	AA05	AA04	AA03	AA02	AA01	AA00	0x0000																								
0x080C	2060	2 辅助ADC数据3	0	0	0	0	AA11	AA10	AA09	AA08	AA07	AA06	AA05	AA04	AA03	AA02	AA01	AA00	0x0000																								
0x080D	2061	5 保留[39:32] 保留[31:16] 保留 [15: 0]	RSVD	0x00 0x0000 0x0000																																							
0x080E	2062	5 保留[39:32] 保留[31:16] 保留 [15: 0]	RSVD	0x00 0x0000 0x0000																																							
0x080F	2063	5 保留[39:32] 保留[31:16] 保留 [15: 0]	RSVD	0x00 0x0000 0x0000																																							
0x0810	2064	5 Safeload Data 0 [39:32] Safeload Data 0 [31:15] Safeload Data 0 [15: 0]	SD39	SD38	SD37	SD36	SD35	SD34	SD33	SD32	SD31	SD30	SD29	SD28	SD27	SD26	SD25	SD24	SD23	SD22	SD21	SD20	SD19	SD18	SD17	SD16	SD15	SD14	SD13	SD12	SD11	SD10	SD09	SD08	SD07	SD06	SD05	SD04	SD03	SD02	SD01	SD00	为0x00 为0x0000 为0x0000
0x0811	2065	5 Safeload Data 1 [39:32] Safeload Data 1 [31:15] Safeload Data 1 [15: 0]	SD39	SD38	SD37	SD36	SD35	SD34	SD33	SD32	SD31	SD30	SD29	SD28	SD27	SD26	SD25	SD24	SD23	SD22	SD21	SD20	SD19	SD18	SD17	SD16	SD15	SD14	SD13	SD12	SD11	SD10	SD09	SD08	SD07	SD06	SD05	SD04	SD03	SD02	SD01	SD00	为0x00 为0x0000 为0x0000
0x0812	2066	5 Safeload Data 2 [39:32] Safeload Data 2 [31:15] Safeload Data 2 [15: 0]	SD39	SD38	SD37	SD36	SD35	SD34	SD33	SD32	SD31	SD30	SD29	SD28	SD27	SD26	SD25	SD24	SD23	SD22	SD21	SD20	SD19	SD18	SD17	SD16	SD15	SD14	SD13	SD12	SD11	SD10	SD09	SD08	SD07	SD06	SD05	SD04	SD03	SD02	SD01	SD00	为0x00 为0x0000 为0x0000
0x0813	2067	5 Safeload Data 3 [39:32] Safeload Data 3 [31:15] Safeload Data 3 [15: 0]	SD39	SD38	SD37	SD36	SD35	SD34	SD33	SD32	SD31	SD30	SD29	SD28	SD27	SD26	SD25	SD24	SD23	SD22	SD21	SD20	SD19	SD18	SD17	SD16	SD15	SD14	SD13	SD12	SD11	SD10	SD09	SD08	SD07	SD06	SD05	SD04	SD03	SD02	SD01	SD00	为0x00 为0x0000 为0x0000
0x0814	2068	5 Safeload Data 4 [39:32] Safeload Data 4 [31:15] Safeload Data 4 [15: 0]	SD39	SD38	SD37	SD36	SD35	SD34	SD33	SD32	SD31	SD30	SD29	SD28	SD27	SD26	SD25	SD24	SD23	SD22	SD21	SD20	SD19	SD18	SD17	SD16	SD15	SD14	SD13	SD12	SD11	SD10	SD09	SD08	SD07	SD06	SD05	SD04	SD03	SD02	SD01	SD00	为0x00 为0x0000 为0x0000
0x0815	2069	2 Safeload地址0	0	0	0	0	SA11	SA10	SA09	SA08	SA07	SA06	SA05	SA04	SA03	SA02	SA01	SA00	为0x0000																								
0x0816	2070	2 Safeload地址1	0	0	0	0	SA11	SA10	SA09	SA08	SA07	SA06	SA05	SA04	SA03	SA02	SA01	SA00	为0x0000																								
0x0817	2071	2 Safeload地址2	0	0	0	0	SA11	SA10	SA09	SA08	SA07	SA06	SA05	SA04	SA03	SA02	SA01	SA00	为0x0000																								
0x0818	2072	2 Safeload地址3	0	0	0	0	SA11	SA10	SA09	SA08	SA07	SA06	SA05	SA04	SA03	SA02	SA01	SA00	为0x0000																								
0x0819	2073	2 Safeload地址4	0	0	0	0	SA11	SA10	SA09	SA08	SA07	SA06	SA05	SA04	SA03	SA02	SA01	SA00	为0x0000																								
0x081A	2074	2 数据采集0	0	0	0	0	PC09	PC08	PC07	PC06	PC05	PC04	PC03	PC02	PC01	PC00	RS01	RS00	为0x0000																								
0x081B	2075	2 数据采集1	0	0	0	0	PC09	PC08	PC07	PC06	PC05	PC04	PC03	PC02	PC01	PC00	RS01	RS00	为0x0000																								

十六进制地址	寄存器名称	MSB																LSB														
		D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1
0x081C	2076	2	DSP核心控制	RSVD	RSVD	GD1	GD0	RSVD	RSVD	RSVD	AA	CV	CSFC	WA	TC	CR	SR1	SR0	为0x0000													
0x081D	2077	1	保留的	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x00																					
0x081E	2078	2	串行输出控制	0	0	OLRP	OBP	MOBF	LOBF	OF0	OL	LF0	EST	TDM	MSB2	MSB1	MSB0	OWL1	OWL0	0x0000												
0x081F	2079	1	串行输入控制	0	0	0	ILP	IBP	M2	M1	M0	为0x00																				
0x0820	2080	3	MP引脚配置. 0 [23:16]	MP53	MP52	MP51	MP50	MP43	MP42	MP41	MP40	0x00																				
			MP引脚配置. 0 [15:8]	MP33	MP32	MP31	MP30	MP23	MP22	MP21	MP20	MP13	MP12	MP11	MP10	MP03	MP02	MP01	MP00	0x0000												
0x0821	2081	3	MP引脚配置. 1 [23:16]	MP113	MP112	MP111	MP110	MP103	MP102	MP101	MP100	0x0000																				
			MP引脚配置. 1 [15:8]	MP93	MP92	MP91	MP90	MP83	MP82	MP81	MP80	MP73	MP72	MP71	MP70	MP63	MP62	MP61	MP60	0x0000												
0x0822	2082	2	辅助ADC和电源控制	RSVD	RSVD	RSVD	RSVD	RSVD	D0PD	D1PD	D2PD	D3PD	0x0000																			
0x0823	2083	2	保留的	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x0000																					
0x0824	2084	2	辅助ADC使能	AAEN	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x0000																				
0x0825	2085	2	保留的	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x0000																					
0x0826	2086	2	振荡器掉电	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x0000																					
0x0827	2087	2	DAC设置	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	0x0000																					



