VS1001k - MPEG AUDIO CODEC

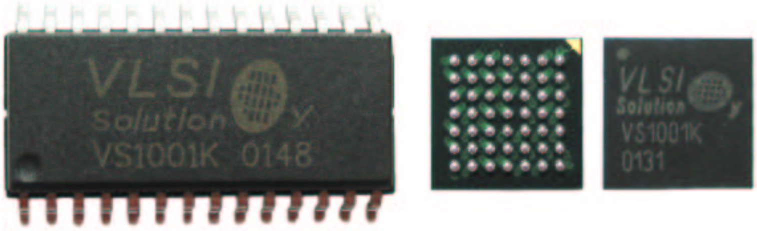
Features

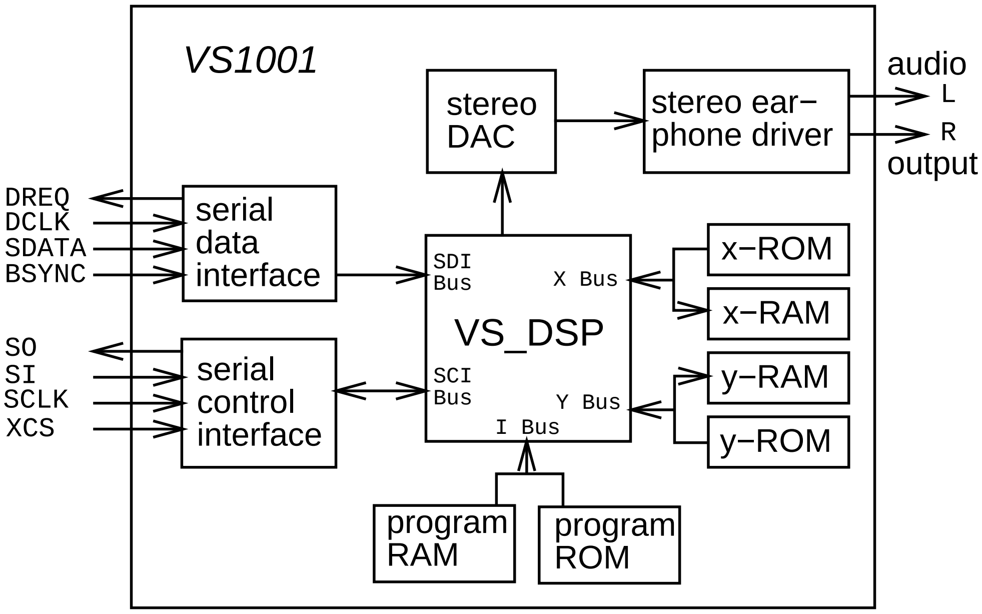
- MPEG audio layer 1, 2, and 3 decoder (ISO 11172-3)
- Supports MPEG 1 & 2 for all layers, and Layer 3's 2.5 extensions, and all their sample rates and bitrates, in mono or stereo
- PCM input mode
- Supports VBR (variable bitrate) for MP3
- Can be used as a slave co-processor
- Operates with single clock 12.288..16 MHz or 24.576..32 MHz.
- Extremely low-power operation
- On-chip high-quality stereo DAC with no phase error between channels
- Internal Op-Amp in BGA-49 package
- Stereo earphone driver capable of driving a 30Ω load.
- Separate 2.5 .. 3.6V operating voltages for analog and digital
- 4 kB On-chip RAM for user code
- Serial control and data interfaces
- New functions may be added with software

Description

VS1001k is a single-chip solution for an MPEG layer 1, 2 and 3 audio decoder. The chip contains a high-performance low-power DSP processor core (VS_DSP), working memory, 4 kBytes program RAM and 0.5 kBytes data RAM for user applications, serial control and input data interfaces, and a high-quality oversampling variable-sample-rate stereo DAC, followed by an earphone amplifier and a ground buffer.

VS1001k receives its input bitstream through a serial input bus, which it listens to as a system slave. The input stream is decoded and passed through a analog/digital hybrid volume control to an 18-bit oversampling, multi-bit, sigma-delta DAC. The decoding is controlled via a serial control bus. In addition to the basic decoding, it is possible to add application specific features, like DSP effects, to the user RAM memory.





Contents

List of Figures

1 License

MPEG Layer-3 audio decoding technology licensed from Fraunhofer IIS and THOMSON multimedia.

Supply of this product only conveys a license for private, non-commercial use.

2 Characteristics & Specifications

Unless otherwise noted: AVDD=2.9..3.6V, DVDD=2.3..3.6V, TA=-30..+85°C, XTALI=24.576MHz, Full-Scale Output Sinewave at 1.526 kHz, measurement bandwidth 20..20000 Hz, analog output load 30Ω (no ground buffer) or 100Ω (with ground buffer), bitstream 128 kbits/s, local components as shown in Figures 4 and 5.

Note, that some analog values are in practice better than in these tables if chips are used within a limited temperature range and not too close to lower voltage limits.

2.1 Analog Characteristics

Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
DAC Resolution			16		bits
Total Harmonic Distortion	THD		0.1	0.2	%
Dynamic Range (DAC unmuted, A-weighted)	IDR		90		dB
S/N Ratio (full scale signal)	SNR	70	87		dB
Interchannel Isolation		50	75		dB
Interchannel Gain Mismatch		-0.5		0.5	dB
Frequency Response		-0.1		0.1	dB
Frequency Response, AVDD = 2.8V		-0.3		0.3	dB
Full Scale Output Voltage (Peak-to-peak)		1.4	1.8[1]	2.0	Vpp
Deviation from Linear Phase				5	°
Out of Band Energy			-60		dB
Out of Band Energy with Analog Filter			-90		dB
Analog Output Load Resistance, no ground buffer	AOLR1	16	30[2]		Ω
Analog Output Load Resistance, ground buffer	AOLR2	16	100[2]		Ω
Analog Output Load Capacitance				1000	pF

[1] 3.6 volts can be achieved with +-to-+ wiring for mono difference sound.

[2] AOLR1/2 may be much lower, but below *Typical* distortion performance may be compromised.

2.2 Power Consumption

Parameter	Symbol	Min	Typ	Max	Unit
Power Supply Rejection			40		dB
Power Supply Consumption AVDD, Reset			0.6	5.0	μA
Power Supply Consumption AVDD, no load		3.0	4.5	6.0	mA
Power Supply Consumption AVDD, output loaded at 30Ω		4.0	5.5	40.0	mA
Power Supply Consumption AVDD, o. @ 30Ω + GND-buf.		6.0	7.5	40.0	mA
Power Supply Consumption DVDD, Reset			3.7	100.0	μA
Power Supply Consumption DVDD			15.0		mA

2.3 DAC Interpolation Filter Characteristics

Parameter	Symbol	Min	Typ	Max	Unit
Passband (to -3dB corner)		0		0.459Fs	Hz
Passband (Ripple Spec)		0		0.420Fs	Hz
Passband Ripple				±0.056	dB
Transition Band		0.420Fs		0.580Fs	Hz
Stop Band		0.580Fs			Hz
Stop Band Rejection		90			dB
Group Delay			15/Fs		s

Fs is conversion frequency

2.4 DAC Interpolation Filter Characteristics

Parameter	Symbol	Min	Typ	Max	Unit
-3 dB bandwidth		300			kHz
Passband Response at 20 kHz		-0.05			dB

2.5 Absolute Maximum Ratings

Parameter	Symbol	Min	Max	Unit
Analog Positive Supply	AVDD	-0.3	3.6	V
Digital Positive Supply	DVDD	-0.3	3.6	V
Current at Any Digital Output			±50	mA
Voltage at Any Digital Input		DGND-1.0	DVCC+1.0	V
Operating Temperature		-30	+85	°C
Functional Operating Temperature		-40	+95	°C
Storage Temperature		-65	+150	°C

2.6 Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Unit
Analog and Digital Ground	AGND DGND		0.0		V
Positive Analog	AVDD	2.5[1]	3.0	3.6	V
Ambient Operating Temperature		-30		+85	°C

[1] If AVDD is below 2.8 V, distortion performance may be compromised.

The following values are to be used when the clock doubler is active:

Parameter	Symbol	Min	Typ	Max	Unit
Positive Digital	DVDD	2.3	2.7	3.6	V
Input Clock Frequency	XTALI		12.288	13	MHz
Internal Clock Frequency[1]	CLKI		24.576	26	MHz

[1] The maximum sample rate that may be decoded with correct speed is CLKI/512.

The following values are to be used when the clock doubler is inactive:

Parameter	Symbol	Min	Typ	Max	Unit
Positive Digital	DVDD	2.3	2.7	3.6	V
Input Clock Frequency	XTALI		24.576	26	MHz
Internal Clock Frequency[1]	CLKI		24.576	26	MHz

[1] The maximum sample rate that may be decoded with correct speed is CLKI/512.

Note: With higher than typical voltages, VS1001k may operate with CLKI upto 30..32 MHz. However, the chips are not qualified for this kind of usage. If necessary, VLSI Solution Oy can qualify chips for higher clock rates for quantity orders.

2.7 Digital Characteristics

Parameter	Symbol	Min	Typ	Max	Unit
High-Level Input Voltage		0.7DVDD			V
Low-Level Input Voltage				0.3DVDD	V
High-Level Output Voltage at I_O = -2.0 mA		0.7DVDD			V
Low-Level Output Voltage at I_O = 2.0 mA				0.3DVDD	V
Input Leakage Current				1.0	μA

2.8 Switching Characteristics - Clocks

Parameter	Symbol	Min	Typ	Max	Unit
Master Clock Frequency [1]	XTALI		12.288		MHz
Master Clock Frequency [2]	XTALI		24.576		MHz
Master Clock Duty Cycle		40	50	60	%
Clock Output	XTALO		XTALI		MHz

[1] Clock doubler active.

[2] Clock doubler inactive.

2.9 Switching Characteristics - DREQ Signal

Parameter	Symbol	Min	Typ	Max	Unit
Data Request Signal	DREQ			200	ns

2.10 Switching Characteristics - SPI Interface Output

Parameter	Symbol	Min	Typ	Max	Unit
SPI Input Clock Frequency				0.25×CLKI	MHz
Rise time for SO				100	ns

2.11 Switching Characteristics - Boot Initialization

Parameter	Symbol	Min	Max	Unit
_RESET active time		2		XTALI
_RESET inactive to software ready			50000	XTALI

2.12 Short-Circuiting Analog Outputs

Although not a recommended practise for prolonged times, short-circuiting analog outputs and/or ground does not cause physical harm to the chip. Nevertheless, if using an earphone connection, high electrical charges may harm VS1001k's analog outputs. Thus it is recommended to follow the safety precautions described in the Application Notes at Chapter 17.

2.13 Using an External Clock

If there is a pulse or sine clock signal available from some other device, it may be connected to pin XTALI. In this case, leave XTALO unconnected, i.e. floating. Note, that unless the clock is stopped while reset is active, reset power consumption may be radically different from what is presented in Chapter 2.1. Note also that when the chip is in hardware reset, XTALO and XTALI are grounded. Thus, if there is an external clock that doesn't go to ground when VS1001 is reset, you should use safety resistors between the clock source and XTALO and XTALI.

3 Packages and Pin Descriptions

3.1 SOIC-28

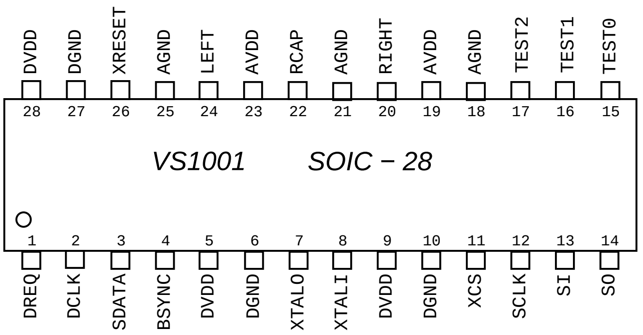


Figure 1: Pin Configuration, SOIC-28.

Pin Name	Pin	Type	Function
DREQ	1	DO	data request, input bus
DCLK	2	DIO	serial input data bus clock
SDATA	3	DI	serial data input
BSYNC	4	DI	byte synchronization signal
DVDD1	5	PWR	digital power supply
DGND1	6	PWR	digital ground
XTALO	7	CLK	crystal output
XTALI	8	CLK	crystal input
DVDD2	9	PWR	digital power supply
DGND2	10	PWR	digital ground
XCS	11	DI	chip select input (active low)
SCLK	12	DI	clock for serial bus
SI	13	DI	serial input
SO	14	DO3	serial output
TEST0	15	DI	reserved for test, connect to DVDD
TEST1	16	DIO	reserved for test, do NOT connect!
TEST2	17	DIO	reserved for test, do NOT connect!
AGND1	18	PWR	analog ground
AVDD1	19	PWR	analog power supply
RIGHT	20	AO	right channel output
AGND2	21	PWR	analog ground
RCAP	22	AIO	capacitance for reference
AVDD2	23	PWR	analog power supply
LEFT	24	AO	left channel output
AGND3	25	PWR	analog ground
XRESET	26	DI	active low asynchronous reset
DGND3	27	PWR	digital ground
DVDD3	28	PWR	digital power supply

Pin types:

- DI = digital input, CMOS Input Pad
- DO = digital output, CMOS Input Pad
- DIO = digital input/output
- DO3 = digital output, CMOS Tri-stated Output Pad
- AO = analog output
- CLK = clock/chrystal connection
- PWR = power supply pin

SOIC-28 package dimensions can be found at http://www.vlsi.fi/vs1001/ait_soic28.pdf .

