VS1010 Datasheet

VS1010b - Mp3 Player IC with USB and SD card Interfaces

Analog Hardware Features

- Stereo 16-bit Mems Mic audio interface
- Two 24-bit audio DACs
- Stereo earphone driver for 30 Ω load
- 12-bit ADC, 3-7 external inputs
- Operation from single power supply, three programmable internal regulators

Digital Hardware Features

- 75 MIPS VS_DSP^4 processor core
- 16 KiB program RAM (4 KiWord)
- 64 KiB data RAM (2×16 KiWord)
- USB 2.0 Hi-Speed (480 Mbit/s) Device / Host
- I2S and SPDIF digital audio interfaces
- SD Card interface
- 2 SPI bus interfaces
- 2 UART interfaces
- All digital pins are user configurable for general purpose IO
- Flexible clock selection, default operation from 12.288 MHz
- Internal PLL clock multiplier for digital logic
- RTC
- 128-bit AES hardware decryption

Firmware and VSOS Features

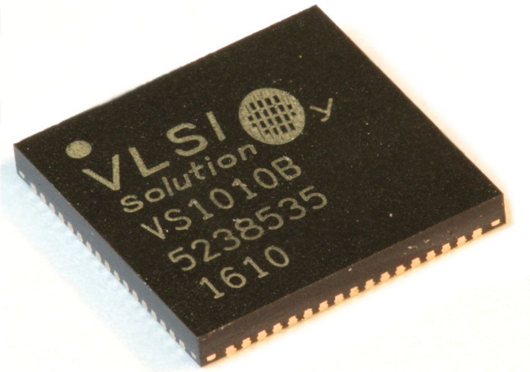
- Decoders: MP3, Ogg Vorbis, WAV PCM
- File I/O for SD cards
- USB host and slave libraries
- Extensive audio DSP library
- Flexible boot options
- Easy-to-write software interface with VSIDE

Applications

- Portable recorders
- Digital docking stations
- MP3 players
- Wireless headphones
- Audio co-processor

Overview

VS1010 is a flexible audio platform device. It is built around VS_DSP^4, which is a powerful DSP (Digital Signal Processor) core. The digital interfaces provide flexible access to external devices in standalone applications and flexible digital audio data inputs and outputs when the device is used as an audio signal processor in more complex systems. The analog interfaces provide high-quality audio outputs, and the control ADC can be used for example for interfacing a resistive touch panel.



Contents

List of Figures

1 Disclaimer

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

2 Licenses

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

Supply of this product does not convey a license nor imply any right to distribute MPEG Layer-3 compliant content created with this product in revenue-generating broadcast systems (terrestrial, satellite, cable and/or other distribution channels), streaming applications (via Internet, intranets and/or other networks), other content distribution systems (pay-audio of audio-on-demand applications and the like) or on physical media (compact discs, digital versatile discs, semiconductior chips, hard drives, memory cards and the like). An independent license for such use is required. For details, please visit http://mp3licensing.com.

3 Definitions

B Byte, 8 bits.

b Bit.

Ki “Kibi” = 2^{10} = 1’024 (IEC 60027-2).

Mi “Mebi” = 2^{20} = 1’048’576 (IEC 60027-2).

Gi “Gibi” = 2^{30} = 1’073’741’824 (IEC 60027-2).

VS_DSP VLSI Solution’s DSP core.

VSOS VLSI Solution’s Operating System

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
Regulator input voltage	VHIGH	-0.3	5.25	V
Analog Positive Supply	AVDD	-0.3	3.6	V
Digital Positive Supply	CVDD	-0.3	1.95	V
Digital RTC Supply	RTCVDD	-0.3	1.95	V
I/O Positive Supply	IOVDD	-0.3	3.6	V
Voltage at Any Digital Input [3]		-0.3	IOVDD+0.3[1]	V
Voltage at power Button	PWRBTN	-0.3	3.6	V
Voltage at XTAL Pins	XTALI, XTALO	-0.3	CVDD+0.3[4]	V
Voltage at RTC Pins	XTALI_RTC, XTALO_RTC	-0.3	CVDD+0.3[4]	V
Total Injected Current on Pins			±200 [2]	mA
Operating Temperature		-40	+85	°C
Storage Temperature		-65	+150	°C

[1] Must not exceed 3.6 V
[2] Latch-up limit
[3] Except RTC and PWRBTN pin
[4] Must not exceed 1.95 V

4.2 Recommended Operating Conditions

Voltage Specification					
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
Operating temperature		-40		+85	°C
Analog and digital ground [1]	AGND DGND		0.0		V
Regulator input voltage[2]	VHIGH	AVDD+0.3	4.0	5.25	V
Analog positive supply[3]	AVDD	2.75	2.8	3.6	V
Digital positive supply [3]	CVDD	1.65	1.8	1.95	V
Digital RTC supply	RTCVDD	1.2	1.5	1.95	V
I/O positive supply[3]	IOVDD	1.8	2.8	3.6	V

[1] Must be connected together as close the device as possible for latch-up immunity.
[2] At least 4.0 V is required for compliant USB level.
[3] Regulator output of the device.

Oscillator Specification					
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
Input clock frequency[1]	XTALI	11	12.288[2]	13	MHz
Input clock duty cycle		40	50	60	%
Oscillator frequency tolerance			+/-10		ppm
Startup time			1		ms
Internal clock frequency, USB connected	CLKU	60		60	MHz
Internal clock frequency, USB disconnected	CLKI			75	MHz
RTC clock frequency[3]	XTALI_RTC		32768		Hz
RTC frequency tolerance			+/-100		ppm
RTC oscillator startup time			1000		ms

[1] The maximum sample rate that can be played with correct speed is XTALI/128. With 11 MHz XTALI sample rates over 85937 Hz are played at 85937 Hz.
[2] When full speed (FS) or Hi-Speed (HS) USB is used it is recommended that XTALI of 12.288 MHz or 12.0 MHz is used. The ROM USB firmware assumes XTALI = 12.288 MHz.
[3] The 32.768 kHz crystal is optional, but required for RTC time counter.

4.3 Analog Characteristics of Audio Outputs

Unless otherwise noted: AVDD=3.6 V, CVDD=1.8 V, IOVDD=2.8 V, V_{ref}=1.6 V, TA=+25°C, XTALI=12 MHz, Internal Clock Multiplier $3.0\times$. DAC tested with full-scale output sinewave, measurement bandwidth 20..20000 Hz, analog output load: LEFT to CBUF 30 Ω, RIGHT to CBUF 30 Ω.

DAC Characteristics					
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
DAC Resolution			24		bits
Dynamic range (DAC unmuted, A-weighted, min gain)	IDR		100		dB
S/N ratio (full scale signal, no load)	SNR		92		dB
S/N ratio (full scale signal, 30 ohm load)	SNRL		90		dB
Total harmonic distortion, -3dB level, no load	THD		0.01		%
Total harmonic distortion, -3dB level, 30 ohm load	THDL		0.05		%
Crosstalk (L/R to R/L), 30 ohm load, without CBUF [1]	XTALK1		-75		dB
Crosstalk (L/R to R/L), 30 ohm load, with CBUF	XTALK2		-54		dB
Gain mismatch (L/R to R/L)	GERR	-0.5		0.5	dB
Frequency response	AERR	-0.05		0.05	dB
Full scale output voltage	LEVEL		1.0		Vrms
Deviation from linear phase	PH		0	5	°
Analog output load resistance	AOLR		30[2]		Ω
Analog output load capacitance	AOLC			100[3]	pF
DC level, V_{ref}=1.2 V (CBUF, LEFT, RIGHT)		1.1		1.3	V
DC level, V_{ref}=1.6 V (CBUF, LEFT, RIGHT)		1.5		1.7	V
CBUF disconnect current (short-circuit protection)			130	200	mA

[1] Loaded from Left/Right pin to analog ground via 100 μF capacitors.

[2] AOLR may be lower than *Typical*, but distortion performance may be compromised. Also, there is a maximum current that the internal regulators can provide.

[3] CBUF must have external 10 Ω + 47 nF load, LEFT and RIGHT must have external 20 Ω + 10 nF load for optimum stability and ESD tolerance.

4.4 SAR Characteristics

SAR Characteristics					
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
SAR resolution			12		bits
Input amplitude range		0		AVDD	V
SAR sample rate [1]				100	kHz
Integral Nonlinearity	INL		+/-2		LSB
Differential Nonlinearity	DNL		+/-0.5		LSB

[1] XTALI dependent

4.5 Analog Characteristics of Regulators

Parameter	Symbol	Min	Typ	Max	Unit
IOVDD					
Recommended voltage setting range		1.7		3.6	V
Voltage setting step size		55	60	65	mV
Default setting, reset mode [1]			1.8		V
Default setting, active mode [2]			3.6		V
Load regulation			4.0		mV/mA
Line regulation from VHIGH			2.0		mV/V
Continuous current			30[4]	60	mA
CVDD					
Recommended voltage setting range		1.65		1.95	V
Voltage setting step size		25	30	35	mV
Default setting, reset mode [1]			1.8		V
Default setting, active mode [2]			1.8		V
Continuous current			25[4]	70	mA
Load regulation			2.0		mV/mA
Line regulation from VHIGH			2.0		mV/V
AVDD					
Recommended voltage setting range		2.6		3.6	V
Voltage setting step size		35	40	45	mV
Default setting, reset mode [1]			2.5		V
Default setting, active mode [2]			2.7		V
Continuous current			30[4]	70	mA
Load regulation			1.5		mV/mA
Line regulation from VHIGH			2.0		mV/V
PWRBTN					
Minimum startup voltage			0.9		V
Minimum startup pulse			100		ms

[1] Device enters reset mode when XRESET pin is pulled low.
[2] Device enters active mode when XRESET pin is pulled high after reset mode. [4] Device is tested with a 30 mA load.

4.6 Analog Characteristics of USB_VDD input

Parameter	Symbol	Min	Typ	Max	Unit
Input voltage		VHIGH+0.5	5.0	5.5	V
Charge cut off voltage			4.20		V
Ihold			0.080		mA
Ioff			0		mA
Icharge1			50		mA
Icharge2			65		mA
Icharge3			75		mA

4.7 Analog Characteristics of VHIGH voltage monitor

Parameter	Symbol	Min	Typ	Max	Unit
Trigger voltage	AMON		1.07×AVDD		V
Hysteresis			50		mV

4.8 Analog Characteristics of CVDD voltage monitor

Parameter	Symbol	Min	Typ	Max	Unit
Trigger voltage	CMON	1.40	1.45		V
Hysteresis			2		mV

4.9 Power Button Characteristics

Unless otherwise noted: VHIGH = 4.0..5.3 V

Parameter	Symbol	Min	Typ	Max	Unit
Power button activation threshold	PBTHR		1.0		V

4.10 Digital Characteristics

Parameter	Symbol	Min	Typ	Max	Unit
High-Level Input Voltage		$0.7\times$IOVDD		IOVDD+0.3	V
Low-Level Input Voltage		-0.2		$0.3\times$IOVDD	V
High-Level Output Voltage, -1.0 mA load [1]		$0.7\times$IOVDD			V
Low-Level Output Voltage, 1.0 mA load [1]				$0.3\times$IOVDD	V
XTALO high-level output voltage, -0.1 mA load		$0.7\times$CVDD			V
XTALO low-level output voltage, 0.1 mA load				$0.3\times$CVDD	V
Input leakage current		-1.0		1.0	μA
Rise time of all output pins, load = 30 pF [1]				50	ns

[1] LQFP pins GPIO0_[8:0], GPIO1_[14:0]. LFGA pins GPIO0_[10:0], GPIO1_[14:0], GPIO2_[15:0].

4.11 Power Consumption

4.11.1 Digital Power Consumption

The following power consumptions are, unless otherwise noted, obtained with the following parameters: decoding 128 kbit/s 44.1 kHz stereo MP3 from RAM memory to analog output, CVDD = 1.67 V, AVDD = 3.6 V, IOVDD = 3.3 V, V_{ref} = 1.6 V, XTALI = 12.288 MHz.

Digital Current Consumption from CVDD, MP3 decode					
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
Firmware default setup after startup, CLKI = 60.000 MHz	ID60MP3		24.4		mA
Using PLL clock instead of RF clock, CLKI = 61.440 MHz			16.5		mA
After powering down unused peripherals, CLKI = 61.440 MHz	ID61MP3		13.4		mA
Setting CLKI = 36.684 MHz[1]	ID36MP3		12.0		mA
Setting CLKI = 24.576 MHz	ID24MP3		11.1		mA
Decode 96 kbit/s 16 kHz stereo MP3, CLKI = 12.288 MHz	ID12MP3		7.4		mA
Decode 56 kbit/s 16 kHz mono MP3, CLKI = 6.144 MHz	ID06MP3		3.8		mA
Check for Key push using GPIO, CLKI = 12.000 kHz	ID12KHZ		0.1		mA

[1] This clock is enough to decode all MP3 streams with some to spare.

The following table shows the digital power consumption when the processor is running but sitting idle $>$95 % of the time.

Digital Current Consumption from CVDD, Processor Idle					
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
CLKI = 61.440 MHz	ID61IDLE		7.1		mA
CLKI = 24.576 MHz	ID24IDLE		4.6		mA
CLKI = 12.288 MHz	ID12IDLE		3.0		mA
CLKI = 6.144 MHz	ID06IDLE		1.6		mA

4.11.2 Analog Power Consumption

The following power consumptions are, unless otherwise noted, obtained with the following parameters: decoding 128 kbit/s 44.1 kHz stereo MP3 from RAM memory to analog output, CVDD = 1.67 V, AVDD = 2.75 V / 3.60 V for V_{ref}=1.2 V / 1.6 V, respectively, IOVDD = 3.3 V, XTALI = 12.288 MHz, CLKI = 24.576 MHz.

Typical Analog Current Consumption from AVDD						
		V_{ref} = 1.2 V		**V_{ref} = 1.6 V**		
		30Ω	**10 kΩ**	**30Ω**	**10 kΩ**	
Parameter	**Symbol**	**Typ**	**Typ**	**Typ**	**Typ**	**Unit**
Full-scale 1 kHz sine wave, full volume[1]	IAFSxxVyy[2]	42.0	5.4	57.4	8.7	mA
Loud music, full volume	IA0DBxxVyy[2]	11.2	5.3	15.6	8.1	mA
Loud music, -20 dB volume	IA20DBxxVyy[2]	5.6	5.3	8.3	8.0	mA
Silence	IASILxxVyy[2]	5.4	5.3	8.1	8.0	mA
Mute (analog drivers off)	IAMxxVyy[2]	1.8	1.8	2.5	2.5	mA

[1] Output signal approximately 660 mVrms for V_{ref}=1.2 V, and 900 mVrms for V_{ref}=1.6 V.

[2] Replace xx with 12 for V_{ref}=1.2 V and 16 for V_{ref}=1.6 V. Replace yy with 30 for 30 Ω load, and with HI for 10 kΩ load.

4.11.3 I/O Power Consumption

The following power consumptions are, unless otherwise noted, obtained with the following parameters: decoding 128 kbit/s 44.1 kHz stereo MP3 from RAM memory to analog output, CVDD = 1.67 V, AVDD = 3.6 V, XTALI = 12.288 MHz, no specific I/O activity.

Digital Current Consumption from CVDD, MP3 decode					
Parameter	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
IOVDD = 3.6 V	IIO36		1.51		mA
IOVDD = 3.3 V	IIO33		1.20		mA
IOVDD = 2.7 V	IIO27		0.85		mA
IOVDD = 1.8 V	IIO18		0.46		mA

4.11.4 Example Power Consumption

Let's assume a system with an earphone output and audio playback capability. Let's further assume that the system could be run at CVDD = 1.67 V, AVDD = 2.70 V (V_{ref} = 1.2 V), IOVDD = 3.3 V.

The VS1010 typical power consumption decoding a 128 kbit/s MP3 stream to 30 Ω earphones, would be approximately:
I_{tot} = ID36MP3 + IA20DB12V30 + IIO33 = 12.0 mA + 5.6 mA + 1.20 mA = 18.8 mA.

This figure needs to be rounded slightly up because the digital current figures don't include reading the file from external memory, or a user interface. Note that the figures assume that all VS1010 peripherals that are not being used have been powered down or their clock gates have been closed (see registers CLK_CF and REGU_CF). Note also that the external memory used for playback, e.g. an SD card, can often consume significant amounts of current.

5 Package and Pin Descriptions

5.1 LQFP-48 Package

LQFP-48 is lead (Pb) free and RoHS-compliant package. 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*.

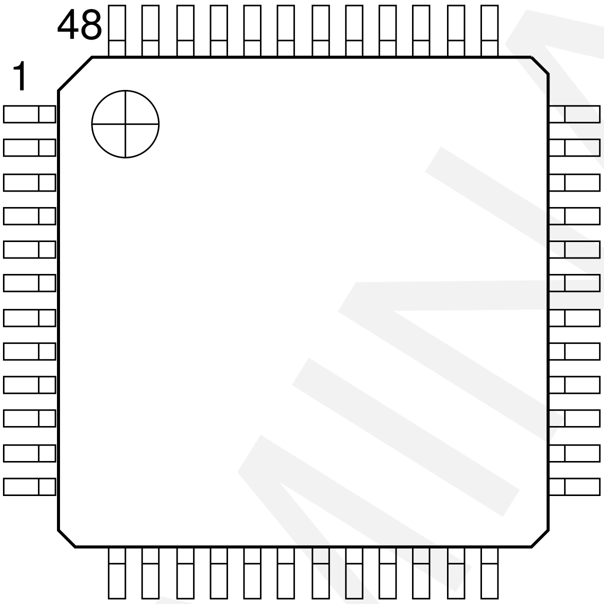


Figure 1: VS1010 pin configuration, LQFP-48.

LQFP-48 package dimensions are at *http://www.vlsi.fi/* .

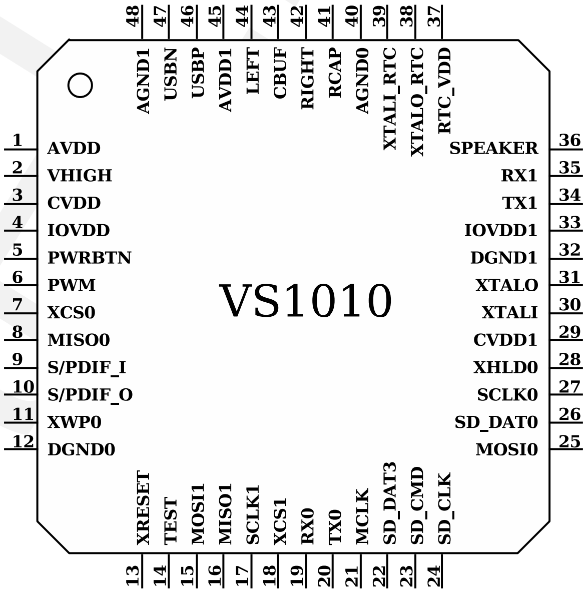


Figure 2: VS1010 pins, LQFP-48.

5.2 VS1010 LQFP-48 Pin Descriptions

Pin Name	LQFP-48 Pin	Pin Type	Function
AVDD	1	APWR	Analog power supply, Regulator output
VHIGH	2	PWR	Power supply, Regulator input
CVDD	3	CPWR	Core power supply, Regulator output
IOVDD	4	IOPWR	I/O power supply, Regulator output
PWRBTN	5	APB / DI	Power button for Regulator startup and Power Key
PWM / GPIO0_0	6	DIO	Pulse width modulation output / General-purpose IO Port 0, bit 0
XCS0 / GPIO1_3	7	DIO	SPI0 XCS / General-Purpose I/O Port 1, bit 3
MISO0 / GPIO1_1	8	DIO	SPI0 MISO / General-Purpose I/O Port 1, bit 1
S/PDIF_I / GPIO0_1	9	DIO	S/PDIF input / General-purpose IO Port 0, bit 1
S/PDIF_O / GPIO1_14	10	DIO	S/PDIF output / General-purpose IO Port 0, bit 14
XWP0 / GPIO1_13	11	DIO	SPI0 Write Protect / General-Purpose I/O Port 1, bit 13
DGND0	12	DGND	Core & I/O ground
XRESET	13	DI	Active low asynchronous reset, schmitt-trigger input
TEST	14	DI	Test mode input (active high), connect to DGND
MOSI1 / GPIO1_4	15	DIO	SPI1 MOSI / General-Purpose I/O Port 1, bit 4
MISO1 / GPIO1_5	16	DIO	SPI1 MISO / General-Purpose I/O Port 1, bit 5
SCLK1 / GPIO1_6	17	DIO	SPI1 CLK / General-Purpose I/O Port 1, bit 6
XCS1 / GPIO1_7	18	DIO	SPI1 XCS / General-Purpose I/O Port 1, bit 7
RX0 / GPIO1_8	19	DIO	UART0 RX / General-Purpose I/O Port 1, bit 8
TX0 / GPIO1_9	20	DIO	UART0 TX / General-Purpose I/O Port 1, bit 9
MCLK / GPIO0_2	21	DIO	Core clock output / General-purpose IO Port 0, bit 2
SD_DAT3 / GPIO0_3	22	DIO	SD Card data IO3 / General-purpose IO Port 0, bit 3
SD_CMD / GPIO0_4	23	DIO	SD Card command IO / General-purpose IO Port 0, bit 4
SD_CLK / GPIO0_5	24	DIO	SD Card clock / General-purpose IO Port 0, bit 5
MOSI0 / GPIO1_0	25	DIO	SPI0 MOSI / General-Purpose I/O Port 1, bit 0
SD_DAT0 / GPIO0_6	26	DIO	SD Card data IO0 / General-purpose IO Port 0, bit 6
SCLK0 / GPIO1_2	27	DIO	SPI0 CLK / General-Purpose I/O Port 1, bit 2
XHLD0 / GPIO1_12	28	DIO	SPI0 Hold / General-Purpose I/O Port 1, bit 12
CVDD1	29	CPWR	Core power supply, connect to regulator CPWR
XTALI	30	AI	Crystal input
XTALO	31	AO	Crystal output
DGND1	32	DGND	Core & I/O ground
IOVDD1	33	IOPWR	I/O power supply
TX1 / GPIO1_10	34	DIO	UART1 TX / General-Purpose I/O Port 1, bit 10
RX1 / GPIO1_11	35	DIO	UART1 RX / General-Purpose I/O Port 1, bit 11
SPEAKER / GPIO0_7	36	DIO	DAC SDM output / General-Purpose I/O Port 0, bit 7
RTC_VDD	37	RPWR	RTC power supply, 1.8V
XTALO_RTC	38	AO	RTC Crystal output
XTALI_RTC	39	AI	RTC Crystal input
AGND0	40	APWR	Analog ground
RCAP	41	AIO	Filtering capacitance for reference
RIGHT	42	AO	Right channel output
CBUF	43	AO	Common voltage buffer for headphones (1.2V nominal)
LEFT	44	AO	Left channel output
AVDD1	45	APWR	Analog power supply
USBP	46	AIO	USB differential + in / out, controllable 1.5kΩ pull-up
USBN	47	AIO	USB differential - in / out
AGND1	48	APWR	Analog ground

Some pins have a secondary peripheral function:

Pin Name	LQFP-48 Pin	Pin Type	Secondary Function
S/PDIF_I / GPIO0_1	9	DIO	MEMS Mic data / I2S data input
S/PDIF_O / GPIO1_14	10	DIO	MEMS Mic clock / I2S data output
XWP0 / GPIO1_13	11	DIO	I2S clock
XHLD0 / GPIO1_12	28	DIO	I2S frame sync
TX1 / GPIO1_10	34	DIO	SAR input 0
RX1 / GPIO1_11	35	DIO	SAR input 1
SPEAKER / GPIO0_7	36	DIO	SAR input 2

Pin types:

Type	Description
DI	Digital input, CMOS input pad
DO	Digital output, CMOS input pad
DIO	Digital input/output
AI	Analog input
AO	Analog output
AIO	Analog input/output
APWR	Analog power supply pin
APB	Analog power button pin
GND	Core or I/O ground pin
CPWR	Core power supply pin
IOPWR	I/O power supply pin
RPWR	RTC power supply pin

5.3 LFGA-68 Package

LFGA-68 is a 8x8x0.8 mm, lead (Pb) free and RoHS-compliant package. 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*.