控制寄存器详细信息

地址2048到地址2055 (0x0800到0x0807) - 接口寄存器

接口寄存器用于自启动模式下进行保存需要写入外部EEPROM的参数。

ADAU1401A然后从这个回忆这些参数下一次复位或上电后的EEPROM.所以制度音量和EQ设置等参数可以在保存期间保存断电并在系统下一次打开时调用。

有8个32位接口寄存器，允许8个28位（加零填充）参数进行保存.参数

被保存在这些寄存器中被选中的图形编程工具.这些寄存器用它们更新每个采样周期对应的参数RAM数据一次。

触发可以设置为上升或下降的边沿 ADAU1401A写入接口的当前内容 保存到EEPROM.有关详细信息，请参阅自启动部分。

用户可以直接写入接口寄存器之后 接口寄存器控制端口写入模式位 (IFCW) 在DSP中 核心控制寄存器已经设置.在这种模式下，数据在 寄存器是从控制端口写入，而不是从DSP内核写入。

表33.接口寄存器位图

D31 D15	D30 D14	D29 D13	D28 D12	D27 D11	D26 D10	D25 D9	D24 D8	D23 D7	D22 D6	D21 D5	D20 D4	D19 D3	D18 D2	D17 D1	D16 D0	默认
0 IF15	0 IF14	0 IF13	0 IF12	IF27 IF11	IF26 IF10	IF25 IF09	IF24 IF08	IF23 IF07	IF22 IF06	IF21 IF05	IF20 IF04	IF19 IF03	IF18 IF02	IF17 IF01	IF16 IF00	为0x0000 为0x0000

表34.接口寄存器位描述

位名称	描述
IF [27: 0]	接口寄存器28位参数



ADDRESS 2056 (0x0808) - GPIO引脚设置寄存器

该寄存器允许用户通过GPIO来设置GPIO引脚控制端口.高或低设置可以直接写入或

设置GPIO引脚设置寄存器后, 从该寄存器读取数据控制端口写模式位 (GPCW) 在DSP核心控制寄存器. 该寄存器每LRCLK帧 (1 / fS) 更新一次.

表35. GPIO引脚设置寄存器位图

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	默认
0	0	0	0	MP11	MP10	MP09	MP08	MP07	MP06	MP05	MP04	MP03	MP02	MP01	MP00	为0x0000

表36. GPIO引脚设置寄存器位说明

位名称	描述
MP [11: 0]	当通过SPI或I2C控制时, 设置相应的多用途引脚



ADAU1401A

ADDRESS 2057 TO ADDRESS 2060 (0x0809 TO 0x080C) - 辅助ADC数据寄存器

这些寄存器保存4通道产生的数据
辅助ADC. ADC具有八位精度和可以
如果通过Bits FIL [1: 0]选择滤波, 则扩展为12位
辅助ADC和功率控制寄存器. SigmaDSP
程序读取这个数据作为一个范围的1.11格式的数据字

为0到1.0.这个数据字被映射到5.23格式
四个MSB和12个LSB设置为0的参数字
满量程代码255的值为1.0.这些寄存器
如果辅助ADC数据寄存器可以直接写入
控制端口写模式位(AACW)设置在DSP内核中
控制寄存器.

表37.辅助ADC数据寄存器位图

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	默认
0	0	0	0	AA11	AA10	AA09	AA08	AA07	AA06	AA05	AA04	AA03	AA02	AA01	AA00	为0x0000

表38.辅助ADC数据寄存器位描述

位名称	描述
AA [11: 0]	辅助ADC输出数据, MSB在先



ADDRESS 2064 TO ADDRESS 2068 (0x0810 TO 0x0814) - 自由数据寄存器

许多应用需要实时微控制器控制信号处理参数，例如滤波器系数，混频器增益，多通道虚拟化参数或动态处理曲线。例如，当控制双二阶滤波器时，所有的参数必须同时更新。这样做可以防止新旧系数混合执行过滤器为一个或两个音频帧，从而避免临时不稳定以及可能需要很长时间来衰减的瞬变。去完成 ADAU1401A 使用安全数据寄存器来同时进行将一组五个 28 位值加载到所需的参数 RAM 中地址。使用五个寄存器是因为二进制滤波器使用五个寄存器系数，如前所述，这是可取的做一个在一个交易中完成更新。

执行安全操作的第一步是写作参数地址给一个安全地址寄存器（地址 2069 到地址 2073）。10 位数据字是写入的是安全装入的参数 RAM 中的地址正在执行。该地址写入后，28 位数据字可以写入相应的安全数据寄存器（地址 2064 到地址 2068）。

这些写入的数据格式详见表 30 和表 31。表 39 概述了五个地址寄存器的每一个映射到其对应的数据寄存器。

地址和数据寄存器加载后，设置启动在 DSP 核心控制寄存器中的安全加载位开始加载到 RAM 中。五个 safeload 寄存器中的每一个都需要 1024 个核心指令之一加载到参数中内存。因此，总节目长度应该是有限的到 1019 个周期（1024 减 5）以确保 SigmaDSP 内核总是有至少五个周期可用。安全装置是有保证的。发生在一个 LRCLK 周期内（对于 48 kHz 的 f_S 为 21μs）初始化安全传输位被设置。

安全加载逻辑自动发送要加载的数据只有那些已被写入的安全加载寄存器的 RAM 自上次安全操作以来。例如，如果两个参数要在 RAM 中更新，只有五个 safeload 寄存器中的两个必须写。当启动安全加载传送位置位时，只有来自这两个寄存器的数据被发送到 RAM；另一个三个寄存器不被发送到 RAM，并可能保持旧的或无效数据。

表 39. Safeload 地址和数据寄存器映射

安全加载寄存器	安全加载地址寄存器	安全加载数据寄存器
Safeload 数据 0	2069	2064
Safeload 数据 1	2070	2065
Safeload Data 2	2071	2066
Safeload 数据 3	2072	2067
Safeload 数据 4	2073	2068

表 40. Safeload 数据寄存器位图

D31	D30	D29	D28	D27	D26	D25	D24	D39	D38	D37	D36	D35	D34	D33	D32	
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	默认
SD31	SD30	SD29	SD28	SD27	SD26	SD25	SD24	SD39	SD38	SD37	SD36	SD35	SD34	SD33	SD32	为 0x00
SD15	SD14	SD13	SD12	SD11	SD10	SD09	SD08	SD23	SD22	SD21	SD20	SD19	SD18	SD17	SD16	为 0x0000
								SD07	SD06	SD05	SD04	SD03	SD02	SD01	SD00	为 0x0000

表 41. Safeload 数据寄存器位描述

位名称	描述
SD [39: 0]	Safeload 数据。数据（程序，参数，寄存器内容）被加载到 RAM 或寄存器。

地址 2069 到地址 2073 (0x0815 到 0x0819) - SAFELoad 地址寄存器

表 42. Safeload 地址寄存器位图

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	默认
0	0	0	0	SA11	SA10	SA09	SA08	SA07	SA06	SA05	SA04	SA03	SA02	SA01	SA00	为 0x0000

表 43. Safeload 地址寄存器位描述

位名称	描述
SA [11: 0]	Safeload 地址。要加载到 RAM 或寄存器的数据的地址。

ADAU1401A

ADDRESS 2074和ADDRESS 2075 (0x081A AND 0x081B) - 数据捕捉寄存器

ADAU1401A数据采集功能允许任何数据信号处理流程中的节点将被发送到两个之一可读寄存器.此功能对于监控和管理非常有用.显示关于内部信号电平的信息或压缩机/限制器活动.

对于每个数据捕捉寄存器,一个捕捉计数和一个寄存器选择必须设置.捕捉计数是一个0到0之间的整数1023对应的程序步数在哪里捕捉是发生.寄存器选择字段编程四个之一寄存存在DSP内核中,将这些信息传送给DSP数据捕捉寄存器当程序计数器到达这一步.