3.2 BGA-49

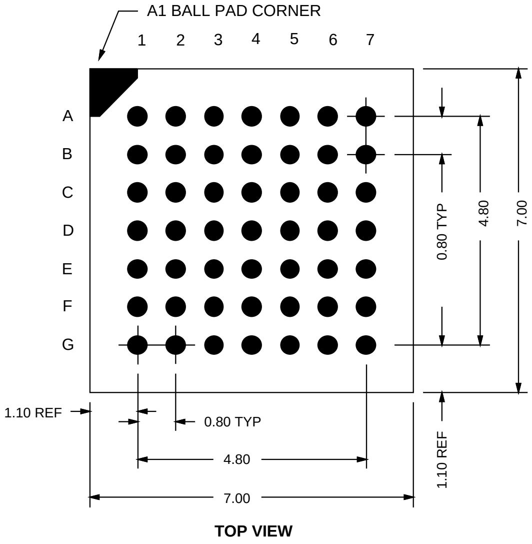
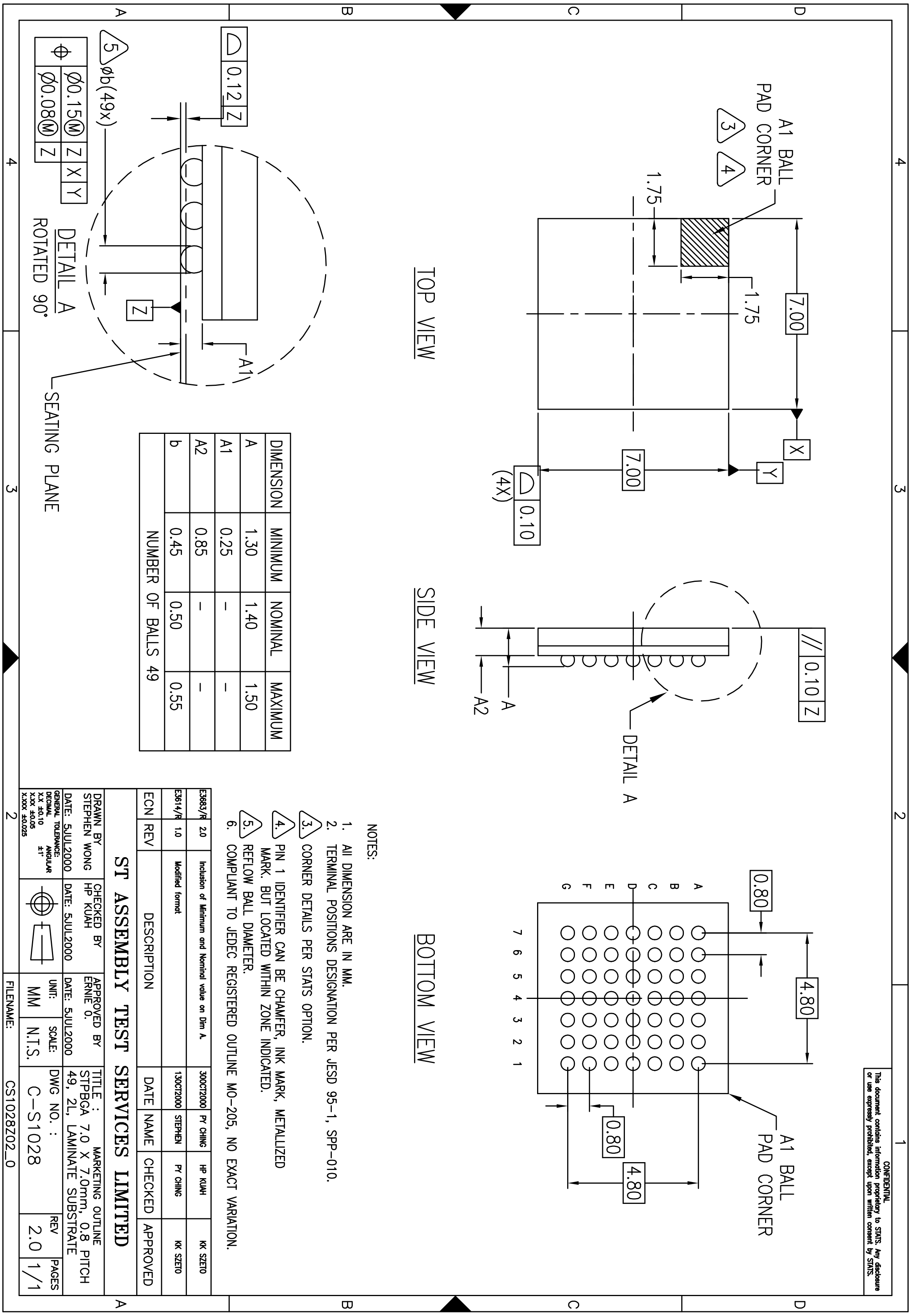


Figure 2: Pin Configuration, BGA-49.

Pin Name	Ball	Type	Function
BSYNC	E3	DI	byte synchronization signal
DVDD1	F3	PWR	digital power supply
DGND1	F4	PWR	digital ground
XTAL0	G3	CLK	crystal output
XTALI	E4	CLK	crystal input
DVDD2	F5	PWR	digital power supply
DGND2	F6	PWR	digital ground
XCS	G6	DI	chip select input (active low)
SCLK	D6	DI	clock for serial bus
SI	E7	DI	serial input
SO	D5	DO3	serial output
TEST0	C6	DI	reserved for test, connect to DVDD
TEST1	C7	DIO	reserved for test, do NOT connect!
TEST2	B6	DIO	reserved for test, do NOT connect!
AGND1	C5	PWR	analog ground
AVDD1	B5	PWR	analog power supply
RIGHT	A6	AO	right channel output
AGND34	B4	PWR	analog ground
GBGND	A5	PWR	analog ground for ground buffer
GBUF	C4	PWR	ground buffer
GBVDD	A4	PWR	analog power supply for ground buffer
RCAP	B3	AIO	capacitance for reference
AVDD45	A3	PWR	analog power supply
LEFT	B2	AO	left channel output
AGND56	A2	PWR	analog ground
XRESET	B1	DI	active low asynchronous reset
DGND3	D2	PWR	digital ground
DVDD3	D3	PWR	digital power supply
DREQ	E2	DO	data request, input bus
DCLK	E1	DIO	serial input data bus clock
SDATA	F2	DI	serial data input

For pin types, look at Chapter 3.1.



DIMENSION	MINIMUM	NOMINAL	MAXIMUM
A	1.30	1.40	1.50
A1	0.25	–	–
A2	0.85	–	–
b	0.45	0.50	0.55
NUMBER OF BALLS 49			

NOTES:

1. All DIMENSION ARE IN MM.
2. TERMINAL POSITIONS DESIGNATION PER JESD 95–1, SPP–010.
3. CORNER DETAILS PER STATS OPTION.
4. PIN 1 IDENTIFIER CAN BE CHAMFER, INK MARK, METALLIZED MARK. BUT LOCATED WITHIN ZONE INDICATED.
5. REFLOW BALL DIAMETER.
6. COMPLIANT TO JEDEC REGISTERED OUTLINE MO–205, NO EXACT VARIATION.

ECN	REV	DESCRIPTION	DATE	NAME	CHECKED	APPROVED
E3683/R	2.0	Inclusion of Minimum and Nominal value on Dim A.	30OCT2000	PY CHING	HP KUAH	KK SZETO
E3614/R	1.0	Modified format	13OCT2000	STEPHEN	PY CHING	KK SZETO

Figure 3: Package, BGA-49.

4 Connection Diagram, SOIC-28

In this connection diagram, a SOIC-28 -packaged VS1001k is used.

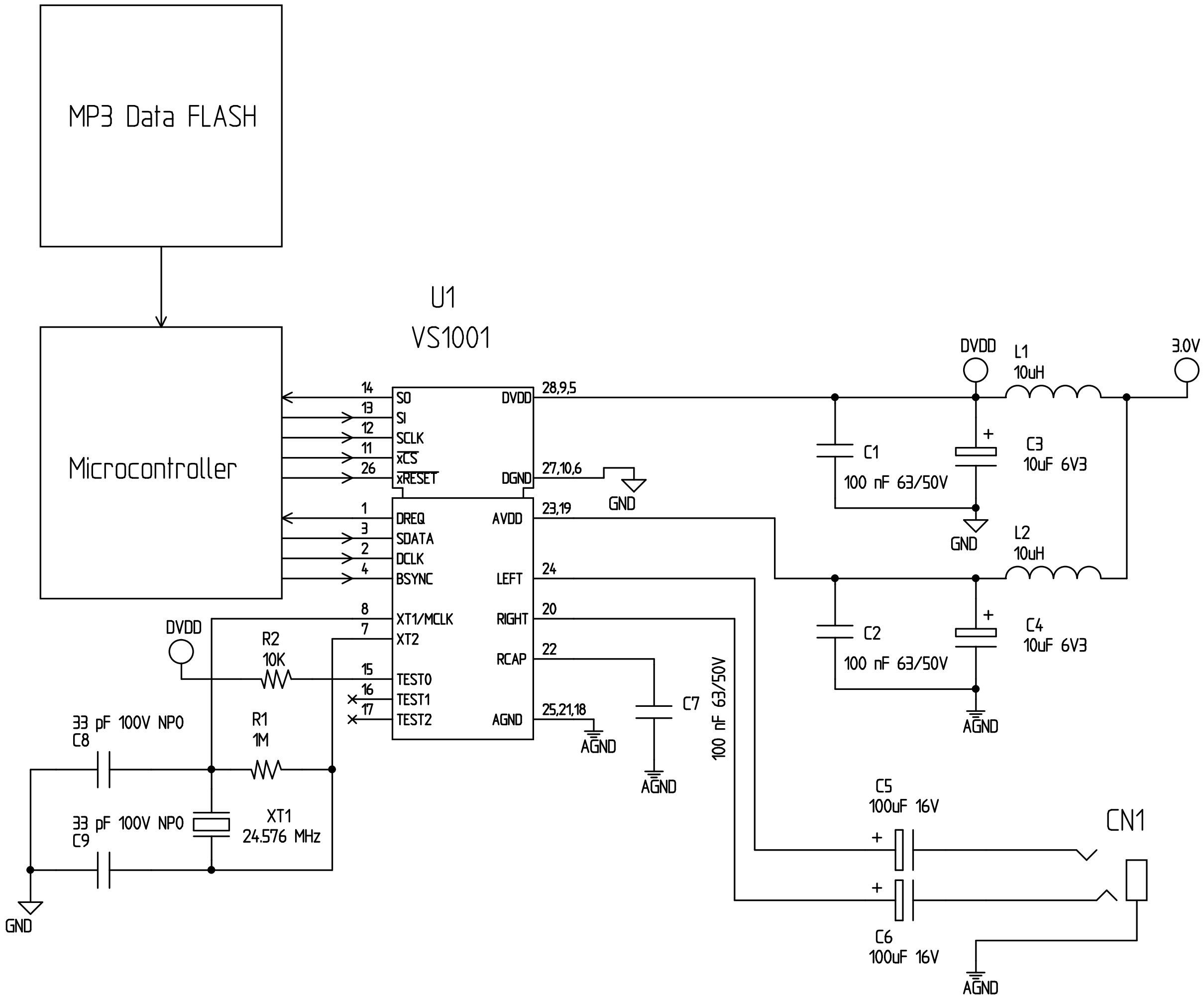


Figure 4: Typical Connection Diagram Using SOIC-28.

Ground buffer is not available for the SOIC-28 package; hence it is not used.

5 Connection Diagram, BGA-49

In this connection diagram, a BGA-49 -packaged VS1001k is used. In this picture, ground buffer is active.