LFGA package and pin dimensions are shown in Figure 3. For more information about the LFGA-68 package and its dimensions visit *http://www.vlsi.fi/* .

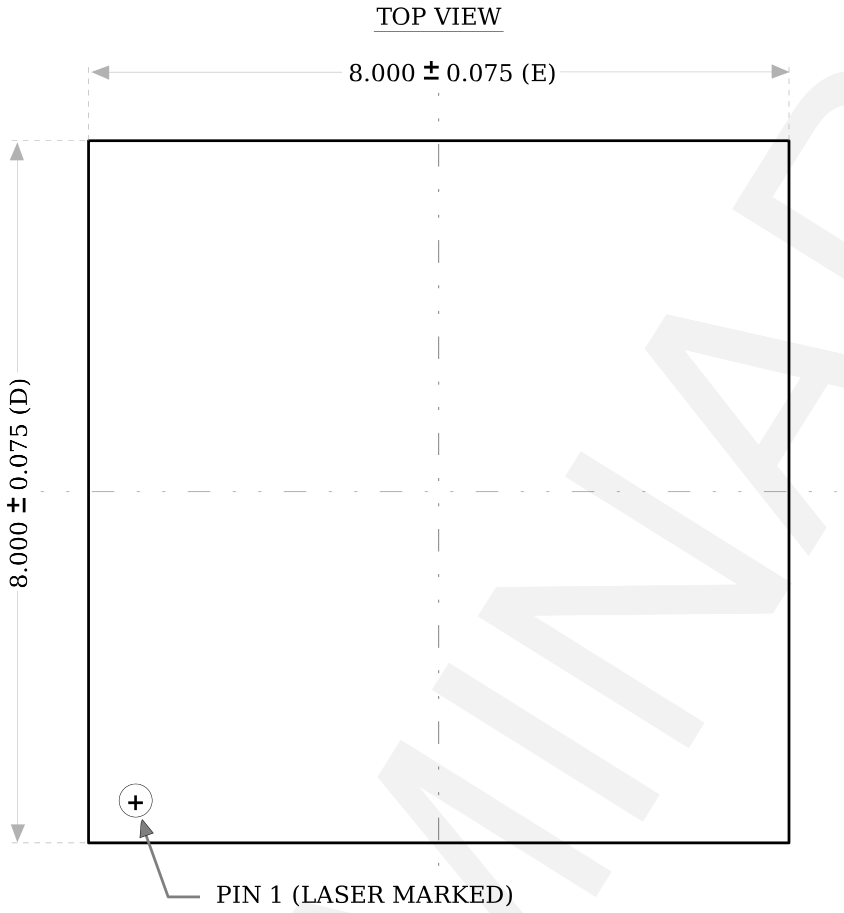


Figure 3: VS1010 top view, LFGA-68

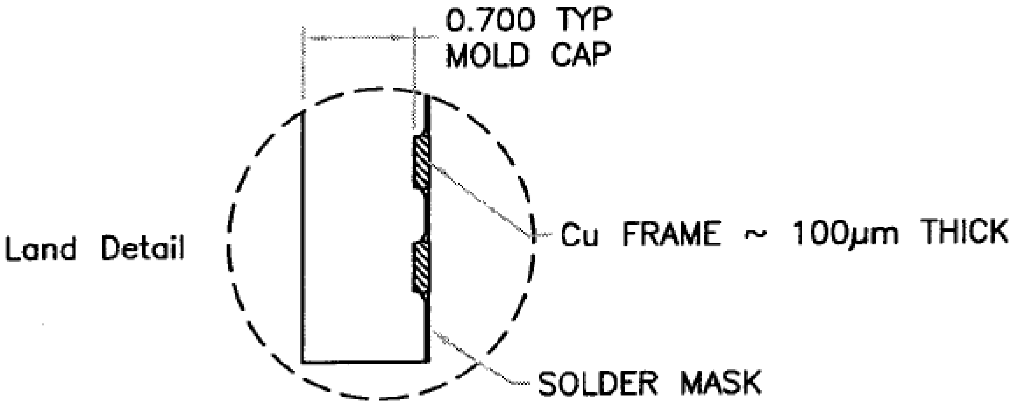


Figure 4: VS1010 corner view, LFGA-68

5.4 LFGA-68 Pin Assignments

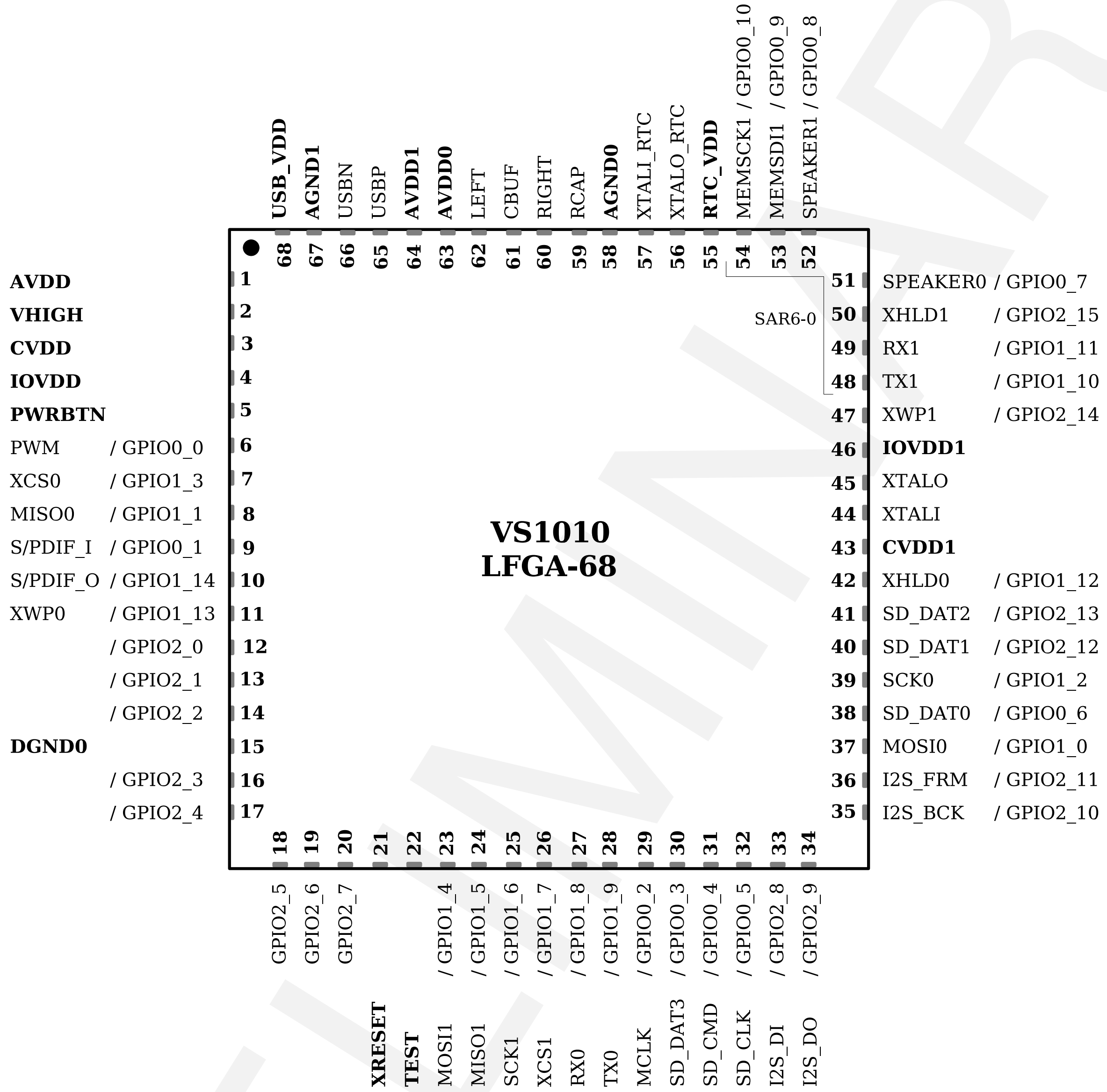


Figure 5: VS1010 68-pin LFGA pin assignment.

5.5 VS1010 LFGA-68 Pin Descriptions

Pin Name	LFGA Pin	Pin Type	Function
GPLATE	(0)	PWR	Center ground plate, use multiple vias to create low-impedance connection to ground network on PCB!

Left Pin Name	LFGA Pin	Pin Type	Function
AVDD	1	APWR	Analog power supply, Regulator output
VHIGH	2	PWR	Power supply, Regulator input
CVDD	3	CPWR	Core power supply, Regulator output
IOVDD	4	IOPWR	I/O power supply, Regulator output
PWRBTN	5	AIO	Power button for Regulator startup (and Power Key)
PWM / GPIO0_0	6	DIO	PWM output / General-Purpose I/O Port 0, bit 0
XCS0 / GPIO1_3	7	DIO	SPI0 XCS / General-Purpose I/O Port 1, bit 3
MISO0 / GPIO1_1	8	DIO	SPI0 MISO / General-Purpose I/O Port 1, bit 1
S/PDIF_I / GPIO0_1	9	DIO	S/PDIF input / General-Purpose I/O Port 0, bit 1
S/PDIF_O / GPIO1_14	10	DIO	S/PDIF output / General-Purpose I/O Port 1, bit 14
XWP0 / GPIO1_13	11	DIO	SPI0 XWP / General-Purpose I/O Port 1, bit 13
GPIO2_0	12	DIO	General-Purpose I/O Port 2, bit 0
GPIO2_1	13	DIO	General-Purpose I/O Port 2, bit 1
GPIO2_2	14	DIO	General-Purpose I/O Port 2, bit 2
DGND0	15	CPWR	Core ground, connect to GPLATE
GPIO2_3	16	DIO	General-Purpose I/O Port 2, bit 3
GPIO2_4	17	DIO	General-Purpose I/O Port 2, bit 4

Bottom Pin Name	LFGA Pin	Pin Type	Function
GPIO2_5	18	DIO	General-Purpose I/O Port 2, bit 5
GPIO2_6	19	DIO	General-Purpose I/O Port 2, bit 6
GPIO2_7	20	DIO	General-Purpose I/O Port 2, bit 7
XRESET	21	DI	Active low asynchronous reset, schmitt-trigger input
TEST	22	DI	Test mode input (active high), connect to ground
MOSI1 / GPIO1_4	23	DIO	SPI1 M0SI / General-Purpose I/O Port 1, bit 4
MISO1 / GPIO1_5	24	DIO	SPI1 MISO / General-Purpose I/O Port 1, bit 5
SCLK1 / GPIO1_6	25	DIO	SPI1 CLK / General-Purpose I/O Port 1, bit 6
XCS1 / GPIO1_7	26	DIO	SPI1 XCS / General-Purpose I/O Port 1, bit 7
RX0 / GPIO1_8	27	DIO	UART0 RX / General-Purpose I/O Port 1, bit 8 (connect with 100 kΩ to IOVDD if not used for UART)
TX0 / GPIO1_9	28	DIO	UART0 TX / General-Purpose I/O Port 1, bit 9
MCLK / GPIO0_2	29	DIO	Master clock output / General-Purpose I/O Port 0, bit 2
SD_DAT3 / GPIO0_3	30	DIO	SD card data line 3 / General-Purpose I/O Port 0, bit 3
SD_CMD / GPIO0_4	31	DIO	SD card cmd line / General-Purpose I/O Port 0, bit 4
SD_CLK / GPIO0_5	32	DIO	SD card clock / General-Purpose I/O Port 0, bit 5
I2S_DI / GPIO2_8	33	DIO	I2S data in / General-Purpose I/O Port 2, bit 8
I2S_DO / GPIO2_9	34	DIO	I2S data out / General-Purpose I/O Port 2, bit 9

Right Pin Name	LFGA Pin	Pin Type	Function
I2S_BCK / GPIO2_10	35	DIO	I2S bit clock / General-Purpose I/O Port 2, bit 10
I2S_FRM / GPIO2_11	36	DIO	I2S frame sync / General-Purpose I/O Port 2, bit 11
MOSI0 / GPIO1_0	37	DIO	SPI0 MOSI / General-Purpose I/O Port 1, bit 0
SD_DAT0 / GPIO0_6	38	DIO	SD card data line 0 / General-Purpose I/O Port 0, bit 6
SCLK0 / GPIO1_2	39	DIO	SPI0 CLK / General-Purpose I/O Port 1, bit 2
SD_DAT1 / GPIO2_12	40	DIO	SD card data line 1 / General-Purpose I/O Port 2, bit 12
SD_DAT2 / GPIO2_13	41	DIO	SD card data line 2 / General-Purpose I/O Port 2, bit 13
XHLD0 / GPIO1_12	42	DIO	SPI0 XWP / General-Purpose I/O Port 1, bit 12
CVDD1	43	CPWR	Core power supply, connect to regulator CPWR
XTALI	44	AI	Crystal input
XTALO	45	AO	Crystal output
IOVDD1	46	IOPWR	I/O power supply, connect to regulator IOPWR
XWP1 / GPIO2_14	47	DIO	SPI1 XWP / General-Purpose I/O Port 2, bit 14
TX1 / GPIO1_10	48	DIO	UART1 TX / General-Purpose I/O Port 1, bit 10
RX1 / GPIO1_11	49	DIO	UART1 RX / General-Purpose I/O Port 1, bit 11
XHLD1 / GPIO2_15	50	DIO	SPI1 XWP / General-Purpose I/O Port 2, bit 15
SPEAKER0 / GPIO0_7	51	DIO	DAC left output / General-Purpose I/O Port 0, bit 7

Top Pin Name	LFGA Pin	Pin Type	Function
SPEAKER1 / GPIO0_8	52	DIO	DAC right output / General-Purpose I/O Port 0, bit 8
MEMSDI1 / GPIO0_9	53	DIO	Mems Mic data / General-purpose I/O Port 0, bit 9
MEMSCK1 / GPIO0_10	54	DIO	Mems Mic clock / General-purpose I/O Port 0, bit 10
RTC_VDD	55	RTCPWR	Real time clock power supply
XTALO_RTC	56	AO	Real time clock crystal output
XTALI_RTC	57	AI	Real time clock crystal input
AGND0	58	APWR	Analog reference ground, connect to both GPLATE and RCAP capacitor without vias in PCB
RCAP	59	AIO	Filtering capacitance for reference
RIGHT	60	AO	Right channel output
CBUF	61	AO	Common voltage buffer for headphones
LEFT	62	AO	Left channel output
AVDD0	63	APWR	Analog power supply, connect to regulator APWR
AVDD1	64	APWR	Analog power supply, connect to regulator APWR
USBP	65	AIO	USB differential + in / out, controllable 1.5kΩ pull-up
USBN	66	AIO	USB differential - in / out
AGND1	67	APWR	USB ground, connect to ground network in PCB and GPLATE
USB_VDD	68	APWR	USB power supply input

Pin type descriptions:

Type	Description
DI	Digital input, CMOS input pad
DIPD	Digital input with weak pull-down resistor (approx. 1 MΩ)
DO	Digital output, CMOS output pad
DIO	Digital input/output
DIOPD	Digital input/output with weak pull-down resistor in input (approx. 1 MΩ)

Type	Description
AI	Analog input
AO	Analog output
AIO	Analog input/output
APWR	Analog power supply pin or ground
APWR1V8	Analog power supply pin, 1.8V
RTCPWR	Real time clock power supply pin, 1.8V
CPWR	Core power supply pin
IOPWR	I/O power supply pin

Package bottom plate is a ground net and it is connected to ground network in PCB.

NOTE: Unused inputs should not be left floating. Tie either HIGH or LOW as appropriate. At power-up all GPIO is three stated and current leakage from IOVDD is cut. Outputs that are three-statable should only be pulled high or low to ensure signals at power-up and in standby.

Some pins have a secondary peripheral function:

Pin Name	**LFGA Pin**	**Pin Type**	**Secondary Function**
S/PDIF_I / GPIO0_1	9	DIO	MEMS Mic data / I2S data input
S/PDIF_O / GPIO1_14	10	DIO	MEMS Mic clock / I2S data output
XWP0 / GPIO1_13	11	DIO	I2S clock
XHLD0 / GPIO1_12	42	DIO	I2S frame sync
TX1 / GPIO1_10	48	DIO	SAR input 0
RX1 / GPIO1_11	49	DIO	SAR input 1
SPEAKER0/ GPIO0_7	51	DIO	SAR input 2
XHLD1/ GPIO2_15	50	DIO	SAR input 3
SPEAKER1/ GPIO0_8	52	DIO	SAR input 4
MEMSCK1 / GPIO0_9	53	DIO	SAR input 5
MEMSDI1 / GPIO0_10	54	DIO	SAR input 6

5.6 PCB Layout Recommendations

The following recommendations should be followed to ensure reliable operation.

- Analog power nets that are connected to regulator APWR/CPWR output should have bypass capacitors.
- USBP and USBN traces should be kept within 2mm of each other and with preferred length of 20-30mm (max 75mm). A solid ground plane is preferred under USBP and USBN traces.
- USBP and USBN traces should be very close to same length, drawn together and their characteristic differential impedance 90 Ohms
- No vias are allowed in USBP or USBN traces, only 45 degree angles should be used.
- USBP and USBN traces should be isolated from all other signal traces.

5.7 Differencies Between LQFP-48 and LFGA-68 Package Options

VS1010 pins that are not connected in LQFP-48 Packgage are AVDD1, USB_VDD, GPIO0[10:8] and GPIO2[15:0]. Therefore the 4-bit mode in SPI1, 4-bit mode in SD card and battery charging from USB are not applicable. I2S peripheral has secondary pins for LQFP-48 package.

VS1010 Pin	Peripheral function	Pin Type	Description
GPIO2[7:0]	NA	DIO	Byte bus
GPIO2_8	I2S_DI	DIO	I2S data input
GPIO2_9	I2S_DO	DIO	I2S data output
GPIO2_10	I2S_BCK	DIO	I2S data bit clock
GPIO2_11	I2S_FRM	DIO	I2S frame sync
GPIO2_12	SD_DAT1	DIO	SD card data line 1
GPIO2_13	SD_DAT2	DIO	SD card data line 2
GPIO2_14	XWP1	DIO	SPI1 XWP (write protect / IO2)
GPIO2_15	XHLD1	DIO	SPI1 XHLD (hold / IO3)
GPIO0_8	SPEAKER1 / sar4	DIO	Sar input 4 / 1-bit DAC right output
GPIO0_9	sar5	DIO	Sar input 5 / memsmic 1 data input
GPIO0_10	sar6	DIO	Sar input 6 / memsmic 1 clock output
AVDD1	NA	APWR	Analog power supply
USB_VDD	Battery charging	APWR	USB power supply input

I2S in LQFP-48 package uses same pins as SPDIF/memsmic0, XWP0 and XHLD0. When I2S is used the 4-bit SPI0 mode, mems mic 0 and SPDIF are not applicable. SPDIF input is applicable when only i2s output is used.

6 Example Schematic

TBD

7 VS1010 General Description

VS1010 architecture is based on VS_DSP core. VS_DSP core architecture is described in VS_DSP User's Manual. Chip is powered with internal regulator which provides voltages for three separate power domains. The core and periphery I/O power domains can be driven off separately, allowing simple I/O interfacing and minimizing power consumption. RTC has its own power supply which enables the RTC usage when the rest of the chip is powered down. RTC also includes a small backup ram. VS1010 has two clock domains which are clocked by PLL. Analog interfaces are clocked with an XTALI clock but the dsp, digital intarfaces and memories are clocked with a multiplied clock. VS1010 external interfaces are shown in Figure 6.

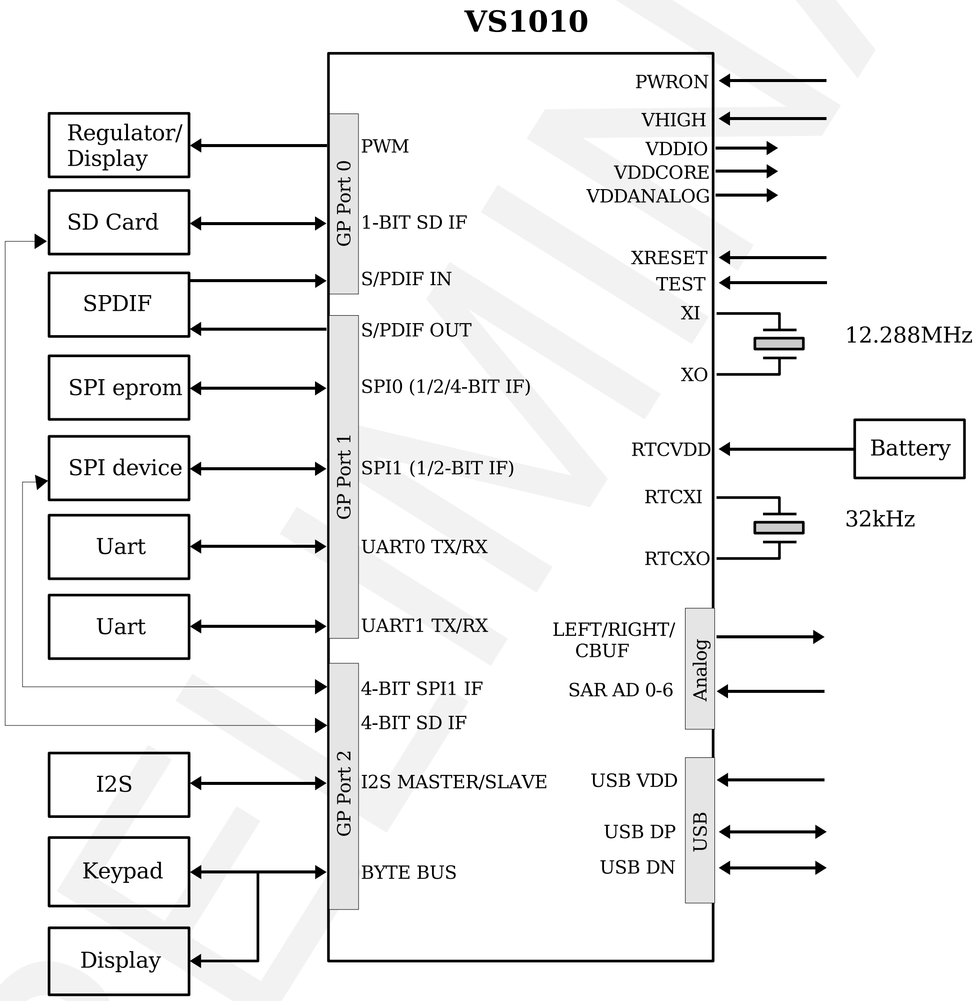


Figure 6: VS1010 external interfaces

7.1 VS1010 Internal Architecture

VS1010 block diagram is shown in Figure 7.

