VS1053b – Ogg Vorbis/MP3/AAC/WMA/MIDI 音 频 编 解 码 器

特性

- **Ogg Vorbis** 解码；
  MPEG 1 & 2 音频阶层 III (CBR +VBR +ABR)；
  阶层 I 和 II 可选；
  MPEG4/2 AAC-LC(+PNS)，
  HE-AAC V2 (级别 3) (SBR + PS)；
  WMA 4.0/4.1/7/8/9 所有特性[注1] （profiles）(5-384 kbps)；
  WAV (PCM + IMA ADPCM)；
  通用 MIDI 1 / SP-MIDI 格式 0 的文件
- **用软件插件进行 Ogg Vorbis 编码**（2007 第四季可用）
- “咪/线路”的输入信号可实现 IMA ADPCM 编码 (**立体声**)
- 支持 MP3 和 WAV 的数据流
- **EarSpeaker 空间效果[注2]处理**
- 低音和高音控制
- 只用一个单独的 12..13MHz 时钟运作
- 也可以使用一个 24..26MHz 时钟运作
- 内建 PLL 时钟乘法器
- 低功耗运作
- 芯片内建高质量和通道间无相位误差的立体声 DAC
- **过零交叉[注3]（Zero-cross）侦测和平滑的音量调整**
- 立体声耳机驱动器可以驱动一个 30Ω 的负载
- 安静的电源通断功能
- 可扩展外部 DAC 的 I2S 接口
- 分离的模拟、数字、IO 供电电源
- 供用户代码和数据使用的片内 RAM
- 用于控制和数据的串行接口
- 可以作为从模式的辅助处理器使用
- 特殊应用可使用 SPI FLASH 存储器引导
- 可用于调试的 UART 接口
- 可用软件增加新功能和提供最多 8 个 GPIO
- 符合 RoHS 无铅标准的封装（绿色）

注 1： 原文中是 profiles，含有轮廓、概要、资料档案等综合性信息的意思，这里将它译为“特性”意思更加贴切一些。

注 2： EarSpeaker Spatial 在这里指用耳机虚拟出真实现场空间的声音效果。

注 3： Zero-cross 指音频信号电平穿越零电位时的交叉点。例如：8bit 的采样数据使用 128 作为零位值，它代表音频信号 0 电平，正电平的音频采样值数据大于 128，负电平的音频采样值数据小于 128，越靠近零位值的采样值表示音频波形振幅越小，反之越大。如果一连串的采样值从大于 128 变成小于 128，或反过来，就是音频信号电平穿越了零电位。

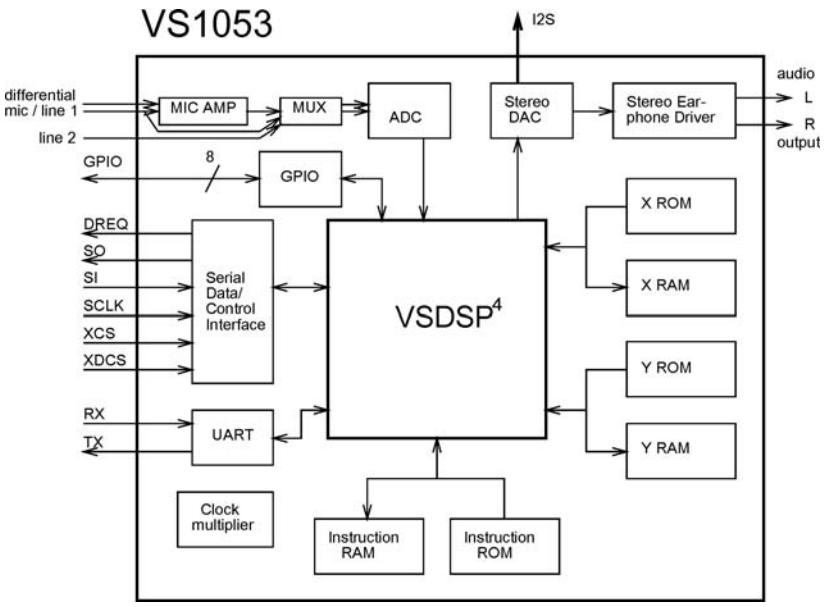
说明

VS1053b 是单片 Ogg Vorbis/MP3/AAC/WMA/MIDI 音频解码器，及 IMA ADPCM 编码器和用户加载的 Ogg Vorbis 编码器。它包含了一个高性能、有专利的低功耗 DSP 处理器内核 VS_DSP[4]、工作数据存储器、供用户应用程序和任何固化解码器一起运行的 16 KiB 指令 RAM 及 0.5 KiB 多的数据 RAM、串行的控制和输入数据接口、最多 8 个可用的通用 I/O 引脚、一个 UART、并有一个优质的可变采样率立体声 ADC（“咪”、“线路”、“线路+咪”或“线路*2”）和立体声 DAC、和跟随的一个耳机功放及一个公共电压缓冲器。

作为一个系统的从属设备，VS1053b 总是通过一个串行输入总线来接收它的输入比特流。该输入流被解码后始终会通过数字音量控制器送至一个 18 比特超采样率的[注4]（oversampling）、多比特的、sigma-delta 型高精度 DAC。此解码器是通过一个串行控制总线来控制的。除了基本的解码功能之外，它还可以增加特殊功能，象 DSP 功能之类等到用户的 RAM 存储器中。

可选的工厂编程单一芯片 ID，提供了数字版权管理的基础或单元识别特性。

注 4：超采样率 oversampling 一般是指超过标称采样率整倍数的更高采样率，常用来提高采样精度。这里是指还原数码声音时，用标称采样率数倍的采样率输出，可以使音频的阶梯陡变趋于平缓，降低数字背景噪音和减小失真，从而获得超过原采样率输出效果的高质量音频。



目录

插图列表

1 许可

MPEG Layer-3 audio decoding technology licensed from Fraunhofer IIS and Thomson.
Note: If you enable Layer I and Layer II decoding, you are liable for any patent issues that may arise from using these formats. Joint licensing of MPEG 1.0 / 2.0 Layer III does not cover all patents pertaining to layers I and II.

VS1053b contains WMA decoding technology from Microsoft.
This product is protected by certain intellectual property rights of Microsoft and cannot be used or further distributed without a license from Microsoft.

VS1053b contains AAC technology (ISO/IEC 13818-7 and ISO/IEC 14496-3) which cannot be used without a proper license from Via Licensing Corporation or individual patent holders.

VS1053b contains spectral band replication (SBR) and parametric stereo (PS) technologies developed by Coding Technologies. Licensing of SBR is handled within MPEG4 through Via Licensing Corporation. Licensing of PS is handled with Coding Technologies.
See http://www.codingtechnologies.com/licensing/aacplus.htm for more information.

To the best of our knowledge, if the end product does not play a specific format that otherwise would require a customer license: MPEG 1.0/2.0 layers I and II, WMA, or AAC, the respective license should not be required. Decoding of MPEG layers I and II are disabled by default, and WMA and AAC format exclusion can be easily performed based on the contents of the SCI_HDAT1 register. Also PS and SBR decoding can be separately disabled.

2 声明

This is a *preliminary* datasheet. All properties and figures are subject to change.

3 定义

B	字节，8 位宽度；
b	一个比特（一位宽度）；
Ki	“Kibi” = 2^{10} = 1024 (IEC 60027-2)；
Mi	“Mebi” = 2^{20} = 1048576 (IEC 60027-2)；
VS_DSP	VLSI Solution 公司的 DSP 内核；
W	VS_DSP 中的字，指令字是 32 位宽，数据字是 16 位宽。
咪	mic 指麦克风，又称“话筒”、“受话器”等，本文直接译作南方习惯用语：“咪”。
线路	line 在本文中是指用来输送经过放大的模拟信号的导线，译作“线路”或“线路输入”。
比特	bit 指二进制中的一位数字。本文根据情况会译作“位”、“位元”、“位域”或“比特”。
流	stream，是指在总线上不停传送的那些数据，因像流水般一直在流动，故称作“流”。
ADC/DAC	即“模数转换器”/“数模转换器”，为了保持译文简洁，本译文将直接使用这两个简写。
标头	Header 是音频文件放在文件开头的综合性信息快，含有：标题、子标题、类型、数据块大小等和音频相关的综合性信息。有些复杂的音频文件类型内容中含有多个标头。

4 特性和说明

4.1 最大极限值

参 数	符号	最小	最大	单位
模拟电源供电电压	AVDD	-0.3	3.6	V
数字电源供电电压	CVDD	-0.3	1.85	V
I/O 电源供电电压	IOVDD	-0.3	3.6	V
任意非电源引脚上的电流[1]			±50	mA
任意数字输入引脚上的电压		-0.3	IOVDD+0.3[2]	V
工作温度		-30	+85	℃
储存温度		-65	+150	℃

注1 大电流可能会造成闩锁效应。
注2 绝对不能超过 3.6 V。

4.2 推荐工作值

参数	符号	最小	典型	最大	单位
工作环境温度		-30		+85	℃
模拟和数字地[1]	AGND DGND		0.0		V
模拟电源供电电压，REF=1.23V	AVDD	2.5	2.8	3.6	V
模拟电源供电电压，REF=1.65V[2]	AVDD	3.3	3.3	3.6	V
数字电源供电电压	CVDD	1.7	1.8	1.85	V
I/O 电源供电电压	IOVDD	1.8	2.8	3.6	V
输入时钟频率[3]	XTALI	12	12.288	13	MHz
内部时钟频率	CLKI	12	36.864	55.3	MHz
内部时钟倍数[4]		1.0×	3.0×	4.5×	
主时钟周期占空比		40	50	60	%

注 1 在这个设备上必须紧密连接到一起，以避免出现闩锁效应。

注 2 参考电压可在内部选择 1.23V 或 1.65V，请参考章节 8.7.2。

注 3 可以用正确速度播放的最大采样率是 XTALI/256（或 XTALI/512，如果 SM_CLK_RANGE 被设置的话）。因此，XTALI 必须是至少 12.288MHz (24.576MHz)才能以正确的速度来演奏 48KHz。

注 4 复位时的初始值是 1.0×。推荐设置为 SC_MULT=3.5×，SC_ADD=1.0×(SCI_CLOCKF=0x8800)。但不能超出 CLKI 所允许的最大值。

4.3 模拟电路特性

如不另外说明，则：AVDD=3.3V，CVDD=1.8V，IOVDD=2.8V，REF=1.65V，TA=-30...+85℃，XTALI=12..13MHz，内部时钟倍数是 3.5×。DAC 的测试将以 1307.894Hz 满幅[1]（Full Scale）来输出正弦波，测试的频率范围为 20...20000Hz，模拟输出负载为：左通道至 GBUF 为 30Ω，右通道至 GBUF 为 30Ω。咪的测试信号幅度 48mVpp，fs=1 kHz，线路输入测试信号幅度 1.26 V，fs=1 kHz。

参数	符号	最小	典型	最大	单位
DAC 分辨率			18		bits
总谐波失真	THD			0.07	%
三次谐波失真				0.02	%
动态范围（DAC 非静音，A-加权）	IDR		100		dB
信噪比（信号满幅）	SNR		94		dB
通道隔离度（串扰），600Ω+GBUF			80		dB
通道隔离度（串扰），30Ω+GBUF			53		dB
通道增益失调		-0.5		0.5	dB
频率响应		-0.1		0.1	dB
满幅输出电压（峰峰值）		1.64	1.85[2]	2.06	Vpp
线性相位偏差				5	°
模拟输出 负载电阻	AOLR	16	30[3]		Ω
模拟输出 负载电容				100	pF
咪 输入放大器增益	MICG		26		dB
咪 输入信号幅度			48	140[4]	mVpp AC
咪 总谐波失真	MTHD		0.03	0.07	%
咪 信噪比	MSNR	60	70		dB
咪 输入阻抗（每只引脚）			45		kΩ
线路输入 信号幅度			2500	2800[4]	mVpp AC
线路输入 总谐波失真	LTHD		0.005	0.014	%
线路输入 信噪比	LSNR	85	90		dB
线路输入 阻抗			80		kΩ

注1 Full Scale 又称“满刻度”、“全刻度”、“满标度”等。

注2 可用单声道差分信号回路 +- 到 -+ 的方式来达到完全的 3.0V 输出幅度。

注3 模拟输出负载电阻也许会小于此值，但会导致失真而使其性能下降。

注4 超过典型幅度值会导致谐波失真增加。

4.4 功率消耗

用 MPEG 1.0 阶层 3 的 128 Kbps 采样和产生的正弦来测试，输出音量最大。内部时钟乘数为 3。环境温度为：TA=+25℃。

参数	最小	典型	最大	单位
电力消耗 AVDD，复位		0.6	5.0	µA
电力消耗 CVDD = 1.8V，复位		12	20.0	µA
电力消耗 AVDD，正弦波测试，30 Ω+ GBUF	30	36.9	60	mA
电力消耗 CVDD = 1.8V，正弦测试	8	10	15	mA
电力消耗 AVDD，无负载		5		mA
电力消耗 AVDD，输出负载 30 Ω		11		mA
电力消耗 AVDD, 30 Ω+ GBUF		11		mA
电力消耗 CVDD = 1.8V		11		mA

4.5 数字电路特性

参数	最小	最大	单位
高电平输入电压	0.7×CVDD	IOVDD+0.3[1]	V
低电平输入电压	- 0.2	0.3×CVDD	V
高电平输出电压 在 XTALO = -0.1 mA 时	0.7×IOVDD		V
低电平输出电压 在 XTALO = 0.1 mA 时		0.3×IOVDD	V
高电平输出电压 在 I_O = -1.0 mA 时	0.7×IOVDD		V
低电平输出电压 在 I_O = 1.0 mA 时		0.3×IOVDD	V
输入漏电流	-1.0	1.0	µA
SPI 输入频率[2]		CLKI / 7	MHz
所有输出引脚的上升时间，负载电容 = 50 pF		50	ns

注 1 绝对不能超过 3.6V

注 2 此值为 SCI 读取操作时的值。在 SCI 和 SDI 写入操作时则允许为 CLKI / 4。

4.6 开关特性 – 启动初始化

参数	符号	最小	最大	单位
XRESET 信号激活时间		2		XTALI
XRESET 信号取消后到软件准备好		22000	50000[1]	XTALI
复位时，电源上升至 CVDD 的时间		10		V/s

注 1 在 VS1053b 完成初始化和 DREQ 信号上升为高电平之前，你不应该发送任何数据或命令给它。

5 封装和引脚说明

5.1 封装

LPQFP-48 是一个无铅和符合 RoHS 标准的封装。RoHS 是官方指令 2002/95/EC 的简称，它限制在电子产品和电子设备中使用被确认的有害物质。

5.1.1 LQFP-48

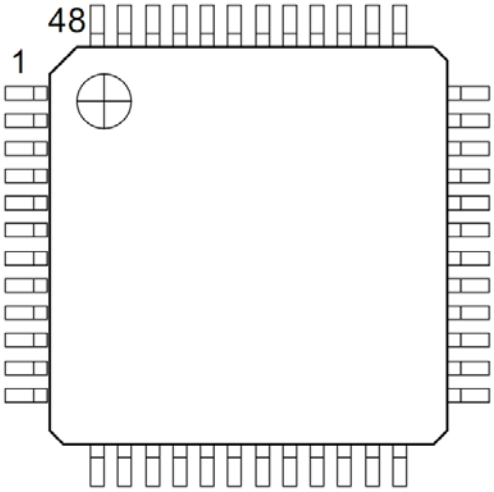


图 1：引脚分配图， LQFP-48

LQFP-48 封装尺寸可在 http://www.vlsi.fi/ 查询和下载。

图 2：VS1053b 的 LQFP-48 封装图

端子名称	LQFP 引脚	引脚类型	功能描述
MICP / LINE1	1	AI	咪的差分输入正极，自偏压 / 线路输入1
MICN	2	AI	咪的差分输入负极，自偏压
XRESET	3	DI	异步复位，低有效，施密特触发输入
DGND0	4	DGND	内核和 IO 接口的地线
CVDD0	5	CPWR	内核的正电源
IOVDD0	6	IOPWR	I/O 的正电源
CVDD1	7	CPWR	内核的正电源
DREQ	8	DO	数据请求，输入总线
GPIO2 / DCLK[1]	9	DIO	通用 IO 2 / 串行输入数据总线时钟
GPIO3 / SDATA[1]	10	DIO	通用 IO 3 / 串行数据输入
GPIO6 / I2S_SCLK[3]	11	DIO	通用 IO 6 / I2S 总线的 SCLK
GPIO7 / I2S_SDATA[3]	12	DIO	通用 IO 7 / I2S 总线的 SDATA
XDCS / BSYNC[1]	13	DI	数据片选 / 字节同步
IOVDD1	14	IOPWR	I/O 的正电源
VCO	15	DO	只用于测试 (时钟 VCO 输出)
DGND1	16	DGND	内核和 IO 接口的地线
XTALO	17	AO	晶振输出
XTALI	18	AI	晶振输入
IOVDD2	19	IOPWR	I/O 的正电源
DGND2	20	DGND	内核和 IO 接口的地线
DGND3	21	DGND	内核和 IO 接口的地线
DGND4	22	DGND	内核和 IO 接口的地线
XCS	23	DI	片选输入（低有效）
CVDD2	24	CPWR	内核的正电源
GPIO5 / I2S_MCLK[3]	25	DIO	通用 IO 5 / I2S 总线的 MCLK
RX	26	DI	UART 接收，如果不使用请连接到 IOVDD
TX	27	DO	UART 发送
SCLK	28	DI	串行总线的时钟
SI	29	DI	串行输入
SO	30	DO3	串行输出
CVDD3	31	CPWR	内核的正电源
XTEST	32	DI	为测试保留，连接到 IOVDD
GPIO0	33	DIO	通用 IO 0 （SPIBOOT），使用 100 kΩ 下拉电阻[2]
GPIO1	34	DIO	通用 IO 1
GND	35	DGND	I/O 地线
GPIO4 / I2S_LROUT[3]	36	DIO	通用 IO 4 / I2S 总线的 LROUT
AGND0	37	APWR	模拟地，低噪声参考
AVDD0	38	APWR	模拟正电源
RIGHT	39	AO	右通道输出
AGND1	40	APWR	模拟地线
AGND2	41	APWR	模拟地线
GBUF	42	AO	耳机的公共缓冲器，不可连接到地线！
AVDD1	43	APWR	模拟正电源
RCAP	44	AIO	参考电压的滤波电容
AVDD2	45	APWR	模拟正电源
LEFT	46	AO	左通道输出
AGND3	47	APWR	模拟地线
LINE2	48	AI	线路输入 2 （右通道）

注1 引脚的首要功能是在新模式下激活，次要功能则是在兼容模式下激活。

注2 除非下拉电阻被使用，否则将会尝试用SPI引导。请参阅章节9.9的描述。

注3 如果 I2S_CF_ENA 被设置为“0”，则引脚会被作为 GPIO 使用。请参阅章节 10.13 中的描述。

引脚类型

类型	说明
DI	数字输入，CMOS 输入端子
DO	数字输出，CMOS 输出端子[1]
DIO	数字输入 / 输出
DO3	数字输出，CMOS 三态输出端子
AI	模拟输入
AO	模拟输出
AIO	模拟输入/输出
APWR	模拟正电源引脚
DGND	内核或 I/O 地线引脚[2]
CPWR	内核正电源引脚
IOPWR	I/O 正电源引脚

注1 原文中误写为“输入端子”。

注2 前面的表格中描述为“内核和 I/O 地线引脚”，此处又描述为“内核或 I/O 地线引脚”，意味着它们在使用上并不需要严格区分。

6 接线图， LQFP-48

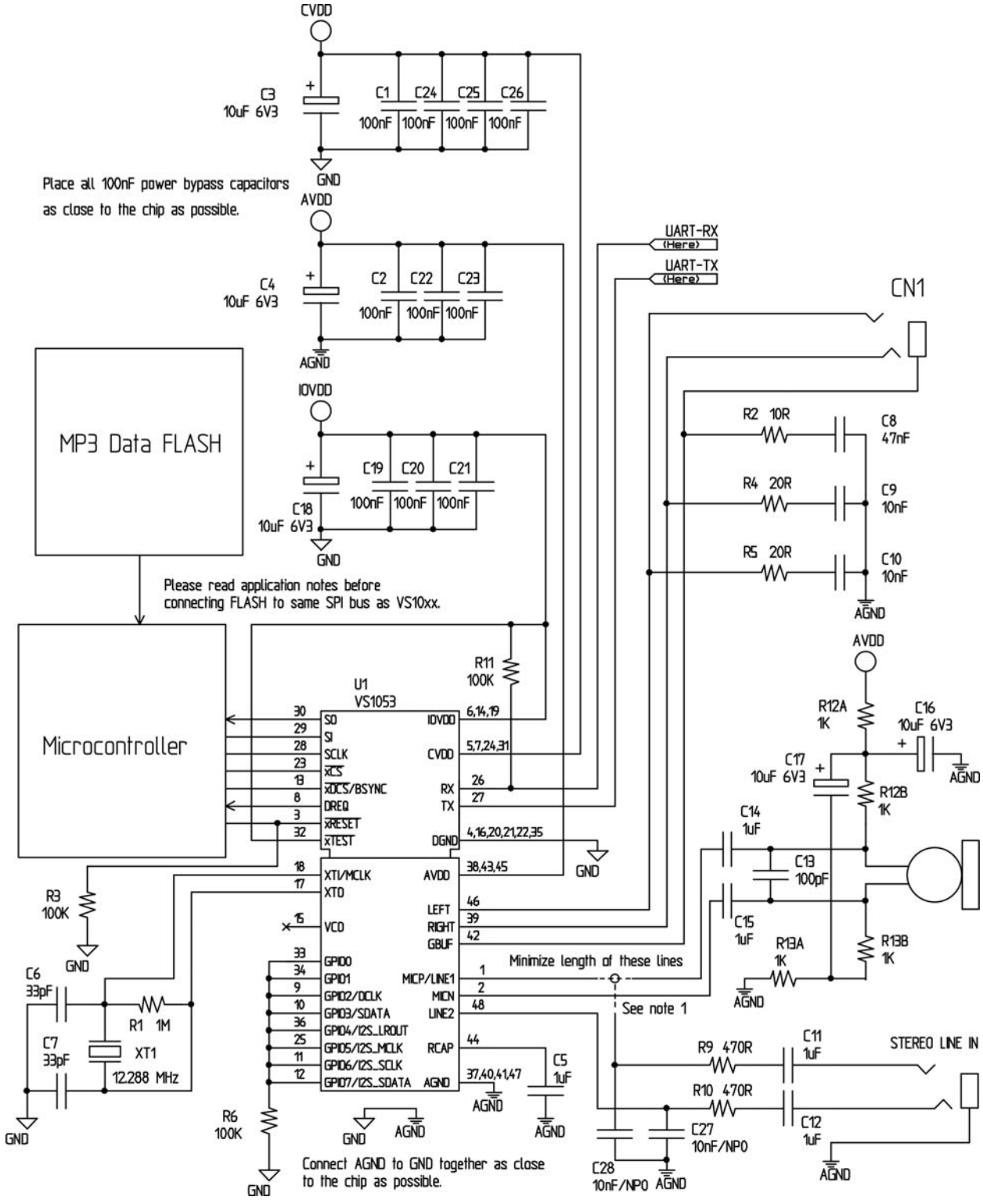


图 3：基于 LQFP-48 的典型接线图

图 3 显示了一个 VS1053 的典型接线图。

图注 1 ：可以连接咪输入或线路输入，但不能两者同时连接。

注意：这份连接假设 SM_SDINEW 是激活的（参看章节 8.7.1）。如果 SM_SDISHARE 也被使用，则 xDCS 应该被连接到低电平或高电平上（参看章节 7.2.1）。