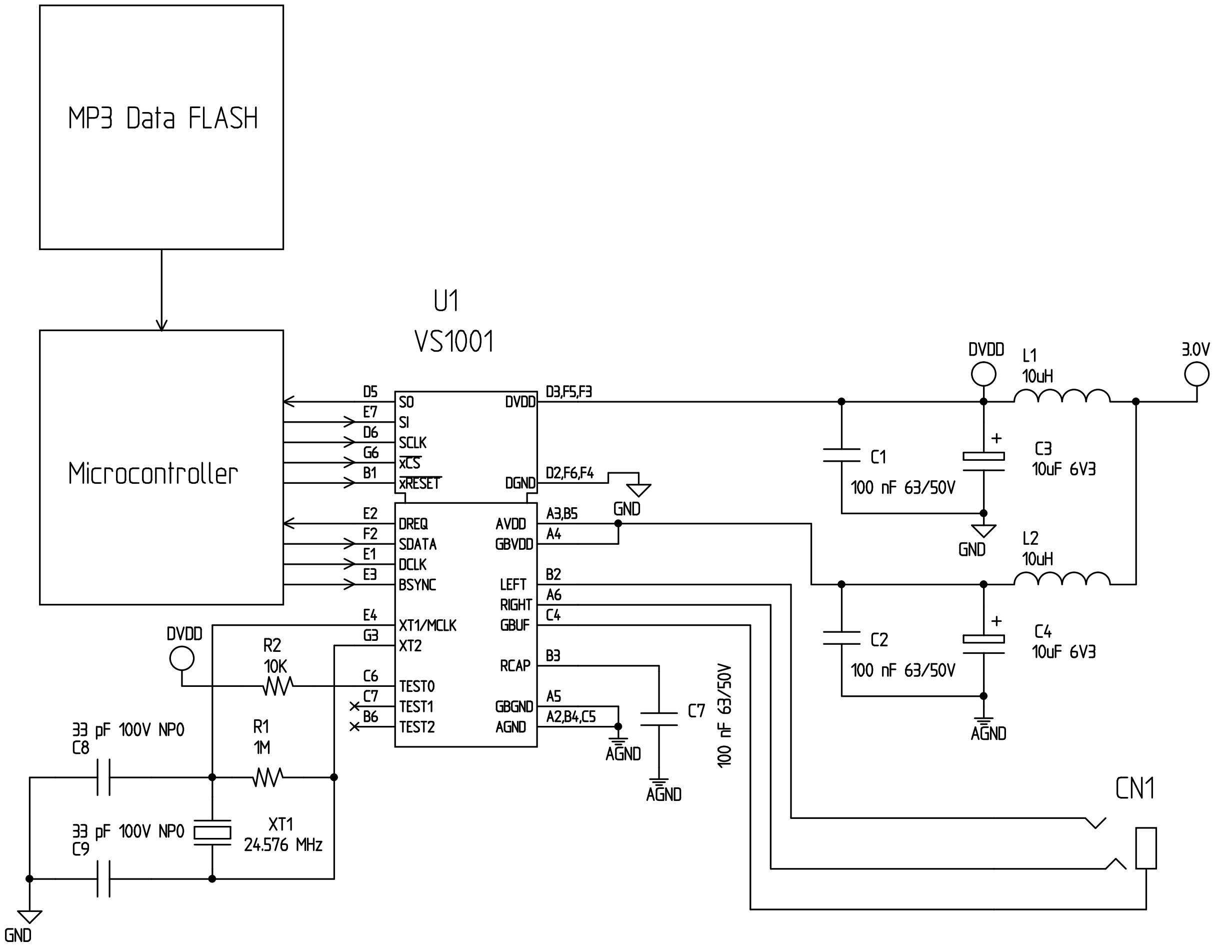


Figure 5: Typical Connection Diagram Using BGA-49.

Ground buffer GBUF can be used for common voltage (1.37 V) for earphones. This will eliminate the need for large isolation capacitors on line outputs, and thus the audio output pins from VS1001k may be connected directly to the earphone connector. If GBUF is not used, GBGND and GBVDD should not be connected.

6 SPI Buses

6.1 General

The SPI Bus - that was originally used in some Motorola devices - has been used for both VS1001k's Serial Data Interface SDI (C.f. Chapters 6.3 and 7.3) and Serial Control Interface SCI (C.f. Chapters 6.4 and 7.4).

6.2 SPI Bus Pin Descriptions

SDI Pin	SCI Pin	Description
-	XCS	Active low chip select input. A high level forces the serial interface into standby mode, ending the current operation. A high level also forces serial output (SO) to high impedance state. There is no chip select for SDI, which is always active.
DCLK	SCK	Serial clock input. The serial clock is also used internally as the master clock for the register interface. SCK can be gated or continuous. In either case, the first rising clock edge after XCS has gone low marks the first bit to be written (clock 0 in the following figures).
SDATA	SI	Serial input. SI is sampled on the rising SCK edge, if XCS is low.
-	SO	Serial output. In reads, data is shifted out on the falling SCK edge. In writes SO is at a high impedance state.

6.3 Serial Protocol for Serial Data Interface (SDI)

The serial data interface can operate in either master or slave mode. In master mode, VS1001k generates the DCLK signal, which can be selected to be either 512 or 1024 kHz. In slave mode, the DCLK signal is generated by an external circuit.

The data (SDATA signal) can be clocked in at either the rising or falling edge of the DCLK. (C.f. Chapter 7.5).

The VS1001k chip assumes its input to be byte-sychronized. I.e. the internal operation of the decoder does not search for byte synchronization of the frames from the data stream, but instead assumes the data to be correctly byte-aligned. The bytes can be transmitted either MSB or LSB first, depending of contents of SCI register MODE (C.f. Chapter 7.5).

To ensure correct byte-alignment of the input bitstream, the serial data interface has a BSYNC signal. The first DCLK sampling edge (rising or falling, depending on selected polarity), during which the BSYNC is high, marks the first bit of a byte (LSB, if LSB-first order is used, MSB, if MSB-first order is used). If BSYNC is not used, it must be tied to VCC externally and the master of the input serial interface must always sustain the correct byte-alignment. Using BSYNC is strongly recommended. For more details, look at the Application Notes in Chapter 16.

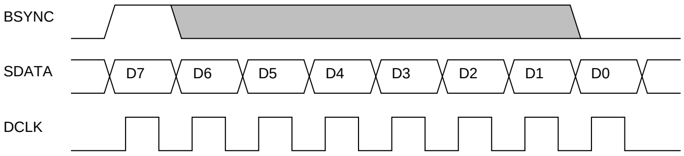


Figure 6: BSYNC Signal.

The DREQ signal of the data interface is used in slave mode to signal if VS1001k's FIFO is capable of receiving more input data. If DREQ is high, VS1001k can take at least 32 bytes of data. When there is less than 32 bytes of free space, DREQ is turned low, and the sender should stop transferring new data. Because of the 32-byte safety area, the sender may send upto 32 bytes of data at a time without checking the status of DREQ, making controlling VS1001k easier for low-speed microcontrollers.

Note: DREQ may turn low or high at any time, even during a byte transmission. Thus, DREQ should only be used to decide whether to send more bytes. It should not abort a transmission that has already started.

6.4 Serial Protocol for Serial Command Interface (SCI)

6.4.1 General

The serial bus protocol for the Serial Command Interface SCI (c.f. Chapter 7.4) consists of an instruction byte, address byte and one 16-bit data word. Each read or write operation can read or write a single register.

The operation is specified by an 8-bit instruction opcode. The supported instructions are read and write. See table below.

Instruction		
Name	Opcode	Operation
READ	0000 0011	Read data
WRITE	0000 0010	Write data

Note: After using the Serial Command Interface, it is not allowed to send SCI or SDI data for 5 microseconds.

6.4.2 SCI Read

VS1001k registers are read by the following sequence. First, XCS line is pulled low to select the device. Then the READ opcode (0x3) is transmitted via the SI line followed by an 8-bit word address. After the address has been read in, any further data on SI is ignored. The 16-bit data corresponding to the received address will be shifted out onto the SO line.

XCS should be driven high after the data has been shifted out. In that case, the word address will be incremented and data corresponding to the next address will be shifted out. After the last word has been shifted out, XCS should be driven high to end the READ sequence.

Word read is shown in Figure 7.

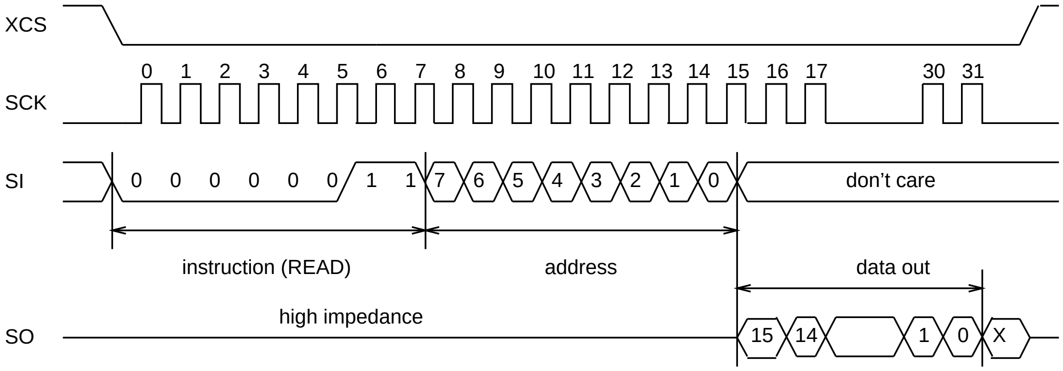


Figure 7: SCI Word Read

6.4.3 SCI Write

VS1001k registers are written by the following sequence. First, XCS line is pulled low to select the device. Then the WRITE opcode (0x2) is transmitted via the SI line followed by an 8-bit word address.

After the word has been shifted in, XCS should be pulled high to end the WRITE sequence. XCS low to high transition must occur after SCLK high to low transition corresponding to LSB of the last word.

Single word write is shown in Figure 8.

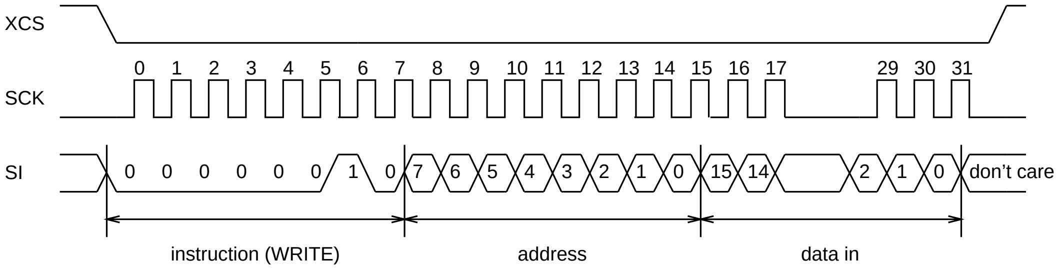


Figure 8: SCI Word Write

6.5 SPI Timing Diagram

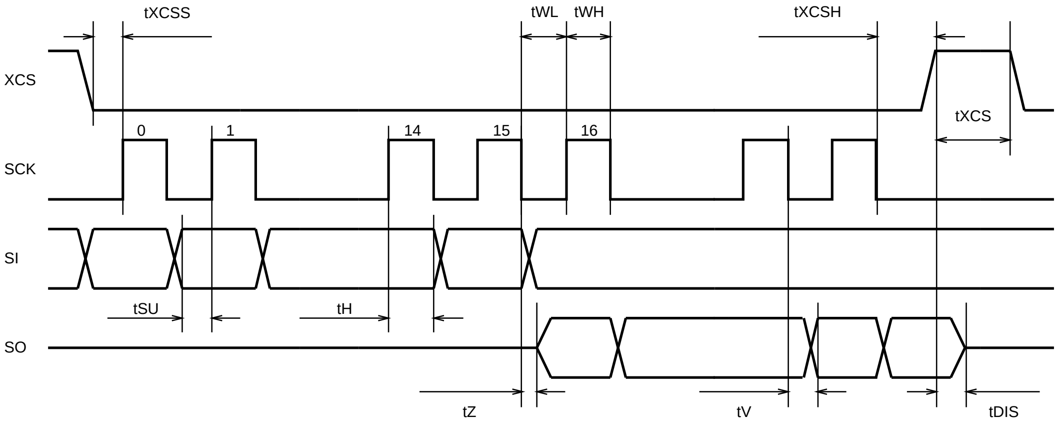


Figure 9: SPI Timing Diagram.

Symbol	Min	Max	Unit
tXCSS	5		ns
tSU	10		ns
tH	42		ns
tZ		42	ns
tWL	100		ns
tWH	100		ns
tV		42	ns
tXCSH	10		ns
tXCS	2		XTALI cycles
tDIS		1	XTALI cycles

Note: As tXCS must be at least 2 clock cycles, the maximum safe speed for the SPI bus is 1/4 of VS1001k's internal clock speed. For details, see Application Note in Chapter 18.

7 Functional Description

7.1 Main Features

VS1001k is based on a proprietary digital signal processor, VS_DSP. It contains all the code and data memory needed for MPEG audio decoding, together with serial interfaces, a multirate stereo audio DAC and analog output amplifiers and filters.

VS1001k can play all MPEG 1 and 2 layer I, II and III files, as well as so-called MPEG 2.5 layer III extension files with all sample rates and bitrates. In addition, variable bitrate (VBR) is also supported. With VBR, and depending on the song, near-cd quality can be achieved with approximately 100 kbits/s for stereo music sampled at 44100 Hz, whereas old encoders required 128 kbits/s for the same task. As both commercial and free (*http://www.lame.org/*) high-quality VBR encoders are nowadays widely available, MP3 format is getting better as it is maturing.

7.2 Data Flow of VS1001k

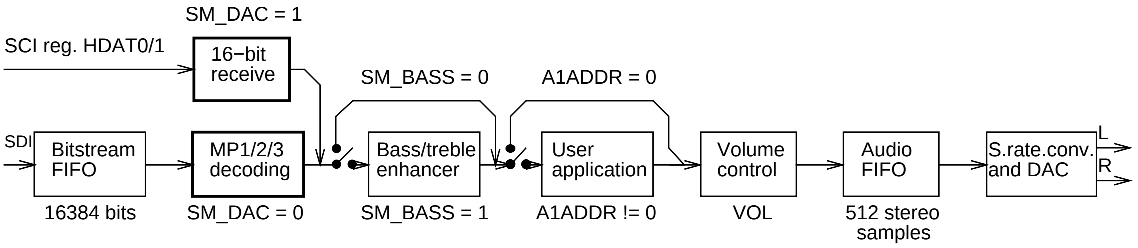


Figure 10: Data Flow of VS1001k.

First, depending on SCI register MODE's bit SM_DAC, MP3 data is input through the SDI bus, or linear 16-bit samples are provided through the SCI bus (Chapter 8.7).

After decoding, data may be sent to the Bass/treble enhancer depending on SCI register MODE's bit SM_BASS.

Then, if SCI register A1ADDR is non-zero, application code is executed from the address pointed to by A1ADDR. For more details, see Chapters 7.5.10 and 7.6.

After the optional user application, the signal is fed to the volume control unit, which also copies the data to the Audio FIFO.