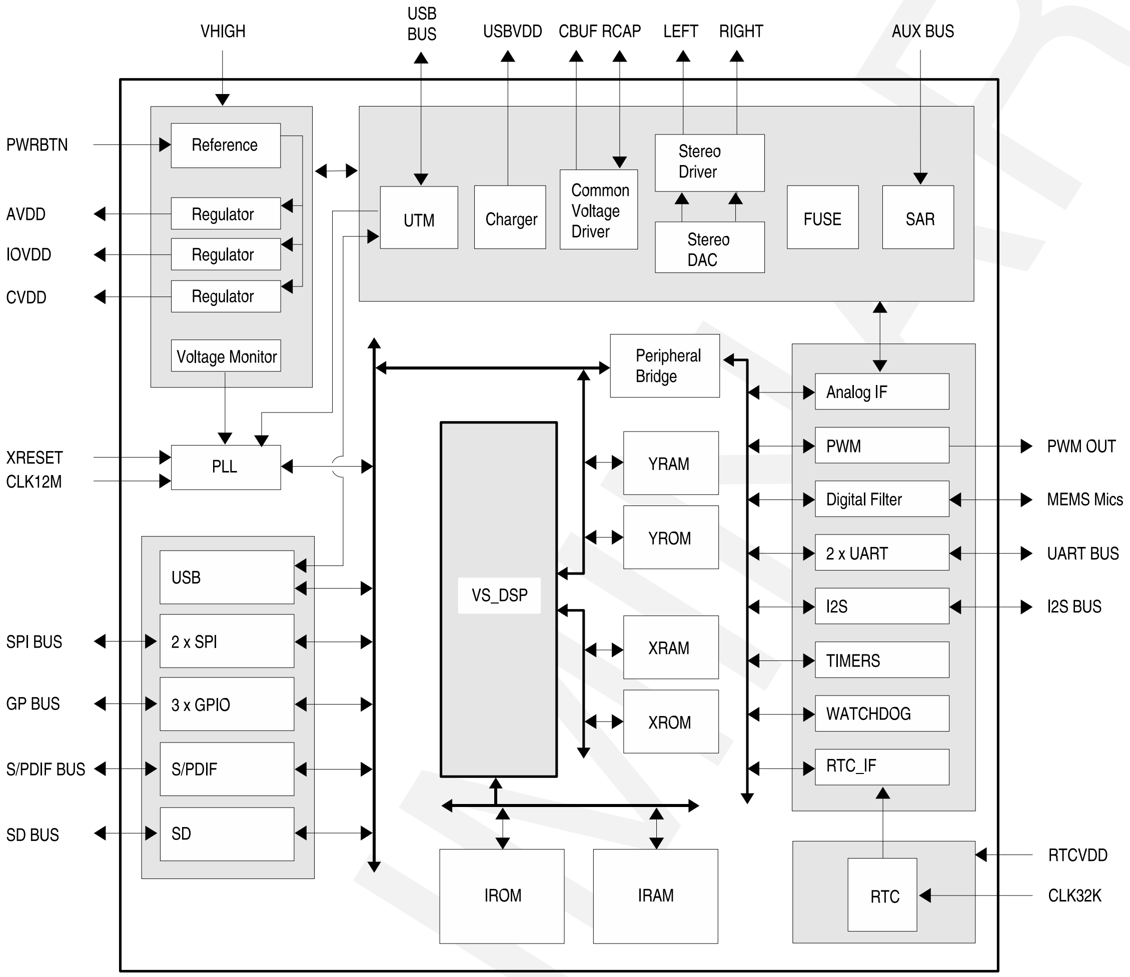


Figure 7: VS1010 block diagram

7.1.1 Regulator Section

The VHIGH pin in the regulator section is used as a common main power supply for voltage regulation. This input is connected to three internal regulators, which are activated when the PWRBTN pin voltage is kept above 0.9 V for about one millisecond, so that AVDD starts to rise and reaches about 1.5 V. After the PWRBTN has given this initial start current, the regulators reach their default voltages even if the PWRBTN is released. VHIGH must be sufficiently (about 0.3 V) above the highest regulated power (normally AVDD) so that regulation can be properly performed.

The PWRBTN state can also be read by software, so it can be used as one of the user interface buttons.

A power-on reset monitors the core voltage and asserts reset if CVDD drops below the CMON level. It is also possible to force a reset by keeping PWRBTN pressed for longer than approximately 5.6 seconds. This feature can be disabled by software. A watchdog counter and the XRESET pin can also generate a reset for the device.

Resets do not cause the regulators to shut down, but they restore the default regulator voltages. After boot the firmware and user software can change the voltages.

Return to power-off is possible only with active software control (VSDSP writes the regulator shutdown bits), or when VHIGH voltage is removed for a sufficiently long time. In the default firmware player the power button has to be pressed for 2 seconds to make the software power-down the system and turn the regulators off.

7.1.2 I/O Section

IOVDD is used for the level-shifters of the digital I/O and crystal oscillator. The regulated IO voltage is internally connected. The IOVDD regulator output must be connected to IOVDD1 input pin. Proper bypass capacitors should also be used.

The firmware uses GPIO1_1 to select I/O voltage level. After reset the I/O voltage is 1.8 V. If GPIO1_1 has a pull-down resistor, 1.8 V I/O voltage is used. If GPIO1_1 has a pull-up resistor, 3.3 V I/O voltage is used.

7.1.3 Digital Section

All digital logic except the real time clock is powered from core voltage CVDD. The regulated core voltage is internally connected. The CVDD regulator output must be connected to CVDD1 input pin. CVDD pins should have proper bypass capacitors.

Real time clock power pin can be connected to CVDD net or it can have its own power supply which enables its use during chip power-down. The inputs and outputs of the RTC logic have level shifters but the RTCVDD voltage should not exceed the CVDD voltage range.

Clock

The crystal amplifier uses a crystal connected to XTALI and XTALO. An external logic-level input clock can also be used. When VS1010 is used with FS USB, 12 MHz crystal allows lower power consumption. With FS/HS USB the input clock of 12 MHz or 12.288 MHz is recommended.

An internal phase-locked loop (PLL) generates the internal clock by multiplying the input clock by 1.0×, 1.5×, . . . , 6.0×. When USB is connected, the clock is 5.0×12 MHz = 60 MHz. When the player is active, the clock will be automatically changed according to the requirements of the song being played.

XRESET disables the clock buffer and puts the digital section into powerdown mode.

In usb suspend state the core clock is switced to RTC clock and the clock oscillator is powered down.

VSDSP^4

VSDSP^4 is VLSI Solution's proprietary digital signal processor with a 32-bit instruction word, two 16-bit data buses, and both 16-bit and 32/40-bit arithmetic.

IROM, XROM, and YROM contain the firmware, including the default player application. Most of the instruction RAM and some of the X and Y data RAM's can be used to customize and extend the functionality of the player.
For software customization the firmware supports nand flash and SD card boot. The VS1010xF version can use also the internal serial flash as a boot device.

UART

An asynchronous serial port is used for debugging and special applications. The default speed is 115200 bps. RX and TX pins can also be used for general-purpose I/O when the UART is not required.

SPIs

A synchronous serial port peripheral is used for SPIEEPROM boot, and can be used to access other SPI peripherals (for example LCD or SED) by using another chip select. The SPI0 is only used for boot if the XCS0 pin has a high level after reset (pull-up resistor attached). These pins can also be used for general-purpose I/O when the SPI is not required.

The default player uses MISO0 and MOSI0 for LED outputs.

SD Card Interface

The SD card interface automates some of the communication with an SD card. Peripheral supports 1-bit and 4-bit data transfers.

The SD card interface pins can also be used as general-purpose I/O.

USB

The USB peripheral handles USB 2.0 Full Speed and Hi-Speed harware protocols. Low speed communication is not supported, but is correctly ignored. The USBP pin has a software-controllable 1.5kΩ pull-up.

A control endpoint (1 IN and 1 OUT) and upto 6 other endpoints (3 IN and 3 OUT) can be used simultaneously. Bulk, interrupt, and isochronous transfer modes are selectable for each endpoint. USB receive from USB host to device (OUT) uses a 2 KiB buffer, thus allowing very high transfer speeds. USB transmit from device to USB host (IN) also uses a 2 KiB buffer and allows all IN endpoints to be ready to transmit simultaneously. Double-buffering is also possible, but not usually required.

7.2 Analog Section

The third regulator provides power for the analog section.

The analog section consists of digital to analog converters and an earphone driver. This includes a buffered common voltage generator (CBUF, around 1.2 V or 1.6 V) that can be used as a virtual ground for headphones.

The regulator AVDD output pin must be connected to AVDD0 and AVDD1 pins with proper bypass capacitors, because they are not connected internally.

The USB pins use the internal AVDD voltage.

AVDD voltage level can be monitored by software. Currently the firmware does not take advantage of this feature.

CBUF contains a short-circuit protection. It disconnects the CBUF driver if pin is shorted to ground. In practise this only happens with external power regulation, because there is a limit to how much power the internal regulators can provide.

8 Oscillator and Reset Configuration

The reset module gathers reset sources and controls the system's internal reset signals. Reset Sources are:

- *POR* : Power-On reset and CVDD voltage monitor
- *XRESET* : External active low reset pin
- *wdog_rst* : Watchdog timer reset
- *PWRBTN* : Power Button reset after 5 seconds

Two clock sources can be used :

- 11 MHz - 13 MHz oscillator (recommended 12.288 MHz)
- 32 kHz RTC oscillator

9 VS1010 Peripherals and Registers

9.1 The Processor Core

VS_DSP is a 16/32-bit DSP processor core that also has extensive all-purpose processor features. VLSI Solution's free VSIDE 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.

9.2 VS1010 Memory Map

TBD

9.3 VS1010 Peripherals

VS1010 system peripherals are located in Y-address space. The peripherals that use PLL clock are in addresses 0xFC00 - 0xFD3F and peripherals that use XTALI clock are in addresses 0xFE00 - 0xFEDF.

Peripheral address spaces are summarized in the following table:

VS1010 peripheral's address space allocation	
PLL clock domain peripherals	
Address	**Device**
0xFC00 - 0xFC1F	Interrupt controller
0xFC20 - 0xFC3F	Control and configuration registers for PLL clock domain
0xFC40 - 0xFC4F	SPI 0
0xFC50 - 0xFC5F	SPI 1
0xFC60 - 0xFC65	SPI Slave
0xFC66 - 0xFC6C	DSP interface for peripheral data buffer
0xFC6D - 0xFC75	AES
0xFC7B - 0xFC7F	SD card interface
0xFC80 - 0xFC9F	Hi-Speed USB
0xFCA0 - 0xFCBF	11-bit GPIO port 0
0xFCC0 - 0xFCDF	15-bit GPIO port 1
0xFCE0 - 0xFCFF	16-bit GPIO port 2
0xFD00 - 0xFD1F	S/PDIF
XTALI clock domain peripherals	
Address	**Device**
0xFE00 - 0xFE0F	UART0
0xFE10 - 0xFE1F	UART1
0xFE20 - 0xFE3F	Watchdog
0xFE60 - 0xFE7F	I2S
0xFE80 - 0xFE9F	Timers
0xFEA0 - 0xFEBF	RTC interface
0xFEC0 - 0xFED9	Control and configuration registers for 12 MHz clock domain
0xFEDA - 0xFEDC	Mems Mic Interface

9.4 Interrupt Controller

VS1010 has 25 maskable interrupt vectors and 27 interrupt sources. The interrupt controller is external to DSP and it prioritizes the requests before forwarding them to the DSP.

Interrupt controller has three levels of priority for simultaneous requests and a global disable/enable for all of the sources. Interrupt sources are divided so that interrupt sources 15-0 are mapped to low registers and 27-16 to high registers.

For an interrupt handler written in C, an assembly language stub that re-enables interrupts before RETI, should be written. The assembly language stub should call the C language handler routine.

VS1010 interrupt vectors

Source	Vector	Address	Device	Read also
INT_SAR	27	0x3b	12-bit ADC (SAR)	Chapter 9.18
INT_PWM	26	0x3a	Pulse width modulator	Chapter 9.20
INT_REGU	25	0x39	Power button	Chapter 9.6.1
INT_STX	23	0x37	S/PDIF transmitter	Chapter 9.12.4
INT_SRX	22	0x36	S/PDIF receiver	Chapter 9.12.1
INT_RTC	21	0x35	RTC time alarm	Chapter 9.17
INT_DAOSET	20	0x34	DAC offset	Chapter 9.7.3
INT_SRC	19	0x33	DAC sample rate converter	Chapter 9.7.4
INT_UART1_TX	18	0x32	UART1 transmit	Chapter 9.13
INT_UART1_RX	17	0x31	UART1 receive	Chapter 9.13
INT_TIMER2	16	0x30	Timer 2	Chapter 9.16

VS1010 interrupt vectors (continued)

Source	Vector	Address	Device	Read also
INT_TIMER1	15	0x2f	Timer 1	Chapter 9.16
INT_TIMER0	14	0x2e	Timer 0	Chapter 9.16
INT_UART0_RX	13	0x2d	UART0 receive	Chapter 9.13
INT_UART0_TX	12	0x2c	UART0 transmit	Chapter 9.13
INT_I2S	11	0x2b	I2S transmitter/receiver	Chapter 9.15
INT_GPIO2	9	0x29	Gpio port 2	Chapter 9.11
INT_GPIO1	8	0x28	Gpio port 1	Chapter 9.11
INT_GPIO0	7	0x27	Gpio port 0	Chapter 9.11
INT_MEMS	6	0x26	Mems Mic	Chapter 9.19
INT_SPI1	4	0x24	SPI 1	Chapter 9.8
INT_SPI0	3	0x23	SPI 0	Chapter 9.8
INT_XPERIP	2	0x22	Common Data Interfaces (SPI, SD, AES)	Chapter 9.9
INT_USB	1	0x21	Full/High-Speed USB	Chapter 9.10
INT_DAC	0	0x20	DAC	Chapter 9.7.1

9.4.1 Interrupt Controller Registers

The interrupt controller has three type of registers:

- Enable registers, which contain enable/disable bits for each interrupt source. Bit pairs configure the interrupt priority and disable.
- Origin registers, which contain the source flags for each interrupt. A request from an interrupt source sets the corresponding bit. A bit is automatically reset when a request for the source is generated.
- Enable counter register, which contains the value of the General Interrupt Enable counter, and two registers for increasing and decreasing the value.

Interrupt Controller Registers

Address	Type	Reset	Abbrev	Description
0xFC02	r/w	0	INT_ENABLE0_HP	Interrupt enable high priority for ints. 0..15
0xFC00	r/w	0	INT_ENABLE0_LP	Interrupt enable low priority for ints 0..15
0xFC03	r/w	0	INT_ENABLE1_HP	Interrupt enable high priority for ints 16..27
0xFC01	r/w	0	INT_ENABLE1_LP	Interrupt enable low priority for ints 16..27
0xFC04	r/w	0	INT_ORIGIN0	Interrupt origin for interrupts 0..15
0xFC05	r/w	0	INT_ORIGIN1	Interrupt origin for interrupts 16..27
0xFC06	r	0	INT_VECTOR[4:0]	Interrupt vector
0xFC07	r/w	0	INT_ENCOUNT[2:0]	Interrupt enable counter
0xFC08	w	0	INT_GLOB_DIS[-]	Interrupt global disable
0xFC09	w	0	INT_GLOB_ENA[-]	Interrupt global enable

9.4.2 Interrupt Enable INT_ENABLE[0/1]_[H/L]P

Interrupt enable registers selectively masks interrupt sources. Enable registers 0 contain sources 0..15 and enable registers 1 contain sources 16..27. Each source has two enable bits: one in the enable high priority (_HP) and one in the enable low priority (_LP) register. If both bits are zero, the corresponding interrupt source is not enabled, otherwise the bits select the interrupt priority.

_HP	_LP	Priority
0	0	Source disabled
0	1	Priority 1 (low)
1	0	Priority 2 (medium)
1	1	Priority 3 (high)

Priorities only matter when the interrupt controller decides which interrupt to generate for the core next. This happens whenever two interrupt sources request interrupts at the same time, or when interrupts become enabled after an interrupt handler routine or a part of code where the interrupts have been disabled.

9.4.3 Interrupt Origin INT_ORIGIN[0/1]

If an interrupt source requests an interrupt, the corresponding bit in the interrupt origin register (INT_ORIGIN0 or INT_ORIGIN1) will be set to ’1’. If an interrupt source is enabled (using

INT_ENABLE[0/1]_[H/L]P registers), the interrupt controller generates an interrupt request signal for VSDSP with the corresponding vector value. The bit in the origin registers is reset automatically after the interrupt is requested.

If the source is not enabled, the processor can read the origin register state and perform any necessary actions without using interrupt generation, i.e. polling of the interrupt sources is also possible. The bits in the interrupt origin registers can be cleared by writing '1' to them.

A read from the interrupt origin register returns the register state.

A write to the interrupt origin register clears the bits in the origin register that are set by the write. In other words, writing *b* to *INT_ORIGINx* performs the logical operation
INT_ORIGINx = INT_ORIGINx and (not b).

Example:
If value for INT_ORIGIN0 is 0x00FF, writing 0xF00F to it will end up with
INT_ORIGIN0 = 0x00FF and (not 0xF00F) = 0x00FF and 0x0FF0 = 0x00F0.

9.4.4 Interrupt Vector INT_VECTOR

The last generated vector value (0..27) can be read from the vector register.

9.4.5 Interrupt Enable Counter INT_ENCOUNT

The global interrupt enable/disable register INT_ENCOUNT is used to control whether an interrupt request is sent to the processor or not. If the 3-bit counter is zero, interrupt signal generation is enabled. While it is non-zero, interrupt requests are not forwarded to VSDSP. The counter is increased by one whenever the interrupt controller generates an interrupt request for VSDSP, or when the register INT_GLOB_DIS is written to. It is decresed by one if it is non-zero and the register INT_GLOB_ENA is written to.

When read, the enable counter register returns the counter value.

Don't write directly to INT_ENCOUNT. Manipulate its value by writing to INT_GLOB_DIS and INT_GLOB_ENA instead.

9.4.6 Interrupt Global Disable INT_GLOB_DIS

A write (of any value) to the global disable register increases the global interrupt enable/disable counter INT_ENCOUNT by one, thus disabling interrupts.

Note: If an interrupt is generated during the same clock cycle as a write to the global disable register, the interrupt enable counter is increased by two.

9.4.7 Interrupt Global Enable INT_GLOB_ENA

If the global interrupt enable/disable counter INT_ENCOUNT is not zero, a write (of any value) to INT_GLOB_ENA the counter by one.

The user must write to this register once at the end of interrupt handlers to re-enable interrupts.

9.5 PLL Clock Domain Control Registers

Peripheral control registers control the logic that is clocked with the PLL or USB clock.

9.5.1 General Purpose Software Registers

SW_REG0, SW_REG1, SW_REG2 and SW_REG3 are general purpose software registers. They are initialized to zero in reset and do not control any logic.

Software Registers				
Address	**Type**	**Reset**	**Abbrev**	**Description**
0xFC20	r/w	0	SW_REG0	16-bit general purpose sw register
0xFC21	r/w	0	SW_REG1	16-bit general purpose sw register
0xFC22	r/w	0	SW_REG2	16-bit general purpose sw register
0xFC23	r/w	0	SW_REG3	16-bit general purpose sw register

9.5.2 Peripheral I/O Control

VS1010 has three general purpose I/O ports. Ports 0 is 11 bits, port1 is 15-bits and port 2 is 16 bits. GPIO pins can be used either in GP mode or they can have also a special peripheral function. GPIO or peripheral function can be defined for each pin separately.

GPIO Mode Registers				
Address	**Type**	**Reset**	**Abbrev**	**Description**
0xFC2F	r/w	0	PERIP_CF	Peripheral operation mode
0xFC30	r/w	0	GPIO0_MODE	Mode control for gpio port 0
0xFC31	r/w	0	GPIO1_MODE	Mode control for gpio port 1
0xFC32	r/w	0	GPIO2_MODE	Mode control for gpio port 2

GPIO0_MODE, GPIO1_MODE and GPIO2_MODE registers are used to select current GPIO mode. By default all VS1010 pins are at GPIO mode and all GPIOx_MODE register are reset. If a peripheral mode is reguired the pin's GPIOx_MODE bit must be set ('1').

PERIP_CF Register Bits			
Name	**Bits**	**type**	**Description**
PERIP_CF_I2S48PIN	3	r/w	Alternate I2S pin configuration
PERIP_CF_MEMSCK1	2	r/w	Enable Mems Mic clock output 1
PERIP_CF_MEMSCK0	1	r/w	Enable Mems Mic clock output 0
PERIP_CF_UDACK	0	r/w	Enable UDA clock output

PERIP_CF_I2S48PIN register configures pins gpio0(1), gpio1(14), gpio1(13) and gpio1(12) for I2s peripheral. This setting overrides any other peripheral setting of pin.

PERIP_CF_MEMSCK1 (gpio0(10)), PERIP_CF_MEMSCK0 (gpio1(14)) and PERIP_CF_UDACK (gpio0(2)) registers enable clock outputs. These setting override any other peripheral setting of pin.

9.5.3 PLL Clock Control

VS1010 has two clock domains, the PLL clock domain and 12 MHz clock domain. The PLL is controlled with one register.

Clock Control Register				
Address	**Type**	**Reset**	**Abbrev**	**Description**
0xFC33	r/w	0	CLK_CF	PLL clock control register

CLK_CF Register Bits			
Name	**Bits**	**type**	**Description**
CLK_CF_EXTOFF	15	r/w	S/PDIF clock gate control
CLK_CF_SDOFF	14	r/w	SD, AES and SPI Slave peripheral clock gate control
CLK_CF_USBOFF	13	r/w	USB peripheral clock gate control
CLK_CF_RTCSLP	12	r/w	RTC power down mode enable
CLK_CF_LCKST	11	r/w	PLL vco lock status
CLK_CF_GDIV256	10	r/w	Global Clock 256-divider enable
CLK_CF_GDIV2	9	r/w	Global clock 2-divider enable
CLK_CF_LCKCHK	8	r/w	PLL vco lock check initialization
CLK_CF_VCOOUT	7	r/w	Enable PLL clock output pad driver
CLK_CF_USBCLK	6	r/w	Hi-Speed usb clock mode control
CLK_CF_FORCEPLL	5	r/w	PLL clock switch control
CLK_CF_DIVI	4	r/w	PLL input clock divider control
CLK_CF_MULT	3:0	r/w	PLL clock multiplier factor

CLK_CF_MULT determines the clock multiplier for input clock. Multiplier is value+1 i.e. value 1 means clock is multiplied by 2. Value 0 disables the PLL.