公共缓冲器 GBUF 可以用作耳机输出通道的公共电压（1.23V）。这样便可取消输出线路上必须使用的大容量电容元件。因此，VS1053b 的音频输出端子可以与耳机的连线直接相连。

GBUF 在任何情况下都不允许连接到地线上。如果不使用 GBUF，则左通道和右通道必须要使用耦合电容器。要保持 GBUF 的稳定，即使不用 GBUF，你也应该总是使用这些电阻和电容。请参看应用指南中的详细注意事项。

没有使用的 GPIO 引脚应该接一个下拉电阻。不使用的线路输入和咪输入则不用连接。

如果不使用 UART，RX 应该连接到 IOVDD 上，TX 悬空。

请不要在 XTALO 上连接外部负载。

7 SPI 总线

7.1 通用

SPI 总线 - 起初是用在某些摩托罗拉的设备上，在这里被用于 VS1053b 的串行数据接口 SDI（章节 7.4 和 8.5）和串行控制接口 SCI（章节 7.5 和 8.6）。

7.2 SPI 总线引脚说明

7.2.1 VS1002 本地模式（新模式）

这个模式是在 VS1053b 的 SM_SDINEW 被设置为 1 时激活的（启动时的缺省值）。DCLK 和 SDATA 的功能被重新分别配置为 GPIO2, GPIO3，不再用于数据传输。BSYNC 功能被修改为数据接口的片选 XDCS。

SDI 引脚	SCI 引脚	说明
XDCS	XCS	**低有效的片选信号**。高电平将强制串行接口结束当前操作，并进入备用模式，它将强行使串行输出进入高阻状态。如果 SM_SDISHARE 是 1，则 XDCS 引脚是不使用的，但这个信号是通过将 XCS 反向来获得的。
SCK		**串行时钟输入**。此时钟也是用作内部寄存器接口的主时钟。SCK 电平在平时可以是脉冲状或平静的。不管在哪种情况之下，只要在 XCS 信号变低之后，首个时钟上升沿将被定义为首个位。
SI		**串行输入**。如果片选有效，SI 是在时钟 SCK 的上升沿上取样的。
–	SO	**串行输出**。在读取时，数据是逐位移动输出在时钟 SCK 的下降沿上。而在写入时，它是处于高阻态的。

7.2.2 VS1001 兼容模式（不推荐）

本模式在 SM_SDINEW 被设置为 0 时激活。在此模式下，DCLK, SDATA 和 BSYNC 功能被使用。

SDI 引脚	SCI 引脚	说明
-	XCS	**低有效的片选信号**。高电平将强制串行接口结束当前操作，并进入备用模式，它将强行使串行输出进入高阻状态。
BSYNC	-	SDI 数据是用 BSYNC 的上升沿来同步的。
DCLK	SCK	**串行时钟输入**。此时钟也是用作内部寄存器接口的主时钟。SCK 电平在平时可以是脉冲状或平静的。不管在哪种情况之下，只要在 XCS 信号变低之后，首个时钟上升沿将被定义为首个位。
SDATA	SI	**串行输入**。如果片选有效，SI 是在时钟 SCK 的上升沿上取样的。
-	SO	**串行输出**。在读取时，数据是逐位移动输出在时钟 SCK 的下降沿上。而在写入时，它是处于高阻态的。

7.3 数据请求引脚 DREQ

这个 DREQ 信号引脚是用来反馈 VS1053b 的 2048 字节 FIFO 是否可以接收数据。如果它为高电平，则 VS1053b 可以接收最少 32 字节的 SDI 数据或者接收一条 SCI 命令。当流缓冲区太满和 SCI 命令正在执行的期间，DREQ 会转换到低电平，此时应该停止向 VS1003 发送数据和新命令。

由于 32 字节是一个安全的范围，发送方可以一次发送完 32 字节的 SDI 数据之后再去检查 DREQ 的状态，这样就可以选择低速的单片机作为 VS1053b 的主控制器。

注意：DREQ 可能在任何时间转变状态，甚至在一个字节传送的期间。因此，DREQ 最好只用来确认是否要发送较多的字节，它不应该中断一个已经开始的发送过程。

译者注：只要当前的 DREQ 被检测有效，就表示 VS1053b 可以接收最少 32 字节的数据块（SDI 总线），或者一个完整的命令序列（SCI 总线）。在你完成这个发送的整个操作过程之前，无须理会 DREQ 的状态。

注意：在从 VS10xx（早期产品）到 VS1002 的产品内，DREQ 只用于 SDI。而在 VS1053b 中，DREQ 也用来反映 SCI 的状态。

要是在 DREQ 为低电平的情况下，你发送了 SCI 命令，因为 DREQ 是在 SDI 和 SCI 之间共享的，你将无法确定 SCI 命令是否会被执行和 SDI 数据是否准备好可以接收。在这种情况下，你必须要在每个 SCI 命令之后给出一个足够长的延时来确保它们不会被遗漏。有一份关于 SCI 的寄存器表在章节 8.7 中，它列出了在最坏情况下，对于各个寄存器进行写入操作时所需要处理的最大时间长度。

译者注：这里有个隐含的软件优化方法，即 DREQ 信号处在高电平的时候，表示 SDI 和 SCI 都是可以接收数据的。因此，你在 SDI 上发送了 32 个字节数据之后，还可以立即在 SCI 上发送一条 SCI 命令，或者将顺序倒过来也行。在这个两个连续的不同操作之间，DREQ 信号的状态是不用去关心的。

7.4 串行数据接口的串行协议（SDI）

7.4.1 通用

这个串行数据接口（SDI）是在“从机模式” 下运作的，因此必须要由外部的电路来产生 DCLK 信号。数据(SDATA 信号)可以任意定义在时钟信号 DCLK 的上升沿或下降沿上（章节 8.7）。

VS1053b 假设它的数据输入是字节同步的。SDI 字节传送可以是 MSb 或 LSb 在前，这取决于 SCI_MODE 中内容的定义（章节 8.7.1）。

此固件可以接受 SDI 所支持的最高比特率。

7.4.2 VS1002 本地模式下的 SDI（新模式）

在 VS1002 本机模式下 (SM_NEWMODE 是 1)， 字节同步是由 XDCS 完成的。在一个字节数据传送的过程中，XDCS 状态是不允许改变的。在实际应用中，推荐经常去翻转 XDCS 信号（注意：这里是指在数据块传送完之后翻转），例如在传送了一个磁盘的数据块之后就这样做一次。这样的话，即使板上的 VS1053b 在使用时出了一点小问题，它也能使数据立即重新同步，如此就能保证主控制器和 VS1003 之间的同步运作了。

如果 SM_SDISHARE 是 1，则 XDCS 信号是通过 XCS 输入信号在内部反相后产生的。在新的设计中，推荐使用 VS1002 的本机模式。

7.4.3 VS1001 兼容模式下的 SDI（不推荐）

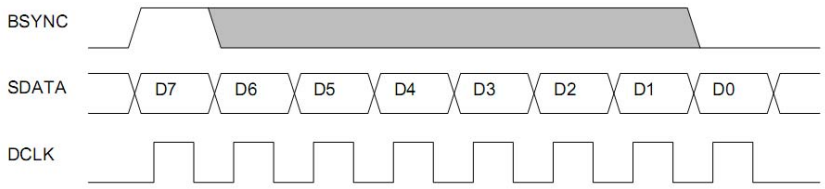


图 4：BSYNC 信号 – 单字节传送

当 VS1053b 运行在 VS1001 兼容模式下时，对输入的比特流必须要产生一个 BSYNC 信号来确保位元可以同步。在第一个 DCLK 的采样沿上（上升或下降，取决于所选择的极性），BSYNC 在此期间为高，标识出此字节的第一位元（如果 LSB 在前被使用，则此位是 LSB，，反之则为 MSB）。如果在接收最后一位时 BSYNC 是“1”，则接收器继续保持活动状态和后面的 8 个位元也同样被接收。

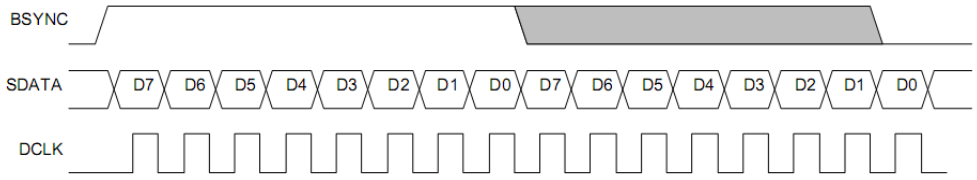


图 5：BSYNC 信号 – 双字节传送

7.4.4 被动 SDI 模式

如果 SM_NEWMODE 是 0 和 SM_SDISHARE 是 1，这个操作将类似于 VS1001 兼容模式，但不同处是那些位元只会在 BSYNC 信号为“1”的期间才被承认。BSYNC 的上升沿仍是用于同步。

7.5 串行命令接口的串行协议（SCI）

7.5.1 通用

此串行命令接口 SCI（章节 8.6）的串行总线协议包括：一个指令字节、一个地址字节和一个 16 位的数据字。每次读取操作或写入操作均可以访问一个寄存器。由于那些数据位是在 SCK 的上升沿读取的，所以用户只能在 SCK 的下降沿上更新数据。每个字节的 MSb 总是被首先发送。在整个传送操作的期间，XCS 必须要保持为低电平，但是在必要时，你可以在那些位元的中间暂停。

译者注：这里的暂停是指暂停 SCK 信号，这样的话，微控制器可以临时去处理别的事物，然后再返回继续传送。在此期间是不可改变 XCS 信号状态，否则将导致当前的传送失败。这种状况多数会出现在用软件模拟的 SPI 总线上，当微控制器去响应一个中断时，它所模拟的当前 SPI 传送周期就必须暂停一下，等处理完中断响应后再返回继续模拟这个 SPI 周期，直到它完全模拟处理完毕。

这些操作是通过 8 位指令码来指定的。所支持的指令是读取操作和写入操作，如下表所示：

指令		
名称	操作码	操作
READ	0b0000 0011	读取数据
WRITE	0b0000 0010	写入数据

注意：VS1053b 在每次的 SCI 操作开始后会将 DREQ 设置为低电平，而这段时间的长短将依赖于具体的操作行为。它在 DREQ 重新恢复到高电平之前是不允许去启动一个新的 SCI / SDI 操作的。

7.5.2 SCI 读操作

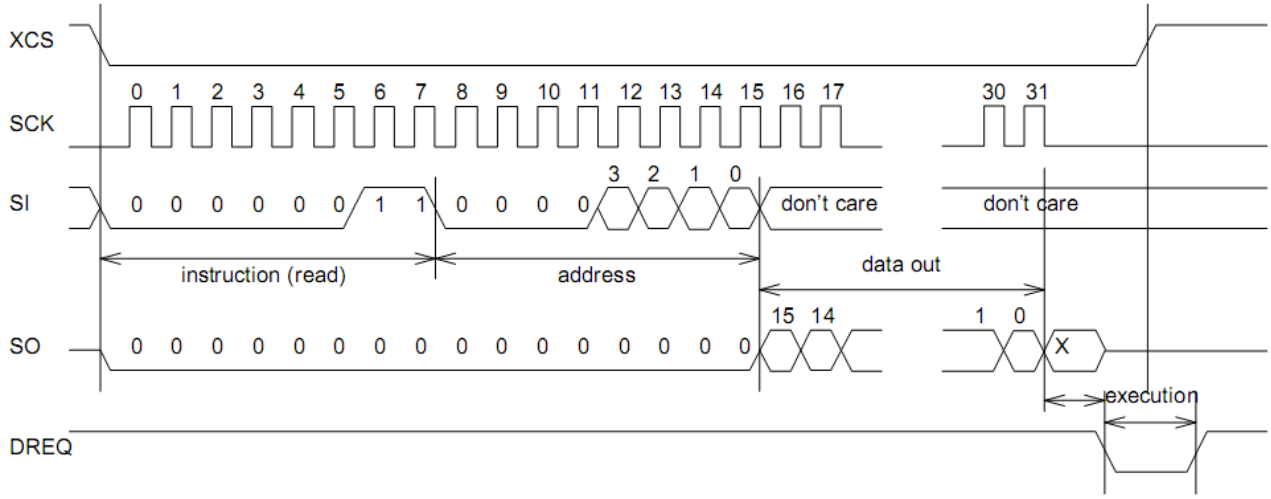


图 6: SCI 字读取操作

VS1053b 使用下列时序对寄存器进行读取操作，它显示在图 6 中。首先，XCS 信号线被拉到低电平来片选此设备。随后，读取操作码(0x3)加上 8 位宽度的地址后，组成的 16 位字通过 SI 信号线发送到设备。在地址被读取之后，SI 信号线上发送的任何数据都将被芯片忽略。而被确认的地址中的十六位宽度数据将在 SO 信号线上移动输出。

XCS 信号应该在数据移动送出之后驱动到高电平。

DREQ 在读取操作期间会被芯片短暂的拉到低电平，这是非常短的时间，并不需要用户特别的留意。

7.5.3 SCI 写操作

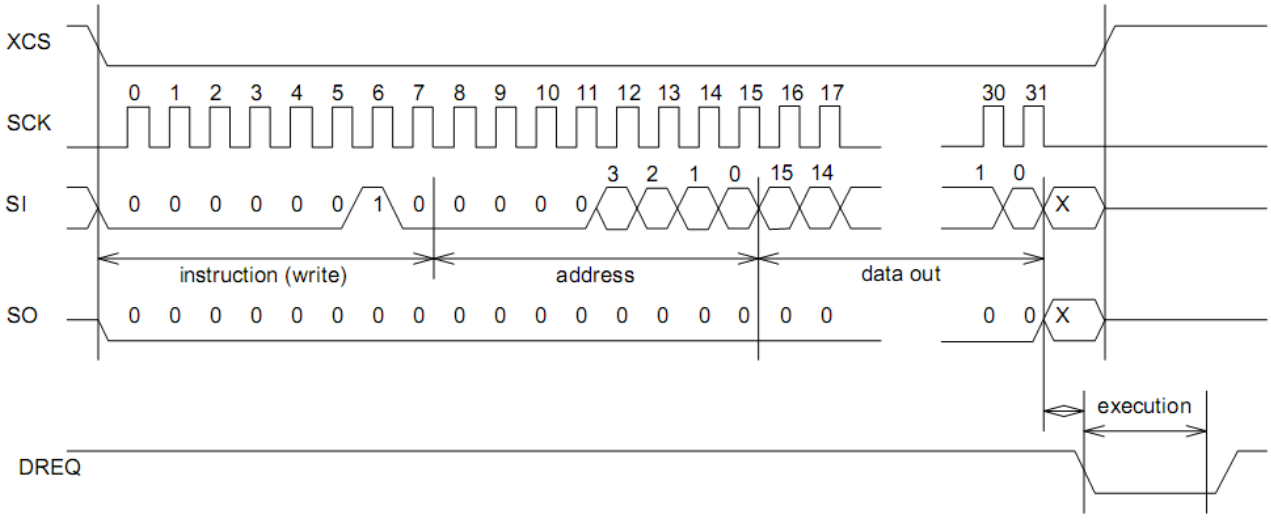


图 7：SCI 字写入操作

写入 VS1053b 寄存器的操作要使用下列顺序，它显示在图 7 中。XCS 信号线先下拉到低电平表示选中该设备。将写操作码 (0x2) 加上 8 位的字地址通过 SI 信号线发送到 VS1053b。

在这个数据字移位发送的最后一个时钟结束之后，XCS 应该上拉到高电平来结束这个写入顺序。

在最后一个位元发送结束后，在这个寄存器被更新的整个期间，DREQ 会被拉到低电平并维持此状态，即图中标识为“execution”的部分。这个时间长短依赖于各寄存器和它所包含的内容（请参阅章节 8.7 中表内的详细描述）。如果该时间长度最大值远超过微控制器可以提供下一条 SCI 命令或 SDI 字节所需的时间，则在下一次 SCI / SDI 的操作开始之前，必须先检查 DREQ 的状态。

译者注：上面这段文字意思隐含着一个意思，即特定条件下，DREQ 在低电平的状态时，VS1053b 也可以接收 SDI 数据和 SCI 指令。这种做法通常是不可靠的，即使某些 SCI 指令确实可以在执行时并不影响下一条 SCI 指令的接收（请参阅章节 7.3 中的最后“译者注”内容描述），但是你可能无法知道这些细节，使用这种不确定的条件，会让你冒上巨大风险。建议：每次操作前，必须检查 DREQ 的状态，确保它是高电平之后，再给 VS1053b 发送 SDI 或 SCI。除非你能完全理解 VS1053b 的工作时序，可以保证在不理会 DREQ 状态的情况下，也知道 SCI 什么时候应该是空闲的。这样的话，利用章节 8.7 中表内的内容确实可以达到这个目的。

7.5.4 SCI 多重写操作

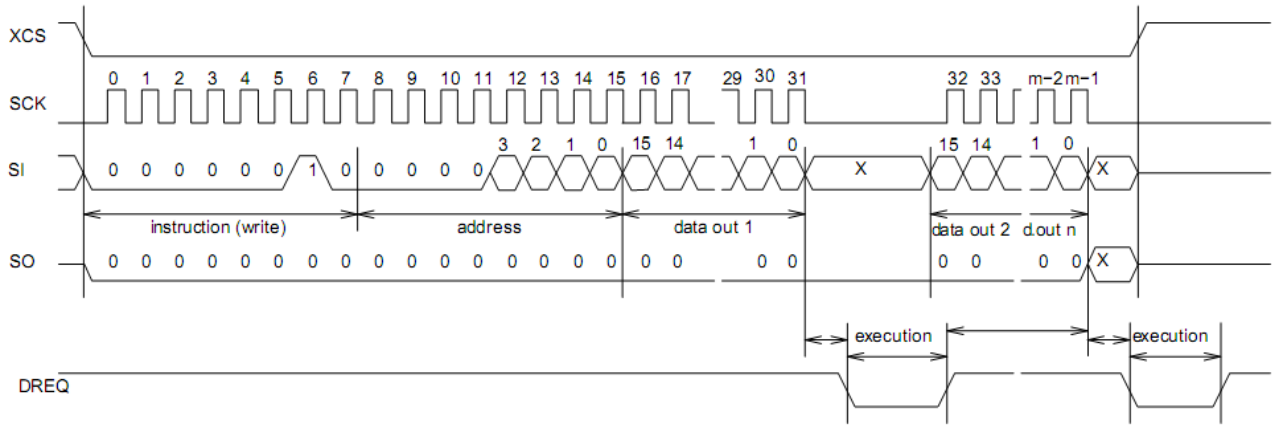


图 8：SCI 多字写操作

如图 8 所示，考虑到用户会发送多个字到同一个 SCI 寄存器，VS1053b 允许快速的 SCI 加载。和单字写入操作的主要区别是：XCS 信号在首个数据字的最后一位发送之后不会立即拉高，这样就可以直接发送下一个数据字。直到最后一个数据字发送之后，才和单字写入操作一样，XCS 被驱动到高电平。

译者注：这样处理的目的就是使后续的数据字不再需要附加写操作码和地址码。

当一个字的最后一个位元发送完之后，DREQ 在寄存器更新的整个期间被驱动到低电平，即上图中标记为“execution”的那一段。这个时间长短依赖于寄存器和它所包含的内容（请参阅章节 8.7 中表内的详细描述）。如果该时间长度最大值远超过微控制器可以提供下一条 SCI 命令或 SDI 字节所需的时间，则在下一次 SCI / SDI 的操作开始之前，必须先检查 DREQ 的状态。

译者注：由图中的时序可以看出，DREQ 在多重写入操作中仍然起着控制作用。虽然后续的数据字可以直接传送到 SCI 总线上，但必须要等 DREQ 重新升高之后才可以开始传送。这里的快速加载仅仅省略了“写操作码”+ “地址码”的部分，而这个 DREQ 的约定绝对不可以忽略。