捕捉的数据是5.19,补码数据格式,它来自内部的5.23数据字与四个LSB被截断.

为了设置数据捕捉必须写入的数据是a 10位程序计数索引与2位的连接注册选择字段.捕捉计数和寄存器选择值对应于要监视的期望点.信号处理流程可以在一个文件输出中找到程序编译器.捕捉寄存器可以被访问.阅读地点2074和地点2075.格式为写入和读取数据捕捉寄存器如图所示表28和表29.

表44.数据捕捉寄存器位图

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	默认
0	0	0	0	PC09	PC08	PC07	PC06	PC05	PC04	PC03	PC02	PC01	PC00	RS01	RS00	为0x0000

表45.数据捕捉寄存器位说明

位名称	描述										
PC [9: 0]	10位程序计数器地址										
RS [1: 0]	选择要传输到数据捕捉输出的寄存器										
	<table border="1"> <thead> <tr> <th>设置</th> <th>功能</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>选择乘法器X输入 (MULT_X_INPUT) 寄存器</td> </tr> <tr> <td>01</td> <td>选择乘数Y输入 (MULT_Y_INPUT) 寄存器</td> </tr> <tr> <td>10</td> <td>选择乘法器累加器输出 (MAC_OUT) 寄存器</td> </tr> <tr> <td>11</td> <td>选择累加器反馈 (ACCUM_FBACK) 寄存器</td> </tr> </tbody> </table>	设置	功能	00	选择乘法器X输入 (MULT_X_INPUT) 寄存器	01	选择乘数Y输入 (MULT_Y_INPUT) 寄存器	10	选择乘法器累加器输出 (MAC_OUT) 寄存器	11	选择累加器反馈 (ACCUM_FBACK) 寄存器
设置	功能										
00	选择乘法器X输入 (MULT_X_INPUT) 寄存器										
01	选择乘数Y输入 (MULT_Y_INPUT) 寄存器										
10	选择乘法器累加器输出 (MAC_OUT) 寄存器										
11	选择累加器反馈 (ACCUM_FBACK) 寄存器										

www.wlxmall.com

地址2076 (0x081C) -DSP内核控制寄存器

表46. DSP内核控制寄存器位图

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	默认
RSVD	RSVD	GD1	GD0	RSVD	RSVD	RSVD	AACW	GPCW	儿童福利论坛	ADM	坝	CR	SR1	SR0		为0x0000

表47. DSP内核控制寄存器位描述

位名称	描述	
GD [1: 0]	GPIO去抖控制.设置设置为GPIO输入的多用途引脚的去抖时间.	
	设置	时间 (毫秒)
	00	20
	01	40
	10	10
	11	五
AACW	辅助ADC数据寄存器控制端口写入模式.设置这个位允许数据直接写入来自控制端口的辅助ADC数据寄存器 (地址2057至地址2060).当这个位被设置时,辅助ADC数据寄存器忽略多用途引脚上的设置.	
GPCW	GPIO引脚设置寄存器控制端口写入模式.该位置1时,GPIO引脚设置寄存器 (地址2056)可以直接从控制端口写入,该寄存器将忽略多用途引脚上的输入设置.	
儿童福利论坛	接口寄存器控制端口写入模式.当该位置位时,数据可以直接写入接口从控制端口注册 (地址2048到地址2055).在这种状态下,接口寄存器不是从SigmaDSP程序写的.	
IST	启动安全传输.将该位置1将启动一个安全传输到参数RAM.这一点是操作完成后自动清除.有五个安全寄存器对 (地址/数据);只有自上次安全加载事件以来写入的寄存器才被传送到参数RAM.	
ADM	静音ADC.该位使ADC的输出静音.该位默认为0,低电平有效;因此,它一定是设置为1以传输来自ADC的音频信号.	
坝	静音DAC.该位将DAC的输出静音.该位默认为0,低电平有效;因此,它一定是设置为1来传输来自DAC的音频信号.	
CR	将内部寄存器清零.该位默认为0,低电平有效.它必须设置为1才能通过信号通过SigmaDSP内核.	
SR [1: 0]	采样率.这些位设置每个采样的DSP指令的数量和采样率 ADAU1401A运行.默认设置为1x,每个音频采样有1024条指令.这个设置应该与采样率一起使用,如48 kHz和44.1 kHz. 在2倍设置下,每帧指令的数量减半为512,ADC和DAC名义上运行采样率为96 kHz. 在4倍设置下,每个周期有256条指令,转换器以192 kHz采样率运行.	
	设置	功能
	00	1x (1024条指令)
	01	2x (512条指令)
	10	4x (256条指令)
	11	保留的

ADAU1401A

地址2078 (0x081E) -SERIAL输出控制寄存器

表48.串行输出控制寄存器位图

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	默认
0	0	OLRP	OBP	女士	OBF1	OBF0	OLF1	OLF0	FST	TDM	MSB2	MSB1	MSB0	OWL1	OWL0	为0x0000