CLK_CF_DIVI controls the input divider of PLL's vco. If CLK_CF_DIVI is set the vco input clock is divided by two. If CLK_CF_DIVI is reset the vco input clock is the XTALI oscillator clock. When divider is used the CLK_CF_MULT can be programmed with values 1-15.

CLK_CF_FORCEPLL register controls the output clock switch. When set the output clock is PLL's vco clock. When reset the output clock is XTALI oscillator clock. It should be noted that the vco must be locked when CLK_CF_CKSW is modified.

CLK_CF_USBCLK selects Hi-Speed USB clock (UTM) insted of PLL vco clock. This clock must be selected before CLK_CF_FORCEPLL is modified. CLK_CF_MULT must have some value other than 0 when this clock mode is used. Also the Hi-Speed USB must be configured properly to output 60 MHz clock for core.

CLK_CF_VCOOUT enables the vco clock's output pad driver to pin gpio0(2). The pad must be in peripheral mode in order to drive clock. The output driver has glitch removal. This register overrides the udac and i2s master clock output setting for gpio0(2) pad.

CLK_CF_LCKCHK and CLK_CF_LCKST are used to poll vco lock status. When CLK_CF_LCKCHK is first set and reset the lock status can be read from CLK_CF_LCKST. If CLK_CF_LCKST remains set the PLL vco is locked.

CLK_CF_GDIV256 and CLK_CF_GDIV2 are the global clock dividers. These divider divide also the 12 MHz clock domain clock. PLL must be disabled when these dividers are used.

CLK_CF_RTCSLP enables RTC clocking mode.

CLK_CF_EXTOFF, CLK_CF_SDOFF and CLK_CF_USBOFF control peripheral clock gates. CLK_CF_SDOFF controls SD card, AES, SPI slave and peripeheral data buffer clocks. CLK_CF_EXTOFF controls S/PDIF peripheral clock. CLK_CF_USBOFF controls USB peripheral clock.

External Clock Output Divider Register				
Address	**Type**	**Reset**	**Abbrev**	**Description**
0xFC3F	r/w	0	VCO_DIV	Clock divider for external clock output (pin gpio0(2))

VCO_DIV Bits			
Name	**Bits**	**type**	**Description**
VCO_DIV_POL	3	r/w	Clock polarity
VCO_DIV_DIV	2:0	r/w	Core clock divider

VCO_DIV_POL register changes the clock polarity. VCO_DIV_DIV sets the core clock divider value. Value zero disables the divider. The core clock is divided by VCO_DIV_DIV + 1.

9.6 XTALI Clock Domain Control Registers

Peripheral control registers control the logic that is clocked with the XTALI clock (12.288 MHz).

9.6.1 Analog Control Registers

Analog Control Registers				
Reg	Type	Reset	Abbrev	Description
0xFECC	r/(w)	0	ANA_CF0	Analog Control register 0
0xFECB	r/w	0	ANA_CF1	Analog Control register 1
0xFED2	r/w	0	ANA_CF2	Analog Control register 2
0xFED3	r/w	0	ANA_CF3	Analog Control register 3
0xFED8	r/w	0	VCO_CCF_HI[9:0]	VCO frequency control MSB bits [25:16]
0xFED7	r/w	0	VCO_CCF_LO[15:0]	VCO frequency control LSB bits [15:0]

VCO_CCF register controls the VCO frequency within the selected VCO divider range.

ANA_CF0 Bits		
Name	Bits	Description
	6-5	Reserved, same value as in bit 4, read only
ANA_CF0_VCMST	4	Ground buffer short circuit monitor, read only
ANA_CF0_VCMDIS	3	Ground buffer driver short circuit protection disable
	2	Reserved, use ’0’
ANA_CF0_HIGH_REF	1	Analog reference voltage V$_{ref}$ = 1.2 V (0) or 1.6 V (1)
ANA_CF0_REF_ENA	0	Analog reference power enable

DAC ground buffer driver has a short circuit protection, which can be disabled with register ANA_CF0_VCMDIS. Short circuit protection is disabled when this register is set. ANA_CF0_VCMST register is a monitor for short circuit. If ground buffer is shorted to ground this flag register is set.

ANA_CF0_REF_ENA is the analog reference voltage power down. To enable reference this register must be set. ANA_CF0_HIGH_REF selects between two reference voltages. The reference voltage can be measured from Rcap pin.

ANA_CF1 register controls several analog module power enables. Each module is enabled when the power enable register bit is set.

ANA_CF1 Bits		
Name	**Bits**	**Description**
	15	Reserved, use '0'
ANA_CF1_VHMON	14	Regulator input voltage monitor (VHIGH)
ANA_CF1_PWRBTN	13	Power button pin state
ANA_CF1_BTNDIS	12	Power button delayed-reset disable
	11	Reserved, use '1'
ANA_CF1_XTDIV	10	Input clock divider for 24.576 MHz XTALI oscillator
ANA_CF1_2G_ENA	9	2 GHz VCO enable
ANA_CF1_SAR_ENA	8	SAR power and enable
ANA_CF1_DRVL_ENA	7	DAC left driver power enable (pwr_drv)
ANA_CF1_DRVR_ENA	6	DAC right driver power enable
ANA_CF1_DRVGB_ENA	5	DAC ground buffer driver power enable
ANA_CF1_DA_ENA	4	DAC power and enable (pwr_core)
ANA_CF1_DAGAIN_R	3:2	DAC right channel gain
ANA_CF1_DAGAIN_L	1:0	DAC left channel gain

ANA_CF1_VHMON register monitors the input voltage of the regulator (VHIGH). When input voltage is too low (about 1.07xAVDD) this register is set.

ANA_CF1_PWRBTN register monitors the current state of the power button pin. It should be noted that the power button can also be used as an interrupt source.

ANA_CF1_BTNDIS register disables the power button reset. When power button is pressed for more then 5s, a system reset is generated. When ANA_CF1_BTNDIS register is set, no reset is generated.

ANA_CF1_XTDIV is the input clock prescaler control register. When register is set the input clock is divided by 2.

ANA_CF1_2G_ENA is an enable register for RF vco. When High Speed USB is used this clock must be enabled.

ANA_CF1_SAR_ENA enables the 12-bit analog to digital convertter (SAR).

ANA_CF1_DRVL_ENA, ANA_CF1_DRVR_ENA, ANA_CF1_DRVGB_ENA and ANA_CF1_DA_ENA are DAC power enable registers. For stereo mode they must all be set.

ANA_CF1_DAGAIN_R and ANA_CF1_DAGAIN_L control DAC gain level.

DAC Gain ANA_CF1_DAGAIN_L[1:0] and ANA_CF1_DAGAIN_R[1:0] Values			
Name	**Value**	Gain	**Description**
ANA_CF1_DAGAIN_M6DB	3	-6 dB	
	2	-2 dB	Causes distortion in sound, do not use
ANA_CF1_DAGAIN_M12DB	1	-12 dB	
ANA_CF1_DAGAIN_0DB	0	0 dB	

ANA_CF3 register controls the RF vco operation.

ANA_CF3 Bits

Name	Bits	Description
ANA_CF3_SDM_ENA	13	2 GHz vco's SDM enable
ANA_CF3_LCKST	12	2 GHz vco lock status
ANA_CF3_LCKCHK	11	2 GHz vco lock check init
ANA_CF3_UTMBIAS	10	USB pad bias enable
ANA_CF3_480_ENA	9	480 MHz clock enable
ANA_CF3_UTM_ENA	8	Hi-Speed USB UTM enable
ANA_CF3_CKDIV[1:0]	7:6	Clock divider select registers, use 0x0
ANA_CF3_DIV[1:0]	5:4	VCO divider select registers, use 0x0
ANA_CF3_2GCNTR[3:0]	3:0	VCO center frequency register

ANA_CF3_SDM_ENA enables the digital SDM which is used to adjust VCO's frequency (fine tuning).

ANA_CF3_LCKCHK and ANA_CF3_LCKST are used to poll 2 GHz vco lock status. When ANA_CF3_LCKCHK is first set and reset the lock status can be read from ANA_CF3_LCKST. If ANA_CF3_LCKST remains set the 2 GHz VCO is locked.

ANA_CF3_UTMBIAS is enable register for High Speed USB pads. In high Speed mode this register must be set. In full speed and suspend modes this register can be reset.

ANA_CF3_480ENA is the 480 MHz clock driver enable for UTM. When set the clock driver is enabled.

ANA_CF3_UTM_ENA is enable for USB UTM logic. When USB is used this register must be set.

ANA_CF3_2GCNTR register is used to match VCO's center frequency to programmed value (dividers and VCO_CCF). The value of this register is sweeped until ANA_CF3_LCKST returns a high value indicating that VCO is in lock to XTAL clock.

ANA_CF3_CKDIV and ANA_CF3_DIV are VCO's divider configuration registers. Use value 0x0 for both.

VCO frequency of 1.92GHz results to 480 MHz USB clock.

USB Clock congiguration for 60MHz

Xtal	ANA_CF3_DIV	ANA_CF3_CKDIV	VCO_CCF
12.288MHz	0x0	0x0	0xFF87 FFFF (-7864321)
12.000MHz	0x0	0x0	0x0000 0000

$F_{vco} = (4 \times VCO_{div} + CCF) \times F_{xtal}$ where CCF is defined as $CCF = \frac{VCO\_CCF_{reg}}{2^{21}} + 16$

and the USB clock frequency can be given as:

$$F_{USB} = ((4 \times VCO_{div} + CCF) \times F_{xtal})/CK_{div}$$

CCF must be equal or lower than VCO Divider. If e.g. VCO Divider = 25, CCF must not be higher than 25, and between 24-25 the frequency is not accurate. The reason for this is the sigma-delta modulator's output that gets clipped. Furthermore, when VCO Divider = 36, CFF must be lower than approximately 31. Output of the SDM may also become unstable if CCF is less than approximately 0.3.

9.6.2 Regulator and Peripheral Clock Control Registers

VS1010 has three internal regulators, one regulator for each power domain. The voltage can be adjusted in about 50mV step size.

Regulator and Clock Control				
Reg	Type	Reset	Abbrev	Description
0xFECE	r/w	0	REGU_CF	Regulator control register
0xFED0	r/w	0	REGU_VOLT	Regulator voltage register

REGU_VOLT Bits		
Name	Bits	Description
REGU_VOLT_AVDD[4:0]	14:10	Analog voltage configuration 2.7V-3.6V
REGU_VOLT_IOVDD[4:0]	9:5	IO voltage configuration, 1.8V-3.6V
REGU_VOLT_CVDD[4:0]	4:0	Core voltage configuration, 1.65V-1.9V

REGU_CF Bits		
Name	Bits	Description
REGU_CF_REGCK	3	Regulator latch enable
REGU_CF_AOFF	2	Analog regulator shutdown
REGU_CF_IOOFF	1	IO regulator shutdown
REGU_CF_COFF	0	Core regulator shutdown

REGU_CF_REGCK is used to latch in the regulator voltage and shutdown bits. Typical values for voltages are calculated from equations:

- $CVDD = 1.24V + (30mV * voltage\ register)$
- $IOVDD = 1.80V + (60mV * voltage\ register)$
- $AVDD = 2.48V + (40mV * voltage\ register)$

9.7 Audio Playback Interfaces

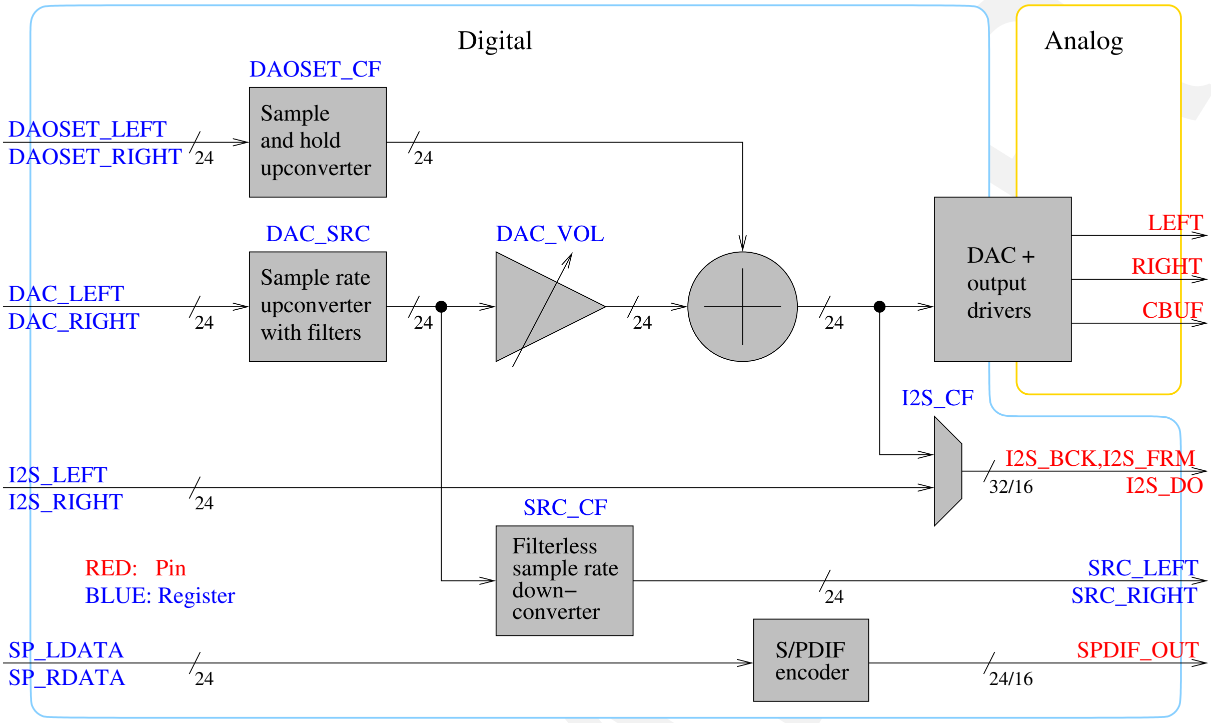


Figure 8: VS1010 playback (DA) audio paths.

The VS1010 playback audio paths are shown in Figure 8.

The nominal high-quality audio path egins from registers DAC_LEFT and DAC_RIGHT, then goes through the Sample Rate Upconverter with Filters and the Volume Control to the DAC, and finally to the LEFT and RIGHT output pins. CBUF is used as a ground reference.

For lower-quality sound effects, secondary audio with a potentially different sample rate can be independently added to the signal through the DAOSET_LEFT and DAOSET_RIGHT registers. The upconverter for this path contains only sample-and-hold filtering, so using low sample rates may result in audible aliasing.

The combined main and secondary audio path signal can optionally also be copied to the I2S output. Alternatively the I2S output can be controlled directly using registers I2S_LEFT and I2S_RIGHT.

The main audio path may be intercepted and downsampled with the Filterless Sample Rate Downconverter. Because the downconverter is filterless, the user has to take care to select sample rates in such a way that doesn't introduce audible aliasing. Aliasing does not occur if the sample rate for the output (SRC_LEFT and SRC_RIGHT) is never lower than the sample rate for the input (DAC_LEFT and DAC_RIGHT), but in special cases even lower Downconverter sample rates may result in audio that is good enough for the application.

Not directly connected to any other part of the playback audio path is the S/PDIF signal path. This signal path does not interact with the other ones, and it is only included in the figure to show all available playback audio paths.

9.7.1 Primary Audio Path 24-bit Sample Rate Upconverter with Filters: DAC Registers

VS1010 has a 24-bit DAC with a programmable sample rate. Sample rates up to 96 kHz are supported.

DAC Interface Registers				
Address	Type	Reset	Abbrev	Description
0xFC34	r/w	0	DAC_SRCL	DAC sample rate, bits 15-0
0xFC35	r/w	0	DAC_SRCH[3:0]	DAC sample rate, bits 19-16
0xFC36	r/w	0	DAC_LEFT_LSB[15:8]	DAC left sample, bits 7-0
0xFC37	r/w	0	DAC_LEFT	DAC left sample, bits 23-8
0xFC38	r/w	0	DAC_RIGHT_LSB[15:8]	DAC right sample, bits 7-0
0xFC39	r/w	0	DAC_RIGHT	DAC right sample, bits 23-8
0xFED9	r/w	0	DAC_MODE	DAC mode configuation

The DAC interpolator frequency is defined with registers DAC_SRCH and DAC_SRCL which combined form the 20-bit register DAC_SRC. Output sample rate is derived from the rollover frequency of a 20-bit interpolator accumulator. Its accumulation rate is specified by ifreq.

Input sample rate f_s can be calculated from the equation

$$f_s = (XTALI/2^{27}) \times DAC\_SRC$$

where DAC_SRC can have values from 1 to 1048575 (0xFFFFF). Value zero of ifreq places the DAC in idle mode. In idle mode all logic is halted. Also the analog clock is halted.

Note that the DAC clock is not controlled by the PLL.

The exact sample rate is xtal dependent and a sample rate of e.g exactly 48 kHz requires that XTALI = 12.288 MHz.

24-bit samples are written to registers DAC_LEFT, DAC_LEFT_LSB, DAC_RIGHT and DAC_RIGHT_LSB after each DAC interrupt.

DAC_MODE Bits		
Register	Bit	Description
DAC_MODE_3MUAD	12	Mems mic / uda 3MHz clock mode
DAC_MODE_96K	11	DAC 96 kHz mode
DAC_MODE_SRCADD	10:4	DAC SRC fine tuning increment bits
DAC_MODE_SRCCNT	2:0	DAC SRC fine tuning counter bits

DAC_MODE_3MUAD register selects between 3 Mhz and 6 Mhz clock for external circuitry. This clock pin can be used with external DAC (6 Mhz mode) and ADC / MEMS MIC (3 Mhz mode) modules. The clock output pin is gpio0(2). For MEMS MICs there are also two 3 Mhz clock output pins gpio0(10) and gpio1(14).

UDAC output pins are left = SPEAKER0 (gpio0(7)) and right = SPEAKER1 (gpio0(8)). The data rate is 6mbs. For pin configuration see Section 9.5.2.

DAC_MODE_96K register enables DAC sample rates over 48 kHz. In this mode the SNR is somewhat degraded. The sample rate is calculated from equation:

$fs = (XTALI/2^{26}) * DAC\_SRC$ where

DAC_SRC can have values from 1 to 532480 (0x82000). Value zero of ifreq places the DAC in idle mode. In idle mode all logic is halted. Also the analog clock is halted. When changing mode the DAC's sample rate must be zero (i.e. DAC disabled).

Registers DAC_MODE_SRCADD and DAC_MODE_SRCCNT are used for fine tuning the DAC sample rate. DAC_MODE_SRCCNT sets the 3-bit counter (0 to 7) to count from 0 to DAC_MODE_SRCCNT. DAC_MODE_SRCADD register bits make an additional increment by one to the interpolator accumulator when they are set and the counter value matches the bit index. If exact sample rates like 8 kHz, 16 kHz or 32 kHz are needed these registers must be used. When DAC_MODE_SRCCNT is zero the logic is disabled.

Configuring Analog DAC Modules

Example values of analog configuration registers with 1.6 V reference are given in next table.

Analog Control Register example for DAC Operation

Address	Register	Value	Description
0xFECB	ANA_CF1	0x00F0	DAC and output drivers power down
0xFECC	ANA_CF0	0x0003	Reference voltage select and reference power down

9.7.2 Primary Audio Path Volume Control

In VS1010 the DAC's volume level can be adjusted in -0.5dB steps.

DAC Volume Registers				
Reg	**Type**	**Reset**	**Abbrev**	**Description**
0xFEC0	r/w	0	DAC_VOL	DAC volume control register

DAC_VOL Bits		
Name	**Bits**	**Description**
DAC_VOL_LADD[3:0]	15:12	Left channel +0.5dB steps
DAC_VOL_LSFT[3:0]	11:8	Left channel -6dB steps
DAC_VOL_RADD[3:0]	7:4	Right channel +0.5dB steps
DAC_VOL_RSFT[3:0]	3:0	Right channel -6dB steps

DAC_VOL_LSFT and DAC_VOL_RSFT are the coarse volume control registers. They suppress channel volume by -6dB steps.

DAC_VOL_LADD and DAC_VOL_RADD are the fine volume control registers. They add channel volume level by +0.5dB steps. Allowed values are from 0 to 11, i.e. maximum is +5.5dB. Values between 12-15 equal to 0dB.

9.7.3 Secondary Audio Path: DAOSET Registers

In VS1010 a secondary audio source can be mixed to the main audio path output. This is done with DAC offset registers. The sample rate is programmable.

DAC Offset Registers				
Reg	**Type**	**Reset**	**Abbrev**	**Description**
0xFEC1	r/w	0	DAOSET_CF	DAC offset configuration register
0xFEC2	r/w	0	DAOSET_LEFT_LSB[15:12]	DAC left offset bits [3:0]
0xFEC3	r/w	0	DAOSET_LEFT	DAC left offset bits [19:4]
0xFEC4	r/w	0	DAOSET_RIGHT_LSB[15:12]	DAC right offset bits [3:0]
0xFEC5	r/w	0	DAOSET_RIGHT	DAC right offset bits [19:4]

DAOSET_CF Bits		
Name	**Bits**	**Description**
DAOSET_CF_URUN	14	Data register underrun flag
DAOSET_CF_FULL	13	Data register full flag
DAOSET_CF_ENA	12	Enable for DAC offset
DAOSET_CF_FS	11:0	DAC offset sample rate

DAOSET_CF_URUN is an underrun flag register. The register is set if data register was read when the full flag was not set.

DAOSET_CF_FULL is a data status register. Flag is set when data is written to DAOSET_LEFT and DAOSET_RIGHT registers and reset when DAC reads the register.

DAOSET_CF_ENA enables DAC offset module.

DAOSET_CF_FS is used to set DAC offset sample rate. This register defines the interval in clock cycles where the samples are added to DAC output. When new samples are read from data registers also an interrupt request is generated.

Sample rate can be calculated from equation:

$fs = F_{clk}/(dacoffset\_cf\_fs + 1))$ where

dacoffset_cf_fs can have values from 0 to 4095 (0xFFF) and F_{clk} is the XTALI clock frequency. E.g. value 0xFFF gives sample rate of 12.288 MHz / (0xFFF+1) = 3.0 kHz.

DAC and DAC offset mixing logic uses saturation to limit samples to 20-bit signed values. The mixed values should not exceed 75% of the full scale values or the signal to noise ratio may be degraded.

9.7.4 Filterless Sample Rate Converter (SRC) Registers

VS1010 has a programmable sample rate converter which can be used to convert DAC's input sample rate to an other sample rate which is higher than the original sample rate.

SRC Characteristics

Item	Value	Description
XTALI Clock	11.0 MHz - 13.0 MHz	Clock frequency
DAC bit width	24	Input data width
SRC bit width	24	Output data width
DAC sample rate [1]	0 Hz - 96 kHz	Input sample rate
Output sample rate [1]	0.97 $\times FS_{in}$ - 192 kHz	Output sample rate
Filter delay [2]	19 input samples	
Gain	0.78	

[1] Assuming 12.288 MHz XTALI clock.
[2] In start-up the SRC output is valid after 19 DAC interrupts.