The Audio FIFO holds the data that is read by the Audio interrupt (Chapter 10.5.1) and fed to the sample rate converter and DACs. The size of the audio FIFO is 512 stereo (2×16-bit) samples

The sample rate converter converts all different sample rates to CLKI/512 and feeds the data to the DAC, which in order makes a stereo in-phase signal. This signal is then forwarded to the earphone amplifier.

7.3 Serial Data Interface (SDI)

The serial data interface is meant for transferring compressed MPEG audio data.

Also several different tests may be activated through SDI as described in Chapter 8.

7.4 Serial Control Interface (SCI)

The serial control interface is compatible with the SPI-bus specification. Data transfers are always 16-bits. The VS1001k is controlled by writing and reading the registers of the interface.

The main controls of the control interface are:

- control of the operation mode
- uploading user programs
- access to header data
- status information
- access to decoded digital data
- feeding input data

7.5 SCI Registers

Name	Type	addr	Function
MODE	RW	0	mode control
STATUS	RW	1	status of VS1001k
INT_FCTLH	-	2	internal register, never use
CLOCKF	RW	3	clock freq + doubler
DECODE_TIME	R	4	decode time in seconds
AUDATA	R	5	misc. audio data
WRAM	W	6	RAM write program
WRAMADDR	W	7	base address for RAM write
HDAT0	R	8	read header data
HDAT1	R	9	read header data
A1ADDR	RW	10	start address of application
VOL	RW	11	volume control
RESERVED	-	12	reserved for VS1002 use, don't touch
A1CTRL[x]	RW	13+x	2 application control registers

x = [0 .. 1]

All registers are filled with zeros at hardware reset.

7.5.1 MODE (RW)

MODE is used to control the operation of VS1001k.

Bit	Name	Function	Value	Description
0	SM_DIFF	differential	0 1	normal in-phase audio left channel inverted
1	SM_FFWD	fast forward	0 1	normal playback fast forward on
2	SM_RESET	soft reset	0 1	no reset reset
3	SM_MP12	decode mp1&2	0 1	only decode mp3 decode mp1, mp2, mp3
4	SM_PDOWN	powerdown	0 1	power on powerdown
5	SM_DAC	dac mode	0 1	normal mpeg decoder decode 16-bit linear
6	SM_DACMONO	dac mode mono	0 1	stereo dac mode mono dac mode
7	SM_BASS	bass/treble enhancer	0 1	off on
8	SM_DACT	DCLK active edge	0 1	rising falling
9	SM_BYTEORD	Byte order on serial input bus	0 1	MSB first MSB last
10	SM_IBMODE	input bus mode	0 1	slave master
11	SM_IBCLK	input bus clk when VS1001k is master	0 1	512 kHz 1024 kHz

When SM_DIFF is set, the player inverts the left output. For a stereo input, this creates a virtual surround, and for a mono input this effectively creates a differential left/right signal.

By setting SM_FFWD the player starts to accept SCI data at a high speed, and just decodes the audio headers silently without playing any data. This can be used to fast-forward data with safe landing. Register DECODE_TIME is updated during a fast-forward just as normal.

By setting SM_RESET to 1, the player is reset.

If SM_MP12 is not set, the decoder only accepts MP3 bitstreams, and refuses to acknowledge valid MP1 and MP2 headers. This is selected as the default operation mode, because currently there are many MP3 encoders that create erraneous file headers (proper MP3 files don't have *any* extra headers) that look as valid MP1 or MP2 headers. Thus, if the user isn't sure he needs the capability to decode MP1 and MP2, this prevents accidental erraneous decoding.

Bit SM_PDOWN overrides any other: it turns VS1001k into powerdown mode, where the only operational part is the control bus.

If SM_DAC is set at the same time as SM_RESET, the player starts in DAC mode. In this mode, the user doesn't provide MP3 data through the SDI bus, but instead he provides linear signed 16-bit data through the SPI bus. For details, look at Chapter 8.7.

If SM_DACMONO is set at the same time as SM_DAC, the user doesn't have to provide samples to HDAT0. Instead, contents of HDAT1 are copied to both the left and right channels. For details, look at Chapter 8.7.

Bit SM_BASS turns on the built-in Bass and Treble enhancer. The frequency response of the enhancer when the sample rate is 44.1 kHz is shown in Figure 11. For other sample frequencies the response frequence axis must be adjusted accordingly. Example: If the sample rate is 48 kHz, the 1 kHz frequency in the figure is actually 1 kHz × 48 kHz / 44.1 kHz = 1.09 kHz. For details of how much extra processing power is needed when activating this feature, see Chapter 9.3.

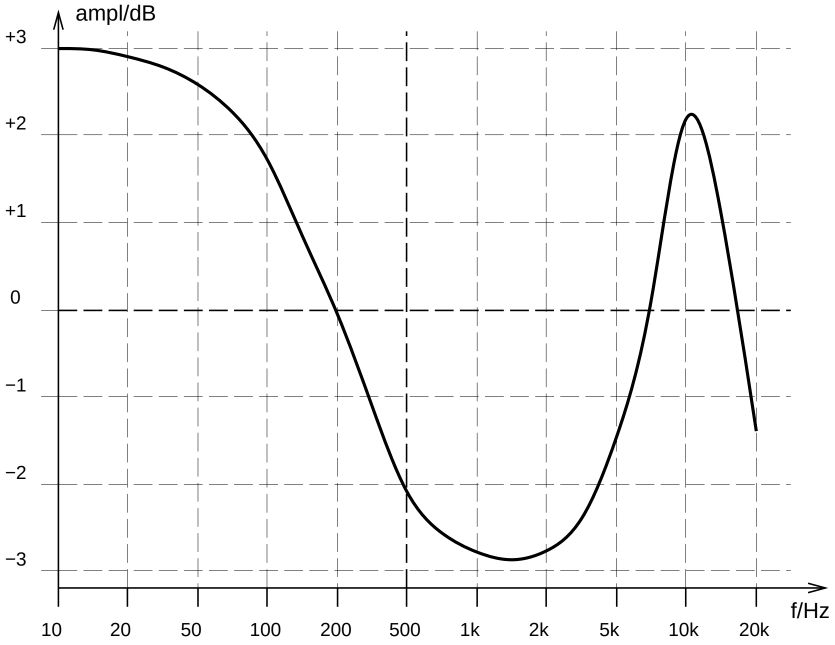


Figure 11: Built-In Bass/Treble Enhancer Frequency Response at 44.1 kHz.

SM_DACT defines the active edge of data clock for SDI.

SM_BYTEORD defines the data order inside a byte for SDI. Bytes are, however, still sent in the default order.

SM_IBMODE sets input bus to master mode. Master mode has not been tested, and its use is not recommended.

SM_IBCLK sets the bus clock speed when VS1001k is the master.

7.5.2 STATUS (RW)

STATUS contains information on the current status of the VS1001k. Bits 1 and 0 are used to control analog output volume: 0 = -0 dB, 1 = -6 dB, 3 = -12 dB. Bit 2 is analog powerdown bit. When set to 1, analog is put to powerdown.

7.5.3 INT_FCNTLH (-)

INT_FCTLH is not a user-accessible register.

7.5.4 CLOCKF (RW)

CLOCKF is used to tell if the input clock XTALI is running at something else than 24.576 MHz. XTALI is set in 2 kHz steps. Thus, the formula for calculating the correct value for this register is $XTALI/2000$ (XTALI is in Hz). Values may be between 0..32767, although hardware limits the highest allowed speed. Also, with lower-than 24.576 MHz speeds all sample rates and bit-stream widths are no longer available.

Setting the MSB of CLOCKF to 1 activates internal clock-doubling. A clock of upto 15 MHz may be doubled depending on the voltage provided to the chip.

Note: CLOCKF must be set before beginning decoding MP3 data; otherwise the sample rate will not be set correctly.

Example 1: For a 26 MHz clock the value would be $26000000/2000 = 13000$.

Example 2: For a 13 MHz external clock and using internal clock-doubling for a 26 MHz internal frequency, the value would be $0x8000 + (13000000/2000) = 39268$.

Example 3: For a 24.576 MHz clock the value would be either $24576000/2000 = 12288$, or just the default value 0. For this clock frequency, CLOCKF doesn't need to be set at all.

7.5.5 DECODE_TIME (R)

When decoding correct data, current decoded time is shown in this register in full seconds.

7.5.6 AUDATA (R)

When decoding correct data, the current bitrate in kbits/s can be found in bits 8..0 of AUDATA. For a variable bitrate bitstream, the current bitstream width is displayed. Bits 12..9 contains an index to the sample rate. The indices are shown in the table below. Bits 14..13 are not in use and always set to 0. Bit 15 is 0 for mono data and 1 for stereo.

Bits 12..9	Sample Rate/Hz
0b0000	Unknown
0b0001	44100
0b0010	48000
0b0011	32000
0b0100	22050
0b0101	24000
0b0110	16000
0b0111	11025
0b1000	12000
0b1001	8000

7.5.7 WRAM (W)

WRAM is used to upload application programs to program RAM. The start address must be initialized by writing to the WRAMADDR register prior to the first call of WRAM. value will be used. As 16 bits of data can be transferred with one WRAM write, and the program word is 32 bits, two consecutive writes are needed for each program word. The byte order is big-endian (i.e. MSBs first). After each full-word write, the internal pointer is autoincremented.

7.5.8 WRAMADDR (W)

WRAMADDR is used to set the program address for following WRAM writes. User program space is between addresses 0x4000 .. 0x43ff (with addresses 0x4000 .. 0x401f being reserved by the system), but for writes through the WRAM mechanism, they are visible at addresses 0x4000 higher. Thus, if the programmer wish to write his application to address 0x4167, he should write 0x4167 + 0x4000 = 0x8167 to WRAMADDR.

7.5.9 HDAT0 and HDAT1 (R)

Bit	Function	Value	Explanation
HDAT1[15:5]	syncword	2047	stream valid
HDAT1[4:3]	ID	3	ISO 11172-3 1.0
		2	MPG 2.0 (1/2-rate)
		1	MPG 2.5 (1/4-rate)
		0	MPG 2.5 (1/4-rate)
HDAT1[2:1]	layer	3	I
		2	II
		1	III
		0	reserved
HDAT1[0]	protect bit	1	CRC protected
		0	No CRC
HDAT0[15:12]	bitrate		c.f. 11172-3
HDAT0[11:10]	sample rate	3	reserved
		2	32/16/8 kHz
		1	48/24/12 kHz
		0	44/22/11 kHz
HDAT0[9]	pad bit	1	additional slot
		0	normal frame
HDAT0[8]	private bit		not defined
HDAT0[7:6]	mode	3	mono
		2	dual channel
		1	joint stereo
		0	stereo
HDAT0[5:4]	extension		c.f. 11172-3
HDAT0[3]	copyright	1	copyrighted
		0	free
HDAT0[2]	original	1	original
		0	copy
HDAT0[1:0]	emphasis	3	CCITT J.17
		2	reserved
		1	50/15 microsec
		0	none

When read, HDAT0 and HDAT1 contain header information that is extracted from MPEG stream being currently being decoded. Right after resetting VS1001k, 0 is automatically written to both registers, indicating no data has been found yet.

The “sample rate” field in HDAT0 is interpreted as follows: if the “ID” field in HDAT1 is ’1’, the highest sample rate is used. If “ID” is ’0’, half sample rate is used. For ’2’ and ’3’, the lowest sample rate is used.

Note: The sample rate, stereo/mono and bitrate information can more easily be read from register AUDATA.

Note: HDAT0 and HDAT1 may not be read when VS1001k is in DAC mode.

7.5.10 A1ADDR (RW)

A1ADDR indicates the start address of the application code written earlier through WRAMADDR and WRAM registers. If no application code is used, this register should not be initialized, or it should be initialized to zero. For more details, see Chapter 7.6.

7.5.11 VOL (RW)

VOL is a volume control for the player hardware. For each channel, a value in the range of 0 .. 255 may be defined to set its attenuation from the maximum volume level (in 0.5 dB steps). The left channel value is then multiplied by 256 and the values are added. Thus, maximum volume is 0 and total silence if 65535. Example: for a volume of -2.0 dB for the left channel and -3.5 dB for the right channel: (4*256) + 7 = 1031. Note, that at startup volume is set to full volume. Resetting the software does not reset the volume setting.

Note: Setting the volume to total silence (255 for both left and right channels), will turn analog power off. This will save power, but also cause a slight snap in the earphones. If you want to turn the volume off but don't want this snap, turn the volume only to 254 for both channels (0xFEFE).

7.5.12 RESERVED (RW)

This register has been reserved for future use.

7.5.13 A1CTRL[x] (RW)

A1CTRL[x] -registers (x=[0 .. 1]) can be used to access the user's application program.

7.6 Application Programs

There is 1 kWord (32-bit) of RAM memory for user code on the VS1001k chip. This can be used for example to provide features like:

- Bass, treble (etc.) controls
- Channel mixing (stereo into mono)
- Digital equalizer

The loading of application programs is initiated by writing a base address to WRAMADDR register. The program is then loaded by writing data to the WRAM register.

Any programs have to be reloaded every time the chip loses its power supply. The application address does not have to be set every time after the system software has been reset.

The C prototype for the application function is as follows:

void (*applAddr) (s_int16 __i0 *data, s_int16 __a0 chan, s_int16 __a1 nSampl);

If $chan$ is either 1 or 2, the user may do in-place filtering of $data$. $chan$ is the number of channels, and $nSampl$ is the total number of samples to be handled. With a stereo signal, the left and right channel samples are interleaved. The program may trust in that $nSampl$ is always divisible by 4.

