VS1011e - MP3 AUDIO DECODER

Features

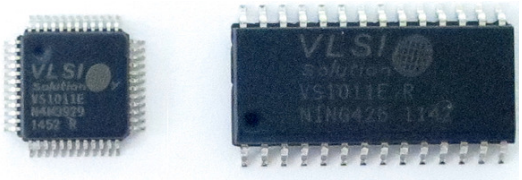
- Decodes MPEG 1.0 & 2.0 audio layer III (MP3: CBR, VBR, ABR); layers I & II (MP1, MP2) optional; WAV (PCM + IMA ADPCM)
- 320 kbit/s MP3 with 12.0 MHz external clock
- Streaming support for all audio formats
- Bass and treble controls
- Operates with single 12..13 MHz or 24..26 MHz external clock
- Internal clock doubler
- Low-power operation
- High-quality stereo DAC with no phase error between channels
- Stereo earphone driver capable of driving a 30 Ω load
- Separate 2.5..3.6 V operating voltages for analog and digital
- Serial control and data interfaces
- Can be used as a slave co-processor
- 5.5 KiB On-chip RAM for user code / data
- SPI boot for standalone applications
- New functions may be added with software and 4 GPIO pins
- Lead-free and RoHS-compliant packages LQFP-48 and SOIC-28

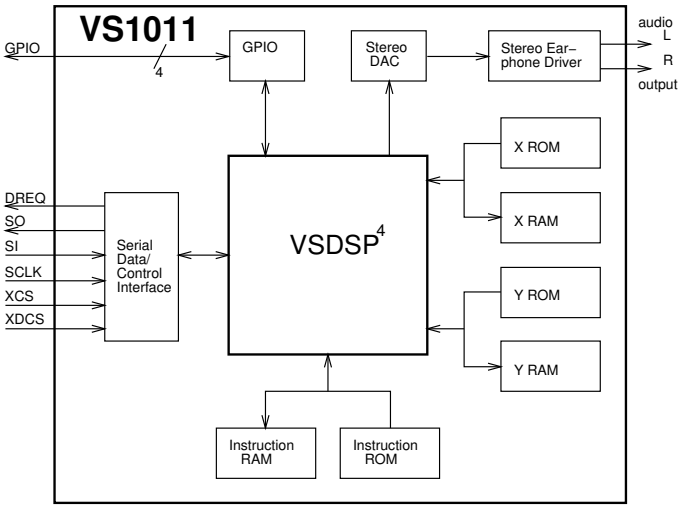
Description

VS1011e is a single-chip MPEG audio (MP3) decoder IC. It contains a high-performance, low-power DSP core VS_DSP^4, working memory, 5 KiB instruction RAM and 0.5 KiB data RAM for user applications, serial control and input data interfaces, 4 general purpose I/O pins, as well as a high-quality variable-sample-rate stereo DAC, followed by an earphone amplifier and a common buffer.

VS1011e 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 digital volume control to an 18-bit oversampling, multi-bit, sigma-delta DAC. The decoding is controlled via a serial control bus. In addition to basic decoding, it is possible to add application specific features, like DSP effects, to the user RAM memory.

VS1011e can boot directly from SPI EEPROM to run standalone applications without a separate microcontroller.





Contents

VS1011 1

Table of Contents 2

List of Figures 5

1 Licenses 6

2 Disclaimer 6

3 Definitions 6

4 Characteristics & Specifications 7
4.1 Absolute Maximum Ratings . . . 7
4.2 Recommended Operating Conditions . . . 7
4.3 Analog Characteristics . . . 8
4.4 Power Consumption . . . 8
4.5 Digital Characteristics . . . 9
4.6 Switching Characteristics - Boot Initialization . . . 9

5 Packages and Pin Descriptions 10
5.1 Packages . . . 10
5.1.1 LQFP-48 . . . 10
5.1.2 SOIC-28 . . . 12
5.2 Pin Descriptions . . . 13
5.2.1 LQFP-48 Pin Descriptions . . . 13
5.2.2 SOIC-28 Pin Descriptions . . . 14

6 Connection Diagram, LQFP-48 15

7 SPI Buses 16
7.1 General . . . 16
7.2 SPI Bus Pin Descriptions . . . 16
7.2.1 VS10xx Native Modes (New Mode, recommended) . . . 16
7.2.2 VS1001 Compatibility Mode (deprecated, do not use in new designs) 16
7.3 Serial Protocol for Serial Data Interface (SDI) . . . 17
7.3.1 General . . . 17
7.3.2 SDI in VS10xx Native Modes (New Mode, recommended) . . . 17
7.3.3 SDI in VS1001 Compatibility Mode . . . 18
7.4 Data Request Pin DREQ . . . 19
7.5 Serial Protocol for Serial Command Interface (SCI) . . . 19
7.5.1 General . . . 19
7.5.2 SCI Read . . . 20
7.5.3 SCI Write . . . 20
7.6 SPI Timing Diagram . . . 21
7.7 SPI Examples with SM_SDINEW and SM_SDISHARED set . . . 22
7.7.1 Two SCI Writes . . . 22
7.7.2 Two SDI Bytes . . . 22
7.7.3 SCI Operation in Middle of Two SDI Bytes . . . 23

8 Functional Description **24**
8.1 Main Features . . . 24
8.2 Supported Audio Codecs . . . 24
8.2.1 Supported MP1 (MPEG layer I) Formats . . . 24
8.2.2 Supported MP2 (MPEG layer II) Formats . . . 25
8.2.3 Supported MP3 (MPEG layer III) Formats . . . 25
8.2.4 Supported RIFF WAV Formats . . . 26
8.3 Data Flow of VS1011e . . . 27
8.4 Serial Data Interface (SDI) . . . 27
8.5 Serial Control Interface (SCI) . . . 28
8.6 SCI Registers . . . 28
8.6.1 SCI_MODE (RW) . . . 29
8.6.2 SCI_STATUS (RW) . . . 30
8.6.3 SCI_BASS (RW) . . . 31
8.6.4 SCI_CLOCKF (RW) . . . 31
8.6.5 SCI_DECODE_TIME (RW) . . . 32
8.6.6 SCI_AUDATA (RW) . . . 32
8.6.7 SCI_WRAM (RW) . . . 32
8.6.8 SCI_WRAMADDR (RW) . . . 32
8.6.9 SCI_HDAT0 and SCI_HDAT1 (R) . . . 33
8.6.10 SCI_AIADDR (RW) . . . 34
8.6.11 SCI_VOL (RW) . . . 34
8.6.12 SCI_AICTRL[x] (RW) . . . 34

9 Operation **35**
9.1 Clocking . . . 35
9.2 Hardware Reset . . . 35
9.3 Software Reset . . . 35
9.4 SPI Boot . . . 36
9.5 Play/Decode . . . 36
9.6 Feeding PCM data . . . 36
9.7 SDI Tests . . . 36
9.7.1 Sine Test . . . 37
9.7.2 Pin Test . . . 37
9.7.3 Memory Test . . . 37
9.7.4 SCI Test . . . 38

10 VS1011e Registers **39**
10.1 Who Needs to Read This Chapter . . . 39
10.2 The Processor Core . . . 39
10.3 VS1011e Memory Map . . . 39
10.4 SCI Registers . . . 39
10.5 Serial Data Registers . . . 40
10.6 DAC Registers . . . 41
10.7 GPIO Registers . . . 41
10.8 Interrupt Registers . . . 42
10.9 System Vector Tags . . . 43
10.9.1 AudioInt, 0x20 . . . 43
10.9.2 SciInt, 0x21 . . . 43
10.9.3 DataInt, 0x22 . . . 43
10.9.4 UserCodec, 0x0 . . . 43

10.10 System Vector Functions 44
10.10.1 WriteIRam(), 0x2 44
10.10.2 ReadIRam(), 0x4 44
10.10.3 DataBytes(), 0x6 44
10.10.4 GetDataByte(), 0x8 45
10.10.5 GetDataWords(), 0xa 45

11 VS1011 Version Changes **46**
11.1 Changes Between VS1011b and VS1011e, 2005-07-13 46
11.2 Migration Checklist from VS1011b to VS1011e, 2005-07-13 46

12 Document Version Changes **47**

13 Contact Information **49**

List of Figures

1 Pin Configuration, LQFP-48. 10
2 Package Outline, LQFP-48. 11
3 Pin Configuration, SOIC-28. 12
4 Typical Connection Diagram Using LQFP-48. 15
5 SDI Timing Diagram. 17
6 BSYNC Signal - one byte transfer. 18
7 BSYNC Signal - two byte transfer. 18
8 SCI Word Read 20
9 SCI Word Write 20
10 SPI Timing Diagram. 21
11 Two SCI Operations. 22
12 Two SDI Bytes. 22
13 Two SDI Bytes Separated By an SCI Operation. 23
14 Data Flow of VS1011e. 27
15 User's Memory Map. 40

1 Licenses

This Chapter intentionally left blank.

2 Disclaimer

All properties and figures are subject to change.

3 Definitions

B Byte, 8 bits.

b Bit.

Ki “Kibi” = 2^{10} = 1024 (IEC 60027-2).

Mi “Mebi” = 2^{20} = 1048576 (IEC 60027-2).

VS_DSP VLSI Solution’s DSP core.

W Word. In VS_DSP, instruction words are 32-bit and data words are 16-bit wide.

4 Characteristics & Specifications

4.1 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	DVDD+1.0[1]	V
Operating Temperature		-30	+85	°C
Functional Operating Temperature		-40	+95	°C
Storage Temperature		-65	+150	°C

[1] Must not exceed 3.6 V

4.2 Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Unit
Ambient Operating Temperature		-40		+85	°C
Analog and Digital Ground [1]	AGND DGND		0.0		V
Positive Analog	AVDD	2.5	2.7	3.6	V
Positive Digital, no clock doubler[2]	DVDD	2.3	2.5	3.6	V
Positive Digital, using clock doubler[2]	DVDD	2.75	3.0	3.6	V
Input Clock Frequency, no clock doubler[2]	XTALI	24	24.576	26	MHz
Input Clock Frequency, using clock doubler[2]	XTALI	12	12.288	13	MHz
Internal Clock Frequency	CLKI	24[3]	24.576	26	MHz
Master Clock Duty Cycle		40	50	60	%

[1] Must be connected together as close to the device as possible for latch-up immunity.
[2] For details on how to control the internal clock doubler, see Chapter 8.6.4, *SCI_CLOCKF*.
[3] The maximum sample rate that can be played is CLKI/512.
Thus, if CLKI is 24 MHz, 48 kHz sample rate is played at $\frac{24000000}{512} = 46875$ Hz, or 2.5% off-key.