7.6 SPI 时序图

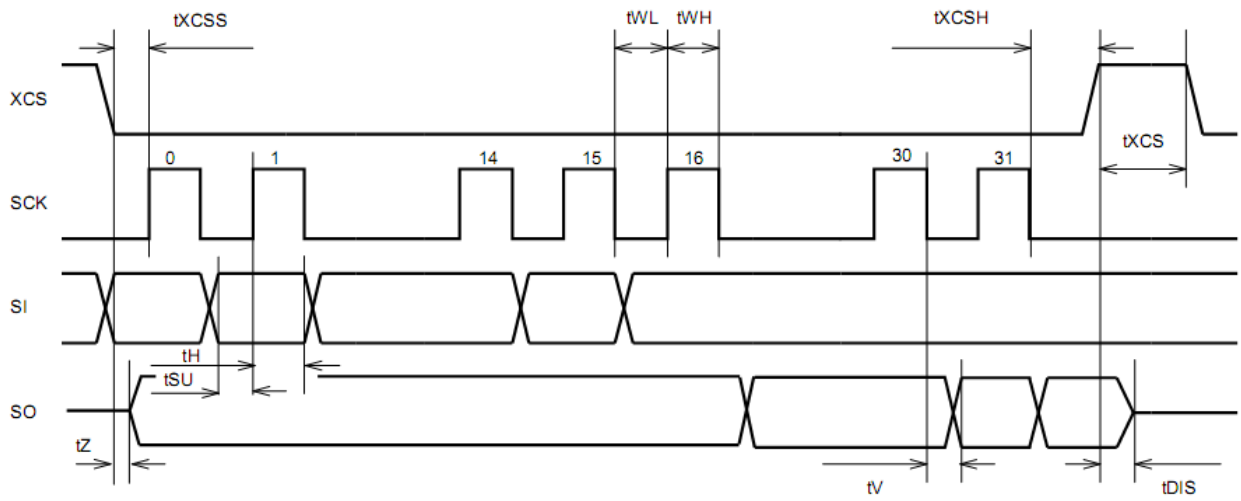


图 9：SPI 时序图

符号	最小	最大	单位
tXCSS	5		ns
tSU	0		ns
tH	2		CLKI 周期
tZ	0		ns
tWL	2		CLKI 周期
tWH	2		CLKI 周期
tV	2 (+ 25 ns[1])		CLKI 周期
tXCSH	1		CLKI 周期
tXCS	2		CLKI 周期
tDIS		10	ns

注 1：25ns 是相对于引脚负载电容为 100pF 时来计算的。当电容值比较小时，这个时间将非常短。

注意：尽管这个定时器是来自于内部时钟 CLKI，但系统在上电时是处于 1.0 倍频模式，即晶振频率（CLKI=XTALI）。当你通过 SCI_CLOCKF 配置了高速时钟和等待 DREQ 变成高电平后，就有高速的 SPI 可用了。

注意：由于 tWL + tWH + tH 是 6×CLKI + 25ns，所以 SCI 的最大读取速度是 CLKI/7。

7.7 SPI 使用 SM_SDINEW 和 SM_SDISHARED 配置的例子

7.7.1 两个 SCI 写操作

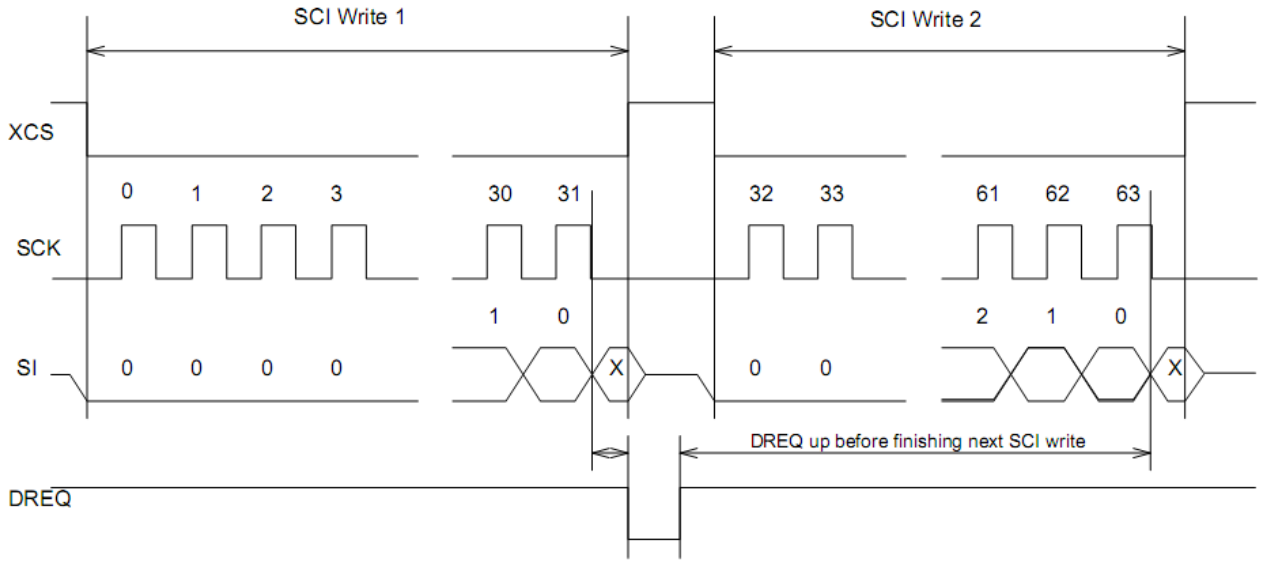


图 10：两个 SCI 的操作

图 10 显示了两个连续的 SCI 操作。注意那个 xCS 信号在两次写入之间必须升高进入失效状态，而且 DREQ 绝对要像图中显示的那样来做参考（译者注：图中 DREQ 信号低电平段后面的那段话意思是 – DREQ 在开始下一个 SCI 之前要升起来）。

7.7.2 两个 SDI 字节

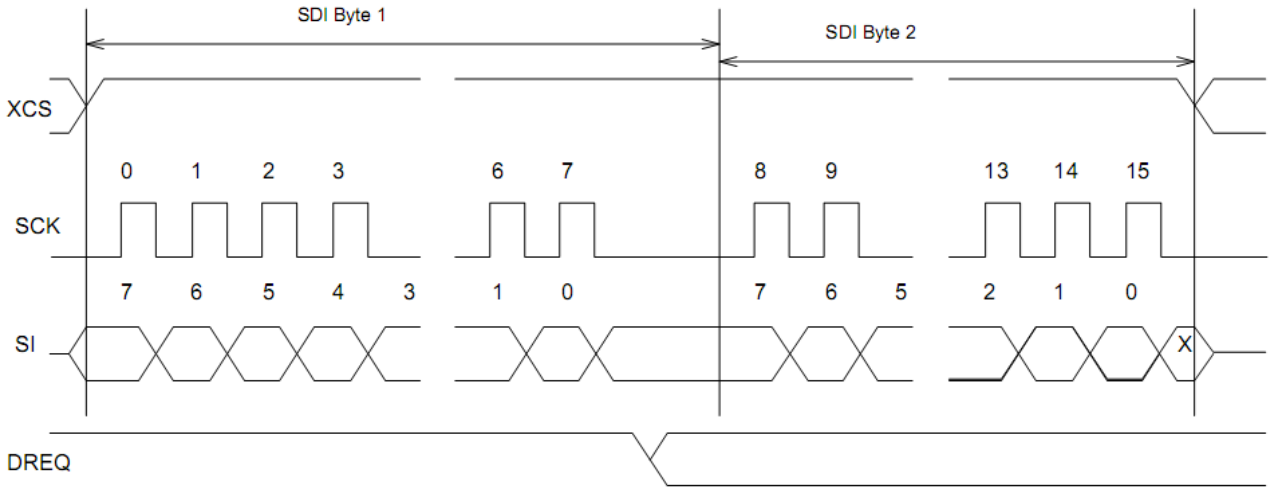


图 11：两个 SDI 字节

如图所示：由于 SDI 数据是在 xCS 的上升沿上同步的。所以，不管在什么情况下，每个字节都不需要再去单独同步了。

7.7.3 在两个 SDI 字节之间的 SCI 操作

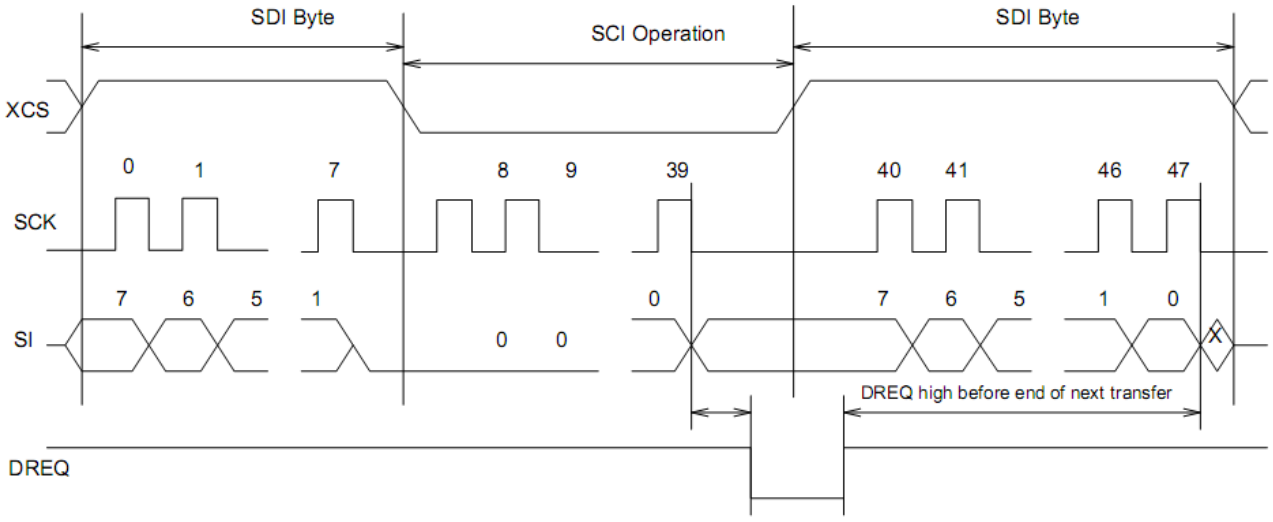


图 12：两个 SDI 字节被一个 SCI 操作分离的图例

图 12 显示了一个 SCI 操作被嵌入到两个 SDI 操作之间的状况。xCS 信号的边沿被用来为 SDI 和 SCI 同步。需要被注意的 DREQ 信号相关细节也显示在图中。（译者注：图中 DREQ 信号低电平段后面的那段话意思是 – DREQ 在下一个传送结束之前为高电平）。

译者注：这里有个前提条件，由于 DREQ 信号是 SDI 和 SCI 共同使用的，所以上图表示：在本章节中提到的操作前，SDI 和 SCI 都是可以接收数据的。在这个条件下，上述的操作才可以正确进行。

8 功能说明

8.1 主要特征

VS1053b 是一个建立在私有基础上的数字信号处理器：VS_DSP。它包含 Ogg Vorbis 、MP3、AAC、WMA 和 WAV PCM + ADPCM 音频解码、MIDI 合成器、串行接口、一个多种速率的立体声 DAC 和模拟输出的放大器加滤波器，和它们所需要的所有相关代码和数据存储器。而且 AD-PCM 音频编码器可以使用一个麦克风放大器和/或线路级的输入，一个立体声 A/D 转换器。还有一个 UART 可以用来调试。

8.2 支持的音频编码格式

定义	
标记	说明
+	支持的格式
?	支持的格式，但没有经过彻底测试
-	已经存在但不支持的格式
	不存在的格式

8.2.1 支持的 MP3 格式（MPEG layer III）

MPEG 1.0[1]:

采样率 / Hz	比特率 / kbit/s													
	32	40	48	56	64	80	96	112	128	160	192	224	256	320
48000	+	+	+	+	+	+	+	+	+	+	+	+	+	+
44100	+	+	+	+	+	+	+	+	+	+	+	+	+	+
32000	+	+	+	+	+	+	+	+	+	+	+	+	+	+

MPEG 2.0[1]:

采样率 / Hz	比特率 / kbit/s													
	8	16	24	32	40	48	56	64	80	96	112	128	144	160
24000	+	+	+	+	+	+	+	+	+	+	+	+	+	+
22050	+	+	+	+	+	+	+	+	+	+	+	+	+	+
16000	+	+	+	+	+	+	+	+	+	+	+	+	+	+

MPEG 2.5[1]:

采样率 / Hz	比特率 / kbit/s													
	8	16	24	32	40	48	56	64	80	96	112	128	144	160
12000	+	+	+	+	+	+	+	+	+	+	+	+	+	+
11025	+	+	+	+	+	+	+	+	+	+	+	+	+	+
8000	+	+	+	+	+	+	+	+	+	+	+	+	+	+

注 1：支持所有的可变比特流（VBR）。

8.2.2 支持的 MP1 格式（MPEG layer I）

注意：阶层 I / II 的解码必须在寄存器 SCI_MODE 中明确使能。（译者注：这是因为受版权限制。）

MPEG 1.0:

采样率 / Hz	比特率 / kbit/s													
	32	64	96	128	160	192	224	256	288	320	352	384	416	448
48000	+	+	+	+	+	+	+	+	+	+	+	+	+	+
44100	+	+	+	+	+	+	+	+	+	+	+	+	+	+
32000	+	+	+	+	+	+	+	+	+	+	+	+	+	+

MPEG 2.0:

采样率 / Hz	比特率 / kbit/s													
	32	48	56	64	80	96	112	128	144	160	176	192	224	256
24000	?	?	?	?	?	?	?	?	?	?	?	?	?	?
22050	?	?	?	?	?	?	?	?	?	?	?	?	?	?
16000	?	?	?	?	?	?	?	?	?	?	?	?	?	?

8.2.3 支持的 MP2 格式（MPEG layer II）

注意：阶层 I / II 的解码必须在寄存器 SCI_MODE 中明确使能。

MPEG 1.0:

采样率 / Hz	比特率 / kbit/s													
	32	48	56	64	80	96	112	128	160	192	224	256	320	384
48000	+	+	+	+	+	+	+	+	+	+	+	+	+	+
44100	+	+	+	+	+	+	+	+	+	+	+	+	+	+
32000	+	+	+	+	+	+	+	+	+	+	+	+	+	+

MPEG 2.0:

采样率 / Hz	比特率 / kbit/s													
	8	16	24	32	40	48	56	64	80	96	112	128	144	160
24000	+	+	+	+	+	+	+	+	+	+	+	+	+	+
22050	+	+	+	+	+	+	+	+	+	+	+	+	+	+
16000	+	+	+	+	+	+	+	+	+	+	+	+	+	+

8.2.4 支持的 Ogg Vorbis 格式

参数	最小	最大	单位
通道数量		2	
窗口大小	64	4096	samples
采样率		48000	Hz
比特率		500	kbit/sec

仅支持 floor1，目前没有已知的编码器还在使用 floor0。所有的单通道和双通道 ogg vorbis 文件都应该用此解码器播放。

8.2.5 支持的 AAC 格式（ISO/IEC 13818-7 和 ISO/IEC 14496-3）

VS1053b 解码器可以支持 MPEG2-AAC-LC-2.0.0.0 和 MPEG4-AAC-LC-2.0.0.0 流，也就是低复杂度和最大两个通道可以解码。如果一个流中包含了一种素材以上和/或多种素材类型，则你可以选择在 16 个单通道、16 个通道对和 16 个低频素材中的一个来解码，缺省情况下会选择流中首先出现的素材。

用户可控制的动态范围控制（DRC）能让用户限制或增强素材中可以用来实现 DRC 的信息。

正弦窗和 Kaiser-Bessel-derived 窗（KBD）均可支持（译者注：这个 Kaiser-Bessel-derived window 可能也被称作“凯撒-贝塞耳-派生窗”，因找不到确切的译法，这里还是用原文比较合适）。

对于 MPEG4 的伪随机噪音替代（PNS）是支持的，但不支持短帧（120 和 960 个样本）。

频带复制（SBR）第 3 级和参量立体声（PS）第 3 级均支持（HE-AAC v2）。这个第 3 级是指可支持最大两个通道和最高采样率为 48KHz，包括和不包括 SBR、有 PS 或没有 PS。另外，混频模式（*Ra* 和 *Rb*），IPD/OPD 合成和 34 个频段的协议都可实现。“降低采样”合成（downsampled synthesis）模式（使用 SBR 的核心编码采样率 >24kHz 和 <=48kHz ）也可实现了。

SBR 和 PS 的解码功能可以被关闭，也有不同的操作模式可以选择。请参考章节 9.11 中的 config1 和 sbrAndPsStatus：“附加的参数” 。

如果（SBR 和 PS）被使能和 AAC 解码，则必须要更高的时钟频率，而内部时钟（CLKI）会自动增加。对于能正确解码的要求来说，如果内部时钟过慢，则 PS 和 SBR 操作会被自动关掉。通常 HE-AAC v2 需要 4.5x 的时钟来解码包含了 SBR 和 PS 的文件。这就是为什么 3.5x+1.0x 的时钟配置被推荐的原因。

AAC 流的 ADTS 格式是被推荐的。因为它容易做到再同步，此格式能简单的实现快放和倒带。

除了 ADTS（.aac）外，MPEG2 ADIF（.aac）和 MPEG4 音频（.mp4 / .m4a）文件都可以播放，但是这些格式不太适合用于倒带和快进的操作。你仍然可以通过它们的安全跳跃点表格的特性来实现，或使用不太健全，但比较容易重新同步的机制（请参阅章节 9.5.4）。

因为 3GPP（.3gp）和 3GPPv2（.3g2）文件恰好是 MPEG4 文件，那些只有包含了 HE-AAC 和 HE-AACv2 内容的文件才可以播放。

注意：要播放“.3gp”、“.3g2”、“.mp4”和“.m4a”文件，这个文件中的**媒体数据（mdat）**原子必须是最后的原子。因为 VS1053b 要接收流中的所有数据，所有的元数据必须是在音乐数据被使用之前被接收。有些 MP4 文件的格式可能达不到这个要求，那么它们就需要转换成所要求的格式。这就是为什么流化的 ADTS 格式被推荐的原因。

已经有程序可以优化“.mp4”和“.m4a”，将其变成媒体数据是最终媒体数据原子的流化格式文件，以适合 web 服务器的音频传送。你还可以使用一些工具来处理 Vs1053b 的文件，例如：mp4creator -optimize file.mp4 。

AAC[1、2]:

采样率 / Hz	最大比特率 kbit/s – 双通道								
	≤96	132	144	192	264	288	384	529	576
48000	+	+	+	+	+	+	+	+	+
44100	+	+	+	+	+	+	+	+	
32000	+	+	+	+	+	+	+		
24000	+	+	+	+	+	+			
22050	+	+	+	+	+				
16000	+	+	+	+					
12000	+	+	+						
11025	+	+							
8000	+								

注 1：64000Hz、88200Hz 和 96000Hz 的 AAC 文件也许是在最高的采样率播放（12.288MHz 晶振下的 48000Hz）。

注 2：同样，所有的可变比特率（VBR）格式也是支持的。注意那张表中提供了 AAC 规格书中的双通道所允许的最大比特率的明确采样率定义，而这个解码器实际上并没有上下限的制约。

8.2.6 支持的 WMA 格式

Windows 媒体音频编码版本 2, 7, 8,和 9 是可支持的。所有的 WMA 特性（L1、 L2 和 L3）都可以支持。早期的流在类别 1、2a、2b 和 3 中已经被分开了。这个解码器通过了符合 Microsoft 的测试程序测试。而 Windows 媒体音频专业版使用了其它的编解码器，它是不支持的。

WMA 4.0 / 4.1:

采样率 / Hz	比特率 / kbit/s																
	5	6	8	10	12	16	20	22	32	40	48	64	80	96	128	160	192
8000	+	+	+		+												
11025			+	+													
16000				+	+	+	+										
22050						+	+	+	+								
32000							+	+	+	+	+	+					
44100									+		+	+	+	+	+	+	
48000															+	+	

WMA 7:

采样率 / Hz	比特率 / kbit/s																
	5	6	8	10	12	16	20	22	32	40	48	64	80	96	128	160	192
8000	+	+	+		+												
11025			+	+													
16000				+	+	+	+										
22050						+	+	+	+								
32000							+		+	+	+						
44100									+		+	+	+	+	+	+	+
48000															+	+	

WMA 8:

采样率 / Hz	比特率 / kbit/s																
	5	6	8	10	12	16	20	22	32	40	48	64	80	96	128	160	192
8000	+	+	+		+												
11025			+	+													
16000				+	+	+	+										
22050						+	+	+	+								
32000							+		+	+	+						
44100									+		+	+	+	+	+	+	+
48000															+	+	+

WMA 9:

采样率 / Hz	比特率 / kbit/s																		
	5	6	8	10	12	16	20	22	32	40	48	64	80	96	128	160	192	256	320
8000	+	+	+		+														
11025			+	+															
16000				+	+	+	+												
22050						+	+	+	+										
32000							+		+	+	+								
44100							+		+		+	+	+	+	+	+	+	+	+
48000												+		+	+	+	+		

在可用的附加 WMA 解码结构特性中，所有其它比特率和采样率的结合体也是被支持的，包括可变比特率的 WMA 流。注意 WMA 不能像 MP3 那样访问均匀的比特流数据，所以你必须要具有高峰值传送的能力，才能完全地播放相同的比特率（而不会出现断续的现象）。

8.2.7 支持的 RIFF WAV 格式

更多被支持的单通道或双通道音频的普通 RIFF WAV 子格式。

格式	名称	支持	注释
0x01	PCM	+	16 和 8 位，任何采样率 ≤ 48kHz
0x02	ADPCM	-	
0x03	IEEE_FLOAT	-	
0x06	ALAW	-	
0x07	MULAW	-	
0x10	OKI_ADPCM	-	
0x11	IMA_ADPCM	+	任何采样率 ≤ 48kHz
0x15	DIGISTD	-	
0x16	DIGIFIX	-	
0x30	DOLBY_AC2	-	
0x31	GSM610	-	
0x3b	ROCKWELL_ADPCM	-	
0x3c	ROCKWELL_DIGITALK	-	
0x40	G721_ADPCM	-	
0x41	G728_CELP	-	
0x50	MPEG	-	
0x55	MPEGLAYER3	+	关于支持的 MP3 模式，请参阅章节8.2.1
0x64	G726_ADPCM	-	
0x65	G722_ADPCM	-	

8.2.8 支持的 MIDI 格式

通用的 MIDI 和 SP-MIDI 格式 0 文件是能播放的，而格式 1 和 2 文件必须由用户自己转换成 0 格式的。最多的复音数为 64，而最大持续的复音数是 40。实际上，复音数是依赖内部的时钟速度（这是用户可选择的）、乐器的用法、是否使用了混响效果和可能存在的整体性后期处理效果。例如低音和高音的加强、EarSpeaker 效果的处理等等。如果还存在 SP-MIDI MIP 表和使用了滑音消除，则制作复音的算法将会受到限制。