If $chan$ is 3, the volume has been set using SPI volume setting. In this case the volume is set automatically, but software is still provided with some data of the new volume setting: $nSampl$ contains the user selected volume setting, and d points to the left volume multiplier register. The right volume multiplier is in $d+1$. The volume value is unsigned, and 32768 corresponds to a gain of 1.0 (e.g. 16384 $->$ 0.5).

If $chan$ is 4..5, A1CTRL[$chan$-4] has been called, and the user given value is in $nSampl$.

Volume control is placed after any user applications. Thus it is generally a better idea to only write filters that attenuate some frequencies and don't emphasize any. To compensate for the lower volume, main volume setting may be set up a few dB whenever tone control is activated.

An example of how to write an application that employs all the features mentioned here, may be found in separate application notes, available on VLSI Solution's Web pages.

7.7 Stereo Audio DAC

The decoded digital data is transformed into analog format by an 18-bit oversampling multi-bit sigma-delta DA-converter. The oversampled output is low-pass filtered by an on-chip analog filter. The output rate of the DA-converter is always 1/4 of the clock rate, or 128 times the highest usable sample rate. For instance for a 24.576 MHz clock, the DA-converter operates at 128x48 kHz, which is 6.144 MHz. If the input sample rate is other than 48 kHz, it is internally converted to 48 kHz by the DAC. This removes the need for complex PLL-based clocking schemes and still allows the use of several sample rates with one fixed master clock frequency.

The outputs can be separately muted by the user. If the output of the decoder is invalid or input data is not received fast enough, analog outputs are automatically muted. The analog outputs have buffers that are capable of driving 30Ω loads with a maximum of 50nF capacitance.

8 Operation

8.1 Clocking

The VS1001k chip operates typically on a single 24.576 MHz master clock frequency. This clock can be generated by an external circuitry (connected to pin XTALI) or by the internal clock chrystal interface (pins XTALI and XTALO). This clock is sufficient to support a high quality audio output for all the standard sample rates.

8.2 Powerdown

In powerdown mode the chip only monitors the control bus. The analog output drivers are turned off and the processor remains in hold-state.

8.3 Hardware Reset

When the XRESET -signal is driven low, VS1001k is reset and all the control registers and internal states are set to the initial values. XRESET-signal is asynchronous to any external clock. The reset mode doubles as a full-powerdown mode, where both digital and analog parts of VS1001k are in minimum power consumption stage, and where clocks are stopped. Also XTALO and XTALI are grounded.

After a hardware reset (or at power-up), set the basic software registers such as VOL for volume (and CLOCKF if the input clock is anything else than 24.576 MHz) before starting decoding.

8.4 Software Reset

Between any two MP3 files, the decoder software has to be reset. This is done by activating bit 2 in SCI's MODE register (c.f. Chapter 7.5.1). Then wait for at least 2 μs, then look at DREQ. DREQ will stay down for at least 6000 clock cycles, which means an approximate 250 μs delay if VS1001k is run at 24.576 MHz. When DREQ goes up, write at least one zero to SDI. After this, you may continue playback as usual.

If you want to make sure VS1001k doesn't cut the ending of low-bitrate data streams, it is recommended to feed 2048 zeros to the SDI bus before activating the reset bit (DREQ must be respected just as with normal SDI data). This will make sure all frames have been decoded before resetting the chip.

8.5 Play/Decode

This is the normal operation mode of VS1001k. The SDI data is decoded. Decoded samples are converted to analog domain by the internal DAC, If there are errors in the decoding process, the error flags of SCI's

HDAT0 and HDAT1 are set accordingly. In case there are serious errors in the input data, decoding is still continued, but the analog outputs are muted.

When there is no valid input for decoding, VS1001k goes into idle mode (lower power consumption than during decoding) and actively monitors the serial data input for valid data. The data input does not need to be clocked (DCLK) when no data is sent.

The software needs to be reset between MPEG audio stream files. See for the Chapter “Testing” to see how it is done.

8.6 Sanity Checks

Although VS1001k checks extensively for bad MP3 streams, it may happen that it encounters a bitstream that makes the firmware’s recovery code fail. This may particularly happen during fast forward and fast backwards operations, where the data where the microcontroller lands the MP3 decoder may not be a valid header.

The microcontroller should keep a look at the data speeds VS1001k requires. If data input either stops completely (DREQ always inactive) for a whole second, or if VS1001k requires more than 60 KB data in any single second, it is the responsibility of the microcontroller to either reset the software. If that doesn’t help, a hardware reset should be issued.

8.7 DAC Mode

VS1001k can be used as a DAC. In this case, instead of decoding an MPEG bitstream from SDI input, it decodes linear 16-bit signed two’s complement stereo or mono audio samples provided through the SCI bus.

To use the DAC mode, the following steps are to be taken:

- VS1001k should first be let to start in a conventional way (deassert hardware reset).
- If required, MP3 data may be played. If MP3 data is played, the user should wait for at least 1 second after stopping playback.
- The sample rate (in Hz) that is to be used for the DAC mode is written to SCI register DECODE_TIME.
- SPI register MODE bits 5 (dac mode) and 2 (soft reset) are set to 1. If the input is to be mono input, also bit 6 (dac mode mono) is set to 1.
- After the system is reset, data may be fed through SCI as described below.
- To exit DAC mode, set SPI register MODE bit 5 (dac mode) to 0, and set bit 2 (soft reset) to 1.

In DAC mode, data is hed through SPI registers HDAT0 (left channel) and HDAT1 (right channel). After each write to HDAT1, the player copies the current contents of both HDAT0 and HDAT1 to its audio buffer, wherefrom the samples are played.

If the player is in DAC Mono Mode, the audio samples are provided to HDAT1. In this mode HDAT0 is not used at all.

When in DAC mode, there may be no activity in the SDI bus.

8.8 Testing

There are several test modes in VS1001k, which allow the user to perform memory tests, SPI bus tests, and several different sine wave tests ranging from 250 Hz to 1500 Hz.

All tests are started in a similar way: if some MP3 has been decoded, the decoding has to be finished by sending 1024 zero bytes. This ensures that the decoder is looking for the next header and will not translate the testing commands to be valid MP3 data. If no MP3 data has been decoded since the last reset, this step doesn't have to be taken.

Each test is started by sending a 4-byte special command sequence. The sequences are described below. Note, that after each special command, at least 4 zeros must be transmitted for the command to take effect.

8.8.1 Memory Test

Memory test mode is initialized by the sequence 0x4D 0xEA 0x6D 0x54. After this command (and its required 4 zeros), wait for 500000 clock cycles. The result can be read from the SPI register HDAT0, and 'one' bits are interpreted as follows:

Bit(s)	**Meaning**
0	Good X ROM
1	Good Y ROM (high)
2	Good Y ROM (low)
3	Good Y RAM
4	Good X RAM
5	Good Instruction RAM (high)
6	Good Instruction RAM (low)
7	Unused

All tests are non-destructive and interrupts are disabled during testing. Thus, no user software or data is harmed by the tests.

Instruction ROM cannot be tested with software.

8.8.2 SCI Test

Sci test is initialized by the sequence: 0x53 0x70 0xEE n, where $n - 48$ is the register number to test. The content of the given register is read and copied to HDAT0. If the register to be tested is HDAT0, the result is copied to HDAT1.

8.8.3 Sine Test

Sine test is initialized by the sequence: 0x53 0xEF 0x6E n, where n (48..119) defines the sine test to use. If we define $FsIdx = (n - 48) mod 9$ and $FSin = (n - 48)/9$, the following tables may be used:

FsIdx	Fs
0	44100 Hz
1	48000 Hz
2	32000 Hz
3	22050 Hz
4	24000 Hz
5	16000 Hz
6	11025 Hz
7	12000 Hz
8	8000 Hz

FSin	Length of Sin
0	32.000 samples
1	16.000 samples
2	10.667 samples
3	8.000 samples
4	6.400 samples
5	5.333 samples
6	4.571 samples
7	4.000 samples

Example: Sine test is called with a test value of 62. 62-48 = 14, FsIdx = 5 and FSin = 1. From the tables we get the sample rate 16000 Hz, and the sine wave length, which is 16 samples. Thus, we'll get a 1 kHz voice.

To exit the sine test, send the sequence 0x45 0x78 0x69 0x74.

Note: The sine test signals go through the digital volume control, so it is possible to test channels separately.

9 Clock Speeds

9.1 General

Using another clock speed than the default 24.576 MHz changes the behaviour of VS1001k in many ways. The effects of different clock rates are discussed in this chapter.

9.2 Maximum Sample Rate

With a 24.576 MHz or higher clock, the maximum sample rate is 48000 Hz. For a lower clock, $48000 * ClockInMHz/24.576$ rounded downwards to the next allowed MP3 sample rate tells the maximum available sample rate. Thus, with a 15 MHz clock it is possible to decode playback at 29297 $- >$ 24000 Hz.

To put it another way: if VS1001k is run with a 26 MHz clock, the maximum playback frequency would be 50781 Hz, and all MPEG audio files can be played correctly. However, with a 24 MHz clock the maximum playback rate is only 46875 Hz. In this case, all sample rates upto 44100 Hz can be played without problems, but 48000 Hz is played at 46875 Hz, or in other words with a 2.5% speed error.

9.3 Maximum Amount of DSP Effects

With a 24.576 MHz clock there is some 7 MHz free processor time (minimum) for a 128 kbits/s 44.1 kHz bitstream. However, with 256 kbits/s at 48 kHz there is no constant free time: although the average load is at approx. 22.5 MHz, peaks require somewhat more processing power (see Chapter 9.4).

If it is desired to be able to play higher-width bitstreams at the same time with special effects or tone controls, a higher clock rate is required. Notice, that the internal tone controller that can be activated with the SM_BASS bit of SPI register MODE, requires 1.59 MHz additional processing power.

If a user wants to have a lower clock but still have DSP special effects, it may be a good idea to research the bitstream through the AUDATA register. If the bitstream width and sample rates seem to be too high, turn DSP effects off using the A1ADDR register; otherwise turn them on.

9.4 Maximum Decodable Bitstream

The maximum bitstream that can be correctly decoded depends on the internal clock frequency CLKI. The following subsections describe what kinds of sample rates and bit rates can be decoded with different clock rates.

If the user tries to play back a stream that has a too high sample rate, it will be played with the highest sample rate available as described in Chapter 9.2. if a user tries to play a 32 kHz MP3 file with a 12.288 MHz clock, it will be played back at 24 kHz, i.e. 2/3 of the correct speed.

If the user tries to play back a stream that has too high a bitrate, it will be played back too slowly and with some gaps in the sound. If the difference is very small, the gaps may be unnoticeable. Example: if one tries to play back a constant bitrate file with 256 kbits/s with a 24.576 MHz clock, the file will at some points be played back a bit slowly. However, in many cases, this cannot be noticed, because decode problems occur usually at transients where the ear doesn't very easy hear small glitches.

9.4.1 CLKI = 12.288 MHz

Sample rates upto 24 kHz may be decoded from bitstreams upto 96 kbits/s.

9.4.2 CLKI = 22.580 MHz

Sample rates upto 44.1 kHz may be decoded from bitstreams upto 160 kbits/s. Normally, variable bitrate files with upto 256 kbits/s peaks may be played without glitches.

9.4.3 CLKI = 24.576 MHz

From this frequency on, all sample rates (i.e. upto 48 kHz) may be decoded from bitstreams upto 192 kbits/s. 256 kbits/s bitstreams usually have so little glitches that they cannot be heard. Normally, variable bitrate files with upto 256 or even 320 kbits/s peaks may be played without any glitches.

9.4.4 CLKI = 26.000 MHz

Bitstreams upto 256 kbits/s can be decoded. Normally, variable bitrate files with upto 320 kbits/s peaks may be played without any glitches.

9.4.5 CLKI = 28.000 MHz

Bitstreams upto 320 kbits/s can be decoded, with both constant and variable bitrates.

10 Writing Software

10.1 When to Write Software

User software is required when a user wishes to add some own functionality like DSP effects or tone controls to VS1001k. Some tone controls are available from VLSI Solution, but if a user wishes to go further than that or use VS1001k in some unexpected way, this is how to do it.

However, most of the users of VS1001k don't need to worry about writing their own code, or this chapter.

10.2 The Processor Core

VS_DSP is a 16/32-bit DSP processor core that can very well also be used as an all-purpose processor. The VLSI Solution's free VSKIT Software Package contains all the tools and documentation needed to write, simulate and debug Assembly Language or Extended ANSI C programs for the VS_DSP processor core.

The VSKIT Software Package is available on request from VLSI Solution.

10.3 User's Memory Map

10.4 Hardware Registers

All hardware registers are located in X memory.

10.4.1 SCI Registers, 0x4000

All SCI registers described in Chapter 7.5 can be found here between 0x4000..0x40FF.

10.4.2 Serial Registers, 0x4100

SER_DATA (0x4100) contains the last data value read from the data bus. The LSB of SER_DREQ (0x4101) defines the status of the DREQ signal.

10.4.3 DAC Registers, 0x4200

DAC data should be written at each audio interrupt to DAC_LEFT (0x4200) and DAC_RIGHT (0x4201) as signed values. INT_FCTLL (0x4202) is not a user-serviceable register.