4.3 Analog Characteristics

Unless otherwise noted: AVDD=2.5..3.6V, DVDD=2.3..3.6V, TA=-40..+85°C, XTALI=12..13MHz, internal Clock Doubler active. DAC tested with 1307.894 Hz full-scale output sinewave, measurement bandwidth 20..20000 Hz, analog output load: LEFT to GBUF 30Ω, RIGHT to GBUF 30Ω.

Parameter	Symbol	Min	Typ	Max	Unit
DAC Resolution			18		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	85		dB
Interchannel Isolation (Cross Talk)		50	75		dB
Interchannel Isolation (Cross Talk), with GBUF			40		dB
Interchannel Gain Mismatch		-0.5		0.5	dB
Frequency Response		-0.1		0.1	dB
Full Scale Output Voltage (Peak-to-peak)		1.4	1.6[1]	2.0	Vpp
Deviation from Linear Phase				5	°
Analog Output Load Resistance	AOLR	16	30[2]		Ω
Analog Output Load Capacitance				100	pF

[1] 3.2 volts can be achieved with +-to-+ wiring for mono difference sound.
[2] AOLR may be much lower, but below *Typical* distortion performance may be compromised.

4.4 Power Consumption

Average current tested with an MPEG 1.0 Layer III 128 kbit/s sample and generated sine, output at full volume, XTALI = 12.288 MHz, internal clock doubler enabled, DVDD = 2.5 V, AVDD = 2.7 V.

Parameter	Min	Typ	Max	Unit
Power Supply Consumption AVDD, Reset		0.5	30	µA
Power Supply Consumption DVDD, Reset		1	30	µA
Power Supply Consumption AVDD, sine test, 30Ω		20		mA
Power Supply Consumption AVDD, sine test, 30Ω + GBUF		39	50	mA
Power Supply Consumption DVDD, sine test		8	17	mA
Power Supply Consumption AVDD, no load		6		mA
Power Supply Consumption AVDD, output load 30Ω		10		mA
Power Supply Consumption AVDD, 30Ω + GBUF		16		mA
Power Supply Consumption DVDD		16		mA

4.5 Digital Characteristics

Parameter	Symbol	Min	Typ	Max	Unit
High-Level Input Voltage		$0.7\times$DVDD		DVDD+0.3[1]	V
Low-Level Input Voltage		-0.2		$0.3\times$DVDD	V
High-Level Output Voltage at I_O = -2.0 mA		$0.7\times$DVDD			V
Low-Level Output Voltage at I_O = 2.0 mA				$0.3\times$DVDD	V
Input Leakage Current		-1.0		1.0	μA
SPI Input Clock Frequency [2]				$\frac{CLKI}{7}$	MHz
Rise time of all output pins, load = 50 pF				50	ns

[1] Must not exceed 3.6V
[2] Value for SCI reads must be sufficiently below $\frac{CLKI}{6}$ to allow timing variation between the systems.
SCI and SDI writes allow $\frac{CLKI}{4}$.

4.6 Switching Characteristics - Boot Initialization

Parameter	Symbol	Min	Max	Unit
XRESET active time		2		XTALI
XRESET inactive to software ready			50000[1]	XTALI
Power on reset, rise time of DVDD		10		V/s

[1] DREQ rises when initialization is complete. You should not send any data or commands before that.

5 Packages and Pin Descriptions

5.1 Packages

LQFP-48 is lead (Pb) free and also RoHS-compliant packages. RoHS is a short name of *Directive 2002/95/EC on the restriction of the use of certain hazardous substances in electrical and electronic equipment*.

SOIC-28 is a lead-free RoHS-compliant package starting from VS1011e.

5.1.1 LQFP-48

LQFP-48 Pin Configuration

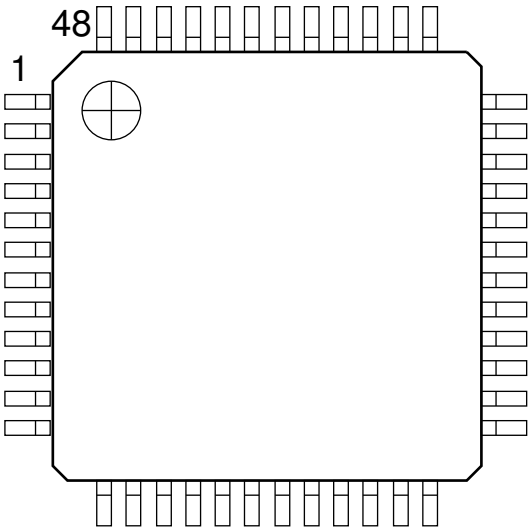
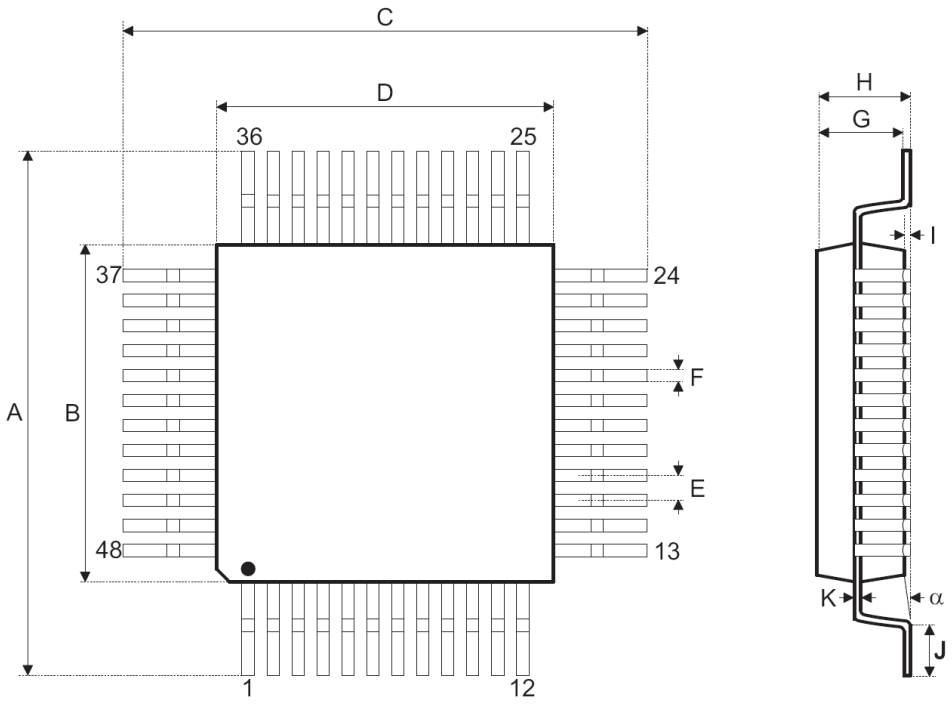


Figure 1: Pin Configuration, LQFP-48.

LQFP-48 package mechanical drawings are available at *http://www.vlsi.fi/fi/support/download.html*.

LQFP-48 Package Outline



Symbol	Dimensions in inch		
	Min	Nom.	Max
A	0.346	0.354	0.362
B	0.272	0.276	0.280
C	0.346	0.354	0.362
D	0.272	0.276	0.280
E	-	0.020	-
F	-	0.008	-
G	0.053	0.055	0.057
H	-	-	0.063
I	0.002	0.004	0.006
J	0.018	0.024	0.030
K	0.004	-	0.008
α	0°	-	7°

Symbol	Dimensions in mm		
	Min	Nom.	Max
A	8.80	9.00	9.20
B	6.90	7.00	7.10
C	8.80	9.00	9.20
D	6.90	7.00	7.10
E	-	0.50	-
F	-	0.20	-
G	1.35	1.40	1.45
H	-	-	1.60
I	0.05	0.10	0.15
J	0.45	0.60	0.75
K	0.09	-	0.20
α	0°	-	7°

Figure 2: Package Outline, LQFP-48.

5.1.2 SOIC-28

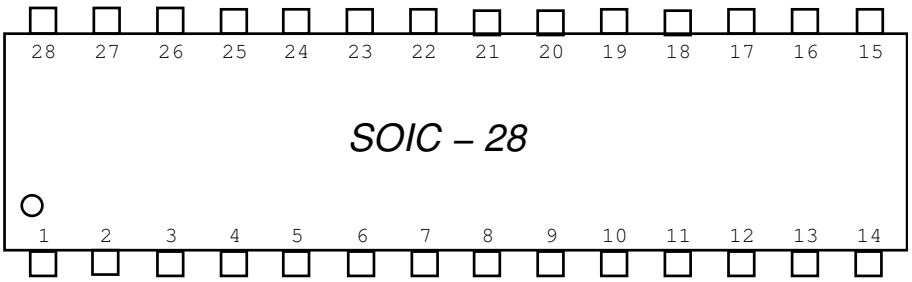


Figure 3: Pin Configuration, SOIC-28.

SOIC-28 package mechanical drawings are available at *http://www.vlsi.fi/fi/support/download.html* .