表49.串行输出控制寄存器位说明

位名称	描述															
OLRP	OUTPUT_LRCLK极性.当此位设置为0时,左声道数据在OUTPUT_LRCLK为时钟时低电平,右声道数据在OUTPUT_LRCLK为高电平时钟.当该位设置为1时,通道数据在OUTPUT_LRCLK为低电平且左通道数据为时钟时钟OUTPUT_LRCLK很高.															
OBP	OUTPUT_BCLK极性.该位控制输出数据在哪个位时钟沿上计时.数据改变当该位设置为0时,OUTPUT_BCLK的下降沿,以及该位设置为1时的上升沿.															
女士	主从.该位设置输出端口是时钟主机还是从机.默认设置是从属.上电时,OUTPUT_BCLK和OUTPUT_LRCLK引脚被设置为输入,直到该位被设置为1,此时他们成为时钟输出.															
OBF [1: 0]	OUTPUT_BCLK频率(仅限主模式).当输出端口被用作主时钟时,这些位设置输出位时钟的频率,从内部1024×fS时钟(49.152 MHz, 48 kHz的fS).															
	<table border="1"> <thead> <tr> <th>设置</th> <th>功能</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>内部时钟/ 16</td> </tr> <tr> <td>01</td> <td>内部时钟/ 8</td> </tr> <tr> <td>10</td> <td>内部时钟/ 4</td> </tr> <tr> <td>11</td> <td>内部时钟/ 2</td> </tr> </tbody> </table>	设置	功能	00	内部时钟/ 16	01	内部时钟/ 8	10	内部时钟/ 4	11	内部时钟/ 2					
设置	功能															
00	内部时钟/ 16															
01	内部时钟/ 8															
10	内部时钟/ 4															
11	内部时钟/ 2															
OLF [1: 0]	OUTPUT_LRCLK频率(仅限主模式).当输出端口用作时钟主机时,这些位被设置OUTPUT_LRCLK引脚上的输出字时钟的频率,从内部分频1024×fS时钟(48 kHz的fS为49.152 MHz).															
	<table border="1"> <thead> <tr> <th>设置</th> <th>功能</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>内部时钟/ 1024</td> </tr> <tr> <td>01</td> <td>内部时钟/ 512</td> </tr> <tr> <td>10</td> <td>内部时钟/ 256</td> </tr> <tr> <td>11</td> <td>保留的</td> </tr> </tbody> </table>	设置	功能	00	内部时钟/ 1024	01	内部时钟/ 512	10	内部时钟/ 256	11	保留的					
设置	功能															
00	内部时钟/ 1024															
01	内部时钟/ 512															
10	内部时钟/ 256															
11	保留的															
FST	帧同步类型.该位设置OUTPUT_LRCLK引脚上的信号类型.当这个位被设置为0时,信号是占空比为50%的字时钟;当该位被设置为1时,该信号是具有一位持续时间的脉冲时钟在数据帧的开始处.															
TDM	TDM启用.将此位设置为1可将输出端口从四个串行立体声输出更改为一个8通道TDM输出流在SDATA_OUT0引脚(MP6)上.															
MSB [2: 0]	MSB职位.这三位设置数据MSB相对于LRCLK边沿的位置.数据ADAU1401A的输出始终为MSB.															
	<table border="1"> <thead> <tr> <th>设置</th> <th>功能</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>延迟1</td> </tr> <tr> <td>001</td> <td>延迟0</td> </tr> <tr> <td>010</td> <td>延迟8点</td> </tr> <tr> <td>011</td> <td>延迟12</td> </tr> <tr> <td>100</td> <td>延迟16</td> </tr> <tr> <td>101</td> <td>保留的</td> </tr> <tr> <td>111</td> <td>保留的</td> </tr> </tbody> </table>	设置	功能	000	延迟1	001	延迟0	010	延迟8点	011	延迟12	100	延迟16	101	保留的	111
设置	功能															
000	延迟1															
001	延迟0															
010	延迟8点															
011	延迟12															
100	延迟16															
101	保留的															
111	保留的															
OWL [1: 0]	输出字长.这些位设置输出数据字的字长.LSB后的所有位都置位为0.															
	<table border="1"> <thead> <tr> <th>设置</th> <th>功能</th> </tr> </thead> <tbody> <tr> <td>00</td> <td>24位</td> </tr> <tr> <td>01</td> <td>20位</td> </tr> <tr> <td>10</td> <td>16位</td> </tr> <tr> <td>11</td> <td>保留的</td> </tr> </tbody> </table>	设置	功能	00	24位	01	20位	10	16位	11	保留的					
设置	功能															
00	24位															
01	20位															
10	16位															
11	保留的															

地址2079 (0x081F) - 系统输入控制寄存器

表50. 串行输入控制寄存器位图

D7	D6	D5	D4	D3	D2	D1	D0	默认
0	0	0	ILP	IBP	M2	M1	M0	为0x00

表51. 串行输入控制寄存器位说明

位名称	描述																		
ILP	INPUT_LRCLK极性. 当该位设置为0时, SDATA_INx引脚上的左声道数据在时钟时钟. 当INPUT_LRCLK为高电平时, INPUT_LRCLK为低电平, 右通道数据为时钟. 当这个位被设置为1时, 这些通道的时钟是相反的. 在TDM模式下, 当该位被设置为0时, 数据被输入, 开始在INPUT_LRCLK引脚的下降沿之后, 下一个适当的BCLK边沿 (在该寄存器的位3中设置). 什么时候该位置1, 器件以TDM模式运行, 输入数据在上升后在BCLK边沿有效. 字时钟边缘 (INPUT_LRCLK). INPUT_LRCLK也可以使用脉冲输入而不是时钟进行操作. 在这种情况下, ADAU1401A使用脉冲的第一个边沿来启动数据帧. 当这个极性位设置为0, 应该使用低脉冲; 当它设置为1时, 应该使用高脉冲.																		
IBP	INPUT_BCLK极性. 该位控制输入数据在哪个位时钟沿上改变, 以及在哪个边沿上主频. 当该位设置为0时, 数据在INPUT_BCLK的下降沿改变; 当此位为上升沿时, 数据改变位设置为1.																		
M[2: 0]	<p>串行输入模式. 这三位控制输入端口期望接收的数据格式. 位3和位4. 该控制寄存器的设置将覆盖位[2: 0]的设置. 因此, 所有的五位必须一起改变. 在某些模式下正常操作. 这些模式的时钟图如图32, 图33和图32所示. 图34. 注意, 对于左对齐和右对齐模式, LRCLK极性为高电平, 然后为低电平, 即与ILP位的默认设置相反.</p> <p>当这些位被设置为接受TDM输入时, ADAU1401A数据将在ILP定义的边沿之后开始. 该ADAU1401A TDM数据流应输入到引脚SDATA_IN0上. 图35显示了一个TDM流, 低触发的LRCLK和数据在BCLK的下降沿改变. ADAU1401A预计最高的MSB. 每个数据插槽从插槽的开始处延迟一个BCLK, 因为它将以立体声I2S格式. 在TDM中模式中, 通道0到通道3在帧的前半部分, 通道4到通道7在第二个半. 图36显示了一个用脉冲字时钟运行的TDM流的例子, 该脉冲字用于接口. Analog Devices编解码器处于辅助模式. 要使用输入或输出串行端口在此模式下工作, 请设置ADAU1401A在LRCLK的上升沿开始帧, 在BCLK的下降沿改变数据, 将字时钟开始处的MSB位置延迟一个BCLK.</p> <table border="1"> <thead> <tr> <th>设置</th> <th>功能</th> </tr> </thead> <tbody> <tr> <td>000</td> <td>我 2 S</td> </tr> <tr> <td>001</td> <td>左对齐</td> </tr> <tr> <td>010</td> <td>TDM</td> </tr> <tr> <td>011</td> <td>右对齐, 24位</td> </tr> <tr> <td>100</td> <td>右对齐, 20位</td> </tr> <tr> <td>101</td> <td>右对齐, 18位</td> </tr> <tr> <td>110</td> <td>右对齐, 16位</td> </tr> <tr> <td>111</td> <td>保留的</td> </tr> </tbody> </table>	设置	功能	000	我 2 S	001	左对齐	010	TDM	011	右对齐, 24位	100	右对齐, 20位	101	右对齐, 18位	110	右对齐, 16位	111	保留的
设置	功能																		
000	我 2 S																		
001	左对齐																		
010	TDM																		
011	右对齐, 24位																		
100	右对齐, 20位																		
101	右对齐, 18位																		
110	右对齐, 16位																		
111	保留的																		

ADAU1401A

ADDRESS 2080和ADDRESS 2081 (0x0820 AND 0x0821) - 多个引脚配置寄存器

每个多用途引脚可以设置为不同的功能。这些寄存器（地址2080和地址2081）两个3字节寄存器被分成12个4位（半字节）的部分。

控制一个不同的MP引脚。表54列出了每个的功能。在MPx引脚配置寄存器中进行半字节设置。该每个引脚的4位配置的MSB反转输入或输出。从引脚内部上拉电阻（大约15kΩ）。每个MPx引脚在被设置为数字输入时被使能（a GPIO输入或串行数据端口输入）。

表52.寄存器2080位图

D15	D14	D13	D12	D11	D10	D9	D8	D23 D7	D22 D6	D21 D5	D20 D4	D19 D3	D18 D2	D17 D1	D16 D0	默认
MP33	MP32	MP31	MP30	MP23	MP22	MP21	MP20	MP 5 3 MP 1 3	MP 5 2 MP 1 2	MP 5 1 MP 1 1	MP50 MP10	MP43 MP03	MP42 MP02	MP41 MP01	MP40 MP00	为0x00 为0x0000