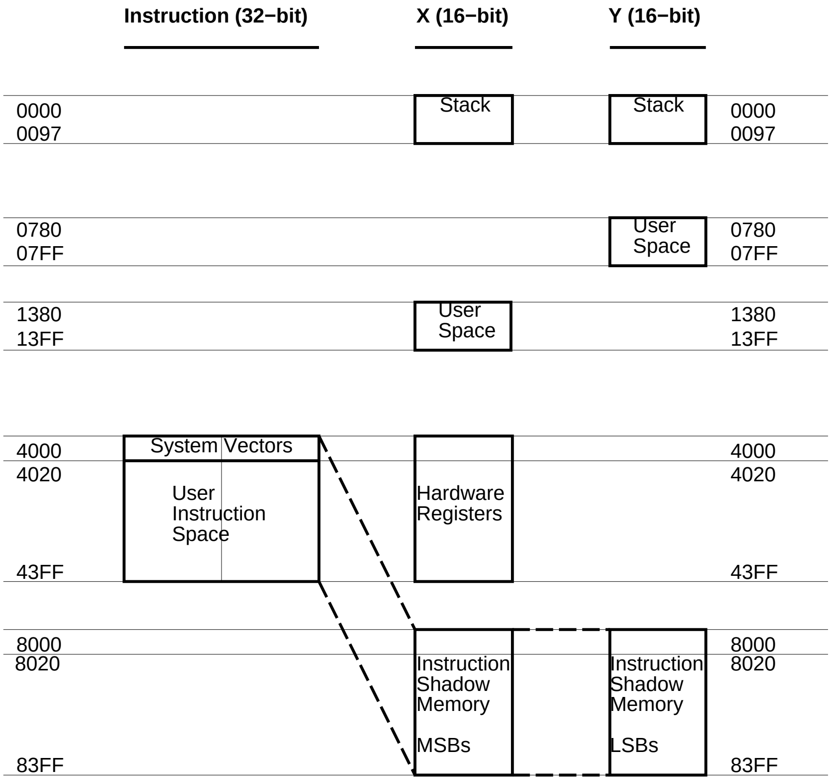


Figure 12: User's Memory Map.

10.4.4 Interrupt Registers, 0x4300

INT_ENABLE (0x4300) controls the interrupts. Bit 0 switches the DAC interrupt on (1) and off (0), bit 1 controls the SPI interrupt, and bit 2 controls the DATA interrupt. It may take upto 6 clock cycles before changing this register has any effect.

By writing any value to INT_GLOB_DIS (0x4301) adds one to the interrupt counter and effectively disables all interrupts. It may take upto 6 clock cycles before writing this register has any effect.

Writing any value to INT_GLOB_ENA (0x4302) subtracts one from the interrupt counter. If the interrupt counter becomes zero, interrupts selected with INT_ENABLE are restored. An interrupt routine should always write to this register as the last thing it does, because interrupts automatically add one to the interrupt counter, but subtracting it back to its initial value is on the responsibility of the user. It may take upto 6 clock cycles before writing this register has any effect.

By reading INT_COUNTER (0x4303) the user may check if the interrupt counter is correct or not. If the register is not 0, interrupts are disabled. This register may not be written to.

10.5 System Vector Tags

The System Vector Tags are tags that may be replaced by the user to take control over several decoder functions.

10.5.1 AudioInt, 0x4000..0x4001

Normally contains the following VS_DSP assembly code:

```
  j dac_int
  stx mr1,(i6)+1  ; sty i7,(i6)
```

The user may, at will, replace the first instruction with either a j or $jmpi$ command to gain control over the audio interrupt. It is not recommended to change the instruction at 0x4001.

10.5.2 SpiInt, 0x4002..0x4003

Normally contains the following VS_DSP assembly code:

```
  j spi_int
  stx mr1,(i6)+1  ; sty i7,(i6)
```

The user may, at will, replace the first address with either a j or jmpi command to gain control over the spi interrupt. It is not recommended to change the instruction at 0x4003.

10.5.3 DataInt, 0x4004..0x4005

Normally contains the following VS_DSP assembly code:

```
  j data_int
  stx mr1,(i6)+1  ; sty i7,(i6)
```

The user may, at will, replace the first address with either a j or jmpi command to gain control over the MP3 data interrupt. It is not recommended to change the instruction at 0x4005.

10.5.4 UserCodec, 0x4008..0x4009

Normally contains the following VS_DSP assembly code:

```
  jr
  nop
```

If the user wants to take control away from the standard decoder, the first instruction should be replaced with an appropriate jump command to user's own code.

Unless the user is feeding MP3 data at the same time, the system activates the user program in less than 1 ms. After this, the user should steal interrupt vectors from the system, and then insert user programs.

10.6 System Vector Functions

The System Vector Functions are pointers to some functions that the user may call to help implementing his own applications.

10.6.1 WriteIRam(), 0x4010

VS_DSP C prototype:

void WriteIRam(register __i0 u_int16 *addr, register __a1 u_int16 msW, register __a0 u_int16 lsW);

This is the only supported way to write to the User Instruction RAM. This is because Instruction RAM cannot be written when program control is in RAM. Thus, the actual implementation of this function is in ROM, and here is simply a tag to that routine.

Note: Instruction RAM is shadowed 0x4000 addresses higher in the X and Y RAMs. Thus, if you want to write to instruction address 0x4020, $addr$ must be 0x4020 + 0x4000 = 0x8020.

10.6.2 ReadIRam(), 0x4011

VS_DSP C prototype:

u_int32 ReadIRam(register __i0 u_int16 *addr);

This is the only supported way to read from the User Instruction RAM. This is because Instruction RAM cannot be read when program control is in RAM. Thus, the actual implementation of this function is in ROM, and here is simply a tag to that routine.

A1 contains the MSBs and a0 the LSBs of the result.

Note: Instruction RAM is shadowed 0x4000 addresses higher in the X and Y RAMs. Thus, if you want to read from instruction address 0x4020, $addr$ must be 0x4020 + 0x4000 = 0x8020.

10.6.3 DataWords(), 0x4012

VS_DSP C prototype:

u_int16 DataWords(void);

If the user has taken over the normal operation of the system by switching the pointer in UserCodec to point to his own code, he may read data from the Data Interface through this and the following two functions. This function returns the number of data words (each containing two bytes of data) that can be read. If there is not enough data available, data acquisition functions GetDataByte() and GetDataWords() may NOT be called!

10.6.4 GetDataByte(), 0x4013

VS_DSP C prototype:

u_int16 GetDataByte(void);

Reads and returns one data byte from the Data Interface.

Before calling this function, always check first that there are at least 1 word waiting with function DataWords().

10.6.5 GetDataWords(), 0x4014

VS_DSP C prototype:

void GetDataWords(register __i0 __y u_int16 *d, register __a0 u_int16 n);

Read n data byte pairs and copy them in big-endian format (first byte to MSBs) to d.

Before calling this function, always check first that there are at least 1+n words waiting with function DataWords().

11 VS1001 Version Changes

This chapter describes changes between different generations of VS1001.

Note: VS1001k is the final, production version of VS1001.

11.1 Changes Between VS1001h and Production Version VS1001k

- When the chip is reset with pin XRESET, XTALO and XTALI are driven to ground.
- Running with normal clock earlier required slightly different clock generation than for clock-doubled (see Chapters 4 and 5). This is no longer the case.
- Lots of new SPI register MODE bits: SM_DIFF, SM_FFWD, SM_MP12, SM_DAC, SM_DMONO, SM_BASS. For details, see Chapter 7.5.1.
- Default is now to only decode MP3. MP1 and MP2 decoding can be activated by setting SM_MP12 in SPI register MODE. For details, see Chapter 7.5.1.
- 20..60 mV DAC offset corrected.
- A firmware bug made it impossible to decode 320 kbits/s MP3 data. This has been corrected.
- A hardware bug made it practically impossible to load code to RAM. This has been corrected.

11.2 Changes Between VS1001g and VS1001h

- Analog voltage requirements have been lowered. Now full gain can be achieved with a 2.7 V analog input voltage, whereas 3.4 V was needed before.

11.3 Changes Between VS1001d to VS1001g

- Clock is now adjustable, in VS1001d only 24.576 MHz could be used.
- Clock doubler added.
- VS1001d played 12 and 24 kHz incorrectly at 48 kHz, this is corrected.

12 Document Version Changes

This chapter describes the most important changes to this document.

12.1 Changes Between Version 4.01 and 4.03 for VS1001k

- Added Chapter 18, Application Note: Highest SPI Speed.
- Replaced CLKF with XTALI.

12.2 Changes Between Version 4.00 and 4.01 for VS1001k

- Corrected description of how to calculate a correct address for WRAMADDR.
- Added Chapter 13, Errata.
- Minor corrections here and there, like table in Chapter 7.5.9 (HDAT1's bits 4:3 were explained incorrectly).
- Replaced and added new chip photographs to pages 1 and 58.

12.3 Changes Between Version 3.56 and 4.00 for VS1001k

- All tables in Chapter 2 are now drived from characterized, tested chips. In other words, all delivered chips have passed the characterization values given in the tables.

12.4 Changes Between Version 3.55 and 3.56 for VS1001k

- Clarified slightly CLOCKF usage.

12.5 Changes Between Version 3.53 and 3.55 for VS1001k

- Slight changes to page 1.
- Corrected "addr" field in table for A1CTRL registers in Chapter 7.5.
- Changed value for C5 and C6 in Figure 4 from 10 μF to 100 μF.

12.6 Changes Between Version 3.52 and 3.53 for VS1001k

- Added mention of FAQ to Chapter 19.

12.7 Changes Between Version 3.51 and 3.52 for VS1001k

- Added package Figure 3.

12.8 Changes Between Version 3.5 and 3.51 for VS1001k

- Corrected documentation for registers A1CTRL[x]. Old A1CTRL[0] was replaced with new RESERVED, and old A1CTRL[1..2] were replaced with new A1CTRL[0..1].

12.9 Changes Between Version 3.3 for VS1001h and Version 3.5 for VS1001k

- Changed descriptions for new registers available in VS1001k, see Chapter 11.1.
- Corrected clock doubler documentation for SPI register CLOCKF to Chapter 7.5.4.

13 Errata

This chapter describes the known problems with the current VS1001 generation.

13.1 MP2 Joint-Stereo Playback

Joint-stereo playback doesn't work properly for MPEG 1.0 or MPEG 2.0 layer 2 files. While the bitstream is decoded properly, signal synthesizing has a bug that results in muffled sound. The bug has been found and will be corrected in any future firmware versions. Mono and dual stereo bitstreams work just fine.

14 Application Note: Quick Startup

14.1 Overview

This chapter is meant to be read after the first device using VS1001k has been built. In this chapter the first steps to get VS1001k up and running are discussed. In addition to this chapter, also Chapter 15 is useful for the first steps.

These tests have been meant to be run one at a time, and the next test is to be done only after the previous one has been passed.

14.2 Seeing If Firmware Wakes Up

When VS1001k is taken out of hardware reset (xRESET is turned to 1), DREQ should turn to 0 after approx. 4096 clock cycles, if it hasn't been down all the time. After yet another 6000 clock cycles, DREQ should turn to 1. If DREQ doesn't change its mode as described, the firmware is not functioning properly.

14.3 Writing to SCI

The first test that does anything useful is to try to write to the SCI bus. A good test is to try to switch the volume setting from powersave mode to full volume. This will cause slight snapping sounds, which can be checked with the earphones.

The following algorithm can be used to test SPI this way:

- Activate xCS by setting it to 0.
- Set full volume by writing four SPI bytes: 0x2, 0xB, 0x0, 0x0
- Finish SCI cycle setting xCS to 1.
- Wait for example 500 ms.
- Activate xCS by setting it to 0.
- Set analog powerdown by writing four SPI bytes: 0x2, 0xB, 0xFF, 0xFF
- Wait for example 500 ms.
- Finish SCI cycle setting xCS to 1.
- Repeat these steps indefinitely.

14.4 Reading from SCI

Now that we know writing to SCI works, we can try reading from it. Write some recognizable pattern to the volume register, like 0xA2F5 (this is a good pattern, because it effectively tests both the MSB and LSB). Then, do as follows:

- Activate xCS by setting it to 0.
- Read the volume register as described in 15.3.3.
- Check, if the result is 0xA2F5.
- Finish SCI cycle setting xCS to 1.

14.5 Writing to SDI

Writing to the actual MP3 data SDI bus is most easily tested with the special VS1001k test header, like sine test (c.f. Chapter 8.8.3).

- Activate the sine test by writing the following eight bytes to SDI: 0x53 0xEF 0x6E 0x30, 0, 0, 0, 0. Remember to take care of BSYNC properly.
- Wait for example 500 ms.
- Deactivate the sine test by writing to SDI: 0x45 0x78 0x69 0x74.
- Wait for example 500 ms.
- Repeat these steps indefinitely.

If you can hear a half-second beep every second, you have properly entered and exited the sine test mode.

Note: Unless you atteanuate volume at least -20 dB from full volume, the sine tone will be very loud. It is thus not recommended to keep your headphones on while conducting this test.

15 Application Note: Saving I/O Pins

15.1 Overview

As can be seen from Figures 4 and 5, connecting VS1001k to a microcontroller requires 8 pins. This can, however be reduced to 7, or even 6 if VS1001k is the only SPI device connected to the microcontroller. A six pin minimal configuration is presented in Figure 13.