5.2 Pin Descriptions

5.2.1 LQFP-48 Pin Descriptions

Pin Name	LQFP-48	Pin Type	Function
XRESET	3	DI	active low asynchronous reset, schmitt-triggered
DGND0	4	PWR	digital ground
DVDD0	6	PWR	digital power supply
DREQ	8	DO	data request, input bus
GPIO2[2] / DCLK[1]	9	DI	general purpose IO 2 / serial input data bus clock
GPIO3[2] / SDATA[1]	10	DI	general purpose IO 3 / serial data input
XDCS[4] / BSYNC[1]	13	DI	data chip select / byte sync, connect to DVDD if not used
DVDD1	14	PWR	digital power supply
DGND1	16	PWR	digital ground
XTALO	17	AO	crystal output
XTALI	18	AI	crystal input
DVDD2	19	PWR	digital power supply
DGND2	20	PWR	digital ground
DGND3	21	PWR	digital ground
DGND4	22	PWR	digital ground
XCS[4]	23	DI	chip select input (active low)
SCLK[2]	28	DI	clock for serial bus
SI[2]	29	DI	serial input
SO	30	DO3	serial output, active when XCS=0, regardless of xRESET
TEST	32	DI	reserved for test, connect to DVDD
GPIO0 / SPIBOOT[2,3]	33	DIO	general purpose IO 0, use 100 kΩ pull-down resistor
GPIO1[2]	34	DIO	general purpose IO 1
AGND0	37	PWR	analog ground, low-noise reference
AVDD0	38	PWR	analog power supply
RIGHT	39	AO	right channel output
AGND1	40	PWR	analog ground
AGND2	41	PWR	analog ground
GBUF	42	AO	common buffer for headphones
AVDD1	43	PWR	analog power supply
RCAP	44	AIO	filtering capacitance for reference
AVDD2	45	PWR	analog power supply
LEFT	46	AO	left channel output
AGND3	47	PWR	analog ground

[1] First pin function is active in New Mode, latter in Compatibility Mode.

[2] If not used, use 100 kΩ pull-down resistor.

[3] Use 100 kΩ pull-down resistor. If pull-up is used instead, SPI Boot is tried. See Chapter 9.4 for details.

[4] If not used, use 100 kΩ pull-up resistor.

Pin types:

Type	Description
DI	Digital input, CMOS Input Pad
DO	Digital output, CMOS Input Pad
DIO	Digital input/output
DO3	Digital output, CMOS Tri-stated Output Pad

Type	Description
AI	Analog input
AO	Analog output
AIO	Analog input/output
PWR	Power supply pin

LQFP-48 no-connect pins: 1, 2, 5, 7, 11, 12, 15, 24, 25, 26, 27, 31, 35, 36, 48.

5.2.2 SOIC-28 Pin Descriptions

Pin Name	Pin	Pin Type	Function
DREQ	1	DO	data request, input bus
GPIO2[2] / DCLK[1]	2	DIO	serial input data bus clock
GPIO3[2] / SDATA[1]	3	DI	serial data input
XDCS[4] / BSYNC[1]	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[4]	11	DI	chip select input (active low)
SCLK[2]	12	DI	clock for serial bus
SI[2]	13	DI	serial input
SO	14	DO3	serial output, active when XCS=0, regardless of XRESET
TEST	15	DI	reserved for test, connect to DVDD
GPIO0 / SPIBOOT[2,3]	16	DIO	general purpose IO 0, use 100 kΩ pull-down resistor
GPIO1[2]	17	DIO	general purpose IO 1
AGND0	18	PWR	analog ground
AVDD0	19	PWR	analog power supply
RIGHT	20	AO	right channel output
AGND2	21	PWR	analog ground
RCAP	22	AIO	filtering 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
DGND0	27	PWR	digital ground
DVDD0	28	PWR	digital power supply

[1] First pin function is active in New Mode, latter in Compatibility Mode.

[2] If not used, use 100 kΩ pull-down resistor.

[3] Use 100 kΩ pull-down resistor. If pull-up is used instead, SPI Boot is tried. See Chapter 9.4 for details.

[4] If not used, use 100 kΩ pull-up resistor.

Pin types:

Type	Description
DI	Digital input, CMOS Input Pad
DO	Digital output, CMOS Input Pad
DIO	Digital input/output
DO3	Digital output, CMOS Tri-stated Output Pad

Type	Description
AI	Analog input
AO	Analog output
AIO	Analog input/output
PWR	Power supply pin

6 Connection Diagram, LQFP-48

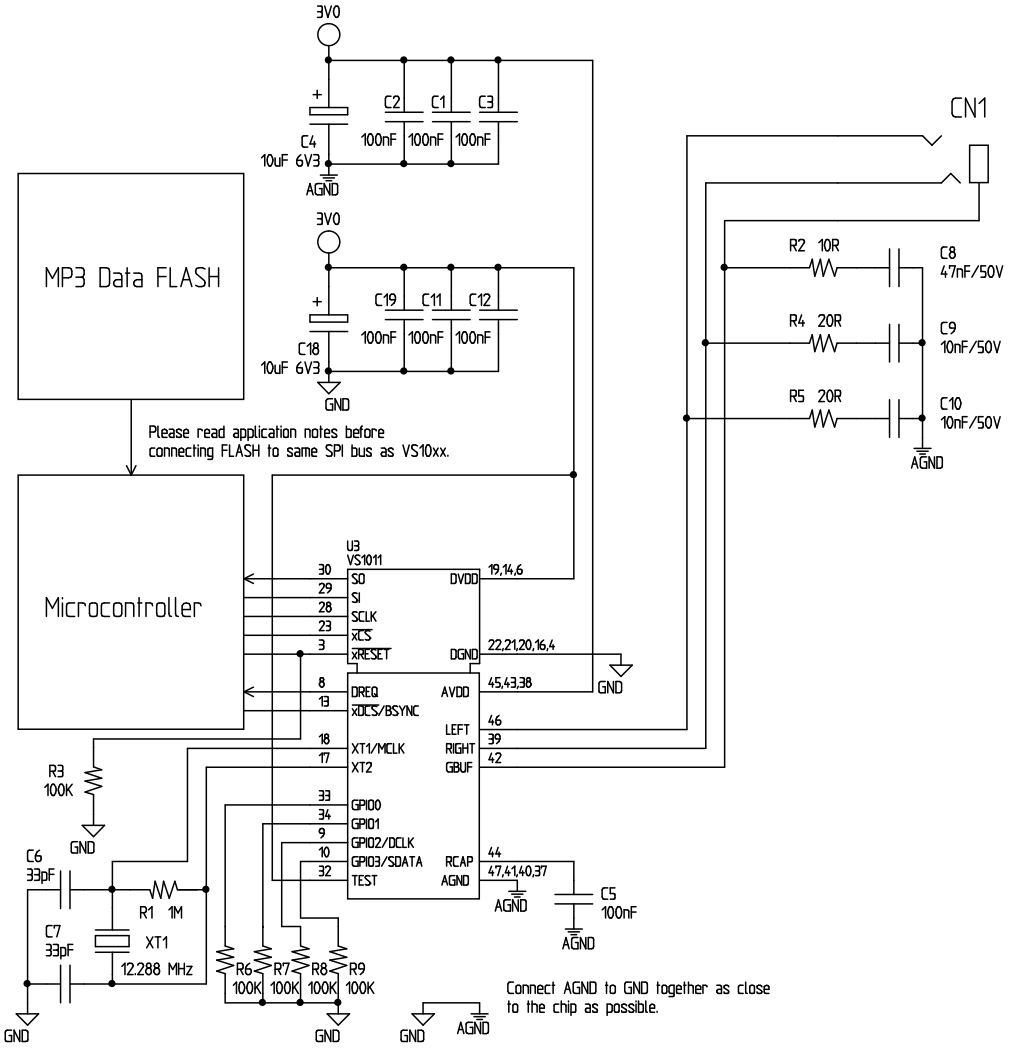


Figure 4: Typical Connection Diagram Using LQFP-48.

The common buffer GBUF can be used for common voltage (1.23 V) for earphones. This will eliminate the need for large isolation capacitors on line outputs, and thus the audio output pins from VS1011e may be connected directly to the earphone connector.

If GBUF is not used, LEFT and RIGHT must be provided with 1-100 μF AC coupling capacitors depending on load resistance.

Note: This connection assumes SM_SDINEW is active (see Chapter 8.6.1). If also SM_SDISHARE is used, xDCS should have a pull-up resistor (see Chapter 7.2.1).

7 SPI Buses

7.1 General

The SPI Bus - that was originally used in some Motorola devices - has been used for both VS1011e's Serial Data Interface SDI (Chapters 7.3 and 8.4) and Serial Control Interface SCI (Chapters 7.5 and 8.5).

7.2 SPI Bus Pin Descriptions

7.2.1 VS10xx Native Modes (New Mode, recommended)

These modes are active on VS1011e when SM_SDINEW is set to 1. DCLK and SDATA are not used for data transfer and they can be used as general-purpose I/O pins (GPIO2 and GPIO3). BSYNC function changes to data interface chip select (XDCS).

SDI Pin	SCI Pin	Description
XDCS	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. If SM_SDISHARE is 1, pin XDCS is not used, but the signal is generated internally by inverting XCS.
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.
SI		Serial input. If a chip select is active, SI is sampled on the rising CLK edge.
-	SO	Serial output. In reads, data is shifted out on the falling SCK edge. In writes SO is at a high impedance state.

7.2.2 VS1001 Compatibility Mode (deprecated, do not use in new designs)

This mode is active when SM_SDINEW is 0 (default). In this mode, DCLK, SDATA and BSYNC are active.

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.
BSYNC	-	SDI data is synchronized with a rising edge of BSYNC.
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.
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.

7.3 Serial Protocol for Serial Data Interface (SDI)

7.3.1 General

The serial data interface operates in slave mode so the DCLK signal must be generated by an external circuit.

Data (SDATA signal) can be clocked in at either the rising or falling edge of DCLK (Chapter 8.6).

VS1011e assumes its data input to be byte-sychronized. SDI bytes may be transmitted either MSb or LSb first, depending of contents of SCI_MODE (Chapter 8.6).

7.3.2 SDI in VS10xx Native Modes (New Mode, recommended)

In VS10xx native modes, when SM_NEWMODE is set to 1, (the default reset value is 0 in VS1011e, i.e. compatibility mode), byte synchronization is achieved by XDCS (or XCS if SM_SDISHARE is 1). The state of XDCS (or XCS) may not change while a data byte transfer is in progress. To always maintain data synchronization even if there may be glitches in the boards using VS1011e, it is recommended to turn XDCS (or XCS) every now and then, for instance once after every flash data block or a few kilobytes, just to keep sure the host and VS1011e are in sync.

For new designs, using VS10xx native modes are recommended.

If SM_SDISHARE is 1, the XDCS signal is internally generated by inverting the XCS input.