SRC Registers

Reg	Type	Reset	Abbrev	Description
0xFEC6	r/w	0	SRC_CF	SRC sampler configuration register
0xFEC7	r/w	0	SRC_LEFT_LSB[15:12]	SRC left sample bits [7:0]
0xFEC8	r/w	0	SRC_LEFT	SRC left sample bits [23:8]
0xFEC9	r/w	0	SRC_RIGHT_LSB[15:12]	SRC right sample bits [7:0]
0xFECA	r/w	0	SRC_RIGHT	SRC right sample bits [23:8]

SRC_CF Bits

Name	Bits	Description
SRC_CF_ORUN	15	SRC overrun flag
SRC_CF_RFULL	14	Right data register full flag
SRC_CF_LFULL	13	Left data register full flag
SRC_CF_ENA	12	Enable for sample rate convertter
SRC_CF_FS	11:0	SRC sample rate

SRC_CF_ORUN is set if data register was full when data registers were modified.

SRC_CF_RFULL and SRC_CF_LFULL status registers for new samples. Flags are set as SRC_LEFT and SRC_RIGHT are modified and reset as they are read.

SRC_CF_ENA enables sample rate converter when set.

SRC_CF_FS is used to set src sample rate. This register defines the interval in clock cycles when the samples are generated. When new samples are stored to data registers also an interrupt request is generated.

Output sample rate can be calculated from equation:

$$fs = XTALI/(2 * (src\_cf\_fs + 1))$$

where src_cf_fs can be between 0 and 4095 (0xFFF).

Example: With src_cf_fs = 0x7FF, the sample rate fs = 12.288 MHz / (2*(0x7FF+1)) = 3000 Hz.

9.8 SPI Peripherals

VS1010 has two SPI (serial-to-paralle) peripherals which can be configured as a master or a slave. Both SPIs support 1-, 2- and 4-bit data transfers. With internal Flash only 1-bit and 2-bit modes can be used. Before SPIs can be used the VS1010 I/Os must be configured to peripheral mode:

- *set I/O pins to peripheral mode* : GPIO1_MODE register selects between spi mode or gpio mode
- *Buffered SPI slave disabled* : SPI2_RXLEN_PMODE bit reset when using SPI1

SPI0 and SPI1 pins are mapped to GPIO1 port. To select peripheral mode the bits in GPIOx_MODE register must be set HIGH.

SPI1 pins are also shared with SPI2 (SPI slave). The spi is selected with register SPI2_RXLEN_PMODE.

SPI pins and their GPIOx_MODE register					
SPI id	**VS1010 pin**	**Type**	**SPI pin**	**GPIO_MODE register**	**Description**
SPI0	MOSI0/GPIO1[0]	i/o	mosi	GPIO1_MODE[0]	Master output / slave input
SPI0	MISO0/GPIO1[1]	i/o	miso	GPIO1_MODE[1]	Master input / slave output
SPI0	SCK0/GPIO1[2]	i/o	sck	GPIO1_MODE[2]	Master/slave clock
SPI0	XCS0/GPIO1[3]	i/o	xcs	GPIO1_MODE[3]	Master/slave chip select
SPI1	MOSI1/GPIO1[4]	i/o	mosi	GPIO1_MODE[4]	Master output / slave input
SPI1	MISO1/GPIO1[5]	i/o	miso	GPIO1_MODE[5]	Master input / slave output
SPI1	SCK1/GPIO1[6]	i/o	sck	GPIO1_MODE[6]	Master/slave clock
SPI1	XCS1/GPIO1[7]	i/o	xcs	GPIO1_MODE[7]	Master/slave chip select

The SPIs are mapped in Y addresses 0xFC40 (SPI0) and 0xFC50 (SPI1).

SPI Registers, Prefix SPIx_					
SPI0 address	**SPI1 address**	**Type**	**Reset**	**Abbrev**	**Description**
0xFC40	0xFC50	r/w	0	CF	Configuration
0xFC41	0xFC51	r/w	0	CLKCF[9:0]	Clock configuration
0xFC42	0xFC52	r/w	0	STATUS[7:0]	Status
0xFC43	0xFC53	r/w	0	DATA	Sent / received data
0xFC44	0xFC54	r/w	0	FSYNC	SSI Sync data in master mode
0xFC45	0xFC55	r/w	0	DEFAULT	Data to send (slave) if SPIx_ST_TXFULL='0'

Main Configuration SPIx_CF Bits		
Name	**Bits**	**Description**
SPI_CF_MBITENA	15	enable for multi bit modes
SPI_CF_BUSMODE	14:13	Select SPI bus bit width, 1-, 2- or 4-bit bus
SPI_CF_EARLYINT	12	’1’ = interrupt when SPI_ST_TXFULL clear (TX mode) ’0’ = interrupt when no transfer ready (RX mode)
SPI_CF_SRESET	11	SPI software reset
SPI_CF_RXFIFOMODE	10	’1’ = interrupt only when FIFO register full or CS deasserted with receive register full ’0’ = interrupt always when a word is received
SPI_CF_RXFIFO_ENA	9	Receive FIFO enable
SPI_CF_TXFIFO_ENA	8	Transmit FIFO enable
SPI_CF_XCSMODE	7:6	xCS mode in slave mode
SPI_CF_MASTER	5	Master mode
SPI_CF_DLEN	4:1	Data length in bits
SPI_CF_FSIDLE	0	Frame sync idle state

SPI_CF_MBITENA register enables the multi bit bus modes.

SPI_CF_BUSMODE register selects SPI bus width.

SPI_CF_BUSMODE Bits		
Value	**Bus Mode**	**Description**
11 or 10	4-bit mode	SPIx master/slave SPI is in 4-bit bus mode, Not supported with iFlash
01	2-bit mode	SPIx master/slave SPI is in 2-bit bus mode
00	1-bit mode	SPIx master/slave SPI is in 1-bit bus mode

SPI_CF_EARLYINT selects whether the SPI interrupt happens immediately when the SPI device is capable of taking new data (1, useful for when transmitting data), or only when the SPI transfer has been fully completed (0, useful when mostly receiving data).

SPI_CF_XCSMODE selects xCS mode for slave operation. ’00’ is interrupted xCS mode, ’10’ is falling edge xCS mode, and ’11’ is rising edge xCS mode.

SPI_CF_MASTER sets master mode. If not set, slave mode is used.

SPI_CF_DLEN+1 is the length of SPI data in bits. Example: For 8-bit data transfers, set SPI_CF_DLEN to 7.

SPI_CF_FSIDLE contains the state of FSYNC when SPI_ST_TXRUNNING is clear. This bit is only valid in master mode.

Clock Configuration SPIx_CLKCF Bits		
Name	**Bits**	**Description**
SPI_CC_CLKDIV	9:2	Clock divider
SPI_CC_INV_CLKPOL	1	Inverse clock polarity selection
SPI_CC_INV_CLKPHASE	0	Inverse clock phase selection

In master mode, SPI_CC_CLKDIV is the clock divider for the SPI block. The generated SCLK frequency $f = \frac{f_m}{2\times(c+1)}$, where f_m is the master clock frequency and c is SPI_CC_CLKDIV.

Example: With a 12 MHz master clock, SPI_CC_CLKDIV=3 divides the master clock by 4, and the output/sampling clock would thus be $f = \frac{12\,MHz}{2\times(3+1)} = 1.5\,MHz$.

SPI_CC_INV_CLKPOL reverses the clock polarity. If SPI_CC_INV_CLKPOL is clear the data is read at rise edge and written at fall edge if SPI_CC_INV_CLKPHASE is clear. When SPI_CC_INV_CLKPHASE is set the data is written at rise edge and read at fall edge.

SPI_CC_INV_CLKPHASE defines the data clock phase. If clear the first data is written when xcs is asserted and data is sampled at first clock edge (rise edge when SPI_CC_INV_CLKPOL = 0 and fall edge if SPI_CC_INV_CLKPOL = 1). If SPI_CC_INV_CLKPHASE is set the first data is written a the first data clock edge and sampled at second.

Status SPIx_STATUS Bits

Name	Bits	Description
SPI_ST_RXFIFOFULL	7	Receiver FIFO register full
SPI_ST_TXFIFOFULL	6	Transmitter FIFO register full
SPI_ST_BREAK	5	Chip select deasserted mid-transfer
SPI_ST_RXORUN	4	Receiver overrun
SPI_ST_RXFULL	3	Receiver data register full
SPI_ST_TXFULL	2	Transmitter data register full
SPI_ST_TXRUNNING	1	Transmitter running
SPI_ST_TXURUN	0	Transmitter underrun

SPI_ST_BREAK is set in slave mode if chip select was deasserted in interrupted xCS mode or a starting edge is encountered in xCS edge modes while a data transfer was in progress. This bit has to be cleared manually.

SPI_ST_RXORUN is set if a received byte overwrites unread data when it is transferred from the receiver shift register to the data register. This bit has to be cleared manually.

SPI_ST_RXFULL is set if there is unread data in the data register.

SPI_ST_TXFULL is set if the transmit data register is full.

SPI_ST_TXRUNNING is set if the transmitter shift register is in operation.

SPI_ST_TXURUN is set if an external data transfer has been initiated in slave mode and the transmit data register has not been loaded with new data to shift out. This bit has to be cleared manually.

Note: Because TX and RX status bits are implemented as separate entities, it is relatively easy to make asynchronous software implementations, which do not have to wait for an SPI cycle to finish.

SPIx_DATA[SPI_CF_DLEN:0] may be written to whenever SPI_ST_TXFULL is clear. In master mode, writing will initiate an SPI transaction cycle of SPI_CF_DLEN+1 bits. In slave mode, data is output as soon as suitable external clocks are offered. Writing to SPI_DATA sets SPI_ST_TXFULL, which will again be cleared when the data word was put to the shift register. If SPI_ST_TXRUNNING was clear when SPI_DATA was written to, data can immediately be transferred to the shift register and SPI_ST_TXFULL won't be set at all.

When SPI_ST_RXFULL is set, SPI_DATA may be read. Bits SPI_CF_DLEN:0 contain the received data. The rest of the 16 register bits are set to 0.

SPIx_FSYNC is meant for generation of potentially complex synchronization signals, including several SSI variants as well as a simple enough automatic chip select signal. SPIx_FSYNC is only valid in master mode.

If a write to SPIx_DATA is preceded by a write to SPIx_FSYNC, the data written to SPIx_FSYNC is sent to FSYNC pin with the same synchronization as the data written to SPIx_DATA is written to MOSI. When SPI_ST_TXRUNNING is clear, the value of SPI_CF_FSIDLE is set to FSYNC pin.

If SPIx_DATA is written to without priorly writing to SPIx_FSYNC, the last value written to SPIx_FSYNC is sent.

SPIx_FSYNC is double-buffered like SPIx_DATA.

The SPI block has one interrupt. Interrupt 0 request is sent when SPI_ST_BREAK is asserted, or when SPI_ST_TXFULL or SPI_ST_TXRUNNING is cleared. This allows for sending data in an interrupt-based routine, and turning chip select off when the device becomes idle.

9.9 Common Data Interfaces

VS1010 has a 3 KiB data buffer which is a dedicated peripheral memory. The memory can be configured to be used with:

- SPI slave interface
- SD Card Interface
- AES Decrypt

Block diagram of the data interfaces is shown in Figure 9.

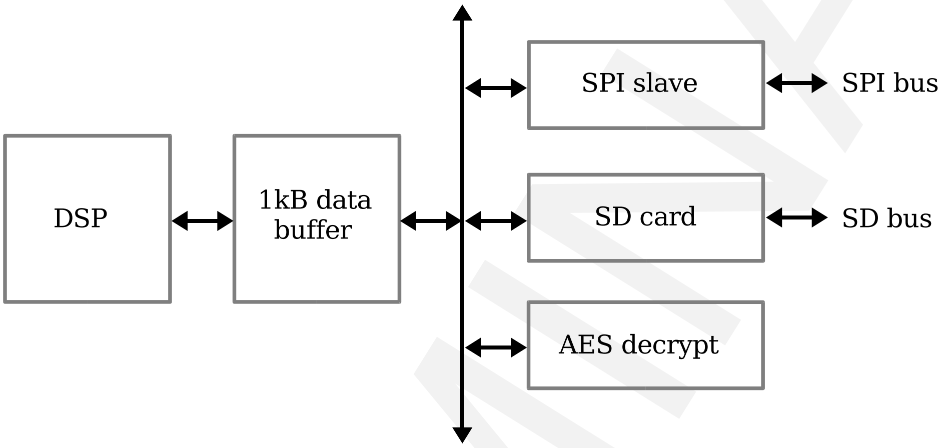


Figure 9: Block diagram of data interfaces.

Each peripheral can be configured to use its own address space. The DSP interface has a read and write port with auto incrementing address register. The read operation is pipelined and requires two reads to fill the pipeline. After that the memory can be read on each instruction cycle. It should be noted that the memory is time multiplexed between the peripherals, with the DSP having absolute highest priority. Because of this, some idle cycles are required during long DSP read/write operations. As a guideline at least every 16th read cycle should be left idle by the DSP. The DSP interface has Error Correction Code (ECC) registers for nand flash. It uses 2D xor to protect and correct data.

DSP Interface Registers for Peripheral Memory				
Reg	Type	Reset	Abbrev	Description
0xFC66	r	0	ECC_LP_LOW	ECC line parity register bits [15:0]
0xFC67	r	0	ECC_CP_LP_HIGH	ECC column parity bits [5:0] and line parity bits [17:16]
0xFC68	r/w	0	XP_CF	Dsp interface control
0xFC69	r/w	0	XP_ADDR	Memory address register for dsp interface, 11 bits
0xFC6A	r/w	0	XP_ODATA	Memory write port for dsp
0xFC6B	r	0	XP_IDATA	Memory read port for dsp
0xFC6C	r/w	0	XP_ST	Interrupt status register for data buffer peripherals

ECC_LP_LOW and ECC_CP_LP_HIGH are the error correction code data registers. They are modified when DSPI_ODATA or DSPI_IDATA ports are accessed. The DSPI_CF_ECCENA must be set in order to use ECC.

XP_CF Bits		
Name	**Bits**	**Description**
XP_CF_AES_ENA	10	AES 128-bit block decrypt enable
XP_CF_ECC_RST	9	ECC reset
XP_CF_ECC_ENA	8	ECC enable
XP_CF_WRBUF_ENA	1	Data buffer write enable
XP_CF_RDBUF_ENA	0	Data buffer read enable

XP_CF_ECC_RST and XP_CF_ECC_ENA control the ECC unit. XP_CF_ECC_RST reset the unit when set. The register is reset automatically after one clock cycle.

XP_CF_ECC_ENA register enables the ECC calculation. Column parity (CP) and line parity (LP) registers are modified when data is read from XP_IDATA or written to XP_ODATA register and XP_CF_ECC_ENA is set.

XP_CF_WRBUF_ENA and XP_CF_RDBUF_ENA enable the dsp access to peripheral data buffer. When either register is set the XP_ADDR is incremented on each memory access and data is read (XP_IDATA) or written (XP_ODATA) to memory.

Data interfaces can generate only one common interrupt request for the DSP, INT_XPERIP (see Chapter 9.4, *Interrupt Controller*. The interrupt source is stored in the interrupt status register XP_ST.

XP_ST status register is used to track the interrupt source of the peripherals using data buffer memory. With the exception of bit XP_ST_INT_ENA, and XP_ST_SPIERR_INT, a write to XP_ST bits clears the bits in the origin register that are set in the data word. I.e. writing for example 0x4020 to XP_ST register clears the XP_ST_SD_INT interrupt request.

Interrupt sources are listed in the table below.

XP_ST Bits		
Name	**Bits**	**Description**
XP_ST_INT_ENA	14	Interrupt enable for data buffer peripherals
XP_ST_SPITXRB_HALF2_INT[1]	13	SPI slave transmit ring buffer second half empty
XP_ST_SPITXRB_HALF1_INT[1]	12	SPI slave transmit ring buffer first half empty
XP_ST_SPIRXRB_HALF2_INT[1]	11	SPI slave receive ring buffer second half full
XP_ST_SPIRXRB_HALF1_INT[1]	10	SPI slave receive ring buffer first half full
XP_ST_SPIERR_INT[2]	9	SPI slave error, transfer was interrupted middle of byte
XP_ST_AES_INT[1]	6	AES block decrypt ready
XP_ST_SD_INT[1]	5	SD card interface ready interrupt
XP_ST_SPI_STOP_INT[1]	3	SPI slave stop interrupt, chip select to inactive state
XP_ST_SPI_START_INT[1]	2	SPI slave start interrupt, chip select to active state

[1] A write with the bit set will clear the bit.
[2] Read-only bit.

XP_ST_INT_ENA is the peripheral interrupt enable. When set the interrupt requests are forwarded to the interrupt controller. Interrupt requests in XP_ST are modified regardless of the value of XP_ST_INT_ENA.

The SPI slave error register (XP_ST_SPIERR_INT) is a read only register which is reset when SPI start is detected in the SPI bus and set if data transfer was interrupted in the middle of a byte.

9.9.1 SD Card Interface

VS1010 has a SD card interface which supports 1-bit and 4-bit data bus.

SD Card Interface Registers				
Reg	Type	Reset	Abbrev	Description
0xFC7C	r/w	0	SD_PTR	SD card memory address pointer
0xFC7D	r/w	0	SD_LEN	SD card data length, in bytes
0xFC7E	r/w	0	SD_CF	SD card configuration register
0xFC7F	r/w	0	SD_ST	SD card status register

SD_PTR is the 11-bit memory pointer register.

SD_LEN defines the number of bytes that are read from or written to SD card. The length is given in bytes.

SD_CF Bits		
Name	Bits	Description
SD_CF_NOCRCTX	12	Do not send crc (continued operation)
SD_CF_NOCRCRST	11	Do not reset crc register (continued operation)
SD_CF_4BIT	10	Use 4-bit data bus mode
SD_CF_ENA	7	Start SD card transfer
SD_CF_READSEL	6	Read (1) or write (0) select
SD_CF_CMDSEL	5	Command or data transfer select
SD_CF_NOSTARTB	4	Skip data start bit (continued operation)
SD_CF_NOSTOPB	3	Do not add data stop bit (continued operation)
SD_CF_CRC16	2	Enable crc16 calculation during write
SD_CF_CRC7	1	Enable crc7 calculation during write
SD_CF_POLL	0	Poll for start bit when read

SD_CF_NOCRCTX makes the interface to skip crc transfer.

SD_CF_NOCRCRST makes the interface to continues crc calculation from previous transfer.

SD_CF_4BIT forces the interface to use 4-bit data transfer instead of 1-bit if set.

SD_CF_ENA start SD card read or write transfer when set.

SD_CF_READSEL register selects a read transfer.
For code clarity SD_CF_WRITESEL has also been defined (as zero).

SD_CF_CMDSEL register selects between command and data transfers.
For code clarity SD_CF_DATASEL has also been defined (as zero).

SD_CF_NOSTARTB register forces the interface to skip start bit when set.

SD_CF_NOSTOPB register forces the interface to skip stop bit when set.

SD_CF_CRC16 and SD_CF_CRC7 enable the crc calculation. Crc is send automatically if SD_CF_NOCRCTX is reset.

SD_CF_POLL forces the SD card interface to search for start bit when reading command response or data. If start bit is not found during 256 SD clock cycles the operation is cancelled and SD_ST_NOSTR error flag is set.

SD_ST Bits		
Name	**Bits**	**Description**
SD_ST_WAITSTATES	12:8	SD card clock configuration
SD_ST_REPEAT	7	Repeat mode enable
Reserved	6	Use '0'
SD_ST_CMDBRK	5	cmd response during data transfer
SD_ST_DAT0	4	SD card dat0 bus state
SD_ST_NOSTOPB_ERR	3	data stop bit missing error
SD_ST_CRC16_ERR	2	crc16 error when reading data
SD_ST_CRC7_ERR	1	crc7 error when reading command response
SD_ST_NOSTARTB_ERR	0	timeout error when reading, no start bit

SD_ST_WAITSTATES configures the length of SD card clock cycle. The cycle time is 2 x (SD_ST_WAITSTATES + 1) dsp clock cycles.

SD_ST_REPEAT sets the interface into a pattern generation mode. In this mode the SD data lines repeat a 512 byte buffer continuously. The buffer's location in memory can be set with registers SD_PTR[10:8]. In this mode all other SD_ST and SD_CF registers should be reset. The SD_ST_WS and SD_CF_4BIT have their usual meaning.

SD_ST_CMDBRK is set if a cmd start bit is found during data transfer. This register is reset at the start of each SD card op.

SD_ST_DAT0 register samples the SD cards data 0 line.

SD_ST_NOSTOPB_ERR is set if stop bit was not found when reading data from SD card.

SD_ST_CRC16_ERR is set if crc16 error was detected when reading data from SD card.

SD_ST_CRC7_ERR is set if command response had a crc7 error.

SD_ST_NOSTARTB_ERR is set if start bit was not found during 256 SD clocks.

For code clarity also SD_ST_ANY_ERR has been defined as (SD_ST_NOSTOPB_ERR | SD_ST_CRC16_ERR | SD_ST_CRC7_ERR | SD_ST_NOSTARTB_ERR).

9.9.2 SPI Slave Peripheral

VS1010 has a secondary SPI slave which uses peripheral ram. In this mode the SPI1 pins are used and they must be configured to peripheral mode with GPIO1_MODE[7:4] registers.

SPI Slave Registers				
Reg	Type	Reset	Abbrev	Description
0xFC60	r/w	0	SPI2_CFG	SPI slave config register
0xFC61	r/w	0	SPI2_TXPTR	SPI slave transmit memory address pointer
0xFC62	r/w	0	SPI2_RXLEN	SPI slave receiver packet length
0xFC63	r/w	0	SPI2_RXPTR	SPI slave receiver memory address pointer
0xFC64	r/w	0	SPI2_RBUF	SPI slave transmitter/receiver ring buffer configuration
0xFC65	r	0	SPI2_RXADDR	SPI slave receiver memory address, 9 bits

SPI2_CFG Bits		
Name	Bits	Description
SPI2_CFG_META	15	SPI slave synhronization configuration
SPI2_CFG_RX_BE	14	Set big endian SPI slave receiver bit order
SPI2_CFG_TX_BE	13	Set big endian SPI slave transmitter bit order

SPI2_CFG_META register enables the use of higher bit rate. If the SPI slave and master are using same clock source this register can be set. The SPI slave synchronization is then made simpler. It is recommended to keep this register reset.

SPI2_CFG_RX_BE and SPI2_CFG_TX_BE are used to reverse bit order in SPI mode. When registers are reset the bits are sent/received lsb bit first (i.e. from 0 to 7). When registers are set the bits are sent/received msb bit first (i.e. from 7 to 0).

SPI2_TXPTR Bits		
Name	Bits	Description
SPI2_TXPTR_SPI_TX_ENA	15	SPI slave transmit enable
SPI2_TXPTR_SPI_RX_ENA	14	SPI slave receive enable
SPI2_TXPTR_BUSY	13	SPI slave transmitter busy
SPI2_TXPTR_PTR[8:0]	8:0	SPI slave transmitter memory address pointer

SPI2_TXPTR_SPI_TX_ENA and SPI2_TXPTR_SPI_RX_ENA are the SPI slave mode enables for transmit and receive. SPI start and stop interrupts are generated even though these registers would be reset. It should be noted that when SPI2_TXPTR_SPI_RX_ENA or SPI2_TXPTR_SPI_TX_ENA is set the receiver/transmitter address pointers must be initialized to a valid data start addresses.