43MHz（3.5x 输入的时钟）能达到同时产生 19-31 个持续的音符，而音符的瞬间数量可以很多。这是为了在功耗和品质之间作平衡而采取的折中办法，如果提高时钟频率则可以增加复音数。

混响效果可以由用户来控制。除了自动混响和混响关闭模式之外，有 14 种不同的延时可以选择，这大致相当于不同大小的房间。此外，每首 midi 音乐中在每种乐器上获得的效果也会决定造成多大影响。由于混响效果会使用大约 4MHz 的处理能力，所以自动混响效果控制只有在最少 3.0x 的时钟下才能使能。

在 VS1053b 中，EarSpeaker 效果和 MIDI 混响是可以同时使用的。这样用头挂式耳机来听 MIDI 歌曲还是比较理想的。

除了 VS1003 能使用的 36 种乐器之外，有新乐器可以使用了。VS1053b 现在可以设置 GM1 完整乐器组中特有的乐器和 GM2 打击乐器组中的一组音色库。

VS1053b 旋律乐器组 (GM1)			
1 Acoustic Grand Piano	33 Acoustic Bass	65 Soprano Sax	97 Rain (FX 1)
2 Bright Acoustic Piano	34 Electric Bass (finger)	66 Alto Sax	98 Sound Track (FX 2)
3 Electric Grand Piano	35 Electric Bass (pick)	67 Tenor Sax	99 Crystal (FX 3)
4 Honky-tonk Piano	36 Fretless Bass	68 Baritone Sax	100 Atmosphere (FX 4)
5 Electric Piano 1	37 Slap Bass 1	69 Oboe	101 Brightness (FX 5)
6 Electric Piano 2	38 Slap Bass 2	70 English Horn	102 Goblins (FX 6)
7 Harpsichord	39 Synth Bass 1	71 Bassoon	103 Echoes (FX 7)
8 Clavi	40 Synth Bass 2	72 Clarinet	104 Sci-fi (FX 8)
9 Celesta	41 Violin	73 Piccolo	105 Sitar
10 Glockenspiel	42 Viola	74 Flute	106 Banjo
11 Music Box	43 Cello	75 Recorder	107 Shamisen
12 Vibraphone	44 Contrabass	76 Pan Flute	108 Koto
13 Marimba	45 Tremolo Strings	77 Blown Bottle	109 Kalimba
14 Xylophone	46 Pizzicato Strings	78 Shakuhachi	110 Bag Pipe
15 Tubular Bells	47 Orchestral Harp	79 Whistle	111 Fiddle
16 Dulcimer	48 Timpani	80 Ocarina	112 Shanai
17 Drawbar Organ	49 String Ensembles 1	81 Square Lead (Lead 1)	113 Tinkle Bell
18 Percussive Organ	50 String Ensembles 2	82 Saw Lead (Lead)	114 Agogo
19 Rock Organ	51 Synth Strings 1	83 Calliope Lead (Lead 3)	115 Pitched Percussion
20 Church Organ	52 Synth Strings 2	84 Chiff Lead (Lead 4)	116 Woodblock
21 Reed Organ	53 Choir Aahs	85 Charang Lead (Lead 5)	117 Taiko Drum
22 Accordion	54 Voice Oohs	86 Voice Lead (Lead 6)	118 Melodic Tom
23 Harmonica	55 Synth Voice	87 Fifths Lead (Lead 7)	119 Synth Drum
24 Tango Accordion	56 Orchestra Hit	88 Bass + Lead (Lead 8)	120 Reverse Cymbal
25 Acoustic Guitar (nylon)	57 Trumpet	89 New Age (Pad 1)	121 Guitar Fret Noise
26 Acoustic Guitar (steel)	58 Trombone	90 Warm Pad (Pad 2)	122 Breath Noise
27 Electric Guitar (jazz)	59 Tuba	91 Polysynth (Pad 3)	123 Seashore
28 Electric Guitar (clean)	60 Muted Trumpet	92 Choir (Pad 4)	124 Bird Tweet
29 Electric Guitar (muted)	61 French Horn	93 Bowed (Pad 5)	125 Telephone Ring
30 Overdriven Guitar	62 Brass Section	94 Metallic (Pad 6)	126 Helicopter
31 Distortion Guitar	63 Synth Brass 1	95 Halo (Pad 7)	127 Applause
32 Guitar Harmonics	64 Synth Brass 2	96 Sweep (Pad 8)	128 Gunshot

VS1053b 打击乐器组 (GM1+GM2)			
27 High Q	43 High Floor Tom	59 Ride Cymbal 2	75 Claves
28 Slap	44 Pedal Hi-hat [EXC 1]	60 High Bongo	76 Hi Wood Block
29 Scratch Push [EXC 7]	45 Low Tom	61 Low Bongo	77 Low Wood Block
30 Scratch Pull [EXC 7]	46 Open Hi-hat [EXC 1]	62 Mute Hi Conga	78 Mute Cuica [EXC 4]
31 Sticks	47 Low-Mid Tom	63 Open Hi Conga	79 Open Cuica [EXC 4]
32 Square Click	48 High Mid Tom	64 Low Conga	80 Mute Triangle [EXC 5]
33 Metronome Click	49 Crash Cymbal 1	65 High Timbale	81 Open Triangle [EXC 5]
34 Metronome Bell	50 High Tom	66 Low Timbale	82 Shaker
35 Acoustic Bass Drum	51 Ride Cymbal 1	67 High Agogo	83 Jingle bell
36 Bass Drum 1	52 Chinese Cymbal	68 Low Agogo	84 Bell tree
37 Side Stick	53 Ride Bell	69 Cabasa	85 Castanets
38 Acoustic Snare	54 Tambourine	70 Maracas	86 Mute Surdo [EXC 6]
39 Hand Clap	55 Splash Cymbal	71 Short Whistle [EXC 2]	87 Open Surdo [EXC 6]
40 Electric Snare	56 Cowbell	72 Long Whistle [EXC 2]	
41 Low Floor Tom	57 Crash Cymbal 2	73 Short Guiro [EXC 3]	
42 Closed Hi-hat [EXC 1]	58 Vibra-slap	74 Long Guiro [EXC 3]	

8.3 VS1053b 的数据流动原理

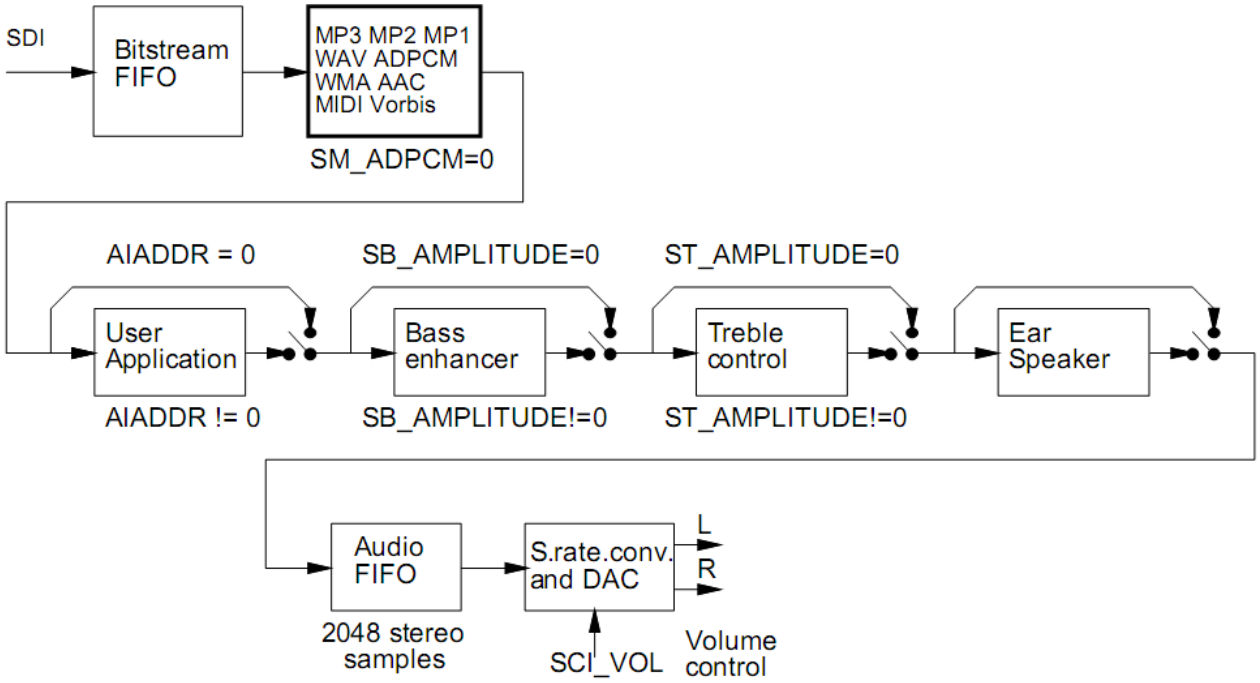


图 13：VS1053b 的数据流动图

首先取决于音频数据，和先假设 ADPCM 编码模式没有被设置，SDI 总线上的 Ogg Vorbis、MP3、WMA、AAC、PCM WAV、IMA ADPCM WAV 或 MIDI 数据已经被确认和解码。

在解码后，如果 SCI_AIADDR 是非 0，这个寄存器中的地址指向的应用程序代码将开始执行。对于更加详细的描述，请参考“Application Notes for VS10XX”（即：VS10XX 的应用注意事项）。

数据将会发送到低音增强和高音控制单元，根据 SCI_BASS 寄存器的设置进行处理。

下一步，则是执行头挂式耳机的处理。如果 EarSpeaker 效果被激活，则会执行真实空间模拟的处理。

之后，那些数据送到音频 FIFO 缓存，直到这些数据通过音频中断被读取和进行“采样率转换”并送至 DAC 组。音频的 FIFO 大小是 2048 立体声（2 X 16 比特）采样数，即 8KiB。

上述的“采样率转换”是指将所有不同的采样率转换到 XTALI / 2，或者是 18 比特精度下的可用采样率的 128 倍那么高。音量控制是在“增加采样”（upsampled）的区域内执行的。新的音量设置是在“增加采样”的信号穿越零位时加载（或在超时之后）。这个“过零交叉”的侦测功能几乎可以消除音量在改变时所产生的所有音频噪音。

这种采样率转换到一个共同的采样率的迁移动作，需要一个用固定输入时钟频率、以复杂的 PLL 为基础的时钟，来使它能够达到几乎无限制的采样率精度。对于一个 12.288MHz 的时钟来说，这个 DA 转换器运作在 128 X 48KHz 上，即 6.144MHz 和创建同相的立体声模拟信号。这个超采样输出的低通滤波器是一个在片的模拟滤波器。随后，信号再被送至耳机功率放大器。

8.4 EarSpeaker 的空间效果处理

在使用头挂式耳机时，声音方向会趋向于听者的头脑内部，使声音扁平化，从而失去空间的感觉。这是一种不自然、尴尬和令人烦恼的状况。这种现象在文献中通常称之为‘单侧化’（lateralization），意思是‘在头部内’本地化。长时间听这种侧面化的声音可能会造成听觉疲劳。

现实中的音源是来自外部的，声波在经过耳朵的通道传送到耳鼓膜时会留下踪迹，大脑的听觉系统会用这些踪迹来判断每个音源的距离和角度。在用音箱听声音的时候，那些声波的踪迹是来自外部的。而头挂式耳机收听会使这些踪迹丢失或混淆不清。

EarSpeaker 效果处理声音时会通过头挂式耳机制造出接近来自真实音箱或现场音乐的音乐效果。一旦 EarSpeaker 效果被激活，能感觉到这些乐器从头脑的内部移动到外部，可以使听者轻易地区分出那些不同的乐器（请参阅图 14）。这样就使听觉的感受变得更加自然和舒适，听者对立体声映像的感受更象在广泛的空间里，而不是局限在头脑的内部。

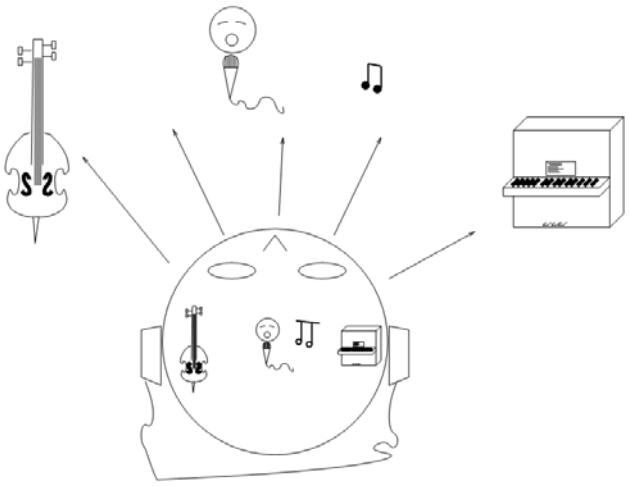


图 14：EarSpeaker 效果扩展的音源和普通头挂式耳机的音效对比图

注意 EarSpeaker 效果和其它普通的空间处理效果是不同的，像回声、混响或低音加强之类等等。EarSpeaker 的模拟效果是建立在人的听觉模型和真实环境声学上的。因此它不会使用人工干预音调的方法来改变音乐的本质。

EarSpeaker 效果处理能通过参数实现稍微有些差别的模式，来模拟每种听觉环境上有细微区分的类型，这样可以适应不同的个人喜好和录音类型。请参阅章节 8.7.1 中如何激活不同的模式。

- *关闭*：较好的收听方法是通过音箱或音频在播放时已经使用了双耳用的预处理。
- *最少*：适合用耳机欣赏普通的大众音乐，非常微妙的。
- *正常*：适合用耳机欣赏普通的大众音乐，和 *最少* 模式比，音源将移到更远的地方。
- *强悍*：适合旧的或干涩的录音或播放人工合成的虚假的声音，例如 MIDI 产生的声音。

8.5 串行数据接口（SDI）

VS1053b 的串行数据接口被指定用来传送不同解码器的压缩数据。

如果输入解码器的数据是残缺的或者它的接收不能很快的被确认，则模拟输出将会被自动静音。

有几个不同的测试可以通过 SDI 来激活，请参见章节 9 中的描述。

8.6 串行控制接口（SCI）

这个串行控制接口兼容 SPI 总线接口中的那些说明。数据传送总是 16 位的。VS1053b 就是通过此接口的读取和写入来控制的。

主控制器可以用此控制接口来进行下列操作：

- 操作模式、时钟和内建效果的控制；
- 访问状态信息和数据标头的数据；
- 访问录音模式下的编码数据；
- 上载和控制用户程序。

8.7 SCI 寄存器

当 VS1053b 侦测到一个 SCI 操作时会设置 DREQ 为低（这个延时是 16 到 40 个 CLKI 周期，它取决于是否有一个中断服务程序被激活），等处理完这个操作后再恢复它。而处理的时间长短倚赖于当前的操作。如果 DREQ 在一个 SCI 操作执行时就是低，则在这个 SCI 操作之后它依然会保持为低（参见章节 7.3 中的译者注，这是 SDI 和 SCI 同时操作时可能会出现的情况）。

如果在一个 SCI 操作之前 DREQ 为高[注]，在 DREQ 再次返回高之前请不要开始一个新的 SCI/SDI 操作。如果在一个 SCI 操作之前 DREQ 为低，由于 SDI 不能接收更多的数据，一定要确认在这个再次发送之前有足够的时间来完成这个操作。

SCI 寄存器组，前缀SCI_					
寄存器	类型	复位值	时间长度[1]	简称[位]	说明
0x0	rw	0x4800	80 CLKI[4]	MODE	模式控制
0x1	rw	0x000C[3]	80 CLKI	STATUS	VS1053b的状态
0x2	rw	0	80 CLKI	BASS	内建的低音/高音控制
0x3	rw	0	1200 XTALI[5]	CLOCKF	时钟频率加乘数
0x4	rw	0	100 CLKI	DECODE_TIME	解码时间长度（秒）
0x5	rw	0	450 CLKI[2]	AUDATA	各种音频数据
0x6	rw	0	100 CLKI	WRAM	RAM 写/读
0x7	rw	0	100 CLKI	WRAMADDR	RAM 写/读的基址
0x8	r	0	80 CLKI	HDAT0	流的数据标头 0
0x9	r	0	80 CLKI	HDAT1	流的数据标头 1
0xA	rw	0	210 CLKI[2]	AIADDR	应用程序的起始地址
0xB	rw	0	80 CLKI	VOL	音量控制
0xC	rw	0	80 CLKI[2]	AICTRL0	应用程序控制寄存器 0
0xD	rw	0	80 CLKI[2]	AICTRL1	应用程序控制寄存器 1
0xE	rw	0	80 CLKI[2]	AICTRL2	应用程序控制寄存器 2
0xF	rw	0	80 CLKI[2]	AICTRL3	应用程序控制寄存器 3

注 1：这是指写入该寄存器后，在最坏情况下 DREQ 停留在低电平的时间长度。对于少于 100 个时钟周期的执行，用户可以在写入这些寄存器后可以用一个固定的延时来替代并跳过 DREQ 检查。

注 2：另外，这些周期必须还要加上消耗在用户应用程序上的时间。

注 3：固件会直接将寄存器的值改为 0x48（模拟单元使能），并在很短的时间内再改为 0x40（模拟驱动单元使能）。

注 4：当模式寄存器写入了指定的软件复位时，最坏情况下的时间长度是 22000 个 XTALI 周期。

注 5：写入 CLOCKF 寄存器将强制内部时钟在一段时间内运行在 1.0×的晶振频率状态下。在它的更新还处在进行中的时候发送 SCI 或 SDI 的那些数据位可不是好主意。

除了从 AIADDR 读取需要 200 个周期外，其它所有 SCI 寄存器读取时间都小于 100 个周期。此外，在读取 AIADDR、AUDATA 和 AICTRL0…3 这些寄存器的时候，必须要将用户应用程序所消耗的周期也加上。

注：这个意思可能是指高速主控制器在发送 SCI 之后，VS1053b 的 DREQ 信号需要一个延时才能变低，在此期间，高速主控制器应该等待 DREQ 变低，然后才能确定 VS1053b 将当前发送的 SCI 执行了，之后再等待 DREQ 变高，才能确定下一个 SCI/SDI 可以被 VS1053b 接收。

8.7.1 SCI_MODE (RW)

SCI_MODE 用来控制 VS1053b 的运作，它的缺省值是 0x0800（SM_SDINEW 被设定）。

位元	名称	功能	值	说明
0	SM_DIFF	差分	0 1	正常的同相音频 左通道反相
1	SM_LAYER12	允许 MPEG layers I & II	0 1	不允许 允许
2	SM_RESET	软件复位	0 1	不用复位 复位
3	SM_CANCEL	取消当前的文件解码	0 1	不取消 取消
4	SM_EARSPEAKER_LO	EarSpeaker 低设定	0 1	关闭 激活
5	SM_TESTS	允许 SDI 测试	0 1	不允许 允许
6	SM_STREAM	流模式	0 1	不是 是
7	SM_EARSPEAKER_HI	EarSpeaker 高设定	0 1	关闭 激活
8	SM_DACT	DCLK 的有效边沿	0 1	上升沿 下降沿
9	SM_SDIORD	SDI 位顺序	0 1	MSb 在前 MSb 在后
10	SM_SDISHARE	共享 SPI 片选	0 1	不共享 共享
11	SM_SDINEW	VS1002 本地 SPI 模式	0 1	非本地模式 本地模式
12	SM_ADPCM	ADPCM 录音激活	0 1	不激活 激活
13	-	-	0 1	正确的 错误的
14	SM_LINE1	咪 / 线路1 选择	0 1	MICP LINE1
15	SM_CLK_RANGE	输入时钟范围	0 1	12..13 MHz 24..26 MHz

当 SM_DIFF 被设置，将会使左通道的输出反相播放。对于立体声输入这会创建一个虚拟环绕效果，而单声道输入将会创建一个差分的左/右信号。

SM_LAYER12 将会使能除 MPEG 1.0 和 2.0 的阶层 III 之外的阶层 I 和 II 的解码。**如果你使能阶层 I 和 II 的解码，则需要自己解决引发的任何专利问题。**MPEG 1.0/2.0 的阶层 III 的关联许可证并没有包含阶层 I 和 II 的任何专利。

软件复位是通过设置 SM_RESET 为 1 来启动的，这个位将自动清除。

如果你想终止一个解码的过程，则设置 SM_CANCEL，并且要遵守 DREQ 的约定继续发送数据。当 SM_CANCEL 被一个编解码器侦测到，它将停止解码并返回主循环。流缓冲器中的内容将被抛弃，SM_CANCEL 也会被清除。SCI_HDAT1 也同时会被清除。请参阅章节 9.5.2 的详细资料。

位 SM_EARSPEAKER_LO 和 SM_EARSPEAKER_HI 用于控制 EarSpeaker 的空间效果处理。如果两个位均为 0，则这个效果将会被关闭。其它组合将会激活 EarSpeaker 效果处理，并有 3 个不同的级别：LO = 1，HI = 0 选择 *最少* 模式；LO = 0，HI = 1 选择 *正常* 模式；LO = 1，HI = 1 选择 *强悍* 模式。EarSpeaker 在 44.1 kHz 的采样率时大约需要 12MIPS。

如果 SM_TESTS 被设置，SDI 测试将被允许。关于更多 SDI 测试的详细资料，请看章节 9.12。

SM_STREAM 是激活 VS1053b 的流模式。在这个模式里，数据的发送应该有个均匀的间隔，最好是少于 512 字节的块，而 VS1053b 将会试图修改它的播放速度最多到 5%来使输入缓冲区保持一半是满的。对于最佳的声音品质，平均速率误差应该在 0.5%以内，比特率不应该超出 160kbit/s 和不使用 VBR（可变比特率）。有关的细节，请参阅 VS10xx 的应用注意事项。这个模式唯一工作在 MP3 和 WAV 文件类型下。

SM_DACT 定义了 SDI 数据在时钟的哪个边沿有效。当它为‘0’时，数据在上升沿上读取，当它为‘1’时，数据在下降沿上读取。

SM_SDIORD 被清除的时候，SDI 上发送的字节是 MSb 在前的，反之，用户可以调转 SDI 上的比特顺序，即“比特 0”被先确认，而“比特 7”（MSb）在最后。但是，字节组的字节顺序还是按照缺省的顺序来发送。这个寄存器位对 SCI 总线没有影响。