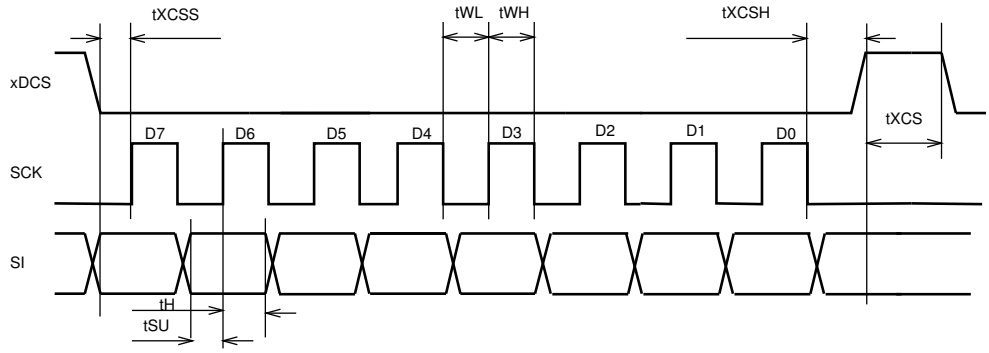


Figure 5: SDI Timing Diagram.

Symbol	Min	Max	Unit
tXCSS	5		ns
tSU	0		ns
tH	2		CLKI cycles
tWL	2		CLKI cycles
tWH	2		CLKI cycles
tXCSH	1		CLKI cycles
tXCS	0		CLKI cycles

Note: xDCS is not required to be deasserted between bytes, thus tXCS is 0.

Note: Although the timing is derived from the internal clock CLKI, the system always starts up in $1.0\times$ mode, thus CLKI=XTALI. After you have configured a higher clock through SCI_CLOCKF and waited for DREQ to rise, you can use a higher SPI speed as well.

7.3.3 SDI in VS1001 Compatibility Mode

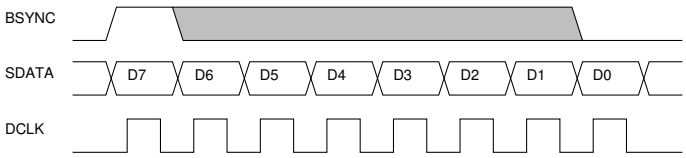


Figure 6: BSYNC Signal - one byte transfer.

When VS1011e is running in VS1001 compatibility mode, a BSYNC signal must be generated to ensure correct bit-alignment of the input bitstream. 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 '1' when the last bit is received, the receiver stays active and next 8 bits are also received.

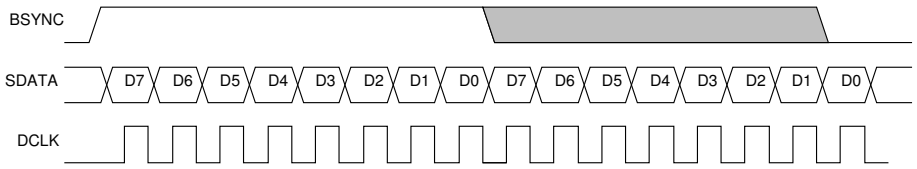


Figure 7: BSYNC Signal - two byte transfer.

Using VS1001 compatibility mode in new designs is strongly discouraged.

7.4 Data Request Pin DREQ

The DREQ pin/signal is used to signal if VS1011e's FIFO is capable of receiving data. If DREQ is high, VS1011e can take at least 32 bytes of SDI data or one SCI command. When these criteria are not met, DREQ is turned low, and the sender should stop transferring new data.

Because of a 32-byte safety area, the sender may send up to 32 bytes of SDI data at a time without checking the status of DREQ, making controlling VS1011e 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.

Note: In VS10XX products up to VS1002, DREQ was only used for SDI. In VS1011e DREQ is also used to tell the status of SCI.

7.5 Serial Protocol for Serial Command Interface (SCI)

7.5.1 General

The serial bus protocol for the Serial Command Interface SCI (Chapter 8.5) 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. Data bits are read at the rising clock edge, so the user should update data at the falling clock edge. Bytes are always sent MSb firrst.

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: VS1011e sets DREQ low after each SCI operation. The duration depends on the operation. It is not allowed to start a new SCI/SDI operation before DREQ is high again.

7.5.2 SCI Read

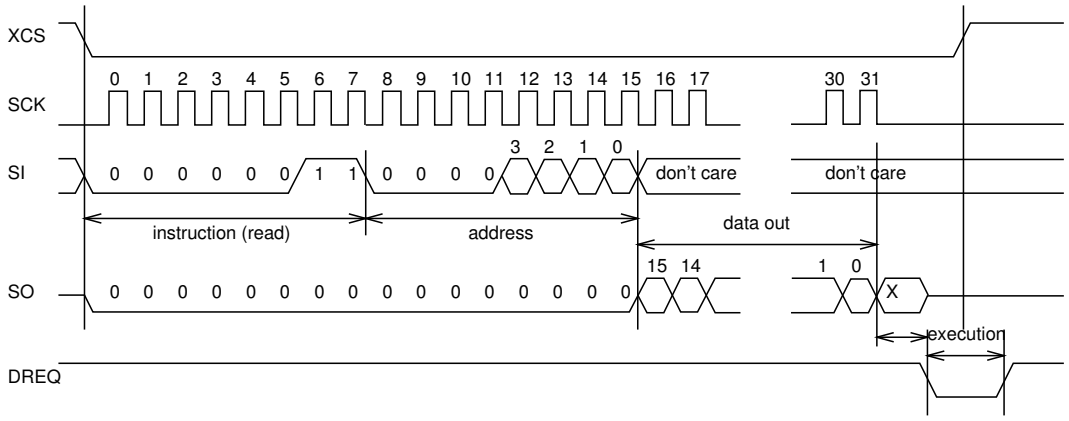


Figure 8: SCI Word Read

VS1011e registers are read by the following sequence, as shown in Figure 8. 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 data has been shifted out.

DREQ is driven low for a short while when in a read operation by the chip. This is a very short time and doesn't require special user attention.

7.5.3 SCI Write

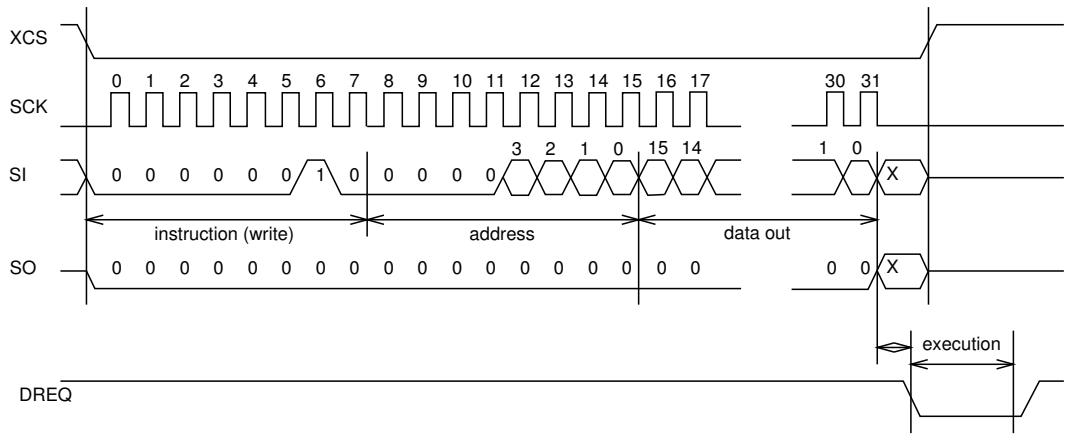


Figure 9: SCI Word Write

VS1011e registers are written to using the following sequence, as shown in Figure 9. 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 and the last clock has been sent, XCS should be pulled high to end the WRITE sequence.

After the last bit has been sent, DREQ is driven low for the duration of the register update, marked “execution” in the figure. The time varies depending on the register and its contents (see table in Chapter 8.6 for details). If the maximum time is longer than what it takes from the microcontroller to feed the next SCI command or SDI byte, it is not allowed to finish a new SCI/SDI operation before DREQ has risen up again.

7.6 SPI Timing Diagram

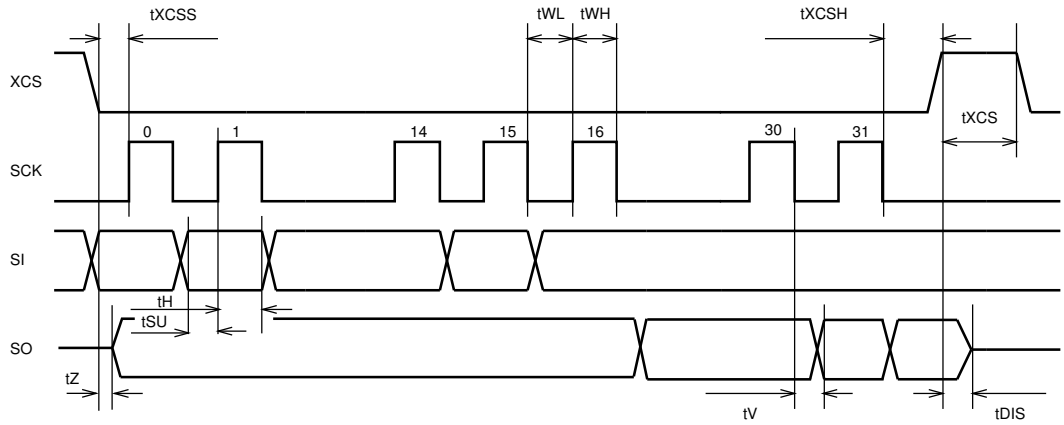


Figure 10: SPI Timing Diagram.

Symbol	Min	Max	Unit
tXCSS	5		ns
tSU	-26		ns
tH	2		CLKI cycles
tZ	0		ns
tWL	2		CLKI cycles
tWH	2		CLKI cycles
tV		2 (+ 25ns[1])	CLKI cycles
tXCSH	-26		ns
tXCS	2		CLKI cycles
tDIS		10	ns

[1] 25ns is when pin loaded with 100pF capacitance. The time is shorter with lower capacitance.

Note: As tWL and tWH, as well as tH require at least 2 clock cycles, the maximum speed for the SPI bus that can be used for read operations is 1/6 of VS1011e’s external clock speed CLKI. For write operations maximum speed is 1/4 of CLKI. Because of asynchronous clocks between the systems, the actual speed must be a bit slower to allow timing variations.