表53.寄存器2081位图

D15	D14	D13	D12	D11	D10	D9	D8	D23 D7	D22 D6	D21 D5	D20 D4	D19 D3	D18 D2	D17 D1	D16 D0	默认
MP93	MP92	MP91	MP90	MP83	MP82	MP81	MP80	MP113 MP 7 3	MP112 MP 7 2	MP 1 1 1 MP 7 1	MP110 MP 7 0	MP103 MP 6 3	MP102 MP 6 2	MP101 MP 6 1	MP100 MP 6 0	为0x00 为0x0000

表54.多用途引脚配置寄存器位描述

位名称	描述
盈丰[3: 0]	设置每个多用途引脚的功能。
	设置
1111	辅助ADC输入（见表63）
1110	保留的
1101	保留的
1100	串行数据端口反转（见表65）
1011	集电极开路输出反相
1010	GPIO输出反相
1001	GPIO输入，不反跳
1000	GPIO输入，去抖反转
0111	N/A
0110	保留的
0101	保留的
0100	串行数据端口（见表65）
0011	集电极开路输出
0010	GPIO输出
0001	GPIO输入，没有反跳
0000	GPIO输入，去抖动

ADDRESS 2082 (0x0822) - AUXILIARY ADC和功率控制寄存器

表55.辅助ADC和功率控制寄存器位图

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	默认
RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	FIL1	FIL0	AAPD	VBPD	VRPD	RSVD	D0PD	D1PD	D2PD	D3PD	为0x0000

表56.辅助ADC和功率控制寄存器位说明

位名称	描述	
FIL [1: 0]	辅助ADC滤波	
	设置	功能
	00	4位滞后 (12位电平)
	01	5位滞后 (12位电平)
	10	过滤器和迟滞绕过
11	低通滤波器绕过	
AAPD	ADC掉电 (两个ADC)	
VBPD	电压基准缓冲器掉电	
VRPD	电压参考关断	
D0PD	DAC0掉电	
D1PD	DAC1掉电	
D2PD	DAC2掉电	
D3PD	DAC3掉电	

ADDRESS 2084 (0x0824) - 辅助ADC使能寄存器

表57.辅助ADC使能寄存器位图

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	默认
AAEN	RSVD	为0x0000														

表58.辅助ADC使能寄存器位描述

位名称	描述
AAEN	启用辅助ADC

地址2086 (0x0826) - 振荡器掉电寄存器

表59.振荡器掉电寄存器位图

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	默认
RSVD	OPD	RSVD	RSVD	为0x0000												

表60.振荡器掉电寄存器位描述

位名称	描述
OPD	振荡器掉电, 关闭振荡器

ADDRESS 2087 (0x0827) -DAC设置寄存器

要正确初始化DAC, 该寄存器中的位DS [1: 0]应设置为01.

表61. DAC设置寄存器位图

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	默认
RSVD	DS1	DS0	为0x0000													

表62. DAC设置寄存器位说明

位名称	描述	
DS [1: 0]	DAC设置	
	设置	功能
	00	保留的
	01	初始化DAC
	10	保留的
11	保留的	

多用途PIN

ADAU1401A具有12个多用途（MP）引脚，可以单独编程用作串行数据输入，串行数据输出，数字控制输入和输出。SigmaDSP内核或输入到4通道辅助ADC。这些引脚允许ADAU1401A与外部ADC一起使用的DAC。他们还使用模拟或数字输入来控制设置如音量控制或使用输出数字信号来驱动LED指标。每个MP引脚都有一个内部15kΩ上拉电阻。

辅助ADC

ADAU1401A具有一个4通道，8位辅助ADC，可以与电位器配合使用来控制音量，音调，或DSP程序中的其他参数设置。四个中的每一个通道以音频采样频率（f_S）进行采样。满量程这个ADC上的输入是3.0V，因此步长是近似的12毫伏（3.0伏/256步）。ADC的输入电阻是大约30kΩ。表63显示了哪四个MP引脚映射到辅助ADC的四个通道。辅助通过将1111写入适当位，ADC可以使能这些引脚多用途引脚配置寄存器的一部分。

辅助ADC通过置1的AAEN位打开辅助ADC使能寄存器（见表58）。

ADC输入端的噪声会导致数字输出不断变化改变几个LSB。如果辅助ADC用于控制体积，这个不断的变化会导致小的增益波动。为避免这种情况，给辅助ADC增加一个低通滤波器或迟滞信号路径通过启用辅助ADC中的任一功能和功率控制寄存器（地址2082），如表56所述。辅助ADC启用时，默认启用滤波器。

当从辅助ADC寄存器读取数据时，有两个字节（12位数据，加上零填充的LSB）可用，因为这个过滤。

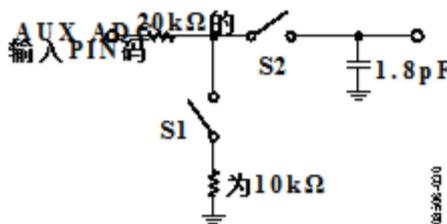


图31. 辅助ADC输入电路

图31显示了辅助ADC的输入电路。开关S1使能辅助ADC，由位15（AAEN）置1辅助ADC使能寄存器。采样开关S2工作于音频采样频率。

辅助ADC数据寄存器可以直接写入AACW已经被设置在DSP核心控制寄存器中。在这种模式下，模拟输入上的电压不写入寄存器，而是从控制端口写入寄存器中的数据。

PVDD为辅助ADC模拟电源提供3.3V电源输入。辅助ADC的数字内核是由A/D转换器供电的1.8V DVDD信号。

表63. 多用途引脚辅助ADC映射

多用途别针	功能
MP0	N / A
MP1	N / A
MP2	ADC1
MP3	ADC2
MP4	N / A
MP5	N / A
MP6	N / A
MP7	N / A
MP8	ADC3
MP9	ADC0
MP10	N / A
MP11	N / A

一般用途输入/输出引脚

通用输入/输出（GPIO）引脚可以用作输入或输出。这些引脚是可读的，可以设置通过控制接口或直接通过SigmaDSP内核。当设置为输入时，这些引脚可以与按钮一起使用开关或旋转编码器来控制DSP程序设置。数字输出可用于驱动LED或外部逻辑指示内部信号的状态并控制其他设备。这种使用的例子包括指示信号过载，信号目前，按下按钮的确认。

当设置为输出时，每个引脚通常可以驱动2mA。这个有足够的电流来直接驱动一些高效率的LED。标准的LED需要大约20毫安的电流，可以通过外部晶体管或缓冲器从GPIO输出驱动。由于同时驾驶或可能产生的问题在许多引脚上沉没大电流，应该小心应用程序设计，以避免连接高效率的LED直接连接到许多或所有的MPx引脚。如果需要很多LED，使用外部驱动程序。

当GPIO引脚设置为集电极开路输出时，应该拉到最大电压3.3V（电压在IOVDD上）。

串行数据输入/输出端口

ADAU1401A的灵活的串行数据输入和输出端口可以设置为以2通道格式接收或传输数据8通道TDM流。数据处理是二进制补码，MSB优先格式。左声道数据字段总是在前面2声道流中的右声道数据字段。在TDM中模式，插槽0到插槽3位于音频帧的前半部分插槽4到插槽7位于帧的后半部分。TDM模式允许使用更少的多用途引脚，释放更多的引脚用于其他功能。串行模式在串行输出中设置和串行输入控制寄存器。

串行数据时钟必须与ADAU1401A同步主时钟输入。串行输入控制寄存器允许控制

时钟极性和数据输入模式.有效的数据格式是 I2S, 左对齐, 右对齐 (24-/20-/18-/16-bit) 和8通道 TDM.在除了右对齐模式之外的所有模式下, 串行端口接受任意数量的比特, 最多为24个位不会导致错误, 但在内部被截断.正确右对齐模式的操作要求确切地存在每个音频帧64个BCLK. TDM数据在SDATA_IN0上输入. TDM模式下的LRCLK可以输入到ADAU1401A 可以是50/50的占空比时钟, 也可以是一个位宽的脉冲. 在TDM模式下, ADAU1401A可以成为48 kHz的主器件和96千赫的数据, 但不是192千赫的数据.表64列出了串行输出端口可以工作的模式.