SPI2_TXPTR_BUSY is the SPI slave transmitter busy flag. This flag is set if transmitter is enabled and chip select line is in its active state (low).

SPI2_TXPTR[8:0] is the SPI transmitter memory address pointer. This pointer is loaded with packet start address before transmitter is enabled.

SPI2_RXLEN Bits		
Name	**Bits**	**Description**
SPI2_RXLEN_PMODE	15	Peripheral pin mode select: SPI1 (0) / SPI slave (1)
SPI2_RXLEN_SPIINVCLK	14	SPI slave transmitter clock configuration
SPI2_RXLEN_LEN[9:0]	9:0	SPI slave receiver packet size in bytes

SPI2_RXLEN_SPIMODE register configures the VS1010 gpio1[7-4 pins to SPI slave mode. When register is reset (default state) the pins are usable by SPI1 peripheral.

SPI2_RXLEN_SPIINVCLK selects SPI slave transmitter clock edge. When register is reset the SPI out data is written after falling SPI clock edge. When register is set the data is written after rise edge. With high SPI bit rates (SPI clock > core clock / 6) the rise edge should be used. It should be noted that the SPI slave clock can not exceed core clock / 4 at any time.

SPI2_RXLEN_LEN[9:0] register is loaded with SPI receiver packet length counter when receiver returns from busy state to idle (packet end). Packet length is given in bytes.

SPI2_RXPTR Bits		
Name	**Bits**	**Description**
SPI2_RXPTR_BUSY	13	SPI slave receiver busy
SPI2_RXPTR_ENA	12	SPI slave receiver auto-enable
SPI2_RXPTR_PTR[8:0]	8:0	SPI slave receiver memory address pointer

SPI2_RXPTR_BUSY is a busy flag for SPI slave receiver. The receiver sets the flag when changes its state from idle to busy state.

SPI2_RXPTR_ENA enables an automatic RX after TX mode. This register controls the SPI receiver enable. When register is set the SPI transmit end automatically enables the SPI receiver. Receiver address pointer must be configured before this register is set.

SPI2_RXPTR[8:0] is the SPI slave receiver memory pointer. This pointer is loaded with packet start address before receiver is enabled. When receiver changes it state from idle to busy this register is loaded to memory write address pointer register.

SPI2_RBUF Bits		
Name	**Bits**	**Description**
SPI2_RBUF_TXENA	7	SPI slave transmitter ring buffer enable
SPI2_RBUF_TXCF	6:4	SPI slave transmitter ring buffer configuration
SPI2_RBUF_RXENA	3	SPI slave receiver ring buffer enable
SPI2_RBUF_RXCF	2:0	SPI slave receiver ring buffer configuration

SPI2_RBUF_TXENA and SPI2_RBUF_RXENA are ring buffer enable registers for transmitters and receiver respectively. Ring buffer size is defined with SPI2_RBUF_TXCF and SPI2_RBUF_RXCF registers as explained in next table.

Ring buffer configuration bits				
Name	**CF register**	**Ring buf. size**	**Locked bits**	**Incremented bits**
SPI2_RBUF_TXCF_512W[1]	011	512 words	[10:9]	[8:0]
SPI2_RBUF_TXCF_256W[1]	010	256 words	[10:8]	[7:0]
SPI2_RBUF_TXCF_128W[1]	001	128 words	[10:7]	[6:0]
SPI2_RBUF_TXCF_64W[1]	000	64 words	[10:6]	[5:0]

[1] For the corresponding RX configuration register, use name SPI2_RBUF_RXCF_xxxW instead, where xxx is the ring buffer size.

SPI2_RXADDR register is the current memory address were receiver stores data. This register is loaded with SPI2_RXPTR[10:0] when new packet start is detected in bus.

SPI slave generates an SPI start interrupt each time a new transmission is started by asserting XCS line low. SPI end interrupt is generated when XCS line is asserted high. When ring buffers are used the interrupt is given also when ring buffer address pointer has reached middle or end of the configured buffer size.

9.9.3 AES Peripheral

VS1010 has an AES (Advanced Encryption Standard) decrypting logic. The supported block size is 128 bits.

For more information about using AES contact VLSI Solution.

9.10 USB Peripheral

VS1010 has a Full Speed / Hi-Speed Universal Serial Bus. The Universal Serial Bus Controller handles USB 2.0 data traffic at 12 Mbit/s signalling speed and high speed USB data at 480 Mbit/s. The devices support a maximum of four endpoints.

The USB implementation is based on transceiver macromodel interface (UTMI). Block diagram of usb modules is shown in Figure 10

Simplified UTM module diagram is shown in Figure 11.

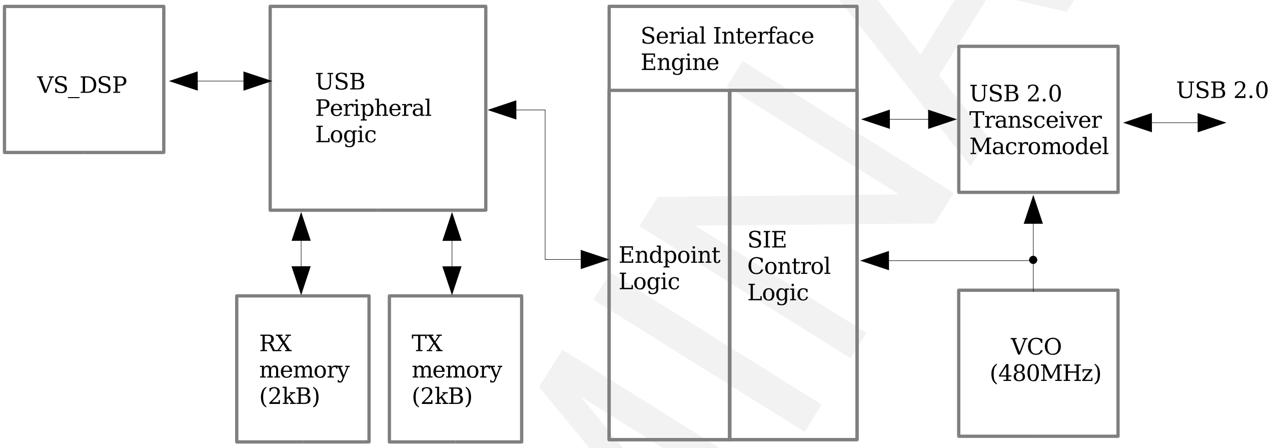


Figure 10: VS1010 USB block diagram.

The USB device can handle traffic for the control endpoint (0) plus three input and output endpoints. Bulk, Isochronous and Interrupt transfer modes are supported at Full Speed (12 Mbit/s). The maximum packet size is 1023 bytes.

4 kilobytes of Y data memory is used as the USB packet buffer: 2 KiB for incoming packets (X:0xF400-0xF7FF) and 2 KiB for outgoing packets (X:0xF800-0xFBFF). The input buffer is a ring buffer with incoming packets consisting of a status word and n data words. The output buffer has 16 possible start locations for outgoing packets at 128-byte (64-address) intervals (note that all data addressing in VS1010 is based on 16-bit words).

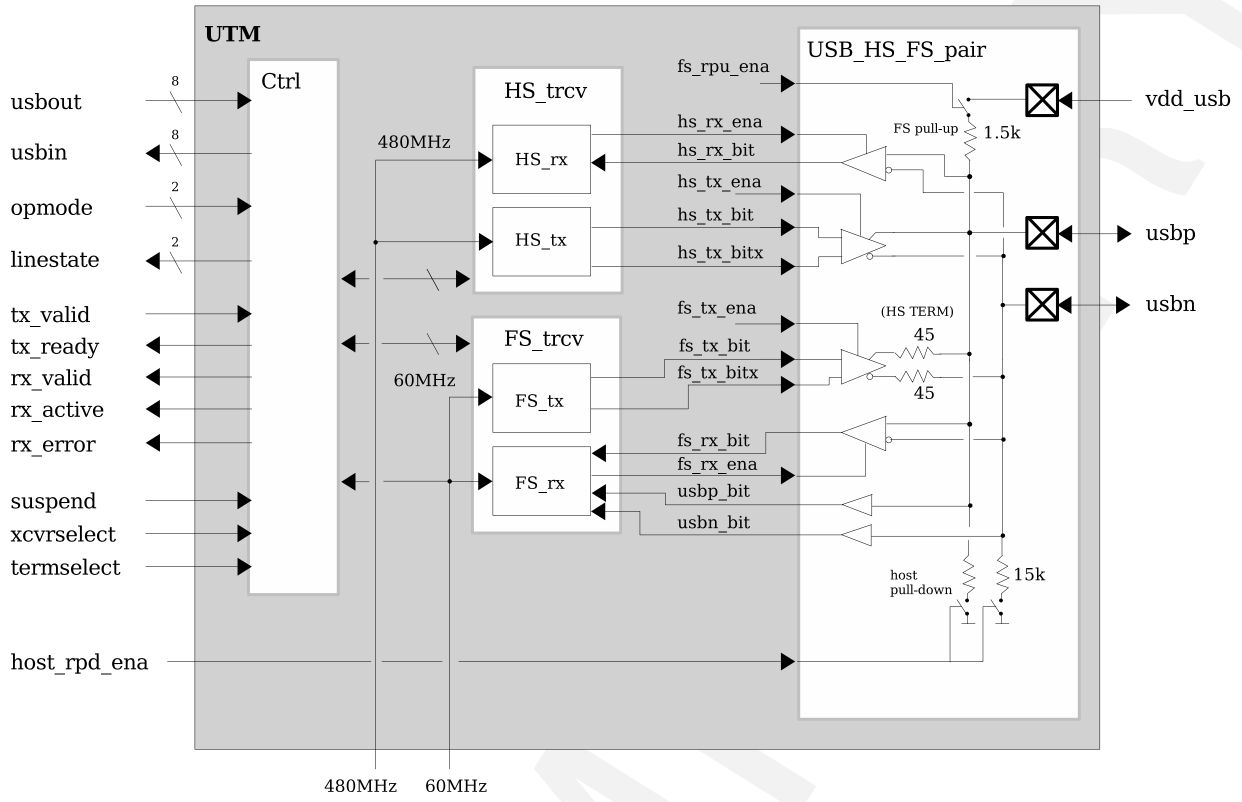


Figure 11: VS1010 UTM functional block diagram.

9.10.1 USB Peripheral Registers

Universal Serial Bus Controller Registers		
Address	**Register**	**Function**
0xFC80	USB_CF	USB Device Config
0xFC81	USB_CTRL	USB Device Control
0xFC82	USB_ST	USB Device Status
0xFC83	USB_RDPTR[9:0]	Receive buffer read pointer
0xFC84	USB_WRPTR[9:0]	Receive buffer write pointer
0xFC85	USB_UTMIR	UTM read control
0xFC86	USB_UTMIW	UTM write control
0xFC87	USB_HOST	Host control
0xFC88	USB_EP_SEND0	EP0IN Transmittable Packet Info
0xFC89	USB_EP_SEND1	EP1IN Transmittable Packet Info
0xFC8A	USB_EP_SEND2	EP2IN Transmittable Packet Info
0xFC8B	USB_EP_SEND3	EP3IN Transmittable Packet Info
0xFC90	USB_EP_ST0	Flags for endpoints EP0IN and EP0OUT
0xFC91	USB_EP_ST1	Flags for endpoints EP1IN and EP1OUT
0xFC92	USB_EP_ST2	Flags for endpoints EP2IN and EP2OUT
0xFC93	USB_EP_ST3	Flags for endpoints EP3IN and EP3OUT

USB_CF Bits		
Name	**Bits**	**Description**
USB_CF_RST	15	Reset Active
USB_CF_HDTOG	14	Reset value of host data toggle (set to 0)
USB_CF_DDTOG	13	Reset value of device data toggle (set to 0)
	12	Reserved, use '0'
USB_CF_NOHIGHSPEED	11	Set to disable high speed functionality.
USB_CF_DTOGERR	10	Data Toggle error control (set to 0)
USB_CF_MASTER	9	Set for master/host mode
USB_CF_RSTUSB	8	Reset receiver (set to 0)
USB_CF_USBENA	7	Enable USB
USB_CF_USBADDR	6:0	Current USB address

USB_CTRL Bits		
Name	**Bits**	**Description**
USB_CTRL_BUS_RESET	15	Interrupt mask for bus reset
USB_CTRL_SOF	14	Interrupt mask for start-of-frame
USB_CTRL_RX	13	Interrupt mask for receive data
USB_CTRL_TX	11	Interrupt mask for transmitter empty (idle)
USB_CTRL_NAK	10	Interrupt mask for NAK packet sent to host
USB_CTRL_TIME	9	Interrupt mask for bus timeout
USB_CTRL_SUSP	8	Interrupt mask for suspend request
USB_CTRL_RESM	7	Interrupt mask for resume request
USB_CTRL_BR_START	6	Interrupt mask for start of bus reset
USB_CTRL_DCON	5	Interrupt mask for usb disconnected
USB_CTRL_CF	0	USB Configured. 0→1 transition loads dtogg-host and dtogg-device

USB_ST Bits		
Name	**Bits**	**Description**
USB_ST_BRST	15	Bus reset occurred
USB_ST_SOF	14	Start-of-frame
USB_ST_RX	13	Receive data
USB_ST_TX_HLD	12	Transmitter holding register empty
USB_ST_TX_EMPTY	11	Transmitter empty (idle)
USB_ST_NAK	10	NAK packet sent to host
USB_ST_TIME	9	Bus time out
USB_ST_SUSPI	8	Device suspended
USB_ST_RES	7	Device resumed
USB_ST_MTERR	6	Bus reset start / USB master toggle error
USB_ST_STAT	5	Device disconnected / Status setup
USB_ST_SPD	4	USB speed
USB_ST_PID	3:0	Packet id / Endpoint number of last rx/tx transaction

The USB_ST_PID can be used mainly for debugging purposes.

USB_RDPTR Bits		
Name	**Bits**	**Description**
USB_RDPTR	9:0	Packet Read Pointer

This buffer marks the index position of the last word that the DSP has successfully read from the receive packet buffer. DSP should control this register and update the position after each packet it has read from the receive buffer. After reset this register is zero.

USB_WRPTR Bits		
Name	**Bits**	**Description**
USB_WRPTR	9:0	Packet Write Pointer

After a packet has been received from the PC, the USB hardware updates this pointer to the receive buffer memory. USB_WRPTR is index location of the next free word location in the USB receive buffer. When USB_RDPTR equals to USB_WRPTR, the packet input buffer is empty. After reset this register is zero.

USB_UTMIR Bits		
Name	**Bits**	**Description**
USB_UTMIR_LSTATE	15:14	USB bus line state
USB_UTMIR_CNT	13:0	USB frame counter, master mode

USB_UTMIW Bits		
Name	**Bits**	**Description**
USB_UTMIW_ORIDE	15	Bus override
	14	Reserved, use ’0’
USB_UTMIW_J	6	Drive chirp J
USB_UTMIW_HSHK	5	Reset handshake
USB_UTMIW_K	4	Drive chirp K
USB_UTMIW_RCVSEL	3	Receiver select
USB_UTMIW_TERMSEL	2	Termination select
USB_UTMIW_OPMOD	1:0	Operation mode

USB_HOST Bits		
Name	**Bits**	**Description**
USB_HOST_PID	15:12	USB host packet id
USB_HOST_ISOC	11	Disable NAK packet send
USB_HOST_TX	9	USB host send packet

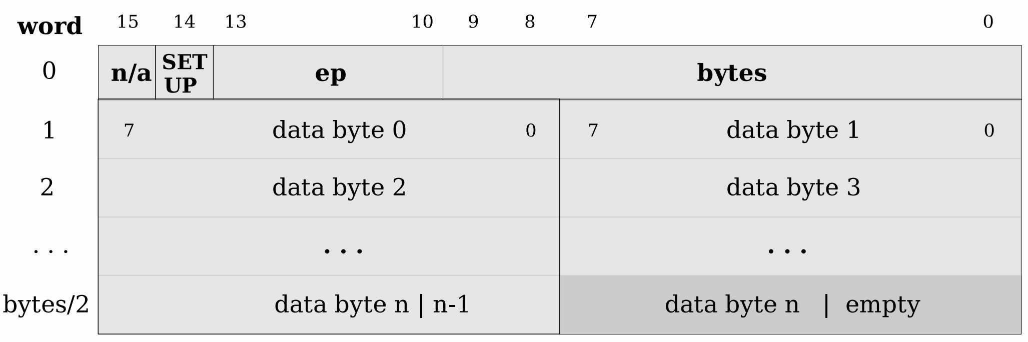
USB_EP_SENDn Bits		
Name	**Bits**	**Description**
USB_EP_SEND_TXR	15	Packet ready for transmission
USB_EP_SEND_ADDR	13:10	Starting location of packet
USB_EP_SEND_LEN	9:0	Length of packet in bytes (0..1023)

When the DSP has written a packet into the transmit buffer, that is ready to be transmitted to the PC by an endpoint, the DSP signals the USB firmware by setting the value of the USB_EP_SENDn register of the endpoint that should transmit the packet (USB_EP_SEND0 for endpoint 0, USB_EP_SEND1 for endpoint 1 etc).

USB_EP_STn Bits		
Name	**Bits**	**Description**
EPnOUT (PC → Device) endpoint (0 .. 3) flags		
USB_EP_ST_OTYP	15:14	00=bulk 01=interrupt 11=isochronous
USB_EP_ST_OENA	13	1=enabled 0=disabled
USB_EP_ST_OSTL	12	Force STALL
USB_EP_ST_OSTL_SENT	11	At least 1 STALL sent
reserved	10:8	Use ’0’
EPnIN (Device → PC) endpoint (0 .. 3) flags		
USB_EP_ST_ITYP	7:6	00=bulk 01=interrupt 11=isochronous
USB_EP_ST_IENA	5	1=enabled 0=disabled
USB_EP_ST_ISTL	4	Force STALL
USB_EP_ST_ISTL_SENT	3	At least 1 STALL sent
USB_EP_ST_INAKSENT	2	At least 1 NAK sent
USB_EP_ST_IXMIT_EMP	1	Transmitter empty
reserved	0	Use ’0’

9.10.2 USB Receiver

The received packet is stored to 2kB USB receive memory in the format shown in Figure 12.



n/a : not available (ignore)
SETUP : 0 = OUT, 1= SETUP packet
ep : endpoint number
bytes : length of payload

Figure 12: USB packet format

9.10.3 USB Clocking Modes

USB usage requires a special clock setup. The core clock must be set to 60 MHz. If only Full Speed USB is used the 60 MHz clock can be achieved byt placing the PLL to 5x clocking mode and using 12.000 MHz XTAL. When Hi-Speed USB is used the core clock must also be 60 MHz but this clock is generated with a PLL which can be programmed with fractional multiplier factors. The xtal oscillator frequencies of 12.000 MHz or 12.288 MHz are recommended in this mode.

9.10.4 USB Host

USB module can be configured as an USB host. In USB host mode the 1.5kOhm pull up resistor in D+ pin is replaced with 15kOhm pull down resistors in in both the D+ and D- pins.

USB host is capable of:

- Send Start of Frame (SOF) packets
- Send SETUP, IN and OUT packets
- Schedule transfers within 1ms frames
- Signal USB bus reset
- Provide USB power management

9.11 Interruptable General Purpose IO Ports 0-2

VS1010 has 3 general purpose IO ports that can operate either in GP mode or in perip mode. In order to use pins as gpio the GPIOx_MODE registers must be reset (default value).

GPIO port 0, 1 and register offsets are 0xFCA0, 0xFCC0 and 0xFCE0 accordingly. GPIO port 0 is 11 bits, gpio 1 is 15 bits wide and GPIO port 2 is 16-bits wide.

Interruptable General I/O Registers, Prefix GPIOx_				
Reg	**Type**	**Reset**	**Abbrev**	**Description**
0	r/w	0	DDR	Data direction
1	r/w	0	ODATA	Data output
2	r	0	IDATA	Data input (I/O pin state)
3	r/w	0	INT_FALL	Falling edge interrupt enable
4	r/w	0	INT_RISE	Rising edge interrupt enable
5	r/w	0	INT_PEND	Interrupt pending source
6	w	0	SET_MASK	Data set (→ 1) mask
7	w	0	CLEAR_MASK	Data clear (→ 0) mask
8	r/w	0	BIT_CONF	Bit engine config 0 and 1
9	r/w	0	BIT_ENG0	Bit engine 0 read/write
10	r/w	0	BIT_ENG1	Bit engine 1 read/write

GPIOx_DDR register configure the directions of each of the 16 I/O pins. A bit set to 1 in the DDR turns the corresponding I/O pin to output mode, while a bit set to 0 sets the pin to input mode. The register is set to all zeros in reset, i.e. all pins are inputs by default. The current state of the DDR can also be read.

GPIOx_ODATA register sets the GPIO pin high or low. Only pins that are configured as outputs are affected.

GPIOx_IDATA monitiors the current state of a pin. The actual logical levels of the I/O pins are seen in the input data register. Note: The pin state can be read even if the pin is in peripheral mode (i.e. GPIOx_MODE[y] is set).

GPIOx_INT_RISE and GPIOx_INT_FALL configures an interrupt trigger edge. If a bit of the falling edge interrupt enable register (GPIOx_INT_FALL) is set to 1, a falling edge in the corresponding pin (even when configured as output) will set the corresponding bit in the interrupt pending source register (GPIOx_INT_PEND).

If a bit of the rising edge interrupt enable register (GPIOx_INT_RISE) is set to 1, a rising edge in the corresponding pin (even when configured as output) will set the corresponding bit in the interrupt pending source register (GPIOx_INT_PEND).

GPIOx_INT_PEND defines the source of a pending interrupt. If any of the bits in the interrupt pending source register (GPIOx_INT_PEND) are set, an interrupt request is generated. Bits in GPIOx_INT_PEND can be cleared by writing a 1-bit to the bit that is to be cleared.

Note: the interrupt request will remain asserted until all GPIOx_INT_PEND bits are cleared.

GPIOx_SET_MASK register can be used to mask GPIO pins high when GPIOx_ODATA register is written. All bits that are set in the mask register also set the corresponding bit in the data output register. Other bits retain their old values.

GPIOx_CLEAR_MASK register can be used to mask GPIO pins low when GPIOx_ODATA register is written. All bits that are set in the mask clear the corresponding bit in the data output register. Other bits retain their old values.

GPIOx_BIT_CONF is a bit engine configuration register and selects a mapping between an I/O bit and a data output/input register bit for each of the bit engine registers.

GPIOx_BIT_CONF Bits		
Name	**Bits**	**Description**
GPIO_BE_DAT1	15:12	Data bit selection (0..15) for bit engine 1
GPIO_BE_IO1	11:8	I/O bit selection (0..15) for bit engine 1
GPIO_BE_DAT0	7:4	Data bit selection (0..15) for bit engine 0
GPIO_BE_IO0	3:0	I/O bit selection (0..15) for bit engine 0

GPIOx_BIT_ENG0 is a register used to read/write a GPIO pin specified in GPIOx_BIT_CONF register.