Note: Negative numbers mean that the signal can change in different order from what is shown in the diagram.

7.7 SPI Examples with SM_SDINEW and SM_SDISHARED set

7.7.1 Two SCI Writes

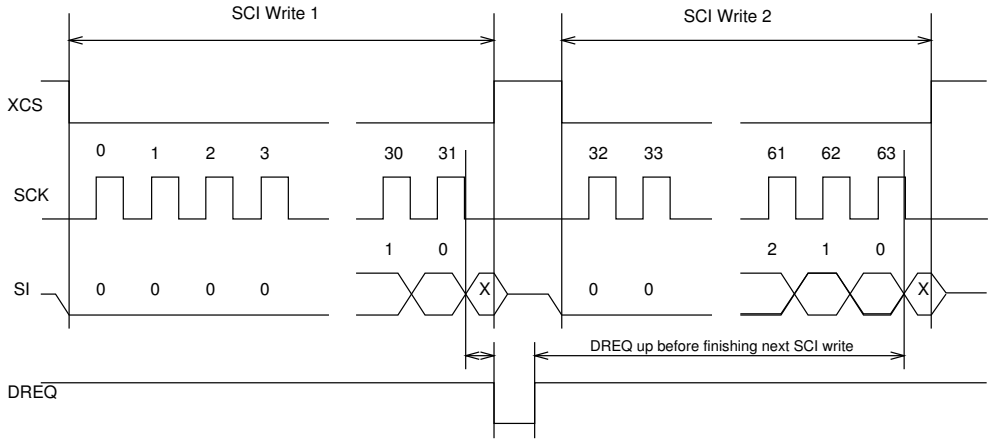


Figure 11: Two SCI Operations.

Figure 11 shows two consecutive SCI operations. Note that xCS *must* be raised to inactive state between the writes. Also DREQ must be respected as shown in the figure.

7.7.2 Two SDI Bytes

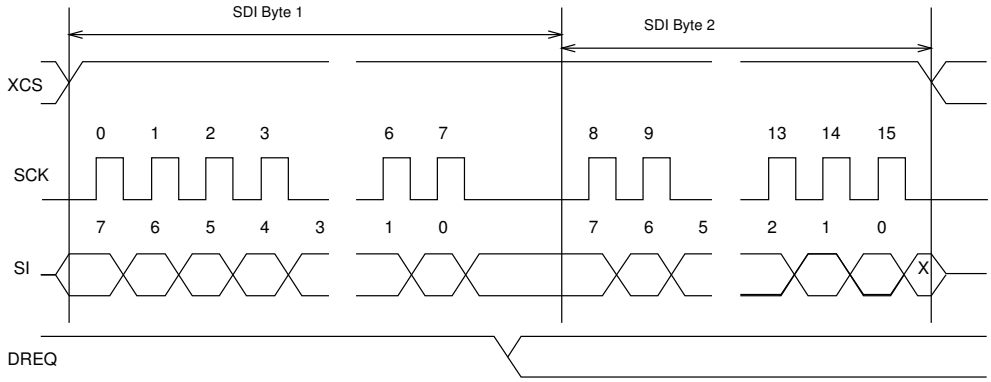


Figure 12: Two SDI Bytes.

SDI data is synchronized with a raising edge of xCS as shown in Figure 12. However, every byte doesn't need separate synchronization.

7.7.3 SCI Operation in Middle of Two SDI Bytes

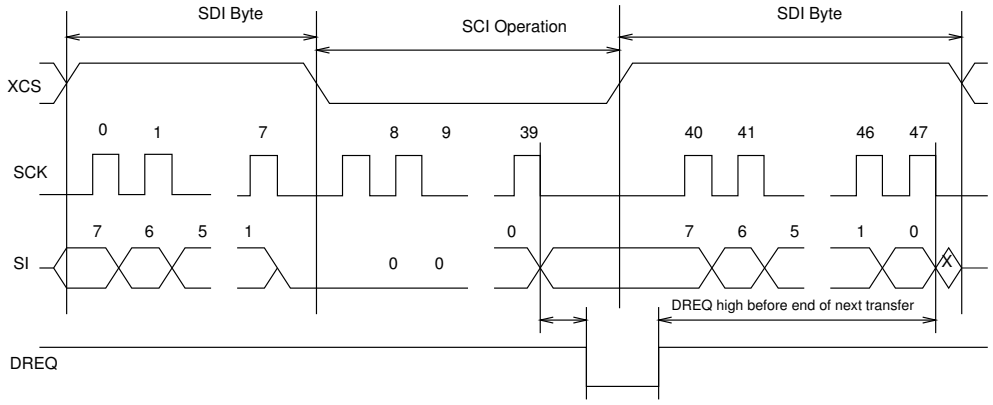


Figure 13: Two SDI Bytes Separated By an SCI Operation.

Figure 13 shows how an SCI operation is embedded in between SDI operations. xCS edges are used to synchronize both SDI and SCI. Remember to respect DREQ as shown in the figure.

8 Functional Description

8.1 Main Features

VS1011e is based on a proprietary digital signal processor, VS_DSP. It contains all the code and data memory needed for MPEG, WAV PCM and WAV IMA ADPCM audio decoding, together with serial interfaces, a multirate stereo audio DAC and analog output amplifiers and filters.

VS1011e can play all MPEG 1.0, and 2.0 layer I, II and III files, as well as MPEG 2.5 layer III files, with all sample rates and bitrates, including variable bitrate (VBR) for layer III. Note, that decoding of layers I and II must be activated separately.

8.2 Supported Audio Codecs

Conventions	
Mark	**Description**
+	Format is supported
-	Format exists but is not supported
?	Format not tested
	Format doesn't exist

8.2.1 Supported MP1 (MPEG layer I) Formats

MPEG 1.0:

Samplerate / Hz	Bitrate / kbit/s													
	32	64	96	128	160	192	224	256	288	320	352	384	416	448
48000	+	+	+	+	+	+	+	+	+	+	+	+	+	+
44100	+	+	+	+	+	+	+	+	+	+	+	+	+	+
32000	+	+	+	+	+	+	+	+	+	+	+	+	+	+

MPEG 2.0:

Samplerate / Hz	Bitrate / kbit/s													
	32	48	56	64	80	96	112	128	144	160	176	192	224	256
24000	?	?	?	?	?	?	?	?	?	?	?	?	?	?
22050	?	?	?	?	?	?	?	?	?	?	?	?	?	?
16000	?	?	?	?	?	?	?	?	?	?	?	?	?	?

8.2.2 Supported MP2 (MPEG layer II) Formats

MPEG 1.0:

Samplerate / Hz	Bitrate / kbit/s													
	32	48	56	64	80	96	112	128	160	192	224	256	320	384
48000	+	+	+	+	+	+	+	+	+	+	+	+	+	+
44100	+	+	+	+	+	+	+	+	+	+	+	+	+	+
32000	+	+	+	+	+	+	+	+	+	+	+	+	+	+

MPEG 2.0:

Samplerate / Hz	Bitrate / kbit/s													
	8	16	24	32	40	48	56	64	80	96	112	128	144	160
24000	+	+	+	+	+	+	+	+	+	+	+	+	+	+
22050	+	+	+	+	+	+	+	+	+	+	+	+	+	+
16000	+	+	+	+	+	+	+	+	+	+	+	+	+	+

8.2.3 Supported MP3 (MPEG layer III) Formats

MPEG 1.0[1]:

Samplerate / Hz	Bitrate / kbit/s													
	32	40	48	56	64	80	96	112	128	160	192	224	256	320
48000	+	+	+	+	+	+	+	+	+	+	+	+	+	+
44100	+	+	+	+	+	+	+	+	+	+	+	+	+	+
32000	+	+	+	+	+	+	+	+	+	+	+	+	+	+

MPEG 2.0[1]:

Samplerate / Hz	Bitrate / kbit/s													
	8	16	24	32	40	48	56	64	80	96	112	128	144	160
24000	+	+	+	+	+	+	+	+	+	+	+	+	+	+
22050	+	+	+	+	+	+	+	+	+	+	+	+	+	+
16000	+	+	+	+	+	+	+	+	+	+	+	+	+	+

MPEG 2.5[1]:

Samplerate / Hz	Bitrate / kbit/s													
	8	16	24	32	40	48	56	64	80	96	112	128	144	160
12000	+	+	+	+	+	+	+	+	+	+	+	+	+	+
11025	+	+	+	+	+	+	+	+	+	+	+	+	+	+
8000	+	+	+	+	+	+	+	+	+	+	+	+	+	+

[1] Also all variable bitrate (VBR) formats are supported.

Note: 24.0 MHz internal clock (24.0 MHz external clock or 12.0 MHz external clock with clock-doubler) is enough for VS1011e to be able to decode all bitrates and sample rates with bass enhancer and treble control active.

8.2.4 Supported RIFF WAV Formats

The most common RIFF WAV subformats are supported.

Format	Name	Supported	Comments
0x01	PCM	+	16 and 8 bits, any sample rate ≤ 48kHz
0x02	ADPCM	-	
0x03	IEEE_FLOAT	-	
0x06	ALAW	-	
0x07	MULAW	-	
0x10	OKI_ADPCM	-	
0x11	IMA_ADPCM	+	Any sample rate ≤ 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	+	For supported MP3 modes, see Chapter 8.2.3
0x64	G726_ADPCM	-	
0x65	G722_ADPCM	-	

8.3 Data Flow of VS1011e

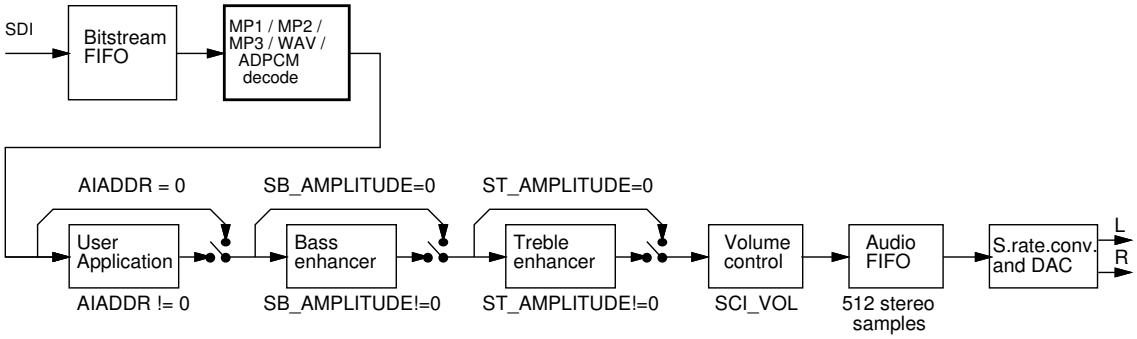


Figure 14: Data Flow of VS1011e.