假如 SM_SDISHARE 被设置，则设定了 SCI 和 SDI 共享一个片选信号，在章节 7.2 中有相关的解释。

设置 SM_SDINEW 将激活在章节 7.2.1 和 7.4.2 中描述的 VS1002 系列的本地模式。注意：这个位在 VS1053b 启动的时候将被缺省设置。

通过在同一时间内设置 SM_ADPCM 和 SM_RESET，用户可以激活 activate IMA ADPCM 录音模式（请参阅章节 9.8）。

SM_LINE_IN 是用来选择 ADPCM 录音的左通道输入。如果为‘0’，差分咪的输入引脚 MICP 和 MICN 会被使用。反之为‘1’的话，则线路级的 MICP/LINEIN1 引脚被使用。

SM_CLK_RANGE 激活晶振输入的时钟分频器。当 SM_CLK_RANGE 被设定时，这个时钟会是输入频率除 2。对于这个芯片来说，24MHz 将变成合适的 12MHz。如果确实需要设置 SM_CLK_RANGE，则在 VS1053b 复位之后要尽快设置它。

8.7.2 SCI_STATUS (RW)

SCI_STATUS 包含了 VS1053b 当前状态的信息。它也控制一些低级的事物，用户通常无需关心。

名称	位域	说明
SS_DO_NOT_JUMP	15	解码中的标头，不能快进/倒退
SS_SWING	14:12	设置摆幅为 +0 dB，+0.5 dB，...，或+3.5 dB
SS_VCM_OVERLOAD	11	GBUF 超负荷标记 '1' = 超负荷
SS_VCM_DISABLE	10	GBUF 超负荷侦测 '1' = 关闭
	9:8	保留
SS_VER	7:4	版本
SS_APDOWN2	3	模拟驱动单元掉电
SS_APDOWN1	2	模拟单元内部掉电
SS_AD_CLOCK	1	AD 时钟选择，'0' = 6 MHz, '1' = 3 MHz
SS_REFERENCE_SEL	0	参考电压选择， '0' = 1.23 V, '1' = 1.65 V

如果 SS_DO_NOT_JUMP 被设定则表示：WAV、Ogg Vorbis、WMA、MP4 或 AAC-ADIF 标头正在解码，和在这个文件内是不允许跳到另外一个位置的。

如果 AVDD 最少是在 3.3V 以上，可以通过 SS_REFERENCE_SEL 来选择 1.65V 的参考电压来增加模拟输出的摆幅。

如果时钟输入频率/2 还是太高，可以设置 SS_AD_CLOCK 来 2 分频作为 AD 调整工作频率。

SS_VER 对于各个芯片来说：0=VS1001、1=VS1011、2=VS1002、3=VS1003、4=VS1053、5=VS1033 和 7=VS1103。

SS_APDOWN2 可控制模拟驱动单元的掉电。SS_APDOWN1 可控制内部的模拟单元掉电。它们只是打算给系统固件使用的。

如果用户想要 VS1053b 掉电和一个很小的电源关闭瞬间，设置 SCI_VOL 到 0xffff，然后在激活复位之前至少要等待几个毫秒。

VS1053b 包含了 GBUF 的保护电路，在有过大的电流来断开 GBUF 驱动，表示短路到地线上了。SS_VCM_OVERLOAD 为高表示过载被侦测到了，而设置 SS_VCM_DISABLE 可以关掉这个保护功能。

SS_SWING 允许你去超过这个 0 dB 的音量设定。值为 0 表示正常模式，1 是提供+0.5 dB 和 2 提供+1.0 dB。尽管这个范围最大可以设置到 7。但大于 2 的设置并不起作用，所以不应该使用大于 2 的值。

注意：由于 VS1053b 固件的一个 bug，音量计算程序会清除 SS_AD_CLOCK 和 SS_REFERENCE_SEL 位。写入 SCI_STATUS 或 SCI_VOLUME，和采样率修改（如果低音增强或高音控制是激活的）会导致音量计算程序被调用。有个应急的方法：你可以在每次音量改变之后，通过 SCI_WRAMADDR 和 SCI_WRAM 来写入 SCI_STATUS。写 0xc001 到 SCI_WRAMADDR，接着写这个值到 SCI_WRAM。然而，这些模式之间的表现差异并非很大，所以使用缺省模式会更加容易一些。

8.7.3 SCI_BASS (RW)

名称	位域	说明
ST_AMPLITUDE	15:12	高音控制，步长为 1.5 dB (-8..7, 0 = 关闭)
ST_FREQLIMIT	11:8	下限频率，步长为 1000 Hz (1..15)
SB_AMPLITUDE	7:4	低音增强，步长为 1 dB (0..15, 0 = 关闭)
SB_FREQLIMIT	3:0	频率上限[注]，步长为 10 Hz (2..15)

这个低音加强 VSBE 是一个强劲的低音增强 DSP 算法，它将试着让用户的耳机获得尽量多的输出又不会产生削波失真。

VSBE 是通过将 SB_AMPLITUDE 设置为非 0 来激活的。SB_AMPLITUDE 可以按照用户的喜好来设定，并且由 SB_FREQLIMIT 设定的用户音频系统中的低频段得到大约 1.5 倍的再生。举例来说：设定 SCI_BASS 为 0x00f6 将会使 60 Hz 以下的频率得到 15 dB 的提升。

注意：因为 VSBE 要设法避免削波失真，它才能提供最佳的低音增强和动态的音乐素材，还有，请不要将播放的音量设定到最大。另外它不能创造出低音：这个声源中首先必须要有低音。

高音控制 VSTC 是在 ST_AMPLITUDE 非 0 时激活的。举例来说：设置 SCI_BASS 为 0x7a00 时，将会使 10 KHz 以上的高频得到 10.5 dB 的提升。

在 44100 Hz 的采样率下，低音增强使用大约 2.1 MIPS 和高音控制使用 1.2 MIPS。这两者是可以同时打开的。

新设定的 VS1053b 低音和高音初始化和音量的改变，会延迟到下一批发送到音频 FIFO 中的采样数据上实现。因此，和早期的 VS10XX 芯片不同，当 SCI_BASS 或 SCI_VOL 被写入的时候，音频中断不再会出现失败的情况。

注：原文中的说明误写为“频率下限”，其实这个值是用来指定使用增强算法的低频上限。

8.7.4 SCI_CLOCKF (RW)

此 SCI_CLOCKF 的操作和 VS1003、VS1033 相比，VS1053b 有一点小小的改动。乘数 1.5x 和附加的 0.5x 已经废除，它允许更高的时钟配置。

SCI_CLOCKF bits		
名称	位域	说明
SC_MULT	15:13	时钟乘数
SC_ADD	12:11	允许的附加乘数
SC_FREQ	10: 0	时钟频率

SC_MULT 激活内建的时钟乘法器。它乘上 XTALI 来创造很高的 CLKI。其数值意义如下所示：

SC_MULT	屏蔽	CLKI
0	0x0000	XTALI
1	0x2000	XTALI×2.0
2	0x4000	XTALI×2.5
3	0x6000	XTALI×3.0
4	0x8000	XTALI×3.5
5	0xa000	XTALI×4.0
6	0xc000	XTALI×4.5
7	0xe000	XTALI×5.0

SC_ADD 用来表明：在解码器固件在解码 WMA 或 AAC 时，临时需要更高时钟的时候，允许在 SC_MULT 的基础上再增加多少乘数。它的值如下：

SC_ADD	屏蔽	附加乘数
0	0x0000	不允许修改
1	0x0800	1.0×
2	0x1000	1.5×
3	0x1800	2.0×

SC_FREQ 用于描述输入的时钟晶振是运行在相对 12.288 MHz 之外的其它点上。晶振是按照 4kHz 的步长来设置的。此寄存器校正数值的计算公式是：（晶振频率 – 8000000）/ 4000（晶振频率单位是 Hz)。

注意：这个缺省值 0 是表示假设晶振频率等于 12.288 MHz。

注意：因为最大采样率是：晶振频率/256，如果晶振频率小于 12.288 MHz，不是所有的采样速率都可利用。

注意：自动修改时钟的情况只会发生在解码 WMA 和 AAC 文件的情况下。自动修改时钟每次是 0.5×。这不会导致落差达到 1.0× 时钟，让你能在整个文件中使用相同的 SCI 和 SDI 时钟。

例如：如果 SCI_CLOCKF 是 0x9BE8，即 SC_MULT = 4、SC_ADD = 3、SC_FREQ = 0x3E8 = 1000。这个晶振频率为 1000*4000+8000000 = 12 MHz（参见公式：（晶振频率 – 8000000）/ 4000 ）。时钟的乘数设定为：3.5*晶振频率 = 42 MHz，加上最大所允许的乘数，该固件将自动的选择使用：(3.5+2.0)* 晶振频率= 66 MHz 。

8.7.5 SCI_DECODE_TIME (RW)

在解码正确的数据时，当前解码所用的时间长度被以秒为单位储存在此寄存器内。

用户可以修改此寄存期的值。在这种情况下新数值应该要写两次，以保证它不会被固件覆盖掉。

一次写 SCI_DECODE_TIME 的动作会重置 byteRate 的计算。

SCI_DECODE_TIME 在每次硬件和软件复位后被重置。当一个文件解码结束时它将不再被清除，这样允许在自动循环和多个文件播放时让编码时间继续记录。

在快速播放时（参见附加的参数 playSpeed），这个编码时间也会快速计数。

一些编解码器（WMA 和 Ogg Vorbis）同样能指明播放的绝对位置，参见章节 9.11 的附加参数 positionMsec。

8.7.6 SCI_AUDATA (RW)

当解码器在解码正确的数据时，当前的采样率和通道的数量可以分别在 SCI_AUDATA 的位域 15:1 和位元 0 中找到。位域 15:1 包含了两个采样率，位元 0 如果是 0 表示单声道，而等于 1 时则是双声道。写入 SCI_AUDATA 将会直接改变采样速率。

举例：

44100 Hz 立体声数据读出为 0xAC45 (44101)。

11025 Hz 单声道数据读出为 0x2B10 (11024)。

写入 0xAC80 将设定采样率为 44160 Hz，立体声模式不会修改。译者注：这里的意思似乎应该是不会改变到立体声模式。

在休眠时，为了降低数字单元的功耗，可以写一个低采样率到这个 SCI_AUDATA。

8.7.7 SCI_WRAM (RW)

SCI_WRAM 用来上载应用程序和数据到指令和数据的 RAM 中。起始地址必须在首次读/写 SCI_WRAM 操作之前事先写入 SCI_WRAMADDR 来初始化。在一次 SCI_WRAM 的读/写操作中可以传送一个 16 位的数据和 32 位长度的指令字，而每个指令字必须用两次连续的读 / 写操作来完成。字节的顺序采用大端模式（big-endian：即高字节在前）。在每个完整的字读 / 写操作之后，内部的（地址）指针会自动增加。

8.7.8 SCI_WRAMADDR (W)

SCI_WRAMADDR 用来设置随后的 SCI_WRAM 读/写操作的程序地址。地址偏移量的 0 用于 X，0x4000 用于 Y，而 0x8000 则是指令存储器。并且外设寄存器也可以访问。

SM_WRAMADDR 起始…结束	目的地址 起始…结束	宽度	说明
0x1800. . . 0x18XX	0x1800. . . 0x18XX	16	X 数据 RAM
0x5800. . . 0x58XX	0x1800. . . 0x18XX	16	Y 数据 RAM
0x8040. . . 0x84FF	0x0040. . . 0x04FF	32	指令 RAM
0xC000. . . 0xFFFF	0xC000. . . 0xFFFF	16	I/O

唯一提供给用户使用的区域在上面列表中的 X、Y 和指令存储器。其它区域可以访问，但是不应该对它们写入，除非用其它特殊的方法。

8.7.9 SCI_HDAT0 和 SCI_HDAT1 (R)

对于 WAV 文件， SCI_HDAT1 包含的数值 0x7665（ “ve” ）。SCI_HDAT0 包含着所有支持的 RIFF WAVE 格式所测量出的字节/秒的数据速率：单声道或双声道 8 位或 16 位 PCM、单声道或双声道 IMA ADPCM。要获得此文件的比特率，用这个值乘以 8。

对于 AAC ADTS 流，SCI_HDAT1 包含的数值 0x4154（ “AT” ）。对于 AAC ADIF 文件，SCI_HDAT1 包含的数值 0x4144（ “AD” ）。对于 AAC .mp4 / .m4a 文件，SCI_HDAT1 包含的数值 0x4D34（ “M4” ）。SCI_HDAT0 包含着字节/秒的平均数据速率。要获得此文件的比特率，用这个值乘以 8。

对于 WMA 文件，SCI_HDAT1 包含的数值 0x574D（ “WM” ）和 SCI_HDAT0 包含着所测量的字节/秒的数据速率。要获得此文件的比特率，用这个值乘以 8。

对于 MIDI 文件，SCI_HDAT1 包含的数值 0x4D54（ “MT” ）和 SCI_HDAT0 包含着字节/秒的平均数据速率。要获得此文件的比特率，用这个值乘以 8。

对于 Ogg Vorbis 文件，SCI_HDAT1 包含的数值 0x4F67（ “Og” ）和 SCI_HDAT0 包含着字节/秒的平均数据速率。要获得此文件的比特率，用这个值乘以 8。

对于 MP3 文件，SCI_HDAT1 会是在 0xFFE0 和 0xFFFF 之间。SCI_HDAT1 / 0 包含了下列信息：

位域	功能	数值	描述
HDAT1[15:5]	同步字	2047	合法流
HDAT1[4:3]	ID	3 2 1 0	ISO 11172-3 MPG 1.0 ISO 13818-3 MPG 2.0 (1/2速率) MPG 2.5 (1/4速率) MPG 2.5 (1/4速率)
HDAT1[2:1]	阶层	3 2 1 0	I II III 保留
HDAT1[0]	保护位	1 0	没有 CRC CRC保护
HDAT0[15:12]	比特率		参阅比特率表
HDAT0[11:10]	采样率	3 2 1 0	保留 32/16/ 8 kHz 48/24/12 kHz 44/22/11 kHz
HDAT0[9]	填充位	1 0	附加位置 标准框架
HDAT0[8]	私有位		无定义
HDAT0[7:6]	模式	3 2 1 0	单通道 双通道 联合立体声 立体声
HDAT0[5:4]	扩展		参阅 ISO 11172-3
HDAT0[3]	版权	1 0	有版权 自由的
HDAT0[2]	原件	1 0	原件 拷贝
HDAT0[1:0]	重点	3 2 1 0	CCITT J.17 保留 50/15 微秒 空

在读取时，SCI_HDAT0 和 SCI_HDAT1 包含的当前正在解码 MP3 流的标头信息。在复位后两个寄存器被清除则表示没有找到数据。

“采样率”区域在 SCI_HDAT0 里的解释如下表所示：

采样率	ID=3	ID=2	ID=0,1
3	-	-	-
2	32000	16000	8000
1	48000	24000	12000
0	44100	22050	11025

“比特率”区域在 SCI_HDAT0 里面的解释如下表所示。注意：对于可变比特率流，值是不断变化的。

比特率	阶层 I ID=3 | ID=0,1,2 kbit/s		阶层 II ID=3 | ID=0,1,2 kbit/s		阶层 III ID=3 | ID=0,1,2 kbit/s	
15	禁止	禁止	禁止	禁止	禁止	禁止
14	448	256	384	160	320	160
13	416	224	320	144	256	144
12	384	192	256	128	224	128
11	352	176	224	112	192	112
10	320	160	192	96	160	96
9	288	144	160	80	128	80
8	256	128	128	64	112	64
7	224	112	112	56	96	56
6	192	96	96	48	80	48
5	160	80	80	40	64	40
4	128	64	64	32	56	32
3	96	56	56	24	48	24
2	64	48	48	16	40	16
1	32	32	32	8	32	8
0	-	-	-	-	-	-

字节/秒的平均数据速率可以在存储器中读取，请参阅附加参数 byteRate。此变量包含了所有编解码器的字节速率。要获得此文件的比特率，用此值乘以 8 即可。

比特率的计算在歌曲之间是不会自动复位的，但它除了软件和硬件复位之外，还可以通过对 SCI_DECODE_TIME 的写操作来复位。

8.7.10 SCI_AIADDR (RW)

SCI_AIADDR 表示应用程序代码的起始地址，它应该和早先的 SCI_WRAMADDR 和 SCI_WRAM 寄存器一起写入。如果不使用应用程序代码，则此寄存器不应该初始化，或者应该初始化为 0。若需要更加详细的描述，请参考“VS10XX 应用注意事项”（Application Notes for VS10XX）。

8.7.11 SCI_VOL (RW)

SCI_VOL 用于播放器的硬件音量控制。这个音量寄存器的高字节是控制左通道音量的，低字节是控制右通道音量的。对每个通道，数值在 0..254 的范围内设置可以实现在最大音量级别内的微调（步长 0.5 dB）。左通道数值是通过乘上 256 后再加到这个数值上的。所以，最大的音量是 0、无声是 0xFEFE。

注意：在硬件复位之后，音量会被设置为最大音量。软件复位则不会重置先前的设定。

设置 SCI_VOL 为 0xFFFF 将会使模拟单元进入掉电模式。

例如：左通道是-2.0 dB 和右通道是-3.5 dB：(2.0/0.5) = 4，3.5/0.5 = 7→SCI_VOL = 0x0407。

例如：SCI_VOL = 0x2424 表示左右通道的音量是 0x24 * -0.5 = -18.0 dB。

新设定的 VS1053b 低音和高音初始化和音量的改变，会延迟到下一批发送到音频 FIFO 中的采样数据上执行。因此，SCI_BASS 或 SCI_VOL 被写入的时候，音频中断不再会出现失败的情况。

这个延时会使音量设定稍微延迟，但是因为音量控制是由 DAC 硬件立即完成的，而不是执行已经进入音频 FIFO 的采样数据，在整体上音量控制的响应要比之前的更好。而且，音量控制有侦测“过零交叉”的功能，几乎消除了音量在突然改变的时候产生的可听见的所有杂音。

8.7.12 SCI_AICTRL[x] (RW)

SCI_AICTRL[x] 寄存器组（x=[0 .. 3]）可以用来访问用户的应用程序。

它们也被 IMA ADPCM 录音模式所使用。

9 操作

9.1 时钟

VS1053b 通常运作在一个频率为 12.288 MHz 基频的主时钟上。时钟可以通过外部的电路（连接到引脚 XTALI）或通过内部的时钟晶振接口（连接到 XTALI 和 XTALO 引脚）来产生的。这个时钟用来给模拟部件确定最高的可用采样率。在 12.288MHz 下，最高到 48000Hz 的所有采样率都可用。

在 SCI_MODE 寄存器的 SM_CLK_RANGE 设置为‘1’时，VS1053b 可以使用 24…26MHz 的时钟。这个系统时钟是通过将时钟输入除上 2 来获得 12…13MHz 的输入时钟。

9.2 硬件复位

当 XRESET 信号被驱动到低电平，VS1053b 将重置所有的控制寄存器和内部状态到它们的初始值。XRESET 信号是和任何的外部时钟异步运作的。该复位模式节省的功耗双倍于全掉电模式，在 VS1003 的数字和模拟部件处于最小功耗状态期间，时钟信号也是停止的。并且 XTALO 被连接到地线上。

在 XRESET 有效时，所有的输出引脚会转变到它们的缺省状态。所有的输入引脚转变到高阻状态（到输入状态）。除了 SO 外，其它的仍然由 XCS 来控制。

在硬件复位（或上电）之后，DREQ 将在低电平上停留最少 22000 个时钟周期，对于运行在 12.288 MHz 的 VS1053b 上来说，大约是 1.8 毫秒。随后用户应该在开始解码之前事先设置 SCI_MODE、SCI_BASS、SCI_CLOCKF 和 SCI_VOL 这几个基本软件寄存器。请参阅章节 8.7 中的说明。

如果输入时钟是 24…26MHz，应该在芯片复位之后尽快设置 SM_CLK_RANGE，这个操作并不需要等待 DREQ 信号。

内部频率可以通过 PLL 来倍增。所支持的乘法器通过 SCI_CLOCKF 寄存器可以将输入时钟增加 1.0 × . . . 5.0×。内部时钟乘法器的复位值是 1.0×。如果需要典型值，则内部乘法器值在复位之后必须设置为 3.5x。等待 DREQ 升高之后，将数值 0x9800 写入 CI_CLOCKF（寄存器 3）。请参阅章节 8.7.4 中的说明。

9.3 软件复位

有些情况下，解码器需要用软件来复位。这是通过激活 SCI_MODE 寄存器中的 SM_RESET 位来实现的（参见章节 8.7.1），然后最少需要等待 2 微秒后再去查询 DREQ 状态。DREQ 将在低电平上停留最少 22000 个时钟周期，对于运行在 12.288 MHz 的 VS1053b 上来说，大约是 1.8 毫秒。然后等 DREQ 上升变为高，就可以像平常那样继续播放了。

和先前所有的 VS10XX 芯片相比，它不推荐在在那些歌曲交替播放的期间用软件复位。这是因为用户可能需要一些低采样率和比特率的文件能平坦的播放到它们的末端。

9.4 低功耗模式

如果你需要保持系统在运行又不需要解码，但希望降低功耗，你可以用这些方法来实现：

- 通过将 0x0000 写入 SCI_CLOCKF 来选择 1.0x 的时钟，这将停掉 PLL 来节省电力。
- 写一个小的、非 0 的值，例如 0x0010 到 SCI_AUDATA，这将减少采样率和减少音频中断需求的数量。在音频中断期间的 VSDSP 内核将等待一个中断，这可以节省电力。
- 将音频后加工处理全部关掉（音调控制和 EarSpeaker 效果）。
- 如果对于此产品有可能的话，写 0xffff 到 SCI_VOL 来关掉模拟驱动单元。

要从低功耗模式返回，要按照相反的顺序来恢复那些值。

注意：这种低功耗模式明显比硬件复位要消耗更多的电力。

9.5 播放和解码

在 VS1053b 正常的操作模式下，SDI 数据被解码。解码的采样数据由内置的 DAC 转换成模拟量。如果没有解码数据被找到，SCI_HDAT0 和 SCI_HDAT1 会被设置为 0。