When writing a value to the bit engine 0 register, the data bit specified in the configuration register is copied to the data output register bit specified in the same register.

When reading a value from the bit engine 0 register, the data input register bit specified in the configuration register is copied to the data bit specified in the same register, other bits read out as 0.

GPIOx_BIT_ENG1 works just like GPIOx_BIT_ENG0.

9.12 S/PDIF Peripheral

9.12.1 S/PDIF Receiver

S/PDIF receiver interface offers a receiver function for serial digital audio. S/PDIF supports two channels which are multiplexed in one signal line. Synchronizing to S/PDIF input data bit frequency is done by the digital frequency divider the clock of which is generated by the low jitter programmable PLL. Supported sampling frequencies are 32.0 kHz, 44.1 kHz, 48.0 kHz, 96.0 kHz and 192.0 kHz.

S/PDIF Receiver peripheral device supports linear PCM sample recovery up to 24 bits, S/PDIF subframe parity check, biphase channel coding check, subframe, frame, and block integrity checks, and read miss notification. This version does not perform cyclic redundancy check (CRC) for channel status bits in hardware. CRC check can be implemented by software if needed.

Frame format is depicted in Figure 13. X, Y, and Z are the allowed preambles of a subframe. An X subframe and an Y subframe constitute a frame. X preamble is replaced by Z preamble every 192 frames to indicate block limit.

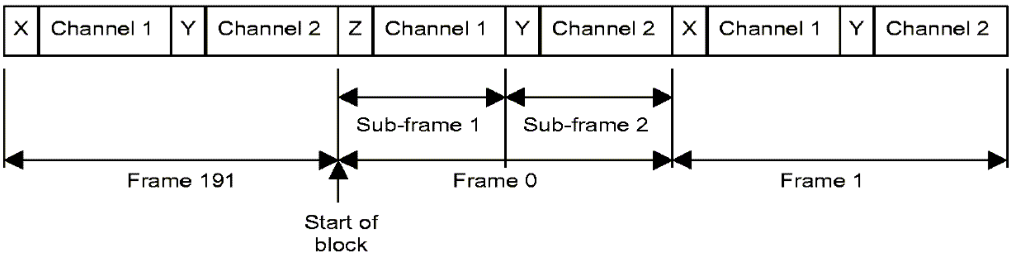


Figure 13: S/PDIF frame format.

Subframe format is depicted in Figure 14. A Preamble is a signal pattern lasting 4 time slots. S/PDIF Receiver decodes it and keeps track of frame and block integrity. A payload is max 24-bit sample word. Validity bit indicates whether the payload is valid audio sample. User data bit allows simultaneous data send. Channel information is conveyed in channel status bits as specified in IEC 60958-1 and IEC 60958-3. S/PDIF Receiver peripheral device uses the parity bit to calculate parity check. The result is shown in SP_CTL register bits LPerr and RPerr. Each bit occupies one time slot of the subframe.

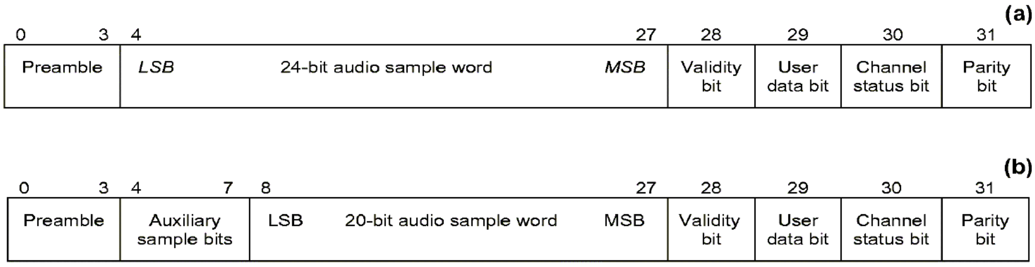


Figure 14: S/PDIF sub-frame format.

9.12.2 S/PDIF Receiver Registers

The base address for S/PDIF Receiver interface registers is Y:0xFD00.

S/PDIF Receiver Registers				
Address	**Type**	**Reset**	**Abbrev**	**Description**
0xFD00	r/w	0	SP_RX_CF	S/PDIF control and status register
0xFD01	r/w	0	SP_RX_CLKDIV	S/PDIF receiver clock divider register
0xFD02	r	0	SP_LDATA_LSB	S/PDIF input left input channel, bits 8-0
0xFD03	r	0	SP_LDATA	S/PDIF left input channel, bits 23-8
0xFD04	r	0	SP_RDATA_LSB	S/PDIF left input channel, bits 8-0
0xFD05	r	0	SP_RDATA	S/PDIF right input channel, bits 23-8
0xFD06	r/w	0	SP_RX_STAT	S/PDIF status register
0xFD07	r	0	SP_RX_BLFRCNT	S/PDIF frame status register

SP_RX_CF Bits		
Name	**Bits**	**Description**
SP_RX_CF_EN	3	S/PDIF receiver enable
SP_RX_CF_INT_ENA	1	Interrupt enable

SP_RX_CF_EN Enables S/PDIF Receiver peripheral. If disabled, i.e. ’0’, most of the peripheral is reset and synchronisation to S/PDIF stream is lost and must be re-acquired after enabling.

SP_RX_CF_INT_ENA, when set, enables S/PDIF receiver interrupt.

SP_RX_CLKDIV Bits		
Name	**Bits**	**Description**
SP_RX_CLKDIV	7:0	Receiver clock divider

SP_RX_CLKDIV is an 8-bit clock divider value that is used to adjust the S/PDIF Receiver peripheral to proper Fs according to master clock frequency. Default value is 8, resulting to Fs = 48 kHz with master clock = 24.576 MHz. Values smaller than 4 are not allowed, since at least 4 samples per audio sample are needed (2 samples per biphase mark).

S/PDIF Receiver peripheral supports audio sampling frequencies up to 192 kHz.

The supported frequencies and corresponding bit rates are summarized in the following table. Bit rate is sampling frequency multiplied by 64, which is channel number (2) times subframe time slot count (32).

While the divider value should be targeted to bit rate of the table below, the peripheral actually operates with quadruple clock rate. This must be accounted for in the system clocking design. The system clock must be at least four (4) times the bit rate if S/PDIF peripheral is to be used. In other words, SP_CF_DIV values less than four (< 4) are forbidden. Divider must be even number.

S/PDIF Frequencies		
Fs	**bit rate (Fs x 64)**	**Minimum system clock rate (4 x bit rate)**
22.05 kHz	1.4112 MHz	5.6448 MHz
24 kHz	1.536 MHz	6.144 MHz
32 kHz	2.048 MHz	8.192 MHz
44.1 kHz	2.8224 MHz	11.2896 MHz
48 kHz	3.072 MHz	12.288 MHz
96 kHz	6.144 MHz	24.576 MHz
192 kHz	12.288 MHz	49.152 MHz

$Divider = Master\,clock\,/\,bit\,rate,$

$Divider > 3$, even number.

SP_RX_LDATA, SP_RX_LDATA_LSB, SP_RX_RDATA and SP_RX_RDATA_LSB registers are received data registers. S/PDIF data is 24 bits and it is divided in two registers. 16 MSB bits are in registers SP_RX_LDATA and SP_RX_RDATA. The remaining 8 LSB bits are in registers SP_RX_LDATA_LSB and SP_RX_RDATA_LSB.

SP_RX_STAT Bits			
Name	**Bits**	**type**	**Description**
SP_RX_STAT_CHSCH	15	r/w	Channel Status Change
SP_RX_STAT_FRCV	14	r	Frame receive
N/A	13		always zero
SP_RX_STAT_MISS	12	r/w	Missed reading previous frame
SP_RX_STAT_BERR	11	r/w	Block error, Z preamble every 192 frames failure
SP_RX_STAT_FERR	10	r/w	Frame error, Y preamble after (X or Z) failure
SP_RX_STAT_SFERR	9	r/w	Subframe error, subframe /= 28 bits
SP_RX_STAT_BIPHERR	8	r/w	Biphase coding error
SP_RX_STAT_RPERR	7	r/w	Parity error, right channel
SP_RX_STAT_LPERR	6	r/w	Parity error, left channel
SP_RX_STAT_RV	5	r	Validity bit, right channel
SP_RX_STAT_RU	4	r	User data bit, right channel
SP_RX_STAT_RC	3	r	Channel status bit, right channel
SP_RX_STAT_LV	2	r	Validity bit, left channel
SP_RX_STAT_LU	1	r	User data bit, left channel
SP_RX_STAT_LC	0	r	Channel status bit, left channel

SP_RX_STAT_CHSCH is a poll bit for channel status change interrupt.

SP_RX_STAT_FRCV is set by the peripheral when a frame is received, and cleared when SP_RX_LDATA is read.

SP_RX_STAT_MISS bit is set if SP_RX_STAT_FRCV is set and new samples are written to SP_RX_LDATA and SP_RX_RDATA. The time to read the samples is a few clock cycles less than the sampling period.

SP_RX_STAT_BERR is set if the period between Z-preambles is not equal to 192 frames.

SP_RX_STAT_FERR is set if Y-preamble does not follow X-preamble or Z-preamble.

SP_RX_STAT_SFERR is set if the previous subframe has not been equal to 32 time slots.

SP_BIPHERR is set if biphase coding of the S/PDIF channel is compromised.

SP_RX_STAT_RPERR and SP_RX_STAT_LPERR are set if the parity count is failed in the respective subframe.

SP_RX_STAT_MISS, SP_RX_STAT_BERR, SP_RX_STAT_FERR, SP_RX_STAT_SFERR, SP_BIPHERR, SP_RX_STAT_RPERR, and SP_RX_STAT_LPERR are "sticky" bits, i.e. if set they keep their state until cleared by sw.

SP_RX_STAT_RV and SP_RX_STAT_LV are validity bits for right channel and left channel, restectively. When validity bit is '0', sample word is a valid PCM sample.

SP_RX_STAT_RU and SP_RX_STAT_LU are user data bits. User data bits should be used as specified in IEC 60958-3.

SP_RX_STAT_RC and SP_RX_STAT_LC are channel status bits. According to the S/PDIF standard, both channels should convey the same bits. Again, for full description of channel status bits, refer to IEC 60958-3.

SP_RX_BLFRCNT Bits			
Name	Bits	Type	Description
SP_TX_FRCNT	15:8	r	Transmitter frame count
SP_RX_FRCNT	7:0	r	Receiver frame count

SP_TX_FRCNT is transmitter frame counter. This counter is incremented each time a new frame is transmitted. Counter counts from 0 to 191. It is reset after the 191th frame is received.

SP_RX_FRCNT is receiver frame counter. Counter counts from 0 to 191. It is reset with every Z-preamble and incremented with every X-preamble.

S/PDIF Receiver issues an interrupt when it has received a frame. When interrupt occurs the channel status bits are updated to the S/PDIF status register. Software must validate the status of the received samples according to the status bits.

9.12.3 S/PDIF Receiver Sample Rate Estimation

S/PDIF receiver sample rate can be estimated with a pulse width counter which tracks the minimum input signal low and high time in clock cycles.

S/PDIF Receiver Registers				
Address	Type	Reset	Abbrev	Description
0xFD0C	r/w	0xFFFF	SP_RX_PW[15:8][7:0]	Pulse width counter for sample rate estimation

SP_RX_PW Bits		
Name	Bits	Description
SP_RX_PW_CNT	15:8	Current value of pulse width counter
SP_RX_PW_MIN	7:0	Minimum pulse width in clock cycles

SP_RX_PW_CNT register counts the pulse high/low time after each change in the S/PDIF input stream. The value is saturated to 0xFF if no change in signal was detected. SP_RX_PW_MIN register logs the minimum pulse width time in clock cycles after the counter was started. This register is initialized to 0xFF when the counter is disabled. This register can also be initialized by software during the pulse width counting.

SP_RX_CF Bits		
Name	**Bits**	**Description**
SP_RX_CF_PWCNT_ENA	0	Pulse width counter enable

SP_RX_CF_PWCNT_ENA enables the pulse width counter. This counter can be used to estimate the input sample rate. The pulse width counter can be used separately from the S/PDIF receiver (i.e. SP_RX_CF_EN register has no effect on the counter operation.)

Input sample rate is

$$FS_{sprx} = \frac{Core\ clock}{(128 \times SP\_RX\_PW\_MIN)}$$

9.12.4 S/PDIF Transmitter

S/PDIF is a serial digital audio transfer standard. Sampling frequencies up to 192 kHz and sample word width of 16 - 24 bits are supported for two channels. S/PDIF transmitter peripheral has a processor interface and one external output signal for digital audio. S/PDIF is described in IEC 60958-1 and IEC 60958-3. Standard connectors are defined in IEC 60268-11:1987 although commercial products feature a variety of connectors both electrical and optical.

The speed of the S/PDIF transmitter depends on the sampling frequency of the audio signal. Since S/PDIF signal is often used to retrieve a clock signal at the receiving end, S/PDIF transmitter must produce an exact frequency with a very low jitter.

Supported sampling frequencies are 32 kHz, 48 kHz, 96 kHz and 192 kHz when master clock frequency is n x 12.288 MHz. 44.1 kHz sampling frequency is supported.

9.12.5 S/PDIF Transmitter Registers

S/PDIF supports audio sample width of 16 to 24 bits. The exact figure is conveyed to the receiver by channel status bits. If the the transmitted sample word is less than 24 bits wide, the remaining LSB's must be zero.

Channel status registers provide interface to the S/PDIF standard implementation channel status bits. The S/PDIF Transmitter inserts the corresponding bits to their proper places in the transfer frame. Channel status data (byte 23) for cyclic redundancy check character (CRCC) is not tested yet.

This document offers a terse description of the channel status bits. Full coverage in IEC 60958-3 is mandatory. Current implementation shares Channel Status Data bits (registers CHS0 and CHS1) for both channels!

S/PDIF Transmitter Registers				
Reg	**Type**	**Reset**	**Abbrev**	**Description**
0xFD02	w	0	SP_LDATA_LSB	Left channel Audio sample bits 7-0
0xFD03	w	0	SP_LDATA	Left channel Audio sample bits 23-8
0xFD04	w	0	SP_RDATA_LSB	Right channel Audio bits sample 7-0
0xFD05	w	0	SP_RDATA	Right channel Audio sample bits 23-8
0xFD08	r/w	0	SP_TX_CHST0	Channel Status 0
0xFD09	r/w	0	SP_TX_CHST1	Channel Status 1
0xFD0A	r/w	0	SP_TX_CHST2	Channel Status 2
0xFD0B	r/w	0x40	SP_TX_CF	Transmitter configuration

SP_TX_LDATA, SP_TX_LDATA_LSB, SP_TX_RDATA and SP_TX_RDATA_LSB registers are transmitter data registers. S/PDIF data is 24 bits and it is divided in two registers. 16 MSB bits are in registers SP_TX_LDATA and SP_TX_RDATA. The remaining 8 LSB bits are in registers SP_TX_LDATA_LSB and SP_TX_RDATA_LSB.

Channel Status SP_TX_CHST0			
Name	**Bits of data word**	**Bits of Channel status**	**Description**
SP_TX_CHST0_CAT	15:8	15:8	Category Code
SP_TX_CHST0_MD0	7:6	7:6	PCM Mode 0
SP_TX_CHST0_PCMM	5:3	5:3	Linear PCM Mode
SP_TX_CHST0_CP	2	2	Copyright
SP_TX_CHST0_PCM	1	1	Linear PCM
SP_TX_CHST0_PROCON	0	0	Professional/Consumer mode

SP_TX_CHST0_CAT indicates to which category the device belongs. Default value is “00000000”.

The default value of SP_TX_CHST0_MD0 is “00”. No other states are defined yet.

When SP_TX_CHST0_PCM is ’0’, SP_TX_CHST0_PCMM selects linear PCM mode. The default value is “000” which corresponds to 2 audio channels without pre-emphasis.

SP_TX_CHST0_CP is a copyright bit. When ’0’, copyright for current stream is asserted.

SP_TX_CHST0_PCM is ’0’ when the audio sample word is linear PCM.

SP_TX_CHST0_PROCON is ’0’ in S/PDIF defining consumer usage. If this bit is ’1’, channel is for professional use and the interface would be called AES/EBU. However, the channel status bits would be different in this case.

Channel Status SP_TX_CHST1			
Name	**Bits of data word**	**Bits of Channel status**	**Description**
-	15:14	31:30	Not specified, “00”
SP_TX_CHST1_CLKA	13:12	29:28	Clock Accuracy
SP_TX_CHST1_FS	11:8	27:24	Sampling Frequency
SP_TX_CHST1_CH	7:4	23:20	Channel Number
SP_TX_CHST1_SRC	3:0	19:16	Source Number

SP_TX_CHST1_CLKA indicates the level of clock accuracy the S/PDIF transmitter is capable of providing to its output.

The sampling frequency of the audio sample stream is defined in SP_TX_CHST1_FS.

SP_TX_CHST1_CH is the number of channels in the transmission. “0011” indicates two channel stereo format.

SP_TX_CHST1_SRC is the number of sources. “0000” is defined as “do not take into account”.

Channel Status SP_TX_CHST2			
Name	**Bits of data word**	**Bits of Channel status**	**Description**
SP_TX_CHST2_ST_NWRQ	13		New Word Request (read only bit)
SP_TX_CHST2_TX_ENA	12		Transmitter enable
SP_TX_CHST2_RS1_RU	11		User Data bit, right channel
SP_TX_CHST2_RS1_RV	10		Validity bit, right channel
SP_TX_CHST2_LS1_RU	9		User Data bit, left channel
SP_TX_CHST2_LS1_RV	8		Validity bit, left channel
SP_TX_CHST2_CH2_FSO	7:4	39:36	Original Sampling Frequency
SP_TX_CHST2_CH2_WRDL	3:1	35:33	Sample Word Length
SP_TX_CHST2_CH2_WRDLM	0	32	Maximum Sample Word Length

SP_TX_CHST2_ST_NWRQ bit is set when new sample words must be written to sample word registers. It is cleared when SP_TX_CHST2_TX_LDATA is written. Hence, SP_TX_CHST2_ST_NWRQ has the same function as S/PDIF Interrupt, but this bit is not controlled by SP_TX_CHST2_CF_IE.

SP_TX_CHST2_TX_ENA is the S/PDIF transmit enable. Transmitter is enabled when this register is set.

SP_TX_CHST2_RS1_RU is a user data bit for the right channel. Default value is ’0’. User data bits can be used to convey an application specific message to the receiver. Some equipment categories dictate the message, see IEC 60958-3.

SP_TX_CHST2_RS1_RV is the validity bit of the right channel sample word. If the audio sample word is not a linear PCM, this bit must be set.

SP_TX_CHST2_LS1_LU is a user data bit for the left channel. Default value is ’0’. User data bits can be used to convey an application specific message to the receiver. Some equipment categories dictate the message, see IEC 60958-3.

SP_TX_CHST2_LS1_LV is the validity bit of the left channel sample word. If the audio sample word is not a linear PCM, this bit must be set.

SP_TX_CHST2_CH2_FSO defines the original sampling frequency of the audio stream. “0000” means the original sampling frequency is not indicated (default).

In SP_TX_CHST2_CH2_WRDL, the sample word length is coded with respect to SP_TX_CHST2_CH2_WRDI
“000” means the word length is not indicated.

SP_TX_CHST2_CH2_WRDLM indicates whether the maximum word length is 24 bits (’1’) or 20 bits (’0’).

S/PDIF TX Configuration SP_TX_CF		
Name	**Bits**	**Description**
SP_TX_CF_CLKDIV	15:2	Clock divider
SP_TX_CF_IE	1	Interrupt enable
SP_TX_CF_SND	0	Send words

SP_TX_CF_CLKDIV contains a clock divider value that is used to generate S/PDIF Transmitter operating frequency. The target is twice the bit rate. Bit rate is sampling frequency of the transmitted signal multiplied by 64. For example, 48 kHz audio signal requires bit rate of 3.072 MHz and consequent clock frequency for the peripheral is 6.144 MHz. Default value for SP_TX_CF_CLKDIV is 4, resulting to Fs = 48 kHz when master clock frequency is 24.576 MHz. Zero is forbidden value.

S/PDIF Transmitter frequencies		
Fs	**bit rate (Fs x 64)**	**Target frequency for clock divider**
32 kHz	2.048 MHz	4.096 MHz
44.1 kHz	2.8224 MHz	5.6448 MHz
48 kHz	3.072 MHz	6.144 MHz
96 kHz	6.144 MHz	12.288 MHz
192 kHz	12.288 MHz	24.576 MHz

Divider = Master clock / Targer frequency, Divider = Master clock / (Fs * 64 * 2).

SP_TX_CF_IE, when ’1’, enables processor interrupt request when new values must be written for the sample word registers: SP_LDATA and SP_RDATA. Default is ’0’.

SP_TX_CF_SND, when ’1’, S/PDIF Transmitter sends the data in the sample word registers. Otherwise only empty subframes with zero payload will be sent. This is because the receiver may use S/PDIF signal as a clock source and hence, the S/PDIF signal must not stop even though no data is sent.

The S/PDIF Transmitter has one interrupt. Interrupt request is issued when SP_ST_NWRQ is set, i.e. when new sample words must be written to the sample word registers.

9.13 UART Peripherals

VS1010 has two UART peripherals. RS232 UART implements a serial interface using rs232 standard.

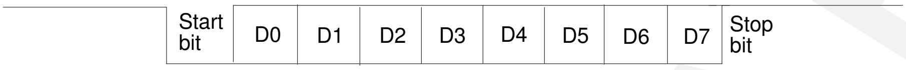


Figure 15: RS232 serial interface protocol

When the line is idling, it stays in logic high state. When a byte is transmitted, the transmission begins with a start bit (logic zero) and continues with data bits (LSB first) and ends up with a stop bit (logic high). 10 bits are sent for each 8-bit byte frame.

9.13.1 UART Peripheral Registers

UART0 Registers				
Reg	**Type**	**Reset**	**Abbrev**	**Description**
0xFE00	r	0	UART0_STATUS[4:0]	Status
0xFE01	r/w	0	UART0_DATA[7:0]	Data
0xFE02	r/w	0	UART0_DATAH[15:8]	Data High
0xFE03	r/w	0	UART0_DIV	Divider

UART1 Registers				
Reg	**Type**	**Reset**	**Abbrev**	**Description**
0xFE10	r	0	UART1_STATUS[4:0]	Status
0xFE11	r/w	0	UART1_DATA[7:0]	Data
0xFE12	r/w	0	UART1_DATAH[15:8]	Data High
0xFE13	r/w	0	UART1_DIV	Divider

UART_STATUS register monitors the UART status.

UART_STATUS Bits		
Name	**Bits**	**Description**
UART_ST_FRAMERR	4	Framing Error (stop bit was 0)
UART_ST_RXORUN	3	Receiver overrun
UART_ST_RXFULL	2	Receiver data register full
UART_ST_TXFULL	1	Transmitter data register full
UART_ST_TXRUNNING	0	Transmitter running

UART_ST_FRAMERR is set at the time of stop bit reception. When reception is functioning normally, stop bit is always “1”. If, however, “0” is detected at the line input at the stop bit time, UART_ST_FRAMERR is set to “1”. This can be used to detect “break” condition in some protocols.