First, depending on the audio data, MPEG or WAV audio is received and decoded from the SDI bus.

After decoding, if SCI_AIADDR is non-zero, application code is executed from the address pointed to by that register. For more details, see VS10XX Application Note: User Applications.

Then data may be sent to the Bass and Treble Enhancer depending on SCI_BASS.

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

The Audio FIFO holds the data, which is read by the Audio interrupt (Chapter 10.9.1) and fed to the sample rate converter and DACs. The size of the audio FIFO is 512 stereo (2×16-bit) samples, or 2 KiB.

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

8.4 Serial Data Interface (SDI)

The serial data interface is meant for transferring compressed MP3 audio data as well as WAV data.

Also several different tests may be activated through SDI as described in Chapter 9.

8.5 Serial Control Interface (SCI)

The serial control interface is compatible with the SPI bus specification. Data transfers are always 16 bits. VS1011e is controlled by writing and reading the registers of the interface.

The main controls of the control interface are:

- control of the operation mode, clock, and builtin effects
- access to status information and header data
- access to encoded digital data
- uploading user programs
- feeding input data

8.6 SCI Registers

SCI registers, prefix SCI_					
Reg	**Type**	**Reset**	**Time**[1]	**Abbrev[bits]**	**Description**
0x0	rw	0	70 CLKI[4]	MODE	Mode control
0x1	rw	0x2C[3]	40 CLKI	STATUS	Status of VS1011e
0x2	rw	0	2100 CLKI	BASS	Built-in bass/treble enhancer
0x3	rw	0	80 XTALI	CLOCKF	Clock freq + multiplier
0x4	rw	0	40 CLKI	DECODE_TIME	Decode time in seconds
0x5	rw	0	3200 CLKI	AUDATA	Misc. audio data
0x6	rw	0	80 CLKI	WRAM	RAM write/read
0x7	rw	0	80 CLKI	WRAMADDR	Base address for RAM write/read
0x8	r	0	-	HDAT0	Stream header data 0
0x9	r	0	-	HDAT1	Stream header data 1
0xA	rw	0	3200 CLKI[2]	AIADDR	Start address of application
0xB	rw	0	2100 CLKI	VOL	Volume control
0xC	rw	0	50 CLKI[2]	AICTRL0	Application control register 0
0xD	rw	0	50 CLKI[2]	AICTRL1	Application control register 1
0xE	rw	0	50 CLKI[2]	AICTRL2	Application control register 2
0xF	rw	0	50 CLKI[2]	AICTRL3	Application control register 3

[1] This is the worst-case time that DREQ stays low after writing to this register. The user may choose to skip the DREQ check for those register writes that take less than 100 clock cycles to execute.

[2] In addition, the cycles spent in the user application routine must be counted.

[3] Firmware changes the value of this register immediately to 0x28, and in less than 100 ms to 0x20.

[4] When mode register write specifies a software reset the worst-case time is 9600 XTALI cycles.

Note that if DREQ is low when an SCI write is done, DREQ also stays low after SCI write processing.

8.6.1 SCI_MODE (RW)

SCI_MODE is used to control operation of VS1011e.

Bit	Name	Function	Value	Description
0	SM_DIFF	Differential	0 1	normal in-phase audio left channel inverted
1	SM_LAYER12	Allow MPEG layers I & II	0 1	no yes
2	SM_RESET	Soft reset	0 1	no reset reset
3	SM_OUTOFWAV	Jump out of WAV decoding	0 1	no yes
4	SM_SETTOZERO1	set to zero	0 1	right wrong
5	SM_TESTS	Allow SDI tests	0 1	not allowed allowed
6	SM_STREAM	Stream mode	0 1	no yes
7	SM_SETTOZERO2	set to zero	0 1	right wrong
8	SM_DACT	DCLK active edge	0 1	rising falling
9	SM_SDIORD	SDI bit order	0 1	MSb first MSb last
10	SM_SDISHARE	Share SPI chip select	0 1	no yes
11	SM_SDINEW	VS10xx native SPI modes	0 1	no yes
12	SM_SETTOZERO3	set to zero	0 1	right wrong
13	SM_SETTOZERO4	set to zero	0 1	right wrong

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

SM_LAYER12 determines whether it is allowed to decode MPEG 1 and 2 layers I and II in addition to layer III.

By setting SM_RESET to 1, the player is software reset. This bit clears automatically.

When the user decoding a WAV file wants to get out of the file without playing it to the end, set SM_OUTOFWAV, and send zeros to VS1002e until SM_OUTOFWAV is again zero. If the user doesn't want to check SM_OUTOFWAV, send 128 zeros.

If SM_TESTS is set, SDI tests are allowed. For more details on SDI tests, look at Chapter 9.7.

SM_STREAM activates VS1011e's stream mode. In this mode, data should be sent with as even intervals as possible (and preferable with data blocks of less than 512 bytes), and VS1011e makes every attempt to keep its input buffer half full by changing its playback speed up to 5%. For best quality sound, the average speed error should be within 0.5%, the bitrate should not exceed 160 kbit/s and VBR should not be used. For details, see VS10XX Application Note: Streaming.

SM_DACT defines the active edge of data clock for SDI. If clear data is read at the rising edge, and if set data is read at the falling edge.

When SM_SDIORD is clear, bytes on SDI are sent as a default MSb first. By setting SM_SDIORD, the user may reverse the bit order for SDI, i.e. bit 0 is received first and bit 7 last. Bytes are, however, still sent in the default order. This register bit has no effect on the SCI bus.

Setting SM_SDISHARE makes SCI and SDI share the same chip select, as explained in Chapter 7.2, if also SM_SDINEW is set.

Setting SM_SDINEW will activate VS10xx native serial modes as described in Chapters 7.2.1 and 7.3.2.

8.6.2 SCI_STATUS (RW)

SCI_STATUS contains information on the current status of VS1011e and lets the user shutdown the chip without audio glitches.

Name	Bits	Description
SS_VER	6:4	Version
SS_APDOWN2	3	Analog driver powerdown
SS_APDOWN1	2	Analog internal powerdown
SS_AVOL	1:0	Analog volume control

SS_VER is 0 for VS1001, 1 for VS1011, 2 for VS1002 and VS1011e, and 3 for vs1003.

You can use SCI_MODE to distinguish between VS1002 and VS1011e. After reset VS1011e has SM_SDINEW=0, while VS1002 has SM_SDINEW=1.

SS_APDOWN2 controls analog driver powerdown. Normally this bit is controlled by the system firmware. However, if the user wants to powerdown VS1011e with a minimum power-off transient, turn this bit to 1, then wait for at least a few milliseconds before activating reset.

SS_APDOWN1 controls internal analog powerdown. This bit is meant to be used by the system firmware only.

SS_AVOL is the analog volume control: 0 = -0 dB, 1 = -6 dB, 3 = -12 dB. This register is meant to be used automatically by the system firmware only.

8.6.3 SCI_BASS (RW)

Name	Bits	Description
ST_AMPLITUDE	15:12	Treble Control in 1.5 dB steps (-8..7, 0 = off)
ST_FREQLIMIT	11:8	Lower limit frequency in 1000 Hz steps (0..15)
SB_AMPLITUDE	7:4	Bass Enhancement in 1 dB steps (0..15, 0 = off)
SB_FREQLIMIT	3:0	Lower limit frequency in 10 Hz steps (2..15)

The Bass Enhancer VSBE is a powerful bass boosting DSP algorithm, which tries to take the most out of the users earphones without causing clipping.

VSBE is activated when SB_AMPLITUDE is non-zero. SB_AMPLITUDE should be set to the user's preferences, and SB_FREQLIMIT to roughly 1.5 times the lowest frequency the user's audio system can reproduce. For example setting SCI_BASS to 0x00f6 will have 15 dB enhancement below 60 Hz.

Note: Because VSBE tries to avoid clipping, it gives the best bass boost with dynamical music material, or when the playback volume is not set to maximum. It also does not create bass: the source material must have some bass to begin with.

Treble Control VSTC is activated when ST_AMPLITUDE is non-zero. For example setting SCI_BASS to 0x7a00 will have 10.5 dB treble enhancement at and above 10 kHz.

Bass Enhancer uses about 2.1 MIPS and Treble Control 1.2 MIPS at 44100 Hz sample rate. Both can be on simultaneously.

8.6.4 SCI_CLOCKF (RW)

SCI_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 $\frac{XTALI}{2000}$ (XTALI is in Hz). Values may be between 0..32767, although hardware limits the highest allowed speed. Also, with speeds lower than 24.576 MHz all sample rates are no longer available. For example with 24 MHz clock 48 kHz is played 2.5% off-key.

Setting the MSB of SCI_CLOCKF to 1 activates internal clock-doubling when the sample rate is next configured.

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

Example 1: For a 24.576 MHz XTALI the value is either $\frac{24576000}{2000} = 12288$, or the reset value 0.

Example 2: For a 12.288 MHz XTALI with clock doubler the value is $0x8000 + \frac{12288000}{2000} = 0x9800$.

Example 3: For a 13 MHz XTALI with clock doubler for a 26 MHz CLKI, the value is $0x8000 + \frac{13000000}{2000} = 39268$.

8.6.5 SCI_DECODE_TIME (RW)

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

The user may change the value of this register. However, in that case the new value should be written twice.

SCI_DECODE_TIME is reset at every software reset.

8.6.6 SCI_AUDATA (RW)

When decoding correct data, the current sample rate and number of channels can be found in bits 15:1 and 0 of SCI_AUDATA, respectively. Bits 15:1 contain the sample rate divided by two, and bit 0 is 0 for mono data and 1 for stereo. Writing to this register will change the sample rate on the run to the number given.