在没有解码数据输入的时候，VS1053b 会进入闲置模式（与解码期间相比功耗要低）并积极监视串行数据输入总线上的合法数据。

9.5.1 播放一个完整文件

这是缺省播放模式，它用下列步骤来实现：

1、 发送音频文件到 VS1053b；

2、 读取附加参数 endFillByte 的值（参见章节 9.11）。

3、 发送 endFillByte[7:0] 中的值最少 2052 个字节。

4、 设置 SCI_MODE 的 SM_CANCEL 位。

5、 发送 endFillByte[7:0] 中的值最少 32 个字节。

6、 读取 SCI_MODE。如果 SM_CANCEL 位依然是设置的，则跳到步骤 5；如果 SM_CANCEL 始终没有被清除，在发送 2048 个字节之后，执行软件复位（这可能是极端的情况）。

7、 这首歌曲现在已经成功被发送了。HDAT0 和 HDAT1 都应该包含 0 来表明现在已经没有正在解码的格式了。现在返回步骤 1。

9.5.2 取消播放

在放音时，当用户想跳到下一首，取消当前歌曲的播放是正常现象。它用下列步骤来实现：

1、 发送一个音频文件的部分到 VS1053b。

2、 设置 SCI_MODE 的 SM_CANCEL 位。

3、 继续发送音频文件，但要在发送 32 字节的数据之后检查 SM_CANCEL。如果它还是被设置，则重新进行步骤 3。如果 SM_CANCEL 始终没有被清除，在发送 2048 个字节之后，执行软件复位（这可能是极端的情况）。

4、 当 SM_CANCEL 被清除，读取附加的参数值 endFillByte（参见章节 9.11）。

5、 发送 endFillByte[7:0] 中的值，共计 2052 个字节。

6、 HDAT0 和 HDAT1 都应该包含 0 来表明现在已经没有正在解码的格式了。你现在可以开始发送下一个音频文件了。

9.5.3 快速播放

VS1053b 允许快速播放。如果你的微控制器可以足够快地为 VS1053b 提供数据，就可以执行快进音频。这些步骤如下：

1、 开始发送音频文件到 VS1053b。

2、 设定快速播放，设置附加参数 playSpeed 的值（参见章节 9.11）。

3、 继续发送音频数据。

4、 要退出快速播放模式，写 1 到 playSpeed。

要估计你的微控制器是否可以为 VS1053b 在快速播放模式下提供足够的数据，请参阅附加参数 byteRate 值的内容（章节 9.11）。注意 byteRate 包含的是这个文件播放数据速度的标称速度，即使在快速播放被激活的状态下也是这样。

注意：播放速度在歌曲改变的时候也不会被重置。

9.5.4 无声的快进和快倒

要做到快进和快退，你必须有能力随机访问音频文件。遗憾的是，快进和快退并非在任何时间都可使用。比如在文件的标头正在读取的时候。这些步骤如下：

1、 发送一个音频文件的部分到 VS1053b。

2、 在需要随机访问的时候，查询 SCI_STATUS 的 SS_DO_NOT_JUMP 位，如果这个位被设置，表示现在不可进行随机访问，返回步骤 1。

3、 读取附加的参数 endFillByte 的值（章节 9.11）。

4、 发送 endFillByte[7:0]中的值最少 2048 个字节。

5、 在这个文件中前进或后退。

6、 继续发送这个文件。

注意：建议在它快进和快退时将音量降低 10dB。

注意：DECODE_TIME 寄存器不会考虑在跳跃时继续记录。

注意：Midi 不合适做随即访问，你可以用附加的 playSpeed 参数来实现可选的 1x 到 128x 的播放速度来快进。SCI_DECODE_TIME 也会加速。如果有必要，快倒可以用重新启动一个 MIDI 文件的解码和快进到适当位置来实现。

SCI_DECODE_TIME 可用来确定是否到达正确的位置。

9.5.5 保持正常的解码时间

当快进和快倒在执行的时候，多数文件无法维持正常的解码时间。然而，在 WMA 和 Ogg Vorbis 文件中可提供准确的时间信息。想使用准确的时间信息，只要有可能，请用下列步骤：

1、 开始发送一个音频文件到 VS1053b。
2、 读取附加参数 positionMsec 的值（章节 9.11）。
3、 如果 positionMsec 是-1，则使用 DECODE_TIME 显示你的解码时间（和如果你执行了快进和快倒操作后，你的文件位置）。
4、 如果 positionMsec 不是-1，则使用这个时间来显示此文件中的明确位置。

9.6 供应 PCM 数据

VS1053b 可通过发送一个 WAV 文件的标头的办法来用作 PCM 解码。如果在 WAV 标头内的长度是 0xFFFFFFFF，VS1053b 将无限期地留在 PCM 模式（或直到 SM_CANCEL 被设置为止）。8 位线性和 16 位线性音频在单声道或立体声模式下均可获得支持。一个 WAV 标头看起来像这样：

文件偏移量	字段名	大小	字节	说明
0	ChunkID	4	"RIFF"	集合的 ID = RIFF
4	ChunkSize	4	0xff 0xff 0xff 0xff	集合的大小
8	Format	4	"WAVE"	格式 = WAVE
12	SubChunk1ID	4	"fmt "	子集合1的 ID = fmt （注意后面有个空格）
16	SubChunk1Size	4	0x10 0x0 0x0 0x0	子集合的的大小 = 16
20	AudioFormat	2	0x1 0x0	音频格式 = 线形 PCM
22	NumOfChannels	2	C0 C1	通道数量，1 是单声道，2 是双声道
24	SampleRate	4	S0 S1 S2 S3	采样率，0x1f40 是 8 kHz
28	ByteRate	4	R0 R1 R2 R3	字节速率，0x3e80 是 8 kHz 16位单声道
32	BlockAlign	2	A0 A1	块对齐，2是单声道，4是立体声（16位）
34	BitsPerSample	2	B0 0xB1	单个采样数据的大小，16 是 16 位数据
52	SubChunk2ID	4	"data"	子集合2的 ID = data
56	SubChunk2Size	4	0xff 0xff 0xff 0xff	子集合2的大小 = 数据大小

四个变量的计算规则如下所示：

- *S* = 采样率单位是 Hz，即 44100 是 44.1KHz。
- 对于 8 位数据 *B* = 8，而 16 位数据是 *B* = 16。
- 对于单声道数据 *C* = 1，对于双声道 *C* = 2。
- *A* = (*C***B*) / 8。
- *R* = *S* × *A*。

例子：一个 44100Hz、16 位立体声 PCM 标头将读取为下面的样子：

```
0000  52 49 46 46 ff ff ff ff  57 41 56 45 66 6d 74 20 |RIFF....WAVEfmt |
0010  10 00 00 00 01 00 02 00  44 ac 00 00 10 b1 02 00 |........D.......|
0020  04 00 10 00 64 61 74 61  ff ff ff ff             |....data....|
```

注：原文中例子的地址数据有错误，这里已经修改为正确的了。

9.7 Ogg Vorbis 录音

Ogg Vorbis 是一个开放的档案格式，允许用低至中等的比特率达到非常高的音质。Ogg Vorbis 的录音是通过加载 Ogg Vorbis 编码器应用程序到 VS1053b 的 16 KiB 程序 RAM 存储器中来使能的。在使能后，编码的结果可以从寄存器 SCI_HDAT0 和 SCI_HDAT1 中读取，和 ADPCM 录音非常相似（章节 9.8）。

有三种特性可提供：一、大约在 140 kbit/s 比特率左右的高品质立体声录音；二、用于语音品质的在 15 和 30 kbit/s 比特率之间的单声道录音。

如果需要 Ogg Vorbis 编码程序，请到 VLSI Solution 的主页上去获取这个应用程序：*http://www.vlsi.fi/software/plugins/plugins.shtml and read the accompanying documentation.*

9.8 ADPCM 录音

这部分章节讲述如何创建 RIFF / WAV 文件和 IMA ADPCM 格式。这是一个可广泛支持的 ADPCM 格式，在 PC 上的大多数音频播放软件可以播放它。IMA ADPCM 录音提供了一个经粗略压缩（比例大约为 4:1）的线性 16 位音频。在 32.44 kbit / s 的基础上能录制频率为 8 kHz 的音频。

VS1053b 有立体声 ADC，因此双声道（如果 AGC 使能，可以使用各自的 AGC）和立体声（如果 AGC 是激活的，将使用共同设置的 AGC）模式是可用的。单声道录音模式选择不是左通道就是右通道。左通道不是“咪”就是“线路 1”，这由 SCI_MODE 寄存器来决定。

9.8.1 激活 ADPCM 模式

寄存器	位域	说明
SCI_MODE	2, 12, 14	开始 ADPCM 模式，选择：咪/线路1
SCI_AICTRL0	15..0	采样率 8000..48000 Hz (在录音启动时读取的)
SCI_AICTRL1	15..0	录音增益 (1024 = 1×) 或 0 是自动增益控制 (AGC)
SCI_AICTRL2	15..0	自动增益放大器的最大值 (1024 = 1×, 65535 = 64×)
SCI_AICTRL3	1..0 2 15..3	0=联合立体声(共用 AGC)，1=双声道(各自的 AGC)，2=左通道，3=右通道 0=IMA ADPCM 模式，1=线性 PCM 模式 保留，设置为 0

IMA ADPCM 录音模式是通过设置 SCI_MODE 中的 SM_RESET 和 SM_ADPCM 位来激活的。如果 SM_LINE1 被设置，则“线路输入 1”会替代“咪”的差分输入。在激活 ADPCM 录音之前，用户必须要将正确的值写入 SCI_AICTRL0 和 SCI_AICTRL3 中。这些值仅在录音被启动时才读取。SCI_AICTRL1 和 SCI_AICTRL2 可以随时更改，但在激活之前最好写入好的初始值。

注：原文将此节漏掉了，这里补充的是 VS1003b 的内容。在后面 9.8.3 的章节会需要这段内容 。

在激活 ADPCM 录音之前，用户应该写个 SCI_AICTRL0 的时钟分频值给 SCI_AICTRL0。采样率是用后面这个公式计算的：$f_s = F_c / (256 \times d)$。$F_c$ 是内部的时钟（CLKI），d 是 SCI_AICTRL0 里的分频值。d 的最小合法值是 4。如果 SCI_AICTRL0 的值是 0，则代表默认值 12。下面是计算示例：

Examples 例子:

F_c =2.0 × 12.288 MHz, d = 12. 则 f_s = (2.0×12288000) / (256×12) = 8000 Hz.

F_c =2.5 × 14.745 MHz, d = 18. 则 f_s = (2.5×14745000) / (256×18) = 8000 Hz.

F_c =2.5 × 13 MHz, d = 16. 则 f_s = (2.5×13000000) / (256×16) = 7935 Hz.

SCI_AICTRL1 控制线形录音的增益。1024 相当于数字增加 1，而 512 相当于数字增加 0.5 并以此类推。如果用户要使用自动增益控制（AGC），SCI_AICTRL1 应设置为 0。录音时，典型的语音应用状况下使用 AGC 通常是较佳的，因为这需要将平静的和喧闹的声音处理成相对统一的响度。

SCI_AICTRL2 控制 AGC 增益的最大值。这可以用来控制在没有信号时，限制噪音被放大。如果 SCI_AICTRL2 是 0，则这个最大增益将被初始化成 65536（64x），即使用全范围。

下面的例子是选择了 16KHz，立体声模式，使用自动增益控制和最大是 4x 的放大倍数：

```
WriteVS10xxRegister( SCI_AICTRL0, 16000U );
WriteVS10xxRegister( SCI_AICTRL1, 0 );
WriteVS10xxRegister( SCI_AICTRL2, 4096U );
WriteVS10xxRegister( SCI_AICTRL3, 0 );
WriteVS10xxRegister( SCI_MODE, ReadVS10xxRegister(SCI_MODE) |
                     SM_RESET | SM_ADPCM | SM_LINE1 );
WriteVS10xxPatch(); /* 仅仅是 VS1053b */
```

WriteVS10xxPatch() 应该执行下列 SCI 写操作（仅仅是 VS1053b）：

寄存器	寄存器编号	数值
SCI_WRAMADDR	0x7	0x8010
SCI_WRAM	0x6	0x3e12
SCI_WRAM	0x6	0xb817
SCI_WRAM	0x6	0x3e14
SCI_WRAM	0x6	0xf812
SCI_WRAM	0x6	0x3e01
SCI_WRAM	0x6	0xb811
SCI_WRAM	0x6	0x0007
SCI_WRAM	0x6	0x9717
SCI_WRAM	0x6	0x0020
SCI_WRAM	0x6	0xffd2
SCI_WRAM	0x6	0x0030
SCI_WRAM	0x6	0x11d1
SCI_WRAM	0x6	0x3111
SCI_WRAM	0x6	0x8024
SCI_WRAM	0x6	0x3704
SCI_WRAM	0x6	0xc024
SCI_WRAM	0x6	0x3b81
SCI_WRAM	0x6	0x8024
SCI_WRAM	0x6	0x3101
SCI_WRAM	0x6	0x8024
SCI_WRAM	0x6	0x3b81
SCI_WRAM	0x6	0x8024
SCI_WRAM	0x6	0x3f04
SCI_WRAM	0x6	0xc024
SCI_WRAM	0x6	0x2808
SCI_WRAM	0x6	0x4800
SCI_WRAM	0x6	0x36f1
SCI_WRAM	0x6	0x9811
SCI_WRAMADDR	0x7	0x8028
SCI_WRAM	0x6	0x2a00
SCI_WRAM	0x6	0x040e

这个补丁也可以利用机器在 VLSI Solution 的页面：

http://www.vlsi.fi/en/support/software/vs10xxpatches.html

通过 *VS1053b IMA ADPCM Encoder Fix* 这个名称来获得。

9.8.2 读取 IMA ADPCM 数据

在激活了 IMA ADPCM 录音之后，寄存器 SCI_HDAT0 和 SCI_HDAT1 有了新的功能。

IMA ADPCM 的采样缓冲区是 1024 个 16 位字。缓冲区的填充状态可以通过 SCI_HDAT1 来读取。如果 SCI_HDAT1 大于 0，则你可以从 SCI_HDAT0 读取许多个 16 位的字。如果数据没能足够快的读取，则缓冲区会溢出并返回空的状态。

注意：如果 SCI_HDAT1≥896，它最好是等待缓冲区溢出和在读取采样数据之前清除。这样是为了让你避开缓冲区混叠。

每个 IMA ADPCM 块是 128 字组，即 256 字节。如果你想暂停一下读取数据并在稍后恢复，请按照 128 字的界限来中止。这样可以跳过完整的块并让编码流保持正确。

9.8.3 加上一个 RIFF 标头

要制作你的 IMA ADPCM 格式的 RIFF / WAV 文件，你必须在实际数据之前加上一个标头。下面显示了一个单声道文件的标头。注意：这些格式中的 2 和 4 字节数值是小端模式的（即低字节在前）。

文件偏移量	字段名	大小	字节	说明
0	ChunkID	4	"RIFF"	集合的 ID = RIFF
4	ChunkSize	4	F0 F1 F2 F3	文件大小减 8
8	Format	4	"WAVE"	格式 = WAVE
12	SubChunk1ID	4	"fmt "	子集合1的 ID = fmt （注意后面有个空格）
16	SubChunk1Size	4	0x14 0x0 0x0 0x0	子集合的的大小 = 20
20	AudioFormat	2	0x11 0x0	音频格式，0x11 = IMA ADPCM
22	NumOfChannels	2	C0 C1	通道数量，1 是单声道 , 2 是双声道
24	SampleRate	4	R0 R1 R2 R3	采样率，0x1f40 是 8 kHz
28	ByteRate	4	B0 B1 B2 B3	字节速率，0xfd7 是 8 kHz 单声道
32	BlockAlign	2	0x0 0x1	块对齐（字节）= 0x100
34	BitsPerSample	2	0x4 0x0	单个采样数据的大小，4 位 ADPCM
36	ByteExtraData	2	0x2 0x0	附加的数据字节 2
38	ExtraData	2	0xf9 0x1	附加的数据，单个采样数据块大小 (505)
40	SubChunk2ID	4	"fact"	子集合2的 ID = fact
44	SubChunk2Size	4	0x4 0x0 0x0 0x0	子集合2的大小 = 4
48	NumOfSamples	4	S0 S1 S2 S3	采样的数量
52	SubChunk3ID	4	"data"	子集合3的 ID = data
56	SubChunk3Size	4	D0 D1 D2 D3	子集合3的数据大小 (文件大小减60)
60	Block1	256		第一个 ADPCM 数据块
316	. . .			更多的 ADPCM 数据块

如果我们有 n 个音频块，则这个表中的值如下：

F = n × C × 256 + 52

R = Fs (请参阅章节 9.8.1 来看如何计算 *Fs*)

B = (Fs×C×256)/505

S = n × 505. D = n × C × 256

如果你事先知道要记录多少，那你能在若干实际数据之前就填充完整的标头了。但是，如果你不知道需要记录多少，那只能在完成记录之后才能去填充标题中的数据 *F*、*S* 和 *D* 了。

一个 ADPCM 块的 128 个字（256 字节）读自 SCI_HDAT0 并写入到随后的文件中。而 SCI_HDAT0 的高 8 位应该作为第一个字节写到文件中，然后才是低 8 位。注意：这和多数微控制器的缺省运作是相反的，并且需要你额外的关注去做这种修正。

有个办法是：如果你写的文件做了修正动作，则可以检查每个 256 字节块的字节 2 和 3（第一个字节的位置是 0，所以字节 2 和字节 3 是最后两个字节）。字节 3 应该总是 0。

下面是一个例子，一个 44.1kHz、立体声的 IMA ADPCM 文件的有效标头最终长度是 10038844（0x992e3c）字节：

```
0000 52 49 46 46 34 2e 99 00 57 41 56 45 66 6d 74 20 |RIFF4...WAVEfmt |
0010 14 00 00 00 11 00 02 00 44 ac 00 00 a7 ae 00 00 |........D.......|
0020 00 02 04 00 02 00 f9 01 66 61 63 74 04 00 00 00 |........fact....|
0030 14 15 97 00 64 61 74 61 00 2e 99 00             |....data....|
```

9.8.4 播放 ADPCM 数据

为了播放您的 IMA ADPCM 录音，你必须有一个和章节 9.8.3 中所述的那样标头的文件。在这种情况下，您需要做的所有事情和你处理任何音频文件一样，就是通过 SDI 提供 ADPCM 文件。

9.8.5 采样率的考虑

VS10xx 芯片支持 IMA ADPCM 播放，能够播放任何采样率的 ADPCM 文件。然而，另一些程序可能会期望 IMA ADPCM 文件有一些精确的采样率，例如 8000 或 11025Hz 。此外，一些程序或系统不支持低于 8000Hz 的采样率。

如果您想要更好的质量就要增加数据传输速率的消耗，您可以使用更高的采样率，例如 16kHz。

9.9 SPI 引导

如果在启动时用上拉电阻将 GPIO0 设置为高，VS1053b 会设法通过外部的 SPI 存储器来引导。SPI 引导将重新定义下列的引脚：

正常模式	SPI 引导模式
GPIO0	xCS
GPIO1	CLK
DREQ	MOSI
GPIO2	MISO

存储器是一个 SPI 总线串行 EEPROM 和 16 位或 24 位地址。VS1053b 在正常 12.288 MHz 时钟下串行速度是 245 kHz。存储器的头三个字节的内容必须是 0x50，0x26，0x48。

9.10 实时 MIDI

如果在启动时 GPIO0 是低和 GPIO1 是高，则实时 MIDI 模式被激活。在此模式下 PLL 是设定为 4.0×，UART 被配置为 MIDI 的数据传输率 31250bps，并且实时 MIDI 数据是从 UART 和 SDI 读出的。这两个输入不应该同时使用。如果您使用的 SDI，首先要发送 0xff 然后发送 MIDI 数据字节。

EarSpeaker 的设置可以用 GPIO2 和 GPIO3 来配置。但 GPIO2 和 GPIO3 的状态只是在启动时读取。

实时 MIDI 可以使用 SCI 加载一个小的补丁代码来开始。

注意：VS1053b 内的实时 MIDI 解析器不知道如何跳过 SysEx 信息。若有需要的话，可以将一个改进的版本加载到 IRAM。

9.11 附加的参数

下面的结构是在 X 存储器的地址 0x1e00 处（注意：VS1033 在不同的位置 ），可以用来修改一些附加的参数或得到有用的信息。该芯片 ID 也能容易获得。

```
#define PARAMETRIC_VERSION 0x0003
struct parametric {
    /* configs are not cleared between files */
    u_int32 chipID;         /*1e00/01 Initialized at reset for your convenience */
    u_int16 version;        /*1e02 - structure version */
    u_int16 config1;        /*1e03 ---- ---- ppss RRRR PS mode, SBR mode, Reverb */
    u_int16 playSpeed;      /*1e04 0,1 = normal speed, 2 = twice, 3 = three times etc. */
    u_int16 byteRate;       /*1e05 average byterate */
    u_int16 endFillByte;    /*1e06 byte value to send after file sent */
    u_int16 reserved[16];   /*1e07..15 file byte offsets */
    u_int32 jumpPoints[8];  /*1e16..25 file byte offsets */
    u_int16 latestJump;     /*1e26 index to lastly updated jumpPoint */
    u_int32 positionMsec    /*1e27-28 play position, if known (WMA, Ogg Vorbis) */
    s_int16 resync;         /*1e29 > 0 for automatic m4a, ADIF, WMA resyncs */
    union {
        struct {
            u_int32 curPacketSize;
            u_int32 packetSize;
        } wma;
        struct {
            u_int16 sceFoundMask;   /*1e2a SCE's found since last clear */
            u_int16 cpeFoundMask;   /*1e2b CPE's found since last clear */
            u_int16 lfeFoundMask;   /*1e2c LFE's found since last clear */
            u_int16 playSelect;     /*1e2d 0 = first any, initialized at aac init */
            s_int16 dynCompress;    /*1e2e -8192=1.0, initialized at aac init */
            s_int16 dynBoost;       /*1e2f 8192=1.0, initialized at aac init */
            u_int16 sbrAndPsStatus; /*0x1e30 1=SBR, 2=upsample, 4=PS, 8=PS active */
        } aac;
        struct {
            u_int32 bytesLeft;
        } midi;
        struct {
            s_int16 gain; /* 0x1e2a proposed gain offset in 0.5dB steps, default = -12 */
        } vorbis;
    } i;
};
```