UART_ST_RXORUN is set if a received byte overwrites unread data when it is transferred from the receiver shift register to the data register, otherwise it is cleared.

UART_ST_RXFULL is set if there is unread data in the data register.

UART_ST_TXFULL is set if a write to the data register is not allowed (data register full).

UART_ST_TXRUNNING is set if the transmitter shift register is in operation.

UART_DATA is the uart data register. A read from UART_DATA returns the received byte in bits 7:0, bits 15:8 are returned as '0'. If there is no more data to be read, the receiver data register full indicator will be cleared.

A receive interrupt will be generated when a byte is moved from the receiver shift register to the receiver data register.

A write to UART_DATA sets a byte for transmission. The data is taken from bits 7:0, other bits in the written value are ignored. If the transmitter is idle, the byte is immediately moved to the transmitter shift register, a transmit interrupt request is generated, and transmission is started. If the transmitter is busy, the UART_ST_TXFULL will be set and the byte remains in the transmitter data register until the previous byte has been sent and transmission can proceed.

UART_DATAH is the same register as the UART_DATA, except that bits 15:8 are used.

UART_DIV register configures uart tansmission speed.

UART_DIV Bits		
Name	**Bits**	**Description**
UART_DIV_D1	15:8	Divider 1 (0..255)
UART_DIV_D2	7:0	Divider 2 (6..255)

The divider is set to 0x0000 in reset. The ROM boot code must initialize it correctly depending on the master clock frequency to get the correct bit speed. The second divider (D_2) must be from 6 to 255.

The communication speed $f = \frac{f_m}{(D_1+1)\times(D_2)}$, where f_m is XTALI, and f is the TX/RX speed in bps.

9.14 Watchdog Peripheral

The watchdog consist of a watchdog counter and some logic. After reset, the watchdog is inactive. The counter reload value can be set by writing to WDOG_CF. The watchdog is activated by writing 0x4ea9 to register WDOG_KEY. Every time this is done, the watchdog counter is reset. Every 65536'th clock cycle the counter is decremented by one. If the counter underflows, it will activate vsdsp's internal reset sequence.

Thus, after the first 0x4ea9 write to WDOG_KEY, subsequent writes to the same register with the same value must be made no less than every $65536 \times$WDOG_CF clock cycles.

Once started, the watchdog cannot be turned off. Also, a write to WDOG_CF doesn't change the counter reload value.

After watchdog has been activated, any read/write operation from/to WDOG_CF or WDOG_DUMMY will invalidate the next write operation to WDOG_KEY. This will prevent runaway loops from resetting the counter, even if they do happen to write the correct number. Writing an incorrect value to WDOG_KEY will also invalidate the next write to WDOG_KEY.

Reads from watchdog registers return undefined values.

9.14.1 Watchdog Registers

Watchdog Registers				
Reg	**Type**	**Reset**	**Abbrev**	**Description**
0xFE20	w	0	WDOG_CF	Configuration
0xFE21	w	0	WDOG_KEY	Clock configuration
0xFE22	w	0	WDOG_DUMMY[-]	Dummy register

9.15 I2S Peripheral

VS1010 has a bi-directional I2S digital interface. I2S is a serial audio interface which uses serial bit clock (i2s_bck), frame sync (i2s_frm) and serial data line (i2s_dout, i2s_din) to transfer data. I2S frame consists of left and right data which is transmitted left word first and MSB bit first. Data is latched out at falling edge of bit clock and latched in at rising edge of bit clock. I2S data format is shown in Figure 16.

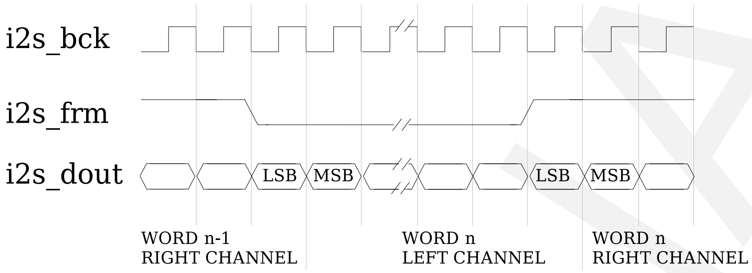


Figure 16: I2S frame format

9.15.1 I2S Peripheral Registers

I2S Registers				
Reg	**Type**	**Reset**	**Abbrev**	**Description**
0xFE60	r/w	0x0001	I2S_CF[13:0]	Configuration and status register
0xFE61	r/w	0	I2S_LEFT_LSB	Left data bits[15:0]
0xFE62	r/w	0	I2S_LEFT	Left data bits[31:16]
0xFE63	r/w	0	I2S_RIGHT_LSB	Right data bits[15:0]
0xFE64	r/w	0	I2S_RIGHT	Right data bits[31:16]

I2S_CF Bits		
Name	**Bits**	**Description**
I2S_CF_32B[1]	13	32-bit mode (1) / 16-bit mode (0) select
I2S_CF_INTENA[1]	12	I2S peripheral interrupt enable
I2S_CF_RXRFULL	11	Receiver right data register full
I2S_CF_RXLFULL	10	Receiver left data register full
I2S_CF_RXORUN	9	Receiver over run flag
I2S_CF_TXRFULL	8	Transmitter right data register full
I2S_CF_TXLFULL	7	Transmitter left data register full
I2S_CF_TXURUN	6	Transmitter under run flag
I2S_CF_MODE[1]	5	I2S output mode: DSP (1) or SRC (0) out
I2S_CF_FS[1:0][1]	4:3	I2S sample rate selection
I2S_CF_ENA	2	I2S peripheral enable
I2S_CF_ENAMCK[1]	1	I2S master clock (12 MHz) pad driver enable
I2S_CF_MASTER[1]	0	I2S master (1) / slave (0) mode select

[1] Value can only be changed if I2S_CF_ENA has previously been cleared to 0.

I2S_CF_MASTER bit is used to select between master (1) and slave (0) modes. In master mode the VS1010 generates bit clock and frame sync signals. In slave mode the external I2S master generates the clock and sync signals.

I2S_CF_ENAMCK is the 12 MHz output clock enable signal. It can be used to clock external I2S circuitry. This clock is the same clock as the XTALI oscillator clock of VS1010.

I2S_CF_ENA is the transmitter and receiver enable signal. When set the receiver and transmitter enter the active state. Other fields of the same register (I2S_CF_32B, I2S_CF_INTENA, I2S_CF_MODE, I2S_CF_FS, I2S_CF_ENAMCK, and I2S_CF_MASTER) can only be changed if I2S_CF_ENA is 0.

I2S_CF_FS register is used to set the I2S peripheral sample rate. This register can be modified only when I2S is in idle state, i.e. I2S_CF_ENA is reset. Next table lists the sample rates when XTALI = 12.288 MHz is used.

I2S Sample Rates		
I2S_CF_FS[1:0]	**16-bit mode**	**32-bit mode**
11	48 kHz	24 kHz
10	192 kHz	96 kHz
01	96 kHz	48 kHz
00	48 kHz	24 kHz

I2S_CF_MODE register selects between DSP mode (1) and SRC mode (0). In DSP mode the data is transferred from registers I2S_LEFT, I2S_LEFT_LSB, I2S_RIGHT and I2S_RIGHT_LSB. In SRC mode which is the default data is sampled from DAC's SRC filter and I2S is operating in master mode only.

I2S_CF_TXURUN is the transmitter under run flag register. It is set if left or right data buffer register was empty as it was copied to shifter register.

I2S_CF_TXLFULL and I2S_CF_TXRFULL registers are the transmitter data buffer full flags for left and right channel. Flags are set when transmitter data buffer registers are modified. The flags are reset as the left and right data buffer is copied to shifter register.

I2S_CF_RXORUN is the receiver over run flag. It is set when receiver data buffers were full and new frame was received. The flag is reset by writing it to '0'.

I2S_CF_RXLFULL and I2S_CF_RXRFULL are the receiver data buffer full flags for left and right channel. Flags are set when receiver data buffer registers are full. The flags are reset as the left and right data buffer is read.

I2S_CF_INTENA enables the I2S interrupt when set.

I2S_CF_32B register selects between 32-bit (1) and 16-bit (0) data format. This register can be modified only in idle state.

I2S_LEFT, I2S_LEFT_LSB, I2S_RIGHT and I2S_RIGHT_LSB are the left and right data registers for receiver and transmitter. Each write to I2S_LEFT and I2S_RIGHT registers sets the I2S_CF_TXLFULL and I2S_CF_TXRFULL flags. Each read from I2S_LEFT and I2S_RIGHT registers resets the I2S_CF_RXLFULL and I2S_CF_RXRFULL flags. In 16-bit mode the registers I2S_LEFT_LSB and I2S_RIGHT_LSB are not used. In 32-bit mode they are used to transfer 16 LSBs of data.

9.16 Timer Peripheral

VS1010 has three 32-bit timers that can be initialized and enabled independently of each other. If enabled, a timer initializes to its user initialized start value, and starts decrementing every clock cycle. When the value goes past zero, an interrupt request is generated, and the timer initializes to the value in its start value register, and continues downcounting. A timer stays in that loop as long as it is enabled. Each timer has its own interrupt request.

A timer has a 32-bit timer register for down counting and a 32-bit TIMER1_LH register for holding the timer start value written by the processor. Timers have also a 3-bit TIMER_ENA register. Each timer is enabled (1) or disabled (0) by a corresponding bit of the enable register.

9.16.1 Timer Peripheral Registers

Timer Registers

Reg	Type	Reset	Abbrev	Description
0xFE80	r/w	0	TIMER_CF[7:0]	Timer configuration
0xFE81	r/w	0	TIMER_ENA[3:0]	Timer enable
0xFE82	r	0	TIMER_CAP0_LO	Timer0 capturevalue - LSBs
0xFE83	r	0	TIMER_CAP0_HI	Timer0 capturevalue - MSBs
0xFE84	r/w	0	TIMER_T0L	Timer0 startvalue - LSBs
0xFE85	r/w	0	TIMER_T0H	Timer0 startvalue - MSBs
0xFE86	r/w	0	TIMER_T0CNTL	Timer0 counter - LSBs
0xFE87	r/w	0	TIMER_T0CNTH	Timer0 counter - MSBs
0xFE88	r/w	0	TIMER_T1L	Timer1 startvalue - LSBs
0xFE89	r/w	0	TIMER_T1H	Timer1 startvalue - MSBs
0xFE8A	r/w	0	TIMER_T1CNTL	Timer1 counter - LSBs
0xFE8B	r/w	0	TIMER_T1CNTH	Timer1 counter - MSBs
0xFE8C	r/w	0	TIMER_T2L	Timer2 startvalue - LSBs
0xFE8D	r/w	0	TIMER_T2H	Timer2 startvalue - MSBs
0xFE8E	r/w	0	TIMER_T2CNTL	Timer2 counter - LSBs
0xFE8F	r/w	0	TIMER_T2CNTH	Timer2 counter - MSBs

TIMER_CF Bits

Name	Bits	Description
TIMER_CF_CLKDIV	7:0	Master clock divider

TIMER_CF_CLKDIV is the master clock divider for all timer clocks. The generated internal clock frequency $f_i = \frac{f_m}{c+1}$, where f_m is the master clock frequency and c is TIMER_CF_CLKDIV. Example: With a 12 MHz master clock, TIMER_CF_DIV=3 divides the master clock by 4, and the output/sampling clock would thus be $f_i = \frac{12MHz}{3+1} = 3MHz$.

TIMER_ENA Bits		
Name	**Bits**	**Description**
TIMER_ENA_CAP0	3	Enable timer 0 capture
TIMER_ENA_T2	2	Enable timer 2
TIMER_ENA_T1	1	Enable timer 1
TIMER_ENA_T0	0	Enable timer 0

TIMER_ENA register has active high enable bits for each of the three timers. TIMER_ENA_CAP0 enables the timer0 capture register. Timer0 value is stored in TIMER_CAP0_HI and TIMER_CAP0_LO registers when a gpio interrupt occurs.

TIMER_Tx[L/H] register defines the Timer X Startvalue. The 32-bit start value TIMER_Tx[L/H] sets the initial counter value when the timer is reset. The timer interrupt frequency $f_t = \frac{f_i}{c+1}$ where f_i is the master clock obtained with the clock divider and c is TIMER_Tx[L/H].

TIMER_TxCNT[L/H] contains the current counter values. By reading this register pair, the user may get knowledge of how long it will take before the next timer interrupt. Also, by writing to this register, a one-shot different length timer interrupt delay may be realized.

Each timer has its own interrupt, which is asserted when the timer counter underflows.

9.17 Real Time Clock (RTC)

RTC is used for accurate time measurements and storing data when CPU is powered down. The oscillator input clock frequency is 32768 Hz. Real time clock is a 32-bit time keeping up counter which has a resolution of 1 second. Additionally the 8-bit clock divider register value is accessible giving 1/128 seconds resolution. Other functions of VS1010 RTC are time alarm and 32 word register memory for battery backup. The RTC consists of two parts, the Real Time Clock module and its dsp interfacing peripheral. The RTC has its own power network which enables its use when the rest of the system is powered off. The interface between these two is bit-serial.

9.17.1 RTC Peripheral Registers

RTC Interface Registers				
Reg	**Type**	**Reset**	**Abbrev**	**Description**
0xFEA0	r/w	0	RTC_LOW	RTC data register bits [15:0]
0xFEA1	r/w	0	RTC_HIGH	RTC data register bits [31:16]
0xFEA2	r/w	0	RTC_CF[4:0]	RTC if control and status register

RTC_CF Bits		
Name	**Bits**	**Description**
RTC_CF_GSCK	4	Generate serial clock for RTC
RTC_CF_EXEC	3	RTC execute instruction
RTC_CF_RDBUSY	2	Read cycle init and busy flag
RTC_CF_DBUSY	1	Data cycle init and busy flag
RTC_CF_IBUSY	0	Instruction cycle init and busy flag

RTC_LOW and RTC_HIGH are the rtc_if data registers. Write to RTC_CF registers busy bits start a data transfer to/from RTC. When the operation has finished the status bit is reset and result can be read from RTC_HIGH and RTC_LOW registers or RTC_HIGH and RTC_LOW registers were transferred to RTC.

RTC_IBUSY is the instruction cycle initialization register. When RTC_IBUSY is set the current content of RTC_HIGH and RTC_LOW registers is transferred to RTC and latched to its instruction register. When rtc_if is ready it resets the RTC_IBUSY.

RTC_DBUSY is the data cycle initialization register. When RTC_DBUSY is set the current content of RTC_HIGH and RTC_LOW registers is transferred to RTC data buffer. When rtc_if is ready it resets the RTC_DBUSY.

RTC_RDBUSY is the data read cycle initialization register. Before reading rtc a valid instruction must be in RTC instruction register (RTC_I_READRTC, RTC_I_RDDIV128). When RTC_RDBUSY is set the rtc_if first samples the selected rtc register to RTC data buffer. When the data is read to RTC_HIGH and RTC_LOW registers. When rtc_if is ready it resets the RTC_RDBUSY.

RTC_EXEC is used to execute current RTC instruction. Before executing an instruction a valid instruction must be in RTC instruction register (RTC_I_RESET, RTC_I_LOADRTC). For

RTC_I_RESET, RTC_I_LOADRTC instructions the RTC_EXEC register must be set for 1 second before the instruction is ececuted. The user must reset the RTC_EXEC register after this time has elapsed.

RTC instructions are 8-bit codes which are written to RTC_HIGH[15:8] before setting RTC_IBUSY.

RTC Instruction Codes

Instruction	Hex code	Delay	Description
RTC_I_RESET	EB	1/128 s	Reset control registers
RTC_I_LOADRTC	59	1 s	Initialize time counter register
RTC_I_READRTC	56	1/12 MHz	Read time counter register
RTC_I_RDDIV128	C7	1/12 MHz	Read 8-bit divider register (1/128s)
RTC_I_ALARM	AC	1/128 s	Set RTC alarm time

9.18 12-Bit Successive Approximation Register Analog-to-Digital Converter (SAR)

VS1010 has a 12-bit ADC with following features:

- Successive Approximation Register conversion (SAR)
- Up to 7 analog input channels
- Up to 0.1Msps conversion speed
- AVDD voltage as reference
- Continuous or software enabled (once only) operation modes
- input range from 0V to AVDD

Before SAR can be used the following analog control registers must be configured.

Analog configuration for SAR		
Register Name	**Bit Value**	**Description**
ANA_CF0_HIGH_REF	Set/Reset	Select high or low reference
ANA_CF0_REF_ENA	Set	Analog reference
ANA_CF1_SAR_ENA	Set	SAR power down

SAR operation is controlled with configuration register and the 12-bit data is stored in the data register. SAR generates an interrupt as the data register is updated.

SAR Data Register				
Reg	**Type**	**Reset**	**Abbrev**	**Description**
0xFECD	r	0	SAR_DAT[11:0]	12-bit SAR data register

The measured value is 12-bit unsigned data that equals to voltage

$$V_{measured} = \frac{SAR\_DAT}{4095} \times AVDD.$$

i.e. the max value 4095 (0x0fff) gives $V_{measured} = AVDD$

SAR Control/Configuration Register				
Reg	**Type**	**Reset**	**Abbrev**	**Description**
0xFED6	r/w	0x003F	SAR_CF[11:0]	SAR control register

SAR_CF Bits		
Name	**Bits**	**Description**
SAR_CF_SEL	11:8	SAR input selection
SAR_CF_ENA	7	SAR initialize read cycle
SAR_CF_MODE	6	SAR operation mode
SAR_CF_CK[5:0]	5:0	SAR Clock divider register

SAR_CF_ENA is used to start SAR cycle. When this register is set the SAR measures voltage from a given channel and stores the 12-bit value to SAR_DAT register. SAR_CF_ENA is reset when the result is ready and can be read from data register.

SAR_CF_CK[5:0] is used to select the interface clock speed divider. The SAR clock runs at

$$clk_{SAR} = \frac{XTALI}{32 \times (SAR\_CF\_CK+1)}.$$

SAR_CF_MODE selects between continuous mode ('1') and run-once ('0') modes.

SAR input channel is selected with SAR_CF_SEL[3:0] register. This register is double buffered against possible conversion time changes. The register is sampled as the SAR is enabled or it is in idle state. In continuous mode the register is sampled at the end of each conversion.

SAR input channel selection					
Decimal	**Hex**	**LFGA-68 Pin**	**LQFP-48 Pin**	**Max Voltage**	**Description**
14	0xE	48	34	3.6 V	aux0
12	0xC	49	35	3.6 V	aux1
10	0xA	Internal	Internal	5.25 V	Divided VHIGH/2
8	0x8	Internal / 59 [1]	Internal / 41 [1]	3.6 V	RCAP 1.2 or 1.6 V reference voltage
7	0x7	50	nc	3.6 V	aux3
6	0x6	Internal / 55 [2]	Internal / 37 [2]	1.95 V	RTC voltage
5	0x5	51	36	3.6 V	aux2
4	0x4	53	nc	3.6 V	aux5
2	0x2	52	nc	3.6 V	aux4
0	0x0	54	nc	3.6 V	aux6

[1] Although connected to a pin, this voltage is normally generated by VS1010. In a typical case this pin should not be driven externally.

[2] Maximum allowed external voltage to this pin is RTCVDD (1.95 V). Failing to follow this limitation may break VS1010's internal digital circuitry.

Other pin values than mentioned in the table are not allowed.

9.19 Mems Mic Interface

VS1010 has a stereo decimation filter for Mems Mic modules. Decimation ratio is 16. The filter decimates 3MHz 1-bit Mems Mic output to 192kHz and 15 bits.

Mems Mic Interface registers				
Reg	**Type**	**Reset**	**Abbrev**	**Description**
0xFEDA	r/w	0x0000	MEMSMIC_CF[3:0]	Mems Mic control register
0xFEDB	r	0x0000	MEMSMIC_L	Mems Mic left output
0xFEDC	r	0x0000	MEMSMIC_R	Mems Mic right output

MEMSMIC_CF is the control and configuration register. When Mems Mic interface is used also either PERIP_CF_MEMSCK1 or PERIP_CF_MEMSCK0 must be set (see Section 9.5.2) These registers enable the 3MHz Mems Mic interface clock pin output.

MEMSMIC_L and MEMSMIC_R are the Mems Mic output registers. The 15-bit decimated data is in MSB bits. Bit 0 is always zero. Data is in signed format.

MEMSMIC_CF Bits		
Name	**Bits**	**Description**
MEMSMIC_CF_LEDGE	3	Mems Mic left channel select
MEMSMIC_CF_REDGE	2	Mems Mic right channel select
MEMSMIC_CF_SELIO	1	Mems Mic interface pin select
MEMSMIC_CF_ENA	0	Mems Mic decimation filter enable

MEMSMIC_CF_LEDGE and MEMSMIC_CF_REDGE select channel polarities. When bit is reset the 1-bit channel data is sampled at clock fall edge. When bit is set the data is sampled at rising clock edge.

MEMSMIC_CF_SELIO selects the Mems Mic interface pins. When MEMSMIC_CF_SELIO is reset the interface uses pins gpio1(14) and gpio0(1) for clock and data. When register is set pins gpio0(10) and gpio0(9) are used.

MEMSMIC_CF_ENA is active high enable for Mems Mic decimation filter. Resetting this register resets the filter.

9.20 Pulse Width Modulation Unit

VS1010 has a PWM output which can be programmed to generate any pulse width within 256 XTALI clock periods.

PWM Registers				
Reg	**Type**	**Reset**	**Abbrev**	**Description**
0xFED4	r/w	0	PWM_FRAMELEN[7:0]	PWM frame length, 2 - 255 clock cycles
0xFED5	r/w	0	PWM_PULSELEN[7:0]	PWM pulse width, 0 - 255 clock cycles

PWM_FRAMELEN defines the pwm frame length. Values 0 and 1 are not allowed and they place the unit in powerdown (output is zero). With frame values > 1 the pwm output is enabled with rising edge at start of frame and falling edge at PWM_PULSELEN. If PWM_PULSELEN is zero the output is always zero. If PWM_PULSELEN > PWM_FRAMELEN the output is always at logic high state.

PWM unit generates interrupt request at the start of each frame.

In VS1010 power-up as the PWRBRTN pin is high the pwm output generates an oscillation for external powering circuitry. The oscillation reguires that there is an external pull-up resistor connected to the pwm pin.

PWM start-up oscillator				
Item	**Min**	**Typical**	**Max**	**Description**
Pull-up resistor		100 kΩ		Value of external pull-up resistor
Start-up frequency		370 kHz		Start-up oscillation frequency

10 Document Version Changes

This chapter describes the most important changes to this document.

Version 0.04, 2017-07-25

- Register descriptions updated.

Version 0.03, 2016-12-29

- Analog characteristics updated.
- LQFP-48 package option restrictions added.

Version 0.02, 2016-12-13

- Removed features left over from VS1005 datasheet.

Version 0.01, 2016-11-29

- First, very preliminary release. All features and figures are subject to change.

11 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