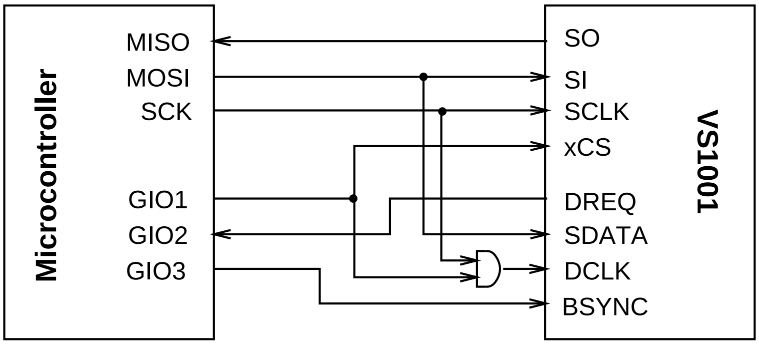


Figure 13: 6-Pin VS1001k Connection.

One pin can still be saved if BSYNC is connected permanently to VDD and it can be trusted that SDI bit sync is never lost.

15.2 Prerequirements

For this scheme to work, the following start conditions apply:

- MISO and GIO2 must be set to inputs, all other pins as outputs.
- SPI clocks have to be set so that when an SPI operation is not active, clock is set low, as in the oscilloscopes picture in Chapter 15.4.
- If there is no SPI bus available, but the microcontroller is very fast, MISO, MOSI and SCK may be implemented with general I/O pins.

15.3 Using the Connection

15.3.1 Selecting the Right Chip

In this connection, GIO1 acts as a bus select. When GIO1 is set to 1, xCS is inactive, and SCK is let through the AND gate to DCLK. Thus, now data is sent to the SDI bus. When GIO1 is set to 0, xCS activates, and DCLK stops because of the AND gate. Thus, data now goes only to the SCI bus.

If other SPI devices in addition to VS1001k is to be connected to the microcontroller, at least one more select pin is needed.

15.3.2 Sending SCI Data

Let’s say we would like to set the volume to -2 dB for the left channel and to -3.5 dB to the right channel. Thus, we would like to write the number 0x0407 to the VOL register.

- Activate the SCI interface by setting GIO1 to 0
- Write the four bytes (0x2, 0xb, 0x4, 0x7) to SCI.
- Finish the SCI transaction by setting GIO1 to 1

15.3.3 Receiving SCI Data

Let’s say we would like to read the contents of the VOL register.

- Activate the SCI interface by setting GIO1 to 0
- Write two bytes (0x3, 0xb) to SCI.
- Write one byte with any contents to SCI. Read the microcontroller’s MISO register, which now contains 8 MSBs of the result.
- Write one byte with any contents to SCI. Read the microcontroller’s MISO register, which now contains 8 LSBs of the result.
- Combine the MSBs and LSBs to form a 16-bit word.
- Finish the SCI transaction by setting GIO1 to 1

15.3.4 Sending SDI/MP3 Data

The following algorithm may be used to send a 32-byte or smaller MP3 data chunk to VS1001k:

- Wait until DREQ is 1.
- Activate DCLK by setting GIO to 1.
- For each byte, do the following:
  - Set BSYNC to 1.
  - Activize SPI transfer.
  - Wait until you know that at least the first bit of data has been sent, and the last bit has not been sent. Although VS1001k can survive several too long BSYNCs, it may be a good idea to disable interrupts to make sure the delay always is the same length.
  - Set BSYNC to 0.
  - Wait for microcontroller’s SPI cycle to finish.

15.4 SDI in Oscilloscope Pictures

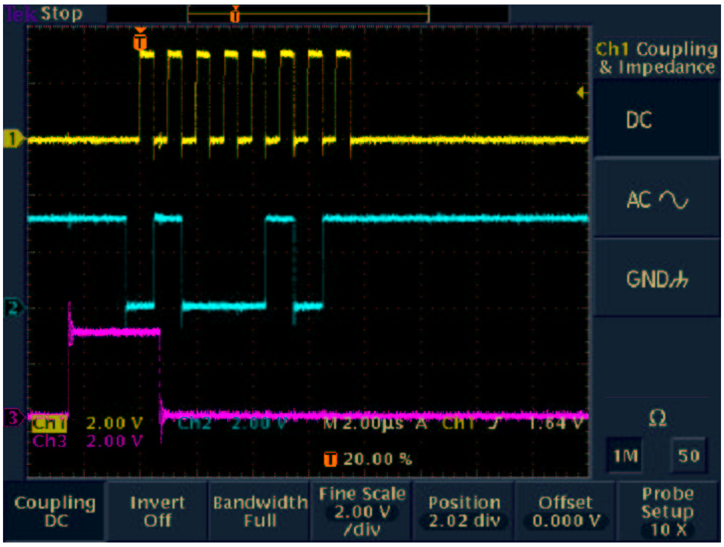


Figure 14: Oscilloscope Picture: Sending One SDI Byte.

In Figure 14 sending of one byte to SDI is presented. The three signals from top to bottom are DCLK (yellow), SDATA (cyan), and BSYNC (magenta).

BSYNC is first turned active. Then data is sent one bit at a time, and the data is read by VS1001k at each rising edge. The number to be sent is 0b01000101, or 0x45. The clock used is 1 MHz, because that is the highest this particular microcontroller can offer when running at 4 MHz.

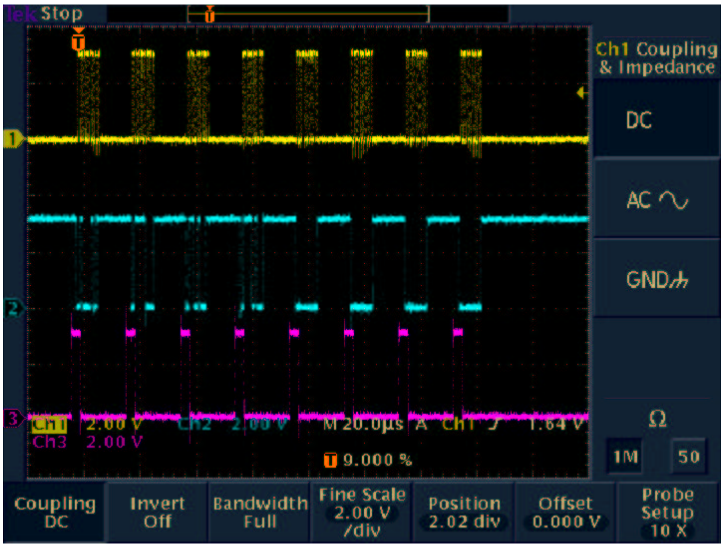


Figure 15: Oscilloscope Picture: Sending Eight SDI Bytes.

In the other example oscilloscope picture, Figure 15, sending a whole SDI test command with padding zeroes is demonstrated. Here eight bytes are transmitted, and the bytes are 0x45 0x78, 0x69, 0x74, 0x0, 0x0, 0x0, 0x0. This is the sequence to exit sine test (C.f. Chapter 8.8.3). Note, that the delay between consecutive byte writes are caused by the slowness of the microcontroller C code and are not needed by VS1001k.

16 Application Note: MP3 Player

16.1 Overview

In this chapter, an MP3 player based on VS1001k is presented. The full schematics, as well as software is presented to make the device operate with a common MultiMediaCard through its SPI mode. The system software for the microcontroller is available at *http://www.vlsi.fi/vs1001/software/*.

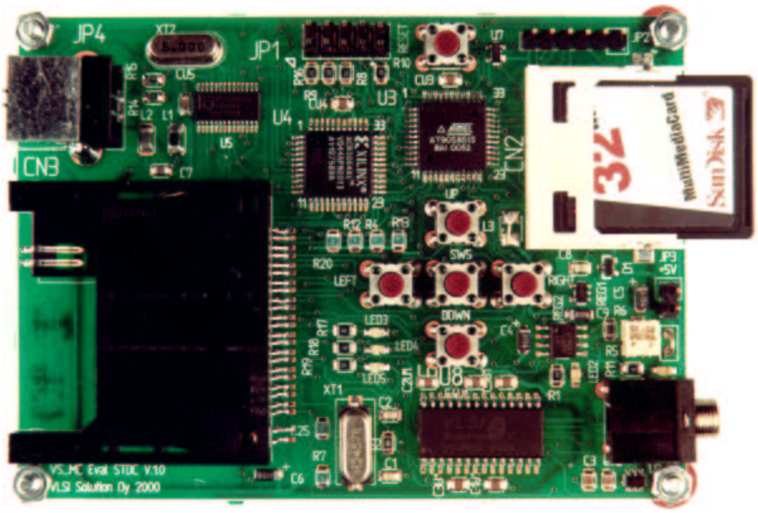


Figure 16: MP3 Evaluation Player.

16.2 Main Components

- Atmel 8-bit microcontroller AT90S8515 (8 KB program FLASH, SPI, etc)
- VS1001k
- SanDisk 32 MByte MultiMediaCard (any brand with SPI support will do)
- Xilinx XCR3064XL Programmable Logic Unit
- Philips PDIUSBD12 USB Controller
- 2 Leds for output
- 5 keys for input

The schematics also allow for on-board parallel FLASH memory as well as a Smart Media Card, but those features have not been tested.

16.3 Schematics for Standalone Unit

In Figures 17 and 18 are the full schematics of the standalone MP3 player. Note that many of the devices are not actually used, but they are there just for testing. Note also, that in page 1 both normal and clock-doubled clock connections are shown. The user needs to implement only one of these two, depending on the clock to be used. See Chapters 4 and 5 for further details.

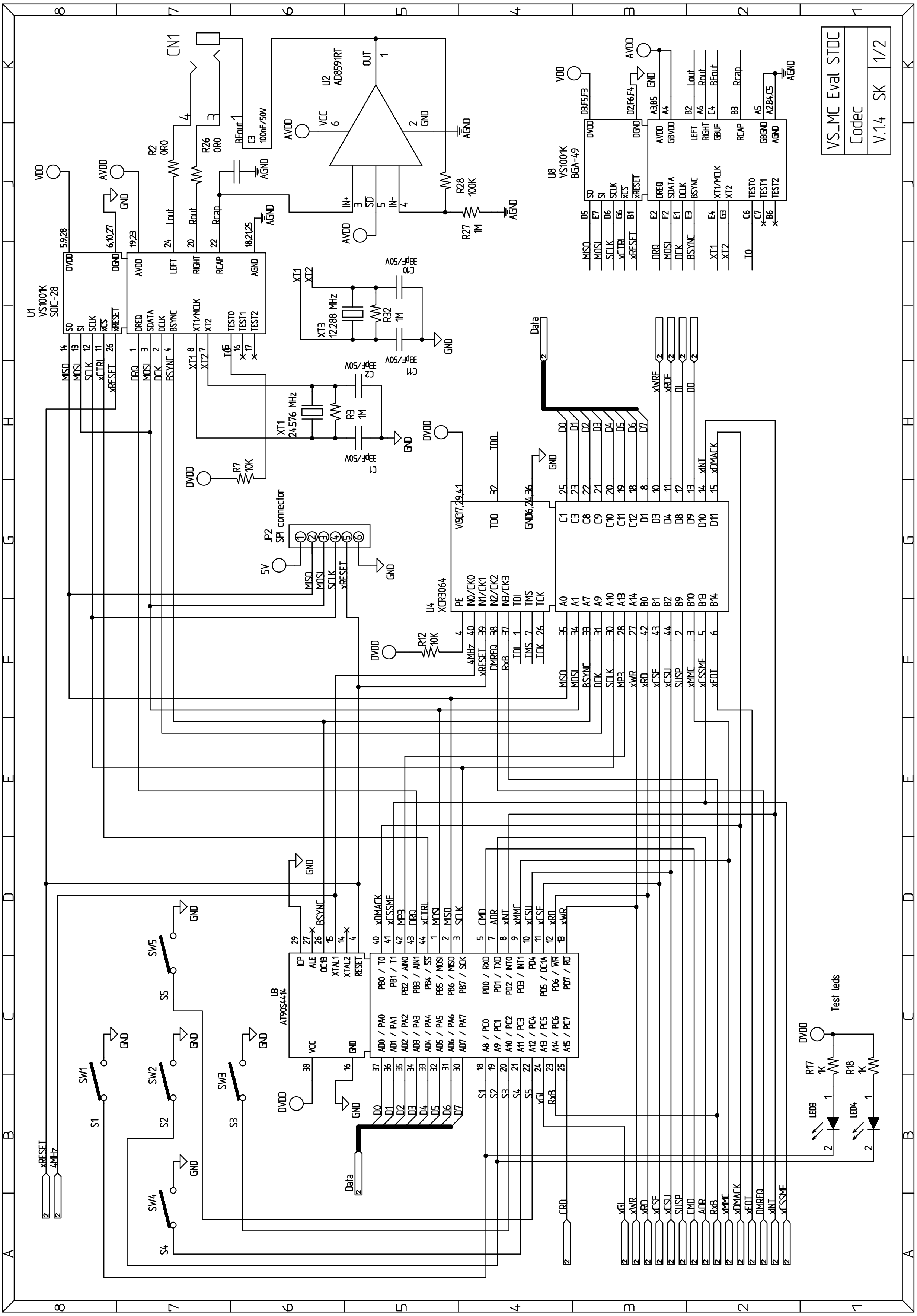


Figure 17: Schematic for Standalone MP3 Player, Page 1/2.