注意：因为 SCI_WRAMADDR 和 SCI_WRAM 接口在读取两个字的变量时没有任何方式的保护。变量可能在读取高，低部分之间的时候被更新。出现这个问题是由于低和高的部分被改变了。如果要判断数值是否正确的话，你应该读取这个数值两次并将结果进行比较。

下面的示例显示了更新发生在读取低和高部分内容之间或读取高的那部分之后，*bytesLeft* 从 0x10000 减少到 0xFFFF 所发生的状况。

无效的读取		正确的读取		无更新	
地址	数值	地址	数值	地址	数值
0x1e2a	0x0000 这之后被修改了	0x1e2a	0x0000	0x1e2a	0x0000
0x1e2b	0x0000	0x1e2b	0x0001 这之后被修改了	0x1e2b	0x0001
0x1e2a	0xffff	0x1e2a	0xffff	0x1e2a	0x0000
0x1e2b	0x0000	0x1e2b	0x0000	0x1e2b	0x0001

你可以看到：在无效的读取中，低的部分从 0x0000 回绕（指出现溢出，无论是进位还是借位）变为 0xffff，高的部分一直没有变化。在这种情况下，第二种读取提供了正确的结果，否则总是使用首次读取的数值。当低的部分可能发生回绕，第二次读取是很必要的，高的部分会出现改变，即低部分是小的部分。*bytesLeft* 只在一个时间减少（即低部分出现借位时，高低部分会同时改变），因此，仅在低的部分为 0 时，才需要重新读取。

9.11.1 通用参数

这些参数是所有编解码器共用的，而其它区域的只在相对应的编解码器是活跃时才有效。当前激活的编解码器可用 SCI_HDAT1 来确定。

参数	地址	用途
chipID	0x1e00-01	用熔丝编程的独一 ID 号 (熔丝的状态拷贝)
version	0x1e02	结构版本 – 0x0003
config1	0x1e03	各种混在一起的不同配置
playSpeed	0x1e04	播放速度：0,1 = 正常速度，2 = 倍速， 3 = 三倍速等等
byteRate	0x1e05	平均字节速率
endFillByte	0x1e06	在文件后发送的填充字节
jumpPoints[8]	0x1e16-25	WMA 和 AAC 的包偏移量
latestJump	0x1e26	最近跳跃点的索引
positionMsec	0x1e27-28	如果可用，表明文件毫秒级的位置
resync	0x1e29	自动重新同步选择

此熔丝编程 ID 是在启动时读取和复制到 chipID 领域的。如果不可用，此值将全部为零。

版本区域可以用来确定结构的其余布局。当结构改变时，版本号也会改变。对于 VS1053b 的结构版本是 3 。

config1 控制 MIDI 的混响和 AAC 的 SBR、PS 设置。

playSpeed 使我们能够快进歌曲。比特流解码的执行只是按照 playSpeed 来逐帧播放。例如：如果你能以足够快的速度填充数据，将 4 写入 playSpeed 后，播放歌曲的速度将是正常速度的四倍。写入 0 或 1 即可恢复正常的速度。SCI_DECODE_TIME 同样也会将计数速度加快。目前所有的编解码器都可支持 playSpeed 的配置。

byteRate 包含了每种代码每秒字节数的平均比特率。该值每秒更新一次，它可以用来估算余下的播放时间。除 MP3，MP2，MP1 之外的所有编解码器均可在 SCI_HDAT0 中获得这个值。

endFillByte 表明在发送文件后和设置 SM_CANCEL 前应该发送什么字节值。

jumpPoints 包含 32 位的文件偏移量。每个有效的（非零）入口表示一个 WMA 包的开始，或 AAC（ADIF，.mp4 / .m4a）的原始数据块的开始。latestJump 包含了最后更新的入口索引。如果你只是用 latestJump 来读取入口点，你不需要两次读取这个入口来确保有效性。WMA 和 AAC（ADIF，.mp4 / .m4a）的跳转点信息可用来实现完美的快进和快退。

positionMsec 是一个域，不论是倒带和快进的操作，它提供了一个在当前文件中的毫秒级位置。该值只适用于编解码器判断流本身的播放位置。目前 WMA 和 Ogg Vorbis 提供了这方面的资料。如果位置是未知的，这个域将包含-1。

resync 域常用来在首个错误出现时强迫 WMA 和 AAC（ADIF，.mp4 / .m4a）的流重新同步，而不是结束解码。这个域常用于实现 WMA 和 AAC（ADIF，.mp4 / .m4a）几乎完美的快进和快退。如果要寻找的数据不在它们的包内或数据块的边界上，用户在执行之前应该设置这个域。这个域的值表示在放弃之前允许进行多少次尝试。数值 32767 表示执行无限次数的尝试

该 resync 域在复位后被设置为 32767，来获得这个重新同步的缺省功能，但可以在复位之后清除它来恢复之前的行为。当 resync 被设置的时候，每种文件的解码结束方式都应该象章节 9.5.1 中说明的那样来终止。

Seek 域不再存在。当需要重新同步时，WMA 和 AAC 的编解码器立即进入广播/流模式而不管文件的大小信息。此外，当 resync 是非 0 时，WAV 文件的大小和采样大小的信息也会被忽略。用户必须使用 SM_CANCEL 或软件复位来结束解码。

注意：WAV、WMA、ADIF 和“.mp4 / .m4a”文件用一个元数据或标头部分作为开始的，在执行任何快进和快退操作之前必须要进行完整的处理。当标头信息处理完和允许跳跃时，SCI_STATUS 寄存器中的 SS_DO_NOT_JUMP 会被清除。

9.11.2 WMA

参数	地址	用途
curPacketSize	0x1e2a/2b	正在处理中的包的大小
packetSize	0x1e2c/2d	ASF 标头中的包的大小

该 ASF 标头包的大小可在 packetSize 中获取。你可以用这个信息和 jumpPoints 中的包起始偏移量来解析这个包的标头和跳过 ASF 文件中的那些包。

当 WMA 解码器侦测到一个文件用当前的时钟不能进行正确解码时，可能会自动增加内部时钟频率。这个最大所允许的时钟是用 SCI_CLOCKF 寄存器来配置的。

9.11.3 AAC

参数	地址	用途
config1	0x1e03(7:4)	SBR 和 PS 选择
sceFoundMask	0x1e2a	单通道元素找到
cpeFoundMask	0x1e2b	通道对（双通道）元素找到
lfeFoundMask	0x1e2c	低频元素找到
playSelect	0x1e2d	播放元素选择
dynCompress	0x1e2e	DRC 的压缩系数， -8192 = 1.0
dynBoost	0x1e2f	DRC 的提升系数， 8192=1.0
sbrAndPsStatus	0x1e30	SBR 和 PS 可用标记

如果流中有多种元素，playSelect 用来指明哪个元素可解码。每次 AAC 解码开始，这个值会设置为 0，这将导致流中首个出现的元素被选择为解码。而其它的值是：0x01 – 选择第一个单通道元素（SCE）；0x02 – 选择第一个通道对元素（CPE）；0x03 – 选择第一个低频元素（LFE），$S*16+5$ – 选择 SCE 编号 S；$P*16+6$ – 选择 CPE 编号 P；$L*16+7$ – 选择 LFE 编号 L。当自动选择被执行的时候，playSelect 将反映出选中的元素。

由于变量最后会被清除，sceFoundMask、cpeFoundMask 和 lfeFoundMask 指出 AAC 流中有哪些元素曾经被找到过。这个值能为那些唯一可用的元素提供一份元素选择菜单。

dynCompress 和 dynBoost 改变动态范围控制（DRC）在目前某些 AAC 流中的表现。这些是在 AAC 解码开始时初始化的。

sbrAndPsStatus 指出频带复制（SBR）和参量立体声（PS）的状态。

位域	用途
0	SBR 存在
1	增加采样有效
2	PS 存在
3	PS 有效

config1 中的位 7 到位 4 可用于控制 SBR 和 PS 解码。位 5 和位 4 选择 SBR 模式，而位 7 和位 6 选择 PS 模式。如果你的 AAC 许可证没有隐藏 SBR 和/或 PS，这些配置就可以利用上。

config1(5:4)	用途
’00’	正常模式，增加采样 < 24kHz AAC 文件
’01’	不自动增加采样 < 24kHz AAC 文件，但在遇到 SBR 时激活增加采样
’10’	从不增加采样
’11’	放弃 SBR（同样放弃PS）

config1(7:6)	用途
’00’	正常模式，如果它是可用的则处理 PS
’01’	如果它是可用的则处理 PS，但是在降低采样模式下
’10’	保留
’11’	放弃 PS 处理

当 AAC 解码器发现一个文件用当前的时钟不能进行正确解码时，此解码器可以自动增加内部时钟。这个最大所允许的时钟是用 SCI_CLOCKF 寄存器来配置的。

如果甚至连所允许的最高的时钟也太慢以至于不能解码一个 AAC 文件的 SBR 和 PS 部件，这些高级解码器特性会逐个自动放弃，直到文件可以被播放为止。首先是参数立体声处理被放弃（这个放音将变成单声道）。如果这还不够，则频带复制会切换到降低采样模式（降低带宽）。最后一招是彻底放弃频带复制。在歌曲更换后，被放弃的特性会恢复。

9.11.4 Midi

参数	地址	用途
config1	0x1e03	各种不同的配置
	bits [3:0]	混响：0 =自动（ 时钟>= 3.0×时打开 ）， 1 = 关，2 - 15 = 空间大小
bytesLeft	0x1e2a/2b	此轨道内剩余的字节数量

config1 的低 4 位控制混响效果。

9.11.5 Ogg Vorbis

参数	地址	用途
增益	0x1e2a	首选的重放增益补偿

Ogg Vorbis 解码器支持重放增益技术。 重放增益技术用于自动地给所有歌曲配置一个相同的音量，使用户不需要在不同的歌曲之间调整音量设置。如果 Ogg Vorbis 解码器在歌曲标头中发现一个重放增益标记，则此标记被解析，并且解码的增益设置可以从增益参数中获取。对于其中没有任何重放增益标记的歌曲，缺省设定-6 dB (增益值 -12) 将被使用。关于重放增益的更多详细资料，请参阅 http://en.wikipedia.org/wiki/Replay_Gain 和 http://www.replaygain.org/。

播放软件可能使用增益值调整音量水平。负值表示这个音量是降低的，正值表示音量是增加的。

举例来说：增益= -11 表示音量将会降低 5.5 dB（−11⁄2 = −5.5），并且左和右的衰减应该是增加了 11。当增益= 2 表示音量将会增加 1 dB（2⁄2 = 1.0），并且左和右的衰减应该是减少了 2。由于音量设置不可以超越+0 dB，这个值应该是饱和值。

增益	音量	SCI_VOL（音量-增益）
-11 (-5.5 dB)	0 (+0.0 dB)	0x0b0b (-5.5 dB)
-11 (-5.5 dB)	3 (-1.5 dB)	0x0e0e (-7.0 dB)
+2 (+1.0 dB)	0 (+0.0 dB)	0x0000 (+0.0 dB)
+2 (+1.0 dB)	1 (-0.5 dB)	0x0000 (+0.0 dB)
+2 (+1.0 dB)	4 (-2.0 dB)	0x0202 (-1.0 dB)

9.12 SDI 测试

在 VS1053b 内有几个测试方式，允许用户将执行存储器测试， SCI 总线测试和几个不同的正弦波测试。

所有的测试用一个相似的方式开始： VS1053b 用硬件复位，设置 SM_TESTS，并将测试命令发送到 SDI 总线上。每个测试通过发送 4 字节的特别指令序列开始，跟随 4 个零。此序列见后面描述。

9.12.1 正弦测试

正弦测试初始化以 8 字节序列 0x53 0xEF 0x6E *n* 0 0 0 0， *n* 是用来定义正弦测试的地方。*n* 被定义如下：

n 位组		
名称	位域	说明
FsIdx	7:5	采样率索引
S	4:0	正弦跳跃率

FsIdx	*Fs*	*FsIdx*	*Fs*
0	44100 Hz	4	24000 Hz
1	48000 Hz	5	16000 Hz
2	32000 Hz	6	11025 Hz
3	22050 Hz	7	12000 Hz

将被输出的正弦频率可以按照后面的公式计算：$F = Fs \times S/128$。

例如：正弦测使用值 126 激活，那将是 0b01111110。拆开 *n* 的部件，*FsIdx* = 0b011 = 3，因此 *Fs* = 22050Hz。*S* = 0b11110 = 30，所以正弦的最终频率是 $F = 22050\text{Hz} \times 30/128 \approx 5168\text{Hz}$。

要退出正弦测试，发送这个系列 0x45 0x78 0x69 0x74 0 0 0 0 即可。

注意：正弦测试信号要经过数字音量控制，所以它可以单独测试不同的通道。

9.12.2 引脚测试

引脚测试以 8 字节序列 0x50 0xED 0x6E 0x54 0 0 0 0 来激活。 此测试仅用于芯片生产测试。

9.12.3 SCI 测试

SCI 测试以 8 字节序列 0x53 0x70 0xEE *n* 0 0 0 0 来初始化， n – 48 是测试寄存器的编号。指定寄存器的内容被读取和被复制到 SCI_HDAT0。如果将被测试的寄存器是 HDAT0，则结果将被复制到 SCI_HDAT1 内。

例如：如果 *n* 是 48，则 SCI 寄存器 0（SCI_MODE）的内容被拷贝到 SCI_HDAT0。

9.12.4 存储器测试

存储器测试方式以 8 字节序列 0x4D 0xEA 0x6D 0x54 0 0 0 0 来初始化的。在这个序列之后，等待 1100000 个时钟周期。 结果可以从 SCI 寄存器 SCI_HDAT0 读取，并且每个位被解释如下：

位域	屏蔽	含义
15	0x8000	测试完成
14:10		不使用
9	0x0200	Mux 测试成功
8	0x0100	好的 MAC RAM
7	0x0080	好的 I RAM
6	0x0040	好的 Y RAM
5	0x0020	好的 X RAM
4	0x0010	好的 I ROM 1
3	0x0008	好的 I ROM 2
2	0x0004	好的 Y ROM
1	0x0002	好的 X ROM 1
0	0x0001	好的 X ROM 2
	0x83ff	全部都是好的

存储器测试会覆盖掉 RAM 存储器中当前的内容。

9.12.5 新的正弦波和扫描测试

一个更加精确频率的正弦波测试可以通过 SCI 开始和控制。SCI_AICTRL0 和 SCI_AICTRL1 分别设置此正弦波频率的左右通道。 这些寄存器：音量(SCI_VOL)和采样率(SCI_AUDATA)在测试之前或之中，可以被设置。 向 SCI_AIADDR 写入 0x4020 开始测试。

SCI_AICTRLn 可以通过所期望的频率 *Fsin* 和 DAC 采样率 *Fs* 用下面这个公式来计算：

$$SCI\_AICTRLn = Fsin \times 65536/Fs$$

SCI_AICTRLn 的最大值是 0x8000U。 对产生正弦波的最佳 S/N 比率来说，SCI_AICTRLn 的三个 LSb 应该是零。频率结果 *Fsin* 可以用 DAC 采样率 *Fs* 和 SCI_AICTRL0、SCI_AICTRL1 从下面的等式计算出来。

$$Fsin = SCI\_AICTRLn \times Fs/65536$$

正弦扫描测试可以通过给 SCI_AIADDR 写入 0x4022 开始。

这两个测试使用了正常音频通道，因此 SCI_BASS、差分输出方式和 EarSpeaker 的设置都会起作用。

10 VS1053b 寄存器

10.1 谁需要阅读这些章节

当用户希望增加一些自己的功能象 DSP 效果到 VS1053b 时，就需要用户自编软件了。

然而，VS1053b 的多数用户不需要担心写他们的扩充代码或者关心本章节，包括从 VLSI Solution 的网站只下载软件插件的那些人。

10.2 处理器内核

VS_DSP 是一个 16/32 位 DSP 处理器内核，也有很多通用处理器的功能。VLSI Solution 的免费 VSKIT 软件包包含了所有的工具和所需的文件。可以为 VS_DSP 处理器内核写、模拟和调试汇编语言或扩展的 ANSI C 程序。

VLSI Solution 还提供了一个完整的和有全功能调试能力的集成开发环境 VSIDE。

10.3 VS1053b 存储器映射

X-memory		Y-memory		I-memory	
地址空间	说明	地址空间	说明	地址空间	说明
0x0000. . 0x17ff	系统 RAM	0x0000. . 0x17ff	系统 RAM	0x0000. . 0x004f	系统 RAM
0x1800. . 0x187f	用户 RAM	0x1800. . 0x187f	用户 RAM	0x0050. . 0x0fff	用户 RAM
0x1880. . 0x197f	堆栈	0x1880. . 0x197f	堆栈	0x1000. . 0x1fff	-
0x1980. . 0x3fff	系统 RAM	0x1980. . 0x3fff	系统 RAM	0x2000. . 0xffff	ROM 56k
0x4000. . 0xbfff	ROM 32k	0x4000. . 0xdfff	ROM 40k		和 banked
0xc000. . 0xc0ff	外设	0xe000. . 0xffff	系统 RAM	0xc000. . 0xffff	ROM4 16k
0xc100. . 0xffff	ROM 15.75k				

10.4 SCI 寄存器

在第 8.7 章中 SCI 寄存器的描述可以发现它在 0xC000..0xC00F 之间。 除这些寄存器外，有一个地址 0xC010，称为 SCI_CHANGE。

SCI 寄存器组，前缀是 SCI_				
寄存器	类型	复位值	缩写[位域]	说明
0xC010	r	0	CHANGE[5:0]	最近的 SCI 访问地址

SCI_CHANGE 位域		
名称	位域	说明
SCI_CH_WRITE	4	1 = 最近的访问是一个写周期
SCI_CH_ADDR	3:0	最近访问的 SCI地址

10.5 串行数据寄存器

SDI 寄存器，前缀是 SER_				
寄存器	类型	复位值	缩写[位域]	说明
0xC011	r	0	DATA	最近收到的 2 字节，大端模式
0xC012	w	0	DREQ[0]	DREQ 引脚控制

10.6 DAC 寄存器

DAC 寄存器，前缀是 DAC_				
寄存器	类型	复位值	缩写[位域]	说明
0xC013	rw	0	FCTLL	DAC 频率控制，16 LSbs
0xC014	rw	0	FCTLH	DAC频率控制 4MSbs，PLL 控制
0xC015	rw	0	LEFT	DAC_LEFT 通道 PCM 值
0xC016	rw	0	RIGHT	DAC_RIGHT通道 PCM 值

每到第四个时钟周期，内部的 26 位计数器会加上 (DAC_FCTLH & 15)*65536 + DAC_FCTLL。每当这个计数器溢出， DAC_LEFT 和 DAC_RIGHT 中的值可被读取，并引发一个 DAC 中断。

10.7 GPIO 寄存器

GPIO 寄存器，前缀是 GPIO_				
寄存器	类型	复位值	缩写[位域]	说明
0xC017	rw	0	DDR[7:0]	方向
0xC018	r	0	IDATA[7:0]	从引脚读取的数值
0xC019	rw	0	ODATA[7:0]	输出到引脚的数值

GPIO_DIR 用于设置 GPIO 的引脚方向，1 是输出。GPIO_ODATA 会记住它的数值，即使 GPIO_DIR 被设置成输入。

GPIO 寄存器不会引起中断。

注意：在 VS1053b 内，VSDSP 寄存器可以通过 SCI_WRAMADDR 和 SCI_WRAM 寄存器来读取和写入。因此你能很方便地使用 GPIO 引脚。

10.8 中断寄存器

中断寄存器，前缀是 INT_				
寄存器	类型	复位值	缩写[位域]	说明
0xC01A	rw	0	ENABLE[7:0]	中断使能
0xC01B	w	0	GLOB_DIS[-]	写增加中断计数器
0xC01C	w	0	GLOB_ENA[-]	写减少中断计数器
0xC01D	rw	0	COUNTER[4:0]	中断计数器

INT_ENABLE 控制中断。 控制位如下所示：

INT_ENABLE 位域		
名称	位元	说明
INT_EN_TIM1	7	使能定时器 1 中断
INT_EN_TIM0	6	使能定时器 0 中断
INT_EN_RX	5	使能 UART RX 中断
INT_EN_TX	4	使能 UART TX中断
INT_EN_SDI	2	使能数据中断
INT_EN_SCI	1	使能 SCI 中断
INT_EN_DAC	0	使能 DAC 中断

注意：在改变 INT_ENABLE 的对应中断后，在它起作用之前也许要花费最多 6 个时钟周期。

给 INT_GLOB_DIS 写入任何值会使中断计数器 INT_COUNTER 加一，并有效地让所有中断失效。写入这个寄存器后，在它生效之前也许要花费最多 6 个时钟周期。

给 INT_GLOB_ ENA 写入任何值会使中断计数器 INT_COUNTER 减一（除非 INT_COUNTER 已经是 0）。如果此中断计数器变成 0，由 INT_ENABLE 选中的中断将会被恢复。一个中断程序总是应该写入这个寄存器来作为它的最后一件事，因为中断自动地给中断寄存器加一，但减去它回到它的原始值是用户的责任。写入这个寄存器后，在它生效之前也许要花费最多 6 个时钟周期。

通过读取 INT_COUNTER，该用户也许会检查中断计数器是否正确。如果计数器不为 0，那些中断是失效的。

10.9 看门狗 v1.0 2002-08-26

该看门狗包括看门狗计数器和一些逻辑。在复位后看门狗是不活动的。可以向 WDOG_CONFIG 写入来给计数器设定重载值。通过向 WDOG_RESET 写入 0x4ea9 来激活看门狗。每次这样做的时候都会使看门狗计数器复位。每过 65536 个时钟周期，此计数器减一。如果计数器产生下溢，它将启动 VSDSP 内部的复位序列。

因此，在第一次 0x4ea9 写入 WDOG_RESET 之后，随后用相同的值给同一寄存器写入必须在不少于 65536×WDOG_CONFIG 个时钟周期内完成。