Example: 44100 Hz stereo data reads as 0xAC45 (44101).
Example: 11025 Hz mono data reads as 0x2B10 (11024).
Example: Writing 0xAC80 sets sample rate to 44160 Hz, stereo mode does not change.

8.6.7 SCI_WRAM (RW)

SCI_WRAM is used to upload application programs and data to instruction and data RAMs. The start address must be initialized by writing to SCI_WRAMADDR prior to the first write/read of SCI_WRAM. As 16 bits of data can be transferred with one SCI_WRAM write/read, and the instruction word is 32 bits long, two consecutive writes/reads are needed for each instruction word. The byte order is big-endian (i.e. MSBs first). After each full-word write/read, the internal pointer is autoincremented.

8.6.8 SCI_WRAMADDR (RW)

SCI_WRAMADDR is used to set the program address and memory bus for following SCI_WRAM writes/reads.

SM_WRAMADDR Start...End	Dest. addr. Start...End	Bits/ Word	Description
0x1380...0x13FF	0x1380...0x13FF	16	X data RAM
0x4780...0x47FF	0x0780...0x07FF	16	Y data RAM
0x8030...0x84FF	0x0030...0x04FF	32	Instruction RAM
0xC000...0xFFFF	0xC000...0xFFFF	16	I/O

8.6.9 SCI_HDAT0 and SCI_HDAT1 (R)

Bit	Function	Value	Explanation
HDAT1[15:5]	syncword	2047	stream valid
HDAT1[4:3]	ID	3	ISO 11172-3 MPG 1.0
		2	ISO 13818-3 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	No CRC
		0	CRC protected
HDAT0[15:12]	bitrate		see bitrate table
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		see ISO 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, SCI_HDAT0 and SCI_HDAT1 contain header information that is extracted from MPEG stream being currently being decoded. Right after resetting VS1011e, 0 is automatically written to both registers, indicating no data has been found yet.

The “sample rate” field in SCI_HDAT0 is interpreted according to the following table:

“sample rate”	ID=3 / Hz	ID=2 / Hz	ID=0,1 / Hz
3	-	-	-
2	32000	16000	8000
1	48000	24000	12000
0	44100	22050	11025

The “bitrate” field in HDAT0 is read according to the following table:

"bitrate"	Layer I		Layer II		Layer III	
	ID=3	ID=0,1,2	ID=3	ID=0,1,2	ID=3	ID=0,1,2
	kbit/s		kbit/s		kbit/s	
15	forbidden	forbidden	forbidden	forbidden	forbidden	forbidden
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	-	-	-	-	-	-

When decoding a WAV file, SPI_HDAT0 and SPI_HDAT1 read as 0x7761 and 0x7665, respectively.

8.6.10 SCI_AIADDR (RW)

SCI_AIADDR indicates the start address of the application code written earlier with SCI_WRAMADDR and SCI_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 VS10XX Application Note: User Applications.

8.6.11 SCI_VOL (RW)

SCI_VOL is a volume control for the player hardware. For each channel, a value in the range of 0..254 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 is 0xFEFE.

Example: for a volume of -2.0 dB for the left channel and -3.5 dB for the right channel: (4*256) + 7 = 0x407. Note, that at startup volume is set to full volume. Resetting the software does not reset the volume setting.

Note: Setting SCI_VOL to 0xFFFF will activate analog powerdown mode.

8.6.12 SCI_AICTRL[x] (RW)

SCI_AICTRL[x] registers (x=[0 .. 3]) can be used to access the user's application program.

9 Operation

9.1 Clocking

VS1011e operates on a single, nominally 24.576 MHz fundamental frequency master clock. This clock can be generated by external circuitry (connected to pin XTALI) or by the internal clock crystal interface (pins XTALI and XTALO). Also, 12.288 MHz external clock can be internally clock-doubled to 24.576 MHz. This clock is sufficient to support a high quality audio output for all codecs, sample rates and bitrates, with bass and treble enhancers.

9.2 Hardware Reset

When the XRESET -signal is driven low, VS1011e 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 VS1011e 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), the user should set such basic software registers as SCI_VOL for volume (and SCI_CLOCKF if the input clock is anything else than 24.576 MHz) before starting decoding.

9.3 Software Reset

In some cases the decoder software has to be reset. This is done by activating bit 2 in SCI_MODE register (Chapter 8.6.1). 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 VS1011e is run at 24.576 MHz. After DREQ is up, you may continue playback as usual.

If you want to make sure VS1011e doesn't cut the ending of low-bitrate data streams and you want to do a software reset, it is recommended to feed 2048 zeros to the SDI bus after the file and before the reset.

9.4 SPI Boot

If GPIO0 is set with a pull-up resistor to 1 at boot time, VS1011e tries to boot from external SPI memory.

SPI boot redefines the following pins:

Normal Mode	SPI Boot Mode
GPIO0	xCS
GPIO1	CLK
DREQ	MOSI
GPIO2	MISO

The memory has to be an SPI Bus Serial EEPROM with 16-bit addresses (i.e. at least 1 KiB). The serial speed used by VS1011e is 490 kHz with the nominal 24.576 MHz clock. The first three bytes in the memory have to be 0x50, 0x26, 0x48. The exact record format is explained in the Application Notes for VS10XX.

If SPI boot succeeds, SCI_MODE is left with value 0x0800.

9.5 Play/Decode

This is the normal operation mode of VS1011e. SDI data is decoded. Decoded samples are converted to analog domain by the internal DAC. If no decodable data is found, SCI_HDAT0 and SCI_HDAT1 are set to 0 and analog outputs are muted.

When there is no input for decoding, VS1011e goes into idle mode (lower power consumption than during decoding) and actively monitors the serial data input for valid data.

9.6 Feeding PCM data

VS1011e can be used as a PCM decoder by sending to it a WAV file header. If the length sent in the WAV file is 0 or 0xFFFFFFF, VS1011e will stay in PCM mode indefinitely. 8-bit linear and 16-bit linear audio is supported in mono or stereo.

9.7 SDI Tests

There are several test modes in VS1011e, which allow the user to perform memory tests, SCI bus tests, and several different sine wave tests.

All tests are started in a similar way: VS1011e is hardware reset, SM_TESTS is set, and then a test command is sent to the SDI bus. Each test is started by sending a 4-byte special command sequence, followed by 4 zeros. The sequences are described below.

9.7.1 Sine Test

Sine test is initialized with the 8-byte sequence 0x53 0xEF 0x6E n 0 0 0 0, where n defines the sine test to use. n is defined as follows:

n bits		
Name	**Bits**	**Description**
F_sIdx	7:5	Sample rate index
S	4:0	Sine skip speed

F_sIdx	F_s
0	44100 Hz
1	48000 Hz
2	32000 Hz
3	22050 Hz
4	24000 Hz
5	16000 Hz
6	11025 Hz
7	12000 Hz

The frequency of the sine to be output can now be calculated from $F = F_s \times \frac{S}{128}$.

Example: Sine test is activated with value 126, which is 0b01111110. Breaking n to its components, $F_sIdx = 0b011 = 3$ and thus $F_s = 22050Hz$. $S = 0b11110 = 30$, and thus the final sine frequency $F = 22050Hz \times \frac{30}{128} \approx 5168Hz$.

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

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

9.7.2 Pin Test

Pin test is activated with the 8-byte sequence 0x50 0xED 0x6E 0x54 0 0 0 0. This test is meant for chip production testing only.

9.7.3 Memory Test

Memory test mode is initialized with the 8-byte sequence 0x4D 0xEA 0x6D 0x54 0 0 0 0. After this sequence, wait for 200000 clock cycles. The result can be read from the SCI register SCI_HDAT0, and 'one' bits are interpreted as follows:

Bit(s)	Mask	Meaning
15	0x8000	Test finished
14:7		Unused
6	0x0040	Mux test succeeded
5	0x0020	Good I RAM
4	0x0010	Good Y RAM
3	0x0008	Good X RAM
2	0x0004	Good I ROM
1	0x0002	Good Y ROM
0	0x0001	Good X ROM
	0x807f	All ok

Memory tests overwrite the current contents of the RAM memories.

9.7.4 SCI Test

Sci test is initialized with the 8-byte sequence 0x53 0x70 0xEE n 0 0 0 0, where n is the register number to test. The content of the given register is read and copied to SCI_HDAT0. If the register to be tested is HDAT0, the result is copied to SCI_HDAT1.

Example: if n is 0, contents of SCI register 0 (SCI_MODE) is copied to SCI_HDAT0.

10 VS1011e Registers

10.1 Who Needs to Read This Chapter

User software is required when a user wishes to add some own functionality like DSP effects to VS1011e.

However, most users of VS1011e don't need to worry about writing their own code, or about this chapter, including those who only download software plug-ins from VLSI Solution's Web site.

10.2 The Processor Core

VS_DSP is a 16/32-bit DSP processor core that also had extensive all-purpose processor features. 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. VLSI Solution also offers a full Integrated Development Environment VSIDE for full debug capabilities.

10.3 VS1011e Memory Map

VS1011e's Memory Map is shown in Figure 15.

10.4 SCI Registers

SCI registers described in Chapter 8.6 can be found here between 0xC000..0xC00F. In addition to these registers, there is one in address 0xC010, called SPI_CHANGE.

SPI registers, prefix SPI_				
Reg	**Type**	**Reset**	**Abbrev[bits]**	**Description**
0xC010	r	0	CHANGE[5:0]	Last SCI access address.

SPI_CHANGE bits		
Name	**Bits**	**Description**
SPI_CH_WRITE	4	1 if last access was a write cycle.
SPI_CH_ADDR	3:0	SPI address of last access.