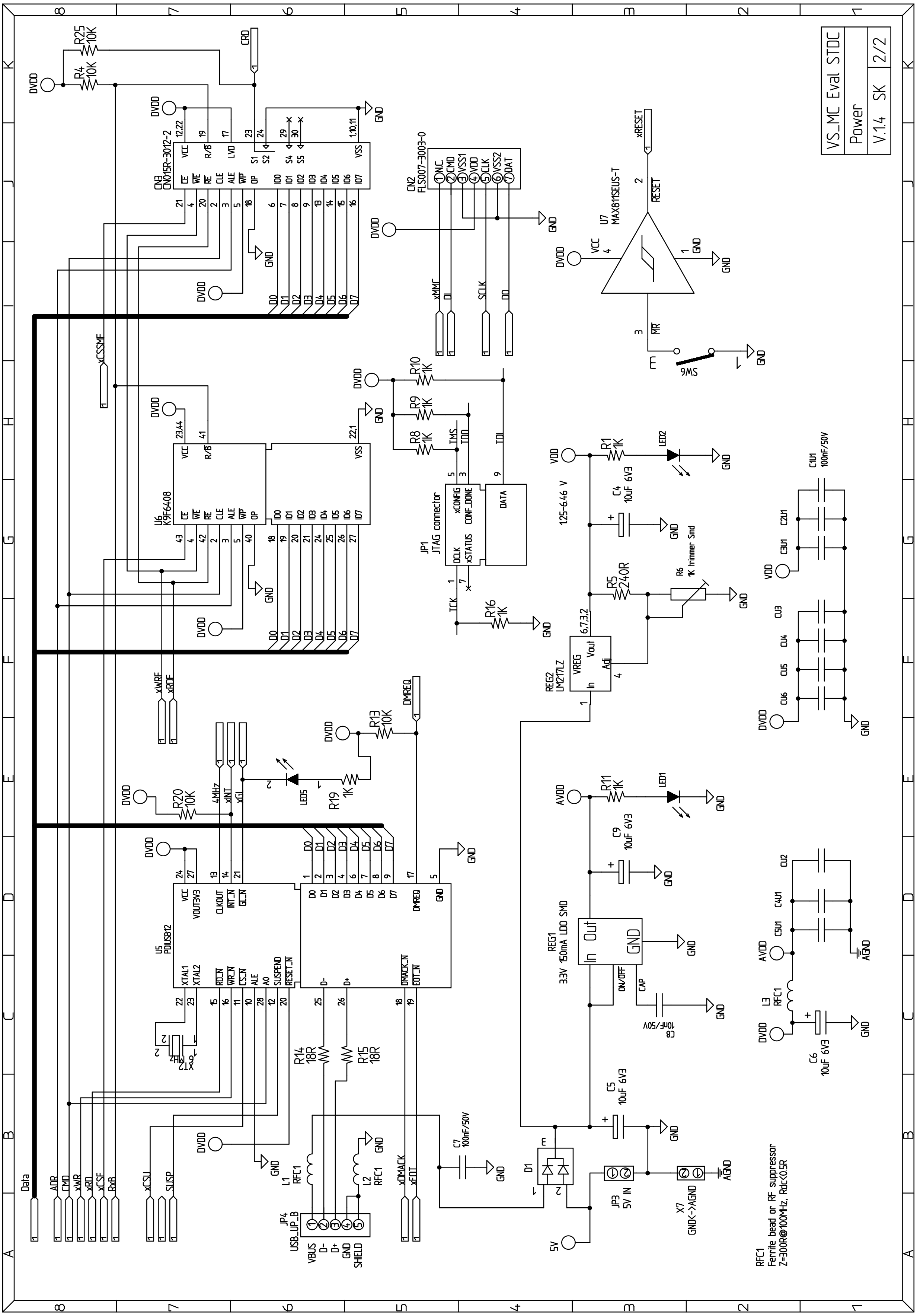


Figure 18: Schematic for Standalone MP3 Player, Page 2/2.

16.4 Xilinx Configuration

In Figure 19 the full Xilinx configuration for the standalone player is displayed. The most complex part of it is the automatic BSYNC generation. This needn't, however, be generated in such a complex way. In many cases (including this one had the system not been built in the way it was), BSYNC may be generated with an I/O pin of the microcontroller.

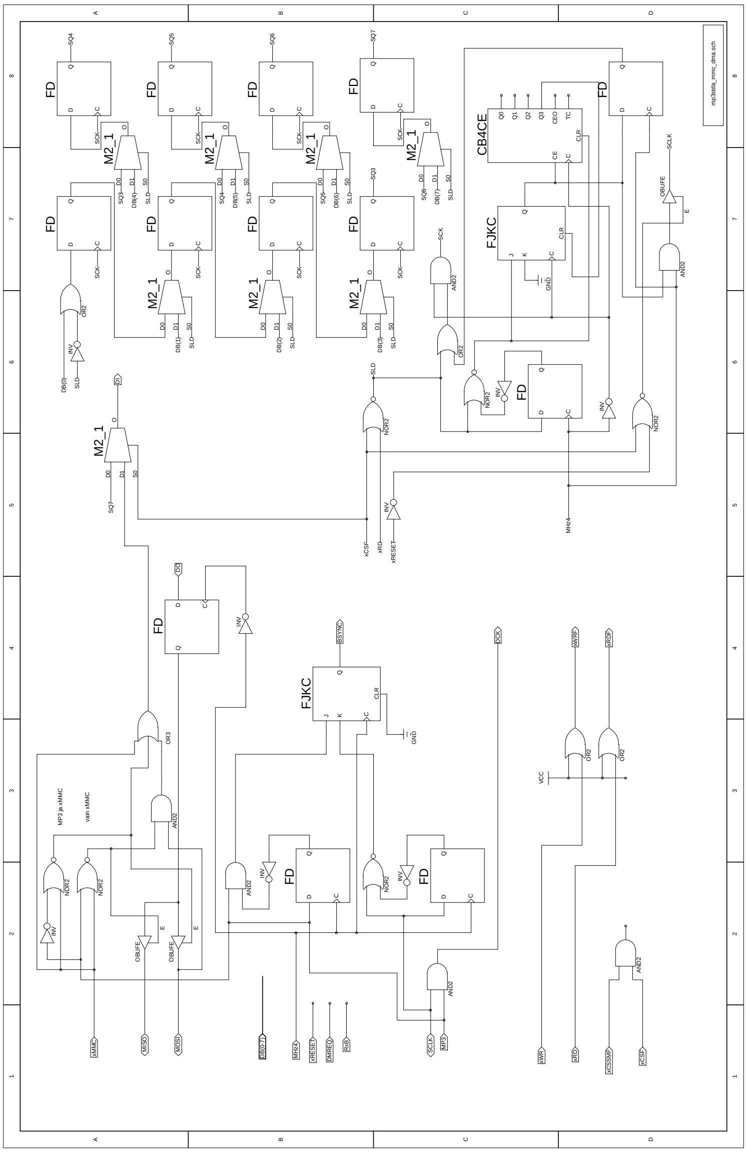


Figure 19: Xilinx Internal Configuration for Standalone MP3 Player.

17 Application Note: ESD Protection

VS1001k can withstand a maximum ESD charge of 2 kV to its analog outputs (RIGHT, LEFT, GBUF). This is a fundamental limitation of CMOS technology and cannot be helped without external protection circuitry. External protection is needed if VS1001k's analog outputs are to be connected to external connectors that may give stronger-than-normal electrical shocks. An example would be earphone plugs, where 15 kV ESD resistance may be required.

The following images show how to protect VS1001k against stronger electrical shocks.

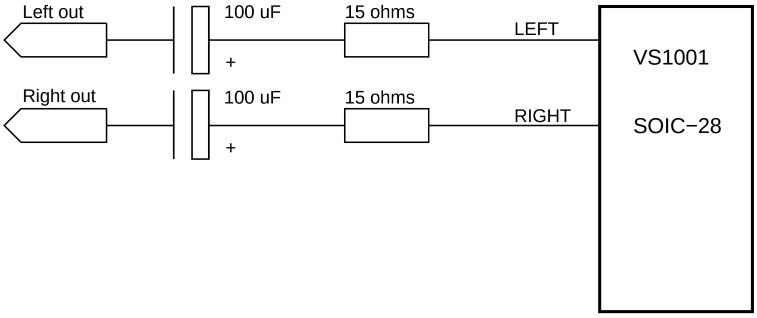


Figure 20: ESD Protection for SOIC-28 Package.

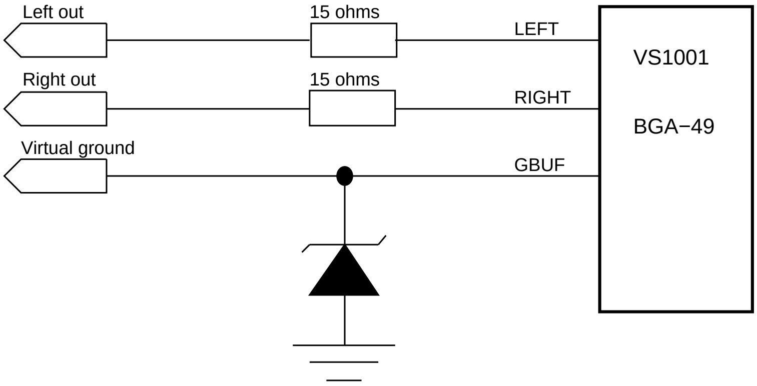


Figure 21: ESD Protection for BGA-49 Package.

Note that 15Ω is a minimum resistor value. If it is known that a product will be used with high-impedance earphones, it is recommended to use resistors that have a resistance of at least 1/5 of the earphone impedance. Example: for 300Ω earphones, at least 60Ω serial resistors are recommended.

For the BGA-49 version, use a fast response time, unidirectional transient suppression diode. Two ST/SMD example components are presented below.

Type	Standoff.volt.	Breakdown.volt.	Clamp.volt.	Peak power
SMLVT3V3	3.3V	4.1V	7.3V	600W
SMAJ5.0CA	5.0V	6.4V	13.4V	400W

18 Application Note: Highest SPI Speed

If VS1001K is used without a clock doubler, maximum SPI speed is a very simple issue: it is 1/4 of the input clock frequcny XTALI, which is also the internal clock speed CLKI. If, for instance, the nominal 24.576 MHz clock is used, the maximum SPI speed is 24.576 MHz / 4, which is slightly over 6 MHz.

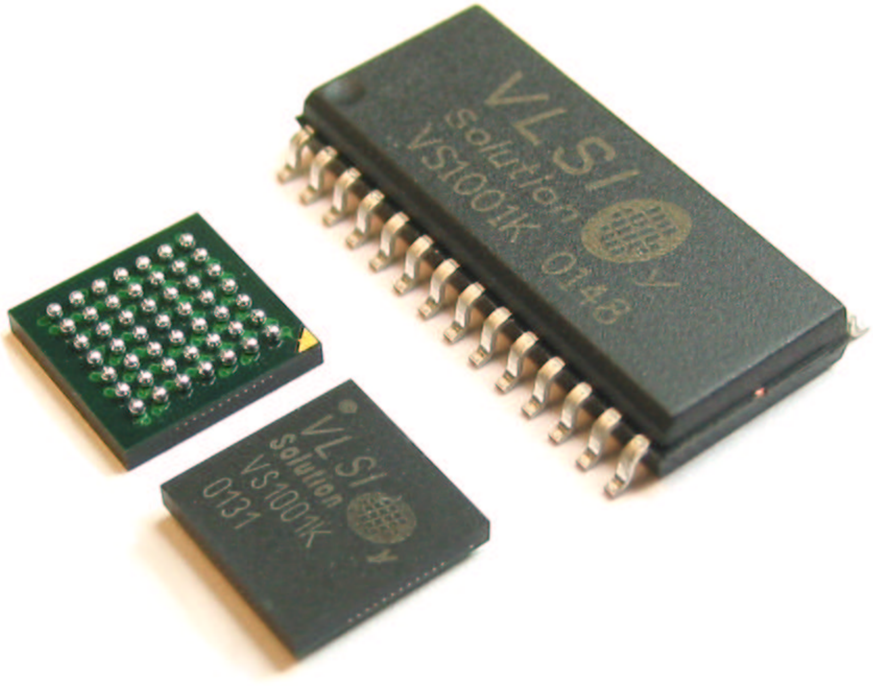
However, if a low-frequency external clock is used, and the clock-doubler is activated, there are some issues of the current firmware to be known. Unfortunately, the new clock setting won't be activated before the first frame of MP3 data has been decoded and VS1001K starts playback. Thus, until the first frame is decoded, if you are using a 12.288 MHz external clock, the maximum SPI clock is 12.288 MHz / 4, which is slightly over 3 MHz.

For applications that would like to have SPI running well over 3 MHz with the clock-doubler activated, waiting for such a badly specified event as "the first MP3 block to be decoded" is clearly a nuisance. Thus, there is a need for An Official Workaround, which is below. This workaround is guaranteed to also work with future versions of VS1001 and VS1002 without breaking anything.

The following example is assuming a 12.288 MHz external clock:

1. After a software or hardware reset, first set your clock speed to 3 MHz or lower.
2. Write the correct value 0x9800 to SCI register CLOCKF (as usual).
3. Force the clock-doubler on by writing 0x8008 to SCI register INT_FCTLH. Don't touch this register after this, as it will be later set automatically to a more suitable value!
4. Now you may set your SPI clock to upto 6 MHz and proceed sending your song and control data to SDI and SCI as usual.

19 Contact Information



VLSI Solution Oy
Hermiankatu 6-8 C
FIN-33720 Tampere
FINLAND

Fax: +358-3-316 5220
Phone: +358-3-316 5230
Email: sales@vlsi.fi
URL: http://www.vlsi.fi/

Note: If you have questions, first see http://www.vlsi.fi/vs1001/faq/ .