一旦启动，看门狗就不可以关闭了。同样，写入 WDOG_CONFIG 的计数器重载值也不可以修改了。

在激活看门狗之后，任何对 WDOG_CONFIG 或 WDOG_DUMMY 的读写操作将会导致下一个对 WDOG_RESET 的写操作失效。这是为了防止跑飞的循环去复位计数器，即使它们偶然写了一个正确值。写入一个错误值到 WDOG_RESET 也同样导致下一个对 WDOG_RESET 的写操作失效。

读取看门狗寄存器将返回不确定的值。

10.9.1 寄存器

看门狗寄存器，前缀 WDOG_				
寄存器	类型	复位值	缩写[位域]	说明
0xC020	w	0	CONFIG	配置
0xC021	w	0	RESET	时钟配置
0xC022	w	0	DUMMY[-]	虚的寄存器

10.10 UART v1.1 2004-10-09

RS232 UART 是使用 RS232 标准的串行接口。

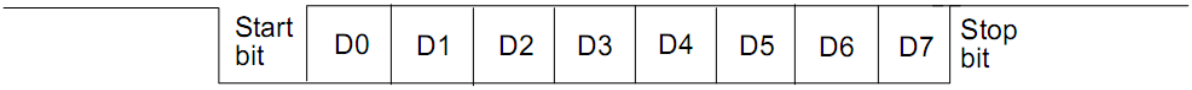


图 15: RS232 串行接口协议

当线路空闲时，它停留在逻辑高状态。当字节被传送时，传输从起始位(逻辑零)开始并继续传送数据位(LSB 在前)，最后用停止位(逻辑高)来结束。每个 8 位的字节帧会发送 10 位。

10.10.1 寄存器

UART 寄存器，前缀 UARTx_				
寄存器	类型	复位值	缩写[位域]	说明
0xC028	r	0	STATUS[4:0]	状态
0xC029	r/w	0	DATA[7:0]	数据
0xC02A	r/w	0	DATAH[15:8]	高位数据
0xC02B	r/w	0	DIV	分频

10.10.2 状态 UARTx_STATUS

读取状态寄存器，返回的是发送和接收的状态。

UARTx_STATUS 位域		
名称	位元	说明
UART_ST_FRAMEERR	4	帧错误（停止位是0）
UART_ST_RXORUN	3	接收过快
UART_ST_RXFULL	2	接收数据寄存器满
UART_ST_TXFULL	1	发送数据寄存器满
UART_ST_TXRUNNING	0	正在发送中

如果接收字节的停止位是 0，则 UART_ST_FRAMEERR 被设置。

当接收移位寄存器传送数据到数据寄存器时，如果一个接收字节覆盖了尚未读取的数据，则 UART_ST_RXORUN 被设置，否则它将被清除。

如果尚有未读取的数据在数据寄存器内，则 UART_ST_RXFULL 被设置。

如果数据寄存器不允许写入（数据寄存器满），则 UART_ST_TXFULL 被设置。

如果发送移位寄存器正在运作中，则 UART_ST_TXRUNNING 被设置

10.10.3 数据 UARTx_DATA

从 UARTx_DATA 读取的数据是标准的字节，内容在位 7:0，而位 15:8 返回的是‘0’。如果那儿没有未读取的数据，则“接收数据寄存器满”标记会被清除。

当一个字节从“接收移位寄存器”被移动到“接收数据寄存器”时，将产生一个接收中断。

向 UARTx_DATA 写入会设置为发送一个字节。 有效的数据在位元 7:0 上，写入的其它位元的值被忽略。 如果发送器是闲置的，则此字节立刻被移动到“发送移位寄存器”内和引发一个发送中断请求，并且开始发送数据。如果发送器是“忙”，将设置 UART_ST_TXFULL 标记，而该字节将停留在“发送数据寄存器”内，直到先前的那个字节被发送完，可以处理此字节为止。

10.10.4 数据高 8 位 UARTx_DATAH

除了数据位 15:8 被使用以外，和 UARTx_DATA 一样。

10.10.5 分频 UARTx_DIV

UARTx_DIV 位域		
名称	位元	说明
UART_DIV_D1	15:8	分频值1 (0..255)
UART_DIV_D2	7:0	分频值2 (6..255)

此分频器在复位时设定的初始值是 0x0000。ROM 内的起动代码必须根据准确的主钟频率初始化它，才能得到正确的位速度。第二个分频值（D_2）必须是在 6 到 255 的范围内。

通讯速度 $f = fm/((D_1+1)*(D_2))$，fm 是主时钟频率的和 f 是 TX/RX 的速度（bps）。

分频器值在 26 兆赫主时钟下的通用通信速度如下所示：

举例 UART 速度 fm = 26MHz		
通用速度 [bps]	UART_DIV_D1	UART_DIV_D2
4800	85	63
9600	42	63
14400	42	42
19200	51	26
28800	42	21
38400	25	26
57600	1	226
115200	0	226

10.10.6 中断和操作

发送操作如下所示：

在一个 8 位字写入传送数据寄存器后，如果发送器没有忙于发送先前的字节，则它立即被传输。当传输开始将会发送 TX_INTR 中断。状态位[1]标示出“发送数据寄存器空”(或“满”的状态)，而位[0]标示出发送器(移位寄存器)“空”的状态。如果“发送数据寄存器”不为“空”(bit [1] = ’0’)，则不能向“发送数据寄存器”写一个新字。当数据被移到发送器并开始传输，“发送数据寄存器”将是空的。在每次产生发送中断的时候，给“发送数据寄存器”写一个新的字是安全的。

接收操作如下所示：

它对 RX 信号线进行采样，如果侦测到高电平到低电平的转换，则起始位被找到。此后，它在 8 位元的每个位元时间中部采样(使用一个恒定的定时器)，和填充接收器（移位寄存器），接收顺序是 LSB 在前。完成后的数据在接收器移动到“接收数据寄存器”时，结束位状态被检查（逻辑高=好，逻辑低=帧错误）并设置状态位[4]，RX_INTR 中断被发送，(“接收数据寄存器满”）被设置，状态位[2]的旧状态被复制到位[3] （“接收数据过快”状态位）。随后接收器回到空闲状态等待一个新的起始位。当“接收数据寄存器”被读取时，状态位[2]被清零。

RS232 通信速度是使用二个时钟分频器来设置的。 基本的时钟是处理器主时钟（即 XTALI，而不是 CLKI）。位元 15-8 在这里是第一分频器，位元 7-0 为第二分频器。 RX 采样频率是从第二分频器输入的时钟频率。

10.11 定时器 v1.0 2002-04-23

有二个彼此独立的 32 位定时器，可以单独初始化和使能。如果使能一个定时器，通过处理器写入一个启动值来初始化它，定时器开始用每个时钟周期倒计时。

当数值被倒减到零时，一个中断被发送。并且定时器用它的启动数值寄存器中的数值初始化，然后继续倒计时。只要它在使能中，定时器将始终在那里循环。

定时器有专用的一个 32 位的倒计数定时寄存器和一个 32 位的 TIMER1_LH，可以通过处理器来写入定时器的启动值。同时还有一个两位的 TIMER_ENA 寄存器。每个定时器是通过设置使能寄存器的对应位来启动（1=使能）或停止（0=失效）的。

10.11.1 寄存器

定时器寄存器，前缀 TIMER_				
寄存器	类型	复位值	缩写[位域]	说明
0xC030	r/w	0	CONFIG[7:0]	定时器配置
0xC031	r/w	0	ENABLE[1:0]	定时器使能
0xC034	r/w	0	T0L	定时器0 启动值 -LSBs
0xC035	r/w	0	T0H	定时器0 启动值 -MSBs
0xC036	r/w	0	T0CNTL	定时器0 计数器 -LSBs
0xC037	r/w	0	T0CNTH	定时器0 计数器 -MSBs
0xC038	r/w	0	T1L	定时器1 启动值 -LSBs
0xC039	r/w	0	T1H	定时器1 启动值 -MSBs
0xC03A	r/w	0	T1CNTL	定时器1 计数器 -LSBs
0xC03B	r/w	0	T1CNTH	定时器1 计数器 -MSBs

10.11.2 配置 TIMER_CONFIG

TIMER_CONFIG 位域		
名称	位域	说明
TIMER_CF_CLKDIV	7:0	主时钟分频

TIMER_CF_CLKDIV 的主时钟分频值是用于产生所有的定时器时钟。产生这个内部时钟频率的公式为：$fi = fm/(c+1)$，其中 fm 是主时钟频率和 c 是 TIMER_CF_CLKDIV。例如：当主时钟是 12MHz、TIMER_CF_DIV=3，则主时钟将经过 4 分频，而输出/采样的时钟将是 fi = 12MHz/(3+1) = 3MHz。

10.11.3 配置 TIMER_ENABLE

TIMER_ENABLE 位域		
名称	位域	说明
TIMER_EN_T1	1	使能定时器1
TIMER_EN_T0	0	使能定时器0

10.11.4 定时器 X 起始值 TIMER_Tx[L/H]

这个 32 位的启动值 TIMER_Tx [L/H] 用来在定时器复位时设置最初的计数值。而定时器中断的频率可以用 *ft* = *fi*/(*c*+1)来计算，*fi* 是主时钟经过分频后的定时器时钟（请参阅章节 10.11.2），而 *c* 是 TIMER_Tx[L/H]。

例如：当主时钟频率为 12MHz 和 TIMER_CF_CLKDIV=3，则定时器时钟[注]为 *fi* = 3MHz。

如果 TIMER TH=0、TIMER TL=99，则时钟中断频率为 *ft* = 3MHz/(99+1) = 30kHz。

注：原文在此处将定时器时钟误写为主时钟。

10.11.5 定时器 X 计数器 TIMER_TxCNT[L/H]

TIMER_TxCNT [L/H] 包含了当前的计数值。 通过读取这个寄存器对，用户可以知道在下一个定时器中断产生之前还有多长时间。并且，通过写入这对寄存器，可以得到一次不同延迟时间的定时器中断。

10.11.6 中断

每个定时器有它自己的中断，在定时器计数下溢时会被声明。

10.12 VS1053b 音频通道

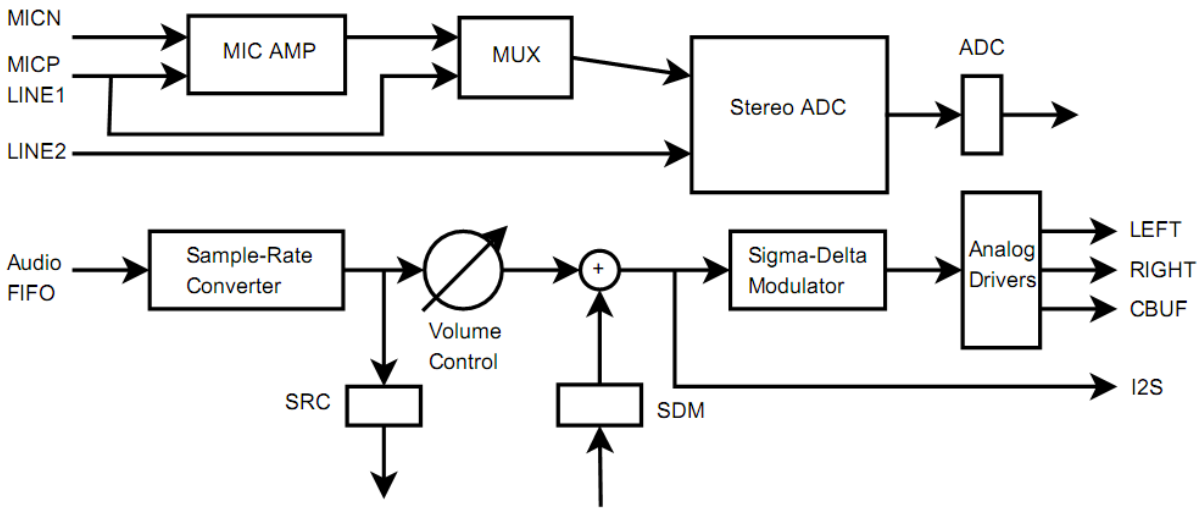


图 16：VS1053b 的 ADC 和 DAC 数据通道

在 IMA ADPCM 编码方式下，从模拟到数字的数据转换首先会用 48kHz 或 24kHz 采样率来处理。固件进行直流漂移补偿和增益控制（自动或固定），然后将数据转送到音频 FIFO。在这儿，数据被重新转换采样率，这只会产生几个采样的延迟。该采样率转换将增加采样数据到 XTALI/2（缺省时钟下是 6.144MHz），在那儿针对所需求的采样率进行 1×、2×或 3×的重采样。另外，IMA ADPCM 编码器或 PCM 采样会使用合适的频率，通过软件进行抽样来获得最终的数据。

10.13 I2S DAC 接口

I2S 接口可以在系统上添加外部 DAC。

注意：VS1053b 和 VS1033 不同，I2S 引脚共享的是不同的 GPIO 引脚，这能让 SPI 引导和 I2S 功能可以在同一个应用中使用。

10.13.1 寄存器

I2S 寄存器，前缀 I2S_				
寄存器	类型	复位	缩写	说明
0xC040	r/w	0	CONFIG[3:0]	I2S 配置

10.13.2 配置 I2S_CONFIG

I2S_CONFIG 位域		
名称	位域	说明
I2S_CF_MCLK_ENA	3	使能 MCLK 输出 (12.288 MHz)
I2S_CF_ENA	2	使能 I2S，否则引脚是 GPIO
I2S_CF_SRATE	1:0	I2S 速率， ”10” = 192, ”01” = 96, ”00” = 48 (kHz)

I2S_CF_ENA 使能 I2S 接口。在复位后此接口是失效的，那些引脚作为 GPIO 使用。

I2S_CF_MCLK_ENA 使能 MCLK 输出。 频率直接从输入时钟(通常是 12.288 兆赫)，或者输入时钟的一半（在输入 24-26 MHz 时钟，模式寄存器位 SM_CLK_RANGE 被设置为 1 时）。

I2S_CF_SRATE 用于控制输出采样率。 当设置为 48kHz， SCLK 是 MCLK/8，当设置为 96kHz 时，SCLK 是 MCLK/4，而设置为 192 kHz 时，SCLK 是 MCLK/2。

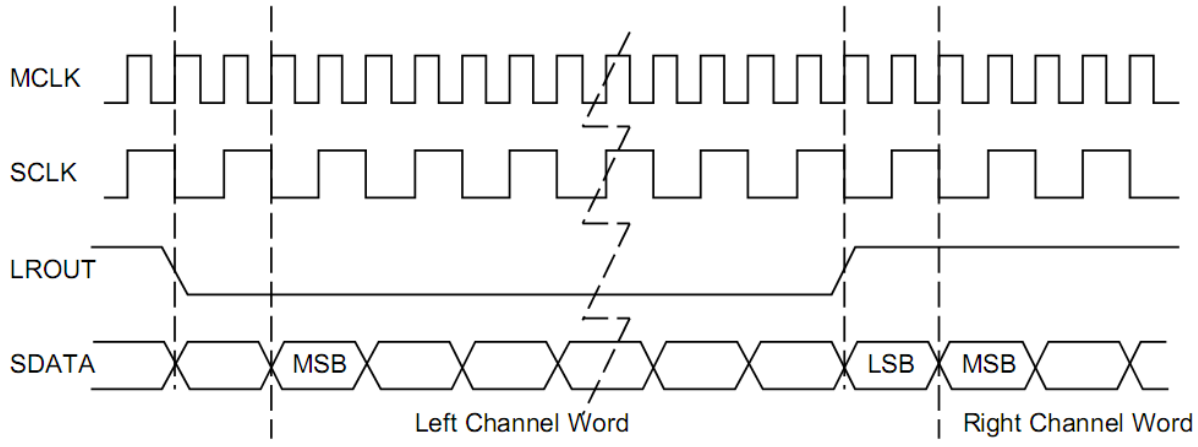


图 17：I2S 接口，192 kHz.

要使能 I2S，首先向 SCI_WRAMADDR 写入 0xc017 和向 SCI_WRAM 写入 0xf3，然后再向 SCI_WRAMADDR 写入 0xc040 和将 0x0c 写入 SCI_WRAM。请参阅应用注意事项中的更多信息。

11 VS1053 版本修订历史

本章描述对 VS1053 所作的最新和已经完成的很多重要改动。

11.1 在 VS1033c 和 VS1053a/b 之间修改的固件, 2007-03-08

全新的或主要的修改：

- I2S 引脚现在使用 GPIO4-GPIO7，并且与 SPI 引导引脚不再重叠。
- 当用法正确的时候，在文件之间不会出现软件复位的需求。
- 增加了 Ogg Vorbis 解码。非致命的 ogg 或 vorbis 解码错误会自动重新同步。 这允许很容易实现倒带和快进。如果到达“最后帧”标记或设置了 SM_CANCEL，则解码结束。
- HE-AAC v2 增加了第 3 级解码。它也许会使 PS 和 SBR 处理失效，和可以通过参数 parametric_x.control1 来控制“增加采样”模式。
- 象 WMA 解码器一样，如果 AAC 解码器需要更多时钟来解码文件，它也会使用时钟增加法(参见 SCI_CLOCKF)。如果此文件已经不能使用更高的时钟了，则 HE-AAC 特效会逐个被放弃，直到能正确解码。参量立体声会是首先被放弃的特效、随后是“减少采样”模式被使用，最后是频段复制特效被放弃。这些特效会在下个文件自动恢复。
- 全新的音量控制和过零交叉侦测功能可以阻止音量改变时产生的爆破音。
- 使用音频 FIFO 低容量侦测（支持逐渐减少到 0）来替代音频缓冲区内容循环。
- 为所有编解码器实行平均比特率计算(字节速率)。
- 所有的编解码器都支持快速播放方式，可为最佳质量选择速度来快速的向前操作。快速播放同时也会让 DECODE_TIME 加速。
- WMA 和 Ogg Vorbis 可以提供毫秒级的绝对解码位置。
- 当侦测到 SM_CANCEL 时，固件同时会放弃流缓冲区中的内容。
- 在当前文件中不能进行跳跃时，表示正在进行标头处理或是 midi 文件。SCI_STATUS 内的 SS_DO_NOT_JUMP [注]会是‘1’。（注：原文笔误，写作 SCIST_DO_NOT_JUMP）。
- IMA ADPCM 编码器现在支持立体声编码和可以选择采样率。

其它的修改或附加的

- 被音量和低音/高音控制计算所延迟的时间由相应的 SCI 操作开销来抵消。这种延迟的处理和新的音量控制硬件可以防止音频采样数据在音量变化期间出现失误。
- SCI_DECODE_TIME 只会在硬件和软件复位并后，直到允许文件被一个接一个播放或循环播放时才会被清除。
- 在 YRAM 的 0xe000..0xffff 读取和写入会增加 SCI_WRAMADDR/SCI_WRAM。
- 在复位后，resync 参数（parametric_x.resync）被设置为 32767，这将允许有限度的“重新同步”尝试（或通过设置 SM_CANCEL 来中止）。之前的运作可以通过在复位后写 0 到 resync 来恢复。
- WMA、AAC：更多健全的重新同步。
- WMA、AAC：如果重新同步被执行，广播方式将自动激活。广播方式使检查文件大小的功能失效，并且继续解码，直到 SM_CANCEL 被设置或执行复位。
- 高音控制修正（音量改变可能会导致产生错误）

- MPEG 阶层 I 单声道的修正。
- MPEG 阶层 II 解决了半速率解码（帧大小被错误计算了）。
- MPEG 阶层 II 解决了准确性问题，无效的组值会设置为 0。
- WAV 解析器现在可以跳过未知的 RIFF 集合。
- IMA ADPCM：最大的块大小现在是 4096 字节（4088 个立体声采样，8184 个单声道采样），因此现在可以播放 44100Hz 立体声了。
- 实时 midi：在复位时如果 GPIO0='0'、GPIO1='1'，即可启动此模式，而 GPIO2 和 3 则提供了 earSpeaker 的设置。
- NewSinTest() 和 NewSinSweep() 增加了 (AIADDR = 0x4020/0x4022) AICTRL0 和 AICTRL1 来设置左/右的正弦频率。
- 清除存储器是在 SPI 引导之前，而不是在 InitHardware() 内。

VS1053b 中已知的问题、故障或特征

- 设置音量会清除 S_REFERENCE_SEL 和 SS_AD_CLOCK 位，请参阅章节 8.7.2。
- 软件复位清除了 GPIO_DDR，这会影响到 I2S 的引脚。
- Ogg Vorbis 在开窗时偶而会溢出，这是音频上一个小故障引发的，已经有补丁可用。
- IMA ADPCM 编码需要一个短的补丁来启动。 补丁的用法在第 9.8.1 章中有说明。

12 文档版本修订历史

本章节描述本文中的重要变动。

VS1053b 的版本 1.01，2008-05-22

- 增加 IMA ADPCM 补丁到章节 9.8.1 中。

VS1053b 的版本 1.0， 2008-05-12

- 产品版本移走“PRELIMINARY”标签。
- 更新第 4 章内表格中的数值。
- 修改最小温度回到-30℃。
- 修改最大 SCI 速度为 CLKI/7。

VS1053b 的版本 0.5，2007-12-03

- Ogg Vorbis 即时录音文档在章节 9.7 中。
- 增加立体声 ADPCM 录音和一个例子到章节 9.8 中。
- 增加 WAV PCM 标头例子到章节 9.6 中。
- 简明的 LQFP-48 接线图（第 6 章）移走了图片中的一张。

VS1053b 的版本 0.4， 2007-09-06

- 第一个公开版本。
- 完全重写的第 9.5 章节：操作/播放和解码。新的设计应该是基于此新版本上。
- 更新图片 3：用于 LQFP-48 的典型接线图。
- 重命名 SM_OUTOFWAV 、SM_CANCEL。
- 很多次要的修正，向象错别字更正等。

VS1053a 的版本 0.3，2007-01-16

- I2S 引脚现在是 GPIO4-GPIO7，它们不再和 GPIO0、GPIO1 重叠。
- 附加参数的结构更新（版本 3），位置修改到 X:0x1e00。

13 联系信息

VLSI Solution Oy
Entrance G, 2nd floor
Hermiankatu 8
FIN-33720 Tampere
FINLAND

Phone: +358-3-3140-8200
Fax: +358-3-3140-8288
Email: sales@vlsi.fi
URL: http://www.vlsi.fi/