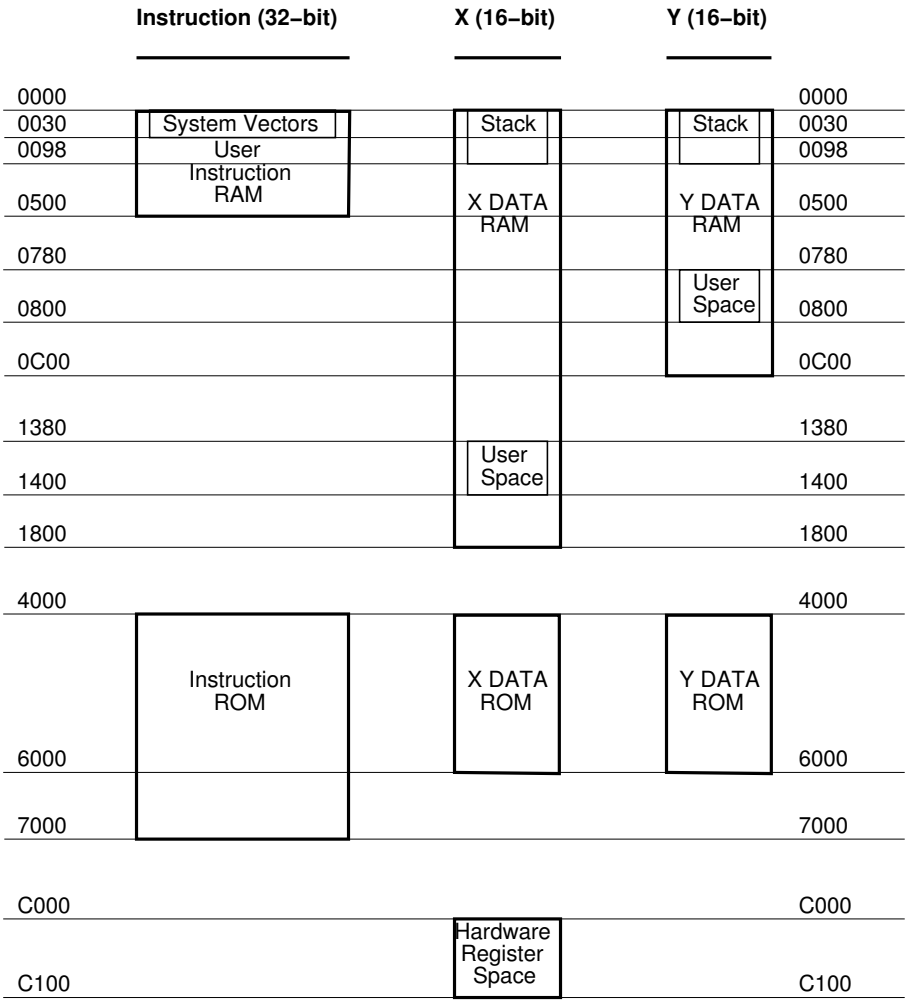


Figure 15: User's Memory Map.

10.5 Serial Data Registers

SDI registers, prefix SER_				
Reg	**Type**	**Reset**	**Abbrev[bits]**	**Description**
0xC011	r	0	DATA	Last received 2 bytes, big-endian.
0xC012	w	0	DREQ[0]	DREQ pin control.

10.6 DAC Registers

DAC registers, prefix DAC_				
Reg	**Type**	**Reset**	**Abbrev[bits]**	**Description**
0xC013	rw	0	FCTLL	DAC frequency control, 16 LSbs.
0xC014	rw	0	FCTLH[4:0]	Clock doubler + DAC frequency control MSbs.
0xC015	rw	0	LEFT	DAC left channel PCM value.
0xC016	rw	0	RIGHT	DAC right channel PCM value.

Every fourth clock cycle an internal 26-bit counter is added to by (DAC_FCTLH & 15) $\times$ 65536 + DAC_FCTLL. Whenever this counter overflows, values from DAC_LEFT and DAC_RIGHT are read and a DAC interrupt is generated.

If DAC_FCTL[4] is 1, the internal clock doubler is activated.

10.7 GPIO Registers

GPIO registers, prefix GPIO_				
Reg	**Type**	**Reset**	**Abbrev[bits]**	**Description**
0xC017	rw	0	DDR[3:0]	Direction.
0xC018	r	0	IDATA[3:0]	Values read from the pins.
0xC019	rw	0	ODATA[3:0]	Values set to the pins.

GPIO_DIR is used to set the direction of the GPIO pins. 1 means output. GPIO_ODATA remembers its values even if a GPIO_DIR bit is set to input.

GPIO registers don't generate interrupts.

10.8 Interrupt Registers

Interrupt registers, prefix INT_				
Reg	**Type**	**Reset**	**Abbrev[bits]**	**Description**
0xC01A	rw	0	ENABLE[2:0]	Interrupt enable.
0xC01B	w	0	GLOB_DIS[-]	Write to add to interrupt counter.
0xC01C	w	0	GLOB_ENA[-]	Write to subtract from interript counter.
0xC01D	rw	0	COUNTER[4:0]	Interrupt counter.

INT_ENABLE controls the interrupts. The control bits are as follows:

INT_ENABLE bits		
Name	**Bits**	**Description**
INT_EN_SDI	2	Enable Data interrupt.
INT_EN_SCI	1	Enable SCI interrupt.
INT_EN_DAC	0	Enable DAC interrupt.

Note: It may take up to 6 clock cycles before changing INT_ENABLE has any effect.

Writing any value to INT_GLOB_DIS adds one to the interrupt counter INT_COUNTER and effectively disables all interrupts. It may take up to 6 clock cycles before writing to this register has any effect.

Writing any value to INT_GLOB_ENA subtracts one from the interrupt counter (unless INT_COUNTER already was 0). 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 the responsibility of the user. It may take up to 6 clock cycles before writing this register has any effect.

By reading INT_COUNTER the user may check if the interrupt counter is correct or not. If the register is not 0, interrupts are disabled.

10.9 System Vector Tags

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

10.9.1 AudioInt, 0x20

Normally contains the following VS_DSP assembly code:

```
jmpi DAC_INT_ADDRESS,(i6)+1
```

The user may, at will, replace the instruction with a $jmpi$ command to gain control over the audio interrupt.

10.9.2 SciInt, 0x21

Normally contains the following VS_DSP assembly code:

```
jmpi SCI_INT_ADDRESS,(i6)+1
```

The user may, at will, replace the instruction with a $jmpi$ command to gain control over the SCI interrupt.

10.9.3 DataInt, 0x22

Normally contains the following VS_DSP assembly code:

```
jmpi SDI_INT_ADDRESS,(i6)+1
```

The user may, at will, replace the instruction with a $jmpi$ command to gain control over the SDI interrupt.

10.9.4 UserCodec, 0x0

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 j 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 insert user programs.

10.10 System Vector Functions

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

10.10.1 WriteIRam(), 0x2

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.

10.10.2 ReadIRam(), 0x4

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.

10.10.3 DataBytes(), 0x6

VS_DSP C prototype:

u_int16 DataBytes(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 bytes that can be read.

10.10.4 GetDataByte(), 0x8

VS_DSP C prototype:

u_int16 GetDataByte(void);

Reads and returns one data byte from the Data Interface. This function will wait until there is enough data in the input buffer.

10.10.5 GetDataWords(), 0xa

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. This function will wait until there is enough data in the input buffer.

11 VS1011 Version Changes

This chapter describes changes between different generations of VS1011.

11.1 Changes Between VS1011b and VS1011e, 2005-07-13

- Faster decoding: all codecs, bitrates and bass + treble controls can be used at CLKI = 24 MHz.
- Register SCI_BASS now also has a treble control (Chapter 8.6.3). **Loudness plugin not required.**
- Can play IMA ADPCM in mono and stereo (Chapter 8.2.4).
- Register space can now be written to with SCI_WRAM (Chapter 8.6.7).
- Memory and register space can now be read from with SCI_WRAM (Chapter 8.6.7).
- Added optional playback of MPEG 1 and 2 layers I and II (MP1 and MP2) (Chapters 8.2.2, 8.2.1 and 8.6.1).
- SPI Boot added (Chapter 9.4).
- MPEG 1, 2 and 2.5 layer III (MP3) decoding more robust against bit errors.
- MPEG 2.5 decoding compatibility enhanced.
- DREQ goes down during SCI operations. (Chapter 7.4).
- DREQ goes down during memory test. (Chapter 7.4).
- In VS1011e the SS_VER field in SCI_STATUS is 2.
- Also SOIC-28 is now a lead-free RoHS-compliant package.

11.2 Migration Checklist from VS1011b to VS1011e, 2005-07-13

- The SS_VER field in SCI_STATUS is 2. You can use SCI_STATUS and SCI_MODE to distinguish between VS1002 and VS1011e. VS1011b has SS_VER=1, SM_SDINEW=0, VS1011e has SS_VER=2, SM_SDINEW=0, and VS1002 has SS_VER=2, SM_SDINEW=1.
- Use built-in bass enhancer and treble control instead of the Loudness plugin. The loudness plugin works, but the builtin controls are much faster.

12 Document Version Changes

This chapter describes the latest and most important changes to this document.

Version 1.23, 2017-09-08

- Removed MP2 and MP3 license descriptions from Chapter 1, *Licenses*, and other relevant places, as all their patents have expired.
- Removed mentions of BGA-49 package.
- Other minor updates.

Version 1.22, 2014-12-19

- Updated telephone number in Chapter 13, *Contact Information*.

Version 1.21 for VS1011e, 2014-12-08

Updated Figure 4, *Typical Connection Diagram Using LQFP-48*.

Version 1.20 for VS1011e, 2014-02-25

Specification change affected recommended operating conditions if clock doubler is used in Chapter 4.2, *Recommended Operating Conditions*.

Version 1.10 for VS1011e, 2012-02-29

- SCI Test description fixed in Chapter 9.7.4.
- Connection diagram in Chapter 6 updated for improved ESD protection (20Ω GBUF resistor replaced with 10Ω).
- Simplified drawing of LQFP48 package dimensions added to Chapter 5.1.1.
- Added SPI timing diagram to Chapter 7.2.1 for newmode SDI.

Version 1.05 for VS1011e, 2009-10-06

- SPI timing clarification (CLKI vs. XTALI) in section 7.6.
- Added CLOCKF example for 12.288 MHz clock to section 8.6.4.

Version 1.04 for VS1011e, 2007-10-08

- Starting from VS1011e also SOIC-28 is a RoHS-compliant lead-free package.

13 Contact Information



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

URL: http://www.vlsi.fi/
Phone: +358-50-462-3200
Commercial e-mail: sales@vlsi.fi

For technical support or suggestions regarding this document, please participate at
http://www.vsdsp-forum.com/
For confidential technical discussions, contact
support@vlsi.fi