表64.串行输出端口主/从模式功能

fS	2通道模式 (I2S, 左对齐, 右对齐)	8通道TDM
48千赫	主人和奴隶	主人和奴隶
96千赫	主人和奴隶	主人和奴隶
192千赫	主人和奴隶	奴隶只

串行输入和输出控制寄存器允许用户控制时钟极性, 时钟频率, 时钟类型和数据格式.在除右对齐模式之外的所有模式 (MSB延迟了8位, 12位或16位), 串口接受一个任意的位数达到24的限制.额外的位不会导致错误, 但在内部被截断.适当的正确操作模式要求LSB与LRCLK的边沿对齐. 所有串口控制寄存器的默认设置相对应到2通道I2S模式.所有寄存器设置适用于两个主和从属模式, 除非另有说明. 串行数据端口模式下每个多用途引脚的功能如表65所示.引脚MP0至引脚MP5支持数字数据输入到ADAU1401A, 引脚MP6到引脚MP11处理数字信号从DSP输出数据.串行数据的配置输入端口设置在串行输入控制寄存器中 (见表51), 并配置相应的输出端口由串行输出控制寄存器控制 (见表49).该

表66.数据格式配置

格式	LRCLK极性	LRCLK类型	BCLK极性	MSB职位
I2S (见图32)	帧在下降沿开始	时钟	数据在下降沿改变	从LRCLK边缘延迟由1 BCLK
左对齐 (见图33)	帧在上升沿开始	时钟	数据在下降沿改变	与LRCLK边缘一致
右对齐 (见图34)	帧在上升沿开始	时钟	数据在下降沿改变	从LRCLK边缘延迟由8,12或16 BCLKs
带时钟的TDM (见图35)	帧在下降沿开始	时钟	数据在下降沿改变	从词的开始延迟时钟由1 BCLK
带有脉冲的TDM (见图36)	帧在上升沿开始	脉冲	数据在下降沿改变	从词的开始延迟时钟由1 BCLK

输入端口的时钟只能作为从机使用, 而输出端口时钟可以设置为主或者功能奴隶. MP4 (INPUT_LRCLK) 和MP5 (INPUT_BCLK) 引脚被用来为SDATA_INx (MP0到MP3) 信号提供时钟, 并且MP10 (OUTPUT_LRCLK) 和MP11 (OUTPUT_BCLK) 引脚用于对SDATA_OUTx (MP6到MP9) 信号进行计时.

如果外部ADC作为ADAU1401A的从机连接, 使用输入和输出端口时钟. MP10 (OUTPUT_LRCLK) 和MP11 (OUTPUT_BCLK) 引脚必须设置为主模式并从外部连接到MP4 (INPUT_LRCLK) 和MP5 (INPUT_BCLK) 引脚, 以及外部ADC时钟输入引脚.数据从外部ADC输出MP0, MP1, MP2或MP3 (SDATA_INx) 引脚上的SigmaDSP.

到外部DAC的连接只能由处理器来处理输出端口引脚. MP10 (OUTPUT_LRCLK) 和MP11 (OUTPUT_BCLK) 引脚可以设置为主模式或从机, 并使用MP6至MP9 (SDATA_OUTx) 引脚将数据从SigmaDSP输出到外部DAC.

表66描述了标准音频的正确配置数据格式.

表65.多用途引脚串行数据端口功能

多用途别针	功能
MP0	SDATA_IN0 / TDM_IN
MP1	SDATA_IN1
MP2	SDATA_IN2
MP3	SDATA_IN3
MP4	INPUT_LRCLK (仅用于从机)
MP5	INPUT_BCLK (仅限从站)
MP6	SDATA_OUT0 / TDM_OUT
MP7	SDATA_OUT1
MP8	SDATA_OUT2
MP9	SDATA_OUT3
MP10	OUTPUT_LRCLK (主或从)
MP11	OUTPUT_BCLK (主或从)

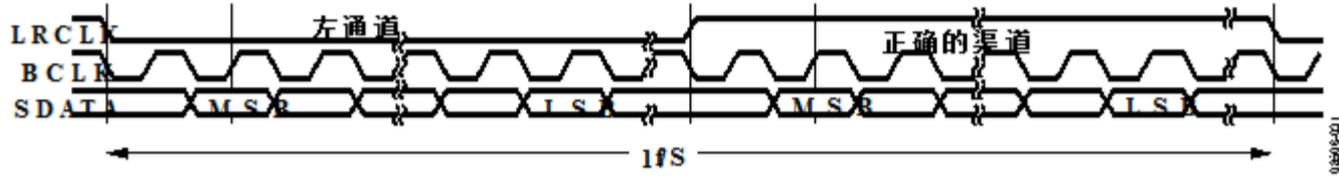


图32. 每个通道的I2S模式-16位至24位



图33.左对齐模式 - 每通道16位至24位



图34.右对齐模式 - 每通道16位至24位

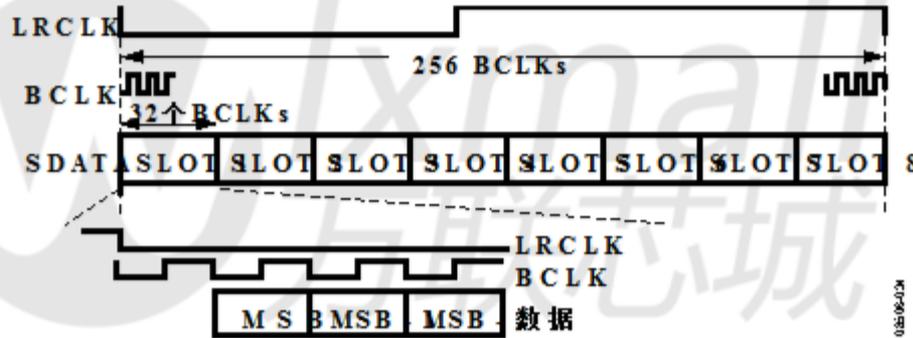


图35. TDM模式



图36.具有脉冲字时钟的TDM模式

布局建议

部件放置

ADC输入电压 - 电流电阻和ADC电流
设置电阻应尽可能靠近2, 3和
4个输入引脚.

所有100 nF旁路电容器, 推荐用于每个电容器
模拟, 数字和PLL电源/地线对, 应放置为
尽可能靠近ADAU1401A. 3.3 V和1.8 V
董事会的信号也应该绕过一个单一的
大容量电容器 (10 μ F至47 μ F) .

晶体振荡器电路中的所有迹线 (见图14) 应该是
尽量缩短杂散电容. 此外,
避免连接到任何这些组件的长电路板走线
因为这样的痕迹可能影响晶体启动和操作.

接地

在应用程序布局中应该使用单个地平面.
模拟信号路径中的组件应放置在远处
从数字信号.



典型应用范式

自我引导模式

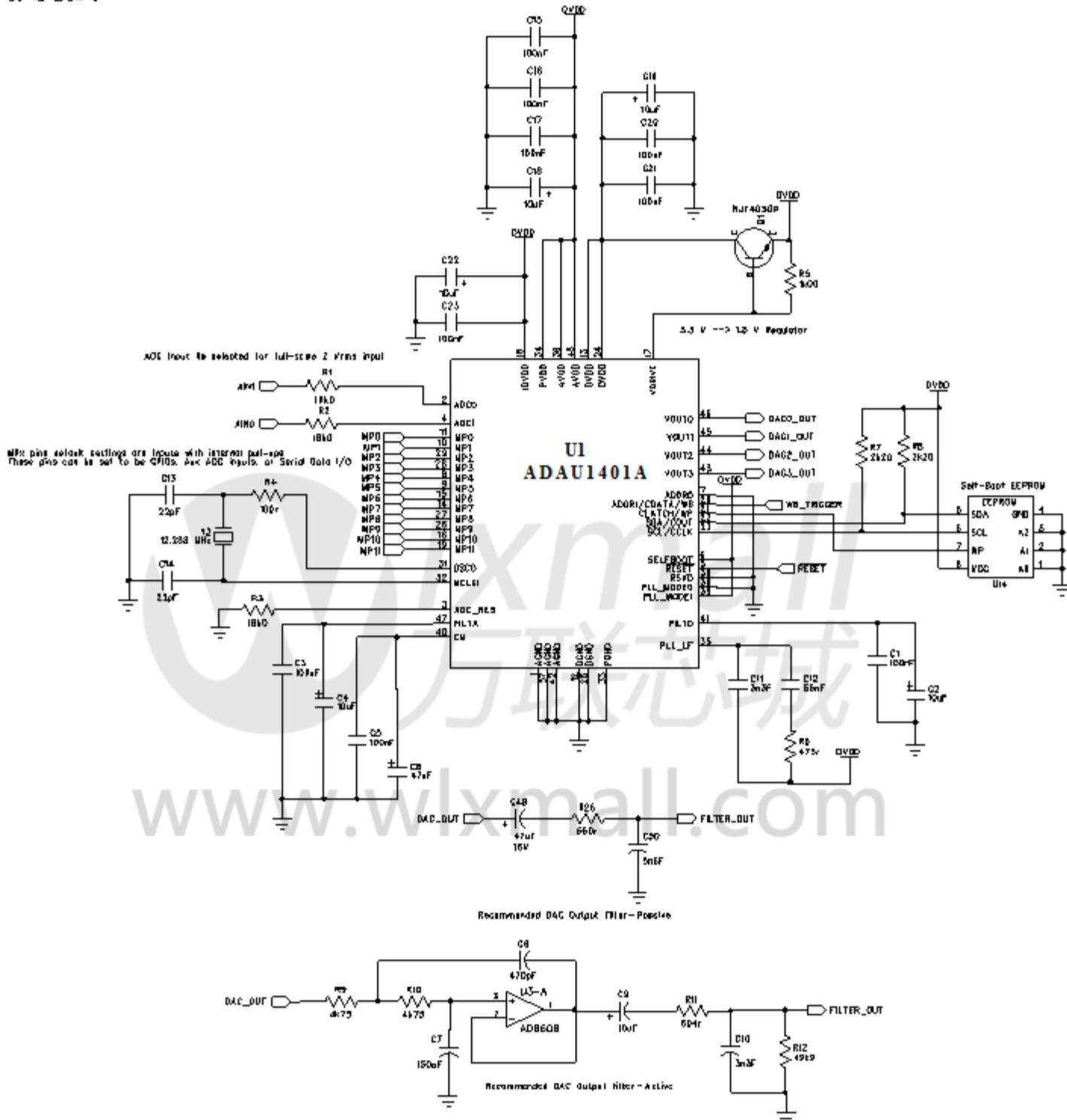


图37.自启动模式原理图

I2C控制

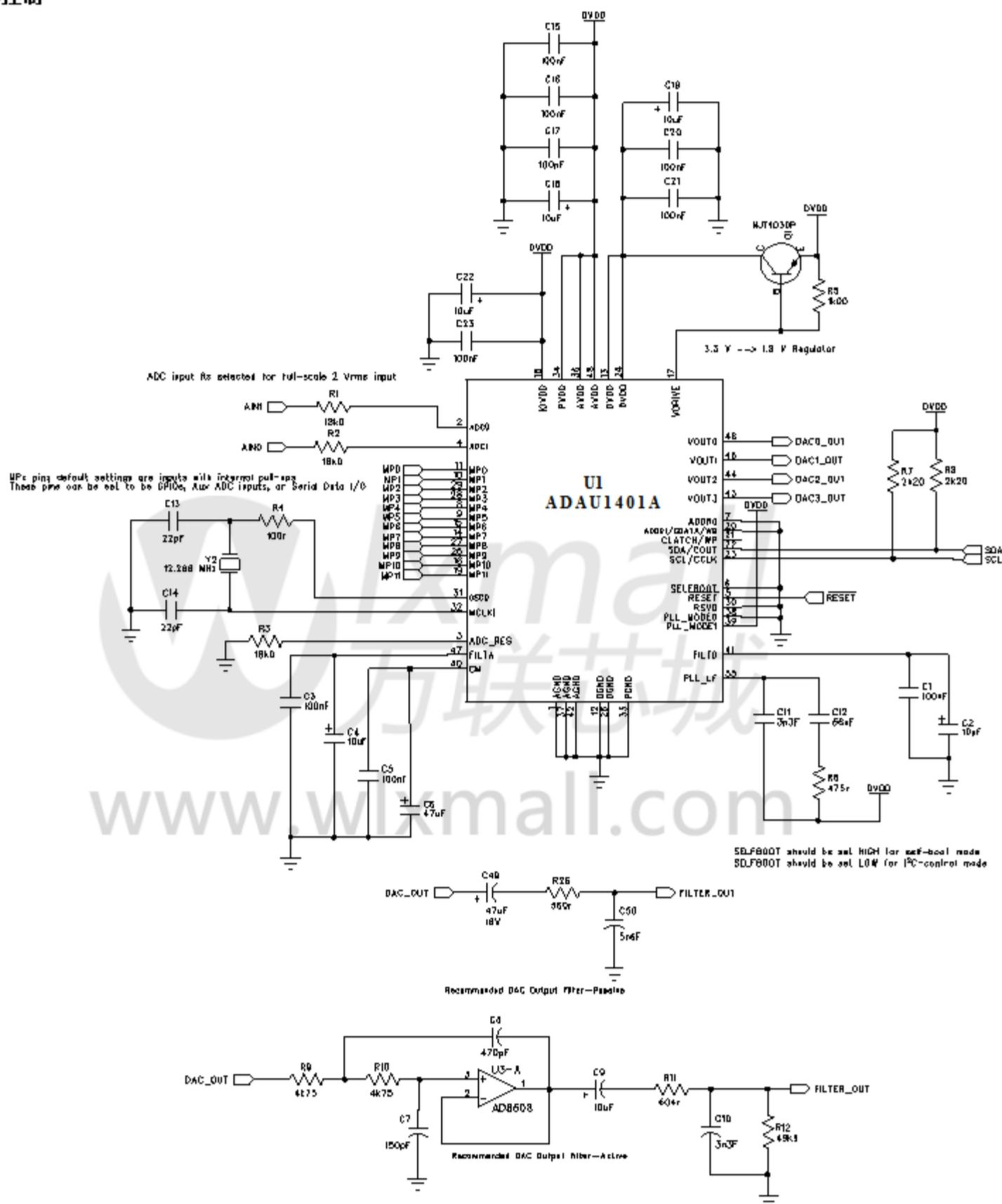


图38. I2C控制原理图

SPI控制

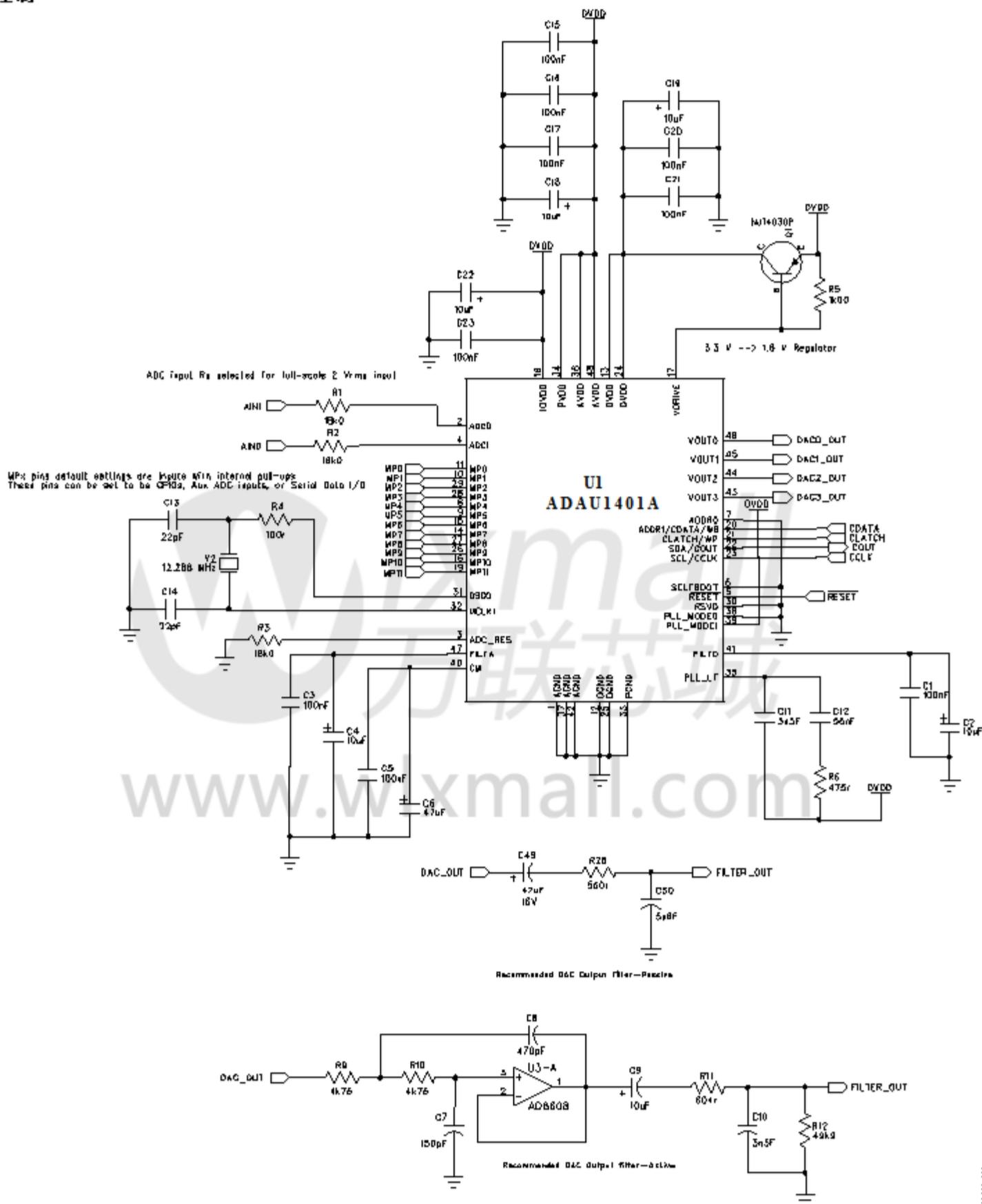
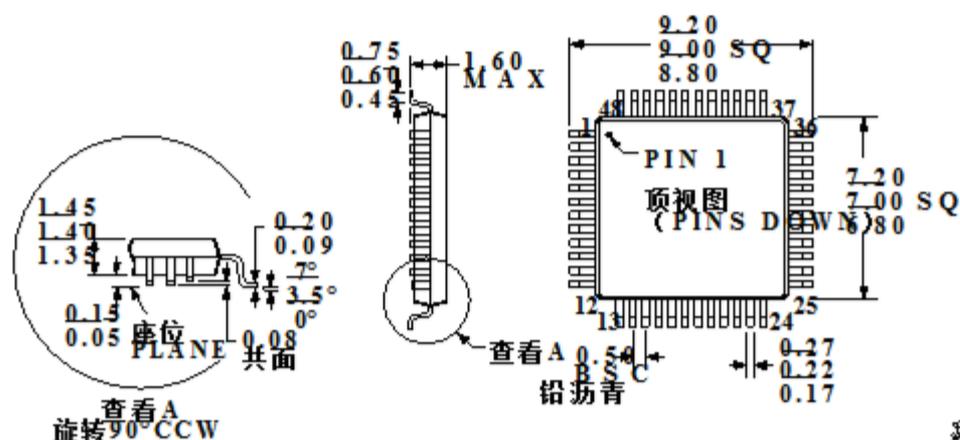


图39. SPI控制原理图

外形尺寸



符合JEDEC标准MS-026-BBC
图40. 48引脚薄型四方扁平封装[LQFP]
(ST-48)
尺寸以毫米为单位显示

订购指南

模型 1, 2	温度范围	包装说明	包装选项
ADAU1401AWBSTZ	-40°C至+ 105°C	48引脚LQFP	ST-48
ADAU1401AWBSTZ-RL EVAL-ADAU1401EBZ	-40°C至+ 105°C	采用13"卷带式48引脚LQFP封装 评估板	ST-48

1 Z =符合RoHS的部分
2 W =符合汽车应用要求

汽车产品

ADAU1401AWBSTZ和ADAU1401AWBSTZ-RL型号提供受控制制造，以支持质量和汽车应用的可靠性要求。请注意，这些汽车模型可能有不同的规格商业模式；因此，设计人员应仔细阅读本数据表的规格部分。只有汽车级所示产品可用于汽车应用。请联系您的ADI公司客户代表了解具体产品订购信息并获取这些型号的特定汽车可靠性报告。

笔记



I2C是指飞利浦半导体（现为恩智浦半导体）最初开发的通信协议。